IMAGE CARD DISPLAY 256*256 PIXELS

นายชูชัย	จรัสวรกุลกิจ	34131107
นายทวีศิลป์	หงษ์สวัสดิ์	34131109
นายประณีต	ตั้งทับสุนทร	34131115

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032667

ปีการศึกษา 2535

IMAGE CARD DISPLAY 256*256 PIXELS

จัดทำโดย

1.นายชูชัย	จรัสวรกุลกิจ	รหัส	34131107	ชั้น	2/1N
2.นายทวีศิลป์	หงษ์สวัสดิ์	รหัส	34131109	ชั้น	2/1N
3.นายประณีต	ตั้งทับสุนทร	รหัส	34131115	ชั้น	2/1N

อาจารย์ที่ปรึกษา

อาจารย์ ไพศาล สิทธิโยภาสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032667

ปริญญานิพนธ์ปีการศึกษา 2535

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาควิชาเทคนิคอุตสาหกรรม

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์(เช้า)

เรื่อง IMAGE CARD DISPLAY 256*256 PIXELS

จัดทำโดย

1.นายชูชัย	จรัสวรกุลกิจ	รหัส	34131107	ชั้น	2/1N
2.นายทวีศิลป์	หงษ์สวัสดิ์	รหัส	34131109	ชั้น	2/1N
3.นายประณีต	ตั้งทับสุนทร	รหัส	34131115	ชั้น	2/1N

อาจารย์ ไพศาล สิทธิโยภาสกุล อาจารย์ที่ปรึกษา

(.....)

ไพศาล สิทธิโยภาสกุล
---------- กรรมการ

(.....)

---------- กรรมการ

(.....)

---------- กรรมการ

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMAGE CARD DISPLAY 256*256 PIXELS

จัดทำโดย	1.นายชูชัย	จรัสวรกุลกิจ	34131107
	2.นายทวีศิลป์	หงษ์สวัสดิ์	34131109
	3.นายประณีต	ตั้งทับสุนทร	34131115

อาจารย์ที่ปรึกษา อาจารย์ ไพศาล สิทธิโยภาสกุล

ปีการศึกษา 2535

บทคัดย่อ

ในปริญญานิพนธ์นี้การประมวลผลภาพ(IMAGE PROCESSING)นั้น เราใช้เครื่องคอมพิวเตอร์ที่มีอยู่โดยทั่วไปคือ IBM PC/XT/AT หรือเครื่องที่ใช้แทนกันได้ ซึ่งสามารถหาได้ง่ายในปัจจุบัน และความละเอียดที่ต้องการใช้มีความละเอียด 256*256 จุดต่อ 1 ภาพ ในขณะเดียวกันปัจจุบันนี้ IBM PC มีราคาถูกลงและมีหน่วยความจำที่มากขึ้น จึงเป็นผลทำให้ประสิทธิภาพสูงขึ้นสามารถนำมาใช้ในการประมวลผลภาพทางดิจิตอลได้ แต่อุปกรณ์ที่จำเป็นต้องใช้ในการเปลี่ยนสัญญาณอนาล็อก เพื่อนำเข้ามาสู่คอมพิวเตอร์ยังมีราคาแพงมาก และส่วนมากยังเป็นของต่างประเทศ ดังนั้นจึงทำให้เกิดโครงงานนี้ขึ้นมาเพื่อให้ได้อุปกรณ์ในการประมวลผลภาพที่มีราคาถูกแต่สามารถนำมาใช้งานได้ จึงคิดว่าโครงงานนี้จะเป็นประโยชน์ต่องานทางด้าน IMAGE PROCESSING ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMAGE CARD DISPLAY 256*256 PIXELS

BY 1. Mr.CHUCHAI JARUSWORAKULKIJ 34131107

2. Mr.TAWEESIL HONGSAWAT 34131109

3. Mr.PRANIT TANGTAPSOONTON 34131115

ADVISER Mr.PRISAN SITTIYOPASKUL

STUDY YEAR 1992

ABSTRACT

This thesis is use associate the popular computer (IBM PC/XT/AT or compatible) because we have find easy present and resolution of card with it can express resolution 256*256 pixels/picture frame is posible. Now IBM PC has low cost and raise memory result increase efficiency with it is use image processing of digital but necessarily element of exchange analog signal into computer with it is expensive cost and this is of foreign. Since occupation project for become element image processing at low cost . However,it is win use. We think the project is advantage on the image processing job.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่1. บทนำ 1

แนวความคิดและขั้นตอนการออกแบบ 5

บทที่2. สัญญาณโทรทัศน์และการสะแกน 7

บทที่3. โครงสร้างของบอร์ด IMAGE CARD 16

บทที่4. หลักการและฮาร์ดแวร์ของบอร์ดดิจิไตเซอร์ 22

4.1 Buffer and Sync. Circuit 22

4.2 Analog to Digital Circuit 24

4.3 Counter Circuit 29

4.4 Sync. Generator Circuit 30

4.5 Control Counter Circuit 32

4.6 Clock Generator Circuit 33

4.7 RAM and Interface Circuit 34

4.8 Digital to Analog Circuit 35

4.9 ซอร์ฟแวร์ 37

บทที่5. การทดลองและผลการทดลอง 39

5.1 ขั้นตอนการทดลองและผลที่ได้ 39

5.2 บทสรุปและวิจารณ์ 41

5.3 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข 42

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ปัจจุบันในชีวิตประจำวัน การสื่อสารโดยใช้ภาพจะสามารถทำความเข้าใจได้ง่ายและรวดเร็วกว่าการอธิบายด้วยตัวอักษร ซึ่งการส่งโทรทัศน์แบบก่อน ๆ จะใช้การส่งภาพแบบ Real Time คือ ส่งภาพต่อเนื่องกันไปโดยที่ไม่มีการข้อมูลเอาไว้เมื่อต้องการที่จะนำภาพกลับมาใช้อีกครั้งจึงไม่สามารถที่จะนำกลับมาได้ เหตุนี้จึงได้มีการพัฒนาให้สามารถเก็บข้อมูลภาพเหล่านั้น ในยุคแรก ๆ การเก็บข้อมูลจะอยู่ในรูปของสัญญาณอะนาล็อค (Analog Signal) แต่ในปัจจุบัน การเก็บข้อมูลภาพจะใช้เก็บแบบ ดิจิตอล (Digital Signal) เพราะสามารถนำภาพเหล่านั้นไปประยุกต์ใช้กับอุปกรณ์ต่าง ๆ ได้ง่าย เช่น ต่อเข้ากับเครื่องคอมพิวเตอร์ โดยอาศัย Software เป็นตัวควบคุม

การประมวลผลภาพในปัจจุบันที่นิยมใช้กันคือ วิเคราะห์และประมวลผลภาพที่ถ่ายจากดาวเทียม (Remote sening via satllites) ,การวิเคราะห์ชนิดของสิ่งของ, การวิเคราะห์ลายมือ การรับรู้ในการมองเห็นของหุ่นยนต์ เป็นต้น

โครงงานนี้เป็นเพียงส่วนหนึ่งของ Image Processing จุดมุ่งหมายของโครงงานก็เพื่อศึกษาระบบ Image Processing เพื่อที่จะสามารถประยุกต์ใช้งานด้านอื่น ๆ โครงงานนี้จะทำการศึกษาและสร้างส่วนที่เรียกว่า Digitizer ซึ่งเป็นส่วนหน้าของระบบ Image Processing ต่อจากนั้นจะนำไปทำการ (Interface) เข้ากับคอมพิวเตอร์

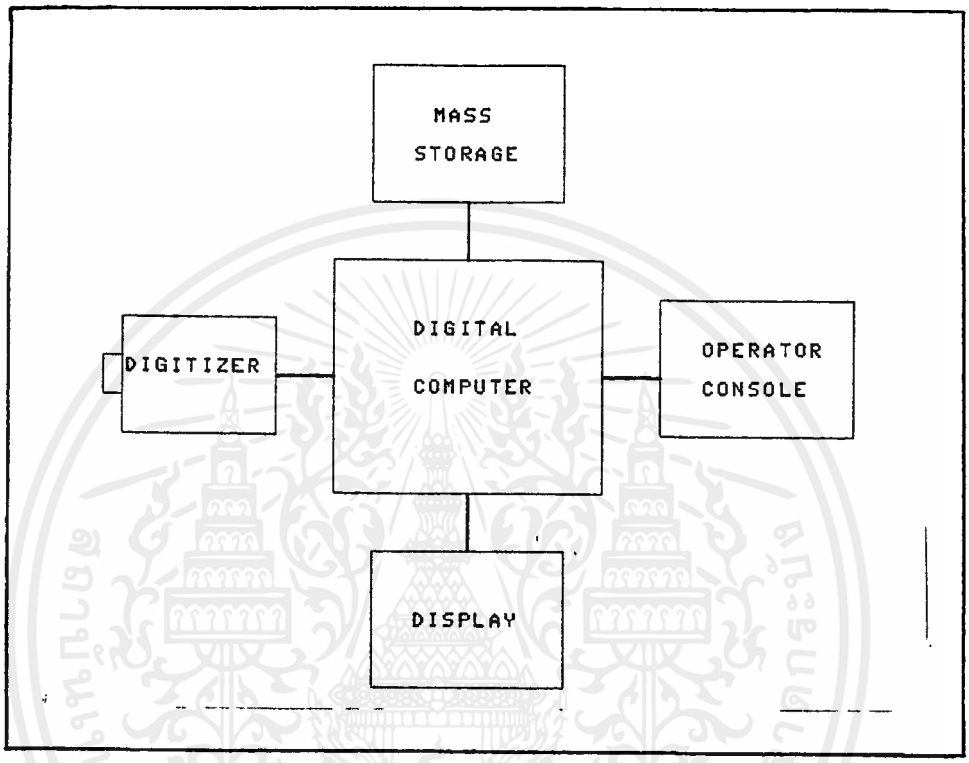
Digital Image Processing เป็นส่วนสำคัญของระบบ Image Proce - ssing ซึ่งจะประกอบด้วยส่วนของฮาร์ดแวร์หลายตัว ซึ่งมีหน้าที่หลัก คือ

1. การควบคุมการรับภาพ Digitizer โดยจะรับข้อมูลเข้ามาทางอินพุทซึ่งอยู่ในรูปของสัญญาณอะนาล็อคจากกล้องถ่ายโทรทัศน์ แล้วจึงทำการให้วงจร Analog To Digital ทำการเปลี่ยนเป็นสัญญาณดิจิตอล

2. การเก็บข้อมูลภาพ โดยต่อแรกจะเก็บข้อมูลภาพไว้ในหน่วยความจำในรูปของสัญญาณดิจิตอล ซึ่งในโครงงานนี้จะทำการสร้างให้หน่วยความจำอยู่บนการ์ด ส่วนการควบคุมการอ่านเขียนหน่วยความจำ จะต้องสัมพันธ์กันกับสัญญาณที่ใช้ควบคุมภาพของโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การแสดงผล (Display) มีหน้าที่ทำการอ่านข้อมูลจากหน่วยความจำแล้วเปลี่ยนจากสัญญาณ ดิจิตอล ให้เป็นสัญญาณอะนาล็อค ส่งไปแสดงที่จอมอนิเตอร์



รูปที่ 1.1 แสดงบล็อคไดอะแกรมของ Image Card

จากรูปที่ 1.1 เป็นบล็อคไดอะแกรมของ Image Cade จะประกอบด้วย

DIGITIZER มีหน้าที่เปลี่ยนสัญญาณภาพที่อยู่ในรูปของสัญญาณอะนาล็อคให้อยู่ในรูปของสัญญาณดิจิตอล เพื่อให้สามารถนำไปประมวลผลข้อมูลได้ อุปกรณ์ที่สำคัญที่เป็นอินพุทของ Digitizer คือ Vidicon Camera , Microdensitometor , Flying Spon Scanner , Photosensitive Solid-State Array เป็นต้น แต่ที่นิยมใช้กันคือ Vidicon camera

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIGITAL COMPUTOR การ์ด Digitizer จะถูก Interface เข้ากับคอมพิวเตอร์ จะเป็นแบบ XT , AT หรือ COMPATIBLE ก็ได้ ซึ่งสามารถนำไปประยุกต์โดยอาศัย software เป็นตัวควบคุม

STORAGE DEVICE การ์ด Image ที่สร้างขึ้นจะมีขนาด 256 X 256 จุดหรือพิกเซล ซึ่งแต่ละพิกเซลจะมีขนาด 8 บิท ทำให้สามารถแสดงความแตกต่างของระดับสีได้ 256 ระดับ เพราะฉะนั้น หน่วยความจำที่ใช้ในการเก็บข้อมูลจะมีขนาด 256 X 256 = 64K byte ถ้าทำการเก็บข้อมูลลงในแผ่น DISK ขนาดความจุ 1.2 MB จะสามารถเก็บข้อมูลภาพได้ถึง 187 ภาพ

DISPLAY AND RECORDING อุปกรณ์ที่ใช้ทำการแสดงผล จะใช้จอสี หรือจอโมโนโครม เป็นตัวแสดงผลของระบบ อุปกรณ์อีกอย่างที่ใช้แสดงผลคือเครื่องพิมพ์ จะใช้แบบ DOT MATRIC ซึ่งสามารถทำการพิมพ์แบบ Bit Image ได้จะช่วยทำให้สามารถแสดงรายละเอียดได้ดีกว่า ในการแสดงผลโดยใช้เครื่องพิมพ์นั้นจะต้องสร้างโปรแกรมที่ใช้ควบคุมการพิมพ์เพื่อให้ได้ภาพที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของการ์ด

- ขนาดของภาพที่เก็บได้	256*256	จุด/ภาพ
- จำนวนบิทต่อจุด	8	บิท/จุด
- ระดับความแตกต่างของสี	256	ระดับ
- ขนาดของ Video RAM	64	Kbyte
- Sampling rate	6	MHz
- Video Input	Composite Video	
- Video Output	Video, H_Sync. ,V_Sync.	
- System Requite	IBM PC/AT or compatible	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวความคิดและขั้นตอนการออกแบบ

แนวความคิด

จากผลการพัฒนาของทางด้าน Computer และการผสมผสานทางด้านเทคโนโลยีอิเล็กโทรนิกส์สื่อสาร ทำให้เกิดผลงานหรือ Project ขึ้นมามากมาย Image Processing ก็เป็น Project ชนิดหนึ่งที่มีการพัฒนาอย่างที่เรียกว่าไม่หยุดยั้งก็ว่าได้ จึงเป็นที่น่าสนใจควรต่อการศึกษาเพิ่มเติมอย่างยิ่งเพราะจะเป็นการช่วยส่งเสริมแนวความคิด ขีดความสามารถในการออกแบบผลงานของนักประดิษฐ์คิดค้นขึ้นมาได้ ซึ่งสามารถนำมาประยุกต์ใช้งานเพื่อตอบสนองความต้องการในงานธุรกิจโดยจะพบเห็นได้ตามแหล่งชุมชนย่านธุรกิจ ศูนย์การค้า บริษัท รวมไปถึงสถานที่หน่วยงานราชการบางแห่ง

ทางคณะผู้จัดทำจึงได้ทำ Project ชนิดนี้ขึ้นเพื่อให้ได้ตรงตามวัตถุประสงค์ดังกล่าว

การวิจัย

เมื่อเราได้กำหนดเป้าหมายและวัตถุประสงค์ของการสร้าง Project ขึ้นนี้ขึ้นมาแล้ว ในเบื้องต้นเราต้องกำหนดถึงรายละเอียดก่อนเพื่อที่จะหาขอบเขตและแนวทางในการวางแผนดำเนินงานการออกแบบให้เป็นขั้นตอน

การพัฒนา

ในขั้นตอนนี้จะเป็นการค้นคว้าหาข้อมูลมาสนับสนุนแนวความคิดที่ได้วางไว้แล้วจากแหล่งวิชาการต่างๆให้ได้ความเป็นจริงขึ้นมา ในบางครั้งแนวความคิดที่วางไว้นั้นอาจจะขัดแย้งกับความเป็นจริงทางทฤษฎี ดังนั้นเราต้องทำการประยุกต์การออกแบบใหม่หรือเปลี่ยนวิธีการออกแบบให้สอดคล้องกับทฤษฎีความเป็นจริง เพื่อให้บรรลุถึงจุดประสงค์ที่ต้องการนั้นๆ รวมทั้งการออกแบบวงจรและการทำการทดลองวงจรที่ละส่วนพร้อมทั้งปรับปรุงแก้ไขให้สอดคล้องกัน เมื่อจะมีการนำเอาแต่ละส่วนมาประกอบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบขั้นสุดท้าย

หลังจากมีการแก้ไขปรับปรุงและดัดแปลงให้มีความเข้ากันได้ดีของHardware แต่ละส่วนและทุกสิ่งทุกอย่างลงตัวดีแล้ว เราก็จะทำการสร้างและพัฒนาSoftware ขึ้นมา โดยจะศึกษาหาเทคนิคต่างๆของรูปแบบการทำงานของโปรแกรม เพื่อเป็นการศึกษาให้รู้ถึงเทคนิคในการแก้ไขและการใช้โปรแกรมให้ได้ประสิทธิภาพของงานมากที่สุด เราก็จะได้ Software ที่สมบูรณ์ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

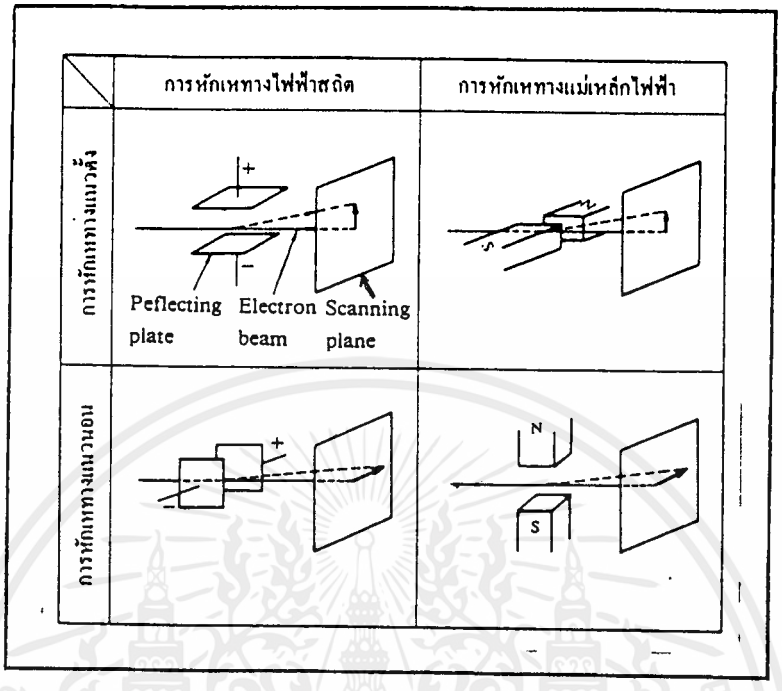
บทที่ 2

สัญญาณโทรทัศน์และการสะแกน

ภาพบนจอหลอดภาพของเครื่องรับโทรทัศน์สีทั่วไป จะประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนเป็นจำนวนมาก ซึ่งแต่ละเส้นเหล่านี้มีทั้งส่วนที่ดำสนิทหรือมีสีเข้ม ส่วนที่ดำจางหรือมีสีจางและส่วนที่สว่างมากปะปนกันอยู่ เส้นขวางเล็กๆ ในแนวนอนเหล่านี้ มีชื่อเรียกว่า เส้นสะแกน เส้นเหล่านี้ประกอบไปด้วยจุดเล็กๆ มีทั้งมืดและสว่างปะปนกันอยู่ ภาพที่ปรากฏบนจอหลอดภาพจึงประกอบด้วยจุดเล็กๆ ที่มีระดับแสงสว่างแตกต่างกันเป็นจำนวนมาก จุดเล็กๆเหล่านี้เรียกว่า ส่วนประกอบของภาพหรือ picture element ซึ่งมีส่วนสัมพันธ์กับความละเอียดของภาพ เช่นเดียวกับจุดดำหรือจุดสีเล็กๆในรูปภาพของสิ่งตีพิมพ์ ภาพที่เห็นบนจอหลอดภาพจะมองดูละเอียดน่าดู หากมีจำนวนจุดเล็กๆ หรือจำนวนเส้นสะแกนในแนวนอนมากเพียงพอ ด้วยเหตุนี้โทรทัศน์ระบบยุโรปซึ่งมีจำนวนเส้นสะแกน 625 เส้นต่อภาพ จึงให้ภาพที่มองดูละเอียดกว่าโทรทัศน์ระบบอเมริกัน ซึ่งมีจำนวนเส้นสะแกนเพียง 525 เส้นต่อภาพเท่านั้น อย่างไรก็ตามภาพที่เห็นบนจอหลอดภาพจะมองดูละเอียดหรือหยาบนั้นยังขึ้นอยู่กับส่วนประกอบอีกหลายอย่างเช่น ความสว่างของภาพและระยะทางที่มองดูภาพ เป็นต้น สำหรับโทรทัศน์ระบบอเมริกัน ซึ่งมีจำนวนเส้นสะแกนน้อยกว่าระบบยุโรป อันอาจทำให้มองเห็นภาพมีความละเอียดน้อยกว่าแต่ถ้าหากมองดูภาพในระยะทางห่างประมาณสี่ถึงแปดเท่าของความสูงของภาพแล้ว ก็จะรู้สึกว่าเป็นภาพที่พอใช้ได้เหมือนกัน

จุดที่เห็นสว่างในจอหลอดภาพของเครื่องรับโทรทัศน์ เกิดขึ้นเพราะอิเล็กตรอนที่หลุดออกไปจากแคโถด และถูกให้ดึงดูดให้วิ่งเป็นลำไปกระทบแอโนด หรือจอหลอดภาพ ซึ่งฉาบวัสดุเรืองแสงบางชนิดเอาไว้ จุดที่มีการกระทบกันก็จะมองเห็นเป็นจุดสว่างขึ้นที่จอ การสะแกนก็คือ การทำให้จุดสว่างเหล่านี้เคลื่อนที่ไปในจังหวะที่ต้องการ ซึ่งในเรื่องของโทรทัศน์ ก็ต้องการให้จุดสว่างนี้เคลื่อนที่ไปในแนวนอนและแนวตั้ง โดยอาศัยความเข้มของสนามแม่เหล็กเข้าช่วยเหลือ ทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอน ในหลักการการทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอนนี้ อาจทำได้โดยวิธีการหักเหของไฟฟ้าสถิต หรือวิธีการหักเหของแม่เหล็กไฟฟ้าตามที่แสดงในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

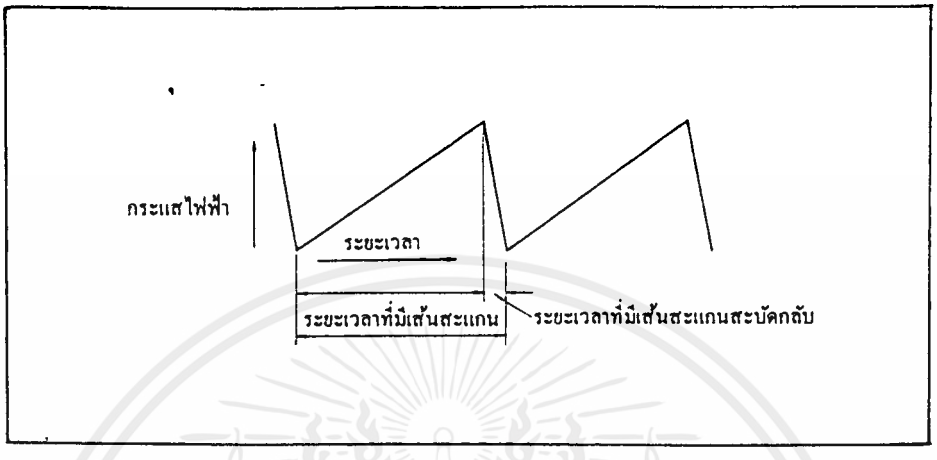


รูปที่ 2.1 ทฤษฎีของการหักเหทางไฟฟ้าสถิต และทางแม่เหล็กไฟฟ้า

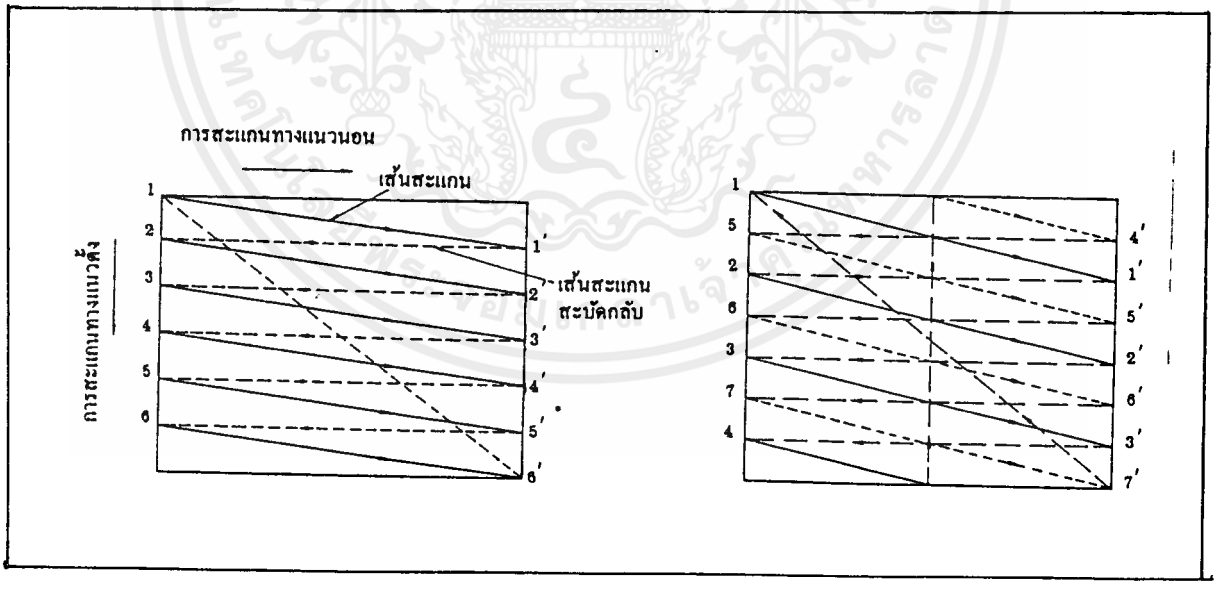
ซึ่งวิธีการหลังนี้ เป็นที่นิยมกันมากในทางปฏิบัติ สนามแม่เหล็กนี้เกิดขึ้นโดยการปล่อยกระแสไฟฟ้ารูปฟันเลื่อย ตามที่แสดงในรูปที่ 2.2 ให้ไหลผ่านขดลวดของการหักเห(deflection coil) ที่พันอยู่รอบๆคอหลอดภาพ ซึ่งมีอยู่สองชุดด้วยกันคือ ขดลวดที่พันอยู่รอบคอหลอดภาพในแนวนอนชุดหนึ่ง และขดลวดที่พันอยู่รอบคอหลอดภาพในแนวตั้งอีกชุดหนึ่ง สำหรับโทรทัศน์ระบบยุโรป ความถี่ของกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวนอนจะมีค่า 16,625 Hz ส่วนกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวตั้งจะมีค่าเพียง50 Hz เท่านั้น โดยปกติการสะแกนจะเริ่มต้นขึ้นโดยการทำให้จุดสว่างบนจอหลอดภาพเคลื่อนที่จากซ้ายมือด้านบนของจอไปทางขวามือในแนวนอน ซึ่งเมื่อถึงตำแหน่งขวามือสุด ก็จะถูกเบนต่ำลงเล็กน้อย อันเป็นผลมาจากการที่มีกระแสรูปฟันเลื่อยไหลผ่านขดลวดของการหักเหในแนวตั้ง แล้วก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่มาทางขวามือในแนวนอนเป็นอยู่เช่นนี้เรื่อยๆ จนกระทั่งจุดสว่างนั้นไปถึงตำแหน่งขวามือข้างล่างสุดของจอหลอดภาพ จึงเป็นอันเสร็จสิ้นการสะแกนภาพนิ่งภาพหนึ่ง หรือเรียกว่า เฟรมหนึ่ง ตามที่แสดงดังรูปที่ 2.3 หลังจากนั้น



