เครื่องตรวจสอบคุณสมบัติไดโอดและทรานซิสเตอร์
DIODE & TRANSISTER CHARACTERISTIC ANALYSER

โดย
นางสาวเพ็ชรดา จินชัย
นางสาวสุวรรณา เทศนาคติธรรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมอิเลคทรอนิคส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
033230

ปริญญานิพนธ์ปีการศึกษา 2536

ภาควิชาอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตรวจสอบคุณสมบัติไดโอดและทรานซิสเตอร์บนคอมพิวเตอร์

ผู้จัดทำ

1. นางสาวเพ็ชรดา จินชัย 33100268
2. นาวสาวสุวรรณา เทศนาคติธรรม 33100460

_____ อาจารย์ที่ปรึกษา

(อ.เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องตรวจสอบคุณสมบัติของไดโอดและทรานซิสเตอร์

DIODE & TRANSISTOR CHARACTERISTIC ANALYZER

โดย นางสาว เพ็ชรดา จินชัย

นางสาว สุวรรณา เทศนาคติธรรม

อาจารย์ที่ปรึกษา อ. เทอดศักดิ์ ลิ่วหาทอง

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้เสนอแนวทางการสร้าง เครื่องตรวจสอบคุณสมบัติของไดโอดและทรานซิสเตอร์โดยดูจากกราฟ I-V CHARACTERISTIC โดยแสดงผลออกทางจอภาพของเครื่องคอมพิวเตอร์ ทำให้สะดวกต่อการพิจารณาเลือกแรงดันและกระแสไบอัสของทรานซิสเตอร์ให้เหมาะสมต่อการทำงาน ในกราฟ V-I CHARAC TERISTIC นี้ เราจะจำกัดแรงดันระหว่างคอลเลคเตอร์และอิมิตเตอร์ (V_{CE}) ไว้ ที่ค่าสูงสุดระหว่าง +10 ถึง -10 โวลท์ และกระแสไบอัสที่ขาเบส (I_b) มีค่าสูงสุด ระหว่าง +100 ถึง -100 มิลลิแอมป์ ทั้งยังสามารถเลือก Range ของเอาท์พุต เพื่อให้ดูจุดต่างๆ ได้อย่างชัดเจน และสามารถเลือก step ของ I_b ได้เพื่อให้สามารถดูค่าของ I_c ที่เปลี่ยนแปลงต่อ V_{CE} ที่ I_b ค่าต่างๆ

ABSTRACT

This thesis introduce method to create V-I characteristic graph of transistor or diode by PC monitoring to consider the transistor operating point. Conviniently it is easy to select proper bias voltage and current at the operating point. In this thesis, we shall supply voltage drop collector and emitter (V_{CE}) in maximum range +10 to -10 volts and current bias to base (I_B) limit in maximum range +100 to -100 mA. Additonally,this can select range of display collector current (I_c) for looking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ในการเลือกใช้งานอุปกรณ์ต่างๆ เราควรจะรู้ถึงคุณสมบัติของอุปกรณ์ตัวนั้นให้ดี ตัวอย่างเช่น ในการเลือกใช้ไดโอดหรือทรานซิสเตอร์ ก็ควรรู้ถึงคุณสมบัติของไดโอดหรือทรานซิสเตอร์ก่อน ซึ่งในปัจจุบันนี้ก็มีข้อมูลเกี่ยวกับทรานซิสเตอร์และไดโอดอยู่ใน DATA BOOK ซึ่งจะเป็นตารางซึ่งจะบอกคุณสมบัติต่างๆ มาพอเป็นตัวอย่างเท่านั้น จะไม่ทราบค่าที่จุดทำงานอื่นได้ สิ่งที่จะบอกได้ละเอียดกว่าตารางก็คือกราฟคุณสมบัติของไดโอดและทรานซิสเตอร์นั่นเอง ดังนั้น Project นี้จึงทำการสร้าง เครื่องสร้างกราฟคุณสมบัติไดโอดและทรานซิสเตอร์บนคอมพิวเตอร์เพื่อให้ผู้ที่ต้องการใช้งานไดโอดหรือทรานซิสเตอร์นั้นสามารถทราบถึงจุดต่างๆ ในการทำงานของไดโอดและทรานซิสเตอร์ได้

นางสาวเพ็ชรดา จินชัย

นางสาวสุวรรณา เทศนาคติธรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	
บทที่ 1 โครงสร้างพื้นฐานไดโอดและทรานซิสเตอร์	1
กราฟคุณสมบัติของไดโอดและทรานซิสเตอร์	3
บทที่ 2 โครงสร้างของเครื่องสร้างกราฟไดโอดและทรานซิสเตอร์	5
โครงสร้างทางสถาปัตยกรรมของ 8031	6
Memory กับ 8031	34
การเชื่อมต่อระหว่าง Memory กับ CPU	44
บทที่ 3 การตรวจสอบไดโอดและทรานซิสเตอร์	47
การตรวจสอบไดโอด	47
การตรวจสอบทรานซิสเตอร์	47
บทที่ 4 ส่วน Control	50
บทที่ 5 วงจรไบอัสและวัดค่ากระแส	55
วงจร Current Source Voltage Monitor	57
วงจร Voltage Source Current Monitor	60
ส่วนของการเลือก $R_1 R_m R_c$	62
บทที่ 6 Software	63
บทที่ 7 การทดลอง	67
สรุปและวิจารณ์	72
บรรณานุกรม	73
Source Code ของ 8031 และ turbo C	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

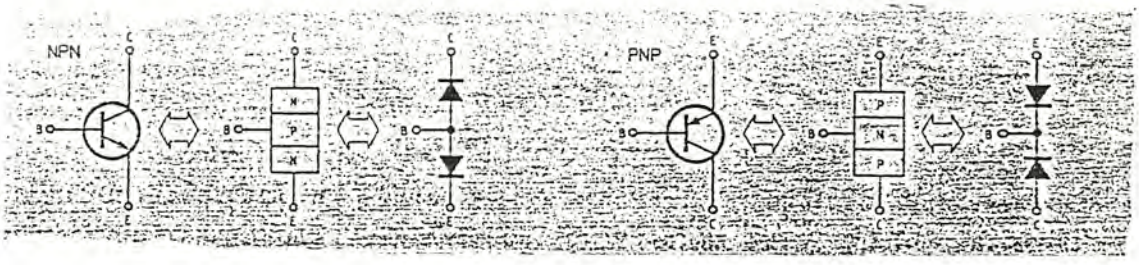
โครงสร้างพื้นฐานไดโอดและทรานซิสเตอร์

การใช้อุปกรณ์ต่างๆ จะได้ผลดีเมื่อเข้าใจถึงโครงสร้างพื้นฐานภายในอุปกรณ์นั้น ตัวอย่างเช่นไดโอดหรือทรานซิสเตอร์ เมื่อเราต้องการใช้งานก็ควรทราบถึงโครงสร้างภายในด้วย โดยไดโอดนั้นลักษณะโครงสร้างจะเป็นรอยต่อพี-เอ็น ซึ่งจะประพฤติตัวคล้ายสวิตช์อิเลคทรอนิกส์เนื่องจากเมื่อไดโอดถูกไบแอสตรง (forward bias) ก็จะสามารถนำกระแสได้แต่เมื่อถูกไบแอสกลับ (reverse bias) ก็จะเป็นตัวกั้นไม่ให้กระแสผ่านเลย

สำหรับไดโอดในอุดมคติขณะที่ถูกไบอัสกลับจะไม่มีกระแสไหลผ่านรอยต่อ พี-เอ็นเลย แต่ในความเป็นจริงแล้วไม่เป็นเช่นนั้น โดยจะมีกระแสบางส่วนไหลผ่านรอยต่อ อันเนื่องมาจากอุณหภูมิ และปริมาณสารเจือ (impurity) ภายในสารกึ่งตัวนำนั้น ถึงแม้ว่าสารเจือจะมีปริมาณน้อยมากก็ตาม แต่ยังก็มีผลทำให้เกิดกระแสไหลผ่านรอยต่อ พี-เอ็น ในขณะไบแอสกลับได้ ซึ่งเราสามารถวัดปริมาณกระแสส่วนนี้ได้ กระแสที่เกิดขึ้นนี้เรียกว่ากระแสรั่วไหล (I_R)

คุณสมบัติอีกอย่างหนึ่งที่สำคัญสำหรับไดโอด ก็คือ แรงดันพังทลาย (Breakdown Voltage) ซึ่งคือค่าแรงดันไบแอสกลับที่เมื่อป้อนให้แก่รอยต่อของไดโอด แล้วมีผลทำให้กระแสไหลผ่านไดโอดอย่างมากมาย ซึ่งจะสามารถทำให้ไดโอดเสียหายได้ ซึ่งกระแสรั่วไหลและแรงดันพังทลายได้ โดยดูจากกราฟ I-V characteristic ของไดโอด

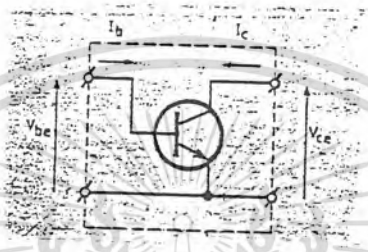
ส่วนภายในตัวทรานซิสเตอร์จะเสมือนมีไดโอดสองตัวต่อกลับขั้วกันอยู่ดังรูปที่ 1.1



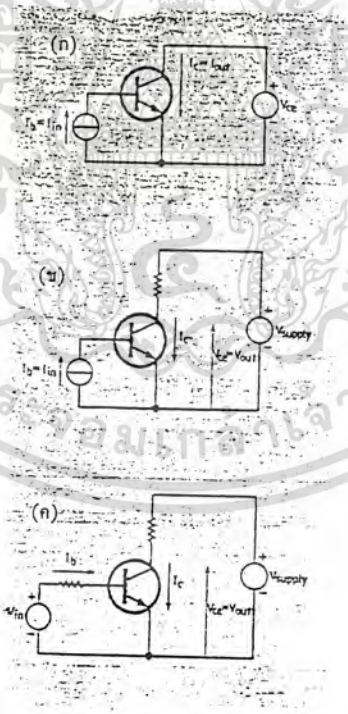
รูปที่ 1.1 โครงสร้างพื้นฐานของทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติทรานซิสเตอร์จะมี 3 ขา ประกอบด้วยขาเบส (B) ขาคอลเลคเตอร์ (C) และ ขาอิมิเตอร์ (E) แต่ในการอธิบายคุณสมบัติต่างๆ นั้นบางครั้งจะใช้วงจร 4 ขั้วในการอธิบายดังในรูป 1.2 เมื่อพิจารณาทางไฟฟ้าจะมีค่าที่สำคัญอยู่ 4 ค่า คือ กระแสอินพุตที่ขาเบส (I_b) แรงดันอินพุต (V_{BE}) กระแสเอาท์พุตที่คอลเลคเตอร์ (I_c) และแรงดันเอาท์พุต (V_{CE})



รูปที่ 1.2 เป็นวงจร 4 ขั้วมีขาอิมิเตอร์เป็นขาร่วมระหว่างอินพุต



รูปที่ 1.3 วงจรใช้งานทรานซิสเตอร์พื้นฐาน

(ก) วงจรขยายกระแสเป็นกระแส

(ข) วงจรขยายกระแสเป็นแรงดัน

(ค) วงจรขยายแรงดันเป็นแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

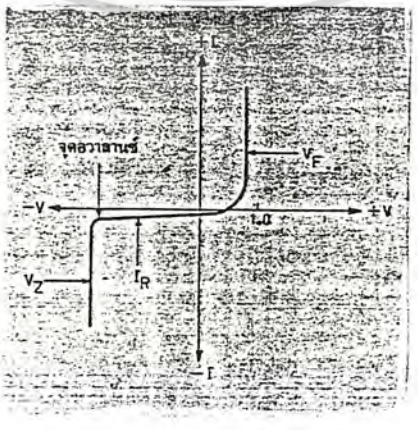
ในรูปที่ 1.3(ก) แสดงถึงกระแสคอลเลคเตอร์ที่มีค่ามากกว่ากระแสเบสตามค่าของ H_{FE} ซึ่งแรงดันอินพุตที่ป้อนเข้ามาจะต้องมากกว่า 0.6-0.4 V (แรงดันตกคร่อมไดโอดเบส-อิมิเตอร์) ทรานซิสเตอร์จึงจะเริ่มทำงาน กระแสคอลเลคเตอร์ที่เกิดขึ้นจะไม่ขึ้นกับแรงดันไฟเลี้ยงวงจรแต่จะขึ้นอยู่กับกระแสเบสเพียงอย่างเดียว เรียกลักษณะวงจรนี้ว่าเป็นวงจรขยายกระแส

เมื่อพิจารณาดูแรงดันเอาท์พุตคอลเลคเตอร์ในรูปที่ 1.3(ข) จะพบว่าแรงดันที่ได้จากแรงดันตกคร่อมของตัวต้านทานที่คอลเลคเตอร์ เมื่อมีกระแสคอลเลคเตอร์ผ่านรูปแบบวงจรนี้เป็นลักษณะ วงจรขยายแบบหนึ่ง ซึ่งเปลี่ยนจากกระแสอินพุตให้เป็นแรงดันทางเอาท์พุต เมื่อวัดแรงดันที่คอลเลคเตอร์เทียกกับอิมิเตอร์จะพบว่าเมื่อกระแสเบสเพิ่มขึ้นแรงดันที่คอลเลคเตอร์จะลดลง จึงเรียกการทำงานตอนนี้ว่าเป็นวงจรขยายแบบกลับเฟส (Inverting Amplifier)

ลักษณะการใช้ทรานซิสเตอร์อีกแบบหนึ่ง คือ ใช้วงจรขยายในลักษณะที่เปลี่ยนแรงดันอินพุตเป็นแรงดันเอาท์พุตที่มากกว่าเดิม ดังรูปที่3(ค) ปกติจะไม่ป้อนแรงดันที่เกิน 0.7 V ให้กับขาเบสโดยตรงเพราะอาจจะเกิดความเสียหายขึ้นได้ ดังนั้นในการใช้งานส่วนใหญ่จะต้องเปลี่ยนจากแรงดันให้เป็นกระแสก่อน แล้วจึงป้อนให้แก่ทรานซิสเตอร์โดยการใช้ตัวต้านทานต่ออนุกรมกับขาเบส

<u>กราฟคุณสมบัติของไดโอดและทรานซิสเตอร์</u>

กราฟคุณสมบัติของไดโอดจะแสดงถึง ความสัมพันธ์ของแรงดันไบแอสกับปริมาณกระแสที่ไหลผ่านไดโอด

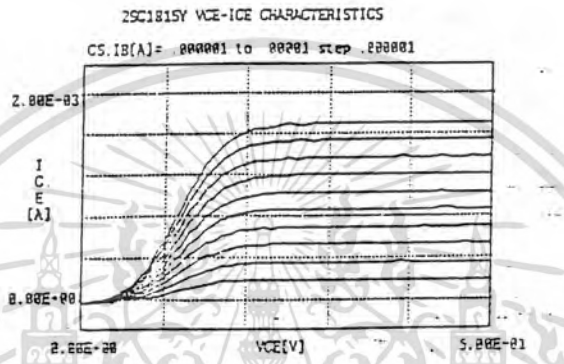


รูปที่ 1.4 กราฟคุณสมบัติของไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟจะเห็นได้ว่า เมื่อเราป้อนแรงดันไบแอสตรงให้กับไดโอดเกิน 0.6-0.7V ไดโอดก็จะนำกระแส และเมื่อป้อนแรงดันไบแอสกลับก็จะเกิดกระแสรั่วไหล และถึงจุดๆ หนึ่งก็จะเกิดแรงดันพังทลายขึ้นซึ่งจากกราฟจะสามารถเห็นได้ชัดว่า กระแสรั่วไหลเท่าไร แรงดันพังทลายเท่าไร forwrd bias เท่าไร

ส่วนของทรานซิสเตอร์กราฟคุณสมบัติของทรานซิสเตอร์สามารถแสดงได้ดังรูปที่ 1.5



รูปที่ 1.5 กราฟคุณสมบัติทางด้านเอาท์พุตของทรานซิสเตอร์เบอร์ 2SC1815

จากกราฟคุณสมบัติของทรานซิสเตอร์เบอร์ 2SC1815 เป็นค่าที่พล็อตขึ้นระหว่างกระแสคอลเลคเตอร์กับแรงดันคร่อมคอลเลคเตอร์กับอิมิเตอร์ โดยมีกระแสเบสเป็นค่าพารามิเตอร์กำหนด ในทางอุดมคติทรานซิสเตอร์จะต้องให้เส้นของกระแสคอลเลคเตอร์ที่ขนานไปกับแกนนอนเมื่อ V_{BE} มีค่ามากกว่า 0.7 V ซึ่งแสดงว่ากระแสคอลเลคเตอร์คงที่เมื่อไบอัสด้วยกระแสเบสค่าหนึ่ง และจะเปลี่ยนแปลงเมื่อกระแสเบสเปลี่ยนไปเท่านั้น ไม่เปลี่ยนแปลงตามแรงดันคร่อมคอลเลคเตอร์กับอิมิเตอร์ สำหรับกราฟในรูปที่ 1.5 นั้นไม่ใช่กราฟคุณสมบัติในทางอุดมคติแต่เป็นกราฟที่ได้จากการทดสอบจริงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

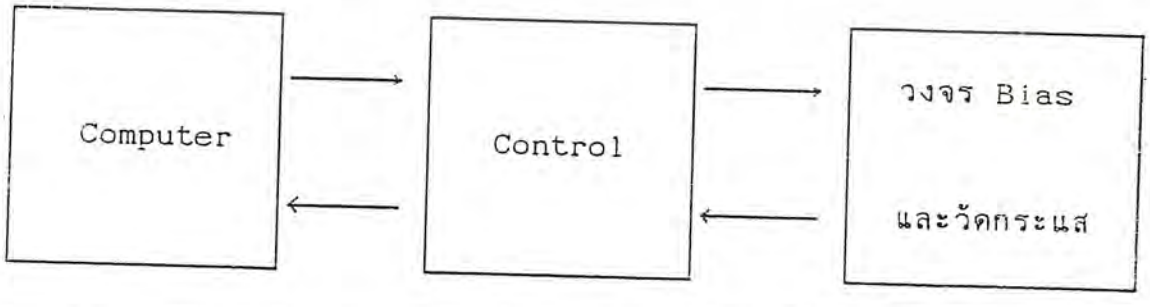
โครงสร้างของเครื่องตรวจสอบคุณสมบัติของไดโอดและทรานซิสเตอร์

Project นี้ประกอบด้วยส่วนหลักๆ 3 ส่วนด้วยกันคือ

1. Computer จะเป็นส่วนที่ติดต่อกันระหว่างผู้ใช้งานกับวงจรไบอัสและวัดค่ากระแส โดยผู้ใช้สามารถ ป้อนพารามิเตอร์ต่างๆ ที่ต้องใช้ในการวัดให้กับคอมพิวเตอร์ ซึ่งคอมพิวเตอร์จะจัดการส่งข้อมูลให้กับส่วน Control ต่อไป และเมื่อวงจร test ทำงานเสร็จเรียบร้อยข้อมูลจะถูกส่งกลับมาที่คอมพิวเตอร์โดยผ่านทางส่วน Control เพื่อแสดงผลในรูปแบบที่เข้าใจได้

2. ส่วน Control ทำหน้าที่รับพารามิเตอร์จากคอมพิวเตอร์และคำนวณหาค่า I_b และ V_{CE} ตามค่าพารามิเตอร์ที่ได้รับ เพื่อส่งให้กับวงจรไบอัสและวัดค่ากระแส และรับค่าจากวงจรนี้ ส่งให้กับคอมพิวเตอร์นำกลับไปแสดงผล โดยส่วน Control นี้ใช้ Microcontroller 8031 ในการควบคุมการทำงานทั้งหมด

3. วงจรไบอัสและวัดค่ากระแส เป็นส่วนที่ใช้วัดตัวทรานซิสเตอร์จริงๆ ดังนั้นสัญญาณต่างๆ ในวงจรนี้จะต้องเป็นสัญญาณ analog ทั้งหมด ซึ่งทำได้โดยการใช้ A/D Converter และ D/A Converter ใช้ในการแปลงสัญญาณ analog เป็น digital และ digital เป็น analog ตามลำดับ โดยวงจรนี้จะมีลักษณะต่างๆ ไม่เหมือนกันขึ้นอยู่กับการทำงาน เช่น จ่ายกระแส I_b จ่าย voltage (V_{CE}) เมื่อวงจร test แล้วข้อมูลที่ได้จากการวัดจะต้องถูกส่งกลับไปที่ ส่วน Control เพื่อส่งให้คอมพิวเตอร์ทำการแสดงผลในรูปแบบต่างๆ กันได้



รูปที่ 2.1 Block Diagram ของเครื่องตรวจสอบคุณสมบัติไดโอดและทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากส่วน Control ได้ใช้ 8031 ในการทำงานดังนั้นจะขอกล่าวถึงรายละเอียดเกี่ยวกับ 8031 ก่อน

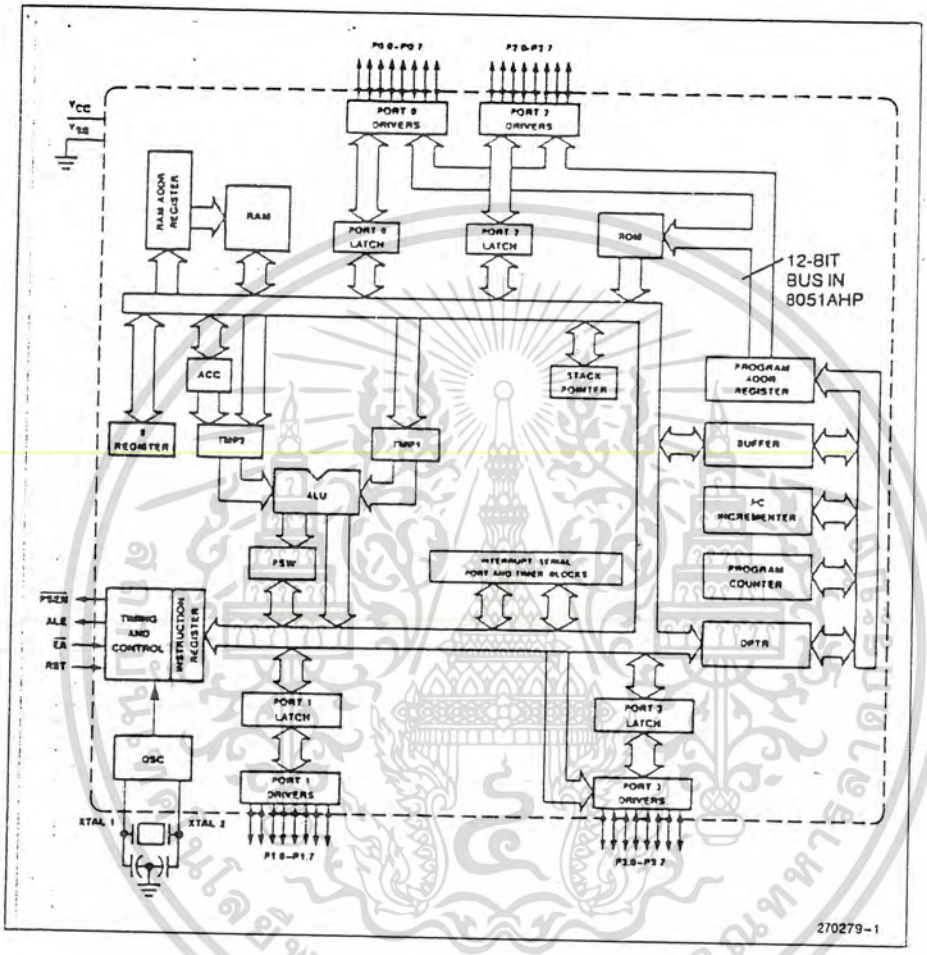
โครงสร้างทางสถาปัตยกรรมของ 8031

คุณสมบัติที่สำคัญของ 80C31BH

- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) อยู่บนชิพจำนวน 128 Byte
- สามารถอ้างอิงหน่วยความจำที่เก็บโปรแกรมและข้อมูลซึ่งอยู่ภายนอกชิพได้ถึง 64 Kbyte
- คำสั่งส่วนใหญ่ใช้เวลา μs เมื่อใช้ความถี่ 12 MHz
- มีพอร์ทแบบ 2 ทิศทางจำนวน 4 พอร์ทๆ ละ 8 บิท ซึ่งสามารถใช้งานเป็นพอร์ทที่มีขนาด 1 บิท ได้ ทำให้มีพอร์ทขนาด 1 บิทใช้งานทั้งสิ้น 32 พอร์ท
- รับส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถตั้งค่าอัตราเร็วในการรับส่งข้อมูล (BAUD RATE) ได้ตั้งแต่ 300 Hz - 19.2 KHz
- จัดลำดับความสำคัญของสัญญาณอินเทอร์รัพท์ได้ 2 ระดับ
- มี TIMER/COUNTER สำหรับนับจำนวนและเวลาขนาด 16 บิท จำนวน 2 ตัว
- สามารถเข้าถึงข้อมูลได้ทั้งแบบเป็นไบท์และแบบบิท
- มีคำสั่งที่ใช้คูณและหารเลขขนาด 8 บิท
- สามารถประมวลผลแบบ Boolean เพื่อใช้ในงานควบคุมโดยเฉพาะ

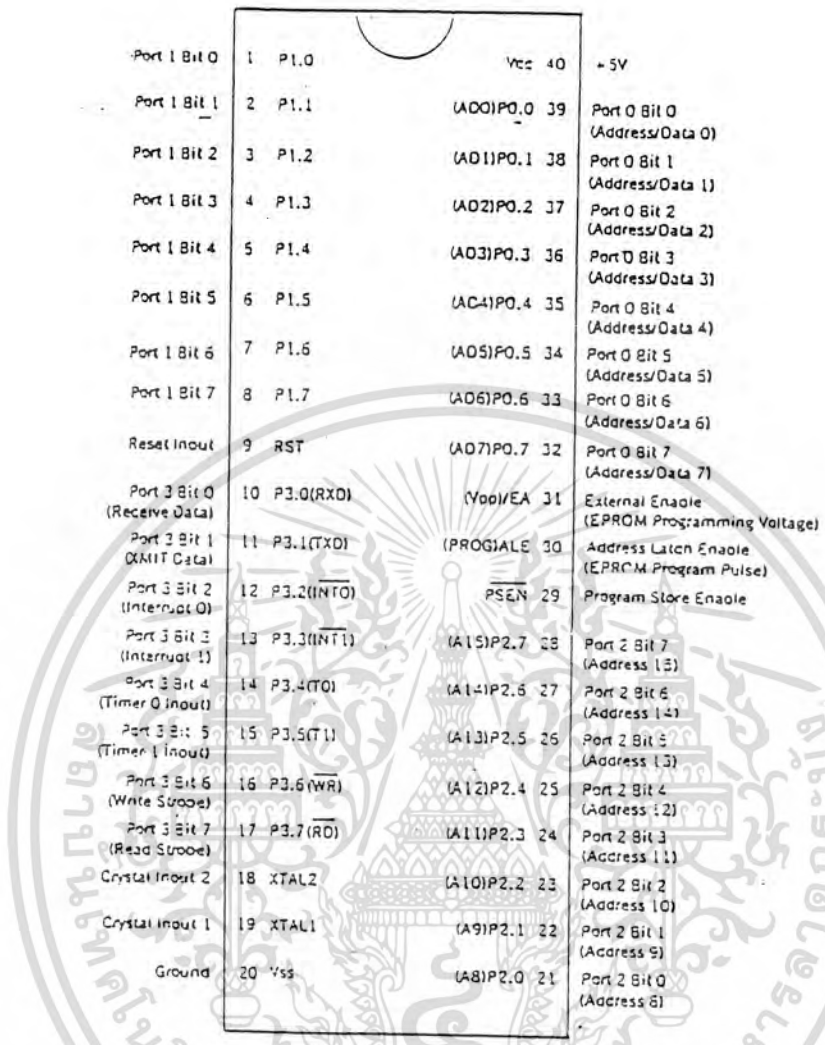
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ 8031 microcontroller แสดงดังรูป



รูปที่ 2.2 โครงสร้างของไมโครคอนโทรลเลอร์ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

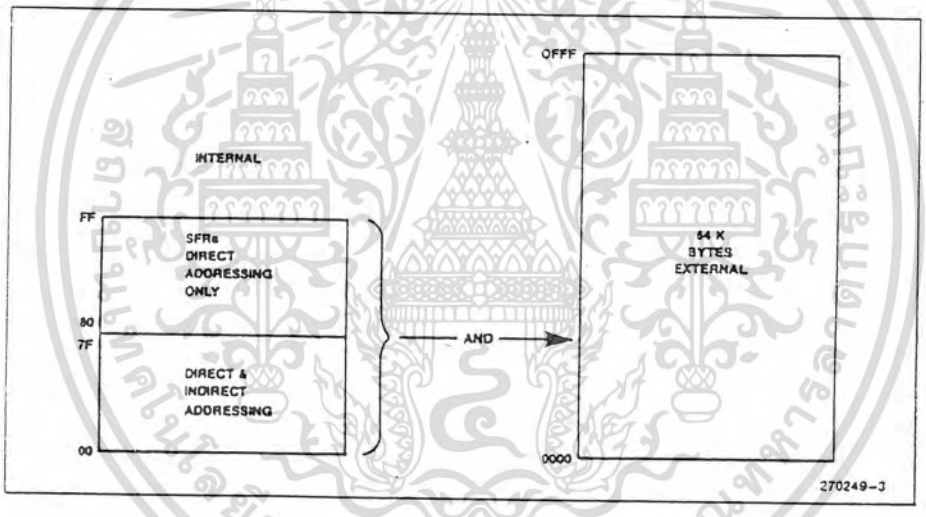


รูปที่ 2.3 แสดงรายละเอียดของขาต่างๆ บนไมโครคอนโทรลเลอร์ 8031

ไมโครคอนโทรลเลอร์ 8031 จะแบ่งหน่วยความจำออกเป็น 2 ส่วน คือ PROGRAM MEMORY (ROM) ซึ่งใช้สำหรับเก็บโปรแกรมที่ควบคุมการทำงาน หน่วยความจำส่วนที่ 2 คือ DATA MEMORY (RAM) ซึ่งใช้เก็บข้อมูลระหว่างการทำงาน ดังแสดงในรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Memory



DATA MEMORY

รูปที่ 2.4 Program Memory และ Data Memory

PROGRAM MEMORY

PROGRAM MEMORY ในไมโครคอนโทรลเลอร์ 8031 จะถูกแบ่งออกเป็น 2 ส่วนคือ INTERNAL PROGRAM MEMORY (ส่วนที่อยู่ในชิพ) และ EXTERNAL PROGRAM MEMORY (ส่วนที่อยู่ภายนอกชิพ)

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
033232

DATA MEMORY

DATA MEMORY ในไมโครคอนโทรลเลอร์ 8031 จะแบ่งออกเป็น 2 ส่วน INTERNAL DATA MEMORY (ส่วนที่อยู่ในชิพ) และ EXTERNAL DATA MEMORY (ส่วนที่อยู่ภายนอกชิพ) โดย INTERNAL DATA MEMORY จะแบ่งออกเป็น 2 ส่วน ดังนี้

- INTERNAL RAM
- SFR (SPECIAL FUNCTION REGISTER)

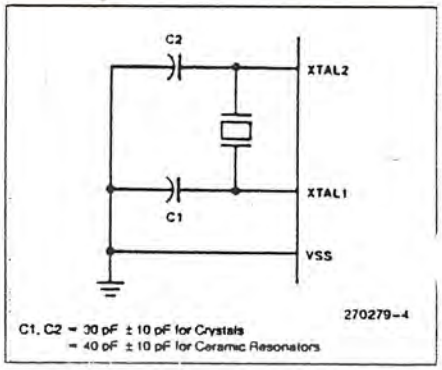
INTERNAL RAM คือ INTERNAL DATA MEMORY ที่ใช้เป็น DATA MEMORY สำหรับเก็บข้อมูล ภายในขณะทำงานส่วน SFR คือ INTERNAL DATA MEMORY ที่ใช้เป็นรีจีสเตอร์สำหรับควบคุมและบอกสถานะของซีพียู

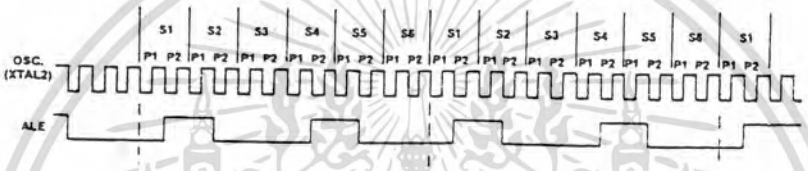
ไมโครคอนโทรลเลอร์ 8031 จะมี INTERNAL RAM 128 Byte ในส่วนของ EXTERNAL DATA MEMORY และ EXTERNAL PROGRAM MEMORY จะอยู่ภายนอกไมโครคอนโทรลเลอร์ซึ่งผู้ใช้ต้องติดตั้งเพิ่มเองโดยสามารถติดต่อกับไมโครคอนโทรลเลอร์ 8031 ผ่านขา 32-39 ที่ทำหน้าที่เป็น ADDRESS BUS ไบท์ต่ำ (A0-A7) และเป็น DATA BUS ด้วย ส่วน ADDRESS BUS ไบท์สูงจะใช้ขา 21-28 ซึ่งจะเห็นว่าตำแหน่งขาที่นำมาใช้ในการติดต่อหน่วยความจำภายนอกนี้จะตรงกับพอร์ท 0 และพอร์ท 2 ตามลำดับ ดังนั้นหากผู้ใช้จำเป็นต้องติดตั้งหน่วยความจำภายนอกเพิ่มขึ้น จะเหลือพอร์ทสำหรับใช้งานอื่นๆ เพียง 2 พอร์ท(พอร์ท 1 และพอร์ท 3) เท่านั้น

8031 oscillator and clock

หัวใจการทำงานไมโครคอนโทรลเลอร์ 8031 คือ วงจรที่ใช้ในการกำเนิดสัญญาณนาฬิกา ให้แก่ การทำงานภายในทั้งหมด ซึ่งจะสร้างสัญญาณนาฬิกาโดยการนำขา XTAL1 และ XTAL2 ของ 8031 มาต่อเข้ากับวงจร oscillator (โดยปกติแล้วจะใช้ quartz crystal และ capacitor) ดังรูปที่ 2.5 ซึ่งความถี่ของ crystal จะเป็นสัญญาณนาฬิกาให้แก่ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.5 แสดงการต่อวงจร oscillator ที่ใช้กับไมโครคอนโทรลเลอร์ 8031

ฮาร์ดแวร์ที่อยู่ในชิพไมโครคอนโทรลเลอร์ 8031 ประกอบด้วย

- PORT DRIVERS และการทำงานทั้งในลักษณะที่เป็นพอร์ทและในการทำงานที่เป็น BUS ของพอร์ท 0 และพอร์ท 2
- TIMER/COUNTER
- SERIAL INTERFACE
- INTERRUPT SYSTEM
- RESET
- REDUCED POWER MODES ใน CHMOS DEVICE

SPECIAL FUNCTION REGISTERS

เนื้อที่ของหน่วยความจำในส่วน SFR แสดงได้ดังรูป 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

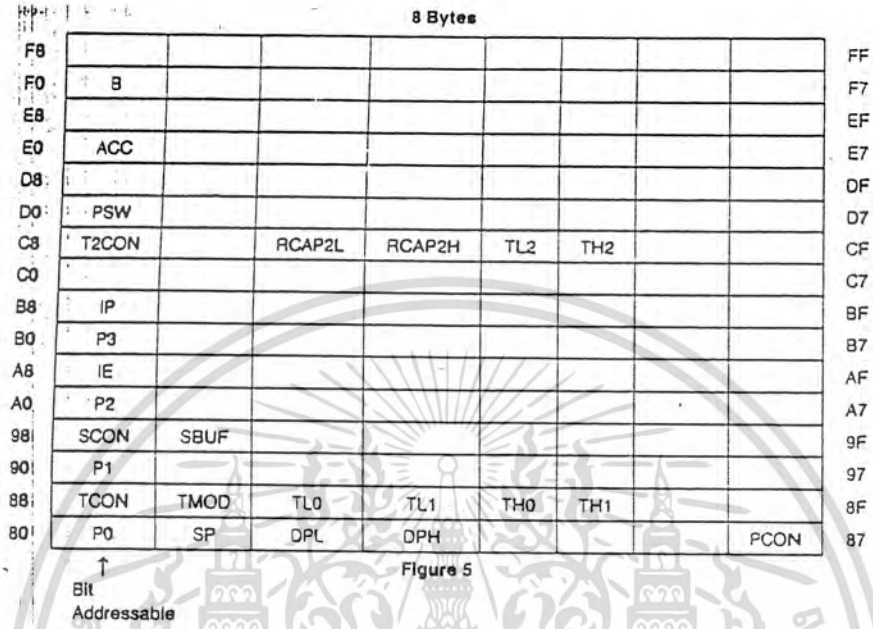


Figure 5

รูปที่ 2.6 แสดง SFR MEMORY MAP

จะสังเกตได้ว่าพื้นที่หน่วยความจำส่วนนี้ไม่ได้ถูกใช้เป็น SFR ทั้งหมด ตำแหน่งของหน่วยความจำในส่วนที่ไม่ได้ถูกใช้จะไม่มีอยู่ในชิพ การอ่านค่าจากตำแหน่งเหล่านี้จะให้ค่าข้อมูลที่ไม่แน่นอน ส่วนการเขียนข้อมูลไปยังตำแหน่งเหล่านี้ จะไม่มีผลใดๆ เกิดขึ้น

โปรแกรมของผู้ใช้ไม่ควรทำการเขียนค่าไปยังตำแหน่งที่ไม่ได้ถูกใช้เหล่านี้เพราะในอนาคตตำแหน่งเหล่านี้อาจถูกใช้เพิ่มเติมคุณสมบัติใหม่ๆ ที่อาจจะมีเพิ่มขึ้นได้ ในกรณีเหล่านี้การรีเซท หรือ INACTIVE VALUEs ของบิทที่อาจจะใช้ในคุณสมบัติที่เพิ่มขึ้นมักจะมีค่าเป็น 0 ส่วน ACTIVE VALUE จะมีค่าเป็น 1

หน้าที่ของ SFRs สามารถอธิบายได้ดังนี้

ACCUMULATOR

ACC เป็นชื่อย่อของ Accumulator Register แต่ในคำสั่งที่มีการระบุว่าต้องกระทำกับ ACCUMULATOR จะอ้างถึง ACCUMULATOR โดยใช้ตัวอักษรย่อ A

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B REGISTER

รีจิสเตอร์ Bจะใช้ในการกระทำคำสั่งคูณและหาร สำหรับในคำสั่งอื่นก็สามารถถูกใช้ได้เหมือนรีจิสเตอร์ทั่วๆ ไปได้

PROGRAM STATUS WORD

PSW ใช้เก็บข้อมูลเกี่ยวกับสถานะของฮาร์ดแวร์ในการทำโปรแกรม

STACK POINTER

STACK POINTER เป็นรีจิสเตอร์ขนาด 8 บิทซึ่งจะถูกทำการเพิ่มค่าก่อนที่จะมีการเก็บข้อมูลในกระว่างการกระทำคำสั่ง PUSH และ CALL STACK อาจจะอยู่บริเวณใดก็ได้ในหน่วยความจำ ส่วนที่เก็บข้อมูลภายในไมโครคอนโทรลเลอร์ 8031 แต่ค่าใน STACK POINTER จะถูกกำหนดค่าเริ่มต้น เป็น 07H หลังจากการรีเซท 8031

DATA POINTER

DATA POINTER ประกอบด้วยรีจิสเตอร์ขนาด 8 บิทจำนวน 2 ตัว คือ DPH และ DPL โดย DPH จะเป็น High Byte ส่วน DPL เป็น Low Byte หน้าที่หลักของรีจิสเตอร์ทั้งสองคือเก็บค่าตำแหน่งขนาด 16 บิท ซึ่งอาจจะใช้งานทั้ง 16 บิทหรือแยกทีละ 8 บิทก็ได้

PORT 0 TO 3

P0, P1, P2, P3 เป็น SFR LATCHES ของพอร์ท 0,1,2,3 ตามลำดับ

SERIAL DATA BUFFER

SERRIAL DATA BUFFER มี 2 ตัว คือ Transmit Buffer และ Recieve Buffer ซึ่งอยู่ตำแหน่งเดียวกัน

เมื่อข้อมูลถูกส่งไปยัง SBUF ข้อมูลจะถูกส่งไปที่ Transmit Buffer และเก็บไว้เพื่อทยอยส่งข้อมูลแบบอนุกรมต่อไป (การเขียนข้อมูลไปที่ SBUF จะเป็นการเริ่มต้นการส่งข้อมูล)

เมื่อข้อมูลถูกอ่านมาจาก SBUF ข้อมูลจะได้รับมาจาก Recieve Buffer

TIMER REGISTER

REGISTER PAIRS (TH0,TL0),(TH1,TL1),(TH2,TL2) เป็นรีจิสเตอร์ที่ใช้นับของ TIMER/COUNTER 0,1,2 ซึ่งมีขนาด 16 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>รีจิสเตอร์สำหรับใช้งานทั่วไป</u> ในไมโครคอนโทรลเลอร์ 8031 มีรีจิสเตอร์ทั่วไปที่ผู้เขียนโปรแกรมสามารถใช้งานได้ คือรีจิสเตอร์ A, B และกลุ่มของรีจิสเตอร์ 8 ตัวซึ่งมีชื่อเรียกตั้งแต่ R0-R7 รีจิสเตอร์ทั้ง 8 ตัวนี้มีอยู่ด้วยกันทั้งหมด 4 กลุ่ม ซึ่งในแต่ละกลุ่มก็มีชื่อเรียกเหมือนกัน คือ R0-R7 ดังนั้นในไมโครคอนโทรลเลอร์ 8031 จึงมีรีจิสเตอร์ใช้งานทั่วไป 34 ตัวในการทำงานขณะใดๆ รีจิสเตอร์ทั้ง 4 กลุ่มจะถูกเลือกใช้งานเพียงกลุ่มเดียวเท่านั้น ซึ่งการเลือกใช้งานรีจิสเตอร์ R0-R7 กลุ่มใดใน 4 กลุ่มจะกระทำได้โดยการเซ็ทหรือเคลียร์บิทในรีจิสเตอร์ PSW (บิท RS0,RS1) ดังต่อไปนี้

RS0	RS1	เลือกแบงค์	ค่าแอดเดรส
0	0	แบงค์ 0	00H-07H
0	1	แบงค์ 1	08H-0FH
1	0	แบงค์ 2	10H-17H
1	1	แบงค์ 3	18H-1FH

ในรีจิสเตอร์ทั้ง 4 กลุ่ม เนื่องจากใช้ชื่อเรียกเหมือนกัน (R0-R7) ดังนั้นเมื่อทำการเลือกรีจิสเตอร์กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่มขึ้นมาใช้งานแล้ว (โดยการเปลี่ยนค่าในบิท RS0,RS1) เวลาใช้งานรีจิสเตอร์ตัวใดตัวหนึ่งในกลุ่มของรีจิสเตอร์ที่ถูกเลือกนี้ ค่าที่เปลี่ยนแปลงไปในรีจิสเตอร์ที่ใช้งานจะไม่มีผลต่อรีจิสเตอร์ที่มีชื่อเดียวกันซึ่งอยู่ในรีจิสเตอร์กลุ่มอื่นเลย ทำให้มีความสะดวกในการเขียนโปรแกรมเป็นอันมาก โดยเฉพาะกับการเขียนโปรแกรมที่มีการเรียกใช้ Subroutine

<u>รีจิสเตอร์สำหรับใช้งานเฉพาะ</u> เนื่องจากไมโครคอนโทรลเลอร์ 8031 ออกแบบไว้ใช้งานเฉพาะทำให้มีความสามารถหลายอย่าง ซึ่งต้องอาศัยวงจรภายในที่มีเพิ่มมากขึ้นจากไมโครโปรเซสเซอร์ทั่วไป การควบคุมการทำงานของวงจรภายในไมโครคอนโทรลเลอร์นี้จะกระทำโดยอาศัยรีจิสเตอร์ซึ่งถูกกำหนดหน้าที่ไว้แล้ว โดยหากต้องการใช้งานไมโครคอนโทรลเลอร์ 8031 ให้มีประสิทธิภาพจำเป็นต้องรู้จักการใช้งานรีจิสเตอร์ใช้งานเฉพาะเหล่านี้ให้ดี รีจิสเตอร์ใช้งานเฉพาะหรือที่ใช้ชื่อเรียกย่อๆ ว่า SFR (SPECIAL FUNCTION REGISTER) จะอยู่ใน INTERNAL DATA MEMORY ส่วนที่เป็น SFR ดังแสดงในรูปที่ 2.6 ซึ่งบริเวณที่เป็น SFR นี้จะประกอบไปด้วยรีจิสเตอร์สำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ 8031 โดยเฉพาะ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลแบบ Boolean ไมโครคอนโทรลเลอร์ 8031 มีความสามารถพิเศษในการประมวลผลแบบ Boolean ซึ่งการประมวลผลแบบนี้มีไว้สำหรับงานทางด้านควบคุมโดยเฉพาะจึงทำให้ไมโครคอนโทรลเลอร์ 8031 เหมาะสมสำหรับที่จะนำไปใช้ควบคุมอุปกรณ์ต่างๆ

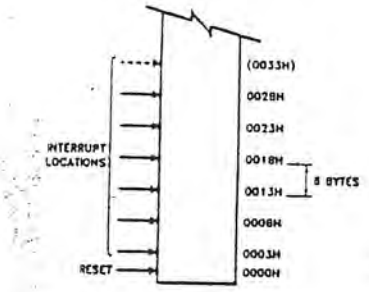
การประมวลผลแบบ Boolean จะเป็นการประมวลผลที่ใช้ข้อมูลขนาด 1 บิท โดยมีหน่วยความจำ ขนาด 1 บิทสำหรับประมวลผล ซึ่งผู้เขียนโปรแกรมสามารถอ้างแอดเดรสได้โดยตรง หน่วยความจำนี้จะอยู่ในบริเวณเดียวกับ INTERNAL RAM และ SFR บางตัว หน่วยความจำขนาด 1 บิท จะอยู่ในแต่ละบิทของ INTERNAL RAM ตำแหน่ง 20H-2FH และ SFR ที่มีค่าตำแหน่ง 3 บิทสุดท้ายเป็น 0 (ค่าแอดเดรสไบท์ต่ำเป็น 0H หรือ 8H)

ในการประมวลผลแบบ Boolean จะใช้ค่าข้อมูลจากหน่วยความจำขนาด 1 บิทเหล่านี้ โดยคำสั่ง ในการประมวลผลแบบนี้จำประกอบด้วย

- เซ็ทบิท
- เคลียร์บิท
- กระทำคำสั่งทางตรรกศาสตร์ระหว่างหน่วยความจำกับ CARRY FLAG
- ตรวจสอบสถานะบิทและข้ามไปทำงานในส่วนอื่นของโปรแกรม
- เคลื่อนย้ายข้อมูลระหว่าง CARRY FLAG กับหน่วยความจำขนาด 1 บิท

โครงสร้างการอินเทอร์รัพท์ ไมโครคอนโทรลเลอร์ 8031 สามารถรับสัญญาณอินเทอร์รัพท์ จะเป็นสัญญาณอินเทอร์รัพที่เกิดจากภายนอก 2 ชนิด และที่เกิดจากภายในชิพอีก 3 ชนิด เมื่อมีสัญญาณอินเทอร์รัพท์เกิดขึ้น ไมโครคอนโทรลเลอร์ 8031 จะละการทำงานโปรแกรมที่กำลังทำงานอยู่ จากนั้นจะข้ามไปทำงานโปรแกรมบริการอินเทอร์รัพท์ซึ่งอยู่ในหน่วยความจำตำแหน่งต่างๆ ขึ้นอยู่กับชนิดสัญญาณอินเทอร์รัพท์ ดังแสดงในรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงตำแหน่งของหน่วยความจำของโปรแกรมบริการอินเทอร์รัพท์

ในการอินเทอร์รัพท์ CPU สามารถทำได้โดยการกำหนดค่าในรีจิสเตอร์ IE และทำให้ควบคุมลำดับความสำคัญสัญญาณอินเทอร์รัพท์ได้ด้วย

<u>โครงสร้างพอร์ท</u> ไมโครคอนโทรลเลอร์ 8031 จะมีพอร์ทขนาด 8 บิท จำนวน 4 พอร์ท (P0-P3) โดยสามารถกำหนดให้ทำงานแบบพอร์ทขนานขนาด 8 บิท 4 พอร์ทหรือจะใช้พอร์ทขนาด 1 บิทได้ถึง 32 พอร์ท ทั้งนี้ผู้ใช้ยังสามารถกำหนดให้แต่ละพอร์ทใช้งานเป็นอินพุตหรือเอาท์พุตพอร์ทได้

ในกรณีที่ผู้ออกแบบต้องการใช้หน่วยความจำภายนอกไม่ว่าจะเป็น EXTERNAL DATA MEMORY หรือ EXTERNAL PROGRAM MEMORY พอร์ท 0 จะถูกกำหนดให้ใช้งานเป็น DATA BUS และ ADDRESS BUS ไบท์ต่ำ ส่วนพอร์ท 2 จะถูกกำหนดให้ใช้เป็น ADDRESS BUS ไบท์สูง และบางส่วนของพอร์ท 3 จะถูกใช้เป็น CONTROL BUS (READ/WRITE) แต่หากหน่วยความจำที่ใช้ภายนอกต้องการไม่ถึง 64 Kbyte พอร์ท 2 ที่ใช้เป็น ADDRESS BUS ไบท์สูง จะไม่ถูกนำมาใช้ทั้งหมด แต่พอร์ท 0 จะถูกใช้หมดทั้ง 8 เส้น เพราะต้องใช้เป็น DATA BUS ส่วนพอร์ท 3 จะนำมาใช้ติดต่อกับหน่วยความจำด้วยหรือไม่นั้นขึ้นอยู่กับหน่วยความจำภายนอกมี EXTERNAL DATA MEMORY ด้วยหรือไม่ (ต้องการสัญญาณ READ/WRITE) ดังนั้นในการออกแบบระบบ หากต้องการใช้หน่วยความจำภายนอกมากขึ้นเพียงใด ก็จะยิ่งทำให้เหลือจำนวนพอร์ทที่นำมาใช้งานลดลง ในการออกแบบจริงจึงต้องพยายามลดความต้องการหน่วยความจำภายนอกให้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ท 3 ซึ่งมีขนาด 8 บิท นอกจากจะใช้เป็นสัญญาณ READ/WRITE ในการติดต่อกับ EXTERNAL DATA MEMORY แล้ว ยังสามารถใช้เป็นสัญญาณอินเทอร์รัพท์ (INT0, INT1) สัญญาณอินพุท ของ COUNTER (T0,T1) รวมทั้งใช้ในการติดต่อสื่อสารแบบอนุกรมกับอุปกรณ์ภายนอก (RXD,TXD) อีกด้วย ดังนั้นเราจึงเหลือจำนวนบิทที่จะนำมาใช้งานเป็นพอร์ทอินพุทเอาท์พุตทั่วไปน้อยลง

ภายในแต่ละพอร์ทที่ใช้เป็นอินพุทหรือเอาต์พุท ผู้ใช้สามารถกำหนดให้ทำงานเป็นอินพุทหรือเอาท์พุตพอร์ทได้อย่างอิสระ โดยอาศัยการควบคุมจากโปรแกรม ซึ่งสามารถควบคุมให้แต่ละพอร์ทถูกใช้เป็นอินพุทในช่วงเวลาหนึ่ง และเอาท์พุตในอีกช่วงเวลาหนึ่งได้

