สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โมเด็มสำหรับเส้นใยนำแสง

MODEM

FOR OPTICAL FIBER

โดย

นาย จตุรงค์ สิทธิโชคธนานนท์

นาย พงศ์พิสิฐ บุญเกิด

นางสาว พรทิพย์ วงศ์ชนะพิบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมอิเลคทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าฯเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

ปพ.
ค136ส
2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032555

ปีการศึกษา 2535

โมเด็มสำหรับเส้นใยนำแสง

โดย

นาย จตุรงค์ สิทธิโชคธนานนท์

นาย พงศ์พิสิฐ บุญเกิด

นางสาว พรทิพย์ วงศ์ชนะพิบูลย์

อาจารย์ที่ปรึกษา

อาจารย์ สมศักดิ์ เชียร์ศิริกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032555

ปริญญานิพนธ์ปีการศึกษา 2535

ภาควิชา อิเลคทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าฯเจ้าคุณทหารลาดกระบัง

เรื่อง โมเด็มสำหรับเส้นใยนำแสง

ผู้จัดทำ

1. นาย จตุรงค์ ลิทธิโชคธนานนท์ 32.1049
2. นาย พงศ์พิสิฐ บุญเกิด 32.1198
3. นางสาว พรทิพย์ วงศ์ชนะพิบูลย์ 32.1204

.....

(อาจารย์ สมศักดิ์ เชียร์ศิริกุล) อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODEM FOR OPTIC FIBER

Mr. Jatturong Sittichokkananon

Mr. Pongpisit Boonkerd

Miss.Prontip Vongchanapiboon

ABSTRACT

Now the communication with optic fiber is very popular because it have very advantages. Especially it have low attenuation and it can send very far too. In this project we learn about the communication with optic fiber and to communicate signal with point to point by invent two modem that transmit and receive data for communicate data from computer pass the optic fiber. It communicate the terminal with RS-232-C in many speed example 1200,2400,4800, 9600...bit per sec. This communicatation are synchronous' and asynchronous that transmit and receive digital signal pass channal.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเด็มสำหรับเส้นใยนำแสง

จตุรงค์ สิทธิโชคคณานนท์
พงศ์พิสิฐ บุญเกิด
พรทิพย์ วงศ์ชนะพิบูลย์

บทคัดย่อ

ในปัจจุบันการสื่อสารด้วยเส้นใยนำแสงนับได้ว่ามีความนิยมสูงมาก เนื่องจากมีข้อดีปลายประการโดยเฉพาะอย่างยิ่งมีการลดทอนของสัญญาณน้อยและยังสามารถส่งไปได้ไกลด้วย ในปริญญานิพนธ์ฉบับนี้ ได้ศึกษาการสื่อสารด้วยเส้นใยนำแสงและทำการสื่อสารแบบระยะสั้นจุดต่อจุด คือทำชุดโมเด็มรับส่งข้อมูล 2 ชุด สำหรับติดต่อสื่อสารข้อมูลจากคอมพิวเตอร์ผ่านเส้นใยนำแสง โดยมีการติดต่อกับอุปกรณ์ปลายทางแบบ RS-232-C ด้วยความเร็วต่างๆ เช่น 1200,2400,4800,... BIT PER SEC ซึ่งการสื่อสารข้อมูลนี้จะเป็นแบบ SYNCRONOUS,ASYNCRONOUS ด้วยการรับส่งสัญญาณเป็นแบบดิจิตอลผ่านช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1	ทฤษฎี และหลักการ	
	การสื่อสาร	1
	ข้อมูล และการส่งข้อมูล	5
	เส้นใยแสง	27
	RS-232-C	33
	Z80180	36
	พอร์ทสื่อสาร 8250	49
	โมเด็ม	59
	BASE BAND TRANSMISSION	63
	PHASE LOCKED LOOP	65
บทที่ 2	การออกแบบวงจร	68
บทที่ 3	การทดสอบ และผลการทดสอบ	74
	กิตติกรรมประกาศ	
	หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

ทฤษฎีและหลักการ

การสื่อสาร

ในชีวิตประจำวันของมนุษย์เรานั้น แทบจะกล่าวได้ว่าต้องเกี่ยวข้องอยู่กับการสื่อสารในรูปแบบใดรูปแบบหนึ่งอยู่ตลอดเวลา ไม่ว่าจะเป็นการสนทนากัน การอ่านหนังสือ การรับ/ส่งจดหมาย การพูดโทรศัพท์ การดูภาพยนต์ ตลอดจนการนั่งฟังบรรยายในห้อง ล้วนเป็นการสื่อสารทั้งสิ้น แต่มีรูปแบบของการสื่อสารที่แตกต่างกันไปตามคุณสมบัติเฉพาะของแต่ละชนิด แต่ไม่ว่าจะเป็นการสื่อสารรูปแบบใดๆ ก็ตามต่างมีคุณสมบัติร่วมกันข้อหนึ่งคือ จุดมุ่งหมาย นั่นคือการส่งข่าวสารหรือถ่ายทอดข่าวสารจากจุดหนึ่งไปยังอีกจุดหนึ่งอย่างถูกต้อง

1.1 หลักการเบื้องต้นของการสื่อสาร

ในการสื่อสารหรือการส่งข่าวสารจากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ไม่ว่าจะเป็นการสื่อสารในรูปแบบใดก็ตามจะต้องประกอบด้วยส่วนประกอบหลักเบื้องต้น 3 ส่วนคือ

1 แหล่งต้นกำเนิด (Source)

2 ตัวกลาง (Medium)

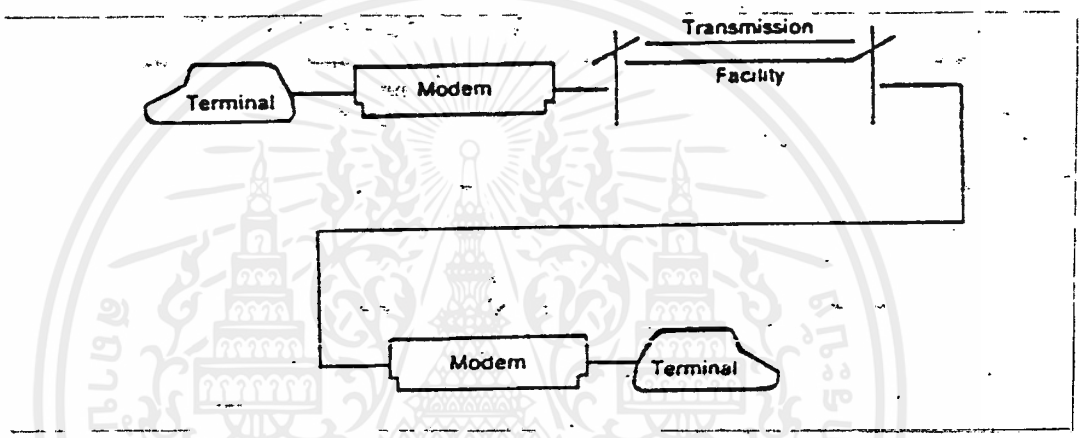
3 แหล่งรับ (Receiver)

แหล่งต้นกำเนิดเป็นส่วนที่ทำหน้าที่สร้างข่าวสารขึ้นมา เพื่อส่งผ่านตัวกลางไปยังแหล่งซึ่งเป็นจุดหมายปลายทาง ส่วนประกอบทั้งสามนี้เป็นส่วนประกอบสำหรับโครงสร้างหลักที่เล็กที่สุดของระบบการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ระบบการสื่อสารข้อมูล

ระบบการสื่อสารข้อมูลก็มีลักษณะเช่นเดียวกับระบบการสื่อสารโดยทั่วไป คือประกอบด้วยแหล่งต้นกำเนิดที่ผลิตข่าวสาร แล้วส่งออกไปยังแหล่งรับผ่านทางตัวกลาง ตัวอย่างของระบบการสื่อสารข้อมูลพื้นฐาน แสดงในรูปที่ 1.1



รูปที่ 1.1 ส่วนประกอบพื้นฐานของระบบการสื่อสารข้อมูล

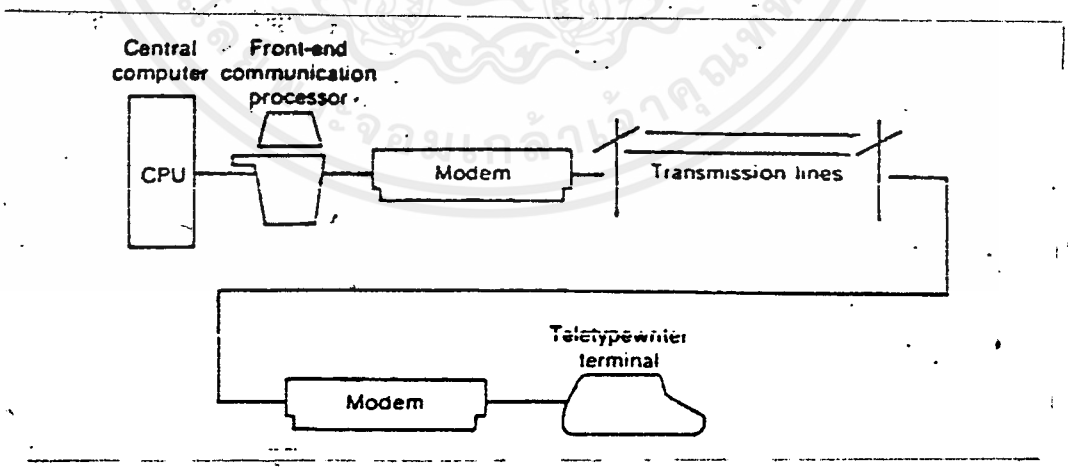
จากรูปที่ 1.1 ระบบประกอบด้วย อุปกรณ์ปลายทาง (Terminal) ทำหน้าที่เป็นแหล่งต้นกำเนิด (Source)ในการส่งข่าวสาร และ แหล่งปลายทาง (Sink) หรือแหล่งรับ (Receiver) ทำหน้าที่รับข่าวสารที่ส่งมาจากแหล่งต้นกำเนิดผ่านตัวกลาง ระหว่างอุปกรณ์ปลายทางทั้งสองจะมีอุปกรณ์ที่เรียกว่า โมเด็ม (MODEM) ต่ออยู่

สำหรับตัวกลางในการสื่อสารข้อมูลนั้น ตามปกติมักใช้สายโทรศัพท์ที่มีบริการเป็นสาธารณะทั่วไป สายโทรศัพท์นั้นเหมาะที่จะใช้ส่งสัญญาณในช่วงของสัญญาณเสียง เมื่อนำมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบการสื่อสารข้อมูล ที่ข้อมูลหรือข่าวสาร มีลักษณะบางอย่าง ไม่เหมาะสมกับการส่งไปตามสายโทรศัพท์ เราใช้โมเด็มสำหรับเปลี่ยนข้อมูลที่ออกมาจากอุปกรณ์ปลายทางให้อยู่ในรูปแบบที่เหมาะสมสำหรับส่งไปตามสายโทรศัพท์ ส่วนทางด้านรับนั้น เราใช้โมเด็มอีกตัวหนึ่งทำหน้าที่เปลี่ยนสัญญาณที่ส่งมาตามสายโทรศัพท์ให้กลับเป็นสัญญาณสำหรับอุปกรณ์ปลายทาง

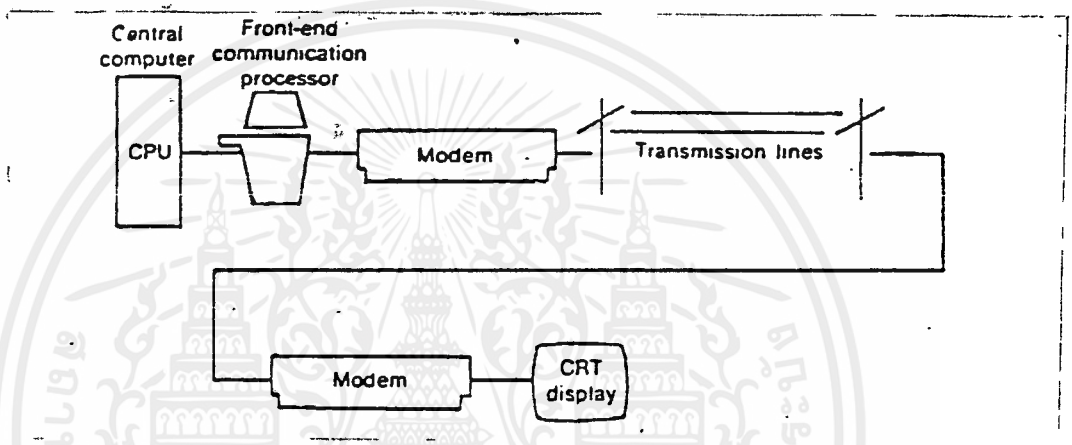
สำหรับอุปกรณ์ปลายทางนั้น อาจเป็นได้ตั้งแต่อุปกรณ์ที่ทำหน้าที่เพียงการรับ หรือแสดงผลข้อมูล ไปจนกระทั่งเป็นตัวคอมพิวเตอร์ก็ได้ รูป 1.2 แสดงถึงระบบการสื่อสารข้อมูลของคอมพิวเตอร์ แบบจุดต่อจุด (Point to Point Data Communication Network) ในรูปอุปกรณ์ปลายทางด้านหนึ่งคือเครื่องคอมพิวเตอร์ และอีกด้านหนึ่งคือเครื่องพิมพ์ดีดข้อมูล (Teletpye write) ที่ต่อเชื่อมด้วยตัวกลางประเภทสาย ในระบบประกอบด้วยโมเด็ม ที่แต่ละด้านของสายทำหน้าที่เปลี่ยนข้อมูล จากรูปแบบที่คอมพิวเตออร์เข้าใจไปเป็นรูปแบบที่ส่งบนสายได้ หรือเปลี่ยนกลับที่ทางด้านรับ เครื่องพิมพ์ดีดข้อมูลนั้นเป็นอุปกรณ์ สำหรับการป้อนข้อมูลไปยังคอมพิวเตอร์ หรือแสดงผลข้อมูลที่มาจากคอมพิวเตอร์



รูปที่ 1.2 ระบบการสื่อสารข้อมูลที่ใช้เครื่องพิมพ์ดีดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.3 แสดงถึงตัวอย่าง ระบบการสื่อสารข้อมูลที่มีอุปกรณ์ปลายด้านหนึ่งเป็นคอมพิวเตอร์ และอีกทางด้านหนึ่งเป็นจอภาพ ความแตกต่างระหว่างรูปที่ 1.2 และรูปที่ 1.3 อยู่ที่อุปกรณ์



รูปที่ 1.3 ระบบการสื่อสารข้อมูลที่ใช้จอภาพ

นอกจากตัวอย่างของระบบโครงข่ายการสื่อสารแบบพื้นฐานที่กล่าวมาแล้ว เรายังสามารถจัดรูปแบบ ของระบบการสื่อสารข้อมูลแบบต่าง ๆ ได้อีกหลายแบบ ซึ่งเราจะกล่าวถึงรายละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลและการส่งข้อมูล

วัตถุประสงค์หลักในระบบการสื่อสารข้อมูลของเราก็คือ การส่งข่าวสารหรือข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่ง ข้อมูลอาจจะอยู่ในรูปของตัวอักษร ตัวเลข หรืออาจจะเป็นเครื่องหมายต่างๆ ระยะทางระหว่างจุดสองจุดอาจจะใกล้หรือไกลก็ได้ ข้อมูลต่างๆที่ใช้ส่งนั้นเราไม่อาจที่จะส่งไป ในลักษณะเดียวกับรูปร่างที่เรามองเห็นในกระดาษพิมพ์ได้ จึงจำเป็นที่จะต้องเปลี่ยนรูปแบบ ให้อยู่ในรูปแบบทางไฟฟ้า ที่เหมาะสมกับการส่งไปตามสายสื่อสาร และข่าวสารที่ถูกเปลี่ยนรูปแบบไปนี้จะถูกนำมาเปลี่ยนกลับให้เป็นรูปแบบเดิมที่จุดรับ รูปแบบของข้อมูลในระบบการสื่อสารข้อมูล คือรูปแบบในลักษณะเลขฐานสองที่มีค่าเพียงสองค่า ซึ่งก็ไม่ได้นำเอารูปแบบของเลขฐานสองมาใช้โดยตรง แต่นำมาสร้างเป็นรหัสต่างๆ สำหรับใช้งานซึ่งมีหลายชนิด ในบทนี้ จะกล่าวถึงลักษณะต่างๆ ของข่าวสาร หรือข้อมูล ซึ่งถูกนำมาเปลี่ยนเป็นรหัสรูปแบบต่างๆ โดยจะเริ่มกล่าวตั้งแต่ลักษณะของเลขฐานสอง การเข้ารหัสของเลขฐานสองชนิดต่างๆ การปฏิบัติเกี่ยวกับข้อผิดพลาดที่อาจจะเกิดขึ้นในการส่งรหัสรวมทั้งการตรวจจับหาความผิดพลาด และการแก้ไขความผิดพลาดที่เกิดขึ้น

ความเร็วในการส่งข้อมูล (Baud)

ในการส่งข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ความเร็วในกาส่งข้อมูลเป็นปัจจัยสำคัญอันหนึ่ง ที่จะบอกให้ทราบถึงความสามารถของระบบ โดยปกติแล้วเราจะมีหน่วยในการวัดความเร็วของการส่งข่าวสาร สำหรับข่าวสาร หรือข้อมูล ที่อยู่ในรูปของเลขฐานสองนั้น เมื่อเรานำมาส่งในระบบสื่อสาร เราต้องเปลี่ยนเลขฐานสอง เป็นรูปแบบทางไฟฟ้าก่อนโดยใช้สัญญาณพัลซ์ (Pulse Signal) เขียนแทน กล่าวคือ ในกรณี 0 เราแทนด้วยการไม่มีสัญญาณพัลซ์ ส่วนกรณีของ 1 เราแทนด้วยการมีสัญญาณพัลซ์ ฉนั้น หน่วยที่ใช้วัดความเร็วในการส่งข้อมูล จึงวัดจากจำนวนพัลซ์ที่ส่งไปได้ใน 1 วินาที เราเรียกหน่วยนี้ว่า โบด(Baud)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ บิทต่อวินาที (Bit per Second : bps)

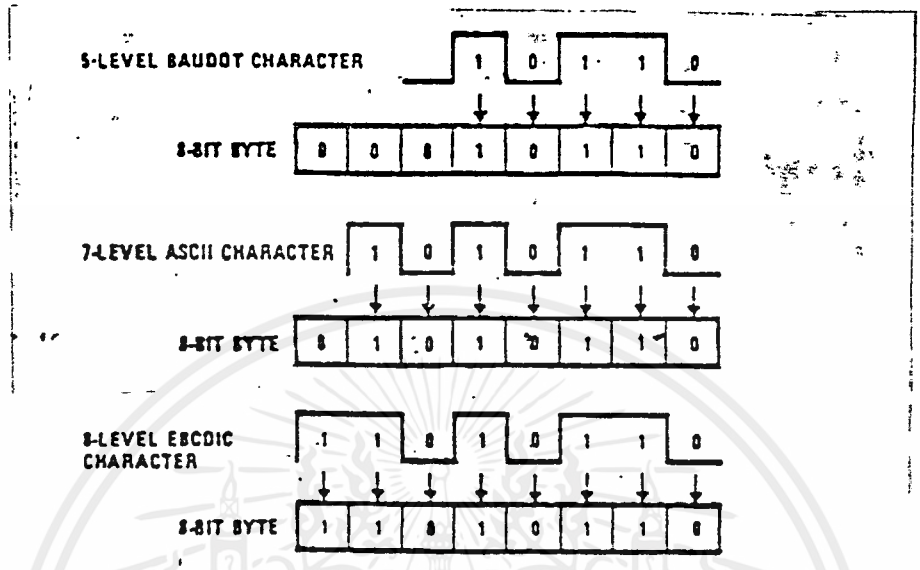
การส่งข้อมูล

ดังได้กล่าวมาแล้วว่าข้อมูลในระบบการสื่อสารข้อมูลนั้น เราเขียนแสดงด้วยค่าในเลขฐานสองหรือที่เรียกว่า ไบนารี่ ที่มีค่าตัวเลขที่ใช้สองตัวคือ 0 กับ 1 ในการใช้งานนั้นเรานำเอา 0 กับ 1 นี้มาประกอบเป็นรหัสชนิดต่าง ๆ แต่ในการส่งข้อมูลนั้น เราอาศัยวิธีการส่งทางไฟฟ้า ข้อมูลที่ส่งจะถูกเปลี่ยน ให้อยู่ในรูปแบบทางไฟฟ้า โดยการใช้ค่าสัญญาณทางไฟฟ้า 2 ระดับคือระดับสูงและระดับต่ำ ระดับสูงใช้แทนค่า 1 และ ระดับต่ำใช้แทนค่า 0 ระดับของสัญญาณนี้ จะถูกนำมาประกอบกัน ตามการประกอบกันของบิท ที่แทนตัวอักษรหรือตัวเลขนั้น ๆ ตัวอย่างสัญญาณทางไฟฟ้า ที่แทนค่าบิทในเลขฐานสอง แสดงในรูปที่ 2.1

ในการส่งข้อมูลนั้น เราสามารถวัดจากความเร็วของการส่ง จากจำนวนบิทที่ส่งไปในหนึ่งหน่วยเวลา โดยทั่วๆไปเราใช้หน่วย บิทต่อวินาที (Bit per seconds) หรือใช้ตัวย่อว่า bps เช่น ความเร็วในการส่ง 110 บิทต่อวินาที หรือ 110 bps นั้นหมายถึงการส่งข้อมูลด้วยอัตรา 110 บิท ในเวลา 1 วินาที

ข่าวสารที่เราส่งในระบบการสื่อสารข้อมูลนั้น อาจจะมีขนาดเท่าใดก็ได้ แต่ในการส่งนั้น จะต้องมีการกำหนดจุดเริ่มต้น และจุดสิ้นสุดของการส่งแต่ละชุด ตามปกติเราจัดแบ่งข่าวสารที่ส่ง ออกเป็นบล็อคของข่าวสาร คือ กลุ่มของบิทจำนวนหนึ่งที่เราส่งออกไปเป็นหน่วยเดียวกัน โดยมีการนำกลุ่มของบิทนั้น ผ่านกระบวนการบางอย่าง เพื่อใช้ในการควบคุมข้อผิดพลาดที่อาจจะเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การแทนอักขระขนาดต่างๆ กันด้วยไบท์

การจำแนกวิธีการส่งข่าวสารตามทิศทางการส่งภายในสาย

สายสื่อสาร (Communication Line) ที่ใช้กันส่วนมาก คือสายโทรศัพท์ (Telephone Line) โดยการส่งข้อมูลออกไปตามสาย ทำให้บางครั้งเรียกสายเหล่านี้ว่าสายข้อมูล (data line) ในการใช้สายสำหรับการส่งข่าวสารนี้ หากพิจารณาตามทิศทางการส่งข้อมูลภายในสายแล้ว สามารถแบ่งการส่งข้อมูลออกได้เป็น 3 ชนิดคือ

1.การส่งแบบทิศทางเดียว (one-way Transmission or Simplex)

2.การส่งแบบทิศทางใดทิศทางหนึ่ง (Either-way Transmission or Half-duplex Transmission)

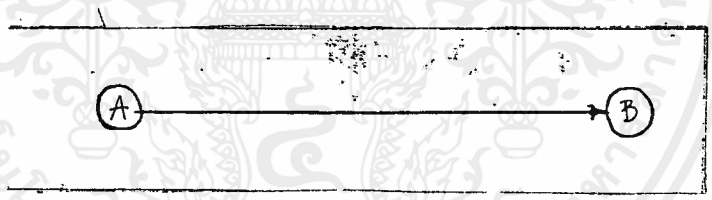
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การส่งข้อมูลแบบสองทิศทาง (Both-way Transmission or Full-duplex Transmission)

1.การส่งแบบทิศทางเดียว (Simplex Transmission)

ตามปกติสายสำหรับส่งข้อมูลนั้น อาจจะประกอบด้วยช่องทางเดิน หรือที่เรียกกันว่า แชนแนล (channel) สำหรับส่ง หนึ่งแชนแนลหรือมากกว่าก็ได้ คำว่าหนึ่งแชนแนลนั้นหมายถึงทางเดิน สำหรับการส่งในทิศทางเดียว (One-Way Transmission) แชนแนลนั้นสามารถที่จะใช้ สำหรับการส่งข่าวสารในทิศทางใดก็ได้ เพียงทิศทางเดียวในเวลาหนึ่งๆ ทิศทางการไหลของข่าวสารนั้นสามารถหาได้ จากคุณสมบัติของอุปกรณ์ ที่ปลายของแต่ละแชนแนล

สำหรับการส่งแบบทิศทางเดียวนั้น หากเราพยายามส่งข้อมูล ไปตามแชนแนลทั้งสองในเวลาเดียวกัน ข่าวสารก็จะเกิดการชนกันขึ้น และอาจจะรวมกัน หรือหักล้างกันกลายเป็นข่าวสารที่ผิดพลาดได้

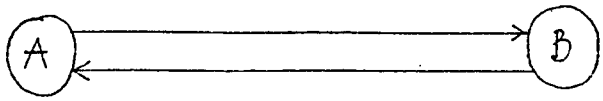


รูปที่ 2.2 ระบบทิศทางเดียว

2.การส่งแบบทิศทางใดทิศทางหนึ่ง (Half-Duplex Transmission)

สำหรับการส่งแบบทิศทางใดทิศทางหนึ่ง (Either-way Transmission) หรือที่เรียกว่า Half-duplex Transmission นั้น เราใช้ทางเดิน หรือแชนแนลการเดินทางของข้อมูลชุดเดียว แต่เปลี่ยน หรือสลับทิศทางการไหลของข้อมูลในแชนแนลให้กลับทิศทางกัน โดยใช้อุปกรณ์ปลายสาย ที่พัฒนาขึ้นมา ให้มีความสามารถ ทั้งการรับและการส่งได้ในเครื่องเดียวกัน วิธีการส่งแบบทิศทางใดทิศทางหนึ่ง แสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การส่งแบบทิศทางใดทิศทางหนึ่ง

จากรูปในตอนแรก A ทำการส่งข่าวสารไปยัง B หลังจากส่งจนหมดแล้ว เราสามารถ เปลี่ยนระบบให้ B ส่งข่าวสารกลับมายัง A ได้ โดยวิธีการนี้ ทำให้เราสามารถส่งข่าวสารระหว่าง A และ B ได้ทั้ง 2 ทิศทาง โดยการสลับกันส่ง

3.การส่งแบบสองทิศทาง (Full-Duplex Transmission)

ในการส่งสัญญาณทั้งสองแบบที่ได้กล่าวมาแล้วนั้น เรามีแชนแนลในการติดต่ออยู่เพียงแชนแนลเดียว ทำให้ไม่สามาถจะส่งสัญญาณ จากทั้งสองด้านของแชนแนลพร้อมๆ กันได้ หากเราเพิ่มแชนแนลของการสื่อสารขึ้นมา เป็น 2 แชนแนล ก็จะทำให้ทั้งสองด้านของแชนแนลสามารถส่งข้อมูลหรือข่าวสารออกมาได้ในเวลาเดียวกัน โดยใช้แชนแนลหนึ่งแชนแนลสำหรับการส่งในทิศทางหนึ่ง และอีกหนึ่งแชลแนล สำหรับการส่งในทิศทางตรงกันข้าม หากอุปกรณ์ปลายทางทั้งสองด้าน มีความสามารถในการรับ หรือส่ง ได้ในเวลาเดียวกันตลอดเวลาก็จะทำให้ระบบนี้สามารถติดต่อกันได้ 2 ทิศทางตลอดเวลา ระบบในลักษณะนี้ เรียกว่าระบบสองทิศทาง ดังรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
032555

รูปที่ 2.4 การส่งแบบ 2 ทิศทาง

การจำแนกวิธีการส่งตามความสัมพันธ์ของข้อมูล

สำหรับวิธีการส่งข้อมูล ที่จำแนกตามความสัมพันธ์ระหว่างข้อมูลที่ส่งนั้น แบ่งออกได้เป็น 2 ชนิด คือ

1)การส่งแบบสัมพันธ์ (Synchronous Transmission)

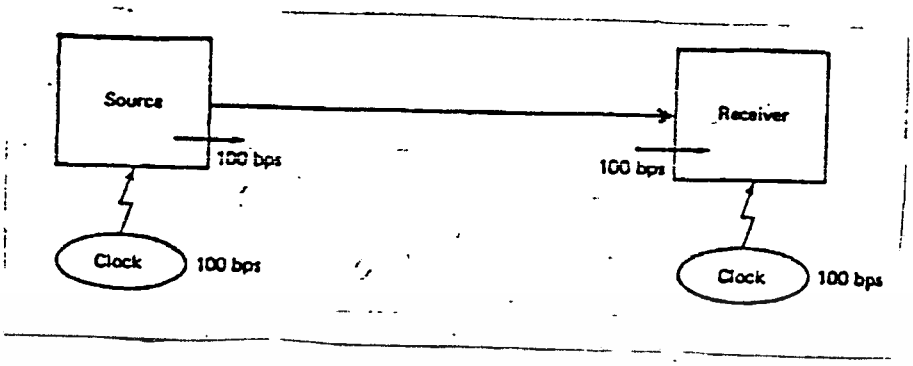
2)การส่งแบบไม่สัมพันธ์ (Asynchronous Transmission)

1)การส่งแบบสัมพันธ์ (Synchronous Transmission)

สำหรับความสัมพันธ์ระหว่างข้อมูลที่เราส่ง ในระบบการสื่อสารข้อมูลนั้น เราจะต้องพิจารณาถึงความสัมพันธ์สองชนิด คือความสัมพันธ์ของบิท ที่ประกอบกันเป็นอักขระหนึ่งตัว และความสัมพันธ์ของอักขระ ที่ประกอบกันเป็นบล็อคความสัมพันธ์ของบิท

ความสัมพันธ์ของบิท หมายถึง ทางด้านรับจะต้องได้รับบิทต่างๆ ที่ทางด้านส่งทำการส่งมาได้อย่างถูกต้อง นั่นหมายถึงว่า ทางด้านรับจะต้องทราบว่าจะรับบิทจากสายส่งนั้นเมื่อใด หลังจากรับตัวแรกมาแล้ว จะรับตัวที่ 2,3 และตัวต่อๆ ไปเมื่อไร ซึ่งสามารถกระทำได้โดยการเพิ่มสัญญาณคล็อค หรือสัญญาณนาฬิกาเข้าไป ที่จุดปลายของระบบทั้งสองด้าน ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การเพิ่ม Clock เพื่อความสัมพันธ์ของบิท

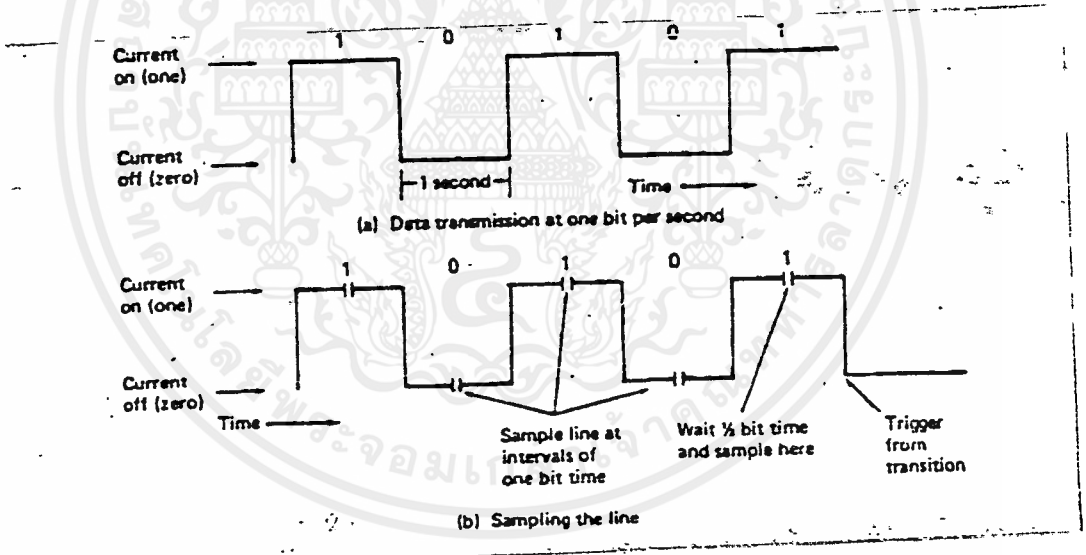
สัญญาณคล็อคของทางด้านส่ง จะเป็นตัวกำหนดเวลาว่า เมื่อใดจะส่งบิทข้อมูลออกไปตามแชนแนล และสัญญาณคล็อคของทางด้านรับจะเป็นตัวกำหนดเวลาว่า จะรับข้อมูลมาจากแชนแนลด้วยอัตราเร็วเท่าใด ในการรับส่งข้อมูลทั่วๆไป คล็อคด้านส่งและด้านรับ จะต้องเท่ากันจึงจะได้ลำดับข้อมูลที่ถูกต้อง

ในการสุ่มตัวอย่างบิทในทางด้านรับนั้น ช่วงเวลาการสุ่มตัวอย่างนี้ ควรจะเป็นช่วงกลางของบิทแต่ละบิท หากสุ่มตัวอย่างในช่วงเวลาที่กำหนดไว้ สำหรับเป็นช่วงการเปลี่ยนแปลงระหว่าง 0 และ 1 นั้นอาจจะทำให้ได้ค่าที่ไม่ถูกต้อง ปกติแล้วการสุ่มตัวอย่างจะอยู่ที่จุดกลางของความกว้างของบิทพอดี ซึ่งสามารถกระทำได้ โดยการใช้จุดของการเปลี่ยนแปลงสัญญาณเป็นจุดอ้างอิงครั้งแรก โดยการรอเวลาผ่านไปเท่ากับ 1/2 ของเวลาของความกว้างหนึ่งบิท แล้วจึงทำการสุ่มตัวอย่าง ณ จุดนั้นๆ หลังจากการสุ่มค่าของบิ๊ทแรกแล้วก็รอเวลาผ่านไป เท่ากับความกว้างของหนึ่งบิท จึงทำการสุ่มค่าของบิทที่ 2 สำหรับบิทที่สามและบิทต่อๆ ไปก็ทำในทำนองเดียวกัน ดังแสดงในรูปที่ 2.10 ซึ่งการสุ่มตัวอย่างค่าบิทแต่ละบิทที่กึ่งกลางของความกว้างของบิท ถ้าหากความเร็วของคล็อค ทั้งทางด้านรับและส่งมีความเร็วเท่ากัน การสุ่มตัวอย่างค่าบิทก็จะกระทำการได้อย่างถูกต้อง ในทางปฏิบัติแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คล็อคของด้านรับและด้านส่งจะเป็นอิสระต่อกัน ค่าความเร็วจึงอาจต่างกันบ้างเล็กน้อย แต่ความเร็วที่แตกต่างกันเพียงเล็กน้อย เช่นประมาณเพียง 0.01% ก็ตาม ความแตกต่างนี้จะเพิ่มขึ้นเรื่อยๆ เพราะสัญญาณนี้จะเกิดต่อกันเป็นวัฏจักร จึงอาจทำให้เกิดความผิดพลาดในการรับสัญญาณของทางด้านรับได้ จำเป็นต้องมีการปรับความเร็วของคล็อคทั้งทางด้านส่งและด้านรับ ให้สัมพันธ์กันที่เรียกว่า รีซิงโครไนซ์(Resynchronization)

