สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์

TELEMONITOR SYSTEM VIA TELEPHONE LINE

T046475

โดย

นายสัญญา สายเจริญ

นายอนุวัฒน์ นุ่มวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปน.
ส3544
854

เลขหมู่.....
เลขทะเบียน 46475
วัน, เดือน, ปี - 2 เม.ย. 2546

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์

TELEMONITOR SYSTEM VIA TELEPHONE LINE

โดย

นายสัญญา สายเจริญ 42015071

นายอนุวัฒน์ นุ่มวงศ์ 42015078

อาจารย์ที่ปรึกษา

ผศ.ดร. สุทธิชัย นพนาคีพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์**

TELEMONITOR SYSTEM VIA TELEPHONE LINE

ผู้จัดทำ

1. นายสัญญา สายเจริญ 42015071

2. นายอนุวัฒน์ นุ่มวงศ์ 42015078

อาจารย์ที่ปรึกษา

(ผศ.ดร.สุทธิชัย นพนาคีพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์

TELEMONITOR SYSTEM VIA TELEPHONE LINE

โดย นายสัญญา สายเจริญ 42015071
นายอนุวัฒน์ นุ่มวงศ์ 42015078

อาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาคีพงษ์

บทคัดย่อ

ในปัจจุบันการสื่อสารข้อมูลผ่านเครือข่ายสายโทรศัพท์ มีการพัฒนามากขึ้นโดยสามารถส่งข้อมูลข่าวสารได้หลายรูปแบบรวมไปถึงสัญญาณภาพด้วย โครงงานนี้เสนอวิธีการที่สามารถแสดงสถานะแวดล้อมของอีกฝั่ง อาจทำได้กับทุกสถานที่และ/หรือทุกเวลา โดยทำการจับภาพแล้วแปลงสัญญาณเข้าไปยังโครงข่ายโทรศัพท์ด้วยการส่งภาพเป็นเฟรมอย่างต่อเนื่อง

ABSTRACT

Nowadays communication via telephone network has been improving to transmit various information formats including video signal. This project present the method that can monitoring faraway conditional situation may in anywhere and/or anytime by capturing a picture then transform to the telephone network with video frame sequence.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการทำงาน	3
2.1 ทฤษฎีโทรทัศน์	3
2.2 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	12
2.3 พอร์ตขนาน(Parallel Port)	15
2.4 การออกแบบส่วนโปรแกรมควบคุม	20
2.5 โปรแกรม Remote Anything	21
บทที่ 3 การคำนวณและการสร้าง	34
3.1 แนวคิดและหลักการทำงาน	34
3.2 หลักการทำงาน	34
3.3 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	35
3.4 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ	38
3.5 ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน	40
3.6 ภาคกำเนิดสัญญาณตอบรับ	42
3.7 ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน	42
3.8 ภาคเลือกเขียนหรืออ่านหน่วยความจำ	44
3.9 ภาคหน่วยความจำ	48
3.10 ภาคอินเตอร์เฟส	48
3.11 โปรแกรม	50
บทที่ 4 การทดลองและผลการทดลอง	52
4.1 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	52
4.2 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ	55
4.3 ภาคชี้แอดเดรสหน่วยความจำ	60
4.4 ภาคสร้างสัญญาณตอบรับ	64
4.5 การทดสอบระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์	64
บทที่ 5 บทวิจารณ์และบทสรุป	68
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 1.1 แสดง Block diagram การทำงานของระบบ	1
รูปที่ 2.1 องค์ประกอบภาพที่ยุคปัจจุบันมีการเอาแต่พิกเซลไปใช้งานกันแล้ว	4
รูปที่ 2.2 แสดงวิธีการเบื้องต้นของการสแกน	4
รูปที่ 2.3 รายละเอียดการสแกนแบบสลับเส้นหรือแบบแทรกสอด	5
รูปที่ 2.4 การสแกนสลับเส้นหรือแบบแทรกสอดในระบบ CCIR	6
รูปที่ 2.5 แสดงระดับสัญญาณ ขาว-ดำ โดยเกรย์สเกล	9
รูปที่ 2.6 แสดงลักษณะสัญญาณภาพรวมเบื้องต้น	10
รูปที่ 2.7 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง	11
รูปที่ 2.8 แสดงระบบพัลส์ในฟิลด์ของการซิงโครไนซ์ของระบบโทรทัศน์	11
รูปที่ 2.9 แสดงการแซมปลิ้ง	12
รูปที่ 2.10 แสดงลักษณะสัญญาณของภาคควอนไตซิ่งและการเข้ารหัส	13
รูปที่ 2.11 แสดงวงจร Flash A/D	14
รูปที่ 2.12 แสดงส่วนประกอบของโปรแกรม Remote Anything	24
รูปที่ 2.13 แสดงปุ่มคำสั่งพื้นฐาน	25
รูปที่ 2.14 แสดงการตั้งค่าตัวเลือกให้กับโปรแกรม Remote Anything	26
รูปที่ 2.15 แสดงเมนูลัดของเครื่องคอมพิวเตอร์ต้นทางเลือกใช้คำสั่งต่างๆ กับเครื่องคอมพิวเตอร์ปลายทางกำลังเชื่อมต่อ	31
รูปที่ 2.16 แสดงรายละเอียด RA Options	32
รูปที่ 2.17 แสดงรายละเอียดของคำสั่ง Security Options	33
รูปที่ 3.1 แสดง Block Diagram การทำงานของระบบ	35
รูปที่ 3.2 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	37
รูปที่ 3.3 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ	39
รูปที่ 3.4 ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน	41
รูปที่ 3.5 วงจรกำเนิดสัญญาณตอบรับหรือสัญญาณ ACK	42
รูปที่ 3.6 ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน	43
รูปที่ 3.7 ภาคเลือกเขียนหรืออ่านหน่วยความจำ	45
รูปที่ 3.8 ภาคหน่วยความจำ	47
รูปที่ 3.9 แสดงบิตที่ใช้งานในพอร์ตข้อมูล	49
รูปที่ 3.10 แสดงบิตที่ใช้งานในพอร์ตสถานะ	49
รูปที่ 3.11 แสดงบิตที่ใช้งานในพอร์ตควบคุม	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.12 แสดงแผนภูมิการทำงานของโปรแกรมวิชวลซี	51
รูปที่ 4.1 สัญญาณภาพรวมขาว-ดำที่ออกจากล้องวีดีโอ	52
รูปที่ 4.2 สัญญาณอินพุตที่ขา 21 ของ U1 (ระดับแรงดันอ้างอิงบวก 5 โวลต์)	53
รูปที่ 4.3 สัญญาณอินพุตที่ขา 21 ของ U1 (ระดับแรงดันอ้างอิงลบ 4.2 โวลต์)	53
รูปที่ 4.4 สัญญาณนาฬิกาความถี่ 15 MHz	54
รูปที่ 4.5 สัญญาณภาพรวม(CH1) เทียบกับ สัญญาณ ID0 (CH2)	54
รูปที่ 4.6 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ ID7 (CH2)	55
รูปที่ 4.7 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ HOR SYNC (CH2)	56
รูปที่ 4.8 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ VER SYNC (CH2)	56
รูปที่ 4.9 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ ODD/EVEN (CH2)	57
รูปที่ 4.10 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ HOR SYNC (CH2)	57
รูปที่ 4.11 สัญญาณ HOR SYNC (CH1) เทียบกับ สัญญาณ ขา 5 ของ U7B (CH2)	58
รูปที่ 4.12 สัญญาณ HOR SYNC (CH1) เทียบกับ สัญญาณ H-COUNT (CH2)	58
รูปที่ 4.13 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ VER SYNC (CH2)	59
รูปที่ 4.14 สัญญาณ VER SYNC (CH1) เทียบกับ สัญญาณ ขา 5 ของ U8B (CH2)	59
รูปที่ 4.15 สัญญาณ VER SYNC (CH1) เทียบกับ สัญญาณ V-COUNT (CH2)	60
รูปที่ 4.16 สัญญาณนาฬิกาความถี่ 5 MHz	60
รูปที่ 4.17 สัญญาณ H-COUNT (CH1) เทียบกับ สัญญาณ CA0 (CH2)	61
รูปที่ 4.18 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ CA0 (CH2)	62
รูปที่ 4.19 สัญญาณ V-COUNT (CH1) เทียบกับ สัญญาณ CA8 (CH2)	62
รูปที่ 4.20 แสดงการเปรียบเทียบระหว่างสัญญาณภาพรวม (CH1) เทียบกับขา CA8 (CH2)	63
รูปที่ 4.21 แสดงสัญญาณที่ขา CA15 เทียบกับสัญญาณ ACK	64
รูปที่ 4.22 แสดงหน้าต่างหลักของซอฟต์แวร์ที่สร้างขึ้นสำหรับโครงงานนี้	65
รูปที่ 4.23 แสดงหน้าต่างแสดงรายละเอียดของซอฟต์แวร์	65
รูปที่ 4.24 แสดงโครงงานขณะกำลังจับภาพจากกล้องวงจรปิด	66
รูปที่ 4.25 แสดงภายในกล่องโครงงาน	66
รูปที่ 4.26 แสดงภาพด้านหน้าของกล่องโครงงาน	67
รูปที่ 4.27 แสดงภาพด้านหลังของกล่องโครงงาน	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

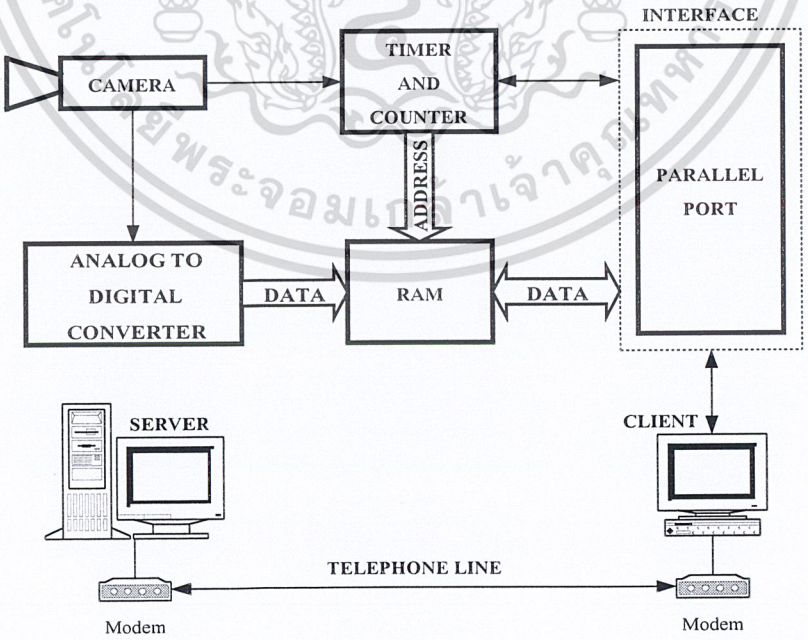
สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 แสดงรายละเอียดขาสัญญาณต่างของพอร์ตขนาน	16
ตารางที่ 2.2 แสดงสัญญาณและขาสัญญาณของพอร์ตขนาน IEEE1284	17
ตารางที่ 2.3 แสดงแอดเดรสฐานของพอร์ตขนาน	18
ตารางที่ 2.4 แสดงการชี้ตำแหน่งต่างๆ ของ EPP	19
ตารางที่ 2.5 แสดงรายละเอียดต่างๆ ของรีจิสเตอร์พอร์ตควบคุม	20
ตารางที่ 2.6 แสดงสถานะข้อมูล	21
ตารางที่ 2.7 แสดงพอร์ตใช้งาน	21
ตารางที่ 3.1 แสดงการเชื่อมต่อฮาร์ดแวร์กับขา DB-25	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ระบบตรวจสอบทางไกลผ่านทางคู่สายโทรศัพท์ เป็นการส่งข้อมูลข่าวสารที่เป็นสัญญาณภาพผ่านเครือข่ายโทรศัพท์ ซึ่งสัญญาณภาพที่ทำการส่งจะเป็นลักษณะการจับภาพทีละเฟรม เฟรมละหนึ่งฟิลด์ และเป็นภาพขาว-ดำ ซึ่งหลักการของระบบคือ จะนำสัญญาณที่ออกจากกล้องวีดีโอ (หรือกล้องวงจรปิด) ซึ่งเป็นสัญญาณอนาลอกที่ประกอบไปด้วยสัญญาณหลายๆ สัญญาณที่มีชื่อเรียกว่า "สัญญาณภาพรวม" (Composite signal) จากนั้นจะนำเอาสัญญาณภาพขาว-ดำ ที่อยู่ในสัญญาณภาพรวมไปทำการแปลงจากสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลขนาด 8 บิตต่อจุดภาพ โดยใช้ไอซีแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลความเร็วสูง (Flash A/D Converter) เบอร์ CA 3318 จากนั้นข้อมูลที่อยู่ในรูปสัญญาณดิจิตอลนี้จะถูกส่งเข้าไปเก็บในหน่วยความจำขนาด 64 กิโลไบต์ เบอร์ KM 681000 โดยการเก็บข้อมูลลงในหน่วยความจำจะต้องสัมพันธ์กับการสแกนเก็บข้อมูลภาพของกล้องวีดีโอ ซึ่งในการทำงานส่วนนี้สามารถควบคุมได้โดยการใช้สัญญาณฮอร์ซิงค์และสัญญาณเวอร์ซิงค์มาทำการสร้างสัญญาณควบคุมการชี้แอดเดรสของหน่วยความจำ ซึ่งสัญญาณซิงค์ทั้งสองนี้อยู่ในสัญญาณภาพรวม ซึ่งสามารถทำการแยกออกมาได้โดยใช้ไอซีแยกสัญญาณซิงค์ เบอร์ LM 1881 เมื่อข้อมูลถูกเก็บลงในหน่วยความจำครบ 1 เฟรม ฮาร์ดแวร์ก็จะส่งข้อมูลไปแจ้งคอมพิวเตอร์ไคลเอนท์ จากนั้นคอมพิวเตอร์ไคลเอนท์ ก็จะทำการส่งสัญญาณนาฬิกามาเพื่อทำการชี้แอดเดรสของหน่วยความจำ เพื่อทำการอ่านข้อมูลเข้ามาแสดงที่หน้าจอโดยผ่านพอร์ตขนาน ซึ่งกระบวนการทั้งหมดนี้จะเกิดขึ้นก็ต่อเมื่อมีการสั่งการจากเครื่องคอมพิวเตอร์เซิร์ฟเวอร์มายังเครื่องคอมพิวเตอร์ไคลเอนท์โดยการควบคุมผ่านโปรแกรม Remote Anything (RA)



รูปที่ 1.1 แสดง Block diagram การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตของงาน

ขอบเขตของงานที่ทำในโครงงานนี้จะอธิบายจาก Block diagram ของระบบจะประกอบไปด้วย 2 ส่วน คือ ฮาร์ดแวร์และซอฟต์แวร์

ฮาร์ดแวร์

1. ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล
2. ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ
3. ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน
4. ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน
5. ภาคเลือกเขียนหรืออ่านหน่วยความจำ
6. ภาคหน่วยความจำ
7. ภาคอินเตอร์เฟส

ซอฟต์แวร์

1. เขียนโปรแกรมควบคุมฮาร์ดแวร์ในส่วนคอมพิวเตอร์ไคลเอนท์ เพื่อทำการจับภาพจากกล้องวงจรปิดมาแสดงที่จอมอนิเตอร์

2. ติดตั้งโปรแกรม Remote Anything (RA) เพื่อใช้ในการติดต่อระหว่างเครื่องคอมพิวเตอร์เซิร์ฟเวอร์และเครื่องคอมพิวเตอร์ไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 ทฤษฎีโทรทัศน์

ภาพเกิดมาจากอะไร อันที่จริงแล้วภาพที่เกิดขึ้นที่จอโทรทัศน์เป็นภาพนิ่งที่เอามาต่อกันโดยแต่ละภาพจะมีความแตกต่างกันเล็กน้อย และเกิดขึ้นในเวลาอันรวดเร็วทำให้สายตาคนเห็นเป็นภาพที่ต่อเนื่องในแต่ละเฟรมโดยตัวของมันเองประกอบด้วยพื้นที่เล็กๆ ของแสงและจุด หากภาพนั้นมีรายละเอียดของแสงและจุดมากภาพที่ออกมาจะมีรายละเอียดมากกว่าภาพมีรายละเอียดของแสงและจุดน้อย (ซึ่งภาพออกมาหยาบ)

2.1.1 องค์ประกอบภาพ

หากตัดภาพจากหนังสือพิมพ์มาสักภาพหนึ่งแล้วขยายให้ใหญ่ขึ้นด้วยแว่นขยาย จะพบว่ามีองค์ประกอบภาพมาจากจุดสีขาวและสีดำมากมายมาเรียงกันประกอบขึ้นเป็นภาพ จุดเหล่านี้เรียกว่าองค์ประกอบภาพ หรือพิคเจอร์อิลีเมนต์ (Picture Element) หรือพิกเซล (Pixel) ทำนองเดียวกัน ภาพที่ปรากฏทางโทรทัศน์ก็เอามาจากหลักการนี้ ภาพที่เกิดขึ้นบนจอโทรทัศน์ประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนจำนวนมาก แต่ละเส้นนั้นจะมีทั้งส่วนที่ดำสนิท, ส่วนที่จาง และส่วนที่สว่างรวมกันอยู่ เส้นเหล่านี้ได้มาจากการกวาดลำแสง (Scan) ซึ่งความแตกต่างกันบนเส้นกวาดลำแสง หรือเส้นสแกนเหล่านี้เองที่จัดว่าเป็นองค์ประกอบภาพ ระบบการสแกน 525 เส้นอาจแบ่งส่วนทางแนวตั้งได้ 700 เส้น ดังนั้นจึงอาจกล่าวได้ว่าหากจะหาองค์ประกอบภาพในระบบสแกน 525 เส้นแล้วจะได้เท่ากับ 525×700 เท่ากับ 367,500 พิกเซล ยิ่งภาพมีจำนวนเส้นมากเท่าไรรายละเอียดของภาพก็ยิ่งมีมากขึ้นเท่านั้นเหมือนอย่างที่กล่าวถึงรายละเอียดภาพมาแต่ข้างต้น โทรทัศน์ที่มีเส้นสแกนมากย่อมมีรายละเอียดหรือความชัดเจนของภาพมากกว่าแต่การออกแบบวงจรจะยิ่งยากไปด้วยเพราะแบนด์วิดธ์ของความถี่จะต้องกว้างขึ้นด้วย 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 อัตราส่วนเอ็กสเปคต์ ซึ่งโดยทั่วไปจะใช้อัตรา 4 ต่อ 3

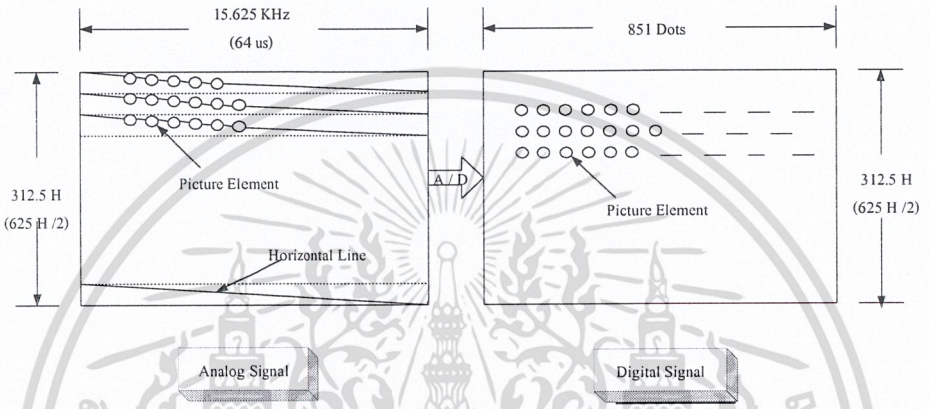
$\frac{Y}{k}$ คือ ค่าเอฟเฟ็คตีฟแฟคเตอร์ มีค่า 0.95/0.84

จากสูตรนี้จะพบว่าถ้าจำนวนเส้นสแกนเพิ่มมากขึ้นความถี่จะเพิ่มตามขึ้นมากด้วย จากที่กล่าวว่าระบบ 525 เส้น นั้นมีองค์ประกอบภาพ 367,500 พิกเซล แต่ในความเป็นจริงแล้วมองเห็นได้ไม่ครบทุกพิกเซล เนื่องจากมีบางส่วนหายไปในการสแกน จากการศึกษาพบว่าภาพที่พอดูได้จะมีองค์ประกอบไม่ต่ำกว่า 200,000 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรทัศน์ที่ใช้อยู่ในประเทศไทยปัจจุบันใช้ระบบการสแกน 625 เส้น กว้าง 7 เมกกะเฮิรตซ์ อย่างไรก็ตามองค์ประกอบภาพจะมีรายละเอียดมากขึ้น โดยสามารถหาองค์ประกอบภาพได้จากจำนวนการสแกน 625 เส้นคูณกับจำนวนจุดหรือองค์ประกอบแนวตั้ง 851 เส้น ได้เท่ากับ 531,875 พิกเซล

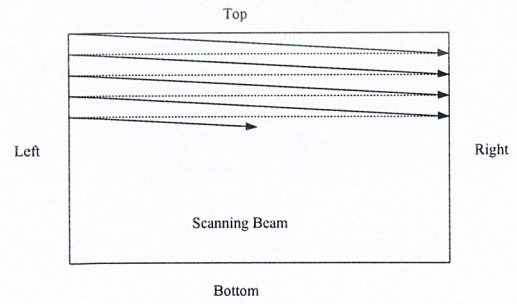
ปัจจุบันส่วนที่เรียกว่าองค์ประกอบภาพได้ถูกนำไปใช้งานอย่างเป็นจริงมากขึ้นในโทรทัศน์ หรือเครื่องเล่นวีดีโอคาสเซตเร็คคอร์ดเดอร์ จะมีการนำเอาพิกเซลเหล่านี้ไปเก็บไว้ในหน่วยความจำเพราะพิกเซลเท่านั้นที่ระบบดิจิตอลจะจัดเก็บข้อมูลได้จะพบวิธีการนี้ในโทรทัศน์ระบบดิจิตอล, โทรทัศน์ระบบคอมพิวเตอร์, โทรทัศน์จอภาพแอลซีดี, ดิจิตอล วีซีอาร์, โทรทัศน์หรือวีซีอาร์ ระบบภาพซ้อนภาพ ฯลฯ



รูปที่ 2.1 องค์ประกอบภาพที่ยุคปัจจุบันมีการเอาแต่พิกเซลไปใช้งานกันแล้ว

2.1.2 การสแกน

องค์ประกอบภาพที่ได้กล่าวมาแล้วในตอนต้น เครื่องส่งจะรับภาพแล้วเปลี่ยนพลังงานแสงเป็นพลังงานไฟฟ้าส่งเป็นจุดขาว-ดำมา เครื่องรับจะรับเอาสัญญาณเหล่านี้มาเรียงกันใหม่ โดยการวาดเป็นเส้นภาพขึ้นที่หน้าจอ วิธีนี้เรียกว่าการสแกน (Scan) โดยตัวที่มีหน้าที่สำคัญคือจอภาพหรือหลอดภาพนั้นเอง หลอดภาพมีโครงสร้างคล้ายกับหลอดสูญญากาศทั่วไปที่ปล่อยอิเล็กตรอนออกมาจากแคโถด แล้วจะมีการดึงลำอิเล็กตรอนให้วิ่งไปเป็นลำกระทบเข้ากับแอโนดหรือหน้าจอ ตรงหน้าจอนั้นมีการฉาบสารเรืองแสงไว้เพื่อให้เกิดการเรืองแสงของจอ



รูปที่ 2.2 แสดงวิธีการเบื้องต้นของการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

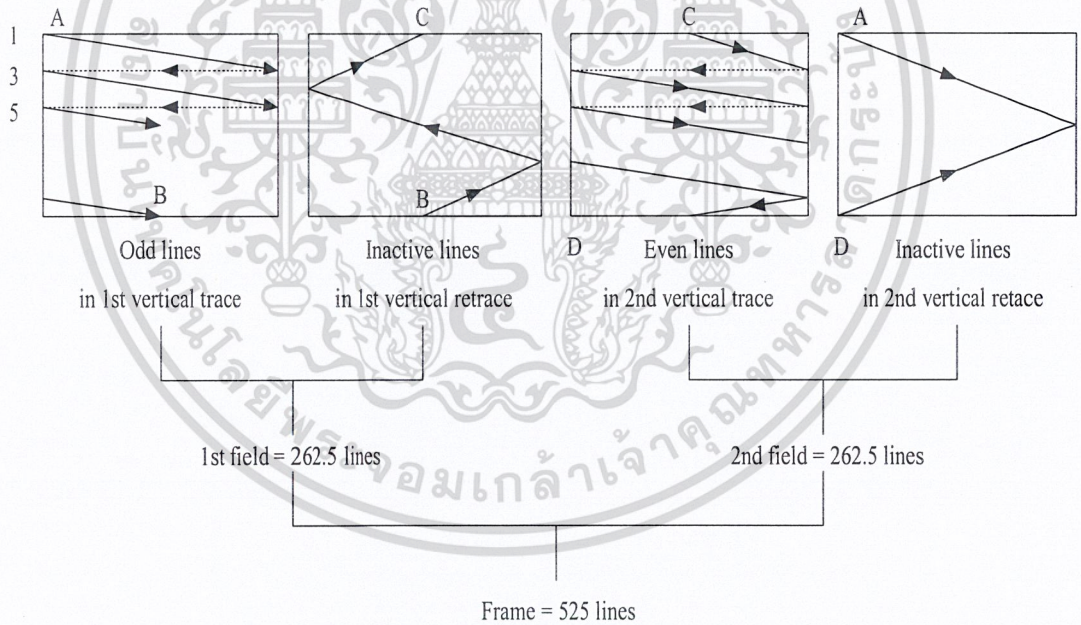
การสแกนของจอมี 2 วิธี คือการสแกนแบบก้าวหน้า (Progressive Scanning) กับการสแกนแบบสลับเส้น (Interlaced Scanning)

การที่จะทำการสแกนมีความต่อเนื่องขององค์ประกอบภาพดังที่กล่าวมาแล้วจะต้องคำนึงถึงหลัก 3 ประการ คือ

1. ลำอิเล็กตรอนกวาดไปทางแนวนอน (Horizontal Scanning) ในแต่ละครั้งต้องสามารถควบคุมองค์ประกอบภาพทั้งหมดของเส้นนั้น

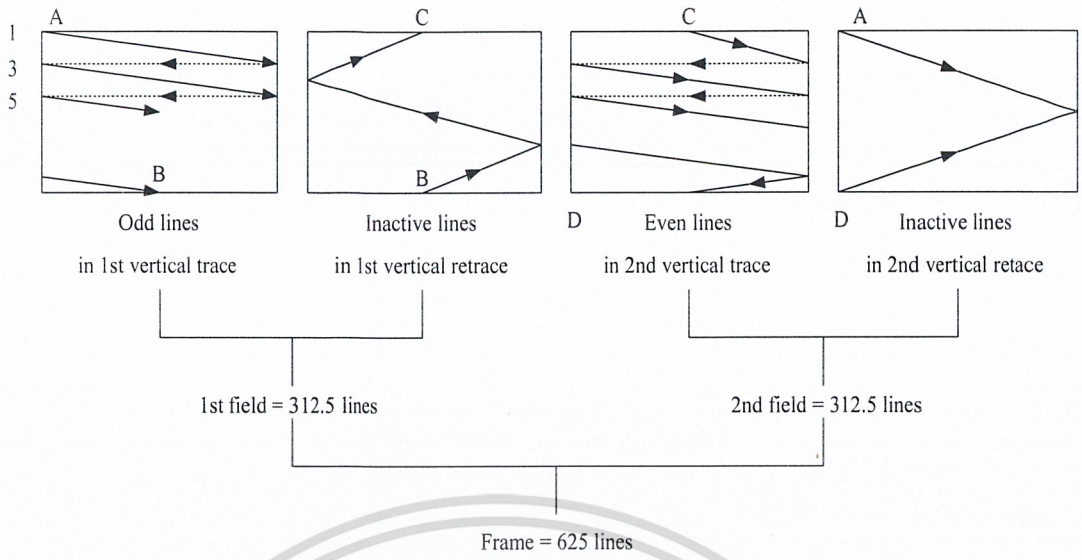
2. ในแต่ละเส้นของการสแกนลำอิเล็กตรอน ลำแสงต้องกวาดกลับเป็นความเร็วสูง ไปยังด้านซ้ายเพื่อเริ่มต้นเส้นภาพทางแนวนอนในลำดับต่อไป ดังนั้นเวลาของการสะบัดกลับเรียกว่า "รีเทรส" (Retrace) หรือฟลายแบ็ค (Flyback) ในกรณีดังกล่าวจะต้องไม่มีข้อมูลภาพใดๆ เพราะว่าทั้งกล้องถ่ายและหลอดภาพจะเกิดการแบลงค์เอาต์ (Blank Out) ในขณะนั้น

3. ในขณะที่เส้นสแกนสะบัดกลับมาเพื่อเริ่มต้นทางซ้ายใหม่ ตำแหน่งทางแนวตั้งจะต้องต่ำกว่าตำแหน่งเดิมเพื่อทำให้การสแกนเส้นต่อไปไม่ทับกัน ทั้งนี้เป็นเพราะการควบคุมของสัญญาณทางแนวตั้ง (Vertical scanning)



รูปที่ 2.3 รายละเอียดการสแกนแบบสลับเส้นหรือแบบแทรกสอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การสแกนสลับเส้นหรือแบบแทรกสอดในระบบ CCIR

การสแกนที่ใช้ในเครื่องรับโทรทัศน์ถึงแม้จะพบว่าหากให้มีการเรียงภาพเกินกว่า 16 ภาพต่อวินาทีแล้วสายตาจะเห็นเป็นภาพต่อเนื่องจากการทดลองสแกนภาพ จะพบว่าแม้ภาพที่เกิดขึ้น 24 ภาพต่อวินาทีแล้วก็ตาม ยังมีการกระพริบ (Flicker) เกิดขึ้นเนื่องจากว่าในขณะที่มีการสแกนเริ่มจากขอบบนลงมาด้านล่าง (ซึ่งคล้ายกับการเขียนหนังสือที่เริ่มจากซ้ายไปขวาและบนลงมาล่าง) เมื่อเส้นสแกนลงมายังด้านล่าง แสงทางด้านบนในความรู้สึกของมนุษย์เริ่มมืดลงกว่าด้านล่าง เวลาที่ลำแสงการสแกนวนกลับไปด้านบน ด้านล่างก็เกิดปัญหาเช่นเดียวกัน ความรู้สึกต่อกรณีนี้คือเกิดแสงกระพริบหรือวูบวาบขึ้น จึงต้องใช้การสแกนแบบสลับเส้นหรือบางคนเรียกว่า “การสแกนแบบสอดแทรก” (Interlaced Scanning) โดยครั้งแรกจะสแกนฟิลด์คี่ (Odd Line Trace) และครั้งต่อไปจะสแกนฟิลด์คู่ (Even Line Trace) เป็นการสแกนแบบเส้นเว้นเส้น นั่นหมายความว่าการได้ภาพ 1 ภาพหรือภาพ 1 เฟรม (Frame) ต้องใช้การสแกนแนวตั้งถึง 2 ครั้ง หรือ 2 ฟิลด์ (Field)

ตามมาตรฐานเอฟซีซี (FCC) ใช้การสแกน 525 เส้นต่อ 1 ภาพ และใช้ภาพ 30 ภาพต่อวินาที ดังนั้นใน 1 ฟิลด์ จะมีเส้นสแกน 262.5 เส้น ภาพนิ่งแต่ละภาพซึ่งเป็นส่วนประกอบขององค์ประกอบจะเกิดขึ้นภายใน 1/30 วินาที ความถี่ที่ใช้เพื่อการหักเหลำอิเล็กตรอนในแนวนอนจึงได้มาจากจำนวนเส้นภาพคูณกับจำนวนภาพในแต่ละเฟรมจึงได้ 525 × 30 = 15,750 เฮิรตซ์ ส่วนความถี่บังคับหักเหทางแนวตั้งในหนึ่งฟิลด์จะใช้เวลาเพียง 1/60 วินาทีเท่านั้น ความถี่จึงเท่ากับ 60 เฮิรตซ์

ตามมาตรฐานซีซีไออาร์ (CCIR) ใช้เส้นสแกน 625 เส้นต่อ 1 ภาพและใช้ภาพ 25 ภาพต่อวินาที ดังนั้นใน 1 ฟิลด์จะมีเส้นสแกน 312.5 เส้น ภาพนิ่งแต่ละภาพซึ่งเป็นส่วนประกอบขององค์ประกอบภาพจะเกิดขึ้นภายใน 1/25 วินาที ความถี่ที่ใช้เพื่อหักเหลำอิเล็กตรอนในแนวนอนจึงได้จากจำนวนเส้นภาพ 625 เส้นคูณกับจำนวนภาพในแต่ละเฟรม ดังนั้นจึงสามารถหาความถี่ได้จาก 625 × 25 = 15,625 เฮิรตซ์ ความถี่หักเหทางแนวตั้งจึงเท่ากับ 50 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 และ 2.4 ได้แสดงวิธีการของการสแกนแบบสอดแทรกของระบบโทรทัศน์ทั้งสองระบบใหญ่ๆ ที่ใช้กันอยู่ในปัจจุบัน โดยเมื่อเริ่มการสแกนสมมติว่าการเริ่มสแกนในกรณีนี้เริ่มจากการสแกนจากเฟรมที่เป็นเส้นสแกนคี่ โดยเริ่มจาก A ซึ่งอยู่ทางซ้ายแล้วกวาดไปทางขวานับเป็นเส้นสแกนเส้นที่ 1 แล้วจึงสแกนเส้นที่ 3, 5, 7, 9 และต่อๆ ไปเรื่อยๆ จนกระทั่งได้เส้นสแกน 262.5 เส้นในระบบเอฟซีซี หรือ 312.5 เส้นในระบบซีซีไออาร์ซึ่งก็คือ สแกนมาถึงจุด B ดังในภาพที่ 2.4 และณ. จุดนี้การสแกนจะถูกความถี่หักเหทางแนวตั้งซึ่งเรียกว่า เวอร์ติคอล รีเทรส (Vertical Retrace) หรือสัญญาณฟลายแบ็ค ดึงกลับไปยังตำแหน่งในจุด C เพื่อเริ่มต้นสแกนเส้นคู่ต่อไป

เวลาของการรีเทรส (Retrace Time) ทั้งการรีเทรสทางเวอร์ติคอลและฮอริซอลตอลเป็นเวลาสั้นๆ ถึงอย่างไรก็ตามไม่ต้องการให้เส้นสแกนที่เป็นช่วงของการสะบัดกลับนี้เข้ามารบกวน การทำให้เกิดสัญญาณภาพในส่วนนี้จึงต้องทำการลบเส้นสะบัดกลับก่อนที่จะถึงจุดที่ว่านั้น ลองมาดูรายละเอียดของการสะบัดกลับอีกนิดว่าในส่วนของการกวาดลำแสงหรือการสแกนในทางแนวนอน (Horizontal) เวลาของการรีเทรสจะได้ประมาณ 10-20 เปอร์เซ็นต์ของเวลาทั้งหมด ในระบบเอฟซีซีใช้เวลาในการสแกนทางแนวนอนเท่ากับ 63.5 uS ดังนั้นเวลาของการรีเทรสจึงเท่ากับ 6.35 uS ส่วนระบบซีซีไออาร์ ใช้เวลาในการสแกนเท่ากับ 64 uS ดังนั้นเวลาของการรีเทรสจะได้ประมาณ 6.4 uS ในส่วนของทางด้านความถี่หักเหทางแนวตั้ง (Vertical) ใช้เวลาในการรีเทรสไม่เกิน 5 - 8 เปอร์เซ็นต์ อย่างเช่นใช้เวลาในส่วนนี้เท่ากับ 3 เปอร์เซ็นต์ของเวลา 1/50 หรือ 1/60 ของวินาที ดังนั้นเวลาของการรีเทรสจึงเท่ากับ 600 uS และ 500 uS ตามลำดับ นั่นหมายความว่าช่วงเวลาของการรีเทรสทางแนวตั้งกินเวลานานกว่าการสแกนทางแนวนอนประมาณ 8 - 10 เส้นภาพ

จากหลักการดังกล่าวสามารถสรุปกฎเกณฑ์บางอย่างได้ว่า ความเป็นจริงแล้วในเส้นภาพ 525 หรือ 625 เส้นนั้นมิอาจจะเห็นได้ครบทุกเส้นอย่างน้อยๆ ในกรณีที่เกิดเวอร์ติคอล รีเทรส จะกินเวลาของการสแกนของภาพทางแนวนอน (Line) ไปด้วยแต่จะกินไปกี่เส้นนั้นขึ้นอยู่กับสัญญาณบังคับการฟลายแบ็ค ซึ่งในเครื่องรับเรียกตัวนี้ว่าสัญญาณแบลงค์กิ้ง