รูปที่ 2.2 รูปร่างของกระแสรูปฟันเลื่อย



รูปที่ 2.3 การสะแกนจากซ้ายไปขวา และจากบนลงล่าง

รูปที่ 2.4 การสะแกนไขว้กัน (interlace scanning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
032667

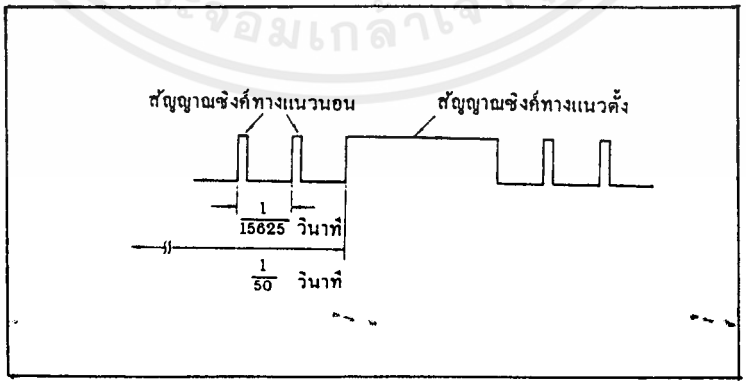
ลำอิเล็กตรอนก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือด้านบนสุดของจอหลอดภาพอีก เพื่อสะแกนภาพนิ่งอันดับถัดไป อย่างไรก็ตามเพื่อลดอาการกระพริบของภาพการสะแกนภาพนิ่งแต่ละภาพ มักนิยมจัดทำสองครั้งในแบบของการสะแกนไขว้กัน ซึ่งเรียกว่า interlace scanning ตามที่แสดงดังรูปที่ 2.4 โดยกำหนดให้ภาพนิ่งหนึ่งเฟรมประกอบด้วยภาพนิ่งสองฟิลด์ และเริ่มต้นด้วยการสะแกนภาพนิ่งฟิลด์เส้นคี่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งขวามือล่างสุดของจอหลอดภาพแล้ว จึงกลับไปตั้งต้นใหม่ทางด้านซ้ายมือสุดของจอ แล้วเริ่มต้นสะแกนภาพนิ่งฟิลด์เส้นคู่ต่อไป จนถึงตำแหน่งขวามือล่างสุด หลังจากนั้น จึงจะเริ่มต้นสะแกนภาพนิ่งอันดับอื่นต่อไปใหม่ ฉะนั้น ภาพนิ่งหนึ่งภาพหรือหนึ่งเฟรม จึงประกอบด้วยฟิลด์เส้นสะแกนเส้นคี่และฟิลด์เส้นสะแกนเส้นคู่ สำหรับโทรทัศน์ระบบยุโรป ซึ่งใช้เส้นสะแกน 625 เส้นต่อภาพและ 50 ภาพต่อวินาทีนั้น ภาพนิ่งแต่ละภาพหรือแต่ละเฟรมจะประกอบด้วยเส้นสะแกนแนวนอน 625 เส้น ภาพนิ่งแต่ละฟิลด์จะมีเส้นสะแกนแนวนอนครึ่งหนึ่งของ 625 เส้น หรือ 312 1/2 เส้น ภาพนิ่งแต่ละภาพนี้จะเกิดขึ้นภายในระยะเวลา 1/25 วินาที ความถี่ของกระแสรูปฟันเลื่อยที่ใช้ในการหักเหทางแนวนอน ซึ่งในระยะเวลา 1/25 วินาที จะเกิดเส้นสะแกน 625 เส้น จะมีค่า (625)(25) หรือ 15,625 Hz ส่วนความถี่ของกระแสรูปฟันเลื่อยที่ใช้ในการหักเหทางแนวตั้ง ซึ่งใช้เวลาในการสะแกนจากบนสุดมาล่างสำหรับฟิลด์หนึ่งๆ เพียง 1/50 วินาที จะมีค่า 50 Hz การสะแกนภาพนิ่งตามที่กล่าวมาแล้วนี้ จะกระทำติดต่อกันไปเรื่อยๆ โดยจะมีจำนวนภาพนิ่งหรือจำนวนเส้นสะแกนต่อภาพ กับจำนวนภาพต่อวินาทีแตกต่างกันไปตามแต่ชนิดของระบบโทรทัศน์ที่ใช้ภาพที่ปรากฏบนจอหลอดภาพของเครื่องรับโทรทัศน์ จึงมีผลคล้ายกับการฉายภาพนิ่งซึ่งแต่ละภาพมีความแตกต่างกันบ้างเพียงเล็กน้อยเป็นจำนวนหลาย ๆภาพต่อหนึ่งวินาที ด้วยเหตุที่สายตาของคนเรามีคุณลักษณะพิเศษในเรื่องของ persistence of vision จึงทำให้ผู้ชมโทรทัศน์สามารถมองเห็นภาพบนจอหลอดภาพของเครื่องรับโทรทัศน์เป็นภาพเคลื่อนไหวติดต่อกันไปตลอดเวลา

เรื่องที่สำคัญอีกอย่างหนึ่งของการส่งและรับโทรทัศน์ก็คือ จะต้องสามารถหาวิธีการซึ่งทำให้การสะแกนของภาพที่เกิดขึ้นในกล้องโทรทัศน์นั้น เกิดขึ้นพร้อมกันกับการสะแกนของภาพที่จอหลอดภาพของเครื่องรับโทรทัศน์ หรือทำให้ความถี่ของกระแสรูปฟันเลื่อยของวงจรหักเหทางแนวนอนและแนวตั้งทางกล้องโทรทัศน์เท่ากันตลอดเวลา กับความถี่ของวงจรหักเหทางแนวนอนและแนวตั้งทางจอหลอดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

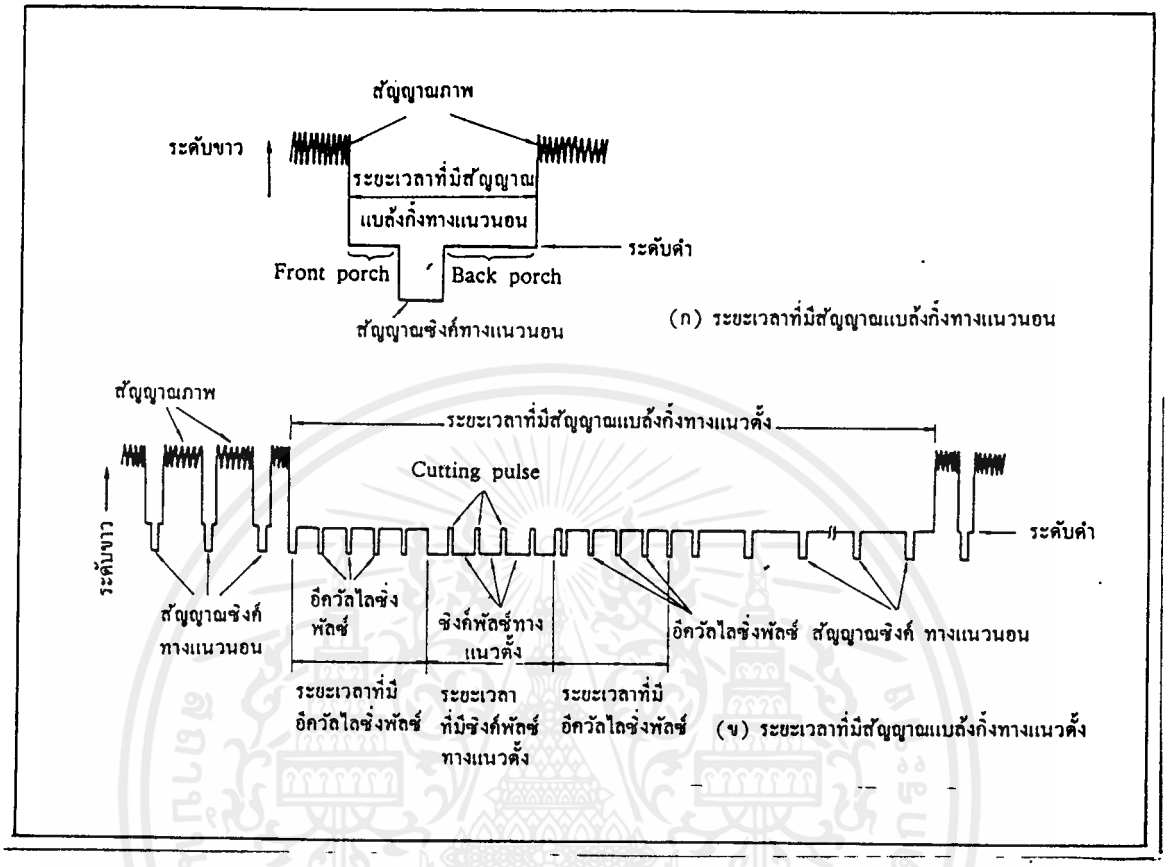
ของเครื่องรับโทรทัศน์ หากความถี่ของกระแสรูปฟันเลื่อยในวงจรทางเครื่องส่งโทรทัศน์ไม่เท่ากันตลอดเวลา กับความถี่ของกระแสรูปฟันเลื่อยในวงจรทางเครื่องรับโทรทัศน์ ก็จะพบว่า ภาพจะล้มหรือไม่มีภาพทางเครื่องรับโทรทัศน์ การทำให้ความถี่ของกระแสรูปฟันเลื่อยทางด้านเครื่องส่งโทรทัศน์เท่ากันตลอดเวลา กับความถี่ของกระแสรูปฟันเลื่อยทางด้านเครื่องรับโทรทัศน์นี้เรียกว่า เกิดการเข้าจังหวะ(synchronization)ขึ้น ในทางปฏิบัติสถานีโทรทัศน์จะต้องส่งสัญญาณชนิดหนึ่งเรียกว่า สัญญาณซิงค์(synchronizing signal หรือ sync pulse signal) ไปพร้อมกับสัญญาณภาพ ตามที่แสดงไว้ในรูปที่ 2.5 และ รูปที่ 2.6 สัญญาณซิงค์นี้จะประกอบด้วยสัญญาณซิงค์ทางแนวนอน (horizontal synchornizing signal)ซึ่งมีความถี่ 15,625 Hz หรือจะมี sync pulse ครั้งหนึ่งในทุก ๆ ครั้งที่มีเส้นสะแกนในแนวนอนกับสัญญาณซิงค์ในแนวตั้ง (vertical synchornizing signal) ซึ่งมีความถี่ 50 Hz หรือจะมี sync pulse ครั้งหนึ่งในขณะที่มีการสะแกนฟิลด์เส้นคี่หรือฟิลด์เส้นคู่เสร็จสิ้นลงสัญญาณซิงค์เหล่านี้จะส่งไปพร้อม ๆ กับสัญญาณภาพ ในช่วงระยะเวลาของเส้นสะแกนสะบัดกลับ หรือช่วงระยะเวลาที่เส้นสะแกนกำลังหันกลับไปเริ่มต้นใหม่(flyback period)

ในทางปฏิบัติ สถานีโทรทัศน์ขาวดำจะต้องส่งสัญญาณต่าง ๆ หลายอย่างออกทางอากาศไปให้เครื่องรับโทรทัศน์ เพื่อทำให้เกิดภาพขาวดำที่จอหลอดภาพของเครื่องรับโทรทัศน์ในลักษณะเดียวกัน และพร้อมกันกับการสะแกนภาพของกล้องโทรทัศน์ สัญญาณต่าง ๆ สำหรับทำให้เกิดภาพขาวดำเหล่านี้ แสดงในรูปที่ 2.6



รูปที่ 2.5 รูปร่างของสัญญาณซิงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 รูปร่างของสัญญาณซิงค์ที่ใช้ในการส่งโทรทัศน์

ซึ่งประกอบด้วย

- สัญญาณภาพ(video signal)
- สัญญาณแบล็งกิ้ง(blanking signal)
- สัญญาณซิงค์(synchornizing signal)
- สัญญาณอีควอไลซิงค์(equalizing signal)

สัญญาณต่าง ๆ ตามรูปนี้ จะรวมอยู่เป็นรูปร่างเดียวกัน ซึ่งเรียกว่า สัญญาณภาพรวม(composite video signal)แล้วใช้คลื่นพาห์ของภาพเป็นตัวพาออกอากาศ รวมกับคลื่นพาห์ของสัญญาณเสียง เหตุผลและความจำเป็นในการใช้สัญญาณต่าง ๆ มีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) สัญญาณภาพ(video signal) และสัญญาณเสียง(sound signal) เป็นสัญญาณที่ใช้เพื่อทำให้เกิดภาพขาวดำที่จอหลอดภาพ และมีเสียงที่ลำโพงเครื่องรับโทรทัศน์ตามต้องการ สัญญาณภาพ (video signal) นี้บางครั้งเรียกว่าสัญญาณส่องสว่าง(brightness signal หรือ luminance signal)

(ข) สัญญาณแบล้งกิ้ง(blanking signal) เป็นสัญญาณเพื่อใช้เพื่อลบเส้นสะแกนสะบัดกลับทั้งในแนวนอนและแนวตั้ง เพื่อมิให้เป็นที่สังเกตเห็นได้ชัดทางจอหลอดภาพ รูปที่ 2.6 (ก) เป็นรูปขยายของระยะเวลาที่มีสัญญาณแบล้งกิ้งทางแนวนอน (horizontal blanking period)และในช่วงระยะเวลาที่มีสัญญาณแบล้งกิ้งทางแนวนอนนี้ ก็จะส่งสัญญาณซิงค์ทางแนวนอน(horizontal synchornizing signal)ไปด้วย แต่จะอยู่ในระดับดำสนิทมากกว่าสัญญาณแบล้งกิ้ง ส่วนที่เหลื่อมล้ำกันระหว่างแบล้งกิ้งพัลซ์กับซิงค์พัลซ์นี้ จะมีอยู่สองส่วนตามรูปที่แสดงไว้ ส่วนหน้าเรียกว่า front porch และส่วนหลังเรียกว่า back porch สำหรับโทรทัศน์ระบบอเมริกัน ความถี่ของกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวนอนมีค่า 15,750 Hz ดังนั้นในระยะเวลา 1/15,750 วินาที หรือ 63.5 ไมโครวินาที จะต้องเกิดเส้นสะแกนสะบัดกลับครั้งหนึ่ง จึงจำเป็นต้องใช้แบล้งกิ้งพัลซ์ทางแนวนอนครั้งหนึ่งโดยมีขนาดประมาณ 10 ไมโครวินาที ส่วนรูปที่ 2.6(ข) นั้น เป็นรูปขยายของระยะเวลาที่มีสัญญาณแบล้งกิ้งทางแนวตั้ง (vertical blanking signal) สำหรับโทรทัศน์ระบบอเมริกัน ทุก ๆ ระยะเวลา 1/60 วินาที หรือ 16.667 ไมโครวินาที จำเป็นต้องให้มีแบล้งกิ้งพัลซ์ทางแนวตั้งครั้งหนึ่งโดยมีขนาดประมาณ 1,250 ไมโครวินาที ในระยะที่มีแบล้งกิ้งพัลซ์ทางแนวตั้งนี้ ก็จะส่งสัญญาณซิงค์ทางแนวตั้งออกไปด้วย และเพื่อประโยชน์ในการช่วยทำให้สัญญาณซิงค์ทางแนวตั้ง ยังคงมีรูปร่างดีเหมือนเดิม หลังจากแยกออกมาจากสัญญาณซิงค์ทางแนวนอนทางเครื่องรับโทรทัศน์แล้ว จะนิยมส่งอีควอไลซิ่งพัลซ์ (equalizing pulses) กับ คัตติ้งพัลซ์(cutting pulses) ไปด้วย ตามรูปที่ 2.6 (ข) ความถี่ของอีควอไลซิ่งพัลซ์และคัตติ้งพัลซ์นี้จะมีค่าเป็นสองเท่าของความถี่สัญญาณซิงค์ทางแนวนอน เพื่อช่วยให้การสะแกนแบบหนึ่งเฟรมแบ่งออกเป็นสองฟิลด์ทางด้านเครื่องรับโทรทัศน์ เป็นไปอย่างถูกต้องเหมาะสม จุดตั้งต้นของสัญญาณซิงค์ทางแนวนอนและสัญญาณซิงค์ทางแนวตั้งนี้ จะต้องมีส่วนวัมพันธ์กันอย่างเหมาะสม คือ เมื่อหมดการสะแกนฟิลด์หนึ่ง ๆ แล้ว จะต้องเกิดขึ้นพร้อมกันเพื่อทำการสะแกนฟิลด์ต่อ ๆ ไป ตามที่ได้แสดงไว้แล้วดังรูปที่ 2.7 มิฉะนั้น การสะแกน

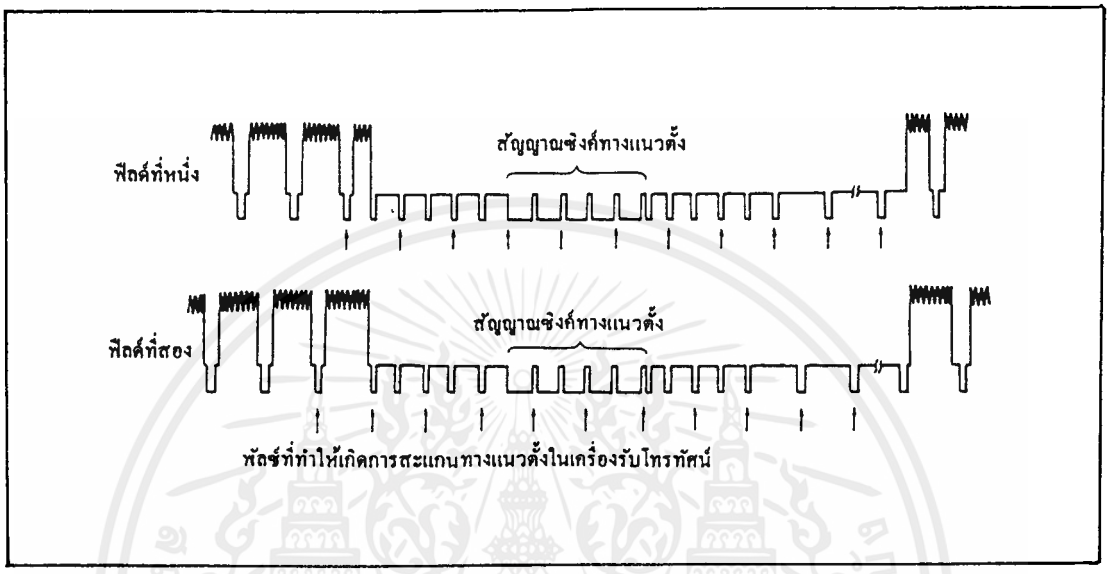
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไขว้กันทางเครื่องรับโทรทัศน์อาจไม่เป็นไปในจังหวะที่ถูกต้องได้

(ค) สัญญาณซิงค์(synchronizing signal) เป็นสัญญาณที่ใช้เพื่อช่วยทำให้ความถี่ของกระแสรูปฟันเลื่อยที่ใช้ในวงจรของการหักเหทางแนวนอน กับวงจรของการหักเหทางแนวตั้งของเครื่องส่งโทรทัศน์มีค่าตรงกันกับที่ใช้ในเครื่องรับโทรทัศน์ อันจะมีผลทำให้การสะแกนของภาพทางด้านเครื่องส่งโทรทัศน์ ตรงกันกับทางด้านเครื่องรับโทรทัศน์ตลอดเวลา สัญญาณซิงค์ทางแนวนอนจะมีความถี่เท่ากันกับความถี่ของกระแสรูปฟันเลื่อยที่ใช้ในวงจรของการหักเหทางแนวนอน และสัญญาณซิงค์ทางแนวตั้งก็จะมีความถี่ของกระแสรูปฟันเลื่อย ที่ใช้ในวงจรของการหักเหทางแนวตั้ง เนื่องจากความถี่ของสัญญาณซิงค์นี้เท่ากันกับความถี่ของสัญญาณแบล้งกิ้ง จึงจำเป็นต้องป้องกันการรบกวนที่อาจเกิดขึ้นโดยจำเป็นต้องกำหนดขนาดของซิงค์พัลซ์ให้น้อยกว่าขนาดของแบล้งกิ้งพัลซ์ กล่าวคือ ทำให้ซิงค์พัลซ์ทางแนวนอนมีขนาดเพียง 5 ไมโครวินาที และ ซิงค์พัลซ์ทางแนวตั้งมีขนาดเพียง 190 ไมโครวินาทีเท่านั้น นอกจากนี้ ยังใช้วิธีส่งซิงค์พัลซ์เหล่านี้ปะปนกับแบล้งกิ้งพัลซ์ โดยทำให้ฐานของซิงค์พัลซ์อยู่ทับขอบบนของแบล้งกิ้งพัลซ์อีกชั้นหนึ่ง เมื่อได้กำหนดให้ระดับสูงสุดของแบล้งกิ้งพัลซ์เป็นระดับดำมืดจนมองไม่เห็นทางจอหลอดภาพแล้ว ระดับของซิงค์พัลซ์ที่อยู่บนยอดสูงสุดของแบล้งกิ้งพัลซ์ ก็จะเป็นระดับดำมืดสนิท และไม่ทำให้เกิดการรบกวนภาพที่จอหลอดภาพแต่ประการใด

(ง) สัญญาณอีควอไลซิ่ง(equalizing signal) เป็นสัญญาณที่ใช้เพื่อช่วยทำให้สัญญาณซิงค์ทางแนวตั้งยังคงมีรูปร่างดีเหมือนเดิม หลังจากแยกออกมาจากสัญญาณซิงค์ทางแนวนอนในเครื่องรับโทรทัศน์แล้ว สัญญาณนี้มีความถี่เป็นสองเท่าของสัญญาณซิงค์ทางแนวนอน ซึ่งจะช่วยทำให้การสะแกนไขว้กันทางเครื่องรับโทรทัศน์เป็นไปโดยเรียบร้อย รวมทั้งสัญญาณซิงค์ทางแนวนอนก็จะไม่ขาดหายไปในช่วงเวลาที่มีสัญญาณซิงค์ทางแนวตั้งอีกด้วย ขนาดของอีควอไลซิ่งพัลซ์ก็มีขนาดประมาณซิงค์พัลซ์ทางแนวตั้ง คือประมาณ 190 ไมโครวินาที หรือประมาณสามเท่าของซิงค์ทางแนวนอน นอกจากนี้ยังนิยมแบ่งพัลซ์นี้ออกเป็นพัลซ์เล็ก ๆ ตามรูปที่ 2.6 เพื่อทำให้เกิดซิงค์พัลซ์ทางแนวนอนครั้งหนึ่ง ในทุก ๆ สองครั้งที่มีพัลซ์เล็ก ๆ เหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 พัลซ์ที่เกิดขึ้นในระยะเวลาที่มีสัญญาณแบล้งกิ้งทางแนวตั้ง ในฟิลด์ที่หนึ่งและฟิลด์ที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โครงสร้างของบอร์ด IMAGE CARD

หลักการทำงานของวงจร

หลักการทำงานของการ์ดประกอบด้วยส่วนสำคัญดังนี้ คือ

1. หน่วยความจำสำหรับเก็บภาพ ซึ่งมีขนาด 64 KB
2. ส่วนแปลงสัญญาณจากกล้องวีดีโอให้เป็นข้อมูลดิจิตอล
3. ส่วนแปลงข้อมูลดิจิตอลเป็นสัญญาณวีดีโอเพื่อออกจอภาพ
4. ส่วนควบคุมการติดต่อระหว่างส่วนต่าง ๆ

ขณะทำการแสดงข้อมูลจากหน่วยความจำภาพ หน่วยความจำจะถูกอ่านออกมาแสดง โดยผ่านวงจรแปลงข้อมูลเป็นสัญญาณอะนาล็อกแล้วจึงนำไปรวมกับสัญญาณซิงค์ ส่วนการถ่ายภาพ (Digitize) สัญญาณวีดีโอจากกล้องจะป้อนเข้าวงจรแปลงสัญญาณอะนาล็อกเป็นข้อมูลดิจิตอล ข้อมูลเหล่านี้จะเขียนลงสู่หน่วยความจำขณะเดียวกัน ก็จะนำออกแสดงผลออกทางจอด้วย

ไมโครคอมพิวเตอร์สามารถติดต่อกับหน่วยความจำได้ โดยผ่านทางพอร์ทควบคุมหมายเลข 300H หน่วยความจำนี้จะถูกแยกออกจากคอมพิวเตอร์ เมื่อต้องการใช้หรือติดต่อกับหน่วยความจำก็จะต้องสั่งให้หน่วยความจำหยุดการแสดงภาพ และให้มาต่อเข้ากับระบบของคอมพิวเตอร์

