RS 232/422 CONVERTER

โดย

นายกิตติพงษ์ ศรีศันสนีย

นายเกรียงไกร ประเสริฐดีงาม

นายจักรา รัตนจรณะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

008436

ปริญญานิพนธ์ ปีการศึกษา 2534

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง RS 232/422 CONVERTER

ผู้จัดทำ

1. นายกิตติพงษ์	ศรีศันสนีย	31.1008
2. นายเกรียงไกร	ประเสริฐดีงาม	31.1012
3. นายจักรา	รัตนจรณะ	31.1030

.....

อาจารย์ สุพรรณ กุลพาณิชย์

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPN 3

สารบัญ

เรื่อง	หน้า
บทนำ	1
บทคัดย่อ	2
Abstract	3
บทที่ 1 มาตรฐานสำหรับการสื่อสารข้อมูล	4
บทที่ 2 พอร์ตอนุกรมของ ไอบี เอ็มพีซี	16
บทที่ 3 ความรู้พื้นฐานสำหรับการทดลอง	30
บทที่ 4 การทดลอง	45
ภาคผนวก	53

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

โครงงาน RS 232/422 CONVERTER เป็นส่วนหนึ่งของวิชา project สำหรับนักศึกษาชั้นปีที่ 4 ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม ซึ่งอยู่ในความดูแลของอาจารย์สุพรรณ กุลพานิชย์ โครงงานนี้มีนักศึกษารับผิดชอบ 3 คน คือ

1. นาย กิตติพงษ์ ศรีศันสนีย
2. นาย เกรียงไกร ประเสริฐดีงาม
3. นายจักรา รัตนจรณะ

โครงงาน RS 232/422 CONVERTER มีระยะเวลาของโครงงาน 2 ภาคการศึกษา

ในภาคการศึกษาที่ 1 โครงงานจะแสดงถึง การสื่อสารข้อมูลเบื้องต้นระหว่าง computer และ Z 80 บน MPF-I-plus ทั้งการส่งและรับข้อมูลระหว่างกันอย่างมีรูปแบบ โดยอุปกรณ์ที่ใช้มี

- computer
- MPF-I-plus
- วงจรด้าน hardware สำหรับการ interface ผ่าน RS 232-C
- software สำหรับ computer และ MPF-I-plus เพื่อสร้างรูปแบบในการรับและส่งข้อมูล

ในภาคการศึกษาที่ 2 โครงงานจะถูกพัฒนาเพิ่มขึ้นให้มี photocall สำหรับเลือก unit no. ของ MPF-I-plus และมีการแปลงสัญญาณจาก RS 232 ผ่าน RS 422 ให้สามารถรับส่งสัญญาณได้ไกลขึ้น โดยอุปกรณ์มี

- computer
- MPF-I-plus
- link addapter
- hardware สำหรับการ interface
- software,photocall สำหรับ computer และ MPF-I-plus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

เป็นที่ทราบกันอยู่แล้วว่า การสื่อสารข้อมูลมีบทบาทอย่างมากในปัจจุบัน มีการนำเอาไปประยุกต์ใช้ในงานอุตสาหกรรม การแพทย์ การวิศวกรรม และอีกหลายแขนง โครงงาน RS232/422 CONVERTER นี้ จะแสดงถึงการสื่อสารข้อมูล ทางการติดต่อระหว่าง computer และ MPF-I-plus ซึ่งเป็นการติดต่อแบบ 1:32 โดยผ่าน RS 232-C แต่การสื่อสารผ่าน RS 232-C นี้มีข้อจำกัดคือ ส่งได้ระยะใกล้ จึงปรับปรุงโดยการ แปลงสัญญาณจาก RS 232-C ผ่าน converter RS 422 เพื่อให้สามารถส่งได้ระยะทางไกลขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

It is nowadays remarkably known that any kind of communication has took an important role in the society. We have applied it largely in industry,medical treatment, engineering, and so on. The project of RS 232/422 CONVERTER will convey the method of transmission of information which will be in the link of operation of the computer and MPF-I-Plus. This type of transformation is the 1:32 type which is to be applied through RS 232-C. However, it is limited in view of short-distance capability. As a result,in order to upgrade the transmission of information, we change the signals from RS 232 through RS 422 to develop the capability of transmisstion in longer distance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

มาตรฐานสำหรับการสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานสำหรับการสื่อสารข้อมูล

ในยุคข้อมูลและข่าวสารนี้ สิ่งที่มีความสำคัญมากที่สุดสิ่งหนึ่งก็คือระบบการสื่อสาร การใช้ระบบการสื่อสารนั้นจำเป็นต้องมีการกำหนดมาตรฐานขึ้น ถ้าหากไม่มีการกำหนดมาตรฐานแล้ว การสื่อสารก็คงจะสับสนอุ่ง เหยิงไร้ระเบียบ เกิดปัญหามากมายไม่รู้จบ ข้อมูลที่มีอยู่ไม่ว่าจะมีคุณค่ามาก เพียงใดก็ไม่สามารถนำมาใช้ให้เกิดประโยชน์อย่างเต็มที่ได้

การสื่อสารข้อมูลแบบอนุกรม

ถ้าขึ้นหัวข้อว่าการสื่อสารข้อมูลแบบอนุกรมแล้ว แสดงว่าจะต้องมีการสื่อสารข้อมูลแบบขนานด้วย การสื่อสารแบบขนานก็คือข้อมูลทุก ๆ บิตในแต่ละเวิร์ดจะถูกส่งออกไปพร้อม ๆ กัน ขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไหร่ โดยทั่วไปก็คือ 1ไบท์หรือ 8 บิตนั่นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อจำกัดทางด้านระยะทางระหว่างต้นทางและปลายทาง โดยทั่วไปจะส่งได้ในระยะไม่เกิน 3-5 ฟุต เท่านั้น ทั้งนี้ขึ้นอยู่กับอัตราที่ใช้ในการส่งข้อมูล ถ้าหากใช้อัตราการส่งข้อมูลสูงก็จะส่งได้ระยะสั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูงมาก ๆ แต่อุปกรณ์อยู่ไม่ห่างกันมากนัก

ส่วนการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกทยอยส่งออกไปทีละบิตจนครบทั้ง เวิร์ดในสายสัญญาณเพียง เส้นเดียว แต่ในการใช้งานจริงจะต้องมีสายสัญญาณอีก เส้น เป็นระดับ Ground ดังนั้น เมื่อเราส่งข้อมูลในแบบอนุกรม เราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุด เพียง 2 เส้น ในขณะที่การส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อย เท่ากับจำนวนบิตบวกกับสายสัญญาณระดับแรงดันต่ำ-Ground อีก 1 เส้น นอกจากนี้ การส่งข้อมูลแบบอนุกรมนั้น จะสามารถส่งข้อมูลได้ไกลกว่ามาก เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่กล่าวต่อไปในภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุตโดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่อย่างใด อย่างไรก็ดี การส่งข้อมูลแบบอนุกรมนั้น จะต้องมีส่วนที่ทำหน้าที่แปลงข้อมูลจากข้อมูลแบบขนานมาเป็นแบบอนุกรม ซึ่งสามารถเป็นได้ทั้งฮาร์ดแวร์และซอฟต์แวร์และในการส่งยังมีข้อกำหนดบางประการเพื่อให้การส่งข้อมูลมีความถูกต้องมากยิ่งขึ้นอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Simplex และ Duplex

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือการสื่อสารทั่วไปนั้น ย่อมจะต้องประกอบด้วยผู้รับและผู้ส่ง ผู้รับในขณะนี้อาจสามารถเป็นผู้ส่งในอนาคตได้ แต่มีบางกรณีสำหรับการสื่อสารข้อมูลที่ผู้รับและผู้ส่งจะตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบ Simplex กล่าวคือการสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลา ซึ่งจะมีที่ใช้น้อยมาก การสื่อสารโดยทั่วไปนั้นจะเป็นลักษณะ Duplex คือมีทิศทางการสื่อสาร 2 ทิศทางทั้งไปและกลับ การสื่อสารในลักษณะ Duplex นี้ ยังแบ่งออกได้เป็น 2 ชนิด คือ Half Duplex (นิยมเขียนย่อว่า HDX) ซึ่งจะมีทิศทางการสื่อสารในลักษณะที่ผลัดกันเป็นผู้ส่งและผู้รับ กับการสื่อสารในลักษณะของ Full Duplex (FDX) ซึ่งอุปกรณ์ที่ปลายทางจะเป็นผู้ส่งและเป็นผู้รับในเวลาเดียวกัน เราอาจเปรียบเทียบการสื่อสารแบบ HDX หรือ Half Duplex เป็นทางรถไฟ ซึ่งจะมีรถไฟเพียงขบวนเดียวเท่านั้นที่วิ่งอยู่บนรางในเวลาหนึ่ง และเปรียบเทียบ FDX หรือ Full Duplex เป็นถนนที่รถสามารถวิ่งสวนกันได้ในเวลาเดียวกันการสื่อสารระหว่างคอมพิวเตอร์มักจะอยู่ในลักษณะของ Duplex แบบใดแบบหนึ่ง

โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมให้ดีจะพบว่า ปัญหาหนึ่งที่ควรจะเกิดขึ้นอยู่เสมอก็คือการตัดสินใจว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้นจึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าว เราเรียกว่า โปรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ โปรโตคอลสำหรับการสื่อสารข้อมูลอนุกรมแบบซิงโครนัส (Synchronous) และโปรโตคอลสำหรับการสื่อสารข้อมูลแบบอซิงโครนัส (Asynchronous) การสื่อสารแบบอซิงโครนัสนั้น ข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอ ช่วงเวลาระหว่างบิตและระหว่างเวิร์ดจะมีค่าเท่ากันเสมอดังนั้นในการสื่อสารข้อมูลอนุกรมในแบบซิงโครนัสจึงต้องมีสายสัญญาณเพิ่มเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใด

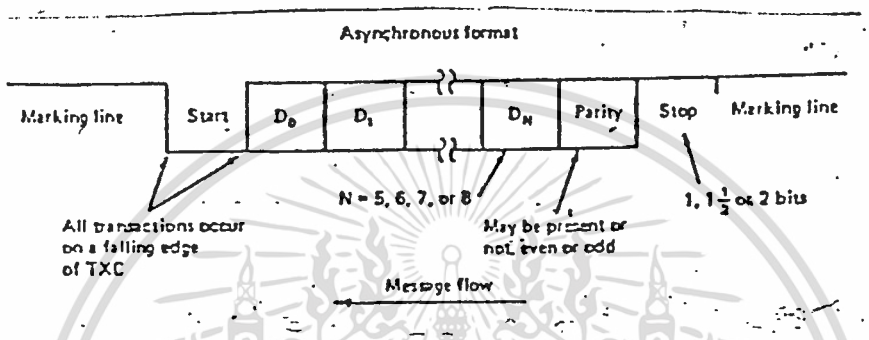
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และควรจะหยุด เมื่อใดระบบที่เป็นซิงโครนัสจะ เป็นระบบที่มีความ เร็วสูงแต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน ที่นี้ เราจะมาดูการสื่อสารแบบอซิงโครนัสกันบ้าง การสื่อสารแบบอซิงโครนัสนี้ เป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ในปัจจุบัน การสื่อสารแบบนี้ช่วงระยะเวลาระหว่างบิตจะมีค่า เท่ากัน เช่น เดียวกับซิงโครนัส แต่จะแตกต่างกันที่ระยะห่างระหว่างเวิร์ด ซึ่งไม่มีข้อกำหนดว่าจะห่างกัน เป็นระยะ เวลานาน เท่าใดกล่าวคือ ระหว่าง เวิร์ดนั้นจะห่างกันกี่วินาที กี่นาที กี่ชั่วโมง กี่วัน กี่เดือน หรือ กี่ปี ก็ได้ทั้งสิ้นขึ้นอยู่กับว่าทางฝ่ายรับสามารถรอได้หรือไม่ เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะ เวลาระหว่าง เวิร์ดแล้ว ทางผู้ส่งและผู้รับจะ เข้าใจตรงกันได้อย่างไรว่าที่ใดคือจุด เริ่มต้นและจุดสิ้นสุดของแต่ละ เวิร์ด เพื่อแก้ปัญหานี้ จึงมีการกำหนดข้อตกลง เกี่ยวกับฟอร์แมตของข้อมูลที่จะส่งให้ทางผู้รับสามารถ เข้าใจได้ว่าจุดใด เป็นจุด เริ่มต้นของ เวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละ เวิร์ดจะต้องขึ้นต้นด้วยบิตที่ เรียกว่า (Start Bit หรือ บิต เริ่มต้น) ซึ่งจะต้อง เป็นลอจิกศูนย์ เสมอจากนั้นตามด้วยบิตข้อมูลที่ต้องการส่งมีความยาว 5 ถึง 8 บิต ถัดจากบิตข้อมูลก็จะ เป็นพาริตีบิตซึ่งทำหน้าที่ เป็นบิตสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่บิตพาริตีนี้มี 2 ประ เภท คือ EvenParity ซึ่งจะ เซ็ต (มีค่า เป็น 1) เมื่อจำนวนบิตที่ เป็นลอจิก 1 ในบิตที่ เป็นข้อมูลมีจำนวน เป็นคี่และ Odd Parity ซึ่งจะ เซ็ต เมื่อจำนวนบิตที่ เป็นลอจิก 1 ในบิตที่ เป็นข้อมูลมีจำนวน เป็นคู่ ในการส่งข้อมูลบางครั้งอาจจะไม่มีการใช้บิตพาริตีก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่า เชื่อถือสูง คือมีสัญญาณรบกวนต่ำ เป็นการ เพิ่มความ เร็วในการสื่อสารได้ด้วย บิตสุดท้ายในฟอร์แมตก็คือ Stop Bit ทำหน้าที่บอกทางผู้รับว่าขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบ เวิร์ดแล้วขอให้ เตรียมตัวรับ เวิร์ดต่อไปได้ Stop Bit นี้ถูกกำหนดให้ เป็นลอจิก 1 เสมอทั้งนี้ เพื่อให้ระบบสามารถตรวจสอบบิต เริ่มต้นได้ (บิต เริ่มต้นมีลอจิก เป็น 0) Stop Bit อาจมีความยาว เป็น 1 บิต 1.5 บิต หรือ 2 บิตก็ได้

จากฟอร์แมตดังกล่าว จะ เห็นว่า เรามีฟอร์แมตสำหรับการสื่อสารมากมายไปหมด เช่น 5E1 (5Data bit,Even Parity,1 Stop Bit),7E1(7Data bit,Even Parity,1 Stop Bit)และ 8N1 (8 Data bit,No Parity,1 Stop Bit) เป็นต้น ในการใช้งานทั่วไป เรานิยมใช้กันอยู่ เพียง 2 ฟอร์แมต คือ 7E1 และ 8N1 จะ เลือกใช้ฟอร์แมตใดขึ้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมาก เพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8N1 จะเร็วกว่า เป็นต้น ทั้งนี้จะต้องมีการตกลงกันล่วงหน้าระหว่างผู้รับและผู้ส่งว่าจะใช้ฟอร์แมตใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูปที่ 1



รูปที่ 1 ตัวอย่างของการฟอร์แมตข้อมูล

OSI มาตรฐานสำหรับซอฟต์แวร์สื่อสาร

แต่เดิมการพัฒนาซอฟต์แวร์สำหรับการสื่อสารนั้นมักจะกระทำกันอย่างไร้ทิศทางขนาดข้อกำหนดที่แน่นอน ทำให้เกิดปัญหาการเข้ากันไม่ได้ของซอฟต์แวร์ กล่าวคือ ในกรณีที่มีการสื่อสารระหว่าง เครื่องคอมพิวเตอร์ 2 เครื่องจะต้องวิ่งโปรแกรมสำหรับการสื่อสารโปรแกรมเดียวกันเท่านั้น ทำให้เกิดความไม่สะดวกและขาดความอ่อนตัว เป็นอย่างมากภายหลังจึงมีการกำหนดมาตรฐานร่วมสำหรับซอฟต์แวร์สื่อสารขึ้นมาชุดหนึ่ง เรียกว่า Open System Interfacing หรือ OSI เป็นการกำหนดระดับหรือ เลเยอร์ของซอฟแวร์สำหรับการสื่อสารขึ้นโดยมีการแบ่งออกเป็น 7 เลเยอร์มีหน้าที่ต่าง ๆ กัน เริ่มจาก Physical Layer เป็นเลเยอร์ที่อยู่ล่างสุดและมักจะเป็นฮาร์ดแวร์ทั้งหมด เป็นข้อกำหนดเกี่ยวกับการเชื่อมต่อของสัญญาณต่าง ๆ และระดับลอจิก ขนาดของกระแส เป็นต้น ถัดจากชั้น Physical Layer ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเป็น Data Link Layer เป็นการกำหนดโฟร์แมตของการสื่อสาร เช่น 8N1 หรือ 7E1 เป็นต้น (ดังที่ได้กล่าวไว้แล้วในตอนต้น) ในการสื่อสารข้อมูลนั้น อย่างน้อยจะต้องมี 2 ระดับนี้ จึงจะสามารถทำการสื่อสารได้ อีก 5 ชั้นที่เหลือนั้น มักจะพบในการสื่อสารระดับสูงและซับซ้อนมาก ๆ เช่นในระบบสื่อสารระหว่างประเทศ หรือในเครื่องคอมพิวเตอร์ระดับมินิและเมนเฟรมเป็นต้น จะขอกล่าวเพียงย่อ ๆ เท่านั้น ได้แก่ชั้น Network Level จะเป็นส่วนที่ทำหน้าที่ควบคุมการไหลของข้อมูลให้ไปในทิศทางที่ถูกต้อง ชั้น Transfer Layer ทำหน้าที่การควบคุมความน่าเชื่อถือ(Reliability) ของข้อมูล มีการตรวจสอบข้อผิดพลาดในระดับสูง เช่น การใช้ CheckSum และ CRC การทำ Error Correction เป็นต้น ชั้น Session Layer ทำหน้าที่เริ่มและสิ้นสุดการติดต่อข้อมูล Presentation Layer ทำหน้าที่กำหนดโครงสร้างของข้อมูลทั้งบล็อก และชั้นสุดท้ายซึ่งอยู่สูงสุดคือ Application Layer ซึ่งเป็นส่วนของโปรแกรมของผู้ใช้

มาตรฐาน EIA

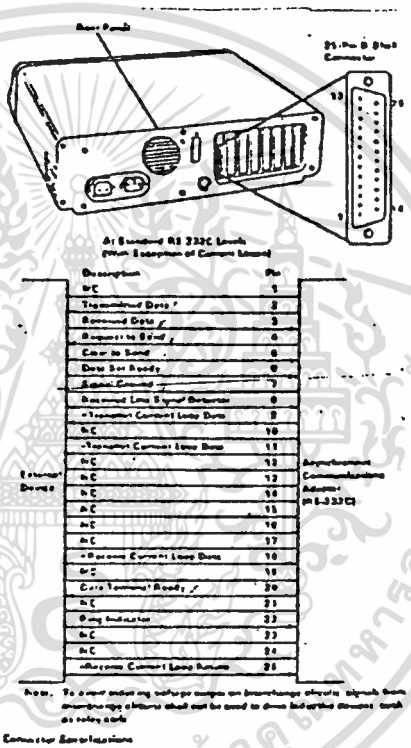
มาตรฐานที่คุ้นหูคุ้นตากันมากสำหรับประเทศไทยเราก็คงจะเป็น EIA หรือมาตรฐาน ที่ขึ้นต้นด้วย RS นั่นเอง มาตรฐานของ EIA (The Electronics Industrial Association) หรือสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ของสหรัฐอเมริกานั้นได้กำหนดมาตรฐานต่าง ๆ สำหรับใช้กับอุปกรณ์ทางด้านอิเล็กทรอนิกส์ซึ่งรวมทั้งอุปกรณ์ทางด้านการสื่อสารข้อมูลด้วยมากมาย ที่รู้จักกันดีที่สุดในวงการคอมพิวเตอร์เมืองไทยก็คือ RS-232 (RS ย่อมาจาก Recommended Standard) หรือบางทีเรียกกันว่า EIA-232 เป็นมาตรฐานที่กำหนดการสัญญาณและการเชื่อมต่อของสัญญาณสำหรับการสื่อสารข้อมูลแบบอนุกรม สำหรับเมืองไทย เราจะคุ้นเคยกันในชื่อของ RS-232 หรือบางทีก็เรียกกันทั่วไปว่าซีเรียลพอร์ต (Serial Port) ทั้งที่แท้จริงแล้วซีเรียลพอร์ตจะหมายถึงพอร์ตที่ส่งข้อมูลในแบบอนุกรมทั้งหมด

ข้อกำหนดของ EIA กำหนดให้อุปกรณ์ในการสื่อสารข้อมูลมี 2 ชนิด คือ DTE หรือ Data Terminal Equipment และ DCE หรือ Data Communication Equipment ซึ่งต่อมาได้เปลี่ยนเป็น Data Circuit Terminating Equipment อุปกรณ์ที่เป็น DTE นั้น จะอยู่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
008436

ปลายของสายสัญญาณสำหรับการสื่อสารข้อมูล ตัวอย่างของ DTE ที่เห็นได้ชัด ก็คือ เครื่องคอมพิวเตอร์นั่นเอง ส่วน DCE นั้นจะทำหน้าที่แปลงสัญญาณที่ได้จาก DTE ให้อยู่ในสภาพที่เหมาะสมที่จะส่งไปตามสายสัญญาณ และทำหน้าที่แปลงสัญญาณที่ได้รับจากสายสัญญาณให้เป็นข้อมูล เพื่อจะส่งให้ DTE ต่อไป

EIA-232 กำหนดให้การเชื่อมต่อระหว่าง DTE และ DCE จะต้องใช้สัญญาณชุดหนึ่งประกอบด้วยสัญญาณต่าง ๆ 21 เส้น แต่ในการใช้งานมักจะใช้เพียง 9 เส้นเท่านั้น ขาสัญญาณที่ใช้งานอยู่เป็นประจำจะมีลักษณะดังรูปที่ 2

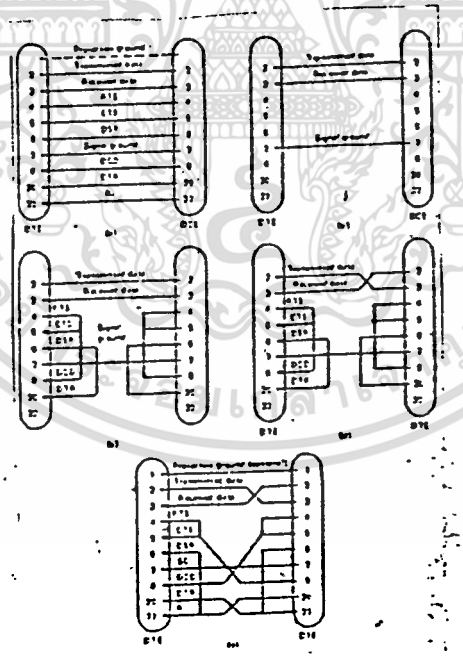


รูปที่ 2 ขาสัญญาณตามมาตรฐาน RS-232 ที่ใช้กันทั่วไปบนเครื่องคอมพิวเตอร์

ในการเชื่อมต่อระหว่างอุปกรณ์โดยใช้ชุดสัญญาณของ RS-232 นั้นสามารถทำได้หลายวิธี ขึ้นอยู่กับว่าเป็นการเชื่อมต่อระหว่างอุปกรณ์ชนิดใด และความต้องการของผู้ใช้ที่ว่า ต้องการให้มีการตรวจสอบฝ่ายตรงข้าม หรือที่เรียกว่าการทำ Hand Shake หรือไม่ การเชื่อมต่อดังกล่าวได้แสดงไว้ในรูปที่ 3 รูป(a) เป็นการเชื่อมต่อระหว่าง DTE กับ DCE ในแบบ FDX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน รูป(b) เป็นการเชื่อมต่อระหว่าง DTE กับ DCE ในลักษณะของ FDX แบบประหยัด คือใช้สายสัญญาณเพียง 3 เส้นเท่านั้น รูป(c) เป็นการเชื่อมต่อระหว่าง DTE กับ DCE ในแบบ Loop Back วิธีนี้จะไม่มีการทำ Hand Shake ระหว่างอุปกรณ์ทั้งสองตัว แต่จะเป็นการหลอกตัวเอง โดยใช้การป้อนกลับของสัญญาณเอาต์พุตกลับไปให้อินพุตของตัวเอง รูป(d) เป็นการต่อระหว่าง DTE กับ DTE ด้วยกัน (อาจจะต่อระหว่าง DCE กับ DCE ด้วยกันก็ได้) และมีการทำ Loop Back สัง เกตที่สัญญาณ TxD และ RxD ของแต่ละตัวจะมีการสลับกัน ลักษณะนี้เรียกว่า Null Modem รูป(e) เป็นการต่อแบบ Null Modem พร้อมกับมีการทำ Hand Shake ระหว่างอุปกรณ์ทั้งสอง เรียกว่า เป็นการทำ Double-Cross ในการเชื่อมต่อ ระหว่างคอมพิวเตอร์ซึ่งเป็น DTE กับโมเด็มซึ่งเป็น DCE จะใช้การเชื่อมต่อในแบบแรก (แบบ a)ส่วนการเชื่อมต่อระหว่างคอมพิวเตอร์ด้วยกัน อาจจะใช้แบบ(d) หรือ(e)ก็ได้ ขึ้นอยู่กับว่าต้องการให้มี Hand Shake หรือไม่ การทำ Hand Shake จะได้ประโยชน์ในด้านที่สามารถรับประกันได้ว่าข้อมูลทางด้านผู้ส่งจะถูกส่งออกไป เมื่อทางผู้รับพร้อมที่จะรับเท่านั้น เท่ากับเป็นการป้องกันความผิดพลาดที่อาจจะ เกิดขึ้นได้ อย่างไรก็ตาม การเขียนโปรแกรม เพื่อควบคุมก็จะยุ่งยากมากขึ้น



