ปีการศึกษา 2531

การใช้คอมพิวเตอร์ในทางดนตรี กลุ่ม1

(COMPUTER MUSIC I)

โดย

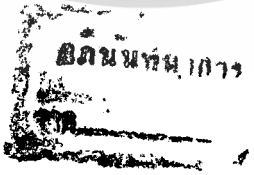
นางสาว เจตนา อัครวงษ์

นางสาว ประศมา ธานินทร์ปฐมรัฐ

นางสาว ปราณี จิงสถาปัตย์ชัย

อาจารย์ที่ปรึกษา

ผ.ศ. ครรชิต ไมตรี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023131 -8.ค.2532

ปริญญานิพนธ์ปีการศึกษา 2531

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การใช้คอมพิวเตอร์ในทางดนตรี กลุ่ม 1 (Computer Music 1)

ผู้จัดทำ

1. นางสาวเจตนา อัครวงษ์
2. นางสาวประศมา ธานินทร์ปฐมรัฐ
3. นางสาวปราณี จิงสถาปัตย์ชัย

.....อาจารย์ที่ปรึกษา

(ผศ. ครรชิต ไมตรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023131

การใช้คอมพิวเตอร์ในทางดนตรี กลุ่ม1

(COMPUTER MUSIC I)

โดย

นางสาว เจตนา อัครวงษ์

นางสาว ประศมา ธานินทร์ปฐมรัฐ

นางสาว ปราณี จิงสถาปัตย์ชัย

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ครรชิต ไมตรี

วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2531

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้คอมพิวเตอร์ในทางดนตรี กลุ่ม 1

นางสาวเจตนา อัครวงษ์
นางสาวประศมา ธานินทร์ปฐมรัฐ
นางสาวปราณี จิงสถาปัตย์ชัย
ผศ.ครรชิต ไมตรี อาจารย์ที่ปรึกษา
ปีการศึกษา 2531

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานที่ได้จากการศึกษาและทดลองสร้างวงจรตัวกลางมิดี้ (MIDI Interface) ซึ่งเป็นเครื่องมือที่ใช้เชื่อมต่อระหว่างระบบคอมพิวเตอร์กับอุปกรณ์ดนตรีที่สามารถเข้าใจมาตรฐานมีดี้ (Potocol) ให้สามารถติดต่อสื่อสารถึงกันได้ มาตรฐานมีดี้กำหนดให้การรับ - ส่งข้อมูลแบบอนุกรมอะซิงโครนัส (Serial Asynchronous) ด้วยความเร็ว 31250 บิท ต่อ วินาที ผ่านทางมีดี้พอร์ท (Port) ดังนั้น จึงใช้ Z-80 SIO เป็นตัวกลางในการรับส่งข้อมูลและมีแหล่งกำเนิดสัญญาณนาฬิกาขนาด 2 เมกกะเฮิร์ต

การใช้งานตัวกลางมีดี้จำเป็นต้องมีโปรแกรมควบคุมการทำงานตามแต่วัตถุประสงค์ ซึ่งตามโครงงานนี้ต้องการนำคอมพิวเตอร์มาช่วยเสริมการเรียนการสอนด้านดนตรี (Music Computer Aided Instruction) โดยเริ่มพัฒนาโปรแกรมสำหรับสอนเด็กหรือผู้ที่ไม่มีพื้นฐานทางด้านดนตรีมาก่อน ให้สามารถจดจำตัวโน้ตและสามารถเล่นเพลงง่าย ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer Music I

Jettana Akaravong

Prasama Thaninprathomrat

Pranee Jungsathapatchai

Professor Assistant Kanchit Maitri Advisor

1988

Abstract

This thesis arranges from the studying and invention of MIDI Interface circuit which is the circuit to interface between computer system and MIDI musical instruments. The MIDI protocol is serial asynchrous communication speed 31,250 bps. via MIDI port. So we use Z-80 SIO to control the data communication and use crystal oscillator 2 MHz. to generate clock signal.

In the use of MIDI Interface ,we must have a control program. In this project , we use a computer to teach music (Music Computer Aided Instruction). We start with a program to teach children and ones who do not have any knowledge about music in order to recognize music notes and can play the easy songs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการที่ใช้	6
2.1 ความรู้ทั่วไปเกี่ยวกับเครื่องดนตรีอิเลคทรอนิคส์	6
2.2 มีดี้กับเครื่องดนตรีอิเลคทรอนิคส์	6
2.3 ประโยชน์ของมีดี้	9
2.4 มาตรฐานมีดี้	11
บทที่ 3 การสร้างและใช้งานตัวกลางมีดี้	16
3.1 การเลือกอุปกรณ์ในการสร้างวงจรตัวกลางมีดี้	16
3.2 หลักการทำงานทั่วไปของ Z-80 SIO ที่ใช้ในวงจร	17
3.3 หลักการทำงานทั่วไปของ 8253 ที่ใช้ในวงจร	20
3.4 การกำหนดรายละเอียดต่าง ๆ ในวงจร	24
บทที่ 4 การสร้างและผลการทำงานของโปรแกรมเสริมการสอนด้านดนตรี	27
บทที่ 5 บทสรุปและวิจารณ์	33
5.1 สรุปผลการทดลองใช้งานวงจรตัวกลางมีดี้	33
5.2 สรุปผลการทดลองใช้งานโปรแกรมสอนดนตรี	33
5.3 ปัญหาที่พบและการแก้ไข	33
5.4 แนวทางในการพัฒนาต่อไป	34
ภาคผนวก	35
กิตติกรรมประกาศ	63
เอกสารอ้างอิง	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ปัจจุบันนี้ คอมพิวเตอร์ได้เข้ามามีบทบาทต่อธุรกิจการงานและชีวิตประจำวันอย่างสูง มีการนำมาประยุกต์ใช้ในด้านสันทนาการ งานอดิเรกต่างๆด้วย เช่น งานด้านศิลปะ งานด้านดนตรี และอื่นๆอีกมาก

สำหรับวงการดนตรีได้มีความก้าวหน้าและพัฒนาไปมาก เครื่องดนตรีไฟฟ้าสามารถผลิตเสียงแบบอิเล็คทรอนิกส์ เปลี่ยนสัญญานอนาลอกเป็นสัญญานดิจิตอลเพื่อให้คุณภาพเสียงที่ดีกว่าและลดขีดจำกัดของเสียงทางด้านอนาลอก จนกระทั่งได้มีการปรับปรุงเอาคอมพิวเตอร์มาใช้ควบคุม โดยยึดเอามาตรฐาน (Protocol) ที่เรียกว่า "มีดี้ (MIDI)" ซึ่งย่อมาจาก Musical Instrument Digital Interface กำหนดขึ้นจากความร่วมระหว่างประเทศญี่ปุ่น และ ประเทศสหรัฐอเมริกา ทำให้คอมพิวเตอร์สามารถควบคุมเครื่องดนตรีชนิดใดก็ได้ที่มีระบบมีดี้นี้ เช่น เครื่องสร้างเสียงกลอง (Drum Machine) และ เครื่องสังเคราะห์เสียงดนตรี (Synthesizer) เป็นต้น เครื่องดนตรีที่นำมาใช้ในโครงงานนี้คือ คาสิโอ รุ่น ซีที-640 (CASIO CT-640)

การที่จะใช้คอมพิวเตอร์ไปควบคุมเครื่องดนตรีที่มีระบบมีดี้นั้น จะต้องมีการเปลี่ยนแปลงระบบของข้อมูล จากคอมพิวเตอร์มาเป็นระบบมาตรฐานมีดี้เพื่อใช้ในการสื่อสารติดต่อกันระหว่างเครื่องมือทั้งสองได้ โดยอาศัยตัวกลางมีดี้ (MIDI Interface) ซึ่งจะส่งข้อมูลผ่านเข้า-ออกทาง พอร์ทมีดี้ (MIDI Port) ของเครื่องดนตรี

เมื่อสามารถเชื่อมโยง นักดนตรี เครื่องดนตรี และ คอมพิวเตอร์เข้าด้วยกันในระบบมีดี้ได้แล้ว กระบวนการพัฒนาทางด้านดนตรีก็ก้าวหน้าเติบโตต่อไปอย่างกว้างขวาง และ ได้เกิดมีซอฟท์แวร์หลายๆ ประเภทเข้ามามีบทบาท เช่น โปรแกรมจัดลำดับตัวโน้ต (Sequencer Program) สำหรับช่วยให้นักดนตรีสร้างเพลงได้ง่ายและมีคุณภาพมากขึ้น โปรแกรมช่วยการเรียนรู้ทางด้านดนตรี เช่น โปรแกรมเสริมการสอนด้านดนตรี

ส่วนในด้านการศึกษา ในต่างประเทศได้มีการตื่นตัวและใช้งานคอมพิวเตอร์อย่างจริงจัง ถือได้ว่าเป็นเทคโนโลยีใหม่ทางการศึกษา โดยใช้เครื่องช่วยสอน (Teaching Machine) คือ คอมพิวเตอร์ช่วยการเรียนการสอน (Computer Aided

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instrution : CAI) ซึ่งมีส่วนประกอบที่สำคัญคือ รายการสอน (Programs) เนื่องจากในประเทศที่เจริญก้าวหน้าทางเทคโนโลยี มีอุปกรณ์ และ บุคคลากรพร้อมมูล การใช้งานคอมพิวเตอร์เพื่อการศึกษาจึงประสบผลสำเร็จเป็นอย่างดี

สำหรับในประเทศไทยการใช้งานคอมพิวเตอร์เพื่อการเรียนการสอนยังเป็นเพียงนวกรรมทางการเรียนการสอน (Instructional Innovation) ซึ่งหมายความว่า เป็นเพียงสิ่งแปลกใหม่ที่ยังไม่มีการใช้งานจริง ปัญหาประการหนึ่งคอมพิวเตอร์ไม่ได้รับการพัฒนาให้เป็นเทคโนโลยีทางการเรียนการสอนภายในประเทศก็คือ ขาดผู้เขียนโปรแกรม (Programer) ทางด้านนี้โดยตรง

ประโยชน์ของการนำคอมพิวเตอร์มาใช้ในการเรียนการสอน สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. ส่งเสริมการเรียนการสอนเป็นรายบุคคล
2. ช่วยประหยัดเวลาในการสอน และ ทุ่นแรงผู้สอน
3. ช่วยให้เกิดการและเปลี่ยนทรรศนะ ระหว่างผู้สอน และ ผู้เรียนได้เร็วขึ้น
4. ช่วยให้ผู้เรียนมีประสบการณ์กว้างขวางยิ่งขึ้น
5. ส่งเสริมให้ผู้เรียนรู้จักคิด และ แก้ปัญหาต่างๆ ได้ดีขึ้น
6. ส่งเสริมการเรียนการสอนให้มีประสิทธิภาพยิ่งขึ้น เพราะการสอนไม่ได้มุ่งที่เนื้อหาอย่างเดียว แต่เน้นวิธีการเรียนรู้ และการแสวงหาความรู้ด้วยตัวเอง

คอมพิวเตอร์เป็นเครื่องกลที่มีประสิทธิภาพสูง ในการทำงานเกี่ยวกับข้อมูล สามารถประมวลผล หรือคำนวณได้อย่างรวดเร็ว แม่นยำ สามารถแสดงผลหรือโต้ตอบกับผู้ใช้ได้ตามขอบเขตของโปรแกรม และตามขนาดของเครื่อง ดังนั้นการนำคอมพิวเตอร์มาใช้เพื่อช่วยในการเรียนการสอนให้บรรลุผล จึงขึ้นอยู่กับการจัดลำดับของโปรแกรมให้ถูกต้องตามหลักของการเรียนรู้ การเลือกเนื้อหาวิชาที่เหมาะสมกับลักษณะการทำงานของคอมพิวเตอร์ ตลอดจนความสามารถของเครื่อง และอุปกรณ์สนับสนุนต่างๆ เช่น เครื่องอ่านและบันทึกข้อมูล อุปกรณ์แสดงผล แผ่นบันทึกข้อมูล เป็นต้น

การใช้คอมพิวเตอร์เพื่อช่วยการเรียนการสอน จะใช้ในฐานะอุปกรณ์โสตทัศนศึกษาทำหน้าที่เป็นสื่อการเรียนการสอน (Instruction Media) ในการเรียนการสอนแบบโปรแกรม ซึ่งหมายถึงการจัดลำดับประสบการณ์การเรียนการสอนตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถของแต่ละคน ผู้เรียนจะเกิดการเรียนรู้ด้วยตัวเองจากการแสดงพฤติกรรมที่ถูกต้องของตนเอง ซึ่งเป็นผลให้ผู้เรียนเกิดการตอบสนองซ้ำอีกถ้ามีเงื่อนไขที่ถูกต้องและเมื่อมีการเสริมแรงที่เหมาะสม การเรียนการสอนแบบโปรแกรมจึงมีลักษณะที่สำคัญดังนี้

1. เปิดโอกาสให้ผู้เรียนร่วมกิจกรรมอย่างกระฉับกระเฉง (Active Participation) คือให้ผู้เรียนได้เรียนรู้ด้วยการลงมือทำกิจกรรมด้วยตัวเอง ได้ตอบคำถาม มีการตอบสนอง หรือมีปฏิกิริยาโต้ตอบกับคำถาม หรือวัสดุอุปกรณ์ เกี่ยวกับที่จะเรียนรู้ ซึ่งจะช่วยให้ผู้เรียนมีความกระตือรือล้น ไม่เบื่อหน่าย และไม่เฉื่อยชา

2. ให้ผู้เรียนได้รู้ผลการเรียนของตนทันที (Feedback) คือให้เรียนรู้ถึงคำตอบที่ถูกต้อง ซึ่งเรียกว่าเป็นการให้ข้อมูลย้อนกลับ ข้อมูลย้อนกลับที่เป็นที่พึงพอใจจะเป็นการเสริมแรง (Reinforcement) ให้ผู้เรียนอยากทำกิจกรรมต่อไป และถ้าได้รับการเสริมแรงอยู่เสมอ จะทำให้เขาเรียนได้ดีที่สุด แต่ในทางตรงกันข้ามข้อมูลย้อนกลับอาจทำให้ผู้เรียนเกิดความท้อแท้ได้ ถ้าข้อมูลไม่เป็นที่พึงพอใจ ในกรณีเช่นนี้ ผู้เรียนต้องได้รับการแนะนำชี้แจงแก้ไขโดยทันที เพื่อให้เขาเรียนรู้ได้ดีขึ้น และฝึกทำซ้ำจนเกิดความชำนาญเพื่อจูงใจให้เขาอยากเรียนต่อไป

3. ให้ผู้เรียนได้มีประสบการณ์แห่งความสำเร็จเป็นระยะๆ (Successful Experiences) หลักจิตวิทยาแห่งการเรียนรู้แสดงให้เห็นว่า การเสริมแรงจะต้องกระทำอย่างเฉียบพลันทันทีจึงจะเป็นผลดีต่อการเรียนรู้ ถ้าหากการเสริมแรงล่าช้าออกไปตัวเสริมแรงจะลดประสิทธิภาพลงไป ดังนั้น การรู้ว่าคำตอบถูกหรือความสามารถในการทำกิจกรรมสำเร็จในขั้นตอนหนึ่ง จะเป็นการเสริมแรงให้ผู้เรียน เรียนรู้ในระดับขั้นตอนอื่นอีกต่อไป

4. การจัดลำดับเนื้อหาและประสบการณ์เป็นขั้นตอนย่อยๆ ต่อเนื่องกัน (Gradual Approximation) เพื่อนำไปสู่การบรรลุเป้าหมายขั้นสุดท้าย เช่น เริ่มจากเรื่องง่ายๆหรือสิ่งที่รู้แล้ว เพื่อสร้างแรงจูงใจเริ่มแรกเสียก่อน แล้วจึงเพิ่มความยากขึ้นไป การเรียนรู้จากสิ่งง่ายๆ ที่คล้ายคลึงกับประสบการณ์เดิม ไปสู่สิ่งใหม่ตามขั้นตอนทีละน้อย จะช่วยให้ผู้เรียนรู้เกิดความรู้สึกว่าบทเรียนนั้นไม่ยากจนเกินไป และไม่เบื่อหน่ายต่อการเรียน ดังนั้นโปรแกรมการเรียนการสอนจะต้องจัดลำดับเนื้อหาเป็นขั้นตอน เริ่มจาก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ง่ายแล้วเพิ่มความยากขึ้นตามลำดับ

ปกติการเลือกสื่อการเรียนการสอน เป็นขั้นตอนหนึ่งของการสร้างบทเรียนแบบโปรแกรม การเรียนที่ใช้สื่อที่เหมาะสมเท่านั้น จึงจะทำให้ผู้เรียนเกิดการเรียนรู้ได้สูงสุด เนื่องจากไม่มีสื่อชนิดใดดีพร้อมสำหรับทุกเนื้อหา ดังนั้นการเลือกใช้คอมพิวเตอร์เพื่อเป็นสื่อในการออกแบบการสอนแบบโปรแกรม จึงต้องพิจารณาความเหมาะสมดังนี้

1. เนื้อหาวิชาที่ต้องอาศัยมโนภาพ หรือจิตนาการเพื่อช่วยทำความเข้าใจ เช่นวิชาแม่เหล็กไฟฟ้า อธิบายถึงการเกิดสนามแม่เหล็กไฟฟ้า คอมพิวเตอร์จะช่วยวาดภาพเพื่อให้เห็นสภาวะเหล่านั้นได้

2. เนื้อหาวิชาที่มีการคำนวณ และใช้ฟังก์ชัน (Function) ทางคณิตศาสตร์ เช่น การคำนวณค่าต่างๆ เพื่อนำมาเขียนกราฟ คอมพิวเตอร์จะช่วยคำนวณ และวาดภาพตามข้อมูลที่คำนวณ หรือประมวลผลได้

3. ลักษณะเนื้อหาที่มีการเกิดหรือการเปลี่ยนแปลงอย่างต่อเนื่อง ที่เกิดขึ้นอย่างรวดเร็ว หรือใช้เวลายาวนาน เช่น การทำงานของเครื่องจักรกล การเจริญเติบโตของพืชและสัตว์ คอมพิวเตอร์จะช่วยจำลองสถานการณ์ให้เห็นได้ในช่วงเวลาที่เหมาะสม

4. เนื้อหาวิชาที่ต้องการการสาธิต แต่มีปัญหาเรื่องอุปกรณ์มีขนาดใหญ่ ราคาแพง บอบบาง จัดหาไม่ได้ มีอันตรายสูง หรือใช้ในพื้นที่ที่กว้างเกินกว่าที่จะสาธิตในห้องทดลองได้ คอมพิวเตอร์จะช่วยสร้างภาพเพื่อจำลองของจริงให้เห็นทางจอภาพได้

5. ลักษณะเนื้อหาวิชาที่มีลำดับขั้นตอนซับซ้อน ต้องการการจัดเรียงลำดับที่เหมาะสม ผู้เรียนต้องการการทบทวน เรียนซ้ำ คอมพิวเตอร์จะช่วยให้ความสะดวกในการจัดเรียงลำดับเนื้อหา และการเรียนซ้ำโดยไม่จำกัดครั้ง

6. ใช้ในเนื้อหารายวิชาที่ผู้สอนต้องการเร้าความสนใจผู้เรียนเป็นพิเศษ หรือในรายวิชาที่เกี่ยวข้องกับคอมพิวเตอร์โดยตรง เช่น การเขียนโปรแกรมคอมพิวเตอร์

แม้ว่าคอมพิวเตอร์จะมีความสามารถสูงเพียงไร แต่ก็ยังไม่เป็นที่ยอมรับว่าจะสามารถใช้แทนผู้สอนที่เป็นมนุษย์ได้จริงๆ เนื่องจากการเรียนการสอนไม่ใช่เป็นเพียงการถ่ายทอดความรู้เพียงอย่างเดียว หากแต่เกี่ยวข้องกับการสร้างทัศนคติ อุดมคติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่านิยม อันเป็นเรื่องละเอียดอ่อน ต้องอาศัยคุณลักษณะที่มีอยู่ในตัวมนุษย์ เป็นส่วนสำคัญในการถ่ายทอด ดังนั้นบทบาทของคอมพิวเตอร์ที่น่าจะเป็น สำหรับงานการศึกษาก็คืออุปกรณ์โสตทัศนศึกษาที่ดีเยี่ยมชิ้นหนึ่ง

ผลตอบแทนที่คุ้มค่าจึงน่าจะมาจากการใช้งานอย่างเหมาะสม นั่นคือ การต้องการการวิจัยอย่างจริงจังก่อนที่จะนำไปใช้จริง สำหรับประเทศที่ยากจนแบบเรา การที่จะซื้อคอมพิวเตอร์ เพื่อนำมาใช้ในสถานศึกษาทั่วประเทศนั้น ไม่ใช่เรื่องเล็กน้อย ต้องการการวิจัยเพื่อให้มองเห็นผลที่คุ้มค่า ทั้งในด้านการลงทุน การบำรุงรักษา อายุการใช้งาน รวมทั้งผลตอบแทนในแง่การศึกษา และแนวโน้มที่จะเกิดขึ้นในอนาคตในเรื่องตลาดและการผลิตอุปกรณ์สนับสนุนการใช้เครื่อง

ปริญญานิพนธ์ฉบับนี้ จะได้กล่าวถึงรายละเอียดของโครงงาน แต่ละบทดังต่อไปนี้

บทที่ 1 : บทนำ

บทที่ 2 : ทฤษฎีและหลักการใช้

บทที่ 3 : การสร้างและใช้งานตัวกลางมีดี้

บทที่ 4 : การสร้างและผลการทำงานของโปรแกรมเสริมการสอนด้านดนตรี

บทที่ 5 : บทวิจารณ์และสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

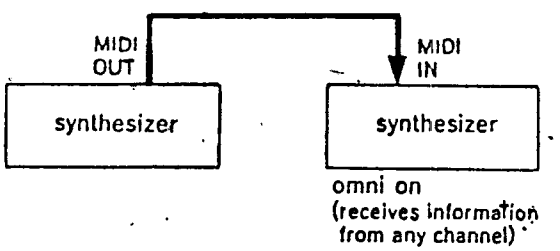
ทฤษฎีและหลักการใช้งาน

2.1 ความรู้ทั่วไปเกี่ยวกับเครื่องดนตรีอิเลคทรอนิคส์

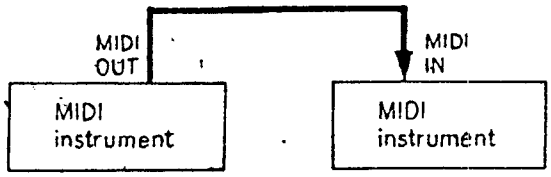
เครื่องดนตรีอิเลคทรอนิคส์ เป็นอุปกรณ์ที่ผลิตเสียงเลียนแบบเครื่องดนตรีจริงโดยใช้วงจรทางอิเลคทรอนิคส์ เครื่องดนตรีลักษณะนี้ที่พบทั่วไปได้แก่ เครื่องสังเคราะห์เสียงดนตรี ซึ่งปกติจะผลิตเสียงผ่านวงจรสร้างความถี่ (Oscillator) , ตัวกรองสัญญาณ (Filter) และ อุปกรณ์ที่ทำงานเกี่ยวกับคลื่น (Wave Shaping) อื่นๆ เครื่องสังเคราะห์เสียงดนตรีสมัยใหม่ สามารถเก็บค่าต่างๆของเสียงที่ผลิตเองได้ไว้ในโปรแกรมภายในตัวเอง แล้วทำให้เกิดเป็นเสียงโดยใช้ แหล่งกำเนิดเสียงของเครื่องสังเคราะห์เสียงดนตรี (Synthesizer's Sound Generator) ผลิตเสียงแบบต่างๆ ตามโปรแกรมที่เปลี่ยนไป เครื่องสังเคราะห์เสียงดนตรีส่วนมากสามารถเก็บโปรแกรมได้ตั้งแต่ 32 ถึง 128 โปรแกรม

