

ปริญญาโท ปีการศึกษา 2531

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง CENTRONIC PRINTER BUFFER

ผู้จัดทำ

1. นายสุธีร์ กิจฉวี

เลขประจำตัว 28.6723



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023126

CENTRONIC PRINTER BUFFER

สุธีร์ กิจฉวี

รศ.กิตติ ศีรเศรษฐ อาจารย์ที่ปรึกษา

ปีการศึกษา 2531

บทคัดย่อ

ปริญญาโทฉบับนี้ เรียบเรียงขึ้นจากโครงงานที่ได้พัฒนาขึ้นเพื่อเป็นเครื่องมือช่วยเพิ่มประสิทธิภาพการทำงานให้กับเครื่องคอมพิวเตอร์ ในการติดต่อกับอุปกรณ์ที่มีความเร็วในการทำงานต่ำกว่าเครื่องพิมพ์ ซึ่งปกติแล้วคอมพิวเตอร์ต้องคอยให้เครื่องพิมพ์ทำงานเสร็จก่อนจึงจะไปทำงานอย่างอื่นหรือส่งข้อมูลใหม่ได้ ซึ่งทำให้ประสิทธิภาพของคอมพิวเตอร์ลดลง CENTRONIC PRINTER BUFFER จะรับข้อมูลจากคอมพิวเตอร์ด้วยความเร็วสูงพอ ๆ กับคอมพิวเตอร์ แล้วค่อยทยอยส่งข้อมูลให้กับเครื่องพิมพ์ในความเร็วที่เครื่องพิมพ์ทำงานได้ทันปกติ ซึ่งจะทำให้คอมพิวเตอร์ไม่ต้องรอเครื่องพิมพ์อีกเพราะมี CENTRONIC PRINTER BUFFER นี้ทำหน้าที่รับช่วงแทนในการส่งข้อมูลให้กับเครื่องพิมพ์ และโดยการใช้โปรแกรมในส่วน RING BUFFER หรือ CIRCULAR QUEUE ซึ่งถือเป็นหัวใจของโครงงานนี้ ในการจัดเก็บข้อมูลในหน่วยความจำ โดยการใช้ตัวชี้ (POINTER) 2 ตัว ทำให้สามารถรับข้อมูลมาแทนที่ส่วนของข้อมูลที่ถูกรับไปแล้ว และยังช่วยในการจัดคิวให้คอมพิวเตอร์ 8 ตัว ในการส่งข้อมูลออกพิมพ์ได้โดยอัตโนมัติ

ในส่วนมาสเตอร์บอร์ด (MASTER BOARD) จะใช้เมนบอร์ด (MAIN BOARD) ของเครื่องคอมพิวเตอร์ IBM PC COMPATIBLE ซึ่งทำให้โครงงานนี้มีประสิทธิภาพการทำงานที่เชื่อถือได้ ซึ่งเราจะใช้ไมโครโปรเซสเซอร์ INTEL 8088 และหน่วยความจำ จากบอร์ดนี้ เราจะใช้ PRINTER INTERFACE CARD ของ IBM PC เป็นส่วน OUTPUT PORT ซึ่งทำหน้าที่เป็น PRINTER OUTPUT PORT เพื่อส่งข้อมูลไปยังเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CENTRONIC PRINTER BUFFER

SUTHEE GITCHARWEE

ASSOCIATE PROFESSOR KITTI TRIRASAT

1988

ABSTRACT

THIS THESIS IS AN APPLICATION OF DEVICE TO INCREASE EFFICIENCY OF COMPUTER IN COMMUNICATION WITH PHERIPHERAL DEVICE WHICH HAVE SLOW VELOCITY IN OPERATION SUCH AS PRINTER. CENTRONIC PRINTER BUFFER RECEIVE DATA FROM COMPUTER BY VELOCITY ABOUT OF COMPUTER AND THEN SEND DATA TO PRINTER BY VELOCITY OF PRINTER. BY USING RING BUFFER OR CIRCULAR QUEUE PROGRAM WHICH IS HEART OF THIS PROJECT. THEY HAVE SAVING DATA IN MEMORY IS CIRCULAR (SIMILAR RING). BY USING TWO POINTER IN BUFFER, WHICH CAN RECEIVE DATA INSTEAD PRINTED DATA AREA AND ARRANG IN QUEUE FOR EIGHT COMPUTER TO SEND DATA TO PRINT.

IN MASTER BOARD PART USE MAIN BOARD OF IBM PC COMPATIBLE WHICH WILL GIVE BELIEVE EFFICIENCY TO THIS PROJECT. THIS PROJECT USE CPU INTEL 8088 AND MEMORY FROM THIS BOARD. WE USE PRINTER INTERFACE CARD OF IBM PC IS OUTPUT PORT WHICH FOR OUTPUT DATA TO PRINTER.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 หลักการทำงานและโครงสร้างของ CENTRONIC PRINTER BUFFER	3
2.1 หลักการทำงาน	3
2.2 บล็อกไดอะแกรม (BLOCK DIAGRAM)	5
2.3 ทฤษฎีการทำงาน	6
2.4 การอินเตอร์เฟสเครื่องพิมพ์ (THE PRINTER INTERFACE)	7
2.4.1 การอินเตอร์เฟสแบบเซนโทรนิค	8
2.4.2 แฮนด์เชค (HANDSHAKE)	8
2.4.3 ขั้วต่อของอินเตอร์เฟสแบบเซนโทรนิค (CONNECTOR)	10
2.5 บอร์ดอะแดปเตอร์ต่อเครื่องพิมพ์ (PRINTER ADAPTER)	12
2.5.1 หมายเลขพอร์ตอินพุทและเอาต์พุทของอะแดปเตอร์เครื่องพิมพ์ (I/O ADDRESSES FOR PRINTER ADAPTERS)	12
2.5.2 พอร์ตการเชื่อมโยงกับเครื่องพิมพ์	15
บทที่ 3 การสร้างและการทำงานของ CENTRONIC PRINTER BUFFER	19
3.1 วงจรและการทำงานของ CENTRONIC PRINTER BUFFER	19
3.2 การทำงานของวงจรมาคัด PRINTER SELECT	25
3.3 การทำงานของวงจรมาคัด DISPLAY	27
3.4 การทำงานของโปรแกรม	29
- MAIN FLOW CHART	32
- CHECK KEY SWITCH FLOWCHART	33
- PRIORITY FLOWCHART	34
- RECEIVE DATA FLOWCHART	35
- PRINT CH_NEXT FLOWCHART	36
- PRINT CH. FLOWCHART	37
- CHECK MAX FLOWCHART	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CHECK SEG. FLOWCHART	38
- โปรแกรม CENTRONIC PRINTER BUFFER	39
บทที่ 4 การทดลองและผลการทดลอง	56
4.1 การสร้างและการทดลองในส่วนทางด้านวงจร (HARDWARE)	56
4.2 การเขียนและการทดลองในส่วนโปรแกรม (SOFTWARE)	57
4.3 การทดลองการใช้งานเครื่อง CENTRONIC PRINTER BUFFER	59
บทที่ 5 บทวิจารณ์ สรุป	61
ภาคผนวก	
ภาคผนวก 1 วงจรอะแดปเตอร์เครื่องพิมพ์ ของ IBM PC/XT/AT	62
ภาคผนวก 2 ความสัมพันธ์ของขาและสัญญาณที่ส่งผ่านการอินเทอร์เฟสแบบขนาน	63
กิตติกรรมประกาศ	66
หนังสืออ้างอิง	67



สารบัญรูปภาพ

	หน้า
รูปที่ 1 CENTRONIC PRINTER BUFFER	3
รูปที่ 2 หลักการทำงานของ CENTRONIC PRINTER BUFFER	4
รูปที่ 3 บล็อกไดอะแกรมของ CENTRONIC PRINTER BUFFER	5
รูปที่ 4 บัฟเฟอร์แบบวงแหวน ซึ่งเป็นหัวใจของ CENTRONIC PRINTER BUFFER	7
รูปที่ 5 ไทม์มิ่งไดอะแกรม (TIMING DIAGRAM) สำหรับอินเตอร์เฟสแบบเซนโทรนิค	9
รูปที่ 6 รายละเอียดของขั้วต่อของอินเตอร์เฟสแบบเซนโทรนิคมาตรฐาน	11
รูปที่ 7 บล็อกไดอะแกรมของอะแดปเตอร์ต่อกับเครื่องพิมพ์	13
รูปที่ 8 แสดงการเชื่อมโยงกับเครื่องพิมพ์	15
รูปที่ 9 INPUT SECTION	21
รูปที่ 10 HANDSHAKING	22
รูปที่ 11 INPORT DECODER	23
รูปที่ 12 PRIORITY	24
รูปที่ 13 PRINTER SELECT	26
รูปที่ 14 DISPLAY	28
รูปที่ 15 วงจรอะแดปเตอร์เครื่องพิมพ์ ของ IBM PC/XT/AT	62

สารบัญตาราง

	หน้า
ตารางที่ 1 หมายเลขพอร์ตอินพุตและเอาต์พุตของอะแดปเตอร์เครื่องพิมพ์	14
ตารางที่ 2 ความสัมพันธ์ของขาและสัญญาณที่ส่งผ่านการอินเตอร์เฟสแบบขนาน	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

เนื่องจากในระบบคอมพิวเตอร์โดยทั่วไป เมื่อมีการติดต่อรับส่งข้อมูลระหว่างคอมพิวเตอร์กับอุปกรณ์สนับสนุนภายนอกใด ๆ จะต้องส่งข้อมูลผ่านบัฟเฟอร์ (BUFFER) ของอุปกรณ์นั้น เครื่องพิมพ์ (PRINTER) ก็เป็นอุปกรณ์ที่มีบัฟเฟอร์นี้เช่นกัน บัฟเฟอร์นี้จะ เป็นหน่วยความจำใช้เก็บข้อมูลชั่วคราว ไม่ว่าจะเป็นการรับหรือส่ง โดยเมื่อมีข้อมูลส่งมาจากเครื่องคอมพิวเตอร์ ข้อมูลจะถูกเก็บไว้ในบัฟเฟอร์ก่อน แล้วจึงทยอยส่งออกพิมพ์ตามลำดับ สำหรับสาเหตุที่ต้องมีบัฟเฟอร์นั้นเนื่องมาจากเครื่องพิมพ์อยู่ในลักษณะเครื่องกึ่งจักรกล มีการทำงานช้ากว่าเครื่องคอมพิวเตอร์มาก ถ้าไม่มีบัฟเฟอร์เป็นตัวช่วยเก็บข้อมูลชั่วคราวแล้ว จะทำให้เครื่องคอมพิวเตอร์ต้องรอเครื่องพิมพ์อยู่ตลอดเวลา ที่สั่งให้เครื่องพิมพ์ พิมพ์งานเอกสารหรือโปรแกรม ทำให้เราไม่สามารถใช้เครื่องคอมพิวเตอร์ทำงานอย่างอื่นได้เลยในช่วงเวลานี้ แต่บัฟเฟอร์ดังกล่าวนี้ในตัวเครื่องพิมพ์จะมีขนาดจำกัดเก็บข้อมูลได้ไม่มากนัก ในกรณีที่ข้อมูลส่งมาพิมพ์มาก ๆ แล้วก็เสียเวลารอนานเช่นกัน จากปัญหานี้เองที่บริษัทอินเทลล์บริษัทนี้ใช้เป็นแนวความคิด เพื่อแก้ปัญหาโดยการสร้างบัฟเฟอร์ชุดหนึ่งขึ้น ต่างหากจากของเครื่องพิมพ์ โดยมีความสามารถและลักษณะสำคัญดังนี้

1. มีหน่วยความจำสามารถเก็บข้อมูลได้ประมาณ 655,360 ตัวอักษร (640 KBYTES)
2. รับข้อมูลจากเครื่องคอมพิวเตอร์ได้ 8 เครื่อง (8 CHANNEL PARALLEL INPUT)
3. ส่งข้อมูลให้เครื่องพิมพ์ได้ 2 เครื่อง (2 CHANNEL PARALLEL OUTPUT)
4. สามารถเลือกให้ข้อมูลออกไปที่เครื่องพิมพ์เครื่องใดเครื่องหนึ่ง เครื่องไหนก็ได้หรือพร้อมกันทั้งสองเครื่อง ซึ่งเลือกได้โดยการกดปุ่มเพียงปุ่มเดียว
5. ใช้การอินเตอร์เฟซมาตรฐานแบบเซนโทรนิค (CENTRONIC INTERFACE)
6. การจัดหน่วยความจำใช้ แบบเซอร์คิวลาร์คิว (CIRCULAR QUEUE) ซึ่งทำให้ข้อมูลมาแทนที่ ส่วนของข้อมูลที่ถูกพิมพ์ไปแล้วทำให้ไม่ต้องรีเซ็ต (RESET) เครื่อง ตลอดการใช้งาน และสามารถทำการเรียงคิวพิมพ์ได้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสามารถของปริณิษานินทร์ฉบับนี้ ทำให้เราสามารถนำไปใช้งานและก่อประโยชน์ได้ตามจุดประสงค์ ดังนี้

1. เพื่อนำไปใช้กับการเรียนการสอนในวิชาทางคอมพิวเตอร์ ที่ต้องใช้เครื่องพิมพ์ โดยทำให้ไม่จำเป็นต้องซื้อเครื่องพิมพ์ให้เท่ากับจำนวนเครื่องคอมพิวเตอร์ ช่วยลดค่าใช้จ่ายได้
2. ทำให้คอมพิวเตอร์สามารถทำงานอย่างอื่นได้ โดยไม่ต้องรอให้เครื่องพิมพ์ทำงานเสร็จก่อน
3. ทำให้สามารถใช้เครื่องพิมพ์ร่วมกันระหว่างคอมพิวเตอร์ 8 เครื่องได้อย่างเป็นคิว โดยถ้าคอมพิวเตอร์เครื่องใดส่งข้อมูลมาให้เครื่องพิมพ์ได้ก่อน คอมพิวเตอร์เครื่องนั้นก็จะได้ใช้เครื่องพิมพ์ก่อน และถ้าคอมพิวเตอร์เครื่องใดส่งข้อมูลมาให้เครื่องพิมพ์เป็นอันดับต่อมาคอมพิวเตอร์เครื่องนั้นก็จะได้ใช้เครื่องพิมพ์ เป็นอันดับต่อมา แล้ว CENTRONIC PRINTER BUFFER นี้ก็จะทยอยส่งข้อมูลให้กับเครื่องพิมพ์ ในอัตราที่เครื่องพิมพ์สามารถทำงานได้ทัน โดยการรับข้อมูลจากคอมพิวเตอร์เพื่อมาเก็บไว้ในหน่วยความจำนั้น จะเร็วเท่ากับที่คอมพิวเตอร์ส่งข้อมูลมาให้เครื่องพิมพ์เครื่องใด

บทที่ 2

หลักการทํางานและโครงสร้างของ CENTRONIC PRINTER BUFFER

2.1 หลักการทํางาน

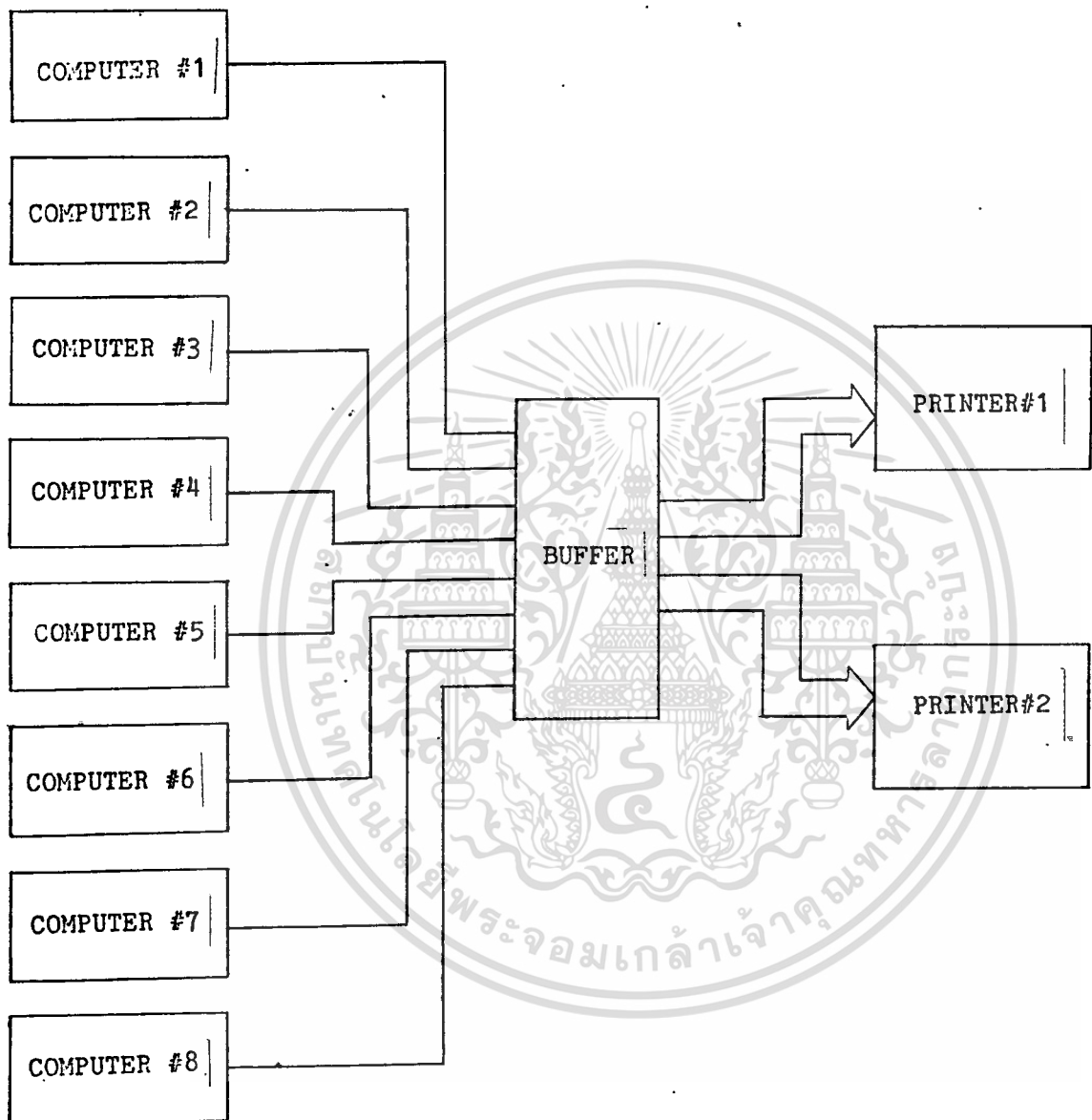
CENTRONIC PRINTER BUFFER เสมือนเป็นกล่อง ๆ หนึ่งที่ต่อคั่นระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ โดยในกล่องจะมีหน่วยความจำซึ่งจะรับข้อมูลจากคอมพิวเตอร์ด้วยความเร็วเท่ากับที่คอมพิวเตอร์ส่งมา แล้วค่อยทยอยส่งข้อมูลให้กับเครื่องพิมพ์ในอัตราที่เครื่องพิมพ์สามารถทํางานได้ทัน



รูปที่ 1 CENTRONIC PRINTER BUFFER เสมือนเป็นกล่องในหนึ่งที่ต่อคั่นระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ โดยภายในมีหน่วยความจำบรรจุอยู่

คอมพิวเตอร์ที่ต่ออยู่จะมอง CENTRONIC PRINTER BUFFER เสมือนเป็นเครื่องพิมพ์ ความเร็วสูงเท่ากับ ความเร็วของคอมพิวเตอร์ ส่วนทางด้านเครื่องพิมพ์จะมอง CENTRONIC PRINTER BUFFER เสมือนเป็นคอมพิวเตอร์ต่ออยู่กับตัวมันปกติ

การใช้งานหลักของ CENTRONIC PRINTER BUFFER ก็คือช่วยให้คอมพิวเตอร์ทํางานอย่างอื่นต่อได้ โดยไม่ต้องรอให้เครื่องพิมพ์ทํางานเสร็จก่อน และยังทำให้เราสามารถให้เครื่องพิมพ์ร่วมกันระหว่างคอมพิวเตอร์ 8 เครื่อง ได้อย่างเป็นคิว

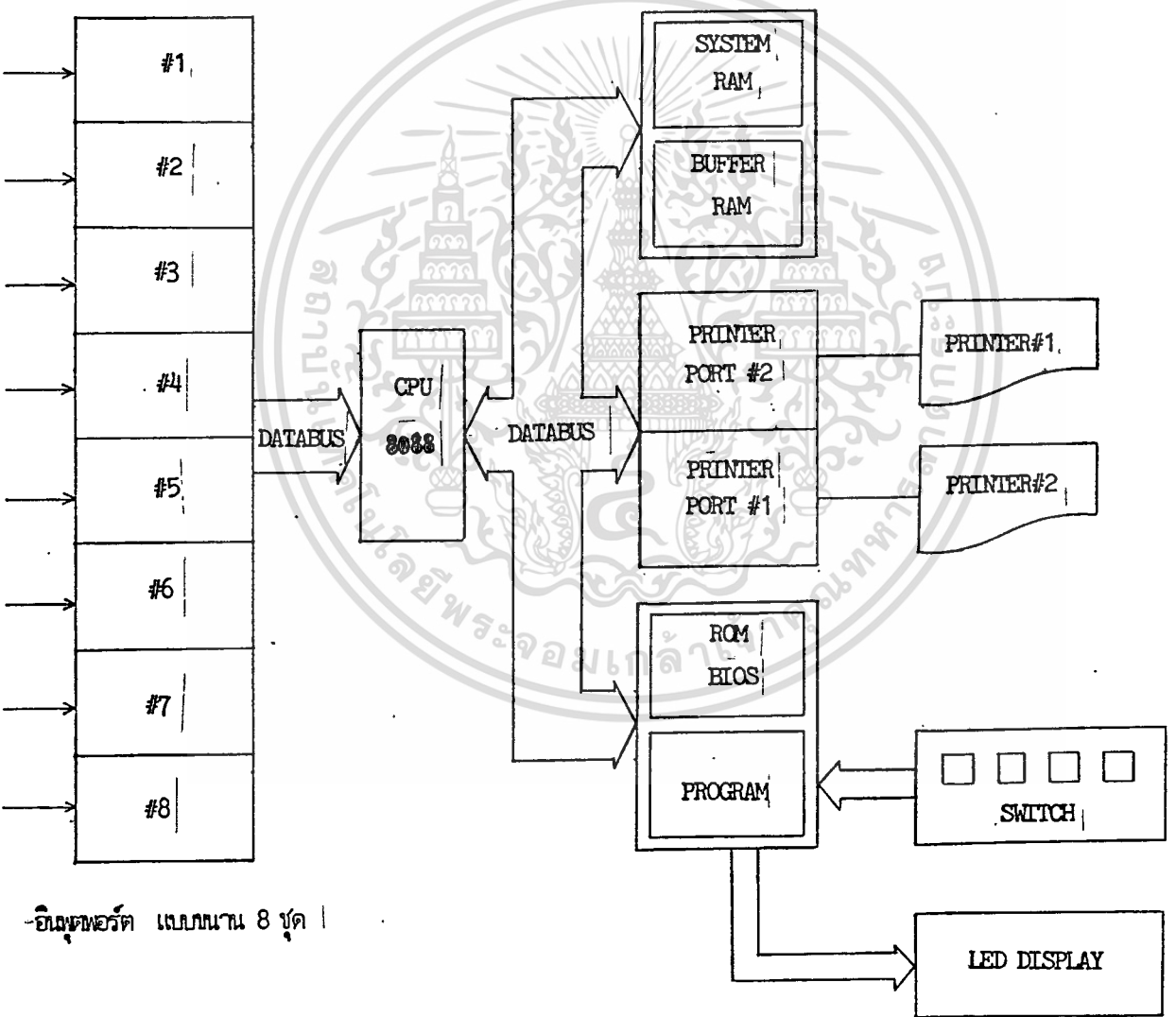


รูปที่ 2 หลักการทำงานของ CENTRONIC PRINTER BUFFER

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 บล็อกโคอะแกรม (BLOCK DIAGRAM)

ผังรูปที่ 3 เป็นบล็อกโคอะแกรมของ CENTRONIC PRINTER BUFFER ประกอบด้วย ไมโครโปรเซสเซอร์อินเทล 8088 (INTEL 8088 CPU), อินพุต/เอาต์พุต (I/O PORT), วงจรอินพุต/เอาต์พุต ประกอบด้วยวงจรอินเตอร์เฟสแบบขนาน ในรูปจะเห็นว่าหน่วยความจำจะถูกแบ่งเป็น 2 ส่วน แต่จริงแล้วหน่วยความจำทั้งหมดต่อเนื่องกันตลอดเป็นบล็อกเดียวกัน โดยในส่วนของหน่วยความจำที่เป็นไดนามิคแรม (DYNAMIC RAM) จะใช้เป็นสแต็ค (STACK) เสียส่วนหนึ่ง และที่เหลือจะใช้เป็นพื้นที่สำหรับเก็บข้อมูลที่รับจากคอมพิวเตอร์



รูปที่ 3 บล็อกโคอะแกรมของ CENTRONIC PRINTER BUFFER

2.3 ทฤษฎีการทำงาน

บัฟเฟอร์แบบวงแหวน (RING BUFFER)

บัฟเฟอร์แบบวงแหวนนับเป็นหัวใจสำคัญของ CENTRONIC PRINTER BUFFER โดยมีการเก็บข้อมูลในหน่วยความจำเป็นวงกลมรอบคล้ายวงแหวน

ภายในส่วนของระบบบัฟเฟอร์จะมีตัวชี้ (POINTER) 2 ตัวเรียกว่า

1. HEAD

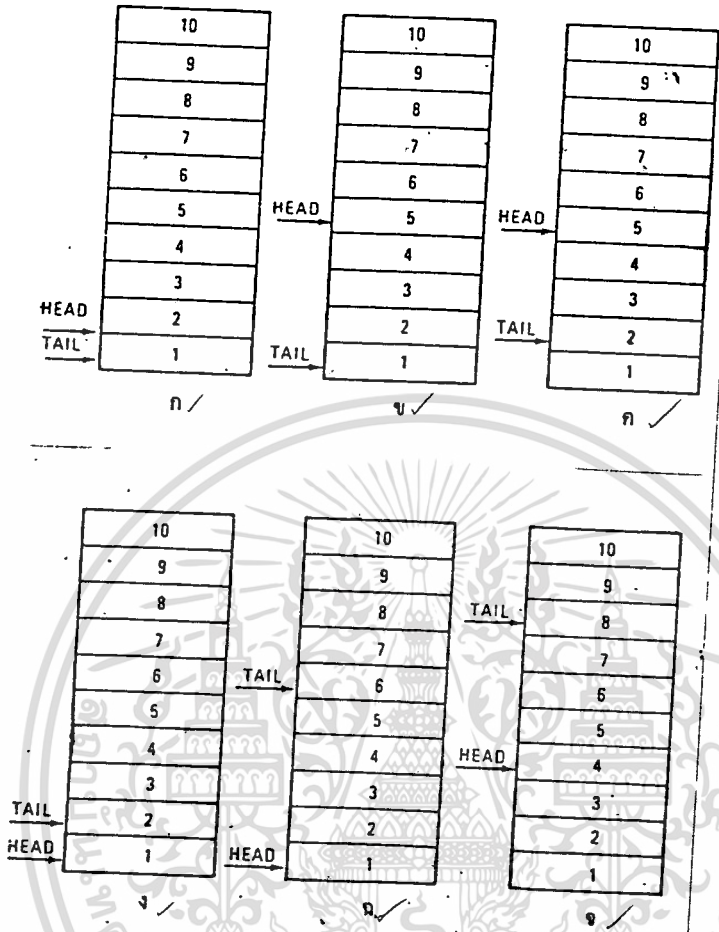
2. TAIL

ตัวชี้ HEAD จะชี้ไปยังตำแหน่งที่ยังว่างอยู่ของหน่วยความจำบัฟเฟอร์ ส่วนตัวชี้ TAIL จะชี้ไปที่ตำแหน่งของตัวอักษรตัวสุดท้าย ที่ถูกส่งออกไปให้เครื่องพิมพ์ เมื่อใดก็ตามที่ HEAD และ TAIL ชี้ที่ตำแหน่งเดียวกัน แสดงว่าบัฟเฟอร์นั้นว่างเปล่าอยู่ คือ ไม่มีข้อมูลอยู่เลย.