รูปที่ 3 การเชื่อมต่อระหว่างอุปกรณ์ในมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CCITT

นอกจากมาตรฐานของ EIA แล้วในด้านการสื่อสารยังมีอีกองค์กรหนึ่งที่จะต้องพูดถึงนั่นคือ CCITT หรือ The Consultative Commitee International Telegraphy and Telephony หน่วยงานหนึ่งของ ITU (International Telecommunication Union) องค์กรระหว่างประเทศที่ทำหน้าที่กำหนดมาตรฐานของการสื่อสารระหว่างประเทศทั้งประเภทการสื่อสารด้วยสายสัญญาณและการสื่อสารแบบไร้สาย CCITT จะเป็นกลุ่มที่ทำหน้าที่ ออกมาตรฐานเกี่ยวกับการสื่อสารด้วยโทรศัพท์และโทรเลข ได้ออกมาตรฐานการสื่อสารข้อมูลออกมามากมาย ที่สำคัญคือ V.28 ที่สามารถใช้แทน RS-232 ได้ และ X.25 ซึ่งเป็นโปรโตคอลการสื่อสารข้อมูลที่ใช้กันแพร่หลายทั่วโลกเป็นต้น รูปที่ 4 เป็นตัวอย่างของมาตรฐาน ที่ทาง CCITT ได้นำออกมาให้ใช้ เป็นที่น่าสังเกตอยู่ว่า CCITT ใช้คำว่า Recommendations ซึ่งแปลว่าข้อแนะนำแทนที่จะเป็น Standard ที่แปลว่ามาตรฐานทั้งนี้เพราะว่าทาง CCITT ไม่มีอำนาจที่จะไปบังคับให้ประเทศต่าง ๆ ใช้ข้อกำหนดดังกล่าวได้ แต่ถ้าไม่ใช้ ก็คงจะติดต่อสื่อสารกับประเทศอื่นไม่ได้

มาตรฐานหรือข้อแนะนำที่ทาง CCITT กำหนดออกมานั้น จะมีเป็นฉบับต่าง ๆ กันทุก ๆ ปี โดยจะมีการประชุมหารือกันระหว่างประเทศสมาชิก เพื่อหารือและกำหนดข้อแนะนำใหม่ ๆ ออกมาให้ใช้กัน การออกข้อแนะนำ หรือความจริงก็คือข้อบังคับนั่นเอง จะออกกันเป็น Series ตั้งแต่ A ไปจนถึง Z ในแต่ละตัวอักษรจะเป็นเรื่องย่อย ๆ ต่าง ๆ กันมากมาย นอกจากนี้ยังมีการกำหนดสีและปีอีกด้วย ดังนั้นการอ้างอิงถึงมาตรฐานของ CCITT จึงจะต้องระบุเป็นปีและสี เช่น CCITT Recommendation Orange Book Series X จะเป็นการกล่าวถึง Pubic Data Network Data Transittion ดังรูปที่ 4

The X Series of CCITT for data transmission over public data networks.

Number	Title
X.1	International user classes of service in public data networks
X.2	International user facilities in public data networks
X.4	General structure of signals of International Alphabet No 5 code for data transmission over public data networks
X.20	Interface between data terminal equipment and data circuit terminating equipment for start-stop transmission services on public data networks
X.21	General purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operation on public data networks
X.24	List of definitions of interchange circuits between data terminal equipment and data circuit-terminating equipment on public data networks

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA TRANSMISSION RECOMMENDATIONS OF THE CCITT

X.25	Interface between data terminal equipment and data circuit-terminating equipment for terminals operating in the packet mode on public data networks
X.26	Electrical characteristics for unbalanced double-current interchange circuits for general use with integrated circuit equipment in the field of data communications
X.27	Electrical characteristics for balanced double-current interchange circuits for general use with integrated circuit equipment in the field of data communications
X.30	Standardization of basic model page printing machine in accordance with International Alphabet No. 5
X.31	Characteristics for start-stop data terminal equipment using International Alphabet No 5
X.32	Answer back units for 200 baud start-stop machines in accordance with International Alphabet No 5
X.33	Standardization of an international test for the measurement of the margin of start-stop machines in accordance with International Alphabet No. 5
X.40	Standardization of frequency-shift modulated transmission systems for the provision of telegraph and data channels by frequency division of a primary group
X.50	Fundamental parameters of a multiplexing scheme for the international interface between synchronous data networks
Number	Title
X.51	Fundamental parameters of a multiplexing scheme for the international interface between synchronous data networks using 10-bit envelope structure
X.60	Common channel signaling for synchronous data applications-data user part
X.70	Terminal and transit control signaling for start-stop services on international circuits between asynchronous data networks
X.71	Decentralized terminal and transit control signaling system on international circuits between synchronous data networks
X.92	Hypothetical reference connections for public synchronous data networks
X.95	Network parameters in public data networks
X.96	Call progress signals in public data networks

The V Series of CCITT for data transmission over the voice circuit or analog networks

V.1	Equivalence between binary notation symbols and the significant conditions of a two-condition code
V.2	Power levels for data transmission over telephone lines
V.3	International Alphabet No. 5 for transmission of data and messages
V.4	General structure of signals of the 5-unit code for data and message transmission
V.10	Use of the telex network for data transmission at the modulation rate of 50 baud
V.11	Automatic calling and/or answering on the telex network
V.13	Answer-back unit simulators
V.15	Use of acoustic couplers for data transmission
V.21	200 baud modem standardized for use in the general switched telephone network
V.22	Standardization of modulation rates and data signaling rates for synchronous data transmission in the general switched telephone network
V.22B	Standardization of modulation rates and data signaling rates on leased telephone circuits
V.23	600/1200 baud modem standardized for use on the general switched telephone networks
V.24	Functions and electrical characteristics of circuits at the interface between data terminal equipment and data circuit-terminating equipment
V.25	Automatic calling and/or answering on the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	general switched telephone network
V.26	2400 bits/second modem for use on four-wire leased point-to-point circuits
V.26B	2400 bits/second modem for use on the general switched telephone network
V.27	Modem for data signaling rate up to 4800 bits/second over leased circuits

Number	Title
V.28	Electrical characteristics for interface circuits
V.30	Parallel data transmission system for universal use on the general switched telephone network
V.31	Electrical characteristics for contact closure-type interface circuits
V.35	Transmission of 48 kilobits/second data using 60 to 108 kHz group bank circuits
V.40	Error indication with electromechanical equipment
V.41	Code-independent error control system
V.50	Standard limits for transmission quality of data transmission
V.51	Organization of the maintenance of international telephone-type circuits used for data transmission
V.52	Characteristics of distortion and error rate measuring apparatus for data transmission
V.53	Limits for the maintenance of telephone-type circuits used for data transmission
V.56	Comprehensive tests for modems that use their own interface circuits
V.57	Comprehensive test set for high data-transmission rates

รูปที่ 4 ตัวอย่างของ CCITT Recommendation Orange Book Series X ที่กล่าวถึงการถ่ายทอดข้อมูลในโครงข่ายข้อมูลสาธารณะ

ANSI และ ISO

ชื่อ ISO นี้น่าจะเป็นที่คุ้นหูของผู้ที่อยู่ในวงการอุตสาหกรรมเป็นอย่างมาก เพราะ ISO เป็นองค์กรระหว่างประเทศที่ทำหน้าที่กำหนดมาตรฐานต่าง ๆ ทางด้านอุตสาหกรรมชื่อเต็ม ๆ ของ ISO ก็คือ International Standard Organization มีที่ทำการอยู่ที่เจนีวาประเทศสวิสเซอร์แลนด์ ISO จะประสานงานกับ CCITT อย่างใกล้ชิด ส่วน ANSI นั้นก็คือ องค์การที่ทำหน้าที่กำหนดมาตรฐานผลิตภัณฑ์อุตสาหกรรมของอเมริกาชื่อของ ANSI นั้นจะเป็นที่รู้จักกันมากในวงการ BBS บ้านเราในฐานะของเจ้าของรหัสมหัศจรรย์ที่สามารถทำให้เราสามารถควบคุมการทำงานของจอภาพ ทางด้านการแสดงสีและเลื่อนเคอร์เซอร์ตลอดจนการควบคุมการใช้งานคีย์บอร์ดจากคอมพิวเตอร์ที่อยู่อีกแห่งหนึ่งได้ทั้งนี้ จุดประสงค์เดิมของการกำหนดชุดรหัสดังกล่าวนั้นมีขึ้นสำหรับใช้ควบคุมจอเทอร์มินัลสำหรับเครื่องคอมพิวเตอร์ในระดับเมนเฟรม มีชื่อเรียกกันว่า ANSI Terminal ต่อมาเมื่อไมโครคอมพิวเตอร์มีราคาถูกลง มีการนำเอาไมโครคอมพิวเตอร์มาใช้เป็นเทอร์มินัลแทนเทอร์มินัลแบบเดิมจึงจำเป็นต้องทำให้ไมโครคอมพิวเตอร์สามารถรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสของ ANSIได้ สำหรับเครื่องในตระกูลไอบีเอ็มนั้น จะสามารถใช้รหัสของ ANSIได้โดยการติดตั้งดีไวซ์ไดร์เวอร์ที่ชื่อว่า ANSI.SYS ซึ่งมากับ DOS ทุกเวอร์ชันอยู่แล้ว

Bell Lab

Bell เป็นองค์การของเอกชนที่ควบคุมทางด้านการสื่อสารของสหรัฐอเมริกา เนื่องจากระบบโทรศัพท์ของสหรัฐอเมริกาส่วนใหญ่เป็นของ Bell ดังนั้น มาตรฐานที่กำหนดโดย Bell จึงเปรียบเสมือนกับเป็นมาตรฐานของระบบโทรศัพท์ของสหรัฐอเมริกา อย่างไรก็ตาม อุปกรณ์ที่ตรงตามมาตรฐานของ Bell ส่วนใหญ่จะสามารถใช้งานร่วมกับอุปกรณ์ที่ตรงตามมาตรฐานของ CCITT เช่น Bell 212A จะเทียบเท่ากับ CCITT V.22 ซึ่งเป็นมาตรฐานสำหรับโมเด็มขนาด 1200 bps เป็นต้น

มาตรฐานต่าง ๆ สำหรับการสื่อสารข้อมูลยังมีอยู่อีกมากมาย และมักจะเป็นสิ่งที่เรามองข้ามอยู่เสมอทั้ง ๆ ที่มาตรฐานเหล่านี้เกี่ยวข้องกับเราอยู่ไม่น้อยทีเดียว โดยเฉพาะบรรดาผู้ที่มีหน้าที่จะต้องออกแบบหรือใช้งานอุปกรณ์กับระบบการสื่อสารต่าง ๆ น่าที่จะศึกษาไว้บ้าง เพื่อยกระดับฐานะของระบบการสื่อสารของเราให้ทัดเทียมกับสากล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

พอร์ตอนุกรมของ ไอบี เอ็มพีซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตอนุกรมของไอบี เอ็มพีซี

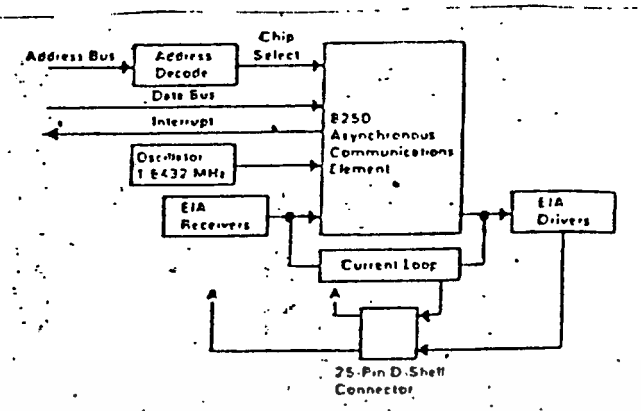
หน้าที่ของพอร์ตอนุกรม

จุดประสงค์แรกเริ่มที่มีการประดิษฐ์พอร์ตอนุกรมขึ้นมานั้น ก็เพื่อใช้เป็นพอร์ตสำหรับสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ด้วยกัน ดังจะเห็นได้จากที่บางครั้งจะมีคนเรียกพอร์ตนี้ว่า Communication Port แต่เนื่องจากในช่วงแรก ๆ ที่เรารู้จักพอร์ตอนุกรมจากเครื่องคอมพิวเตอร์ในระดับไมโครคอมพิวเตอร์นั้นการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ไม่ใช่เรื่องที่น่าสนใจนัก จึงทำให้ความหมายของพอร์ตชนิดนี้เปลี่ยนแปลงไปจนกระทั่งมีผู้ประยุกต์นำเอาพอร์ตอนุกรมมาใช้ต่อกับเมาส์ จึงทำให้มันเริ่มมีความสำคัญมากขึ้นอีกครั้ง

โครงสร้างของพอร์ตอนุกรมบนไอบี เอ็มพีซี

ในรูปที่ 1 เป็นการแสดงถึงส่วนประกอบต่าง ๆ ของพอร์ตอนุกรม จะเห็นว่าพอร์ตอนุกรมบนไอบี เอ็มพีซีมีชิป 8250 เป็นหัวใจ ทุก ๆ ส่วนที่เกี่ยวข้องกับพอร์ตอนุกรมจะต้องมีการติดต่อกับชิปนี้ ไม่มีทางตรงก็ทางอ้อม ส่วนอื่น ๆ นอกเหนือจากนี้ก็มีการติดต่อกับซีพียู จะมีการ Decode ค่าของ Address Bus ตามโครงการทั่วไปของระบบคอมพิวเตอร์ ส่วนที่เป็น Oscillator จะทำหน้าที่สร้างสัญญาณนาฬิกาเพื่อใช้เป็นเวลาอ้างอิงสำหรับชิป 8250 ในการทำงานต่าง ๆ เช่นในการรับส่งข้อมูล เป็นต้น ส่วนไดร์เวอร์ทำหน้าที่ขับสัญญาณให้มีคุณสมบัติตามมาตรฐาน EIA-232 ทั้งนี้การเชื่อมต่อระหว่างอุปกรณ์ผ่านซีเรียลพอร์ตของไอบี เอ็มพีซีจะใช้มาตรฐานของ EIA-232(หรือ RS-232) เป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Asynchronous Communications Adapter Block Diagram

รูปที่ 1 โครงสร้างของพอร์ตอนุกรมของไอบีเอ็มพีซี

8250 พอร์ตอนุกรมสำเร็จรูป

ชิป 8250 ผลิตโดยบริษัท Intel เป็นชิปที่ทำหน้าที่สำหรับการสื่อสารแบบ Serial Asynchronous โดยเฉพาะ สัญญาณต่าง ๆ จะคอมแพทกับมาตรฐานของ EIA-232 โดยต้องใช้ LineDriver เป็นตัวขับสัญญาณ การใช้งาน 8250 นั้นจะกระทำโดยการโปรแกรมค่าของริจิสเตอร์ต่าง ๆ ภายในชิปให้เป็นไปตามต้องการ ริจิสเตอร์ต่าง ๆ ภายใน 8250 จะมีแอดเดรสดังรูปที่ 2 โดย Primary Address จะเป็นรีจิสเตอร์สำหรับ COM1 และ Secondary Address จะเป็นริจิสเตอร์สำหรับ COM2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O Decode (in Hex)			
Primary Adapter	Alternate Adapter	Register Selected	DLAB State
3F8	2F8	TX Buffer	DLAB=0 (Write)
3F8	2F8	RX Buffer	DLAB=0 (Read)
3F8	2F8	Divisor Latch LSB	DLAB=1
3F9	2F9	Divisor Latch MSB	DLAB=1
3F9	2F9	Interrupt Enable Register	
3FA	3FA	Interrupt Identification Registers	
3FB	2FB	Line Control Register	
3FC	2FC	Modem Control Register	
3FD	2FD	Line Status Register	
3FE	2FE	Modem Status Register	

I/O Decodes

Hex Address 3F8 to 3FF and 2F8 to 2FF											
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	DLAB	Register
1	1=C	1	1	1	1	1	x	x	x		
							0	0	0	0	Receive Buffer (read), Transmit Holding Reg (write)
							0	0	1	0	Interrupt Enable
							0	1	0	x	Interrupt Identification
							0	1	1	x	Line Control
							1	0	0	x	Modem Control
							1	0	1	x	Line Status
							1	1	0	x	Modem Status
							1	1	1	x	None
							0	0	0	1	Divisor Latch (LSB)
							0	0	1	1	Divisor Latch (MSB)

Note: A9 will be logical 1 for the adapter designated as primary or a logical 0 for the adapter designated as alternate (as defined by the address jumper module on the adapter).

A2, A1 and A0 bits are "don't cares" and are used to select the different registers of the communications chip.

Address Bits

รูปที่ 2 แอดเดรสของรีจิสเตอร์ต่าง ๆ ของ 8250 Primary Address จะเป็นรีจิสเตอร์ของ COM1 และ Secondary Address จะเป็นของ COM2

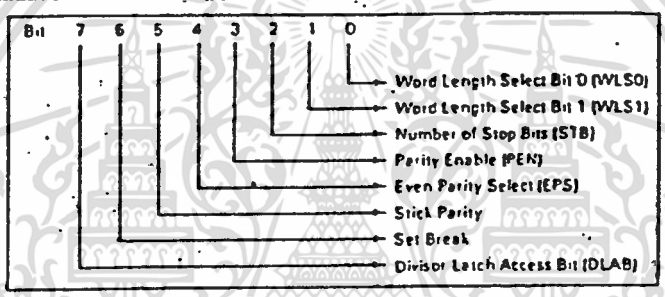
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ริจิส เตอร์ของ 8250

จากนี้ไป เราจะมาดูถึงหน้าที่ของริจิส เตอร์ต่าง ๆ ของ 8250 ว่ามีหน้าที่ใดบ้าง

Line Control Register (LCR)

LCR จะอยู่ที่ Offset ที่ 3 นับจาก Base Address เช่น พอร์ต COM1 มี Base Address ที่ 03FB เราจะได้ตำแหน่งของ LCR สำหรับ COM1 ที่ 03F8+3 คือที่ 03FB นั่นเอง LCR เป็นริจิส เตอร์สำหรับการควบคุมฟอร์แมตของข้อมูลที่ใช้ในการสื่อสาร เช่น ความยาวของข้อมูล จำนวน Stop Bit ชนิดของ Parity Bit และอื่น ๆ ดังรูปที่ 3



รูปที่ 3 หน้าที่ของบิตต่าง ๆ ใน LCR

บิตที่ 0 และ 1 เป็นบิตที่ให้ควบคุมความยาวของข้อมูล สามารถกำหนดได้ตั้งแต่ 5 ถึง 8 บิตดังตารางที่ 1

Bit 0	Bit 1	Word Length
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 2 จะเป็นตัวกำหนดความยาวของ Stop Bit โดยถ้าหากบิต 2 มีค่าเป็น 0 จะเป็นการเลือกใช้ Stop Bit ที่ยาวเพียงบิตเดียว แต่ถ้าบิต 2 เป็น 1 และเลือกใช้ความยาวของเวิร์ดเป็น 5 บิตจะเป็นการเลือกใช้ 1.5 Stop Bit และถ้าความยาวของเวิร์ดเป็น 6 ถึง 8 บิต จะเป็นการเลือกใช้ 2 Stop Bit

ถ้าข้อมูลในบิตที่ 3 มีค่าเป็น 1 จะหมายความถึงการใช้ Parity Bit ในการฟอร์แมต แต่ถ้าเป็น 0 จะหมายความว่าไม่ใช่ Parity Bit ในการสื่อสาร

บิตที่ 4 จะเป็นการเลือกใช้ว่า ถ้าหากจะใช้ Parity Bit แล้ว จะเลือกใช้แบบใดในกรณีที่บิตนี้มีค่าเป็น 1 จะเป็นการใช้ Even Parity และถ้าเป็น 0 จะเป็นการใช้ Odd Parity ทั้งนี้บิตนี้จะไม่มีความหมายถ้าหากบิตที่ 3 มีค่าเป็น 0

บิตที่ 5 จะเป็นบิตที่มีความซับซ้อนเล็กน้อย และไม่ค่อยได้ใช้ในการสื่อสารทั่วไป มักจะเป็นการใช้ในการ Initial การสื่อสาร ถ้าหากบิตที่ 5 นี้มีค่าเป็น 1 พร้อมกับบิตที่ 3 Parity Bit ที่ถูกส่งออกไปจะมีค่าตรงข้ามกับค่าในบิตที่ 4 เช่นถ้าบิตที่ 4 มีค่าเป็น 0 ในขณะที่บิตที่ 3 และที่ 5 มีค่าเป็นหนึ่งแล้ว ค่า 1 จะถูกส่งออกไปในตำแหน่งของ Parity Bit ของข้อมูล แต่ถ้าบิตที่ 4 มีค่าเป็น 1 ข้อมูลที่ถูกส่งออกไปจะมีค่าเป็น 0 แทน

เมื่อใดก็ตามที่ข้อมูลในบิตที่ 6 มีค่าเป็น 1 ระดับสัญญาณที่ขา SCOUT ซึ่งเป็นเอาต์พุตของข้อมูลที่ถูกส่งออกไปยังจุดหมายปลายทางจะถูกกดให้เป็น 0 อยู่ตลอดเวลา ไม่ว่าในขณะนี้จะมีการส่งข้อมูลหรือไม่ก็ตาม จนกว่าจะมีการเคลียร์ค่าของบิตที่ 6 นี้ให้เป็น 0 ค่าของเอาต์พุตดังกล่าวจึงจะขึ้นอยู่กับข้อมูลที่ส่งออกไปอีกครั้ง บิตนี้จะไม่ใช้งานในการใช้งานปกติ

บิตที่ 7 จะเป็นบิตพิเศษที่ใช้ภายใน 8250 เอง เรียกว่า Divisor Latch Access Bit หรือ DLAB ในกรณีที่ DLAB มีค่าเป็น 1 จะเป็นการบอก 8250 ว่า เราต้องการโปรแกรมค่าของ Bit Rate และถ้าหากเป็น 0 จะเป็นการใช้งานปกติ

Divisor Latch Least Singnificant (DLL)