TIMER/COUNTER ในไมโครคอนโทรลเลอร์ 8031 มีวงจรที่ทำหน้าที่นับสัญญาณนาฬิกา (T0, T1) กำเนิดความถี่และวงจรที่ทำหน้าที่นับจำนวนครั้งในการเปลี่ยนสถานะของสัญญาณ (COUNTER) ที่ ขา T0, T1 ของพอร์ท 3 โดยมีรีจิสเตอร์ที่ใช้ในการนับมีขนาด 16 บิทจำนวน 2 ตัว ดังนั้นไมโครคอนโทรลเลอร์ 8031 จึงสามารถวัดช่วงห่างของเวลา วัดความกว้างของพัลส์ นับจำนวนครั้งของเหตุการณ์ที่เกิดอยู่ในรูปของสัญญาณไฟฟ้าแล้ว หรือกำเนิดสัญญาณอินเทอร์รัพท์ที่มีคาบเวลาที่แน่นอนได้

รีจิสเตอร์ที่ใช้เป็น TIMER/COUNTER ในไมโครคอนโทรลเลอร์ 8031 มีอยู่ด้วยกัน 2 ตัว (T0, T1) ขนาด 16 บิท ซึ่งสามารถแยกใช้งานได้อย่างอิสระ T0 สามารถโปรแกรมให้ทำงานแตกต่างกัน ได้ถึง 4 อย่างด้วยกันดังนี้

- โหมด 0 ใช้เป็น TIMER หรือ COUNTER ขนาด 13 บิท
- โหมด 1 ใช้เป็น TIMER หรือ COUNTER ขนาด 16 บิท
- โหมด 2 ใช้เป็น TIMER หรือ COUNTER ขนาด 8 บิท ที่มีการโหลดค่าใหม่ทุกครั้ง หลังจากนับครบจำนวนที่ตั้งเอาไว้
- โหมด 3 ใช้เป็น TIMER หรือ COUNTER ขนาด 8 บิท

T1 จะมีการทำงานย่อย 3 โหมดแรกเหมือน T0 ส่วนการทำงานในโหมดที่ 3 จะมีลักษณะที่แตกต่างออกไปจากนี้ ดังจะกล่าวรายละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งในไมโครคอนโทรลเลอร์ 8031 คำสั่งที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ 8031 มีการอ้างแอดเดรสได้หลายวิธีดังนี้

- DIRECT ADDRESSING
- INDIRECT ADDRESSING
- REGISTER INSTRUCTIONS
- REGISTER-SPECIFIC INSTRUCTION
- IMMEDIATE CONSTANTS
- INDEXED ADDRESSING

DIRECT ADDRESSING : เป็นวิธีการเข้าถึงข้อมูลโดยการใช้ค่าตำแหน่งที่อยู่ของข้อมูลระบุในตัวคำสั่ง ข้อมูลที่สามารถอ้างได้โดยวิธีนี้จะเป็นข้อมูลใน INTERNAL DATA MEMORY บริเวณ LOWER 128 และ SFR เท่านั้น

INDIRECT ADDRESSING : เป็นวิธีการเข้าถึงข้อมูลโดยทางอ้อม นั่นคือแทนที่ผู้เขียนโปรแกรมจะระบุค่าตำแหน่งข้อมูลเหมือนแบบ DIRECT ADDRESSING วิธีนี้จะใช้ค่าที่อยู่ในรีจิสเตอร์เพื่อชี้ไปยังตำแหน่งของข้อมูลแทน รีจิสเตอร์ที่สามารถนำมาใช้เป็นตัวชี้ตำแหน่งของหน่วยความจำมีดังต่อไปนี้

- R0, R1 ของแต่ละ Register Bank ทั้ง 4
- SP (STACK POINTER)
- DPTR (DATA POINTER)

การใช้วิธีการเข้าถึงข้อมูลแบบ INDIRECT ADDRESSING นี้ Operand ซึ่งเป็นรีจิสเตอร์จะต้องระบุเครื่องหมาย "@" ไว้ข้างหน้า ดังตัวอย่าง

```
MOV A,@R0

MOV @DPTR,A
```

REGISTER INSTRUCTION : เป็นวิธีการเข้าถึงข้อมูลที่อยู่ในรีจิสเตอร์ R0-R7 ของแต่ละ REGISTER BANK โดยระบุข้อมูลเป็นรีจิสเตอร์แต่ละตัวเลย (R0-R7)

REGISTER-SPECIFIC : คำสั่งบางคำสั่งไมโครคอนโทรลเลอร์ 8031 จะระบุไว้แล้วว่าต้องอ้างข้อมูลในรีจิสเตอร์ตัวไหน เช่น A, DPTR, SP ดังนั้น Opcode ของ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งประเภทนี้ MCS-31 จะรู้ได้เองว่าต้องประมวลผลกับรีจิสเตอร์ตัวใด ด้วยเหตุนี้คำสั่งในกลุ่มนี้จึงไม่ต้องการ Operand ที่จะมาเป็นข้อมูลซ้ำอีก ตัวอย่างเช่น

```
MOV A,#data

MOV DPTR,#data
```

จากตัวอย่างนี้เราไม่จำเป็นต้องระบุตำแหน่งของรีจิสเตอร์ ,DPTR เลยเพราะไมโครคอนโทรลเลอร์ 8031 จะทราบได้เองจาก OPCODE ของคำสั่ง

IMMEDIATE CONSTANT : เป็นการกำหนดค่าให้กับ Operand โดยตรง ข้อมูลที่นำมาประมวลผลในคำสั่งจะอยู่ตามหลัง OPCODE ทั้งนี้จะต้องใช้เครื่องหมาย "#" ระบุหน้าข้อมูลที่ต้องการ เช่น

```
MOV A,#100
```

INDEXED ADDRESS : ข้อมูลที่ใช้วิธีการอ้างแบบนี้ จะเป็นข้อมูลที่อยู่ในหน่วยความจำส่วนที่เป็น PROGRAM MEMORY เท่านั้น จุดประสงค์ของการอ้างข้อมูลแบบนี้มีไว้เพื่อใช้ในการเปิดหาค่าข้อมูลที่เก็บไว้ในหน่วยความจำถาวร ซึ่งเป็น PROGRAM MEMORY ซึ่งไม่สูญหายแม้ไม่มีพลังงานจ่ายให้ โดยในการทำงานของคำสั่งที่ใช้การอ้างวิธีนี้จะใช้ค่าของรีจิสเตอร์ DPTR หรือ PC มารวมกับค่าในรีจิสเตอร์ A เพื่อชี้ไปยังตำแหน่งของหน่วยความจำบริเวณ PROGRAM MEMORY ซึ่งเก็บข้อมูลไว้ ดังนั้นค่าในรีจิสเตอร์ DPTR,PC จะต้องมีค่าเท่ากับตำแหน่งต้นของหน่วยความจำส่วนที่เก็บข้อมูลที่ต้องการ ส่วนค่าของรีจิสเตอร์ A จะเป็นตัวเลือกข้อมูลที่อยู่ในหน่วยความจำ เช่น

```
MOVC A,@A+DPTR

MOVC A,@A+PC
```

ความจริงการอ้างข้อมูลวิธีนี้ยังมีที่ใช้ในกลุ่มคำสั่งควบคุมโปรแกรม โดยการบังคับให้โปรแกรมกระโดดไปทำงานที่ตำแหน่งใดๆ ในหน่วยความจำซึ่งมีค่าตำแหน่งเท่ากับผลรวมของค่าในรีจิสเตอร์ DPTR และ A ดังตัวอย่าง

```
JMP @A+DPTR
```

ซึ่งสามารถนำไปใช้ประโยชน์ในกรณีของการเลือกการทำงานของการเลือกการทำงานของโปรแกรมเป็น Switch Case เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งทั้งหมดในไมโครคอนโทรลเลอร์ 8031 สามารถแยกประเภทตามลักษณะการทำงานได้ดังนี้

- กลุ่มคำสั่งทางคณิตศาสตร์ (ARITHMETIC INSTRUCTIONS) เป็นกลุ่มคำสั่งทางคณิตศาสตร์ทั้งหมด ซึ่งจะประกอบไปด้วยคำสั่งในการบวก ลบ คูณ หาร และมีวิธีการอ้างถึงข้อมูลที่จะมาดำเนินการได้หลายวิธี

ข้อมูลที่เห็นเป็นตัวอักษร จะหมายถึงข้อมูลขนาด 8 บิท ซึ่งสามารถใช้วิธีการเข้าถึงได้หลายวิธีหรือวิธีเดียวขึ้นกับชนิดของคำสั่ง

-กลุ่มคำสั่งทางตรรกศาสตร์ (LOGIC INSTRUCTIONS) ประกอบด้วยกลุ่มคำสั่งทางตรรกศาสตร์ ซึ่งจะมีคำสั่งในการ AND OR EXCLUSIVE-OR COMPLEMENT

กลุ่มคำสั่งในกลุ่มนี้ยังรวมไปถึงคำสั่งในการเลื่อนบิทของข้อมูลเป็นวงรอบ (ROTATE) ทั้งทางซ้ายและทางขวา โดยจะเลื่อนผ่านบิท C หรือไม่ก็ได้ นอกจากนี้ยังมีคำสั่งพิเศษในการสลับที่ข้อมูล 4 บิทบน และ 4 บิทล่าง (SWAP)

- กลุ่มคำสั่งเคลื่อนย้ายข้อมูล (DATA TRANSFER INSTRUCTIONS) เป็นคำสั่งที่ใช้เคลื่อนย้ายข้อมูล เพื่อใช้ในการนำข้อมูลที่ได้จากการประมวลผลไปเก็บยังหน่วยความจำบริเวณใดบริเวณหนึ่ง หรือเพื่อเคลื่อนย้ายข้อมูลไปยังรีจิสเตอร์เพื่อประมวลผล ทั้งนี้เนื่องจากคำสั่งบางคำสั่ง จำเป็นต้องปฏิบัติงานที่รีจิสเตอร์เฉพาะตัวเท่านั้น เช่น คำสั่งในการคูณ

- กลุ่มคำสั่งในการควบคุมโปรแกรม PROGRAM CONTROL INSTRUCTIONS) เป็นคำสั่งที่ใช้ในการควบคุมลำดับการทำงานของโปรแกรม โดยทำให้โปรแกรมสามารถตัดสินใจการทำงานได้จากเงื่อนไขต่างๆ และสามารถย้างการทำงานไปยังที่อื่นได้ เมื่อเงื่อนไขที่ตั้งไว้เป็นจริง

-กลุ่มคำสั่งการประมวลผลแบบ Boolean (Boolean INSTRUCTIONS) ไมโครคอนโทรลเลอร์ 8031 มีความสามารถพิเศษในการประมวลผลแบบบูลีนซึ่งไม่มีในไมโครโปรเซสเซอร์ชนิดอื่น

serial interface

ไมโครคอนโทรลเลอร์ 8031 มี Serial port ซึ่งสามารถรับและส่งข้อมูลแบบอนุกรมได้ โดยผู้ใช้ไม่ต้องใช้ชิพอื่นเพิ่มเลยทำให้มีความสะดวกในการในไปประยุกต์ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ต้องมีการติดต่อข้อมูลแบบอนุกรมมาก

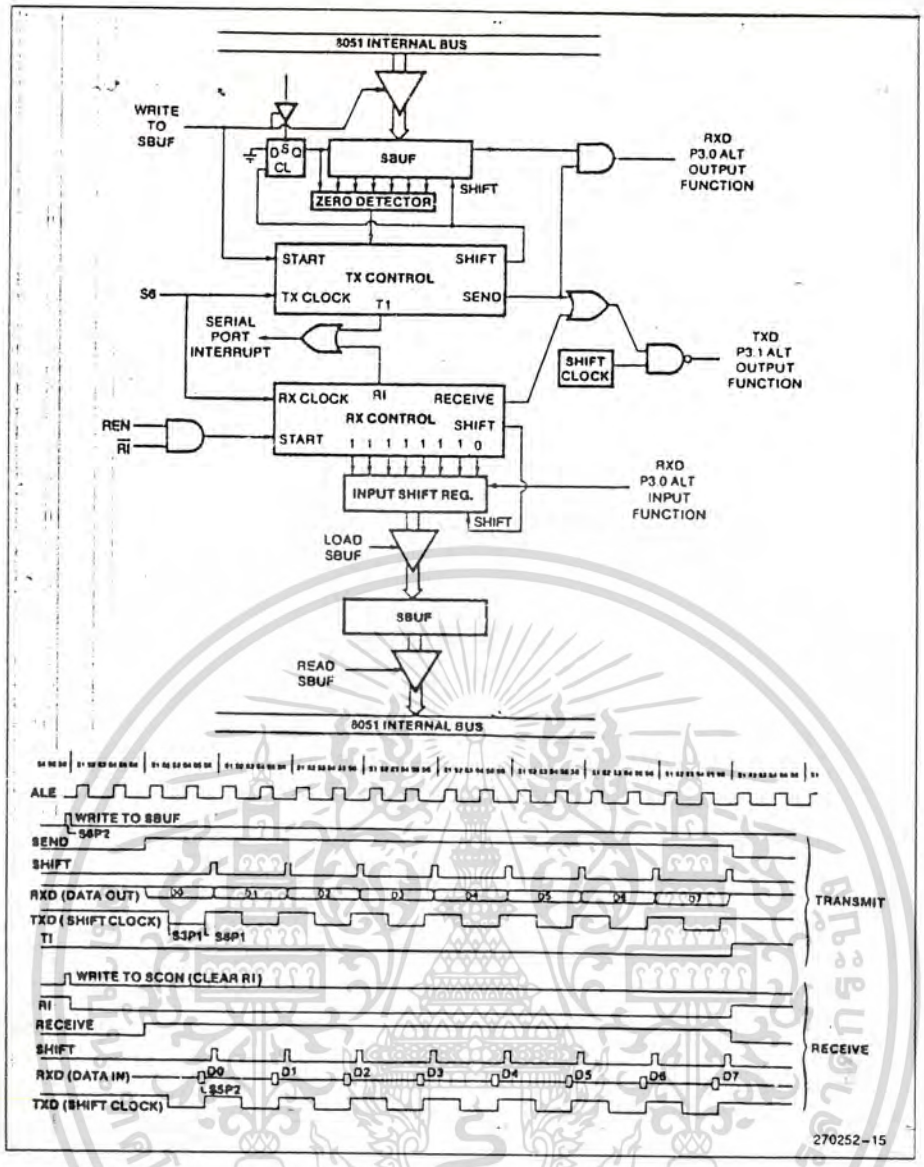
SERIAL PORT ที่มีในไมโครคอนโทรลเลอร์ 8031 สามารถทำงานได้ในแบบ Full Duplex คือ สามารถรับและส่งข้อมูลได้พร้อมๆ กัน โดยในการรับข้อมูลจะมีบัฟเฟอร์ให้ด้วยทำให้ด้วย ทำให้ 8031 สามารถรับข้อมูลไบท์ที่ 2 ซึ่งถูกส่งตามเข้ามาหลังไบท์แรกได้ บัฟเฟอร์ที่มีใน 8031 นี้จะสามารถรับข้อมูลไบท์ที่สองก่อนที่ไบท์แรกซึ่งรับเข้ามาก่อนจะถูกอ่านจาก Recievever Register ไปเก็บไว้ในหน่วยความจำ (แต่ถ้าไบต์แรกยังไม่ถูกอ่านเมื่อเวลาที่การรับจองไบต์ที่สองสิ้นสุดลง หนึ่งไบต์ในสองไบต์จะสูญหายไป

Serial Port ประกอบด้วยรีจิสเตอร์ขนาด 8 บิท จำนวน 2 ตัว แต่ละตัวมีชื่อเรียกตามหน่าที่ดังนี้คือ Reciever Register สำหรับการรับข้อมูล และ Transmit Register สำหรับการส่งข้อมูล รีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันใน SFR คือตรงกับตำแหน่ง SBUF(99H) โดยการเข้าถึงข้อมูลของรีจิสเตอร์แต่ละตัว 8031 จะทราบเองว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใดโดยดูจากคำสั่ง ทั้งนี้เพราะในการเขียนข้อมูลไปที่รีจิสเตอร์ SBUF จะหมายถึงการโหลดค่าไปยัง Transmit Register เพื่อส่งข้อมูลออกไปภายนอก ส่วนการอ่านข้อมูล Receive Register หมายถึงการรับข้อมูลจากภายนอกแบบอนุกรม

การใช้งาน Serial Port ใน 8031 มีความสะดวกและคล่องตัวสูง ทั้งนี้เนื่องจากผู้ใช้สามารถกำหนดการทำงานแตกต่างกันได้ถึง 4 ประเภท โดยการเลือกใช้งาน Serial Port ประเภทใดจะขึ้นอยู่กับค่าของบิทในรีจิสเตอร์ SCON การใช้งานที่มีแตกต่างกัน 4 ประเภทนี้มีจุดประสงค์เพื่อความเหมาะสมกับการรับส่งข้อมูลแบบอนุกรมแต่ละชนิดดังนี้

<u>MODE 0</u> : ขา RXD มีไว้สำหรับการรับและส่งข้อมูลแบบอนุกรมทั้ง 2 กรณี ส่วนขา TXD มีไว้เพื่อใช้กำเนิดสัญญาณ Shift Clock เพื่อเป็นตัวกำหนดจังหวะในการรับหรือส่งข้อมูล (ข้อมูลจะถูกรับหรือส่งตามสัญญาณ Shift Clock) ในโหมดนี้การรับส่งข้อมูลจะเป็นแบบ 8 บิท (DATA Bits) โดยรับและส่งบิทต่ำสุดก่อน (LSB) อัตราการรับส่งข้อมูล (BAUD RATE) ถูกกำหนดไว้ที่ 1/12 ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบการทำงานในโหมดนี้จะไม่มี start bit และ stop bit เพราะข้อมูลที่รับหรือส่งออกไปถูกกำหนดโดยใช้สัญญาณจาก Shift CLock

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Serial Port Mode 0

รูปที่ 2.8 แสดงถึงโครงสร้างในการทำงานคร่าวๆ ของ Seraial Port ในโหมด 0 รวมทั้ง Timing Diagram ที่เกี่ยวข้องสำหรับการส่งและการรับข้อมูล

จากรูป 2.8 การส่งข้อมูลถูกทำให้เริ่มต้นโดยคำสั่งใดๆ ที่ใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง (destination Register) สัญญาณ Write to SBUF (ในภาพ) ที่เกิดขึ้นขณะ S6P2 จะเป็นการโหลดค่า 1 ไปที่ตำแหน่งที่ 9 ของ transmit shift register (จากภาพ บิทที่ 9 คือ D-FLIP FLOP) และบอกให้วงจรควบคุมการส่งข้อมูล (TX Contrl Unit) เริ่มต้นส่งข้อมูล แผนผังเวลาภายในที่แสดงรายละเอียดการส่งข้อมูลจะเห็นได้ว่ามี 1 machine cycle เต็มอยู่ระหว่างช่วงสัญญาณ write to SBUF ที่ S6P2 และสัญญาณ SEND ที่ถูกกระตุ้นให้เริ่มทำงาน (ขณะเริ่ม active)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ SEND ที่ถูก enable (มีค่าเป็น 1) จะเป็นสัญญาณออกจากวงจรควบคุมการส่งข้อมูลไปควบคุมให้ shift register ส่งข้อมูลออกไปทีละบิทข้อมูลที่ส่งออกมาจะผ่านออกไปยังขา RXD ซึ่งตรงกับ P3.0 และสัญญาณ SEND นี้ยังไปควบคุมให้วงจร Shift Clock ซึ่งทำหน้าที่สร้าง Shift clock ผ่านออกไปทางขา TXD ซึ่งตรงกับ P3.1 โดย Shift Clock จะเป็นสัญญาณที่มีค่าเป็นๆ low ระหว่าง S3, S4 ,S5 ของทุก machine และมีค่า high ระหว่าง S6, S1, S2 ของทุก machine cycle ดังแสดงให้เห็นในภาพ เมื่อ S6P2 ของทุกๆ machine cycle ซึ่งสัญญาณ SEND ยังคงถูก enable ข้อมูลของ transmit shift register จะถูก shift ไปทางขวา 1 ตำแหน่ง ทำให้ข้อมูลที่ได้ผ่านออกมาทางขา RXD ทีละ 1 บิท โดยเริ่มจากบิทต่ำสุดก่อน

ขณะที่ข้อมูลซึ่งถูก shift ออกมาทางขวาทีละบิทนี้ ค่า 0 จะถูกนำมาแทนที่ตำแหน่งซึ่งถูก shift ออกไป โดยเข้ามาทางด้านซ้ายของ Transmit Shift Register (มาจาก D-Flip Flop) การส่งข้อมูลจะดำเนินไปเรื่อยๆ จนกระทั่งข้อมูลซึ่งเป็นบิทสูงสุด (MSB) ของ data byte ถูกเลื่อนไปอยู่ที่ตำแหน่งเอาท์พุท (Output Position) ของ Shift Register และค่า 1 ที่เคยถูกโหลดไปที่ตำแหน่งที่ 9 ที่กล่าวไว้ในตอนแรกจะอยู่ถัดจากบิทสูงสุดมาทางซ้าย โดยทุกตำแหน่งทางซ้ายของมันจะมีค่าเป็น 0 หมด (มีศูนย์มา ทางซ้ายรวม 7 ตัว) ในสภาวะเช่นนี้จะส่งผลให้วงจรตรวจจับศูนย์ (ZERO Detector) ซึ่งทำงานโดยตรวจจับเพียงแค่ 7 บิท เริ่มทำงานโดยการส่งสัญญาณไปบอกวงจรควบคุม การส่งข้อมูลทำให้การ shift ข้อมูลครั้งสุดท้ายอีก 1 ครั้ง เพื่อส่งบิทสูงสุดออกไป แต่บิทที่ 9 จะไม่ถูกส่งตามออกมาโดยจะยังคงอยู่ที่ตำแหน่งเอาท์พุทของ shift register เท่านั้น จากนั้นการส่งข้อมูลจะสิ้นสุดลง โดยวงจรส่วนควบคุมการส่งข้อมูลจะทำให้สัญญาณ SEND หยุดทำงาน (Deactive SEND) เมื่อข้อมูลแต่ละบิทในรีจิสเตอร์ SBUF ถูกส่งออกไปภายนอกเรียบร้อยแล้ว บิท TI จะถูกเซทเพื่ออินเทอร์รัพท์ซีพียู การกระทำทั้งสองนี้จะเกิดขึ้นที่ขณะ S1P1 ของ machine cycle ที่ 10 หลังจากที่มีสัญญาณ write to SBUF

การรับข้อมูลในโหมด 0 เกิดขึ้นได้ก็ต่อเมื่อบิท REN = 1 และ RI = 0 (พิจารณาจากรูปที่12) โดยขณะ S6P2 ของ machine cycle ถัดไป วงจรควบคุมการรับข้อมูล (RX control block) จะทำการเขียนข้อมูล 11111110B ไปที่ Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

register และในสัญญาณนาฬิกาเฟสถัดไป สัญญาณ RECEIVE จากวงจรควบคุมการรับข้อมูลจะเริ่มต้นทำงาน (Active)

สัญญาณ RECEIVE ที่ได้มาจากวงจรควบคุมการรับข้อมูลจะทำการ enable shift clock ไปที่ขา TXD ซึ่งมีการเปลี่ยนสถานะทุกๆ S3P1 และ S6P1 ของทุกๆ machine cycle ในช่วงการรับข้อมูล โดยขณะ S6P2 ของแต่ละ machine cycle ซึ่งสัญญาณ RECEIVE ยังคงทำงานอยู่ข้อมูลของ RECEIVER shift register จะถูก shift มาทางซ้าย 1 ตำแหน่ง เนื่องจากมีข้อมูลที่รับได้จากภายนอกถูก shift จากทางขวาเข้ามาในรีจิสเตอร์ ค่าของข้อมูลที่เข้ามาจากทางขวาจะเป็นข้อมูลซึ่งได้รับการตรวจสอบ (sampled) ที่ขา RXD ในช่วง S5P2 ของ machine cycle เดียวกัน

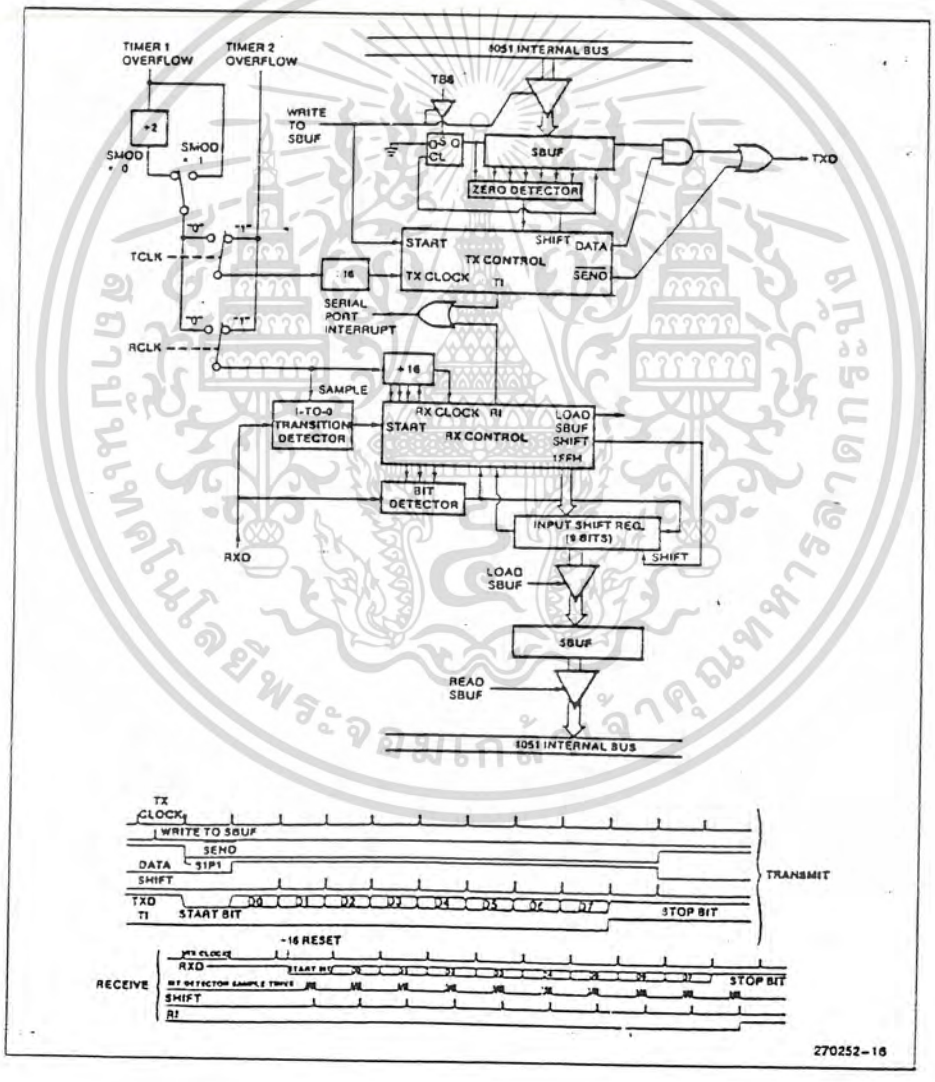
ขณะที่ข้อมูลถูกเลื่อนเข้ามาจากทางขวาทีละบิท ค่าของ 1 ซึ่งถูกนำไปไว้ใน RECEIVER shift register ในตอนแรกจะถูกเลื่อนออกไปทางซ้าย (shift out) เมื่อ 0 ซึ่งถูกโหลดไปไว้ที่บิทต่ำสุดตั้งแต่ตอนแรก (1111110) ถูกเลื่อนมาอยู่ในตำแหน่งซ้ายสุดของ shift register จะทำให้เกิดสัญญาณส่งไปบอกวงจรส่วนควบคุมการรับข้อมูลให้ทราบว่าขณะนี้ข้อมูลถูกรับเข้า 7 บิทแล้ว เมื่อวงจรควบคุมการรับข้อมูลได้รับสัญญาณนี้ก็จะทำการรับข้อมูลอีก 1 บิทโดยการ shift มาทางซ้ายอีก 1 ครั้ง และส่งสัญญาณ LOAD SBUF เพื่อให้ข้อมูลเข้าไปอยู่ในรีจิสเตอร์ SBUF การกระทำทั้งหมดนี้เกิดขึ้นในช่วง S1P1 ของ machine cycle ที่ 10 หลังจากที่มีการ write ไปที่รีจิสเตอร์ SCON ซึ่งไปเคลียร์บิท RI (ดูรายละเอียดจากภาพ) เมื่อการรับข้อมูลสิ้นสุดลงแล้ว สัญญาณ RECEIVE จะหยุดทำงานและบิท RI จะถูกเซท

ในการส่งและการรับข้อมูล serial port ในโหมด 0 นี้จะเห็นว่าจังหวะในการรับหรือส่งขึ้นกับ machine cycle ของซีพียู นั่นคือ ข้อมูลจะถูกรับเข้ามาหรือส่งออกไปทุกๆ machine cycle จึงทำให้ Baud Rate ของการทำงานโหมด 0 มีค่า 1/2 ของความถี่ออสซิลเลเตอร์ (เท่ากับความถี่ของ machine cycle)

สัญญาณ shift Clock ที่เกิดขึ้นขณะส่งข้อมูลจะเป็นสัญญาณเพื่อควบคุมจังหวะในการรับข้อมูลของวงจรภายนอกที่รับข้อมูลจาก 8031 ส่วนสัญญาณ Shift Clock ที่เกิดขึ้นขณะ 8031 รับข้อมูลจากภายนอกจะถูกสร้างขึ้น เพื่อเป็นสัญญาณบอกให้วงจรภายนอกส่งข้อมูลให้แก่ 8031

เอกสารนี้มีลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 1 : จะมีการรับและส่งข้อมูลครั้งละ 10 บิท โดยการส่งข้อมูลจะส่งผ่านทางขา TXD ส่วนการรับข้อมูลจะรับทางขา RXD ข้อมูลทั้ง 10 บิท จะประกอบไปด้วย 1 start bit (บิทที่ 0 มีค่าเป็น 0) 8 data bits (รับและส่งบิทต่ำก่อน) และ 1 stop bit (มีค่าเป็น 1) โดยขณะทำการรับข้อมูล ค่าของ stop bit ที่รับได้จะไปอยู่ในบิท RB8 ของรีจิสเตอร์ SCON อัตราเร็วในการรับหรือส่งข้อมูล (BAUD RATE) ของการรับและส่งข้อมูลในโหมดนี้สามารถเปลี่ยนแปลงได้



รูป 2.9 Serial Port Mode 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud rate ถูกกำหนดโดยอัตราการเกิด Overflow ของ Timer 1 การส่งข้อมูลจะเริ่มต้นด้วยคำสั่งใดๆ ที่ใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง (destination Register) สัญญาณ Write to SBUF จะเป็นการโหลดค่า 1 ไปที่ตำแหน่งที่ 9 ของ transmit shift register (บิทที่ 9 คือ D-FLIP FLOP ซึ่งมีอินพุทต่อลงกราวด์ดังแสดงในภาพ) และจะไปกระตุ้นให้วงจรควบคุมการส่งข้อมูลทราบว่าขณะนี้กำลังมีความต้องการที่จะส่งข้อมูล การส่งข้อมูลจริงๆ จะเกิดในช่วง S1P1 ของ machine cycle ที่ถัดจากการเกิด Rollover ครั้งถัดไปใน Counter ที่ถูกหารด้วย 16 (Counter จะถูกเพิ่มค่าเมื่อนับสัญญาณที่เป็นพัลส์ได้ครบ 16 ลูก) ดังนั้นจังหวะการส่งข้อมูลแต่ละบิทจะซิงโครไนซ์กับค่าของ Counter ที่ถูกหารด้วย 16 ไม่ใช่กับสัญญาณ write to SBUF

การส่งเริ่มต้นด้วยสัญญาณ SEND ที่ถูกกระตุ้นให้เริ่มทำงาน (Active) และทำการใส่ start bit (0) ไปยังขา TXD เมื่อเวลาของการส่งบิทแรกผ่านไป ข้อมูลจะถูกส่งออกมาทีละบิทจาก Transmit shift register โดยการ enable output bit ของ transmit shift register ไปที่ขา TXD สัญญาณ shift จะให้ pulse ลูกที่ 1 ออกมาภายหลัง การส่งข้อมูลผ่านไป 1 บิท

ขณะที่ data bit shift out ไปทางขวา ค่า 0 จะถูก clock เข้ามาทางซ้าย เมื่อ MSB ของ data bit ไปอยู่ที่ตำแหน่ง output ของ shift register แล้ว 1 ซึ่งถูกโหลดไปไว้ในตำแหน่งที่ 9 ในตอนแรกจะอยู่ถัดจาก MSB ไปทางซ้าย ส่วนตำแหน่งอื่นๆ ทางซ้ายทั้งหมดมีค่าเป็น 0 ซึ่งในสภาวะนี้ วงจรตรวจจับ 0 (ขนาด 7 บิท) จะทำงานโดยการส่งสัญญาณไปบอกวงจรควบคุมการส่งข้อมูลให้ทำการ shift ครั้งสุดท้ายอีก 1ครั้ง แล้วจึงหยุดการส่งข้อมูล (Deactive SEND) พร้อมทั้งเซทบิท TI เพื่ออินเทอร์รัพซีพียู เหตุการณ์ที่เกิดขณะการส่งข้อมูลเสร็จสิ้นลงจะเกิดขึ้นที่ขณะเกิด Rollover ครั้งที่ 10 ของ counter ที่ถูกหาร 16 หลังจากสัญญาณ write to SBUF เกิดขึ้น

การรับข้อมูลเริ่มต้นเมื่อตรวจพบการเปลี่ยนสถานะจาก 1 เป็น 0 (ตรวจพบ start bit) ที่ขา RXD ซึ่งการตรวจจับสถานะนี้จะถูก sampled ด้วยอัตรา 16 ครั้ง (อัตราเดียวกับการเกิด rollover ซึ่งถูกส่งต่อไปให้ counter หาร 16 ต่อไป) ไม่ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate จะถูกกำหนดเป็นเท่าใดก็ตามเมื่อการเปลี่ยนสถานะของสัญญาณถูกตรวจพบ Counter หาร 16 จะถูกรีเซทในทันที และข้อมูล 1FFH จะถูกนำไปไว้ใน input shift register การรีเซท counter หาร 16 ก็เพื่อตั้งการ rollover ให้เริ่มตรงกับการเริ่มต้นของบิทที่ได้รับเข้ามา

จากรูป counter จะถูกนับขึ้นไป 1 ก็ต่อเมื่อมีการเกิด Overflow ขึ้น 16 ครั้ง หรือกล่าวง่ายๆ ได้ว่าเมื่อเกิด overflow 16 ครั้ง ค่าของ counter จะถูกเพิ่มขึ้น 1 เสมอซึ่งเวลาในการรับข้อมูลแต่ละบิทจะนำเอาอัตราการเกิด Overflow มาเป็นตัวตรวจสอบ นั่นคือ แบ่งเวลาในการรับข้อมูลแต่ละบิทออกเป็น 16 State โดยมีตัวตรวจสอบข้อมูลแต่ละบิท (BIT Detector) ทุกๆ state ที่ 7,8,9 ของเวลาในข้อมูลแต่ละบิท ตัวตรวจสอบจะตรวจสอบค่าสถานะที่ขา RXD ซึ่งมีข้อมูลรออยู่ ค่าที่รับเข้ามาจะถือว่าเป็นบิทข้อมูลที่ถูกต้องก็เมื่อการตรวจสอบพบว่าเป็นค่าเดียวกันอย่างน้อย 2 ใน 3 ของ state 7,8,9 ที่ต้องทำเช่นนี้ เพื่อเป็นการป้องกันสัญญาณรบกวน (noise rejection) แต่ในช่วงของการรับบิทแรก (start bit) ถ้าค่าที่ถูกตรวจสอบได้ใน state 7,8,9 ไม่เป็น 0 วงจรที่ทำหน้าที่รับข้อมูลจะถูกรีเซท และระบบจะเริ่มกลับไปตรวจหาการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD ต่อไปที่ต้องทำเช่นนี้ก็เพื่อยกเลิกการรับข้อมูลหาก start bit ไม่ถูกต้องแต่ถ้า start bit ถูกตรวจสอบแล้วปรากฏว่ามีค่าเป็น 1 การรับข้อมูลบิทต่อไปจะเริ่มต้นทันทีโดย start bit จะถูก shift ไปไว้ใน input shift register และบิทข้อมูลที่เหลือจะถูกรับตามมา

ขณะที่ data bit เข้ามาทางขวา 1 (1FFFH ที่ถูกโหลดไว้ก่อน) จะถูก shift out ออกไปทางซ้ายจนกระทั่งเมื่อ start bit ถูกเลื่อนไปถึงตำแหน่งซ้ายสุดใน shift register (ในโหมด 1 รีจิสเตอร์ทั้งหมดจะมีขนาด 9 บิท) มันจะส่งสัญญาณไปบอก RX control block ให้ทำการ shift ครั้งสุดท้ายอีก 1 ครั้ง เพื่อรับ stop bit ไปไว้ในบิท RB8 ในรีจิสเตอร์ SCON จากนั้นจึงส่งสัญญาณไปโหลดข้อมูลจาก input shift register เพื่อนำไปเก็บไว้ใน รีจิสเตอร์ SBUF และ RX control block จะส่งสัญญาณ RI (RI ถูกรีเซท) เพื่ออินเทอร์รัพท์ซีพียู แต่สัญญาณที่จะโหลดข้อมูลไปไว้ในรีจิสเตอร์ SBUF และรับ stop bit ไปไว้ในบิท RB8 รวมทั้งการเซ็ทบิท RI จะถูกสร้างขึ้นก็ต่อเมื่อเงื่อนไขต่อไปนี้เป็นจริงขณะสัญญาณ shift pulse ลูกสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกสร้างขึ้น

- บิท RI ถูกเคลียร์ (RI = 0) และ

- บิท SM2 = 0 และ stop bit = 1

ถ้าเงื่อนไขอย่างใดอย่างหนึ่งทั้งสองอย่างนี้ไม่ตรงกัน กลุ่มข้อมูลที่รับเข้ามาจะหายไปโดยไม่สามารถเรียกคืนได้ ถ้าเงื่อนไขทั้งสองถูกต้อง stop bit จะถูกนำไปไว้ใน บิท RB8 ในรีจิสเตอร์ SCON ส่วนบิทที่เป็นข้อมูลจำนวน 8 บิท (data bit) จะถูกโหลดไปไว้ในรีจิสเตอร์ SBUF และบิท RI จะถูกเซทเมื่อถึงตอนนี้ไม่ว่าเงื่อนไขข้างต้นจะตรงหรือไม่ ระบบการรับส่งข้อมูลจะถือว่าการรับข้อมูลจำนวน 1 ไบท์ (รับข้อมูลเข้ามา 10 บิท แต่เป็น data จริงๆ เพียง 8 บิท) เสร็จสิ้นลงแล้ว และเริ่มรีเซ็ทตัวเองเพื่อกลับไปรอรับการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ ขา RXD เพื่อรับข้อมูลไบท์ต่อไป

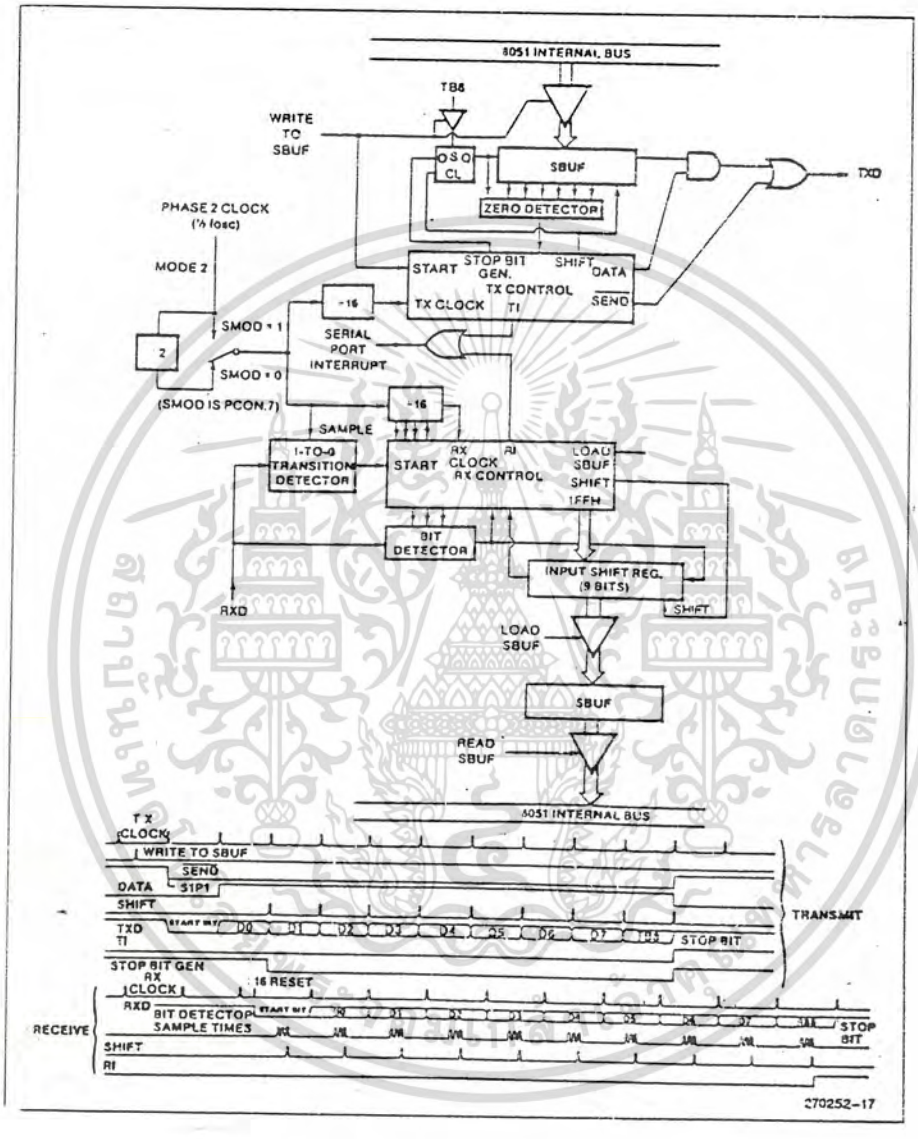
MODE 2 : จะมีการรับและส่งข้อมูลครั้งละ 11 บิท โดยข้อมูลถูกส่งผ่านทางขา TXD และ รับเข้ามาผ่านทางขา RXD ข้อมูลที่รับและส่งทั้ง 11 บิท ประกอบด้วย 1 start bit (บิทที่ 0 ที่ค่าเป็น 0) 8 databits (บิทที่ 1-8) โดยรับหรือส่งบิทต่ำก่อน ส่วนบิทที่ 9 เป็นบิทที่สามารถโปรแกรมให้มีค่าเป็น 0 หรือ 1 ก็ได้ (Programmable 9th Data bit) และบิทสุดท้ายคือ stop bit (มีค่าเป็น 1)

ในขณะทำการส่งข้อมูล ข้อมูลบิทที่ 9 จะเป็นค่าของ TB8 ในรีจิสเตอร์ SCON ดังแสดงในรูปที่ 14 บิทนี้สามารถถูกกำหนดให้มีค่าเป็น 0 หรือ 1 ก็ได้ ส่วนใหญ่ในการใช้งานจริง มักจะใช้บิทนี้เป็นบิทสำหรับการตรวจสอบข้อมูลที่รับหรือส่ง (parity bit) ว่าข้อมูลที่รับส่งถูกต้องหรือไม่ โดยจะนำเอาบิท P (parity bit) ในรีจิสเตอร PSW ไปไว้ในบิท TB8 ส่วนในขณะรับข้อมูล ข้อมูลบิทที่ 9 จะอยู่ในบิท RB8 ของรีจิสเตอร์ SCON โดยไม่สนใจ Stop bit ค่า Baud Rate ในโหมดนี้สามารถตั้งให้เป็น 1/32 หรือ1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบ

MODE 3 : ส่งข้อมูลครั้งละ 11 บิท โดยข้อมูลถูกส่งผ่านทางขา TXD และรับเข้ามาผ่านทางขา RXD ข้อมูลที่รับและส่งทั้ง 11 บิท ประกอบด้วย 1 start bit (บิทที่ 0 ที่ค่าเป็น 0) 8 data bits (บิทที่ 1-8) ส่วนบิทที่ 9 เป็นบิทที่สามารถ โปรแกรมให้มีค่าเป็น 0 หรือ 1 ก็ได้ (Programmable 9th Data bit) โดยในการส่ง บิทที่ 9 (บิท TB8 ในรีจิสเตอร์ SCON) ส่วนในการรับบิทที่ 9 จะไปปรากฏอยู่ที่บิท RB8 ในรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCON เช่นเดียวกับการส่งและบิทสุดท้ายคือ stop bit (มีค่าเป็น 1) อัตราเร็วในการรับส่งข้อมูลสามารถเปลี่ยนแปลงได้ จะเห็นว่า โหมด 3 ทำงานเหมือนโหมด 2 ทุกอย่างยกเว้น Baud Rate สามารถเปลี่ยนแปลงได้เท่านั้น



รูปที่ 2.10 Serial port mod 2,3

การส่งข้อมูลจะเริ่มต้นด้วยคำสั่งใดๆ ที่ใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง (destination Register) สัญญาณ Write to SBUF จะเป็นการโหลดค่า TB8 1 ซึ่งมีค่า เป็น 0 หรือ 1 ตามการกำหนดล่วงหน้าของผู้ใช้ไปที่ตำแหน่งที่ 9 ของ transmit shift register และจะไปกระตุ้นให้วงจรควบคุมการส่งข้อมูลทราบว่าขณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้กำลังมีความต้องการที่จะส่งข้อมูล การส่งข้อมูลจะเกิดในช่วง S1P1 ของ machine cycle ที่ถัดจากการเกิด Rollover ครั้งถัดไปใน Counter ที่ถูกหารด้วย 16 ครั้งถัดไป (ดังนั้นจังหวะการส่งข้อมูล แต่ละบิทจะซิงโครไนซ์กับค่าของ Counter ที่ถูกหารด้วย 16 ไม่ใช่กับสัญญาณ write to SBUF)

การส่งเริ่มต้นด้วยสัญญาณ SEND ที่ถูกกระตุ้นให้เริ่มทำงาน (Active) และทำการใส่ start bit (0) ไปยังขา TXD เมื่อเวลาของการส่งบิทแรกผ่านไป สัญญาณ data จะถูกกระตุ้นให้ทำงานตามมา ซึ่งจะเป็นการ enable output bit ของ transmit shift register ไปที่ขา TXD สัญญาณ shift จะให้ pulse ลูกที่ 1 ออกมาภายหลังการส่งข้อมูลผ่านไป 1 บิท (1 bit time) สัญญาณ shift นี้จะไปทำให้ 1 (ต่อไปจะกลายเป็น stop bit) ถูกใส่เข้าไปไว้ในตำแหน่งที่ 9 ของ shift register หลังจากนั้นจะมีเพียง 0 ที่ถูกใส่ไปใน shift register ดังนั้นขณะที่ data bit ถูก shift out ออกไปทางขวา 0 ก็จะถูกใส่เข้าไปทางซ้าย การส่งจะดำเนินไปเรื่อยๆ จนกระทั่ง TB8 ซึ่งถูกใส่ไว้ในตำแหน่งที่ 9 ของ shift register ในตอนแรกถูกเลื่อนไปอยู่ตำแหน่ง output ของ shift register และทางด้านซ้ายของบิท TB8 นี้จะมีค่า 1 (stop bit) จากการโหลดด้วย shift pluse ลูกแรก ส่วนตำแหน่งอื่นที่เหลือทางด้านซ้ายจะมีค่าเป็น 0 หมด สภาวะที่มี 0 อยู่ทางด้านซ้ายของ stop bit นี้จะถูกตรวจพบได้ด้วยวงจรตรวจจับศูนย์ (ZERO DETECTOR) ซึ่งเมื่อตรวจจับได้ก็จะส่งสัญญาณไปบอก TX control block ให้ทำการ shift ครั้งสุดท้ายอีก 1 ครั้ง แล้วจึงค่อยให้ สัญญาณ SEND หยุดทำงาน (Deactive SEND) พร้อมทั้งเซ็ทบิท TI เพื่ออินเทอร์รัพซีพียู เหตุการณ์ที่เกิดขึ้นหลังการตรวจจับ 0 ได้ 7 บิทนั้นจะเกิดขณะที่มีการ Rollover ครั้งที่ 11 ของ counter ที่ถูกหาร 16 หลังจาก สัญญาณ write to SBUF เกิดขึ้น

การรับข้อมูลเริ่มต้นเมื่อตรวจพบการเปลี่ยนสถานะจาก 1 เป็น 0 (ตรวจพบ start bit) ที่ขา RXD ซึ่งการตรวจจับสภานะนี้จะถูก sampled ด้วยอัตรา 16 ครั้ง (อัตราเดียวกับการเกิด rollover ซึ่งถูกส่งต่อไปให้ counter หาร 16 ต่อไป) ไม่ว่า Baud Rate จะถูกกำหนดเป็นเท่าใดก็ตาม เหมือนในโหมด 1 เมื่อการเปลี่ยนสถานะของสัญญาณถูกตรวจพบ Counter หาร 16 จะถูกรีเซ็ทในทันที และข้อมูล 1FFH จะถูกนำไปไว้ใน input shift register ที่ 7,8,9 ของเวลาในข้อมูลแต่ละบิท ตัวตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สอบจะตรวจสอบค่าสถานะที่ขา RXD ซึ่งมีข้อมูลรออยู่ ค่าที่รับเข้ามาจะถือว่าเป็นบิทข้อมูลที่ถูกต้องก็เมื่อการตรวจสอบพบว่าเป็นค่าเดียวกัน อย่างน้อย 2 ใน 3 ของ state 7,8,9 ที่ต้องทำเช่นนี้เพื่อเป็นการป้องกันสัญญาณรบกวน แต่ในช่วงของการรับบิทแรก (start bit) ถ้าค่าที่ถูกตรวจสอบได้ใน state 7,8,9 ไม่เป็น 0 วงจรที่ทำหน้าที่รับข้อมูลจะถูกรีเซทและระบบจะเริ่มกลับไปตรวจหาการเปลี่ยนสถานะที่ขา RXD ตามเดิม ที่ต้องทำเช่นนี้ก็เพื่อให้แน่ใจว่าบิทแรกที่เข้ามาเป็น start bit แน่นอนนั่นเอง หากบิทแรกตรวจพบว่ามีค่าเป็น 1 นั่นคือ start bit แน่นอน การรับข้อมูลบิทต่อไปจะเริ่มต้นทันที โดยจะทำการ shift start bit ไปไว้ที่ input shift register

ขณะที่ data bit เข้ามาทางขวา ใน input shift register จะทำให้ 1FFFH ที่ถูกนำไปไว้ที่ input shift register ในตอนแรกจะถูก shift out ออกไปทางซ้ายจนกระทั่งเมื่อ start bit ถูกเลื่อนไปถึงตำแหน่งซ้ายสุดใน shift register (ในโหมด 2 และ 3 รีจิสเตอร์ขนาด 9 บิท) มันจะส่งสัญญาณไปบอก RX control block ให้ทำการ shift ครั้งสุดท้ายอีก 1 ครั้ง แล้วจึงส่งสัญญาณไปโหลดเอาข้อมูลจาก shift register มาไว้ที่รีจิสเตอร์ SBUF และนำบิทที่ถูกรับเข้ามาครั้งสุดท้ายไปไว้ในบิท RB8 รวมทั้งเซทบิท RI ซึ่งเหตุการณ์ทั้งหมดนี้เกิดขึ้นก็ต่อเมื่อเงื่อนไขต่อไปนี้เป็นจริงขณะสัญญาณ shift pluse ลูกสุดท้ายถูกสร้างขึ้น

