ปีการศึกษา 2533

ยูนิเวอร์แซล I/O พอร์ท

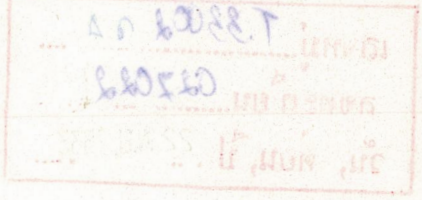
โดย

นายวรเดช คุณานพรัตน์ เลขประจำตัว 313312

นายวุฒพงศ์ จิรพรเจริญ เลขประจำตัว 313314

อาจารย์ที่ปรึกษา

ดร.ไพศาล นาคพิพัฒน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2533
ภาควิชา เทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ยูนิเวอร์แซล I/O พอร์ท

ผู้จัดทำ

1 นายวรเดช คุณานพรัตน์ เลขประจำตัว 313312
2 นายวุฒพงศ์ จิรพรเจริญ เลขประจำตัว 313314

_____อาจารย์ที่ปรึกษา
(ดร.ไพศาล นาคพิพัฒน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการเสนอวิธี การออกแบบและสร้างวงจรที่ใช้ในการควบคุมอุปกรณ์ภายนอก โดยใช้คอมพิวเตอร์พีซี ควบคุมการทำงานของระบบ โดยสามารถที่จะทำการควบคุมอุปกรณ์ภายนอกได้ 32 CH สามารถที่จะกำเนิดแรงดันคงที่ได้ 0 V ถึง 10 V และทำหน้าที่กำเนิดความถี่ซายน์เวฟได้ อีกทั้งยังมีตัว Timer/Counter เพื่อที่จะศึกษาการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABTRACT

THE OBJECTION OF THIS THESIS IS TO INTRODUCE COPUTER IS PROCESS , 'DESIGNING AND MAKING UP ELECTRONICS'S CIRCUITRY FOR EXTERNAL CONTROL BY USE PERSONEL COMPUTER . THIS SYSTEM IS ABLE TO CONTROL 32 CH EXTERNAL CHANNEL , GENERATING REGULATED VOLTAGE OV - 10V , AND SOURCE FOR SINUSOILDAL WAVE PLUS PROVIDING TIMER/COUNTER FOR OBSERVINSG SYSTEM 'S OPERATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

ABTRACT

บทที่

1. บทนำ 1
   - 1.1 ความเป็นมาและความสำคัญของปัญหา 1
   - 1.2 วัตถุประสงค์ของโครงงาน 1
   - 1.3 ขอบเขตของโครงงาน 2
   - 1.4 วิธีการดำเนินงาน 2

2. ทฤษฎีและหลักการ 4
   - 2.1 ทฤษฎีที่เกี่ยวข้องกับเครื่องวิเคราะห์วงจรแอนะลอก 4
     - 2.1.1 Slot บน IBM 5
     - 2.1.2 Input/Output MAP ของ IBM 9
     - 2.1.3 การใช้งาน 8255 11
     - 2.1.4 ความรู้เกี่ยวกับ IC PAL 17
     - 2.1.5 Digital To Analog And Analog To Digital 20
     - 2.1.6 การใช้งาน IC 8253 28
   - 2.2 หลักการของเครื่องวิเคราะห์วงจรแอนะลอก 36

3. การออกแบบวงจร 40
   - 3.1 ส่วนควบคุมการสร้างความถี่ 41
   - 3.2 ส่วนควบคุมการติดต่อกับ IBM PC 48
   - 3.3 ส่วนการติดต่อกับอุปกรณ์ภายนอก 54
   - 3.4 ส่วนการแปลงสัญญาณ DAC 55
   - 3.5 การออกแบบส่วนควบคุมอุปกรณ์ภายนอก 58
   - 3.6 ส่วนของแหล่งจ่ายไฟ 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทดลองและผลการทดลอง 61
Program DEMO 63
Program DEMO1 65
Program DEMO2 67
Program DEMO3 69
Program DEMO4 71
Program DEMO5 73

5. สรุปผลของโครงงานและข้อเสนอแนะ 75
5.1 สรุปผลที่ได้จากโครงงาน 75
5.2 ข้อเสนอแนะในการพัฒนาโครงงาน 75

เอกสารอ้างอิง 77
กิตติกรรมประกาศ 78
ภาคผนวก ก Circuit 79
ภาคผนวก ข PROGRAM 88
ภาคผนวก ค Data IC 127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

จะพบว่าในทุกวันนี้ Computer นับได้ว่ามีบทบาทมากในชีวิตประจำวันของคนเราเกือบทุกคน ไม่ว่าจะเป็นทั้งโรงงานอุตสาหกรรมหรือบริษัทห้างร้านต่างๆ ต่างก็ใช้คอมพิวเตอร์ทั้งนั้น ซึ่งจะพบว่าคอมพิวเตอร์นั้นไม่ได้จำกัดอยู่แต่เพียง Software เท่านั้น โดยเราสามารถที่จะนำคอมพิวเตอร์นี้ไปประยุกต์ใช้งานทางด้าน Hardware ได้อีกมากมายจะทำให้คอมพิวเตอร์นี้มีประโยชน์มากขึ้นและ จะไม่ถูกจำกัดขอบเขตของตัวมันเองอีกต่อไป ในปริญญานิพนธ์นี้ก็ได้นำคอมพิวเตอร์ไปประยุกต์ใช้งานอีกด้านหนึ่ง ซึ่งได้เล็งเห็นว่าผู้ใช้งานนั้นสามารถที่จะทำความเข้าใจถึง การใช้คอมพิวเตอร์ร่วมกับอุปกรณ์ภายนอกได้ว่าทำกันอย่างไรซึ่งอาจจะเห็นได้ว่ามันง่าย แต่จะมีสักกี่คนที่จะเข้าใจเกี่ยวกับเรื่องนี้ ปริญญานิพนธ์นี้จึงได้ทำการออกแบบวงจร และใช้ Program ควบคุมการทำงานของระบบ (ใช้ภาษา Pascal) ซึ่งสามารถที่จะให้ผู้ใช้งานสามารถเรียนรู้การติดต่อกับอุปกรณ์ภายนอก และผู้ที่ต้องการนำเอาไปใช้งานในการควบคุมอุปกรณ์ภายนอกได้เช่นการ ปิด-เปิด อุปกรณ์ที่ใช้ไฟ AC ต่างๆ หรืออาจจะทำการตั้งเวลาการปิดเปิดได้แล้วแต่ความต้องการ และสามารถที่จะทำการเรียนรู้เกี่ยวกับการส่งข้อมูลและรับข้อมูลออกทาง Port8255 รวมไปถึงการศึกษา IC8253 ซึ่งทำหน้าที่เป็น Timer/Counter โดยอาจจะพบได้ว่ามีใช้ในพวก Single Board แล้วซึ่งมีขนาด 8 Bit แต่ในที่นี้เราจะเล่นกับเครื่อง 16 Bit บ้างว่าทำกันอย่างไร

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อที่จะสามารถควบคุมอุปกรณ์ภายนอกได้ 32 CH ขึ้นอยู่กับความต้องการ

1.2.2 เพื่อที่จะสามารถเรียนรู้เกี่ยวกับ IC 8255 เกี่ยวกับการติดต่อกับอุปกรณ์ภายนอก

1.2.3 เพื่อที่จะสามารถเรียนรู้การทำงานของ IC 8253 ใน Mode การทำงานต่างๆ เพื่อนำไปประยุกต์ใช้งาน

1.2.4 เพื่อที่จะเรียนรู้วิธีการออกแบบและสร้างวงจรในการติดต่อกับคอมพิวเตอร์พีซี และวงจรต่างๆ ภายในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.5 เพื่อให้ผู้ใช้งานสามารถได้เข้าใจ และ เรียนรู้ถึงวิธีการติดต่อกับอุปกรณ์ภายนอกทำกันอย่างไร

1.2.6 สามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง

1.3 ขอบเขตของโครงงาน

1.3.1 ทำการใช้งาน ณ.อุณหภูมิห้อง

1.3.2 สามารถติดต่อกับอุปกรณ์ภายนอกได้ 32 CH

1.3.3 กำเนิดแรงดันอยู่ในช่วง 0-10 V

1.3.4 กำเนิดความถี่ Sine Wave ได้ 1Hz - 2MHz

1.3.5 สามารทำหน้าที่เป็น Timer/Counter

1.4 วิธีการดำเนินงาน

ในการดำเนินงานของปริญญานิพนธ์นี้ได้แบ่งออกเป็น 7 ส่วนด้วยกัน ซึ่งแต่ละส่วนจะมีรายละเอียดดังต่อไปนี้

1.4.1 ทำการศึกษาและออกแบบวงจร ในส่วนนี้เราจะต้องศึกษาถึงคุณสมบัติของวงจรและอุปกรณ์ที่จะนำมาใช้ในการทดลองและจะต้องจัดหา อุปกรณ์ที่มีแล้วนำมาใช้งาน อุปกรณ์ที่ได้เหล่านั้นจะนำมาใส่แทน Block ต่างๆ ที่ได้ทำการออกแบบไว้ให้เหมาะสมกับความต้องการ

1.4.2 สร้างวงจร เมื่อได้แบบมาแล้วก็จะต้องนำวงจรที่ได้นั้นมาทำการทดลองบนแผ่น Proto Board ว่าเป็นไปตามที่คาดหวังหรือไม่

1.4.3 เขียนโปรแกรมควบคุม โดยจะเอาทฤษฎีที่ได้ทำการเขียนโปรแกรมโดยจะนำแนวทฤษฎีมาเขียน โฟล์ชาร์ตก่อน แล้วจึงถอดมาเป็นโปร-แกรม ซึ่งในโปรแกรมที่ใช้ในการควบคุมนั้นจะใช้ Pascal V5.5 ซึ่งเป็นเป็นของบริษัท Borland

1.4.4 ทดลองวงจรเมื่อผ่านขั้นตอนที่ 2,3 แล้วถึงขั้นตอนว่าด้วยการทดลองเป็นไปดังเป้าหมายหรือไม่ โดยจะตรวจสอบผลจากการทดลองว่ามีข้อผิดพลาดอะไรหรือไม่ และวงจรจะทำงานหรือไม่

1.4.5 แก้ไขปรับปรุงโปรแกรมและวงจรให้ดีขึ้น ในส่วนของโครงงานจะประกอบไปด้วยโปรแกรม และวงจรเราจะต้องแก้ไขข้อผิดพลาดที่เกิดขึ้นจากขั้นที่ 4 ให้ดีขึ้น เพื่อที่จะให้โครงงานนี้มีประสิทธิภาพที่ดีที่สุดเท่าที่จะทำได้ โดยการนำข้อผิดพลาดที่ได้จากขั้น 4 มาหาจุดบกพร่องและแก้ไข ก่อนที่จะสรุปผลต่างๆ ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.6 สรุปผล ร่างปริญญานิพนธ์ และตรวจทาน ซึ่งในขั้นตอนนี้จะพบว่าเมื่อชิ้นงานที่ทำสำเร็จลุล่วงลงแล้วเราต้องทำการจัดทำ ปริญญานิพนธ์ขึ้นมาเพื่อที่จะทำให้รู้ลักษณะของขอบเขตของโครงงานที่ทำ อีกทั้งยังเป็นแนวทางให้รุ่นต่อๆ ไป

1.4.7 พิมพ์ ตรวจทานและแก้ไข พร้อมเสนอรายงานเราจะทำการจัดพิมพ์ปริญญานิพนธ์ที่ได้จัดทำขึ้นแล้วตรวจสอบความเรียบร้อยเป็นอันเสร็จภาระกิจที่ทำพร้อมที่จะเสนอรายงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ทฤษฎีของระบบ

ในส่วนนี้จะเป็นการอธิบายทฤษฎีที่เกี่ยวข้องกับ PROJECT ชิ้นนี้ ซึ่งได้แบ่งออกเป็นหัวข้อต่างๆดังต่อไปนี้

2.1.1 SLOT บน IBM/PC
2.1.2 I/O MAP ของ IBM
2.1.3 การใช้งาน IC.8255
2.1.4 IC. PAL
2.1.5 DAC,ADC
2.1.6 การใช้งาน IC.8253

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 สัญญาณต่างๆ บน Slot IBM/PC

จะพบว่าภายใน IBM/PC ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรอินเทอร์เฟสเข้าไปในระบบได้ในภายหลัง โดยผ่านทางสล็อตที่อยู่บน Main Board สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต (สำหรับใน IBM PC/XT) ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขาแบ่งออกเป็น 2 ข้างๆ ละ 31 ขาส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้ จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อตโดยขาที่อยู่ทางด้านซ้ายของสล็อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขาเช่นขา B16 ก็คือขาที่อยู่ทางด้านซ้ายของสล็อตขาที่ 16 (นับจากทางด้านท้ายของเครื่อง) ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกโดยใช้อักษร "A" นำหน้าเลขตำแหน่งของขา เช่น ขา A24 ก็คือขาทางด้านขวาของสล็อตขาที่ 24 (นับจากทางด้านท้ายของเครื่อง) แต่ละขาของสล็อตเหล่านี้จะเชื่อมต่อกับเส้นสัญญาณต่างๆบนเมนบอร์ด ทำให้การวางจรอินเทอร์เฟสกับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus), บัสข้อมูล (Data Bus), บัสควบคุมสำหรับการเขียน/อ่านข้อมูลจากหน่วยความจำหรือพอร์ท I/O, เส้นสัญญาณสำหรับการขออินเทอร์รัพท์ของวงจรอินเทอร์เฟส, เส้นสัญญาณสำหรับการขอ DMA, สัญญาณฐานเวลา (Timing Signal) ต่างๆที่ใช้ในระบบ, เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำและสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHCK)