จากหลักการดังกล่าวนี้เองที่ไขประตูไปสู่โทรทัศน์ระบบดิจิตอลทำไมเวลาที่เอาสัญญาณภาพเข้าสู่หน่วยความจำของระบบดิจิตอลจึงเอาเส้นภาพไปเก็บเพียงครั้งละ 308 เส้นภาพเท่านั้น (แทนที่จะเอาไปเก็บทั้ง 312.5 เส้นภาพ) เพราะจากมาตรฐานโทรทัศน์ในปัจจุบันที่กำหนดมาตรฐาน สัญญาณในกรณีรีเทรสทางแนวตั้งเท่ากับ 3 เปอร์เซ็นต์ จึงได้เส้นภาพประมาณ 625 - 10 เส้นภาพ (615 เส้นภาพ) เมื่อแบ่งเฟรม แล้วจะได้ค่าเส้นภาพประมาณ 308 เส้นภาพ

2.1.3 สัญญาณภาพรวม

หากจะถามว่าเครื่องส่งจะทำส่งสัญญาณอะไรมาให้เครื่องรับบ้าง หากตอบกันง่ายๆ ก็ต้องตอบว่าส่งสัญญาณภาพรวม (Composite Video Signal) ซึ่งการจะทำให้เครื่องรับบรรลุวัตถุประสงค์นั้นต้องให้สัญญาณโทรทัศน์ส่งสัญญาณต่างๆ ดังต่อไปนี้

1. สัญญาณเสียง
2. สัญญาณภาพ
3. สัญญาณแบลงค์กิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัญญาณซิงโครไนซ์

5. สัญญาณอีควอไลซิ่ง

ในส่วนของระบบสัญญาณเสียงจะใช้คลื่นพาห์ (Carrier) เฉพาะ เพราะทราบกันเบื้องต้นแล้วว่าระบบเสียงในโทรทัศน์เป็นแบบเอฟเอ็ม ส่วนสัญญาณภาพและอื่นๆ ที่เหลือนั้นจะส่งเป็นสัญญาณภาพรวม หรือคอมโพสิทวิดีโอซิกแนล (Composite Video Signal) แล้วใช้คลื่นพาห์ภาพส่งออกไปการที่ต้องส่งสัญญาณทั้ง 5 ตัวออกอากาศแพร่คลื่นออกไปเพื่อวัตถุประสงค์ดังนี้

1. สัญญาณภาพและสัญญาณเสียง เป็นสัญญาณที่ส่งออกไปเพื่อให้เกิดภาพและเสียงขึ้นในเครื่องรับโทรทัศน์

2. สัญญาณแบลงค์กิ้ง เป็นสัญญาณที่ส่งเพื่อให้ลบเส้นสะบัดกลับทางแนวตั้งและแนวนอน

3. สัญญาณซิงโครไนซ์ เป็นสัญญาณที่ส่งมาเพื่อช่วยให้วงจรหักเหทางแนวตั้งและวงจรหักเหทางแนวนอนเพื่อให้เครื่องส่งกับเครื่องรับทำงานสอดคล้องตรงกัน

4. สัญญาณอีควอไลซิ่ง เป็นสัญญาณที่ช่วยให้สัญญาณซิงโครไนซ์ทั้งแนวตั้งและแนวนอนยังคงรูปเดิมอยู่ได้แม้ว่าจะเป็นการสแกนแบบสลับเส้นก็ตาม

ก. สัญญาณขาว-ดำ

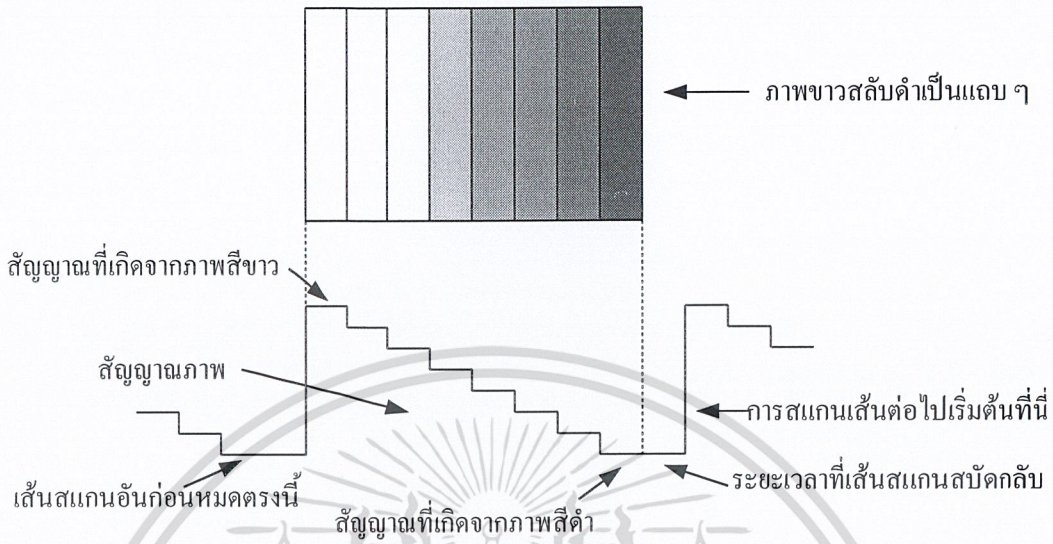
สมมติว่าจะทำการดูระดับสัญญาณขาว-ดำ กรณีที่กล่าวถึงสัญญาณขาว-ดำ หรือสัญญาณโมโนโครมได้ดีที่สุดต้องกล่าวว่าภาพที่เป็นสีขาวคือ ภาพที่มีความสว่างมากที่สุด และภาพที่เป็นสีดำคือ ภาพที่ไม่มีความสว่างเลย ภาพจำลองที่ดีที่สุดของกรณีนี้คือ แถบความถี่ที่มีความแตกต่างของสัญญาณขาว–ดำ ทีละน้อยซึ่งเรียกว่า ระดับเกรย์สเกล (Gray Scale) นั่นเอง

จากรูปภาพที่ 2.5 แสดงให้เห็นระดับของการเกรย์สเกล ในกรณีที่ระดับสัญญาณภาพเป็นระดับสัญญาณสีขาวเข้ามานั่นหมายความว่า ระดับความแรงของสัญญาณภาพที่มากที่สุดจึงให้ความสว่างที่หน้าจอสว่างที่สุด และเมื่อระดับสัญญาณที่ขาวลดลงเป็นสีมัว, เทา และดำนั้น ระดับสัญญาณจะลดลงเรื่อยๆ นั้น หมายความว่าสัญญาณมีความแรงน้อยลงจะมีความสว่างน้อยลงด้วย อันที่จริงแล้วภาพแต่ละภาพเป็นสัญญาณความถี่ทางไฟฟ้าที่มีความถี่สูงต่ำไม่เท่ากัน โดยความถี่สูงสุดจะไม่เกิน 4 เมกกะเฮิรตซ์ในระบบเอฟซีซี และไม่เกิน 5 เมกกะเฮิรตซ์ในระบบซีซีไออาร์ ภาพที่เกิดขึ้นจากความถี่สูงย่อมมีความละเอียดกว่าภาพที่เกิดจากความถี่ต่ำ (มีจำนวนจุดดำมากกว่า)

ข. สัญญาณแบลงค์กิ้ง

ทราบแล้วว่าเมื่อมีการสแกนลำอิเล็กตรอนที่หน้าจอจะเกิดเส้นรีเทรส หรือเส้นสะบัดกลับซึ่งเป็นเส้นภาพที่ไม่ต้องการ เครื่องส่งจึงต้องส่งสัญญาณแบลงค์กิ้งเพื่อให้เครื่องรับสามารถลบเส้นสะบัดกลับได้ สัญญาณแบลงค์กิ้งส่วนหนึ่งเครื่องรับจะต้องสร้างขึ้นเหมือนการสร้างสัญญาณซิงโครไนซ์ แต่มันจะสัมพันธ์กับเครื่องส่งได้อย่างไร จึงต้องมีการส่งสัญญาณแบลงค์กิ้งมาจากเครื่องส่งเพื่อลบเส้นสะบัดกลับในเครื่องรับ สัญญาณแบลงค์กิ้งมีอยู่ 2 อย่าง คือ เวอร์ติคอลแบลงค์กิ้ง และฮอร์ริซอลตอลแบลงค์กิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงระดับสัญญาณ ขาว-ดำโดยเกรย์สเกล

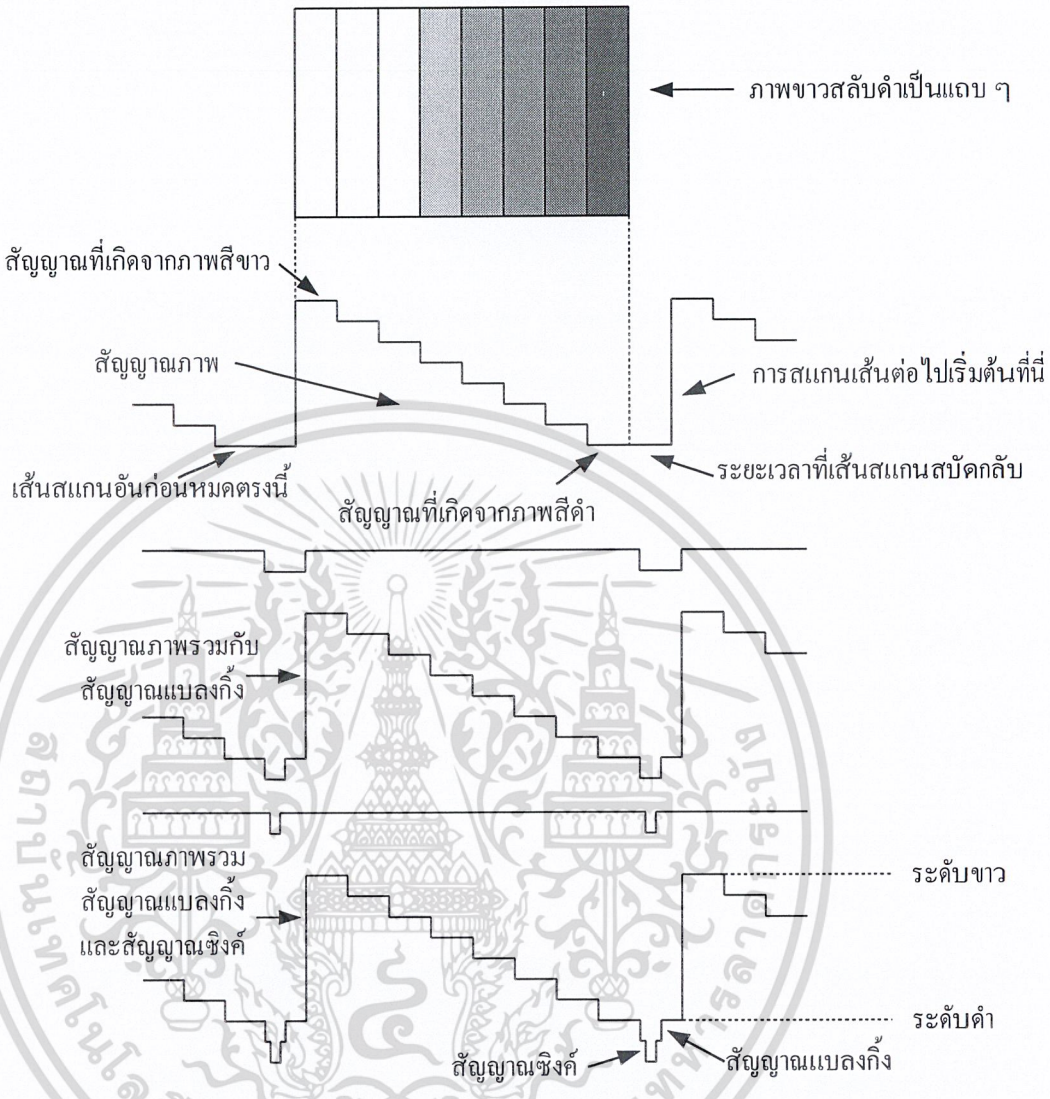
ค. สัญญาณซิงโครไนซ์

เป็นสัญญาณเพื่อให้การสแกนเป็นไปอย่างถูกต้องทั้งการสแกนแนวตั้งและแนวนอนนั้นทำโดยสัญญาณซิงโครไนซ์หรือสัญญาณซิงค์มีอยู่ 2 อย่าง คือ

1. ฮอร์ริซอลตอลซิงโครไนซ์ (Horiaontal Synchronize) เป็นสัญญาณซิงโครไนซ์ทางแนวนอนซึ่งมีความถี่ 15,625 เฮิรตซ์ (ในระบบCCIR) หรือ 15,750 เฮิรตซ์ (ในระบบFCC) ถ้าไม่มีสัญญาณส่วนนี้ส่งมาจะทำให้ภาพเกิดการล้มได้

2. เวอร์ติคอลซิงโครไนซ์ (Vertical Synchronize) เป็นสัญญาณซิงโครไนซ์ซึ่งมีความถี่ทางแนวตั้ง 50 เฮิรตซ์ (ในระบบ CCIR) หรือ 60 เฮิรตซ์ (ในระบบFCC) ถ้าไม่มีสัญญาณซิงโครไนซ์กับสัญญาณแบลงค์กิ้งไม่ว่าทางแนวนอนหรือแนวตั้ง จะมีความถี่เท่ากันเวลาส่งจึงต้องทำการกำหนดตำแหน่งให้ถูกต้องแล้ว มิฉะนั้นแล้วจะเกิดรบกวนกันได้ในทางปฏิบัติจึงให้สัญญาณซิงค์พัลส์มีขนาดความกว้างน้อยกว่าสัญญาณแบลงค์กิ้งพัลส์แล้วใช้วิธีการส่งแบบผสมกล่าวคือ ส่งซิงค์พัลส์กับแบลงค์กิ้งพัลส์ไปด้วยกัน ให้แบลงค์กิ้งพัลส์เป็นฐานของซิงค์พัลส์ เมื่อมีการจัดระดับของสัญญาณส่วนนี้เทียบกับเกรย์สเกลระดับของสัญญาณแบลงค์กิ้งจะอยู่ที่ระดับต่ำกว่าดำ ส่วนสัญญาณซิงค์จะอยู่ที่ระดับต่ำกว่าดำลงไปอีกสัญญาณเหล่านี้จึงไม่มีการส่งผลต่อการมองเห็น (หรือไม่รบกวนสัญญาณภาพ) ดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

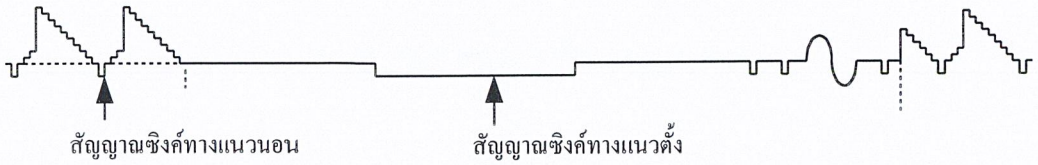


รูปที่ 2.6 แสดงลักษณะสัญญาณภาพรวมเบื้องต้น

ง. สัญญาณอีควอไลซิ่ง

เป็นสัญญาณบังคับรูปร่างของสัญญาณซิงโครไนซ์ทางแนวตั้ง เพื่อให้สามารถคงรูปถูกต้องแล้ว ยังช่วยให้การสแกนสลับเส้นเป็นไปได้ถูกต้องด้วย สามารถขึ้นเส้นคู่ในตำแหน่งที่ถูกต้องได้และยังส่งผลให้สัญญาณซิงโครไนซ์ทางแนวนอนไม่ขาดช่วงหายไปในระหว่างการส่งสัญญาณทางแนวตั้ง ซึ่งสัญญาณนี้มีขนาดของพัลส์รวมเท่ากับเวอร์ติคอลซิงโครไนซ์พัลส์ สัญญาณส่วนนี้จะถูกแบ่งออกเป็น 6 ลูกเล็กๆ ในระบบ 525 เส้น และถูกแบ่งเป็น 5 ลูกในระบบ 625 เส้น ดังแสดงไว้ในรูป 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

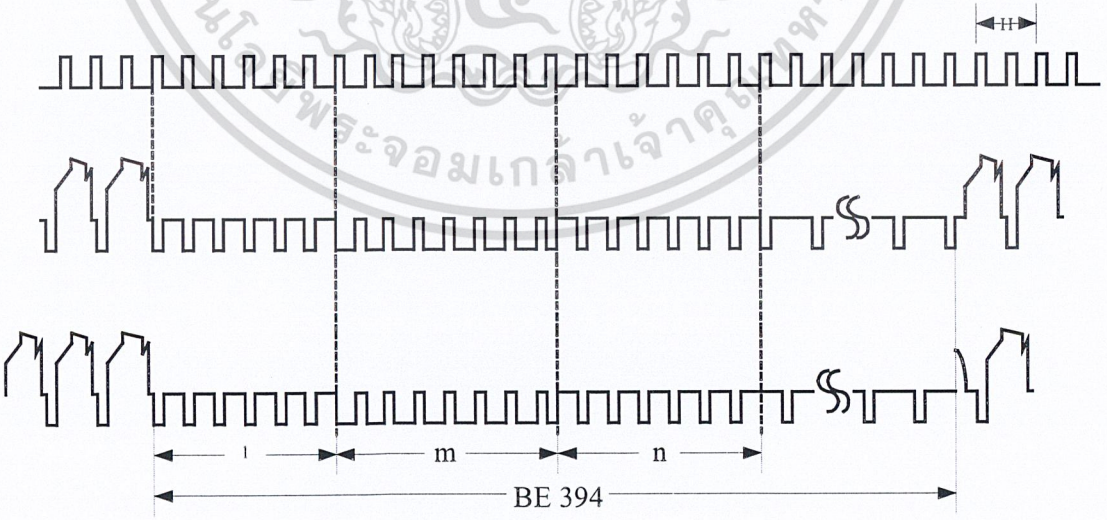


ก.สัญญาณภาพรวม แสดงให้เห็นสัญญาณภาพกับสัญญาณซิงค์ทางแนวนอนและสัญญาณซิงค์ทางแนวตั้ง

ข.รายละเอียดของสัญญาณซิงค์ทางแนวตั้ง (ภายหลังการสแกนฟิลด์เส้นคู่แล้ว)

ค.รายละเอียดสัญญาณซิงค์ทางแนวตั้ง (ภายหลังการสแกนฟิลด์เส้นคี่แล้ว)

รูปที่ 2.7 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง



รูปที่ 2.8 แสดงระบบพัลส์ในฟิลด์ของการซิงโครไนซ์ของระบบโทรทัศน์

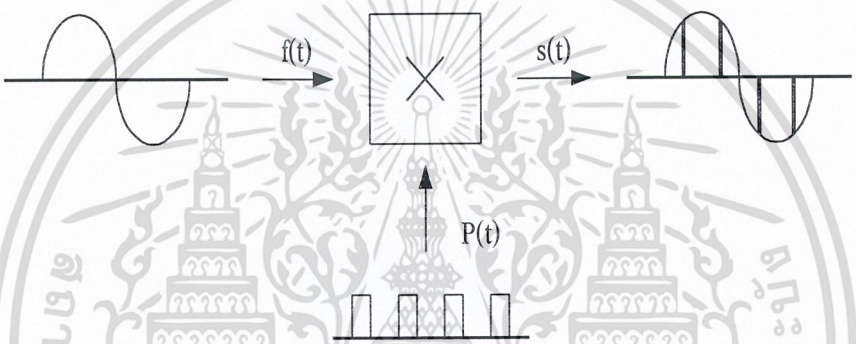
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital Conversion)

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลประกอบด้วยขบวนการสำคัญ 3 ส่วน คือ

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

ส่วนประกอบที่สำคัญที่สุดคือ การแซมปลิ้งเพราะความผิดพลาดของสัญญาณดิจิตอลที่แปลงมาจากสัญญาณอนาลอกนั้นจะมากหรือน้อยขึ้นอยู่กับความสัมพันธ์ของความถี่แซมปลิ้งกับความถี่สูงสุดของสัญญาณอนาลอกโดยทฤษฎีการแซมปลิ้ง (Sampling Theory) ซึ่งความสัมพันธ์ตามทฤษฎีการแซมปลิ้งดูได้จากรูป 2.9



รูปที่ 2.9 แสดงการแซมปลิ้ง

จากรูปจะได้ $S(t) = P(t) \times f(t)$

เมื่อดู P(t) ซึ่งเป็นพัลส์รูปสี่เหลี่ยม ถ้านำมาเขียนสมการฟูเรียร์ (Fourier) จะได้

$$P(t) = DC + (a_0 \cos\ w_0 t + a_1 \cos 3w_0 t + a_2 \cos 5w_0 t + \ldots\ldots$$

ซึ่งP(t) ประกอบไปด้วยความถี่พื้นฐานรวมกับฮาร์โมนิคที่เป็นเลขคี่ไปจนถึง infinite และถ้าคูณ P(t) ด้วย f(t) จะได้ S(t)

$$S(t) = f(t) \times DC + (a_0 \cos w_0 t) \times f(t) + (a_1 \cos 3w_0 t) \times f(t) + \ldots\ldots$$

เมื่อพิจารณาดูเทอมที่ 2 จะพบว่ามีรูปแบบเหมือนแอมปลิจูดมอดดูเลชั่น (AM)

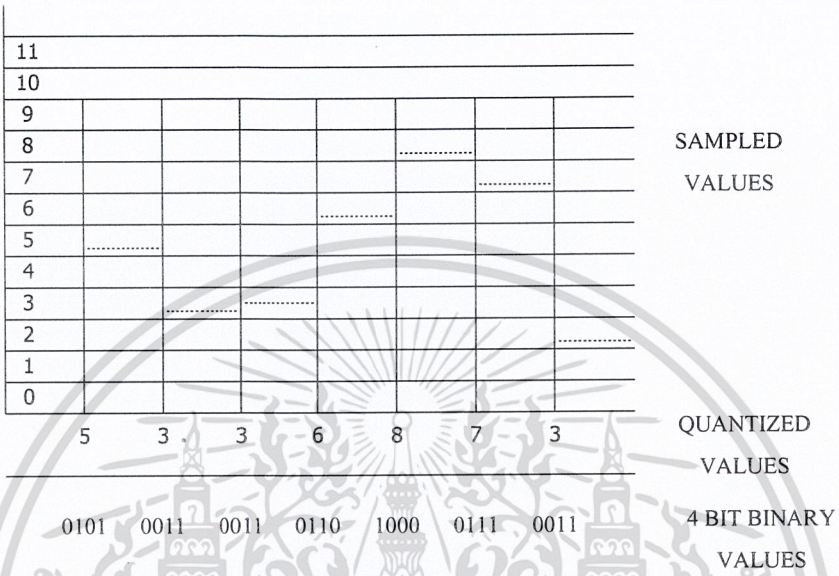
โดยถ้า $f(t) = B \cos w_m t$

$$F(t) \times (a_0 \cos w_0 t) = (Ba_0/2) \cos (w_0 - w_m)t + (Ba_0/2) \cos (w_0 + w_m)t$$

ซึ่งความถี่ w_0 ที่ใช้สำหรับการแซมปลิ้งและการดีเทคสัญญาณที่ได้คืนจะได้วงจรกรองความถี่ต่ำ (Low Pass Filter) กรองเอาเฉพาะ $f(t) \times DC$ ออกมาเท่านั้น ซึ่งถ้า w_0 มีค่าน้อยกว่า 2 เท่าของ w_m แล้วจะทำให้ความถี่ซึ่งเป็นผลต่างของ $w_0 - w_m$ เข้ามาแทรกใน $f(t) \times DC$ ด้วย ซึ่งจะมีผลให้สัญญาณที่ดีเทคกลับคืนมามีความผิดพลาด ดังนั้นจึงต้องเลือก W_0 หรือความถี่แซมปลิ้งให้มีค่ามากกว่า 2 เท่าของความถี่สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสัญญาณก่อนที่จะมีการแซมปลิ้ง หรือ w_m ซึ่งสัญญาณที่ได้ออกมาจากภาพแซมปลิ้งนี้เรียกว่า พัลส์แอมปลิจูดมอดดูเลชั่น (Pulse Amplitude Modulation: PCM)

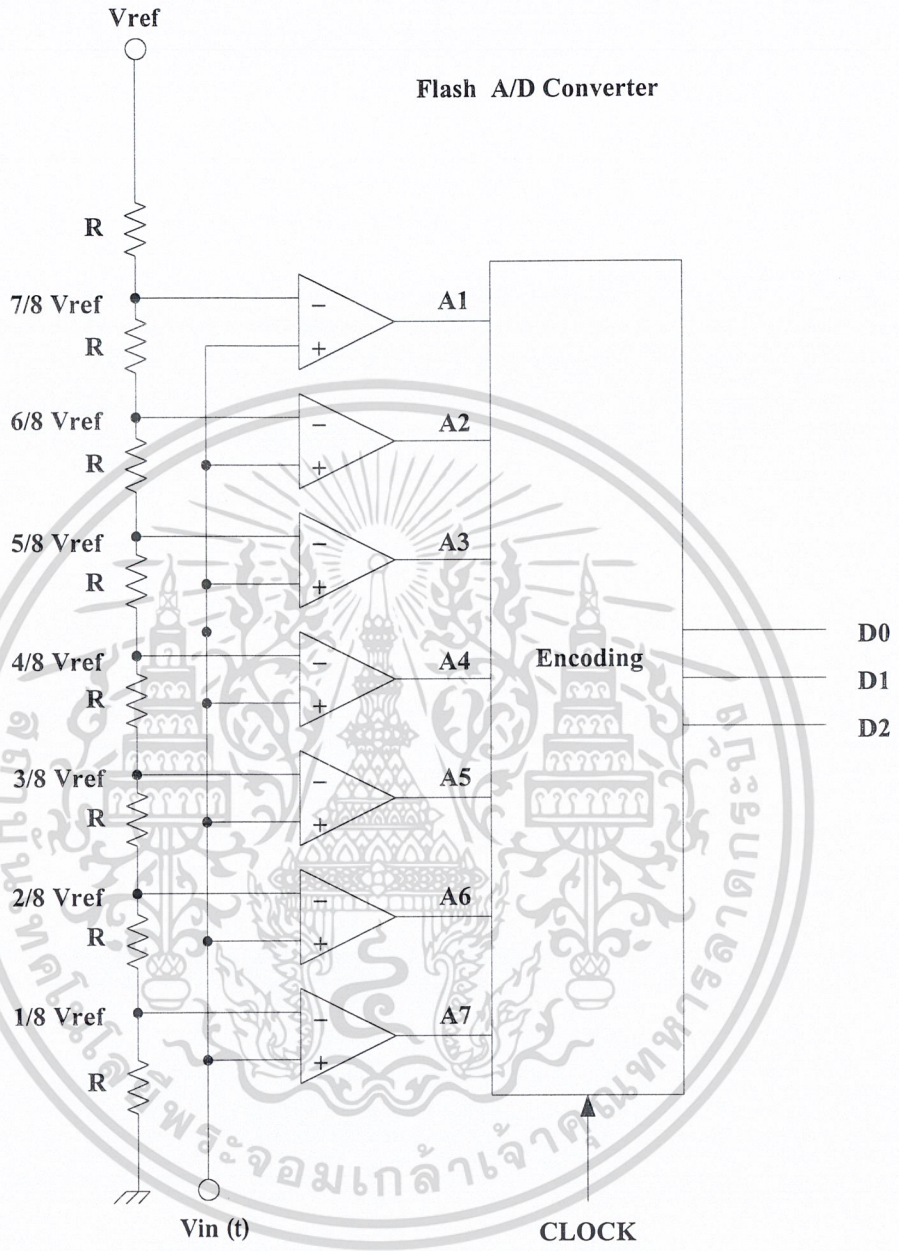


รูปที่ 2.10 แสดงลักษณะสัญญาณของภาคควอนไตซิ่งและการเข้ารหัส

ภาคควอนไตซิ่ง เป็นการจัดระดับของสัญญาณ PCM ซึ่งอาจมีระดับที่ไม่แน่นอนให้อยู่ในระดับที่แน่นอนซึ่งในขั้นตอนนี้จะมีความผิดพลาดจากการจัดระดับอยู่จึงเรียกว่า ควอนไตซิ่งเออร์เรอร์ (Quantizing Error) หรือสัญญาณรบกวนควอนไตซิ่ง (Quantizing Noise) ซึ่งจะมีค่ามากหรือน้อยขึ้นอยู่กับระดับสัญญาณที่จะแบ่ง 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 Converter) คือ วงจรที่ทำหน้าที่เปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (A/D Converter) ที่มีความเร็วสูงในการเปลี่ยนและเนื่องจากแฟลชเอทูดีจะใช้การโปรแกรมเอาต์พุตเอาไว้ แล้วในส่วนของวงจรเปรียบเทียบจะใช้ออปแอมป์ที่มีจำนวนเท่ากับจำนวนของสัญญาณเอาต์พุตที่จะเกิดขึ้น เช่น ถ้าสัญญาณเอาต์พุตเป็นดิจิตอล 8 บิตจะใช้ออปแอมป์ $2^8 = 256$ ตัวซึ่งแต่ละตัวจะมีแรงดันอ้างอิงที่เป็นค่าคงที่อยู่ค่าหนึ่งตามระดับสัญญาณดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงวงจร Flash A/D

เมื่อสัญญาณอินพุตเข้ามาจะถูกส่งไปให้กับออปแอมป์ทุกตัวเพื่อเปรียบเทียบแรงดันอ้างอิงของแต่ละตัว ถ้าสัญญาณอินพุตไปตรงกับออปแอมป์ตัวใดก็จะให้สัญญาณเอาต์พุตออกมาส่งไปให้กับวงจรเข้ารหัส (Encoder) เพื่อทำการเข้ารหัสสัญญาณดิจิตอลซึ่งเป็นสัญญาณเอาต์พุตที่ได้ให้ตรงตามค่าของสัญญาณทางด้านอินพุตที่ส่งเข้ามาซึ่งการเปลี่ยนสัญญาณในรูปนี้ไม่ต้องใช้วงจรนับ (Counter) แล้วป้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับมาเปรียบเทียบทีละค่าจึงทำให้ความเร็วในการเปลี่ยนสัญญาณสูงกว่าแบบแรกมากจึงเหมาะที่จะนำมาใช้กับสัญญาณภาพซึ่งมีความถี่สูง

ข้อดีและข้อเสียของวงจร Flash A/D

ข้อดี คือ มีความสามารถในการเปลี่ยนสัญญาณได้เร็วมาก

ข้อเสีย คือ จะต้องใช้จำนวนตัวเปรียบเทียบมากเป็น 2 เท่าเมื่อต้องการบิตเพิ่ม 1 บิต หรือถ้าเป็นสมการจำนวนออปแอมป์เท่ากับ $2^N - 1$ ตัวโดย N คือจำนวนบิต ดังนั้นถ้าใช้สัญญาณดิจิตอล 8 บิต จะต้องใช้ตัวเปรียบเทียบ 256 ตัว ซึ่งทำให้อุปกรณ์มีราคาสูงมาก

2.3 พอร์ตขนาน (Parallel Port)

พอร์ต หมายถึง ฮาร์ดแวร์ที่ทำหน้าที่รับส่งข้อมูลกับอุปกรณ์อื่นใดของหน่วยประมวลผล ซึ่งแบ่งเป็น 2 ลักษณะใหญ่ๆ คือ พอร์ตอนุกรมและพอร์ตขนาน

พอร์ตอนุกรม คือ พอร์ตที่จะทำการรับส่งข้อมูลทีละบิตโดยใช้สายเพียงเส้นเดียว ตัวอย่างเช่น RS-232, USB เป็นต้น

พอร์ตขนาน คือ พอร์ตที่ทำการรับส่งข้อมูลครั้งละมากกว่า 1 บิตโดยใช้สายมากกว่า 1 เส้น ตัวอย่างเช่น IEEE 1284 หรือที่เรียกกันทั่วไปว่า พอร์ตปริ้นเตอร์ ซึ่งมีความเร็วในการส่งข้อมูลได้ถึง 150 กิโลไบต์ต่อวินาที

ลักษณะโดยทั่วไปของพอร์ตขนานจะประกอบไปด้วย 3 พอร์ตได้แก่

1. พอร์ตข้อมูล (Data Port) เป็นเอาต์พุตพอร์ต ซึ่งพอร์ตนี้จะทำหน้าที่ในการส่งข้อมูลจากเครื่องคอมพิวเตอร์ไปยังอุปกรณ์ภายนอก ซึ่งที่ใช้กันทั่วไปก็คือ ปริ้นเตอร์หรือสแกนเนอร์

2. พอร์ตสถานะ(Status Port) เป็นอินพุตพอร์ต ซึ่งพอร์ตนี้จะทำหน้าที่ตรวจสอบสถานะที่แจ้งมาจากอุปกรณ์ภายนอกเพื่อให้การรับส่งข้อมูลเป็นไปอย่างถูกต้อง เนื่องจากว่าอุปกรณ์ภายนอกที่รับข้อมูลจากเครื่องคอมพิวเตอร์จะทำการรับข้อมูลได้ช้ากว่าความเร็วที่เครื่องคอมพิวเตอร์สามารถส่งได้ดังนั้นเพื่อไม่ให้เกิดความผิดพลาดในการรับส่งข้อมูลเครื่องคอมพิวเตอร์จะส่งข้อมูลเครื่องคอมพิวเตอร์จะทำการส่งข้อมูลก็ต่อเมื่ออุปกรณ์ภายนอกพร้อมที่จะรับข้อมูลโดยตรวจได้จากพอร์ตนี้

3. พอร์ตควบคุม(Control Port) เป็นเอาต์พุตพอร์ต ซึ่งพอร์ตนี้จะทำหน้าที่ควบคุมการทำงานที่อุปกรณ์ภายนอกโดยเครื่องคอมพิวเตอร์จะแจ้งถึงข้อมูลต่างๆ ที่อุปกรณ์ภายนอกจะต้องเตรียมและปฏิบัติเพื่อที่จะรับข้อมูลที่ส่งมา รวมถึงในกรณีที่มีอุปกรณ์ภายนอกมากกว่าหนึ่งตัวต่อเข้ากับพอร์ตขนานเครื่องคอมพิวเตอร์สามารถที่จะเลือกติดต่อกับอุปกรณ์ใดก็ได้โดยผ่านพอร์ตนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงรายละเอียดขาสัญญาณต่างๆ ของพอร์ตขนาน

ขา	หน้าที่การทำงาน
DataBit 0 – 7	ส่งข้อมูลจากเครื่องคอมพิวเตอร์ไปยังอุปกรณ์ภายนอก
nError(nFault)	แจ้งให้เครื่องคอมพิวเตอร์ทราบว่าอุปกรณ์ที่เชื่อมต่อมีความผิดพลาดเกิดขึ้น
nSelectIn	แจ้งให้เครื่องคอมพิวเตอร์ทราบว่าเครื่องพิมพ์ทำงานอยู่
PaperEnd	แจ้งให้เครื่องคอมพิวเตอร์ทราบว่าเครื่องพิมพ์ไม่มีกระดาษ
nAck	แจ้งให้เครื่องคอมพิวเตอร์ทราบว่าอุปกรณ์ภายนอกพร้อมที่จะรับข้อมูล
Busy	แจ้งให้เครื่องคอมพิวเตอร์ทราบว่าอุปกรณ์ภายนอกไม่พร้อมที่จะรับข้อมูล
nStrobe	แจ้งให้อุปกรณ์ภายนอกทราบว่าเครื่องคอมพิวเตอร์พร้อมที่จะส่งข้อมูลให้รอรับข้อมูล
nAutoLF	แจ้งให้อุปกรณ์ภายนอกที่เป็นปริ้นเตอร์เลื่อนกระดาษไป1บรรทัด
nInit	แจ้งให้อุปกรณ์ภายนอกทราบว่าเครื่องคอมพิวเตอร์จะทำการกำหนดค่าเริ่มต้นให้
nSelectIn	แจ้งให้อุปกรณ์ภายนอกทราบว่าเครื่องคอมพิวเตอร์ต้องการเลือกอุปกรณ์ตามที่ระบุมาบนขาสัญญาณนี้

หมายเหตุ อักษร 'n' ที่อยู่หน้าขาต่างๆแสดงถึงสถานะลอจิกที่ขาของพอร์ตขนานและรีจิสเตอร์มีสถานะตรงข้ามกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.2 แสดงสัญญาณและขาสัญญาณของพอร์ตขนาน IEEE 1284