สมมุติว่าเรามีบัฟเฟอร์ที่เก็บข้อมูลได้ 10 ไบท์ ดังในรูปที่ 4 (ก) หลังจากที่เปิดเครื่อง ทั้ง HEAD และ TAIL ชี้อยู่ที่ตำแหน่งที่ 1 เหมือนกัน เมื่อมีข้อมูลเข้ามา 1 ตัวอักษร HEAD จะเลื่อนไปชี้ที่ตำแหน่ง 2 และเมื่อมีข้อมูลเข้ามาอีกเรื่อย ๆ HEAD ก็จะไปเลื่อนตำแหน่งไปด้วย เป็นดังรูป 4 (ข) เมื่อบัฟเฟอร์ส่งตัวอักษรไปยังเครื่องพิมพ์ TAIL ก็จะขยับเลื่อนขึ้นมาเรื่อยๆ ดังรูป 4 (ค)

จากที่อธิบายผ่านมามีเห็นว่าไม่มีอะไรยุ่งยาก แต่ถ้ามีข้อมูลเข้ามาอีกสองสามตัวหลังจากตำแหน่งที่ 10 มีข้อมูลอยู่แล้วจะเกิดอะไรขึ้น เนื่องจากในบัฟเฟอร์ไม่มีที่ที่จะเก็บข้อมูลแล้ว ขณะนี้บัฟเฟอร์ก็จะรู้ว่า ตัวชี้ HEAD ชี้อยู่กับตำแหน่งสูงสุดของหน่วยความจำแล้ว ก็จะเช้ให้ HEAD ไปชี้อยู่ที่ตำแหน่ง 1/ ดังในรูป 4 (ง) และเมื่อ HEAD และ TAIL มาชี้ที่ตำแหน่งเดียวกัน ก็ จะบอกให้คอมพิวเตอร์หยุดส่งข้อมูลมาให้บัฟเฟอร์ก่อน จนกว่าจะมีที่ว่างให้เก็บข้อมูลเสียก่อน ดังในรูป 4 (จ)

พื้นที่ส่วนล่างจะมีการนำกลับมาใช้ใหม่อยู่เรื่อย ๆ โครงสร้างของ HEAD/TAIL ทำให้หน่วยความจำแรม (RAM) มองดูเหมือนกับเป็นวงแหวนที่ต่อเนื่อง จึงเรียกบัฟเฟอร์แบบนี้ว่า บัฟเฟอร์วงแหวน



รูปที่ 4 บัฟเฟอร์แบบวงแหวน ซึ่งเป็นหัวใจของ CENTRONIC PRINTER BUFFER

2.4 การอินเตอร์เฟสเครื่องพิมพ์ (THE PRINTER INTERFACE)

การต่อคอมพิวเตอร์เพื่อใช้งานกับอุปกรณ์หรือระบบหลักอื่น ๆ เราจะต้องต่อผ่านส่วนที่เรียกว่า อินเตอร์เฟส (INTERFACE) เสมอ ซึ่งส่วนนี้ประกอบไปด้วย ขาข้อมูล (DATA BUS) และขาควบคุม (CONTROL BUS) การอินเตอร์เฟสมีใช้อยู่มากมายหลายแบบ มีทั้งที่ติดมาพร้อมกับเครื่องคอมพิวเตอร์ ซึ่งเป็นแบบเฉพาะจะใช้ได้กับอุปกรณ์แบบเดียวกันเท่านั้น ไม่สามารถนำไปใช้กับเครื่องในแบบอื่นได้เลย และมีแบบมาตรฐานที่คอมพิวเตอร์โดยทั่วไปใช้กัน ซึ่งช่วยให้สามารถเลือกใช้อุปกรณ์ต่าง ๆ ได้มากมาย การอินเตอร์เฟสแบบมาตรฐานมีอยู่หลายแบบ แต่ที่พบเห็นและใช้กันค่อนข้างจะแพร่หลายคือ เซนโทรนิค , RS-232C , IEEE-488 , RS-100 , DATA PRODUCT , และ GPIB เป็นต้น การใช้อินเตอร์เฟสแบบมาตรฐานนี้ค่อนข้างง่ายในการทำงาน ปัญหาที่เกี่ยวกับการต่อสายระหว่างคอมพิวเตอร์กับอุปกรณ์ที่ใช้ระบบอินเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟสแบบเดียวกัน จะต้องถูกต้องเท่านั้น และการใช้ขั้วต่อต้องใช้ตามแบบมาตรฐานของการอินเตอร์เฟสแต่ละชนิด ตัวอย่างเช่น คอมพิวเตอร์ที่มีพอร์ตต่อเครื่องพิมพ์แบบเซนโทรนิค ก็ต้องใช้ขั้วต่อแบบ 36 ขาตามมาตรฐาน

จากลักษณะการอินเตอร์เฟสแบบเซนโทรนิคมาตรฐาน จะมีการส่งข้อมูลแบบทางเดียว คือเป็นเอาต์พุตออกจากคอมพิวเตอร์ได้อย่างเดียว ข้อมูลจะไม่มีทางส่งกลับเข้าสู่คอมพิวเตอร์เลย จึงเหมาะสมกับอุปกรณ์พวกเครื่องพิมพ์ แต่ไม่สามารถใช้กับอุปกรณ์ที่ต้องมีการรับส่งข้อมูลแบบสองทิศทางเช่น โมเด็ม (MODEM) ได้ ปรวิญญานพจนรจฉบับนี้จึงใช้และกล่าวเฉพาะส่วนของอินเตอร์เฟสแบบเซนโทรนิคเท่านั้น

2.4.1 การอินเตอร์เฟสแบบเซนโทรนิค

เป็นการอินเตอร์เฟสที่นิยมใช้ในพอร์ตประเภทเครื่องพิมพ์ทั่วไป เช่นโทรนิคเป็นอินเตอร์เฟสแบบขนาน (STANDARD PARALLEL INTERFACE) โดยข้อมูลจะถูกส่งออกไปแบบขนานคือข้อมูล 1 ไบท์ จะส่งออกไปพร้อมกันทั้ง 8 บิต (BIT)/โดยใช้สายสัญญาณ 8 เส้นกับกราวด์ (GROUND) อีก 1 เส้นแต่คอมพิวเตอร์บางเครื่องก็ไม่ได้ใช้ประโยชน์จากการส่งข้อมูลได้พร้อมกันทั้ง 8 บิต นี้ได้อย่างเต็มที่ บางครั้งจะใช้เพียงแค่ 7 บิต โดยต่อบิตสูงสุดไว้กับกราวด์ตลอด เพราะเครื่องคอมพิวเตอร์นั้นส่วนมากใช้รหัสแอสกี (ASCII CODE) ซึ่งใช้เพียง 7 บิต เท่านั้น บางครั้งการต่อบิตสูงสุดลงกราวด์นี้จะก่อปัญหาเกี่ยวกับเครื่องพิมพ์ที่ต้องการใช้บิตที่ 8 ในการส่งรหัสควบคุมของเครื่องพิมพ์ เพราะจะไม่สามารถส่งรหัสควบคุมไปยังเครื่องพิมพ์ได้เลย

✱ ในทางปฏิบัติจริง ๆ แล้วการใช้สายสัญญาณ 8 เส้นกับกราวด์อีก 1 เส้นนี้ ก็ยังไม่สามารถใช้งานได้จริง ๆ เพราะเครื่องพิมพ์จะไม่มีทางรู้ได้เลยว่า มีข้อมูลชุดใหม่มาเรียง หรือว่าขณะนี้เครื่องพิมพ์กำลังพิมพ์อยู่ยังไม่สามารถที่จะรับข้อมูลเข้าไปได้จึงได้เพิ่มสัญญาณสโตรบ (STROBE) ✱ เป็นเอาต์พุตจากคอมพิวเตอร์ ซึ่งจะส่งพัลส์ลบ 1 ลูกทุกครั้งที่ข้อมูลไบท์ใหม่ เข้ามาพร้อมอยู่ที่บัลลูนข้อมูลเรียบร้อยแล้ว รอเครื่องพิมพ์มารับไปพิมพ์ ที่ตัวของเครื่องพิมพ์จะมีตัวแลทซ์ข้อมูล 8 บิตอยู่ และจะใช้สัญญาณสโตรบในการอ่านข้อมูลเข้าไป

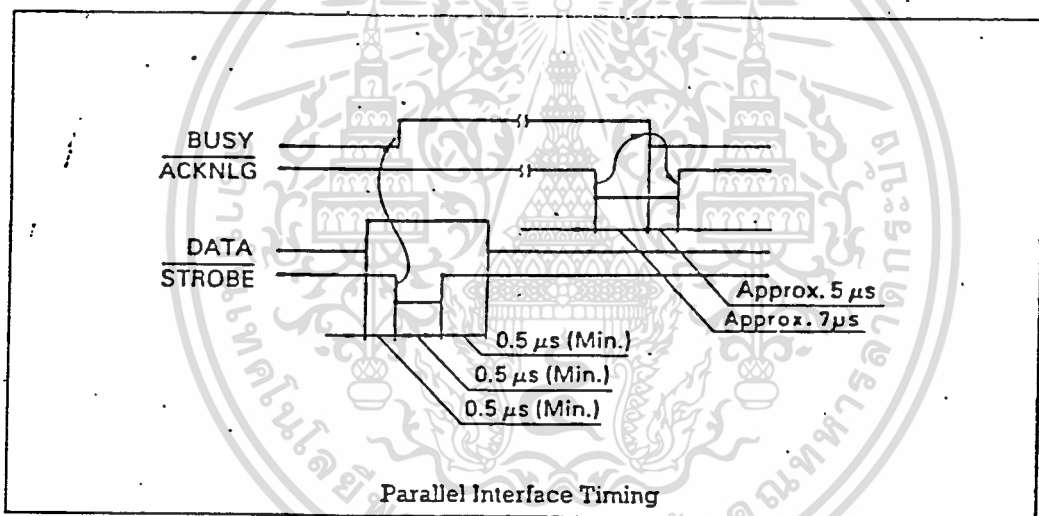
2.4.2 แฮนด์เชค (HANDSHAKE)

ระบบที่กล่าวมาในหัวข้อ 2.4.1 จะทำงานได้จริงก็ต่อเมื่อเครื่องพิมพ์สามารถรับข้อมูลได้เร็วเท่ากับที่คอมพิวเตอร์ส่งมาให้ แต่ความจริงแล้ว คอมพิวเตอร์สามารถส่งข้อมูลได้หลายกิโลไบท์ภายใน 1 วินาที / แต่เครื่องพิมพ์ที่ทำงานเร็วที่สุดสามารถพิมพ์ได้อย่างมากประมาณเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อที่ 8 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2-300 ตัวอักษรต่อวินาทีเท่านั้น ถึงแม้ว่าเครื่องจะมีบัฟเฟอร์ช่วยในการเก็บข้อมูลก่อนจะพิมพ์ เพื่อให้สามารถรับข้อมูลได้เร็วขึ้น แต่ขนาดของบัฟเฟอร์ก็อาจจะไม่เพียงพอ ถ้ามีข้อมูลมาก

การควบคุมข้อมูลที่ส่งให้เครื่องพิมพ์ ในความเร็วที่เครื่องพิมพ์สามารถรับได้นี้ นับว่ามีส่วนสำคัญทีเดียว จึงมีสัญญาณ BUSY และ ACKNOWLEDGE เพื่อช่วยในการทำแฮนด์เชค ซึ่งอาจจะใช้เพียงอันใดอันหนึ่งเพียงสัญญาณเดียว หรืออาจใช้ทั้งสองสัญญาณร่วมกันก็ได้ ในเครื่องพิมพ์ส่วนมากจะมีเอาท์พุททั้งสองสัญญาณนี้ แต่คอมพิวเตอร์หลาย ๆ ตัวจะมีอินพุทสำหรับสัญญาณเท่านั้น เช่น BBC MODEL B จะมีเฉพาะช่องอินพุทของ ACKNOWLEDGE เท่านั้น



รูปที่ 5 ไทม์มิ่งไดอะแกรม (TIMING DIAGRAM) สำหรับอินเตอร์เฟสแบบเซนโทรนิค สัญญาณ DATA และ STROBE ส่งโดยคอมพิวเตอร์และสัญญาณ BUSY กับ ACKNLG ส่งโดยเครื่องพิมพ์

อนึ่ง คอมพิวเตอร์นั้นสามารถส่งข้อมูลออกได้สูงถึง 150 KILOBYTES/SECOND เครื่องพิมพ์ที่รับข้อมูลจากคอมพิวเตอร์ จึงจำเป็นต้องบอกให้คอมพิวเตอร์รอมัน โดยการส่งสัญญาณ BUSY เป็น 1 ไปให้ ซึ่งเราเรียกสิ่งนี้ว่า "SECOND TWO-WAYHANDSHAKE" ซึ่งนำมาใช้ในการอินเตอร์เฟสแบบเซนโทรนิคนี้

จากรูปที่ 5 เริ่มแรกนั้นสัญญาณ BUSY ในช่วงปกติ (ยังไม่มีข้อมูลเข้ามาที่เครื่องพิมพ์) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา 9 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็น 0 (LOW) เมื่อส่งข้อมูลมา 1 ไบท์ เครื่องพิมพ์จะเซ็ท (SET) ให้สัญญาณ BUSY เป็น 1 (HIGH) ตลอดจนกว่าข้อมูลจะเข้าไปเก็บในบัฟเฟอร์หรือนำไปพิมพ์เสร็จเรียบร้อยแล้ว คอมพิวเตอร์ก็จะคอยตรวจที่สัญญาณนี้ โดยใช้โปรแกรมย่อยที่เป็นซอฟต์แวร์ (SOFTWARE) จนกระทั่งสัญญาณนี้เป็น 0 อีกครั้งหนึ่งจึงจะส่งข้อมูลไบท์ต่อไปให้เครื่องพิมพ์ การทำงานจะเป็นเช่นนี้ตลอดจนกว่าข้อมูลจะถูกส่งไปทั้งหมด

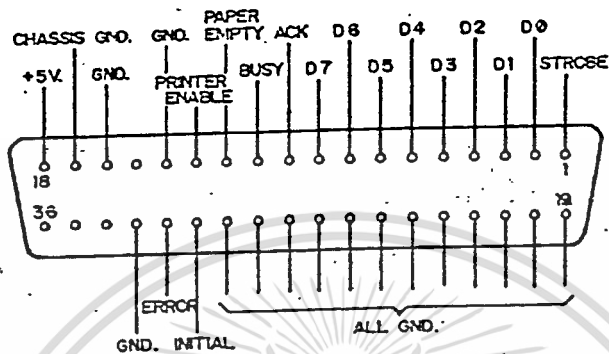
สำหรับสัญญาณ ACKNOWLEDGE ปกติจะเป็น 1 เมื่อมีข้อมูลเข้ามาสัญญาณนี้ก็ยังคงเป็น 1 อยู่ จนกระทั่งข้อมูลถูกนำมาเก็บในบัฟเฟอร์หรือถูกพิมพ์เรียบร้อยแล้ว/เครื่องพิมพ์จะส่งพัลส์ (PLUS) ลบออกมา 1 ลูก, ที่สัญญาณ ACKNOWLEDGE นี้ ที่เครื่องคอมพิวเตอร์จะใช้ซอฟต์แวร์ หรือโปรแกรมย่อย ในการเช็คพัลส์ลบนี้ก่อนที่จะส่งข้อมูลไบท์ต่อไปให้เครื่องพิมพ์

จะเห็นได้ว่าการแฮนด์เชคนั้น เมื่อคอมพิวเตอร์ส่งไบท์ข้อมูลสู่เครื่องพิมพ์ คอมพิวเตอร์จะส่งพัลส์ลบลงสู่เส้น STROBE เพื่อบอกเครื่องพิมพ์ว่า "เฮ้ ฉันให้ข้อมูลไบท์นี้แก่คุณ" เมื่อเครื่องพิมพ์อ่านข้อมูลไบท์นั้นแล้ว จะตอบสนองด้วยการส่งพัลส์ตอบกลับไปยังเส้น ACKNLG สู่คอมพิวเตอร์ว่า " โอ เค ฉันได้อ่านมันเรียบร้อยแล้ว"

2.4.3 ขั้วต่อของอินเตอร์เฟสแบบเซนโทรนิก (CONNECTOR)

ขั้วต่อมาตรฐานของเซนโทรนิก ใช้แบบแอมป์เฟอร์นอล (AMPHENAL) 36 ขา เรียกว่า ขั้วต่อเซนโทรนิก ซึ่งมีทั้งแบบปลั๊ก (PLUG) และซ็อกเก็ต (SOCKET) ที่ตัวเครื่องพิมพ์จะมีซ็อกเก็ตติดอยู่ แต่ที่พอร์ตของคอมพิวเตอร์ส่วนมากจะเป็นขั้วต่อแบบ IDC หรือแบบ D-TYPE ก็ได้

ในรูปที่ 6 จะแสดงถึงสัญญาณที่ขาต่าง ๆ ของการอินเตอร์เฟสแบบมาตรฐานของเซนโทรนิก ซึ่งมีสัญญาณหลายสัญญาณ จะเห็นว่าเส้นกราวด์จะมีอยู่หลายเส้น ก็เพื่อช่วยแยกกระหว่างสายสัญญาณข้อมูล เมื่อเราใช้สายริบบอน (RIBBON WIRE) ในการติดต่อเพื่อลดค่าความจุแฝงที่เกิดขึ้นภายในสายเคเบิล ซึ่งอาจทำให้การส่งข้อมูลเกิดการผิดพลาดได้ แต่ก็ไม่ควรใช้สายเคเบิลที่ยาวเกิน 2 เมตร แต่ทางปฏิบัติแล้วเราอาจจะต่อกราวด์เพียงเส้นเดียวก็ได้ แต่ความยาวควรลดลงเหลือเพียงครึ่งเมตรเท่านั้น



รูปที่ 6 รายละเอียดของขั้วต่อของอินเตอร์เฟสแบบเซ็นโทรนิคมาตรฐาน

ลักษณะการเชื่อมต่อของขาสัญญา ระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ จะมีบางเส้นที่ต้องใช้ต่ออีก นอกเหนือจากสัญญาณ STROBE , ACKNLG , BUSY , DATA คือสัญญาณ PE ซึ่งเป็นสัญญาณอินพุทเข้าสู่คอมพิวเตอร์ เพื่อแสดงว่า "ขณะนี้เครื่องพิมพ์อยู่ในกรณีที่กระดาษหมด" สัญญาณนี้จะแอกทีฟ (ACTIVE) ที่ลอจิก 1 จึงต้องต่อไว้ที่ลอจิก 0 สัญญาณ PE ย่อมาจาก PAPER EMPTRY (หรือบางครั้งเรียกว่า PAPER END) จะเป็น 1 เมื่อเครื่องพิมพ์ตรวจพบว่ากระดาษหมด

สัญญาณ ERROR เป็นสัญญาณที่ส่งเข้าสู่คอมพิวเตอร์ เพื่อแสดงว่า "เครื่องพิมพ์รับรู้ว่ามีข้อผิดพลาดเกิดขึ้น " เช่น กระดาษหมด , อยู่ในระหว่าง OFF LINE , ข้อมูลที่รับเข้ามาผิดพลาดไม่สามารถตีความได้ สัญญาณนี้แอกทีฟที่ลอจิก 0 จึงต้องต่อให้สัญญาณนี้มีสภาวะเป็นลอจิก 1 .

