ระบบรักษาความปลอดภัยผ่านสายโทรศัพท์

SECURITY SYSTEM VIA TELEPHONE LINE

โดย

นายประดิษฐ์ เกตุแก้ว

นายปิติ อู่เจริญ

เลขเรียกหนังสือ ปพ ปประ มาสฐ

เลขทะเบียน 041796

วัน เดือน ปี 21 ม.ค. 43

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 041796

ระบบรักษาความปลอดภัยผ่านสายโทรศัพท์

SECURITY SYSTEM VIA TELEPHONE LINE

โดย

นาย ประดิษฐ์ เกตุแก้ว 40013059

นาย ปิติ อู่เจริญ 40013061

อาจารย์ที่ปรึกษา

ดร.สุทธิชัย นพนาคีพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยผ่านสายโทรศัพท์

SECURITY SYSTEM VIA TELEPHONE LINE

โดย นายประดิษฐ์ เกตุแก้ว 40013059
นายปิติ อู่เจริญ 40013061

อาจารย์ที่ปรึกษา ดร.สุทธิชัย นพนาคีพงษ์

บทคัดย่อ

ระบบรักษาความปลอดภัยเป็นสิ่งจำเป็นอย่างยิ่งในปัจจุบัน โดยเฉพาะในอาคารสำนักงาน ร้านค้า หรือแม้แต่ที่พักอาศัย กล้องวงจรปิดจึงเป็นอุปกรณ์ที่นิยมกันมาก ในขณะเดียวกันนั้นระบบสื่อสารข้อมูลผ่านเครือข่ายสายโทรศัพท์ก็มีการพัฒนาก้าวไกลขึ้น สามารถส่งข้อมูลข่าสารได้หลายรูปแบบรวมทั้งสัญญาณภาพ ในโครงงานนี้จึงได้นำเอาระบบการสื่อสารผ่านสายโทรศัพท์นี้มาใช้กับระบบรักษาความปลอดภัย โดยจะเป็นลักษณะของการเรียกดูสัญญาณภาพจากกล้องวีดีโอผ่านสายโทรศัพท์ โดยใช้อุปกรณ์ I/O ในการเชื่อมต่อระบบสื่อสารข้อมูล

ABSTRACT

Security system is currently necessary , especially in the office building , super market or residence . Thus a closed circuit camera has became a popular equipment . As communication via telephone line has been improving to transmit a various information form including video signal , so in this project has applied a communication via telephone line to use with security system. It will can transmit a video signal from a camera via telephone line and then display on monitor by using I/O instrument to connect communicating system .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบรักษาความปลอดภัยผ่านสายโทรศัพท์**

SECURITY SYSTEM VIA TELEPHONE LINE

ผู้จัดทำ

1. นายประดิษฐ์ เกตุแก้ว 40013059

2. นายปิติ อู่เจริญ 40013061

_____อาจารย์ที่ปรึกษา

(ดร.สุทธิชัย นพนาคีพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทฤษฎีโทรทัศน์	2
2.1.1 องค์ประกอบภาพ	2
2.1.2 การสแกน	3
2.2 โครงสร้างของ MCS-51	11
2.2.1 คุณสมบัติของไมโครคอลโทรลเลอร์ตระกูล MCS-51	11
2.2.2 โครงสร้างภายในของ 8051	11
2.2.3 พอร์ตของ 8051	13
2.2.4 การแบ่งประเภทของหน่วยความจำ	16
2.3 การแปลงสัญญาณอนาลอกเป็นดิจิตอล	20
2.4 พอร์ตขนาน	24
2.5 RAS (Remote Access Service)	26
2.5.1 การติดตั้ง Remote Access Service	27
2.5.2 การติดตั้งไคลเอนส์ RAS สำหรับ Windows 95	36
2.5.3 การบริหารและควบคุมระบบ RAS	39
2.6 การเขียนโปรแกรมแบบวิชวล	43
บทที่ 3 การคำนวณและการสร้าง	51
3.1 แนวคิดและการทำงาน	51
3.2 หลักการทำงาน	52
3.3 วงจรกำเนิดความถี่	52
3.4 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล	52
3.5 วงจรแยกซิงค์ (SYNC SEPERATOR)	55
3.6 วงจรนับ (COUNTER)	57
3.7 วงจรเลือกแอดเดรสหน่วยความจำ	59
3.8 วงจรหน่วยความจำภาพ	61
3.9 วงจรควบคุมการเขียนข้อมูลภาพ	63
3.10 วงจรอินเตอร์เฟส	65
3.11 การเขียนโปรแกรมไมโครคอลโทรลเลอร์	67
3.12 การติดต่อส่วนรับส่งข้อมูล	67
3.13 การออกแบบโปรแกรม	68
บทที่ 4 การทดลองและผลการทดลอง	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การทดลองภาคแปลงสัญญาณอนาลอกเป็น
สัญญาณดิจิตอล(Analog To Digital Converter) 70
4.2 การทดลองภาคการแยกสัญญาณซิงค์ 73
4.3 การทดลองภาควงจรนับ(Counter) 77
4.4 การทดลองภาคแอดเดรส 79
4.5 การทดลองภาคหน่วยความจำ 80
4.6 การทดสอบโดยการเขียนและอ่านภาพจากหน่วยความจำและ
แสดงบนหน้าจอคอมพิวเตอร์ตัวเก็บข้อมูลภาพ 82
บทที่ 5 บทวิจารณ์และบทสรุป 86
ภาคผนวก 87
หนังสืออ้างอิง 136

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 องค์ประกอบภาพที่ยุคปัจจุบันเอาแต่พิกเซลไปใช้งาน	3
รูปที่ 2.2 แสดงวิธีการเบื้องต้นของการสแกน	3
รูปที่ 2.3 รายละเอียดการสแกนแบบสลับเส้นหรือแบบแทรกสอด	4
รูปที่ 2.4 การสแกนสลับเส้นหรือการแทรกสอดในระบบ CCIR	5
รูปที่ 2.5 แสดงระดับสัญญาณขาว ดำ โดยเกรย์สเกล	7
รูปที่ 2.6 แสดงลักษณะสัญญาณภาพรวมเบื้องต้น	8
รูปที่ 2.7 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง	10
รูปที่ 2.8 แสดงระบบพัลซ์ในฟิลด์ของการซิงโครไนซ์ของระบบโทรทัศน์	10
รูปที่ 2.9 a 8051 บล็อคไดอะแกรมของ MCS-51	11
รูปที่ 2.9 b ตำแหน่งของรีจีสเตอร์ต่าง ๆและหน่วยความจำภายใน	12
รูปที่ 2.10 การจัดวางขาของ 8051	13
รูปที่ 2.11 แสดงโครงสร้างของ พอร์ต 0 บิต	13
รูปที่ 2.12 แสดงโครงสร้างของ พอร์ต 1 บิต	14
รูปที่ 2.13 แสดงโครงสร้างของ พอร์ต 2 บิต	14
รูปที่ 2.14 แสดงโครงสร้างของ พอร์ต 3 บิต	15
รูปที่ 2.15 การต่อขารีเซ็ตให้กับ 8051	16
รูปที่ 2.16 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051	16
รูปที่ 2.17 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052	17
รูปที่ 2.18 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8051	17
รูปที่ 2.19 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8052	18
รูปที่ 2.20 128 ไบต์ของแรมที่เข้าถึงข้อมูลแบบตรงและแบบทางอ้อม	19
รูปที่ 2.21 แสดงการแซมปลิ้ง	21
รูปที่ 2.22 แสดงลักษณะสัญญาณของภาคควอนไตซิ่งและการเข้ารหัส	22
รูปที่ 2.23 แสดงวงจร Flash A/D	23
รูปที่ 2.24 แสดงหลักการของ Open Loop Convertion	23
รูปที่ 2.25 แสดงหน้าต่างเพื่อเข้าสู่การติดตั้ง Remote Access service	27
รูปที่ 2.26 แสดงช่องรายการ Network Service	28
รูปที่ 2.27 แสดง รายการ Remote Access Service ในช่องรายการ Network Service	28
รูปที่ 2.28 แสดงรายการอุปกรณ์สื่อสารที่ ใช้กับระบบ Remote Access Service	29
รูปที่ 2.29 แสดงหน้าต่างที่ใช้ปรับแต่งคุณสมบัติของอุปกรณ์สื่อสาร	29
รูปที่ 2.30 แสดงกรอบ Server Setting ที่ใช้ปรับแต่งโปรโตคอลของ RAS	30
รูปที่ 2.31 แสดงกรอบการปรับแต่งโปรโตคอล NetBUEI	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.32 แสดงกรอบการปรับแต่งโปรโตคอล TCP/IP 31
รูปที่ 2.33 แสดงกรอบการปรับแต่งโปรโตคอล IPX 32
รูปที่ 2.34 แสดงการเข้าสู่ Remote Access Admin 34
รูปที่ 2.35 แสดงกรอบ User Manager 34
รูปที่ 2.36 แสดงกรอบรายละเอียดของ User 35
รูปที่ 2.37 แสดงกรอบการกำหนด Permission ของ User 35
รูปที่ 2.38 แสดงกรอบการเริ่มติดตั้งโมเด็ม 36
รูปที่ 2.39 แสดงกรอบการสร้าง New Connection 37
รูปที่ 2.40 แสดงกรอบการกำหนดหมายเลขโทรศัพท์ให้ New Connection 37
รูปที่ 2.41 แสดงกรอบแสดงไอคอนใน Dialing-Up Networking 37
รูปที่ 2.42 แสดงกรอบที่ใช้ติดต่อยังเครื่อง Server 38
รูปที่ 2.43 แสดงกรอบขณะจะติดต่อกับเครื่อง Server 39
รูปที่ 2.44 แสดงหน้าต่างเพื่อเข้าสู่ Remote Access Admin 40
รูปที่ 2.45 แสดงกรอบสภาวะของ RAS 40
รูปที่ 2.46 แสดงกรอบสภาวะของ Port ที่ใช้ในการสื่อสาร 40
รูปที่ 2.47 แสดงกรอบรายละเอียดการส่งผ่านข้อมูลของ Port 41
รูปที่ 2.48 แสดงการเข้าสู่การกำหนด Permission ให้ User 42
รูปที่ 2.49 แสดงกรอบ Permission ในการใช้งาน RAS ของ User 43
รูปที่ 2.50 ลักษณะของโปรแกรมที่ทำงานด้วยอีเวนต์ 46
รูปที่ 2.51 หน้าต่างออปเจ็กอินสเปกเตอร์ (Object Inspector) 48
รูปที่ 2.52 แสดงหน้าต่างซอร์สโค้ด (Source Code) 48
รูปที่ 2.53 หน้าต่างการบันทึกโครงงาน (Save Project) 49
รูปที่ 3.1 แสดงข้อมูลแต่ละจุดบนจอภาพ 51
รูปที่ 3.2 แสดง Block Diagram การทำงานของระบบ 51
รูปที่ 3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล 54
รูปที่ 3.4 วงจรแยกสัญญาณซิงค์ 56
รูปที่ 3.5 วงจรนับ(Counter) 58
รูปที่ 3.6 วงจรเลือกแอดเดรสหน่วยความจำ 60
รูปที่ 3.7 วงจรหน่วยความจำภาพ 62
รูปที่ 3.8 วงจรควบคุมการเขียนข้อมูลภาพ 64
รูปที่ 3.9 วงจรอินเตอร์เฟส 66
รูปที่ 3.10 แสดงแผนภูมิการทำงานของโปรแกรมไมโครคอนโทรลเลอร์ 67
รูปที่ 3.11 แสดงแผนภูมิการทำงานของโปรแกรมที่เครื่องเก็บไฟล์ภาพ 68
รูปที่ 3.12 แสดงแผนภูมิการทำงานของโปรแกรมที่เครื่องอ่านไฟล์ภาพ 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 แสดงสัญญาณภาพขาว-ดำที่ได้จากกล้องวีดีโอ 70
รูปที่ 4.2 แสดงสัญญาณภาพที่ผ่านการยกระดับสัญญาณเข้าสู่ขาอินพุต (ขา21)
ของIC CA3318 71
รูปที่ 4.3 แสดงสัญญาณนาฬิกาความถี่ 5 MHz 71
รูปที่ 4.4 แสดงสัญญาณ /CR1 (CH1)ซึ่งใช้เป็นสัญญาณควบคุมการสร้าง
สัญญาณนาฬิกา 5 MHz(CH2) 72
รูปที่ 4.5 a แสดงสัญญาณดิจิตอลที่ได้จากการ IC CA3318 ที่ขาเอาท์พุท (ขา 1-8) 72
รูปที่ 4.5 b แสดงสัญญาณดิจิตอลที่ได้จากการ IC CA3318 ที่ขาเอาท์พุท (ขา 1-8) 73
รูปที่ 4.6 แสดงสัญญาณฮอร์ซิงค์(CH2)ซึ่งถูกแยกออกมาจากสัญญาณภาพ(CH1) 73
รูปที่ 4.7 แสดงสัญญาณเวอร์ซิงค์(CH2)ซึ่งถูกแยกออกมาจากสัญญาณภาพ(CH1) 74
รูปที่ 4.8 แสดงการเปรียบเทียบสัญญาณฮอร์ซิงค์(CH2)และเวอร์ซิงค์(CH1) 74
รูปที่ 4.9 แสดงสัญญาณ ODD/EVEN (CH2)ที่แยกออกมาจากสัญญาณภาพ(CH1) 75
รูปที่ 4.10 แสดงสัญญาณฮอร์เคานท์ 75
รูปที่ 4.11 แสดงการเปรียบเทียบสัญญาณหน่วงเวลา(CH1)กับสัญญาณฮอร์เคานท์(CH2) 76
รูปที่ 4.12 แสดงการเปรียบเทียบสัญญาณฮอรซิงค์(CH1)กับสัญญาณฮอร์เคานท์(CH2) 76
รูปที่ 4.13 แสดงสัญญาณเวอร์เคานท์ 77
รูปที่ 4.14 แสดงสัญญาณฮอร์เคานท์(CH1)ในการควบคุมการสร้าง
สัญญาณเคาน์เตอร์ CA0 (CH2) 77
รูปที่ 4.15 แสดงการใช้สัญญาณเวอร์เคานท์(CH1)ในการควบคุมการสร้าง
สัญญาณเคาน์เตอร์ CA8 (CH2) 78
รูปที่ 4.16 a แสดงสัญญาณเคาน์เตอร์ CA0 (CH1) และ CA1 (CH2) 78
รูปที่ 4.16 b แสดงสัญญาณเคาน์เตอร์ CA2 (CH1) และ CA3 (CH2) 78
รูปที่ 4.17 แสดงสัญญาณเคาน์เตอร์ที่ผ่านเข้าและออกบัฟเฟอร์
74LS244 (U12) CA0 (CH1) และ A0(CH2) 79
รูปที่ 4.18 แสดงสัญญาณเคาน์เตอร์ที่ผ่านเข้าและออกบัฟเฟอร์
74LS244 (U13) CA8 (CH1) และ A8(CH2) 80
รูปที่ 4.19 แสดงสัญญาณข้อมูลดิจิตอลที่ได้จากการแซมปลิ้ง ID0 (U16)
และข้อมูลออกบัฟเฟอร์ D0 (U16) 80
รูปที่ 4.20 แสดงสัญญาณข้อมูลดิจิตอลที่ได้จากการแซมปลิ้ง ID1 (U16)
และข้อมูลออกบัฟเฟอร์ D1 (U16) 81
รูปที่ 4.21 แสดงหน้าต่างแรกเมื่อเริ่มรันโปรแกรมที่เครื่องเก็บไฟล์ภาพ 82
รูปที่4.22 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องเก็บไฟล์ภาพ
(ไม่มีการแก้ไขข้อมูลภาพ) 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่4.23 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องเก็บไฟล์ภาพ
(มีการแก้ไขข้อมูลภาพด้วย Software) 83
รูปที่4.24 แสดงหน้าต่างแรกเมื่อเริ่มรันโปรแกรมที่เครื่องอ่านไฟล์ภาพ 84
รูปที่4.25 แสดงหน้าต่างหลังจากคลิกปุ่ม Connect เพื่อการติดต่อระยะไกล 84
รูปที่4.26 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องอ่านไฟล์ภาพ
(เมื่อเครื่องเก็บไฟล์ภาพหยุดเก็บข้อมูลภาพ) 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 2.1 แสดงสัญลักษณ์ชื่อและตำแหน่งที่มีอยู่ใน SFR 19
ตารางที่ 2.2 หน้าที่และการทำงานของขาต่าง ๆ ของพอร์ตขนาน 24
ตารางที่ 2.3 แสดงแอดเดรสของรีจิสเตอร์ของพอร์ตขนาน 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ระบบรักษาความปลอดภัยผ่านสายโทรศัพท์เป็นการส่งข้อมูลข่าวสารที่เป็นสัญญาณภาพผ่านเครือข่ายสายโทรศัพท์ โดยการแปลงสัญญาณภาพที่เป็นสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลก่อนที่ส่งไป โดยเริ่มจากการแปลงสัญญาณภาพรวม(Composite Signal) เป็นสัญญาณดิจิตอลขนาด 8 บิตต่อ 1 จุดภาพ ซึ่งใช้ไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอลความเร็วสูง(Flash A/D Converter)เบอร์ CA3318 โดยข้อมูลนี้จะถูกส่งไปเก็บยังหน่วยความจำภายนอกขนาด 64 Kbyte เบอร์ 681000 ซึ่งจะถูกควบคุมการเก็บ แล้วจึงใช้ไมโครคอนโทรลเลอร์ MCS-51 เบอร์ 89C51 อ่านข้อมูลจากหน่วยความจำไปเก็บไว้ในคอมพิวเตอร์ และนำเอาค่าดาต้ามาเรียงเป็นไฟล์ภาพ แล้วทำการบันทึกไฟล์ภาพไว้ในคอมพิวเตอร์ จากนั้นถ้าต้องการดูภาพจากเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง เครื่องคอมพิวเตอร์ที่ต้องการดูภาพจะต้องทำการติดต่อผ่านทางโมเด็ม เพื่อใช้ทรัพยากรของเครื่องคอมพิวเตอร์ที่เก็บภาพไว้ จากนั้นก็ทำการอ่านไฟล์ภาพที่เก็บไว้ในเครื่องคอมพิวเตอร์ที่ถูกติดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 ทฤษฎีโทรทัศน์

ภาพเกิดมาจากอะไร อันที่จริงแล้วภาพที่เกิดขึ้นที่จอโทรทัศน์ เป็นภาพนิ่งที่เราเอามาต่อกัน โดยแต่ละภาพจะมีความแตกต่างกันเล็กน้อย และเกิดขึ้นในเวลาอันรวดเร็วทำให้สายตาคนเราเห็นเป็นภาพที่ต่อเนื่อง ในแต่ละเฟรมโดยตัวของมันเองประกอบด้วยพื้นที่เล็ก ๆ ของแสงและจุด หากภาพนั้นมีรายละเอียดของแสงและจุดมากภาพที่ออกมาจะมีรายละเอียดมากกว่าภาพมีรายละเอียดของแสงและจุดน้อย (ซึ่งภาพออกมาหยาบ)

2.1.1 องค์ประกอบภาพ

หากเราตัดภาพจากหนังสือพิมพ์มากสักภาพหนึ่ง แล้วขยายให้ใหญ่ขึ้นด้วยแว่นขยาย จะพบว่ามีองค์ประกอบภาพมาจากจุดสีขาวและสีดำมากมาย มาเรียงกันประกอบขึ้นเป็นภาพ จุดเหล่านี้องค์ประกอบภาพ หรือพิคเจอร์อีลีเมนต์ (Picture Element) หรือพิกเซล (Pixel) ทำนองเดียวกัน ภาพที่ปรากฏทางโทรทัศน์ก็เอามาจากหลักการนี้ ภาพที่เกิดขึ้นบนจอโทรทัศน์ประกอบด้วยเส้นขวางเล็ก ๆ ในแนวนอนจำนวนมาก แต่ละเส้นนั้นทั้งส่วนที่ดำสนิท ส่วนที่จางและส่วนที่สว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสง (Scan) ความแตกต่างกันบนเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองที่เราจัดว่าเป็นองค์ประกอบภาพ ระบบการสแกน 525 เส้นเราอาจแบ่งส่วนทางแนวตั้งได้ 700 เส้น ดังนั้นจึงอาจกล่าวได้ว่าหากจะหาองค์ประกอบภาพในระบบสแกน 525 เส้นแล้ว จะได้เท่ากับ 525×700 เท่ากับ 367,500 พิกเซล ยิ่งภาพมีจำนวนเส้นมากเท่าไรรายละเอียดของภาพก็ยิ่งมีมากขึ้นเท่านั้น เหมือนอย่างที่เรากล่าวถึงรายละเอียดภาพมาแต่ข้างต้น โทรทัศน์ที่มีเส้นสแกนมากย่อมมีรายละเอียดหรือความชัดเจนของภาพมากกว่า แต่การออกแบบวงจรจะยิ่งยากไปด้วยเพราะแบนด์วิดธ์ของความถี่จะต้องกว้างขึ้นด้วย โดยพิจารณาจากสูตรต่อไปนี้

$$f\,max = \frac{1}{2} Kn^2 f_p \frac{b}{h} \frac{y}{k}$$

เมื่อ f max คือ ความถี่สูงสุด

K คือ ค่าคงที่ประมาณ 0.64 ถึง 0.7

n คือ จำนวนเส้นสแกน

f_p คือ จำนวนภาพต่อวินาที

$\frac{b}{h}$ คือ อัตราส่วนแอสเป็ค ซึ่งโดยทั่วไปเราจะใช้อัตรา 4 ต่อ 3

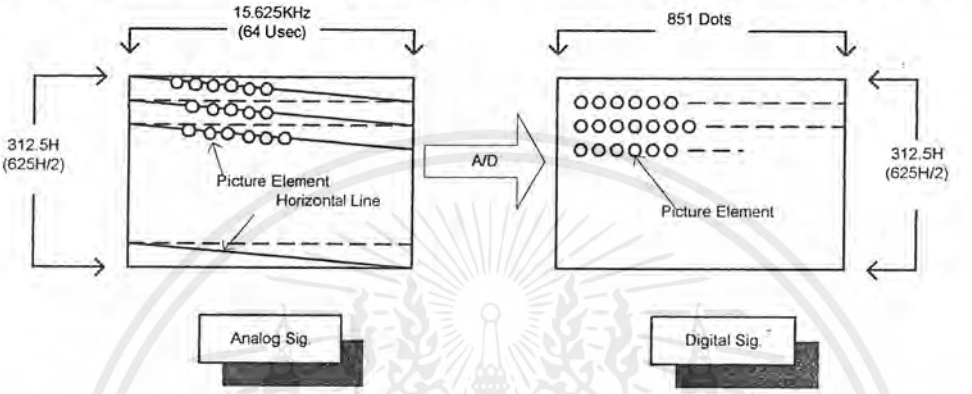
$\frac{y}{k}$ คือ ค่าเอฟเฟ็คดีฟแฟ็คเตอร์ มีค่า 0.95/0.84

จากสูตรนี้เราจะพบว่าถ้าจำนวนเส้นสแกนเพิ่มมากขึ้น ความถี่จะเพิ่มตามขึ้นมากด้วย จากที่กล่าวว่าระบบ 525 เส้นนั้นมีองค์ประกอบภาพ 367,500 พิกเซล แต่ในความเป็นจริงแล้วเรามองเห็นได้ไม่ครบทุกพิกเซล เนื่องจากมีบางส่วนหายไปในการสแกน จากการศึกษาพบว่าภาพที่พอดูได้จะมีองค์ประกอบไม่ต่ำกว่า 200,000 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบที่ใช้อยู่ในประเทศไทย ปัจจุบันเราใช้ระบบการสแกน 625 เส้น กว้าง 7 เมกะเฮิรตซ์ อย่างไรก็ตามองค์ประกอบภาพจะมีรายละเอียดมากขึ้น โดยสามารถหาองค์ประกอบภาพได้จากจำนวนการสแกน 625 เส้น คูณกับจำนวนจุดหรือองค์ประกอบทางแนวตั้ง 851 เส้น ได้เท่ากับ 531,875 พิกเซล

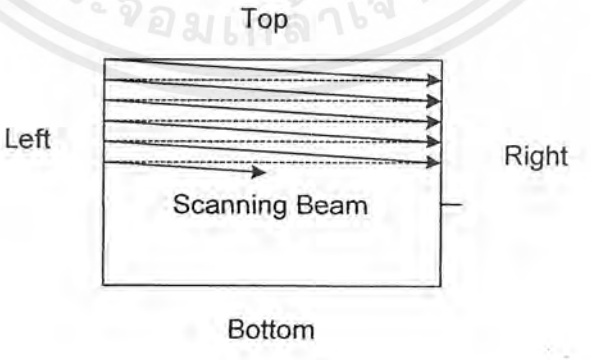
ปัจจุบันส่วนที่เรียกว่าองค์ประกอบภาพได้ถูกนำไปใช้งานอย่างเป็นจริงมากขึ้น ในโทรทัศน์ หรือเครื่องเล่นวีดีโอ คาสเซ็ทเร็คคอร์ดเดอร์ จะมีการนำเอาพิกเซลเหล่านี้ไปเก็บไว้ในหน่วยความจำ เพราะพิกเซลเท่านั้นที่ระบบดิจิตอลจะจัดเก็บข้อมูลได้ เราจะพบวิธีการนี้ในโทรทัศน์ระบบดิจิตอล โทรทัศน์ระบบคอมพิวเตอร์, โทรทัศน์จอภาพแอลซีดี, ดิจิตอล วีซีอาร์, โทรทัศน์หรือวีซีอาร์ ระบบภาพซ้อนภาพ ฯลฯ



รูปที่ 2.1 องค์ประกอบภาพที่ยุคปัจจุบันมีการเอาแต่พิกเซลไปใช้งานกันแล้ว

2.1.2 การสแกน

องค์ประกอบภาพที่ได้กล่าวมาแล้วในตอนต้น เครื่องส่งจะรับภาพแล้วเปลี่ยนพลังงานแสงเป็นพลังงานไฟฟ้าส่งเป็นจุดขาว ดำมา เครื่องรับจะรับเอาสัญญาณเหล่านี้มาเรียงกันใหม่ โดยการวาดเป็นเส้นภาพขึ้นที่หน้าจอ วิธีนี้เราเรียกว่าการสแกน (Scan) โดยตัวที่มีหน้าที่สำคัญคือจอภาพหรือหลอดภาพนั้นเอง หลอดภาพมีโครงสร้างคล้ายกับหลอดสุญญากาศทั่วไปที่ปล่อยอิเล็กตรอนออกมาจากแคโถด แล้วจะมีการดึงลำอิเล็กตรอนให้วิ่งไปเป็นลำกระทบเข้ากับอาโนดหรือหน้าจอ ตรงหน้าจอนั้นมีการฉาบสารเรืองแสงไว้เพื่อให้เกิดการเรืองแสงของจอ



รูปที่ 2.2 แสดงวิธีการเบื้องต้นของการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

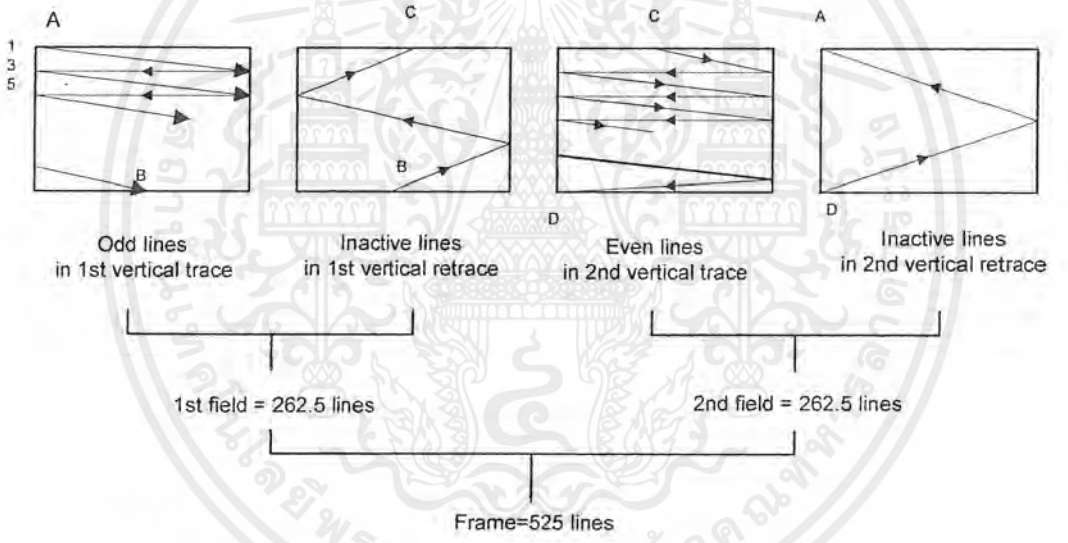
การสแกนของจอมีสองวิธีคือการสแกนแบบก้าวหน้า (Progressive Scanning) กับการสแกนแบบสลับเส้น (Interlaced Scanning)

การที่จะทำการสแกนมีความต่อเนื่องขององค์ประกอบภาพดังที่กล่าวมาแล้วจะต้องคำนึงถึงหลัก 3 ประการ คือ

1.ลำอิเล็กตรอนที่กวาดไปทางแนวนอน (Horizontal Scanning) ในแต่ละครั้ง จะต้องสามารถควบคุมองค์ประกอบภาพทั้งหมดของเส้นนั้น

2.ในแต่ละเส้นของการสแกนลำอิเล็กตรอน ลำแสงต้องกวาดกลับเป็นความเร็วสูงไปยังด้านซ้ายเพื่อเริ่มต้นเส้นภาพทางแนวนอนในลำดับต่อไป เวลาของการสะบัดกลับเรียกว่า "รีเทรซ"(Retrace) หรือ ฟลายแบ็ค (Flyback) ในกรณีดังกล่าวจะต้องไม่มีข้อมูลภาพใด ๆ เพราะว่าทั้งกล้องถ่ายและหลอดภาพจะเกิดการแบล็งค์เอาท์ (Blank Out) ในขณะนั้น

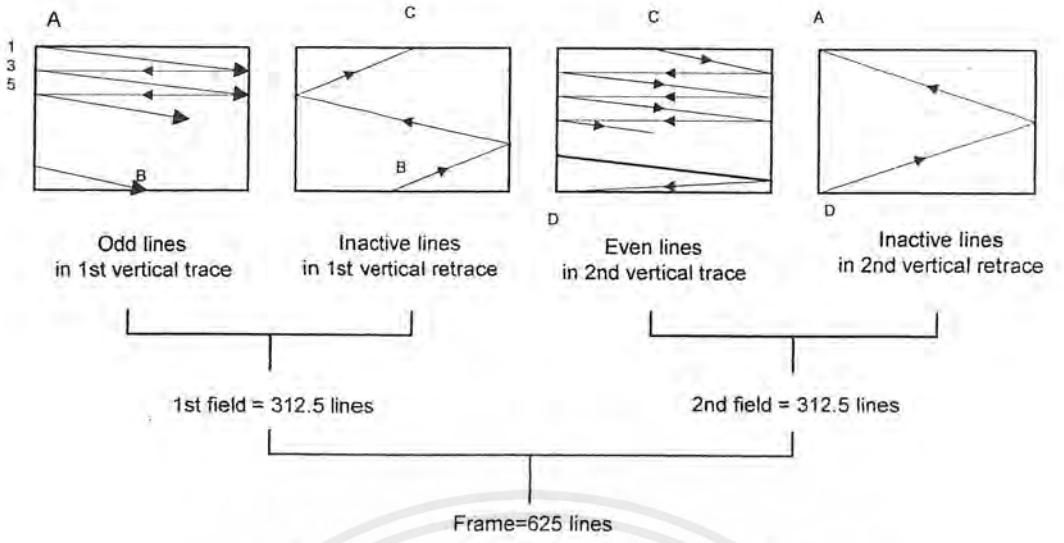
3.ในขณะที่เส้นสแกนสะบัดกลับมาเพื่อเริ่มต้นทางซ้ายใหม่ ตำแหน่งทางแนวตั้งต้องต่ำกว่าตำแหน่งเดิมเพื่อทำให้การสแกนเส้นต่อไปไม่ทับกัน ทั้งนี้โดยการควบคุมของสัญญาณทางแนวตั้ง



(Vertical Scanning)

รูปที่ 2.3 รายละเอียดการสแกนแบบสลับเส้นหรือแบบแทรกสอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การสแกนสลับเส้นหรือแบบแทรกสอดในระบบ CCIR

การสแกนที่ใช้ในเครื่องรับโทรทัศน์ถึงแม้เราจะพบว่าหากให้มีการเรียงภาพเกินกว่า 16 ภาพต่อวินาทีแล้ว สายตาจะเห็นเป็นภาพต่อเนื่อง จากการทดลองสแกนภาพเราพบว่าแม้ภาพที่เกิดขึ้น 24 ภาพต่อวินาทีแล้วก็ตาม ยังมีการกระพริบ (Flicker) เกิดขึ้น เนื่องจากว่าในขณะที่มีการสแกนเริ่มจากขอบบนลงมาด้านล่าง (ซึ่งคล้ายกับการเขียนหนังสือที่เริ่มจากซ้ายไปขวา บนลงมาล่าง) เมื่อเส้นสแกนลงมายังด้านล่างแสงทางด้านบนในความรู้สึกของมนุษย์เริ่มมืดลงกว่าด้านล่างเวลาที่ลำแสงการสแกนวนกลับไปด้านบนด้านล่างก็เกิดปัญหาเช่นเดียวกัน ความรู้สึกต่อกรณีนี้คือเกิดแสงกระพริบหรือวูบวาบขึ้น (Flicker) จึงต้องใช้การสแกนแบบสลับเส้นหรือบางคนเรียกว่าการสแกนแบบสอดแทรก (Interlaced Scanning) โดยครั้งแรกจะสแกนฟิลด์คี่ (Odd Line Trace) และครั้งต่อไปจะสแกนฟิลด์คู่ (Even Line Trace) เป็นการสแกนแบบเส้นเว้นเส้น นั่นหมายความว่าการได้ภาพ 1 ภาพหรือภาพ 1 เฟรม (Frame) ต้องใช้การสแกนแนวตั้งถึง 2 ครั้ง หรือ 2 ฟิลด์ (Field)

ตามมาตราฐาน เอฟซีซี (Fcc) ใช้การสแกน 525 เส้นต่อ 1 ภาพ และใช้ภาพ 30 ภาพต่อวินาที ดังนั้นใน 1 ฟิลด์ จะมีเส้นสแกน 262.5 เส้น ภาพนิ่งแต่ละภาพซึ่งเป็นส่วนประกอบขององค์ประกอบจะเกิดขึ้นภายใน 1/30 วินาที ความถี่ที่ใช้เพื่อการหักเหลำอิเล็กตรอนในแนวนอนจึงได้มาจากจำนวนเส้นภาพ คูณกับจำนวนภาพในแต่ละเฟรม จึงได้ 525 × 30 = 15750 เฮิรตซ์ ส่วนความถี่บังคับหักเหทางแนวตั้งในหนึ่งฟิลด์จะใช้เวลาเพียง 1/60 วินาทีเท่านั้น ความถี่จึงเท่ากับ 60 เฮิรตซ์

ตามมาตราฐาน ซีซีไออาร์ (CCIR) ใช้เส้นสแกน 625 เส้น ต่อ 1 ภาพและใช้ภาพ 25 ภาพต่อวินาที ดังนั้นใน 1 ฟิลด์จะมีเส้นสแกน 312.5 เส้น ภาพนิ่งแต่ละภาพซึ่งเป็นส่วนประกอบขององค์ประกอบภาพจะเกิดขึ้นภายใน 1/25 วินาที ความถี่ที่ใช้เพื่อหักเหลำอิเล็กตรอนในแนวนอนจึงได้จากจำนวนเส้นภาพ 625 เส้น คูณกับจำนวนภาพในแต่ละเฟรม ดังนั้นเราจึงสามารถหาความถี่ได้จาก 625 × 25 = 15625 เฮิรตซ์ ความถี่หักเหทางแนวตั้งจึงเท่ากับ 50 เฮิรตซ์

จากรูปที่ 2.3 และ 2.4ได้แสดงวิธีการของการสแกนแบบสอดแทรกของระบบโทรทัศน์ทั้ง 2 ระบบใหญ่ที่ใช้กันอยู่ในปัจจุบัน โดยเมื่อเริ่มการสแกนสมมุติว่าการเริ่มสแกนในกรณีนี้เริ่มจากการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สแกนจากเฟรมที่เป็นเส้นสแกนคี่ โดยเริ่มจาก A ซึ่งอยู่ทางซ้ายแล้วกวาดไปทางขวา นับเป็นเส้นสแกนเส้นที่ 1 แล้วจึงสแกนเส้นที่ 3 5 7 9 และต่อ ๆ ไปเรื่อย ๆ จนกระทั่งได้เส้นสแกน 262.5 เส้นในระบบเอฟซีซี หรือ 312.5 เส้นในระบบ ซีซีไออาร์ ซึ่งก็คือสแกนมาถึงจุด B ดังในภาพที่ 2.4 และ ณ จุดนี้การสแกนจะถูกความถี่หักเหทางแนวตั้งซึ่งเราเรียกว่าเวอร์ติคอล รีเทรซ (Vertical Retrace) หรือสัญญาณฟลายแบ็ค (Flyback) ดึงกลับไปยังตำแหน่งในจุด C เพื่อเริ่มต้นสแกนเส้นคู่ต่อไป

เวลาของการรีเทรซ (Retrace Time) ทั้งการรีเทรซทางเวอร์ติคอลและฮอริซอลตอลเป็นเวลาสั้น ๆ ถึงอย่างไรก็ตามเราไม่ต้องการให้เส้นสแกนที่เป็นช่วงของการสะบัดกลับนี้เข้ามารบกวนการทำให้เกิดสัญญาณภาพ ในส่วนนี้จึงต้องทำการลบเส้นสะบัดกลับ ก่อนที่จะถึงจุดที่ว่านั้นเราลองมาดูรายละเอียดของการสะบัดกลับอีกนิดว่าในส่วนของการกวาดลำแสงหรือการสแกนในทางแนวนอน (Horizontal) เวลาของการรีเทรซจะได้ประมาณ 10-16 เปอร์เซ็นต์ของเวลาทั้งหมด ในระบบเอฟซีซีเราใช้เวลาในการสแกนทางแนวนอนเท่ากับ 63.5 uS ดังนั้นเวลาของการรีเทรซจึงเท่ากับ 6.35 uS ส่วนระบบซีซีไออาร์ เราใช้เวลาในการสแกนเท่ากับ 64 uS ดังนั้นเวลาของการรีเทรซจะได้ประมาณ 6.4 uS ในส่วนของทางด้านความถี่หักเหทางแนวตั้ง (Vertical) เราใช้เวลาในการรีเทรซไม่เกิน 5-8 เปอร์เซ็นต์ อย่างเช่นเราใช้เวลาในส่วนนี้เท่ากับ 3 เปอร์เซ็นต์ของเวลา 1/50 หรือ 1/60 ของวินาที ดังนั้นเวลาของการรีเทรซจึงเท่ากับ 600 uSและ 500 uS ตามลำดับ นั่นหมายความว่าช่วงเวลาของการรีเทรซทางแนวตั้งกินเวลานานกว่าการสแกนทางแนวนอนประมาณ 8-10 เส้นภาพ

จากหลักการดังกล่าวเราสามารถสรุปกฎเกณฑ์บางอย่างได้ว่า ตามความเป็นจริงแล้วในเส้นภาพ 525 หรือ 625 เส้นนั้นเรามิอาจจะเห็นได้ครบทุกเส้น อย่างน้อย ๆ ในกรณีที่เกิดเวอร์ติคอลรีเทรซจะกินเวลาของการสแกนทางแนวนอน (Line) ไปด้วย แต่จะกินไปกี่เส้นนั้นขึ้นอยู่กับสัญญาณบังคับการฟลายแบ็ค ซึ่งในเครื่องรับเราเรียกตัวนี้ว่าสัญญาณแบลงกิ้ง

จากหลักการดังกล่าวนี้เองที่ไขประตูไปสู่โทรทัศน์ระบบดิจิตอลทำไมเวลาที่เขาเอาสัญญาณภาพเข้าสู่หน่วยความจำของระบบดิจิตอล เขาจึงเอาเส้นภาพไปเก็บเพียงครั้งละ 308 เส้นภาพเท่านั้น (แทนที่จะเอาไปเก็บทั้ง 312.5 เส้นภาพ) เพราะจากมาตรฐานโทรทัศน์ในปัจจุบันที่กำหนดมาตรฐานสัญญาณในกรณีรีเทรซทางแนวตั้งเท่ากับ 3 เปอร์เซ็นต์ เราจึงได้เส้นภาพประมาณ 625 – 10 เส้นภาพ (615 เส้นภาพ) เมื่อแบ่งเฟรมแล้วจะได้ค่าเส้นภาพประมาณ 308 เส้นภาพ

2.1.3 สัญญาณภาพรวม

หากจะถามว่าเครื่องส่งส่งสัญญาณอะไรมาให้เครื่องรับบ้าง หากตอบกันง่าย ๆ ก็ต้องตอบว่าส่งสัญญาณภาพรวม (Composite Video Signal) ซึ่งการจะทำให้เครื่องรับบรรลุวัตถุประสงค์นั้นต้องให้สัญญาณโทรทัศน์ส่งสัญญาณต่าง ๆ ดังต่อไปนี้

1.สัญญาณเสียง

2.สัญญาณภาพ

3.สัญญาณแบลงกิ้ง

4.สัญญาณซิงโครไนซ์

5.สัญญาณอีควอไลซิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของระบบสัญญาณเสียงจะใช้คลื่นพาห์ (Carier) เฉพาะ เพราะทราบกันเบื้องต้นแล้วว่าระบบเสียงในโทรทัศน์เป็นแบบ เอฟเอ็ม. ส่วนสัญญาณภาพและอื่น ๆ ที่เหลือนั้นเราจะส่งเป็นสัญญาณภาพรวมหรือคอมโพสิท วิดีโอ ซิกแนล (Composit Video Signal) แล้วใช้คลื่นพาห์ภาพส่งออกไป การที่เราต้องส่งสัญญาณทั้ง 5 ตัวออกอากาศแพร่คลื่นออกไปเพื่อวัตถุประสงค์ดังนี้

1.สัญญาณภาพและสัญญาณเสียง เป็นสัญญาณที่ส่งออกไปเพื่อให้เกิดภาพและเสียงขึ้นในเครื่องรับโทรทัศน์

2.สัญญาณแบลงกิ้ง เป็นสัญญาณที่ส่งเพื่อให้ลบเส้นสะบัดกลับทางแนวตั้งและแนวนอน

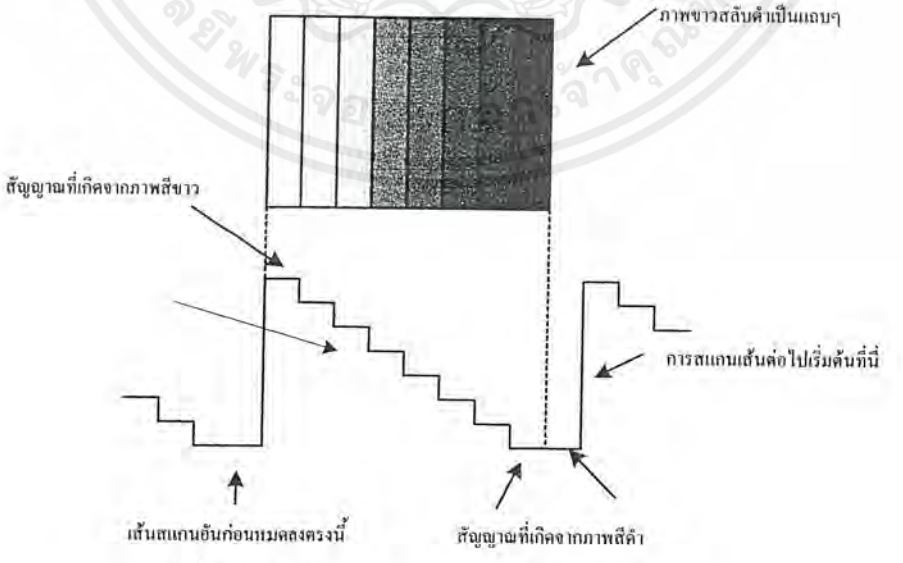
3.สัญญาณซิงโครไนซ์ เป็นสัญญาณที่ส่งมาเพื่อช่วยให้วงจรหักเหทางแนวตั้งและวงจรหักเหทางแนวนอน เพื่อให้เครื่องส่งกับเครื่องรับทำงานสอดคล้องตรงกัน

4.สัญญาณอีควอไลซิ่ง เป็นสัญญาณที่ช่วยให้สัญญาณซิงโครไนซ์ทั้งแนวตั้งและแนวนอนยังคงรูปเดิมอยู่ได้ แม้ว่าจะเป็นการสแกนแบบสลับเส้นก็ตาม

ก.สัญญาณขาว-ดำ

สมมุมติว่าเราจะดูระดับสัญญาณขาว – ดำ กรณีที่เรากล่าวถึงสัญญาณขาว – ดำ หรือสัญญาณโมโนโครมได้ดีที่สุดต้องกล่าวว่าภาพที่เป็นสีขาวคือภาพที่มีความสว่างมากที่สุด และภาพที่เป็นสีดำคือภาพที่ไม่มีความสว่างเลย ภาพจำลองที่ดีที่สุดของกรณีคือแถบความถี่ที่มีความแตกต่างของสัญญาณขาว – ดำ ที่ละน้อย ซึ่งเราเรียกว่าระดับเกรย์สเกลนั้นเอง

จากภาพที่ 2.5 แสดงให้เห็นระดับของการเกรย์สเกล ในกรณีที่ระดับสัญญาณภาพเป็นระดับสัญญาณสีขาวเข้ามา นั่นหมายความว่าระดับความแรงของสัญญาณภาพที่มากที่สุดจึงให้ความสว่างที่หน้าจอสว่างที่สุด และเมื่อระดับสัญญาณที่ขาวลดลงเป็นสีมัว , เทา และดำ นั้น ระดับสัญญาณจะลดลงเรื่อย ๆ นั่นหมายความว่าสัญญาณมีความแรงน้อยลงจะมีความสว่างน้อยลงด้วย อันที่จริงแล้วภาพแต่ละภาพเป็นสัญญาณความถี่ทางไฟฟ้าที่มีความถี่สูงต่ำไม่เท่ากัน โดยความถี่สูงสุดจะไม่เกิน 4 เมกะเฮิรตซ์ในระบบ



รูปที่ 2.5 แสดงระดับสัญญาณ ขาว-ดำ โดยเกรย์สเกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอฟซีซี และไม่เกิน 5 เมกะเฮิรตซ์ ในระบบซีซีไออาร์ ภาพที่เกิดขึ้นจากความถี่สูงย่อมมีความละเอียดกว่าภาพที่เกิดจากความถี่ต่ำ (มีจำนวนจุดดำมากกว่า)

ข.สัญญาณแบลงกิ้ง ทราบแล้วว่าเมื่อมีการสแกนลำอิเล็กตรอนที่หน้าจอจะเกิดเส้นรีเทรสหรือเส้นสะบัดกลับ ซึ่งเป็นเส้นภาพที่เราไม่ต้องการ เครื่องส่งจึงต้องส่งสัญญาณแบลงกิ้ง (Blanking) เพื่อให้เครื่องรับสามารถลบเส้นสะบัดกลับได้ สัญญาณแบลงกิ้งส่วนหนึ่งเครื่องรับจะต้องสร้างขึ้นเหมือนการสร้างสัญญาณซิงโครไนซ์แต่มันจะสัมพันธ์กับเครื่องส่งได้อย่างไร จึงต้องมีการส่งสัญญาณแบลงกิ้งมาจากเครื่องส่งเพื่อลบเส้นสะบัดกลับในเครื่องรับ สัญญาณแบลงกิ้งมีอยู่ 2 อย่าง คือ เวอร์ติคอลแบลงกิ้ง และฮอริซอลตอลแบลงกิ้ง

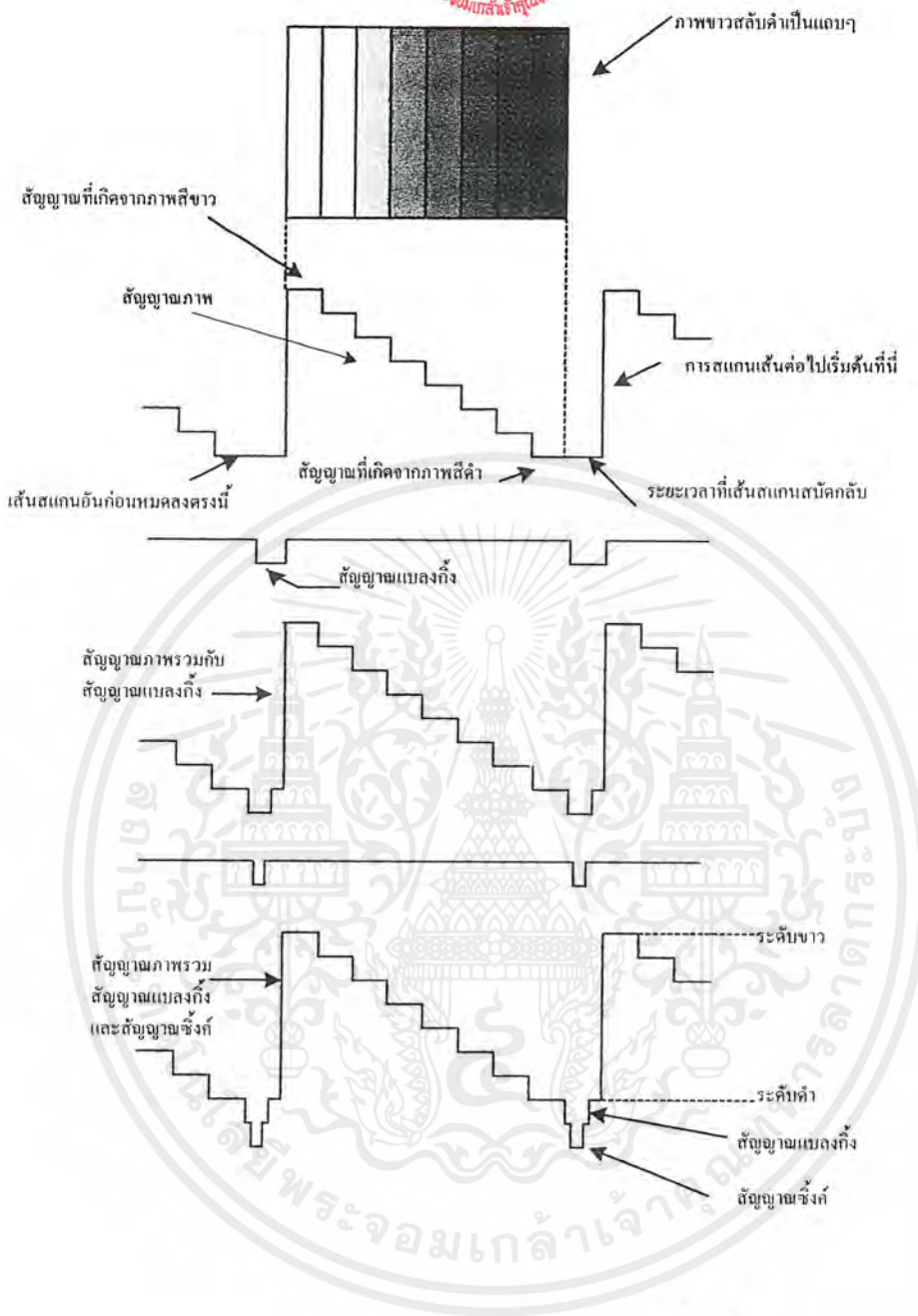
ค.สัญญาณซิงโครไนซ์

เป็นสัญญาณเพื่อให้การสแกนเป็นไปอย่างถูกต้อง ทั้งการสแกนแนวตั้งและแนวนอน โดยสัญญาณซิงโครไนซ์หรือสัญญาณซิงค์มีอยู่ 2 อย่าง คือ

1.ฮอริซอลตอลซิงโครไนซ์ (Horizontal Synchronize) เป็นสัญญาณซิงโครไนซ์ทางแนวนอน ซึ่งมีความถี่ 15625 เฮิรตซ์ (ในระบบ CCIR) หรือ 15750 เฮิรตซ์ (ในระบบ FCC) ถ้าไม่มีสัญญาณส่วนนี้ส่งมาจะทำให้ภาพเกิดการล้มได้

2.เวอร์ติคอลซิงโครไนซ์ (Vertical Synchronize) เป็นสัญญาณซิงโครไนซ์ซึ่งมีความถี่ทางแนวตั้ง 50 เฮิรตซ์ (ในระบบ CCIR) หรือ 60 เฮิรตซ์ (ในระบบ FCC) ถ้าไม่มีสัญญาณส่วนนี้จะทำให้ภาพเลื่อน เนื่องจากสัญญาณซิงโครไนซ์กับสัญญาณแบลงกิ้งไม่ว่าทางแนวนอนหรือแนวตั้ง จะมีความถี่เท่ากันเวลาส่งจึงต้องทำการกำหนดตำแหน่งให้ถูกต้อง มิฉะนั้นแล้วจะเกิดกวนกันได้ ในทางปฏิบัติจึงให้สัญญาณซิงค์พัลซ์มีขนาดความกว้างน้อยกว่าสัญญาณแบลงกิ้งพัลซ์ แล้วใช้วิธีการส่งแบบผสม กล่าวคือส่งซิงค์พัลซ์ กับ แบลงกิ้งพัลซ์ไปด้วยกัน ให้แบลงกิ้งพัลซ์เป็นฐานของซิงค์พัลซ์ เมื่อมีการจัดระดับของสัญญาณส่วนนี้เทียบกับเกรย์สเกล ระดับของสัญญาณแบลงกิ้งจะอยู่ที่ระดับต่ำกว่าดำ ส่วนสัญญาณซิงค์จะอยู่ที่ระดับต่ำกว่าดำลงไปอีก สัญญาณเหล่านี้จึงไม่มีการส่งผลต่อการมองเห็น (หรือไม่กวนสัญญาณภาพ) ดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

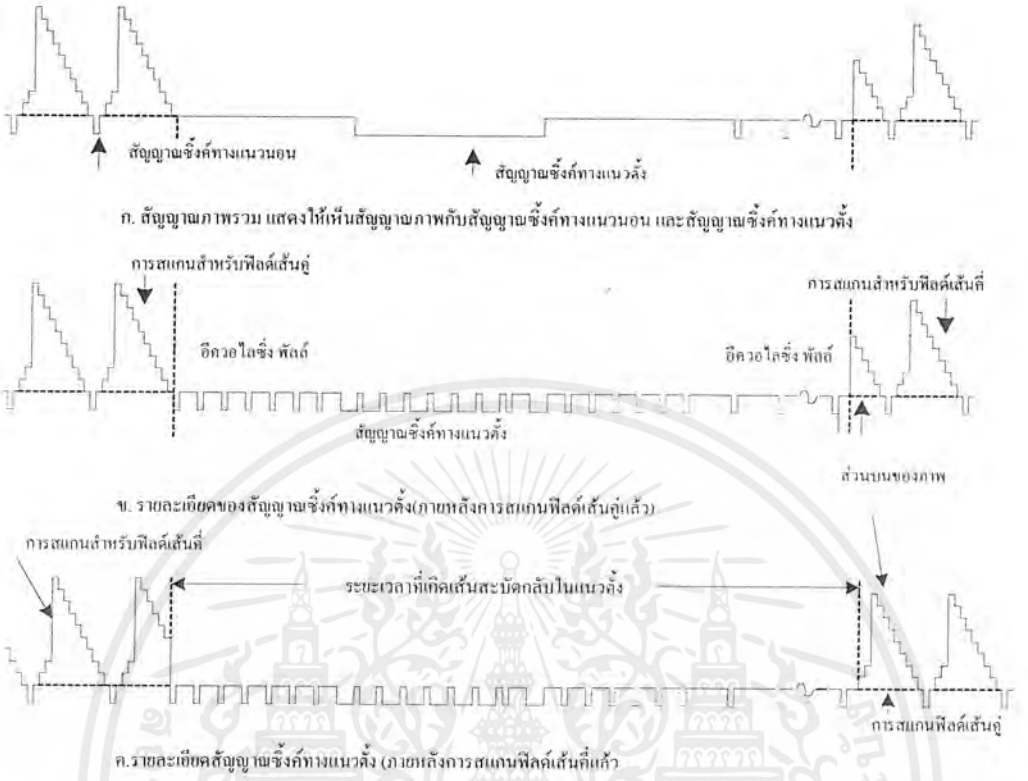


รูปที่ 2.6 แสดงลักษณะสัญญาณภาพรวมเบื้องต้น

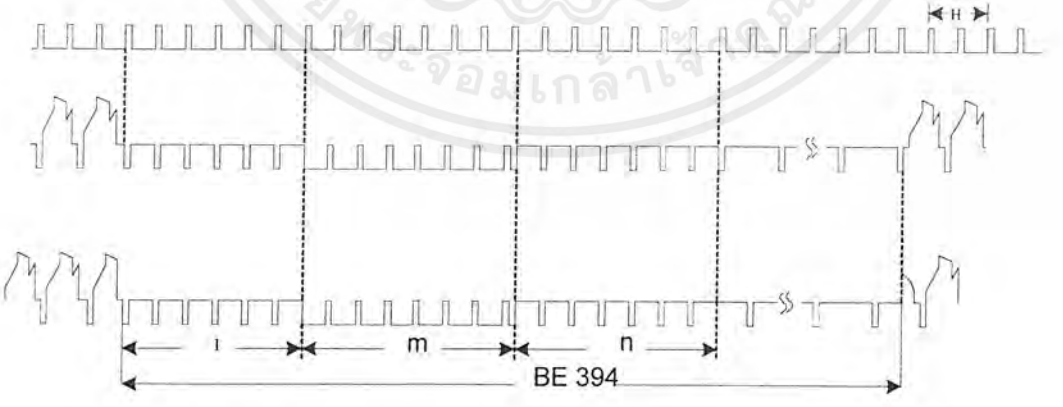
ง.สัญญาณอีควอไลซิงก์

เป็นสัญญาณบังคับรูปร่างของสัญญาณซิงโครไนซ์ทางแนวตั้ง เพื่อให้สามารถคงรูปถูกต้อง แล้วยังช่วยให้การสแกนสลับเส้นเป็นไปได้ถูกต้องด้วย สามารถขึ้นเส้นคู่เส้นคี่ในตำแหน่งที่ถูกต้องได้ ทั้งยังส่งผลให้สัญญาณซิงโครไนซ์ทางแนวนอนไม่ขาดช่วงหายไปในระหว่างการส่งสัญญาณทางแนวตั้ง สัญญาณตัวนี้มีขนาดของพัลซ์รวมเท่ากับเวอร์ติคอลซิงโครไนซ์พัลซ์ สัญญาณส่วนนี้จะถูกแบ่งออกเป็น 6 ลูกเล็ก ๆ ในระบบ 525 เส้น และถูกแบ่งเป็น 5 ลูกในระบบ 625 เส้น ดังแสดงไว้ในรูป 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง



รูปที่ 2.8 แสดงระบบพัลซ์ในฟิลด์ของการซิงโครไนซ์ของระบบโทรทัศน์

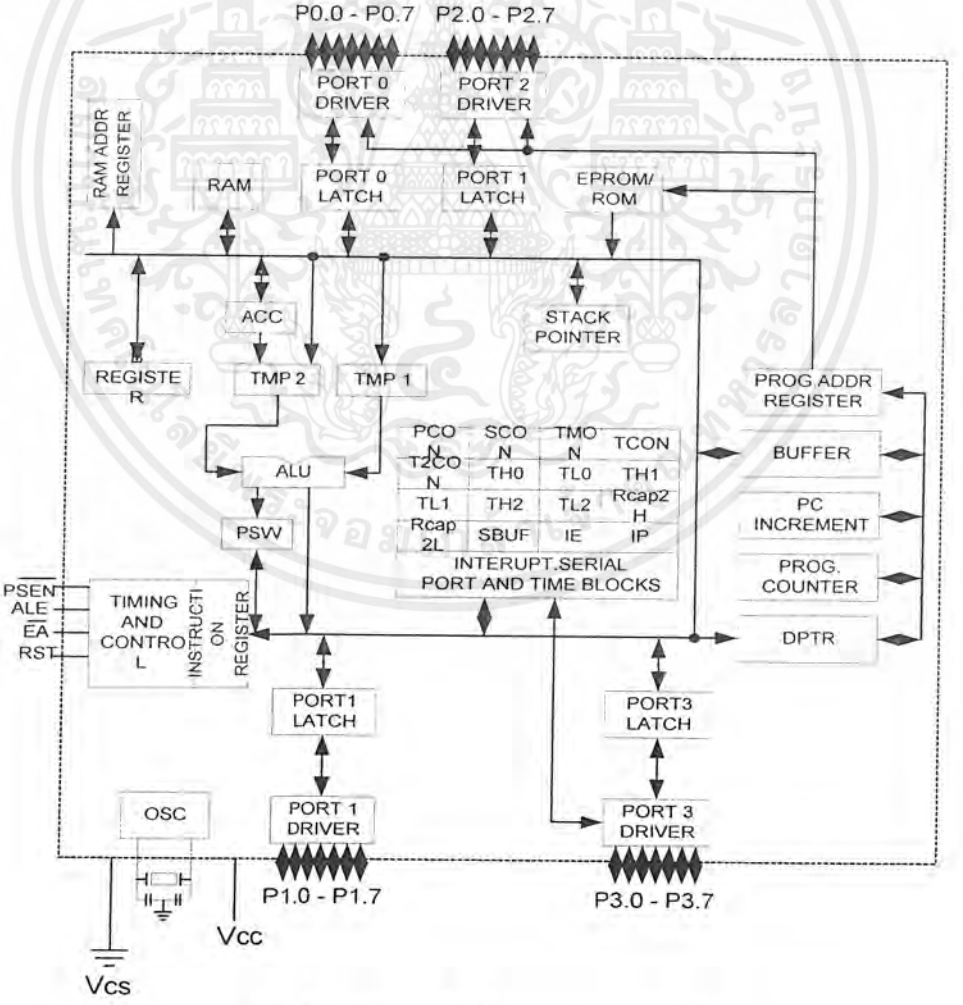
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างของ MCS-51

2.2.1 คุณสมบัติของไมโครคอลโทรลเลอร์ตระกูล MCS-51

- ต้องการแหล่งจ่ายไฟ +5 V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์ สำหรับเบอร์ 8051 และเบอร์ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำ 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 มาถึง 256 ไบต์
- หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทม์เมอร์ เคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับเบอร์ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเตอร์รัปต์ได้ 6 แหล่ง 5 เวกเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปได้ 4 โหมด
- มีคำสั่งในการทำ AND , OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

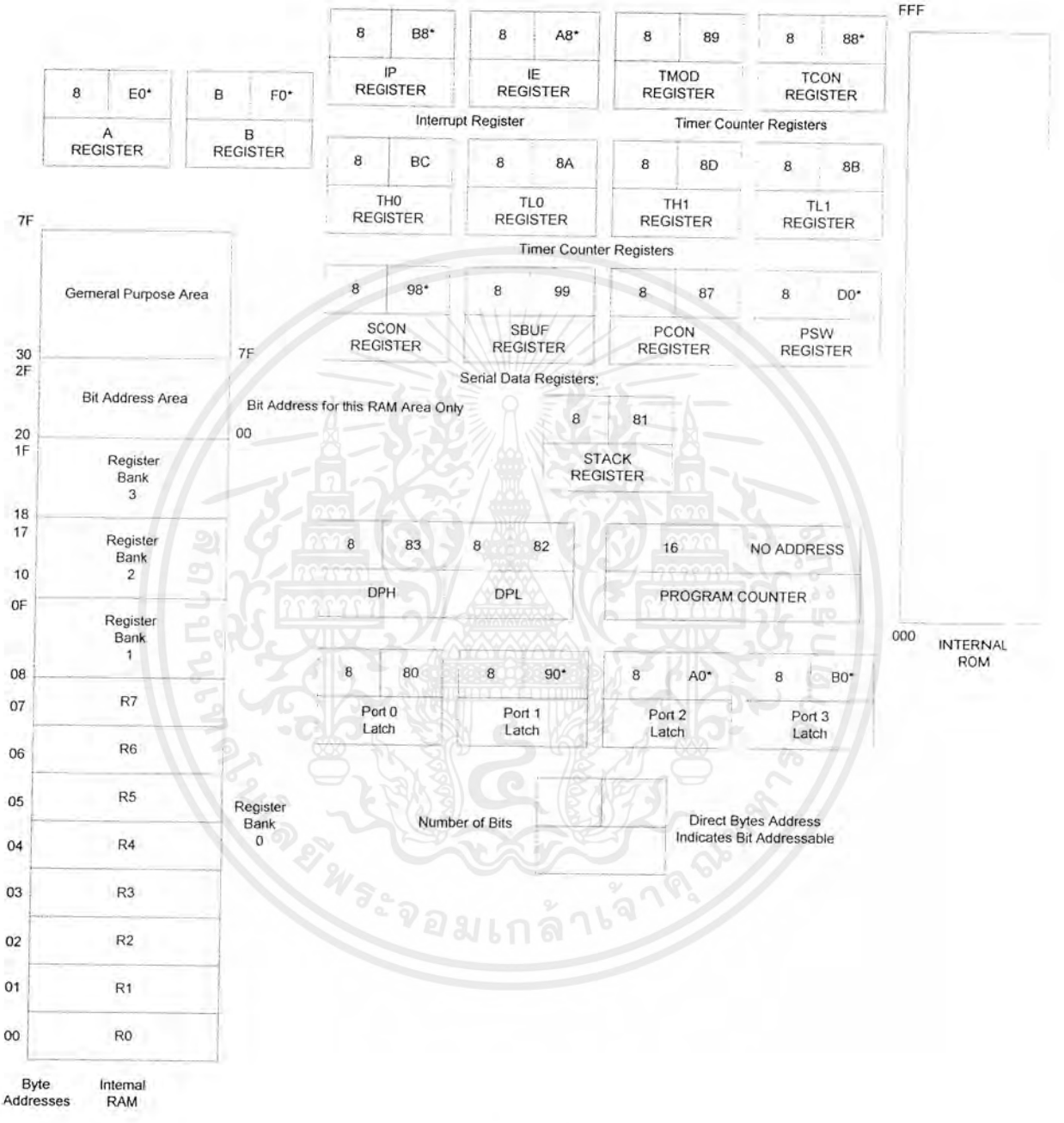
2.2.2 โครงสร้างภายในของ 8051



รูปที่ 2.9 a 8051 บล็อกไดอะแกรมของ MCS-51

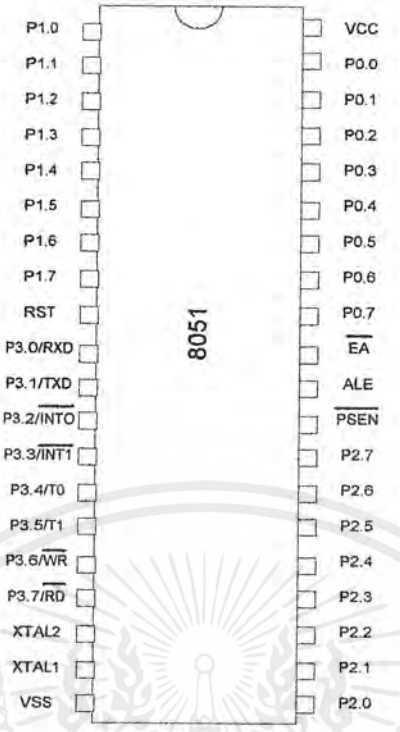
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS-51 ใช้เทคโนโลยีแบบ NMOS และ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายในจึงสะดวกสำหรับโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษาเบสิก โครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงในรูป 2.9 a และ 2.9 b



รูปที่ 2.9 b ตำแหน่งของรีจีสเตอร์ต่าง ๆ และหน่วยความจำภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

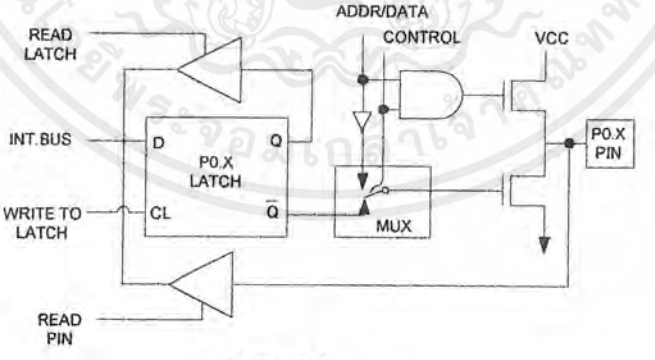


รูปที่ 2.10 การจัดวางขาของ 8051

2.2.3 พอร์ทของ 8051

8051 เป็นไมโครคอลโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่าง ๆ ดังนี้

- Vcc (ขา 40) ต่อกับ + 5 V
- Vss (ขา 20) เป็นขา GND
- พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7 – P0.0) มีโครงสร้างแบบ Open Drain Bi-Directional ดังแสดงในรูปที่ 2.11



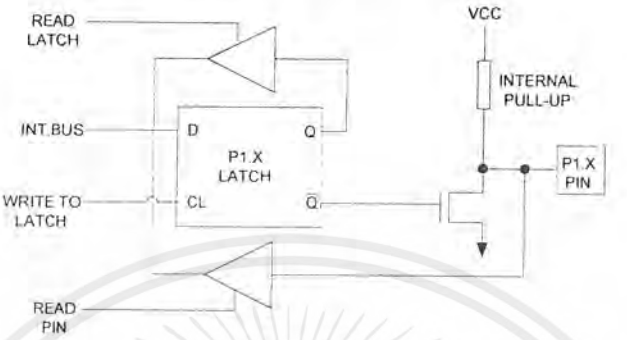
รูปที่ 2.11 แสดงโครงสร้างของพอร์ท 0 บิต

พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ 2 หน้าที่ คือแอดเดรสบัสและดาต้าบัส เมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ท ถ้าต้องการให้ทำงานเป็นอินพุทพอร์ทจะต้องส่งลอจิก 1 ไปยังพอร์ทนี้ จะมีผลให้ Q ของ D-FF เป็น 0 ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ทแลทช์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบน และการอ่าน Port (pin) จะใช้สัญญาณ Read (pin)

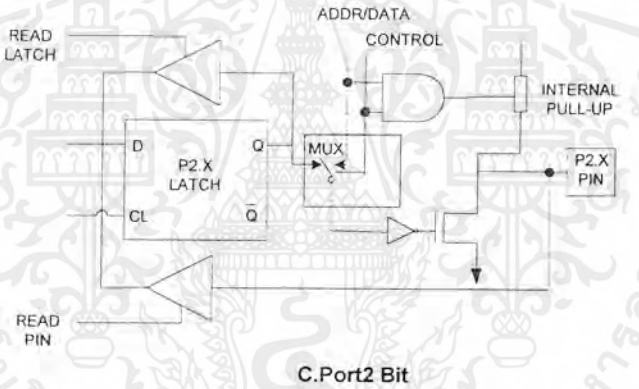
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ใช้อ่านอินพุทพอร์ตแลทช์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบน และการอ่าน Port (pin) จะใช้สัญญาณ Read (pin)

พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้ายพอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน Internal Pull up Register มีโครงสร้างดังรูป 2.12



รูปที่ 2.12 แสดงโครงสร้างของพอร์ต 1 บิต



รูปที่ 2.13 แสดงโครงสร้างของพอร์ต 2 บิต

พอร์ต 2 (ขา 21-28) มีทั้งหมด 8 บิต คือ (P2.7-2.0) มีโครงสร้างคล้ายพอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนตัวบนใช้ความต้านทานพูลอัพแทน (Internal Pull up) พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ตที่ใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุทพอร์ตต้องส่งลอจิก 1 มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FET อยู่ในสภาวะ off ดังแสดงในรูป 2.13

พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ ขา(P3.7-P3.0) มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ตถ้าจะโปรแกรมให้เป็นอินพุทพอร์ตต้องส่งลอจิก 1 มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งคือใช้ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไปสัญญาณต่าง ๆ มีดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)

P3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.3/INT1(External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

P3.4/T0 (Counter 0 External Input) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 0 (เป็นอินพุทโหมดเคาน์เตอร์)

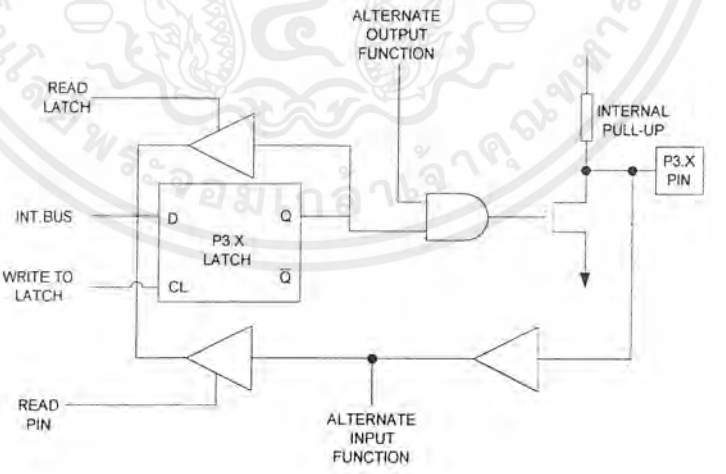
P3.5/T1 (Counter 1 External Input) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 1 (เป็นอินพุทโหมดเคาน์เตอร์)

P3.6/$\overline{WR}$ (External Data Memory Write Strope) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.7/$\overline{RD}$ (External Data Memory Read Strope) ขาสัญญาณควบคุมการอ่านข้อมูลลงหน่วยความจำข้อมูลภายนอก

โครงสร้างพอร์ต 3 ดังแสดงในรูปที่ 2.14

- **ALE** (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตช์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอคทีฟทุก ๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล
- $\overline{PSEN}$ (ขา 29) เป็นขาสโตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Memory จะไม่มีสัญญาณออกขานี้
- (ขา 30) ใช้เลือกหน่วยความจำโปรแกรมภายนอก
  ป้อน "0" จะอ่านโปรแกรมจากภายนอกชิพ
  ป้อน "1" จะอ่านโปรแกรมจากภายในชิพ

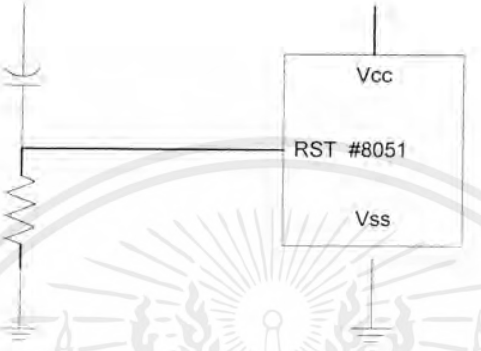


D.Port3 Bit

รูปที่ 2.14 แสดงโครงสร้างของพอร์ต 3 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **RST** (ขา 9) ขารีเซ็ตจะรีเซ็ตได้ก็ต่อเมื่อป้อนลอจิก 1 เข้าไปที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล
- **XTAL1** (ขา 19) ใช้ต่อคริสตอลภายนอกเพื่อใช้เป็นอินพุทเข้าสู่วงจรออสซิลเลเตอร์ภายใน
- **XTAL2** (ขา 19) ใช้ต่อคริสตอลภายนอกเพื่อใช้เป็นเอาท์พุทเข้าสู่วงจรออสซิลเลเตอร์ภายใน



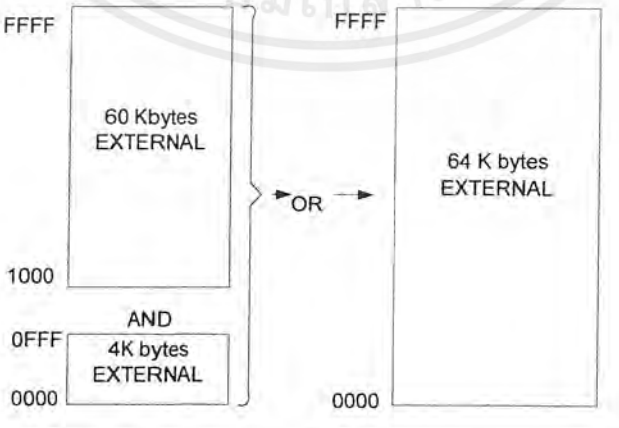
รูปที่ 2.15 การต่อขารีเซ็ตให้กับ 8051

2.2.4 การแบ่งประเภทของหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS-51 มีอยู่ด้วยกัน 2 ชนิด คือ

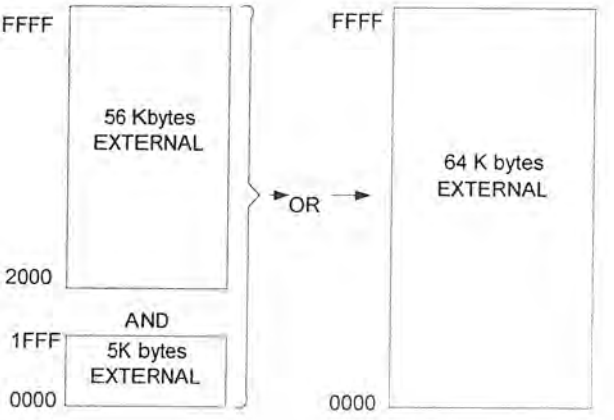
- หน่วยความจำสำหรับโปรแกรม (Program Memory)
- หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หน่วยความจำสำหรับโปรแกรม เป็นหน่วยความจำที่ใช้เก็บโปรแกรมสำหรับสั่งงานบรรจุอยู่ในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 กิโลไบต์ นั่นเอง แต่ถ้าเป็นเบอร์ 8052 จะมี ROM ขนาด 8 กิโลไบต์ ดังแสดงในรูป 2.16 และ 2.17



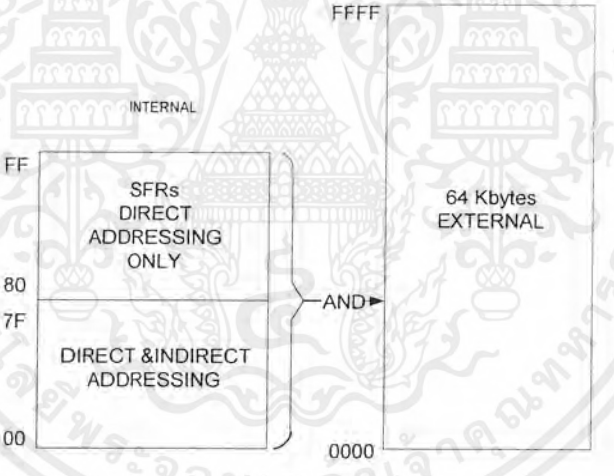
รูปที่ 2.16 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052

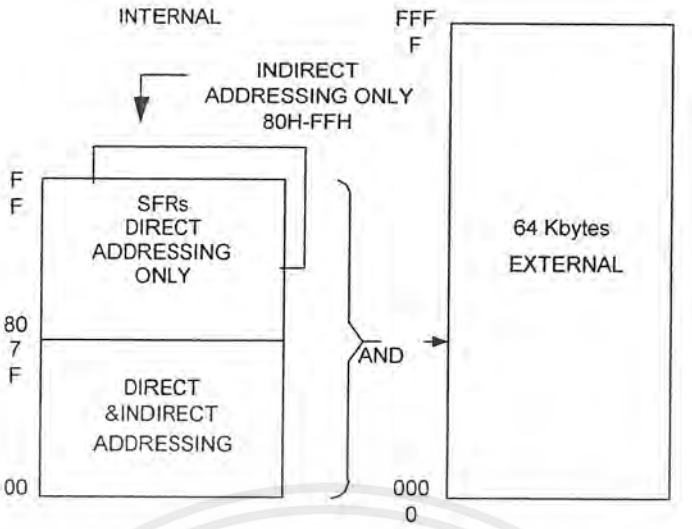
Data Memory (RAM) แบ่งเป็น 2 ส่วนคือหน่วยความจำข้อมูลภายในชิพมีเพียง 128 ไบต์ สำหรับ



รูปที่ 2.18 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8051

เบอร์ 8051 และ 256 ไบต์สำหรับเบอร์ 8052 ขึ้นไปและหน่วยความจำข้อมูลภายนอกชิพมีความจุ 64 กิโลไบต์ ดังแสดงในรูป 2.18 และ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ผังหน่วยความจำสำหรับ Program Memory ของ 8052

บางครั้งอาจสงสัยว่าผังหน่วยความจำสำหรับโปรแกรมและดาต้ามีตำแหน่งซ้อนกันอยู่ซีพียูจะรู้ได้อย่างไรว่าติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูล บริษัทอินเทลได้ออกแบบคำสั่งแยกออกเป็น 3 ส่วนคือ

MOV ใช้ติดต่อกับ RAM ภายใน

MOVC ใช้ติดต่อ Program Memory

MOVX ใช้ติดต่อ Data Memory ภายนอกชิพ โดยระบุตำแหน่งผ่าน DPTR และ PC

* ชิพเบอร์ 8052 จะมีพื้นที่บริเวณ 80h-FFh ซึ่งถ้าจะเขียนอ่านข้อมูล ณ บริเวณนี้จะเข้าถึงข้อมูลโดยทางอ้อมเท่านั้น ดังผังหน่วยความจำดังรูป 2.19 *

พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยทางอ้อมเท่านั้น (Indirect Address Area)

พื้นที่หน่วยความจำบริเวณ (80h-FFh) ตามรูปที่ 2.19 เป็นพื้นที่ซ้อนกันอยู่อย่างละ 128 ไบต์ ส่วนแรกจะเป็น SFR แอดเดรสและ Indirect Address Area ดังนั้นผู้เขียนถ้าจะติดต่อกับ SFR จะต้องใช้คำสั่งแบบเข้าถึงข้อมูลโดยตรงเท่านั้น (Direct Address Area) ส่วนพื้นที่อีกส่วนหนึ่งจะเข้าถึงข้อมูลแบบทางอ้อมเท่านั้น (Indirect Address Area) ส่วนตำแหน่ง (00h-7Fh) จะเข้าถึงข้อมูลได้ทั้ง 2 แบบ

พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยทางตรงและทางอ้อม (Direct and Indirect Address Area)

พื้นที่ 128 ไบต์ล่างสุดจะเป็น 3 ส่วนดังรูป 2.20

1.รีจีสเตอร์ แบงค์ (Register Bank 0-3)

ตั้งแต่ตำแหน่ง (00h-1Fh) จะเป็นส่วนของรีจีสเตอร์แบงค์ (0-3) โดยแบ่งเป็นแบงค์ละ ไบต์ (แต่ละแบงค์จะมีรีจีสเตอร์ R0,R1,R2,R3,R4,R5,R6,R7) ถ้าซีพียูทำงานอยู่ที่แบงค์ 3 เมื่อถูกรีเซ็ตจะกลับมาทำงานที่แบงค์ 0 เสมอ และ SP จะมาเริ่มต้นที่ตำแหน่ง 07h ทันที

2.บริเวณหน่วยความจำที่ใช้ตำแหน่งอ่านเขียนทีละบิตได้ (Bit Address Area)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นที่ตั้งแต่แอดเดรส (20h-7Fh) จำนวน 16 ไบต์หรือแบ่งเป็นบิตจะได้เท่ากับ 128 บิต ซึ่งตำแหน่งบิตมีดังนี้ 00,01,02,03,04,05,06,07 จนถึง 7Fh

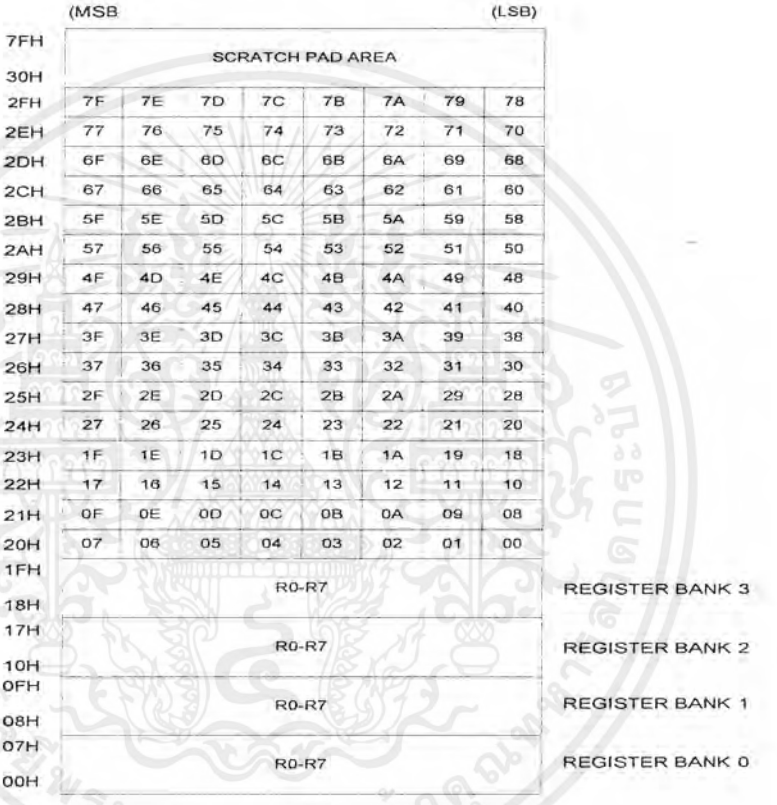
เช่น บิต 00 ก็คือ D0 ของหน่วยความจำตำแหน่ง 20h

บิต 01 ก็คือ D1 ของหน่วยความจำตำแหน่ง 20h

ดูรูปที่ 2.20 ประกอบเช่นต้องการเซ็ตบิต 00 ต้องเขียนคำสั่งว่า SET 00h

3.บริเวณหน่วยความจำที่ใช้งานทั่วไป (Scrath Pad Area)

พื้นที่ตั้งแต่ (30h-7Fh) จะเขียนข้อมูลได้ที่ละไบต์เท่านั้นไม่สามารถใช้คำสั่งเกี่ยวกับบิตได้ ถ้าย้ายเนื้อที่สแตคมายังบริเวณนี้โปรดระวังในการเขียนข้อมูลทับสแตค



รูปที่ 2.20 128 ไบต์ของ RAM ที่เข้าถึงข้อมูลแบบทางตรงและทางอ้อม

Special function Register (SFR) มีรายละเอียดดังตารางที่ 2.1

ตารางที่ 2.1 แสดงสัญลักษณ์ชื่อและตำแหน่งต่าง ๆ ที่มีอยู่ใน **SFR**

SYMBOL	NAME	ADDRESS
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DPTR	Data Pointer 2 Byte	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

2.3 การแปลงสัญญาณอนาลอกเป็นดิจิตอล(Analog to Digital Conversion)

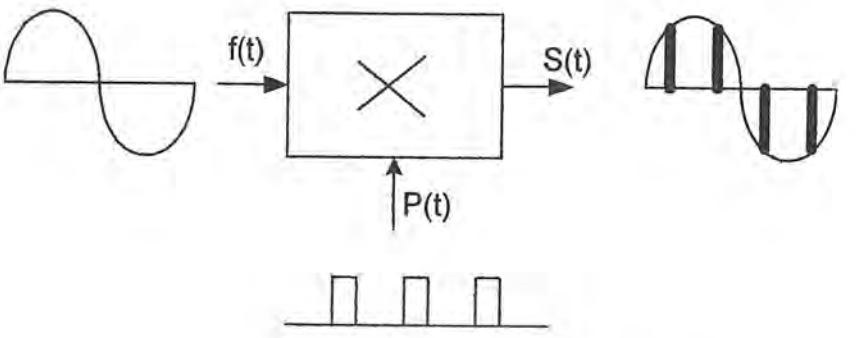
การแปลงสัญญาณอนาลอกเป็นดิจิตอลประกอบด้วยขบวนการสำคัญ 3 ส่วน คือ

1. การแซมปลิ้ง(Sampling)
2. การควอนไตซิ่ง(Quantizing)
3. การเข้ารหัส(Encoding)

ซึ่งส่วนประกอบที่สำคัญที่สุดคือการแซมปลิ้งเพราะความผิดพลาดของสัญญาณดิจิตอลที่แปลงมาจากสัญญาณอนาลอกนั้นจะมากหรือน้อยขึ้นอยู่กับความสัมพันธ์ของความถี่แซมปลิ้งกับความถี่สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสัญญาณอนาลอก โดยทฤษฎีการแซมปลิ้ง(Sampling Theory)ซึ่งความสัมพันธ์ตามทฤษฎีการแซมปลิ้ง ดูได้จากรูป 2.21



รูปที่ 2.21 แสดงการแซมปลิ้ง

จากรูปจะได้ $S(t)=P(t)*f(t)$

เมื่อดูP(t) ซึ่งเป็นพัลซ์รูปสี่เหลี่ยม ถ้านำมาเขียนสมการฟูเรียร์(Fourier)จะได้

$$P(t)=DC + a_0\cos w_0t + a_1\cos 3w_0t + a_2\cos 5w_0t + \ldots\ldots$$

ซึ่ง P(t)ประกอบไปด้วยความถี่พื้นฐานรวมกับฮาโมนิคที่เป็นเลขคี่ไปจนถึง infinite และถ้าคูณ P(t)ด้วย f(t) จะได้ S(t)

$$S(t)=f(t)*DC + (a_0\cos w_0t)*f(t) +(a_1\cos 3w_0t)*f(t) + \ldots\ldots$$

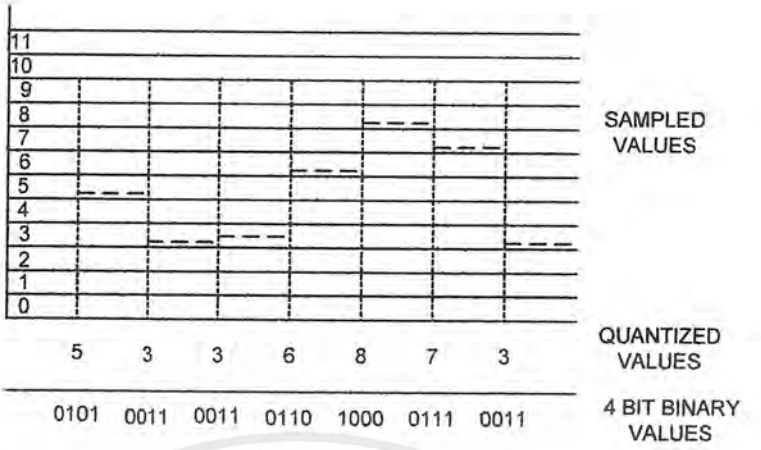
เมื่อพิจารณาดูเทอมที่ 2 จะพบว่ามีรูปแบบเหมือนแอมปลิจูดมอดูเลชั่น(AM)

โดยถ้า $f(t)=B\cos w_mt$

$$f(t)* (a_0\cos w_0t)=(Ba_0/2)\cos(w_0-w_m)t + (Ba_0/2)\cos(w_0+w_m)t$$

ซึ่งความถี่ w_0 ที่ใช้สำหรับการแซมปลิ้งและการดีเทค สัญญาณที่ได้คืนจะได้วงจรกรองความถี่ต่ำ(Low Pass Filter) กรองเอาเฉพาะ f(t)*DC ออกมาเท่านั้น ซึ่งถ้า w_0 ซึ่งถ้า w_0 มีค่าน้อยกว่า 2 เท่าของ w_m แล้วจะทำให้ความถี่ซึ่งเป็นผลต่างของ w_0 - w_m เข้ามาแทรกใน f(t)*DC ด้วย ซึ่งจะมีผลให้สัญญาณที่ดีเทคกลับคืนมามีความผิดพลาด ดังนั้นจึงต้องเลือก W_0 หรือความถี่แซมปลิ้งให้มีค่ามากกว่า 2 เท่าของความถี่สูงสุดของของสัญญาณก่อนที่จะมีการแซมปลิ้ง หรือ w_m ซึ่งสัญญาณที่ได้ออกมาจากภาคแซมปลิ้งนี้เรียกว่าพัลซ์แอมปลิจูดมอดูเลชั่น(Pulse Amplitude Modulation:PCM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

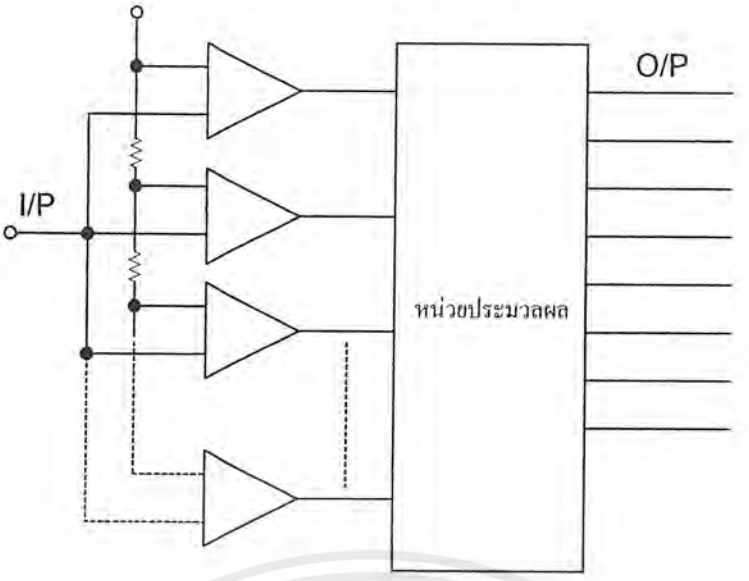


รูปที่ 2.22 แสดงลักษณะสัญญาณของภาคควอนไตซิ่งและการเข้ารหัส

ภาคควอนไตซิ่ง(Quantizing)เป็นการจัดระดับของสัญญาณ PCM ซึ่งอาจมีระดับที่ไม่แน่นอนให้อยู่ในระดับที่แน่นอน ซึ่งในขั้นตอนนี้จะมีความผิดพลาดจากการจัดระดับอยู่ เรียกว่าควอนไตซิ่งเออเรอร์หรือสัญญาณรบกวนควอนไตซิ่ง (Quantizing Error, Quantizing Noise) ซึ่งจะมีค่ามากหรือน้อยขึ้นอยู่กับระดับสัญญาณที่เราจะแบ่ง ซึ่งลักษณะของสัญญาณของภาคควอนไตซิ่งและข้อผิดพลาดของสัญญาณเมื่อได้รับสัญญาณควอนไตซิ่งแล้วก็จะนำไปเข้าวงจรเข้ารหัสเป็นสัญญาณดิจิตอลซึ่งมีค่า 2 ระดับ คือ 0 กับ 1 เท่านั้น ถ้าส่งข้อมูลเป็นแบบ 8 บิตเพราะฉะนั้นจะส่งข้อมูลได้ 2^8 =256 ระดับ โดยกำหนดให้ระดับต่ำของสัญญาณควอนไตซิ่ง(Quantizing Signal)เท่ากับ 1111 1111 ดังนั้นข้อมูลที่ออกมาจะเป็นสัญญาณดิจิตอลที่มีค่าตามระดับตามที่ตั้งไว้

แฟลชอนาลอกทูดิจิตอลคอนเวอร์เตอร์(Flash A/D) คือวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล(A/D Converter)ที่มีความเร็วสูงในการเปลี่ยน เนื่องจากแฟลชเอทูดีจะใช้การโปรแกรมเอาท์พุทเอาไว้ แล้วส่วนวงจรเปรียบเทียบจะใช้ออปแอมป์ที่มีจำนวนเท่ากับจำนวนของสัญญาณเอาท์พุทที่จะเกิดขึ้น เช่นถ้าสัญญาณเอาท์พุทเป็นดิจิตอล 8 บิตจะใช้ออปแอมป์ 2^8 =256 ตัวซึ่งแต่ละตัวจะมีแรงดันอ้างอิงที่เป็นค่าคงที่อยู่ค่าหนึ่งตามระดับสัญญาณดังรูปที่ 2.23

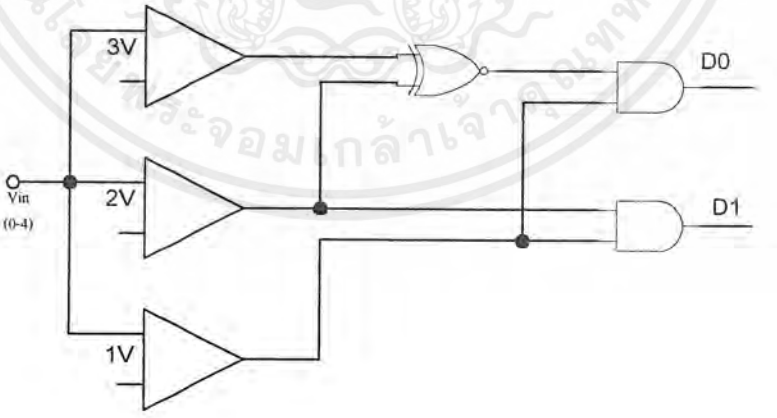
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงวงจร Flash A/D

เมื่อสัญญาณอินพุทเข้ามาจะถูกส่งไปให้กับออปแอมป์ทุกตัวเพื่อเปรียบเทียบแรงดันอ้างอิงของแต่ละตัว ถ้าสัญญาณอินพุทไปตรงกับออปแอมป์ตัวใดก็จะให้สัญญาณเอาท์พุทออกมาส่งไปให้กับวงจรประมวลผล เพื่อจัดหาค่าสัญญาณดิจิตอล สัญญาณเอาท์พุทให้ได้ตามค่าของสัญญาณอินพุทที่ส่งเข้ามาซึ่งการเปลี่ยนสัญญาณในรูปนี้ไม่ต้องใช้วงจรนับ แล้วป้อนกลับมาเปรียบเทียบทีละค่า จึงทำให้ความเร็วในการเปลี่ยนสัญญาณสูงกว่าแบบแรกมาก จึงเหมาะที่จะนำมาใช้กับสัญญาณภาพซึ่งมีความถี่สูง

บางครั้งเรียกแฟลชเอทูดีว่าเป็น Open Loop Conversion เนื่องจาก ไม่มีสัญญาณป้อนกลับไปยังอินพุท ดังรูปที่ 2.24



รูปที่ 2.24 แสดงหลักการของ Open Loop Conversion

จากรูปที่ 2.24 จะใช้ลอจิกเกทเป็นวงจรเปลี่ยนระดับของสัญญาณอินพุทให้เป็นสัญญาณดิจิตอล โดยเริ่มจากเมื่อมีอินพุท 0 Volt เข้าจะทำให้เอาท์พุททุกตัวของคอมพาเรเตอร์ (Comparator) เป็น 0 หมด และเมื่อผ่านเอ็กคูซีฟนอร์เกท(Exclusive – NOR Gate) จะทำให้เอาท์พุทเป็น 1 ซึ่งจะไปเข้า AND Gate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์1 เป็น 0 ทำให้ D_0=0และ AND Gate เบอร์ 2 จะมีเอาท์พุทเป็น 0 ดังนั้น D_1=0 เพราะฉะนั้นระดับ 0 โวลท์เอาท์พุทของเอทูดีจะเท่ากับ 0 0 เมื่อ Vin=1 โวลท์จะทำให้เอาท์พุทของคอมพาเรเตอร์ตัวที่1เป็น 1 ไป AND กับเอาท์พุทของเอ็กคลูซีฟนอร์เกทซึ่งเป็น 1 ทำให้ D_0=1 ส่วนD_1ได้จากการ AND กันของเอาท์พุทคอมพาเรเตอร์ตัวที่1กับตัวที่2 เป็น 1 0 ดังนั้นเอาท์พุท D_1=0 เพราะฉะนั้น Vin=2 โวลท์ จะทำให้เอาท์พุทของคอมพาเรเตอร์ตัวที่ 1 กับ 2 เป็น 1 ทำให้เอาท์พุทของD_1=1และ D_2=0 เพราะฉะนั้นเอาท์พุทของเอทูดีจะเท่ากับ 1 0 แล้วเมื่อ Vin= 3 โวลท์จะทำให้เอาท์พุทของคอมพาเรเตอร์ทุกตัวเป็น 1 ดังนั้นเมื่อผ่านลอจิกเกทจะได้เอาท์พุทเป็น 11

ข้อดีของวงจรแฟลชเอทูดีคือมีความสามารถในการเปลี่ยนสัญญาณได้เร็วมาก แต่ข้อเสียก็คือจะต้องใช้จำนวนคอมพาเรเตอร์มากเป็น 2 เท่าเมื่อต้องการบิตเพิ่ม 1 บิต หรือถ้าเป็นสมการจำนวนออปแอมป์เท่ากับ 2^N-1 ตัวโดย N คือจำนวนบิต ดังนั้นถ้าเราใช้สัญญาณดิจิตอล 8 บิต เราจะต้องใช้คอมพาเรเตอร์ 256 ตัว ซึ่งทำให้อุปกรณ์มีราคาสูงมาก

2.4 พอร์ตขนาน

พอร์ตขนานหรือพอร์ตเครื่องพิมพ์ของคอมพิวเตอร์ มีขั้วต่อเป็นคอนเนกเตอร์ แบบ D ขนาด 25 ขา (DB 25)และมีหน้าที่การทำงานดังตารางที่ 2.2 ภายในพอร์ตขนานประกอบด้วยรีจิสเตอร์พื้นฐานที่ใช้ในการรับส่งข้อมูล 3 ตัวคือ รีจิสเตอร์ DATA ,STATUS และ CONTROL โดยแอดเดรสของรีจิสเตอร์ของทั้งสามจะมีตำแหน่งเรียงกันไปตามลำดับขึ้นอยู่กับแอดเดรสของพอร์ตขนานเช่นที่ พอร์ตขนาน LPT1 แอดเดรสของรีจิสเตอร์ DATA อยู่ที่ &H378 (&H เป็นตัวอักษรที่ แสดงว่าข้อมูลนี้เป็นข้อมูลฐานสิบหก) ในขณะที่แอดเดรสของรีจิสเตอร์ STATUS จะอยู่ที่ &H379 และแอดเดรสของรีจิสเตอร์ CONTROL อยู่ที่ &H37A โดยความสัมพันธ์ระหว่างแอดเดรสพอร์ตขนานและแอดเดรสของรีจิสเตอร์พื้นฐานทั้งสามตัวดังแสดงในตารางที่ 2 .2

ตารางที่ 2.2 หน้าที่และการทำงานของขาต่างๆ ของพอร์ตขนาน

ขา	หน้าที่	พอร์ต		ทิศทาง
		ชื่อ	บิต	เอาท์พุท
1	/Strobe	Control	/C0	เอาท์พุท
2	Data บิต 0	Data	D0	เอาท์พุท
3	Data บิต 1	Data	D1	เอาท์พุท
4	Data บิต 2	Data	D2	เอาท์พุท
5	Data บิต 3	Data	D3	เอาท์พุท
6	Data บิต 4	Data	D4	เอาท์พุท
7	Data บิต 5	Data	D5	เอาท์พุท
8	Data บิต 6	Data	D6	เอาท์พุท
9	Data บิต 7	Data	D7	เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10	Acknowledge	Status	S6	อินพุท
11	/Busy	Status	/S7	อินพุท
12	Out of Paper	Status	S5	อินพุท
13	Select	Status	S4	อินพุท
14	/Line Feed	Control	/C1	เอาท์พุท
15	Error	Status	S3	อินพุท
16	Initial	Control	C2	เอาท์พุท
17	/Select In	Control	/C3	เอาท์พุท
18-25	Ground			

ตำแหน่งของพอร์ตขนานส่วนใหญ่ จะมีตำแหน่งพอร์ตเริ่มต้นอยู่ที่ &H378 แต่ก็ไม่ใช่ ตำแหน่งสำหรับคอมพิวเตอร์ทุกเครื่อง ดังนั้นวิธีง่ายๆ เพื่อตรวจสอบตำแหน่งของพอร์ตขนานที่ใช้งานอยู่ สามารถทำได้โดยการใช้โปรแกรม DEBUG แสดงค่าหน่วยความจำตำแหน่ง 0040:08 ซึ่งเป็นที่เก็บค่าแอดเดรสของพอร์ตขนาน LPT1 ,LPT2, และ LPT3 เอาไว้ดังตัวอย่าง

```
C:\dos\debug
-d 0040:80  78 03 78 02 00 00 00
```

จากตัวอย่างข้างต้นพอร์ต LPT1 มีแอดเดรสอยู่ที่ &H378 ส่วนพอร์ต LPT2 มีแอดเดรสอยู่ที่ &H278 สำหรับ LPT3 และ LPT4 นั้นไม่ได้กำหนดเอาไว้ นอกจากการใช้โปรแกรม MSD.EXE ของ DOS เพื่อตรวจสอบไว้อีกด้วย

ตารางที่ 2.3 แสดงแอดเดรสของรีจิสเตอร์ของพอร์ตขนาน

พอร์ตขนาน	แอดเดรสของ DATA	แอดเดรสของ STATUS	แอดเดรสของ CONTROL
LPT1	&H378	&H379	&H37A
LPT2	&H3BC	&H3BD	&H3BE
LPT3	&H278	&H279	&H37A

จากตารางที่ 2.2 สามารถแบ่งคุณสมบัติของพอร์ตขนานได้ 2 รูปแบบคือ พอร์ตที่หน้าที่ เอาท์พุท และพอร์ตที่ทำหน้าที่อินพุท ดังมีรายละเอียดดังต่อไปนี้

พอร์ตเอาท์พุท

พอร์ตที่ทำหน้าที่เป็นพอร์ตเอาท์พุทคือ พอร์ตDATA และ พอร์ต CONTROL ตำแหน่งพอร์ต D0-D7 สามารถใช้งานเป็นเอาท์พุท ได้ทั้งหมด ส่วนพอร์ต CONTROL มี บิต ที่ทำหน้าที่เป็นเอาท์พุท เพียง 4 บิต คือ C0-C3 ส่วนตำแหน่งอื่นๆ ไม่ได้ใช้งานหรือถูกสงวนไว้ใช้กับงานอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนข้อมูลไปยังรีจิสเตอร์ DATA นั้น ข้อมูลที่ส่งออกไปกับสถานะลอจิกที่ขาพอร์ต DATA จะตรงกัน แต่สำหรับพอร์ต CONTROL นั้น บิต C0,C1 และ C3 จะกลับสถานะโดยเมื่อป้อนค่าใดค่าหนึ่งไปยังรีจิสเตอร์ CONTROL ที่พอร์ต CONTROL จะทำการแปลงข้อมูลที่ อยู่ในบิต C0,C1 และ C3 ให้มีสถานะตรงกันข้าม ดังนั้นเมื่อจะต้องส่งข้อมูลไปยังพอร์ต CONTROL จะต้องมีการกลับค่าของของข้อมูลในบิต C0,C1และ C3 ก่อนเสมอ

พอร์ตอินพุท

พอร์ตที่ทำหน้าที่เป็นพอร์ตอินพุท คือ พอร์ต STATUS มีบิตที่ใช้งานเป็นอินพุทเพื่อรับข้อมูลภายนอก 5 บิต คือ S3-S7 โดยบิต 7 นั้นจะกลับสถานะอยู่ ดังนั้นการอ่านข้อมูลในบิต 7 จะได้ค่าออกมาเป็นค่าตรงกันข้าม ตำแหน่งแอดเดรสของพอร์ต STATUS จะอยู่ถัดจากแอดเดรสของพอร์ต DATA หนึ่งตำแหน่ง ดังนั้นเพื่อเป็นการง่ายต่อการทำความเข้าใจ การเขียนโปรแกรมไปยังพอร์ต STATUS จึงนิยมระบุตำแหน่งแอดเดรสเป็น BaseAddr+1

พอร์ตคอมพิวเตอร์รุ่นใหม่

สำหรับคอมพิวเตอร์รุ่นใหม่ๆ นั้นพอร์ต DATA สามารถใช้เป็นอินพุทได้ด้วย โดยจะมีบิตที่ทำหน้าที่กำหนดทิศทางที่พอร์ต CONTROL บิต 5 (C5) โดยถ้าบิตนี้เป็น "0" จะเป็นการกำหนดให้พอร์ตนี้เป็นเอาท์พุท ถ้าบิตนี้เป็น "1" พอร์ตนี้จะเป็นอินพุท ทำให้สามารถอ่านสถานะลอจิกจากจากภายนอกได้

2.5 RAS (Remote Access Service)

RAS เป็นคำย่อมาจากคำว่า Remote Access Service ซึ่งหมายความถึง ระบบที่ให้บริการใช้งานจากระยะไกล ระบบนี้เป็นระบบสื่อสารที่มีอยู่ในซีดีรอมของแผ่นติดตั้ง Windows NT 4.0 อยู่แล้ว คุณอาจติดตั้งโปรแกรมนี้ พร้อมๆกับการติดตั้ง Windows NT 4.0 ตั้งแต่แรก หรืออาจติดตั้งภายหลังต่างหากก็ได้

RAS สามารถใช้งานกับระบบสื่อสารหลายแบบ ไม่ว่าจะเป็นโทรศัพท์ธรรมดา,ISDN หรือ WAN สามารถใช้งานผ่านพอร์ตอนุกรมแบบธรรมดา(COM1,COM2) และยังสามารถรองรับการขยายได้ถึง 256 พอร์ต ถ้าติดตั้งเน็ตเวิร์คเทอร์มินอล(Network Terminal) ทำให้สามารถรองรับผู้ใช้งานได้ จำนวนมากพร้อมๆกันได้

ถ้าที่ตั้งของเซอร์ฟเวอร์ของพื้นที่ของคุณเป็นพื้นที่สามารถใช้บริการ ISDN ได้ คุณอาจเลือกใช้ ISDN เพื่อเพิ่มความเร็วในการสื่อสาร จะช่วยให้ผู้ใช้สามารถใช้งานได้ในความเร็วสูงกว่าสายโทรศัพท์

ความสามารถที่สำคัญของ Windows NT คือการเพิ่มความเร็วในการสื่อสาร โดยใช้การเชื่อมต่อแบบมัลติลิงค์(Multilink) คุณสมบัตินี้จะช่วยให้ผู้ใช้สามารถใช้โทรศัพท์หลายๆเส้น รวมเข้าเป็นเส้นเดียวในการติดต่อ ช่วยให้ความเร็วสูงขึ้น แต่ผู้ใช้บริการจะต้องมีโทรศัพท์มากกว่าหนึ่งเส้นด้วยเช่นกัน

คุณสามารถใช้วินโดวส์ได้ทุกรุ่นนำมาสร้างเป็นไคลเอนส์ของ RAS ไม่ว่าจะเป็น Windows3.x ,Windows for Workgroup ,Windows 95 หรือ Windows NT Workstation และยังสนับสนุนไคลเอนส์ระบบอื่นๆ ที่สามารถใช้โปรโตรคอล PPP ได้อีกด้วย ยกเว้นไคลเอนส์แบบหนึ่งที่ไม่สนับสนุนคือ ARAP (Apple Talk Remote Access Protocol) ซึ่งเป็นโปรโตคอลสื่อสารในเครื่องแมคอินทอช ดังนั้นถ้าคุณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการใช้เครื่องแมคอินทอช เป็นไคลเอนส์แบบ RAS แล้วละก็ คุณจำเป็นต้องใช้ MACTCP ซึ่งมีเฉพาะใน System 7.5 O/S เท่านั้น

การเลือกวิธีการสื่อสารทางไกล

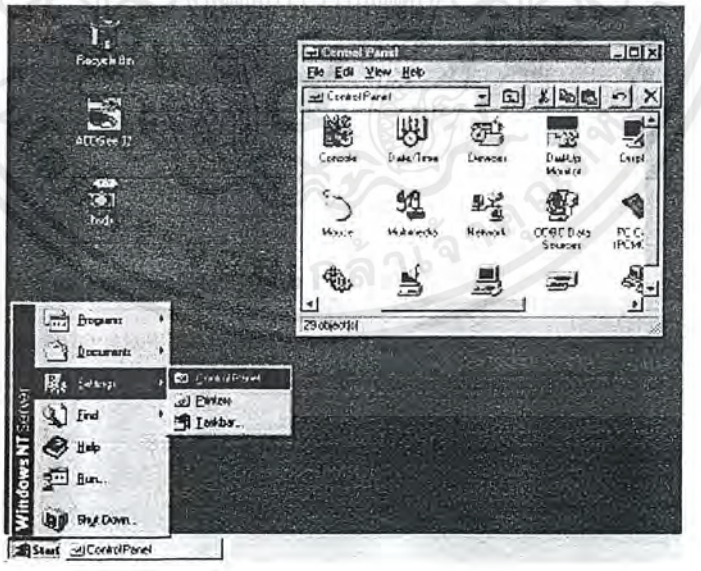
สายสื่อสารที่นิยมมากที่สุดสำหรับ RAS คือสายโทรศัพท์ เนื่องจากมีความง่ายและมีราคาประหยัดเหมาะสมกับการใช้งาน ซึ่งถ้าคุณต้องการติดตั้งสายสื่อสารสำหรับผู้ใช้หนึ่งสายหรือสองสาย คุณสามารถเลือกใช้พอร์ตอนุกรม COM1 และ COM2 ได้ทันที แต่การทำเช่นนี้อาจทำให้ CPU ของเครื่องเซอร์ฟเวอร์ของคุณช้าลง เนื่องจากพอร์ตอนุกรมนั้นใช้สัญญาณอินเตอร์รัพท์ในการติดต่อ จึงรบกวนต่อการทำงานของ CPU โดยตรง

ดังนั้นถ้าคุณต้องการจะจัดทำระบบ RAS ที่มีความสามารถสูง สามารถรองรับผู้ใช้ได้หลายคนพร้อมๆกันโดยไม่จำเป็นต้องรบกวนการทำงานของ CPU มาก คุณควรจัดหาเน็ตเวอร์คเทอมินอล (Network Terminal) สำหรับขยายช่องโมเด็ม ข้อดีของเน็ตเวอร์คเทอมินอล คือสามารถจะต่อสายสัญญาณได้มากขึ้นและอุปกรณ์นี้ยังมีบัฟเฟอร์(Buffer) ซึ่งจะช่วย CPU ทำงานได้ราบรื่นขึ้นโดยไม่ถูกสัญญาณอินเตอร์รัพท์รบกวนบ่อยๆ

2.5.1 การติดตั้ง Remote Access Service

ถ้าคุณยังไม่ได้ติดตั้ง Remote Access Service ในขณะที่ติดตั้ง Windows NT Server ตั้งแต่แรก คุณอาจติดตั้ง RAS ลงไปใหม่ โดยใส่ซีดีสำหรับติดตั้ง Windows NT 4.0 ลงในซีดีรอมไดรว์ จากนั้นจึงติดตั้งจากขั้นตอนการติดตั้ง Remote Access Service ดังต่อไปนี้

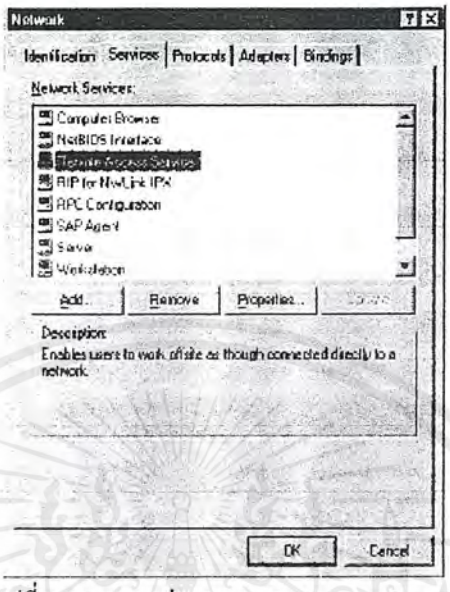
1. คลิกปุ่ม Start แล้วเลือก Start>Setting>Control Panel เพื่อเปิด Control Panel
2. ดับเบิ้ลคลิก ไอคอน Network



รูปที่ 2.25 แสดงหน้าต่างเพื่อเข้าสู่การติดตั้ง Remote Access service

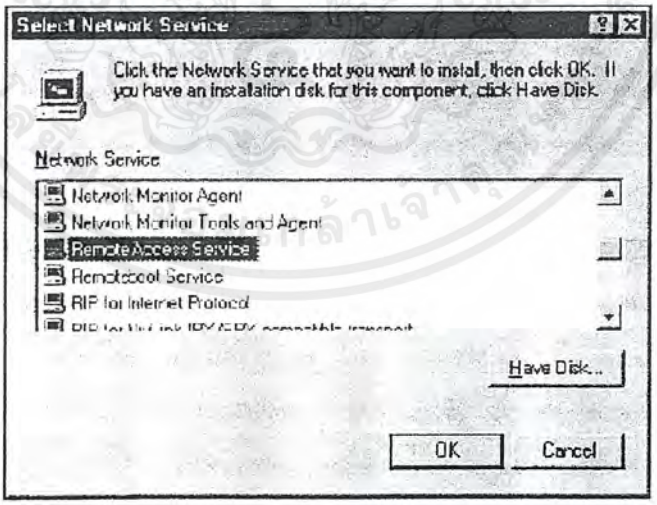
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าคุณพบคำว่า Remote Access Service อยู่ในช่องรายการ Network Service: แสดงว่าคุณได้ติดตั้ง RAS ไว้เรียบร้อยแล้ว แต่ถ้าคุณไม่พบให้คลิกปุ่ม ADD
4. แดร็กเมาส์ ลากแถบเลื่อนขึ้นลง จนพบคำว่า Remote Access Service แล้วคลิกเลือก



รูปที่ 2.26 แสดงช่องรายการ Network Service

5. คลิกปุ่ม OK จะทำให้ RAS ติดตั้งจากซีดีรอมที่คุณเตรียมไว้ทันที

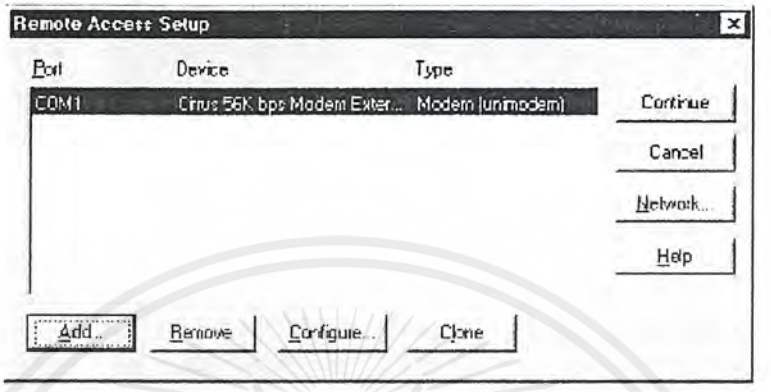


รูปที่ 2.27 แสดง รายการ Remote Access Service ในช่องรายการ Network Service

6. เลือกพอร์ตที่มีอุปกรณ์สื่อสารที่คุณติดตั้งไว้ล่วงหน้า ซึ่งถ้าคุณยังไม่ได้ติดตั้งโมเด็มให้คลิกปุ่ม Install Modem เพื่อทำการเลือกรุ่นโมเด็มเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. คลิกปุ่ม OK เพื่อเลือกว่าจะใช้ พอร์ตนี้
8. คุณจะพบว่า พอร์ตที่คุณเลือกถูกแสดงในช่องรายการ
9. ถ้าคุณต้องการเพิ่มอุปกรณ์สื่อสารอีก ให้คลิกปุ่ม ADD แต่ถ้าต้องการจบการติดตั้ง ให้คลิกปุ่ม Continue



รูปที่ 2.28 แสดงรายการอุปกรณ์สื่อสารที่ ใช้กับระบบ Remote Access Service

เมื่อมาถึงขั้นตอนนี้ คุณสามารถทำการปรับแต่ง RAS ได้ทันที โดยศึกษาวิธีการจากหัวข้อต่อไปนี้

การปรับแต่ง Remote Access Service

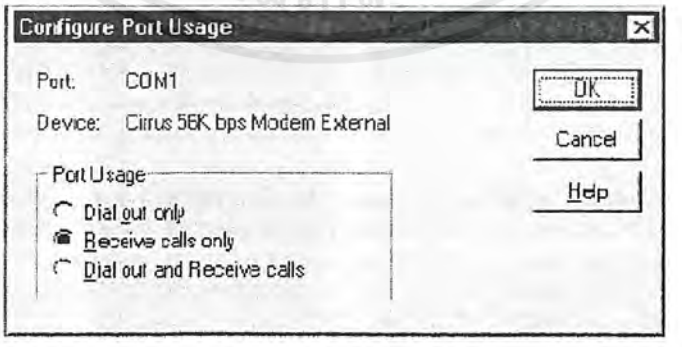
หลังจากผ่านขั้นตอนในหัวข้อก่อน คุณสามารถปรับแต่ง RAS ได้ทันทีดังต่อไปนี้

เพิ่มและลบอุปกรณ์สื่อสาร

เมื่อคุณเข้ามาสู่หน้าต่างของ Remote Access Setup แล้วคุณสามารถที่จะเพิ่มอุปกรณ์สื่อสารได้ โดยคลิกปุ่ม Add หรือลบอุปกรณ์สื่อสาร โดยคลิกปุ่ม Remove ตามที่ต้องการ

กำหนดการใช้งานพอร์ตสื่อสาร

คุณสามารถปรับให้ระบบ RAS ของคุณเป็นระบบที่รับสายเพียงอย่างเดียว หรือเป็นระบบที่สามารถโทรออกภายนอกได้ โดยการคลิกที่ปุ่ม Configure..



