สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ออสซิลโลสโคปบนเครื่องคอมพิวเตอร์

Oscilloscope On PC

โดย

นาย พิศิษฐ์ รัตตไพศาล รหัส 39014371

นาย เมธา เตชะทรงคุณ รหัส 39014411

อาจารย์ที่ปรึกษา

อาจารย์ โกศล ชวนขยัน

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 36892
วัน, เดือน, ปี 29 ส.ค. 254[illegible]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ออสซิลโลสโคปบนเครื่องคอมพิวเตอร์

ผู้จัดทำ

1. นาย พิศิษฐ์ รัตตไพศาล รหัส 39014371
2. นาย เมธา เตชะทรงคุณ รหัส 39014411

.....อาจารย์ที่ปรึกษา

(อ. โกศล ชวนขยัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานเรื่อง ออสซิลโลสโคปบนเครื่องคอมพิวเตอร์

Oscilloscope On PC

จัดทำโดย นาย พิศิษฐ์ รัตตไพศาล

นาย เมธา เตชะทรงคุณ

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(อ. โกศล ชวนขยัน)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออสซิลโลสโคปบนเครื่องคอมพิวเตอร์

นายพิศิษฐ์ รัตตไพศาล
นายเมธา เตชะทรงคุณ
อ. โกศล ชวนขยัน (อาจารย์ปรึกษา)
ภาคการศึกษาที่ 2 ปีการศึกษา 2542

บทคัดย่อ

ออสซิลโลสโคป ถือได้ว่าเป็นหนึ่งในเครื่องมือวัดที่สำคัญในงานด้านอิเล็กทรอนิกส์ เนื่องจากทำให้ทราบลักษณะรูปร่างของสัญญาณทางไฟฟ้าต่างๆได้ โครงงานออสซิลโลสโคปบนเครื่องคอมพิวเตอร์นี้ก็มีความสามารถหลักๆ เช่นเดียวกันกับเครื่องออสซิลโลสโคปที่ใช้กันอยู่ทั่วไป แต่มีข้อดีคือ การนำคอมพิวเตอร์ส่วนบุคคลที่ใช้กันอย่างแพร่หลายในปัจจุบันมาใช้งานเป็นออสซิลโลสโคปได้ โดยการใส่การ์ดอินเทอร์เฟสที่สร้างขึ้นทางสล๊อต ISA แล้วทำการเขียนโปรแกรมนำสัญญาณอินพุทที่ผ่านเข้ามาทางการ์ดไปวาดเป็นรูปสัญญาณขึ้นมาบนหน้าจอคอมพิวเตอร์ ส่วนประกอบที่จำเป็นของโครงงานชิ้นนี้ได้แก่ ส่วนปรับขนาดและยกระดับสัญญาณอนาลอก , ส่วนแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital Converter : A/D) , ส่วนเก็บข้อมูลลงในหน่วยความจำภายนอก , การ์ดอินเทอร์เฟส (ISA) , และส่วนการเขียนโปรแกรม

ประสิทธิภาพสามารถดูได้จาก waveform ของสัญญาณที่แสดงผลออกมาที่คอมพิวเตอร์ว่ามีความผิดเพี้ยนไปจากสัญญาณจริงหรือไม่ , ขนาดและความถี่สูงสุดของสัญญาณที่สามารถวัดได้โดยไม่ผิดเพี้ยน , และความละเอียดของส่วนแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscilloscope On PC

Mr. Pisit Rattapaisam
Mr. Meta Tachasongkun
Mr. Kosol Chuankayan (Advisor)
2nd Semester , Education Year 1999

Abstract

Oscilloscope is an important instrument in electronic work and other. Because it can use for displaying the signal waveform . Oscilloscope On PC has a basic capability same as an ordinary oscilloscope. Using Personal Computer (PC) as Oscilloscope is an important objective of this project. By adding the interface card through ISA slots and programming the software for receiving the input signal and display this signal on a PC's monitor. The significant parts of this Oscilloscope on PC are amplitude and DC level adjustment part, Analog to Digital Converter part , storage part , interface card (ISA) , and programming part.

The efficiency of this Oscilloscope On PC can be observe from the signal waveform that display on PC , maximum amplitude and frequency of signal that can be measured without error and distortion , resolution of Analog to Digital Converter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	V
บทที่ 1 บทนำ	1
บทที่ 2 หลักการเบื้องต้นของออสซิลโลสโคป	2
2.1 เกริ่นนำ	2
2.2 สัญญาณกับออสซิลโลสโคป	2
2.3 กระบวนการสร้างภาพของออสซิลโลสโคป	4
2.4 การเบี่ยงเบนแนวตั้ง	6
2.5 การเบี่ยนเบนแนวนอน	7
2.6 การปรับแต่งความสว่าง	12
บทที่ 3 ทฤษฎีของวงจรส่วนต่างๆ ที่เกี่ยวข้อง	13
3.1 เกริ่นนำ	13
3.2 ทฤษฎีการ Sampling	14
3.3 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (ADC)	22
3.4 ทฤษฎีของออปแอมป์	29
3.5 ผลตอบสนองทางความถี่ของออปแอมป์	35
บทที่ 4 โครงสร้างของคอมพิวเตอร์และการอินเตอร์เฟส	37
4.1 โครงสร้างพื้นฐานของคอมพิวเตอร์	37
4.2 Memory Map ของเครื่องไอบีเอ็ม พีซี	39
4.3 รายละเอียดของส่วนอินพุท / เอาท์พุท	40
4.4 การกำหนดหมายเลขพอร์ตอินพุท / เอาท์พุท	40
4.5 เมนบอร์ดของเครื่องพีซีตระกูลเอที	42
4.6 สล๊อตหรือช่องต่อสำหรับอินพุท / เอาท์พุท	43
4.7 การอินเตอร์เฟส I/O กับ ISA Bus ผ่านทางสล๊อต	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 5 การออกแบบวงจรส่วนต่างๆ	50
5.1 โครงสร้างและส่วนต่างๆ โดยภาพรวม	50
5.2 ส่วนปรับแต่งขนาดและยกระดับสัญญาณอินพุท	51
5.3 ส่วนแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	53
5.4 ส่วนอ่าน / เขียนข้อมูลใน SRAM และการควบคุม	53
5.5 ส่วนกำเนิดสัญญาณนาฬิกาสำหรับการสุ่ม (Clock generator)	58
5.6 ส่วนเลือกความถี่ของ Clock generator เพื่อใช้ในการสุ่ม	59
5.7 ส่วนควบคุมให้มีการเก็บข้อมูลต่อ เนื่องจาก SRAM ไม่พอ	60
5.8 พอร์ตข้อมูล และวงจรถอดรหัสแอดเดรส	62
บทที่ 6 การเขียนโปรแกรม	65
6.1 บทนำ	65
6.2 บล็อกไดอะแกรมเบื้องต้นในการออกแบบโปรแกรม	65
6.3 การทำความเข้าใจในขอบเขตของงาน	66
6.4 การทำความเข้าใจกับฮาร์ดแวร์	67
6.5 รู้และเข้าใจในจุดเด่นจุดด้อยของภาษาที่เลือกใช้	68
6.6 การวิเคราะห์ตีความเป็น Algorithm	72
บทที่ 7 การวิเคราะห์และตีความหมายเชิง Algorithm	73
7.1 Initial Value Section	74
7.2 Control Process Section	76
7.3 I/O Section	81
7.4 อุปกรณ์ปลายทาง	81
บทที่ 8 ผลการทดลองและสรุปผล	84
8.1 ผลการทดลอง	84
8.2 สรุปผลการทดลองและปัญหาต่างๆ	92
8.3 แนวทางการปรับปรุงแก้ไข	93
8.4 ประโยชน์ที่ได้รับจากการทำโครงงาน	93
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 การวาดสัญญาณของออสซิลโลสโคป	2
รูปที่ 2.2 การวาดสัญญาณของออสซิลโลสโคป (แสดงเส้นสะบัดกลับ)	3
รูปที่ 2.3 การแก้ไขเส้นสะบัดกลับ	3
รูปที่ 2.4 การเบี่ยงเบนของรังสี	4
รูปที่ 2.5 ทางเดินของรังสี	5
รูปที่ 2.6 บล็อกไดอะแกรมของออสซิลโลสโคป	5
รูปที่ 2.7 โครงสร้างการทำงานในแกนแนวตั้ง	6
รูปที่ 2.8 การตอบสนองความถี่ของสัญญาณ	7
รูปที่ 2.9 โครงสร้างการทำงานในแกนแนวนอน	8
รูปที่ 2.10 โครงสร้างและ Timing Diagram ของส่วน Trigger Pulse Generator	9
รูปที่ 2.11 บล็อกไดอะแกรมของ Gate Signal Generator	10
รูปที่ 2.12 Timing Diagram ของ Gate Signal Generator	11
รูปที่ 2.13 การควบคุมความสว่างของจอภาพ	12
รูปที่ 3.1 สัญญาณอนาลอกและจุดของการสุ่ม	14
รูปที่ 3.2 สเปกตรัมของสัญญาณที่ถูกสุ่ม และสเปกตรัมหลังการสุ่ม	15
รูปที่ 3.3 แสดง Error จากการวัดใน Aperture Time	16
รูปที่ 3.4 การสุ่มสัญญาณ	17
รูปที่ 3.5 บล็อกไดอะแกรมของวงจร Sample & Hold	18
รูปที่ 3.6 Timing Diagram แสดงถึง Track & Hold	19
รูปที่ 3.7 ความสัมพันธ์ระหว่างอนาลอกกับดิจิตอล	21
รูปที่ 3.8 บล็อกไดอะแกรมของ Flash Converter	23
รูปที่ 3.9 Timing Diagram โดยปราศจากส่วน Latch	24
รูปที่ 3.10 Timing Diagram โดยที่มีส่วน Latch	25
รูปที่ 3.11 บล็อกไดอะแกรมของ Half Flash Converter	26
รูปที่ 3.12 บล็อกไดอะแกรมของ Multistep Flash Converter	28
รูปที่ 3.13 Equivalent circuit ของออปแอมป์	29
รูปที่ 3.14 การต่อวงจรแบบ Noninverting Amplifier	30
รูปที่ 3.15 การหา Rin และ Rout ของ Noninverting Amplifier	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 3.16 การต่อวงจรแบบ Inverting Amplifier 32
รูปที่ 3.17 การหา Rin ของ Inverting Amplifier 33
รูปที่ 3.18 การหา Rout ของ Inverting Amplifier 34
รูปที่ 3.19 ความสัมพันธ์ระหว่างอัตราขยายแรงดันกับความถี่ 35
รูปที่ 4.1 โครงสร้างพื้นฐานของระบบไมโครคอมพิวเตอร์ 37
รูปที่ 4.2 ส่วนประกอบต่างๆ ของระบบไมโครคอมพิวเตอร์ 38
รูปที่ 4.3 Memory Map ของเครื่องไอบีเอ็ม พีซี 39
รูปที่ 4.4 ไมโครคอมพิวเตอร์กับ I/O Subsystem 40
รูปที่ 4.5 บล็อกไดอะแกรมของเมนบอร์ดตระกูลเอที 42
รูปที่ 4.6 รูปทรงของเมนบอร์ดทั่วๆ ไป และตำแหน่งของส่วนต่างๆ 43
รูปที่ 4.7 ตำแหน่งของขาสัญญาณต่างๆ บนสล็อตของเครื่องตระกูลเอที 44
รูปที่ 4.8 ตัวอย่างการต่อวงจรอินเตอร์เฟสกับพอร์ตเอาท์พุท 48
รูปที่ 4.9 ตัวอย่างการต่อวงจรอินเตอร์เฟสกับพอร์ตอินพุท 49
รูปที่ 5.1 บล็อกไดอะแกรมแสดงโครงสร้างส่วนต่างๆ 50
รูปที่ 5.2 วงจรปรับแต่งขนาดและยกระดับสัญญาณอินพุท 52
รูปที่ 5.3 การต่อไอซี A/D เบอร์ TDA8703 เพื่อใช้งาน 53
รูปที่ 5.4 บล็อกไดอะแกรมแสดงส่วนควบคุมการอ่าน / เขียนลงใน SRAM 54
รูปที่ 5.5 วงจรส่วนควบคุมการอ่าน / เขียนลงใน SRAM 56
รูปที่ 5.6 บล็อกไดอะแกรมแสดงการสร้าง Sampling clock 4 ความถี่ 58
รูปที่ 5.7 วงจรกำเนิดสัญญาณนาฬิกา 4 ความถี่และ Clock selector 59
รูปที่ 5.8 หลักการเลือกความถี่ของ Sampling clock ให้กับ A/D 60
รูปที่ 5.9 ผลของการปรับ Time / Div ให้มีค่าต่างกัน 60
รูปที่ 5.10 บล็อกไดอะแกรมแสดงส่วนควบคุมการเก็บข้อมูลต่อ 61
รูปที่ 5.11 วงจรควบคุมการเก็บข้อมูลต่อลงใน SRAM 62
รูปที่ 5.12 การออกแบบพอร์ตข้อมูลโดยใช้ 8255 63
รูปที่ 5.13 พอร์ตเอาท์พุทของสัญญาณ Select ที่ถูกส่งไปยัง Analog switch 64
รูปที่ 6.1 บล็อกไดอะแกรมของการออกแบบโปรแกรมเบื้องต้น 65
รูปที่ 7.1 บล็อกไดอะแกรมโครงสร้างหลักๆ ในการวิเคราะห์ Algorithm 73
รูปที่ 7.2 โครงสร้างพื้นฐานของภาษา C/C++ 74
รูปที่ 7.3 ตารางแสดงคุณสมบัติของตัวแปรแต่ละชนิด 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 7.4 Flow chart ของส่วนโปรแกรม 82

รูปที่ 8.1 สัญญาณรูปซายน์ขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV) 84

รูปที่ 8.2 สัญญาณรูปซายน์ขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV) 85

รูปที่ 8.3 สัญญาณรูปซายน์ที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV) 85

รูปที่ 8.4 สัญญาณรูปซายน์ที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV) 86

รูปที่ 8.5 สัญญาณรูปซายน์ที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV) 86

รูปที่ 8.6 สัญญาณรูปสามเหลี่ยมขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV) 87

รูปที่ 8.7 สัญญาณรูปสามเหลี่ยมขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV) 87

รูปที่ 8.8 สัญญาณรูปสามเหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV) 88

รูปที่ 8.9 สัญญาณรูปสามเหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV) 88

รูปที่ 8.10 สัญญาณรูปสามเหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV) 89

รูปที่ 8.11 สัญญาณรูปสี่เหลี่ยมขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV) 89

รูปที่ 8.12 สัญญาณรูปสี่เหลี่ยมขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV) 90

รูปที่ 8.13 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV) 90

รูปที่ 8.14 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV) 91

รูปที่ 8.15 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV) 91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในงานทางด้านวิศวกรรมไฟฟ้าสาขาต่างๆ หรือสาขาอิเล็กทรอนิกส์นั้น เครื่องมือวัดต่างๆ ถือเป็นอุปกรณ์ที่สำคัญ โดยเฉพาะออสซิลโลสโคปซึ่งถือได้ว่าเป็นหนึ่งในเครื่องมือวัดที่สำคัญในงานด้านอิเล็กทรอนิกส์เป็นอย่างมาก เนื่องจากทำให้ทราบลักษณะรูปร่าง (Waveform) ของสัญญาณทางไฟฟ้าต่างๆ ได้ และในปัจจุบันเทคโนโลยีต่างๆ มีความก้าวหน้าไปมาก โดยเฉพาะเทคโนโลยีทางด้านคอมพิวเตอร์ที่มีประสิทธิภาพสูงขึ้น จึงมีการพยายามคิดหาวิธีที่จะนำเอาความสามารถของคอมพิวเตอร์ที่มีประสิทธิภาพสูงนี้มาใช้ให้เกิดประโยชน์กับงานทางด้านวิศวกรรมต่างๆ ได้ ดังจะเห็นได้ในปัจจุบันที่มีอุปกรณ์ต่างๆ มากมายที่ทำงานร่วมกับคอมพิวเตอร์ โครงงานออสซิลโลส – โคปบนเครื่องคอมพิวเตอร์ชิ้นนี้ก็เช่นเดียวกัน ซึ่งมีความสามารถหลักๆ เช่นเดียวกันกับเครื่องออสซิลโลสโคปที่ใช้กันอยู่ทั่วไป แต่มีข้อดีคือ การนำคอมพิวเตอร์ส่วนบุคคลที่ใช้กันอย่างแพร่หลายในปัจจุบันมาใช้งานเป็นออสซิลโลสโคปได้ โดยการใส่การ์ดอินเตอร์เฟสที่สร้างขึ้นผ่านทางสล๊อต ISA (Industry Standard Architecture) แล้วทำการเขียนโปรแกรมนำสัญญาณอินพุทที่ผ่านเข้ามาทางการ์ดไปวาดเป็นรูปสัญญาณขึ้นมาบนหน้าจอคอมพิวเตอร์ โดยทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าในการคิดค้นสร้างผลงานชิ้นนี้ขึ้นมาจะเป็นประโยชน์ หรือสร้างแนวคิดใหม่ได้ไม่มากก็น้อย หากพบข้อผิดพลาดที่เกิดขึ้นกับโครงงานชิ้นนี้ ทางคณะผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย แล้วยินดีที่จะทำการปรับปรุงแก้ไขในโอกาสต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

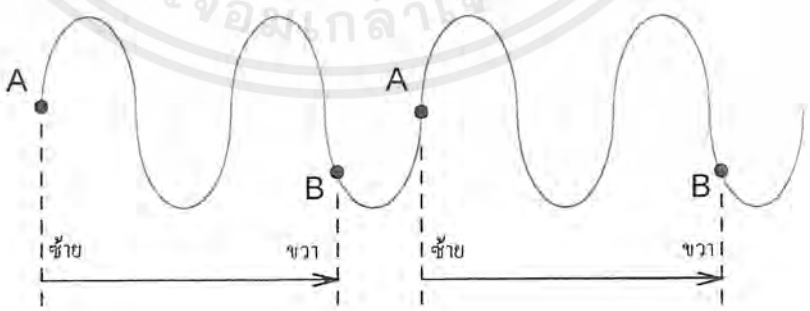
บทที่ 2
หลักการเบื้องต้นของออสซิลโลสโคป
(Introduction to Oscilloscope)

2.1) เกริ่นนำ

Oscillscope มีรากศัพท์มาจากคำว่า **Oscillare** และ **Skopein** ผสมกัน ในส่วนแรก Oscillare มาจากภาษาลาติน หากแปลตามรากศัพท์จะมีความหมายว่า การห้อยโหนบนต้นไม้ การเคลื่อนไหวไปได้ง่ายตามสายลม ส่วนที่สอง Skopein หมายถึง การสังเกต, การทดสอบ คำคำนี้ภายหลังได้มีการเรียกใหม่เป็น **Scopium** เมื่อรวมความหมายทั้งหมดแล้วตีความอาจให้คำจำกัดความของคำว่า Oscillscope ได้คือ เป็นเครื่องมือชนิดหนึ่งที่สามารถวัดและสังเกตลักษณะของสัญญาณต่าง ๆ ได้

2.2) สัญญาณกับออสซิลโลสโคป

ออสซิลโลสโคปสามารถแสดงภาพรูปคลื่นต่าง ๆ ได้โดยการผสมผสานระหว่างสามองค์ประกอบด้วยกัน คือ เวลา (แกน X) , ระดับแรงดัน (แกน Y) , ความสว่าง (แกน Z) ออสซิลโลสโคปจะทำการวาดภาพตามลักษณะสัญญาณที่เข้ามา 3 ปัจจัยที่กล่าวมานี้ แต่เนื่องจากจอภาพที่ใช้มีขนาดจำกัดที่แน่นอน ดังนั้นในการวาดภาพของ ออสซิลโลสโคป นั้นต้องมีจุดสิ้นสุดในการวาดแล้วจึงทำการวาดใหม่



รูปที่ 2.1 การวาดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการวาดรูปใหม่นั้นจะเกิดปัญหาเกิดขึ้น กล่าวคือในขณะที่ทำการวาดภาพย้อนกลับมายังจุดเดิมทางซ้ายนั้นมือนั้น เป็นการวาดภาพจากขวาไปซ้าย ทำให้มีเส้นจาก B -> A ทับกับช่วง A -> B ทำให้เกิดเส้นดังรูป

รูปที่ 2.2 การวาดสัญญาณของสโคป (แสดงเส้นสะบัดกลับ)

การแก้ไขก็คือต้องทำให้ช่วงเวลาของ B->A มีความสว่างเป็นศูนย์ ก็จะได้ดังรูป

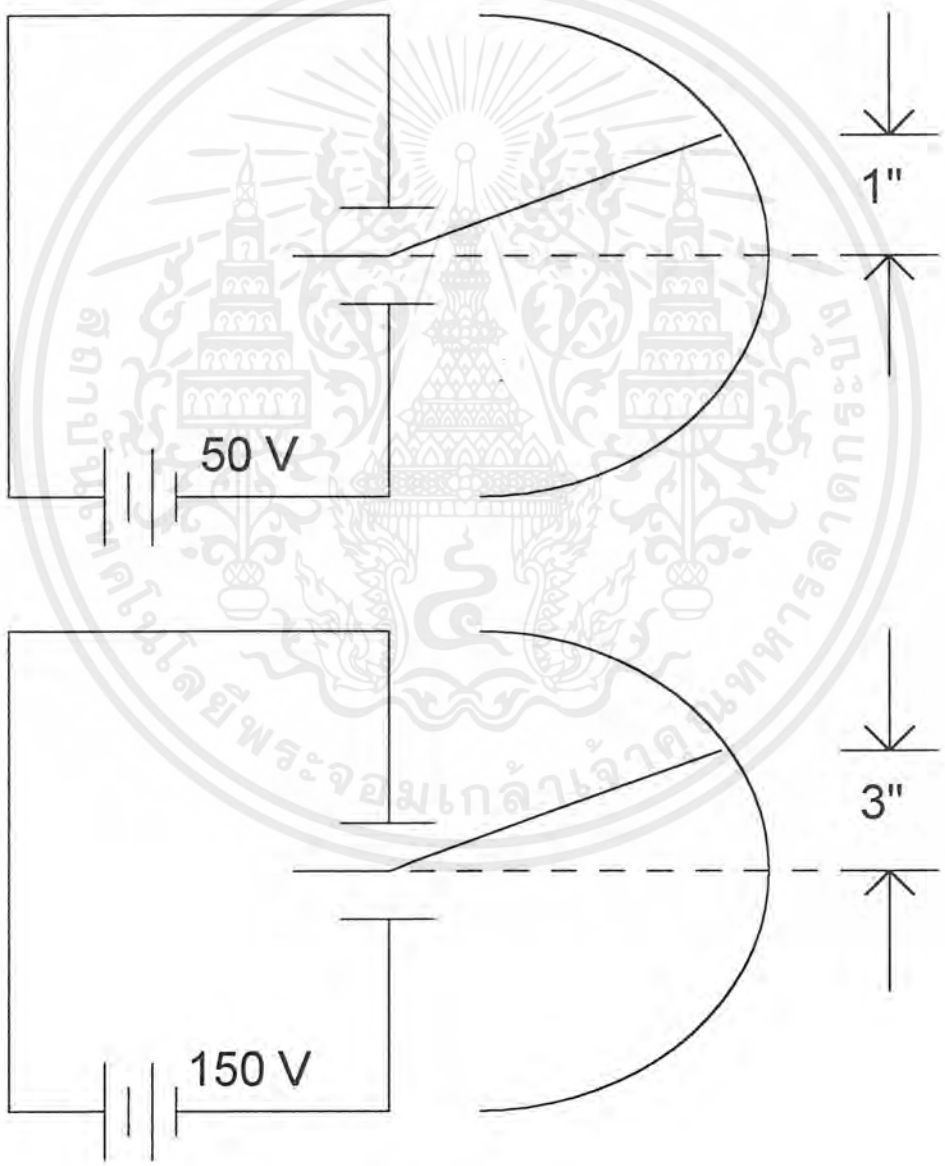
รูปที่ 2.3 การแก้ไขเส้นสะบัดกลับ

แต่อย่างไรก็ตามในการวาดภาพอาจจะไม่ได้รูปกราฟที่นิ่งได้ ทั้งนี้จากปัญหาเรื่องสัญญาณซิงค์อีก (Synchronize) ซึ่งจะกล่าวอีกทีในช่วงหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

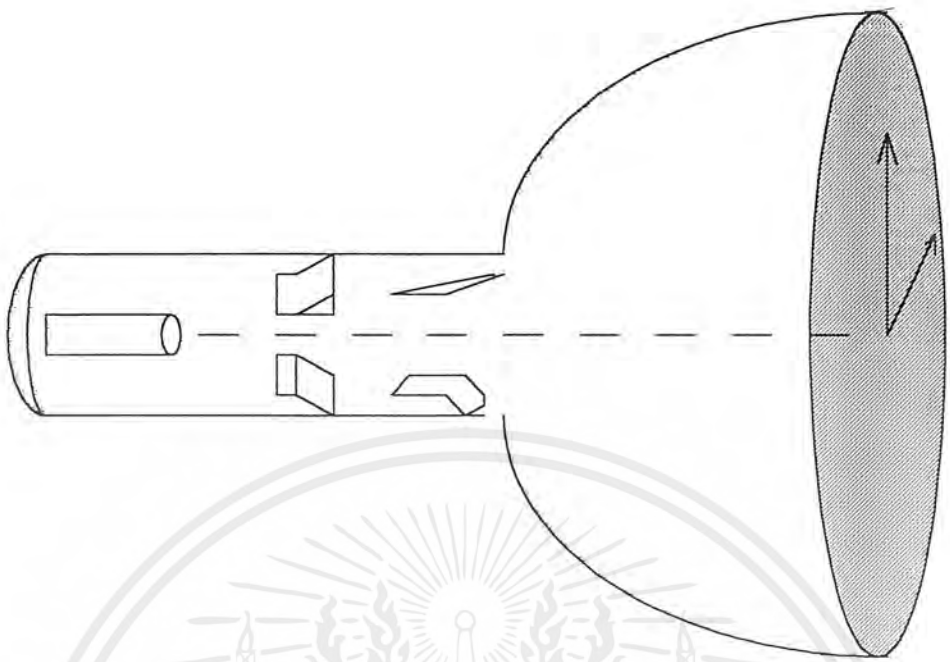
2.3) กระบวนการสร้างภาพของ ออสซิลโลสโคป

หลักการก็คือจะใช้ Cathode Ray เป็นตัวสร้างภาพ โดยอาศัย Deflection plates เป็นตัวเบี่ยงเบนเส้นทางเดินของรังสีซึ่ง Deflection plates จะทำการสร้างสนามไฟฟ้ามาหักเหขนาดของสนามไฟฟ้านั้น จะถูกควบคุมจากความแรงของแรงดันที่ได้รับการขยายมาจากสัญญาณเดิมแล้ว หากเป็น Deflection plates ในแกน X ส่วนที่ทำหน้าที่ขยายสัญญาณก็คือ Horizontal Amplifier และ Deflection plates ในแกน Y ก็คือ Vertical Amplifier

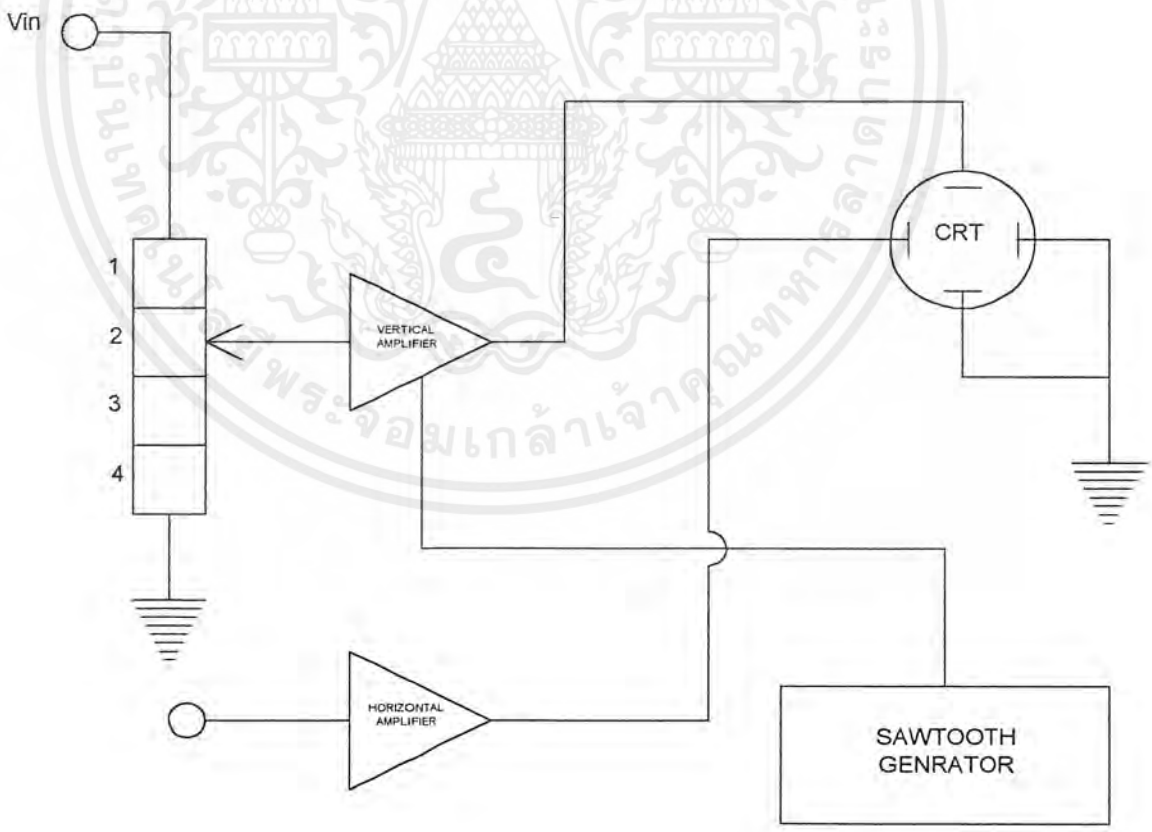


รูปที่ 2.4 การเบี่ยงเบนของรังสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ทางเดินของรังสี

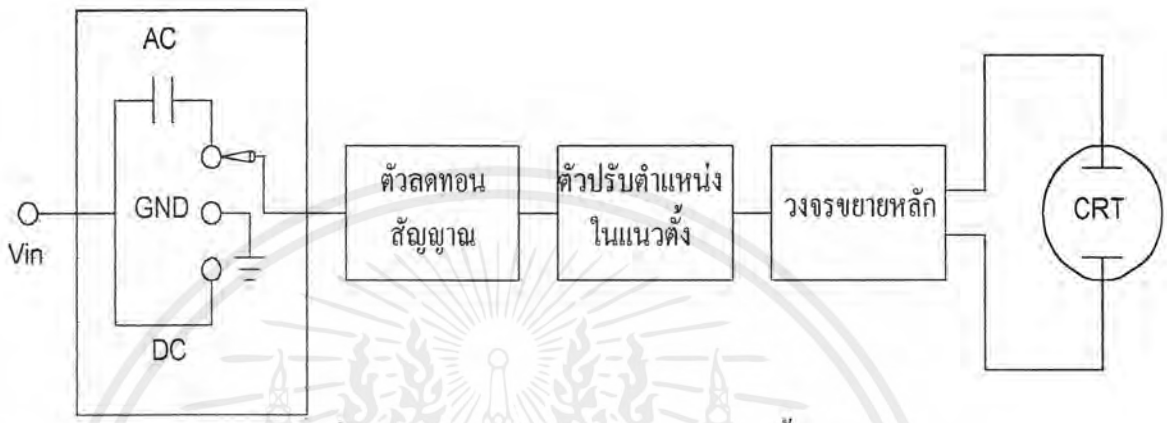


รูปที่ 2.6 บล็อกไดอะแกรมของสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4) การเบี่ยงเบนในแนวตั้ง (Vertical Deflection System)

เป็นการทำงานที่ตรวจจับความแรงของสัญญาณที่เข้ามา (Amplitude) หรือก็คือในแกน Y นั่นเองโดยมีหลักการทำงานดังรูป



รูปที่ 2.7 โครงสร้างการทำงานในแกนแนวตั้ง

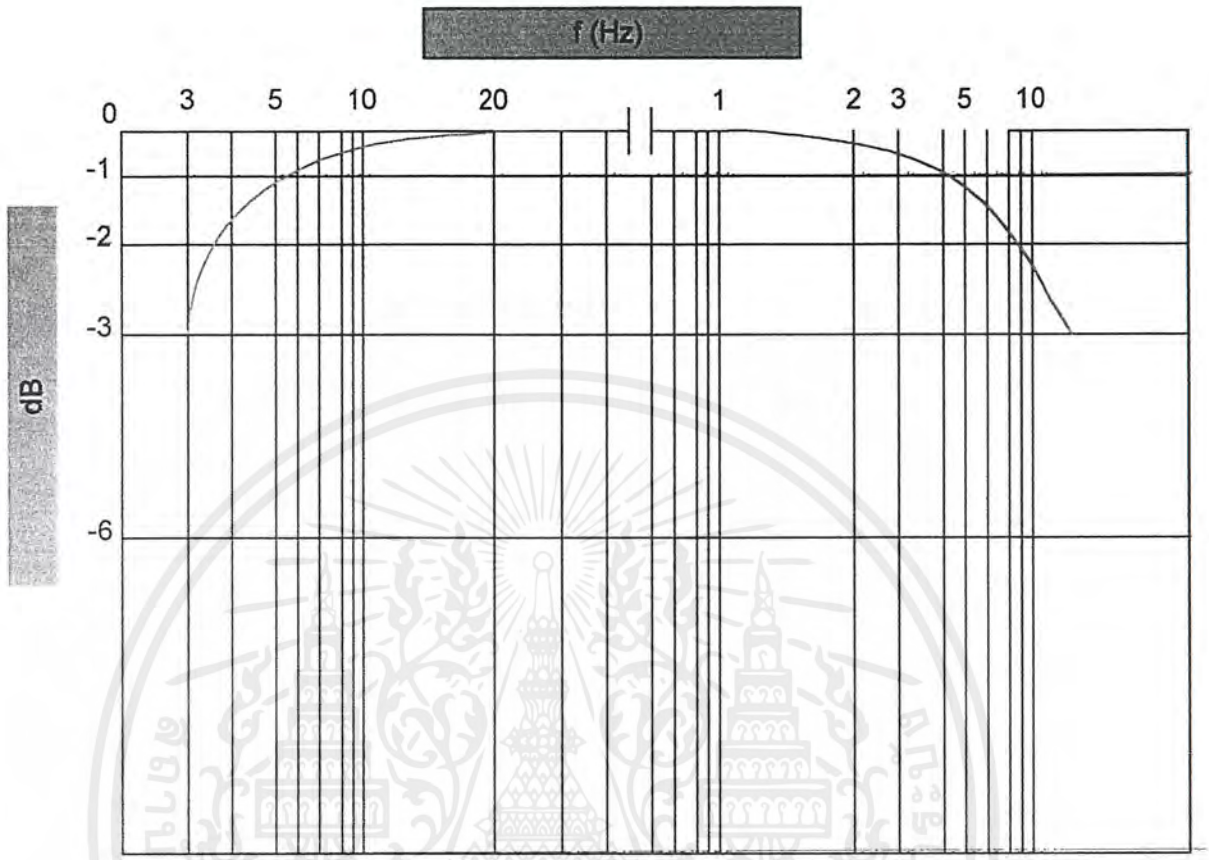
การทำงาน

สัญญาณที่ผ่านสวิตซ์เลือกสัญญาณจะผ่านไปยังตัวลดทอนสัญญาณ ซึ่งตัวลดสัญญาณจะทำการลดทอนสัญญาณจนได้สัญญาณขนาดเล็กค่าหนึ่ง กล่าวคือไม่ว่าสัญญาณที่เข้ามาจะมีขนาดใหญ่เพียงใด สัญญาณที่ได้ก็จะได้ขนาดเล็กขนาดหนึ่งเท่านั้น โดยที่ตัวลดสัญญาณจะมีความต้านทานที่ลดหลั่นกันไปตามอัตราส่วนของสัญญาณ (Step attenuator) สัญญาณที่ได้รับการลดทอนแล้วจะถูกปรับแต่งตำแหน่งของการวาดโดยตัวปรับตำแหน่ง แล้วส่งผ่านไปยังวงจรขยาย (Vertical Amplifier) หลังจากนั้นก็นำสัญญาณที่ขยายได้ดังกล่าวไปควบคุม Deflection plates ต่อไป

ข้อจำกัดของ Vertical Amplifiers

จากที่ได้กล่าวมาแล้วว่า Vertical Amplifiers เป็นวงจรขยายคงที่ แต่ลักษณะดังกล่าวนี้จะไม่เป็นจริงเมื่อใช้งานในย่านความถี่สูง เนื่องจาก Vertical Amplifiers จะประพฤติตัวเป็น Low Pass Filter แทน ดังนั้นที่ความถี่สูงจะได้กราฟในแกนแนวตั้งที่ต่ำกว่าที่ควรจะเป็น ด้วยสาเหตุนี้เองจึงเป็นตัวจำกัดความสามารถของ ออสซิลโลสโคป ทำให้ในการเลือกใช้เครื่องออสซิลโลสโคปต้องดูลักษณะรุ่นที่ใช้ด้วย เพราะแต่ละเครื่องจะถูกออกแบบให้ใช้อย่างถูกต้องที่ความถี่ไม่เกินค่า ๆ หนึ่ง โดยหากเราทำการวัดความถี่สูงสุดของแถบความสัญญาณที่เครื่องระบุมา เราจะได้กราฟที่เล็กกว่าที่ควรจะเป็นอยู่ 3 dB หากต้องการความแม่นยำ ช่วงความถี่สูงสุดของสัญญาณที่จะวัดได้ควรมีค่าประมาณ 1 ใน 3 ของแถบความถี่ของออสซิลโลสโคป หรือกล่าวกลับกันก็คือ หากต้องการวัดความถี่ f ให้ได้แม่นยำ จะต้องใช้ออสซิลโลสโคปที่มีช่วงความถี่ตั้งแต่ 3f ขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

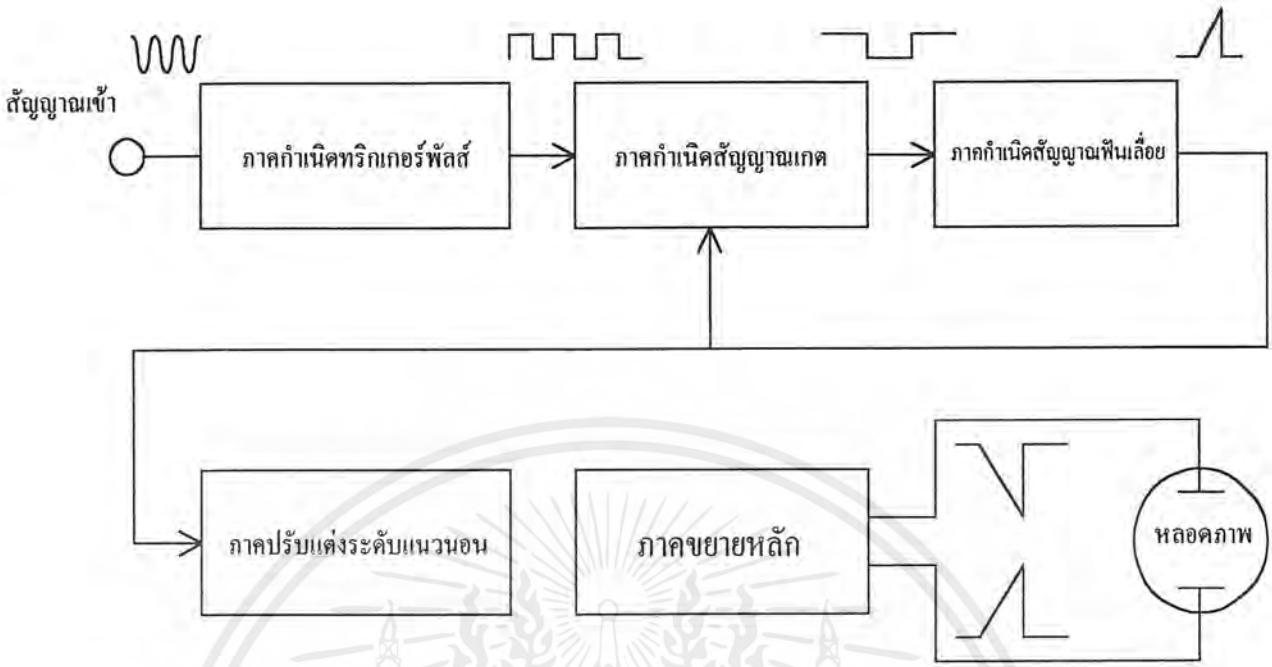


รูปที่ 2.8 การตอบสนองความถี่ของสัญญาณ

2.5) การเบี่ยงเบนในแนวนอน (Horizontal Deflection System)

เป็นการทำงานที่ตรวจจับเวลาของสัญญาณ (Time) หรือก็คือแกน X นั่นเอง เนื่องมาจากเหตุผลในข้อ 2.2 ทำให้ในส่วนนี้ต้องมีสัญญาณควบคุม หรือ สัญญาณซิงค์ (Synchronize) คอยควบคุมการทำงาน โดยมี Block Diagram ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



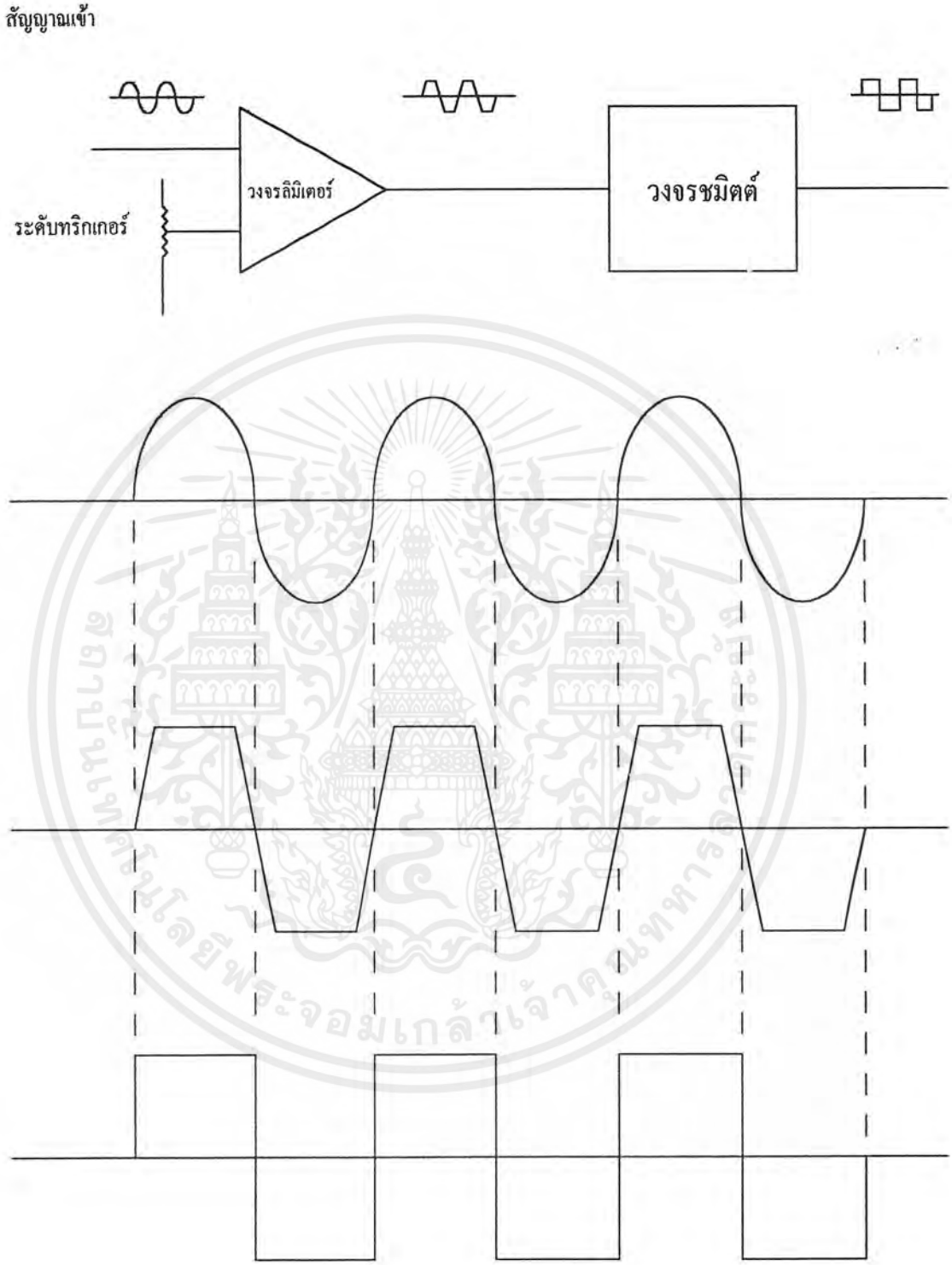
รูปที่ 2.9 บล็อกไดอะแกรม การทำงานในแกนแนวนอน

ที่ Deflection plates ในแนวนอน จะมีสัญญาณ Sawtooth ป้อนเข้ามา สัญญาณ Sawtooth นี้จะถูกทริกเกอร์ (Trigger) หรือกระตุ้นให้เข้าจังหวะกับสัญญาณเข้า (สัญญาณที่จะวัด) โดยช่วง A->B เป็นช่วงกวาดภาพ (Sweep time) และช่วง B->A เป็นช่วง Retrace ซึ่งในช่วงนี้จะไม่มีการวาดภาพแต่อย่างใด

2.5.1) Trigger Pulse Generator

เป็นวงจรกำเนิดทริกเกอร์พัลส์ หรือที่เรียกว่า วงจร Limitter วงจรส่วนนี้ประกอบขึ้นมาจากวงจร Comparator และ วงจร Schmitt การทำงานก็คือ เมื่อสัญญาณทริกเกอร์เข้ามาที่วงจรลิมิตเตอร์ สัญญาณที่ผ่านออกมาจะถูกตัดส่วนบนและส่วนล่างออกตามค่าที่กำหนดไว้ค่าหนึ่ง ค่าแรงดันที่กำหนดขอบเขตในการตัดสัญญาณ เรียกว่า ระดับทริกเกอร์ (Trigger Level) อย่างไรก็ตามสัญญาณออกที่ได้จากวงจรลิมิตเตอร์ยังไม่เหมาะสมเนื่องจากมี Duty Cycle ที่ไม่สูงพอจึงต้องทำการแต่งรูปสัญญาณใหม่ โดยใช้วงจร Schmitt With Hysteresis ก็จะได้สัญญาณตามต้องการ โดยสัญญาณที่ได้เรียกว่า Trigger Pluseการที่ต้องมี Hysteresis อยู่ด้วย ก็เพื่อป้องกันไม่ให้เกิดความผิดพลาดเนื่องมาจากผลของสัญญาณรบกวน (Noise) ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