ชื่อสัญญาณ	รีจิสเตอร์บิต	ขาสัญญาณ		ขากราวด์อ้างอิง	
		DB 25 IEEE1284 A	Centronics36 IEEE 1284 B	DB 25 IEEE1284 A	Centronics36 IEEE 1284 B
Data Bit 0	D0	2	2	19	20
Data Bit 1	D1	3	3	19	21
Data Bit 2	D2	4	4	20	22
Data Bit 3	D3	5	5	20	23
Data Bit 4	D4	6	6	21	24
Data Bit 5	D5	7	7	21	25
Data Bit 6	D6	8	8	22	26
Data Bit 7	D7	9	9	22	27
nError(nFault)	+S3	15	32	23	29
nSelectIn	+S4	13	13	24	28
PaperEnd	+S5	12	12	24	28
nAck	+S6	10	10	24	28
Busy	-S7	11	11	23	29
nStrobe	-C0	1	1	18	19
nAutoLF	-C1	14	14	25	30
nInit	+C2	16	31	25	30
nSelectIn	-C3	17	36	25	30
HostLogicHigh	-	-	18	-	-
PeriphLogicHigh	-	-	36	-	-

หมายเหตุ เครื่องหมายบวก(+) และ ลบ(-) มีความหมายดังนี้

+ คือ สถานะลอจิกที่ขาของพอร์ตขนานและในรีจิสเตอร์มีสถานะเหมือนกัน เช่น ขาสัญญาณ nAck ที่อุปกรณ์ภายนอกป้อนเข้าเป็น ‘0’ เมื่อผ่านเข้ามาเก็บในรีจิสเตอร์ค่าที่อ่านได้จะมีค่าเป็น ‘0’

- คือ สถานะลอจิกที่ขาของพอร์ตขนานและในรีจิสเตอร์มีสถานะต่างกัน เช่น ขาสัญญาณ Busy ที่อุปกรณ์ภายนอกป้อนเข้าเป็น ‘0’ เมื่อผ่านเข้ามาเก็บในรีจิสเตอร์ค่าที่อ่านได้จะมีค่าเป็น ‘1’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46475

โดยทั่วไปแล้วพอร์ตขนานจะมีตำแหน่งฐาน (Base Address) อยู่ซึ่งแสดงดังตารางด้านล่างนี้

ตารางที่ 2.3 แสดงตำแหน่งฐานของพอร์ตขนาน

ตำแหน่ง	คำอธิบาย
3BCh – 3BFh	ใช้สำหรับพอร์ตขนานที่รวมอยู่ในการ์ด – ไม่สามารถใช้กับ ECP แอดเดรสได้
378h – 37Fh	นิยมใช้กับแอดเดรสสำหรับ LPT1
278h – 27Fh	นิยมใช้กับแอดเดรสสำหรับ LPT2

ในช่วงเริ่มแรกนั้นพอร์ตข้อมูลของพอร์ตขนานสามารถส่งข้อมูลออกได้เพียงอย่างเดียว ซึ่งจะเรียกพอร์ตขนานแบบนี้ว่า พอร์ตขนานแบบมาตรฐาน (Standard Parallel Port : SPP) ดังนั้นเมื่อทำการต่อกับอุปกรณ์ภายนอกที่ต้องการส่งข้อมูลมายังเครื่องคอมพิวเตอร์จะส่งข้อมูลผ่านพอร์ตสถานะซึ่งสามารถส่งได้สูงสุดครั้งละ 5 บิต ในกรณีที่มีความต้องการใช้ข้อมูลครั้งละ 1ไบต์จะต้องทำการรับข้อมูล 2 ครั้ง โดยจะต้องทำการเก็บข้อมูล 4 บิตแรกที่รับเข้าไปนำไปเก็บในบัฟเฟอร์ก่อนจากนั้นจึงอ่านข้อมูลอีก 4 บิตเข้าไปซึ่งเป็นการยุ่งยากในการใช้งาน ต่อมาได้มีการพัฒนาพอร์ตขนานให้พอร์ตข้อมูลสามารถรับข้อมูลได้ซึ่งจะเรียกพอร์ตขนานแบบนี้ว่า พอร์ตขนานสองทิศทาง (Bi-directional Parallel Port) ทำให้อุปกรณ์ภายนอกสามารถส่งข้อมูลเข้ามายังพอร์ตข้อมูลครั้งละ 8 บิตหรือ 1 ไบต์เป็นผลให้การติดต่อผ่านพอร์ตขนานมีความเร็วและสะดวกต่อการใช้งานยิ่งขึ้น

ในปัจจุบันนี้พอร์ตขนานได้มีการพัฒนาให้มีความเร็วในการใช้งานมากขึ้น ซึ่งได้มีการแบ่งออกเป็น 2 ชนิดใหญ่ๆ คือ

- Enhanced Parallel Port (EPP) ซึ่งพอร์ตชนิดนี้ใช้กับอุปกรณ์ภายนอกต่างๆ ที่ไม่ใช่ ปริ้นเตอร์
- Extended Capabilities Port (ECP) ซึ่งพอร์ตชนิดนี้ใช้กับอุปกรณ์ภายนอกที่เป็น ปริ้นเตอร์ หรือสแกนเนอร์

พอร์ตทั้ง 2 ชนิดนี้มีความเร็วในการส่งข้อมูลได้ถึง 2 เมกกะไบต์ต่อวินาทีและสามารถเซตให้เป็นได้ทั้งพอร์ตขนานทิศทางเดียวหรือสองทิศทาง

จะพบว่าในพอร์ตขนานหนึ่งตัวจะประกอบไปด้วยพอร์ตหลายส่วนคือ พอร์ตข้อมูล, พอร์ตสถานะและพอร์ตควบคุม แต่ใน EPP จะมีเพิ่มขึ้นมาอีก 2 พอร์ตคือ พอร์ตชี้ตำแหน่ง i/o และพอร์ตข้อมูล i/o ซึ่งสามารถที่จะทำการชี้พอร์ตเหล่านี้ได้โดยการอ้างอิงตำแหน่งตามตารางด้านล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงการชี้ตำแหน่งต่างของ EPP

แอดเดรส	ชื่อพอร์ต	อ่าน/เขียน
Base+0	พอร์ตข้อมูล (ทั้งพอร์ตทิศทางเดียวและEPP)	เขียน
Base+1	พอร์ตข้อมูล (ทั้งพอร์ตทิศทางเดียวและEPP)	อ่าน
Base+2	พอร์ตควบคุม (ทั้งพอร์ตทิศทางเดียวและEPP)	เขียน
Base+3	พอร์ตชี้ตำแหน่ง (EPP)	อ่าน/เขียน
Base+4	พอร์ตข้อมูล (EPP)	อ่าน/เขียน
Base+5	พอร์ตขนส่งข้อมูล 16/32 บิต	-
Base+6	พอร์ตขนส่งข้อมูล 32 บิต	-
Base+7	พอร์ตขนส่งข้อมูล 32 บิต	-

หมายเหตุ - แอดเดรสในตำแหน่ง Base+0, Base+3 และ Base+4 จะทำการติดต่อกับอุปกรณ์ภายนอกผ่านขา 2 ถึง 9 ดังนั้นจึงเลือกใช้งานได้เพียงครั้งละแอดเดรสใดแอดเดรสหนึ่งเท่านั้น

- Base คือ แอดเดรสฐานของพอร์ตขนาน ตามที่แสดงไว้ดังตารางที่ 2.3

โดยปกติหากใช้พอร์ตขนานที่เป็น EPP ก็สามารถที่จะให้ทำการรับข้อมูลจากภายนอกโดยผ่านขาข้อมูล 2 ถึง 9 ไปยังแอดเดรสพอร์ตชี้ตำแหน่ง i/o หรือพอร์ตข้อมูล i/o ก็ได้โดยไม่ต้องเซตเป็นพอร์ตขนานสองทิศทาง แต่ถ้าหากทำการเซต EPP ให้เป็นพอร์ตขนานสองทิศทางจะสามารถรับข้อมูลจากภายนอกได้โดยใช้พอร์ตข้อมูลโดยตรง

การเซตพอร์ตขนานให้ทำงานเป็นพอร์ตขนานสองทิศทาง ที่สามารถจะรับข้อมูลทางพอร์ตข้อมูล(แอดเดรสฐาน+0) จะต้องทำการเซตบิตที่ 5 ของพอร์ตควบคุมให้เป็น '1' หรือสถานะ 'สูง' ซึ่งบิตนี้จะไม่มีขาต่อออกมาภายนอกแต่จะเป็นรีจิสเตอร์ภายใน ดังนั้นในการเซตบิตที่ 5 นี้จะใช้โปรแกรมเป็นตัวเซต ตัวอย่างเช่น

```
outportb(port+2,0x221);// ทำการเซตบิตที่ 5 เป็น '1' เพื่อเตรียมรับข้อมูล
inportb(port+0,rw);// รอรับข้อมูลที่เข้ามาทางพอร์ตข้อมูลและนำไปเก็บในตัวแปร rw
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 แสดงรายละเอียดต่างๆ ของรีจิสเตอร์พอร์ตควบคุม

บิต	ขาที่	รายละเอียด
7	-	ถูกจองไว้
6	-	ถูกจองไว้
5	-	อิเนเบิลพอร์ตสองทิศทาง
4	10	อิเนเบิล IRQ ผ่าน ACK
3	17	เลือกปริ้นเตอร์
2	16	เริ่มต้นปริ้นเตอร์
1	14	เลื่อนกระดาษหนึ่งบรรทัดโดยอัตโนมัติ
0	1	สโตรป

2.4 การออกแบบส่วนโปรแกรมควบคุม

โครงงานนี้มีหลักการทำงานโดยการรับสัญญาณจากเครื่องคอมพิวเตอร์ ไปควบคุมอุปกรณ์ทางด้านฮาร์ดแวร์และรับข้อมูลภาพ การออกแบบส่วนโปรแกรมควบคุมจึงเป็นการติดต่อควบคุมระหว่างฮาร์ดแวร์ที่สร้างขึ้นและการรับข้อมูลภาพที่ได้จากกล้อง โดยใช้โปรแกรม BORLAND C++ BUILDER 5.0 เป็นคอมไพล์เลอร์ในการเขียนโปรแกรม ในส่วนขั้นตอนการทดสอบเราได้ใช้โปรแกรม TURBO C เป็นคอมไพล์เลอร์เนื่องจากสามารถทดสอบโปรแกรมได้ง่าย โดยโปรแกรมที่ใช้ได้รวมไว้ในภาคผนวก

2.4.1 การเขียนโปรแกรมรับ-ส่งข้อมูล

ในการเขียนโปรแกรมเพื่อติดต่อกับอุปกรณ์ภายในเครื่องคอมพิวเตอร์ เมื่อใช้คอมไพล์เลอร์ภาษา C โดยเช่น Turbo C จะใช้คำสั่ง