- บิท RI = 0 และ
- บิท SM2 = 0 หรือบิทที่ 9 ที่รับเข้ามามีค่าเป็น 1

ถ้าเงื่อนไขอย่างใดอย่างหนึ่งทั้งสองอย่างนี้ไม่ตรงกัน กลุ่มข้อมูลที่รับเข้ามาจะหายไปโดยไม่สามารถเรียกคืนได้และบิท RI จะไม่ถูกเซ็ท ถ้าเงื่อนไขทั้งสองถูกต้อง stop bit จะถูกนำไปไว้ในบิท RB8 ในรีจิสเตอร์ SCON ส่วนบิทที่เป็นข้อมูลจำนวน 8 บิท (data bit) จะถูกโหลดไปไว้ในรีจิสเตอร์ SBUF เมื่อเวลาในการรับผ่านไปอีก 1 บิท ไม่ว่าเงื่อนไขทั้งสองข้างต้นจะตรงหรือไม่ ระบบการรับข้อมูลก็จะกลับไปตรวจหาการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD ต่อโดยถือว่ารับข้อมูลครบ 1 ไบท์ (1 ไบท์นี้จะนับรวม start bit,programmable bit,stop bit รวมเป็น 11 บิท) เรียบร้อยแล้ว

สังเกตว่าค่าของ stop bit ที่ถูกรับเข้ามาจะไม่เกี่ยวข้องกับรีจิสเตอร์ SBUF บิท RB8 หรือ RI เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน Serial Port ทั้ง 4 โหมดที่กล่าวมานี้ การส่งข้อมูลจะเริ่มทันทีเมื่อมีคำสั่งใดๆ ที่ใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง (Destination Register) เช่น

```
MOV SBUF,A
```

ส่วนในการรับข้อมูลจะเริ่มต้นโดยมีเงื่อนไขดังนี้

- โหมด 0 บิท RI = 0 และ REN = 1
- ในโหมดอื่นๆ การรับข้อมูลเริ่มเมื่อ 8031 ได้รับ start bit เข้ามา ถ้า REN = 1

Serial Port Control Register

รีจิสเตอร์ SCON (Serial Port Control And Status Register) เป็นรีจิสเตอร์ที่ใช้สำหรับควบคุมการทำงาน Serial Port ภายใน 8031 โดยค่าของแต่ละบิทในรีจิสเตอร์ตัวนี้จะใช้สำหรับควบคุมและตรวจสอบการทำงานของ Serial Port ใน 8031 ก่อนใช้งาน Serial Port ผู้เขียนโปรแกรมจำเป็นต้องทราบถึงความหมายของบิทต่างๆ ในรีจิสเตอร์ตัวนี้

รีจิสเตอร์ตัวนี้ไม่เพียงแค่ใช้ควบคุมการทำงานของ Serial Port ในโหมดต่างๆ เท่านั้น เพราะบางบิทของรีจิสเตอร์นี้ยังใช้เป็นที่เก็บข้อมูลซึ่งเป็นบิทที่ 9 สำหรับการรับและการส่งข้อมูลในโหมด 2 และ 3 (บิท TB8 และ RB8) และนอกจากนี้รีจิสเตอร์ SCON ยังมีบิทที่ทำหน้าที่อินเทอร์รัพท์ซีพียู (Serial Port Interrupt) คือ บิท TI และ RI อีกด้วย

การใช้งาน Timer 1 เป็นตัวสร้าง Baud Rate

เมื่อ Timer 1 ถูกใช้เป็นตัวกำหนด Buad Rate (Baud Rate Generator) สำหรับการทำงานของ Serial Port ในโหมด 1 และ 3 ค่าของ Baud Rate ที่ได้จะถูกกำหนดด้วยอัตราการเกิด Overflow ของ Timer 1 และขึ้นอยู่กับบิท SMOD ในรีจิสเตอร์ PCON ซึ่งเราอาจเขียนเป็นสมการที่ใช้คำนวณหา Baud Rate ได้ดังนี้

$$\text{Baud Rate ในโหมด 1,3} = \frac{[2^{(\text{smod})} \times (\text{Oscillator Frequency})]}{}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเมื่อเกิด overflow ใน Timer ตัวใดจะทำให้ซีพียูถูกอิเทอร์รัพท์ ดังนั้น เมื่อเรานำ Timer 1 มาเป็นตัวสร้าง Baud Rate จึงควรห้ามไม่ให้เกิดอินเทอร์รัพของ Timer/Counter ขึ้นในระหว่างการรับหรือส่งข้อมูล นอกจากนี้เนื่องจากตัว Timer เองยังสามารถถูกกำหนดให้ทำงานเป็น Timer หรือ Counter อย่างใดอย่างหนึ่ง

ในการใช้งาน Serial Port ที่พบบ่อยที่สุดนั้น Timer 1 ถูกกำหนดให้ทำงานในโหมด Auto Reload (ค่า 4 บิทบนของ Tmod คือ 0010B) ในกรณีนี้ Baud Rate จะถูกกำหนด โดยสมการดังนี้

$$\text{Baud Rate ในโหมด 1,3} = \frac{[2^{(\text{smod})} \times (\text{Oscillator Frequency})]}{32 \times 12 \times [265-(\text{TH1})]}$$

เราสามารถที่จะสร้าง Baud Rate ค่าต่ำๆ ได้ด้วย Timer 1 โดยการปล่อยให้ Timer สามารถอินเทอร์รัพท์ซีพียูได้ และกำหนดการทำงานเป็น Timer ขนาด 16 บิท (โหมด 1 ค่า บิทบนของ TMOD คือ 0010B) เนื่องจากในโหมด 16 บิทไม่มีการทำงานแบบ Auto Reload

เนื่องจากในการทำงานของ Serial Port โหมด 0 จะมีความเร็วในการส่งมากที่สุดเปรียบเทียบกับโหมดอื่นที่คริสตรอลความถี่ค่าเดียวกัน และจะเห็นว่าหากเลือกใช้คริสตรอลความถี่ 11.0592 MHz จะสามารถตั้งค่า Baud Rate ในโหมด 1 และ 3 ให้เป็นค่ามาตรฐานที่ใช้กัน ทั่วไปได้ เช่น 1200, 2400, 4800, 9600, 19200 จึงเป็นเหตุผลสำคัญที่ในระบบควบคุมส่วนใหญ่ เลือกใช้คริสตรอลความถี่ 11.0592 MHz มากกว่า 12 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Memory กับ 8031

Memory ที่ใช้กับ 8031 จะมีอยู่ 2 แบบคือ

1. PROGRAM MEMORY (หน่วยความจำส่วนคำสั่ง)

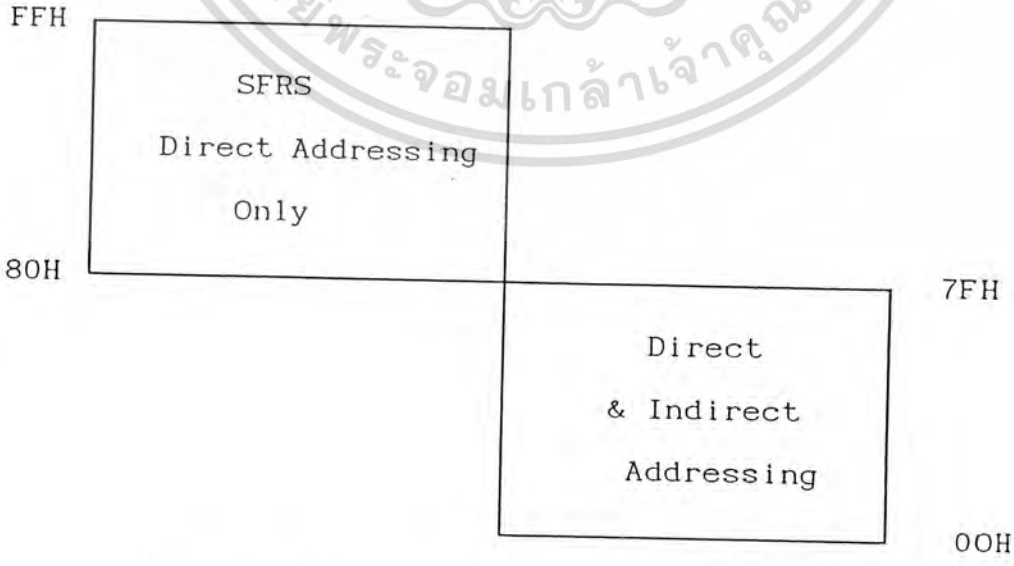
2. DATA MEMORY (หน่วยความจำส่วนของข้อมูล)

PROGRAM MEMORY คือหน่วยความจำที่ใช้ในการเก็บคำสั่ง เพื่อใช้การโปรแกรมให้เครื่องทำงาน PROGRAM MEMORY นี้ 8031 นี้จะไม่มี ROM ภายใน จึงต้องอาศัย ROM จากภายนอก เพื่อช่วยในการเก็บโปรแกรมในส่วนนี้ ดังนั้นขา EA (External Access) ต้องต่อลงกราวด์คือเลือกใช้แต่ ROM ภายนอก ตั้งแต่ ADDRESS ที่ 0000H-FFFFH

DATA MEMORY คือหน่วยความจำที่ใช้เก็บข้อมูล เพื่อให้ CPU นำข้อมูลไปใช้ ซึ่งในที่นี้ เราจะหมายถึง RAM นั่นเอง ซึ่งประกอบด้วย INTERNAL RAM และ EXTERNAL RAM

INTERNAL RAM จะมีอยู่ 128 byte สำหรับใช้เป็น Special Function Register (SFRS) และอีก 128 byte สำหรับใช้เป็นการ Access ข้อมูลซึ่งทำได้ทั้งการให้ตำแหน่งทางตรงและทางอ้อม (Direct & Indirect Accessing) ดังนั้นจะมี RAM ภายในทั้งหมด 256 byte

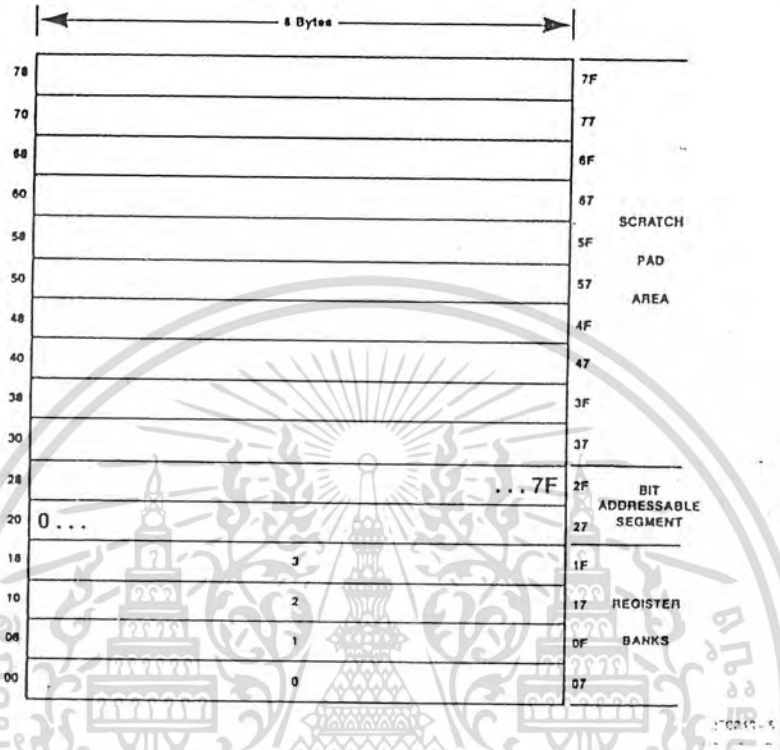
สำหรับตำแหน่ง SFRS กับ Direct & Indirect Accessing นั้น เราจะให้ SFRS เป็น 128 byte ที่อยู่สูงกว่า Direct & Indirect Accessing ซึ่งอยู่ byte ที่ต่ำกว่า จะเป็นไป ตามรูปที่ 3.1



รูปที่ 2.11 ตำแหน่ง SFRS กับ Direct & Indirect Accessing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง RAM จะประกอบด้วยส่วนต่าง ๆ ดังรูป



รูปที่ 2.12 ส่วนประกอบภายใน RAM

Direct & Indirect Addressing Area

มีอยู่ 128 byte ซึ่งเป็น byte ต่ำ เราสามารถแบ่งได้ดังนี้

1. Register bank 0-3 จะอยู่ที่ 00H-07H เป็นจำนวน 32 byte โดยที่ว่าเมื่อเริ่มเปิดเครื่องทำงาน เครื่องจะทำงานที่ bank 0 ก่อน ต่อจากนั้นจะกระโดดไปทำที่อื่นส่วน bank ที่เหลือไม่ทำ แต่จะขึ้นอยู่กับการเรียกใช้ซึ่งสามารถเรียกได้โดยใช้ Software เช่น เราจะเก็บค่า Stack Pointer (SP) ไว้ที่ตำแหน่ง 07H ตัวของ CPU จะเพิ่มตำแหน่งขึ้น 1 ตำแหน่ง แล้วจึงค่อยทำที่ 08H คือที่ R_0 ของ bank 1 จะสังเกตเห็นว่าจะต้องใช้มากกว่า 1 bank ดังนั้นการเก็บค่า stack pointer เราควรจะเก็บไว้ในตำแหน่งที่แตกต่างกันของ RAM เมื่อไม่ได้ใช้ในการเก็บค่า data

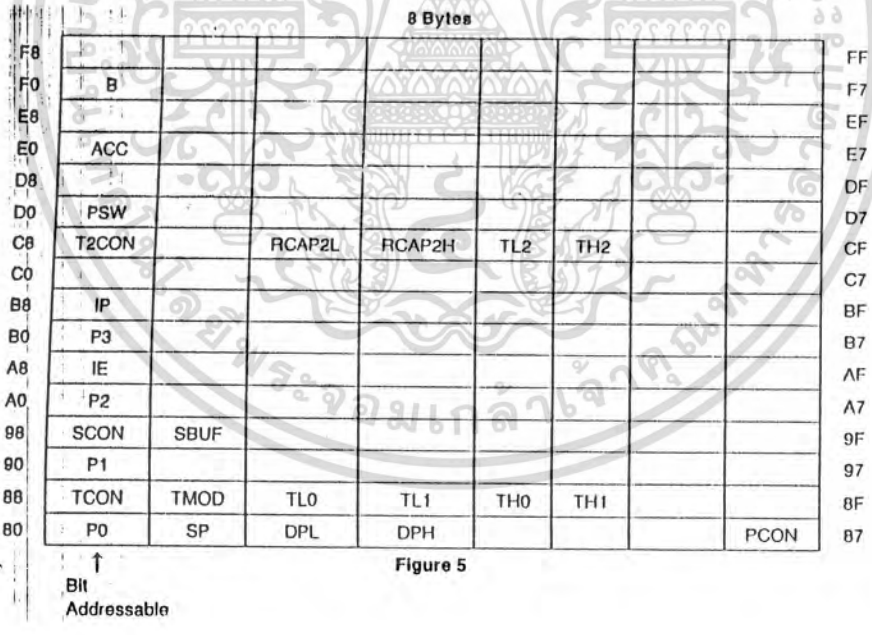
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. BIT ADDRESS AREA จะอยู่ที่ 20H-2FH ซึ่งจะมีทั้งหมด 16 byte หรือ 128 bit ลักษณะเด่นของ RAM ที่ ADDRESS นี้ คือ สามารถเก็บข้อมูลเป็นบิทได้ นั่นคือถ้าเลือกใช้เป็น byte จะเป็น 20H-2FH แต่ถ้าใช้เป็น BIT เราสามารถเจาะจงเก็บเป็น byte ใด byte หนึ่งก็ได้ โดยการเขียนที่โปรแกรม เช่นถ้า byte ที่ 20 เราก็จะเลือกใช้ได้ตั้งแต่ 20.1-20.7 ได้หรือเป็น 21.0-21.7 เป็นต้น

3. Scratch Pad Area คือตั้งแต่บริเวณ 30H-7FH จะสามารถเก็บข้อมูลไว้ได้ตามปกติ แต่อย่างไรก็ตาม ถ้า Stack Pointer ถูกเก็บอยู่บริเวณนี้ ควรจะมีจำนวน byte ที่เพียงพอสำหรับเก็บค่าดังกล่าว และให้ Address เหล่านี้แยกตัวออกมาจาก Memory เพื่อป้องกัน ค่าของ SP เปลี่ยน

Special Function Register (SFRS)

จะเป็น RAM ขนาด 128 byte อยู่ที่ Address 80H-FFH ซึ่งมีค่าต่างๆ ดังรูปที่ 2.13



รูปที่ 2.13 Memory Map ของ SFRS

ตรง Address ที่ว่างอยู่ สามารถนำข้อมูลที่ต้องการไปเก็บได้

ส่วนตรงที่มีค่าต่างๆ จะถูกจองไว้เพื่อเป็นค่าต่างเหล่านี้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = 8052 only

ตารางที่ 2.1 แสดงค่าต่างๆ ใน SFRS

ซึ่งเมื่อเวลาเปิดเครื่อง CPU จะเริ่มทำงาน ค่าต่างๆ ใน SFRS จะถูก set ดังตารางนี้

Register	Value In Binary
*ACC	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*P0	11111111
*P1	11111111
*P2	11111111
*P3	11111111
*IP	8051 XXX00000, 8052 XX000000
*IE	8051 0XX00000, 8052 0X000000
TMOD	00000000
*TCON	00000000
*+T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
+TH2	00000000
+TL2	00000000
+RCAP2H	00000000
+RCAP2L	00000000
*SCON	00000000
SBUF	Indeterminate
PCON	HMOS 0XXXXXXX CHMOS 0XXX0000

X = Undefined
* = Bit Addressable
+ = 8052 only

ตารางที่ 2.2 แสดงค่าต่างๆ ใน SFRS หลังการรีเซ็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน byte ต่างๆ ที่ถูกจองไว้ สามารถอธิบายได้คร่าวๆ ดังนี้

PSW (Programable Status Word)

จะเปรียบเสมือน FLAG ของ Z-80 ใน byte นี้จะมีค่าต่างๆ ใน bit เป็นดังรูป

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	—	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).
OV	PSW.2	Overflow Flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator.

NOTE:
1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

รูปที่ 2.14

PCON (Power Control Register)

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD	Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.
—	Not implemented, reserved for future use.*
—	Not implemented, reserved for future use.*
—	Not implemented, reserved for future use.*
GF1	General purpose flag bit.
GF0	General purpose flag bit.
PD	Power Down bit. Setting this bit activates Power Down operation in the 80C51BH. (Available only in CHMOS).
IDL	Idle Mode bit. Setting this bit activates Idle Mode operation in the 80C51BH. (Available only in CHMOS).

If 1s are written to PD and IDL at the same time, PD takes precedence.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

รูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการ Interrupt ก็จะมีตัวคำสั่งที่ใช้ควบคุมอยู่ใน byte ต่างๆ ใน RAM ดังนี้

IE (Interrupt Enable Register)

เป็นการ set การเลือกการ Interrupt ว่าจะขอ Interrupt ให้กับใคร ดังรูปที่ 3.8 ซึ่งแต่ละ bit ก็คือ การ Interrupt ให้กับแต่ละ Mode การทำงานนั่นเอง

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
ET2	IE.5	Enable or disable the Timer 2 overflow or capture interrupt (8052 only).
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

รูปที่ 2.16

IP (Interrupt Priority Register)

คือการเซทว่าให้ใน Mode ใด มีค่าสำคัญมากกว่ากัน ปกติถ้าไม่มีการ set จะเรียงตามลำดับจากมากไปหาน้อย ดังนี้

IE0

TF0

IE

TF1

R1 or T1

TF2 or EXF2

แต่ถ้ามีการเซทขึ้นมา เราจะพิจารณา Mode ที่ set ไว้ก่อน เช่น set ว่าให้ความสำคัญของ IE ก่อน เมื่อมีการ Interrupt เราจะไปที่ mode ของ IE ก่อนที่จะพิจารณาที่ IE0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อควรระวังในการเซ็ทก็คือ ถ้ามีการให้ความสำคัญพร้อมกันมากกว่า 1 Mode วิธีพิจารณาเรียงตามความสำคัญดังที่กล่าวมาเช่นเดิมเช่น ถ้ามีการให้ความสำคัญระหว่าง IE พร้อมกันกับ TF1 ถ้ามีการขอ Interrupt พร้อมๆ กัน ทาง CPU จะให้ใน Mode ของ IE ก่อน

การเซ็ทเราจะดูได้ว่า bit ได้เป็นของ mode ใดจากรูปที่ 2.17

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

—	—	PT2	PS	PT1	PX1	PT0	PX0

—	IP. 7 Not implemented, reserved for future use.*
—	IP. 6 Not implemented, reserved for future use.*
PT2	IP. 5 Defines the Timer 2 interrupt priority level (8052 only).
PS	IP. 4 Defines the Serial Port interrupt priority level.
PT1	IP. 3 Defines the Timer 1 interrupt priority level.
PX1	IP. 2 Defines External Interrupt 1 priority level.
PT0	IP. 1 Defines the Timer 0 interrupt priority level.
PX0	IP. 0 Defines the External Interrupt 0 priority level.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

รูปที่ 2.17

ในการเซทในส่วนของ Timer เราจะมีอยู่ใน RAM 2 byte ก็คือ

TCON (Timer/Counter Control Register)

การเซททีนี่ใช้สำหรับเซทว่าจะให้ Timer/Counter ทำงานเป็นอะไรในวงจร เช่นที่ bit ที่ TCON.4 เราจะให้ Timer 0 ทำงานและจะ SET/CLEAR โดย Software เพื่อที่จะกลับไปที่ Timer/Counter 0 on/off เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับจะให้ทำงานเป็นอะไร เราจะเซ็ทได้ตามรูปที่ 2.18 ดังนี้

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON. 7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON. 6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON. 5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON. 4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON. 3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON. 2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON. 1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON. 0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

รูปที่ 2.18

TIMER/COUNTER Mode Control Register

ใช้สำหรับเซทค่าต่างๆว่าจะให้ทำงาน Mode ใด ซึ่ง Mode ต่างๆ สามารถดูได้จากรูปที่ 2.19

M1	M0	Operating Mode	
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

รูปที่ 2.19

ในการเซทค่าบิทต่างๆ เพื่อเลือก Mode เราสามารถจะเลือกโดยดูจาก bit ต่างๆ ตามรูปที่ 2.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/$\overline{T}$ Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

รูปที่ 2.20

ในส่วนของการส่งข้อมูลแบบอนุกรม (Seraial Interface) เราสามารถจะเซ็ทค่าได้ โดยการเซ็ทที่ byte ดังกล่าว ซึ่งจะมีค่าตาม bit ดังนี้

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 SCON. 7 Serial Port mode specifier. (NOTE 1).

SM1 SCON. 6 Serial Port mode specifier. (NOTE 1).

SM2 SCON. 5 Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 9).

REN SCON. 4 Set/Cleared by software to Enable/Disable reception.

TB8 SCON. 3 The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.

RB8 SCON. 2 In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.

TI SCON. 1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.

RI SCON. 0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

รูปที่ 2.21

ซึ่งในการ set Mode ของการส่งข้อมูลแบบอนุกรม เราสามารถจะเซ็ทได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

SERIAL PORT SET-UP:

Table 9

MODE	SCON	SM2 VARIATION
0 1 2 3	10H 50H 90H D0H	Single Processor Environment (SM2 = 0)
0 1 2 3	NA 70H B0H F0H	Multiprocessor Environment (SM2 = 1)

รูปที่ 2.22

การเซ็ทค่าต่างๆ เราสามารถทำได้โดยการโปรแกรม ซึ่งเราจะส่งข้อมูลไปที่ตำแหน่งนั้นแล้วทางซีพียูจะเช็คเองว่าอยู่ในเงื่อนไขใด และจะทำการเซ็ทค่าต่างๆ ในนั้นเอง เช่น

```
MOV A8,#8FH
```

นั่นคือ 8FH จะถูกส่งค่าไปที่ Address A8 ซึ่งจากรูปจะเป็นๆ Mode ของ IE ซึ่งก็คือ การ Interrupt ดังนั้น IE จะถูก set ด้วยค่า 8FH ก็คือค่าต่างๆ ใน IE จะเป็นดังรูป

EA	--	ET2	ES	ET1	EX2	ET0	EX0

จะเป็น

1	0	0	0	1	1	1	1

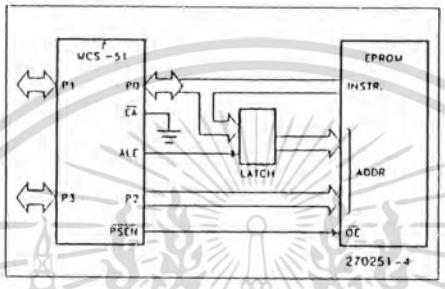
เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อระหว่าง Memory กับ CPU

เนื่องจากเรามี Memory ภายนอกที่ต้องการใช้ในการเก็บค่าข้อมูลหรือคำสั่งจึงจำเป็นต้องให้ CPU ติดต่อกับ Memory เหล่านั้นได้ โดยการเชื่อมต่อกัน ซึ่งเราจะมาพิจารณาเป็นกรณีดังต่อไปนี้

1. ติดต่อกับ ROM หรือ EPROM ภายนอก เราสามารถต่อได้ดังนี้



รูปที่ 2.23 การเชื่อมต่อ CPU กับ ROM ภายนอก

การติดต่อกับ ROM ภายนอก เราต้องใช้ขา PSEN (Program Store Enable) ของ CPU โดยเงื่อนไขที่ว่า ถ้า PSEN ทำงาน นั่นหมายถึงเราต้องการติดต่อกับ ROM ภายนอก ซึ่งถ้าเราพิจารณาแล้วจะเห็นว่าจะคล้ายกับขา MREQ ของ Z-80 เพียงแต่เราจะใช้กับ ROM เท่านั้น โดยการนำ PSEN ไปต่อกับ OE (Output Enable) ของ ROM เพื่อเป็นตัวบอกว่า CPU ต้องการจะติดต่อกับ ROM ตัวนั้น

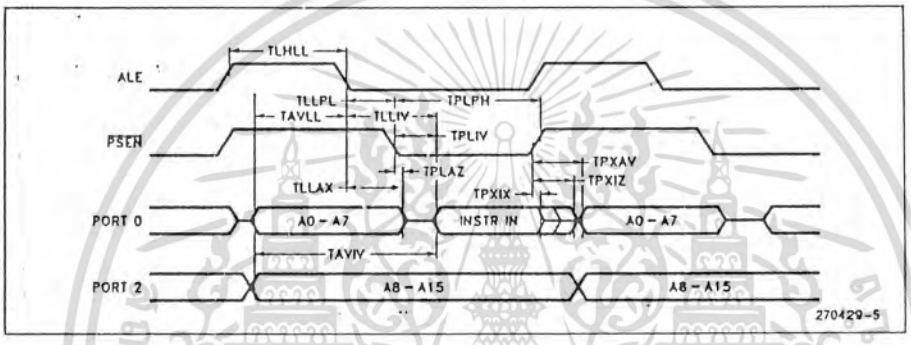
เนื่องจาก 8031 จะต่อขา EA ลงกราวด์ทำให้สามารถใช้ ROM ภายนอกได้ทั้งหมด 0000H-FFFFH จากรูป เราจะนำเอา port 2 มาทำเป็น Address ไบท์สูงในขณะที่นำ port 0 มาทำเป็น address ไบท์ต่ำ

จะสังเกตุได้ว่า port 0 เราใช้ร่วมกันในการส่งค่า data เข้า ROM กับการส่งค่า Address ไบท์ต่ำ ดังนั้นจึงมีการไป latch ค่าของข้อมูลไว้ก่อน ที่จะมีข้อมูลตามมา จะทำให้ ค่า Address ไม่ให้หายไป แต่การที่จะเลือกว่า data ที่ส่งออกมามีค่าเป็น Address หรือ Data เราจะมีขาอีก 1 ขาที่ CPU คือ ขา ALE คือถ้า ALE ACTIVE จะทำให้ latch ทำงานคงค่าไว้ นั่นหมายความว่าข้อมูลที่เข้ามาเป็น Address ไบท์ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าเป็น 0 ก็จะเป็นข้อมูลธรรมดา ดังนั้นเราสามารถที่จะติดต่อกับ ROM ได้ เช่น ถ้ามี ROM ขนาด 4 byte เราจะต้องใช้ Address Bus 12 เส้น คือ A0-A7 ออกมาจาก port 0 และใช้ port 2 จำนวน 4 เส้น เป็นต้น

เราวิเคราะห์การทำงานของ CPU ในการติดต่อ ROM ภายนอกได้โดยดูจาก TIming Diagram



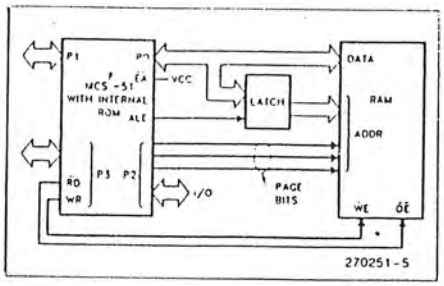
รูปที่ 2.24

2. การติดต่อกับ RAM ภายนอก

การติดต่อกับ RAM ภายนอกจะคล้ายกับการต่อแบบ ROM เพียงแต่เราจะไม่ใช้ PSEN แต่เราจะใช้ขา WR, RD ในการอ่านค่าหรือเขียนข้อมูล จากส่วนของ RAM ภายนอกดังกล่าวโดยนำไปต่อกับขา WR, OE ของ RAM เพื่อเป็นตัวบอก RAM ว่าเราเขียนหรืออ่าน และเช่นเดียวกันกับ ROM จะต้องมีการ latch ข้อมูล เพราะเกิดมีการใช้ร่วมกันเหมือนกรณีของ ROM

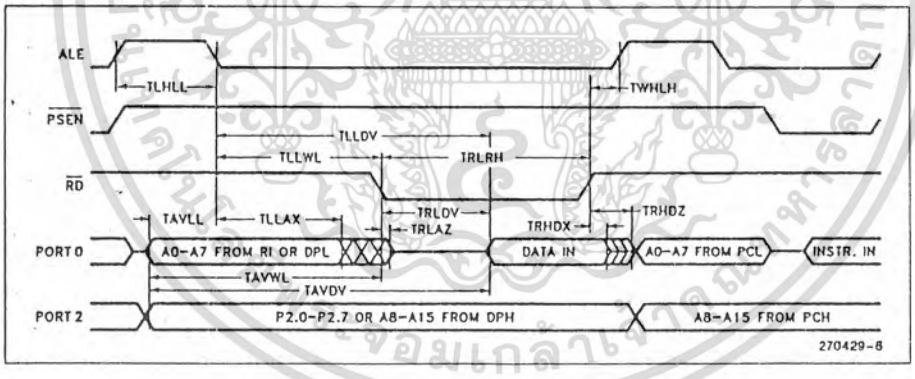
ในเรื่องจำนวนสายก็ขึ้นอยู่กับขนาดของ RAM ถ้าไม่เต็ม 64 byte ก็จะใช้สายไม่หมดทั้ง 16 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 การเชื่อมต่อ CPU กับ RAM ภายนอก

เราสามารถจะวิเคราะห์การทำงานของ CPU ในการติดต่อกับ RAM ได้โดยดูจาก Timing Diagram



รูปที่ 2.26

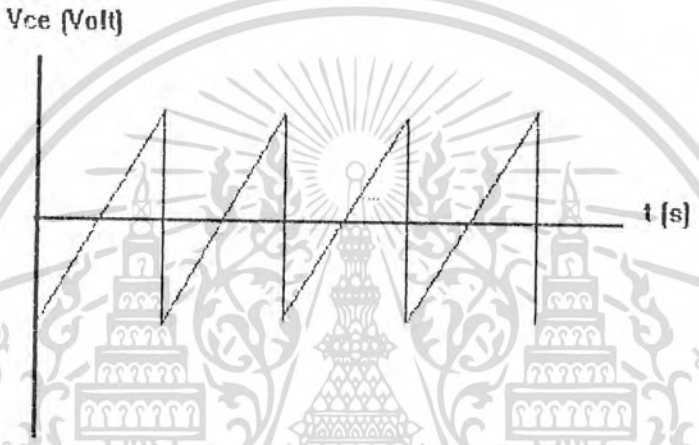
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การตรวจสอบไดโอดและทรานซิสเตอร์

การตรวจสอบไดโอด

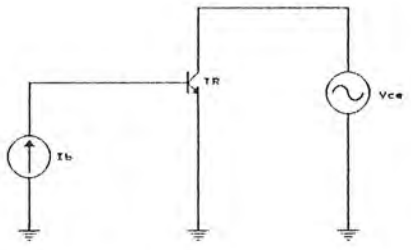
สำหรับการการตรวจสอบไดโอดนั้นทำได้โดยการป้อนแรงดันให้แก่ไดโอด โดยจะเปลี่ยนค่าตั้งแต่ลบมากที่สุดถึงบวกมากที่สุด โดยจะเปลี่ยนไปตาม range ที่ผู้ต้องการทดสอบกำหนด ซึ่งกราฟของแรงดันที่ป้อนจะเป็นลักษณะของฟันเลื่อยโดยแสดงไว้ในรูป 3.1



รูปที่ 3.1

การตรวจสอบทรานซิสเตอร์

ในการตรวจสอบทรานซิสเตอร์นั้นจะต้องมีแหล่งจ่าย 2 อย่าง คือแหล่งจ่ายกระแสและแหล่งจ่ายแรงดัน ซึ่งสามารถแสดงได้ดังรูปที่ 3.2



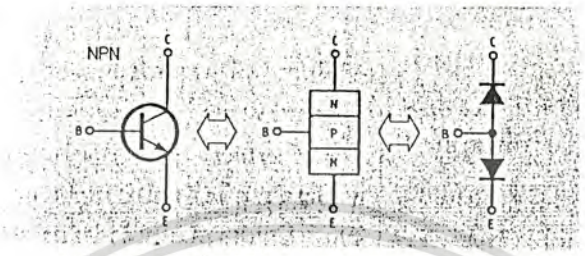
รูปที่ 3.2 การไบอัสทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากทรานซิสเตอร์มีอยู่ 2 ชนิด คือ NPN และ PNP ดังนั้นกระแสและแรงดันที่ป้อนให้ทรานซิสเตอร์ทั้งสองจะต่างกัน

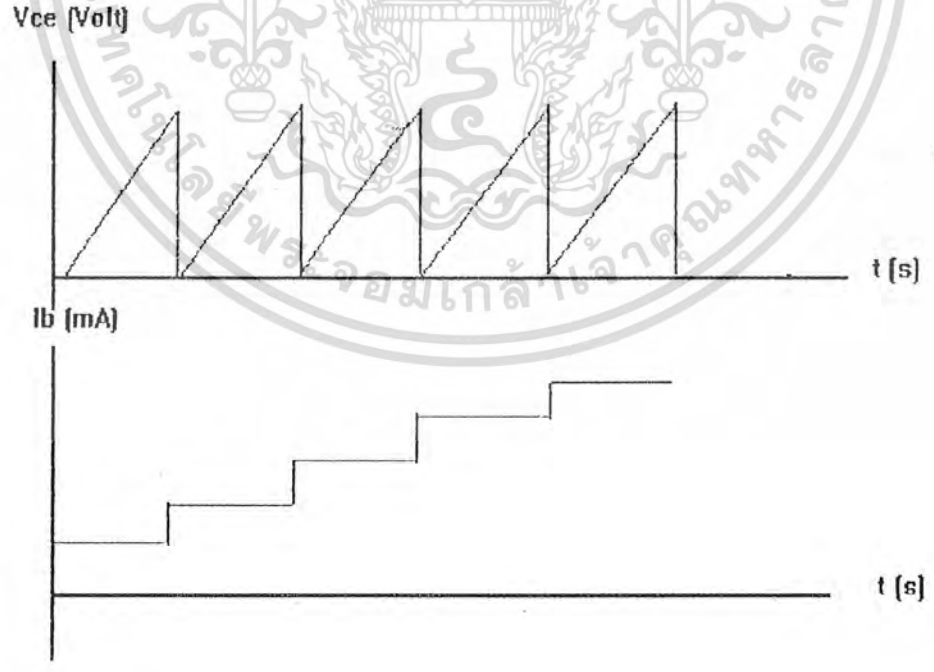
<u>การตรวจสอบทรานซิสเตอร์ชนิด NPN</u>

โครงสร้างของทรานซิสเตอร์ NPN แสดงไว้ดังรูปที่ 3.3



รูปที่ 3.3 โครงสร้างทรานซิสเตอร์ชนิด NPN

เนื่องจากกราฟของทรานซิสเตอร์เกิดจาก การเปลี่ยนแปลงของกระแส I_c ต่อการเปลี่ยนแปลงของ V_{CE} ที่ I_b ค่าต่างๆ ดังนั้นเครื่องตรวจสอบไดโอดและทรานซิสเตอร์จะต้องทำการสร้างแหล่งจ่ายกระแส (I_b) และแหล่งจ่ายแรงดัน (V_{CE}) โดยเปลี่ยนค่าของ V_{CE} ตั้งแต่ค่าน้อยที่สุดถึงมากที่สุด (บวก) ต่อค่าของกระแส (I_b) ค่าหนึ่ง ซึ่งสามารถแสดงได้ดังรูป

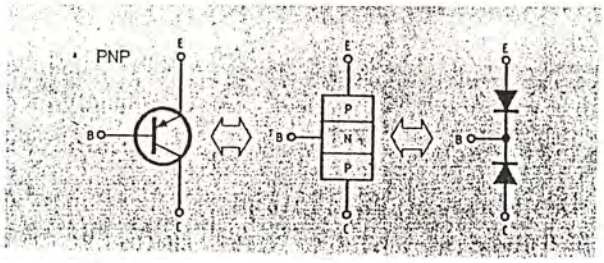


รูปที่ 3.4 การป้อนแรงดันและกระแสให้ทรานซิสเตอร์ชนิด NPN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

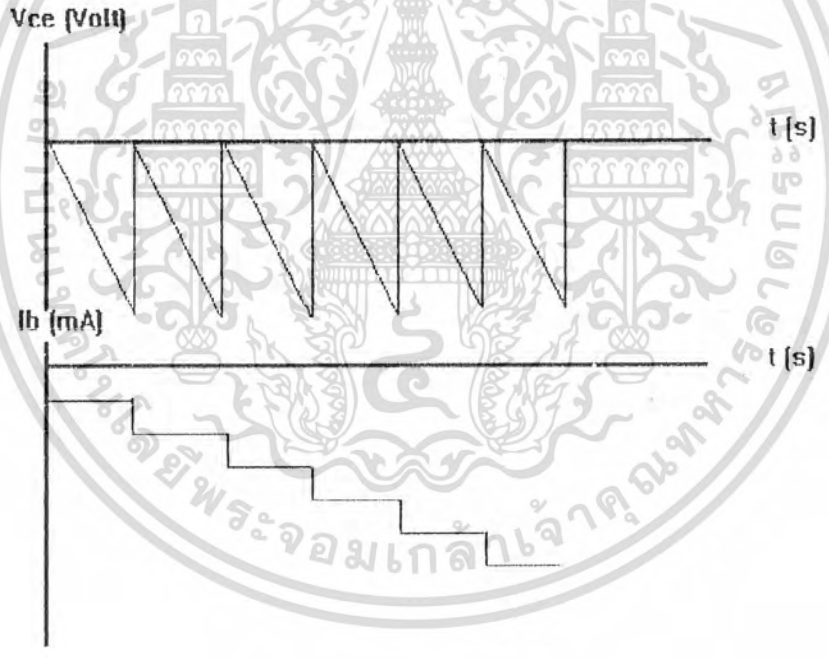
การตรวจสอบทรานซิสเตอร์ชนิด PNP

โครงสร้างของทรานซิสเตอร์ PNP แสดงไว้ดังรูปที่ 3.5



รูปที่ 3.5 โครงสร้างทรานซิสเตอร์ชนิด PNP

เช่นเดียวกับทรานซิสเตอร์ชนิด NPN แต่ในการป้อนกระแสและแรงดันจะกลับกันคือ จะเป็นค่าลบ ซึ่งก็สามารถแสดงได้ดังรูป

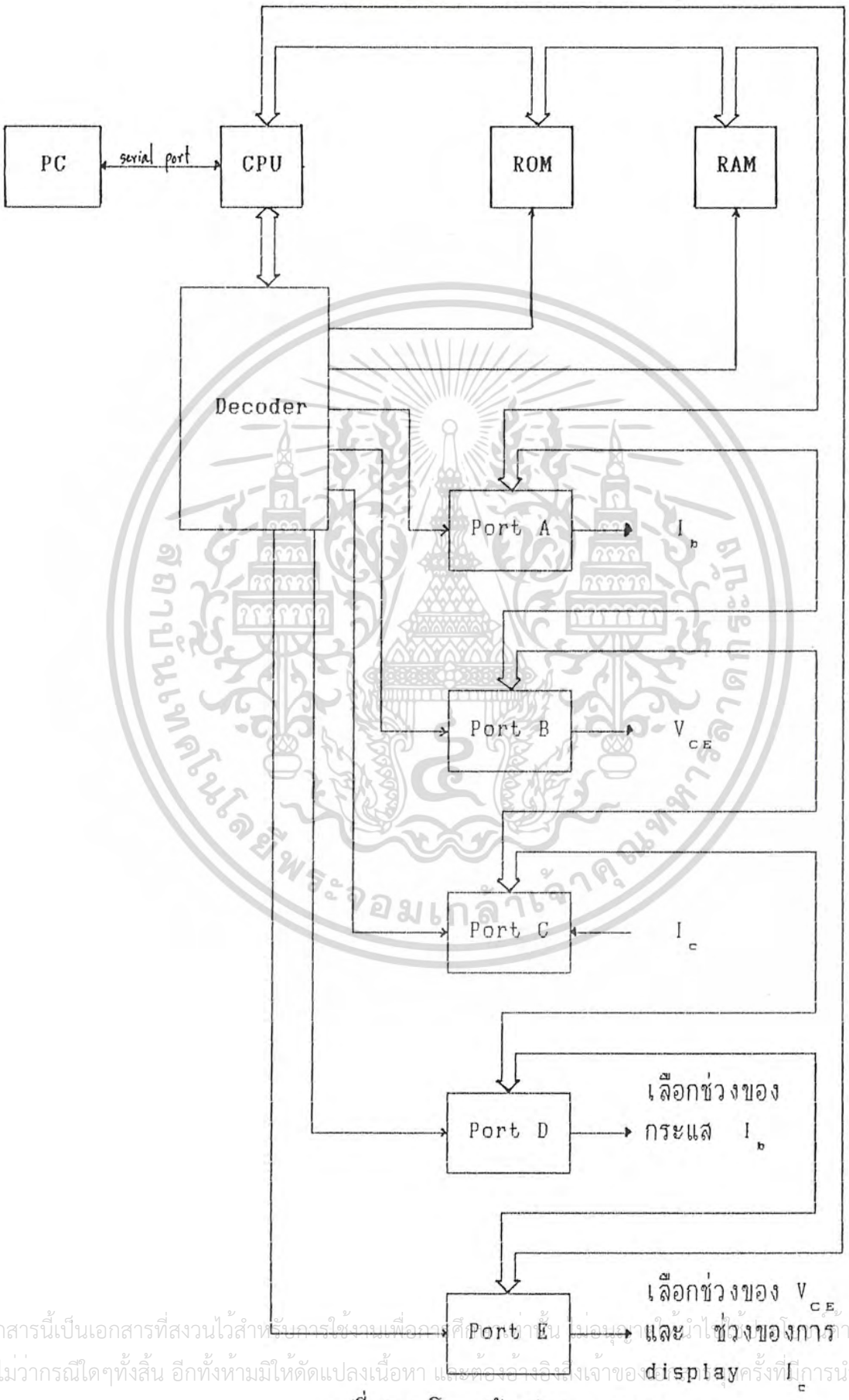


รูปที่ 3.6 การป้อนแรงดันและกระแสให้ทรานซิสเตอร์ชนิด PNP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ส่วน Control



รูปที่ 4.1 โครงสร้างส่วน Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานครั้งที่มีการนำไปใช้

ส่วน Control นี้ใช้ Microprocessor 8031 เป็น CPU ในการควบคุมการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับส่วน Control และระหว่างวงจรไบอัสและวัดกระแสกับส่วน Control โดยส่วนควบคุมนี้จะรับข้อมูล (ซึ่งถูกกำหนดจากผู้ใช้) ที่เป็นตัวกำหนดชนิดของอุปกรณ์ ช่วงของแหล่งจ่ายกระแส (CSVM) I_B ช่วงของแหล่งจ่ายแรงดัน (VSCM) V_{CE} และช่วงของการ display กระแส I_C จากคอมพิวเตอร์ผ่านทาง serial port และนำข้อมูลนี้ไปกำหนด ช่วงของการเปลี่ยนแปลงแรงดัน V_{CE} และกระแส I_B และส่งค่าที่กำหนดนี้ให้กับวงจรไบอัสและวัดกระแส ส่วนการรับค่ากระแส I_C จากวงจรไบอัสและวัดกระแสจะรับจนครบตามจำนวน step ของ I_B เสียก่อนจึงส่งค่ากระแส I_C ให้กับคอมพิวเตอร์ คอมพิวเตอร์จะนำค่า I_C ที่รับได้ประมวลผลเพื่อแสดงเป็น V-I CHaracteristic curve บนจอคอมพิวเตอร์ต่อไป

การรับส่งข้อมูลจากส่วน Control ไปยังวงจรไบอัสและวัดกระแส สามารถส่งได้โดยการสร้าง I/O port ที่มีคุณสมบัติของการ latch ค่าไว้ได้โดยใช้ทั้งหมด 5 port

port A สำหรับส่งค่ากระแส I_B

port B สำหรับส่งค่าแรงดัน V_{CE}

port C สำหรับการรับค่ากระแส I_C

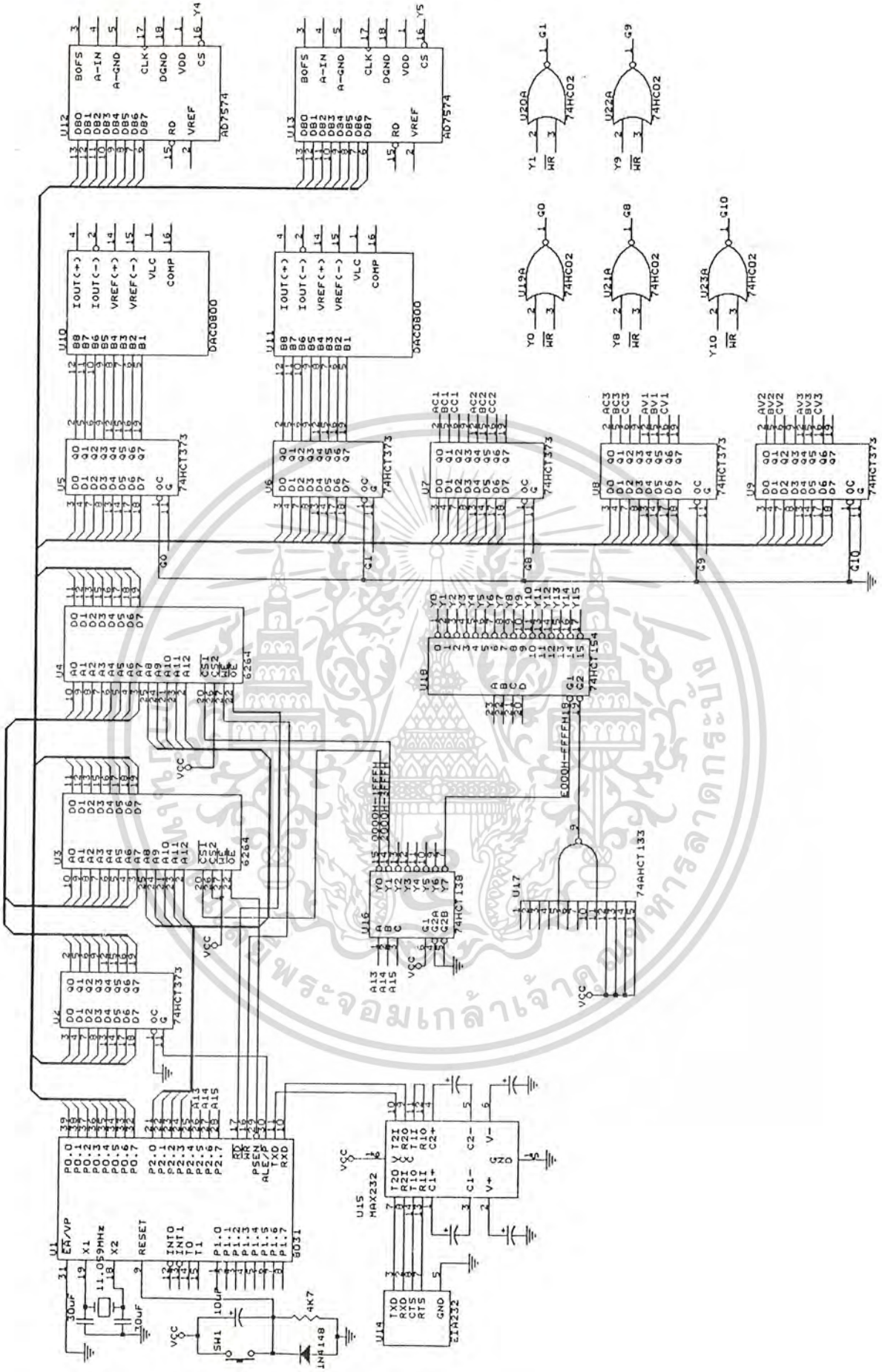
port D สำหรับใช้ในการควบคุม analog switch ที่ใช้เลือกช่วงของการจ่ายกระแส $I_{b,max}$

port E สำหรับใช้ในการควบคุม analog switch ที่ใช้เลือกช่วงของการจ่ายแรงดัน $V_{CE,max}$ และช่วงของกระแส $I_{c,max}$

ในส่วนของการจ่ายแรงดัน V_{CE} และกระแส I_B จะต้องมี digital to analog converter และส่วนของการรับกระแสจะต้องมี analog to digital converter เพื่อแปลงลักษณะสัญญาณจาก analog เป็น digital ด้วย

ส่วนการต่อ Memory ของ 8031 ก็ดังที่ได้กล่าวมาแล้วในเรื่องการเชื่อมต่อ 8031 กับ Memory แล้ว ซึ่ง ROM ที่เราใช้คือ RAM-PACK ขนาด 8 Kbyte การติดต่อกับ RAM-PACK เราต้องใช้ขา PSEN (Program Store Enable) ของ CPU ไปต่อกับ OE (Output Enable) ของ RAM-PACK เพื่อเป็นตัวบอกว่า CPU ต้องการจะติดต่อกับ RAM-PACK

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 วงจรส่วน Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก 8031 จะต่อขา EA ลงกราวด์ ทำให้สามารถใช้ตำแหน่งของ External memory 8031 ได้ทั้งหมด ตั้งแต่ 0000H-FFFFH จากรูปที่ 4.2 เราใช้ port 2 เป็น Address ไบท์สูง และใช้ port 0 เป็น Address ไบท์ต่ำ

เนื่องจาก port 0 เราใช้ร่วมกันในการส่งค่า dataและ address byte ต่ำ ดังนั้นจึงมีการ latch ค่าของข้อมูลที่เป็นค่า address ไว้ก่อนที่จะมีข้อมูลตามมาจะทำให้ค่า Address ไม่ให้หายไป แต่การที่จะเลือกว่า data ที่ส่งออกมามีค่าเป็น Address หรือ Data เราจะมีขาอีก 1 ขาที่ CPU คือ ขา ALE คือถ้า ALE ACTIVE จะทำให้ latch ทำงานคงค่าไว้ นั่นหมายความว่าข้อมูลที่เข้ามาเป็น Address ไบท์ต่ำ แต่ถ้า ALE ไม่ ACTIVE ข้อมูลที่ port 0 ก็จะไม่ถูก latch ไว้

ส่วนการติดต่อกับ RAM ภายนอก เราจะใช้ขา WR, RD ของ 8031 ไปต่อกับขา WR, OE ของ RAM เพื่อเป็นตัวบอก RAM ว่าเราเขียนหรืออ่าน