2.2 มีดี้กับเครื่องดนตรีอิเลคทรอนิคส์

เราอาจเรียกได้ว่า มีดี้เป็นภาษาระหว่างเครื่องดนตรี ที่มีการใช้งานอย่างกว้างขวาง ใช้ติดต่อกับเครื่องดนตรีหลายชนิดรวมกันได้ ที่เครื่องดนตรีจะมีตัวเชื่อมต่อ (Connector) ของมีดี้ คือ มีดี้เข้า (MIDI In) มีดีออก (MIDI Out) และมีดี้ผ่าน (MIDI Thru) ซึ่งทำหน้าที่เป็นเหมือนหูและปากของการติดต่อ การติดต่อทางมีดี้จะส่งข้อมูลจากมีดีออกของเครื่องหนึ่งไปยังมีดี้เข้าของอีกเครื่องหนึ่ง โดยผ่านทางสายเคเบิ้ล ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

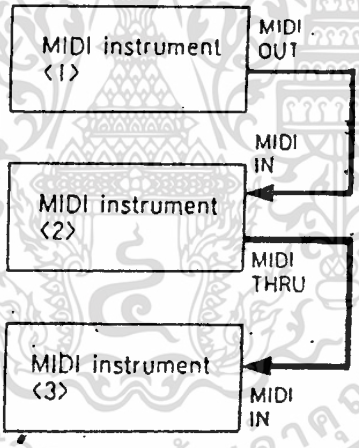


(รูปแสดงการติดต่อ มีดี้ออก และ มีดี้เข้า)

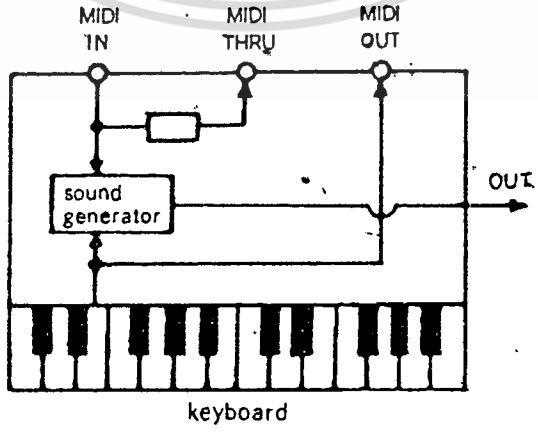
มีดี้เข้า : ใช้สำหรับเป็นทางเข้าของข้อมูล

มีดี้ออก : ใช้สำหรับเป็นทางออกของข้อมูล

มีดี้ผ่าน : ใช้สำหรับเป็นทางออกของข้อมูลที่รับผ่านมาจากมีดี้เข้า ไม่ใช่ข้อมูลที่เครื่องนั้นส่งออกไป



● construction of MIDI Keyboard

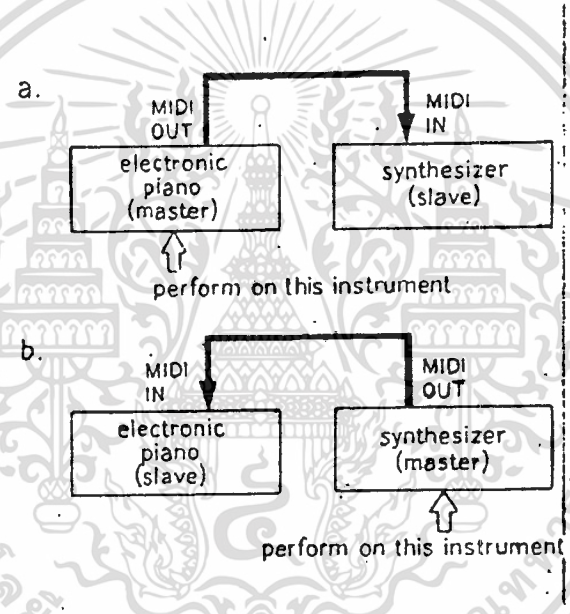


(รูปแสดงการติดต่อของอุปกรณ์ 3 ตัว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป ข้อมูลมีดี้จะออกจากมีดี้ออกของอุปกรณ์ตัวที่หนึ่ง เข้าไปยังมีดี้เข้าของอุปกรณ์ตัวที่สอง และผ่านทางมีดี้ผ่านไปยังมีดี้เข้าของอุปกรณ์ตัวที่สาม ข้อมูลจากอุปกรณ์ตัวที่หนึ่งสามารถควบคุมได้ทั้งอุปกรณ์ตัวที่สองและสาม การทำงานร่วมกันของอุปกรณ์ตั้งแต่ 2 ตัวขึ้นไปเรียกว่า ระบบมีดี้ (MIDI System)

ข้อมูลที่ส่งจากมีดี้ออกไปยังมีดี้เข้า หรือจากมีดี้ผ่านไปยังมีดี้เข้า โดยมากเป็นแบบทางเดียว ดังนั้นอุปกรณ์จะแยกเป็น อุปกรณ์ตัวแม่ (Master) ทำหน้าที่เหมือนเป็นผู้พูด และอุปกรณ์ตัวลูก (Slave) ทำหน้าที่เหมือนเป็นผู้ฟัง



(รูปแสดงการติดต่อระหว่าง เครื่องสังเคราะห์เสียงดนตรี และเปียโนไฟฟ้า)

รูปบน ข้อมูลจะถูกส่งจากเปียโนไฟฟ้า (Electronic Piano) ไปยังเครื่องสังเคราะห์เสียงดนตรี กรณีนี้เปียโนไฟฟ้าเป็นอุปกรณ์ตัวแม่ เครื่องสังเคราะห์เสียงดนตรีเป็นอุปกรณ์ตัวลูก

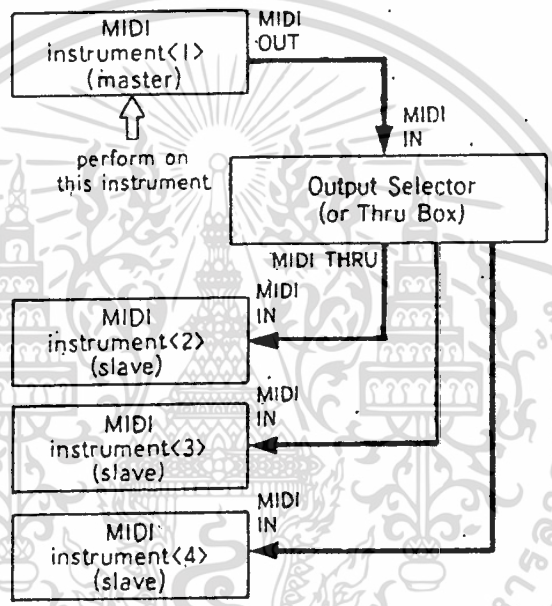
รูปล่าง ข้อมูลจะถูกส่งจากเครื่องสังเคราะห์เสียงดนตรีไปยังเปียโนไฟฟ้า กรณีนี้เครื่องสังเคราะห์เสียงดนตรีเป็นอุปกรณ์ตัวแม่ และอุปกรณ์ไฟฟ้าเป็นอุปกรณ์ตัวลูก

นอกจากนี้ความสัมพันธ์ระหว่างอุปกรณ์ตัวแม่ และอุปกรณ์ตัวลูก อาจไม่เป็นแบบหนึ่งต่อหนึ่ง โดยอุปกรณ์ตัวแม่หนึ่งตัวอาจติดต่อกับอุปกรณ์ตัวลูกสองตัว หรือสามตัว หรือมากกว่านี้ก็ได้ ในระบบมีดี้ใหญ่ๆ เราสามารถเห็นความสัมพันธ์ระหว่างอุปกรณ์ตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และอุปกรณ์ตัวลูกได้อย่างชัดเจน และทำงานได้ง่าย

การติดต่อโดยใช้มีดี้ผ่าน อาจทำให้ยุ่งยากขึ้น ถ้าใช้การผ่านทางมีดี้ผ่านหลายครั้งจนเกินไป จะรบกวนการติดต่อ อุปกรณ์ตัวหลังๆ อาจได้รับข้อมูลไม่ถูกต้อง เพื่อหลีกเลี่ยงสถานการณ์เช่นนี้ เราต้องวางอุปกรณ์ตัวแม่และอุปกรณ์ตัวลูก ให้ใกล้กันมากที่สุดเท่าที่จะทำได้ ซึ่งไม่ได้หมายความว่าต้องวางติดกันจริงๆ แต่มีการใช้กล่องมีดี้ผ่าน (MIDI Thru Box) หรือ ตัวเลือกอุปกรณ์รับ (Output Selector) แทน



2.3 ประโยชน์ของมีดี้

1. ทำให้เครื่องดนตรีสามารถเล่นพร้อมกันได้ ตัวอย่างเช่น การเล่นเครื่องดนตรี 2 ชิ้นพร้อมกัน โดยส่งข้อมูลจากเปียโนไฟฟ้าทางมีดี้ออก ไปยังมีดี้เข้าของเครื่องสังเคราะห์เสียงดนตรี ถ้าเครื่องสังเคราะห์เสียงดนตรีถูกกำหนดให้ทำเสียงสตริง การเล่นเปียโนก็จะทำให้เครื่องสังเคราะห์เสียงดนตรีเล่นไปด้วยเสียงสตริง ในระบบใหญ่ๆ มักใช้เล่นเป็นวงดนตรี เช่น ใช้เสียงฮาร์ปซีคอร์ด (Harpsichord) จากเปียโน และเสียงขลุ่ย (Flute) จากเครื่องสังเคราะห์เสียงดนตรีเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

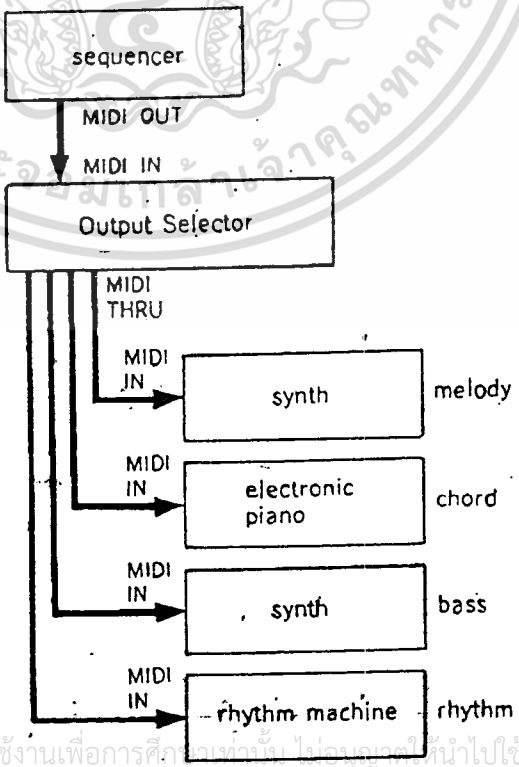
023131

นอกจากนี้ ยังเป็นวิธีทำให้เกิดการผสมเสียงของอุปกรณ์หลายๆ ตัว เป็นเสียงใหม่ที่มีเสียงเด่นและเสียงช่วยสนับสนุน ทำให้เกิดเสียงดนตรีต่างๆขึ้นมากมาย

2. ทำให้เครื่องดนตรีต่างชนิดใช้งานร่วมกันและติดต่อกันได้ โดยผ่านมีดี้ เช่น การติดต่อเปียโนกับเครื่องสร้างเสียงกลอง ให้เปียโนเป็นอุปกรณ์ตัวแม่ เครื่องสร้างเสียงเป็นอุปกรณ์ตัวลูก เมื่อเราเล่นเปียโนจะได้ยินเสียงจากเครื่องสร้างเสียงกลองด้วย แต่อาจเป็นโน้ตคนละตัว เราสามารถเล่นเสียงกลองได้โดยกดคีย์ที่เหมาะสมบนคีย์บอร์ด (Keyboard) ของเปียโน ซึ่งเราอาจเปลี่ยนสลับให้เครื่องสร้างเสียงกลองเป็นอุปกรณ์ตัวแม่ แล้วเปียโนเป็นอุปกรณ์ตัวลูก หรือใช้อุปกรณ์ชนิดอื่นๆ ก็ได้

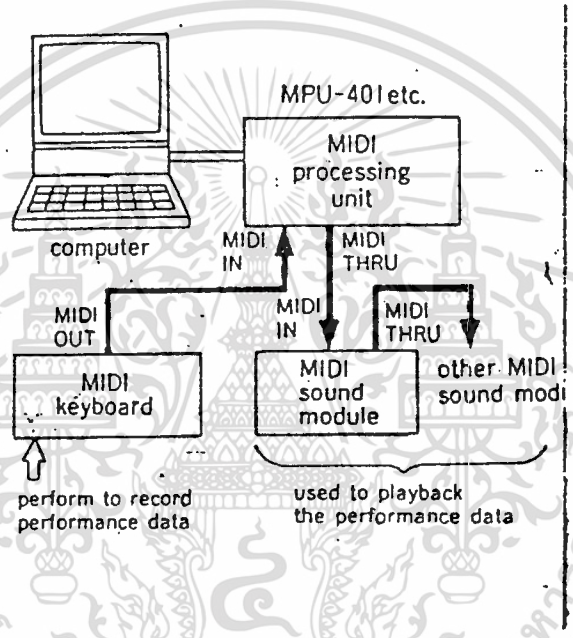
3. ในระบบจัดลำดับตัวโน้ต การใช้มีดี้ให้ประโยชน์มาก การเล่นได้พร้อมกันสามารถให้เสียงที่พิเศษขึ้น ช่วยในการเรียบเรียงเพลง หรือการเล่นเป็นวง แต่มีข้อจำกัดคือ ขึ้นอยู่กับความสามารถของเครื่องจัดลำดับตัวโน้ตที่เป็นอุปกรณ์ตัวแม่

ตัวอย่างเช่น เครื่องจัดลำดับตัวโน้ตทำหน้าที่เป็นอุปกรณ์แม่ ควบคุมให้เครื่องสังเคราะห์เสียงดนตรีเล่นทำนองเพลง (Melody) เปียโนเล่นคอร์ด (Chord) เครื่องสังเคราะห์เสียงอีกตัวเล่นเสียงเบส (Bass) และเสียงกลองในเวลาเดียวกัน ข้อมูลทั้งหมดจะเก็บไว้ในโปรแกรมจัดลำดับตัวโน้ต และถูกส่งไปยังอุปกรณ์ทั้ง 4 ได้เอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นำคอมพิวเตอร์มาจัดการทางด้านดนตรีได้ แต่คอมพิวเตอร์ไม่อาจติดต่อกับระบบมีดี้นี้ได้โดยตรง เพราะทั้งสองระบบมีการทำงานต่างกัน ต้องอาศัยตัวกลางมีดี้เป็นตัวแปลงข้อมูลจากคอมพิวเตอร์ไปเป็นข้อมูลแบบมีดี้ และ ข้อมูลแบบมีดี้เป็นข้อมูลคอมพิวเตอร์ เราใช้ซอฟท์แวร์เป็นตัวควบคุมการทำงาน เพื่อบอกให้คอมพิวเตอร์รู้ว่า จะต้องทำอะไรบ้าง โดยที่ซอฟท์แวร์หนึ่งอาจจะทำหน้าที่ให้คอมพิวเตอร์สร้างเสียง และอีกซอฟท์แวร์หนึ่งเป็นโปรแกรมจัดลำดับตัวโน้ต



จากรูป แสดงตัวอย่างการใช้ระบบคอมพิวเตอร์จัดการทางด้านดนตรีกับโปรแกรมจัดลำดับตัวโน้ต ประโยชน์ที่ระบบได้รับจากคอมพิวเตอร์คือ หน่วยความจำขนาดใหญ่สำหรับเก็บเพลงต่างๆ การแสดงผลบนจอภาพของคอมพิวเตอร์ ทำให้ง่ายต่อการตรวจสอบข้อมูล

5. ระบบมีดี้ยังสามารถนำมาประยุกต์ใช้งานด้านอื่นๆ อีก เช่น เปลี่ยนเสียงในระบบมีดี้เป็นตัวควบคุมการเปิดปิด การหน่วงเวลาต่างๆ

2.4 มาตรฐานมีดี้

มาตรฐานมีดี้เป็นชุดของข้อมูลที่มีความยุ่งยากซับซ้อน ใช้แสดงลักษณะการทำงานต่างๆ ของระบบมีดี้ ข้อมูลที่ส่งผ่านตัวกลางมีดี้จะส่งเป็นแบบอนุกรมอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

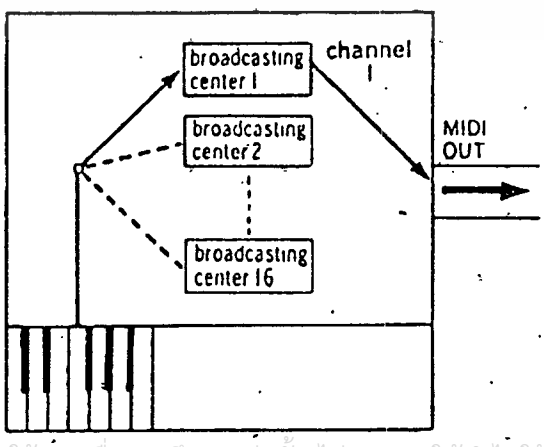
(Serial Asynchronous) ด้วยความเร็ว 31,250 บิทต่อวินาที และ แบ่งเป็น 2 ประเภทคือ ข้อมูลแสดงสถานะ (Status Byte) ประกอบด้วย 8 บิท บิทที่สำคัญที่สุดคือบิทที่ 7 จะถูกกำหนดให้เป็น 1 เสมอ และข้อมูลบอกรายละเอียด (Data Byte)

จากการรวมกันของข้อมูลทั้งสองประเภทนี้ ทำให้ได้ข่าวสาร (Message) สำหรับใช้ในการติดต่อสื่อสาร โดยที่ข้อมูลแสดงสถานะจะถูกส่งไปก่อน แล้วตามด้วยข้อมูลบอกรายละเอียด ซึ่งอาจจะมีชุดเดียวหรือหลายชุดก็ได้ ข่าวสารมีดี้แบ่งออกเป็น 2 ชุดคือ ข่าวสารแสดงชานแนล (Channel Message) และข่าวสารแสดงระบบ (System Message)

2.4.1 การควบคุมชานแนล

เมื่อระบบมีดี้สามารถใช้งานกับเครื่องดนตรีหลายๆ ชิ้นพร้อมกัน การส่งข้อมูลจึงจำเป็นต้องใช้ชานแนล มีดี้จะประกอบด้วย 16 ชานแนล หลักการของการส่งข่าวสารคล้ายกับการกระจายเสียง ซึ่งสถานีต่างๆ ส่งโปรแกรมในเวลาเดียวกัน อุปกรณ์รับก็จะรับพร้อมกันทั้ง 16 ชานแนล แต่สามารถเลือกชานแนลใดก็ได้เพราะสถานีส่งจะส่งต่างชานแนลกัน ระบบมีดี้ต่างจากระบบกระจายเสียง คือ ใช้ตัวกลางเป็นสายเคเบิ้ลแทนที่จะเป็นอากาศ

ข้อมูล 4 บิทล่างของไบท์แรกของข้อมูลแสดงชานแนลของเสียง ใช้กำหนดเลขที่ชานแนล เพื่อทำการส่งข้อมูลไปยัง 1 ใน 16 อุปกรณ์ที่ต่ออยู่กับทางเดินของตัวกลางมีดี้ (MIDI Interface Bus) ดังนั้นการใช้ชานแนลทำให้เครื่องดนตรี 16 ชิ้นสามารถเล่นผ่านทางเดินของตัวกลางมีดี้เพียงเส้นเดียวได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง ถ้าอุปกรณ์ถูกกำหนดดังรูป เครื่องสังเคราะห์เสียงดนตรีตัวที่ 1 จะได้รับข้อมูลจากชานแนล 1 ผ่านทางทางเดินของตัวกลางมิดี้ เครื่องสังเคราะห์เสียงดนตรีตัวที่ 2 รับเฉพาะชานแนล 2 ถ้าเครื่องดนตรีตัวแม่กำหนดส่งข้อมูลทางมิดี้ชานแนล 1 ก็จะมีผลต่อเครื่องสังเคราะห์เสียงตัวที่ 1 เท่านั้น ทั้งๆที่เครื่องสังเคราะห์เสียงตัวที่ 2 ได้รับข้อมูลเช่นเดียวกันแต่ไม่มีผล เพราะต่างชานแนลกัน

การควบคุมทำได้ทั้งให้อุปกรณ์รับสนอง 16 ชานแนลพร้อม(Omni-on)หรือเลือกให้ตอบสนองชานแนลใดชานแนลหนึ่ง(Omni-off)ก็ได้ นอกจากนี้ยังสามารถเลือกได้ว่า เป็นการส่งแบบข้อมูลตัวโน้ตเดียวโมโนโฟนิก(Monophonic) หรือแบบข้อมูลหลายตัวโน้ตโพลีโฟนิก(polyphonic)ได้อีกด้วย

2.4.2 ข่าวสารแสดงชานแนล

ข่าวสารนี้ถูกส่งผ่านมิดี้ชานแนล เพื่อกำหนดการทำงานของอุปกรณ์ในระบบที่ต่อกับแชลแนลนั้น แบ่งออกเป็น

1. ข่าวสารแสดงชานแนลของเสียง(Channel Voice Message) ใช้ในการเปลี่ยนสถานะเสียงของเครื่องสังเคราะห์เสียงดนตรี มีรายละเอียดดังนี้

1.1 โน้ต-ออน (Note-On Channel Voice Message) จะถูกส่งเมื่อคีย์ถูกกด ข้อมูลบอกรายละเอียดในข่าวสารนี้ จะมีเลขที่ของโน้ต (Note Number) และความเร็วในการกดคีย์ (Velocity)

1.2 โน้ต-ออฟ (Note-Off Channel Voice Message) จะถูกส่งเมื่อปล่อยคีย์ประกอบด้วย เลขที่ของโน้ต และความเร็วในการกดคีย์เช่นเดียวกับโน้ตออน

1.3 ข่าวสารการเปลี่ยนโปรแกรม (Program Change Message) ถูกส่งเมื่อมีการเปลี่ยนโปรแกรม ข้อมูลบอกรายละเอียดเป็นเลขที่ของโปรแกรมใหม่

1.4 ข่าวสารการเปลี่ยนการควบคุม (Control Channel Message) ใช้เฉพาะกับเครื่องสังเคราะห์ดนตรีที่มี โน้ต-เบนเดอร์ (Note-Bender) หรือ พิทช์ วีล (Pitch Wheel) ในแง่ของมิดี้จะรู้จักในฐานะตัวควบคุมแบบต่อเนื่อง (Continuous Controller) โดยมีข่าวสารการเปลี่ยนการควบคุมเป็นตัวควบคุมการเปลี่ยนแปลงเลขที่ของตัวควบคุมและค่าจะถูกส่งตามหลังข้อมูลแสดงสถานะ เลขที่ของตัวควบคุม 122-127 จะถูกสงวนไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข่าวสารแสดงการควบคุมชานแนล (Channel Mode Message) เป็นการบอกว่าจะเลือกการควบคุมชานแนลในลักษณะใด จาก 4 ลักษณะดังนี้

2.1 ข้อมูลเสียงจะถูกรับพร้อมกันทุกชานแนล เสียงเป็นแบบโพลีโฟนิก(Omni-On Poly)

2.2 ข้อมูลเสียงจะถูกรับพร้อมกันทุกชานแนล เสียงเป็นแบบโมโนโฟนิก(Omni-On Mono)

2.3 ข้อมูลเสียงจะถูกรับโดยชานแนลใดชานแนลหนึ่งที่เลือกไว้ เสียงเป็นแบบโพลีโฟนิก(Omni-Off Poly)

2.4 ข้อมูลเสียงจะถูกรับโดยชานแนลใดชานแนลหนึ่งที่เลือกไว้ เสียงเป็นแบบโมโนโฟนิก(Omni-Off Mono)

	poly	mono
omni on	mode 1	mode 2
omni off	mode 3	mode 4

ข่าวสารการควบคุมชานแนลในลักษณะต่างๆที่กล่าวมา มีค่าระหว่าง 122 ถึง 127 จึงทำให้เครื่องสังเคราะห์เสียงบางเครื่องใช้ได้ถึง 6 เสียง โดยแยกชานแนลกัน