ในบางระบบ อาจจะใช้วิธีการส่งสัญญาณคล็อค จากทางด้านตัวส่งไปให้ทางด้านตัวรับ ทางด้านรับจะนำเอาสัญญาณที่รับมานั้น มาเป็นตัวขับการทำงานของคล็อคทางด้านรับเพื่อให้สัญญาณคล็อคของทางด้านรับมีความเร็วเดียวกัน



รูปที่ 2.6 แสดงการสุ่มตัวอย่างการรับส่งอย่างง่ายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ของตัวอักขระ

ในการรับข่าวสารตามสายนั้น แม้ว่าเราจะมีการจัดเกี่ยวกับความสัมพันธ์ของบิทแล้วก็ตาม ยังมีปัญหาที่ตามมาอีกก็คือ ในการส่งนี้เรานำเอาบิทของอักขระหลายๆ ตัวมารวมกันเป็นบล็อค ฉนั้นถึงแม้ว่าบิทต่างๆ จะได้รับมาอย่างถูกต้องแล้วก็ตาม เรายังต้องทราบอีกว่ากลุ่มของบิทที่แสดงถึงตัวอักขระต่างๆ นั้นเริ่มต้นที่บิทใด รูปที่ 2.6 แสดงถึงการส่งชุดของบิทของอักขระในรหัสแบบแอสกี 2 ตัวที่ส่งไปอย่างอนุกรมตามสายสื่อสาร โดยมีข้อมูลขนาด 8 บิทรวม 2 ชุดสำหรับตัวอักษรตัวที่ 1 และตัวที่ 2 ตามลำดับ เมื่อบิทของข้อมูลทั้งสองชุดนี้ถูกส่งไปติดกัน ทางด้านรับเมื่อรับบิทดังกล่าวมาแล้ว จะทราบได้อย่างไรว่าชุดของบิทจำนวน 8 บิทของตัวอักษรแต่ละตัวเริ่มตั้งแต่บิทใด วิธีการแก้ปัญหานี้กระทำได้ถ้าหากว่าเราทราบว่าบิทใดเป็นบิทเริ่มต้นของตัวอักษร และถ้าหากทราบว่า

1) ในตัวอักษรหนึ่งตัวมีกี่บิท

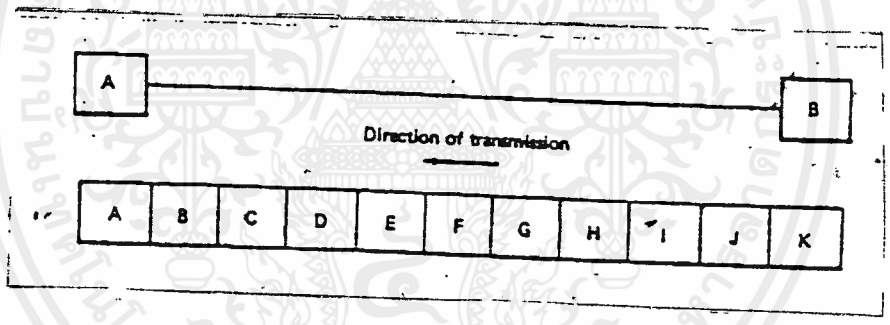
และ 2) ความเร็วของการส่งบิทต่างๆ มาตามสาย โดยการนับจำนวนบิทที่ได้รับมาตามสายหลังจากทราบบิทแรก ก็จะแยกตัวอักษรออกจากกันได้

เทคนิคการส่งแบบ Synchronous นั้น ใช้สำหรับส่งข้อมูลทั้งชุดไปครั้งเดียว ในการส่งแบบนี้นั้นช่วงความกว้าง (เวลา) ระหว่างบิทแต่ละบิทจะมีค่าเท่ากัน และสำหรับระบบที่มีการส่งครั้งละตัวอักษรนั้น ช่วงเวลาระหว่างการสิ้นสุด ของบิทสุดท้าย ของตัวอักษรตัวหนึ่งกับการเริ่มต้นของบิทแรก ของตัวอักษรตัวต่อไปจะมีค่า 0 หรือมีเวลาเท่ากับเวลาทั้งหมดสำหรับการส่งหนึ่งตัวอักษร ทำให้การส่ง มีลักษณะคล้ายกับการส่งข่าวสารในรูปของเลขฐานสองที่มีจำนวนบิทติดต่อกันไป โดยไม่ได้แยกว่าความยาวนั้นเป็นของช่วงตัวอักษรใด ในระบบเช่นนี้บิทแต่ละบิทจะมีความยาวเท่ากัน สำหรับวิธีการในการแยกตัวอักษร จะกล่าวต่อไปในหัวข้อความสัมพันธ์ของข่าวสาร

ในรูป 2.7 แสดงให้เห็นว่าการส่งตัวอักษรต่างๆ ไปแบบ Synchronous โดยตัวอักษรแต่ละตัว มีช่วงเวลาห่างกันเท่ากับศูนย์ ทางด้านรับนั้น เพียงหาว่าบิทแรกของตัวอักษรตัวแรกคือบิทใด และ ทราบขนาดหรือจำนวนบิท ในหนึ่งตัวอักษรพร้อมทั้งความเร็วในการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

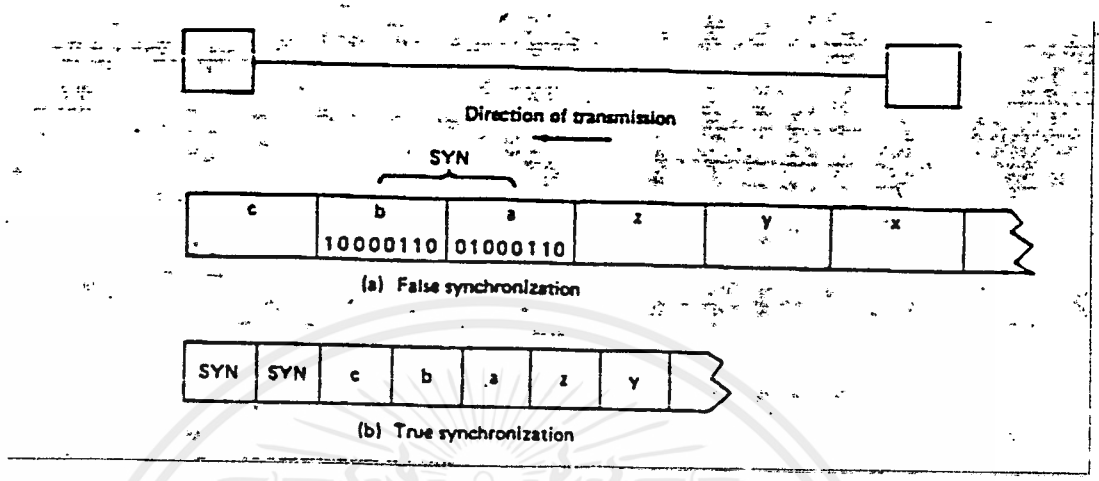
ก็จะสามารถแยกข่าวสารของแต่ละตัวอักษรออกมาได้ ในกรณีที่ข้อมูลเป็นรหัสแบบแอสกี ตัวอักษรแต่ละตัวจะประกอบด้วยบิท 8 บิท หากทราบว่าบิทใดเป็นบิทเริ่มต้นของตัวอักษร โดยใช้วิธีการรับมากลุ่มละ 8 บิท เราก็สามารถแยกตัวอักษรต่างๆได้ เพื่อให้การหาบิทแรกของตัวอักษรตัวแรกเป็นไปอย่างถูกต้อง จึงมักส่งชุดของข้อมูลชุดหนึ่ง ก่อนหน้าการส่งข้อมูลตัวอักษร โดยการส่งตัวอักษรควบคุมความสัมพันธ์ (SYN Transmissiom Control Character : TC) ชุดข่าวสารดังกล่าว ประกอบด้วยบิทขนาด 8 บิท คือ 00010110 (มีพาริตี้คี่) สำหรับทางด้านรับนั้น จะถูกออกแบบมาให้นำบิทที่รับ มาเปรียบเทียบกับชุดของบิทของตัวอักษรควบคุมความสัมพันธ์ (Looking for SYNC) โดยกระทำทุกครั้ง ที่รับบิทใหม่เข้ามา จนกว่าจะได้ชุดของบิทที่ต้องการดังกล่าว



รูปที่ 2.7 การส่งแบบ SYNCHRONOUS

แต่วิธีดังกล่าวนี้ อาจจะเกิดความผิดพลาดได้ ถ้าหากชุดของบิทของตัวอักษร ที่ตามกันมา มีลักษณะของบิท เหมือนกับชุดของบิทของตัวอักษรควบคุมความสัมพันธ์ ทำให้ความสัมพันธ์ที่ได้เกิดผิดพลาดไป กรณีดังกล่าวแสดงในรูป 2.8 (a) เพื่อแก้ไขข้อผิดพลาดดังกล่าวนี้ เราจึงส่งตัวอักษร SYN จำนวน 2 ตัวไปก่อนข้อมูล ดังรูป 2.8 (b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



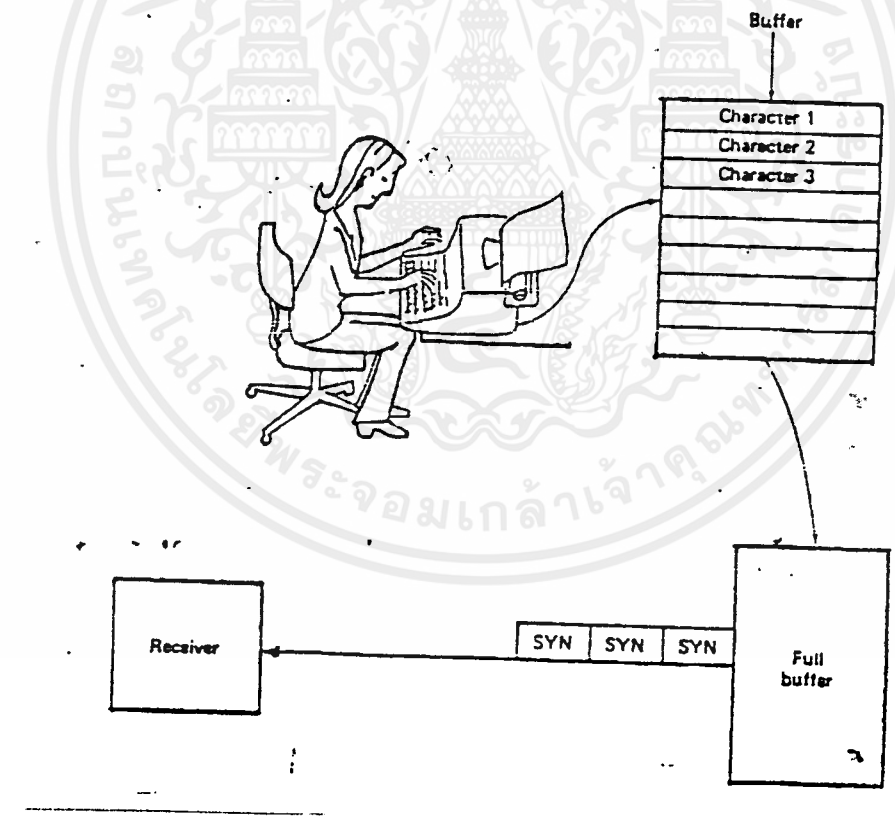
รูปที่ 2.8 การส่งตัวอักษร SYN ไป 2 ตัวเพื่อป้องกันความผิดพลาด

สำหรับทางด้านรับเมื่อรับข่าวสารมา และตรวจสอบได้ว่าเป็นตัวอักษร SYN แล้ว จะต้องนำ ชุดบิท 8 บิท ต่อไปมาตรวจสอบอีกว่า เป็นอักษร SYN หรือไม่ หากชุดที่ 2 เป็นอักษร SYN ก็แสดงว่าชุด 8 บิท ที่นำมาคิดนั้นถูกต้อง ถ้าหากว่า 8 บิท ชุดแรกเป็นอักษร SYN แต่ชุดที่ 2 ไม่ใช่ ก็จะต้องตรวจสอบต่อไป ในระบบส่วนมากมักจะมีตัวอักษร SYN นำหน้าข้อมูล 3-4 ตัว เพื่อให้แน่ใจว่าชุดของบิทสำหรับอักขระที่รับต่อมา คือข้อมูลที่ถูกต้อง

การส่งแบบ Synchronous นี้ส่วนมากแล้วข่าวสารที่จะส่งนั้น มักจะถูกรวบรวม หรือเก็บรักษาได้ในอุปกรณ์ Terminal ก่อนที่จะส่งออกไป ทั้งนี้เพราะในการส่งแบบ Synchronous นั้น อักษรทุกตัว จะต้องถูกส่งออกไปด้วยเวลาคงที่ ซึ่งเป็นการยาก ที่โอเปอร์เรเตอร์ที่ป้อนข่าวสาร จะกระทำได้ และนอกจากนี้ความเร็วในการส่งข้อมูลของสายนั้นมีความเร็ว สูงกว่าการทำงานของโอเปอร์เรเตอร์มาก ฉนั้นอุปกรณ์ Terminal โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั่วไปมักมีส่วนสำหรับเก็บข่าวสารที่เรียกว่า บัฟเฟอร์ (Buffer) สำหรับเก็บรวบรวมข่าวสารที่โอเปอร์เรเตอร์ป้อนเข้าไป แล้วส่งไปด้วยความรวดเร็วการทำงานของอุปกรณ์ Terminal เอง อุปกรณ์เก็บรวบรวมข่าวสารนี้ ก็คืออุปกรณ์ทางดิจิตอลที่เรียกว่า เมมโมรี่ (Memory) นั่นเอง สำหรับการใส่ตัวอักษร SYN นั้น โอเปอร์เรเตอร์ไม่จำเป็นต้องใส่เข้าไป แต่อุปกรณ์ Terminal จะใส่ให้เองก่อนการส่งข้อมูลไปตามสาย สำหรับปลายอีกด้านหนึ่งนั้น ก็มีอุปกรณ์ ที่เรียกว่าบัฟเฟอร์เพื่อรับข้อมูลที่ส่งมาจากทางด้านส่ง วิธีการส่งแบบ Synchronous นี้เราสามารถใช้สายสื่อสารได้อย่างเต็มรูปแบบ รูปที่ 2.9 แสดงการส่งแบบ Synchronous ที่มีบัฟเฟอร์



รูปที่ 2.9 การส่งแบบ Synchronous ที่มีบัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่ง และ การรับ แบบ Synchonous

Synchonous Send

สำหรับรูปแบบการส่งแบบ Synchonous จะมีลักษณะที่คล้ายกับ การส่งแบบ Asynchonous คือมีการส่งสัญญาณเริ่มต้น สัญญาณเช็คข้อมูล สัญญาณปิดท้ายข้อมูล แต่ก็มีข้อแตกต่าง คือ แบบ Synchonous จะเริ่มส่งสัญญาณ โดยมีสัญญาณเริ่มต้น ที่มีขนาดเป็น 1 ไบต์ ข้อมูล โดยรูปแบบของการส่ง ได้อธิบายในหัวข้อข้างต้น การส่งข้อมูล ที่มีขนาดหลายไบต์ ติดต่อกับหลายๆไบต์ จะเป็นบล็อคข้อมูล โดยลักษณะเด่นของการส่งแบบนี้ คือ 1 บล็อค จะประกอบด้วยส่วนของ Head of Block , Number of Block , End of Block ในส่วนของ DSI (DATA SET INTERFACE) จะเป็นส่วนของการสร้างสัญญาณ Head of Block , Number of Block , End of Block ส่วนสัญญาณ Break เป็นสัญญาณที่บอกให้รู้ว่า หยุดการส่ง

Synchonous Send ประกอบด้วยส่วนต่างๆ 7 ส่วน คือ

1)ส่วน control จะเป็นส่วนที่ทำหน้าที่ ควบคุมการส่งทั้งหมด ซึ่งถูกควบคุมโดยไมโครโปรเซสเซอร์อีกทีหนึ่ง ส่วน control นี้ จะทำหน้าที่ส่งสัญญาณมา Enable ทุกส่วนให้ทำงานตามลำดับ

2)ส่วน PISO ทำหน้าที่ แปลงสัญญาณจาก pararell เป็น serial

3)Head of Block เป็นส่วนสร้างสัญญาณ ขนาด 1 byte ที่มีค่าไม่เหมือนกับข้อมูลที่ส่งมา คือมีค่าเป็น 01111110 ซึ่งด้านรับจะทำการตรวจสอบสัญญาณ Head of Block ทุกการเริ่มต้น บล็อคข้อมูล

4)Number of Block หรือ ลำดับของบล็อคข้อมูลในการส่ง เพื่อที่จะให้ฝ่ายรับสามารถตรวจสอบ ลำดับบล็อคข้อมูลได้ถูกต้อง

5)ส่วน End of Block เป็นส่วนของการสร้างสัญญาณปิดท้ายบล็อค โดยทางด้านรับ เมื่อรับสัญญาณ Head of Block มา ก็จะทำการตรวจสอบข้อมูลที่ตามมา แล้วทำการแปลง มาเก็บไว้ในส่วนที่เป็น Buffer หลังจากทำการตรวจสอบข้อมูลขนาด 1 บล็อคแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะทำการ ตรวจสอบสัญญาณ ปิดท้ายบล็อค

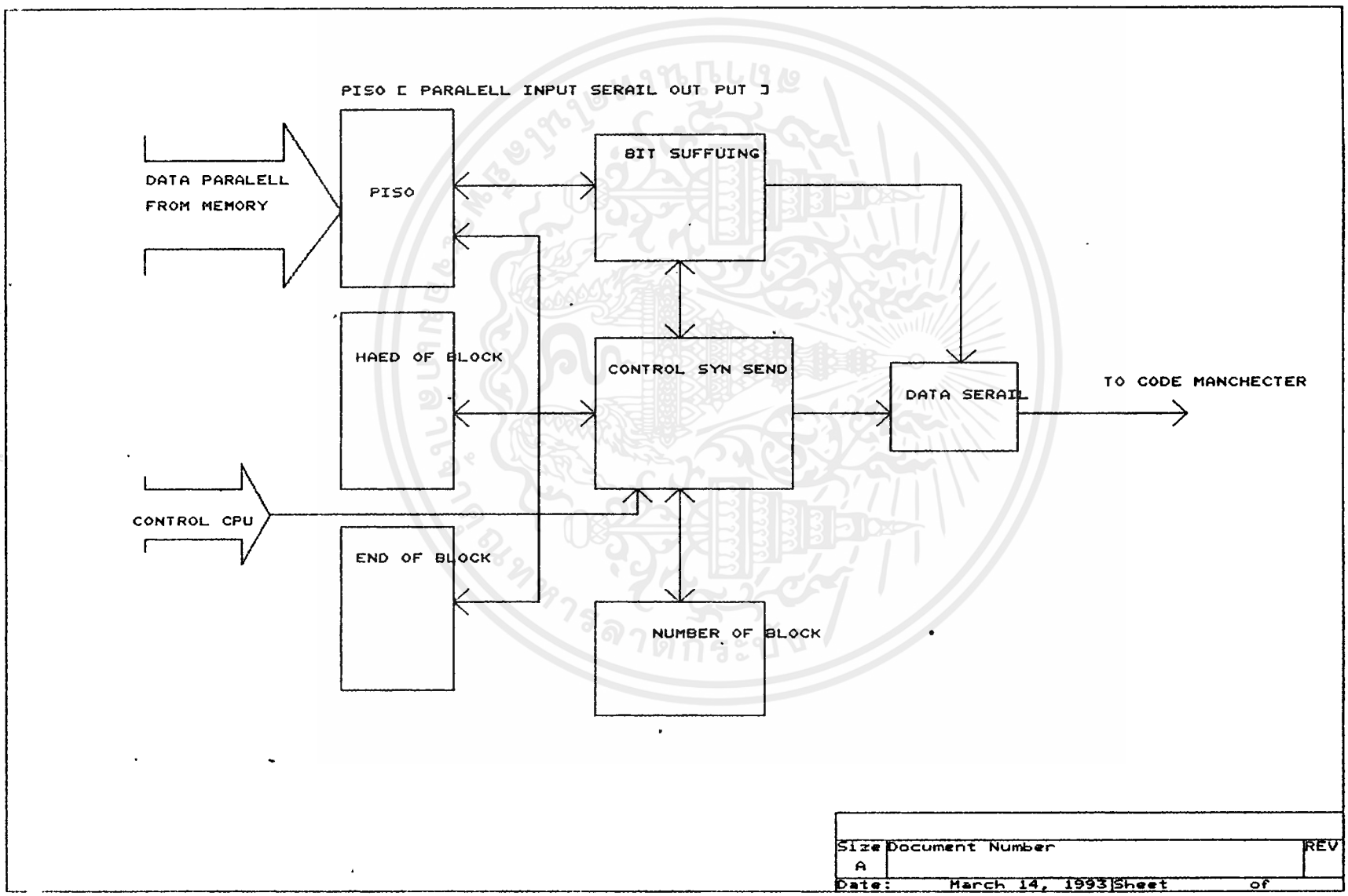
ข้อดีของการมีสัญญาณ End of Block ก็เพื่อทำการบอกฝ่ายรับ ในการส่งว่ามีการตรวจสอบได้ถูกต้อง ซึ่งอาจเกิดจากความผิดพลาดในส่วนการรับการส่งเช่น ในกรณีเครื่องรับเกิดการรับการส่ง ไม่ตรงกัน

6)BIT SUFFING เป็นส่วนที่แปลงสัญญาณข้อมูล ให้มีค่าที่แตกต่างไปจากสัญญาณ HEAD OF BLOCK ทั้งนี้โดยหลักการรับส่ง ตัวรับจะรับสัญญาณ HEAD OF BLOCK ในกรณีที่เริ่มต้นของ BLOCK เท่านั้น ในส่วนที่เป็นข้อมูลจะไม่มีการที่ข้อมูลมาตรงกับสัญญาณ HEAD OF BLOCK ดังนั้น ในกรณีที่สัญญาณข้อมูลมีค่าเป็น 011111X (X คือค่า 0 หรือ 1)ตัว DSI จะทำการเพิ่ม 0 อีก 1 ตำแหน่งเข้าไปซึ่งจะทำให้ข้อมูลที่ได้รับ ไม่เหมือนกับสัญญาณ HEAD OF BLOCK แต่ฝ่ายรับ SYNCHORNOUS RECIVE จะต้องทำการถอดรหัส หรือ การแปลงข้อมูลที่มี 1 ติดกัน 5 ตัว โดยเอา 0 ตัวที่ตามมาออกด้วย

7)ในส่วนของ MANCHESTER เป็นการเข้ารหัสเพื่อทำการส่งโดยจะเป็นส่วนที่เหมาะสมกับฝ่ายรับ

ในส่วนของ SYNCHRONOUS SEND จะประกอบด้วยส่วนต่างๆ ดังในไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


PISO [PARALELL INPUT SERAIL OUT PUT]
DATA PARALELL
FROM MEMORY
PISO
BIT SUFFUING
HAED OF BLOCK
CONTROL SYN SEND
DATA SERAIL
TO CODE MANCHECTER
CONTROL CPU
END OF BLOCK
NUMBER OF BLOCK
Size
A
Document Number
REV
Date:
March 14, 1993
Sheet
of

ส่วน SYNCHRONOUS SEND ประกอบด้วย 7 ส่วน

1)ส่วน CONTROL จะเป็นส่วนที่ทำหน้าที่ ควบคุมการส่งทั้งหมดโดยจะถูกควบคุมโดยไมโครโปรเซสเซอร์ อีกทีหนึ่ง ส่วน CONTROL จะทำหน้าที่ ส่งสัญญาณมา ENABLE ทุกส่วนให้ทำงานตามลำดับ

2)ส่วน PISO ทำหน้าที่ในการแปลงข้อมูลจาก PARURELL เป็น SERIAL

3)ส่วนที่สร้างสัญญาณ HEAD OF BLOCK จะทำการสร้างสัญญาณ 01111110 ซึ่งเป็นสัญญาณ HEAD OF BLOCK สัญญาณนี้จะต้องไม่มีการซ้ำกับสัญญาณของข้อมูลที่ทำการส่งในรูปแบบที่ส่งติดกัน ดังนั้นจะมีสัญญาณของ BIT SUFFING ขึ้นมา

4)NUMBER OF BLOCK จะทำการใส่ NUMBER OF BLOCK หรือ ลำดับของ BLOCK

5)ด้าน END OF BLOCK เป็นส่วนของการสร้างสัญญาณปิดท้าย BLOCK โดย ในภาครับจะตรวจสอบเมื่อครบ 1 BLOCK

6)ส่วนของ BIT SUFFING เป็นส่วนที่เคยอธิบาย คือในรูปแบบการส่งแบบ SYNCHRONOUS จะมีสัญญาณ HEADING OF BLOCK ซึ่งมีค่า 01111110 โดยในการส่งเราจะพบว่าอาจมีสัญญาณ DATA ซึ่งอาจมีการซ้ำกับสัญญาณ HEAD OF BLOCK ได้ ดังนั้นในส่วน BIT SUFFING ในการแปลงสัญญาณข้อมูลโดยถ้าเราพิจารณาสัญญาณ HEAD OF BLOCK จะพบมีสัญญาณ 1 ติดกัน 6 ตัว ดังนั้น ถ้าเราแทรกสัญญาณ 0 ในกรณีที่มี 1 ติดกัน 5 ตัว ตามหลัง 1 5 ตัว จะทำให้สัญญาณที่เป็นข้อมูลไปซ้ำกับสัญญาณ HEAD OF BLOCK แต่ในส่วนรับจะต้องทำการ ตรวจสอบสัญญาณข้อมูลในกรณีที่พบ 1 ติดกัน 5 ตัว จะต้องเอา 0 ตัวที่ตามมาออกข้อมูลที่ได้รับ ก็จะถูกต้องเช่นเดียวกับด้านส่ง

7)ในส่วน MANCHESTER เป็นการเข้ารหัสสัญญาณโดยได้อธิบายในหัวข้อของ MANCHESTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYNCHONOUS RECEIVE

SYNCHRONOUSE RECIEVE เป็นรูปแบบในการรับการส่งแบบ SYNCHRONOUS โดย ในส่วนนี้ประกอบด้วยส่วนที่ทำหน้าที่ในการตรวจสอบ สัญญาณต่างๆ ที่ส่งโดย มีสัญญาณ HEAD OF BLOCK , NUMBER OF BLOCK , END OF BLOCK เช่น CHECK ERROR ส่วนการ CONTROL โดยไมโครโปรเซสเซอร์ และส่วน ANTI BIT SUFFING

อธิบายไดอะแกรม

ส่วน SYNCHORNOUS RECIVE ประกอบด้วยส่วน

1)CHECK HEAD OF BLOCK ทำหน้าที่ในการตรวจหาสัญญาณการเริ่มต้นของ BLOCI

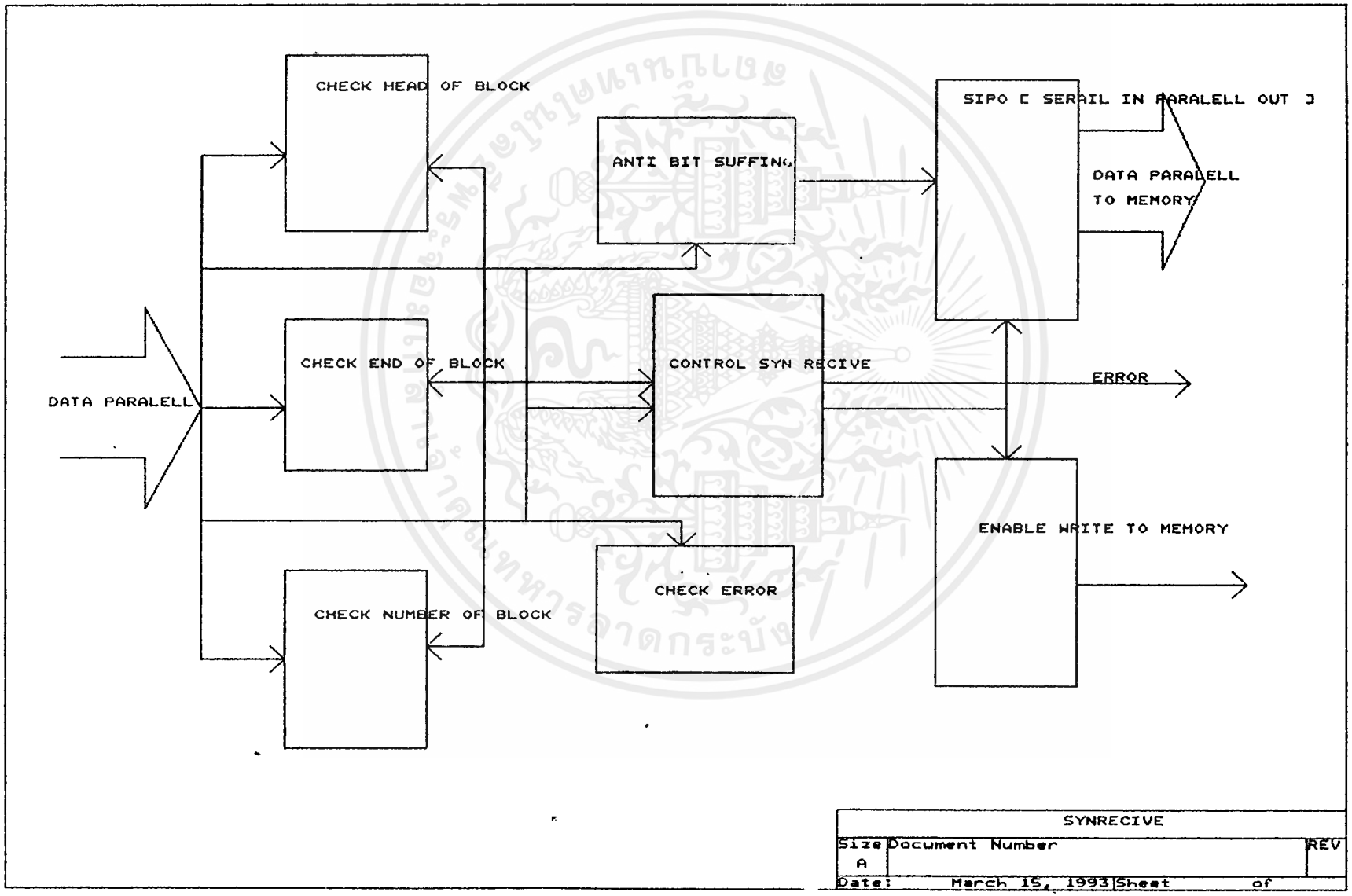
2)CONTROL เป็นส่วนควบคุมการทำงานทุกส่วน โดยควบคุมการทำงานโดยไมโคร โปรเซสเซอร์

3)CHECK ERROR เป็นส่วนตรวจสัญญาณข้อมูล ที่รับมาว่ามีการผิดพลาดหรือไม่โดย ข้อมูลที่ได้ จะต้องไม่มี 1 ติดกัน 6 ตัว

4)END OF BLOCK เป็นส่วน CHECK สัญญาณปิดท้าย BLOCK ข้อมูล เพื่อตรวจ สอบการ SYNCHRONOUSE ระหว่างผู้รับ และผู้ส่ง ในกรณีที่ตรวจสอบไม่พบสัญญาณ END OF BLOCK ก็จะส่งสัญญาณไปที่ส่วน CHECK ERROR ทำการบอกไมโครโปรเซสเซอร์ ทาง PORT 8255 โดยไมโครโปรเซสเซอร์ จะทำการเช็คตลอดการรับข้อมูล ดังนั้น การทำงาน ในส่วนทั้งสอง จำเป็นต้องแยกจากกัน โดยเพิ่มเทคนิคบางประการ ดังจะได้กล่าวในหัวข้อ ABS , DBS (ADDRESS BUFFER SELECT และ DATA BUFFER SELECT)

5)ANTI BIT SUFFING เป็นส่วนถอดรหัส ที่ทำการส่งโดย ใส่ 0 ตามหลัง 1 ติดกัน 5 ตัว โดยเป็นวิธี BIT SUFFING คือทำการเอา 0 ออก หลังจากพบว่า มี 1 ติด กัน 5 ตัว ในส่วนของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

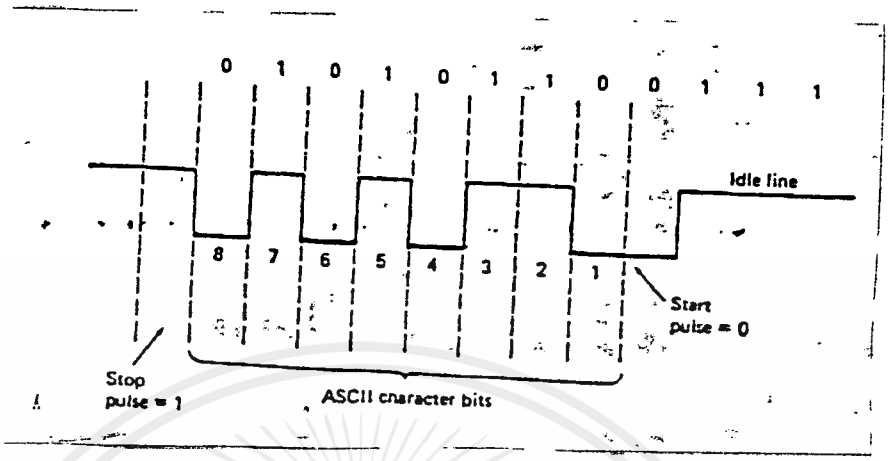
2)การส่งแบบไม่สัมพันธ์ (Asynchronous Transmission)