เนื่องจาก RAM-PACK และ RAM ภายนอกมีขนาด 8 Kbyte ดังนั้นในการเชื่อมต่อเราจึงใช้ Address Bus 13 เส้นในการเชื่อมต่อ คือ A0-A7 ออกมาจาก port 0 และใช้ port 2 จำนวน 5 เส้น

สำหรับ Memory MAP ในวงจรส่วน Control นี้กำหนดให้ RAM-PACK และ ROM อยู่ใน page เดียวกันรวมทั้ง I/O port ดังนั้นจะใช้ Decoder ขนาด 3 to 8 (74HCT138) ในการแยก Address โดยใช้ A13, A14, A15 มา decode โดยให้

RAM-PACK อยู่ในช่วง 0000H-1FFFH

RAM อยู่ในช่วง 2000H-3FFFH

I/O port อยู่ในช่วง E000H-FFFFH

ในส่วนของ I/O port address ที่ใช้งานจริงมี 6 ตำแหน่งคือ

E000H ใช้เป็น port A

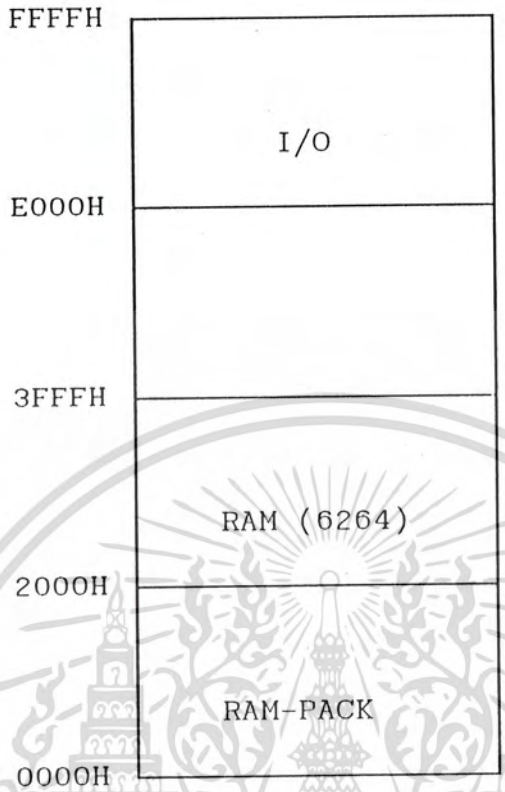
E001H ใช้เป็น port B

E004H ใช้เป็น port C

E008H-E00AH ใช้เป็น port D และ E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถเขียนเป็น Memory Map ได้ดังนี้



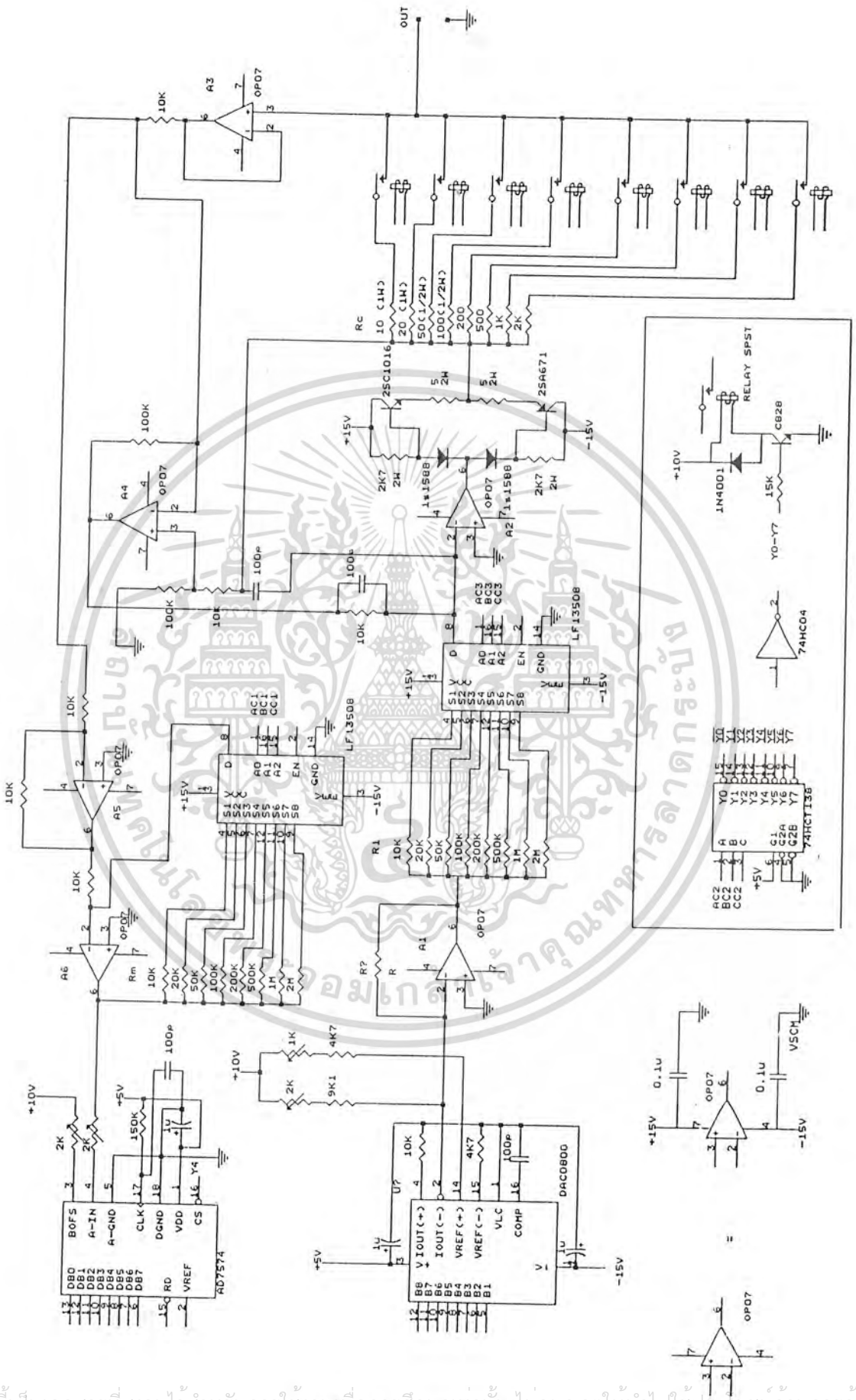
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วงจร Bias และวัดค่ากระแส

ในการสร้างกราฟ Characteristic ต้องมีการสร้าง source ให้ไดโอด คือ Voltage Source Current Monitor (VSCM) เพื่อป้อนให้ไดโอด แล้วจะส่งเป็นค่ากระแสกลับไปยังคอมพิวเตอร์เพื่อสร้างกราฟต่อไป ส่วนทรานซิสเตอร์จะต้องมี Source อีกตัว คือ Current Source Votage Monitor (CSVM) เพื่อไบอัสให้ทรานซิสเตอร์ โดยจะป้อน VSCM ที่ CE (V_{CE}) ของทรานซิสเตอร์ และ CSVM ที่ขาเบสเพื่อให้เป็นกระแสไบอัส I_b เมื่อทรานซิสเตอร์ได้รับไบอัสแล้ว ทรานซิสเตอร์ก็จะทำงานโดยจะมีกระแส I_c ไหล เราจะนำค่า I_c มาวิเคราะห์เพื่อสร้างกราฟต่อไป

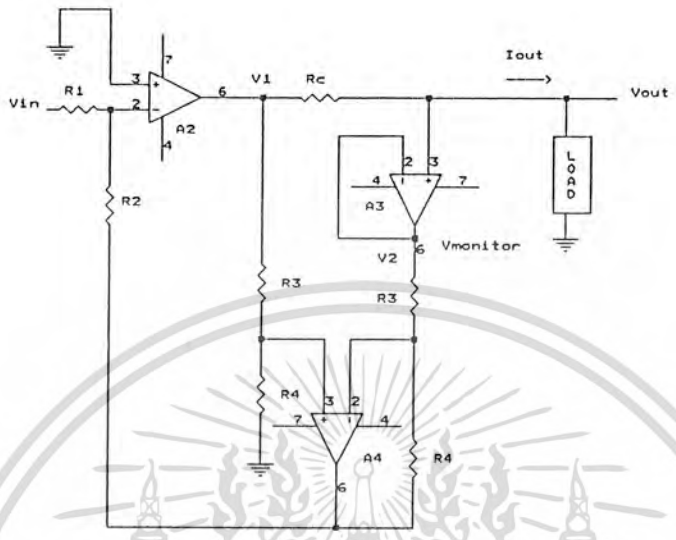
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 วงจร CSVM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนวงจร CSVM สามารถเขียนเป็น block diagram ได้ดังนี้



รูปที่ 5.2 Block diagram ของ CSVM

จะได้ค่าของ $I_{out} = \frac{R_2}{R_1}\frac{R_3}{R_4}\left(\frac{-V_{in}}{R_c}\right)$

$V_{monitor} = V_{out}$

ค่า I_{out} จะเป็นฟังก์ชันของ V_{in} ซึ่ง V_{in} ของวงจรนี้ได้มาจาก o/p ของวงจร D/A Converter (DAC0800) โดย V_{in} นี้จะมากขึ้นหรือน้อยลงตามค่าของ DB0-DB7 ที่ส่งมาจาก 8031 microcontroller โดยค่าที่ส่งมานี้จะมีค่าตั้ง แต่ 00H-FFH โดยที่ค่า 80H จะให้ค่าของ V_{in} เป็น 0 V และ 00H จะให้ค่าสูงสุดทางลบ (V_{in} เป็น บวก) และ FFH จะให้ค่าสูงสุดทางบวก (V_{in} เป็นลบ) เนื่องจาก A1 มีอัตราขยายเป็นลบ โดย V_{in} สูงสุดและต่ำสุดจะอยู่ในช่วง +10V ถึง -10V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R_1 จะเป็นตัวกำหนด Range อย่างกว้างให้กระแส O/P และ R_c จะทำหน้าที่ปรับ range ให้ละเอียดขึ้นอีก คือเปลี่ยนค่า min และ max ให้ลดลงเป็นจำนวนเท่าของ range ที่กำหนดโดยชุด CSVM นี้จะได้ o/p เป็นกระแส ซึ่งเราจะนำไปเป็นกระแสไบอัส I_b ดังนั้น จึงนำไปต่อกับขาเบสของทรานซิสเตอร์ที่เราต้องการจะวัด Characteristic ค่าที่ส่งกลับมานั้นจะเป็นโวลต์เตจที่ตกคร่อมขาเบสที่เปลี่ยนแปลงตามค่าของกระแส I_b

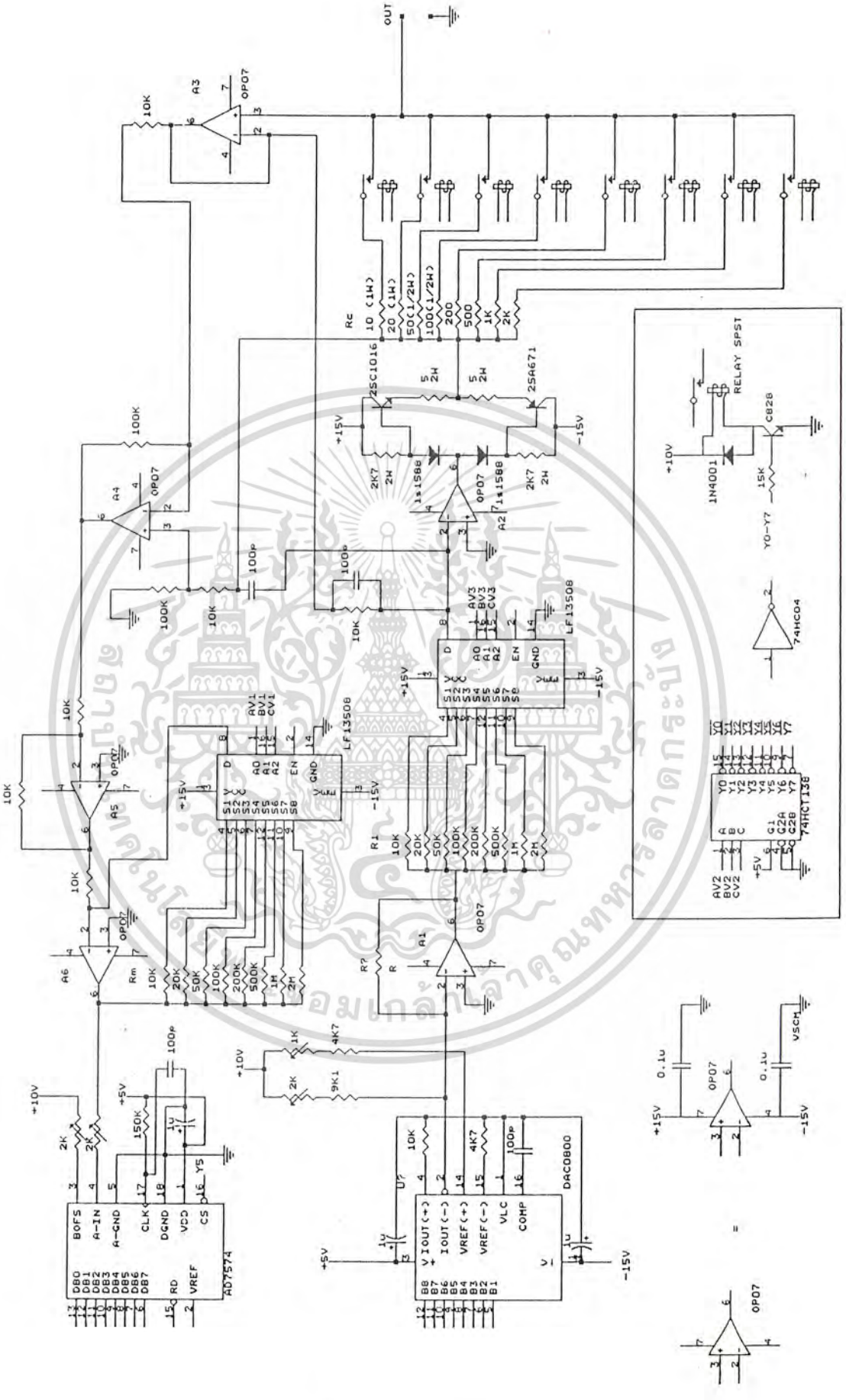
出力電流						検出電圧		
R_1SW	R_1	CS	倍率	R_cSW	R_c	R_MSW	R_M	VM
1	10k	±100mA	1	1	10Ω	1	10k	±10V
2	20k	±50mA	1/2	2	20Ω	2	20k	±5V
3	50k	±20mA	1/5	3	50Ω	3	50k	±2V
4	100k	±10mA	1/10	4	100Ω	4	100k	±1V
5	200k	±5mA	1/20	5	200Ω	5	200k	±0.5V
6	500k	±2mA	1/50	6	500Ω	6	500k	±0.2V
7	1M	±1mA	1/100	7	1k	7	1M	±0.1V
8	2M	±500μA	1/200	8	2k	8	2M	±0.05V

ตารางที่ 5.1 ตารางแสดงเอาท์พุตของ CSVM

ตัวอย่างจากตาราง

ถ้าเราเลือก R_1 10K R_c 20 โอห์ม (1/2 เท่า) เราจะได้ range ของ I_b จะอยู่ในช่วง -100 mV ถึง +100 mV และ range จะถูกทำให้แคบลงโดย R_c ดังนั้นกระแสเอาท์พุต I_b อยู่ในช่วง $\frac{+100mA}{2}$ และ $\frac{-100mA}{2}$ ซึ่งเท่ากับ +50mA และ -50mA ส่วน $V_{monitor}$ จะมีค่าเท่ากับ V_{out} (V_b) ซึ่งเราไม่ได้นำกลับมาแสดงผล จึงไม่ขอกล่าวในที่นี้

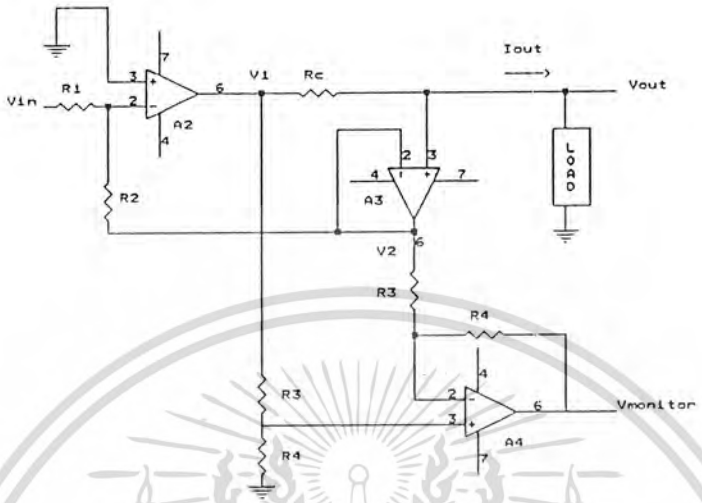
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 วงจร VSCM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร VSCM สามารถเขียนเป็น block diagram ได้ดังนี้



รูปที่ 5.4 block diagram ของวงจร VSCM

จะได้ค่าของ

$$V_{out} = \frac{R_2}{R_1}(-V_{in})$$

$$V_{monitor} = \frac{R_4}{R_3} R_c I_{out}$$

ซึ่ง V_{in} ของวงจรนี้ได้มาจาก o/p ของวงจร D/A Converter (DAC0800) ซึ่ง DAC0800 จะเป็นตัวแปลงสัญญาณจาก 8031 มาเป็น analog เนื่องจากเอาท์พุต ของ D/A Converter เป็นกระแสจะถูกเปลี่ยนเป็น Voltage โดย A1 โดย V_{in} นี้จะมากขึ้นหรือน้อย ลงตามค่าของ DB0--DB7 ที่ส่งมาจาก 8031 microcontroller โดยค่าที่ส่งมานี้จะมีค่าตั้ง แต่ 00H-FFH โดยที่ค่า 80H จะให้ค่าของ V_{in} เป็น 0 V และ 00H จะให้ค่าสูงสุดทางลบ (V_{in} เป็นบวก) และ FFH จะให้ค่าสูงสุดทางบวก (V_{in} เป็นลบ) เนื่องจาก A1 มีอัตราขยายเป็น ลบ โดย V_{in} สูงสุดและต่ำสุดจะอยู่ในช่วง +10V ถึง -10V

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร VSCM นี้ o/p จะเป็น voltage (V_{out}) เพื่อป้อนให้ V_{ce} หรือป้อนให้ไดโอดเนื่องจาก R_2 มีค่า 10K ดังนั้นเราจะสามารถกำหนด V_{out} ได้ด้วยค่า R_1 ซึ่งเราจะใช้ analog switch (LF15308) เพื่อจะเลือก ค่าของ R_1 โดยจะส่ง AV1-CV1 มาจาก 8031 (Latch 74HCT373)

ส่วนของ I_{out} หรือ I_c จะส่งกลับไปยังคอมพิวเตอร์โดยผ่านทาง $V_{monitor}$ โดยจะมี R_c เป็นตัวกำหนด และก่อนเข้า A/D Converter (AD7574) จะมี R_m เป็นตัวกำหนด range โดยละเอียดอีกครั้งหนึ่ง AD7574 จะทำหน้าที่แปลงสัญญาณ analog ให้เป็น digital ซึ่งจะส่งกลับไปยัง 8031 แล้วส่งต่อไปยังคอมพิวเตอร์เพื่อแสดงผล

出力電圧			検出電流					
R_1SW	R_1	VS	R_cSW	R_c	CM	倍率	R_mSW	R_m
1	10k	±10V	1	10Ω	±100mA	1	1	10k
2	20k	± 5V	2	20Ω	± 50mA	1/2	2	20k
3	50k	± 2V	3	50Ω	± 20mA	1/5	3	50k
4	100k	± 1V	4	100Ω	± 10mA	1/10	4	100k
5	200k	±0.5V	5	200Ω	± 5mA	1/20	5	200K
6	500K	±0.2V	6	500Ω	± 2mA	1/50	6	500K
7	1M	±0.1V	7	1k	± 1mA	1/100	7	1M
8	2M	±0.05V	8	2k	±500μA	1/200	8	2M

ตารางที่ 5.2 ตารางแสดงค่าเอาท์พุตของ VSCM

ตัวอย่างจากตาราง

ถ้าเราเลือก R_1 50K R_c 20 โอห์ม และ R_m 10K เราจะได้ V_{out} ต่ำสุดและสูงสุดจากการป้อนค่า DB0-DB7 จะเท่ากับ -2V และ +2V V_{out} ต่ำสุดและสูงสุดเท่าไรขึ้นอยู่กับเราเลือก R_1 ค่าเท่าใด จากตัวอย่าง I_c (I_{out}) จะอยู่ในช่วง -50 mV ถึง +50 mV (เราเลือก R_m ที่ 1) แต่ถ้าเราเลือก R_m ไปที่ตำแหน่ง 100K range ของ I_c จะถูกทำให้แคบลงคือมีค่าละเอียดขึ้นเป็น +50mA และ -50mA ซึ่งเท่ากับ +5mA และ -5mA ค่า min และ max จะเปลี่ยนไป

10 10

เป็นจำนวนค่าของ range เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการเลือก $R_1 R_m R_c$

ส่วนของการเลือก $R_1 R_m R_c$ คือการเลือก Range ของเอาท์พุตการเลือก R_1 R_m สามารถใช้อนาล็อกสวิทช์ LF13508 เลือกได้ โดยจะมีค่า AV1-CV1 หรือ AC1-CC1 และ AV3-CV3 หรือ AC3-CC3 ที่ส่งมาจากคอมพิวเตอร์ เพื่อเลือกค่า R ตามต้องการ ส่วนของ R_c เนื่องจากกระแสที่ส่วนของ R_c มีค่ามาก และ LF13508 ทนกระแสสวิทช์ได้เพียง 10 mA จึงต้องใช้ Relay เป็นสวิทช์ โดยต่อร่วมกับ 74HCT138 ดังรูปที่ 5.1 และ 5.3 โดยจะมีค่า AV2-CV2 หรือ AC2-CC2 ที่ส่งมาจากคอมพิวเตอร์เพื่อเลือก R (เลือก range) ที่ต้องการ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

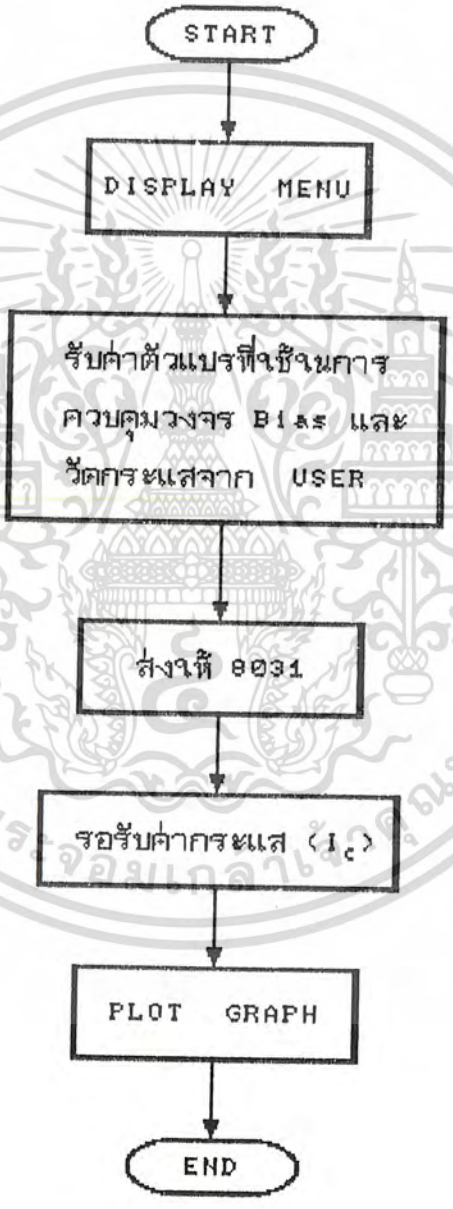
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

Software

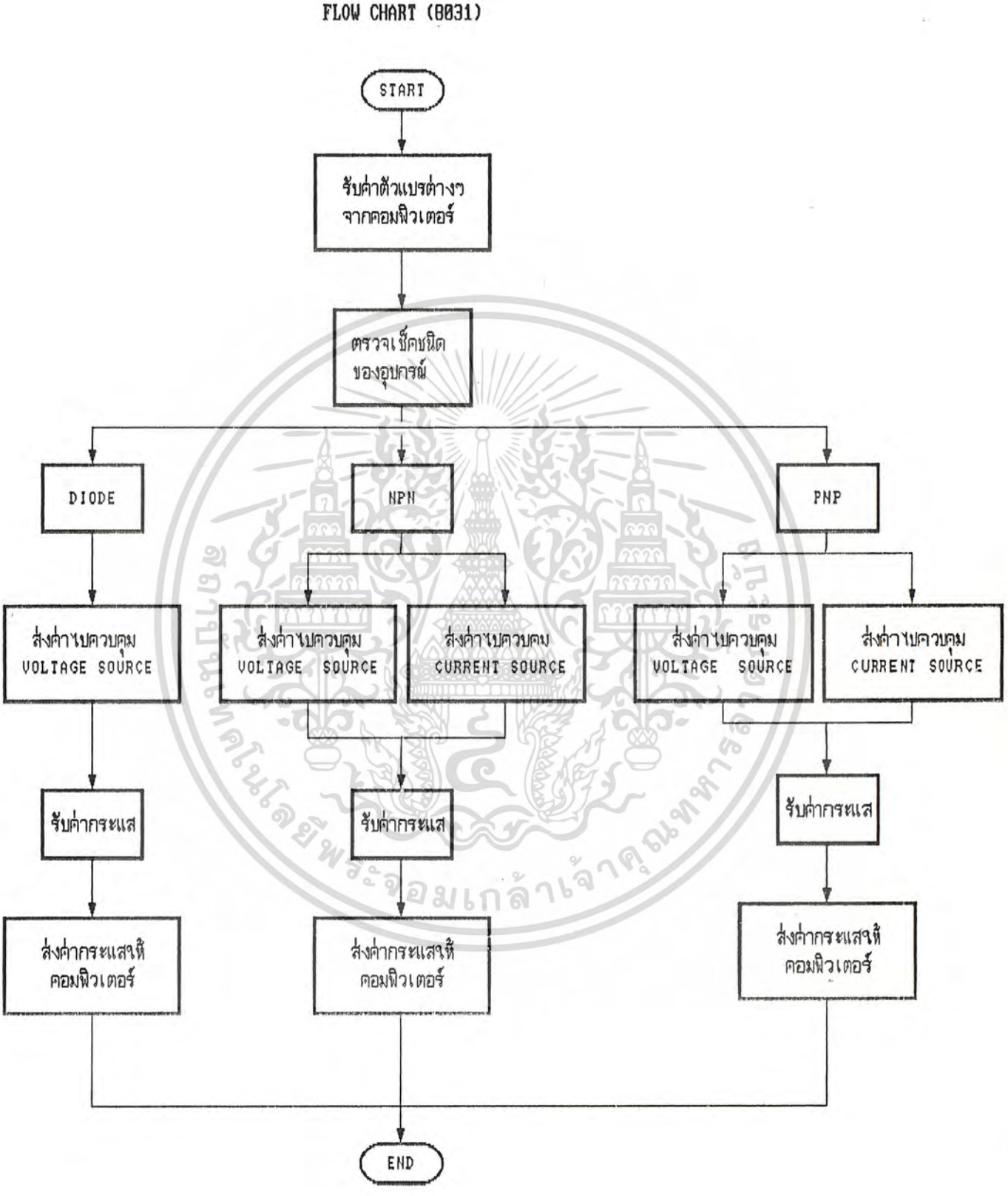
การทำงานของคอมพิวเตอร์และ 8031 microcontroller สามารถแสดงได้ดัง รูป 6.1 และ 6.2

FLOW CHART (PC)



รูปที่ 6.1 การทำงานของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 การทำงานของ 8031 microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของคอมพิวเตอร์ร่วมกับ 8031 microcontroller

1. คอมพิวเตอร์จะรับค่าของตัวแปรที่ต้องใช้ในการควบคุมให้ Hardware ทำงานได้ถูกต้อง ซึ่งค่าเหล่านั้นได้แก่

1. ชนิดของ device

2. ค่าของ Voltage Source หรือ Current Source ซึ่งค่าของ Voltage Source หรือ Current Source นี้จะนำไปเป็นเงื่อนไขในการกำหนดค่าของ input ที่จะป้อนให้แก่ LF13508 ซึ่งเป็นตัวกำหนด Range ของ Voltage Source และ Current Source

3. ค่าของ Display Current และ Step Current (I_b) ในกรณีเป็นทรานซิสเตอร์ ซึ่งค่านี้จะนำไปกำหนด I/P ของ LF13508 ชุดที่ควบคุม

4. ค่าของ serial port ที่ใช้งานอยู่

2. คอมพิวเตอร์จะส่งค่าของตัวแปรในข้อ 1. ไปยัง 8031 แล้ว 8031 จะนำค่าที่รับได้ไปประมวลผลเพื่อสร้างข้อมูลส่งไปควบคุมชุดวัด เมื่อ 8031 ส่งข้อมูลไปควบคุมชุดไบอัสก็จะต้องรับค่าของกระแส I_c กลับมาเก็บไว้ใน Memory ส่วน RAM ของ 8031 ไว้ก่อน เมื่อได้ครบตามจำนวนที่กำหนด จึงจะส่งกลับมาให้คอมพิวเตอร์ภายในครั้งเดียว

3. คอมพิวเตอร์รับค่าที่ส่งมาจาก 8031 นำมาประมวลผลเพื่อนำไป display เป็น I-V Characteristic curve ได้

4. การรับส่งระหว่างคอมพิวเตอร์และ 8031 จะต้องตั้งค่าของ Baud rate start bit และ stop bit ให้ตรงกัน ในโปรแกรมนี้ใช้ค่า Baud Rate 9600 start bit 0 และ stop bit 1 non parity bit

5. 8031 มี mode ของการส่งแบบ serial ให้เลือกใช้ 3 mode ซึ่งในเครื่องสร้างกราฟคุณสมบัติของไดโอดและทรานซิสเตอร์นี้ใช้ Mode 1 ซึ่งตั้งค่าต่างๆไว้ตรงกับในข้อ 4

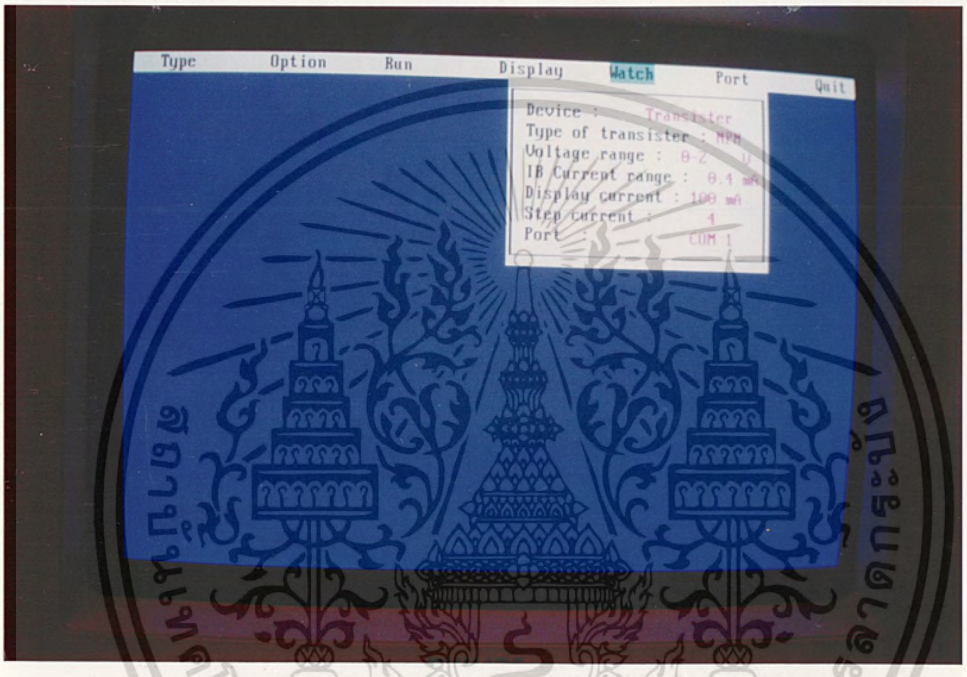
6. ก่อนทำการรับส่งข้อมูลแต่ละชุดจะต้องมีการ Check ความพร้อมของฝ่ายตรงกันข้ามก่อน เช่น ถ้าคอมพิวเตอร์ต้องการส่งข้อมูลไปยัง 8031 คอมพิวเตอร์จะต้องส่งข้อมูล byte หนึ่งไปตรวจสอบความพร้อมของ 8031 ก่อน เมื่อ 8031 พร้อมที่จะรับก็จะส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอีก byte หนึ่งตอบกลับมาแสดงความพร้อมหลังจากนั้นจึงเริ่มทำการรับส่งข้อมูลกัน เพื่อป้องกันข้อมูลสูญหายขณะรับส่ง

ส่วนของ Menu

หน้าจอของ Menu แสดงได้ดังรูป



รูปที่ 6.3 Menu

การใช้ Menu

1. ป้อนชนิดของ device ว่าเป็นไดโอดหรือทรานซิสเตอร์

2. ป้อน ขนาดของ Voltage Source, Current Source และ Range ของ Display Current

3. ป้อน serial port ที่ใช้อยู่ (เนื่องจากเครื่องสร้างกราฟคุณสมบัติไดโอดและทรานซิสเตอร์นี้ติดต่อกับ computer ทาง serial port ใช้วิธีส่งข้อมูลแบบอนุกรม โดยใช้มาตรฐาน RS232-C

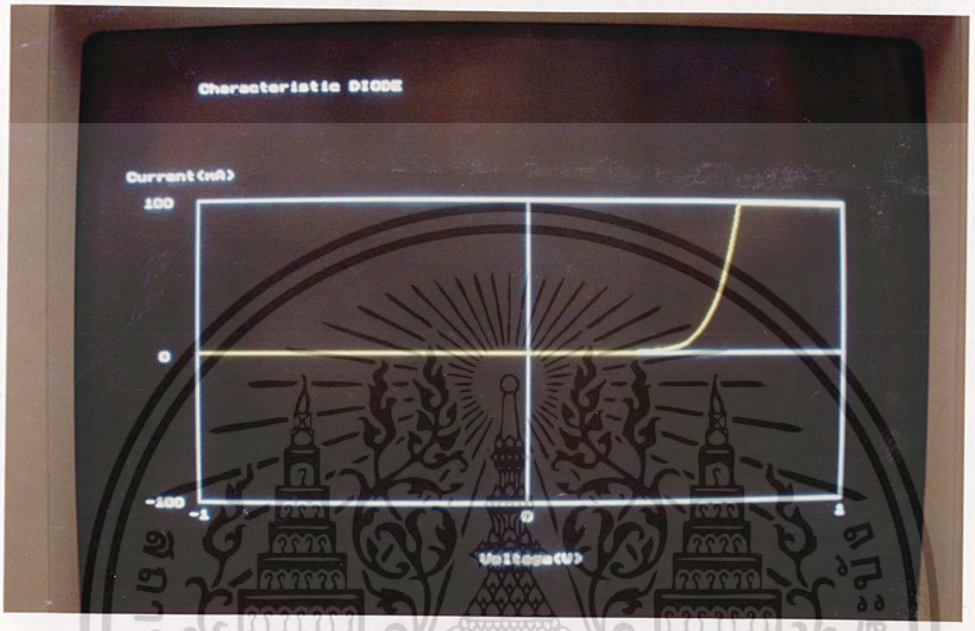
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทดลอง

ไดโอด

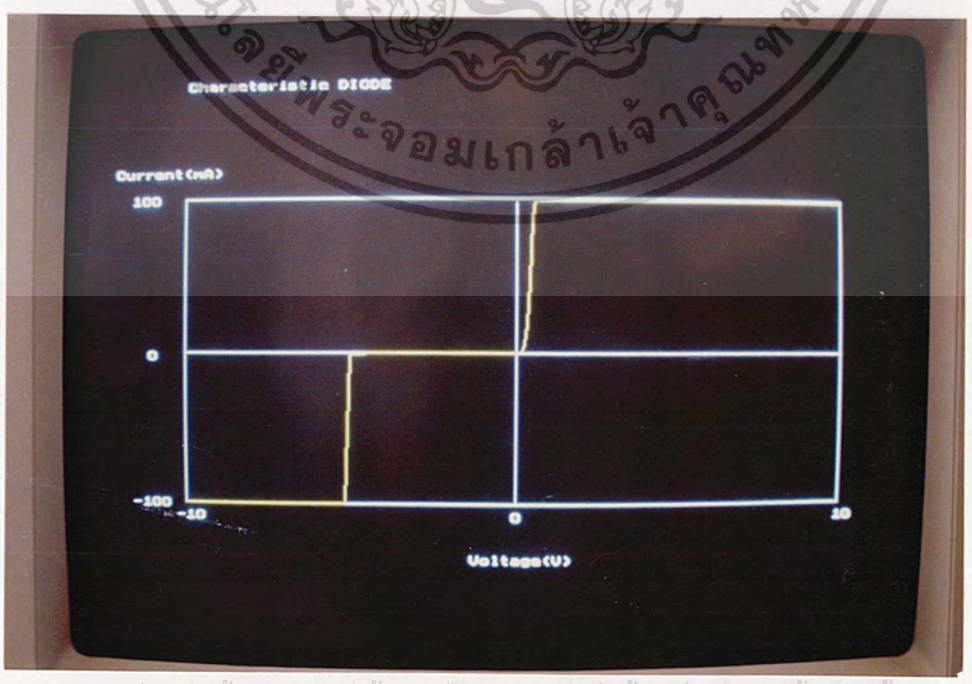
ได้ทำการทดลองโดยใช้ไดโอดเบอร์ 1N4148 ผลการทดลองแสดงดังรูป 7.1



รูปที่ 7.1

จากรูป Forward Voltage ประมาณ 0.6 V

ได้ทำการทดลองโดยใช้ไดโอดเบอร์ 1N5242B ผลการทดลองแสดงดังรูป 7.2



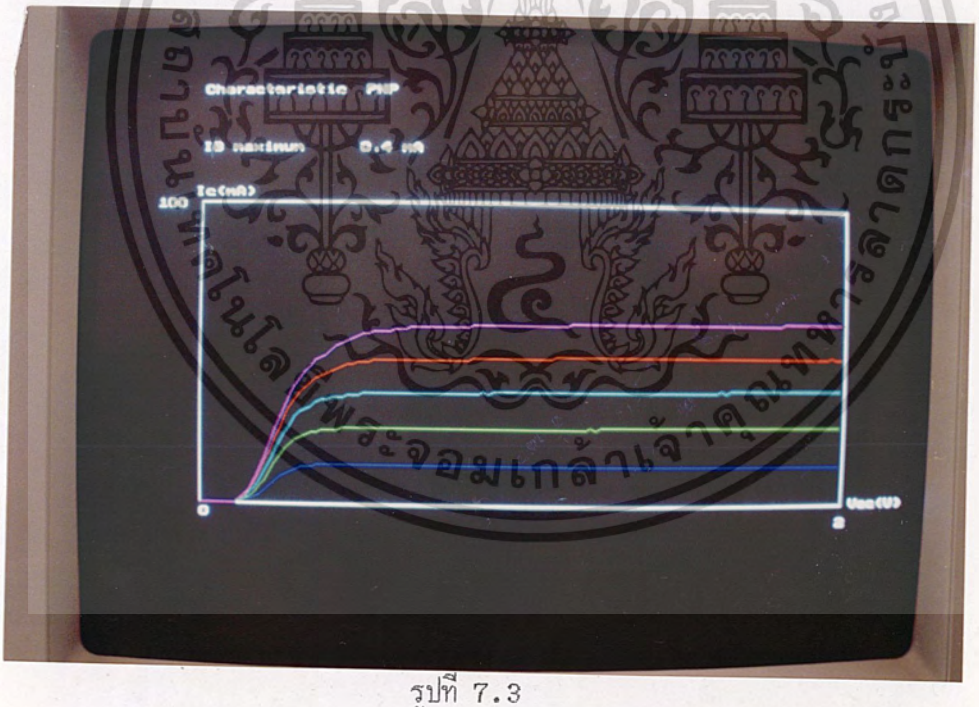
รูปที่ 7.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็น Forward Voltage (V_F) ประมาณ 0.6 V และเนื่องจากเป็นซีเนอร์ไดโอดจึงสามารถเห็น Breakdown Voltage (V_Z) ซึ่งประมาณ 6 V ได้อีกด้วย

ทรานซิสเตอร์

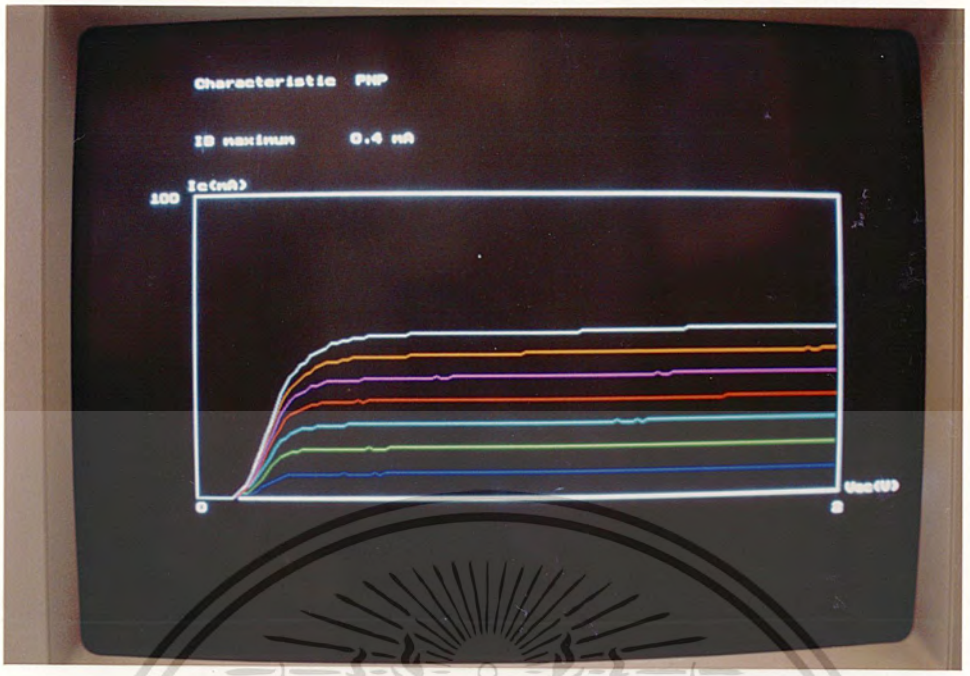
ได้ทดลองโดยใช้ทรานซิสเตอร์ PNP เบอร์ 2SA696A โดยกำหนด $I_{b\ maximum}$ เท่ากับ 0.4 mA, step ของ I_b 5 step, range ของ I_c เท่ากับ 100 mA และ V_{CE} เท่ากับ 2 V ผลการทดลองเป็นดังรูป 7.3



รูปที่ 7.3

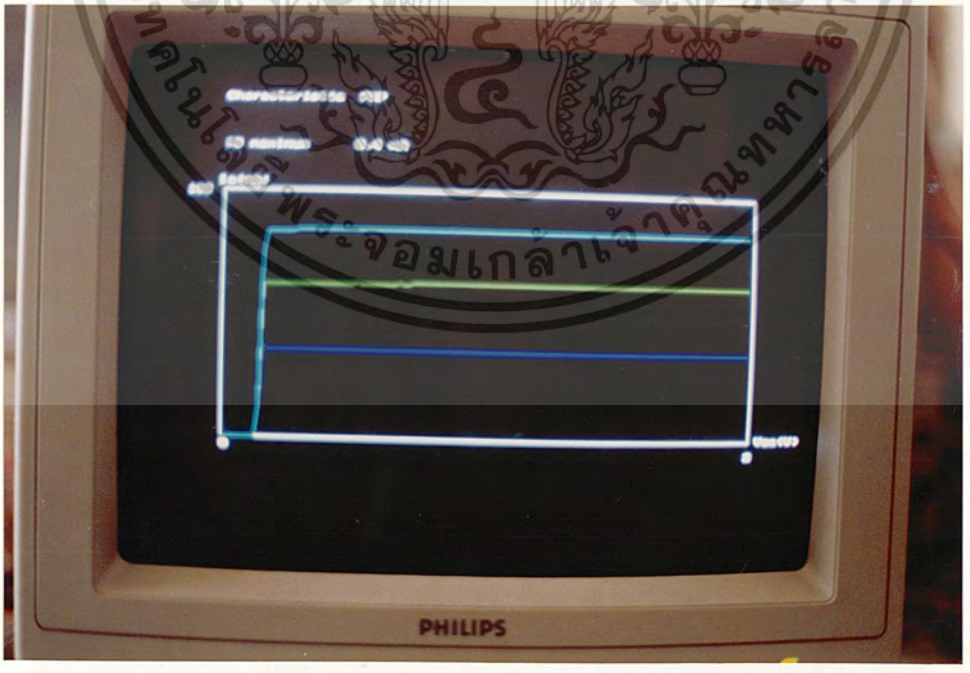
จากนั้นได้ทำการเปลี่ยน step ของ I_b เป็น 7 step ได้ผลการทดลองเป็นดังรูป 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4

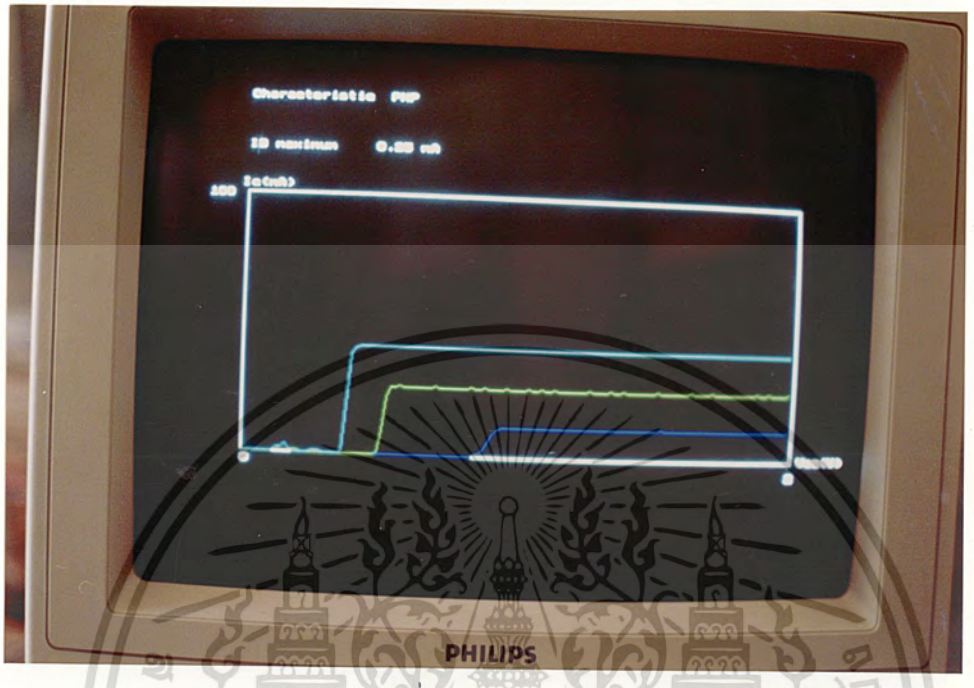
ใช้ทรานซิสเตอร์เบอร์ 2SA1015 โดยกำหนด I_b maximum เท่ากับ 0.4 mA, step ของ I_b 5 step, range ของ I_c เท่ากับ 100 mA และ V_{CE} เท่ากับ 2 V ผลการทดลองเป็นดังรูป 7.5



รูปที่ 7.5

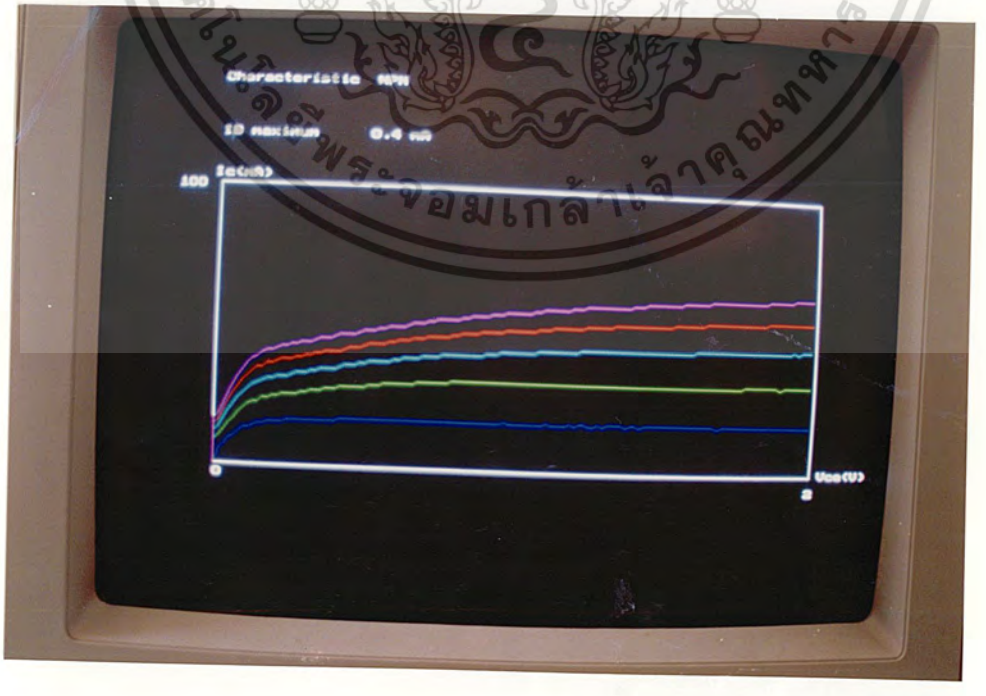
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากผลการทดลองออกมาไม่ดีนักจึงได้เปลี่ยน $I_{b\ maximum}$ เป็น 0.25 mA ได้ผลการทดลองดังรูป



รูปที่ 7.6

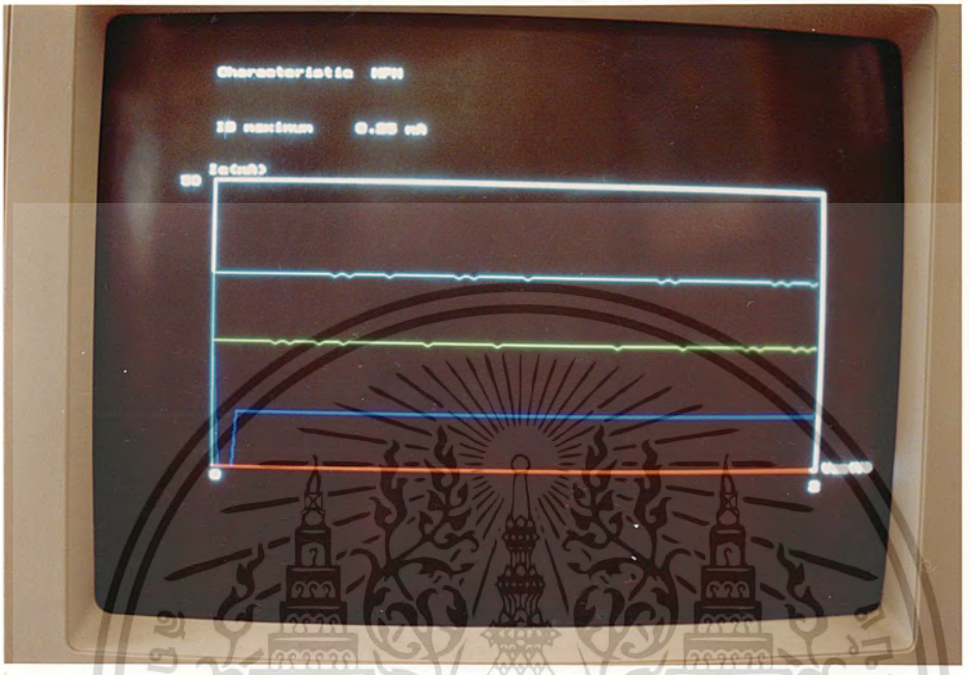
ทดลองใช้ทรานซิสเตอร์ NPN เบอร์ 2SC1815 ใช้ข้อกำหนดต่างๆ เหมือนทรานซิสเตอร์ PNP ผลการทดลองเป็นดังรูป 7.7