รูปที่ 2.29 แสดงหน้าต่างที่ใช้ปรับแต่งคุณสมบัติของอุปกรณ์สื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสามารถเลือกใช้งานพอร์ตได้ 3 วิธีดังนี้

►คลิก Dial Out Only เพื่อกำหนดให้ระบบ RAS เป็นระบบที่ต่อภายนอกเพียงอย่างเดียว

►คลิก Receive Calls Only เพื่อกำหนดให้ระบบ RAS เป็นระบบที่รับการโทรเข้าเพียงอย่างเดียว

►คลิก Dial Out and Receive Calls เพื่อกำหนดให้ระบบ RAS เป็นระบบที่ทั้งโทรออก(Dial Out)และรับการโทรเข้า(Receive Calls)ได้

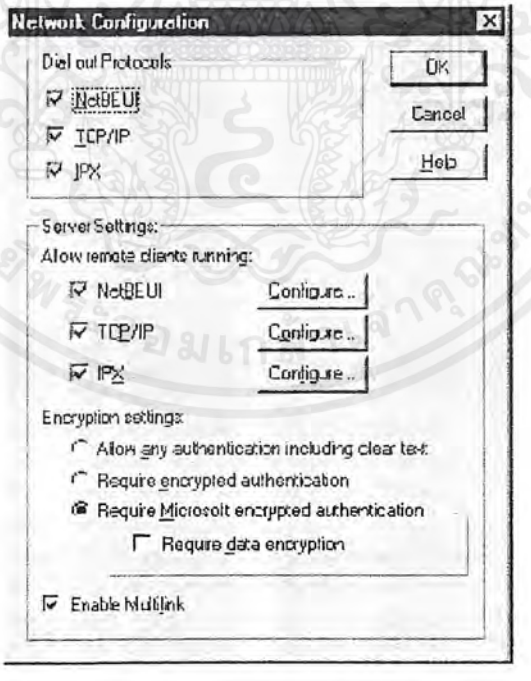
กำหนดรูปแบบเครือข่ายการติดต่อให้กับ Remote Access Service

เนื่องจากไคลเอนส์แต่ละชนิดใช้โปรโตคอลสื่อสารที่ไม่เหมือนกัน คุณจึงต้องปรับแต่ง RAS ให้รองรับโปรโตคอลที่ต้องการด้วยตัวเอง

จากหน้าต่าง Remote Access Service ให้คลิกปุ่ม Network... คุณจะพบกับหน้าต่างของการปรับแต่งเครือข่าย

กรอบ Dial – Out Protocols

ถ้าคุณเลือกให้ RAS มีความสามารถในการโทรออก คุณจะสามารถคลิกเพื่อเลือกโปรโตรคอลของไคลเอนส์ที่ต้องการให้ระบบ RAS สามารถติดต่อไปได้ โดยคุณสามารถเลือกที่จะใช้โปรโตคอลพร้อมๆกันได้ด้วย โดยการคลิกเลือกหลาย ๆโปรโตคอล



รูปที่ 2.30 แสดงกรอบ Server Setting ที่ใช้ปรับแต่งโปรโตคอลของ RAS

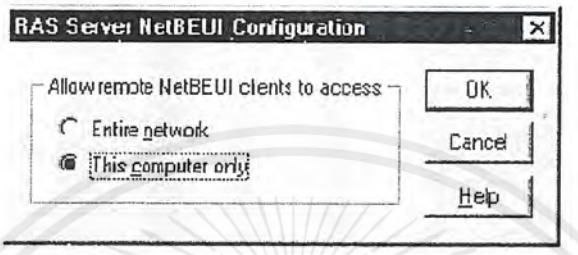
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรอบ Server Setting

สำหรับการปรับแต่งโปรโตคอลของ RAS เอง คุณสามารถปรับแต่งได้ในกรอบนี้ โดยคลิกเลือกโปรโตคอลที่ต้องการ ดังนี้

▶NetBUEI

ถ้าคุณเลือกใช้โปรโตคอลชนิดนี้ คุณควรคลิกปุ่ม Configure...เพื่อปรับแต่งความสามารถในการเข้าถึง Server ของลูกข่าย



รูปที่ 2.31 แสดงกรอบการปรับแต่งโปรโตคอล NetBUEI

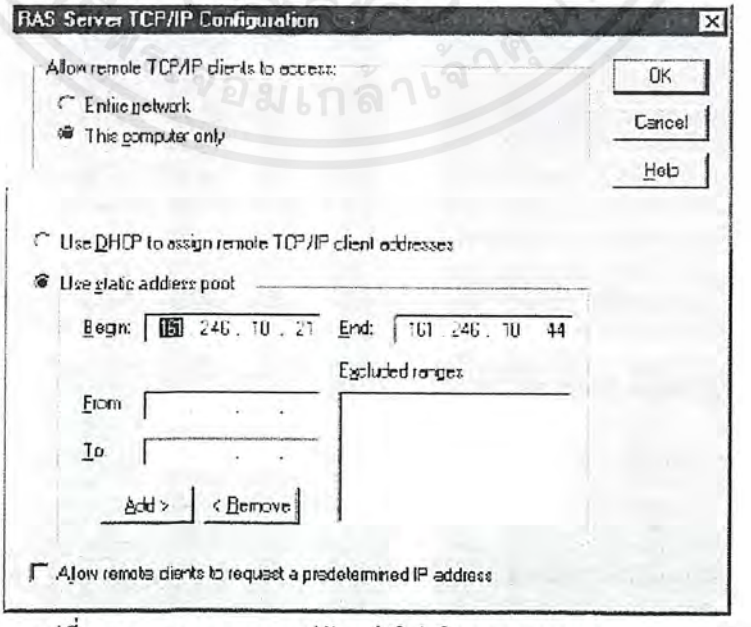
Entire Network อนุญาตให้เข้าถึงเครือข่ายทั้งหมด

This Computer Only อนุญาตให้เข้าถึงเฉพาะคอมพิวเตอร์เครื่องนี้เท่านั้น

เมื่อเลือกเสร็จ ให้คลิกปุ่ม OK เพื่อกลับสู่การปรับแต่งต่อไป

▶TCP/IP

หากคุณเลือกโปรโตคอล TCP/IP ควรจะเลือก TCP/IP แล้วคลิกปุ่ม Configure... เพื่อกำหนด IP Address ของไคลเอนส์ที่สามารถเข้าถึงเซอร์ฟเวอร์ได้



รูปที่ 2.32 แสดงกรอบการปรับแต่งโปรโตคอล TCP/IP

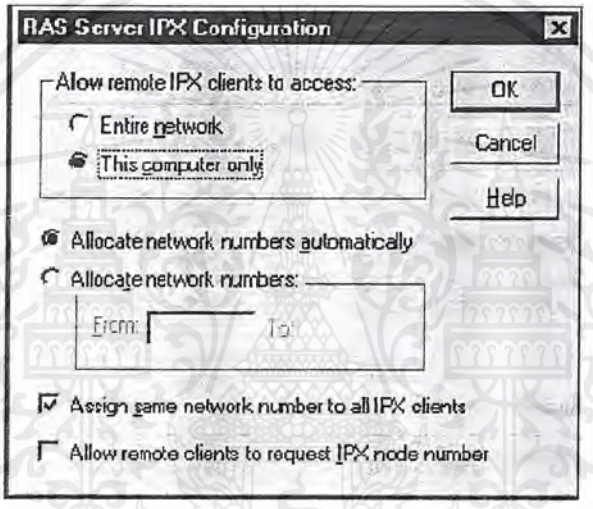
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าคุณได้ปรับแต่ง DHCP Server ไว้แล้ว คุณสามารถคลิกเลือกหัวข้อ Use DHCP to assign remote TCP/IP address client address เพื่อให้เซอร์ฟเวอร์กำหนด IP Address ที่เหมาะสมให้กับไคลเอนส์โดยอัตโนมัติ

แต่ถ้าไคลเอนส์ที่จะมาเชื่อมต่อมี IP Address แบบถาวร คุณสามารถเลือกช่วงของ IP Address ที่ยอมให้เข้าถึงเซอร์ฟเวอร์ของคุณ โดยการกรอกช่วงในช่อง Begin: และ End: แต่ถ้าคุณต้องการป้องกัน IP address บางช่วงไม่ให้เข้าถึงเซอร์ฟเวอร์ ให้คุณกรอกช่วงของ IP Address นั้นในช่อง From: และ To: จากนั้นให้คลิกปุ่ม Add>

►IPX

คุณสามารถเลือกโปรโตคอล IPX โดยการคลิกปุ่ม Configure... เพื่อกำหนดคุณสมบัติ ต่างๆ



รูปที่ 2.33 แสดงกรอบการปรับแต่งโปรโตคอล IPX

เลือก allocate network numbers automatically เพื่อกำหนดให้เซอร์ฟเวอร์ทำหน้าที่จัดการ Network Number อัตโนมัติ

เลือก assign same network number to all IPX clients จะช่วยลดขนาดของ RIP(Routing Information Protocol) ในเครือข่ายได้ โดยทุกๆไคลเอนส์จะถูกกำหนดเป็น Network address เดียวกันหมด

เลือก Allow remote clients request IPX node number เพื่อกำหนดให้ไคลเอนส์สามารถร้องขอเซอร์ฟเวอร์ให้กำหนด node number ให้กับไคลเอนส์ ทั้งนี้เพื่อให้เกิดความปลอดภัยกับเครือข่ายเนื่องจากถ้ายอมให้ด้านไคลเอนส์สามารถกำหนด node number ได้ด้วยตนเอง จะเป็นการเปิดโอกาสให้เกิดการปลอม node number โดย Hacker ได้

►การรักษาความปลอดภัยในการสื่อสารด้วยการเข้ารหัส(Encryption Setting)

การสื่อสารระหว่างไคลเอนส์กับเซอร์ฟเวอร์จะมีการรับส่งข้อมูลและพาสเวิร์ด ข้อมูลเหล่านี้อาจเป็นข้อมูลที่ลับที่คุณไม่ต้องการให้มีผู้ลักลอบดักสัญญาณไปใช้ประโยชน์ คุณจึงควรเข้ารหัส (Encrypt) ข้อมูลเหล่านั้น เพื่อปกป้องข้อมูลให้พ้นจากมือของผู้ไม่ประสงค์ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Allow any authentication including clear text

ระดับนี้ไม่รักษาความปลอดภัย เนื่องจากจะส่งทุกอย่าง (รวมทั้งพาสเวิร์ด) ในรูปของอักษรธรรมดา เหมาะกับการสื่อสารกับไคลเอนส์ที่ไม่ใช่ Windows เช่น Unix และ Macintosh

Request encrypted authentication

ระดับนี้เป็นระดับที่รักษาความปลอดภัยของพาสเวิร์ดได้ คุณจึงควรเลือกระดับนี้เป็นอย่างน้อย

Request Microsoft encrypted authentication

ระดับนี้เหมาะสำหรับระบบที่มีไคลเอนส์เป็น Windows เท่านั้น ระดับนี้มีความพิเศษคือ นอกจากจะเข้ารหัสพาสเวิร์ดแล้ว คุณสามารถจะเลือกเข้ารหัสข้อมูล(Data) ได้อีกด้วย โดยการคลิกที่ Request data encryption

ถึงแม้ว่าการเข้ารหัสข้อมูลจะให้ความปลอดภัยสูง แต่คุณก็ควรคำนึงถึงโหลดในการทำการเข้ารหัสจะมีผลให้เซอร์ฟเวอร์ทำงานช้าลงด้วย ดังนั้นคุณจึงควรพิจารณาความเหมาะสมระหว่างความปลอดภัยและความเร็วก่อนที่จะเลือกใช้การเข้ารหัสระดับใด

การใช้ Multilink

คลิกที่ตัวเลือก Enable Multilink จะทำให้คุณสามารถเพิ่มความเร็วในการสื่อสาร โดยรวมเอาสายสัญญาณหลายๆเส้นมาสื่อสารในช่องเดียวกัน

การสื่อสารแบบนี้เดิมใช้กับระบบ ISDN มีชื่อเรียกว่า Multilink PPP(Point to Point Protocols) แต่ Windows NT 4.0 ได้นำเทคนิคนี้มาใช้กับสายโทรศัพท์และ X.25 ซึ่งใช้ในเครือข่าย WAN ด้วย

สิ่งจำเป็นอย่างยิ่งในการสื่อสารแบบ Multilink คือคุณต้องมีสารสื่อสารอย่างน้อย 2 เส้นทั้งด้านเซอร์ฟเวอร์และด้านไคลเอนส์

จบการปรับแต่ง

เมื่อคุณปรับแต่งจนเป็นที่พอใจแล้ว ให้คลิกปุ่ม OK เพื่อปิดหน้าต่าง Network Configuration จากนั้นคลิกปุ่ม Close เพื่อปิดหน้าต่าง Network

จากนั้น คลิกปุ่ม Start แล้ว เลือกคำสั่ง Shut Down เพื่อปิดเซอร์ฟเวอร์ จากนั้นให้คุณบู๊ตเครื่องใหม่ จึงจะสามารถใช้ RAS ที่ติดตั้งและปรับแต่งใหม่นี้ได้ตามต้องการ

2.5.2 การติดตั้งไคลเอนส์ RAS สำหรับ Windows 95

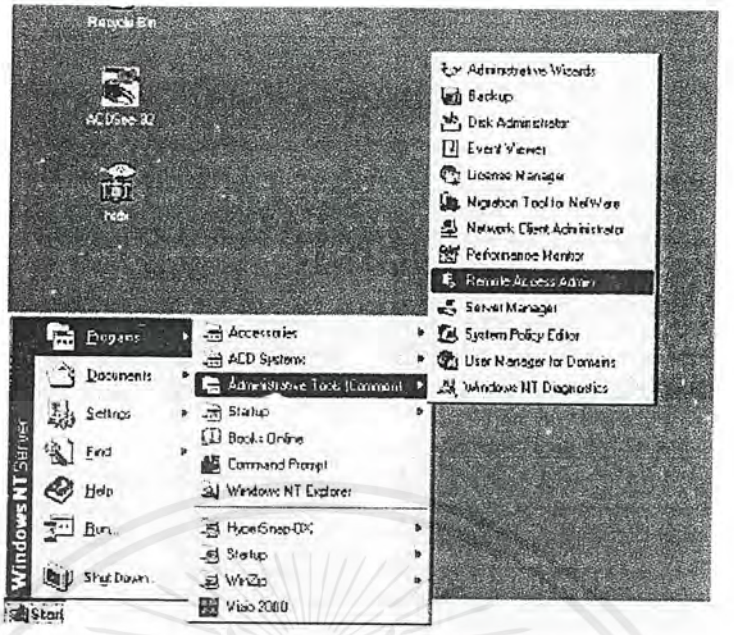
ปรับแต่ง Username ให้สามารถใช้งานได้จากทางไกล

แต่ละ User ของ Windows NT จะไม่มีทางใช้งานจากทางไกลได้ถ้าไม่ได้รับการปรับแต่ง ดังนั้นคุณจะต้องกำหนดให้ Username มีความสามารถใช้งานผ่านทาง Dial- up ได้เสียก่อน ซึ่งถ้าไม่ปรับแต่งตามคุณสมบัตินี้ จะไม่มีสิทธิ (Permission) ในการ Login เข้าใช้เซอร์ฟเวอร์ผ่านทางระบบ RAS ได้

ขั้นตอนต่อไปนี้เป็นการกำหนด Permission เพื่อให้ User สามารถ Login เข้าสู่ Windows –NT ได้ผ่านทาง Dial -Up Network

1. บนเครื่อง Windows NT Server ให้เลือก Start > Program > User Manager for Domains

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



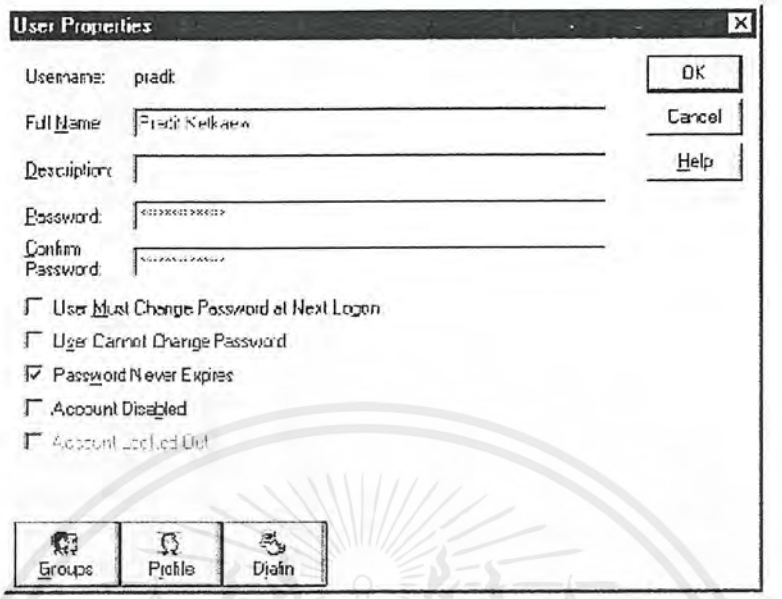
รูปที่ 2.34 แสดงการเข้าสู่ Remote Access Admin

2. ดับเบิ้ลคลิก User ที่ต้องการปรับแต่ง

รูปที่ 2.35 แสดงกรอบ User Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกปุ่ม Dialin



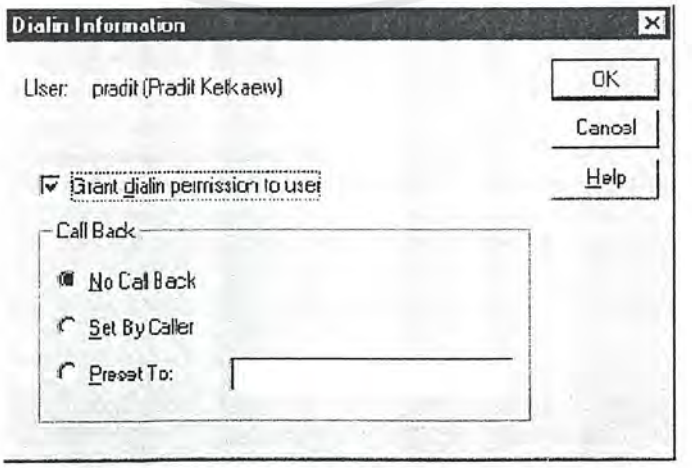
รูปที่ 2.36แสดงกรอบรายละเอียดของ User

4. คลิกใส่เครื่องหมายถูกหน้าข้อความ Grant Dialin Permission to user
5. เลือกตัวเลือกในกรอบ Call Back สำหรับวิธีตอบรับโทรศัพท์ของเซอร์ฟเวอร์

▶ คลิก No Call Back ถ้าไม่ต้องการให้เซอร์ฟเวอร์หมุนโทรศัพท์กลับไปหา Userติดต่อมา

▶ คลิก Set By Caller ถ้าไม่ต้องการให้เซอร์ฟเวอร์โทรศัพท์กลับไปหา User ที่ติดต่อมาตามเบอร์ที่ User ได้กำหนดไว้ใน Dial-up Networking ของเขา วิธีนี้จะช่วยให้ประหยัดค่าโทรศัพท์ของผู้ที่ติดต่อมาได้ เนื่องจากทางด้านเซอร์ฟเวอร์จะเป็นผู้เสียค่าโทรเอง

▶ คลิก Present To แล้วกรอกเบอร์โทรศัพท์ที่ต้องการให้เซอร์ฟเวอร์หมุนกลับในช่องว่างด้านหลัง วิธีนี้จะเป็นการหมุนโทรศัพท์ตายตัว ช่วยเพิ่มความปลอดภัยให้ เซอร์ฟเวอร์ เนื่องจาก User จะติดต่อเฉพาะ User ที่ ต้องการและให้โทรศัพท์ไว้ก่อนแล้วเท่านั้น



รูปที่ 2.37 แสดงกรอบการกำหนด Permission ของ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คลิกปุ่ม OK หลังจากเลือกตัวเลือกในกรอบ Call Back เสร็จเรียบร้อย

7. คลิกปุ่ม OK เพื่อจบการปรับแต่ง

ติดตั้งไคลเอนส์สำหรับ Windows 95

คุณสามารถติดตั้งไคลเอนส์ของ RAS ได้ง่ายๆ ถ้าคุณใช้Windows 95 โดยสร้างไคลเอนส์สำหรับต่อโทรศัพท์อัตโนมัติไว้ใน Dial-up network

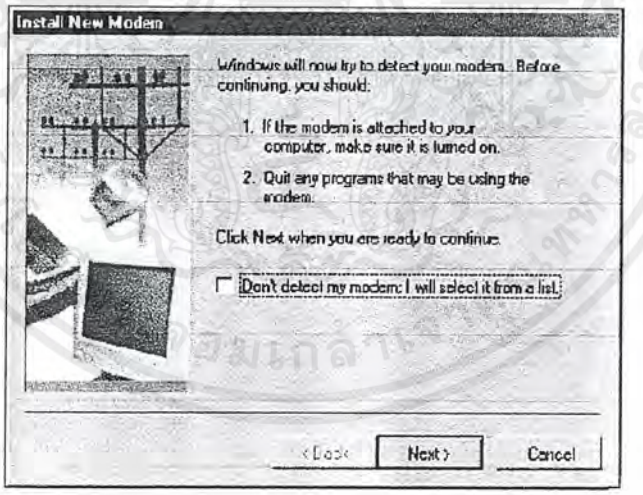
การติดตั้งไคลเอนส์สำหรับ Windows 95 มีขั้นตอนตัวอย่างดังต่อไปนี้

1. บนเครื่อง Windows 95 ที่ต้องการนำมาติดตั้ง ให้ดับเบิ้ลคลิก ไอคอน My computer
2. ดับเบิ้ลคลิก Dial-up Networking
3. ถ้าคุณยังไม่ได้ติดตั้งโมเด็ม คุณต้องติดตั้งโมเด็มเสียก่อนโดยคลิกที่ปุ่ม Next

การติดตั้งโมเด็ม

หากคุณได้ติดตั้งโมเด็มไว้เรียบร้อยแล้ว คุณสามารถข้ามขั้นตอนนี้ได้ทันที แต่ถ้าคุณต้องการศึกษาวิธีการติดตั้งโมเด็ม สามารถทำตามขั้นตอนต่อไปนี้

1. คลิกปุ่ม Next เพื่อให้ Windows 95 ตรวจสอบสอบหาโมเด็มโดยอัตโนมัติ
2. คลิกปุ่ม Finish เมื่อตรวจสอบโมเด็มเสร็จเรียบร้อบแล้ว



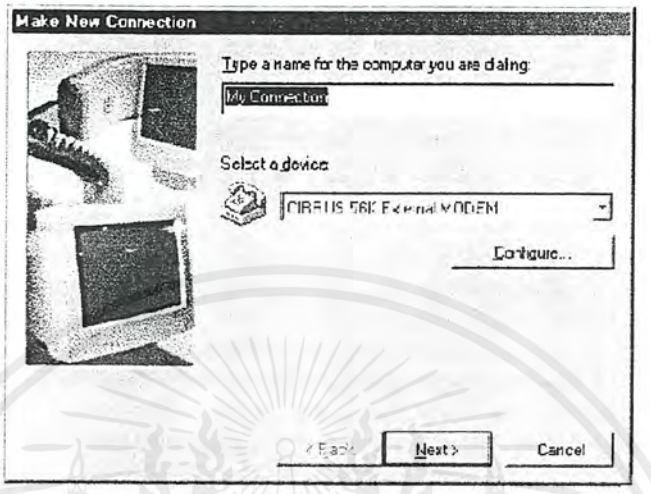
รูปที่ 2.38 แสดงกรอบการเริ่มติดตั้งโมเด็ม

การติดตั้งหมายเลขโทรศัพท์ของเซอร์ฟเวอร์

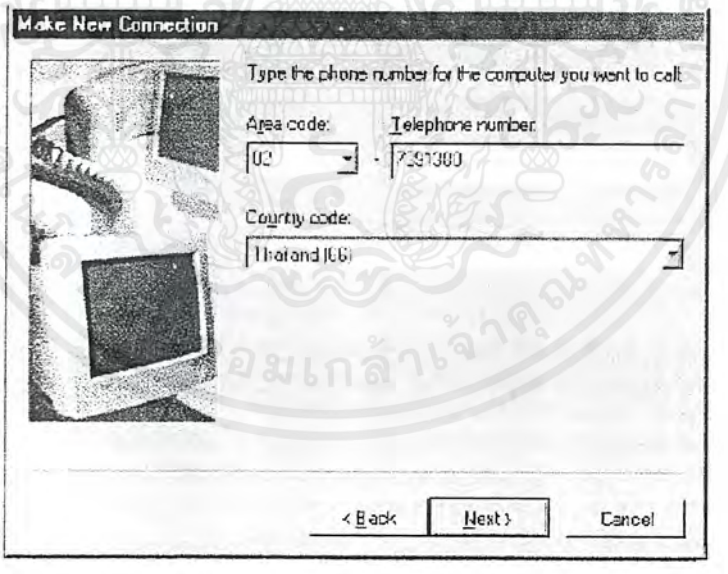
1. ตั้งชื่อ เซอร์ฟเวอร์ที่คุณต้องการติดต่อด้วย ในช่อง Type the name for the computer you are dialing.
2. คลิกปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกประเทศที่ตั้งเซอร์ฟเวอร์ลงในช่องรายการ Country code: (ถ้าไม่ใช้ให้ปล่อยว่างไว้)
4. กรอกหมายเลขโทรศัพท์ของเซอร์ฟเวอร์ลงในช่อง Telephone Number:
5. คลิกปุ่ม Finish



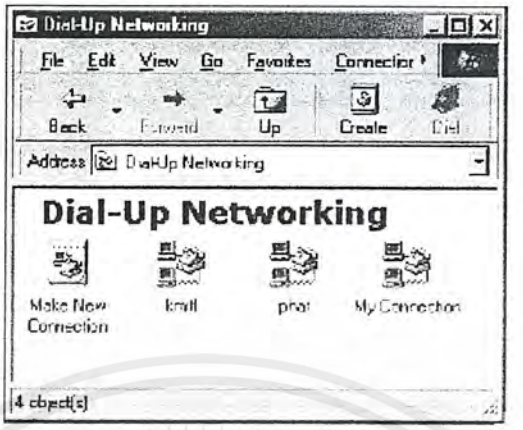
รูปที่ 2.39 แสดงกรอบการสร้าง New Connection



รูปที่ 2.40 แสดงกรอบการกำหนดหมายเลขโทรศัพท์ให้ New Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คุณจะพบกับ ไอคอนสำหรับต่อโทรศัพท์อัตโนมัติ ปรากฏขึ้นในหน้าต่าง Dial-up Networking

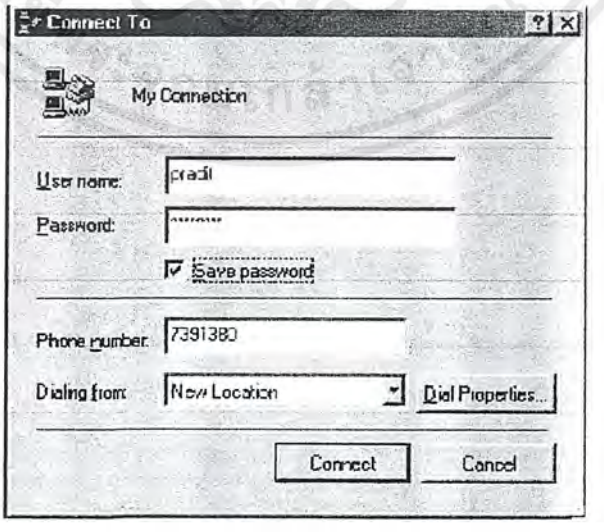


รูปที่ 2.41 แสดงกรอบแสดงไอคอนใน Dialing-Up Networking

ทดสอบการใช้งาน

เมื่อคุณสร้างไอคอนสำหรับต่อโทรศัพท์ไปยังเซอร์ฟเวอร์เสร็จเรียบร้อยแล้ว คุณสามารถทดลองต่อโทรศัพท์ เพื่อ Login ไปยังเซอร์ฟเวอร์ได้ทันที

1. ดับเบิ้ลคลิก ไอคอน Dial-up Networking
2. ดับเบิ้ลคลิก ไอคอนที่ ใช้สำหรับต่อโทรศัพท์ไปยังเซอร์ฟเวอร์
3. กรอก ชื่อสำหรับ Login ลงในช่อง User name:
4. กรอก พาสเวิร์ดสำหรับ Login ของคุณ ลงในช่อง Password:
5. ถ้าคุณต้องการปรับแต่งหมายเลขโทรศัพท์หรือต้องการให้ ตัด 9 ก่อนโทรออก ให้คลิกปุ่ม Dial Properties



รูปที่ 2.42 แสดงกรอบที่ใช้ติดต่อยังเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กรอกเลข 9 ถ้าโทรศัพท์ที่คุณใช้ผ่าน PABX ที่ต้องกดเลข 9 ก่อนการโทรออก
7. คลิกปุ่ม OK เมื่อปรับแต่งเรียบร้อยแล้ว
8. คลิกปุ่ม Connect
9. โมเด็มจะทำการเชื่อมต่อ และ Login เข้าสู่เซอร์ฟเวอร์โดยอัตโนมัติ
10. คุณจะเข้าสู่เซอร์ฟเวอร์ได้ราวกับคุณเป็นผู้ใช้ที่ต่ออยู่ในเครือข่ายปกติทุกประการ

รูปที่ 2.43 แสดงกรอบขณะจะติดต่อกับเครื่อง Server

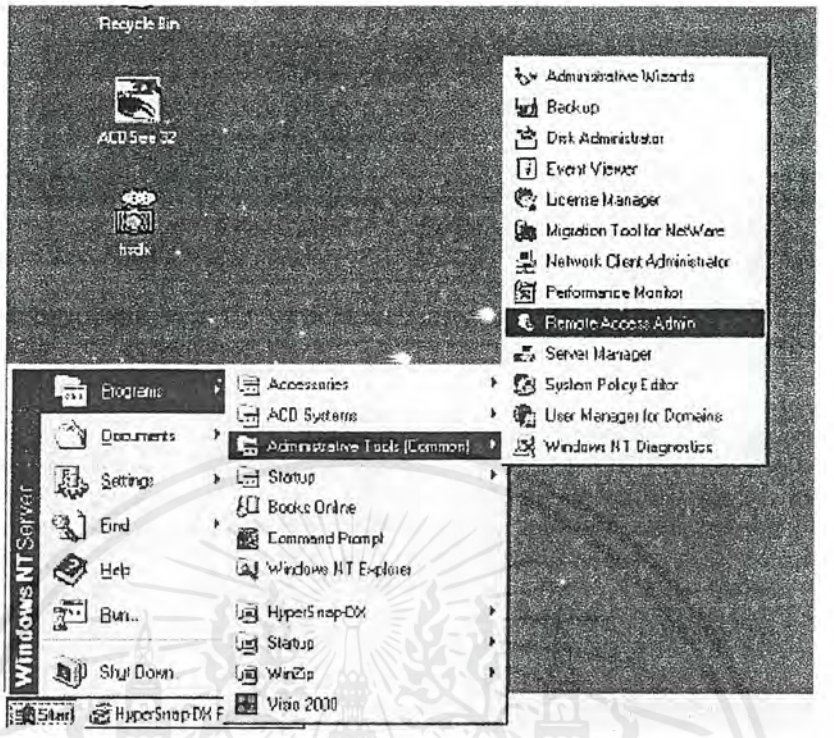
2.5.3 การบริหารและควบคุมระบบ RAS

ควบคุมระบบRAS

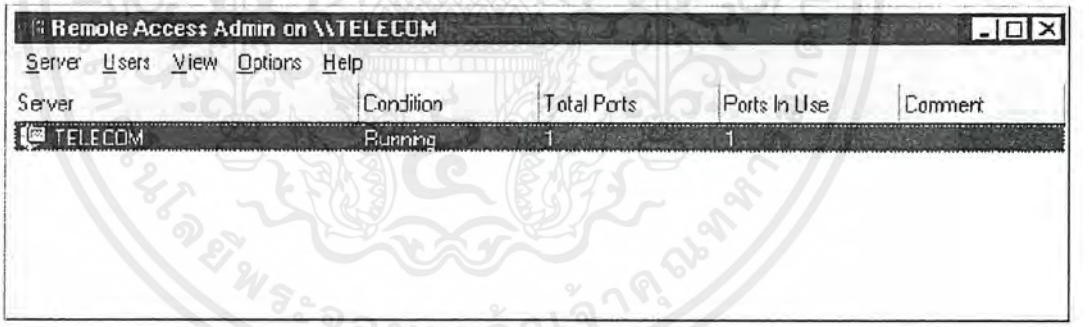
จอรายงานผลของ RAS (Remote Access Admin) เป็นโปรแกรมที่ใช้ แสดงภาวะการเชื่อมต่อของผู้ใช้ผ่านทางระบบ RAS คุณสามารถควบคุมการเชื่อมต่อของอุปกรณ์การสื่อสารต่างๆ ได้ง่ายๆผ่านโปรแกรมนี้

1. เลือก Start > Administrative Tool > Remote Access Admin
2. คุณจะพบกับโปรแกรม Remote Access Admin ซึ่งแสดงการเชื่อมต่อขณะนั้น
3. คับเบิ้ลคลิก ที่รายการแสดงการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.44 แสดงหน้าต่างเพื่อเข้าสู่ Remote Access Admin



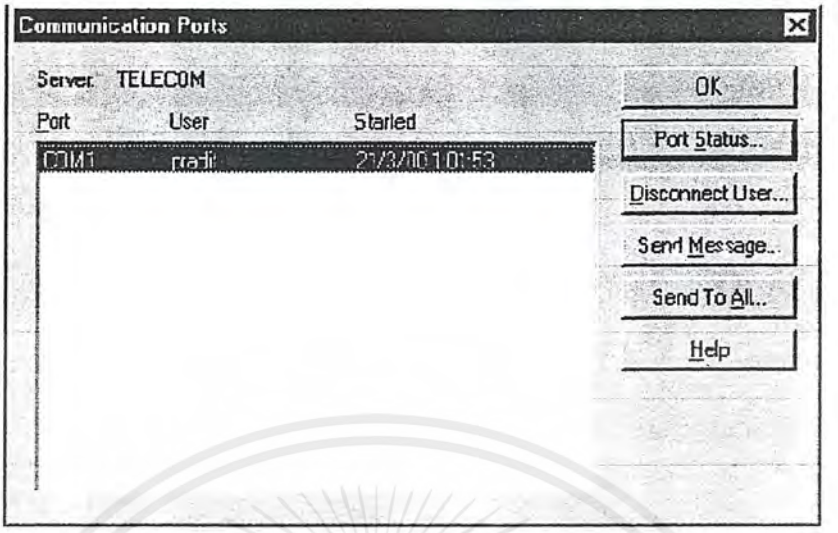
รูปที่ 2.45 แสดงกรอบสภาวะของ RAS

4. คุณจะพบกับรายละเอียดของ Communication Ports ซึ่งคุณสามารถควบคุมการเชื่อมต่อ นี้ โดยคลิกที่ปุ่มต่อไปนี้

Port Status	ใช้สถิติของการส่งข้อมูลระหว่าง user กับเซอร์ฟเวอร์
Disconnect User	ใช้ตัดการติดต่อระหว่าง User กับ เซอร์ฟเวอร์ทันที
Send Message	ส่งข้อมูลให้ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sent to All ส่งข้อมูลให้ user ที่เชื่อมต่ออยู่ทั้งหมดพร้อมกัน

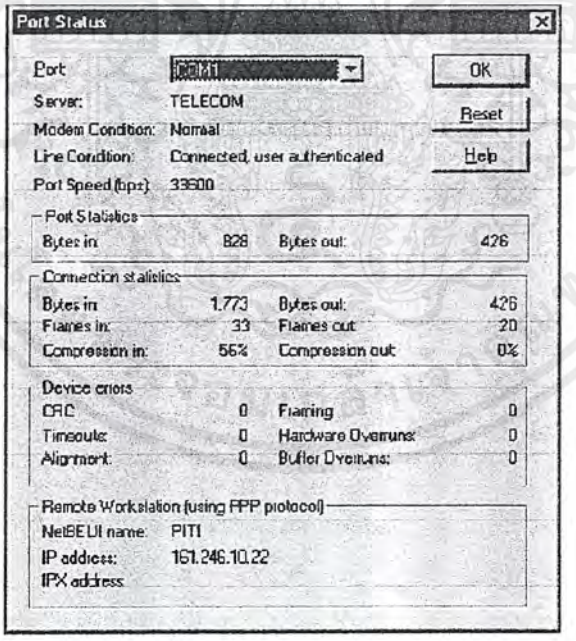


รูปที่ 2.46 แสดงกรอบสภาวะของ Port ที่ใช้ในการสื่อสาร

ให้คุณคลิกปุ่ม Port Status เพื่อศึกษาคุณสมบัติของ Port Status ในหัวข้อต่อไปนี้

การตรวจสอบข้อมูลกับการเชื่อมต่อ

เมื่อคุณคลิกปุ่ม Port Status คุณจะพบกับหน้าต่างแสดงข้อมูลของการเชื่อมต่อ



รูปที่ 2.47 แสดงกรอบรายละเอียดการส่งผ่านข้อมูลของ Port

Port

แสดงว่า User ได้เข้ามาเชื่อมต่อผ่านพอร์ตใดของเซอร์ฟเวอร์

Server

แสดงชื่อของเซอร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรอบ Port Statistics

แสดงข้อมูลที่เข้าสู่ เซอร์ฟเวอร์ (Bytes In) และข้อมูลที่ส่งออกจากเซอร์ฟเวอร์(Bytes Out) ผ่านทางพอร์ตนี้ โดยข้อมูลที่แสดงนี้จะเป็นการส่งข้อมูลออกจริงๆที่ได้รับการบีบอัดแล้ว

กรอบ Connection Statics

แสดงข้อมูลที่เข้าสู่ เซอร์ฟเวอร์ (Bytes In)และข้อมูลที่ออกจากเซอร์ฟเวอร์(Byte Out) ผ่านทางพอร์ตนี้โดยข้อมูลที่แสดงนี้เป็นขนาดของข้อมูลที่ยังไม่ได้รับการบีบอัด

กรอบ Device Errors

แสดงการเสียหายของอุปกรณ์ต่างๆ หากมีสิ่งผิดพลาด และคุณสามารถตรวจสอบได้ทันทีจากกรอบนี้

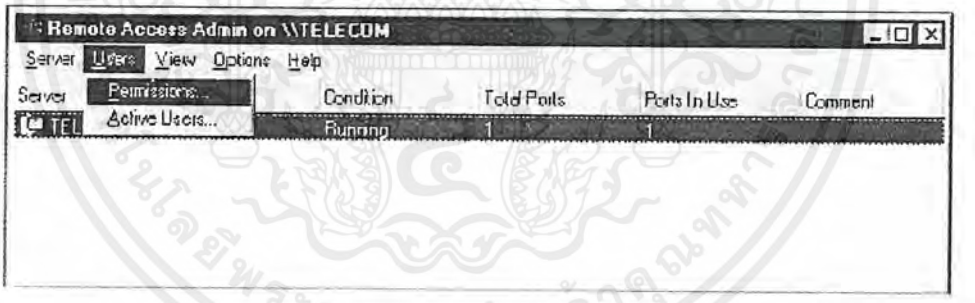
กรอบ Remote Workstation

ใช้แสดงชนิดของโปรโตคอล และเน็ตเวอร์คแอสเดรสของเครื่องที่ต่อเข้ามา

การกำหนดสิทธิของUser

นอกจากกำหนดสิทธิของ User ด้วยวิธีที่แสดงข้างต้นแล้ว คุณยังสามารถกำหนดสิทธิ ต่างๆ ของผู้ใช้แต่ละคนได้โดยใช้ Remote Admin อีกด้วย

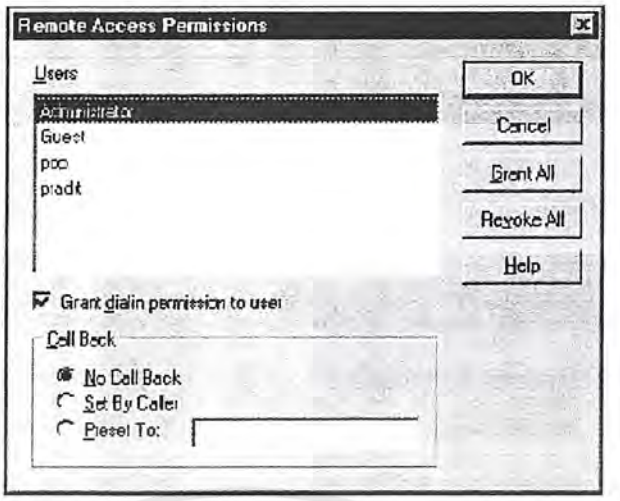
1. คลิกเลือก Connection ที่ต้องการ
2. เลือกเมนู User > Permission



รูป 2.48 แสดงการเข้าสู่การกำหนด Permission ให้ User

3. คลิกเลือก User ที่ต้องการปรับแต่ง
4. คลิก Grant dialin permission to user เพื่อให้ผู้ใช้สามารถใช้บริการ RAS ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.49 แสดงกรอบ Permission ในการใช้งาน RAS ของ User

5. ในกรอบ Call Back คุณสามารถเลือกได้ดังนี้

No Call back

เซอร์ฟเวอร์ไม่หมุนโทรศัพท์กลับ

Set by Caller

ขึ้นอยู่กับทางผู้ใช้ ที่หมุนเข้ามา วิธีนี้จะช่วยให้ผู้ที่หมุนโทรศัพท์เข้ามาสามารถกำหนดเบอร์โทรศัพท์ได้ด้วยตัวเอง

Present To:

เจาะจงให้หมุนโทรศัพท์ที่กำหนดเท่านั้น วิธีนี้จะช่วยให้คุณสามารถกำหนดเบอร์โทรกลับได้อย่างเจาะจง ทำให้เซอร์ฟเวอร์หมุนโทรศัพท์ไปยังผู้รับที่ถูกกำหนดไว้เท่านั้น สามารถป้องกันผู้ที่ไม่ประสงค์ดีได้

2.6 การเขียนโปรแกรมแบบวิชวล

วิชวล (Visual) หมายถึง ภาพหรือสิ่งที่เรามองเห็น ดังนั้น วิชวลโปรแกรมมิง (ViSual Programming – การเขียนโปรแกรมแบบวิชวล) จึงหมายถึง การเขียนโปรแกรมด้วยภาพ หรือการเขียนโปรแกรมด้วยสิ่งที่เรามองเห็น

ภาพประกอบเบื้องต้นสำหรับใช้เขียนโปรแกรมแบบวิชวลมี 4 รายการดังต่อไปนี้

1.ฟอร์ม (Form) ฟอร์มเป็นวินโดวส์ว่างใช้สำหรับออกแบบโปรแกรม การทำงานต่างๆ ของโปรแกรมจะปรากฏอยู่บนฟอร์ม

2.คอมโพเนนต์ (Component) คอมโพเนนต์เป็นส่วนประกอบต่างๆ ที่จะต้องใช้ในโปรแกรม เช่น เมนู (Menu) ปุ่มหรือบัตทอน (Button – ปุ่มสำหรับกดให้โปรแกรมทำงานชนิดหนึ่ง) เมสเสจบ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Message Box – กรอบแสดงข้อความไดอะล็อกบ็อกซ์) ไดอะล็อกบล็อกซ์ (Dialog Box – กรอบสำหรับโต้ตอบกับโปรแกรม)

3.คำสั่งสำหรับจัดลักษณะและการทำงานของคอมโพเนนต์

4.คำสั่งสำหรับควบคุมการทำงานของโปรแกรม

วิชวลโปรแกรมมิงช่วยให้เราสามารถเขียนโปรแกรมได้ง่าย สะดวก รวดเร็ว และถูกต้อง

ปาสคาล ออบเจ็กต์ปาสคาล และเดลไฟ

ปาสคาล (Pascal) เป็นภาษาคอมพิวเตอร์ที่สร้างขึ้นโดยมีจุดมุ่งหมายเพื่อให้ผู้เขียนโปรแกรมได้เรียนรู้เกี่ยวกับวิธีการเขียนโปรแกรมที่ดี ซึ่งจะเป็นพื้นฐานในการเขียนโปรแกรมขนาดใหญ่ที่มีความซับซ้อนได้โดยสะดวก ปาสคาลที่นิยมใช้กันมากในไมโครคอมพิวเตอร์ได้แก่ เทอร์โบปาสคาล (Terbo Pascal) ซึ่งเป็นผลงานของบริษัท บอร์แลนด์อินเตอร์เนชันแนล (Borland International, Inc.) ประเทศสหรัฐอเมริกา เทอร์โบปาสคาลเวอร์ชัน 1.0 วางจำหน่ายในปี 1984 ตั้งแต่ เวอร์เวอร์ชัน 5.5 ซึ่งวางจำหน่ายในปี 1999 เป็นต้นมา เทอร์โบปาสคาลได้เพิ่มวิธีการเขียนโปรแกรมแบบออบเจ็กต์ ซึ่งเรียกว่า การเขียนโปรแกรมแบบ OOP (Object – Oriented Programming) และเรียกชื่อปาสคาลชนิดนี้ว่า ออบเจ็กต์ปาสคาล (Object Pascal) ออบเจ็กต์ปาสคาลมีทั้งชนิดที่รันในดอสและรันในวินโดวส์

ออบเจ็กต์ปาสคาลและเดลไฟ (Delphi) ต่างก็เป็นภาษาคอมพิวเตอร์ซึ่งใช้สำหรับเขียนโปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำงานต่างๆ ตามที่เราต้องการ ออบเจ็กต์ปาสคาลและเดลไฟมีคำและกฎเกณฑ์ในการเขียนโปรแกรมเหมือนกันเกือบทุกประการ แต่มีวิธีเขียนโปรแกรมต่างกัน กล่าวคือ ออบเจ็กต์ปาสคาลใช้วิธีเขียนโปรแกรมด้วยอักษรและเครื่องหมาย แต่เดลไฟใช้วิธีเขียนโปรแกรมแบบวิชวล ดังนั้นผู้ที่เขียนโปรแกรมด้วยออบเจ็กต์ปาสคาลจึงสามารถเปลี่ยนมาเขียนโปรแกรมด้วยเดลไฟได้อย่างรวดเร็ว เพราะเปลี่ยนแปลงเฉพาะวิธีการเท่านั้น ซึ่งวิธีการของเดลไฟจะทำให้เราสามารถเขียนโปรแกรมได้ง่ายและเร็วกว่าวิธีการของออบเจ็กต์ปาสคาล

เดลไฟผลิตโดยบริษัท บอร์แลนด์อินเตอร์เนชันแนล เดลไฟเวอร์ชันแรกนั้นวางจำหน่ายในปี 1995 เดลไฟเวอร์ชัน 4 ซึ่งมีชื่อว่า Delphi 4 วางจำหน่ายประมาณกลางปี 1998 โดยแบ่งการจำหน่ายเป็น 4 ชุด คือ

1. Delphi Standard (ชุดมาตรฐาน)
2. Delphi Professional (ชุดระดับมืออาชีพ)
3. Delphi Client/Server Suite (ชุดสำหรับเครือข่ายคอมพิวเตอร์)
4. Delphi Enterprise (ชุดสำหรับองค์การ)

เดลไฟเป็นวิชวลโปรแกรมมิง ซึ่งมีวิธีการเขียนโปรแกรมโดยการนำคอมโพเนนต์ที่เรามองเห็นมาวางในฟอร์ม แล้วกำหนดลักษณะและการทำงานให้กับคอมโพเนนต์นั้น ในบางโปรแกรมอาจจะต้องเขียนคำสั่งควบคุมการทำงานของโปรแกรมเพิ่มเติม ต่อจากนั้นจึง คอมไพล์ (Compile) และ รัน (run) โปรแกรมนั้นได้ ผลจากการคอมไพล์โปรแกรมจะได้ไฟล์ชนิด .EXE ซึ่งสามารถนำไปรันในคอมพิวเตอร์เครื่องอื่นได้ อย่างอิสระ โดยไม่ต้องเกี่ยวข้องกับเดลไฟอีกเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดลไฟเป็นออบเจ็กต์ปาสคาลประเภทที่สามารถสร้างโปรแกรมใช้งานได้อย่างรวดเร็ว ภาษาคอมพิวเตอร์ประเภทนี้มีชื่อย่อว่า RAD (Rapid Application Development) ซึ่งหมายความว่า ด้วยเดลไฟเราจะสามารถเขียนโปรแกรมเพื่อรันในวินโดวส์ได้เร็วกว่าและง่ายกว่าการเขียนโปรแกรมออบเจ็กต์ปาสคาลแบบธรรมดา

อีเวนต์

อีเวนต์ หมายถึง ทุกสิ่งทุกอย่างที่เกิดขึ้นในระบบคอมพิวเตอร์ เช่น การเลื่อนเมาส์ การคลิกเมาส์ การกดคีร์ การเลื่อนวินโดวส์ การเปลี่ยนขนาดของวินโดวส์ วินโดวส์จะเก็บอีเวนต์ทั้งหมดไว้ในคิว (Queue) ที่มีชื่อว่า โกลบอลอีเวนต์คิว (Global Even Queue – คิวของอีเวนต์ทั้งหมด ซึ่งบางทีเรียกว่า System – Wide Queue หรือ Hardware Even Queue) โกลบอลอีเวนต์คิวใช้สำหรับเก็บอีเวนต์ของทุกแอปพลิเคชันที่รันในวินโดวส์

เมสเสจ

เมสเสจ หมายถึง ชุดของข่าวสารที่เป็นรายละเอียดของแต่ละอีเวนต์ วินโดวส์จะสร้างเมสเสจของอีเวนต์ โดยการส่งอีเวนต์ที่เกิดขึ้นให้แก่ ไดรเวอร์ (Driver – โปรแกรมที่คอมพิวเตอร์ใช้ติดต่อกับอุปกรณ์ชนิดนั้น) ที่สอดคล้องกับอีเวนต์นั้น เช่น เมื่อเราคลิกเมาส์จะเกิดอีเวนต์คลิกเมาส์ วินโดวส์จะส่งอีเวนต์นี้ให้แก่เมาส์ไดรเวอร์ และเมาส์ไดรเวอร์จะสร้างเมสเสจคลิกเมาส์ วินโดวส์จะส่งอีเวนต์นี้ให้แก่เมาส์ไดรเวอร์ และเมาส์ไดรเวอร์จะสร้างเมสเสจคลิกเมาส์ให้แก่วินโดวส์ ไดรเวอร์จะเก็บเมสเสจจากนี้ไว้ในซิสเต็มคิว (System Queue – คิวของระบบ) ต่อจากนั้นวินโดวส์จึงส่งเมสเสจจากซิสเต็มคิวไปที่ แอปพลิเคชันคิว (Application Queue – คิวเก็บเมสเสจของแอปพลิเคชัน) ของแอปพลิเคชันต่างๆ ที่กำลังทำงานอยู่ในขณะนั้น ซึ่งเป็นผลให้ แอปพลิเคชันได้รับเมสเสจจากวินโดวส์ เนื่องจากวินโดวส์สามารถรันแอปพลิเคชันคิวสำหรับแอปพลิเคชันนั้นโดยเฉพาะความสัมพันธ์ระหว่างอีเวนต์กับเมสเสจ