2.4.3 ข่าวสารแสดงระบบ

เนื่องจากอุปกรณ์ทุกตัวต่ออยู่กับทางเดินของตัวกลางมิดี้ จึงต้องกำหนดข่าวสารที่ใช้ควบคุมทั้งระบบ เพื่อหลีกเลี่ยงปัญหาต่างๆ โดยแบ่งเป็น

1. ข่าวสารที่จำเป็นในการควบคุมเวลาและการซิงโครไนซ์ของระบบ (System Synchronization) ซึ่งเรียกว่า ข่าวสารระบบควบคุมเวลาจริง (System Real Time Message) มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 ข่าวสารสัญญาณนาฬิกา (Timing Clock Message) ใช้ซิงโครไนซ์อุปกรณ์ต่างๆ ในระบบ เช่น เครื่องสร้างเสียงกลอง เครื่องจัดลำดับตัวโน้ต ข้อมูลนี้จะส่งด้วยความเร็ว 24 บิทต่อควอเตอร์โน้ต (Beat Per Quarter Note)

1.2 ข่าวสารบอกการเริ่มต้น (Start Message) บอกอุปกรณ์ในระบบให้เริ่มทำงาน

1.3 ข่าวสารต่อเนื่อง (Continuous Message) บอกให้อุปกรณ์ทำงานต่อที่สัญญาณนาฬิกาถัดไป หลังจากหยุดการทำงาน

1.4 ข่าวสารบอกการสิ้นสุด (Stop Message) บอกให้อุปกรณ์ในระบบหยุดทำงาน

1.5 ข่าวสารบอกการเริ่มต้นระบบใหม่ (System Reset Message) เป็นการเริ่มต้นการทำงานทั้งหมด เหมือนกับเปิดเครื่องใหม่

2. ข่าวสารระบบปกติ (System Common Message) ใช้กำหนดการทำงานและหน้าที่ต่างๆของอุปกรณ์ในระบบ มีรายละเอียดดังนี้

2.1 ข่าวสารชี้บอกตำแหน่งเพลง(Song Position Pointer Message) เป็นการกำหนดการนับจังหวะ (Beat Counter) ภายในของอุปกรณ์ให้มีค่าตามที่ต้องการ ตัวชี้บอกตำแหน่งเป็นรีจิสเตอร์ (Register) ภายใน ซึ่งเก็บค่าจังหวะมีดี้เริ่มต้นของเพลง และจะกำหนดให้เป็น 0 เมื่อข่าวสารบอกการเริ่มต้นถูกส่งออกไป

2.2 ข่าวสารสำหรับเลือกเพลง (Song Select Message) เป็นการกำหนดเพลงที่จะเล่น สามารถเลือกได้ถึง 128 เพลง

2.3 ข่าวสารการขอปรับแต่ง (Tune Request Message) ใช้กับเครื่องสังเคราะห์เสียงดนตรีแบบอนาล็อก เป็นการขอให้ปรับแต่งวงจรสร้างความถี่

3. ข่าวสารระบบพิเศษ (System Exclusive Message) เป็นการกำหนดเลขหมายที่แน่นอนแก่เครื่องดนตรีแต่ละชนิด (Manufacturers' ID Number) เพื่อให้เครื่องดนตรีหลายๆยี่ห้อสามารถเล่นร่วมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและใช้งานตัวกลางมีดี้

โครงงานนี้แบ่งการทำงานออกเป็น 2 ส่วน คือ สร้างวงจรตัวกลางมีดี้ และเขียนโปรแกรมใช้งานตัวกลางมีดี้ที่สร้างขึ้น

3.1 การเลือกอุปกรณ์ในการสร้างวงจรตัวกลางมีดี้

มีดี้มีการทำงานรับ-ส่งข้อมูลแบบอนุกรมอะซิงโครนัส ซึ่งโดยทั่วไปสิ่งที่อุปกรณ์ที่ใช้กับระบบมีดี้ควรจะมีคือ

1. พอร์ทมีดี้เข้า (MIDI IN Port) ใช้รับข้อมูลมิดี้
2. พอร์ทมีดี้ออก (MIDI OUT Port) ใช้ในการส่งข้อมูล
3. พอร์ทมีดี้ผ่าน (MIDI THRU Port) ใช้ในการทวนสัญญาณข้อมูลซึ่งรับมาจากพอร์ทมีดี้เข้า

การติดต่อแบบอนุกรมในระบบมีดี้ มีความเร็วในการรับ-ส่งข้อมูลประมาณ 31,250 บิทต่อวินาที (ผิดพลาดไม่เกิน 1 เปอร์เซนต์) ซึ่งสามารถสร้างความเร็วขนาดนี้ได้ง่ายโดยใช้แหล่งกำเนิดความถี่ 2 เมกกะเฮิร์ท (2MHz) ข้อมูลประกอบด้วย 1 บิทเริ่มต้น (1 Start Bit) 8 บิทข้อมูล (8 Data Bits) และ 1 บิทสิ้นสุด (1 Stop Bit)

การทำงานของวงจรนี้ใช้กระแสขนาด 5 มิลลิแอมแปร์(5 Milli-Ampare) เนื่องจากตัวรับข้อมูลนั้นเป็นอ๊อปโตไอโซเลต (Optoisolate) ดังนั้นจึงไม่ควรใช้กระแสเกิน 5 มิลลิแอมแปร์ สายที่ใช้ในการเชื่อมต่อระหว่างตัวกลางมิดี้กับอุปกรณ์อื่นไม่ควรมีความยาวเกิน 50 ฟุต และที่ปลายของสายด้านหนึ่งมีปลั๊กตัวผู้ 5 ขา เพื่อใช้เสียบเข้ากับเครื่องดนตรี

เนื่องจากตัวกลางมิดี้นี้มีการส่งข้อมูลแบบอนุกรมจึงต้องใช้ยูอาร์ท (UART) ในการเปลี่ยนโอนข้อมูล (Transfer) ในที่นี้เลือกใช้ Z-80 SIO

การจะเก็บค่าข้อมูลต่างๆนั้น เราจำเป็นต้องใช้ตัวนับเวลา (Timer) จึงได้เลือกตัวนับเวลาที่สามารถโปรแกรมได้ของบริษัทอินเทล เบอร์ 8253 ตัวนับเวลานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

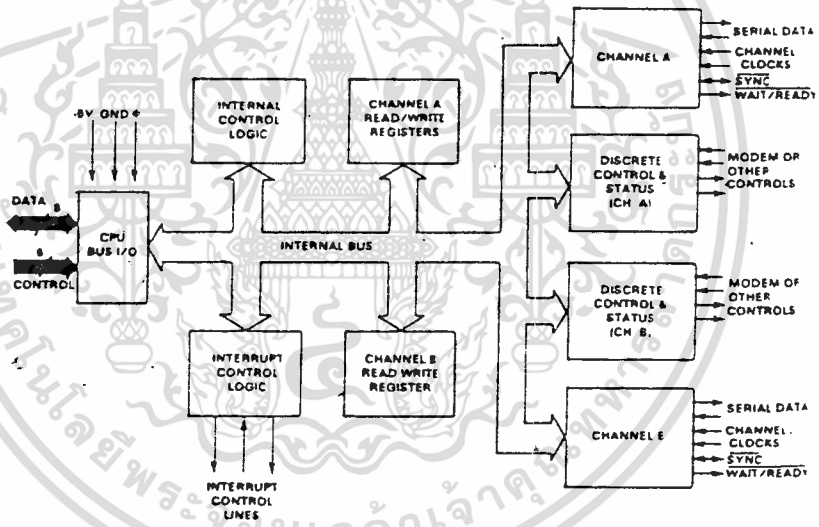
แบ่งเป็นตัวนับเวลาลง 3 ชุด สำหรับวงจรนี้ใช้เพียงแค่ 2 ชุดเท่านั้น

เพราะฉะนั้นวงจรตัวกลางมิดี้นี้จึงประกอบด้วยส่วนสำคัญ 3 ส่วนคือ

1. ส่วนของการเชื่อมระหว่างตัวกลางมีดี้และเครื่องดนตรี
2. ส่วนของวงจรสำหรับเชื่อมต่อ Z-80 SIO เข้ากับคอมพิวเตอร์
3. ส่วนของตัวนับเวลาและแหล่งกำเนิดความถี่

3.2 หลักการทำงานทั่วไปของ Z-80 SIO ที่ใช้ในวงจร

เป็นอุปกรณ์ที่มีลักษณะการทำงานแตกต่างกันหลายโหมด และประยุกต์ใช้งานได้หลายแบบ โดยจะเน้นถึงระบบการรับส่งข้อมูลแบบอนุกรมที่ใช้กับอุปกรณ์อินพุท-เอาท์พุท (I/O Device)



จากรูปภาพข้างต้น เป็นบล็อกไดอะแกรมของ Z-80 SIO จะเห็นว่าประกอบด้วยชานแนลการรับส่งข้อมูลแบบอนุกรมซึ่งทำงานเป็นอิสระต่อกันถึง 2 ชานแนล คือ ชานแนลเอ และชานแนลบี โดยแต่ละบล็อกจะทำงานร่วมกับหน่วยควบคุม (Control Block) และหน่วยแสดงสถานะ (Status Bus) ซึ่งหน่วยควบคุมและหน่วยแสดงสถานะจะเชื่อมโยงกับอินพุท เอาท์พุทจำนวนมาก

โหมดการทำงานของ Z-80 SIO สามารถควบคุมได้ด้วยซอฟท์แวร์ ซึ่งจะต้องมีการโปรแกรมบล็อกเหล่านี้ เพื่อจัดลักษณะการทำงานให้พร้อมไว้ก่อนใช้งาน

Z-80 SIO ประกอบด้วย รีจีสเตอร์สำหรับการเขียน (Internal Write

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register) 8 ตัว สำหรับชานแนลบี และสำหรับชานแนลเอ มีเพียง 7 ตัว โดยรีจิสเตอร์เขียนตัวที่สอง ใช้เก็บอินเทอร์รัพท์เวกเตอร์ (Interrupt) สำหรับทั้งสองชานแนล

ขั้นตอนทั่วไปในการเริ่มต้น (Initialize)

SIO สามารถรับ-ส่งข้อมูลครั้งละ5,6,7 หรือ 8 บิทต่อตัวอักษร ซึ่งสามารถโปรแกรมจำนวนบิทได้ทั้งทางด้านรับและด้านส่ง ในกรณีการส่งข้อมูลอนุกรมแบบอะซิงโครนัส SIO จะสร้างบิทเริ่มต้นซึ่งเป็นลอจิก 0 และสร้างบิทสิ้นสุดมีลอจิกเป็น 1 ซึ่งเป็นระดับลอจิกของสภาวะมาร์กกิ้ง (Marking)

สัญญาณนาฬิกาที่ป้อนให้กับ SIO เพื่อนำไปใช้ควบคุมอัตราการรับ-ส่งข้อมูลจะมี 4 สัญญาณด้วยกัน ทั้ง 4 สัญญาณจะเกี่ยวข้องกับขา RxA, RxB, TxA, TxB สัญญาณนาฬิกานี้มีความถี่เป็น 1, 16, 32 หรือ 64 เท่าของอัตราการส่งหรือรับข้อมูล ขั้นตอนที่ใช้สามารถสรุปได้ดังนี้

1. เขียนข้อมูลให้แก่ รีจีสเตอร์เขียน 0 (WR0) เพื่อเป็นการเริ่มการทำงาน

2. เขียนข้อมูลให้แก่ รีจิสเตอร์เขียน 1 (WR1) เพื่อเป็นการกำหนดตัวหารความถี่ (Baud Rate Factor) ที่ใช้ในการรับ-ส่งข้อมูลโดยให้เป็นค่าคูณ 64 และกำหนดขนาดของบิทสิ้นสุดเท่ากับ 1 บิท

3. เขียนข้อมูลให้แก่รีจิสเตอร์เขียน 3 (WR3) เพื่อเป็นการกำหนดจำนวนบิทของข้อมูลที่จะรับให้เท่ากับ 8 บิท และทำการอีนาเบิ้ลตัวรับ (Receiver Enable)กับอีนาเบิ้ลอัตโนมัติ (Auto Enable)

4. เขียนข้อมูลให้กับ รีจิสเตอร์เขียน 5 (WR5) เพื่อกำหนดจำนวนบิทของข้อมูลที่จะส่งเท่ากับ 8 บิทและทำการอีนาเบิ้ลตัวส่ง (Transmitter Enable)

5. เขียนข้อมูลลงใน รีจิสเตอร์เขียน 1 (WR1) เพื่อทำการอีนาเบิ้ล หรือดิสเอเบิ้ล (Disable) การขออินเทอร์รัพท์ของ SIO

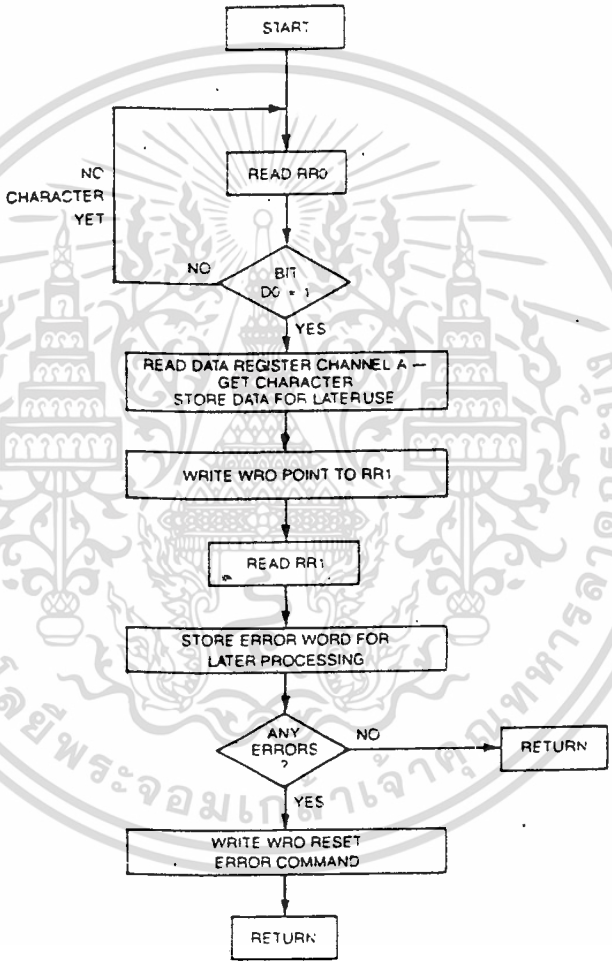
ขั้นตอนการรับข้อมูลอนุกรม

หลังจากทำการกำหนดการเริ่มต้นแล้ว หน่วยประมวลผลของคอมพิวเตอร์ก็พร้อมที่จะรับข้อมูลจาก SIO ได้ ใน Z-80 SIO มีหน่วยความจำแบบเข้าก่อนออกก่อน FIFO (First In First Out) ซึ่งช่วยให้การรับข้อมูลไม่เกิดความผิดพลาดโอเวอรัน (Overrun Error) ข้อมูลที่ส่งผ่าน SIO จะส่งผ่านทีละ 8 บิท ทั้งนี้เพื่อให้หน่วยประ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มวลผลนำข้อมูลไปทำตามขบวนการต่างๆ ได้ทันก่อนที่ข้อมูลใหม่จะผ่านเข้ามา

ในกระบวนการรับส่งข้อมูล จะต้องมีการตรวจสอบความผิดพลาดโอเวอร์รันในแต่ละตัวอักษรเสมอ ถ้าหากเกิดมีการผิดพลาดขึ้นมา หน่วยประมวลผลจะทำการรีเซ็ทแฟลก (Resat Flag) ที่แสดงชนิดของความผิดพลาด ก่อนที่จะรับตัวอักษรอื่นๆต่อไป ค่าความผิดพลาดในแฟลกจะคงอยู่จนกว่าจะถูกรีเซ็ท ดังแสดงขั้นตอนต่างๆ ไว้ในรูป



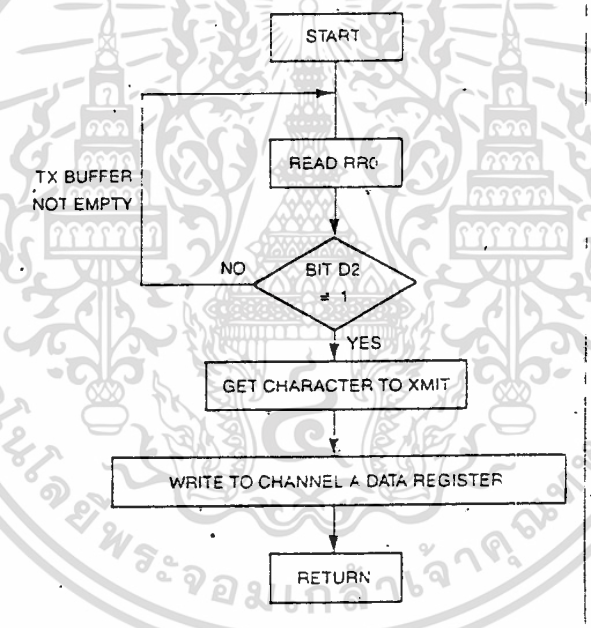
จากรูป เริ่มต้นต้องอ่านค่าข้อมูลจากรีจีสเตอร์สถานะ (RR0) ก่อน เพื่อตรวจสอบว่ามีข้อมูลเข้ามาหรือยัง โดยตรวจสอบต่อไปว่าบิทที่ 1 ของรีจีสเตอร์เป็น 0 หรือ 1 ถ้าเป็น 1 แสดงว่ามีข้อมูลอยู่ในหน่วยความจำเข้าก่อนออกก่อน แต่ถ้าเป็น 0 แสดงว่ายังไม่มี จะต้องวนกลับไปตรวจสอบใหม่ หลังจากรับตัวอักษรมาแล้วจะทำการอ่านรีจีสเตอร์อ่าน 1 (RR1) เพื่อตรวจสอบว่ามีความผิดพลาดเกิดขึ้นในตัวอักษรที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับเข้ามาหรือไม่ โดยมีบิทที่ 6 แสดงความผิดพลาดโอเวอร์รัน ถ้าเกิดความผิดพลาดขึ้นสามารถรีเซ็ทความผิดพลาดได้โดยเขียนข้อมูลไปยังรีจีสเตอร์เขียน 0 (WR0)

ขั้นตอนการส่งข้อมูล

ขั้นตอนแรกสุด ทำการอ่านข้อมูลในรีจีสเตอร์สถานะ เพื่อตรวจสอบว่าบัฟเฟอร์ตัวส่ง (Transmit Buffer) ว่างหรือไม่ โดยถ้าบิทที่ 3 มีค่าเป็น 1 แสดงว่าบัฟเฟอร์ว่าง แต่ถ้าบิทนี้มีค่า 0 แสดงว่าบัฟเฟอร์มีข้อมูล จะทำการตรวจสอบจนกระทั่งบัฟเฟอร์ว่างแล้ว จึงทำงานในขั้นต่อไปคือ เขียนตัวอักษรที่จะส่งออกไปยังพอร์ทที่ใช้ในการส่งข้อมูล หลังจากนั้น SIO ก็จะเริ่มทำการส่งข้อมูลได้โดยอัตโนมัติ ดังรูป



3.3 หลักการทำงานทั่วไปของ 8253 ที่ใช้ในวงจร

ดังที่ได้กล่าวมาแล้วว่า ตัวนับเวลา 8253 มีตัวนับเวลาอยู่ 3 ชุด ซึ่งทั้ง 3 ชุดนี้มีลักษณะเหมือนกัน ดังแสดงในบล็อกไดอะแกรม

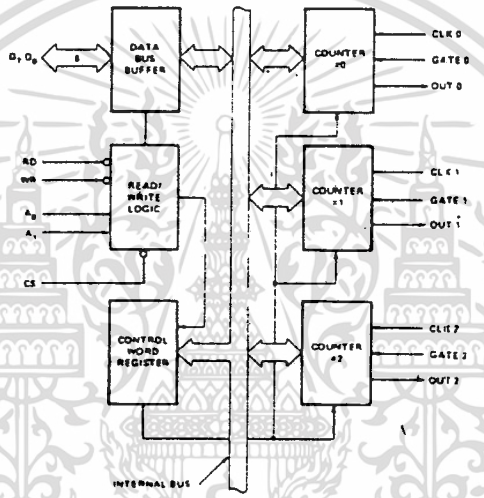
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE INTERVAL TIMER

- MCS-85™ Compatible 8253-5
- 3 Independent 16-Bit Counters
- DC to 2 MHz
- Programmable Counter Modes
- Count Binary or BCD
- Single +5V Supply
- 24-Pin Dual In-Line Package

The Intel® 8253 is a programmable counter/timer chip designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.