รูปที่ 7.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลองเปลี่ยน $I_{b\ maximum}$ เป็น 0.25 mA ผลการทดลองได้ดังรูป 7.8



รูปที่ 7.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปและวิจารณ์

ข้อดีของเครื่องตรวจสอบคุณสมบัติของไดโอดและทรานซิสเตอร์

1. สามารถใช้งานกับไดโอดและทรานซิสเตอร์ NPN และ PNP ได้

2. สามารถเลือก range ของ I_c และ V_{CE} ได้หลายค่า และสามารถเลือก step ของ I_b ได้ถึง 7 step

3. สามารถปรับเปลี่ยนช่วงของ range ของ V_{CE} และ I_c เพื่อจะดูค่าที่จุดต่างๆ ได้ละเอียดขึ้นได้

4. เมื่อนำเครื่องนี้ไปใช้กับไดโอดที่มี breakdown voltage ไม่เกิน 10 V จะเห็นจุด breakdown ได้อย่างชัดเจน และยังสามารถเห็น leakage curreent ได้ด้วย

ข้อเสียของเครื่องตรวจสอบคุณสมบัติทรานซิสเตอร์และไดโอด

1. จะต้องป้อนค่าของกระแสและโวลต์เตจที่เหมาะสม เพื่อให้ทรานซิสเตอร์ทำงาน จึงจะสามารถวัดกระแสของ I_c ได้อย่างถูกต้อง

2. เนื่องจากการสร้างวงจรใช้การวายแลบทำให้เกิดสัญญาณรบกวนในสายข้อมูลได้ง่าย

3. ในการใช้งานต้องทราบขาของทรานซิสเตอร์เสียก่อน

4. เครื่องนี้ใช้ได้เฉพาะ small signal transister เนื่องจากมี limit ของ VSCM ที่ +10 V และ -10V และ CSVM ที่ +100 mA -100 mA

5. เมื่อใช้กับทรานซิสเตอร์ NPN ซึ่งมีค่า gain ของการขยายกระแสสูงกว่าชนิด PNP มากจะทำให้เกิดการ oscillate

แนวทางแก้ไขและปรับปรุง

1. ควรทำให้วงจรไบอัสและวัดกระแสกลับมี Voltage Source และ Current Source มากกว่านี้ เพื่อให้ใช้ได้กับทรานซิสเตอร์หลายชนิดและยังทำให้สามารถเห็น breakdown voltage ของไดโอดได้มากขึ้น

2. ควรทำการออกแบบลายปรินซ์เพื่อไม่ให้มีปัญหาเรื่องสายวายแลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลงได้ด้วยดีผู้จัดทำต้องขอขอบพระคุณ รศ.ดร.มนัส สังวรศิลป์ ที่กรุณาให้คำปรึกษาและใช้อุปกรณ์ภายในห้องทดลอง และขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ทุกคนที่ให้คำปรึกษาและคอยให้กำลังใจ ขอขอบคุณชุมนุมอิเลคทรอนิกส์ที่เอื้อเฟื้ออุปกรณ์ รวมทั้งผู้ที่มีส่วนช่วยให้ปริญญานิพนธ์นี้สำเร็จไปด้วยดี

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "stdio.h"
#include "dos.h"
#include "stdlib.h"
#include "alloc.h"
#include "graphics.h"
#include "ctype.h"
#include "bios.h"

#define BORDER      1
#define ESC         27
#define MAX_FRAME   64
#define REV_VID     0x70
#define NORM_VID    7
#define THRE        0x20
#define DR          0x01
#define TX          0
#define RX          0
#define LSR         5

void save_video(),restore_video(),pd_driver(),graph();
void write_video(),write_string(),display_menuxy(),write_char();
void display_menu(),display_menux(),draw_border(),draw_borderx();
void sport(),port_init(),dis_diode(),dis_npn(),dis_pnp(),initial();

int p,cc;
int port;
int j;
unsigned long int i;
int c[1200];
int xmax,xmin,ymax,ymin;
char far *vid_mem;

struct menu_frame{
int startx,starty,endx,endy;
unsigned char *p;
char **menu;
char *keys;
int border,count;


int active;
} frame[MAX_FRAME];

char *device[] =
{
"Transister",
"Diode",
};

char *option[] =
{
"Voltage",
"Current",
};

char *type[] =
{
"NPN",
"PNP",
};

char *display[] = {
"V-I characteristic",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"Gain",
};

char *main_menu[] =
{
"Type",
"Option",
"Run",
"Display ",
"Watch",
"Port",
"Quit",
};



char *current[] =
{"100 mA","50 mA","25 mA","20 mA","10 mA","5 mA","4 mA","2.5 mA",
"2 mA" ,"1 mA","0.5 mA","0.4 mA","0.25 mA","0.2 mA","0.1 mA","0.05 mA",
"0.04 mA","0.025 mA","0.02 mA","0.01 mA","0.005 mA","0.0025 mA",
};
char *range_I[] =
{"100 mA","50 mA","25 mA","20 mA","10 mA","5 mA","4 mA","2.5 mA",
"2 mA" ,"1 mA","0.5 mA","0.4 mA","0.25 mA","0.2 mA","0.1 mA","0.05 mA",
"0.04 mA","0.025 mA","0.02 mA","0.01 mA","0.005 mA","0.0025 mA",
};



char *voltage[] =
{"0-10   V","0-5    V","0-2    V","0-1    V","0-0.5  V","0-0.2  V","0-0.1  V",
"0-0.05 V",};


char *step_I[] =
{"1","2","3","4","5","6","7","8",};

char *display_I[] = {"Range of current","Step of current",};

char *watch[] = {"Device :    " ,"Type of transister :     ","Voltage range : ",
"IB Current range :  ","Display current : ","Step current : ",
 "Port  : "};
char *port_select[] = {"COM 1","COM 2",};


int moc(c,d)
int c,d;
{
int e;
e = c/d;
return e;
}

bkcolor(n)
int n;
{
        int i,j;

        clrscr();
        for(i=0;i<81;i++)
        {       for(j=0;j<26;j++)
                {
                        textattr(n);
                        gotoxy(i,j);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        putch(0);
                }
        }
}

main()
{
        clrscr();
        bkcolor(0x11);
        gotoxy(0,0);

        make_menux(0,main_menu,"tordwpq",7,3,0,BORDER);
        make_menu(1,device,"td",2,2,1,BORDER);
        make_menu(2,type,"np",2,14,2,BORDER);
        make_menu(3,option,"vc",2,13,1,BORDER);
        make_menu(4,display,"vg",2,10,9,BORDER);
        make_menu(5,voltage,"",8,22,1,BORDER);
        make_menuxy(6,current,22,4,5,BORDER);
        make_menu(7,display_I,"rs",2,32,1,BORDER);
        make_menu(8,step_I,"",8,50,3,BORDER);
        make_menuxy(9,range_I,22,4,5,BORDER);
        make_menu(10,watch,"",7,40,1,BORDER);
        make_menu(11,port_select,"12",2,62,1,BORDER);
        pd_driver();


}

void dis_npn(s,e)
unsigned long int s,e;
{

      int  ch;
      unsigned  int k;
      moveto(xmin,ymin);
      k=0;
      for(i=s;i<e;i++)
      {
      plot_npn(k,c[i]);
      k=k+1;
      }

}


tr_dis(typ, d,volt,curr,cur)
int typ,d,volt,curr,cur;
{
        unsigned long int x;
        int n,ch;
        char *a,*tp,*vo,*cu,*vmax,*imax,*ibmax,*IB;
        while((cc=rport(port))!=-1) sport(port,-1);
        sport(port,-2);



        x=(d+1)*128;
        for(i=0;i<x;i++)
        {
             cc = rport(port);
             c[i] = cc;
        }
        if(typ==0)
        {
             tp = "Characteristic  NPN" ;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        vo = "Vce(V)";
    }

    if(typ==1)
    {
        tp = "Characteristic  PNP";
        vo = "Vec(V)";
    }
    initial();
    setbkcolor(12);
    a = "0";

    IB="IB maximum";
    cu = "Ic(mA)";

    switch(cur)
    {
            case 0 :  ibmax="   100 mA"; break;
            case 1 :  ibmax="    50 mA"; break;
            case 2 :  ibmax="    25 mA"; break;
            case 3 :  ibmax="    20 mA"; break;
            case 4 :  ibmax="    10 mA"; break;
            case 5 :  ibmax="     5 mA"; break;
            case 6 :  ibmax="     4 mA"; break;
            case 7 :  ibmax="   2.5 mA"; break;
            case 8 :  ibmax="     2 mA"; break;
            case 9 :  ibmax="     1 mA"; break;
            case 10:  ibmax="   0.5 mA"; break;
            case 11:  ibmax="   0.4 mA"; break;
            case 12:  ibmax="  0.25 mA"; break;
            case 13:  ibmax="   0.2 mA"; break;
            case 14:  ibmax="   0.1 mA"; break;
            case 15:  ibmax="  0.05 mA"; break;
            case 16:  ibmax="  0.04 mA"; break;
            case 17:  ibmax=" 0.025 mA"; break;
            case 18:  ibmax="  0.02 mA"; break;
            case 19:  ibmax="  0.01 mA"; break;
            case 20:  ibmax=" 0.005 mA"; break;
            case 21:  ibmax="0.0025 mA"; break;
    }
    switch(volt)
    {
            case 0:   vmax = "10 "; break;
            case 1:   vmax = "5  "; break;
            case 2:   vmax = "2  "; break;
            case 3:   vmax = "1  "; break;
            case 4:   vmax = "0.5"; break;
            case 5:   vmax = "0.2"; break;
            case 6:   vmax = "0.1"; break;
            case 7:   vmax = "0.05"; break;
    }
    switch(curr)
    {
            case 0 :  imax="   100 "; break;
            case 1 :  imax="    50 ";  break;
            case 2 :  imax="    25 "; break;
            case 3 :  imax="    20 "; break;
            case 4 :  imax="    10 "; break;
            case 5 :  imax="     5 "; break;
            case 6 :  imax="     4 "; break;
            case 7 :  imax="   2.5 "; break;
            case 8 :  imax="     2 "; break;
            case 9 :  imax="     1 "; break;
            case 10:  imax="   0.5 "; break;
            case 11:  imax="   0.4 "; break;
            case 12:  imax="  0.25 "; break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 13:  imax="   0.2 "; break;
                case 14:  imax="   0.1 "; break;
                case 15:  imax="  0.05 "; break;
                case 16:  imax="  0.04 "; break;
                case 17:  imax=" 0.025 "; break;
                case 18:  imax="  0.02 "; break;
                case 19:  imax="  0.01 "; break;
                case 20:  imax=" 0.005 "; break;
                case 21:  imax="0.00251"; break;
        }

        xmin = 70;
        xmax=xmin+(2*256);
        ymax=115;
        ymin=ymax+256;

        outtextxy(xmin,ymin+5,a);
        outtextxy(xmin,ymax-100,tp);
        outtextxy(xmax+8,ymin-5,vo);
        outtextxy(xmin-5,ymax-12,cu);
        outtextxy(xmax-3,ymin+12,vmax);
        outtextxy(xmin-60,ymax-2,imax);
        outtextxy(xmin,ymax-50,IB);
        outtextxy(xmin+100,ymax-50,ibmax);
        line(xmin,ymin,xmax,ymin);
        line(xmin,ymax,xmax,ymax);
        line(xmax,ymin,xmax,ymax);
        line(xmin,ymin,xmin,ymax);

}
npn(d,vmax,imax,cur)
int d,vmax,imax,cur;
{
        unsigned long int l,m;
        int k;
        int n;

        tr_dis(0,d,vmax,imax,cur);
        l=0;
        n=1;
        for(k=0;k<d+1;k++)
        {
              setcolor(n);
              n=n+1;
              m=l+128;
              dis_npn(l,m);
              l=m;
        }
        getch();
        closegraph();

}
plot_pnp(unsigned int xi,unsigned int yi)
{
        int xdi,ydi;

        xdi =  xmin + ((xmax-xmin)*xi)/128;
        ydi =  ymax + ((ymin-ymax)*yi)/128;
        if(ydi>ymin) ydi=ymin;
        lineto(xdi,ydi);
}

plot_npn(unsigned  int xi,unsigned int yi)
{
        int xdi,ydi;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        xdi =  xmin + ((xmax-xmin)*xi)/128;
        ydi =  ymax + ((ymax-ymin)*yi)/128;
        if(ydi<ymax+1) ydi=ymin;
        if(ydi>ymin) ydi=ymin;
        lineto(xdi,ydi);
}

void dis_pnp(s,e)
unsigned long int s,e;
{

        int  ch;
        unsigned  int k;
        moveto(xmin,ymin);
        k=0;
        for(i=s;i<e;i++)
        {
        plot_pnp(k,c[i]);
        k=k+1;
        }

}


pnp(d,vmax,imax,cur)
int d,vmax,imax,cur;
{
        unsigned long int l,m;
        int n,k;

        tr_dis(1,d,vmax,imax,cur);
        l=0;
        n=1;
        for(k=0;k<d+1;k++)
        {
              setcolor(n);
              n=n+1;
              m=l+128;
              dis_pnp(l,m);
              l=m;
        }
        getch();
        closegraph();
}


diode(cur,vol)
int cur,vol;
{

        while((cc=rport(port))!=-1)
        sport(port,3);
        sport(port,-2);

        for(i=0;i<256;i++)
        {
        cc = rport(port);
        c[i] = cc;

        }
        dis_diode(cur,vol);
        getch();
        closegraph();
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
plot_diode(unsigned long int xi,  int yi)
{
        int xdi,ydi;

        xdi =  xmin + ((xmax-xmin)*xi)/256;
        ydi =  (ymax+ymin)/2-((ymin-ymax)*(yi-128)/256);
        lineto(xdi,ydi);
}

void initial()
{
        int g_driver,g_mode;

        detectgraph(&g_driver,&g_mode);
        initgraph(&g_driver,&g_mode,"c:\\");
}

void dis_diode(curr,vol)
int curr,vol;


{
int ch;
char *a,*tp,*vo,*cu,*imin,*imax,*vmin,*vmax;

switch(curr)
        {
                case 0 : imin= "  -100"; imax="  100"; break;
                case 1 : imin= "   -50"; imax="   50"; break;
                case 2 : imin= "   -25"; imax="   25"; break;
                case 3 : imin= "   -20"; imax="   20"; break;
                case 4 : imin= "   -10"; imax="   10"; break;
                case 5 : imin= "    -5"; imax="    5"; break;
                case 6 : imin= "    -4"; imax="    4"; break;
                case 7 : imin= "  -2.5"; imax="  2.5"; break;
                case 8 : imin= "  -2.0"; imax="  2.0"; break;
                case 9 : imin= "    -1"; imax="    1"; break;
                case 10: imin= "  -0.5"; imax="  0.5"; break;
                case 11: imin= "  -0.4"; imax="  0.4"; break;
                case 12: imin= " -0.25"; imax=" 0.25"; break;
                case 13: imin= "  -0.2"; imax="  0.2"; break;
                case 14: imin= "  -0.1"; imax="  0.1"; break;
                case 15: imin= " -0.05"; imax=" 0.05"; break;
                case 16: imin= " -0.04"; imax=" 0.04"; break;
                case 17: imin= " -0.025";imax="0.025"; break;
                case 18: imin= " -0.02"; imax=" 0.02"; break;
                case 19: imin= " -0.001";imax="0.001"; break;
                case 20: imin= " -0.002";imax="0.002"; break;
                case 21: imin= "-0.0025";imax="0.0025"; break;
        }
        switch(vol)
        {
                case 0:  vmin = "-10"; vmax = "10 "; break;
                case 1:  vmin = "-5"; vmax = "5  "; break;
                case 2:  vmin = "-2"; vmax = "2  "; break;
                case 3:  vmin = "-1"; vmax = "1  "; break;
                case 4:  vmin = "-0.5"; vmax = "0.5"; break;
                case 5:  vmin = "-0.2"; vmax = "0.2"; break;
                case 6:  vmin = "-0.1"; vmax = "0.1"; break;
                case 7:  vmin = "-0.05"; vmax = "0.05"; break;
        }
        a = "0";
        tp = "Characteristic DIODE" ;
        vo = "Voltage(V)";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cu = "Current(mA)";

        xmin=70;
        xmax=xmin+(2*256);
        ymax=115;
        ymin=ymax+256;

        initial();
        setbkcolor(12);

        outtextxy((xmin+xmax)/2-3,ymin+8,a);
        outtextxy(xmin-30,(ymax+ymin)/2,a);
        outtextxy(xmin,ymax-100,tp);
        outtextxy((xmax+xmin)/2-35,ymin+45,vo);
        outtextxy(xmin-58,ymax-25,cu);
        outtextxy(xmin-8,ymin+8,vmin);
        outtextxy(xmax-3,ymin+8,vmax);
        outtextxy(xmin-60,ymin-2,imin);
        outtextxy(xmin-60,ymax-2,imax);

        line(xmin,(ymin+ymax)/2,xmax,(ymin+ymax)/2);
        line((xmin+xmax)/2,ymax,(xmin+xmax)/2,ymin);
        line(xmin,ymin,xmax,ymin);
        line(xmin,ymax,xmax,ymax);
        line(xmax,ymin,xmax,ymax);
        line(xmin,ymin,xmin,ymax);
        j =  (ymax+ymin)/2-((ymin-ymax)*(c[0]-128)/256);
        setcolor(14);
        moveto(xmin,j);

        for(i=0;i<256;i++)
        {
        plot_diode(i,c[i]);
        }
}


void pd_driver()
{
        register int device;
        int mai,typ,typ1,typ2,opt,curr,disp,vr1,vrc,vrm,cr1,crc;
        int c,assign,stp=100,dis_curr=100,volt=100;
        int step;
        int c0,c1,cc,c2;
        do
        {
        mai=pulldownx(0);
        {
                typ=mai;
                write_string(frame[0].startx+(mai*12),frame[0].starty,
                frame[0].menu[mai],0x30);
                switch(mai)
                {
                case 0: /*type*/
                while((typ1=pulldown(1)) != -1)
                {
                        write_string(frame[1].startx+2,frame[1].starty+1,
                        frame[1].menu[typ1],0x30);
                        device=typ1;
                        if(typ1==0) /*transister*/
                        {
                        typ2=pulldown(2);/*npn or pnp*/
                        restore_video(2);
                        }
                        if(typ1==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
}
            restore_video(1);
            break;
case 1: /*option*/

while((opt=pulldown(3)) != -1)
{
write_string(frame[3].startx+2,frame[3].starty+opt+1,
frame[3].menu[opt],0x30);

            if(opt==0) /*voltage*/
            {
            volt=pulldown(5);/*rang of voltage*/
            restore_video(5);
            }
            if(device==0)
            if(opt==1) /*current*/
            {
            curr=pulldownxy(6);/*range of current*/
            restore_video(6);
            }
}
            restore_video(3);
            break;

case 2: /*run*/
            write_string(frame[0].startx+(mai*12),
            frame[0].starty,frame[0].menu[mai],0x70);

            if((((device==0)&&((stp!=-1)&&(stp!=100)&&
            (typ2!=-1)))||(device==1))&&((volt!=-1)&&
            (volt!=100))&&((dis_curr!=-1)&&(dis_curr!=100)))

            {       vr1 = volt;
                    switch(curr)
                    {
                    case 0: /*100 mA*/
                            cr1 = 0; crc = 0;
                            break;
                    case 1: /*50 mA*/
                            cr1 = 1; crc = 0;
                            break;
                    case 2: /*25 mA*/
                            cr1 = 1; crc = 1;
                            break;
                    case 3: /*20 mA*/
                            cr1 = 2; crc = 0;
                            break;
                    case 4: /*10 mA*/
                            cr1 = 3; crc = 0;
                            break;
                    case 5: /*5 mA*/
                            cr1 = 4; crc = 0;
                            break;
                    case 6: /*4 mA*/
                            cr1 = 2; crc = 2;
                            break;
                    case 7: /*2.5 mA*/
                            cr1 = 4; crc = 1;
                            break;
                    case 8: /*2 mA*/
                            cr1 = 5; crc = 0;
                            break;
                    case 9: /*1 mA*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                crl = 6; crc = 0;
                break;
        case 10: /*0.5 mA*/
                crl = 7; crc = 0;
                break;
        case 11: /*0.4 mA*/
                crl = 2; crc = 5;
                break;
        case 12: /*0.25 mA*/
                crl = 7; crc = 1;
                break;
        case 13: /*0.2 mA*/
                crl = 6; crc = 2;
                break;
        case 14: /*0.1 mA*/
                crl = 6; crc = 3;
                break;
        case 15: /*0.05 mA*/
                crl = 7; crc = 3;
                break;
        case 16: /*0.04 mA*/
                crl = 5; crc = 5;
                break;
        case 17: /*0.025 mA*/
                crl = 7; crc = 4;
                break;
        case 18: /*0.02 mA*/
                crl = 6; crc = 5;
                break;
        case 19: /*0.01 mA*/
                crl = 6; crc = 6;
                break;
        case 20: /*0.005 mA*/
                crl = 7; crc = 6;
                break;
        case 21: /*0.0025 mA*/
                crl = 7; crc = 7;
                break;
        }
switch(dis_curr)
{
        case 0: /*100 mA*/
                vrc = 0; vrm = 0;
                break;
        case 1: /*50 mA*/
                vrc = 1; vrm = 0;
                break;
        case 2: /*25 mA*/
                vrc = 1; vrm = 1;
                break;
        case 3: /*20 mA*/
                vrc = 2; vrm = 0;
                break;
        case 4: /*10 mA*/
                vrc = 3; vrm = 0;
                break;
        case 5: /*5 mA*/
                vrc = 4; vrm = 0;
                break;
        case 6: /*4 mA*/
                vrc = 2; vrm = 2;
                break;
        case 7: /*2.5 mA*/
                vrc = 4; vrm = 1;
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 8: /*2 mA*/
        vrc = 5; vrm = 0;
        break;
case 9: /*1 mA*/
        vrc = 6; vrm = 0;
        break;
case 10: /*0.5 mA*/
        vrc = 7; vrm = 0;
        break;
case 11: /*0.4 mA*/
        vrc = 2; vrm = 5;
        break;
case 12: /*0.25 mA*/
        vrc = 7; vrm = 1;
        break;
case 13: /*0.2 mA*/
        vrc = 6; vrm = 2;
        break;
case 14: /*0.1 mA*/
        vrc = 6; vrm = 3;
        break;
case 15: /*0.05 mA*/
        vrc = 7; vrm = 3;
        break;
case 16: /*0.04 mA*/
        vrc = 5; vrm = 5;
        break;
case 17: /*0.025 mA*/
        vrc = 7; vrm = 4;
        break;
case 18: /*0.02 mA*/
        vrc = 6; vrm = 5;
        break;
case 19: /*0.01 mA*/
        vrc = 6; vrm = 6;
        break;
case 20: /*0.005 mA*/
        vrc = 7; vrm = 6;
        break;
case 21: /*0.0025 mA*/
        vrc = 7; vrm = 7;
        break;
}

if((device==0)&&(typ2==0))  assign=1;/*npn*/
if((device==0)&&(typ2==1))  assign=2;/*pnp*/
if(device==1)               assign=0;/*diode*/
step = 128/(stp+1);

port_init(p,227);
if(p==0) port = 0x3f8;
else     port = 0x2f8;

do
{
        do
        {
        sport(port,-1);
        cc= rport(port);
        }while(cc!=-2);


        sport(port,assign);
        sport(port,cr1);
        sport(port,vrm);
        sport(port,vrc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sport(port,crc);
                    sport(port,vr1);
                    sport(port,stp+1);
                    sport(port,step);

                    c2= (assign+cr1+vrm+vrc+crc+vr1+(stp+1)+step)
                    c0=rport(port);
                    c1=rport(port);
                    cc = c0+(c1*256);
                }while(cc!=c2);
                sport(port,0);

                if((device==0)&&(typ2==0))
                npn(stp,volt,dis_curr,curr); /*npn*/
                if((device==0)&&(typ2==1))
                pnp(stp,volt,dis_curr,curr); /*pnp*/
                if(device==1)               diode(dis_curr,volt);
                }

                break;

case 3: /*display*/
        while((disp = pulldown(7)) != -1)
        {
        write_string(frame[7].startx+2,frame[7].starty+disp+1,
        frame[7].menu[disp],0x30);
                if(disp==0)
                { dis_curr = pulldownxy(9);
                  restore_video(9);
                }
                if(device==0)
                if(disp==1)
                {
                 stp = pulldown(8);
                 restore_video(8);
                }
        }
        restore_video(7);
        break;
case 4: /*watch*/

        if((((device==0)&&((stp!=-1)&&(stp!=100)&&
        (typ2!=-1)))||(device==1))&&((volt!=-1)&&
        (volt!=100))&&((dis_curr!=-1)&&(dis_curr!=100)))
        {
                if(!frame[10].active)
                save_video(10);
                do
                {
                        draw_border(10);
                        display_menu(10);

                        write_string(55,2,frame[1].menu[device],0x75);
                        if(device==0)
                        {
                        write_string(63,3,frame[2].menu[typ2],0x75);
                        write_string(62,5,frame[6].menu[curr],0x75);
                        if((stp!=-1)&&(stp!=100))
                        write_string(62,7,frame[8].menu[stp],0x75);
                        }
                        write_string(59,4,frame[5].menu[volt],0x75);
                        write_string(60,6,frame[9].menu[dis_curr],0x75);
                        write_string(60,8,frame[11].menu[p],0x75);
                }while (c=getch() != ESC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        restore_video(10);
                    }
                    break;

            case 5: p = pulldown(11);
                    restore_video(11);
                    break;

                }

        }
        }while (mai!=6);
        restore_video(0);
        bkcolor(0x07);
        clrscr();
}

void port_init(char c,unsigned char code)
{
        union REGS r;
        r.x.dx =  c;
        r.h.ah = 0;
        r.h.al = code;
        int86(0x14,&r,&r);
}

void sport(int port,char data)
{
        char c;
        while(!((c=inportb(port+LSR))&THRE)) /*test THRE */
        ;
        outportb(port+TX,data);
}

rport(int port)
{
        char c,r;

        while(!((c = inportb(port+LSR))&DR))
        if(kbhit())
        {
                getch();
                exit(0);
        }
        r = inportb(port+RX);
        return r;
}


int pulldown(num)
int num;
{
        int vmode,choice;

        vmode = video_mode();
        if((vmode!=2)&&(vmode!=3)&&(vmode!=7))
        {
                printf("video must  be in colum 80");
                exit(1);
        }

        if(vmode==7) vid_mem = (char far*) 0xB0000000;
        else vid_mem = (char far*) 0xB8000000;
        if(!frame[num].active)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                save_video(num);
                frame[num].active = 1;
        }
        if(frame[num].border) draw_border(num);

        display_menu(num);
        return get_resp(num);
}

int pulldownx(num)
int num;
{
        int vmode,choice;

        vmode = video_mode();
        if((vmode!=2)&&(vmode!=3)&&(vmode!=7))
        {
                printf("video must  be in colum 80");
                exit(1);
        }

        if(vmode==7) vid_mem = (char far*) 0xB0000000;
        else vid_mem = (char far*) 0xB8000000;

        if(!frame[num].active)
        {
                save_video(num);
                frame[num].active = 1;
        }
        if(frame[num].border) draw_borderx(num);

        display_menux(num);
        return get_respx(num);
}

int pulldownxy(num)
int num;
{
        int vmode,choice;

        vmode = video_mode();
        if((vmode!=2)&&(vmode!=3)&&(vmode!=7))
        {
                printf("video must  be in colum 80");
                exit(1);
        }

        if(vmode==7) vid_mem = (char far*) 0xB0000000;
        else vid_mem = (char far*) 0xB8000000;

        if(!frame[num].active)
        {
                save_video(num);
                frame[num].active = 1;
        }
        if(frame[num].border) draw_border(num);

        display_menuxy(num);
        return get_respxy(num);
}

make_menu(num,menu,keys,count,x,y,border)
int num;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  char *menu[];
  char *keys;
  int count;
  int x,y;
  int border;
  {
        register int i,len;
        int endx,endy,choice,vmode;
        unsigned char *p;

        if(num>MAX_FRAME)
        {
              printf("Too many menus !\n");
              return 0;
        }

        if((y>24)||(y<0)||(x>79)||(x<0))
        {
              printf("Range menu error !\n");
              return 0;
        }

        len=0;
        for(i=0;i<count;i++)
        if(len<strlen(menu[i]))
              len=strlen(menu[i]);
        endx = x+len+3;
        endy = count+1+y;
        if((endy>25)||(endx>80))
        {
              printf("Menu won't fit \n");
              return 0;
        }

        p = (unsigned char*)malloc(2*(endx-x+1)*(endy-y+1));

        if(!p)
              exit(1);

        frame[num].startx = x;
        frame[num].starty = y;
        frame[num].endx = endx;
        frame[num].endy = endy;
        frame[num].p = p;
        frame[num].menu = (char**) menu;
        frame[num].border = border;
        frame[num].keys = keys;
        frame[num].count = count;
        frame[num].active = 0;
        return 1;
}

make_menux(num,menu,keys,count,x,y,border)

int num;
char *menu[];
char *keys;
int count;
int x,y;
int border;
{
        register int i,len;
        int endx,endy,choice,vmode;
        unsigned char *p;

        if(num>MAX_FRAME)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                printf("Too many menus !\n");
                return 0;
        }

        if((y>24)||(y<0)||(x>79)||(x<0))
        {
                printf("Range menu error !\n");
                return 0;
        }

        len=0;
        for(i=0;i<count;i++)
        if(len<strlen(menu[i]))
                len=strlen(menu[i]);
        endy = y;
        endx = 80;
        if((endy>25)||(endx>80))
        {
                printf("Menu won't fit \n");
                return 0;
        }

        p = (unsigned char*)malloc(2*(endx-x+1)*(endy-y+1));

        if(!p)
                exit(1);

        frame[num].startx = x;
        frame[num].starty = y;
        frame[num].endx = endx;
        frame[num].endy = endy;
        frame[num].p = p;
        frame[num].menu = (char**) menu;
        frame[num].border = border;
        frame[num].keys = keys;
        frame[num].count = count;
        frame[num].active = 0;
        return 1;
}

make_menuxy(num,menu,count,x,y,border)

int num;
char *menu[];
int count;
int x,y;
int border;
{
        register int i,len;
        int d,endx,endy,choice,vmode;
        unsigned char *p;

        if(num>MAX_FRAME)
        {
                printf("Too many menus !\n");
                return 0;
        }

        if((y>24)||(y<0)||(x>79)||(x<0))
        {
                printf("Range menu error !\n");
                return 0;
        }

        len=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(i=0;i<count;i++)
        if(len<strlen(menu[i]))
                len=strlen(menu[i]);
        d = mod(60,(len+1));
        endy = y + mod(count,d)+2;

        endx = x+(d*(len+1));
        if((endy>25)||(endx>80))
        {
                printf("Menu won't fit \n");
                return 0;
        }

        p = (unsigned char*)malloc(2*(endx-x+1)*(endy-y+1));

        if(!p)
                exit(1);

        frame[num].startx = x;
        frame[num].starty = y;
        frame[num].endx = endx;
        frame[num].endy = endy;
        frame[num].p = p;
        frame[num].menu = (char**) menu;
        frame[num].border = border;
        frame[num].count = count;
        frame[num].active = 0;
        return 1;
}


void display_menux(num)
int num;
{
        register int x,j;
        char **m;
        x = frame[num].startx;
        m = frame[num].menu;

        for  (j=0;j<frame[num].count;j++,x=x+12)
        write_string(x,frame[num].starty,m[j],0x70);
}

void display_menuxy(num)
int num;
{
        register int x,j,y,i,len;
        char **m;
        x = frame[num].startx+2;
        y = frame[num].starty+1;
        m = frame[num].menu;
        len=0;
        for(i=0;i<frame[num].count;i++)
        if(len<strlen(frame[num].menu[i]))
                len=strlen(frame[num].menu[i]);

                for  (j=0;j<frame[num].count;j++,x=x+1+len)
                {
                if(x>frame[num].endx) {
                                y = y+1;
                                x = frame[num].startx+2;
                        }
                write_string(x,y,m[j],0x70);
                }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void display_menu(num)
int num;
{
        register int i,y,j;
        char **m;
        y = frame[num].starty+1;
        m = frame[num].menu;

        for  (j=0;j<frame[num].count;j++,y++)
        write_string(frame[num].startx+2,y,m[j],0x70);
}

void draw_border(num)
int num;
{
        register int i,j,k,l;
        char far *v,far *t;

        v = vid_mem;
        t=v;

        for(k=frame[num].startx;k<frame[num].endx+1;k++)
        {       for(l=frame[num].starty;l<frame[num].endy+1;l++)
                {
                v += l*160+(k*2);
                *v++ = 0;
                *v =  0x71;
                v=t;
                }
        }


        for(i=frame[num].startx+1;i<frame[num].endx+1;i++)
        {
                v += frame[num].starty*160+(i*2);
                *v++ = 196;
                *v =  0x70;
                v=t;
                v += frame[num].endy*160+(i*2);
                *v++ = 196;
                *v =  0x70;
                v = t;
        }
        for(i=frame[num].starty+1;i<frame[num].endy+1;i++)
        {
                v += i*160+(frame[num].startx*2);
                *v++ = 179;
                *v =   0x70;
                v = t;
                v += (i*160)+frame[num].endx*2;
                *v++ = 179;
                *v =  0x70;
                v = t;
        }
        write_char(frame[num].startx,frame[num].starty,218,0x70);
        write_char(frame[num].startx,frame[num].endy,192,0x70);
        write_char(frame[num].endx,frame[num].starty,191,0x70);
        write_char(frame[num].endx,frame[num].endy,217,0x70);

}

void draw_borderx(num)
int num;
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        register int i,j,k,l;
        char far *v,far *t;

        v = vid_mem;
        t=v;

        for(k=0;k<80;k++)
        {       for(l=frame[num].starty;l<frame[num].endy+1;l++)
                {
                v += l*160+(k*2);
                *v++ = 0;
                *v =  0x71;
                v=t;
                }
        }
}

get_resp(num)
int num;
{
        union inkey{
                char ch[2];
                int i;
                } c;

        int arrow_choice = 0, key_choice;
        int x,y;
        x = frame[num].startx+2;
        y = frame[num].starty+1;

        write_string(x,y+arrow_choice,frame[num].menu[0],0x30);

        for(;;)
        {
                while(!bioskey(1));
                c.i = bioskey(0);

                write_string(x,y+arrow_choice,
                frame[num].menu[arrow_choice],0x70);

                if(c.ch[0])
                {
                        key_choice=is_in(frame[num].keys,tolower(c.ch[0]));
                        if(key_choice) return key_choice -1;

                        switch(c.ch[0])
                        {
                                case '\r': return arrow_choice;
                                case ' '  : arrow_choice ++;
                                break;
                                case ESC : return -1 ;
                        }
                }
                else
                {
                        switch(c.ch[1])
                        {
                                case 72 : arrow_choice--;
                                          break;
                                case 80 : arrow_choice++;
                                          break;
                        }
                }
                if(arrow_choice==frame[num].count) arrow_choice = 0;
                if(arrow_choice<0)arrow_choice=frame[num].count-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                write_string(x,y+arrow_choice,
                frame[num].menu[arrow_choice],0x30);

        }
}

get_respxy(num)
int num;
{
        union inkey{
                char ch[2];
                int i;
                } c;

        register int arrow_choicex= 0 ,arrow_choicey = 0;
        int a,b,x,y,len,i;
        x = frame[num].startx+2;
        y = frame[num].starty+1;

        len=0;
        for(i=0;i<frame[num].count;i++)
        if(len<strlen(frame[num].menu[i]))
                len=strlen(frame[num].menu[i]);
        write_string(x+(arrow_choicex*(len+1)),y+arrow_choicey,
        frame[num].menu[arrow_choicex+(arrow_choicey*mod(frame[num].endx,(len+1)))],0
        for(;;)
        {
                while(!bioskey(1));
                c.i = bioskey(0);

                write_string(x+(arrow_choicex*(len+1)),y+arrow_choicey,
                frame[num].menu[arrow_choicex+(arrow_choicey*mod(frame[num].endx,(len
);

                if(c.ch[0])
                {

                        switch(c.ch[0])
                        {
                                case '\r': a = arrow_choicex;
                                           b = arrow_choicey;
                                           return (a+(b*mod(frame[num].endx,(len+1))));
                                           break;
                                case ' '  :arrow_choicey ++;
                                           break;
                                case ESC : return -1 ;
                        }
                }
                else
                {
                        switch(c.ch[1])
                        {
                                case 72 : arrow_choicey--;
                                          break;
                                case 80 : arrow_choicey++;
                                          break;
                                case 75 : arrow_choicex--;
                                          break;
                                case 77 : arrow_choicex++;
                                          break;
                        }
                }
                if(arrow_choicex>=mod(frame[num].endx,(len+1))) arrow_choicex = 0;
                if(arrow_choicey>(frame[num].endy-frame[num].starty-1))
                arrow_choicey = 0;
                if(arrow_choicex+((arrow_choicey)*(mod(frame[num].endx,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
                (len+1))))>=frame[num].count)
                                arrow_choicey = 0;
                if(arrow_choicex<0) arrow_choicex= mod(frame[num].endx,(len+1))-1;
                if(arrow_choicey<0)
                {
                if((arrow_choicey<0)&&((arrow_choicex+((arrow_choicey)*
                (mod(frame[num].endx,(len+1)))))>=frame[num].count))
                arrow_choicey = (frame[num].endy-frame[num].starty-3);

                arrow_choicey=(frame[num].endy-frame[num].starty-2);
                }

                write_string(x+(arrow_choicex*(len+1)),y+arrow_choicey,
                frame[num].menu[arrow_choicex+(arrow_choicey*mod(frame[num].endx,
                (len+1)))],0x30);

        }
}

get_respx(num)
int num;
{
        union inkey{
                char ch[2];
                int i;
                } c;

        int arrow_choice = 0, key_choice;
        int x,y;
        x = frame[num].startx;
        y = frame[num].starty;

        write_string(x,y,frame[num].menu[0],0x30);

        for(;;)
        {
                while(!bioskey(1));
                c.i = bioskey(0);

                write_string(x+(12*arrow_choice),y,
                frame[num].menu[arrow_choice],0x70);

                if(c.ch[0])
                {
                        key_choice=is_in(frame[num].keys,tolower(c.ch[0]));
                        if(key_choice) return key_choice -1;

                        switch(c.ch[0])
                        {
                                case '\r': return arrow_choice;
                                case ' '  : arrow_choice ++;
                                break;
                                case ESC : return -1 ;
                        }
                }
                else
                {
                        switch(c.ch[1])
                        {
                                case 75 : arrow_choice--;
                                          break;
                                case 77 : arrow_choice++;
                                          break;
                        }
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(arrow_choice==frame[num].count) arrow_choice = 0;
                if(arrow_choice<0)arrow_choice=frame[num].count-1;

                write_string(x+(12*arrow_choice),y,
                frame[num].menu[arrow_choice],0x30);
        }
}

void write_string(x,y,p,attrib)
int x,y;
char *p;
int attrib;
{
        register int i;
        char far *v;

        v = vid_mem;
        v += y*160 + x*2;
        for(i=x; *p; i++)
        {
                *v++ = *p++;
                *v++ = attrib;
        }
}

void write_char(x,y,ch,attrib)
int x,y;
char ch;
int attrib;
{
        register int i;
        char far *v;

        v =  vid_mem;
        v += y*160 + x*2;
        *v++ = ch;
        *v = attrib;
}

void save_video(num)
int num;
{
        register int i,j;
        char *buf_ptr;
        char far *v,  far *t;

        buf_ptr = frame[num].p;
        v = vid_mem;
        for(i=frame[num].starty;i<frame[num].endy+1;i++)
                for(j=frame[num].startx;j<frame[num].endx+1;j++)
                {
                t = v + (i*160) + (j*2);
                *buf_ptr++ = *t++;
                *buf_ptr++ =*t;
                *(t-1) = ' ';
                }


}

void restore_video(num)
int num;
{
        register int i,j;
        char far *v,far *t;
        char *buf_ptr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        buf_ptr = frame[num].p;
        v = vid_mem;
        t = v;
        for(i=frame[num].starty;i<frame[num].endy+1;i++)
                for(j=frame[num].startx;j<frame[num].endx+1;j++)
                {
                v = t;
                v += (i*160) + (j*2);
                *v++ = *buf_ptr++;
                *v = *buf_ptr++;
                }
        frame[num].active = 0;
        free(frame[num].p);

}
video_mode()
{
        union REGS r;

        r.h.ah = 15;
        return int86(0x10,&r,&r)&255;
}
is_in(s,c)
char *s,c;
{
        register int i;
        for(i=0;*s;i++)
                if(*s++==c) return i+1;
        return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        org 8000h
        ajmp START
        org 8050h
START:  mov TH1,#0fdh
        mov SCON,#52h
        mov TMOD,#20H
        setb TR1
        mov  SP,#6Ch
        mov PSW,#00h

lop:    acall E
        mov r3,#00h
        mov r4,#08h
        mov a,#00h
        mov r2,#00h
        clr c
        mov r0,#20h
ll:     mov a,@r0
        addc a,r2
        mov r2,a
        mov a,r3
        addc a,#00h
        mov r3,a
        inc r0
        djnz r4,ll
        mov a,r2
        acall SBYTE
        mov a,r3
        acall SBYTE
        acall RBYTE
        cjne a,#00h,lop


        mov a,22h           ;vrm
        rl a
        rl a
        rl a
        rl a
        orl a,21h           ;cr1
        mov 29h,a
        mov a,25h           ;vr1
        rl a
        rl a
        rl a
        rl a
        orl a,23h           ;vrc
        mov 2Ah,a
        mov a,29h           ;vrm,cr1
        mov dptr,#0E009h
        movx @dptr,a
        mov a,2Ah           ;vr1,vrc
        mov dptr,#0E00Ah
        movx @dptr,a
        mov a,24h           ;crc
        rl a
        rl a
        rl a
        rl a
        mov dptr,#0E008h
        movx @dptr,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select0:    mov PSW,#00h
            mov r0,20h
            cjne r0,#01,SELECT1     ;if not NPN go to SELECT1
            ajmp NPN                ;case NPN
SELECT1:    jc SELECT2              ;case diode
            ajmp PNP                ;case PNP
SELECT2:    ajmp DIODE

 DIODE:     acall AA
            acall BB
    n:      acall AA
            ajmp n
  NPN:      acall F
            acall G
   nu:      acall F
            ajmp nu
  PNP:      acall H
            acall G
  nuc:      acall H
            ajmp nuc

  END:      ajmp select0


 IREC1:     mov a,#00h
            acall SBYTE
 IREC:      acall RBYTE
            cjne a,#0ffh,IREC1
            mov a,#0feh
            acall SBYTE
            ret
 ISEND:     mov a,#0ffh
            acall SBYTE
            acall RBYTE
            cjne a,#0fch,ISEND
            ret
 SBYTE:     jnb TI,SBYTE
            clr TI
            mov sbuf,a
            ret
 RBYTE:     jnb RI,RBYTE
            clr RI
            mov a,sbuf
            ret


   AA:      mov PSW,#00h
            mov r3,#00h
            mov r0,#00h             ;begin current address lowbyte
            mov r2,#00h
            mov a,r2
            mov dptr,#0E001h
            movx @dptr,a
            mov r7,#00h
 suwa:      djnz r7,suwa
 SA11:      mov a,r2
            mov dptr,#0E001h
            movx @dptr,a
            mov dptr,#0E005h
            movx @dptr,a
            nop                     ;wait for ADC
            nop
            nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nop
        nop
        nop
        movx a,@dptr
        mov P2,#20h          ;current address highbyte
        movx @r0,a           ;begin at 2000h
        inc r0               ;inc address
        inc r2               ;inc volt
        djnz r3,SA11
        ret

BB:     mov r2,#00h          ;send 256 of current value to com
        mov dptr,#2000h      ;from address 2000h
        acall ISEND
SA22:   movx a,@dptr
        acall SBYTE
        inc dptr
        djnz r2,SA22
        ret
E:      mov PSW,#00h
        mov r2,#08h
        mov r0,#20h
        acall IREC
loop1:  acall RBYTE
        mov @r0,a
        inc r0
        djnz r2,loop1
        ret

        mov PSW,#00h
F:      mov r1,#20h
        mov r0,#00h
        mov r5,26h           ;number of step IB
        mov r6,27h           ;IB value for one step
        mov r4,27h           ;IB value
        mov a,#80h
        add a,r4
        mov r4,a
SF22:   mov r3,#80h          ;counter
        mov r2,#80h          ;volt value
        mov a,r2
        mov dptr,#0E001h
        movx @dptr,a
        mov a,r4
        mov dptr,#0E000h
        movx @dptr,a
        mov r7,#00h
suwa2:  djnz r7,suwa2

SF11:   mov a,r2
        mov dptr,#0E001h
        movx @dptr,a
        mov a,r4
        mov dptr,#0E000h
        movx @dptr,a
        mov dptr,#0E005h
        movx @dptr,a
        mov 30h,#00h
ss:     djnz 30h,ss
        nop
        nop
        nop
        nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nop
        nop
        movx a,@dptr            ;Ic
        mov dph,r1
        mov dpl,r0
        movx @dptr,a
        inc dptr
        mov r1,dph
        mov r0,dpl
        inc r2                  ;inc volt
        djnz r3,SF11
        mov a,r6
        add a,r4
        mov r4,a
        djnz r5,SF22
        ret


  G:    mov r5,26h              ;number of step
        mov dptr,#2000h
        acall ISEND
SG22:   mov r2,#80h
SG11:   movx a,@dptr
        acall SBYTE
        inc dptr
        djnz r2,SG11
        djnz r5,SG22
        ret


  H:    mov r0,#00h
        mov r1,#20h
        mov r5,26h              ;number of step IB
        mov r6,27h              ;IB value for one step
        mov r4,27h              ;IB value
        mov a,#80h
        subb a,r4
        mov r4,a
SH22:   mov r2,#80h             ;volt

        mov a,r2
        mov dptr,#0E001h
        movx @dptr,a
        mov a,r4
        mov dptr,#0E000h
        movx @dptr,a

        mov r3,#30h
 sa:    mov r7,#00h
 wa:    djnz r7,wa
        djnz r3,sa

SH11:   mov a,r2
        mov dptr,#0E001h
        movx @dptr,a
        mov a,r4
        mov dptr,#0E000h
        movx @dptr,a
        mov dptr,#0E005h
        movx @dptr,a
        nop
        nop
        nop
        nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nop
nop
nop
movx a,@dptr
mov dph,r1
mov dpl,r0
movx @dptr,a
inc dptr
mov r1,dph
mov r0,dpl
djnz r2,SH11
mov a,r4
subb a,r6
mov r4,a
djnz r5,SH22
ret
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	plastic	commercial	no
D	cerdip	commercial	no
N	PLCC	commercial	no
R	LCC	commercial	no
TD	cerdip	extended	no
QP	plastic	commercial	yes
QD	cerdip	commercial	yes
LD	cerdip	extended	yes

Please note:

- Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
- Burn-in is dynamic, for a minimum time of 160 hours at 125°C, V_{CC} = 5.5V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples: N8752BH indicates 8752BH in a PLCC package and specified for commercial temperature range, without burn-in. LD8752BH indicates 8752BH in a cerdip package and specified for extended temperature range with burn-in.

80C51BH/80C51BH-1/80C51BH-2 CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER WITH FACTORY MASK-PROGRAMMABLE ROM

80C31BH/80C31BH-1/80C31BH-2 CHMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU WITH RAM AND I/O

80C51BH/80C31BH—3.5 to 12 MHz, V_{CC} = 5V ±20%
80C51BH-1/80C31BH-1—3.5 to 16 MHz, V_{CC} = 5V ±20%
80C51BH-2/80C31BH-2—0.5 to 12 MHz, V_{CC} = 5V ±20%

- Power Control Modes
- 128 x 8-Bit RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 64K Program Memory Space
- High Performance CHMOS Process
- Boolean Processor
- 5 Interrupt Sources
- Programmable Serial Port
- 64K Data Memory Space

The MCS®-51 CHMOS products are fabricated on Intel's CHMOS III process and are functionally compatible with the standard MCS-51 HMOS and EPROM products. CHMOS III is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CHMOS. This combination expands the effectiveness of the powerful MCS-51 architecture and instruction set.

Like the MCS-51 HMOS versions, the MCS-51 CHMOS products have the following features: 4K byte of ROM (80C51BH/80C51BH-1/80C51BH-2 only); 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-source two-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, the MCS-51 CHMOS products have two software selectable modes of reduced activity for further power reduction—Idle and Power Down.

The Idle mode freezes the CPU while allowing the RAM, timer/counters serial port and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

October 1988
Order Number: 270064-006

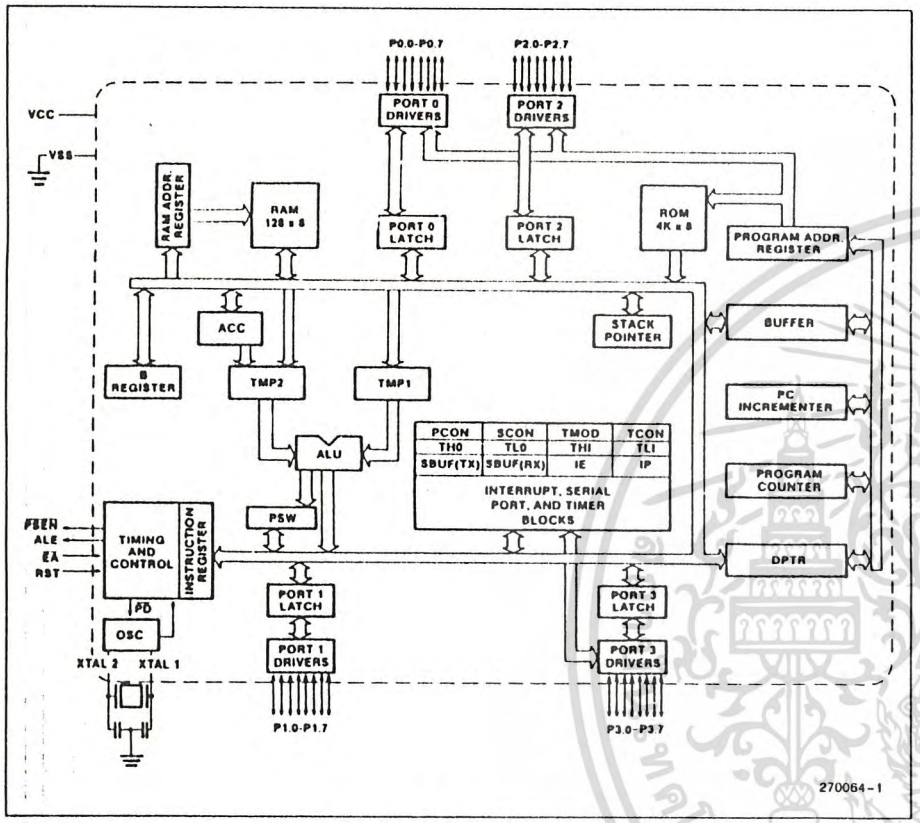


Figure 1. Block Diagram

IDLE MODE

In the Idle mode, the CPU puts itself to sleep while all the on chip peripherals stay active. The instruction that invokes the Idle mode is the last instruction executed in the normal operating mode before Idle mode is activated. The content of CPU, the on chip RAM, and all the Special Function Registers remain intact during this mode. The Idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which starts the processor the same as a power on reset.