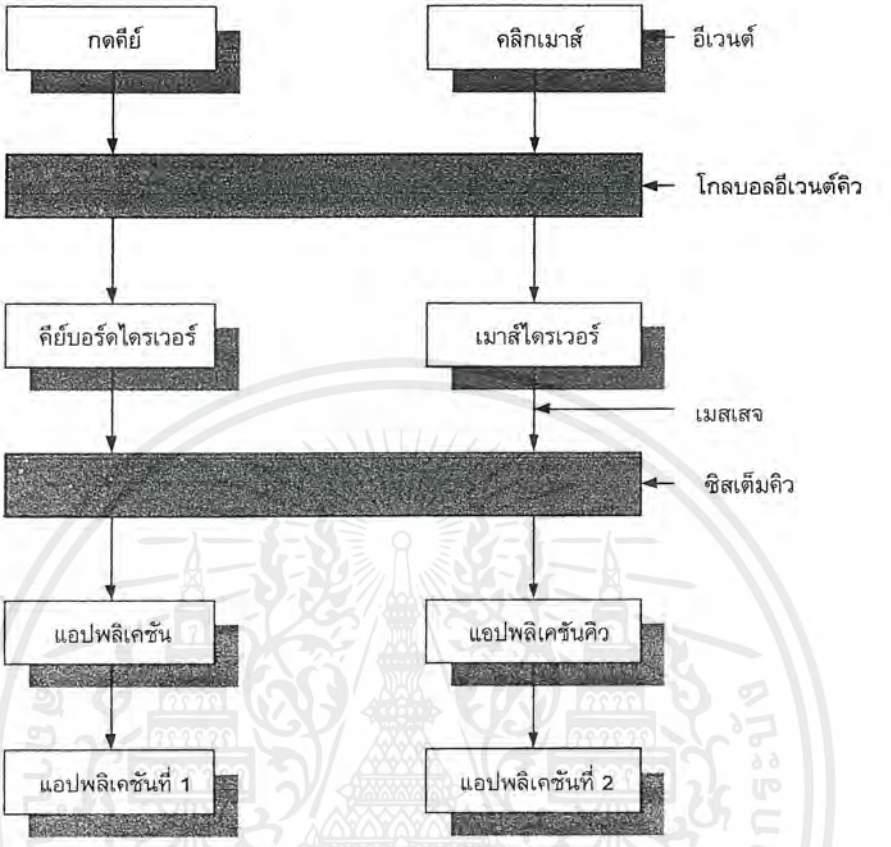
หลักการทำงานของแอปพลิเคชันที่รันในวินโดวส์ก็คือ การดำเนินการกับเมสเสจ การรับส่งเมสเสจกับวินโดวส์หรือกับแอปพลิเคชันอื่น เมสเสจทั้งหมดของวินโดวส์กำหนดไว้เป็นคอนสแตนต์ (Constant – ค่าที่ไม่เปลี่ยนแปลง) เช่น เมื่อเราเลือกรายการหนึ่งจากเมนูที่ปรากฏบนจอภาพจะมี เมสเสจ WM_COMMAND เกิดขึ้น ในบรรดาเมสเสจเหล่านี้มีหลายเมสเสจที่เกิดขึ้นพร้อมกับแวริเอเบิล (Variable – ค่าที่เปลี่ยนแปลงได้) จำนวนหนึ่งซึ่งนำมาใช้ในแอปพลิเคชันได้ เช่น เมื่อเรากดปุ่ม 3 วินโดวส์จะสร้างเมสเสจ WM_LBUTTONDOWN ขึ้นมา เมสเสจนี้ประกอบด้วยแวริเอเบิล ซึ่งเป็นโคออร์ดิเนต (Coordinate – ตำแหน่ง) ของเมาส์พอยน์เตอร์ เราสามารถนำค่าดังกล่าวนี้มาใช้ได้ เช่น ใช้สำหรับแสดงผลบนจอภาพ ใช้ในการตรวจสอบตำแหน่งของเมาส์พอยน์เตอร์

โปรแกรมที่ทำงานด้วยอีเวนต์

แอปพลิเคชันที่รันในวินโดวส์ จะใช้เวลาส่วนใหญ่เพื่อการรับและส่งเมสเสจชนิดต่าง ๆ จึงนิยมเรียกแอปพลิเคชัน (หรือโปรแกรม) ที่ทำงานในลักษณะนี้ว่าโปรแกรมที่ทำงานด้วยอีเวนต์ (Even – Drive Program) ซึ่งเป็นโปรแกรมที่มีหลักการทำงานที่มีหลักการทำงานดังต่อไปนี้คือ ในโปรแกรมที่ทำงานด้วยอีเวนต์จะประกอบด้วย เมธอด (Method – สเตตเมนต์ชุดหนึ่งซึ่งทำงานอย่างหนึ่ง และมีชื่อเฉพาะสำหรับให้ส่วนอื่นของโปรแกรมเรียกใช้ได้)จำนวนหนึ่งซึ่งเรียกว่าเมธอดตอบสนองอีเวนต์ ซึ่งแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดจะมีหน้าที่ตอบสนองกับแต่ละอีเวนต์โดยเฉพาะและเป็นอิสระต่อกัน ทั้งนี้เนื่องจากในวินโดวส์จะมีอีเวนต์เกิดขึ้นเวลาใดก็ได้ เราจึงไม่ทราบล่วงหน้าว่า จะเรียกใช้เมธอดอะไร เมื่อใด



รูปที่ 2.50 ลักษณะของโปรแกรมที่ทำงานด้วยอีเวนต์

โครงสร้างโปรแกรมของเดลไฟ (The Structure of a Delphi Program)

การเขียนโปรแกรมเดลไฟให้ได้ประโยชน์มากขึ้น ถ้านักพัฒนาโปรแกรมเข้าใจโครงสร้างของ Object Pascal ซึ่งเป็นรากฐานของเดลไฟ

IDE เป็นเครื่องมือที่ช่วยให้สามารถเขียนโปรแกรมได้เร็วขึ้น โดยที่ IDE ช่วยเขียนคำสั่งที่จำเป็นต้องใช้ และช่วยให้นักพัฒนาสามารถออกแบบจอภาพและเขียนคำสั่งที่ต้องการให้ทำงานเมื่อมีเหตุการณ์ (even) เกิดขึ้น

IDE จะช่วยเขียนโค้ดคำสั่งงานซึ่งจะอยู่ภายใต้กฎเกณฑ์ของ Object Pascal และช่วยสร้าง resource ที่ต้องใช้ตามกฎเกณฑ์การทำงานของระบบปฏิบัติการแบบ Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหัวข้อนี้ จะได้ศึกษาถึงโครงสร้างของโปรแกรมและส่วนของคำสั่งงานในโปรแกรมที่จำเป็นต้องใช้

โพรซีเดอร์ (Procedure)

โพรซีเดอร์ เป็นกลุ่มคำสั่งงานที่ใช้ในการทำงานอย่างใดอย่างหนึ่ง อาจจะเรียกอีกอย่างว่า "โปรแกรมย่อย" เช่น การคำนวณภาษี การแสดงจอภาพเพื่อรับค่าจากผู้ใช้งาน เป็นต้น

ตัวอย่างของโพรซีเดอร์

```
1.procedure Button1Click;
2.begin
3.Edit1.Text:='Hello World';
4.End;
```

ในโพรซีเดอร์ตัวอย่างนี้จะแสดงข้อความ 'Hello World' ในคอนโทรล Edit1 เมื่อผู้ใช้งานกดปุ่ม Button1 โดยที่ชื่อโพรซีเดอร์จะอยู่บรรทัดที่ 1 และมีคำสั่ง begin และ end กลุ่มคำสั่งที่ใช้ในการทำงานของโพรซีเดอร์นี้

- คำสั่งอยู่ในโพรซีเดอร์อาจมีได้ตั้งแต่คำสั่งเดียวจนถึงหลาย ๆ หน้ากระดาษ
- เนื่องจากภาษา Object Pascal เป็นภาษาที่ Case-Insensitive ดังนั้น การตั้งชื่อต่าง ๆ ที่ใช้อักขระตัวใหญ่และอักขระตัวเล็กจะไม่แตกต่างกัน เช่น Edit1 หรือ edit1 จะเป็นการอ้างถึงชื่อเดียวกัน
- เพื่อความสะดวกในการอ่านโปรแกรมและความง่ายในการแก้ไขโปรแกรม ควรจะใช้อักขระตัวใหญ่ผสมกับตัวเล็ก

การสร้างโพรซีเดอร์

ปกติจะสร้างโดยใช้ IDE อย่างไรก็ตามเราสามารถสร้างขึ้นเองได้ถ้ามีความเข้าใจโครงสร้างของ Object Pascal อย่างดี

ตัวอย่างการสร้างโพรซีเดอร์โดยใช้ IDE ให้ทำดังนี้

1.ใช้เมนู File/New Application เพื่อสร้างโปรเจ็คใหม่

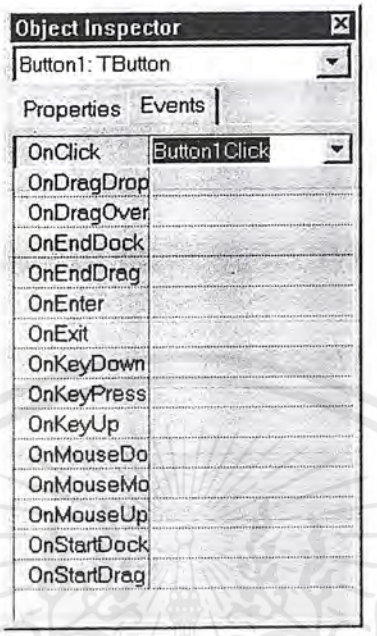
2.เพิ่มคอนโทรล Edit และ Button1

3.สร้างโพรซีเดอร์เพื่อทำงานตามเหตุการณ์ ซึ่งมี 2 วิธีดังนี้

- ดับเบิลคลิกที่คอนโทรลใดบนฟอร์ม IDE จะสร้างโพรซีเดอร์เพื่อทำงานตาม Event ที่ต้องการ วิธีนี้จะได้ default event สำหรับคอนโทรลนั้นหรือ
- คลิกที่คอนโทรลเพื่อให้คอนโทรลนั้นได้โฟกัส แล้วดูที่ Object Inspector ที่ Event Page จากนั้นดับเบิลคลิกที่ Event ที่ต้องการ

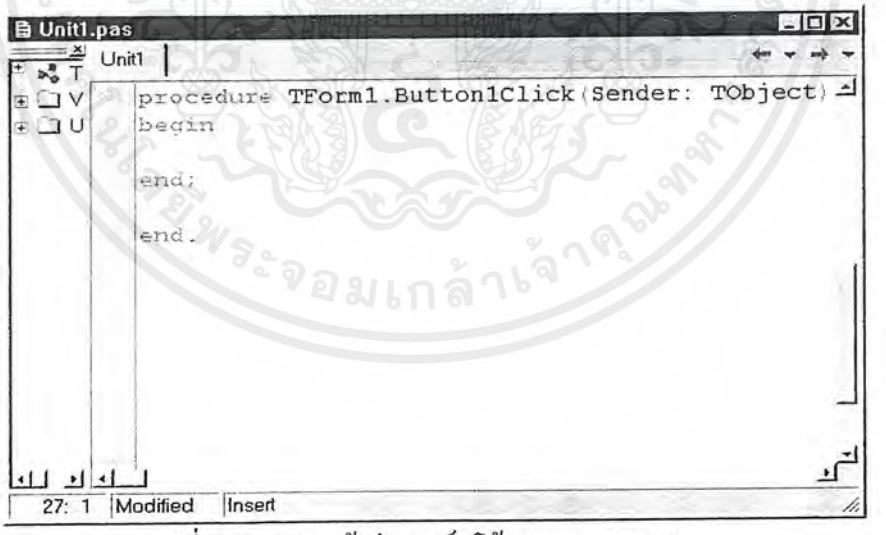
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างเราต้องการสร้างโพรซีเดอร์สำหรับ Button1 เมื่อผู้ใช้คลิกที่ปุ่ม ดังนั้น ให้คลิกที่ Button1 ให้ได้โฟกัสก่อนแล้วเลื่อนเมาส์ไปที่ Object Inspector คลิกที่ Event Page ซึ่งจะปรากฏจอภาพดังรูป



รูปที่ 2.51 หน้าต่างออปเจ็กอินสเปกเตอร์ (Object Inspector)

4.ดับเบิลคลิกบรรทัด OnClick จะทำให้ IDE การสร้างโค้ดคำสั่งพร้อมทั้งเปิด Editor Window ขึ้นมาดังรูป 2.52



รูปที่ 2.52 แสดงหน้าต่างซอร์สโค้ด (Source Code)

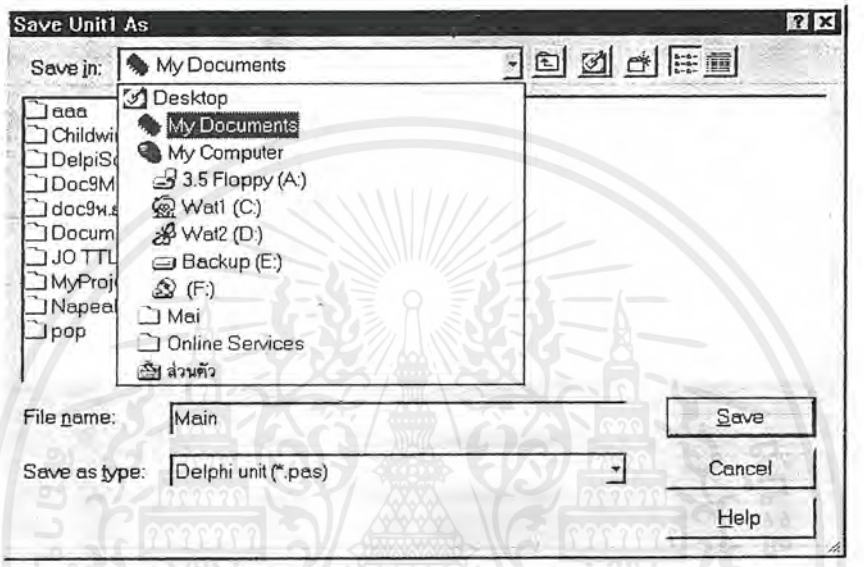
ในกรณีที่ดับเบิ้ลคลิกในจุดที่ไม่ถูกต้องแล้วทำให้เดลไฟสร้างโค้ดคำสั่งที่เราไม่ต้องการแล้ว เราไม่จำเป็นต้องลบโพรซีเดอร์ออก เนื่องจากเมื่อเดลไฟเริ่มแปลโปรแกรมมันจะตรวจสอบโพรซีเดอร์ที่ไม่มีการเขียนคำสั่ง หากพบโพรซีเดอร์ที่มีลักษณะนี้มันจะทำการลบออกให้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลบโพรซีเดอร์โดยผู้เขียนโปรแกรมเอง ถ้าลบไม่ถูกต้องอาจทำให้เกิดข้อผิดพลาดได้ เนื่องจากเคลไฟสร้างโค้ดคำสั่งขึ้นมาสองแห่ง ถ้าต้องการลบเองต้องลบตัวโพรซีเดอร์ และการประกาศโพรซีเดอร์ในส่วนหัวของโปรแกรมด้วย

5.เขียนคำสั่งต่อไปนี้ระหว่างบรรทัดของ begin และ end Edit1.Text:'Hello World': การสลับหน้าต่างของเอดิเตอร์กับฟอร์มให้ใช้ปุ่ม

6.ใช้เมนู File/Save Project แล้วสร้างไดเร็คทอรี่ย่อยชื่อ Control 1 ดังรูปที่2.53



รูปที่ 2.53 หน้าต่างการบันทึกโครงงาน (Save Project)

ทำการตั้งชื่อยูนิต "Main.Pas" และชื่อโปรเจ็ค "Control1.DPR" เพื่อเก็บฟอร์มและโปรเจ็ค ไว้ในไดเร็คทอรีที่สร้างขึ้นใหม่

การเก็บโปรแกรมที่เขียนโดยใช้เคลไฟนั้นควรจะเก็บแยกแต่ละโปรเจ็คไว้ที่คนละไดเร็คทอรี่กัน และฟอร์มที่ถูกกำหนดให้เปิดขึ้นเมื่อเริ่มรันโปรแกรมควรจะตั้งชื่อว่า Main
จากซอร์สโค้ดของยูนิต/ฟอร์ม มีส่วนประกอบสำคัญ 3 ส่วนคือ

Title	ส่วนที่เป็นชื่อยูนิต
Interface	ส่วนที่เป็นการติดต่อกับยูนิต/ฟอร์มอื่น เพื่อกำหนดขอบเขตของข้อมูลและฟังก์ชั่นที่ยอมให้ใช้ได้ โดยใช้หลักการแบบ Object Oriented Programming
Inplementation	เป็นส่วนที่ทำงานตามฟังก์ชั่นที่ได้กำหนดไว้ในส่วน Interface

นอกจากนี้เราอาจเพิ่มส่วนของ initialization เพื่อกำหนดค่าเริ่มต้นครั้งแรกให้กับฟอร์ม/ยูนิต โดยคำสั่งภายใน initialization จะถูกเรียกทำงานครั้งเดียวเพื่อเริ่มทำงานเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของยูนิต

รายละเอียดของส่วน Interface มีดังนี้

คำสั่ง uses...	ใช้สำหรับอ้างอิงถึงยูนิตที่ต้องการนำไปใช้ในขณะที่ทำการคอมไพล์และลิงค์โปรแกรม
คำสั่ง type...	ใช้สำหรับการประกาศรูปแบบของ Object ชื่อ Tform1 ซึ่งได้นำต้นแบบมาจาก Object ของ TFrom โดยที่ TForm เป็น Object ของฟอร์มเปล่า - สำหรับคำสั่ง Edit1 : TEdit; เป็นการกำหนดว่าบนฟอร์มนี้มีคอนโทรลชื่อ Edit1 ซึ่งได้ต้นแบบมาจาก Object Tedit - คำสั่ง Button1:Tbutton; เป็นการกำหนดคอนโทรล Button1 ซึ่งได้ต้นแบบมาจาก Tbutton - คำสั่ง procedure Button1Click... เป็นการกำหนดเมธอด (method) หรือเป็นโพรซีเดอร์ที่ต้องการสั่งให้ทำงาน เมื่อมีการคลิกที่ Button1 โดยที่ IDE จะทำการเชื่อมส่วนนี้ให้ โดยผ่านทางฟอร์ม (Main.DEF) ซึ่งได้กำหนดไว้โดยใช้ Object Inspector โดยทั่วไปชื่อเมธอดจะถูกตั้งโดยใช้ชื่อคอนโทรลและตามด้วยเหตุการณ์ (event) เช่น Button1Click มาจาก Button1 ซึ่งเป็นคอนโทรล กับ Click ซึ่งเป็นเหตุการณ์ที่เกิดกับ Button1 อย่างไรก็ตามชื่อเมธอดนี้ นักพัฒนาระบบสามารถเปลี่ยนแปลงได้
คำสั่ง Var	เป็นการกำหนดตัวแปรชื่อ Form1 ซึ่งได้ต้นแบบมาจาก TForm1 ซึ่งได้กำหนดไว้ข้างต้น

รายละเอียดของส่วน Implementation

คำสั่ง

```
procedure Tform1.Button1Click(Sender:Tobject);
begin
   Edit1.Text:='Hello World";
end;
```

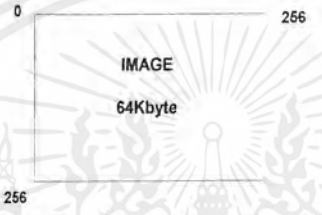
เป็นส่วนที่แสดงรายละเอียดการทำงานของเมธอดที่ได้กำหนดไว้ข้างต้น คือ Button1Click ส่วน Tform1 ที่อยู่ด้านหน้าก็จะเป็นการบอกว่า Button1Click เป็นเมธอดของ Object TForm1 สำหรับ (Sender:Tobject) จะเป็น Object ของคอนโทรลที่เป็นผู้ที่สั่งการทำงานนี้ซึ่งในที่นี้คือ Button1 โดยที่ Runtime Library (RTL) จะทำการส่ง Object นี้ผ่านทางพารามิเตอร์มาให้กับโพรซีเดอร์นี้ ส่วนคำสั่งงานที่อยู่ระหว่าง begin....end ก็จะเป็นคำสั่งจะถูกสั่งให้ทำงานต่อไป ซึ่งก็คือ Edit1.Text:='Hello World'; ก็จะเป็นการกำหนดค่าพร้อบเพอร์ตี้ Text ของ Edit1 ให้มีค่าเท่ากับ 'Hello World' ซึ่งจะมีผลทำให้แสดงค่า Hello World ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

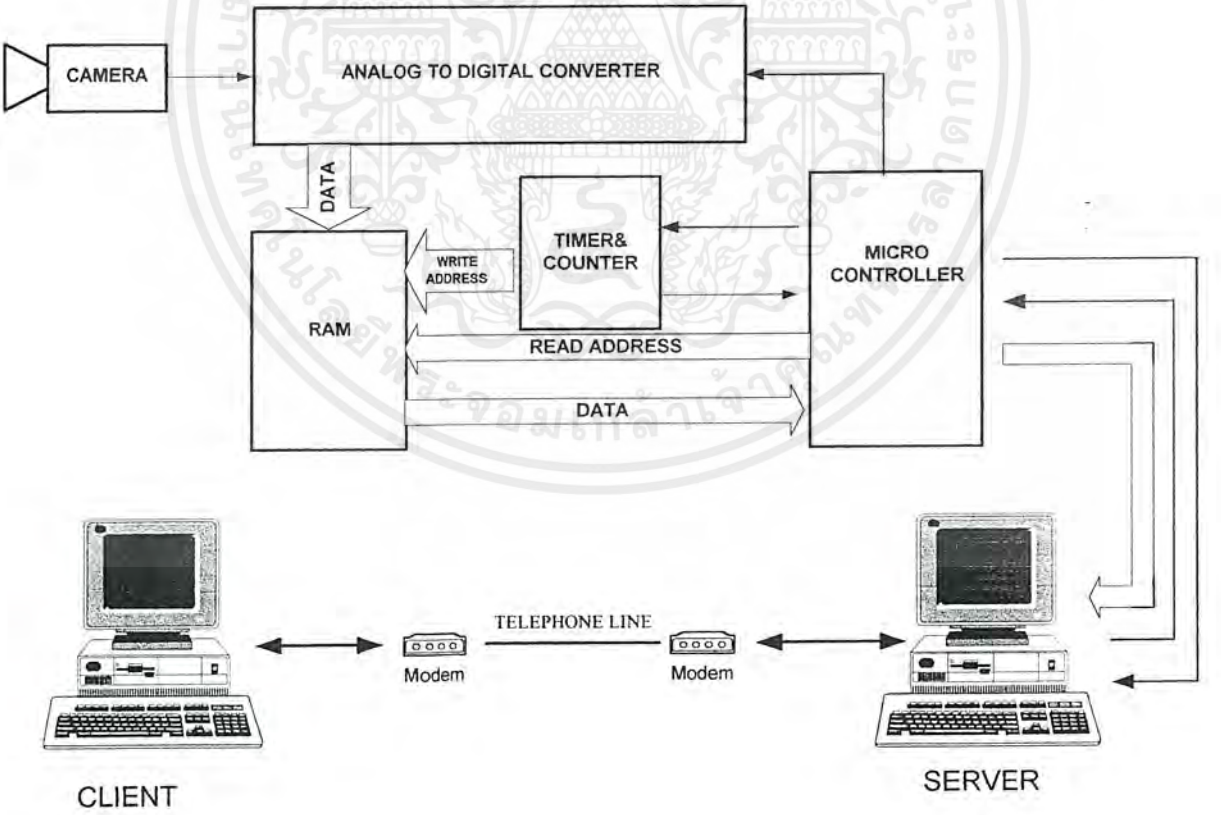
บทที่ 3
การคำนวณและการสร้าง

3.1 แนวคิดและการทำงาน

จุดมุ่งหมายในการออกแบบระบบรักษาความปลอดภัยผ่านสายโทรศัพท์(Security System Via Telephone Line) ก็เพื่อนำสัญญาณภาพรวม(Composite Signal)ผ่านสายโทรศัพท์ไปแสดงผลยังฝ่ายรับได้ โดยการออกแบบจะต้องทำการแปลงสัญญาณภาพซึ่งเป็นสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลก่อน เพื่อให้สามารถส่งไปในเครือข่ายโทรศัพท์ซึ่งจะทำให้การความผิดพลาดของการส่งส่งข้อมูลน้อยกว่าการส่งสัญญาณที่เป็นสัญญาณอนาลอกไปเลย โดยการแปลงสัญญาณให้เป็นสัญญาณดิจิตอลขนาด 8 บิตต่อ 1 จุดภาพ ซึ่งในการแสดงผลใน 1 ภาพจะแสดงเพียง 1 ฟิลด์โดยจะแสดง 256 จุดต่อ 1เส้นสแกน และจะแสดง 256 เส้นสแกน เพราะฉะนั้นใน 1ภาพจะมีจำนวน 256*256*8 =524288 บิต (64 Kbyte)



รูปที่ 3.1 แสดงข้อมูลแต่ละจุดบนจอภาพ



รูปที่ 3.2 แสดง Block Diagram การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หลักการทำงาน

จากรูปที่ 3.2 เมื่อเริ่มรันโปรแกรมที่เครื่อง Server โปรแกรมจะสั่งให้ไมโครคอนโทรลเลอร์สั่งงานให้ระบบทำการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลด้วยสัญญาณนาฬิกา 5 MHz แล้วเก็บข้อมูลภาพไว้ในหน่วยความจำขนาด 64 Kbyte โดยใช้วงจรนับเป็นตัวอ้างแอดเดรส จากนั้นไมโครคอนโทรลเลอร์จะทำการอ่านข้อมูลภาพจากหน่วยความจำแล้วส่งข้อมูลนั้นผ่านพอร์ตขนานเข้าสู่ความพิวเตอร์ จากนั้นคอมพิวเตอร์จะนำข้อมูลภาพมาเรียงเป็นไฟล์ภาพแล้วเก็บแต่ละภาพไว้ภายในคอมพิวเตอร์ เมื่อเครื่องคอมพิวเตอร์ Client ต้องการดูภาพก็ทำการติดต่อเข้ามายังเครื่อง Server แล้วทำการอ่านไฟล์ภาพเก็บไว้ในเครื่อง Server

3.3 วงจรกำเนิดความถี่

จากรูปที่ 3.3 การทำงานของวงจรกำเนิดความถี่ จะประกอบด้วย U2 (Oscillator 10MHz) ซึ่งเป็นตัวกำเนิดสัญญาณนาฬิกา 10 MHz และ U3A(74LS74) ทำหน้าที่หารความถี่ โดยจะหารสองความถี่ที่ มาจาก U2 ให้เหลือความถี่ 5 MHz เพื่อป้อนให้วงจร ADC และวงจร COUNTER ต่อไป

3.4 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (ANALOG TO DIGITAL CONVERTER:ADC)

จากรูปที่ 3.3 การทำงานของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล จะเป็นวงจรที่รับสัญญาณภาพที่มาจากกล้องวีดีโอ(Composite Video) สัญญาณนี้จะผ่านเข้าสู่วงจรป้องกันความผิดพลาดของสัญญาณซึ่งประกอบด้วย R1(75 โอมห์) ทำหน้าที่ปรับอิมพีแดนซ์ของวงจรให้เหมาะสมกับอิมพีแดนซ์ของกล้องวีดีโอ จากนั้นสัญญาณจะส่งผ่าน C1 ซึ่งทำหน้าที่ตัดแรงดันไฟตรงออก โดยจะมี เฉพาะสัญญาณกระแสสลับผ่านไปเท่านั้น หลังจากนั้นสัญญาณจะยกระดับด้วย D1 และ VR1 เพื่อลดระดับสัญญาณรบกวนที่จะเกิดกับสัญญาณภาพ สัญญาณที่ยกระดับแล้วจะถูกส่งผ่าน R2 และ C2 แล้วเข้าวงจรขยายซึ่งประกอบด้วย Q1(BC547) และ R2 เพื่อขยายสัญญาณก่อนที่จะส่งเขาสู่ ขา 21(Vin) ของ U1 (CA3318) ซึ่งเป็นไอซีที่ทำหน้าที่แปลงสัญญาณภาพที่เป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล โดยไอซีตัวนี้จะทำงานได้จะต้องประกอบด้วย

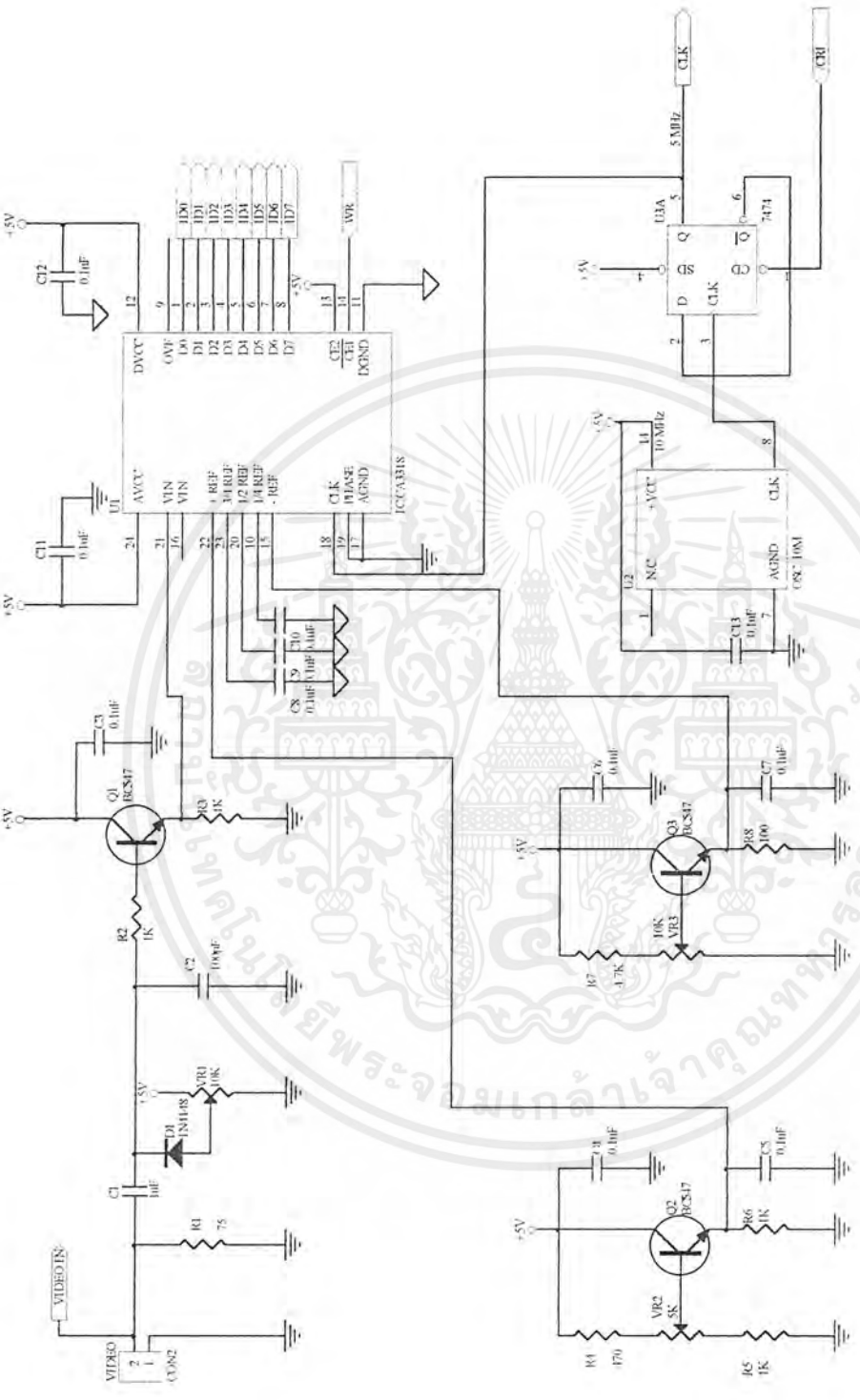
1. สัญญาณอินพุท (ขา 21) ซึ่งเป็นสัญญาณที่ผ่านจากการขยายของ Q1 และ R2
2. ระดับแรงดันอ้างอิงด้านบวก(ขา22)จะได้จากวงจรที่ประกอบด้วยR4,R5,R6,VR2,C4,C5 และ Q2(BC547) ซึ่งสามารถปรับค่าแรงดันอ้างอิงด้านบวกได้ตั้งแต่ 0.17-4 V ซึ่งจะถูกปรับค่าให้มีขนาดแรงดันเท่ากับแรงดันสูงสุดของสัญญาณอินพุท
3. ระดับสัญญาณอ้างอิงด้านลบ(ขา15)จะได้จากวงจรที่ประกอบด้วย R7,R8,VR3,C6,C7 และ Q3(BC 547) ซึ่งจะถูกปรับค่าให้มีขนาดแรงดันเท่ากับแรงดันต่ำสุดของสัญญาณอินพุท
4. ระดับสัญญาณอ้างอิงที่เหลือคือ ¾ , ½ และ ¼ เท่า (ขา 23,20,10 ตามลำดับ)นั้นไม่ได้ถูกใช้งาน ดังนั้นจึงต่อผ่าน C8,C9,C10 ลงกราวด์ตามลำดับ เพื่อป้องกันสัญญาณรบกวนที่อาจเกิดขึ้นได้
5. สัญญาณนาฬิกา(ขา 18) จะได้จากวงจรความถี่ขนาด 5 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. สัญญาณควบคุม จะต้องมีการควบคุมการทำงานของไอซีที่ขา /CE1(ขา 14)และCE2(ขา 13) เพื่อทำให้ไอซีสามารถทำงานได้

สัญญาณภาพที่เป็นสัญญาณอนาล็อกเมื่อผ่านเข้า U1 แล้วจะถูกแปลงให้เป็นสัญญาณดิจิตอล ขนาด 8 Bit ซึ่งมีความแตกต่างสัญญาณ 256 ระดับ หลังจากนั้นจะถูกส่งไปยังวงจรหน่วยความจำต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



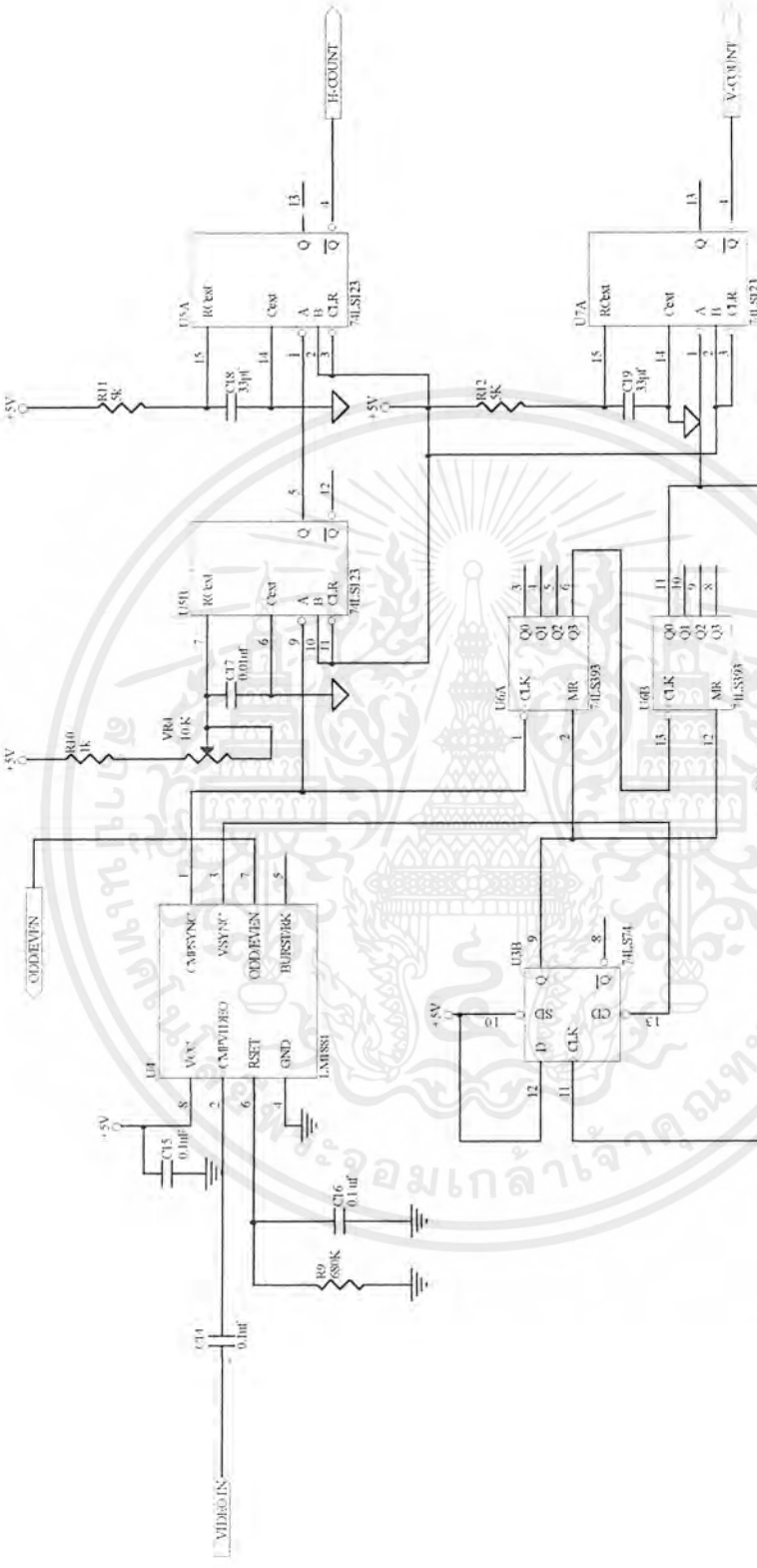
รูปที่ 3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจรแยกซิงค์ (SYNC SEPERATOR)

จากรูปที่ 3.4 การทำงานของวงจรแยกซิงค์จะทำหน้าที่แยกซิงค์ออกจากสัญญาณที่มาจากกล้องวีดีโอ โดยสัญญาณอินพุท ซึ่งเป็นสัญญาณ Composite Video จะเข้าที่ขา 2 ของ U4 (LM1881) ซึ่งเป็นไอซีที่ทำหน้าที่แยกสัญญาณซิงค์ออกจากสัญญาณภาพ โดยมี C14 ทำหน้าที่ป้องกันแรงดันไฟฟ้ากระแสตรง นอกจากนี้ U4 จะทำงานได้จะต้องกับ R9 และ C16 เพื่อกำหนดช่วงเวลาของวงจรผลิตความถี่ภายใน สัญญาณที่ออกจาก U4 จะเป็นสัญญาณซิงค์รวม(ขา 1) สัญญาณซิงค์ทางแนวตั้ง (ขา 3) และสัญญาณฟิลด์คู่และสัญญาณฟิลด์คี่ (ขา 7) สัญญาณซิงค์รวมจะถูกส่งไปยังวงจรหน่วงเวลาทางด้านแนวนอนซึ่งประกอบด้วย U5B(74LS123),R10,VR4,C17 ซึ่งจะทำการหน่วงซิงค์รวมประมาณ 9 uS หลังจากนั้นสัญญาณจะออกมาทางขา 5 ของ U5B จะถูกส่งเข้าสู่วงจรผลิตพัลส์เล็กๆ ประมาณ 140 nS (H- Count) ซึ่งประกอบด้วย U5A(74LS123),R11,C18 โดยสัญญาณพัลส์จะออกมาทางขา 4 ของ U5A ซึ่งสัญญาณนี้จะถูกส่งไปยังวงจรควบคุมการนับทางแนวนอนเพื่อทำให้วงจรนับทางแนวนอนเริ่มทำการนับ วัตถุประสงค์ของวงจรหน่วงเวลาก็เพื่อให้วงจรนับทางแนวนอนเริ่มทำการนับที่เวลาเริ่มต้นของการสแกนเท่ากันทุกเส้นสแกน นอกจากนี้สัญญาณซิงค์รวมจะถูกส่งไปยังวงจรควบคุมการนับเพื่อหน่วงเวลาให้กับสัญญาณซิงค์ทางแนวตั้ง วงจรนับซิงค์ประกอบด้วย U3B(74LS74),U6A(74LS123),U6B(74LS123) โดยสัญญาณซิงค์ทางแนวตั้งจะต่อเข้ากับขารีเซ็ทของชุดนับซิงค์รวม เมื่อสัญญาณซิงค์ทางแนวตั้งเข้ามาจะทำให้ เกิดการรีเซ็ทวงจรนับซิงค์รวม แล้วจะเกิดสัญญาณออกมาที่ขา 9 ของ U3B ทำให้ U6A และ U6B(74LS393) เริ่มทำการนับสัญญาณซิงค์รวมนั้น ซึ่งจะทำการนับ 32 (ประมาณ 2 mS) แล้วจะถูกส่งเข้าวงจรผลิตพัลส์เล็กๆ ซึ่งประกอบด้วย U7A(74LS123) ,R12,C19 เพื่อทำการผลิตพัลส์เล็กๆประมาณ 140 nS ออกมาที่ขา 4 ของ U7A (V- Count) โดยสัญญาณพัลส์นี้จะถูกส่งไปรีเซ็ทวงจรนับทางแนวตั้งเพื่อทำให้เกิดการนับซิงค์ทางแนวตั้งต่อไปและจะทำการนับซิงค์อีกครั้งเมื่อมีสัญญาณซิงค์ทางแนวตั้งลูกใหม่เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



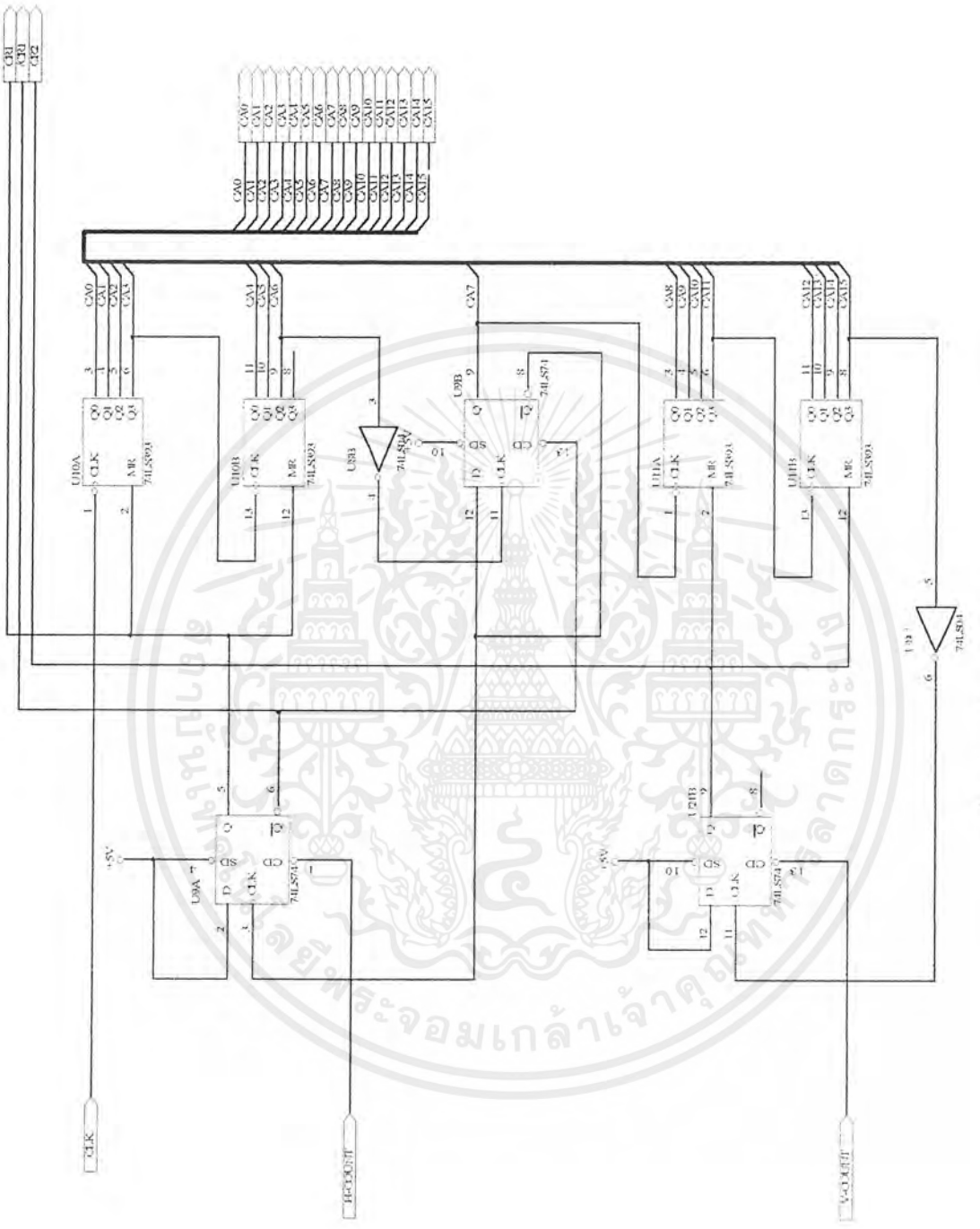
รูปที่ 3.4 วงจรแยกสัญญาณซิงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจรนับ (COUNTER)

จากรูปที่ 3.5 วงจรนับและวงจรควบคุมทั้งแนวตั้งและแนวนอน จะประกอบด้วยU9A,U9B,U21B (74LS74)และ U10A,U10B,U11A,U11B(74LS 393) เมื่อมีสัญญาณนาฬิกาความถี่ 5 MHz ที่มาจากวงจรกำเนิดความถี่ จะถูกส่งเข้าขา 1 ของ U10A ซึ่งเป็นวงจร Binary Asynchronous Counter ขนาด 4 Bit และเอาท์พุท CA3 (ขา 6) ถูกต่อเข้ากับขาอินพุท(ขา13) ของ U10B ขาเอาท์พุท CA 6(ขา 9) ของ U10B จะถูกต่อกับ U9B(74LS74) ผ่าน U8B(74LS04) เอาท์พุท Q (CA7)ของ U9B จะถูกต่อเข้ากับอินพุท(ขา 1) ของ U11A และเอาท์พุท CA11 จะต่อกับขาอินพุท(ขา 1)ของ U11B ทั้งนี้เพื่อให้สัญญาณเอาท์พุท ของวงจรนับรวมทั้งหมด 16 เส้น เพื่อให้สามารถใช้อ้างแอดเดรสของหน่วยความจำซึ่งใช้ในการเก็บข้อมูลสัญญาณภาพ 1 ฟิลด์ ซึ่งใช้หน่วยความจำขนาด 64 Kbyte เส้นสัญญาณที่ใช้ในการอ้างแอดเดรสจะแบ่งออกเป็น 2 ส่วน คือ ส่วนแรกใช้อ้างแอดเดรสหน่วยความจำภาพทางแนวนอนจำนวน 8 เส้น ประกอบด้วย U10A,U10B และ U9B ใช้สำหรับอ้างตำแหน่งหน่วยเก็บความจำภาพทางแนวนอน(1เส้นสแกนมีขนาด 256 จุดภาพ) ส่วนที่สองใช้อ้างแอดเดรสหน่วยความจำภาพทางแนวตั้งจำนวน 8 เส้น ประกอบด้วย U11A,U11B ใช้สำหรับอ้างตำแหน่งหน่วยเก็บความจำภาพทางแนวตั้ง (จำนวนเส้นสแกนภาพ) ซึ่งสามารถอ้างได้ 256 เส้น ส่วนอ้างตำแหน่งหน่วยความจำนี้จะต้องมีส่วนควบคุมการนับทั้งแนวนอนและแนวตั้ง โดย U9A ใช้ควบคุมการนับการนับทางแนวนอนซึ่งจะรับสัญญาณ H-Count จากวงจรแยกซิงค์ ส่วน U21B ใช้ควบคุมการนับทางแนวตั้ง ซึ่งจะรับสัญญาณ V-Count จากวงจรแยกซิงค์เช่นกัน วงจรนับทางแนวนอนจะเริ่มนับก็ต่อเมื่อเอาท์พุท Q (ขา 5)ของ U9A ซึ่งเป็นสัญญาณคบคุมการนับทางแนวนอนเป็นสัญญาณ “0” ให้ U10A และ U10B และสัญญาณ “1”ให้ U9B ส่วนวงจรนับทางแนวตั้งจะเริ่มนับเมื่อเอาท์พุท Q ของ U21B ซึ่งเป็นสัญญาณควบคุมการนับทางแนวตั้ง เป็นสัญญาณ “0” ให้กับ U11Aและ U11B โดยสัญญาณควบคุมการนับทางแนวนอนจะถูกควบคุมจากสัญญาณ H-Count และสัญญาณควบคุมการนับทางแนวตั้งก็จะถูกควบคุมจากสัญญาณ V-Count ถ้าสัญญาณ H-Count และ V-Count นี้เป็น “0” ก็จะทำให้เกิดการเริ่มนับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



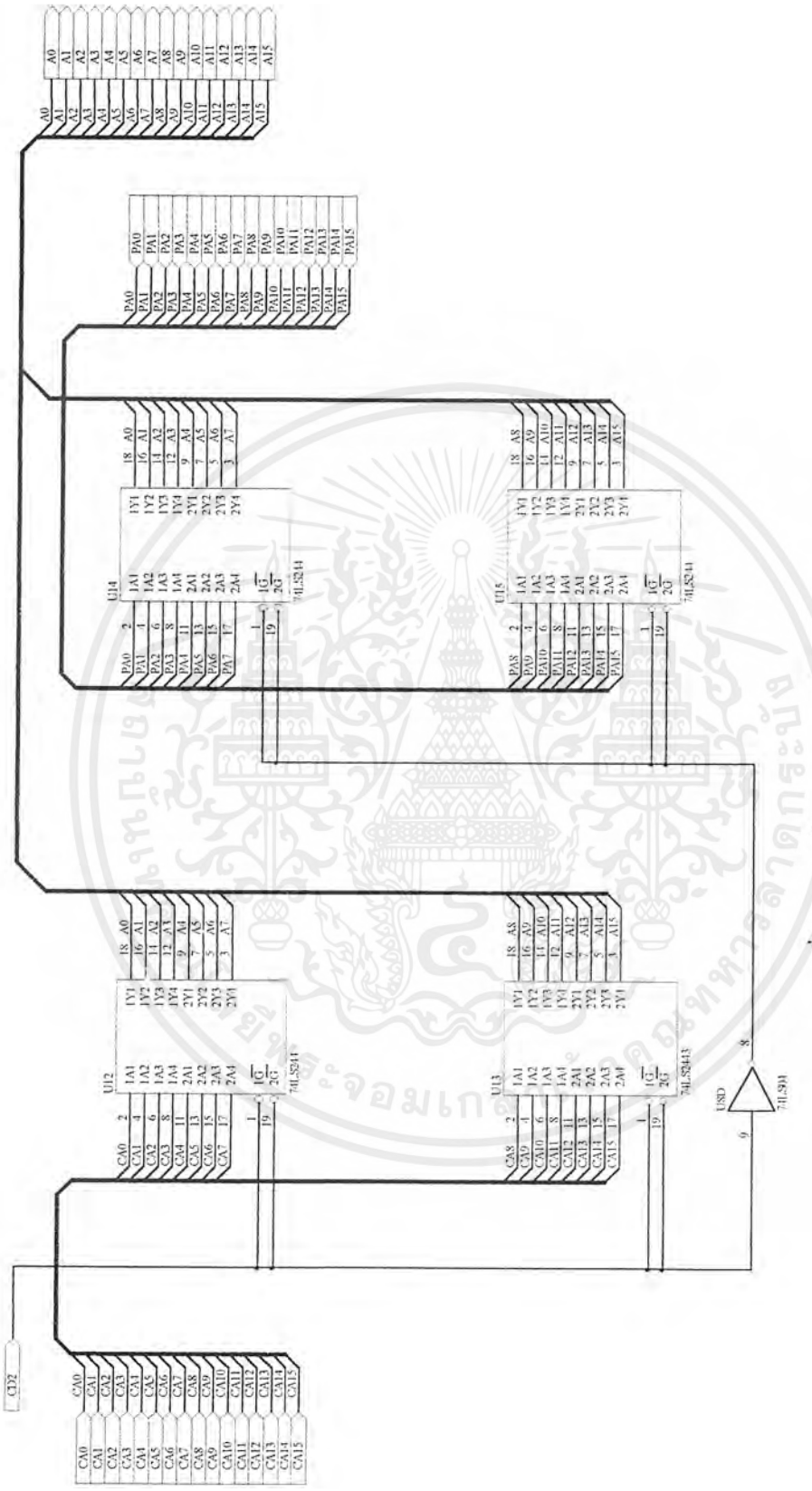
รูปที่ 3.5 วงจรนับ(Counter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 วงจรเลือกแอดเดรสหน่วยความจำ

จากรูปที่ 3.6 วงจรเลือกแอดเดรสหน่วยความจำนี้ จะใช้ในการเลือกว่าจะเขียนหรืออ่านข้อมูลจากหน่วยความจำ ในกรณีแรกคือการเขียนข้อมูลลงในหน่วยความจำซึ่งสายสัญญาณอ้างแอดเดรสหน่วยความจำจะต่อกับวงจรนับ(CA0-CA15) ซึ่งประกอบด้วย U12 และ U13 ทำหน้าที่เป็นบัฟเฟอร์ ในกรณีสองคือการอ่านข้อมูลจากหน่วยความจำซึ่งสายสัญญาณอ้างแอดเดรสหน่วยความจำจะต่อกับ Port 0 และ Port 2 ของ U25(89C51)ในวงจรอินเตอร์เฟส ผ่านทาง U14 และ U15 ซึ่งทำหน้าที่เป็นบัฟเฟอร์เช่นกัน โดยทั้งสองกรณีจะทำงานสลับกัน ซึ่งจะถูกควบคุมจาก CD2 ที่มาจากขา P1.2 (ขา 3) ของ U25 สัญญาณ CD2 จะถูกต่อกับขา 1 และขา 19 (U12 และU13) ส่วน /CD จะถูกต่อกับขา 1 และขา19(U14และU15) เมื่อสัญญาณ CD2 มาเป็น "0" ก็จะทำให้ชุดใดชุดหนึ่ง Enabled ส่วนอีกชุดหนึ่งก็จะเป็น High Impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



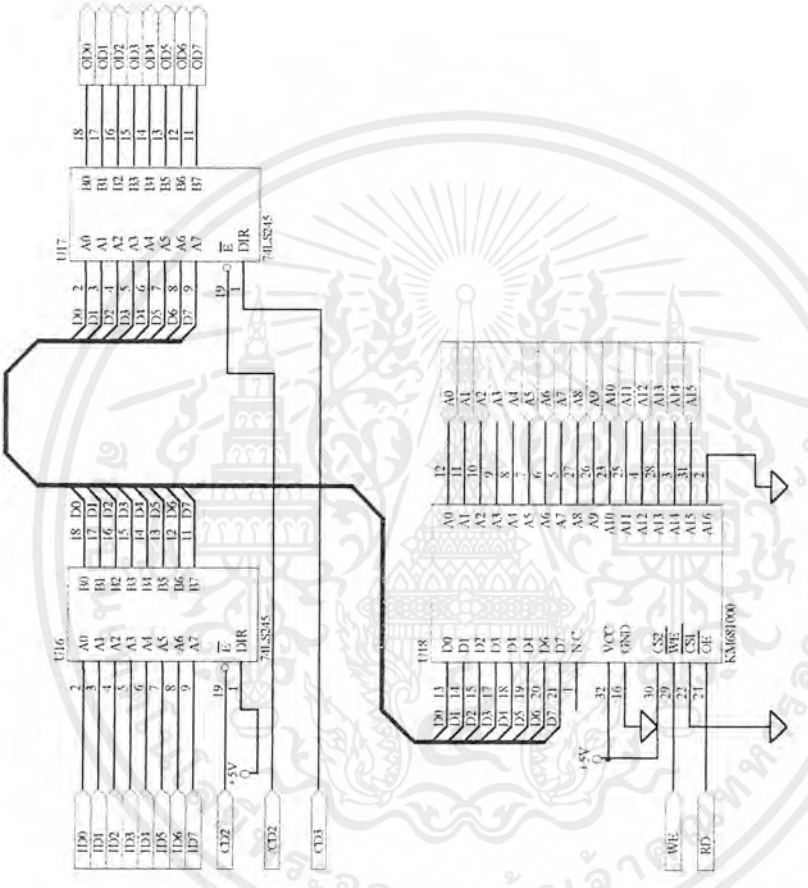
รูปที่ 3.6 วงจรเลือกแอดเดรสหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 วงจรหน่วยความจำภาพ