ในการส่งแบบนี้ ตัวอักขระจะถูกส่งออกไปที่เวลาใดๆ ก็ได้ โดยไม่จำเป็นต้องมีความสัมพันธ์ระหว่างตัวอักษร ว่าจะต้องมีเวลาแน่นอนอย่างไร โดยเราอาจจะส่งตัวอักขระติดต่อกันไป ดังเช่น ในการส่งแบบSynchronous หรืออาจจะเว้นช่วงเวลา ระหว่างการส่งตัวอักขระเป็นเวลาเท่าใดก็ได้ ในกรณีเช่นนี้ ทางด้านรับจำเป็นต้องสร้างความสัมพันธ์ขึ้นใหม่สำหรับอักขระแต่ละตัวที่รับมา ฉนั้น ทางด้านรับจะต้องทราบถึงบิทเริ่มต้นของอักขระแต่ละตัวว่าเริ่มต้นเมื่อใด วิธีดังกล่าวสามารถกระทำได้ โดยการเพิ่มบิทที่เรียกว่า พัลซ์เริ่มต้น (Start Pulse) โดยเติมเข้าไปข้างหน้าชุดของบิทของทุกๆ ตัวอักขระสำหรับบอกให้ทางด้านรับทราบถึงการเริ่มต้นบิทของตัวอักษรใดๆ

ในทางปฏิบัตินั้นด้านรับจะสามารถรู้ได้ว่ามี "พัลซ์เริ่มต้น" หรือยัง โดยการตรวจสอบสถานะของสาย ซึ่งตามปกติแล้วสายจะมีสถานะเป็น 1 หรือ 0 ก็ได้ แต่โดยทั่วๆ ไปแล้วเมื่อไม่มีการส่งข่าวสารนั้น มักจะกำหนดให้สายมีสถานะเป็น 1 บางครั้งเรียกสถานะนี้ว่า MARK โดยสถานะที่มีค่าตรงข้ามกันคือ 0 หรือที่เรียกว่า SPACE

ฉนั้น ตราบเท่าที่ยังไม่มีการส่งข้อมูลหรือข่าวสาร สถานะจะเป็น 1 ติดต่อกันตลอด เมื่อต้องการส่งตัวอักษร เครื่องส่งจะต้องส่งบิทที่บอกถึงการเริ่มต้นที่เรียกว่า บิทเริ่มต้น (Start Bit) หรือการส่งพัลซ์การเริ่มต้น โดยการส่ง 0 ออกไปตามสาย เพื่อบอกให้ทางด้านรับทราบว่า บิทที่ตามมานั้นเป็นค่าบิทของข่าวสาร ลักษณะดังกล่าวนี้แสดงเป็นแผนภูมิทางไฟฟ้าดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การส่งแบบ Asynchronous

ทางด้านส่ง จะส่งบิทเริ่มต้นโดยการเปลี่ยนสถานะของสายจากค่าปกติที่เป็น 1 มาเป็น 0 หลังจากนั้น จึงส่งบิทของข่าวสารสำหรับหนึ่งตัวอักขระออกไป สำหรับทางด้านรับเมื่อทราบถึงค่าเปลี่ยนสถานะจาก 1 ไปยัง 0 ซึ่งแสดงการเริ่มต้น การทำงานจะเริ่มภายหลังจากเวลาผ่านไป เท่ากับครึ่งหนึ่งเวลาของความกว้างขนาด 1 บิท โดยจะสุ่มตัวอย่าง (ตรวจสอบสาย) สายว่า ขณะนั้นมีสถานะ 0 หรือ 1 ถ้าค่าของสายมีค่า 0 ก็แสดงว่าการเปลี่ยนแปลงดังกล่าวเกิดขึ้น เพื่อส่งบิทเริ่มต้น หลังจากนั้นทางด้านรับก็จะทำการสุ่มตัวอย่างสถานะของสายทุกๆ ช่วงเวลา 1 บิท เพื่อหาค่ารหัส ของตัวอักขระที่ส่งมา ในการสุมตัวอย่างครั้งแรกหลังจากที่รับการเปลี่ยนแปลง จาก 1 ไป 0 เป็นเวลา 1 บิทนั้น หากค่าสถานะของสายที่รับมามีค่า 1 ก็แสดงว่าบิทที่รับมาไม่ใช่บิทเริ่มต้น อาจเป็นเพียงสัญญาณรบกวนในช่วงเวลาสั้นๆ ที่เรียกว่า สัญญาณรบกวนอิมพัลซ์ (Noise Impulse) ในกรณีนี้ ทางด้านรับก็จะทิ้งสัญญาณนี้ไป การตรวจสอบบิทเริ่มต้นนี้ กระทำทุกครั้งที่มีอักขระตัวใหม่เข้ามา ฉนั้น เราจึงสามารถ เว้นระยะระหว่างตัวอักขระให้มีเวลาเท่าใดก็ได้ และเมื่อการส่ง ส่งครบบิทของตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรแล้ว จะต้องมีบิทสำหรับบอกถึงการสิ้นสุดที่เรียกว่า พัลซ์การสิ้นสุด (Stop Pulse) ส่งมา เพื่อให้ทางด้านรับมีเวลา สำหรับการเตรียมรับข้อมูลของตัวอักษรตัวต่อไป ดังแสดงในรูป 2.10 จะพบว่าบิทของข้อมูลนั้น มีบิทเริ่มต้นนำหน้า และเมื่อจบบิทของข้อมูลแล้วจะมีบิทหยุดตามมา ด้วยวิธีที่ การส่งบิทเริ่มต้นก่อนและจบด้วยบิทหยุด บางครั้งจึงเรียกระบบนี้ว่า ระบบส่งแบบ เริ่ม-หยุด (Start stop Transmission)

สำหรับการส่งแบบไม่สัมพันธ์นี้ เนื่องจากอักขระทุกตัวที่ส่งออกมา ต่างเป็นอิสระต่อกัน และทุกๆ อักขระจะมีบิทเริ่มต้นและบิทสิ้นสุดอย่างน้อยอักขระละ 1 บิท ฉนั้น หากเกิดสัญญาณรบกวนจะทำให้เกิดความผิดพลาดในข้อมูล ความผิดพลาดที่เกิดขึ้นนี้ จะเกิดขึ้นกับข้อมูลของอักขระเพียงตัวเดียว เพราะอักขระแต่ละตัวมีความสัมพันธ์ กับบิทเริ่มต้น และบิทสิ้นสุดของตัวมันเองเท่านั้น ความผิดพลาดเช่นนี้ ถ้าเกิดขึ้นกับข่าวสารในการส่งแบบ Synchronous อาจทำลายข่าวสารทั้งบล็อคได้ เพราะความสัมพันธ์จะเสียทั้งบล็อค

ทางด้านส่ง จะส่งบิทเริ่มต้น โดยการเปลี่ยนสถานะของสายจากค่าปกติที่เป็น 1 มาเป็น 0 หลังจากนั้นจึงส่งบิทของข่าวสาร สำหรับหนึ่งตัวอักขระออกไป สำหรับทางด้านรับเมื่อทราบถึงการเปลี่ยนสถานะจาก 1 ไปยัง 0 ซึ่งแสดงการเริ่มต้นการทำงาน จะเริ่มภายหลังเวลาผ่านไป เท่ากับครึ่งหนึ่งเวลาของความกว้างขนาด 1 บิท โดยจะสุ่มตัวอย่าง (ตรวจสอบสาย) สายว่าขณะนั้น มีสถานะ 0 หรือ 1 ถ้าค่าของสายมีค่า 0 ก็แสดงว่าการเปลี่ยนแปลงดังกล่าวเกิดขึ้นเพื่อส่งบิทเริ่มต้น หลังจากนั้น ทางด้านรับก็จะทำการสุ่มตัวอย่างสถานะของสายทุกๆ ช่วงเวลา 1 บิท เพื่อหาค่ารหัส ของตัวอักขระที่ส่งมา ในการสุ่มตัวอย่างครั้งแรกหลังจากที่รับการเปลี่ยนแปลง จาก 1 ไป 0 เป็นเวลา 1 บิทนั้น หากค่าสถานะของสายที่รับมามีค่า 1 ก็แสดงว่าบิทที่รับมาไม่ใช่บิทเริ่มต้น อาจเป็นเพียงสัญญาณรบกวนในช่วงเวลาสั้นๆ ที่เรียกว่า สัญญาณรบกวนอิมพัลซ์(Noise Impulse) ในกรณีนี้ทางด้านรับก็จะทิ้งสัญญาณนี้ไป การตรวจสอบบิทเริ่มต้นนี้กระทำทุกครั้ง ที่มีอักขระตัวใหม่เข้ามา ฉนั้น เราจึงสามารถเว้นระยะระหว่างตัวอักขระให้มีเวลาเท่าใดก็ได้ และเมื่อการส่ง ส่งครบบิทของตัวอักษรแล้ว จะต้องมีบิทสำหรับบอกถึงการสิ้นสุด ที่เรียกว่า พัลซ์การสิ้นสุด(Stop Pulse)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งมา เพื่อให้ทางด้านรับมีเวลา สำหรับการเตรียมรับข้อมูลของตัวอักษรตัวต่อไป จะพบว่า บิทของข้อมูลนั้น มีบิทเริ่มต้นน่าหน้า และเมื่อจบบิทของข้อมูลแล้วจะมีบิทหยุดตามมา ด้วยวิธีการที่การส่งบิทเริ่มต้นก่อน และจบด้วยบิทหยุด บางครั้งจึงเรียกระบบนี้ว่า ระบบส่งแบบ เริ่ม-หยุ (Start stop Transmission)

สำหรับการส่งแบบไม่สัมพันธ์นี้ เนื่องจากอักขระทุกตัว ที่ส่งออกมาต่างเป็นอิสระต่อกัน และทุกๆ อักขระจะมีบิทเริ่มต้น และบิทสิ้นสุดอย่างน้อยอักขระละ 1 บิท ฉนั้น หากเกิดสัญญาณรบกวน ทำให้เกิดความผิดพลาดในข้อมูล ความผิดพลาดที่เกิดขึ้นนี้ จะเกิดขึ้นกับข้อมูลของอักขระเพียงตัวเดียว เพราะอักขระแต่ละตัว มีความสัมพันธ์กับบิทเริ่มต้น และบิทสิ้นสุดของตัวมันเองเท่านั้น ความผิดพลาดเช่นนี้ ถ้าเกิดขึ้นกับข่าวสารในการส่งแบบ Synchronous อาจทำลายข่าวสารทั้งบล็อคได้ เพราะความสัมพันธ์จะเสียทั้งบล็อค

อีกประการหนึ่ง โมเด็มและอุปกรณ์ที่เกี่ยวข้องกับการส่งแบบ Synchronous มีราคาแพงกว่าอุปกรณ์ที่ใช้กับการส่งแบบ Asynchronous

อาจสรุปถึงความแตกต่างระหว่างการส่งแบบ Synchronous กับแบบ Asynchronous ว่า สำหรับการส่งแบบ Asynchronous นั้นอักขระ แต่ละตัวจะถูกส่งไปอย่างอิสระ ไม่ขึ้นกับอักขระตัวอื่นๆ โดยมีบิทเริ่มต้น และบิทสิ้นสุดประกอบอยู่เป็นของตัวเอง สำหรับบอกเครื่องรับให้ทราบถึงจุดเริ่มต้น และจุดสิ้นสุด แต่ในระบบการส่งแบบ Synchronous นั้น ข้อมูลจะถูกส่งไปเป็นหน่วยหนึ่ง หลังจากปรับเครื่องรับ และเครื่องส่งให้ทำงานสัมสันพันธ์กันอย่างถูกต้องแล้ว จึงทำการส่งข้อมูลบล็อคนั้นๆ

ASYNCHONOUS SEND

ในส่วนของการส่งข้อมูลแบบ ASYNCHORNOUS มีรูปแบบดังอธิบายในส่วน---โดยหลักการจะมีส่วนสำคัญ คือ มีส่วนสัญญาณเริ่มต้น (start bit) ข้อมูลขนาด 1 byte ตามด้วยส่วน bit parity และส่วนสุดท้ายตามด้วยส่วนปิดท้ายข้อมูล โดยจะทำการสร้างส่วนเริ่มต้น (start bit) , parity bit และ stop bit ทุก 1 byte ข้อมูล

ในที่นี้ ในส่วนของ ASYNCHORNOUS SEND จะมีรูปแบบในการเลือกส่งโดยจะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกส่งแบบ parity คู่ (EVEN) parity คี่ (ODD) หรือ non parity และสามารถเลือกจำนวน stop bit ว่ามี 1 หรือ 2 stop bit ได้ด้วย สำหรับการทำงานในแต่ละส่วนโดยอาศัยไดอะแกรม

ในส่วนของ PISO (Parrarell Input Serial Output) จะทำการแปลงสัญญาณ จาก Parrarell เป็น Serial โดยจะมีส่วน Control ทำการควบคุม ในส่วนของ check parity จะทำการ check ข้อมูลที่จะส่งโดยในส่วนนี้สามารถที่จะ Control โดยการเลือก Parity ว่าจะเป็นแบบใด ในส่วนของการสร้างสัญญาณ start bit stop bit ก็จะทำการสร้างสัญญาณ start bit stop bit

ในส่วน Control จะทำหน้าที่ในการควบคุมการส่งและรับสัญญาณ ในการเลือกรูปแบบการสร้างสัญญาณ จะ load ข้อมูลในการส่งและหยุดการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASYNCHONOUS RECEIVE

ส่วน Asynchzonous Recive จะเป็นส่วนรับข้อมูลโดยจะแปลงข้อมูลจาก Serial เป็น Pararell โดยเริ่มต้น จะทำการวน check start bit เมื่อพบ start bit ก็จะทำการแปลง data ขนาด 1 bit หลังจากนั้นจะทำการตรวจสอบ Parity bit และ stop bit ถ้าถูกต้องก็จะส่งสัญญาณมาเขียนลงใน memory อธิบายตาม Block Diagram

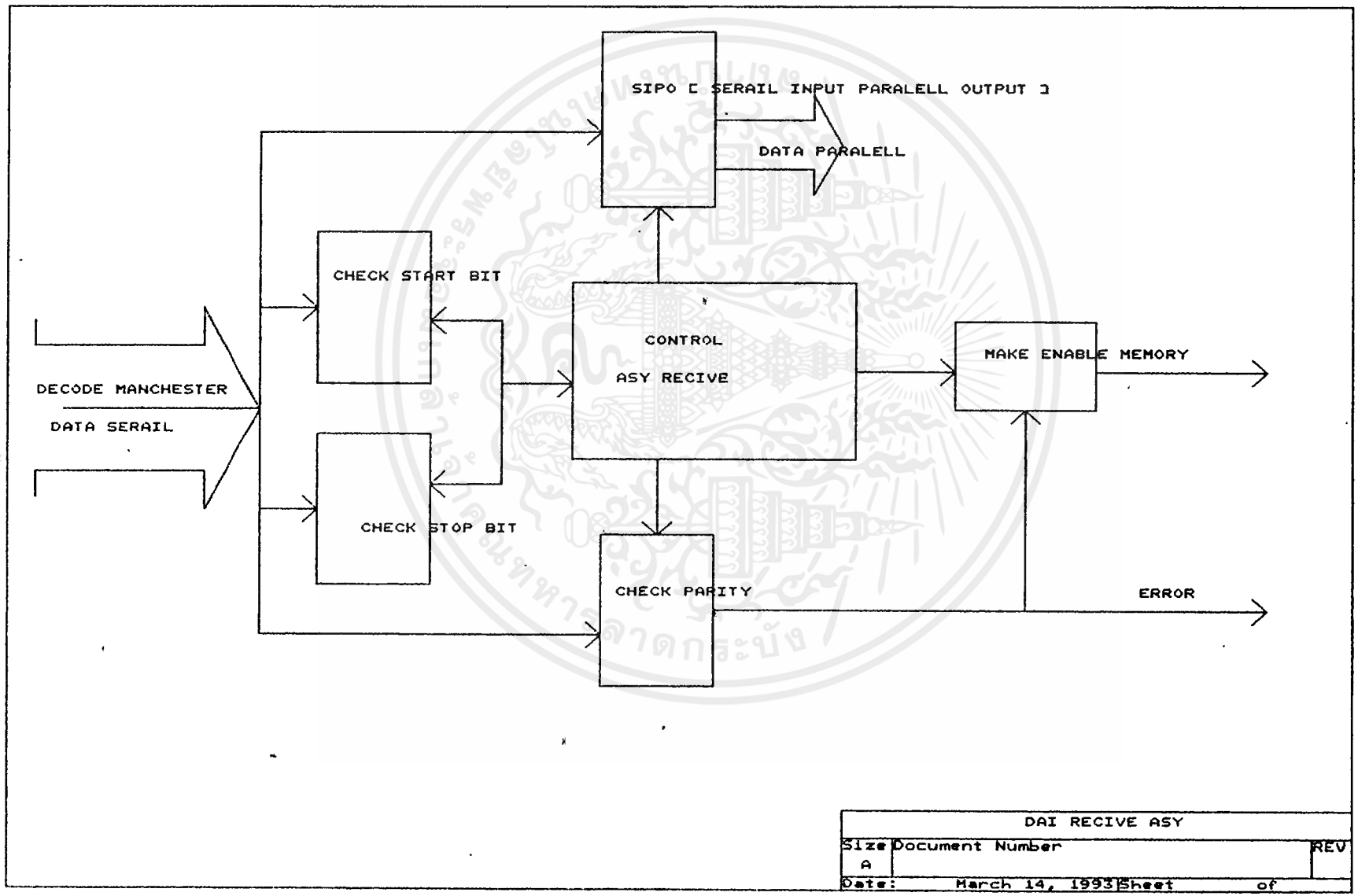
Block Diagram

ประกอบด้วยส่วน Control ส่วน check start bit ส่วน check stop bit และส่วนเช็ค parity โดยส่วน start bit จะวน check จนพบสัญญาณ start bit หลังจากพบ start bit แล้วก็จะส่งสัญญาณบอกส่วน Control ส่วน Contorl จึงจะทำการส่งสัญญาณไปบอกให้ส่วน SIPO ทำการแปลงข้อมูลจาก Serial เป็น Pararell หลังจากครบ 8 bit ก็จะตรวจส่วน parity bit และทำการตรวจสอบ stop bit

ในส่วนของ Control จะประกอบด้วยการทำงาน ในการส่งสัญญาณมาควบคุมส่วน check parity ส่วน stop bit , ส่วน start bit และรับค่าสิ่งในการตรวจสอบว่าจะเป็นแบบใด สามารถควบคุมได้โดยไมโครโปรเซสเซอร์

ในการตรวจสอบสัญญาณ (Deteat Signal) นั้นจำเป็นอย่างยิ่งที่จะต้องทำการตรวจสอบในช่วงกลาง bit ของข้อมูลทั้งนี้ เพื่อป้องกันการผิดพลาดในการเปลี่ยนสถานะ ดังนั้นในวงจรจริงจะต้องสร้างหรือเปลี่ยนแปลง clock ที่รับมา เพื่อเหมาะสมในการ Detect สัญญาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


SIPO [SERAIL INPUT PARALELL OUTPUT]
DATA PARALELL
CHECK START BIT
CONTROL
ASY RECIVE
MAKE ENABLE MEMORY
DECODE MANCHESTER
DATA SERAIL
CHECK STOP BIT
CHECK PARITY
ERROR
DAI RECIVE ASY
Size A
Document Number
REV
Date: March 14, 1993
Sheet of

เส้นใยแสง

เส้นใยแสงคืออะไร

เส้นใยแสง คือเส้นใยขนาดเล็กที่ทำหน้าที่เป็นตัวนำแสง โครงสร้างของเส้นใยแสงประกอบด้วย ส่วนที่แสงเดินทางผ่านเรียกว่า CORE และส่วนที่หุ้ม CORE อยู่เรียกว่า CLAD ทั้ง CORE และ CLAD เป็น DIELECTRIC ใส 2 ชนิด (DIELECTRIC หมายถึง สารที่ไม่เป็นตัวนำไฟฟ้า เช่น แก้ว พลาสติก) โดยการทำให้ค่าดัชนีการหักเหของ CLAD มีค่าน้อยกว่าค่าดัชนีการหักเหของ CORE เล็กน้อยประมาณ 0.2-3% และอาศัยปรากฏการณ์สะท้อนกลับหมดของแสง สามารถทำให้แสงที่ป้อนเข้าไปใน CORE เดินทางไปได้ นอกจากนั้นเนื่องจากกล่าวกันว่าเส้นใยแสงมีขนาดเล็กมาก ขนาดเท่าเส้นผมนั้น หมายถึงขนาดเส้นผ่าศูนย์กลางด้านนอกของ CLAD ซึ่งมีขนาดประมาณ 0.1 ม.ม ส่วน CORE ที่แสงเดินทางผ่านนั้นมีขนาดเล็กลงไปอีก คือประมาณหลายไมครอน ซึ่งมีค่าเป็นหลายสิบเท่าของความยาวคลื่นของแสงที่ใช้งาน ค่าต่างๆ เหล่านี้เป็นค่าที่กำหนดขึ้น จากคุณสมบัติการส่ง และคุณสมบัติทางเมคานิกส์ที่ต้องการ เส้นใยแสงนอกจากมีคุณสมบัติการส่งดีเยี่ยมแล้ว ยังมีลักษณะเด่นอย่างอื่นอีก เช่น ขนาดเล็ก และน้ำหนักเบา

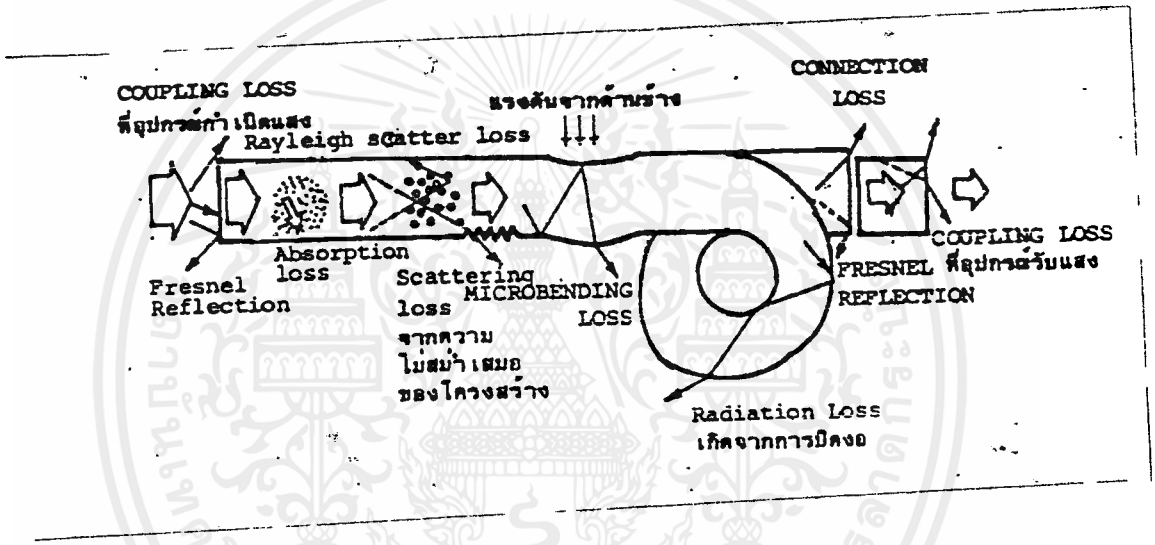
คุณสมบัติต่างๆ ของเส้นใยแสง

1. การสูญเสียแสงในเส้นใยแสง

ในระบบการสื่อสารด้วยเส้นใยแสง มีองค์ประกอบสำคัญ 2 อัน ที่จะกำหนดความเร็วในการส่ง (TRANSMISSION SPEED) และระยะห่างของการถ่ายทอดสัญญาณ (REPEATE SPAN) องค์ประกอบเหล่านั้น คือการสูญเสียแสง กับ BANDWIDTH ของการส่ง (TRANSMISSI BAND WIDTH) การสูญเสียแสง (Optical Loss) เป็นตัวบอกให้ทราบว่า กำลังของแสงที่เดินทางไปในเส้นใยแสงนั้นจะลดลงไปจากเดิมเท่าไร ซึ่งการสูญเสียแสงนี้ ยิ่งมีค่าน้อยเท่าไร จะทำให้สามารถส่งสัญญาณแสงได้ไกลมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสูญเสียแสง แบ่งออกตามความแตกต่างของโครงสร้างที่เกิดขึ้นได้เป็นการสูญเสียที่มีอยู่ในตัวของเส้นใยแสงเอง (FIXED LOSS) และการสูญเสียที่เพิ่มขึ้นมา (Additio Loss) ตอนนำเส้นใยแสงไปใช้ในระบบการสื่อสาร สาเหตุที่ทำให้เกิดการสูญเสียแสงแสดงดังรูป



รูปที่ 1. แสดงสาเหตุต่าง ๆ ที่ทำให้เกิดการสูญเสียแสง

จากรูปที่ 1. จะเห็นว่าการสูญเสีย ที่มีอยู่ในตัวของเส้นใยแสงเองนั้น เกิดจากสาเหตุต่างๆ ได้แก่ การดูดแสง (Absorption Loss) , การกระจัดกระจายแสงแบบเรลี่ (Rayleigh Scattering Loss) การกระจัดกระจายแสงเนื่องจากความไม่สม่ำเสมอของโครงสร้าง (Ununiformity Loss) ส่วนการสูญเสียที่เพิ่มขึ้นมานั้น เกิดจากสาเหตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ ได้แก่ การงอโค้งของเส้นใยแสง (Bending Loss) การงอโค้งแบบ Microbendin (microbending Loss) การต่อเส้นใยแสง (Connection Loss ได้แก่ การสูญเสียจากการสะท้อนกลับของแสง , จากการที่แสงกระจายแสง) , การเชื่อมต่อแสงระหว่างอุปกรณ์ต้นกำเนิดแสง หรืออุปกรณ์รับแสง กับเส้นใยแสง (COUPLING LOSS)

2. การขยายกว้างออก (Dispersion) และ Transmission Band Width ของเส้นใยแสง

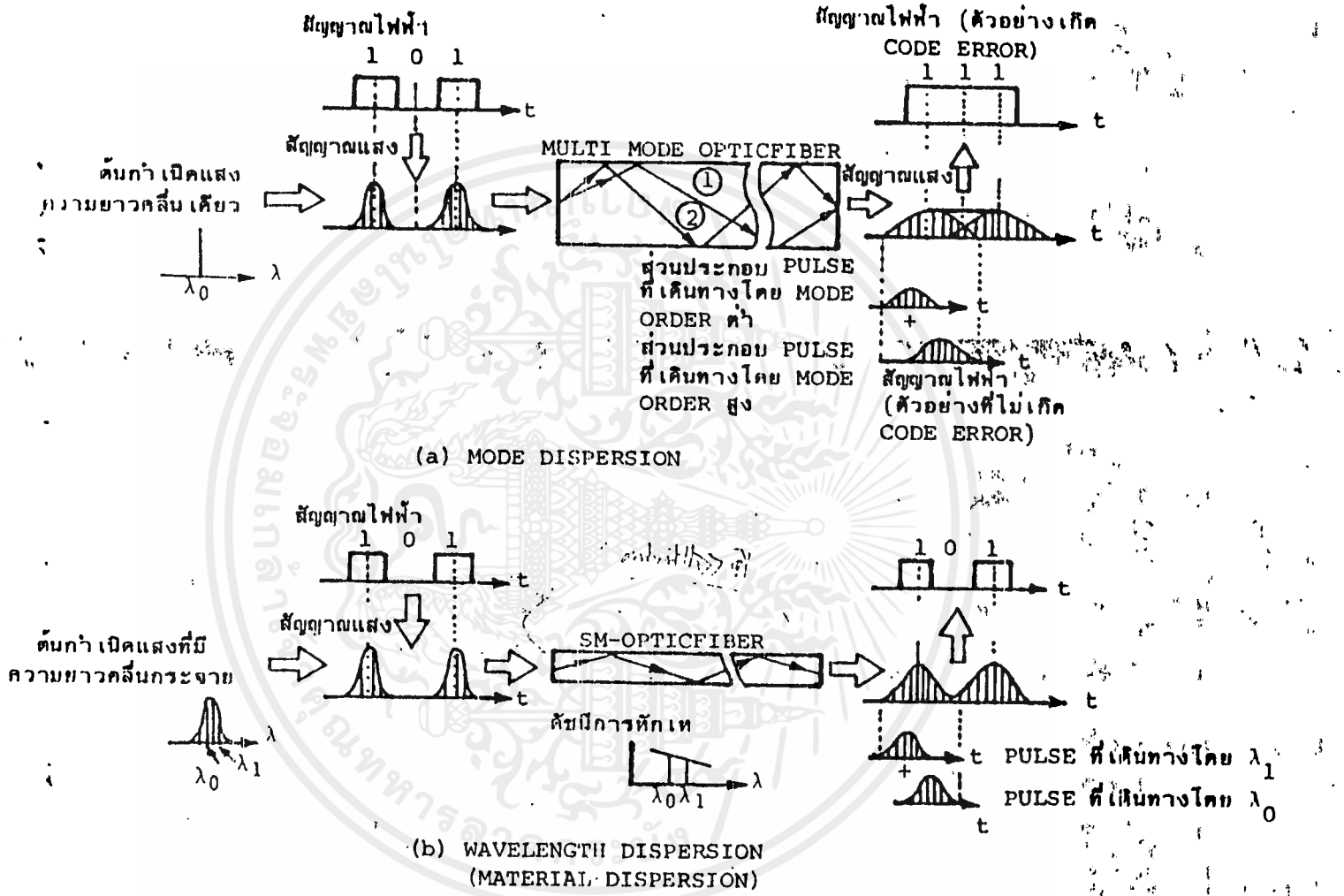
(1) การขยายกว้างออกของ MODE (MODE DISPERSION)

สำหรับ MULTI MODE OPTIC FIBER นั้น ความเร็วในการเดินทางของแต่ละ MODE ภายในเส้นใยแสงจะต่างกัน ดังตัวอย่างที่แสดงในรูป 2.(a) เมื่อป้อน pulse ที่ถูกผสมแบบ Pulse Modulation เข้าไปในเส้นใยแสงแบบ Multi Mode optic Fiber Pulse ของแสงที่ออกมาจากปลายอีกข้างหนึ่งของเส้นใยแสง เมื่อนำไปเปรียบเทียบกับ Pulse ของแสง ทางด้านที่ป้อนเข้าไปจะเห็นว่า ขยายออกกว้างกว่า การที่เกิดอย่างนี้นั้น แม้ว่าความยาวคลื่นของต้นกำเนิดแสง จะเหมือนกันก็ตาม แต่ทว่า Pulse ของแสงที่ป้อนเข้าไปนั้น จะแบ่งออกเป็น MODE ต่างๆ ที่มีความเร็วในการเดินทางต่างกัน ดังนั้น ในการสื่อสารด้วยเส้นใยแสง จึงเรียกปรากฏการณ์นี้ว่า MODE DISPERSION ทำให้ไม่สามารถทำให้ระยะห่างระหว่าง pulse เล็กลงมากได้

ดังนั้นจะเห็นได้ว่า MULTI MODE OPTIC FIBER นั้นส่วนใหญ่ MODE DISPERSION จะเป็นตัวกำหนด BAND WIDTH แม้ว่าจะพยายามทำให้ MODE DISPERSION นี้น้อยลงโดยการทำให้ค่าดัชนีการหักเหของ CORE เป็นลักษณะ PARABOLA ซึ่งเรียกว่า GRADED INDEX OPTIC FIBER แล้วก็ตาม แต่เนื่องจากมีขีดจำกัด ในความแน่นอนของรูปร่างของค่าดัชนีการหักเห ด้วยเหตุนี้ในระบบการส่งข่าวสารจำนวนมาก ที่ต้องการคุณสมบัติ BAND WIDTH กว้างนั้นจะใช้ SINGLE MODE OPTIC FIBER ที่ไม่มี MODE DISPERSION โดยการทำให้มี MODE เดียว

(2) การขยายกว้างออกของความยาวคลื่น (WAVE LENGTH DISPERSION)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 5.16 แสดงการเกิด DISPERSION ของเส้นใยแสง

ในตัวกลางที่มีเนื้อเดียวกัน แต่ถ้าหากความยาวคลื่นต่างกันแล้ว ค่าดัชนีการหักเหจะต่างกัน เป็นผลให้ความเร็วในการเดินทางต่างกัน ตามค่าความยาวคลื่น ปรากฏการณ์ที่เห็นได้ เกี่ยวกับค่าดัชนีการหักเหที่ขึ้นอยู่กับความยาวของแสง คือการกระจายแสงออกเป็นสีต่างๆ ของ PRISM , การเกิดรุ้งกินน้ำที่มีสี 7 สีเกิดขึ้น ในตอนหลังจากฝนตก เป็นต้น แสงที่ใช้ในการสื่อสารใยแสงนั้น ความจริงแล้วไม่ใช่แสงที่มีความยาวคลื่นเพียงอันเดียวเท่านั้น ดังแสดงในรูป 2.(b) กล่าวคือ จะมีความยาวคลื่นอื่นอยู่ด้วยแต่น้อยมาก ทำให้เกิดการขยายกว้างออกของความยาวคลื่น ด้วยเหตุนี้จะเกิดความแตกต่างของความเร็วในการเดินทางตามความยาวคลื่นที่ต่างกัน (ความยาวคลื่นยิ่งมีค่ามากเท่าใด ค่าดัชนีการหักเหจะน้อยลงเท่านั้น และจะเป็นผลให้ความเร็วในการเดินทางเพิ่มขึ้น) การขยายกว้างออกของความยาวคลื่นนี้เป็นสาเหตุให้ BAND WIDTH ถูกจำกัดค่าความกว้าง เช่นเดียวกันกับกรณีของ MODE DISPERSION ดังที่ได้กล่าวมาแล้วในข้อ (1) และสาเหตุนี้เรียกว่า MATERIAL DISPERSION

นอกจากนั้นกรณีที่ค่าผลต่างของดัชนีการหักเหของ CORE กับ CLAD มีค่าน้อยนั้น ปรากฏการณ์การสะท้อนกลับหมดของแสงที่รอยต่อของ CORE กับ CLAD จะไม่เหมือนกับที่เกิดในกระจกเงากล่าวคือ ส่วนหนึ่งของแสงจะทะลุเข้าไปใน CLAD ก่อน แล้วจึงสะท้อนกลับหมด และ ขนาดการทะลุเข้าไปนี้จะแตกต่างกันตามความยาวคลื่น เป็นผลให้ระยะทางการเดินทางของแสงนี้ ขึ้นอยู่กับความยาวคลื่น (ความยาวคลื่นยิ่งมีค่ามากเท่าใดขนาดการทะลุเข้าไปใน CLAD จะมากขึ้นเท่านั้น และจะทำให้เส้นทางการเดินทางยาวมากขึ้น) การขยายกว้างออกของระยะทางการเดินทางของแสงนี้ เรียกว่า STRUCTURE DISPERSION ในการสื่อสารด้วยเส้นใยแสงนั้นการเกิด MATERIAL DISPERSION และ STRUCTURE DISPERSION รวมเรียกว่า WAVE LENGTH DISPERSION