DLL จะอยู่ที่ตำแหน่ง Base Address ของพอร์ต เช่น 03F8 สำหรับ COM1 หรือ 02F8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ COM2 ทั้งนี้ตำแหน่งนี้จะเป็น DULL ก็ต่อเมื่อ DLAB มีค่าเป็น 1 ค่าที่เราจะใส่ใน DULL จะเป็นไบต์ที่ต่ำของตัวหารที่จะนำไปใช้ในการหารสัญญาณนาฬิกา 1.8432 MHz ให้มีค่าเป็น 0 เป็น Bit Rate สำหรับการส่งข้อมูล ในกรณีที่ DLAB มีค่าเป็น 0 ริจิสเตอร์ DULL ตัวนี้จะกลายเป็น RBR หรือ THR ซึ่งจะกล่าวต่อไปในภายหลัง

Divisor Latch Most Significant (DLM)

DLM จะอยู่ที่ Offset ที่ 1 นับจาก Base Address เช่น 03F8+1 คือ 03F9 สำหรับ COM1 เป็นต้น เป็นริจิสเตอร์สำหรับใส่ค่าไบต์สูงของตัวหารสัญญาณนาฬิกาเพื่อให้ได้ Bit Rate ที่ต้องการ เช่นเดียวกับ DULL ตำแหน่งที่ DLM อยู่จะทำหน้าที่เป็น DLM ก็ต่อเมื่อ DLAB เป็น 1 ถ้าหากเป็น 0 ก็จะกลายเป็น IER ซึ่งจะกล่าวต่อไปนี้

การโปรแกรมค่า Bit Rate นั้น จะคำนวณโดยใช้สูตร

$$\text{Divisor} = \frac{1.8432\ \text{MHz}}{\text{Bit Rate} \times 16}$$

โดยค่าที่ Divisor จะต้องปัดให้เป็นเลขจำนวนเต็ม และจะต้องมีค่าไม่เกิน 65535 เราจะนำค่านี้ไปใส่ใน DULL และ DLM เพื่อให้ได้ Bit Rate ที่ต้องการ การปัดให้เป็นจำนวนเต็มนี้อาจจะทำให้เกิดค่าความผิดพลาดขึ้น เราจำเป็นจะต้องคำนวนค่าความผิดพลาดแล้วดูด้วยว่ามีค่ามากเกินไปหรือไม่ ค่าผิดพลาดที่ดีไม่ควรจะเกิน 0.5 เปอร์เซนต์ การคำนวนหาค่าผิดพลาดนั้น คำนวนได้จากสูตร

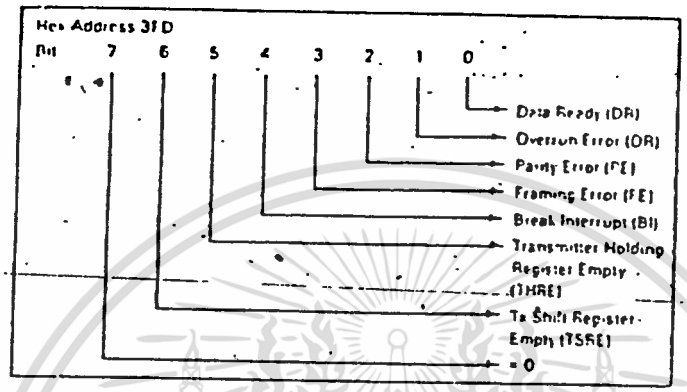
$$\text{Error}(\%) = \frac{\text{Bit Rate} - \text{Actual Bit Rate} \times 100}{\text{Bit Rate}}$$

โดยค่า Bit Rate ก็คือค่า Bit Rate ที่เราต้องการ ส่วน Actual Bit Rate ก็คือ Bit Rate ที่ได้จากการปัดเศษตัวหารแล้วคำนวณกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line Status Register (LSR)

LSR เป็นริจิสเตอร์ที่อยู่ที่ Offset ที่ 5 นับจาก Base Address ทำหน้าที่ เก็บค่าสถานะต่าง ๆ ในขณะนี้ไว้ โดยมีฟอร์แมตดังรูปที่ 4



รูปที่ 4 ตำแหน่งและหน้าที่ของบิตต่าง ๆ ใน LSR

บิตที่ 0 หรือ DR (Data Ready) เมื่อมีค่าเป็น 1 จะหมายความว่าได้รับข้อมูลจากทางผู้ส่งและข้อมูลได้รออยู่ในบัฟเฟอร์เรียบร้อยแล้ว

บิตที่ 1 หรือ OR (Overrun Error) เมื่อข้อมูลมีค่าเป็น 1 จะแสดงว่าเกิดการ Overrun คือ ข้อมูลที่ได้รับในบัฟเฟอร์ได้ถูกเขียนทับโดยข้อมูลที่เข้ามาใหม่เรียบร้อยแล้ว นั่นคือข้อมูลที่อยู่ในขณะนี้เป็นข้อมูลที่เกิดการสูญหาย อาจจะนำไปใช้งานไม่ได้อีกต่อไป

บิตที่ 2 หรือ PE (Parity Error) เมื่อมีค่าเป็น 1 แสดงว่าข้อมูลที่ได้รับมี Parity ที่ได้รับไม่ตรงกัน หรือพูดง่าย ๆ ก็คือ ข้อมูลที่ได้รับอาจจะผิดพลาดนั่นเอง

บิตที่ 3 หรือ FE (Framing Error) เมื่อมีค่าเป็น 1 แสดงว่า Stop Bit ผิดพลาดหรือสูญหายได้

บิตที่ 4 หรือ BI (Break Interrupt) จะมีค่าเป็น 1 เมื่อ 8250 ได้รับลอจิก 0 จากทางผู้ส่งเป็นระยะเวลานานเกินกว่าความยาวของแต่ละเวิร์ด แสดงว่าต้นทางได้ทำการ Break โดยการเซ็ตบิตที่ 6 ของ LSR ให้เป็น 1

บิตที่ 5 หรือ THRE (Transmitter Holding Register Empty) เป็นบิตที่บอกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

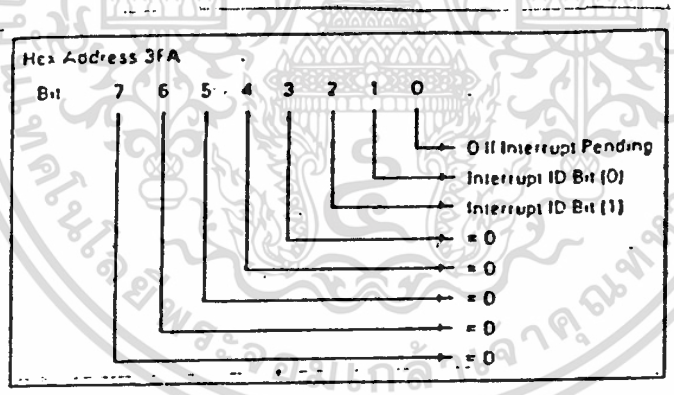
บัฟเฟอร์สำหรับการส่งข้อมูลออกนั้นได้ว่างแล้วโดยจะมีค่าเป็น 1 ถ้าหากบัฟเฟอร์ดังกล่าวว่างลง

บิตที่ 6 หรือ TSRE (Transmitter Shift Register Empty) เป็นบิตที่เมื่อเซ็ตเป็น 1 แล้วจะหมายความว่าขณะนี้บัฟเฟอร์ที่ทำหน้าที่แปลงข้อมูลจากขนานไปเป็นอนุกรมได้ว่างลงแล้ว พร้อมที่จะรับข้อมูลตัวต่อไป ริจิสเตอร์ตัวนี้จะรับข้อมูลโดยตรงจาก THR หรือ Transmitter Holding Register อีกทีหนึ่งไม่สามารถเรียกใช้งานโดยตรงได้

บิตที่ 7 ไม่มีความหมายใด ๆ แต่จะมีค่าเป็น 0 ตลอดเวลา เมื่อใดที่อ่านแล้วได้ข้อมูลเป็น 1 อาจเป็นไปได้ 2 กรณีคือ อ่านผิดที่หรือ 8250 เสีย

Interrupt Identification Register (IIR)

หน้าที่ของ IIR ก็คือ รายงานประเภทของการอินเทอร์รัปต์ที่เกิดขึ้นมีที่ใช้น้อยมาก เพราะการเขียนโปรแกรมให้ใช้การอินเทอร์รัปต์นั้นยุ่งยากมาก หน้าที่และตำแหน่งของบิตต่าง ๆ ของ IIR แสดงไว้ในรูปที่ 5 IIR จะอยู่ที่ Offset ที่ 2 จาก Base Address



รูปที่ 5 หน้าที่และตำแหน่งของบิตต่าง ๆ ของ IIR

บิตที่ 0 ของ IIR จะเป็นบิตสำหรับตรวจสอบว่ามีอินเทอร์รัปต์เกิดขึ้นหรือไม่ ถ้าหากบิตนี้ถูกเซ็ตให้เป็น 1 หมายความว่าไม่มีการใช้อินเทอร์รัปต์สำหรับการสื่อสาร ในกรณีที่ต้องการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตอนุกรมนี้ในลักษณะของการอินเทอร์รัปต์ เราจะต้องมีการโปรแกรมชิป 8259 อีกชั้นหนึ่ง โดยกำหนดค่าต่าง ๆ ของริจิสเตอร์ใน 8259 ให้อนุญาตให้พอร์ตอนุกรมสามารถร้องขอการอินเทอร์รัปต์ได้ สำหรับวิธีการโปรแกรม 8259 นั้นสามารถหาได้จากหนังสือคู่มือ 8259 ทั่วไป ในกรณีที่เราไม่ได้ทำการเซ็ตค่าของ 8259 เพื่อใช้งานพอร์ตในลักษณะของการอินเทอร์รัปต์ เราอาจสามารถใช้งานในลักษณะกึ่งอินเทอร์รัปต์ได้โดยการเซ็ตค่าต่าง ๆ ในริจิสเตอร์ IER ที่จะกล่าวต่อไป และเขียนโปรแกรมทำการตรวจสอบค่าของ IER เพื่อตรวจสอบการอินเทอร์รัปต์ด้วยตนเองอีกทีหนึ่ง

Interrupt ID Register				Interrupt Set and Reset Functions		
Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	1		None	None	
1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Interrupt	Reading the Line Status Register
1	0	0	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register
0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register
0	0	0	Fourth	Modem Status	Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect	Reading the Modem Status Register

ตารางที่ 2 แสดงประเภทของอินเทอร์รัปต์ที่เกิดขึ้นโดย IIR

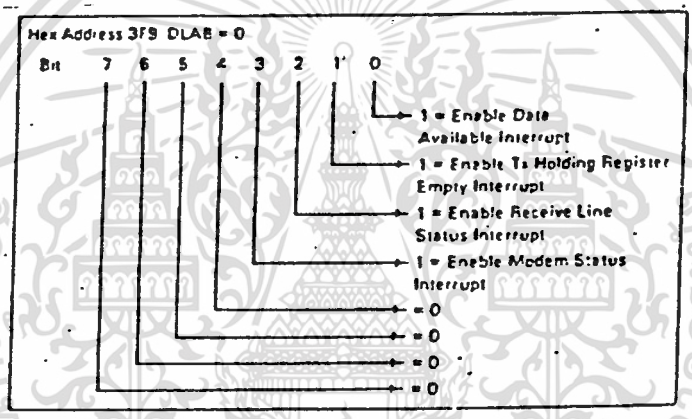
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 1 และบิตที่ 2 จะเป็นตัวบอกว่าอินเทอร์รัปต์ที่เกิดขึ้นเป็นอินเทอร์รัปต์ประเภทใดซึ่งมีรายละเอียดดังตารางที่ 2

บิตที่ 3 ถึง 7 จะมีค่าเป็น 0 ตลอดและไม่มีความหมายใด ๆ

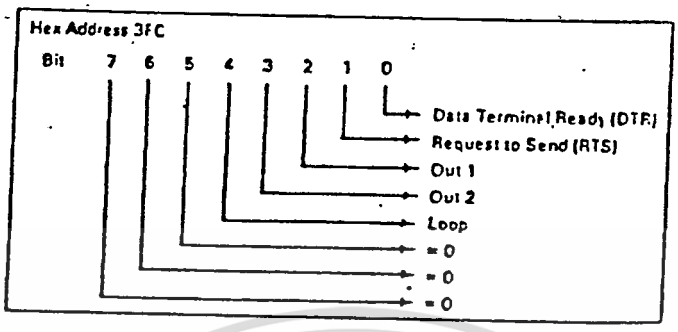
Interrupt Enable Register (IER)

ริจิสเตอร์ตัวนี้มักจะใช้คู่กันกับ IIR เพราะเป็นริจิสเตอร์สำหรับตรวจสอบสถานะ แต่ IER เป็นริจิสเตอร์สำหรับกำหนดสถานะการอินเทอร์รัปต์ต่าง ๆ มีตำแหน่งดังรูปที่ 6 ริจิสเตอร์ตัวนี้จะอยู่ที่ Offset ที่ 1 และ DLAB จะต้องเป็น 0



รูปที่ 6 ตำแหน่งและหน้าที่ของบิตต่าง ๆ ใน IER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ตำแหน่งและบิตต่าง ๆ ของ MCR

บิตที่ 0 ถ้าเซ็ตให้เป็น 1 จะเป็นการอนุญาตให้มีการอินเทอร์รัปต์เมื่อมีข้อมูลมารออยู่ในบัฟเฟอร์ข้อมูล

บิตที่ 1 จะเป็นการอนุญาตให้มีการอินเทอร์รัปต์เมื่อบัฟเฟอร์สำหรับข้อมูลขาออกว่างลง ถ้าเซ็ตให้มีค่าเป็น 1

บิตที่ 2 เมื่อมีค่าเป็น 1 จะเป็นการอนุญาตให้อินเทอร์รัปต์ถ้ามีการเปลี่ยนแปลงของข้อมูลใน LSR หรือมีการเปลี่ยนแปลงของสถานะของระบบนั่นเอง

บิตที่ 3 เมื่อมีค่าเป็น 1 จะเป็นการอนุญาตให้มีการอินเทอร์รัปต์เมื่อมีการเปลี่ยนแปลงสถานะของสัญญาณควบคุมโมเด็ม

บิตที่เหลือจะมีค่าเป็น 0 ตลอดเวลา

Modem Control Register (MCR)

MCR เป็นริจิสเตอร์สำหรับให้เราแก้ไขเปลี่ยนแปลงสถานะของสัญญาณควบคุมโมเด็ม ตามมาตรฐาน EIA-232 เช่น DTR หรือ RTS โดย MCR จะอยู่ที่ตำแหน่ง Offset ที่ 4 จาก Base Address มีตำแหน่งและหน้าที่ของบิตต่าง ๆ เป็นดังรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

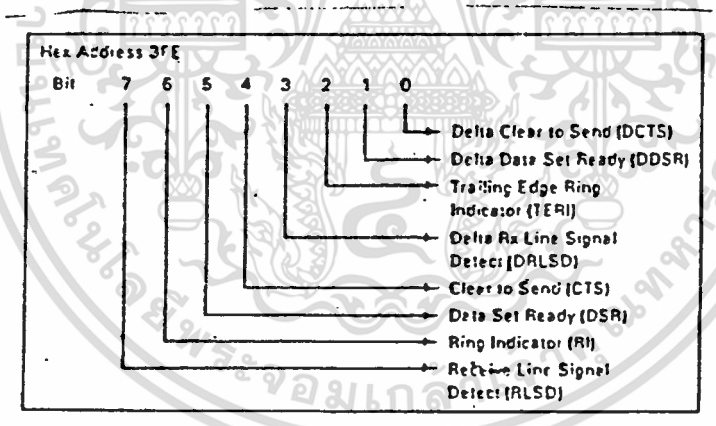
บิตที่ 0 ใน MRC จะทำหน้าที่เป็นสัญญาณ DTR ตามมาตรฐาน EIA-232 แต่จะมีสถานะตรงข้ามกับที่เอาต์พุต DTR คือ ถ้าเซ็ตเป็น 1 จะทำให้สัญญาณ DTR เป็น 0 และถ้าเคลียร์ให้เป็น 0 จะทำให้ DTR เป็น 1 ทั้งนี้เพราะจะนอกทีพที่ลอจิก 0

บิตที่ 1 ทำหน้าที่เป็น RTS ตามมาตรฐาน EIA-232 จะเป็น Negative Logic คือ ให้เอาต์พุตตรงข้ามกับค่าที่ป้อนให้ เช่นเดียวกับบิตที่ 0

บิตที่เหลือไม่ได้ใช้งานในไอบี เอ็มพีซี

Modern Status register (MSR)

เป็นริจิสเตอร์ที่แสดงสถานะต่าง ๆ ของสัญญาณ EIA-232 โดยจะอยู่ที่ Offset ที่ 6 จาก Base Address และจะมีตำแหน่งและความหมายของบิตต่าง ๆ เป็นดังรูปที่ 8



รูปที่ 8 ตำแหน่งและหน้าที่ของบิตต่าง ๆ ในMSR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receiver Buffer Register และ Transmitter Holding Register

ริจิส เตอร์ 2 ตัวนี้ เป็นริจิส เตอร์ที่ใช้ เก็บข้อมูลที่ได้รับจากสายส่ง และเป็นที่ เก็บข้อมูลที่ต้องการส่งก่อนที่จะทำการส่ง RBR และ THR จะอยู่ที่ตำแหน่ง Base Address และจะเป็น RBR หรือ THR ก็ต่อ เมื่อ DLAB มีค่า เป็น 0 เท่านั้น ข้อแตกต่างของ RBR และTHR ก็คือ เมื่ออ่านจะเป็นการอ่านจาก RBR และ เมื่อ เขียนจะ เป็นการ เขียนลง THR

ใช้งานพอร์ตผ่าน BIOS

ในการใช้งานพอร์ตอนุกรมนี้ เราสามารถทำได้หลายทาง วิธีแรกก็คือ เขียนโปรแกรม เพื่อควบคุมการทำงานที่ชิป 8250 โดยการ เขียนโปรแกรมค่าให้แก่ริจิส เตอร์ต่าง ๆ ภายในชิปโดยตรง วิธีนี้จะยุ่งยากมากแต่จะมีความ เร็วสูง เพราะ เป็นการ เข้าถึงระดับฮาร์ดแวร์โดยตรง อีกวิธีหนึ่งที่อาจจะช้าบ้างแต่ก็สะดวกกว่ามากก็คือการ เรียกใช้บริการของ BIOS ผ่านซอฟต์แวร์อิน เทอร์รัปต์หมาย เลข 14h

อิน เทอร์รัปต์หมาย เลขนี้มีฟังก์ชันให้ใช้งานที่จำ เป็นไว้อย่างครบครันการใช้งาน เพียงแต่โหลดค่าฟังก์ชันที่ต้องการไว้ในริจิส เตอร์ AH และหมาย เลขพอร์ตไว้ใน DX โดยหมาย เลขพอร์ตนี้จะมีค่าน้อยกว่าหมาย เลขพอร์ตที่แท้จริงอยู่ 1 เช่นหมาย เลขพอร์ตของ COM1 จะ เป็น 0 เป็นต้น แล้วทำการ เรียกอิน เทอร์รัปต์หมาย เลข 14h บางครั้งอาจจะต้องมีการโหลดข้อมูล เพิ่ม เติมไว้ใน AL

ฟอสซิลไดรเวอร์

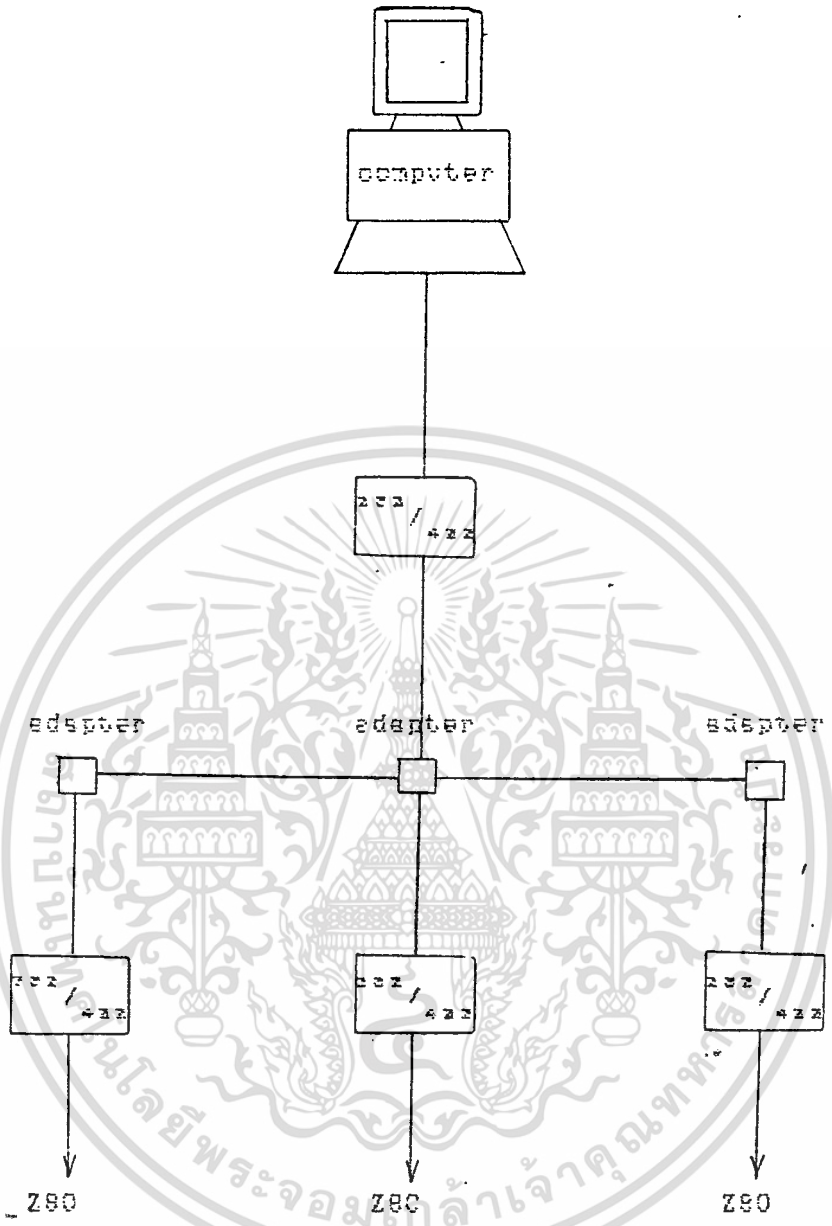
การทำงานของ BIOS ในการใช้งานพอร์ตอนุกรมนั้น จะ เป็นลักษณะของ Character Device คือ เป็นการรับส่งข้อมูลทีละ 1 ตัวอักขระยังต้องการการตรวจสอบสถานะอยู่ตลอด เวลาทำให้ เกิดปัญหาการ เกิด Overrun อยู่ เสมอ การแก้ปัญหานี้ เราสามารถทำได้โดยใช้ไดร์ เวอร์ที่ เรียกว่า Fossil Driver มาช่วยโดยฟอสซิลไดร์ เวอร์จะใช้การอิน เทอร์รัปต์และ DMA เข้าช่วยในการรับส่งข้อมูลกับพอร์ตอนุกรมโดยกันหน่วยความจะส่วนหนึ่งมา เป็นบัฟ เฟอร์ จึงทำให้ซี เรียลพอร์ตมีลักษณะ เป็น Block Device อย่างกลาย ๆ จึงสามารถช่วยแก้ปัญหาการ เกิด Overrun ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ความรู้พื้นฐานสำหรับการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงาน RS 232/422 CONVERTER

ก่อนจะทำการออกแบบทั้ง hardware และ software เราต้องเข้าใจหลักการก่อน

1. การสื่อสารแบบอนุกรม

ข้อมูลอยู่ในรูปของบิท ข้อมูลแต่ละบิทจะถูกส่งออกไปตามลำดับต่อเนื่องกันทีละบิท เช่น ถ้าข้อมูลเป็น 1010 เลข 0 ทางขวามือสุดซึ่งเป็น LSB(Least Significant Bit)ส่งออกไปก่อน ตามด้วยบิทที่ 2 คือเลข 1 บิทที่ 3 คือเลข 0 และบิทสุดท้ายคือเลข 1 ซึ่งเป็น MSB (Most Significant Bit) ตามลำดับโดยสายข้อมูลมีเส้นเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.มาตรฐาน RS 232-C