จากรูปที่ 3.7 วงจรหน่วยความจำภาพ ประกอบด้วย U16 ,U17 (74LS245) และ U18(KM681000) โดย U16 เป็นบัฟเฟอร์ชนิดสองทาง(ซึ่งจะถูกควบคุมทิศทางที่ขา 1) อยู่ระหว่างวงจรแปลงสัญญาณ อนาล็อกเป็นสัญญาณดิจิตอล กับหน่วยความจำภาพ (KM681000) ส่วน U17 จะเป็นบัฟเฟอร์ที่อยู่ ระหว่าง U25(89C51) กับหน่วยความจำภาพ ในการใช้งานเราจะเลือกติดต่อว่าจะเขียนข้อมูลซึ่งจะติดต่อกับวงจร ADC หรือ จะอ่านข้อมูลซึ่งจะติดต่อกับ U25 โดย U16 และ U17 จะทำงานไม่พร้อมกัน การควบคุมการทำงานจะอาศัย CD2 ซึ่งจะเป็นตัวเลือกการทำงานโดยจะต่อกับขา Enabled (ขา19) ของ U16 และจะใช้สัญญาณ /CD2 ต่อกับขา Enabled (ขา19) ของ U17 ดังนั้นถ้าสัญญาณ CD2 เป็น " 0" U16 ก็สามารถส่งผ่านข้อมูลจากวงจร ADC มาเก็บในหน่วยความจำได้ ส่วน /CD2 จะเป็น "1" ทำให้ U17 เป็นไฮอิมพีแดนซ์ ในทางกลับกันถ้า CD2 เป็น "1" ทำให้ U16 เป็นไฮอิมพีแดนซ์ แต่ U17 จะทำงานในการอ่านข้อมูลจากหน่วยความจำไปยัง 89C51และไมโครคอมพิวเตอร์ที่วงจรอินเตอร์เฟส โดยสัญญาณ CD3 จะต้องเป็น "1" ซึ่งสามารถสั่งได้จากขา P1.3 ของ U25 ในวงจรอินเตอร์เฟส ส่วน U18(KM681000) เป็นไอซีที่มีหน่วยความจำขนาด 128 Kbyte แต่ในการใช้งานจะใช้เพียง 64 Kbyteเพื่อใช้เก็บข้อมูลภาพ 1ฟิลด์ ดังนั้นจึงต่อขา A16(ขา12) ลงกราวด์ ซึ่งจะใช้สายสัญญาณอ้างแอดเดรส 16 เส้น (A0-A15) โดยสายสัญญาณนี้จะถูกอ้างอิงจากวงจรนับ และ U25(89C51) การที่หน่วยความจำจะทำงานได้ จะต้องมีสัญญาณอื่นๆ โดยจะต้องต่อขา /WE(ขา 29) กับสัญญาณควบคุมการเขียน และขา /OE (ขา 24) กับขา /RD(ขา17) ของ U25 คือถ้าต้องการเขียนข้อมูล ขา /WE จะต้องเป็น "0" ขา /OE จะต้องเป็น "1" และเมื่อต้องการอ่านข้อมูลขา /OE จะต้องเป็น "0"และขา /WE จะต้องเป็น "1" การอ่านและการเขียนข้อมูลนั้นจะต้องทำงานร่วมกับการกำหนดตำแหน่งบนหน่วยความจำด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



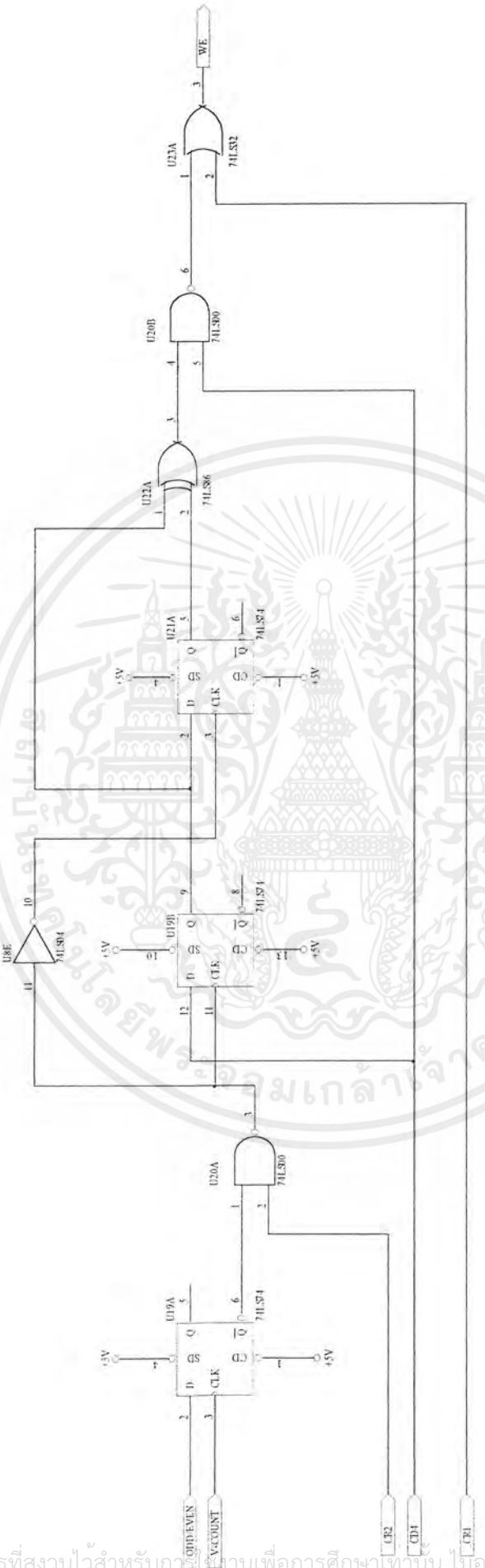
รูปที่ 3.7 วงจรหน่วยความจำภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 วงจรควบคุมการเขียนข้อมูลภาพ

จากรูปที่ 3.8 วงจรควบคุมการเขียนข้อมูลภาพ มีการทำงานดังนี้คือ เมื่อสัญญาณอินพุท(ขา3) ของ U19B เปลี่ยนสถานะจาก "0" เป็น "1" ก็จะเกิดการเปลี่ยนแปลง โดยสัญญาณอินพุทจะได้จากแนนด์ (U20A) กันระหว่างสัญญาณควบคุมทางแนวตั้ง(CR2) และสัญญาณ ODD/EVEN กล่าวคือสัญญาณนี้จะเป็น "0" เมื่อสัญญาณ CR2 เป็น "1" และสัญญาณ ODD/EVEN (ได้จาก ขา Q ของ U19A)เป็น "1" เท่านั้น และสัญญาณนี้จะเป็น "1" เมื่อสัญญาณ CR2 เป็น "1" และ/หรือสัญญาณ ODD/EVEN เป็น "1" ด้วยเหตุนี้ ถ้าเราให้มีการเปลี่ยนแปลงของทางด้านอินพุท ของ U19B จาก "0"เป็น "1" แล้วเอาท์พุท ของ U19B จะเกิดการเปลี่ยนแปลงตามอินพุท เมื่อสัญญาณควบคุมการการนับทางแนวตั้ง(CR2) เป็น "0" จากนั้นเอาท์พุท ของ U19B จะถูกต่อเข้าเป็นอินพุท ให้ U21A และเอาท์พุท ของ U21A จะเกิดการเปลี่ยนแปลงก็ต่อเมื่อสัญญาณที่ขา 3 เปลี่ยนสถานะจาก "0" เป็น "1" แต่ สัญญาณที่ขา 3 ของ U21A จะต่อกลับเฟสกับสัญญาณอินพุท ขา 3 ของ U19B โดยใช้ U8E เป็นตัวกลับเฟส ในช่วงที่สัญญาณ CR2 เป็น "0" หมายถึงช่วงที่วงจรนับทางแนวตั้งกำลังทำการนับเพื่อกำหนดแอดเดรสให้หน่วยความจำภาพ จังหวะที่สัญญาณ CR2 จะเปลี่ยนจาก "1" เป็น "0" ทำให้ สัญญาณที่ขา 3 ของ U19B เปลี่ยนจาก "0"เป็น "1" นั่นหมายถึงถ้าสัญญาณอินพุท ของ U19B มีการเปลี่ยนสถานะก่อนหน้านี้และคงค่าไว้ จะทำให้ เอาท์พุท ของ U19B เกิดการเปลี่ยนแปลงสถานะตามอินพุทในจังหวะนี้ จากที่กล่าวมาแล้วว่าเอาท์พุท ของ U19B จะต่อเข้ากับอินพุท ของ U21A ดังนั้น U21A จะมีการเปลี่ยนแปลงทางเอาท์พุท ตามอินพุทก็ต่อเมื่อสัญญาณ CR2 เปลี่ยนสถานะจาก "0" เป็น "1" และสัญญาณ ODD/EVEN เป็น "1" จากการเปลี่ยนแปลงนี้เราก็จะนำไปควบคุมการเขียนข้อมูลลงในหน่วยความจำ โดยต่อเอาท์พุท ของ U19B และเอาท์พุท ของ U21A เข้ากับ U22A เพื่อเปรียบเทียบข้อมูลกันคือ ถ้าเอาท์พุท ของ U19B และ U21A เหมือนกัน จะทำให้ เอาท์พุท ของ U22A เป็น "0" แต่ถ้าข้อมูลของ U19B และ U21A ต่างกัน เอาท์พุท ของ U22A จะเป็น "1" เอาท์พุท U19B จะต่อเป็นอินพุท ให้กับ U20B โดยอินพุทอีกด้านหนึ่งของของ U20B จะต่อกับ CD4 (ขา P1.4 ของ U25) U20จะให้ค่าเอาท์พุท เปลี่ยนแปลงตามอินพุท ที่ ได้จาก U22A หรือไม่ก็ขึ้นอยู่กับสัญญาณควบคุม CD4 ถ้า CD4 เป็น "1" สัญญาณเอาท์พุท ของ U20B จะเปลี่ยนแปลงตามอินพุทที่ต่อกับ U22A แต่ถ้า CD4 เป็น "0" เอาท์พุท ของ U22A จะมีสถานะเป็น "0" ได้ เมื่อสัญญาณ CD4 มีค่าเป็น "1" และเอาท์พุท ของ U22A เป็น "1" (ซึ่งจะเกิดในช่วงที่ข้อมูลที่อินพุท ของ U19B ต่อเข้ากับอินพุท ของ U23A และอินพุทอีกขาหนึ่งของ U23A ถูกต่อเข้ากับสัญญาณควบคุมการนับทางแนวนอน(CR1) ซึ่งเอาท์พุท ของ U23A จะเป็น "0" ได้ต่อเมื่อสัญญาณ(CR1) เป็น "0" และสัญญาณเอาท์พุท ของ U20B เป็น "0" เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



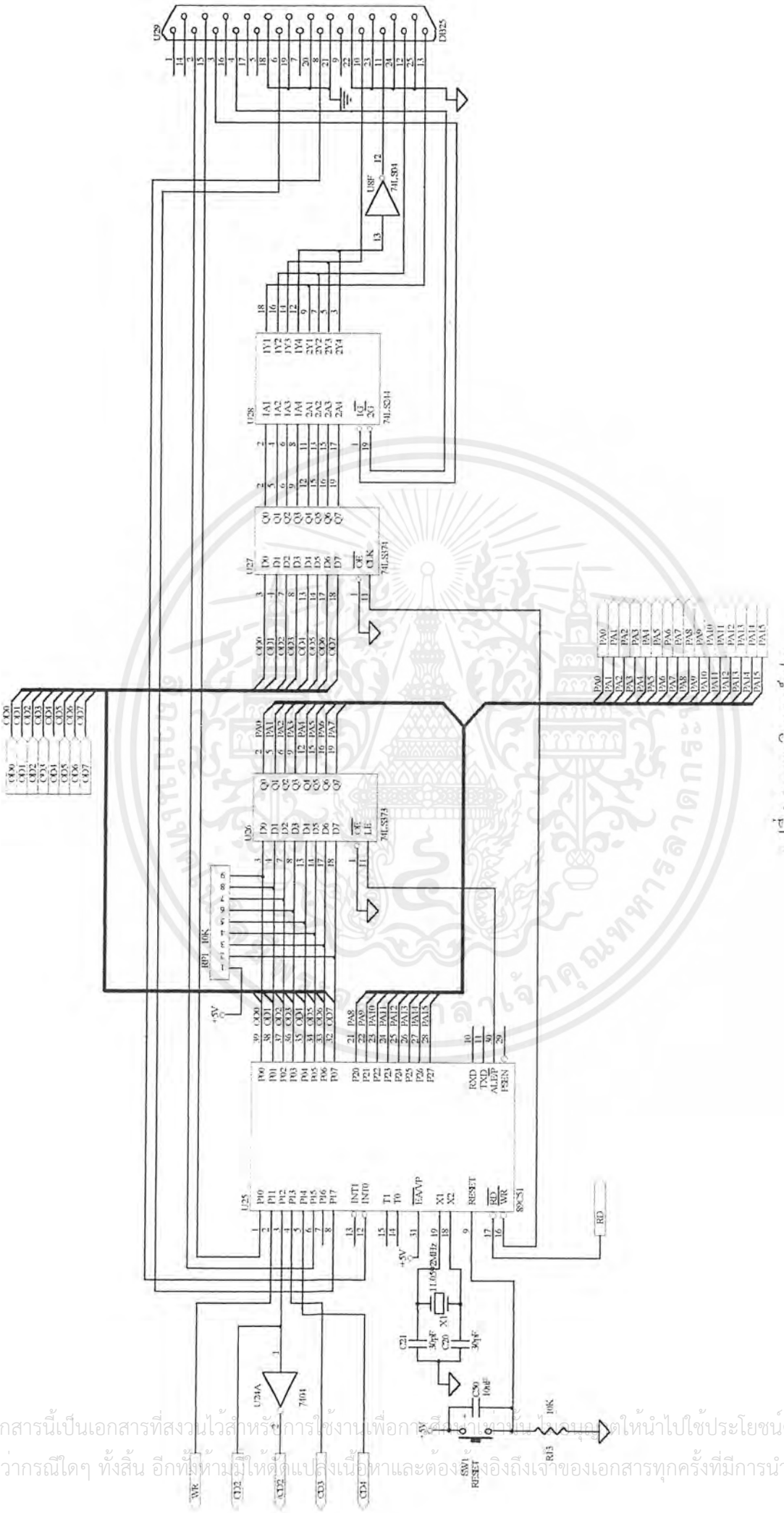
รูปที่ 3.8 วงจรควบคุมการเขียนข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 วงจรอินเตอร์เฟส

จากรูปที่ 3.9 การทำงานของวงจรอินเตอร์เฟส จะขึ้นอยู่กับ U25(89C51) ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 เริ่มต้น U25 จะรอรับลอจิก "0" จากคอมพิวเตอร์ จนกระทั่งคอมพิวเตอร์ส่งลอจิก "0" มาที่ ขา P1.5 (ขา6) จากนั้น U25 จะส่งสัญญาณควบคุมออกมาที่ P1.1(WR),P1.2(CD2),P1.3 (CD3),P1.4(CD4) เพื่อควบคุมให้เกิดการเขียนและอ่านข้อมูลภาพจากหน่วยความจำ ในการเขียนข้อมูลภาพ U25 จะไม่ทำการเขียนโดยตรงเนื่องจากความเร็วการทำงานของไมโครคอนโทรลเลอร์ไม่เร็วพอ ดังนั้นไมโครคอนโทรลเลอร์จะควบคุมให้วงจรเขียนข้อมูลภายนอกทำงานซึ่งเราได้ออกแบบไว้แล้ว หลังจากนั้นไมโครคอนโทรลเลอร์จะอ่านข้อมูลจากหน่วยความจำไปเก็บยังคอมพิวเตอร์ โดยจะทำการอ้างแอดเดรสไบต์ต่ำโดยใช้ Port 0 (P1.0-P1.7) แล้วทำการ Latch ด้วย U26 (74LS373) หลังจากนั้นจะอ้างแอดเดรสไบต์สูงโดยใช้ Port 2(P2.0-P2.7) หลังจากนั้น ขา /RD จะถูกให้ทำงาน(Logic 0) ข้อมูลจะถูกดึงจากหน่วยความจำมารอที่ U27 ซึ่งทำหน้าที่เป็นแลทช์ข้อมูล หลังจากนั้น U25 จะส่งลอจิก "0" ออกจากขา P1.0 เพื่อบอกให้คอมพิวเตอร์อ่านข้อมูลเข้าทางพอร์ทพรินท์เตอร์ที่ขา 10,11,12,13 ซึ่งคอมพิวเตอร์จะอ่านข้อมูลครั้งละ 4 บิต โดยจะอ่าน 2 ครั้ง โดยคอมพิวเตอร์จะส่งสัญญาณมาควบคุมที่ขา 1 และ 19 ของ U28 ซึ่งเป็นบัฟเฟอร์ เมื่ออ่านข้อมูลครบ 8บิต คอมพิวเตอร์จะส่งสัญญาณลอจิก "0" มาที่ขา P1.7 ของ U25 เอให้ส่งข้อมูลไปให้ ครบ 64 Kbyte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

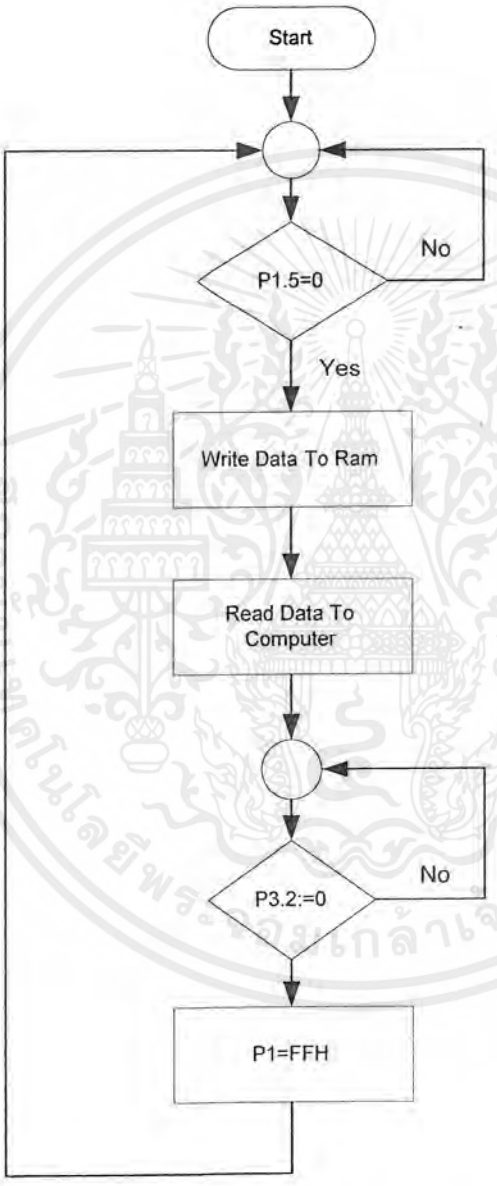


รูปที่ 3.9 วงจรอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 การเขียนโปรแกรมบนไมโครคอนโทรลเลอร์

การเขียนโปรแกรมบนไมโครคอนโทรลเลอร์จะเขียนด้วยภาษา ASSEMBLY การเขียนจะเริ่มจากการรอรับคำสั่งจากคอมพิวเตอร์ เมื่อได้รับคำสั่งจะทำการสั่งให้ เริ่มทำการ Sampling แล้วทำการส่งข้อมูลไปยังคอมพิวเตอร์ โดยการติดต่อจะติดต่อผ่านพอร์ตพรินซ์เตอร์ ซึ่งจะทำการส่งข้อมูลไปยังคอมพิวเตอร์ โดยคอมพิวเตอร์จะทำการรับข้อมูลครั้งละ 4Bit ซึ่ง MCS51 จะส่งข้อมูลจนครบ 64 Kbyte แล้วจะรอคำสั่งจากคอมพิวเตอร์อีกเพื่อเก็บข้อมูลครั้งต่อไป แผนภูมิการทำงานสามารถแสดงได้ดังรูปที่ 3.10



รูปที่ 3.10 แสดงแผนภูมิการทำงานของโปรแกรมไมโครคอนโทรลเลอร์

3.12 การออกแบบส่วนติดต่อรับส่งข้อมูล

ในการรับส่งข้อมูลจะใช้ ระบบ Remote Access Service ซึ่งจะทำการติดตั้งบน Windows NT Server โดยจะทำการสื่อสารผ่านสายโทรศัพท์และโมเด็มซึ่งจะเลือกใช้พอร์ตอนุกรม ข้อมูลภาพที่อ่านได้

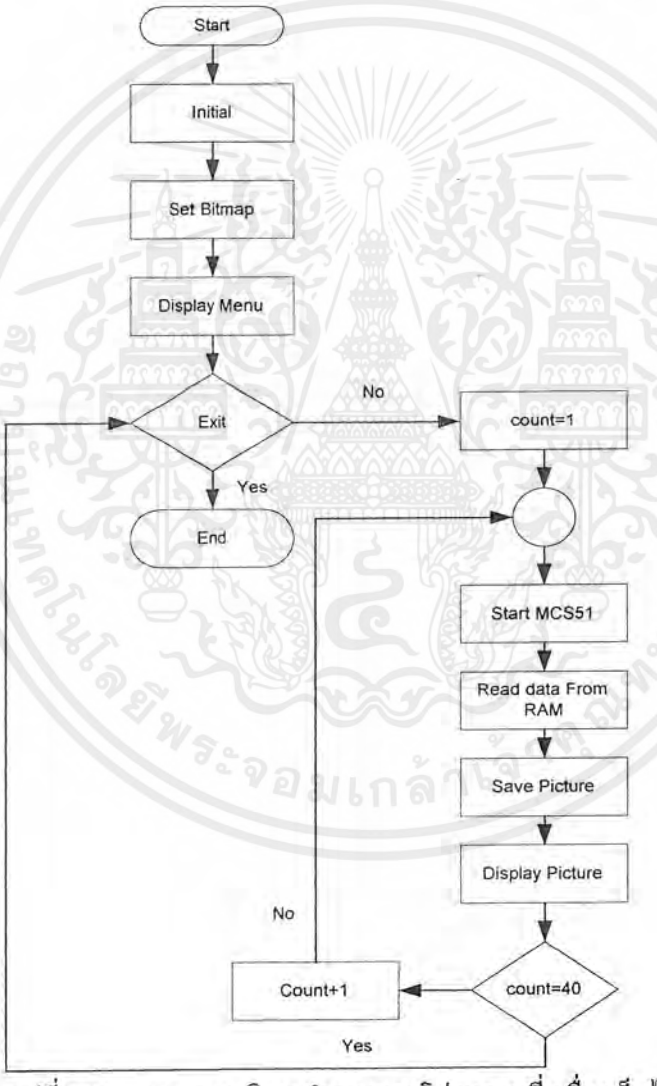
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก RAM จะทำการเรียงเป็นไฟล์ภาพที่เป็น BMP FILES และบันทึกเก็บไว้ที่เครื่อง Windows NT Server ซึ่งจะรอรับการติดต่อจากเครื่องลูกข่ายซึ่งจะทำการติดต่อเข้ามาเพื่อทำการเพื่อทำการอ่านไฟล์ภาพที่ได้ทำการเก็บไว้ ซึ่งระบบการติดต่อระบบ Windows NT Serverจะเป็นตัวจัดการควบคุมการติดต่อ ส่วนการอ่านไฟล์ภาพจะใช้โปรแกรมที่เขียนขึ้นสำหรับโครงงานนี้ซึ่งเขียนด้วย Delphi5

3.13 การออกแบบโปรแกรม

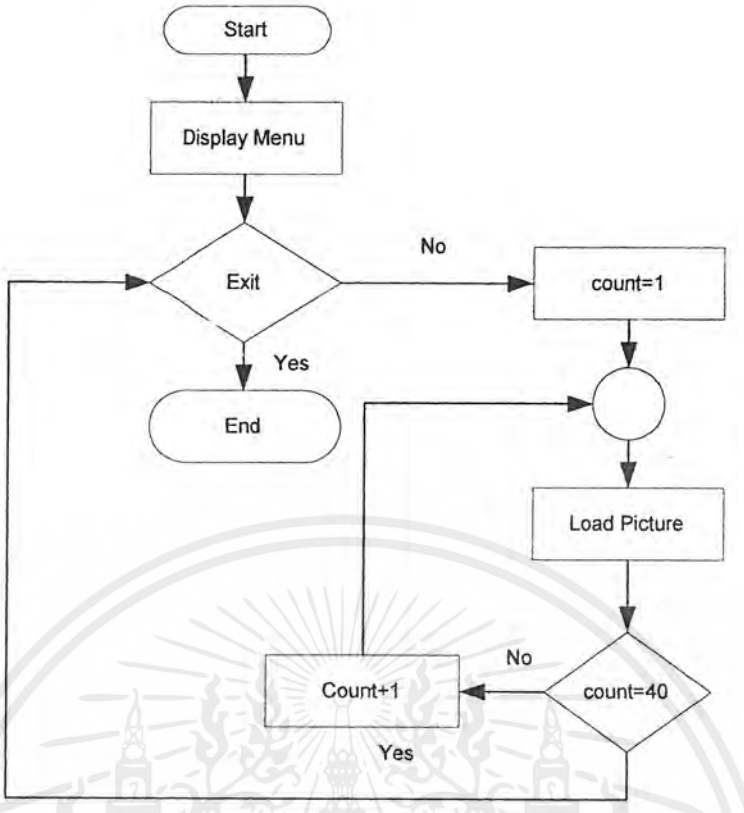
สำหรับโปรแกรมที่ใช้ในโครงงานนี้จะเขียนด้วย Delphi 5 โดยจะแบ่งออกเป็น 2โปรแกรมคือ โปรมแกรมที่เครื่องเก็บไฟล์ภาพและโปรแกรมที่เครื่องอ่านไฟล์ภาพ

1. โปรแกรมสำหรับเครื่องเก็บไฟล์ภาพ ซึ่งสามารถเขียนแผนภูมิการทำงานได้ดังรูปที่ 3.11
2. โปรแกรมสำหรับเครื่องอ่านไฟล์ภาพ ซึ่งสามารถเขียนแผนภูมิการทำงานได้ดังรูปที่ 3.12



รูปที่ 3.11 แสดงแผนภูมิการทำงานของโปรแกรมที่เครื่องเก็บไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงแผนภูมิการทำงานของโปรแกรมที่เครื่องอ่านไฟล์ภาพ

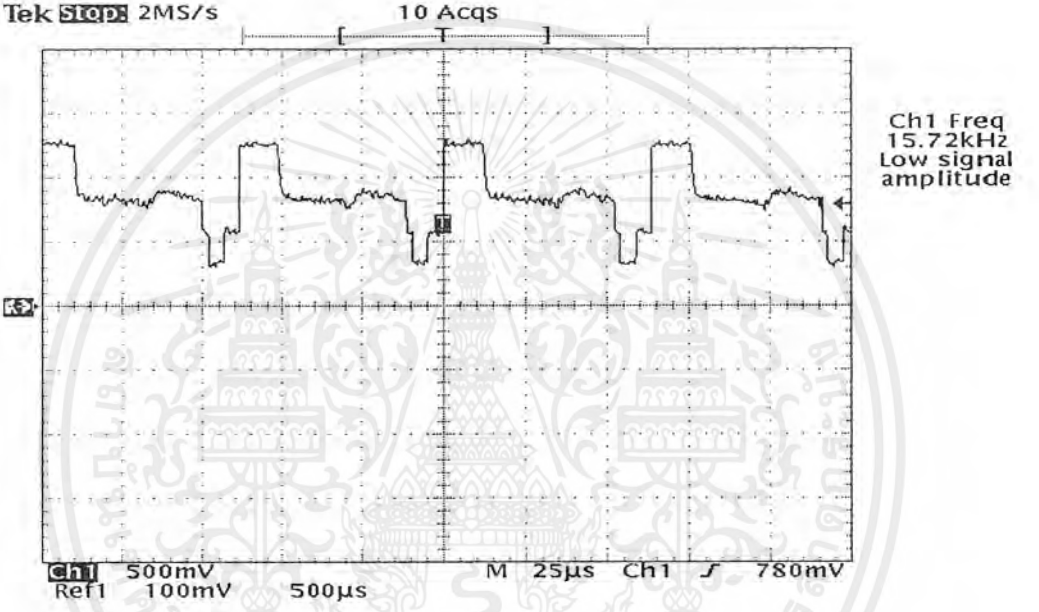
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล(Analog To Digital Converter)

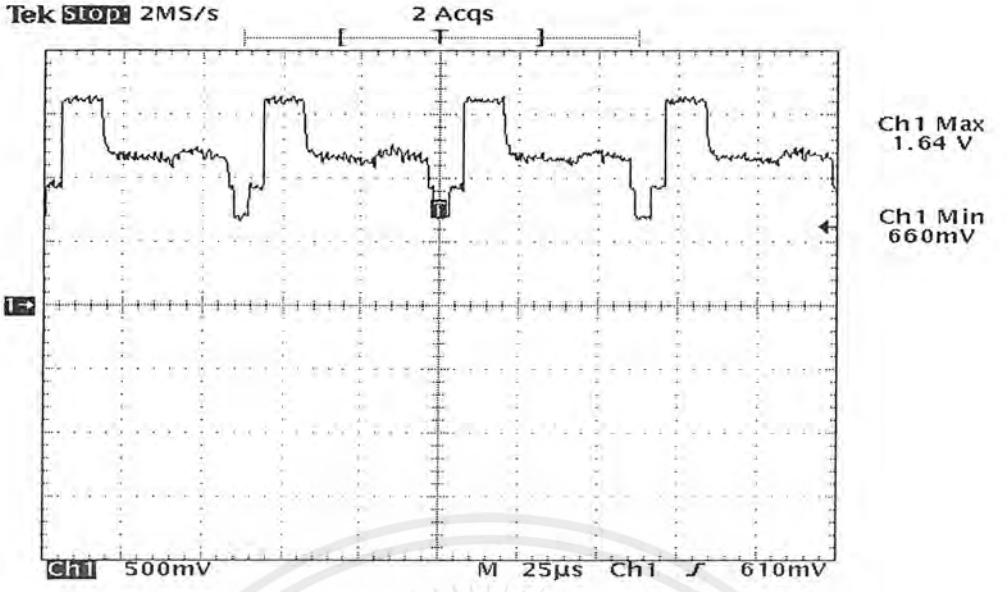
การทดลองในภาคนี้จะทำการวัดสัญญาณภาพขาว-ดำ ที่ได้จากจากกล้องวีดีโอ ดังรูปที่ 3.2 ซึ่งจะถูกแปลงเป็นสัญญาณดิจิตอล 8 บิต โดยใช้ IC CA3318 (U1) เป็น IC อุปกรณ์หลักในการแปลงสัญญาณ โดยใช้ความถี่แซมปลิ้ง 5 MHz ซึ่งถูกควบคุมจากวงจรกำเนิดสัญญาณนาฬิกา 5 MHz ผลการวัดที่ได้มีดังนี้



รูปที่ 4.1 แสดงสัญญาณภาพขาว-ดำที่ได้จากกล้องวีดีโอ

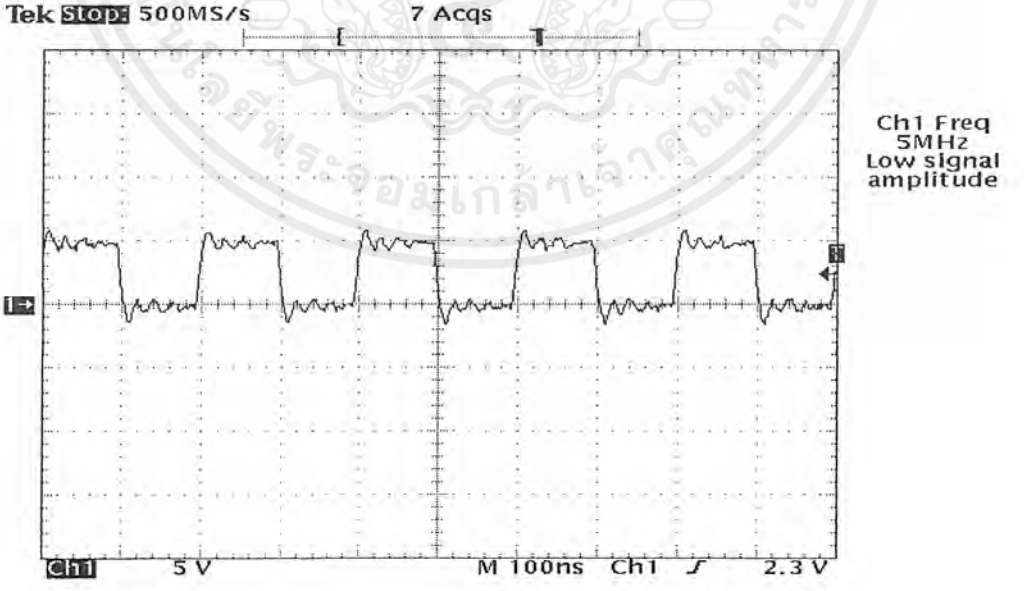
รูปที่ 4.1 ได้แสดงผลการวัดสัญญาณภาพที่ได้จากกล้องวีดีโอ โดยสัญญาณภาพที่ได้เป็นสัญญาณภาพขาว-ดำ มีความถี่ประมาณ 15625 Hz ซึ่งในภาพจะเป็นส่วนประกอบของเส้นภาพแต่ละเส้น โดยมีฮอร์ชิงค์เป็นสัญญาณควบคุมการสบัดกลับทางแนวนอนแทรกอยู่ระหว่างเส้นภาพ จากรูปสัญญาณฮอร์ชิงค์จะอยู่ส่วนล่างสุดระหว่างสัญญาณภาพแต่ละเส้น ส่วนที่อยู่ระดับบนสุดของสัญญาณภาพจะแสดงถึงสัญญาณภาพที่มีสีขาว ส่วนที่อยู่ระดับล่างลงมาเรื่อยจะแสดงถึงระดับความสว่างภาพที่ลดลงมาเรื่อย ๆ จน ถึงส่วนล่างสุดเป็นสัญญาณสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงสัญญาณภาพที่ผ่านการยกระดับสัญญาณเข้าสู่ขาอินพุท (ขา21) ของIC CA3318

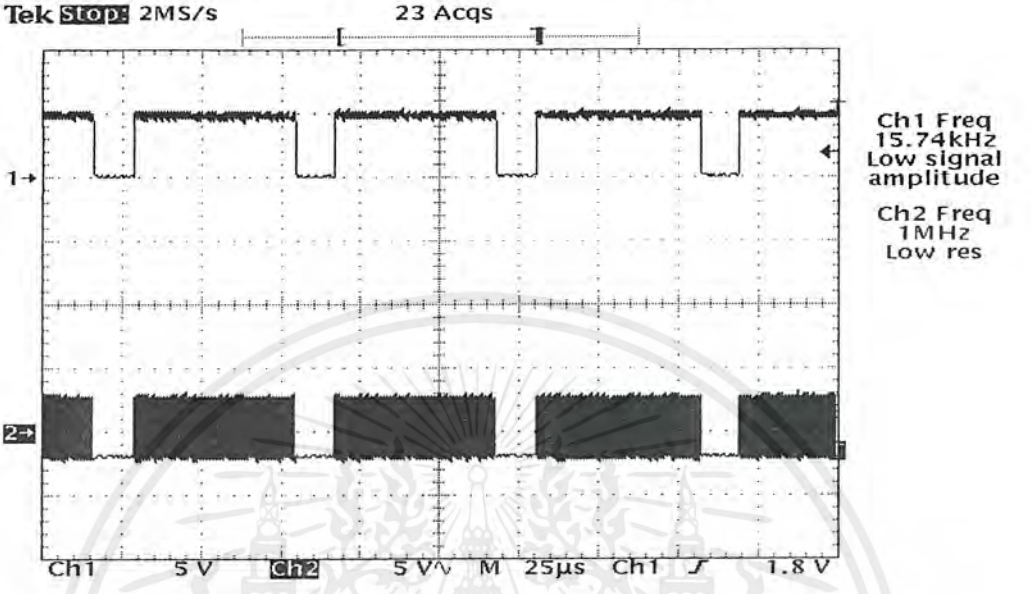
รูปที่ 4.2 ได้แสดงสัญญาณภาพเดิมจากรูปที่ 4.1 โดยสัญญาณภาพนี้จะถูกยกระดับขึ้นไปและถูกขยายให้มีความแรงขึ้น โดยวงจรยกระดับและวงจรขยายก่อนเข้าสู่ขาอินพุทของ IC CA3318 (U1) เพื่อให้สามารถปรับค่าแรงดันอ้างอิงเพื่อใช้ในการเข้ารหัสในการแปลงสัญญาณเป็นไบนารี 8 บิตได้อย่างเหมาะสม ซึ่งผลการวัดได้ค่า แรงดันสูงสุด 1.64 V และค่าแรงดันต่ำสุด 0.660 V ดังนั้นในการปรับแรงดันอ้างอิงด้านบวกและด้านลบที่ ขา 22 และ ขา 25 ตามลำดับจึงควรปรับให้ครอบคลุมสัญญาณนี้ โดยในที่นี้ได้ปรับแรงดันอ้างอิงด้านบวกไว้ที่ 2 V และแรงดันอ้างอิงด้านลบไว้ที่ 0.5 V



รูปที่ 4.3 แสดงสัญญาณนาฬิกาความถี่ 5 MHz

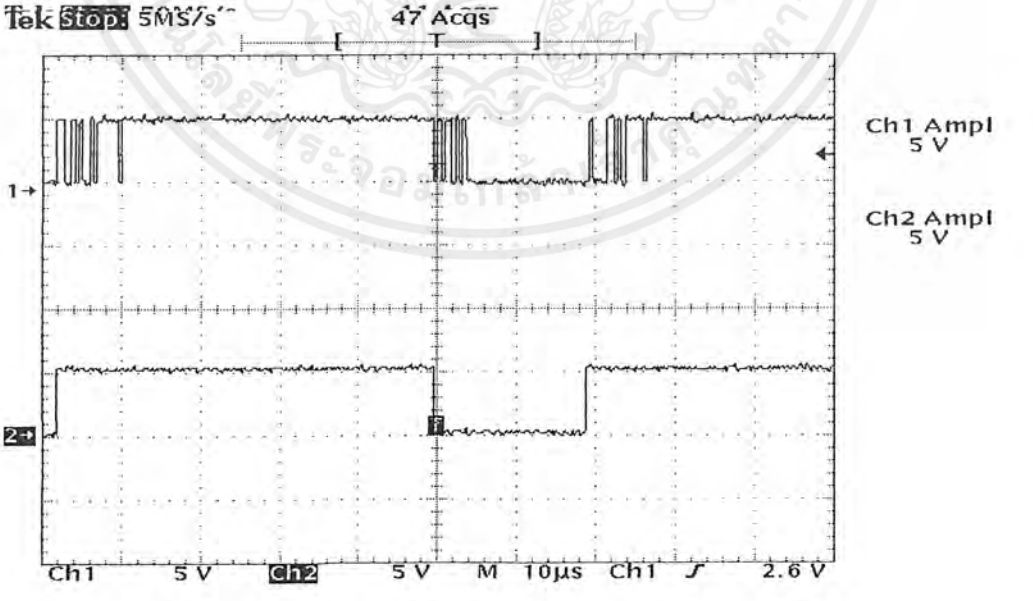
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 ได้แสดงสัญญาณนาฬิกาความถี่ 5 MHz ซึ่งเป็นสัญญาณที่ใช้เป็นความถี่อ้างอิงในการแซมปลิ้งสัญญาณภาพ สาเหตุที่ใช้ความถี่ 5 MHz เนื่องจากโครงงานนี้ต้องการความละเอียดของภาพ 256*256 จุด โดยสัญญาณเส้นภาพแต่ละเส้นมีคาบเวลา 64 uS นำ 64 uS หารด้วย 256 จะได้ 250 nS ซึ่งเป็นความถี่ 4 MHz ซึ่งเป็นค่าความถี่ต่ำสุดที่ควรจะใช้ในการแซมปลิ้งสัญญาณนี้



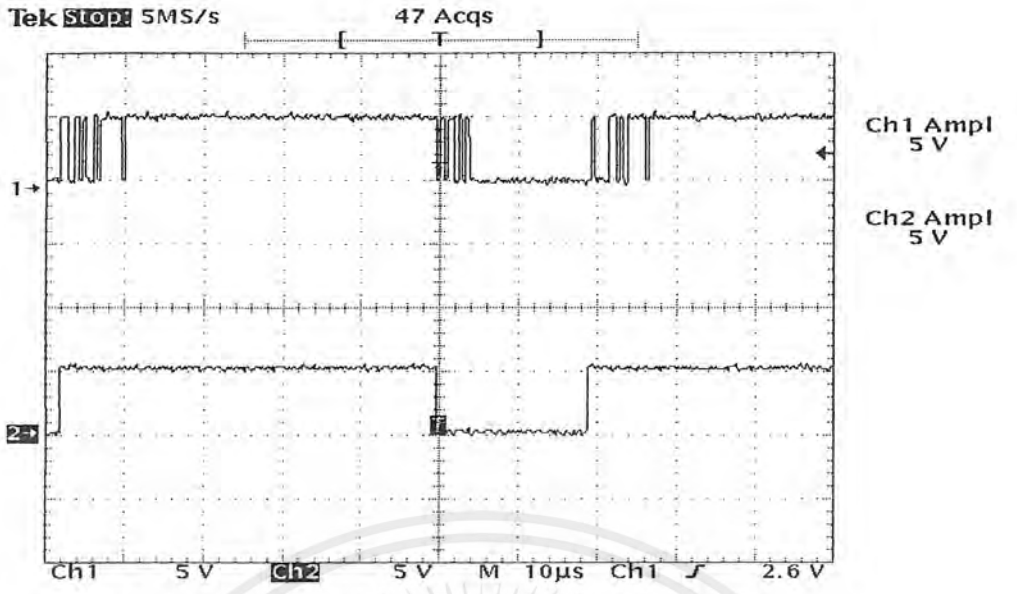
รูปที่ 4.4 แสดงสัญญาณ /CR1 (CH1)ซึ่งใช้เป็นสัญญาณควบคุมการสร้างสัญญาณนาฬิกา 5 MHz(CH2)

รูปที่ 4.4 ได้แสดงสัญญาณ /CR1 ซึ่งเป็นสัญญาณที่ได้มาจากไมโครคอลโทรลเลอร์ เพื่อใช้เป็นสัญญาณรีเซ็ต วงจรสร้างสัญญาณนาฬิกา เพื่อควบคุมการสร้างสัญญาณนาฬิกาขึ้นมาให้ตรงกับการเก็บข้อมูลลงหน่วยความจำและการอ่านข้อมูลจากหน่วยความจำในเวลาที่เหมาะสมถูกต้อง



รูปที่ 4.5 a แสดงสัญญาณดิจิตอลที่ได้จากการ IC CA3318 ที่ขาเอาท์พุท (ขา 1-8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

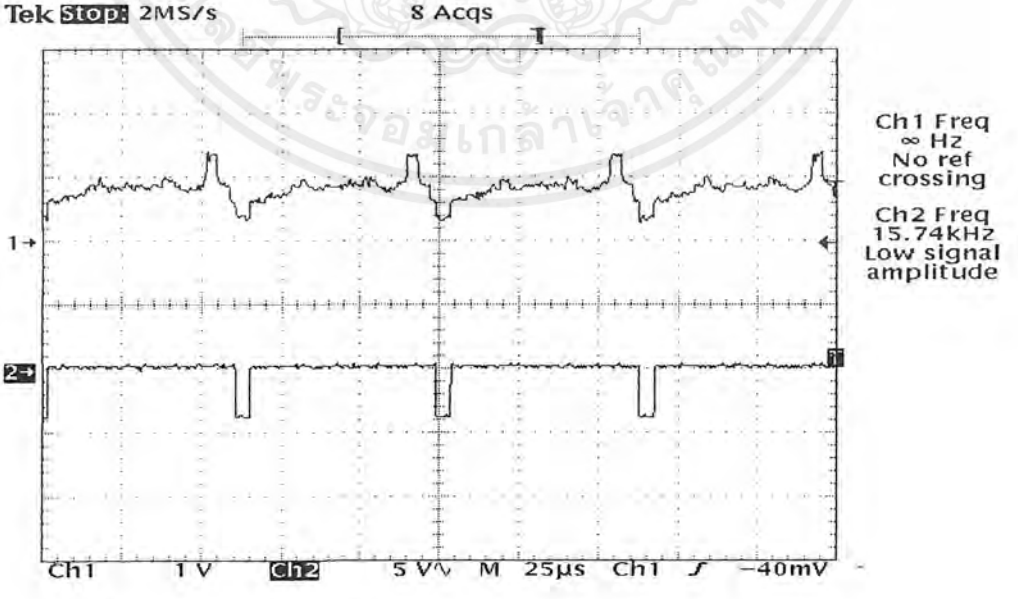


รูปที่ 4.5 b แสดงสัญญาณดิจิตอลที่ได้จากการ IC CA3318 ที่ขาเอาท์พุท (ขา 1-8)

รูปที่ 4.5 a และ 4.5 b ได้แสดงผลการวัดสัญญาณดิจิตอลที่ได้มาจากการแปลงสัญญาณภาพขาว-ดำ ของ IC CA3318 ซึ่งจะเห็นได้ว่าเป็นขบวนของพัลซ์ลอจิก 0 และ 1 ปนกันไป ซึ่งเป็นรูปแบบของสัญญาณดิจิตอลไบนารี่ แต่ยังไม่สามารถบอกได้ถึงความถูกต้องของสัญญาณ โดยในที่นี้ได้ทำการวัดออกมาเป็นตัวอย่างไว้ 4 สัญญาณจาก 4 ขาเอาท์พุท

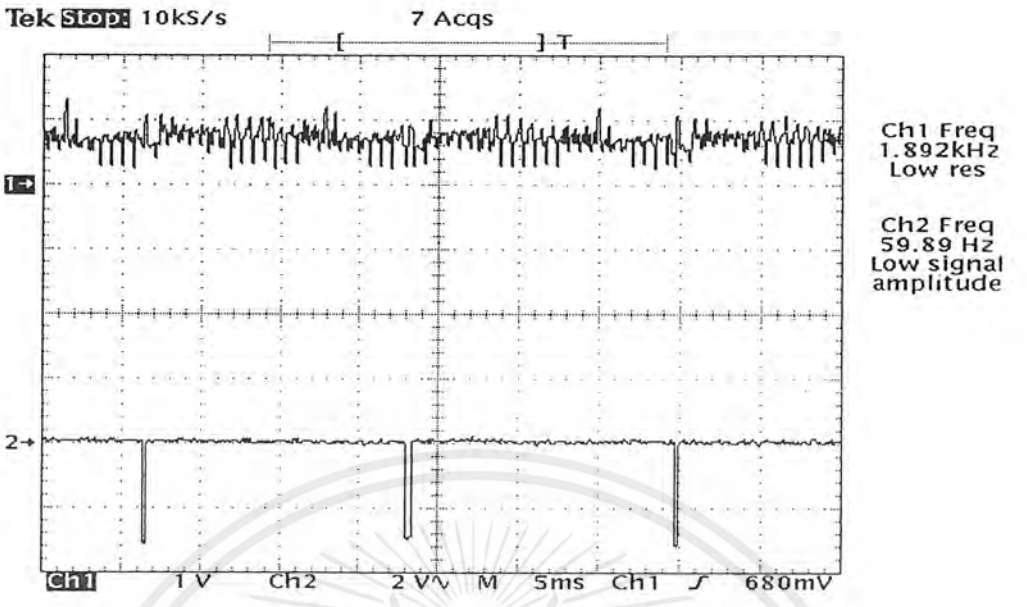
4.2 การทดลองภาคการแยกสัญญาณซิงค์

จากรูปที่ 3.3 ในภาคการแยกสัญญาณซิงค์นี้จะเป็นการนำสัญญาณภาพจากกล้องมาทำการแยกเอาสัญญาณฮอร์ซิงค์และเวอร์ซิงค์เพื่อที่จะนำสัญญาณ 2 สัญญาณนี้ไปสร้างสัญญาณฮอร์เคานท์และเวอร์เคานท์ตามลำดับต่อไป โดยใช้ IC LM 1881 ในการแยกสัญญาณซิงค์ออกจากสัญญาณภาพ จากการทดลองได้ผลดังนี้

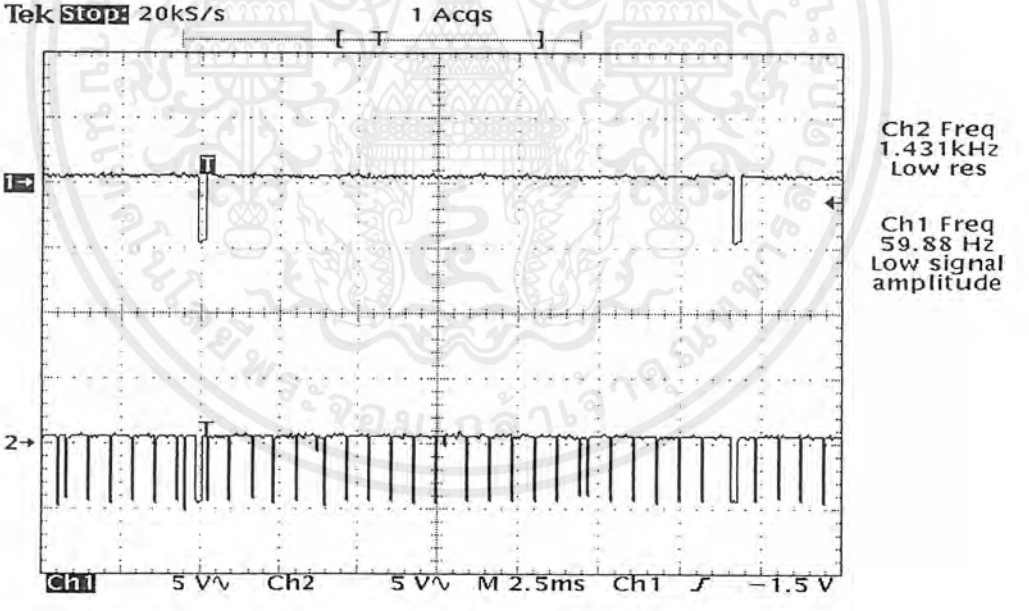


รูปที่ 4.6 แสดงสัญญาณฮอร์ซิงค์(CH2)ซึ่งถูกแยกออกมาจากสัญญาณภาพ(CH1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

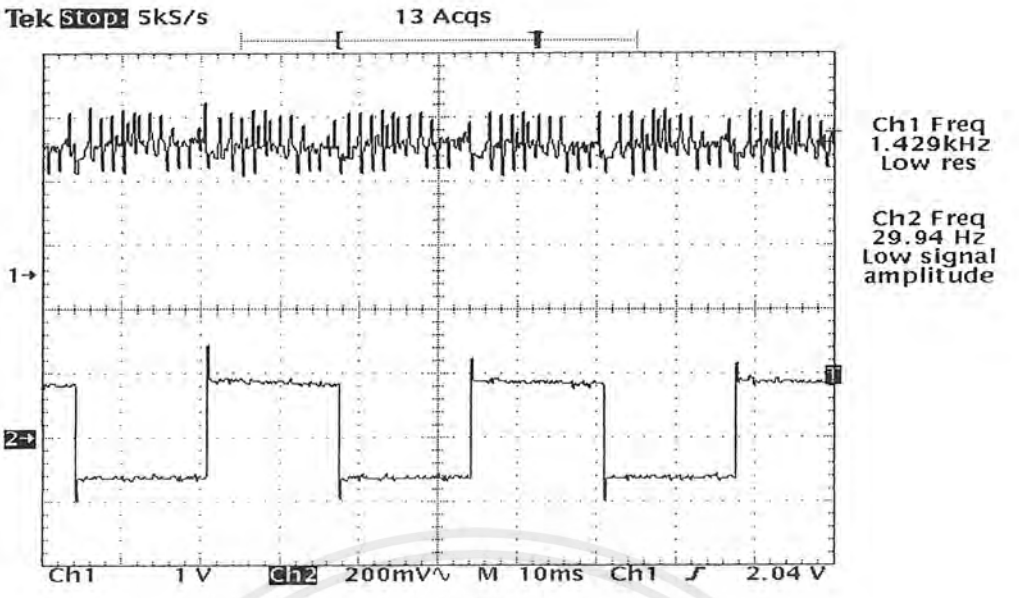


รูปที่ 4.7 แสดงสัญญาณเวอร์ชิงค์(CH2)ซึ่งถูกแยกออกมาจากสัญญาณภาพ(CH1)



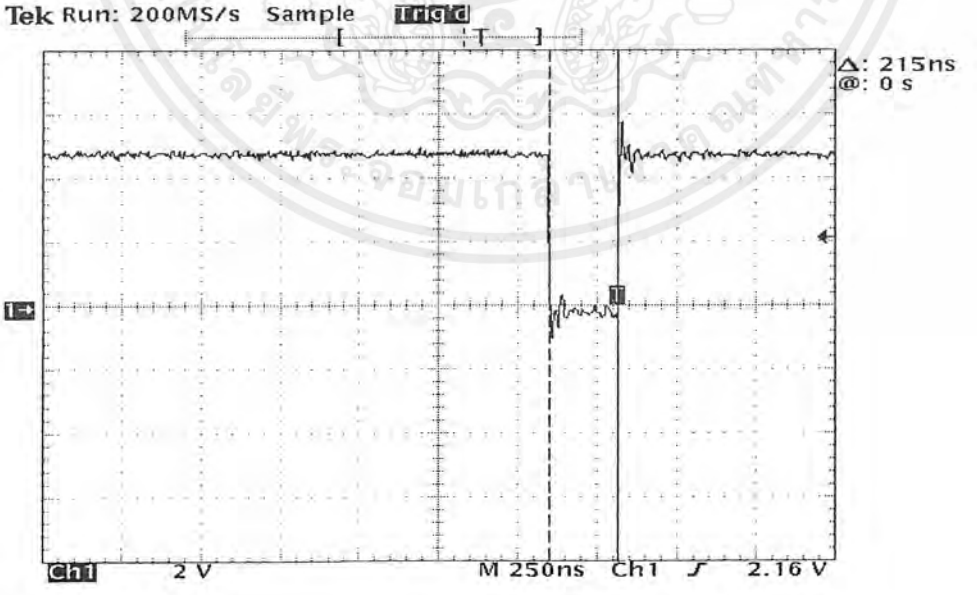
รูปที่ 4.8 แสดงการเปรียบเทียบสัญญาณฮอร์ชิงค์(CH2)และเวอร์ชิงค์(CH1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



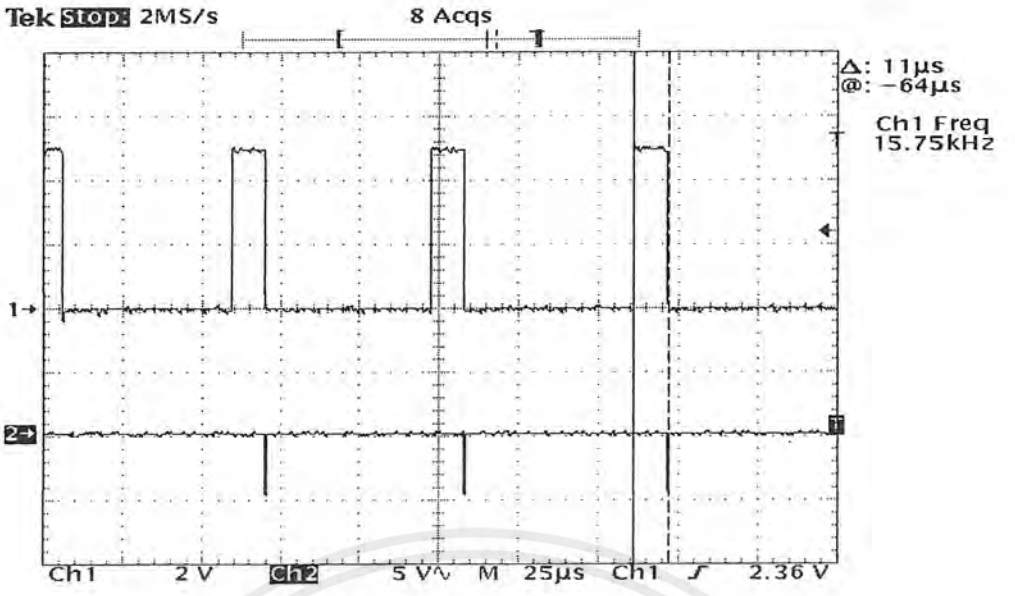
รูปที่ 4.9 แสดงสัญญาณ ODD/EVEN (CH2)ที่แยกออกมาจากสัญญาณภาพ(CH1)