POWER DOWN MODE

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

The only exit from Power Down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

The control bits for the reduced power modes are in the Special Function Register PCON.

NOTE:
For more detailed information on these reduced power modes refer to Application Note AP-252, "Designing with the 80C51BH".

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

PACKAGES

Part	Prefix	Package Type
80C51BH/ 80C31BH*	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC

*The 80C51BH-1, 80C51BH-2, 80C31BH-1, and 80C31BH-2 have the same package types.

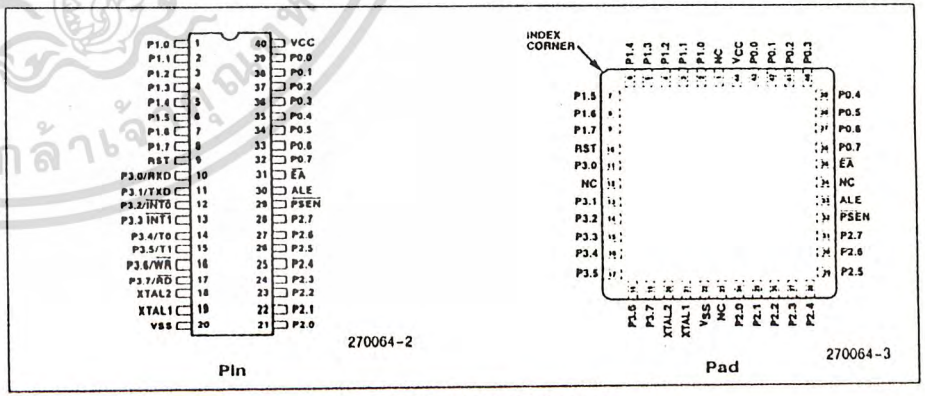


Figure 2. Connection Diagrams

PIN DESCRIPTIONS

V_{CC}

Supply voltage during normal, Idle, and Power Down operations.

V_{SS}

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

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

Port 1

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

Port 2

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

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

Port 3

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

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

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

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits Power-On reset using only an external capacitor to V_{CC}.

ALE

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

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

$\overline{PSEN}$

Program Store Enable is the read strobe to external Program Memory.

When the 80C51BH is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.

$\overline{EA}$

External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from external Program Memory locations starting at 0000H up to FFFFH. If $\overline{EA}$ is strapped to V_{CC} the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock generator circuits.

XTAL2

Output from the inverting oscillator amplifier.

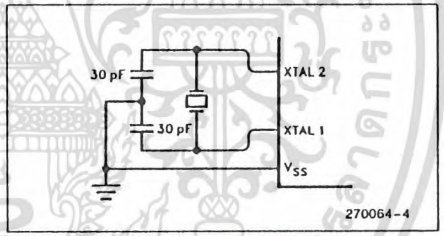


Figure 3. Crystal Oscillator

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillator for Microcontrollers".

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

Design Considerations

- At power on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up.
- Before entering the Power Down mode the contents of the Carry Bit and B.7 must be equal.
- When the Idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

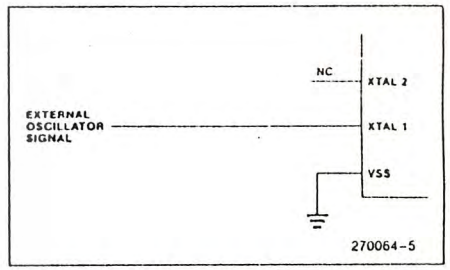


Figure 4. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Voltage on any Pin to V_{SS} ..... −0.5V to V_{CC} +0.5V
Voltage on V_{CC} to V_{SS} ..... −0.5V to 6.5V
Maximum I_{OL} per I/O pin ..... 15 mA
Power Dissipation 1.0W*

*This value is based on the maximum allowable die temperature and the thermal resistance of the package.

**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.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ(3)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		0.2 V_{CC} − 0.1	V	
V_{IL1}	Input Low Voltage ($\overline{EA}$)	−0.5		0.2 V_{CC} − 0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (6) (Ports 1, 2, 3)			0.45	V	I_{OL} = 1.6 mA (1)
V_{OL1}	Output Low Voltage (6) (Port 0, ALE, $\overline{PSEN}$)			0.45	V	I_{OL} = 3.2 mA (1)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4			V	I_{OH} = −60 µA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −25 µA
		0.9 V_{CC}			V	I_{OH} = −10 µA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −800 µA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −300 µA
		0.9 V_{CC}			V	I_{OH} = −80 µA (2)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			−50	µA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)			−650	µA	V_{IN} = 2V
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)			±10	µA	0.45 < V_{IN} < V_{CC}
RRST	Reset Pulldown Resistor	50		150	KΩ	
CIO	Pin Capacitance			10	pF	Test Freq = 1 MHz, T_A = 25°C
I_{CC}	Power Supply Current: Active Mode, 12 MHz (4)		11	20	mA	(5)
	Idle Mode, 12 MHz (4)		1.7	5	mA	
	Power Down Mode		5	50	µA	

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. "Typicals" are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temperature, 5V.
4. ICCMAX at other frequencies is given by
   Active Mode: ICCMAX = 1.47 × FREQ + 2.35
   Idle Mode: ICCMAX = 0.33 × FREQ + 1.05
   where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See Figure 5.
5. See Figures 6 through 9 for I_{CC} test conditions. Minimum V_{CC} for Power Down is 2V.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port-
   Port 0: 26 mA
   Ports1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

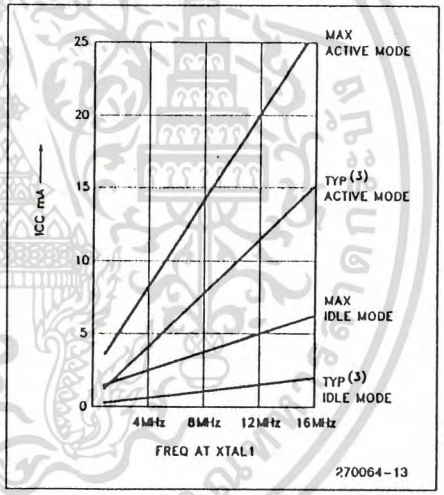


Figure 5. I_{CC} vs. Frequency. Valid only within frequency specifications of the device under test.

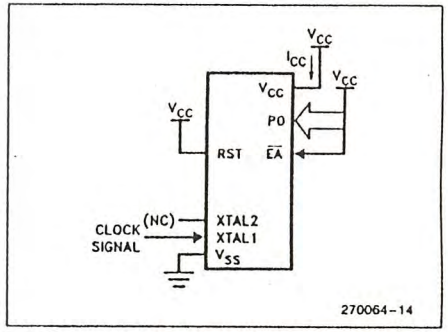


Figure 6. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

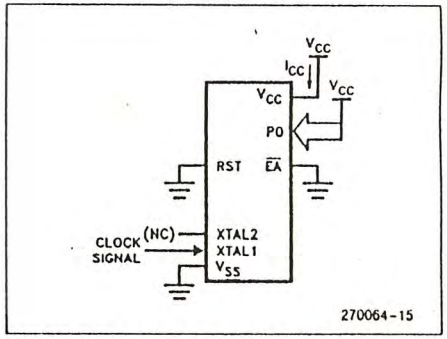


Figure 7. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

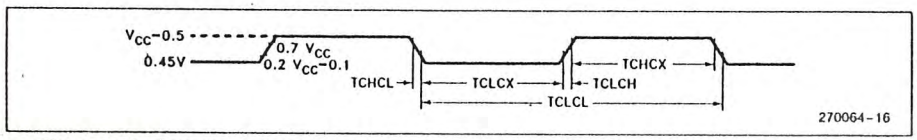


Figure 8. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

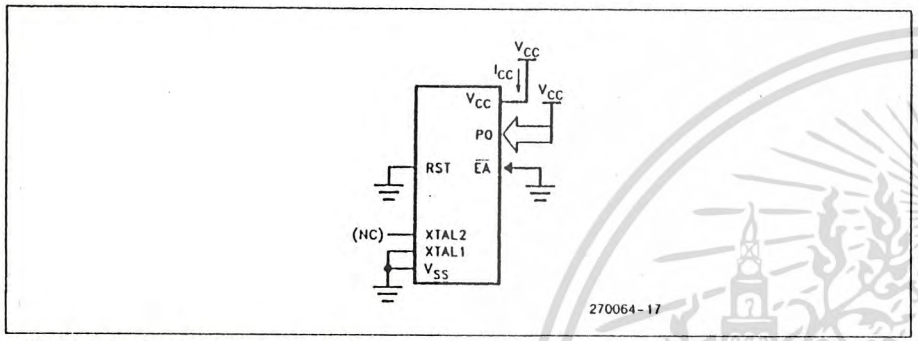


Figure 9. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected. V_{CC} = 2V to 6V.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address.
C: Clock.
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: $\overline{PSEN}$.
Q: Output data.
R: $\overline{RD}$ signal.
T: Time.
V: Valid.
W: $\overline{WR}$ signal.
X: No longer a valid logic level.
Z: Float.

EXAMPLE:

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{PSEN}$ Low.

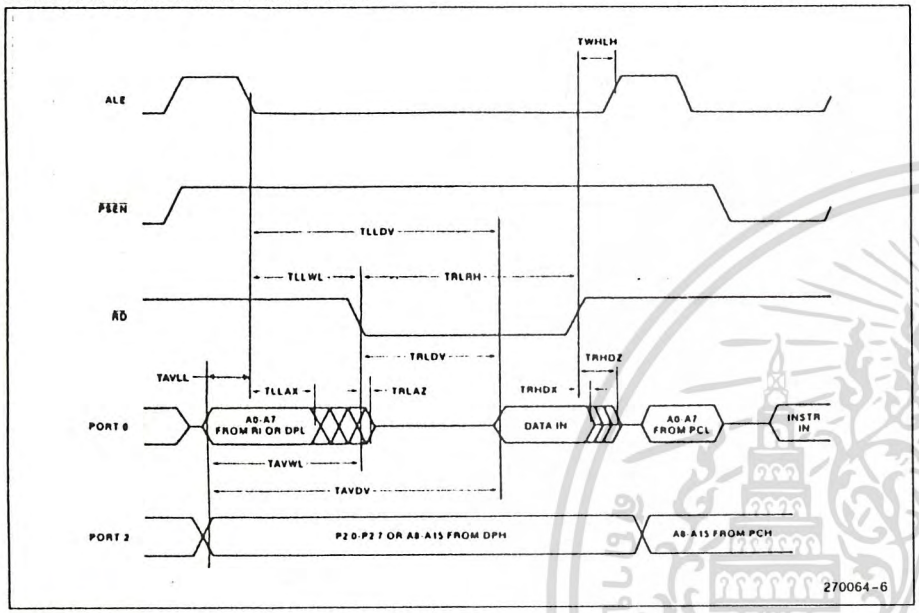
A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

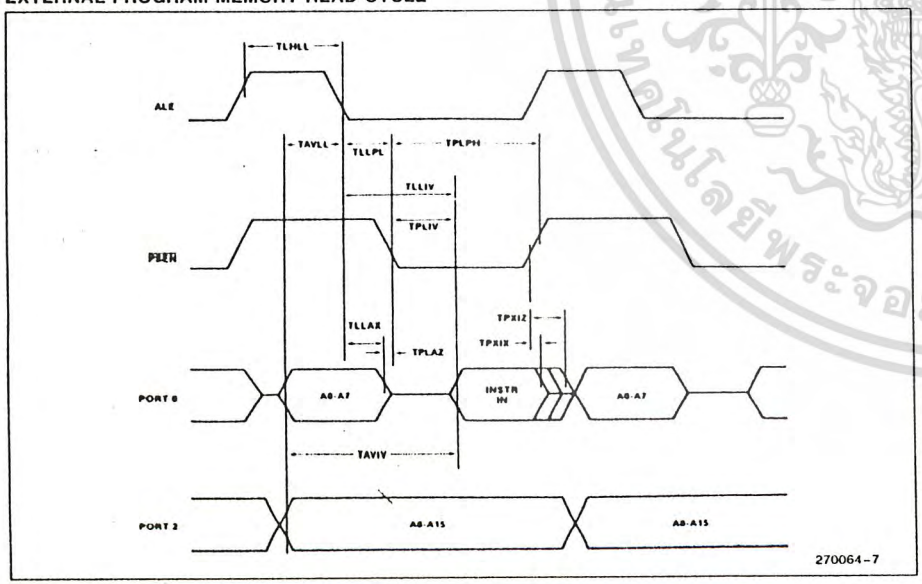
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL − 40		ns
TAVLL	Address Valid to ALE Low	28		TCLCL − 55		ns
TLLAX	Address Hold After ALE Low	48		TCLCL − 35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL − 100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	43		TCLCL − 40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL − 45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		145		3TCLCL − 105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		59		TCLCL − 25	ns
TAVIV	Address to Valid Instr In		312		5TCLCL − 105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL − 100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL − 100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL − 165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL − 70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL − 150	ns
TAVDV	Address to Valid Data In		585		9TCLCL − 165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL − 50	3TCLCL + 50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL − 130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL − 60		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL − 50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL − 40	TCLCL + 40	ns

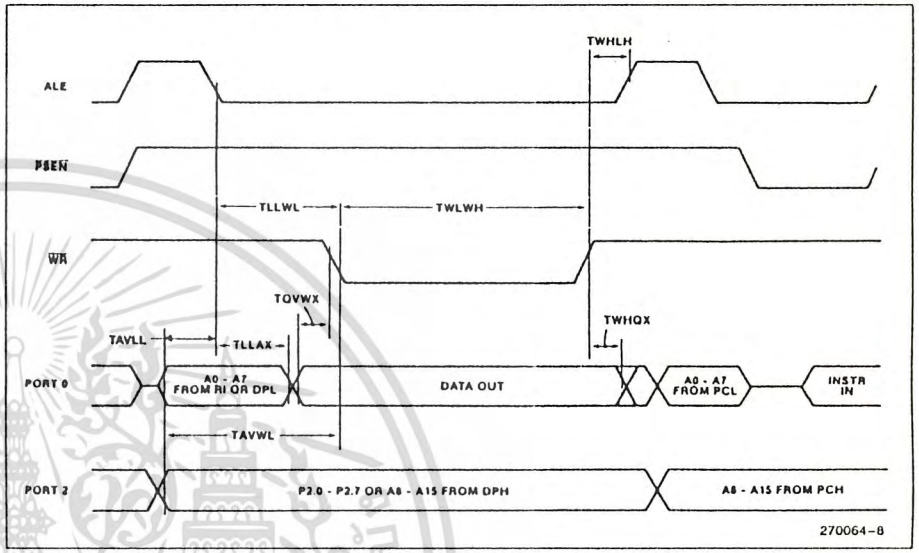
EXTERNAL DATA MEMORY READ CYCLE

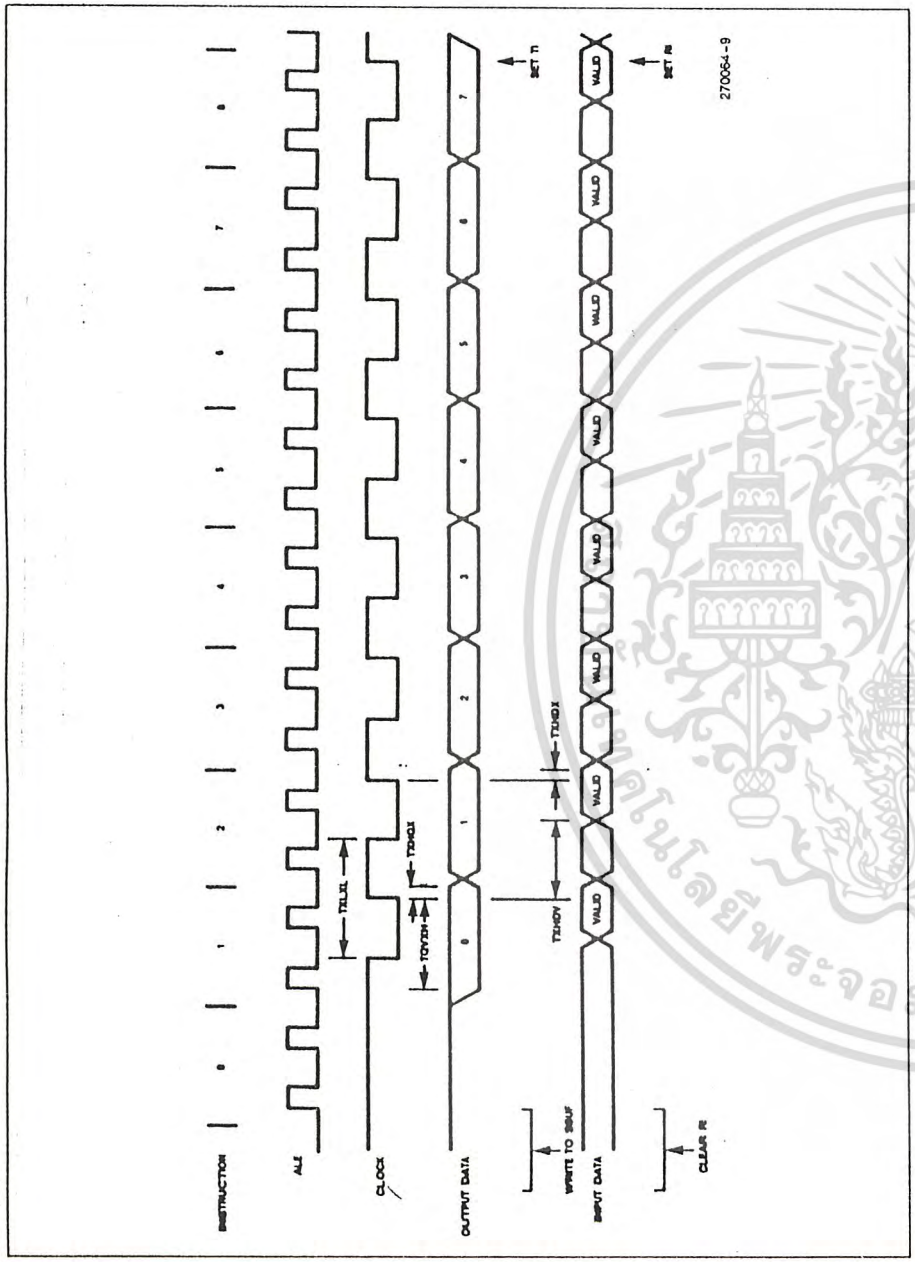


EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE





Shift Register Mode Timing Waveforms

EXTERNAL CLOCK DRIVE

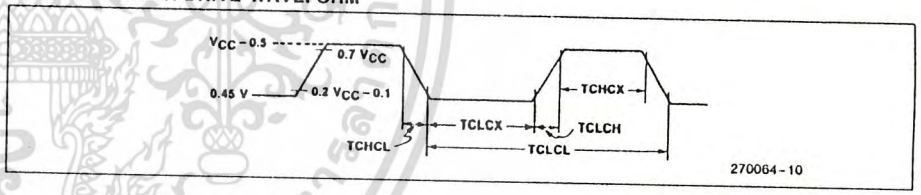
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

SERIAL TIMING—SHIFT REGISTER MODE

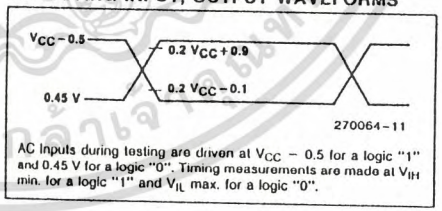
Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL − 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL − 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL − 133	ns

EXTERNAL CLOCK DRIVE WAVEFORM

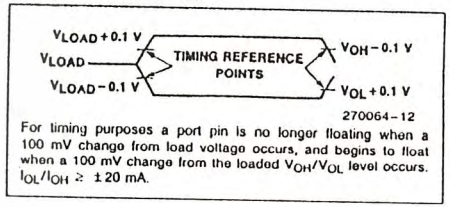


AC TESTING INPUT, OUTPUT WAVEFORMS



AC Inputs during testing are driven at V_{CC} − 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".

FLOAT WAVEFORMS



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -005 version of the 80C51BH data sheet:

1. Package table was added.
2. Note 6 on maximum current specifications added to DC Characteristics.
3. Data Sheet Revision Summary was added.

80C31BH/80C51BH
EXPRESS

- **Extended Temperature Range**
- **3.5 to 12 MHz V_{CC} = 5V ±20%**
- **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 160 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

September 1987
Order Number: 270218-002

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS T_A = −40°C to +85°C; V_{CC} = 5V ±20%; V_{SS} = 0V

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except EA)	−0.5	$0.2V_{CC}$ − 0.15	V	
V_{IL1}	EA	−0.5V	$0.2V_{CC}$ − 0.35	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	$0.2V_{CC}$ + 1	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage to XTAL1, RST	$0.7V_{CC}$ + 0.1	V_{CC} + 0.5	V	
I_{IL}	Logical 0 Input Current (Port 1, 2, 3)		−75	µA	V_{in} = 0.45V
I_{TL}	Logical 1 to 0 transition Current (Ports 1, 2, 3)		−750	µA	V_{in} = 2.0V

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	Plastic	Commercial	No
D	Cerdip	Commercial	No
N	PLCC	Commercial	No
TP	Plastic	Extended	No
TD	Cerdip	Extended	No
TN	PLCC	Extended	No
QP	Plastic	Commercial	Yes
QD	Cerdip	Commercial	Yes
QN	PLCC	Commercial	Yes
LP	Plastic	Extended	Yes
LD	Cerdip	Extended	Yes
LN	PLCC	Extended	Yes

NOTE:
• Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
• Burn-in is dynamic for a minimum time of 160 hours at 125°C, V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P80C31BH indicates 80C31BH in a plastic package and specified for commercial temperature range, without burn-in.
LD80C51BH indicates 80C51BH in a cerdip package and specified for extended temperature range with burn-in.

PM-7574

CMOS MICROPROCESSOR-COMPATIBLE 8-BIT A/D CONVERTER

FEATURES

- 8-Bit Resolution and Accuracy
- No Missing Codes Over Full Temperature Range
- 15μs Conversion Time
- Flexible μP Interface
- 5mA Maximum Standby Current
- Low Cost
- Pin and Function Compatible With AD7574

ORDERING INFORMATION†

		PACKAGE: 18-PIN DIP AND SO		
INL (LSB)	DNL (LSB)	MILITARY* TEMPERATURE −55°C TO +125°C	INDUSTRIAL TEMPERATURE −40°C TO +85°C	COMMERCIAL TEMPERATURE 0°C TO 70°C
±1/2	±3/4	PM7574AX	PM7574EX	PM7574GP
±3/4	±7/8	PM7574BX	PM7574FX	PM7574HP
±1/2	±3/4	—	—	PM7574GS††

* For devices processed in total compliance to MIL-STD-883, add /883 after part number. Consult factory for 883 data sheet.

† Burn-in is available on commercial and industrial temperature range parts in cerdip, plastic dip, and TO-can packages. For ordering information, see 1988 Data Book, Section 2.

†† For availability and burn-in information on SO and PLCC packages, contact your local sales office.

GENERAL DESCRIPTION

The PM-7574 is a monolithic CMOS successive-approximation analog-to-digital converter. When used with a 550kHz clock, a conversion time of 15μs is achieved, with full accuracy over the operating temperature range.

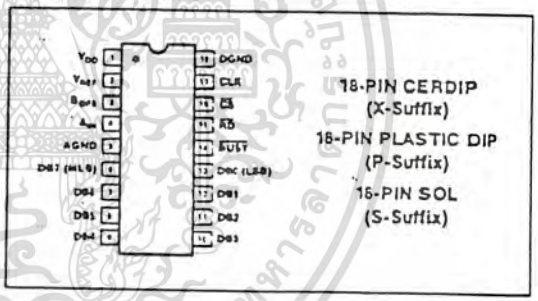
The PM-7574 outputs use 3-state logic, allowing direct connection to the data bus or system input port. Active-LOW chip select ($\overline{CS}$) and read/write ($\overline{RD}$) inputs are used to control all operations. This simplicity permits the PM-7574 to be used as a memory-mapped input device. Depending on the control timing waveforms, the PM-7574 is interfaced like static RAM, ROM, or slow memory.

The low power consumption of the PM-7574 is derived from a single +5V supply. A negative reference voltage must also be supplied. Optimum accuracy is achieved when the reference is at −10.00V with low output resistance. For a low-cost precision −10V/−10.24V reference, ask your PMI sales representative about the REF-08.

With its on-board comparator, interface logic, optional internal clock, and +5V operation, the PM-7574 is the ideal low-cost solution for microprocessor-based 8-bit A/D systems.

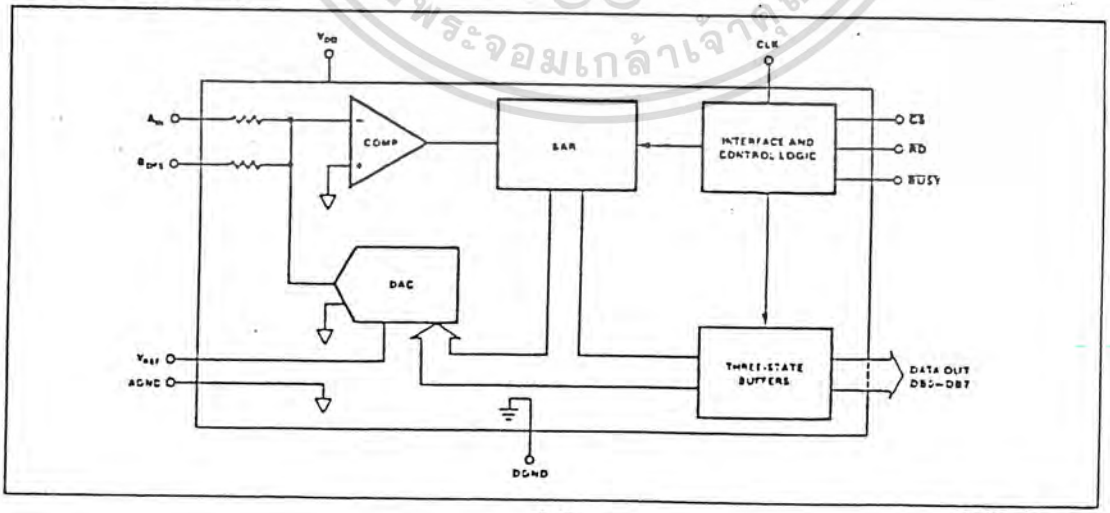
For new designs, PMI's ADC-908 is pin-and-function compatible with the PM-7574, but offers faster conversion time and faster microprocessor bus interface timing.

PIN CONNECTIONS



18-PIN CERDIP (X-Suffix)
18-PIN PLASTIC DIP (P-Suffix)
18-PIN SOL (S-Suffix)

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS

(T_A = +25°C, unless otherwise noted.)

V_{DD} to AGND ..... 0V, +7.0V
V_{DD} to DGND ..... 0V, +7.0V
AGND to DGND ..... −0.3V, V_{DD}
$\overline{CS}$, $\overline{RD}$ to DGND ..... −0.3V, V_{DD} + 0.3V
DB0-DB7 to DGND ..... −0.3V, V_{DD}
CLK, $\overline{BUSY}$ to DGND ..... −0.3V, V_{DD}
B_{OFS}, A_{IN} ..... ±20V
V_{REF} ..... 0V, −20V
Operating Temperature Range
PM-7574AX, BX ..... −55°C to +125°C
PM-7574EX, FX ..... −40°C to +85°C
PM-7574GP, GS, HP ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Lead Temperature (Soldering, 10 sec) ..... +300°C
Power Dissipation (Package)
Ceramic (Suffix X) and Plastic (Suffix P)
To +75°C 450mW
Derate Above +75°C by ..... 6mW/°C
Derate Plastic Above +70°C by ..... 8.3mW/°C
Small Outline Wide
To +70°C 400mW
Derate Above 70°C by ..... 10mW/°C

NOTE:
Digital pins are Zener protected. However, proper ESD handling precautions are recommended.

ELECTRICAL CHARACTERISTICS

at V_{DD} = +5V, V_{REF} = −10V, Unipolar Configuration, R_{CLK} = 150kΩ, C_{CLK} = 100pF; −40°C ≤ T_A ≤ +85°C for PM-7574E/F, 0°C ≤ T_A ≤ +70°C for PM-7574G/H, −55°C ≤ T_A ≤ +125°C for PM-7574A/B, unless otherwise noted.

PARAMETER	SYMBOL	CONDITIONS	PM-7574 MIN	TYP	MAX	UNITS
ACCURACY						
Resolution	N		8	—	—	Bits
Integral Nonlinearity	INL	A/E/G Grades	—	—	±1/2	LSB
		B/F/H Grades	—	—	±3/4	
Differential Nonlinearity	DNL	A/E/G Grades	—	—	±3/4	LSB
		B/F/H Grades	—	—	±7/8	
Gain Error	G_{FSE}	A/E/G Grades T_A = +25°C	—	—	±3	LSB
		T_A = Full Temp Range	—	—	±4.5	
		B/F/H Grades T_A = +25°C	—	—	±5	
		T_A = Full Temp Range	—	—	±6.5	
Offset Error	V_{ZSE}	A/E/G Grades T_A = +25°C	—	—	±30	mV
		T_A = Full Temp Range	—	—	±50	
		B/F/H Grades T_A = +25°C	—	—	±60	
		T_A = Full Temp Range	—	—	±80	
ANALOG INPUTS						
Resistance Mismatch B_{OFS} to A_{IN}	ΔR_{AB}		—	±1.5	—	%
Input Resistance at V_{REF} (Note 1)	R_{REF}		5	—	15	kΩ
Input Resistance at B_{OFS}, A_{IN}	R_{BOFS} R_{IN}		10	—	30	kΩ
Reference Voltage	V_{REF}	Specified Conversion Accuracy	—	−10	—	V
Reference Voltage Range	V_{REF}	Degraded Conversion Accuracy	−5	—	−15	V
Reference Current (Note 6)	I_{REF}	Conversion Complete Prior to Reset	—	—	2.4	mA
Nominal Analog Input Range						
Unipolar Mode	V_{INU}		—	0 to +$\lvert V_{REF}\rvert$	—	V
Bipolar Mode	V_{INB}		—	−$\lvert V_{REF}\rvert$ to +$\lvert V_{REF}\rvert$	—	
LOGIC INPUTS						
Input HIGH Voltage $\overline{RD}$, $\overline{CS}$ Inputs	V_{IH}		2.4	—	—	V
Input LOW Voltage $\overline{RD}$, $\overline{CS}$ Inputs	V_{IL}		—	—	0.8	V

ELECTRICAL CHARACTERISTICS at V_{DD} = +5V, V_{REF} = −10V, Unipolar Configuration, R_{CLK} = 150kΩ, C_{CLK} = 100pF; −40°C ≤ T_A ≤ +85°C for PM-7574E/F, 0°C ≤ T_A ≤ +70°C for PM-7574G/H, −55°C ≤ T_A ≤ +125°C for PM-7574A/B, unless otherwise noted. (Continued)

PARAMETER	SYMBOL	CONDITIONS	PM-7574 MIN	TYP	MAX	UNITS
Input Current $\overline{RD}$, $\overline{CS}$ Inputs	I_{IN}	T_A = +25°C	—	—	1	µA
		T_A = Full Temp Range	—	—	10	
Input Capacitance $\overline{RD}$, $\overline{CS}$ Inputs (Note 6)	C_{IN}		—	—	5	pF
Input HIGH Voltage, Clock Input	V_{IH}		3	—	—	V
Input LOW Voltage, Clock Input	V_{IL}		—	—	0.4	V
Input HIGH Current, Clock Input	I_{IH}		—	—	2	mA
Input LOW Current, Clock Input	I_{IL}	T_A = +25°C	—	—	1	µA
		T_A = Full Temp Range	—	—	10	
LOGIC OUTPUTS						
Output HIGH Voltage $\overline{BUSY}$, DB0-7	V_{OH}	I_{SOURCE} = 40µA	4.0	—	—	V
Output LOW Voltage $\overline{BUSY}$, DB0-7	V_{OL}	I_{SINK} = 1.6mA	—	—	0.4	V
Floating Leakage Current, DB0-7	I_{LKG}	T_A = +25°C	—	—	1	µA
		T_A = Full Temp Range	—	—	10	
Floating State Output Capacitance		(Note 6)	—	—	7	pF
POWER REQUIREMENTS						
Standby Current	I_{DD}	V_{DD} = +4.75V to +5.25V	—	—	5	mA
DIGITAL INTERFACE TIMING						
$\overline{CS}$ Minimum Pulse Width (Note 6)	t_{CS}	T_A = +25°C	100	—	—	ns
		T_A = Full Temp Range	150	—	—	
$\overline{RD}$ to $\overline{CS}$ Setup Time (Note 6)	t_{WSCS}		0	—	—	ns
$\overline{CS}$ to $\overline{BUSY}$ Propagation Delay (Note 6)	t_{CSPD}	BUSY Load = 20pF				ns
		T_A = +25°C	—	—	120	
		T_A = T_{MIN}	—	—	120	
		T_A = T_{MAX}	—	—	180	
		BUSY Load = 100pF				
		T_A = +25°C	—	—	150	
		T_A = T_{MIN}	—	—	150	
		T_A = T_{MAX}	—	—	200	
$\overline{BUSY}$ to $\overline{RD}$ Setup Time (Notes 2, 6)	t_{BSR}		0	—	—	ns
$\overline{BUSY}$ to $\overline{CS}$ Setup Time (Note 6)	t_{BSCS}		0	—	—	ns
Data Access Time (Note 6)	t_{RAD}	C_L = 20pF				ns
		T_A = +25°C	—	—	150	
		T_A = T_{MIN}	—	—	150	
		T_A = T_{MAX}	—	—	220	
		C_L = 100pF				
		T_A = +25°C	—	—	300	
		T_A = T_{MIN}	—	—	300	
		T_A = T_{MAX}	—	—	400	
Data Hold Time (Notes 3, 6)	t_{RHD}	T_A = +25°C (Note 2)	50	—	120	ns
		T_A = T_{MIN}	30	—	80	
		T_A = T_{MAX}	80	—	140	

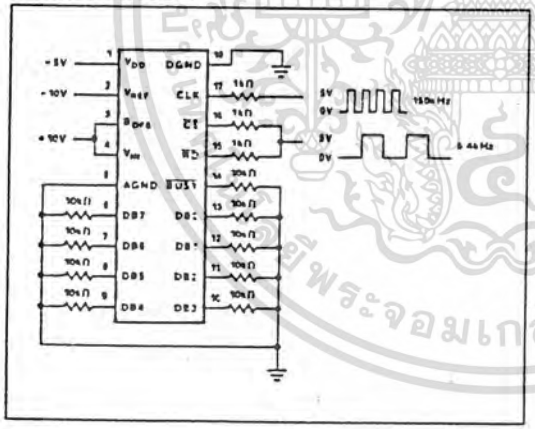
ELECTRICAL CHARACTERISTICS at V_{DD} = +5V, V_{REF} = −10V, Unipolar Configuration, R_{CLK} = 150kΩ, C_{CLK} = 100pF; −40°C ≤ T_A ≤ +85°C for PM-7574E/F, 0°C ≤ T_A ≤ +70°C for PM-7574G/H, −55°C ≤ T_A ≤ +125°C for PM-7574A/B, unless otherwise noted. (Continued)

PARAMETER	SYMBOL	CONDITIONS	PM-7574 MIN	TYP	MAX	UNITS
$\overline{CS}$ to $\overline{RD}$ Hold Time (Note 6)	t_{RHCS}	T_A = +25°C	—	—	250	ns
		T_A = T_{MIN}	—	—	200	
		T_A = T_{MAX}	—	—	500	
Reset Time Requirement (Note 6)	t_{RESET}		3	—	—	µs
Conversion Time (Note 4) (Notes 4, 5, 6)	$t_{CONVERT}$	Static RAM Mode External Clock f = 550kHz	—	—	15	µs
		ROM Mode Internal Clock	—	—	17	
$\overline{RD}$ HIGH to $\overline{BUSY}$ Propagation Delay, ROM Mode (Notes 4, 5, 6)	t_{HBPD}	C_L = 20pF T_A = +25°C	—	—	1.5	µs
		T_A = T_{MIN}	—	—	1.0	
		T_A = T_{MAX}	—	—	2.0	

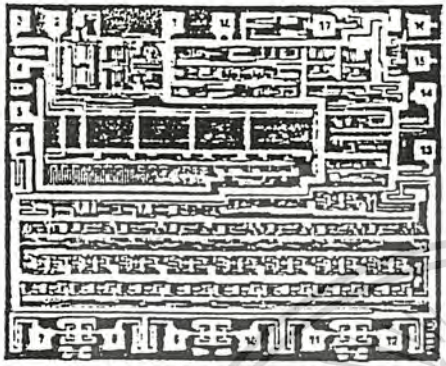
NOTES:

1. For optimum gain accuracy over the full temperature range, the source resistance at pin 2 should be kept low.
2. In ROM mode, $\overline{RD}$ can go LOW prior to $\overline{BUSY}$ = HIGH, but must not return HIGH until $\overline{BUSY}$ = HIGH.
3. Output loading 10pF. A 3kΩ pullup resistor to +5V is used for V_{OL} to High-Z; for V_{OH} to High-Z, a 3kΩ pulldown to GND is used. Measured to 0.5V output change.
4. When using the PM-7574 internal oscillator, actual conversion time depends on clock resistor and capacitor as well as temperature.
5. ROM interface mode conversion times are typically 2µs longer than conversion times for other modes, but the ROM interface mode includes an automatic reset in the conversion time.
6. Guaranteed but not tested.

BURN-IN CIRCUIT



DICE CHARACTERISTICS



DIE SIZE 0.129 X 0.103 inch, 13,287 sq. mils
(3.28 X 2.62 mm, 8.58 sq. mm)

1. V_{DD}
2. V_{REF}
3. B_{OFS}
4. A_{IN}
5. AGND
6. DB7(MSB)
7. DB6
8. DB5
9. DB4
10. DB3
11. DB2
12. DB1
13. DB0(LSB)
14. $\overline{BUSY}$
15. $\overline{RD}$
16. $\overline{CS}$
17. CLK
18. DGND

For additional DICE ordering information, refer to 1988 Data Book, Section 2.

WAFER TEST LIMITS at V_{DD} = +5V, V_{REF} = +10.000V, AGND = DGND = 0V, T_A = +25°C, unless otherwise noted.

PARAMETER	SYMBOL	CONDITIONS	PM-7574G LIMIT	UNITS
STATIC ACCURACY				
Resolution	N		8	Bits MIN
Integral Nonlinearity	INL		±3/4	LSB MAX
Differential Nonlinearity	DNL		±7/8	LSB MAX
Gain Error	G_{ERR}		±5	LSB MAX
Offset Error	V_{ZSE}		±50	mV MAX
ANALOG INPUTS				
Resistance Mismatch B_{OFS} to A_{IN}	ΔR_{AB}		±1.5	% MAX
Input Resistance at V_{REF}	R_{REF}		5/15	kΩ MIN/MAX
Input Resistance at B_{OFS}, A_{IN}	R_{BOFS}, R_{IN}		10/30	kΩ MIN/MAX
DIGITAL INPUTS				
Input HIGH Voltage at $\overline{RD}$, $\overline{CS}$ Inputs	V_{IH}		3	V MIN
Input LOW Voltage at $\overline{RD}$, $\overline{CS}$ Inputs	V_{IL}		0.8	V MAX
Input Current $\overline{RD}$, $\overline{CS}$ Inputs	I_{IN}		±1	μA MAX
Input HIGH Voltage Clock Input	V_{IH}		3	V MIN
Input LOW Voltage Clock Input	V_{IL}		0.4	V MAX
Input HIGH Current Clock Input	I_{IH}		2	mA MAX
Input LOW Current Clock Input	I_{IL}		1	μA MAX

WAFER TEST LIMITS at V_{DD} = +5V, V_{REF} = −10.000V, AGND = DGND = 0V, T_A = +25°C, unless otherwise noted. (Continued)

PARAMETER	SYMBOL	CONDITIONS	PM-7574G LIMIT	UNITS
DIGITAL OUTPUTS				
Output HIGH Voltage BUSY, DB0-7	V_{OH}	I_{SOURCE} = 40μA	4	VMIN
Output LOW Voltage BUSY, DB0-7	V_{OL}	I_{SINK} = 1.6mA	0.4	VMAX
Floating Leakage Current	I_{LKG}		1	μA
POWER REQUIREMENTS				
Standby Current	I_{DD}	V_{DD} = +4.75V to 5.25V	5	mA MAX
TIMING				
Conversion Time	$t_{CONVERT}$	Static RAM Mode, External Clock, f = 550kHz	15	μs MAX

NOTE:
Electrical tests are performed at wafer probe to the limits shown. Due to variations in assembly methods and normal yield loss, yield after packaging is not guaranteed for standard product dice. Consult factory to negotiate specifications based on dice lot qualification through sample lot assembly and testing.

GENERAL CIRCUIT INFORMATION

The PM-7574 is an 8-bit analog-to-digital converter which uses a successive approximation technique to convert an unknown analog input into a digital code output. The control logic inputs allow easy interface to most microprocessors while three-state outputs allow direct connection to the data bus. Most applications require only passive RC clock components, a −10V reference, and a +5V power supply. The RC-timed internal clock may be used, or an external clock may be applied to the ADC to maximize performance.

When a Start Conversion command is applied to the $\overline{CS}$ or $\overline{RD}$ inputs (see Operating Descriptions for details), $\overline{BUSY}$ goes LOW indicating a conversion in progress. $\overline{BUSY}$ may be used as an interrupt to halt the controlling microprocessor during conversion or may be polled to prevent premature data reads.

Starting with the most significant bit (MSB), each successive bit in the DAC is turned on (see Figure 1). The comparator then decides if the DAC output is less than or greater than the signal being converted, and that bit is latched on or off, respectively, before proceeding to the next lower bit and repeating the cycle. When all eight bits have been tested, $\overline{BUSY}$ goes HIGH, signaling a completed conversion.

Under control of the $\overline{RD}$ input, the three-state data outputs (D0-D7) change from high-impedance to presenting the new conversion results to the data bus. Following the data read, $\overline{RD}$ returns HIGH resetting the SAR to 1000 0000 and preparing the ADC for its next conversion.

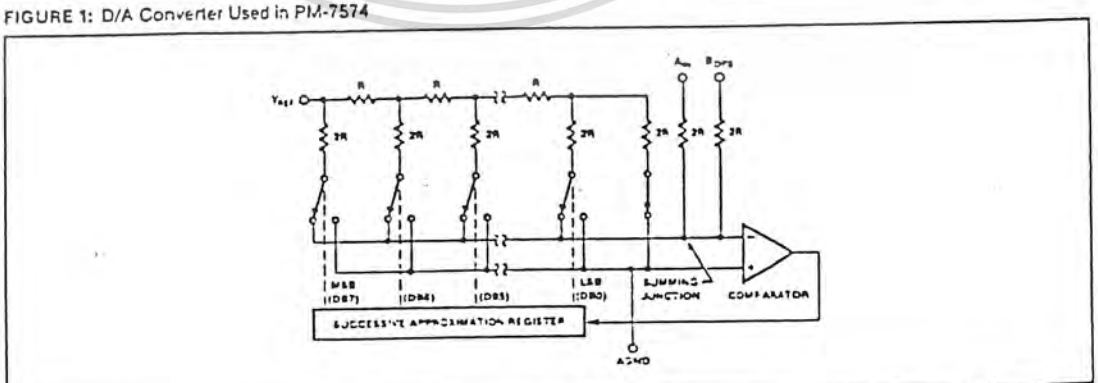
APPLICATIONS INFORMATION

The PM-7574 may be interfaced as if it were a static RAM, a ROM, or a slow-memory device. Each of these interface modes has its own timing and software requirements as described below. These requirements must be rigidly met, as improper timing may cause the PM-7574 to change modes.

HOW TO CHOOSE AN OPERATING MODE

The static-RAM interface mode offers advantages in a tightly controlled hardware and software environment, where the relationship between WRITE and READ instruction pairs is certain. As long as minimum timing is satisfied, converted data may be read at any convenient time after conversion. The use of separate commands to start a conversion, and then read the results, is conceptually easy. However, if the software is subject to uncontrolled modifications, then the paired relationship between WRITE and READ instructions may be lost. The

FIGURE 1: D/A Converter Used in PM-7574



resulting software bugs may result in converted data of unknown age, or altogether invalid data being read.

By contrast, the ROM mode may be more resistant to software bugs. As long as minimum timing is satisfied, each READ instruction obtains new, valid data. However, since the data output at any previous READ instruction is obtained from a conversion performed just after the previous READ instruction, data may be out-of-date. To be sure of obtaining up-to-date data, READ instructions may be coded in pairs (with some NOPs between them); use only the data from the second READ in each pair. The first READ starts the conversion, acting as a substitute for the static-RAM mode WRITE command; the second READ gets the results. The advantage of the ROM mode is the use of a single command, rather than the alternating READ-WRITE required by static-RAM mode.

The slow-memory mode is the simplest mode of all. It is the method of choice where compact coding is essential, or where software bugs are a hazard. In this mode, a single READ instruction will initiate a data conversion, interrupt the microprocessor until completion (WAIT states are introduced), then read the results. If the system throughput tolerates WAIT states, and the hardware is correct, then the slow-memory mode is virtually immune to subsequent software modifications.

OPERATING DESCRIPTION: STATIC-RAM MODE

In this mode, input $\overline{CS}$ is derived from the PM-7574 address decoder, and input $\overline{RD}$ is derived from an active-LOW memory READ signal. (See Figure 2.)

To start a conversion, execute a memory WRITE to the PM-7574. The completed conversion data is obtained by executing a memory READ to the PM-7574. During conversion, output $\overline{BUSY}$ will be LOW. Do not attempt to read data until $\overline{BUSY}$ returns HIGH. The required minimum time between WRITE and READ is usually obtained by including one or more NOP or other program instructions. The use of branch or conditional commands between the WRITE and READ instructions is not recommended due to the possibility of software bugs.

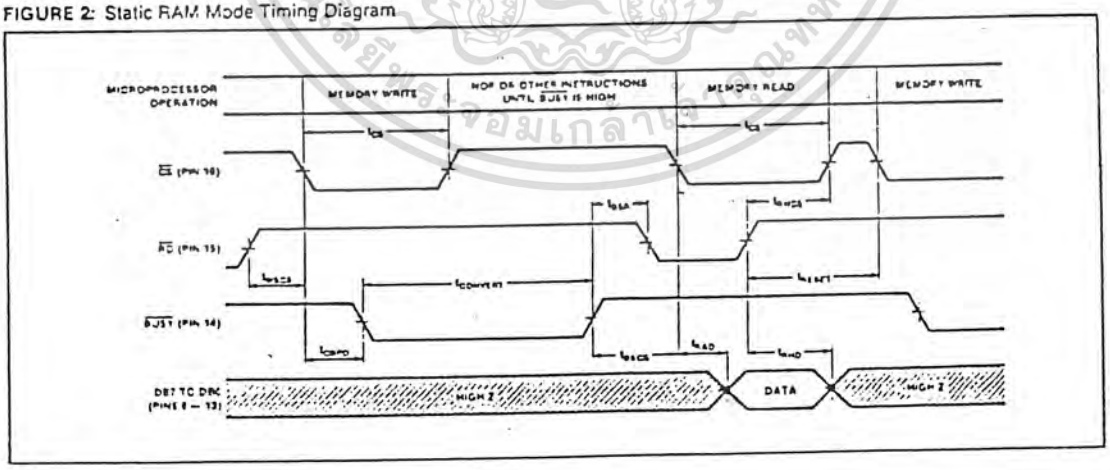
It is important that the WRITE and READ commands be alternately executed. A WRITE instruction has no effect unless the results of the previous WRITE have already been read. Once data has been read, the PM-7574 is internally reset. In other words, two or more READ operations cannot be used in succession, since only the first READ will produce valid data. A new conversion must be started using WRITE, and the conversion must be completed, before a new READ will produce valid data.

TABLE 1: Truth Table, Static RAM Mode

INPUTS $\overline{CS}$	INPUTS $\overline{RD}$	OUTPUTS $\overline{BUSY}$	OUTPUTS DB7-DB0	PM-7574 OPERATION
↓	H	↓	HIGH-Z	Start Convert (Write Cycle)
L	↓	H	HIGH-Z to DATA	Read Data (Read Cycle)
L	↑	H	DATA to HIGH-Z	Reset Converter
H	X (Note 1)	X	HIGH-Z	No Effect (Not Selected)
L	H	L	HIGH-Z	No Effect (Converter Busy)
L	↓	L	HIGH-Z	No Effect (Converter Busy)
L	↑ (Note 1)	L	HIGH-Z	Conversion Error Not Allowed

NOTE 1: If $\overline{RD}$ goes LOW to HIGH, the ADC is internally reset, regardless of the states of $\overline{CS}$ or $\overline{BUSY}$.

FIGURE 2: Static RAM Mode Timing Diagram



OPERATING DESCRIPTION: ROM MODE

In ROM mode, input $\overline{CS}$ is tied LOW, and input $\overline{RD}$ is derived from the PM-7574 address decoder. To satisfy timing, it is recommended that the decoder be enabled by a system MEMRD (8080), VMA (6800), or similar strobe. (See Figure 3.)

TABLE 2: Truth Table, ROM Mode

INPUTS $\overline{CS}$	INPUTS $\overline{RD}$	OUTPUTS $\overline{BUSY}$	OUTPUTS DB7-DB0	PM-7574 OPERATION
L	↓	H	HIGH-Z to DATA	Read Data
L	↑	↓	DATA to HIGH-Z	Reset and Start New Conversion
L	↓	L	HIGH-Z	No Effect (Converter Busy)
L	↑ (Note 1)	L	HIGH-Z	Conversion Error Not Allowed

NOTE 1: If $\overline{RD}$ goes LOW to HIGH, the ADC is internally reset, regardless of the states of $\overline{CS}$ or $\overline{BUSY}$.

In ROM mode, data is read by executing a READ instruction to the PM-7574 address. At the conclusion of the READ instruction, the PM-7574 automatically resets itself and then proceeds to perform a new data conversion. Output $\overline{BUSY}$ is LOW during conversion. A new READ instruction to the PM-7574 must not be executed until $\overline{BUSY}$ returns HIGH. This requirement may be met by inserting NOP or other program instructions between consecutive READ operations. Conditional or branch instructions may be used, but keep in mind that data may become out-of-date if excessive time elapses between consecutive READ instructions.

OPERATING DESCRIPTION: SLOW-MEMORY MODE

The slow-memory mode is intended for systems in which the PM-7574 $\overline{BUSY}$ output is used as an interrupt to force the microprocessor into WAIT states during data conversion.

In slow-memory mode, inputs $\overline{CS}$ and $\overline{RD}$ are tied together. The common $\overline{RD}$ and $\overline{CS}$ signal is derived from the PM-7574 address decoder. To satisfy the timing requirements, it is advisable to latch the address using ALE (8085) or SYNC (8080). For 8080 or 8085-based systems, connect the microprocessor READY input to the PM-7574 $\overline{BUSY}$ output. (See Figure 4.)