```
int             inport(int portid);
unsigned char   inportb(int portid);
void            outport(int portid, int value);
void            outportb(int portid, unsigned char value);
```

คำสั่ง outportb( ) เป็นฟังก์ชั่นที่ใช้ในการส่งข้อมูลออกทางพอร์ตของเครื่องพีซีครั้งละ 1 ไบต์ตามหมายเลขพอร์ตที่กำหนดถูกนิยามไว้ใน dos.h เรียกใช้โดยใส่ #include < dos.h > ที่โปรแกรมส่วนบน

รูปแบบ void outportb(int portid, int value);

โดยที่ portid คือ หมายเลขพอร์ต

value คือ ข้อมูล

ตัวอย่าง outportb(0x378,0xff);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างคือ การส่งข้อมูล “FF” ซึ่งเป็นฐาน 16 ออกทางพอร์ต 378H (พอร์ตขนานที่ 1) หรือเขียนในรูปแบบเลขฐานสิบได้คือ 255 ข้อมูลที่ออกทางพอร์ตคือ “FF” หรือ “11111111” ฐานสองจะมีสถานะดังตารางที่ 2.6

ตารางที่ 2.6 แสดงสถานะข้อมูล

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

แต่ในการเขียนโปรแกรมติดต่อรับ-ส่งข้อมูล โดยใช้ Borland C++ Builder 5.0 จะไม่สามารถใช้คำสั่งของโปรแกรม Turbo C ได้เนื่องจากการเขียนโปรแกรมภายใต้ระบบปฏิบัติการ WINDOWS ซึ่งเป็นการทำงานแบบมัลติทาสกิ้ง (Multi Tasking) การติดต่อรับส่งข้อมูลโดยตรงอาจทำให้มีปัญหากับระบบปฏิบัติการได้ ทางบริษัทบอร์แลนด์ (Borland) จึงไม่สร้างคำสั่ง inport, inportb, outport และ outportb ไว้ในโปรแกรม แต่สามารถเขียนเป็นโปรแกรมย่อยจากภาษาแอสแซมบลี้ โดยสามารถดูคำสั่งได้จากภาคผนวก ซึ่งสามารถเขียนโปรแกรมได้โดยตรง ซึ่งโปรแกรม BCB มีคุณสมบัติพิเศษสามารถคอมไพล์โปรแกรมภาษาแอสแซมบลี้ภายในโปรแกรมได้โดยต้องมี “asm” นำหน้าคำสั่งภาษาแอสแซมบลี้ทุกครั้ง

การรับ - ส่งข้อมูลภายในโครงงานจะเป็นติดต่อควบคุมโดยใช้พอร์ตขนานที่ 1 ซึ่งอยู่ที่ตำแหน่ง 378H และมีรีจีสเตอร์ต่างๆ ดังนี้

ตารางที่ 2.7 แสดงพอร์ตใช้งาน

รีจีสเตอร์	ตำแหน่ง	หมายเลขพอร์ต	หน้าที่
ข้อมูล	Base+0	0x378	เขียน
สถานะ	Base+1	0x379	อ่าน
ควบคุม	Base+2	0x37a	อ่านและเขียน

2.5 โปรแกรม Remote Anything

โปรแกรม Remote Anything เป็นโปรแกรมสำหรับการควบคุมและจัดการการทำงานของเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายคอมพิวเตอร์ต่างๆ อาทิเช่น ระบบเครือข่ายระยะใกล้ (LAN – Local Area Network), ระบบเครือข่ายระยะไกล (WAN – Wide Area Network) และระบบเครือข่ายอินเทอร์เน็ต (Internet) ผ่านคู่สายโทรศัพท์ซึ่งโปรแกรม Remote Anything นี้ สามารถใช้งานกับระบบเครือข่ายดังกล่าวนี้ได้อย่างมีประสิทธิภาพ อีกทั้งการใช้งานโปรแกรมควบคุมเครื่องคอมพิวเตอร์ระยะไกลนี้สามารถนำมาใช้งานให้เกิดประโยชน์ในลักษณะงานประเภทต่างๆ ได้ไม่ว่าจะใช้งานในสถาบันการศึกษา, บริษัท, ห้างร้าน ที่มีระบบเครือข่าย เป็นต้น ประโยชน์ที่ว่านี้หมายถึง การนำโปรแกรมตัวนี้มาใช้ควบคุมเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ปลายทาง (Client) ได้ทั้งหมด ยกตัวอย่างเช่น การตรวจสอบรายละเอียดอุปกรณ์ของแต่ละเครื่อง (Hardware Inventory) จากเครื่องคอมพิวเตอร์ปลายทาง การโยกย้ายไฟล์ (Copy) จากเครื่องคอมพิวเตอร์ปลายทาง, การโยกย้ายไฟล์ (Transfer File) จากเครื่องคอมพิวเตอร์ปลายทาง การสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง (Shut Down) เป็นต้น ทั้งนี้อาจจะดูเหมือนเครื่องคอมพิวเตอร์ต้นทางนั้นมีสิทธิ์ทุกประการในการที่จะเข้าไปควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทางได้อย่างไม่มีข้อแม้ แต่ในทางกลับกันผู้ใช้เครื่องคอมพิวเตอร์ปลายทางก็สามารถกำหนดระบบรักษาความปลอดภัย เพื่อมิให้เครื่องคอมพิวเตอร์ต้นทางเข้ามาได้ เช่น การกำหนดรหัสผ่าน (Password) เป็นต้น

ลักษณะการทำงานของโปรแกรม Remote Anything จะมีโปรแกรมเล็กๆ 2 ตัว สำหรับติดตั้งให้กับเครื่องคอมพิวเตอร์ต้นทางชื่อโปรแกรมคือ Master.exe เพื่อเอาไว้ควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทางและเครื่องคอมพิวเตอร์ปลายทางจะต้องลงโปรแกรมที่ชื่อ Slave.exe เพื่อไว้สำหรับให้เครื่องคอมพิวเตอร์ต้นทางเข้ามาควบคุมและจัดการ ซึ่งสามารถลงโปรแกรม Slave.exe ได้ทุกเครื่องในระบบเครือข่าย (Network) โดยอาศัยภาษาหรือโปรโตคอล (Protocol) ที่คุยระหว่างกันคือ TCP/IP เป็นตัวกลางในการติดต่อสื่อสารซึ่งกันและกัน และใช้ Port หมายเลข 4000 เป็นช่องทางในการติดต่อสื่อสาร หลักการทำงานนั้นเครื่องคอมพิวเตอร์ต้นทางจะเปิดโปรแกรมที่ชื่อ Master.exe ขึ้นมาเพื่อเข้าไปดูรายละเอียดต่างๆ ของเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ โดยโปรแกรมจะตรวจสอบรหัสผ่านระหว่างเครื่องต้นทางและปลายทางให้ตรงกันก่อน จึงจะผ่านเข้าสู่เครื่องปลายทางได้ ซึ่งเมื่อผ่านเข้าไประบบแล้วเครื่องคอมพิวเตอร์ต้นทางสามารถควบคุมการทำงานและใช้คำสั่งต่างๆ จากเครื่องคอมพิวเตอร์ปลายทางได้ทันที

คุณสมบัติของเครื่องคอมพิวเตอร์ต้นทาง (Master.exe)

โปรแกรม Remote Anything นั้น มีโปรแกรมย่อยที่สำคัญ คือ โปรแกรมที่ใช้สำหรับเครื่องคอมพิวเตอร์ต้นทาง (Master.exe) เพื่อใช้ควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทาง ซึ่งมีคุณสมบัติต่างๆมากมาย สามารถอธิบายได้ดังนี้

- สามารถควบคุมการทำงานเครื่องคอมพิวเตอร์ได้หลายเครื่องพร้อมๆ กัน
- สามารถตรวจสอบเครื่องคอมพิวเตอร์ปลายทางที่อยู่ในระบบเครือข่ายได้ (Ping)
- สามารถดูการทำงานของเครื่องคอมพิวเตอร์ปลายทางได้ (Monitor Only)
- สามารถควบคุมการทำงานของเครื่องคอมพิวเตอร์ปลายทางได้ (Control)
- สามารถสั่งปิดเครื่องคอมพิวเตอร์ปลายทางได้ (Shut Down)
- สามารถสั่งปิด/เปิดเครื่องคอมพิวเตอร์ปลายทางได้ (Reboot)
- สามารถสั่งออกจากระบบเครื่องคอมพิวเตอร์ปลายทางได้ (Log Off)
- สามารถสั่งหยุดการทำงานทั้งหมดของเครื่องคอมพิวเตอร์ปลายทางได้ (Lock Up)
- สามารถสั่งยกเลิกโปรแกรม Slave.exe เครื่องคอมพิวเตอร์ปลายทางได้ (Uninstall)
- สามารถตรวจสอบรายละเอียดของฮาร์ดแวร์ (Hardware) เครื่องปลายทางได้
- สามารถเก็บบันทึกข้อมูลเครื่องคอมพิวเตอร์ปลายทางได้ (Address Book)
- สามารถพิมพ์ข้อความโต้ตอบกับเครื่องคอมพิวเตอร์ปลายทางได้ (SOS: Chat)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องคอมพิวเตอร์ต้นทางสามารถดึงข้อมูลเครื่องคอมพิวเตอร์ปลายทางได้ (File Transfer)
- สามารถเก็บประวัติการเข้าเครื่องปลายทางได้ (Log File)
- สามารถดูรหัสผ่านที่ซ่อนไว้ของเครื่องคอมพิวเตอร์ปลายทางได้

2.5.2 คุณสมบัติของเครื่องคอมพิวเตอร์ปลายทาง (Slave.exe)

ในส่วนของเครื่องคอมพิวเตอร์ปลายทาง (Slave.exe) มีคุณสมบัติของตัวโปรแกรมในการที่จะกำหนดสิทธิ์การเข้ามาควบคุม และการจัดการจากเครื่องคอมพิวเตอร์ต้นทางได้ซึ่งอธิบายได้ดังนี้

- สามารถที่จะตั้งรหัสผ่านสำหรับการเข้ามาของเครื่องคอมพิวเตอร์ต้นทางได้ (Security Password)
- สามารถตัดคุณสมบัติที่ไม่ต้องการให้เครื่องคอมพิวเตอร์ต้นทางทำได้ เช่น การดูดแฟ้มข้อมูล (File Transfer), หรือการสั่งปิดเครื่อง (Shut Down) จากเครื่องคอมพิวเตอร์ต้นทางได้ (Define Enabled Master Features)
- สามารถให้มีการแสดงข้อความเพื่อยอมรับ/ไม่ยอมรับ เมื่อเครื่องคอมพิวเตอร์ต้นทางติดต่อมา (Accept/Reject incoming connections manually)
- สามารถเก็บประวัติการเข้ามาของเครื่องคอมพิวเตอร์ต้นทางได้ (Log File)
- สามารถสั่งแสดง/ซ่อนไอคอนของ Slave.exe บนวินโดวส์ได้ (Tray Icon)
- สามารถสั่งให้แสดงหน้าจอตัวเอง ให้กับเครื่องคอมพิวเตอร์ต้นทางได้ (Show my Screen)
- สามารถส่งข้อความสนทนากับเครื่องคอมพิวเตอร์ต้นทางได้ (Chat)

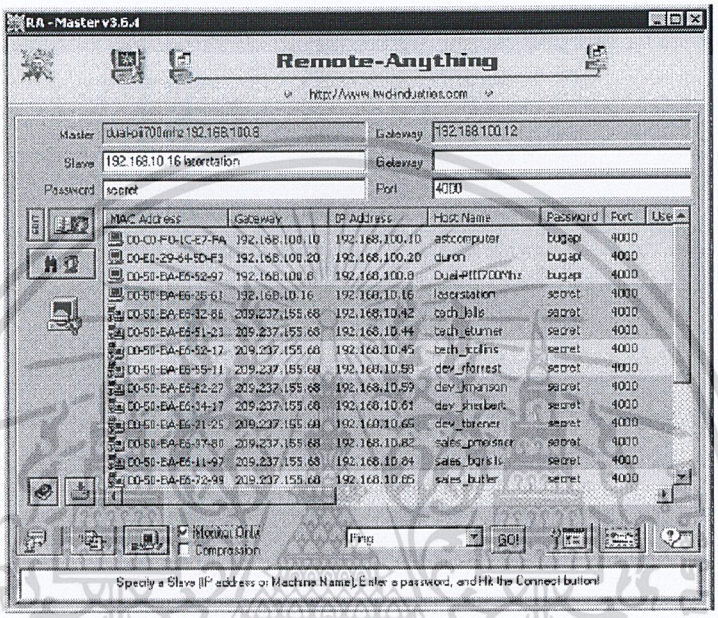
2.5.3 การใช้งานโปรแกรม Remote Anything ผ่านระบบเครือข่าย

ในการใช้งานโปรแกรม Remote Anything ผ่านระบบเครือข่ายคอมพิวเตอร์นั้น คุณสามารถใช้งานโปรแกรม Remote Anything นี้ผ่านระบบเครือข่ายได้ตั้งแต่ LAN, ADSL, WAN และ INTERNET ทั้งนี้ขึ้นอยู่กับประสิทธิภาพของระบบเครือข่ายนั้นๆ ว่ามีโครงข่ายเป็นอย่างไร ถ้านำโปรแกรม Remote Anything มาใช้ในระบบเครือข่ายภายใน (LAN) ก็จะสามารถใช้งานได้รวดเร็ว มีประสิทธิภาพสูง เพราะข้อมูลจะวิ่งผ่านสายสัญญาณเดียวกัน แต่ถ้านำโปรแกรม Remote Anything นี้ไปใช้งานในระบบเครือข่ายที่ใหญ่ขึ้นไม่ว่าจะเป็นระบบเครือข่ายระยะใกล้ (WAN) หรือระบบเครือข่ายอินเทอร์เน็ตผ่านสายโทรศัพท์ ความเร็วในการเข้าถึงข้อมูลนั้นก็จะแตกต่างกันไป ขึ้นอยู่กับอุปกรณ์และระบบโครงข่ายนั้นๆ แต่การใช้งานโปรแกรมนั้นยังใช้งานเหมือนเดิมทุกประการ ซึ่งจะขอยกตัวอย่างการใช้งานโปรแกรม Remote Anything ผ่านระบบเครือข่ายอินเทอร์เน็ตผ่านสายโทรศัพท์

อย่างที่กล่าวมาแล้วข้างต้นแล้ว ไม่ว่าจะใช้โปรแกรม Remote Anything ผ่านระบบเครือข่ายใดก็ตามลักษณะการใช้งานนั้นจะเหมือนกัน โดยผู้ที่ทำหน้าที่บริหารระบบ (Administrator) หรือเครื่องคอมพิวเตอร์ต้นทางที่เรียกว่า Master.exe จะต้องไปติดตั้งโปรแกรม Slave.exe ให้กับเครื่องคอมพิวเตอร์ปลายทางในทุกๆ เครื่องที่ต้องการจะควบคุม ซึ่งวิธีการติดตั้งโปรแกรม Slave.exe นั้น เพียงแค่ดับเบิลคลิกไฟล์ Slave.exe ลงไปในเครื่องคอมพิวเตอร์ปลายทาง โปรแกรมจะถูกนำไปติดตั้งทันที ในทางกลับกันเครื่องคอมพิวเตอร์ต้นทางจะต้องลงโปรแกรมที่ชื่อ Master.exe ไว้ที่เครื่องคอมพิวเตอร์ต้นทาง โดยการดับเบิลคลิกที่ไฟล์ Master.exe นั้นเช่นกันเพื่อไว้ควบคุมเครื่องคอมพิวเตอร์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่จะนำไฟล์ Salve.exe ไปติดตั้งที่เครื่องคอมพิวเตอร์ปลายทางนั้น ถ้าอยู่ระบบเครือข่ายเดียวกันอย่างเช่น ระบบเครือข่ายภายในก็สามารถนำไฟล์ใส่ดิสเก็ตต์เดินไปติดตั้งตามเครื่องคอมพิวเตอร์ปลายทางได้ทันที แต่ถ้าจะใช้งานผ่านระบบเครือข่ายอินเทอร์เน็ตก็คงยากสำหรับนำไฟล์นี้ไปติดตั้ง แต่การติดตั้งนั้นจะต้องอาศัยเครื่องมือที่ใช้สำหรับการติดต่อสื่อสารมาช่วย เช่น การส่งโปรแกรม Slave.exe ไปยังเครื่องคอมพิวเตอร์ปลายทางผ่านทางระบบจดหมายอิเล็กทรอนิกส์หรือผ่านทางโปรแกรม ICQ เป็นต้น ไปให้กับเครื่องคอมพิวเตอร์ปลายทางเพื่อให้ผู้ที่ได้รับไฟล์ Slave.exe นั้นนำไปติดตั้งต่อไป



รูปที่ 2.12 แสดงส่วนประกอบของโปรแกรม Remote Anything

2.5.4 ส่วนประกอบของโปรแกรม Remote Anything

1. Master คือ ชื่อและหมายเลข IP Address ของเครื่องคอมพิวเตอร์ต้นทาง พร้อมกับหมายเลข Gateway คือ ช่องทางการเข้าออกของเครื่องต้นทาง

2. Slave คือ ชื่อและหมายเลข IP Address ของเครื่องคอมพิวเตอร์ปลายทางที่ต้องการควบคุม พร้อมกับหมายเลข Gateway คือ ช่องทางการเข้าออกของเครื่องปลายทาง

3. Password คือ รหัสผ่านของเครื่องคอมพิวเตอร์ปลายทางที่ต้องการจัดการควบคุม

4. Port คือ หมายเลขช่องทางสำหรับการติดต่อสื่อสารกันผ่านเครือข่ายจะเป็นหมายเลข 4000

5. ปุ่ม Edit จะเป็นปุ่มสำหรับการเพิ่ม, ลบ, แก้ไข ชื่อเครื่องคอมพิวเตอร์ปลายทางลงในสมุดบันทึกโปรแกรม Remote Anything

6. ปุ่ม Address Book จะเป็นปุ่มสำหรับการดูรายละเอียดต่างๆ ของเครื่องคอมพิวเตอร์ปลายทางจากสมุดบันทึก

7. ปุ่ม Detect Slave จะเป็นปุ่มสำหรับตรวจสอบเครื่องปลายทางอัตโนมัติที่มีโปรแกรม Slave ไปติดตั้งไว้ และจะแสดงผลออกในเครื่องคอมพิวเตอร์ต้นทางทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ปุ่ม Manual จะเป็นปุ่มคู่มือสำหรับการใช้งานโปรแกรม Remote Anything (ภาษาอังกฤษ) ในลักษณะของไฟล์ Adobe Acrobat PDF File ซึ่งจะเชื่อมโยงไปยังเว็บไซต์ของผู้พัฒนาโปรแกรม **Remote Anything http: // www.twd–industries.com/prod_remote_doc_us.htm**

9. ปุ่ม Save Slave Text File จะเป็นปุ่มสำหรับการบันทึกข้อมูลรายละเอียดของเครื่องคอมพิวเตอร์ปลายทางทุกเครื่องที่แสดงอยู่บนเครื่องคอมพิวเตอร์ต้นทางในรูปแบบของ Text File

10. ปุ่ม Chat จะเป็นปุ่มสำหรับการสนทนาระหว่างเครื่องคอมพิวเตอร์ต้นทางและเครื่องคอมพิวเตอร์ปลายทางในระบบเครือข่ายเดียวกัน

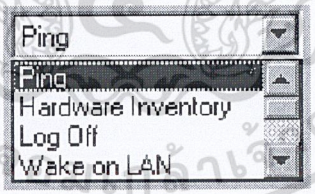
11. ปุ่ม File Browser to Transfer File จะเป็นปุ่มสำหรับการดูรายละเอียดของไฟล์ทั้งหมดที่เครื่องคอมพิวเตอร์ปลายทางในรูปแบบของ Explorer อีกทั้งสามารถโอนย้ายไฟล์ ลบไฟล์ และเปลี่ยนชื่อไฟล์ได้ทันที

12. ปุ่ม Remote Control จะเป็นปุ่มสำหรับการควบคุมเครื่องคอมพิวเตอร์ปลายทางซึ่งเครื่องคอมพิวเตอร์ต้นทางสามารถเห็นหน้าจอของเครื่องคอมพิวเตอร์ปลายทางได้ทุกอย่าง เสมือนไปนั่งอยู่ที่หน้าเครื่องคอมพิวเตอร์ปลายทาง อีกทั้งสามารถใช้เมาส์ควบคุมการทำงานได้อีกด้วย

13. ปุ่ม Monitor Only จะเป็นปุ่มสำหรับดูรายละเอียดบนหน้าจอเครื่องคอมพิวเตอร์ปลายทางที่มีการเคลื่อนไหวตลอดเวลา แต่ไม่สามารถควบคุมและจัดการได้

14. ปุ่ม Compression การบีบขนาดเพื่อรักษาความกว้างของช่องสัญญาณ

15. ปุ่มคำสั่งพื้นฐาน จะเป็นปุ่มคำสั่งต่างๆ ที่ใช้สำหรับควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทางประกอบไปด้วย เช่น การสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง การดูรายละเอียดฮาร์ดแวร์เครื่องคอมพิวเตอร์ปลายทาง การเปิดเครื่องคอมพิวเตอร์ปลายทาง เป็นต้น
รายละเอียดคำสั่งมีดังนี้



รูปที่ 2.13 แสดงปุ่มคำสั่งพื้นฐาน

- Ping
- Hardware Inventory
- Log Off
- Wake on LAN
- Shut Down
- Reboot
- Lock up
- Uninstall

16. ปุ่ม Option จะเป็นปุ่มสำหรับตั้งค่าต่างๆ ที่จะใช้ควบคุมเครื่องคอมพิวเตอร์ปลายทาง

17. ปุ่ม Licenses จะเป็นปุ่มใบอนุญาตสำหรับการใช้งานโปรแกรม Remote Anything

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

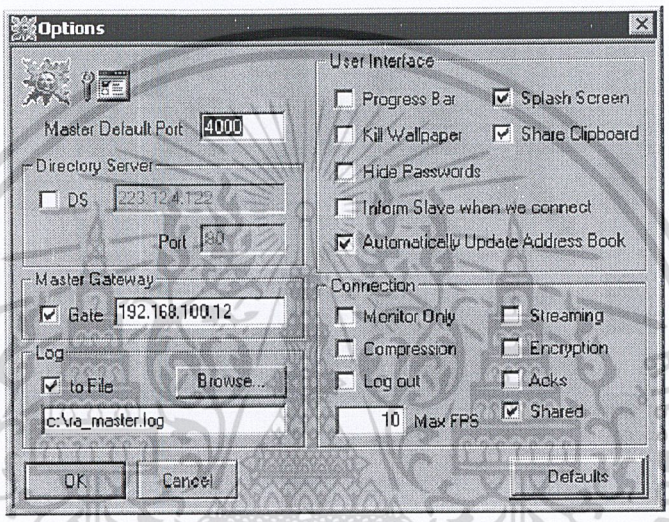
18. ปุ่ม About จะเป็นปุ่มสำหรับรายละเอียดเกี่ยวกับโปรแกรม Remote Anything

19. รายละเอียดสำหรับแสดงเครื่องคอมพิวเตอร์ปลายทาง เป็นที่สำหรับแสดงรายละเอียดของเครื่องคอมพิวเตอร์ปลายทางทั้งหมดที่อยู่ในระบบเครือข่าย

2.5.5 การตั้งค่าตัวเลือกให้กับโปรแกรม Remote Anything

การติดตั้งค่าตัวเลือกต่างๆ ให้กับเครื่องคอมพิวเตอร์ต้นทาง (Master.exe) สำหรับการใช้ควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทางต่อไป

- กดปุ่ม Option บนโปรแกรม Remote Anything
- หน้าจอของ RA Options จะปรากฏขึ้นมา ซึ่งมีรายละเอียดต่างๆ ดังนี้



รูปที่ 2.14 แสดงการตั้งค่าตัวเลือกให้กับโปรแกรม Remote Anything

2.5.6 การตรวจสอบหมายเลขอินเทอร์เน็ตของเครื่องคอมพิวเตอร์ปลายทาง

ก่อนการใช้โปรแกรม Remote Anything เพื่อให้เครื่องคอมพิวเตอร์ต้นทาง (Master.exe) ไปจัดการและควบคุมเครื่องคอมพิวเตอร์ปลายทาง (Slave.exe) นั้นสิ่งที่สำคัญที่สุดในการเข้าไปควบคุมเครื่องคอมพิวเตอร์ปลายทางก็คือ จะต้องรู้หมายเลขอินเทอร์เน็ตของเครื่องปลายทาง (IP Address) ซึ่งถ้าการควบคุมนั้นผ่านระบบเครือข่ายภายใน (LAN, WAN) ผู้บริหารระบบ (Administrator) นั้นสามารถจะรู้หมายเลขของเครื่องอินเทอร์เน็ตปลายทางได้อยู่แล้ว อีกทั้งตัวโปรแกรม Remote Anything ยังมีปุ่ม Detect available Slave on you LAN ไว้สำหรับตรวจสอบพบแล้วจะแสดงหมายเลขอินเทอร์เน็ตของเครื่องที่มีตัว Slave.exe ฝังอยู่แสดงขึ้นมาบนหน้าจอทุกเครื่อง

ในทางกลับกัน ถ้าการควบคุมนั้นผ่านระบบเครือข่ายอินเทอร์เน็ต การที่จะสามารถรู้หมายเลขอินเทอร์เน็ตของเครื่องปลายทางนั้น เป็นสิ่งที่ลำบากกว่าเพราะหมายเลขอินเทอร์เน็ตที่ต่อผ่านโมเด็มนั้นจะถูกเปลี่ยนแปลงไปทุกครั้ง (IP DYNAMIC) แต่สามารถทราบหมายเลขอินเทอร์เน็ตดังกล่าวได้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาศัยเทคนิควิธีการ โดยที่เครื่องคอมพิวเตอร์ปลายทางนั้นจะต้องใช้คำสั่ง WINIPCFG ในการที่จะแจ้งหมายเลขอินเทอร์เน็ต (IP Address) ให้กับเครื่องคอมพิวเตอร์ต้นทางได้ทราบ มีวิธีการใช้งานคือ

ขั้นตอนการทำงาน

- กดปุ่ม **Start** เลือกรายการ **Run**
- พิมพ์คำสั่งว่า **winipcfg** แล้วกดปุ่ม **OK**
- หน้าจอจะแสดงรายการเกี่ยวกับหมายเลขอินเทอร์เน็ต (IP Address) ของเครื่องคอมพิวเตอร์เครื่องนั้นออกมา ซึ่งจะต้องนำหมายเลขที่ได้นี้ไปใช้ร่วมกับโปรแกรม Master.exe เพื่อเข้าไปควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทาง

2.5.7 การควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทาง (Master to Slave)

1. การดูสถานภาพของเครื่องคอมพิวเตอร์ปลายทาง (PING) บน Windows

การดูสถานภาพของเครื่องคอมพิวเตอร์ปลายทางนั้น เพื่อจะดูว่าเครื่องคอมพิวเตอร์ปลายทางนั้นอยู่ในระบบเครือข่ายที่เครื่องคอมพิวเตอร์ต้นทางสามารถติดต่อได้หรือไม่ เหมือนกับการใช้คำสั่ง C:\>PING บนระบบปฏิบัติการวินโดวส์ เป็นต้น เช่น ต้องการทราบว่าเครื่องคอมพิวเตอร์ของสามารถเห็นหมายเลขอินเทอร์เน็ตของเครื่องคอมพิวเตอร์ที่ต้องการได้หรือไม่ และอยู่ภายในระบบเดียวกันหรือไม่ สามารถทำได้โดยใช้คำสั่งดังนี้

ขั้นตอนการทำงาน

- กดปุ่ม Start เรียกรายการที่ชื่อ Run
- พิมพ์คำสั่ง c:>ping 202.183.153.184 หรือหมายเลขที่ต้องการทดสอบ
- หน้าจอจะแสดงรายละเอียดการ ping ไปยังเครื่องคอมพิวเตอร์ปลายทาง
- ในกรณีถ้าใช้ผ่านโปรแกรม Remote Anything สามารถทำได้ดังนี้
- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Ping
- กดปุ่ม GO!

สังเกตได้ว่าถ้าเครื่องคอมพิวเตอร์ต้นทางสามารถติดต่อกับเครื่องคอมพิวเตอร์ปลายทางได้ รูปหน้าจอของเครื่องคอมพิวเตอร์ปลายทางนั้นจะเป็นสีฟ้า

2. การดูรายละเอียดของเครื่องคอมพิวเตอร์ปลายทาง (Hardware Inventory)

Hardware Inventory จะเป็นการดูรายละเอียดของเครื่องคอมพิวเตอร์ปลายทางว่ามีรายละเอียดเกี่ยวกับเครื่องคอมพิวเตอร์ปลายทางนั้นประกอบไปด้วยอะไรบ้าง เช่น หมายเลขอินเทอร์เน็ต (IP Address), ชื่อเครื่องคอมพิวเตอร์ปลายทาง (Host Name) และอื่นๆ สามารถทำได้ดังนี้คือ

ขั้นตอนการทำงาน

- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Hardware Inventory
- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- กดปุ่ม GO! เพื่อดูรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การสั่งให้ออกจากระบบวินโดวส์ (Log Off)

การสั่งให้ออกจากระบบวินโดวส์ (Log Off) กล่าวคือเมื่อต้องการให้ระบบวินโดวส์ของเครื่องคอมพิวเตอร์ปลายทางมีการ Log On สำหรับผู้ใช้คนใหม่สามารถสั่งการจากเครื่องคอมพิวเตอร์ต้นทางได้ดังนี้

ขั้นตอนการทำงาน

- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Log Off
- กดปุ่ม GO!
- โปรแกรมจะสอบถามต่อไปว่า Log Off selected Slave PC(s)? ให้คุณกดปุ่ม Yes เมื่อต้องการจะให้เครื่องคอมพิวเตอร์ปลายทางเข้าสู่ระบบ Log Off เครื่องคอมพิวเตอร์ปลายทางจะทำการ Log Off หน้าจอวินโดวส์ทันที ซึ่งจะมีการให้ใส่ Windows Log On อีกครั้ง
- ถ้ามองภาพในฝั่งเครื่องคอมพิวเตอร์ปลายทาง หน้าจอของวินโดวส์เมื่อถูกคำสั่ง Log Off แล้วจะปรากฏหน้าจอให้ Log On ใหม่ดังนี้

4. การสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง (Shut Down)

เครื่องคอมพิวเตอร์ต้นทาง สามารถสั่งปิด (Shut Down) เครื่องคอมพิวเตอร์ปลายทางได้ ซึ่งเป็นผลทำให้เครื่องคอมพิวเตอร์ปลายทางนั้นปิดตัวเองทันที มีขั้นตอนดังนี้

ขั้นตอนการทำงาน

- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Shut Down
- กดปุ่ม GO!
- โปรแกรมจะสอบถามต่อไปว่า Shut Down selected Slave PC(s)? ให้คุณกดปุ่ม Yes เมื่อต้องการจะให้ปิดเครื่องคอมพิวเตอร์ปลายทางลงทันที ซึ่งหลังจากนั้นเครื่องคอมพิวเตอร์ปลายทางจะถูกปิดทันทีเช่นกัน

5. การสั่งเปิดเครื่องคอมพิวเตอร์ปลายทาง (Reboot)

การสั่งเปิดเครื่องคอมพิวเตอร์ปลายทางนั้นจะสามารถกระทำได้ต่อเมื่อเครื่องคอมพิวเตอร์ต้นทางนั้นเปิดไว้อยู่ก่อนแล้ว แล้วสามารถสั่งปิดและเปิดเครื่องคอมพิวเตอร์ปลายทางนั้นใหม่ทันที เหมือนกับกดปุ่ม Restart เครื่องคอมพิวเตอร์ทั่วไปซึ่งสามารถทำได้ดังนี้

ขั้นตอนการทำงาน

- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Reboot
- กดปุ่ม GO!
- โปรแกรมจะสอบถามต่อไปว่า Reboot selected Slave PC(s)? ให้คุณกดปุ่ม Yes เมื่อต้องการจะให้เครื่องคอมพิวเตอร์ปลายทางที่เปิดอยู่นั้นปิดและเปิดขึ้นมาใหม่อีกครั้ง

6. การสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง (Lock Up)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง Lock Up นั้นจะเหมือนกับการสั่งปิดเครื่องคอมพิวเตอร์ปลายทาง Shut Down แต่การ Lock Up นั้นจะเป็นการปิดเครื่องคอมพิวเตอร์ทันที ต่างกับการ Shut Down ที่เป็นการปิดอย่างเป็นระบบ ซึ่งการปิดเครื่องคอมพิวเตอร์ปลายทางแบบ Lock Up สามารถทำได้ดังนี้

ขั้นตอนการทำงาน

- ให้ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการ
- ให้มาเลือกรายการที่ Pull Down Menu ด้านล่างให้เลือกเป็นชื่อ Lock Up
- กดปุ่ม GO!
- โปรแกรมจะสอบถามต่อไปว่า Lock Up selected Slave PC(s)? ให้คุณกดปุ่ม Yes เมื่อต้องการจะปิดเครื่องคอมพิวเตอร์ปลายทางนั้นแบบทันที

7. **การเข้าสู่หน้าจอเครื่องคอมพิวเตอร์ปลายทาง (Monitor & Control)**

การเข้าสู่หน้าจอเครื่องคอมพิวเตอร์ปลายทาง วิธีการนี้เครื่องคอมพิวเตอร์ต้นทางนั้น (Master.exe) จะสามารถเข้าไปดูหน้าจอของเครื่องคอมพิวเตอร์ปลายทางได้ทั้งหมด (Monitor) แบบ Real Time กล่าวคือถ้าเครื่องคอมพิวเตอร์ปลายทางมีการใช้งานโปรแกรมใดๆ อยู่ก็ตาม เครื่องคอมพิวเตอร์ต้นทางก็เห็นเช่นเดียวกับเครื่องคอมพิวเตอร์ปลายทางทุกประการ

นอกจากจะเข้าไปดูการทำงานของเครื่องคอมพิวเตอร์ปลายทางได้แล้ว ยังสามารถควบคุมการทำงานของเครื่องคอมพิวเตอร์ปลายทางได้อีก (Slave Control) สามารถขยับเมาส์บนหน้าจอของเครื่องคอมพิวเตอร์ปลายทางได้เหมือนกับไปนั่งอยู่ที่เครื่องคอมพิวเตอร์ปลายทาง อีกทั้งยังใช้คำสั่งต่างๆ ของโปรแกรม Remote Anything เป็นตัวจัดการ (Manage) กับเครื่องคอมพิวเตอร์ปลายทางได้ อาทิเช่น สั่งปิด-เปิดเครื่องคอมพิวเตอร์ เป็นต้น

เครื่องคอมพิวเตอร์ต้นทางสามารถเลือกที่จะเข้าไปดูหน้าจอเครื่องคอมพิวเตอร์ปลายทางเพียงอย่างเดียว (Monitor Only) หรือเลือกที่จะเข้าไปควบคุมและจัดการกับเครื่องคอมพิวเตอร์ปลายทางได้ ซึ่งมีวิธีการดังนี้

ในกรณีที่ต้องการเข้าไปดูหน้าจอการทำงานของเครื่องคอมพิวเตอร์ปลายทางอย่างเดียว (Monitor Only)

- ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการเข้าไปจัดการแฟ้มข้อมูล โดยเลือกคลิกเครื่องหมายถูกที่ปุ่ม Monitor Only นั้น กดปุ่ม Remote Control the Selected Slave PC

ในกรณีที่ต้องการเข้าไปดูหน้าจอและควบคุมจัดการทำงานของเครื่องคอมพิวเตอร์ปลายทาง (Monitor & Control)

- ใช้เมาส์เลือกเครื่องคอมพิวเตอร์ปลายทางที่ต้องการเข้าไปจัดการแฟ้มข้อมูลนั้นแล้วกดปุ่ม Remote Control the Selected Slave PC
- เมื่อคลิกที่ปุ่ม Remote Control แล้ว โปรแกรมจะแสดงข้อความว่า Remote-Anything is trying to establish the connection with the distant PC (time out 1 minute) หมายถึงการพยายามเชื่อมต่อเข้าสู่เครื่องคอมพิวเตอร์ปลายทาง ซึ่งจะใช้เวลา 1 นาที ถ้าไม่สำเร็จระบบถึงจะตัดการเชื่อมต่อโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อเข้าสู่ระบบแล้ว หน้าจอเครื่องคอมพิวเตอร์ของคุณจะเป็นสีน้ำเงิน พร้อมกับมีข้อความว่า Building - View หมายถึง โปรแกรมกำลังจะแสดงภาพหน้าจอของเครื่องคอมพิวเตอร์ปลายทาง
- หน้าจอเครื่องคอมพิวเตอร์ปลายทางจะปรากฏขึ้นมาทันที ซึ่งคุณสามารถดูพฤติกรรมการใช้งานหรือสามารถควบคุมและจัดการกับเครื่องคอมพิวเตอร์ปลายทางนั้นได้ทันทีเช่นเดียวกัน ตัวอย่างเช่น การเข้าไปเปลี่ยนภาพพื้นหลัง (Background) สามารถใช้เมาส์ควบคุมในการเปลี่ยนได้ทันที หรือไม่ว่าจะเป็นการเรียกดูข้อมูลต่างๆ ที่อยู่ในเครื่องคอมพิวเตอร์ปลายทางมาแสดงได้ และสามารถเลือกลบแฟ้มข้อมูลได้ เป็นต้น

2.5.8 เมนูลัดสำหรับการควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทาง (Master Menu Features)

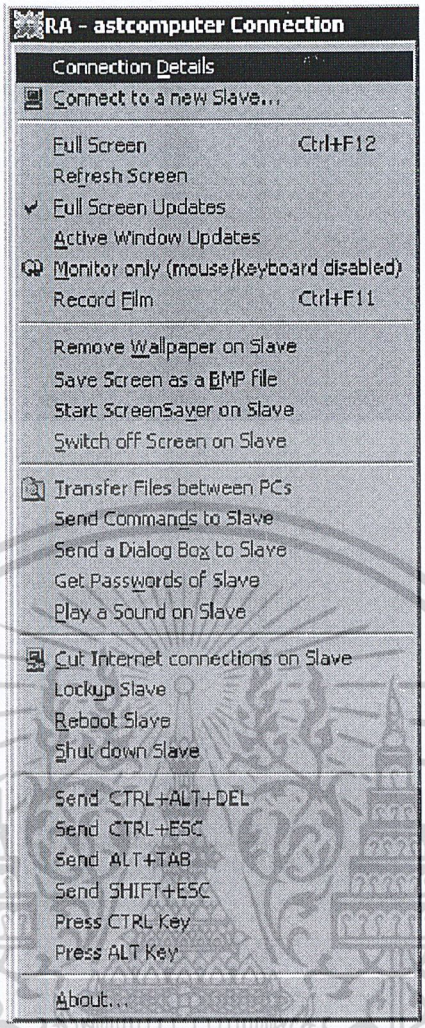
นอกจากการใช้งานแบบปกติแล้ว โปรแกรม Remote Anything ยังมีเมนูลัดเพื่อให้เครื่องคอมพิวเตอร์ต้นทางเลือกใช้คำสั่งต่างๆ กับเครื่องคอมพิวเตอร์ปลายทางกำลังเชื่อมต่อยู่ได้โดยสามารถทำได้ดังนี้

ขั้นตอนการทำงาน

คลิกเมาส์ด้านขวาตรงมุมด้านซ้ายมือของหน้าจอเครื่องคอมพิวเตอร์ปลายทางเครื่องนั้นจะปรากฏรายการคำสั่งต่าง ๆ ขึ้น ซึ่งมีรายละเอียดของคำสั่งดังนี้

- **Connection Details** หมายถึง โปรแกรมจะแสดงรายละเอียดของการเชื่อมต่อ
- **Connect to a new Slave** หมายถึง ถ้าต้องการจะเชื่อมต่อกับเครื่องคอมพิวเตอร์ปลายทางเครื่องใหม่
- **Full Screen หรือปุ่ม Ctrl+F12** หมายถึง การแสดงภาพของเครื่องคอมพิวเตอร์ปลายทางแบบเต็มหน้าจอเสมือนยกเครื่องคอมพิวเตอร์ปลายทางมาไว้ที่เครื่องคอมพิวเตอร์ต้นทาง
- **Refresh Screen** หมายถึง การทำให้หน้าจอของเครื่องคอมพิวเตอร์ปลายทางนั้นเป็นปัจจุบัน ใช้เมื่อถ้าหน้าจอของเครื่องคอมพิวเตอร์ปลายทางนั้นค้างอยู่
- **Full Screen Updates** หมายถึง การแสดงภาพของเครื่องคอมพิวเตอร์ปลายทางแบบเต็มหน้าจอและแสดงภาพที่เป็นปัจจุบัน
- **Active Window Updates** หมายถึง การแสดงภาพของเครื่องคอมพิวเตอร์ปลายทางที่เลือกนั้นให้แสดงข้อมูลที่เป็นปัจจุบัน
- **Monitor Only (mouse/keyboard disabled)** หมายถึง การเปลี่ยนสถานะให้เครื่องคอมพิวเตอร์ต้นทางดูการทำงานของเครื่องคอมพิวเตอร์ปลายทางได้เพียงอย่างเดียว ไม่สามารถควบคุมและจัดการเครื่องคอมพิวเตอร์ปลายทางได้
- **Remove Wallpaper on Slave** หมายถึง การตัด Wallpaper ที่อยู่บนเครื่องคอมพิวเตอร์ปลายทางเพื่อเพิ่มความเร็วในการทำงาน ใช้ได้เฉพาะ Windows9X เท่านั้น
- **Save Screen as a BMP file** หมายถึง การคัดลอกภาพบนหน้าจอของเครื่องคอมพิวเตอร์ปลายทางและบันทึกไฟล์ในรูปแบบแฟ้มข้อมูลแบบ BMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงเมนูลัดของเครื่องคอมพิวเตอร์ต้นทางเลือกใช้คำสั่งต่างๆ กับเครื่องคอมพิวเตอร์ปลายทางกำลังเชื่อมต่อ

- **Start Screen Saver on Slave** หมายถึง การสั่งให้เครื่องคอมพิวเตอร์ปลายทางแสดง Screen Saver ขึ้นมา
- **Switch off Screen on Slave** หมายถึง การป้องกันไม่ให้บุคคลอื่นเข้ามาทำงานบนเครื่องคอมพิวเตอร์ปลายทาง สามารถใช้เฉพาะ Windows9X เท่านั้น
- **Transfer Files between PCs** หมายถึง การเรียกโปรแกรมแสดงโครงสร้างของเครื่องคอมพิวเตอร์ปลายทางขึ้นมา
- **Send Commands to Slave** หมายถึง การส่งคำสั่งให้กับเครื่องคอมพิวเตอร์ปลายทาง
- **Send Dialog Box to Slave** หมายถึง การส่งข้อความต่าง ๆ ให้กับเครื่องคอมพิวเตอร์ปลายทาง ให้พิมพ์ข้อความที่ต้องการลงในช่องว่าง เมื่อเรียบร้อยแล้วกดปุ่ม OK เครื่องคอมพิวเตอร์ปลายทางก็จะได้รับข้อความที่เครื่องคอมพิวเตอร์ต้นทางพิมพ์ไปให้ทันที
- **Get Passwords of Slave** หมายถึง โปรแกรมจะทำการเรียกรหัสผ่านที่ซ่อนอยู่ทั้งหมดที่ใช้กับเครื่องคอมพิวเตอร์ปลายทางขึ้นมาแสดงให้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Play a Sound on Slave** หมายถึง การสั่งให้เสียงดังขึ้นที่เครื่องคอมพิวเตอร์ปลายทาง 1 ครั้ง
- **Cut Internet connections on Slave** หมายถึง การตัดบริการอินเทอร์เน็ตทั้งหมดของเครื่องคอมพิวเตอร์ปลายทาง ซึ่งจะมีหน้าจอแสดงขึ้นมาแสดงรายละเอียดการตัดการใช้บริการอินเทอร์เน็ตโดยประกอบไปด้วย HTTP, FTP, EMAIZ และอื่นๆ เมื่อต้องการยืนยันให้ทำงานกดปุ่ม Yes
- **Lockup Salve** หมายถึง การสั่งให้เครื่องคอมพิวเตอร์ปลายทางหยุดค้างไม่สามารถทำอะไรได้นอกจากจะกดปุ่ม Reboot ใหม่อีกครั้ง ซึ่งจะมีหน้าจอให้ยืนยันการ Lockup Slave เมื่อต้องการยืนยันให้ทำงานกดปุ่ม Yes
- **Reboot Slave** หมายถึง การสั่งให้เครื่องคอมพิวเตอร์ปลายทางทำการปิดและทำการเปิดเครื่องขึ้นมาใหม่ เมื่อต้องการยืนยันให้ทำงานกดปุ่ม Yes
- **Shut down Slave** หมายถึง การสั่งให้เครื่องคอมพิวเตอร์ปลายทางปิดเครื่องทันที เมื่อต้องการยืนยันให้ทำงานกดปุ่ม Yes

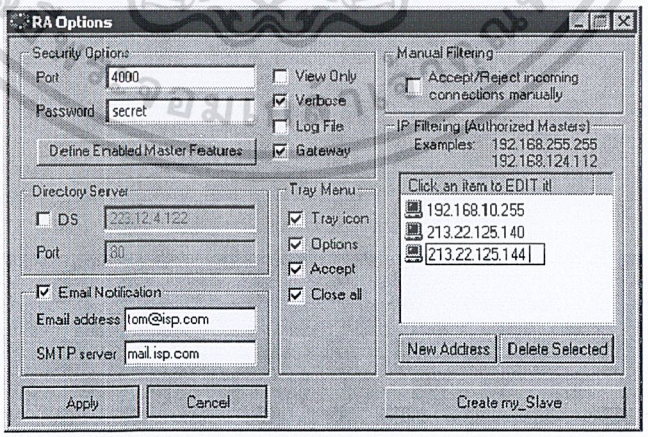
2.5.9 การใช้งานโปรแกรม Remote Anything สำหรับเครื่องคอมพิวเตอร์ปลายทาง (Slave.exe)

1. การตั้งติดค่าต่างๆ สำหรับเครื่องคอมพิวเตอร์ปลายทาง (Security Options)

การตั้งติดค่าต่างๆ สำหรับเครื่องคอมพิวเตอร์ปลายทางนั้น ก็เพื่อให้โปรแกรม Slave.exe ทำงานตามคำสั่งของผู้ใช้งานเครื่องคอมพิวเตอร์ปลายทางได้ไม่ว่าจะเป็นกำหนดสิทธิ์ต่างๆ ในการเข้ามาที่เครื่องคอมพิวเตอร์ปลายทางของเครื่องคอมพิวเตอร์ต้นทาง ซึ่งมีอยู่หลายคำสั่ง อาทิ กำหนดรหัสผ่าน(Password) เพื่อมิให้เครื่องคอมพิวเตอร์ต้นทางเข้ามาถึงได้ เป็นต้น ซึ่งยังมีรายละเอียดอื่นๆ อีกดังนี้

ขั้นตอนการทำงาน

- ให้คลิกเมาส์ด้านขวาตรงวงกลมสีแดง ด้านล่างจอภาพ (Task Bar) ซึ่งจะมีรายละเอียดแสดงขึ้นมา ให้เลือกรายการ Option...
- รายละเอียด RA Options จะปรากฏขึ้นมาเพื่อให้คุณสามารถเปลี่ยนแปลงแก้ไขค่าต่างๆ ได้ดังนี้



รูปที่ 2.16 แสดงรายละเอียด RA Options

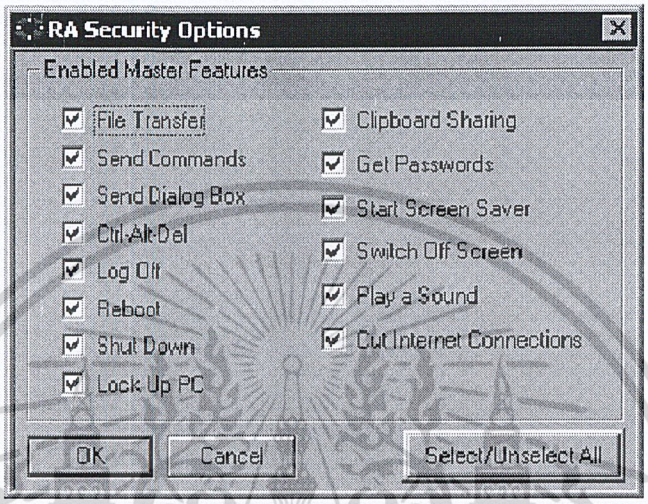
2. ตัวเลือกระบบรักษาความปลอดภัย (Security Options)

ตัวเลือกมีไว้สำหรับให้เครื่องคอมพิวเตอร์ปลายทางกำหนดสิทธิ์ในการเข้ามาของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ต้นทางได้ตัวอย่างเช่น เปลี่ยนช่องทางการติดต่อสื่อสาร (Port) หรือเปลี่ยนรหัสผ่านใหม่ (Password) เพื่อป้องกันมิให้เครื่องคอมพิวเตอร์ต้นทางเข้ามาในเครื่องคอมพิวเตอร์ปลายทาง เป็นต้น

- การกำหนดสิทธิ์ในการใช้คำสั่งต่างๆ กับเครื่องคอมพิวเตอร์ปลายทาง (Define Enabled Master Features)
- รายละเอียดเกี่ยวกับสิทธิ์ต่างๆ จะปรากฏขึ้นดังนี้



รูปที่ 2.17 แสดงรายละเอียดของคำสั่ง Security Options

รายการทั้งหมดข้างต้น ถ้าไม่ต้องการให้เครื่องคอมพิวเตอร์ต้นทางนั้นกระทำอย่างไร ก็ให้ยกเลิกสิทธิ์ตรงนั้นออกไปโดยเอาเครื่องหมายถูกออกไป แล้วกด OK เมื่อเครื่องคอมพิวเตอร์ต้นทางมีการเข้ามาในเครื่องคอมพิวเตอร์ปลายทางได้แล้ว จะไม่สามารถใช้คำสั่ง File Transfer กับเครื่องคอมพิวเตอร์ปลายทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การคำนวณและการสร้าง

3.1 แนวคิดและหลักการทำงาน

จุดประสงค์ของการออกแบบระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์ (Telemonitor System Via Telephone Line) ก็เพื่อนำภาพจากกล้องโทรทัศน์วงจรปิด ส่งผ่านคู่สายโทรศัพท์ไปแสดงผลยังเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ได้ โดยการออกแบบจะทำการแปลงสัญญาณภาพที่ได้จากกล้องวงจรปิด ซึ่งเป็นสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอลก่อนเพื่อทำการส่งไปในเครือข่ายระบบโทรศัพท์ เนื่องจากการส่งข้อมูลในรูปแบบของสัญญาณดิจิตอลจะมีความผิดพลาดของข้อมูลในการส่งน้อยกว่าการส่งในรูปแบบสัญญาณอนาลอก ซึ่งสัญญาณที่แปลงเป็นสัญญาณดิจิตอลแล้วใน 1 จุดภาพ (pixel) จะมีขนาด 8 บิต เพราะฉะนั้นในการแสดงภาพ ภาพจะแสดงเพียง 1 ฟิลด์ต่อ 1 เฟรม โดยจะแสดง 256 จุด ต่อ 1 เส้นสแกน และใน 1 ภาพจะแสดงด้วย 256 เส้นสแกน ซึ่งจะทำให้ใน 1 ภาพจะมีข้อมูลทั้งหมด $256 \times 256 \times 8 =$ 52,428 บิต หรือ 64 กิโลไบต์

3.2 หลักการทำงาน

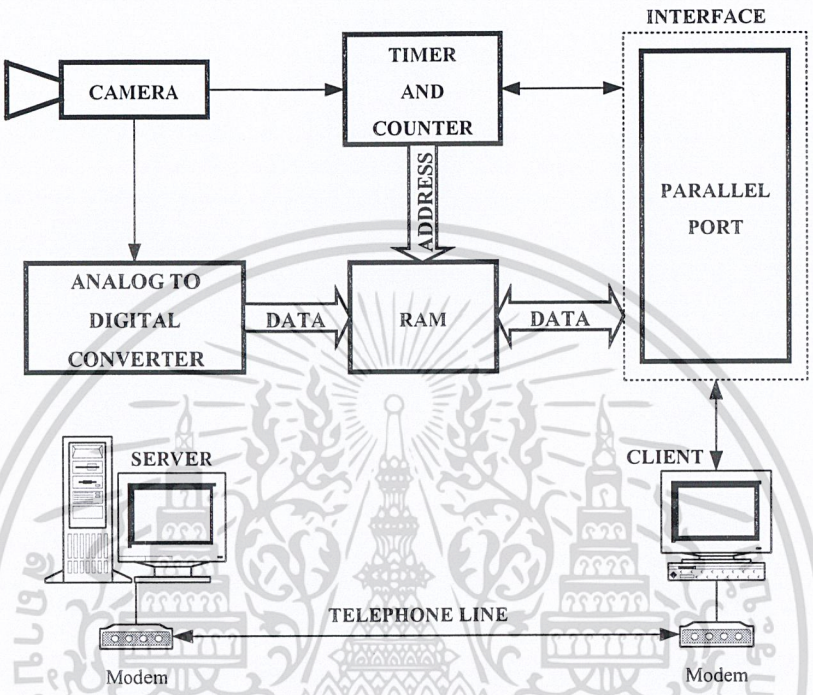
จากรูปที่ 3.1 เมื่อเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ (หรือเซิร์ฟเวอร์) ทำการติดต่อมายังเครื่องคอมพิวเตอร์ไคลเอนท์ (หรือไคลเอนท์) โดยใช้โปรแกรม Remote Anything (RA) เพื่อที่จะทำการควบคุมการทำงานของไคลเอนท์ทั้งหมดและเมื่อเซิร์ฟเวอร์ทำการควบคุมการทำงานของไคลเอนท์ได้แล้ว ก็จะสามารถที่จะเห็นภาพที่ปรากฏที่หน้าจอมอนิเตอร์ของไคลเอนท์ได้ รวมถึงสามารถควบคุมการทำงานคีย์บอร์ดและเมาส์ของไคลเอนท์ได้ด้วยทำให้เซิร์ฟเวอร์สามารถที่จะดูข้อมูล หรือใช้โปรแกรมต่างๆ ที่เก็บไว้ในไคลเอนท์ได้และเมื่อเครื่องคอมพิวเตอร์ทำการเข้าสู่โปรแกรมจับภาพก็จะเกิดเป็นระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์เมื่อเซิร์ฟเวอร์เข้าสู่โปรแกรมจับภาพ โปรแกรมก็จะทำการสั่งฮาร์ดแวร์ให้ทำการเขียนข้อมูลลงในหน่วยความจำแล้วทำการแสดงข้อมูลออกมาที่หน้าจอมอนิเตอร์ ซึ่งการติดต่อระหว่างฮาร์ดแวร์และโปรแกรมจะทำการติดต่อกันผ่านพอร์ตขนาน

การเขียนข้อมูลลงในหน่วยความจำจะเริ่มจากการนำสัญญาณเอาต์พุตที่ออกจากกล้องวงจรปิด ซึ่งเรียกว่า สัญญาณภาพรวมนำไปเข้าวงจร 2 ภาค โดยภาคแรกก็คือ ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลซึ่งจะให้สัญญาณเอาต์พุตออกมาเป็นสัญญาณดิจิตอลของสัญญาณภาพจากนั้นก็ทำการส่งเข้าไปเก็บในหน่วยความจำ ส่วนอีกภาคที่นำสัญญาณภาพรวมไปป้อนคือ ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำเพื่อที่จะนำไปสร้างสัญญาณชี้แอดเดรสหน่วยความจำที่สามารถชี้แอดเดรสได้สัมพันธ์กับสัญญาณภาพที่นำไปเก็บที่หน่วยความจำเพื่อไม่ให้เกิดภาพล้มเมื่อทำการแปลงข้อมูลกลับมาเป็นสัญญาณภาพ ซึ่งกระบวนการเขียนข้อมูลภาพจะดำเนินไปจนกระทั่งทำการเขียนข้อมูลภาพครบ 1 เฟรม (เต็มหน่วยความจำ)

เมื่อฮาร์ดแวร์ทำการเขียนข้อมูลครบ 1 เฟรมก็จะส่งสัญญาณไปแจ้งให้เครื่องคอมพิวเตอร์ทราบ และหลังจากนั้นเครื่องคอมพิวเตอร์จะสั่งการให้ฮาร์ดแวร์หยุดทำการเขียนข้อมูลและทำการอ่านข้อมูลมาแสดงที่หน้าจอมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจากหน่วยความจำเข้ามาแสดงที่หน้าจอนี้จะมีกระบวนการคือ โปรแกรมจะทำการอ่านข้อมูลเข้ามาครั้งละ 1 จุดภาพโดยการใช้สัญญาณนาฬิกาที่สร้างจากเครื่องคอมพิวเตอร์ไคลเอนท์มาทำการชี้แอดเดรสข้อมูลของหน่วยความจำและทำการอ่านข้อมูลเข้ามา 1 จุดภาพหลังจากนั้นจึงทำการส่งสัญญาณนาฬิกาเพื่อทำการอ่านข้อมูลในแอดเดรสต่อไปจนกระทั่งครบ 1 เฟรม



รูปที่ 3.1 แสดง Block diagram การทำงานของระบบ

3.3 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

3.3.1 วงจรกำเนิดความถี่

วงจรกำเนิดความถี่ เป็นวงจรที่ใช้กำเนิดความถี่เพื่อใช้ในการแซมปลิ้งสัญญาณภาพโดยสัญญาณภาพที่ออกจากกล้องวงจรปิดจะเป็นสัญญาณภาพรวม ซึ่งมีความถี่สูงสุดประมาณ 5 MHz ดังนั้นจึงออกแบบโดยใช้ความถี่ 15 MHz ในการแซมปลิ้งสัญญาณภาพเพื่อให้เป็นไปตามทฤษฎีของไนควิสท์

วงจรกำเนิดความถี่จะประกอบด้วย U2 (Oscillator 30 MHz) ซึ่งเป็นตัวกำเนิดสัญญาณนาฬิกา 30 MHz จากนั้นนำสัญญาณเอาต์พุตที่ได้ผ่าน U3 (74LS74) ซึ่งทำหน้าที่เป็นวงจรหารความถี่ลงครึ่งหนึ่งซึ่งจะทำให้ได้ความถี่ 15 MHz ที่ต้องการใช้ในการแซมปลิ้ง

3.3.2 วงจรบัฟเฟอร์และวงจรยกระดับสัญญาณ

สัญญาณภาพรวมที่ได้จากกล้องวงจรปิดจะถูกเข้าวงจรนี้ และที่ทางด้านอินพุตของวงจรจะมีตัวต้านทาน R1 (75 Ω) ต่อคร่อมเพื่อทำการปรับอิมพีแดนซ์ของวงจรให้เหมาะสมกับอิมพีแดนซ์เอาต์พุตของกล้องวงจรปิด จากนั้นสัญญาณภาพรวมจะถูกส่งเข้าวงจรยกระดับโดยผ่าน C1 ซึ่งทำหน้าที่ตัดแรงดันไฟตรงที่อาจปะปนมาจากเอาต์พุตของกล้องวงจรปิดออก เมื่อสัญญาณมาถึงวงจรยกระดับซึ่งประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วย D1 และ VR1 สัญญาณจะถูกยกระดับขึ้นตามค่าแรงดันอ้างอิงที่ต้องการ โดยทำการปรับที่ VR1 สาเหตุที่ต้องมีวงจรยกระดับเนื่องจาก สัญญาณที่ออกจากกล้องวงจรปิดมีค่าแรงดันต่ำทำให้ถูกรบกวนจากสัญญาณรบกวนได้ง่าย ดังนั้นจึงต้องยกระดับสัญญาณนี้ขึ้นเพื่อลดสัญญาณรบกวนที่จะเกิดกับสัญญาณภาพจากนั้นสัญญาณที่ได้ทำการยกระดับสัญญาณแล้วจะผ่านวงจรบัฟเฟอร์ซึ่งประกอบด้วย Q1 (BC547) และ R3 โดยผ่าน C2 และ R2 จากนั้นสัญญาณเอาต์พุตจะออกจากขาอิมิตเตอร์ของ Q1 ไปเป็นอินพุตให้กับวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

3.3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

หัวใจสำคัญของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลก็คือ U1 (CA3318) ซึ่งการทำให้ U1 ทำงานได้ถูกต้องจะต้องประกอบด้วยส่วนต่างๆ ดังนี้

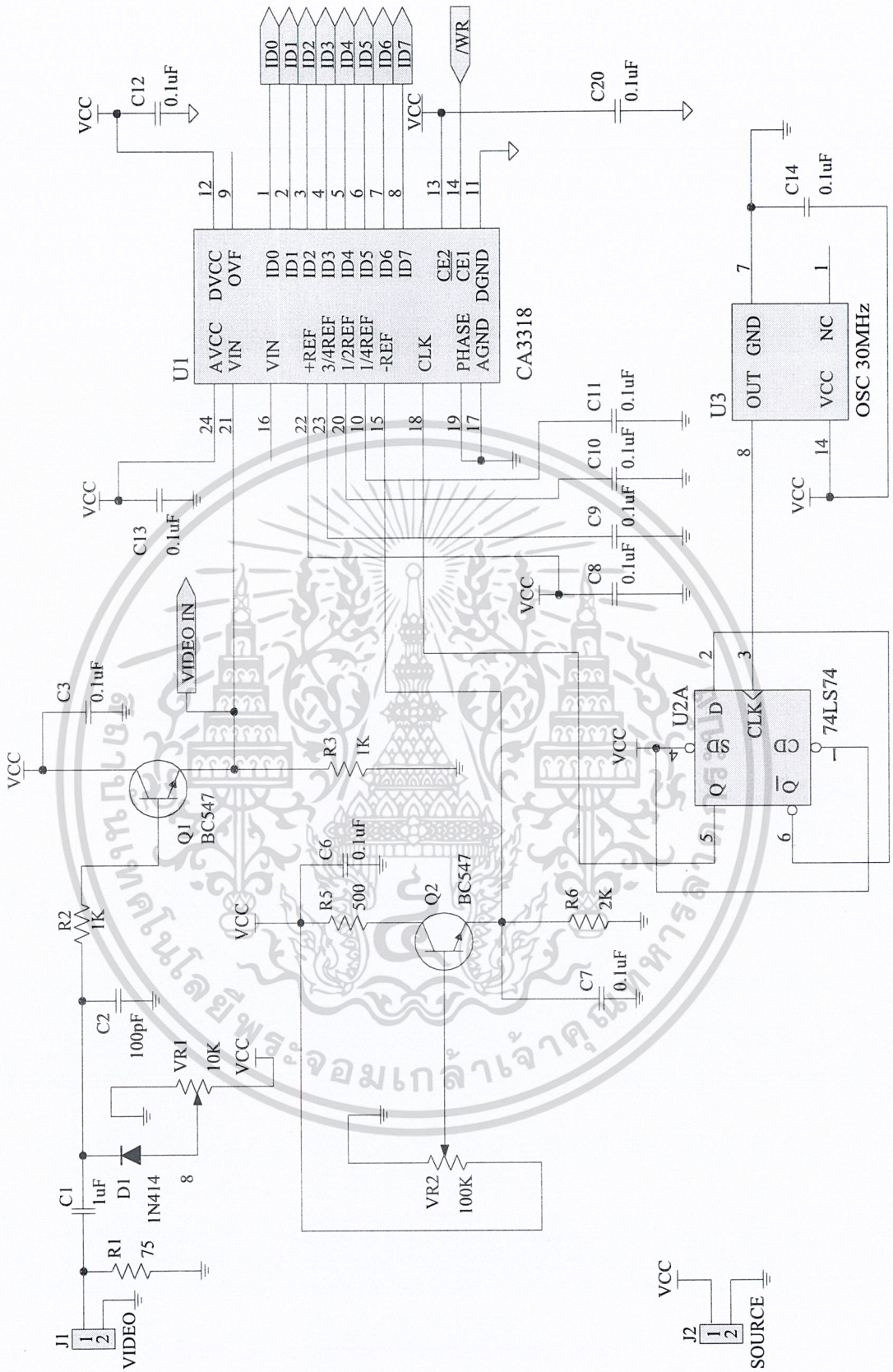
1. สัญญาณอินพุต (ขา 21) ซึ่งได้รับมาจากวงจรบัฟเฟอร์และวงจรยกระดับสัญญาณ

2. ระดับแรงดันอ้างอิง (ขา 22) และ (ขา 15) โดยค่าระดับแรงดันอ้างอิงจะมี 2 ค่า คือ ระดับแรงดันอ้างอิงบวกและระดับอ้างอิงลบ โดยได้ทำการออกแบบไว้ให้สัญญาณภาพรวมที่ทำการยกระดับสัญญาณแล้วมีค่าแรงดันสูงสุดประมาณ 5 โวลต์ และแรงดันสัญญาณภาพต่ำสุดประมาณ 4.2 โวลต์ เพราะฉะนั้นแรงดันอ้างอิงบวก (ขา 22) จึงมีค่า 5 โวลต์ และแรงดันอ้างอิงลบ (ขา 15) มีค่าประมาณ 4.2 โวลต์ ส่วนระดับสัญญาณอ้างอิงอื่นๆ คือ $\frac{1}{4}$ REF , $\frac{1}{2}$ REF , $\frac{3}{4}$ REF (ขา 23,20,10 ตามลำดับ) ไม่ได้ใช้งานจึงต่อลงกราวด์โดยผ่าน C เพื่อป้องกันสัญญาณรบกวนที่อาจเกิดขึ้นได้

3. สัญญาณนาฬิกา (ขา 18) ได้จากวงจรกำเนิดความถี่ซึ่งค่าความถี่ที่ใช้คือ 15 MHz

4. สัญญาณควบคุมโดย U1 จะทำงานและมีเอาต์พุตออกมาได้ก็ต่อเมื่อขา /CE1 (ขา 14) มีสถานะลอจิกเป็น '0' และ CE2 (ขา 13) มีสถานะลอจิกเป็น '1' เอาต์พุตที่ออกจาก U1 จะเป็นสัญญาณดิจิตอลขนาด 8 บิต ซึ่งจะทำให้ได้ความแตกต่างของสัญญาณ $2^8 = 256$ บิต ซึ่งเอาต์พุตนี้จะถูกส่งไปเก็บในหน่วยความจำต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ

3.4.1 วงจรแยกสัญญาณซิงค์

เนื่องจากเราต้องการที่จะเก็บข้อมูลลงในหน่วยความจำขณะที่มีสัญญาณภาพรวมมาจากกล้องวงจรปิดเท่านั้น ดังนั้นเราจึงต้องใช้ข้อมูลจากสัญญาณภาพรวมมาใช้ในการควบคุมจังหวะการชี้แอดเดรสของหน่วยความจำซึ่งข้อมูลที่เราใช้ในการควบคุมก็คือสัญญาณซิงค์ทั้งสัญญาณเวอร์ซิงค์ และสัญญาณฮอร์ซิงค์ที่ได้จากสัญญาณภาพรวม

จากรูปที่ 3.3 วงจรแยกสัญญาณซิงค์จะทำหน้าที่แยกสัญญาณซิงค์ออกมาจากสัญญาณภาพรวม โดยมีหัวใจสำคัญอยู่ที่ U6 (LM1881) เมื่อทำการป้อนสัญญาณภาพรวมเข้าที่ขา 2 ของ U6 โดยผ่าน C4 จะทำให้เราได้สัญญาณซิงค์แนวนอน (Horizontal Sync) หรือสัญญาณฮอร์ซิงค์, สัญญาณซิงค์ทางแนวตั้ง (Vertical Sync) หรือสัญญาณเวอร์ซิงค์, สัญญาณฟิลด์คี่และฟิลด์คู่ (Odd /Even) ออกมาที่ขา 1, 3 และ 7 ตามลำดับ

3.4.2 วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวนอน

วงจรนี้จะรับสัญญาณฮอร์ซิงค์ (ขา 1 ของ U6) มาเป็นอินพุตของวงจรซึ่งวงจรนี้ประกอบไปด้วย 2 ส่วน คือ วงจรหน่วงเวลาสัญญาณฮอร์ซิงค์ ซึ่งประกอบด้วย VR3, C16, U7B (74LS123) และ วงจรกำเนิดพัลส์ ซึ่งประกอบด้วย R7, C17, U7A (74LS123)

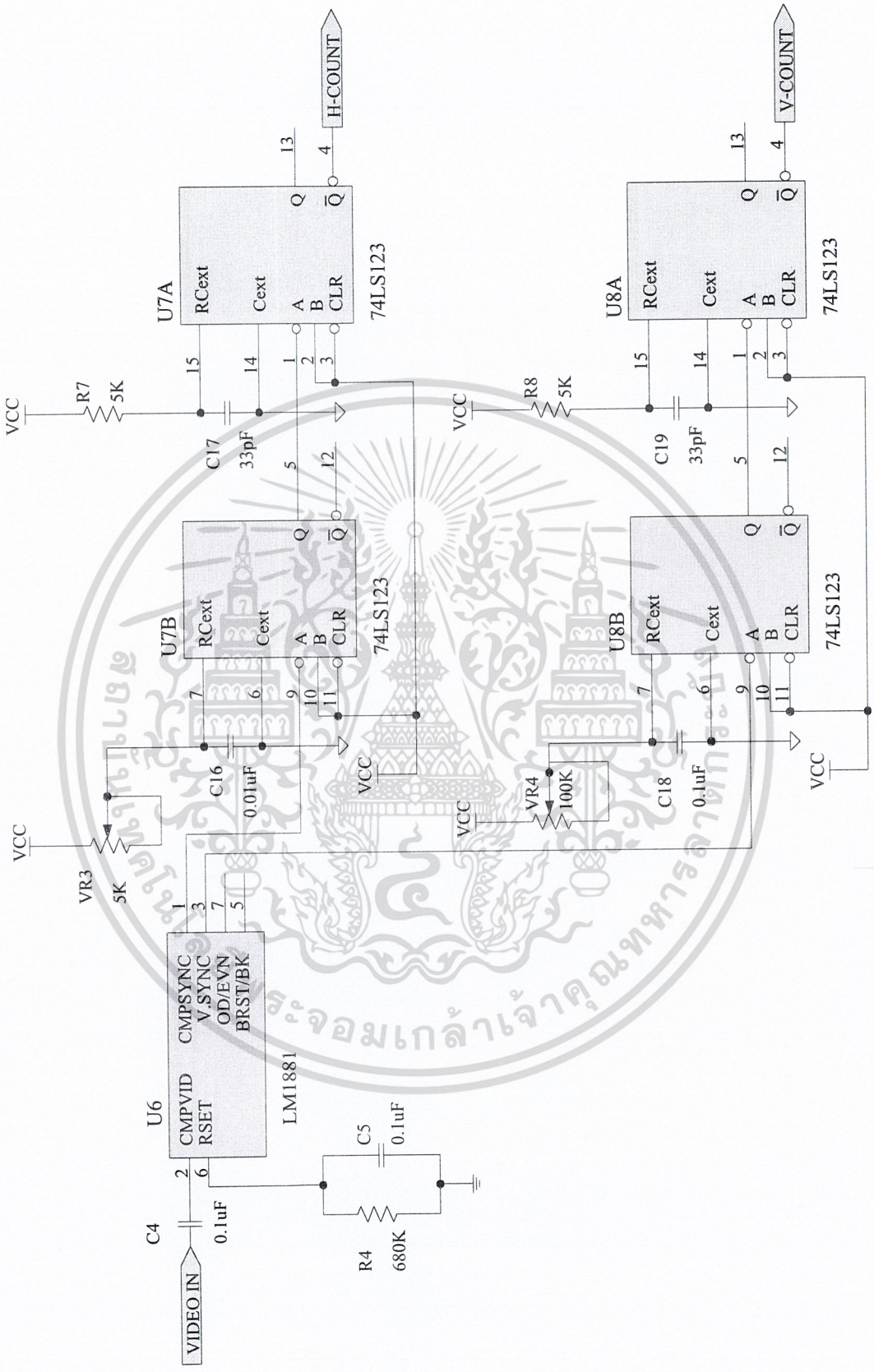
วงจรหน่วงเวลาสัญญาณฮอร์ซิงค์ จะทำการหน่วงเวลาสัญญาณฮอร์ซิงค์ประมาณ 10.5 uS เพื่อให้วงจรชี้แอดเดรส 8 บิตล่างเริ่มทำงาน ณ. จุดเริ่มต้นของเส้นสแกนทุกครั้งที่มีการเริ่มเส้นสแกนใหม่ และเอาต์พุตที่ได้จะถูกป้อนเป็นอินพุตของวงจรผลิตพัลส์ ซึ่งให้เอาต์พุตออกมาเป็นสัญญาณ H-Count ซึ่งจะใช้ในการควบคุมการชี้แอดเดรส 8 บิตล่างในภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน

3.4.3 วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวตั้ง

วงจรนี้จะรับสัญญาณเวอร์ซิงค์ (ขา3 ของU6) มาเป็นอินพุตของวงจรซึ่งวงจรนี้จะประกอบด้วย 2 ส่วน คือ วงจรหน่วงเวลาสัญญาณเวอร์ซิงค์ ซึ่งประกอบด้วย VR4, C18, U8B (74LS123) และวงจรกำเนิดพัลส์ ซึ่งประกอบด้วย R8, C19, U8A (74LS123)

วงจรหน่วงเวลาสัญญาณเวอร์ซิงค์จะทำการหน่วงเวลาสัญญาณเวอร์ซิงค์ประมาณ 1.07 mS เพื่อให้วงจรชี้แอดเดรส 8 บิตบนเริ่มทำงาน ณ.จุดเริ่มต้นของเส้นสแกนทุกครั้งที่มีการเริ่มฟิลด์ใหม่ และเอาต์พุตที่ได้จะถูกป้อนเป็นอินพุตของวงจรผลิตพัลส์ ซึ่งให้เอาต์พุตออกมาเป็นสัญญาณ V-Count ซึ่งจะใช้ในการควบคุมการชี้แอดเดรส 8 บิตบนในภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน

3.5.1 วงจรกำเนิดสัญญาณนาฬิกา 5 MHz

การที่เราจะชี้แอดเดรสของหน่วยความจำขนาด 64 กิโลไบต์ เพื่อให้ง่ายต่อการพิจารณาจะทำการสมมติให้หน่วยความจำมีการเก็บข้อมูลทางแนวนอน 256 ไบต์ และแนวตั้ง 256 ไบต์ ดังนั้นจึงต้องมีวงจรที่ชี้แอดเดรสทั้งทางด้านแนวตั้งและแนวนอน ซึ่งทั้งสองวงจรนี้จะต้องทำงานร่วมกัน นอกจากนี้สัญญาณนาฬิกาที่ใช้ก็จะต้องให้สัมพันธ์กับขนาดหน่วยความจำโดยความถี่สัญญาณนาฬิกาที่ใช้คือ 5MHz ซึ่งสาเหตุที่ใช้ความถี่ 5 MHz เนื่องจากในโครงงานนี้เราต้องการความละเอียดของภาพ 256 × 256 จุด ดังนั้นในแต่ละเส้นสแกนภาพมีคาบเวลา 64 uS แต่ช่วงเวลาที่มีสัญญาณภาพจะประมาณ 80 % ของเวลา 64 uS ซึ่งจะได้เท่ากับ (64 uS) × 0.8 = 51.2 uS นำเอาค่า 51.2 uS หารด้วย 256 จะได้ 200 nS ซึ่งเป็นความถี่ 5 MHz

3.5.2 วงจรชี้แอดเดรส 8 บิตล่าง

วงจรชี้แอดเดรส 8 บิตล่าง ประกอบด้วย U9A (74LS74), U11A (74HC393) และ U11B (74HC393) โดยวงจรนับทางด้านแนวจะทำการชี้แอดเดรส 8 บิตล่าง (CA0 ~ CA7) ของหน่วยความจำ

จากรูปที่ 3.4 จะพบว่าวงจรชี้แอดเดรส 8 บิตล่างจะไม่ทำงาน ถ้าหากไม่มีพัลส์ H-Count มากระตุ้นที่ขา 1 ของ U9A ซึ่งเมื่อมีพัลส์ H-Count มากระตุ้นที่ขา 1 ของ U9A จะทำให้วงจรชี้แอดเดรส 8 บิตล่างทำการนับตั้งแต่ 0 ~ 255 (256 ค่า) เมื่อนับถึง 255 จะทำให้ขา 8 ของ U11B (74HC393) มีสถานะเป็นลอจิก '1' ซึ่งเอาต์พุตขา 8 ของ U10B จะทำหน้าที่ 3 อย่าง คือ

1. นำเอาต์พุตไปชี้แอดเดรสของหน่วยความจำ (CA7)
2. นำไปเข้าเป็นสัญญาณนาฬิกาให้ U9A เพื่อควบคุมให้วงจรชี้แอดเดรส 8 บิตล่างหยุดนับ ทั้งนี้เนื่องจากการทำงานของ U9A เมื่อมีสัญญาณนาฬิกา (ขา 3) ขอบขาขึ้น, (ขา 2) เป็นลอจิก '1' และ(ขา 1) เป็นลอจิก '1' จะทำให้เอาต์พุต Q (ขา 5) เป็นลอจิก '1' ไปทำการเคลียร์วงจรชี้แอดเดรส 8 บิตล่างให้ทำการหยุดนับ
3. นำไปป้อนเป็นอินพุตให้วงจรชี้แอดเดรส 8 บิตบน

3.5.3 วงจรชี้แอดเดรส 8 บิตบน

วงจรชี้แอดเดรส 8 บิตบนจะประกอบด้วย U9B (74LS74) , U12A (74HC393) และ U12B (74HC393) โดยวงจรชี้แอดเดรส 8 บิตบนจะให้เอาต์พุตที่ใช้ชี้แอดเดรส 8 บิตบน (CA8 ~ CA15) ของหน่วยความจำ

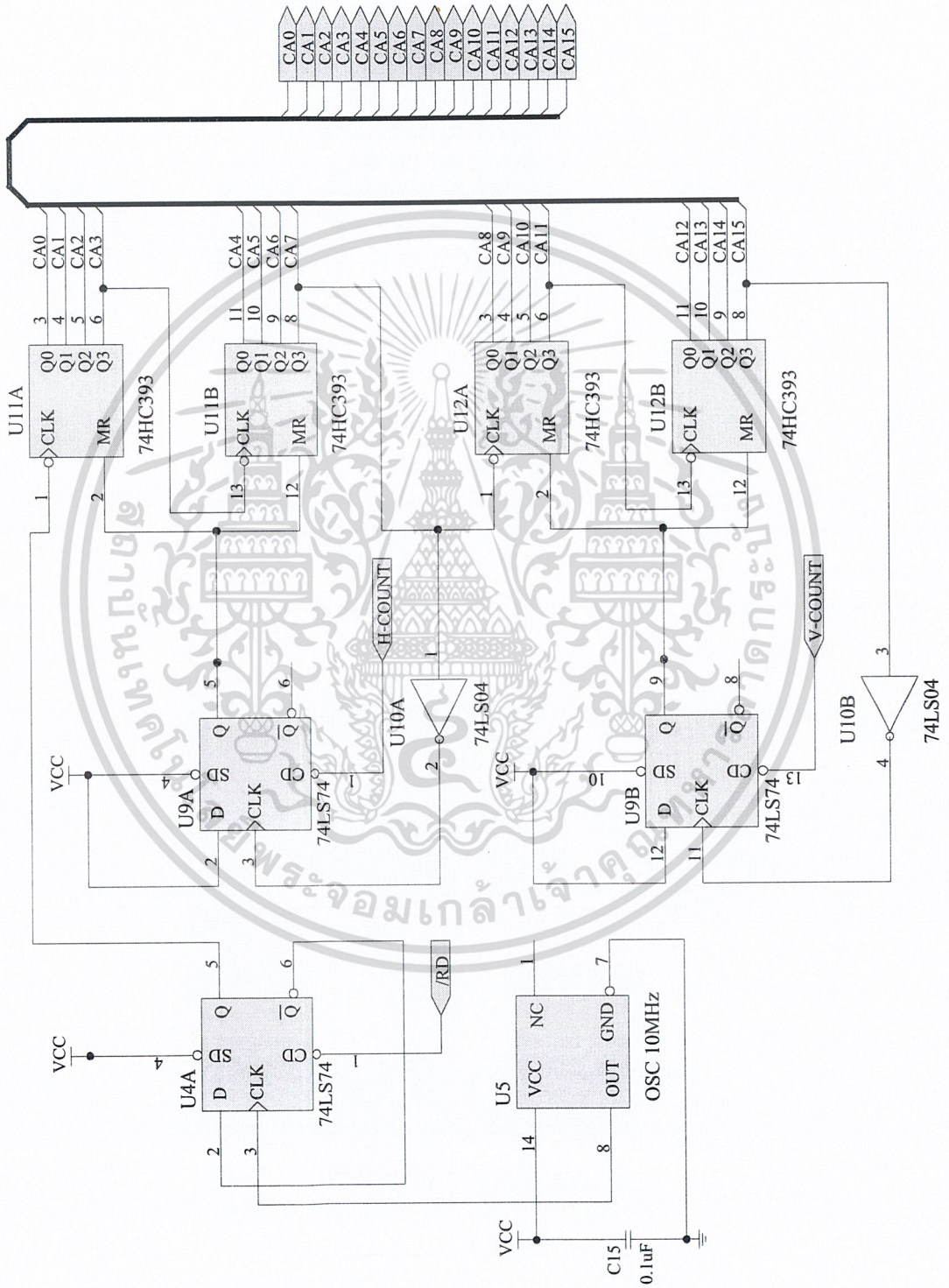
จากรูปที่ 3.4 วงจรชี้แอดเดรส 8 บิตบนจะไม่ทำงานถ้าหากไม่มีพัลส์ V-Count มากระตุ้นที่ขา 13 ของ U9B แต่ถ้าหากมีพัลส์มากระตุ้นที่ขา 13 ของ U9B จะทำให้วงจรชี้แอดเดรส 8 บิตบนทำการนับค่าตั้งแต่ 0 ~ 255 (256 ค่า) เมื่อนับถึง 255 จะทำให้ขา 8 ของ U12B มีสถานะเป็นลอจิก '1' ซึ่งเอาต์พุตขา 8 ของ U12B จะทำหน้าที่ 2 อย่าง คือ

1. นำไปชี้แอดเดรสของหน่วยความจำ (CA15)
2. นำไปเข้าเป็นสัญญาณนาฬิกาให้กับ U9B เพื่อทำให้วงจรชี้แอดเดรส 8 บิตบนหยุดนับ ทั้งนี้เนื่องจากการทำงานของ U9B เมื่อขา 3 มีสัญญาณนาฬิกาในช่วงขอบขาขึ้น, ขา 2 เป็นลอจิก '1'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และขา 13 เป็น ลอจิก '1' จะทำให้เอาต์พุต Q ขา 9 ไปทำการเคลียร์วงจรชี้แอดเดรส 8 บิตบน ให้หยุดนับ

3. นำไปเป็นอินพุตในการสร้างสัญญาณตอบรับเพื่อแจ้งเครื่องคอมพิวเตอร์ว่าได้ทำการเขียนข้อมูลครบ 1 เฟรมแล้ว



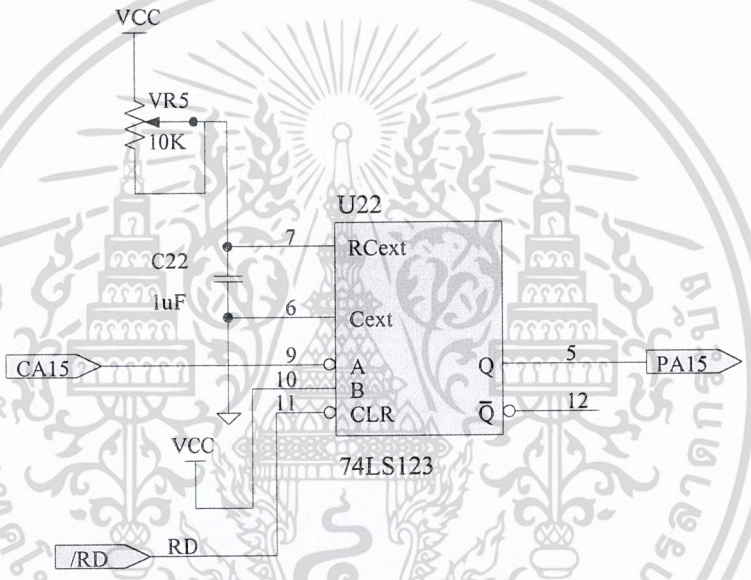
รูปที่ 3.4 ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจรกำเนิดสัญญาณตอบรับ

ในโครงงานนี้เครื่องคอมพิวเตอร์จะอ่านข้อมูลจากหน่วยความจำได้ก็ต่อเมื่อฮาร์ดแวร์ได้ทำการเขียนข้อมูลครบ 1 เฟรม ดังนั้นจึงต้องมีสัญญาณแจ้งจากฮาร์ดแวร์ไปยังเครื่องคอมพิวเตอร์โดยในโครงงานนี้เราจะทำการสร้างสัญญาณนี้จากแอดเดรส CA15 เนื่องจากแอดเดรส CA15 เป็นMSBของแอดเดรสที่ใช้ในการชี้หน่วยความจำและการเขียนจะสมบูรณ์ก็ต่อเมื่อแอดเดรสนี้มีการนับและเปลี่ยนสถานะจาก '1' เป็น '0' เราจึงนำสัญญาณนี้ไปเข้าวงจรกำเนิดสัญญาณตอบรับซึ่งประกอบด้วย U22 (74lLS123), VR 5, C22

โดยสัญญาณ ACK ที่ได้จากวงจรกำเนิดสัญญาณตอบรับจะถูกสร้างขึ้นก็ต่อเมื่อได้รับการทริกจากสัญญาณอินพุตที่ได้จากแอดเดรส CA15



รูปที่ 3.5 วงจรกำเนิดสัญญาณตอบรับหรือสัญญาณ ACK

3.7 ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน

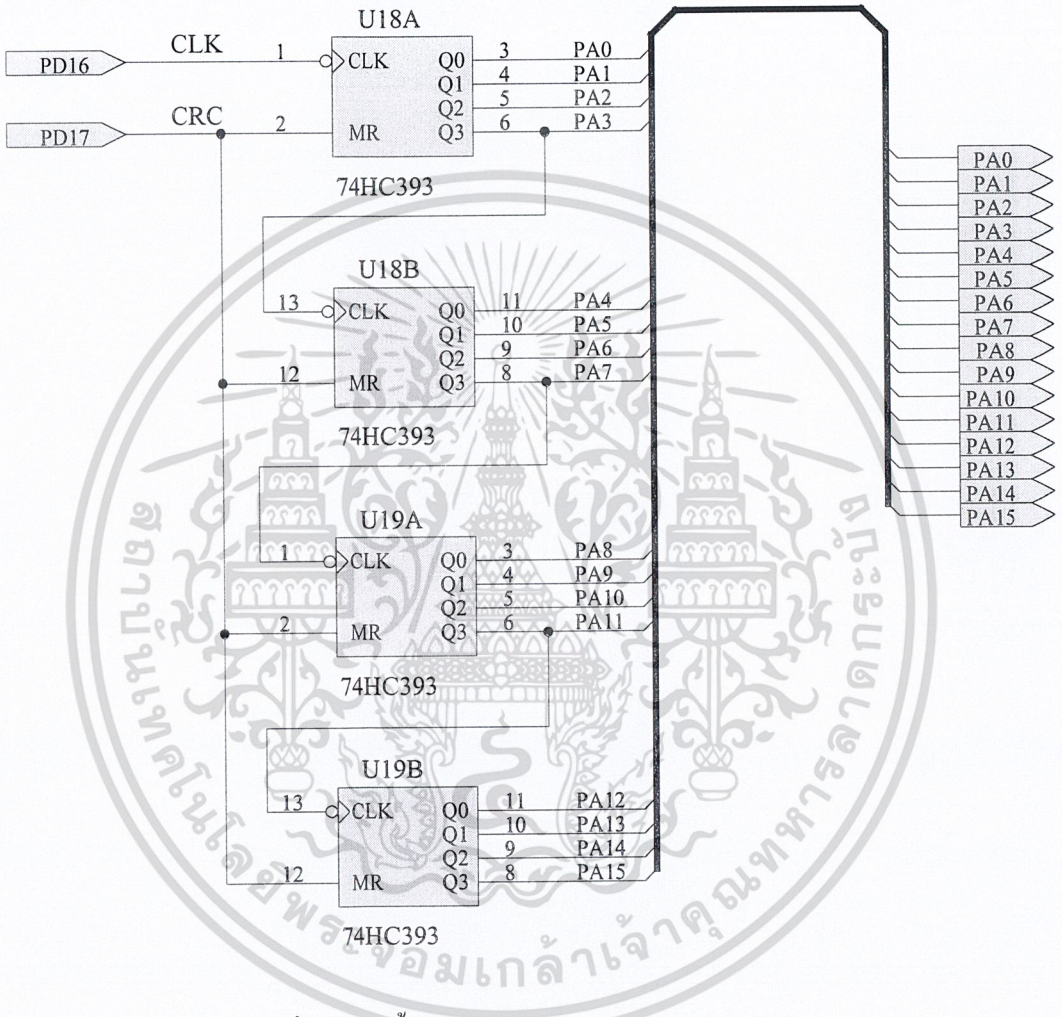
หลังจากที่ฮาร์ดแวร์ได้ทำการเขียนข้อมูลครบ 1 เฟรม ฮาร์ดแวร์จะทำการส่งสัญญาณตอบรับมาแจ้ง เมื่อเครื่องคอมพิวเตอร์ได้รับสัญญาณนี้ก็จะทำการอ่านข้อมูลจากหน่วยความจำ

ส่วนประกอบสำคัญในภาคนี้คือ U18 และ U19 (74HC163) โดยมีหลักการทำงานคือ เมื่อเครื่องคอมพิวเตอร์จะทำการอ่านข้อมูลจากหน่วยความจำจะต้องทำการรีเซต U18 และ U19 โดยผ่านขา 17 ของ DB-25 (Clear Read Counter:CRC) จากนั้นเครื่องคอมพิวเตอร์จะทำการอ่านข้อมูลจากหน่วยความจำโดยผ่านขา1 ของ DB-25(R/W) จากนั้นเครื่องคอมพิวเตอร์จะทำการส่งสัญญาณนาฬิกามา 1 ลูกโดยผ่านขา 16 ของ DB-25เข้ามาที่ขา 1 ของ U17 เพื่อที่จะทำการชี้แอดเดรสของหน่วยความจำ หลังจากนั้นเครื่องคอมพิวเตอร์ก็จะทำการอ่านข้อมูลที่แอดเดรสนั้นเข้าไปแสดงที่หน้าจอเครื่องคอมพิวเตอร์ แล้วทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณนาฬิกาลูกต่อไปเข้ามาซึ่งจะทำกระบวนการเช่นนี้ไปเรื่อยๆ จนกระทั่งอ่านข้อมูลที่เก็บในหน่วยความจำได้ครบ 1 เฟรมเครื่องคอมพิวเตอร์ก็จะทำการหยุดการอ่าน

โดย U18 ใช้สำหรับชี้แอดเดรสหน่วยความจำ 8 บิตล่าง และ U19 ใช้สำหรับการชี้แอดเดรสหน่วยความจำ 8 บิตบน



รูปที่ 3.6 ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

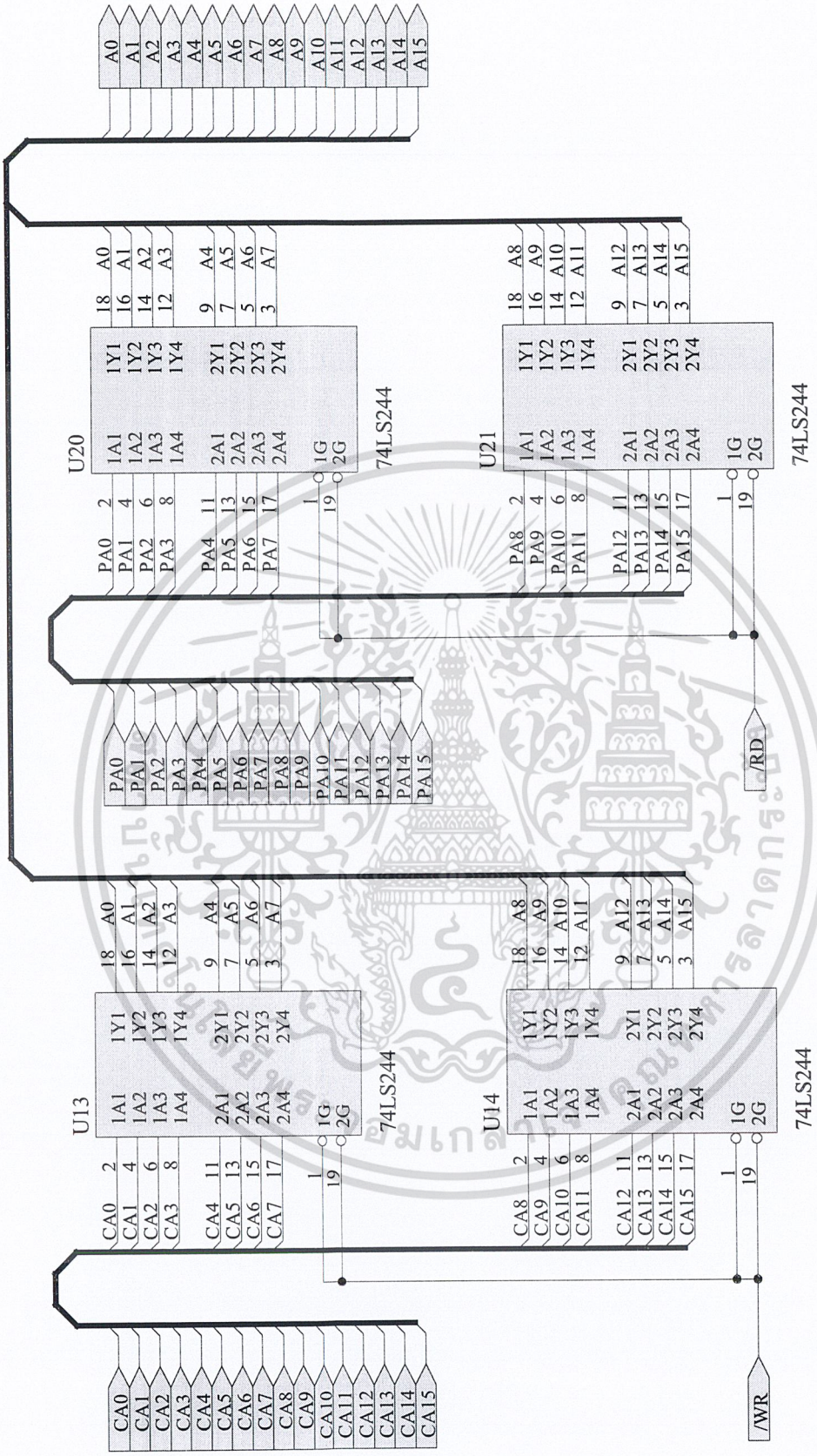
3.8 ภาคเลือกเขียนหรืออ่านหน่วยความจำ

ในโครงงานนี้เราจะต้องทำการเขียนและอ่านข้อมูลจากหน่วยความจำ ดังนั้นจึงต้องมีการควบคุมการทำงานในขั้นตอนนี้ ซึ่งในภาคนี้จะประกอบไปด้วย 2 ส่วน คือ วงจรชี้แอดเดรสเพื่อเขียนข้อมูลลงในหน่วยความจำ และวงจรชี้แอดเดรสเพื่ออ่านข้อมูลจากหน่วยความจำ โดยทั้งสองวงจรนี้จะทำงานสลับกัน ซึ่งจะถูกควบคุมโดยสัญญาณ R/W จากเครื่องเครื่องคอมพิวเตอร์

วงจรชี้แอดเดรสเพื่อเขียนข้อมูลลงในหน่วยความจำประกอบด้วย U13 (74LS244) และ U14 (74LS244) ซึ่งทำหน้าที่เป็นบัฟเฟอร์โดยจะรับอินพุตของการชี้แอดเดรสมาจากวงจรชี้แอดเดรสสำหรับการเขียนเพื่อนำไปชี้แอดเดรสในการเขียนข้อมูลลงไปในหน่วยความจำ

วงจรชี้แอดเดรสเพื่ออ่านข้อมูลจากหน่วยความจำประกอบด้วย U20 (74LS244) และ U21 (74LS244) ซึ่งทำหน้าที่เป็นบัฟเฟอร์ โดยจะรับอินพุตของการชี้แอดเดรสมาจากวงจรชี้แอดเดรสสำหรับการอ่านเพื่อนำไปชี้แอดเดรสในการอ่านข้อมูลจากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ภาคเลือกเขียนหรืออ่านหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

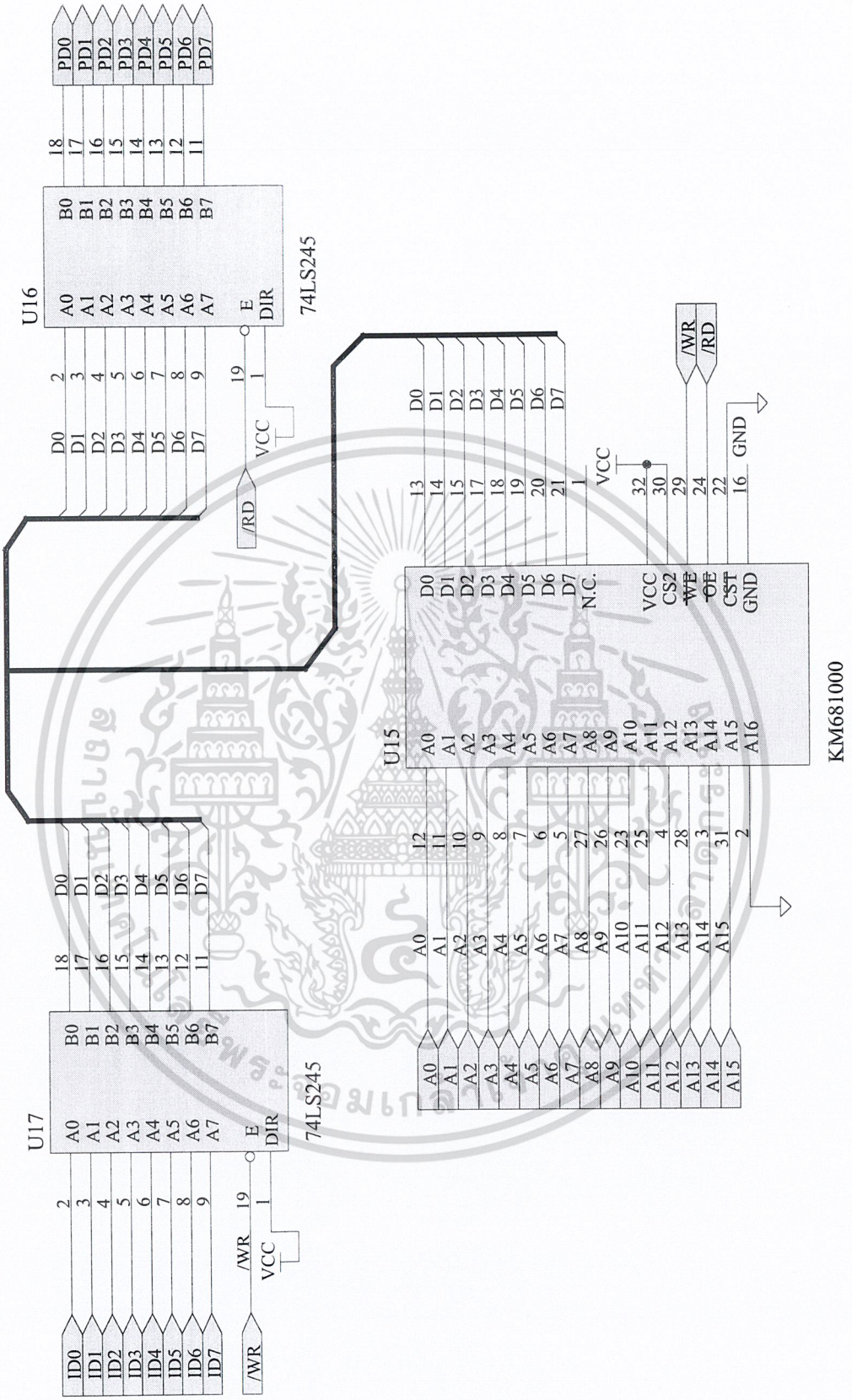
3.9 ภาคหน่วยความจำ

ภาคหน่วยความจำ จะทำหน้าที่ 2 อย่างคือ การเขียนข้อมูลลงในหน่วยความจำและการอ่านข้อมูลออกจากหน่วยความจำ (Memory) โดยทั้งสองหน้าที่นี้จะทำงานสลับกันซึ่งจะถูกควบคุมการทำงานโดยเครื่องเครื่องคอมพิวเตอร์

การเขียนข้อมูลลงในหน่วยความจำจะเป็นการทำงานร่วมกันของ U17 (74LS245) และหน่วยความจำ U15 (KM681000) โดยการเขียนข้อมูลลงในหน่วยความจำจะนำข้อมูลจากเอาต์พุตของ U1 (ID0 ~ ID7) ในภาค Analog To Digital Converter มาทำการเขียนลงในหน่วยความจำโดยผ่านบัฟเฟอร์ U17

การอ่านข้อมูลจากหน่วยความจำจะเป็นการทำงานร่วมกันของ U16 (74LS245) และหน่วยความจำ U15 (KM681000) โดยการอ่านข้อมูลจากหน่วยความจำเป็นการอ่านข้อมูลของหน่วยความจำเพื่อนำไปเก็บในเครื่องเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 ภาคหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 ภาคอินเตอร์เฟส

ในโครงงานนี้จะทำการเชื่อมต่อระหว่างเครื่องเครื่องคอมพิวเตอร์และฮาร์ดแวร์โดยผ่านพอร์ตขนานและทำการรับข้อมูลครั้งละ 8 บิต โดยใช้พอร์ตข้อมูลของพอร์ตขนาน (ขา2-9) ทำการควบคุมอุปกรณ์ภายนอกโดยผ่านพอร์ตควบคุม และอุปกรณ์ภายนอกจะแจ้งความพร้อมในการส่งข้อมูลโดยผ่านพอร์ตสถานะ

การเชื่อมต่อกันระหว่างพอร์ตขนานของเครื่องคอมพิวเตอร์แบบ DB-25 กับฮาร์ดแวร์จะทำการต่อดังตารางด้านล่างนี้

ตารางที่ 3.1 แสดงการเชื่อมต่อฮาร์ดแวร์กับขา DB-25

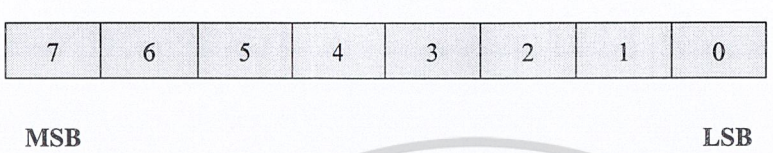
DB-25	ฮาร์ดแวร์
1	R/W
2	ข้อมูลบิต0(LSB)
3	ข้อมูลบิต1
4	ข้อมูลบิต2
5	ข้อมูลบิต3
6	ข้อมูลบิต4
7	ข้อมูลบิต5
8	ข้อมูลบิต6
9	ข้อมูลบิต7(MSB)
10	GND
11	GND
12	GND
13	GND
14	-
15	ACK
16	CLK
17	CRC
18-25	GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10.1 แอดเดรสพอร์ตขนานที่ใช้งาน

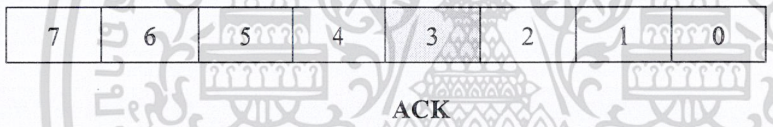
การอ้างอิงแอดเดรสของพอร์ตขนานในโครงงานนี้จะเลือกใช้แอดเดรสฐานของ LPT1 โดยจะมีแอดเดรสเริ่มต้นที่ 0x378h และทำการเซ็ตพอร์ตเป็นพอร์ตขนานสองทิศทางซึ่งจะได้แอดเดรสของพอร์ตต่างๆ ดังนี้

1. พอร์ตข้อมูล ทำหน้าที่รับข้อมูลจากหน่วยความจำภายนอกเข้ามาโดยจะใช้ตำแหน่ง 0x378h (ตำแหน่งฐาน) ทั้ง 8 บิต (ขา 2 - 9 ของ DB-25)



รูปที่ 3.9 แสดงบิตที่ใช้งานในพอร์ตข้อมูล

2. พอร์ตสถานะ ทำหน้าที่รับสัญญาณที่ส่งมาจากฮาร์ดแวร์เพื่อแจ้งให้เครื่องคอมพิวเตอร์ทราบว่าทางด้านฮาร์ดแวร์ได้ทำการเขียนข้อมูลครบ 1 ฟิลด์ให้มาเตรียมอ่านข้อมูลออกไปโดยจะใช้ตำแหน่ง 0x379h บิต 3 (ขา 15 ของ DB-25)

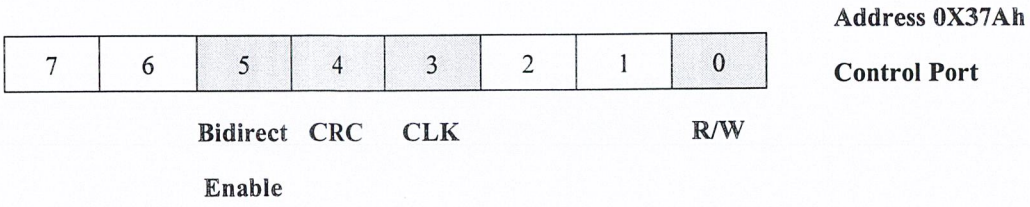


รูปที่ 3.10 แสดงบิตที่ใช้งานในพอร์ตสถานะ

3. พอร์ตควบคุม ทำหน้าที่ในการควบคุมการทำงานของฮาร์ดแวร์โดยจะใช้ตำแหน่ง 0x37Ah แบ่งเป็น 4 ส่วน

- ทำการควบคุมฮาร์ดแวร์ให้ทำการอ่าน หรือเขียนข้อมูลที่หน่วยความจำโดยใช้บิต 0 ซึ่งหากบิต 0 เป็น '0' จะเป็นสภาวะการเขียนข้อมูล และ '1' จะเป็นสภาวะการอ่านข้อมูล
- ทำการส่งสัญญาณนาฬิกาเพื่อใช้ในการชี้แอดเดรสข้อมูลที่จะอ่านออกมาจากหน่วยความจำโดยใช้บิต 3
- ทำการเคลียร์วงจรชี้แอดเดรสสำหรับการอ่านโดยใช้บิต 4
- ทำการอิเนเบิลให้พอร์ตขนานทำหน้าที่เป็นพอร์ตสองทิศทางโดยใช้บิต 5 ซึ่งจะต้องทำการเซ็ตให้บิต 5 เป็น '1'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงบิตที่ใช้งานในพอร์ตควบคุม

3.11 โปรแกรม

3.11.1 อัลกอริทึมทางด้านซอร์ฟแวร์

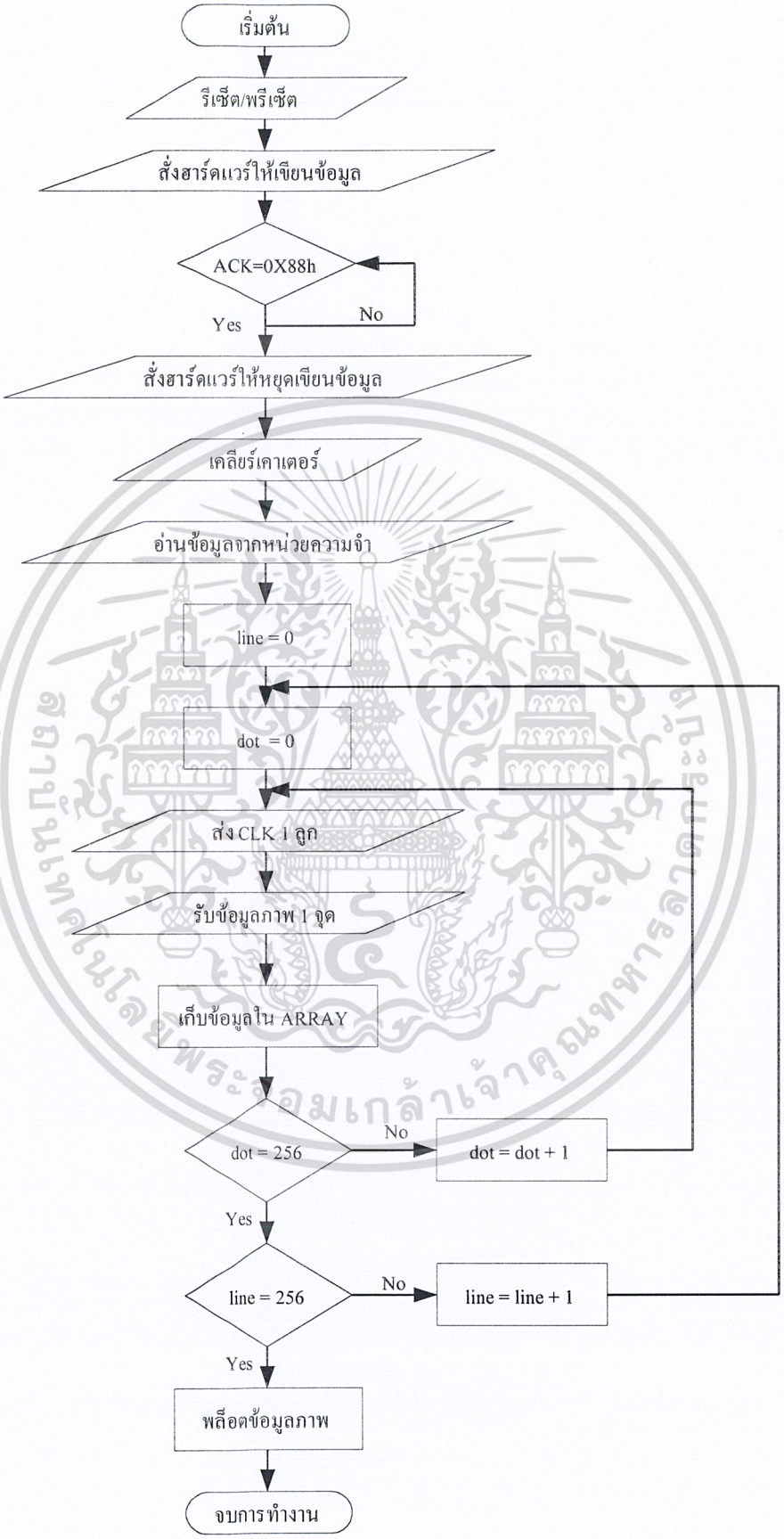
ในโครงงานนี้ซอร์ฟแวร์จะเปรียบเสมือนสมองที่ใช้ในการตัดสินใจและควบคุมการทำงานของส่วนฮาร์ดแวร์ทั้งหมด ซึ่งมีลำดับการทำงานดังนี้

1. ทำการรีเซ็ตฮาร์ดแวร์โดยการทำให้ภาคชี้แอดเดรสอยู่ในสภาวะเริ่มต้น และเนื่องจากว่าไอซีที่ใช้ในการนับเพื่ออ่านข้อมูลเป็นแบบซิงโครนัสจึงต้องใช้สัญญาณนาฬิกา 1 ลูกในการรีเซ็ต ซึ่งการรีเซ็ตเราจะส่งข้อมูล 0x0Bh และ 0x0Fh ตามลำดับผ่านแอดเดรส 0x37Ah
2. ทำการปรีเซ็ตฮาร์ดแวร์ให้อยู่ในสภาวะที่จะทำการเขียนข้อมูลลงในหน่วยความจำ ซึ่งคำสั่งที่ใช้ในการทำปรีเซ็ตจะรวมอยู่ในคำสั่งการทำรีเซ็ต
3. ทำการตรวจสอบสัญญาณ ACK ที่จะส่งมาจากฮาร์ดแวร์เพื่อจะแจ้งให้เครื่องคอมพิวเตอร์ทราบว่าในขณะนี้ฮาร์ดแวร์ได้ทำการเขียนข้อมูลจนเต็มหน่วยความจำแล้ว โดยหากฮาร์ดแวร์ทำการเขียนข้อมูลจนเต็มแล้วบิต 3 ของแอดเดรส 0x379h จะมีค่าเป็น ‘0’ และเนื่องจากบิต 7 – 4 ของแอดเดรสนี้เราจะต่อลงกราวด์ทั้งหมด ดังนั้นค่าที่จะทำให้เงื่อนไขข้างต้นเป็นจริงก็คือค่า 0x80h
4. ทำการรีเซ็ตภาคชี้แอดเดรสที่ใช้ในการอ่านข้อมูลออกจากหน่วยความจำอยู่ในสภาวะเริ่มต้นโดยส่งข้อมูล 0x0Bh และ 0x0Fh ตามลำดับผ่านแอดเดรส 0x37Ah
5. ทำการส่งสัญญาณนาฬิกาออกไปครั้งละลูกของรอบการทำงาน เพื่อทำการอ่านข้อมูลจากหน่วยความจำเข้ามาครั้งละ 1 จุด จากนั้นนำไปพล็อตครั้งละจุดจนกระทั่งครบ

256 จุด × 256 เส้น = 65,536 จุดภาพ (1ภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11.2 แผนผังการทำงานของโปรแกรม



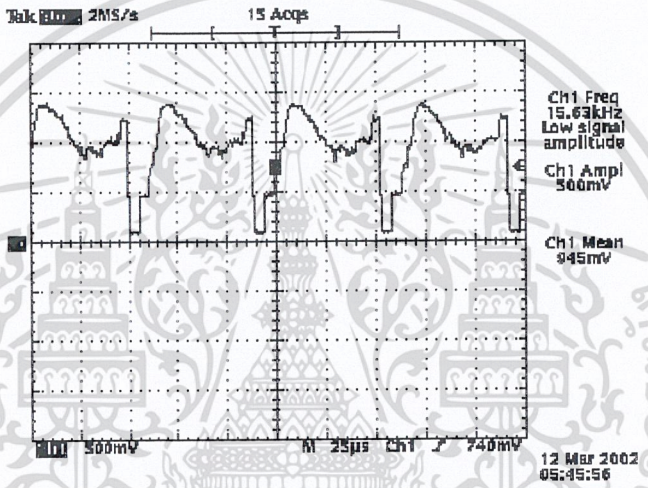
รูปที่ 3.12 แสดงแผนภูมิการทำงานของโปรแกรมวิชวลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

การทดลองในภาคนี้เริ่มจากทำการวัดสัญญาณภาพขาว-ดำที่ออกจากกล้องวีดีโอที่จะนำมาใช้เป็นอินพุตให้กับภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล ซึ่งสัญญาณภาพขาว-ดำจะถูกแปลงเป็นสัญญาณดิจิตอลที่มีความละเอียด 8 บิตต่อหนึ่งพิกเซล โดยใช้ไอซีเบอร์ CA3318 (U1) ซึ่งเป็นอุปกรณ์หลักในภาคนี้โดยความถี่ของสัญญาณนาฬิกาที่ใช้ในการแซมปลิ้งสัญญาณคือ 15 MHz ซึ่งผลการทดลองที่ได้มีดังนี้



รูปที่ 4.1 สัญญาณภาพรวมขาว-ดำ ที่ออกจากกล้องวีดีโอ

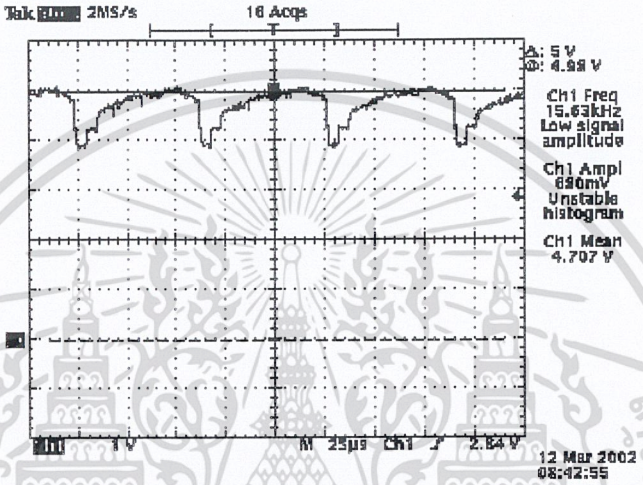
รูปที่ 4.1 แสดงถึงสัญญาณภาพรวมที่ออกจากกล้องวีดีโอ (หรือกล้องวงจรปิด) โดยสัญญาณภาพที่ออกมาจะเป็นสัญญาณภาพขาว-ดำที่มีความถี่ประมาณ 15,625 Hz ซึ่งในรูปจะแสดงถึงเส้นภาพแต่ละเส้นที่กล้องได้ทำการสแกนทางแนวนอน โดยสัญญาณฮอร์ซิงค์ทำหน้าที่ในการควบคุมการสะบัดกลับทางด้านแนวนอนแทรกอยู่ระหว่างเส้นสแกนแต่ละเส้น โดยสัญญาณซิงค์จะเป็นสัญญาณที่มีระดับแรงดันต่ำที่สุดในสัญญาณภาพรวม จากรูปพบว่าส่วนที่มีระดับแรงดันสูงสุดก็คือสัญญาณภาพที่มีความสว่างมากที่สุดหรือสีขาวนั่นเอง ส่วนสัญญาณที่มีระดับแรงดันลดลงมามีระดับความสว่างลดลงมาจนกระทั่งถึงสัญญาณที่มีระดับแรงดันต่ำที่สุดก็คือสีดำนั่นเอง

เมื่อสัญญาณภาพถูกป้อนเข้ามาในภาคนี้ก็จะถูกส่งเข้าวงจรยกระดับสัญญาณ (C1, D1, VR1) และวงจรบัฟเฟอร์ (C2, R2, Q1, C3, R3) เนื่องจากระดับสัญญาณภาพที่ออกมาจากกล้องวีดีโอเป็นสัญญาณที่มีระดับแรงดันต่ำซึ่งจะทำให้สัญญาณถูกรบกวนจากสัญญาณรบกวนได้ง่าย ดังนั้นจึงต้องทำการยกระดับสัญญาณ ซึ่งเราได้ทำการยกระดับสัญญาณให้มีระดับสัญญาณสูงสุดเท่ากับ 5 โวลต์ ซึ่งเป็นค่าเฉลี่ยของระดับแรงดันสูงสุดของสัญญาณที่ได้จากการทำการทดลองให้กล้องถ่ายไปที่หลอดไฟ (เพื่อให้ได้สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

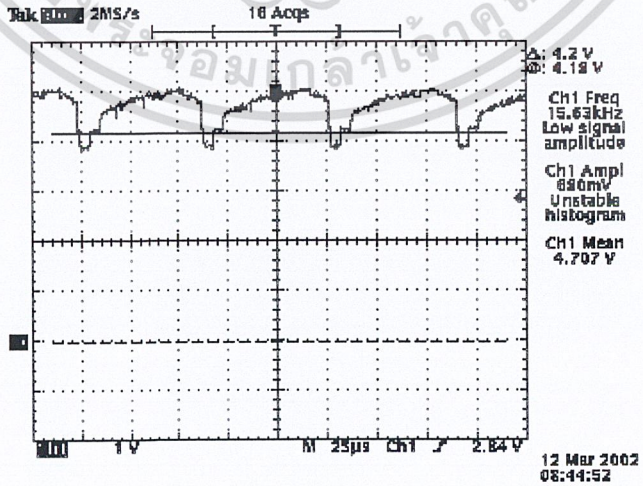
ที่มีระดับความสว่างสูงที่สุด) จากนั้นก็จะนำสัญญาณที่ได้ทำการยกระดับแล้วส่งไปเข้ายังวงจรบัฟเฟอร์เพื่อส่งต่อเข้าไปป้อนเป็นอินพุตให้กับขา 21 ของ U1 (CA 3318)