DCE:อุปกรณ์ที่มีฟังก์ชั่นการทำงานต่างๆที่ทำให้เกิดการเชื่อมต่อ,ทำให้การเชื่อมต่อยังดำรงต่อไปและยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่างๆที่จำเป็นต้องใช้ในการสื่อสารข้อมูลระหว่าง DTE [data terminal eqiupment]และ data circuit โดย DCE อาจเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์หรือไม่ก็ได้

DTE:

1. เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล [data source] หรือตัวรับข้อมูล [data sink] หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้
2. เป็นอุปกรณ์ที่ประกอบด้วย function unit ต่อไปนี้ conitrol logic , buffer store และอุปกรณ์inputหรือ output จำนวนหนึ่งตัวหรือมากกว่าก็ได้หรือรวมเครื่องคอมพิวเตอร์เข้าไปด้วยก็ได้ DTE อาจจะรวมส่วน error control , synchronizationและความสามารถในการบ่ง หรือ ระบุว่าต้องการเกี่ยวข้องกับอุปกรณ์ตัวใด [station identification capability] เข้าไปด้วยก็ได้

มาตรฐาน RS 232 มีสาระสำคัญดังนี้

- Baud period เป็นอัตราการสื่อสารข้อมูลหน่วยเป็น บิทต่อวินาที clock ของตัวรับและตัวส่งต้องมี baud period เหมือนกัน
- Marking state คือคาบเวลาระหว่างการไม่มีข้อมูลจะส่งระหว่างช่วงเวลานี้ตัวส่งจะ steady high
- Start bit คือ บิทแรกที่เป็นตัวชี้ว่ามีข้อมูลมา บิทนี้จะมีสถานะ low
- Charactor bit ประกอบด้วยข้อมูลขนาด 5,6,7,8 บิท โดยเริ่มส่งจาก LSB
- Parity bit จะมีหรือไม่มีก็ได้ ถ้ามีจะตามหลัง charector bit มา เพื่อเช็คความผิดพลาดของการส่ง ถ้าเราเลือกใช้พาริตี้คู่จะมีบิท 1 ตามหลัง ถ้าจำนวนบิทของ 1 เป็นจำนวนคู่ และถ้าเราใช้พาริตี้เดี่ยวจะมีบิท 1 ตามหลัง ถ้าจำนวนบิทของ 1 เป็นจำนวนคี่ ถ้ามีข้อผิดพลาด ที่ตัวรับจะเกิด error flag ที่รีจิสเตอร์พิเศษ
- Stop bit บิทสูง 1-2 บิท สุดท้าย เป็นการบอกตัวรับว่าหมดข้อมูลเดิมให้เตรียมรับข้อมูลชุดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS 232-C มีแบบมาตรฐานที่ใช้กับ IBM computer ดังนี้

Baud rate : 50,110,300,600,1200,2400,4800,9600,19200

Data bit : 5,6,7,8

Parity bit: คู่,คี่,ไม่มี

Stop bit : 1,1.5,2

การต่อ terminal

ในที่นี้ใช้แบบ DB-9

ขา	หน้าที่การทำงาน	code no.	direction
1	carrier detect	CD	
2	receive data	RD	input
3	transmit data	TD	output
4	data terminal ready	DTR	output
5	chassis ground	G	
6	data set ready	DSR	input
7	request to send	RTS	output
8	clear to send	CTS	input
9	ring indicator	RI	input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของสัญญาณไฟฟ้าสำหรับ RS 232-C

1. สัญญาณที่ขาทุกขาที่ connector ของ RS 232-C จะเป็นสภาวะใดสภาวะหนึ่งในแต่ละคู่ของคู่ต่อไปนี้

MARK/SPACE

ON/OFF

logic 0/logic 1

RS 232-C ใช้ logic ลบแทนระดับแรงดันต่างๆ logic ลบ คือวิธีการเปรียบเทียบระดับแรงดันแบบหนึ่ง ถ้าระดับแรงดันหนึ่งมีค่าเป็นลบมากกว่าอีกระดับแรงดันหนึ่ง ระดับแรงดันที่มีค่าเป็นลบมากกว่า จะเป็น logic "สูง" โดยแรงดันของระดับสัญญาณต่างๆจะถูกวัดเทียบกับ circuit signal ground นอกจากนี้ ช่วงของแรงดันระดับระหว่าง -3 ถึง +3 จะเป็นช่วงของการเปลี่ยนแปลง logic ดังนั้นจึงไม่มีการระบุสถานะของสัญญาณในช่วงนี้

2. ในการแทน logic หนึ่ง หรือสถานะ MARK ตัวขับสัญญาน (Driver) ต้องจ่ายแรงดันระหว่าง -5 ถึง -15 volt ส่วนในการแทน logic 0 หรือ SPACE ตัวขับสัญญานต้องจ่ายแรงดันระหว่าง +5 ถึง +15 volt

จากข้อ 1 และข้อ 2 แสดงว่า RS 232-C ยอมให้มี noise margin ได้ไม่เกิน 2 volt

3. ตัวเก็บประจุ C ที่ต่อขนานกับอุปกรณ์รับข้อมูลปลายทาง จะต้องมีค่าไม่เกิน 2500 pF โดยค่านี้ ไม่รวมค่าความจุไฟฟ้าของสาย cable เข้าไปด้วย

หมายเหตุ ตามข้อกำหนดนี้ ระยะทางที่สามารถใช้การสื่อสารข้อมูลได้ ต้องไม่เกิน 50 ฟุต ซึ่งถูกกำหนดไว้ในมาตรฐาน RS 232-C

4. แรงดันขณะเปิดวงจร หรือขณะไม่มี load (V0) จะต้องไม่เกิน 25 volt ซึ่งก็คือแรงดันใดๆใน circuit ของการ interface แบบ RS 232-C ต้องไม่เกิน 25 volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

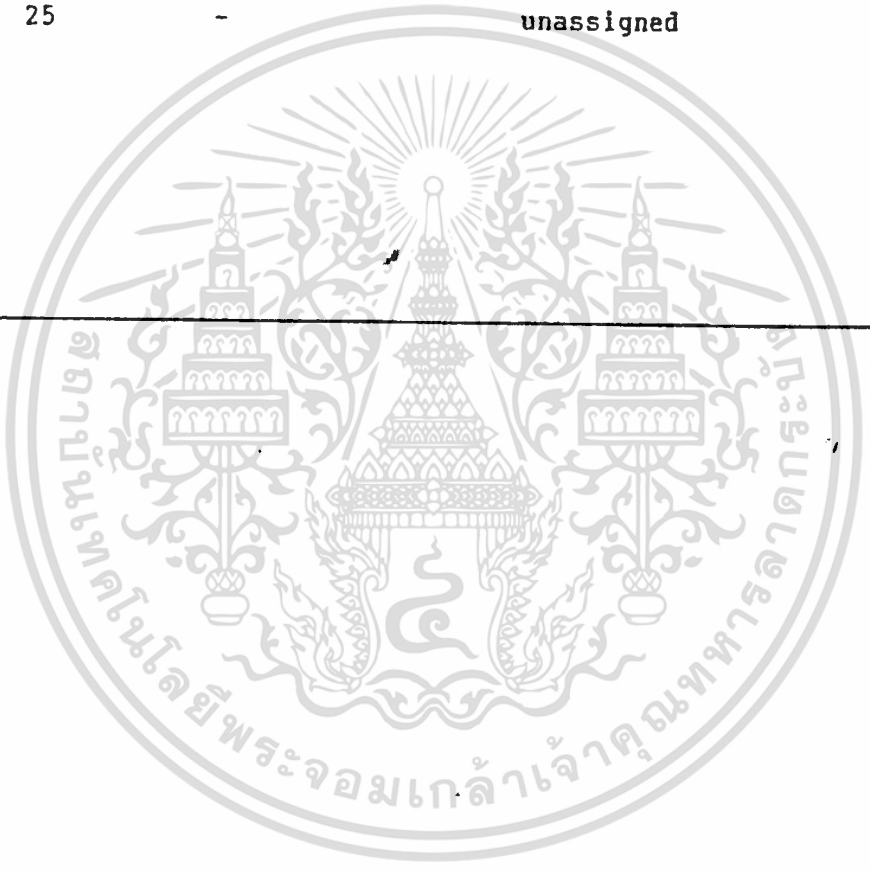
5. วงจรรับสัญญานที่ใช้กับ RS 232-C ต้องสามารถทนต่อการลัดวงจรที่เกิดขึ้นได้ โดยไม่ทำให้เกิดความเสียหายต่อตัวมันเอง หรืออุปกรณ์ที่เกี่ยวข้องด้วย เช่น terminal, modem, port I/O และอุปกรณ์ต่างๆที่ต่อเข้ากับ cable ที่ใช้ในการ interface แบบ RS 232-C

รายละเอียดของขาต่างๆ พร้อมด้วยสัญญานกำกับของ connector

ขา	circuit	ความหมายของ circuit
1	AA	protective ground
2	BA	transmitted data
3	BB	received data
4	CA	request to send
5	CB	clear to send
6	CC	data set ready
7	AB	signal ground
8	CF	received line signal detector
9/10	-	(reserved for data set testing)
11	-	unassigned
12	SCF	secondary received line signal detector
13	SCB	secondary clear to send
14	SVA	secondary transmitted data
15	DB	transmit signal element timing
16	SBB	secondary received data
17	DD	receive signal element timing
18	-	unassigned

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19	SCA	secondary request to send
20	CD	data terminal ready
21	CG	signal quality detector
22	CE	ring indicator
23	CH/CI	data signal rate select
24	DA	transmit signal element timing
25	-	unassigned



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คุณสมบัติของ 8251

8251 เป็น IC ขนาด 28 ขา และเราสามารถที่จะแบ่งขาของ 8251 ออกเป็นกลุ่มๆได้ดังนี้

1.1 D0-D7 ใช้ในการติดต่อกับ DATA BUS ของ CPU โดยตรงซึ่งจะทำหน้าที่ในการรับส่งข้อมูลและคำสั่งต่างๆระหว่าง 8251 กับ CPU

1.2 RESET 8251 จะถูกRESET เมื่อขานี้ได้รับลอจิก 1 ซึ่งเราอาจจะต่อมาจากขาRESET ของ Z80 โดยผ่าน INVERTERก่อนก็ได้

1.3 CLK (CLOCK) ใช้ในการควบคุมช่วงเวลาการทำงานภายในของ 8251 สำหรับการใช้งานนั้นจะต่อเข้าโดยตรงกับระบบ อย่างไรก็ตามสัญญาณที่ขา CLK นี้ไม่เกี่ยวข้องกับอัตราการรับส่งข้อมูลหรือBAUD RATE แต่อย่างใด

1.4 RD เมื่อขานี้ได้รับลอจิก 0 8251 จะทำการส่งข้อมูลแบบขนานออกมาที่ DATA BUS เพื่อส่งให้กับ CPU

1.5 WR เมื่อขานี้ได้รับลอจิก 0 8251 จะทำการรับข้อมูลแบบขนานจาก DATA BUS ของระบบ

1.6 C/D (CONTROL/DATA) ขา C/D นี้ จะใช้ในการทำให้ 8251 ทราบว่า CPU ต้องการที่จะติดต่อกับ CONTROL REGISTER หรือ DATA REGISTER โดยที่ถ้าขานี้ได้รับลอจิก 1 ก็แสดงว่า CPU ต้องการที่จะติดต่อกับ CONTROL REGISTER แต่ถ้าได้รับลอจิก 0 ก็แสดงว่า CPU ต้องการที่จะติดต่อกับ DATA REGISTER

1.7 CS (CHIP c) ในกรณีที่ขานี้ได้รับลอจิก 0 ก็จะเป็นการ ENABLE 8251 โดยทั่วไปแล้ว สัญญาณที่ขานี้จะได้มาจากการถอดรหัสแอดเดรส ดังที่ใช้กับ CHIP SUPPORT อื่นๆ

กลุ่มที่ใช้ในการติดต่อกับ MODEM

2.1 DSR (DATA SET READY) ขานี้เป็นขาที่ใช้ในการรับสัญญาณจากอุปกรณ์ภายนอกซึ่ง CPU สามารถที่จะตรวจสอบสัญญาณที่ขานี้ได้ โดยการอ่านค่าใน REGISTER สถานะ และระดับของสัญญาณที่ขานี้ จะใช้ในการแสดงว่า อุปกรณ์ภายนอกพร้อมที่จะทำการติดต่อด้วยหรือยัง

2.2 DTR (DATA TERMINAL READY) ขานี้เป็น OUTPUT ที่ใช้ในการบอกให้อุปกรณ์ภายนอกทราบว่า CPU พร้อมที่จะทำการติดต่อด้วย

2.3 CTS (CLEAR TO SEND) ขานี้เป็นขา INPUT ที่ใช้ในการทำให้ 8251 เริ่มทำการส่งข้อมูลได้ สิ่งที่ต้องระวังในการใช้งานขานี้ก็คือ เมื่อไม่ได้ใช้งานขานี้ จะต้องถูกต่อเข้ากับลอจิก 0 ถ้าไม่เช่นนั้น 8251 จะทำการส่งข้อมูลไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 RTS (READY TO SEND) ขานี้เป็น OUTPUT ที่ CPU จะเป็นผู้ควบคุมสัญญานที่ขานี้เอง

กลุ่มที่ใช้ในการส่งข้อมูล

3.1 TxD (TRANSMIT DATA OUTPUT) เป็นขาที่ใช้ในการส่งข้อมูลไปตามสายส่ง

3.2 TxC (TRANSMIT BAUD RATE CLOCK) ขานี้เป็นขาที่ใช้ในการส่งสัญญาน CLOCK ที่ใช้ในการส่งข้อมูล ซึ่งก็คือความถี่ที่ใช้ในการกำหนด BAUD RATE นั่นเอง โดยปกติแล้ว จะต้องช้ากว่าสัญญาน CLOCK ของระบบไม่น้อยกว่า 30 เท่า

3.3 TxRDY ขานี้จะใช้ในการทำให้ CPU ทราบว่า 8251 พร้อมที่จะรับข้อมูลจาก CPU เพื่อที่จะทำการส่งต่อไปแล้วหรือยัง และขานี้อาจจะนำไปใช้ในการขออินเทอร์รัพก็ได้

3.4 TxEMPTY ขานี้จะใช้ในการแสดงว่า ข้อมูลที่ CPU ส่งให้กับ 8251 นั้น ได้ถูกส่งออกไปให้อุปกรณ์อื่นๆหมดแล้ว โดยที่ 8251 จะทำให้ขานี้เป็นหนึ่ง และเมื่อ CPU ทำการส่งข้อมูลชุดต่อไปให้กับ 8251 ขานี้ก็จะเป็น 0 จนกว่า 8251 จะทำการส่งข้อมูลนี้ออกไปหมด 8251 ก็จะทำให้ขานี้กลับเป็น 0 อีกครั้ง

กลุ่มที่ใช้ในการรับข้อมูล

4.1 RxD ใช้ในการรับข้อมูลแบบอนุกรมจากสายส่ง

4.2 Rxc เป็นขาที่ใช้ในการรับสัญญาน CLOCK ที่ใช้ในการรับข้อมูล โดยปกติแล้วจะทำการต่อเข้ากับ TxC โดยตรง

4.3 RxRDY จะใช้ในการแสดงว่า 8251 พร้อมที่จะส่งข้อมูลให้กับ CPU และขานี้อาจจะใช้ในการขออินเทอร์รัพท์ได้เช่นเดียวกับขา TxRDY

4.4 SYNDET ขานี้จะใช้ในการรับข้อมูลแบบ SYNCHRONOUS เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 8250 UART

ในการรับส่งข้อมูลทางไมโครคอมพิวเตอร์นั้น จะใช้ 8250 UNIVERSAL ASYNCHRONUS RECEIVER AND TRANSMITTER (UART) เป็นตัวควบคุม ซึ่งมีลักษณะภายในดังรูป 4.1 ซึ่งจะมี 10 INTERNAL REGISTER และสามารถเข้าถึงได้โดยเข้าทาง PORT ต่างๆ ซึ่งถ้าเป็นเครื่อง AT จะใช้ PORT COM1 ซึ่งมีเบอร์ PORT ตั้งแต่ 3F8H-3FEH ดังตารางที่ 4.1

ส่วนในตาราง 4.2 จะแสดง INTERNAL REGISTER MAPPING ของ 8250 UART ในเครื่อง IBM PCjr ในการสร้าง BAUD RATE นั้นจะสามารถดูได้จากตาราง 4.3 ซึ่งจะแสดงการกำหนด BAUD RATE ต่างๆและค่า ERROR

THE CONTROLLER'S REGISTERS

- THE TRANSMITTER HOLDING REGISTER: จะเก็บค่าที่จะส่งออกไป ในการส่ง จะส่ง BIT ต่ำออกไปก่อน

- THE RECEIVER DATA REGISTER: เรียกอีกอย่างได้ว่า RECEIVER BUFFER REGISTER เก็บค่าที่รับได้

- THE BAUD RATE DIVISOR REGISTER: อาจเรียกได้ว่าเป็น DIVISOR LATCH REGISTERS ใช้ในการโปรแกรม BAUD RATE GENERATOR

- THE INTERRUPT ENABLE REGISTER: ใช้ในการควบคุมการ INTERRUPT ในการควบคุม ดูได้จากรูป 4.2

- THE INTERRUPT IDENTIFICATION REGISTER: ดูรูป 4.3 และตาราง 4.4

- THE LINE CONTROL REGISTER: เก็บค่าที่ใช้ในการ INITIAL 8250 ซึ่งสามารถกำหนดได้จากรูป 4.4

- THE MODEM CONTROL REGISTER: เก็บค่าที่ใช้ในการควบคุมการรับและการส่ง ซึ่งสามารถกำหนดได้จากรูป 4.5

- THE LINE STATUS REGISTER: เป็น REGISTER ที่สามารถเรียกดูสถานะของข้อมูลที่รับส่ง ดูรูป 4.6

- THE MODEM STATUS REGISTER: สามารถเรียกดูสถานะของสายควบคุม MODEM ดูรูป 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

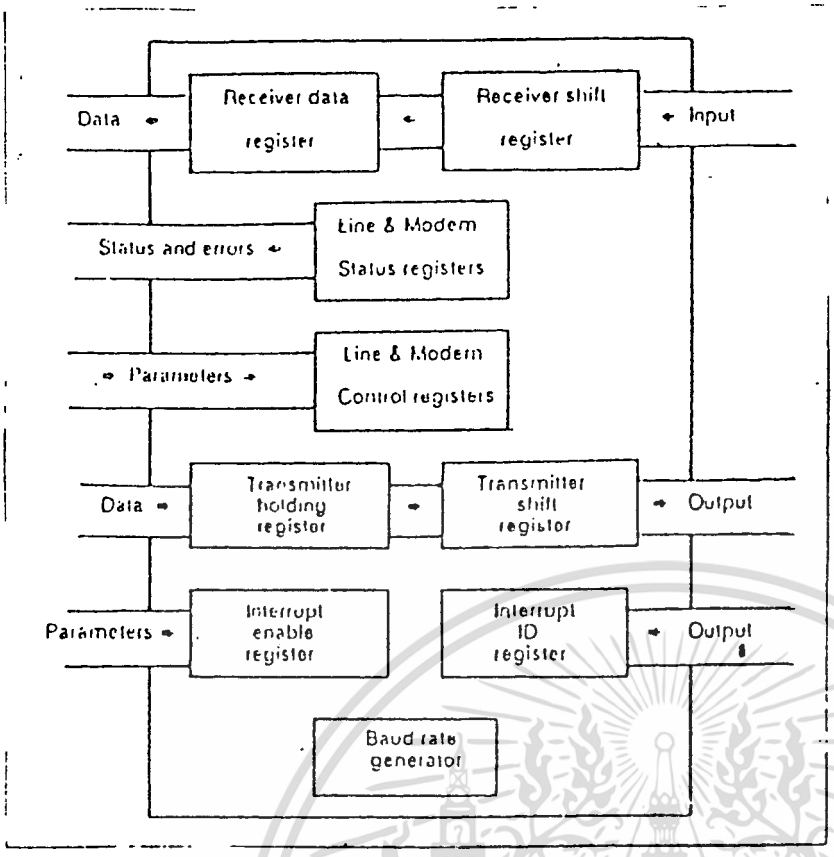


FIGURE 4.1 REGISTERS AND FUNCTIONS OF THE IBM SERIAL COMMUNICATION

TABLE 4.1 INTERNAL REGISTER MAPPING OF SERIAL COMMUNICATION

REGISTER NAME	REGISTER CODE NAME	PORT ADDRESS		FUNCTION
		COM1	COM2	
Transmitter holding register	THR	3F8H	2F8H	Output
Receiver data register	RDR	3F8H	2F8H	Input
Baud rate divisor (LSB)	BRDL	3F8H	2F8H	Output
Baud rate divisor (MSB)	BRDH	3F9H	2F9H	Output
Interrupt enable register	IER	3F9H	2F9H	Output
Interrupt ID register	IID	3FAH	2FAH	Input
Line control register	LCR	3FBH	2FBH	Output
Modem control register	MDC	3FCH	2FCH	Output
Line status register	LST	3FDH	2FDH	Input
Modem status register	MSR	3FEH	2FEH	Input

TABLE 4.2 INTERNAL REGISTER MAPPING OF THE 8250 UART

REGISTER NAME	REGISTER CODE NAME	INTERNAL MODEM	SERIAL PORT	FUNCTION
Transmitter holding register	THR	3F8H	2F8H	Output
Receiver data register	RDR	3F8H	2F8H	Input
Baud rate divisor (LSB)	BRDL	3F8H	2F8H	Output
Baud rate divisor (MSB)	BRDH	3F9H	2F9H	Output
Interrupt enable register	IER	3F9H	2F9H	Output
Interrupt ID register	IID	3FAH	2FAH	Input
Line control register	LCR	3FBH	2FBH	Output
Modem control register	MDC	3FCH	2FCH	Output
Line status register	LST	3FDH	2FDH	Input
Modem status register	MSR	3FEH	2FEH	Input

Note: If the PC's internal modem is installed, the serial port is logically COM2 and the modem is COM1. With no modem, the serial port is COM1. In the PC's the serial port's base address is always 2F8H.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

BAUD RATE	PC-AT and PS/2 CLOCK SPEED, 1.6432 MHz			PCjr CLOCK SPEED, 1.7095 MHz		
	DECIMAL	HEX	% ERROR	DECIMAL	HEX	% ERROR
50	2304	900H	0	2337	8BDH	.008
75	1536	600H	0	1491	5D3H	.017
110	1047	417H	026	1017	3F9H	023
134.5	857	359H	.058	832	340H	054
150	768	300H	0	746	2EAH	.050
300	384	180H	0	373	175H	050
600	192	C0H	0	186	BAH	.218
1200	96	60H	0	93	5DH	.218
1800	64	40H	0	62	3EH	.218
2000	58	3AH	690	56	38H	.140
2400	48	30H	0	47	2FH	.855
3600	32	20H	0	31	1FH	.218
4800	24	18H	0	23	17H	1.291
7200	16	10H	0	- NOT RECOMMENDED -		
9600	12	CH	0	- NOT RECOMMENDED -		
19200	6	6H	0	- NOT RECOMMENDED -		

TABLE 4.3 DIVISOR VALUES FOR PROGRAMMING THE BAUD RATE GENERATOR

FIGURE 4.2

THE INTERUPT ENABLE

REGISTOR

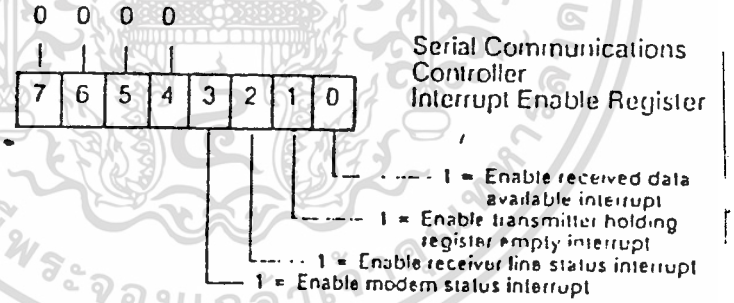
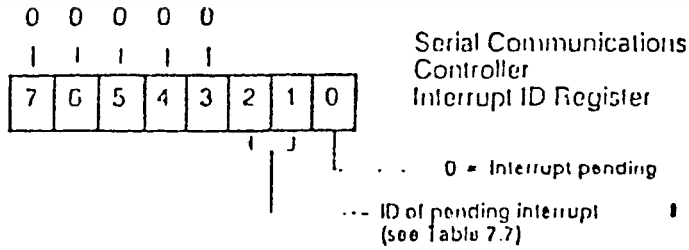