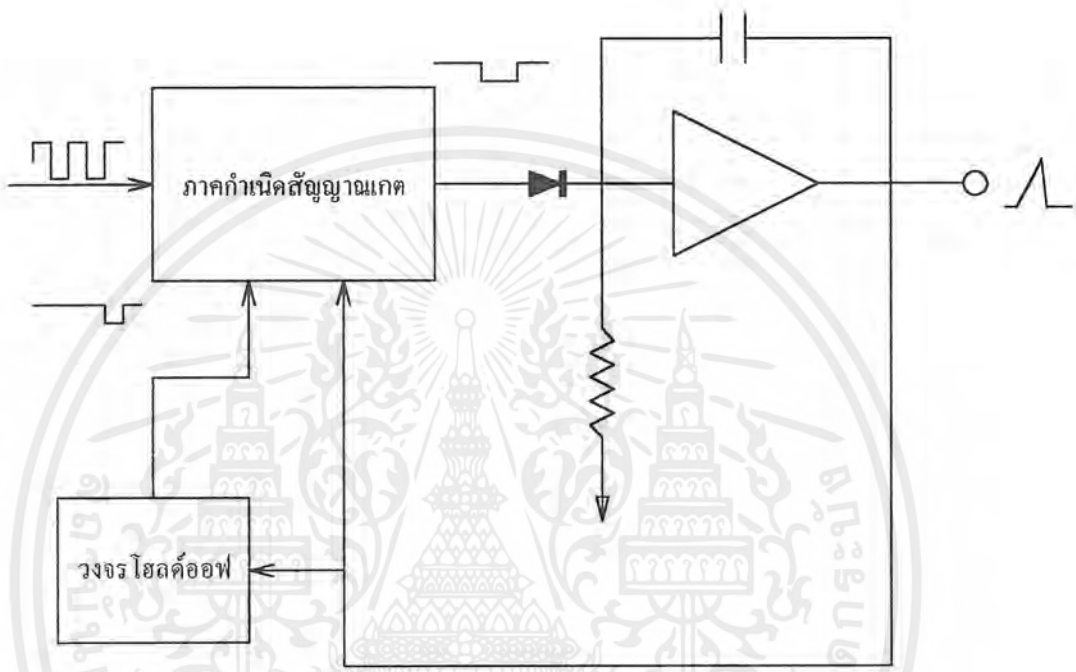


รูปที่ 2.10 โครงสร้างและ Timing Diagram ของส่วน Trigger Pulse Generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2) Gate Signal Generator

เมื่อได้สัญญาณทริกเกอร์พัลส์แล้วสัญญาณส่วนนี้จะไปสร้างสัญญาณเกทเพื่อกวาดภาพ วงจรเกทส่วนนี้จะต่อเข้าเป็นในลักษณะของลูป (Loop) โดยจะต่อกับวงจร Miller Integrator ดังรูป

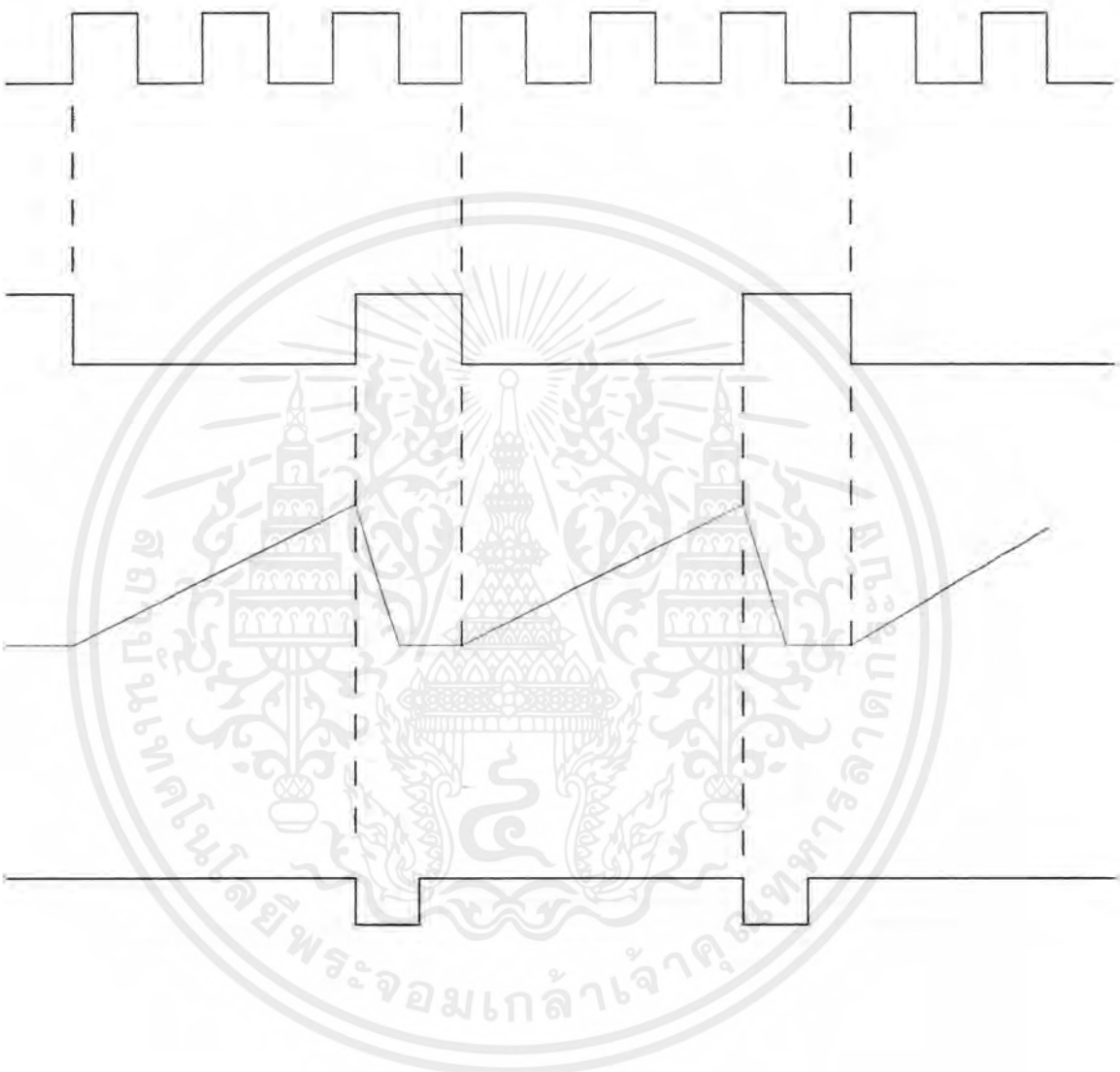


รูปที่ 2.11 บล็อกไดอะแกรม Gate Signal Generator

เมื่อมีสัญญาณทริกเกอร์พัลส์เข้ามาที่ภาคกำเนิดสัญญาณเกท สัญญาณออกของวงจรเกตจะเป็น 0 ทำให้ไดโอด D ถูก Reverse Bias ทำให้ไม่นำกระแส ทำให้ตัวเก็บประจุ C ถูกประจุด้วยค่าเวลาคงที่ค่าหนึ่ง (RC Time Constant) ขณะเดียวกันวงจรเข้าของภาคกำเนิดสัญญาณเกตก็จะปิดและไม่รับทริกเกอร์พัลส์เข้ามาอีก ด้วยเหตุนี้สัญญาณออกของวงจรมิลเลอร์จะค่อย ๆ เพิ่มสูงขึ้นจนถึงระดับแรงดันที่ทำให้จุดบนจอภาพเคลื่อนมายังด้านขวาสุด แรงดันค่านี้จะถูกป้อนกลับ (FeedBack) มายังที่ภาคกำเนิดสัญญาณเกท และจะทำการกลับค่าสัญญาณออกของวงจรภาคนี้ ดังนั้นไดโอด D ก็จะถูก Forward Bias แทน ทำให้การนำกระแสเกิดขึ้น เมื่อนำกระแส ประจุที่เก็บไว้ที่ตัวเก็บประจุ C ก็จะคายออกมาอย่างฉับพลัน จนกระทั่งแรงดันลดลงเท่ากับแรงดันที่ทำให้จุดบนจอภาพเคลื่อนมาอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางซ้ายสุดของจอภาพ เมื่อคายประจุเสร็จสิ้นลงแล้ว วงจรขาเข้าของภาคกำเนิดสัญญาณเกทก็จะเปิดออก ทำให้เกิดการรับสัญญาณทริกเกอร์พัลส์ใหม่อีกครั้ง ดังรูป



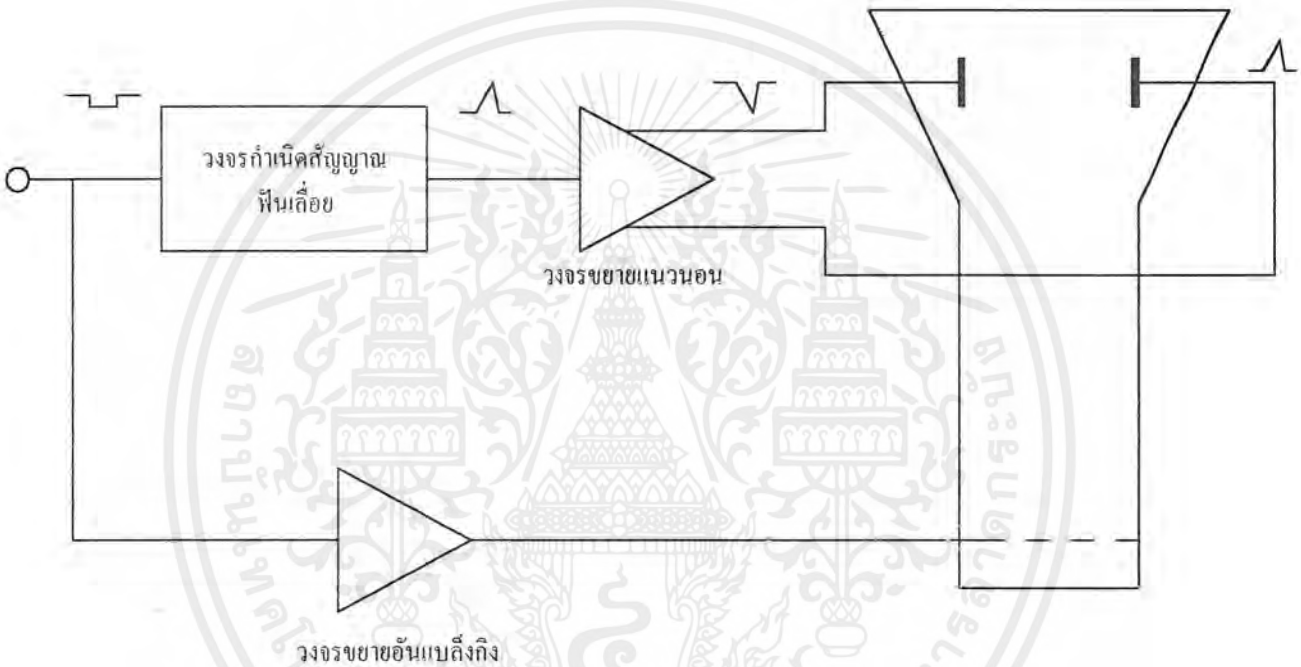
รูปที่ 2.12 Timing Diagram ของ Gate Signal Generator

เมื่อได้สัญญาณ Sawtooth แล้วก็จะส่งต่อให้กับภาคปรับตำแหน่งในแนวนอน ตามด้วยวงจรขยาย Horizontal Amplifier สัญญาณสุดท้ายที่ได้จะนำไปควบคุมการทำงานของ Deflection plates

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6) การปรับแต่งความสว่าง

การจะปรับความสว่างของรูปคลื่นสัญญาณ ทำได้โดยการเปลี่ยนแปลงค่าแรงดันไบอัส (Bias Voltage) ที่ให้แก่กริดของหลอดภาพ นอกจากนี้ยังมีสัญญาณที่ควบคุมให้เกิดส่วนสว่างบนจอภาพโดยสัมพันธ์กับการกวาดภาพ การควบคุมเช่นนี้เรียกว่า UnBlaking และสัญญาณที่ใช้ก็คือสัญญาณเกทในการกวาดภาพนั่นเอง



รูปที่ 2.13 การควบคุมความสว่าง

การทำงานของวงจรควบคุมความสว่างจะเป็นดังรูป กล่าวคือสัญญาณเกทในการกวาดภาพจะถูกป้อนเข้าที่วงจรขยาย UnBlaking สัญญาณออกซึ่งถูกกลับเฟสจะถูกป้อนเข้าสู่กริทของหลอดภาพ การเปลี่ยนระดับสัญญาณออกของวงจรจะช่วยปรับแต่งความสว่างบนจอหลอดภาพได้ กล่าวคือในช่วงที่สัญญาณเกทเปิดรับทริกเกอร์พัลส์ วงจร UnBlaking Amplifiers จะทำให้เกิดจุดสว่างบนจอภาพตามการเปลี่ยนแปลงของสัญญาณฟันเลื่อย (Sawtooth) และสัญญาณเข้า แต่เมื่อสัญญาณเปิดรับทริกเกอร์พัลส์อันเป็นช่วงที่การกวาดภาพกลับมาที่จุดเริ่มต้นใหม่ วงจร UnBlaking Amplifiers จะทำให้ไม่เกิดจุดสว่างซึ่งช่วยให้ไม่มีเส้นสว่างปรากฏซ้อนทับกับรูปคลื่นสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีของวงจรส่วนต่างๆที่เกี่ยวข้อง

3.1) เกริ่นนำ

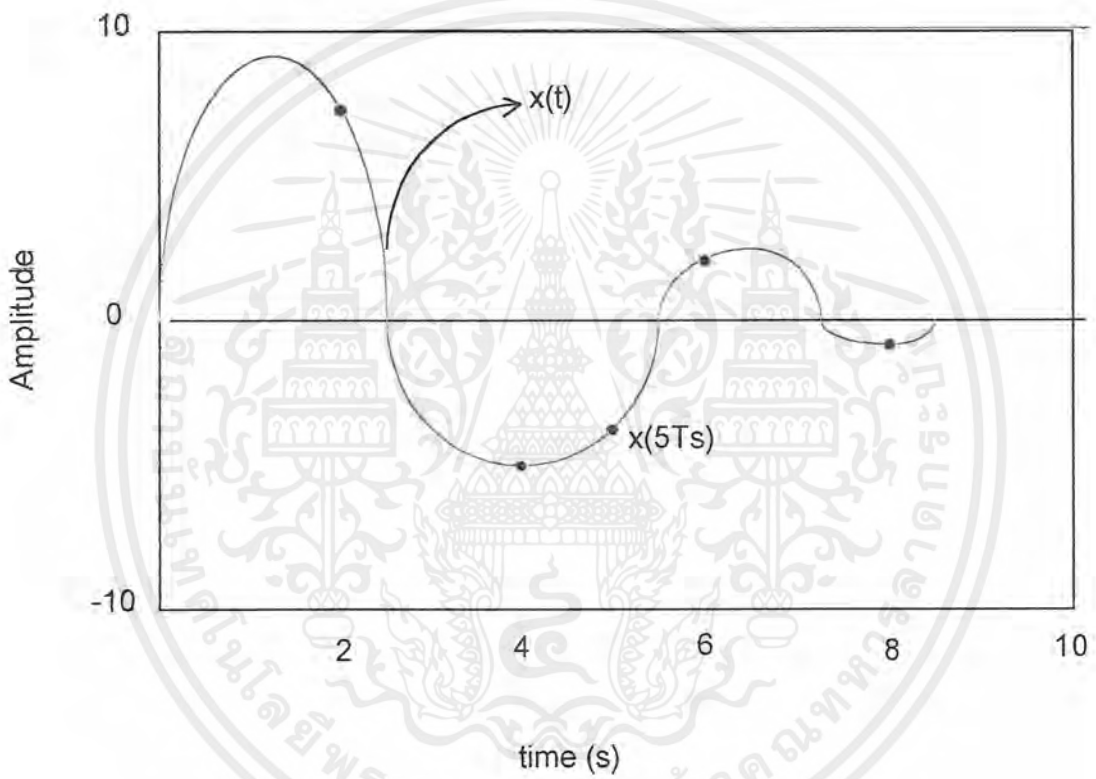
รูปแบบสัญญาณไฟฟ้าที่เราพบเห็นและคุ้นเคยในชีวิตประจำวันจะอยู่ในรูปของสัญญาณที่ต่อเนื่องหรือสัญญาณอนาลอก ซึ่งแต่เดิมการนำเอาสัญญาณอนาลอกมาประมวลผลเพื่อให้มีรูปแบบที่เหมาะสมจะกระทำในแบบอนาลอก แต่เมื่อเทคนิคและอุปกรณ์การประมวลสัญญาณทางดิจิตอลได้รับการพัฒนาขึ้นมา พบว่าการประมวลผลโดยสัญญาณดิจิตอลทำได้ง่ายและมีประสิทธิภาพมากกว่า และเนื่องจากในโครงงานชิ้นนี้จะต้องมีการเชื่อมต่อหรืออินเตอร์เฟส (Interface) เข้ากับคอมพิวเตอร์ ซึ่งมีการทำงานเป็นแบบดิจิตอล แต่สัญญาณที่เราต้องการจะวัดนั้นเป็นสัญญาณอนาลอก ดังนั้นการที่จะทำให้คอมพิวเตอร์รับสัญญาณอนาลอกจากภายนอกมาแสดงผลได้นั้น จะต้องทำการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลเสียก่อน (Analog to Digital Conversion : A/D) แต่ก่อนที่จะกล่าวถึงรายละเอียดของ A/D นั้น จะขอกล่าวถึงทฤษฎีการ Sampling ก่อนเพราะมีความเกี่ยวข้องกันกับส่วน A/D ด้วยซึ่งจะได้กล่าวถึงต่อไป

นอกจากทฤษฎีการ Sampling และวงจร A/D ซึ่งมีความสำคัญต่อการอินเตอร์เฟสกับคอมพิวเตอร์แล้ว ทฤษฎีที่จำเป็นต้องใช้และเกี่ยวข้องอีกทฤษฎีหนึ่งก็คือ ทฤษฎีของออปแอมป์ (Operational Amplifier) ซึ่งจะถูกนำมาใช้ในส่วนของวงจรอนาลอกในโครงงานชิ้นนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) ทฤษฎีการ Sampling

เนื่องจากข้อมูลในสัญญาณแบบ Analog เป็นข้อมูลต่อเนื่อง (Continuous) ส่วนสัญญาณแบบ Digital เป็นข้อมูลแบบไม่ต่อเนื่อง (Discrete) เราจึงต้องทำการสุ่มตัวอย่างมา (Sampling) ทำให้ในการแปลงจาก Analog มาเป็น Digital จะมีข้อมูลบางช่วงขาดหายไป กล่าวคือนำมาไม่หมดนั่นเอง ดังนั้นกระบวนการได้มาซึ่งข้อมูล Digital จึงมีความสำคัญมาก ทำอย่างไรที่ถึงแม้นำข้อมูลมาไม่หมดก็ไม่ทำให้ข้อมูลทั้งหมดเสียความหมายไปโดยรวม



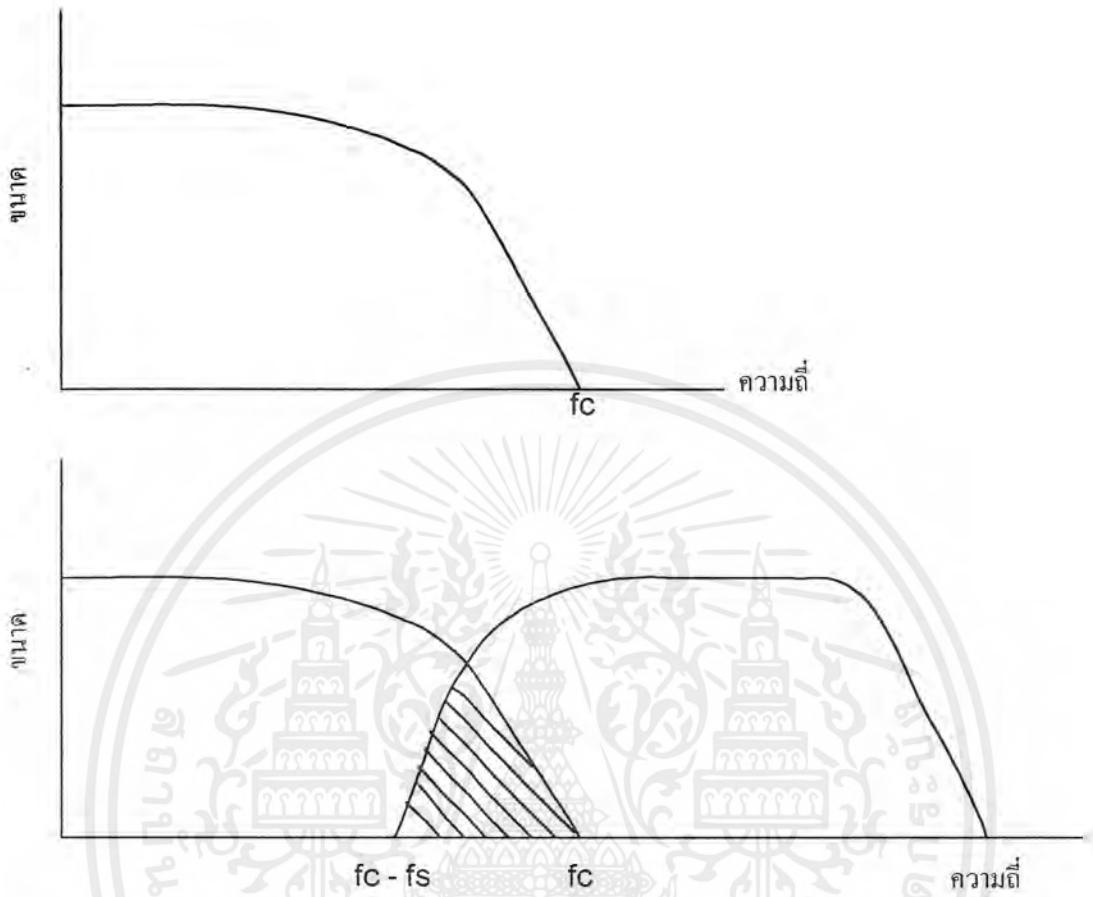
รูปที่ 3.1 สัญญาณอนาลอกและจุดของการสุ่ม

จากรูปเป็นตัวอย่างสัญญาณคาบ x(t) ตรงส่วนจุดวงกลมเป็นจุดที่สัญญาณได้รับการสุ่ม (Sampling) ที่จุด x(5Ts) หมายถึงสัญญาณได้รับการสุ่มเป็นครั้งที่ 5 จำนวนของการสุ่มนั้นจะถูกจำกัดที่จำนวนบิต ของ ข้อมูล Digital ที่เราต้องการแปลงจะถูกจำกัดอยู่ที่ 2^n โดยที่ n คือจำนวนบิต

3.2.1) Sampling frequency limit

จากทฤษฎีของการสุ่มสามารถอธิบายด้วยลักษณะรูปสเปกตรัมของสัญญาณดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 สเปกตรัมของสัญญาณที่ถูกสุ่ม และสเปกตรัมหลังการสุ่ม

จากรูปที่ 3.2 แสดงให้เห็นว่าสเปกตรัมของสัญญาณที่ถูกสุ่มซึ่ง Bandwidth ไม่เกิน f_c ในขณะที่สัญญาณนี้จะถูกสุ่มด้วยความถี่ f_s ขบวนการมอดูเลชัน จะทำให้แถบสเปกตรัมของสัญญาณสุ่มขยายกว้างออกจาก f_s เป็น $2f_s$,$3f_s$ ... ถ้าความถี่ของสัญญาณสุ่ม f_s ไม่สูงพอหลังจากการสุ่มสเปกตรัมบางส่วนของ f_s จะไปซ้อนทับกับสเปกตรัมซึ่งเรียกว่า **Frequency folding** หากเป็นเช่นนี้ก็จะทำให้เกิดความผิดเพี้ยนของสัญญาณ หากทำการแปลงสัญญาณกลับไปเป็นอนาลอก

ดังนั้นจะต้องให้ $f_s > 2f_c$ เพื่อขจัดการซ้อนทับกันของสเปกตรัม ซึ่งกระทำได้สองวิธี วิธีที่หนึ่งด้วยการใช้อัตราการสุ่มที่สูงพอ และอีกวิธีหนึ่งคือ การทำการฟิวเตอร์ความถี่ของสัญญาณอนาลอกก่อนการสุ่ม (Antialiasing filters) เพื่อจำกัด Bandwidth ของสัญญาณที่จะถูกแปลงให้ไม่เกินกว่า $f_s / 2$

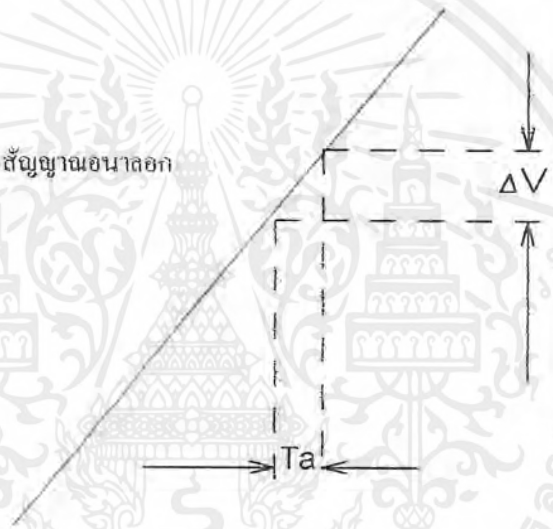
ในทางปฏิบัติแล้วจะยังคงเกิด frequency folding ได้เสมอจากส่วนของฮาร์โมนิคของสัญญาณรวมทั้งสเปกตรัมของสัญญาณรบกวนที่ยังคงอยู่ แม้ว่าทำการใช้ฟิวเตอร์ฮาร์โมนิคสูง ๆ ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณแล้วก็ตาม หนทางที่ดีที่สุดคือ พยายามให้การสุ่มสัญญาณเป็นไปอย่างรวดเร็วมากที่สุดเท่าที่จะกระทำได้ ซึ่งปกติจะสูงกว่าความถี่ต่ำสุดทางทฤษฎี Sampling คือ $2f_c$ เสมอ

3.2.2) Aperture time Error

ในการแปลงสัญญาณนั้น A/D จะต้องใช้เวลาช่วงหนึ่งในการจัดการ (Process) ซึ่งช่วงเวลาดังกล่าวขึ้นอยู่กับหลาย ๆ องค์ประกอบ เช่น ความละเอียดของการแปลงสัญญาณ , วิธีในการแปลงสัญญาณ อีกทั้งความเร็วของอุปกรณ์ในการทำงานร่วมกันอื่น ๆ ช่วงเวลาเรียกว่า **Aperture time** ซึ่งหมายถึงช่วงเวลาที่เกิดความไม่แน่นอนขึ้นในการวัด และผลที่ตามมาคือเกิดความผิดพลาดต่อค่าที่วัดได้



รูปที่ 3.3 แสดง error จากการวัดใน Aperture time

ในรูปสัญญาณ Analog V(t) มีอัตราการเปลี่ยนแปลง $\frac{dV(t)}{dt}$ ในช่วง Aperture time (T_a) ดังนั้นช่วงเวลาของการเปลี่ยนแปลงสัญญาณ จะเป็น

$$\Delta V = T_a \frac{dV(t)}{dt}$$

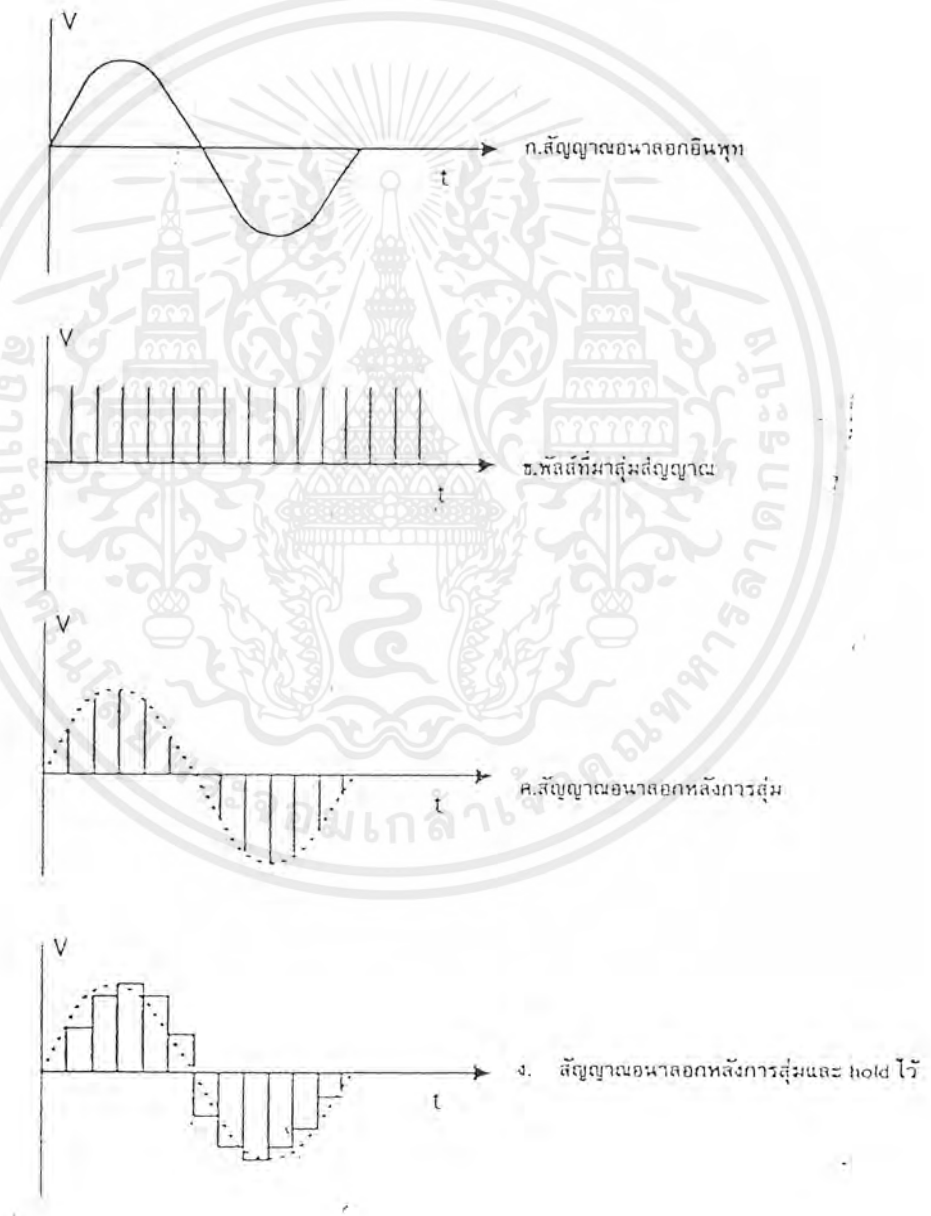
ดังนั้นหากเวลาที่ A/D ใช้ในการเปลี่ยนสัญญาณในช่วงเวลา T_a นี้ รหัส Digital ที่ได้อาจจะตรงกับขนาดของสัญญาณ Analog ในช่วงเวลานี้ และส่วนอื่น ๆที่เหลือคือ Error เรียกว่า Aperture time error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.2.3) Sample and Hold

เพื่อตัดปัญหา Aperture time error ดังกล่าวจึงมีแนวความคิดให้มีการคงค่าของระดับสัญญาณไว้จนกว่า process ของการ converse จะเสร็จสมบูรณ์เรียกวงจรส่วนนี้ว่า Sample and Hold ซึ่งวงจรส่วนนี้จะทำการสุ่ม (Sampling) สัญญาณอินพุท และนำสัญญาณที่สุ่มนั้นมาเก็บ (Hold)ไว้ในช่วงเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้วิธีการเก็บประจุบนตัวเก็บตัวประจุที่รั่วไหลต่ำในการสุ่มสัญญาณอนาลอกจะถูกสุ่มเป็นระยะ ๆ คงที่ตามรูปที่ 3.4



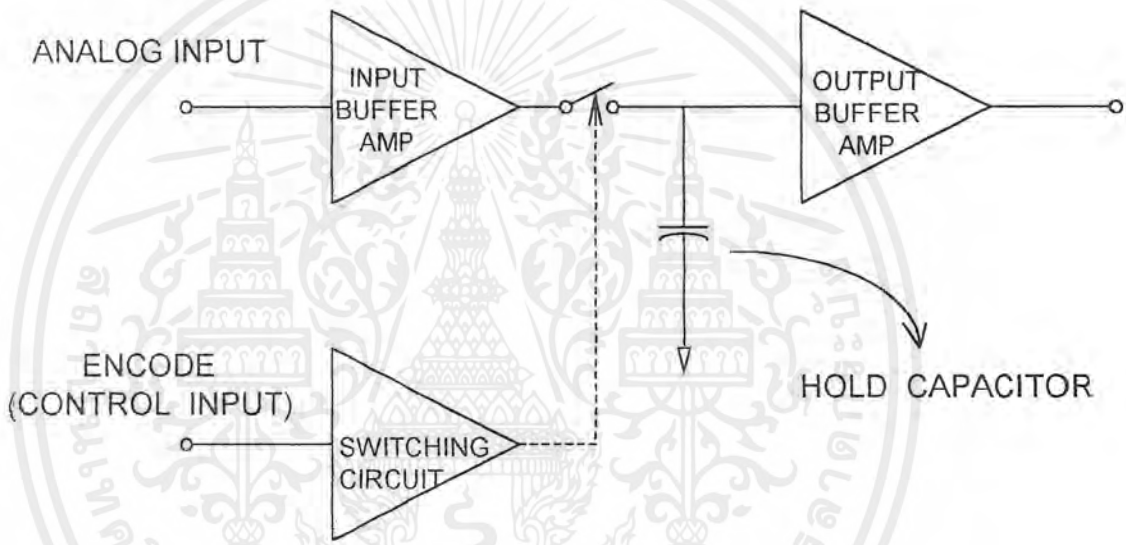
รูปที่ 3.4 การสุ่มของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36892

การสุ่มจะเป็นการตัดต่อสัญญาณ Analog ในช่วงเวลาอันสั้นด้วยสวิทซ์ที่ทำงานด้วยความเร็วสูง ผลของการสุ่มสัญญาณด้วยความเร็วจะเสมือนกับการคูณขบวนสัญญาณพัลส์แคบ ๆ กับสัญญาณ Analog ซึ่งจะได้เป็นสัญญาณที่มอดดูเลทระหว่างขบวนพัลส์กับสัญญาณ Analog โดยเสมือนว่าสัญญาณ Analog จะขี่มาบนขบวนพัลส์

3.2.3.1) Sample & Hold Operation

วงจร Sample & Hold โดยพื้นฐานแล้วจะมีโครงสร้างดังแสดงดังรูป



รูปที่ 3.5 บล็อกไดอะแกรม Sample & Hold

จากรูป สามารถแบ่งส่วนการทำงานได้เป็น 4 ส่วนด้วยกัน ดังนี้คือ

1. **The energy-storage device** เป็นอุปกรณ์ในการเก็บค่าของพลังงานเป็นหัวใจหลักของเรื่องนี้ ซึ่งส่วนใหญ่มักใช้ ตัวเก็บประจุ

2. **The Input Amplifier Buffers** ทำให้วงจรมี Input Impedance สูงเพื่อสะดวกต่อการใช้งานในการนำไปต่อรวมกับวงจรส่วนอื่น และยังช่วยเพิ่มกระแสไป Charge ตัวเก็บประจุในช่วง Track Mode ให้เร็วขึ้น

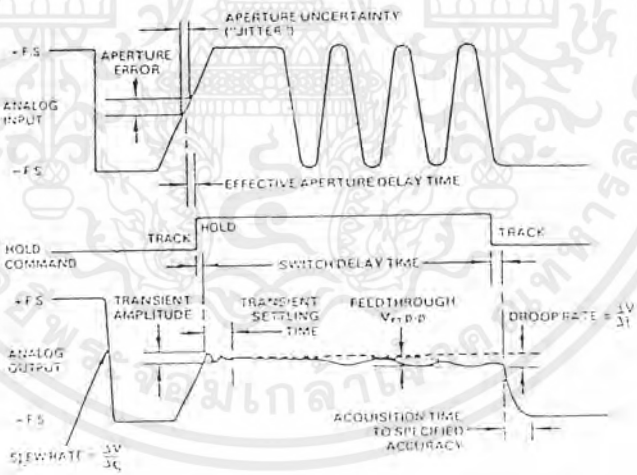
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. **The Output Amplifier Buffers** มี Input Impedance สูง เพื่อให้ตัวเก็บประจุยังคงรักษาระดับแรงดันอยู่ได้ในช่วง Hold Mode เพราะแทบไม่มีการ Discharge ของตัวเก็บประจุเลย นอกจากนี้ยังทำให้วงจรมี Output Impedance ต่ำเพื่อไม่ให้เกิดผลของ Loading effect จึงนำไปขับวงจร A/D ได้ง่าย

4. **Switching Circuit** ทำหน้าที่ตัดและเชื่อมต่อระหว่างภาค Input Amplifier Buffers กับตัวเก็บประจุ หาก Switch ปิดวงจร กระแสก็จะไหลได้ ตัวเก็บประจุก็จำทำการ Charge จนได้ระดับแรงดันเท่ากับอินพุท เรียกช่วงนี้ว่า **Track Mode** และเมื่อ Switch ถูกเปิด จะเป็นการตัดการเดินกระแสทิ้งเสีย ตัวเก็บประจุจะทำการ Discharge ประจุออกมาทาง Output เรียกช่วงนี้ว่า **Hold Mode**

3.2.3.2) Sample & Hold Mode

การทำงานดังที่ได้กล่าวไปแล้วว่ามีการทำงานอยู่ 2 Mode ด้วยกันดังนี้



รูปที่ 3.6 Timing Diagram แสดงถึง Track & Hold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

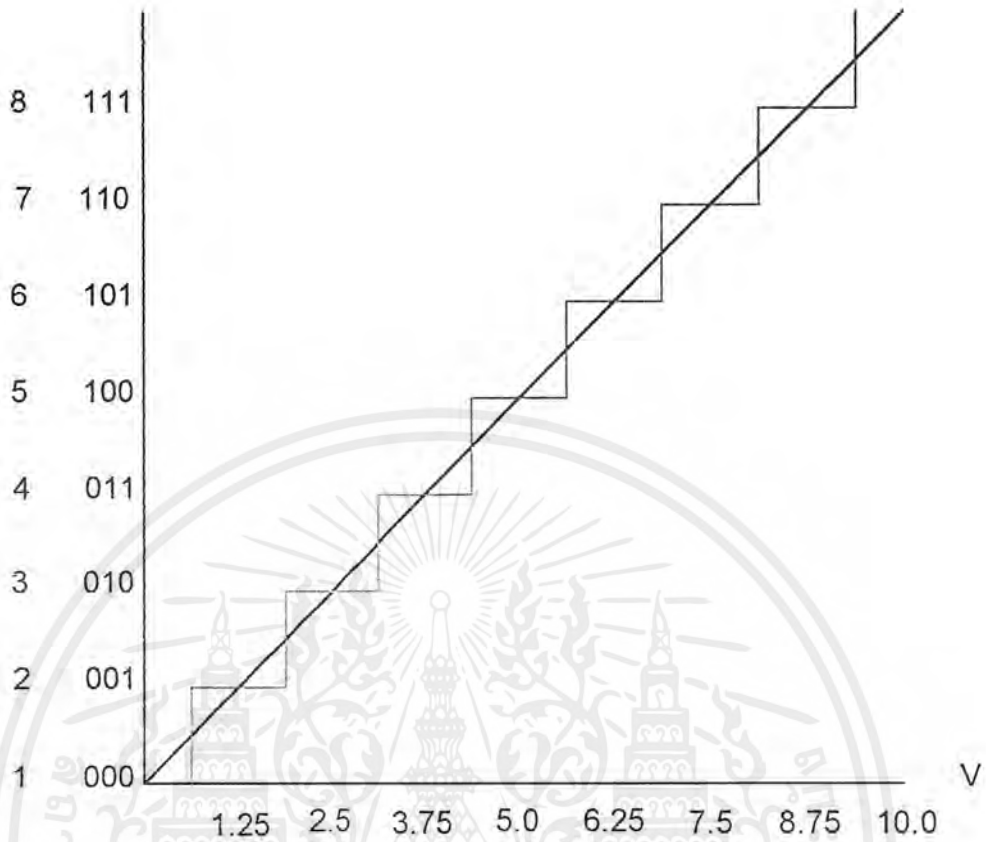
(1) Track Mode จากรูปจะได้ว่าในช่วง Track Mode Command สัญญาณ Analog output จะมีลักษณะตามสัญญาณ Analog Input แต่ในช่วงเริ่มแรกสัญญาณ Analog output จะไม่เปลี่ยนแปลงอย่างฉับพลันเหมือนกับสัญญาณ Analog Input แต่จะเปลี่ยนแปลงแรงดันด้วยอัตราเร็วค่าหนึ่ง เรียกว่า **Slewrate (SR)** = $\Delta v / \Delta t$ ในออกแบบวงจรต้องระวังในเรื่อง **Offset** ด้วยกล่าวคือ หาก Input ที่เข้ามาเป็น 0 V Output ที่ออกมาต้องได้ 0 Vด้วย

(2) Hold Mode สัญญาณ Analog Output จะมีค่าประมาณว่าเท่ากับจุดสุดท้ายของช่วง Track หรือก็คือจุดที่ Switch เปิดออกนั่นเอง เมื่อเริ่มเข้าสู่ Hold Mode Command แล้วสัญญาณที่ Analog Output จะยังไม่หยุดหรือคงค่าไว้ทันทีทันใดจะยังคงเปลี่ยนแปลงตาม Analog Input อยู่ชั่วระยะเวลาหนึ่ง ซึ่งเวลาดังกล่าวจะใช้ไปในการตอบสนองของวงจร เรียกเวลาดังกล่าวนี้ว่า **Aperture Delay** และเรียกระดับแรงดันที่เปลี่ยนแปลงในช่วง Aperture Time นี้ว่า **Transient Amplitude** เมื่อวงจรตอบสนองต่อการเปลี่ยนแปลงของ Hold Mode Command แล้วก็จะใช้เวลาระยะหนึ่งในการทำให้ Analog Output คงค่าอยู่ได้ ซึ่งในช่วงเวลานี้จะมีการแกว่งของสัญญาณ Analog Output อยู่ด้วย เรียกเวลาดังกล่าวนี้ว่า **Transient Settling Time** และเรียกเวลาทั้งหมดที่สูญเสียจนได้ระดับ Analog Output ที่คงตัวนี้ว่า **Aperture Time** และระหว่างการคงค่าไว้ อาจเกิดการแกว่งของสัญญาณได้เนื่องมาจากผลของตัวเก็บประจุของส่วน Switch เอง เรียกว่า **Feedthrough** เมื่อสิ้นสุด Hold Mode Command แล้ว เข้าสู่ Track Mode Command อีกครั้งหนึ่ง จะสังเกตเห็นได้ว่า ระดับแรงดันสุดท้ายของการ Hold จะตกลงอยู่เล็กน้อยทั้งนี้เกิดจากผลของ leakage ของ Hold Capacitance

3.2.4) Quantizing theory

Quantizing เป็นขบวนการที่เปลี่ยนสัญญาณ Analog เป็นสัญญาณที่ไม่ต่อเนื่อง โดยผ่านขบวนการเข้ารหัส (Coding) จัดให้สัญญาณที่ไม่ต่อเนื่องนั้นอยู่ในรูปที่ง่ายต่อการประมวลผล เช่นในรูปของ Binary ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ความสัมพันธ์ระหว่าง อนาลอกกับดิจิตอล

ในรูปกราฟแสดงให้เห็นถึงความสัมพันธ์กันระหว่างสัญญาณ Analog ที่ขนาด 0 ถึง 10 V ถูก Quantize และ encode เป็น Binary 3 bit ได้ระดับ 000 ถึง 111 เนื่องจากในระบบ Binary รหัส digital แต่ละค่าจะแทนขนาดของสัญญาณ Analog แต่ละค่าที่เป็นสัดส่วนกับค่าเต็มสเกล โดยค่าสูงสุดของรหัสดิจิตอลคือ ทุกบิตเป็น 1 จะเท่ากับสัญญาณ Analog เต็มสเกลคูณด้วยค่า Weight ของรหัสนั้นหารด้วย 2^n

3.2.5) Quantizer Resolution and Error

แต่ในละสถานะของสัญญาณเอาท์พุทจะแทนขนาดของสัญญาณ Analog ค่าใดค่าหนึ่งในช่วงเล็ก ๆ ระหว่างจุดแบ่งระดับ เรียกช่วงเล็ก ๆ นี้ว่าหนึ่ง Analog Quantization หรือ หนึ่งควอนตัม (Quantum) หรือ 1 LSB (Least Significant Bit) ของการแปลงสัญญาณ

$$Q = \frac{FSR}{2^n}$$

FSR : ช่วงเต็มสเกลของแรงดัน Analog

n : จำนวนบิตของสัญญาณ digital

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่านี้จะบ่งบอกถึงความผิดพลาดที่เกิดขึ้นได้ ดังนั้นค่านี้ยิ่งมีค่าน้อยเพียงใดก็ยิ่งดี แต่ในการทำให้ค่านี้ลดน้อยลง จากสมการแล้วจะเห็นได้ว่าไม่มีวิธีอื่นเลย นอกจากเพิ่มจำนวน Bit ของสัญญาณ digital ให้เพิ่มขึ้น ซึ่งก็จะหมายถึงค่าใช้จ่ายที่สูงขึ้นตามเช่นกัน

3.3) การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital Conversion)

ในการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล มีวิธีการอยู่หลายชนิดด้วยกัน แต่ละชนิดก็มีลักษณะเฉพาะและความเหมาะสมในการใช้งานแตกต่างกันไป ในที่นี้จะขอกล่าวถึงชนิดของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลที่มีความเร็วสูง และนำมาใช้ในโครงงานนี้คือ