สัญญาณ INITIAL เป็นอินพุทของเครื่องพิมพ์ ใช้สำหรับรีเซ็ตตัวเครื่องพิมพ์ เช่น กรณีหลังจากเกิดข้อผิดพลาดที่เครื่องพิมพ์ ซึ่งถ้าไม่มีสัญญาณอินพุทนี้ก็สามารถทำได้โดยการปิดเครื่องพิมพ์แล้วเปิดใหม่อีกครั้งหนึ่งจะให้ผลเหมือนกัน

สัญญาณ PRINT ENABLED เป็นเอาต์พุทของเครื่องพิมพ์ ซึ่งจะเป็น 1 เมื่อเครื่องพิมพ์นี้

ถูกเลือกใช้งาน ขานี้บางครั้งเรียกว่า ขาเลือก (SELECT LINE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสัญญาณ +5 โวลต์ (VOLT) เป็นขาไฟตรง 5 โวลต์ จากภาคจ่ายไฟสามารถจ่ายกระแสได้ประมาณ 100 มิลลิแอมป์ (MILLIAMP)

เครื่องพิมพ์แต่ละยี่ห้อ อาจจะมีสัญญาณที่แตกต่างจากนี้ไปบ้างก็เพียงเล็กน้อยเท่านั้น เช่น สัญญาณนาฬิกา หรือ สัญญาณในการเลื่อนกระดาษเป็นสองเท่าแบบอัตโนมัติ แต่จะใช้ขาที่ยังว่างอยู่หรืออาจจะแทนที่ขาสัญญาณที่มีความสำคัญน้อย

หนึ่งขาสัญญาณ SELECT INPUT เป็นขาที่สำคัญขาหนึ่ง ขานี้จะต้องอยู่ในสถานะที่ถูกดึงก่อนเสมอ เครื่องพิมพ์จึงจะตอบสนองต่อสัญญาณอื่น ๆ และจึงจะพิมพ์ให้ และสัญญาณจะต้องอยู่บนเส้นขานี้เป็นเวลาอย่างน้อย 50 ไมโครวินาที (MICROSECOND)

2.5 บอร์ดอะแดปเตอร์ต่อเครื่องพิมพ์ (PRINTER ADAPTERS)

เครื่องคอมพิวเตอร์ต่อกับเครื่องพิมพ์โดยใช้พาราเรลอินเตอร์เฟส ซึ่งอาจใช้บอร์ดโมโนโครม (MONOCHROME DISPLAY) และบอร์ดอะแดปเตอร์เครื่องพิมพ์ (PRINTER ADAPTER) หรือบอร์ดพาราเรลอะแดปเตอร์เครื่องพิมพ์ (PARALLEL PRINTER ADAPTER)

2.5.1 หมายเลขพอร์ตอินพุทและเอาต์พุทของอะแดปเตอร์เครื่องพิมพ์

(I/O ADDRESSES FOR PRINTER ADAPTERS)

การเชื่อมโยงกับเครื่องพิมพ์จะผ่านทางพอร์ตขนาน 8 บิต โดยบริษัท ไอ บี เอ็ม ได้ออกแบบให้สามารถต่อได้ตามมาตรฐานพอร์ตเซนโทรนิค ซึ่งมีขนาด 25 ขา พอร์ตนี้ เป็นไปได้ทั้งทั้งอินพุทและเอาต์พุท โดยมีกำลังขับได้ถึง 12 ทีทีแอลเอาต์พุท (TTL OUTPUT)

โดยทั่วไปพอร์ตนี้ปกติจะใช้เป็นเอาต์พุทเพื่อส่งข้อมูลให้กับเครื่องพิมพ์ แต่ยังสามารถรับข้อมูลเป็นอินพุทพอร์ตได้เช่นกัน พอร์ตของเครื่องพิมพ์จะมีสัญญาณต่างๆ เพื่อใช้ในการตรวจสอบสัญญาณซึ่งกันและกันด้วย

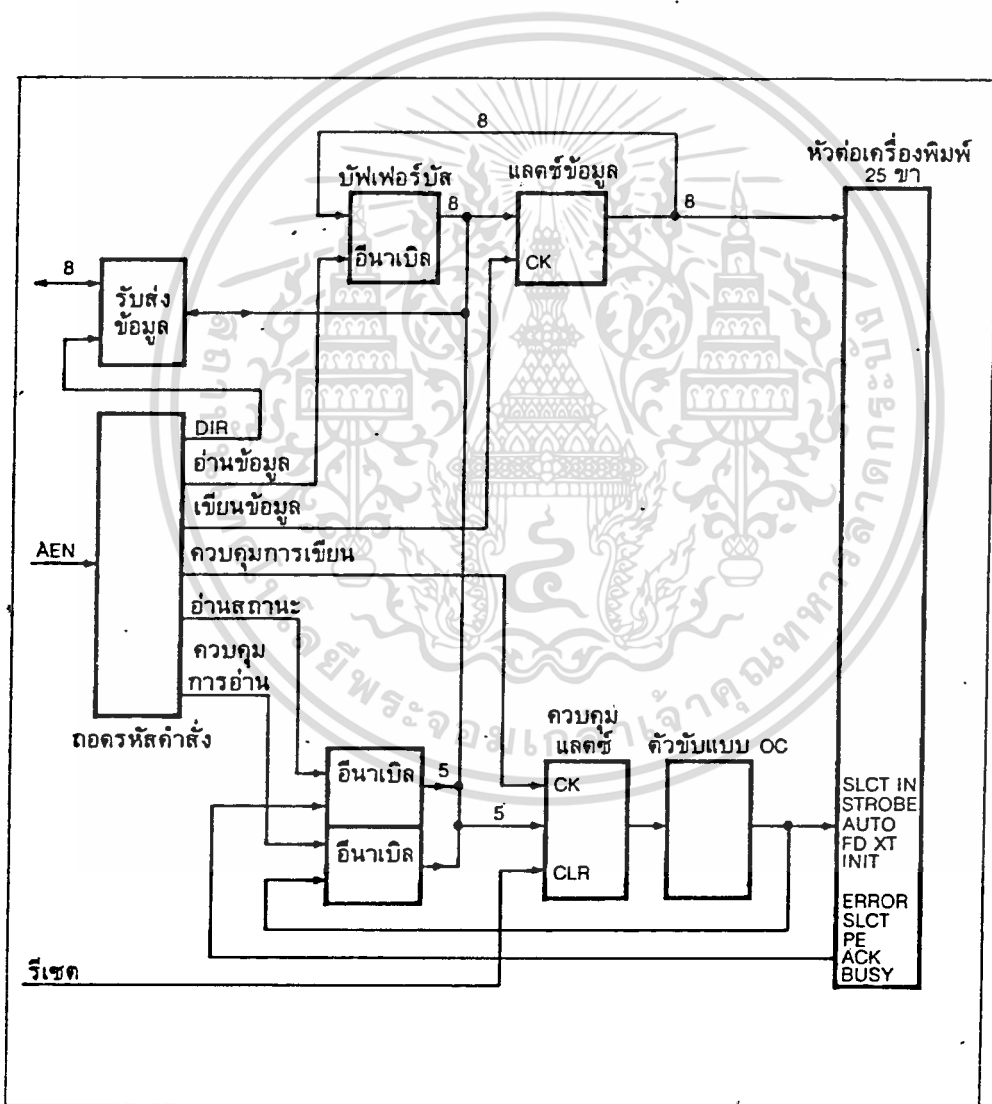
ปกติบริษัท ไอ บี เอ็ม ได้ออกแบบไบออส (BIOS) ไว้ให้ต่อกับเครื่องพิมพ์ได้ถึง 4 ตัวพร้อมกันดังนั้นจึงต้องเซ็ทหมายเลขพอร์ตอินพุทเอาต์พุทไว้แตกต่างกัน เช่น ในการใช้งานพอร์ตเครื่องพิมพ์ทั่วไป อาจอยู่บนอะแดปเตอร์โมโนโครม หรือบนอะแดปเตอร์เครื่องพิมพ์ต่างหากจะมีหมายเลขพอร์ตเพื่อกำหนดเป็นพอร์ตแอลพีที 1 (LPT1) โดยมีบล็อกไดอะแกรมดังรูปที่ 7

บนบอร์ดเชื่อมต่อกับเครื่องพิมพ์นี้ ได้จัดการเรื่องพอร์ตที่ซีพียู 8088 มองเห็นได้เป็นพอร์ตเอาต์พุท 2 พอร์ตและพอร์ตอินพุท 2 พอร์ต ซีพียู 8088 อาจจะมีอินพุทหรือเอาต์พุทข้อมูลเพื่อควบคุมหรืออ่านบนสายอินเตอร์เฟสเครื่องพิมพ์แบบขนาน (PARALLEL PRINTER INTERFACE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็ได้ แอดแตรสของอินพุทและเอาต์พุทพอร์ตจะแตกต่างกัน ระหว่างบอร์ดโมโนโครมและบอร์ดอะแดปเตอร์ เครื่องพิมพ์กับบอร์ดพาราเรลอะแดปเตอร์ เครื่องพิมพ์ ดังแสดงในตารางที่ 2 เช่น เมื่อต้องการ เอาท์พุทข้อมูลเพื่อไปทำการพิมพ์ เราควรใช้หมายเลขพอร์ตเท่ากับ 3BC_H กรณีใช้บอร์ดโมโนโครมและบอร์ดอะแดปเตอร์ เครื่องพิมพ์ หรือใช้หมายเลขพอร์ตเท่ากับ 378_H สำหรับบอร์ดพาราเรลอะแดปเตอร์ เครื่องพิมพ์



รูปที่ 7 บล็อกไดอะแกรมของอะแดปเตอร์ต่อกับเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา 13 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 หมายเลขพอร์ตอินพุทและเอาต์พุทของอะแดปเตอร์ เครื่องพิมพ์

หน้าที่	บอร์ดโมโนโครมและบอร์ด อะแดปเตอร์ เครื่องพิมพ์	บอร์ดพาราเรลอะแดป เตอร์ เครื่องพิมพ์
ส่งข้อมูลออก	3BCH	378H
อ่านสถานะเข้า	3BDH	379H
ควบคุมข้อมูล	3BEH	37AH

พอร์ตส่งข้อมูลออกเครื่องพิมพ์ (OUTPUT DATA PORT)

- พอร์ตหมายเลข 378H สำหรับบอร์ดอะแดปเตอร์ เครื่องพิมพ์
- พอร์ตหมายเลข 3BCH สำหรับบอร์ดโมโนโครม

พอร์ตส่งคำสั่งเพื่อควบคุมเครื่องพิมพ์สำหรับตอบโต้สัญญาณ (OUTPUT CONTROL PORT)

- พอร์ตหมายเลข 37AH สำหรับบอร์ดอะแดปเตอร์ เครื่องพิมพ์
- พอร์ตหมายเลข 3BEH สำหรับบอร์ดโมโนโครม

สำหรับอินพุทบนการ์ด (CARD) แต่ละการ์ดมี 3 พอร์ต

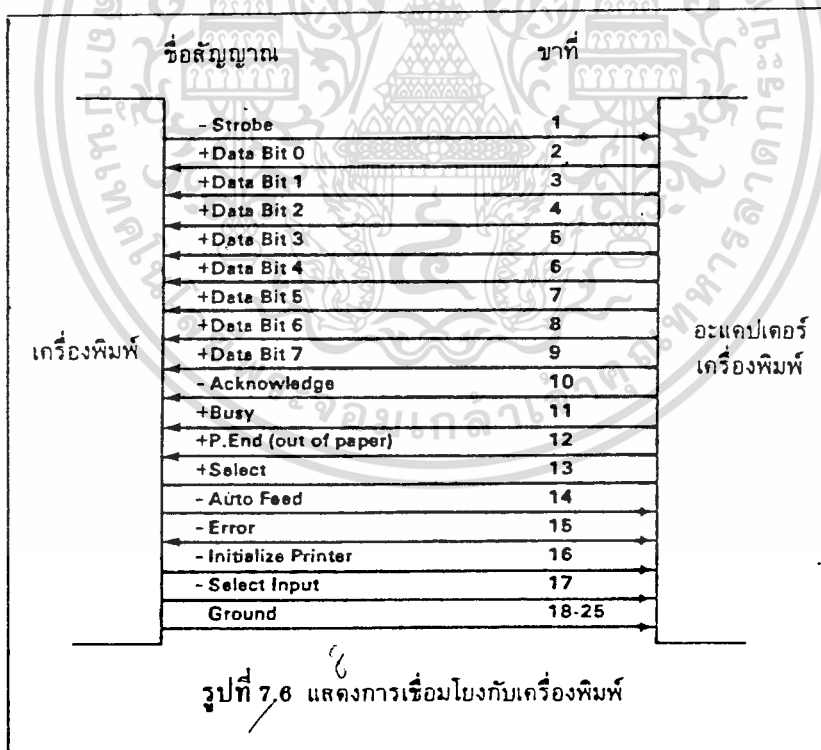
- พอร์ตหมายเลข 378H สำหรับการอ่านข้อมูลจากพอร์ตขนาน กรณีบอร์ดอะแดปเตอร์ เครื่องพิมพ์
- พอร์ตหมายเลข 3BCH สำหรับอ่านข้อมูลจากพอร์ตขนาน กรณีบอร์ดโมโนโครม
- พอร์ตหมายเลข 379H สำหรับการอ่านสถานะจากพอร์ตอินพุท ในกรณีบอร์ดอะแดปเตอร์ เครื่องพิมพ์
- พอร์ตหมายเลข 3BDH สำหรับการอ่านสถานะจากพอร์ตอินพุท ในกรณีบอร์ดโมโนโครม
- พอร์ตหมายเลข 37AH สำหรับการอ่านสถานะของ เครื่องพิมพ์ สำหรับบอร์ดอะแดปเตอร์ เครื่องพิมพ์
- พอร์ตหมายเลข 3BEH สำหรับการอ่านสถานะเครื่องพิมพ์ สำหรับบอร์ดโมโนโครม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 พอร์ตการเชื่อมโยงกับเครื่องพิมพ์

แสดงดังรูปที่ 8 เป็นการสรุปถึงสายที่ใช้สำหรับเครื่อง ไอบีเอ็ม พีซี (IBM PC) ที่เป็นอินเตอร์เฟซแบบเซนโทรนิก 25 ขา เช่นโทรนิกตามแบบมาตรฐานใช้คอนเนคเตอร์ 36 ขา ดังนั้นจำนวนขาของ ไอบีเอ็ม นี้จึงไม่มาตรฐาน สังเกตในส่วนที่เพิ่มเติมจากสาย BUSY , ACKNLG ,DATA,STROBE และเส้นกราวด์คือ PAPER END,SELECT, AUTOFEED, ERROR, INITIALIZE PRINTER และ SELECT INPUT

เส้น SELECT INPUT นี้จะต้องอยู่ในสภาวะ 0 ในขณะที่เครื่องพิมพ์ถูกเลือก ดังนั้นโดยการให้ SELECT INPUT เฉพาะเส้นสำหรับเครื่องพิมพ์แต่ละตัว เราสามารถใช้การอินเตอร์เฟซเดียวกัน เพื่อขับเครื่องพิมพ์ทั้งหมด โดยการแอนด์ไวด์เดียวกันและเส้นข้อมูลเดียวกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิวส์สัญญาณ ขาไป	นิวส์สัญญาณ ขากลับ	ชื่อสัญญาณ	ทิศทาง	คำอธิบาย
1	19	STROBE	อินพุท	เป็นสัญญาณที่ส่งจากอุปกรณ์ภายนอก เช่น ชุดไมโครคอมพิวเตอร์ เพื่อบอกให้เครื่องพิมพ์รับข้อมูลไปได้ โดยทั่วไปแล้วจะมีความกว้างพัลส์ ประมาณ 0.5 ถึง 1 ไมโครวินาที
2-9	20-27	DATA	อินพุท	เป็นสายสัญญาณหรือข้อมูลที่ส่งจากอุปกรณ์ภายนอก เป็นข้อมูลขนาด 8 บิต
10	28	ACKNLG	เอาท์พุท	เป็นสัญญาณที่ส่งจากเครื่องพิมพ์ เพื่อบอกให้อุปกรณ์ภายนอกได้รับรู้ ว่าได้รับข้อมูลไว้เรียบร้อยแล้ว โดยทั่วไปจะมีความกว้างของพัลส์ประมาณ 5 ถึง 12 ไมโครวินาที
11	29	BUSY	เอาท์พุท	เป็นสัญญาณที่ส่งจากเครื่องพิมพ์ เพื่อบอกให้อุปกรณ์ภายนอกทราบว่าขณะนี้เครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูล โดยจะให้สัญญาณ "1" ออกมา
12	30	PE	เอาท์พุท	เป็นสัญญาณที่ส่งจากเครื่องพิมพ์ ในกรณีที่กระดาษพิมพ์หมด
13	-	SELECT	เอาท์พุท	เป็นสัญญาณที่ส่งออกจากเครื่องพิมพ์ เพื่อบอกให้ทราบว่าขณะนี้เครื่องพิมพ์อยู่ในสถานะใด (ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14	-	AUTO FEEDXT	อินพุท	งานหรือไม่ใช้งาน โดยที่สัญญาณเป็น "1" หมายถึง เครื่องพิมพ์อยู่ในสถานะที่ใช้งานอยู่ถ้าเป็น "0" อยู่ในสถานะที่ไม่ใช้งานหรือไม่สามารถรับข้อมูลได้ เป็นสัญญาณที่ส่งมาจากอุปกรณ์ภายนอก เมื่อเครื่องพิมพ์ได้รับสัญญาณนี้ เครื่องพิมพ์จะเลื่อนบรรทัดใหม่ 1 บรรทัดหลังจากที่พิมพ์เสร็จ หมายเหตุ สัญญาณนี้ เครื่องพิมพ์บางเครื่องจะไม่มี
15	-	NC	-	ไม่ได้ต่อใช้งาน
16	-	OV	-	ขั้ว 0 โวลท์
17	-	CHASSIS GND	-	ต่อกับโครงของ เครื่องพิมพ์
18	+5V	-	-	ขั้ว +5 โวลท์ (บางเครื่องไม่มี)
19-30	-	GND	-	สัญญาณกลับหรือกราวด์ของระบบ
31	-	INIT	อินพุท	เป็นสัญญาณที่ส่งจากอุปกรณ์ภายนอก โดยจะทำงานเมื่อมีสัญญาณ "LOW" เมื่อเครื่องพิมพ์ได้รับสัญญาณนี้ เครื่องพิมพ์จะเลื่อนหัวพิมพ์สู่จุดเริ่มต้นพิมพ์ ข้อมูลต่าง ๆ จะถูกละทิ้งหมด โดยทั่วไปแล้วสัญญาณนี้ จะมีความกว้างของพัลส์ประมาณ 50 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32	-	ERROR	เอาที่พูด	เป็นสัญญาณที่ส่งออกจากเครื่องพิมพ์เพื่อบอกให้อุปกรณ์ภายนอกรับรู้ว่ามีข้อผิดพลาดเกิดขึ้นเช่น -กระดาษหมด -อยู่ในระหว่าง OFF LINE -ข้อมูลที่รับเข้ามาผิดพลาด ไม่สามารถตีความได้
33	-	GND	-	เหมือน 19-30
34	-	NC	-	ไม่ได้ต่อใช้งาน
35	-	NC	-	ไม่ได้ต่อใช้งาน
36	-	SECT IN	-	คล้ายกับสัญญาณ BUSY แต่สัญญาณกลับกัน

ตารางที่ 2 ความสัมพันธ์ของขาและสัญญาณที่ส่งผ่านการอินเตอร์เฟสแบบขนาน

อนึ่ง ขั้วสัญญาณที่ 11 BUSY จะแสดงภาวะ "BUSY" เนื่องจากสาเหตุดังนี้

1. เครื่องพิมพ์อยู่ในระหว่างกระทำข้อมูลที่รับเข้าไป
2. อยู่ในระหว่างการพิมพ์ข้อมูล
3. อยู่ในระหว่าง OFF-LINE
4. ข้อมูลที่ส่งเข้ามาไม่อยู่ในวิสัยที่ตีความได้ เช่น ข้อมูลผิดพลาดหรือใช้รหัสไม่ถูกต้อง

และขั้วสัญญาณที่ 13 จะมีสถานะเกิดขึ้นเนื่องจาก

1. สถานะที่ใช้งานได้ (SELECT STATE)

-มีการกดปุ่ม SELECT หรือ ON LINE ในขณะที่อยู่ในสถานะ DESELECT หรือ OFF LINE

2. สถานะที่ไม่ใช้งาน (DESELECT STATE)

-มีการกดปุ่ม SELECT หรือ ON LINE ในขณะที่อยู่ในสถานะ SELECT หรือ ON LINE

-ได้รับรหัส DC3 (13H)

-เมื่อกระดาษพิมพ์หมด

-เมื่อเครื่องพิมพ์อยู่ในสถานะ FAULT

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การสร้างและการทำงานของ CENTRONIC PRINTER BUFFER

3.1 วงจรและการทำงานของ CENTRONIC PRINTER BUFFER

จากรูปที่ 9 เป็นวงจรส่วน INPUT SECTION ของ CENTRONIC PRINTER BUFFER โดยสัญญาณต่าง ๆ ที่จะใช้ในการติดต่อกับคอมพิวเตอร์ จะมีดังนี้

BUSY : เป็นสัญญาณที่ส่งจาก CENTRONIC PRINTER BUFFER ไปยังคอมพิวเตอร์เพื่อบ่งบอกถึงสถานะของตัว CENTRONIC PRINTER BUFFER ว่าพร้อมที่จะรับข้อมูลจากคอมพิวเตอร์หรือไม่ โดยถ้า

BUSY = "1" แสดงว่า CENTRONIC PRINTER BUFFER ไม่พร้อมจะรับข้อมูล

BUSY = "0" แสดงว่า CENTRONIC PRINTER BUFFER พร้อมจะรับข้อมูล

STB : เป็นสัญญาณจากคอมพิวเตอร์ส่งมายัง CENTRONIC PRINTER BUFFER เพื่อบ่งบอกว่าคอมพิวเตอร์ได้ส่งข้อมูลให้กับ CENTRONIC PRINTER BUFFER แล้ว

ACKNLG : เป็นสัญญาณจาก CENTRONIC PRINTER BUFFER ส่งไปยังคอมพิวเตอร์เพื่อบ่งบอกว่า ขณะนี้ CENTRONIC PRINTER BUFFER รับข้อมูลไว้เรียบร้อยแล้ว

สัญญาณในการติดต่อกันระหว่าง CENTRONIC PRINTER BUFFER กับ คอมพิวเตอร์ จะใช้ฮาร์ดแวร์ (HARDWARE) เป็นตัวสร้างสัญญาณเหล่านี้ขึ้น ซึ่งมีลักษณะการทำงานดังนี้

ในช่วงแรกเมื่อเริ่มการทำงาน สัญญาณรีเซ็ต (POWER ON RESET) ของระบบ ซึ่งแอกทีฟที่โลจิก "1" (ACTIVE HIGH) ผ่านอินเวอร์เตอร์ (INVERTER) U4B เข้ามายังแอนด์เกต (AND GATE) U5A ส่วนอีกอินพุตหนึ่งของแอนด์เกต U5A นั้นจะต่ออยู่กับ PSEL (PORT SELECT) ขณะนี้มีสถานะโลจิก "1" เนื่องจาก ซีพียู ยังไม่ได้ติดต่อกับพอร์ตนี้นี้ ดังนั้นเอาต์พุตของแอนด์เกต U5A จะมีสถานะโลจิก "0" ทำให้ ดี - ฟลิปฟลอป (D-FLIP FLOP) U3A ซึ่งขา CLEAR ต่ออยู่กับเอาต์พุตของแอนด์เกต U5A ให้เอาต์พุต Q มีสถานะโลจิก "0" สัญญาณ BUSY ซึ่งต่ออยู่กับเอาต์พุต Q ของ ดี-ฟลิปฟลอป U3A นี้ก็จะเป็นโลจิก "0" ซึ่งเป็นการแสดงว่าขณะนี้ CENTRONIC PRINTER BUFFER พร้อมที่จะรับข้อมูลจากคอมพิวเตอร์แล้ว

เมื่อคอมพิวเตอร์ส่งข้อมูลให้กับ CENTRONIC PRINTER BUFFER สัญญาณ STB ซึ่งต่ออยู่

กับขา CLOCK ของ U2 (IC # 74LS374) เพื่อ แลทช์ (LATCH) ข้อมูลไว้ ขณะเดียวกัน

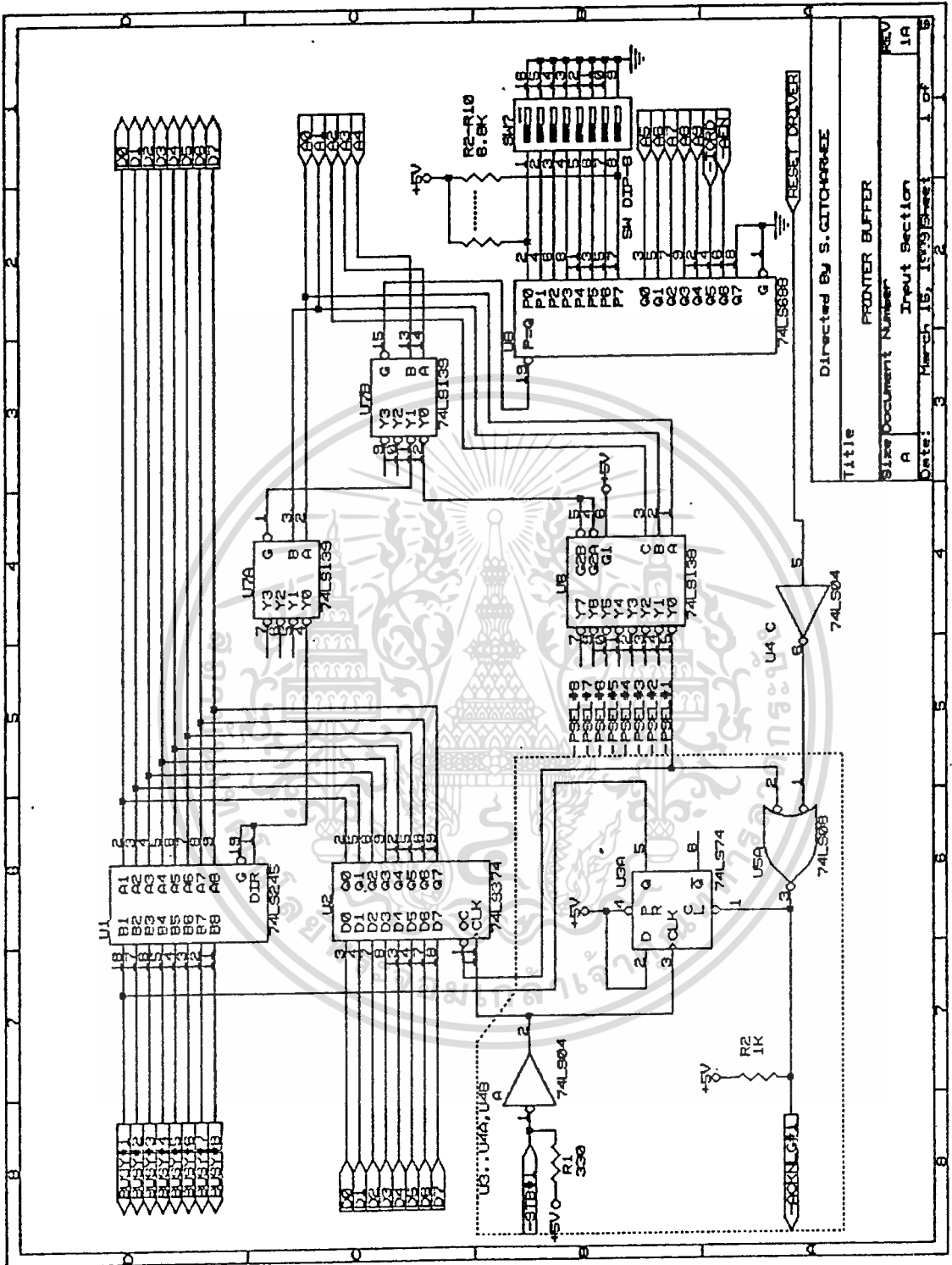
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์อื่น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ STB นี้ก็จะเป็น CLOCK ให้ ดี - ฟลิปฟลอป U3A ด้วยทำให้เอาท์พุทของ ดี - ฟลิปฟลอป ซึ่งต่อเป็นสัญญาณ BUSY มีสถานะลอจิก "1" เพื่อส่งเป็นสัญญาณ INPUT BUFFER FULL ให้กับคอมพิวเตอร์ นอกจากนี้สัญญาณดังกล่าว ยังใช้เป็นสัญญาณบอกสถานะของส่วนรับข้อมูลจากคอมพิวเตอร์ ทั้ง 8 เครื่อง โดยสัญญาณ BUSY จาก ดี - ฟลิปฟลอป ทั้ง 8 ชุด จะถูกนำมาเป็นข้อมูลบอกสถานะทางพอร์ต 308H (IC # 74LS245 U1) ข้อมูลสถานะของพอร์ต จะถูกตรวจสอบอยู่ตลอดเวลา โดยในส่วนของโปรแกรม เมื่อพบว่ามีคำสั่งส่งข้อมูลจากคอมพิวเตอร์เข้ามา โปรแกรมจะทำการอ่านข้อมูลจากส่วนรับข้อมูล (PORT 300 - 307H) ซึ่งถูกแลทซ์ไว้ที่ IC # 74LS374 U2 อยู่ก่อนเข้ามาแล้ว (ในกรณีที่มีการส่งข้อมูลเข้ามามากกว่า 2 เครื่องในเวลาเดียวกัน โปรแกรมจะทำการจัด PRIORITY ให้กับคอมพิวเตอร์แต่ละเครื่อง)