FIGURE 4.3

THE INTERUPT IDENTIFICATION

REGISTER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

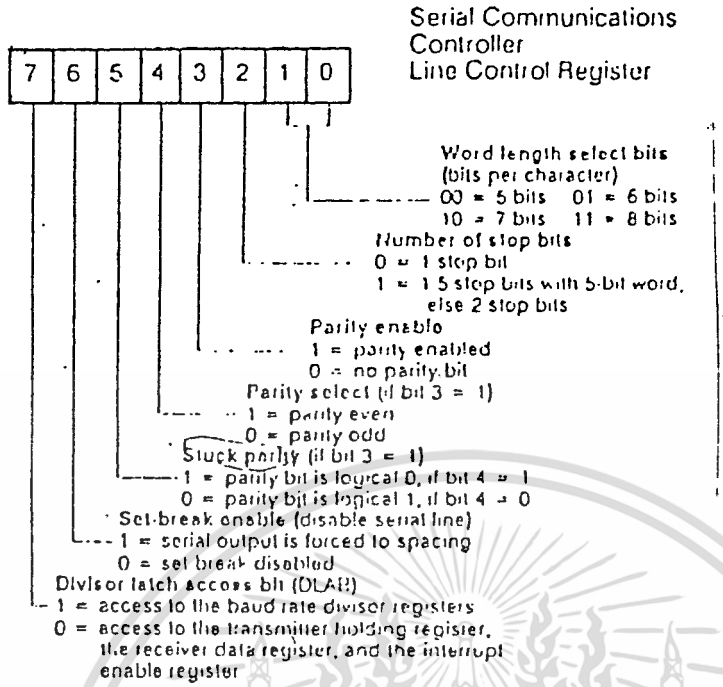


FIGURE 4.4 THE LINE CONTROL REGISTER

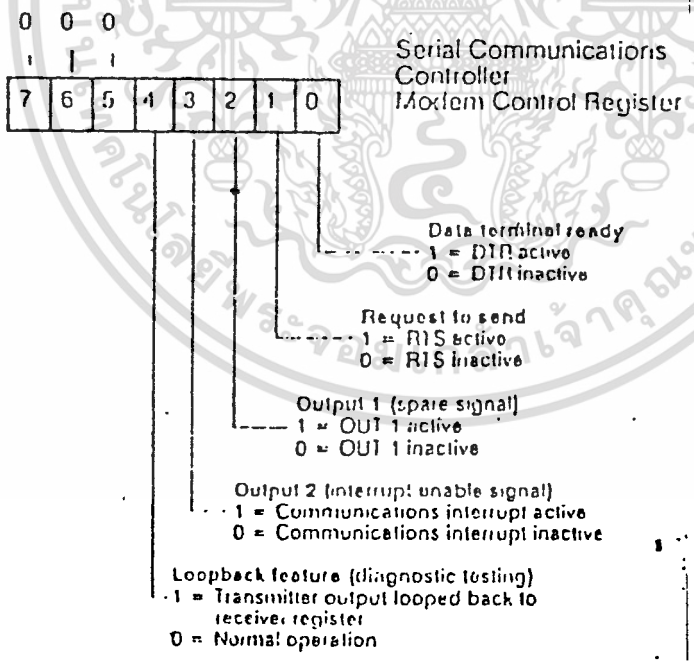


FIGURE 4.5 THE MODEM CONTROL REGISTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

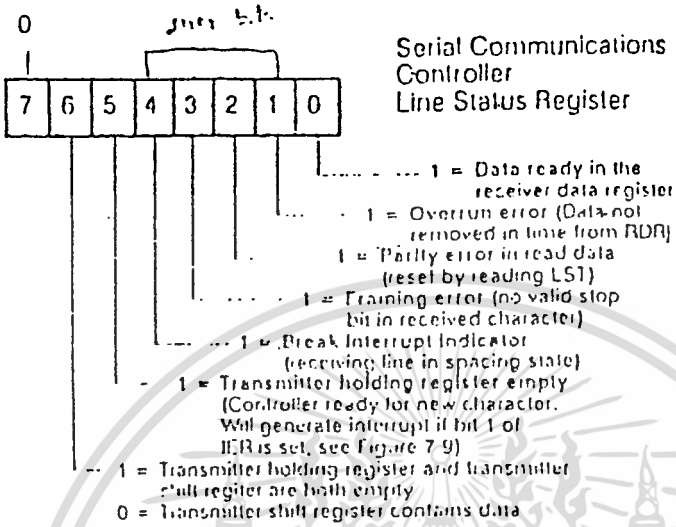


FIGURE 4.6 THE LINE STATUS REGISTER

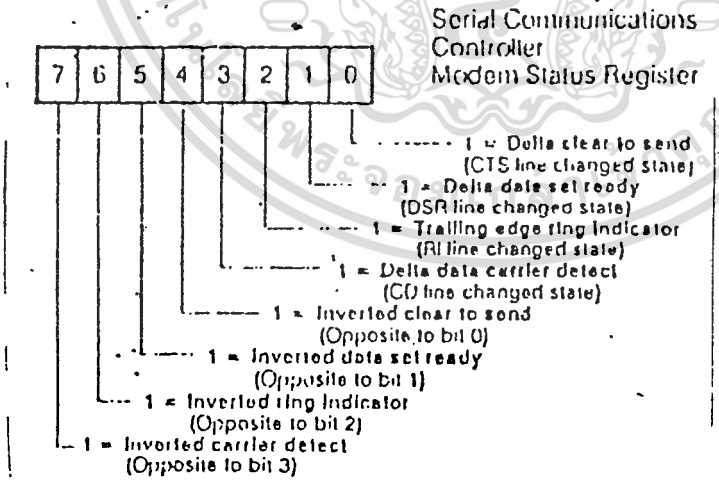


FIGURE 4.7 THE MODEM STATUS REGISTER

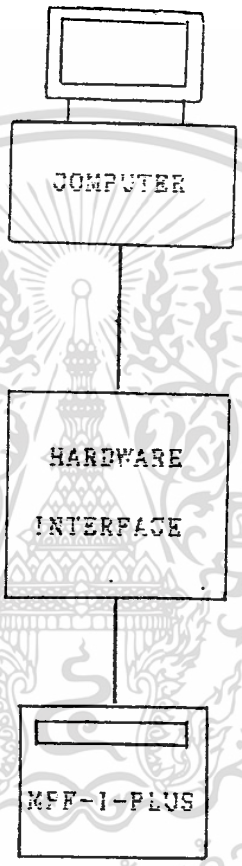
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการออกแบบด้าน HARDWARE

1. IC 8251 ทำหน้าที่ในการแปลงข้อมูลแบบอนุกรมเป็นแบบขนานและแปลงข้อมูลแบบขนานเป็นแบบอนุกรม

ขา C/$\overline{D}$ ต่อกับ A0 จาก Z80 เพื่อให้ A0 เป็นตัวเลือกว่า ข้อมูลที่ขา D0-D7 จะเป็นDATA เมื่อ A0 เป็นลอจิกศูนย์และจะเป็น CONTROL WORD เมื่อ A0 เป็นลอจิกหนึ่ง

ขา TxC,RxC ต่อกับ IC 8253 เพื่อให้ 8253 สร้าง BAUD RATE

ขา D0-D7 ต่อกับ IC 74245 เพื่อใช้เป็น BUFFER DATA

ขา CLOCK ต่อกับ CLOCK ของ Z80

ขา RESET ต่อกับ GROUND ผ่าน R 10k เพื่อใช้ INTERNAL RESET

ขา $\overline{RD}$ ต่อกับสัญญาน $\overline{RD}.\overline{IORQ}$ จาก Z80 เป็นสัญญานการอ่าน

ขา $\overline{WR}$ ต่อกับสัญญาน $\overline{WR}.\overline{IORQ}$ จาก Z80 เป็นสัญญานการเขียน

ขา $\overline{CS}$ ต่อกับ IC 138 เพื่อใช้เป็นตัวถอดรหัส ADDRESS โดยใช้ ADDRESS 10 และ 11

ขา TxRDY ต่อกับ LED เพื่อบอกให้ทราบว่า 8251 ได้ส่งข้อมูลออกไปแล้ว

ขา TxEMP ต่อกับ LED เพื่อบอกให้ทราบว่า 8251 ไม่มีข้อมูลจะส่ง

ขา RxRDY ต่อกับ LED เพื่อบอกให้ทราบว่า 8251 รับข้อมูลมาแล้ว

ขา TxD,$\overline{DTR}$,$\overline{RTS}$ ต่อกับ IC 1488 เพื่อใช้ทำการส่งสัญญานได้ตามมาตรฐาน RS 232 โดยลอจิกหนึ่ง(+5V) แปลงเป็น -12V และลอจิกศูนย์(0V) แปลงเป็น +12V

ขา Rxd,$\overline{DSR}$,$\overline{CTS}$ ต่อกับ IC 1489 เพื่อใช้รับสัญญาน TTL จากการแปลง

2. IC 8253 ทำหน้าที่ในการสร้าง BAUD RATE

ขา $\overline{CS}$ ต่อกับ IC 138 เพื่อใช้เป็นตัวถอดรหัส ADDRESS โดยใช้ ADDRESS 30 และ 33

ขา A0 และ A1 ต่อกับขาเดียวกันจาก Z80 เพื่อใช้ในการเลือก CHANEL ที่จะใช้งาน และเป็นสัญญาน CONTROL

ขา $\overline{WR}$ ต่อกับสัญญาน $\overline{WR}.\overline{IORQ}$ จาก Z80 เป็นสัญญานการเขียน

ขา $\overline{RD}$ ต่อกับสัญญาน $\overline{RD}.\overline{IORQ}$ จาก Z80 เป็นสัญญานการอ่าน

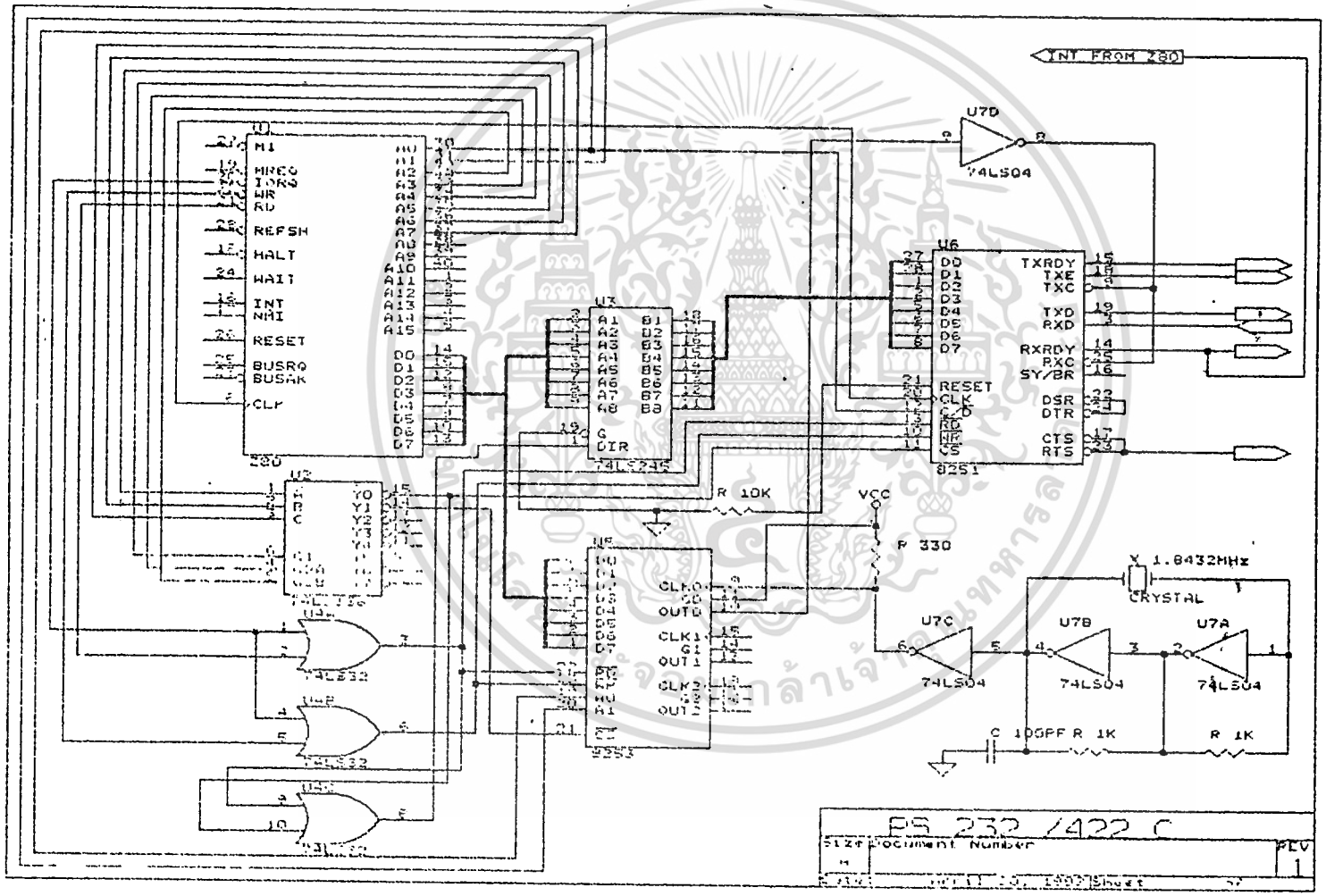
ขา D0-D7 ต่อกับของ Z80

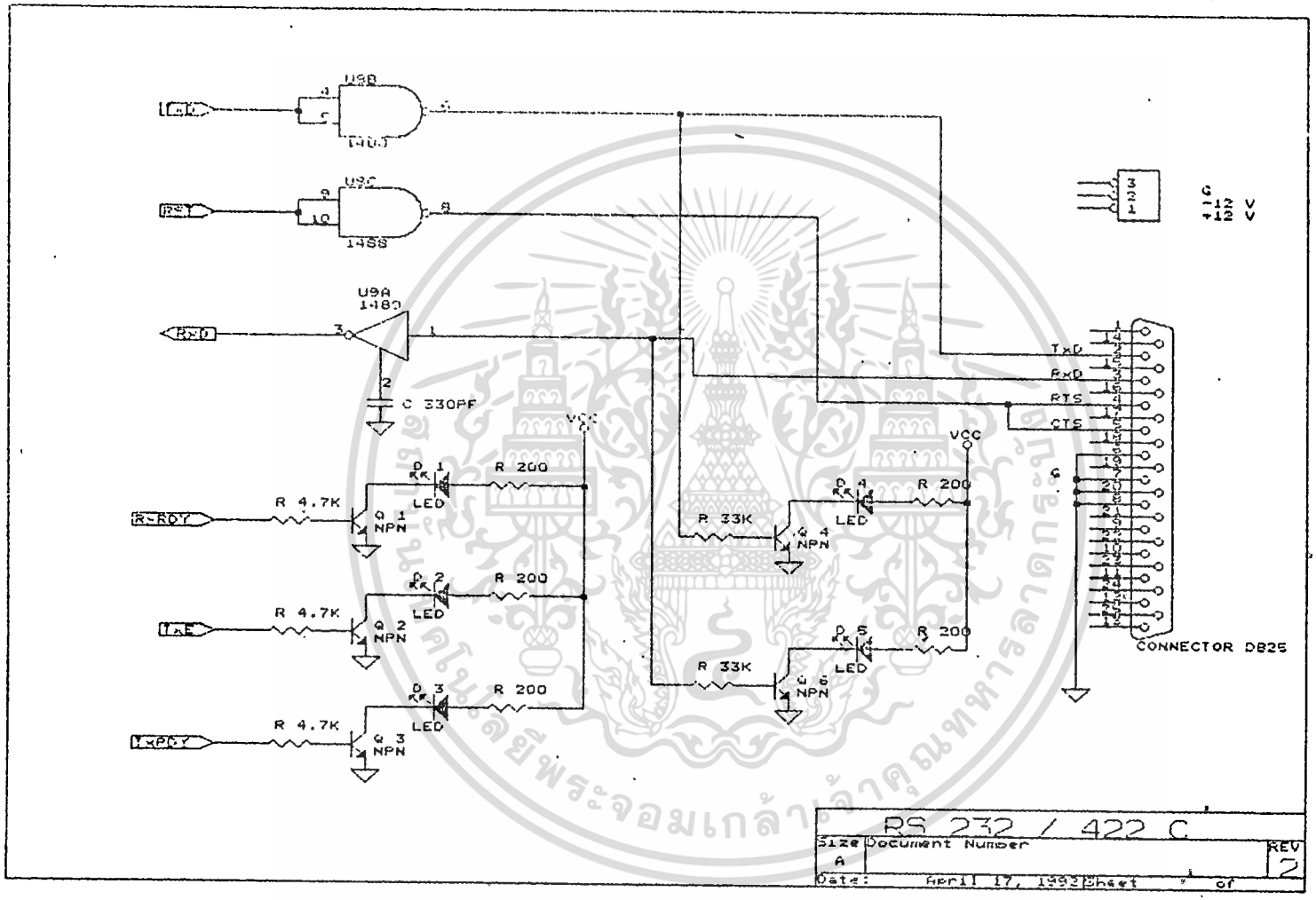
ขา CLOCK 0 ต่อกับ CLOCK จาก Z80

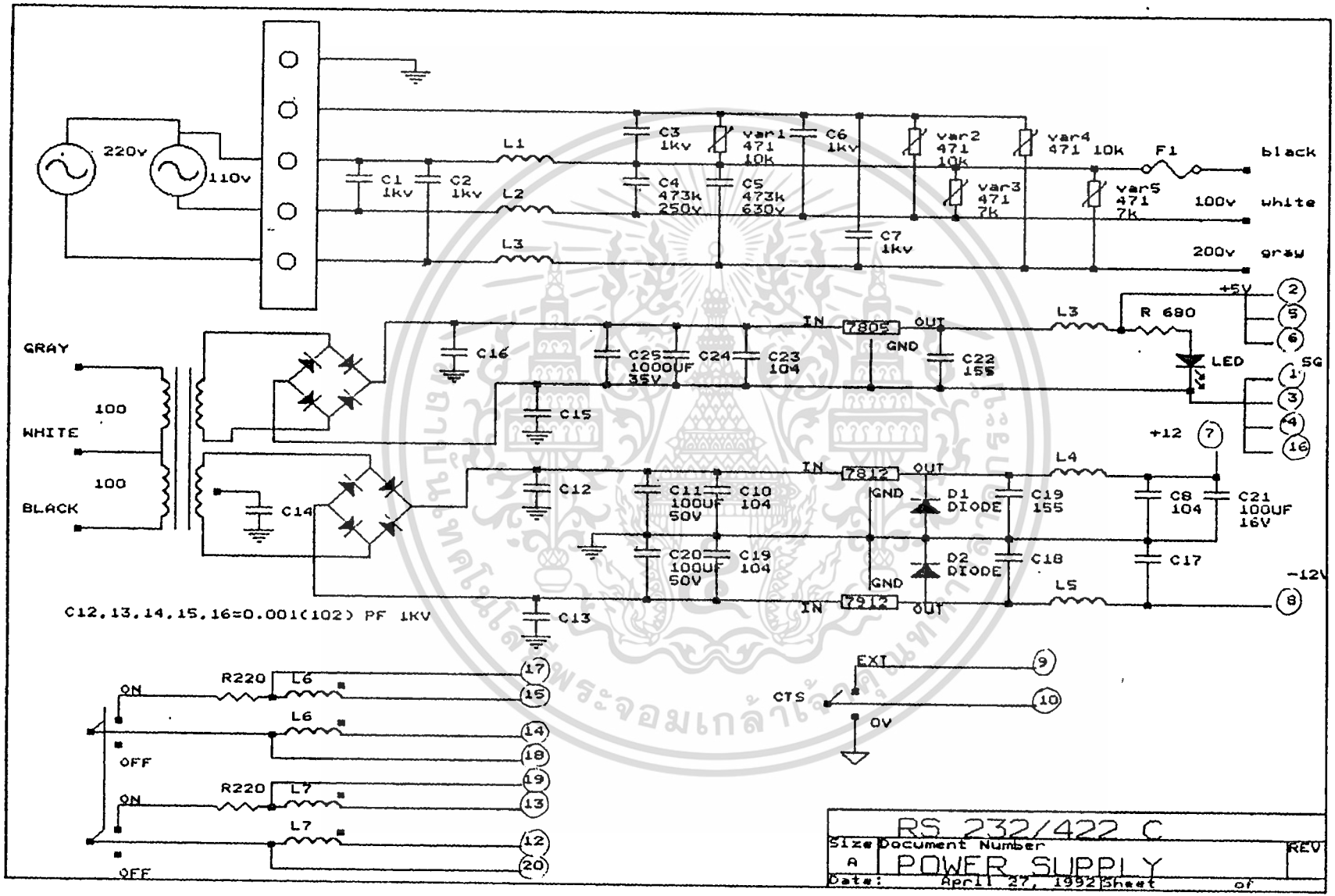
ขา GATE 0 ต่อกับ +5V เพื่อให้ CHANEL 0 ทำงานได้

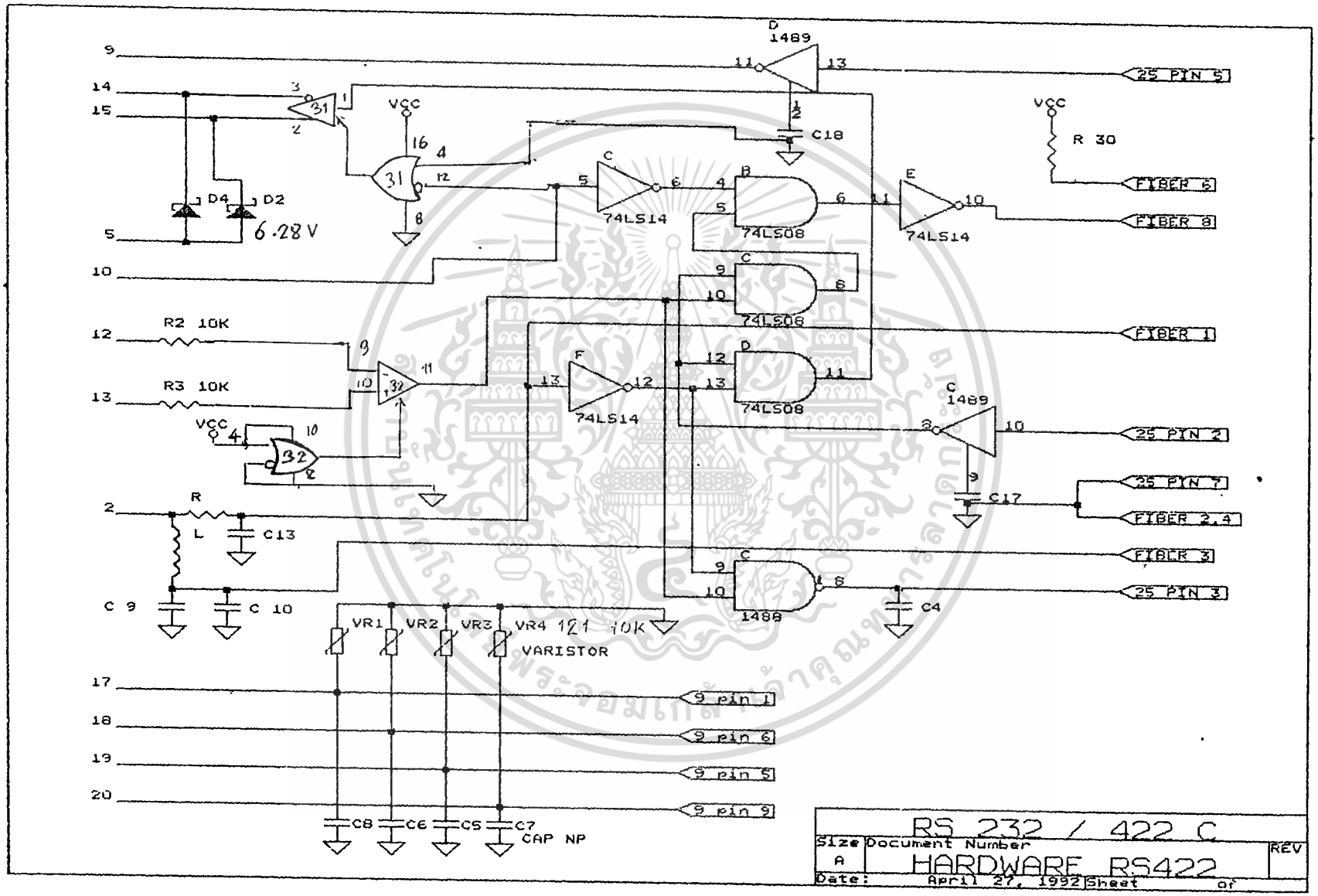
ขา OUT 0 ต่อกับ TxC,RxC เพื่อส่ง BAUD RATE ให้ 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