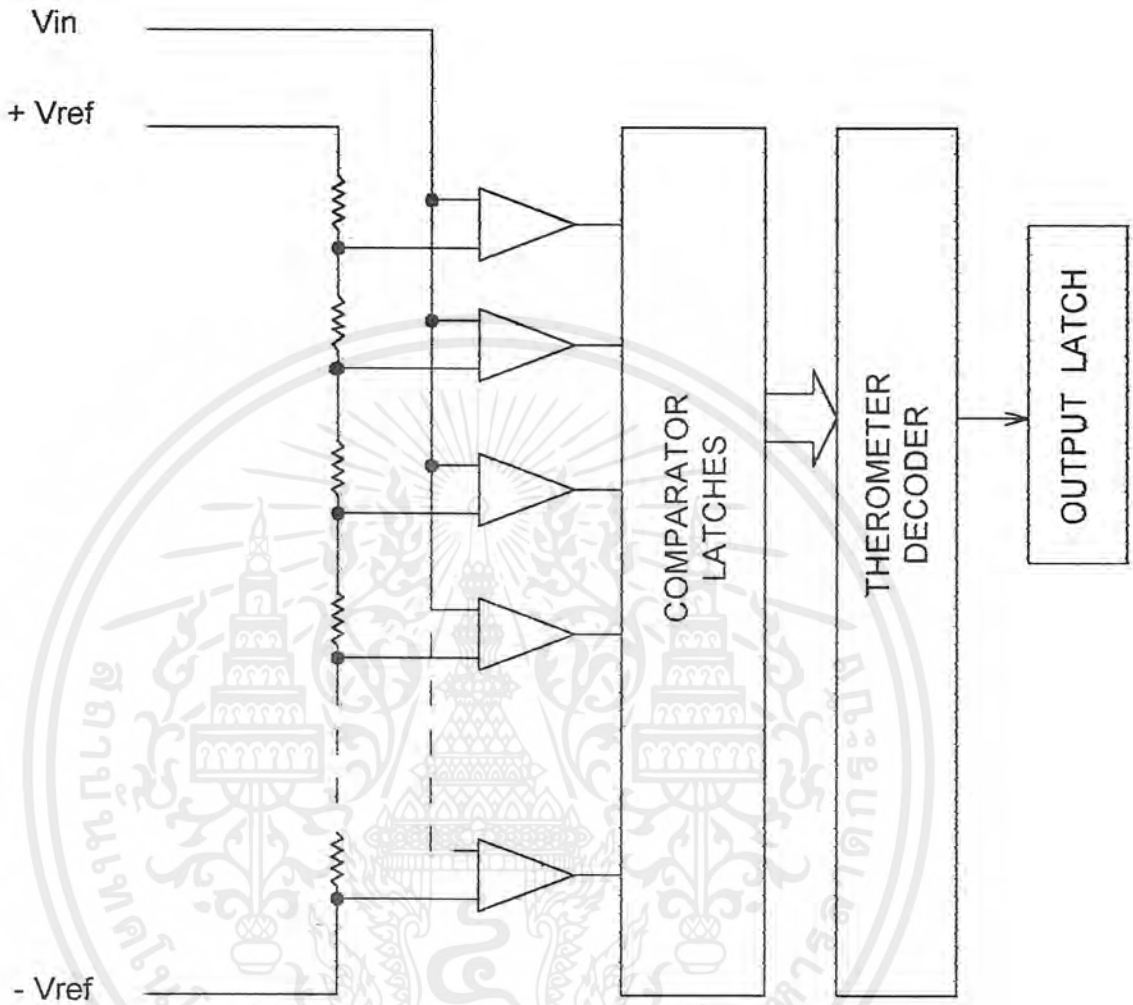
3.3.1) Parallel (Flash) A/D

สำหรับการแปลงสัญญาณที่ต้องการความเร็วสูงมาก ๆ เช่น การแปลงสัญญาณภาพโทรทัศน์ เรดาห์ การสื่อสารระบบดิจิตอล เป็นต้น การทำงานคือ จะใช้ Comparator ทำการเปรียบเทียบสัญญาณอนาลอกอินพุท กับ แรงดันอ้างอิงที่แบ่งแรงดันให้สอดคล้องกับรหัสดิจิตอล โดยใช้ตัวต้านทานแล้วแปลงเอาท์พุทจาก Comparator ให้ตรงกับรหัสดิจิตอล ซึ่งจะเห็นได้ว่าอุปสรรค์ด้านความเร็วมีเพียง Propagation time ของ Comparator เท่านั้น ดังนั้นข้อดีก็คือ มีความเร็วที่สูงมาก แต่ข้อเสียก็คือ ต้องใช้ Comparator ถึง $2^n - 1$ ตัว (n คือจำนวนบิตของสัญญาณดิจิตอล) ซึ่งจะเห็นได้ว่าสิ้นเปลืองมาก ทำให้วงจรขนาดใหญ่ ต้องการกำลังงานสูง ที่สำคัญคือมีราคาแพงอีกด้วย

3.3.1.1) โครงสร้างพื้นฐานของ Flash A/D Converters

ลักษณะโดยทั่ว ๆ ไปของ Flas A/D Converters จะเป็นดัง Block diagram ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



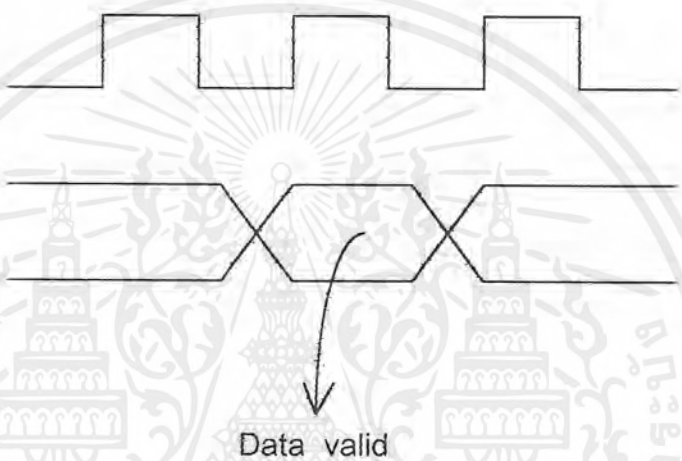
รูปที่ 3.8 บล็อกไดอะแกรมของ Flash Converter

Flash Converters จะทำการเปรียบเทียบสัญญาณที่เข้ามากับระดับสัญญาณอ้างอิงระดับหนึ่ง ซึ่งจะใช้ Comparator เป็นตัวเปรียบเทียบ หลังจากตรวจสอบแล้วจะส่งต่อไปยัง Large Digital encoding section ตัว comparator จะมีจำนวนเท่ากับ 2^n-1 เมื่อ n เป็นจำนวนบิตที่ต้องการแปลง ซึ่งจะเห็นได้ว่าต้อนสิ้นเปลืองไปกับ Op-Amp เป็นจำนวนมาก เช่น 8 bit ต้องใช้ Op-amp มากถึง 2^8-1 = 255 ตัว ทำให้วงจรมีขนาดใหญ่และต้องการใช้พลังงานจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.2) Timing Requirements

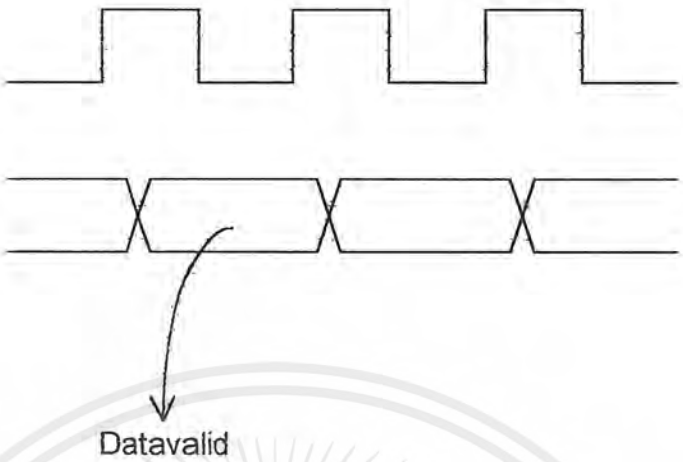
ในกรณีที่เรานำสัญญาณดิจิตอลที่ได้จากการแปลงของ A/D แบบ Flash Converters ไปต่อเข้ากับหน่วยความจำจะทำให้ A/D แบบ Flash Converters ถูกจำกัดความเร็วอยู่ที่ความเร็วของการทำงานในส่วนของหน่วยความจำ ดังนั้น A/D แบบ Flash Converters จึงจำเป็นต้องมี Clock เพื่อควบคุมการ Latch ข้อมูลไว้ด้วย A/D แบบ Flash Converters ที่ปราศจากการ latch ค่า ต้องการช่วงเวลาใน process ประมาณเพียง clock 1 ลูกเท่านั้น ดังตัวอย่างในรูป



รูปที่ 3.9 Timing Diagram without latch

จากรูปในตัวอย่าง แสดงให้เห็นว่า Input เข้ามาขณะที่ Clock อยู่ระดับ Low และ Output exist อยู่ที่ระดับ High ในช่วงเวลาระหว่าง Clock 2 phase จะมี Output ที่มีค่าที่ไม่แน่นอนอยู่ จุดนี้เป็นเรื่องที่สำคัญมาก Sampled Input จะใช้เวลา 1 ลูก (2 phase) ของสัญญาณเวลา ซึ่งเท่ากับเวลาในการสุ่ม ดังนั้นขั้นต่ำของความเร็วของหน่วยความจำในการ process โดยรวมต้องมีใช้เวลาน้อยกว่า 1 ลูกของสัญญาณเวลาอยู่เล็กน้อย ในบางครั้งเพื่อความเหมาะสมแล้วอาจจำเป็นต้องขยายเวลาของ Data Valid ให้ยาวนานขึ้น ซึ่งทำได้ด้วยการ latch สำหรับหน่วยความจำที่มีความเร็วไม่สูงนัก Flash Converters ที่มีประโยชน์และง่ายขึ้น ควรมี latch อยู่ภายในตัวเอง ด้วยเหตุผลที่กล่าวมาแล้ว ดังรูปข้างล่างเป็น Timing ที่ได้รับการ latch ค่าไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Timing Diagram with latch

3.3.3.3) Encoder Requirements

เมื่อมีสัญญาณ Input เข้ามา Comparator ก็จะทำการเปรียบเทียบแล้วจะให้ค่า Logic เป็น 0 (Low) หรือ 1 (High) ออกมา สำหรับ Input มีค่ามากหรือน้อยกว่า ระดับสัญญาณอ้างอิงอย่างเด่นชัด ปัญหาและข้อจำกัดอยู่ที่ Input ที่มีค่าเข้าใกล้ระดับสัญญาณอ้างอิงมาก ๆ แล้ว ตัว Comparator จะไม่สามารถแยกแยะความแตกต่างได้อย่างชัดเจนได้ ก็จะได้ Output เป็นสถานะที่ไม่แน่นอนเกิดขึ้น จึงก่อให้เกิดความผิดพลาดได้ เรียกความผิดพลาดลักษณะนี้ว่า **Metastability**

กรณีที่เลวที่สุด ก็คือ Bit สูงสุด MSB ผิดพลาดไป ในเหตุการณ์เช่นนี้จะกระเทือนต่อนัยยะสำคัญโดยรวมของข้อมูลได้มากที่สุด

Metastable Example 10 bit A/D Code

True code = 1000000000

Actual code = 0000000000

เรียกความผิดพลาดนี้ว่า **Gross Error** บ่อยครั้งอาจจะได้ลักษณะของสัญญาณที่คล้ายกับการสปาร์ค เรียกว่า **Sparkle code**

มีหนทางที่จะช่วยลดข้อผิดพลาดนี้ของ Sparkle codes ได้ โดยใช้ Additional Logic เป็นตัวถอดรหัส ตัวอย่างเช่น หากเป็นสัญญาณภาพ ก็จะใช้ Gray Code โดยมันจะยอมรับและสร้างความผิดพลาดได้สูงสุดเพียง 1-LSB เท่านั้น อย่างไรก็ดีการทำเช่นนี้จะเป็นตัวจำกัดความเร็วของ Converters ดังนั้นเราต้องพึงพิจารณาและเลือกใช้ให้ดีระหว่างความถูกต้องกับความเร็วที่ต้องเสียไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

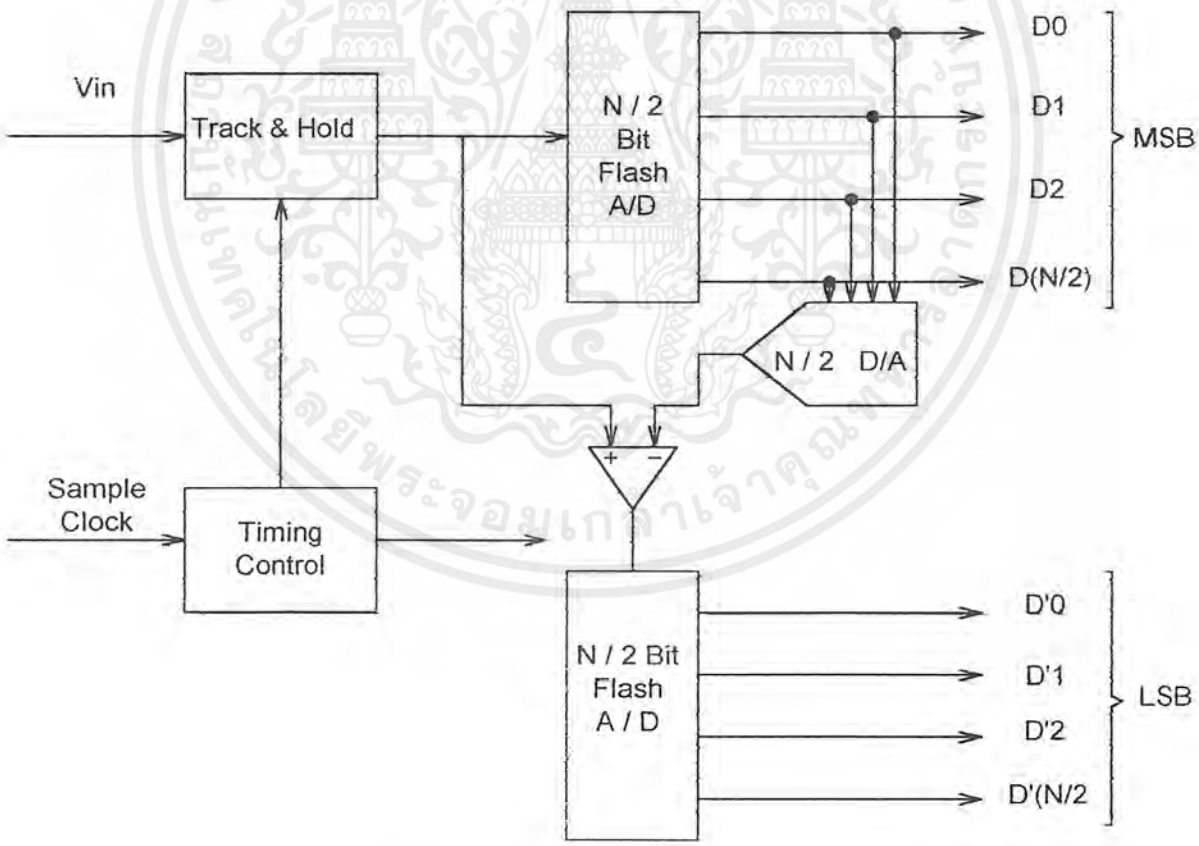
(ทั้งนี้ ในเรื่องการจัดเก็บเป็นข้อมูลแบบ Digital ใน Code ต่าง ๆ จะกล่าวอีกทีในช่วงหลัง)

3.3.2) Subranging Flash A/D Converter

เนื่องจาก A/D แบบ Flash Converter หากต้องการความละเอียดสูง ๆ คือ มีจำนวนบิต มากๆ แล้ว จะทำให้ Flash Converter มีราคาสูงและสิ้นเปลืองพลังมากอีกด้วย เพราะจะมีวงจรภายในใหญ่และซับซ้อนขึ้น จึงมีวิธีการต่อ Flash Converter ในลักษณะอื่น ๆ ที่ช่วยลดปัญหาดังกล่าวได้ แต่มีข้อจำกัดคือ ความเร็วและความถูกต้องจะลดลงนั่นเอง *แต่ถึงอย่างไรก็ตามก็ยังมีความเร็ว และความถูกต้องสูงอยู่ในระดับที่ยอมรับได้* วงจร Subranging Flash A/D Converter มีอยู่ 2 ลักษณะด้วยกัน คือ

(1) Half Flash Converters

วงจร A/D แบบ Half Flash Converters นี้ยังคงมีความเร็วสูงอยู่ แต่จะช่วยลดต้นทุน ลดจำนวนของ Comparator ที่ใช้ในวงจร ลดการกินกำลังงาน ลดขนาดของ Encoding Logic ลงได้ A/D แบบ Half Flash Converters นี้มีโครงสร้างดังรูป



รูปที่ 3.11 บล็อกไดอะแกรมของ Half Flash Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป A/D แบบ Half Flash Converter จะมีการแปลงสัญญาณ 2 ตอน คือ ในตอนแรกจะเป็นการแปลงโดยหยาบ ๆ ก่อน (ได้สัญญาณดิจิตอล n/2 บิตด้าน High ออกมาก่อน) โดยใช้ Flash Converter ตัวที่หนึ่ง ซึ่งการประมาณโดยหยาบ ๆ ก็คือไม่ละเอียดนั่นก็หมายถึงเราจะประมาณแล้วให้ค่าที่ บิตสูงก่อน หลังจากนั้นนำสัญญาณดิจิตอล n/2 บิตด้าน High แปลงกลับเป็น Analog โดยใช้ DAC แล้วนำไปลบกับสัญญาณ Input เดิม (ในการลบกันให้สัญญาณ Input เดิมเป็นตัวตั้ง และสัญญาณที่ได้จาก DAC เป็นตัวลบ) โดยใช้ Differential Amplifiers ที่มีอัตราการขยายเป็น 1 เป็นวงจรลบสัญญาณระหว่างสัญญาณ Input กับ สัญญาณที่ได้จาก DAC ค่าความแตกต่างที่ได้นี้จะถูกส่งไปยัง Flash Converter ตัวที่สอง ทำการแปลงเป็นสัญญาณดิจิตอลออกมา ซึ่งการประมาณในช่วงหลังนี้เป็นการประมาณโดยละเอียด (ได้สัญญาณดิจิตอล n/2 บิตด้าน Low ออกมา) เมื่อรวมกับสัญญาณดิจิตอลที่ได้จาก Flash Converter ตัวที่หนึ่ง ก็จะได้สัญญาณดิจิตอลจำนวน n บิตออกมาตามต้องการ

ข้อควรระวัง คือ Differential Amplifiers ที่ใช้ทำหน้าที่ลบสัญญาณอาจมีค่า Offset เกิดขึ้น ทำให้เกิดข้อผิดพลาด (Error) ขึ้นได้ โดยส่วนใหญ่แล้วความผิดพลาดดังกล่าวจะไปปรากฏที่ LSB สุดท้าย วิธีแก้ไขก็คือ ให้ออกแบบโดยใช้เป็น N+1 bit เมื่อ N คือจำนวนบิตที่ต้องการ แล้วไม่ใช้หลักของ LSB

จะเห็นได้ว่า Half Flash Converters N Bit ก็คือการใช้ Flash Converters $\frac{N}{2}$ Bit จำนวน 2 ชุดด้วยกัน การต่อลักษณะนี้จะใช้ Comparator ลดลงอย่างมากทีเดียว เช่น

8 bit data

Full Flash Comparator $= 2^n - 1$

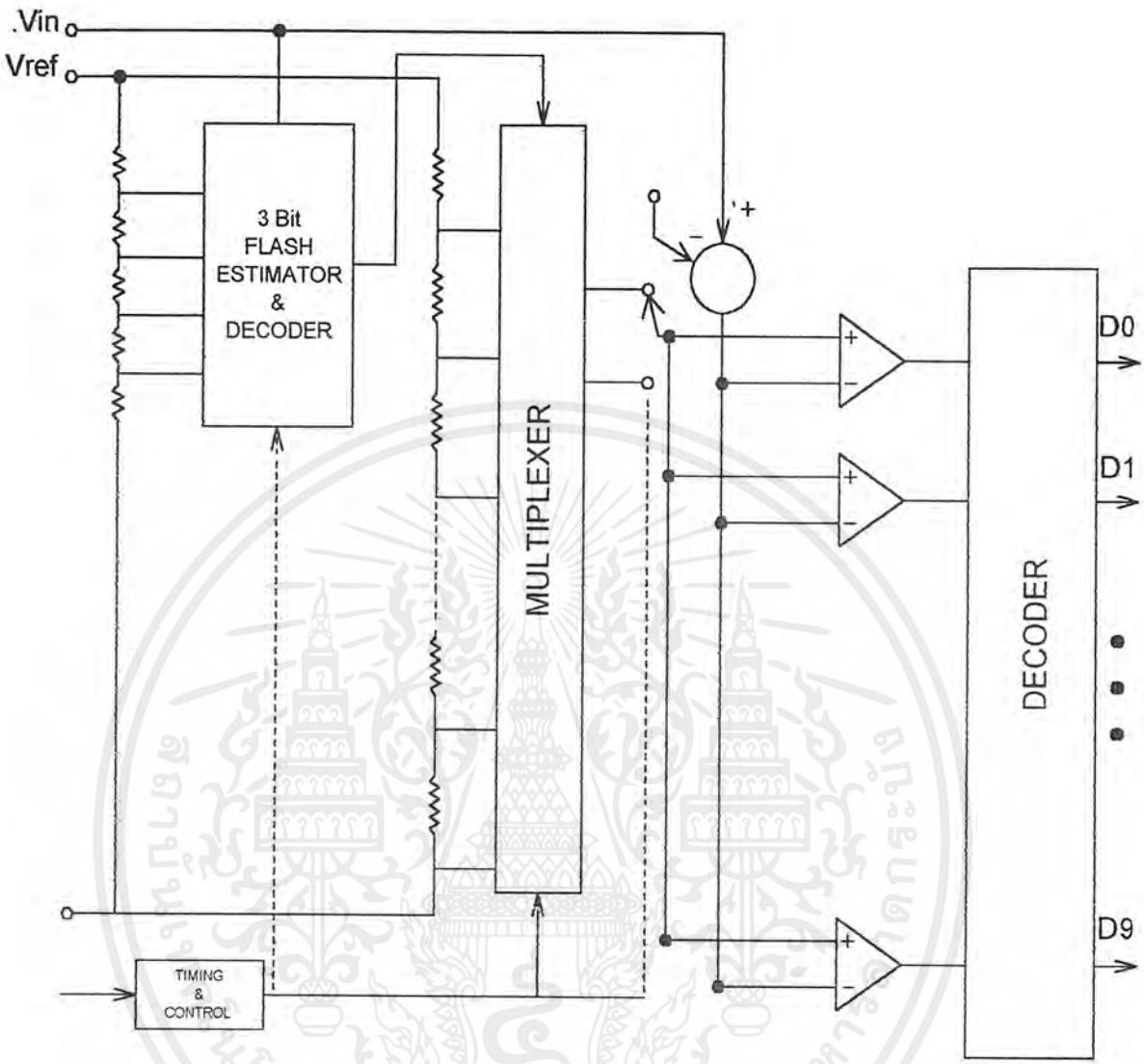
$= 255$

Half Flash Comparator $= 2(2^{n/2} - 1)$

$= 30$

(2) Multistep Flash Converters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 บล็อกไดอะแกรมของ Multistep Flash Converter

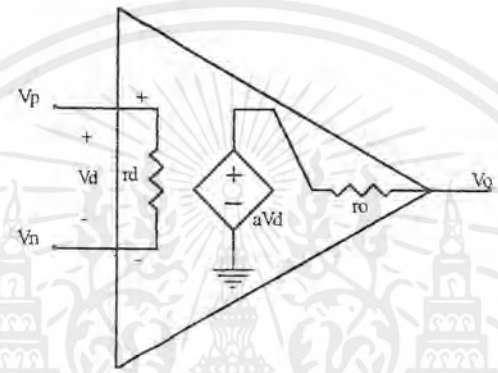
การต่อในลักษณะนี้จะเป็นการประหยัด Comparator ได้มากกว่าแบบ Half Flash Converters เสียอีก แต่จะทำงานได้ช้ากว่า แบบ Half Flash

ใน Half Flash Converters จะมีการประมาณค่าอยู่ 2 ครั้ง คือ ในช่วงแรกจะเป็นการประมาณอย่างหยาบ ๆ และในช่วงหลังเป็นการประมาณโดยละเอียด ทั้งนี้ในการประมาณ 2 ครั้งจะใช้ Flash Converters 2 ชุด กล่าวคือประมาณ 1 ครั้ง ก็ใช้ 1 ชุด แต่ใน Multistep Flash Converters นี้จะมีการประมาณหลาย ๆ ครั้ง โดยใช้ Flash Converters เพียงชุดเดียวด้วยเหตุผลดังกล่าวนี้ทำให้เป็นการลด Comparator ได้อย่างมากมายเลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4) ทฤษฎีของออปแอมป์ (Operational Amplifier Theory)

ออปแอมป์ คือวงจรขยายแรงดันที่มีค่าอัตราขยายแรงดันสูงมาก องค์ประกอบสำคัญที่ใช้วิเคราะห์คุณสมบัติของออปแอมป์ได้แก่ ความต้านทานที่ขั้วอินพุทของออปแอมป์ (Differential input resistance : rd) , อัตราขยายแรงดัน (Voltage differential gain : a) และความต้านทานที่ทางออกของออปแอมป์ (Output resistance : ro) ซึ่งองค์ประกอบเหล่านี้เป็นองค์ประกอบแบบไม่มีการป้อนกลับ (Open-loop parameter) ของออปแอมป์ และสามารถแสดงได้ดังรูป



รูปที่ 3.13 Equivalent Circuit ของออปแอมป์

จากวงจรสมมูลของออปแอมป์ในรูปที่3.13 สามารถแสดงความสัมพันธ์ของแรงดันระหว่างอินพุทและเอาท์พุทได้ดังนี้

$$Vd = Vp - Vn$$

$$Vo = a(Vp - Vn) = aVd$$

โดย Vd คือ ผลต่างของแรงดันที่ขั้วอินพุททั้งสอง (Differential input voltage)

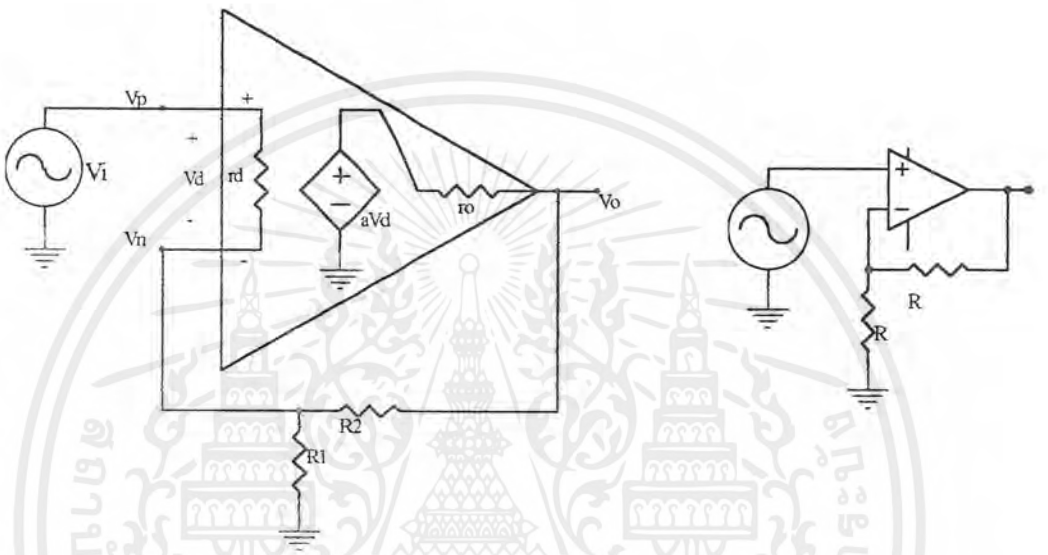
เนื่องจากองค์ประกอบที่กล่าวมานี้เป็นองค์ประกอบแบบไม่ป้อนกลับหรือ Open-loop แต่ในการใช้ออปแอมป์ต่อเป็นวงจรขยายนั้นมักจะมีการต่อออปแอมป์แบบให้มีการป้อนกลับด้วย โดยใช้ Feedback resistor เป็นตัวลดอัตราขยายแรงดันให้เหลือน้อยลงเรียกว่า Close-loop voltage gain : A_{CL} ซึ่งผลของการลดลงของอัตราขยายแรงดันนี้เป็นการปรับปรุงวงจรให้มีคุณสมบัติต่างๆ ให้ดีขึ้นดังนี้

- ความต้านทานที่ทางเข้าสูงขึ้น
- ความต้านทานที่ทางออกลดลง
- อัตราขยายแรงดันมีความเสถียรภาพดีกว่า และสามารถปรับ Gain ได้จากความต้านทานที่ต่ออยู่ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผลตอบสนองความถี่ (Frequency response) สูงขึ้น
- ช่วยลดสัญญาณรบกวนได้
- วงจรขยายมีความเป็นเชิงเส้น (Linear) มากขึ้น

3.4.1) Noninverting Amplifier มีการต่อวงจรเป็นดังรูปที่ 3.14



รูปที่ 3.14 การต่อวงจรแบบ Noninverting Amplifier

(1) อัตราขยายแรงดันของ Noninverting Amplifier

พิจารณาสมการ KCL ที่ Node Vn , Vo ดังนี้

Node Vn : $$\frac{Vi - Vn}{rd} - \frac{Vn}{R1} + \frac{Vo - Vn}{R2} = 0$$

Node Vo : $$\frac{Vn - Vo}{R2} + \frac{a(Vi - Vn) - Vo}{ro} = 0$$

แก้สมการทั้งสองเพื่อหา A = Vo/Vi จะได้

$$A = \frac{Vo}{Vi} = \frac{(1+R2/R1)a + ro/rd}{1 + a + R2/R1 + (R2+ro)/rd + ro/R1}$$

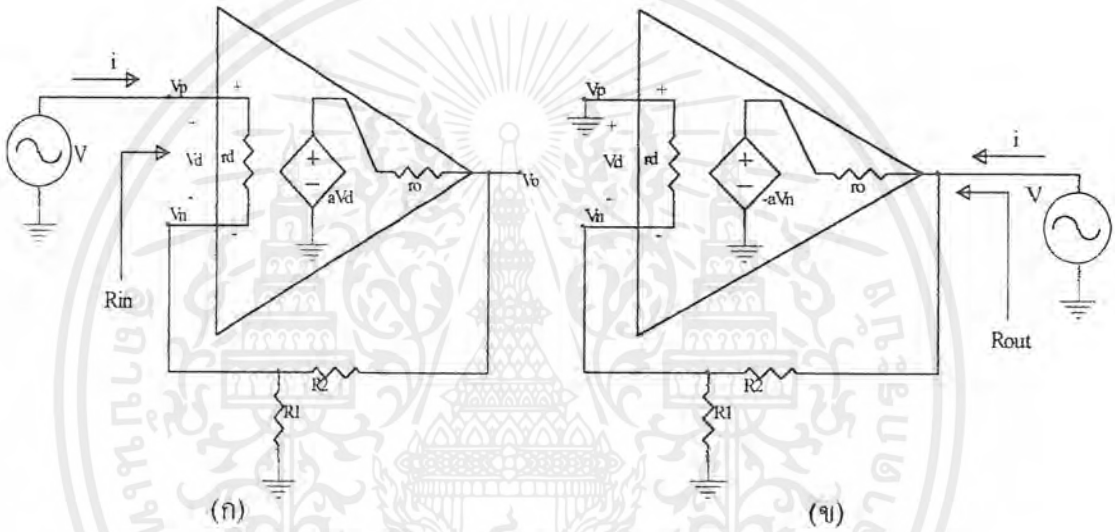
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในวงจรขยายที่ดีนั้น อัตราส่วน $\frac{ro}{rd}, \frac{R2+ro}{rd}, \frac{ro}{R1}$ สามารถตัดทิ้งได้ ดังนั้นจะได้

$$A = \frac{Vo}{Vi} \approx (1+\frac{R2}{R1})(\frac{1}{1+(1+R2/R1)/a})$$

สำหรับออปแอมป์ในอุดมคติ ($a = \infty$) จะได้ $A = \frac{Vo}{Vi} = (1+\frac{R2}{R1})$

(2) Rin และ Rout ของ Noninverting Amplifier



รูปที่ 3.15 การหา Rin และ Rout ของ Noninverting Amplifier

(2.1) หาค่า Rin จากรูปที่ 3.15 ก

Node Vn :
$$\frac{V - Vn}{rd} - \frac{Vn}{R1} + \frac{a(V - Vn) - Vn}{R2 + ro} = 0$$

เนื่องจาก $Vn = V - Vd = V - i.rd$ แก้สมการจะได้ Rin คือ

$$Rin = \frac{V}{i} = rd(1 + \frac{a}{1 + (R2+ro)/R1}) + R1||(R2 + ro)$$

สำหรับออปแอมป์ทั่วไป ค่า a มักจะมีค่ามากๆ ดังนั้นเราสามารถตัดเทอมสุดท้ายทิ้งได้ และในการออกแบบที่ดีนั้น เรามักจะให้ $ro << R2$ ดังนั้นจะได้

$$Rin \approx rd(1 + T) \qquad \text{โดย } T = \frac{aR1}{R1 + R2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2.2) หาค่า Rout จากรูปที่ 3.15 ข

Voltage divider : $$Vn = \left(\frac{R1\|rd}{R1\|rd + R2}\right)V$$

Node output : $$i + \frac{Vn - V}{R2} + \frac{-aVn - V}{ro} = 0$$

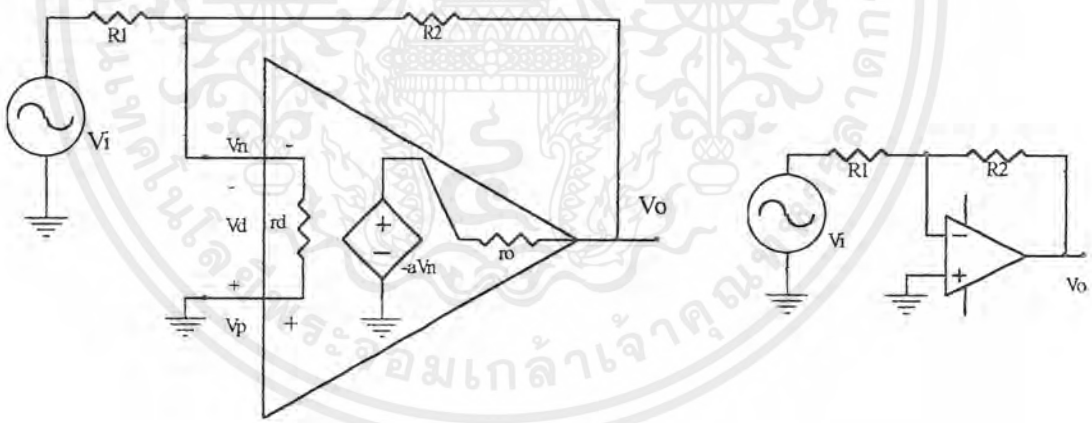
แก้สมการทั้งสองโดยกำจัดตัวแปร Vn ทิ้งจะได้ Rout คือ

$$Rout = \frac{V}{i} = \frac{ro}{1 + (a + ro/R1 + ro/rd)/(1 + R2/R1 + R2/rd)}$$

โดยทั่วไปแล้ว ค่า rd จะมีค่ามากกว่า R1 , R2 มาก และค่า ro มักจะมีค่าน้อยเมื่อเทียบกับ R1 , R2 , rd เราจึงสามารถตัดเทอม ro/R1 , ro/rd , R2/rd ทิ้งได้ ดังนั้นจะได้

$$Rout \approx \frac{ro}{1 + T} \qquad \text{โดย } T = \frac{aR1}{R1 + R2}$$

3.4.2) Inverting Amplifier มีการต่อวงจรเป็นดังรูปที่ 3.16



รูปที่ 3.16 การต่อวงจรแบบ Inverting Amplifier

(1) อัตราขยายแรงดันของ Inverting Amplifier พิจารณา KCL ที่ Node Vn , Vo ดังนี้

Node Vn : $$\frac{Vi - Vn}{R1} - \frac{Vn}{rd} + \frac{Vo - Vn}{R2} = 0$$

Node Vo : $$\frac{Vn - Vo}{R2} + \frac{-aVn - Vo}{ro} = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้สมการทั้งสองเพื่อหา A = Vo/Vi จะได้

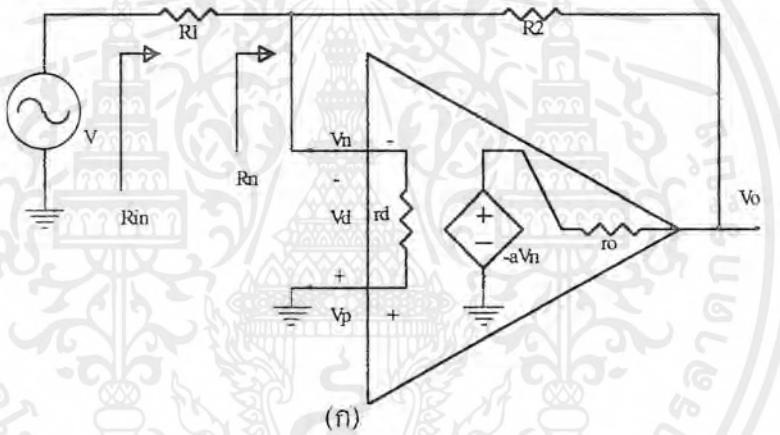
$$A = \frac{Vo}{Vi} = \frac{aR2 - ro}{(1+a)R1 + (R2+ro)(1+R1/rd)}$$

สำหรับการออกแบบวงจรที่ดีนั้นมักจะทำให้ ro << R2 และ R1/rd << 1 ดังนั้นจะได้ แก้สมการทั้งสองเพื่อหา A = Vo/Vi จะได้

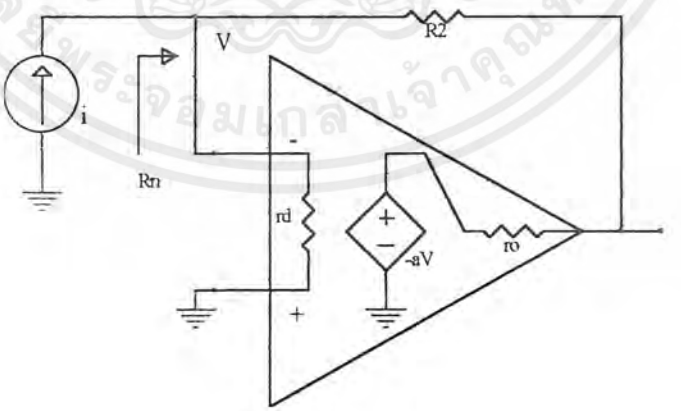
$$A \approx (-\frac{R2}{R1})\frac{1}{1+1/T} \qquad \text{โดย } T = \frac{aR1}{R1+R2}$$

สำหรับออปแอมป์ในอุดมคติ (a = ∞) จะได้ $A = \frac{Vo}{Vi} = -\frac{R2}{R1}$

(2) Rin และ Rout ของ Inverting Amplifier



(ก)



(ข)

รูปที่ 3.17 การหาค่า Rin ของ Inverting Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2.1) หาค่า Rin จากรูปที่ 3.17

จากรูปที่ 3.17 (ก) จะได้ $Rin = R1 + Rn$

จากรูปที่ 3.17 (ข) ทำการหาค่า Rn โดยพิจารณา KCL ที่ Node V จะได้

Node V : $$i - \frac{V}{rd} + \frac{-aV - V}{R2 + ro} = 0$$

$$Rn = \frac{V}{i} = \frac{R2 + ro}{1 + a + (R2 + ro)/rd}$$

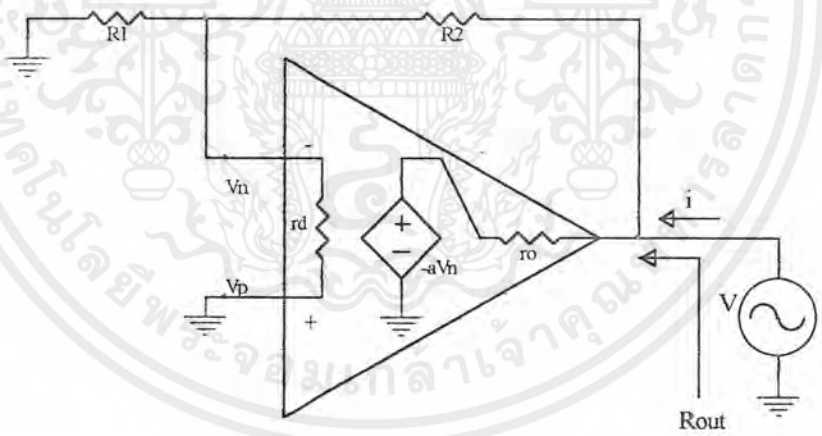
โดยทั่วไปแล้วเทอม (R2 + ro)/rd สามารถตัดทิ้งได้ ดังนั้น

$$Rn \approx \frac{R2 + ro}{1 + a}$$

$$Rin \approx R1 + \frac{R2 + ro}{1 + a}$$

สำหรับออปแอมป์ในอุดมคติ ($a = \infty$) จะได้ $Rin = R1$

(2.2) หาค่า Rout จากรูปที่ 3.18



รูปที่ 3.18 ค่า Rout ของ Inverting Amplifier

จะเห็นได้ว่า ค่า Rout ของ Inverting Amplifier จะมีค่าเท่ากับ Rout ของ Noninverting Amplifier นั่นคือ

$$Rout = \frac{V}{i} = \frac{ro}{1 + (a + ro/R1 + ro/rd)/(1 + R2/R1 + R2/rd)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปแล้ว ค่า rd จะมีค่ามากกว่า R1 , R2 มาก และค่า ro มักจะมีค่าน้อยเมื่อเทียบกับ R1 , R2 , rd เราจึงสามารถตัดเทอม ro/R1 , ro/rd , R2/rd ทิ้งได้ ดังนั้นจะได้

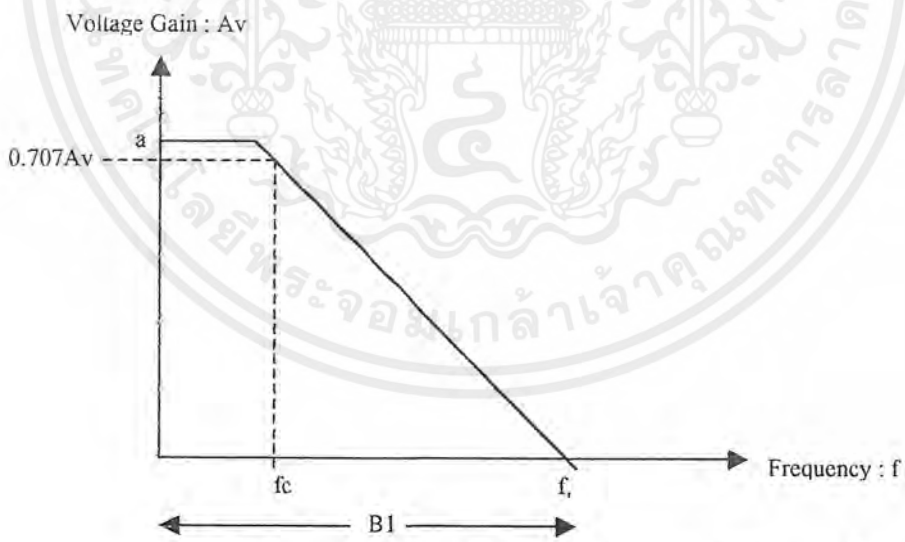
$$Rout \approx \frac{ro}{1+T} \qquad \text{โดย } T = \frac{aR1}{R1+R2}$$

3.5) ผลตอบสนองทางความถี่ของออปแอมป์

ออปแอมป์ได้ถูกออกแบบให้เป็นวงจรขยายที่มีอัตราขยายแรงดันสูง และมี Bandwidth กว้าง การทำงานในลักษณะเช่นนี้จะนำไปสู่สภาวะที่ไม่เสถียร (Unstable) ได้อันเนื่องจากการป้อนกลับทางบวก (Positive feedback) ดังนั้นเพื่อให้ออปแอมป์ทำงานอย่างมีเสถียรภาพได้นั้น จึงต้องมีการต่อวงจรชดเชยภายในไว้ข้างในออปแอมป์ แต่ก็มีผลทำให้อัตราขยายแรงดันมีค่าลดลงเมื่อความถี่เพิ่มขึ้นด้วย การลดลงของอัตราขยายแรงดันนี้ถูกเรียกว่า roll-off ซึ่งในออปแอมป์ส่วนใหญ่ roll-off จะมีอัตรา 20 dB per decade (-20 dB/decade)

3.5.1) Gain – Bandwidth

เนื่องจากผลของวงจรชดเชยภายในตัวออปแอมป์ ค่าอัตราขยายแรงดันของออปแอมป์จะมีค่าลดลงเมื่อความถี่เพิ่มสูงขึ้น ดังแสดงได้ดังรูปที่ 3.19



รูปที่ 3.19 ความสัมพันธ์ระหว่างอัตราขยายแรงดันกับความถี่

ที่ช่วงความถี่ของสัญญาณอินพุทค่าต่ำๆ ค่าอัตราขยายแรงดันของออปแอมป์ก็คือ ค่า Voltage differential gain : a นั่นเอง เมื่อความถี่ของสัญญาณอินพุทมีค่าสูงขึ้นเรื่อยจนถึงความถี่ค่าหนึ่งที่ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าอัตราขยายแรงดันตกลงเหลือ 1 (Unity) ความถี่นี้ถูกเรียกว่า Unity – gain frequency : f_t ความสัมพันธ์ระหว่าง Cutoff frequency : fc กับ f_t สามารถแสดงได้โดย

$$f_t = a.fc \quad \text{หรือก็คือ} \quad fc = \frac{f_t}{a} \quad \text{นั่นเอง}$$

จากสมการแสดงให้เห็นว่า ค่า unity – gain frequency : f_t อาจถูกเรียกว่า ผลคูณของ gain กับ bandwidth ของออปแอมป์หรือ Gain – bandwidth product นั่นเอง และจากความสัมพันธ์นี้จะเห็นได้ว่า ยิ่งค่า Voltage differential gain : a ซึ่งเป็น Open-loop voltage gain มีค่าสูงมากเท่าไรก็ยิ่งทำให้ค่าความถี่คัทออฟของออปแอมป์มีค่าน้อยลงเท่านั้น ทำให้ออปแอมป์มีผลตอบสนองความถี่ได้ไม่ดี แต่เมื่อเราทำการต่อออปแอมป์ให้มีการป้อนกลับโดยใช้ feedback resistors ค่าอัตราขยายแรงดันจะลดลงกลายเป็น Close loop voltage gain : A ผลคือจะทำให้ออปแอมป์มีค่าความถี่คัทออฟ : fc ได้สูงขึ้นด้วยดังสมการ

$$fc = \frac{f_t}{A}$$

3.5.2) Slew Rate : SR

องค์ประกอบที่สำคัญอีกอันหนึ่งที่มีผลต่อออปแอมป์ในด้านการปฏิบัติการกับสัญญาณที่มีค่าแรงดันเปลี่ยนแปลงตลอดเวลาก็คือ Slew Rate

Slew rate คือ อัตราการเปลี่ยนแปลงแรงดันสูงสุดที่เอาท์พุทของออปแอมป์ มีหน่วยเป็น volts per microsecond (V/μs)

$$SR = \frac{\Delta Vo}{\Delta t} \quad V/\mu s \quad (\text{โดย t มีหน่วยเป็น } \mu s)$$

ดังนั้นจะเห็นได้ว่าค่าความถี่สูงสุดที่ออปแอมป์สามารถทำงานได้นั้น นอกจากจะขึ้นอยู่กับค่า gain - bandwidth product แล้วยังขึ้นอยู่กับค่า slew rate ของออปแอมป์อีกด้วย สำหรับสัญญาณรูปซายน์ ซึ่งมีสมการเป็น Vo = Vp.sin(2πf.t) สามารถพิจารณาผลของ slew rate ได้ดังนี้