ขณะเดียวกันสัญญาณ PSEL (PORT SELECT) ซึ่งต่ออยู่กับแอนด์เกต U5A จะเป็นสถานะลอจิก "0" เนื่องจากมีการอ่านพอร์ต (READ PORT) ทำให้เอาท์พุทของแอนด์เกต U5A ซึ่งต่อเป็นสัญญาณ ACKNLG จะมีสถานะเป็นลอจิก "0" ซึ่งเป็นการแสดงว่า ขณะนี้เครื่องพิมพ์ รับข้อมูลจากคอมพิวเตอร์ไว้เรียบร้อยแล้ว ขณะเดียวกันสัญญาณนี้ ก็จะไปเคลียร์ ดี - ฟลิปฟลอป ทำให้ BUSY เป็น "0" พร้อมทั้งจะรับข้อมูลใหม่เข้ามา

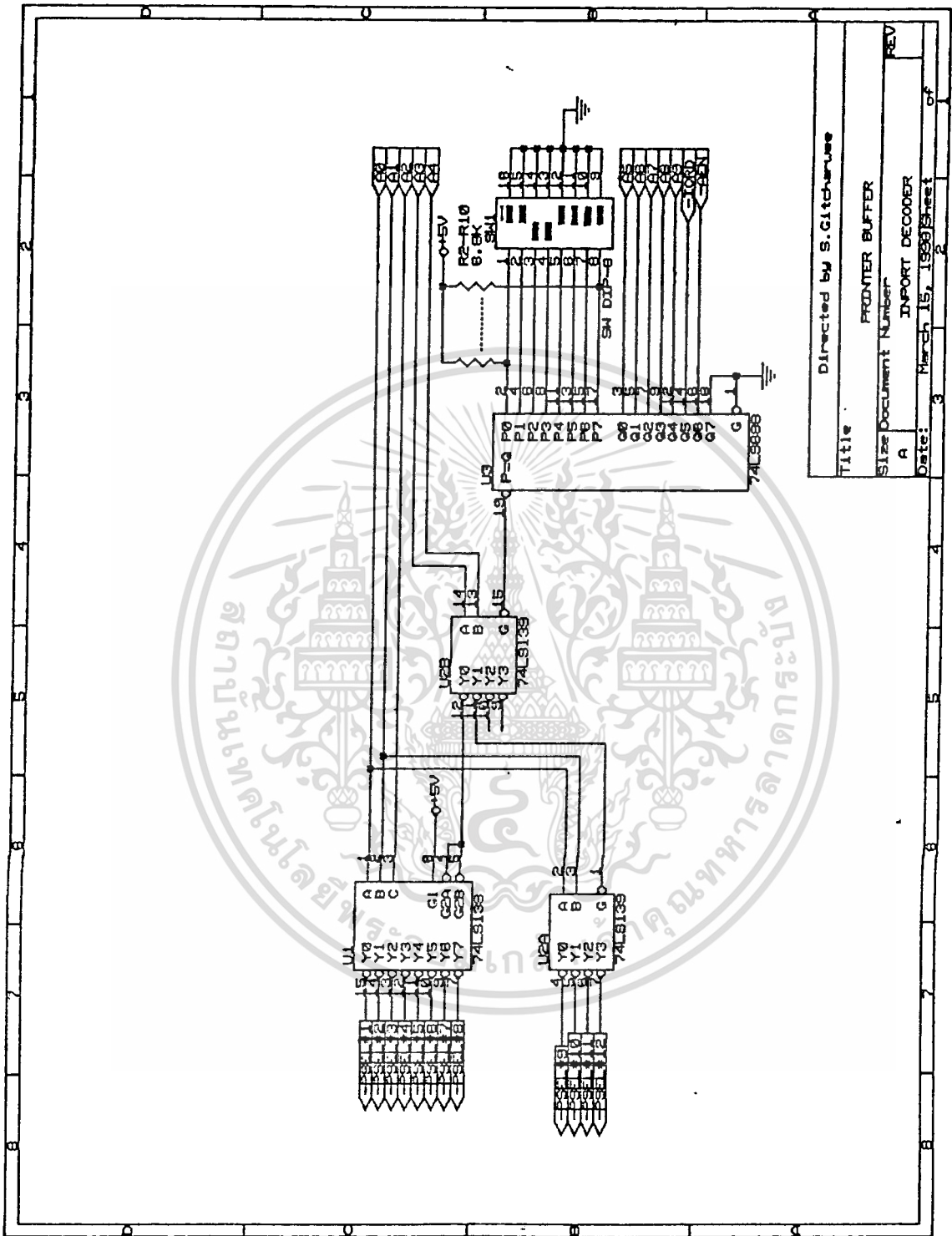
รูบที่ 10 , รูบที่ 11 ,รูบที่ 12 เป็นส่วนรายละเอียดในแต่ละส่วนของรูบที่ 9 ซึ่งการทำงานเป็นดังที่กล่าวมาแล้วข้างต้น โดย รูบที่ 10 เป็นส่วนการทำ HANDSHANK ,รูบที่ 11 เป็นส่วน INPORT DECORDER ,รูบที่ 12 เป็นส่วนการจัด PRIORITY



Title FRONTIER BUFFER
 Size Document Number A
 Input Section 1A
 Date: March 15, 1979 Sheet 1 of 3

รูปที่ 9 เป็นวงจรส่วน INPUT SECTION ของ CENTRONIC PRINTER BUFFER

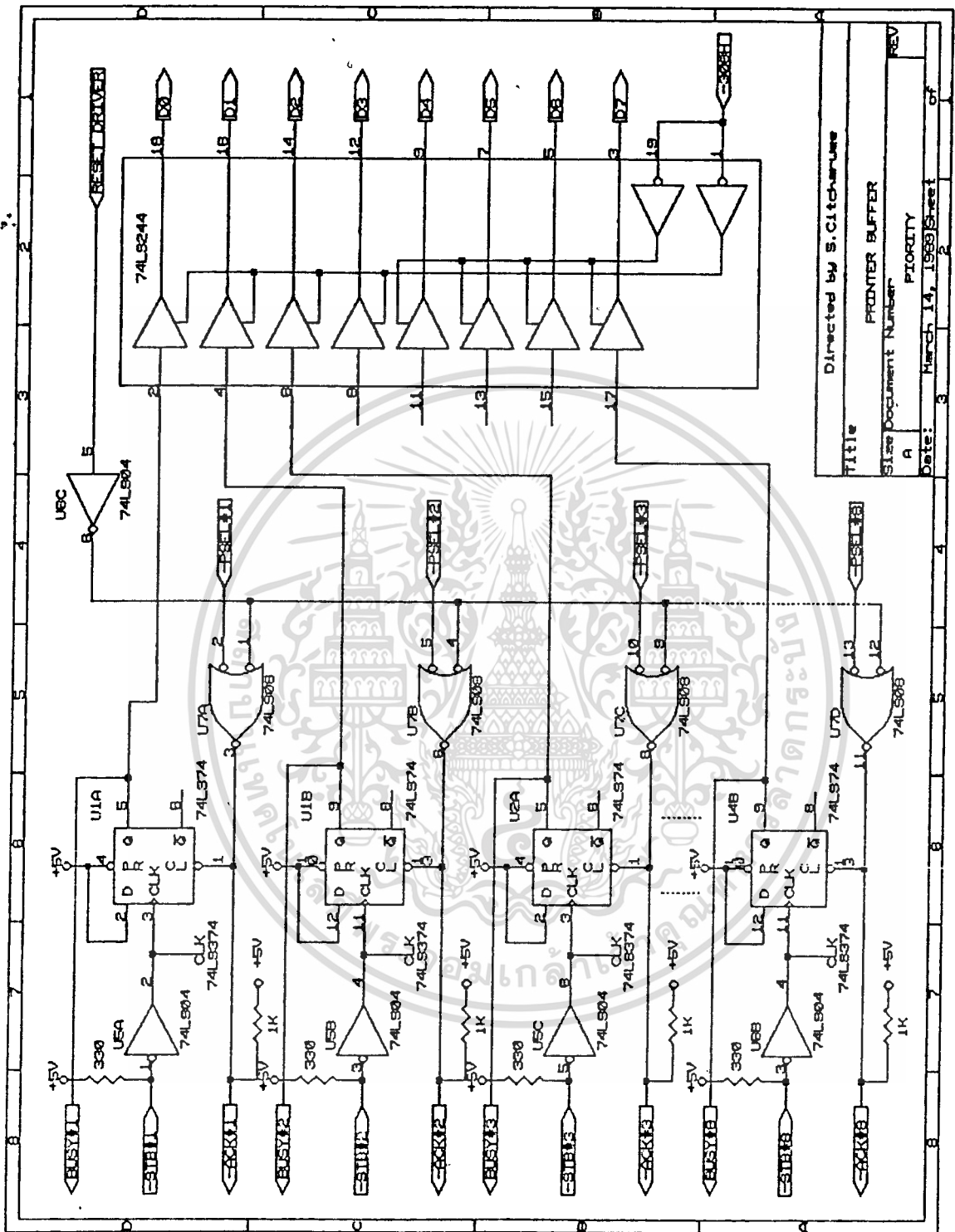
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Directed by S. Citcharusee	
Title	FRONTIER BUFFER
Size Document Number	A
Date	March 15, 1999
Sheet	2
of	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **รูปที่ 11** เป็นส่วน INPORT DECODER ที่นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 เป็นส่วนการจัด PRIORITY (เป็นส่วนรายละเอียดในแต่ละส่วนของรูปที่ 9)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาเท่านั้น ผิดลิขสิทธิ์หากมีการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

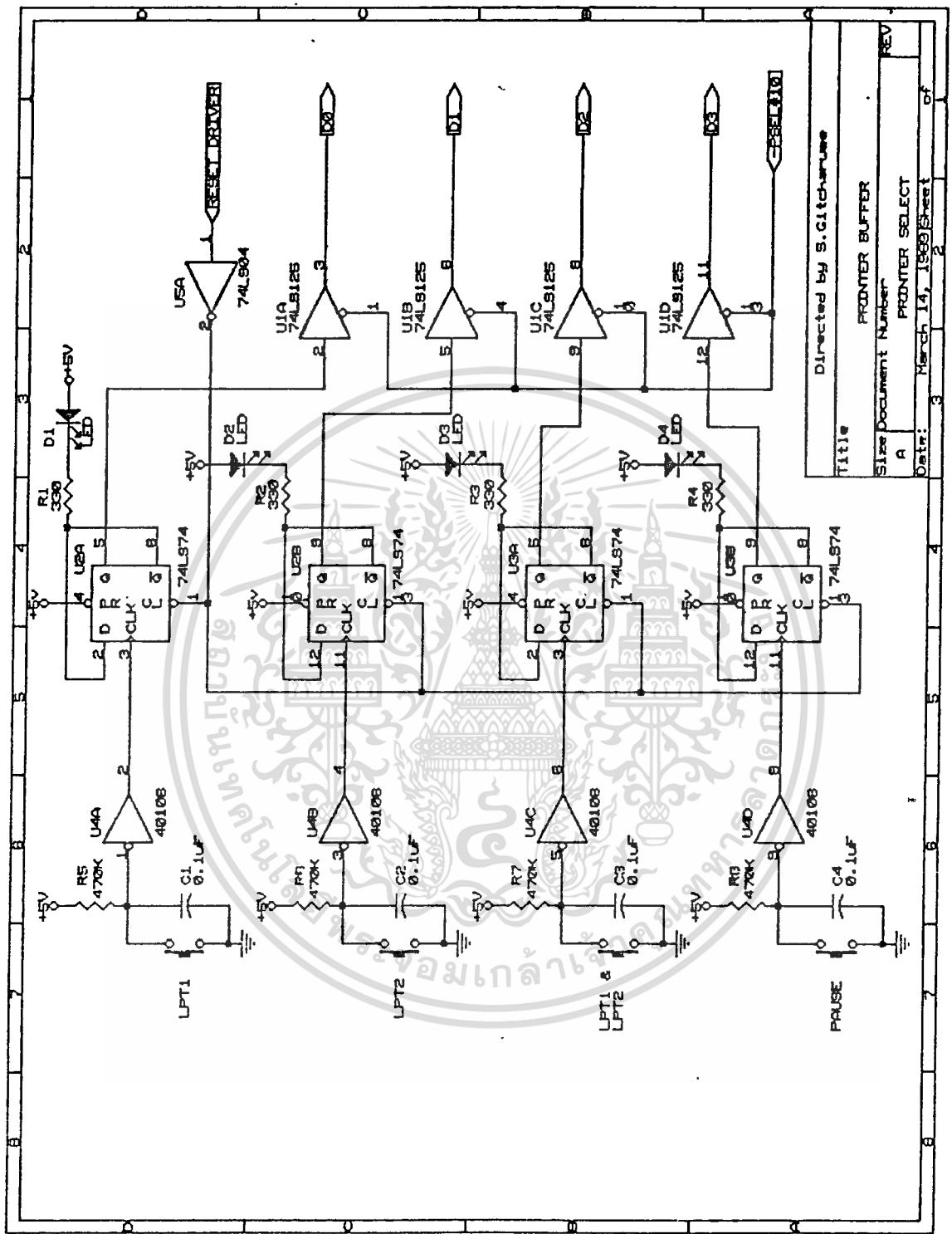
3.2 การทำงานของวงจรรภาค PRINTER SELECT

ดังรูปที่ 13 แสดงวงจรของส่วน PRINTER SELECT ที่ใช้เลือก ฟังก์ชัน (FUNCTION) ต่าง ๆ โดยการกดสวิตช์ ซึ่งมีฟังก์ชันที่เลือกได้ คือ

1. ถ้ากดสวิตช์ LPT1 จะส่งข้อมูลไปพิมพ์ที่เครื่องพิมพ์ตัวที่ 1
2. ถ้ากดสวิตช์ LPT2 จะส่งข้อมูลไปพิมพ์ที่เครื่องพิมพ์ตัวที่ 2
3. ถ้ากดสวิตช์ LPT1&LPT2 จะส่งข้อมูลไปพิมพ์ที่เครื่องพิมพ์ตัวที่ 1 และ 2 โดยสลับกันพิมพ์ทีละใบ
4. ถ้ากดสวิตช์ PAUSE จะหยุดส่งข้อมูลออกพิมพ์ชั่วคราว

การทำงานของวงจรมี โดยเมื่อเริ่มการทำงาน (POWER ON) จะมีสัญญาณรีเซ็ต(RESET DRIVER) จากระบบซึ่งเป็นลอจิก "1" เข้ามาทางขา 1 ของ IC#74LS04 (INVERTER) จะทำให้เอาต์พุตออกที่ขา 2 เป็นลอจิก "0" แล้วต่อเข้าสู่ขาเคลียร์ (CL) ของ ดี - ฟลิปฟลอป U2A,U2B,U3A,U3B ซึ่งจะทำให้เอาต์พุตที่ขา Q ของ ดี - ฟลิปฟลอป ทั้ง 4 ตัว เป็น "0" เอาต์พุต Q นี้จะต่อไปยังอินพุตของ U1A,U1B,U1C,U1D (ขา 2,5,9,12 ของ IC# 74LS125) ที่เป็นตัวแลทช์ข้อมูล (QUAD 3 STATE BUFFER) ซึ่งข้อมูลทางอินพุตนี้จะออกสู่เอาต์พุตได้ก็ต่อเมื่อขา PSEL เป็นลอจิก "0" เท่านั้น และเอาต์พุตทั้ง 4 จะต่อไปเป็นข้อมูลในบิต D0,D1,D2,D3 ที่จะถูกอ่านโดยผ่านพอร์ต 309H

เมื่อ ดี - ฟลิปฟลอป ทั้ง 4 ตัวถูกเคลียร์แล้ว เมื่อมีการกดปุ่มสวิตช์ใด ๆ ที่ปกติจะถูกต่อเป็นลอจิก "1" (+5V) อยู่ จะทำให้ลอจิกกลายเป็น "0" ซึ่งเมื่อผ่านแต่ละอินพุตของ IC# 40106 (HEX SCHMITT TRIGGER) ที่ต่ออยู่กับสวิตช์นั้น ๆ ลอจิกก็จะเปลี่ยนเป็น "1" และเข้าขา CK ของ ดี - ฟลิปฟลอป แต่ละตัว เอาต์พุต Q ก็จะเป็น "1" และเมื่อผ่านตัวแลทช์ข้อมูล 74LS125 แต่ละตัวจะกลายเป็นข้อมูลตามแต่ละบิต ซึ่งถ้ากด LPT1 บิต D0 จะเป็น 1 , กด LPT2 บิต D1 จะเป็น 1 , กด LPT1&LPT2 บิต D2 จะเป็น 1 , และเมื่อ กด PAUSE บิต D3 จะเป็น 1 ซึ่งข้อมูลในบิตเหล่านี้จะอ่านผ่านทางพอร์ต 309H เพื่อไปเลือกเครื่องพิมพ์ให้ตามต้องการ โดย โปรแกรม



Directed by S. Citcharuee
 Title PRINTER BUFFER
 Size Document Number A
 Date: March 14, 1999 Sheet 2 of 2

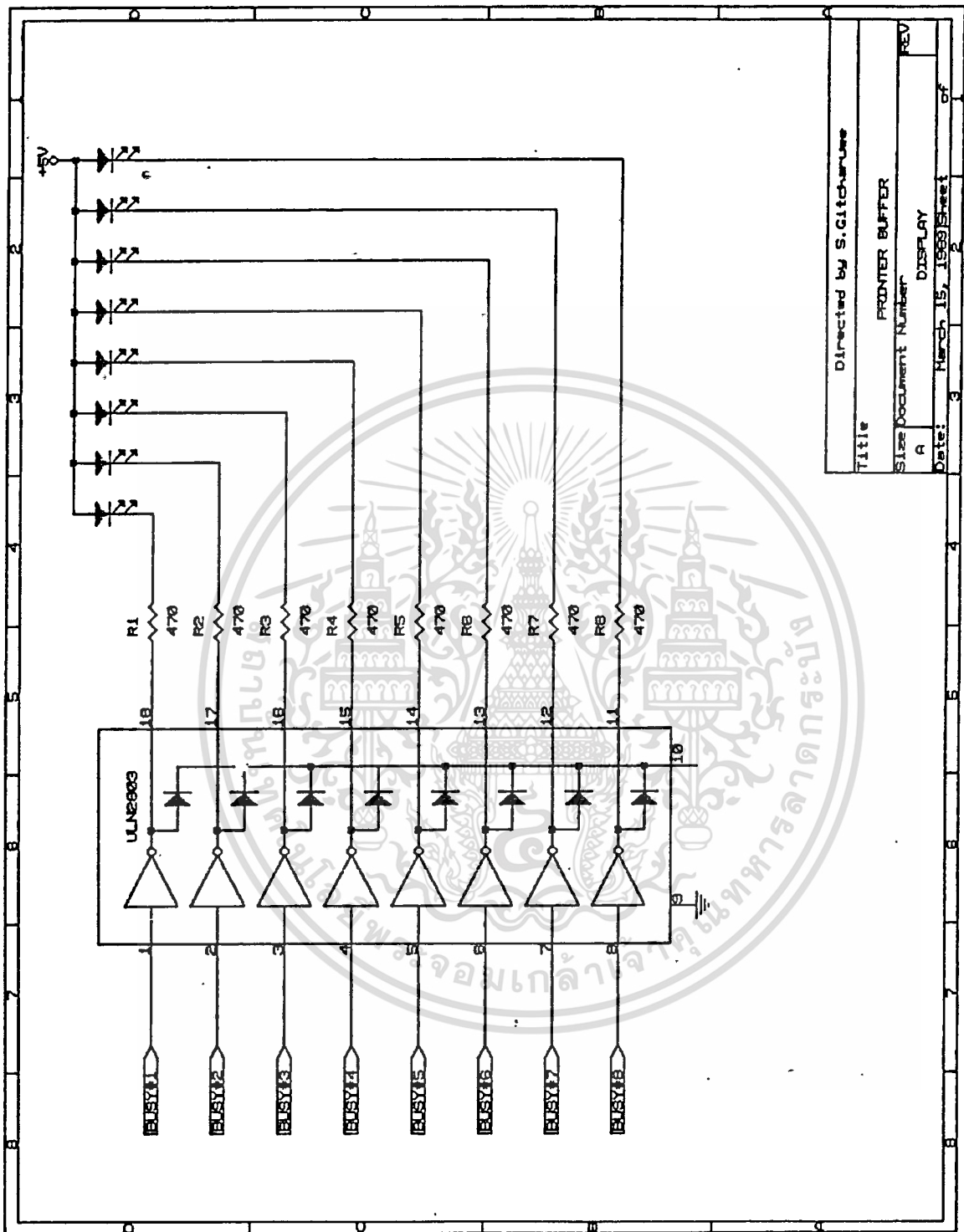
รูปที่ 13 แสดงวงจรของส่วน PRINTER SELECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลระบบใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทำงานของวงจรมภาค DISPLAY

จากรูปที่ 14 แสดงภาค DISPLAY โดยการทำงาน จะให้ขาอินพุตทั้ง 8 ของ IC#ULN 2803 รับสัญญาณ BUSY ที่ต่อมาจาก ดี - ฟลิปฟลอป ทั้ง 8 ในวงจรส่วน INPUT SECTION โดยถ้าคอมพิวเตอร์เครื่องใดต้องการส่งข้อมูลออกสู่เครื่องพิมพ์ จะมีสัญญาณ STROBE มาทริกที่ขา CK (CLOCK) ของ ดี - ฟลิปฟลอป ตัวที่ต่ออยู่ ทำให้มีเอาต์พุต Q เป็นลอจิก "1" แล้วจะส่งเข้ามาเป็นสัญญาณ BUSY สู่ภาค DISPLAY นี้ เมื่อผ่าน BUFFER DRIVE แต่ละตัวของ IC#ULN2803 ก็จะทำให้เอาต์พุตเป็น "0" จะไปไบอัส (BIAS) ให้กับ LED ที่ต่อทางขาแอนโอด (ANODE) ไว้ที่ +5V ซึ่งจะทำให้ LED ตัวนี้สว่าง โดยจะสว่างแบบกะพริบหรี่ ๆ ในขณะที่กำลังส่งออกพิมพ์ ส่วนถ้ามีคอมพิวเตอร์อีกตัวส่งข้อมูลเข้ามา ในขณะที่ตัวแรกกำลังส่งออกพิมพ์อยู่ LED ของตัวนี้จะสว่างตลอด ซึ่งหมายถึงข้อมูลยังไม่ได้ส่งออกพิมพ์ จนกระทั่งตัวแรกส่งหมด LED ของตัวหลังนี้จึงสว่างแบบกะพริบหรี่ ๆ เมื่อได้ส่งออกพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Directed by S.Citcharuee	
Title	FRONTIER BUFFER
Size	Document Number
A	DISPLAY
Date:	March 15, 1999
Sheet	2
of	2

รูปที่ 14 แสดงภาค DISPLAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของโปรแกรม

เริ่มต้นของโปรแกรม จะทำการเช็คคีย์สวิตช์ (KEY SWITCH) ว่ามีการกดคีย์สวิตช์หรือไม่ โดยถ้ามีการกดคีย์สวิตช์ ก็จะไปกระทำโปรแกรมน้อย CHECK KEY SWITCH โปรแกรมย่อยนี้ก็จะทำการตรวจสอบว่าคีย์สวิตช์ ที่กดเป็นคีย์อะไร และก็จะทำการกำหนดพอร์ตเครื่องพิมพ์ (PRINTER PORT) ตามคีย์ที่กด (LPT1, LPT2, LPT1&LPT2) ในกรณีที่ไม่มีมีการกดคีย์สวิตช์ในช่วงเวลาที่กำหนด จะถือว่าพอร์ตเครื่องพิมพ์ที่ใช้จะใช้ LPT1 (ในกรณีนี้ถือว่า LPT1 จะต่อไว้กับเครื่องพิมพ์ที่นิยมใช้กันมากในปัจจุบัน เช่น เครื่องพิมพ์ที่ใช้ KU CODE) เมื่อทำการกำหนดพอร์ตเครื่องพิมพ์ได้แล้ว ก็จะมาทำการเช็คดูว่า มีการส่งข้อมูลจากเครื่องคอมพิวเตอร์มายังบัฟเฟอร์หรือไม่ โดยทำการเช็ค BUSY (BUSY เกิดขึ้นโดยการจัดทำ HARD WARE ในลักษณะ HANDSHAKE กับเครื่องคอมพิวเตอร์ คือเมื่อคอมพิวเตอร์ส่ง STROBE เพื่อขอให้เครื่องพิมพ์ที่บัฟเฟอร์ก็จะทำการเช็ค BUSY ให้เป็น "1") ถ้าหากยังไม่มีการส่งข้อมูลเข้ามาก็จะทำการรอจนกว่าจะมีการส่งข้อมูลเข้ามา