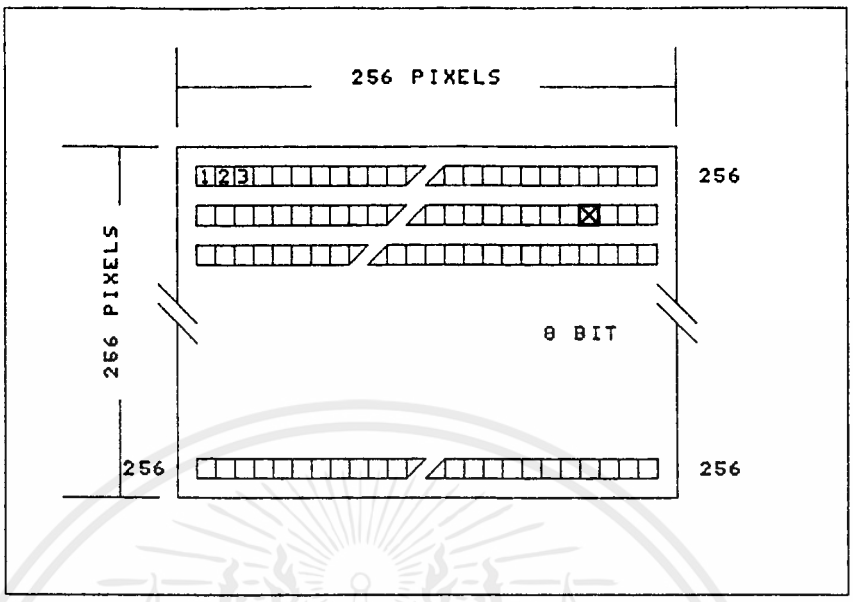
คำสั่งที่ส่งผ่านพอร์ทควบคุมมีดังนี้

00 = แสดงภาพเคลื่อนไหวพ
02 = แสดงภาพนิ่ง
03 = เคลื่อนย้ายข้อมูล

ลักษณะของข้อมูลภาพ

ภาพขาวดำที่ใช้แสดงบนจอขนาด 256 X 256 จุดนั้นแต่ละจุดจะต้องใช้เนื้อที่ขนาด 1 ไบต์ หรือขนาด 8 บิทเรียงกันไปในหน่วยความจำขนาด 64 KB ดังแสดงดังรูปที่ 3.1 โดยข้อมูลจุดแรกที่อยู่มุมบนซ้ายของจอภาพจะตรงกับหน่วยความจำแอดเดรสแรก คือ 0 ข้อมูลภาพจุดต่อมา (ตามแนวนอน) ก็จะอยู่ที่แอดเดรส 1 จนจบข้อมูลภาพเส้นแรก (จุดมุมบนขวา) ก็จะตรงกับแอดเดรส 255 หรือ 00FFH ที่เส้นถัดมาจุดแรกก็จะมีแอดเดรสเป็น 256 หรือ 0100H เรียงกันเช่นนี้ไปจนจบที่จุดสุดท้ายของจอภาพคือ จุดมุมล่างด้านขวาจะมีแอดเดรสเป็น 65535 หรือ 0FFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 การจัดหน่วยความจำวีดีโอแรมและระบบการเก็บภาพ

การติดต่อกับหน่วยความจำ

ดังที่ได้กล่าวมาแล้วว่า การอ่านหรือเขียนข้อมูลลงในหน่วยความจำของการ์ดสามารถกระทำได้โดยผ่านทางพอร์ทหมายเลข 300H จะสังเกตเห็นว่าหลังจากให้หน่วยความจำภาพมาต่อเข้ากับระบบของไมโครคอมพิวเตอร์แล้ว ภาพบนจอจะกระพริบ จึงทำการดับจอภาพเสีย แต่ถ้าต้องการไม่ให้ภาพบนจอกระพริบ เราจะต้องทำการอ่านเขียนข้อมูลภาพในช่วงที่มีสัญญาณ เวอร์แบล้ง (Vert. Blanking) ของจอภาพเท่านั้น

การเคลื่อนย้ายข้อมูลจะทำตั้งแต่เริ่มต้นช่วงเวอร์แบล้ง จะสามารถทำการเคลื่อนข้อมูลได้ประมาณ 1 กิโลไบต์ เพราะฉะนั้นใน 1 วินาที จะมีเวอร์แบล้งอยู่ 50 ครั้ง ฉะนั้นข้อมูลภาพขนาด 64KB จะใช้เวลาเคลื่อนย้ายข้อมูลประมาณ 1 วินาทีเศษ การเคลื่อนย้ายข้อมูลแบบนี้จะช้า แต่มีข้อดีตรงเวลาอ่าน หรือ เขียนข้อมูลจะไม่ต้องดับจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

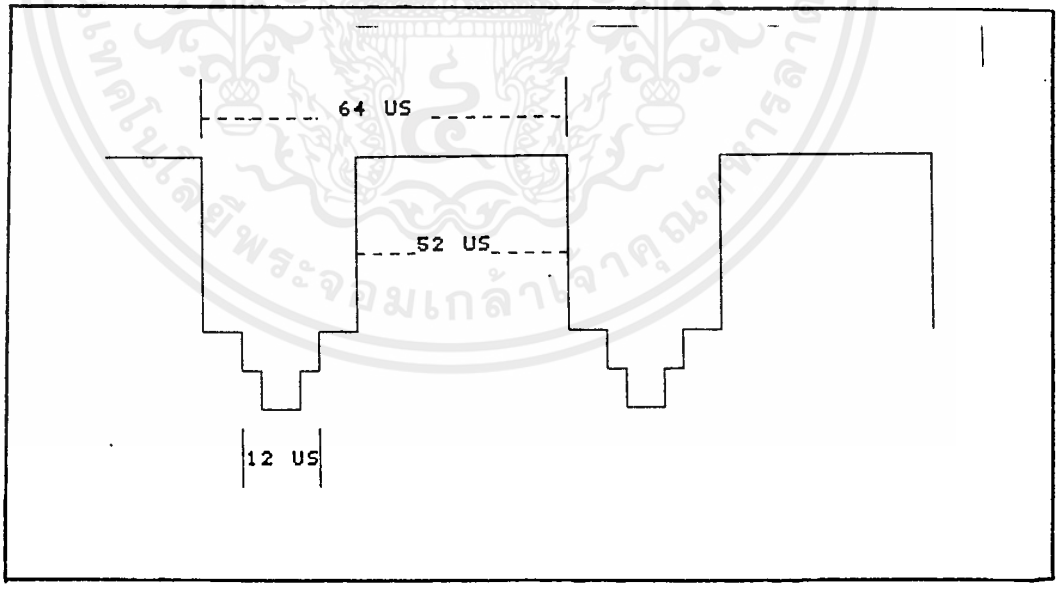
บล็อกไดอะแกรมของการ์ดดิจิไตซ์

จากบล็อกไดอะแกรมดังแสดงในรูปที่ 3.2 เป็นส่วนประกอบของการ์ด ซึ่งสามารถอธิบายการทำงานได้ 2 วิธีดังนี้

- การแสดงภาพเคลื่อนไหว
- การแสดงภาพนิ่ง

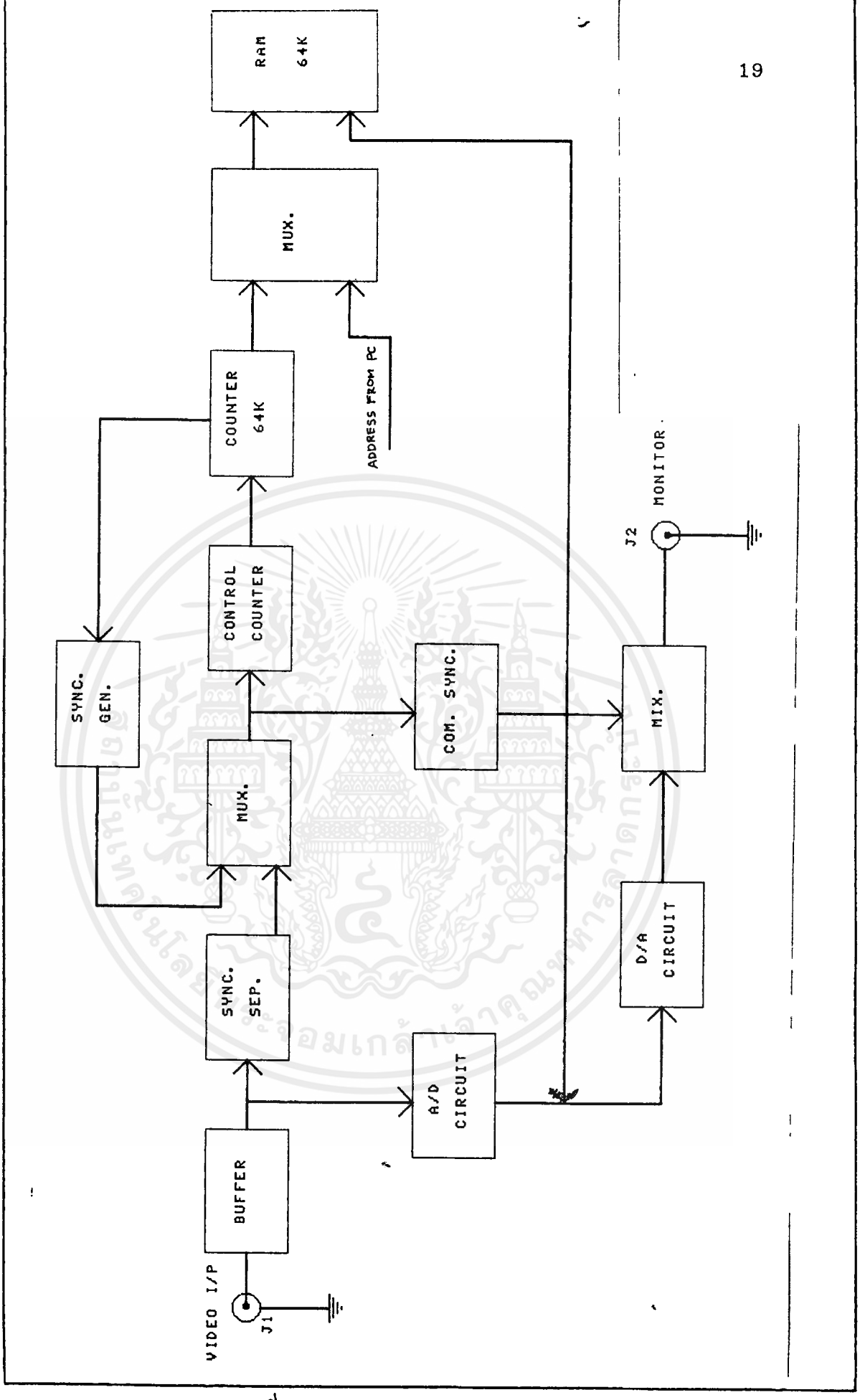
การแสดงภาพเคลื่อนไหว

เริ่มแรกสัญญาณอินพุท ถูกป้อนเข้าวงจร Buffer เพื่อป้องกันการสูญเสียและเป็นตัว Matching ระหว่างภาคสัญญาณวีดีโอจะถูกแยกเป็นสองส่วน ส่วนแรกจะถูกส่งผ่านไปเข้าวงจรแยก Sync. เพื่อทำการแยก Sync. ทางแนวนอนและแนวตั้ง เพื่อนำไปสร้างสัญญาณควบคุมการนับของวงจร Counter โดยสัญญาณวีดีโอหนึ่งเส้นสแกนจะมีระยะเวลา 64 μs จะมีสัญญาณภาพจริง ๆ เพียง 52 μs อีก 12 μs เป็นสัญญาณ Hor. Sync. และ Hor. Blank ดังแสดงในรูป 3.3



รูปที่ 3.3 แสดงคาบเวลาของสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดง Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ Sync. ที่ได้ออกมาจะนำมารวมกันกับสัญญาณภาพ เพื่อนำไปแสดงผลที่จอ Monitor สัญญาณที่แยกออกมาอีกส่วนหนึ่งจะป้อนผ่านวงจร Analog to Digital ทำการเปลี่ยนรูปสัญญาณให้เป็นสัญญาณ Digital ขนาด 8 บิท ซึ่งสามารถแสดงความแตกต่างสีได้ 256 ระดับ ADC ที่ใช้จะเป็นแบบ Flash ซึ่งสามารถใช้กับความถี่สูง ๆ เช่นสัญญาณวีดีโอได้ สัญญาณ Digital ที่ได้จะแยกเป็นสองส่วน ส่วนหนึ่งจะเก็บไว้ใน RAM อีกส่วนหนึ่งจะผ่านวงจร Digital to Analog Circuit เพื่อเปลี่ยนสัญญาณ Digital ให้เป็นสัญญาณ Analog แล้วนำไปรวมกับสัญญาณ Sync. ออกสู่จอ Monitor วิธีนี้ข้อมูลสัญญาณภาพจะถูกแปลงเป็นสัญญาณ Digital นำไปเก็บไว้ใน RAM ตลอดเวลา

การแสดงภาพนิ่ง

จากขั้นตอนการแสดงภาพเคลื่อนไหว เมื่อจะแสดงภาพนิ่ง จะส่งสัญญาณจากเครื่องคอมพิวเตอร์ คือสัญญาณการเขียนโดยผ่านทาง Port หมายเลข 300 มาทำการสร้างสัญญาณควบคุม และ Address จาก PC จะต่อเข้ากับการ์ด สัญญาณควบคุมที่สร้างขึ้นจะไปทำการหยุดการทำงานของ ADC เมื่อหยุดการทำงานแล้วข้อมูลภาพที่แปลงจะถูกเก็บไว้ใน RAM

เมื่อต้องการแสดงภาพนิ่งสัญญาณ Sync. จะมาควบคุมวงจร Counter ให้นับเพื่อนำข้อมูลจาก RAM ออกมาผ่านวงจร DAC เพื่อทำการแปลงสัญญาณดิจิตอล ให้เป็นสัญญาณอนาล็อก แล้วนำไปร่วมกับสัญญาณ Sync.ที่สร้างขึ้นออกแสดงผลที่จอ Monitor

สัญญาณที่ใช้ในการควบคุม

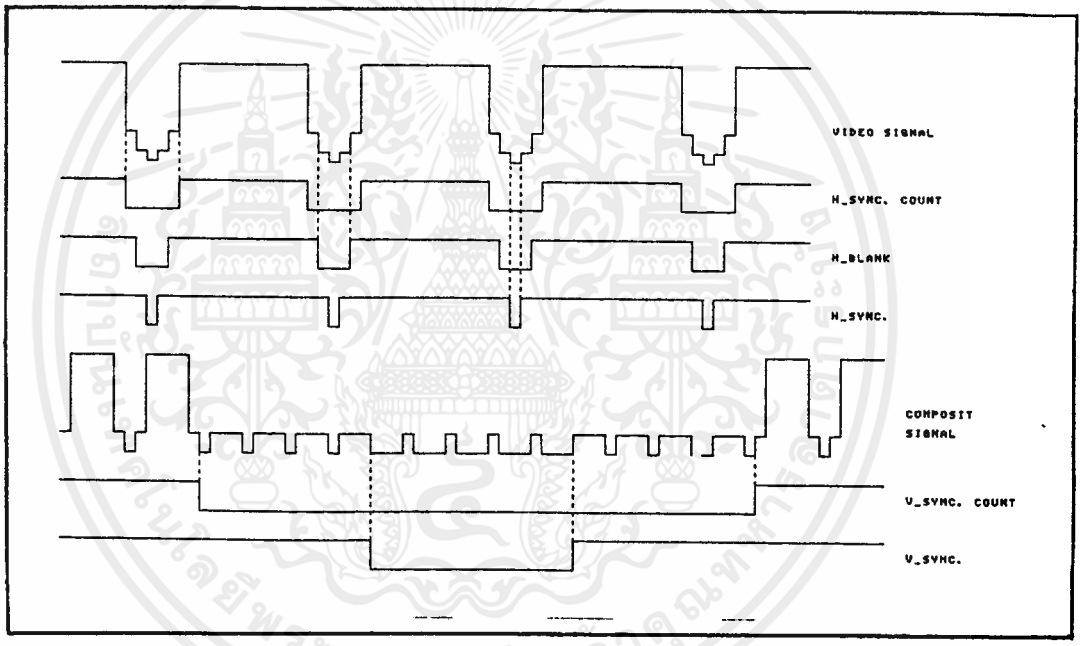
สัญญาณควบคุมต่าง ๆ แสดงดังรูป 3.4 โดยการนำสัญญาณ Sync. ที่แยกออกมาได้มาผ่านวงจร Monostable เพื่อสร้างสัญญาณควบคุมดังนี้

- **สัญญาณ H-Sync. Count** จะเป็นสัญญาณควบคุมที่เป็นตัวบอก ให้วงจร Counter เริ่มนับเพื่อส่งให้ RAM ทำการอ่านหรือเขียนข้อมูล
- **สัญญาณ H-Blank** สัญญาณนี้เป็นตัวบอกว่า ได้สิ้นสุดการสแกนในหนึ่งเส้นสแกน สัญญาณนี้เมื่อใช้รวมกับสัญญาณ H-Sync. Count จะใช้ในการควบคุมวงจร Counter เพื่อให้วงจร Counter ทำงานเก็บข้อมูลที่ตำแหน่งต่างๆ ในแต่ละไลน์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณ V-Sync. Count เป็นสัญญาณที่บอกให้ H-Sync. Count ทำการเริ่มต้นนับใหม่เมื่อจบข้อมูลในหนึ่งเฟรม

- สัญญาณ V-Blank เป็นสัญญาณที่ใช้ในการบอกให้ทราบว่าภาพที่แสดงให้เห็นบนหน้าจอได้แสดงครบหนึ่งฟิลด์แล้ว ซึ่งเราจะใช้สัญญาณนี้ควบคุมในการเก็บข้อมูลให้ตรงกับตำแหน่งไลน์แรกของภาพพอดี



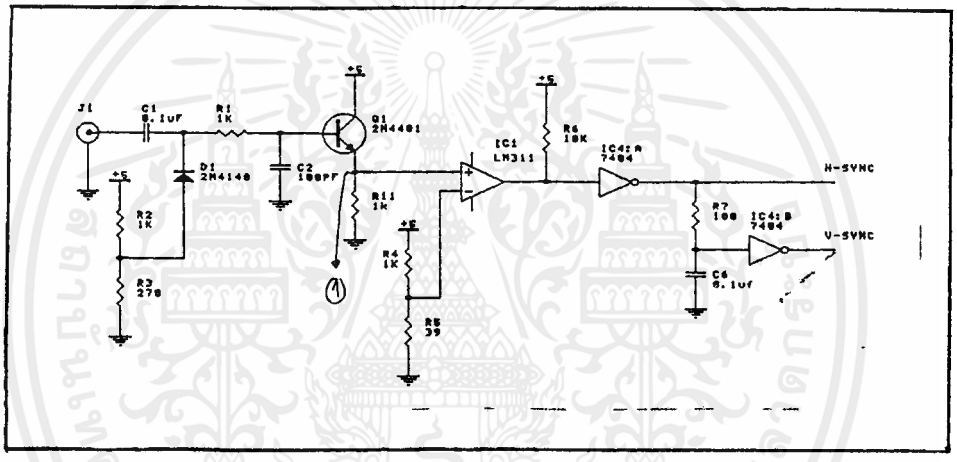
รูปที่ 3.4 แสดงสัญญาณที่ได้จากวงจรสัญญาณควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
หลักการและฮาร์ดแวร์ของบอร์ดดิจิไตซ์

1. Buffer and Sync. Sep.

จากวงจรดังรูปที่ 4.1 เป็นวงจร Clamp ที่ใช้ยกระดับสัญญาณวีดีโอที่ป้อนเข้ามา และเป็นวงจร Buffer ด้วย โดยมี R2,R3 เป็นวงจร Devider เพื่อสร้างระดับอ้างอิง



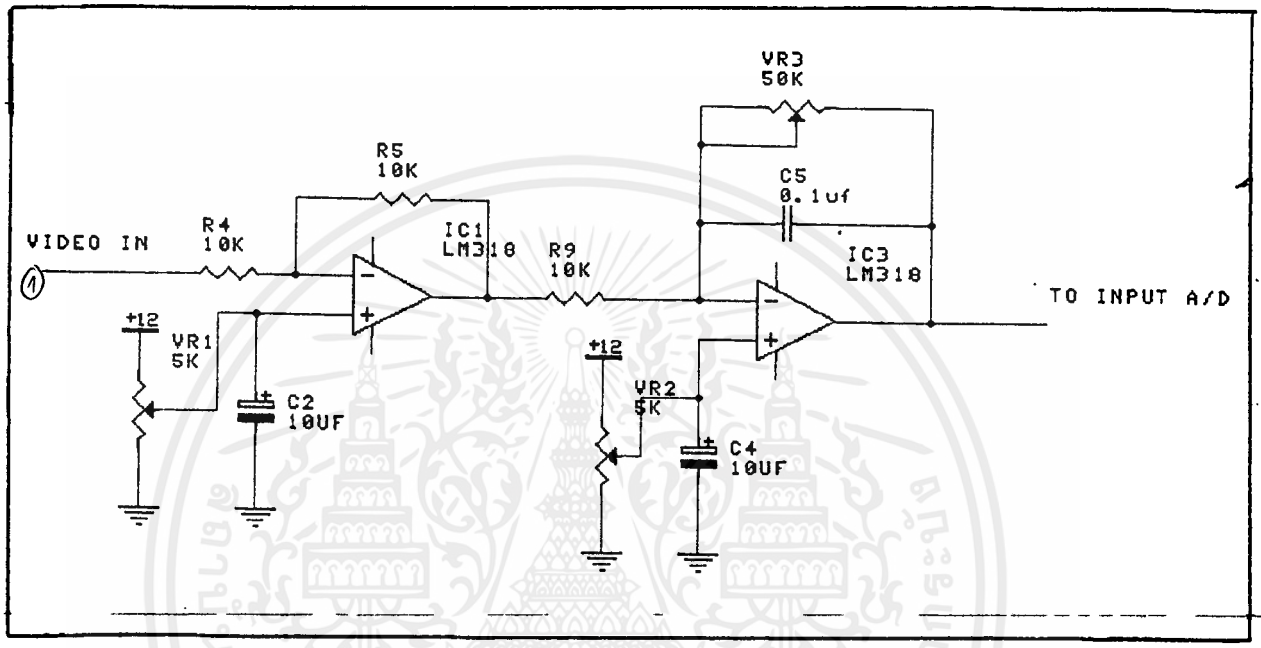
รูปที่ 4.1 แสดงวงจร Buffer และ Sync. Sep.

สัญญาณที่ผ่านการ Clamp จะแยกออกเป็น 2 ทางคือ

1. สัญญาณที่ถูก Clamp จะผ่าน IC2 ซึ่งต่อเป็นวงจร Comparater เพื่อทำการแยกเอาสัญญาณ Sync. ออกมาสัญญาณ Sync. ที่ได้จะมีทั้ง Hor. Sync. และ Vert.Sync. ในการที่จะแยกเอา Sync. ทางด้าน Vert. ออกมานั้นจะนำไปผ่านวงจร Low pass Filter เพื่อกรองเอา Hor.Sync. ซึ่งเป็นความถี่สูงทิ้งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สัญญาณที่ถูก Clamp จะถูกส่งไปที่ IC1 ซึ่งมี Gain เป็น Unity และที่ขา Non-inverting จะมี VR1เป็นตัวปรับระดับของสัญญาณ Sync. ส่วน IC3 จะเป็นการปรับอัตราการขยายสัญญาณ หรือเป็นการปรับความสว่างของสัญญาณภาพ



รูปที่ 4.2 แสดงวงจรปรับ Gain และความสว่างของสัญญาณภาพ

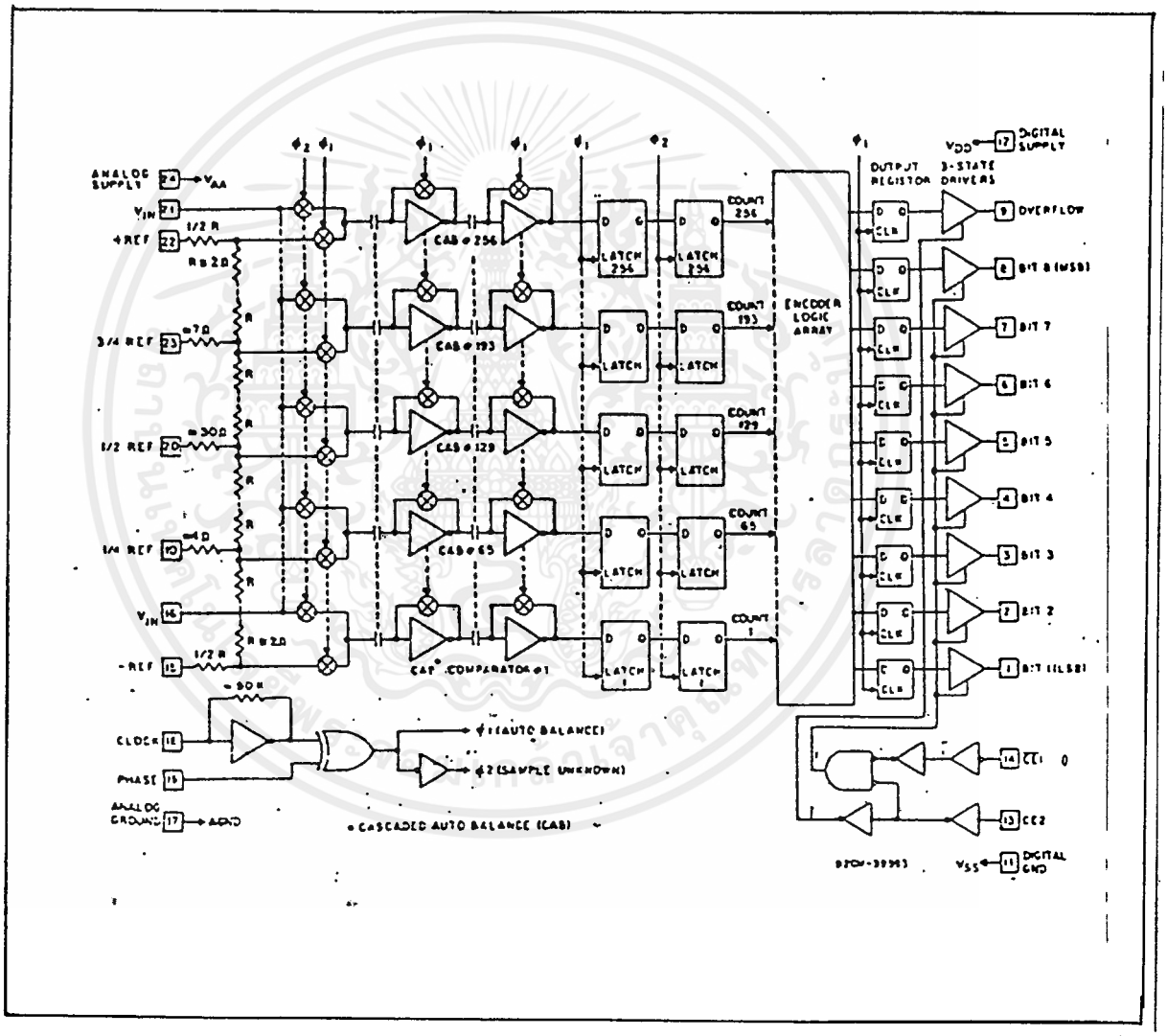
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Analog to Digital Circuit

การทำงานของส่วนวงจร Analog to Digital นี้ จะใช้ตัวแปลง เบอร์ CA 3318 ซึ่งเป็น ADC ขนาด 8 บิท แบบ Flash มี Sampling rate สูงสุด 20 MHz แต่ในวงจรจริง จะใช้เพียง 6 MHz เท่านั้น