ในส่วนของ U1 จะทำการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลได้อย่างถูกต้องนั้นจะต้องทำการป้อนแรงดันอ้างอิงให้กับขาแรงดันอ้างอิงบวก (ขา 22 (+REF) ของU1) และขาแรงดันอ้างอิงลบ (ขา 15 (-REF) ของU1) อย่างถูกต้อง ซึ่งได้ตั้งค่าระดับแรงดันอ้างอิงบวกไว้ที่ 5 โวลต์ ซึ่งเท่ากับระดับแรงดันเฉลี่ยสูงสุดที่ได้จากวงจรยกระดับ ส่วนค่าระดับแรงดันอ้างอิงลบนั้นเราจะทำการตั้งไว้ที่ 4.2 โวลต์ ซึ่งเป็นระดับแรงดันที่วัดได้จากระดับสัญญาณสีดำของสัญญาณของสัญญาณภาพรวม



รูปที่ 4.2 สัญญาณอินพุตที่ขา 21 ของ U1 (ระดับแรงดันอ้างอิงบวก 5 โวลต์)

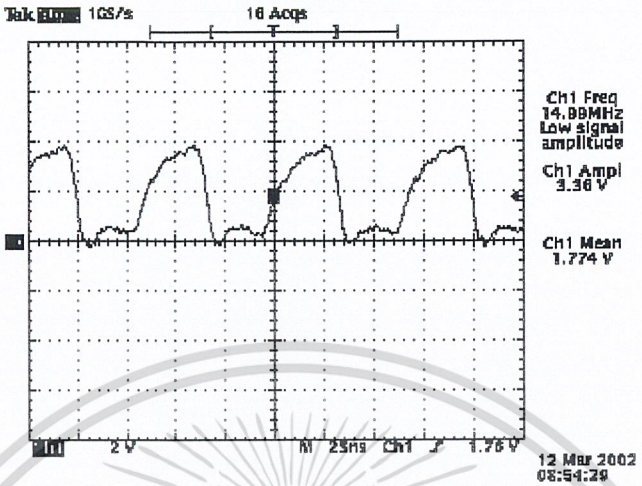
รูปที่ 4.2 แสดงถึงสัญญาณอินพุตที่ผ่านการยกระดับแล้วทำการป้อนเข้ามาเป็นสัญญาณอินพุตให้กับ U1 ทำการวัดค่าระดับแรงดันเฉลี่ยสูงสุดเพื่อใช้ในการตั้งระดับแรงดันอ้างอิงบวกของ U1 ด้วย



รูปที่ 4.3 สัญญาณอินพุตที่ขา 21 ของ U1 (ระดับแรงดันอ้างอิงลบ 4.2 โวลต์)

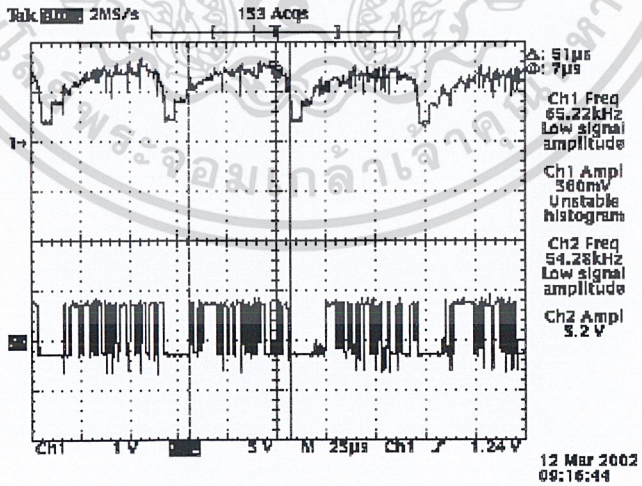
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงถึงการวัดค่าระดับสัญญาณสีดำในสัญญาณภาพรวม เพื่อนำไปใช้สร้างระดับแรงดันอ้างอิงลบของ U1



รูปที่ 4.4 สัญญาณนาฬิกาความถี่ 15 MHz

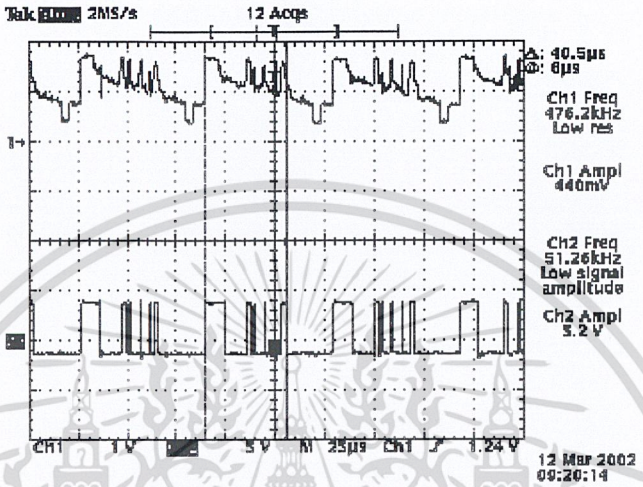
รูปที่ 4.4 แสดงถึงสัญญาณนาฬิกาที่ใช้ในการป้อนเข้าที่ขา CLK (ขา 18) ของ U1 ซึ่งก็คือความถี่ที่ใช้ในการแซมปลิ้งนั่นเอง สาเหตุที่ใช้ความถี่เท่ากับ 15 MHz ก็เนื่องจากความถี่สูงสุดของสัญญาณภาพมีค่าประมาณ 5 MHz ดังนั้นเพื่อการแซมปลิ้งสัญญาณให้เป็นไปตามทฤษฎีการแซมปลิ้งของไนควิสท์ที่กล่าวว่าการที่จะทำการแซมปลิ้งสัญญาณแล้วไม่ให้เกิดการเอเลียสซิ่ง (Aliasing) หรือใช้ความถี่มากกว่า 2 เท่าของความถี่สูงสุดของสัญญาณนั้น ซึ่งในที่นี้ก็คือมากกว่า 10 MHz นั่นเอง



รูปที่ 4.5 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ ID0 (CH2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แสดงถึงการเปรียบเทียบระหว่างสัญญาณภาพรวมที่ออกมาจากกล้องวีดีโอกับสัญญาณเอาต์พุตที่ออกมาจาก U1 (CA3318) เพื่อตรวจสอบว่าระดับสัญญาณอ้างอิงลบที่ตั้งไว้ถูกต้องหรือไม่ ถ้าหากระดับสัญญาณอ้างอิงลบที่ตั้งไว้ไม่ถูกต้อง ช่วงที่สัญญาณเป็นสัญญาณแบลงกิ้งและสัญญาณฮอร์ซิงค์จะถูกแปลงเป็นสัญญาณดิจิตอลออกมาทางเอาต์พุตของ U1 ด้วย แต่ในที่นี้สัญญาณเอาต์พุตที่ออกมาจาก U1 มีเฉพาะในส่วนที่เป็นสัญญาณภาพขาว-ดำ ซึ่งให้ผลถูกต้องตามต้องการ



รูปที่ 4.6 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ ID7 (CH2)

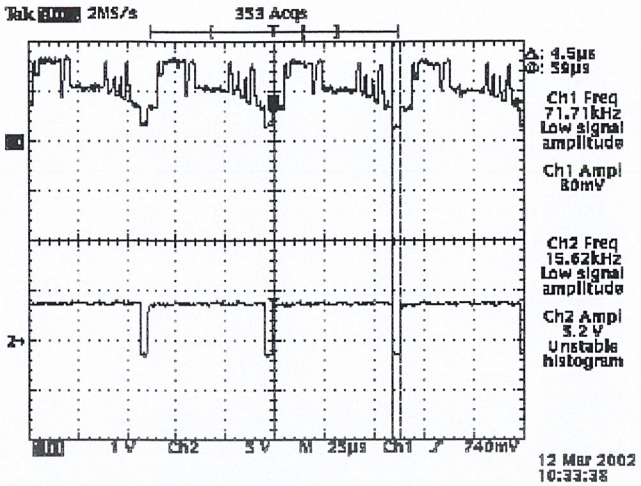
รูปที่ 4.6 แสดงถึงสัญญาณภาพรวมที่ได้จากกล้องวีดีโอ และสัญญาณที่ขา ID7 (บิตที่7 = MSB) ของ U1 ที่ได้จากการแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอล

4.2 ภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำ

การชี้แอดเดรสของหน่วยความจำเพื่อทำการเก็บข้อมูล ที่ได้จากภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลได้อย่างถูกต้องนั้น จะต้องใช้สัญญาณฮอร์ซิงค์และสัญญาณเวอร์ซิงค์ในการกำเนิดสัญญาณเพื่อควบคุมการชี้แอดเดรสหน่วยความจำให้สัมพันธ์กับข้อมูลดิจิตอลที่เก็บในหน่วยความจำ โดยแสดงวงจรดังรูปในที่ 3.3 ภาคนี้จะประกอบไปด้วยวงจรแยกสัญญาณซิงค์, วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวนอน, วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวตั้ง

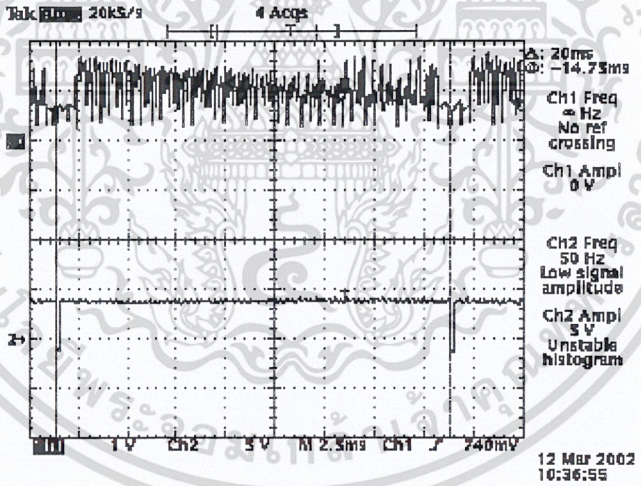
สัญญาณในรูปที่ 4.7 - 4.9 แสดงถึงสัญญาณเอาต์พุตที่ออกจากวงจรแยกสัญญาณซิงค์ โดยส่วนประกอบสำคัญของวงจรแยกสัญญาณซิงค์จะอยู่ที่ U6 (LM 1881)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ HOR SYNC (CH2)

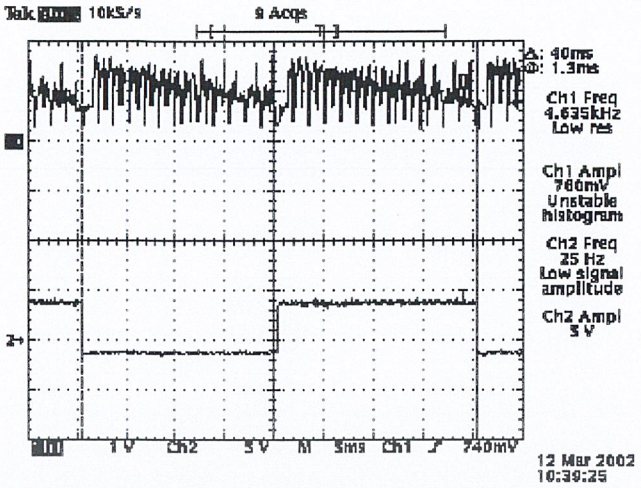
รูปที่ 4.7 แสดงถึงสัญญาณฮอร์ซิงค์ที่ออกจากขา 1 ของ U6 เปรียบเทียบกับสัญญาณภาพรวมที่ป้อนเป็นอินพุตที่ขา 2 ของ U6 เพื่อตรวจสอบว่าสัญญาณฮอร์ซิงค์ได้จาก U6 นั้นถูกต้องหรือไม่ ซึ่งผลที่ได้ถูกต้องเป็นไปตามทฤษฎี คือได้ความถี่ประมาณ 15,625 Hz



รูปที่ 4.8 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ VER SYNC (CH2)

รูปที่ 4.8 แสดงถึงสัญญาณเวอร์ซิงค์ที่ออกจากขา 3 ของ U6 เปรียบเทียบกับสัญญาณภาพรวมที่ป้อนเป็นอินพุตที่ขา 2 ของ U6 เพื่อตรวจสอบว่าสัญญาณซิงค์ที่ได้จาก U6 นั้นถูกต้องหรือไม่ ซึ่งผลที่ได้ถูกต้องเป็นไปตามทฤษฎี คือได้ความถี่ประมาณ 50 Hz

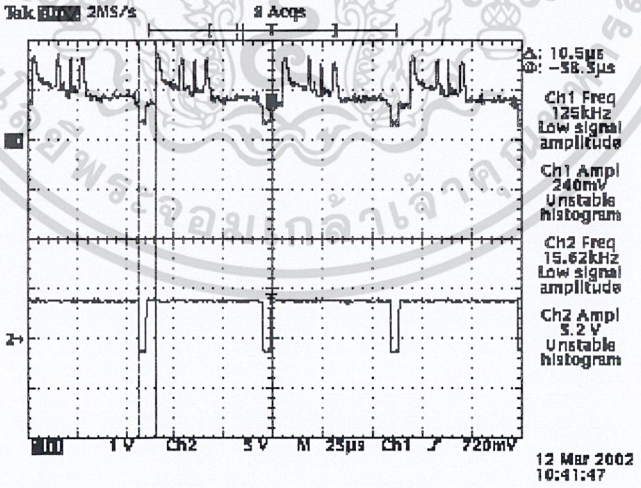
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ ODD/EVEN (CH2)

รูปที่ 4.9 แสดงถึงสัญญาณ ODD/EVEN ที่ออกจากขา 7 ของ U6 เปรียบเทียบกับสัญญาณภาพรวมที่ป้อนเป็นอินพุตที่ขา 2 ของ U6 เพื่อตรวจสอบว่าสัญญาณ ODD/EVEN ที่ได้จาก U6 นั้นถูกต้องหรือไม่ ซึ่งผลที่ได้ถูกต้องเป็นไปตามทฤษฎี คือได้ความถี่ประมาณ 25 Hz

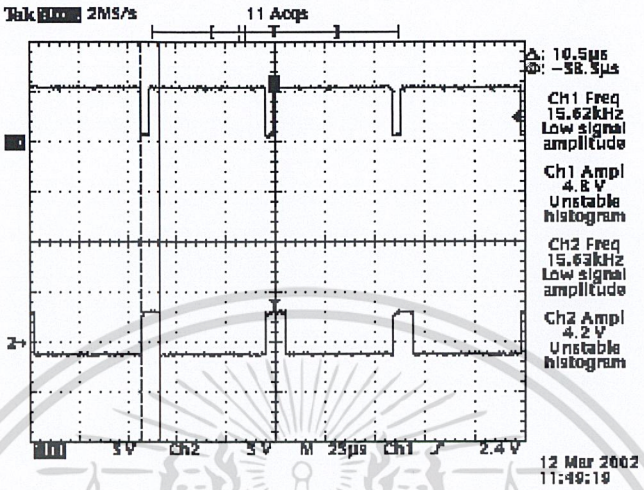
สัญญาณฮอร์ซิงค์ที่ออกจากขา 1 ของ U6 จะถูกส่งป้อนเข้าที่วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวนอน ซึ่งในวงจรนี้จะประกอบไปด้วย 2 ส่วน คือ ส่วนของวงจรหน่วงเวลาฮอร์ซิงค์และวงจรผลิตพัลส์โดยรูปที่ 4.10 - 4.12 จะแสดงสัญญาณเอาต์พุตที่ได้จากวงจร



รูปที่ 4.10 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ HOR SYNC (CH2)

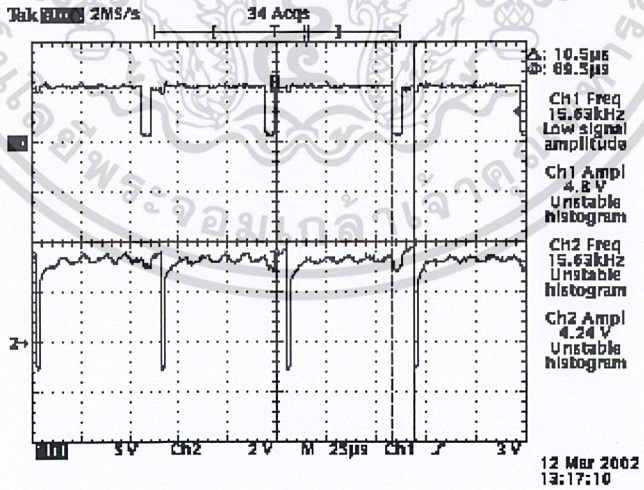
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.10 แสดงถึงการเปรียบเทียบผลต่างทางเวลาระหว่างการเริ่มต้นการสแกนเส้นสแกนใหม่ของสัญญาณภาพรวมกับสัญญาณขอบขาลงของสัญญาณฮอร์ซิงค์ ซึ่งค่าที่วัดได้มีค่าประมาณ 10.5 uS ซึ่งเป็นไปตามทฤษฎีจากนั้นจะนำค่าเวลานี้ไปทำการสร้างวงจรหน่วงเวลา



รูปที่ 4.11 สัญญาณ HOR SYNC (CH1) เทียบกับ สัญญาณ ขา 5 ของ U7B (CH2)

รูปที่ 4.11 แสดงถึงสัญญาณเอาต์พุตที่ได้จากวงจรหน่วงเวลาเทียบกับสัญญาณฮอร์ซิงค์ ซึ่งผลที่ออกมาสามารถสร้างสัญญาณที่มีค่าหน่วงเวลา 10.5 uS เมื่อเทียบกับสัญญาณฮอร์ซิงค์ ได้ตามต้องการ (พิจารณาเทียบที่ขอบขาลงของสัญญาณทั้งสอง)

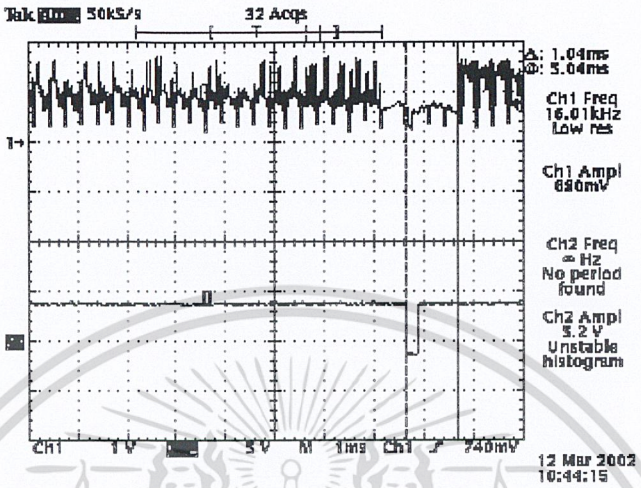


รูปที่ 4.12 สัญญาณ HOR SYNC (CH1) เทียบกับ สัญญาณ H-COUNT (CH2)

รูปที่ 4.12 แสดงถึงสัญญาณเอาต์พุตที่ออกจากวงจรผลิตพัลส์เทียบกับสัญญาณฮอร์ซิงค์ ซึ่งจะพบว่าสัญญาณที่ออกจากวงจรผลิตพัลส์จะเกิดหลังสัญญาณซิงค์รวมประมาณ 10.5 uS ตามต้องการ

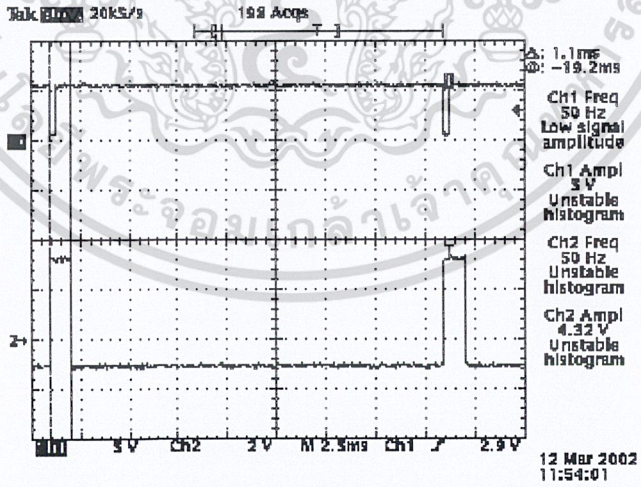
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเวอร์ซิงค์ที่ออกจากขา 3 ของ U6 จะถูกส่งป้อนเข้าที่วงจรกำเนิดสัญญาณควบคุมการชี้แอดเดรสทางด้านแนวตั้ง ซึ่งในวงจรนี้จะประกอบไปด้วย 2 ส่วน คือ ส่วนของวงจรหน่วงเวลาเวอร์ซิงค์ และวงจรผลิตพัลส์โดยรูปที่ 4.13 - 4.15 จะแสดงสัญญาณเอาต์พุตที่ได้จากวงจร



รูปที่ 4.13 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ VER SYNC (CH2)

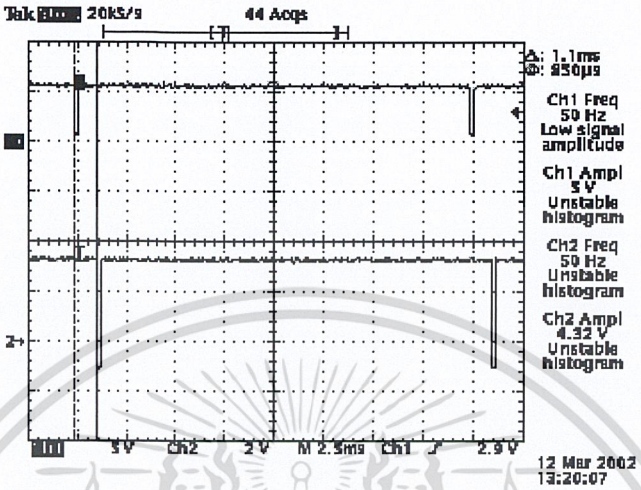
รูปที่ 4.13 แสดงถึงการเปรียบเทียบผลต่างทางเวลาระหว่างการเริ่มต้นการสแกนฟิลด์ใหม่ของสัญญาณภาพรวมกับสัญญาณขอบขาลงของสัญญาณเวอร์ซิงค์ ซึ่งค่าที่วัดได้มีค่าประมาณ 1.05 mS ซึ่งเป็นไปตามทฤษฎีจากนั้นจะนำค่าเวลานี้ไปทำการสร้างวงจรหน่วงเวลา



รูปที่ 4.14 สัญญาณ VER SYNC (CH1) เทียบกับ สัญญาณ ขา 5 ของ U8B (CH2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 แสดงถึงสัญญาณเอาต์พุตที่ได้จากวงจรหน่วงเวลาเทียบกับสัญญาณเวอร์ซิงค์ ซึ่งผลที่ออกมาสามารถสร้างสัญญาณที่มีค่าหน่วงเวลา 1.1 mS ซึ่งใกล้เคียงกับค่า 1.05 mS ที่ต้องการ (พิจารณาเทียบที่ขอบขาลงของสัญญาณทั้งสอง)

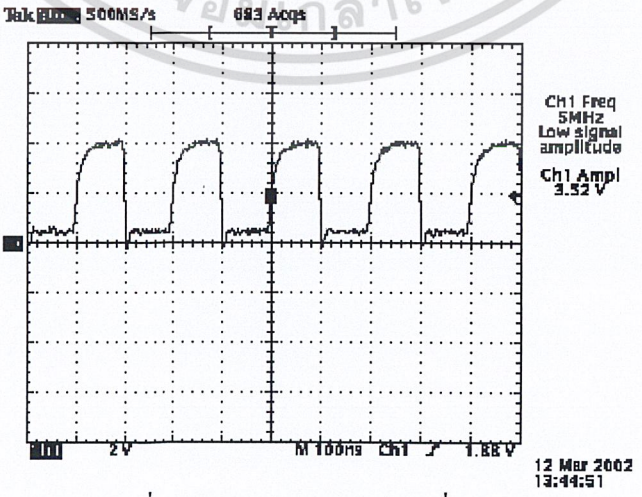


รูปที่ 4.15 สัญญาณ VER SYNC (CH1) เทียบกับ สัญญาณ V-COUNT (CH2)

รูปที่ 4.15 แสดงถึงสัญญาณเอาต์พุตที่ออกจากวงจรผลิตพัลส์เทียบกับสัญญาณเวอร์ซิงค์ ซึ่งจะพบว่าสัญญาณที่ออกจากวงจรผลิตพัลส์จะเกิดหลังสัญญาณเวอร์ซิงค์ 1.1 mS ซึ่งใกล้เคียงกับค่า 1.05 mS

4.3 ภาคชี้แอดเดรสหน่วยความจำ

ภาคชี้แอดเดรสหน่วยความจำ ทำหน้าที่ในการชี้แอดเดรสเพื่อเขียนข้อมูลลงในหน่วยความจำ โดยแสดงวงจรดังในรูปที่ 3.4 ซึ่งในภาคนี้จะประกอบไปด้วยวงจรกำเนิดสัญญาณนาฬิกา 5 MHz, วงจรการชี้แอดเดรส 8 บิตล่าง, วงจรการชี้แอดเดรส 8 บิตบน

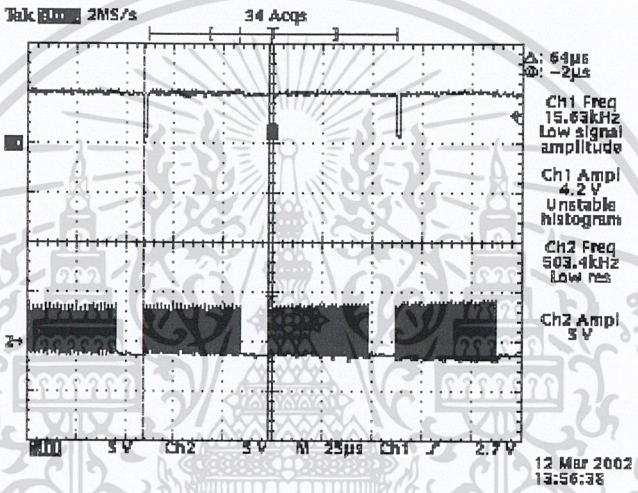


รูปที่ 4.16 สัญญาณนาฬิกาความถี่ 5 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.16 แสดงถึงสัญญาณนาฬิกาที่ใช้ในการป้อนให้กับวงจรชี้แอดเดรสหน่วยความจำ ซึ่งค่าความถี่ที่ใช้งานจะมีค่าเท่ากับ 5 MHz สาเหตุที่ต้องใช้ค่าความถี่เท่านี้เนื่องจากในโครงงานนี้เราต้องการหาความละเอียดของข้อมูลภาพ 256 × 256 จุด ซึ่งในแต่ละเส้นสแกนภาพมีคาบเวลาเท่ากับ 64 uS แต่ละช่วงเวลาที่มีสัญญาณภาพจะมีค่าประมาณ 80 % ของเวลาในการสแกนภาพแต่ละเส้น (64 uS) ซึ่งจะมีค่าเท่ากับ 51.2 uS จากนั้นจึงนำค่าที่ได้ไปหารด้วยจำนวนจุดทางแนวนอนที่มีค่าเท่ากับ 256 จุดเราจะได้ค่าเวลาแต่ละจุดเท่ากับ 200 nS ซึ่งก็คือความถี่ค่าเท่ากับ 5 MHz

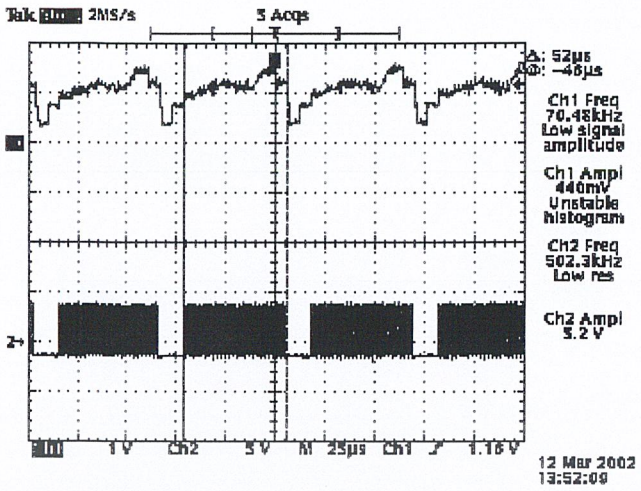
รูปที่ 4.17 - 4.18 จะแสดงถึงสัญญาณเอาต์พุตที่ออกจากวงจรการชี้แอดเดรส 8 บิตล่าง เทียบกับสัญญาณ H-COUNT จากภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำและสัญญาณภาพรวมจากกล้องวีดีโอตามลำดับ



รูปที่ 4.17 สัญญาณ H-COUNT (CH1) เทียบกับ สัญญาณ CA0 (CH2)

รูปที่ 4.17 แสดงถึงการเปรียบเทียบระหว่างสัญญาณ H-COUNT ที่ได้มาจากภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำกับสัญญาณเอาต์พุตที่ได้มาจากวงจรการชี้แอดเดรส 8 บิตล่าง (ในที่นี้ใช้ขา CA0 เป็นตัวเปรียบเทียบ) เพื่อเป็นการตรวจสอบว่าวงจรการชี้แอดเดรส 8 บิตล่างทำงานได้ถูกต้องหรือไม่ ซึ่งจากรูปจะเห็นว่าวงจรการชี้แอดเดรส 8 บิตล่างจะเริ่มทำการชี้แอดเดรสก็ต่อเมื่อมีสัญญาณขอบขาลงของ H-COUNT เข้ามากระตุ้น

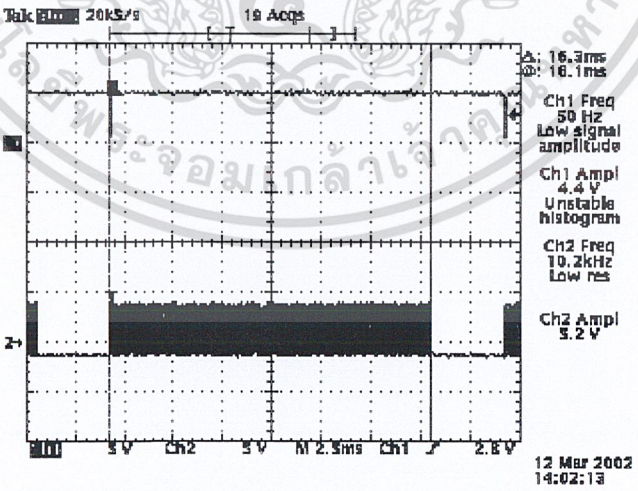
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 สัญญาณภาพรวม (CH1) เทียบกับ สัญญาณ CA0 (CH2)

รูปที่ 4.18 แสดงถึงการเปรียบเทียบระหว่างสัญญาณภาพรวมจากกล้องวีดีโอกับสัญญาณเอาต์พุตจากวงจรการชี้แอดเดรส 8 บิตล่าง (ในที่นี้ใช้ขา CA0 เป็นตัวเปรียบเทียบ) เพื่อเป็นการตรวจสอบว่าวงจรการชี้แอดเดรส 8 บิตล่างทำงานได้ถูกต้องหรือไม่ ซึ่งจากรูปจะเห็นว่าวงจรการชี้แอดเดรส 8 บิตล่าง จะเริ่มทำการชี้แอดเดรสทุกครั้งที่มีการเริ่มเส้นสแกนใหม่ และจะหยุดทำการการชี้แอดเดรสที่จุดสิ้นสุดของสัญญาณภาพ ซึ่งผลที่ได้ดังกล่าวก็ถูกต้องเป็นไปตามต้องการ

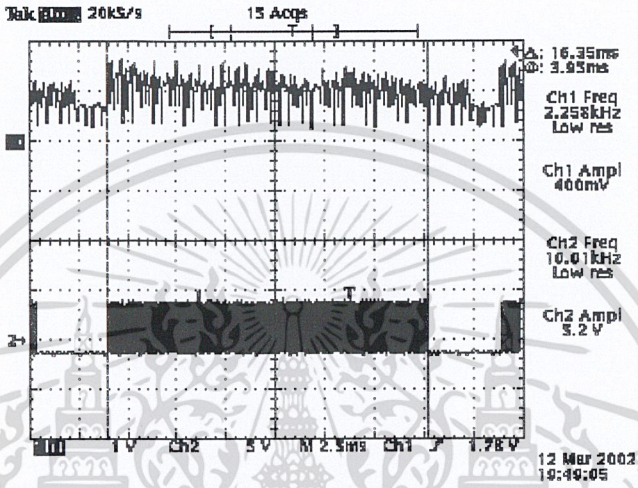
รูปที่ 4.19 - 4.20 จะแสดงถึงสัญญาณเอาต์พุตที่ออกจากวงจรการชี้แอดเดรส 8 บิตบนเทียบกับสัญญาณ V-COUNT จากภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำและสัญญาณภาพรวมจากกล้องวีดีโอตามลำดับ



รูปที่ 4.19 สัญญาณ V-COUNT (CH1) เทียบกับ สัญญาณ CA8 (CH2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 แสดงถึงการเปรียบเทียบระหว่างสัญญาณ V-COUNT ที่ได้มาจากภาคกำเนิดสัญญาณควบคุมการชี้แอดเดรสหน่วยความจำกับสัญญาณเอาต์พุตที่ได้มาจากวงจรการชี้แอดเดรส 8 บิตบน (ในที่นี้ใช้ขา CA8 เป็นตัวเปรียบเทียบ) เพื่อเป็นการตรวจสอบว่าวงจรการชี้แอดเดรส 8 บิตบนทำงานได้ถูกต้องหรือไม่ ซึ่งจากรูปจะเห็นว่าวงจรการชี้แอดเดรส 8 บิตบนจะเริ่มทำการชี้แอดเดรสก็ต่อเมื่อมีสัญญาณขอบขาลงของ V-COUNT เข้ามากระตุ้น



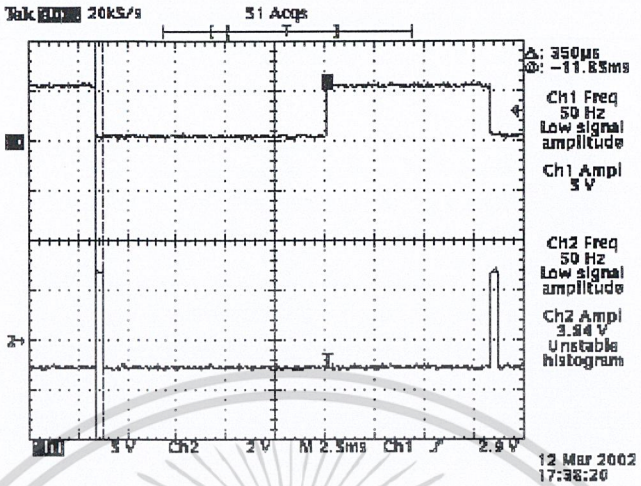
รูปที่ 4.20 สัญญาณภาพรวม (CH1) เทียบกับ ขา CA8 (CH2)

รูปที่ 4.20 แสดงถึงการเปรียบเทียบระหว่างสัญญาณภาพรวมจากกล้องวีดีโอกับสัญญาณเอาต์พุตจากวงจรการชี้แอดเดรส 8 บิตบน (ในที่นี้ใช้ขา CA8 เป็นตัวเปรียบเทียบ) เพื่อเป็นการตรวจสอบว่าวงจรการชี้แอดเดรส 8 บิตบนทำงานได้ถูกต้องหรือไม่ ซึ่งจากรูปจะเห็นว่าวงจรการชี้แอดเดรส 8 บิตบนจะเริ่มทำการชี้แอดเดรสทุกครั้งที่มีการเริ่มจากรูปจะพบว่า วงจรการชี้แอดเดรส 8 บิตบนจะหยุดชี้แอดเดรสก่อนที่จะสิ้นสุดฟิลด์แต่ละฟิลด์ เนื่องจากในแต่ละฟิลด์ของการสแกนภาพถ้าหากไม่รวมช่วงของการสะบัดกลับทางด้านแนวตั้งจะประกอบไปด้วยเส้นสแกนประมาณ 308 เส้น แต่เนื่องจากในโครงงานนี้เราทำการเก็บข้อมูล 1 ฟิลด์ลงในหน่วยความจำโดยใช้เส้นสแกนจำนวน 256 เส้นเท่านั้นจึงทำให้วงจรการชี้แอดเดรส 8 บิตบนหยุดทำการชี้แอดเดรสก่อนที่จะสิ้นสุดฟิลด์แต่ละฟิลด์

วงจรการชี้แอดเดรส 8 บิตบนนี้จะช่วยให้การเขียนข้อมูลภาพลงหน่วยความจำเริ่มต้นขึ้น เมื่อเริ่มต้นฟิลด์ใหม่ของสัญญาณภาพ ทำให้ภาพที่เก็บลงหน่วยความจำไม่เกิดการผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ภาคสร้างสัญญาณตอบรับ



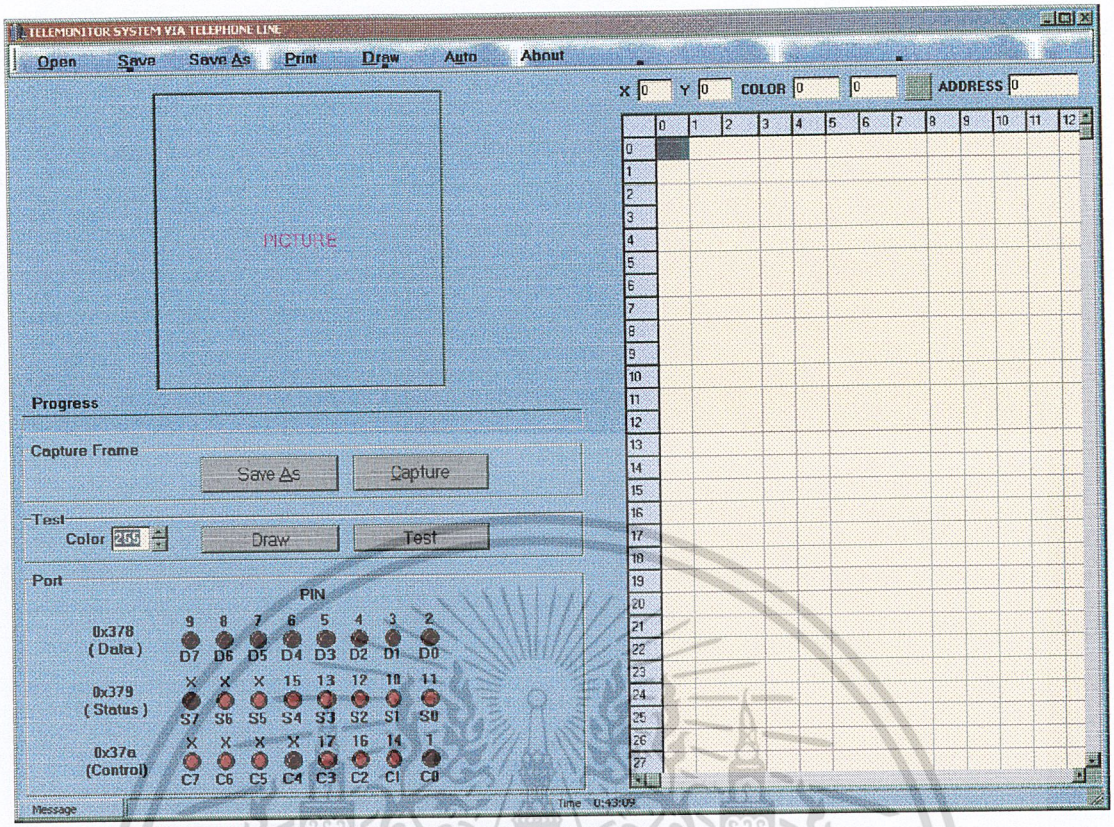
รูปที่ 4.21 สัญญาณ ขา CA15 (CH1) เทียบกับ สัญญาณ ACK (CH2)

รูปที่ 4.21 แสดงสัญญาณ ACK ที่ใช้ในการแจ้งให้คอมพิวเตอร์ทราบว่า ในขณะนี้ฮาร์ดแวร์ได้ทำการเขียนข้อมูลครบ 1 ฟิลด์แล้ว โดยสัญญาณนี้จะสร้างได้จากการนำแอดเดรส CA15 มาทำการสร้างสัญญาณพัลส์ไปแจ้งที่ขา 15 ของพอร์ตขนานสาเหตุที่ใช้แอดเดรส CA15 เนื่องจากเป็นแอดเดรสที่มีค่าบิตสูงสุดและการเขียนข้อมูลจะครบ 1 ฟิลด์ก็ต่อเมื่อขาแอดเดรสนี้เปลี่ยนสถานะจาก '1' เป็น '0'

4.5 การทดสอบระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์

ระบบตรวจสอบทางไกลผ่านทางคู่สายโทรศัพท์จะประกอบไปด้วย3ส่วนใหญ่ของคอมพิวเตอร์เซิร์ฟเวอร์,คอมพิวเตอร์ไคลเอนท์และตัวจับภาพ ในการทดสอบจะเริ่มจากการทดสอบโดยใช้ไคลเอนท์ทำการควบคุมตัวจับภาพ จากนั้นจึงทำการทดสอบโดยใช้เซิร์ฟเวอร์มาทำการควบคุมไคลเอนท์ให้ทำการจับภาพมาแสดงที่หน้าจอของเซิร์ฟเวอร์

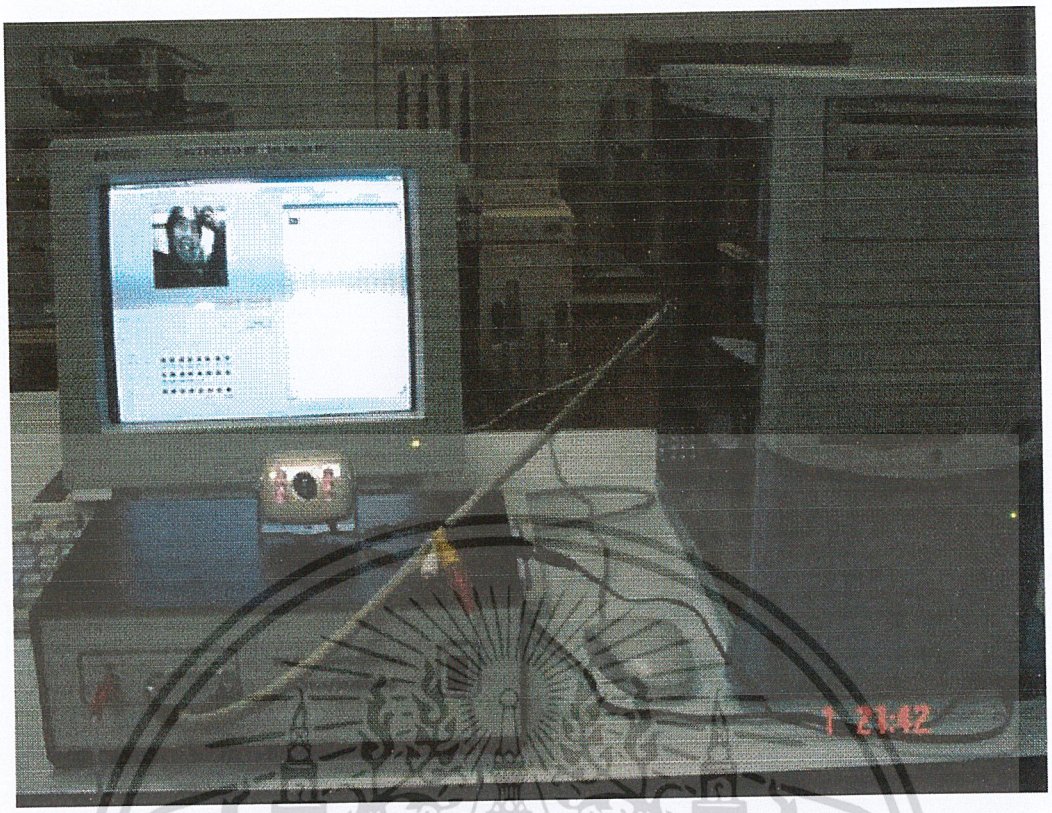
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



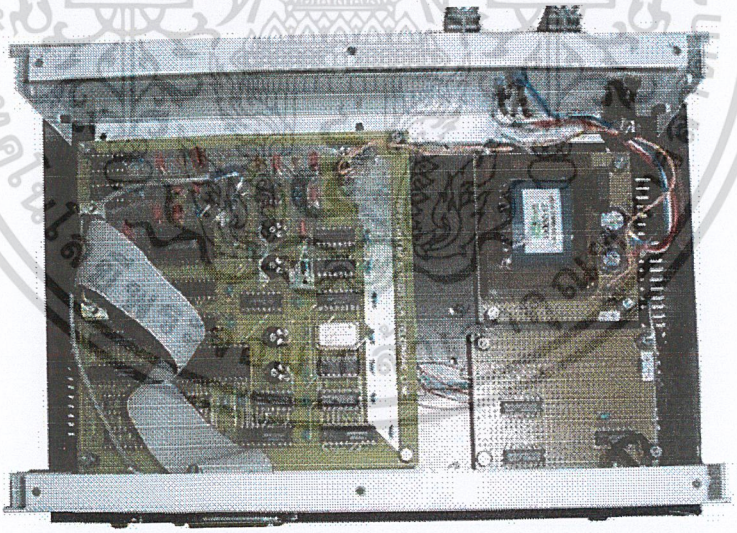
รูปที่ 4.22 แสดงหน้าต่างหลักของซอฟต์แวร์ที่สร้างขึ้นสำหรับโครงงานนี้

รูปที่ 4.23 แสดงหน้าต่างแสดงรายละเอียดของซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แสดงโครงงานขณะกำลังจับภาพจากกล้องวงจรปิด



รูปที่ 4.25 แสดงภายในกล่องโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.26 แสดงภาพด้านหน้าของกล่องโครงงาน

รูปที่ 4.27 แสดงภาพด้านหลังของกล่องโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
บทวิจารณ์และบทสรุป

จากโครงงานสามารถทำการทดลองการทำงานส่วนต่าง ๆ ของวงจรได้ดังนี้

ภาคแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล วงจรสามารถสร้างสัญญาณดิจิตอลออกมาได้ถูกต้องซึ่งสามารถตรวจสอบได้โดยใช้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก(ซึ่งได้แสดงวงจรไว้ในภาคผนวก)มาตรวจสอบ

ภาคการแยกสัญญาณซิงค์ สามารถแยกสัญญาณฮอร์ซิงค์และเวอร์ซิงค์ได้ถูกต้องตามทฤษฎี

ภาคกำเนิดสัญญาณควบคุมเพื่อชี้แอดเดรสหน่วยความจำ สามารถสร้างสัญญาณควบคุมให้ภาคกชี้แอดเดรสหน่วยความจำทำงานได้ตามต้องการ

ภาคชี้แอดเดรสหน่วยความจำสำหรับการเขียน สามารถสร้างสัญญาณชี้แอดเดรสในการเขียนข้อมูลลงหน่วยความจำได้ตามต้องการ ซึ่งทำการตรวจสอบโดยการใช้วงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกมาทำการแปลงข้อมูลจากหน่วยความจำมาแสดงที่หน้าจอโทรทัศน์

ภาคชี้แอดเดรสหน่วยความจำสำหรับการอ่าน สามารถสร้างสัญญาณชี้แอดเดรสในการอ่านข้อมูลได้ถูกต้องตามต้องการ ซึ่งทำการตรวจสอบได้จากภาพที่ปรากฏที่หน้าจอคอมพิวเตอร์ว่าเกิดการล้มหรือเบี้ยวหรือไม่

ภาคเลือกเขียนหรืออ่านหน่วยความจำ สามารถเลือกเขียนหรืออ่านข้อมูลได้ตามการสั่งงานจากคอมพิวเตอร์

ภาคหน่วยความจำ สามารถรับข้อมูลที่สั่งให้เขียนและส่งข้อมูลเมื่อสั่งให้อ่านได้ตามคำสั่งจากคอมพิวเตอร์

วงจรกำเนิดสัญญาณตอบรับ สามารถสร้างพัลส์แจ้งคอมพิวเตอร์ได้ถูกต้องตามที่ออกแบบ

ภาคอินเตอร์เฟส สามารถเซตค่ารีจิสเตอร์ของพอร์ทขนานให้ทำงานได้ถูกต้องตามที่ออกแบบ

โปรแกรมจับภาพ สามารถทำการสั่งงานให้ตัวจับภาพทำการจับภาพและนำภาพแสดงที่หน้าจอคอมพิวเตอร์ได้ถูกต้อง

โปรแกรม Remote Anything สามารถใช้ติดต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้

ปัญหาของระบบ

ในโครงงานนี้มีปัญหาและข้อจำกัดต่าง ๆ ดังนี้

1. ไอซีเบอร์ CA3318 ซึ่งใช้ในการแปลงสัญญาณสัญญาณอนาลอกเป็นดิจิตอลหาได้ยากและมีราคาแพงเนื่องจากปัจจุบันไอซีเบอร์นี้ได้เลิกผลิตแล้ว
2. การศึกษาสถานะลอจิกของพอร์ตขนานที่ขาของ DB–25 และรีจิสเตอร์มีบางขาที่มีสถานสลับกันจึงต้องใช้เวลาในการศึกษาพอสมควรเนื่องจากในช่วงแรกที่ทำการทดสอบพอร์ตข้อมูลไม่สามารถรับข้อมูลได้ซึ่งการที่จะให้พอร์ตข้อมูลรับข้อมูลได้จะต้องเซตบิต 5 ของพอร์ควบคุมให้เป็น ‘1’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไอซีที่ใช้ในการชี้แอดเดรสหน่วยความจำสำหรับการอ่านในตอนแรกใช้ไอซีเบอร์ 74LS393 ไม่สามารถที่จะชี้แอดเดรสได้ตามต้องการ ทำให้ไม่สามารถนำข้อมูลจากหน่วยความจำมาแสดงที่จอคอมพิวเตอร์ได้จึงเปลี่ยนมาใช้ไอซีเบอร์ 74HC393 ซึ่งสามารถทำการชี้แอดเดรสได้ตามที่ออกแบบไว้

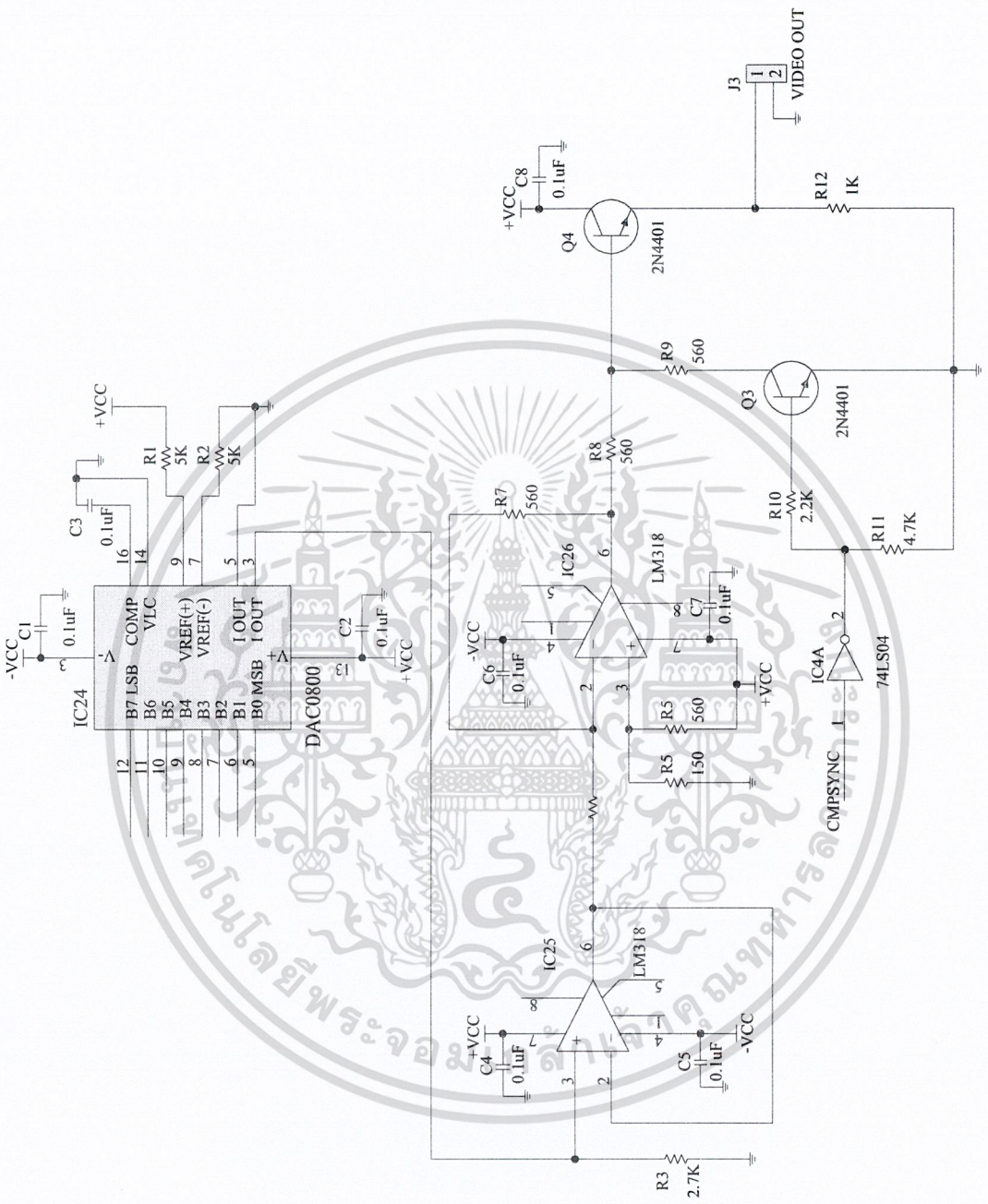
แนวทางในการพัฒนาโครงงาน

1. พัฒนาความเร็วในการเขียน/อ่านข้อมูลภาพโดยอาจใช้หน่วยความจำ 2 ตัวสลับการทำงาน คือเมื่อตัวหนึ่งถูกเขียนข้อมูลก็ทำการอ่านข้อมูลจากหน่วยความจำอีกตัวซึ่งอาจจะต้องเพิ่มภาคควบคุมการเขียน/อ่านภายนอกขึ้นมา
2. ทำการแสดงภาพเป็นภาพสีโดยจะต้องทำการเพิ่มหน่วยความจำเป็น 3 ตัวเพื่อทำการเก็บข้อมูลภาพที่เป็นแม่สีทางแสงคือแดง, เขียว และน้ำเงิน
3. ทำการติดต่อข้อมูลระหว่างต้นทางและปลายทางโดยการเขียนคำสั่งผ่านโมเด็มโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมติดต่อกับพอร์ตขนานของBorland C++ Builder

โปรแกรมสำหรับส่งข้อมูลออกทางพอร์ตขนานครั้งละ 1 ไบต์