เนื่องจากอัตราการเปลี่ยนแปลงแรงดันสูงสุดของสัญญาณ = 2πf.Vp V/s

ดังนั้นในกรณีของสัญญาณรูปซายน์ เพื่อป้องกันการเกิด distortion ของสัญญาณที่เอาท์พุทของออปแอมป์ อัตราการเปลี่ยนแปลงนี้จะต้องน้อยกว่าค่า slew rate นั่นคือ

$$2\pi f.Vp \leq SR$$

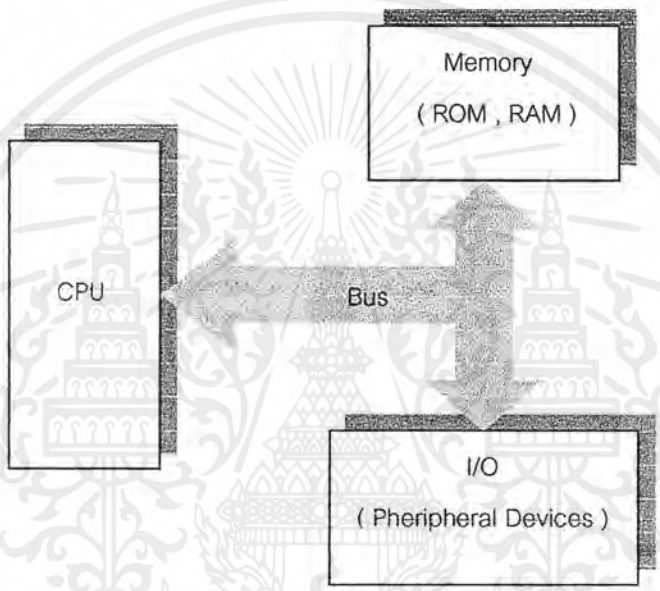
$$\therefore f_{max} = \frac{SR}{2\pi f.Vp}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โครงสร้างของคอมพิวเตอร์และการอินเตอร์เฟส

4.1) โครงสร้างพื้นฐานของคอมพิวเตอร์

โครงสร้างพื้นฐานของระบบไมโครคอมพิวเตอร์ทั่วๆไป มีองค์ประกอบหลักๆ ดังรูปต่อไปนี้



รูปที่ 4.1 โครงสร้างพื้นฐานของระบบไมโครคอมพิวเตอร์

4.1.1) Central Processing Unit (CPU) subsystem : เป็นส่วนประกอบที่สำคัญที่สุดของระบบไมโครคอมพิวเตอร์ เปรียบเทียบได้กับเป็นส่วนสมองของมนุษย์คือ มีหน้าที่คำนวณ,ประมวลผลคำสั่งหรือข้อมูลต่างๆ ในลักษณะทางคณิตศาสตร์ โดยมีส่วนควบคุมจังหวะเวลาในการทำงานขั้นพื้นฐาน (Timing) รวมอยู่ด้วย

4.1.2) Memory subsystem : เป็นส่วนที่ถูกใช้สำหรับเก็บทั้งคำสั่งและข้อมูลของระบบไมโครคอมพิวเตอร์ โดยประเภทของหน่วยความจำมี 2 ชนิดหลักๆ คือ

(1) ROM (Read Only Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งหรือข้อมูลที่ต้องการให้อยู่ในเครื่องคอมพิวเตอร์อย่างถาวร และไม่สามารถเปลี่ยนแปลงแก้ไขได้ โดยคำสั่งหรือข้อมูลที่เก็บอยู่ใน ROM นี้ จะไม่สูญหายไปถึงแม้จะมีการปิดเครื่องคอมพิวเตอร์แล้วก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) RAM (Random Access Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งหรือข้อมูลที่สามารถเปลี่ยนแปลงแก้ไขได้ตลอดเวลา โดยข้อมูลที่อยู่ใน RAM นี้จะสูญหายไปด้วยเมื่อปิดเครื่องคอมพิวเตอร์

4.1.3) Input / Output Subsystem : ส่วนนี้เป็นส่วนที่ MPU มีการติดต่อสื่อสารกับอุปกรณ์ภายนอกต่างๆ (Peripheral devices) เช่น คีย์บอร์ด ดิสก์ไดรว์ หน้าจอมอนิเตอร์ เป็นต้น และเป็นส่วนที่ทำให้ไมโครคอมพิวเตอร์สามารถติดต่อสื่อสารกับโลกภายนอกได้

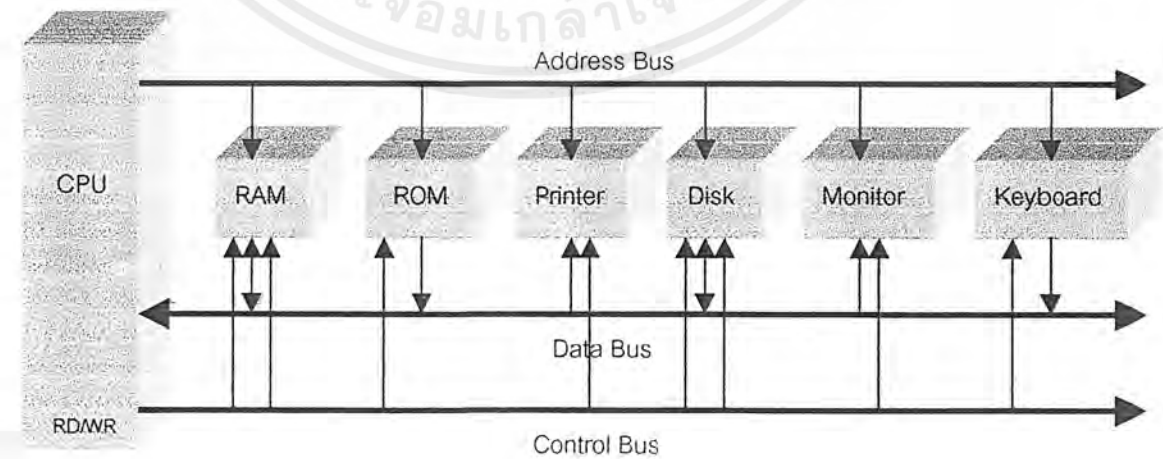
4.1.4) Bus Subsystem : ทำหน้าที่เป็นตัวกลางหรือทางเดินข้อมูลในการเคลื่อนย้ายข้อมูลระหว่างส่วน MPU , Memory ,และ Input/Output ระบบบัสแบ่งได้เป็น 3 ชนิดได้แก่

(1) Data Bus เป็นบัสที่ถูกใช้เป็นทางเดินของข้อมูลและคำสั่งต่างๆ โดยจำนวนบิตของ Data bus นี้จะมีมากหรือน้อยนั้นขึ้นอยู่กับรุ่นของ MPU(Microprocessor unit) หรือ CPU แต่ละรุ่น ซึ่งจะกล่าวถึงอีกทีในตอนหลัง

(2) Address Bus ทำหน้าที่เป็นตัวชี้ตำแหน่งแอดเดรสของหน่วยความจำ (Memory) และตำแหน่งของอุปกรณ์ภายนอกต่างๆ (I/O) โดยจำนวนบิตของหน่วย Address bus นี้จะเป็นตัวบอกถึงขนาดของหน่วยความจำที่ไมโครคอมพิวเตอร์เครื่องนั้นจะมีได้ ซึ่งขึ้นอยู่กับรุ่นของ CPU แต่ละรุ่น ซึ่งจะกล่าวถึงอีกทีในตอนหลัง

(3) Control Bus เป็นบัสที่ใช้สำหรับเคลื่อนย้ายสัญญาณควบคุมต่างๆ ที่มาจาก CPU ในการควบคุมการทำงานของส่วนต่างๆ ในระบบไมโครคอมพิวเตอร์

จากโครงสร้างพื้นฐานต่างๆที่กล่าวมาข้างต้นนี้สามารถเขียนเป็นบล็อกไดอะแกรมของโครงสร้างภายในไมโครคอมพิวเตอร์ได้ละเอียดขึ้นดังนี้

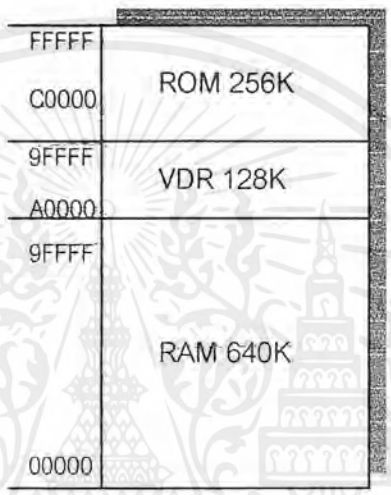


รูปที่ 4.2 ส่วนประกอบต่างๆ ของระบบไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2) Memory Map ของเครื่อง IBM PC

เมื่อซีพียู 80X86 ทำงานในแบบเรียลโหมด มันจะใช้งานแอดเดรสบัสจำนวน 20 บิต ดังนั้น จำนวนของตำแหน่งแอดเดรสของหน่วยความจำที่สามารถจัดการได้คือ 2^{20} = 1 Mbyte โดยได้มีการกำหนดตำแหน่งพื้นที่ของหน่วยความจำออกเป็น 3 ส่วนหลักๆ คือ RAM(640K) , ROM(256K) , Video Display RAM: VDR(128K) โดยมีการจัดตำแหน่งแอดเดรสของแต่ละส่วนไว้ดังนี้



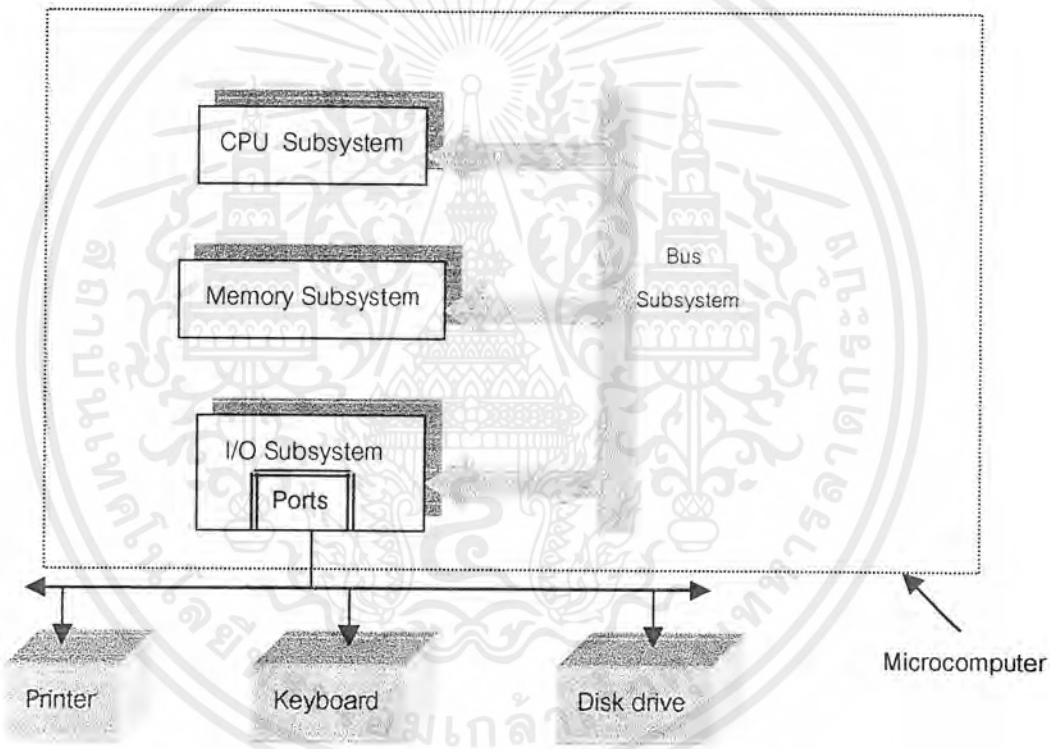
รูปที่ 4.3 Memory Map ของ IBM PC

นอกจากนี้ในส่วนของพื้นที่ของ RAM นั้นยังได้มีการถูกจัดแบ่งย่อยลงไปอีกได้แก่ ในส่วน 1K แรก (00000 – 003FF) ถูกจัดไว้สำหรับ Interrupt vector table ,ในส่วน 256 byte ถัดมา (00400 – 004FF) ถูกจัดไว้สำหรับเก็บข้อมูลชั่วคราวของ BIOS ,ส่วน 256 byte ถัดมา (00500 – 005FF) ถูกจัดสำหรับเก็บค่าพารามิเตอร์ต่างๆของ DOS และภาษา BASIC และในส่วนสุดท้ายอีกจำนวนหลาย Kbyte ที่ถูกใช้สำหรับระบบปฏิบัติการ (Operating System : OS) ของตัวมันเองซึ่งมีขนาดขึ้นกับรุ่นหรือเวอร์ชันของระบบปฏิบัติการ ดังนั้นในการเขียนโปรแกรมที่ต้องมีการติดต่อกับหน่วยความจำจึงจำเป็นที่จะต้องทราบ Memory Map ด้วยเพื่อที่จะให้โปรแกรมทำงานได้อย่างถูกต้อง และไม่ไปเขียนโปรแกรมทับลงในพื้นที่ของหน่วยความจำที่ถูกใช้งานอยู่ก่อนแล้วด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3) รายละเอียดของส่วนอินพุท / เอาท์พุท (I/O Subsystem)

จากที่ได้เคยกล่าวมาบ้างแล้วในหัวข้อ 4.1 เกี่ยวกับส่วนอินพุท / เอาท์พุท ของระบบไมโครคอมพิวเตอร์ว่า มีหน้าที่รับผิดสำหรับการรับส่ง เคลื่อนย้ายข้อมูลระหว่างระบบไมโครคอมพิวเตอร์กับอุปกรณ์เสริมภายนอก (Peripheral , External devices) มายังตัวมัน (I/O Subsystem) โดยจะมีการรับส่งข้อมูลกับ Peripheral devices ผ่านวงจรอินเตอร์เฟสที่เรียกว่า พอร์ท **(Port)** ซึ่งอุปกรณ์เสริมภายนอกเหล่านี้จะถูกต่อเข้ากับพอร์ท และพอร์ทก็จะต่อเข้ากับวงจรควบคุมการอินเตอร์เฟสดังรูป



รูปที่ 4.4 Microcomputer with I/O Subsystem

4.4) การกำหนดหมายเลขพอร์ตอินพุท / เอาท์พุต (I/O Map)

เพื่อให้โครงสร้างทางฮาร์ดแวร์ของเครื่องพีซีตระกูลเอที (AT) มีลักษณะเป็นมาตรฐานเดียวกัน (Compatibility) และสามารถใช้ซอฟแวร์ร่วมกับเครื่องพีซีตระกูลเอ็กซ์ที (XT) รุ่นก่อนได้ จึงจำเป็นต้องมีการกำหนดให้โครงสร้างทางอินพุท / เอาท์พุท และหมายเลขพอร์ตที่ใช้ในการติดต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์หรือชิปเสริมต่างๆ ที่ช่วยในการทำงานให้เหมือนกัน ตารางต่อไปนี้เป็น I/O Map ของเครื่องพีซีตระกูลเอที

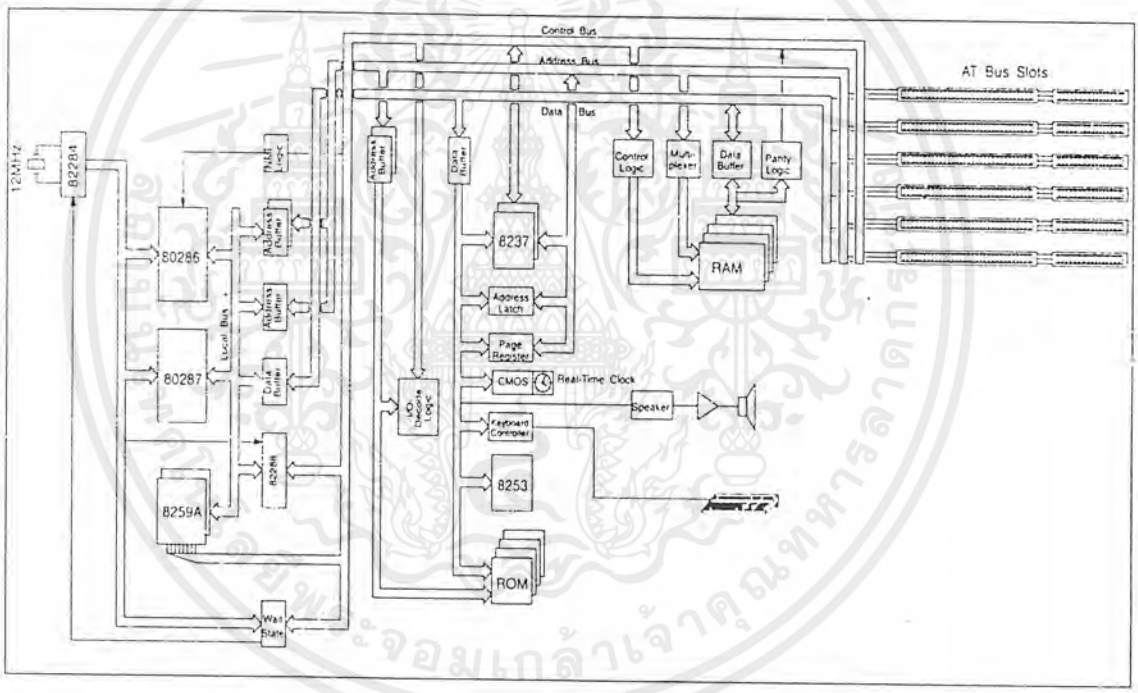
หมายเลขพอร์ต	ชื่ออุปกรณ์
000-01F	1st DMA controller , 8237A
020-03F	1st Interrupt controller , 8259A , master
040-05F	Timer , 8254-2
060-06F	Keyboard controller , 8042
070-07F	Real-time clock and NMI (Non-Maskable interrupt)
080-09F	DMA page registers
0A0-0BF	2nd Interrupt controller , 8259A , slave
0C0-0DF	2nd DMA controller , 8237A
0E0-0EF	Reserved
0F0-0FF	Reserved for Math coprocessor 80287
1F0-1F8	Fixed disk
200-207	Game adapter
20C-20D	Reserved
21F	Reserved
278-27F	Parallel printer port 2 , LPT2
2F8-2FF	Serial port 2 , COM2
300-31F	Prototype card / adapter
360-36F	Reserved
378-37F	Parallel printer port1 , LPT1
380-38F	SDLC adapter , Bisynchronous 1
3A0-3AF	Bisynchronous 1
3B0-3BF	Monochrome adapter and printer adapter
3C0-3CF	Enhanced graphics adapter , EGA
3D0-3DF	Color/graphics monitor adapter , CGA
3F0-3F7	Floppy disk controller
3F8-3FF	Serial port 1 , COM1

ตารางที่ 4.1 หมายเลขพอร์ตของเครื่องพีซีเอที (IBM PC/AT I/O Address Map)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

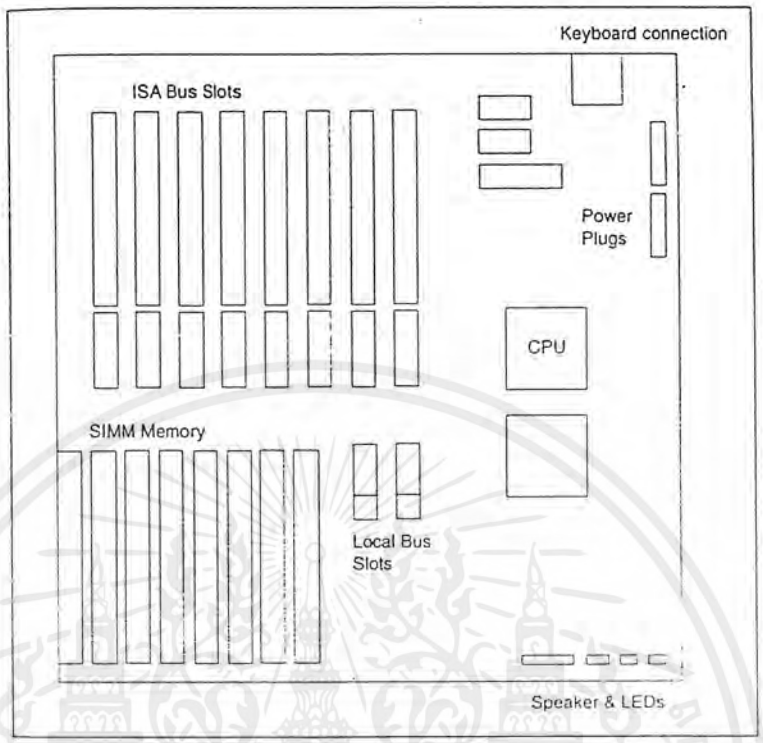
4.5) เมนบอร์ดของเครื่องพีซีตระกูลเอที

เมนบอร์ด (Main board , motherboard ซึ่งเป็นชื่อที่รู้จักกันดีของ system board) ถือเป็นส่วนประกอบที่สำคัญของระบบไมโครคอมพิวเตอร์ เมนบอร์ดเกิดจากการนำส่วนประกอบทางด้านฮาร์ดแวร์ต่างๆ เช่น หน่วยประมวลผล หน่วยความจำ ไอซีหรือชิปที่มีหน้าที่เฉพาะอย่าง มารวมเข้าด้วยกันทำให้ฮาร์ดแวร์ส่วนต่างๆ มีการทำงานร่วมกัน จนทำให้เกิดเป็นเครื่องคอมพิวเตอร์ขึ้นได้ ส่วนประกอบพื้นฐานของเมนบอร์ดได้แก่ ซีพียู หน่วยประมวลผลร่วมทางคณิตศาสตร์ Clock ของซีพียู วงจรสนับสนุนการทำงานของซีพียู หน่วยความจำ เวลาของระบบ ส่วนเชื่อมต่อกับคีย์บอร์ด ส่วนเชื่อมต่อกับอินพุท / เอาท์พุท ดังรูป



รูปที่ 4.5 บล็อกไดอะแกรมของเมนบอร์ดตระกูลเอที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงรูปทรงของเมนบอร์ดทั่วๆไป และตำแหน่งของส่วนต่างๆ

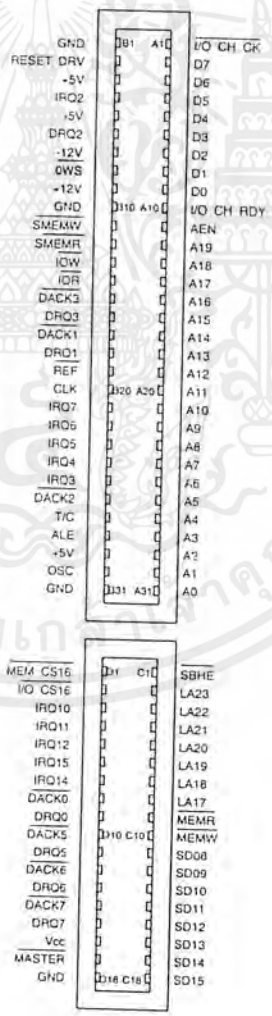
4.6) สล๊อตหรือช่องต่อสำหรับอินพุท / เอาท์พุท

เนื่องจากภายใน IBM/PC , XT ,AT ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรอินเทอร์-เฟสเข้าไปในภายหลังได้โดยผ่านทางสล๊อตที่อยู่บนเมนบอร์ด โดยในเครื่องตระกูลเอ็กซ์ทีจะมีจำนวน 8 สล๊อต ซึ่งแต่ละสล๊อตจะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็น 2 ข้างๆ ละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล๊อตเหล่านี้จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใดของสล๊อต โดยขาที่อยู่ทางด้านซ้ายจะใช้อักษร B นำหน้าเลขตำแหน่งของขา ส่วนขาที่อยู่ทางด้านขวาของสล๊อตจะเรียกโดยใช้อักษร A นำหน้าเลขตำแหน่งของขา ต่อมาเมื่อพัฒนาเป็นเครื่องเอทีทำให้มีขีดความสามารถเพิ่มขึ้น เช่น มี Data bus 16 bits เป็นต้น จึงจำเป็นต้องปรับปรุงสล๊อตเพิ่มเติมและสามารถใช้กับวงจรของเดิมได้ (เอ็กซ์ที) โดยการเพิ่มสล๊อตเสริมขึ้นมาซึ่งมีจำนวนขาเพิ่มขึ้นมาอีก 36 ขา โดยขาที่อยู่ทางด้านซ้ายจะใช้อักษร D นำหน้าเลขตำแหน่งของขา ส่วนขาที่อยู่ทางด้านขวาของสล๊อตจะเรียกโดยใช้อักษร C

แต่ละขาของสล๊อตเหล่านี้จะเชื่อมต่อกับเส้นสัญญาณต่างๆ บนเมนบอร์ด ทำให้การสร้างวง-จรอินเตอร์เฟสกับเครื่องพีซีทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล๊อตเหล่านี้ประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บัสแอดเดรส (Address Bus)
- บัสข้อมูล (Data Bus)
- บัสควบคุมสำหรับการเขียน / อ่านข้อมูลจากหน่วยความจำ หรือพอร์ต I/O
- เส้นสัญญาณสำหรับการขออินเทอร์รัพท์ของวงจรอินเตอร์เฟส
- เส้นสัญญาณสำหรับการขอ DMA
- เส้นสัญญาณฐานเวลา (Timing signal) ต่างๆ
- เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ
- เส้นสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHCK)
- ขาไฟเลี้ยง +5V , -5V , +12V , -12V



รูปที่ 4.7 ตำแหน่งของขาสัญญาณต่างๆ บนสล๊อตของเครื่องพีซีตระกูลเอทีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ต่อเชื่อมกับอุปกรณ์อินพุท / เอาท์พุท เป็นสัญญาณที่มีขนาด 5 V ตามมาตรฐาน TTL โดยที่แต่ละสล๊อตจะเชื่อมต่อกับ LS TTL ได้ 2 อินพุท ดังนั้นการต่อกับสล๊อตจึงจำเป็นต้องคำนึงถึงโหลดดังกล่าวนี้ด้วย สัญญาณที่ขาต่างๆ ของสล๊อตมีความหมายดังนี้

OSC (Oscillator) : O/P

เป็นขาที่ต่อกับสัญญาณนาฬิกาของ Oscillator ความถี่ 14.31818 MHz และมี Duty Cycle 50% สัญญาณนี้จะไม่ synchronize กับระบบ

CLK (Clock) : O/P

เป็นสัญญาณ clock ของระบบมีความถี่ 6 MHz หรือสูงกว่านี้ในรุ่นใหม่ๆ สัญญาณนี้จะ synchronize กับระบบ

RESET DRV : O/P

จะแอคทีฟที่ลอจิก 1 เป็นขาที่ใช้สำหรับรีเซตระบบในขณะเปิดเครื่อง หรือขณะที่แหล่งจ่ายไฟเลี้ยงขาด หรือไฟตก

$A_0 - A_{19}$ **:** I/O

เป็นขาแอดเดรสของระบบใช้สำหรับกำหนดแอดเดรสของหน่วยความจำ และ I/O สัญญาณนี้จะใช้ได้ต่อเมื่อ ALE เป็น 1 โดยจะถูกแลตช์ไปใช้เมื่อ ALE เปลี่ยนจาก 1 เป็น 0 (ALE เป็นสัญญาณที่มาจากซีพียู หรือ DMA Controller)

$LA_{17} - LA_{23}$ **:** I/O

เป็นสัญญาณแอดเดรสที่ไม่ผ่านการแลตช์มาเลยจากซีพียู ใช้ในการต่อขยายหน่วยความจำ สัญญาณนี้จะใช้ได้ต่อเมื่อ ALE เป็น 1

$D_0 - D_{15}$ **:** I/O

เป็นสัญญาณข้อมูลขนาด 16 บิต ที่ใช้ติดต่อกับหน่วยความจำ และ I/O

ALE (Address latch enable) : O/P

เป็นสัญญาณที่ใช้สำหรับการแลตช์แอดเดรสของระบบ

AEN (Address enable) : O/P

เป็นสัญญาณเพื่อใช้ในการแยกบัสแอดเดรสในขบวนการ DMA กล่าวคือ ถ้า AEN = 1 DMA controller กำลังควบคุมการทำงานของแอดเดรสบัส ถ้า AEN = 0 ซีพียูกำลังควบคุมการทำงานของแอดเดรสบัสอยู่

$\overline{\text{REF}}$ (Refresh) : I/O

ใช้ในการแสดงสัญญาณรีเฟรชไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

T/C (Terminal Count) : O/P

เป็นสัญญาณพลส์เมื่อ DMA นับจำนวนครบตามที่กำหนด

$\overline{\text{SBHE}}$ (System Bus High Enable) : I/O

สัญญาณนี้จะแอคทีฟเมื่อมีกระบวนการถ่ายเทข้อมูลจาก D0-D15 เข้าสู่บัฟเฟอร์

MASTER : I/P

สัญญาณนี้มีจุดมุ่งหมายใช้บอกว่า ต้องการให้บัสระบบถูกควบคุมโดยซีพียู หรือว่าจะให้ซีพียูส่งอำนาจการควบคุมให้กับสล๊อตที่มีการ์ดต่อเข้ามาในระบบซึ่งสามารถทำได้โดยการแอคทีฟขาสัญญาณนี้ (แอคทีฟ 0)

$\overline{\text{I/O CHK}}$ (I/O Channel Check) : I/P

เป็นสัญญาณตรวจสอบของอินพุท / เอาท์พุท เพื่อบอกข้อมูลกับระบบเช่นเดียวกับการตรวจสอบพาริตี้ ดังนั้นหากอินพุท / เอาท์พุทมีข้อผิดพลาด ก็จะมีผลให้ขาสัญญาณนี้แอคทีฟ

I/O CHRDY (I/O Channel Ready) : I/P

สัญญาณนี้จะได้รับการทำให้แอคทีฟโดยหน่วยความจำหรืออุปกรณ์อินพุท / เอาท์พุท การใช้สัญญาณนี้ก็เพื่อให้อุปกรณ์อินพุท / เอาท์พุทที่ช้าจะได้ติดต่อกับระบบด้วยการส่งสัญญาณมายังซีพียู เพื่อที่จะสามารถ synchronize กับระบบโดยการเพิ่มเวลาในบัสไซเคิลเข้าไป (Wait state)

$\overline{\text{0WS}}$ (0 Wait State): I/P

เป็นสัญญาณที่ใช้บอกซีพียูว่า การทำงานในหนึ่งรอบของบัสไม่จำเป็นต้องแทรกสถานะรอ หรือ Wait state เข้าไป

IRQ3-IRQ7 , IRQ9-IRQ12 , IRQ14-IRQ15 : I/P

เป็นขาสัญญาณอินเทอร์รัพที่ต่อมาจากชิปควบคุมการอินเทอร์รัพท์ 8259 ทั้ง 2 ตัว ซึ่งได้กล่าวไปแล้ว ทำให้สามารถรับสัญญาณอินเทอร์รัพท์ที่ส่งมาจากการ์ดผ่านทางสล๊อตได้

DRQ0-DRQ3 , DRQ5-DRQ7 : I/P และ **$\overline{\text{DACK0}}$-$\overline{\text{DACK3}}$, $\overline{\text{DACK5}}$-$\overline{\text{DACK7}}$:** O/P

DRQ เป็นสัญญาณการขอกระบวนการทำงานแบบ DMA จากชิปควบคุมการทำ DMA เบอร์ 8237 ส่วน DACK เป็นสัญญาณที่ส่งมาจาก 8237 เพื่อบอกให้ทราบว่าการขอ DMA ได้รับการตอบสนองแล้ว

$\overline{\text{IOR}}$: I/O

เป็นสัญญาณที่ใช้ในการอ่านอินพุท / เอาท์พุท การควบคุมสัญญาณนี้สามารถควบคุมได้โดยซีพียู หรือชิปควบคุมการทำ DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{IOW}}$: I/O

เป็นสัญญาณที่ใช้ในการเขียนข้อมูลลงบนอุปกรณ์ อินพุท / เอาท์พุท การควบคุมสัญญาณนี้สามารถควบคุมได้โดยซีพียู หรือชิปควบคุมการทำ DMA

$\overline{\text{SMEMR}}$ (O/P) , **$\overline{\text{MEMR}}$** (I/O)

เป็นสัญญาณควบคุมการอ่านข้อมูลการอ่านข้อมูลจากหน่วยความจำโดย SMEMR ถูกใช้สำหรับติดต่อข้อมูลกับหน่วยความจำในส่วน 1Mbytes แรก ส่วน MEMR ถูกใช้ในการติดต่อกับหน่วยความจำได้ทั้งหมด 16 Mbytes

$\overline{\text{SMEMW}}$ (O/P) , **$\overline{\text{MEMW}}$** (I/O)

เป็นสัญญาณควบคุมการอ่านข้อมูลการเขียนข้อมูลลงหน่วยความจำโดย SMEMW ถูกใช้สำหรับติดต่อข้อมูลกับหน่วยความจำในส่วน 1Mbytes แรก ส่วน MEMW ถูกใช้ในการติดต่อกับหน่วยความจำได้ทั้งหมด 16 Mbytes

$\overline{\text{MEM CS16}}$: I/P

เป็นสัญญาณที่ส่งมาจากการ์ดผ่านทางสล๊อต เพื่อใช้ในการบอกกับเครื่องพีซีว่าต้องการติดต่อกับหน่วยความจำในแบบ 8 บิต หรือ 16 บิต

$\overline{\text{I/O CS16}}$: I/P

เป็นสัญญาณที่ส่งมาจากการ์ดผ่านทางสล๊อต เพื่อใช้ในการบอกกับเครื่องพีซีว่าให้มีการติดต่อกับอุปกรณ์อินพุท / เอาท์พุทเป็นแบบ 8 บิต หรือ 16 บิต

โดยบัสหรือสายสัญญาณต่างๆ ที่ต่อออกมาผ่านทางทางสล๊อตแบบนี้ถูกเรียกว่า บัส ISA (Industry Standard Architecture) ซึ่งเป็นระบบบัสมาตรฐานแบบหนึ่ง จะเห็นได้ว่าการใช้สล๊อตแบบนี้ทำให้อุปกรณ์ภายนอกใช้บัสข้อมูลได้สูงสุด 16 บิต ถึงแม้ซีพียูจะเป็นแบบ 32 บิตก็ตาม ดังนั้นในการออกแบบการ์ดอินเตอร์เฟสจึงสามารถใช้ดาต้าบัสได้เพียง 8 หรือ 16 บิตเท่านั้น ไม่สารมารถใช้ได้ถึง 32 บิทได้ แต่ถ้าต้องการติดต่อกับซีพียูได้ครบ 32 บิท อาจต้องไปใช้ EISA (Extended ISA) bus , PCI (Peripheral Component Interfce) bus , MCA (Micro Channel Architecture) bus แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

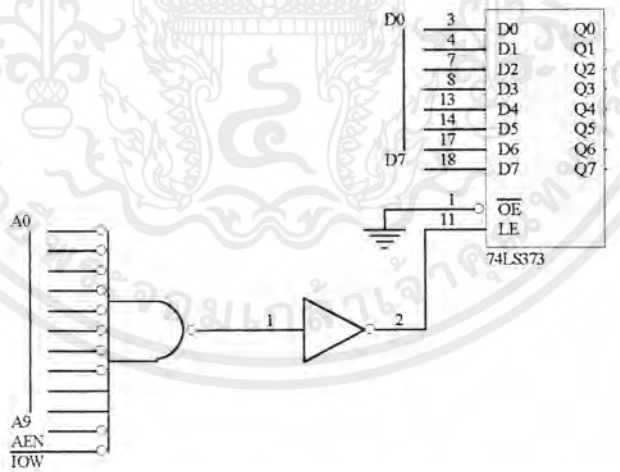
4.7) การอินเตอร์เฟส I/O กับ ISA bus ผ่านทางสล๊อต

จาก I/O Map ของเครื่อง IBM/PC , XT,AT และเครื่อง Compatible ในตารางที่ 4.2 จะเห็นว่าตำแหน่งแอดเดรสของอุปกรณ์อินพุท / เอาท์พุท ที่ถูกกำหนดไว้สำหรับการ์ดอินเตอร์เฟสต่างๆ ที่ต่อผ่านทางสล๊อตนั้นอยู่ในช่วง 300H – 31FH (สามารถมีได้ทั้งหมด 32 อุปกรณ์) ในการอินเตอร์เฟสกับเครื่องพีซีโดยใช้การ์ดเสริม สัญญาณพื้นฐานที่จำเป็นในการอินเตอร์เฟสมีดังนี้

- สัญญาณ $\overline{IOR}$, $\overline{IOW}$
- สัญญาณ AEN (AEN = 0 เมื่อซีพียูเป็นตัวควบคุมระบบบัส)
- สัญญาณ $A_0 – A_9$ สำหรับ Address decoding (ใน IBM/PC มีการใช้แอดเดรสบัสเพียง 10 เส้นสำหรับการถึงแอดเดรสของพอร์ต)
- สัญญาณ $D_0 – D_7$ (สำหรับข้อมูลขนาด 8 บิท)

4.7.1) การอินเตอร์เฟสกับพอร์ตเอาท์พุท

อาจทำได้โดยการ โดยการนำสัญญาณ A0 - A9 , $\overline{IOW}$, AEN มาต่อเข้ายังอินพุทของ AND gate แล้วจึงนำเอาท์พุทของ AND gate ไปต่อเข้ากับขา Enable ของวงจร Latch ดังรูป



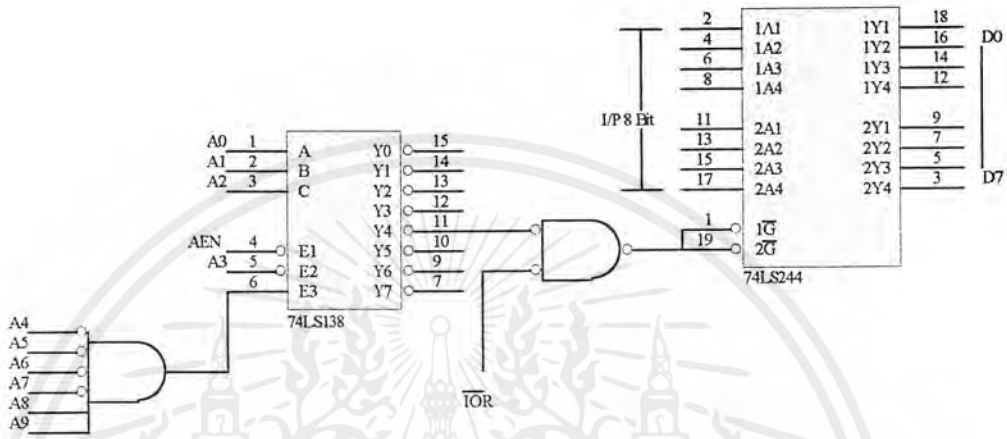
รูปที่ 4.8 ตัวอย่างการต่อวงจรอินเตอร์เฟสกับพอร์ทเอาท์พุท

จากรูปเป็นการใช้ AND gate ในการเลือกพอร์ตหมายเลข 300H (11 0000 0000) ซึ่งเราอาจใช้วงจร Decoder เป็นตัวเลือกหมายเลขพอร์ตแทนก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.2) การอินเตอร์เฟสกับพอร์ตอินพุท

สามารถทำได้เช่นเดียวกันกับการอินเตอร์เฟสกับพอร์ตเอาท์พุท แต่เปลี่ยนจากการการใช้วงจร Latch มาใช้ Buffer แทนดังรูป



รูปที่ 4.9 ตัวอย่างการต่อวงจรอินเตอร์เฟสกับพอร์ตอินพุท

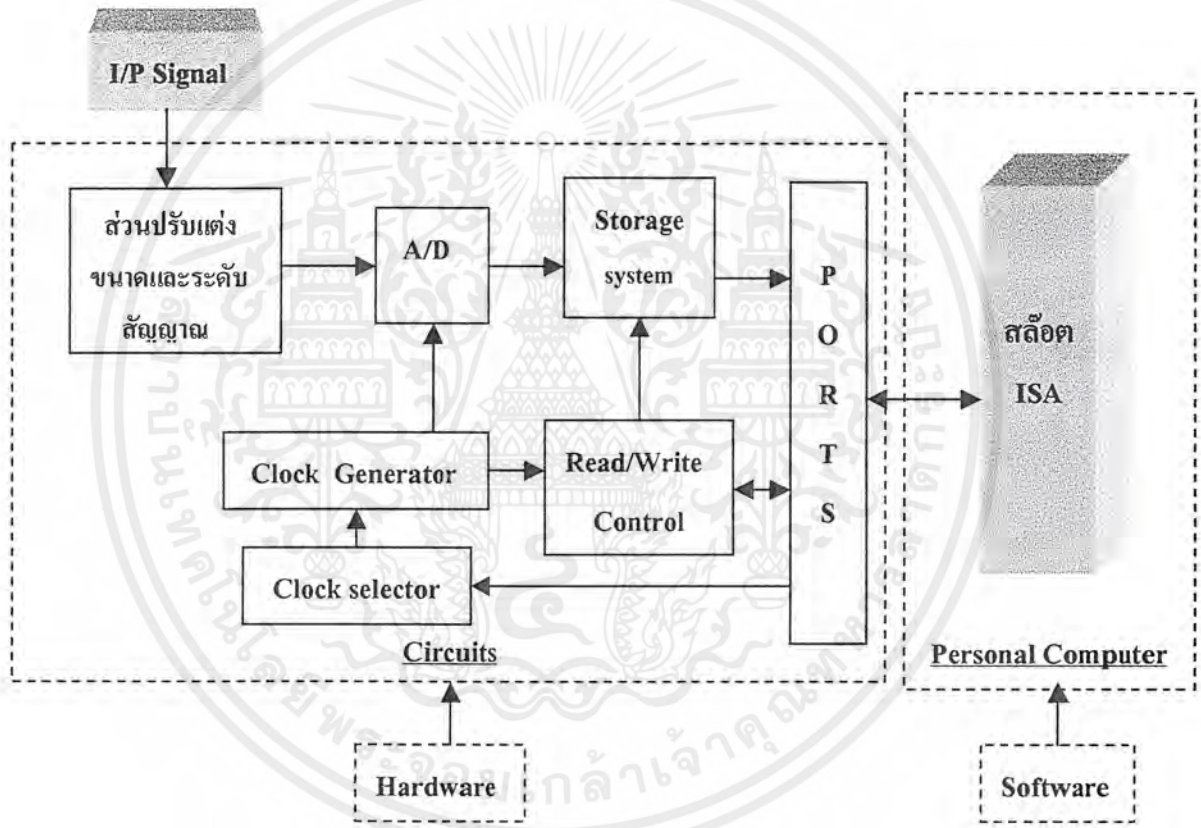
จากรูปเป็นการต่อวงจรเลือกพอร์ตอินพุทหมายเลข 304H (11 0000 0100) โดยใช้ Decoder และ AND gate ต่อร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบวงจรส่วนต่างๆ

5.1) โครงสร้างและส่วนต่างๆ โดยภาพรวม

ส่วนประกอบโดยภาพรวมของโครงงานออสซิลโลสโคปบนคอมพิวเตอร์ (Oscilloscope on PC) นี้ สามารถแสดงเป็นบล็อกไดอะแกรมได้ดังนี้



รูปที่ 5.1 บล็อกไดอะแกรมแสดงโครงสร้างส่วนต่างๆ

จากรูปที่ 5.1 จะเห็นว่าต้องอาศัยทั้งด้านการออกแบบวงจรหรือด้านฮาร์ดแวร์ (Hardware) และด้านการเขียนโปรแกรมหรือด้านซอฟแวร์ (Software) หลักการทำงานคือ จะทำการรับสัญญาณอินพุทซึ่งเป็นสัญญาณอนาลอกที่ต้องการจะวัดเข้ามา แล้วจึงทำการแปลงสัญญาณอินพุทที่เป็นสัญญาณอนาลอกนี้ให้เป็นสัญญาณดิจิตอลโดยใช้วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital Converter) นำข้อมูลนี้ไปเก็บในหน่วยความจำภายนอกก่อน เมื่อหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอกเก็บข้อมูลจนเต็มแล้ว ก็จะส่งสัญญาณไปบอกคอมพิวเตอร์ให้ทราบ โปรแกรมที่เขียนเอาไว้ก็จะสั่งการให้เครื่องคอมพิวเตอร์ทำการอ่านข้อมูลที่เก็บไว้ในหน่วยความจำภายนอกนี้เข้าไปโดยผ่านทางสล๊อต ISA เพื่อนำไปเก็บไว้ในหน่วยความจำภายในอีกที จากนั้นโปรแกรมจึงนำข้อมูลที่เก็บไว้ในหน่วยความจำภายในนี้ ไปวาดเป็นรูปสัญญาณออกมาที่หน้าจอคอมพิวเตอร์

จากรูปที่ 5.1 จะเห็นว่าในส่วนของฮาร์ดแวร์นั้นประกอบด้วย ส่วนปรับแต่งขนาดสัญญาณ , ส่วน A/D , หน่วยความจำภายนอกหรือ SRAM , ส่วนสร้างสัญญาณนาฬิกา , ส่วนเลือกความถี่ในการ Sampling , ส่วนควบคุมการอ่าน / เขียนข้อมูลใน SRAM , และส่วนถอดรหัสแอดเดรสของบัฟเฟอร์ Address Decoding) ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

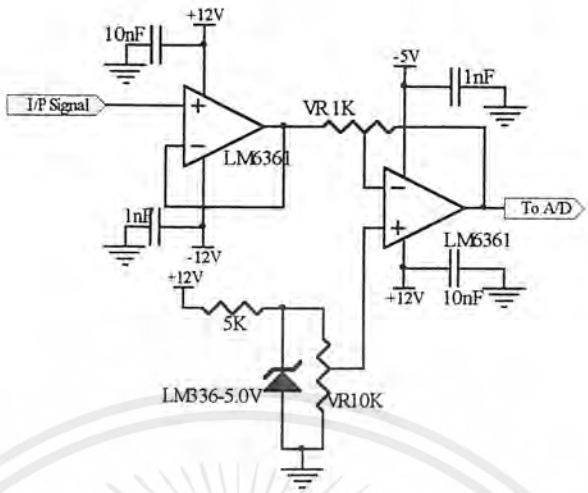
5.2) ส่วนปรับแต่งขนาดและยกระดับสัญญาณอินพุท

สาเหตุที่ต้องมีส่วนนี้ก็เนื่องจาก ในส่วนแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (A/D) นั้นได้มีการใช้วงจรรวมสำเร็จรูป (Integrated Circuit : IC) แทนการสร้างวงจร A/D เอง (เพราะสร้างได้ยากถ้าต้องการความละเอียดสูง) ซึ่งไอซี A/D ที่เลือกใช้นี้คือเบอร์ TDA 8703 ซึ่งเป็น 8-bit high-speed analog-to-digital converter ของ PHILIPS แต่ก็มีข้อจำกัดคือ

- ไอซี A/D ที่ใช้นี้สามารถแปลงแรงดันของสัญญาณอนาลอกได้อยู่ช่วงหนึ่งเท่านั้น (รายละเอียดสามารถดูได้จาก Datasheet ของไอซี A/D ซึ่งอยู่ในภาคผนวก)
- ไอซี A/D ที่ใช้นี้ไม่สามารถแปลงแรงดันของสัญญาณอนาลอกที่มีค่าเป็นลบได้

ข้อจำกัดข้างต้นนี้สามารถแก้ไขได้โดย ทำการปรับหรือลดทอนขนาดของสัญญาณอินพุท และยกระดับแรงดันสัญญาณอินพุทเพื่อให้มีแต่ระดับแรงดันที่เป็นบวกเท่านั้น (ระดับแรงดันที่ยกขึ้นมานั้นขึ้นอยู่กับไอซี A/D แต่ละเบอร์) เราสามารถออกแบบวงจรส่วนนี้ได้โดยใช้ออปแอมป์ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 วงจรปรับแต่งและยกระดับสัญญาณอินพุท