การที่เราจะติดต่อกับวงจร InterFace ที่ต่อเพิ่มเข้าไปในระบบจะต้องรู้เกี่ยวกับตำแหน่งหน้าที่ของแต่ละขาบน Slot ที่จำเป็นในการใช้งานก่อน เพื่อที่จะให้เรียนรู้และเข้าใจได้ง่าย เพื่อที่จะสามารถต่อสัญญาณต่างๆ ของระบบกับ ชิ้นงานที่สร้างขึ้นได้ และสามารที่จะทำการดึงสัญญาณ Address มาใช้ในการ Decode เพื่อใช้ในการติดต่อกับ ชิ้นงานได้

1) รายละเอียดเกี่ยวกับขาสัญญาณต่างๆที่จำเป็น

Reset DRV (ขา B2) : ขาสัญญาณนี้เป็นเอาท์พุท ซึ่งจะแอคทีฟ (ลอจิก "1") ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอคทิฟไปจนกว่าระบบต่างๆ ภายใน IBM/PC จะพร้อมที่จะทำงานได้จากนั้นสัญญาณนี้ก็จะเปลี่ยนกลับเป็นลอจิก "0" นอกจากนี้ในระหว่างการทำงานของ IBM/PC ถ้าระดับแรงดันของแหล่งจ่ายไฟตกลง สัญญาณนี้ก็จะถูกทำให้แอคทีฟเช่นกัน โดยทั่วไปแล้วสัญญาณนี้จะถูกนำไปใช้ในการรีเซ็ทวงจรอินเทอร์เฟสหรืออุปกรณ์ I/O ต่างๆ ในช่วงที่เริ่มจ่ายไฟให้ กับระบบซึ่งจะเป็นการทำให้วงจรหรืออุปกรณ์เหล่านั้นถูกปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้อยู่ในสภาวะที่แน่นอนก่อนที่จะเริ่มต้นการทำงานในระบบ (สภาวะนี้เป็นสภาวะที่เราทราบ และต้องการให้วงจรทำงานในขณะที่ระบบถูกรีเซ็ท)

A0-A19 (Addresss Bus: ขา A31-A12) : ขาสัญญาณทั้ง 20 ขานี้เป็นเอาท์พุท ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำ หรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อด้วย โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (Least Significant Bit) และ A19 จะมีนัยสำคัญสูงสุด (Most Significant Bit) สำหรับค่าแอดเดรสบนบัสแอดเดรส A0-A19 นี้ จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน/เขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA-Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ) จะเห็นได้ว่าจำนวนเส้นแอดเดรสนี้จะมีอยู่ 20 เส้น ซึ่งสามารถที่จะอ้างแอดเดรสของหน่วยความจำได้ถึง 1Mbyte แต่อย่างไรก็ตามจะมีแอดเดรสบางแอดเดรสที่ถูกใช้งานโดย IBM/PC อยู่ก่อนแล้ว คือ แอดเดรสของหน่วยความจำ RAM บนเมนบอร์ดที่ถูกใช้โดยระบบจำนวน 64Kbyte(สำหรับ IBM PC/XT จะเป็นจำนวน 256 Kbyte) และแอดเดรสสำหรับหน่วยความจำ ROM อีก 48Kbyte ซึ่งถูกจัดในช่วงของแอดเดรสบนสุดใน 1Mbyte คือ OFC00H จนถึง OFFFFFH (สำหรับ IBM PC/XT จะเป็น 64Kbyte)

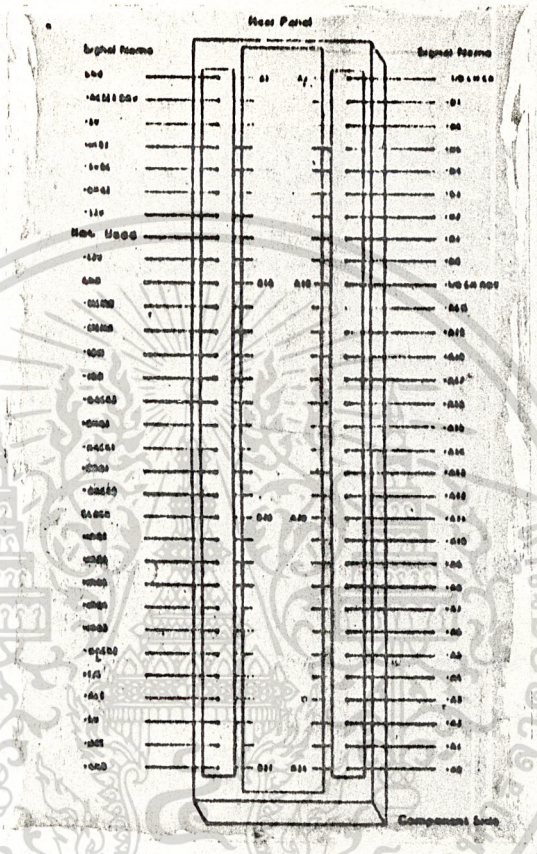
สำหรับการอ้างแอดเดรสของพอร์ท I/O นั้น จะใช้เส้นแอดเดรสเพียง 16 เส้น คือ A0-A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ทได้ 64K พอร์ท โดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอดเดรสที่เหลือคือ A16-A19 นั้นจะไม่ถูกใช้งาน อย่างไรก็ตามภายใน IBM/PC จะใช้เส้นแอดเดรสในการอ้างแอดเดรสของพอร์ทเพียง 10 เส้น คือจาก A0-A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

D0-D7 (Data Bus; ขา A9-A2) : ขาสัญญาณนี้จะเป็นแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ท I/O กับ IBM/PC โดยบิท D0 จะมีนัยสำคัญต่ำสุดและบิท D7 จะมีนัยสำคัญสูงสุด สำหรับในบัสไซเคิลของการเขียนข้อมูลที่สร้างขึ้นโดย 8088 นั้น ข้อมูลจะถูกส่งออกมาบนบัสข้อมูล ก่อนที่สัญญาณ IOW (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ท) หรือ MEMW (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น) ซึ่งโดยทั่วไปขอบขาขึ้นของสัญญาณ IOW หรือ MEMW นี้ จะถูกใช้เพื่อสั่งให้พอร์ท I/O หรือหน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลไปเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ท I/O หรือหน่วยความจำที่ถูกอ้างถึงจะต้องส่งข้อมูลออกมาบนบัสข้อมูล ก่อนที่สัญญาณ IOR (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ท) หรือ MEMW(ในกรณีที่ต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น)



รูปที่ 2.1 แสดงตำแหน่งขาต่างๆ บน Slot IBM

IOR (I/O Read; ขา B14) : ขาสัญญาณนี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอบขาขึ้นของสัญญาณ IOR เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง สำหรับในขบวนการ DMA 8237A-5 DMA Controller จะทำการสร้างสัญญาณ IOR เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำ (แทนที่จะเป็นแอดเดรสของพอร์ท I/O) ที่พอร์ท I/O ที่ขอ DMA ต้องการจะนำข้อมูลไปเก็บ การที่พอร์ทใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้น จะอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนด เช่นกรณีที่สัญญาณ DACK1 แอคทีฟก็จะแสดงว่าพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O ที่จะต้องส่งข้อมูลออกมาบนบัสข้อมูลก็คือพอร์ท I/O ที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) เป็นต้น

IOW (I/O Write; ขา B13) : ขาสัญญาณนี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลที่อยู่บนบัสข้อมูลข้อมูลอาจจะยังไม่สมบูรณ์ ดังนั้นในการออกแบบจึงควรใช้ขอบขาขึ้นของสัญญาณ IOW แทนขอบขาลงในการทำให้พอร์ท I/O ที่เกี่ยวข้องรับข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลสมบูรณ์เสียก่อน สำหรับในขบวนการ DMA นั้น DMA-Controller จะทำการสร้างสัญญาณ IOW เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำที่พอร์ท I/O ที่ขอ DMA ต้องการจะอ่านข้อมูล

AEN สัญญาณนี้เป็นเอาท์พุทที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก "1") นั้นเป็นบัสไซเคิลของขบวน การ DMA สำหรับบนเมนบอร์ดของ IBM/PC นั้นจะใช้สัญญาณนี้ในการดิสเอเบิล (Disable) 8288 Bus Controller และจะใช้ดิสเอเบิลพอร์ท I/O ต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้นนี้ ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส และจะทำให้สัญญาณ IOR หรือ IOW แอคทีฟด้วย ดังนั้นถ้าไม่ทำการดิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็อาจจะทำให้พอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรส (ซึ่งเป็นแอดเดรสของหน่วยความจำ) นั้น ทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดขึ้นได้

2) การจัดสัญญาณบนสล็อตของ IBM PC/XT

สำหรับใน IBM PC/XT นั้นจะมีสล็อตสำหรับเชื่อมต่อกับวงจรภายนอกได้มากขึ้นคือใน IBM PC/XT จะทำการเพิ่มจำนวนสล็อตบนเมนบอร์ดขึ้นเป็น 8 สล็อต จากเดิมที่มีอยู่เพียง 5 สล็อตบน IBM PC โดยการจัดสัญญาณต่างๆ ในทั้ง 8 สล็อตจะยังคงเหมือนกับใน IBM PC เพียงแต่สัญญาณต่างๆ ที่จะถูกส่งออกมายังขาของสล็อตที่ 8 นั้น จะถูกต่อผ่านวงจรขับกระแส (Buffer) ก่อน และในสล็อตที่ 8 นี้ขา B8 จะถูกใช้งานด้วย โดยจะถูกใช้เป็นขา CARD SLCTD (หรือ Card Selected) ซึ่งขาสัญญาณนี้จะเป็นสัญญาณอินพุทจากวงจรภายนอกที่เสียบอยู่บนสล็อตที่ 8 เพื่อให้วงจรบนเมนบอร์ดทราบว่าคาร์ดที่อยู่บนสล็อตนี้ถูกเลือกใช้งานอยู่ ซึ่งจะทำให้ Driver บนเมนบอร์ดทำการอ่านหรือส่งข้อมูลไปยังสล็อตที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

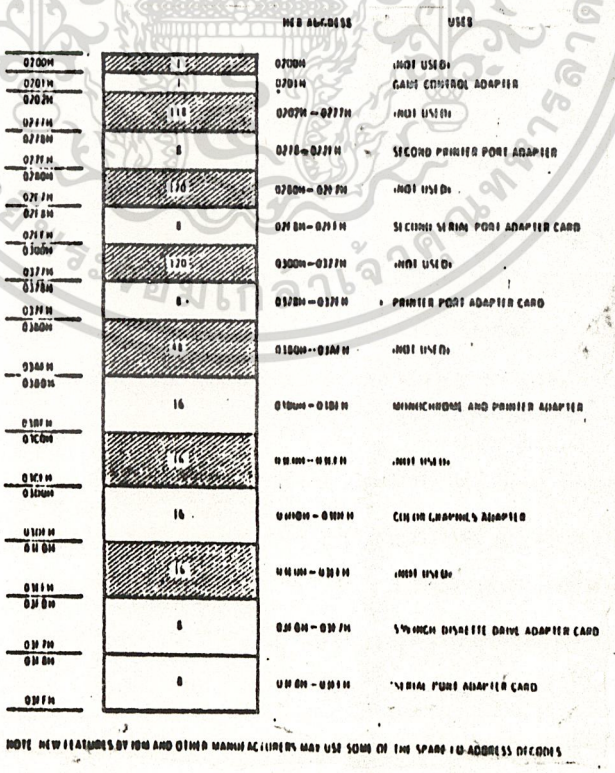
2.1.2 การจัดแอดเดรสสำหรับพอร์ท I/O ใน IBM PC