เมื่อมีการส่งข้อมูลมาแล้ว ก็จะไปทำโปรแกรมน้อย PRIORITY BIT เพื่อทำการตรวจสอบว่า ข้อมูลที่รับเข้ามาที่บัฟเฟอร์ เป็นข้อมูลของ เครื่องคอมพิวเตอร์เครื่องใด เพื่อทำการจัดคาค่าพอร์ต (DATA PORT) ไว้สำหรับรับข้อมูลจากเครื่องคอมพิวเตอร์เครื่องนั้น โดยการจัดคาค่าพอร์ตให้กับเครื่องคอมพิวเตอร์ ก็จะจัดแบบเครื่องคอมพิวเตอร์เครื่องใดส่งข้อมูลมาก่อนก็จะรับเครื่องคอมพิวเตอร์เครื่องนั้นก่อน และในกรณีที่เครื่องคอมพิวเตอร์ 2 เครื่องขึ้นไปทำการส่งข้อมูลมาพร้อม ๆ กัน การจัดคาค่าพอร์ตก็จะจัดเรียงตามลำดับความสำคัญ โดยให้พอร์ตสุดท้าย (307H) เป็นพอร์ตที่มีความสำคัญสูงสุด

เมื่อกำหนดคาค่าพอร์ต ที่จะรับได้แล้วในส่วนโปรแกรมต่อมา ก็จะเป็นโปรแกรมการหน่วงเวลา เพื่อตรวจสอบการหยุดส่งข้อมูลจากเครื่องคอมพิวเตอร์เครื่องนั้น ซึ่งในส่วนของการหน่วงเวลานี้จะเช็คเวลาให้มีเวลานานพอที่เครื่องคอมพิวเตอร์จะส่งข้อมูลมาทำการประมวลผลข้อมูลที่ส่งเรียบร้อยแล้ว ในส่วนหน่วงเวลานี้จะตรวจสอบว่า หากในช่วงเวลาที่กำหนดไม่มีการส่งข้อมูลเข้ามา นั่นคือ BUSY BIT ของเครื่องคอมพิวเตอร์เครื่องนั้นเป็น "0" ก็จะถือว่าคอมพิวเตอร์เครื่องนั้นหยุดการส่งข้อมูลหรือจบไฟล์ (FILE) แล้ว และทำการนำเอาข้อมูลสุดท้ายที่รับเข้ามานั้น ตรวจสอบดูว่าเป็นรหัสของ FORM FEED หรือไม่หากเป็นของ FORM FEED ก็จะทำการส่งข้อมูลในบัฟเฟอร์ไปยัง เครื่องพิมพ์ แต่หากข้อมูลตัวสุดท้ายไม่ใช่รหัส FORM FEED เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะทำการใส่รหัส FORM FEED ให้ ทั้งนี้ก็เพื่อที่จะแยกการพิมพ์ของแต่ละไฟล์ออกจากกัน ในกรณีที่มีพิมพ์ข้อมูลออกที่กระดาษพิมพ์

ในกรณีที่เครื่องคอมพิวเตอร์ยังมีการส่งข้อมูลมาตลอด โปรแกรมในส่วน RECEIVE DATA ก็จะทำให้การรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์ โดยทำการเก็บข้อมูลที่เข้ามาลงในหน่วยความจำแบบคิว นั่นคือโปรแกรมในส่วนรับข้อมูลนี้จะทำการตรวจสอบว่า ตำแหน่งในหน่วยความจำที่ทำการจัดเก็บข้อมูลนั้นถึงตำแหน่งสูงสุดใน 1 เซกเมนต์ (SEGMENT, 0000-FFFFH) หรือยัง หากสูงสุดแล้วก็จะทำการเปลี่ยนเซกเมนต์ที่ทำการเก็บข้อมูล เป็นเซกเมนต์ถัดไป และเช็คตำแหน่งในการรับข้อมูลของเซกเมนต์ถัดไปนั้นเป็นตำแหน่งเริ่มต้น (ตำแหน่ง 0000H) ในการเปลี่ยนเซกเมนต์นั้น (ในกรณีที่ข้อมูลเต็ม 1 เซกเมนต์) ก็จะต้องมีการตรวจสอบว่าเซกเมนต์ที่เปลี่ยนนั้นเป็นเซกเมนต์สุดท้ายหรือยัง หากเป็นเซกเมนต์สุดท้ายแล้ว ก็จะต้องเปลี่ยนเซกเมนต์นั้นเป็นเซกเมนต์เริ่มต้นในการเก็บข้อมูลจากเครื่องคอมพิวเตอร์

สิ่งสำคัญที่จะต้องทำการตรวจสอบในการรับข้อมูล ก็คือว่า ตำแหน่งในการเก็บข้อมูลนั้น (DATA ADDRESS) ไปทับกับตำแหน่งที่จะเอาข้อมูลไปพิมพ์ (PRINT ADDRESS) หรือไม่ หากตำแหน่งที่รับข้อมูลนั้นไปทับกับตำแหน่งที่จะเอาข้อมูลไปพิมพ์จะต้องทำการนำเอาข้อมูลที่ตำแหน่งที่จะเอาข้อมูลไปพิมพ์นั้นส่งออกไปยังเครื่องพิมพ์ก่อน แล้วจึงทำการรับข้อมูลเข้ามาทับข้อมูลเก่าที่พิมพ์ออกไปแล้วได้ ในการกำหนดการส่งข้อมูลไปยังเครื่องพิมพ์ นอกจากการตรวจสอบว่าไม่มีการส่งข้อมูลจากเครื่องคอมพิวเตอร์แล้ว (ในกรณีจบไฟล์ หรือคอมพิวเตอร์หยุดส่งข้อมูล) เรายังต้องกำหนดค่าสูงสุดของตำแหน่งในการรับข้อมูล (M.RECIVE = MAXIMUM RECIVE) นั่นคือ ในกรณีที่คอมพิวเตอร์มีการส่งข้อมูล โดยมีไฟล์ที่ใหญ่มากเกินกว่าหน่วยความจำ หรือ ค่าสูงสุดของตำแหน่งในการรับข้อมูล ก็จะต้องทำการส่งข้อมูลไปยังเครื่องพิมพ์ก่อนแล้วจึงรับข้อมูลเข้ามาใหม่ ซึ่งสามารถนำข้อมูลใหม่นี้ไปเก็บในตำแหน่งของข้อมูลที่ถูกพิมพ์ไปแล้วในหน่วยความจำได้

ในส่วนของการส่งข้อมูลออกไปยังเครื่องพิมพ์ ก็จะทำให้การนำเอาพอร์ตเครื่องพิมพ์ (ที่ได้กำหนดไว้แล้วในการเช็คคีย์สวิตช์ ตอนเริ่มต้น) มาเป็นพอร์ตที่จะทำการส่งข้อมูลไปยังเครื่องพิมพ์ (ในที่นี้มีพอร์ต 37BH และ 3BCH) โดยที่พอร์ตเครื่องพิมพ์เป็นพอร์ต LPT1&LPT2 (จากการกดคีย์สวิตช์ตัวที่ 3) การส่งข้อมูลออกเครื่องพิมพ์ก็จะทำการส่งข้อมูลออกเครื่องพิมพ์ทีละไบต์ ออกไปยังพอร์ตเครื่องพิมพ์ 2 พอร์ต (LPT1&LPT2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

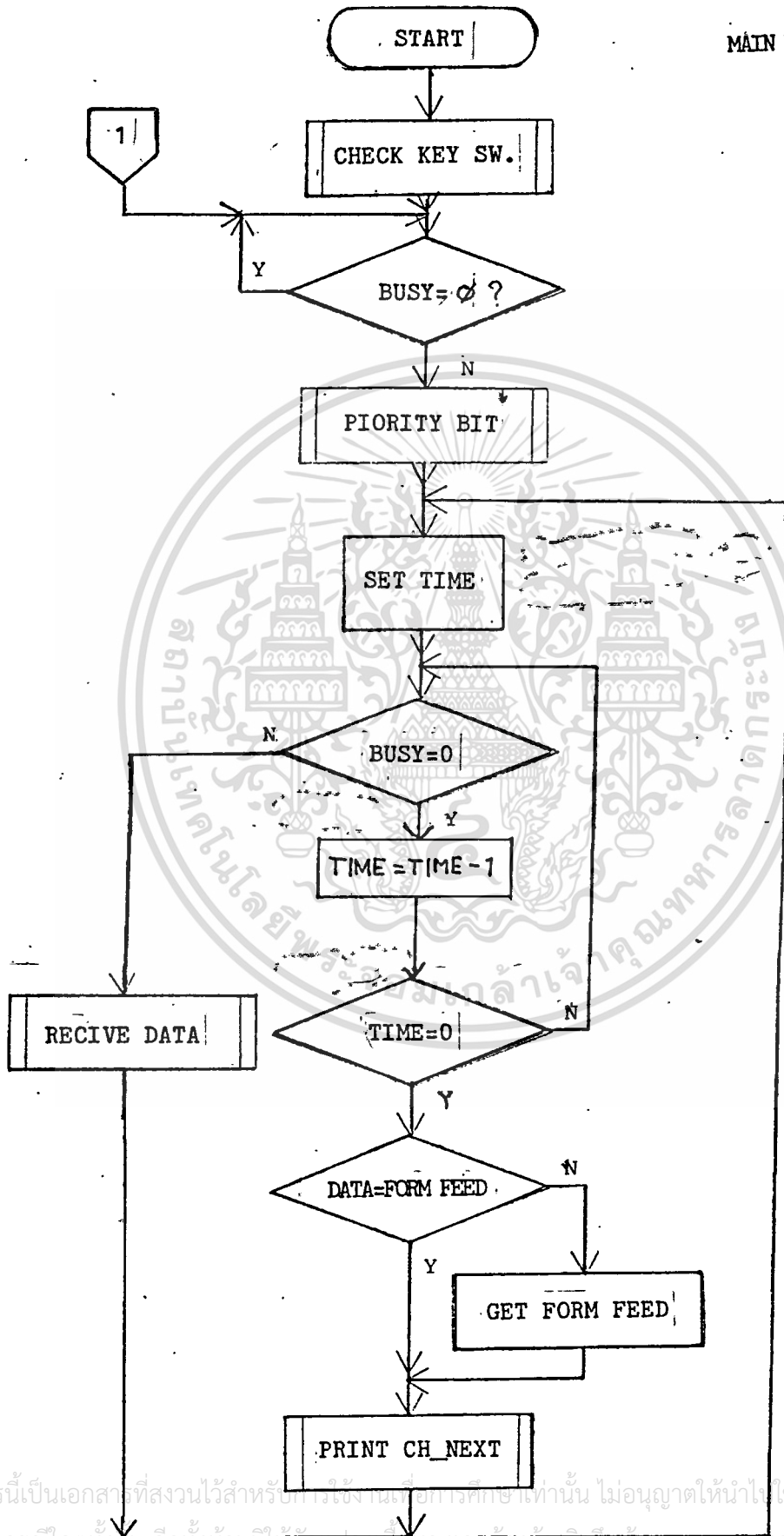
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลออกไปยัง เครื่องพิมพ์นี้ เราจะต้องนำเอาข้อมูลในตำแหน่งที่จะส่งออกไปยัง เครื่องพิมพ์ (A.PRT) ซึ่ง ในโปรแกรมก็จะมีตัวชี้ (POINTER) เป็นตัวชี้ตำแหน่งนี้ เมื่อส่งข้อมูล ไปยัง เครื่องพิมพ์แต่ละไบต์ ก็จะต้องทำการตรวจสอบว่าตำแหน่งในการส่งข้อมูลออก เครื่องพิมพ์ นั้นเป็นตำแหน่งสูงสุดใน 1 เซกเมนต์ และตัวชี้ก็ต้องเริ่มต้นชี้ที่ตำแหน่ง 0H ในเซกเมนต์ถัด ไปนั้นด้วย เช่นเดียวกับกับกรณีการรับข้อมูล คือ ต้องตรวจสอบว่า เซกเมนต์ที่เปลี่ยนนั้นเป็น เซกเมนต์สุดท้ายหรือไม่ หากเป็นเซกเมนต์สุดท้าย ก็จะเซ็ทเซกเมนต์ที่เปลี่ยนเป็นเซกเมนต์เริ่มต้น และในการส่งข้อมูลออก เครื่องพิมพ์แต่ละไบต์นั้น ก็จะมีการตรวจสอบว่าข้อมูลที่อยู่ในบัฟเฟอร์ นั้นถูกส่งพิมพ์หมดหรือยัง โดยการตรวจสอบตำแหน่งสุดท้ายในการรับข้อมูล (A.DAT) กับ ตำแหน่งเก็บข้อมูลที่ส่งออกพิมพ์ (A.PRT) หากเท่ากัน ก็แสดงว่าข้อมูลในบัฟเฟอร์ได้จัดส่งพิมพ์ หมดแล้ว (END BUF) การทำงานของโปรแกรมก็จะกลับไปเริ่มต้น โปรแกรมใหม่ คือ เริ่มต้น เซ็ท BUSY ใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

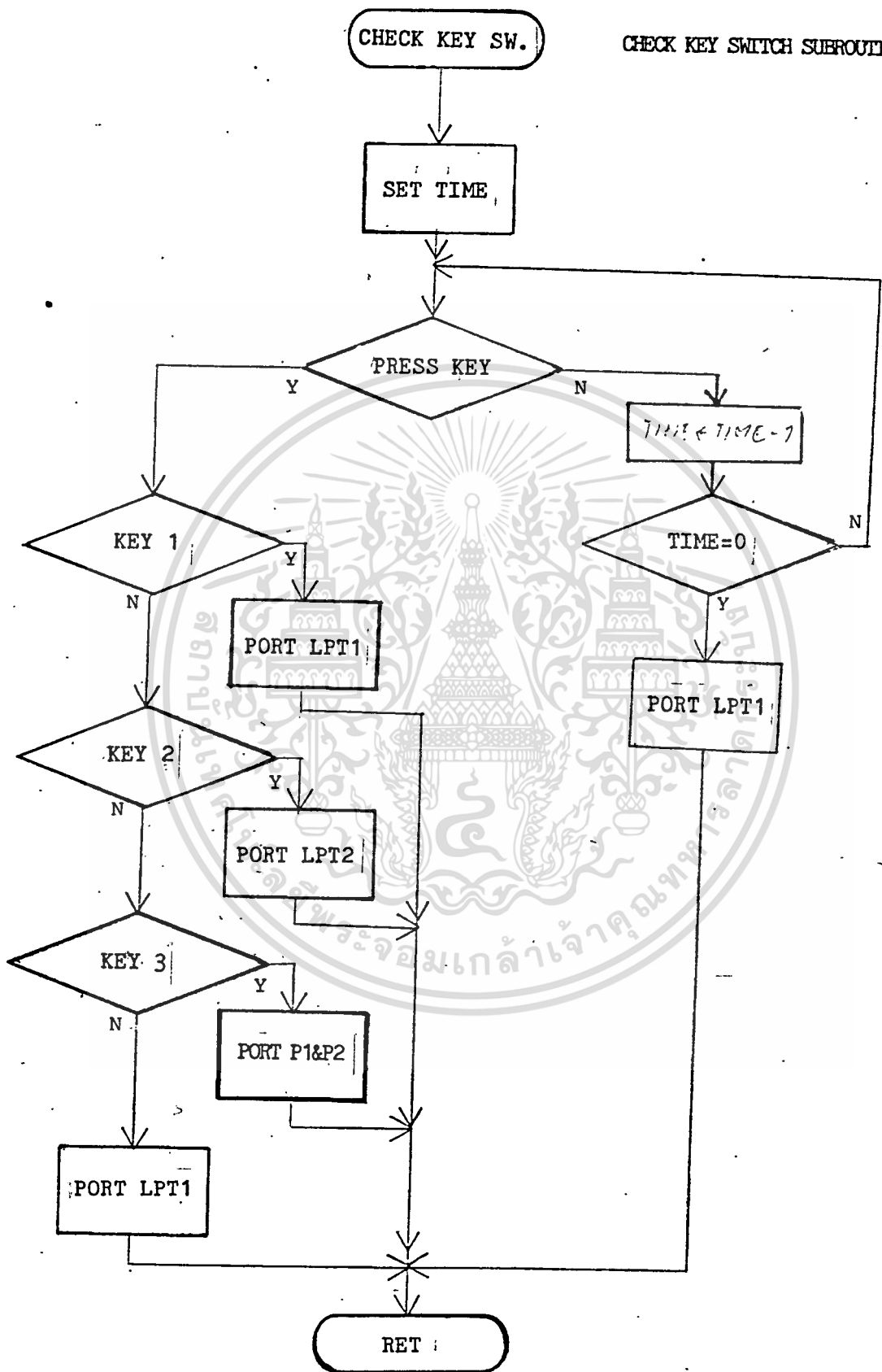
MAIN FLOW CHART



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นที่มามีเหตุที่เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHECK KEY SW.

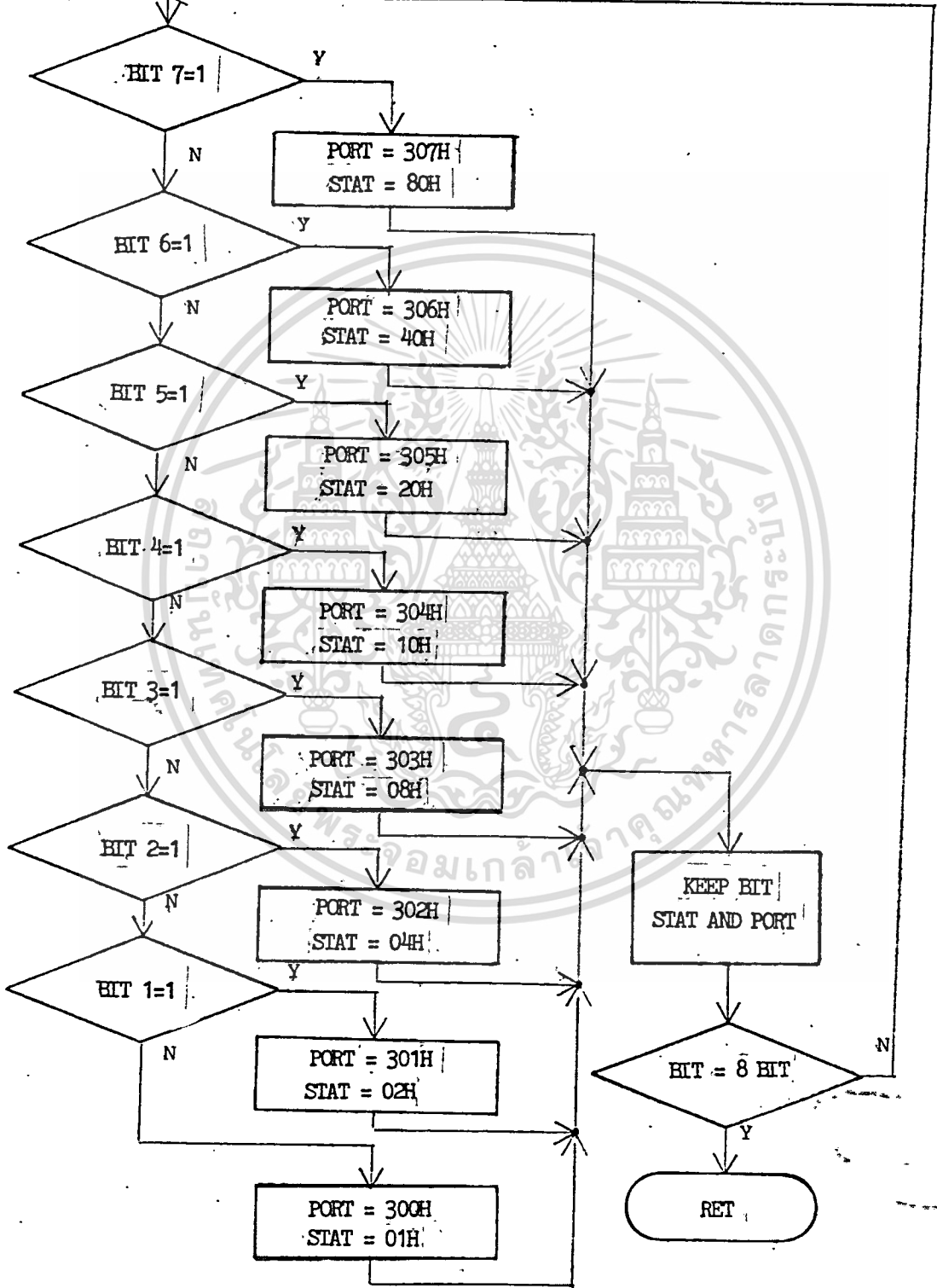
CHECK KEY SWITCH SUBROUTINE



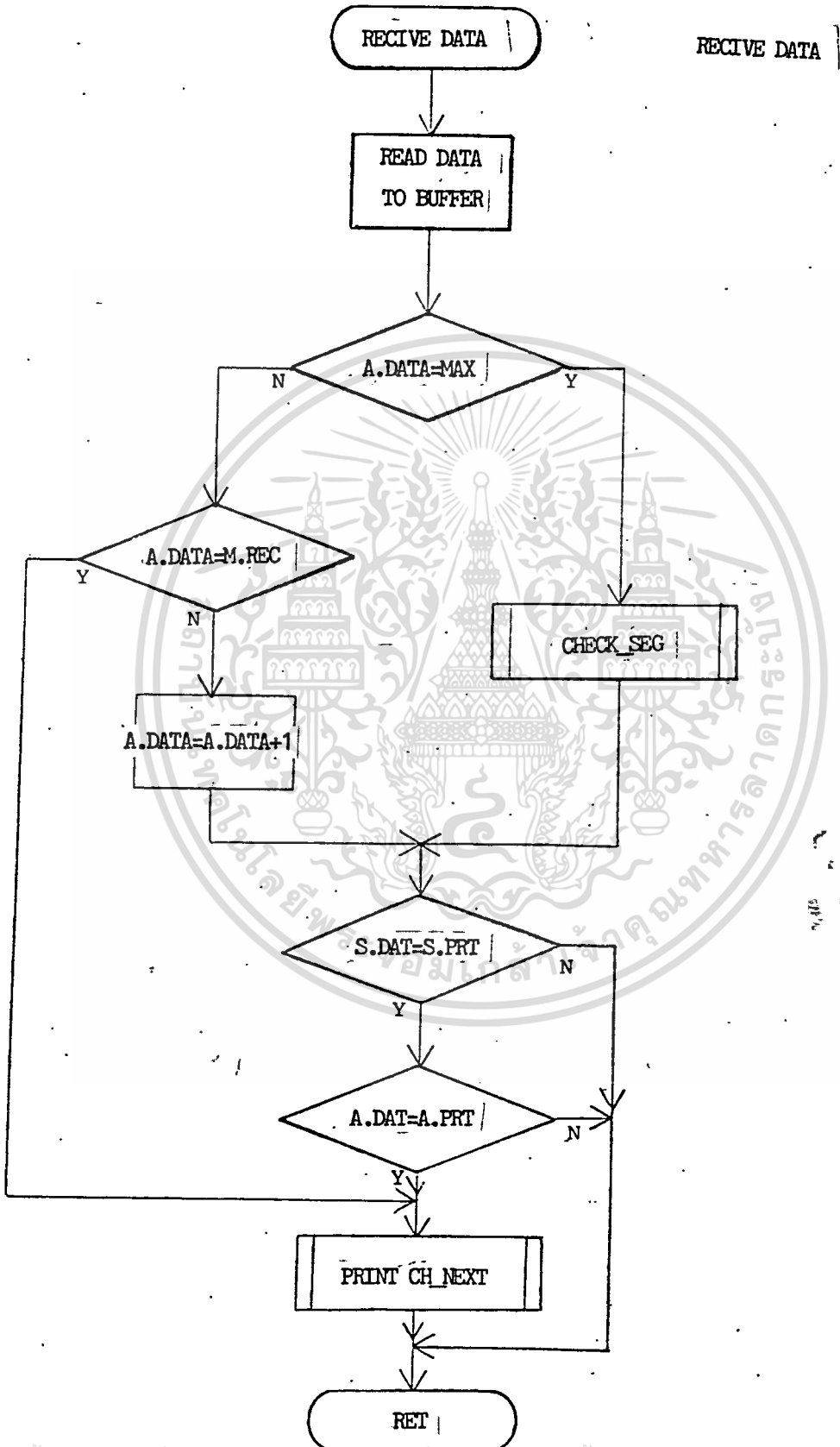
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRIORITY BIT //