(3) ขนาดความกว้าง BANDWIDTH ของการส่ง (TRANSMISSION BANDWIDTH)

ปัจจุบัน TRANSMISSION BAND WIDTH ที่ใช้ในการสื่อสารด้วยเส้นใยแสงนั้น เป็นตัวบอกให้ทราบว่า จะสามารถส่งสัญญาณแสง ที่ผสมกับสัญญาณของข่าวสารด้วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถี่สูงมากแค่ไหน ปกติจะแสดงขนาดความกว้าง ตรงจุดที่คุณสมบัติความถี่ base band (Base Band Frequency Characteristic) ของเส้นใยแสงตกลง 6 dB

สมมุติว่าเราทำการป้อน IMPULSE ซึ่งเป็น WAVE FORM ที่สมบูรณ์แบบกล่าว คือ ตรงฐานของ WAVE FORM ไม่ขยายกว้างออก เมื่อ IMPULSE นี้เดินทางไปในเส้นใยแสงได้ระยะทางค่าหนึ่ง ตรงจุดปลายทางของเส้นใยแสงนั้น IMPULSE ที่เดินทางมาจะขยายกว้างออก เนื่องจากคุณสมบัติ dispersion ของแสง ตามที่ได้กล่าวมาแล้วในหัวข้อที่ผ่านมา แม้ว่าในระหว่างทางที่แสงเดินทางมาจะไม่มี RADIATION MODE (แสงกระจายออกจาก CORE) เลยก็ตาม แต่ทว่าการที่ IMPULSE ขยายกว้างออกไปนั้น จะต้องทำให้ขนาดของ IMPULSE ลดลงอย่างแน่นอน จากการตรวจดู IMPULSE WAVE FORM ตรงจุดปลายทางของเส้นใยแสง (เรียกว่า IMPULSE RESPONNSE WAVE FORM) นี้สามารถใช้เป็นขั้นตอนของการประเมินค่า คุณสมบัติการขยายกว้างออก (DISPERSION) ของเส้นใยแสงได้ โปรดสังเกตุว่าการใช้หลักการ เช่นนี้ในย่านของความถี่จึงเรียกว่า BASEBAND FREQUENCY CHARACTERISTIC

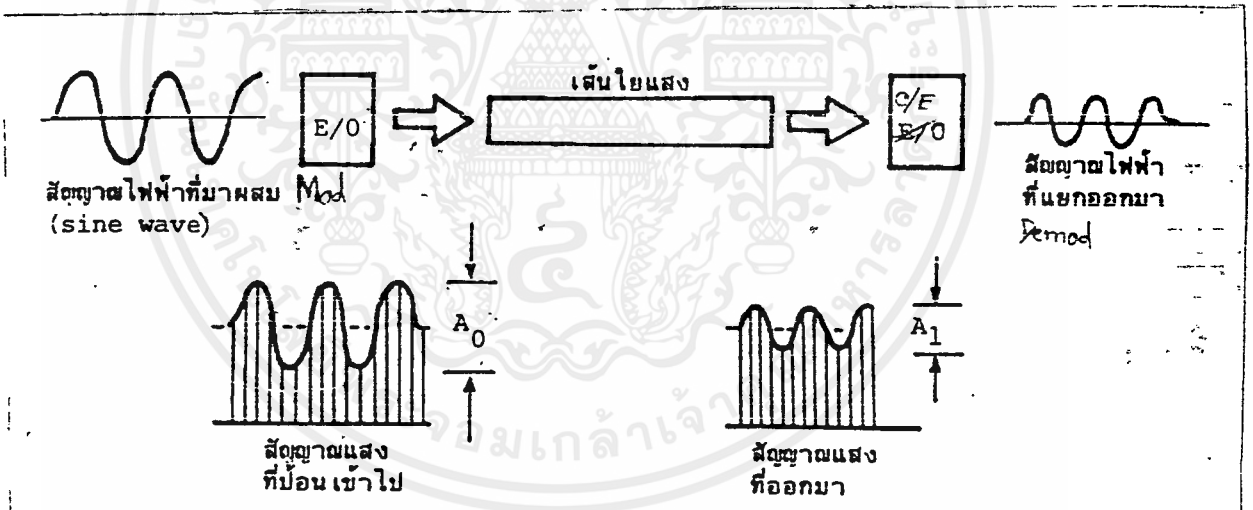
คำว่า BASE BAND FREQUENCY CHARACTERRISTIC นั้นหมายถึง คุณสมบัติความถี่ของอัตราส่วนของขนาดของสัญญาณไฟฟ้าซึ่งเป็น SINE WAVE ที่นำมาผสมทางด้าน INPUT กับสัญญาณไฟฟ้า ที่แยกออกมาจากการผสม (DEMODULATION) ทางด้าน OUTPUT ในสายส่งที่เป็นเส้นใยแสดง ดังในรูป 2.(a) สำหรับเส้นใยที่แสดงเป็น MULTI MODE เนื่องจากผลของ MODE DISPERTION ดังที่ได้กล่าวมาแล้วจะทำให้ได้ว่า เมื่อความถี่ที่นำมาผสมทางด้าน INPUT สูงขึ้น ขนาดของสัญญาณที่แยกออกมาจากการผสมทางด้าน OUTPUT จะลดลง ดังแสดงด้วยกราฟในรูป 2.(b) นอกจากนั้นยังขึ้นกับระยะทางด้วย ขนาดความกว้างของ BANDWIDTTH ที่ 6 dB นั้น เรียกว่าเป็นช่วงความถี่ ที่ขนาดของสัญญาณไฟฟ้าที่แยกออกมาจากการผสม ภายหลังจากแสงเดินทางได้ 1 Km แล้ว ปรากฏว่าขนาดของมันลดลงเป็น 1/2 (ปริมาณการสูญเสียพลังของแสงเป็น 3 dB) เมื่อเทียบกับขนาดของสัญญาณไฟฟ้าที่นำมาผสมทางด้าน INPUT และใช้หน่วยเป็น MHz.Km

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

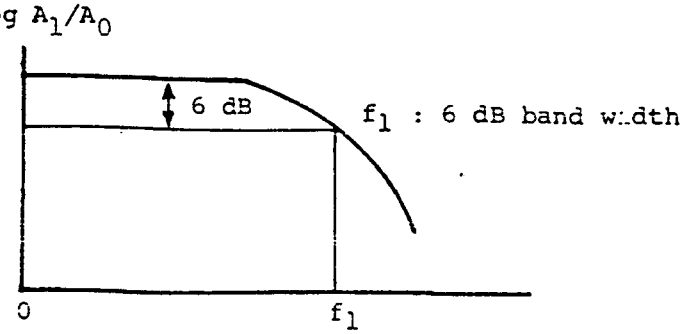
(3) ความแข็งแรงทาง MECHANIC ของเส้นใยแสง

ความแข็งแรงทาง MECHANIC ของเส้นใยแสงนั้น เป็นองค์ประกอบที่สำคัญอย่างหนึ่ง ที่จะต้องคำนึงถึง ในการนำไปใช้งาน ในระบบการสื่อสารด้วยเส้นใยแสง โดยเฉพาะอย่างยิ่ง ความแข็งแรงในการดึง (TENSION STRENGTH) สำหรับสายเคเบิ้ลโลหะนั้น เนื่องจากมันมีความยืดหยุ่น (FLEXIBILITY) อยู่ในตัวจึงไม่ค่อยมีปัญหามาก แต่สำหรับเส้นใยแสงนั้นเนื่องจากมีความเปราะของแก้วอยู่ จึงจำเป็นต้องคำนึงถึงเรื่องนี้อย่างมาก

ความแข็งแรงในการดึงของเส้นใยแสงที่ทำด้วยแก้วนั้น ในปัจจุบันมีค่าสูงประมาณ 300 Kg/mm^2 กล่าวคือมีค่าประมาณ 2 เท่าของเหล็กกล้า และมีค่ามากกว่า 10 เท่าของทองแดง, อะลูมิเนียม แต่ทว่าถ้าหากมีแรงดึงมากระทำต่อเส้นใยแสงนั้น แรงที่มากระทำนี้จะไปรวมกันตรงที่มีรอยชีดข่วน และถ้าหากมีค่าเกินกว่า แรงเครียดที่มันทนได้แล้ว เส้นใยแสงจะขาดออกทันที



(a) การจัด base band frequeney chorocteristic



(b) คุณสมบัติ base band frequeney และ 6 dB band width

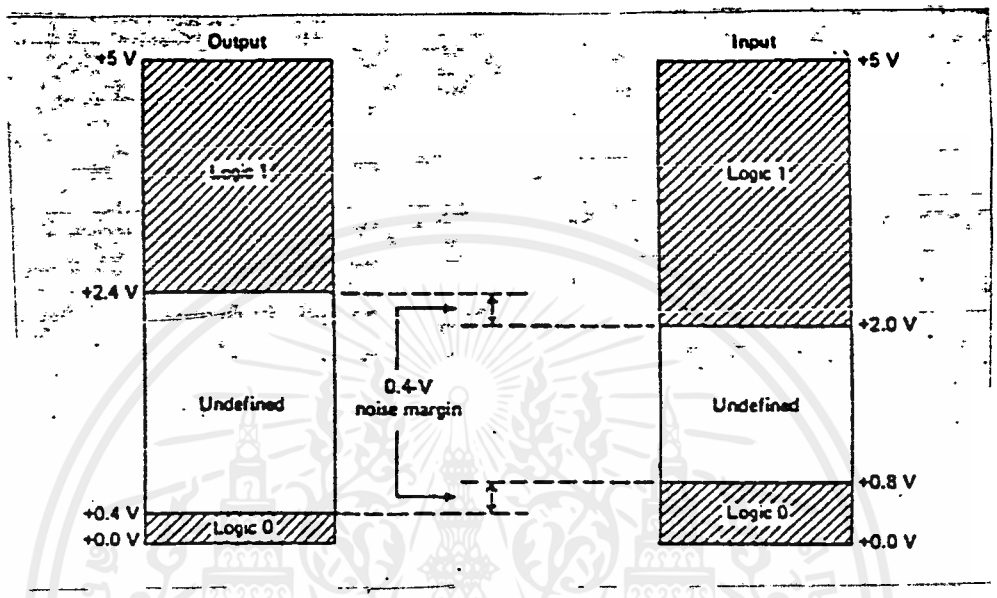
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-232-C

ก่อนอื่นเราจะกล่าวถึงปัญหา ที่เกิดขึ้นในการส่งข้อมูลระหว่าง อุปกรณ์อิเลคทรอนิคส์ ซึ่งการส่งข้อมูล จะเกิดขึ้นได้นั้น จะต้องมีแหล่งกำเนิดสัญญาณ แหล่งรับสัญญาณ และการเชื่อมต่อระหว่างแหล่งสัญญาณสองแหล่งนี้ ในการส่งข้อมูลระหว่างชิพไอซีสสองตัว ที่อยู่บนแผ่นวงจรพิมพ์แผ่นเดียวกัน เช่น ในการอ่านข้อมูลจากหน่วยความจำ ซึ่งเป็นขบวนการที่เกิดขึ้นระหว่าง CPU และ หน่วยขบวนการนี้จะมีปัญหาบางอย่างเกิดขึ้นดังจะอธิบายต่อไปนี้

โดยทั่วๆ ไปข้อมูลจะถูกแทนด้วยเลขฐานสอง ระดับแรงดัน 0 โวลต์ จะแทนลอจิก 0 การแทนลอจิก ด้วยระดับแรงดันเหล่านี้ เรียกว่าการแทนระดับสัญญาณของ อุปกรณ์ TTL ซึ่งการแทนระดับสัญญาณแบบนี้ ถูกใช้กันทั่วไป ในการส่งข้อมูลระหว่างระดับอุปกรณ์ชิ้นหนึ่งไปยังอีกชิ้นหนึ่งภายใน เครื่องคอมพิวเตอร์ แต่มันเป็นการไม่เหมาะสมที่จะกำหนดระดับแรงดันในการแทนลอจิก 0 และ 1 ที่จำเพาะเจาะจงลงไป เป็นค่าเฉพาะเพียงค่าเดียว ดังนั้นระดับแรงดันที่ใช้แทนลอจิก 0 และ 1 นั้นจึงถูกกำหนดเป็นพิสัย พิสัยนี้แสดงไว้ดังในรูปจากรูปจะเห็นได้ว่าพิสัยของแรงดันที่ส่งจากอุปกรณ์ที่เป็นตัวรับ (พิสัยที่แทนลอจิก 0 และ 1 นั่นก็ คือ ตัวส่ง ใน บางครั้งเราเรียกว่าตัวสร้างสัญญาณ) จะต้องจ่ายแรงดัน ที่มีระดับสัญญาณต่ำที่สุดเท่ากับ 2.4 โวลต์ ในการสั่งลอจิก 1 แต่ในภาครับสัญญาณ ตัวรับ จะถือว่าระดับสัญญาณที่มีระดับสัญญาณระหว่าง 2.4-2.0 โวลต์ เป็นลอจิก 1 ที่เนื่องมาจากมีการสูญเสียของสัญญาณระหว่างตัวส่งและตัวรับขึ้น ความคลาดเคลื่อนของระดับแรงดัน ที่เกิดขึ้นนี้เรียกว่า "NOISE MARGIN" ซึ่งมีค่าเท่ากับ 0.4 โวลต์ ไม่ว่าจะเป็นของลอจิก 0 หรือ 1 (ขอให้สังเกตจากรูป) สำหรับลอจิก 0 ความคลาดเคลื่อนเกิดขึ้นจากสัญญาณรบกวน ที่ปนเข้ามา ในทางปฏิบัติความคลาดเคลื่อนถูกยอมรับ ให้ใช้ได้ในการใช้งานทั่วไป ซึ่งเป็นข้อกำหนดทางไฟฟ้าของการอินเทอร์เฟสแบบมาตราฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป แสดงคุณสมบัติทางไฟฟ้า

จากรูป จะเห็นว่ามีระดับแรงดันอยู่ช่วงหนึ่ง เรียกว่าช่วงการเปลี่ยนสถานะซึ่งไม่อาจระบุสัญญาณในช่วงนั้น ให้เป็นลอจิก 0 หรือ 1 ช่วงของสัญญาณนี้ทางด้านรับจะมีช่วงแคบกว่าทางด้านส่ง ดังนั้นระดับแรงดันทางด้านรับ ที่ใช้แทนลอจิก 0 หรือ 1 จะมีช่วงกว้างกว่าทางด้านส่ง

ในการส่งข้อมูลภายในเครื่องคอมพิวเตอร์ เราจะถือว่าระดับสัญญาณ ที่ใช้ส่งและรับเป็นแบบอุดมคติ เนื่องจากเหตุผลต่อไปนี้

1) กำลังที่ใช้ และการกระจายความร้อนมีค่าต่ำ

2) สัญญาณที่ใช้เป็นระดับสัญญาณลอจิกแบบ TTL ซึ่งสามารถจ่ายให้แก่ชิพ IC ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตรงโดย ไม่ต้องใช้ LINE DRIVER วงจรรับข้อมูลราคาแพง

3)การอินเทอร์เฟสระหว่างอุปกรณ์ TTL จะทำงานที่ความถี่สูง ซึ่งจำเป็นต้องใช้ในการส่งข้อมูล ภายในคอมพิวเตอร์

คุณสมบัติของสัญญาณไฟฟ้าของ RS-232

1)สัญญาณที่ขาทุกขาที่คอนเน็คเตอร์ ของ RS-232 นั้นจะมีสภาวะใดสภาวะหนึ่งในแต่ละคู่ของคู่ต่อไปนี้ MARE/SPACE , ON/OFF โดยที่สัญญาณ ที่แทนลอจิกหนึ่งจะแทนด้วยระดับแรงดันที่เป็น -15 โวลต์ ส่วนสัญญาณที่แทนลอจิก 0 จะแทนด้วยระดับแรงดันที่เป็น +15 โวลต์ ส่วนในช่วงแรงดันระหว่าง -3 ถึง +3 นั้นเป็นช่วงของการเปลี่ยนแปลง จะไม่มีการระบุสถานะของสัญญาณในช่วงนี้

2)ความเร็วสูงสุดประมาณ 19.2 Kbps ทั้งนี้เนื่องมาจากค่าของตัวเก็บประจุ

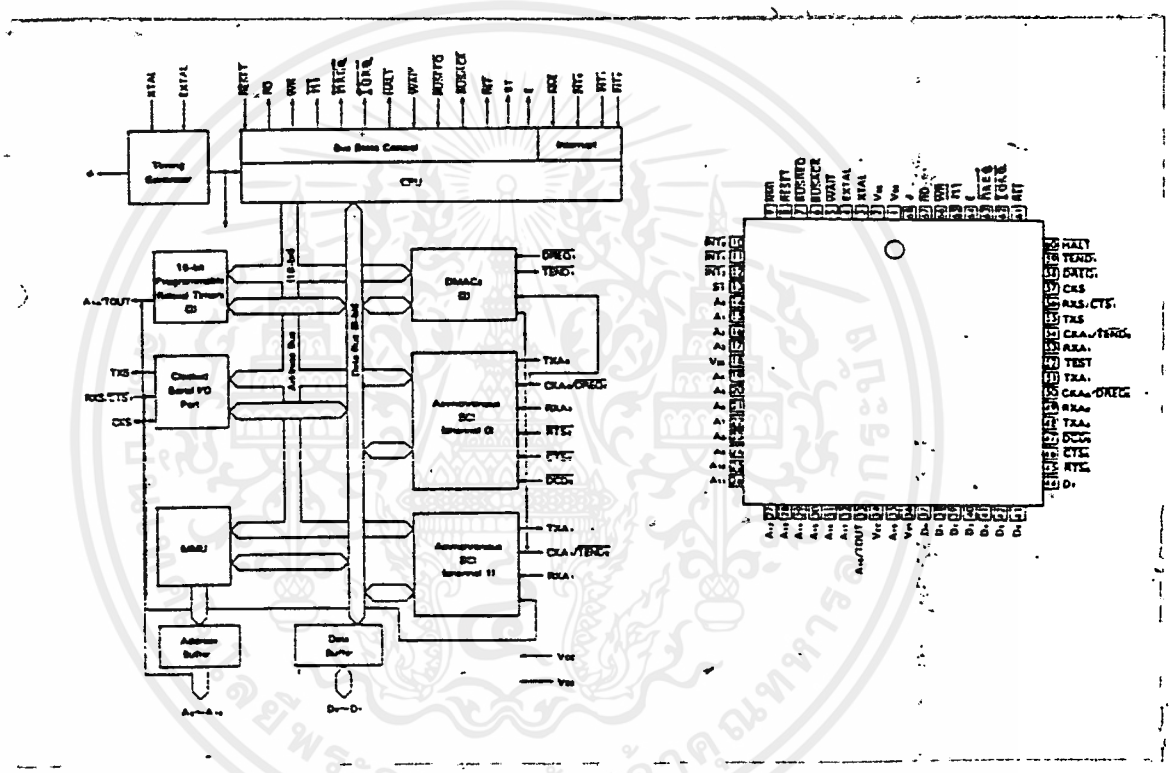
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80180

Z80180 เป็น CPU ที่มีความสามารถสูงที่ได้รวม CHIP สำคัญอื่นๆไว้ใน CPU CHIP เดียวจึงทำให้มีลักษณะคล้ายกับ CPU ที่ใช้ในงาน CONTROL ในจำพวก "SINGLE CHIP" แต่เนื่องจาก SINGLE CHIP มีข้อดี คือ เป็นระบบเล็กราคาถูก แต่ข้อเสีย คือ การโปรแกรม CONTROL ค่อนข้างยากในตอนเริ่มต้น และกับระบบงานที่ใหญ่ขึ้น แต่ Z80180 ทางด้านโปรแกรมจะสะดวกอย่างมาก เพราะคำสั่งที่ใช้มีมาก และตรงไปตรงมาทั้งคู่มือภาษาไทยและตัวอย่างการใช้งานที่มีมาก เพราะ CPU Z80180 นี้เป็น SUPPER COMPAT Z80 คือ คำสังทั้งหมดยังเป็น Z80 แต่ได้เพิ่มชุดคำสั่งขึ้นมา เพื่อเพิ่มความสะดวกในการใช้งานขึ้นอีก

เมื่อมองดูระบบ MICRO CONTROLLER "SINGLE CHIP" แล้ว Z80180 จะด้อยกว่าตรงที่ไม่มี ROM ,RAM และ PORT แต่ถ้าเป็นในระดับงานอุตสาหกรรมแล้ว ระบบของ Z80180 กับ CHIP MICOR CONTROLLER จะไม่ต่างกันเลย เพราะความต้องการเนื้อที่ในการเก็บข้อมูล และ PORT มากทำให้ต้องต่อเพิ่มภายนอกขึ้น จึงทำให้ Z80180 ในระดับงาน CONTROL อุตสาหกรรมคล่องตัวกว่ามากเพราะภายใน Z80180 ประกอบด้วย ส่วนที่เป็น CMOS , OSCILATOR ในตัว RUN ที่ 10MHz , MMU CHIP อ้าง MEMORYได้ 1 MBYTE , DMA 2 CHANEL , PORT สื่อสาร UART 2 CHANEL , CLOCK SERIAL I/O , 16 BIT TIMER COUNTER และเกี่ยวกับ PORT สื่อสารสามารถทำ MUULTI PROCESSOR COMMUNICATION ซึ่งโครงสร้างของ CHIP นี้เป็นดังรูป:-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาการใช้งาน

A0-A19	ADDRESS BUS ระหว่าง RESET จะเป็น HIGH IMPEDANCE
BUSAK	BUS ACKNOWLEDGE เป็นขา OUTPUT ACTIVE LOW ทำงานก็ต่อเมื่อ Z80180 ตอบสนองต่อการขอ BUS ของ BUSRQ และจะทำให้ BUS ข้อมูล BUS ADDRESS และสัญญาณ CONTROL บางเส้นเป็น HIGH IMPEDANCE
BUSRQ	BUS REQUEST เป็นขา INPUT ACTIVE LOW ซึ่งจะมีความสำคัญสูงกว่า NMI โดยจะมีการตรวจสอบสัญญาณนี้ ทุกๆการสิ้นสุดของ MACHINE CYCLE
CKA0 , CKA1	ASYNCHRONOUS CLOCK 0 และ 1 เป็นขาสัญญาณ CLOCK แบบ 2 ทิศทาง คือ จะใช้เป็นขา INPUT หรือ OUTPUT ก็ได้
CKS	SERIAL CLOCK เป็นขา CLOCK 2 ทิศทางของ CSI/O
CLOCK	เป็นขา OUTPUT โดยจะเป็นครึ่งหนึ่งของ X'TAL หรือ CLOCK OUT เช่น X'TAL 12 MHz Z80180 จะRUN ที่ 6 MHz
CTS0-CTS1	CLEAR TO SEND 0 และ 1 เป็นขา INPUT ACTIVE LOW ใช้ในการควบคุม MODEM
D0-D7	DATA BUS เป็นแบบ 2 ทิศทาง
DCD0	DATA CARRIER DETECT 0 เป็นขา INPUT ACTIVE LOW ใช้ควบคุมในการติดต่อกับ MODEM ของ ASCI CHANEL 0
DREQ0-DREQ1	DMA REQUTSE 0 และ 1 เป็นขา INPUT ACTIVE LOW ใช้ในการขอ DMA และขานี้จะโปรแกรมได้ ว่าให้ตรวจสัญญาณที่ขอบ หรือระดับได้
E	ENABLE CLOCK เป็นขา OUTPUT ACTIVE HIGH ซึ่งใช้ชิงค์การทำงานกับอุปกรณ์ภายนอก ระหว่างการทำงานเกี่ยวกับ BUS และใช้เชื่อมกับอุปกรณ์ในตระกูล 68XX และ 80XX
HALT	เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อทำคำสั่ง HALT หรือ SLP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTO	MASKABLE INTERRUPT 0 เป็นขา INPUT ACTIVE LOW สัญญาณที่ขานี้ จะถูกตรวจทุกๆ การสิ้นสุดของคำสั่ง
INT1, INT2	เช่นเดียวกับ INTO แต่มีระดับความสำคัญรองลงมาตามลำดับ
IORQ	เป็นขา OUTPUT เพื่อบอกว่ากำลังติดต่อกับ I/O หรือ IOE ใน 64180
M1	MACHINE CYCLE 1 เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อ FETCH OP-CODE หรือเป็นขา LIR ของ 64180
NMT	NON MASKABLE INTERRUPT เป็นขา INPUT ACTIVE LOW ขานี้จะตอบรับการ INTERRUPT เสมอ โดยไม่สามารถหยุดด้วย SOFTWARE
RD	เป็นขาที่ใช้ทำการอ่านข้อมูลจาก MEMORY หรือ I/O
RFSH	เป็นขาที่ให้ ADDRESS LOW (A0-A7) ไป REFRESH DYNAMIC RAM หรือ ขา REF ของ 64180
RTSO	REQUEST TO SEND เป็นขา OUTPUT ACTIVE LOW ขานี้ใช้ โปรแกรมสัญญาณควบคุมโมเด็มของ ASCI CANEL 0
RXAO, RXA1	RECEIVE DATA 0 และ 1 เป็นขารับสัญญาณจาก SERIAL PORT ของ ASCI
RXS	CLOCK SERIAL RECEIVE DATA เป็นขารับสัญญาณ SERIAL PORT ของ CSIO
ST	STATUS เป็นขา OUTPUT ACTIVE HIGH ใช้แสดงสถานะการทำงานของ CPU โดยร่วมกับ M1 และHALT ดังตาราง:-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ST	$\overline{HALT}$	$\overline{M1}$	Operation
0	1	0	CPU operation (1st op-code fetch)
1	1	0	CPU operation (2nd op-code and 3rd op-code fetch)
1	1	1	CPU operation (MC except for op-code fetch)
0	X	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (including SYSTEM STOP mode)

NOTE X: Don't care
MC: Machine cycle

TEND0-TEND1 TRANSFER END 0 และ 1 เป็นขา OUTPUT ACTIVE LOW ใช้แสดงถึงว่าทำ DMA สิ้นสุดลงแล้ว

TOUT TIMER OUT ใช้กำเนิดพัลส์จาก PRT CHANEL 1

TXA0 , TXA1 TRANSMIT DATA 0 และ 1 เป็นขาส่งข้อมูล SERIAL ของ ASCI CLOCK SERIAL TRANSMIT DATA เป็นขาส่งข้อมูล SERIAL ของ CSIO

WAIT ขา INPUT ACTIVE LOW จะถูกตรวจที่ขอบขาลงของ CLOCK ลูกที่ 2 ของทุกๆ MACHINE เพื่อเป็นการรอ ให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ CPU

WR ใช้สำหรับการส่งข้อมูลไปยัง I/O หรือ MEMORY

X'TAL เป็นขาที่ใช้ต่อกับ X'TAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ MULTIPLEX

A18/TOUT ระหว่าง RERET จะเป็น A18 แต่ถ้ามีการเลือก SET BIT TOC1 หรือ TOCO ใน TIMER CONTROL REGISTOR (TCR) ก็จะทำหน้าที่เป็น TOUT

CKAO/DREQO ระหว่าง RESET ขานี้จะเป็น CKAO แต่ถ้า DM1 หรือ SM1 ใน DMA MODE REGISTOR (DMODE) ถูก SET เป็น 1 จะเป็นขา DERQO

CKA1/TENDO ระหว่าง RESET จะเป็นขา CKA1 แต่ถ้า BIT CKA1D ใน ASCI ถูก SET จะเป็นขา TENDO

RXS/CTS1 ระหว่าง RESET ขานี้จะเป็นขา RXS ถ้า BIT CTS1E ใน ASCI ถูก SET จะเป็นขา CTS1

INTERNAL I/O REGISTOR

ซึ่งมีด้วยกัน 64 I/O ADDRESS ดังแสดงในรูป :-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERNAL I/O REGISTOR

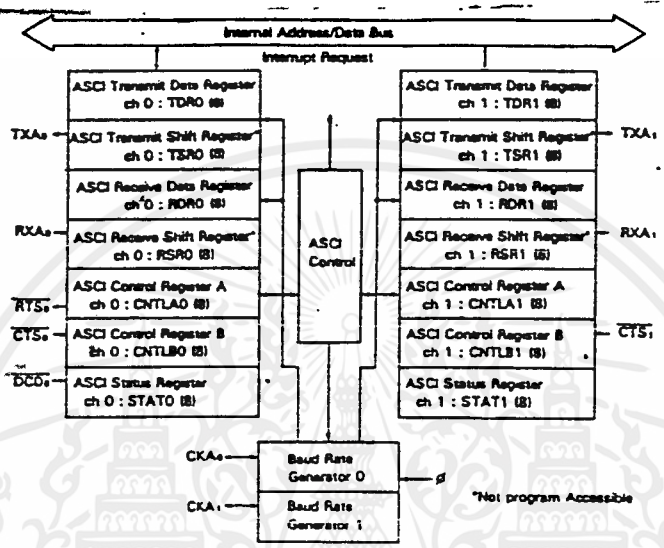
ซึ่งมีด้วยกัน 64 I/O ADDRESS ดังแสดงในรูป :-

	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCI	ASCI Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCI Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCI Control Register B Ch 0	CNTLB0	XX000010	02H
	ASCI Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCI Status Register Ch 0	STAT0	XX000100	04H
	ASCI Status Register Ch 1	STAT1	XX000101	05H
	ASCI Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCI Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCI Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCI Receive Data Register Ch 1	RDR1	XX001001	09H
CSI/O	CSI/O Control Register	CNTR	XX001010	0AH
	CSI/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch 0L	TMDR0L	XX001100	0CH
	Timer Data Register Ch 0H	TMDR0H	XX001101	0DH
	Reload Register Ch 0L	RLDR0L	XX001110	0EH
	Reload Register Ch 0H	RLDR0H	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001	11H
			⟩	⟩
			XX010011	13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
	Reload Register Ch 1L	RLDR1L	XX010110	16H
	Reload Register Ch 1H	RLDR1H	XX010111	17H
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			⟩	⟩
			XX011111	1FH
DMA	DMA Source Address Register Ch 0L	SAR0L	XX100000	20H
	DMA Source Address Register Ch 0H	SAR0H	XX100001	21H
	DMA Source Address Register Ch 0B	SAR0B	XX100010	22H
	DMA Destination Address Register Ch 0L	DAR0L	XX100011	23H
	DMA Destination Address Register Ch 0H	DAR0H	XX100100	24H
	DMA Destination Address Register Ch 0B	DAR0B	XX100101	25H
	DMA Byte Count Register Ch 0L	BCR0L	XX100110	26H
	DMA Byte Count Register Ch 0H	BCR0H	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Reserved		XX101101	2DH
	DMA Byte Count Register Ch 1L	BCR1L	XX101110	2EH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
	DMA/WAIT Control Register	DCNTL	XX110010	32H
INT	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H
	INT/TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
MMU	MMU Common Base Register	CBR	XX111000	38H
	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
I/O	Reserved		XX111011	3BH
			⟩	⟩
			XX111101	3DH
	Operation Mode Control Register	OMCR	XX111110	3EH
	I/O Control Register	ICR	XX111111	3FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASYNCHONOUS SERIAL COMMUNICATION INTERFACE (ASCI)

มีด้วยกัน 2 CHANEL โดยมี BLOCK DIAGRAM ดังรูป:-



TSR 0,1 เป็น SHIFT REGISTOR ที่รับข้อมูลจาก TRANSMIT DATA REGISTOR (TDR) แล้วนำข้อมูลนั้น SHIFT ออกที่ขา TXA

TAR 0, 1 (I/O ADDRESS 06H , 07H) เป็น REGISTOR ที่ใช้ส่ง DATA ออกไปที่ขา TXA โดยการนำข้อมูลใน TDR ส่งไปที่ TSR เมื่อ TSR ว่างลง และสามารถที่จะเขียนข้อมูลเข้าไปที่ TDR ได้อีกในขณะที่ TSR กำลัง SHIFT ข้อมูลออกไปที่ขา TXA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RSR 0,1 เป็น REGISTOR ที่รับข้อมูลจาก RXA PIN เมื่อรับเต็ม BUFFER แล้ว ก็จะ SHIFT ไปที่ RSR ถ้า RSR ไม่ว่าง เมื่อมีการรับข้อมูล BYTE ต่อไปเข้าอีกจะเกิดข้อมูลทับซ้อนกันขึ้น จะทำให้เกิดการผิดพลาด และผลของการผิดพลาด ก็จะแสดงที่ REGITOR สถานะ REGISTOR นี้ไม่สามารถโปรแกรมได้

RDR 0,1 (I/O ADRESS 08H ,09H) คือ REGISTOR ที่ใช้เก็บข้อมูลที่รับเข้ามาจาก RXA PIN และในขณะที่ RDR กำลังบรรจุข้อมูลที่รับเข้ามาจาก RSR ข้อมูล BYTE ถัดไป สามารถรับเข้ามาต่อได้

STAT 0,1 (I/O ADDRESS 04H, 05H) แต่ละ CHANEL จะมี REGISTOR ใช้สำหรับตรวจสอบการสื่อสาร เกี่ยวกับการผิดพลาด และสถานะสัญญาณ CONTROL MODEM การ ENABLE และ DISABLE ASCI ดังรูป:-

BIT	7	6	5	4	3	2	1	0
STAT 0	RDRF	OVRN	PE	FE	RIE	DCDO	TDRE	TIE

BIT	7	6	5	4	3	2	1	0
STAT 1	RDRF	OVRN	PE	FE	RIE	CTSIE	TDRE	TIE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RDRF : RECEIVE DATA REGISTOR FULL จะถูก SET เป็น 1 เมื่อข้อมูลที่รับเข้ามาถูกส่งเข้ามาที่ RDR เรียบร้อยแล้ว (ครบ BYTE) แต่ถ้าการรับเกิด ERROR ขึ้น RDRF ก็จะถูก SET ค้าง และข้อมูลที่ผิดนั้น ก็จะถูกส่งมาที่ RDR และคงอยู่ ดังนั้นจะต้องทำการ CLEAR FLAG ERROR RDRF จะถูก CLEAR เป็น 0 เมื่ออ่าน RDR , DCDO เป็น HIGH สำหรับ CHANEL 0, IOSTOP และการ RESET

OVRN : OVERUN ERROR จะเป็น 1 เมื่อ RDR เต็ม และ RSR เต็ม แล้วยังมีการรับข้อมูลเข้ามาอีก จะถูก CLEAR ได้ เมื่อ EFR BIT ใน CNTLA เป็น0, DCDO เป็น HIGH IOSTOP และ RESET

PE : PARITY ERRORเป็น 1 เมื่อข้อมูลที่รับเข้ามา PARITY ผิด และ CLEAR ได้ เช่นเดียวกับ OVERUN