INT FROM Z80
U7D
74LS04
U6
8251
TXRDY
TXE
TXC
TXD
RXD
RXRDY
RXC
SY/BR
DSR
DTR
CTS
RTS
RESET
CLK
D0
D1
D2
D3
D4
D5
D6
D7
M1
MREQ
IORQ
WR
RD
REFSH
HALT
WAIT
INT
NMI
RESET
BUSRQ
BUSAK
CLK
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
Z80
U3
74LS245
G
DIR
R 10K
VCC
R 330
U5
8253
CLK0
OUT0
CLK1
OUT1
CLK2
OUT2
GATE0
GATE1
GATE2
U2
74LS138
74LS32
Y 1.8432MHz
CRYSTAL
U7C
U7B
U7A
74LS04
C 100PF
R 1K
R 1K
RS 232 /422 C
Size
Document Number
REV
1
Sheet


U9B
U9C
U9A
1489
C 330PF
R 200
R 4.7K
R 33K
LED
NPN
VCC
TxD
RxD
RTS
CTS
CONNECTOR DB25
+12 V
-12 V
RS 232 / 422 C
Size
A
Document Number
REV
2
Date: April 17, 1992
Sheet
of


220v
110v
L1
L2
L3
C1 1kv
C2 1kv
C3 1kv
C4 473k 250v
C5 473k 630v
C6 1kv
C7 1kv
var1 471 10k
var2 471 10k
var3 471 7k
var4 471 10k
var5 471 7k
F1
black
100v white
200v gray
GRAY
100
WHITE
100
BLACK
C16
C15
C14
C12
C13
C25 1000UF 35V
C24
C23 104
C22 155
IN 7805 OUT
GND
L3
R 680
LED
+5V
SG
+12
IN 7812 OUT
L4
C11 100UF 50V
C10 104
C20 100UF 50V
C19 104
D1 DIODE
D2 DIODE
C19 155
C18
C8 104
C21 100UF 16V
C17
-12V
L5
IN 7912 OUT
C12,13,14,15,16=0.001(102) PF 1KV
ON
OFF
R220
L6
L7
EXI
CTS
0V
RS 232/422 C
Size A
Document Number
POWER SUPPLY
REV
Date: April 27, 1992
Sheet of


D 1489
25 PIN 5
VCC
R 30
FIBER 6
FIBER 8
C18
74LS14
74LS08
74LS14
R2 10K
R3 10K
FIBER 1
74LS14
74LS08
C 1489
25 PIN 2
25 PIN 7
FIBER 2,4
C17
R
L
C13
FIBER 3
25 PIN 3
C 9
C 10
1488
C4
VR1
VR2
VR3
VR4 121 10K
VARISTOR
9 pin 1
9 pin 6
9 pin 5
9 pin 9
C8
C6
C5
C7
CAP NP
D4
D2
6.28 V
RS 232 / 422 C
Size A
Document Number
HARDWARE RS422
REV
Date: April 27, 1992
Sheet of

ขั้นตอนการออกแบบโปรแกรมบนเครื่องคอมพิวเตอร์

1.การ initial 8250 ให้เป็นแบบ

- 8 bitdata , 2 stopbit , evenparity โดยการให้ค่า $1F เข้าไปที่ port[$3FB]

- baudrate = 2400 โดยการให้ค่า $30 และ $00 เข้าไปที่ port[$3F8] และ port[$3F9] ตามลำดับ

2.ทำให้ 8250 ไม่สามารถรับการ interrupt ได้โดยการนำค่า $00 ไปเข้าที่port[$3F9]

3.input command โดยการระบุ unit number ของ microprocessor (00 หรือ 01) ตามด้วยคำสั่ง MW ซึ่งเป็นคำสั่งในการอ่านข้อมูลใน memory ของ microprocessor จากนั้นใส่จำนวน address ลงไป

4.การส่งข้อมูล (transmit)

มีขั้นตอนดังนี้

4.1 ตรวจดูว่า transmitter holding register ว่างหรือไม่ โดยดูจาก port[$3FD] ว่า bit ที่ 5 เท่ากับ 1 หรือไม่ ถ้าเท่ากับ 1 แสดงว่าว่าง สามารถส่งข้อมูลออกไปได้ ถ้าไม่ว่างให้รอจนกว่าจะว่างแล้วจึงส่ง ทาง port[$3F8]

4.2 กลับไปข้อ 3 เพื่อส่งคำสั่งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

intel®

8251A
PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
- Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate—DC to 64K Baud
- Asynchronous Baud Rate—DC to 19.2K Baud
- Full-Duplex, Double-Buffered Transmitter and Receiver
- Error Detection—Parity, Overrun and Framing
- Compatible with an Extended Range of Intel Microprocessors
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible
- Available in EXPRESS
  —Standard Temperature Range
  —Extended Temperature Range

The Intel® 8251A is the enhanced version of the industry standard, Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's microprocessor families such as MCS-48, 80, 85, and iAPX-86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is fabricated using N-channel silicon gate technology.

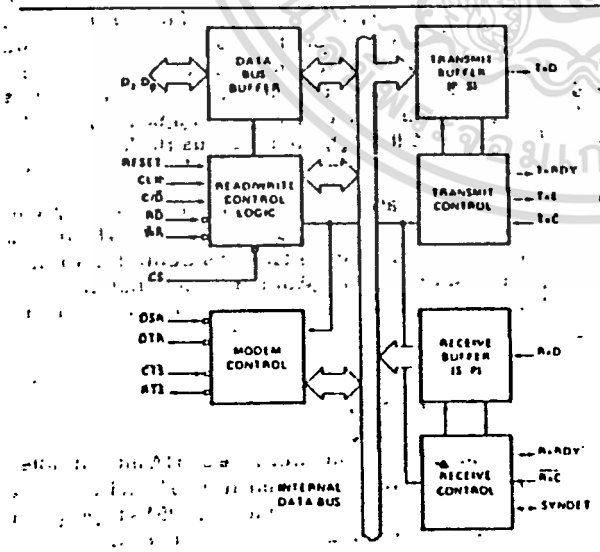


Figure 1. Block Diagram

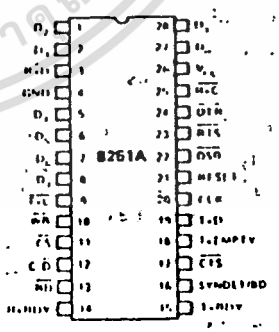


Figure 2. Pin Configuration

Intel Corporation Assumes No Responsibility for the Use of Any Circuitry Other Than Circuitry Embodied in an Intel Product. No Other Circuit Patent Licenses are Implied.
INTEL CORPORATION 1984

September 1984
205722-002

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of Intel microcomputers such as 8048, 8080, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync."

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

A command reset operation also puts the device into the "Idle" state.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

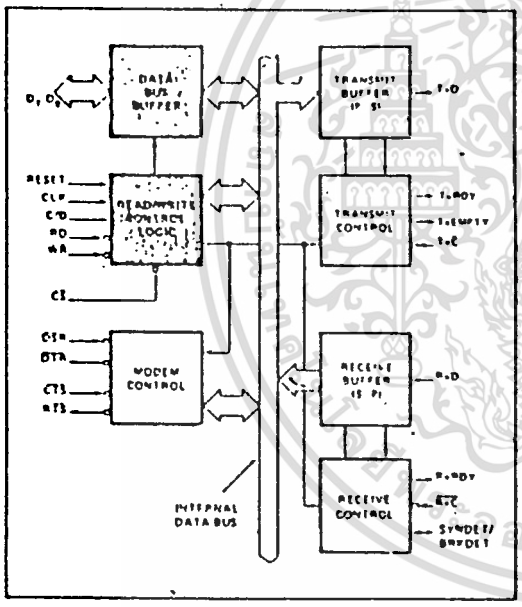


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3 STATE
X	X	X	1	DATA BUS → 3 STATE

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA.

$\overline{CS}$ (Chip Select)

A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus is in the float state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable or $\overline{CTS}$ off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYNDET (SYNC Detect/ BRKDET Break Detect)

This pin is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bisync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "high" input signal can be removed. When External SYNC Detect is programmed, Internal SYNC Detect is disabled.

BREAK (Async Mode Only)

This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

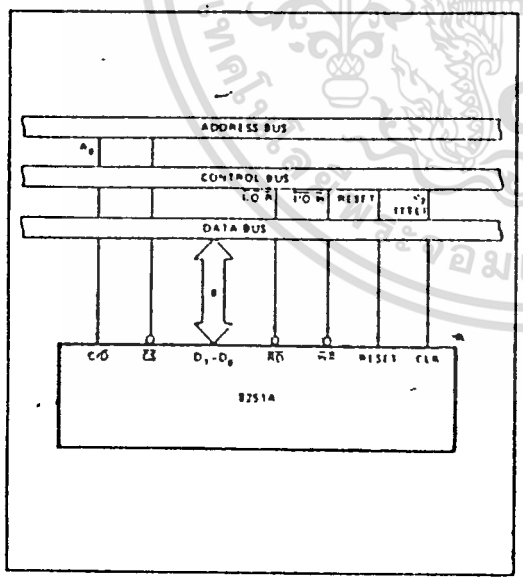


Figure 6. 8251A Interface to 8080 Standard System Bus

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.

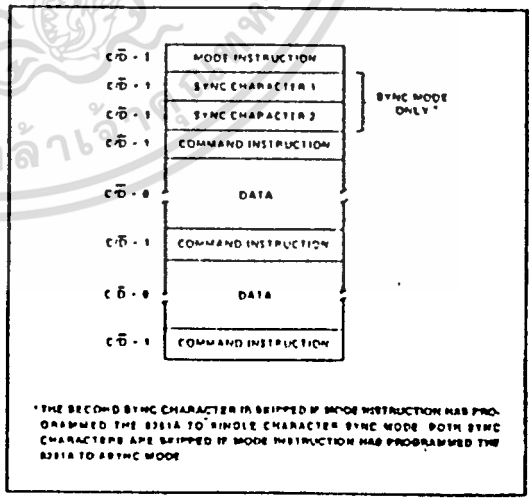


Figure 7. Typical Data Block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receiver Control

This functional block manages all receiver-related activities which consists of the following features

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition." Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid low (Start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = low).

Parity error detection sets the corresponding status bit.

The Framing Error status bit is set if the Stop bit is absent at the end of the data byte (asynchronous mode).

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

$\overline{RxC}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{RxC}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{RxC}$ frequency. A portion of the mode instruction selects this factor: 1, 1/16 or 1/64 the $\overline{RxC}$.

For example:

Baud Rate equals 300 Baud, If
$\overline{RxC}$ equals 300 Hz in the 1x mode;
$\overline{RxC}$ equals 4800 Hz in the 16x mode;
$\overline{RxC}$ equals 19.2 kHz in the 64x mode.

Baud Rate equals 2400 Baud, If
$\overline{RxC}$ equals 2400 Hz in the 1x mode;
$\overline{RxC}$ equals 38.4 kHz in the 16x mode;
$\overline{RxC}$ equals 153.6 kHz in the 64x mode.

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{TxC}$ and $\overline{RxC}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

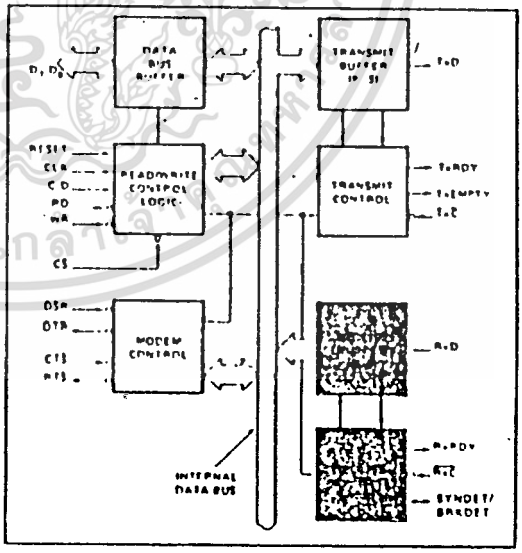


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is *not* masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to send, the TxEMPTY output will go "high." It resets upon receiving a character from CPU if the transmitter is enabled. TxEMPTY remains high when the transmitter is disabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplex operational mode.

In the Synchronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers." TxEMPTY does not go low when the SYNC characters are being shifted out.

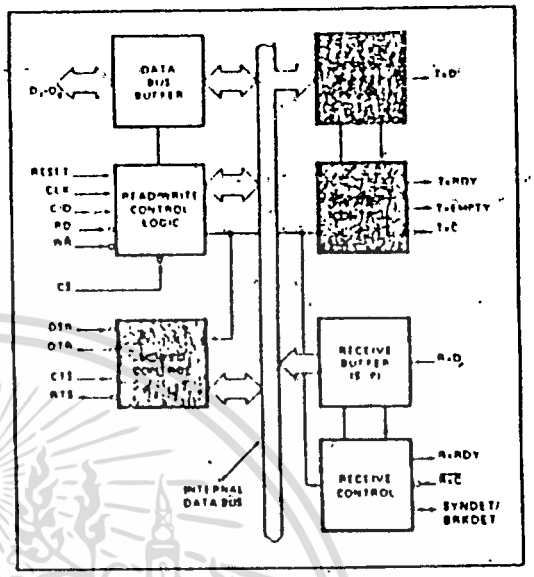


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

$\overline{TxC}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{TxC}$ frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual $\overline{TxC}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{TxC}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{TxC}$ equals 110 Hz in the 1x mode.
$\overline{TxC}$ equals 1.72 kHz in the 16x mode.
$\overline{TxC}$ equals 7.04 kHz in the 64x mode.

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of $\overline{RxC}$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

This instruction defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be written.

Command Instruction

This instruction defines a word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation (see Figure 7). The Mode Instruction must be written immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components, one Asynchronous and the other Synchronous, sharing the same package. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A, and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

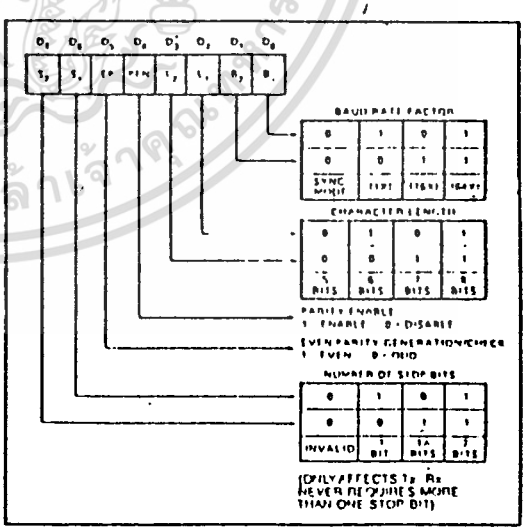


Figure 8. Mode Instruction Format, Asynchronous Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the *receiver* requires only *one* stop bit, regardless of the number of stop bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

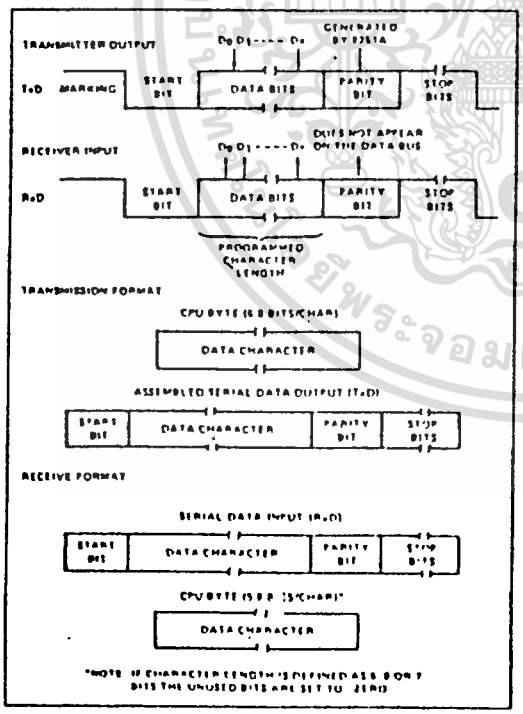
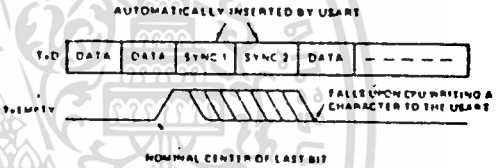


Figure 9. Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go low when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first command instruction word written. Data on the RxD pin is then sampled on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

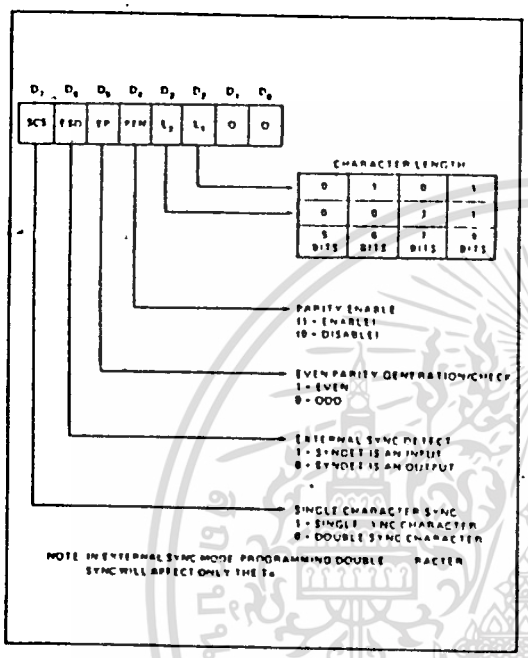


Figure 10. Mode Instruction Format, Synchronous Mode

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one," thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication.) When external SYNDET mode is selected, Internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.

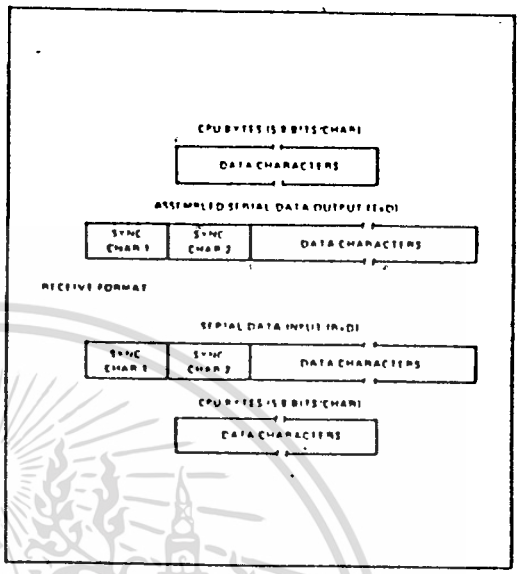


Figure 11. Data Format, Synchrono Mode

COMMAND INSTRUC ION DE INITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the sy c characters are loaded (if in Sync Mode) the the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command In truction

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" (C/D = 1) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

Note: Internal Reset on Power-up

When power is first applied, the 8251A may come up in the Mode, Sync character or Command format. To guarantee that the device is in the Command Instruction format before the Reset command is issued, it is safest to execute the worst-case initialization sequence (sync mode with two sync characters). Loading three 00Hs consecutively into the device with C/D = 1 configures sync operation and writes two dummy 00H sync characters. An Internal Reset command (40H) may then be issued to return the device to the "idle" state.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

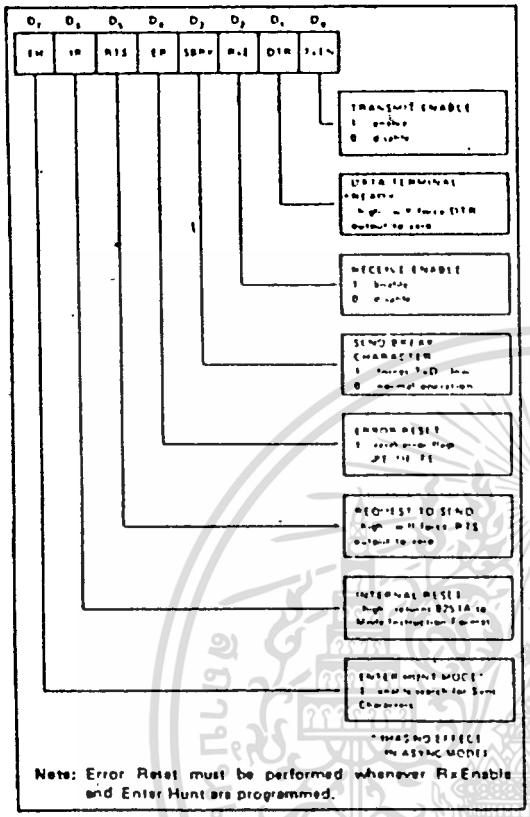


Figure 12. Command Instruction Format

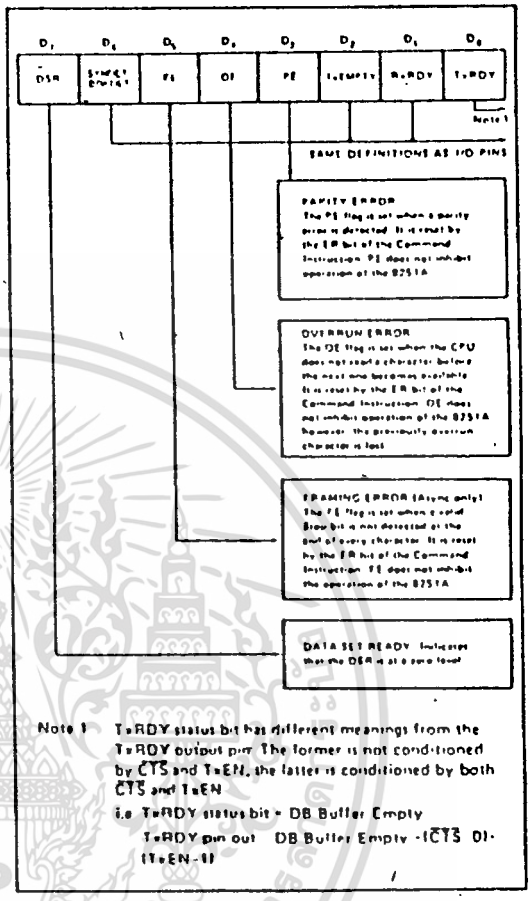


Figure 13. Status Read Format

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (Status update is inhibited during status read.)

A normal "read" command is issued by the CPU with C/$\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely polled or interrupt-driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.