รูปที่ 4.6 , 4.7 , 4.8 และ 4.9 ได้แสดงผลของใช้ IC LM1881 ในการแยกสัญญาณที่เป็นส่วนประกอบของสัญญาณภาพ โดยได้ทำการแยกสัญญาณภาพออกเป็นสัญญาณฮอร์ซิงค์ออกมาทางขา 1 ซึ่งมีความถี่ประมาณ 15625 Hz สัญญาณเวอร์ซิงค์ออกมาทางขา 3 ซึ่งมีค่าความถี่ประมาณ 60 Hz โดยในรูปที่ 4.6 ,4.7 และ 4.8 ได้แสดงตำแหน่งของสัญญาณฮอร์ซิงค์และเวอร์ซิงค์ซึ่งอยู่ในสัญญาณภาพและผลของสัญญาณที่ถูกแยกออกมาที่ตำแหน่งเดียวกันอย่างชัดเจน สำหรับในรูปที่ 4.9 เป็นสัญญาณ ODD/EVEN ซึ่งเป็นสัญญาณแสดงฟิลด์คู่และฟิลด์คี่ของสัญญาณภาพ(สัญญาณภาพ 1 เฟรม หรือ 1 ภาพประกอบด้วย 2 ฟิลด์คือฟิลด์คู่และฟิลด์คี่)

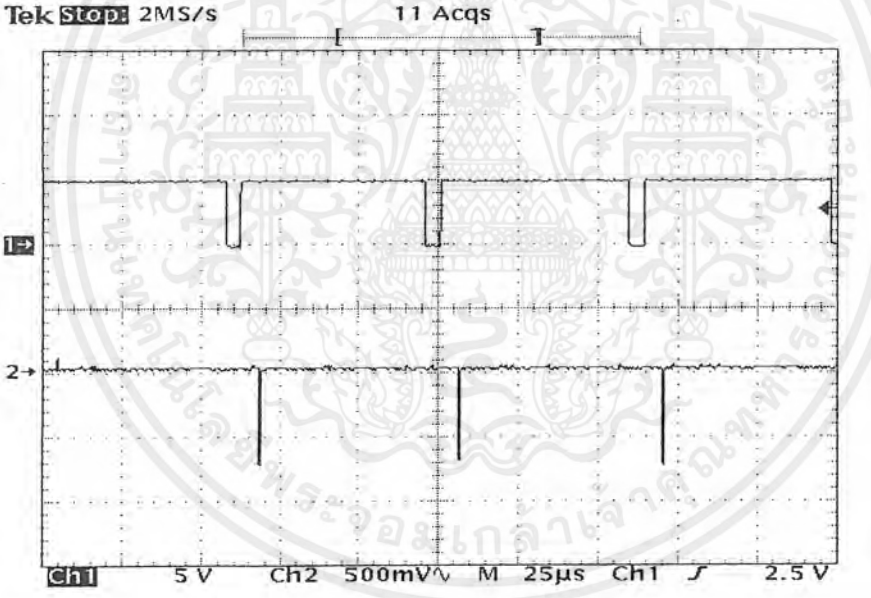


รูปที่ 4.10 แสดงสัญญาณฮอร์เคานท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



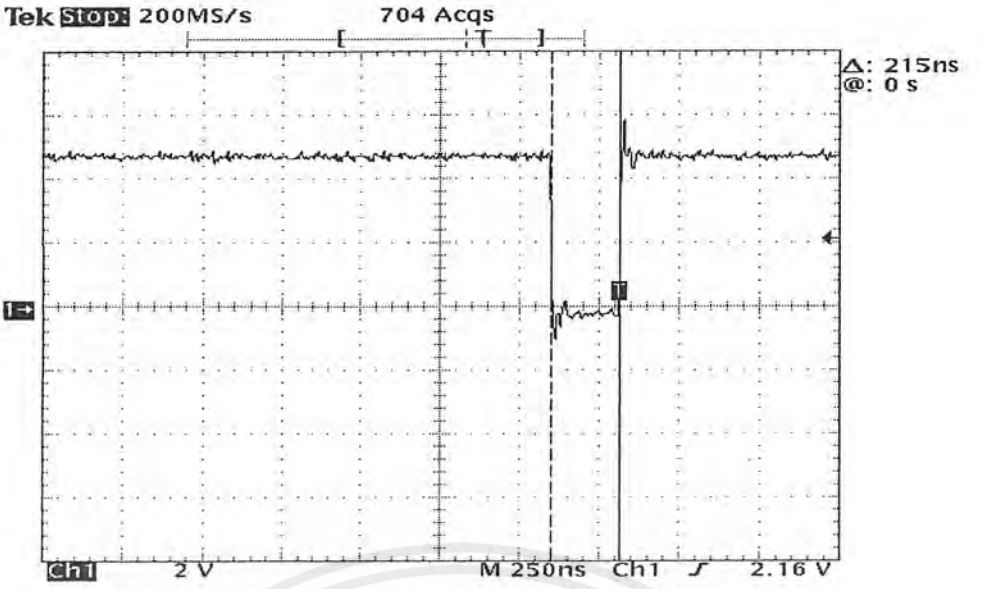
รูปที่ 4.11 แสดงการเปรียบเทียบสัญญาณหน่วงเวลา(CH1)กับสัญญาณฮอร์เคานท์(CH2)



รูปที่ 4.12 แสดงการเปรียบเทียบสัญญาณฮอร์ชิงค์(CH1)กับสัญญาณฮอร์เคานท์(CH2)

รูปที่ 4.10 , 4.11 และ 4.12 ได้แสดงกระบวนการสร้างสัญญาณฮอร์เคานท์ซึ่งเป็นสัญญาณขอบขาลงที่ขนาดเพียงประมาณ 200 nS ดังแสดงในรูปที่ 4.10 ซึ่งใช้เป็นสัญญาณไปรีเซ็ตวงจรนับให้เริ่มทำการนับให้ในภาคต่อไป โดยสัญญาณฮอร์เคานท์ที่สร้างมาจะถูกหน่วงเวลาชั่วขณะหนึ่งประมาณ 10 uS เพื่อให้ให้มีการเริ่มนับและเก็บข้อมูลในส่วนที่เป็นสัญญาณภาพพร้อมกันทุกครั้งไปดังในรูปที่ 4.11 เป็นส่วนของสัญญาณที่หน่วงเวลาการเกิดสัญญาณฮอร์เคานท์ และในรูปที่ 4.12 เป็นการแสดงให้เห็นถึงการล้าหลังของสัญญาณฮอร์เคาน์ต่อสัญญาณฮอร์ชิงค์เป็นเวลาเท่ากับสัญญาณหน่วงเวลานั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

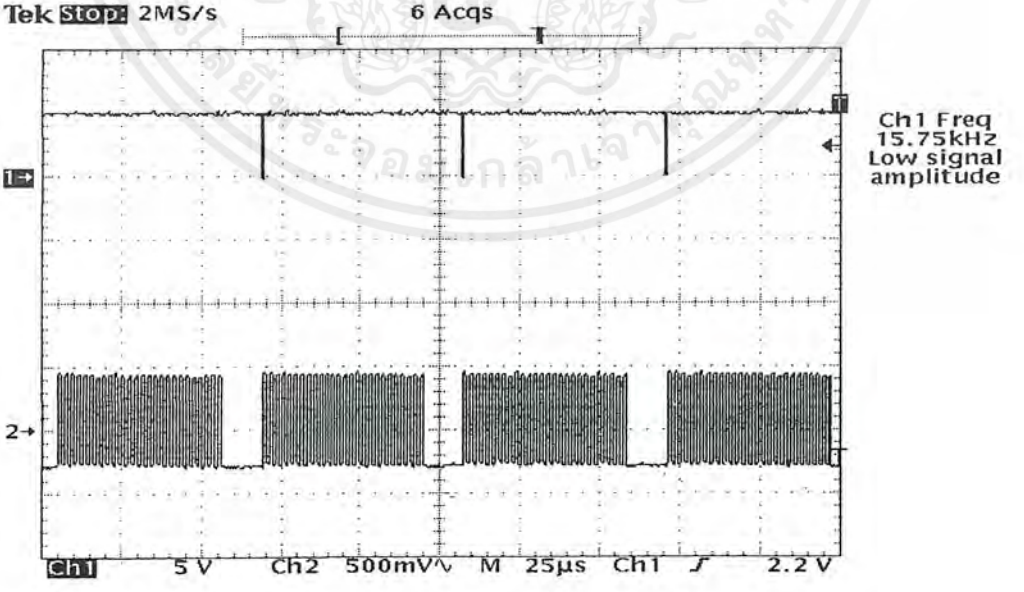


รูปที่ 4.13 แสดงสัญญาณเวอร์เคานท์

รูปที่ 4.13 ได้แสดงขนาดของสัญญาณเวอร์เคานท์ซึ่งจะมีขนาดใกล้เคียงกับสัญญาณฮอร์เคานท์มากเพราะสร้างมากจากไอซีโมโนสเตเบิลที่มีการคำนวณค่าเวลาสัญญาณพัลซ์ที่จะกำเนิดเท่ากัน แต่สัญญาณเวอร์เคานท์นี้จะถูกหน่วงเวลาไว้ด้วยวงจรนับ โดยจะให้สัญญาณฮอร์ซิงค์เข้ามาครบ 32 ลูกแรกก่อนจึงเกิดการสร้างสัญญาณเวอร์เคานท์ขึ้นมา แต่ไม่สามารถที่จะวัดเพื่อมาเปรียบเทียบได้

4.3 การทดลองภาควงจรนับ(Counter)

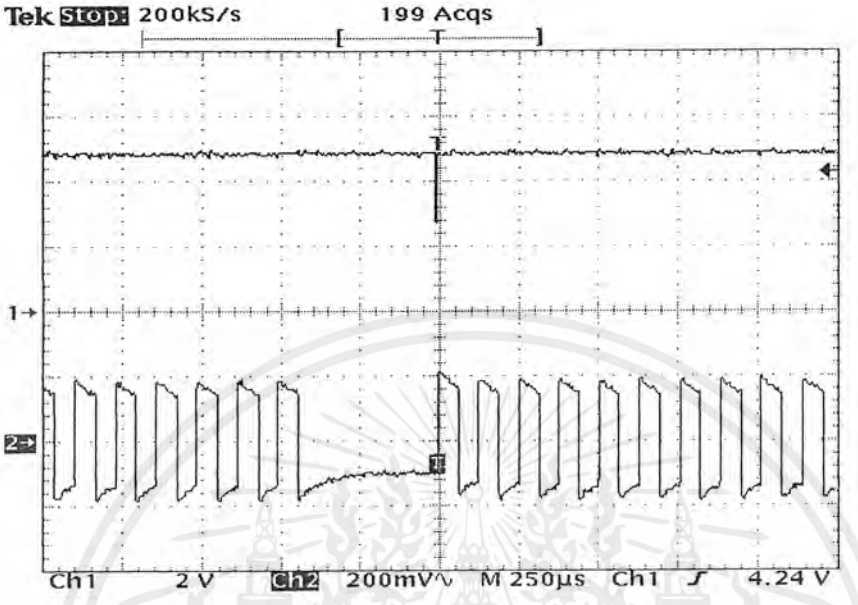
จากรูปที่ 3.4 ในการทดลองภาคนี้เป็นการทดลองการใช้สัญญาณฮอร์เคานท์และเวอร์เคานท์มาควบคุมการสร้างสัญญาณเคาน์เตอร์เพื่อใช้ในการระบุตำแหน่งแอดเดรสในการเขียนและอ่านข้อมูลจากหน่วยความจำ โดยมีผลการทดลองดังนี้



รูปที่ 4.14 แสดงสัญญาณฮอร์เคานท์(CH1)ในการควบคุมการสร้างสัญญาณเคาน์เตอร์ CA0 (CH2)

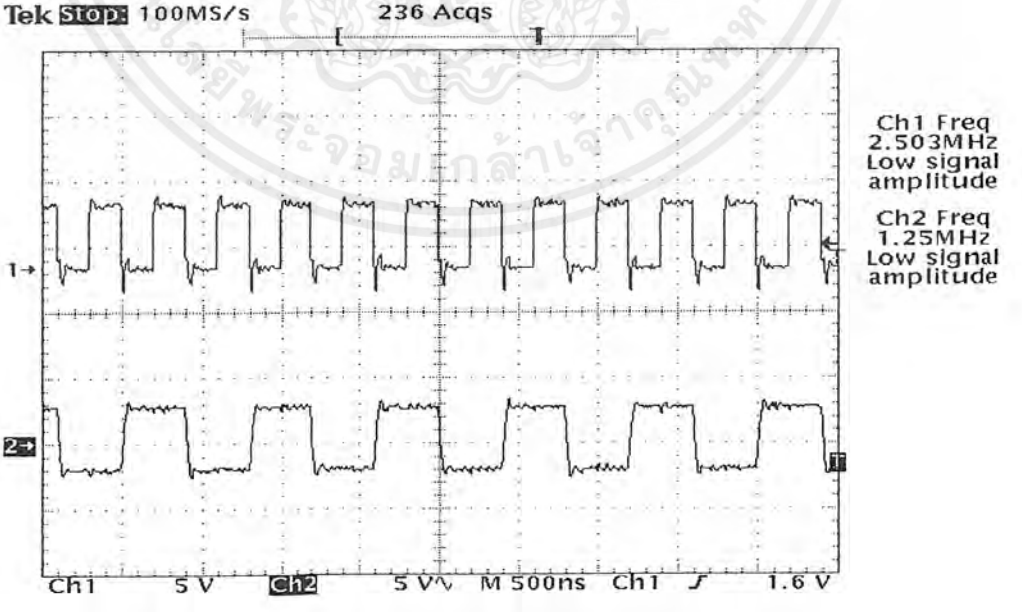
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 ได้แสดงการใช้สัญญาณฮอร์เคานท์ในการควบคุมการสร้างสัญญาณเคาน์เตอร์โดยในที่นี้ใช้ CA0 เป็นตัวเปรียบเทียบเทียบ จะเห็นได้ว่าสัญญาณฮอร์เคานท์จะเป็นตัวรีเซ็ตการการนับโดยจากรูปที่ 3.4 สัญญาณฮอร์เคานท์จะเป็นตัวควบคุมการนับ CA0-CA7



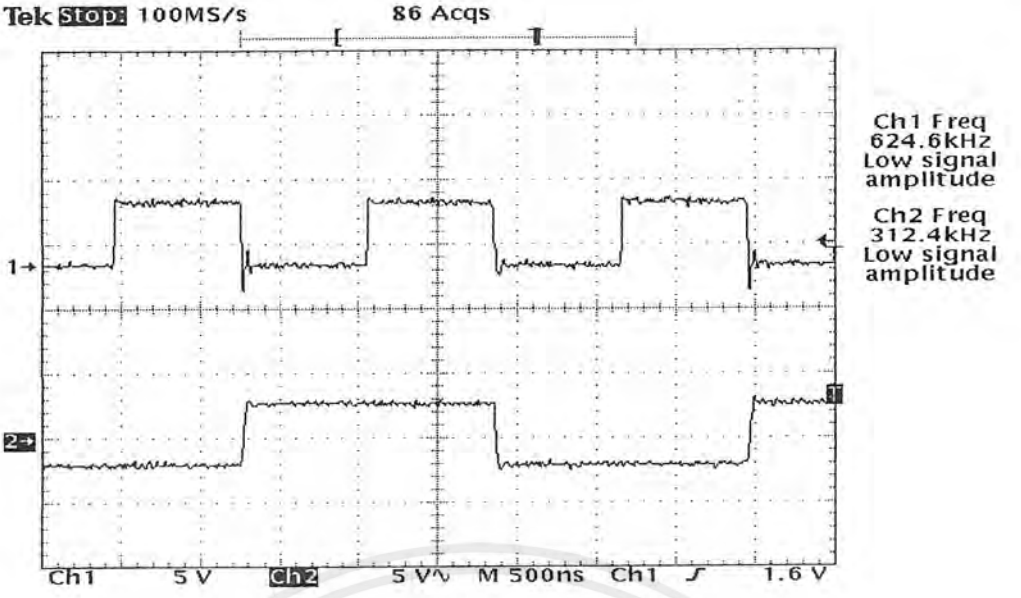
รูปที่ 4.15 แสดงการใช้สัญญาณเวอร์เคานท์(CH1)ในการควบคุมการสร้างสัญญาณเคาน์เตอร์ CA8(CH2)

รูปที่ 4.15 ได้แสดงการใช้สัญญาณเวอร์เคานท์ในการควบคุมการสร้างสัญญาณเคาน์เตอร์โดยในที่นี้ใช้ CA8 เป็นตัวเปรียบเทียบเทียบ จะเห็นได้ว่าสัญญาณฮอร์เคานท์จะเป็นตัวรีเซ็ตการการนับโดยจากรูปที่ 3.4 สัญญาณฮอร์เคานท์จะเป็นตัวควบคุมการนับ CA8-CA15



รูปที่ 4.16 a แสดงสัญญาณเคาน์เตอร์ CA0 (CH1) และ CA1 (CH2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

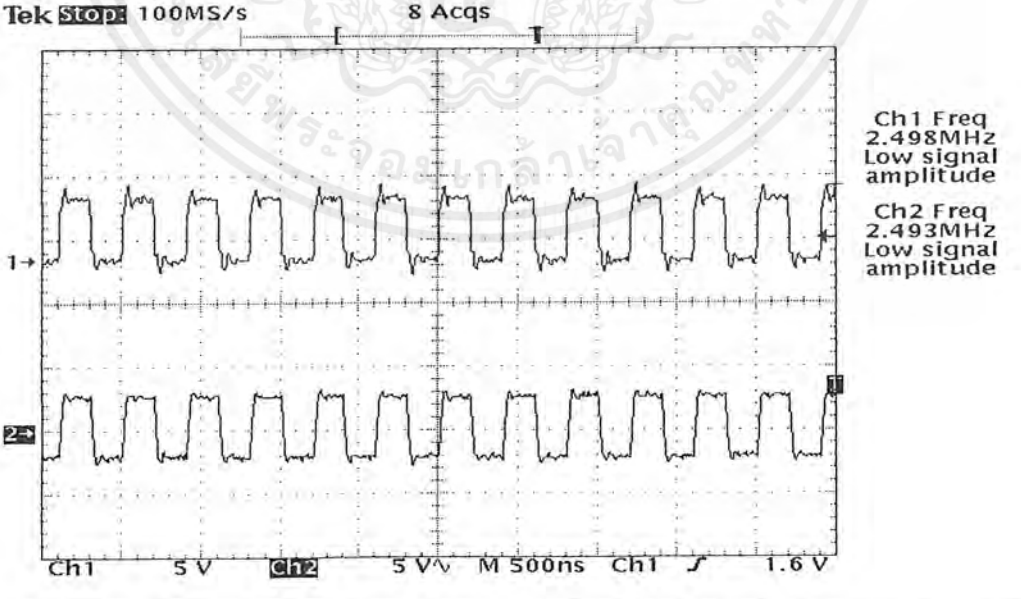


รูปที่ 4.16 b แสดงสัญญาณเคาน์เตอร์ CA2 (CH1) และ CA3 (CH2)

รูปที่ 4.16 a และ 4.16 b ได้แสดงสัญญาณของวงจรเคาน์เตอร์โดยในที่นี้ได้วัดมาเป็นตัวอย่าง 4 สัญญาณ จากรูปที่ 3.4 คือสัญญาณ CA0 CA1 CA2 และ CA3 ตามลำดับ โดยจะสังเกตุเห็นได้ว่าสัญญาณนั้นจะมีความถี่ลดลงเป็น 2 เท่า ลงไปเรื่อยตามหลักการนับตำแหน่งของเลขไบนารี่

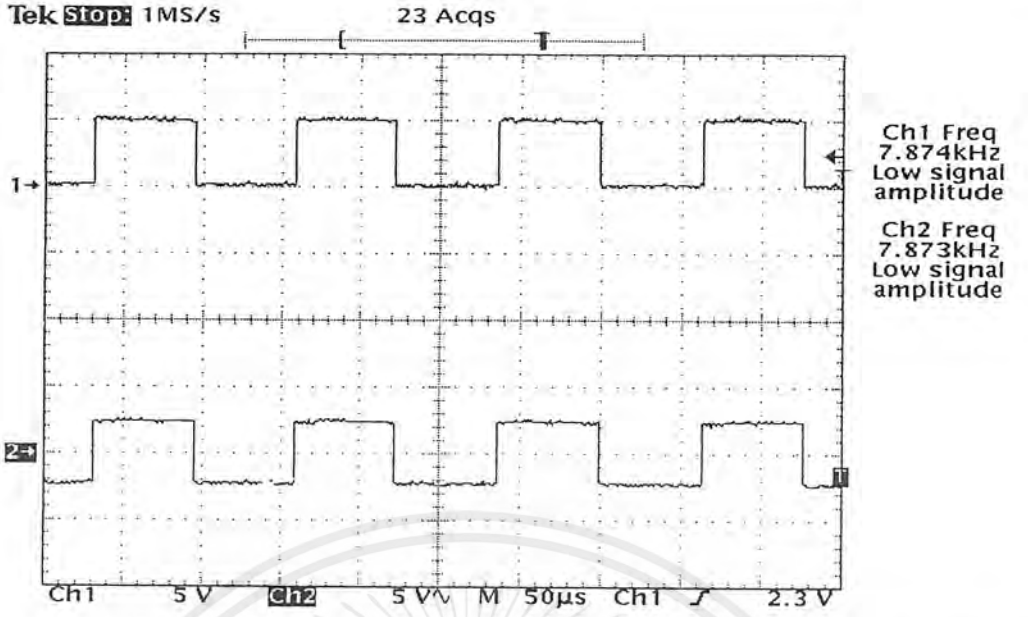
4.4 การทดลองภาคแอดเดรส

จากวงจรรูปที่ 3.5 ซึ่งเป็นวงจรภาคแอดเดรสโดยจะมีลักษณะการทำงานเป็น 2 ช่วงคือช่วงของการระบุแอดเดรสเพื่อทำการเขียนข้อมูลลงหน่วยความจำซึ่งจะได้สัญญาณเคาน์เตอร์ CA0-CA15 มาเป็นตัวอ้างแอดเดรส และช่วงการระบุแอดเดรสเพื่อทำการอ่านข้อมูลจากหน่วยความจำซึ่งจะได้สัญญาณอ้างแอดเดรส PA0-PA15 จากไมโครคอลโทรลเลอร์ โดยทั้ง 2 ช่วงจะทำงานสลับกันไปโดยมีสัญญาณ CD2 เป็นตัวเลือกการทำงานในแต่ละช่วง โดยผลการวัดและทดลองมีดังนี้



รูปที่ 4.17 แสดงสัญญาณเคาน์เตอร์ที่ผ่านเข้าและออกบัฟเฟอร์ 74LS244 (U12) CA0 (CH1) และ A0(CH2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

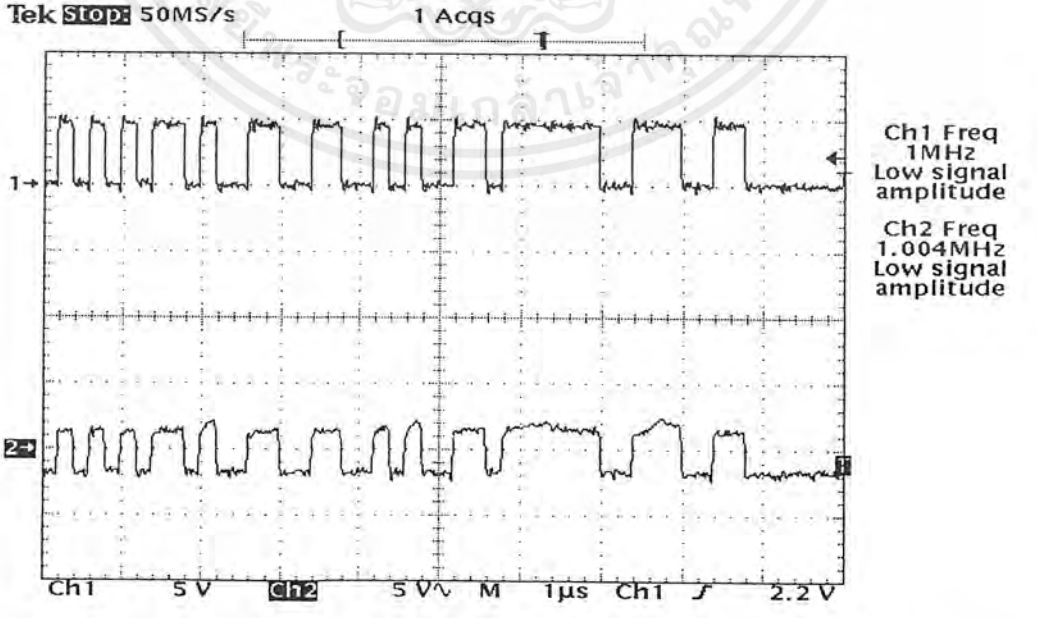


รูปที่ 4.18 แสดงสัญญาณเคาน์เตอร์ที่ผ่านเข้าและออกบัฟเฟอร์ 74LS244 (U13) CA8 (CH1) และ A8(CH2)

รูปที่ 4.17 และ รูปที่ 4.18 ได้แสดงความถูกต้องของสัญญาณเคาน์เตอร์ที่ผ่านเข้าไปยังบัฟเฟอร์ 74LS244 U12 และ U13 โดยวัดมาเป็นตัวอย่าง 2 สัญญาณ คือ CA0 กับ A0 และ CA8 กับ A8 จากรูปสัญญาณทั้งสองรูปจะเห็นได้ว่าสัญญาณเข้าและออกมาอย่างถูกต้อง

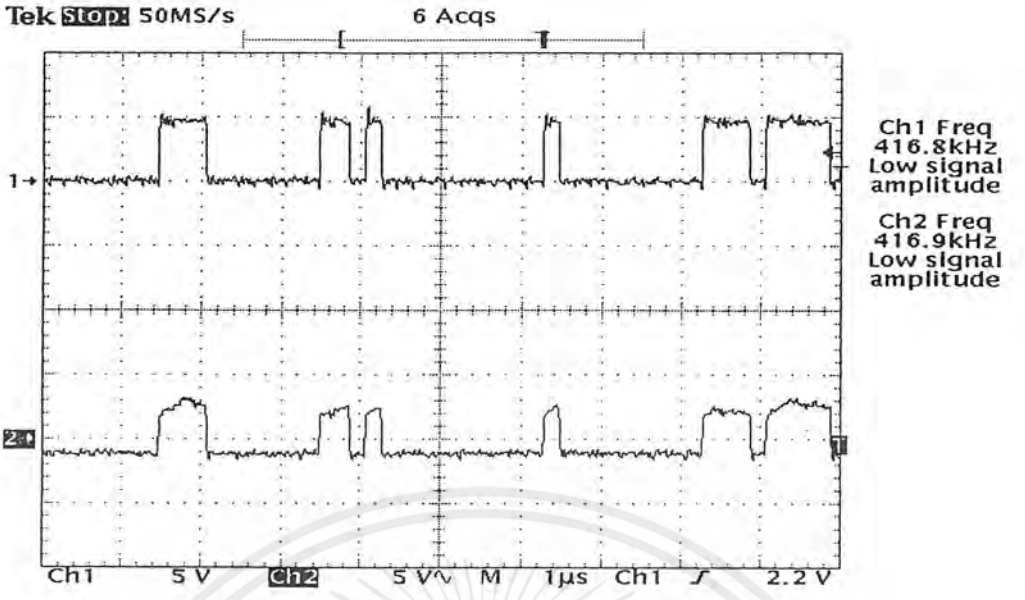
4.5 การทดลองภาคหน่วยความจำ

จากรูปที่ 3.6 เป็นวงจรที่ใช้ในการทดลองภาคหน่วยความจำ ซึ่งใช้ในการเก็บข้อมูลสัญญาณภาพ โดยแบ่งการทำงานเป็น 2 ช่วง คือ ช่วงที่มีการเขียนข้อมูลลงหน่วยความจำ มีสัญญาณ WE เป็นสัญญาณควบคุมการเขียนและช่วงที่มีการอ่านข้อมูลจากหน่วยความจำ มีสัญญาณ RD เป็นสัญญาณควบคุมการอ่าน ผลการวัดจะทำการวัดความถูกต้องของสัญญาณในขณะที่ทำการเขียนลงบนหน่วยความจำผ่านทางบัฟเฟอร์ 74LS245 ซึ่งเป็นบัฟเฟอร์ 2 ทาง มีผลดังนี้



รูปที่ 4.19 แสดงสัญญาณข้อมูลดิจิตอลที่ได้จากการแซมปลิ้ง ID0 (U16) และข้อมูลออกบัฟเฟอร์ D0 (U16)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



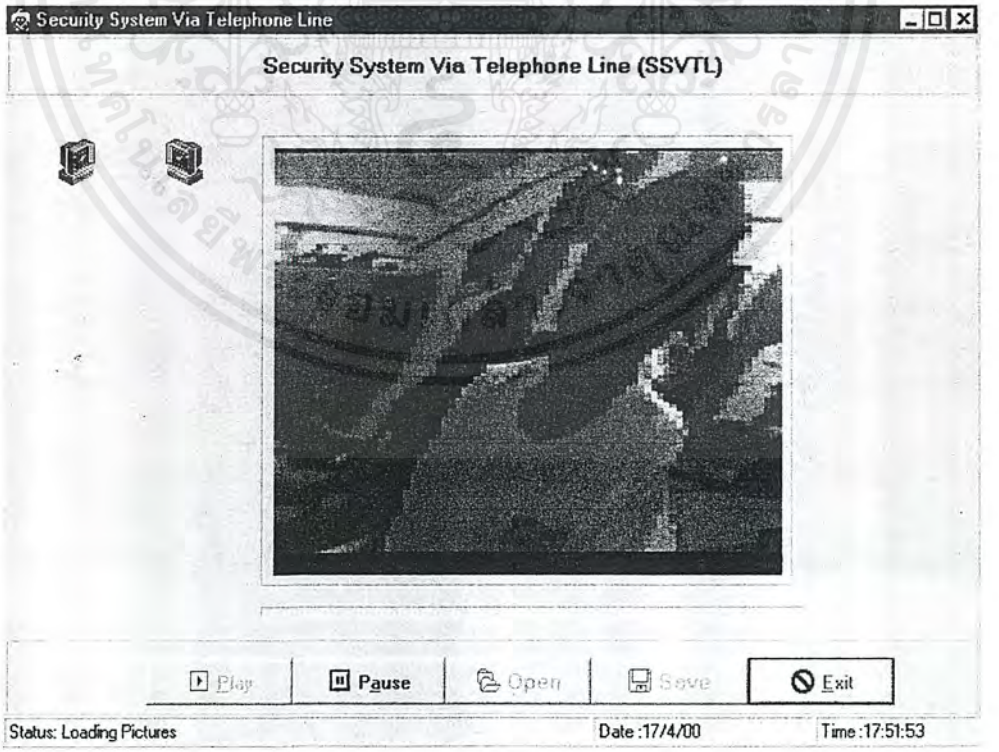
รูปที่ 4.20 แสดงสัญญาณข้อมูลดิจิตอลที่ได้จากการแซมปลิ้ง ID1 (U16) และข้อมูลออกบัฟเฟอร์ D1 (U16)

รูปที่ 4.19 และ 4.20 ได้แสดงถึงการวัดสัญญาณภาพดิจิตอลที่ได้จากการแซมปลิ้งออกมาจาก IC CA3318 และผ่านเข้าไปยังบัฟเฟอร์ เปรียบเทียบกับสัญญาณที่ออกมาจากบัฟเฟอร์ โดยในที่นี้วัดสัญญาณมาเป็นตัวอย่าง 2 สัญญาณ คือ ID0 กับ D0 และ ID1 กับ D1 จะเห็นได้ว่าสัญญาณที่เข้าและออกมีความถูกต้องเหมือนกัน แม้ว่าจะมีสัญญาณรบกวนอยู่บ้างก็ตาม แสดงว่าข้อมูลที่จะเขียนลงหน่วยความจำก็จะมีความถูกต้องเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

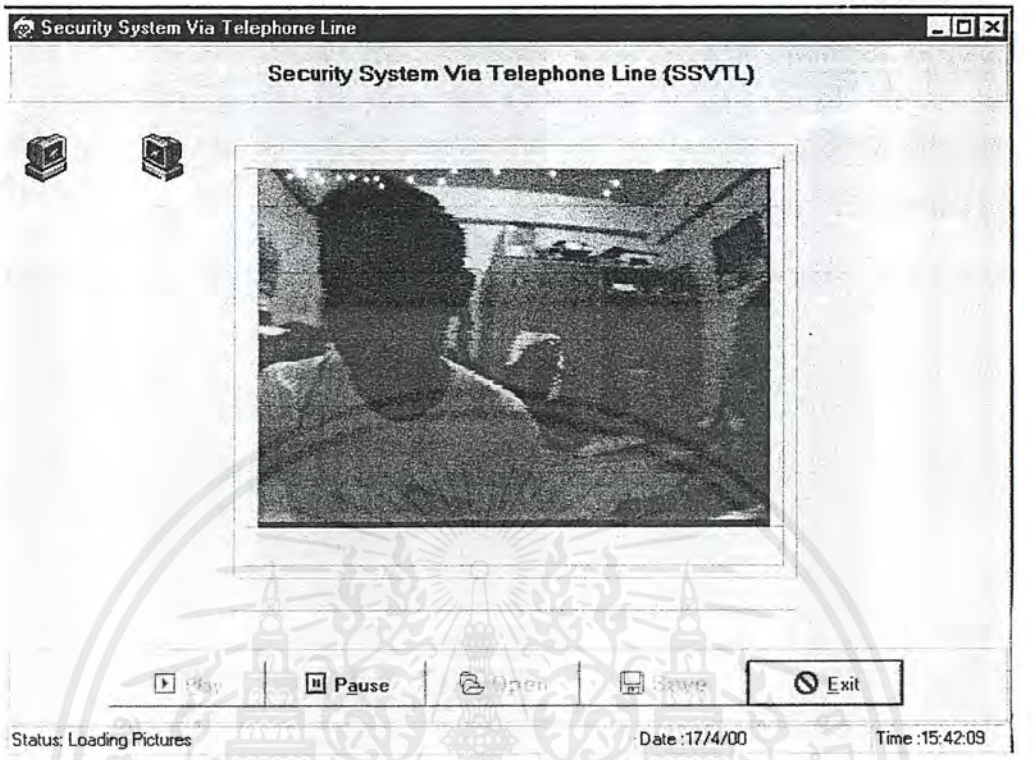
4.6 การทดสอบโดยการเขียนและอ่านภาพจากหน่วยความจำและแสดงบนหน้าจอคอมพิวเตอร์ตัวเก็บข้อมูลภาพ

รูปที่ 4.21 แสดงหน้าต่างแรกเมื่อเริ่มรันโปรแกรมที่เครื่องเก็บไฟล์ภาพ



รูปที่4.22 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องเก็บไฟล์ภาพ(ไม่มีการแก้ไขข้อมูลภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

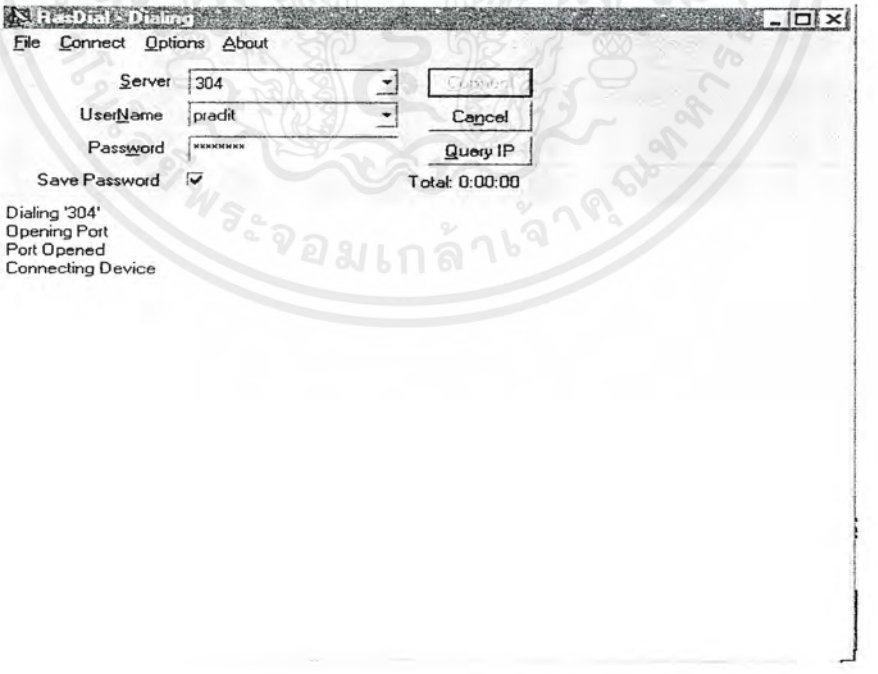


รูปที่4.23 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องเก็บไฟล์ภาพ(มีการแก้ไขข้อมูลภาพด้วย Software)

เมื่อเริ่มรันโปรแกรมที่เครื่องเก็บไฟล์ภาพจะพบกับหน้าต่างดังรูปที่ 4.21 หลังจากนั้นให้คลิกปุ่ม Play เพื่อให้โปรแกรมเริ่มทำการเก็บข้อมูลภาพแล้วแปลงเป็นไฟล์ BMP แล้วเก็บไว้ภายใน Hard Disk จำนวน 40 ไฟล์ภาพ เมื่อเขียนครบ 40ไฟล์ภาพแล้วจะวนกลับมาเขียนทับไฟล์แรกใหม่ หลังจากการทดลองรันโปรแกรมในครั้งแรกจะพบว่า ภาพที่ได้จะเกิดการเอียงและมีการเริ่มต้นภาพผิด ดังรูปที่ 4.22 ดังนั้นจึงได้ใช้ SoftWare หาจุดเริ่มต้นภาพแล้วทำการเรียงข้อมูลภาพใหม่ ทำให้ได้ภาพที่ถูกต้องออกมาดังรูปที่ 4.23 แต่เมื่อมีการ Connect จากเครื่องอ่านไฟล์ภาพเข้ามาเพื่อทำการอ่านไฟล์ภาพบนเครื่องเก็บไฟล์ภาพจะทำให้การเรียงข้อมูลภาพบนเครื่องเก็บไฟล์ภาพเกิดการผิดพลาดเนื่องจากการ Interrupt CPU ของ Serial Port จากการขอติดต่อของเครื่องอ่านไฟล์ภาพ ซึ่งจะต้องใช้ Network Terminal โดยจะทำให้ CPU ทำงานได้ราบรื่นขึ้นโดยไม่ถูก Interupt รบกวนบ่อยๆ เมื่อต้องการออกจากโปรแกรมก็คลิกปุ่ม Exit เพื่อออกจากโปรแกรม

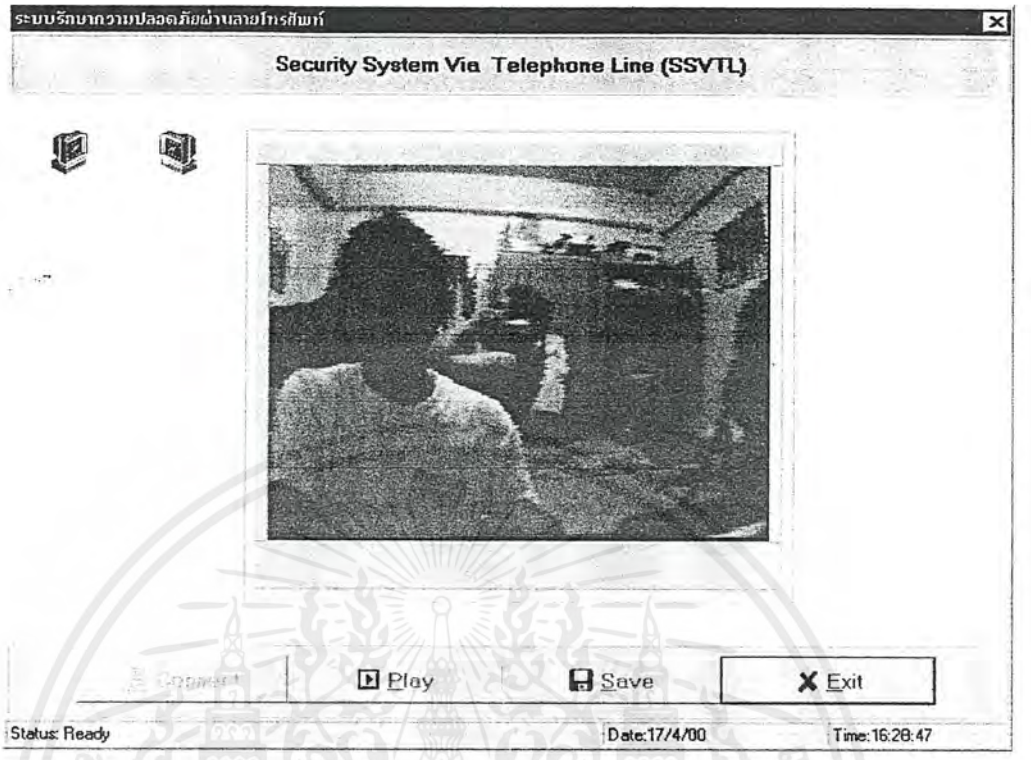
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่4.24 แสดงหน้าต่างแรกเมื่อเริ่มรันโปรแกรมที่เครื่องอ่านไฟล์ภาพ



รูปที่4.25 แสดงหน้าต่างหลังจากคลิกปุ่ม Connect เพื่อการติดต่อระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.26 แสดงหน้าต่างของโปรแกรมแสดงภาพที่เครื่องอ่านไฟล์ภาพ(เมื่อเครื่องเก็บไฟล์ภาพหยุดเก็บข้อมูลภาพ)

เมื่อต้องการไปอ่านข้อมูลภาพยังเครื่องเก็บไฟล์ภาพ โดยเริ่มรันโปรแกรมที่เครื่องอ่านข้อมูลภาพ จะพบกับหน้าต่างรูปที่ 4.24 หลังจากนั้นให้คลิกปุ่ม Connect จะพบกับหน้าต่างดังรูปที่ 4.25 แล้วกรอก User และ Password ที่สามารถติดต่อไปยังเครื่องเก็บไฟล์ภาพ จากนั้นก็คลิกปุ่ม Connect เพื่อทำการติดต่อ เมื่อติดต่อเสร็จ ก็คลิกปุ่ม Play เพื่อทำการอ่านข้อมูลภาพที่เก็บไว้ที่เครื่องเก็บไฟล์ภาพจะพบกับหน้าต่างดังรูปที่ 4.26 ซึ่งเป็นไฟล์ภาพที่เก็บไว้ก่อนที่เครื่องอ่านข้อมูลจะทำการติดต่อซึ่งเป็นข้อมูลภาพที่ถูกต้อง เมื่อต้องการออกจากโปรแกรมก็คลิกปุ่ม Exit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

จากการออกแบบ และทำการทดลองโครงงานที่ผ่านมาสามารถสรุปผลการทำงานส่วนต่าง ๆ ได้ดังนี้

ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล วงจรสามารถสร้างสัญญาณดิจิตอลได้ที่ความถี่ 5 MHz แม้ว่าผลของสัญญาณดิจิตอลที่ได้อาจมีสัญญาณรบกวนอยู่บ้าง

ภาคแยกสัญญาณซิงค์ สามารถแยกสัญญาณฮอร์ซิงค์และเวอร์ซิงค์ ออกมาจากสัญญาณคอมโพสิตวีดีโอได้ และนำไปสร้างเป็นสัญญาณฮอร์เคานท์และเวอร์เคานท์ต่อไปได้

ภาควงจรนับ วงจรสามารถสร้างสัญญาณนับได้สอดคล้องกับสัญญาณนาฬิกา และสัญญาณฮอร์เคานท์และเวอร์เคานท์ที่เป็นตัวควบคุมได้

ภาคแอดเดรส และภาคหน่วยความจำสามารถควบคุมสัญญาณแอดเดรสที่มาจากวงจรนับได้ ที่ใช้ในการเขียนข้อมูลลงบนหน่วยความจำ และสัญญาณแอดเดรสที่มาจากไมโครคอลโทรลเลอร์ซึ่งใช้ในการอ่านข้อมูลจากหน่วยความจำ ในการทำงานสามารถทำงานได้สอดคล้องถูกต้องกับสัญญาณควบคุมต่าง ๆ ได้ดี

ภาคอินเตอร์เฟส สามารถทำการติดต่อกับไมโครคอมพิวเตอร์โดยการทำงานทั้งหมดคอมพิวเตอร์จะสั่งงานให้ไมโครคอลโทรลเลอร์ทำการควบคุมการเขียนและอ่านข้อมูลในหน่วยความจำ ตลอดจนส่งข้อมูลภาพไปยังคอมพิวเตอร์ได้

จากทำงานโดยรวมสามารถนำข้อมูลภาพมาจะเรียงเป็นภาพบนคอมพิวเตอร์ได้ และมีการเคลื่อนไหวเรียงกันเป็นเฟรมเวลาแต่ละเฟรมห่างกันประมาณ 5 วินาที แต่ปัญหาที่พบคือ

- ภาพที่ออกมามีการล้มเอียงไป สามารถทำการแก้โดยใช้ซอฟแวร์จัดเรียงข้อมูลภาพใหม่
- สัญญาณรบกวน ทำการแก้ไขโดยแยกกราวด์อนาลอกและกราวด์ดิจิตอลออกจากกัน
- เมื่อมีการติดต่อมายังคอมพิวเตอร์เครื่องที่กำลังจัดเรียงข้อมูล จะทำให้ภาพที่ออกมาเกิดความผิดเพี้ยนมากขึ้น เนื่องจากว่าเกิดการอินเตอร์รัปต์การทำงานของซีพียูจากพอร์ต SERIAL ซึ่งถูกเป็นระบบ Remote Access Service ปัญหานี้ไม่สามารถแก้ไขได้ด้วยซอฟแวร์หรืออุปกรณ์ที่มีอยู่โดยจากการศึกษาพบว่าต้องใช้อุปกรณ์ NETWORK TERMINAL เพื่อลดภาระการทำงานของซีพียู

สำหรับแนวทางการแก้ปัญหาและพัฒนาต่อ คือการเพิ่มความละเอียดของภาพให้มากกว่าเดิมจะทำให้ได้ภาพที่คมชัดขึ้นและเพิ่มความเร็วในการส่งให้มากขึ้นเพื่อให้ภาพมีความต่อเนื่องกว่านี้ ซึ่งต้องอาศัยการพัฒนาเทคโนโลยีการส่งข้อมูลปัจจุบันให้รวดเร็วกว่านี้ด้วย นอกจากนี้ยังต้องแก้ปัญหาในเรื่องการอินเตอร์รัปต์ซีพียูจากที่กล่าวมาแล้วด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมของเครื่องคอมพิวเตอร์ที่ติดต่อไปยังเครื่องคอมพิวเตอร์ที่เก็บไฟล์ภาพ

```
unit recrive;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  ExtCtrls, ComCtrls, StdCtrls, Buttons, execute, ExtDlgs, jpeg;

type
  TForm2 = class(TForm)
    Panel1: TPanel;
    StatusBar1: TStatusBar;
    Panel2: TPanel;
    B_play: TBitBtn;
    B_connect: TBitBtn;
    B_save: TBitBtn;
    B_help: TBitBtn;
    B_mapnetwork: TBitBtn;
    Timer1: TTimer;
    ProgressBar1: TProgressBar;
    Bevel1: TBevel;
    Image1: TImage;
    Label1: TLabel;
    Label3: TLabel;
    Image2: TImage;
    SavePictureDialog1: TSavePictureDialog;
    B_Exit: TBitBtn;
    Execute1: TExecute;
    procedure B_exitClick(Sender: TObject);
    procedure B_playClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure B_connectClick(Sender: TObject);
    procedure B_saveClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation
 uses about;

{$R *.DFM}


procedure TForm2.B_exitClick(Sender: TObject);
begin
      Form1.close;
      Form2.close;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm2.B_playClick(Sender: TObject);
var
    count:integer;
begin
    B_exit.SetFocus;
    count:=1;
    Progressbar1.Enabled:=true;
    progressbar1.Visible:=true;
    B_play.Enabled:=false;
    repeat
    case count of
    1:Image2.Picture.LoadFromFile('F:\data\s1.bmp');
    2:Image2.Picture.LoadFromFile('F:\data\s2.bmp');
    3:Image2.Picture.LoadFromFile('F:\data\s3.bmp');
    4:image2.picture.LoadFromFile('F:\data\s4.bmp');
    5:image2.picture.LoadFromFile('F:\data\s5.bmp');
    6:image2.picture.LoadFromFile('F:\data\s6.bmp');
    7:image2.picture.LoadFromFile('F:\data\s7.bmp');
    8:image2.picture.LoadFromFile('F:\data\s8.bmp');
    9:image2.picture.LoadFromFile('F:\data\s9.bmp');
    10:image2.picture.LoadFromFile('F:\data\s10.bmp');
    end;
    Image2.Repaint;
    ProgressBar1.StepIt;
 // sleep(100) ;
    count:=count+1; ;
    until count=11;
    B_play.Enabled:=true;
    B_save.Enabled:=true;
    ProgressBar1.Visible:=false;
    ProgressBar1.Enabled:=false;

end;


procedure TForm2.Timer1Timer(Sender: TObject);
begin
     statusbar1.Panels[0].Text:='Status: Ready';
     statusbar1.Panels[1].Text:='Date:'+ datetostr(Date);
     statusbar1.Panels[2].Text:='Time:'+ Timetostr(Time);
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
        ProgressBar1.Enabled:=false;
        ProgressBar1.Visible:=false;
        B_mapnetwork.Enabled:=false;
        B_play.Enabled:=false;
        B_save.Enabled:=false;
end;

procedure TForm2.B_connectClick(Sender: TObject);
begin
         Execute1.Command:='Rasdial.exe';
         Execute1.Execute;
         B_connect.Enabled:=false;
         B_mapnetwork.Enabled:=true;
         B_play.Enabled:=true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure TForm2.B_saveClick(Sender: TObject);
var name:string;
begin
     if SavePictureDialog1.Execute then
     begin
     name:=SavePictureDialog1.FileName;
     image2.Picture.SaveToFile(name);
     end;
end;

end.
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่เครื่องเก็บไฟล์ภาพและมีการแก้ไขภาพแล้ว

```
unit Sent;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, ExtCtrls, ComCtrls, Buttons,Menus, ToolWin, ExtDlgs,
HW_32,
  GIFImage, TimerEx;
type
  TForm2 = class(TForm)
    Image1: TImage;
    ProgressBar1: TProgressBar;
    B_play: TBitBtn;
    B_exit: TBitBtn;
    StatusBar1: TStatusBar;
    B_pause: TBitBtn;
    VicHw32: TVicHw32;
    SavePictureDialog1: TSavePictureDialog;
    OpenPictureDialog1: TOpenPictureDialog;
    Timer1: TTimer;
    B_open: TBitBtn;
    B_save: TBitBtn;
    Panel1: TPanel;
    Panel2: TPanel;
    GIFImage1: TGIFImage;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Panel6: TPanel;
    procedure FormDestroy(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure B_playClick(Sender: TObject);
    procedure B_exitClick(Sender: TObject);
    procedure SaveClick(Sender: TObject);
    procedure OpenClick(Sender: TObject);
    procedure B_pauseClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure SavePicture(count:integer);

    private
    { Private declarations }
    Bitmap: TBitmap;
    Procedure Initial;
    Procedure start;
    Procedure Next;
    Procedure wait;
    Procedure Newpicture;
    procedure ScrambleBitmap;
    Procedure ReaddataAll;
    Function Read4BitHigh:byte;
    Function Read4Bitlow:byte;
    procedure check;
  public
    { Public declarations }
  end;

  var
   Form2:TForm2;
   data:byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  implementation
uses home;
  {$R *.DFM}



Function TForm2.Read4BitHigh:byte;
var data:byte;
begin
        with VicHw32 do
        begin
        Port[$378]:=$FB;
        data:=port[$379] and $F0;
        end;
        Read4BitHigh:=data;
end;

Function TForm2.Read4BitLow:byte;
var data:byte;

begin
        with VicHw32 do
        begin
        Port[$378]:=$FD;
        data:=Port[$379] shr 4;
        end;
        Read4BitLow:=data;
end;

Procedure Tform2.Initial;
begin
        VicHw32.Port[$378]:=$FF;
end;

Procedure Tform2.start;
begin
        VicHw32.Port[$378]:=$FE;
end;

Procedure Tform2.Next;
begin
        with VicHw32 do
        begin
        Port[$378]:=$EF;
        Port[$378]:=$FF;
        end;
end;

Procedure Tform2.Wait;
var ready:boolean;
begin
        with VicHw32 do
        begin
            repeat
            ready:=not(Pin[18]);
            Application.ProcessMessages;
            until(ready);
         end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure Tform2.check;
var a,i:integer;
begin
        i:=1 ;
        a:=1;
        repeat
            repeat
            wait;
            data:=read4BitHigh;
            data:=data or read4BitLow;
            Next;
            a:=a+1;
            if  a=256 then
                begin
                Newpicture;
                initial;
                start;
                end;
            until data<68;
            i:=i+1;
           until  i>2;

end;
Procedure Tform2.Newpicture;
begin
        with VicHw32 do
        begin
        Port[$378]:=$bf;
        end;
end;

Procedure TForm2.FormCreate(Sender: TObject);
begin
        Bitmap := TBitmap.Create;
        Bitmap.LoadFromFile('buffer.bmp');
        VicHw32.OpenDriver('TVicDevice0');
        if not VicHw32.ActiveHW then
        begin
            messagebeep(0);
            ShowMessage('Can''t open the driver!');
        end
        else begin
        statusbar1.Panels[0].Text:='Status: Wait To Play ';
        initial;
        end;
end;


procedure TForm2.FormDestroy(Sender: TObject);
begin
        Bitmap.Free;
        Vichw32.Port[Vichw32.LPTBasePort+2]:=$00;
        VicHw32.CloseDriver;

end;

procedure TForm2.ScrambleBitmap;
var pal: PLogPalette;
    hpal: HPALETTE;
    i: Integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
        pal := nil;
        try
        GetMem(pal, sizeof(TLogPalette) + sizeof(TPaletteEntry) *
255);
        pal.palVersion := $300;
        pal.palNumEntries := 256;
        for i := 0 to 255 do
        begin
           pal.palPalEntry[i].peRed :=i;
           pal.palPalEntry[i].peGreen :=i;
           pal.palPalEntry[i].peBlue := i;
        end;
        hpal := CreatePalette(pal^);
        if hpal <> 0 then
          Bitmap.Palette := hpal;
        finally
          FreeMem(pal);
        end;
end;