FIGURE 3: ROM Mode Timing Diagram ($\overline{CS}$ Held LOW)

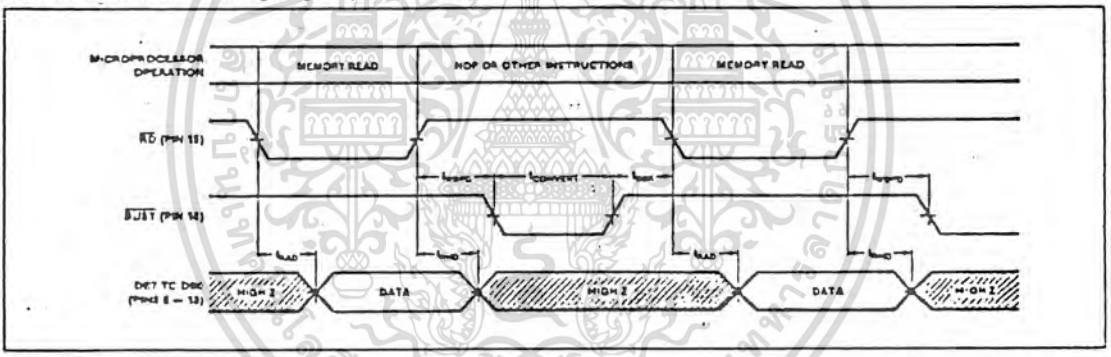


FIGURE 4: Slow-Memory Mode Timing Diagram ($\overline{CS}$ and $\overline{RD}$ Tied Together)

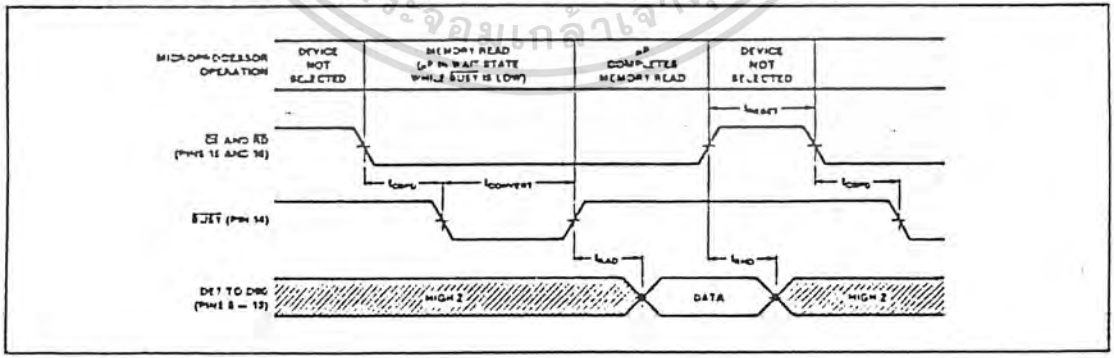


TABLE 3: Truth Table, Slow-Memory Mode

INPUTS $\overline{CS}$ & $\overline{RD}$	OUTPUTS $\overline{BUSY}$	OUTPUTS DB7-DB0	PM-7574 OPERATION
H	H	HIGH-Z	No Effect (Not Selected)
↓	↓	HIGH-Z	Start Conversion
L	L	HIGH-Z	Conversion in Progress. µP in WAIT State
L	↑	HIGH-Z to DATA	Conversion Complete. Read Data
↑ (Note 1)	H	DATA to HIGH-Z	Reset and Deselect Converter

NOTE 1: If $\overline{RD}$ goes LOW to HIGH, the ADC is internally reset, regardless of the states of $\overline{CS}$ or $\overline{BUSY}$.

Do not execute a WRITE instruction at the PM-7574 address when in slow-memory mode, since bus conflicts will arise. In some architectures, an accidental WRITE instruction may be locked out in hardware, by proper strobing of the PM-7574 address decoder.

INITIALIZATION

In all operating modes, the PM-7574 is initialized by executing a READ instruction to the PM-7574 address. The data obtained should be ignored.

CLOCK OSCILLATOR

The PM-7574 may be used with its internal asynchronous clock oscillator. An external resistor and capacitor are required. Typical values are R = 150kΩ and C = 100pF, for conversion times in the 15µs range. For applications in which the fastest conversion times are required, an external clock is recommended. The external clock must be gated by the use of a 74C125-type three-state buffer, with an output pullup resistor. Optimum conversion accuracy is obtained when $\overline{CS}$ goes LOW on a positive clock edge. The maximum external clock frequency is 550kHz. (See Figure 5 and 6.)

FIGURE 5: Using the Internal Clock Oscillator

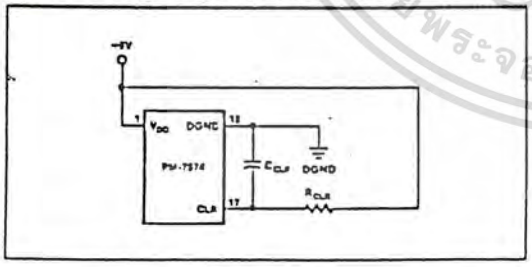
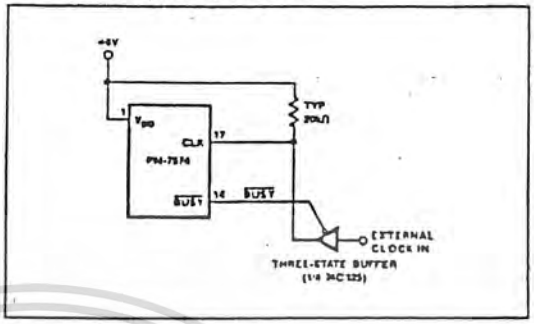


FIGURE 6: Using an External Clock



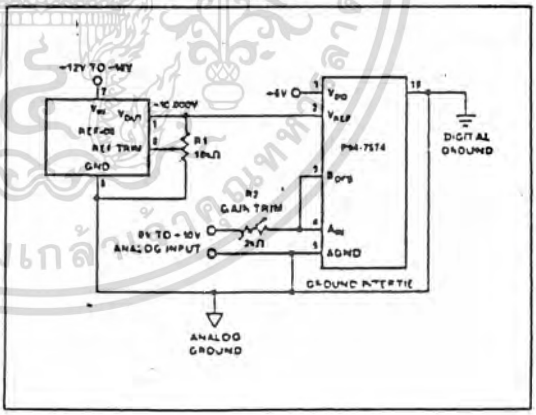
REFERENCE VOLTAGE

A negative reference voltage must be applied to the PM-7574 V_{REF} input. Optimum full-scale accuracy is obtained using −10.00V, although V_{REF} may be −5.00V, −10.24V, or other voltages within its specified range.

Over the full temperature range, optimum gain accuracy is obtained when the input to the V_{REF} pin is from a low-impedance source. A resistor or trimmer may be used in series with the V_{REF} pin, but this trim technique is not as accurate as a low-impedance source. (See Figure 7.)

For a cost-effective −10.00V or −10.24V reference with excellent accuracy and low temperature coefficient, ask for PMI's REF-08. Consult your sales representative for availability.

FIGURE 7: Unipolar Operation



ANALOG INPUT VOLTAGE

The PM-7574 unipolar operation is obtained when the analog input voltage is between 0V and $|V_{REF}|$. With the A_{IN} and B_{OFS} pins tied together, input 0V will correspond to code 0000 0000, and input full-scale will correspond to code 1111 1111.

Bipolar operation is obtained by using the B_{OFS} input to offset the A_{IN} input voltage. For example, with $V_{REF} = -10V$, an offset voltage of +10V may be applied to B_{OFS}. The analog signal range will then be −10V to +10V at A_{IN}. Code 0000 0000 will correspond to −10V, and positive full scale will be code 1111 1111. Calibration may be performed using trimmers in series with A_{IN} and B_{OFS}. (See Figure 8).

Another method of obtaining bipolar operation is to use an op amp with gain = −1/2, to sum the analog signal with the reference voltage. With a −10V reference and −10V to +10V analog signal, the op amp output will then be 0V to +10V. This signal is then treated as an ordinary unipolar input to the PM-7574. With this arrangement, input +10V corresponds to code 0000 0000, and negative full-scale corresponds to code 1111 1111.

UNIPOLAR BINARY OPERATION

Figure 7 shows the analog circuit connections for unipolar operation. The REF-08 supplies the necessary −10V reference input.

Calibration for offset should be made before gain calibration is attempted.

Offset calibration must be performed in the signal conditioning circuitry which drives the A_{IN} input.

To adjust offset:

1) Apply −39.1mV (1 LSB) to the input of the buffer amplifier driving A_{IN}.
2) While performing continuous conversions, adjust the buffer amplifier's offset adjustment potentiometer until DB7 to DB1 are LOW and DB0 (LSB) flickers.

Following offset calibration, full scale gain can be calibrated:

1) Apply −9.961V to the input of the buffer amplifier.
2) While performing continuous conversions, adjust the reference trim pot until DB7 to DB1 are HIGH, and DB0 (LSB) flickers.

BIPOLAR OPERATION

Offset Binary—Figure 8 shows a circuit for offset binary bipolar operation. Offset correction should be made at the buffer amplifier driving A_{IN}. Gain error correction should be accomplished by adjusting V_{REF}.

To calibrate this circuit:

1) Adjust R1 until $V_{REF} = -10.00V$.
2) Adjust R2 and R3 to their mid-points.
3) Apply +10.000V to the input buffer amplifier.
4) While performing continuous conversions, adjust R2 until DB7 to DB1 are LOW and DB0 (LSB) flickers.
5) Ground the input of the input buffer circuit.
6) While performing continuous conversions, adjust R3 until the ADC's output code flickers between 0111 1111 and 1000 0000.
7) Apply −10.000V to the signal input.
8) While performing continuous conversions, adjust R1 until DB7 to DB1 are LOW and the DB0 (LSB) flickers.
9) Apply +9.922V to the signal input.
10) If the ADC output code is not 1111 1110 ±1 bit, repeat the calibration procedure, omitting step 1.

Complementary Offset Binary—Figure 9 shows a complementary offset binary circuit. In this bipolar mode, the −10V to −10V analog input is conditioned to a 0 to +10V signal range for normal unipolar conversion.

In calibrating this circuit, adjust offset before gain.

Offset Adjustment:

1) Adjust R1 until $V_{REF} = -10.000V$.

FIGURE 8: Offset Binary Operation

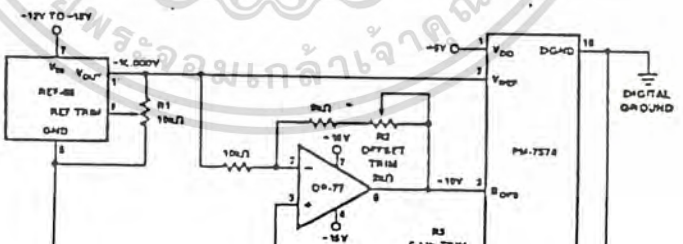
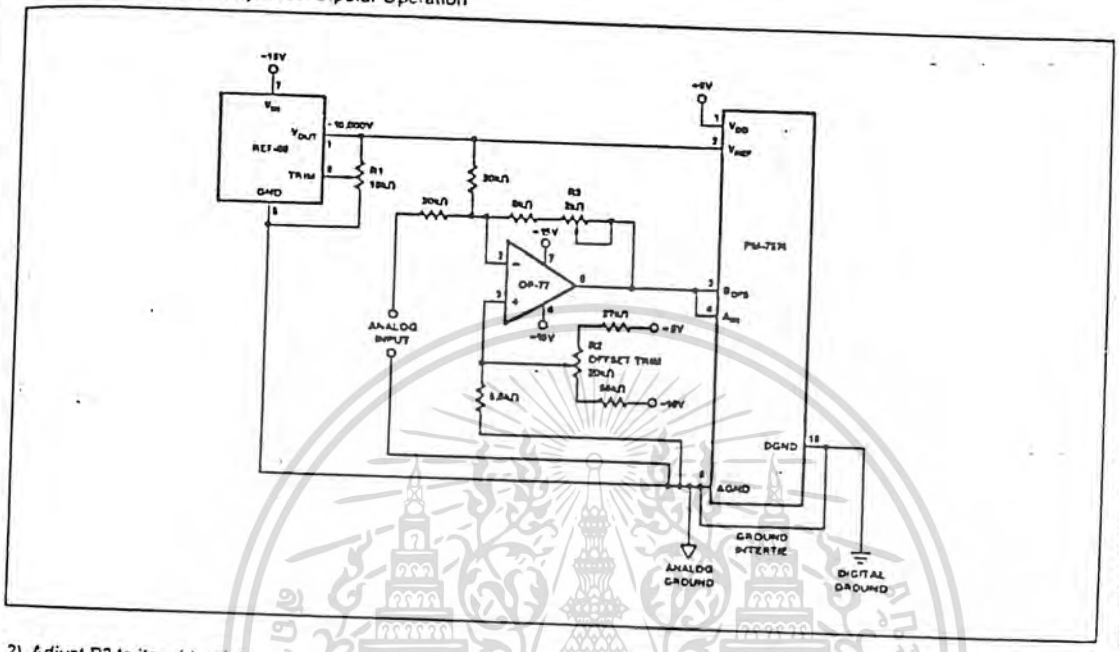


FIGURE 9: Complementary Offset Bipolar Operation



2) Adjust R3 to its mid-point.
3) Adjust R2 until its tap is at 0V.
4) Ground the analog input.
5) While performing continuous conversions, adjust R2 until the ADC output flickers between 0111 1111 and 1000 0000.

Gain Adjustment:
1) Apply +9.922V across the analog input.
2) While performing continuous conversions, adjust R3 until DB7 to DB1 are HIGH and DB0 (LSB) flickers.

DIGITAL CONSIDERATIONS

Control Timing—Fresh data from a recent conversion must be read before beginning a new conversion. Following the data READ, as $\overline{RD}$ goes HIGH, it resets the SAR and clears the data from the previous conversion.

The timing restrictions detailed in the interface timing diagrams must be observed to prevent the PM-7574 from changing interface modes. For example, if $\overline{CS}$ is held LOW too long while in RAM mode, the converter will change to ROM mode and initiate a new conversion.

Logic Deglitching—Unrelated activity on the address bus may cause unexpected glitch inputs to the ADC. The glitches may cause unwanted READs, resets, or conversions. In ROM or RAM modes, these may be avoided by gating the address decode logic with $\overline{RD}$ or $\overline{WR}$ (8080) or VMA (6800). In slow-memory mode, ALE (8085) or SYNC (8080) may be used to latch the address.

Initialization—Following power-up, the SAR is in an unknown state. Executing a memory READ (disregard the data) will reset the ADC.

ANALOG CONSIDERATIONS

Analog Input Impedances—Low impedance sources must be used to drive the V_{REF}, A_{IN}, and B_{OFS} inputs. Excessive source impedances may cause errors due to the loading effects of the inputs' finite impedances.

Ground Management—AGND and DGND pins should be connected at or near the ADC to minimize noise effects. If the two grounds cannot be connected near the ADC, the grounds should be clamped with back-to-back Schottky diodes between the AGND and DGND pins.

Offset Correction—Conversion offset errors may be corrected by counter-offsetting the buffer amplifier driving A_{IN}. This offset correction may be accomplished by applying a correction current to the buffer's summing junction or by tapping a voltage divider sitting between V_{DD} and V_{REF}, and applying this tap voltage to the noninverting input of the buffer.

Ratiometric Operation—The R-2R type DAC in the PM-7574 permits ratiometric operation of the ADC. Performance degradation may, however, occur as V_{REF} varies from -10.000V. This decrease in performance is due to comparator limitations including offset voltage, gain, and input noise.

The PM-7574 uses the reference as a power supply for the comparator to increase speed and accuracy. Reference voltages of a magnitude less than -9V must be avoided for accurate comparator operation. For best accuracy, the use of a 0.1 µF bypass capacitor (Pin 2 to AGND) is recommended.

Power Supply Bypassing—For best accuracy, V_{DD} (Pin 1) should be bypassed to AGND with a 0.1 µF capacitor.

ICL232
+5 Volt Powered Dual RS-232 Transmitter/Receiver

PRELIMINARY

INTERSIL

GENERAL DESCRIPTION

The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two on-board charge pump voltage converters which generate +10V and −10V supplies from the 5V supply.

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300 ohms power-off source impedance. The receivers can handle up to ±30 volts, and have a 3 to 7 kohms input impedance. The receivers also have hysteresis to improve noise rejection.

Typical Applications

Any System Requiring RS-232 Communications Port:

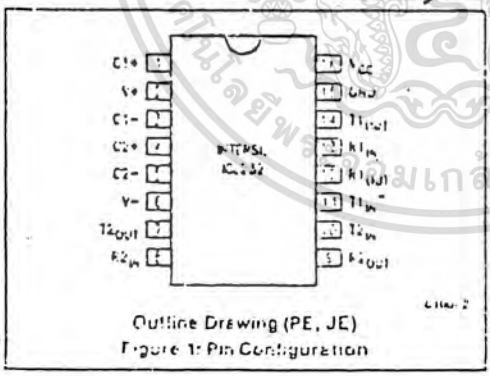
- Computers—Portable and Mainframe
- Peripherals—Printers and Terminals
- Portable Instrumentation
- Modems
- Dataloggers

FEATURES

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Quadrupler
- Low Power Consumption
- ESD Protection >2000V
- 2 Drivers
  - —±10V Output Swing for +5V Input
  - —300 Ohms Power-off Source Impedance
  - —Output Current Limiting
  - —TTL/CMOS Compatible
  - —30 V/µs Maximum Slew Rate
- 2 Receivers
  - —±30V Input Voltage Range
  - —3 to 7 kohms Input Impedance
  - —0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

ORDERING INFORMATION

Part	Temperature Range	Package
ICL232CPE	0°C to +70°C	16 Pin Plastic DIP
ICL232CJE		16 Pin CERDIP
ICL232IPE	−25°C to +85°C	16 Pin Plastic DIP
ICL232IJE		16 Pin CERDIP
ICL232MJE	−55°C to +125°C	16 Pin CERDIP



Outline Drawing (PE, JE)

Figure 1. Pin Configuration

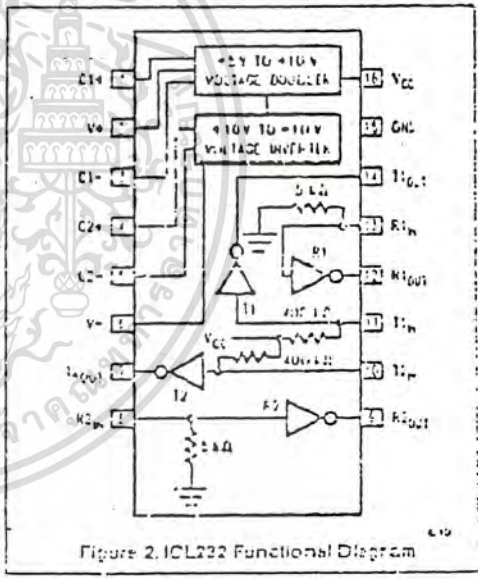


Figure 2. ICL232 Functional Diagram

The information and descriptions contained herein are subject to change without notice. Intersil assumes no responsibility for the use of any circuits described herein and makes no representations that they are free from patent infringement.

ABSOLUTE MAXIMUM RATINGS

V_{CC} to ground ..... + 6 Volts
V^+ to ground ..... + 12 Volts
V^- to ground ..... − 12 Volts
Input Voltages
$T1_{IN}$, $T2_{IN}$ ..... − 0.3 to (V_{CC} + 0.3V)
$R1_{IN}$, $R2_{IN}$ ..... ± 30V
Output Voltages
$T1_{OUT}$, $T2_{OUT}$ ..... (V^+ + 0.3) to (V^- − 0.3)
$R1_{OUT}$, $R2_{OUT}$ ..... − 0.3 to (V_{CC} + 0.3V)
Short Circuit Duration
$T1_{OUT}$, $T2_{OUT}$ ..... Continuous
Continuous Total Power Dissipation (T_A = 25°C)
CERDIP Package 500mW
derate 8.5 mW/°C above 70°C
Plastic Package 375mW
derate 7.0 mW/°C above 70°C
Storage Temperature Range ..... − 65°C to + 150°C
Lead Temperature (Soldering, 10 sec) ..... + 300°C
Operating Temperature Range
ICL232C ..... 0°C to + 70°C
ICL232I ..... − 25°C to + 85°C
ICL232M ..... − 55°C to + 125°C

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

ELECTRICAL CHARACTERISTICS

Test Conditions: V_{CC} = + 5V, T_A = operating temperature range, Test Circuits as in Figure 3 (unless otherwise specified)

Symbol	Parameter	Test Conditions	Limits			Units
			Min	Typ	Max	
T_{OUT}	Transmitter Output Voltage Swing	$T1_{IN}$, $T2_{IN}$ Loaded with 3 kΩ to ground	± 5	± 9	± 10	V
I_{CC}	Power Supply Current	Outputs Unloaded, T_A = 25°C		5	10	mA
V_{IL}	TIN Input Logic Low				0.8	V
V_{IH}	TIN Input Logic High		2.0			V
I_I	Logic Pullup Current	$T1_{IN}$, $T2_{IN}$ = 0V		15	200	µA
V_I	RS-232 Input Voltage Range		− 30		+ 30	V
R_I	Receiver Input Impedance	V_{IN} = ± 3V, ± 25V	3.0	5.0	7.0	kΩ
V_{IN}	Receiver Input Low Threshold	V_{CC} = 5.0V, T_A = 25°C	0.8	1.2		V
V_{IH}	Receiver Input High Threshold	V_{CC} = 5.0V, T_A = 25°C		1.7	2.4	V
V_{HY}	Receiver Input Hysteresis		0.2	0.5	1.0	V
V_O	TTL/CMOS Receiver Output Voltage Low	I_{OUT} = 3.2mA		0.1	0.4	V
V_{OH}	TTL/CMOS Receiver Output Voltage High	I_{OUT} = − 1.0mA	3.5	4.6		V
t_{PR}	Propagation Delay	RS-232 to TTL or TTL to RS-232		0.5		µs
SR	Instantaneous Slew Rate	C_L = 10pF, R_L = 3 kΩ, T_A = 25°C (Note 1, 2)			30	V/µs
SR_T	Transition Region Slew Rate	R_L = 3 kΩ, C_L = 2500 pF Measured from + 3V to − 3V or − 3V to + 3V		3		V/µs
R_{OUT}	Output Resistance	V_{CC} = V^+ = V^- = 0V, V_{OUT} = ± 2V	300			Ω
I_{SC}	RS-232 Output Short Circuit Current	$T1_{OUT}$, $T2_{OUT}$ shorted to GND		± 10		mA

1. TTL Input is square wave
2. See Figure 3 for definition

ICL232

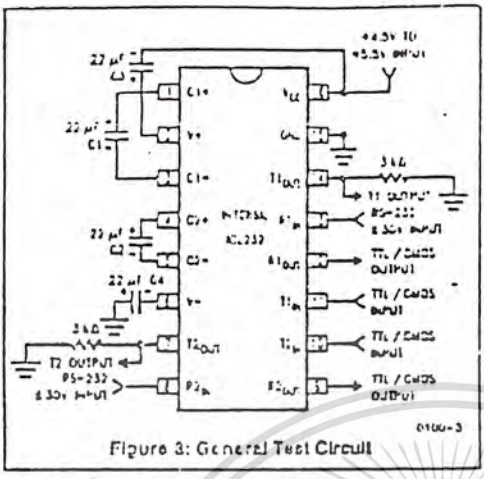
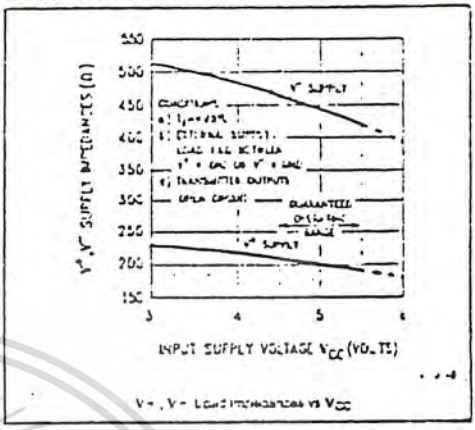


Figure 3: General Test Circuit

TYPICAL PERFORMANCE CHARACTERISTICS



V+, V- Load Impedance vs V_{CC}

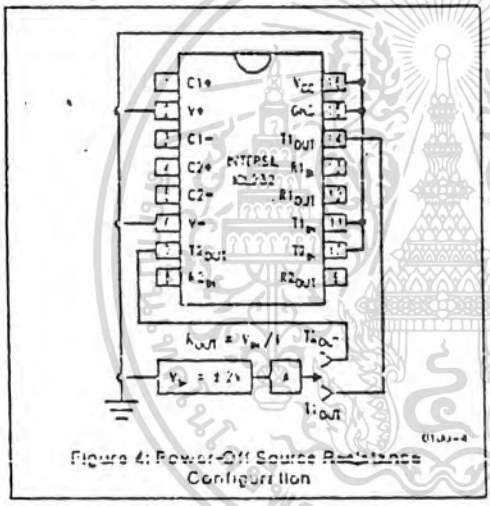
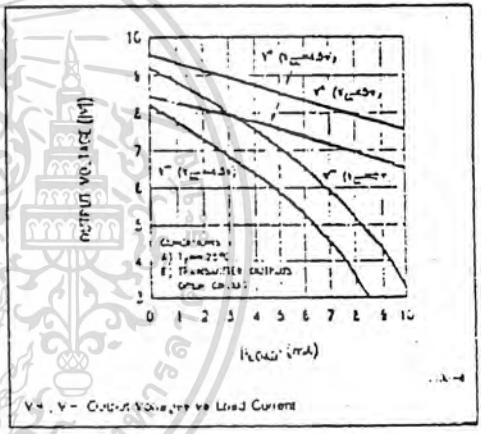


Figure 4: Power-Off Source Resistance Configuration



V+, V- Output Voltages vs Load Current

INTERSIL DOES NOT ASSUME ANY RESPONSIBILITY FOR USE OF ANY CIRCUITRY DESCRIBED NO CIRCUIT PATENT LICENSES ARE IMPLIED AND INTERSIL RESERVES THE RIGHT TO MAKE CHANGES AT ANY TIME IN ORDER TO IMPROVE DESIGN AND SUPPLY THE BEST PRODUCT POSSIBLE

DETAILED DESCRIPTION

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS-232C specifications and features low power consumption. The functional diagram (Figure 2) illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage quadrupler, dual transmitters, and dual receivers.

Voltage Converter

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and −10V. During phase one of the clock, capacitor C1 is charged to V_{CC}. During phase two, the voltage on C1 is added to V_{CC}, producing a signal across C2 equal to twice V_{CC}. At the same time, C3 is also charged to $2V_{CC}$, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to $-2V_{CC}$. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V^+) is approximately 200 ohms, and the output impedance of the inverter (V^-) is approximately 450 ohms. The test circuit (Figure 3) uses 22 µF capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V^+ and V^- supplies.

Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V_{CC}, or 1.3V for V_{CC} = 5V. A logic 1 at the input results in a voltage of between −5V and V^- at the output, and a logic 0 results in a voltage between +5V and (V^+ −0.6V). Each transmitter input has an internal 400 kilohm pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of ±5V minimum with the worst case conditions of both transmitters driving 3kohm minimum load impedance, V_{CC} = 4.5V, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/µs. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300 ohms with ±2V applied to the outputs and V_{CC} = 0V.

Receivers

The receiver inputs accept up to ±30V while presenting the required 3 to 7 kilohms input impedance even if the power is off (V_{CC} = 0V). The receivers have a typical input threshold of 1.3V which is within the ±3V limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V_{CC}. The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and −30V. The receivers feature 0.5V hysteresis to improve noise rejection.

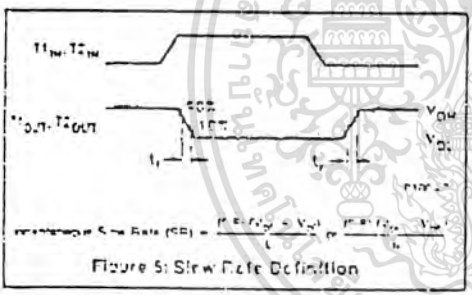


Figure 5: Slew Rate Definition

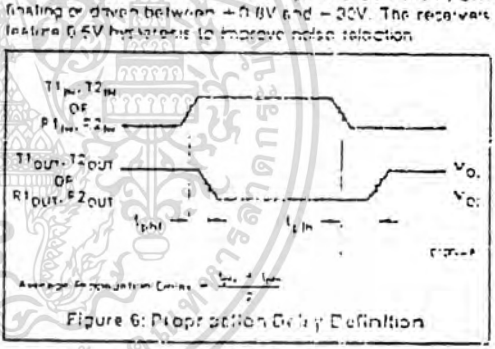


Figure 6: Propagation Delay Definition

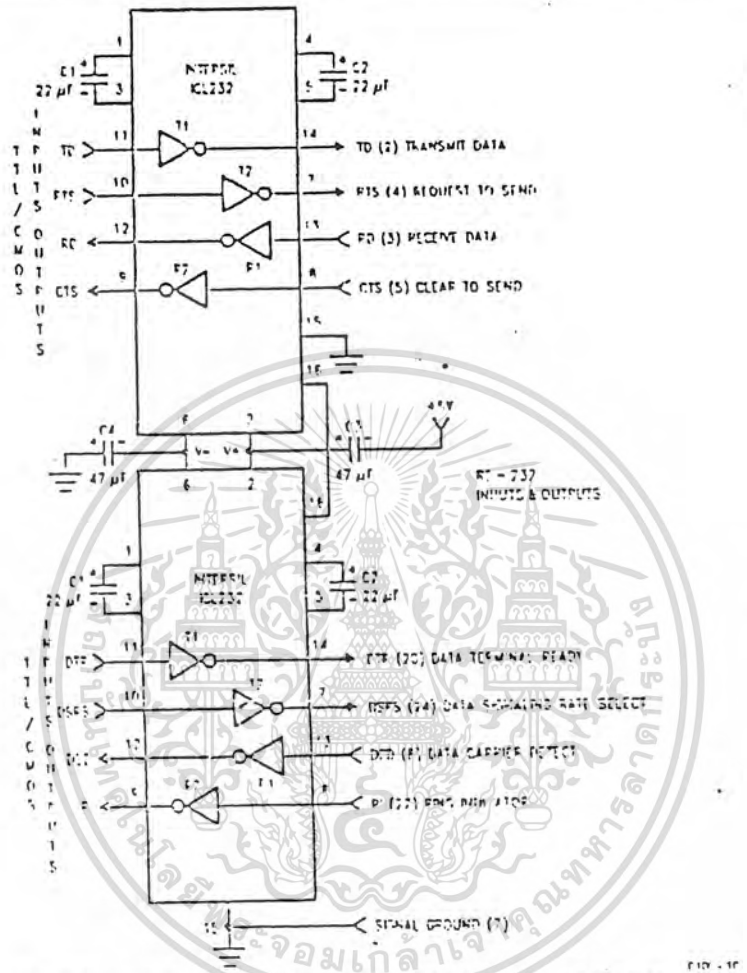


Figure 8. Combining 2 ICL232 for 4 Pairs of RS-232 Inputs and Outputs

National Semiconductor Corporation

DAC0800/DAC0801/DAC0802 8-Bit Digital-to-Analog Converters

General Description

The DAC0800 series are monolithic 8-bit high-speed current-output digital-to-analog converters (DAC) featuring typical settling times of 100 ns. When used as a multiplying DAC, monotonic performance over a 40 to 1 reference current range is possible. The DAC0800 series also features high compliance complementary current outputs to allow differential output voltages of 20 Vp-p with simple resistor loads as shown in *Figure 1*. The reference-to-full-scale current matching of better than ±1 LSB eliminates the need for full-scale trims in most applications while the nonlinearities of better than ±0.1% over temperature minimizes system error accumulations.

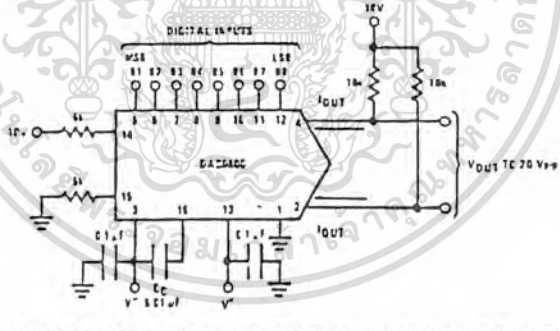
The noise immune inputs of the DAC0800 series will accept TTL levels with the logic threshold pin, V_{LC}, grounded. Changing the V_{LC} potential will allow direct interface to other logic families. The performance and characteristics of the device are essentially unchanged over the full ±4.5V to ±18V power supply range; power dissipation is only 33 mW with ±5V supplies and is independent of the logic input states.

The DAC0800, DAC0802, DAC0800C, DAC0801C and DAC0802C are a direct replacement for the DAC-08, DAC-08A, DAC-08C, DAC-08E and DAC-08H, respectively.

Features

- Fast settling output current — 100 ns
- Full scale error — ±1 LSB
- Nonlinearity over temperature — ±0.1%
- Full scale current drift — ±10 ppm/°C
- High output compliance — −10V to +18V
- Complementary current outputs
- Interface directly with TTL, CMOS, PMOS and others
- 2 quadrant wide range multiplying capability
- Wide power supply range — ±4.5V to ±18V
- Low power consumption — 33 mW at ±5V
- Low cost

Typical Applications



TL/H/5585–1

FIGURE 1. ±20 V_{P-P} Output Digital-to-Analog Converter (Note 4)

Ordering Information

Non-Linearity	Temperature Range	Order Numbers				
		J Package (J16A)*		N Package (N16A)*		SO Package (M16A)
±0.1% FS	−55°C ≤ T_A ≤ +125°C	DAC0802LJ	DAC-08AQ			
±0.1% FS	0°C ≤ T_A ≤ +70°C	DAC0802LCJ	DAC-08HQ	DAC0802LCN	DAC-08HP	DAC0802LCM
±0.19% FS	−55°C ≤ T_A ≤ +125°C	DAC0800LJ	DAC-08Q			
±0.19% FS	0°C ≤ T_A ≤ +70°C	DAC0800LCJ	DAC-08EQ	DAC0800LCN	DAC-08EP	DAC0800LCM
±0.39% FS	0°C ≤ T_A ≤ +70°C	DAC0801LCJ	DAC-08CQ	DAC0801LCN	DAC-08CP	DAC0801LCM

*Devices may be ordered by using either order number.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage ($V^+ - V^-$)	± 18V or 36V
Power Dissipation (Note 2)	500 mW
Reference Input Differential Voltage (V14 to V15)	V^- to V^+
Reference Input Common-Mode Range (V14, V15)	V^- to V^+
Reference Input Current	5 mA
Logic Inputs	V^- to V^- plus 36V
Analog Current Outputs ($V_S^- = -15V$)	4.25 mA
ESD Susceptibility (Note 3)	TBD V
Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C

Operating Conditions (Note 1)

	Min	Max	Units
Temperature (T_A)			
DAC0802L	−55	+125	°C
DAC0800L	−55	+125	°C
DAC0800LC	0	+70	°C
DAC0801LC	0	+70	°C
DAC0802LC	0	+70	°C

Electrical Characteristics The following specifications apply for $V_S = \pm 15V$, $I_{REF} = 2$ mA and $T_{MIN} \le T_A \le T_{MAX}$ unless otherwise specified. Output characteristics refer to both I_{OUT} and $\overline{I_{OUT}}$.

Symbol	Parameter	Conditions	DAC0802L/ DAC0802LC			DAC0800L/ DAC0800LC			DAC0801LC			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Resolution		8	8	8	8	8	8	8	8	8	Bits
	Monotonicity		8	8	8	8	8	8	8	8	8	Bits
	Nonlinearity				± 0.1			± 0.19			± 0.39	%FS
	Settling Time	To ± ½ LSB, All Bits Switched "ON" or "OFF", $T_A = 25°C$		100	135					100	150	ns
		DAC0800L					100	135				ns
		DAC0800LC					100	150				ns
t_{PLH}, t_{PHL}	Propagation Delay	$T_A = 25°C$										
	Each Bit			35	60		35	60		35	60	ns
	All Bits Switched			35	60		35	60		35	60	ns
T_{CS}	Full Scale Tempco			± 10	± 50		± 10	± 50		± 10	± 50	ppm/°C
V_O	Output Voltage Compliance	Full Scale Current Change < ½ LSB, $R_{OUT} > 20$ MΩ Typ	−10		18	−10		18	−10		18	V
I_{FS}	Full Scale Current	$V_{REF} = 10.000V$, R14 = 5.000 kΩ R15 = 5.000 kΩ, $T_A = 25°C$	1.984	1.992	2.000	1.94	1.99	2.04	1.94	1.99	2.04	mA
	Full Scale Symmetry	$I_{FS4} - I_{FS2}$		± 0.5	± 4.0		± 1	± 8.0		± 2	± 16	μA
	Zero Scale Current			0.1	1.0		0.2	2.0		0.2	4.0	μA
I_O	Output Current Range	$V^- = -5V$	0	2.0	2.1	0	2.0	2.1	0	2.0	2.1	mA
		$V^- = -5V$ to $-18V$	0	2.0	4.2	0	2.0	4.2	0	2.0	4.2	mA
	Logic Input Levels											
	Logic "0"	$V_{LC} = 0V$			0.8			0.8			0.8	V
	Logic "1"		2.0			2.0			2.0			V
	Logic Input Current	$V_{LC} = 0V$										
	Logic "0"	$-10V \le V_{IN} \le +0.6V$		−2.0	−10		−2.0	−10		−2.0	−10	μA
	Logic "1"	$2V \le V_{IN} \le +18V$		0.002	10		0.002	10		0.002	10	μA
	Logic Input Swing	$V^- = -15V$	−10		18	−10		18	−10		18	V
V_{TH}	Logic Threshold Range	$V_S = \pm 15V$	−10		13.5	−10		13.5	−10		13.5	V
	Reference Bias Current			−1.0	−3.0		−1.0	−3.0		−1.0	−3.0	μA
	Reference Input Slew Rate	*(Figure 12)*	4.0	8.0		4.0	8.0		4.0	8.0		mA/μs
PSS_+	Power Supply Sensitivity	$4.5V \le V^+ \le 18V$		0.0001	0.01		0.0001	0.01		0.0001	0.01	%/%
PSS_-		$-4.5V \le V^- \le 18V$ $I_{REF} = 1mA$		0.0001	0.01		0.0001	0.01		0.0001	0.01	%/%
	Power Supply Current	$V_S = \pm 5V$, $I_{REF} = 1$ mA		2.3	3.8		2.3	3.8		2.3	3.8	mA
				−4.3	−5.8		−4.3	−5.8		−4.3	−5.8	mA
		$V_S = 5V, -15V$, $I_{REF} = 2$ mA		2.4	3.8		2.4	3.8		2.4	3.8	mA
				−6.4	−7.8		−6.4	−7.8		−6.4	−7.8	mA
		$V_S = \pm 15V$, $I_{REF} = 2$ mA		2.5	3.8		2.5	3.8		2.5	3.8	mA
				−6.5	−7.8		−6.5	−7.8		−6.5	−7.8	mA

Electrical Characteristics (Continued)

The following specifications apply for $V_S = \pm 15V$, $I_{REF} = 2$ mA and $T_{MIN} \le T_A \le T_{MAX}$ unless otherwise specified. Output characteristics refer to both I_{OUT} and $\overline{I_{OUT}}$.

Symbol	Parameter	Conditions	DAC0802L/ DAC0802LC			DAC0800L/ DAC0800LC			DAC0801LC			U
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
P_D	Power Dissipation	$\pm 5V$, $I_{REF} = 1$ mA		33	48		33	48		33	48	
		$5V, -15V$, $I_{REF} = 2$ mA		108	136		108	135		108	136	
		$\pm 15V$, $I_{REF} = 2$ mA		135	174		135	174		135	174	

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

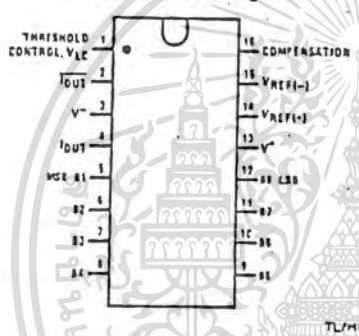
Note 2: The maximum junction temperature of the DAC0800, DAC0801 and DAC0802 is 125°C. For operating at elevated temperatures, devices in the Dual-In-Line J package must be derated based on a thermal resistance of 100°C/W, junction-to-ambient, 175°C/W for the molded Dual-In-Line N package and 100°C/W for the Small Outline M package.

Note 3: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 4: Pin-out numbers for the DAC080X represent the Dual-In-Line package. The Small Outline package pin-out differs from the Dual-In-Line package.

Connection Diagrams

Dual-In-Line Package

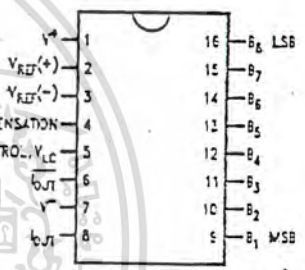


Top View

TL/H/5686–13

See Ordering Information

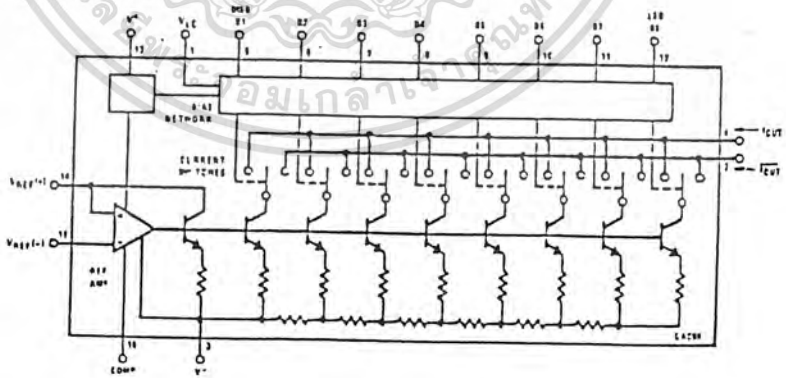
Small Outline Package



Top View

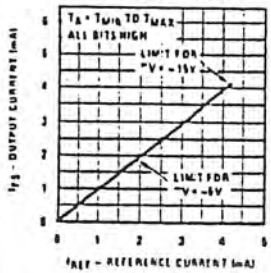
TL/H/5686–14

Block Diagram (Note 4)

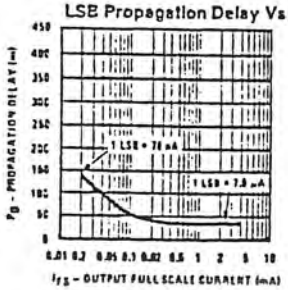


Typical Performance Characteristics

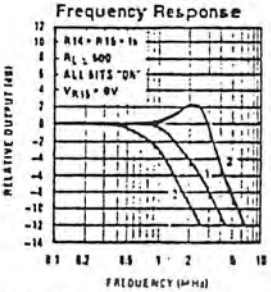
Full Scale Current vs Reference Current



LSB Propagation Delay Vs I_{FS}

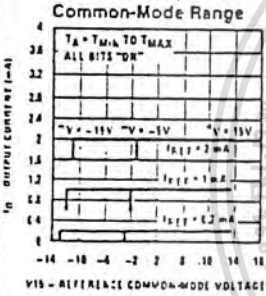


Reference Input Frequency Response



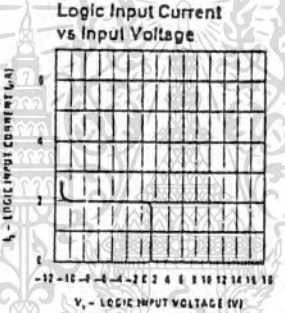
Curve 1: C_C = 15 pF, V_{IN} = 2 Vp-p centered at 1V.
Curve 2: C_C = 15 pF, V_{IN} = 50 mVp-p centered at 200 mV.
Curve 3: C_C = 0 pF, V_{IN} = 100 mVp-p at 0V and applied through 50 Ω connected to pin 14.2V applied to R14.

Reference Amp Common-Mode Range

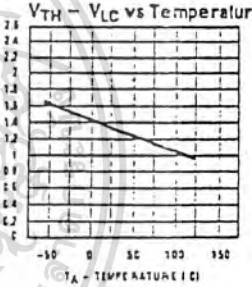


Note: Positive common-mode range is always $(V^+) - 1.5V$

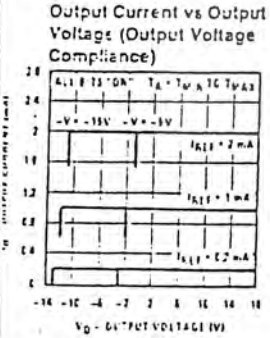
Logic Input Current vs Input Voltage



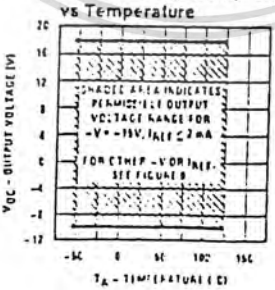
$V_{TH} - V_{LC}$ vs Temperature



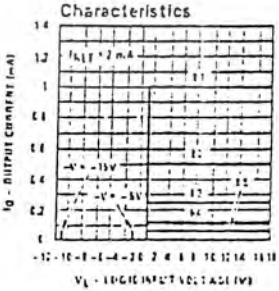
Output Current vs Output Voltage (Output Voltage Compliance)



Output Voltage Compliance vs Temperature



Bit Transfer Characteristics

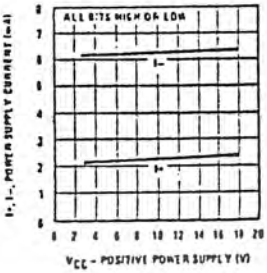


TL/H/5691-3

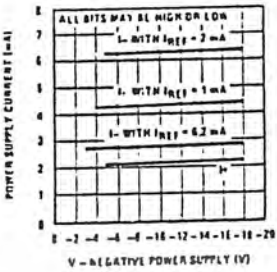
Note: E1-E8 have identical transfer characteristics. Bits are fully switched with less than ½ LSB error at less than ± 100 mV from actual threshold. These switching points are guaranteed to be between 0.8 and 2V over the operating temperature range (V_{LC} = 0V).

Typical Performance Characteristics (Continued)

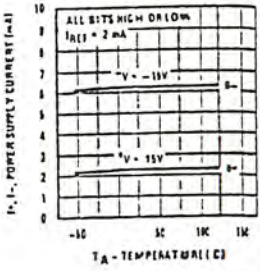
Power Supply Current vs +V



Power Supply Current vs −V

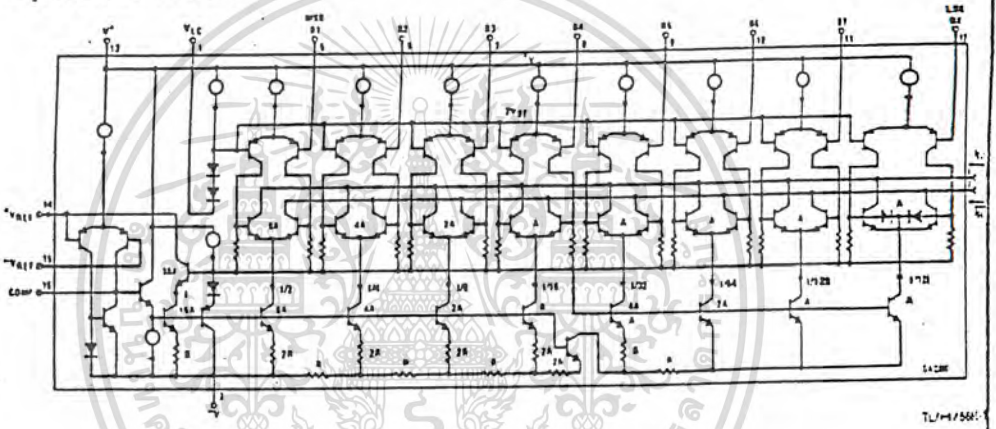


Power Supply Current vs Temperature



TL/H/5695-

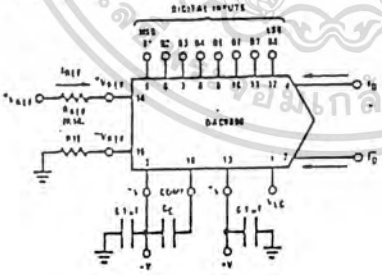
Equivalent Circuit



TL/H/5695-

FIGURE 2

Typical Applications (Continued)



$$I_{FS} = \frac{+V_{REF}}{R_{REF}} \times \frac{255}{256}$$

$I_O + \overline{I_O} = I_{FS}$ for all logic states

For fixed reference, TTL operation, typical values are:

V_{REF} = 10.000V

R_{REF} = 5.000k

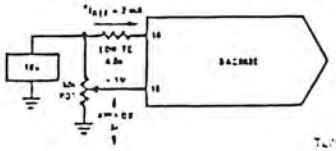
R15 = R_{REF}

C_C = 0.01 µF

V_{LC} = 0V (Ground)

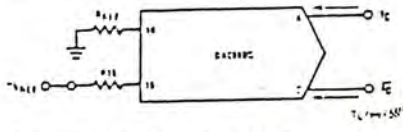
TL/H/5695-5

FIGURE 3. Basic Positive Reference Operation (Note 4)



TL/H/5695-21

FIGURE 4. Recommended Full Scale Adjustment Circuit (Note 4)



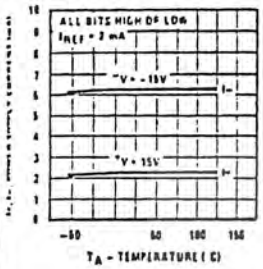
$$I_{FS} = \frac{-V_{REF}}{R_{REF}} \times \frac{255}{256}$$

Note: Represents R15 is for bias current cancellation

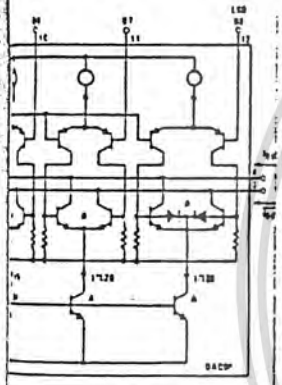
FIGURE 5. Basic Negative Reference Operation (Not

Typical Applications (Continued)

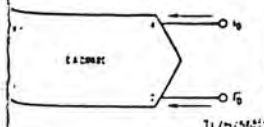
Power Supply Current vs Temperature



TL/H/5686-

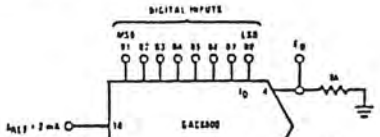


TL/H/5686-



TL/H/5686-

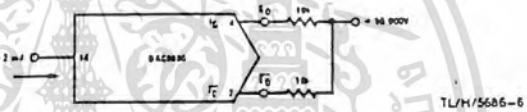
Bipolar Reference Operation (Note 4)



TL/H/5686-17

	B1	B2	B3	B4	B5	B6	B7	B8	I_O mA	$\overline{I_O}$ mA	E_O	$\overline{E_O}$
Full Scale	1	1	1	1	1	1	1	1	1.992	0.000	−9.960	0.000
Full Scale − LSB	1	1	1	1	1	1	1	0	1.984	0.008	−9.920	−0.040
Half Scale + LSB	1	0	0	0	0	0	0	1	1.008	0.984	−5.040	−4.920
Half Scale	1	0	0	0	0	0	0	0	1.000	0.992	−5.000	−4.960
Half Scale − LSB	0	1	1	1	1	1	1	1	0.992	1.000	−4.960	−5.000
Zero Scale + LSB	0	0	0	0	0	0	0	1	0.008	1.984	−0.040	−9.920
Zero Scale	0	0	0	0	0	0	0	0	0.000	1.992	0.000	−9.960

FIGURE 6. Basic Unipolar Negative Operation (Note 4)



TL/H/5686-6

	B1	B2	B3	B4	B5	B6	B7	B8	E_O	$\overline{E_O}$
Pos. Full Scale	1	1	1	1	1	1	1	1	−9.920	+10.000
Pos. Full Scale − LSB	1	1	1	1	1	1	1	0	−9.840	+9.920
Zero Scale + LSB	1	0	0	0	0	0	0	1	−0.080	+0.160
Zero Scale	1	0	0	0	0	0	0	0	0.000	+0.080
Zero Scale − LSB	0	1	1	1	1	1	1	1	+0.080	0.000
Neg. Full Scale + LSB	0	0	0	0	0	0	0	1	+9.920	−9.840
Neg. Full Scale	0	0	0	0	0	0	0	0	+10.000	−9.920

FIGURE 7. Basic Bipolar Output Operation (Note 4)

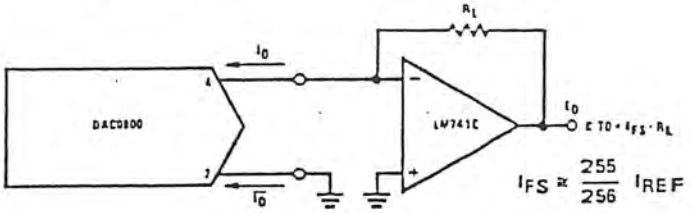
TL/H/5686-16

If $R_1 = \overline{R_1}$ within ± 0.05%, output is symmetrical about ground

	B1	B2	B3	B4	B5	B6	B7	B8	E_O
Pos. Full Scale	1	1	1	1	1	1	1	1	+9.960
Pos. Full Scale − LSB	1	1	1	1	1	1	1	0	+9.880
(+) Zero Scale	1	0	0	0	0	0	0	0	+0.040
(−) Zero Scale	0	1	1	1	1	1	1	1	−0.040
Neg. Full Scale + LSB	0	0	0	0	0	0	0	1	−9.880
Neg. Full Scale	0	0	0	0	0	0	0	0	−9.960

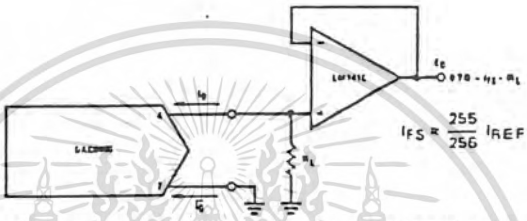
FIGURE 8. Symmetrical Offset Binary Operation (Note 4)

Typical Applications (Continued)



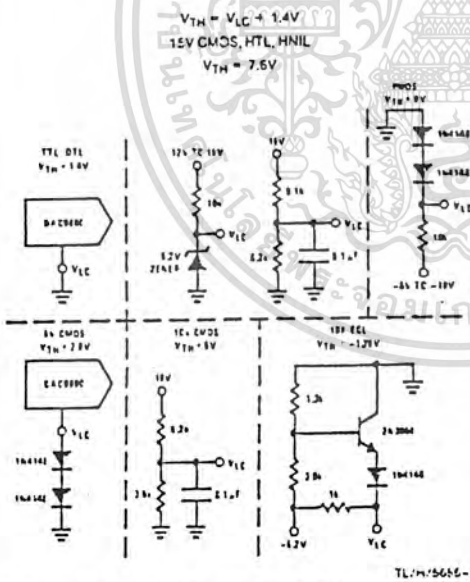
For complementary output (operation as negative logic DAC), connect inverting input of op amp to $\overline{I_O}$ (pin 2), connect I_O (pin 4) to ground.