ส่วนของบัฟเฟอร์ทางเดินข้อมูล (Data Bus Buffer) เป็นที่พักข้อมูลเข้าและออกระหว่างเครื่องคอมพิวเตอร์กับรีจิสเตอร์ของ 8253 ส่วนของลอจิกการอ่านและเขียน (Read/Write Logic) ใช้สำหรับควบคุมการอ่านและเขียนของรีจิสเตอร์ของตัวนับเวลา และส่วนสุดท้ายที่จะกล่าวถึงเป็นส่วนที่เก็บข้อความที่ถูกโปรแกรมไว้ (Control Word Register)

รายละเอียดรีจิสเตอร์ภายในของ 8253 มีดังนี้

1. รีจิสเตอร์ควบคุมข้อความ (Control Word Register) ใช้ควบคุมการทำงานแบบต่างๆ และเลือกใช้วิธีการนับเวลาว่าจะให้นับแบบเลขฐานสอง หรือ แบบบีซีดี (Binary Code Decimal) ก่อนจะใช้งานต้องโปรแกรมข้อมูลให้กับรีจิสเตอร์นี้เสียก่อน เพื่อกำหนดการทำงานของตัวนับเวลา

2. ตัวนับเวลา 0 (ชุดที่ 1) , ตัวนับเวลา 1 (ชุดที่ 2) และตัวนับเวลา2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ชุดที่ 3) แต่ละตัวนับเวลามี 16 บิท ทำหน้าที่นับลง ซึ่งสามารถนับได้ทั้งแบบเลขฐานสอง หรือ แบบบีซีดีก็ได้ โดยที่การอ่านข้อมูลภายในตัวนับเวลาด้วยเครื่องคอมพิวเตอร์จะไม่รบกวนการนับเวลาของตัวนับเวลา การเลือกชุดนับเวลาทำได้โดยกำหนดการเลือกไว้ที่บิทที่ 7 และ 8 ของรีจิสเตอร์ควบคุมข้อความ

ลักษณะการทำงานแบ่งเป็น 6 โหมด คือ

1. โหมด 0 (Interrupt On Terminal Count)
2. โหมด 1 (Programmable One-Shot)
3. โหมด 2 (Rate Generater)
4. โหมด 3 (Square Wave Generater)
5. โหมด 4 (Software Triggered Strobe)
6. โหมด 5 (Hardware Triggered Strobe)

การเลือกโหมดการทำงานทำได้โดยกำหนดการเลือกไว้ที่บิท 2, 3 และ4 ของรีจิสเตอร์ข้อความ ในวงจรตัวกลางมีดี้นี้เลือกใช้เฉพาะ โหมด 0 และ โหมด 2 ซึ่งมีรายละเอียดการทำงานดังนี้

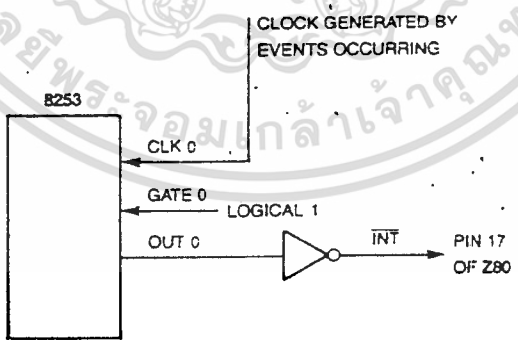
1. การทำงานของโหมด 0 ตัวนับเวลาจะเริ่มนับเวลาลงเมื่อขาออก (Out) มีค่าลอจิกเป็น 1 ค่าเริ่มต้นของการนับและรูปแบบการนับ (แบบเลขฐานสองหรือแบบบีซีดี) จะถูกโปรแกรมไว้ก่อน ตัวนับเวลาจะนับด้วยอัตราความถี่ของสัญญาณนาฬิกาที่ป้อนเข้ามา เมื่อนับลงจนถึงค่า 0 แล้ว ขาออกจะมีลอจิกเป็น 1 อีก ซึ่งค่าเริ่มต้นอาจใช้ค่าเดิมหรือเปลี่ยนค่าใหม่ก็ได้

2. การทำงานของโหมด 2 นี้ 8253จะถูกใช้เป็นตัวนับเวลาที่ทำหน้าที่ให้ค่าลอจิกเป็น 0 และ 1 สลับกันไปด้วยอัตราความถี่เท่ากับสัญญาณนาฬิกาที่เข้ามา หารด้วยค่าๆหนึ่ง ซึ่งทำให้ความกว้างของลอจิก 0 มีค่าเท่ากับ 1 คาบของสัญญาณนาฬิกาที่เข้ามา ส่วนช่วงความกว้างของลอจิก 1 มีค่าเท่ากับคาบของสัญญาณนาฬิกาที่เข้ามาคูณด้วยเลขจำนวนนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

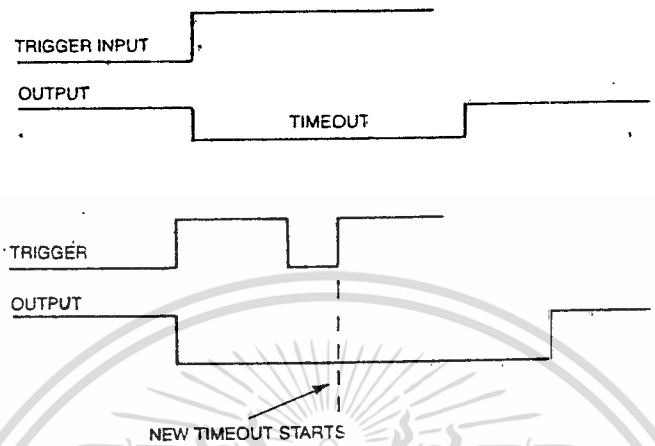
	RD	WR	A0	A1	
COUNTER 0	1	0	0	0	LOAD COUNTER 0
	0	1	0	0	READ COUNTER 0
COUNTER 1	1	0	0	1	LOAD COUNTER 1
	0	1	0	1	READ COUNTER 1
COUNTER 2	1	0	1	0	LOAD COUNTER 2
	0	1	1	0	READ COUNTER 2
MODE WORD OR	1	0	1	1	WRITE MODE WORD
CONTROL WORD	0	1	1	1	NO-OPERATION

รูปแสดงลอจิกที่กำหนดให้กับตัวนับเวลา 8253



รูปภาพแสดงการทำงานของ ตัวนับเวลาในโหมด 0 เมื่อตัวนับเวลา 8253 รับลอจิก 1 แล้วสัญญาณนาฬิกาจะถูกป้อนเข้ามา ตัวนับเวลาจึงเริ่มนับลง เมื่อนับถึง 0 จะให้ลอจิก 1 เพื่อขออินเทอร์รัพท์จาก Z-80

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ซึ่งค่าของเลขที่นำมาคูณนี้ เป็นค่าที่ได้จากการโปรแกรมให้กับตัวนับเวลาในโหมด 2 ถ้าขณะที่ตัวนับเวลายังทำงานในช่วงของลอจิก 1 อยู่ เกิดมีค่าตัวคูณค่าใหม่กำหนดเข้ามา ตัวนับเวลาก็จะยังทำงานในลักษณะเดิมจนกระทั่งหมดคาบ แล้วจึงจะทำตามค่าตัวคูณใหม่ที่เข้ามา

3.4 การกำหนดรายละเอียดต่างๆในวงจร

สำหรับเครื่องไมโครคอมพิวเตอร์ของบริษัท ไอบีเอ็ม (IBM) มักจะใช้ข้อมูลขนาด 9 บิท เพื่อกำหนดแอดเดรส (Address) ของอุปกรณ์อินพุท-เอาท์พุท แต่เราจะใช้ข้อมูลขนาด 15 บิทแทน เพื่อหลีกเลี่ยงความสับสนระหว่างการกำหนดแอดเดรสให้กับอุปกรณ์ตัวอื่นๆ ในระบบ

Z-80 SIO จะประกอบไปด้วย พอร์ทสถานะ และ พอร์ทข้อมูลสำหรับแต่ละชานแนล โดย

พอร์ทสถานะ : กำหนดแอดเดรสไว้ที่ตำแหน่ง FFA2h และ FFA3h

พอร์ทข้อมูล : สำหรับชานแนลเอ และ บี กำหนดแอดเดรสไว้ที่ตำแหน่ง FFA0h และ FFA1h ตามลำดับ

ตัวนับเวลา 8253 จะประกอบไปด้วย พอร์ทของแต่ละชุดของตัวนับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ พอร์ทของรีจิสเตอร์ควบคุมข้อความ โดย

พอร์ทควบคุมข้อความ : กำหนดแอดเดรสไว้ที่ตำแหน่ง FFA7h

พอร์ทของตัวนับเวลา : สำหรับตัวนับเวลา 0 และ 1 (ชุดที่ 1 และ 2) กำหนดแอดเดรสไว้ที่ตำแหน่ง FFA4h และ FFA5h ตามลำดับ

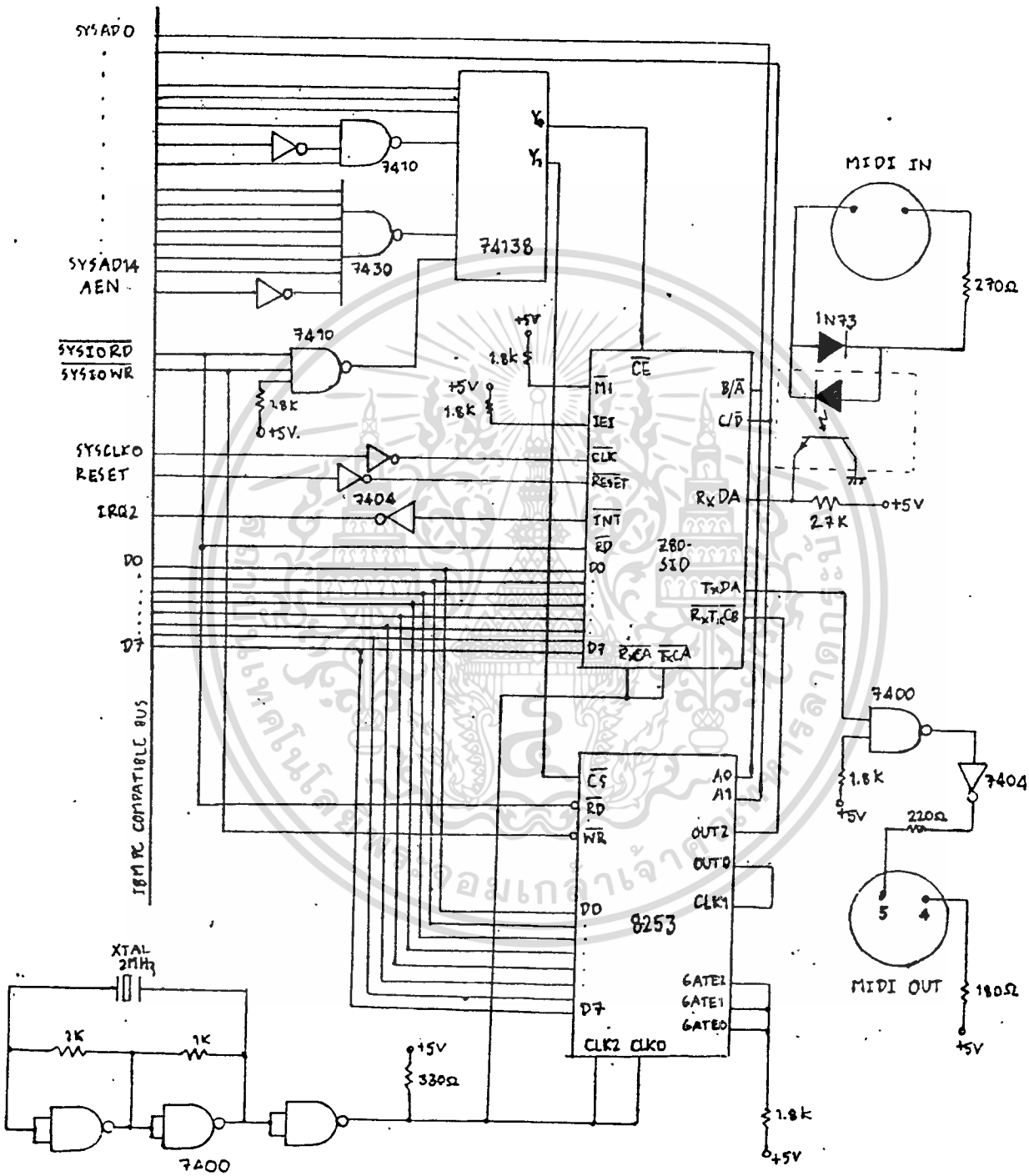
สัญญาณต่างๆ ที่ Z-80 SIO ต้องการจากเครื่องไมโครคอมพิวเตอร์ ได้แก่ สัญญาณ IORQ (Input/Output Request) ซึ่งถูกสร้างขึ้นโดยการนำเอาสัญญาณ SYSIORD (System Input/Output Read) กับ สัญญาณ SYSIOWR (System input/output Write) มาทำการออร์ (OR Operation) กัน

สำหรับตัวรับข้อมูล (RxDa) บนชานแนลเอ ของ SIO จะทำการเชื่อมกับทางเดินข้อมูล โดยผ่านอ๊อปโตไอโซเลเตอร์ ทีไอแอล 111 (TIL 111) สัญญาณมีดี้ที่เข้ามา ถูกป้องกันโดยไดโอดไอเอ็น 751 (IN 751) และถูกบัฟเฟอร์โดย อินเวอร์เตอร์ (Inverter)

และสำหรับตัวส่งข้อมูล (TxDa) บนชานแนลเอ จะรับส่งสัญญาณผ่านแนนเกต (NAND Gate) และถูกบัฟเฟอร์โดยอินเวอร์เตอร์ ข้อมูลที่ออกจากอินเวอร์เตอร์เหล่านี้ จะถูกส่งเป็นแบบอนุกรมไปเข้าตัวต้านทานขนาด 220 โอห์ม เพื่อทำการเข้าคู่อย่างอิสระกับทางเดินตัวกลางมีดี้ จึงทำให้สามารถสร้างได้ถึง 3 สัญญาณเอาท์พุท โดยไม่สิ้นเปลืองค่าใช้จ่ายมากนัก แต่ในโครงงานนี้ เลือกทดลองใช้เพียงสัญญาณเอาท์พุทเดียว

ตัวนับเวลาแต่ละชุดของ 8253 จะต้องมีลอจิก 1 ไปกระตุ้น (Trig) เพื่อเริ่มการทำงาน โดยตัวนับเวลา 0 จะได้รับสัญญาณนาฬิกาจากวงจรสร้างความถี่ที่ทำด้วยคริสตัล (Crystal Oscillator) ขนาด 2 เมกกะเฮิร์ต แล้วทำการหารความถี่ลงไปเป็นระดับ 24 จังหวะต่อควอเตอร์โน้ต เพื่อใช้เป็นสัญญาณนาฬิกาสำหรับตัวนับเวลา 1 ตัวนับเวลา 1 จะเป็นตัวเชื่อมระหว่างซอฟท์แวร์ที่ใช้ในการบันทึกเพลงและการทำงานของตัวกลางมีดี้ ให้สามารถสอดคล้องเล่นเป็นเพลงได้อย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การสร้างและผลการทำงานของโปรแกรมเสริมการสอนด้านดนตรี

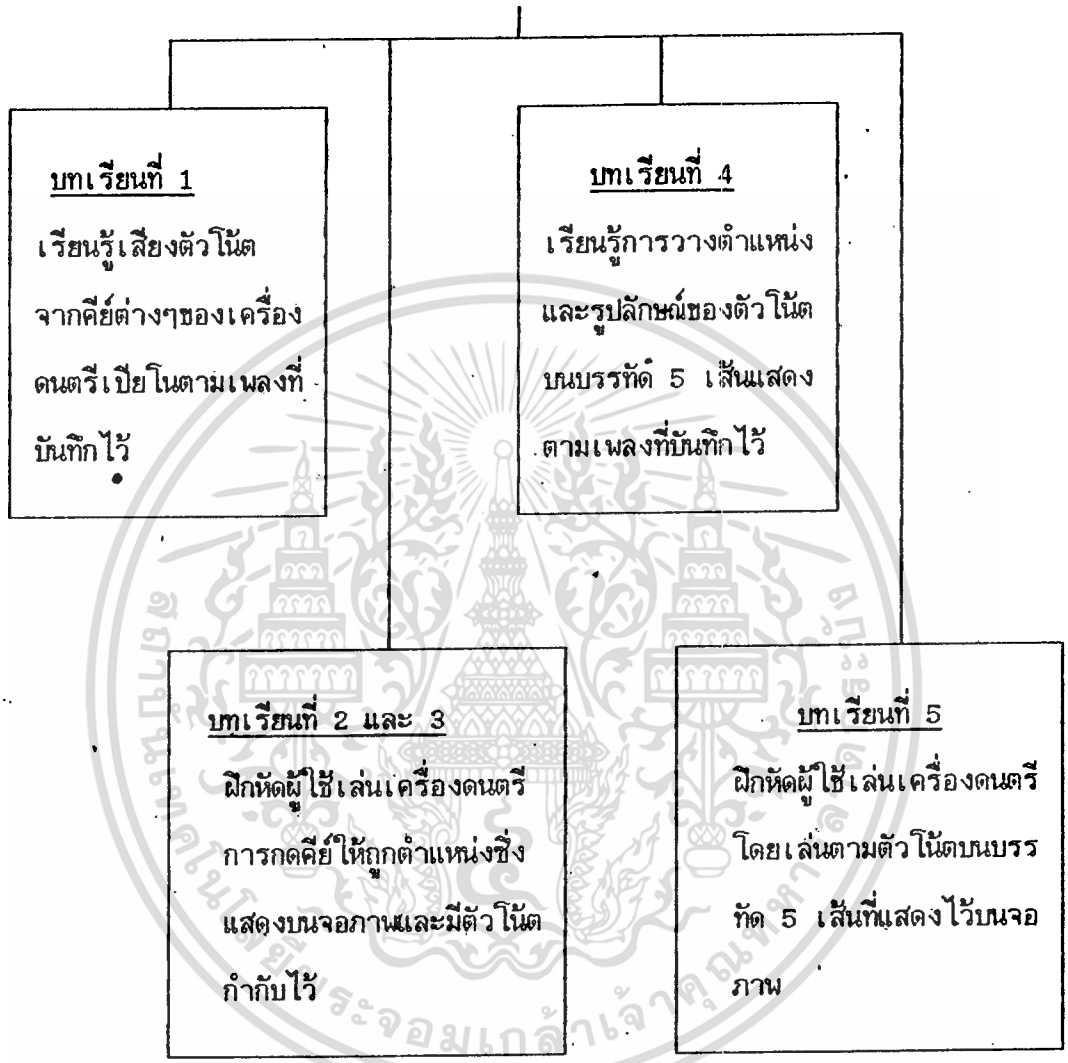
โปรแกรมดนตรีนี้สร้างขึ้น โดยมีจุดประสงค์หลักในการช่วยสอนความรู้พื้นฐานของดนตรีสากลแก่ผู้ที่ไม่เคยเรียนมาก่อนเลย ดังนั้น โปรแกรมจะแสดงผลออกมาในรูปแบบของบทเรียนที่ฝึกให้เรียนรู้และจดจำความสัมพันธ์ระหว่างตัวโน้ตกับเสียงดนตรี และคีย์ต่างๆของเครื่องดนตรีประเภทเปียโน และใช้สื่อรูปภาพมากกว่าการแสดงด้วยข้อความ เพื่อให้เหมาะสมแก่การเรียนรู้ และง่ายต่อการจดจำ

การทำงานของโปรแกรมแบ่งออกเป็นสองส่วน คือ โปรแกรมการทำงานรับส่งข้อมูลมีดี้ และ โปรแกรมแสดงผลบนจอภาพ ซึ่งแยกเป็นประเภทแสดงบนรูปเ เครื่องดนตรีเปียโน และ แสดงบนรูปบรรทัด 5 เส้น โดยให้โปรแกรมแสดงผลบนจอภาพสามารถเรียกใช้โปรแกรมรับ-ส่งข้อมูลมีดี้ได้ เมื่อต้องการติดต่อรับ-ส่งเสียงเพลงกับเครื่องดนตรี (ตามรูปแผนผัง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

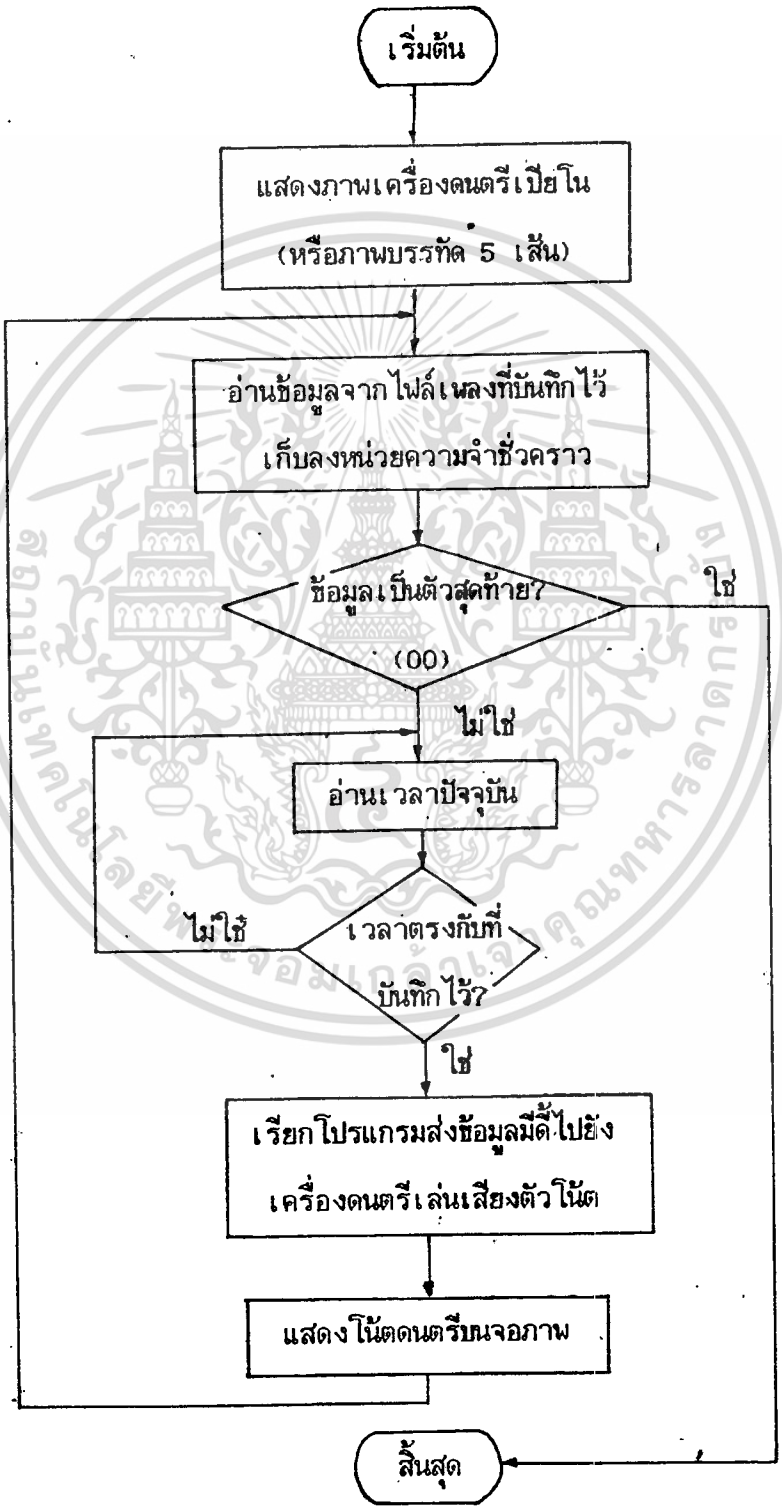
รูปแผนผังที่ 1

โปรแกรมหลัก



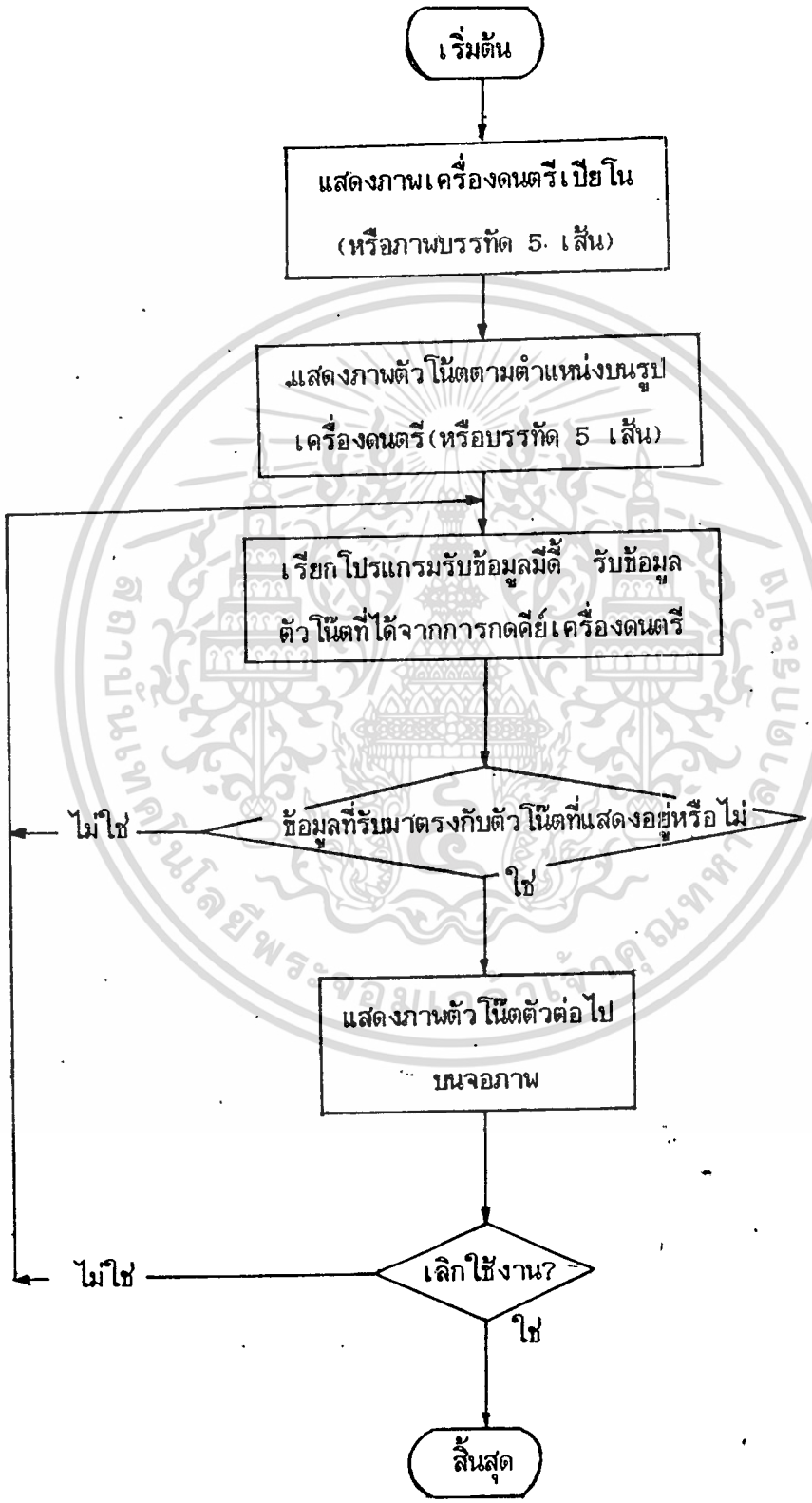
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแผนผังที่ 2 (บทเรียนที่ 1 และ 4)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแผนผังที่ 3 (บทเรียนที่ 3 และ 5)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำงานรับ-ส่งข้อมูลมีดี้ ใช้สำหรับบันทึกเพลง เล่นเพลงตามที่บันทึกไว้ และรับข้อมูลตามโน้ตที่ผู้ใช้งานเล่นตามเครื่องดนตรีขณะนั้น เพื่อแสดงออกทางจอภาพ ในส่วนของการบันทึกเพลงจะถูกนำมาใช้เฉพาะช่วงต้น ก่อนนำไปใช้งานสอนจริง สำหรับบันทึกเพลงที่เป็นต้นแบบ