โครงสร้างของ IC เบอร์ CA 3318

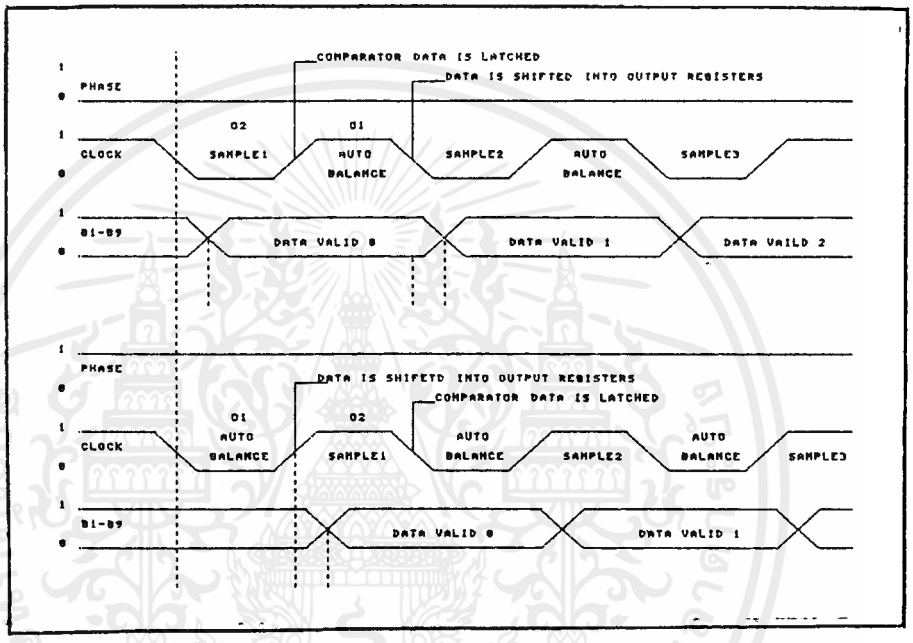
โครงสร้างภายในตัว CA 3318 ดังแสดงในรูป 4.3



รูปที่ 4.3 แสดงโครงสร้างภายใน CA 3318

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดสวิทช์ Electronic จะมีอยู่ 256 ชุด โดยจะทำสัญญาณ Input มาเปรียบเทียบกับแรงดันอ้างอิงของตัว Comparater ภายใน ข้อมูลทั้งหมดที่ได้จากตัวเปรียบเทียบจะส่งเข้า D F/F ทั้ง 256 ชุด โดยตรง โดยมี Clock เป็นตัว Shift ว่าจะเข้า D F/F ตัวไหน D F/F จะทำการ Latch ข้อมูลไว้ชั่วขณะจนกว่าจะมีข้อมูลใหม่เข้ามา จากนั้นข้อมูลทั้งหมดจะถูกส่งเข้าวงจร Encoder Logic Array เพื่อแปลงเป็นข้อมูลขนาด 8 บิท รวมบิทเกินอีก 1 บิท ที่ Output จะมี Buffer แบบ 3 สถานะ ซึ่งถูกควบคุมด้วยขา /CE1,CE2 การควบคุมอาจทำได้ที่ขา Phase อีกด้วย



รูปที่ 4.4 แสดงการควบคุมขา Phase ของ CA 3318

การควบคุม CA 3318

เราสามารถใช้ขาควบคุมเฟส (ขา 19) ควบคุมความเร็วการแปลงสัญญาณของ CA 3318 ได้ 2 วิธี คือ

วิธีแรกโดยการป้อนลอจิก "0" เข้าที่ขาควบคุมเฟส แสดงดังรูปที่ 4.4 (ก) $\phi 1$ จะถูกจัดให้อยู่ในลอจิก "1" และ $\phi 2$ ถูกจัดอยู่ในลอจิก "0" ของสัญญาณนาฬิกา ข้อมูลจากตัวเปรียบเทียบ (อะนาลอก) จะถูกแลตช์ไว้ที่ขอบขาขึ้น ของพัลส์ $\phi 1$ และเมื่อถึงขอบขาลงของพัลส์ $\phi 1$แล้ว ก็จะทำการเลื่อนข้อมูลนั้นส่งเข้าเอาต์พุตรีจีสเตอร์ต่อไป ซึ่งวิธีการนี้เป็นการแปลงข้อมูลเสร็จสิ้นภายในครึ่งคาบ เวลาของสัญญาณนาฬิกาเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีที่สองโดยการป้อนลอจิก "1" เข้าที่ขาควบคุมเฟสเช่นกัน แสดงดังรูปที่ 4.4 (ข) $\phi 1$ ถูกจัดให้อยู่ในลอจิก "0" และ $\phi 2$ ถูกจัดให้อยู่ ในลอจิก "1" ของสัญญาณนาฬิกา เมื่อถึงช่วงขอบขาลงของพัลส์ $\phi 2$ ข้อมูลจากตัวเปรียบเทียบ (อะนาลอก) ถูกแลตช์ไว้จนกว่าจะถึงช่วงขอบขาขึ้นของพัลส์$\phi 2$ ลูกต่อมาจึงทำการเลื่อนข้อมูลส่งเข้าเอาต์พุตรีจิสเตอร์ วิธีนี้จะแปลงสัญญาณเสร็จสิ้นภายใน 1 คาบเวลาของสัญญาณนาฬิกา

จะเห็นว่าวิธีแรกใช้เวลาในการแปลงสัญญาณน้อยกว่าวิธีที่สอง และสำหรับงานที่ต้องการความรวดเร็วในการแปลงสัญญาณควรเลือกควบคุม CA 3318 ด้วยวิธีแรกจะเหมาะสมกว่า

คุณสมบัติของ CA 3318

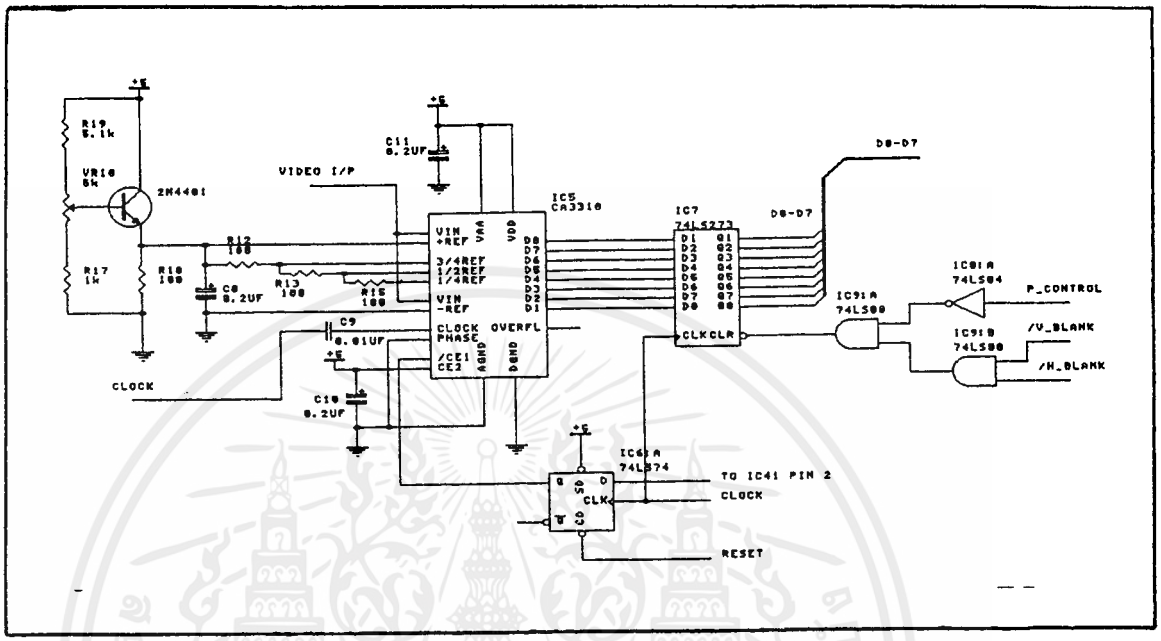
คุณสมบัติแบบคร่าว ๆ ของ CA 3318 มีดังนี้

- ใช้เทคโนโลยี CMOS/SOS
- ใช้เทคนิคการแปลงข้อมูลแบบขนาน
- อัตราการแปลงข้อมูล 15 MSPS ที่ 5 โวลต์
- ใช้เอาต์พุตขนาน 8 บิท
- ใช้แหล่งจ่ายไฟชุดเดียว 4 โวลต์ ถึง 6.5 โวลต์
- แยกระบบกราวด์ของอะนาลอกกับดิจิตอลออกจากกันโดยเด็ดขาด
- กำลังงานสูญเสีย 200 มิลลิวัตต์
- แรงดันอินพุตอยู่ในช่วง 0 - 6.4 โวลต์
- สัญญาณนาฬิกา 20 MHz

การทำงานของวงจร

วงจร Analog to Digital ดังแสดงในรูป 4.5 Vin จะถูกป้อนเข้าที่ขา 21 และ 16 โดยมีชุดสร้างแรงดันอ้างอิง ทั้งทางด้านบวก และด้านลบ ขา Phase เราจะใส่เป็น "0" ข้อมูลจะถูก Latch ขึ้นตรงขอบขาขึ้นของ Clock และเมื่อถึงขอบขาลงแล้วข้อมูลจะถูก Shift ไป Output วิธีนี้จะทำการแปลงข้อมูลได้รวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงวงจร Analog to Digital ที่ใช้ IC เบอร์ CA 3318

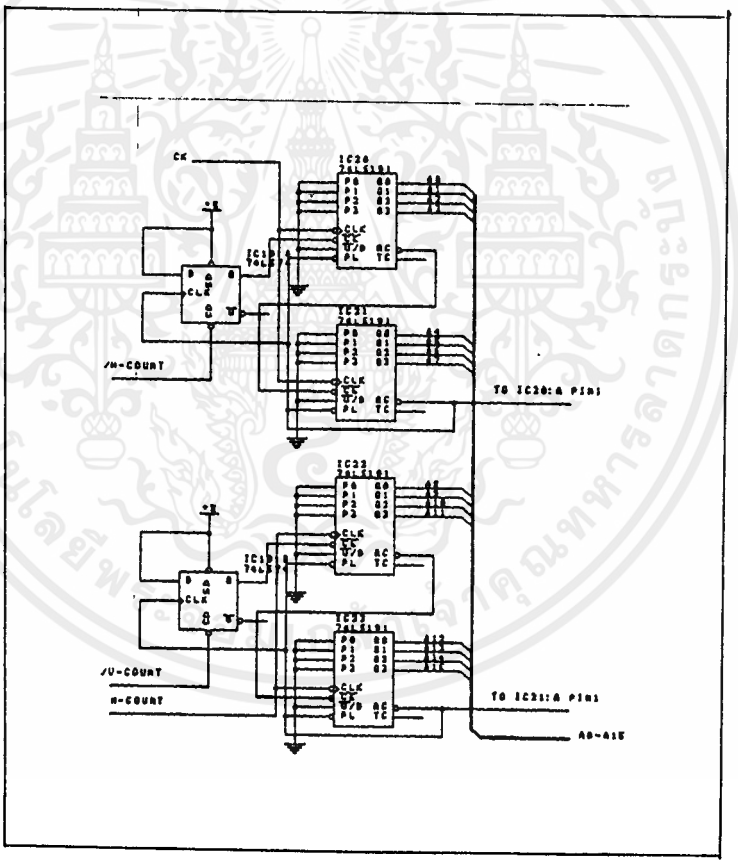
Output ของ CA 3318 จะถูกส่งไปให้ IC7 ซึ่งเป็น Buffer ด้วยมีสัญญาณ /V-Blank , /H-Blank cJt P-Control เป็นตัวควบคุม เมื่อภาพสิ้นสุดลงในแต่ละเส้นสแกน สัญญาณ /H-Blank ก็จะเปลี่ยนจาก "H" เป็น "L" ไปทำการเคลียร์ IC7 ก็จะไม่มีข้อมูลไปเก็บที่ RAM และเมื่อสิ้นสุดการสแกนในแต่ละ 1 ฟิลด์ และเมื่อสัญญาณ /V-Blank ก็จะเปลี่ยนจาก "H" เป็น "L" ในทำนองเดียวกันก็จะไม่มีข้อมูลส่งออกที่ IC7

เมื่อทำการแสดงภาพนิ่ง เมื่อขา /CE1 เป็นลอจิก 1 ก็จะทำให้ Buffer ภายใน IC อยู่ในสภาวะ High impedence ก็จะไม่มีข้อมูลออกที่เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.วงจร COUNTER

การทำงานของวงจร Counter จะใช้ IC เบอร์ 74LS191 จำนวน 4 ตัว ทำการนับให้ได้ 65536 ตำแหน่ง จากวงจรในรูป 5.6 แสดงวงจร Counter การทำงานของวงจรจะให้ขาอินพุตของ IC ทุกตัวเป็น "0" เป็นการบอกให้ Counter เริ่มนับจาก 0 ที่ขา Up/Down ให้เป็น "0" เพื่อให้ Counter นับขึ้น ขา Ripple ของ IC20 จะต่อเข้ากับขา /CE ของ IC21 ในทำนองเดียวกัน ขา Ripple ของ IC22 ชุดล่าง ก็จะต่อเข้ากับขา /CE ของ IC23



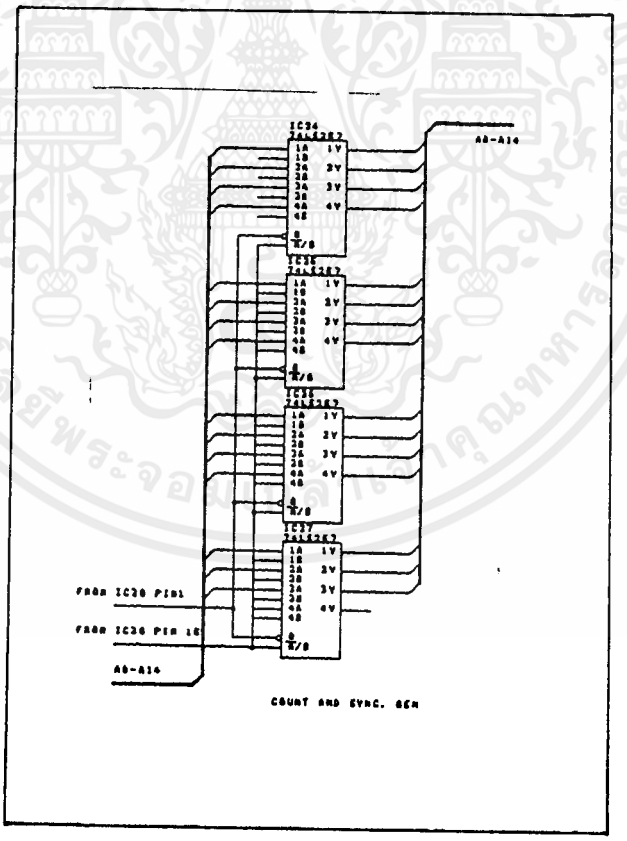
รูปที่ 5.6 แสดงวงจร Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสัญญาณ /H-Count เปลี่ยนจาก "H" เป็น "L" ที่ขา Q ของ D F/F จะเป็น "L" IC20 จะทำการนับ และส่ง Ripple ออกมาที่ขา RC ก็จะส่งไปที่ IC21 ทำการนับต่อไปอีกจนครบ 256 ตำแหน่ง

ส่วน IC ชุดล่างก็จะทำงานคล้ายกับชุดบน คือ เมื่อสัญญาณ /V-Count เป็น "L" ที่ขา /CE ของ IC22 จะเป็น "L" Counter ชุดล่างก็จะทำการนับจนครบ 65536 ตำแหน่ง

สัญญาณเอาต์พุตที่ได้จาก Counter จะถูกส่งผ่านเข้าวงจร Multiplex เพื่อทำการตัดต่อ Address ของการ์ดเข้ากับ PC เมื่อทำการแสดงภาพนิ่งหรือภาพเคลื่อนไหว โดยถ้าจะทำการแสดงภาพเคลื่อนไหว จะต้องป้อนที่ขา Output Control เป็นลอจิก "0" ที่ขา Select (A/B) เป็นลอจิก "1"

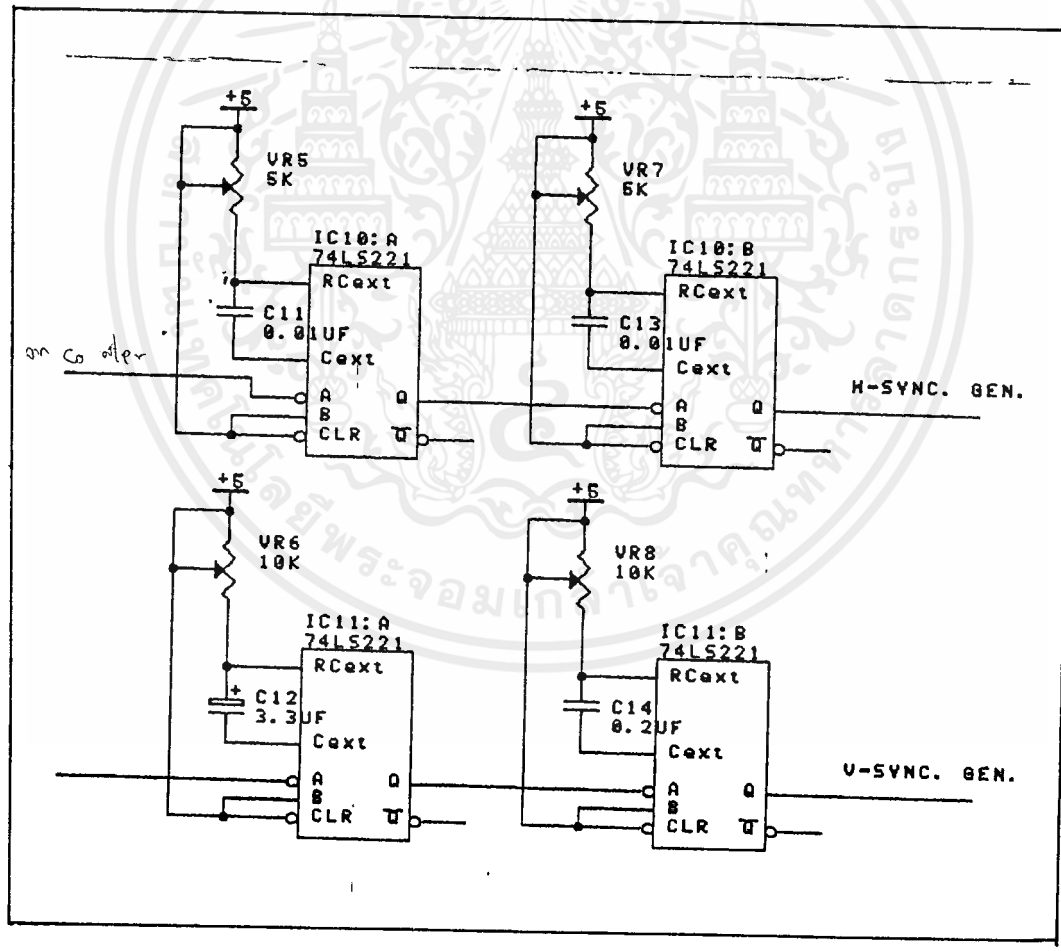


รูปที่ 5.7 แสดงวงจร Multiplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Sync. Generator

การทำงานของ Sync. Gen. จะอาศัยวงจร Monostable เป็นตัวกำเนิด โดยรับอินพุตจากวงจร Counter โดยที่วงจร Counter ชุดบนทำการนับจนครบ 256 ตำแหน่ง ซึ่งหมายถึงการสแกนครบ 1 เส้นทางด้านแนวนอน ที่ขา Ripple จะปรากฏ Plus ออกมา 1 ลูก เราจะใช้ Plus นี้ไปทริกวงจร Monostable ทำการสร้างสัญญาณ Sync. ทางด้าน Hor. ออกมา 1 ลูก โดยที่มี VR5 เป็นตัวปรับความกว้างของสัญญาณ Sync. ส่วน VR7 นั้น จะเป็นการปรับตำแหน่งของสัญญาณ Sync. การสร้างสัญญาณ V-Sync. นั้นเมื่อ Counter ชุดล่างทำการนับได้ 64 K



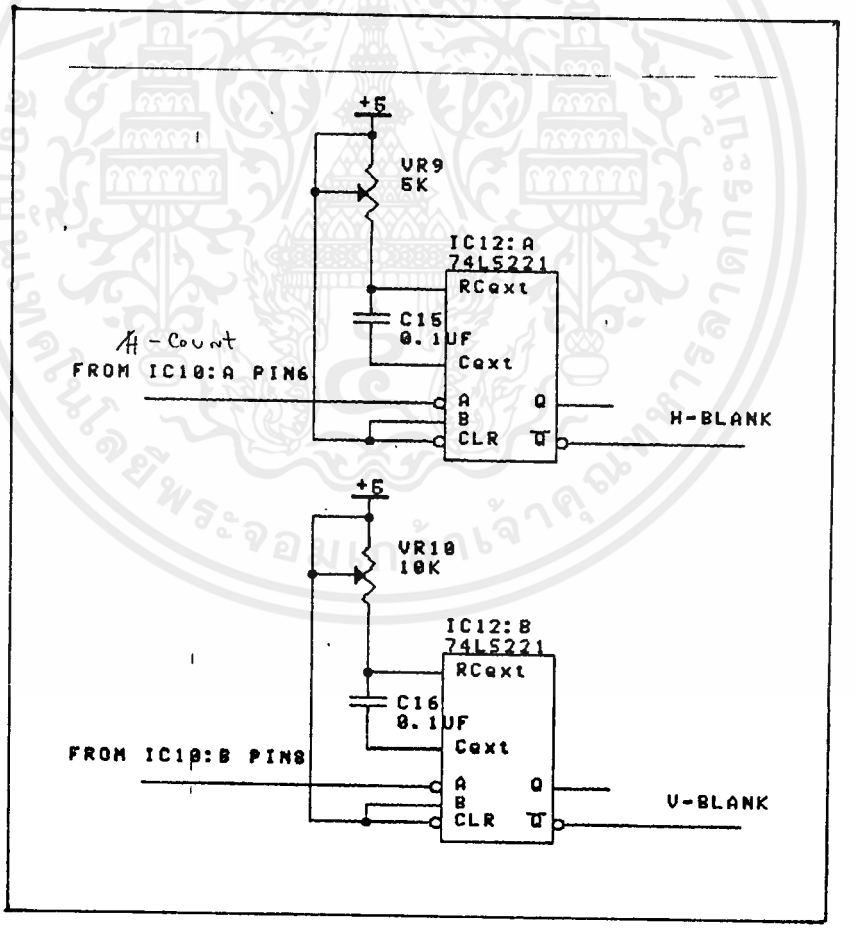
รูปที่ 5.8 แสดง การสร้างสัญญาณ V-Sync. และ H-Sync.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งแล้วที่ขา Ripple จะให้ Plus ออกมา 1 ลูกเช่นกัน Plus ที่ออกมาก็จะนำไปทริก IC12:A ให้ Monostable ทำงาน โดยมี VR5 และ VR7 เป็นตัวปรับความกว้าง และตำแหน่งของสัญญาณ Sync. วงจรการสร้าง H-Sync. และ V-Sync. แสดงดังรูป 5.8

การสร้างสัญญาณ H-Blank และ V-Blank นั้น ก็จะใช้วงจร Monostable เป็นตัวสร้างเช่นกัน การสร้าง H-Blank จะนำสัญญาณ /H-Count มาทำการทริกวงจร Monostable ให้ทำงาน ซึ่งเราสามารถปรับความกว้างของสัญญาณได้โดยการปรับ VR9

การสร้าง V-Blank เมื่อขา /Q เป็นลอจิก "0" วงจร Monostableจะกำเนิด Plus ขึ้นมา 1 ลูก ซึ่งสามารถปรับความกว้างได้โดยปรับ VR10วงจรการสร้างแสดงดังรูป 5.9

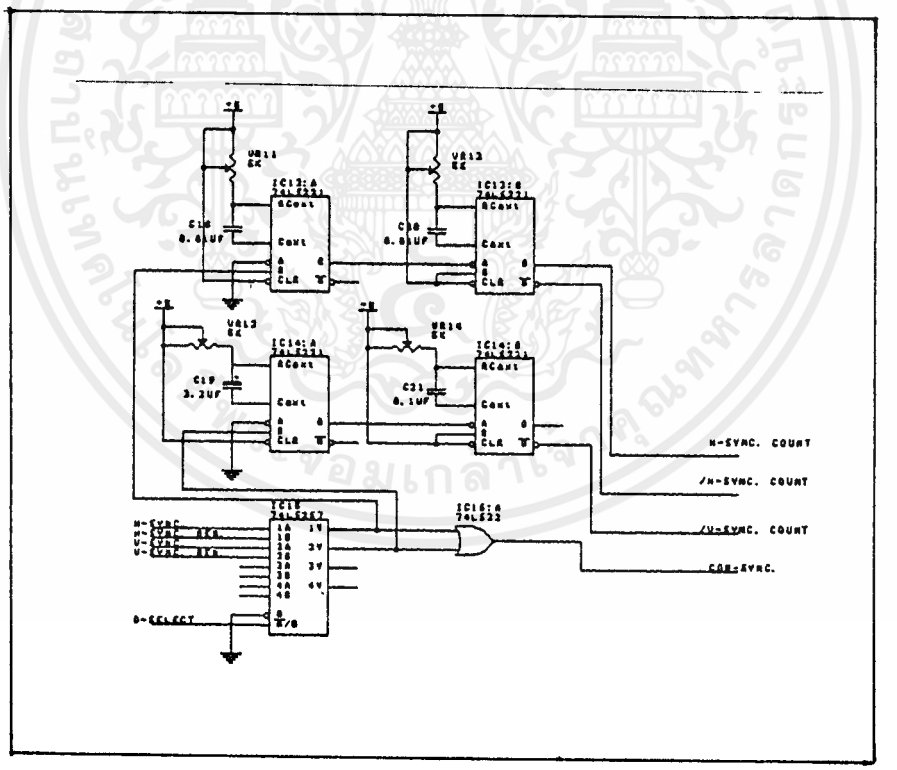


รูปที่ 5.9 แสดงการสร้าง H-Blank และ V-Blank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.CONTROL COUNTER

การสร้างสัญญาณควบคุมวงจร Counter จะทำงานได้ 2 ลักษณะคือการแสดงภาพนิ่ง และการแสดงภาพเคลื่อนไหว โดยทั้ง 2 จะทำงานเหมือนกัน เพียงแต่จะใช้สัญญาณ Sync. คนละชุดกันเท่านั้น โดยเมื่อจะแสดงภาพเคลื่อนไหว IC29 ซึ่งเป็น IC Multiplex จะทำการเลือกสวิทช์ชุด A ออกทางเอาต์พุต สัญญาณ H-Sync. และ V-Sync. จะผ่านวงจร Monostable โดยชุดแรกจะทำการปรับความกว้างของสัญญาณ H-Sync. Count และ V-Sync. Count ส่วนชุดที่ 2 จะทำการปรับตำแหน่งของสัญญาณทั้งสอง ในขณะเดียวกันที่เอาต์พุตของ IC Multiplex จะผ่าน OR Gate เพื่อรวมสัญญาณทั้งสองเข้าด้วยกัน แล้วนำไปรวมกับสัญญาณภาพเมื่อทำการส่งออกแสดงผล วงจรการสร้างแสดงดังรูป 5.10