ในหัวข้อนี้จะกล่าวถึงวิธี การอ้างและใช้งานแอดเดรสต่างๆของพอร์ท I/O ที่ใช้งานอยู่ใน IBM/PC ในการควบคุมและตรวจสอบสภาวะการทำงาน รวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพซัพพอร์ทหรือคาร์ดต่างๆ ที่ใช้ระบบของ IBM/PC นั้นจะกระทำโดยผ่านทางพอร์ท I/O ของระบบ ดังนั้นในการที่จะใช้งานหรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ท I/O ต่าง ๆ ของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ทเหล่านี้ต้องกระทำ โดยการอ้างถึงแอดเดรสของพอร์ท I/O เหล่านั้นโดยตรง เราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสของ 8088 ใน IBM/PC ด้วยสำหรับแอดเดรสของพอร์ท I/O ต่างๆ นั้น จะเป็นแอดเดรสที่ถูกสร้างขึ้นโดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะคือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ส่วนการส่งข้อมูลให้กับพอร์ทเหล่านี้จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้นไปยังแอดเดรสของพอร์ทที่ต้องการ และ สำหรับการตรวจสอบหรือการอ่านข้อมูลจากพอร์ท ก็จะทำได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสของพอร์ทที่ต้องการเช่นกัน ภายในไมโครโปรเซสเซอร์เบอร์ 8088 นี้จะมีแอดเดรสสำหรับใช้กับพอร์ท I/O อยู่ทั้งสิ้น 65,536 หรือ 64K แอดเดรส (ในขณะที่มีแอดเดรสสำหรับหน่วยความจำอยู่ 1Mbyte) ซึ่งทำให้การอ้างแอดเดรสของพอร์ท I/O ที่ทำงานร่วมกับ 8088 นั้น ต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้นคือ A0-A15 แต่สำหรับใน IBM/PC นี้ ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ 10 เส้นล่าง คือ A0-A9 เท่านั้น ดังนั้นในการอ้างถึงแอดเดรสของพอร์ทของอุปกรณ์หรือชิพซัพพอร์ทใด ๆ ที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วย โดยเส้นแอดเดรสที่เหลือคือ A10-A15 นั้นจะไม่ถูกนำไปใช้งาน อย่างไรก็ตามถึงแม้ว่าเส้นแอดเดรส A10-A15 นี้จะไม่ถูกนำไปใช้งาน แต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วยเพียงแต่ไม่ได้ถูกนำมาดีโค้ดร่วมกับแอดเดรส A0-A9 เท่านั้น ตัวอย่างเช่นในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0010H นั้น จะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0410H, 0810H, 0C10H, ทั้งนี้เนื่องจากแอดเดรส 6 บิทบนไม่ได้ถูกใช้งาน จึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10-A15 นั้น ไม่ทำให้เกิดความแตกต่างใด ๆ ขึ้น

เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0-A9)ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
027022

(จากจำนวน 64K พอร์ท) เท่านั้นนอกจากนี้ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ทของ IBM/PC ข้อมูลในบิท A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1024 พอร์ทออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ท) อีกด้วย กล่าวคือถ้าข้อมูลในบิท A9 เป็น "0" แล้วเราจะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์หรือชิพชัพพอร์ทต่างๆที่อยู่บนเมนบอร์ด (Main Board)ของ IBM/PC เช่น 8253-5, 8237-5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิท A9 นี้เป็น "1" ก็จะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนคาร์ดต่างๆ เท่านั้นจากที่ได้กล่าวมานั้นจะสรุปได้ว่าพอร์ทบน IBM/PC ทั้ง 1024 พอร์ท เราสามารถที่จะเลือกส่งไปยังพอร์ทใดๆ ใน IBM/PC ได้ดังนั้นการเลือกแอดเดรสสำหรับพอร์ทที่อยู่บนคาร์ดจึงสามารถทำได้โดยสะดวกแต่อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงก็คือ ถ้าแอดเดรสที่เราเลือกให้กับพอร์ทนี้ตรงกับค่าแอดเดรสเดิมที่มีอยู่บนเมนบอร์ดแล้ว เมื่อเราทำการส่งข้อมูลให้กับพอร์ทที่อยู่บนคาร์ดด้วย ซึ่งในกรณีเช่นนี้อาจจะก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรสให้กับพอร์ทที่ถูกสร้างขึ้นบนคาร์ดต่างๆ จึงควรจะใช้ค่าแอดเดรสที่แอดเดรสบิท A9 มีค่าเป็น "1" คือ แอดเดรส 0FE00H จนถึง 0FFFFH เท่านั้น (แอดเดรสบิท A10-A15 ไม่ถูกใช้ในการดีโค้ด แต่เพื่อความสะดวกจึงกำหนดให้มีค่าเป็น "1" ในฐานสองทั้งหมด แต่ในการใช้งานจริงอาจเปลี่ยนให้แอดเดรส A10-A15 แต่ละบิทมีค่าเป็น "1" หรือ "0" ก็ได้)



รูปที่ 2.2 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนการ์ดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การใช้งาน 8255 PPI

จะพบว่าถ้าต้องการให้เครื่อง Computer ทำการติดต่อกับอุปกรณ์ภายนอก (Peripheral Interface) นั้น เราจะต้องมี IC ที่ทำหน้าที่เป็น I/P และ O/P โดยทั่วไปแล้วมักจะใช้ IC จำพวก TTL (หรือจะเป็น CMOS ก็ได้) ในการ Interface กับ Computer ซึ่งจะพบว่าถ้าต้องการติดต่ออุปกรณ์ I/O ที่มากๆ เช่น 32 CH เท่ากับว่าเราจะต้องมี IC ที่หน้าที่เป็น I/P (เช่นใช้เบอร์ 74LS374 D FLIP FLOP ซึ่งมี I/P DATA 8 I/P) ใช้ 4 ตัว และตัวที่ทำหน้าที่เป็น O/P (เช่น ใช้เบอร์ 74LS244 Buffer 3 state)ใช้ 4 ตัวเช่นกัน รวมกันแล้วจะต้องใช้ถึง 8 ตัวถึงจะทำหน้าที่เป็น I/O 32 CH และจะยุ่งยากในการเลือก I/O แต่ละชุดอีก เราสามารถที่จะขจัดปัญหาในการติดต่อเหล่านั้นให้หมดไปได้โดยใช้ IC สำเร็จรูปที่ทำหน้าที่เป็นทั้ง I/P และ O/P อยู่ในตัวเดียวกันและสามารถที่จะควบคุมได้ด้วยว่าต้องการที่จะให้ Chip ตัวนี้เป็นอะไรก็ได้ (เป็นทั้ง I/P และ O/P,O/P อย่างเดียว, I/P อย่างเดียว) ขึ้นอยู่กับโปรแกรมที่ใช้ควบคุม Chip IC ตัวนี้จะมีประโยชน์มากต่องาน Electronic และใช้ได้ทั่วๆ ไปมีชื่อเรียกว่า 8255 PPI (Programmable Peripheral Interface) ใน Project นี้จะใช้ 82C55A เป็น C-MOS High speed ทำงานที่ความถี่ 8 MHz ซึ่งสามารถ Interface กับ IBM ได้ เป็นของบริษัท UMC

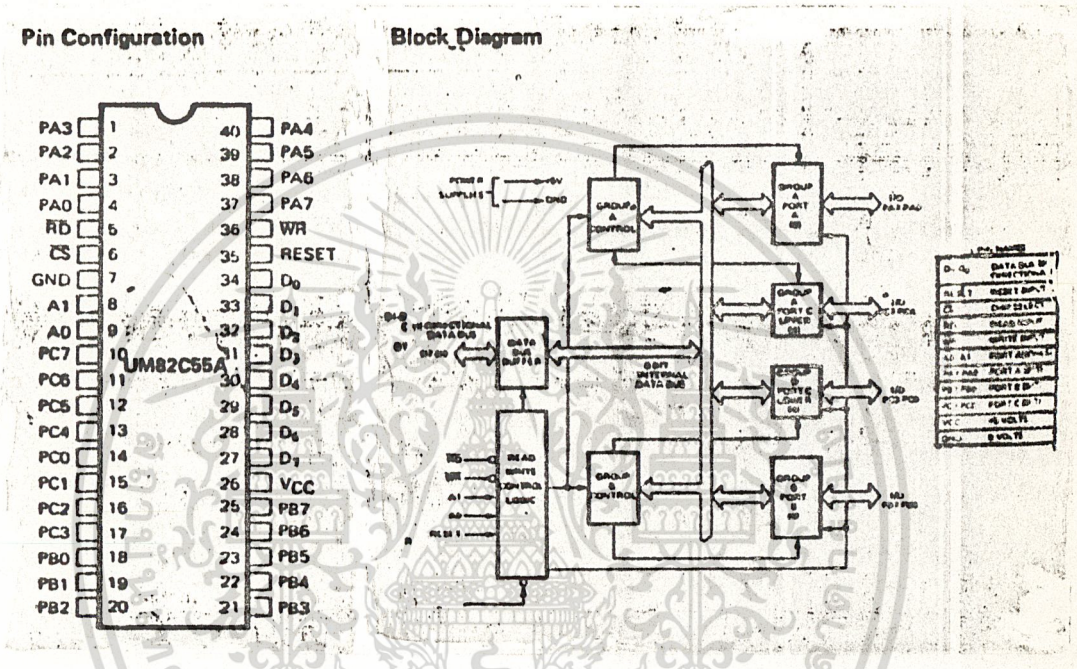
1) รายละเอียดเกี่ยวกับ 82C55A

82C55A นี้เป็นอุปกรณ์ประเภท Cmos ซึ่งมาตรฐานเดียวกับ 8255A ภายในจะเป็นเกทจำพวก Cmos ทั้งสิ้น มันเป็นอุปกรณ์จำพวก I/O โดยประกอบไปด้วย 24 I/O (มี 3 Prot A,B,C) และมีด้วยกัน 3 mode ซึ่งสามารถใช้งานร่วมกับ 8088 , 8086 , 8048 , 8051 ฯลฯ รายละเอียดและ Block diageam ภายใน IC เบอร์นี้ดังรูป 2.3

Block กลุ่มแรกที่จะกล่าวถึงนี้ ได้แก่ Block จำนวน 4 Block ที่อยู่ทางด้านขวาสุดของรูปซึ่งจะเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกอื่นๆ โดยมีสาย PA0-PA7 ,PB0-PB7 , PC0-PC7 เป็นทางผ่านของข้อมูล ระหว่างอุปกรณ์ภายนอกกับ 82C55A สายสัญญาณเหล่านี้จะถูกแบ่งออกเป็น 3 I/O Port ได้แก่ Port A (PA) ,Port B (PB) และ Port C (PC) Portเหล่านี้แต่ละ Port สามารถเป็นได้ทั้ง Port I/P และ O/P และแต่ละ Port จะมีสายสัญญาณเชื่อมเข้ากับบัสข้อมูลภายในของ 82C55A Block กลุ่มถัดมาได้แก่ Group A control และ Group B control ซึ่งจะเป็นตัว กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการทำงานของทั้ง 3 I/O Port(82C55A มีลักษณะการทำงานที่แตกต่างกัน 3 mode สามารถกำหนดได้โดยการส่ง control word ให้กับ 82C55A) จากรูปที่ 2.3 จะเห็นว่า Port C ซึ่งประกอบด้วยพอร์ทขนาด 4 bit 2 Prot กลุ่มหนึ่งจะถูกควบคุมโดย Group A Control และอีกกลุ่มหนึ่งจะประกอบไปด้วย Group B Control



รูปที่ 2.3 แสดง Block Diagram และการวางตำแหน่งของ 82C55A

Block กลุ่มสุดท้ายได้แก่ DATA BUS BUFFER และ Read/write control Logic ซึ่ง Bolck เหล่านี้จะเป็นส่วนที่ติดต่อกับ CPU DATA BUS ของ CPU ส่วน Read/Write control Logic จะเป็นส่วนที่ควบคุมให้ข้อมูลเข้าออกจาก Register ภายในตัวที่ถูกต้องและในเวลาที่เหมาะสม

2) รายละเอียดของขาต่างๆ ภายใน 82C55A

ในส่วนนี้ เราจะพิจารณาหน้าที่ต่างๆของขา 82C55A ซึ่งจะได้สามารถใช้งานร่วมกับ CPU ได้ถูกต้อง

รายละเอียดของขาต่างๆมีดังนี้

ก) D0-D7เป็นสายข้อมูล INPUT/OUTPUT แบบสองทิศทาง (Bi-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

directionnal Bus) ซึ่งจะเป็นทางผ่านของข้อมูลระหว่าง PORT ต่างๆของ 82C55A กับ CPU โดยสามารถที่จะต่อขนานกับ CPU ได้เลย

ข) CS(Chip Select Input):ขานี้จะทำงาน(Active) ที่สภาวะ Logic"0" เป็นการ Enable 82C55A เพื่อที่จะทำการอ่านหรือเขียนข้อมูล

ค) RD (Read Input): ขานี้จะทำงาน (Active) ที่สภาวะ Logic "0" และสัญญาณ CS ต้องเป็น Logic"0" ข้อมูลนั้นก็จะถูกอ่านเข้ามาสู่ระบบ Bus ข้อมูลของ CPU ได้

ง) WR (Write Input): ขานี้จะทำงาน (Active) ที่สภาวะ Logic "0" และสัญญาณ CS ต้องเป็น Logic"0" ข้อมูลนั้นก็จะถูกเขียนเข้าไปยัง 82C55A ได้

จ) A0-A1 (Address Input): จะเป็นตัวกำหนดการเลือกใช้ Register ภายในของ 82C55A

ฉ) RESET:จะทำงาน (Active) ที่สภาวะ Logic "1"จะทำให้ 82C55A อยู่ในสภาวะ Reset ซึ่งจะทำให้ทุกๆ Port ถูก Setให้อยู่ใน Mode I/P

ช) PA0-PA7,PB0-PB7 ขาสัญญาณเหล่านี้จะถูกใช้เป็นพอร์ท I/O ขนาด 8 บิท ใช้ต่อเข้ากับอุปกรณ์ภายนิกอื่นๆ

ซ) PC0-PC7 ขาสัญญาณนี้ถูกใช้เป็นพอร์ท I/o ขนาด 8 บิท เช่นเดียวกับ PA0-PA7 และ PB0-PB7 แต่กลุ่มของขาสัญญาณเหล่านี้สามารถแบ่งออกเป็น 2 กลุ่ม โดยแต่ละกลุ่มมีขนาด 4 บิท ได้แก่ กลุ่มแรก จะใช้ควบคุม PB0-PB7 และกลุ่มที่ 2 ใช้ควบคุม PA0-PA7