จากรูปที่ 5.2 ออปแอมป์เบอร์ LM6361 เป็น High Speed Operational Amplifier (ดูรายละเอียดได้จาก Datasheet ในภาคผนวก) ออปแอมป์ตัวแรกซึ่งมีอัตราขยายแรงดันเป็น 1 (R2 = 0 , R1 = ∞) ทำหน้าที่เป็นบัฟเฟอร์เพื่อให้วงจรส่วนนี้มีความต้านทานที่ทางเข้าสูงๆ (High input impedance) เพื่อป้องกันผลของ Loading effect อันเกิดจากจุดที่เราต้องการจะวัดสัญญาณนั้นมี ความต้านทานที่ทางออกค่อนข้างสูง (High output impedance) ดังนั้นหากวงจรส่วนแรกของเรามีความต้านทานที่ทางเข้าไม่สูงพอก็ทำให้เกิดความผิดพลาดในการวัดขึ้นได้ เราสามารถหาค่าความต้านทานที่ทางเข้า : Rin ของบัฟเฟอร์ได้โดยวิธีในหัวข้อ 3.4.1 ดังนี้

$$\text{จาก } Rin = rd\left(1 + \frac{a}{1 + (R2+ro)/R1}\right) + R1 \| (R2 + ro) \qquad \text{โดย } R2 = 0 \text{ , } R1 = \infty$$

ดังนั้น $Rin = rd\,(1 + a) + ro$ จาก Datasheet ของ LM6361 จะได้ $rd_{min} \approx 325\ k\Omega$, $ro_{max} \approx 20k\Omega$, $a \approx 350$ V/V

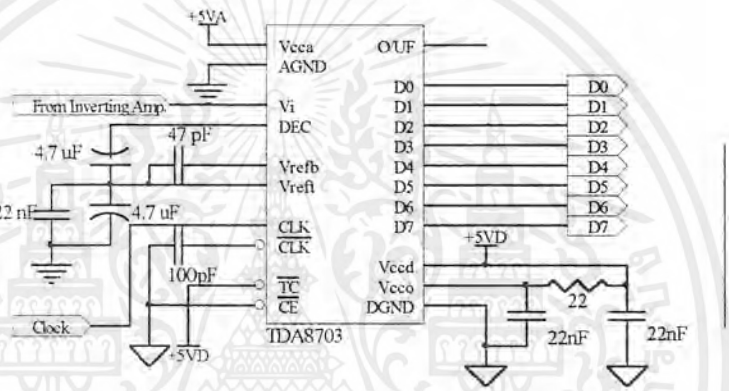
$\therefore$ ค่าความต้านทานที่ทางเข้า : $Rin \approx 114\ M\Omega$

ส่วนออปแอมป์ตัวที่ 2 ต่อแบบ Inverting Amplifier ที่สามารถปรับอัตราขยายของวงจรได้จาก VR $1k\Omega$ และที่ขาบวกของออปแอมป์จะต่อเข้ากับแรงดันบวกค่าหนึ่งที่มาจาก Voltage - reference LM336-5.0V เพื่อทำการยกระดับสัญญาณที่เอาท์พุทขึ้นมา Vout ที่ออกมาจากวงจร Inverting Amplifier นี้จะเป็นสัญญาณอินพุทให้แก่ A/D ในส่วนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3) ส่วนแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

ในส่วนของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (A/D) ที่ใช้ในโครงงานชิ้นนี้ จะใช้ชิปวงจรรวม A/D แบบสำเร็จรูปเบอร์ TDA8703 ของ PHILIPS ซึ่งมีคุณสมบัติที่สำคัญคือ มีความละเอียด 8 บิต , สามารถป้อนสัญญาณนาฬิกาสำหรับการสุ่ม (Sampling Clock) ได้ความถี่สูงสุดถึง 40 MHz , มีวงจรสุ่มสัญญาณอยู่ภายในตัว (Internal Sampling & Hold Circuit) ทำให้ไม่ต้องสร้างวงจรสุ่มสัญญาณอนาลอกภายนอก , ใช้ระดับแรงดันอ้างอิงภายในตัวไอซีเอง (นอกจากนี้ยังมีรายละเอียดและคุณสมบัติอื่นๆ อีกมากซึ่งสามารถดูได้จาก Datasheet ของไอซีเบอร์นี้ซึ่งอยู่ในภาคผนวก) สำหรับการต่อเพื่อใช้งานสามารถต่อตาม Datasheet ได้ดังรูปที่ 5.3



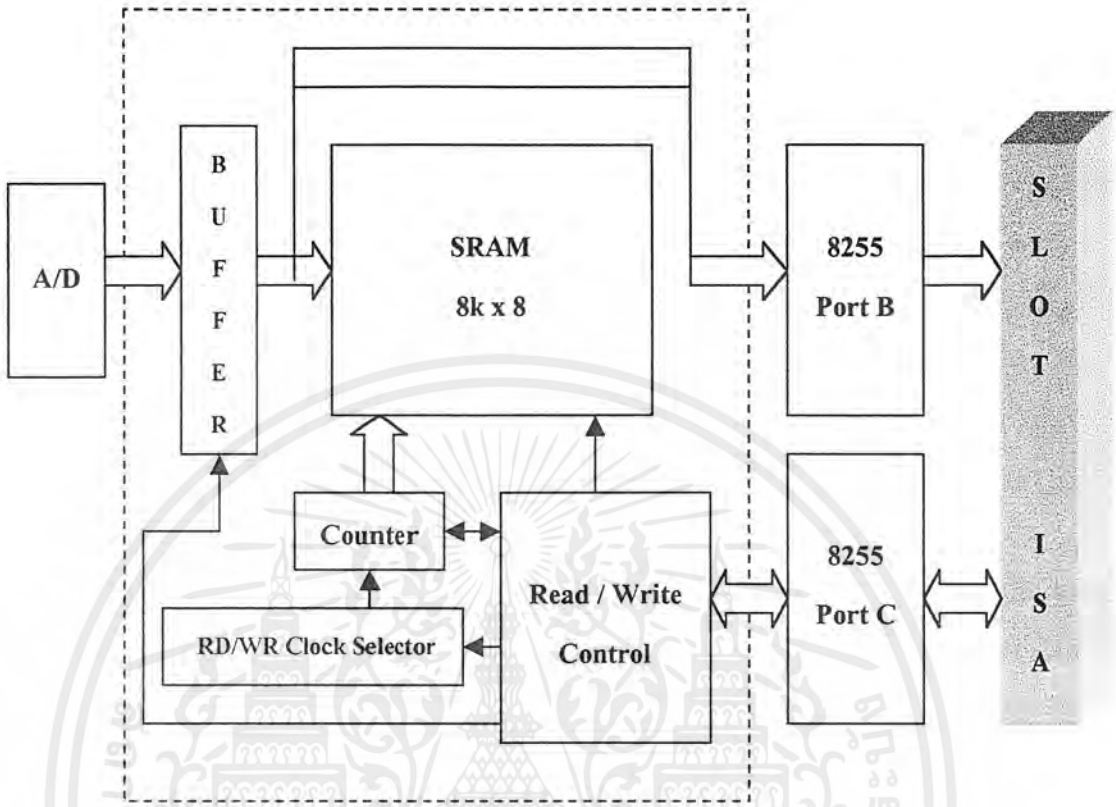
รูปที่ 5.3 การต่อไอซี A/D เบอร์ TDA8703 เพื่อใช้งาน

จากรูปที่ 5.3 เป็นการต่อไอซี A/D เพื่อใช้งานโดยขา Vi ของไอซีจะถูกต่อเข้ากับขา Vout ของวงจร Inverting Amplifier ในรูปที่ 5.2 ส่วนขา CLK ของไอซี A/D จะถูกต่อเข้าวงจรกำเนิดสัญญาณนาฬิกาซึ่งจะกล่าวถึงอีกทีในภายหลัง และขา $D_0 - D_7$ นี้คือสัญญาณดิจิตอลที่มีความละเอียด 8 บิต ซึ่งจะถูกต่อเข้ากับบัฟเฟอร์ในส่วนถัดไป

5.4) ส่วนอ่าน / เขียนข้อมูลใน SRAM และการควบคุม

อาจถือได้ว่าเป็นส่วนที่สำคัญที่สุดของการออกแบบทางด้านฮาร์ดแวร์ เนื่องจากการที่มีหน่วยความจำภายนอกซึ่งเป็น Static RAM : SRAM เป็นตัวเก็บข้อมูลที่มาจาก A/D เอาไว้ก่อน แล้วค่อยส่งเข้าคอมพิวเตอร์ในภายหลังนั้น ทำให้เราสามารถวัดสัญญาณที่มีความถี่สูงๆ ได้ดีกว่าการออกแบบที่ไม่มี SRAM เก็บข้อมูลเอาไว้ก่อนมาก ส่วนประกอบต่างๆ สามารถแสดงได้โดยบล็อกไดอะแกรมดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 บล็อกไดอะแกรมแสดงส่วนควบคุมการอ่าน / เขียนของ SRAM

จากรูปที่ 5.4 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

ขั้นตอนที่ 1 : โปรแกรมที่เขียนเอาไว้จะสั่งให้ระบบทำการเก็บ (Write data) ข้อมูลที่มาจาก A/D ในส่วนข้างหน้าลงในแต่ละแอดเดรสของ SRAM โดย

- ส่วน RD / WR control จะสั่งให้ SRAM อยู่ในสถานะ Write data เพื่อทำการเก็บข้อมูล และทำการ Reset วงจรนับให้เริ่มนับที่ 0
- ทำการ Disable ขา Tri – state output ของ SRAM
- ทำการ Enable บัฟเฟอร์ให้ข้อมูลจาก A/D สามารถผ่านเข้ามาถึงขา data ของ SRAM ได้
- ส่วน RD / WR clock selector จะทำการเลือก clock ซึ่งเป็นตัวเดียวกับ Sampling clock ของ A/D มาเป็น clock ของวงจรนับเพื่อใช้เลื่อนแอดเดรสของ SRAM ในการเก็บข้อมูล

เมื่อ A/D แปลงสัญญาณออกมาทีละครั้ง ตาม Sampling clock วงจรนับก็จะเพิ่มค่าทีละ 1 ตามด้วยเช่นกันเพื่อเก็บข้อมูลที่ถูกแปลงมาแต่ละครั้งลงใน SRAM จนเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 : เมื่อข้อมูลถูกเขียนลงใน SRAM จนเต็ม (สามารถตรวจสอบได้จากขาแอด - เดรสบัสที่ต่อกับวงจรนับว่า ขณะนั้นหมายเลขแอดเดรสที่ถูกนับเป็นแอดเดรสสุดท้ายของ SRAM หรือไม่) ขั้นตอนนี้จะเป็นการสั่งให้ระบบเตรียมพร้อมสำหรับการขนถ่ายข้อมูลจาก SRAM ไปยังหน่วยความจำภายในคอมพิวเตอร์ โดย

- ส่วน RD / WR control จะสั่งให้ SRAM เปลี่ยนสถานะเป็นการ Read data เพื่อเตรียมพร้อมสำหรับกระบวนการอ่านข้อมูล
- ทำการ Disable บัฟเฟอร์ เพื่อไม่ให้ข้อมูลจาก A/D ผ่านเข้ามาถึงขา data ของ SRAM ได้อีก เพราะในขณะนี้จะมีข้อมูลถูกอ่านออกมาจาก SRAM
- ทำการ Enable ขา Tri – state output ของ SRAM เพื่อให้ข้อมูลที่ถูกเก็บเอาไว้ออกมาได้
- ส่วน RD / WR clock selector จะทำการเลือก clock ที่มาจากคอมพิวเตอร์ (โดยการเขียนโปรแกรม) ให้แก่วงจรนับเพื่อใช้เลื่อนแอดเดรสของ SRAM ในการอ่านข้อมูลออกมา

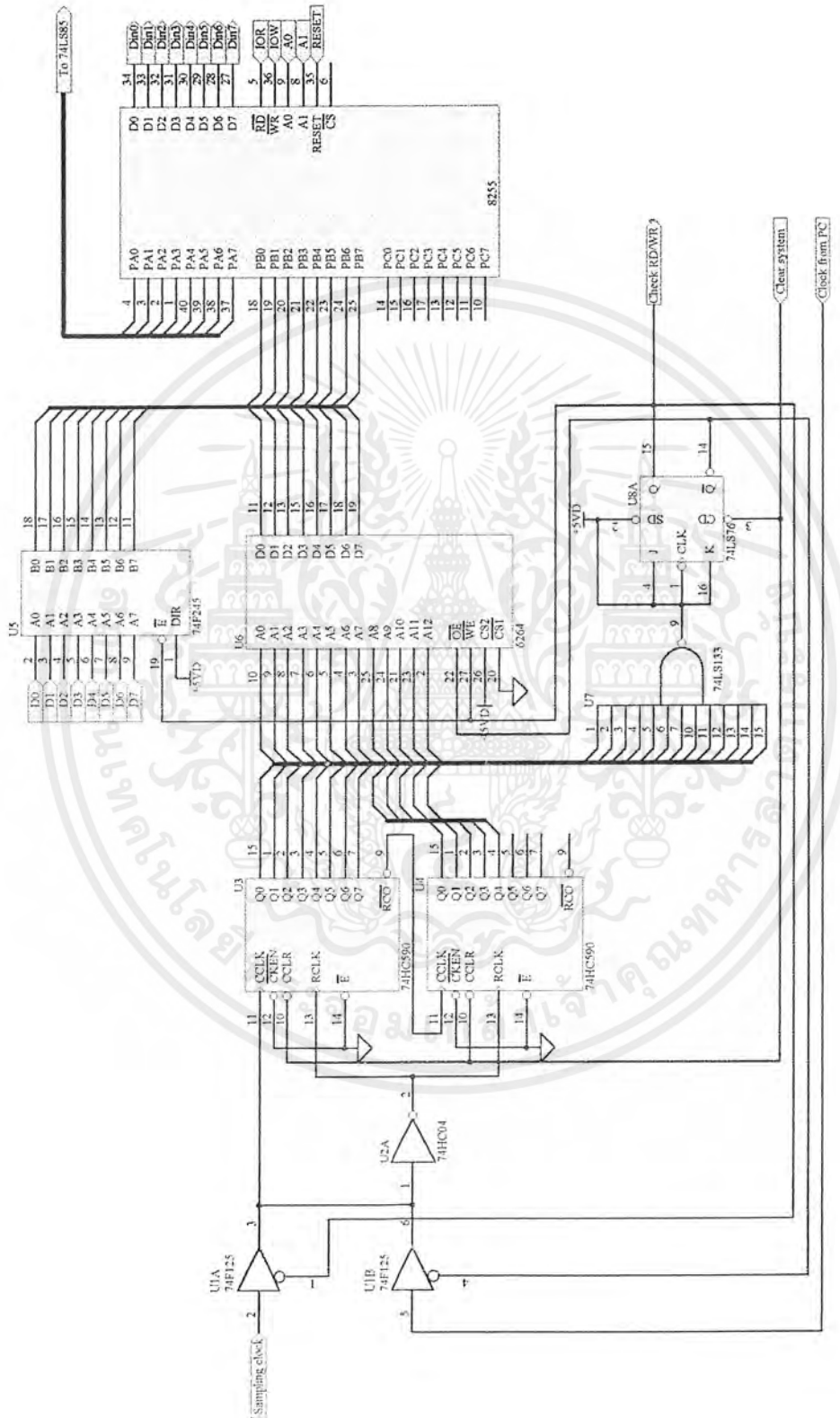
ขั้นตอนที่ 3 : เมื่อโปรแกรมที่เขียนเอาไว้ทำการตรวจสอบจนพบว่า ขณะนี้ SRAM อยู่ในสถานะการอ่านข้อมูลแล้ว ก็จะทำการอ่านข้อมูลข้อมูลจาก SRAM เข้ามาเก็บไว้ใน RAM ภายในคอมพิวเตอร์ โดย

- โปรแกรมจะสร้าง clock ขึ้นมาทีละ 1 ลูกแล้วส่งไปยังวงจรนับ เพื่อใช้เลื่อนตำแหน่งแอดเดรสของ SRAM
- โปรแกรมจะทำการอ่านข้อมูลที่ออกมาจาก SRAM และเอาไปเก็บไว้ใน RAM ภายในฯ ทีละ 1 bytes จากนั้นก็จะวนไปสร้าง clock ใหม่แล้วทำซ้ำไปเรื่อยๆ จนกว่าจะขนถ่ายข้อมูลจาก SRAM ได้ครบ

ขั้นตอนที่ 4 : เมื่อทำการขนถ่ายข้อมูลที่ถูกเก็บไว้ใน SRAM จนหมด และโปรแกรมได้นำข้อมูลไปวาดเป็นรูปสัญญาณออกมาทางหน้าจอเรียบร้อยแล้ว โปรแกรมก็จะสั่งการให้ระบบเริ่มทำการอ่านข้อมูลเข้ามาใหม่อีก โดยกลับไปเริ่มทำขั้นตอนที่ 1 ใหม่อีกครั้ง

จากขั้นตอนต่างๆ ที่กล่าวมาทั้งหมดนี้ สามารถนำมาสร้างเป็นวงจรส่วนควบคุมการอ่าน / เขียนข้อมูลลงใน SRAM ได้ดังรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 วงจรส่วนควบคุมการอ่าน / เขียนของ SRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซี U1 เป็นบัฟเฟอร์ที่นำเอาท์พุทต่อเข้าด้วยกัน แล้วนำไปต่อเข้าขา clock ของวงจรนับเพื่อใช้ในการเลือก clock ที่จะไปนับตัววงจรนับว่าจะใช้ Sampling clock หรือ clock ที่มาจากคอมพิวเตอร์

ไอซี U3 , U4 เป็น counter ขนาด 8 บิต สาเหตุที่ต้องใช้ 2 ตัวต่อคาสเคดกันก็เนื่องจาก SRAM ที่ใช้มีขนาด 8 Kbytes (8,192 bytes) ดังนั้นจำนวนบิตของแอดเดรสบัสเพื่อใช้ในการอ้างตำแหน่งแอดเดรสของ SRAM ให้ครบก็คือ 13 บิต ($2^{13} = 8,192$) แต่ counter แต่ละตัวมีขนาด 8 บิต ดังนั้นจึงต้องใช้ 2 ตัวต่อคาสเคดกัน

ไอซี U2 เป็น Inverter ธรรมดาเพื่อใช้หน่วง clock ให้ช้าไปครึ่งลูก สาเหตุที่ต้องทำเช่นนี้ก็เนื่องจาก counter ที่ใช้นั้น ถึงแม้จะเป็น Synchronous counter แต่โครงสร้างภายในเป็นแบบ Asynchronous counter (ดูรายละเอียดได้จาก datasheet ของ 74HC590 ในภาคผนวก) เพียงแต่มี latch ต่อเอาไว้อยู่ภายในตัวเลย จึงทำให้การนับที่ออกมาเป็นแบบ Synchronous ดังนั้นจึงต้องใช้ clock อีกตัวหนึ่งเพื่อไป Enable ค่าที่อยู่ใน latch ให้ออกมาพร้อมๆ กันทุกบิต

ไอซี U5 เป็นบัฟเฟอร์ข้อมูลที่ใช้กั้นระหว่างขา data ของ SRAM กับ ไอซี A/D

ไอซี U6 เป็น SRAM (Static RAM) ที่ใช้สำหรับเก็บข้อมูลที่มาจากไอซี A/D เอาไว้ก่อน หลังจากนั้นข้อมูลที่ถูกเก็บไว้ก็จะถูกอ่านเข้าไปยังคอมพิวเตอร์ตามขั้นตอนที่ได้อธิบายไปแล้ว

ไอซี U7 , U8 ก็คือ ส่วน RD/WR control ของระบบนั่นเอง ไอซี U8 เป็น JK Flip-Flop ที่ทำงานในโหมด Toggle (เอาท์พุทจะเป็น 0 , 1 สลับกันไปทุกครั้งเมื่อมีพัลส์เข้ามาแต่ละลูกที่ขา clock) เพื่อใช้เป็นตัวเปลี่ยนสถานะการอ่าน / เขียนของ SRAM ไอซี U7 เป็น 13-input NAND gate ซึ่งขาอินพุททั้ง 13 ขานี้จะถูกต่อเข้ากับขาแอดเดรสบัสของ SRAM เพื่อตรวจสอบว่า ขณะนี้ข้อมูลได้ถูกเขียนลงในตำแหน่งแอดเดรสสุดท้าย (แอดเดรสที่ 8,191 หรือ 1111111111111_B) ของ SRAM แล้วหรือยัง โดยถ้าข้อมูลได้ถูกเขียนลงบนแอดเดรสสุดท้ายนี้แล้ว จะทำให้เอาท์พุทของ NAND gate เปลี่ยนจาก 1 เป็น 0 และทำให้ JK Flip-Flop เกิดการ Toggle ขึ้น ผลก็คือ SRAM เปลี่ยนสถานะจากการเขียนเป็นการอ่านข้อมูลทันที

สัญญาณ Check RD/WR ? เป็นสัญญาณที่ถูกส่งเข้าไปยังคอมพิวเตอร์ เพื่อให้โปรแกรมทำการตรวจสอบได้ว่า ขณะนี้ SRAM กำลังอยู่ในสถานะอ่านหรือเขียนข้อมูลอยู่ สามารถรับข้อมูลจาก SRAM เข้ามาเพื่อนำไปวาดรูปสัญญาณได้แล้วหรือยัง

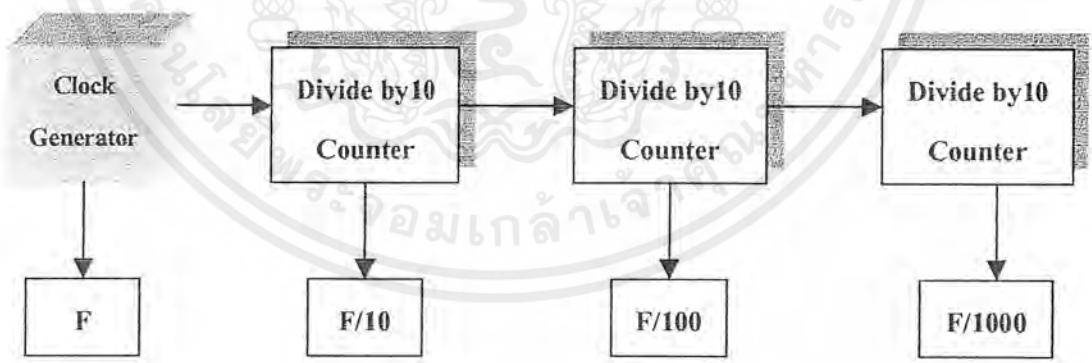
สัญญาณ System Reset เป็นสัญญาณที่ส่งมาจากคอมพิวเตอร์ และจากส่วน Logic control ซึ่งจะกล่าวถึงในภายหลังโดยการเขียนโปรแกรม ใช้เพื่อทำการ Clear counter และ Reset ระบบให้เริ่มทำการเขียนข้อมูลลงใน SRAM ใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5) ส่วนกำเนิดสัญญาณนาฬิกาสำหรับการสุ่ม (Clock generator)

เนื่องจากสัญญาณนาฬิกาที่ใช้ในการสุ่ม (Sampling) ให้แก่ A/D นั้น จะต้องมีความเที่ยงตรงสูง ด้วยเหตุนี้จึงใช้วงจร Crystal oscillator ที่ใช้ Crystal ความถี่ 10 MHz เป็นตัวสร้าง Sampling clock ให้แก่ไอซี A/D แต่เนื่องจากเราต้องการให้โครงงานออสซิลโลสโคปบนเครื่องคอมพิวเตอร์เครื่องนี้สามารถวัดสัญญาณได้ตั้งแต่ความถี่ต่ำๆ ที่มีค่าแค่ไม่กี่ Hz ไปจนถึงสัญญาณที่มีความถี่สูงๆ ประมาณเกือบ 1 MHz ถ้าหากเรากำหนดให้ระบบมี Sampling clock เพียงความถี่เดียวคือ 10 MHz แล้ว จะทำให้มีข้อจำกัดคือไม่สามารถวัดสัญญาณที่มีความถี่ต่ำๆ ได้ เพราะ SRAM ไม่สามารถเก็บรูปสัญญาณได้ครบ 1 คาบเวลานั่นเอง เช่น หากสมมุติว่า Sampling clock มีความถี่ 10 MHz ถูกนำไปใช้สุ่มสัญญาณอนาลอกที่มีความถี่ 1 kHz ผลก็คือ สัญญาณอนาลอกจะถูกสุ่มได้ 10,000 ค่า (10 MHz /1kHz = 10,000) ใน 1 คาบเวลา ดังนั้นหากใช้ SRAM เพื่อทำการเก็บข้อมูลให้ครบ 1 คาบ จะต้องใช้หน่วยความจำมากถึง 10,000 Bytes ซึ่งมีไม่พอเพราะ SRAM ที่ใช้มีความจุ 8 kbytes เท่านั้น

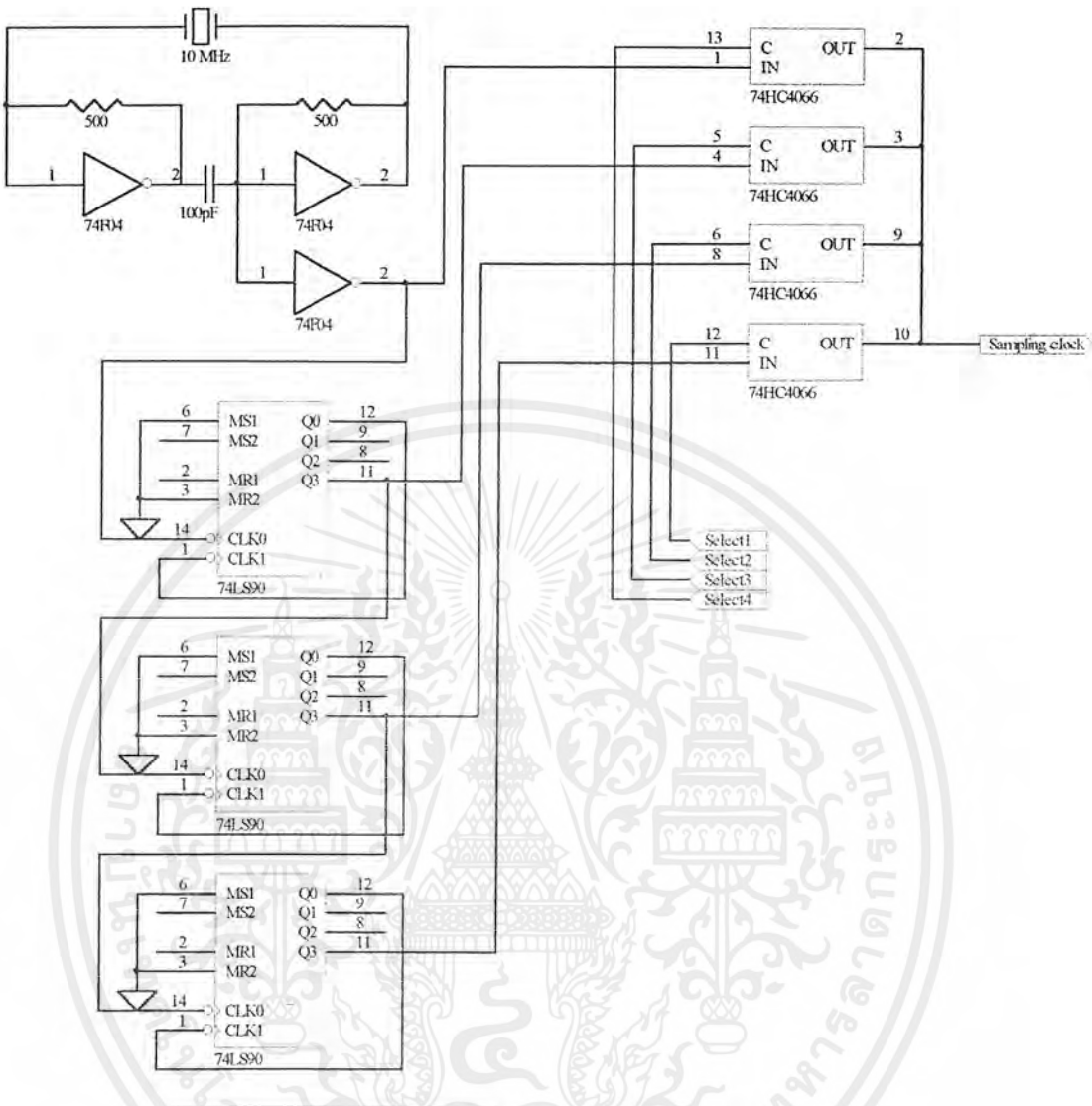
ดังนั้นจึงมีความจำเป็นที่จะต้องมี Sampling clock หลายความถี่ เพื่อใช้ Sampling สัญญาณอินพุทที่มีช่วงความถี่แตกต่างกันออกไป ซึ่งสามารถทำได้โดยการต่อวงจรนับ หรือวงจรหารความถี่ ซึ่งในที่นี้จะใช้วงจรนับ 10 เพื่อเป็นตัวลดความถี่ที่ถูกสร้างจาก Crystal oscillator ลงมาเป็นทอดๆ ทำให้สามารถวัดสัญญาณที่มีความถี่ต่ำๆ ได้ การต่อวงจรหารความถี่เข้ากับ Clock generator มีลักษณะตามบล็อกไดอะแกรมดังรูปที่ 5.6



รูปที่ 5.6 บล็อกไดอะแกรมแสดงการสร้าง Sampling clock 4 ความถี่

จากบล็อกไดอะแกรมในรูปที่ 5.6 Clock Generator ที่ใช้จะเป็นแบบ Crystal oscillator ที่ผลิตความถี่ 10 MHz ดังนั้นความถี่ของ Sampling clock จะมีอยู่ 4 ความถี่ได้แก่ 10MHz , 1MHz , 100kHz และ 10 kHz ดังวงจรในรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

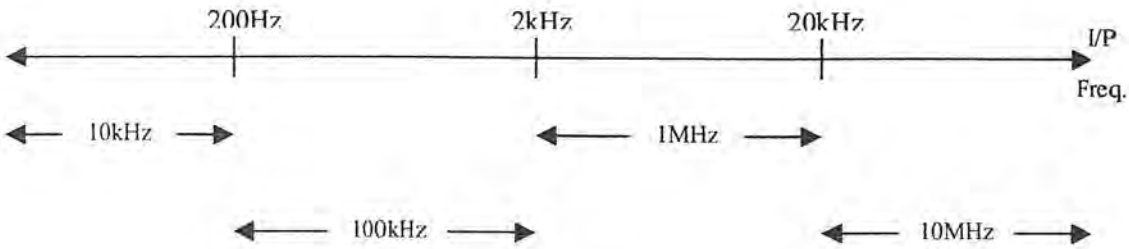


รูปที่ 5.7 วงจร Clock generator 4 ความถี่ และ Clock selector

5.6) ส่วนเลือกความถี่ของ Clock generator เพื่อใช้เป็น Sampling clock

ส่วน Clock selector ซึ่งใช้สำหรับเลือกความถี่จาก Clock generator ที่เหมาะสมให้กับไอซี A/D เพื่อใช้เป็น Sampling clock ของระบบนั้น จะถูกสั่งการจากโปรแกรมที่เขียนเอาไว้ โดยโปรแกรมจะทำการคำนวณหาความถี่ของสัญญาณอินพุทจากข้อมูลที่ถูกเก็บไว้ใน RAM ภายในของคอมพิวเตอร์อยู่ก่อนแล้ว เมื่อได้ค่าความถี่ออกมาแล้วก็จะทำการประมวลผลว่า สัญญาณอินพุทที่ความถี่นี้จะต้องใช้ Sampling clock ความถี่เท่าไร ซึ่งได้กำหนดหลักการเลือก Sampling clock เอาไว้ดังรูปที่ 5.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

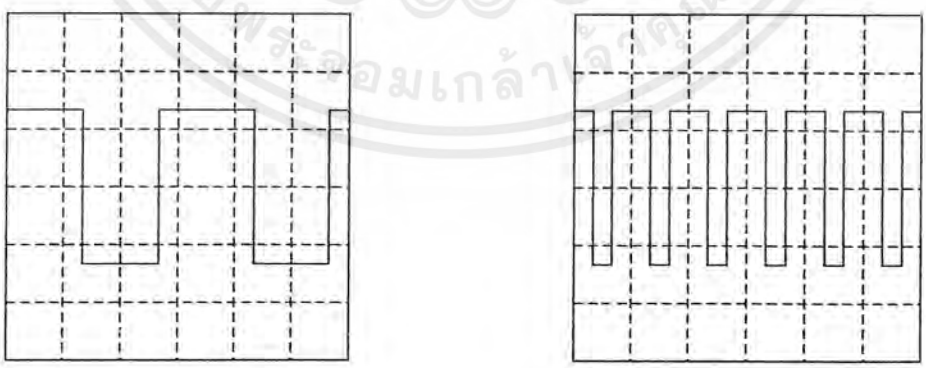


รูปที่ 5.8 หลักการเลือกความถี่ของ Sampling clock ให้กับ A/D

เมื่อประมวลผลได้แล้วก็จะส่งสัญญาณ Select ออกมายังขา Control ของ Analog switch เบอร์ 74HC4066 เพื่อทำการ Enable ให้แก่ Analog switch ตัวใดตัวหนึ่งจาก 4 ตัวเปิดออก ทำให้ clock สามารถผ่านออกมาเป็น Sampling clock ของ A/D ได้ ดังรูปที่ 5.7

5.7) ส่วนควบคุมให้มีการเก็บข้อมูลต่อ เนื่องจาก RAM ไม่พอ

ถึงแม้ว่าเราได้ทำการออกแบบระบบให้สามารถวัดสัญญาณได้หลายช่วงความถี่ โดยการสร้าง Sampling clock ให้มี 4 ความถี่ เพื่อป้องกันปัญหาเนื่องจาก SRAM เก็บสัญญาณที่มีความถี่ต่ำๆ ได้ไม่ครบ 1 คาบ ทำให้นำไปแสดงผลได้ไม่หมด เอาไว้แล้วก็ตาม แต่ยังมีอีกสาเหตุหนึ่งที่ทำให้ SRAM ต้องทำการเก็บข้อมูลเข้ามาใหม่อีกครั้ง และต้องให้ต่อเนื่องจากข้อมูลสุดท้ายก่อนหน้านั้นด้วย คือ การปรับค่า Time / Div ที่ไม่เหมาะสมหรือปรับให้มีค่ามากไปดังตัวอย่างในรูปที่ 5.9

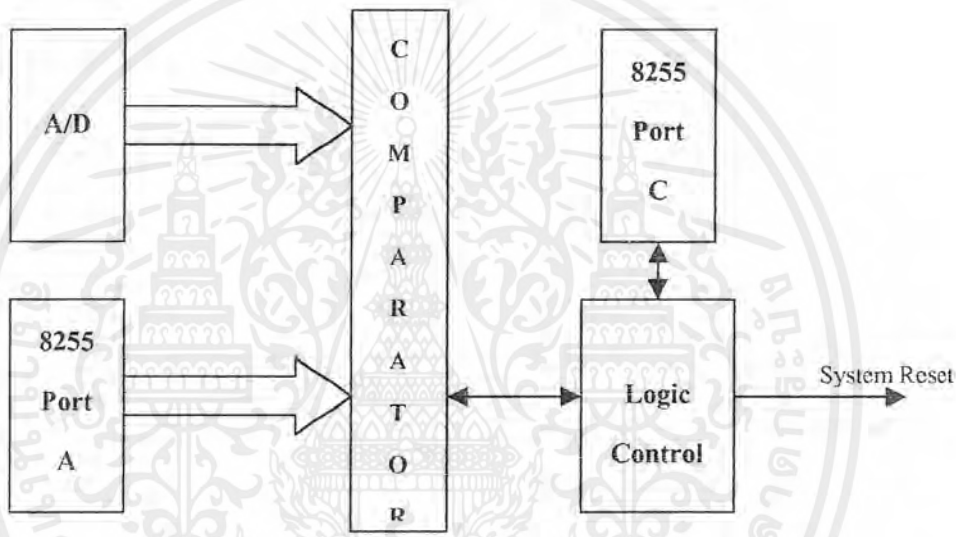


(ก) Time/Div ค่าน้อย (ข) Time/Div ค่ามาก

รูปที่ 5.9 ผลของการปรับ Time/Div ให้มีค่าต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.9 จะเห็นว่าถ้าเราปรับค่า Time/Div ให้มีค่าสูงขึ้น ก็จะทำให้จำนวนคาบของสัญญาณถูกแสดงผลออกมาที่หน้าจอมากขึ้นด้วย นั่นหมายถึงจะต้องดึงข้อมูลที่เก็บไว้ใน RAM ออกมาพล็อตเป็นรูปสัญญาณมากขึ้นด้วย ดังนั้นถ้าเราปรับค่า Time/Div มากเกินไป ก็อาจทำให้ RAM มีข้อมูลไม่เพียงพอที่จะนำมาวาดรูปสัญญาณให้เต็มหน้าจอได้ ดังนั้นจึงต้องแก้ปัญหาโดยสั่งให้ SRAM ภายนอก เริ่มบันทึกรูปคลื่นสัญญาณเข้ามาอีก และต้องให้ต่อเนื่องกับจุดสุดท้ายที่ถูกพล็อตเอาไว้แล้วบนหน้าจอด้วย ส่วนควบคุมการเขียนข้อมูลต่อลงใน SRAM มีรายละเอียดดังบล็อกไดอะแกรมต่อไปนี้

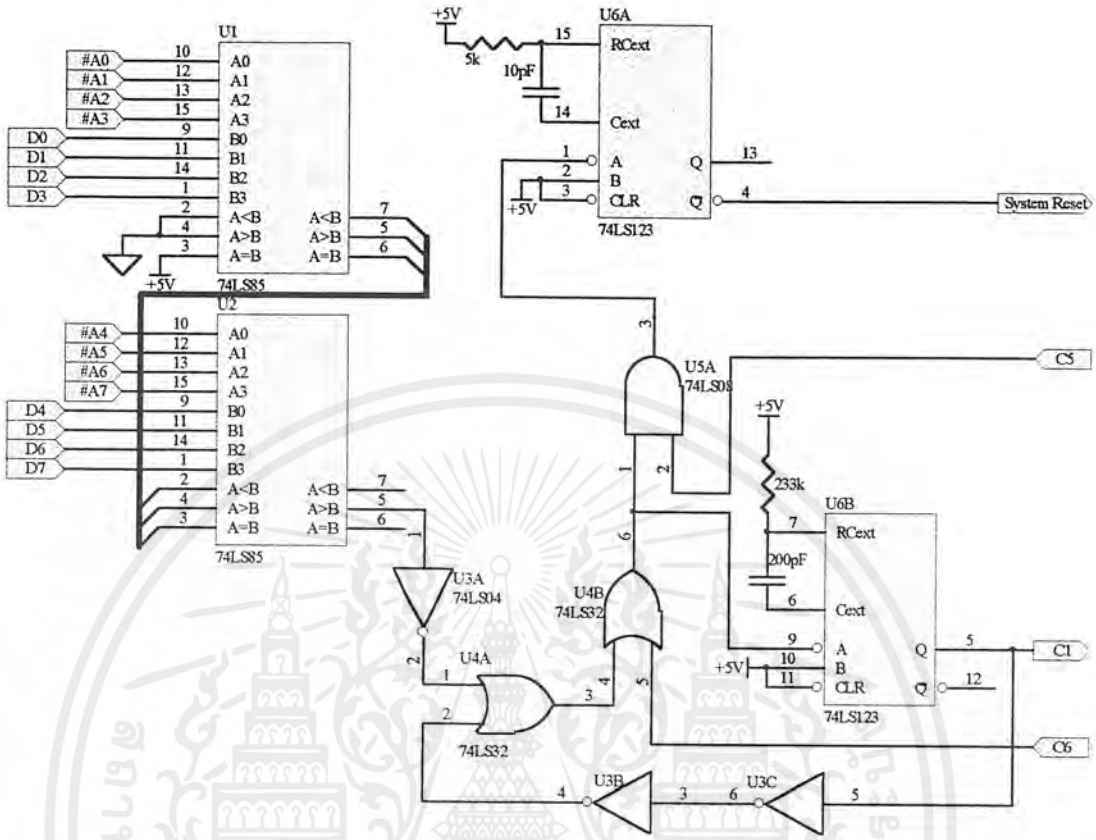


รูปที่ 5.10 บล็อกไดอะแกรมแสดงส่วนควบคุมการเก็บข้อมูลต่อ

จากรูปที่ 5.10 เมื่อมีความจำเป็นต้องสั่งให้ SRAM เก็บข้อมูลต่อจากค่าสุดท้ายที่ได้เก็บไว้ก่อนหน้านั้น โปรแกรมก็จะทำการส่งค่าสุดท้ายที่อยู่ใน RAM ภายในฯ ออกมาทางพอร์ต A ของ 8255 และส่งสัญญาณควบคุมการทำงานไปให้แก่ Logic Control โดยผ่านทางพอร์ต C เมื่อถึงช่วงเวลาที่อินพุททั้งสองของ comparator มีค่าเท่ากันแล้ว ส่วน Logic Control ก็จะสร้างสัญญาณ System Reset ส่งไปให้ส่วน RD/WR Control ในหัวข้อ 5.2.3 เพื่อทำการ Clear counter และ Reset ระบบให้เริ่มทำการเขียนข้อมูลลงใน SRAM ใหม่อีกครั้ง

จากบล็อกไดอะแกรมที่กล่าวมานี้สามารถนำมาออกแบบวงจรได้ดังรูปที่ 5.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 วงจรควบคุมการเขียนข้อมูลต่อ ลงใน SRAM

จากรูปที่ 5.11 สัญญาณ C6 ซึ่งมาจากพอร์ต C เป็นสัญญาณ Enable ให้วงจรส่วนนี้ทำงาน สัญญาณ C6 เป็นสัญญาณที่ใช้ Reset ระบบโดยถูกสั่งมาจากโปรแกรม สัญญาณ C1 จะถูกส่งไปยังพอร์ต C เพื่อให้โปรแกรมสามารถตรวจสอบได้ว่า ขณะนี้ส่วน Logic control ได้ทำการ Reset ระบบเพื่อเก็บข้อมูลลงใน SRAM แล้วหรือยัง สาเหตุที่ต้องใช้ไอซี U6 ซึ่งเป็น Monostable multivibrator ก็คือ ใช้เพิ่มความกว้างของพัลส์หรือขยายช่วงเวลาของการเกิดพัลส์ที่เกิดขึ้นให้ยาวขึ้นอีก เพื่อให้โปรแกรมสามารถตรวจสอบได้นั่นเอง

5.8) พอร์ตข้อมูล และวงจรถอดรหัสแอดเดรส

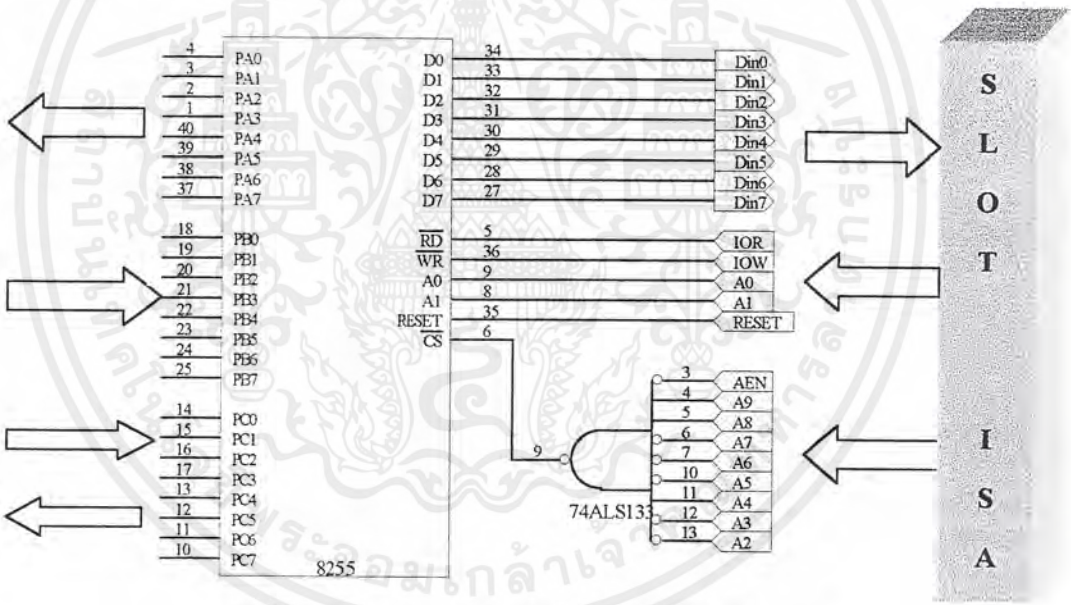
พอร์ตข้อมูลจะถูกใช้เพื่อเป็นช่องทางให้เครื่องคอมพิวเตอร์สามารถติดต่อกับโลกภายนอกได้ ข้อมูลหรือสัญญาณต่างๆ ที่มาจากส่วนของวงจรที่เราออกแบบไว้ จะต้องผ่านพอร์ตข้อมูลเพื่อให้สามารถอินเตอร์เฟสกับคอมพิวเตอร์ได้โดยผ่านทางสล๊อต ISA จากวงจรที่ได้ออกแบบไว้นั้น จะใช้ไอซี 8255 (8255 Programmable Peripheral Interfacing : PPI) เป็นพอร์ตข้อมูลแบบขนานสำหรับการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับบัสในเครื่องคอมพิวเตอร์ โดยในไอซี 8255 1 ตัวจะมีพอร์ตอยู่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตได้แก่ พอร์ต A , พอร์ต B และพอร์ต C โดยได้ทำการกำหนดหน้าที่ และหมายเลขแอดเดรสของแต่ละพอร์ตไว้ดังนี้

- พอร์ต A (หมายเลขแอดเดรสของพอร์ตคือ 310H) : กำหนดให้เป็นพอร์ตเอาท์พุทสำหรับส่งข้อมูลออกมาให้แก่ ส่วนเก็บข้อมูลต่อเนื่อง ลงใน SRAM ดังในรูปที่ 5.10
- พอร์ต B (หมายเลขแอดเดรสของพอร์ตคือ 311H) : กำหนดให้เป็นพอร์ตอินพุทสำหรับรับข้อมูลที่ถูกส่งออกมาจากจาก SRAM ของวงจร
- พอร์ต C (หมายเลขแอดเดรสของพอร์ตคือ 313H) : กำหนดให้เป็นทั้งพอร์ตอินพุท และเอาท์พุท ใช้สำหรับรับส่งสัญญาณต่างๆ ที่ใช้ควบคุมการทำงานของวงจร