การทำงานเริ่มต้นด้วยการกำหนดค่าต่างๆแก่ Z-80 SIO ทำให้การรับ-ส่งข้อมูลระหว่างเครื่องดนตรีและเครื่องคอมพิวเตอร์เป็นไปตามมาตรฐาน ข้อมูลมีดี้ที่รับเข้ามาเก็บไว้ในไฟล์ประกอบด้วย ข้อมูลบอกสถานะ ข้อมูลเวลา และ เลขที่ตัวโน้ต ซึ่งมีลักษณะรูปแบบดังรูป

ไบท์ที่ 1	ไบท์ที่ 2	ไบท์ที่ 3	ไบท์ที่ 4
ข้อมูล บอกสถานะ	ข้อมูลเวลา (Time Max)	ข้อมูลเวลา (Time Min)	เลขที่ตัวโน้ต 00 (สิ้นสุด)

ข้อมูลแสดงสถานะและข้อมูลเลขที่ตัวโน้ต ได้มาจากข่าวสารแสดงชานแนลของเสียง ข้อมูลเวลาได้จากการอ่านตัวนับเวลา 1 ของ 8253 และเนื่องจาก ตัวนับเวลา 0 ทำหน้าที่ให้สัญญาณนาฬิกาแก่ ตัวนับเวลา 1 เราจึงใช้การเปลี่ยนค่า กำหนดตัวนับเวลา 0 สำหรับเปลี่ยนค่าความเร็วของเพลง (Tempo) ตามที่ต้องการ ซึ่งจากการทดลองไม่สามารถอ่านเวลาจาก 8253 ได้ถูกต้อง จึงเปลี่ยนเป็นอ่านเวลาด้วยวิธีอื่นแทน คือ ใช้เวลานาฬิกาจากเครื่องคอมพิวเตอร์ แต่ทำให้ไม่อาจกำหนดความเร็วของเพลงขณะเล่นให้เร็วขึ้นหรือช้าลงได้

ข้อมูลบอกสถานะนั้น แบ่งออกเป็น โน้ต-ออน และ โน้ต-ออฟ ด้วยความเร็วในการกดคีย์เครื่องดนตรี (Velocity) เท่ากับ 64

ข้อมูลค่าของตัวโน้ต (Note Number) มีค่าตั้งแต่ 0 ถึง 127 ตามลำดับของเสียงเครื่องดนตรีเปียโนไฟฟ้า ซึ่งปกติให้เสียงอยู่ในช่วงระหว่างค่า 36 ถึง 127 ดังนั้น จึงแทนการสิ้นสุดการเล่นเพลงหรือโน้ตสุดท้ายด้วยค่าของโน้ต เท่ากับ 0

เอกสารนี้ ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มต้นบันทึกเพลง จะจองเนื้อที่ในหน่วยความจำชั่วคราวยาวประมาณ 100 ข้อมูลมีดี้ (400 ไบท์) เพื่อให้พอเพียงแก่เพลงที่มีความยาวสูงสุดประมาณ 45 ตัวโน้ต หลังจากสิ้นสุดการเล่นเพลงแล้ว โปรแกรมจะเก็บข้อมูลทั้งหมดลงในหน่วยความจำสำรอง หรือ เก็บลงแฟ้มข้อมูล ความยาวของแฟ้มข้อมูลขึ้นอยู่กับจำนวนตัวโน้ตที่เล่น ส่วนการเล่นเพลงที่บันทึกไว้จากการทดลองครั้งแรก ใช้การเปรียบเทียบข้อมูลเวลาที่เก็บไว้ของแต่ละข้อมูลมีดี้ กับ ค่าที่อ่านจากตัวนับเวลา 1 ของ 8253 และเมื่อข้อมูลทั้งสองพอดีกันจึงจะส่งข้อมูลมีดี้ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

จากบทที่ผ่านมาทั้งหมด ก็คงจะทำให้ผู้ที่อ่านปริญญานิพนธ์มาจนถึงบทนี้พอที่จะทราบถึงการประยุกต์นำคอมพิวเตอร์มาใช้งานด้านดนตรี และ ช่วยในการสอนดนตรีสากล รวมทั้งความหมายของมีดี้ โดยแบ่งการสร้างและการทดลองออกเป็น 2 ส่วนคือ ด้านฮาร์ดแวร์ สร้างวงจรตัวกลางมีดี้ และ ด้านซอฟท์แวร์ สร้างโปรแกรมใช้งาน

5.1 สรุปการทดลองการใช้งานวงจรตัวกลางมีดี้

วงจรตัวกลางมีดี้ที่สร้างขึ้นประกอบด้วยอุปกรณ์หลัก คือ Z-80 SIO ตัวนับเวลา 8253 ที่สามารถโปรแกรมได้ ตัวสร้างความถี่ ซึ่งทำหน้าที่ให้สัญญาณนาฬิกาแก่อุปกรณ์อื่นๆ และ ส่วนเชื่อมต่อรับและส่งข้อมูลกับเครื่องดนตรี ซึ่งวงจรสามารถทำงานเปลี่ยนระหว่างข้อมูลมีดี้และข้อมูลคอมพิวเตอร์ได้ตามที่ต้องการ ยกเว้นมีปัญหาเรื่องการนับเวลา ดังจะกล่าวต่อไป

5.2 สรุปการทดลองใช้งานโปรแกรมสอนดนตรี

สร้างโปรแกรมในลักษณะบทเรียน ที่แบ่งออกเป็นบทต่างๆ เพื่อให้สามารถเลือกเรียนได้และใช้ภาษาปาสคาลในการเขียน เนื่องจากแสดงเป็นรูปภาพ จึงเลือกใช้เครื่องไมโครคอมพิวเตอร์ซึ่งทำงานแสดงรูปภาพ หรือ ใช้งานในโหมดกราฟฟิก(Graphic Mode)ได้ เนื้อหาของบทเรียนเป็นการเรียนรู้ตัวโน้ตดนตรี และเรียนรู้การเล่นเครื่องดนตรีประเภทเปียโน ว่าแต่ละคีย์ของเครื่องดนตรีให้เสียงตัวโน้ตใด พร้อมทั้งมีตัวอย่างการเล่นเพลงง่ายๆด้วย

5.3 ปัญหาที่พบและการแก้ไข

1. โปรแกรมที่เขียนขึ้นไม่อาจใช้ได้กับเครื่องคอมพิวเตอร์ทุกเครื่อง เนื่องจากใช้ภาษาปาสคาลในโหมดกราฟฟิก ซึ่งใช้งานได้เฉพาะเครื่องที่มีตัวขับสัญญาณ (Driver) ชนิด CGA, MCGA, EGA, VGA, Herrcules, AT&T400Line และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3270PC เท่านั้น จึงต้องทำการตรวจสอบเสียก่อน

2. เนื่องจากตัวขับสัญญาณ แต่ละประเภทให้ขนาดจอภาพไม่เท่ากัน ทำให้ขนาดของรูปภาพที่ปรากฏบนจอภาพหนึ่ง อาจไม่พอดีเมื่อปรากฏบนอีกจอภาพหนึ่ง ซึ่งแก้ไขโดยเขียนโปรแกรมที่สามารถแสดงรูปภาพมีขนาดพอดีกับจอภาพทุกแบบ แต่อาจดูเล็กไปบ้างสำหรับจอภาพบางชนิด

3. โปรแกรมเหมาะกับการเล่นบนจอสี คำสั่ง Getimage ในภาษาปาสคาลใช้สำหรับการเก็บภาพบนหน้าจอไปไว้ในหน่วยความจำ จะต้องมีการจองเนื้อที่หน่วยความจำไว้ล่วงหน้า ซึ่งแต่ละครั้งมีขนาดไม่เกิน 64,000 บิท เมื่อใช้งานกับจอโมโนโครม (Monochrome) จะไม่มีปัญหา แต่สำหรับจอสีต้องใช้หน่วยความจำเพิ่มขึ้นในการเก็บค่าสี (Color Attribute) จึงต้องจองเนื้อที่เป็น 2 เท่าในการเก็บภาพแต่ละครั้ง

4. การบันทึกเพลงและเล่นเพลงให้ถูกต้องตามจังหวะ ต้องอาศัยการบันทึกและนับเวลาที่ถูกต้อง ซึ่งการทดลองพบปัญหาการนับเวลาผิดพลาดจากตัวนับเวลา 8253 อาจเป็นเพราะการรับสัญญาณจากตัวนับเวลา ไม่พอดีกับการเปลี่ยนแปลงลอจิกของสัญญาณ ทำให้ต้องใช้การนับเวลาวิธีอื่น

5.4 แนวทางในการพัฒนาต่อไป

เพิ่มความสามารถของโปรแกรม มีบทเรียนอื่นๆเพิ่มขึ้น และใช้กับตัวโน้ตหลายชนิดที่เกิดจากการเล่นอย่างซับซ้อนขึ้นได้ นอกจากนี้ ยังมีแนวทางประยุกต์ให้โปรแกรมหนึ่งสอนการเรียนรู้ตัวโน้ตกับเครื่องดนตรีหลายชนิดพร้อมกันได้ โดยนำเอาข่าวสารมีดี้ประเภทอื่นมาใช้ เช่น ข่าวสารแสดงโหมดการทำงานของชานแนล เป็นต้น และสามารถพัฒนาให้แต่งเพลงบนหน้าจอภาพได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appendix

WRITE REGISTERS

The Z80-SIO contains eight registers (WR0-WR7) in each channel that are programmed separately by the system program to configure the functional personality of the channels. With the exception of WR0, programming the write registers requires two bytes. The first byte contains three bits (D_0-D_2) that point to the selected register, the second byte is the actual control word that is written into the register to configure the Z80 SIO.

Note that the programmer has complete freedom, after pointing to the selected register, of either reading to test the read register or writing to initialize the write register. By designing software to initialize the Z80-SIO in a modular and structured fashion, the programmer can use powerful block I/O instructions.

WR0 is a special case in that all the basic commands (CMD_0-CMD_2) can be accessed with a single byte. Reset (internal or external) initializes the pointer bits (D_0-D_2) to point to WR0.

The basic commands (CMD_0-CMD_2) and the CRC controls (CRC_0, CRC_1) are contained in the first byte of any write register access. This maintains maximum flexibility and system control. Each channel contains the following control registers. These registers are addressed as commands (not data).

WRITE REGISTER 0

WR0 is the command register, however, it is also used for CRC reset codes and to point to the other registers.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CRC Reset Code 1	CRC Reset Code 0	CMD 2	CMD 1	CMD 0	PTR 2	PTR 1	PTR 0

Pointer Bits (D_0-D_2): Bits D_0-D_2 are pointer bits that determine which other write register the next byte is to be written into or which read register the next byte is to be read from. The first byte written into each channel after a reset (either by a Reset command or by the external reset input) goes into WR0. Following a read or write to any register (except WR0), the pointer will point to WR0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command Bits (D_3-D_5) Three bits D_3-D_5 are encoded to issue the seven basic Z80-SIO commands

COMMAND	CMD_2	CMD_1	CMD_0	
0	0	0	0	Null Command (no effect)
1	0	0	1	Send Abort (SDLC Mode)
2	0	1	0	Reset External/Status Interrupts
3	0	1	1	Channel Reset
4	1	0	0	Enable Interrupt on next Rx Character
5	1	0	1	Reset Transmitter Interrupt Pending
6	1	1	0	Error Reset (latches)
7	1	1	1	Return from Interrupt (Channel A

Command 0 (Null). The Null command has no effect. Its normal use is to cause the Z80-SIO to do nothing while the pointers are set for the following byte.

Command 1 (Send Abort). This command is used only with the SDLC mode to generate a sequence of eight to thirteen 1's.

Command 2 (Reset External/Status Interrupts). After an External/Status interrupt (a change on a modem line or a break condition, for example), the status bits of RR0 are latched. This command re-enables them and allows interrupts to occur again. Latching the status bits captures short pulses until the CPU has time to read the change.

Command 3 (Channel Reset). This command performs the same function as an External Reset, but only on a single channel. Channel A Reset also resets the interrupt prioritization logic. All control registers for the channel must be rewritten after a Channel Reset command.

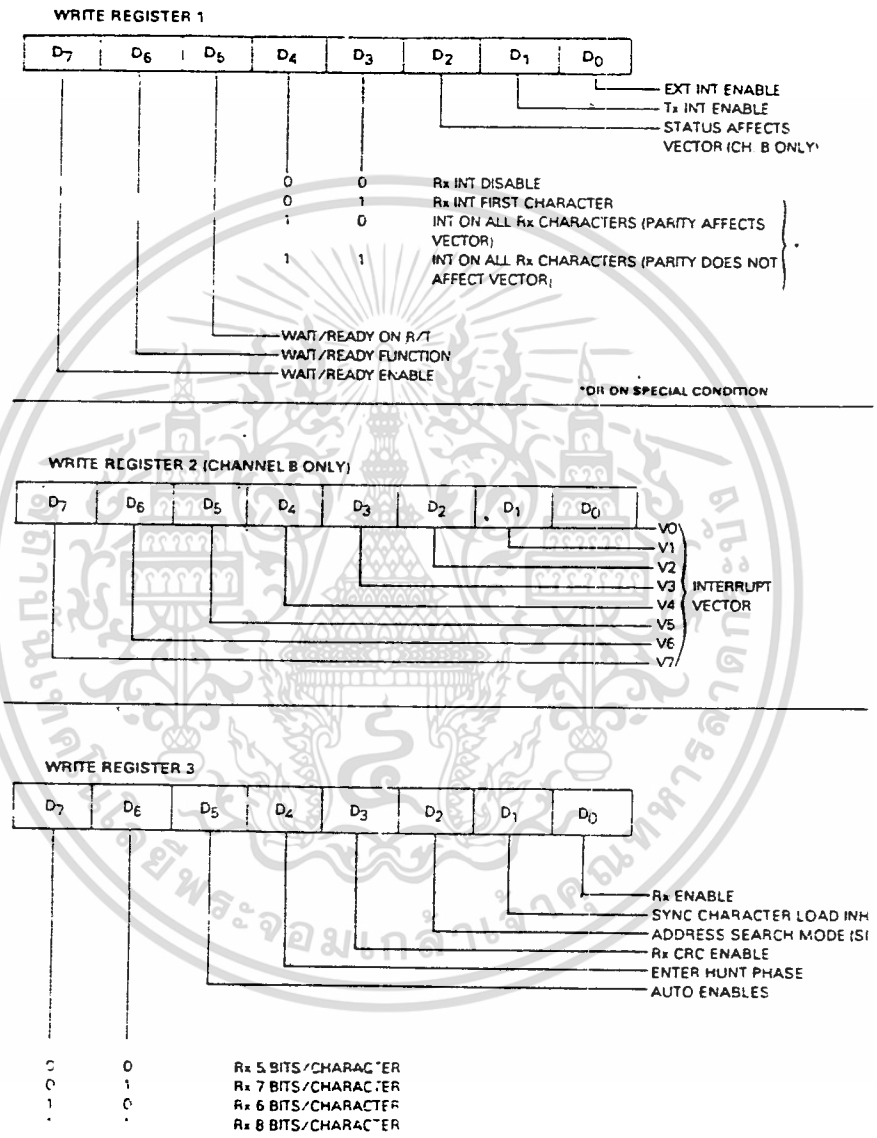
WRITE REGISTER BIT FUNCTIONS

WRITE REGISTER 0

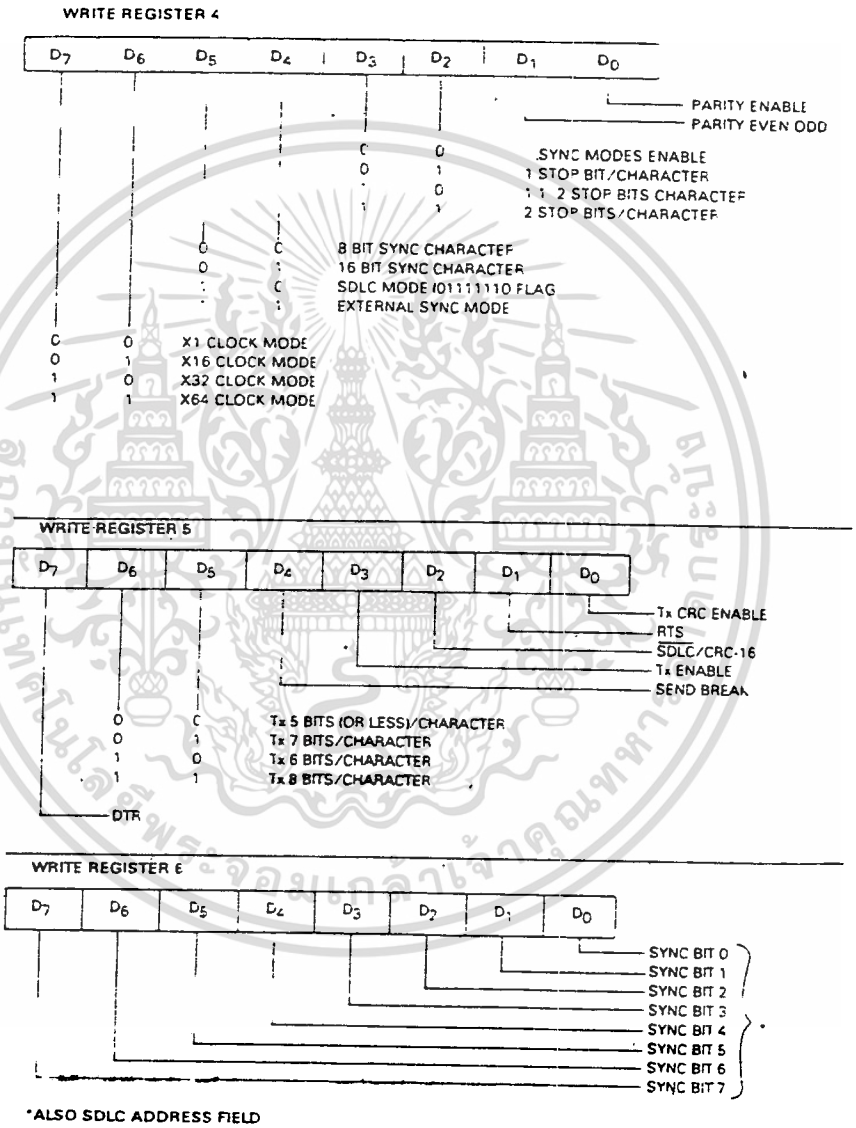
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
					0	0	0	REGISTER 0
					0	0	1	REGISTER 1
					0	1	0	REGISTER 2
					0	1	1	REGISTER 3
					1	0	0	REGISTER 4
					1	0	1	REGISTER 5
					1	1	0	REGISTER 6
					1	1	1	REGISTER 7
		0	0	0	NULL CODE			
		0	0	1	SEND ABORT (SDLC)			
		0	1	0	RESET EXT/STATUS INTERRUPTS			
		0	1	1	CHANNEL RESET			
		1	0	0	ENABLE INT ON NEXT Rx CHARACTER			
		1	0	1	RESET Tx INT PENDING			
		1	1	0	ERROR RESET			
		1	1	1	RETURN FROM INT (CH-A ONLY)			

0	0	NULL CODE
0	1	RESET Rx CRC CHECKER
1	0	RESET Tx CRC GENERATOR
1	1	RESET Tx UNDERRUN/EOM LATCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



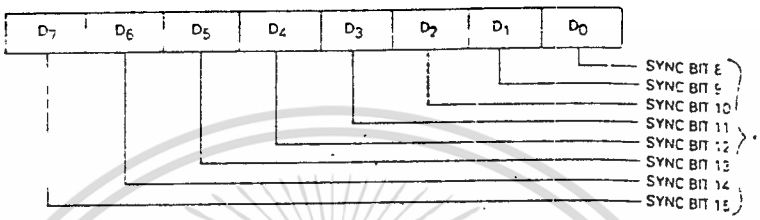
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*ALSO SDLC ADDRESS FIELD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE REGISTER 7



*FOR SDLC IT MUST BE PROGRAMMED TO "01111110" FOR FLAG RECOGNITION

After a Channel Reset, four extra system clock cycles should be allowed for Z80-SIO reset time before any additional commands or controls are written into that channel. This can normally be the time used by the CPU to fetch the next op code.

Command 4 (Enable Interrupt On Next Character). If the Interrupt On First Receive Character mode is selected, this command reactivates that mode after each complete message is received, to prepare the Z80-SIO for the next message.

Command 5 (Reset Transmitter Interrupt Pending). The transmitter interrupts when the transmit buffer becomes empty if the Transmit Interrupt Enable mode is selected. In those cases where there are no more characters to be sent (at the end of message, for example), issuing this command prevents further transmitter interrupts until after the next character has been loaded into the transmit buffer or until CRC has been completely sent.

Command 6 (Error Reset). This command resets the error latches. Parity and Overrun errors are latched in RR1 until they are reset with this command. With this scheme, parity errors occurring in block transfers can be examined at the end of the block.

Command 7 (Return From Interrupt). This command must be issued in Channel A and is interpreted by the Z80-SIO in exactly the same way it would interpret a RETI command on the data bus. It resets the interrupt-under-service latch of the highest-priority internal device under service and thus allows lower priority devices to interrupt via the daisy chain. This command allows use of the internal daisy chain even in systems with no external daisy chain or RETI command.

CRC Reset Codes 0 and 1 (D_6 and D_7). Together, these bits select one of the three following reset commands:

CRC Reset Code 1	CRC Reset Code 0	
0	0	Null Code (no effect)
0	1	Reset Receive CRC Checker
1	0	Reset Transmit CRC Generator
1	1	Reset Tx Underrun/End Of Message (EOM)

The Reset Transmit CRC Generator command normally initializes the CRC generator to all 0's. If the SDLC mode is selected, this command initializes the CRC generator to all 1's. The [illegible] CRC checker is also initialized to all 1's for the SDLC mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE REGISTER 1

WR1 contains the control bits for the various interrupt and Wait/Ready modes

D_7	D_6	D_5	D_4
Wait/Ready Enable	Wait Or Ready Function	Wait/Ready On Receive/Transmit	Receive Interrupt Mode 1
D_3	D_2	D_1	D_0
Receive Interrupt Mode 0	Status Affects Vector	Transmit Interrupt Enable	External Interrupts Enable

External/Status Interrupt Enable (D_0). The External/Status Interrupt Enable allows interrupts to occur as a result of transitions on the $\overline{DCD}$, $\overline{CTS}$ or $\overline{SYNC}$ inputs, as a result of a Break/Abort detection and termination, or at the beginning of CRC or sync character transmission when the Transmit Underrun/EOM latch becomes set

Transmitter Interrupt Enable (D_1). If enabled, the interrupts occur whenever the transmitter buffer becomes empty

Status Affects Vector (D_2). This bit is active in Channel B only. If this bit is not set, the fixed vector programmed in WR2 is returned from an interrupt acknowledge sequence. If this bit is set, the vector returned from an interrupt acknowledge is variable according to the following interrupt conditions

	V_3	V_2	V_1	
Ch B	0	0	0	Ch B Transmit Buffer Empty
	0	0	1	Ch B External/Status Change
	0	1	0	Ch B Receive Character Available
	0	1	1	Ch B Special Receive Condition*
Ch A	1	0	0	Ch A Transmit Buffer Empty
	1	0	1	Ch A External/Status Change
	1	1	0	Ch A Receive Character Available
	1	1	1	Ch A Special Receive Condition*

*Special Receive Conditions: Parity Error, Rx Overrun Error, Framing Error, End Of Frame (SDLC)

Receive Interrupt Modes 0 and 1 (D_3 and D_4). Together these two bits specify the various character-available conditions. In Receive Interrupt modes 1, 2 and 3 a Special Receive Condition can cause an interrupt and modify the interrupt vector

D_4 Receive Interrupt Mode 1	D_3 Receive Interrupt Mode 0		
0	0	0	Receive Interrupts Disabled
0	1	1	Receive Interrupt On First Character Only
1	0	2	Interrupt On All Receive Characters—parity error is a Special Receive condition
1	1	3	Interrupt On All Receive Characters—parity error is not a Special Receive condition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wait/Ready Function Selection (D_5-D_7): The Wait and Ready functions are selected by controlling D_5, D_6 and D_7. Wait/Ready function is enabled by setting Wait/Ready Enable (WR1, D_7) to 1. The Ready Function is selected by setting D_6 (Wait/Ready function) to 1. If this bit is 1, the $\overline{WAIT}$/$\overline{READY}$ output switches from High to Low when the Z80-SIO is ready to transfer data. The Wait function is selected by setting D_6 to 0. If this bit is 0, the $\overline{WAIT}$/$\overline{READY}$ output is in the open-drain state and goes Low when active.