Procedure TForm2.ReaddataAll;
var x,y:integer;
    step:integer;
    data,data1,data2:byte ;
    p:pByteArray;
begin
         y:=0;
         repeat
         x:=0;
         check;
         progressbar1.StepIt;
         p:=Bitmap.ScanLine[y];
         wait;
         data:=read4Bithigh;
         data1:=data or read4Bitlow;
         Next;
            repeat
            wait;
            data:=read4bithigh;
            data2:=data or read4bitlow;
            step:=(data2-data1) div 4;
            p[x]:=data1;
            x:=x+1;
            p[x]:=data1+step ;
            x:=x+1;
            p[x]:=data1+(2*step);
            x:=x+1;
            p[x]:=data1+(3*step);
            x:=x+1;
          //  p[x]:=data1+(4*step);
         //   x:=x+1;
            data1:=data2;
            next;
          until x>1000;
         y:=y+1;
         until y>265;

end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm2.B_playClick(Sender: TObject);
var count:integer;
begin
            repeat
            statusbar1.Panels[0].Text:='Status: Loading Pictures';
            B_exit.SetFocus;
            B_play.Enabled:=false;
            B_pause.Enabled:=true;
            count:=0;
            Scramblebitmap;
            repeat
            initial;
            Progressbar1.Enabled:=true;
            progressbar1.Visible:=true;
            start;
            Readdataall;
            progressbar1.Visible:=false;
            Progressbar1.Enabled:=false;
            image1.Canvas.Draw(0,0,Bitmap);
            SavePicture(count);
            count:=count+1;
            Newpicture;
            until count=41;
            until count=0;
end;

procedure Tform2.SavePicture(count:integer);
begin
            case count of
            1: begin
               image1.Picture.SaveToFile('H:\data\s1.bmp');
               image1.Picture.SaveToFile('H:\data2\s1.bmp');
               end;
            2: begin
               image1.Picture.SaveToFile('H:\data\s2.bmp');
               image1.Picture.SaveToFile('H:\data2\s2.bmp');
               end;
            3: begin
               image1.Picture.SaveToFile('H:\data\s3.bmp');
               image1.Picture.SaveToFile('H:\data2\s3.bmp');
               end;
            4: begin
               image1.Picture.SaveToFile('H:\data\s4.bmp');
               image1.Picture.SaveToFile('H:\data2\s4.bmp');
               end;
            5: begin
               image1.Picture.SaveToFile('H:\data\s5.bmp');
               image1.Picture.SaveToFile('H:\data2\s5.bmp');
               end;
            6: begin
               image1.Picture.SaveToFile('H:\data\s6.bmp');
               image1.Picture.SaveToFile('H:\data2\s6.bmp');
               end;
            7: begin
               image1.Picture.SaveToFile('H:\data\s7.bmp');
               image1.Picture.SaveToFile('H:\data2\s7.bmp');
               end;
            8: begin
               image1.Picture.SaveToFile('H:\data\s8.bmp');
               image1.Picture.SaveToFile('H:\data2\s8.bmp');
               end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
9: begin
   image1.Picture.SaveToFile('H:\data\s9.bmp');
   image1.Picture.SaveToFile('H:\data2\s9.bmp');
   end;
10: begin
   image1.Picture.SaveToFile('H:\data\s10.bmp');
   image1.Picture.SaveToFile('H:\data2\s10.bmp');
   end;
11: begin
   image1.Picture.SaveToFile('H:\data\s11.bmp');
   image1.Picture.SaveToFile('H:\data2\s11.bmp');
   end;
12: begin
   image1.Picture.SaveToFile('H:\data\s12.bmp');
   image1.Picture.SaveToFile('H:\data2\s12.bmp');
   end;
13: begin
   image1.Picture.SaveToFile('H:\data\s13.bmp');
   image1.Picture.SaveToFile('H:\data2\s13.bmp');
   end;
14: begin
   image1.Picture.SaveToFile('H:\data\s14.bmp');
   image1.Picture.SaveToFile('H:\data2\s14.bmp');
   end;
15: begin
   image1.Picture.SaveToFile('H:\data\s15.bmp');
   image1.Picture.SaveToFile('H:\data2\s15.bmp');
   end;
16: begin
   image1.Picture.SaveToFile('H:\data\s16.bmp');
   image1.Picture.SaveToFile('H:\data2\s16.bmp');
   end;
17: begin
   image1.Picture.SaveToFile('H:\data\s17.bmp');
   image1.Picture.SaveToFile('H:\data2\s17.bmp');
   end;
18: begin
   image1.Picture.SaveToFile('H:\data\s18.bmp');
   image1.Picture.SaveToFile('H:\data2\s18.bmp');
   end;
19: begin
   image1.Picture.SaveToFile('H:\data\s19.bmp');
   image1.Picture.SaveToFile('H:\data2\s19.bmp');
   end;
20: begin
   image1.Picture.SaveToFile('H:\data\s20.bmp');
   image1.Picture.SaveToFile('H:\data2\s20.bmp');
   end;
21: begin
   image1.Picture.SaveToFile('H:\data\s21.bmp');
   image1.Picture.SaveToFile('H:\data2\s21.bmp');
   end;
22: begin
   image1.Picture.SaveToFile('H:\data\s22.bmp');
   image1.Picture.SaveToFile('H:\data2\s22.bmp');
   end;
23: begin
   image1.Picture.SaveToFile('H:\data\s23.bmp');
   image1.Picture.SaveToFile('H:\data2\s23.bmp');
   end;
24: begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    image1.Picture.SaveToFile('H:\data\s24.bmp');
    image1.Picture.SaveToFile('H:\data2\s24.bmp');
    end;
25: begin
    image1.Picture.SaveToFile('H:\data\s25.bmp');
    image1.Picture.SaveToFile('H:\data2\s25.bmp');
    end;
26: begin
    image1.Picture.SaveToFile('H:\data\s26.bmp');
    image1.Picture.SaveToFile('H:\data2\s26.bmp');
    end;
27: begin
    image1.Picture.SaveToFile('H:\data\s27.bmp');
    image1.Picture.SaveToFile('H:\data2\s27.bmp');
    end;
28: begin
    image1.Picture.SaveToFile('H:\data\s28.bmp');
    image1.Picture.SaveToFile('H:\data2\s28.bmp');
    end;
29: begin
    image1.Picture.SaveToFile('H:\data\s29.bmp');
    image1.Picture.SaveToFile('H:\data2\s29.bmp');
    end;
30: begin
    image1.Picture.SaveToFile('H:\data\s30.bmp');
    image1.Picture.SaveToFile('H:\data2\s30.bmp');
    end;
31: begin
    image1.Picture.SaveToFile('H:\data\s31.bmp');
    image1.Picture.SaveToFile('H:\data2\s31.bmp');
    end;
32: begin
    image1.Picture.SaveToFile('H:\data\s32.bmp');
    image1.Picture.SaveToFile('H:\data2\s32.bmp');
    end;
33: begin
    image1.Picture.SaveToFile('H:\data\s33.bmp');
    image1.Picture.SaveToFile('H:\data2\s33.bmp');
    end;
34: begin
    image1.Picture.SaveToFile('H:\data\s34.bmp');
    image1.Picture.SaveToFile('H:\data2\s34.bmp');
    end;
35: begin
    image1.Picture.SaveToFile('H:\data\s35.bmp');
    image1.Picture.SaveToFile('H:\data2\s35.bmp');
    end;
36: begin
    image1.Picture.SaveToFile('H:\data\s36.bmp');
    image1.Picture.SaveToFile('H:\data2\s36.bmp');
    end;
37: begin
    image1.Picture.SaveToFile('H:\data\s37.bmp');
    image1.Picture.SaveToFile('H:\data2\s37.bmp');
    end;
38: begin
    image1.Picture.SaveToFile('H:\data\s38.bmp');
    image1.Picture.SaveToFile('H:\data2\s38.bmp');
    end;
39: begin
    image1.Picture.SaveToFile('H:\data\s39.bmp');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            image1.Picture.SaveToFile('H:\data2\s39.bmp');
            end;
         40: begin
            image1.Picture.SaveToFile('H:\data\s40.bmp');
            image1.Picture.SaveToFile('H:\data2\s40.bmp');
            end;
       end;
end;

procedure TForm2.B_exitClick(Sender: TObject);
begin
      halt;
      form1.Close;
      close;
end;

procedure TForm2.SaveClick(Sender: TObject);
var name:string;
begin
    if SavePictureDialog1.Execute then
    begin
    name:=SavePictureDialog1.FileName;
    Image1.Picture.SaveToFile(name);
    end;
end;

procedure TForm2.OpenClick(Sender: TObject);
var name:string;
begin
     if OpenPictureDialog1.Execute then
     begin
     name:=OpenPictureDialog1.FileName;
     image1.Picture.LoadFromFile(name);
     end;
end;

procedure TForm2.B_pauseClick(Sender: TObject);
begin
      B_pause.Enabled:=false;
      B_open.Enabled:=true;
      B_save.Enabled:=true;
      sleep(1000);
      B_pause.Enabled:=true;
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
     statusbar1.Panels[1].Text:='Date :'+datetostr(date);
     statusbar1.Panels[2].Text:='Time :'+Timetostr(Time);
end;