3) การ READ และ WRITE REGISTER ภายใน 82C55A

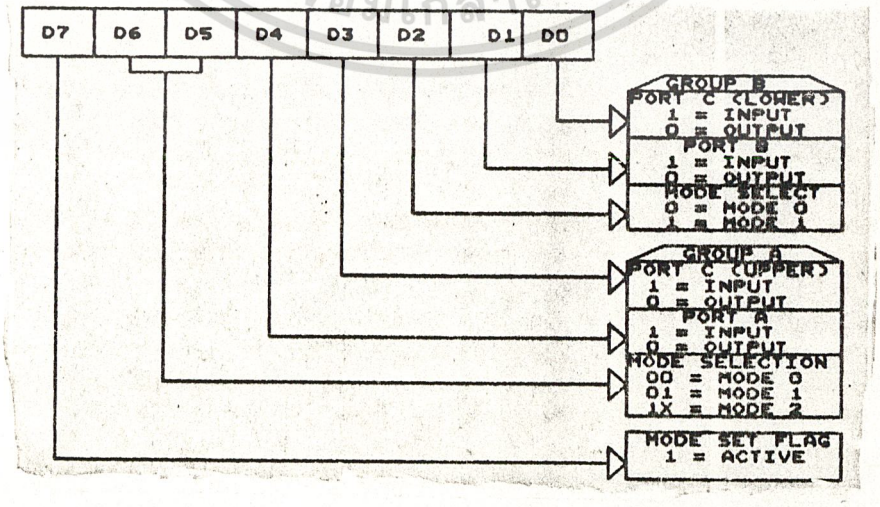
ขณะนี้เราได้ทำการต่อ 82C55A เข้ากับระบบของ 8088(8086) แล้วต่อไปเราจะศึกษาการโปรแกรมใช้งาน 82C55A เพื่อที่จะให้ทำงานตามที่เราต้องการจะเริ่มต้นพิจารณา Register ภายใน 4 ตัวของ 82C55A สำหรับในตัวอย่างการถอดรหัสของเรานี้ตำแหน่งของ Register จะอยู่ใน Address 300H,301H,302H,303H รายละเอียดต่างๆและขาสัญญาณที่ใช้ในการControl มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงการควบคุม Register ภายในของ 82C55

A1	A0	RD	WR	CS	Input Operation (Read)
0	0	0	1	0	Port A ----- Data Bus
0	1	0	1	0	Port B ----- Data Bus
1	0	0	1	0	Port C ----- Data Bus
1	1	0	1	0	Control Word ---- Data Bus
					OutPut Operation (Write)
0	0	1	0	0	Data bus ---- Port A
0	1	1	0	0	Data Bus ---- Port B
1	0	1	0	0	Data Bus ---- Port C
1	1	1	0	0	Data Bus ---- Control
					Disable Function
X	X	X	X	1	Data Bus Tri-State
X	X	1	1	0	Data Bus Tri-State

หน้าที่ของ Register หมายเลข 0-2 จะถูกกำหนดลักษณะการทำงานจาก Register หมายเลข 3 (Control Register) รูปที่ 2.4 จะแสดงรายละเอียดของแต่ละ Bit ของ Register ควบคุมต่อไปจะกล่าวถึงลักษณะการทำงานของ 82C55A ใน Mode ต่างๆ (มี3 Mode) และการ Program ให้อยู่ในโหมดต่างๆดังต่อไปนี้



รูปที่ 2.4 แสดงรายละเอียดแต่ละบิทของ Register ควบคุมของ 82C55A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode ต่างๆ ภายใน 82C55A ซึ่งมีอยู่ด้วยกัน 3 Mode ดังนี้

Mode 0 : Basic Input/Output

Mode 1 : strobed Input/Output

Mode 2 : Bi-Directional Bus

เมื่อขา Resetของ 82C55A มีสภาวะเป็น Logic"1" Port ต่างๆ (Port A-C) ก็จะถูก Set ให้เป็น I/P Mode โดย Bus ภายในจะถูก Hold ไว้หลังจากนั้นถ้าขา Reset ถูกเอาออก (Logic"0") 82C55A ก็ยังคงเป็นสภาวะเดิมจนกว่าเราจะมีการ Initial (การกำหนดให้ก่อนเริ่มทำงาน) ก่อน ใน Project ชิ้นนี้จะกล่าวเฉพาะเพียง Mode เดียวเท่านั้น คือ Mode 0 เพราะเราเพียงแต่ต้องการให้มันเป็นแค่อินพุทและเอาท์พุทแบบธรรมดาเท่านั้น

Mode 0 : รายละเอียดต่างๆใน Mode นี้

1. 8-Bit ขนาด 2 Port และ 4-Bit ขนาด 2 Port
2. ทุกๆ Port สามารถเป็นได้ทั้ง Input และ Output Port
3. Output จะถูก Latch
4. Input จะไม่ถูก Latch
5. มีความแตกต่างระหว่าง Input กับ Output 16 ระดับที่เป็นไปได้

โดยในการ set 82C55A อยู่ใน Mode 0 นั้นเราจะต้องส่งคำสั่งควบคุม (Control Word) ให้แก่ Register ควบคุมก่อน คำสั่งควบคุมนี้จะกำหนดลักษณะการทำงานให้แก่แต่ละ Port ของ 82C55A ตัวอย่างคำสั่งควบคุมที่จะสั่งให้ 82C55A ทำงานใน Mode 0 นี้ได้แก่

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

จากรูปที่ 2.4 เราจะได้ว่า

Bit D_7 เป็นตัวกำหนดว่าเป็นคำสั่งควบคุม (Control Word)

Bit D_6 และ D_5 กำหนด Mode การทำงานของ Port A

D_6, D_5 มีค่าเป็น 0 แสดงว่าอยู่ใน Mode 0

Bit D_4 = "0" กำหนดให้ Port A เป็น Output Port

Bit D_3 = "0" กำหนดให้ Port C 4 Bit บนเป็น Output Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

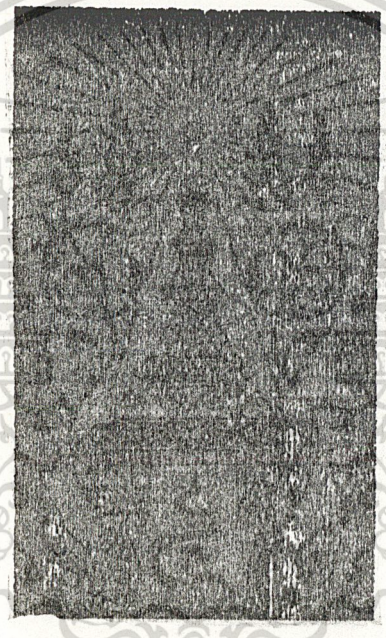
Bit D_2 = "0" Set Mode ของ Port B ให้อยู่ใน Mode 0
Bit D_1 = "0" Set Port B เป็น Port Output
Bit D_0 = "0" Set Port C ให้ 4 Bit ล่างเป็น Port Output

คำสั่งควบคุมนี้จะกำหนดให้ Port ทั้ง 3 ของ 82C55A ทำงานอยู่ใน Mode 0 และเป็น Port Output จะได้สายสัญญาณซึ่งสามารถติดต่อกับอุปกรณ์ภายนอกได้ถึง 24 สายในการที่จะทำให้ 82C55A ทำงานตามที่เราต้องการต่อไปก็ขึ้นอยู่กับการเขียน Soft-ware และส่ง Data (Out Port)ออกไปยังตำแหน่งที่ต้องการ เมื่อเราทำการ Initial Port(โดยการ Out ไปที่ Port 303H) แล้วต่อไปเราก็จะส่ง Data ไปยังแต่ละ Port(ในที่นี้เราทำให้ทุก Port ทำหน้าที่เป็น Output หมด)โดย Port A Out ไปที่ 300H, Port B Out ไปที่ 301H,Port C Out ไปที่ 302H นั่นเอง ซึ่งหลังจากทำการ Run แล้ว Port ต่างๆก็จะมีข้อมูลปรากฏตามที่เราใส่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 ความรู้เกี่ยวกับไอซี PAL

IC PAL นับได้ว่าเป็น อุปกรณ์ที่มีประโยชน์มากชนิดหนึ่ง ซึ่งลักษณะโดยทั่วๆไปแล้วมันมีลักษณะที่คล้ายอุปกรณ์จำพวก PROM ซึ่งก็เป็นที่รู้จักกันมานานแล้วแต่มีลักษณะแตกต่างกันที่โครงสร้างภายในเท่านั้น IC PAL มีลักษณะโครงสร้างภายในที่ตรงข้ามกับ PROM คืออินพุทของ AND เป็นแบบโปรแกรมได้ ส่วนอินพุทของ OR จะคงที่ดังตัวอย่างรูปที่ 2.5 ซึ่งแสดงโครงสร้างภายในของ PAL ขนาด 4-อินพุท-4 เอ๊าท์พุท จากตัวอย่างนี้สังเกตว่าสัญญาณอินพุต I_0-I_3



รูปที่ 2.5 แสดงโครงสร้างภายในของ PAL ขนาด 4-อินพุท 4-เอาท์พุท

จะส่งผ่านบัฟเฟอร์ และอินเวอร์เตอร์ออกมาเป็นสัญญาณ 8 เส้น (I_0-I_3, I_0-I_3) ป้อนเข้า AND ทุกตัว ซึ่ง AND ทุกตัวจะต้องมีอินพุตอยู่ 8 ขา โดยที่สามารถโปรแกรมได้ว่าจะต่อเข้ากับ I_0-I_3 เส้นใด ส่วนเอาต์พุตของ And จะถูกจัดแบ่งเป็น 4 ชุดป้อนเข้า OR อย่างตายตัวคือเอาท์พุต 4 ตัวบนจะป้อนเข้า OR ตัวแรก ถัดลงมา 4 ตัวจะป้อนเข้า OR ตัวที่สอง เป็นเช่นนี้ลงมาเรื่อยๆ ในกรณีนี้หมายถึงว่า OR จะต้องมีอินพุตอยู่ 4 ขาเท่านั้นและตามความหมายของเบอร์ IC PAL ที่กำหนดขึ้นจะมีดังนี้ ในที่นี้จะขอยกตัวอย่างเบอร์ที่ใช้งานใน PROJECT นี้

PAL 16 L 8

ชุดที่ 1 2 3 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดที่1 บอกว่าเป็น IC PAL
ชุดที่2 คือจำนวนอินพุท (มี 8,10,12,14,16,20)
ชุดที่3 คือชนิดของเอาท์พุท (H,L,C,R,X,A)
ชุดที่4 คือจำนวนของเอาท์พุท (2,4,6,8)
ซึ่งรายละเอียดมีดังต่อไปนี้

ก) จำนวนอินพุตเราสามารถเลือกได้หลายขนาดซึ่งขึ้นอยู่กับความต้องการ มี 8,10,12,14,16,18,20 ขา
ข) จำนวนเอาท์พุต มี 2,4,6,8 ขา
ค) บัฟเฟอร์ทางด้านเอาท์พุทสามารถจะป้อนกลับเข้ามาเป็นอินพุตได้
ง) โปรแกรมได้ทั้งทางอินพุตและเอาต์พุท
จ) ทำหน้าที่คำนวณทางคณิตศาสตร์ได้ขึ้นอยู่กับโปรแกรมที่ต้องการ

การกำหนดเบอร์ของ PAL จะแสดงถึงจำนวนอินพุตและเอาต์พุตและและชนิดของเอาต์พุท ดังแสดงความหมายของเบอร์ที่กล่าวมาแล้ว ตัวอักษรซึ่งบอกชนิดของเอาต์พุทมีรายละเอียดดังนี้

H แอคทีฟที่ Logic'1'
L แอคทีฟที่ Logic'0'
C มีให้เลือกทั้งแอคทีฟที่ Logic '1' และ Logic '0'
R เป็นรีจิสเตอร์ หมายถึงรักษาสถานะเอาต์พุตให้ค้างเอาไว้ได้ด้วย พลิบฟลอบและเอาต์พุทยังป้อนย้อนกลับมาที่อินพุตของ AND ได้ (โปรแกรมได้ด้วย)
X เป็นรีจิสเตอร์แบบ EX-OR
A เป็นรีจิสเตอร์คำนวณทางคณิตศาสตร์ได้

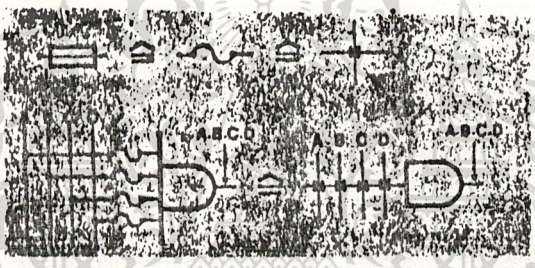
1) สัญญลักษณ์ของลอจิกใน IC PAL

เนื่องจากวงจรลอจิกภายในตัว PAL ค่อนข้างซับซ้อนและมีเป็นจำนวนมาก การเขียนสัญญลักษณ์แบบมาตรฐานทั่วไปเพื่อแสดงวงจรภายใน PAL จึงทำให้ดูสับสนวุ่นวายและต้องเปลือง เนื้อที่มาก โรงงานผู้ผลิตส่วนใหญ่จึงนิยมใช้สัญญลักษณ์ที่กำหนดขึ้นมาใหม่ เพื่อให้เขียนแสดงวงจรภายในได้สะดวกขึ้นและอ่านความหมายจากวงจรได้ง่ายขึ้น ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6 สัญญลักษณ์ของ Gate ในไอซี PAL