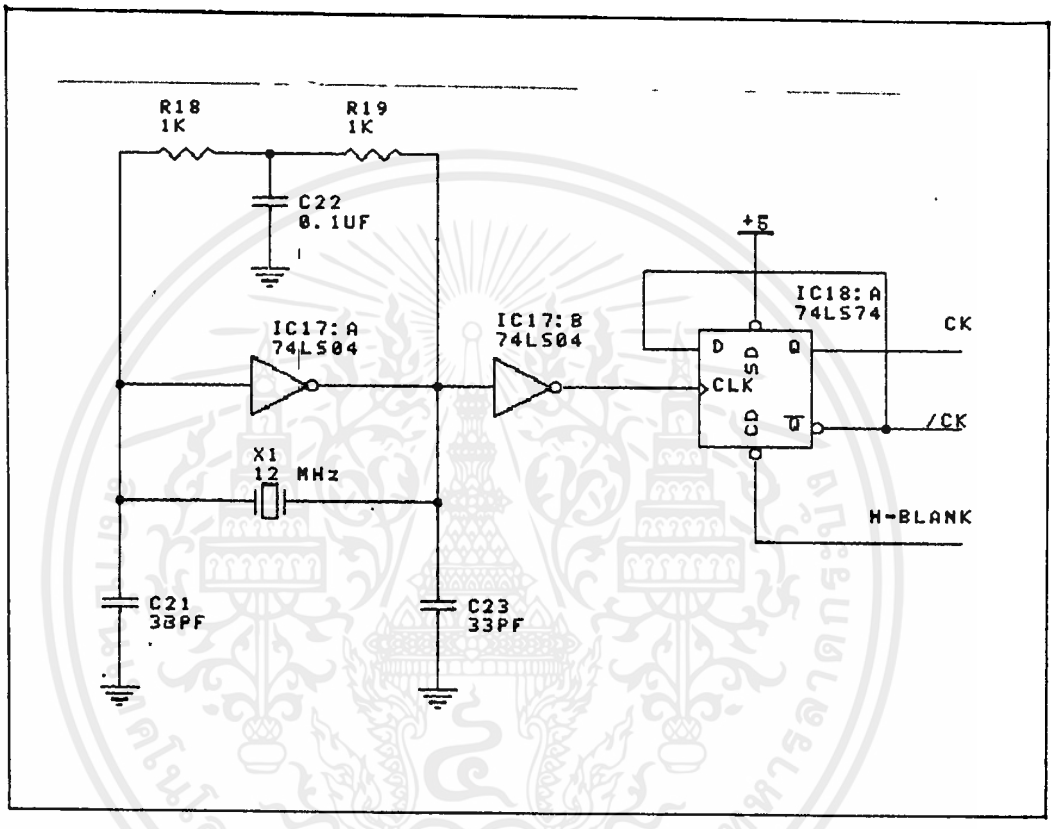


รูปที่ 5.10 แสดงการสร้างวงจรควบคุม counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. CLOCK CIRCUIT

การสร้างสัญญาณ Clock แสดงดังรูป 5.11



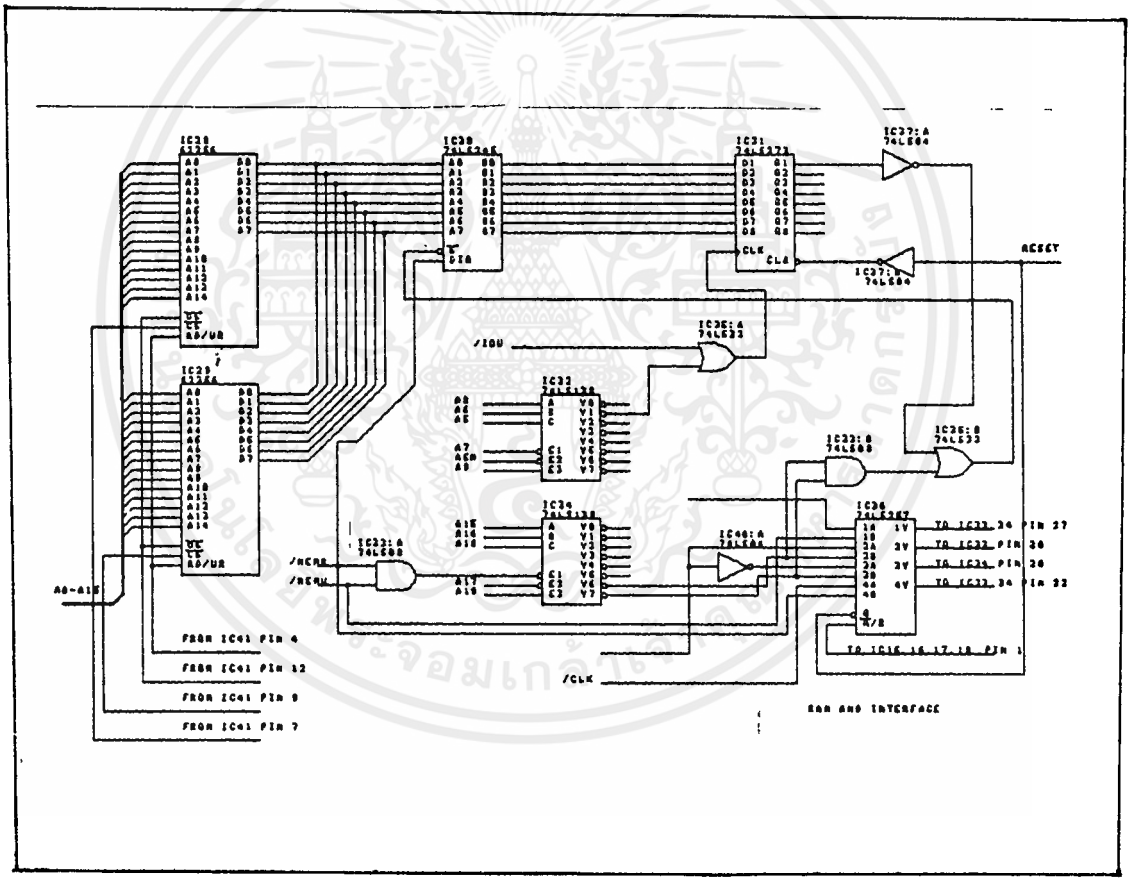
รูปที่ 5.11 แสดงวงจร Clock Circuit

การทำงานจะอาศัย NOT Gate เป็นตัวสร้างสัญญาณ CK ร่วมกับ Cystal ขนาด 12 MHz จากนั้นจะถูก D F/F ทำการหาร 2 เพื่อให้ได้ CK ขนาด 6 MHz ที่ใช้ในการ Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. RAM And Interface

RAM ที่ใช้ในการ์ดจะเป็น RAM ขนาด 32K 8 บิท จำนวน 2 ตัว โดยแต่ละตัวจะมีขา /CE เป็นตัวควบคุมว่าจะเลือกใช้ตัวไหน การเลือก RAM จะใช้ IC 74LS138 ทำการ Decode สัญญาณ Address ที่ใช้ในการเลือก ที่ขา Data ของ RAM จะต่อผ่าน Buffer และ IC Multiplex

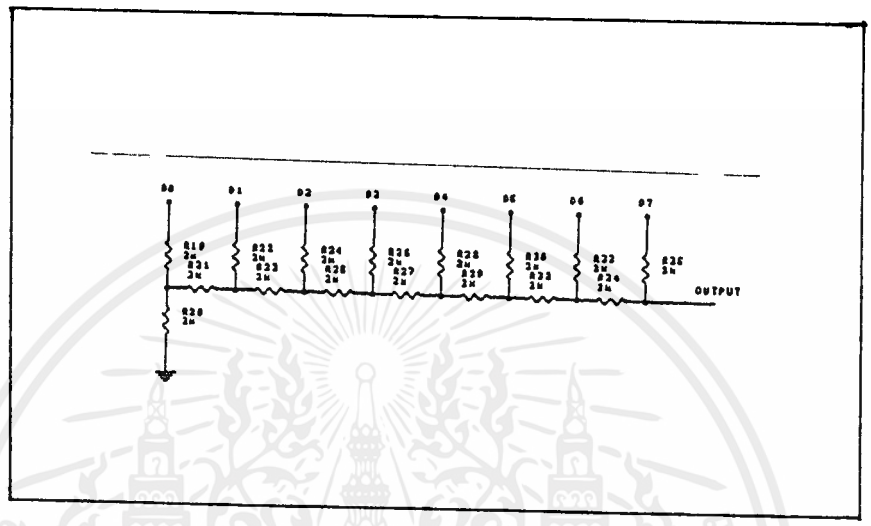


รูปที่ 12 แสดงวงจรการเชื่อมต่อ RAM และส่วน Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Digital To Analog

การเปลี่ยนสัญญาณ Digital ไปเป็นสัญญาณ Analog จะใช้ Network แบบ R-2R Ladder ซึ่งแสดงดังรูปที่ 5.12



รูปที่ 5.12 แสดงวงจร R-2R Ladder

Output ของวงจร R-2R Ladder จะสามารถให้ความแตกต่างของ ระดับสัญญาณได้ 256 ระดับ (2^8) ซึ่งผลรวมของ Voltage Output ที่ได้จากบิทต่าง ๆ สามารถหาได้จากสมการ

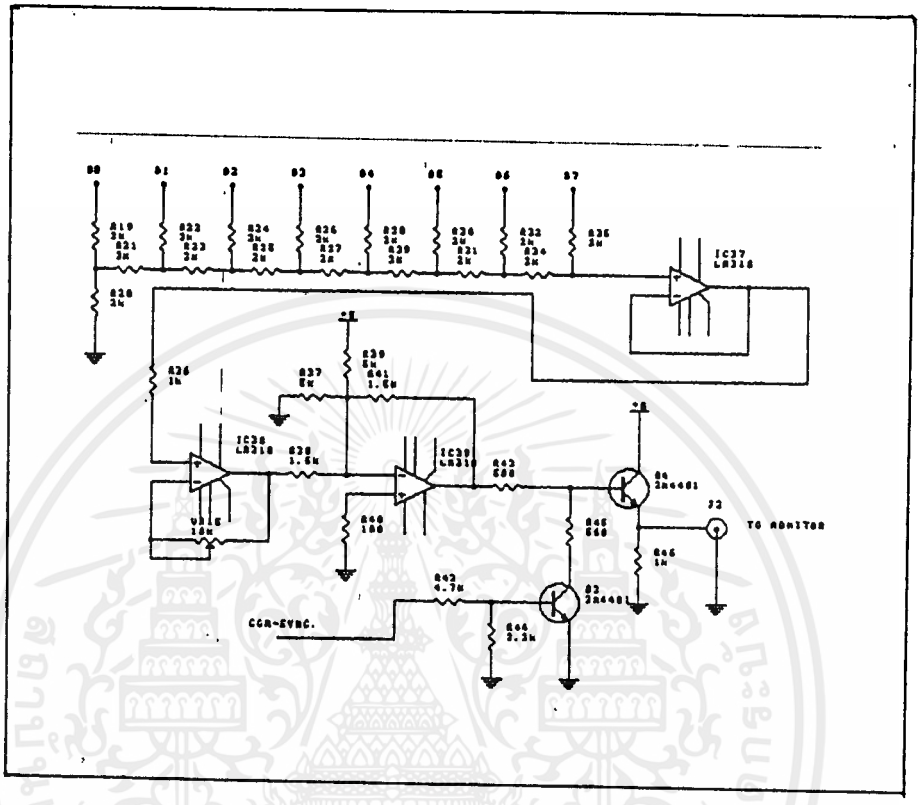
$$Va = (V0\ 2^0 + V1\ 2^1 + V2\ 2^2 + V3\ 2^3 + \ldots\ldots + Vn\ 2^n) / 2^n$$

โดยที่ n = จำนวน Input ที่ใช้

V0,V1,V2,V3,Vn = ระดับของ Voltage ที่ Input

สัญญาณ Digital เมื่อเปลี่ยนเป็นสัญญาณอนาล็อกแล้ว จะผ่าน IC44 ซึ่งเป็น Buffer ของวงจร จากนั้นจะถูกไปให้ IC 42 ทำการปรับอัตราการขยายสัญญาณ แล้วจึงนำไปรวมกับสัญญาณ H-Sync. , V-Sync. , H-Blank , V-Blank ส่งออกแสดงผลที่จอ Monitor ซึ่งวงจรสมบูรณ์แสดงดังรูป 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 แสดงวงจรสมบูรณ์ของวงจร DAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์ฟแวร์

การทำงานของวงจรอิเล็กทรอนิกส์ ที่มีส่วนประกอบหรือส่วนที่เกี่ยวข้องเป็นไมโครโปรเซสเซอร์(Microprocessor) หรือที่นิยมเรียกกันว่าซีพียู(CPU: Central processing Unit) นั้นขั้นตอนการทำงานของระบบเพื่อให้บรรลุตามจุดประสงค์ที่วางไว้ จะต้องมีโปรแกรมควบคุมการทำงานระบบ ตัวโปรแกรมหรือที่เรียกว่า ซอร์ฟแวร์(Soft Ware) นี้จะประกอบขึ้นด้วยล่าดับคำสั่งต่าง ๆ ที่ต้องการให้ซีพียูดำเนินการตามความต้องการของเราในขอบข่าย ที่ซีพียูจะสามารถทำได้ ประสิทธิภาพและความสามารถของวงจรส่วนหนึ่งจะขึ้นอยู่กับฮาร์ดแวร์ (Hard Ware) อย่างเรื่องดิจิไตเซอร์ (Digitizer) นี้ความสามารถของฮาร์ดแวร์คือสามารถดิจิไตซ์ (Digitize) ได้ครั้งละ 8 บิท (bit) นั้นก็หมายความว่าเครื่องดิจิไตเซอร์เครื่องนี้ สามารถให้ความแตกต่างของภาพได้เท่ากับ 2^8 หรือเท่ากับ 256 ระดับเทา (64 gray level) นั้นเองซึ่งค่านี้จะคงที่ตลอดจนกว่าเราจะเปลี่ยนตัวไอซีที่ทำหน้าที่เปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter) นี้ใหม่ ขอบข่ายความสามารถส่วนนี้ก็จะเปลี่ยนไป ในอีกส่วนหนึ่งนั้นความสามารถของวงจรจะขึ้นอยู่กับซอร์ฟแวร์ ซึ่งส่วนนี้เป็นความสามารถที่ไม่ฟิกซ์ตายตัว ขึ้นอยู่กับความสามารถของผู้เขียนโปรแกรมว่าจะสามารถเขียนโปรแกรมสั่งงานเครื่องในระดับไหน เปลี่ยนแปลงความต้องการได้ตามลักษณะของงานโดยไม่ต้องไปยุ่งเกี่ยวกับฮาร์ดแวร์เลย

โปรแกรมของเครื่องดิจิไตซ์ที่เขียนขึ้นนี้ มีฟังค์ชั่นๆ หลักอยู่ 3 โปรแกรม ซึ่งเขียนด้วยภาษาปาสคาล (เทอร์โบปาสคาล) ดังมีรายละเอียดต่อไปนี้

1. <u>โปรแกรมแสดงผล</u> เป็นโปรแกรมแสดงภาพที่เก็บอยู่ภายในแผ่นดิสก์ ซึ่งเป็นไฟล์ขนาด 256*256 จุด ภาพมีระดับความสว่างไม่เกิน 256 ระดับ เมื่อแสดงภาพบนจอ Monitor แต่ถ้าต้องการแสดงภาพบนจอ Computor เนื่องจากโปรแกรมที่เขียนขึ้นนี้แสดงผลใน Mode CGA ความละเอียด 320x200 Pixel ใน Mode นี้จะสามาถแสดงความแตกต่างของสีได้เพียง 4 สีเท่านั้น

2. <u>โปรแกรมเก็บภาพ</u> เป็นโปรแกรมเก็บภาพจากกล้องสำหรับในโครงงานนี้ใช้กล้องโทรทัศน์วงจรปิด หลักการเก็บภาพคือส่งค่า 02H ออกมาที่พอร์ท0300H เพื่อทำการส่งให้การด์ทำการเก็บภาพที่ดิจิไตซ์ โปรแกรมจะคอยตรวจสอบว่ามีการกดคีย์บอร์ดหรือไม่ ถ้ามีจะส่งค่า 080H ออกมาที่พอร์ทเดิม ทำให้เกิดเป็นภาพนิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บนจอทีวี และเมื่อต้องการเก็บภาพก็ส่งค่า 061H ออกมาที่พอร์ท 0300H แล้วจึงทำการอ่านข้อมูลเพื่อเก็บลงในไฟล์ต่อไป

3. โปรแกรมพิมพ์ภาพ ใช้พิมพ์ภาพที่มีระดับความสว่างออกทางเครื่องพิมพ์ โดยมีหลักการคือ ในหนึ่งจุดภาพมีระดับความสว่าง 256 ระดับ ทำการกดลงมาให้เหลือเพียง 64 ระดับ โดยสร้างตารางค้นหา (LOOK UP TABLE) แล้วขยายหนึ่งจุดภาพนี้เป็น 8*8 โดยสร้างเมตริกซ์ขนาด 8*8 ใช้แทนระดับความสว่างได้ 64 ระดับรูปแบบของเมตริกซ์มีมากมายแล้วแต่ว่าจะใช้รูปแบบใดที่จะให้ความเข้มดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

1. ขั้นตอนการทดลองและผลที่ได้

จากทฤษฎีที่กล่าวมาแล้วข้างต้นจะเห็นว่าโครงงานนี้จะประกอบไปด้วย 3 ส่วนคือ ส่วนของวงจรแปลงอนาล็อกเป็นดิจิตอล,ส่วนของการเก็บภาพและส่วนของการติดต่อกับเครื่องคอมพิวเตอร์ เมื่อเราเอาทั้ง 3 ส่วนมาประกอบเข้าด้วยกัน ปัญหาแรกที่พบคือข้อมูลออกมาผิดพลาดดูไม่เป็นรูปตามที่อินพุทป้อนเข้ามา รวมทั้งมีสัญญาณรบกวนปะปนเข้ามาด้วย

จากการวิเคราะห์แล้วว่าช่วงเวลาเขียนภาพไม่เหมาะสม จึงคิดวิธีแก้ไขโดยการใช้ monostable ที่สามารถปรับค่าเวลาที่เอ้าท์พุทได้ เมื่อมีสัญญาณเข้ามา trigger แล้วป้อนเข้าขา /RW ของ RAM แต่ละตัว แล้วทำการปรับคาบเวลาครั้งแรกปรับคาบเวลาที่แคบแล้วทำการทดลอง ผลการทดลองที่ได้จะเหมือนที่กล่าวมาข้างต้น ต่อมาเราจะปรับให้ช่วงเวลามีค่ามากที่สุด ภาพที่ปรากฏออกมาจะดูไม่เป็นภาพ เนื่องจาก noise กวนซึ่งต่างจากแบบแรกซึ่งแบบแรกจะไม่เกิดภาพ ดังนั้นเราจึงสรุปวิธีการแก้ไข โดยใช้ผลการทดลองทั้งสองครั้งที่ผ่านมา ซึ่งสรุปได้ว่า ช่วงเวลาในการเขียนภาพไม่เหมาะสมจึงไม่ทำให้เกิดภาพ ดังนั้นเราต้องหาช่วงเวลาที่เหมาะกับการเขียนภาพกล่าวคือ จะต้องกำหนด cycle ของการเขียนกับสัญญาณข้อมูลที่เข้ามาให้พอดีกันทั้งสองสัญญาณ

ในการปรับช่วงเวลาเพื่อให้ได้ภาพที่สมบูรณ์ที่สุด ทำได้โดยการคำนวณดังนี้ จากทฤษฎีข้างต้น 1 pixel เราใช้ความถี่ 6 MHz ซึ่งใน 1 pixel คือข้อมูลที่เราต้องการจะเขียนเข้าไปในหน่วยความจำ 1 address การเขียนข้อมูลนี้จากที่กล่าวมาแล้วข้างต้น เราต้องหาค่าที่เหมาะสมก็คือค่า 6 MHz และถ้าเราแปลงมาเป็นเวลาที่เท่ากับ 0.166 mS (T = 1/f) เราจะเอาค่านี้มาเป็นช่วงของการเขียนหน่วยความจำยังไม่ได้เพราะจะต้องคำนึงถึงช่วงเวลา access time ของ RAM

และอีกวิธีหนึ่งคือการปรับโดยการประมาณค่าช่วงเวลา โดยให้ภาพที่ออกมาสมบูรณ์ที่สุด สาเหตุที่ทำเช่นนี้ก็เพราะเราต้องคำนึงถึงค่า delay ของอุปกรณ์รวมด้วย ดังนั้นเราจึงไม่สามารถใช้การคำนวณได้

เมื่อเราปรับช่วงให้เหมาะสมแล้วจะให้ภาพออกมาแต่ยังมี noise ปนมาอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ยังมีน้อยกว่าตอนแรกที่ยังไม่มีการต่อ monostable สาเหตุที่ noise เข้ามาปนมากก็อาจจะเนื่องจากในขณะที่เราทำการทดลองเรายังใช้ board ทดลองอยู่แล้วใช้สายโทรศัพท์เชื่อมระหว่างภาคแต่ละภาคที่กล่างมาแล้วเข้าด้วยกัน ประกอบกับสัญญาณที่มาควบคุมโครงงานนี้ที่ได้จากเครื่องคอมพิวเตอร์ ยังมี noise อยู่และที่สำคัญที่สุดที่ทำให้เกิด noise ก็คือภาควงจร A/D ซึ่งจากการทดลองเราวัดสัญญาณที่อินพุท ซึ่ง A/D เบอร์ 3318 แล้วปรากฏว่าสัญญาณข้อมูลที่เข้ามามีnoise ปนมาเพิ่มมากขึ้นเพราะวงจรของ A/D เองก็มี noise อยู่แล้ว

การทดลองโหลดไฟล์ข้อมูลภาพออกมาแสดง

- หลังจากที่ได้เก็บภาพไปแล้ว ต่อไปก็จะนำภาพเดิมที่เก็บไว้กลับมาดูหลังจากที่เก็บภาพแล้วเราก็เลือกฟังก์ชั่นของซอร์ฟแวร์เป็นภาพต่อเนื่อง คือฟังก์ชั่น Display เพื่อให้ภาพเดิมเปลี่ยนไปก่อนที่จะเอาภาพเก่าออกมาแสดง เพื่อให้เห็นว่าเราเก็บภาพเอาไว้ได้

- ต่อจากนี้ก็เลือกฟังก์ชั่นโหลด ซึ่งก็คือการนำเอาข้อมูลภาพที่เก็บไว้ในดิสก์มาลงใน Memory ของบอร์ดจะเห็นได้ว่าภาพที่เก็บไว้ สามารถนำกลับมาแสดงได้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บทสรุปและวิจารณ์

การแปลงภาพเป็นข้อมูลคอมพิวเตอร์ ภาพมีลักษณะเป็น 2 มิติโดยเราอาจกำหนดให้มีแกนแนวนอนเป็น X และแนวตั้งเป็น Y ที่จุดพิกัด (X,Y) ใด ๆ จะมีค่าความเข้มของภาพซึ่งเราเรียกว่า ระดับเทา (gray level)หากกล่าวในเชิงคณิตศาสตร์ก็คือที่จุดพิกัด (X,Y) ใด ๆ จะมีค่าระดับเทาเขียนเป็นฟังก์ชั่นได้ว่า P (X , Y)

ประโยชน์ของการใช้คอมพิวเตอร์ประมวลสัญญาณภาพ

เมื่อเราสามารถแปลงและเก็บข้อมูลภาพไว้ในรูปแบบของข้อมูลคอมพิวเตอร์แล้ว จากนั้นจะเป็นการใช้ความรู้เรื่องคณิตศาสตร์ในการประมวลข้อมูลภาพต่อไป เช่นการกำจัดสัญญาณรบกวน (noise reduction) การค้นหาขอบ (edge delection) การลดข้อมูล (data reduction) การจำแนกภาพ (pattern recognition) ตลอดจนการสร้างภาพ (reconstrution) เป็นต้น

ซึ่งในการกระทำขบวนการต่าง ๆ เหล่านี้นั้นต้องอาศัยซอร์ฟแวร์ที่เขียนขึ้นเพื่อทำขบวนการต่าง ๆ การประมวลผลต่าง ๆ เหล่านี้ด้วยคอมพิวเตอร์จึงจะต้องอาศัยข้อมูลของภาพที่เป็นข้อมูลทางดิจิตอลจึงจะสามารถทำได้และการ์ดดิจิไตซ์เซอร์เป็นอุปกรณ์ที่ใช้ในการแปลงข้อมูลภาพจากสัญญาณอนาล็อก เป็นสัญญาณดิจิตอล แต่การ์ดที่ทำนี้จะมีการแปลงข้อมูลดิจิตอลเพียง 8 บิทเท่านั้น

ถ้าหากเพิ่มจำนวนบิทมากขึ้นจะสามารถลด round of noise ลงได้ ส่วนการเพิ่มรายละเอียดให้ภาพมีรายละเอียดที่เก็บข้อมูลภาพเพิ่มขึ้น ก็โดยการเพิ่มความเร็วในการ Sampling ให้สูงขึ้นแต่ก็จะต้องเพิ่มขนาดของหน่วยความจำในการเก็บข้อมูลขึ้นอีก ดังนั้นรายละเอียดของภาพก็ขึ้นอยู่กับว่าเรานำไปใช้กับงานประเภทใด

จะเห็นได้ว่าโครงงานนี้เพิ่งเป็นจุดเริ่มต้นของการทดลองพัฒนาเท่านั้น ในการที่จะนำไปใช้งาน หรือพัฒนาต่อไปยังสามารถที่จะดำเนินการได้อีกหลายอย่าง อย่างที่เห็นได้ชัด ๆ ที่น่าควรจะปรับปรุงคือการเพิ่มความแตกต่างของระดับเกย์ที่ภาพสามารถให้ความแตกต่างได้โดยการเปลี่ยนไอซี A to D และในการปรับปรุงขั้นต่อไปคือการทำให้สามารถที่จะทำการดิจิไตซ์และเก็บภาพที่เป็นสีได้ โดยการเพิ่มเติมวงจรและหน่วยความจำขึ้นอีกเพื่อให้สามารถเพียงพอที่จะเก็บภาพสีได้เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่าได้ถึงขนาดนี้แล้วก็สามารถน่าไปใช้งานด้านต่าง ๆ ได้กว้างขวางมากยิ่งขึ้น

3. ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

ปัญหาหลักที่เกิดขึ้นก็คือ ปัญหาจาก noise ที่เกิดจากเดินสาย ซึ่งมีผลท่าให้ภาพมีสัญญาณรบกวนเวลาที่เก็บข้อมูลลงในหน่วยความจ่าแล้วอ่านกลับออกมา(read memory) การแก้ไขที่ได้ท่าไปแล้วในขณะนี้ก็คือ เอาตัวถังของ X-tal ต่อลงกราวด์แล้วจะสามารถที่จะลดปัญหาที่เกิดจากสัญญาณรบกวนได้บ้าง

สาเหตุของการเกิด noise ในภาพ สามารถสรุปเป็นหัวข้อใหญ่ ๆ ดังนี้คือ

1. ระบบของการเชื่อมต่อวงจร
2. ความผิดพลาดของอุปกรณ์ เช่น เมื่อต่อ IC หลายตัวจะต้องค่านึงถึงค่า Delay ของสัญญาณด้วย
3. noise ของแหล่งจ่ายไฟ
4. ความไม่เสถียรภาพของ clock ที่สร้างขึ้น