Both the Wait and Ready functions can be used in either the Transmit or Receive modes, but not both simultaneously. If D_5 (Wait/Ready or Receive/Transmit) is set to 1, the Wait/Ready function responds to the condition of the receive buffer (empty or full). If D_5 is set to 0, the Wait/Ready function responds to the condition of the transmit buffer (empty or full).

The logic states of the $\overline{WAIT}$/$\overline{READY}$ output when active or inactive depend on the combination of modes selected. Following is a summary of these combinations:

	And D_6 = 1	If D_7 = 0	And D_6 = 0
	$\overline{READY}$ is High		$\overline{WAIT}$ is floating
	And D_6 = 0	If D_7 = 1	And D_5 = 1
$\overline{READY}$ / $\overline{WAIT}$	Is High when transmit buffer is full. Is Low when transmit buffer is full and an SIO data port is selected.	$\overline{READY}$ / $\overline{WAIT}$	Is High when receive buffer is empty. Is Low when receive buffer is empty and an SIO data port is selected.
$\overline{READY}$ / $\overline{WAIT}$	Is Low when transmit buffer is empty. Is floating when transmit buffer is empty.	$\overline{READY}$ / $\overline{WAIT}$	Is Low when receive buffer is full. Is Floating when receive buffer is full.

The $\overline{WAIT}$ output High-to-Low transition occurs when the delay time $t_{DI}C(WR)$ after the I/O request. The Low-to-High transition occurs with the delay $t_{DH}\Phi(WR)$ from the falling edge of Φ. The $\overline{READY}$ output High-to-Low transition occurs with the delay $t_{DL}\Phi(WR)$ from the rising edge of Φ. The $\overline{READY}$ output Low-to-High transition occurs with the delay $t_{D}IC(WR)$ after $\overline{IORQ}$ falls.

The Ready function can occur anytime the Z80-SIO is not selected. When the $\overline{READY}$ output becomes active (Low), the DMA controller issues $\overline{IORQ}$ and the corresponding B/$\overline{A}$ and C/$\overline{D}$ inputs to the Z80-SIO to transfer data. The $\overline{READY}$ output becomes inactive as soon as $\overline{IORQ}$ and $\overline{CS}$ become active. Since the Ready function can occur internally in the Z80-SIO whether it is addressed or not, the $\overline{READY}$ output becomes inactive when any CPU data or command transfer takes place. This does not cause problems because the DMA controller is not enabled when the CPU transfer takes place.

The Wait function—on the other hand—is active only if the CPU attempts to read Z80-SIO data that has not yet been received, which occurs frequently when block transfer instructions are used. The Wait function can also become active (under program control) if the CPU tries to write data while the transmit buffer is still full. The fact that the $\overline{WAIT}$ output for either channel can become active when the opposite channel is addressed (because the Z80-SIO is addressed) does not affect operation of software loops or block move instructions.

WRITE REGISTER 2

WR2 is the interrupt vector register. It exists in Channel B only. V_4-V_7 and V_0 are always returned exactly as written. V_1-V_3 are returned as written if the Status Affects Vector (WR1, D_2) control bit is 0. If this bit is 1, they are modified as explained in the previous section.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE REGISTER 3

WR3 contains receiver logic control bits and parameters

D_7	D_6	D_5	D_4
Receiver Bits/ Char 1	Receiver Bits/ Char 0	Auto Enables	Enter Hunt Phase
D_3	D_2	D_1	D_0
Receiver CRC Enable	Address Search Mode	Sync Char Load Inhibit	Receiver Enable

Receiver Enable (D_0). A 1 programmed into this bit allows receive operations to begin. This bit should be set only after all other receive parameters are set and receiver is completely initialized.

Sync Character Load Inhibit (D_1). Sync characters preceding the message (leading sync characters) are not loaded into the receive buffers if this option is selected. Because CRC calculations are not stopped by sync character stripping, this feature should be enabled only at the beginning of the message.

Address Search Mode (D_2). If SDLC is selected, setting this mode causes messages with addresses not matching the programmed address in WR6 or the global (11111111) address to be rejected. In other words, no receive interrupts can occur in the Address Search mode unless there is an address match.

Receiver CRC Enable (D_3). If this bit is set, CRC calculation starts (or restarts) at the beginning of the first character transferred from the receive shift register to the buffer stack, regardless of the number of characters in the stack. See "SDLC Receive CRC Checking" (SDLC Receive section) and "CRC Error Checking" (Synchronous Receive section) for details regarding when this bit should be set.

Enter Hunt Phase (D_4). The Z80-SIO automatically enters the Hunt phase after a reset; however, it can be re-entered if character synchronization is lost for any reason (Synchronous mode) or if the contents of an incoming message are not needed (SDLC mode). The Hunt phase is re-entered by writing a 1 into bit D_4. This sets the Sync/Hunt bit (D_4) in RR0.

Auto Enables (D_5). If this mode is selected, $\overline{DCD}$ and $\overline{CTS}$ become the receiver and transmitter enables, respectively. If this bit is not set, $\overline{DCD}$ and $\overline{CTS}$ are simply inputs to their corresponding status bits in RR0.

Receiver Bits/Character 1 and 0 (D_7 and D_6). Together, these bits determine the number of serial receive bits assembled to form a character. Both bits may be changed during the time that a character is being assembled, but they must be changed before the number of bits currently programmed is reached.

D_7	D_6	Bits/Character
0	0	5
0	1	7
1	0	6
1	1	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE REGISTER 4

WR4 contains the control bits that affect both the receiver and transmitter. In the transmit/receive initialization routine these bits should be set before issuing WR1, WR3, WR5, WR6 and WR7.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Clock Rate	Clock Rate	Sync Modes	Sync Modes	Stop Bits	Stop Bits	Parity Even/Odd	Parity
1	0	*	0	1	0		

Parity (D_0). If this bit is set, an additional bit position (in addition to those specified in the bits/character control) is added to transmitted data and is expected in receive data. In the async mode, the parity bit received is transferred to the CPU as part of the character unless 8 bits/character is selected.

Parity Even/$\overline{\text{Odd}}$ (D_1). If parity is specified, this bit determines whether it is sent and checked as even or odd (1 = even).

Stop Bits 0 and 1 (D_2 and D_3). These bits determine the number of stop bits added to each asynchronous character sent. The receiver always checks for one stop bit. A special mode (D_3, D_2 = 0, 0) indicates that a synchronous mode is to be selected.

D_3 Stop Bits 1	D_2 Stop Bits 0	
0	0	Sync modes
0	1	1 stop bit per character
1	0	1½ stop bits per character
1	1	2 stop bits per character

Sync Modes 0 and 1 (D_4 and D_5). These bits select the various options for synchronization.

Sync Mode 1	Sync Mode 0	
0	0	8 bit programmed sync
0	1	16-bit programmed sync
1	0	SDLC mode (01111110 flag pattern)
1	1	External Sync mode

Clock Rate 0 and 1 (D_6 and D_7). These bits specify the multiplier between the clock rate and data rates. For synchronous modes the x1 clock rate must be specified. Any rate may be used for asynchronous modes; however, the same rate must be used for both the receiver and transmitter. The system clock in all modes must be at least 5 times the data rate. If the x1 clock rate is used, synchronization must be accomplished externally.

Clock Rate 1	Clock Rate 0	
0	0	Data Rate x1 Clock Rate
0	1	Data Rate x16 Clock Rate
1	0	Data Rate x32 Clock Rate
1	1	Data Rate x64 Clock Rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE REGISTER 5

WR5 contains control bits that affect the operation of transmitter, with the exception of D_2, which affects the transmitter and receiver.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
DTR	Tx Bits/ Char 1	Tx Bits/ Char 0	Send Break	Tx Enable	CRC-16/ SDLC	RTS	Tx CRC Enable

Transmit CRC Enable (D_0). This bit determines if CRC is calculated on a particular transmit character. If it is set at the time the character is loaded from the transmit buffer into the transmit shift register, CRC is calculated on the character. CRC is not automatically sent unless this bit is set when the Transmit Underrun condition exists.

Request To Send (D_1). This is the control bit for the $\overline{RTS}$ pin. When the $\overline{RTS}$ bit is set, the $\overline{RTS}$ pin goes Low; when reset, $\overline{RTS}$ goes High. In the Asynchronous mode, $\overline{RTS}$ goes High only after all the bits of the character are transmitted and the transmitter buffer is empty. In Synchronous modes, the pin directly follows the state of the bit.

CRC-16/$\overline{SDLC}$ (D_2). This bit selects the CRC polynomial used by both the transmitter and receiver. When set, the CRC-16 polynomial ($X^{16} + X^{15} + X^2 + 1$) is used; when reset, the SDLC polynomial ($X^{16} + X^{12} + X^5 + 1$) is used. If the SDLC mode is selected, the CRC generator and checker are preset to all 1's and a special check sequence is used. The SDLC CRC polynomial must be selected when the SDLC mode is selected. If the SDLC mode is not selected, the CRC generator and checker are preset to all 0's (for both polynomials).

Transmit Enable (D_3). Data is not transmitted until this bit is set and the Transmit Data output is held marking. Data or sync characters in the process of being transmitted are completely sent if this bit is reset after transmission has started. If the transmitter is disabled during the transmission of a CRC character, sync or flag characters are sent instead of CRC.

Send Break (D_4). When set, this bit immediately forces the Transmit Data output to the spacing condition, regardless of any data being transmitted. When reset, TxD returns to marking.

Transmit Bits/Character 0 and 1 (D_5 and D_6). Together, D_6 and D_5 control the number of bits in each byte transferred to the transmit buffer.

D_6 Transmit Bits/ Character 1	D_5 Transmit Bits/ Character 0	Bits/Character
0	0	Five or less
0	1	7
1	0	6
1	1	8

Bits to be sent must be right justified, least-significant bits first. The Five Or Less mode allows transmission of one to five bits per character, however, the CPU should format the data character as shown in the following table.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	D	Sends one data bit
1	1	1	0	0	0	D	D	Sends two data bits
1	1	0	0	0	D	D	D	Sends three data bits
1	0	0	0	D	D	D	D	Sends four data bits
0	0	0	D	D	D	D	D	Sends five data bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Terminal Ready (D_7). This is the control bit for the $\overline{DTR}$ pin. When set, $\overline{DTR}$ is active (Low) when reset; $\overline{DTR}$ is inactive (high).

WRITE REGISTER 6

This register is programmed to contain the transmit sync character in the Monosync mode, the first eight bits of a 16-bit sync character in the Bisync mode or a transmit sync character in the External Sync mode. In the SDLC mode, it is programmed to contain the secondary address field used to compare against the address field of the SDLC frame.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Sync 7	Sync 6	Sync 5	Sync 4	Sync 3	Sync 2	Sync 1	Sync 0

WRITE REGISTER 7

This register is programmed to contain the receive sync character in the Monosync mode, a second byte (last eight bits) of a 16-bit sync character in the Bisync mode and a flag character (01111110) in the SDLC mode. WR7 is not used in the External Sync mode.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Sync 15	Sync 14	Sync 13	Sync 12	Sync 11	Sync 10	Sync 9	Sync 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

READ REGISTERS INTRODUCTION

The Z80 SIO contains three registers, RR0-RR2 (Figure 7.1) that can be read to obtain the status information for each channel (except for RR2-Channel B only). The status information includes error conditions, interrupt vector and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

READ REGISTER 0

This register contains the status of the receive and transmit buffers, the $\overline{DCD}$, $\overline{CTS}$ and $\overline{SYNC}$ inputs, the Transmit Underrun/EOM latch, and the Break/Abort latch.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Break Abort	Transmit Underrun/ EOM	$\overline{CTS}$	Sync/ Hunt	$\overline{DCD}$	Transmit Buffer Empty	Interrupt Pending (Ch. A only)	Receive Character Available

Receive Character Available (D_0). This bit is set when at least one character is available in the receive buffer; it is reset when the receive FIFO is completely empty.

Interrupt Pending (D_1). Any interrupting condition in the Z80-SIO causes this bit to be set; however, it is readable only in Channel A. This bit is mainly used in applications that do not have vectored interrupts available. During the interrupt service routine in these applications, this bit indicates if any interrupt conditions are present in all Z80-SIO. This eliminates the need for analyzing all the bits of RR0 in both Channels A and B. Bit D_1 is reset when all the interrupting conditions are satisfied. This bit is always 0 in Channel B.

Transmit Buffer Empty (D_2). This bit is set whenever the transmit buffer becomes empty, except when a CRC character is being sent in a synchronous or SDLC mode. The bit is reset when a character is loaded into the transmit buffer. This bit is in the set condition after a reset.

Data Carrier Detect (D_3). The DCD bit shows the inverted state of the $\overline{DCD}$ input at the time of the last change of any of the five External/Status bits (DCD, $\overline{CTS}$, Sync/Hunt, Break/Abort or Transmit Underrun/EOM). Any transition of the $\overline{DCD}$ input causes the DCD bit to be latched and causes an External/Status interrupt. To read the current state of the DCD bit, this bit must be read immediately following a Reset External/Status Interrupt command.

Sync/Hunt (D_4). Since this bit is controlled differently in the Asynchronous, Synchronous and SDLC modes, its operation is somewhat more complex than that of the other bits and therefore requires more explanation.

In Asynchronous modes, the operation of this bit is similar to the DCD status bit, except that Sync/Hunt shows the state of the $\overline{SYNC}$ input. Any High-to-Low transition on the $\overline{SYNC}$ pin sets this bit and causes an External/Status interrupt (if enabled). The Reset External/Status Interrupt command is issued to clear the interrupt. A Low-to-High transition clears this bit and sets the External/Status interrupt. When the External/Status interrupt is set by the change in state of any other input or condition, this bit shows the inverted state of $\overline{SYNC}$ pin at the time of the change. This bit must be read immediately following a Reset External/Status Interrupt command to read the current state of the $\overline{SYNC}$ input.

In the External Sync mode, the Sync/Hunt bit operates in a fashion similar to the Asynchronous mode except the Enter Hunt Mode control bit enables the external sync detection logic. When the External

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

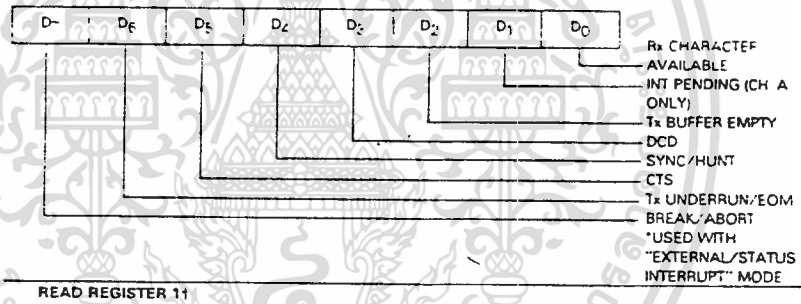
Sync Mode and Enter Hunt Mode bits are set (for example, when the receiver is enabled following a reset), the $\overline{SYNC}$ input must be held High by the external logic until external character synchronization is achieved. A High at the $\overline{SYNC}$ input holds the Sync/Hunt status bit in the reset condition.

When external synchronization is achieved, $\overline{SYNC}$ must be driven Low on the second rising edge of $\overline{RxC}$ on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive clock cycles to activate the $\overline{SYNC}$ input. Once $\overline{SYNC}$ is forced Low, it is a good practice to keep it Low until the CPU informs the external sync logic that synchronization has been lost or a new message is about to start. Refer to Figure 8-6 for timing details. The High-to-Low transition of the $\overline{SYNC}$ input sets the Sync/Hunt bit, which—in turn—sets the External/Status interrupt. The CPU must clear the interrupt by issuing the Reset External/Status interrupt command.

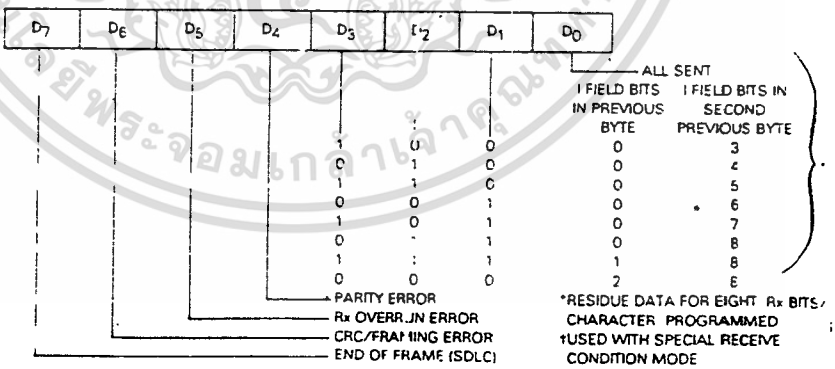
When the $\overline{SYNC}$ input goes High again, another External/Status interrupt is generated that must also be cleared. The Enter Hunt Mode control bit is set whenever character synchronization is lost or the end of message is detected. In this case, the Z80-SIO again looks for a High-to-Low transition on the $\overline{SYNC}$ input and the operation repeats as explained previously. This implies the CPU should also inform the external logic that character synchronization has been lost and that the Z80-SIO is waiting for $\overline{SYNC}$ to become active.

READ REGISTER BIT FUNCTIONS

READ REGISTER 0



READ REGISTER 1†

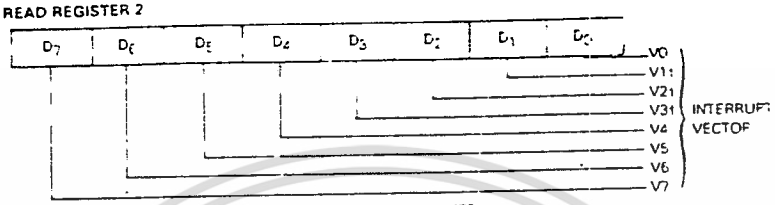


1 FIELD BITS IN PREVIOUS BYTE	1 FIELD BITS IN SECOND PREVIOUS BYTE
0	3
0	4
0	5
0	6
0	7
0	8
1	8
2	8

*RESIDUE DATA FOR EIGHT Rx BITS/CHARACTER PROGRAMMED
†USED WITH SPECIAL RECEIVE CONDITION MODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

READ REGISTER 2



†VARIABLE IF "STATUS AFFECTS VECTOR" IS PROGRAMMED

In the Monosync and Bisync Receive modes, the Sync/Hunt status bit is initially set to 1 by the Enter Hunt Mode bit. The Sync/Hunt bit is reset when the Z80-SIO establishes character synchronization. The High-to-Low transition of the Sync/Hunt bit causes an External/Status interrupt that must be cleared by the CPU issuing the Reset External/Status interrupt command. This enables the Z80-SIO to detect the next transition of other External/Status bits.

When the CPU detects the end of message or that character synchronization is lost, it sets the Enter Hunt Mode control bit, which—in turn—sets the Sync/Hunt bit to 1. The Low-to-High transition of the Sync/Hunt bit sets the External/Status interrupt, which must also be cleared by the Reset External/Status interrupt command. Note that the SYNC pin acts as an output in this mode and goes Low every time a sync pattern is detected in the data stream.

In the SDLC mode, the Sync/Hunt bit is initially set by the Enter Hunt mode bit or when the receiver is disabled. In any case, it is reset to 0 when the opening flag of the first frame is detected by the Z80-SIO. The External/Status interrupt is also generated and should be handled as discussed previously.

Unlike the Monosync and Bisync modes, once the Sync/Hunt bit is reset in the SDLC mode, it does not need to be set when the end of message is detected. The Z80-SIO automatically maintains synchronization. The only way the Sync/Hunt bit can be set again is by the Enter Hunt Mode bit or by disabling the receiver.

Clear to Send (D_5). This bit is similar to the DCD bit, except that it shows the inverted state of the $\overline{CTS}$ pin.

Transmit Underrun/End of Message (D_6). This bit is in a set condition following a reset (internal or external). The only command that can reset this bit is the Reset Transmit Underrun/EOM Latch command (WR0 D_6 and D_7). When the Transmit Underrun condition occurs, this bit is set, its becoming set causes the External/Status interrupt, which must be reset by issuing the "Reset External/Status Interrupt" command (WR0). This status bit plays an important role in conjunction with other control bits in controlling a transmit operation. Refer to "Bisync Transmit Underrun" and "SDLC Transmit Underrun" for additional details.

Break/Abort (D_7). In the Asynchronous Receive mode, this bit is set when a Break sequence (null character plus framing error) is detected in the data stream. The External/Status interrupt, if enabled, is set when Break is detected. The interrupt service routine must issue the Reset External/Status interrupt command (WR0, CMD_2) to the break detection logic so the Break sequence termination can be recognized.

The Break/Abort bit is reset when the termination of the Break sequence is detected in the incoming data stream. The termination of the Break sequence also causes the External/Status interrupt to be set. The Reset External/Status interrupt command must be issued to enable the break detection logic to look for the next Break sequence. A single extraneous null character is present in the receiver after the termination of a break. It should be read and discarded.

In the SDLC Receive mode, this bit is set by the detection of an Abort sequence (seven or more 1's). The External/Status interrupt is handled the same way as in the case of a Break. The Break/Abort bit is not used in the Synchronous Receive mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

READ REGISTER 1

This register contains the Special Receive condition status bits and Residue codes for the I-field in the SDLC Receive Mode

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
End of Frame (SDLC)	CRC/ Framing Error	Receiver Overrun Error	Parity Error	Residue Code 2	Residue Code 1	Residue Code 0	All Sent

All Sent (D_0). In Asynchronous modes, this bit is set when all the characters have completely cleared the transmitter. Transitions of this bit do not cause interrupts. The bit is always set in Synchronous modes.

Residue Codes 0, 1, and 2 (D_1-D_3). In those cases of the SDLC receive mode where the I-field is not an integral multiple of the character length, these three bits indicate the length of the I-field. These codes are meaningful only for the transfer in which the End Of Frame bit is set (WR1-C). For a receive character length of eight bits per character, the codes signify the following:

Residue Code 2	Residue Code 1	Residue Code 0	I-Field Bits In Previous Byte	I-Field Bits In Second Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

I-Field bits are right-justified in all cases.

If a receive character length different from eight bits is used for the I-field, a table similar to the previous one may be constructed for each different character length. For no residue (i.e., the last character boundary coincides with the boundary of the I-field and CRC field), the Residue codes are:

Bits per Character	Residue Code 2	Residue Code 1	Residue Code 0
8 Bits per Character	0	1	1
7 Bits per Character	0	0	0
6 Bits per Character	0	1	0
5 Bits per Character	0	0	1

Parity Error (D_4). When parity is enabled, this bit is set for those characters whose parity does not match the programmed sense (even/odd). The bit is latched, so once an error occurs it remains set until the Error Reset command (WR0) is given.