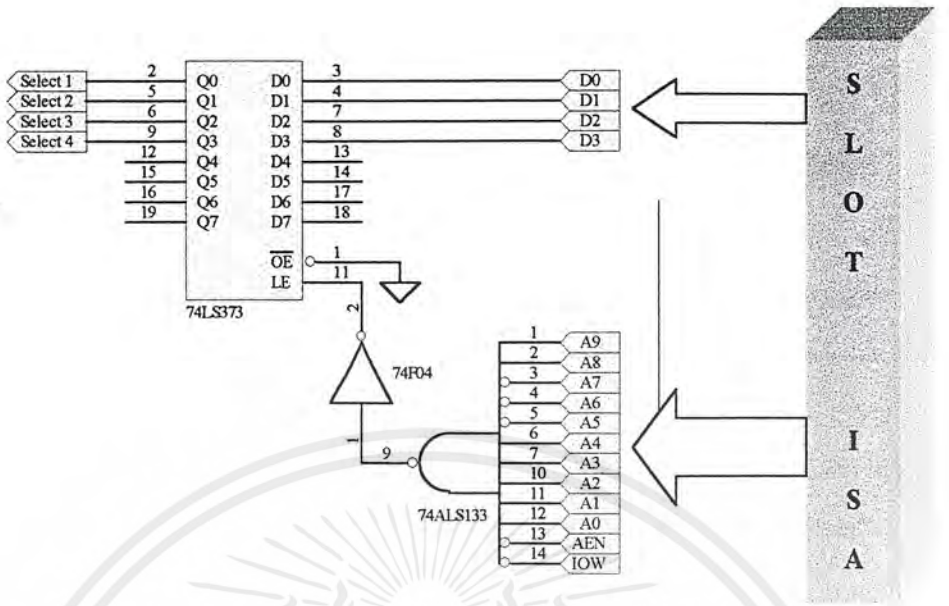
สำหรับวิธีการออกแบบวงจรถอดรหัสแอดเดรสของพอร์ตนั้นใช้หลักการเดียวกับที่กล่าวมาแล้วในบทที่ 4 และจากการกำหนดหน้าที่ของพอร์ตต่างๆ ในไอซี 8255 จึงสามารถออกแบบวงจรได้ดังรูปที่ 5.12



รูปที่ 5.12 การออกแบบพอร์ตข้อมูลโดยใช้ ไอซี 8255

นอกจากพอร์ตข้อมูลโดยใช้ไอซี 8255 แล้ว ยังมีพอร์ตเอาท์พุทอีกชุดหนึ่งจำนวน 4 บิตซึ่งถูกใช้เป็นสัญญาณ select ให้แก่ขาควบคุมของ Analog switch เพื่อใช้เลือก Sampling clock ให้แก่ A/D โดยได้ทำการออกแบบให้หมายเลขแอดเดรสของพอร์ตนี้คือ 31FH ดังรูปที่ 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 พอร์ตเอาท์พุทของสัญญาณ Select ที่ถูกส่งไปยัง Analog switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การเขียนโปรแกรม

6.1 บทนำ

โปรแกรมหรือ Software นั้นมีความสำคัญมาก พอ ๆ กับ Hardware. Hareware ที่ดีเพียงใดก็ตามจะไม่มีความหมายเลยถ้าปราศจาก Software เปรียบเสมือน มนุษย์ที่มีร่างกายแข็งเรง สมบูรณ์พร้อมแต่ไม่มีสมองในการคิด คำนวณ เฉกเช่นเดียวกัน Software ก็จะขาด Hardware ไม่ได้ สมอง จะมีความหมายอะไรหากไม่มีร่างกายที่สมบูรณ์พร้อมในการใช้งาน ดังนั้น Hardware และ Software เป็นสิ่งที่ต้องพึ่งพาอาศัยกัน ทำงานร่วมกันอยู่ตลอดเวลา

6.2 บล็อกไดอะแกรมเบื้องต้นในการออกแบบโปรแกรม

ในการทำงานใดก็ตามต้องมีการวางแผนหรือออกแบบก่อนเสมอ เพื่อก่อให้เกิดประสิทธิผลสูงสุด โดยทั่ว ๆ ไป มีโครงสร้างดังนี้

รูปที่ 6.1 บล็อกไดอะแกรมของการออกแบบโปรแกรมเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทำความเข้าใจในขอบเขตของงาน

การทำความเข้าใจในขอบเขตของงานนั้นถึงได้ว่ามีความสำคัญมากที่สุด ทุกกระบวนการถัดจากนี้จะมาจากขอบเขตหรือเป้าหมายของงานทั้งสิ้น หากความเข้าใจยังคลุมเครืออยู่ ก็เป็นการยากยิ่งในการคิด สร้างสรรค์กระบวนการต่าง ๆ ต่อจากนี้ ขอบเขตของงานนั้นต้องชัดเจนแต่มีความยืดหยุ่นได้ตามสถานการณ์ การตั้งเป้าหมายที่สูงไปนักไม่ใช่สิ่งที่ดีเสมอไป อาจทำความยุ่งยาก ความลำบากอย่างมาก และความย่อท้อของผู้ดำเนินการ ในทำนองเดียวกันการตั้งเป้าหมายที่ต่ำเกินไปก็ไม่ได้ก่อให้เกิดประโยชน์เท่าใดนัก ก็ตรงกับคำกล่าวที่ว่าทุกสิ่งหากมากไปก็ไม่ดี น้อยไปก็ไม่ดี

การทำความเข้าใจในขอบเขตของงาน ในที่นี้คือ การวิเคราะห์เป้าหมายหลักของโครงงานมาเป็น เป้าหมายทาง Software ดังนี้

เป้าหมายหลักของโครงงาน ในที่นี้คือ การทำ Oscilloscope on PC ซึ่งมีผลสืบเนื่องดังนี้

1. มีการรับข้อมูลผ่าน Port ของ Computer (8255)
2. การส่งคำสั่งควบคุมเพื่อการทำงานที่ถูกต้องของ ISA Card
3. การคำนึงถึงข้อจำกัดต่าง ๆ ของ Hardware เช่น ความเร็วรับ / ส่ง ข้อมูล

วิเคราะห์ และตีความมาเป็นเป้าหมายทาง Software ดังนี้

1. เขียนโปรแกรมในการ Outport ออกไปควบคุมการทำงานได้
2. เขียนโปรแกรมเพื่อรับข้อมูลภายนอกได้
3. นำข้อมูลที่ได้มาคำนวณและทำการวาดกราฟ ผ่านทางหน้าจอคอมพิวเตอร์ได้
4. เขียนโปรแกรมที่สามารถปรับแต่ง V/D และ Time/D ได้อย่างอิสระ
5. เขียนโปรแกรมควบคุมการทำงานการ Trigger ทางได้
6. สามารถ เก็บ / เปิด (Save/Open) ข้อมูลได้ตามที่ผู้ใช้ต้องการ
7. เขียน Interface ทางหน้าจอคอมพิวเตอร์ให้ ดูง่าย และสะดวก ที่สุดเท่าที่จะเป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เขียนโปรแกรมให้ตรงกับหลัก **Robust Programming** [1] ให้มากที่สุดเท่าที่เป็นไปได้

6.4 การทำความเข้าใจกับ Hardware

เมื่อเราทราบเป้าหมายทาง Software แล้ว ก็ต้องมาทำความเข้าใจกับ Hardware ต้องทราบถึงโครงสร้าง ลักษณะเด่น ลักษณะด้อย เพราะSoftware จะเรียกใช้ทรัพยากรของ Hardware เปรียบเสมือน สมองต้องทราบว่า แขนขา ของเราสั้นยาวเท่าไหร่ สามารถเอื้อมไปหยิบสิ่งของถึงหรือไม่ มีแรงเท่าไหร่ ถ้าสมองไม่ทราบก็ไม่สามารถคาดคะแน และสั่งการทำงานได้ เนื่องจากโครงสร้างและการทำงานต่าง ๆ ทาง Hardware ได้กล่าวไว้ในบทก่อนหน้าที่แล้ว ดังนั้นจึงไม่ขอนำมากล่าวซ้ำอีก

Robust Programming [1] การเขียนโปแกรมลักษณะหนึ่ง ที่คำนึงและให้ความสำคัญของการป้องกันการTerminate ของโปรแกรม มากที่สุด เหนือกว่าสิ่งอื่นใด มีลักษณะเด่น 4 ประการคือ

1. สันนิษฐานไว้ก่อนว่า ผู้ใช้ไม่มีความรู้เลยในการใช้ (Stupid), ผู้ใช้ไม่สามารถใช้งานได้เลยหากไม่ได้รับการบอกกล่าวแบบ Step by Step, ผู้ใช้ไม่มีทักษะ ไม่มีความคิด ไม่มีไหวพริบในการใช้โปรแกรมเลย
2. สันนิษฐานไว้ก่อนว่า ผู้ใช้เป็นผู้มีเจตนาไม่บริสุทธิ์ ผู้ใช้พยายามที่จะทำให้โปรแกรมมีการ Terminate
3. โปรแกรมต้องมีอิสระในการ Terminate ได้โดยไม่ขึ้นแก่กัน เพื่อไม่ให้เกิดลักษณะของ Domino Termination.
4. หากเกิดการ Terminate ต้องมีระบบในการ Feedback หรือข้อมูลกลับไป ให้กลับผู้เขียนโปรแกรมได้ โดยข้อมูลดังกล่าวต้องออกแบบไม่ให้ผู้อื่นล่วงรู้ด้วย ผู้มีอำนาจ หรือผู้เขียนโปรแกรม หรือ ผู้มีสิทธิ์เท่านั้นที่จะได้ข้อมูลนี้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 รู้และเข้าใจในจุดเด่นจุดด้อยในภาษาที่เลือกใช้

เมื่อรู้และเข้าใจ ทั้งเป้าหมาย, โครงสร้างและการทำงานของ Hardware แล้ว ก็มาถึงการตัดสินใจในการเลือกใช้ ภาษาที่จะเขียน ซึ่งมีอยู่ 4 ภาษา ที่อยู่ในขอบเขตการพิจารณา ดังนี้

1. C
2. C++
3. Visual Basic
4. Delphi

ภาษา C / C++อันที่จริงเป็น 2 ภาษาด้วยกัน แต่ในทั่วไปจัดอยู่ในประเภทเดียวกันทั้งสองภ เป็นภาษาที่รู้จักกันดีในหมู่นักเรียน นักศึกษา เป็นภาษาที่มีการทำงาน การควบคุม ระดับล่างที่สุด ในบรรดา High Language ด้วยกัน จะขออธิบายแยกดังนี้

ภาษา C เป็นภาษาที่ได้ถือกำเนิดมาแล้วประมาณ 30 ปี ผู้ที่ทำการเขียน C Compiler คือ ทอมสัน จุดประสงค์แรกเริ่มเดิมทีทอมสันไม่ได้ตั้งใจที่จะเขียน C ขึ้นมาสักทีเดียว จุดประสงค์หลักในขณะนั้นคือ การเขียนระบบปฏิบัติการยูนิกซ์ประวัติความเป็นมาว่ากว่าจะได้ C มีดังนี้

ยูนิกซ์แรกเริ่มถือกำเนิดมาในปี 2512 โดยสืบเนื่องจากการที่สถาบัน MIT , ศูนย์วิจัยเบลล์ ซึ่งเป็นสาขาของ AT&T และบริษัท GE ได้ร่วมมือกันออกแบบและพัฒนา O.S. บนเครื่อง GE645 ซึ่งเป็นเครื่องระดับเมนเฟรม O.S. ตัวนี้มีชื่อว่า Mulitics ซึ่งนับเป็นตัวต้นแบบ O.S. ในปัจจุบันอีกหลาย ๆ ตัว แต่ Mulitics ที่สร้างขึ้นมามีจุดอ่อนอยู่มาก ต่อมาจึงมาการพัฒนาขึ้น เคน ทอมป์สัน (Ken Thompson) ได้พัฒนาโดยใช้ภาษาแอสเซมบลีบนเครื่อง PDP-7 ต่อมาก็ได้มีการจัดสร้างระบบไฟล์ และโปรแกรมสำเร็จรูปต่าง ๆ ผลลัพธ์ก็คือเขาได้ระบบ O.S. ตัวใหม่ได้ให้ชื่อว่า ยูนิกซ์ (UNIX) ทอมป์สันพบว่า O.S. ที่เขาสร้างขึ้นมานั้นยังมีความสามารถจำกัดอยู่ เขาเลยเริ่มนำภาษาอื่น ๆ มาเขียนลองใช้ดู เขาได้นำภาษาฟอร์แทรนมาใช้ก่อน แต่แล้วก็ต้องเลิกใช้ในเวลาที่รวดเร็ว ทั้งนี้ก็เพราะคุณสมบัติของภาษาที่ไม่เอื้ออำนวย ต่อมาได้ลองกับภาษา BCPL (Basic Combined Programming Language) ซึ่งเป็นภาษาที่ย่อมาจากภาษา Algol 60 อีกที ปรากฏว่า BCPL ทำงานได้ค่อนข้างดี ทอมป์สันจึงได้ทำการปรับปรุงภาษานี้อย่างจริงจัง โดยเลือกเอาเฉพาะโครงสร้าง และคุณสมบัติที่จำเป็นจริง ๆ กับการใช้งานมาสร้างเป็นภาษาใหม่และตั้งชื่อ C หลังจากนั้นยูนิกซ์ก็ได้รับปรับปรุงเรื่อยมาตราบจนถึงทุกวันนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา C++ เป็นภาษาที่ได้รับการพัฒนาจากภาษา C อีกทีหนึ่ง อาจกล่าวได้ว่า C เป็นสับเซตของ C++ ทุกคำสั่งใน C สามารถใช้กับ C++ ได้ แต่ บางคำสั่งใน C++ ไม่สามารถใช้กับภาษา C ได้ C++ ได้รับการเพิ่มความสามารถทางด้าน OOP เข้าไปซึ่งเป็นคุณสมบัติที่สำคัญมากในการเขียนโปรแกรม มีการมองเป็นวัตถุเป้าหมายแต่ละตัว (Object) ซึ่งสามารถสืบทอด, อนุพันธ์ , นำกลับมาใช้ใหม่ได้ C++ เป็นต้นแบบในการพัฒนาภาษาสำคัญต่างๆ ต่อไปอีกจำนวนมาก ภาษาที่สำคัญมากในปัจจุบันดัง ภาษา JAVA ก็เป็นหนึ่งในนั้น

Visual Basic (VB)

ในปัจจุบันระบบปฏิบัติการ Windows ได้เข้ามาแทนที่ระบบปฏิบัติการในลักษณะเดิม ซึ่งส่วนใหญ่ที่นิยมคือ MS-DOS ซึ่งเป็นลักษณะของ Command Line แต่ใน Windows จะมีรูปแบบของจอภาพที่ใช้ติดต่อระหว่างคอมพิวเตอร์และผู้ใช้เป็นแบบรูปภาพต่าง เรียกว่า GUI (Graphic User Interface)

ภาษา VB เป็นภาษาสูงอีกตัวหนึ่ง จัดอยู่ในภาษาระดับที่ 4 เป็นโปรแกรมหนึ่งที่ทาง Microsoft ทำขึ้นโดยพัฒนามาจาก ภาษา Basic เพื่อให้ผู้ใช้สามารถใช้งานต่าง ๆ ได้โดย ง่าย สะดวก และรวดเร็ว มี Interface ที่สวยงาม มีลักษณะการทำงานแบบ Visualize เป็นภาษาที่ทำงานบนระบบปฏิบัติการ Window ของ บริษัท Microsoft เท่านั้นในช่วงก่อน ในทว่าในปัจจุบันได้มีผู้นำมาพัฒนาเพื่อให้สามารถใช้งานได้บนระบบปฏิบัติการ Linux เนื่องจากคุณสมบัติข้อดีดังที่ได้กล่าวไปแล้ว ทำให้มีผู้นิยมใช้กันอย่างแพร่หลายในขณะนี้ ที่ได้กล่าวไว้ว่า VB พัฒนามาจะภาษา Basic นั้น จะขอกล่าวความเป็นมาของภาษา Basic อย่างคร่าว ๆ ดังนี้

ภาษา Basic ได้รับการสร้างขึ้นในปี 1963 โดย Hohn Keneny และ Thomas Kurtz ที่วิทยาลัย Dartmouth พวกเขามีจุดมุ่งหมายในการพัฒนา เพื่อใช้ในการสอนโปรแกรม (Programming Concept) โดยเน้นให้รูปแบบของภาษานั้นง่ายต่อการเข้าใจและการใช้งาน รวมทั้งเป็นการทำงานแบบ Interpreter (ตัวแปลภาษาทีละบรรทัด) ซึ่งต่างกับในสมัยนั้นที่ส่วนใหญ่จะอาศัย JCL (Job Control Language) ในการทำการ Compile และ Link ทำให้ในช่วงนั้นได้รับความนิยมอย่างกว้างขวาง โดยเฉพาะคอมพิวเตอร์ส่วนบุคคล PC (Personal Computer) ในปี 1970 Microsoft ได้เริ่มผลิตตัวแปลภาษา Basic ใน ROM ซึ่งเรียกว่า ROM-Based BASIC ขึ้น ต่อมาได้พัฒนาเป็น GW-BASIC ซึ่งเป็นตัวแปลภาษาที่ใช้กับ MS-DOS และในปี 1982 ก็ได้รับการพัฒนาเป็น Microsoft QuickBasic โดยการเพิ่มความสามารถในการ Compile ให้เป็น Executed Program รวมทั้งมีความสามารถเป็น Structed Programming มากขึ้น โดยการตัด Line Number ทิ้งไปเพื่อลบข้อกล่าวหาที่ว่าเป็นภาษาที่มีลักษณะ Spaghetti Code (Logical Flow ของภาษที่ขาดโครงสร้าง) มาใช้รูปแบบของ Sub Program และ User Defined รวมทั้งการใช้ Structured Data Type ต่อมาทาง Microsoft ก็พัฒนาจนกลายเป็น Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Delphi

มีลักษณะคล้าย ๆ VB ได้รับการพัฒนามาจาก ภาษา Pascal โดยบริษัท Borland การทำงาน, รูปแบบ เกือบจะเหมือนกับ VB แตกต่างกันอยู่ 2 ด้านที่เห็นอย่างเด่นชัดคือ

1. ภาษาที่ใช้โดย VB ใช้ภาษา Basic ส่วน Delphi ใช้ภาษา Pascal
2. ลักษณะ Object ของ โดย VB เป็นลักษณะของ Object เทียม กล่าวคือ มีคุณสมบัติการเป็น OOP (Object Oriented Progrmming) ไม่สมบูรณ์ ส่วน Delphi เป็น Object แท้ เป็น OOP ที่สมบูรณ์ จุดนี้เป็นข้อที่ Delphi ได้เปรียบ VB เป็นอย่างมาก

จาก 4 ภาษาที่กล่าวมาสามารถจัดได้เป็น 2 กลุ่ม ตามลักษณะ ลักษณะเด่น และการใช้งาน ได้ดังนี้ คือ

กลุ่มที่ 1 กลุ่มที่มีพื้นฐานการทำงานบน Windows อันได้แก่ Visual Basic, Delphi
กลุ่มที่ 2 กลุ่มที่มีพื้นฐานการทำงานบน Dos อันได้แก่ C / C++
ซึ่งเราจะพิจารณาถึง ข้อดี , ข้อเสีย ดังต่อไปนี้

6.5.1 Visual Basic, Delphi

ข้อดี

1. การเขียนโปรแกรมนั้นสามารถทำได้ง่าย สะดวก และรวดเร็ว
2. ในโปรแกรมมี Tools และ Components ต่าง ๆ ให้ไว้สำเร็จรูปแล้ว ไม่ต้องยุ่งยาก และลำบากในการเขียนโปรแกรมเอง
3. มีความเข้ากันได้กับ Active X / OLE ต่าง ๆ บน Windows ทำให้เป็นการง่ายยิ่งขึ้นในการเรียกใช้ความสามารถอื่น ๆ ของโปรแกรมอื่นบน Windows
4. Application ที่ได้รับการพัฒนาออกมา มักอยู่มี Interface ที่สวยงาม และผู้ใช้โปรแกรม (User) ก็ใช้งานได้ง่าย

ข้อเสีย

1. Programmer ไม่ได้เขียนโปรแกรมติดต่อกับ Haredware โดยตรง แต่จะติดต่อผ่านทาง ฟังก์ชัน หรือ Components สำเร็จรูปที่มีให้อยู่แล้วในโปรแกรม
2. Programmer ขาดทักษะ ความรู้ ความเข้าใจ ในการทำงานทาง Hardware (เนื่องจากผลของข้อที่ 1)
3. ขณะที่โปรแกรมกำลังทำงานอยู่ (Run) ไม่ทราบได้เลยว่า โปรแกรมมีการสั่งงานอย่างไร อะไรบ้าง เนื่องจากมีระดับชั้นของ Components สำเร็จรูป กลั้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ลักษณะคล้าย ๆ การทำงานในแต่ละ Layer ใน OSI ที่แต่ละชั้น ไม่จำเป็นต้องรู้ถึงการทำงานในชั้นอื่น ๆ)

6.5.2 C / C++

ข้อดี

1. Progrmmer สามารถเขียนโปรแกรมควบคุมการทำงานของ Hardware ได้ตรง
2. สามารถประยุกต์การทำงานได้หลายลักษณะ
3. การทำงานมีความรวดเร็วมากกว่า เนื่องจากใช้ทรัพยากรของระบบน้อยกว่า
4. ทราบการทำงานแต่ละขั้นอย่างละเอียด ทำให้ง่ายต่อการแก้ไขปัญหา (Debug)

ข้อเสีย

1. Programmger ต้องความรู้ และทักษะทางด้านProgramming สูงกว่า
2. การเขียนโปรแกรมทำได้ยาก, ไม่สะดวก และช้ากว่า
3. Application ที่ได้รับการพัฒนาออกมามักอยู่ในรูปที่ไม่ค่อยสวยงามนัก
4. ไม่เหมาะกับงานประยุกต์บางอย่าง ในลักษณะการใช้ทรัพยากรหรือความสามารถร่วมกัน กับ Application อื่น ๆ

การตัดสินใจและเหตุผลในการเลือกใช้

1. C / C++ เราจะเลือกใช้ภาษานี้ เพราะเป็นภาษาสูงที่ทำงานในระดับ Hardware ได้ที่สุด หากเราทำการศึกษาอย่างจริงจัง เราก็จะรู้ลักษณะการทำงานของ Hardware ได้ระเอียดมากทีเดียว (ยังสู้ Assemble ไม่ได้ เพราะเป็นภาษาระดับต่ำ ใกล้ Hardware มากที่สุด) อีกทั้งยังเป็นการฝึกการเขียนโปรแกรมแบบ OOP ด้วย ซึ่งเราเล็งเห็นว่ามีความสำคัญและเป็นประโยชน์ต่อมาในอนาคต เป็นการสร้างแนวความคิดเบื้องต้นก่อน

2. Visual Basic (VB) เป็นภาษาที่เราเลือกใช้เช่นกัน เนื่องจาก C/ C++ ที่เราได้เลือกใช้ไปก่อนหน้านี้ มีลักษณะการทำงานใน DOS จึงต้องการทดสอบการเขียนโปรแกรมบน Windows บ้าง อีกทั้ง VB กำลังทวีบทบาทมากขึ้นในปัจจุบันเป็นอย่างมาก มี Interface ที่สวยงาม หากเราได้ศึกษาไว้ก็เป็นการดีอย่างยิ่ง

3. Delphi เราไม่เลือกใช้ในขณะนี้ เพราะ Delphi มีลักษณะที่คล้าย กับ VB มาก เมื่อเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกใช้ VB แล้วก็ไม่มีความจำเป็นเท่าไหร่นักที่ต้องศึกษา Delphi อีก

6.6 การวิเคราะห์ ตีความเป็น Algorithms

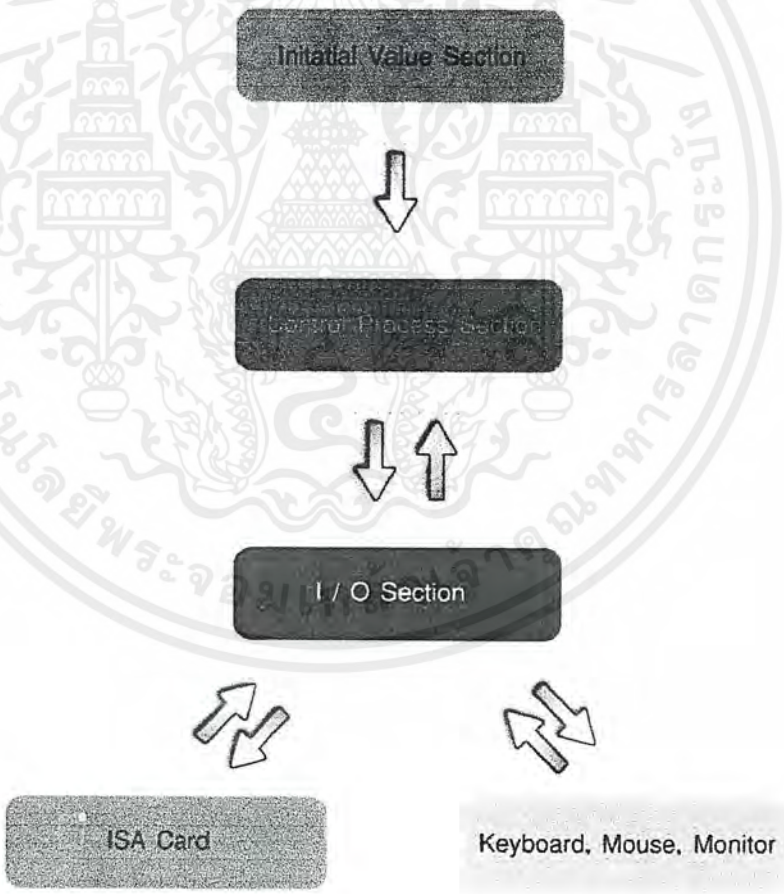
เนื่องจากเนื่องหาในส่วนนี้มีรายละเอียดค่อนข้างมาก จึงขอยกขึ้นเป็นบท ซึ่งอยู่ในบทถัดไป คือ บทที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การวิเคราะห์และตีความหมายเชิง Algorithms

นำความรู้ ความเข้าใจ และข้อมูลต่าง ๆ จากกระบวนการก่อนหน้านี้ มาวิเคราะห์เพื่อคิดออกเป็น Algorithms ทาง Software ซึ่งเป็นกระบวนสุดท้ายในการออกแบบ Software ได้ดังนี้

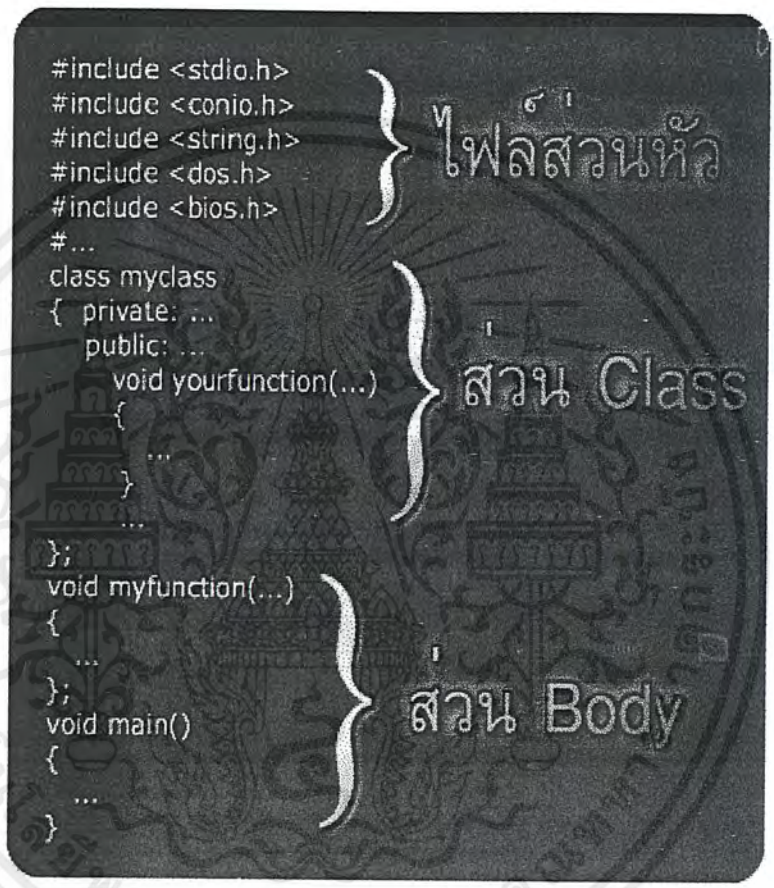
ก่อนอื่นขอแบ่งเป็น Block Diagram หลัก ๆ ก่อน และค่อยลงรายละเอียดในแต่ละส่วนลงไปในภายหลัง เพื่อให้ง่ายในการเห็นถึงภาพรวมก่อน ดังนี้



รูปที่ 7.1 บล็อกไดอะแกรมโครงสร้างหลัก ๆ ในการวิเคราะห์ Algorithms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา C / C++ โดยทั่วไปมีโครงสร้างดังนี้



รูปที่ 7.2 โครงสร้างพื้นฐานของภาษา C / C++

7.1 Initial Value Section

เป็นส่วนของการกำหนดค่าเริ่มต้น ข้อตกลง ในการเขียนโปรแกรม ซึ่งจะกระทำเพียงครั้งเดียวตอนเริ่มต้นการทำงานของโปรแกรมเท่านั้น ในที่นี้ก็คือ ฟังก์ชันส่วนหัว, การประกาศตัวแปร และการกำหนดค่าคงที่ของตัวแปร นั่นเอง เราต้องทราบว่า ฟังก์ชันที่เราต้องการใช้อยู่ใน ไฟล์ส่วนหัวตัวไหน แล้วทำการผนวกเข้าไปเป็นส่วนหนึ่งในการ Compile ของโปรแกรม หากไม่กำหนดไว้ก่อน ก็จะไม่สามารถทำการ Compile ได้ ตัว Compiler จะฟ้องข้อผิดพลาดออกมา เช่น การต้องการใช้ฟังก์ชัน clrscr() ในการลบหน้าจอใน Text Mode เราก็ต้องประกาศฟังก์ชันส่วนหัวที่ชื่อว่า conio.h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้น เหตุที่ว่าทำไมถึงไม่เป็นอัตโนมัติล่ะ ทำไม Compiler ถึงไม่หาเอาเองใน Include File ที่มีอยู่ เหตุผลที่ว่ามีดังนี้

1. เสียเวลาและสิ้นเปลืองทรัพยากรโดยใช่เหตุ ในการค้นหาฟังก์ชันต่าง ๆ ในไฟล์ส่วนหัวเหล่านี้
2. ในการพัฒนาฟังก์ชันหนึ่ง ๆ อาจสามารถพบได้ในไฟล์ส่วนหัวหลายไฟล์ โดยในการทำงานของฟังก์ชันในแต่ละไฟล์ส่วนหัว ก็อาจจะมีไม่เหมือนกัน กล่าวคือ จะเป็นง่ายและสะดวกในการกำหนดไฟล์ส่วนหัวเอง กรณีดังกล่าวนี้พบมากในลักษณะของพัฒนาไฟล์ส่วนหัวตัวเดิม มาเป็นตัวใหม่ที่มีลักษณะอื่น ๆ มากกว่า
3. ง่ายและสะดวก ในการ Compile แต่ละส่วน

ส่วนการประกาศตัวแปรต่าง ๆ นั้น ได้มาจากความต้องการตัวแปรนั้น ๆ มาใช้สอยในโปรแกรมของเรา เราต้องประกาศตัวแปรให้ถูกต้องตามลักษณะ ชนิดของตัวแปร และความเข้ากันได้ของตัวแปร ชนิดตัวแปรพื้นฐาน มีดังนี้ (ในที่นี้คือตัวแปรพื้นฐานใน C++)

ชนิดตัวแปร	คิดเครื่องหมาย	ช่วงจำนวน		จำนวนตัวเลขความแม่นยำ	จำนวนไบต์
		ต่ำ	สูง		
Char	✓	-128	127	N/A	1
Unsigned char	✗	0	255	N/A	1
Int	✓	-32,768	32,767	N/A	2
Unsigned int	✗	0	65,535	N/A	2
Long	✓	-2,147,483,648	2,147,483,647	N/A	4
Unsiged long	✗	0	4,294,967,295	N/A	4
Float	✓	3.4×10^{-38}	$3.4 \times ^{38}$	7	4
Double	✓	1.7×10^{-308}	1.7×10^{308}	15	8
long double	✓	3.4×10^{-4932}	1.1×10^{4932}	19	10

รูปที่ 7.1 ตารางแสดงคุณสมบัติของตัวแปรแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 Control Process Section

ทำหน้าที่เป็นส่วนกลางในการคิด คำนวณ ควบคุม การทำงานทั้งหมดของโปรแกรม และ Hardware ภายนอก อันได้แก่

1. ฟังก์ชันต่าง ๆ ในโปรแกรม
2. เงื่อนไขการควบคุม (Control Statements) อันประกอบด้วย การดำเนินการทางลูป (Loop) และการตัดสินใจ (Decisions)
3. การร้องขอ เรียกใช้ ทรัพยากรที่มีอยู่ในระบบ
4. การจัดการข้อผิดพลาดต่าง ๆ ที่จะบังเกิดขึ้นในระบบ

ซึ่งในการเขียนโปรแกรมนั้นต้องคำนึงถึง Syntax (ไวยากรณ์ของคำสั่ง หรือข้อกำหนด ต่าง ๆ) ด้วย

7.2.1 ฟังก์ชันต่าง ๆ

ฟังก์ชันที่ใช้ในภาษา C++ นั้น มีอยู่ด้วยกันมากมาย ดังจะพอยกตัวอย่างได้ดังนี้ (อนึ่ง สามารถหาข้อมูลเพิ่มเติมได้ที่ Help ของโปรแกรม ซึ่งมีให้อย่างครบถ้วน ทุกฟังก์ชันในทุกไฟล์ส่วนหัว โดยกล่าวรายละเอียดอย่างคร่าว ๆ พร้อมตัวอย่างให้ดูพอสังเขป)

ฟังก์ชัน / คำสั่ง ที่ควรทราบในเบื้องต้น

1. printf เป็นคำสั่งในภาษา C (C ++ ก็ใช้ได้แต่ไม่นิยมใช้) การทำงานของมันก็คือ ส่งข้อความที่เราต้องการออกสู่หน้าจอ
2. scanf เป็นคำสั่งใช้ในภาษา C เช่นเดียวกัน (C++ ไม่นิยมใช้) หน้าที่คือ รับ input ทีเข้ามาทาง Keyboard
3. cout เป็นคำสั่งในภาษา C++ ซึ่งใน C ไม่มีคำสั่งนี้ มีลักษณะการทำงานคล้าย printf มีความยืดหยุ่น และความสามารถมากกว่า สามารถทำการเปลียนทิศทางได้ โดยค่า default ของมันคือการส่งออกหน่วยแสดงผลมาตรฐาน หรือ จอภาพนั่นเอง
4. cin เป็นคำสั่งในภาษา C++ (ใน C ไม่มี) หน้าที่คล้าย scanf แต่สามารถเปลี่ยนทิศทางได้เช่นเดียวกับ cout
5. getche มีทั้งใน C และ C++ ใช้สำหรับรับค่า อักขระ 1 ตัวอักษรจาก Keyboard โดยจะไม่ แสดงค่าที่ให้ลงไปในหน่วนแสดงผล
6. gotoxy(x,y) เป็นคำสั่งให้ตำแหน่ง curcur ไปที่ใดที่หนึ่งในหน้าจอที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. initgraph(&driver,&mode,\\tc\\bgi) เป็นคำสั่งในการเริ่มต้นการทำงานในสภาวะ กราฟฟิก โดยการตั้งค่าผ่านทางพารามิเตอร์ driver , mode ส่วน \\tc\\bgi เป็นที่อยู่ของไฟล์ที่เก็บทรัพยากรของกราฟฟิกไว้
8. closegraph() ทำหน้าที่ออกจากสภาวะ กราฟฟิก ไปสู่สภาวะ ข้อความ
9. setcolor(color) เป็นตัวกำหนดสีที่ใช้ในการวาดเส้น
10. setlinestyle(style) ใช้สำหรับเส้นตรง ทำหน้าที่กำหนดรูปแบบการวาดเส้น เช่น วาดเป็นเส้นตรงธรรมดา วาดเป็นเส้นประ วาดเป็นเส้นจุดไข่ปลา เป็นต้น
11. floodfill(x,y,linecolor) ทำหน้าที่ลงสีภายในวงกลมให้เต็ม คำสั่งนี้จะเริ่มต้นที่ตำแหน่ง x,y และจะลงสีตามรูปแบบที่กำหนดใน setfillstyle ขยายจนไปกระทั่งถึงขอบ
12. setfillstyle(style,fillcolor) เป็นการกำหนดรูปแบบการลงสี
13. inportb(IO_Address) ทำหน้าที่รับข้อมูลจาก Port ที่ระบุ
14. int86(0xXX,&iregs,&oregs) เป็นคำสั่งที่ใช้ในการ Interrupt Dos โดย XX คือหมายเลขการ Interrupt ค่าที่ระบุไว้ใน register AX เป็นตัวบอกถึงบริการที่เท่าไหร่ของการ Interrupt นั้นๆ ส่วน iregs,oregs เป็นค่าที่ส่งกลับมา โดยจะมีได้ในหลาย register เช่น CX,DX เป็นต้น
15. กลุ่มคำสั่งประเภทการตัดสินใจ เช่น if _*then*_ else, Switch_case, While do, Do while, for loop, goto เป็นต้น
16. line(x1,y1,x2,y2) คำสั่งที่ใช้ในการลากเส้นตามพิกัดที่ระบุ
17. circle(x,y,r) ทำหน้าที่วาดวงกลมโดยมีจุดศูนย์กลางอยู่ x,y มีรัศมี r
18. rectangle(x1,y1,x2,y2) วาดสี่เหลี่ยมตามพิกัดที่ระบุ

7.2.2 เงื่อนไขการควบคุม (Control Statements)

โปรแกรมที่เราพอโดยทั่วไปนั้น มีอยู่จำนวนไม่มากนักที่กระทำการทางคำสั่ง หรือฟังก์ชันทั้งหมดในลักษณะเรียงลำดับตั้งแต่ต้นจนจบ โปรแกรมส่วนใหญ่มักจะต้องตัดสินใจว่าจะทำอะไรอย่างไร ในลักษณะที่ตอบสนองต่อสภาวการณ์ที่กำลังเปลี่ยนแปลง การ Flow ของตัวควบคุมจะกระโดดจากส่วนหนึ่งของโปรแกรมไปยังอีกส่วนหนึ่ง ซึ่งก็จะขึ้นกับกระบวนการคำนวณและสถานการณ์นั้น ๆ

7.2.2.1 การดำเนินการทางลูป (Loop)

ลูปจะทำให้เกิดการกระทำส่วนหนึ่ง ๆ ของโปรแกรมซ้ำไปเรื่อย ๆ ตราบใดที่เป็นสถาวะเป็นจริง (True) การสิ้นสุดของลูปจะเกิดขึ้นก็ต่อเมื่อ สถาวะเป็นเท็จ (False)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน C++ มีอยู่ สามชนิดด้วยกันคือ ดังนี้

1) ลูป for เป็นลูปที่ จะกระทำงานซ้ำ ตามจำนวนครั้งที่ระบุไว้ เช่น

```
for ( i = 0; i < 10 ; i++ )
    cout<< i <<",";
```

Output: 1,2,3,4,5,6,7,8,9,10

2) ลูป while เป็นลูปที่จะวนซ้ำจนกว่าเงื่อนไขที่เช็กจะมีสถาวะเป็นเท็จ จึงออกจากลูป เช่น

```
int n=0;
while ( n!=5 )
    { cout<< n <<","; ++n; }
```

Output: 0,1,2,3,4

3) ลูป do หรือที่เรียกอีกชื่อว่า ลูป do while เหตุผลตอนท้ายของลูปต้องมี while เพื่อเป็นตัวเช็กเงื่อนไข จากลักษณะดังกล่าว จึงมีลักษณะเด่นคือ มีการดำเนินการคำสั่งในลูป อย่างน้อยหนึ่งครั้ง เช่น

```
int n=0;
do
    { cout<< n <<","; ++n;}
while (n!=0)
```

Output: 0

7.2.2.2 การตัดสินใจ เป็นการดำเนินการในลักษณะการตัดสินใจ ผลของการตัดสินใจทำให้มีการกระโดด ของ Flow Command ไปจุดอื่น ๆ

1) if เป็นการติดสินใจว่า หากสถาวะเป็นจริง จึงทำคำสั่งถัดไปจาก if หากเป็นเท็จก็ให้ข้ามคำสั่งที่ถัดจาก if ไปยังคำสั่งตำแหน่งถัดออกไปอีก เช่น

```
if (x>100)
statement;   //คำสั่งถัดไปซึ่ง มีคำสั่งเดียว จะกระทำเมื่อเงื่อนไขเป็นจริง
statement; // คำสั่งถัดไปอีก จะกระทำเมื่อเงื่อนไขเป็นเท็จ (เงื่อนไขจริงก็ยังกระทำเนื่อง
           // จากเป็นการ Flow command ลงมา )
```

```
if (x>100)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ statement;
  statement;    // มีหลายคำสั่ง
  statement;
}
```

2) if else

มีลักษณะคล้าย if ต่างตรงที่ว่า ใน if else หากเงื่อนไขเป็นเท็จ ก็จะมีคำสั่งให้มันกระทำด้วยไม่ใช่ Flow ผ่านไปยังคำสั่งถัดไปเหมือน if

```
if (x>100)
statement; // เมื่อ เงื่อนไขเป็นจริง
else
statement; // เมื่อเงื่อนไขเป็นเท็จ
statement; // สิ้นสุดขอบเขตขอบ if else ดังนั้น เงื่อนไขทั้งจริง หรือเท็จก็ไม่มีผล
           // แต่อย่างใด นั่นคือคำสั่งนี้จึงมีการกระทำ ตามการ Flow Command
```

```
if (x>100)
{ statement;
  statement;
  statement;
}
else
{ statement;
  statement;
  statement;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) switch ทำหน้าที่เลือกการ Flow Command ในลักษณะเป็นชุด คล้าย ๆ กับ การใช้ if else ซ้อนกันหลาย ๆ ครั้ง

```
switch (x)
{
  case 1: cout<< " hi " <<endl; break;   // เมื่อ x มีค่าเป็น 1
  case 2: cout<< " hello" <<endl; break; // เมื่อ x มีค่าเป็น 2
  case 3: cout<< " hey" <<endl; break;   // เมื่อ x มีค่าเป็น 3
  default: cout<< " oh no "<<endl;   // เมื่อ x มีค่าเป็นค่าอื่น ๆ
}
```

4) goto ทำหน้ากระโดดไปยังส่วนอื่น ของโปรแกรมแบบไม่มีเงื่อนไข ควรหลีกเลี่ยงการใช้คำสั่งนี้ เพราะการดำเนินการของ goto จะทำให้เกิด *รหัส สปาเกตตี* ขึ้น ซึ่งยากต่อการที่จะทำความเข้าใจและแก้ไขจุดบกพร่อง