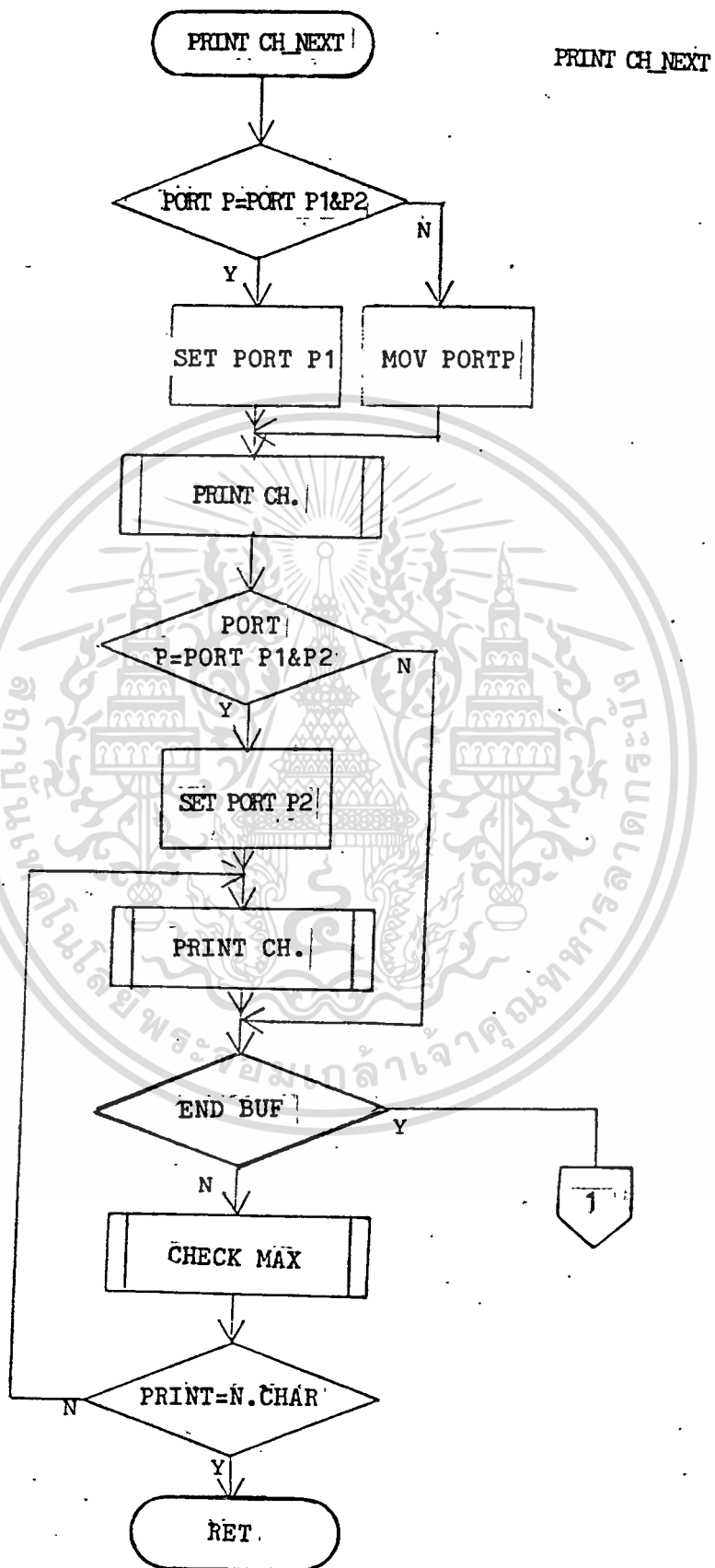
PRIORITY



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

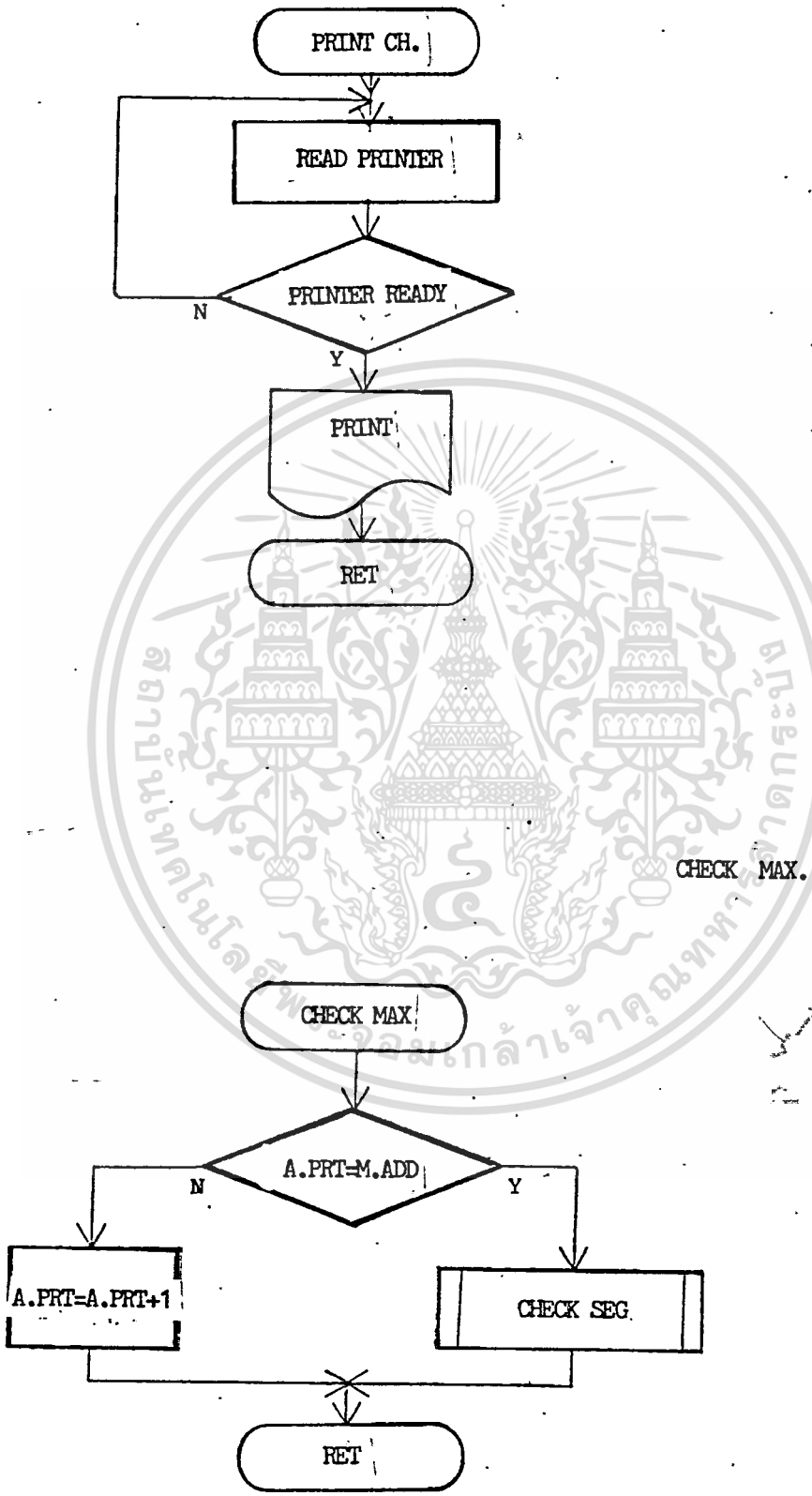


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

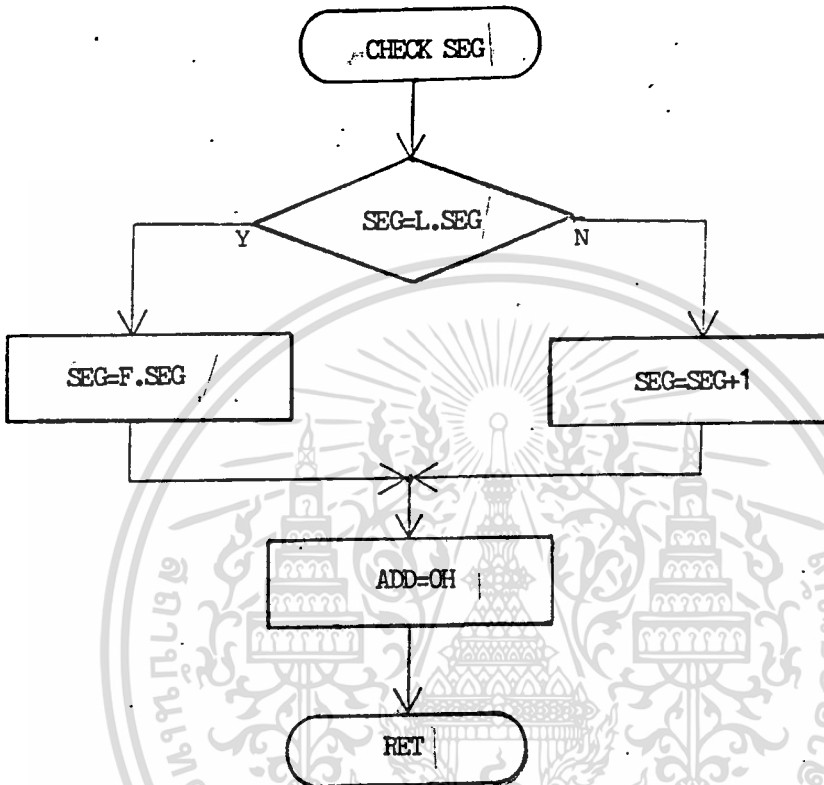


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRINT CH.



CHECK SEGMENT



page 60,132
title PRINTER BUFFER

comment *

program name: buffer.asm
Purpose : centronic printer spooler
Written by : Mr.Suthee gijchawee 28.6723
: Industrial computer technology KMIT.L
last update : 17:28:37 3/18/1989
*

00
00

pro segment
assume cs:pro,ds:pro
org 100h

00 E9 01BF R

JMP over

;-----data area-----

03
08

message DB 10,10,10,10,13
DB 26 DUP (),'CENTRONIC PRINTER BUFFER',10,10,13

00

)

43 45 4E 54 52 4F 4E
49 43 20 50 52 49 4E
54 45 52 20 42 55 46
46 45 52 0A 0A 0D

00
00

DB 15 DUP (),'THIS PROGRAM WAS WRITTEN BY Mr.Suthee gijchawee 28.6723',13,10

)

54 48 49 53 20 50 52
4F 47 52 41 4D 20 57
41 53 20 57 52 49 54
54 45 4E 20 42 59 20
4D 72 2E 53 75 74 68
65 65 20 67 69 6A 63
68 61 77 65 65 20 32
38 2E 36 37 32 33 0D
0A

00
00

DB 23 DUP (),10,10,10, '.....PLEASE RESET TO START.....'

)

0A 0A 0A 2E 2E 2E 2E
2E 50 4C 45 41 53 45
20 52 45 53 45 54 20
54 4F 20 53 54 41 52
54 2E 2E 2E 2E 2E

24

DB '\$'

over:

;-----WRITE TITLE-----

;set cursor to top left corner of screen

B4 B2
B7 00
BA 0000
CD 10

mov ah,02H ;function to set cursor
mov bh,0 ;page number
mov dx,0 ;coordinate 0,0
int 10h

;write the space chacters 2000 times

B4 09

mov ah,09H ;function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

01CA B9 07D0      mov cx,2000      ;number of time to write
01CD B8 20      mov al,' '      ;space character in al
01CF B3 07      mov bl,07H      ;normal attribute in bl
01D1 CD 18      int 10H

```

;write title

```

01D3 B4 0E      mov ah,14      ;function
01D5 B7 00      mov bh,0
01D7 B3 07      mov bl,07H      ;normal attribute in bl
01D9 BE 0103 R   mov si,offset messege
01DC 8A 04      _next: mov al,[si]  ;character in al
01DE 46      inc si          ;next character
01DF 3C 24      cmp al,'$'     ;end of date
01E1 74 04      je SPOOLER
01E3 CD 18      int 10H
01E5 EB F5      jmp _next

```

;-----PRINTER SPOOLER-----

```

01E7 SPOOLER:
01E7 E8 0473 R   call key_fun    ;recive key sw.

```

```

01EA nuki:
;set start segment at 0f000h-0f001h

```

```

01EA B8 0000      mov ax,0h
01ED BE D8      mov ds,ax
01EF B8 F000      mov bx,0f000h
01F2 B8 1000      mov ax,1000h
01F5 B9 0002      mov cx,02h
01F8 B8 27      xe: mov [bx],ah
01FA 43      inc bx
01FB B8 07      mov [bx],al
01FD 43      inc bx
01FE E2 FB      loop xe
0200 BB 1000      mov ax,1000h
0203 BE D8      mov ds,ax
0205 33 F6      xor si,si
0207 33 FF      xor di,di

```

;__Deffinition part__

```

= 0378      portp1      equ 378h      ;lpt1 port
= 03BC      portp2      equ 3bch      ;lpt2 port
= 0033      portp3      equ 033h      ;referent lpt1;lpt2 portr
= 0400      nyaber_cha equ 0400h      ;char. for send to printer
= 001D      time_1      equ 1dh      ;time delay for recive data
= 0300      time_2      equ 0300h      ;form computer
= 2000      timebit     equ 02000h      ;time delay for each computer
= 0000      bitstrobe  equ 0h      ;no data form computer
= FFFE      #rec       equ 0ffffh      ;max addr. for data reciver
= FFFF      #add       equ 0ffffh      ;max add in one segment
= 1000      fseg       equ 01000h      ;first segment
= 3000      lseg       equ 03000h      ;last segment
= F000      seg0       equ 0f000h      ;for segment data
= F002      seg0_1     equ seg0+2      ;for segment print.
= F004      keep_portp equ seg0+4      ;number port printer
= F008      add_keepbi equ seg0+8      ;addr. for number port data

```

นี้เป็นเอกสารที่สงวนไว้สำหรับใช้ศึกษา; นโยบายด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

= F00A          addbit      equ seg0+0ah      ;computer status when send data
= 0300          porta      equ 300h        ;busy status port
= 0309          portb      equ 309h        ;key sw.status port

;___Check busy begin___
0209          test_0:          ;check busy
0209 BA 0300          mov dx,porta
020C          test_1:
020C EC          in al,dx
020D 3C 00          cmp al,bitstrobe
020F 74 FB          jz test_1
0211 EB 04D9 R      call check_port
0214          check_r:
0214 BD 0000          mov bp,0h
0217          check_s:
0217 B7 00          mov bh,0h
0219          check_0:
0219 B9 0010          mov cx,time_1 ;check busy with time delay
021C          check_1:
021C 51          push cx
021D B9 FFFF          mov cx,0ffffh
;___check busy again___
0220 BA 0300          mov dx,porta
0223          zz:
0223 EC          in al,dx
0224 3C 00          cmp al,bitstrobe
0226 75 1A          jnz yx
0228 E2 F9          loop zz
022A 59          pop cx
022B E2 EF          loop check_1
022D E9 02FE R      jmp endf
0230          ab:
0230 51          push cx
0231 B9 0300          mov cx,time_2
0234          ac:
0234 BA 0300          mov dx,porta
0237 EC          in al,dx
0238 3C 00          cmp al,bitstrobe
023A 75 06          jnz yx
023C E2 F6          loop ac
023E 59          pop cx
023F E9 02FE R      jmp endf

;-----
; Process data reciver
; And
; Circular queue
;-----

;___when data come in___
0242          yx:
0242 59          pop cx
0243 50          push ax
0244 53          push bx
0245 51          push cx
0246 52          push dx

```

0246 52 นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ทางการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0247 1E          push ds
0248 B9 2000      mov cx,timebit ;time delay for recive data
024B          yx0:
024B BA 0308      mov dx,porta
024E EC          in al,dx
024F BB 0800      mov bx,0h
0252 BE DB       mov ds,bx
0254 BB F00A      mov bx,addbit
0257 BA 1F       mov bl,[bx]
0259 B4 D8       test bl,al ;test old busy
025B 75 0D       jnz yy1
025D E2 EC       loop yx0
025F EB 04D9 R    call check_port
0262 1F          pop ds
0263 5A          pop dx
0264 59          pop cx
0265 5B          pop bx
0266 58          pop ax
0267 EB 06 90     jmp yy
026A          yy1:
026A 1F          pop ds
026B 5A          pop dx
026C 59          pop cx
026D 5B          pop bx
026E 58          pop ax
026F          yy:
026F 80 FF 01      cmp bh,01h
0272 75 12       jnz p1
0274 B3 FE FF     cmp si,madd
0277 75 0C       jnz vv
0279 EB 03E2 R    call check_seg
027C EB 03B8 R    call keep_segd
027F BE 0000      mov si,0h
0282 EB 02 90     jmp p1
0285          vv:
0285 46          inc si
0286          p1:
0286 EB 03CE R    call used_segd
0289 EB 0585 R    call pdata_1
028C EC          in al,dx
028D B8 04       mov [si],al
```

;___check address data full segment___

```
028F B3 FE FF     cmp si,madd
0292 75 2D       jnz check_2
0294 E8 03E2 R    call check_seg
0297 E8 03B8 R    call keep_segd
029A EB 03FE R    call check_dp
029D B1 F9 AAAA    cmp cx,0aaaah
02A1 75 18       jnz check_3
02A3 B3 FF 00     cmp di,0h
02A6 75 13       jnz check_3
02A8 56          push si
02A9 E8 032E R    call print
02AC 5E          pop si
02AD BE 0000      mov si,0h
```

สงวนลิขสิทธิ์เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
02B0 B0 FC 00      cmp ah,0h
02B3 74 03          jz p2
02B5 E9 0214 R     jmp check_r
02B8                p2:
02B8 E9 01EA R     jmp nuki
02BB                check_3:
02BB BE 0000      mov si,0h
02BE E9 0214 R     jmp check_r
02C1                check_2:
02C1 E8 0427 R     call check_ep
02C4 3D 00AA      cmp ax,00aah
02C7 75 16        jnz p4
02C9 83 FD FE      cmp bp,mrec
02CC 75 11        jnz p4
02CE 56           push si
02CF EB 032E R     call print
02D2 5E           pop si
02D3 80 FC 00      cmp ah,0h
02D6 74 04        jz p3
02D8 46           inc si
02D9 E9 0214 R     jmp check_r
02DC                p3:
02DC E9 01EA R     jmp nuki
02DF                p4:
02DF 46           inc si
02E0 45           inc bp
02E1 E8 03FE R     call check_dp
02E4 81 F9 AAAA   cmp cx,0aaaah
02E8 75 0E        jnz eb
02EA 3B F7        cmp si,di
02EC 75 0A        jnz eb
02EE 56           push si
02EF EB 032E R     call print
02F2 5E           pop si
02F3 80 FC 00      cmp ah,0h
02F6 74 03        jz p5
02FB                eb:
02FB E9 0217 R     jmp check_s
02FB                p5:
02FB E9 01EA R     jmp nuki
```

```
;-----  
; Process check end file  
; And  
; Get form feed code  
;-----
```

```
;___when no data come in___  
endf:  
02FE 80 FF 00      cmp bh,0h  
0301 75 1B        jnz p6  
0303 8B DE        mov bx,si  
0305 83 EB 01      sub bx,01h  
0308 BA 07        mov al,[bx]  
030A 3C 0C        cmp al,0ch  
030C 74 05        jz px  
030E 8B 0C        mov al,0ch
```

สามารถเป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0310 88 04      mov [si],al
0312 46         inc si
0313          px:
0313 80 0D      mov al,0dh
0315 88 04      mov [si],al
0317 46         inc si
0318 80 0A      mov al,0ah
031A 88 04      mov [si],al
031C B7 01      mov bh,1h
031E          p6:
031E 56         push si
031F EB 032E R   call print
0322 5E         pop si
0323 80 FC 00    cmp ah,0h
0326 74 03      jz p7
0328 E9 0230 R   jmp ab
032B          p7:
032B E9 01EA R   jmp nuk1
```

```
-----
; Process print data
; To
; Printer
;-----
```

```
032E          ;__print data from buffer__
032E B9 0400    print: mov cx,number_cha
0331          check_i:
0331 51         push cx
0332 EB 045F R   call used_segp
0335 EB 04C3 R   call used_portp
0338 83 FA 33    cmp dx,portp3
033B 74 06      jz pp
033D E8 039D R   call p_numch
0340 EB 07 90    jmp ppl
0343          pp:
0343 BA 0378     mov dx,portp1
0346 EB 039D R   call p_numch
0349          ppl:
0349 E8 04C3 R   call used_portp
034C 83 FA 33    cmp dx,portp3
034F 74 03      jz pp3
0351 EB 07 90    jmp yjb
0354          pp3:
0354 BA 03BC     mov dx,portp2
0357 EB 039D R   call p_numch
```

```
-----
;__check code end file__
yjb:
```

```
035A          call check_dp
035A E8 03FE R   cmp cx,0aaaah
035D 81 F9 AAAA jnz p8
0361 75 22     mov al,[di]
0363 8A 05     cmp al,0ch
0365 3C 0C     jnz p8
0367 47         inc di
```

เอกสารที่ส่งวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
036A 8A 05          mov al,[di]
036C 3C 0D          cmp al,0dh
036E 75 27          jnz pa
0370 EB 039D R      call p_numch
0373 47             inc di
0374 8A 05          mov al,[di].
0376 3C 0A          cmp al,0ah
0378 75 1D          jnz pa
037A EB 039D R      call p_numch
;__check address print=address data__
037D 3B FE          cmp di,si
037F 75 04          jnz p8
0381 B4 00          mov ah,0h
0383 59             pop cx
0384 C3             ret
;__check max__
p8:
0385 83 FF FF      cmp di,madd
0388 75 0C          jnz p0
038A EB 03E2 R      call check_seg
038D EB 0449 R      call keep_seg
0390 BF 0000        mov di,0h
0393 EB 02 90      jmp pa
p0:
0396 47             inc di
pa:
0397 59             pop cx
0398 E2 97          loop check_i
039A B4 01          mov ah,1h
039C C3             ret
p_numch:
039D 50             push ax
039E 51             push cx
039F 52             push dx
03A0 57             push di
03A1 8A 05          mov al,[di]
03A3 EE             out dx,al ;send char.
03A4 42             inc dx
busy:
03A5 EC             in al,dx
03A6 AB 00        test al,00h ;test printer ready
03A8 75 02          jnz send
03AA EB F9        jmp busy
send:
;send status control printer
03AC 42             inc dx
03AD B0 0D          mov al,0dh
03AF EE             out dx,al
03B0 B0 0C          mov al,0ch
03B2 EE             out dx,al
03B3 5F             pop di
03B4 5A             pop dx
03B5 59             pop cx
03B6 5B             pop ax
03B7 C3             ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; Process mov segment data  
; To  
; Memory addr. 0f000-0f001h  
;-----
```

```
03B8 keep_segd:  
03B8 1E push ds  
03B9 53 push bx  
03BA 50 push ax  
03BB 0C D8 mov ax,ds  
03BD 0B 0000 mov bx,0h  
03C0 0E D8 mov ds,bx  
03C2 0B F000 mov bx,seg0  
03C5 0B 27 mov [bx],ah  
03C7 43 inc bx  
03C8 0B 07 mov [bx],al  
03CA 58 pop ax  
03CB 58 pop bx  
03CC 1F pop ds  
03CD C3 ret
```

```
;-----  
; Process mov segment data  
; From  
; Memory addr. 0f000-0f001h  
;-----
```

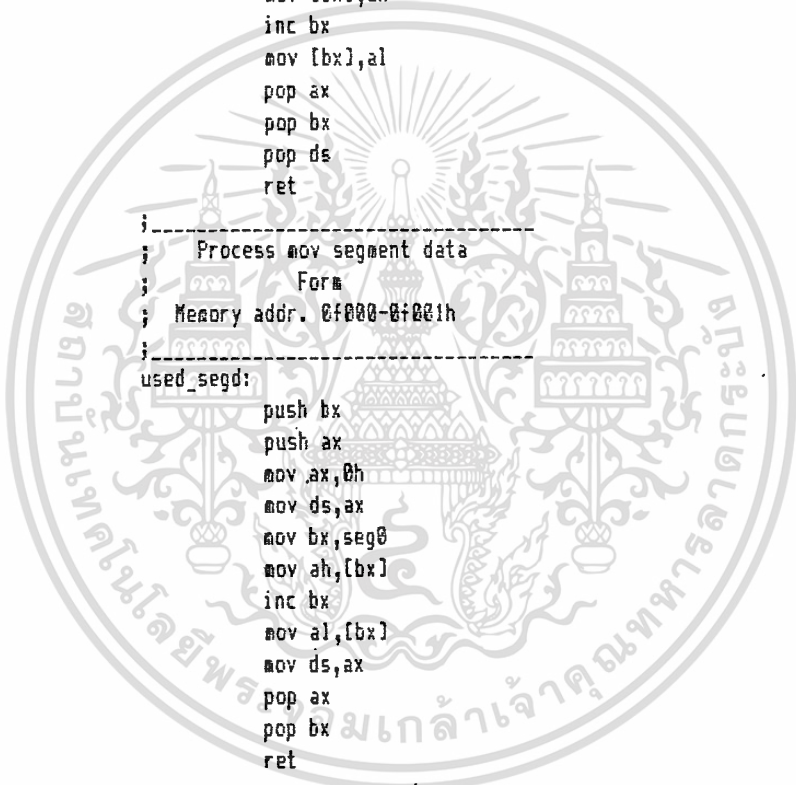
```
03CE used_segd:  
03CE 53 push bx  
03CF 50 push ax  
03D0 0B 0000 mov ax,0h  
03D3 0E D8 mov ds,ax  
03D5 0B F000 mov bx,seg0  
03D8 0A 27 mov ah,[bx]  
03DA 43 inc bx  
03DB 0A 07 mov al,[bx]  
03DD 0E D8 mov ds,ax  
03DF 58 pop ax  
03E0 58 pop bx  
03E1 C3 ret
```

```
;-----  
; Process check last segment  
; And  
; Set first segment  
;-----
```

```
03E2 check_seg:  
03E2 53 push bx  
03E3 50 push ax  
03E4 0C D8 mov bx,ds  
03E6 0B 3000 mov ax,1seg  
03E9 3B D8 cmp bx,ax  
03EB 75 08 jnz 1s  
03ED 0B 1000 mov ax,fseg  
03F0 0E D8 mov ds,ax  
03F2 EB 07 90 jmp fs
```