สัญญาณจากขาอินพุตมักจะต้องป้อนเข้าบัฟเฟอร์ 2 ตัวก่อนเพื่อให้ได้ 2 สัญญาณที่ตรงกันข้ามกัน คือบัฟเฟอร์ตัวหนึ่งจะไม่กลับลอจิกแต่อีกตัวหนึ่งจะกลับลอจิกเพื่อให้การเขียนวงจรลอจิกภายใน PAL ทำได้ง่ายขึ้นแทนที่จะต้องเขียนสัญญลักษณ์ของบัฟเฟอร์ทั้ง 2 ตัว จึงรวมมาเขียนเป็นตัวเดียวกันเลยโดยให้มีเอาต์พุต 2 สัญญาณดังรูปที่ 2.6



รูปที่ 2.7 แสดงสัญญลักษณ์ของอินพุท Gate แบบ Matrix

ขาอินพุตของเกตมักจะมีขาอินพุตอยู่หลายสัญญาณ และจะต่อเข้ากับฟิวส์ในลักษณะแมทริกซ์ เพื่อให้เขียนวงจรได้ง่ายขึ้นจึงใช้เครื่องหมายกากบาทแสดงถึงว่ามีฟิวส์ต่ออยู่ระหว่างขาอินพุตของเกต กับขาสัญญาณทางแนวตั้งตามตำแหน่งของเครื่องหมายกากบาทนั้นๆ จึงทำให้เขียนขาสัญญาณอินพุทของเกตเพียงเส้นเดียว แสดงจำนวนขาสัญญาณอินพุตของเกตหลายสัญญาณได้ (จำนวนอินพุทของเกต = จำนวนขาสัญญาณอินพุตทางแนวตั้ง) ดังรูปที่ 2.7

รูปที่ 2.8 แสดงสัญญลักษณ์แทนฟิวส์ยังคงอยู่

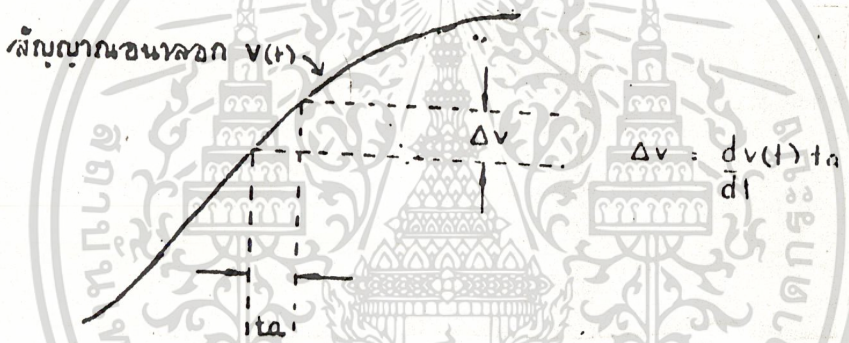
ในกรณีที่ฟิวส์ทุกตัวที่ต่ออยู่เข้ากับ อินพุตของเกตไม่ถูกระเบิดทิ้งไปเลยแม้แต่ตัวเดียว จะเขียนเครื่องหมายกากบาทเพียงตัวเดียวเข้าที่สัญลักษณ์ของเกตเลย ในกรณีเช่นนี้ยังแสดงถึงว่าอินพุทของเกตตัวนั้นจะเป็น '0' เสมอ ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การแปลงสัญญาณแอนะลอกเป็นดิจิตอล และ ดิจิตอลเป็นแอนะลอก

1) ทฤษฎีการสุ่มตัวอย่าง

ในการแปลงสัญญาณแอนะลอกเป็นรหัสดิจิตอลนั้น ADC จะต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวนั้นขึ้นอยู่กับหลายแฟคเตอร์ เช่น ความละเอียดของการเปลี่ยนสัญญาณ เทคนิคของการแปลงสัญญาณ และความเร็วในการทำงานของอุปกรณ์รวมอื่นๆ ความเร็วในการแปลงสัญญาณนี้จำเป็นสำหรับการประยุกต์ใช้งานเฉพาะอย่าง และความแม่นยำที่ต้องการ Aperture Time เป็นช่วงเวลาในการแปลงสัญญาณ ซึ่งคำว่า Aperture Time โดยทั่วไปหมายถึงช่วงเวลาที่เกิดความไม่แน่นอนในการวัด และผลก็คือเกิดผิดพลาดต่อค่าที่วัดได้



รูปที่ 2.9 แสดงข้อผิดพลาดจากการวัดใน Aperture Time

ดังนั้นหากเวลาที่ ADC ใช้ในการเปลี่ยนสัญญาณในเวลา t_a นี้ รหัสดิจิตอลที่ได้อาจจะตรงกับขนาดของ สัญญาณแอนะลอกค่าใดค่าหนึ่งในช่วงนี้และส่วนอื่นๆ ที่เหลือคือความผิดพลาดที่เกิดขึ้น ซึ่งแน่นอนในบางครั้งเป็นไปได้ที่รหัสดิจิตอลจะตรงกับค่าของแอนะลอกที่ถูกต้อง

2) Sample and Hold และ Aperture Error

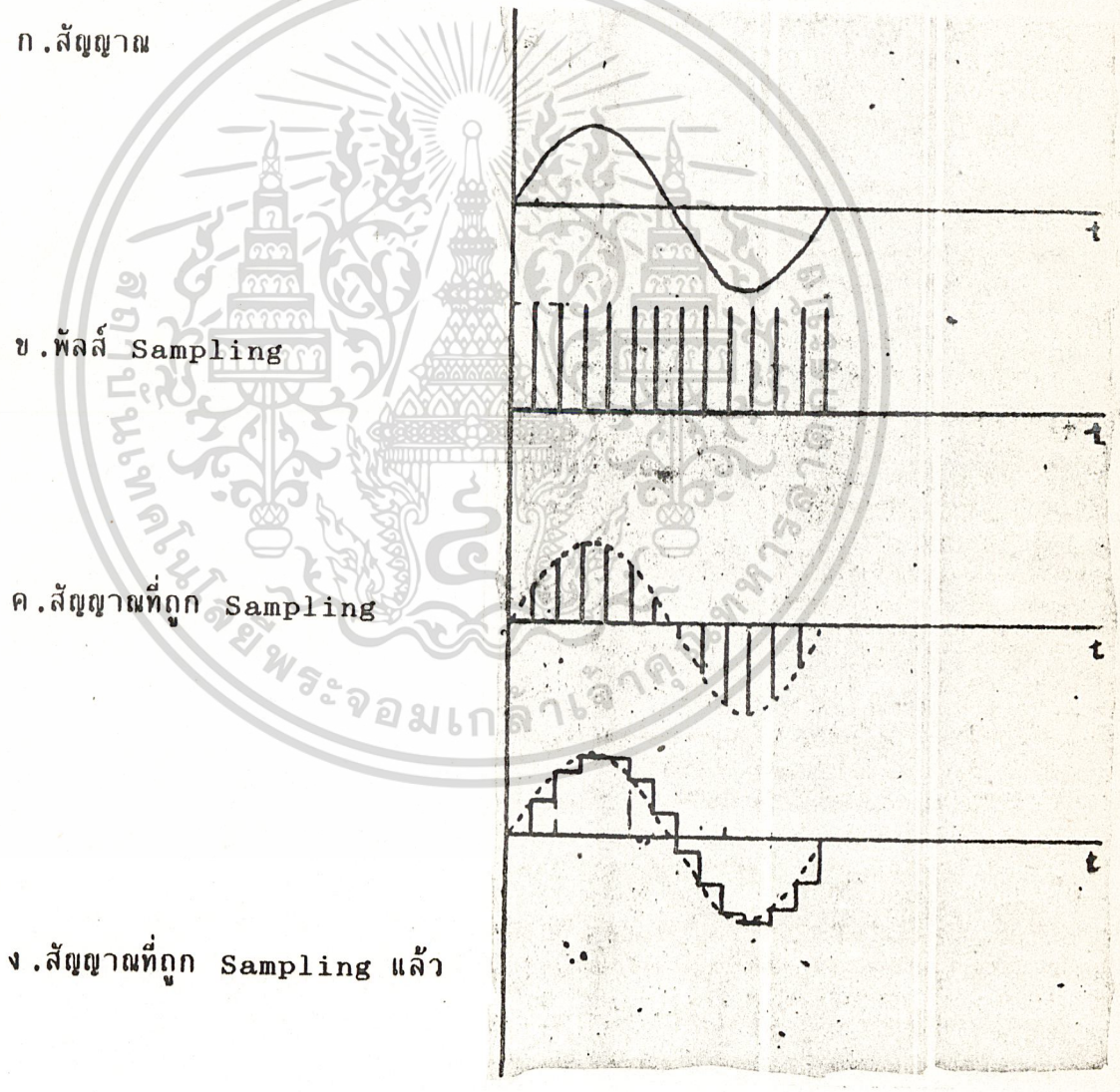
วงจร Sample And Hold จะทำการสุ่มสัญญาณอินพุท และนำสัญญาณที่สุ่มนั้นมาเก็บหรือโฮลด์ไว้ในเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นไว้ในตัวเก็บประจุที่รั่วไหลต่ำ ดังนั้นเมื่อแรงดันอินพุทสามารถคงอยู่ได้นานพอ ADC จึงไม่จำเป็นต้องมีเวลาในการแปลง (Conversion Time) อย่างรวดเร็วนัก Aperture Time ของ Sample And Hold คือเวลาตั้งแต่เริ่มสุ่มสัญญาณ ตัวเก็บประจุค่าแรงดันจนถึงค่าที่สุ่ม ซึ่งสำหรับ Sample

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

And Hold แล้ว Aperture Time ขึ้นอยู่กับแบนด์วิดธ์ และ Switching Time ของอุปกรณ์แอคตีพที่ใช้ในวงจร ซึ่งหาและสร้างได้ง่ายตลอดจนราคาถูกกว่าการสร้าง ADC ความเร็วสูง

3) ระบบ Sample ข้อมูลและทฤษฎีการ Sample

ในระบบการ Sampling สัญญาณแอนะลอกจะถูกสุ่มเป็นระยะๆ คงที่ตาม รูปที่ 2.10 กลุ่มของสัญญาณสุ่มจะแทนสวิทช์ที่ทำงานด้วยความเร็วสูงซึ่งจะทำการตัดต่อสัญญาณแอนะลอกในช่วงเวลาอันสั้น



รูปที่ 2.10 การ Sampling สัญญาณ

ผลของการสุ่มสัญญาณด้วยความเร็ว จะเสมือนกับการคูณขบวนสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัลส์กับสัญญาณแอนะลอก ซึ่งจะได้สัญญาณที่มอดดูเลท (Modulate) ระหว่างขบวนพัลส์ กับ สัญญาณแอนะลอกดังแสดงใน รูปที่ 2.10ค โดยสัญญาณแอนะลอกจะขี่มาบนพัลส์ ถ้าหากเอาตัวเก็บประจุแทนสวิทช์แล้วสัญญาณแอนะลอกที่ถูกสุ่มจะถูก Hold ไว้ในตัวเก็บประจุคอยจนกว่าสัญญาณค่าใหม่ถูกสุ่มเข้ามาใหม่ ลักษณะของเอาท์พุทที่ได้แสดงในรูปที่ 2.10ง

มีปัญหาที่ว่าอัตราการสุ่มสัญญาณนั้นควรมีค่าเท่าใดที่จะไม่ทำให้ ข้อมูลเสียไป เมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเช่นเดิม คำตอบคือขึ้นอยู่กับความถี่ของสัญญาณแอนะลอกและทฤษฎีของการสุ่มกล่าวไว้ว่า "ถ้าสัญญาณต่อเนื่องซึ่งมีความถี่และฮาร์โมนิคส์ไม่เกิน f_s แล้ว สัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาได้อย่างเดิมโดยไม่สูญเสียรายละเอียด หรือผิดเพี้ยนไป ถ้าอัตราการสุ่มไม่น้อยกว่า $2f_s$ ต่อวินาที"