Receive Overrun Error (D_5). This bit indicates that more than three characters have been received without a read from the CPU. Only the character that has been written over is flagged with this error, but when this character is read, the error condition is latched until reset by the Error Reset command. If Status Affects Vector is enabled, the character that has been overrun interrupts with a Special Receive Condition vector.

CRC/Framing Error (D_6). If a Framing Error occurs (asynchronous modes), this bit is set (and not latched) for the receive character in which the Framing error occurred. Detection of a Framing Error adds an additional one-half of a bit time to the character time so the Framing error is not interpreted as a new start bit. In Synchronous and SDLC modes, this bit indicates the result of comparing the CRC checker to the appropriate check value. This bit is reset by issuing an Error Reset command. The bit is

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

not latched, so it is always updated when the next character is received. When used for CRC error and status in Synchronous modes, it is usually set since most bit combinations result in a non-zero CRC except for a correctly completed message.

End of Frame (D_7). This bit is used only with the SDLC mode and indicates that a valid ending flag has been received and that the CRC Error and Residue codes are also valid. This bit can be reset by issuing the Error Reset command. It is also updated by the first character of the following frame.

READ REGISTER 2 (Ch. B Only)

This register contains the interrupt vector written into WR2 if the Status Affects Vector control bit is not set. If the control bit is set, it contains the modified vector shown in the Status Affects Vector paragraph of the Write Register 1 section. When this register is read, the vector returned is modified by the highest priority interrupting condition at the time of the read. If no interrupts are pending, the vector is modified with V_3=0, V_2=1, and V_1=1. This register may be read only through Channel B.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0
				Variable if Status Affects Vector is enabled			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NEC

μPD8253-2
μPD8253-5

PROGRAMMABLE INTERVAL TIMER

DESCRIPTION

The NEC μPD8253 contains three independent, programmable, multi-modal 16 bit counter/timers. It is designed as a general purpose device, fully compatible with the 8080 family. The μPD8253 interfaces directly to the busses of the processor and an array of I/O ports.

The μPD8253 can generate accurate time delays under the control of system software. The three independent 16 bit counters can be clocked at rates from DC to 4 MHz. The system software controls the loading and starting of the counters to provide accurate multiple time delays. The counter output flags the processor at the completion of the time out cycles.

System overhead is greatly improved by relieving the software from the maintenance of timing loops. Some other common uses for the μPD8253 in microprocessor based systems are:

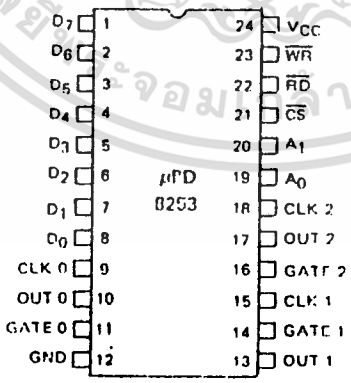
- Programmable Baud Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

FEATURES

- Three Independent 16 Bit Counters
- Clock Rate: DC to 4 MHz
- Count Binary or BCD
- Single +5 Volt Supply ±10%
- 24 Dual-In-Line Plastic Package

6

PIN CONFIGURATION



PIN NAMES

D_7-D_0	Data Bus (8 Bit)
CLK N	Counter Clock Inputs
GATE N	Counter Gate Inputs
OUT N	Counter Outputs
$\overline{RD}$	Read Counter
$\overline{WR}$	Write Command or Data
$\overline{CS}$	Chip Select
A_0, A_1	Counter Select
V_{CC}	+5 Volts
GND	Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

μPD8253

FUNCTIONAL DESCRIPTION

Data Bus Buffer

The 3-state, 8 bit, bi-directional Data Bus Buffer interfaces the μPD8253 to the 8080A/8085A microprocessor system. It will transmit or receive data in accordance with the INput or OUTput instructions executed by the processor. There are three basic functions of the Data Bus Buffer.

1. Program the modes of the μPD8253.
2. Load the count registers.
3. Read the count values.

Read/Write Logic

The Read/Write Logic controls the overall operation of the μPD8253 and is governed by inputs received from the processor system bus.

Control Word Register

Two bits from the address bus of the processor, A_0 and A_1, select the Control Word Register when both are at a logic "1" (active-high logic). When selected, the Control Word Register stores data from the Data Bus Buffer in a register. This data is then used to control:

1. The operational MODE of the counters.
2. The selection of BCD or Binary counting.
3. The loading of the count registers.

$\overline{RD}$ (Read)

This active-low signal instructs the μPD8253 to transmit the selected counter value to the processor.

$\overline{WR}$ (Write)

This active-low signal instructs the μPD8253 to receive MODE information or counter input data from the processor.

A_1, A_0

The A_1 and A_0 inputs are normally connected to the address bus of the processor. They control the one-of-three counter selection and address the control word register to select one of the six operational MODES.

$\overline{CS}$ (Chip Select)

The μPD8253 is enabled when an active low signal is applied to this input. Reading or writing from this device is inhibited when the chip is disabled. The counter operation, however, is not affected.

Counters #0, #1, #2

The three identical, 16 bit down counters are functionally independent allowing for separate MODE configuration and counting operation. They function as Binary or BCD counters with their gate, input and output line configuration determined by the operational MODE data stored in the Control Word Register. The system software overhead time can be reduced by allowing the control word to govern the loading of the count data.

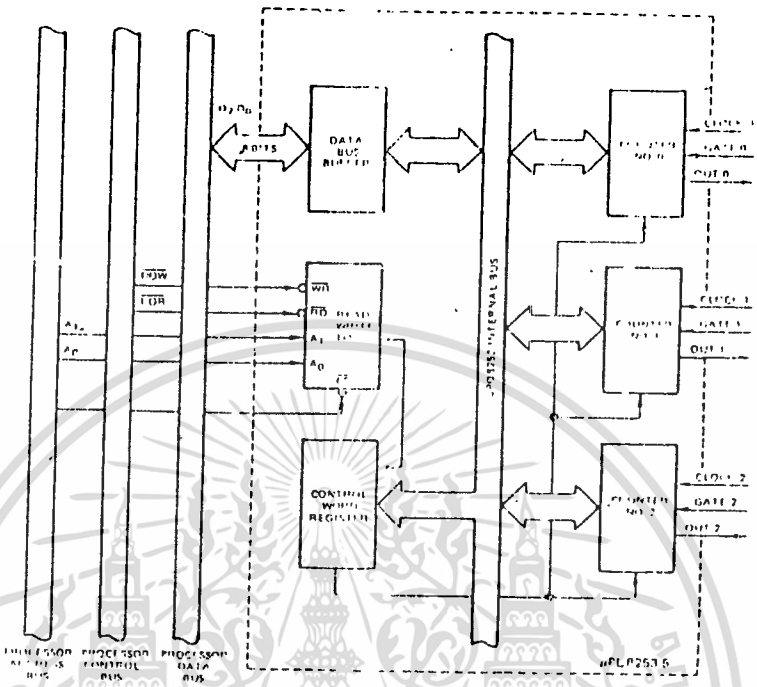
The programmer with READ operations, has access to each counter's contents. The μPD8253 contains the commands and logic to read each counter's contents while still counting without disturbing its operation.

The following is a table showing how the counters are manipulated by the input signals to the Read/Write Logic.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	FUNCTION
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation, 3-State
1	X	X	X	X	Disable, 3-State
0	1	1	X	X	No-Operation, 3-State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	0°C to +70°C
Storage Temperature	−65°C to +150°C
Voltage on Any Pin	0.5 to +7 Volts (1)

Note (1) With respect to ground.

T_a = 25°C

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

DC CHARACTERISTICS T_a = 0°C to +70°C; V_{CC} = +5V ± 10%

PARAMETER	SYMBOL	LIMITS MIN	TYP	MAX	UNIT	TEST CONDITIONS
Input Low Voltage	V_{IL}	-0.5		0.8	V	
Input High Voltage	V_{IH} (1)	2.0		V_{CC}+0.5	V	
Output Low Voltage	V_{OL}			0.45	V	I_{OL} = 2.2 mA
Output High Voltage	V_{OH}	2.4			V	I_{OH} = −400 μA
Input Load Current	I_{IL}			±10	μA	$0 \le V_{IN} \le V_{CC}$
Output Float Leakage Current	I_{OFL}			±10	μA	$0.45 \le V_{OUT} \le V_{CC}$
V_{CC} Supply Current	I_{CC}			140	mA	

Note (1) V_{IH} 2.2 min for 8253-2.

CAPACITANCE T_a = 25°C; V_{CC} = GND = 0V

PARAMETER	SYMBOL	LIMITS MIN	TYP	MAX	UNIT	TEST CONDITIONS
Input Capacitance	C_{IN}			10	pF	f_c = 1 MHz
Input/Output Capacitance	$C_{I/O}$			20	pF	Unmeasured pins returned to V_{SS}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

μPD8253

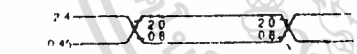
T_a = 0°C to +70°C; V_{CC} = +5V ± 10%; GND = 0V

AC CHARACTERISTICS①

PARAMETER	SYMBOL	LIMITS μPD8253-2		LIMITS μPD8253-5		UNIT	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
READ							
Address Stable Before $\overline{READ}$	t_{AR}	0		0		ns	
Address Hold Time for $\overline{READ}$	t_{RA}	0		0		ns	
$\overline{READ}$ Pulse Width	t_{RR}	200		250		ns	
Data Delay from $\overline{READ}$	t_{RD}		140		170	ns	CL = 150 pF
$\overline{READ}$ to Data Floating	t_{DF}	10	100	25	100	ns	CL = 150 pF
Recovery Time Between $\overline{READ}$	t_{RV}	200		1000		ns	
WRITE							
Address Stable Before $\overline{WRITE}$	t_{AW}	0		0		ns	
Address Hold Time for $\overline{WRITE}$	t_{WA}	20		0		ns	
$\overline{WRITE}$ Pulse Width	t_{WW}	200		250		ns	
Data Set Up Time for $\overline{WRITE}$	t_{DW}	150		150		ns	
Data Hold Time for $\overline{WRITE}$	t_{WD}	20		0		ns	
Recovery Time Between $\overline{WRITES}$	t_{RV}	200		1000		ns	
CLOCK AND GATE TIMING							
Clock Period	t_{CLK}	300		250	DC	ns	
High Pulse Width	t_{PWH}	100		160		ns	
Low Pulse Width	t_{PWL}	100		90		ns	
Gate Pulse Width High	t_{GW}	150		150		ns	
Gate Set Up Time to Clock ↑	t_{GS}	100		100		ns	
Gate Hold Time After Clock ↑	t_{GH}	100		50		ns	
Low Gate Width	t_{GL}	50		100		ns	
Output Delay from Clock ↓	t_{OD}		300		300	ns	CL = 150 pF
Output Delay from Gate	t_{ODG}		300		300	ns	CL = 150 pF

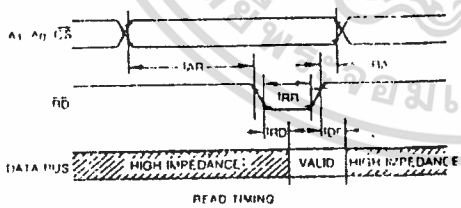
Note: ① AC Timing Measured at V_{OH} = 2.0V, V_{OL} = 0.8V

TIMING WAVEFORMS

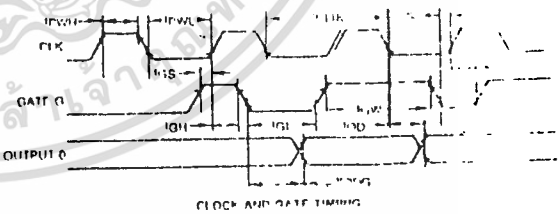


INPUT/OUTPUT WAVEFORM FOR AC TEST

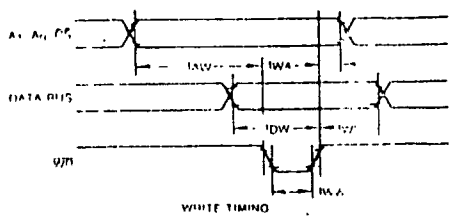
READ AND WRITE TIMING



READ TIMING



CLOCK AND GATE TIMING



WRITE TIMING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMING THE μPD8253

The programmer can select any of the six operational MODES for the counters using system software. Individual counter programming is accomplished by loading the CONTROL WORD REGISTER with the appropriate control word data (A_0, A_1 = 11).

CONTROL WORD FORMAT

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

SC – Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Invalid

RL – Read/Load

RL1	RL0	
0	0	Counter Latching Operation
1	0	Read/Load Most Significant Byte Only
0	1	Read/Load Least Significant Byte Only
1	1	Read/Load Least Significant Byte First, Then Most Significant Byte

BCD

0	Binary Counter, 16 Bits
1	BCD Counter, 4-Decades

M Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

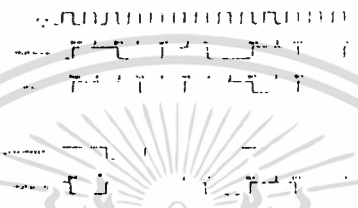
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATIONAL MODES (Cont.)

Mode 3: Square Wave Generator

MODE 3 resembles MODE 2 except the OUTPUT will be high for half of the count and low for the other half (for even values of data). For odd values of count data the OUTPUT will be high one clock cycle longer than when it is low (High Period = $\frac{N+1}{2}$ clock cycles; Low Period = $\frac{N-1}{2}$ clock periods, where N is the decimal value of count data). If the count register is reloaded with a new value during counting, the new value will be reflected immediately after the output transition of the current count.

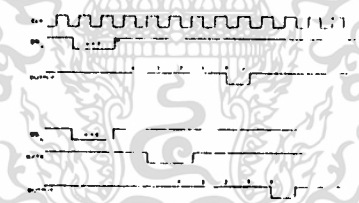
The OUTPUT will be held in the high state while GATE is asserted. Counting will start from the full count data after the GATE has been removed.



Mode 4: Software Triggered Strobe

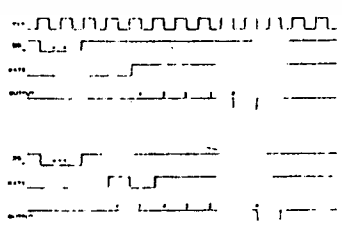
The OUTPUT goes high when MODE 4 is set, and counting begins after the second byte of data has been loaded. When the terminal count is reached, the OUTPUT will pulse low for one clock period. Changes in count data are reflected in the OUTPUT as soon as the new data has been loaded in to the count registers. During the loading of new data, the OUTPUT is held high and counting is inhibited.

The OUTPUT is held high for the duration of GATE. The counters are reset and counting begins from the full data value after GATE is removed.



Mode 5: Hardware Triggered Strobe

Loading MODE 5 sets OUTPUT high. Counting begins when count data is loaded and GATE goes high. After terminal count is reached, the OUTPUT will pulse low for one clock period. Subsequent trigger pulses will restart the counting sequence with the OUTPUT pulsing low on terminal count following the last rising edge of the trigger input (Reference bottom half of timing diagram).



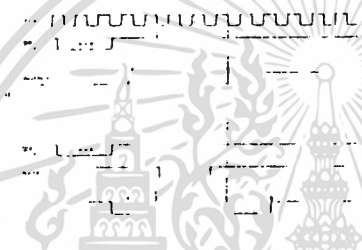
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATIONAL [illegible]

Each of the three counters can be individually programmed with different operating MODES by appropriately formatted Control Words. The following is a summary of the MODE operations

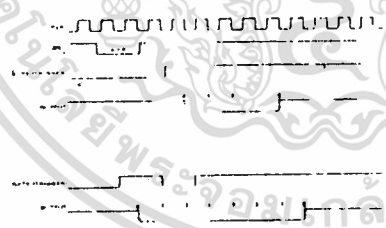
Mode 0: Interrupt on Terminal Count

The initial MODE set operation forces the OUTPUT low. When the specified counter is loaded with the count value, it will begin counting. The OUTPUT will remain low until the terminal count sets it high. It will remain in the high state until the trailing edge of the second $\overline{WR}$ pulse loads in COUNT data. If data is loaded during the counting process, the first $\overline{WR}$ stops the count. Counting starts with the new count data triggered by the falling clock edge after the second $\overline{WR}$. If a GATE pulse is asserted while counting, the count is terminated for the duration of GATE. The falling edge of CLK following the removal of GATE restarts counting from the terminated point.



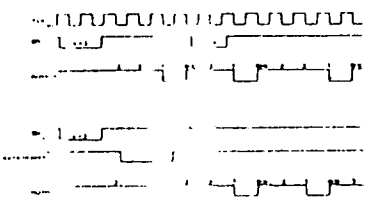
Mode 1: Programmable One-Shot

The OUTPUT is set low by the falling edge of CLOCK following the trailing edge of GATE. The OUTPUT is set high again at the terminal count. The output pulse is not affected if new count data is loaded while the OUTPUT is low. The new data will be loaded on the rising edge of the next trigger pulse. The assertion of a trigger pulse while OUTPUT is low, resets and retriggers the One-Shot. The OUTPUT will remain low for the full count value after the rising edge of TRIGGER.



Mode 2: Rate Generator

The RATE GENERATOR is a variable modulus counter. The OUTPUT goes low for one full CLOCK period as shown in following timing diagram. The count data sets the time between OUTPUT pulses. If the count register is reloaded between output pulses the present period will not be affected. The subsequent period will reflect the new value. The OUTPUT will remain high for the duration of the asserted GATE input. Normal operation resumes on the falling CLOCK edge following the rising edge of GATE



Note ① All internal counter events occur at the falling edge of the associated clock in all modes of operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES TIL111, TIL114, TIL116, TIL117
OPTOCOUPLERS

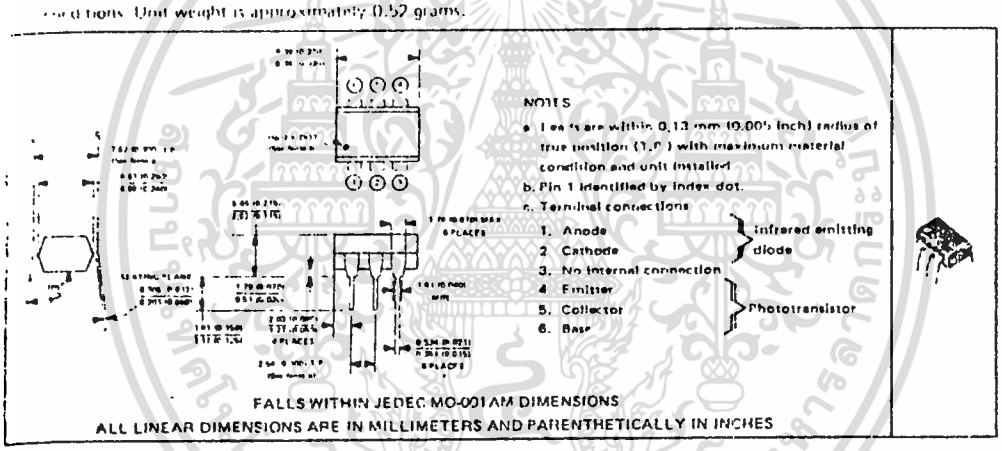
D1667, NOVEMBER 1973 - REVISED FEBRUARY 1983

COMPATIBLE WITH STANDARD TTL INTEGRATED CIRCUITS

- Gallium Arsenide Diode Infrared Source Optically Coupled to a Silicon N-P-N Phototransistor
- High Direct-Current Transfer Ratio
- High-Voltage Electrical Isolation . . . 1.5-kV or 2.5-kV Rating
- Plastic Dual-In-Line Package
- High-Speed Switching: t_r = 5 μs, t_f = 5 μs Typical

mechanical data

The package consists of a gallium arsenide infrared emitting diode and an n-p-n silicon phototransistor mounted on a lead frame encapsulated within an electrically nonconductive plastic compound. The case will withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high humidity conditions. Unit weight is approximately 0.52 grams.



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)

Input to Output Voltage: TIL111	±1.5 kV
TIL114, TIL116, TIL117	±2.5 kV
Collector Base Voltage	70 V
Collector Emitter Voltage (See Note 1)	30 V
Emitter Collector Voltage	7 V
Emitter Base Voltage	7 V
Input Diode Reverse Voltage	3 V
Input-Diode Continuous Forward Current at (or below) 25°C Free-Air Temperature (See Note 2)	100 mA
Continuous Power Dissipation at (or below) 25°C Free Air Temperature:	
Infrared Emitting Diode (See Note 3)	150 mW
Phototransistor (See Note 4)	150 mW
Total, Infrared-Emitting Diode plus Phototransistor (See Note 5)	250 mW
Storage Temperature Range	-55°C to 150°C
Lead Temperature 1,6 mm (1/16 Inch) from Case for 10 Seconds	260°C

NOTES: 1. This value applies when the base-emitter diode is open-circuited.
2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mA/°C.
3. Derate linearly to 100°C free-air temperature at the rate of 2 mW/°C.
4. Derate linearly to 100°C free-air temperature at the rate of 2 mW/°C.
5. Derate linearly to 100°C free-air temperature at the rate of 3.33 mW/°C.