```
goto  log;
statement;      //คำสั่งอื่น ๆ
log:
statement   // จะกระโดดมาทำงาน ที่จุดนี้
```

7.2.2.3 การเรียกใช้ทรัพยากรของระบบ

เป็นส่วนที่มีการเรียกใช้ทรัพยากรที่พึงหาได้ในระบบ เช่น การร้องการใช้งานฟังก์ชันทางกราฟฟิกโดยผ่าน initgraph การใช้ฟังก์ชันทาง Dos เช่น mouse interupt , keyboard interupt

7.2.2.4 การจัดการข้อผิดพลาดที่จะบังเกิดขึ้น

ทำหน้าที่ควบคุมการทำงานของโปรแกรมให้ถูกต้อง ในงานฟังก์ชันส่วนใหญ่ จะมีการ Return ค่ากลับมาเพื่อบอกให้ทราบถึงผลของการดำเนินการคำสั่งนั้น ๆ ว่าสำเร็จหรือล้มเหลว ซึ่งอาจ Return ค่ากลับมาที่จุดเรียกใช้คำสั่งนั้น ๆ เลย หรือ ส่งค่าผ่านตัวแปรพิเศษของระบบ เรานำค่าต่าง ๆ เหล่าที่ได้เหล่านี้มาประมวลว่า ถ้าล้มเหลวแล้วควรทำอย่างไร จะทำพยายามเรียกซ้ำ หรือลองวิธีอื่น เช่น การ Save ชื่อไฟล์ที่มีการตั้งชื่อที่ผิด (Bad filename) เช่น <>myname.pdf เมื่อเราทราบว่าการดำเนินการล้มเหลว ก็ทำการส่งข้อความเตือนให้ผู้ใช้ (User) ออกไป เพื่อให้ผู้ใช้ได้แก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 I/O Secsion

เป็นส่วนที่มีทำหน้าที่รับ / ส่งข้อมูล ติดต่อกับอุปกรณ์ Hardware ภายนอก อันได้แก่ Mouse, Keyboard, Monitor, ISA card โดยติดต่อผ่านคำสั่ง inport , outport ใน ISA card ติดต่อผ่านการฟังกัชันทาง dos interupt เช่น int86, intdos เป็นต้น

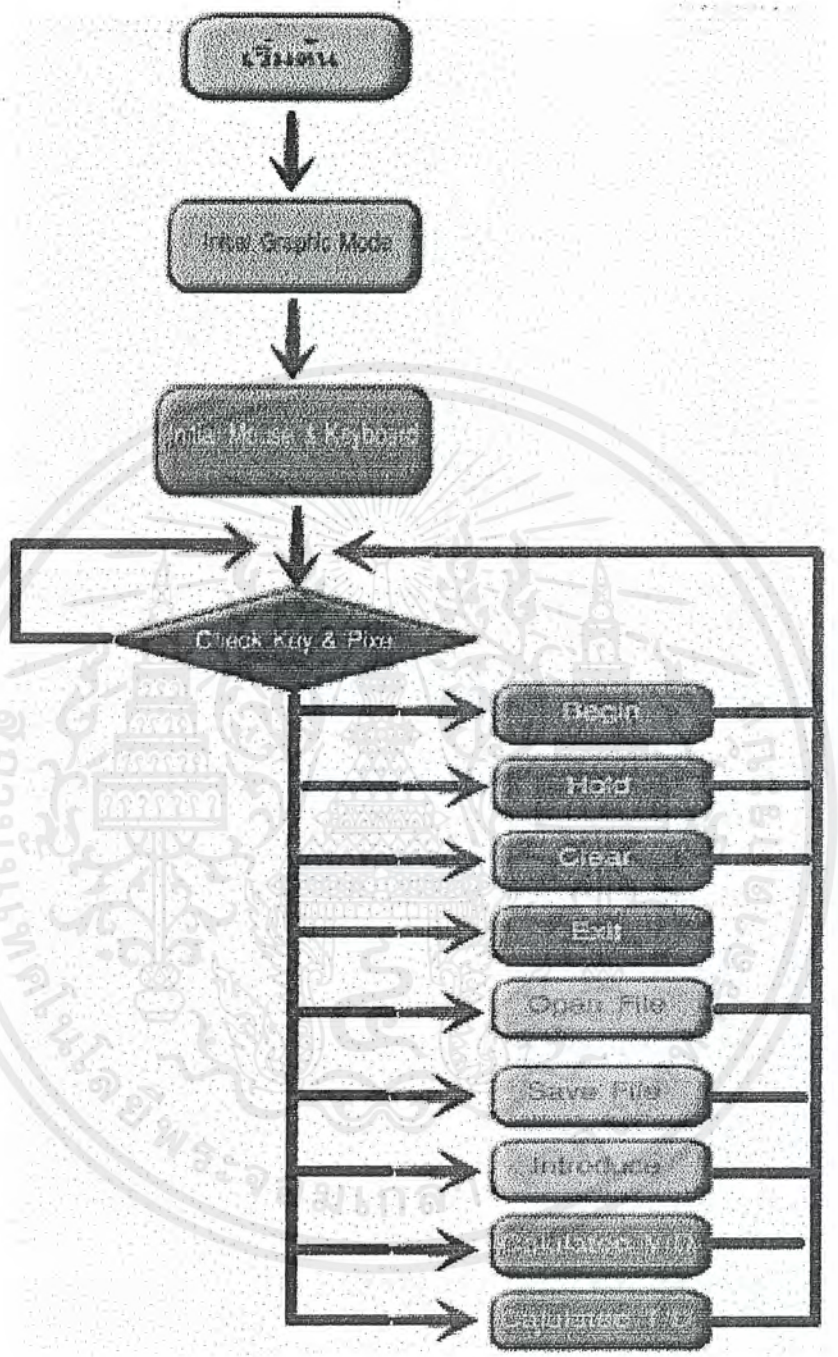
7.4 อุปกรณ์ปลายทาง

อันได้แก่ mouse, keyboard, monitor,ISA card ทำหน้าที่เป็น Input , Output ให้กับโปรแกรม อุปกรณ์เหล่านี้ผลิตมาจากหลายบริษัท หลายยี่ห้อ จึงมีคุณลักษณะพิเศษบางอย่างที่ไม่เหมือนกัน ตามแต่ Option ที่ใส่ลงไป แต่ทุกอันจะมี Standart ร่วมกันอยู่ ซึ่งทั้งหมดสามารถรู้และเข้าใจคำสั่ง ในการทำงานคำสั่งพื้นฐานได้ เรียกว่า Standart Function โดยปกติแล้วขณะที่โปรแกรมมีการร้องขอการใช้ทรัพยากรของระบบ ผ่านทาง I/O โดยใช้ Standart Driver ซึ่งจะเป็น Standart Function นั่นเอง หากต้องการใช้ความสามารถเต็ม (Full Option) ของอุปกรณ์ ต้องระบุ Driverให้กับโปรแกรมด้วย โดยใส่ Driver เฉพาะนั้น ๆ ซึ่งจะหาได้จากผู้ผลิตอุปกรณ์นั้น ๆ เพื่อความเข้าใจที่ชัดเจนยิ่งขึ้น จะขอยกตัวอย่าง ดังนี้

สถานการณ์

นาย A ซื้อ Mouse 4D ของ A4TECH มา ซึ่งมีลักษณะพิเศษที่พึงสังเกตุได้ภายนอกคือ มี 3 ปุ่ม มีลูกกลิ้งตรงระหว่างปุ่ม 2 อัน นาย A นำมา Plug กับ Computer ของตน แล้วทำการเปิด Windows 95 เมื่อเข้าไปแล้วนาย A ก็สามารถใช้ Mouse ได้ตามปกติ โดยไม่ต้องป้อน Driver ให้ Windows แต่อย่างใด หลังจากใช้งานได้สักพักนาย A สังเกตุได้ว่า ใช้ได้เพียง 2 ปุ่มเท่านั้น ปุ่มกลางและลูกกลิ้งไม่สามารถใช้งานได้ ที่ใช้ได้ 2 ปุ่มนั้น นั่นคือ Standard Function หรือ Standard Driver ของ mouse แบบ 2 ปุ่ม ที่ Windows เรียกใช้อยู่ ตัว Windows เองจะไม่รู้จัก ปุ่มที่ 3 หรือ ลูกกลิ้งแต่อย่างใด หากต้องการใช้ความสามารถให้เต็มประสิทธิภาพของ mouse ที่ซื้อมา นาย A ต้องป้อน Driver ตัวใหม่ที่จะได้มาด้วยพร้อมกับ mouse ที่ซื้อมา ให้กับ Windows เพื่อที่ Windows จะได้รู้จักและสามารถติดต่อกับ mouse ในการทำงานลักษณะพิเศษอื่น ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 Flow chart ของส่วนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้เปิดโปรแกรมแล้ว Run โปรแกรมก็จะโหลด Graphic Driver แล้วเข้าสู่ Graphic Mode หลังจากนั้นก็ทำการใช้ Interupt เพื่อขอใช้ทรัพยากร 2 ตัวคือ Mouse และ Keyboard โดย Mouse จะใช้ Software Interrupt Interface ผ่านคำสั่ง intx86 ซึ่งใช้อินเตอร์รัพท์ที่ 0x33 โปรแกรมย่อยที่ 3 ส่วน Keyboard จะใช้ Dos Interrupt Interface ผ่านคำสั่ง intdos โดยเซ็ท Ah=0x06 และ Dl=0xff

การทำงานของ Mouse ในโหมดนี้ ทำให้ระบบรับรู้ว่ามี Mouse อยู่ และสร้าง Pointer รู้ลูกศรเป็น Mouse Pointer พร้อมทั้งรับรู้ถึงการคืนไหวของ Mouse โดยมีการคืนค่ากลับเป็นปุ่มที่กด (ปุ่มซ้าย, ปุ่มกลาง, ปุ่มขวา) กับพิกัดที่มันอยู่

การทำงานของ Keyboard ในโหมดนี้เป็นโหมดที่ตรวจสอบว่าที่ Buffer ของ Keyboard มีตัวอักษรอะไรบ้าง ถ้ามีก็จะรับค่าไป แต่ถ้าไม่พบก็จะไม่รอให้ผู้ใช้ (user) กดคีย์แต่อย่างใด

หลังจากนั้นก็ทำการตรวจสอบการกดคีย์ และการกด Mouse ของผู้ใช้ว่าตรงกับ คีย์หรือพิกัดอยู่ในช่วงที่เราต้องการหรือไม่ (ตรวจสอบการกดของ Hotkey และ การกด Mouse ในช่วง Pixel ของปุ่มบนหน้าจอ) ถ้าใช่ก็ให้ไปทำในโปรแกรมย่อยของแต่ละการทำงานของแต่ละรหัส หากไม่มีการกดคีย์หรือเมาส์ หรือ เป็นรหัสที่ไม่ต้องการ ก็ให้รอจนกว่าผู้ใช้จะกด

หากเข้าสู่การทำงานของโปรแกรมย่อยแล้ว หลังจากเสร็จสิ้นการทำงานของโปรแกรมย่อยก็จะเข้าสู่การรอการกดคีย์หรือเมาส์อย่างเดิม ยกเว้นผู้ใช้ป้อนรหัสที่เป็นการทำงานของ Exit หรือการออกจากโปรแกรมนั่นเอง

การทำงานของ Begin ก็จะเป็นการทำงานในการวาดกราฟ ที่ได้จากการวัดสัญญาณภายนอกที่ผ่านมาทาง ISA Card เมื่อ Begin ทำงานหน้าจอจะวาดกราฟ พร้อมคำนวณความถี่ แล้วบอกความถี่ผ่านหน้าจอคอมพิวเตอร์, Hold ทำหน้าที่หยุดกราฟนั้นไว้ชั่วคราว, Clear ทำหน้าที่ลบกราฟบนหน้าจอ, Exit ออกจะโปรแกรม

การทำงานใน Open และ Save จะมีลักษณะคล้าย ๆ กัน เมื่อเข้าไปแล้วโปรแกรมจะทำการโหลดชื่อไฟล์ทั้งหมดใน Directory ที่ทำงานอยู่ พร้อมทั้งมีแถบสว่าง ให้ผู้ใช้สามารถ กดลูกศรเลื่อนขึ้นหรือลง เพื่อเลือกไฟล์ที่ต้องการ Open หรือ Save ซึ่งในการทำงานของ Save นั้นจะพิเศษกว่าว Open เล็กน้อย คือสามารถตั้งชื่อไฟล์ใหม่ เพื่อทำการจัดเก็บ หากไฟล์ซ้ำ ก็จะเตือนว่าจะเขียนทับหรือไม่

ส่วน V/D และ T/D เป็นการตั้งค่าในการวาดของกราฟ โดยสามารถกดเมาส์ตามขนาดตามต้องการ หรือ ใช้ Manual Mode เพื่อใส่ค่าของขนาดที่ต้องการก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
ผลการทดลองและสรุปผล

8.1) ผลการทดลอง

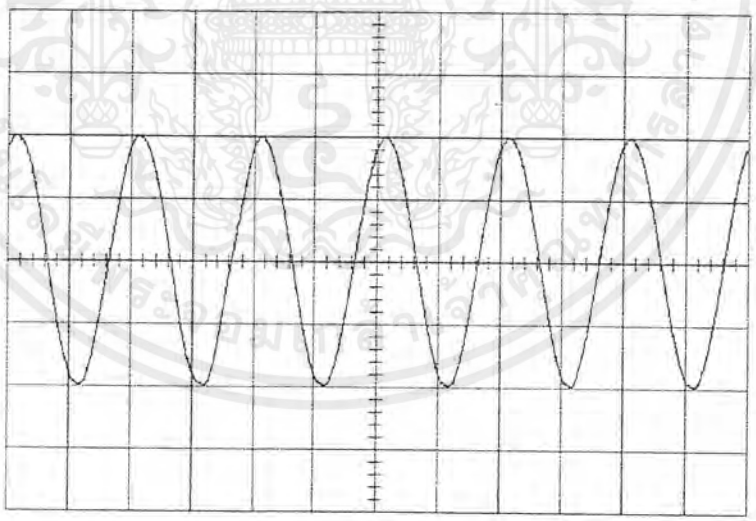
เมื่อทำการออกแบบวงจรส่วนต่างๆ และเขียนโปรแกรมเป็นที่เรียบร้อยแล้วขั้นตอนต่อไปคือการทดสอบการทำงานของเครื่องออสซิลโลสโคปบนคอมพิวเตอร์เครื่องนี้ โดยการทดลองวัดรูปคลื่นสัญญาณแบบต่างๆ ได้แก่ สัญญาณรูปซายน์ (Sine wave) สัญญาณรูปสามเหลี่ยม (Triangle wave) และสัญญาณรูปสี่เหลี่ยม (Square wave) ที่ความถี่และขนาดต่างๆ กันดังนี้

8.1.1) การทดลองวัดสัญญาณรูปซายน์

ทำการป้อนสัญญาณอินพุทเป็นรูปซายน์ที่ขนาดและ ความถี่ต่างๆ กันดังนี้

8.1.1.1) สัญญาณรูปซายน์ที่ขนาดต่างๆ โดยให้ความถี่มีค่าคงที่ (50 kHz)

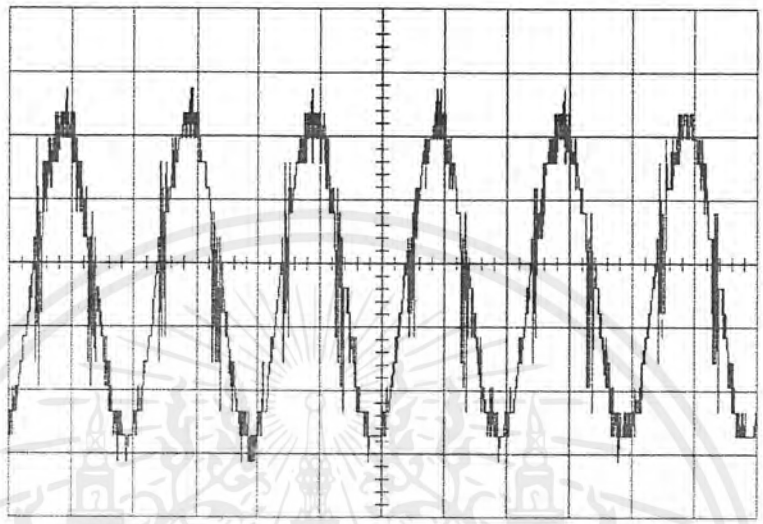
1) ป้อนสัญญาณรูปซายน์ขนาด 10 Vp (20 Vpp) ความถี่ 50 kHz



รูปที่ 8.1 สัญญาณรูปซายน์ขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

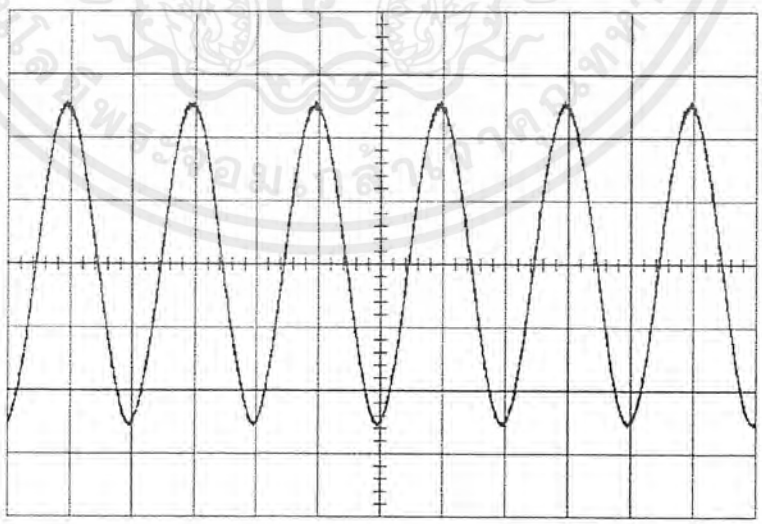
2) ป้อนสัญญาณรูปซายน์ขนาด 0.5 Vp (1 Vpp) ความถี่ 50 kHz



รูปที่ 8.2 สัญญาณรูปซายน์ขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV)

8.1.1.2) สัญญาณรูปซายน์ที่ความถี่ต่างๆ โดยให้ขนาดมีค่าคงที่ (5 Vp)

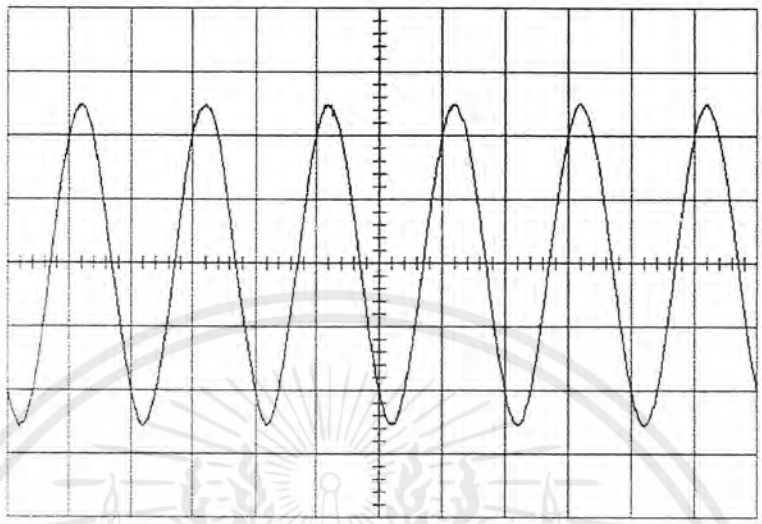
1) ป้อนสัญญาณรูปซายน์ที่ความถี่ 10 Hz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.3 สัญญาณรูปซายน์ที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV)

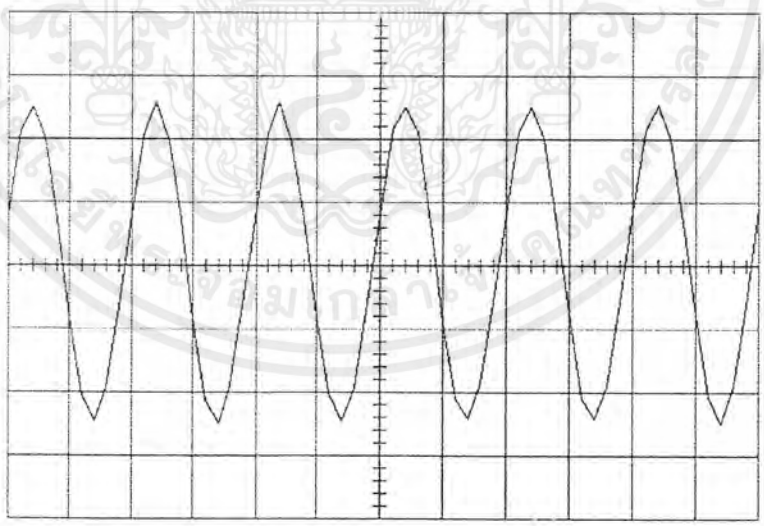
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ป้อนสัญญาณรูปซายน์ที่ความถี่ 10 kHz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.4 สัญญาณรูปซายน์ที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV)

3) ป้อนสัญญาณรูปซายน์ที่ความถี่ 1 MHz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.5 สัญญาณรูปซายน์ที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV)

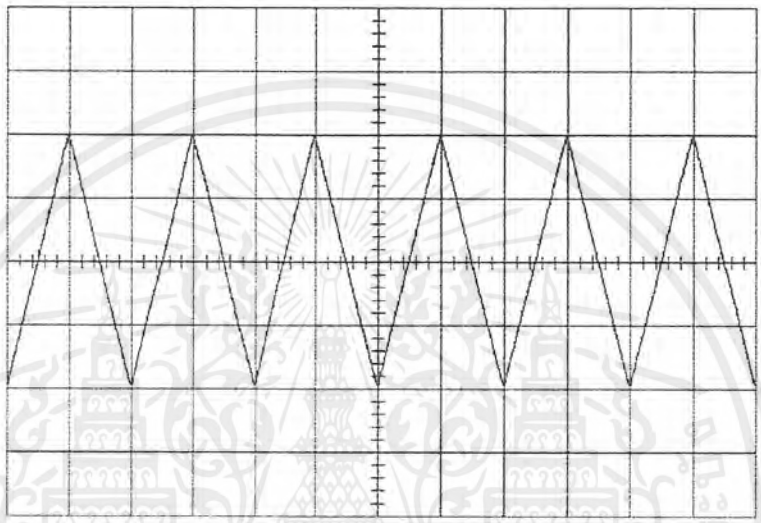
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.2) การทดลองวัดสัญญาณรูปสามเหลี่ยม

ทำการป้อนสัญญาณอินพุทเป็นสามเหลี่ยมที่ขนาดและ ความถี่ต่างๆ กันดังนี้

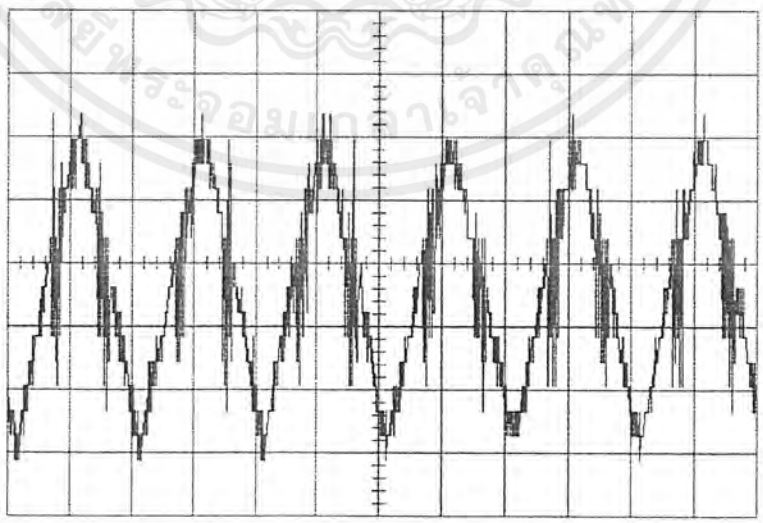
8.1.2.1) สัญญาณรูปสามเหลี่ยมที่ขนาดต่างๆ โดยให้ความถี่มีค่าคงที่ (50kHz)

1) ป้อนสัญญาณรูปสามเหลี่ยมขนาด 10 Vp (20 Vpp) ความถี่ 50 kHz



รูปที่ 8.6 สัญญาณรูปสามเหลี่ยมขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV)

2) ป้อนสัญญาณรูปสามเหลี่ยมขนาด 0.5 Vp (1 Vpp) ความถี่ 50 kHz

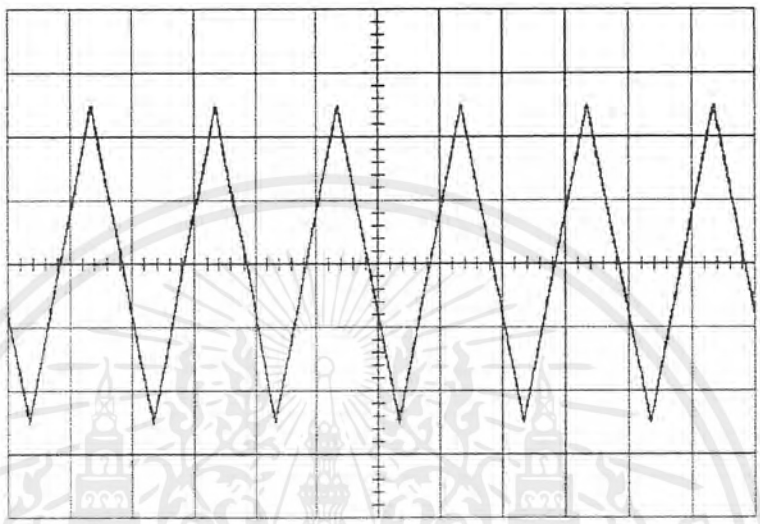


รูปที่ 8.7 สัญญาณรูปสามเหลี่ยมขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

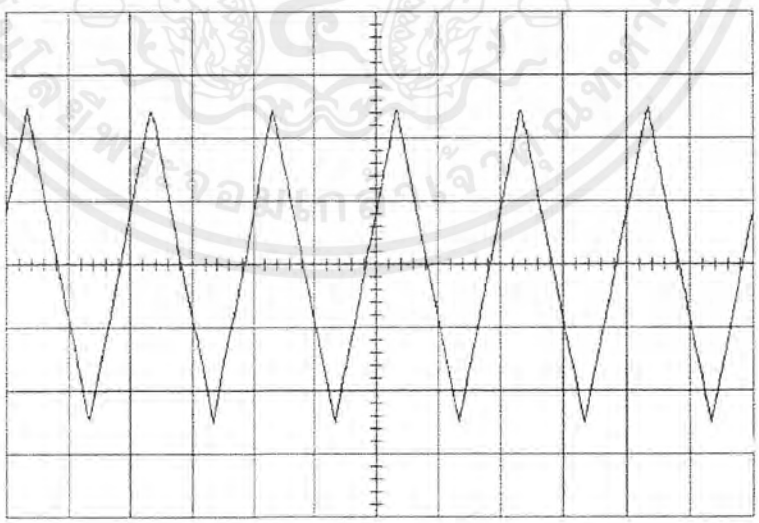
8.1.2.2) สัญญาณรูปสามเหลี่ยมที่ความถี่ต่างๆ โดยให้ขนาดมีค่าคงที่ (5 Vp)

1) ป้อนสัญญาณรูปสามเหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.8 สัญญาณรูปสามเหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV)

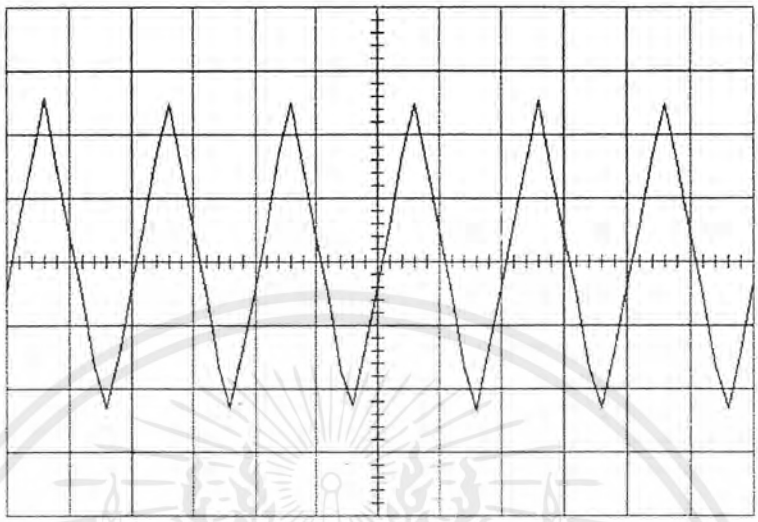
2) ป้อนสัญญาณรูปสามเหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.9 สัญญาณรูปสามเหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ป้อนสัญญาณรูปสามเหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (10 Vpp)



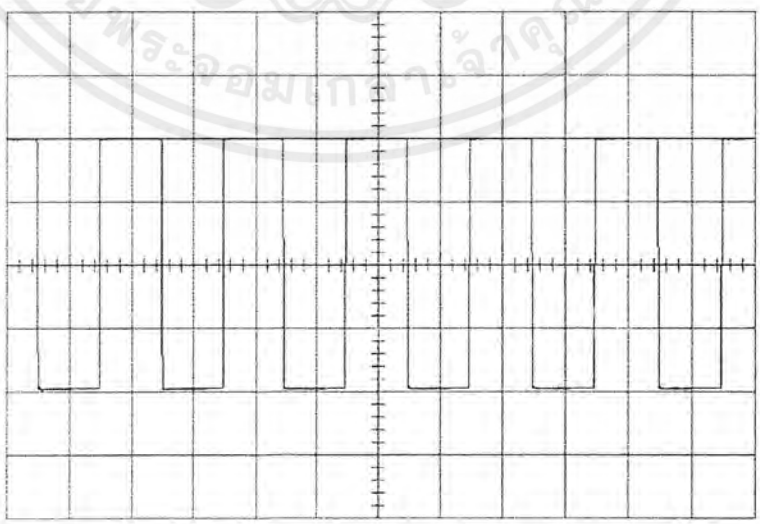
รูปที่ 8.10 สัญญาณรูปสามเหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV)

8.1.3) การทดลองวัดสัญญาณรูปสี่เหลี่ยม

ทำการป้อนสัญญาณอินพุทเป็นรูปสี่เหลี่ยมที่ขนาดและ ความถี่ต่างๆ กันดังนี้

8.1.3.1) สัญญาณรูปสี่เหลี่ยมที่ขนาดต่างๆ โดยให้ความถี่มีค่าคงที่ (50kHz)

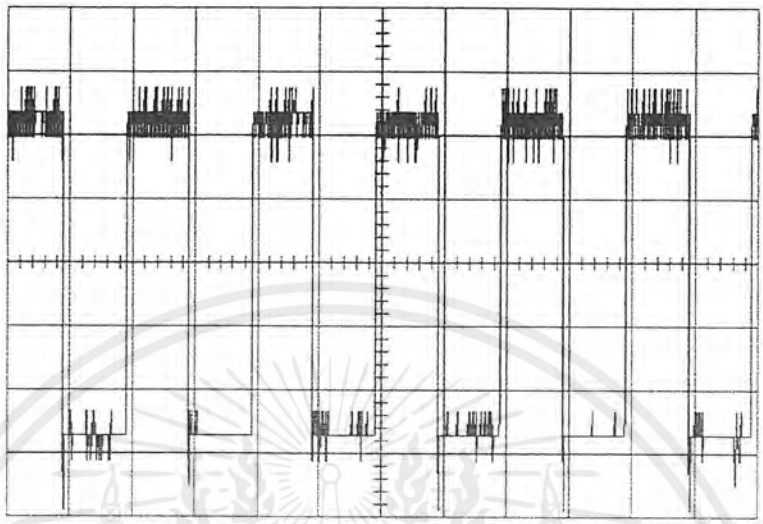
1) ป้อนสัญญาณรูปสี่เหลี่ยมขนาด 10 Vp (20 Vpp) ความถี่ 50 kHz



รูปที่ 8.11 สัญญาณรูปสี่เหลี่ยมขนาด 10 Vp ความถี่ 50 kHz (5 V/DIV , 10 μS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

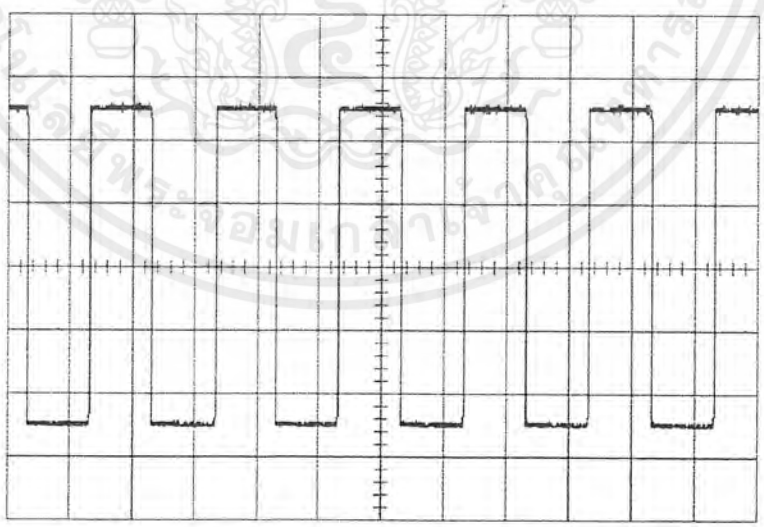
2) ป้อนสัญญาณรูปสี่เหลี่ยมขนาด 0.5 Vp (1 Vpp) ความถี่ 50 kHz



รูปที่ 8.12 สัญญาณรูปสี่เหลี่ยมขนาด 0.5 Vp ความถี่ 50 kHz (0.2 V/DIV , 10 μS/DIV)

8.1.3.2) สัญญาณรูปสี่เหลี่ยมที่ความถี่ต่างๆ โดยให้ขนาดมีค่าคงที่ (5 Vp)

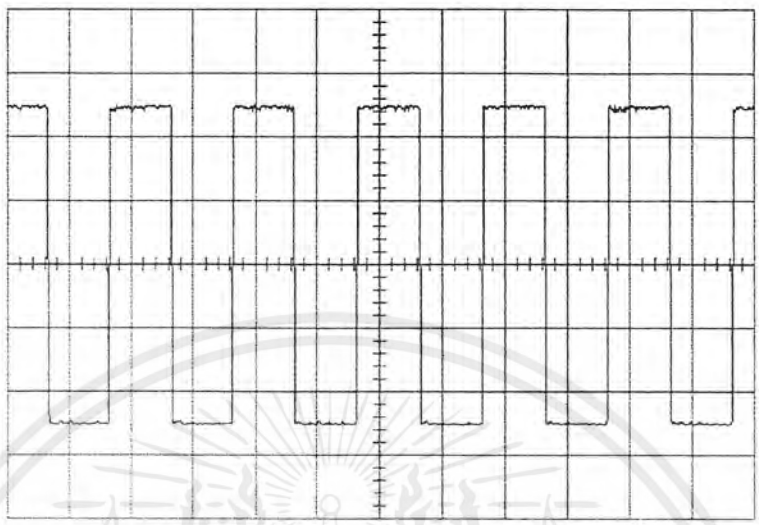
1) ป้อนสัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.13 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 Hz ขนาด 5 Vp (2 V/DIV , 50 mS/DIV)

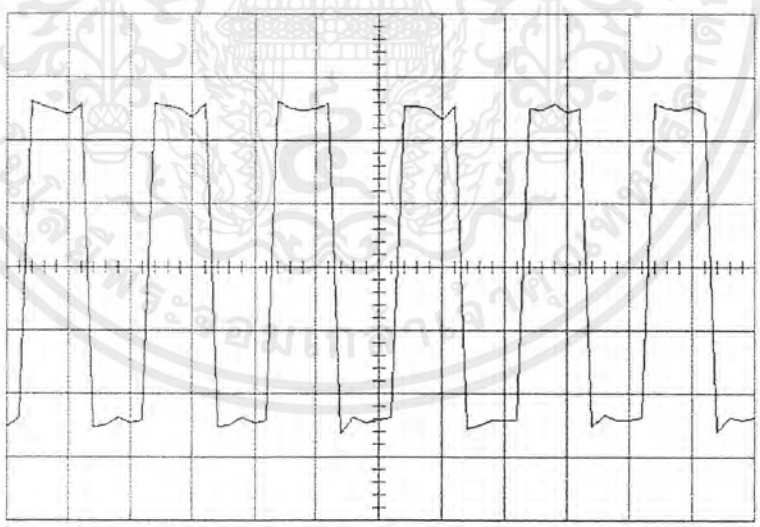
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ป้อนสัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.14 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 10 kHz ขนาด 5 Vp (2 V/DIV , 50 μS/DIV)

3) ป้อนสัญญาณรูปสี่เหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (10 Vpp)



รูปที่ 8.15 สัญญาณรูปสี่เหลี่ยมที่ความถี่ 1 MHz ขนาด 5 Vp (2 V/DIV , 0.5 μS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.4) การคำนวณเปอร์เซ็นต์ความผิดพลาดของความถี่ของสัญญาณที่วัดได้

ทำการวัดความถี่ของสัญญาณที่ปรากฏทางหน้าจอคอมพิวเตอร์ แล้วไปเปรียบเทียบกับความถี่ที่วัดได้จริงจากเครื่องออสซิลโลสโคป แล้วคำนวณหาเปอร์เซ็นต์ความผิดพลาดโดยมีสูตรคือ

$$\% \text{ Error} = \frac{\text{ค่าที่วัดได้ - ค่าอ้างอิง}}{\text{ค่าอ้างอิง}} \times 100$$

ทำการคำนวณหาความผิดพลาดของสัญญาณรูปต่างๆ ที่ความถี่ต่างๆ ดังนี้

รูปสัญญาณ	ความถี่จริง	ความถี่ที่วัดได้	% Error
รูปซายน์	10 Hz	9.634 Hz	-3.66 %
	10 kHz	9.803 kHz	-1.977 %
	1 MHz	1 MHz	0 %
รูปสามเหลี่ยม	10 Hz	9.615 Hz	-3.815 %
	10 kHz	9.803 kHz	-1.97 %
	1 MHz	1.02 MHz	+2.00 %
รูปสี่เหลี่ยม	10 Hz	9.709 Hz	-2.91 %
	10 kHz	9.901 kHz	-0.99 %
	1 MHz	1.053 MHz	+5.3 %

ตารางที่ 8.1 แสดงความผิดพลาดของสัญญาณที่ความถี่ต่างๆ

8.2) สรุปผลการทดลอง และปัญหาต่างๆ

จากผลการทดลองวัดสัญญาณรูปคลื่นที่ขนาด และความถี่ต่างๆ ทำให้ทราบถึงปัญหา และข้อจำกัดต่างๆ ในการทำงานของเครื่องออสซิลโลสโคปบนเครื่องคอมพิวเตอร์ ดังนี้

1) ยังไม่สามารถวัดสัญญาณที่มีขนาดเล็กหรือมีความละเอียดมากได้ เพราะรูปสัญญาณที่ปรากฏบนหน้าจอคอมพิวเตอร์นั้นมีลักษณะไม่เรียบ โดยเฉพาะการวัดสัญญาณที่มีแรงดันต่ำๆ จะสามารถเห็นความไม่เรียบนี้ได้อย่างชัดเจน ซึ่งมีสาเหตุมาจาก

- ความละเอียดของ A/D ที่ใช้มีขนาด 8 บิต ดังนั้นจึงมีความละเอียดสูงสุดได้ 256 ค่า และเราออกแบบให้สามารถวัดสัญญาณขนาดใหญ่สุดได้ 20 Vpp ทำให้ A/D เปลี่ยนค่าไป 1 ค่าเมื่อแรงดันเปลี่ยนไป 0.078 V
- สัญญาณรบกวนจากภายนอก หรือเกิดจากไอซี A/D เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ยังไม่สามารถวัดสัญญาณที่มีความถี่สูงมากกว่า 1 MHz ได้ เพราะถูกจำกัดโดยความถี่ของ Sampling clock ซึ่งมีความถี่ 10 MHz เป็นความถี่สูงสุด ทำให้เกิดความผิดพลาดของรูปสัญญาณมากขึ้นเมื่อความถี่สูงขึ้น ซึ่งสอดคล้องกับผลการทดลองในตารางที่ 8.1 เพราะจะเห็นได้ว่า ถ้าสัญญาณที่ต้องการวัดเป็นสัญญาณที่มีการเปลี่ยนแปลงแบบทันทีทันใดอย่างสัญญาณรูปสี่เหลี่ยมแล้ว รูปสัญญาณจะมีความผิดพลาดมากขึ้นเมื่อความถี่สูงเกือบ 1 MHz ทำให้ความถี่ที่วัดได้นั้นมีความผิดพลาดมากขึ้นตามด้วย

สาเหตุที่ไม่ออกแบบวงจร Clock generator ให้มีความถี่สูงกว่านี้ ก็เนื่องจากความเร็วในการอ่าน / เขียน หรือ Access time ของ SRAM ที่ใช้นี้มีค่าจำกัดอยู่ที่ประมาณ 70 ns ดังนั้นความถี่สูงสุดที่ SRAM สามารถอ่าน / เขียน ได้ทันคือ 1/(70 ns) = 14.3 MHz แต่เราก็ควรใช้ความถี่ให้ต่ำลงมาอีกเพื่อให้มั่นใจได้ว่า SRAM สามารถอ่าน / เขียนได้ทันแน่นอน

8.3) แนวทางการปรับปรุงแก้ไข

จากปัญหาที่เกิดขึ้นสามารถมีแนวทางการปรับปรุงแก้ไขได้ดังนี้

1) จากข้อจำกัดที่ไม่สามารถวัดสัญญาณขนาดเล็กได้นั้น ถ้าปัญหาเกิดจากสัญญาณรบกวนที่มาจากคอมพิวเตอร์ต้องหาวิธีกำจัดสัญญาณรบกวนให้ได้ แต่ถ้ามีสาเหตุจากไอซี A/D ต้องลองเปลี่ยนไอซีเป็นเบอร์อื่นดู ส่วนข้อจำกัดเนื่องจากความละเอียดของ A/D นั้น อาจแก้โดยใช้วงจรขยายทำการขยายสัญญาณที่มีขนาดเล็กๆ ให้มีขนาดใหญ่ขึ้น ก็จะทำให้วัดสัญญาณที่มีขนาดเล็กๆ ได้ดีขึ้น

2) ข้อจำกัดอันเนื่องจาก Access time ของ SRAM สามารถแก้ไขได้ โดยการหา SRAM ที่มี Access time เร็วกว่านี้

8.4) ประโยชน์ที่ได้รับจากการทำโครงงาน

ประโยชน์อันดับแรกที่ได้รับโดยตรงและเป็นจุดประสงค์หลักที่ภาควิชาฯ กำหนดให้ต้องทำโครงงานก็คือ ความรู้และความเข้าใจที่มีต่อตัวโครงงานที่เราตัดสินใจเลือกหัวข้อในการทำ ซึ่งความรู้และความเข้าใจที่ว่านี้เป็นความรู้ความเข้าใจจริงๆ เพราะได้ลงมือปฏิบัติ ใช่ว่าเรียนแต่ทฤษฎีแต่เฉพาะในห้องเรียน การลงมือปฏิบัติจะไม่ราบรื่นเหมือนในทางทฤษฎี จะต้องเกิดปัญหา ข้อติดขัดต่างๆ นานัปการ เมื่อเราได้ประสบความยากลำบากเหล่านี้ต้องไม่ย่อท้อ พยายามคิดถึงเหตุความเป็นไปได้ของปัญหาต่างๆ แล้วตั้งสมมุติฐานขึ้นมา แล้วทำการตรวจสอบเป็นข้อๆ ไป จนพบต้นเหตุของปัญหาที่แท้จริง หากพยายามแล้วก็ยังไม่พบปัญหาก็หาทางเลือกอื่นๆ เช่น ถามเพื่อนๆ พี่ๆ ที่พวกเขาอาจมีความรู้และมุมมองต่างๆ ที่ไม่เหมือนกับที่เราคิด จากนั้นจึงนำข้อเสนอแนะเหล่านั้นกลับมาคิดพิจารณาลำดับความ และตั้งสมมุติฐานใหม่แล้วทำการตรวจสอบอีกครั้งหนึ่ง หากยังไม่พบสาเหตุก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะนำปัญหาต่างๆ ไปปรึกษากับอาจารย์ที่ปรึกษา ซึ่งอาจารย์ก็จะให้ข้อคิด คำแนะนำดีๆ กลับมาเสมอ จะเห็นได้ว่ากระบวนการดังกล่าวเป็นกระบวนการคิดเป็นขั้นเป็นตอนเป็นลำดับความ ซึ่งจะก่อให้เกิดประโยชน์ที่สองตามมาคือ กระบวนการคิดเป็นลำดับ หากทำการฝึกบ่อยๆ ก็จะเป็นการจุดประกายความคิด ซึ่งเป็นบ่อเกิดของความคิดสร้างสรรค์ที่จะเกิดขึ้นต่อมาภายหลังนั่นเอง ประโยชน์ที่สามคือ เป็นการฝึกการอยู่ร่วมกันในสังคม รู้จักการช่วยเหลือซึ่งกันและกัน ทั้งผู้ร่วมงาน และผู้ทำโครงงานอื่น ฝึกการอยู่ในกลุ่มพี่ กลุ่มเพื่อน กลุ่มอาจารย์ อีกทั้งเป็นการใช้เวลาให้เกิดประโยชน์อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

May 1999

LM6161/LM6261/LM6361
High Speed Operational Amplifier

General Description

The LM6161 family of high-speed amplifiers exhibits an excellent speed-power product in delivering 300 V/µs and 50 MHz unity gain stability with only 5 mA of supply current. Further power savings and application convenience are possible by taking advantage of the wide dynamic range in operating supply voltage which extends all the way down to +5V.

These amplifiers are built with National's VIP™ (Vertically Integrated PNP) process which provides fast PNP transistors that are true complements to the already fast NPN devices. This advanced junction-isolated process delivers high speed performance without the need for complex and expensive dielectric isolation.

Features

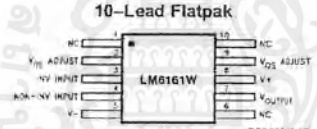
- High slew rate 300 V/µs
- High unity gain freq 50 MHz
- Low supply current 5 mA
- Fast settling 120 ns to 0.1%
- Low differential gain <0.1%
- Low differential phase 0.1°
- Wide supply range 4.75V to 32V
- Stable with unlimited capacitive load
- Well behaved; easy to apply

Applications

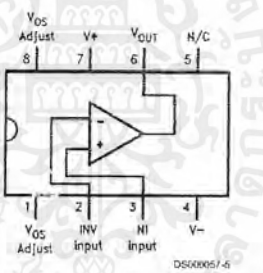
- Video amplifier
- High-frequency filter
- Wide-bandwidth signal conditioning
- Radar
- Sonar

Connection Diagrams

10-Lead Flatpak



See NS Package Number W10A



See NS Package Number J08A, N08E or M08A

Temperature Range			Package	NSC Drawing
Military $-55^{\circ}C \le T_A \le +125^{\circ}C$	Industrial $-25^{\circ}C \le T_A \le +85^{\circ}C$	Commercial $0^{\circ}C \le T_A \le +70^{\circ}C$		
	LM6261N	LM6361N	8-Pin Molded DIP	N08E
LM6161J/883 5962-8962101PA		LM6361J	8-Pin Ceramic DIP	J08A
	LM6261M	LM6361M	8-Pin Molded Surface Mt.	M08A
LM6161WG/883 5962-8962101XA			10-Lead Ceramic SOIC	WG10A
LM6161W/883 5962-8962101HA			10-Pin Ceramic Flatpak	W10A

VIP™ is a trademark of National Semiconductor Corporation.

© 1999 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 12)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage ($V^+ - V^-$)	36V
Differential Input Voltage (Note 8)	±8V
Common-Mode Voltage Range (Note 10)	$(V^+ - 0.7V)$ to $(V^- + 0.7V)$
Output Short Circuit to GND (Note 1)	Continuous
Soldering Information	
Dual-In-Line Package (N, J) Soldering (10 sec.)	260°C
Small Outline Package (M) Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Storage Temp Range	−65°C to +150°C
Max Junction Temperature	150°C
ESD Tolerance (Notes 6, 7)	±700V

Operating Ratings (Note 12)

Temperature Range (Note 2)	
LM6161	$-55°C \le T_J \le +125°C$
LM6261	$-25°C \le T_J \le +85°C$
LM6361	$0°C \le T_J \le +70°C$
Supply Voltage Range	4.75V to 32V

DC Electrical Characteristics

The following specifications apply for Supply Voltage = ±15V, V_{CM} = 0, $R_L \ge 100\ k\Omega$ and $R_S = 50\Omega$ unless otherwise noted. **Boldface** limits apply for $T_J = T_{MIN}$ to T_{MAX}; all other limits T_J = 25°C.

Symbol	Parameter	Conditions	Typ	LM6161	LM6261	LM6361	Units
				Limit (Notes 3, 11)	Limit (Note 3)	Limit (Note 3)	
V_{OS}	Input Offset Voltage		5	7	7	20	mV
				10	**9**	**22**	Max
V_{OS} Drift	Input Offset Voltage Average Drift		10				µV/°C
I_b	Input Bias Current		2	3	3	5	µA
				6	**5**	**6**	Max
I_{OS}	Input Offset Current		150	350	350	1500	nA
				800	**600**	**1900**	Max
I_{OS} Drift	Input Offset Current Average Drift		0.4				nA/°C
R_{IN}	Input Resistance	Differential	325				kΩ
C_{IN}	Input Capacitance	A_V = +1 @ 10 MHz	1.5				pF
A_{VOL}	Large Signal Voltage Gain	V_{OUT} = ±10V, R_L = 2 kΩ (Note 9)	750	550	550	400	V/V
				300	**400**	**350**	Min
		R_L = 10 kΩ (Note 9)	2900				V/V
V_{CM}	Input Common-Mode Voltage Range	Supply = ±15V	+14.0	+13.9	+13.9	+13.8	Volts
				+13.8	**+13.8**	**+13.7**	Min
			−13.2	−12.9	−12.9	−12.8	Volts
				−12.7	**−12.7**	**−12.7**	Min
		Supply = +5V (Note 4)	4.0	3.9	3.9	3.8	Volts
				3.8	**3.8**	**3.7**	Min
			1.8	2.0	2.0	2.1	Volts
				2.2	**2.2**	**2.2**	Max
CMRR	Common-Mode Rejection Ratio	$-10V \le V_{CM} \le +10V$	94	80	80	72	dB
				74	**76**	**70**	Min
PSRR	Power Supply Rejection Ratio	$\pm 10V \le V^+ \le \pm 16V$	90	80	80	72	dB
				74	**76**	**70**	Min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC Electrical Characteristics (Continued)

The following specifications apply for Supply Voltage = ±15V, V_{CM} = 0, $R_L \geq 100\ k\Omega$ and $R_S = 50\Omega$ unless otherwise noted. **Boldface** limits apply for $T_J = T_{MIN}$ to T_{MAX}; all other limits $T_J = 25°C$.

Symbol	Parameter	Conditions	Typ	LM6161 Limit (Notes 3, 11)	LM6261 Limit (Note 3)	LM6361 Limit (Note 3)	Units
V_O	Output Voltage Swing	Supply = ±15V and R_L = 2 kΩ	+14.2	+13.5	+13.5	+13.4	Volts
				+13.3	**+13.3**	**+13.3**	Min
			−13.4	−13.0	−13.0	−12.9	Volts
				−12.7	**−12.8**	**−12.8**	Min
		Supply = +5V and R_L = 2 kΩ (Note 4)	4.2	3.5	3.5	3.4	Volts
				3.3	**3.3**	**3.3**	Min
			1.3	1.7	1.7	1.8	Volts
				2.0	**1.9**	**1.9**	Max
	Output Short Circuit Current	Source	65	30	30	30	mA
				20	**25**	**25**	Min
		Sink	65	30	30	30	mA
				20	**25**	**25**	Min
I_S	Supply Current		5.0	6.5	6.5	6.8	mA
				6.8	**6.7**	**6.9**	Max

AC Electrical Characteristics

The following specifications apply for Supply Voltage = ±15V, V_{CM} = 0, $R_L \geq 100\ k\Omega$ and $R_S = 50\Omega$ unless otherwise noted. **Boldface** limits apply for $T_J = T_{MIN}$ to T_{MAX}; all other limits $T_J = 25°C$.

Symbol	Parameter	Conditions	Typ	LM6161 Limit (Notes 3, 11)	LM6261 Limit (Note 3)	LM6361 Limit (Note 3)	Units
GBW	Gain-Bandwidth Product	f = 20 MHz	50	40	40	35	MHz
				30	**35**	**32**	Min
		Supply = ±5V	35				MHz
SR	Slew Rate	A_V = +1 (Note 8)	300	200	200	200	V/µs
				180	**180**	**180**	Min
		Supply = ±5V (Note 8)	200				V/µs
PBW	Power Bandwidth	$V_{OUT} = 20\ V_{PP}$	4.5				MHz
t_S	Settling Time	10V Step to 0.1% A_V = −1, R_L = 2 kΩ	120				ns
ϕm	Phase Margin		45				Deg
A_D	Differential Gain	NTSC, A_V = +4	<0.1				%
ϕD	Differential Phase	NTSC, A_V = +4	0.1				Deg
e_{np-p}	Input Noise Voltage	f = 10 kHz	15				$nV/\sqrt{Hz}$
i_{np-p}	Input Noise Current	f = 10 kHz	1.5				$pA/\sqrt{Hz}$

Note 1: Continuous short-circuit operation at elevated ambient temperature can result in exceeding the maximum allowed junction temperature of 150°C.

Note 2: The typical junction-to-ambient thermal resistance of the molded plastic DIP (N) is 105°C/W, the molded plastic SO (M) package is 155°C/W, and the cerdip (J) package is 125°C/W. All numbers apply for packages soldered directly into a printed circuit board.

Note 3: Limits are guaranteed by testing or correlation.

Note 4: For single supply operation, the following conditions apply: V^+ = 5V, V^- = 0V, V_{CM} = 2.5V, V_{OUT} = 2.5V. Pin 1 & Pin 8 (Vos Adjust) are each connected to Pin 4 (V^-) to realize maximum output swing. This connection will degrade V_{OS}, V_{OS} Drift, and Input Voltage Noise.

Note 5: $C_L \leq 5$ pF.

Note 6: In order to achieve optimum AC performance, the input stage was designed without protective clamps. Exceeding the maximum differential input voltage results in reverse breakdown of the base-emitter junction of one of the input transistors and probable degradation of the input parameters (especially Vos, Ios, and Noise).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Continued)

Note 7: The average voltage that the weakest pin combinations (those involving Pin 2 or Pin 3) can withstand and still conform to the datasheet limits. The test circuit used consists of the human body model of 100 pF in series with 1500Ω.

Note 8: V_{IN} = 8V step. For supply = ±5V, V_{IN} = 5V step.

Note 9: Voltage Gain is the total output swing (20V) divided by the input signal required to produce that swing.

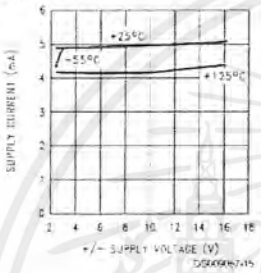
Note 10: The voltage between V^+ and either input pin must not exceed 36V.

Note 11: A military RETS electrical test specification is available on request. At the time of printing, the RETS6161X specs complied with all **Boldface** limits in this column.

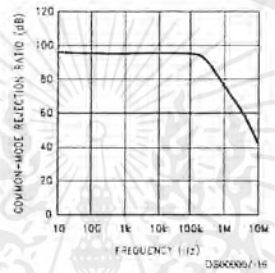