FE : FRAMING ERROR เมื่อข้อมูลที่รับเข้ามา มีรูปแบบผิดไปจากที่กำหนด BIT FE จะถูก SET เป็น 1 และการ CLEAR ก็เช่นเดียวกับ OVERUN

RIE : RECEIVE INTERRUPT ENABLE เมื่อเป็น 1 จะอนุญาติให้ ASCI ทำการขอ INTERRUPT ได้ เมื่อ RDRF , OVRN , PE หรือ FE ถูก SET เป็น 1 ด้วยเมื่อนั้น ASCI ก็จะให้สัญญาณ INTERRUPT สำหรับ ASCI CHANEL 0 INTERRUPT สามารถเกิดขึ้นโดย การเปลี่ยนแปลงที่ขารับสัญญาณ INPUT ภายนอก ที่ขา DCDO จาก LOW เป็น HIGH และ RIE จะถูก CLEAR เป็น 0 ระหว่าง RESET

DCDO : DATA CARRIER DETECT BIT นี้จะถูก SET เป็น 1 เมื่อขา INPUT DCDO เป็น HIGH และจะถูก CLEAR เป็น 0 จากการอ่าน STAT 0 ครั้งแรก จากนั้นขา INPUT DCDO จะถูกเปลี่ยนจาก HIGH เป็น LOW และระหว่าง RESET เมื่อ DCDO เป็น 1 ส่วนของภาครับจะไม่ทำงาน

CTSIE : CHANEL 1 CTS ENABLE ที่ CHANEL 1 มีขา INPUT CTS1 ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง MULTIPLEX กับ RXS เมื่อ SET BIT นี้เป็น 1 จะถูกเลือกเป็นขา CTS1

TDRE : TRANSMIT DATA REGISTOR EMPTY เป็นตัวบอกว่า ข้อมูลพร้อมที่จะส่งได้หรือไม่ ถ้าเป็น 1 คือ พร้อมที่จะส่งข้อมูลแล้วให้เขียนข้อมูลเข้าไปที่ TDRได้ และ เมื่อมีการเขียนข้อมูลเข้าไปที่ TDR ก็จะทำให้ TDRE เป็น 0 และข้อมูลใน TDR ก็จะถูกส่งให้ TSR จน TDR ว่างลง TDRE ก็จะกลับเป็น 1 อีกครั้ง

TIE : TRANSMIT INTERRUPT ENABLE เมื่อเป็น 1 จะอนุญาติให้ ASCI ใช้การส่งแบบ INTERRUPT ได้ โดยที่ TDRE ต้องเป็น 1 ด้วย TIE จะถูก CLEAR เป็น 0 ระหว่าง RESET

CNTLA 0 , 1 (I/O ADDRESS 00H-01H) เป็น REGISTOR กำหนดการทำงานประกอบด้วย

BIT	7	6	5	4	3	2	1	0
CNTLA 1	MPE	RE	TE	CKAID	MPBR/ EFR	MOD2	MOD1	MOD0

BIT	7	6	5	4	3	2	1	0
CNTLA 0	MPE	RE	TE	RTS0	MPBR/ EFR	MOD2	MOD1	MODE0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MPE : MULTIPROCESSOR MODE ENABLE ใช้ ENABLE ในการสื่อสาร แบบไมโครโปรเซสเซอร์ เริ่มจากเมื่อมีการเลือก MODE การสื่อสารแล้ว (MP = 1ใน CNTLB) ในการสื่อสารแบบนี้ FORMAT ของการสื่อสารจะมี BIT พิเศษเพิ่มเข้ามาเรียกว่า MPB BIT ซึ่ง BIT นี้จะถูกใช้ในการตรวจสอบ หรือใช้งาน เมื่อ ENABLE MPE ให้เป็น 1 และถ้า MPB=1 เมื่อนั้น ภาครับของ MULTIPROCESSOR จะทำงานคือ RDRF และ ERROR FLAG จะทำงาน แต่ถ้า MPB = 0 ASCI จะไม่สนใจข้อมูล BYTE นั้น ถ้า MPE=0 จะไม่สามารถ ทำการสื่อสารแบบไมโครโปรเซสเซอร์ร่วมได้ แม้จะ SET MP เป็น 1 แล้วก็ตาม

RE : RECEIVER ENABLE ถ้าเป็น 1 จะ ENABLE การรับของ ASCI แต่ถ้าเป็น 0 จะ DISABLE การรับ แต่ RDRF และ ERROR FLAG จะไม่ถูก RESET ตาม

TE : TRANSMIT ENABLE เป็น 1 จะ ENABLE การส่ง ถ้าเป็น 0 จะ DISABLE แต่ TDRE FLAG จะไม่ถูก RESET ตาม

RTSO : REQUEST TO SEND CHANEL 0 เป็น BIT ที่ให้ผลเช่นเดียวกับขา OUTPUT RTSO คือ ถ้า BIT นี้เป็น 1 ขา OUTPUT RTSO ก็จะเป็น 1 ถ้า BIT นี้เป็น 0 ขา OUTPUT ก็เป็น 0 RTSO BIT นี้จะถูก SET เป็น 1 ระหว่าง RESET

CXA1D : CKA1 CLOCK DISABLE ซึ่งขา CKA1 จะ MULTIPLEX กับ TENDO เมื่อ BIT นี้เป็น 1 จะเลือกเป็นขา TENDO แต่ถ้าเป็น 0 ก็จะเป็นขา CLOCK ของ ASCI CHANEL 1 BIT นี้จะเป็น 0 ระหว่าง RESET

MPBR/EFR MULTIPROCESSOR BIT RECEIVE / ERROR FLAG เมื่อ BIT นี้ถูกอ่าน จะใช้ดู MPB BIT ในกรณีที่ พบไมโครโปรเซสเซอร์ ที่จะทำการติดต่อแล้ว และจะ DISABLE ไมโครโปรเซสเซอร์ตัวอื่นๆ ก็โดยการส่ง MPB BIT ให้เป็น 0 เมื่ออ่านจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รู้ว่า MPB เป็น 0 จริง แต่ถ้าเขียน 0 ให้ BIT นี้จะเป็นการ RESET ERROR FLAG ในการรับ

MOD 2 , 1 ,0 : ASCI DATA FORMAT MODE 2 , 1 , 0 โดย

MOD 2 = 0 7 BIT , 1 = 8 BIT MOD1 = 0 NOPARITY ,

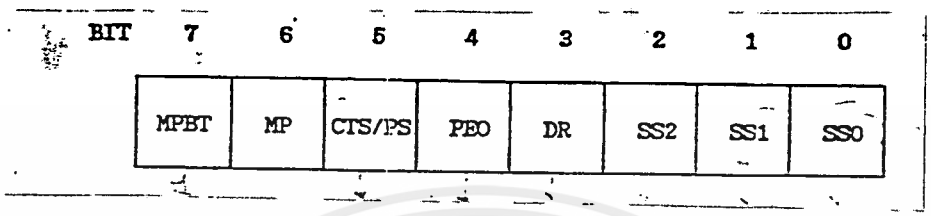
1 = PARITY ENABLE

MOD 0 = 0 1 STOP BIT , 1 = 2 STOP BIT สรุปได้ดังตาราง

MOD 2	MOD 1	MOD 0	DATA FORMAT
0	0	0	START + 7 BIT DATA + 1 STOP
0	0	1	START + 7 BIT DATA + 2 STOP
0	1	0	START + 7 BIT DATA + PARITY + 1 STOP
0	1	1	START + 7 BIT DATA + PARITY + 2 STOP
1	0	0	START + 8 BIT DATA + 1 STOP
1	0	1	START + 8 BIT DATA + 2 STOP
1	1	0	START + 8 BIT DATA + PARITY + 1 STOP
1	1	1	START + 8 BIT DATA + PARITY + 2 STOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCI CONTROL REGISTOR B 0 , 1 (CNTLB 0 ,1 I/O ADDRESS 02H ,03H) ประกอบด้วย



MPBT : MULTIPROCESSOR BIT TRANSMIT ใช้ส่ง MPB BIT โดยถ้า MPBT = 1 เมื่อนั้น MPB BIT =1 และ MPBT = 0 MPB ก็ = 0 ด้วย ระหว่าง RESET ไม่สามารถกำหนดได้

MP : MULTIPROCESSOR MODE ถ้าเป็น 1 จะเป็นการ SET การติดต่อ แบบไมโครโปรเซสเซอร์ร่วม โดยใช้ FORMAT ของ MOD 2 กับ MOD 0 โดยยกเว้น MOD 0 ดังนี้

START BIT + 7 หรือ 8 DATA BIT + MPB BIT + 1 หรือ 2 STOP BIT

ระหว่าง RESET MP จะเป็น 0

CTS/PS : CLEAR TO SEND / PRESCALE เมื่ออ่าน BIT นี้จะใช้แสดงสถานะของขา INPUT CTS ภายนอก ถ้าขา CTS เป็น HIGH ภาคส่งของ ASCI จะไม่ทำงาน แต่ถ้าเขียนเข้าไปที่ BIT นี้ จะเป็นการกำหนด BAUD RATE BIT นี้ เป็น 0 ระหว่าง RESET

PEO : PARITY EVEN ODD BIT นี้จะไม่มีผลต่อการ ENABLE หรือ DISABLE ของ PARITY (MOD 1 ใน CNTLA) แต่จะใช้วิธีเลือกว่า เมื่อมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ENABLE PARITY ใน MOD 1 จะให้ PARITY คู่ หรือ คี่ ถ้า PEO = 0 คือคู่ แต่ถ้า = 1 คือ คี่

DR : DIVIDE RATIO ใช้กำหนด BAUD RATE BIT นี้จะเป็น 0 ระหว่าง RESET

SS2 , 1 , 0 : SOURCE / SPEED SELECT 2 , 1 , 0 ใช้กำหนด CLOCK ว่าจะให้เป็นภายใน หรือภายนอก (โดยกำหนด คือ ขา CLOCK CKA) และ เป็นตัวกำหนด BAUD RATE ด้วย ระหว่าง RESET ทั้ง 3 BIT นี้จะเป็น 1 คือ เป็นการใช้ CLOCK จากภายนอกนั่นเอง ซึ่งจากที่กล่าวมา ในการกำหนด BAUD RATE จึงมีด้วยกันหลายตัว สามารถสรุปเป็นตารางดังนี้ :-

Prescaler		Sampling Rate		Baud Rate				General Divide Ratio	Baud Rate (Example) BPS)			CKA	
PS	Divide Ratio	DR	Rate	SS2	SS1	SS0	Divide Ratio		ϕ = 6.144 MHz	ϕ = 4.608 MHz	ϕ = 3.072 MHz	I/O	Clock Frequency
0	ϕ ÷ 10	0	16	0	0	0	÷ 1	ϕ ÷ 160	38400		19200	O	ϕ ÷ 10
				0	0	1	2	320	19200		9600		20
				0	1	0	4	640	9600		4800		40
				0	1	1	8	1280	4800		2400		80
				1	0	0	16	2560	2400		1200		160
				1	0	1	32	5120	1200		600		320
				1	1	0	64	10240	600		300		640
				1	1	1	–	fc ÷ 16	–	–	–	I	fc
		1	64	0	0	0	÷ 1	ϕ ÷ 640	9600		4800	O	ϕ ÷ 10
				0	0	1	2	1280	4800		2400		20
				0	1	0	4	2560	2400		1200		40
				0	1	1	8	5120	1200		600		80
				1	0	0	16	10240	600		300		160
				1	0	1	32	20480	300		150		320
				1	1	0	64	40960	150		75		640
				1	1	1	–	fc ÷ 64	–	–	–	I	fc
1	ϕ ÷ 30	0	16	0	0	0	÷ 1	ϕ ÷ 480		9600		O	ϕ ÷ 30
				0	0	1	2	960		4800			60
				0	1	0	4	1920		2400			120
				0	1	1	8	3840		1200			240
				1	0	0	16	7680		600			480
				1	0	1	32	15360		300			960
				1	1	0	64	30720		150			1920
				1	1	1	–	fc ÷ 16	–	–	–	I	fc
		1	64	0	0	0	÷ 1	ϕ ÷ 1920		2400		O	ϕ ÷ 30
				0	0	1	2	3840		1200			60
				0	1	0	4	7680		600			120
				0	1	1	8	15360		300			240
				1	0	0	16	30720		150			480
				1	0	1	32	61440		75			960
				1	1	0	64	122880		37.5			1920
				1	1	1	–	fc ÷ 64	–	–	–	I	fc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์สื่อสาร 8250

พอร์ทสื่อสารเป็นส่วนสำคัญส่วนหนึ่ง ที่มีอยู่บนไมโครคอมพิวเตอร์ 16 บิต พอร์ทสื่อสารนี้มีชื่ออีกอย่างหนึ่งว่า คอมพอร์ท (com port) ผู้ออกแบบพอร์ทสื่อสาร ต้องการให้เป็นไปตามมาตรฐานการเชื่อมต่อแบบอนุกรม ที่เรียกว่า RS-232C พอร์ทนี้เป็นทางออกของข้อมูลที่ผู้ใช้สามารถ รับส่งกับระบบอื่นได้ พอร์ทสื่อสาร RS-232C จึงเป็นพอร์ทที่จำเป็น

พอร์ทสื่อสาร RS-232C บนเครื่องไมโครคอมพิวเตอร์ 16 บิต มีโครงสร้างที่สามารถโปรแกรมได้ ด้วยการส่งรหัสคำสั่งให้กับ ชิพหลัก อย่างไรก็ตามผู้ออกแบบได้ให้ ทางเลือกในการสื่อสารด้วยกระแส (CURRENT LOOP) หรือ แบบแรงดัน คือ RS-232C โดยใช้ jumper เพื่อให้เลือกระบบ สำหรับตัว RS-232C นี้เป็นมาตรฐานแบบ อะซิงโครนัส (ASYNCHRONOUS) START BIT , STOP BIT และ PARITY BIT อัตราการส่งก็สามารถกำหนดได้ ตั้งแต่ 50 baud ถึง 9600 baud ลักษณะพิเศษของวงจร คือสามารถส่งสัญญาณ มาอินเตอร์รัพต์ (INTERUPT) CPU ตามเงื่อนไขได้ และยังมีโครงสร้างฮาร์ดแวร์ป้อนกลับ เพื่อใช้ในการตรวจสอบระบบ ว่าทำงานปกติดีหรือไม่ ได้ดีกด้วย

วงจรพอร์ทสื่อสารของไมโครคอมพิวเตอร์ 16 บิต นี้ใช้ไอซีหมายเลข 8250 เป็นตัวสำคัญของระบบ 8250 เป็นไอซีขนาด 40 ขา มีข้อความสามารถพิเศษดังนี้

- มีบัพเฟอร์ในตัวเพื่อทำให้ไม่จำเป็นต้องซิงโครไนซ์การรับส่ง
- ใช้สัญญาณนาฬิกาอิสระต่างหากไม่ขึ้นกับสัญญาณนาฬิกาของระบบ
- มีสัญญาณตอบโต้ควบคุมโมเด็มทั้ง CTS(clear to send),RTS(request to send), DSR(data set ready) , DTR(data terminal ready),RC(ring inductor)และสัญญาณดีเทคตัวพาหะ (carrier detect)
- ตรวจสอบสตาร์ทบิตที่ผิดพลาด

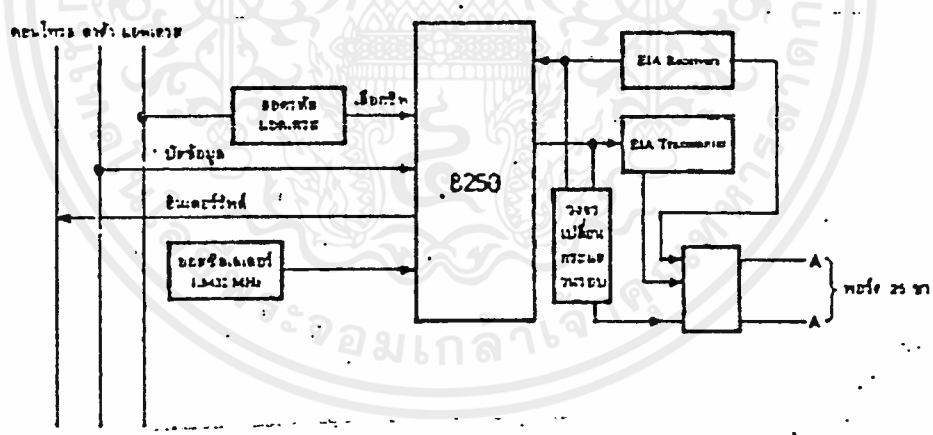
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรวจสอบและสร้างสัญญาณสายขาด (line break) เพื่อใช้ในการตรวจสอบการทำงานของระบบ

ชิพ 8250 มีโครงสร้างการทำงานคล้ายกับ 8251 ที่นักฮาร์ดแวร์ทั่วไปรู้จักดี การทำงานของ 8250 ต้องได้รับการโปรแกรมก่อน หลังจากนั้น จะทำงานตามรูปแบบที่ได้โปรแกรมคำสั่งใหม่ อย่างไรก็ตาม 8250 มีขีดความสามารถในการทำงาน ได้สูงกว่าจึงเป็นชิพประจำที่ใช้กับเครื่องไมโครคอมพิวเตอร์ 16 บิต

โครงสร้างของพอร์ทสื่อสาร

พอร์ทสื่อสารของเครื่องไมโครคอมพิวเตอร์ 16 บิต ที่จะกล่าวถึง เป็นระบบมาตรฐานตามแบบเครื่องไอบีเอ็ม-พีซีเอกซ์ที โครงสร้างบล็อกไดอะแกรมแสดงดังรูป



รูป แสดงโครงสร้างพอร์ทสื่อสารที่ใช้ 8250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า 8250 เป็นตัวรับข้อมูลจากบัสของระบบ ซึ่งก็คือ สล็อตขนาด 31*2 ขาของระบบนั่นเอง ซีพียูติดต่อกับ 8250 ในลักษณะพอร์ทที่เป็นอินพุตเอาต์พุท การจัดพอร์ตนี้กำหนดหมายเลขพอร์ตอย่างเจาะจง

ระบบไมโครคอมพิวเตอร์ 16 บิตมีพอร์ทสื่อสารสองพอร์ทคือ คอม 1 (COM_1) และ คอม 2 (COM_2) มีหมายเลขอินพุทเอาท์พุทพอร์ต ดังตาราง

อินพุทเอาท์พุทพอร์ท		เลือกรีจีสเตอร์	สถานะ DLAB
พอร์ท COM 1	พอร์ท COM 2		
3F8	2F8	บัฟเฟอร์ IX	DLAB =0 (เขียน)
3F8	2F8	บัฟเฟอร์ RX	DLAB =0 (เขียน)
3F8	2F8	แลทตัวหาร LSB	DLAB =1
3F9	2F9	แลทตัวหาร MSB	DLAB =1
3F9	3F9	รีจีสเตอร์อินเทอร์รัพอินนาเบิ้ล	
3FA	2FA	รีจีสเตอรเลือกอินเทอร์รัพ	
3FB	2FB	รีจีสเตอร์ควบคุมสายสื่อสาร	
3FC	2FC	รีจีสเตอร์ควบคุมโมเด็ม	
3FD	2FD	รีสีสเตอร์แสดงสถานะสายสื่อสาร	
3FE	2FE	รีสีสเตอร์แสดงสถานะโมเด็ม	

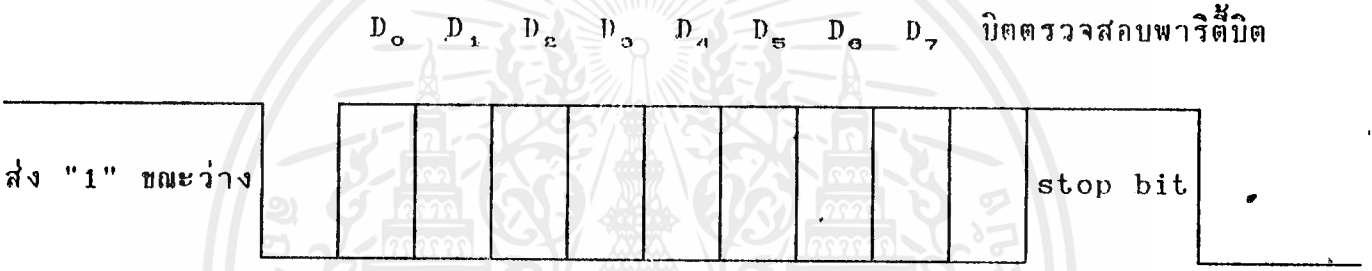
ตาราง หมายเลขของอินพุทเอาท์พุทพอร์ทของ COM 1 และ COM 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกหมายเลขอินพุตเอาต์พุตพอร์ทแยกเป็นสองกลุ่ม กลุ่มหนึ่งคือ COM 1 จะกำหนดหมายเลขพอร์ทจาก 3F8 - 3FE อีกกลุ่มหนึ่ง ถ้ากำหนดหมายเลขพอร์ตเป็น 2F8-2FE ในการเลือกหมายเลขรีจีสเตอร์ภายใน กำหนดด้วยแอดเดรส 3 บิต คือ A_0 , A_1 และ A_2 สำหรับการเลือก COM 1 และ COM 2 เราใช้เอดเดรส A_8 เป็นตัวเลือก

รูปแบบข้อมูลที่รับหรือส่ง

การสื่อสารข้อมูลของระบบนี้เป็นการสื่อสารแบบอะซิงโครนัส รูปแบบของข้อมูลจะมีสตาร์ตบิต บิตตรวจสอบพาริตี้ และ สต็อปบิต โครงสร้างของข้อมูลแต่ละเฟรมเป็นดังรูป



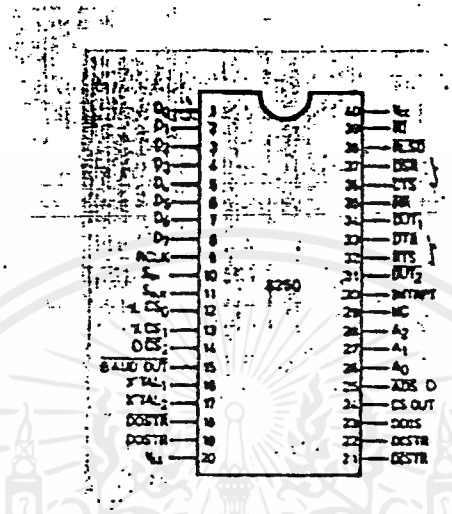
รูป แสดงรูปแบบข้อมูล 1 เฟรม

ข้อมูลบิตแรกของการส่งเป็นสตาร์ตบิต ระบบจะส่งสตาร์ตบิตก่อน หลังจากนั้นจะตามด้วยข้อมูล และแทรกด้วยบิตตรวจสอบพาริตี้บิต ตามด้วยสต็อปบิต ขนาดของข้อมูลมีค่าได้ตั้งแต่ 5-8 บิต สต็อปบิตมีได้ 1,1.5 หรือ 2 บิต ค่าเหล่านี้ สามารถกำหนดลงไปในรีจีสเตอร์ควบคุมสายสื่อสาร

ทีพ 8250

8250 เป็นไอซีขนาด 40 ขา มีการทำงานเพื่อควบคุมสิ่งต่างๆ ที่เกี่ยวกับการสื่อสารแบบอนุกรมได้หมด โครงสร้างทางฮาร์ดแวร์ของอะแดปเตอร์การ์ดนี้ จึงไม่ยุ่งยากมากนัก การจัดวางขาของไอซี เป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป แสดงการจัดวางขาของไอซี 8250

สัญญาณอินพุต

ขาเลือกชิพ CS_0,CS_1,CS_2 (chip select) ขา 12-14 เป็นสัญญาณเลือกชิพ โดยที่เมื่อต้องการเลือกชิพจะให้ CS_0 , CS_1 เป็น "1" และ CS_2 เป็น "0" สัญญาณเลือกชิพนี้จะได้รับการเลือก แลตช์ไว้ในขณะที่สัญญาณ ADS มีค่าเป็น "0" การเลือกชิพนี้จะมีไว้เพื่อให้ ซีพียูติดต่อกับ 8250

สโตรบข้อมูลอินพุต

DISTR , DISTR (data input stobe) ขา 22,21 เมื่อสัญญาณที่ DISTR เป็น "1" และ DISTR เป็น "0" ในขณะที่มีการเลือกชิพ เป็นขณะที่ซีพียู จะอ้างข้อมูลจากรีจิสเตอร์ภายในที่ได้รับการกำหนดไว้แล้ว มายังซีพียู สัญญาณนี้จึงเป็นสัญญาณอ่านข้อมูลหรือ read นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาสโตรบข้อมูลเอาต์พุท

DOSTR , DOSTR ขา 19 และขา 18 เป็นสัญญาณ ที่แอคตีฟขึ้นเพื่อให้ซีพียูเขียนข้อมูลลงมายังรีจิสเตอร์ของ 8250

ขาสโตรบแอดเดรส

ADS ขา 25 เมื่อมีค่าเป็น "0" จะแอคทีฟเพื่อแลตช์ค่า A_0-A_2 เลือกรีจีสเตอร์ภายในการเลือกรีจีสเตอร์ จะทำขณะที่การเลือกชิพแอคทีฟอยู่

ขาเลือกรีจีสเตอร์

A_0, A_1, A_2 ขา 26-28 เป็นตัวกำหนดแอดเดรสของรีจีสเตอร์ภายใน เพื่อให้ซีพียูทำการติดต่อกับ 8250 ตามค่ารีจีสเตอร์ที่กำหนด เพื่อการเขียนหรืออ่าน อย่างไรก็ตาม การทำงานของ A_0-A_2 นี้ก็ขึ้นกับสัญญาณเลือกตัวหาร DLAB (divisor latch access bit) ซึ่งกำหนดค่ารีจีสเตอร์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DLAB	A_2	A_1	A_0	รีจีสเตอร์
0	0	0	0	บัฟเฟอร์ตัวรับ(อ่าน)
0				รีจีสเตอร์โฮลดิ้ง(เขียน)
0	0	0	1	อินเทอร์รัพอีนาเบิ้ล
x	0	1	0	อินเทอร์รัพอีนาเบิ้ล
x	0	1	0	ควบคุมสายสื่อสาร
x	1	0	0	ควบคุมโมเด็ม
x	1	0	1	สถานะสายสื่อสาร
x	1	1	0	สถานะโมเด็ม
x	1	1	1	ไม่ใช้
1	0	0	0	แลตช์ตัวหาร(LSB)
1	0	0	0	แลตช์ตัวหาร(MSB)

รีเซต MR (master reset)

ขา 35 เมื่อมีค่าเป็น "1" จะรีเซตการทำงานของชิพ 8250 โดยทำให้ค่าต่างๆ ในรีจีสเตอร์ถูกเคลียร์หมด (ยกเว้นบัฟเฟอร์ของตัวรับ ตัวส่ง และตัวหาร) ฌณะทำการรีเซตแสดงไว้ในตาราง

สัญญาณนาฬิกาตัวรับ

RCLK (receiver clock) ขา 9 เป็นขาที่ตัวรับสัญญาณนาฬิกา เพื่อกำหนดอัตราบอด สัญญาณนาฬิกานี้จะมีค่าเป็น 16 เท่าของที่นำมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาอินพุตข้อมูลอนุกรม

SIN (serial input) เป็นขารับข้อมูลอนุกรมจากสายส่ง ในการเชื่อมโยงการติดต่อสื่อสาร

ขาเคลียร์ทูเซนด์

CTS (clear to send) ขา 36 เป็นสัญญาณที่ใช้ในการติดต่อกับโมเด็ม เงื่อนไขของสัญญาณนี้ สามารถเก็บไว้ภายในชิพ 8250 ที่จะให้ซีพียูอ่านไปตรวจสอบได้ โดยเก็บไว้ที่บิต 0 ของรีจีสเตอร์ แสดงสถานะโมเด็ม ส่วนบิต 0 ของรีจีสเตอร์ แสดงสถานะจะเป็นตัวบอกว่า CTS ได้เปลี่ยนสถานะไป หลังจากการอ่านครั้งก่อนแล้วหรือไม่

ขาดาต้าเซ็ตรีดี้

DSR (data set ready) ขา 37 เมื่อขานี้เป็น "0" จะแสดงว่าโมเด็ม หรือข้อมูลได้รับการเซ็ตรีดี้ เตรียมพร้อมแล้วสำหรับการเชื่อมต่อกับสายสื่อสาร และส่งข้อมูลระหว่าง 8250 กับโมเด็ม สัญญาณ DSR เป็นสัญญาณอินพุตของ 8250 ที่ซีพียู สามารถอ่านไปดูได้ทางบิตที่ 5 ของรีจีสเตอร์ แสดงสถานะจะเป็นตัวบอกว่าสัญญาณ DSR ได้เปลี่ยนสถานะไปหลังจากที่อ่านครั้งก่อนแล้วหรือไม่

หมายเหตุ ทั้ง CTS และ DSR เมื่อมีการเปลี่ยนสถานะ และถ้าได้รับการอีนาเบิ้ล modem status interrupt จะส่งผลในการสร้างสัญญาณอินเทอร์รัพ

ขาตรวจสอบสายสื่อสาร

RLSD (receive line signal detect) ขา 38 ถ้าเป็น "0" หมายถึงแอคทีฟ 8250 รับสัญญาณตรวจสอบสัญญาณพาหะจากโมเด็ม ว่า โมเด็มตรวจสอบได้แล้ว หรือข้อมูลได้รับการรีเซตแล้ว ซีพียูสามารถตรวจสอบสัญญาณนี้ทางบิต 7 ของรีจีสเตอร์แสดงสถานะ ส่วนบิตที่ 3 จะเป็น ตัวที่แสดงสถานะว่าสัญญาณนี้ ได้รับการเปลี่ยนหลังจากอ่านไปแล้วหรือยัง

ขาแสดงวงจรเรียก

RI (ring indicator) ขา 39 สัญญาณนี้แอคทีฟลอจิก "0" เป็นสัญญาณ ที่ส่งมาจากโมเด็ม โมเด็มตรวจสอบสัญญาณการเรียก (ringing) สัญญาณนี้ตรวจสอบได้ทาง บิต 6 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูสถานะการเปลี่ยนหลังจากอ่านแล้วจากบิตที่ 2

ขาไฟเลี้ยง

V_{cc} ขา 40 ,V_{ss} ขา 20 เป็นสัญญาณจากแหล่งจ่ายไฟเลี้ยง 5 โวลต์และกราวด์

Request to Send (RTS)

ขา 32 เริ่มขานี้มีลอจิกเป็น "0" หมายความว่า 8250 พร้อมที่จะส่งข้อมูลแล้ว สัญญาณขานี้ จะได้รับการเซตให้แอคทีฟ ด้วยการโปรแกรมค่าลงไปในรีจีสเตอร์ควบคุมบิตที่ 1

เอาท์พุต 1

OUT1 ขา 34 เป็นขาที่ผู้ใช้สามารถโปรแกรมให้แอคทีฟเป็น "0" ด้วยการโปรแกรมลงไปในบิตที่ 2 ของรีจีสเตอร์ควบคุมโมเด็ม

เอาท์พุท 2

OUT2 ขา 31 เป็นขาที่ผู้ใช้สามารถโปรแกรมให้แอคทีฟเป็น "0" ด้วยการโปรแกรมลงไปในบิตที่ 3 ของรีจีสเตอร์ควบคุมโมเด็ม

เลือกชิพเอาต์

CSOUT (chip select out) ขา 24 เมื่อมีค่าเป็น "1" จะบอกว่าชิพนี้ได้รับการเลือกชิพโดยซีพียูทางขา CS_0,CS_1,CS_2

ไดร์ฟเวอร์ดีสเอเบิ้ล

DDIS (driver disable) ขา 23 เป็นลอจิก "0" เมื่อซีพียูกำลังอ่านข้อมูลจาก 8250 สัญญาณ DDIS เป็น "1" มีไว้สำหรับการดิสเอเบิ้ลการรับส่งภายนอก ในกรณีที่ใช้ 8250 กับซีพียูผ่านทางบิต D_0-D_7 เพื่อบอกเวลาว่าซีพียูเป็น 8250 ติดต่อกันอย่างไร

สัญญาณบอดเอาท์

BAUDOUT ขา 15 เป็นสัญญาณนาฬิกาที่มีความถี่เป็น 16 เท่าสัญญาณนาฬิกา แล้วหารด้วยค่าที่โปรแกรมกำหนดในตัวทาง

อินเทอร์รัพ

INTRPT ขา 30 เมื่อเป็น "1" เป็นการส่งสัญญาณอินเทอร์รัพออกไปจาก 8250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเอาท์พุต

SOUT ขา 11 เป็นขาที่ใช้ส่งข้อมูลอนุกรมออกไปยังสายสื่อสาร

สัญญาณอินพุทเอาท์พุท

ข้อมูล D_0-D_7 เป็นสัญญาณต่อเชื่อมกับบัสข้อมูลระบบ

ขาสัญญาณ $XTAL_1$, $XTAL_2$ ขา 16-17 เป็นขาต่อกับคริสตอลเพื่อสร้างสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODEM

MODEM เป็นคำย่อ ซึ่งย่อมาจากคำว่า MODULATOR DEMODUALTOR ซึ่งถ้าเราจะแปลตรงความหมายก็คือ อุปกรณ์ที่ทำหน้าที่ MODULATE และ DEMODULATE สัญญาณ หรืออาจพูดง่ายๆ ว่า เป็นการนำสัญญาณข้อมูลฝากไปกับสัญญาณพาหะ เพื่อให้ง่ายต่อการส่งสัญญาณ

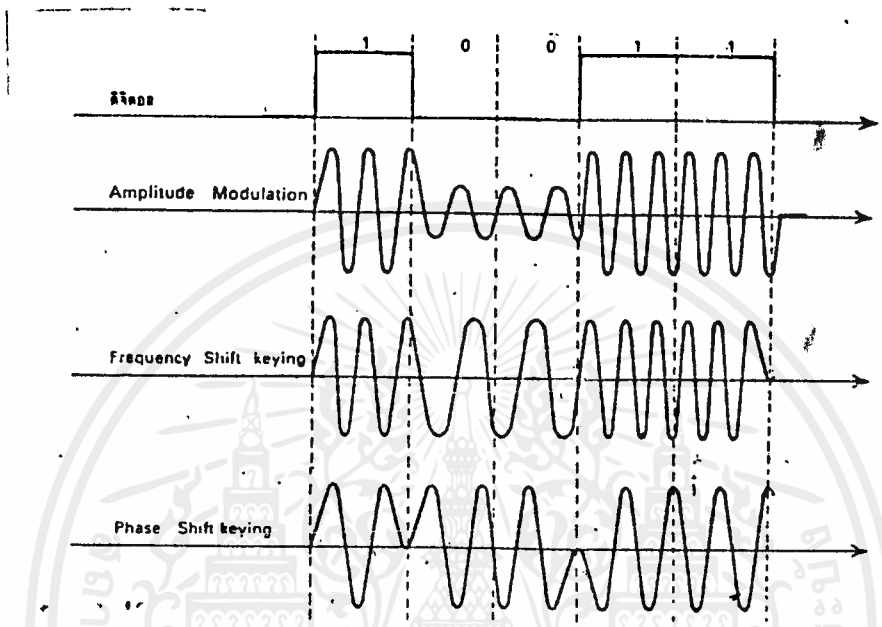
สำหรับการทำงานของ MODEM ทั่วๆไปนั้น จะเป็นการแปลงสัญญาณข้อมูลทางดิจิตอลไปเป็นสัญญาณเพื่อใช้ในการส่ง โดยทั่วไปจะมีด้วยกัน 3 รูปแบบ