1s:

```
03F5 01 C3 1000 add bx,01000h  
03F9 0E D8 mov ds,bx
```



สารที่สงวนไว้สำหรับการใช้ภายในศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
03FB          fs:
03FB 5B                pop ax
03FC 5B                pop bx
03FD C3                ret
;-----
;      Process check seg.data
;      And
;      seg.print equal
;-----
```

```
03FE          check_dp:
03FE 1E                push ds
03FF 52                push dx
0400 53                push bx
0401 50                push ax
0402 BB 0000          mov ax,0h
0405 8E DB          mov ds,ax
0407 BB F000          mov bx,seg0
040A 8A 27          mov ah,[bx]
040C 43                inc bx
040D 8A 07          mov al,[bx]
040F 43                inc bx
0410 8A 2F          mov ch,[bx]
0412 43                inc bx
0413 8A 0F          mov cl,[bx]
0415 3B C1          cmp ax,cx
0417 75 06          jnz mi
0419 B9 AAAA          mov cx,0aaaah
041C EB 04 90        jmp ti
041F          mi:
041F B9 BBBB          mov cx,0bbbbh
0422          ti:
0422 5B                pop ax
0423 5B                pop bx
0424 5A                pop dx
0425 1F                pop ds
0426 C3                ret
```

```
;-----
;
;      Process ckeck last segment
;-----
```

```
0427          check_ep:
0427 1E                push ds
0428 53                push bx
0429 50                push ax
042A BB 0000          mov ax,0h
042D 8E DB          mov ds,ax
042F BB F000          mov bx,seg0
0432 8A 27          mov ah,[bx]
0434 43                inc bx
0435 8A 07          mov al,[bx]
0437 3D 3000          cmp ax,lseg
043A 75 06          jnz nil
043C B9 00AA          mov cx,00aah
043F EB 04 90        jmp ti
0442          nil:
```

```
0442 B9 AA00      mov cx,0aa00h
0445             til:
0445 5B            pop ax
0446 5B            pop bx
0447 1F            pop ds
0448 C3            ret
;-----
; Process mov seg.print
; To
; Memory addr. 0f002-0f003h
;-----
```

```
0449             keep_seg:
0449 1E            push ds
044A 53            push bx
044B 50            push ax
044C 8C D8        mov ax,ds
044E BB 0000      mov bx,0h
0451 8E DB        mov ds,bx
0453 BB F002      mov bx,seg0_1
0456 88 27        mov [bx],ah
0458 43            inc bx
0459 88 07        mov [bx],al
045B 58            pop ax
045C 5B            pop bx
045D 1F            pop ds
045E C3            ret
;-----
; Process mov seg.print
; Form
; Memory addr. 0f002-0f003h
;-----
```

```
045F             used_seg:
045F 53            push bx
0460 50            push ax
0461 BB 0000      mov ax,0h
0464 8E DB        mov ds,ax
0466 BB F002      mov bx,seg0_1
0469 8A 27        mov ah,[bx]
046B 43            inc bx
046C 8A 07        mov al,[bx]
046E 8E DB        mov ds,ax
0470 58            pop ax
0471 5B            pop bx
0472 C3            ret
;-----
; Process check key.sw
; And
; Convert port
;-----
```

```
0473             key_fun:
0473 1E            push ds
0474 51            push cx
0475 53            push bx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0476 50          push ax
0477          fine:
0477 B9 000A      mov cx,0ah
047A          find:
047A 51          push cx
047B B9 FFFF      mov cx,0ffffh ;time delay for key.sw
047E          find1:
047E BA 0309      mov dx,portb
0481 EC          in al,dx
0482 24 0F        and al,0fh
0484 3C 00        cmp al,0h
0486 75 0B        jnz find3
0488 E2 F4        loop find1
048A 59          pop cx
048B E2 ED        loop find
048D          findj:
048D B9 0378      mov cx,portp1
0490 EB 1F 90     jmp kp
0493          find3:
;check bit key.sw
0493 59          pop cx
0494 AB 01        test al,01h
0496 75 0A        jnz k1
0498 AB 02        test al,02h
049A 75 0C        jnz k2
049C AB 04        test al,04h
049E 75 0E        jnz k3
04A0 EB EB      jmp findj
k1:
;convert bit to port printer
04A2 B9 0378      mov cx,portp1
04A5 EB 0A 90     jmp kp
k2:
04A8 B9 038C      mov cx,portp2
04AB EB 04 90     jmp kp
k3:
04AE B9 0033      mov cx,portp3
04B1          kp:
;mov port printer to memory
04B1 BB 0000      mov bx,0h
04B4 BE DB      mov ds,bx
04B6 -BB F004    mov bx,keep_portp
04B9 8B 2F      mov [bx],ch
04BB 43          inc bx
04BC 8B 0F      mov [bx],cl
04BE 58          pop ax
04BF 5B          pop bx
04C0 59          pop cx
04C1 1F          pop ds
04C2 C3          ret
;
;-----
; Process mov no.port printer
; For#
; Memory addr. 0f004-0f005h
;-----
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
04C3          used_portp:
04C3 1E          push ds
04C4 51          push cx
04C5 53          push bx
04C6 50          push ax
04C7 8B 0000     mov ax,0h
04CA 8E D8      mov ds,ax
04CC 8B F004     mov bx,keep_portp
04CF 8A 37       mov dh,[bx]
04D1 43          inc bx
04D2 8A 17       mov dl,[bx]
04D4 58          pop ax
04D5 5B          pop bx
04D6 59          pop cx
04D7 1F          pop ds
04D8 C3          ret
```

```
;  
;-----  
; Process check no.port data  
; And  
; Piority order  
;-----  
;
```

```
04D9          check_port:
04D9 50          push ax
04DA 53          push bx
04DB 51          push cx
04DC 52          push dx
04DD 1E          push ds
04DE B3 00      mov bl,000h
04E0 B9 0000     mov cx,00h
04E3          tes_1:
04E3 8A D0       mov dl,al
04E5 8A FB       mov bh,bl
04E7 22 FA       and bh,dl
04E9 2A FB       sub bh,bl
04EB 80 FF 00    cmp bh,0h
04EE 74 04      jz keepbit
04F0 D0 CB      ror bl,01h
04F2 E2 EF      loop tes_1
```

```
keepbit:
04F4 80 FB 00     cmp bl,00h
04F7 74 24      jz getp8
04F9 80 FB 40    cmp bl,40h
04FC 74 28      jz getp7
04FE 80 FB 20    cmp bl,20h
0501 74 2C      jz getp6
0503 80 FB 10    cmp bl,10h
0506 74 30      jz getp5
0508 80 FB 08    cmp bl,08h
050B 74 34      jz getp4
050D 80 FB 04    cmp bl,04
0510 74 38      jz getp3
0512 80 FB 02    cmp bl,02
0515 74 3C      jz getp2
0517 BA 0300     mov dx,0300h
051A EB 56 90     jmp port_data
```

เอกสารที่สงวนไว้สำหรับการใช้งานในสถานศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
051D          getp8:
051D EB 0565 R      call kbit
0520 BA 0307      mov dx,0307h
0523 EB 4D 90      jmp port_data
0526          getp7:
0526 EB 0565 R      call kbit
0529 BA 0306      mov dx,0306h
052C EB 44 90      jmp port_data
052F          getp6:
052F EB 0565 R      call kbit
0532 BA 0305      mov dx,0305h
0535 EB 3B 90      jmp port_data
0538          getp5:
0538 EB 0565 R      call kbit
053B BA 0304      mov dx,0304h
053E EB 32 90      jmp port_data
0541          getp4:
0541 EB 0565 R      call kbit
0544 BA 0303      mov dx,0303h
0547 EB 29 90      jmp port_data
054A          getp3:
054A EB 0565 R      call kbit
054D BA 0302      mov dx,0302h
0550 EB 20 90      jmp port_data
0553          getp2:
0553 EB 0565 R      call kbit
0556 BA 0301      mov dx,0301h
0559 EB 17 90      jmp port_data
055C          getp1:
055C EB 0565 R      call kbit
055F BA 0300      mov dx,0300h
0562 EB 0E 90      jmp port_data
0565          kbit:
;mov busy bit to memory addr. 0f00ah
0565 BB 0000      mov ax,0h
0568 BE 08      mov ds,ax
056A BA D3      mov dl,bl
056C BB F00A      mov bx,addbit
056F B8 17      mov [bx],dl
0571 C3      ret
0572          port_data:
;mov no.port data to memory addr. 0f008-0f009h
0572 B8 0000      mov ax,0h
0575 BE 08      mov ds,ax
0577 BB F008      mov bx,add_keepbit
057A B8 37      mov [bx],dh
057C 43      inc bx
057D B8 17      mov [bx],dl
057F 1F      pop ds
0580 5A      pop dx
0581 59      pop cx
0582 5B      pop bx
0583 58      pop ax
0584 C3      ret
0585          pdata_1:
;mov no.port data from memory addr. 0f008-0f009h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อ 51 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0585 50      push ax
0586 53      push bx
0587 1E      push ds
0588 B8 0000  mov ax,0h
058B 8E D8    mov ds,ax
058D BB F008  mov bx,add_keepbit
0590 9A 2F    mov ch,[bx]
0592 43      inc bx
0593 8A 0F    mov cl,[bx]
0595 1F      pop ds
0596 5B      pop bx
0597 58      pop ax
0598 8B D1    mov dx,cx
059A C3      ret
059B      pro ends
end
```

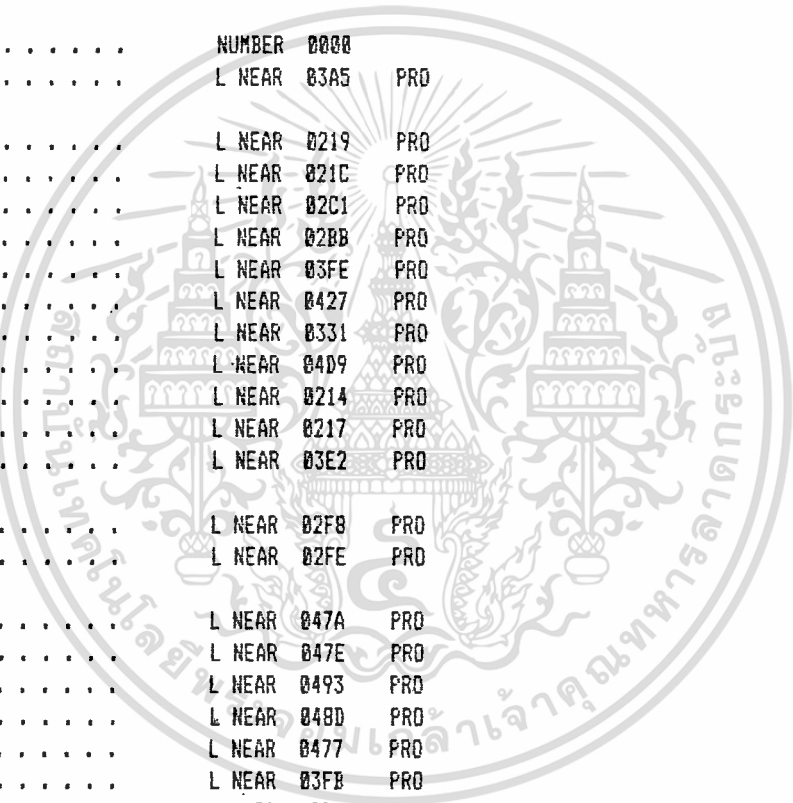


Segments and Groups:

Name	Length	Align	Combine	Class
PRO	059B	PARA	NONE	

Symbols:

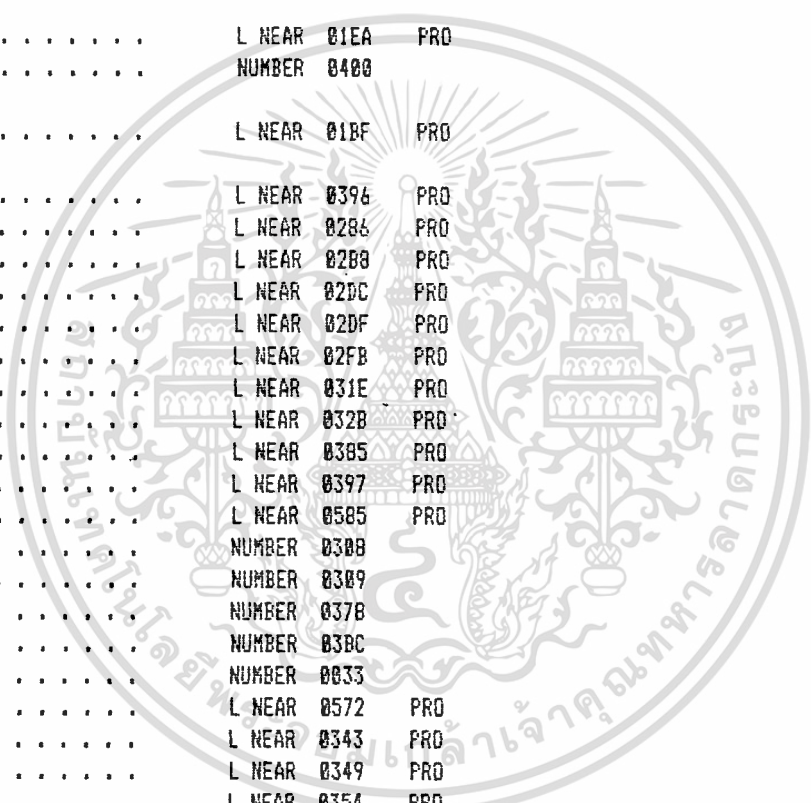
Name	Type	Value	Attr
AB	L NEAR	0230	PRO
AC	L NEAR	0234	PRO
ADDBIT	NUMBER	F00A	
ADD_KEEPBIT	NUMBER	F008	
BITSTROBE	NUMBER	0000	
BUSY	L NEAR	03A5	PRO
CHECK_0	L NEAR	0219	PRO
CHECK_1	L NEAR	021C	PRO
CHECK_2	L NEAR	02C1	PRO
CHECK_3	L NEAR	02BB	PRO
CHECK_DP	L NEAR	03FE	PRO
CHECK_EP	L NEAR	0427	PRO
CHECK_I	L NEAR	0331	PRO
CHECK_PORT	L NEAR	04D9	PRO
CHECK_R	L NEAR	0214	PRO
CHECK_S	L NEAR	0217	PRO
CHECK_SEG	L NEAR	03E2	PRO
EB	L NEAR	02F8	PRO
ENDF	L NEAR	02FE	PRO
FIND	L NEAR	047A	PRO
FIND1	L NEAR	047E	PRO
FIND3	L NEAR	0493	PRO
*FINDJ	L NEAR	048D	PRO
FINE	L NEAR	0477	PRO
FS	L NEAR	03FB	PRO
FSE6	NUMBER	1000	
GETP1	L NEAR	055C	PRO
GETP2	L NEAR	0553	PRO
GETP3	L NEAR	054A	PRO
GETP4	L NEAR	0541	PRO
GETP5	L NEAR	0538	PRO
GETP6	L NEAR	052F	PRO
GETP7	L NEAR	0526	PRO
GETP8	L NEAR	051D	PRO
K1	L NEAR	04A2	PRO
K2	L NEAR	04A8	PRO
K3	L NEAR	04AE	PRO
KBIT	L NEAR	0565	PRO
KEEPBIT	L NEAR	04FA	PRO
KEEP_PORTP	NUMBER	F00A	



นี่เป็นเอกสารที่สงวนไว้สำหรับบุคลากรที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEEP_SEG0	L NEAR 0300	PRO
KEEP_SEG1	L NEAR 0449	PRO
KEY_FUN	L NEAR 0473	PRO
KP	L NEAR 04B1	PRO
LS	L NEAR 03F5	PRO
LSEG	NUMBER 3000	
MADD	NUMBER FFFF	
MESSEGE	L BYTE 0103	PRO
MI	L NEAR 041F	PRO
MI1	L NEAR 0442	PRO
MREC	NUMBER FFFE	
NUK1	L NEAR 01EA	PRO
NUMBER_CHA	NUMBER 0400	
OVER	L NEAR 01BF	PRO
P0	L NEAR 0396	PRO
P1	L NEAR 0286	PRO
P2	L NEAR 0200	PRO
P3	L NEAR 02DC	PRO
P4	L NEAR 02DF	PRO
P5	L NEAR 02FB	PRO
P6	L NEAR 031E	PRO
P7	L NEAR 032B	PRO
P8	L NEAR 0385	PRO
PA	L NEAR 0397	PRO
PDATA_1	L NEAR 0585	PRO
PORTA	NUMBER 0300	
PORTB	NUMBER 0309	
PORTP1	NUMBER 0378	
PORTP2	NUMBER 03BC	
PORTP3	NUMBER 0033	
PORT_DATA	L NEAR 0572	PRO
PP	L NEAR 0343	PRO
PP1	L NEAR 0349	PRO
PP3	L NEAR 0354	PRO
PRINT	L NEAR 032E	PRO
PX	L NEAR 0313	PRO
P_NUMCH	L NEAR 039D	PRO
SE00	NUMBER F000	
SE00_1	NUMBER F002	
SEND	L NEAR 03AC	PRO
SPOOLER	L NEAR 01E7	PRO
TEST_0	L NEAR 0209	PRO
TEST_1	L NEAR 020C	PRO
TES_1	L NEAR 04E3	PRO
TI	L NEAR 0422	PRO
TI1	L NEAR 0445	PRO
TIMEBIT	NUMBER 2000	
TIME_1	NUMBER 001D	
TIME_2	NUMBER 0300	



```
USED_PORTP . . . . . L NEAR 04C3 PRO
USED_SEGD . . . . . L NEAR 03CE PRO
USED_SEGP . . . . . L NEAR 045F PRO

VV . . . . . L NEAR 0235 .PRO

XE . . . . . L NEAR 01F8 PRO

YJB . . . . . L NEAR 035A PRO
YX . . . . . L NEAR 0242 PRO
YX0 . . . . . L NEAR 024B PRO
YY . . . . . L NEAR 026F PRO
YY1 . . . . . L NEAR 026A PRO

ZZ . . . . . L NEAR 0223 PRO

@FILENAME . . . . . TEXT BF
_NEXT . . . . . L NEAR 01DC PRO
```

720 Source Lines
720 Total Lines
104 Symbols

51206 + 35466 Bytes symbol space free

0 Warning Errors
0 Severe Errors



บทที่ 4

การทดลองและผลการทดลอง

โครงการนี้ได้แบ่งการทดลองออกเป็น 3 ส่วน คือ ทางด้านวงจร , ทางด้านโปรแกรม, และการทดลองการใช้งานเครื่อง CENTRONIC PRINTER BUFFER ดังนี้

4.1 การสร้างและการทดลองในส่วนทางด้านวงจร (HARDWARE)

ตามที่กล่าวมาแล้วในหัวข้อ 3.1, 3.2 และ 3.3 เป็นส่วนที่ได้มีการทดลองและพัฒนาตามขั้นตอนดังนี้

1.ขั้นแรกได้ทดลองโดยสร้างส่วน อินพุตพอร์ต เพียงช่องเดียวก่อน โดยการต่ออุปกรณ์ทั้งหมดลงบน WIREWRAP CARD โดยการใช้ IC#8255 เป็นตัวรีเซ็ต ดี - ฟลิปฟลอป (IC# 74LS74) ในการทำแฮนด์เชคระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ ในการทดลองส่วนนี้ได้ผลการทดลองเป็นที่น่าพอใจ ได้การทำงานที่ถูกต้อง แต่มีข้อเสียที่การทำงานของ IC# 8255 จะช้า และจะต้องใช้ถึง 8 ตัว ในการจะสร้างอินพุตทั้ง 8 ช่อง

2.การทดลองตามข้อ 1. ข้อมูลที่เราส่งเข้าทางอินพุตพอร์ตเราจะใช้ LOGIC PULSER เป็นตัวส่งสัญญาณ STROBE แทนสัญญาณ STROBE จากคอมพิวเตอร์ ส่วนข้อมูลจะใช้การเลือกด้วย DIP SWITCH

3.ส่วนเอาต์พุตพอร์ต ในการทดลองได้ใช้ PRINTER INTERFACE CARD ของ IBM PC ซึ่งจะใช้เป็น PRINTER OUTPUT PORT ทำหน้าที่ส่งข้อมูลออก

4.ส่วนมาสเตอร์บอร์ดในการทดลองได้ใช้เมนบอร์ดของเครื่องคอมพิวเตอร์ IBM PC ซึ่งจะใช้ไมโครโปรเซสเซอร์ 8088 และหน่วยความจำเป็นส่วนบัฟเฟอร์จากบอร์ดนี้ ผลจากการทดลอง จะทำให้การทำงานของวงจรอินพุตพอร์ตที่สร้างขึ้นทำงานได้ถูกต้องและมีประสิทธิภาพสูง และเป็นการประหยัดเวลาและค่าใช้จ่าย เพราะประสิทธิภาพดีกว่าการสร้างขึ้นเอง

5.ได้ทดลองโดยสร้างส่วนอินพุตพอร์ตขึ้น ตามวงจรรูปที่ 9 ซึ่งได้แก้ไขในส่วนการทำ HANDSHAKE โดยการใช้สัญญาณ RESET DRIVER จากระบบเข้ามาเคลียร์ ดี - ฟลิปฟลอป แทนการใช้ IC#8255 ซึ่งให้การทำงานที่รวดเร็วกว่า ง่ายในการทำงาน และประหยัดกว่า

6.ได้ทดลองสร้างส่วน PRINTER SELECT ขึ้นเพื่อใช้เลือกเครื่องพิมพ์ที่จะใช้พิมพ์ ตามรูปที่ 13 ซึ่งผลการทดลองตามวงจร ทำงานถูกต้อง สามารถเลือกได้ 3 แบบ คือ เลือกออกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องพิมพ์ตัวที่ 1 (LPT1) , เลือกออกเครื่องพิมพ์ตัวที่ 2 (LPT2) , และเลือกออกพิมพ์ 2 ตัวพร้อมกัน (LPT1&LPT2) ส่วนปุ่ม PAUSE ตามวงจรไม่ได้นำมาใช้ แต่สามารถต่อหรือพัฒนาได้ง่าย

7. ทดลองส่วน DISPLAY ตามรูปที่ 14 ที่ใช้ LED เป็นตัวแสดงผล เพื่อใช้บอกสถานะข้อมูลจากคอมพิวเตอร์ที่ส่งออกพิมพ์ โดยใช้ LED 8 ตัว เพื่อบอกสถานะคอมพิวเตอร์ 8 เครื่อง การทดลองและการใช้งานจะใช้สัญญาณ BUSY จากส่วนอินพุตพอร์ตเข้ามาแสดง โดยผลการทดลองจะได้ว่า

7.1 ในสภาวะเริ่มต้นที่คอมพิวเตอร์ตัวแรกส่งข้อมูลมาแล้วส่งออกพิมพ์ LED จะแสดงผลโดยจะสว่างแบบกะพริบถี่ ๆ ส่วนถ้า

7.2 มีคอมพิวเตอร์อีกตัวส่งข้อมูลเข้ามา ในขณะที่ตัวแรกกำลังส่งข้อมูลออกพิมพ์อยู่ LED ของตัวนี้จะสว่างจ้าตลอด จนกระทั่งข้อมูลจากตัวแรกหมด ข้อมูลจากตัวที่สองจึงจะส่งและที่ตัว LED แสดงผลจะสว่างแบบกะพริบถี่ ๆ

ในส่วน PRINTER SELECT และ DISPLAY ได้ทดลองและสร้างบนแผ่นวงจรเดียวกัน

4.2 การเขียนและการทดลองในส่วนโปรแกรม (SOFTWARE)

1. ทำการเขียนโปรแกรมทดสอบการส่งข้อมูลออกเครื่องพิมพ์ 1 ตัว โดยใช้หลักการตามทฤษฎี คือ ในการส่งข้อมูลออกเครื่องพิมพ์ จะต้องทำการส่งสัญญาณ STROBE ไปให้กับเครื่องพิมพ์ก่อน ซึ่งในโปรแกรมก็ได้ทำการส่งรหัส OD และ OC ออกไปที่พอร์ตเครื่องพิมพ์ ซึ่งต้องส่งในพอร์ตแสดงสถานะของเครื่องพิมพ์

ปัญหาที่เกิดขึ้น ก็คือ ในการทดลองเราใช้ PRINTER CARD ของเครื่องคอมพิวเตอร์ที่เป็น OUTPUT PORT PRINTER ให้กับบัพเฟอร์ของเรา ซึ่งการเชื่อมต่อเครื่องพิมพ์ จึงขึ้นอยู่กับ PRINTER CARD นั้น ๆ ดังนั้น หากทำการทดลองกับการ์ดที่เชื่อมต่อเครื่องพิมพ์ (PORT PRINTER LPT1&LPT2) ไม่ตรงกับพอร์ตเครื่องพิมพ์ที่เราเชื่อมต่อไว้ในโปรแกรม จะทำให้โปรแกรมไม่สามารถทำงานถูกต้องได้