4) Quantizing Theory

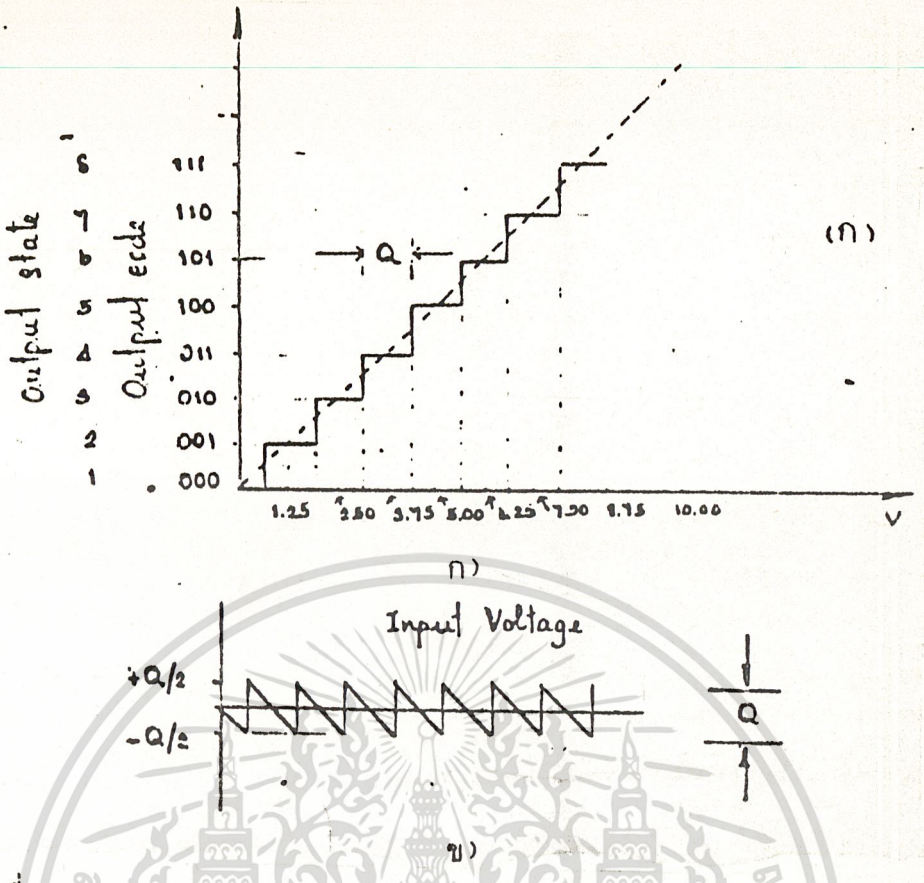
Quantizingเป็นขบวนการที่แปลงสัญญาณแอนะลอกเป็นสัญญาณที่ไม่ต่อเนื่อง (Discrete Signal) ซึ่งจากนั้นก็จะผ่านขบวนการ Coding จัดให้สัญญาณที่ไม่ต่อเนื่องนั้นอยู่ในรูปที่ง่ายต่อการประมวลทำความเข้าใจ และเป็นสัดส่วนสัมพันธ์กับสัญญาณแอนะลอกเช่น อยู่ในรูปของรหัสไบนารี เป็นต้น หากนำเอาขนาดของสัญญาณแอนะลอกและสัญญาณดิจิตอลที่สัมพันธ์จากการ Quantize และการเข้ารหัส (Encode) แล้วมาเขียนกราฟก็จะได้กราฟแสดง Quantize Transfer Function ดังแสดงในรูปที่ 2.11

จุดสำคัญที่เกี่ยวข้องกับกราฟ Transfer Function ในรูปที่ 2.11 อันแรก ได้แก่ รีโซลูชัน (Resolution) ของ Quantizer ซึ่งกำหนดได้จากจำนวนบิทของรหัสดิจิตอล หรือจากกราฟ คือขนาดความกว้างของ Step ทางแกนแอนะลอกว่าเป็นสัดส่วนเท่าใดระหว่างเต็มสเกลแอนะลอกกับค่า 2^n โดยที่ nเป็นจำนวนบิทของรหัสดิจิตอล

5) Quantizer Resolution และ Error

ในแต่ละสถานะของ สัญญาณดิจิตอลเอาท์พุทจะแทนขนาดของสัญญาณแอนะลอกค่าใดค่าหนึ่งในช่วงเล็กๆ ระหว่างจุดแบ่งระดับสองจุด เรียกช่วงเล็กๆ นี้ว่า เป็นขนาดหนึ่ง Analog Quantization หรือหนึ่งควันตัม (Quantum) หรือ 1 LSB (Least Sinificant Bit) ของการแปลงสัญญาณ ตัวอย่างในรูปที่ 2.11ก ควันตัม คือ 1.25 โวลต์ ค่านี้ได้จากการคำนวณดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 Transfer Function ของ Quantize 3 บิท

$$Q = (FSR)/(2^n)$$

FSR คือ ช่วงเต็มสเกลของแรงดันแอนะลอก (Full Scale Range)

n คือ จำนวนบิทของรหัสดิจิตอล

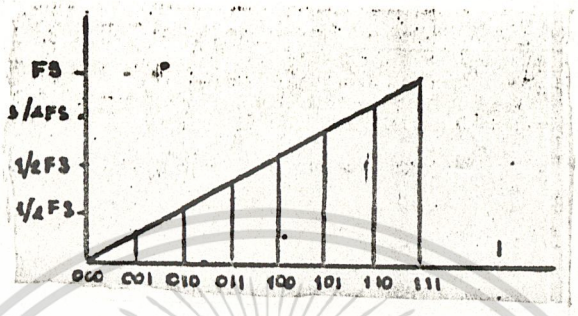
จากสมการจะเห็นว่าหากจำนวนบิทมากขนาดของ Quantizing ก็จะลดลงและถ้าให้อินพุทของ Quantizer กวาดไปตลอดช่วงของสัญญาณแอนะลอกจะเห็นช่วงของผลต่างแอนะลอกอินพุทและดิจิตอลเอาท์พุทเป็นช่วงซึ่งพล็อตได้เป็นรูปฟันเลื่อย ดังรูปที่ 2.11ข เรียกว่า Quantizing Error ซึ่ง Error นั้นคือ 1 ช่วงสัญญาณแอนะลอกแปลงเป็นรหัสดิจิตอล 1 สถานะ Error นี้เป็นธรรมชาติของ Quantizing ซึ่งจะทำการแก้ไขไม่ได้นอกจากการเพิ่มจำนวนบิทของ Quantizer ให้มากขึ้น และเอาท์พุท Error จะอยู่ระหว่าง 0 ถึง Q/2

6) วงจร DAC (Digital To Analog Converter)

DAC นับเป็นอุปกรณ์สำคัญที่ทำให้ดิจิตอลคอมพิวเตอร์เชื่อมโยงกับอุปกรณ์หรือวงจรแอนะลอกอื่นๆ ตัวอย่างการใช้งาน DAC คือ ระบบแสดงผลบนจอภาพ ระบบสังเคราะห์เสียง เป็นต้น และที่สำคัญ DAC ยังเป็นส่วนประกอบชิ้น

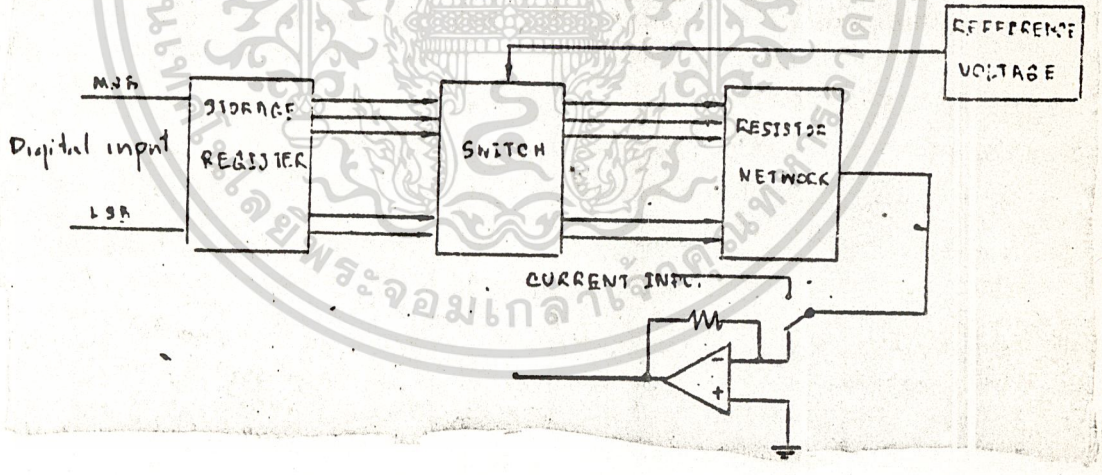
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำคัญในระบบ ADC ที่ใช้กันอยู่ในสมัยปัจจุบัน รูปที่ 2.12 แสดงทรานสเฟอร์ฟังชันของ DAC ขนาด 3 บิท จะเห็นว่ารหัสดิจิตอลอินพุท 1 Word จะแปลงเป็นแรงดันแอนะลอก 1 ค่า



รูปที่ 2.12 ทรานสเฟอร์ฟังชันของ DAC 3 บิทตามทฤษฎี

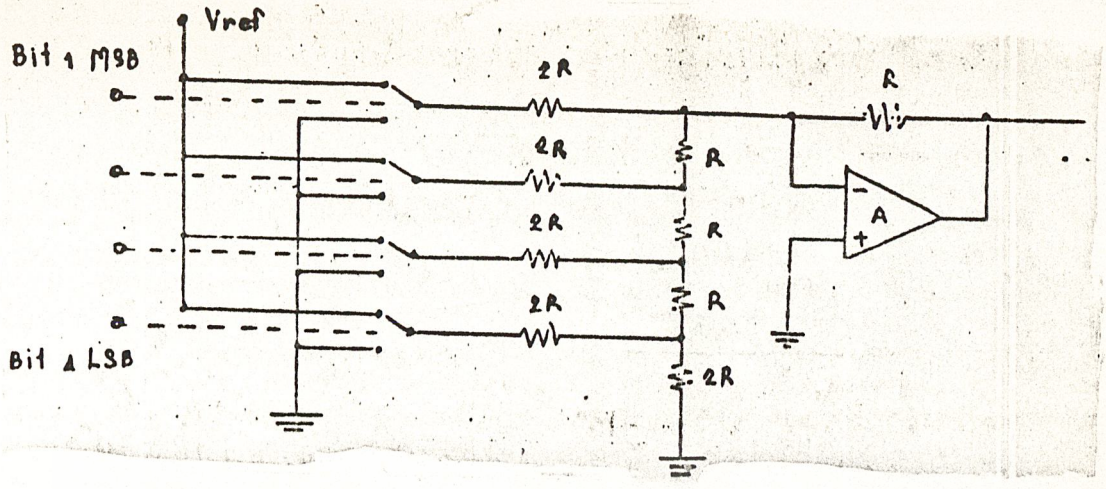
ลักษณะการจัดวงจร DAC เป็นลักษณะดังรูปที่ 2.13



รูปที่ 2.13 บล็อกไดอะแกรมของ DAC

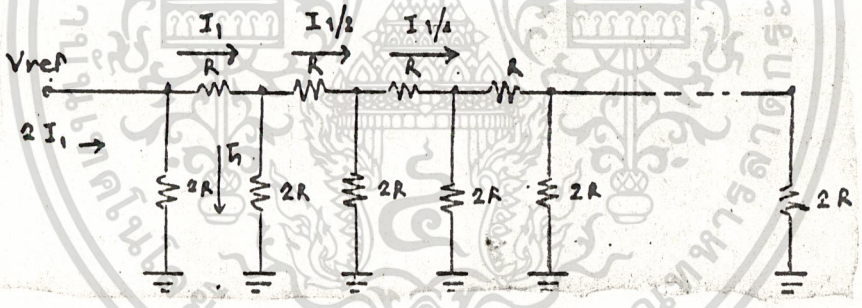
วงจร DAC ที่ใช้กันมีหลายแบบแต่ในที่นี้จะขอกล่าวถึงวงจร DAC แบบ R-2R Ladder ซึ่งเป็นแบบที่ใช้ในบริญญานิพนธ์นี้ ลักษณะของวงจรแสดงดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 วงจร DAC แบบ R-2R Ladder ขนาด 4 บิท

ในวงจรนี้สวิทช์จะตัดต่อให้แรงดันอ้านอิงต่อเข้ากับวงจรแลดเดอร์หรือต่อแลดเดอร์ลงกราวด์ที่ขา 2R จะเห็นได้ว่าสวิทช์อินพุทรีซีสเตอร์ (2R) มองเข้าไปจะเห็นคู่ของรีซีสเตอร์ระหว่างจุดต่อ R-2R ที่ติดกันกระแสจะถูกบั่นทอนไปในอัตรา 2/1 ซึ่งสอดคล้องกับรหัสไบนารี ดังแสดงในรูปที่ 2.15