1. แบบ AM (AMPLITURE MODULATE) เป็นการนำสัญญาณดิจิตอล เข้าไปเปลี่ยนความสูงของสัญญาณพาหะ

2. แบบ FSK (FREQUENCY SHIFT KEYING) เป็นการใช้ความถี่ของสัญญาณพาหะ 2 ความถี่ สำหรับแทนสัญญาณดิจิตอล

3. แบบ PSK (PHASE SHIFT KEYING) ใช้ในการเปลี่ยนเฟสของสัญญาณพาหะ ในการเปลี่ยนเฟสของสัญญาณพาหะ ในการแทนสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง หลักการทำงานของโมเด็มทั่วๆไป

ก่อนอื่นเราต้องทราบความความหมายของ BIT RATE และ BAUD RATE ซึ่งมีความหมายต่างกัน สำหรับคำว่า BAUD RATE นั้น ก็คืออัตราการเปลี่ยนแปลงสัญญาณใน 1 วินาที สัญญาณที่เปลี่ยนแปลงไปหนึ่งครั้ง อาจจะมีข้อมูลที่มากกว่า 1 บิต ในกรณี เช่นนี้ BIT RATE จะต่างกับ BAUD RATE ขึ้นอยู่กับสภาพของสัญญาณ ถ้าสัญญาณมี 4 สถานะภาพ การเปลี่ยนแปลงสัญญาณ 1 ครั้ง ก็จะให้ข้อมูล 2 บิต ในกรณีเช่นนี้ BIT RATEเป็น 2 เท่าของ BAUD RATE

สำหรับการรับ และการส่งข้อมูลของ MODEM นั้น ดังได้กล่าวมาแล้วว่าการรับส่งมีด้วยกันหลายรูปแบบ ตัวอย่างเช่น FULL DUPLEX และ HALF DUPLEX สำหรับโมเด็มทั่วไปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้กันจะเป็นแบบ FULL DUPLEX คือ จะมีการส่งในรูปแบบ ที่มีการตอบโต้โดยที่ความถี่การรับการส่งต่างละความถี่ สำหรับรูปแบบการส่งแบบ FULL DUPLEX นั้น อักขระที่ส่งไปยังฝ่ายตรงกันข้าม ซึ่งเป็นคู่สนทนาโดยการพิมพ์นั้น การแสดงผลของอักขระที่พิมพ์ จะไม่ปรากฏบนจอต้องรอให้ฝ่ายตรงข้าม สะท้อนกลับตัวอักขระนั้นกลับมา ส่วนในแบบ HALF DUPLEX นั้นนอกจากจะส่งไปทาง MODEM แล้วก็จะแสดงบนจอ

รูปแบบของ MODEM อาจแบ่งได้เป็น MODEN ภายใน และ MODEM ภายนอก (INTERNAL MODEM AND EXTERNAL MODEN) สำหรับ MODEM ภายในนั้นเป็น MODEM ที่อยู่ในการ์ดที่พร้อมจะเสียบลงเครื่องคอมพิวเตอร์ ซึ่งความสามารถในการใช้งานนั้น เป็นการที่จะส่งข้อมูลแบบ ASYSCHRONOUS โดยจะมี ตัวจัดการคือ 8250 ในการกำหนดรูปแบบการส่ง ซึ่งสามารถควบคุมโดย CPU สำหรับ MODEM ภายนอก ที่พบเห็นโดยทั่วไป จะมีรูปแบบดังได้กล่าวมาแล้ว คือจะทำการเปลี่ยนข้อมูลสัญญาณดิจิตอล ไปเป็นสัญญาณที่เหมาะสมที่จะทำการส่ง

ข้อดีของ MODEM ภายใน ก็คือมีขนาดเล็กสะดวก ส่วนข้อดีของ MODEM ภายนอกนั้นคือแสดง สถานะในการทำงาน ง่ายในการตรวจสอบ แต่ก็มีขนาดโตหน่อย

FIBER OPTICAL MODEM

สำหรับ MODEM ที่เรากำลังจะกล่าวถึงนี้ อาจกล่าวได้ว่าเป็น MODEM ที่ใช้งานกับ FIBER OPTIC โดยเฉพาะ ทั้งนี้เนื่องมาจากการทำงานของ FIBER OPTICAL MODEM นี้จะไม่มีการ MODULATE หรือ DEMODULATE ทั้งนี้เนื่องมาจากคุณสมบัติของ FIBER OPTIC ที่เราสามารถใช้ในการส่งข้อมูลเป็นสัญญาณดิจิตอลได้ และจากคุณสมบัติข้อนี้ ทำให้เป็นข้อแตกต่างกับ MODEM ทั่วๆไป สำหรับคุณสมบัติที่เด่นๆ ของ FIBER OPTIC ก็คือสามารถที่จะส่งด้วยความเร็วสูงๆ ทำให้ตัว MODEM ที่ใช้งานร่วมกัน จำเป็นต้องมีความเร็วในการส่งสูงเช่นเดียวกัน จึงเหมาะสมกับการใช้งาน (เพื่อที่ประหยัดเวลาในการส่ง เพราะ FIBER OPTIC ในปัจจุบันมีราคาแพงมาก)

หลักการของ FIBER OPTICAL MODEM นั้น เรา สามารถกล่าวให้เห็นภาพพจน์โดยรวม แบบง่ายๆ ซึ่งเป็นความคิดสร้างสรรค์ ในการนำมาซึ่งการประยุกต์ใช้งานจริง หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การในการทำงานนั้น เนื่องจากความเร็วในการส่งข้อมูลของเครื่องคอมพิวเตอร์ เป็นแบบ ASYSCHRONOUS ที่มีความเร็วจำกัด คือสูงสุดประมาณ 19.2 Kbps ดังนั้น ในการที่เราจะส่งข้อมูลให้ได้ความเร็วสูงนั้น จำเป็นจะต้องทำการนำข้อมูล มาทำการเก็บไว้ใน BUFFER เสียก่อนแล้วจึงทำการส่งข้อมูลไปในทีเดียวกัน ซึ่งความเร็วที่จะทำการส่งนั้น จะใช้ประมาณ 300 Kbps สำหรับรูปแบบในการส่งนั้น เราสามารถเลือกได้ ว่าจะเป็นการส่งแบบ SYNCHRONOUS หรือ ASYNCHRONOUS การเลือกพาริตี้บิต การกำหนด START BIT , STOP BIT หรือรูปแบบอื่นๆ ซึ่งในการส่งนั้น เราสามารถที่จะเลือกโปรโตคอลได้ โดยการควบคุมไมโครโปรเซสเซอร์ ที่เราทำการโปรแกรมไว้แล้ว

สำหรับรูปแบบการติดต่อ หรือการเลือกโปรโตคอล นั้นสามารถที่จะกำหนดได้ โดยการที่จะสั่งงานผ่านเครื่องคอมพิวเตอร์ซึ่งทำการเชื่อมโยงกับ FIBER OPTICAL MODEM ไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BASEBAND TRANSMISSION

สำหรับการทำงานในส่วนนี้ จะเป็นการนำสัญญาณข้อมูลที่จะทำการส่งซึ่งก็คือ นำสัญญาณที่ส่ง มาทำการเปลี่ยนแปลงให้เหมาะสมกับการส่ง ทั้งนี้เนื่องมาจากสัญญาณที่จะทำการส่งนั้นไม่เหมาะสม อาจเกิดการผิดพลาดได้ อันเนื่องมาจากคุณสมบัติของตัวส่ง และตัวรับซึ่งต่อไปนี้ เราจะแสดงรูปแบบต่างในการส่ง

ชนิดของคลื่น PCM ต่างๆ

1. NONRETURN TO ZERO (NRZ) แบ่งเป็น

- NRZ-L เลขไบนารี 1 ถูกแทนด้วยระดับ 1 เลขไบนารี 0 ถูกแทนด้วยอีกระดับหนึ่ง มีการเปลี่ยนแปลงระดับ เมื่อข้อมูลมีการเปลี่ยนจาก 1 เป็น 0 หรือจาก 0เป็น 1

- NRZ-S (S FOR SPACE)

- NRZ-M ไบนารี 1 ถูกแทนด้วยการเปลี่ยนระดับ และไบนารี 0 ถูกแทนด้วยการไม่เปลี่ยนระดับ

- NRZ-S ไบนารี 1 ถูกแทนด้วยการไม่เปลี่ยนระดับ และไบนารี 0 ถูกแทนด้วยการเปลี่ยนระดับ

2. RETURN TO ZERO(RZ)

- UNIPOLAR-RZ ไบนารี 1 ถูกแทนด้วยความกว้างครึ่งบิตพัลส์ และไบนารี 0 ถูกแทนด้วยการไม่มีพัลส์

- BIPOLAR-RZ ไบนารี 1 และ 0 ถูกแทนด้วยการเปลี่ยนบิต ที่ครึ่งบิตพัลส์ในแต่ละครึ่งบิตพัลส์

- RZ-AMI ไบนารี 1 ถูกแทนด้วยความกว้างที่ครึ่งบิตพัลส์ และมีการเปลี่ยนแปลงไปเรื่อย ๆ ส่วนไบนารี 0 ถูกแทนด้วยไม่มีพัลส์

3. PHASE ENCODED

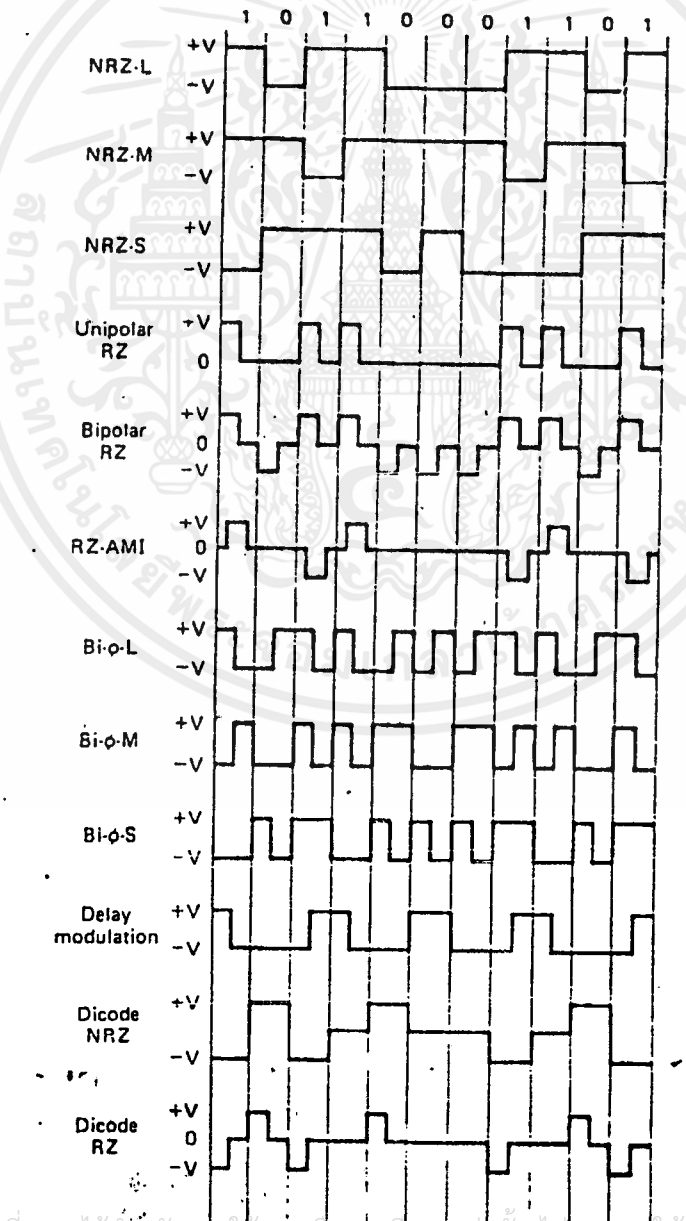
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- BI-O-L : รู้จักกันดีในชื่อของ รหัสแมนเชสเตอร์ โดยไบนารี 1 ถูกแทนด้วยพัลส์ครึ่งบิตแรก และไบนารี 0 ถูกแทนด้วยพัลส์ครึ่งบิตหลัง ภายในแต่ละบิต

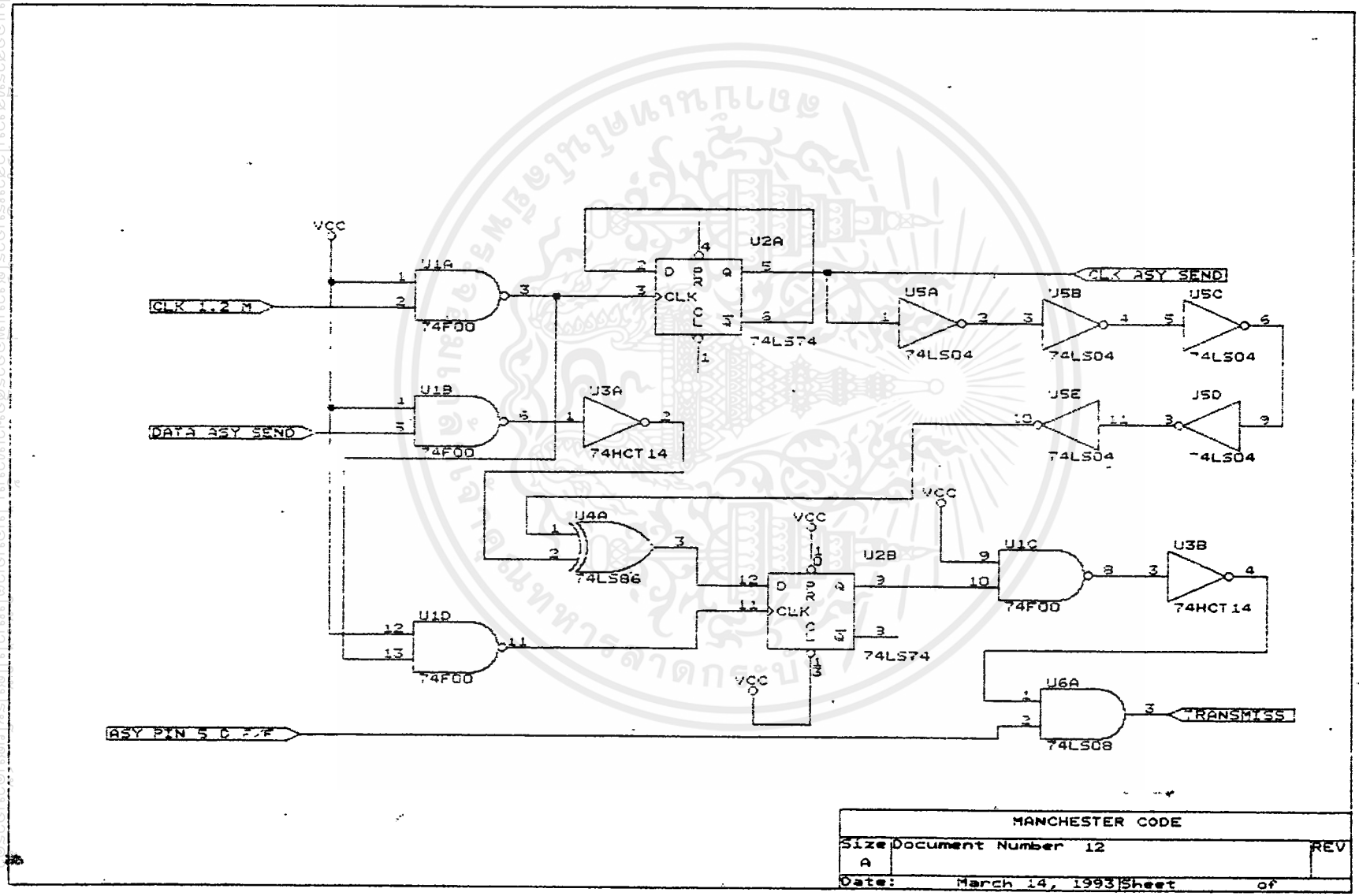
- BI-O-M : การส่งเกิดขึ้น ที่จุดเริ่มต้นของภายในแต่ละบิต โดย 1 ถูกแสดงด้วยครึ่งบิตแรก โดยมีการเปลี่ยนแปลงทุกๆ ครึ่งบิต

- BI-O-S : การส่งที่เกิดขึ้น ที่จุดเริ่มต้นของภายในแต่ละบิต โดย 0 แทนด้วยครึ่งบิตหลังในแต่ละพัลส์ โดยมีการเปลี่ยนแปลงทุกๆ ครึ่งบิต

-DELAY MODULATE : การส่งเกิดขึ้นที่จุดกลางบิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
U1A
CLK 1.2 M
74F00
U2A
D
CLK
Q
74LS74
CLK ASY SEND
U5A
74LS04
U5B
74LS04
U5C
74LS04
U1B
DATA ASY SEND
74F00
U3A
74HCT14
U5E
74LS04
U5D
74LS04
U4A
74LS86
VCC
U2B
74LS74
U1C
74F00
U3B
74HCT14
U1D
74F00
VCC
U6A
74LS08
TRANSMISS
ASY PIN 5 D F/F
MANCHESTER CODE
Size A
Document Number 12
REV
Date: March 14, 1993
Sheet of

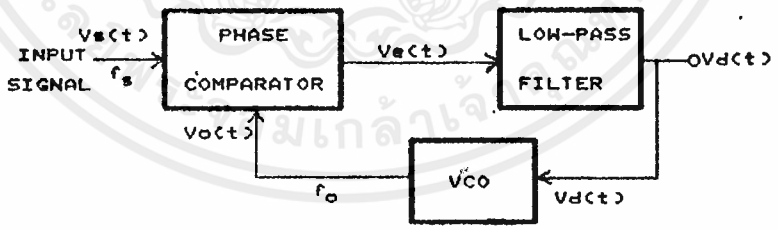
PHASE-LOCK LOOPS

Phase-Locked Loops (PLL) เป็น กระบวนการ ที่อาศัยการป้อนกลับของสัญญาณ O/P เพื่อทำการเปรียบเทียบความถี่ และ เฟส กับสัญญาณ I/P แล้วสร้างสัญญาณเพื่อไปบังคับ ให้การกำเนิดสัญญาณที่ภาค O/P มีเฟส และ ความถี่ ตรงกับภาค I/P ตามที่เราต้องการ

PLL มีส่วนประกอบสำคัญพื้นฐาน 3 ส่วน คือ

1. Phase comparator
2. Low-pass filter
3. Voltage Controlled Oscillator(VCO)

ดังรูปข้างล่าง



รูปแสดง block diagram ของระบบ PLL

ถ้ายังไม่มีสัญญาณป้อนเข้ามา V_d (error voltage) จะมีค่าเป็นศูนย์ ส่วนVCO จะกำเนิดสัญญาณ ที่ความถี่ที่ได้ตั้งค่าไว้ f_o ซึ่งเรียกกันว่า free-running frequency

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามีสัญญาณป้อนให้ระบบ phase comparator จะทำการเปรียบเทียบสัญญาณนี้ กับสัญญาณที่ได้จากส่วนVCO จะได้ค่า error voltage $V_d(t)$ ค่าหนึ่ง ซึ่งขึ้นอยู่ กับค่าความต่างกันของความถี่ และเฟสของ 2 สัญญาณนี้ ค่า error voltage นี้ จะถูก filter และป้อนเป็นสัญญาณควบคุมให้กับส่วน VCO ถ้าความถี่สัญญาณที่เข้ามา f_s มีค่าเข้าใกล้ค่า f_o เพียงพอแล้ว ระบบ PLL นี้ จะบังคับให้ส่วน VCO สร้างสัญญาณที่ synchronize หรือ lock กับสัญญาณที่ป้อนเข้ามา ในภาวะ synchronize นี้ ความถี่ของสัญญาณที่ได้จาก VCO จะเหมือนกับสัญญาณที่เข้ามา ยกเว้นเฟสที่อาจจะมีผลต่างของเฟสคงที่ได้

Lock range : เป็นย่านความถี่ที่ใกล้เคียงกับ f_o ที่ยังคงทำให้ระบบ PLL ยังคง Lock กับสัญญาณ I/P หรือที่เรียกกันว่า Tracking หรือ Holding range ก่อนที่จะหลุดจากการlock

: Lock range นี้จะเพิ่มขึ้นตามค่า gain ของ PLL ที่เพิ่มขึ้น

Capture range : ค่าของแถบความถี่ที่ใกล้เคียง f_o ซึ่งระบบ PLL สามารถจับสัญญาณ I/P และเริ่มทำการล็อคเฟส และความถี่ตามสัญญาณ I/P

: โดยปกติค่านี้ จะอยู่ในย่าน (range) ที่แคบกว่า ซึ่งค่านี้ขึ้นอยู่กับ bandwidth ของส่วน low-pass filter

ค่าLock range และค่า capture range ของส่วน PLL แสดงได้ดังรูป ซึ่งแสดง frequency-to-voltage characteristics ของ PLL จากรูป ให้I/P มีการเปลี่ยนแปลงความถี่อย่างช้า ๆ ตลอดแนวแถบความถี่ และในแนวตั้งเป็นค่าของ loop error voltage (V_d)

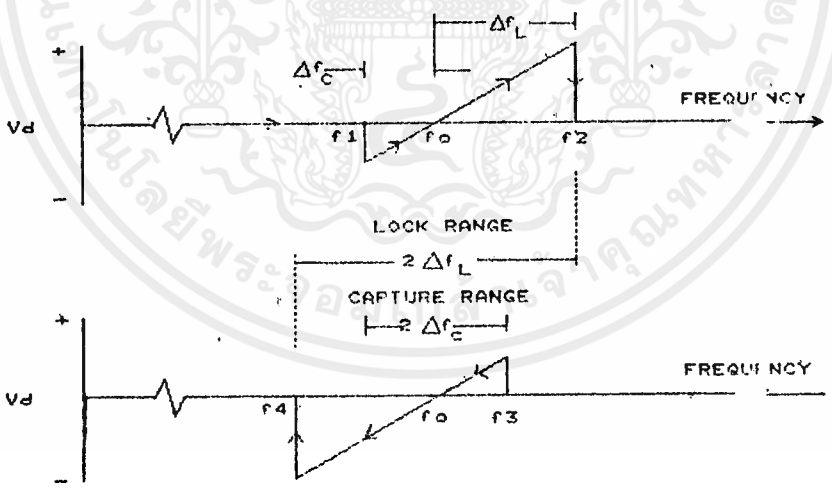
ในรูปบน ความถี่ของสัญญาณที่ป้อนเข้ามา มีค่าเพิ่มขึ้นอย่างต่อเนื่อง จนถึงค่าหนึ่ง f_1 ซึ่งเป็นค่า lower edge ของ capture range ระบบจะ lock กับสัญญาณ I/P ทันที ทำให้ค่า error voltage (V_d) ลงมาเป็นลบ ต่อมา V_d จะแปรผัน ตามความถี่ตามค่าความชัน (slope) ของเส้นกราฟ ซึ่งมีค่าเท่ากับ ค่าgainของส่วน VCO (Voltage-to-Frequency Conversion Gain) V_d จะมีค่าเป็น ศูนย์ ขณะที่ $f_s = f_o$ ระบบPLL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะยังคงlock กับ สัญญาณ I/P จนกระทั่งความถี่ I/P สูงถึงค่า f_2 (upper edge ของ lock range) ระบบจะหลุดจากการ lock ค่า error voltage ตกลงเป็น ศูนย์ อีกครั้งหนึ่ง

เมื่อลดความถี่ I/P ที่ป้อนเข้ามา ดังแสดงในรูปล่าง ระบบเข้าสู่ช่วง capture range ที่จุด f_3 และลดลงไปเรื่อย ๆ ถึง f_4 จะเห็นว่า ความถี่จะกระจายอยู่ระหว่างช่วง (f_1, f_3) และ (f_2,f_4) นั่นคือ $f_3 - f_1$ = capture range และ $f_2 - f_4$ = lock range

สรุปก็คือระบบ PLL จะตอบสนอง (response) ต่อความถี่สัญญาณ I/P ที่ใกล้เคียงกับค่า VCO frequency (f_o) ซึ่งถ้าอยู่ในช่วง "Lock" หรือ "Capture" range ของระบบ ดังนั้นประสิทธิภาพของระบบ ก็จะดูได้จากการเลือกค่า f_o ที่ทำให้ได้ ค่าที่ทำให้ช่วงความถี่ที่ครอบคลุมช่วง capture range และ lock range มีค่าสูงนั่นเอง ดังในรูป



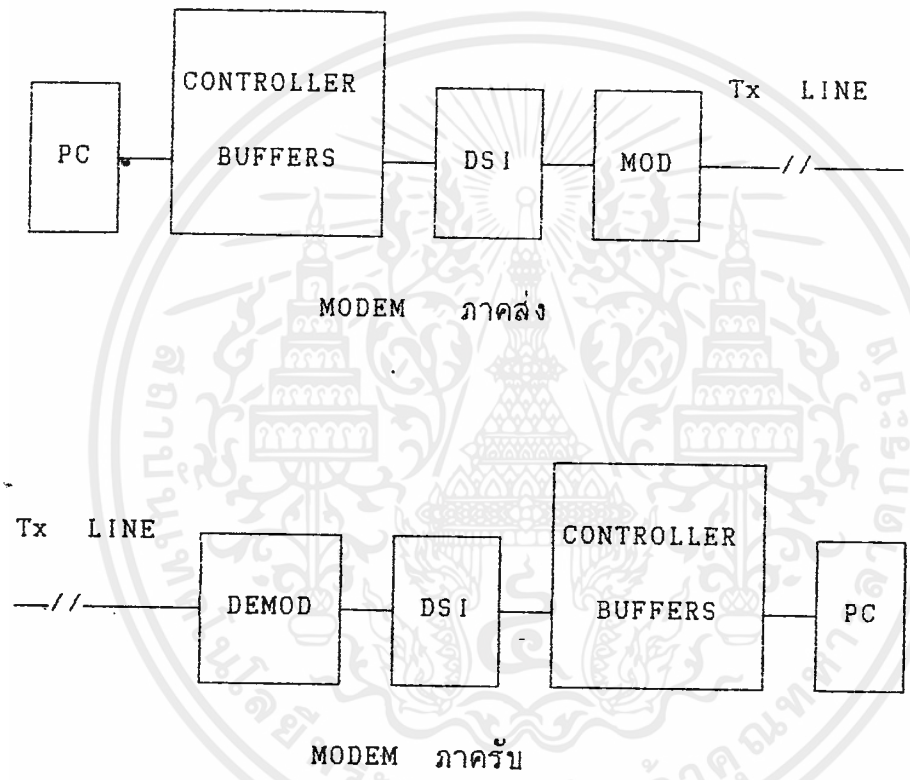
รูป แสดง frequency-to-voltage transfer characteristics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การออกแบบวงจร

สำหรับการทำงานของ optical fiber modem จะมีการทำงานดัง แผนภูมิข้างล่างนี้



การทำงานเริ่มจาก คอมพิวเตอร์ภาคส่งจะทำการส่งข้อมูลไปยัง CONTROLLER ภาคส่งนำข้อมูลที่ได้นำไปเก็บไว้ยัง BUFFER ซึ่งก็คือ RAM ของ CONTROLLER นั้นเอง หลังจากที่รับข้อมูลมาจนหมด หรือ BUFFER เต็ม ก็จะทำการส่งข้อมูลไปยัง ส่วนของ DSI ซึ่งส่วนนี้มีหน้าที่เปลี่ยนข้อมูลให้อยู่ในรูปแบบที่ต้องการส่ง เช่น ถ้าต้องการส่งข้อมูลแบบ ASYNCRONOUS ส่วนของ DSI จะทำหน้าที่ในการใส่ start bit,parity bit และ stop bit ถ้าต้องการส่งข้อมูลแบบ SYNCRONOUS ส่วนของ DSI ก็จะมีหน้าที่ในการใส่ head of block,number of block,end of block เป็นต้น เมื่อได้ข้อมูลตามรูปแบบแล้วข้อมูลก็จะผ่านไปทำการ MODULATION ในที่นี้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ MOD ทาง DIGITAL จากนั้นก็จะทำการ DRIVE ข้อมูลออกทาง TRANSMISSION LINE ซึ่งก็คือผ่านเส้นใยนำแสง เมื่อข้อมูลผ่านมาถึงภาครับ ก็จะผ่านส่วนของการ DEMOD จากนั้นก็จะนำข้อมูลผ่านไปยัง ส่วนของ DSI ภาครับ เพื่อทำการตัดสิ่งที่ไม่ใช่ ข้อมูลออก เช่น ตัด start bit, stop bit ออก จากนั้นก็นำข้อมูลผ่านไปยัง CONTROLLER ภาครับ CONTROLLER ภาครับก็จะนำข้อมูลไปเก็บไว้ที่ BUFFER เมื่อครบแล้วจึงส่งไป ยังคอมพิวเตอร์ภาครับอีกครั้งหนึ่ง

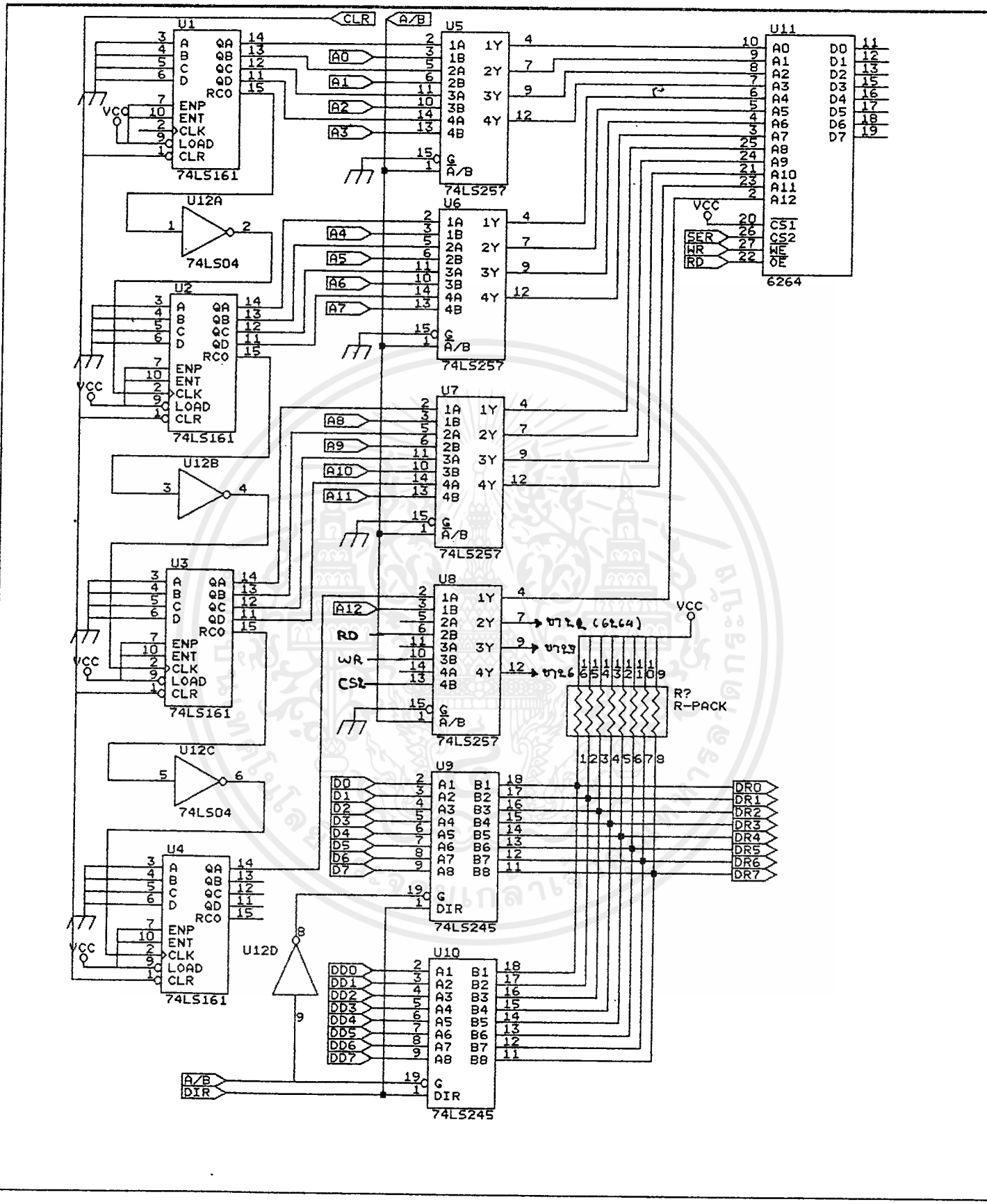
การออกแบบ CONTROLLER

การเลือกใช้ CONTROLLER นั้นสามารถเลือกใช้ CONTROLLER ได้หลายเบอร์ แต่ในที่นี้จะเลือกใช้ CONTROLLER เบอร์ Z80180 ซึ่งเป็น CONTROLLER ที่มีลักษณะของคำสั่งเหมือนของ Z80 แต่ว่าจะมีการเพิ่ม SERIAL PROT ขึ้นมา จึงทดำให้มีคำสั่งเพิ่มขึ้นมา 9 คำสั่ง เป็นคำสั่งที่เกี่ยวกับ SERIAL PORT ตามที่ได้แสดงรายระเอียดไว้ในบทที่ 1 แล้ว