2. หลังจากทำการทดลองเขียนโปรแกรมส่งข้อมูลออกเครื่องพิมพ์ 1 ตัวได้แล้ว ก็ทำการส่งข้อมูลออกเครื่องพิมพ์ 2 ตัว โดยทำการสลับพอร์ตเครื่องพิมพ์ ซึ่งในที่นี้เราได้ใช้ MULTI I/O CARD เป็น O/P PORT PRINTER ซึ่งจากการทดลองต้องมีความระมัดระวัง ในการเชื่อมต่อ

ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องพิมพ์ บน MULTI I/O CARD คือ บน MULTI I/O CARD จะสามารถเซ็ทพอร์ตเครื่องพิมพ์ ให้เป็น LPT1 (NO.PORT 378H) และ LPT2 (NO.278H) ดังนั้น เมื่อทำการทดลองด้วย MULTI I/O CARD จะต้องเซ็ทพอร์ตเครื่องพิมพ์บน MULTI I/O CARD ไม่ให้ตรงกับพอร์ตเครื่องพิมพ์เดิมที่ใช้อยู่ (อาจเป็น MONOCHROME CARD)

3. ทดลองในส่วนโปรแกรมรับข้อมูล ในขั้นแรกก็ทำการเขียนโปรแกรม ให้สามารถรับข้อมูลได้ โดยใช้ตัวส่ง STROBE เป็น LOGIC PULSER และใช้ DIP SWITCH เป็นข้อมูล ซึ่งเมื่อรับส่งข้อมูลมาไว้ในบัฟเฟอร์แล้วก็สามารถดูข้อมูลในหน่วยความจำได้โดยใช้ DEBUG DUMP คู่มือที่ตำแหน่งที่เรากำหนดใน โปรแกรม

จากนั้นก็ทำการเขียนโปรแกรมการรับข้อมูลในลักษณะคิว โดยเขียนโปรแกรมให้สามารถรับข้อมูลได้เป็นคิว โดยรับข้อมูลจำนวนน้อย ๆ (20 BYTES) จากนั้นจึงเพิ่มจำนวนการรับข้อมูลให้มากขึ้นเป็น 1 เซกเมนต์ (64 KBYTES)

ปัญหาที่เกิดขึ้นในส่วนนี้ คือ เมื่อเราทำการทดลองบนเครื่อง IBM PC ดังนั้นข้อมูลที่รับเข้ามาในกรณีที่มีจำนวนน้อย ๆ ไม่กี่ไบต์ ก็จะไม่เกิดปัญหาอะไรขึ้น แต่เมื่อทำการรับข้อมูลมากขึ้น (64 KBYTES) ปรากฏว่าเครื่องคอมพิวเตอร์ เกิดหยุดทำงาน และบอกว่า DIVIDED OVER FLOW และเมื่อทดลองครั้งที่ 2 ก็จะเป็นเช่นเดิมอีก ปัญหานี้เกิดขึ้นเพราะว่าข้อมูลที่เรารับเข้ามาเป็นจำนวนมากนั้นไปทับกับส่วนของข้อมูลในหน่วยความจำ หรืออาจเป็นส่วนของ BIOS จึงทำให้เครื่องคอมพิวเตอร์เกิดการ ERROR ขึ้น

เราทำการแก้ไขปัญหาที่เกิดขึ้นได้โดยการ เปลี่ยนตำแหน่งของบัฟเฟอร์ในโปรแกรม ให้อยู่ในตำแหน่งที่ว่าง ที่ผู้ใช้สามารถใช้ได้ จากนั้นก็ได้ทำการเขียนโปรแกรมเพื่อรับข้อมูลได้เพิ่มหลายเซกเมนต์

4. การทดลองโปรแกรมทั้งหมด โดยใช้เครื่องคอมพิวเตอร์เป็นตัวส่งข้อมูลแทน LOGIC PULSER ที่ใช้ตอนแรก ปัญหาที่เกิดขึ้น ก็คือ ข้อมูลที่ทำการส่งออกเครื่องพิมพ์ โดยผ่านบัฟเฟอร์เกิดการผิดพลาด เนื่องจากโปรแกรมมีการตรวจสอบสถานะผิด ทั้งนี้ในตอนต้นนั้นเราได้ทำการทดลอง โดยใช้ LOGIC PULSER เป็นตัวส่งข้อมูล ซึ่งการส่งข้อมูลโดยใช้ LOGIC PULSER นี้ จะส่งข้อมูลที่มีลักษณะซ้ำกันหลาย ๆ ตัว (ไม่สามารถเปลี่ยนแปลงข้อมูลได้ ในเวลาสั้น ๆ) ดังนั้น เมื่อแสดงผลทางกระดาษ เราจึงไม่สามารถทราบได้ว่าข้อมูลที่ส่งมานั้น กับข้อมูลที่รับเข้ามาถูกต้องเหมือนกันหมด แต่เมื่อทดลองส่งข้อมูลจากเครื่องคอมพิวเตอร์ (โดยใช้ RAJAVITHI เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WORD) จึงทำให้เห็นข้อผิดพลาดได้ชัดเจน

ในส่วนของการทดลองนี้ ยังพบปัญหาจากการส่งข้อมูลจากเครื่องคอมพิวเตอร์ ในการส่งข้อมูลแต่ละครั้ง คอมพิวเตอร์จะต้องใช้เวลาในการประมวลผลข้อมูล ดังนั้นเราจะต้องคำนึงถึงเวลาในการประมวลผลข้อมูลของคอมพิวเตอร์ก่อนที่จะทำการส่งข้อมูลมาด้วย ในการทดลองส่วนนี้ ครั้งแรกไม่มีการหน่วงเวลา ในการรับข้อมูลจากเครื่องคอมพิวเตอร์ จึงทำให้ข้อมูลที่ทำการรับเข้ามานั้นผิดพลาด ดังนั้น จึงทำส่วนการหน่วงเวลา ในการรับข้อมูล เพื่อให้ได้ข้อมูลที่ถูกต้อง

ในกรณีที่เรทำการส่งข้อมูล โดยใช้ WORD PROCESSOR ต่าง ๆ ข้อควรระวัง ก็คือ เราต้องเช็ครหัส ของข้อมูลที่จะส่งให้ตรงกับเครื่องพิมพ์ที่ใช้ หรือ WORD PROCESSOR บางตัวก็จะสามารถเช็คชนิดเครื่องพิมพ์ได้ด้วย เพราะถ้าหากต่างรหัสกันแล้วข้อมูลที่ส่งออกเครื่องพิมพ์ก็จะผิดพลาดด้วย

ปัญหาที่เกิดขึ้น ในการรับข้อมูลจากเครื่องคอมพิวเตอร์ ที่สำคัญ คือ ในการส่งข้อมูลจากเครื่องคอมพิวเตอร์ที่รหัสพิเศษ (ไม่ใช้ MONOCHROM CARD) จะทำให้ข้อมูลที่รับได้ผิดพลาด ทั้งนี้เกิดขึ้นเนื่องจาก บนการ์ดนั้นอาจจะมีการเบิร์คสภาวะต่าง ๆ ของเครื่องพิมพ์เป็นกรณีพิเศษ แล้วทำการส่งรหัสควบคุมพิเศษมา ดังนั้นข้อมูลที่รับได้จึงผิดพลาด

5. ส่วนโปรแกรมการรับคีย์สวิตช์ ในส่วนนี้เกิดปัญหาในการอ่านสถานะของคีย์สวิตช์ ซึ่งเป็นปัญหาที่เกิดจากฮาร์ดแวร์ คือ มีสัญญาณรบกวนเกิดขึ้น ทำให้เกิดสถานะที่คีย์สวิตช์ผิดพลาดขึ้น จึงทำให้การเลือกเครื่องพิมพ์ผิดพลาด ปัญหานี้เกิดจากสัญญาณไฟ ดี.ซี. ที่จ่ายมาเลี้ยงส่วน LED และ IC ในภาคสวิตช์เลือกเครื่องพิมพ์ไม่ดี แก้ปัญหาโดยการต่อคาปาซิเตอร์ ครอบระหว่างไฟบวกกับกราวด์ ที่มาเลี้ยงภาคสวิตช์เลือกเครื่องพิมพ์นี้

4.3 การทดลองการใช้งานเครื่อง CENTRONIC PRINTER BUFFER

1. ทดลองรับข้อมูลที่เขียนด้วย RAJAWITHI WORD ในแบบ DRAFF จะได้ว่าข้อมูลขนาด 1 หน้าซึ่งประมาณ 2,200 ตัวอักษร (นับสระ พยัญชนะ ช่องว่าง) จะใช้เวลาส่งประมาณ 3 วินาที เท่านั้น

2. ทดลองรับข้อมูลที่เขียนด้วย RAJRWITHI WORD ในแบบ NLQ ในขนาดข้อมูลเท่ากับข้อแรก จะใช้เวลาส่งประมาณ 45 วินาที สาเหตุที่ส่งช้ากว่าแบบ DRAFF เพราะในแบบนี้ตัวเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรเป็นกราฟิก จึงต้องแปลงรหัสก่อนส่ง

3. ทดลองส่ง 2 ตัวพร้อมกัน โดยใช้เครื่องคอมพิวเตอร์ 2 ตัวส่งข้อมูลขนาดเดียวกัน แต่ตัวหนึ่งส่งตัวอักษรแบบ DRAFF อีกตัวส่งแบบ NLQ จะได้ว่า แบบ DRAFF จะใช้เวลาส่งประมาณ 3 วินาที ส่วนอีกตัวจะใช้เวลาส่งประมาณ 45 วินาที

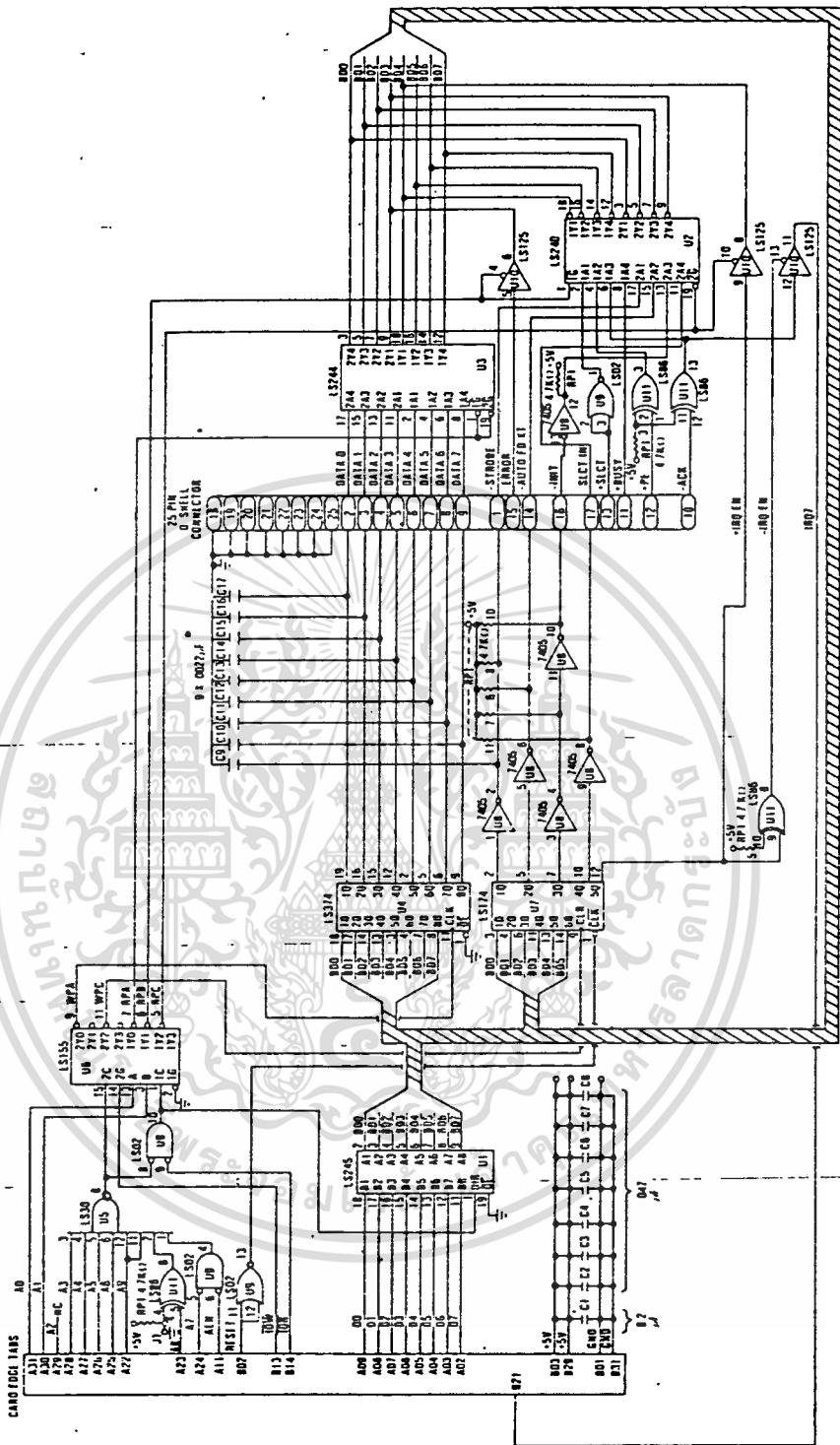
4. ทดลองส่งข้อมูลที่เป็นกราฟิก ส่งรูปขนาดของตัวภาพประมาณ 6.5 นิ้ว x 8.5 นิ้ว จะใช้เวลาส่งประมาณ 22 วินาทีเท่านั้น

จากผลการทดลองทั้ง 4 ข้อจะเห็นได้ว่า CENTRONIC PRINTER BUFFER จะช่วยให้เครื่องคอมพิวเตอร์สามารถส่งข้อมูลที่จะไปออกพิมพ์ที่เครื่องพิมพ์ได้เร็วมาก และเร็วกว่าในกรณีต่อโดยตรงระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ ซึ่งจากการทดลองต่อตรงนี้ จะได้ว่าต้องใช้เวลาในการส่งข้อมูลแบบ NLQ ถึงประมาณ 4 นาที , และใช้เวลาประมาณ 1 นาที 20 วินาทีในการส่งข้อมูลแบบ DRAFF (ข้อมูลแบบ NLQ และ DRAFF จะมีขนาดจำนวนตัวอักษรเท่ากันและเท่ากับที่ใช้ในการส่งผ่าน CENTRONIC PRINTER BUFFER) ส่วนกรณีส่งกราฟิกโดยการใช้อุปกรณ์เดียวกันกับในข้อที่ 4 จะใช้เวลาประมาณ 2 นาที 30 วินาที จากการเปรียบเทียบแล้วก็จะเห็นประโยชน์ของ CENTRONIC PRINTER BUFFER ซึ่งทำให้คอมพิวเตอร์สามารถไปทำงานอย่างอื่นต่อได้ เมื่อส่งข้อมูลหมดแล้ว

บทวิจารณ์ สรุป

ถึงแม้ว่าการทำงานของ CENTRONIC PRINTER BUFFER จะได้ตรงตามจุดประสงค์ที่วางไว้แล้วก็ตาม แต่เนื่องจากเวลาในการทำโครงการนี้มีจำกัด จึงทำให้ไม่สามารถพัฒนาในคุณสมบัติพิเศษอื่น ๆ ได้ทัน ดังนั้น เพื่อเป็นแนวทางแก่นักศึกษารุ่นน้องหรือผู้ที่สนใจ ในการใช้พัฒนาเครื่อง CENTRONIC PRINTER BUFFER นี้ต่อไปให้มีความสมบูรณ์ขึ้นไปอีก ผู้จัดทำก็ขอเสนอแนวทางพัฒนาที่น่าจะเป็นไปได้ ไว้เป็นข้อ ๆ ดังนี้

- 1.พัฒนาในการขยายหน่วยความจำ (BUFFER) ให้มากขึ้น
- 3.พัฒนาในส่วนของการรับข้อมูล โดยมีการแสดงผลตำแหน่งหน่วยความจำที่ว่างที่สามารถรับข้อมูลได้ ในแบบ DIGITAL DISPLAY
- 4.พัฒนาให้มีการส่งข้อมูลแบบอนุกรม เพื่อใช้ในการติดต่อสื่อสารระยะไกล ที่จะประหยัดเรื่องอุปกรณ์และสายมากกว่า
- 5.บรรจุโปรแกรมลงใน EPROM ในส่วนของ INT 18H (BASIC ROM) เพื่อที่จะทำงานได้โดยไม่ต้อง RUN บน DOS



วงจรอะแดปเตอร์คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B. Parallel Interface

The FX-80/FX-100 includes as standard equipment a parallel interface, which is described below.

(1) Specifications

- (a) Synchronization: - By externally supplied STROBE pulses.
- (b) Handshaking: By ACKNLG or BUSY signals.
- (c) Logic level: Input data and all interface control signals are compatible with the TTL level.

(2) Connector

Plug: 57-30360 (AMPHENOL)

It is recommended that interface cables be kept as short as possible.

(3) Connector pin assignment and descriptions of signals.

Connector pin assignment and descriptions of respective interface signals are provided in Table B-1.

Table B-1 Connector Pin Assignment and Descriptions of Interface Signals

Signal Pin No.	Return Pin No.	Signal	Direction	Description
1	19	<u>STROBE</u>	In	STROBE pulse to read data in. Pulse width must be more than 0.5 μ s at receiving terminal.
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data, respectively. Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0".
3	21	DATA 2	In	
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	<u>ACKNLG</u>	Out	Approx. 12 μ s pulse. "LOW" indicates that data has been received and that the printer is ready to accept other data.
11	29	<u>BUSY</u>	Out	A "HIGH" signal indicates that the printer cannot receive data. The signal becomes "HIGH" in the following cases: <ol style="list-style-type: none"> 1. During data entry 2. During printing operation 3. In OFF-LINE state 4. During printer error status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal Pin No.	Return Pin No.	Signal	Direction	Description
12	30	PE	Out	A "HIGH" signal indicates that the printer is out of paper.
13	—	—	—	Pulled up to +5V through 3.3 k Ω resistance
14	—	AUTO FEED XT	In	With this signal at "LOW" level, the paper is automatically fed one line after printing. (The signal level can be fixed to "LOW" with DIP SW pin 2-4 provided on the control circuit board.)
15	—	NC	—	Not used.
16	—	0V	—	Logic GND level.
17	—	CHASSIS GND	—	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	—	NC	—	Not used.
19 to 30	—	GND	—	TWISTED-PAIR RETURN signal GND level.
31	—	INIT	In	When the level of this signal becomes "LOW", the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "HIGH" level, and its pulse width must be more than 50 μ s at the receiving terminal.
32	—	ERROR	Out	The level of this signal becomes "LOW" when the printer is in— 1. PAPER END state 2. OFF-LINE state 3. Error state
33	—	GND	—	Same as with Pin Nos. 19 to 30.
34	—	NC	—	Not used.
35	—	—	—	Pulled up to +5V through 3.3 k Ω resistance.
36	—	SLCT IN	In	The DC1/DC3 code is only valid when this signal is "HIGH" level. (Internal fixing can be carried out with DIP SW pin 2-1. The level of this signal is factory-set to "LOW".)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES:

1. "Direction" refers to the direction of signal flow as viewed from the printer.
2. "Return" denotes "TWISTED PAIR RETURN" and is to be connected at signal ground level.

As to the wiring for the interface, be sure to use a twisted-pair cable for each signal and never fail to complete connection on the Return side. To prevent noise effectively, these cables should be shielded and connected to the chassis of the host computer and the printer, respectively.

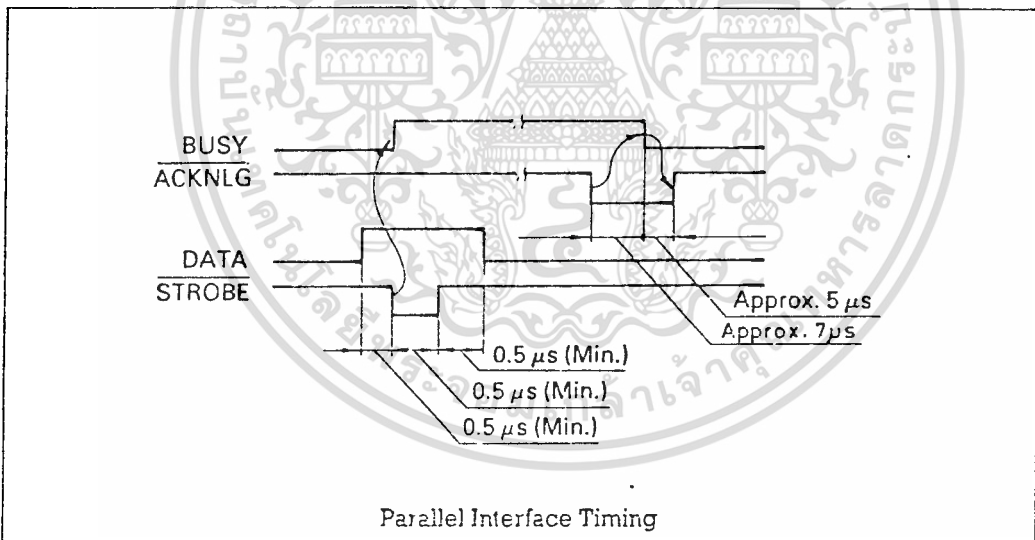
3. All interface conditions are based on TTL level. Both the rise and fall times of each signal must be less than $0.2 \mu s$.
4. Data transfer must not be carried out by ignoring the \overline{ACKNLG} or $BUSY$ signal. (Data transfer to this printer can be carried out only after confirming the \overline{ACKNLG} signal or when the level of the $BUSY$ signal is "LOW".)
5. Under normal conditions, printer cable pins 11, 12 and 32 are activated when out of paper is detected.

ESC 8 code disable pins 11 and 32 from the PE (Paper-End) signal, but it does not disable pin 12.

Although most computers do not monitor pin 12, those which do (e.g, TRS-80) will halt printing when the paper is out. Therefore, ESC 8 code is ineffective with these computers.

(4) Data transfer sequence

The figure below shows the sequence for data transmission.





กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ สำเร็จได้ด้วยความช่วยเหลือจาก รศ.กิตติ ตีระเศรษฐ์ ที่เป็นอาจารย์ที่ปรึกษา และได้ให้คำแนะนำที่เป็นประโยชน์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ของการทำวิจัยมาด้วยดีโดยตลอด อีกทั้งคณาจารย์ทุกท่าน แห่งภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่มีส่วนทำให้ปริญญาบัตรนี้สำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ข้าพเจ้าขอขอบคุณ คุณเกษม หลังสัน งานเอกสารการพิมพ์ วิทยาลัยเทคนิคมีนบุรี ที่ได้จัดพิมพ์เอกสารทั้งหมดให้เป็นระเบียบเรียบร้อย และที่ลืมไม่ได้คือ คุณกิตติ แซ่เอี้ยว นักศึกษาแผนกอิเล็กทรอนิกส์ วิทยาลัยเทคนิคมีนบุรี ที่ได้จัดทำแผ่น PRINTED CIRCUIT BOARD ตั้งแต่ ออกแบบจนประกอบวงจรสำเร็จ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

หนังสือภาษาไทย

1. ธาณินทร์ ถาวรศาสนวงศ์ , ทินกร คุ้ม , " การอินเทอร์เฟส IBM PC " , พลิทส์เซ็นเตอร์การพิมพ์ , 256 หน้า
2. ยืน ภู่วรวรรณ , ชัยยงค์ วงศ์ชัยสุวัฒน์ และ ไพศาล สงวนหมู่ , " เทคโนโลยีไมโครคอมพิวเตอร์ 16 บิต " , ซีเอ็ดดูเคชั่น , 342 หน้า , 2531
3. ศิริวรรณ ฉันทาคิตย์ , " หลักการเขียนโปรแกรมภาษาแอสเซมบลี 8088 " , ประกายพริก , 391 หน้า , 2531
4. สุรียัน ศรีสวัสดิ์กุล , " การใช้เครื่องพิมพ์ คอทแมทริก สำหรับไมโครคอมพิวเตอร์ " , สารมวลชน , 192 หน้า , 2530

หนังสือภาษาอังกฤษ

1. GORSLINE W. GEORGE , " 16-BIT MODERN MICROCOMPUTER THE INTEL I 8086 FAMILY " , PRENTICE - HALL INC. , 516 P.
2. IBM , " IBM TECHNICAL REFERENCE " , INTERNATIONAL BUSINESS MACHINES CORPORATION , 605 P. , 1983
3. JOWRDAIN ROBERT , " PROGRAMMER 'S PROBLEM SOLVER FOR THE IBM PC , XT AND AT " , PRENTICE - HALL PRESS , 473 P. , 1986
4. MURRAY SARGENT LLL AND RICHARD L. SHOEMALEER , " THE IBM PERSONAL COMPUTER FROM THE INSIDE OUT " ADDISON - WESTEY PUBLISHING COMPUTER , 483 P. , 1985

วารสารภาษาไทย

1. สุรียา สุนทรารุณ , " ปริ้นท์เตอร์ บัฟเฟอร์ " , วารสารคอมพิวเตอร์อิเล็กทรอนิกส์ เวลด์ , ปีที่ 11 ฉบับที่ 111 , 2530 , หน้า 14 - 19
2. สุรียา สุนทรารุณ , " คอมพิวเตอร์ อินเทอร์เฟส " , วารสารคอมพิวเตอร์ ฮาร์ดแวร์ ซอฟต์แวร์ คอมพิวเตอร์อิเล็กทรอนิกส์ , ฉบับที่ 1 , หน้า 19 - 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้