APPLICATIONS OF THE 8251A

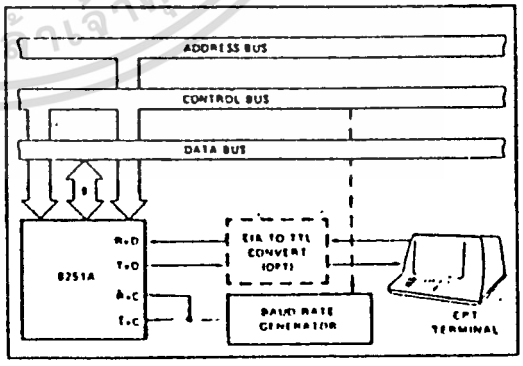


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

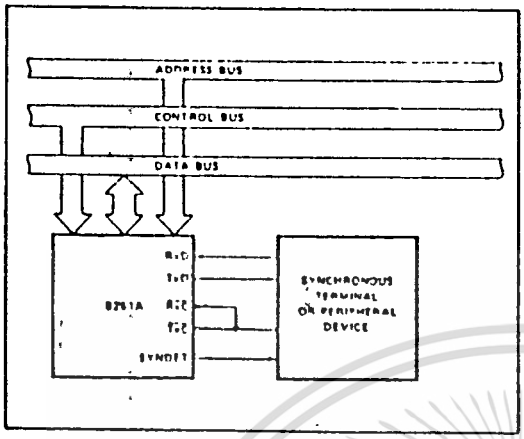


Figure 15. Synchronous Interface to Terminal or Peripheral Device

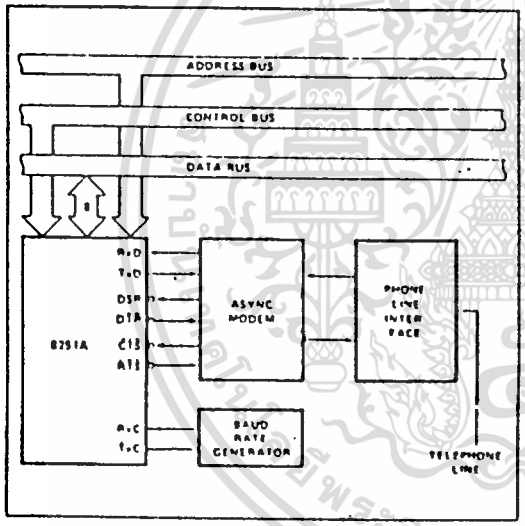


Figure 16. Asynchronous Interface to Telephone Lines

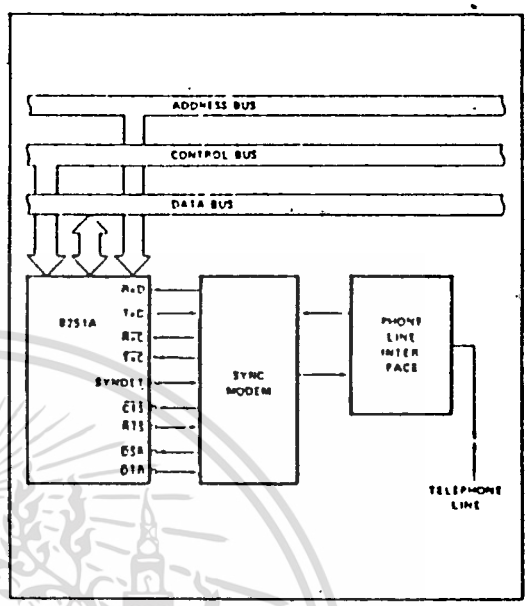


Figure 17. Synchronous Interface to Telephone Lines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

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

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ± 5%, GND = 0V)*

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OL} = −400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} TO 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} TO 0.45V
I_{CC}	Power Supply Current		100	mA	All Outputs = High

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

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

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ±10%, GND = 0V)*

Bus Parameters (Note 1)

READ CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	Note 2
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	Note 2
t_{RR}	$\overline{READ}$ Pulse Width	250		ns	
t_{RD}	Data Delay from $\overline{READ}$		200	ns	3. C_L = 150 pF
t_{DF}	$\overline{READ}$ to Data Floating	10	100	ns	

WRITE CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Stable Before $\overline{WRITE}$	0		ns	
t_{WA}	Address Hold Time for $\overline{WRITE}$	0		ns	
t_{WW}	$\overline{WRITE}$ Pulse Width	250		ns	
t_{DW}	Data Set-Up Time for $\overline{WRITE}$	150		ns	
t_{WD}	Data Hold Time for $\overline{WRITE}$	20		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	Note 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS (Continued)

OTHER TIMINGS

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Clock Period	320	1350	ns	Notes 5, 6
t_{ϕ}	Clock High Pulse Width	120	$t_{CY}-90$	ns	
$t_{\overline{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time		20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{TxC}$		1	μs	
f_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{TPW}	Transmitter Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
f_{Rx}	Receiver Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of Last Bit		14	t_{CY}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$		400	ns	Note 7
t_{RxRDY}	RxRDY Pin Delay from Center of Last Bit		26	t_{CY}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$		400	ns	Note 7
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$		26	t_{CY}	Note 7
t_{ES}	External SYNDET Set-Up Time After Rising Edge of $\overline{RxC}$	18		t_{CY}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit		20	t_{CY}	Note 7
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)		8	t_{CY}	Note 7
t_{CR}	Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)	20		t_{CY}	Note 7

*NOTE:

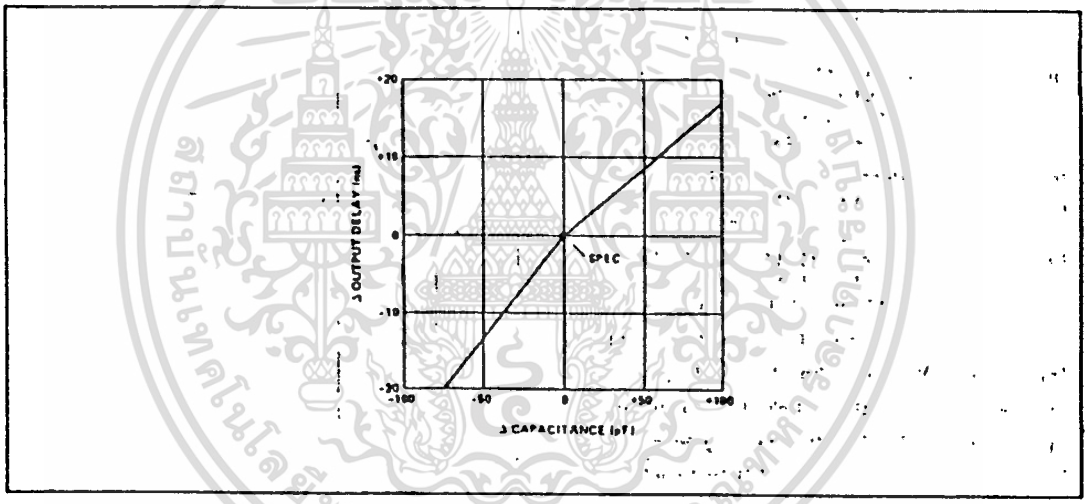
1. For Extended Temperature EXPRESS, use M8251A electrical parameters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

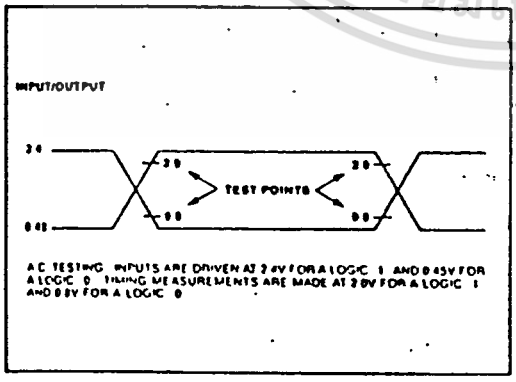
A.C. CHARACTERISTICS (Continued)

NOTES:
1. AC timings measured V_{OH} = 2.0, V_{OL} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select (CS) and Command/Data (C/D) are considered as Addresses.
3. Assumes that Address is valid before R_D↓.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1. Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}
5. The TxC and RxC frequencies have the following limitations with respect to CLK: For 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$.
   For 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$.
6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

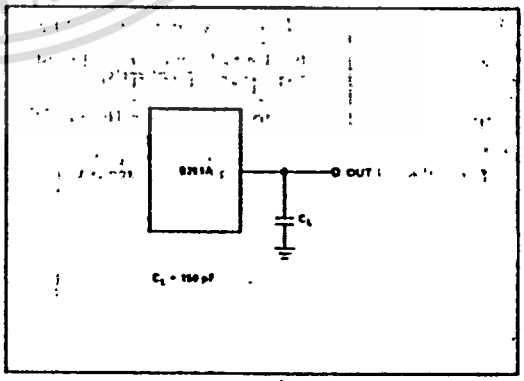
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)



A.C. TESTING INPUT, OUTPUT WAVEFORM



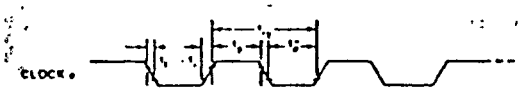
A.C. TESTING LOAD CIRCUIT



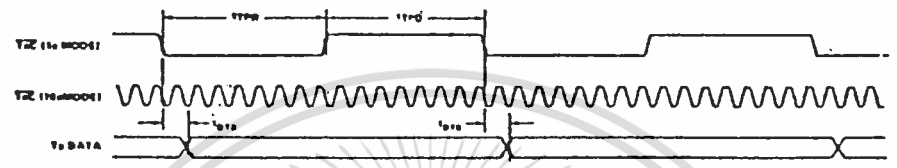
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

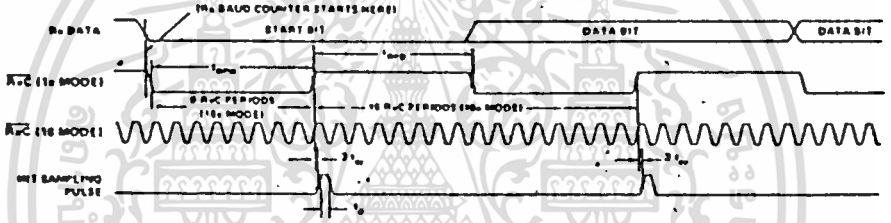
SYSTEM CLOCK INPUT



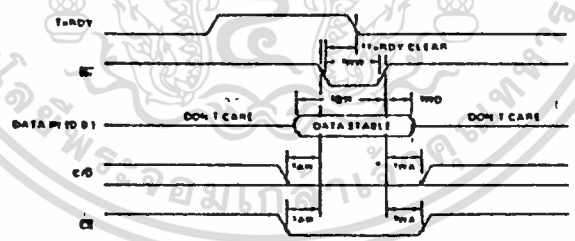
TRANSMITTER CLOCK AND DATA



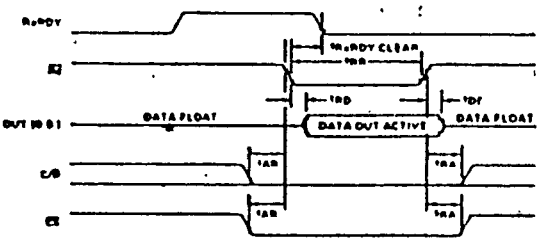
RECEIVER CLOCK AND DATA



WRITE DATA CYCLE (CPU → USART)



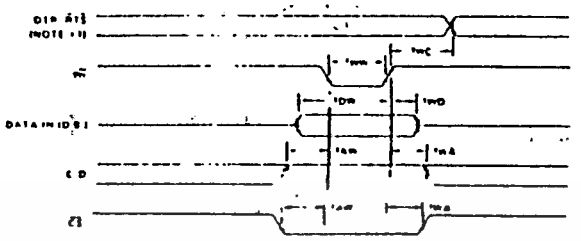
READ DATA CYCLE (CPU ← USART)



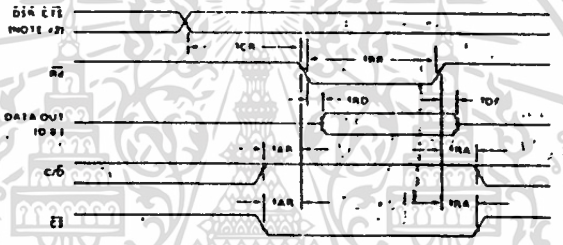
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS (Continued)

WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)



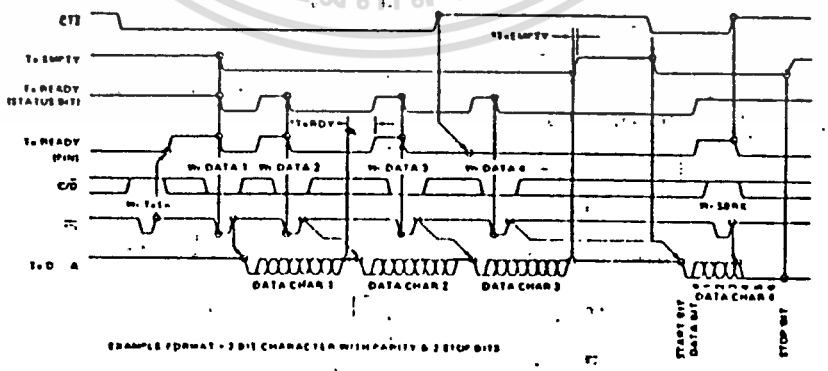
READ CONTROL OR INPUT PORT (CPU ← USART)



NOTE #1 T_{RC} INCLUDES THE RESPONSE TIMING OF A CONTROL BYTE

NOTE #2 T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxRDY CIRCUITRY.

TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)

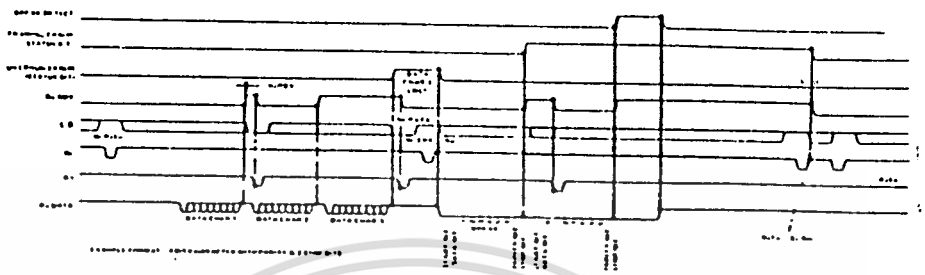


EXAMPLE FORMAT = 2 BIT CHARACTER WITH PARITY & 2 STOP BITS

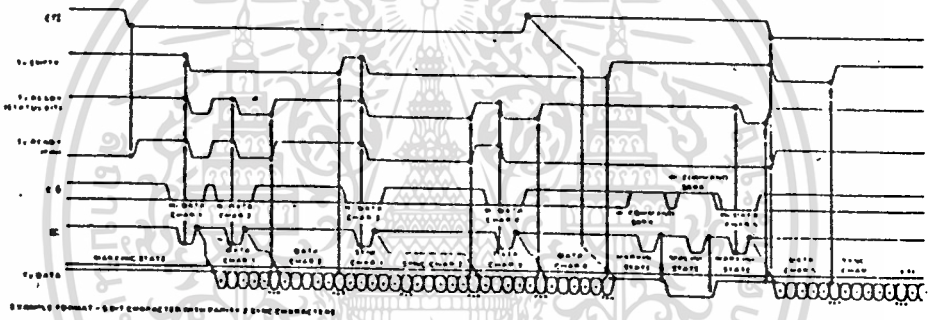
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS (Continued)

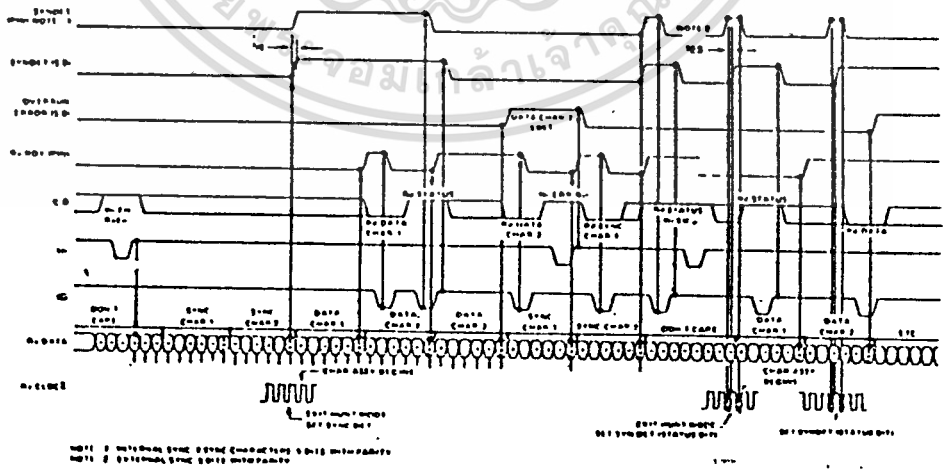
RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)

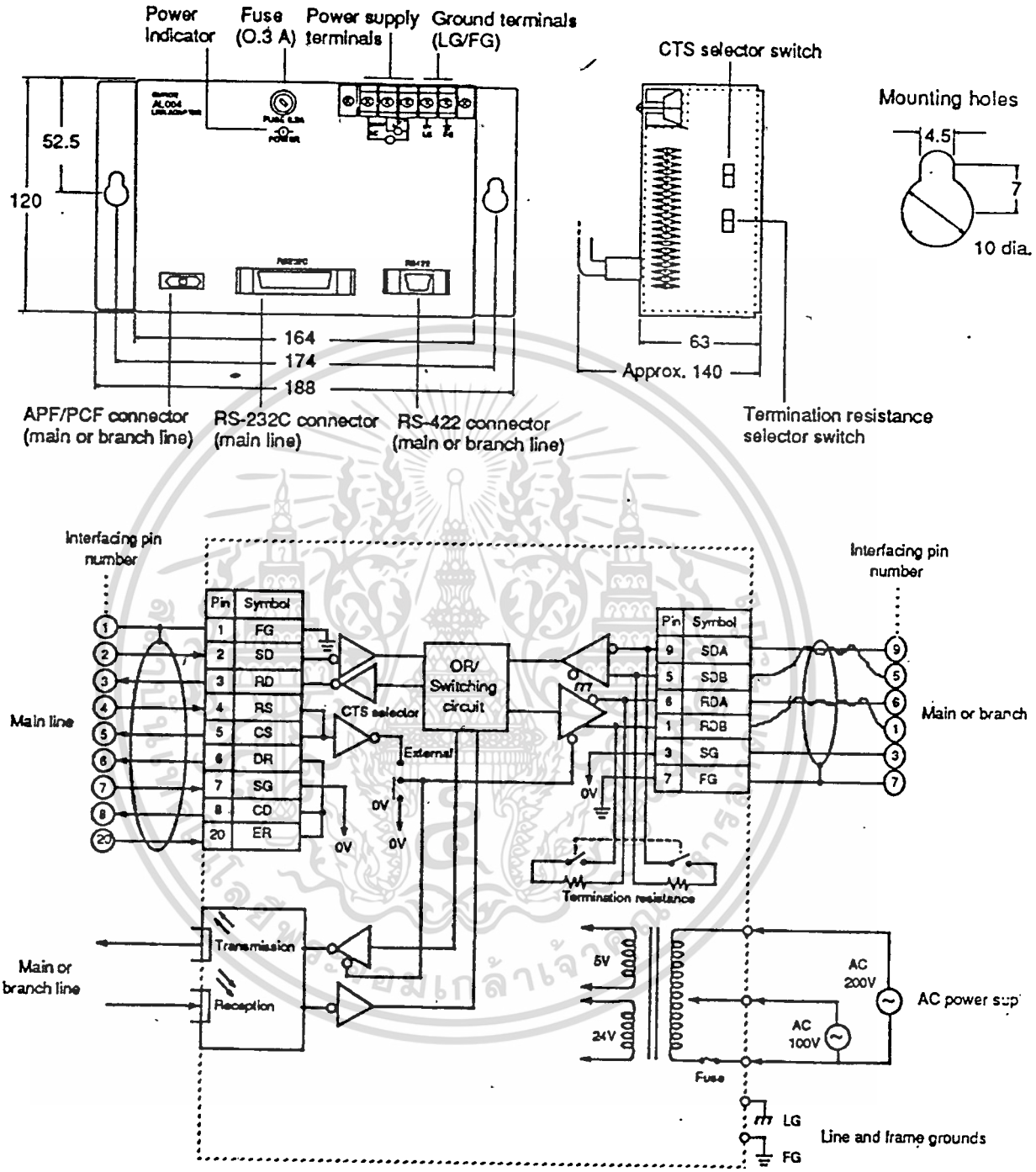


RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5–3 Link Adapter AL004-(P)E



Switch Settings

CTS Selector

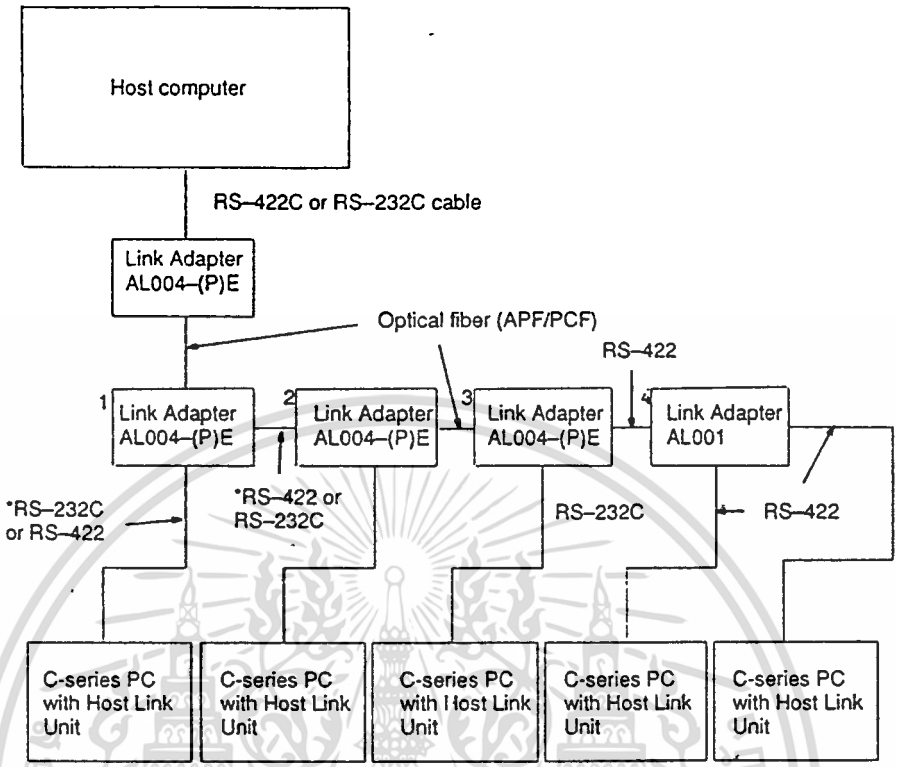
Set to O V to have the clear-to-send (CTS) signal always ON. Set to external to receive an external CTS signal.

Termination Resistance

When set to ON, connects the built-in termination resistance (220Ω); when set to OFF, disconnects it. Turn ON the termination resistance of the last Link Adapter at each end of each Subsystem.

เอกสารนี้มีขึ้นเพื่อใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example 2
Parallel Multiple-link System with Optical Fiber Cable



Note *When using AL004–(P)E Link Adapters, the connections can only include the following: one optical fiber, one RS–232C, and one RS–422. As the Host Link Units can only be connected to either RS–232C or RS–422 cables (depending on the model), the rest of the System must be configured according to these restrictons. To meet these limitations, and having AL004–(P)E models as the first three Link Adapters, the rightmost Link Adapter in the above System would need to be an AL001 with three RS–422 cable connections, and the corresponding Host Link Unit would need to be an LK202–EV1 model. (Other configurations are possible, e.g., if the top Link Adapter was omitted, the cable between Link Adapters 1 and 2 would need to be optical fiber, and this would affect the available connections for the remaining Units.) See *Appendix B* or a *Link Adapters Manual* for further information.