ส่วนแนวทางแก้ไข ซึ่งปัญหาจากสัญญาณรบกวนเกิดจากการเดินสายเป็นจ่านวนมากจึงควรออกแบบเป็น PCB สำเร็จในแผ่นเดียวกัน ซึ่งจะลดจ่านวนสายที่ใช้ในการต่อวงจรไปเป็นจ่านวนมาก ท่าให้ไม่เกิดความยุ่งยากเหมือนที่ท่าอยู่ขณะนี้ นอกจากนี้ควรใช้วงจรก่าเนิดสัญญาณ clock ที่เป็นวงจรสำเร็จรูป (module clock) ซึ่งจะช่วยลดการแพร่กระจายของสัญญาณรบกวนจากวงจร clock ลงได้ ที่ IC TTL ทุกตัวควรจะต่อ Capacitor คร่อมที่ขา Vcc เพื่อลดแรงดัน Transian ที่เกิดขึ้น

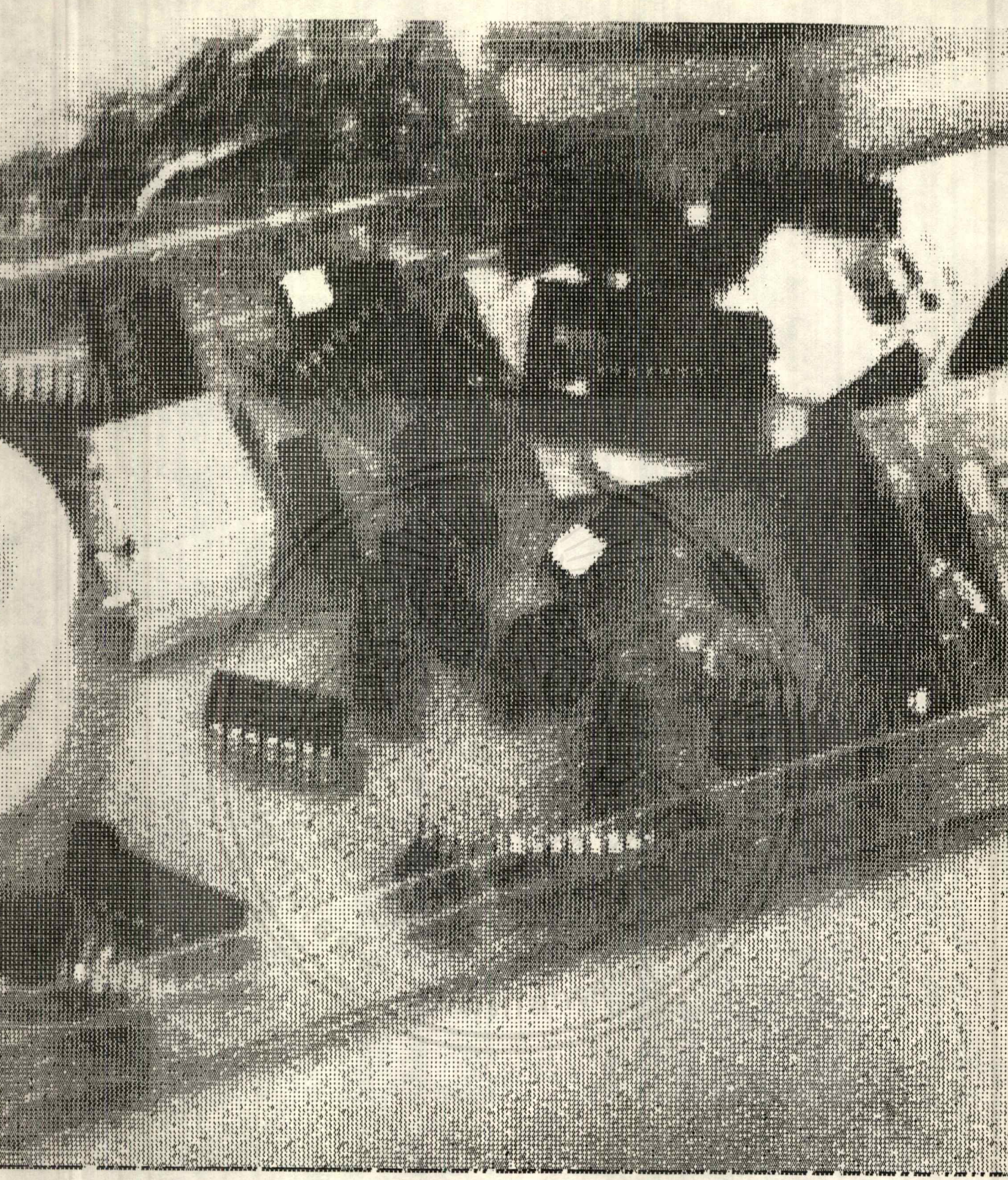
สัญญาณ Sync. ที่สร้างขึ้นใช้งานควรออกแบบให้สามารถปรับค่าความกว้างและต่าแหน่งของสัญญาณได้ เพื่อสะดวกในการรวมสัญญาณ และการน่าสัญญาณไปควบคุมภายในวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

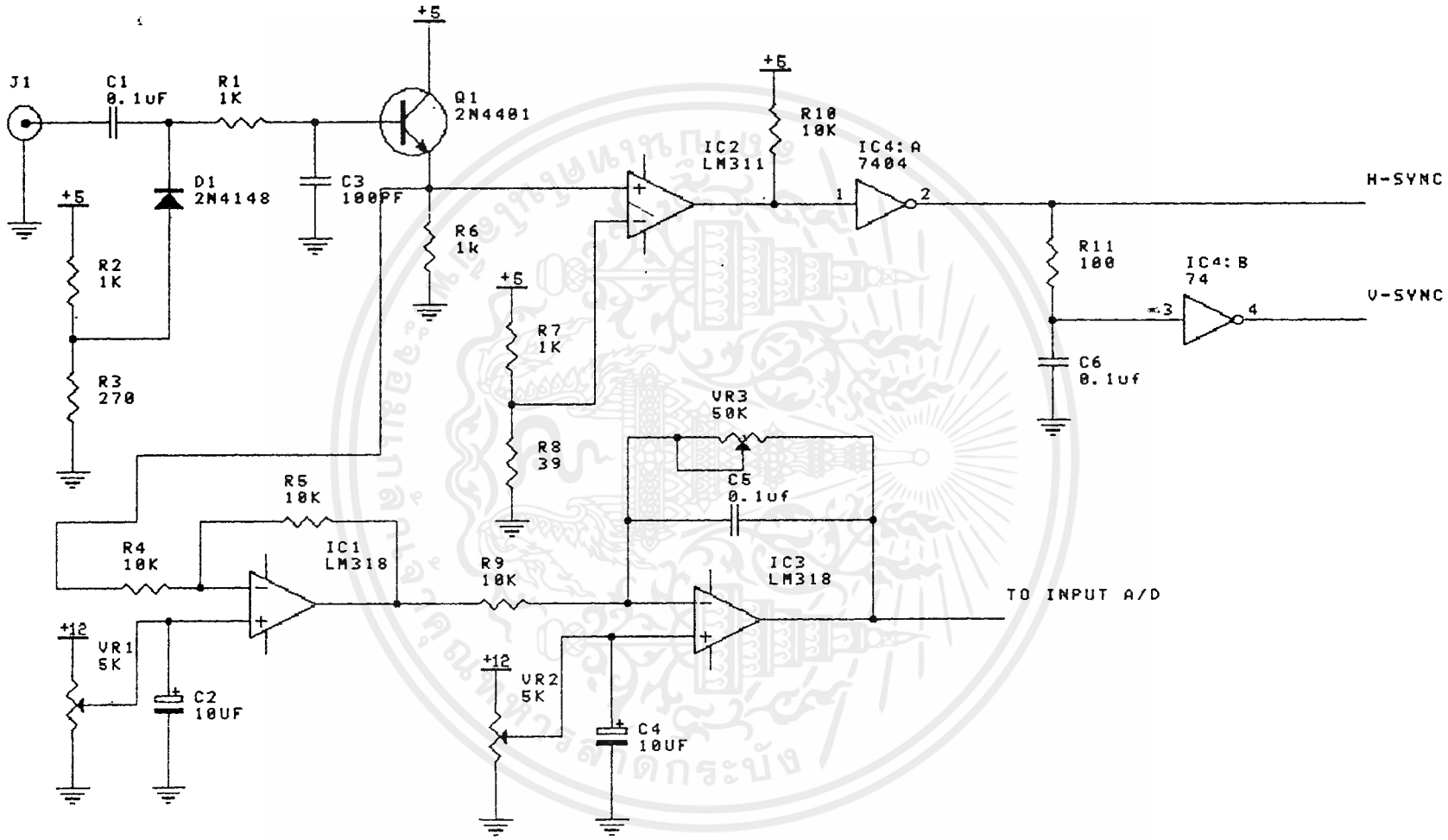
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

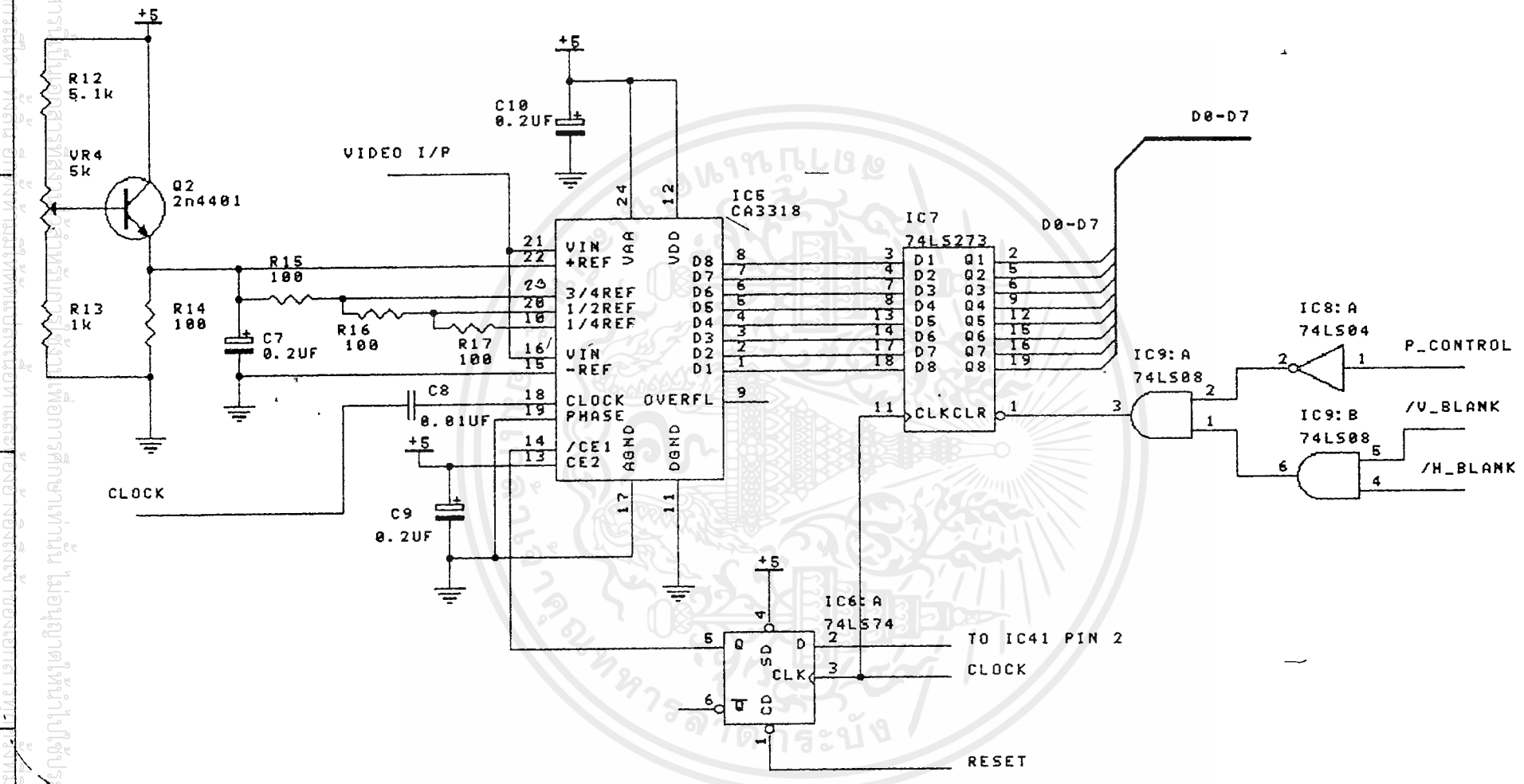


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

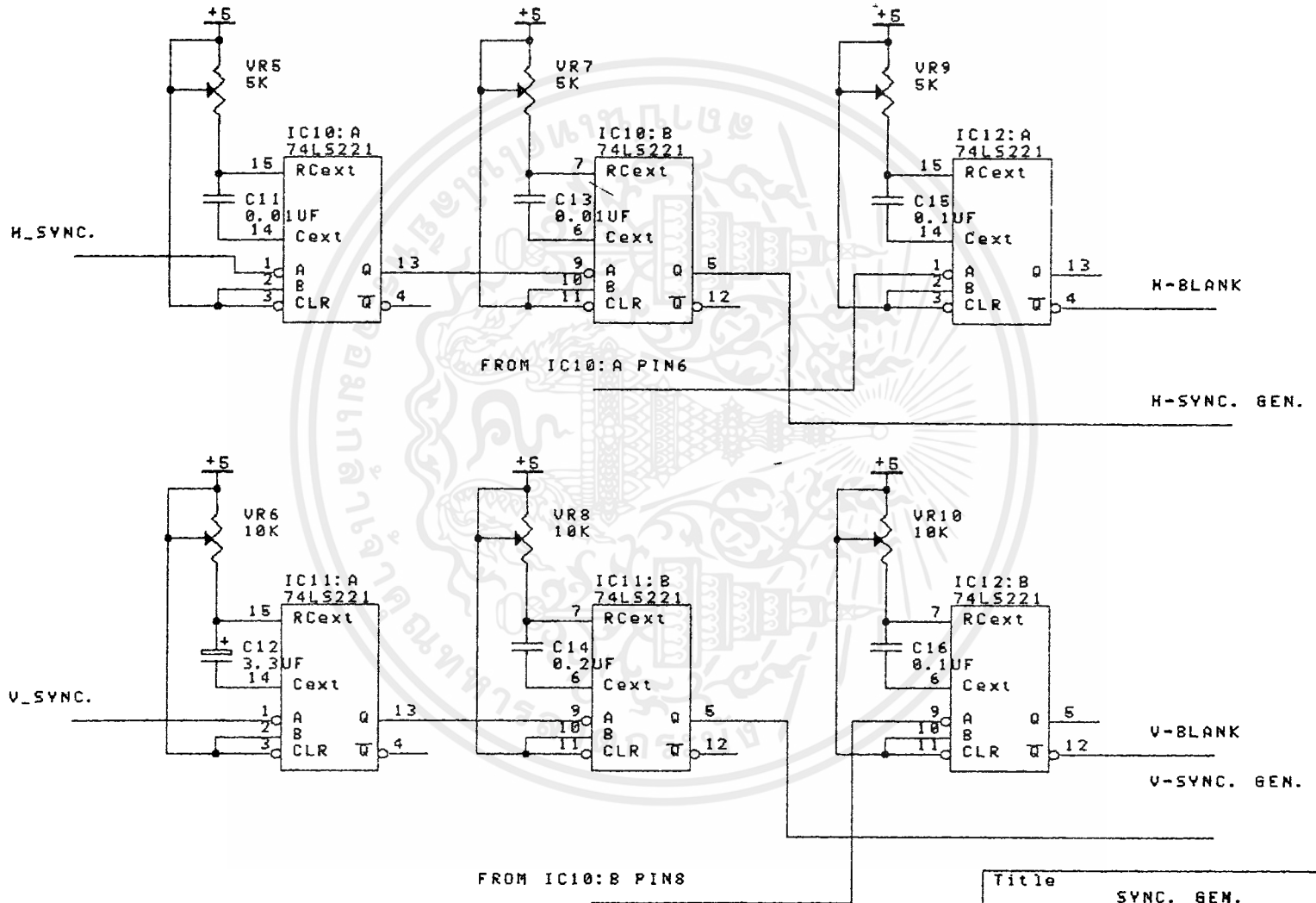
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



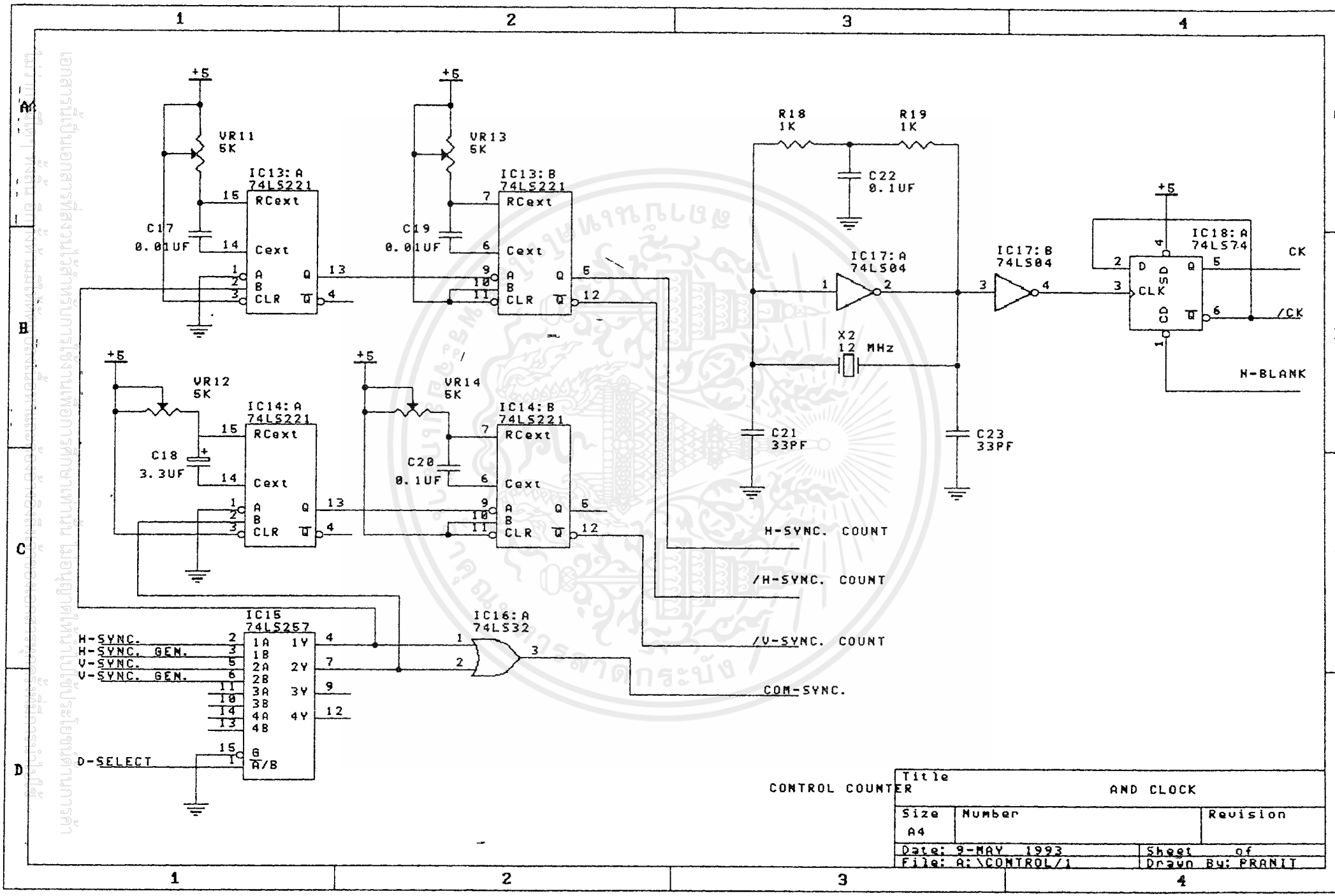
Title: BUFFER AND SYNC. SEP		
Size: A4	Number:	Revision:
Date: 9-MAY 1993		Sheet of
File: A:\BUFFER/1		Drawn By: PRANIT

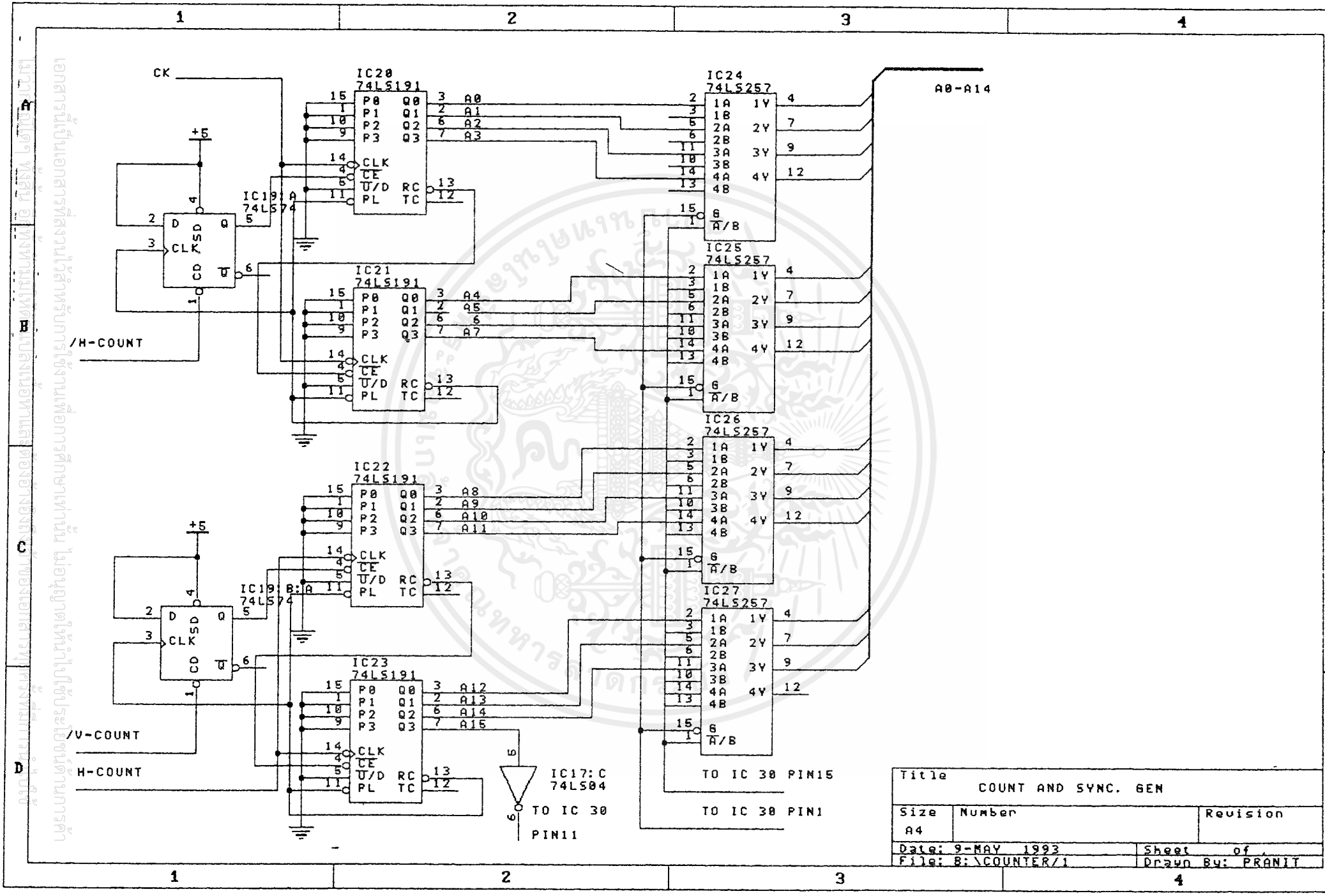


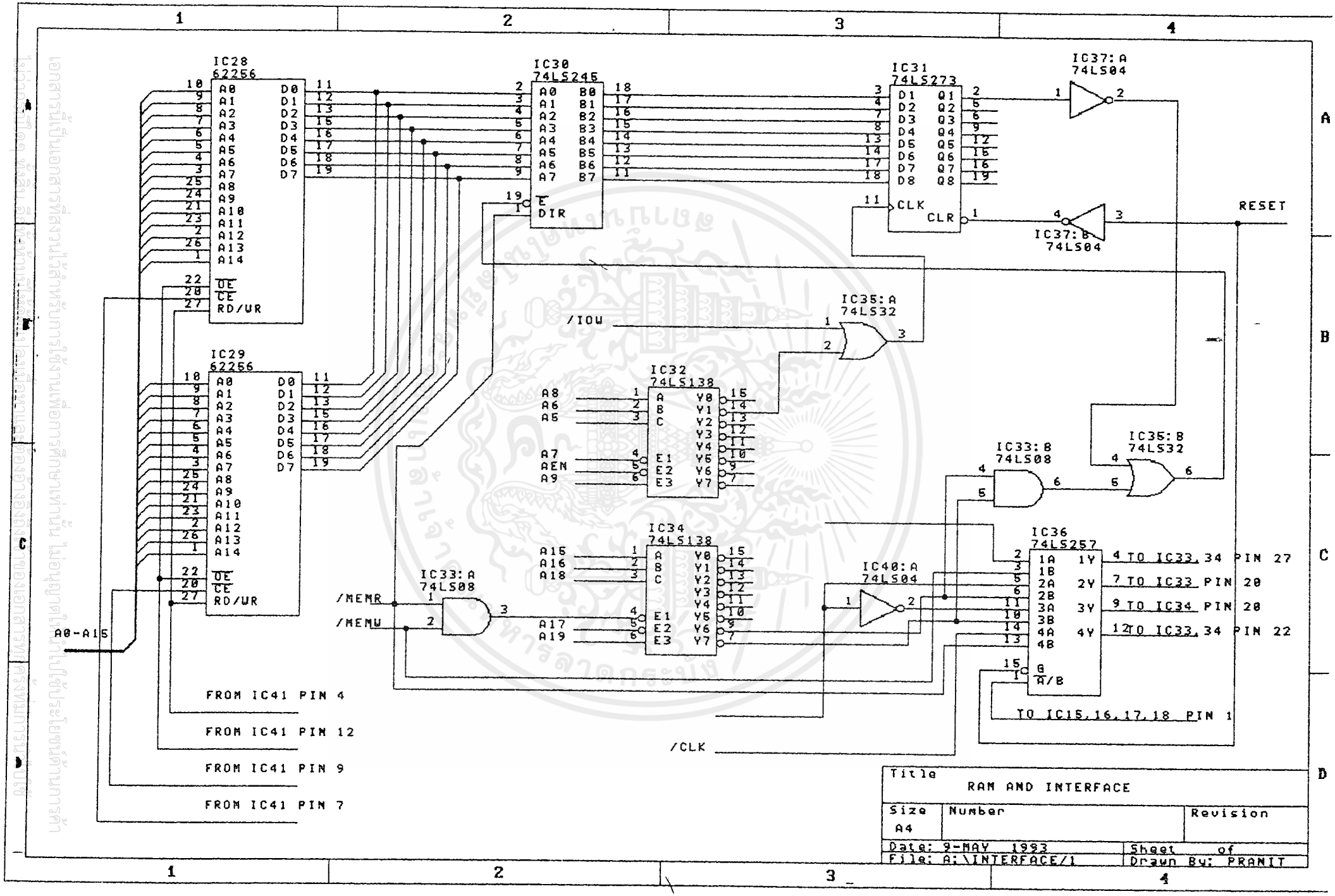
Title: ANALOG TO DIGITAL CIRCUIT		
Size: A4	Number	Revision
Date: 9-MAY 1993		Sheet of
File: A:\ADC/1		Drawn By: PRANIT