```
void Outp(unsigned short pAddress , unsigned short pValue)
     {
          asm {
                    mov dx,pAddress
                    mov al,pValue
                    out dx,al
               }
     }
```

รูปแบบการเรียกใช้งาน

```
   Outp(Address Port,Value to Port)
```

โปรแกรมสำหรับรับข้อมูลเข้าทางพอร์ตขนานครั้งละ1ไบต์

```
unsigned char Inp( pAddress)
{
   unsigned char  pStatus;
   asm {
             mov dx,pAddress
             in al,dx
             mov pStatus,al
        }
    return pStatus;
}
```

รูปแบบการเรียกใช้งาน

```
   value=Inp(pAddress)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการอ่านและเขียนข้อมูลระบบตรวจสอบทางไกลผ่านคู่สายโทรศัพท์

```
#include <vcl.h>
#include <vcl\printers.hpp>
#pragma hdrstop
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include "ts.h"
#include"ports.h"
#include "about.h"
#include <math.h>
#define port 0x378


int line=1,dot=1,i=1,j=1;
int pix,ack=0x01,data[256][256];
int aut=0; //  status auto button


#pragma package(smart_init)
#pragma link "zLed"
#pragma link "zAnimate"
#pragma link "Std2"
#pragma resource "*.dfm"
TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
 Application->Title = " TS KMITL ";
}

void __fastcall TForm1::T_TimeTimer(TObject *Sender)
{
 StatusBar1->Panels->Items[2]->Text="  Time     "+TimeToStr(Time());
}


void __fastcall TForm1::TB_OpenClick(TObject *Sender)
{
 if(OpenPictureDialog1->Execute())
    {
    Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);

       Caption ="  TS  "+  OpenPictureDialog1->FileName;
    }
}

void __fastcall TForm1::TB_SaveClick(TObject *Sender)
{
  if(OpenPictureDialog1->FileName!="")
 {
 Image1->Picture->SaveToFile(OpenPictureDialog1->FileName);
 }
 else
 TB_SaveAsClick(Sender);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void __fastcall TForm1::TB_SaveAsClick(TObject *Sender)
{
 if(SavePictureDialog1->Execute())
  {
  Image1->Picture->SaveToFile(SavePictureDialog1->FileName);
  }
}

void __fastcall TForm1::Draw(TObject *Sender)
{
   int  x, y,nPixelColor,nRed,nGreen,nBlue,nGray;
    // use bmp and class TBitmap
   Graphics::TBitmap *bmpProcess = new Graphics::TBitmap();
    // draw into image
   TCanvas *IpCanvas = Image1->Canvas;
  try{
        bmpProcess->Width  = 255;
        bmpProcess->Height = 255;
        bmpProcess->Assign(Image1->Picture);//Assign(bmpSource);
        bmpProcess->Dormant();              // Free up GDI resources
        bmpProcess->FreeImage();            // Free up Memory.
   for(x=0;x<=255;x++)
    {
     for(y=0;y<=255;y++)
     {
      bmpProcess->Canvas->Pixels[x][y]  =data[x][y]+(data[x][y]<<
                                          8)+(data[x][y]<< 16);

     }
      PB_Data->Position=x;
   }
      //draww
       IpCanvas->Draw(0,0,bmpProcess);
       bmpProcess->ReleaseHandle();
   } // end try
     catch (...)
  {
    MessageBeep(0);
  }
      delete bmpProcess;
}

void __fastcall TForm1::FormCreate(TObject *Sender)

{
  int n;
  for(n=0;n<=255;n++)
      {
       SG_Data->Cells[n+1][0]=n;
       SG_Data->Cells[0][n+1]=n;
      }
   Application->OnHint=&OnHint;
}


void __fastcall TForm1::SG_DataClick(TObject *Sender)

{
 int n,dt,data;
 char datahex[2];
 char rBuf[10];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 char cBuf[10];
 int x,y,c;

  itoa(SG_Data->Col,cBuf, 10);
  itoa(SG_Data->Row,rBuf, 10);

 x=StrToInt(cBuf)-1;
 y=StrToInt(rBuf)-1;

 if(SG_Data->Cells[x][y]!="")  ///// portect error ""
 {

 E_CDec->Text=SG_Data->Cells[x+1][y+1];


  c=StrToInt(E_CDec->Text);
     data=c;

    for(n=0;n<=2;n++)
    {
       dt=data%16;

       switch(dt)
       {
        case 0:  datahex[n]='0'; break;
        case 1:  datahex[n]='1'; break;
        case 2:  datahex[n]='2'; break;
        case 3:  datahex[n]='3'; break;
        case 4:  datahex[n]='4'; break;
        case 5:  datahex[n]='5'; break;
        case 6:  datahex[n]='6'; break;
        case 7:  datahex[n]='7'; break;
        case 8:  datahex[n]='8'; break;
        case 9:  datahex[n]='9'; break;
        case 10: datahex[n]='A'; break;
        case 11: datahex[n]='B'; break;
        case 12: datahex[n]='C'; break;
        case 13: datahex[n]='D'; break;
        case 14: datahex[n]='E'; break;
        case 15: datahex[n]='F'; break;

       }

       data=data/16;
    }

 E_CHex->Text="0x"+(AnsiString)datahex[1]+(AnsiString)datahex[0];
 P_C->Color=c+(c<<8)+(c<<16);

 E_X->Text=IntToStr(x);
 E_Y->Text=IntToStr(y);
 E_Address->Text=IntToStr((x*255)+y);
 }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void __fastcall TForm1::TB_PrintClick(TObject *Sender)
{
 if(PrintDialog1->Execute())
 {
  Printer()->BeginDoc();
  Printer()->Canvas->Draw(0,0,Image1->Picture->Bitmap);
  Printer()->EndDoc();
 }
}

void __fastcall TForm1::TB_AutoClick(TObject *Sender)
{

  aut =! aut;

  if(aut==0)
  {
   TB_Auto->Caption="Auto";
   T_Read->Enabled=false;

  }
  else
  {

   TB_Auto->Caption="Off";

   T_Read->Enabled=true;

  }

}

void __fastcall TForm1::T_ReadTimer(TObject *Sender)
{
    zCB_StartClick(Sender);
}

void __fastcall TForm1::zCB_DrawClick(TObject *Sender)
{
    for(i=0;i<256;i++)
        {
               for(j=0;j<256;j++)
          {
              pix=StrToInt(E_Color->Text);
              data[i][j]=pix;
               SG_Data->Cells[i+1][j+1]=data[i][j];
          }
        }

       Draw(Sender);     ////////// draw pic
}

void __fastcall TForm1::zCB_TestClick(TObject *Sender)
{
  for(i=0;i<=255;i++)
    {
     for(j=0;j<=255;j++)
     {
       data[i][j]= i;

       SG_Data->Cells[i+1][j+1]=data[i][j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }   /////////// end j
    }
    Draw(Sender); ////////////// draw
}

void __fastcall TForm1::zT_StatusTimer(TObject *Sender)
{
      int d;
      ////////////////////////////// data
      d=inp(0x378);
      DecToBin(d);

      zL_D7->Enabled=datap[7];
      zL_D6->Enabled=datap[6];
      zL_D5->Enabled=datap[5];
      zL_D4->Enabled=datap[4];
      zL_D3->Enabled=datap[3];
      zL_D2->Enabled=datap[2];
      zL_D1->Enabled=datap[1];
      zL_D0->Enabled=datap[0];

      ////////////////////////////// status
      d=inp(0x379);
      DecToBin(d);
      zL_S7->Enabled=datap[7];
      zL_S6->Enabled=datap[6];
      zL_S5->Enabled=datap[5];
      zL_S4->Enabled=datap[4];
      zL_S3->Enabled=datap[3];
      zL_S2->Enabled=datap[2];
      zL_S1->Enabled=datap[1];
      zL_S0->Enabled=datap[0];
      ////////////////////////////// control
      d=inp(0x37a);
      DecToBin(d);
      zL_C7->Enabled=datap[7];
      zL_C6->Enabled=datap[6];
      zL_C5->Enabled=datap[5];
      zL_C4->Enabled=datap[4];
      zL_C3->Enabled=datap[3];
      zL_C2->Enabled=datap[2];
      zL_C1->Enabled=datap[1];
      zL_C0->Enabled=datap[0];
}

void __fastcall TForm1::zCB_StartClick(TObject *Sender)
{
    int i;
    //************PRESET & RESET*****************//

            outp(port+2,0x02);
    //************WRITE STATE*********************//

            outp(port+2,0x01);


       TimeGO->Enabled=true;
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void __fastcall TForm1::TB_AboutClick(TObject *Sender)
{
 F_About->ShowModal();
}

 void __fastcall TForm1::OnHint(TObject *Sender)
{
  StatusBar1->Panels->Items[1]->Text=Application->Hint;
}

void __fastcall TForm1::TB_DrawClick(TObject *Sender)
{
  zCB_DrawClick(Sender);
}

void __fastcall TForm1::TimerGO(TObject *Sender)
{
UpUp(Sender);
}

void __fastcall TForm1::UpUp(TObject *Sender)
{
    outp(port+2,0x02);//reset hardware
         for(dot=0;dot<256;dot++)
              {
                for(line=0;line<256;line++)
              {
                  //////////////
                  outp(port+2,0x2c);//addressing every rising edge
logic'0'
              outp(port+2,0x28);//addressing every rising edge
logic'1'
              pix = inportb(port+0);
              data[line][dot]=pix;
                  SG_Data->Cells[line+1][dot+1]=data[line][dot];
                  }
              }
        Draw(Sender);     ////////// draw pic
        TimeGO->Enabled=false;
}

void __fastcall TForm1::zColorBtn1Click(TObject *Sender)
{
TB_SaveAsClick(Sender);
}


void __fastcall TForm1::zColorBtn1Click(TObject *Sender)
{
TB_SaveAsClick(Sender);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intersil

OBSOLETE PRODUCT
FOR A POSSIBLE SUBSTITUTE PRODUCT
call Central Applications 1-888-INTERSIL
or email: centapp@intersil.com

CA3318

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

August 2000

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.

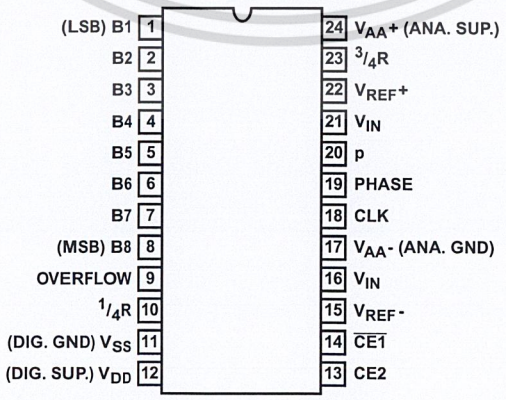
255 comparators are required to quantize all input voltage levels in this 8-bit converter, and the additional comparator is required for the overflow bit.

Part Number Information

PART NUMBER	LINEARITY (INL, DNL)	SAMPLING RATE	TEMP. RANGE (^{o}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

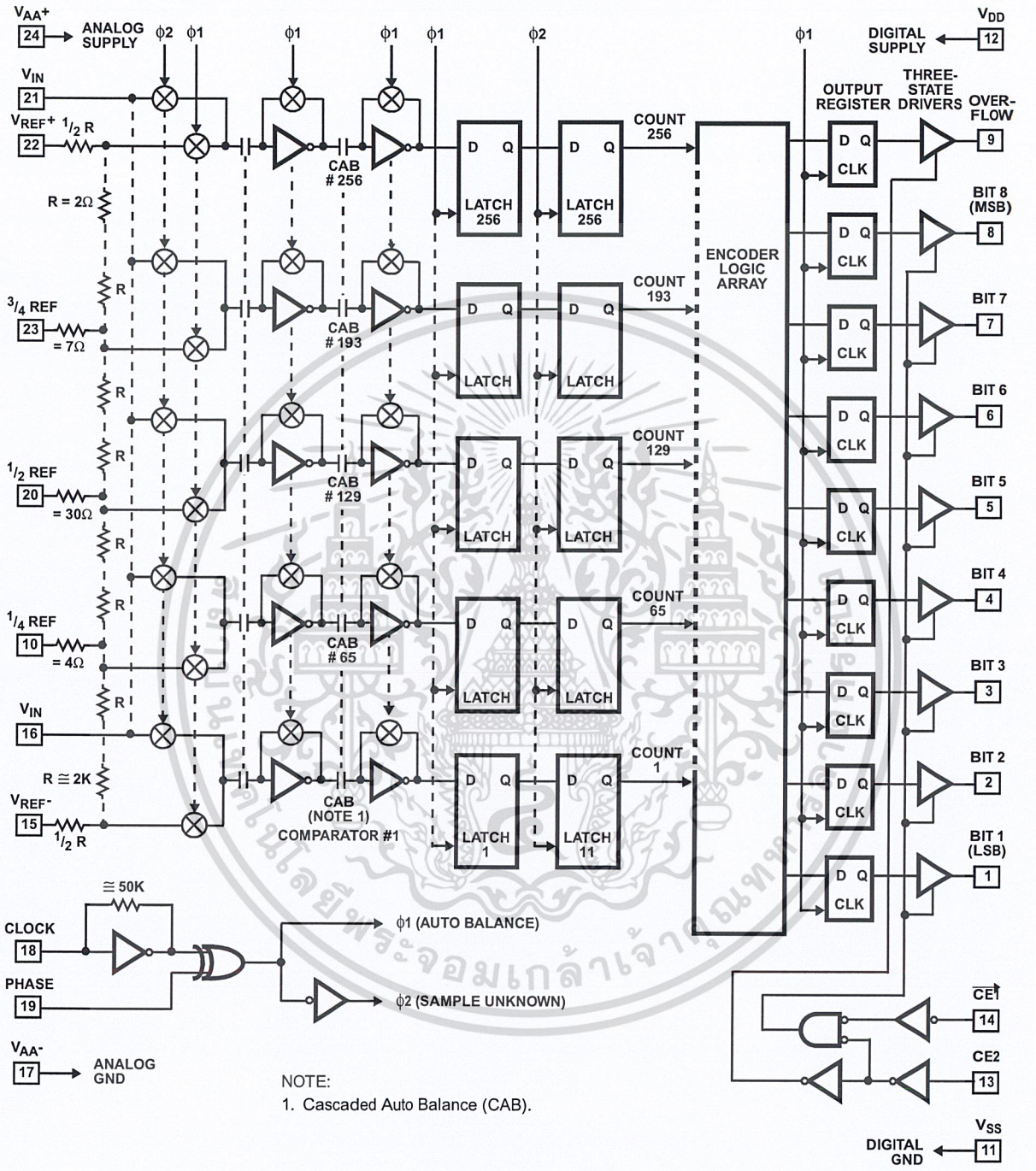


CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
1-888-INTERSIL or 321-724-7143 | Copyright © Intersil Corporation 2000

File Number **3103.2**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Block Diagram



NOTE:

1. Cascaded Auto Balance (CAB).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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} (°C/W)	θ_{JC} (°C/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) $=\frac{RMS\,Signal}{RMS\,Noise}$	f_S = 15MHz, f_{IN} = 100kHz	-	47	-	dB
	f_S = 15MHz, f_{IN} = 4MHz	-	43	-	dB
Signal to Noise Ratio (SINAD) $=\frac{RMS\,Signal}{RMS\,Noise+Distortion}$	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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

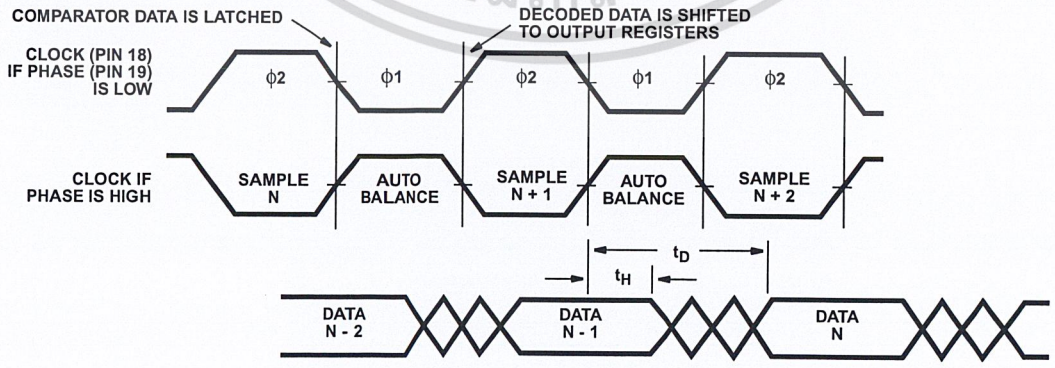


FIGURE 1. INPUT TO OUTPUT TIMING DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing Waveforms (Continued)

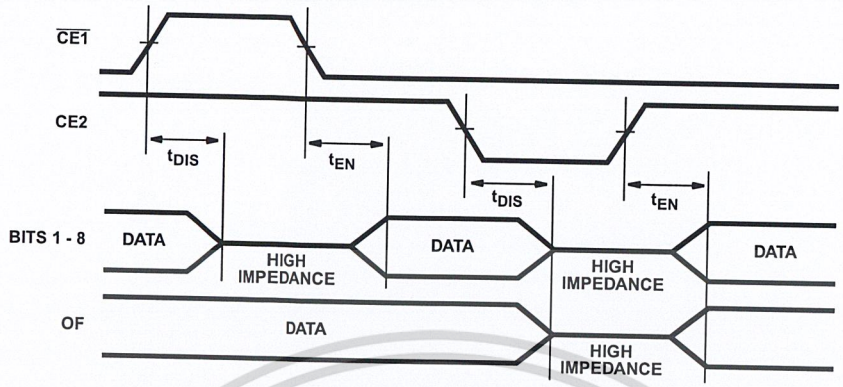


FIGURE 2. OUTPUT ENABLE TIMING DIAGRAM

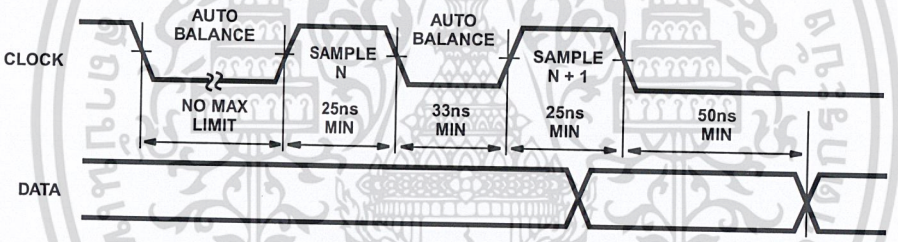


FIGURE 3A. STANDBY IN INDEFINITE AUTO BALANCE (SHOWN WITH PHASE = LOW)

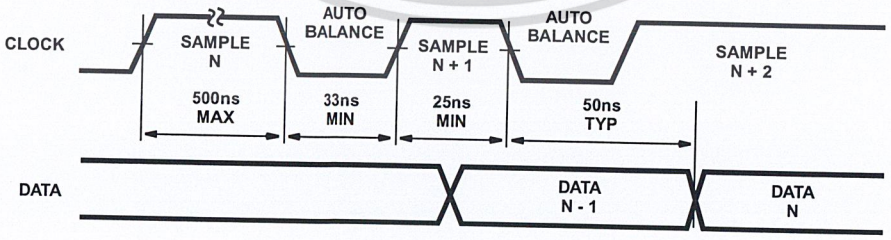


FIGURE 3B. STANDBY IN SAMPLE (SHOWN WITH PHASE = LOW)

FIGURE 3. PULSE MODE OPERATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves

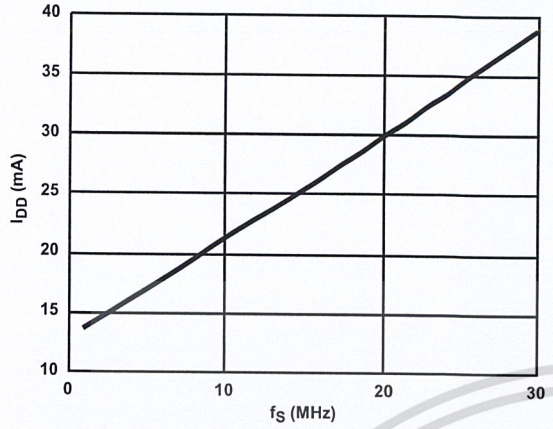


FIGURE 4. DEVICE CURRENT vs SAMPLE FREQUENCY

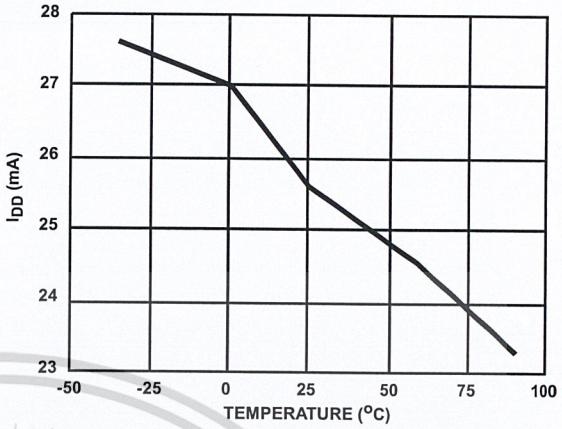


FIGURE 5. DEVICE CURRENT vs TEMPERATURE

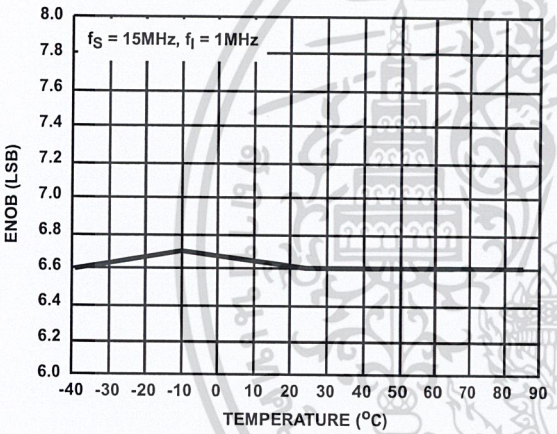


FIGURE 6. ENOB vs TEMPERATURE

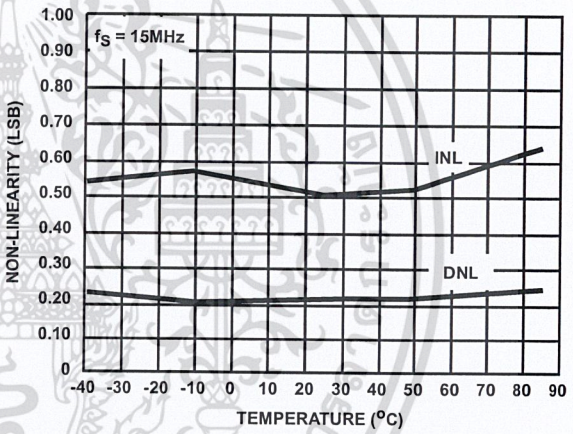


FIGURE 7. NON-LINEARITY vs TEMPERATURE

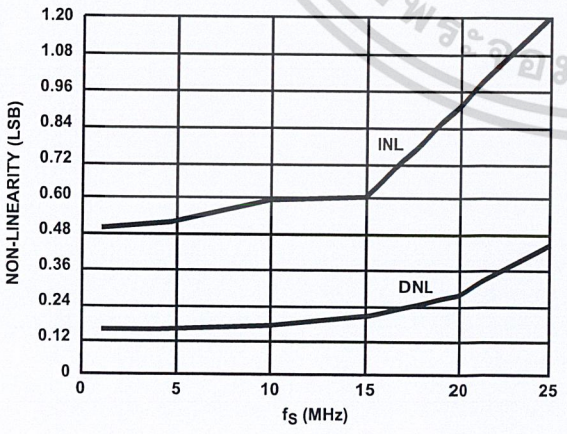


FIGURE 8. NON-LINEARITY vs SAMPLE FREQUENCY

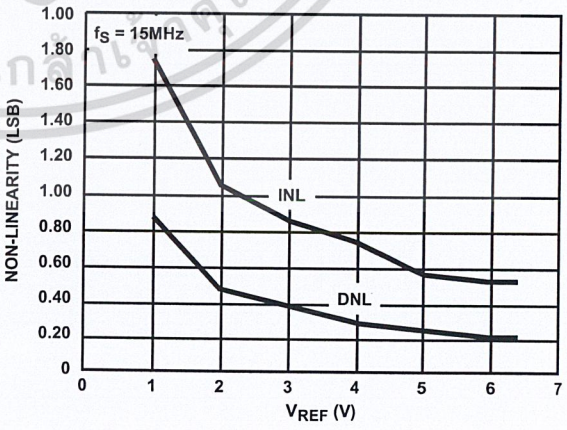


FIGURE 9. NON-LINEARITY vs REFERENCE VOLTAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves (Continued)

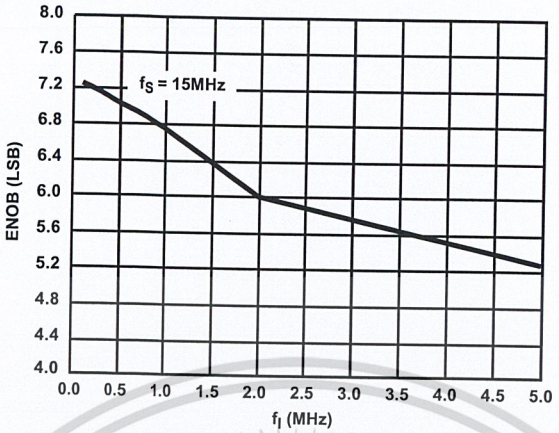


FIGURE 10. ENOB vs INPUT FREQUENCY

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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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:

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

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.

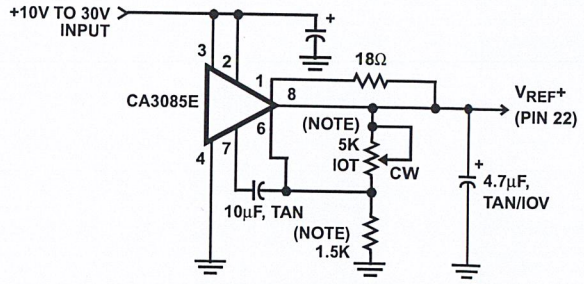
Gain Trim

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:

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

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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

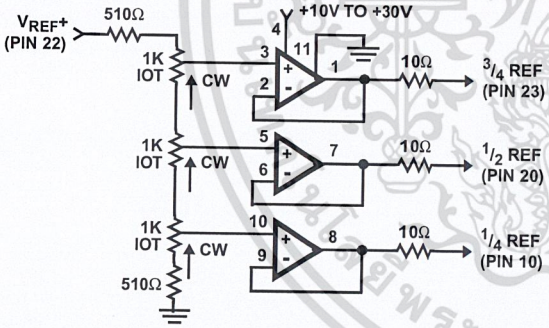


NOTE: Bypass $V_{REF}+$ to analog GND near A/D with 0.1µF ceramic cap. Parts noted should have low temperature drift.

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 overtlow 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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.

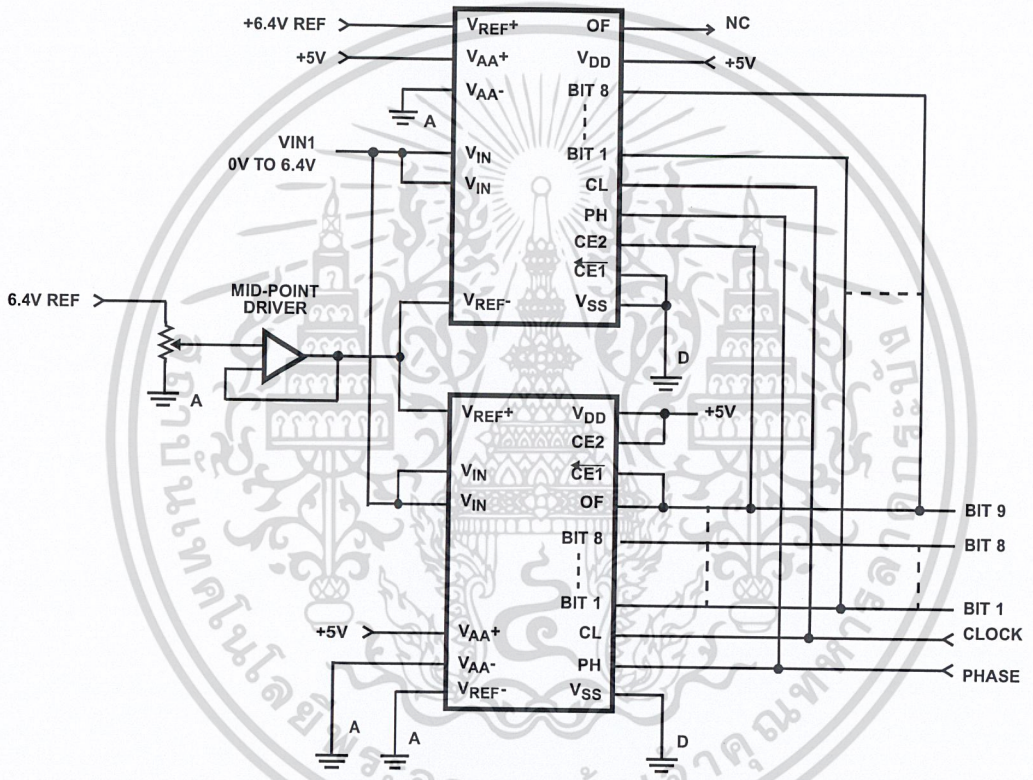


FIGURE 13. USING TWO CA3318s FOR 9-BIT RESOLUTION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

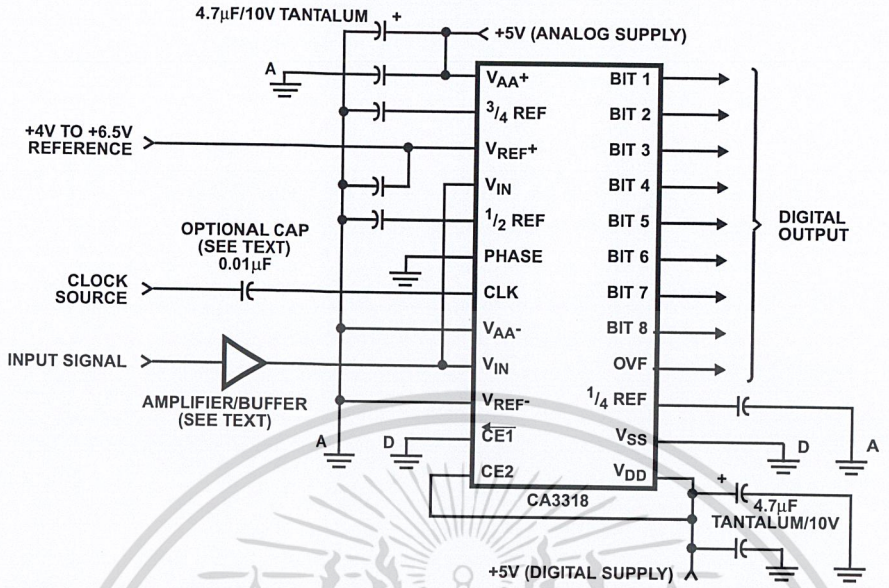


FIGURE 14. TYPICAL CIRCUIT CONFIGURATION FOR THE CA3318 WITH NO LINEARITY ADJUST

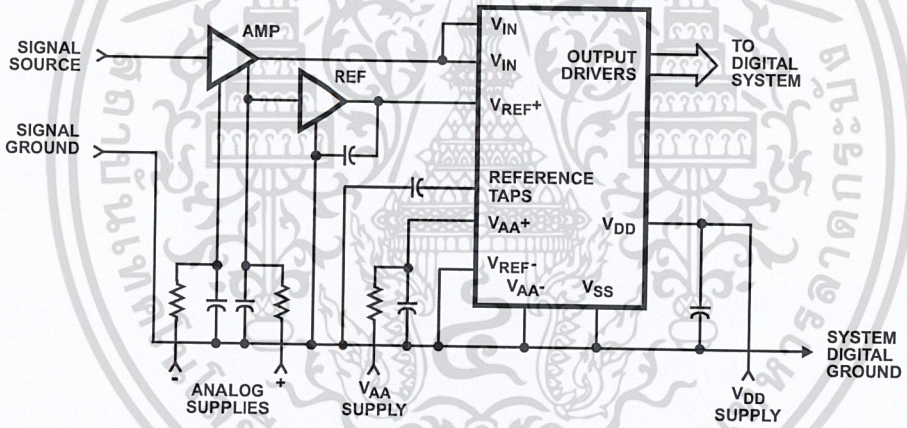
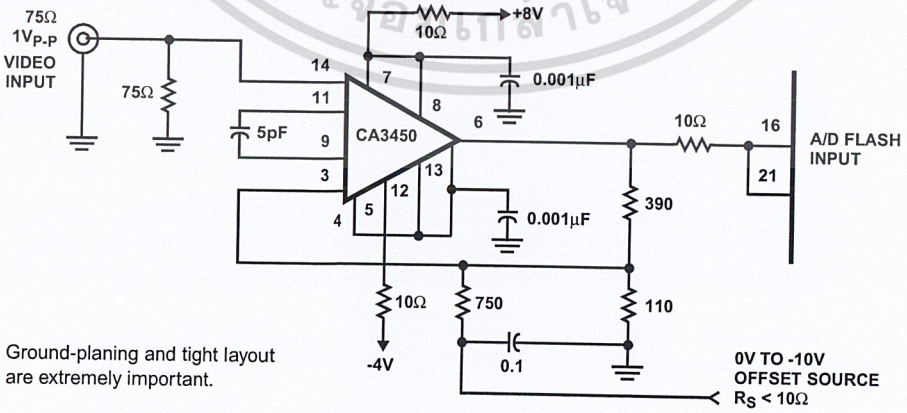


FIGURE 15. TYPICAL SYSTEM GROUNDING/BYPASSING



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

FIGURE 16. TYPICAL HIGH BANDWIDTH AMPLIFIER FOR DRIVING THE CA3318

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ต้องขอขอบพระคุณบิดามารดา ผู้ซึ่งให้โอกาสและการสนับสนุนที่ดีตลอดมา และกราบขอบพระคุณท่านอาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาคีพงษ์ ที่คอยให้คำปรึกษาและคำแนะนำต่างๆ ตลอดจนเครื่องมือและอุปกรณ์ต่างๆ ที่ใช้ในการทำโครงงานนี้ นอกจากนี้ต้องขอขอบคุณ นายพงศกร สีขาว นายชนาเนตร์ ชวรางกูร นายดิเรก เม่งเตี๋ยน รวมถึงเพื่อนๆ พี่น้องชาวพระนครเหนือทุกคน ทางคณะผู้จัดทำต้องขอขอบพระคุณไว้ ณ. ที่นี้ด้วย

คณะผู้จัดทำ
นายสัญญา สายเจริญ
นายอนุวัฒน์ นุ่มวงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1) อรรถสิทธิ์ หล่าสกุล, “การเก็บภาพขนาด 512*256 จุด โดยใช้หน่วยความจำที่มีความเร็วต่ำ”, วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2532

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

3) ชัยยุทธ์ ลิมลาวัลย์, “ควบคุมคอมพิวเตอร์ระยะไกล”, ซีเอ็ดยูเคชั่น, กรุงเทพฯ, 2544

4) นุกูล กระจาย, “การเขียนโปรแกรมแบบวิชวลด้วย C++ Builder 5”, สุวีริยาสาสน์, กรุงเทพฯ, 2544

5) Digital Principle and Applications, Malvino Leach, (THIRD EDITION)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้