end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของไมโครคอนโทรลเลอร์

```
            ORG 0000H
            WRITE   EQU     0010H
            READ    EQU     000EH

MAIN:       CLR  P1.4
            JB   P1.5,$
            ANL  P1,#0E1H
            ORL  P1,#WRITE
            CALL DELAY
            ANL  P1,#0E1H
            ORL  P1,#READ
            CALL  SENT
            JB   P3.2,$
            MOV P1,0FFH
            JMP  MAIN

SENT:       MOV  DPTR,#0000H
CHECK1:     MOVX A,@DPTR
            MOVX @DPTR,A
            CLR P1.0
            JB   P1.7,$
            SETB P1.0
            MOV A,#00H
            MOV  A,DPL
            CJNE A,#0FFH,CHECK2
            MOV  A,DPH
            CJNE A,#0FFH,CHECK2
            RET
CHECK2:     INC  DPTR
            JMP  CHECK1

DELAY:      MOV  R2,#8
DEL1:       MOV  R3,#50
DEL2:       MOV  R4,#50
            DJNZ R4,$
            DJNZ R3,DEL2
            DJNZ R2,DEL1
            RET
            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intersil

CA3318

CMOS Video Speed, 8-Bit, Flash A/D Converter

August 1997

Features

- CMOS Low Power with SOS Speed (Typ)..... 150mW
- Parallel Conversion Technique
- 15MHz Sampling Rate (Conversion Time)..... 67ns
- 8-Bit Latched Three-State Output with Overflow Bit
- Accuracy (Typ)..... ±1 LSB
- Single Supply Voltage..... 4V to 7.5V
- 2 Units in Series Allow 9-Bit Output
- 2 Units in Parallel Allow 30MHz Sampling Rate

Applications

- TV Video Digitizing (Industrial/Security/Broadcast)
- High Speed A/D Conversion
- Ultrasound Signature Analysis
- Transient Signal Analysis
- High Energy Physics Research
- General-Purpose Hybrid ADCs
- Optical Character Recognition
- Radar Pulse Analysis
- Motion Signature Analysis
- µP Data Acquisition Systems

Description

The CA3318 is a CMOS parallel (FLASH) analog-to-digital converter designed for applications demanding both low power consumption and high speed digitization.

The CA3318 operates over a wide full scale input voltage range of 4V up to 7.5V with maximum power consumption depending upon the clock frequency selected. When operated from a 5V supply at a clock frequency of 15MHz, the typical power consumption of the CA3318 is 150mW.

The intrinsic high conversion rate makes the CA3318 ideally suited for digitizing high speed signals. The overflow bit makes possible the connection of two or more CA3318s in series to increase the resolution of the conversion system. A series connection of two CA3318s may be used to produce a 9-bit high speed converter. Operation of two CA3318s in parallel doubles the conversion speed (i.e., increases the sampling rate from 15MHz to 30MHz).

256 paralleled auto balanced voltage comparators measure the input voltage with respect to a known reference to produce the parallel bit outputs in the CA3318.

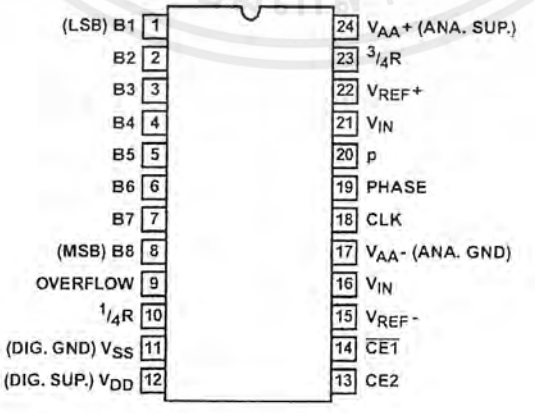
255 comparators are required to quantize all input voltage levels in this 8-bit converter, and the additional comparator is required for the overflow bit.

Ordering Information

PART NUMBER	LINEARITY (INL, DNL)	SAMPLING RATE	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
CA3318CE	±1.5 LSB	15MHz (67ns)	-40 to 85	24 Ld PDIP	E24.6
CA3318CM	±1.5 LSB	15MHz (67ns)	-40 to 85	24 Ld SOIC	M24.3
CA3318CD	±1.5 LSB	15MHz (67ns)	-40 to 85	24 Ld SBDIP	D24.6

Pinout

CA3318
(PDIP, SBDIP, SOIC)
TOP VIEW



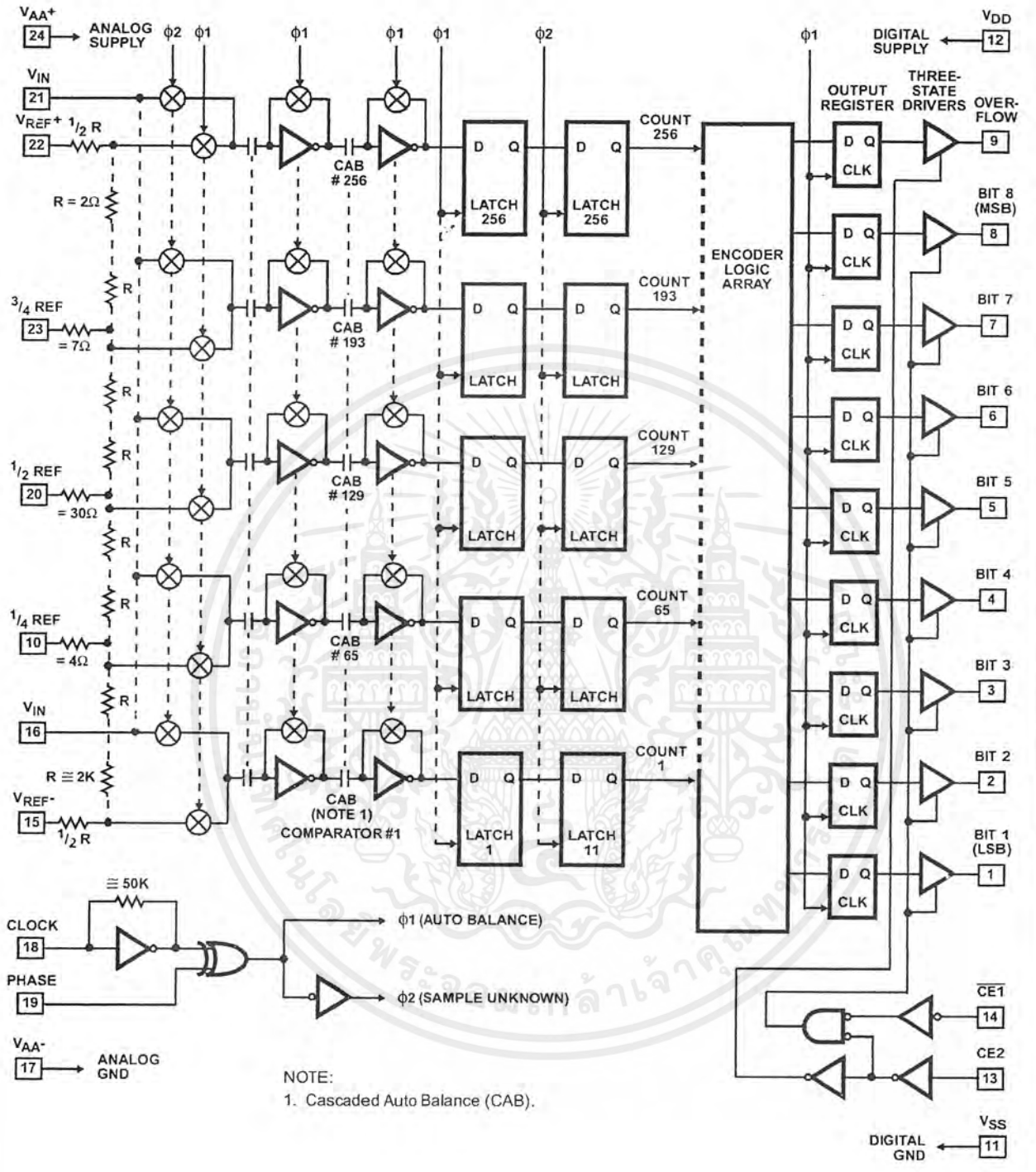
CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.

http://www.intersil.com or 407-727-9207 | Copyright © Intersil Corporation 1999

File Number 3103.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Block Diagram



NOTE:

1. Cascaded Auto Balance (CAB).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

DC Supply Voltage Range (V_{DD} or V_{AA}+) . . . . . -0.5V to +8V
(Referenced to V_{SS} or V_{AA}- Terminal, Whichever is More Negative)
Input Voltage Range
CE2 and $\overline{CE1}$ . . . . . V_{AA}- -0.5V to V_{DD} + 0.5V
Clock, Phase, V_{REF}-, $^1/_2$ Ref . . . . . V_{AA}- -0.5V to V_{AA}+ + 0.5V
Clock, Phase, V_{REF}-, $^1/_4$ Ref . . . . . V_{SS}- -0.5V to V_{DD} + 0.5V
V_{IN}, $^3/_4$ REF, V_{REF}+ . . . . . V_{AA}- -0.5V to V_{AA}- + 7.5V
Output Voltage Range, ... V_{SS} - 0.5V to V_{DD} + 0.5V
Bits 1-8, Overflow (Outputs Off)
DC Input Current . . . . . ±20mA
Clock, Phase, $\overline{CE1}$, CE2, V_{IN}, Bits 1-8, Overflow

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} ($^oC/W$)	θ_{JC} ($^oC/W$)
SBDIP Package	60	22
PDIP Package	60	N/A
SOIC Package	75	N/A

Maximum Junction Temperature
Ceramic Package . . . . . 175°C
Plastic Packages . . . . . 150°C
Maximum Storage Temperature Range . . . . . -65°C to 150°C
Maximum Lead Temperature (Soldering 10s) . . . . . 265°C
(SOIC - Lead Tips Only)

Operating Conditions

Operating Voltage Range (V_{DD} or V_{AA}+) . . . 4V (Min) to 7.5V (Max)
Recommended V_{AA}+ Operating Range . . . . . V_{DD} ±1V
Recommended V_{AA}- Operating Range . . . . . V_{SS} ±1V
Operating Temperature Range (T_A) . . . . . -40°C to 85°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Electrical Specifications

At 25°C, V_{AA}+ = V_{DD} = 5V, V_{REF}+ = 6.4V, V_{REF}- = V_{AA}- = V_{SS}, CLK = 15MHz, All Reference Points Adjusted, Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
SYSTEM PERFORMANCE					
Resolution		8	-	-	Bits
Integral Linearity Error		-	-	±1.5	LSB
Differential Linearity Error		-	-	+1, -0.8	LSB
Offset Error, Unadjusted	V_{IN} = V_{REF}- + $^1/_2$ LSB	-0.5	4.5	6.4	LSB
Gain Error Unadjusted	V_{IN} = V_{REF}+ - $^1/_2$ LSB	-1.5	0	1.5	LSB
DYNAMIC CHARACTERISTICS					
Maximum Input Bandwidth	(Note 1) CA3318	2.5	5.0	-	MHz
Maximum Conversion Speed	CLK = Square Wave	15	17	-	MSPS
Signal to Noise Ratio (SNR) $=\frac{RMSSignal}{RMSNoise}$	f_S = 15MHz, f_{IN} = 100kHz	-	47	-	dB
	f_S = 15MHz, f_{IN} = 4MHz	-	43	-	dB
Signal to Noise Ratio (SINAD) $=\frac{RMSSignal}{RMSNoise+Distortion}$	f_S = 15MHz, f_{IN} = 100kHz	-	45	-	dB
	f_S = 15MHz, f_{IN} = 4MHz	-	35	-	dB
Total Harmonic Distortion, THD	f_S = 15MHz, f_{IN} = 100kHz	-	-46	-	dBc
	f_S = 15MHz, f_{IN} = 4MHz	-	-36	-	dBc
Effective Number of Bits (ENOB)	f_S = 15MHz, f_{IN} = 100kHz	-	7.2	-	Bits
	f_S = 15MHz, f_{IN} = 4MHz	-	5.5	-	Bits
Differential Gain Error	Unadjusted	-	2	-	%
Differential Phase Error	Unadjusted	-	1	-	%
ANALOG INPUTS					
Full Scale Range, V_{IN} and (V_{REF}+) - (V_{REF}-)	Notes 2, 4	4	-	7	V
Input Capacitance, V_{IN}		-	30	-	pF
Input Current, V_{IN}, (See Text)	V_{IN} = 5V, V_{REF}+ = 5V	-	-	3.5	mA
REFERENCE INPUTS					
Ladder Impedance		270	500	800	Ω

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Specifications At 25°C, $V_{AA}+ = V_{DD} = 5V$, $V_{REF}+ = 6.4V$, $V_{REF}- = V_{AA}- = V_{SS}$, CLK = 15MHz, All Reference Points Adjusted, Unless Otherwise Specified **(Continued)**

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
DIGITAL INPUTS					
Low Level Input Voltage, V_{OL}					
$\overline{CE1}$, CE2	Note 4	-	-	$0.2V_{DD}$	V
Phase, CLK	Note 4	-	-	$0.2V_{AA}$	V
High Level Input Voltage, V_{IN}					
$\overline{CE1}$, CE2	Note 4	$0.7V_{DD}$	-	-	V
Phase, CLK	Note 4	$0.7V_{AA}$	-	-	V
Input Leakage Current, I_I (Except CLK Input)	Note 3	-	±0.2	±5	µA
Input Capacitance, C_I		-	3	-	pF
DIGITAL OUTPUTS					
Output Low (Sink) Current	$V_O = 0.4V$	4	10	-	mA
Output High (Source) Current	$V_O = 4.5V$	-4	-6	-	mA
Three-State Output Off-State Leakage Current, I_{OZ}		-	±0.2	±5	µA
Output Capacitance, C_O		-	4	-	pF
TIMING CHARACTERISTICS					
Auto Balance Time (ϕ1)		33	-	∞	ns
Sample Time (ϕ2)	Note 4	25	-	500	ns
Aperture Delay		-	15	-	ns
Aperture Jitter		-	100	-	ps
Data Valid Time, t_D	Note 4	-	50	65	ns
Data Hold Time, t_H	Note 4	25	40	-	ns
Output Enable Time, t_{EN}		-	18	-	ns
Output Disable Time, t_{DIS}		-	18	-	ns
POWER SUPPLY CHARACTERISTICS					
Device Current ($I_{DD} + I_A$) (Excludes I_{REF})	Continuous Conversion (Note 4)	-	30	60	mA
	Auto Balance (ϕ1)	-	30	60	mA

NOTES:

1. A full scale sine wave input of greater than $f_{CLOCK}/2$ or the specified input bandwidth (whichever is less) may cause an erroneous code. The -3dB bandwidth for frequency response purposes is greater than 30MHz.
2. V_{IN} (Full Scale) or $V_{REF}+$ should not exceed $V_{AA}+ + 1.5V$ for accuracy.
3. The clock input is a CMOS inverter with a 50kΩ feedback resistor and may be AC coupled with $1V_{P-P}$ minimum source.
4. Parameter not tested, but guaranteed by design or characterization.

Timing Waveforms

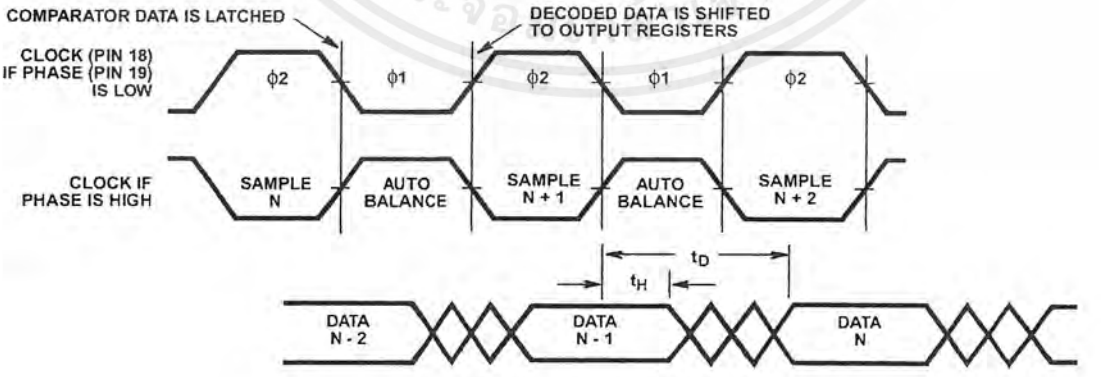


FIGURE 1. INPUT TO OUTPUT TIMING DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing Waveforms (Continued)

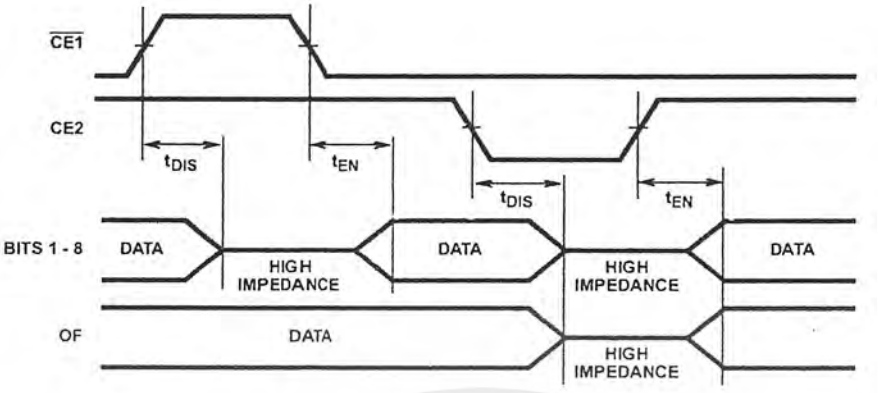


FIGURE 2. OUTPUT ENABLE TIMING DIAGRAM

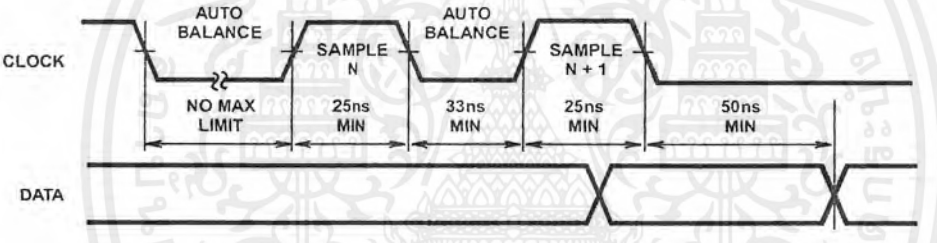


FIGURE 3A. STANDBY IN INDEFINITE AUTO BALANCE (SHOWN WITH PHASE = LOW)

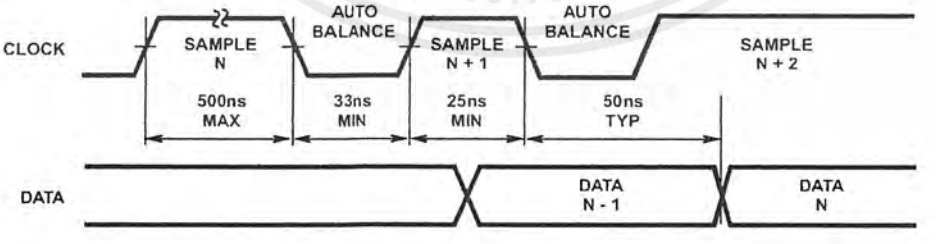


FIGURE 3B. STANDBY IN SAMPLE (SHOWN WITH PHASE = LOW)

FIGURE 3. PULSE MODE OPERATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves

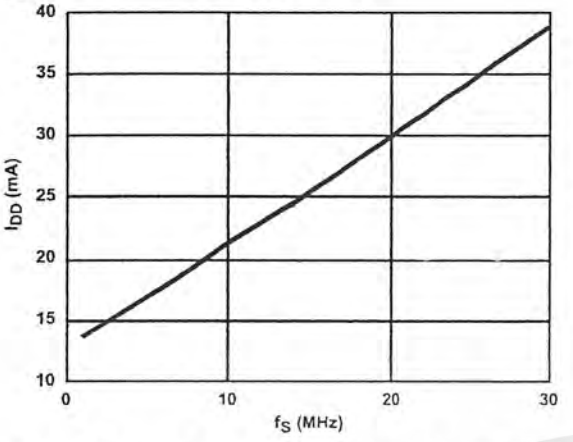


FIGURE 4. DEVICE CURRENT vs SAMPLE FREQUENCY

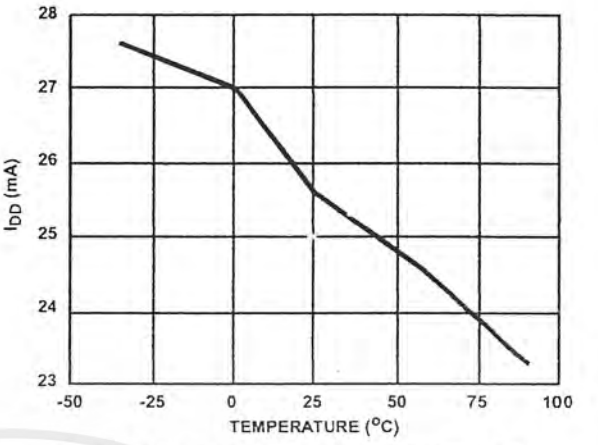


FIGURE 5. DEVICE CURRENT vs TEMPERATURE

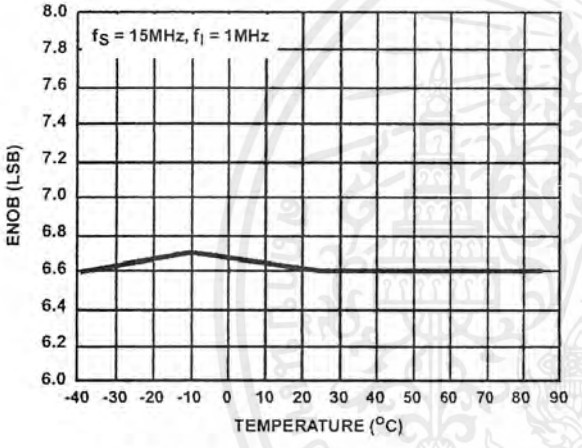


FIGURE 6. ENOB vs TEMPERATURE

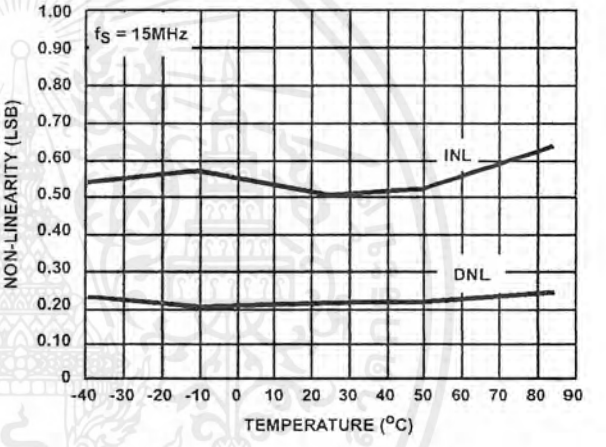


FIGURE 7. NON-LINEARITY vs TEMPERATURE

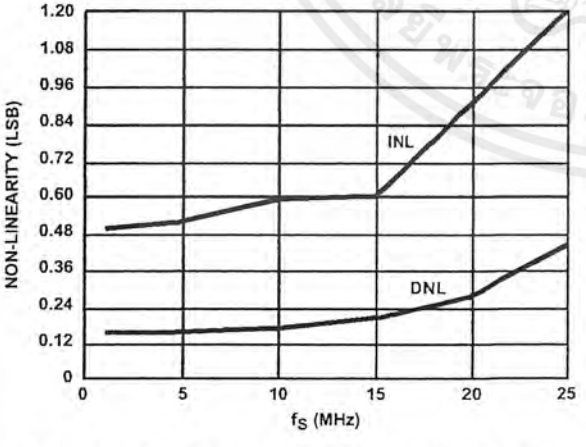


FIGURE 8. NON-LINEARITY vs SAMPLE FREQUENCY

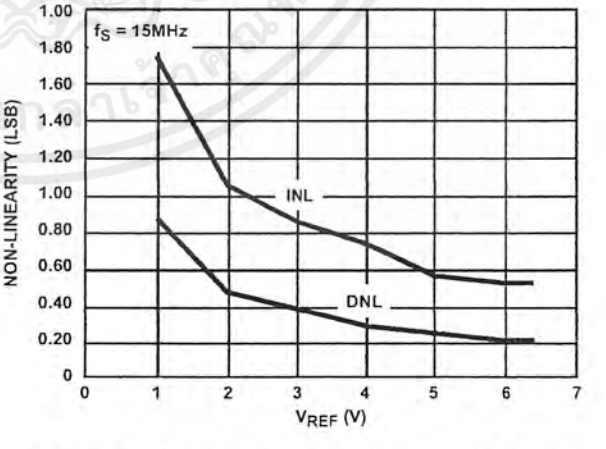


FIGURE 9. NON-LINEARITY vs REFERENCE VOLTAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves (Continued)

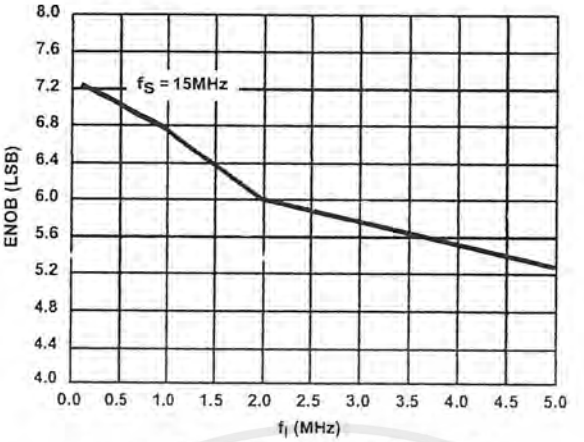


FIGURE 10. ENOB vs INPUT FREQUENCY

Pin Descriptions

PIN	NAME	DESCRIPTION	
1	B1	Bit 1 (LSB)	Output Data Bits (High = True)
2	B2	Bit 2	
3	B3	Bit 3	
4	B4	Bit 4	
5	B5	Bit 5	
6	B6	Bit 6	
7	B7	Bit 7	
8	B8	Bit 8 (MSB)	
9	OF	Overflow	
10	$^1/_4$ R	Reference Ladder $^1/_4$ Point	
11	V_{SS}	Digital Ground	
12	V_{DD}	Digital Power Supply, +5V	
13	CE2	Three-State Output Enable Input, Active Low, See Truth Table.	
14	$\overline{CE1}$	Three-State Output Enable Input Active High. See Truth Table.	
15	V_{REF}-	Reference Voltage Negative Input	
16	V_{IN}	Analog Signal Input	
17	V_{AA}-	Analog Ground	
18	CLK	Clock Input	
19	PHASE	Sample clock phase control input. When PHASE is low, "Sample Unknown" occurs when the clock is low and "Auto Balance" occurs when the clock is high (see text).	
20	$^1/_2$ R	Reference Ladder Midpoint	
21	V_{IN}	Analog Signal Input	
22	V_{REF}+	Reference Voltage Positive Input	
23	$^3/_4$ R	Reference Ladder $^3/_4$ Point	
24	V_{AA}+	Analog Power Supply, +5V	

CHIP ENABLE TRUTH TABLE

$\overline{CE1}$	CE2	B1 - B8	OF
0	1	Valid	Valid
1	1	Three-State	Valid
X	0	Three-State	Three-State

X = Don't Care

Theory of Operation

A sequential parallel technique is used by the CA3318 converter to obtain its high speed operation. The sequence consists of the "Auto-Balance" phase, ϕ1, and the "Sample Unknown" phase, ϕ2. (Refer to the circuit diagram.) Each conversion takes one clock cycle (see Note). With the phase control (pin 19) high, the "Auto-Balance" (ϕ1) occurs during the high period of the clock cycle, and the "Sample Unknown" (ϕ2) occurs during the low period of the clock cycle.

NOTE: The device requires only a single phase clock The terminology of ϕ1 and ϕ2 refers to the high and low periods of the same clock.

During the "Auto-Balance" phase, a transmission switch is used to connect each of the first set of 256 commutating capacitors to their associated ladder reference tap. Those tap voltages will be as follows:

$$V_{TAP}(N) = [(N/256)\,V_{REF}] - (1/512)\,V_{REF}]$$
$$= [(2N - 1)/512]\,V_{REF},$$

Where:

V_{TAP} (n) = reference ladder tap voltage at point n,

V_{REF} = voltage across V_{REF}- to V_{REF}+,

N = tap number (1 through 256).

The other side of these capacitors are connected to single-stage amplifiers whose outputs are shorted to their inputs by switches. This balances the amplifiers at their intrinsic trip points, which is approximately (V_{AA}+ - V_{AA}-)/2. The first set of capacitors now charges to their associated tap voltages.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

At the same time a second set of commutating capacitors and amplifiers is also auto-balanced. The balancing of the second-stage amplifier at its intrinsic trip point removes any tracking differences between the first and second amplifier stages. The cascaded auto-balance (CAB) technique, used here, increases comparator sensitivity and temperature tracking.

In the "Sample Unknown" phase, all ladder tap switches and comparator shorting switches are opened. At the same time V_{IN} is switched to the first set of commutating capacitors. Since the other end of the capacitors are now looking into an effectively open circuit, any input voltage that differs from the previous tap voltage will appear as a voltage shift at the comparator amplifiers. All comparators that had tap voltages greater than V_{IN} will go to a "high" state at their outputs. All comparators that had tap voltages lower than V_{IN} will go to a "low" state.

The status of all these comparator amplifiers is AC coupled through the second-stage comparator and stored at the end of this phase ($\phi 2$) by a latching amplifier stage. The latch feeds a second latching stage, triggered at the end of $\phi 1$. This delay allows comparators extra settling time. The status of the comparators is decoded by a 256 to 9-bit decoder array, and the results are clocked into a storage register at the end of the next $\phi 2$.

A 3-stage buffer is used at the output of the 9 storage registers which are controlled by two chip-enable signals. $\overline{CE1}$ will independently disable B1 through B6 when it is in a high state. CE2 will independently disable B1 through B8 and the OF buffers when it is in the low state.

To facilitate usage of this device, a phase control input is provided which can effectively complement the clock as it enters the chip.

Continuous-Clock Operation

One complete conversion cycle can be traced through the CA3318 via the following steps. (Refer to timing diagram.) With the phase control in a "low" state, the rising edge of the clock input will start a "sample" phase. During this entire "high" state of the clock, the comparators will track the input voltage and the first-stage latches will track the comparator outputs. At the falling edge of the clock, all 256 comparator outputs are captured by the 256 latches. This ends the "sample" phase and starts the "auto-balance" phase for the comparators. During this "low" state of the clock, the output of the latches settles and is captured by a second row of latches when the clock returns high. The second-stage latch output propagates through the decode array, and a 9-bit code appears at the D inputs of the output registers. On the next falling edge of the clock, this 9-bit code is shifted into the output registers and appears with time delay t_D as valid data at the output of the three-state drivers. This also marks the end of the next "sample" phase, thereby repeating the conversion process for this next cycle.

Pulse-Mode Operation

The CA3318 needs two of the same polarity clock edges to complete a conversion cycle: If, for instance, a negative going clock edge ends sample "N", then data "N" will appear after the next negative going edge. Because of this requirement, and because there is a maximum sample time of 500ns (due to capacitor droop), most pulse or intermittent sample applications will require double clock pulsing.

If an indefinite standby state is desired, standby should be in auto-balance, and the operation would be as in Figure 3A.

If the standby state is known to last less than 500ns and lowest average power is desired, then operation could be as in Figure 3B.

Increased Accuracy

In most cases the accuracy of the CA3318 should be sufficient without any adjustments. In applications where accuracy is of utmost importance, five adjustments can be made to obtain better accuracy, i.e., offset trim; gain trim; and $^1/_4$, $^1/_2$ and $^3/_4$ point trim.

Offset Trim

In general, offset correction can be done in the preamp circuitry by introducing a DC shift to V_{IN} or by the offset trim of the op amp. When this is not possible the $V_{REF}-$ input can be adjusted to produce an offset trim. The theoretical input voltage to produce the first transition is $^1/_2$ LSB. The equation is as follows:

$$V_{IN}\ (0 \text{ to } 1 \text{ transition}) = {}^1/_2 \text{ LSB} = {}^1/_2\ (V_{REF}/256)$$
$$= V_{REF}/512.$$

If V_{IN} for the first transition is less than the theoretical, then a single-turn 50Ω pot connected between $V_{REF}-$ and ground will accomplish the adjustment. Set V_{IN} to 1/2 LSB and trim the pot until the 0-to-1 transition occurs.

If V_{IN} for the first transition is greater than the theoretical, then the 50Ω pot should be connected between $V_{REF}-$ and a negative voltage of about 2 LSBs. The trim procedure is as stated previously.

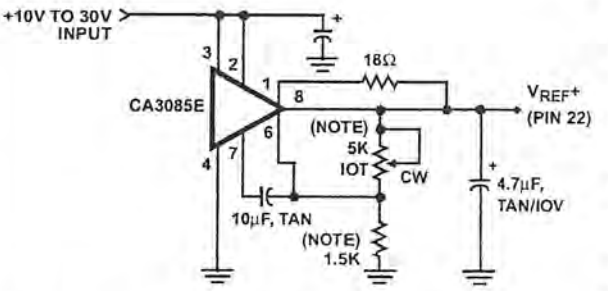
Gain Trim

In general, the gain trim can also be done in the preamp circuitry by introducing a gain adjustment for the op amp. When this is not possible, then a gain adjustment circuit should be made to adjust the reference voltage. To perform this trim, V_{IN} should be set to the 255 to overflow transition. That voltage is $^1/_3$ LSB less than $V_{REF}+$ and is calculated as follows:

$$V_{IN}\ (255 \text{ to } 256 \text{ transition}) = V_{REF} - V_{REF}/512$$
$$= V_{REF}(511/512).$$

To perform the gain trim, first do the offset trim and then apply the required V_{IN} for the 255 to overflow transition. Now adjust $V_{REF}+$ until that transition occurs on the outputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

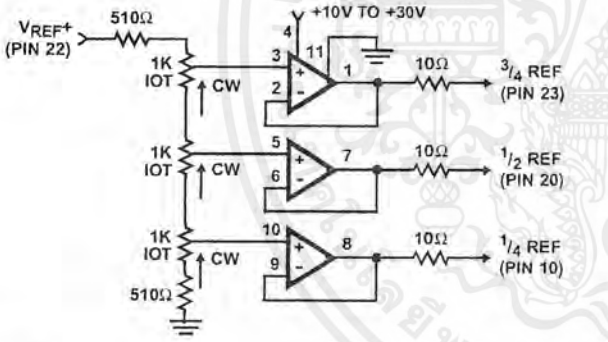


NOTE: Bypass V_{REF}+ to analog GND near A/D with 0.1μF ceramic cap. Parts noted should have low temperature drift.

FIGURE 11. TYPICAL VOLTAGE REFERENCE SOURCE FOR DRIVING V_{REF}+ INPUT

$^1/_4$ Point Trims

The $^1/_4$, $^1/_2$ and $^3/_4$ points on the reference ladder are brought out for linearity adjusting or if the user wishes to create a nonlinear transfer function. The $^1/_4$ points can be driven by the reference drivers shown (Figure 12) or by 2-K pots connected between V_{REF}+ and V_{REF}-. The $^1/_2$ (mid-) point should be set first by applying an input of 257/512 x (V_{REF}) and adjusting for an output changing from 128 to 129. Similarly the $^1/_4$ and $^3/_4$ points can be set with inputs of 129/512 and 385/512 x (V_{REF}) and adjusting for counts of 192 to 193 and 64 to 65. (Note that the points are actually $^1/_4$, $^1/_2$ and $^3/_4$ of full scale +1 LSB.)



NOTES:

1. All Op Amps = $^3/_4$ CA324E.
2. Bypass all reference points to analog ground near A/D with 0.1μF ceramic caps.
3. Adjust V_{REF}+ first, then $^1/_3$, $^3/_4$ and $^1/_4$ points.

FIGURE 12. TYPICAL $^1/_4$ POINT DRIVERS FOR ADJUSTING LINEARITY (USE FOR MAXIMUM LINEARITY)

9-Bit Resolution

To obtain 9-bit resolution, two CA3318s can be wired together. Necessary ingredients include an open-ended ladder network, an overflow indicator, three-state outputs, and chip-enable controls - all of which are available on the CA3318.

The first step for connecting a 9-bit circuit is to totem-pole the ladder networks, as illustrated in Figure 13. Since the absolute resistance value of each ladder may vary, external trim of the mid-reference voltage may be required.

The overflow output of the lower device now becomes the ninth bit. When it goes high, all counts must come from the upper device. When it goes low, all counts must come from the lower device. This is done simply by connecting the lower overflow signal to the CE1 control of the lower A/D converter and the CE2 control of the upper A/D converter. The three-state outputs of the two devices (bits 1 through 8) are now connected in parallel to complete the circuitry. The complete circuit for a 9-bit A/D converter is shown in Figure 13.

Grounding/Bypassing

The analog and digital supply grounds of a system should be kept separate and only connected at the A/D. This keeps digital ground noise out of the analog data to be converted. Reference drivers, input amps, reference taps, and the V_{AA} supply should be bypassed at the A/D to the analog side of the ground. See Figure 15 for a block diagram of this concept. All capacitors shown should be low impedance 0.1μF ceramics and should be mounted as close to the A/D as possible. If V_{AA}+ is derived from V_{DD}, a small (10Ω resistor or inductor and additional filtering (4.7μF tantalum) may be used to keep digital noise out of the analog system.

Input Loading

The CA3318 outputs a current pulse to the V_{IN} terminal at the start of every sample period. This is due to capacitor charging and switch feedthrough and varies with input voltage and sampling rate. The signal source must be capable of recovering from the pulse before the end of the sample period to guarantee a valid signal for the A/D to convert. Suitable high speed amplifiers include the HA-5033, HA-2542; and CA3450. Figure 16 is an example of an amplifier which recovers fast enough for sampling at 15MHz.

Output Loading

The CMOS digital output stage, although capable of driving large loads, will reflect these loads into the local ground. It is recommended that a local QMOS buffer such as CD74HC541 E be used to isolate capacitive loads.

Definitions

Dynamic Performance Definitions

Fast Fourier Transform (FFT) techniques are used to evaluate the dynamic performance of the converter. A low distortion sine wave is applied to the input, it is sampled, and the output is stored in RAM. The data is then transformed into the frequency domain with a 4096 point FFT and analyzed to evaluate the dynamic performance of the A/D. The sine wave input to the part is -0.5dB down from fullscale for all these tests.

Signal-to-Noise (SNR)

SNR is the measured RMS signal to RMS noise at a specified input and sampling frequency. The noise is the RMS sum of all of the spectral components except the fundamental and the first five harmonics.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal-to-Noise + Distortion Ratio (SINAD)

SINAD is the measured RMS signal to RMS sum of all other spectral components below the Nyquist frequency excluding DC.

Effective Number of Bits (ENOB)

The effective number of bits (ENOB) is derived from the SINAD data. ENOB is calculated from:

$$ENOB = (SINAD - 1.76 + V_{CORR})/6.02,$$

where: $V_{CORR} = 0.5dB$.

Total Harmonic Distortion (THD)

THD is the ratio of the RMS sum of the first 5 harmonic components to the RMS value of the measured input signal.

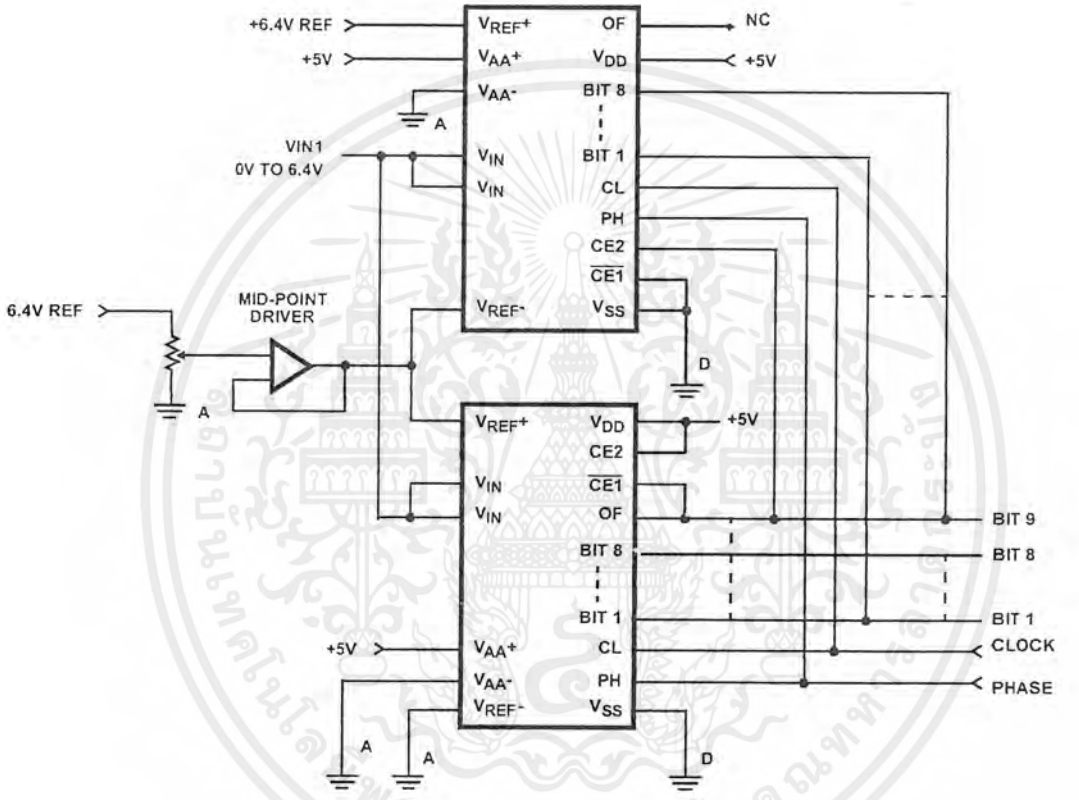


FIGURE 13. USING TWO CA3318s FOR 9-BIT RESOLUTION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

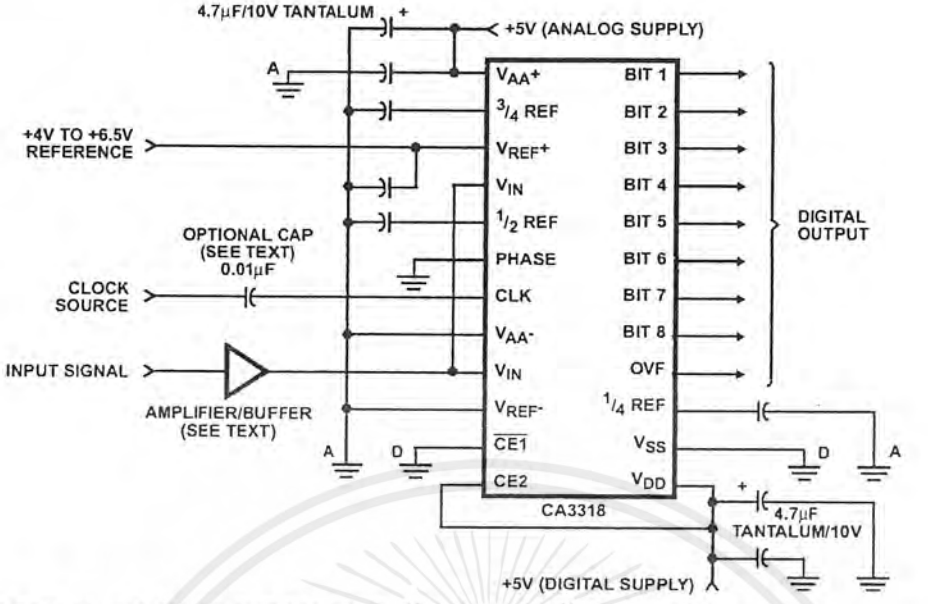


FIGURE 14. TYPICAL CIRCUIT CONFIGURATION FOR THE CA3318 WITH NO LINEARITY ADJUST

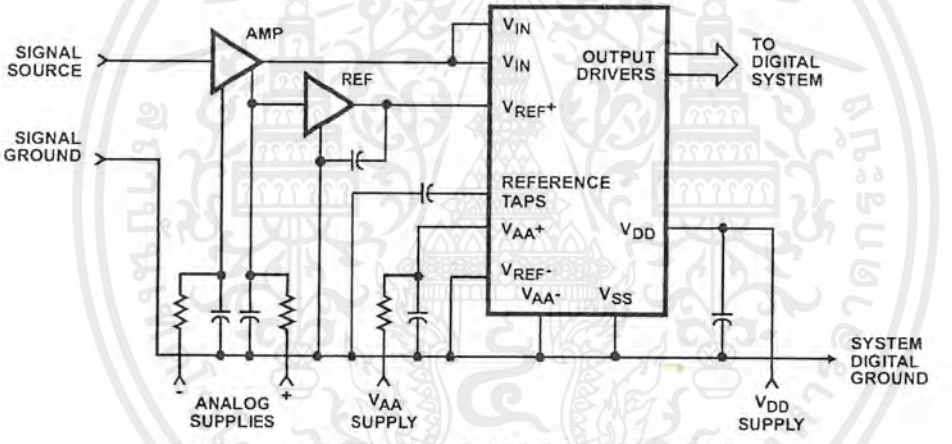
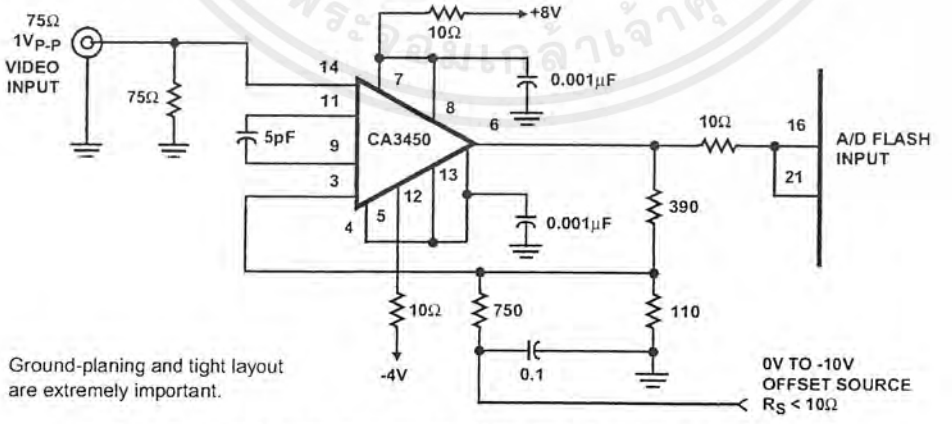


FIGURE 15. TYPICAL SYSTEM GROUNDING/BYPASSING



NOTE: Ground-planing and tight layout are extremely important.

FIGURE 16. TYPICAL HIGH BANDWIDTH AMPLIFIER FOR DRIVING THE CA3318

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1. OUTPUT CODE TABLE

CODE DESCRIPTION	(NOTE 1) INPUT VOLTAGE		BINARY OUTPUT CODE									DECIMAL COUNT
	V_{REF} 6.40V (V)	V_{REF} 5.12V (V)	OF	MSB B8	B7	B6	B5	B4	B3	B2	LSB B1	
Zero	0.00	0.00	0	0	0	0	0	0	0	0	0	0
1 LSB	0.025	0.02	0	0	0	0	0	0	0	0	1	1
2 LSB	0.05	0.04	0	0	0	0	0	0	0	1	0	2
•	•	•					•					•
•	•	•					•					•
•	•	•					•					•
$^1/_4$ Full Scale	1.60	1.28	0	0	1	0	0	0	0	0	0	64
•	•	•					•					•
•	•	•					•					•
•	•	•					•					•
$^1/_2$ Full Scale - 1 LSB	3.175	2.54	0	0	1	1	1	1	1	1	1	127
$^1/_2$ Full Scale	3.20	2.56	0	1	0	0	0	0	0	0	0	128
$^1/_2$ Full Scale + 1 LSB	3.225	2.58	0	1	0	0	0	0	0	0	1	129
•	•	•					•					•
•	•	•					•					•
•	•	•					•					•
$^3/_4$ Full Scale	4.80	3.84	0	1	1	0	0	0	0	0	0	192
•	•	•					•					•
•	•	•					•					•
•	•	•					•					•
Full Scale - 1 LSB	6.35	5.08	0	1	1	1	1	1	1	1	0	254
Full Scale	6.375	5.10	0	1	1	1	1	1	1	1	1	255
Over Flow	6.40	5.12	1	1	1	1	1	1	1	1	1	511

NOTE: 1. The voltages listed above are the ideal centers of each output code shown as a function of its associated reference voltage.

Reducing Power

Most power is consumed while in the auto-balance state. When operating at lower than 15MHz clock speed, power can be reduced by stretching the sample ($\phi 2$) time. The constraints are a minimum balance time ($\phi 1$) of 33ns, and a maximum sample time of 500ns. Longer sample times cause droop in the auto-balance capacitors. Power can also be reduced in the reference string by switching the reference on only during auto-balance.

Clock Input

The Clock and Phase inputs feed buffers referenced to $V_{AA}+$ and $V_{AA}-$. Phase should be tied to one of these two potentials, while the clock (if DC coupled) should be driven at least from 0.2 to 0.7 x ($V_{AA}+ - V_{AA}-$). The clock may also be AC coupled with at least a $1V_{P-P}$ swing. This allows TTL drive levels or 5V QMOS levels when $V_{AA}+$ is greater than 5V.

All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see web site **http://www.intersil.com**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- **Compatible with MCS-51™ Products**
- **4K Bytes of In-System Reprogrammable Flash Memory**
  - **Endurance: 1,000 Write/Erase Cycles**
- **Fully Static Operation: 0 Hz to 24 MHz**
- **Three-Level Program Memory Lock**
- **128 x 8-Bit Internal RAM**
- **32 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **Six Interrupt Sources**
- **Programmable Serial Channel**
- **Low Power Idle and Power Down Modes**

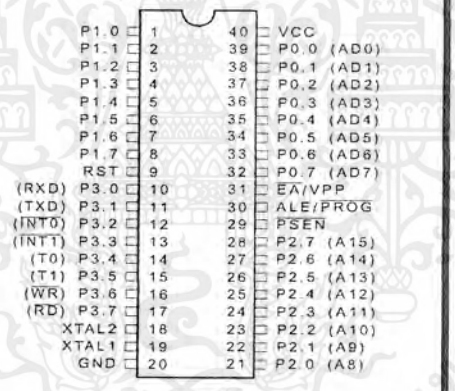
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

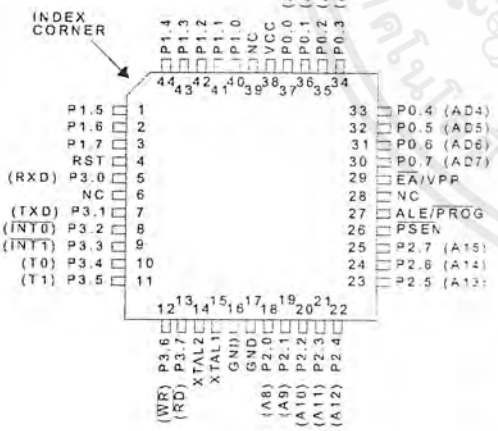
(continued)

Pin Configurations

PDIP



PQFP/TQFP



PLCC

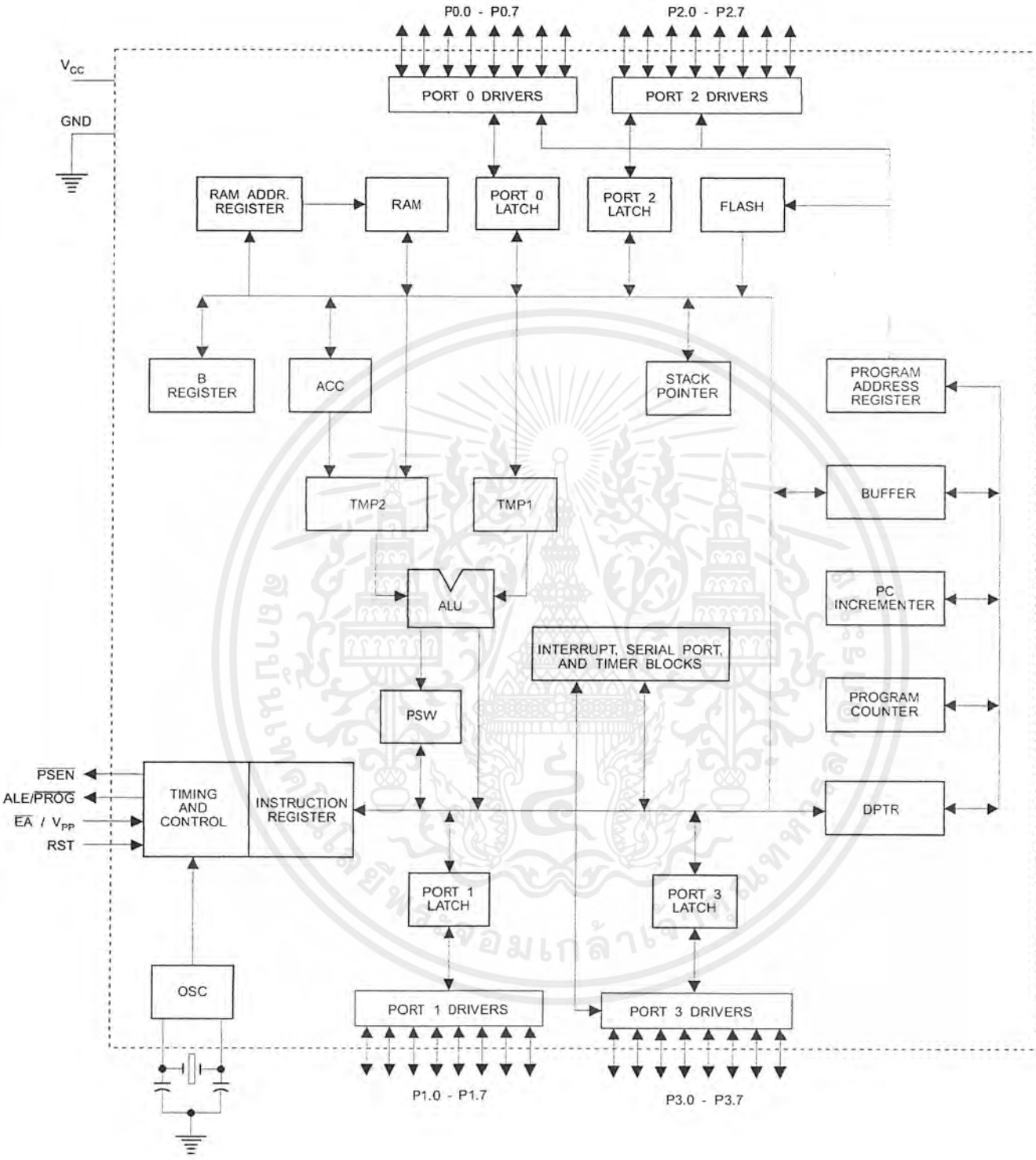
8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 0
Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1
Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2
Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3
Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$
Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$
Program Store Enable is the read strobe to external program memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When the AT89C51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/V_{PP}
External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier.

Oscillator Characteristics

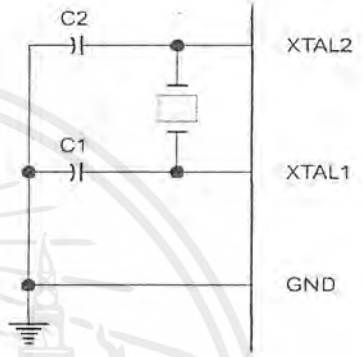
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

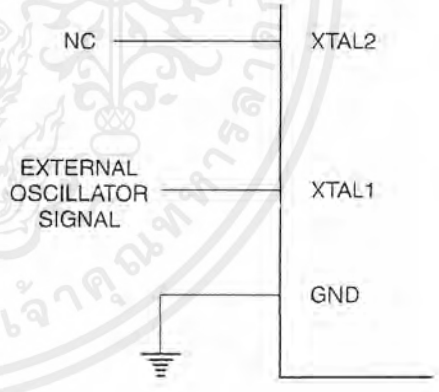
It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Programming the Flash

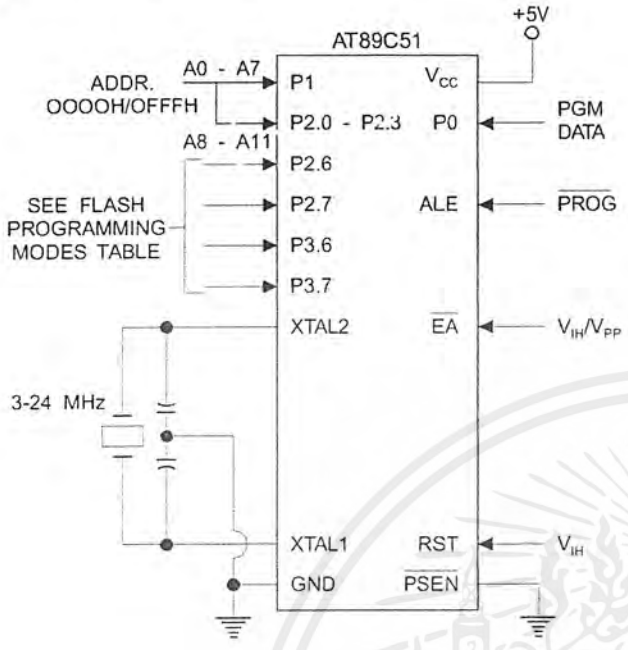
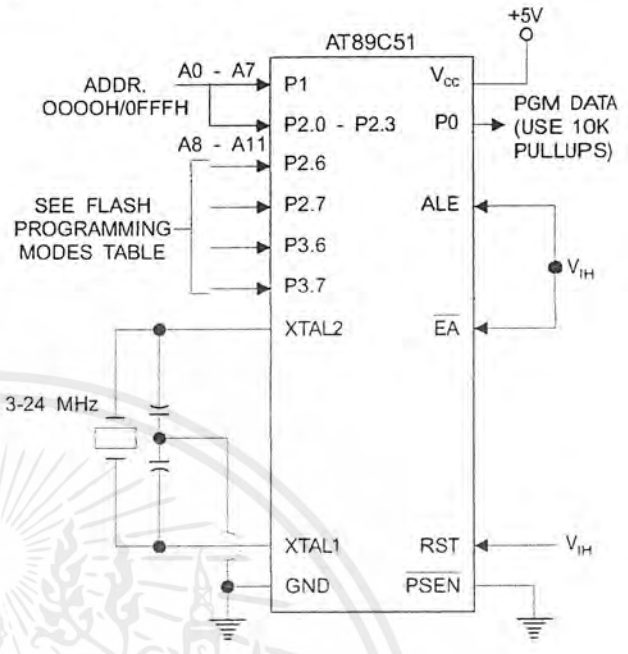


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

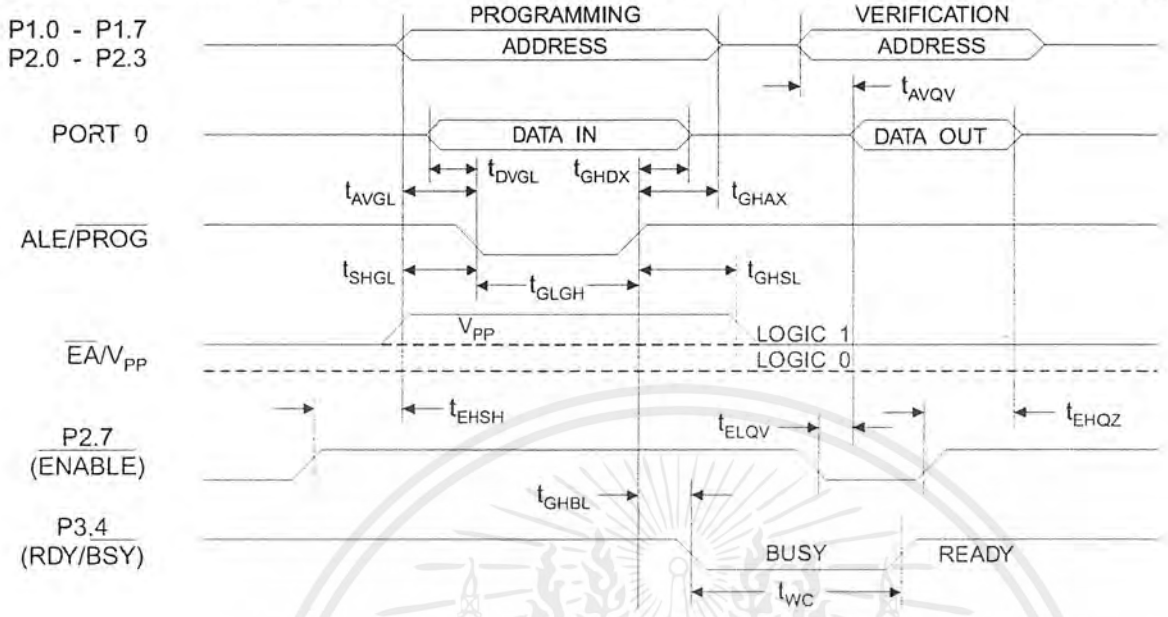
T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

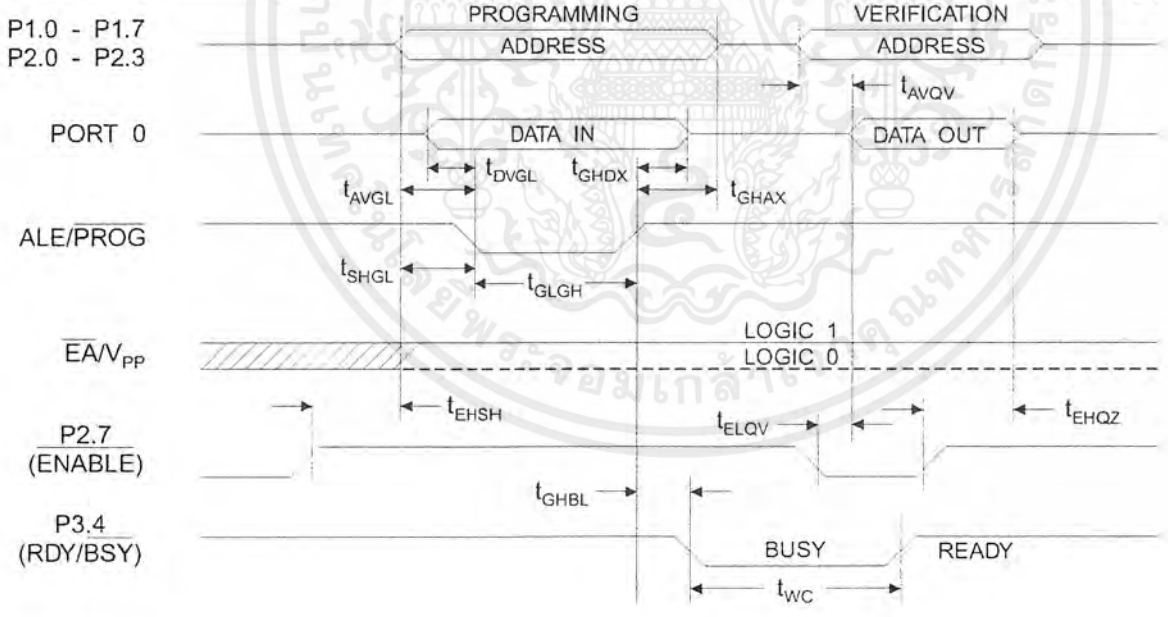
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low Voltage Mode (V_{PP} = 5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

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

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode[2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port: Port 0: 26 mA
Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

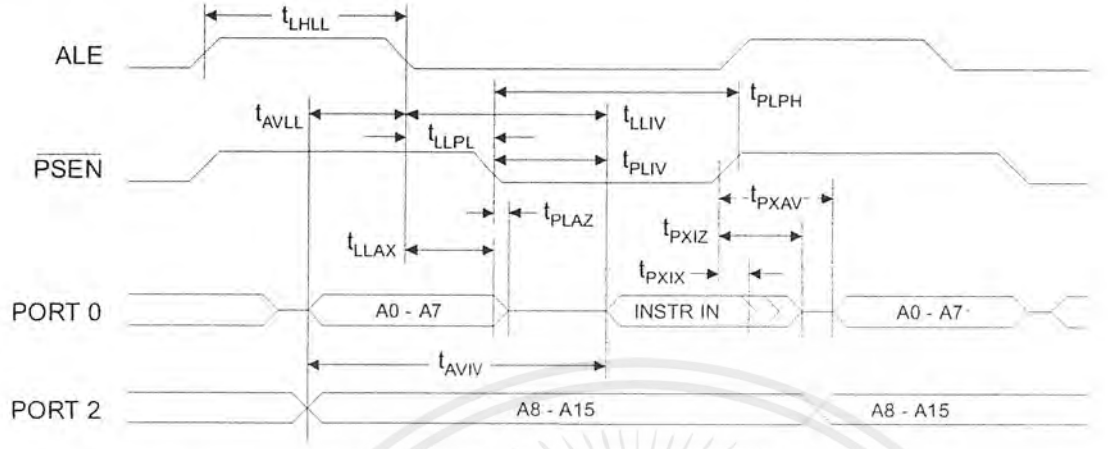
(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

External Program and Data Memory Characteristics

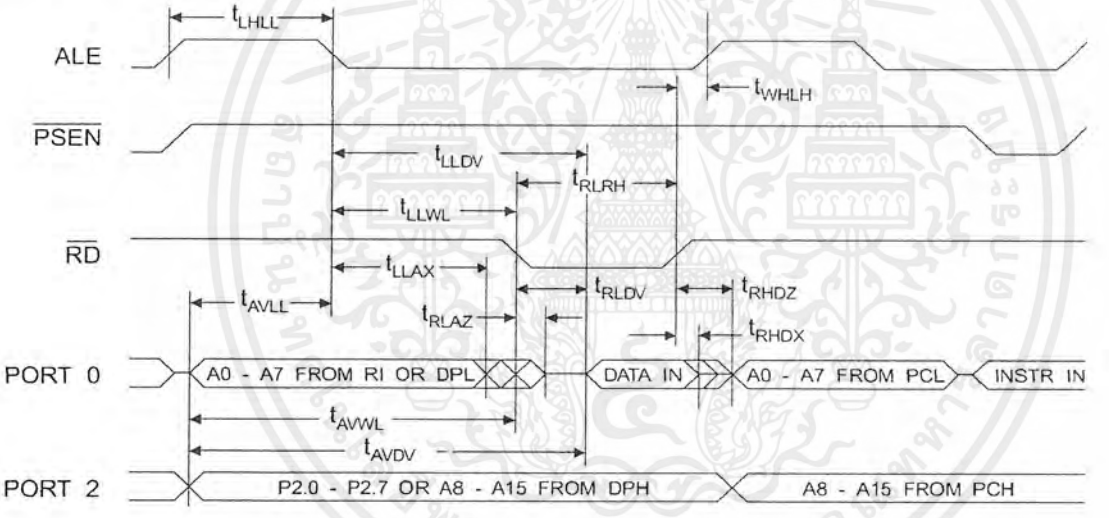
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL}-10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL}-20$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

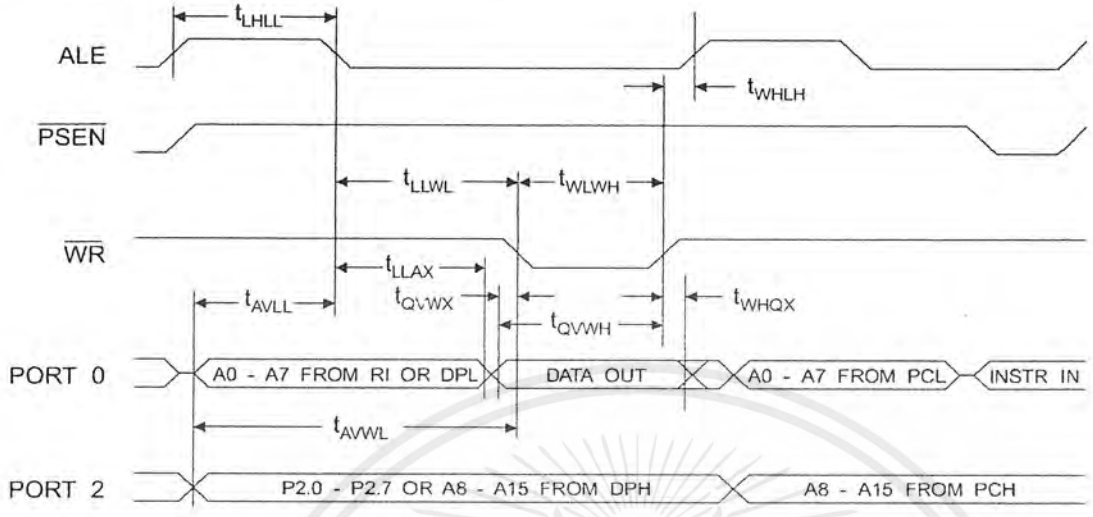


External Data Memory Read Cycle

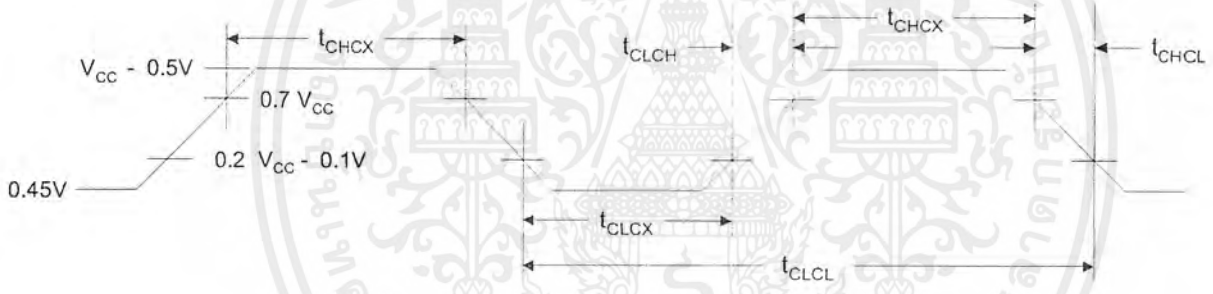


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

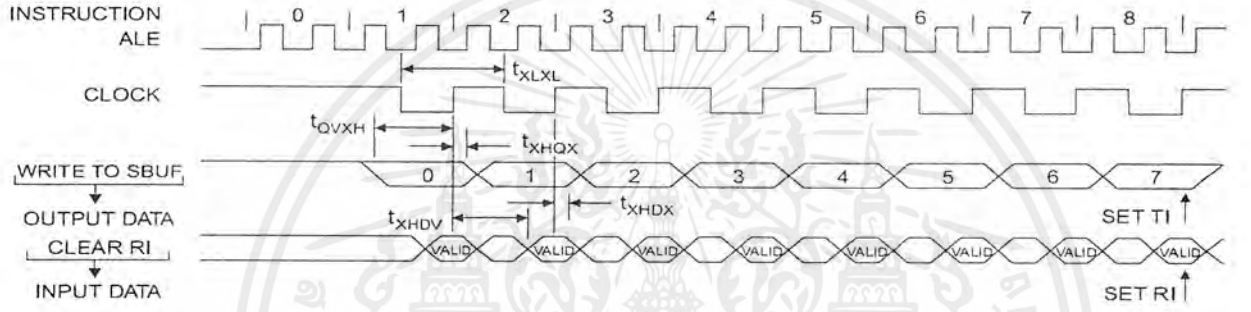
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

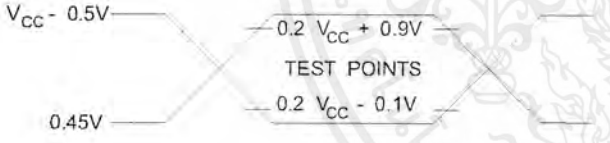
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

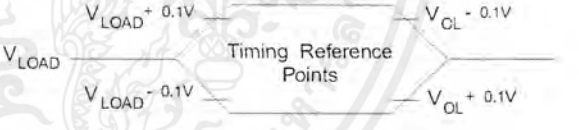


AC Testing Input/Output Waveforms(1)



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms(1)



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
		AT89C51-12AA	44A	Automotive
		AT89C51-12JA	44J	(-40°C to 105°C)
		AT89C51-12PA	40P6	
		AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
		AT89C51-16AA	44A	Automotive
		AT89C51-16JA	44J	(-40°C to 105°C)
		AT89C51-16PA	40P6	
		AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Document Title

128K x8 bit Low Power CMOS Static RAM

Revision History

Revision No.	History	Draft Data	Remark
0.0	Initial draft for commercial product - Commercial Product only	October 28, 1992	Preliminary
0.1	- Initial draft for Extended/Industrial Product - Datasheet for Extended/Industrial Product	September 1, 1993	Preliminary
1.0	Finalized - Commercial product finalized at 1993 - Extended and industrial product finalized at 1994	September 1, 1993 September 24, 1994	Final
2.0	Revised - Change datasheet format : one datasheet for commercial, extended industrial product	April 12, 1996	Final
3.0	Revised - Change datasheet format - Erase 100ns part from extended and industrial product - Erase Low power part from TSOP package	January 20, 1998	Final

The attached datasheets are provided by SAMSUNG Electronics. SAMSUNG Electronics CO., LTD. reserve the right to change the specifications and products. SAMSUNG Electronics will answer to your questions about device. If you have any questions, please contact the SAMSUNG branch offices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KM681000B Family CMOS SRAM

128K x8 bit Low Power CMOS Static RAM

FEATURES

- Process Technology : Poly Load
- Organization : 128Kx8
- Power Supply Voltage : 4.5~5.5V
- Low Data Retention Voltage : 2V(Min)
- Three state output and TTL Compatible
- Package Type : 32-DIP-600, 32-SOP-525
  32-TSOP1-0820F/R

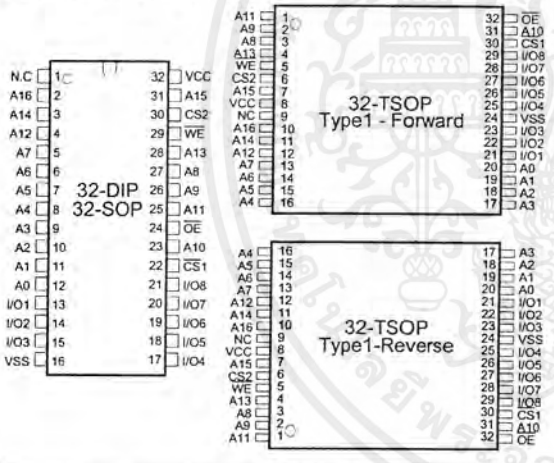
GENERAL DESCRIPTION

The KM681000B families are fabricated by SAMSUNG's advanced CMOS process technology. The families support various operating temperature ranges and have various package types for user flexibility of system design. The families also support low data retention voltage for battery back-up operation with low data retention current.

PRODUCT FAMILY

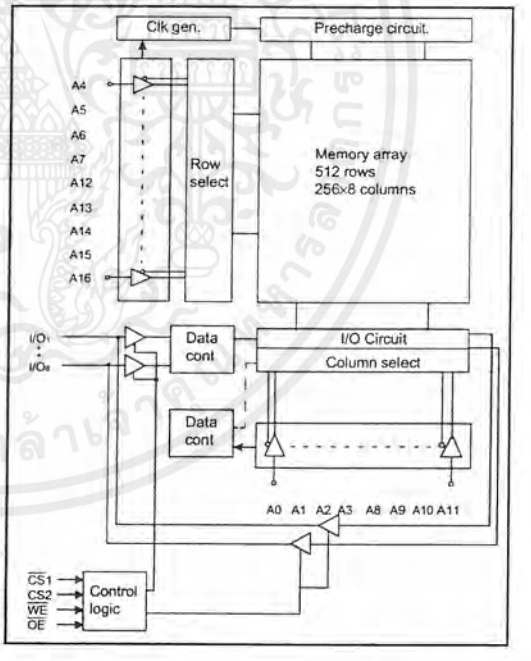
Product Family	Operating Temperature	Vcc Range	Speed(ns)	Power Dissipation		PKG Type
				Standby (I_{SB1}, Max)	Operating (I_{CC2}, Max)	
KM681000BL	Commercial(0~70°C)	4.5 to 5.5V	55/70	100μA	70mA	32-DIP,32-SOP 32-TSOP1 R/F
KM681000BL-L				20μA		
KM681000BLE	Extended(-25~85°C)		70	100μA		32-SOP 32-TSOP1 R/F
KM681000BLE-L				50μA		
KM681000BLI	Industrial(-40~85°C)		70	100μA		32-SOP 32-TSOP1 R/F
KM681000BLI-L				50μA		

PIN DESCRIPTION



Name	Function
$\overline{CS_1}$,CS_2	Chip Select Inputs
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
A_0~A_{16}	Address Inputs
I/O_1~I/O_8	Data Inputs/Outputs
Vcc	Power
Vss	Ground
N.C	No Connection

FUNCTIONAL BLOCK DIAGRAM



SAMSUNG ELECTRONICS CO., LTD. reserves the right to change products and specifications without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT LIST

Commercial Temperature Products (0~70°C)		Extended Temperature Products (-25~85°C)		Industrial Temperature Products (-40~85°C)	
Part Name	**Function**	**Part Name**	**Function**	**Part Name**	**Function**
KM681000BLP-5	32-DIP,55ns,L-pwr	KM681000BLGE-7	32-SOP,70ns,L-pwr	KM681000BLGI-7L	32-SOP,70ns,LL-pwr
KM681000BLP-5L	32-DIP,55ns,LL-pwr	KM681000BLGE-7L	32-SOP,70ns,LL-pwr	KM681000BLTI-7L	32-TSOP F,70ns,LL-pwr
KM681000BLP-7	32-DIP,70ns,L-pwr	KM681000BLTE-7L	32-TSOP F,70ns,LL-pwr	KM681000BLRI-7L	32-TSOP R,70ns,LL-pwr
KM681000BLP-7L	32-DIP,70ns,LL-pwr	KM681000BLRE-7L	32-TSOP R,70ns,LL-pwr		
KM681000BLG-5	32-SOP,55ns,L-pwr				
KM681000BLG-5L	32-SOP,55ns,LL-pwr				
KM681000BLG-7	32-SOP,70ns,L-pwr				
KM681000BLG-7L	32-SOP,70ns,LL-pwr				
KM681000BLT-5L	32-TSOP F,55ns,LL-pwr				
KM681000BLT-7L	32-TSOP F,70ns,LL-pwr				
KM681000BLR-5L	32-TSOP R,55ns,LL-pwr				
KM681000BLR-7L	32-TSOP R,70ns,LL-pwr				

Note : LL means Low Low standby current.

FUNCTIONAL DESCRIPTION

$\overline{CS_1}$	CS_2	$\overline{OE}$	$\overline{WE}$	I/O Pin	Mode	Power
H	X[1]	X[1]	X[1]	High-Z	Deselected	Standby
X[1]	L	X[1]	X[1]	High-Z	Deselected	Standby
L	H	H	H	High-Z	Output Disabled	Active
L	H	L	H	Dout	Read	Active
L	H	X[1]	L	Din	Write	Active

1. X means don't care.(Must be low or high state.)

ABSOLUTE MAXIMUM RATINGS[1]

Item	Symbol	Ratings	Unit	Remark
Voltage on any pin relative to Vss	V_{IN},V_{OUT}	-0.5 to 7.0	V	-
Voltage on Vcc supply relative to Vss	V_{CC}	-0.5 to 7.0	V	-
Power Dissipation	P_D	1.0	W	-
Storage temperature	T_{STG}	-65 to 150	°C	-
Operating Temperature	T_A	0 to 70	°C	KM681000BL
		-25 to 85	°C	KM681000BLE
		-40 to 85	°C	KM681000BLI
Soldering temperature and time	T_{SOLDER}	260°C, 10sec (Lead Only)	-	-

1. Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation should be restricted to recommended operating condition. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECOMMENDED DC OPERATING CONDITIONS[1)]

Item	Symbol	Min	Typ	Max	Unit
Supply voltage	Vcc	4.5	5.0	5.5	V
Ground	Vss	0	0	0	V
Input high voltage	V_{IH}	2.2	-	Vcc+0.5[2)]	V
Input low voltage	V_{IL}	-0.5[3)]	-	0.8	V

Note
1. Commercial Product : T_A=0 to 70°C, unless otherwise specified
   Extended Product : T_A=-25 to 85°C, unless otherwise specified
   Industrial Product : T_A=-40 to 85°C, unless otherwise specified
2. Overshoot : Vcc+3.0V in case of pulse width≤30ns
3. Undershoot : -3.0V in case of pulse width≤30ns
4. Overshoot and undershoot are sampled, not 100% tested

CAPACITANCE[1)] (f=1MHz, T_A=25°C)

Item	Symbol	Test Condition	Min	Max	Unit
Input capacitance	C_{IN}	V_{IN}=0V	-	6	pF
Input/Output capacitance	C_{IO}	V_{IO}=0V	-	8	pF

1. Capacitance is sampled, not 100% tested

DC AND OPERATING CHARACTERISTICS

Item	Symbol	Test Conditions		Min	Typ	Max	Unit
Input leakage current	I_{LI}	V_{IN}=Vss to Vcc		-1	-	1	μA
Output leakage current	I_{LO}	$\overline{CS_1}$=V_{IH} or CS_2=V_{IL} or $\overline{OE}$=V_{IH} or $\overline{WE}$=V_{IL}, V_{IO}=Vss to Vcc		-1	-	1	μA
Operating power supply	I_{CC}	$\overline{CS_1}$=V_{IL}, CS_2=V_{IH}, I_{IO}=0mA, V_{IN}= V_{IL} or V_{IH}		-	7	15[1)]	mA
Average operating current	I_{CC1}	Cycle time=1μs, 100% duty, I_{IO}=0mA, $\overline{CS_1}$≤0.2V, CS_2≥Vcc-0.2V, V_{IN}≤0.2V or V_{IN}≥Vcc-0.2V		-	-	10[2)]	mA
	I_{CC2}	Cycle time=Min, 100% duty, I_{IO}=0mA, $\overline{CS_1}$=V_{IL}, CS_2=V_{IH}, V_{IN}=V_{IL} or V_{IH}		-	-	70	mA
Output low voltage	V_{OL}	I_{OL}=2.1mA		-	-	0.4	V
Output high voltage	V_{OH}	I_{OH}=-1.0mA		2.4	-	-	V
Standby Current(TTL)	I_{SB}	$\overline{CS_1}$=V_{IH}, CS_2=V_{IL}, Other input=V_{IL} or V_{IH}		-	-	3	mA
Standby Current(CMOS)	I_{SB1}	$\overline{CS_1}$≥Vcc-0.2V, CS_2≥Vcc-0.2V or CS_2≤0.2V Other input=0~Vcc	KM681000BL	-	-	100	μA
			KM681000BL-L	-	-	20	
			KM681000BLE	-	-	100	μA
			KM681000BLE-L	-	-	50	
			KM681000BLI	-	-	100	μA
			KM681000BLI-L	-	-	50	

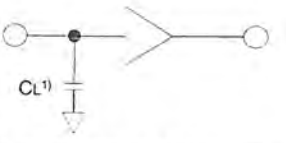
1. 20mA for Exteneded and Industrial Products
2. 15mA for Extended and Industrial Products

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC OPERATING CONDITIONS

TEST CONDITIONS (Test Load and Test Input/Output Reference)

Input pulse level : 0.8 to 2.4V
Input rising and falling time : 5ns
Input and output reference voltage : 1.5V
Output load (See right) : C_L=100pF+1TTL



1. Including scope and jig capacitance

AC CHARACTERISTICS (V_{CC}=4.5~5.5V, KM681000B Family : T_A=0 to 70°C, KM681000BE Family : T_A=-25 to 85°C, KM681000BI Family : T_A=-40 to 85°C)

Parameter List		Symbol	Speed Bins				Units
			55ns		70ns		
			Min	Max	Min	Max	
Read	Read cycle time	t_{RC}	55	-	70	-	ns
	Address access time	t_{AA}	-	55	-	70	ns
	Chip select to output	t_{CO1},t_{CO2},	-	55	-	70	ns
	Output enable to valid output	t_{OE}	-	25	-	35	ns
	Chip select to low-Z output	t_{LZ}	10	-	10	-	ns
	Output enable to low-Z output	t_{OLZ}	5	-	5	-	ns
	Chip disable to high-Z output	t_{HZ}	0	20	0	25	ns
	Output disable to high-Z output	t_{OHZ}	0	20	0	25	ns
	Output hold from address change	t_{OH}	10	-	10	-	ns
Write	Write cycle time	t_{WC}	55	-	70	-	ns
	Chip select to end of write	t_{CW}	45	-	60	-	ns
	Address set-up time	t_{AS}	0	-	0	-	ns
	Address valid to end of write	t_{AW}	45	-	60	-	ns
	Write pulse width	t_{WP}	40	-	50	-	ns
	Write recovery time	t_{WR}	0	-	0	-	ns
	Write to output high-Z	t_{WHZ}	0	20	0	25	ns
	Data to write time overlap	t_{DW}	25	-	30	-	ns
	Data hold from write time	t_{DH}	0	-	0	-	ns
	End write to output low-Z	t_{OW}	5	-	5	-	ns

DATA RETENTION CHARACTERISTICS

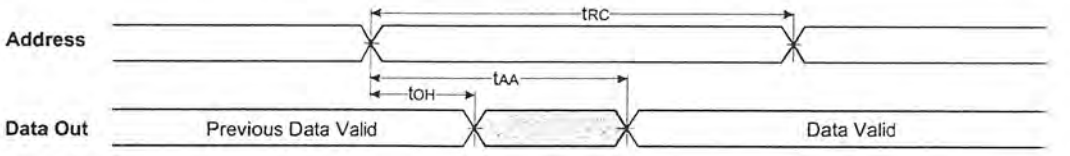
Item	Symbol	Test Condition		Min	Typ	Max	Unit
Vcc for data retention	V_{DR}	$\overline{CS_1}$[1]≥Vcc-0.2V		2.0	-	5.5	V
Data retention current	I_{DR}	Vcc=3.0V $\overline{CS_1}$≥Vcc-0.2V, CS_2≥Vcc-0.2V or CS_2≤0.2V Other Input+0~Vcc	KM681000BL	-	1	50	µA
			KM681000BL-L	-	0.5	10	
			KM681000BLE	-	-	50	
			KM681000BLE-L	-	-	25	
			KM681000BLI	-	-	50	
			KM681000BLI-L	-	-	25	
Data retention set-up time	t_{RDR}	See data retention waveform		0	-	-	ms
Recovery time	t_{RDR}			5	-	-	

1. $\overline{CS_1}$≥Vcc-0.2V,CS_2≥Vcc-0.2V($\overline{CS_1}$ controlled) or CS_2≤0.2V(CS_2 controlled)

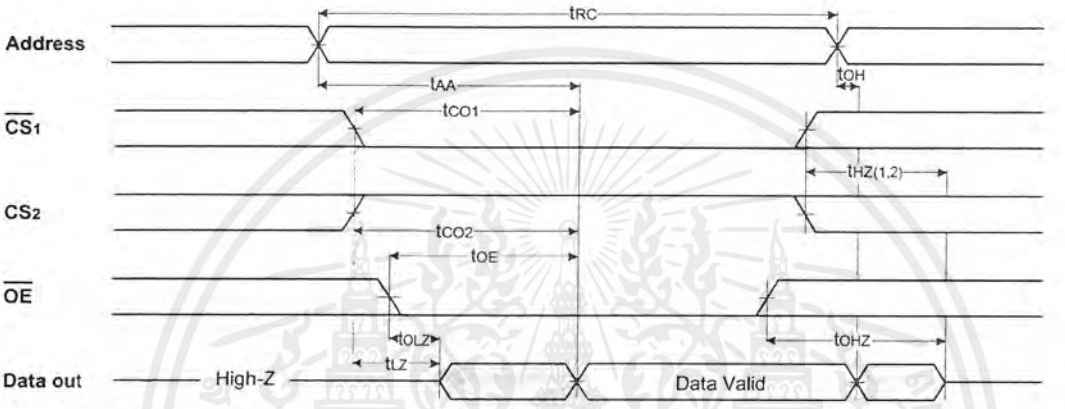
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMMING DIAGRAMS

TIMING WAVEFORM OF READ CYCLE(1) (Address Controlled, $\overline{CS}1=\overline{OE}=V_{IL}$, $\overline{WE}=V_{IH}$)



TIMING WAVEFORM OF READ CYCLE(2) ($\overline{WE}=V_{IH}$)

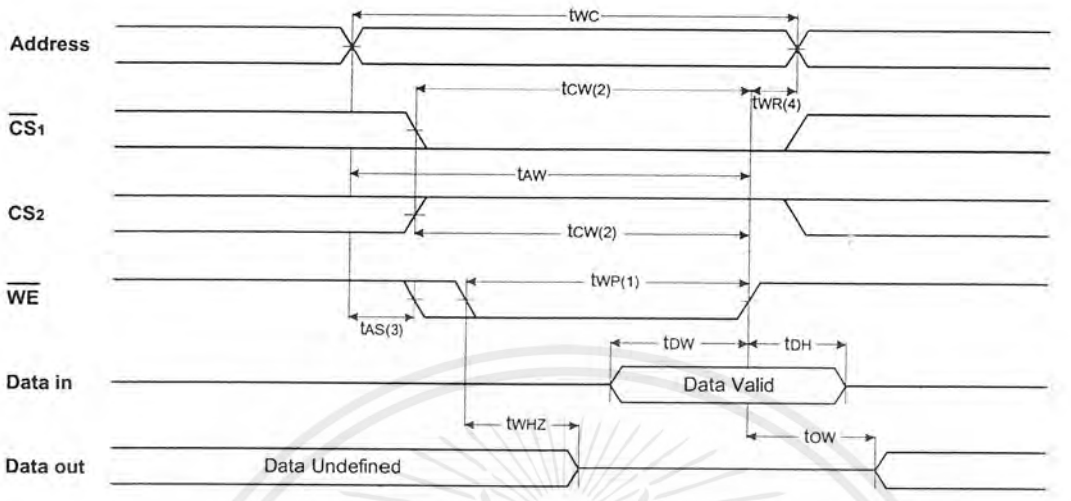


NOTES (READ CYCLE)

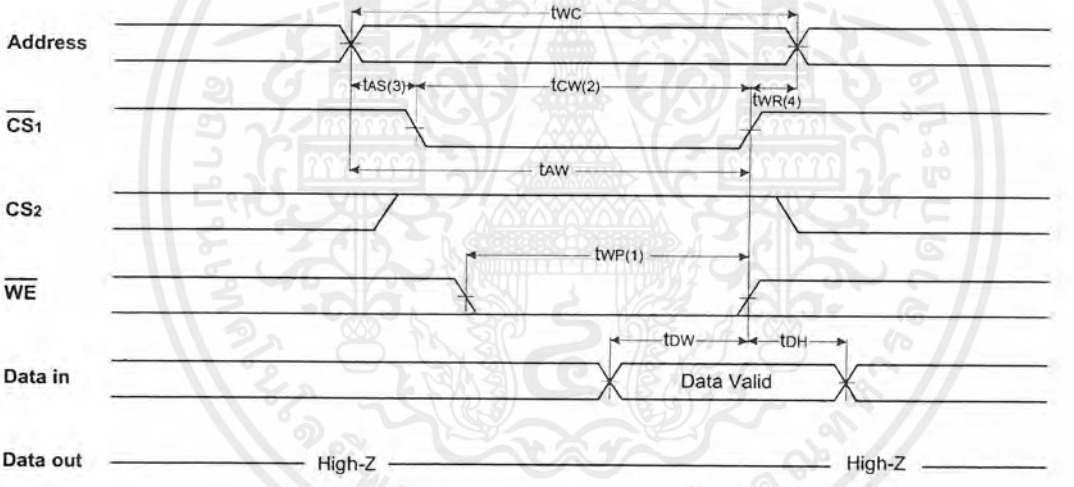
1. t_{HZ} and t_{OHZ} are defined as the time at which the outputs achieve the open circuit conditions and are not referenced to output voltage levels.
2. At any given temperature and voltage condition, t_{HZ}(Max.) is less than t_{LZ}(Min.) both for a given device and from device to device interconnection.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING WAVEFORM OF WRITE CYCLE(1) ($\overline{WE}$ Controlled)

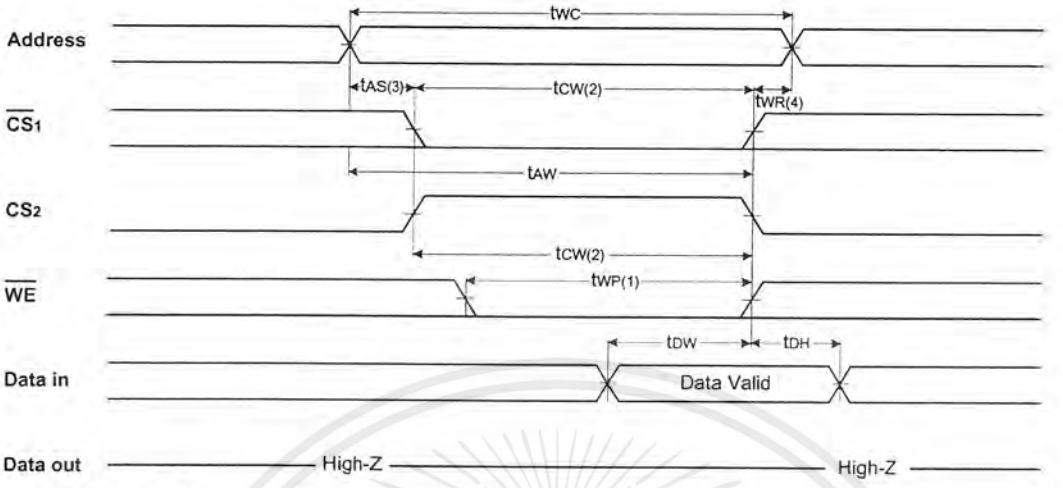


TIMING WAVEFORM OF WRITE CYCLE(2) ($\overline{CS_1}$ Controlled)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING WAVEFORM OF WRITE CYCLE(3) (CS_2 Controlled)

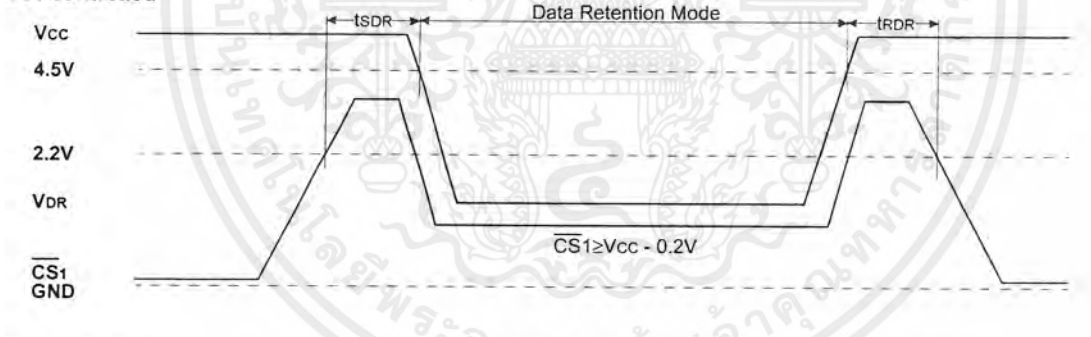


NOTES (WRITE CYCLE)

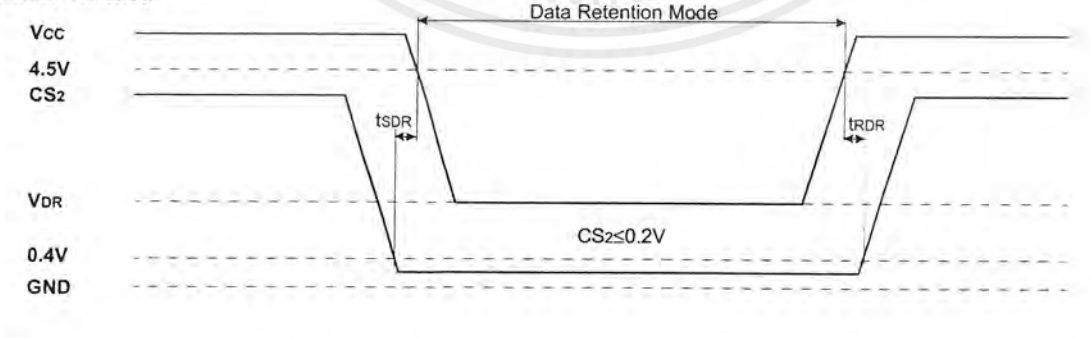
1. A write occurs during the overlap of a low $\overline{CS_1}$, a high CS_2 and a low $\overline{WE}$. A write begins at the latest transition among $\overline{CS_1}$ goes low, CS_2 going high and $\overline{WE}$ going low : A write end at the earliest transition among $\overline{CS_1}$ going high, CS_2 going low and $\overline{WE}$ going high, t_{WP} is measured from the beginning of write to the end of write.
2. t_{CW} is measured from the $\overline{CS_1}$ going low or CS_2 going high to the end of write.
3. t_{AS} is measured from the address valid to the beginning of write.
4. t_{WR} is measured from the end of write to the address change. $t_{WR(1)}$ applied in case a write ends as $\overline{CS_1}$ or $\overline{WE}$ going high $t_{WR(2)}$ applied in case a write ends as CS_2 going to low.

DATA RETENTION WAVE FORM

$\overline{CS_1}$ controlled



CS_2 controlled

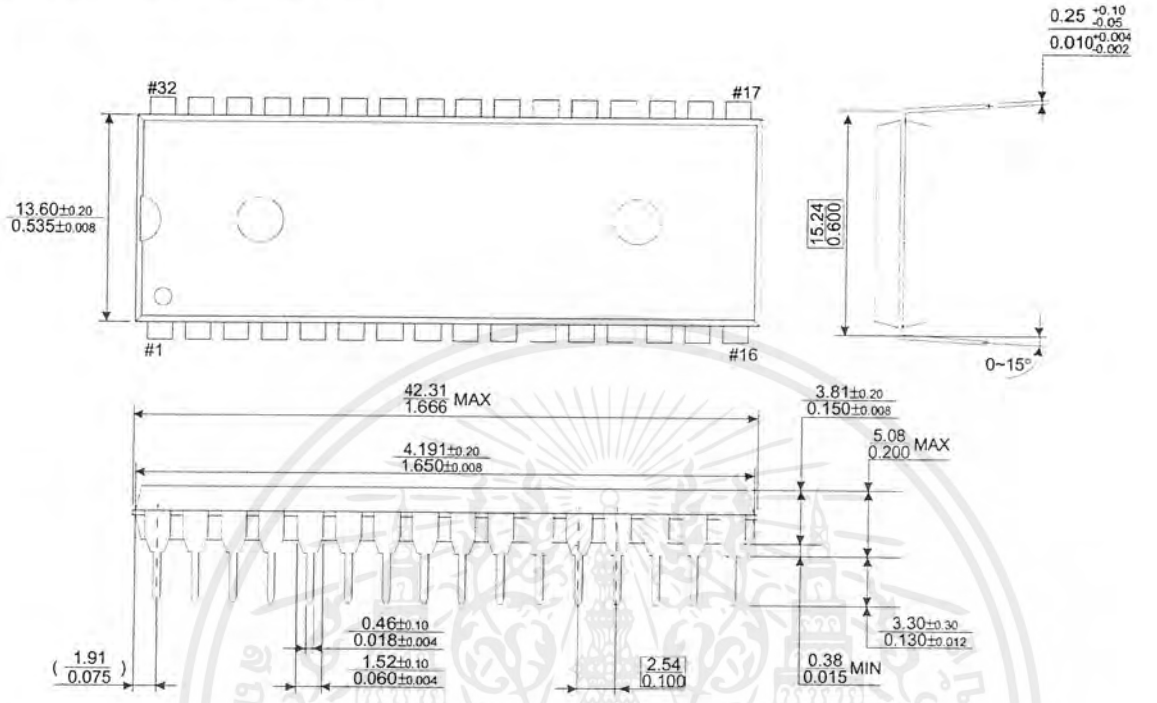


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

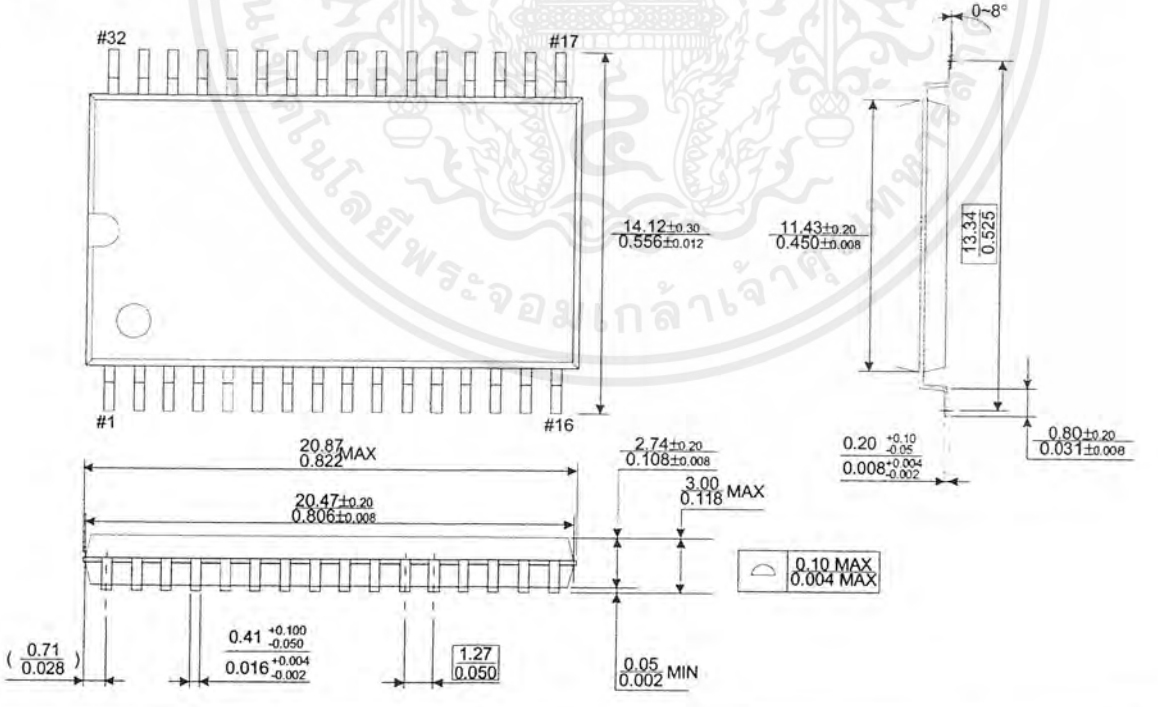
PACKAGE DIMENSIONS

Units : millimeter(inch)

32 DUAL INLINE PACKAGE (600mil)



32 PLASTIC SMALL OUTLINE PACKAGE (525mil)

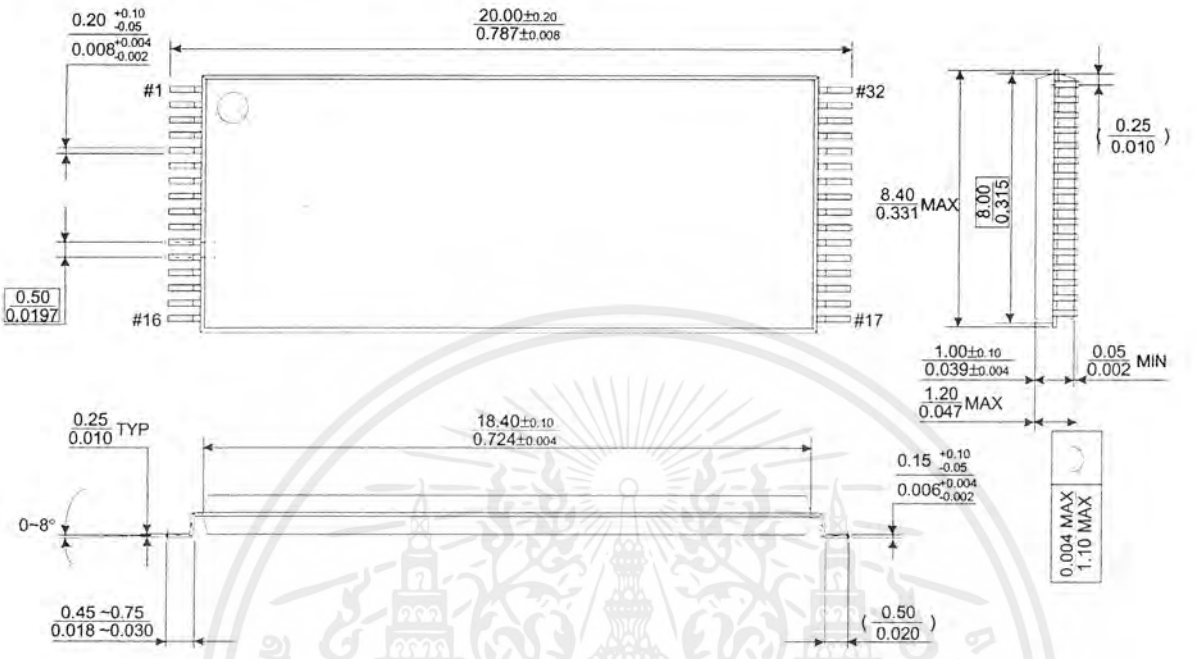


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

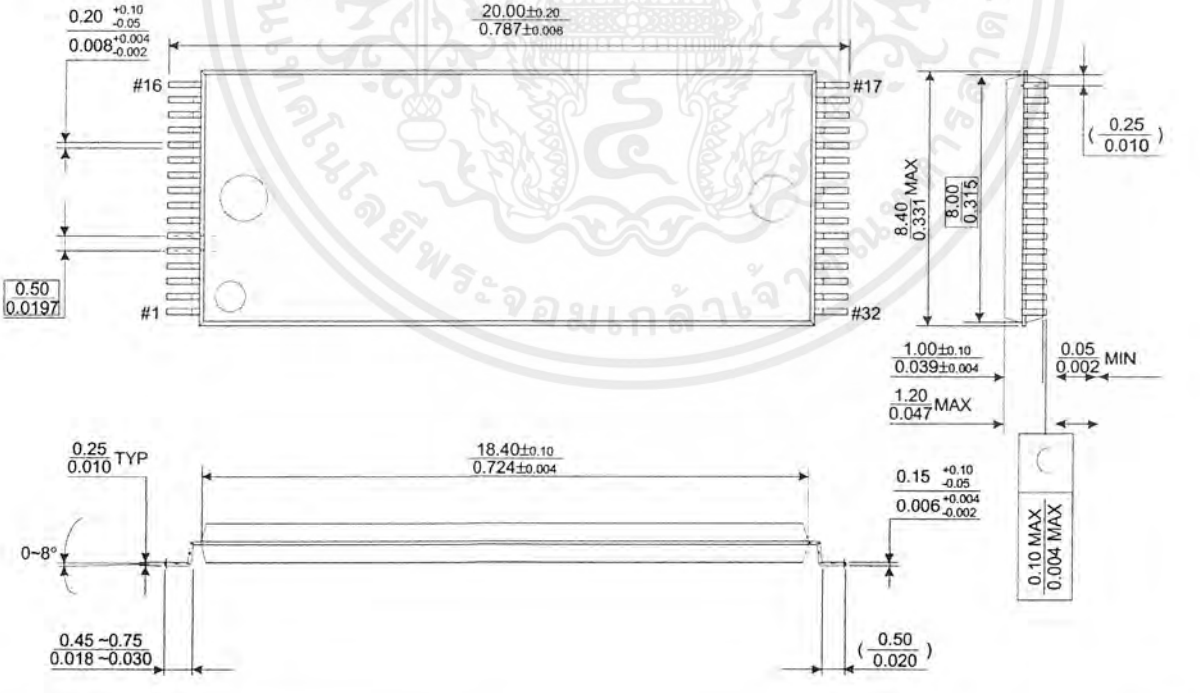
PACKAGE DIMENSIONS

Units : millimeter(inch)

32 THIN SMALL OUTLINE PACKAGE TYPE I (0820F)



32 THIN SMALL OUTLINE PACKAGE TYPE I (0820R)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1) รศ.สมยศ จุณณะปิยะ , "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51" , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2541

2) เจน สงสมพันธุ์ และ นิคม อนันต์ทิพย์,"เทคโนโลยีโทรทัศน์",สำนักพิมพ์สถาบันอิเล็กทรอนิกส์ กรุงเทพฯ , 2534

3) สุนทร วิทูสุรพจน์ , "การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051" , บริษัทซีเอ็ดยูเคชั่น จำกัด , 2537

4) เศกสิทธิ์ คำชมพู , "อุปกรณ์น่าใช้ไอซีน่าสนใจ" , วารสารเซมิคอนดักเตอร์ อิเล็กทรอนิกส์ ฉบับที่ 152 , 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้