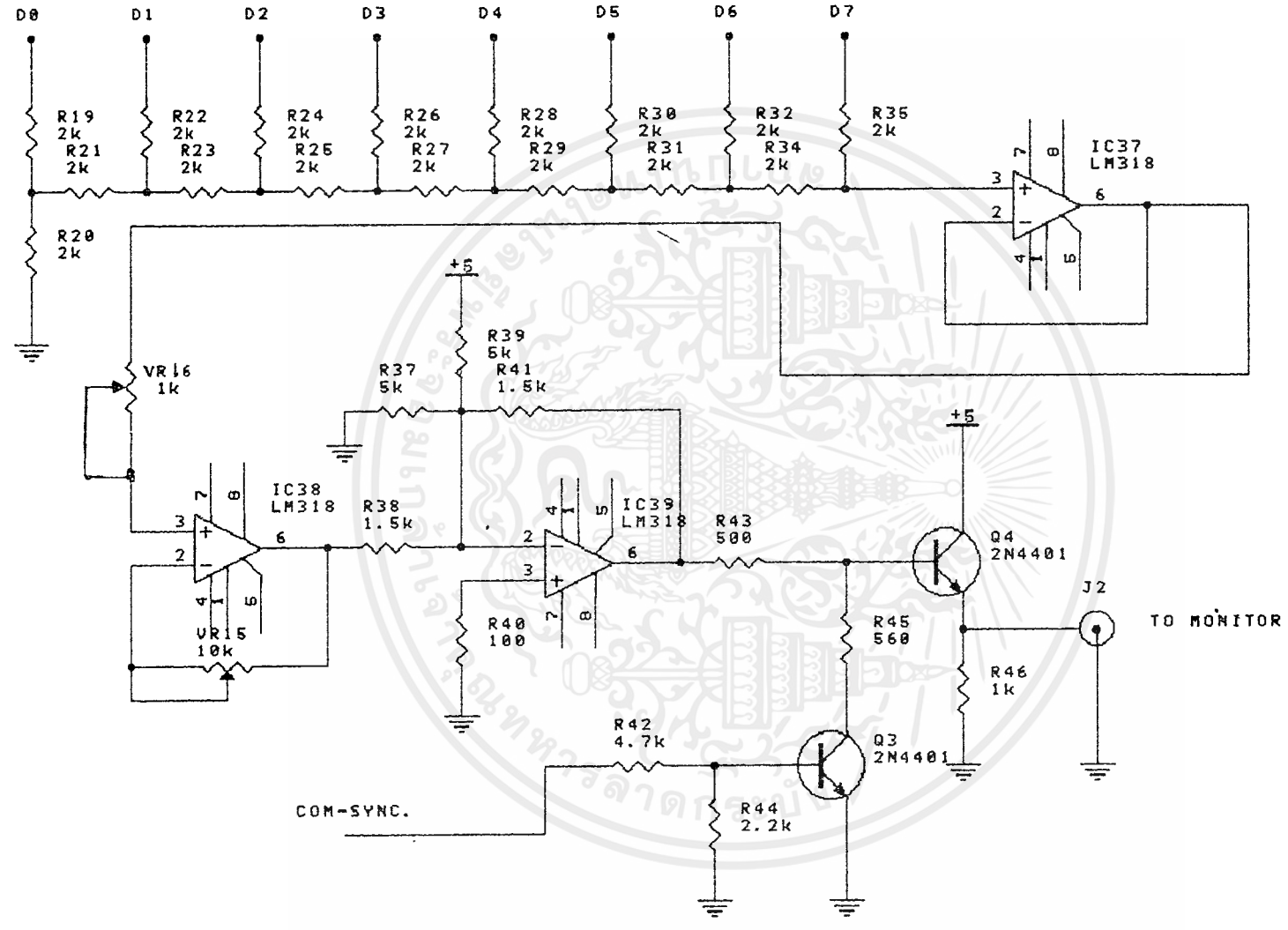


Title: SYNC. GEN.		
Size: A4	Number:	Revision:
Date: 9-MAY 1993		Sheet of
File: A:\SYNC_GEN/1		Drawn By:


+5
VR11
5K
IC13:A
74LS221
RCext
Cext
C17
0.01UF
A
B
CLR
Q
VR13
5K
IC13:B
74LS221
C19
0.01UF
R18
1K
R19
1K
C22
0.1UF
IC17:A
74LS04
IC17:B
74LS04
IC18:A
74LS74
D
CLK
CK
/CK
X2
12 MHz
H-BLANK
VR12
5K
IC14:A
74LS221
C18
3.3UF
VR14
5K
IC14:B
74LS221
C20
0.1UF
C21
33PF
C23
33PF
H-SYNC. COUNT
/H-SYNC. COUNT
/V-SYNC. COUNT
COM-SYNC.
IC15
74LS257
H-SYNC.
H-SYNC. GEN.
V-SYNC.
V-SYNC. GEN.
D-SELECT
IC16:A
74LS32
CONTROL COUNTER
Title
AND CLOCK
Size
A4
Number
Revision
Date: 9-MAY 1993
Sheet of
File: A:\CONTROL/1
Drawn By: PRANIT


CK
IC20
74LS191
IC21
74LS191
IC22
74LS191
IC23
74LS191
IC19:A
74LS74
IC19:B
74LS74
+5
/H-COUNT
/V-COUNT
H-COUNT
IC24
74LS257
IC25
74LS257
IC26
74LS257
IC27
74LS257
A0-A14
IC17:C
74LS04
TO IC 30
PIN11
TO IC 30 PIN15
TO IC 30 PIN1
Title
COUNT AND SYNC. GEN
Size
A4
Number
Revision
Date: 9-MAY 1993
Sheet of
File: B:\COUNTER/1
Drawn By: PRANIT


IC28
62256
IC29
62256
IC30
74LS245
IC31
74LS273
IC32
74LS138
IC34
74LS138
IC35:A
74LS32
IC35:B
74LS32
IC33:A
74LS08
IC33:B
74LS08
IC37:A
74LS04
IC37:B
74LS04
IC40:A
74LS04
IC36
74LS257
RESET
/IOW
/MEMR
/MEMW
/CLK
A0-A15
FROM IC41 PIN 4
FROM IC41 PIN 12
FROM IC41 PIN 9
FROM IC41 PIN 7
4 TO IC33, 34 PIN 27
7 TO IC33 PIN 20
9 TO IC34 PIN 20
12 TO IC33, 34 PIN 22
TO IC15, 16, 17, 18 PIN 1
Title
RAM AND INTERFACE
Size
A4
Number
Revision
Date: 9-MAY 1993
Sheet of
File: A:\INTERFACE/1
Drawn By: PRANIT



Title: DIGITAL TO ANALOG CIRCUIT		
Size: A4	Number	Revision
Date: 9-MAY 1993		Sheet of
File: A:\DAC/1		Drawn By:

CA3318, CA3318C

ANA-DIGIT CO., LTD.
บริษัท แอนนาดิจิท จำกัด
144 ถนนอัษฎางค์ แขวงพระราชวัง กทม.10200
โทร. 2217040, 2214592, 2214557
Fax: 225-2232

Product Preview

CMOS High-Speed 8-Bit Flash A/D Converter

@ 2,250.-

Features:

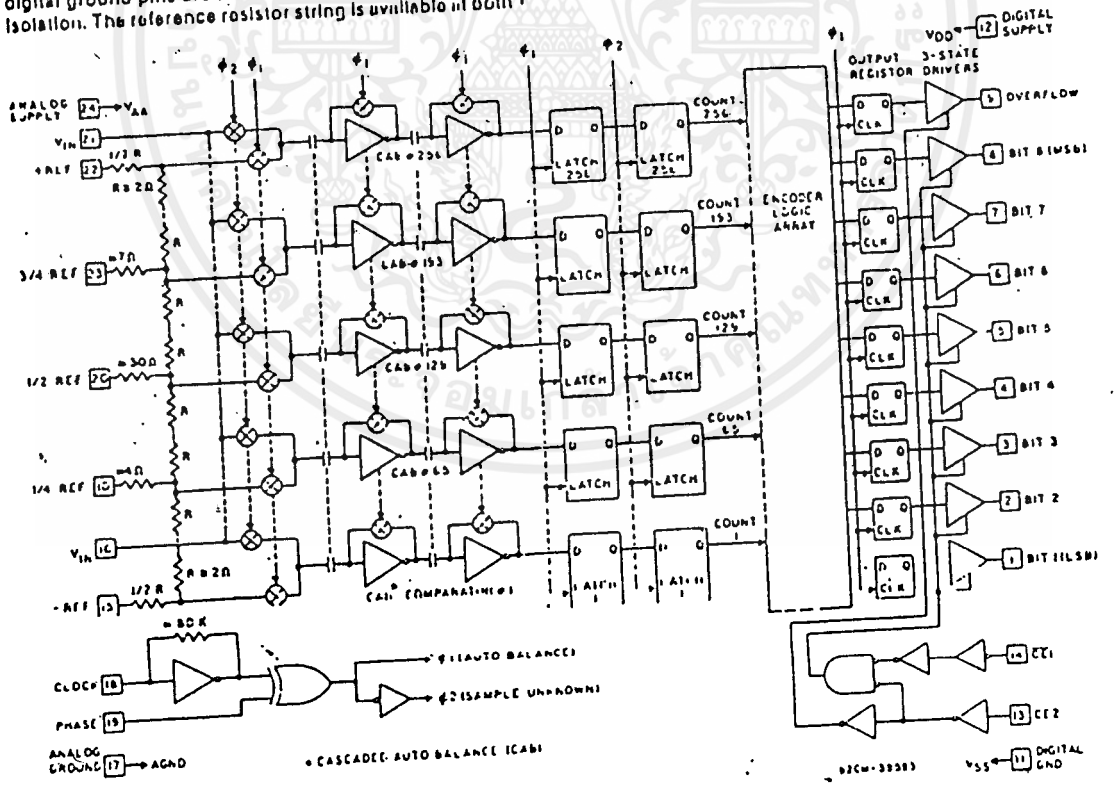
- *Pin compatible with 41051/CA3308*
- *CMOS/SOS low power*
- *Flash (Parallel) conversion technique*
- *15 MSPS conversion rate at 5 V (CA3318C)*
- *20 MSPS conversion rate at 5 V (CA3318)*
- *1 LSB differential linearity*
- *1.5 LSB integral linearity*
- *Single 4 to 6.5 V supply*
- *8 latched bit outputs plus overflow*
- *May be stacked for higher resolution*
- *May be paralleled for double speed*

Applications:

- *Especially suited for high speed applications where low power is also important*
- *TV video digitizing (industrial/security)*
- *Ultrasound signature analysis*
- *Transient signal analysis*
- *General-purpose hybrid ADC's*
- *Optical character recognition*
- *Radar pulse analysis*
- *Motion signature analysis*

The RCA CA3318 and CA3318C are pin compatible retrofits for the 41051/CA3308, but with the output data changing 1/2 clock cycle later. They have features similar to the CA3300 (File No. 1316), such as the control inputs and outputs necessary to allow stacking or paralleling for higher resolution or doubled speed. Separate analog and digital ground pins are available to allow analog to digital isolation. The reference resistor string is available at both + and - ends, and at the 1/4, 1/2, and 3/4 points, thus allowing the tailoring of non-linear transfer functions. In addition, the + reference (positive full scale) may be used above the analog + supply.

The CA3318 and CA3318C are available in a 24-lead dual-in-line plastic package (E suffix) and in a 24-lead dual-in-line ceramic package (D suffix).



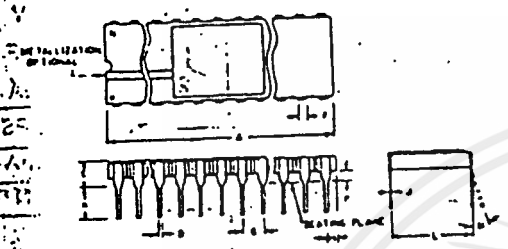
Block diagram of the CA3318 and CA3318C.

Preview Data only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIMENSIONAL OUTLINE

UFFIX
d Dual-In-Line
Brazed Ceramic Package



SYMBOL	INCHES		NOTE	MILLIMETERS	
	MIN.	MAX.		MIN.	MAX.
A	1.180	1.220		29.98	30.98
C	0.085	0.145		2.16	3.68
D	0.015	0.023		0.39	0.58
F	0.040 REF.			1.02 REF.	
G	0.100 BSC			2.54 BSC	
H	0.030	0.070		0.77	1.77
J	0.008	0.012	3	0.21	0.30
K	0.125	0.175		3.18	4.44
L	0.580	0.620	2	14.74	15.74
M	—	7°		—	7°
P	0.025	0.050		0.64	1.27
N	24			24	

92CS-30986R1

NOTES:

1. Leads within 0.005" (0.13 mm) radius of True Position at maximum material condition.
2. Center to center of leads when formed parallel.
3. When this device is supplied solder dipped, the maximum lead thickness (narrow portion) will not exceed 0.013" (0.33 mm).

RNA-DIGIT CO., LTD.
บริษัท รอยนาดิจิท จำกัด
144 ถนนอิษฎางก์ แขวง...เขตธนบุรี กทม.10200
โทร. 2217040, 2214592, 2214557
FAX:225-9232

When incorporating RCA Solid State Devices in equipment, it is recommended that the designer refer to "Operating Considerations for RCA Solid State Devices" Form No. 1CE-402, available on request from RCA Solid State Division, Box 3200, Somerville, N.J. 08876.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา รวมถึงลบข้อความนี้ด้วย มิฉะนั้นจะมีความผิดตามกฎหมายลิขสิทธิ์ และขอสงวนสิทธิ์ในการดำเนินคดีกับผู้ละเมิดลิขสิทธิ์ต่อไป

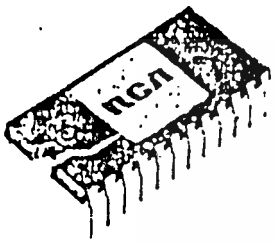
RCA
Solid State Division

Linear Integrated Circuits
Monolithic Silicon

Objective Data

CA3308, CA3308A

File Number 1352



(D) SUFFIX
24-Lead Dual-In-Line
Side-Brazed Ceramic Package

CMOS Video Speed 8-Bit Flash Analog-to-Digital Converter

For Use in Low-Power Consumption, High-Speed Digitization Applications

Features:
- *CMOS low power with SOS speed*
- *Parallel conversion technique*
- *15-MHz sampling rate (66-ns conversion time)*
- *8-bit latched 3-state output with overflow bit*
- *± ½ LSB accuracy (typ.)*
- *Single supply voltage (4 to 8 V)*
- *2 units in series allow 9-bit output*
- *2 units in parallel allow 30-MHz sampling rate*

Applications:
- *The CA3308 is especially suited for high-speed conversion applications where low power is also important*
- *TV video digitizing (industrial/security/broadcast)*
- *High-speed A/D conversion*
- *Ultrasound signature analysis*
- *Transient signal analysis*
- *High-energy physics research*
- *High-speed oscilloscope storage/display*
- *General-purpose hybrid ADCs*
- *Optical character recognition*
- *Radar pulse analysis*
- *Motion signature analysis*
- *μP data acquisition systems*

The RCA CA3308* is a CMOS 200-mW parallel (FLASH) analog-to-digital converter designed for applications demanding both low-power consumption and high-speed digitization.

The CA3308 operates over a wide full-scale input-voltage range of 4 volts up to 8 volts with maximum power consumptions as low as 200 mW, depending upon the clock frequency selected. When operated from a 5-volt supply at a clock frequency of 15 MHz, the power consumption of the CA3308 is less than 150 mW.

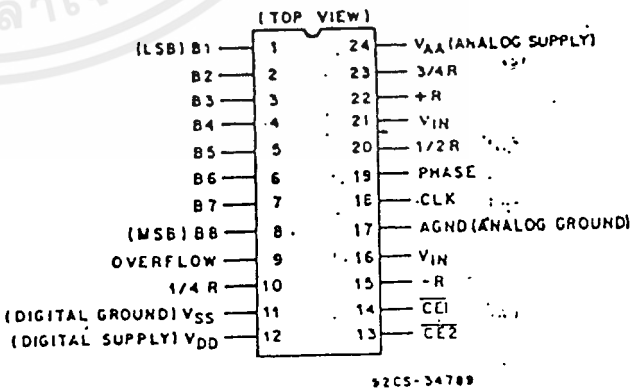
The intrinsic high conversion rate makes the CA3308 ideally suited for digitizing high-speed signals. The overflow bit makes possible the connection of two or more CA3308s in series to increase the resolution of the conversion system. A series connection of two CA3308s may be used to produce a 9-bit high-speed converter. Operation of two CA3308s in parallel doubles the conversion speed (i.e., increases the sampling rate from 15 to 30 MHz). CA3308s may be combined with a high-speed 8-bit D/A converter, a binary adder, control logic, and an op amp to form a very high-speed 15-bit A/D converter.

256 paralleled auto-balanced voltage comparators measure the input voltage with respect to a known reference to produce the parallel-bit outputs in the CA3308.

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.

The voltage supply for analog circuitry is termed V_{AA} and AGND. The voltage supply for digital circuitry is termed V_{DD} and V_{SS}.

The CA3308 type is available in a 24-lead dual-in-line ceramic package (D suffix).



92CS-34789

TERMINAL ASSIGNMENT

*Formerly Developmental Type No. TA11279.

ANA-DIGIT CO., LTD.
บริษัท อนันท์ดิจิท จำกัด
144 ถนนยัชญาเงค์ แขวงพระราชวัง กทม.10200
โทร. 2217040, 2214592, 2214557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา หรือนำไปเผยแพร่ต่อ ก่อนได้รับอนุญาตจากมหาวิทยาลัย

...ation on these devices is intended for engineering evaluation. ...ge designations and data are subject to change, unless otherwise ...ed. No obligations are assumed for notice of change or future ...acture of these devices.

Information furnished by RCA is believed to be accurate and reliable. However, no responsibility is assumed by RCA 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 RCA.

Trademark(s)® Registered
Marca(s) Registrada(s)

Printed In USA/6-82

CA3308 CMOS VIDEO SPEED 8-BIT FLASH A/D CONVERTER

rice Operation

sequential parallel technique is used by the CA3308 nverter to obtain its high-speed operation. The sequence nsists of the "Auto Balance" phase, Ø1, and the "Sample known" phase Ø2. (Refer to the circuit diagram.) Each nversion takes one clock cycle.* With the phase control n 8) high, the "Auto Balance" (Ø1) occurs during the High riod of the clock cycle, and the "Sample Unknown" (Ø2) curs during the low period of the clock cycle.

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

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

here:

$V_{tap}(n)$=reference ladder tap voltage at point n.

V_{REF}=voltage across −REF to +REF

N=tap number (1 through 256)

he other side of these capacitors are connected to single age amplifiers whose outputs are shorted to their inputs switches. This balances the amplifiers at their intrinsic p points, which is approximately, $V_{DD}-V_{SS}/2$. The first t of capacitors, now charge to their associated tap oltages.

the same time a second set of commutating capacitors d amplifiers are also auto-balanced. The balancing of the cond stage amplifier at its intrinsic trip point removes any acking differences between the first and second amplifier ages. The cascaded auto-balance (CAB) technique, used re, increases comparator sensitivity and temperature acking.

the "Sample Unknown" phase, all ladder tap switches d comparator shorting switches are opened. At the same e 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 "low" state at their outputs. All comparators that had tap voltages lower than V_{in} will go to a "high" state.

The status of all these comparator amplifiers are ac coupled through the second stage comparator and stored at the end of this phase (Ø2), by a latching amplifier stage. Once latched, the status of the comparators are decoded by a 256 to 8-bit decode array and the results are clocked into a storage register at the rising edge of the next Ø2.

A 3-state buffer is used at the output of the 8 storage registers which are controlled by two chip-enable signals. CE1 will independently disable B1 through B8 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 CA3308 via the following steps: (Refer to timing diagram No. 1.) 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 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 propagates through the decode array and a 9-bit code appears at the D inputs of the output registers. On the next rising 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 3-state drivers. This also marks the start of a new "sample" phase, thereby repeating the conversion process for this next cycle.

*This device requires only a single phase clock. The terminology of Ø1 and Ø2 refers to the High and Low periods of the same clock.

RCA-DIGIT CO., LTD.
บริษัท อาร์ซีเอ ดิจิท จำกัด
144 ถนนอโศกมนตรี แขวงสีลม เขตบางรัก กทม.10200
โทร. 2217040,2214592,2214557
FAX:225-9232

OPERATING AND HANDLING CONSIDERATIONS

Handling

All inputs and outputs of RCA CMOS devices have a network for electrostatic protection during handling. Recommended handling practices for CMOS devices are described in ICAN-8525 "Guide to Better Handling and Operation of CMOS Integrated Circuits."

Operating

Operating Voltage

During operation near the maximum supply voltage limit, care should be taken to avoid or suppress power supply turn-on and turn-off transients, power supply ripple, or ground noise; any of these conditions must not cause $V_{DD}-V_{SS}$ to exceed the absolute maximum rating.

Input Signals

To prevent damage to the input protection circuit, input signals should never be greater than V_{DD} nor less than V_{SS}. Input currents must not exceed 10 mA even when the power supply is off.

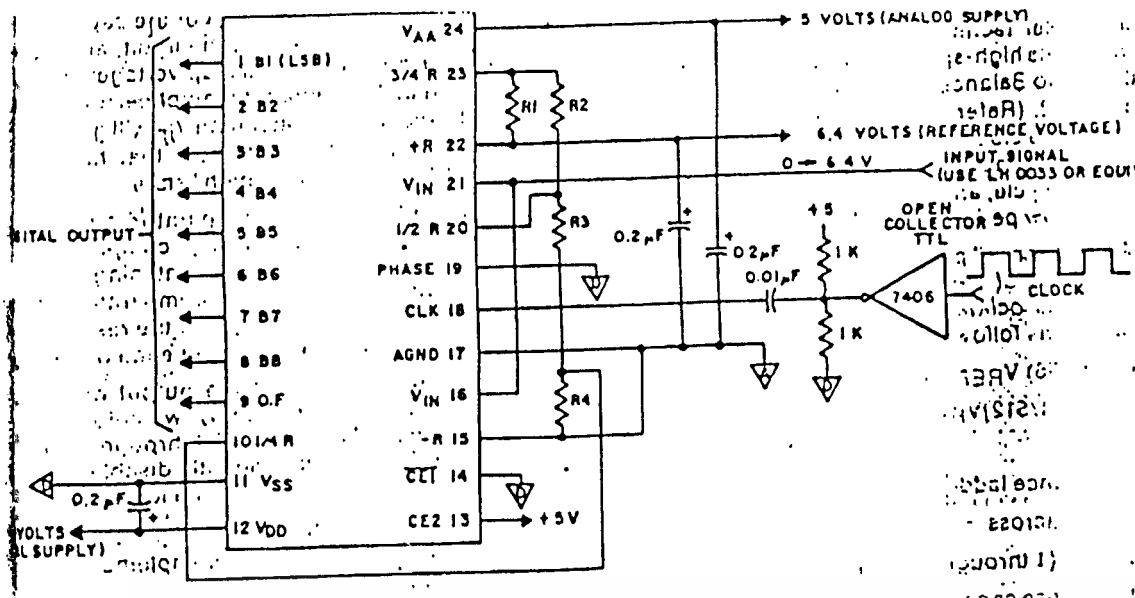
Unused Inputs

A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

Output Short Circuits

Shorting of outputs to V_{DD} or V_{SS} may damage CMOS devices by exceeding the maximum device dissipation.

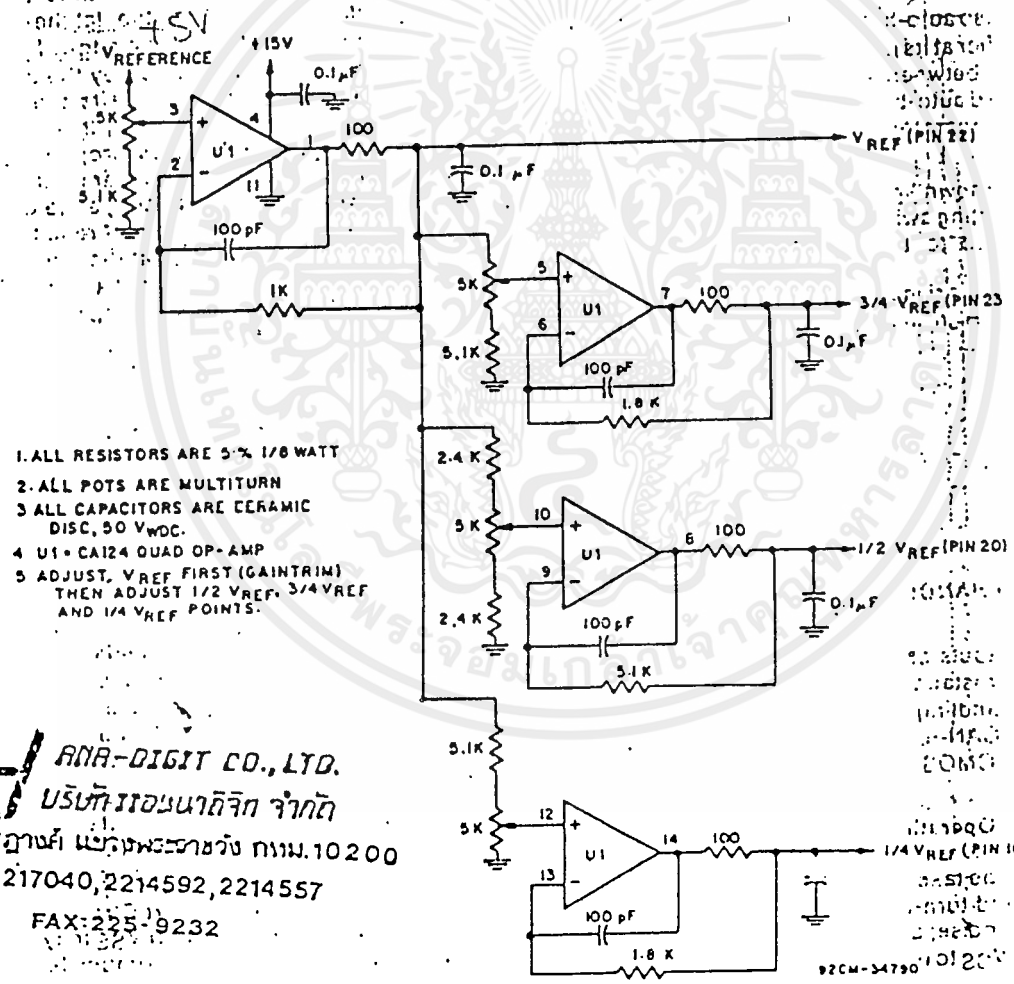
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTES

1. R1 — R4 = 1000 Ω 0.1 % 1/8 WATT (DELETE WHEN USING REFERENCE DRIVER CIRCUIT)
2. A GROUND AND D GROUND MUST BE CONNECTED TO EACH OTHER NEAR THE CHIP.
3. V_{AA} = +6V WILL IMPROVE LINEARITY

Fig. 3-Typical circuit configuration for the CA3308. (15-MHz sampling rate)



1. ALL RESISTORS ARE 5% 1/8 WATT
2. ALL POTS ARE MULTITURN
3. ALL CAPACITORS ARE CERAMIC DISC, 50 V_{WDC}.
4. U1 = CA124 QUAD OP-AMP
5. ADJUST V_{REF} FIRST (GAINTRIM) THEN ADJUST 1/2 V_{REF}, 3/4 V_{REF} AND 1/4 V_{REF} POINTS.