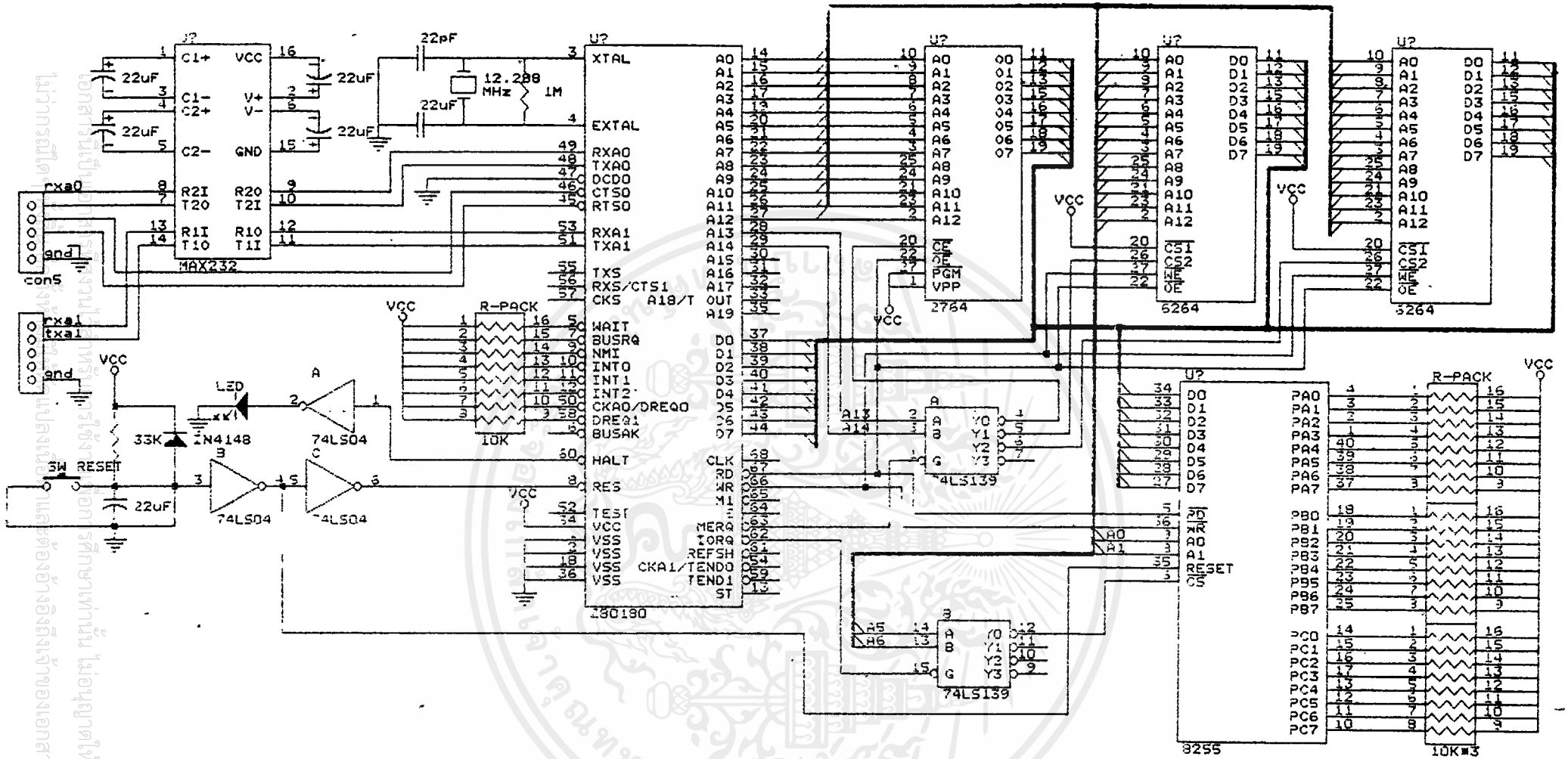
ในการออกแบบจะเป็นดังรูปข้างหลัง คือใช้ EPROM ขนาด 8 K และ RAM 8 K 2 ตัว แต่ในวงจรจริงจะมีการแก้ไขดัดแปลงเล็กน้อย คือ ในส่วนของ address bus (A_0-A_{12}), CS_2,RD,WR จะยังไม่ทำการเชื่อมเข้าที่ RAM โดยตรงแต่จะนำไปเข้า ที่ IC เบอร์ 74LS257 เป็น IC ทำหน้าที่เลือกสัญญาณ ซึ่งเข้ามา 2 CHANNEL ให้ออกมา 1 CHANNEL ตามแต่จะเลือกสัญญาณ CONTROL ว่าจะให้ CHANNEL ออก ทั้งนี้เนื่องจาก ในการติดต่อกับ RAM นั้นไม่ได้มีเพียงแต่ CONTROLLER เท่านั้น ยังมีส่วนของ DSI อีกด้วยซึ่งทั้ง 2 ส่วนนี้จะสลับกันติดต่อกับ RAM และนอกจากนี้ data bus ก็ไม่ได้ต่อโดยตรงกับ RAM เช่นกันจะนำไปผ่าน IC เบอร์ 74LS245 ทำหน้าที่ ปล่อย ข้อมูลออก หรือ LATCH ข้อมูลไว้ ซึ่งจะใช้ 2 ตัว คือตัวหนึ่งต่อกับ data bus ของ CONTROLLER และ อีกตัวหนึ่งต่อกับ data bus ของ DSI ซึ่ง 2 ตัวนี้จะสลับกันทำงานเช่นกัน นอกจากนี้ IC 74LS245 ยังเป็นตัวเลือกว่าจะเป็นการส่งข้อมูลออกหรือรับข้อมูลเข้ามาอีกด้วย ซึ่งในวงจรจะนำขาที่เลือกนี้มาต่อกับ สัญญาณ WR ของ CONTROLLER

ส่วนของ การดึงข้อมูล หรือ เขียนข้อมูลของ DSI จะมีการระบุ address โดย ใช้ COUNTER (IC เบอร์ 74LS161) เป็นตัวนับ แล้วส่งสัญญาณไปยัง RAM ก็จะสามารถระบุ address ของ RAM ได้ ทั้งนี้หมายความว่าขณะนี้เป็นการเลือกการติดต่อระหว่าง DSI กับ CONTROLLER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


MAX232
C1+
C1-
C2+
C2-
VCC
GND
V+
V-
R2I
R2O
T2O
T2I
R1I
R1O
T1O
T1I
22uF
22pF
12.288 MHz
1M
con5
rxa0
gnd
rxa1
txa1
R-PACK
10K
LED
1N4148
33K
74LS04
3W RESET
180180
XTAL
EXTAL
RXA0
TXA0
DCD0
CTS0
RTS0
RXA1
TXA1
TXS
RXS/CTS1
CKS
WAIT
BUSRQ
NMI
INT0
INT1
INT2
CKA0/DREQ0
DREQ1
BUSAK
HALT
RES
TEST
VCC
VSS
REFSH
CKA1/TEND0
TEND1
ST
A18/T
OUT
A19
RD
WR
M1
MERQ
IORQ
2764
6264
5264
74LS139
8255
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
RESET
10K*3

อธิบายวงจร

๙

ที่ 74LS257

ที่ขา A/B จะเป็นการเลือกว่าขณะนี้จะทำการติดระหว่าง CONTROLLER หรือ DSI กับ RAM

ถ้า A/B เป็น HIGH จะเป็นการเลือกการติดต่อระหว่าง CONTROLLER กับ RAM โดย INPUT PORT B จะออกที่ขา OUTPUT

ถ้า A/B เป็น LOW จะเป็นการเลือกการติดต่อระหว่าง DSI กับ RAM โดย INPUT PORT A จะออกที่ ขา OUTPUT

ที่ 74LS245

ที่ขา DIR จะต่อกับสัญญาณ WR ของ CONTROLLER

ถ้า DIR เป็น HIGH จะเป็นการรับข้อมูลมาจาก RAM

ถ้า DIR เป็น LOW จะเป็นการส่งข้อมูลไปเขียนลง RAM

ที่ขา G จะต่อกับ ขา A/B ของ 74LS257

ถ้าขา G เป็น HIGH จะเป็นการ LATCH ข้อมูลคือไม่มีการส่งข้อมูลออกหรือเข้า

ถ้าขา G เป็น LOW จะเป็นการ ENABLE ให้มีการส่งข้อมูลเข้า,ออกได้

DR0-DR7 เป็น RAM DATA BUS

DD0-DD7 เป็น DSI DATA BUS

D0-D7 เป็น CONTROLLER DATA BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

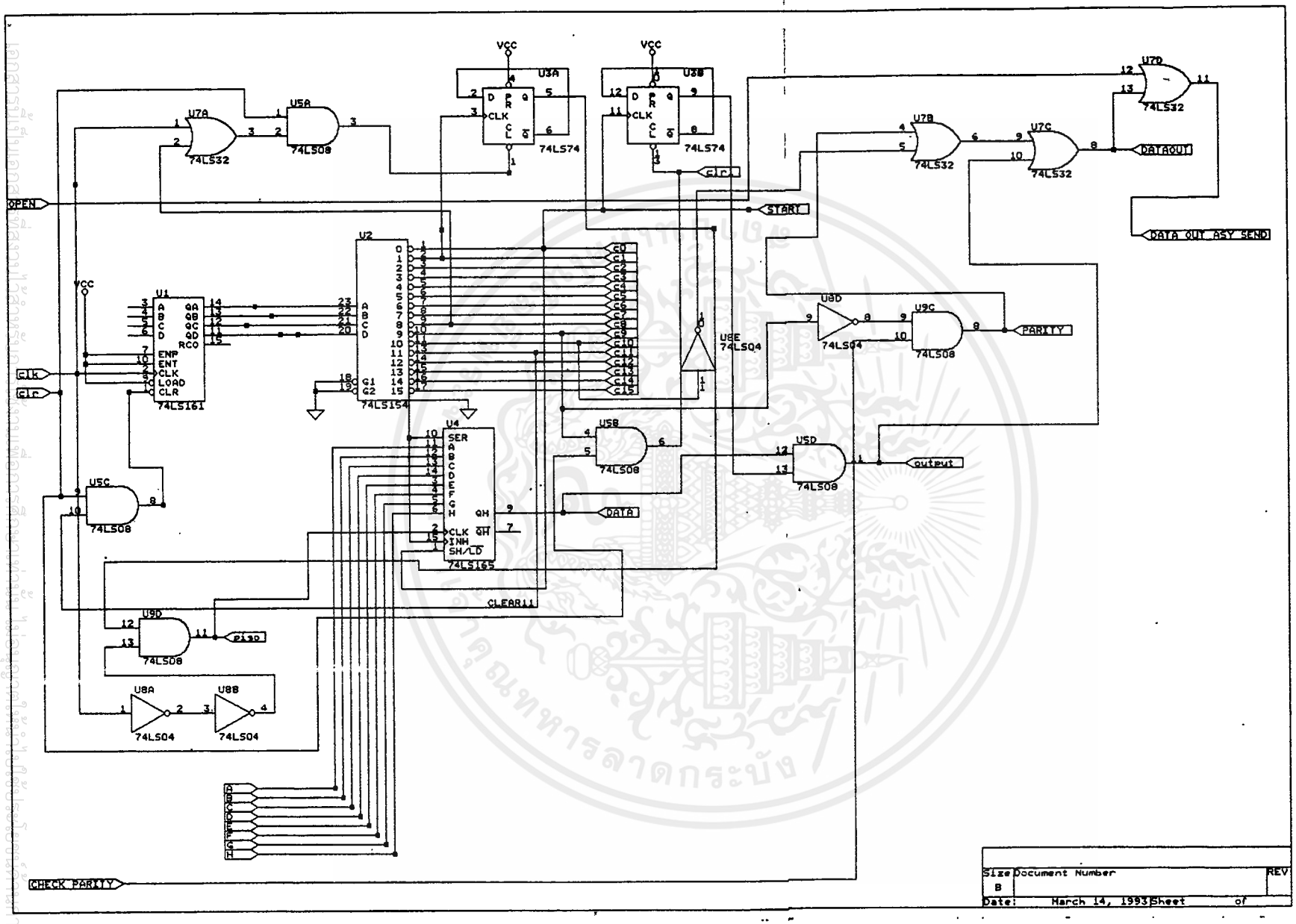
วงจร ASYNCHONOUS SEND

ในวงจรส่วน check parity จะประกอบด้วย IC เบอร์ 74LS86 , 74LS00 74LS08 และ 74LS04 ในรูปวงจร IC เบอร์ 74LS86 เป็น EXCUSIVE-OR ทำการตรวจ check ข้อมูล 8 bit ถ้าข้อมูลที่เข้า 8 bit มีจำนวน 1 เป็นจำนวนคี่ output ที่ขา 8ของ U15c จะเป็น 0 โดยเพิ่ม 74LS00 เป็นตัว set เพื่อเลือกรูปแบบในการส่งแบบ parity คู่ หรือคี่

ในส่วนของ PISO จะทำการแปลงข้อมูลจาก Pararell เป็น Serial โดยใช้ IC เบอร์ 74LS165 ทำการแปลง โดยจะต้องสร้างสัญญาณ load ข้อมูลเข้ามาก่อน คือจะสร้างมาจากส่วนของ Control

ในส่วนของ Control จะประกอบด้วยส่วนของ counter และ decoder IC เบอร์ที่เป็น gate counter จะทำหน้าที่ในการนับสัญญาณการส่งโดย เริ่มแรกจะนับ 0 และทำการส่ง start bit หลังจากนับครบ 9 ก็จะส่งสัญญาณไป control ส่วน parity bit ให้ส่ง parity bit และนับ 10 จะส่ง stop bit แล้วก็วนกลับมาทำการ laod ข้อมูลลงในส่วนของ PISO จากนั้นจึงทำการส่งเช่นเดิมจนหมดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

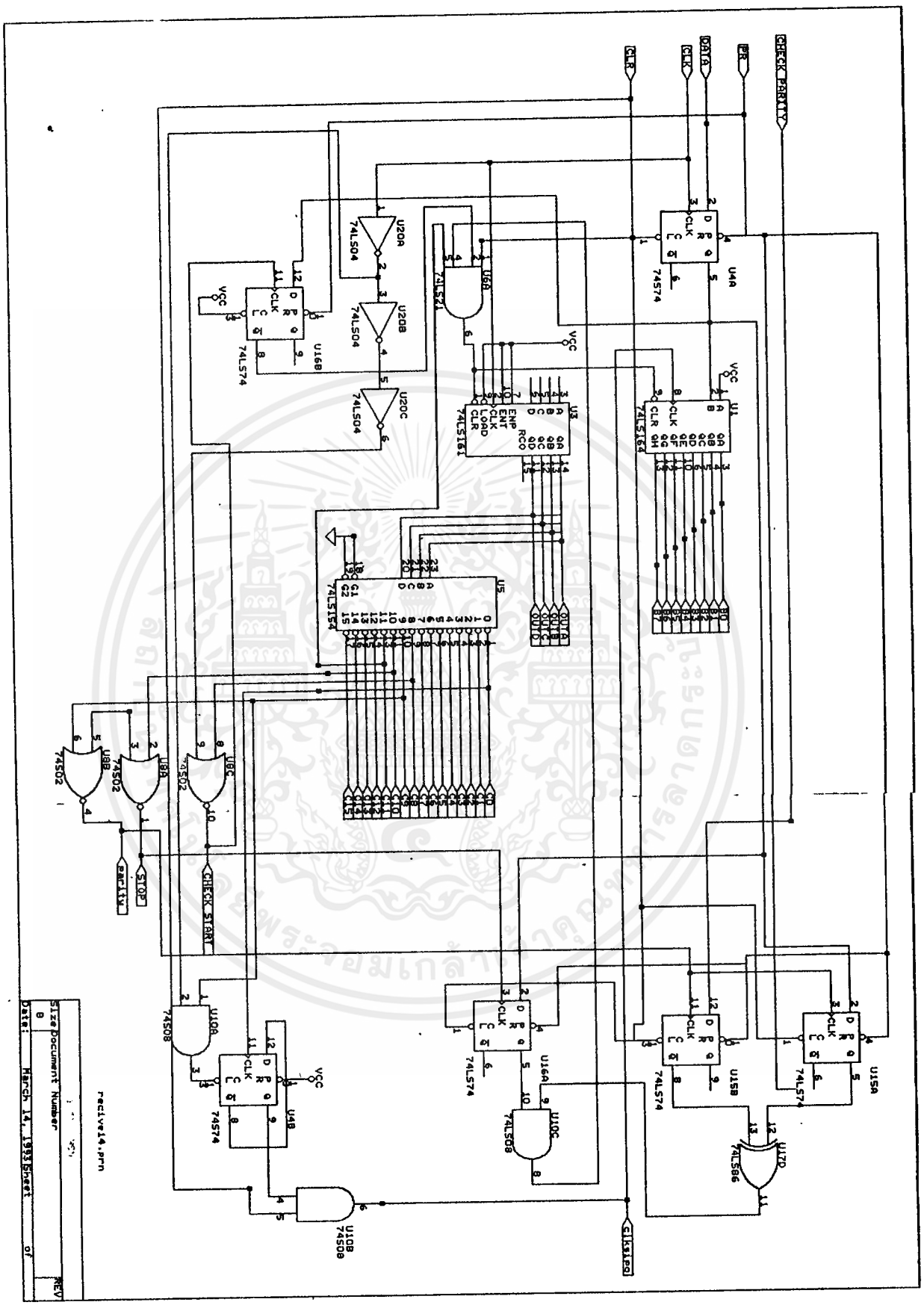

VCC
U3A
U3B
74LS74
U7A
74LS32
U5A
74LS08
U7D
U7B
U7C
DATAOUT
OPEN
START
DATA OUT ASY SEND
U2
74LS154
U1
74LS161
U8E
74LS04
U8D
U9C
PARITY
U4
74LS165
U5C
U8B
U5D
output
DATA
CLEAR11
U9D
Pin9
U8A
CHECK PARITY
Size
B
Document Number
REV
Date: March 14, 1993
Sheet
of

วงจร ASYNCHONOUS RECEIVER

ในส่วนของ check start bit จะประกอบด้วย IC เบอร์ 74LS74 ซึ่งทำการตรวจสอบสัญญาณที่ได้รับในขา 12 โดยจะมีสัญญาณจากส่วน Control มาวน check ในกรณีรับแบบ Asychornous หลังจากที่พบสัญญาณ start bit (ที่มีค่าเป็น low) ก็จะส่งสัญญาณให้ countor เริ่มนับโดย countor จะนับ 1-8 ซึ่งจะเป็นช่วงของข้อมูลซึ่งหลังจากรับ start bit ได้ ก็จะส่งสัญญาณมาที่ IC เบอร์ 74LS164 ทำการแปลงข้อมูลจาก Serial เป็น Pararellจนครบ 8 bit ส่วนของ check parity จะทำการ check parity ว่าถูกหรือไม่ หลังจากนั้นก็จะ check stop bit ในกรณีที่ มีการผิดพลาดของ parity หรือ stop bit ก็จะส่งสัญญาณ ERROR มาที่ Port 8255 ซึ่ง Port 8255 นี้ไมโครโปรเซสเซอร์ ทำการตรวจสอบในระหว่างการรับ

หลังจากที่แปลงครบ 8 บิทข้อมูล แล้วตรวจพบ parity bit , stop bit เรียบร้อยส่วน Control ก็จะทำการเก็บข้อมูลลงใน RAM โดยส่งข้อมูล 8 bit และทำการ ENABLE ในส่วนของการที่ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYNCHRONOUSE RECIVE

อธิบายตามวงจร

1) ส่วนการตรวจสอบสัญญาณ HEAD OF BLOCK จะประกอบด้วยส่วยของ IC เบอร์ 74LS273 เป็น D F/F 8 ตัว กับ 74LS30 ซึ่งจะทำการตรวจสอบสัญญาณ ที่รับเข้ามา ซึ่งถ้าตรงกับสัญญาณ HEAD OF BLOCK ก็จะส่งสัญญาณบอกส่วน CONTROL ให้ทำการรับข้อมูล

2) ส่วนการแปลงข้อมูลประกอบด้วย IC เบอร์ 74LS164 74LS11 ทำหน้าที่ในการแปลงข้อมูลจาก SERIAL เป็น PARARELL

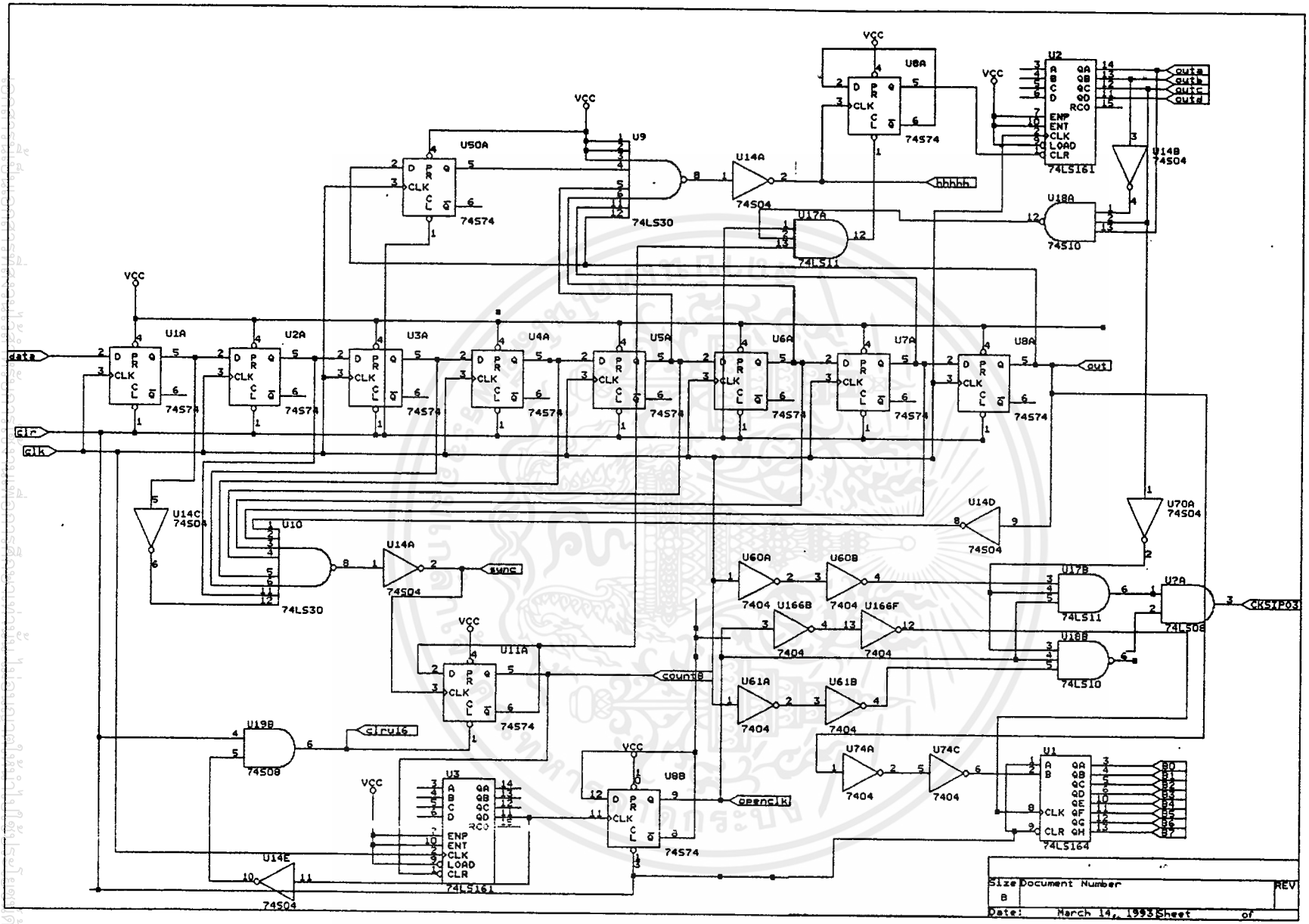
3) ส่วนตรวจสอบ NUMBER OF BLOCK จะเป็นส่วนที่ทำการตรวจสอบ เปรียบเทียบกับ BLOCK ที่ได้รับในวงจรประกอบด้วย 74LS169 เป็นตัวนับ และ IC เบอร์ 74LS08 ทำการตรวจสอบสัญญาณ ลำดับ BLOCK ข้อมูล และ D/FLIP FLOP

4) END OF BLOCK ประกอบด้วย IC เบอร์ 161 กับ GATE แบบเดียวกับข้อ 3) แต่จะทำการตรวจสอบ ในตำแหน่ง END OF BLOCK

5) ANTI BIT SUFFING เป็นการตรวจสอบสัญญาณข้อมูลที่ได้รับ ว่าจะพบ 1 ติดกัน 5 ตัวหรือไม่ แล้วทำการถอด 0 ที่ตามหลัง 1 มาออก แล้วจะต่อไปส่วนเปลี่ยนแปลงสัญญาณจาก SERIAL เป็น PARARELL

6) ส่วน CONTROL ประกอบด้วย IC ที่เป็น GATE ต่างๆ D-FLIP FLOP COUNTER MULTIPLEXER ทำการสร้างสัญญาณควบคุมด้านต่างๆ ให้ทำงานได้ครบขั้นตอนการตรวจเช็คข้อมูล และทำการเขียนข้อมูลลงในหน่วยความจำด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
U8A
74S74
U2
74LS161
U14B
74S04
U50A
U9
74LS30
U14A
74S04
hhhhh
U17A
74LS11
U18A
74S10
quta
qutb
qutc
qutd
U1A
U2A
U3A
U4A
U5A
U6A
U7A
U8A
data
clr
clk
qut
U14C
74S04
U10
U14A
74S04
sync
74LS30
U14D
74S04
U70A
74S04
U60A
U60B
7404
U166B
U166F
U17B
74LS11
U2A
74LS08
CKSTP03
U18B
74LS10
U11A
count8
U61A
U61B
U19B
74S08
clrv16
U74A
U74C
U1
74LS164
U3
74LS161
U14E
74S04
U8B
gensclk
Size
Document Number
REV
Date: March 14, 1993
Sheet
of

SYNCHRONOUSE SEND

อธิบายตามวงจร

1) ส่วนของ PISO (PARARELL INPUT SERIAL OUTPUT) ใช้ IC เบอร์ 74LS165 เป็นส่วนที่ทำการ LOAD ข้อมูลจาก MEMORY โดยตรงมาทำการเก็บไว้แล้วทำการแปลง โดยจะเริ่มแปลงหลัง การได้รับสัญญาณจากส่วน CONTROL โดยจะแปลง เป็นแบบอนุกรม ส่งผ่านตลอดเวลา

2) HEAD OF BLOCK ประกอบไปด้วย IC 74LS165 ทำการส่งสัญญาณ HEAD OF BLOCK มีค่าเป็น 01111110 โดยจะทำงานเมื่อได้รับสัญญาณมา ENABLE จากส่วน CONTROL

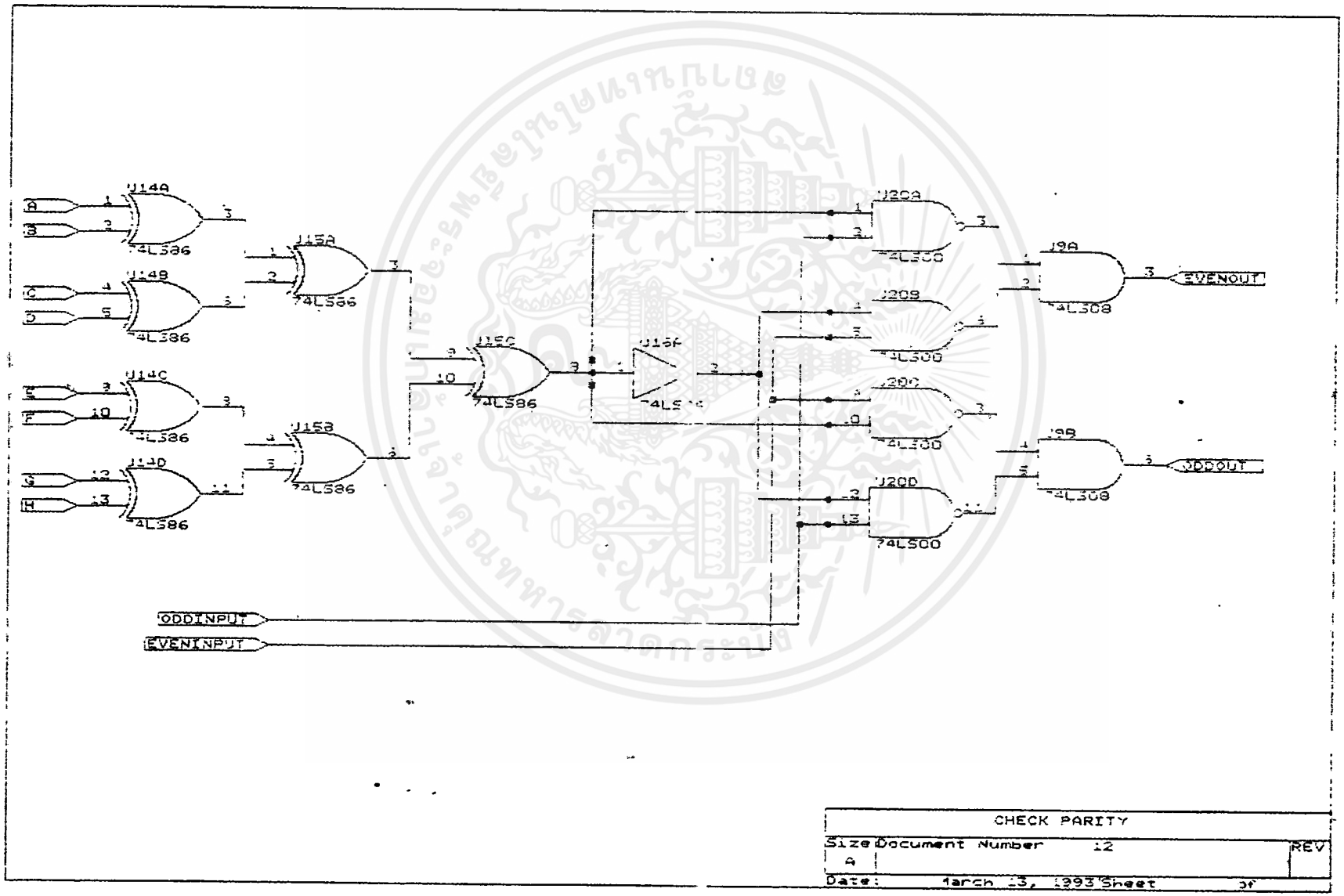
3) END OF BLOCK เป็นส่วนของการสร้างสัญญาณปิดด้วย BLOCK โดยจะทำงานหลังจากส่งครบ 1 BLOCK ข้อมูล

4) NUMBER OF BLOCK เป็นส่วนที่ประกอบด้วย IC 74LS161 ซึ่งเป็น CONTROL ทำหน้าที่หลัก ในการนับ BLOCK ข้อมูลในการส่งโดยจะส่งแทรก ต่อจาก HEAD OF BLOCK

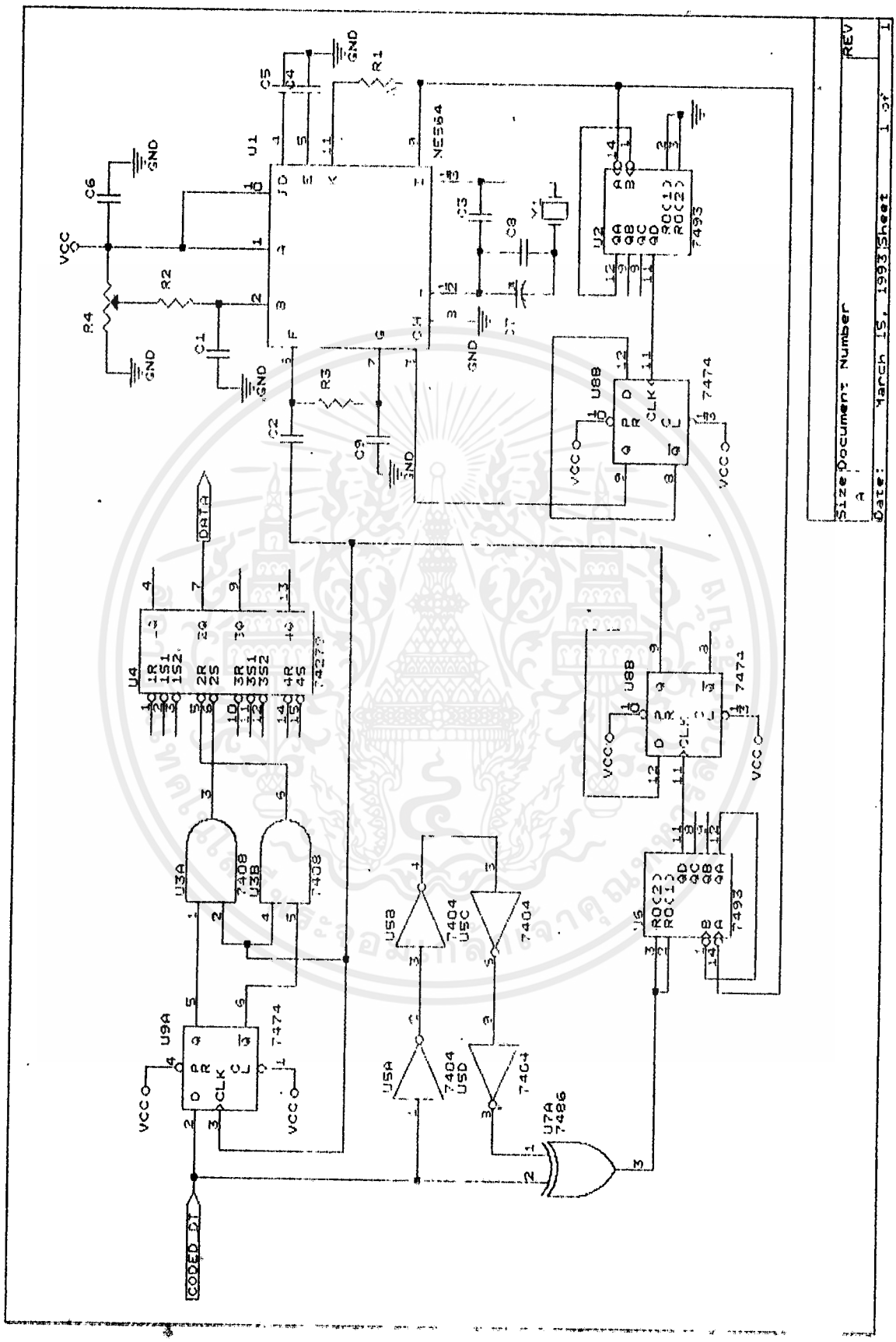
5) BIT SUFFING เป็นส่วนที่ประกอบด้วย D/FLIP FLOP และ NAND GATE 8 INPUT ซึ่งทำการตรวจสอบสายสัญญาณข้อมูลที่เป็น SERIAL ตลอด และเมื่อพบสัญญาณข้อมูลที่มี 1 ติดกัน 5 ตัว จะทำการแทรกข้อมูลพร้อมกับ BREAK การส่งสัญญาณนาฬิกา