Note 12: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed.

Typical Performance Characteristics (R_L = 10 kΩ, T_A = 25˚C unless otherwise specified)

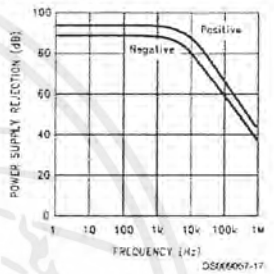
Supply Current vs Supply Voltage



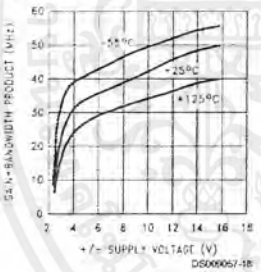
Common-Mode Rejection Ratio



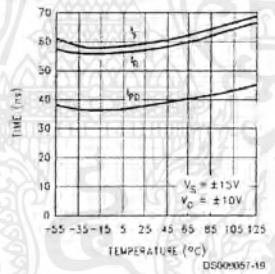
Power Supply Rejection Ratio



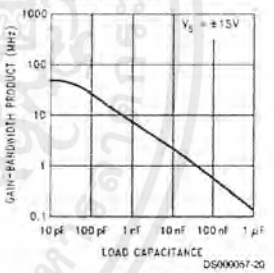
Gain-Bandwidth Product



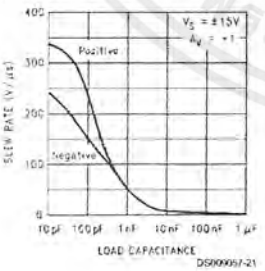
Propagation Delay Rise and Fall Times



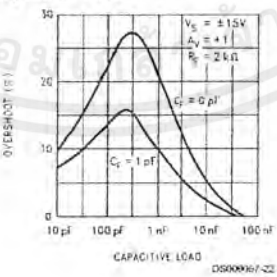
Gain-Bandwidth Product vs Load Capacitance



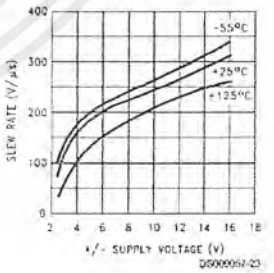
Slew Rate vs Load Capacitance



Overshoot vs Capacitive Load



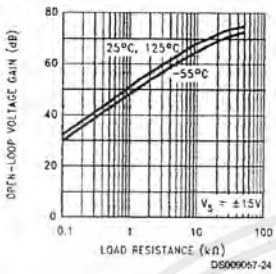
Slew Rate



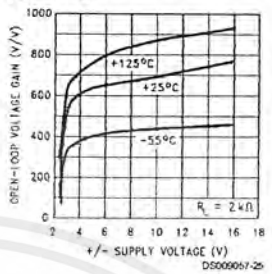
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (R_L = 10 kΩ, T_A = 25°C unless otherwise specified) (Continued)

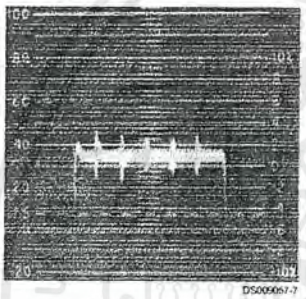
Voltage Gain vs Load Resistance



Gain vs Supply Voltage



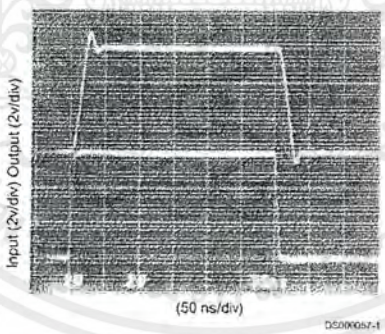
Differential Gain (Note 13)



Differential Phase (Note 13)

Note 13: Differential gain and differential phase measured for four series LM6361 op amps configured as unity-gain followers, in series with an LM6321 buffer. Error added by LM6321 is negligible. Test performed using Tektronix Type 520 NTSC test system.

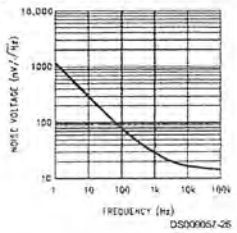
Step Response; Av = +1



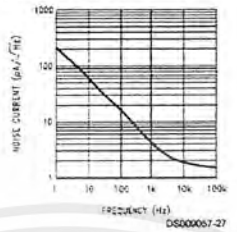
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (R_L = 10 kΩ, T_A = 25˚C unless otherwise specified) (Continued)

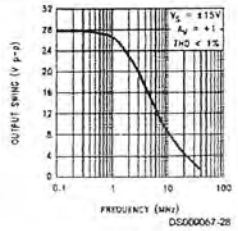
Input Noise Voltage



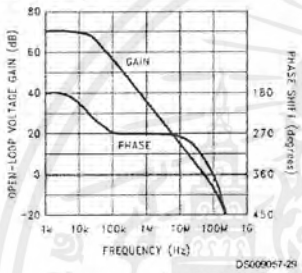
Input Noise Current



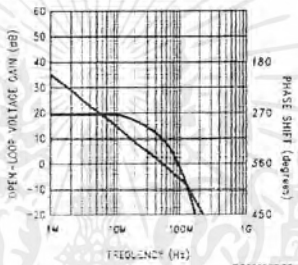
Power Bandwidth



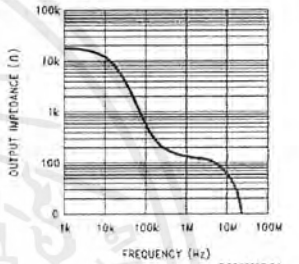
Open-Loop Frequency Response



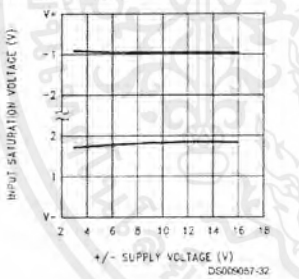
Open-Loop Frequency Response



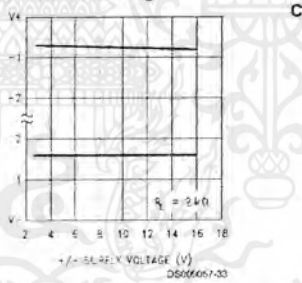
Output Impedence (Open-Loop)



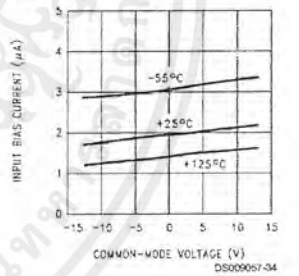
Common-Mode Input Saturation Voltage



Output Saturation Voltage

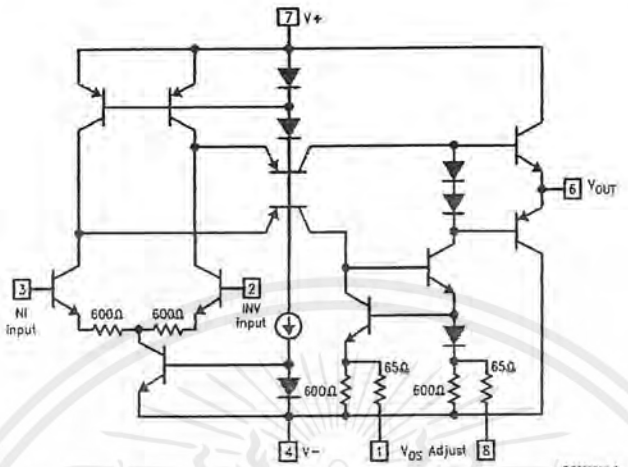


Bias Current vs Common-Mode Voltage



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Simplified Schematic



Applications Tips

The LM6361 has been compensated for unity-gain operation. Since this compensation involved adding emitter-degeneration resistors to the op amp's input stage, the open-loop gain was reduced as the stability increased. Gain error due to reduced A_{VOL} is most apparent at high gains; thus, for gains between 5 and 25, the less-compensated LM6364 should be used, and the uncompensated LM6365 is appropriate for gains of 25 or more. The LM6361, LM6364, and LM6365 have the same high slew rate, regardless of their compensation.

The LM6361 is unusually tolerant of capacitive loads. Most op amps tend to oscillate when their load capacitance is greater than about 200 pF (especially in low-gain circuits). The LM6361's compensation is effectively increased with load capacitance, reducing its bandwidth and increasing its stability.

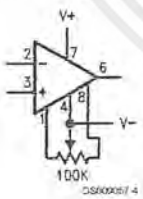
Power supply bypassing is not as critical for the LM6361 as it is for other op amps in its speed class. Bypassing will, however, improve the stability and transient response and is recommended for every design. 0.01 µF to 0.1 µF ceramic capacitors should be used (from each supply "rail" to ground). If the device is far away from its power supply source, an additional 2.2 µF to 10 µF of tantalum may provide extra noise reduction.

Keep all leads short to reduce stray capacitance and lead inductance, and make sure ground paths are low-impedance, especially where heavier currents will be flowing. Stray capacitance in the circuit layout can cause signal coupling across adjacent nodes and can cause gain to unintentionally vary with frequency.

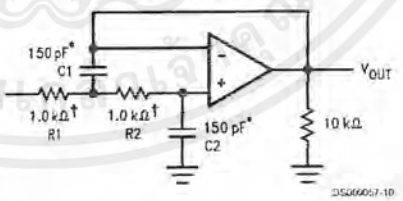
Breadboarded circuits will work best if they are built using generic PC boards with a good ground plane. If the op amps are used with sockets, as opposed to being soldered into the circuit, the additional input capacitance may degrade circuit performance.

Typical Applications

Offset Voltage Adjustment



1 MHz Low-Pass Filter



†1% tolerance

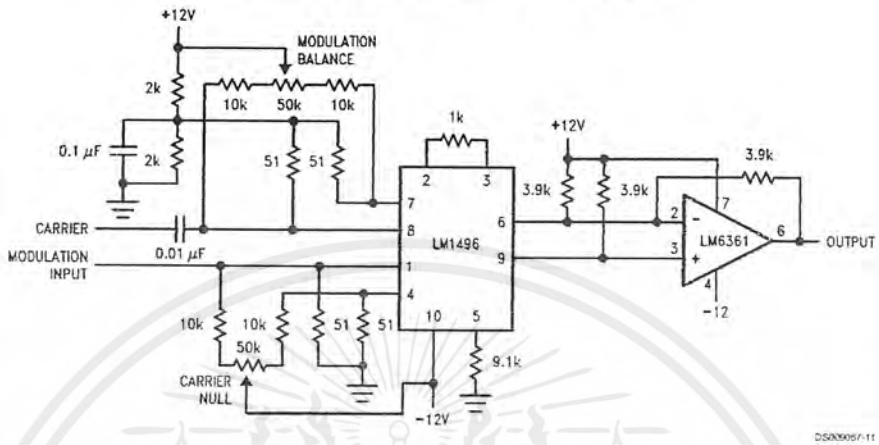
*Matching determines filter precision

$f_c = (2\pi \sqrt{(R1\ R2\ C1\ C2)})^{-1}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

Modulator with Differential-to-Single-Ended Converter



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

TDA8703
8-bit high-speed analog-to-digital converter

Product specification
Supersedes data of April 1993
File under Integrated Circuits, IC02

1996 Aug 26

Philips
Semiconductors

PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-bit high-speed analog-to-digital converter

TDA8703

FEATURES

- 8-bit resolution
- Sampling rate up to 40 MHz
- High signal-to-noise ratio over a large analog input frequency range (7.1 effective bits at 4.43 MHz full-scale input)
- Binary or two's complement 3-state TTL outputs
- Overflow/underflow 3-state TTL output
- TTL compatible digital inputs
- Low-level AC clock input signal allowed
- Internal reference voltage generator
- Power dissipation only 290 mW (typical)
- Low analog input capacitance, no buffer amplifier required
- No sample-and-hold circuit required.

APPLICATIONS

- General purpose high-speed analog-to-digital conversion
- Digital TV, IDTV
- Subscriber TV decoder
- Satellite TV decoders
- Digital VCR.

GENERAL DESCRIPTION

The TDA8703 is an 8-bit high-speed Analog-to-Digital Converter (ADC) for video and other applications. It converts the analog input signal into 8-bit binary-coded digital words at a maximum sampling rate of 40 MHz. All digital inputs and outputs are TTL compatible, although a low-level AC clock input signal is allowed.

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
TDA8703	DIP24	plastic dual in-line package; 24 leads (600 mil)	SOT101-1
TDA8703T	SO24	plastic small outline package; 24 leads; body width 7.5 mm	SOT137-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{CCA}	analog supply voltage		4.5	5.0	5.5	V
V_{CCD}	digital supply voltage		4.5	5.0	5.5	V
V_{CCO}	output stages supply voltage		4.2	5.0	5.5	V
I_{CCA}	analog supply current		–	28	36	mA
I_{CCD}	digital supply current		–	19	25	mA
I_{CCO}	output stages supply current		–	11	14	mA
ILE	DC integral linearity error		–	–	±1	LSB
DLE	DC differential linearity error		–	–	±1/2	LSB
AILE	AC integral linearity error	note 1	–	–	±2	LSB
B	–3 dB bandwidth	note 2; f_{CLK} = 40 MHz	–	19.5	–	MHz
$f_{CLK}/f_{\overline{CLK}}$	maximum conversion rate	note 3	40	–	–	MHz
P_{tot}	total power dissipation		–	290	415	mW

Notes

1. Full-scale sinewave (f_i = 4.4 MHz; f_{CLK}; $f_{\overline{CLK}}$ = 27 MHz).
2. The –3 dB bandwidth is determined by the 3 dB reduction in the reconstructed output (full-scale signal at input).
3. The circuit has two clock inputs CLK and $\overline{CLK}$. There are four modes of operation:
   a) TTL (mode 1); $\overline{CLK}$ decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
   b) TTL (mode 2); CLK decoupled to DGND by a capacitor. $\overline{CLK}$ input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
   c) AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the $\overline{CLK}$ input with such a signal, sampling takes place on the HIGH-to-LOW transition.
   d) If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM

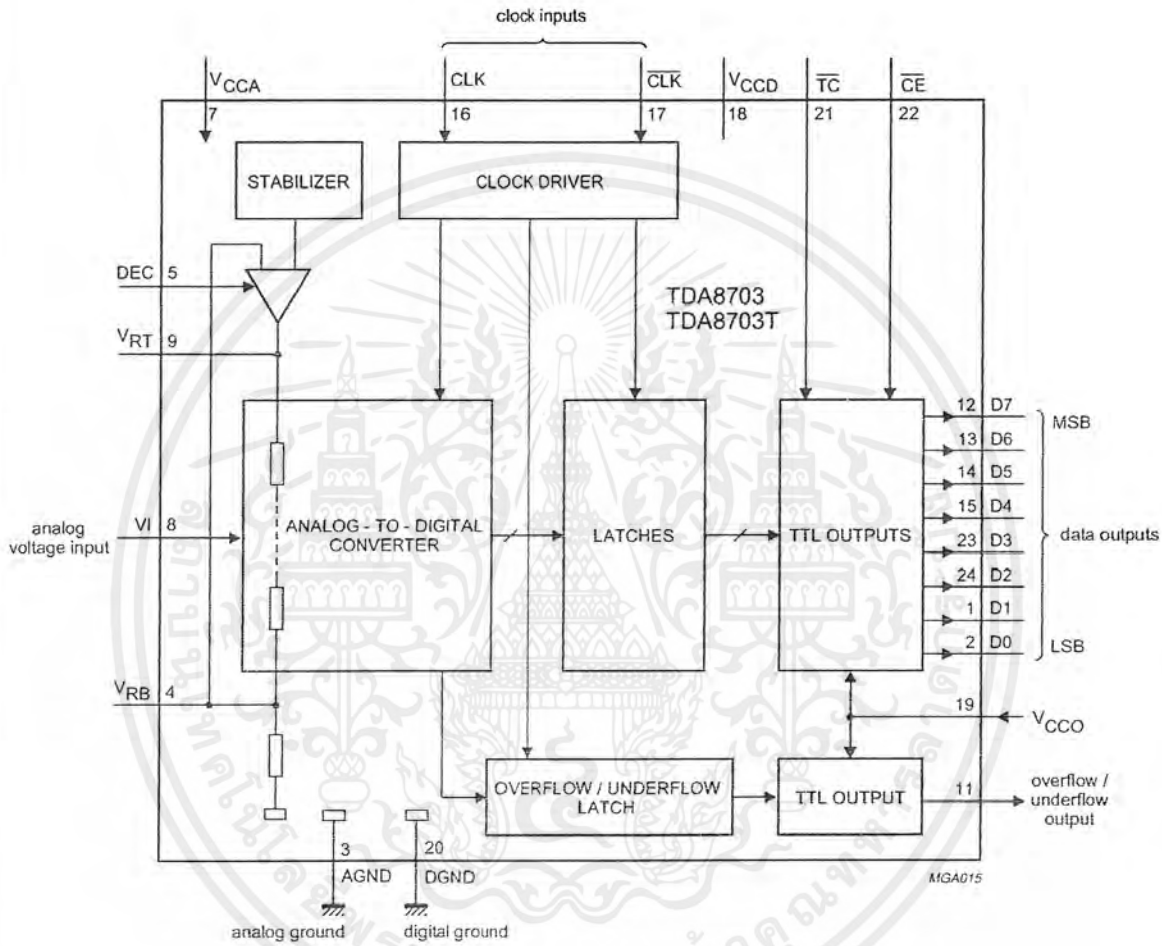


Fig.1 Block diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PINNING

SYMBOL	PIN	DESCRIPTION
D1	1	data output; bit 1
D0	2	data output; bit 0 (LSB)
AGND	3	analog ground
V_{RB}	4	reference voltage bottom (decoupling)
DEC	5	decoupling input (internal stabilization loop decoupling)
n.c.	6	not connected
V_{CCA}	7	positive supply voltage for analog circuits (+5 V)
VI	8	analog voltage input
V_{RT}	9	reference voltage top (decoupling)
n.c.	10	not connected
O/UF	11	overflow/underflow data output
D7	12	data output; bit 7 (MSB)
D6	13	data output; bit 6
D5	14	data output; bit 5
D4	15	data output; bit 4
CLK	16	clock input
$\overline{CLK}$	17	complementary clock input
V_{CCD}	18	positive supply voltage for digital circuits (+5 V)
V_{CCO}	19	positive supply voltage for output stages (+5 V)
DGND	20	digital ground
$\overline{TC}$	21	input for two's complement output (TTL level input, active LOW)
$\overline{CE}$	22	chip enable input (TTL level input, active LOW)
D3	23	data output; bit 3
D2	24	data output; bit 2

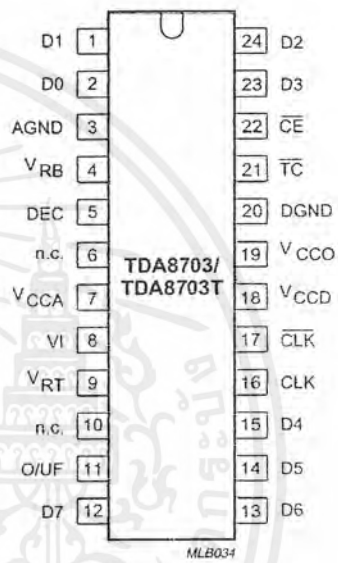


Fig.2 Pin configuration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CCA}	analog supply voltage		−0.3	+7.0	V
V_{CCD}	digital supply voltage		−0.3	+7.0	V
V_{CCO}	output stages supply voltage		−0.3	+7.0	V
$V_{CCA} - V_{CCD}$	supply voltage differences		−1.0	+1.0	V
$V_{CCO} - V_{CCD}$	supply voltage differences		−1.0	+1.0	V
$V_{CCA} - V_{CCO}$	supply voltage differences		−1.0	+1.0	V
V_{VI}	input voltage range	referenced to AGND	−0.3	+7.0	V
$V_{CLK}/V_{\overline{CLK}}$	AC input voltage for switching (peak-to-peak value)	note 1; referenced to DGND	–	2.0	V
I_O	output current		–	+10	mA
T_{stg}	storage temperature		−55	+150	°C
T_{amb}	operating ambient temperature		0	+70	°C
T_j	junction temperature		–	+125	°C

Notes

1. The circuit has two clock inputs CLK and $\overline{CLK}$. There are four modes of operation:
   a) TTL (mode 1); $\overline{CLK}$ decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
   b) TTL (mode 2); CLK decoupled to DGND by a capacitor. $\overline{CLK}$ input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
   c) AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the $\overline{CLK}$ input with such a signal, sampling takes place on the HIGH-to-LOW transition.
   d) If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.

HANDLING

Inputs and outputs are protected against electrostatic discharges in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling integrated circuits.

THERMAL RESISTANCE

SYMBOL	PARAMETER	VALUE	UNIT
$R_{th\ j\text{-}a}$	from junction to ambient in free air		
	SOT101-1	55	K/W
	SOT137-1	75	K/W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHARACTERISTICS

$V_{CCA} = V_7 - V_3 = 4.5$ V to 5.5 V; $V_{CCD} = V_{18} - V_{20} = 4.5$ V to 5.5 V; $V_{CCO} = V_{19} - V_{20} = 4.5$ V to 5.5 V; AGND and DGND shorted together; $V_{CCA} - V_{CCD} = -0.5$ V to +0.5 V; $V_{CCO} - V_{CCD} = -0.5$ V to +0.5 V; $V_{CCA} - V_{CCD} = -0.5$ V to +0.5 V; $T_{amb} = 0$ °C to +70 °C; unless otherwise specified (typical values measured at $V_{CCA} = V_{CCD} = V_{CCO} = 5$ V and $T_{amb} = 25$ °C).

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{CCA}	analog supply voltage		4.5	5.0	5.5	V
V_{CCD}	digital supply voltage		4.5	5.0	5.5	V
V_{CCO}	output stages supply voltage		4.2	5.0	5.5	V
I_{CCA}	analog supply current		–	28	36	mA
I_{CCD}	digital supply current		–	19	25	mA
I_{CCO}	output stage supply current	all outputs LOW	–	11	14	mA
Inputs						
CLOCK INPUT $\overline{CLK}$ AND CLK (note 1; REFERENCED TO DGND)						
V_{IL}	LOW level input voltage		0	–	0.8	V
V_{IH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{IL}	LOW level input current	$V_{CLK}/V_{\overline{CLK}} = 0.4$ V	−400	–	–	µA
I_{IH}	HIGH level input current	$V_{CLK}/V_{\overline{CLK}} = 0.4$ V	–	–	100	µA
		$V_{CLK}/V_{\overline{CLK}} = V_{CCD}$	–	–	300	µA
Z_i	input impedance	$f_{CLK}/f_{\overline{CLK}} = 10$ MHz	–	4	–	kΩ
C_i	input capacitance	$f_{CLK}/f_{\overline{CLK}} = 10$ MHz	–	4.5	–	pF
$V_{CLK} - V_{\overline{CLK}}$	AC input voltage for switching (peak-to-peak value)	note 1; DC level = 1.5 V	0.5	–	2.0	V
$\overline{TC}$ AND $\overline{CE}$ (REFERENCED TO DGND)						
V_{IL}	LOW level input voltage		0	–	0.8	V
V_{IH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{IL}	LOW level input current	$V_{IL} = 0.4$ V	−400	–	–	µA
I_{IH}	HIGH level input current	$V_{IH} = 2.7$ V	–	–	20	µA
VI (ANALOG INPUT VOLTAGE REFERENCED TO AGND)						
$V_{VI(B)}$	input voltage (bottom)		1.33	1.41	1.48	V
$V_{VI(0)}$	input voltage	output code = 0	1.455	1.55	1.635	V
$V_{OS(B)}$	offset voltage (bottom)	$V_{VI(0)} - V_{VI(B)}$	0.125	–	0.155	V
$V_{VI(T)}$	input voltage (top)		3.2	3.36	3.5	V
$V_{VI(255)}$	input voltage	output code = 255	3.115	3.26	3.385	V
$V_{OS(T)}$	offset voltage (top)	$V_{VI(T)} - V_{VI(255)}$	0.085	–	0.115	V
$V_{VI(p-p)}$	input voltage amplitude (peak-to-peak value)		1.66	1.71	1.75	V
I_{IL}	LOW level input current	$V_{VI} = 1.4$ V	–	0	–	µA
I_{IH}	HIGH level input current	$V_{VI} = 3.6$ V	60	120	180	µA
Z_i	input impedance	$f_i = 1$ MHz	–	10	–	kΩ
C_i	input capacitance	$f_i = 1$ MHz	–	14	–	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Reference resistance						
R_{ref}	reference resistance	V_{RT} to V_{RB}	–	220	–	Ω
Outputs						
DIGITAL OUTPUTS (D7 - D0) (REFERENCED TO DGND)						
V_{OL}	LOW level output voltage	I_O = 1 mA	0	–	0.4	V
V_{OH}	HIGH level output voltage	I_O = –0.4 mA	2.7	–	V_{CCD}	V
I_{OZ}	output current in 3-state mode	0.4 V < V_O < V_{CCD}	–20	–	+20	µA
Switching characteristics (note 2; see Fig.3)						
$f_{CLK}/f_{\overline{CLK}}$	maximum clock frequency		40	–	–	MHz
Analog signal processing (f_{CLK} = 40 MHz)						
B	–3 dB bandwidth	note 3	–	19.5	–	MHz
G_d	differential gain	note 4	–	0.6	–	%
ϕ_d	differential phase	note 4	–	0.8	–	deg
f_1	fundamental harmonics (full-scale)	f_i = 4.43 MHz	–	–	0	dB
f_{all}	harmonics (full-scale), all components	f_i = 4.43 MHz	–	–55	–	dB
SVRR1	supply voltage ripple rejection	note 5	–	–28	–25	dB
SVRR2	supply voltage ripple rejection	note 5	–	1	2.5	%/V
Transfer function						
ILE	DC integral linearity error		–	–	±1	LSB
DLE	DC differential linearity error		–	–	±1/2	LSB
AILE	AC integral linearity error	note 6	–	–	±2	LSB
EB	effective bits	f_i = 4.43 MHz	–	7.1	–	bits
Timing (note 7; see Figs 3 to 6; **f_{CLK} = 40 MHz)**						
t_{dS}	sampling delay		–	–	2	ns
t_{HD}	output hold time		6	–	–	ns
t_{dLH}	output delay time	LOW-to-HIGH transition	–	8	10	ns
t_{dHL}	output delay time	HIGH-to-LOW transition	–	16	20	ns
t_{dZH}	3-state output delay times	enable-to-HIGH	–	19	25	ns
t_{dZL}	3-state output delay times	enable-to-LOW	–	16	20	ns
t_{dHZ}	3-state output delay times	disable-to-HIGH	–	14	20	ns
t_{dLZ}	3-state output delay times	disable-to-LOW	–	9	12	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Notes

1. The circuit has two clock inputs CLK and $\overline{CLK}$. There are four modes of operation:
   a) TTL (mode 1); $\overline{CLK}$ decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
   b) TTL (mode 2); CLK decoupled to DGND by a capacitor. $\overline{CLK}$ input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
   c) AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the $\overline{CLK}$ input with such a signal, sampling takes place on the HIGH-to-LOW transition.
   d) If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.
2. In addition to a good layout of the digital and analog ground, it is recommended that the rise and fall times of the clock must not be less than 2 ns.
3. The –3 dB bandwidth is determined by the 3 dB reduction in the reconstructed output (full-scale signal at the input).
4. Low frequency ramp signal ($V_{VI(p-p)}$ = 1.8 V and f_i = 15 kHz) combined with a sinewave input voltage ($V_{VI(p-p)}$ = 0.5 V, f_i = 4.43 MHz) at the input.
5. Supply voltage ripple rejection:
   a) SVRR1; variation of the input voltage producing output code 127 for supply voltage variation of 1 V: $SVRR1 = 20 \log (\Delta V_{VI(127)} / \Delta V_{CCA})$
   b) SVRR2; relative variation of the full-scale range of analog input for a supply voltage variation of 1 V: $SVR2 = \{\Delta(V_{VI(0)} - V_{VI(255)}) / (V_{VI(0)} - V_{VI(255)})\} \div \Delta V_{CCA}$.
6. Full-scale sinewave (f_i = 4.4 MHz; f_{CLK}; $f_{\overline{CLK}}$ = 27 MHz).
7. Output data acquisition:
   a) Output data is available after the maximum delay of t_{dHL} and t_{dLH}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1 Output coding and input voltage (referenced to AGND; typical values)

			BINARY OUTPUT BITS								TWO'S COMPLEMENT OUTPUT BITS							
STEP	$V_{VI(p-p)}$	O/UF	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Underflow	<1.55	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1.55	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	–	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
254	.	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0
255	3.26	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Overflow	>3.26	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Table 2 Mode selection

$\overline{TC}$	$\overline{CE}$	D7-D0	O/UF
X(1)	1	high impedance	high impedance
0	0	active; two's complement	active
1	0	active; binary	active

Note

1. X = don't care.

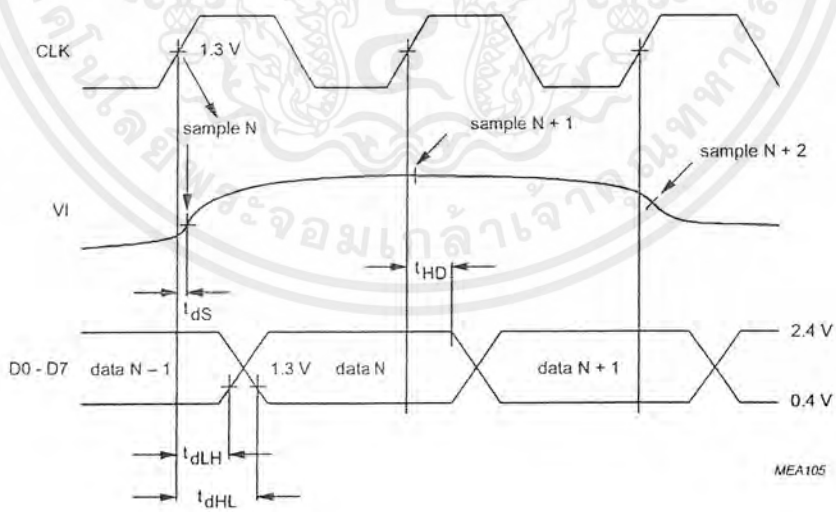


Fig.3 Timing diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

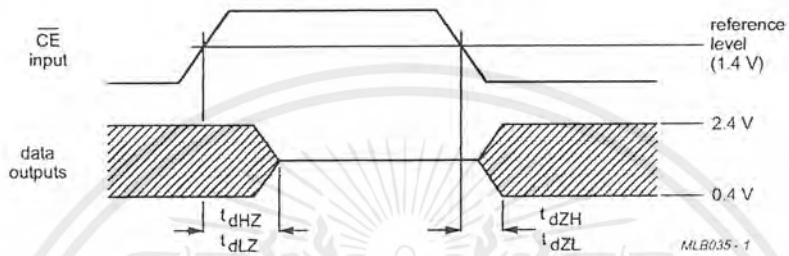


Fig.4 3-state delay timing diagram.

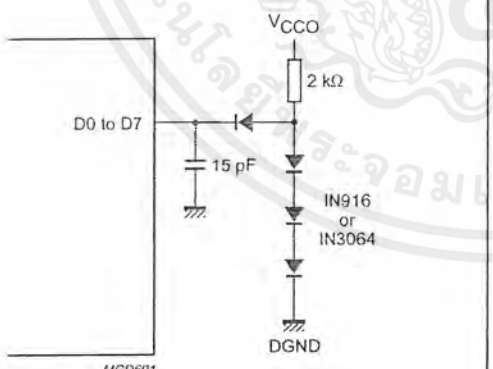


Fig.5 Load circuit for timing measurement; data outputs ($\overline{CE}$ = LOW).

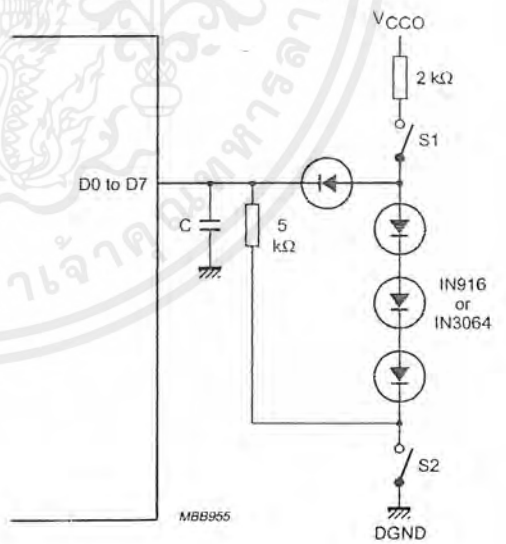


Fig.6 Load circuit for timing measurement; 3-state outputs ($\overline{CE}$: f_i = 1 MHz; V_{VI} = 3 V); see Table 3.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3 Mode selection

TIMING MEASUREMENT	SWITCH S1	SWITCH S2	CAPACITOR
t_{dZH}	open	closed	15 pF
t_{dZL}	closed	open	15 pF
t_{dHZ}	closed	closed	5 pF
t_{dLZ}	closed	closed	5 pF

INTERNAL PIN CONFIGURATIONS

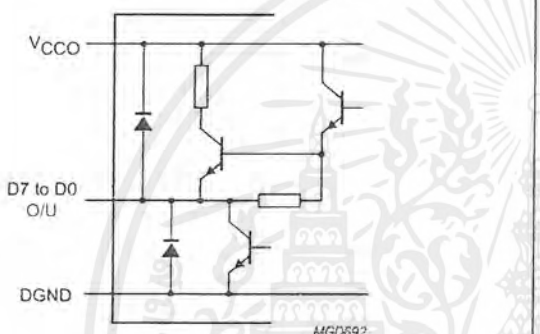


Fig.7 TTL data and overflow/underflow outputs.

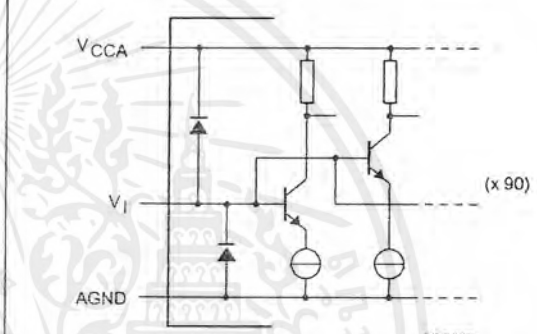


Fig.8 Analog inputs.

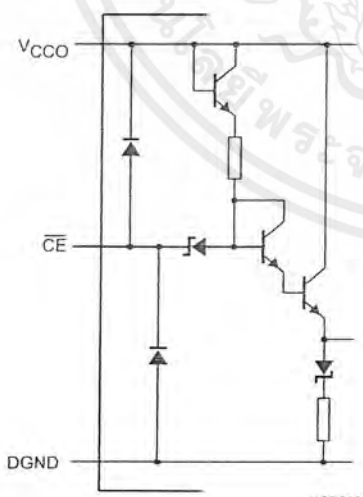


Fig.9 $\overline{CE}$ (3-state) input.

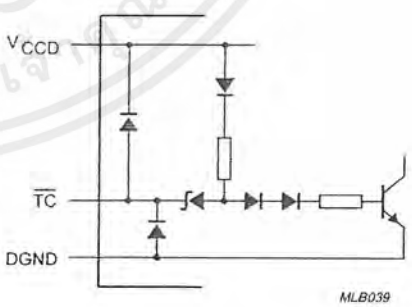


Fig.10 $\overline{TC}$ (two's complement) input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

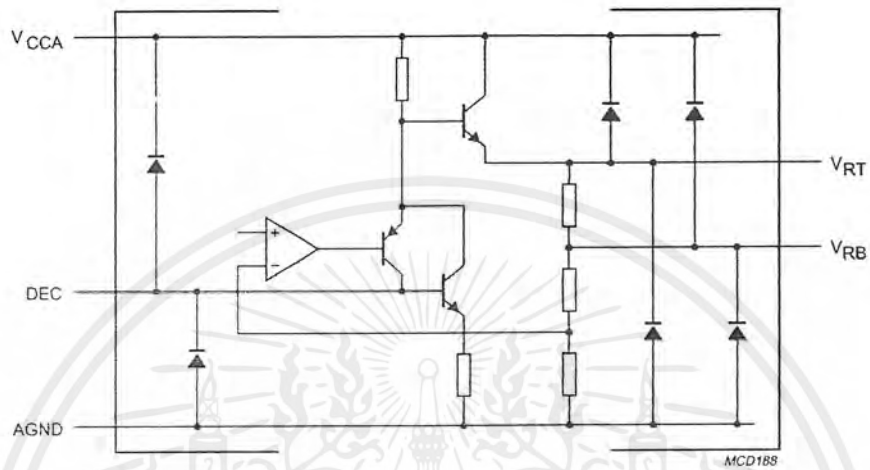


Fig.11 V_{RB}, V_{RT} and DEC.

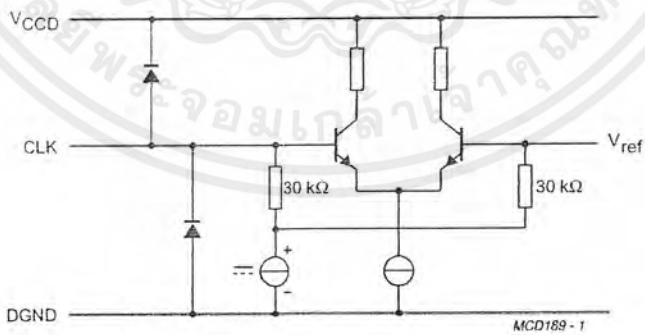
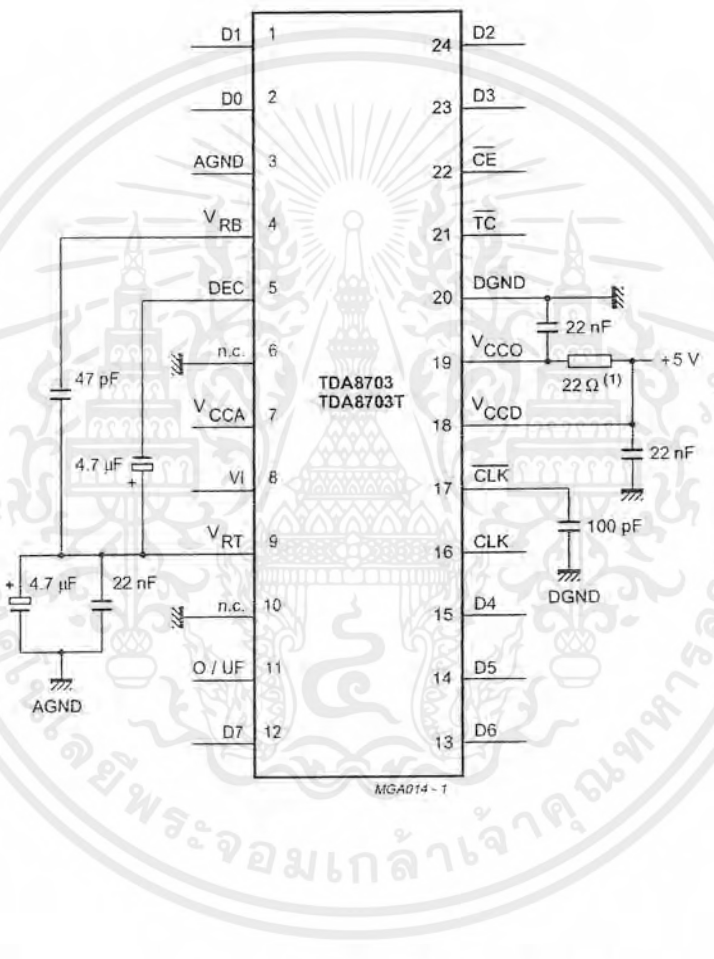


Fig.12 CLK and $\overline{CLK}$ inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

Additional application information will be supplied upon request (please quote number FTV/8901).



CLK should be decoupled to the DGND with a 100 nF capacitor, if a TTL signal is used on CLK (see Chapter "Characteristics", note 1).

CLK and $\overline{CLK}$ can be used in a differential mode (see Chapter "Characteristics", note 1).

V_{RB} and V_{RT} are decoupling pins for the internal reference ladder, do not draw current from these pins in order to achieve good linearity.

If it is required to use the TDA8703 in a parallel system configuration, the references (V_{RB} and V_{RT}) of each TDA8703 can be connected together. Code 0 will be identical and code 255 will remain in the 1 LSB variation for each TDA8703.

Analog and digital supplies should be separated and decoupled.

Pins 6 and 10 should be connected to AGND in order to prevent noise influence.

(1) It is recommended to decouple V_{CCO} through a 22 Ω resistor especially when the output data of the TDA8703 interfaces with a capacitive CMOS load device.

Fig.13 Application diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานชิ้นนี้จะไม่มีทางสำเร็จลุล่วงขึ้นมาได้เลย หากขาดบุคคลเหล่านี้ อาจารย์โกศล ชวนขยัน ซึ่งเป็นผู้ให้คำปรึกษาแนวทางในการทำงาน ช่วยแนะแนวทางในการแก้ปัญหาและอุปสรรคต่าง ๆ ที่เกิดขึ้น ทำให้เข้าใจสิ่งทีเกิดขึ้นต่าง ๆ ได้ ทั้งนี้ต้องขอขอบคุณบิดา มารดา ผู้ให้กำเนิด ที่เข้าใจและเป็นกำลังใจให้อยู่เสมอ และคอยอุปการะทางด้านการเงิน ค่าการใช้จ่ายในการทำโครงงานชิ้นนี้ รวมถึงขอขอบคุณพี่ ๆ เพื่อน ๆ และบุคคลที่มีส่วนเกี่ยวข้องทุกคน ที่คอยเกื้อหนุนต่าง ๆ ตลอดมา

(นาย พิศิษฐ์ รัตตไพศาล)

(นาย เมธา เตชะทรงคุณ)

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ยืน ภู่วรรณ , "เทคโนโลยีฮาร์ดแวร์ IBM PC" , ซีเอ็ดยูเคชัน , 2537.
2. ธานินทร์ ถาวรศาสนวงศ์ , ทินกร ดุ๊ก , "การอินเตอร์เฟส IBM/PC" , สำนักพิมพ์ฟิสิกส์เซ็นเตอร์ , 2535.
3. พ.อ. เจนวิทย์ เหลืองอร่าม , "การใช้ Turbo C++ เขียนโปรแกรมภาษาซี" , ซีเอ็ดยูเคชัน , 2538.
4. Jack Purdum , "C Programmer 's Toolkit" , 2nd ed. , Que , 1992.
5. Chris H.Pappas , William H.Murray , "Turbo C++ professional Handbook" , McGraw - Hill , 1990.
6. Sergio Franco , "Design with Operational Amplifiers and Analog Integrated Circuits" , 2nd ed. , McGraw-Hill , 1998.
7. William D. Stanley , "Operational Amplifiers with Linear Integrated Circuits" , 3rd ed. , Chap 5 , Merrill Pub. , 1994.
8. Robert Boylestad , Louis Nashelsky , "Electronic Devices and Circuit Theory" , 6th ed. , Prentice-Hall , 1996.
9. Albert Paul Malvino , "Electronic Principles" , 6th ed. , McGraw-Hill , 1999.
10. James E. Buchanan , "CMOS / TTL Digital System Design" , McGraw-Hill , 1990.
11. Stuart M. Asser , Vincent J. Stigliano , Richard F. Bahrenburg , "Microcomputer Theory And - Servicing" , 2nd ed. , Merrill Pub.
12. Jim Aspinwall , Rory Burke , Mike Todd , "Troubleshooting Your PC" , 2nd ed. , MIS : Press Pub. , 1994.
13. Hans-Peter Messmer , "The Indispensable PC Hardware Book" , Addison-Wesley , 1994.
14. Muhammad Ali Mazidi , Janice Gillispie Mazidi , "The 80x86 IBM PC&Compatible computers , Assembly language , design and interfacing , Vols I&II" , Prentice-Hall , 1995.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้