Copyright 1983 by Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES TIL111, TIL114, TIL116, TIL117
OPTOCOUPLERS

electrical characteristics at 25°C free-air temperature

PARAMETER			TEST CONDITIONS	TIL111 TIL114			TIL116			TIL117		
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX
$V_{(BR)CBO}$	Collector Base Breakdown Voltage		I_C = 10 µA, I_E = 0, I_F = 0	70			70			70		
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage		I_C = 1 mA, I_B = 0, I_F = 0	30			30			30		
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage		I_E = 10 µA, I_C = 0, I_F = 0	7			7			7		
I_R	Input Diode Static Reverse Current		V_R = 3 V			10			10			
$I_{C(on)}$	On State Collector Current	Phototransistor Operation	V_{CE} = 0.4 V, I_F = 16 mA, I_B = 0	2	7							
			V_{CE} = 10 V, I_F = 10 mA, I_B = 0				2	5		5	9	
		Photodiode Operation	V_{CB} = 0.4 V, I_F = 16 mA, I_E = 0	7	20		7	20		7	20	
$I_{C(off)}$	Off-State Collector Current	Phototransistor Operation	V_{CE} = 10 V, I_F = 0, I_B = 0		1	50		1	50		1	
		Photodiode Operation	V_{CB} = 10 V, I_F = 0, I_E = 0		0.1	20		0.1	20		0.1	
h_{FE}	Transistor Static Forward Current Transfer Ratio		V_{CE} = 5 V, I_C = 10 mA, I_F = 0	100	300					200	600	
			V_{CE} = 5 V, I_C = 100 µA, I_F = 0				100	300				
V_F	Input Diode Static Forward Voltage		I_F = 16 mA		1.2	1.4					1.2	1.4
			I_F = 60 mA					1.2	1.5			
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage		I_C = 2 mA, I_F = 16 mA, I_B = 0		0.25	0.4						
			I_C = 2.2 mA, I_F = 15 mA, I_B = 0					0.25	0.4			
			I_C = 0.5 mA, I_F = 10 mA, I_B = 0									
r_{IO}	Input-to-Output Internal Resistance		V_{I-O} = ±1.5 kV for TIL111, ±2.5 kV for all others, See Note 6	10^{11}			10^{11}			10^{11}		
c_{IO}	Input-to-Output Capacitance		V_{I-O} = 0, f = 1 MHz, See Note 6		1	1.3		1	1.3		1	1.5

NOTE 6: These parameters are measured between both input diode leads shorted together and all the phototransistor leads shorted together.

switching characteristics at 25°C free-air temperature

PARAMETER			TEST CONDITIONS	TIL111 TIL114			TIL116			TIL117		
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX
t_r	Rise Time	Phototransistor Operation	V_{CC} = 10 V, $I_{C(on)}$ = 2 mA, R_L = 100 Ω, See Test Circuit A of Figure 1		5	10		5	10		5	
t_f	Fall Time				5	10		5	10		5	10
t_r	Rise Time	Photodiode Operation	V_{CC} = 10 V, $I_{C(on)}$ = 20 µA, R_L = 1 kΩ, See Test Circuit B of Figure 1		1			1			1	
t_f	Fall Time				1			1			1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

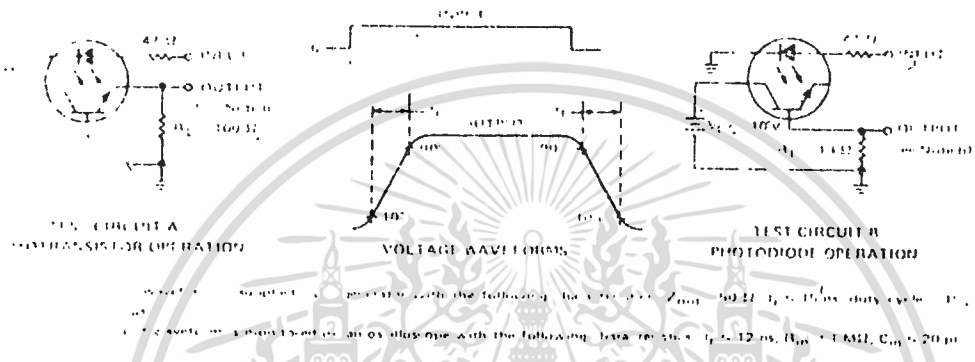


FIGURE 1–SWITCHING TIMES

TYPICAL CHARACTERISTICS

TIL111, TIL114
COLLECTOR CURRENT
vs
INPUT DIODE FORWARD CURRENT

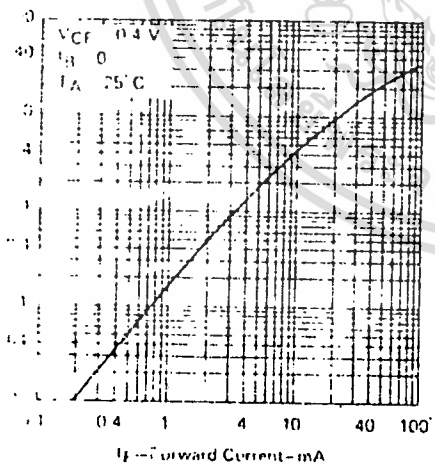


FIGURE 2

TIL116, TIL117
COLLECTOR CURRENT
vs
INPUT DIODE FORWARD CURRENT

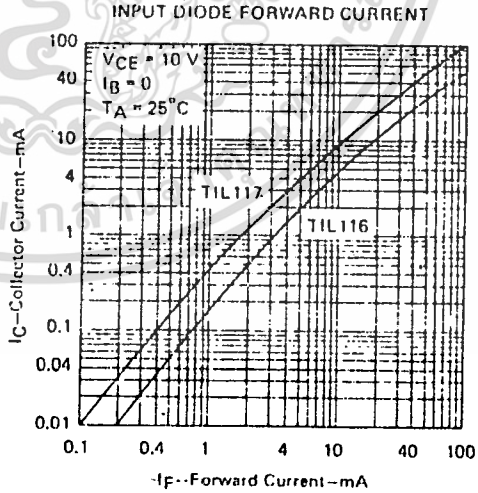


FIGURE 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

TIL111, TIL114

COLLECTOR CURRENT
vs
COLLECTOR-EMITTER VOLTAGE

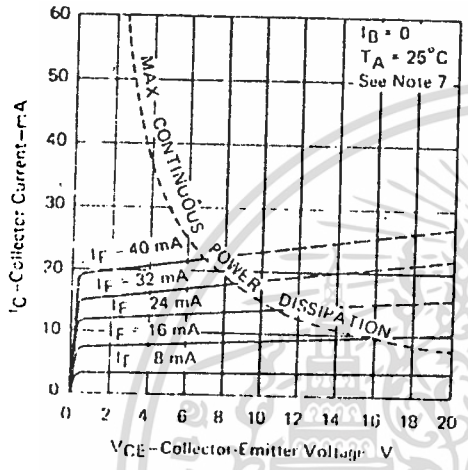


FIGURE 4

TIL116

COLLECTOR CURRENT
vs
COLLECTOR-EMITTER VOLTAGE

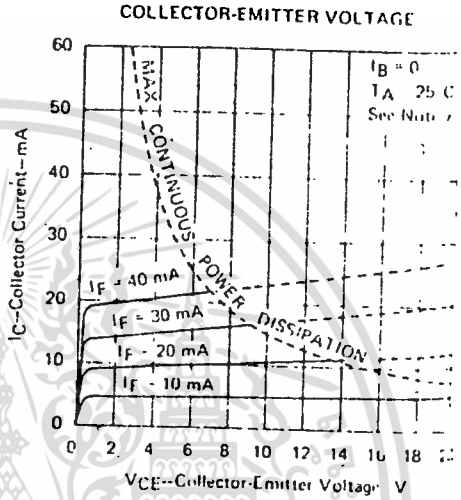


FIGURE 5

TIL117

COLLECTOR CURRENT
vs
COLLECTOR-EMITTER VOLTAGE

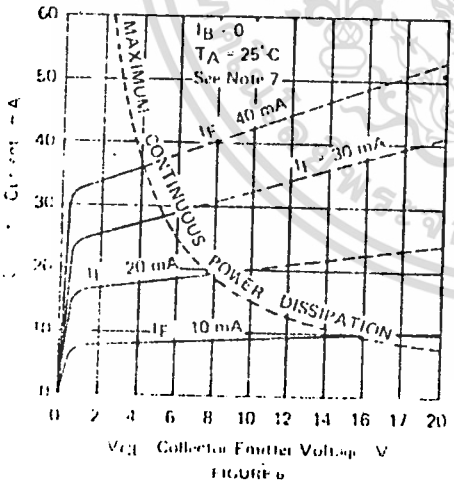


FIGURE 6

RELATIVE ON-STATE COLLECTOR CURRENT
vs
FREE-AIR TEMPERATURE

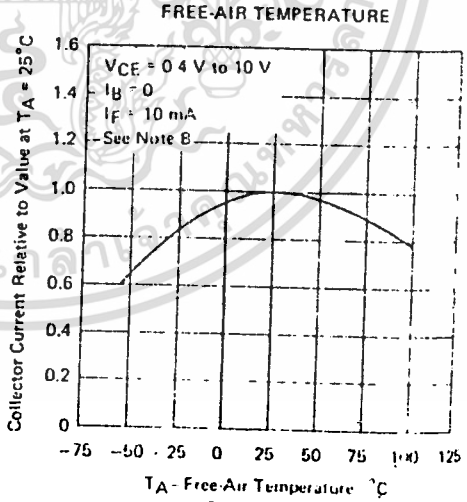


FIGURE 7

[illegible] operation of input diodes at currents beyond limits shown by dotted lines
[illegible] parameters were measured using pulse techniques. t_w = 1 ms, duty cycle ≤ 2%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

OFF-STATE COLLECTOR CURRENT
vs
FREE-AIR TEMPERATURE

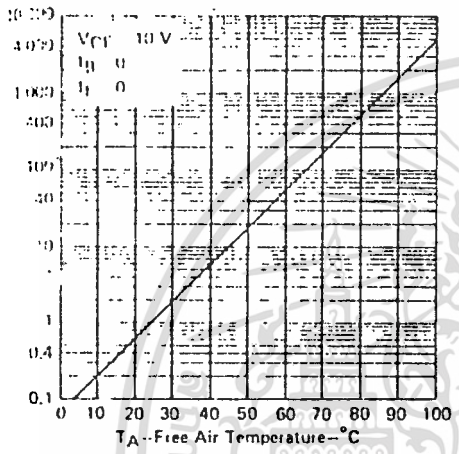


FIGURE 8

NORMALIZED TRANSISTOR STATIC FORWARD CURRENT TRANSFER RATIO
vs
ON-STATE COLLECTOR CURRENT

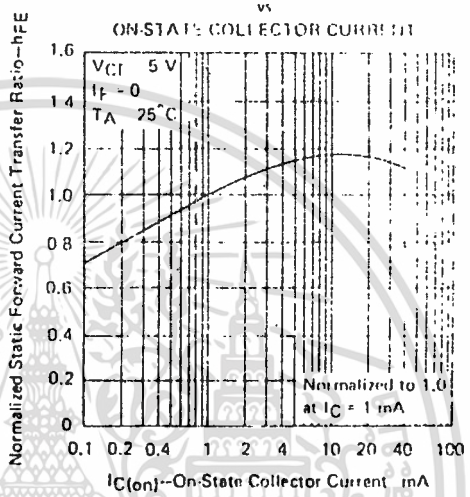


FIGURE 9

INPUT DIODE FORWARD
CONDUCTION CHARACTERISTICS

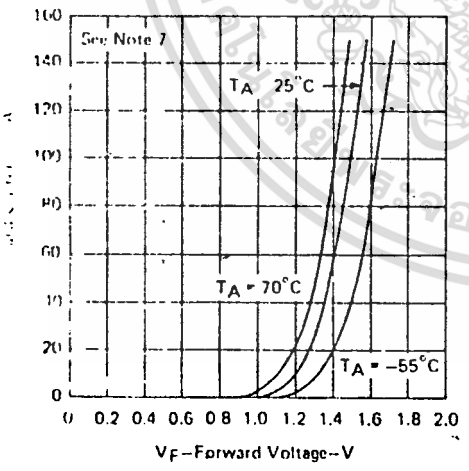


FIGURE 10

COLLECTOR CURRENT
vs
MODULATION FREQUENCY

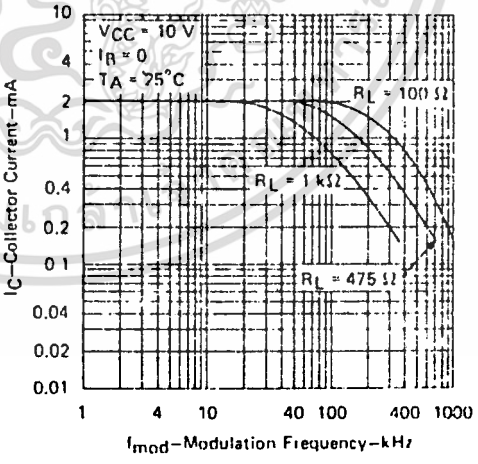


FIGURE 11

† These parameters were measured using pulse techniques, t_w = 1 ms, duty cycle < 2%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model CT-640 MIDI Implementation Chart

Version : 1

Function...		Transmitted	Recognized	Remarks
Basic	Default	1	1–4	(*1)
Channel	Changed	×	×	
Mode	Default	Mode 3	Mode 3	
	Messages	×	×	
	Altered	**********	×	
Note Number:	True voice	36–96 **********	0–127 36–96	0–11, 12–23, 24–35 = 36–47, 97–108, 109–120, 121–127 = 85–96 (*2)
Velocity	Note ON	× 9n v = 64	× 9n v = 1–127→64	(*2)
	Note OFF	× 9n v = 0	× 9n v = 0,8n v = xx	xx...No function
After Touch	Key's	×	×	
	Ch's	×	×	
Pitch Bender		×	×	
Control Change	64	O	O	Sustain pedal (*2)
Prog Change:	True#	O 0–29 **********	O 0–29	(*3)
System Exclusive		×	×	
System Common	: Song Pos	×	×	
	: Song Sel	×	×	
	: Tune	×	×	
System Real Time	: Clock	O	×	Continue not sent (*3)
	: Command	O	×	
Aux Mes-sages	: Local ON/OFF	×	×	
	: All Notes OFF	×	×	
	: Active Sense	×	×	
	: Reset	×	×	
Notes		MIDI messages transmitted/received only when set to the MIDI mode. *1) Multi messages received on CH 1–4 *2) Not received on CH-4 *3) Transmission/reception of 0–19 on CH-4		

Mode 1: OMNI ON, POLY
Mode 2: OMNI ON, MONO
Mode 3: OMNI OFF, POLY
Mode 4: OMNI OFF, MONO

O: Yes
×: No

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARD JUNE 88

TABLE I

SUMMARY OF STATUS BYTES

STATUS D7—D0	# OF DATA BYTES	DESCRIPTION
Channel Voice Messages		
1000nnnn	2	Note Off event
1001nnnn	2	Note On event (velocity=0 Note Off)
1010nnnn	2	Polyphonic key pressure/aftertouch
1011nnnn	2	Control change
1100nnnn	1	Program change
1101nnnn	1	Channel Pressure (Aftertouch)
1110nnnn	2	Pitch wheel change
CHANNEL MODE MESSAGES		
1011nnnn	2	Selects Channel Mode
SYSTEM MESSAGES		
11110000	******	System Exclusive
11110sss	0 to 2	System Common
11111ttt	0	System Real Time

NOTES:

nnnn: N−1, where N = Channel #.
i.e. 0000 is Channel 1.
0001 is Channel 2.
.
.
.
1111 is Channel 16.

*****: 0iiiiiii, data, . . . , EOX
iiiiiii: Identification
sss: 1 to 7
ttt: 0 to 7

TABLE II

CHANNEL VOICE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
1000nnnn	0kkkkkkk	Note Off (see notes 1-4)
	0vvvvvvv	vvvvvvv: note off velocity
1001nnnn	0kkkkkkk	Note On (see Notes 1-4)
	0vvvvvvv	vvvvvvv≠0: velocity vvvvvvv=0: note off
1010nnnn	0kkkkkkk	Polyphonic Key Pressure (After-Touch)
	0vvvvvvv	vvvvvvv: pressure value
1011nnnn	0ccccccc	Control Change
	0vvvvvvv	ccccccc: control # (0-121) (see notes 5-8) vvvvvvv: control value ccccccc=122 thru 127: Reserved. See Table III.
1100nnnn	0ppppppp	Program Change ppppppp: program number (0-127)
1101nnnn	0vvvvvvv	Channel Pressure (After-Touch) vvvvvvv: pressure value
1110nnnn	0vvvvvvv	Pitch Wheel Change LSB (see note 10)
	0vvvvvvv	Pitch Wheel Change MSB

Notes for Table II on the following page.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES:

1. nnnn: Voice Channel # (1–16, coded as defined in Table I notes)
2. kkkkkkk: note # (0–127)
   kkkkkkk=60: Middle C of keyboard

0	12	24	36	48	60	72	84	96	108	120	127
		c	c	c	c	c	c	c	c		
		piano range									

3. vvvvvvv: key velocity
   A logarithmic scale would be advisable.

0	1			64				127
off	ppp	pp	p	mp	mf	f	ff	fff

   vvvvvvv=64: in case of no velocity sensors
   vvvvvvv=0: Note Off, with velocity=64
4. Any Note On message sent should be balanced by sending a Note Off message for that note in that channel at some later time
5. ccccccc: control number

ccccccc	Description
0	Continuous Controller 0 MSB
1	Continuous Controller 1 MSB (MODULATION WHEEL)
2	Continuous Controller 2 MSB
3	Continuous Controller 3 MSB
4–31	Continuous Controller 4–31 MSB
32	Continuous Controller 0 LSB
33	Continuous Controller 1 LSB (MODULATION WHEEL)
34	Continuous Controller 2 LSB
35	Continuous Controller 3 LSB
36–63	Continuous Controller 4–31 LSB
64–95	Switches (On/Off)
96–121	Undefined
122–127	Reserved for Channel Mod message (see Table III).

6. The controllers are not specifically defined. A manufacturer can assign the logical controllers to physical ones as necessary. The controller allocation table must be provided in the user's operation manual.
7. Continuous controllers are divided into Most Significant and Least Significant Bytes. If only seven bits of resolution are needed for any particular controllers, only the MSB is sent. It is not necessary to send the LSB. If more resolution is needed, then both are sent, first the MSB, then the LSB. If only the LSB has changed in value, the LSB may be sent without re-sending the MSB.
8. vvvvvvv: control value (MSB)
   (for controllers)

0	127
min	max

   (for switches)

0	127
off	on

   Numbers 1 through 126, inclusive, are ignored.
9. Any messages (e.g. Note On), which are sent successively under the same status, can be sent without a Status byte until a different Status byte is needed.
10. Sensitivity of the pitch bender is selected in the receiver. Center position value (no pitch change) is 2000H, which would be transmitted EnH-00H-40H.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE III
CHANNEL MODE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
1011nnnn	0ccccccc 0vvvvvvv	Mode Messages
		ccccccc=122: Local Control vvvvvvv=0, Local Control Off vvvvvvv=127, Local Control On
		ccccccc=123: All Notes Off vvvvvvv=0
		ccccccc=124: Omni Mode Off (All Notes Off) vvvvvvv=0
		ccccccc=125: Omni Mode On (All Notes Off) vvvvvvv=0
		ccccccc=126: Mono Mode On (Poly Mode Off) (All Notes Off) vvvvvvv=M, where M is the number of channels vvvvvvv=0, the number of channels equals the number of voices in the receiver.
		ccccccc=127: Poly Mode On (Mono Mode Off) vvvvvvv=0 (All Notes Off)

OTES:

nnnn: Basic Channel # (1-16, coded as defined in Table I)
Messages 123 thru 127 function as All Notes Off messages. They will turn off all voices controlled by the ssigned Basic Channel. Except for message 123, All otes Off, they should not be sent periodically, but only a specific purpose. In no case should they be used in of Note Off commands to turn off notes which have previously turned on. Therefore any All Notes Off mand (123-127) may be ignored by receiver with no bility of notes staying on, since any Note On command must have a corresponding specific Note Off command.

Control Change #122, Local Control, is optionally used to interrupt the internal control path between the keyboard, for example, and the sound-generating circuitry. If 0 (Local Off message) is received, the path is disconnected: the keyboard data goes only to MIDI and the sound-generating circuitry is controlled only by incoming MIDI data. If a 7FH (Local On message) is received, normal operation is restored.

4. The third byte of 'Mono' specifies the number of channels in which Monophonic Voice messages are to be sent. This number, "M," is a number between 1 and 16. The channel(s) being used, then, will be the current Basic Channel (=N) thru N+M−1 up to a maximum of 16. If M=0, this is a special case directing the receiver to assign all its voices, one per channel, from the Basic Channel N through 16.

TABLE IV
SYSTEM COMMON MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11110001		Undefined
11110010	0lllllll 0hhhhhhh	Song Position Pointer lllllll: (Least significant) hhhhhhh: (Most significant)
11110011	0sssssss	Song Select sssssss: Song #
11110100		Undefined
11110101		Undefined
11110110	none	Tune Request
11110111	none	EOX: "End of System Exclusive" flag

ES:

Song Position Pointer: Is an internal register which holds the number of MIDI beats (1 beat = 6 MIDI clocks) since the start of the song. Normally it is set to 0 when the START switch is pressed, which starts sequence playback. It then increments with every sixth MIDI clock receipt, until STOP is pressed. If CONTINUE is pressed, it continues to increment. It can be arbitrarily preset (to a resolution of 1 beat) by the SONG POSITION POINTER message.

2. Song Select: Specifies which song or sequence is to be played upon receipt of a Start (Real-Time) message.
3. Tune Request: Used with analog synthesizers to request them to tune their oscillators.
4. EOX: Used as a flag to indicate the end of a System Exclusive transmission (see Table VI).

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามการใช้งานเพื่อการค้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE V

SYSTEM REAL TIME MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11111000		Timing Clock
11111001		Undefined
11111010		Start
11111011		Continue
11111100		Stop
11111101		Undefined
11111110		Active Sensing
11111111		System Reset

NOTES:

1. The System Real Time messages are for synchronizing all of the system in real time.
2. The System Real Time messages can be sent at any time. Any messages which consist of two or more bytes may be split to insert Real Time messages.
3. Timing Clock (F8H)
   The system is synchronized with this clock, which is sent at a rate of 24 clocks/quarter note.
4. Start (from beginning of song) (FAH)
   This byte is immediately sent when the PLAY switch on the master (e.g. sequencer or rhythm unit) is pressed.
5. Continue (FBH)
   This is sent when the CONTINUE switch is hit. A sequence will continue at the time of the next clock.
6. Stop (FCH)
   This byte is immediately sent when the STOP switch is hit. It will stop the sequence.
7. Active Sensing (FEH)
   Use of this message is optional, for either receivers or transmitters. This is a "dummy" Status byte that is sent every 300 ms (max), whenever there is no other activity on MIDI. The receiver will operate normally if it never receives FEH. Otherwise, if FEH is ever received, it will expect to receive FEH or a transmission of any type every 300 ms (max). If a period of 300 ms passes with no activity, the receiver will turn off the voices and return to normal operation.
8. System Reset (FFH)
   This message initializes all of the system to the condition of just having turned on power. The System Reset message should be used sparingly, preferably under manual command only. In particular, it should not be sent automatically on power up.

TABLE VI

SYSTEM EXCLUSIVE MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11110000	0iiiiiii	Bulk dump etc iiiiiii: identification
	(0*******)	Any number of bytes may be sent here, for any purpose, as long as they all have a zero in the most isgnificant bit.
	11110111	EOX: "End of System Exclusive"

NOTES:

1. iiiiiii: identification ID (0-127)
2. All bytes between the System Exclusive Status byte and EOX of the next Status byte must have zeroes in the MSB.
3. The ID number can be obtained from the MIDI committee. See Table VII.
4. In no case should other Status or Data bytes (except Real-Time) be interleaved with System Exclusive, regardless of whether or not the ID code is recognized.
5. EOX or any other Status byte, except Real-Time, will terminate a System Exclusive message, and should be sent immediately at its conclusion.

TABLE VII

MANUFACTURERS' ID NUMBERS

Sequential Circuits, Inc.	01H
Big Briar	02H
Octave/Plateau	03H
Moog Music	04H
Passport Designs	05H
Lexicon	06H
Oberheim	10H
Bon Tempi	20H
S.I.E.L.	21H
Kawai	40H
Roland	41H
Korg	42H
Yamaha	43H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณ

-ผศ. ครรชิต ไมตรี ผู้เป็นอาจารย์ที่ปรึกษาและเอื้อเฟื้ออุปกรณ์ตลอดจนสถานที่ ตลอดระยะเวลา 1 ปีของการทำโครงงาน

-คุณ อาจ วิเชียรเจริญ ผู้เชี่ยวชาญทางคอมพิวเตอร์ แผนกประมวลผลข้อมูล แห่งสหประชาชาติ, คุณ ศุภชัย แห่งบริษัท วรัญดา, คุณ วิศิษฐ์ แห่งโรงเรียนดนตรีสยามกลการ สาขาปทุมวัน และ นักดนตรีวงบัตเตอร์ฟลายที่ช่วยให้ข้อมูลเกี่ยวกับมีดี้ตลอดจนเอกสารต่างๆ

-คุณ วรปัญญา ปฐมกสิกุล และ คุณ ภาคภูมิ เสตะรัต ที่ช่วยให้คำปรึกษาเกี่ยวกับปัญหาของวงจร

-คุณ กิจจา นาคใหม่ ที่กรุณาเอื้อเฟื้อให้ยืมเครื่องดนตรี

-คุณ ริลดา ติรณสวัสดิ์ และคุณ มีโชค อมรพัฒนกุล(Computer MusicII) ที่ให้ความร่วมมือและช่วยเหลือต่างๆ ตลอดระยะเวลา 1 ปี

นอกจากนี้ ยังขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจและคอยรับฟังปัญหาต่างๆ ทำให้โครงงานนี้สำเร็จลงตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ซีเอ็ดยูเคชั่น , "คู่มือการใช้ไอซีซัพพอร์ทและหน่วยความจำ" , บริษัท ซีเอ็ดยูเคชั่น จำกัด , 605 หน้า , 2529

2. ซีเอ็ดยูเคชั่น , "คู่มือ/เทียบเบอร์ไอซี TTL" , บริษัท ซีเอ็ดยูเคชั่น จำกัด , 398 หน้า , 2529

3. ศิวพงษ์ ตั้งสุจริต , "มีดี้ : ดนตรีผสานคอมพิวเตอร์" , วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิกส์ , ฉบับที่ 90 , 2532 , หน้า 203 - 211

4. Roger Powell , "The Challenge of Music Software" , BYTE , vol.11 , No.6 , 1986 , pp.145 - 150

5. Jay Kubicky , "A MIDI Project" , BYTE , vol.11 , No.6 , 1986 , pp.199 - 208

6. James W. Coffron , "Z-80 Application" , SYBEX Inc. , 295 p., 1983

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้