6) ส่วน CONTROL จะประกอบด้วย IC เบอร์ 74LS00 74LS02 74S04 74LS04 74LS08 74S08 74LS32 74LS74 74LS11 74LS174 74LS273 74LS241 ทำหน้าที่แตกต่างกันไปตามคุณสมบัติ เพื่อสร้างสัญญาณในการส่ง โดยจะสร้างสัญญาณ CONTROL ตามฐานเวลาที่กำหนดให้ โดยสามารถที่จะสั่งงานโดยไมโครโปรเซสเซอร์ผ่าน PORT 8255

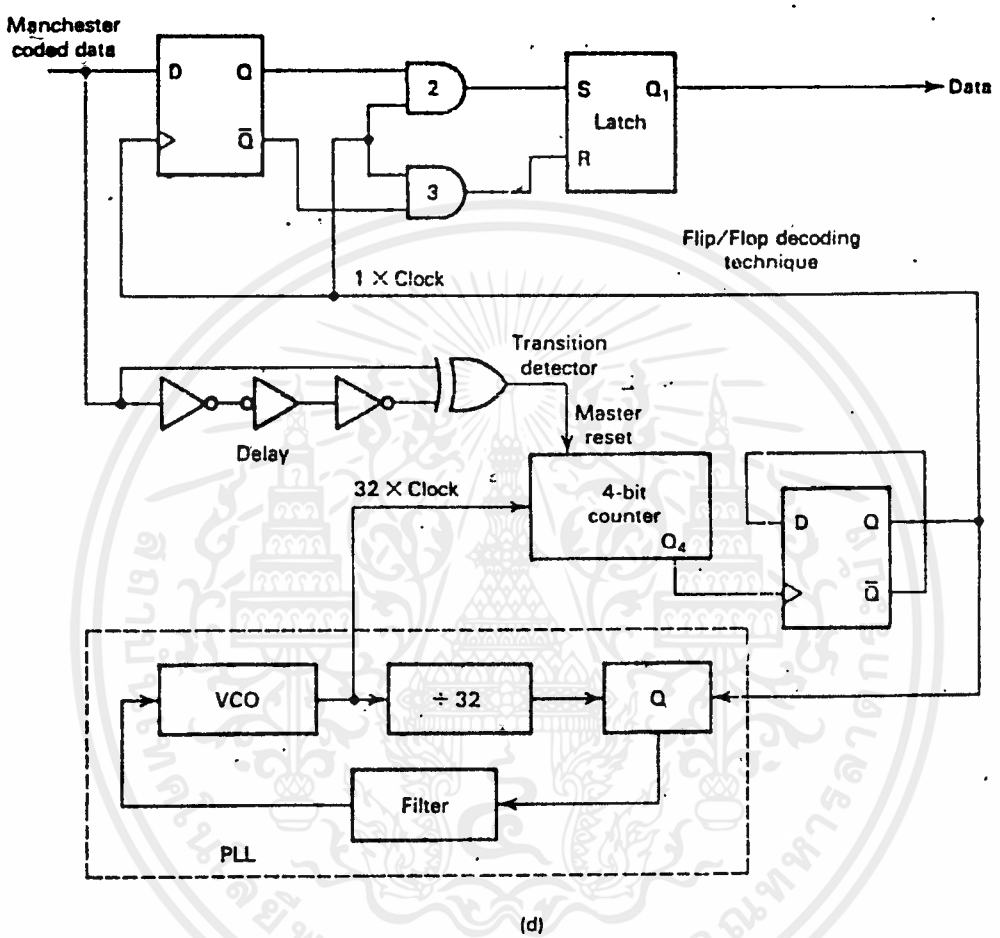
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


A
B
C
D
E
F
G
H
U14A
U14B
U14C
U14D
74LS86
U15A
U15B
U15C
U15F
74LS14
U20A
U20B
U20C
U20D
74LS00
U9A
U9B
74LS08
EVENOUT
ODDOUT
ODDINPUT
EVENINPUT
CHECK PARITY
Size
A
Document Number
12
REV
Date:
March 13, 1993
Sheet
of

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

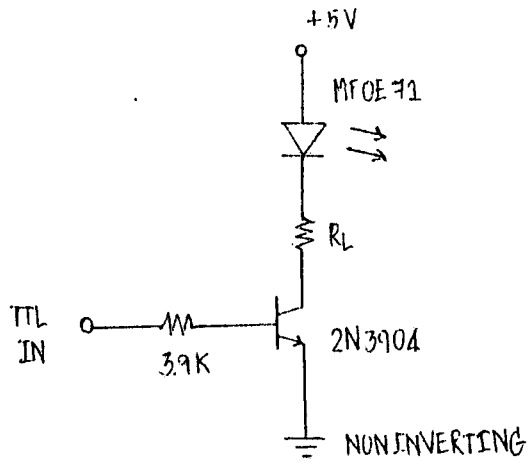


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

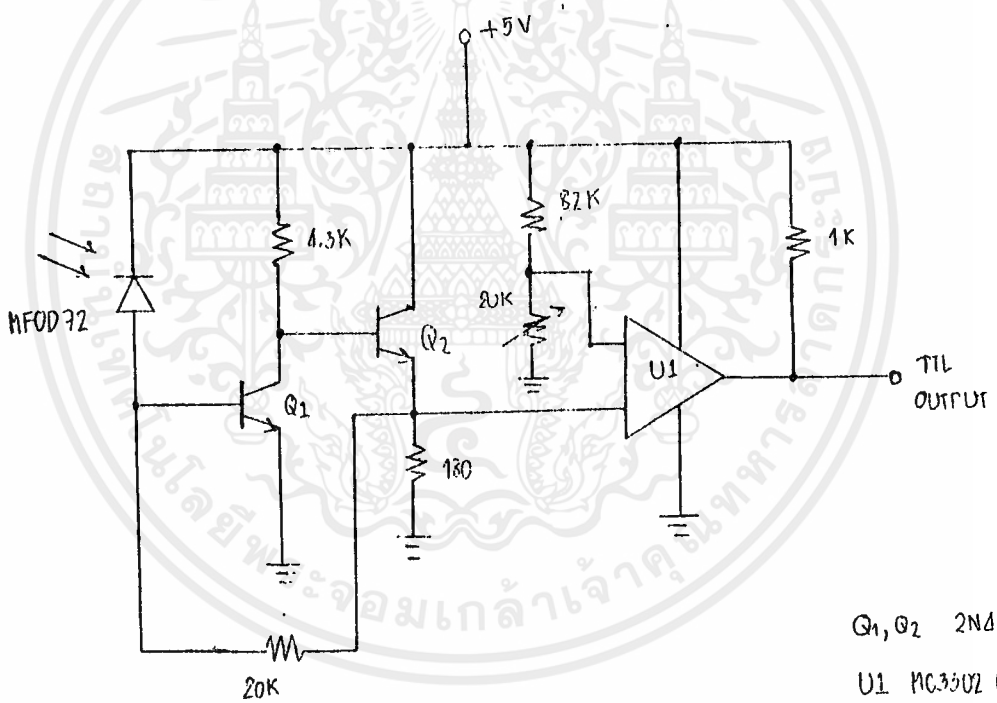


(d)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TTL TRANSMITTERS



Q_1, Q_2 2N4401

U1 MC3302 (1/4)

TTL RECEIVERS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ HARDWARE นี้เป็นการนำเอาข้อมูลที่ได้จากส่วนของ DSI มาทำการส่งและรับโดยประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

1. สัญญาณข้อมูลที่ได้จากส่วน DSI จะนำมาเข้า EXCLUSIVE- OR กับ clock ซึ่งมีความถี่ 600KHz เพื่อให้ข้อมูลเปลี่ยนไปอยู่ในรูปของ รหัสแมนเชสเตอร์ แล้วจึงทำการส่งต่อไปอีกที

2. TTL TRANSMITTER จะนำสัญญาณแมนเชสเตอร์มาแปลงให้อยู่ในรูปของสัญญาณแสงโดยตัวกำเนิดแสง MFOE71 เพื่อส่งผ่านสาย FIBER OPTIC

3. TTL RECEIVER จะทำการแปลงสัญญาณแสงที่ส่งผ่านสาย FIBER OPTIC มาเป็นสัญญาณทางไฟฟ้า โดยตัว PHOTO DETECTOR MFOD72

4. จากนั้นภาครับ จะเปลี่ยนสัญญาณแมนเชสเตอร์ ให้กลับออกมาเป็น DATA เดียวกันกับที่ส่งมา โดยส่วนต่างๆ ดังนี้

- PHASE LOCK LOOP จะทำหน้าที่ผลิตความถี่ โดยค่าความถี่นี้เราสามารถเปลี่ยนได้จากค่า Cs ในรูป

- เมื่อสัญญาณที่ผลิตได้ไปเข้า countor ก็จะมีสัญญาณที่ออกจาก EXCLUSIVE-OR มาเป็นตัว RESET ให้ทำการนับ และเมื่อสัญญาณที่นับแล้วนี้ไปผ่าน D-FLIP FLOP ซึ่งทำหน้าที่หารความถี่เราก็จะได้ clock ออกมาตามที่เราต้องการ

- clock ที่ได้ก็จะมา TRIG สัญญาณแมนเชสเตอร์ สัญญาณที่ถูก TRIG ออกมาก็จะมา AND กับ clock ที่ถูก DETECT ได้อีกทีนึง แล้วสัญญาณที่ออกจาก AND ทั้ง 2 ตัวจะถูก LATCH ค่าไว้ โดย S-R FLIP FLOP ทำให้ได้ DATA ออกมาตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ ๘

การทดสอบ และ ผลการทดสอบ

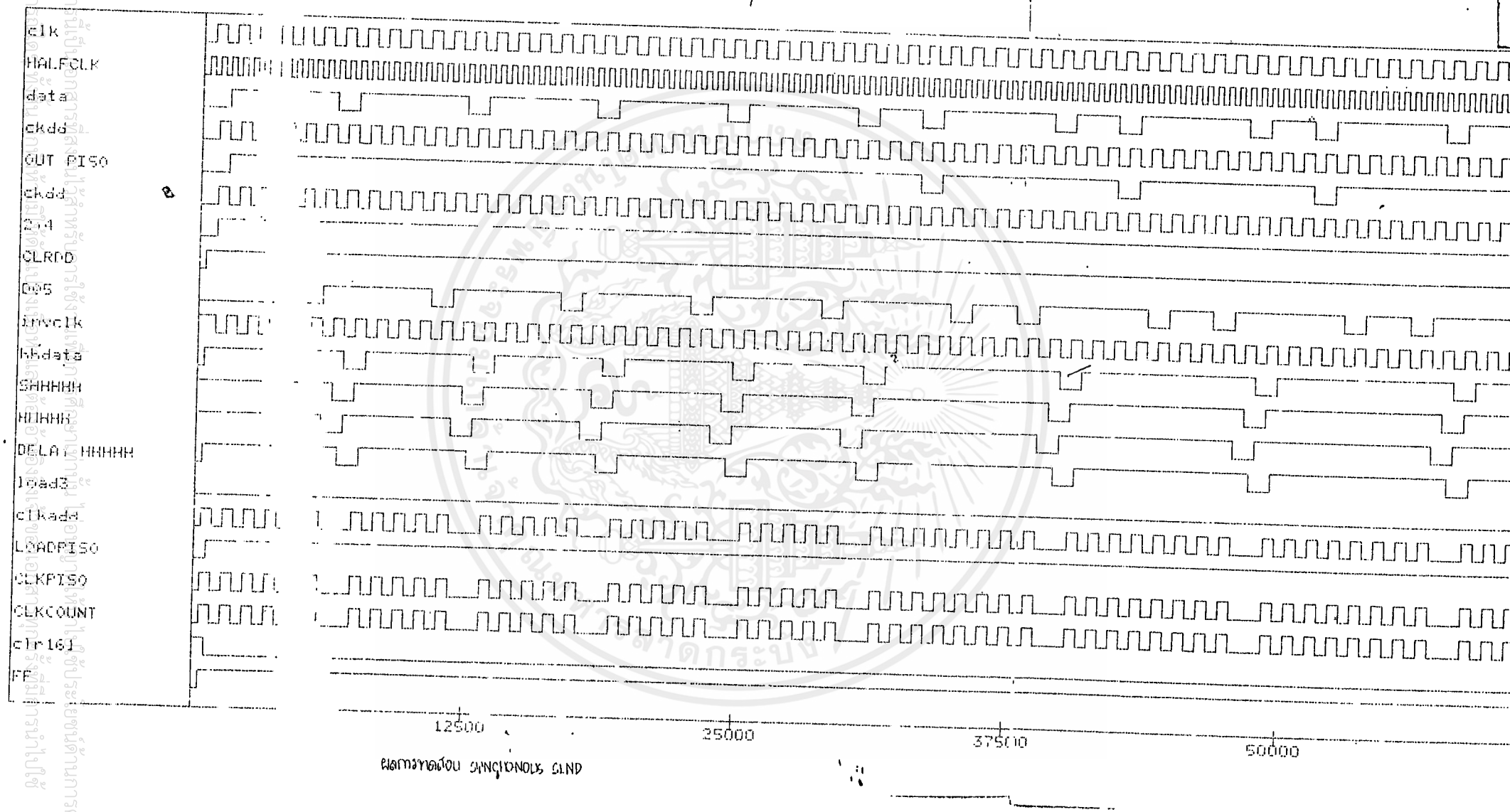
ในการทดลองการสร้างวงจร DSI (DATASET INTERFACE) การสร้างจะทำโดยนำอุปกรณ์มาต่อเข้ากันเลย(DATASET INTERFACE) นั้นมักมีปัญหา ในส่วนตัวเนื่องจากข้อจำกัดของอุปกรณ์ ดังนั้นในการสร้างจริง ในส่วนของโครงงานนี้ได้นำ SOFTWARE ซึ่งเป็น SOFTWARE สำเร็จรูปใช้ในการออกแบบ แต่มีข้อจำกัดของ SOFTWARE ตัวที่ออกแบบนั้นเป็นตัวที่ใช้เพื่อการศึกษา ไม่ใช่ SOFTWARE ตัวจริง จึงไม่สามารถที่จะทำการ SIMULATE ได้ครบทุกวงจร และข้อจำกัดของ SOFTWARE ทาง DIGITAL ทำให้ไม่สามารถ SIMULATE วงจรทาง ANALOG ดังนั้นการ SIMULATE วงจรจริงจริง จำเป็นต้องใช้ IC ประเภท TTL และบางเบอร์ FILE ที่มีผลในการ SIMULATE นั้นไม่สามารถใช้ได้ จำเป็นต้องใช้ตัวแทน ดังนั้นวงจรจริงจริงสามารถจะปรับแต่งได้

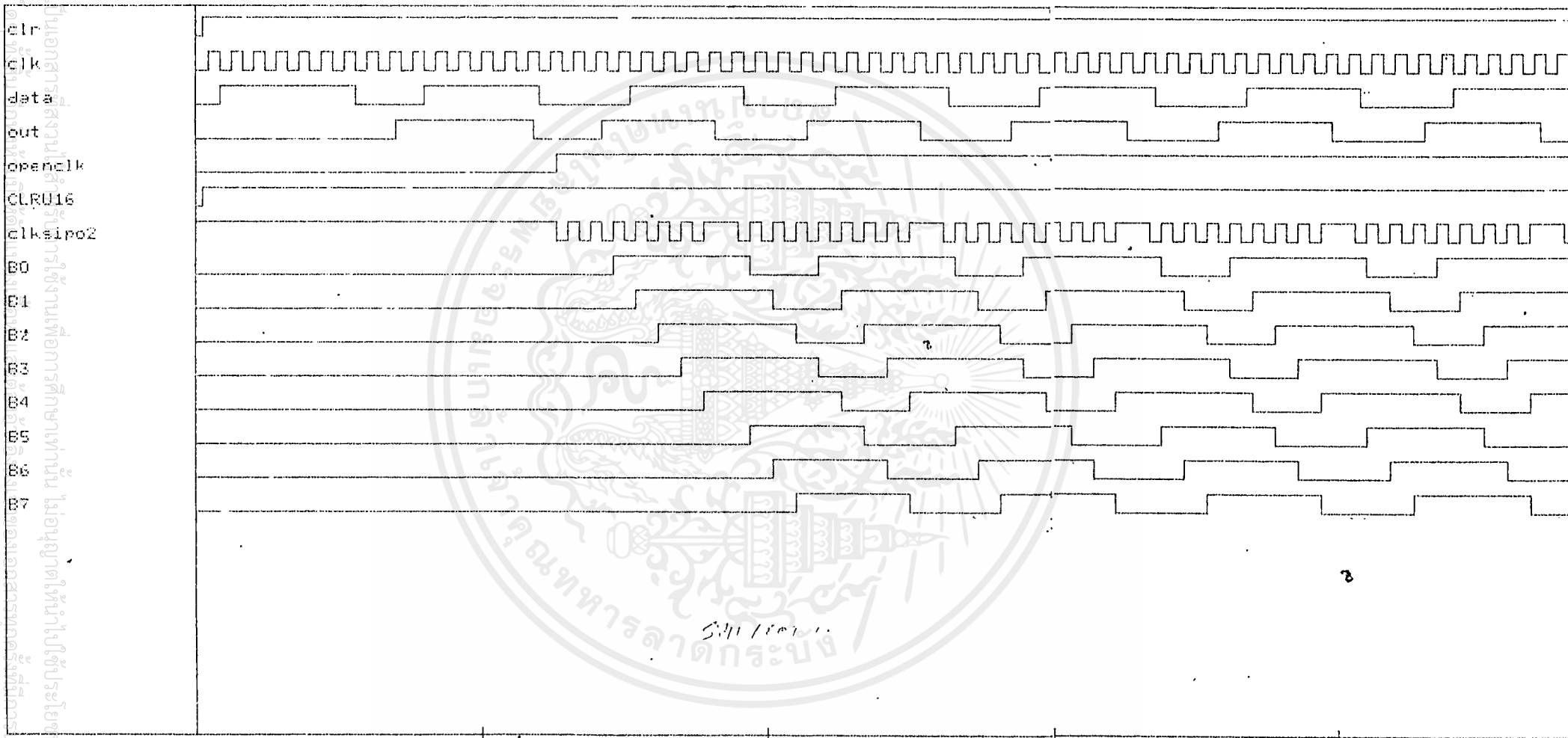
ในโครงงานฉบับนี้ในส่วนการทดลองในส่วนวงจร TTL นั้นเราทำการ SIMULATE แล้ว ได้ผลที่ค่อนข้างละเอียด ซึ่งในรายงานฉบับนี้ได้นำผลการ SIMULATE มาใส่ในส่วนท้ายของการทดลอง

ผลการทดสอบ

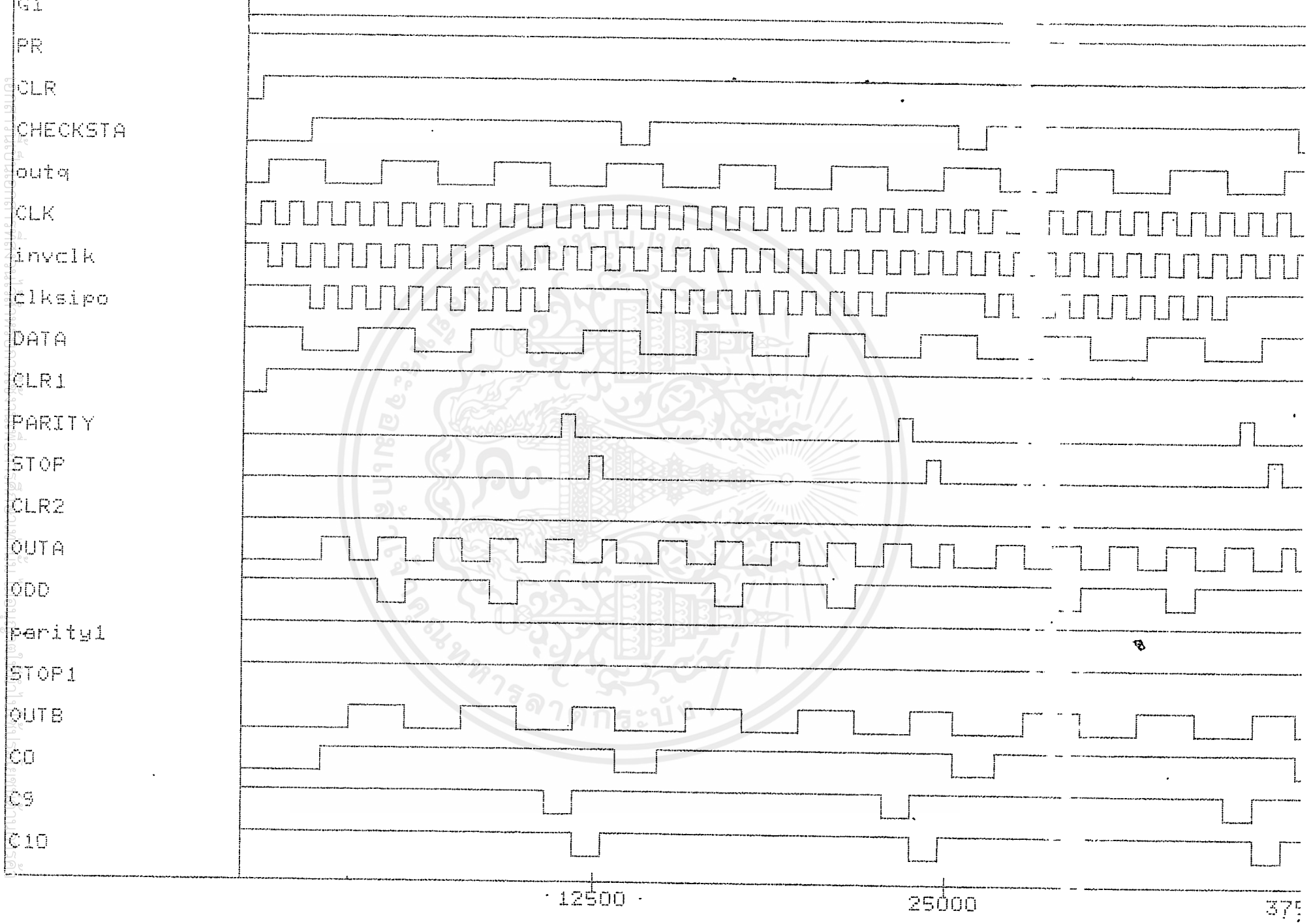
ในส่วนของวงจรสามารถศึกษาได้จาก ส่วนของวงจรควบคู่ไปกับ TIME DIAGRAM ซึ่ง INPUT OUTPUT กำหนดได้ดังในวงจร และสัญญาณที่ได้ จะมีลักษณะเช่นเดียวกับที่ได้กับการต่อจริง ๆ ในวงจรจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

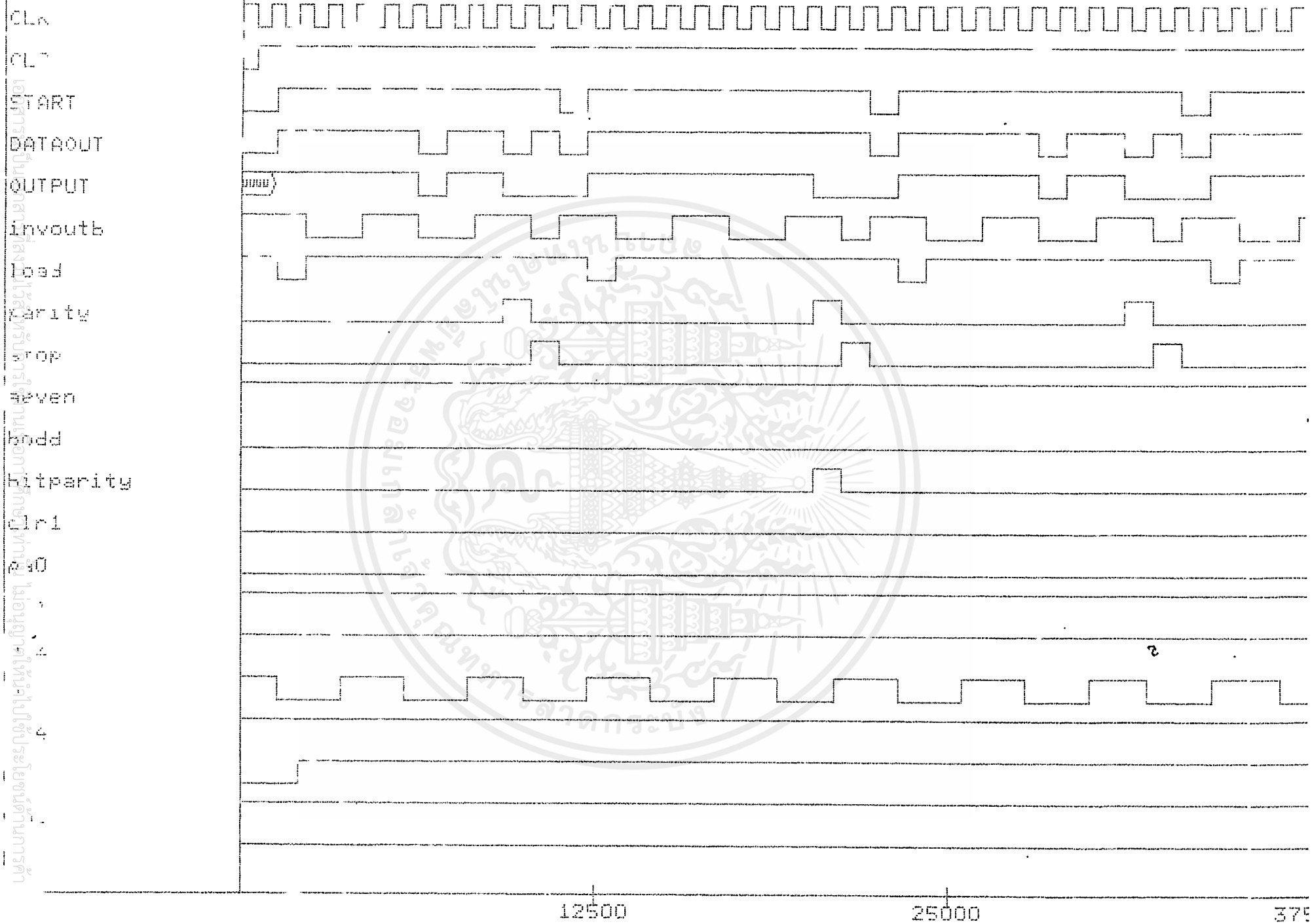

clk
HALFCLK
data
ckdd
OUT PISO
ckdd
Q4
CLRDD
D05
invclk
hkdata
SHHHHH
HHHHH
DELAY HHHHH
load3
clkadd
LOADPISO
CLKPISO
CLKCOUNT
clr161
FF
12500
25000
37500
50000



ผลการจากการใช้งาน SYNCHONOUS RECIVE



ผลการจำลองการทำงาน ASYCHONOUS RECEIVE



ผลการทดสอบ ASYNCHONOUS SEND

สรุปผลการทดลอง

การทดลองใช้ SOFTWARE ORCAD VST (VERIFICATION SIMULATION TOOLS) ดังนั้นผลที่ได้ จะมีความละเอียดเท่ากับการต่อวงจรจริง เพราะแต่ละตัวมีคุณสมบัติตามที่ระบุไว้ในอุปกรณ์ ดังนั้นวงจรที่ทำการ SIMULATE ด้วย SOFTWARE ORCAD VTS จะสามารถใช้งานได้จริง สำหรับการใช้งานนั้น สามารถศึกษาได้จากคู่มือการใช้ในส่วนหนังสืออ้างอิง

การทดลองในส่วน ภาครับ นั้น ได้มีการจำลองข้อมูล มาทำการส่งเข้าภาครับ ได้ผลเป็นดังกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากส่วน HARDWARE ที่กล่าวมาทั้งหมดแล้ว ยังมีส่วนของ SOFTWARE ที่ใช้ในการ PROGRAM CONTROLLER ให้ทำงานตามต้องการ ในที่นี้จะกล่าวถึง SOFTWARE อย่างง่ายๆ สามารถใช้ส่งและรับข้อมูลได้ แต่ว่ายังไม่มีการตรวจสอบการผิดผลาดใดๆ จะแบ่งได้เป็น 4 ส่วนคือ ส่วนแรก จะเป็น SOFTWARE ที่ใช้ส่งข้อมูลจาก คอมพิวเตอร์ไปยัง CONTROLLER โดยจะถือว่าทางฝ่ายรับพร้อมที่จะรับข้อมูลอยู่ตลอดเวลา ภาษาที่ใช้เขียนคือภาษา ASSEMBLY 8088 PROGRAM มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov   ah,readseq
            int   doscall
;print record from disk tranfer area
            mov   cx,80h
            mov   bx,0h
hold:       mov   dx,serial+5
            in    al,dx
            test  al,20h
            jz    hold
            mov   dx,serial
            mov   al,[dta+bx]
            out   dx,al
            cmp   al,eof
            je    done
            mov   ah,1        ;check if keyboard had been pressed
            int   16h
            jnz   keypress
11:         inc   bx
            loop  hold
            jmp   read_rec
;check keyboard interupt
keypress:   mov   ah,0
            int   16h
            cmp   al,3
            je    done
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            jmp    l1
done:       ret
main        endp
code_seg    end
            end
```

ให้ PROGRAM นี้ชื่อ SEND.ASM หลังจากที่ COMPILE จนได้เป็น SEND.EXE แล้ว เมื่อต้องการจะทำการส่งข้อมูลออกทาง SERIAL PORT สมมุติจะส่ง FILE ชื่อ KID.DAT ก็ให้ พิมพ์ดังนี้ SEND KID.DAT ข้อมูลก็จะส่งออกไปทาง SERIAL PORT ในที่นี้จะส่งออกทาง COM2

PROGRAM รับข้อมูลจาก CONTROLLER มาเข้าไปเก็บไว้ใน คอมพิวเตอร์

```
serial      equ   02f8h
doscall     equ    21h
display     equ    02h
create      equ    16h
writesq     equ    15h
close       equ    10h
fcb         equ    5ch
return      equ    0dh
lfeed       equ    0ah
eof         equ    1ah
data_seg    segment
     org    7ch
recno       db     ?
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        org     80h
dta             db      80hdup(?)
dtat_seg        ends
code_seg        segment
main            proc    far
        assume          cs:code_seg,ds:data_seg
;set up stack for return
                push    ds
                sub     ax,ax
                push    ax
;*********************
;  set serial port *
;*********************
                mov     dx,serial+3     ;control register(3fb)
                mov     al,80h
                out     dx,al
                mov     ax,12           ;divisor for 9600 baud
                mov     dx,serial
                out     dx,al
                mov     al,ah
                mov     dx,serial+1
                out     dx,al
                mov     dx,serial+3
                mov     al,00000011b    ;no parity 8 bit character
                out     dx,al
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;create disk file ,set recno number to 0
            mov   dx,fcb
            mov   ah,create
            int   doscall
            mov   recno,0
;blank buffer by filling with returns
newline:    mov   bx,0
            mov   cx,80h
erase:      mov   [bx+dta],return
            inc   bx
            loop erase
;get line
            mov   cx,78h
            mov   bx,0
recv:       mov   dx,serial+5
            in    al,dx
            test  al,1
            jz    recv
            mov   dxserial
            in    al,dx
            mov   [dta+bx],al
            cmp   eof
            je    exit
            mov   dx,al
            mov   ah,display
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                int     doscall
                inc     bx
                loop    recv
                mov     [dta+bx],lfeed
                inc     bx
                mov     [dta+bx],return
;write record to file with sequential mode
                mov     dx,fcb
                mov     ah,writesq
                int     doscall
                mov     ah,1
                int     16h
                jnz     press
                jmp     newline
;close file and exit
exit:           mov     dx,fcb
                mov     ah,writesq
                int     doscall
                mov     dx,fcb
                mov     ah,close
                int     doscall
press:          ret
main            endp
code_seg        ends
                end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ CONTROLLER ฝ่ายส่ง

```
        cpu "64180.tbl"
        hof "int8"
        org 0
cntla0: equ   0h
cntla1: equ   1h
cntlb0: equ   2h
cntlb1: equ   3h
stat0:  equ   4h
stat1:  equ   5h
tdr0:   equ   6h
tdr1:   equ   7h
rdr0:   equ   8h
rdr1:   equ   9h
il:     equ   33h
stack:  equ   0ff00h
eof:    equ   1ah
pctrl:  equ   0c083h
pdata:  equ   0c080h
        ld    bc,pctrl
        ld    a,80h
        out   (c),a
        ld    a,veclow
        out0  (il),a
        ld    a,64h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม ที่ CONTROLLER ฝ่ายรับ ภาษาที่ใช้คือ Z80180

```
        cpu "64180.tbl"

        hof "int8"

        org  0


cntla0:     equ   0h

cntla1:     equ   1h

cntlb0:     equ   2h

cntlb1:     equ   3h

stat0:      equ   4h

stat1:      equ   5h

tdr0:       equ   6h

tdr1:       equ   7h

rdr0:       equ   8h

rdr1:       equ   9h

il:         equ   33h

stack:      equ   0ff00h

lf:         equ   0ah

cr:         equ   0dh

eof:        equ   1ah

pctrl:      equ   0c083h

pdata:      equ   0c080h

            ld    bc,pctrl

            ld    a,80h

            out   (c),a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ld      a,veclow
        out0    (il),a
        ld      a,64h
        out0    (cntla0),a
        out0    (cntla1),a
        ld      a,2
        out0    (cntlb0),a
        out0    (cntlb1),a
        ld      a,8
        out0    (stat0),a
        out0    (stat1),a
        ld      hl,8000h
loop:   in0     e,(stat1)
        bit     7,e
        jr      z,loop
        in0     a,(rdr1)
        ld      (hl),a
        inc     hl
        jr      loop
ends:   end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        out0    (cntal0),a
        out0    (cntal1),a
        ld      a,2
        out0    (cntlb0),a
        out0    (cntlb1),a
        ld      a,8
        out0    (stat0),a
        out0    (stat1),a
        ld      hl,8000h
loop:   in0     e,(stat0)
        bit     7,e
        jr      z,loop
        ld      a,(hl)
        cp      eof
        jr      z,eds
        out0    (tdro),a
        inc     hl
        jr      loop
eds:    in0     e,(stst0)
        bit     1,e
        jr      z,eds
        ld      a,eof
        out0    (tdr0),a
        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณอย่างสูงต่อ อาจารย์สมศักดิ์ เชียร์ศิริกุล ในฐานะอาจารย์ที่ปรึกษา และ คุณเกียรติศักดิ์ ศรีพิมานวัฒน์ ที่ได้ช่วยให้คำปรึกษาและคำแนะนำในระหว่างการทำโครงงานและปริญญานิพนธ์ฉบับนี้ตั้งแต่ต้นจนจบการศึกษาให้สำเร็จลุล่วงไปด้วยดี รวมทั้งเพื่อนนักศึกษาภาควิชาอิเลคทรอนิคส์ทุกคนที่ให้ความช่วยเหลือด้วยดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้