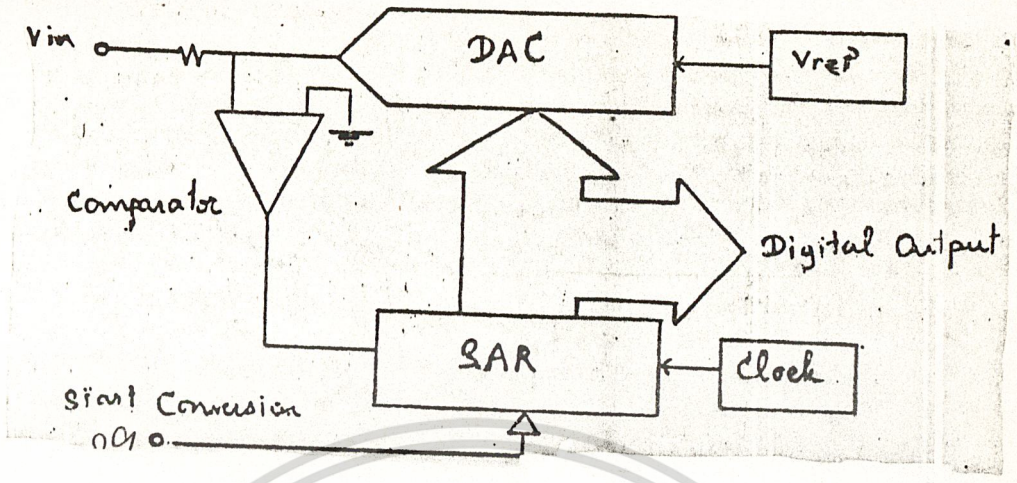


รูปที่ 2.15 วงจรรีซีสทีพแลดเดอร์ DAC

7) Analog To Digital Converter Circuit

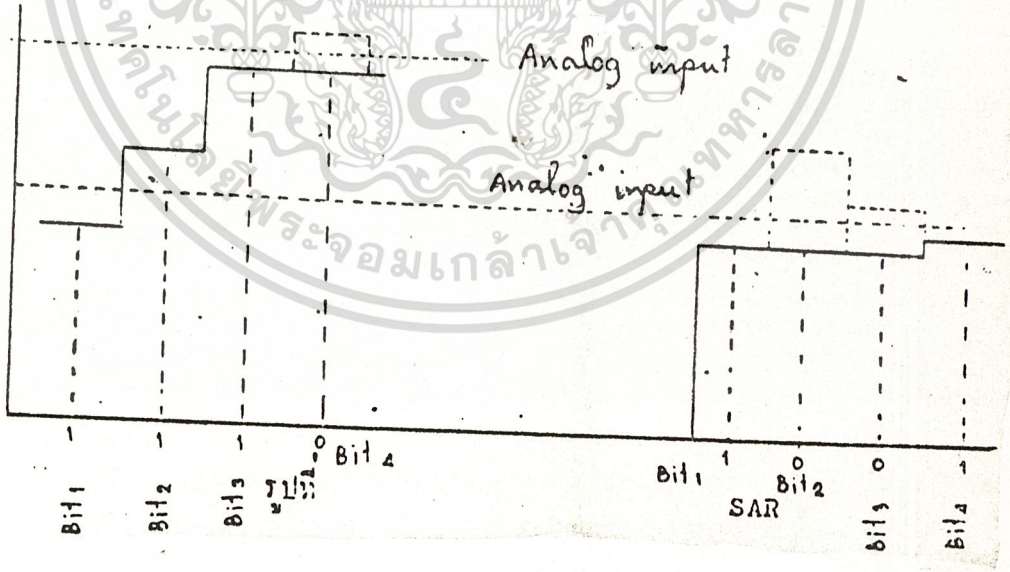
ลักษณะการจัดวงจร ADC มีหลายแบบแต่ที่นิยมใช้มีอยู่ไม่กี่แบบ และส่วนใหญ่จะอยู่ในรูปของวงจรรวม ในที่นี้จะกล่าวเฉพาะ ADC แบบการประมาณซัคเซสซีฟ (Successive Approximation ADC) วงจร ADC ชนิดนี้เป็นเทคนิคที่ได้รับความนิยมในงานประยุกต์ที่ต้องการความเร็วสูงและปานกลาง การจัดวงจรจะคล้ายกันแบบเคาท์เตอร์ที่ทำงานในลักษณะป้อนกลับ บล็อกไดอะแกรมในรูปที่ 2.16 แสดงฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 บล็อกไดอะแกรมของ Successive Approximation Converter

ต่างๆใน ADC ชนิดนี้คอมพอเรเตอร์จะคอยเปรียบเทียบเอาท์พุทจาก DAC กับแอนะลอกอินพุท V_{in} เอาท์พุทจะไปควบคุม Successive Approximation Register (SAR) ซึ่งเป็นไอซี MSI ที่ได้รับการออกแบบเป็นพิเศษเพื่อทำหน้าที่นี้โดยเฉพาะการทำงานของ SAR เป็นดังนี้



รูปที่ 2.17 ไทมิ่งไดอะแกรมของ SAR

รูปที่ 2.17 แสดงไทมิ่งไดอะแกรมของ ADC ที่มีระดับแอนะลอกอินพุท 1 และ 2 เมื่อสัญญาณนาฬิกา (Clock) เข้าไป 1 ลูกจะทำให้ MSB (most significant Bit) เป็น1 บิทอื่นๆ ยังคงเป็นศูนย์ DAC จะเปลี่ยนเอาท์พุทของ SAR เป็นแอนะลอกเปรียบเทียบกับสัญญาณแอนะลอกอินพุท ถ้าผลการเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่คอมพาเรเตอร์ว่าน้อยกว่าอินพุทให้คงบิทนั้นเป็น 1 แต่ถ้ามากกว่าจะให้บิทนั้นเป็น 0 จากนั้นจะทำการทดสอบบิทถัดไปโดยทำให้เป็น 1 หากผลของสองบิทหรือบิทหลังมากกว่าก็จะทำให้บิทนั้นเป็น 0 แต่ถ้าน้อยกว่าให้คง 1 ไว้แล้วทดสอบบิทถัดไปกรรมวิธีดังกล่าวจะกระทำต่อไปจนครบทุกบิท หรือจนกว่าเอาท์พุทจะต่างจาก V_{2n} ไม่เกิน 1 LSB (least significant Bit) ในตัวอย่างแสดงการทำงานเมื่อ V_{2n} ต่ำลงมาอีกระดับหนึ่ง

มีข้อจำกัดประการหนึ่งสำหรับการทำงาน คือสัญญาณแอนะลอกอินพุทจะต้องคงที่ในช่วงเวลาที่ทำการเปลี่ยนแปลงสัญญาณ โดยเปลี่ยนไม่เกิน 1/2 LSB ในช่วงสุดท้ายของการเปลี่ยนสัญญาณดิจิตอล เอาท์พุทจะออกมาขนานกันทุกบิท แต่บางแบบจะให้เอาท์พุทออกมาในลักษณะอนุกรม

วงจร ADC แบบนี้สามารถทำงานได้สองโหมดคือโหมดที่ทำงานอิสระ (Free Run) และโหมดที่รอคำสั่งเริ่มต้นการทำงาน (Start Conversion) จากภายนอก เวลาที่ใช้ในการเปลี่ยนสัญญาณจะใช้ N+1 ลูกของพัลส์ Clock โดย Clock ลูกแรกจะใช้ในการรีเซ็ทรีจีสเตอร์ภายใน ADC เทคนิคนี้จะเที่ยงตรงมากหากมีแรงดันอ้างอิงและ D/A รวมอยู่บนชิพเดียวกันแต่คุณภาพของระบบจะแย่ลงหาก DAC ไม่มีคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การใช้งาน IC 8253

ในการออกแบบวงจร Interface โดยทั่วไป มักจะมีความจำเป็นต้องใช้ฐานเวลาหรือวงจรนับ (Counter) ต่างๆ ในการทำงานของวงจรด้วย ซึ่งจะมี Chip ตัวหนึ่งที่ทำหน้าที่เป็นวงจรฐานเวลา และ Counter โดยสามารถควบคุมได้ Chip Support ตัวนี้เบอร์ 8253 (Programmable Interval Timer) สำหรับ 8253 นี้จะมี Channel ที่ใช้งานเป็นวงจรสร้างฐานเวลาหรือวงจรนับอยู่ 3 Channel คือ CH 0, CH 1, CH 2 โดยที่เราสามารถจะ Program และใช้งาน CH ทั้ง 3 นี้แยกกันได้อย่างอิสระ

1) สายสัญญาณ Clock ,Gate และ Out ของ Counter

Counter แต่ละตัวใน Block Diagram ของรูปที่ 2.18 จะมีสายสัญญาณต่อกันแต่ละ Block อยู่ 3 เส้น โดยสายสัญญาณนี้มีชื่อว่า Clock และ Gate ใช้เป็น I/P ส่วน Out ใช้เป็น O/P หน้าที่ในการทำงานของสายเหล่านี้เปลี่ยนแปลงได้ขึ้นอยู่กับว่า อุปกรณ์เหล่านี้ถูกกำหนดหน้าที่การทำงานเบื้องต้นไว้อย่างไรหรือถูก Program มาอย่างไร

Clock : เป็น I/P ที่ใช้ป้อนสัญญาณ Clock ให้แก่ Counter ซึ่ง Counter ในที่นี้มีขนาด 16 บิท ความถี่ของสัญญาณ CLK ที่มากที่สุดที่ป้อนให้แก่ Counter เป็น 2.6 MHz และความถี่ของ CLK ที่น้อยที่สุดเป็น 0 Hz (DC) หรือ Static Operation

Gate : เป็นสายสัญญาณ I/P ที่ทำตัวเสมือน Gate ที่จะยอมหรือไม่ยอมให้สัญญาณ CLK ผ่านเข้าไปยัง Counter และ Gate สามารถใช้เป็นสายสัญญาณที่ป้อน Pulse เพื่อกระตุ้นให้ Counter เริ่มนับซึ่งขึ้นอยู่กับ Mode ที่ Program ให้กับ Counter

Out : เป็นสายสัญญาณ เอ้าท์พุทของ Counter ซึ่งการทำงานขึ้นอยู่กับการ Program Internal Register ของ 8253

2) Internal Register ของ 8253

ในรูปที่ 2.18 นั้นเป็น Register ภายในของ 8253 ในขั้นต้นนี้จะพิจารณา Mode Word Register ก่อน Register นี้เป็นตัวกำหนดการทำงานทั้งหมดของ 8253 จากที่ทราบมาแล้วว่าการทำงานของแต่ละ Counter ภายใน 8253 เป็นอิสระต่อกันอย่างสมบูรณ์ ทำให้สามารถ Program การทำงานใน Counter แต่ละตัวได้โดยการให้ข้อมูลที่ถูกต้องกับ Mode Word Register ต่อไปนี้ เราจะพิจารณา Register ภายในทั้ง 4 ดังรูปที่ 2.18

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-OPERATE

รูปที่ 2.18 แสดงลอจิกของ Register ภายใน 8253 ที่จะ Program ให้กับ Counter

Control Word Register: เป็น Register ที่ใช้ควบคุม Mode การทำงานและใช้เลือกวิธีการนับของ Counter ว่าจะให้นับแบบ Binary หรือ BCD (Binary Code Decimal) ก่อนที่จะใช้งานจะต้อง Program ข้อมูลให้กับ Register นี้เสียก่อน ซึ่งข้อมูลที่ Program ต่อไปนี้จะเป็นตัวกำหนดลักษณะการทำงานของ Counter Register นี้สามารถเขียนข้อมูลเข้าไปได้อย่างเดียวไม่สามารถอ่านออกมาได้ และจะติดต่อกับ Register นี้ได้เมื่อขา A0 ,A1 มี Logic "1"

Counter #0, #1, #2 :Counter ทั้ง 3 นี้มีลักษณะที่เหมือนกันและทำงานอย่างเป็นอิสระต่อกันและกัน แต่ละ Counter นี้มีขนาด 16 บิท Pre-Settable Down Counter และสามารถนับได้เป็น Binary หรือ BCD ก็ได้ข้อมูลที่อยู่ภายใน Counter เหล่านี้สามารถถูกอ่านโดย Microprocesser ได้โดยไม่ทำให้ข้อมูลภายใน Counter นั้นเสียหาย ซึ่งระบบสามารถจะแสดงค่าใน Counter ได้ตลอดเวลา โดยไม่กระทบกระเทือนการทำงานทั้งหมดของ Counter

3) การ Program 8253 (Control Word Format)

Mode การทำงานของ Counter ทั้งหมด สามารถเลือกได้โดยการเขียน ข้อมูลเข้าไปใน Register ควบคุมซึ่งมีรูปแบบของคำสั่งควบคุม (Control Word Format)ดังรูปที่ 2.19 ทำให้ Address ของคำสั่งควบคุมนี้เป็น Address ที่มีค่าของ A0 ,A1 เป็น Logic "1" ซึ่งในระบบที่เราพิจารณาอยู่นี้ให้ Address ของคำสั่งควบคุมเป็น 303H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก Counter ของ 8253 มีจำนวน 3 ตัวด้วยกันฉะนั้นการ Program Counter นั้น จำเป็นต้องกำหนด Counter ที่ต้องการจะ Program เสียก่อนการกำหนดทำได้โดยให้ Logic ที่ถูกต้องกับ Bit D7 ,D6 ซึ่งมีชื่อว่า SC1 ,SC2 ตามลำดับ ของรูปที่ 2.19 เมื่อได้ Counter ที่ต้องการแล้ว Counter นั้นก็จะถูก Set และจะอยู่ในสภาพนั้น จนกว่าจะมีคำสั่งควบคุมอื่นๆ มาทำให้เปลี่ยนแปลง ส่วนการกำหนดค่า Logic ของ Bit D7 ,D6 สำหรับในการเลือก Counter มีดังนี้

Control Byte D7-D0

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCP

รูปที่ 2.19 แสดงข้อกำหนดของแต่ละบิทของ Register ควบคุม

<u>หมายเหตุ</u> ใน 1 ไบท์ของ Counter จะประกอบด้วย 16 บิท โดยบิท D0-D7 เป็น Least-Signification Byte และ D8-D15 เป็น Most-significant Byte