The above example uses AL004–(P)E Adapters to connect the Units in parallel. These Link Adapters bypass any Host Link Unit not properly connected so that the rest of the system can operate normally. Even if a power failure occurs in a Host Link Unit connected to a branch line of a Link Adapter, signals are still transmitted to the other Host Link Units.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UNIT RS_232;
INTERFACE
FUNCTION Hex(Hextodec : byte):string;
procedure initial8250;
procedure disableinterrupt;
procedure calcFCS( command:string;var fcs : string);
IMPLEMENTATION
procedure initial8250;
var LSB,MSB : byte;
begin
    LSB := $00;
    MSB := $30;
    port[$3FB]:= $80;
    port[$3F8]:= LSB;
    port[$3F9]:= MSB;
    port[$3FB]:= $1E;
    port[$3FC]:= $03;
end;
procedure DisableInterrupt;
begin
     port[$3F9]:= $00;
end;
Function Hex(HextoDec : byte):string;
var      LSB,MSB :byte;
         buffer  :char;
         Hexbuf  :char;
procedure Convert (HextoDec : byte; var Hex : char);
begin
     case HextoDec of
          0..9 :Hex := char(HextoDec + 48);
         10..15:Hex := char(HextoDec + 55);
     end;
end;
begin
     LSB := HextoDec mod 16;
     MSB := HextoDec div 16;
     Convert (LSB,Hexbuf);
     buffer := Hexbuf;
     Convert (MSB,Hexbuf);
     Hex := concat(Hexbuf,buffer);
end;
function CalcFCSDec(Var command : string):byte;
var      len,buffer,count : integer;
         ch : char;
begin
     len := length(command);
     buffer := 0;
     count:=1;
     while count <= len do
           begin
                buffer := ord(command[count]) xor buffer;
                count := count+1;
           end;
     calcFCSDec := buffer;
end;{CalcFCSDec}
procedure calcfcs(command: string;var fcs:string);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var fcsdec:integer;
begin
     fcsdec:= calcfcsdec(command);
     fcs := hex(fcsdec);
end;{ calcfcs}
END.
```

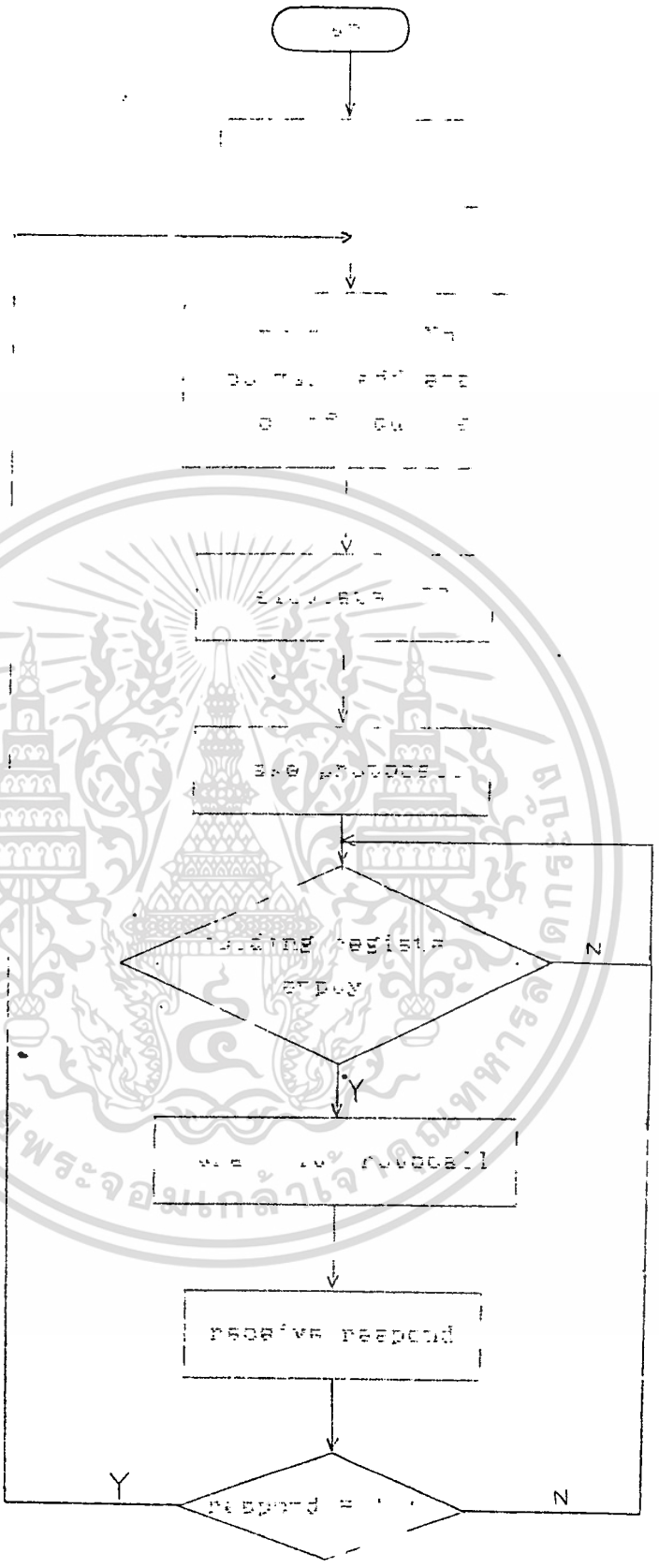
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program readfile;
uses crt,dos,rs_232;
var fp :file of string;
    data : string;
    filename : string[12];
    i : integer;
    dec : byte;
procedure initial8250;
var lsb,msb :byte;
begin
     msb :=$00;
     lsb :=$30;
     port[$3FB]:=$80;
     port[$3F8]:=lsb;
     port[$3F9]:=msb;
     port[$3FB]:=$1F;
     port[$3FC]:=$03;
end;
procedure convert(data:string;var dec:byte);
var i,x : integer;
    a   : char;
    mb,buffer : integer;
begin
     for i := 1 to 2 do
     begin
          case data[i] of
               '0': x:=0;        '8': x:=8;
               '1': x:=1;        '9': x:=9;
               '2': x:=2;        'A': x:=10;
               '3': x:=3;        'B': x:=11;
               '4': x:=4;        'C': x:=12;
               '5': x:=5;        'D': x:=13;
               '6': x:=6;        'E': x:=14;
               '7': x:=7;        'F': x:=15;
          end;{case}
          case i of
               1 : mb := x*16;
               2 : buffer := x;
          end;{case}
     end;
     dec := mb+buffer;
end;{convert}
procedure transmit(data:byte);
const Txdelay = 50;
var  linestatusregister:byte;
     modemstatusregister:byte;
begin
     repeat
          modemstatusregister:=port[$3FE];
     until (modemstatusregister and $10) <> 0 ;
     while (modemstatusregister and $10) <> 0 do
     begin
          linestatusregister :=port[$3FD];
          modemstatusregister :=port[$3FE];
          writeln('              ',hex(modemstatusregister));
          if (linestatusregister and $20) <> 0  THEN
             begin
                delay(txdelay);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    port[$3F8]:=data;
                    writeln(i,' : ',hex(i) ,': ' ,hex(data));
               end;
          modemstatusregister:=port[$3FE];
     end;
     delay(txdelay);
end;{transmit}
begin
     initial8250;
     write('filename ? ');
     readln(filename);
     assign(fp,filename);
     reset(fp);i:=1;
     while not eof(fp) do
     begin
          read(fp,data);
          convert(data,dec);{writeln(i,' : ',hex(dec));}
          transmit(dec);
          i:=i+1;
     end;
     close(fp);
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

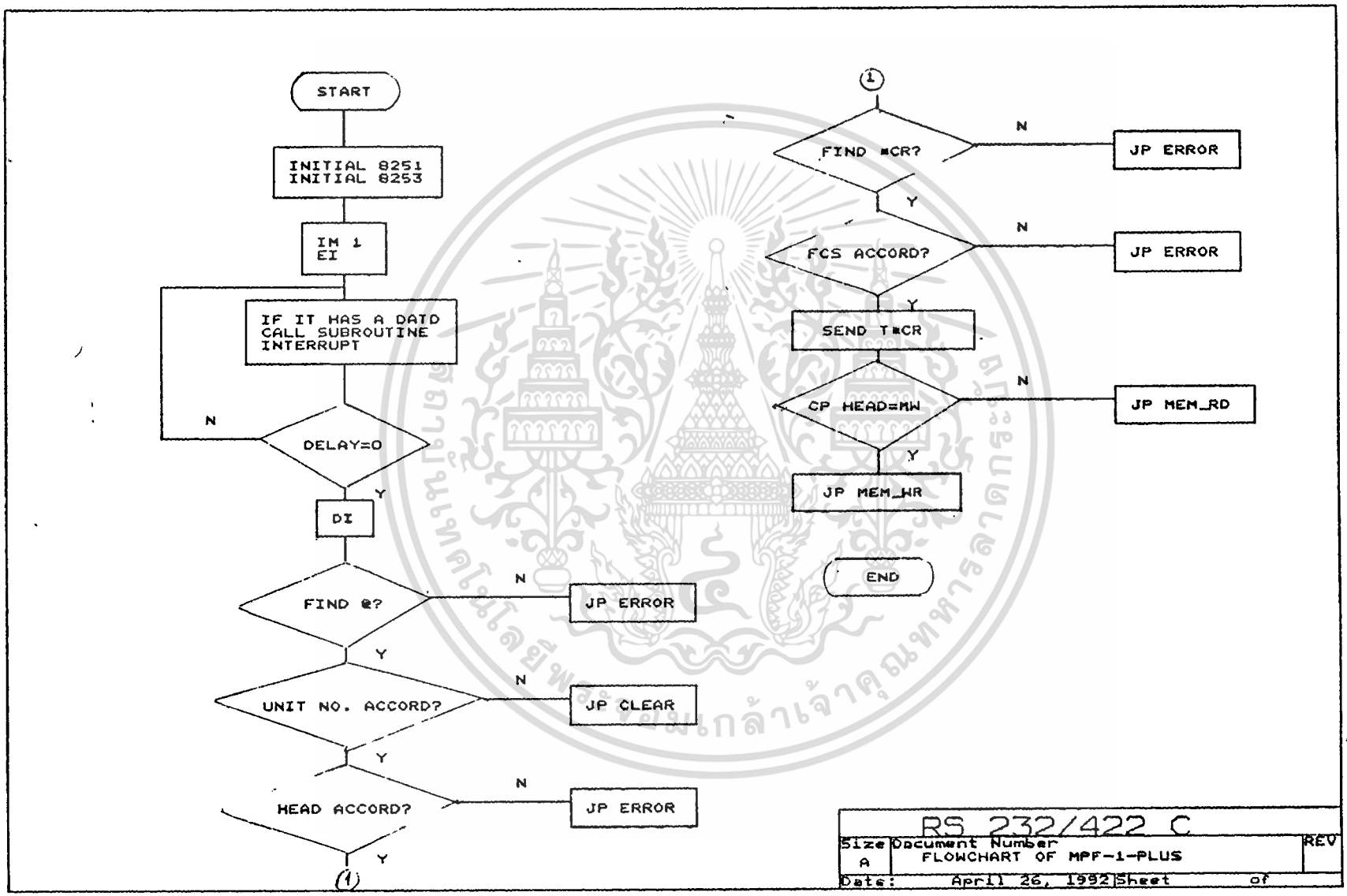
```
program new;
uses crt,rs_232;
var No,command,address,fcs,tprotocall,rprotocall  :string;
    prepro,rfcs,newdata,data:string;
    answer:boolean;
    respond :char;
    i,CALFCS,finish,fin:integer;
    receivedataregister,linestatusregister:byte;
procedure initial8250;
var LSB,MSB :byte;
BEGIN
     MSB :=$00;
     LSB :=$30;
     PORT[$3FB] :=$80;
     PORT[$3F8] :=LSB;
     PORT[$3F9] :=MSB;
     PORT[$3FB] :=$1E;
END;
procedure DisableInterrupt;
begin
     port[$3F9] := $00;
end;
PROCEDURE Transmit(tprotocall : string);
const Txdelay =100;
var len,count,LineStatusRegister : byte;
begin
     len :=length(tprotocall);
     count := 1;
     while count <= len do
     begin
     linestatusregister:=port[$3FD];
     if (linestatusregister and $20 )<> 0 then
     begin
          delay(Txdelay);
          port[$3F8] := ord(tprotocall[count]);
          writeln(HEX(ord(tprotocall[count])));
          count := count + 1 ;
          linestatusregister:=port[$3FD];
     end;
     end;
     delay(Txdelay);
end;{transmit}

procedure Receive(var rprotocall : string);
var ReceiveDataRegister : byte;
    chrpro:char;
begin
     rprotocall := '';
     chrpro:=' ';
     while chrpro <> '*'  do
          begin
               ReceiveDataRegister := port[$3F8];
               chrpro := chr( ReceiveDataRegister );
               rprotocall := concat(rprotocall,chrpro);
          end;
end;{Receive}
begin{main}
     initial8250;
     DisableInterrupt;
     repeat
     write('input unit number :'); readln(No);if No='#' then halt;
     write('MW/MR :');readln(command);
     write('Input address :');readln(address);
     write('input data :');readln(data);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      newdata:='';
      fin:=ord(data[0]);
      for i:=1 to fin do
      begin
           if data[i] <> #32 then
              begin
                   newdata:=concat(newdata,data[i]);
              end;
      end;
      prepro:=concat('@',No,command,address,newdata);
      calcfcs(prepro,fcs);
      tprotocall:=concat('@',No,command,address,newdata,fcs,'*CR');
      WRITELN(TPROTOCALL);
      repeat
      respond:=' ';
      while  respond <> 'T' do
          begin
               transmit(tprotocall);
               delay(2000);
               linestatusregister:=port[$3FD];
               if (linestatusregister and $1E) <> 0 then
               begin
                    writeln('error');
               end;
               receivedataregister:=port[$3F8];
               respond:=chr(receivedataregister);
          end;
      answer:=true;
      if command ='MR'  then
         begin
              receive(rprotocall);
              finish := ord(rprotocall[0]);writeln('finish =',finish);
              CALFCS:=0;
              for i := 1 to finish-3 do
                  begin
                       calfcs:=calfcs xor ord(rprotocall[i]);
                       WRITELN(I,' :',HEX(CALFCS));
                  end;
              writeln('calfcs =',calfcs);
              rfcs := concat(rprotocall[finish-2],rprotocall[finish-1]
              writeln('rfcs =',rfcs);
              if HEX(calfcs) <> rfcs then answer:=false
              else answer := true;
         if answer = true then
            begin
            WRITE(ADDRESS,' :');
                 for i := 1 to finish-3  do
                     begin
                          write(rprotocall[i]);
                     end;
                 writeln;
            end;
         end;{command= MR}
      until answer = true;
      until 1=2;
end.{main}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


START
INITIAL 8251
INITIAL 8253
IM 1
EI
IF IT HAS A DATD
CALL SUBROUTINE
INTERRUPT
DELAY=0
N
Y
DI
FIND @?
N
JP ERROR
Y
UNIT NO. ACCORD?
N
JP CLEAR
Y
HEAD ACCORD?
N
JP ERROR
Y
(1)
(1)
FIND *CR?
N
JP ERROR
Y
FCS ACCORD?
N
JP ERROR
Y
SEND T*CR
CP HEAD=MW
N
JP MEM_RD
Y
JP MEM_WR
END
RS 232/422 C
Size A
Document Number
FLOWCHART OF MPF-1-PLUS
REV
Date: April 26, 1992 Sheet of

```
                ORG     2000H
                LD      BC,300H
RE_START        CPD
                JP      PE,RE_START
                LD      A,40H           ;RESET 8251
                OUT     (11H),A
                LD      A,0FAH          ;MODE WORD TO 8251
                OUT     (11H),A
                LD      A,37H           ;COMMAND WORD TO 8253
                OUT     (33H),A
                LD      A,50H           ;LOW VALUES
                OUT     (30H),A
                LD      A,00H           ;HIGH VALVUES
                OUT     (30H),A
                LD      HL,0FE00H
                LD      D,(HL)          ;KEEP UNIT_NO. TO REG_D
                IM      1
;***************************************************************
MAIN:
                LD      A,D
                LD      (0FE00H),A
                LD      HL,INT
                LD      (0FF01H),HL
                LD      E,00H
                LD      HL,0FD00H
                LD      SP,0FEA0H
                LD      A,15H           ;COMMAND WORD TO 8251
                OUT     (11H),A
                EI
NO_DATA         LD      BC,DELAY
DE_L            DEC     BC
                LD      A,C
                OR      B
                JP      NZ,DE_L
                LD      A,E
                CP      00H
                JP      Z,NO_DATA
                DI
                LD      A,E
                CP      FAULT
                JP      Z,ERROR
                LD      HL,0FCFFH
CP_ALFA         INC     HL
                LD      A,L
                CP      0FFH
                JP      Z,ERROR
                LD      A,40H
                CP      (HL)
                JP      NZ,CP_ALFA
                LD      E,L             ;@_RIGHT
                INC     HL
                CALL    HIGH_LOW
                CP      D               ;COMPARE_UNIT NO.
                JP      NZ,CLEAR
                INC     HL
                LD      A,4DH
                CP      (HL)            ;COMPARE_M
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JP      NZ,ERROR
                INC     HL
                LD      A,(HL)

                CP      52H             ;COMPARE_R
                JP      Z,ANT
                CP      57H             ;COMPARE_W
                JP      NZ,ERROR
ANT             INC     HL
                CALL    HIGH_LOW
                LD      B,A
                INC     HL
                CALL    HIGH_LOW
                LD      C,A
                PUSH    BC
                EX      (SP),IX         ;REG_IX KEEP ADD
                POP     BC
                INC     HL
                LD      HL,0FDFFH
CP_END          DEC     HL
                LD      A,L
                CP      00H
                JP      Z,ERROR
                LD      A,52H           ;CP_R
                CP      (HL)
                JP      NZ,CP_END
                DEC     HL
                LD      A,43H           ;CP_C
                CP      (HL)
                JP      NZ,ERROR
                DEC     HL
                LD      A,2AH           ;CP_*
                CP      (HL)
                JP      NZ,ERROR
                DEC     HL
                DEC     HL              ;TEST FCS
                LD      A,L
                LD      (0FDFFH),A
                CALL    HIGH_LOW
                LD      B,A             ;REG_B KEEP FCS
                DEC     HL
                DEC     HL
                DEC     E
                LD      C,00H
CHK_FCS         LD      A,(HL)
                XOR     C
                LD      C,A
                DEC     HL
                LD      A,L
                CP      E
                JP      NZ,CHK_FCS
                INC     E
                LD      A,C
                CP      B
                JP      NZ,ERROR        ;FCS ERROR
                LD      HL,HONDA        ;SEND T*CR
                CALL    SEND
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD      H,0FDH
                LD      A,04H
                ADD     A,E
                LD      L,A
                LD      A,52H
                CP      (HL)
                JP      NZ,MEM_WR

                JP      MEM_RD
;*****************************************************************
MEM_RD:
                LD      H,0FDH
                LD      A,09H
                ADD     A,E
                LD      L,A
                CALL    HIGH_LOW
                LD      E,A
                LD      C,00H
                LD      HL,0FD02H
                LD      (HL),0DH
RRE_1           LD      A,(IX+00)
                LD      IY,0FD00H
                LD      HL,0FD03H
                CALL    READ
                INC     IX
                DEC     E
                JP      NZ,RRE_1
                LD      A,C
                LD      IY,0FD00H
                LD      HL,0FD03H
                CALL    READ
                LD      HL,BMW
                CALL    SEND
                JP      MAIN
;*****************************************************************
MEM_WR:
                LD      H,0FDH
                LD      A,09H
                ADD     A,E
                LD      L,A
DATA            LD      A,(0FDFFH)
                CP      L
                JP      Z,MAIN
                CALL    HIGH_LOW
                LD      (IX+00),A
                INC     IX
                INC     HL
                JP      DATA
;*****************************************************************
ERROR:
                LD      HL,BENZ
                CALL    SEND            ;SEND F*CR
CLEAR           LD      B,0FFH
                LD      HL,0FD00H
                LD      A,0FFH
DIRTY           LD      (HL),A
                INC     HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ    DIRTY
                JP      MAIN
;***************************************************************
;FUNCTION: CONVERT ASC II CODES TO CORRESPONDING
;          HEXADECIMAL VALUES.
HIGH_LOW:
                PUSH    BC
                LD      A,(HL)
                CP      41H
                JP      C,HEN
                SUB     07H

HEN             LD      B,04H
FOX             SLA     A
                DJNZ    FOX
                PUSH    AF
                INC     HL
                LD      A,(HL)
                CP      41H
                JP      C,CAT
                SUB     07H
CAT             AND     0FH
                LD      C,A
                POP     AF
                ADD     A,C
                POP     BC
                RET
;***************************************************************
;FUNCTION: CONVERT HEXADECIMAL VALUES TO CORRESPONDING
;          ASC II CODE AND SEND OUT.
READ:
                LD      B,02H
                LD      (HL),A
COW             AND     0FH
                RLD
                CP      0AH
                JP      C,RAT
                ADD     A,07H
RAT             ADD     A,30H
                LD      (IY+00),A
                XOR     C
                LD      C,A
                INC     IY
                DJNZ    COW
                LD      HL,0FD00H
                CALL    SEND
                RET
;***************************************************************
SEND:
                LD      A,31H
                OUT     (11H),A
OUTT            LD      A,(HL)
                OUT     (10H),A         ;OUTPUT CHARACTER TO TX
TESTT           IN      A,(11H)         ;READ STATUS REGISTER
                BIT     0,A             ;TEST BIT D0=1
                JP      Z,TESTT         ;BUFFER NOT EMPTY
                LD      B,0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DE_LAY          DJNZ    DE_LAY
                INC     HL
                LD      A,(HL)
                CP      0DH
                JP      NZ,OUTT
                LD      A,10H
                OUT     (11H),A
                RET
;***************************************************************
;SUBRUTINE FOR THE INTERRUPT MODE 1 WHEN HAS RX_RDY INTO
                IN      A,(10H)
                LD      (HL),A
                INC     HL
                LD      A,H
                CP      0FEH

                JP      NZ,EXIT
                LD      BC,0001H
                LD      E,FAULT
                RETI
EXIT            LD      BC,DELAY
                LD      E,MESSAGE
                EI
                RETI
;***************************************************************
BENZ            DB      'F*CR'
                DB      0DH
HONDA           DB      'T*CR'
                DB      0DH
BMW             DB      '*CR'
                DB      0DH
DELAY           EQU     0FFFFH
INT             EQU     219EH
FAULT           EQU     0DDH
MESSAGE         EQU     0AAH
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WILLIAM COMPUT Z80 ASSEMBLER -  VERSION 1.03M  SERIAL #0010e

SOURCE FILE NAME: RS.ASM
                                PAGE  6

FFFF            DELAY           EQU     OFFFFH
219E          . INT             EQU     219EH
OODD            FAULT           EQU     ODDH
OOAA            MESSAGE         EQU     OAAH
0000                            END
```

```
WILLIAM COMPUT Z80 ASSEMBLER -  VERSION 1.03M  SERIAL #0010e

SOURCE FILE NAME: RS.ASM
                                PAGE  7
----- SYMBOL TABLE -----

INT         207B            OP_BUFF     2052            ERR_NR      231C
INT         219F            RAT         216E            RENZ        2318
OP_END      208D            EXIT        21AF            MAIN        2000
READ        2160            BAUD        21C1            DATA        2110
FAULT       0000            MEM_LO      20DB            RE_START    2000
CAT         2159            DELAY       FFFF            TOA         2141
MEM_WR      2113            RRE_1       20ED            CHE_TOS     0017
DE_L        203C            MEN         2149            MESSAGE     00AA
SEND        2180            CMPAR       2132            DA_LO       0000
HIGH_LOW    2140            NO_DATA     2039            TEST1       2182
COW         2163            DIRT        2139            ILLEGAL     0000
WITT        2184
***** NO ERRORS DETECTED *****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Z80 ASSEMBLY LANGUAGE PROGRAMMING, Lance A. leventhal

2. มาตรฐานสำหรับการสื่อสารข้อมูล, ชวลิต ลิมป์โสวรรณ. บทความพิเศษ จาก คอมพิวเตอร์รีวิว ฉบับที่ 82

3. การใช้งาน Z80, ศูนย์ภาษาคอมพิวเตอร์

4. การสื่อสารข้อมูล, ชุชัย ธนสารตั้งเจริญ, ทินกร ดุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำขอกราบขอบคุณ อาจารย์ สุพรรณ กุลพานิชย์ และคณาจารย์ทุกท่าน ที่กรุณา แนะนำ และให้คำปรึกษาทั้งทางทฤษฎี และปฏิบัติด้วยดีเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้