FIGURE 9. Positive Low Impedance Output Operation (Note 4)



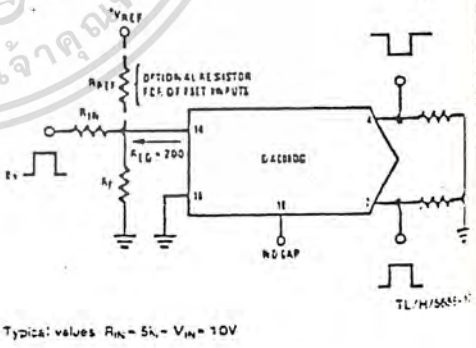
For complementary output (operation as a negative logic DAC) connect non-inverting input of op am to $\overline{I_O}$ (pin 2), connect I_O (pin 4) to ground.

FIGURE 10. Negative Low Impedance Output Operation (Note 4)



Note: Do not exceed negative logic input range of DAC.

FIGURE 11. Interfacing with Various Logic Families

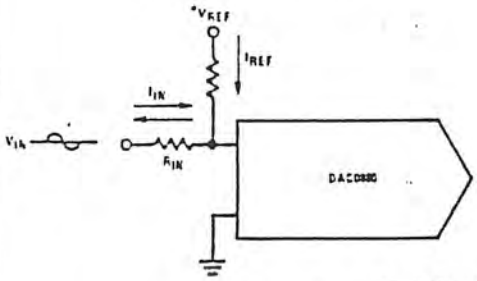


Typical values $R_{IN} = 5k$, $V_{IN} = 10V$

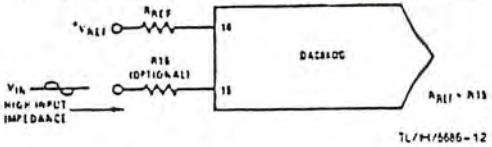
FIGURE 12. Pulsed Reference Operation (Note 4)

Typical Applications (Continued)

(a) $I_{REF} \geq$ peak negative swing of I_{IN}



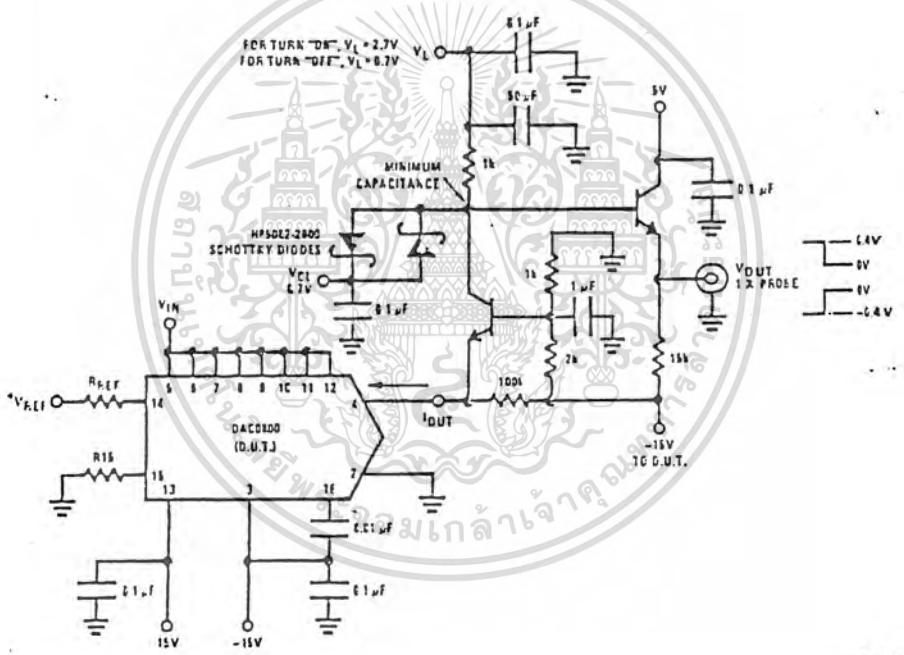
(b) + V_{REF} must be above peak positive swing of V_{IN}



TL/H/5686–12

TL/H/5686–11

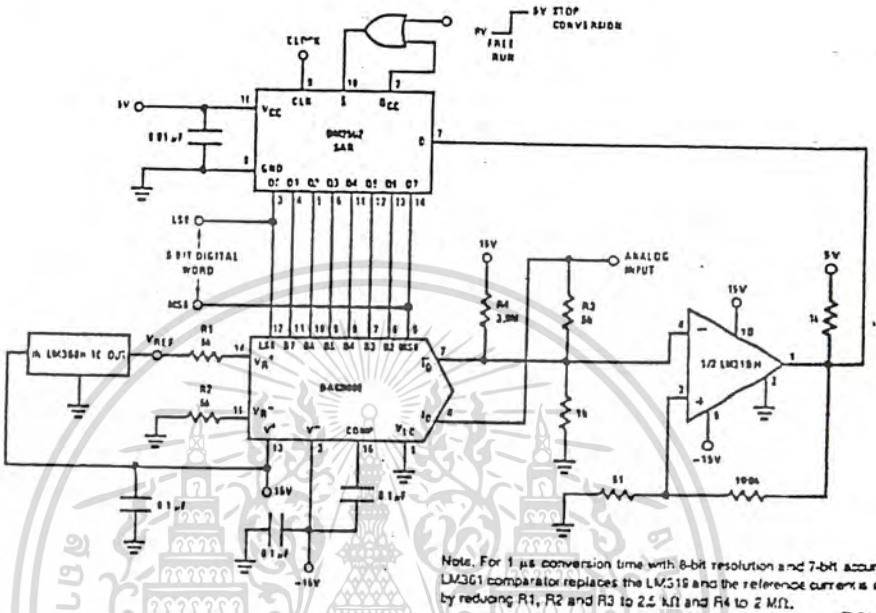
FIGURE 13. Accommodating Bipolar References (Note 4)



TL/H/5686–7

FIGURE 14. Settling Time Measurement (Note 4)

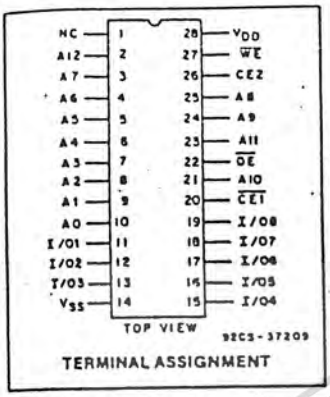
Typical Applications (Continued)



Note. For 1 μs conversion time with 8-bit resolution and 7-bit accuracy, an LM361 comparator replaces the LM319 and the reference current is doubled by reducing R1, R2 and R3 to 2.5 kΩ and R4 to 2 MΩ.

TL/H/5665–8

FIGURE 15. A Complete 2 μs Conversion Time, 8-Bit A/D Converter (Note 4)



TERMINAL ASSIGNMENT

CMOS 8192-Word by 8-Bit LSI Static RAM

Features:

- *Fully static operation*
- *Single power supply: 4.5 V to 5.5 V*
- *All inputs and outputs directly TTL compatible*
- *3-state outputs*
- *Industry standard 28-pin configuration*
- *Input address buffers gated off with chip disable*
- *Fast access time: t_{AA}=150 ns/120 ns (CDM6264-3/CDM6264-4)*
- *Low standby and operating power: I_{DO3}=2 μA typical, I_{OPER2}=40 mA maximum*
- *Data retention voltage: 2 V min.*
- *Operating temperature range (max. rating): 0° to 70°C*

The RCA-CDM6264 is a 8192-word by 8-bit static random-access memory. It is designed for use in memory systems where high-speed, low power and simplicity in use are desirable. This device has common data input and data output and utilizes a single power supply of 4.5 V to 5.5 V.

Either chip enable ($\overline{CE1}$ or CE2), when not valid, will gate off the address and output buffers and power down the chip to minimum standby power with inputs toggling. The output enable ($\overline{OE}$) controls the output buffers to eliminate bus contention.

The CDM6264 is supplied in 28-lead, hermetic, dual-in-line side-brazed ceramic (D suffix) and in 28-lead dual-in-line plastic (E suffix) packages.

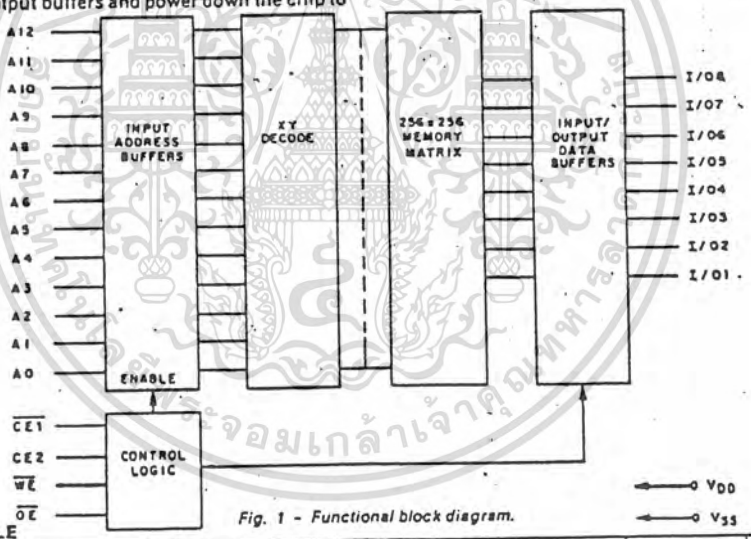


Fig. 1 - Functional block diagram.

TRUTH TABLE

$\overline{CE1}$	CE2	$\overline{OE}$	$\overline{WE}$	A0 TO A12	MODE	DATA I/O	DEVICE CURRENT
H	X	X	X	X	NOT SELECTED	HIGH Z	STANDBY
X	L	X	X	X	NOT SELECTED	HIGH Z	STANDBY
L	H	L	H	STABLE	READ	DATA OUT	ACTIVE
L	H	X	L	STABLE	WRITE	DATA IN	ACTIVE
L	H	H	H	STABLE	OUTPUT DISABLE	HIGH Z	ACTIVE

L = LOW H = HIGH X = H OR L

CDM6264

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD}):
(Voltage referenced to V_{SS} terminal) ..... -0.3 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS ..... -0.3 to +7 V
POWER DISSIPATION PER PACKAGE (P_D):
For T_A = 0° to +60°C (PACKAGE TYPE E) ..... 500 mW
For T_A = +60° to +70°C (PACKAGE TYPE E) ..... Derate Linearly at 8 mW/°C to 420 mW
For T_A = 0° to +70°C (PACKAGE TYPE D) ..... 500 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A = FULL PACKAGE-TEMPERATURE RANGE ..... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D ..... 0 to +70°C
PACKAGE TYPE E ..... 0 to +70°C
STORAGE TEMPERATURE RANGE (T_{stg}) ..... -55 to +125°C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s max. ..... +265°C

OPERATING CONDITIONS at T_A = 0 to +70° C
For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC		LIMITS ALL TYPES MIN.	LIMITS ALL TYPES MAX.	UNITS
DC Operating Voltage Range		4.5	5.5	V
Input Voltage Range	V_{IH}	2.2	V_{DD} + 0.3	V
	V_{IL}	-0.3	0.8	V
Input Signal Rise or Fall Time[Δ]	t_r, t_f	—	5	μs

[Δ] Input signal rise and fall times with a duration greater than the maximum value can cause loss of stored data in the selected mode.

STATIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70° C, V_{DD} = 5 V ± 10%, Except as noted

CHARACTERISTIC		CONDITIONS		LIMITS ALL TYPES Min.	Typ.*	Max.	UNITS
Standby Device Current	I_{DDS}	$\overline{CE1}$=V_{IH} or CE2=V_{IL}		—	1.5	3	mA
	I_{DDS1}	$\overline{CE1}$=CE2 ≥ V_{DD}-0.2 V or CE2 ≤ 0.2 V		—	2	100	μA
Output Voltage Low Level	V_{OL} Max.	I_{OL}=2.1 mA		—	—	0.4	V
		I_{OL}=1 μA		—	0.1	—	
Output Voltage High Level	V_{OH} Min.	I_{OH}=-1 mA		2.4	—	—	V
		I_{OH}=-1 μA		—	V_{DD}-0.1	—	
Input Leakage Current	I_{IN} Max.	V_{IN}=0 V to V_{DD}		—	±0.1	±2	μA
3-State Output Leakage Current	I_{OUT}	$V_{I/O}$=0 V to V_{DD}		—	±0.5	±2	μA
Operating Device Current	I_{OPER1}[#]	V_{IN}=V_{IL}, V_{IH}	t_{cyc}=1 μs	—	4.5	9	mA
			t_{cyc}=120 ns	—	22.5	45	
	I_{OPER2}[#]	V_{IN}=0.2 V, V_{DD}-0.2 V	t_{cyc}=1 μs	—	2	4	
			t_{cyc}=120 ns	—	20	40	
Input Capacitance	C_{IN}	V_{IN}=0 V, f=1 MHz, T_A=25°C		—	4	6	pF
Output Capacitance	$C_{I/O}$	$V_{I/O}$=0 V, f=1 MHz, T_A=25°C		—	6	8	pF

*Typical values are for T_A=25° C and nominal V_{DD}

#Outputs open circuited.

SIGNAL DESCRIPTIONS

A0-A10 (Address Inputs): These inputs must be stable prior to a write operation, but may change asynchronously during read functions.

I/01-I/08: 8-bit tristate data bus.

$\overline{CE1}$, CE2 (Chip Enable): Either chip enable, when not true, powers down the chip, disables Read and Write functions, and gates off address and output buffers.

$\overline{OE}$ (Output Enable): Enables tristate outputs if $\overline{CE1}$ and CE2 are valid and $\overline{WE}$ is high.

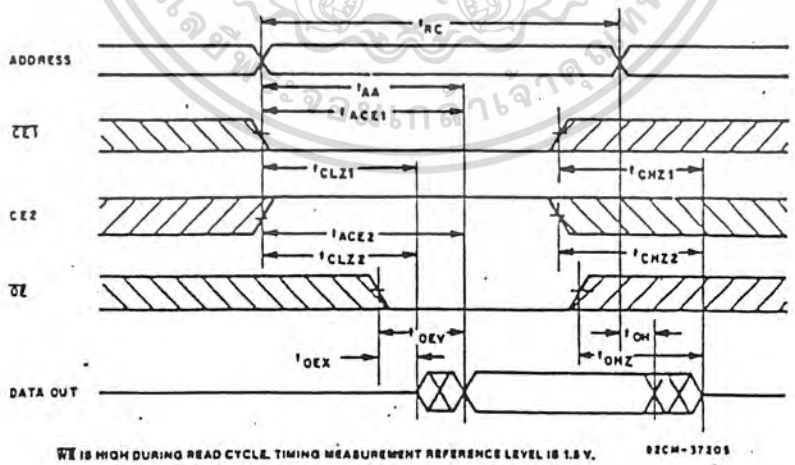
$\overline{WE}$ (Write Enable): Enables Write function, if $\overline{CE1}$ and CE2 are valid. $\overline{WE}$ will dominate if both $\overline{WE}$ and $\overline{OE}$ are low (i.e., the bus will be tristated and a Write will occur).

V_{DD}, V_{SS}: Power supply connections.

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70°C, V_{DD} = 5 V ± 10%, Input t_r, t_f = 10 ns; C_L = 100 pF and 1 TTL Load, Input Pulse Levels: 0.8 V to 2.4 V

CHARACTERISTIC		LIMITS				UNITS
		CDM6264-3		CDM6264-4		
Read Cycle Times, See Fig. 2		MIN.†	MAX.	MIN.†	MAX.	
Read Cycle Time	t_{RC}	150	—	120	—	ns
Address Access Time	t_{AA}	—	150	—	120	
Chip Enable Access Time	t_{ACE1}, t_{ACE2}	—	150	—	120	
Chip Enable to Output Active	t_{CLZ1}, t_{CLZ2}	10	—	10	—	
Output Enable to Output Valid	t_{OEV}	—	70	—	60	
Output Enable to Output Active	t_{OEX}	5	—	5	—	
Chip Disable to Output "High Z"	t_{CHZ1}, t_{CHZ2}	0	70	0	50	
Output Disable to Output "High Z"	t_{OHZ}	0	60	0	40	
Output Hold from Address Change	t_{OH}	30	—	30	—	

†Time required by a limit device to allow for the indicated function.



WE IS HIGH DURING READ CYCLE. TIMING MEASUREMENT REFERENCE LEVEL IS 1.5 V. 82CM-37305

Fig. 2 – Read-cycle timing waveforms.

CDM6264

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70° C, V_{CC} = 5 V ± 10%, Input t_r, t_f = 10 ns; C_L = 100 pF and 1 TTL Load, Input Pulse Levels: 0.8 V to 2.4 V

CHARACTERISTIC		LIMITS				UNITS
		CDM6264-3		CDM6264-4		
Write Cycle Times, See Fig. 3		MIN.†	MAX.	MIN.†	MAX.	
Write Cycle Time	t_{WC}	150	—	120	—	ns
Chip Enable to End of WRITE	t_{CW1}, t_{CW2}	120	—	100	—	
Address Valid to End of WRITE	t_{AW}	120	—	100	—	
Address Setup Time	t_{AS}	0	—	0	—	
Write Enable Width	t_{WW}	100	—	80	—	
Write Recovery Time	t_{WR}	0	—	0	—	
Write to Output "High Z"	t_{WHZ}	—	70	—	50	
Input Data Setup Time	t_{DW}	60	—	50	—	
Input Data Hold Time	t_{DH}	0	—	0	—	
Output Active from End of Write	t_{OW}	10	—	10	—	

†Time required by a limit device to allow for the indicated function.

WRITE CYCLE 1 ($\overline{CE1}$ CONTROL)

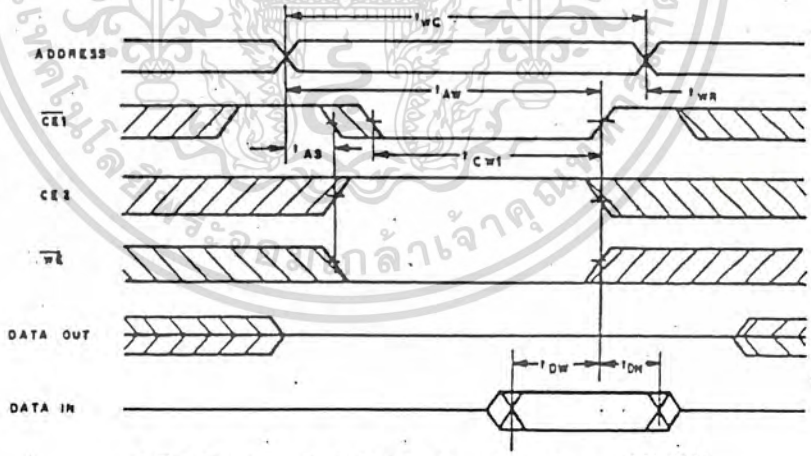
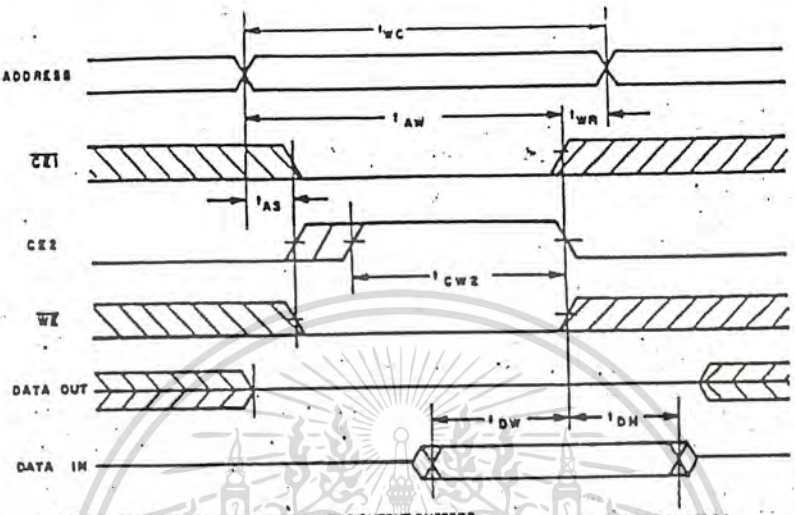


Fig. 3 – Write-cycle timing waveforms.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

WRITE CYCLE 2 (CE2 CONTROL)



WRITE CYCLE 3 ($\overline{WE}$ CONTROL)

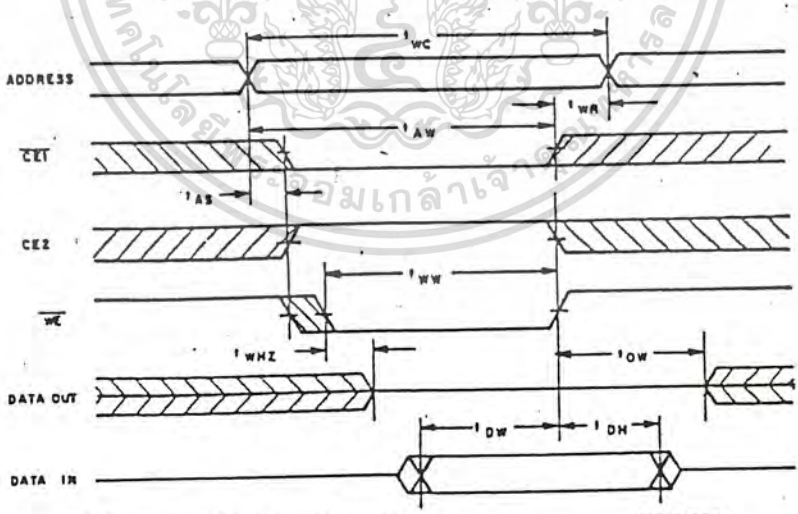


Fig. 3 - Write-cycle timing waveforms (cont'd).

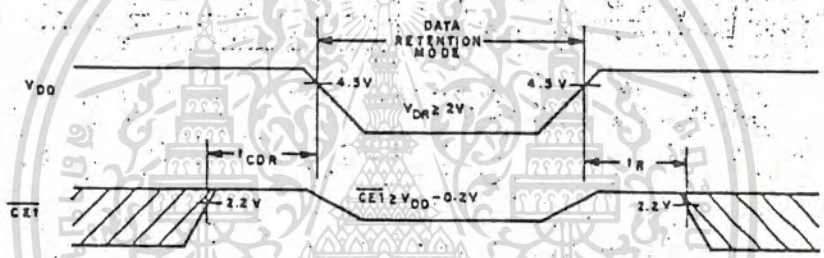
CDM6264

DATA RETENTION CHARACTERISTICS at T_A = 0 to 70°C; See Fig. 4

CHARACTERISTIC		TEST CONDITIONS	LIMITS		UNITS
			ALL TYPES		
			MIN.	MAX.	
Minimum Data Retention Voltage	V_{DR}	$\overline{CE1} \geq V_{DD} - 0.2$ V or $CE2 \leq 0.2$ V	2	5.5	V
Data Retention Quiescent Current	I_{DDR}	V_{DD} = 3 V, $\overline{CE1}$, CE2 ≥ $V_{DD} - 0.2$ V or CE2 ≤ 0.2 V	—	50	µA
Chip Disable to Data Retention Time	t_{CDR}	See Fig. 4	0	—	ns
Recovery to Normal Operation Time	t_R	See Fig. 4	*t_{RC}	—	ns

*t_{RC} = Read Cycle Time.

DATA RETENTION WAVEFORM 1 ($\overline{CE1}$ CONTROL)



DATA RETENTION WAVEFORM 2 (CE2 CONTROL)

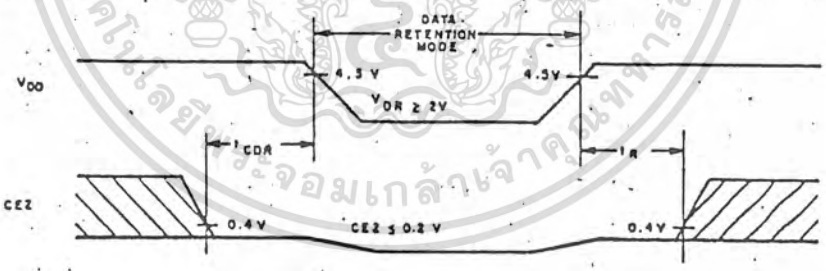


Fig. 4 – Low V_{DD} data-retention timing waveforms. 92CM-37208

National Semiconductor

microCMOS

MM54HC138/MM74HC138 3-to-8 Line Decoder

General Description

This decoder utilizes microCMOS Technology, 3.5 micron silicon gate P-well CMOS, and is well suited to memory address decoding or data routing applications. The circuit features high noise immunity and low power consumption usually associated with CMOS circuitry, yet has speeds comparable to low power Schottky TTL logic.

The MM54HC138/MM74HC138 has 3 binary select inputs (A, B, and C). If the device is enabled these inputs determine which one of the eight normally high outputs will go low. Two active low and one active high enables (G1, $\overline{G2A}$ and $\overline{G2B}$) are provided to ease the cascading of decoders.

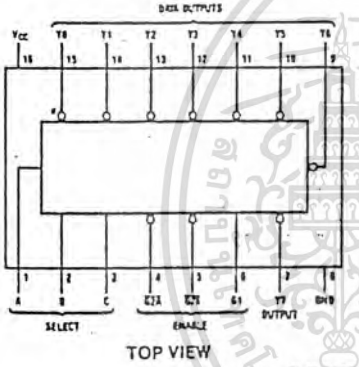
The decoder's outputs can drive 10 low power Schottky TTL equivalent loads, and are functionally and pin equivalent to the 54LS138/74LS138. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

- Typical propagation delay: 20 ns
- Wide power supply range: 2V–6V
- Low quiescent current: 80 μA maximum (74HC series)
- Low input current: 1 μA maximum
- Fanout of 10 LS-TTL loads

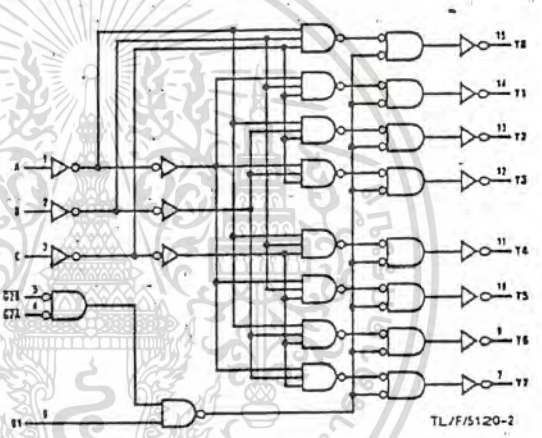
Connection Diagram

Dual-In-Line Package



TL/F/5120–1

Logic Diagram



TL/F/5120–2

MM54HC138/MM74HC138

54HC138 (J) 74HC138 (J,N)

Truth Table

Inputs					Outputs							
Enable		Select										
G1	$\overline{G2}$*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

* $\overline{G2} = \overline{G2A} + \overline{G2B}$

H = high level, L = low level, X = don't care

Absolute Maximum Ratings (Notes 1 & 2)

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−1.5 to V_{CC} +1.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC} +0.5V
Clamp Diode Current (I_{IK}, I_{OK})	±20 mA
DC Output Current, per pin (I_{OUT})	±25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	±50 mA
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D) (Note 3)	500 mW
Lead Temperature (T_L) (Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage(V_{CC})	2	6	V
DC Input or Output Voltage (V_{IN}, V_{OUT})	0	V_{CC}	V
Operating Temperature Range(T_A)			
MM74HC	−40	+85	°C
MM54HC	−55	+125	°C
Input Rise or Fall Times (t_r, t_f) V_{CC} = 2.0V		1000	ns
V_{CC} = 4.5V		500	ns
V_{CC} = 6.0V		400	ns

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C	74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ	Guaranteed Limits		
V_{IH}	Minimum High Level Input Voltage		2.0V		1.5	1.5	1.5	V
			4.5V		3.15	3.15	3.15	V
			6.0V		4.2	4.2	4.2	V
V_{IL}	Maximum Low Level Input Voltage		2.0V		0.3	0.3	0.3	V
			4.5V		0.9	0.9	0.9	V
			6.0V		1.2	1.2	1.2	V
V_{OH}	Minimum High Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 20\ \mu A$	2.0V	2.0	1.9	1.9	1.9	V
			4.5V	4.5	4.4	4.4	4.4	V
			6.0V	6.0	5.9	5.9	5.9	V
		V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 4.0$ mA	4.5V	4.2	3.98	3.84	3.7	V
		$\lvert I_{OUT}\rvert \le 5.2$ mA	6.0V	5.7	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 20\ \mu A$	2.0V	0	0.1	0.1	0.1	V
			4.5V	0	0.1	0.1	0.1	V
			6.0V	0	0.1	0.1	0.1	V
		V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 4.0$ mA	4.5V	0.2	0.26	0.33	0.4	V
		$\lvert I_{OUT}\rvert \le 5.2$ mA	6.0V	0.2	0.26	0.33	0.4	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND	6.0V		±0.1	±1.0	±1.0	μA
I_{CC}	Maximum Quiescent Supply Current	V_{IN} = V_{CC} or GND I_{OUT} = 0 μA	6.0V		8.0	80	160	μA

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: −12 mW/°C from 65°C to 85°C; ceramic "J" package: −12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V ±10% the worst case output voltages (V_{OH}, and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{IH} and V_{IL} occur at V_{CC} = 5.5V and 4.5V respectively. (The V_{IH} value at 5.5V is 3.85V.) The worst case leakage current (I_{IN}, I_{CC}, and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

AC Electrical Characteristics

V_{CC} = 5V, T_A = 25°C, C_L = 15 pF, $t_r = t_f$ = 6 ns

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PLH}	Maximum Propagation Delay, Binary Select to any Output		18	25	ns
t_{PHL}	Maximum Propagation Delay, Binary Select to any Output		28	35	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay, G1 to any Output		18	25	ns
t_{PHL}	Maximum Propagation Delay $\overline{G2A}$ or $\overline{G2B}$ to Output		23	30	ns
t_{PLH}	Maximum Propagation Delay $\overline{G2A}$ or $\overline{G2B}$ to Output		18	25	ns

AC Electrical Characteristics C_L = 50 pF, $t_r = t_f$ = 6 ns (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C		74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ		Guaranteed Limits		
t_{PLH}	Maximum Propagation Delay Binary Select to any Output Low to High		2.0V	75	150	189	224	ns
			4.5V	15	30	38	45	ns
			6.0V	13	26	32	38	ns
t_{PHL}	Maximum Propagation Delay Binary Select to any Output High to Low		2.0V	100	200	252	298	ns
			4.5V	20	40	40	60	ns
			6.0V	17	34	43	51	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay G1 to any Output		2.0V	75	150	189	224	ns
			4.5V	15	30	38	45	ns
			6.0V	13	26	32	38	ns
t_{PHL}	Maximum Propagation Delay $\overline{G2A}$ or $\overline{G2B}$ to Output		2.0V	82	175	221	261	ns
			4.5V	28	35	44	52	ns
			6.0V	22	30	37	44	ns
t_{PLH}	Maximum Propagation Delay $\overline{G2A}$ or $\overline{G2B}$ to Output		2.0V	75	150	189	224	ns
			4.5V	15	30	38	45	ns
			6.0V	13	26	32	38	ns
t_{TLH}, t_{THL}	Output Rise and Fall Time		2.0V	30	75	95	110	ns
			4.5V	8	15	19	22	ns
			6.0V	7	13	16	19	ns
C_{IN}	Maximum Input Capacitance			3	10	10	10	µF
C_{PD}	Power Dissipation Capacitance	(Note 5)		75				µF

Note 5: C_{PD} determines the no load dynamic power consumption, $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$, and the no load dynamic current consumption, $I_S = C_{PD} V_{CC} f + I_{CC}$.

Note 6: Refer to back of this section for Typical MM54/74HC AC Switching Waveforms and Test Circuits.

National Semiconductor

microCMOS

MM54HC154/MM74HC154 4-to-16 Line Decoder

General Description

This decoder utilizes microCMOS Technology, 3.5 micron silicon gate P-well CMOS, and is well suited to memory address decoding or data routing applications. It possesses high noise immunity, and low power consumption of CMOS with speeds similar to low power Schottky TTL circuits.

The MM54HC154/MM74HC154 have 4 binary select inputs (A, B, C, and D). If the device is enabled these inputs determine which one of the 16 normally high outputs will go low. Two active low enables ($\overline{G1}$ and $\overline{G2}$) are provided to ease cascading of decoders with little or no external logic.

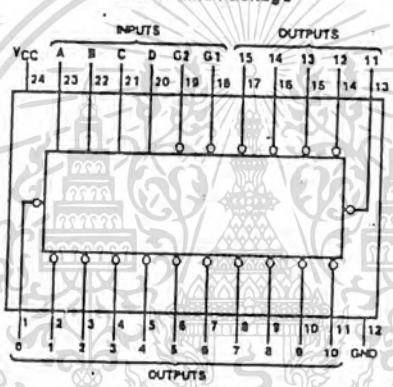
Each output can drive 10 low power Schottky TTL equivalent loads, and is functionally and pin equivalent to the 54LS154/74LS154. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

- Typical propagation delay: 21 ns
- Power supply quiescent current: 80 μA (74HC)
- Wide power supply voltage range: 2–6V
- Low input current: 1 μA maximum

Connection Diagram

Dual-In-Line Package



TL/F/5122–1

TOP VIEW

MM54HC154/MM74HC154

54HC154 (J) 74HC154 (J,N)

Truth Table

Inputs						Low Output*
$\overline{G1}$	$\overline{G2}$	D	C	B	A	
L	L	L	L	L	L	0
L	L	L	L	L	H	1
L	L	L	L	H	L	2
L	L	L	L	H	H	3
L	L	L	H	L	L	4
L	L	L	H	L	H	5
L	L	L	H	H	L	6
L	L	L	H	H	H	7
L	L	H	L	L	L	8
L	L	H	L	L	H	9
L	L	H	L	H	L	10
L	L	H	L	H	H	11
L	L	H	H	L	L	12
L	L	H	H	L	H	13
L	L	H	H	H	L	14
L	L	H	H	H	H	15
L	H	X	X	X	X	—
H	L	X	X	X	X	—
H	H	X	X	X	X	—

*All others high

Absolute Maximum Ratings (Notes 1 & 2)

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−1.5 to V_{CC} +1.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC} +0.5V
Clamp Diode Current (I_{IK}, I_{OK})	±20 mA
DC Output Current, per pin (I_{OUT})	±25 mA
DC V_{CC} or GND Current, per pin (I_{CC})	±50 mA
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D) (Note 3)	500 mW
Lead Temperature (T_L) (Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage(V_{CC})	2	6	V
DC Input or Output Voltage (V_{IN},V_{OUT})	0	V_{CC}	V
Operating Temperature Range(T_A)			
MM74HC	−40	+85	°C
MM54HC	−55	+125	°C
Input Rise or Fall Times (t_r, t_f) V_{CC} = 2.0V		1000	ns
V_{CC} = 4.5V		500	ns
V_{CC} = 6.0V		400	ns

DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C	74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ	Guaranteed Limits		
V_{IH}	Minimum High Level Input Voltage		2.0V		1.5	1.5	V
			4.5V		3.15	3.15	V
			6.0V		4.2	4.2	V
V_{IL}	Maximum Low Level Input Voltage		2.0V		0.3	0.3	V
			4.5V		0.9	0.9	V
			6.0V		1.2	1.2	V
V_{OH}	Minimum High Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 20\ \mu A$	2.0V	2.0	1.9	1.9	V
			4.5V	4.5	4.4	4.4	V
			6.0V	6.0	5.9	5.9	V
		V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 4.0$ mA	4.5V	4.2	3.98	3.84	V
		$\lvert I_{OUT}\rvert \le 5.2$ mA	6.0V	5.7	5.48	5.34	V
V_{OL}	Maximum Low Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 20\ \mu A$	2.0V	0	0.1	0.1	V
			4.5V	0	0.1	0.1	V
			6.0V	0	0.1	0.1	V
		V_{IN} = V_{IH} or V_{IL} $\lvert I_{OUT}\rvert \le 4.0$ mA	4.5V	0.2	0.26	0.33	V
		$\lvert I_{OUT}\rvert \le 5.2$ mA	6.0V	0.2	0.26	0.33	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND	6.0V		±0.1	±1.0	µA
I_{CC}	Maximum Quiescent Supply Current	V_{IN} = V_{CC} or GND I_{OUT} = 0 µA	6.0V		8.0	80	µA

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: −12 mW/°C from 65°C to 85°C; ceramic "J" package: −12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V ±10% the worst case output voltages (V_{OH}, and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{IH} and V_{IL} occur at V_{CC} = 5.5V and 4.5V respectively. (The V_{IH} value at 5.5V is 3.85V.) The worst case leakage current (I_{IN}, I_{CC}, and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, C_L = 15 pF, $t_r = t_f$ = 6 ns

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL}, t_{PLH}	Maximum Propagation Delay, $\overline{G1}$, $\overline{G2}$ or A, B, C, D		21	32	ns

AC Electrical Characteristics V_{CC} = 2.0V to 6.0V, C_L = 50 pF, $t_r = t_f$ = 6 ns (unless otherwise specified)

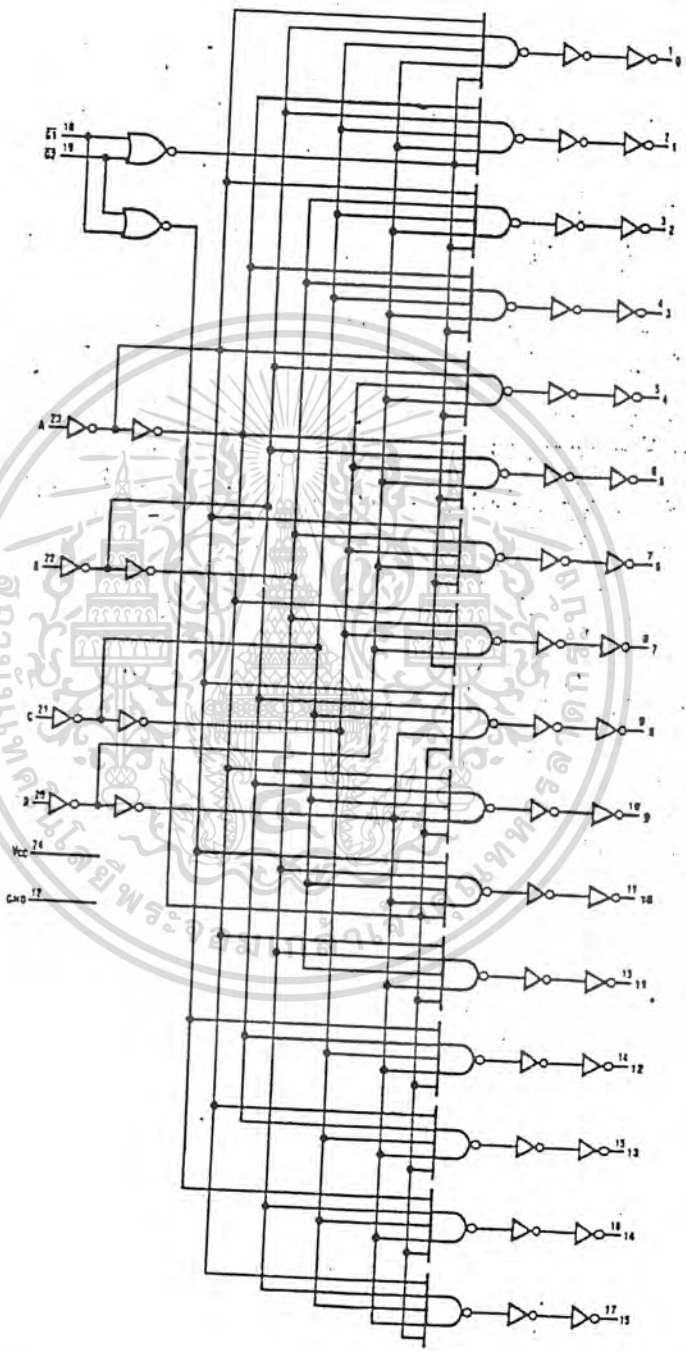
Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C	74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ	Guaranteed Limits		
t_{PHL}, t_{PLH}	Maximum Propagation Delay, $\overline{G1}$ or $\overline{G2}$ or A, B, C, D		2.0V	63	160	190	220	ns
			4.5V	24	36	42	46	ns
			6.0V	20	30	35	39	ns
t_{TLH}, t_{THL}	Maximum Output Rise and Fall Time		2.0V	25	75	95	110	ns
			4.5V	7	15	19	22	ns
			6.0V	6	13	16	19	ns
C_{PD}	Power Dissipation Capacitance (Note 5)			90				pF
C_{IN}	Maximum Input Capacitance			5	10	10	10	pF

Note 5: C_{PD} determines the no load dynamic power consumption, $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$, and the no load dynamic current consumption, $I_S = C_{PD} V_{CC} f + I_{CC}$.

Note 6. Refer to back of this section for Typical MM54/74HC AC Switching Waveforms and Test Circuits.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Logic Diagram



TL/F/5122–2

National Semiconductor

microCMOS

MM54HC373/MM74HC373 TRI-STATE® Octal D-Type Latch

General Description

These high speed OCTAL D-TYPE LATCHES utilize microCMOS Technology, 3.5 micron silicon gate P-well CMOS. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

When the LATCH ENABLE input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

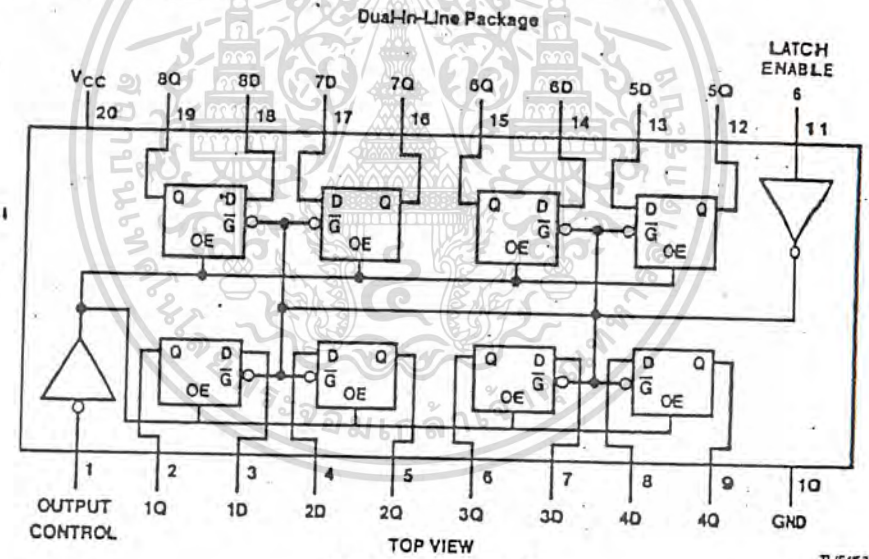
The 54HC/74HC logic family is speed, function, and pinout compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μA maximum
- Low quiescent current: 80 μA maximum (74 series)
- Output drive capability: 15 LS-TTL loads

Connection Diagram

Dual-In-Line Package



TOP VIEW

TL/F/5333-1

MM54HC373/MM74HC373

54HC373 (J) 74HC373 (J,N)

Truth Table

Output Control	Latch Enable	Data	373 Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

H = high level, L = low level

Q_0 = level of output before steady-state input conditions were established.

Z = high impedance

Absolute Maximum Ratings (Notes 1 & 2)

Voltage (V_{CC})	−0.5 to +7.0V
Input Voltage (V_{IN})	−1.5 to V_{CC} +1.5V
Output Voltage (V_{OUT})	−0.5 to V_{CC} +0.5V
Clamp Diode Current (I_{IK}, I_{OK})	±20 mA
DC Output Current, per pin (I_{OUT})	±35 mA
DC V_{CC} or GND Current, per pin (I_{CC})	±70 mA
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D)	500 mW
Lead Temperature (T_L) (Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	2	6	V
DC Input or Output Voltage (V_{IN}, V_{OUT})	0	V_{CC}	V
Operating Temperature Range (T_A)			
MM74HC	−40	+85	°C
MM54HC	−55	+125	°C
Input Rise or Fall Times (t_r, t_f) V_{CC} = 2.0V		1000	ns
V_{CC} = 4.5V		500	ns
V_{CC} = 6.0V		400	ns

DC Electrical Characteristics

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C	74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ	Guaranteed Limits		
V_{IH}	Minimum High Level Input Voltage		2.0V		1.5	1.5	1.5	V
			4.5V		3.15	3.15	3.15	V
			6.0V		4.2	4.2	4.2	V
V_{IL}	Maximum Low Level Input Voltage		2.0V		0.3	0.3	0.3	V
			4.5V		0.9	0.9	0.9	V
			6.0V		1.2	1.2	1.2	V
V_{OH}	Minimum High Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \le 20\ \mu A$	2.0V	2.0	1.9	1.9	1.9	V
			4.5V	4.5	4.4	4.4	4.4	V
			6.0V	6.0	5.9	5.9	5.9	V
		V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \le 6.0$ mA	4.5V	4.2	3.98	3.84	3.7	V
		$\|I_{OUT}\| \le 7.8$ mA	6.0V	5.7	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \le 20\ \mu A$	2.0V	0	0.1	0.1	0.1	V
			4.5V	0	0.1	0.1	0.1	V
			6.0V	0	0.1	0.1	0.1	V
		V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \le 6.0$ mA	4.5V	0.2	0.26	0.33	0.4	V
		$\|I_{OUT}\| \le 7.8$ mA	6.0V	0.2	0.26	0.33	0.4	V
I_I	Maximum Input Current	V_{IN} = V_{CC} or GND	6.0V		±0.1	±1.0	±1.0	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{IN} = V_{IH} or V_{IL}, OC = V_{IH} V_{OUT} = V_{CC} or GND	6.0V		±0.5	±5	±10	μA
I_{CC}	Maximum Quiescent Supply Current	V_{IN} = V_{CC} or GND I_{OUT} = 0 μA	6.0V		8.0	80	160	μA

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: −12 mW/°C from 65°C to 85°C; ceramic "J" package: −12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V ±10% the worst case output voltages (V_{OH} and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{IH} and V_{IL} occur at V_{CC} = 5.5V and 4.5V respectively. (The V_{IH} value at 5.5V is 3.85V.) The worst case leakage current (I_I, I_{CC}, and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, $t_r = t_f$ = 6 ns (Note 6)

Symbol	Parameter	Conditions	Typ	Guaranteed Limit
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Data to Q	C_L = 45 pF	18	25
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Q	C_L = 45 pF	21	30
t_{PZH}, t_{PZL}	Maximum Output Enable Time	R_L = 1 kΩ C_L = 45 pF	20	28
t_{PHZ}, t_{PLZ}	Maximum Output Disable Time	R_L = 1 kΩ C_L = 5 pF	18	25
t_S	Minimum Set Up Time			5
t_H	Minimum Hold Time			10
t_W	Minimum Pulse Width		9	16

AC Electrical Characteristics V_{CC} = 2.0–6.0V, C_L = 50 pF, $t_r = t_f$ = 6 ns (unless otherwise specified) (Note

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C		74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C
				Typ		Guaranteed Limits	
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Data to Q	C_L = 50 pF	2.0V	50	150	188	225
		C_L = 150 pF	2.0V	80	200	250	300
		C_L = 50 pF	4.5V	22	30	37	45
		C_L = 150 pF	4.5V	30	40	50	60
		C_L = 50 pF	6.0V	19	26	31	39
		C_L = 150 pF	6.0V	26	35	44	53
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Q	C_L = 50 pF	2.0V	63	175	220	263
		C_L = 150 pF	2.0V	110	225	280	338
		C_L = 50 pF	4.5V	25	35	44	52
		C_L = 150 pF	4.5V	35	45	56	68
		C_L = 50 pF	6.0V	21	30	37	45
		C_L = 150 pF	6.0V	28	39	49	59
t_{PZH}, t_{PZL}	Maximum Output Enable Time	R_L = 1 kΩ					
		C_L = 50 pF	2.0V	50	150	188	225
		C_L = 150 pF	2.0V	80	200	250	300
		C_L = 50 pF	4.5V	21	30	37	45
		C_L = 150 pF	4.5V	30	40	50	60
		C_L = 50 pF	6.0V	19	26	31	39
		C_L = 150 pF	6.0V	26	35	44	53
t_{PHZ}, t_{PLZ}	Maximum Output Disable Time	R_L = 1 kΩ C_L = 50 pF	2.0V	50	150	188	225
			4.5V	21	30	37	45
			6.0V	19	26	31	39
t_S	Minimum Set Up Time		2.0V	5	25	31	38
			4.5V	2	5	6	8
			6.0V	2	5	6	8
t_H	Minimum Hold Time		2.0V	20	50	60	75
			4.5V	6	10	13	20
			6.0V	6	10	13	20
t_W	Minimum Pulse Width		2.0V	30	80	100	120
			4.5V	10	16	20	24
			6.0V	9	14	18	20
t_{THL}, t_{TLH}	Maximum Output Rise and Fall Time	C_L = 50 pF	2.0V	25	60	75	90
			4.5V	7	12	15	18
			6.0V	6	10	13	15
C_{PD}	Power Dissipation Capacitance (Note 5)	(per latch) OC = V_{CC} OC = GND		30 50			
C_{IN}	Maximum Input Capacitance			5	10	10	10
C_{OUT}	Maximum Output Capacitance			15	20	20	20

Note 5: C_{PD} determines the no load dynamic power consumption, $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$, and the no load dynamic current consum $I_S = C_{PD} V_{CC} f + I_{CC}$.

Note 6: Refer to back of this section for Typical MM54/74HC AC Switching Waveforms and Test Circuits.