D7	D6	Counter Select
0	0	0
0	1	1
1	0	2
1	1	ไม่มีความหมาย

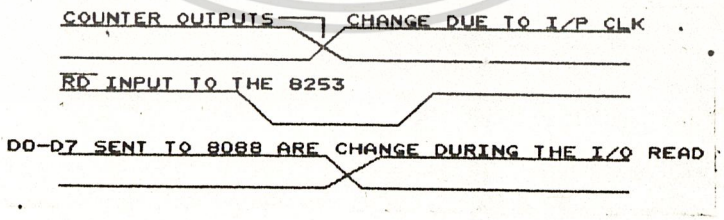
เมื่อเลือก Counter จากการใช้ Bit D7, D6 ได้แล้วต่อไป Bit D5,D4 จะเป็นตัวกำหนดว่า Counter นี้ (หรือ Register) จะใช้ในการ Read/Load Mode ซึ่ง MOde การอ่าน (Read Mode) เป็น Mode ที่ Microprocesser อ่านข้อมูลจาก Counter ส่วน Mode การ LOad (Load Mode) เป็น Mode ที่ Microprocesser เขียนข้อมูลเข้าไปใน Counter Bit D5 และ D4 ถูกกำหนดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D5	D4	R/L Difination
0	0	ค่าใน Counter ถูก Latch หมายความว่าค่าที่มีอยู่ใน Counter ที่ถูกกำหนดนี้ จะนำเข้าไปเก็บไว้ใน F/F ซึ่ง CPU สามารถ อ่านออกไปได้
0	1	Read/Load เฉพาะไบท์ที่มีนัยสำคัญต่ำ (Least-Significant Byte)
1	0	Read/Load เฉพาะไบท์ที่มีนัยสำคัญสูง (Most-significant Byte)
1	1	Read/Load ไบท์ที่มีนัยสำคัญต่ำก่อนเสร็จแล้วตามด้วยไบท์ที่มีนัยสำคัญสูง

เมื่อ D5, D4 มีค่าเป็น 00H Counter จะถูกทำให้อยู่ในโหมดการ Latch ซึ่งเป็นโหมดที่ใช้สำหรับการอ่านค่าของ Counter ขณะที่ Counter ยังทำงานอยู่การเขียนโหมดนี้ให้กับ Register ควบคุม จะทำให้ค่าที่อยู่ใน Counter ถูก Latch ให้กับ Register ภายใน และเมื่อทำการอ่าน Counter ค่านี้จะถูกอ่านออกไป

ถ้าไม่อยู่ในโหมดการ Latch แล้วการอ่านข้อมูลจะเกิดข้อผิดพลาดขึ้นได้เพราะขณะที่ทำการอ่านข้อมูลนั้น ขบวนการที่เกิดขึ้นใน Counter จะทำให้ข้อมูลที่อยู่เดิมเปลี่ยนไป(ดังแสดงใน Timing Diagram ดังรูปที่ 2.20)เป็นผลทำให้ข้อมูลที่ป้อนเข้า CPU เกิดผิดพลาดขึ้นฉะนั้นเพื่อที่จะอ่านค่าของ Counter ให้ถูกต้องในขณะที่ Counter กำลังอยู่ในขบวนการนับอยู่นั้น สามารถทำได้โดยกำหนดคำสั่งควบคุมการ Latchก่อนแล้วจึงให้คำสั่งควบคุมอื่นที่เป็นคำสั่งการอ่านในไบท์ต่อไป Counter ของ 8253 ซึ่งสามารถแก้ไขได้โดยการที่ Micro-processer Latch ข้อมูลจาก เอ๊าท์พุทของ Counter ก่อนที่จะทำการอ่าน



รูปที่ 2.20 เป็น Timing Diagram ที่แสดงการผิดพลาดระหว่างการอ่านข้อมูลจาก เคาน์เตอร์ของ 8253

ยังมีอีก 4บิทที่เหลือของคำสั่งควบคุมในรูปที่ 2.19 คือ D3, D2, D1, D0 แต่จะกล่าวถึง 3 บิทแรกก่อนคือ D3, D2, D1 บิทเหล่านี้เป็นบิทที่กำหนดโหมดการทำงานพื้นฐานของ Counter ซึ่งต่อไปจะได้อธิบายและแสดงตัวอย่างการใช้ Counter ในแต่ละโหมดทั้ง 5โหมด ในที่นี้เรามาดูลอจิกที่ให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D3, D2, D1 ในแต่ละโหมดดังนี้

D3	D2	D1	Mode Value
0	0	0	Mode 0 : Interrupt On Terminal Count
0	0	1	Mode 1 : Programable One-Shot
X	1	0	Mode 2 : Rate Generter
X	1	1	MOde 3 : Square Wave Generter
1	0	0	Mode 4 : Softwere Triggered Strobe
1	0	1	Mode 5 : Hardware Triggered Strobe

บิทสุดท้ายของคำสั่งควบคุมคือ D0 ใช้กำหนดลักษณะการนับของ Counter ว่ามกษณะการนับเป็นอย่างไร นั่นคือจะนับเป็น BCD หรือ Binary ถ้า D0 มีลอจิกเป็น "1" Counter จะนับแบบ BCD ถ้า D0 มีลอจิก เป็น "0" จะนับเป็นแบบ Binary ค่าที่มากที่สุดสำหรับการนับในโหมดการนับแบบ Binary มีค่าเท่ากับ 2^{16} และในโหมดการนับแบบ BCD เป็น 10^4

4) การทำงานในโหมดต่างๆของ 8253

<u>การทำงานใน Mode 0 :Interupt on Terminal control</u>

ลักษณะการทำงานในโหมด 0 ของ 8253 Counter จะนับแบบนับลงเมื่อค่าที่ Counter นับมีค่าเป็น 0 ขา Out ของ Counter จะมีลอจิกเป็น "1" ฉะนั้นในโหมด 0 นี้ Counter จะต้องถูกโปรแกรมค่าเริ่มแรกที่จะนับเสียก่อน จากนั้นก็ทำการนับลงด้วยอัตราเท่ากับความถี่ของ CLK ที่ป้อนเข้ามา เมื่อค่าที่นับมีค่าเป็น 0000H แล้วขา OUT ก็จะมีลอจิกเป็น "1" ซึ่งการใช้งานของขา OUT นี้สามารถนำไปใช้กับการอินเทอร์รัพท์ Microprocesser ได้ ขา OUT จะมีลอจิกเป็น "1" จนกว่าเคาน์เตอร์จะถูกป้อนค่าที่ต้องการนับเข้าไปอีก (อาจเป็นค่าเก่าหรือค่าใหม่ก็ได้) หรือเมื่อมีการเขียนคำสั่งเลือกโหมด เข้าไปให้กับ 8253

เมื่อ Counter เริ่มทำการนับ (นับลง) แล้วเราสามารถหยุดการนับของ Counter ได้ โดยการให้ลอจิก "0" ที่ขาอินพุท GATE ของ Counter ตัวอย่างในการเซ็ท Counter ใน Mode 0 โดยเราจะต้องทำการโปรแกรมให้กับคำสั่งควบคุมโดยให้บิทต่างๆ เป็นไปตามข้อกำหนดที่ได้แสดงดังรูป 5 สำหรับตัวอย่างนี้มีคำสั่งควบคุมเป็น 0011001B ซึ่งแต่ละบิทมีความหมายดังนี้

Bit D7 และ D6 = 00 เป็นการกำหนดให้ใช้ Counter หมายเลข 0

Bit D5 และ D4 = 11 เป็นการเซ็ทให้ Counter ทำการโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	LSB เข้าไปก่อนแล้วตามด้วย MSB
Control Bit D3,D2 และ D1 = 000	เป็นการกำหนดให้ Counter ใช้งานใน MOde 0
D0 = 1	เป็นการกำหนดให้การนับของ Counter เป็นแบบ BCD

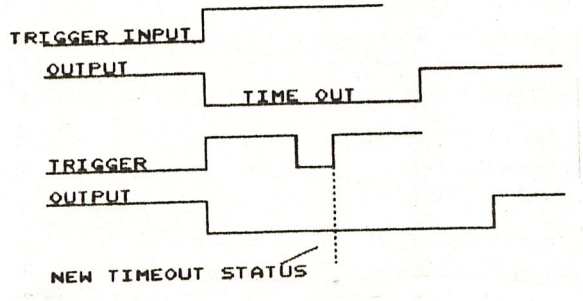
ฉะนั้นคำสั่งควบคุมซึ่งเท่ากับ 0011001B จะถูกเขียนเข้าไปที่พอร์ต 303H ซึ่งเป็นแอดเดรสของ Control Word Register

เมื่อทำการเซ็ทเคาน์เตอร์แล้ว เราจะต้องเขียนค่าเริ่มแรกของการนับให้กับรีจิสเตอร์ของเคาน์เตอร์ ให้มีลักษณะของการนับแบบ BCD โดยให้ LSB มีค่าเป็น 37 (เป็นเลขฐาน 10) และให้ MSB เป็น 01 ทั้งสองไบท์นี้จะถูกเขียนไปยังพอร์ทที่ 300H

ทันทีที่ไบท์ที่สอง (MSB) ได้ป้อนให้กับรีจิสเตอร์ของ Counters หมายเลย 0 การนับก็จะเริ่มขึ้นและเมื่อนับถึง 125 (ก็คือรีจิสเตอรต์ของเคาน์เตอร์มีค่าเป็น 0) ขา OUT ของเคาน์เตอร์ก็จะมีลอจิกเป็น "1" ซึ่งเราอาจจะนำเอาขานี้ไปใช้ในการควบคุมอุปกรณ์ต่างๆ ได้

การทำงานใน Mode 1 :Programmable One-Shot

ในโหมด 1 นี้ 8253 จะทำงานในลักษณะของ One-Shot คือสามารถให้เอาท์พุทในรูปของพัลส์ และความกว้างของพัลส์นี้ มีค่าเป็นจำนวนเท่าที่เป็นเลขจำนวนเต็มของพัลส์ของสัญญาณนาฬิกาที่ป้อนเข้าที่ขา CLOCK การทำงานของ One-Shot จะเกิดขึ้นเมื่อมีสัญญาณขอบขาขึ้น (Rising Edge) เข้ามากระตุ้น (Trigger) ที่ขาอินพุท GATE ทำให้เกิดพัลส์ที่เอาท์พุท แต่ถ้ามีสัญญาณกระตุ้น เข้ามาที่ขา GATE ในขณะที่เอาท์พุทยังทำงานไม่เสร็จสิ้น สัญญาณที่มากระตุ้นใหม่นี้ก็จะทำให้เกิดพัลส์ลูกใหม่ดังแสดงในรูปที่ 2.21



รูปที่ 2.21 แสดงไดอะแกรมเวลาของ 8253 เมื่อโปรแกรมให้ทำงานในแบบ One-Shot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานใน Mode 2 : Rate Generator

ในโหมด 2 นี้ 8253 ถูกใช้เป็นเคาน์เตอร์ที่ทำหน้าที่หารด้วย "N" ทำให้ได้สัญญาณที่ขาเอาท์พุท เป็นลอจิก "0" และ "1" สลับกันไปด้วยอัตราความถี่เท่ากับสัญญาณอินพุทคล็อกหารด้วย "N" ซึ่งทำให้มีช่วงกว้างของลอจิก "1" มีค่าเท่ากับคาบของสัญญาณอินพุทคล็อกคูณอ้วยจำนวนค่า "N" ดังรูปที่ 2.22 ซึ่งค่า "N" นี้เป็นค่าที่ได้จากการโหลดให้กับเคาน์เตอร์ในโหมด 2 แต่ถ้าขณะที่เอาท์พุทของเคาน์เตอร์ยังทำงานอยู่ในช่วงของลอจิก "1" เกิดมีค่า "N" ค่าใหม่โหลดเข้ามาเอ๊าท์พุทของเคาน์เตอร์ก็ยังคงทำงานในลักษณะเดิมจนกระทั่งหมดคาบ แล้วจึงจะทำตามค่า "N" ที่โหลดเข้ามาใหม่

รูปที่ 2.22 แสดงไดอะแกรมเวลาที่เอ๊าท์พุทของ 8253 ในโหมด 2

การทำงานใน Mode 3: Square Wave Generator

การทำงานในโหมด 3 นี้มีลักษณะคล้ายกับโหมด 2 ที่กล่าวมาแล้ว เว้นแต่เอ๊าท์พุทที่ได้เป็น Square Wave ที่มีช่วงกว้างของลอจิก "1" และ "0" สมมาตรกัน (Duty Cycle = 50 %) แต่ถ้าค่า "N" เป็นเลขคี่ เอาท์พุทที่ได้จะมีช่วงกว้างของลอจิก "1" เท่ากับ (N+1)/2 * คาบเวลาของสัญญาณ CLK และช่วงกว้างของลอจิก "0" เท่ากับ (N-1)/2 * คาบเวลาของสัญญาณ CLK

การทำงานใน Mode 4 : Software Triggered Stobe

ในโหมดนี้ ผู้เขียนโปรแกรมสามารถเซ็ทเคาน์เตอร์ให้มีเอาท์พุทหน่วงเวลาออกไปหลังจากที่เคาน์เตอร์เริ่มทำงาน (โดยเริ่มนับค่าที่โหลดให้) เอาท์พุทที่ได้จะมีลักษณะเป็นลอจิก "0" มีคาบเวลาเท่ากับคาบเวลาของ CLK 1 ลูก และจะกลับไปเป็นลอจิก "1" อีก (เอ๊าท์พุทเป็นลอจิก "1" ทันทีเมื่อเราเซ็ทโหมด 4 ให้) เอ๊าท์พุทนี้จะเกิดเมื่อเคาน์เตอร์นับค่าที่เราโหลดให้ลดลง (นิยมนับลง) จนค่าที่ปรากฏที่เคาน์เตอร์เท่ากับ 0 ดังรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้