ANA-DIGIT CO., LTD.
บริษัท แอนาดิจิท จำกัด
ถนนอินทรพิทักษ์ แขวงสวนหลวง กทม.10200
โทร. 2217040, 2214592, 2214557
FAX: 225-9232

Fig. 4-Reference driver circuit. (Use for maximum linearity)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS

CHARACTERISTIC	TEST CONDITIONS $V_{AA} = V_{DD}$	LIMITS MIN.	LIMITS TYP.	LIMITS MAX.	UNITS
Resolution		—	—	8	Bits
Linearity Error	V_{DD}=5 V, V_{REF}=6.4 V CLK=15 MHz, gain adjusted	—	—	±0.5 ±1	(CA3308A) (CA3308)
Differential Linearity Error	V_{DD}=5 V_{REF}=6.4 V CLK=15 MHz	—	—	±0.5 ±1	(CA3308A) (CA3308)
Quantizing Error		−½	—	½	LSB
Analog Input: Full Scale Range Input Capacitance Input Current	V_{DD}=5 V CLK=15 MHz V_{IN}= 6.4 V	4 — —	— 50 1000	8 — 2000	V pF μA
Maximum Conversion Speed	V_{DD}=5 V	15 M	17 M	—	SPS
Device Current (Excludes I_{REF})	V_{DD}=5 V (CLK=15 MHz)	—	50	—	mA
Ladder Impedance		300	600	900	Ω
Digital Inputs: Low Voltage High Voltage Input Current (Except Pin 18)	V_{DD}=5 V	— 3.5 —	— — ±1	1.5 — —	V V μA
Digital Outputs: Output Low (Sink) Current Output High (Source) Current	V_{DD}=5 V, V_O=0.4 V V_{DD}=5 V, V_O=4.6 V	3.2 1.6	10 −6	— —	mA
Digital Output Delay, t_d	V_{DD}=5 V	—	25	—	ns

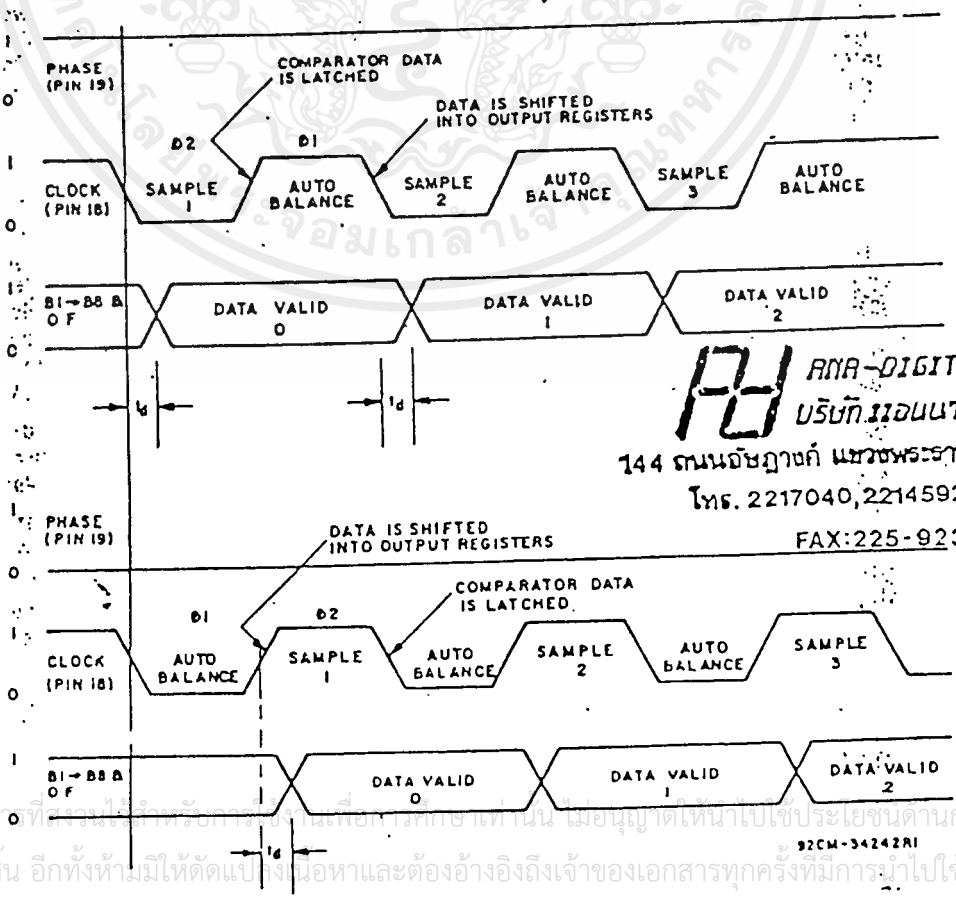


Fig. 2-Timing diagram for the CA3308.

AMA-DIGIT CO.,LTD.
บริษัท เอเอ็นนาดิจิท จำกัด
144 ถนนจันทน์ แขวงพระยาวัง กทม.10200
โทร. 2217040,2214592,2214557
FAX:225-9232

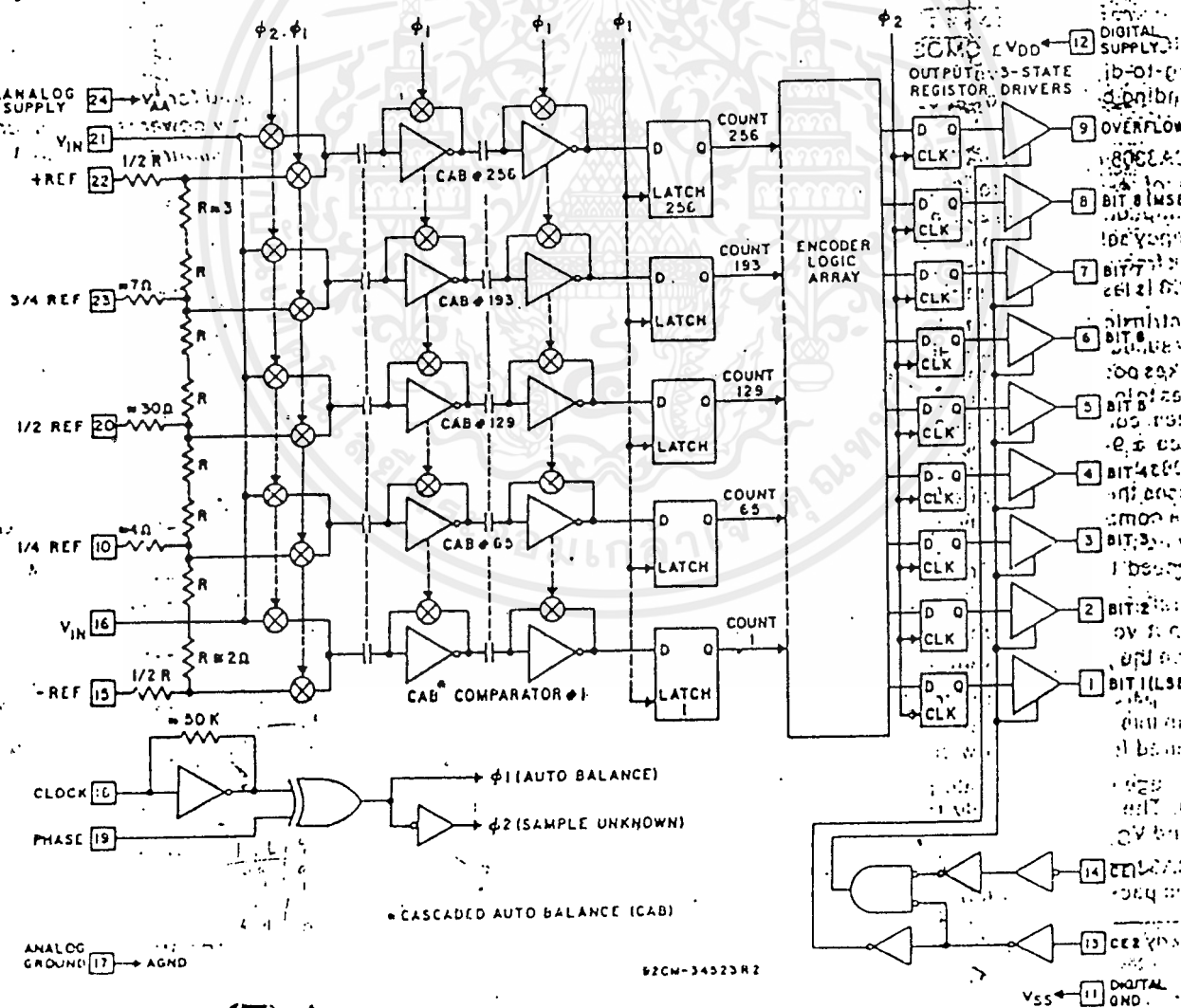
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CA3308

Solid State Division

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY VOLTAGE RANGE (V_{DD} AND V_{AA}) (VOLTAGE REFERENCED TO V_{SS} TERMINAL)	−0.5 to +8 V
INPUT VOLTAGE RANGE ALL INPUTS	−0.5 to V_{DD} +0.5 V
DC INPUT CURRENT CLK, PH, CE1, CE2, V_{IN}	±10 mA
POWER DISSIPATION PER PACKAGE (P_D) FOR T_A=−40 to 55°C	315 mW
FOR T_A=55°C to 85°C	Derate linearly at 3.3 mW/°C
TEMPERATURE RANGE OPERATING	−40 to +85°C
STORAGE	−65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING) AT DISTANCE 1/16 ± 1/32 in. (1.59 ± 0.79 mm) FROM CASE FOR 10 s MAX.	+265°C



ANA-DIGIT CO., LTD.
บริษัท แอนนาดิจิท จำกัด
โทร. 2217040, 2214592, 2214557
FAX: 225-9232

เอกสารนี้เผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{$R-,I-}
Program Image_Processing;
{$F+,O+}
Uses
  Overlay,
  Crt,
  Dos,
  Turbo3,
  Graph3,
  Printer,
  Graph,
  Image5;

  {$O Image5}

const MaxGrayLevel = 255;
      XMaxElement  = 256;
      YMaxElement  = 256;
      ArraySize    = 50;
type  vector = array[1..ArraySize] of real;
      matrix = array[1..ArraySize] of vector;
      TemDataPointerB = ^TemDataPtr;
      TemDataPtr      = array[0..32767] of byte;
      DataPointerB  = ^DataZTB;
      DataXPointerB = ^DataTypeB;
      DataZTB       = record
                        DataY : array [1..YMaxElement] of DataXPointerB;
                      end;
      DataTypeB     = record
                        DataX : array [1..XMaxElement] of byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    end;
    DataPointerI  = ^DataZTI;
    DataXPointerI = ^DataTypeI;
    DataZTI       = record
                      DataY : array [1..YMaxElement] of DataXPointerI;
                    end;
    DataTypeI     = record
                      DataX : array [1..XMaxElement] of Integer;
                    end;
    RegType       = record case integer of
                      0 : (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Word);
                      1 : (AL,AH,BL,BH,CL,CH,DL,DH          : byte);
                    end;

var Regs                    : RegType;
    DataFile                : File;
    TemData1,TemData2       : TemDataPointerB;
    FileName                : String[40];
    DataIn,DataOut,
    Direction,SubData       : DataPointerB;
    IntOut                  : DataPointerI;
    MaxElementX,MaxElementY,
    xmax,ymax,x1,y1         : integer;
    AxGlb,AyGlb,BxGlb,ByGlb : real;
    ch,Choose,disp          : char;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{$I Image.inc}


procedure Digitize;


var Minimum                : Boolean;
    x,y,i,min,max,ys,Us : integer;
    num                    : real;
    nk                     : array [0..MaxGrayLevel] of real;
    nk1                    : array [0..MaxGrayLevel] of integer;


begin
    Port[$300]:=$00; {02}
    gotoXY(1,25); write('Digitize ');
    ch := readkey;
    Port[$300]:=$02;  {80}
    Tone(2000,300);
    ClearData(DataIn);
    FillChar(nk1,SizeOf(nk1),0);
    Port[$300]:=$03;  {61}
    for Y := 1 to YMaxEleMent do
      Move(mem[$D000:((Y-1)*XMaxElement)],DataIn^.
                        DataY[Y],XMaxElement);
    Port[$300]:=$02;   {80}
    if MaxElementY>128 then ys := 128
                       else ys := MaxElementY;
    for Y := 1 to ys do
    for X := 1 to MaxElementX do
    nk1[DataIn^.DataY[Y]^.DataX[X]] :=
        nk1[DataIn^.DataY[Y]^.DataX[X]]+1;
    for i := 0 to MaxGrayLevel do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if nk1[i]>=0 then nk[i] := nk1[i]
          else  nk[i] :  65536.0 + nk1[i];
   if MaxElementY>128 then
    begin
    FillChar(nk1,SizeOf(nk1),0);
     for Y := ys+1 to MaxElementY do
      for X := 1 to MaxElementX do
        nk1[DataIn^.DataY[Y]^.DataX[X]] :=
            nk1[DataIn^.DataY[Y]^.DataX[X]]+1;

     for i := 0 to MaxGrayLevel do
    if nk1[i]>= 0 then nk[i] := nk[i] + nk1[i]
                  else nk[i] := nk[i] + 65536.0 + nk1[i];
end;
Tone(2000,300);
HiRes;
max := 0;
min := 255;
num := 0;
for i := 0 to MaxGrayLevel do
  if nk[i]>num then num := nk[i];
  Minimum := true;
  for i := 0 to  MaxGrayLevel do
  begin
    if Minimum then
       if nk[i]>0 then
       begin
         Minimum := false;
         min := i;
       end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if nk[i]>0 then max := i;
        Draw(60+i*2,185,60+i*2,Round(185-(nk[i]/num)*183),1);
    end;
    gotoXY(9,25);
    write('Minimum Gray Level = ',min:3);
    gotoXY(48,25);
    write('Maximum Gray Level = ',max:3);
    WaitForReadKey(ch);
    TextMode(3);
  end;

begin {Main}
  TextBackGround(1);
  Monitor;
  Initialize;
  OvrInit('rv1.ovr');
  repeat
   ClrScr;
   TextBackGround(1);
   TextMode(3);
   gotoXY(24,5);  write('Main Menu');
   gotoXY(22,8);  write('1) Neighborhood Averaging');
   gotoXY(22,9);  write('2) Roberts Gradient');
   gotoXY(22,10); write('3) Sobel Operators');
   gotoXY(22,11); write('4) Laplacian Operator');
   gotoXY(22,13); write('6) Histogram Equalization');
   gotoXY(22,14); write('7) Scene Analysis Using Regions');
   gotoXY(22,17); write('A) Digitize');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gotoXY(22,18); write('B) Display');
gotoXY(22,19); write('Q) Exit');
gotoXY(22,22); write('Select ');
Tone(2000,200);
repeat
   WaitForReadKey(ch);
   Choose := ch;
 until Choose in ['1'..'7','A','B','Q'];
 clrscr;
 if choose in ['1'..'7'] then
 begin
   GetCondition;
   case choose of
    '1': Neighborhood_Averaging;
    '2': Roberts_Gradient;
    '3': Sobel_Operators;
    '4': Laplacian_Operator;
    '5': Histogram_Equalization;
    '6': Scene_Analysis_Using_Regions;
   end;
   Tone(2000,300);
 end;
 if choose='A' then
 begin
   TextMode(3);
   GetSubCondition(MaxElementX,MaxElementY);
   Digitize;
   GraphicsVga;
   DisplayScreen(DataIn,MaxElementX,MaxElementY);
 end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if choose='B' then
    begin
      ch := ' ';
      while ch<>'Q' do
      begin
        clrscr;
        TextBackGround(1);
        TextMode(3);
        gotoXY(1,25);
         write('Uses [O]:Old , [N]:New , [L]:Load
                   Data for Display  or [Q]:Exit');
        repeat
          WaitForReadKey(ch);
        until Ch in ['O','N','L','Q'];
        case ch of
         'O' : DisplayScreen(DataIn,MaxElementX,MaxElementY);
         'N' : DisplayScreen(DataOut,MaxElementX,MaxElementY);
         'L' : begin
                 TextMode(3);
                 GetSubCondition(MaxElementX,MaxElementY);
                 ReadDataPicture(DataIn,MaxElementX,MaxElementY,true);
               end;
        end;
      end;
    end;
  until choose='Q';
end. {Main}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Unit IMAGE5;
{$O+,F+}
{$I+}
{$R+}
Interface
Uses  Crt,
      Dos,
      Graph,
      Printer,
      Graph3;

const MaxGrayLevel = 255;
      XMaxElement  = 256;
      YMaxElement  = 256;
      ArraySize    = 50;    {Size of the matrix}
type  vector = array[1..ArraySize] of real;
      matrix = array[1..ArraySize] of vector;
      TemDataPointerB = ^TemDataPtr;
      TemDataPtr      = array[0..32767] of byte;
      DataPointerB  = ^DataZTB;
      DataXPointerB = ^DataTypeB;
      DataZTB       = record
                        DataY : array [1..YMaxElement] of DataXPointerB;
                      end;
      DataTypeB     = record
                        DataX : array [1..XMaxElement] of byte;
                      end;
      DataPointerI  = ^DataZTI;
      DataXPointerI = ^DataTypeI;
      DataZTI       = record
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    DataY : array [1..YMaxElement] of DataXPointerI;
                  end;
  DataTypeI    = record
                   DataX : array [1..XMaxElement] of Integer;
                 end;
  RegType      = record case integer of
                   0 : (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Word);
                   1 : (AL,AH,BL,BH,CL,CH,DL,DH          : byte);
                 end;

var  Regs                       : RegType;
     DataFile                   : File;
     TemData1,TemData2          : TemDataPointerB;
     FileName                   : String[40];
     DataIn,DataOut,
     Direction,SubData          : DataPointerB;
     IntOut                     : DataPointerI;
     MaxElementX,MaxElementY,
     xmax,ymax,x1,y1            : integer;
     AxGlb,AyGlb,BxGlb,ByGlb    : real;
     ch,Choose,disp             : char;

Procedure Scene_Analysis_Using_Regions;

Procedure Wait_Message;

Procedure DisplayScreen(Data          :  DataPointerB;
                        MaxElementX   :  Integer;
                        MaxElementY   :  Integer);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure MoveData(     Data1 : DataPointerB;
                    var  Data2 : DataPointerB);

Procedure WaitForReadKey(var  ch : char);

{================================================================}

Implementation
{$I Image.inc}

procedure Scene_Analysis_Using_Regions;



label exit2;
type  k = array[-1..1,-1..1] of integer;
const Neighborhood2 : k = (( 1, 2, 1),( 2, 4, 2),( 1, 2, 1));
var   Ri  : array [0..1000] of integer;
      s                    : real;
      num,sum,min,max,a,c,
      ex,ey,r1,r2,X,Y,i,j  : integer;
      Di,Di1,Di2,Di3,g     : byte;
      start                : boolean;


 procedure SubProcess(m: integer);
 var X,Y,Xs,a : integer;

 begin
   TextMode(3);
   Wait_Message;
   di3 := 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
num := 0;
if m=0 then Xs := 2
       else Xs := 4;
repeat
 di2 := 0;
 di3 := di3+1;
 for Y := 3 to MaxElementY-3 do
 begin
  gotoxy(20,25); write(Y:3);
  for X := Xs to MaxElementX-4 do
  begin
   if (not odd(x)) and odd(y) then
    if DataOut^.DataY[Y]^.DataX[X]=255 then
    begin
     r1 := 0;
     r2 := 0;
     i := x;
     j := y;
     di := 1;
     di1 := 0;
     max := 0;
     repeat
      Fillchar(Ri,SizeOf(Ri),0);
      c := 1;
      sum := 0;
      repeat
       sum := sum+1;
       Ri[c] := Ri[c]+1;
       if di1=1 then
        if (c=r1) or (c=r2) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case di of
       1:begin
          DataOut^.DataY[j]^.DataX[i] := 0;
          DataOut^.DataY[j+1]^.DataX[i] := 0;
         end;
       2:begin
          DataOut^.DataY[j]^.DataX[i] := 0;
          DataOut^.DataY[j]^.DataX[i-1] := 0;
         end;
       3:begin
          DataOut^.DataY[j]^.DataX[i] := 0;
          DataOut^.DataY[j-1]^.DataX[i] := 0;
         end;
       4:begin
          DataOut^.DataY[j]^.DataX[i] := 0;
          DataOut^.DataY[j]^.DataX[i+1] := 0;
         end;
        end;
       case di of
   1: begin
       if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
       begin
         if (DataOut^.DataY[j+2]^.DataX[i]=255) or
            (DataOut^.DataY[j+1]^.DataX[i-1]=255) then c:= c+1;
              di := 4;
              i := i+1;
              j := j+1;
            end
         else
         if (DataOut^.DataY[j+2]^.DataX[i]=255) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      begin
       if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then c := c+1;
       j := j+2;
      end
      else
        if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
         begin
          di := 2;
          i := i-1;
          j := j+1;
         end;
    end;
  2: begin
     if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
     begin
      if (DataOut^.DataY[j]^.DataX[i-2]=255) or
       (DataOut^.DataY[j-1]^.DataX[i-1]=255) then c := c+1;
      di := 1;
      i := i-1;
      j := j+1;
     end
    else
    if (DataOut^.DataY[j]^.DataX[i-2]=255) then
    begin
     if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then c := c+1;
      i := i-2;
       end
      else
       if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          di := 3;
          i := i-1;
          j := j-1;
        end;
    end;
  3: begin
    if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
    begin
    if (DataOut^.DataY[j-2]^.DataX[i]=255) or
       (DataOut^.DataY[j-1]^.DataX[i+1]=255) then c := c+1;
      di := 2;
      i := i-1;
      j := j-1;
     end
   else
   if (DataOut^.DataY[j-2]^.DataX[i]=255) then
   begin
   if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then c := c+1;
        j := j-2;
      end
     else
      if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
       begin
        di := 4;
        i := i+1;
        j := j-1;
       end;
     end;
  4: begin
      if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
 if (DataOut^.DataY[j]^.DataX[i+2]=255) or
 (DataOut^.DataY[j+1]^.DataX[i+1]=255) then c := c+1;
 di := 3;
  i := i+1;
  j := j-1;
end
 else
 if (DataOut^.DataY[j]^.DataX[i+2]=255) then
  begin
   if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then c := c+1;
   i := i+2;
  end
 else
      if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
       begin
        di := 1;
        i := i+1;
        j := j+1;
       end;
  end;
 end;
until ((x=i) and (y=j)) or (sum=max);
if di1=0 then
 begin
 if m=0 then
  begin
   case di3 of
    1: begin
         s := 0.53;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          di2 := 1;
         end;
       2: begin
          s := 0.5;
          di2 := 1;
         end;
     else
      s := 0.47;
     end;
     for a := 2 to c-1 do
      if Ri[a]/sum>=s then
       begin
        r1 := a;
        max := sum;
        di1 := 1;
        di2 := 1;
       end;
     if (Ri[1]+Ri[c])/sum>=s then
      begin
       r1 := 1;
       r2 := c;
       max := sum;
       di1 := 1;
       di2 := 1;
      end;
   end
  else
   begin
   if num<sum then num := sum;
   if sum<m then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            begin
             max := Ri[1]+Ri[c];
             r1 := 1;
             r2 := c;
             for a := 2 to c-1 do
             if max<Ri[a] then
              begin
               r1 := a;
               r2 := 0;
              end;
             max := sum
             di1 := 1;
             di2 := 1;
            end;
           end;
         end
        else
         di1 := 0;
       until di1=0;
      end;
    end;
   end;
  until di2=0;

 for Y := 1 to MaxElementY do
 begin
  gotoxy(20,25); write(Y:3);
  for X := 1 to MaxElementX do
  begin
   if m=0 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if DataOut^.DataY[Y]^.DataX[X]<>255 then
       DataOut^.DataY[Y]^.DataX[X] := 0;
    if (not odd(x)) and (not odd(y)) then
     if DataOut^.DataY[Y]^.DataX[X]<>255 then
      if (X<>MaxElementX) and (Y<>MaxElementY) then
       if (DataOut^.DataY[Y-1]^.DataX[X]=255) or
          (DataOut^.DataY[Y]^.DataX[X-1]=255) or
          (DataOut^.DataY[Y]^.DataX[X+1]=255) or
          (DataOut^.DataY[Y+1]^.DataX[X]=255) then
          DataOut^.DataY[Y]^.DataX[X] := 255;
   end;
  end;
 end;

begin
  Wait_Message;
  New(SubData);
    for Y := 1 to MaxElementY do New(SubData^.DataY[Y]);

  for Y := 1 to MaxElementY do
    Fillchar(SubData^.DataY[Y]^,MaxElementX,127);
  max := 0;
  min := 255;
  for Y := 3 to MaxElementY-3 do
  begin
   gotoxy(20,25); write(Y:3);                          (*-1-*)
   for X := 3 to MaxElementX-3 do
    if odd(x) and odd(y) then
    begin
      sum:=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for j := -1 to 1 do
     for i := -1 to 1 do
     sum := sum + DataIn^.DataY[Y+j]^.DataX[X+i]*Neighborhood2[j,i];
     IntOut^.DataY[Y]^.DataX[X] := Round(sum/16);
     if IntOut^.DataY[Y]^.DataX[X]>max then
       max := IntOut^.DataY[Y]^.DataX[X];
     if IntOut^.DataY[Y]^.DataX[X]<min then
       min := IntOut^.DataY[Y]^.DataX[X];
  end;
end;
s := 15/(max-min);
for Y := 3 to MaxElementY-3 do
begin
  gotoxy(20,25); write(Y:3);                    (*-2-*)
  for X := 3 to MaxElementX-3 do
    if odd(x) and odd(y) then
      SubData^.DataY[Y]^.DataX[X] :=
      round((IntOut^.DataY[Y]^.DataX[X]-min)*s);
end;

ClearData(DataOut);
for Y := 1 to MaxElementY-3 do
begin
  gotoxy(20,25); write(Y:3);                    (*-3-*)
  for X := 1 to MaxElementX-3 do
    if odd(x) and odd(y) then
    begin
      if SubData^.DataY[Y]^.DataX[X]<>
         SubData^.DataY[Y]^.DataX[X+2] then
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DataOut^.DataY[Y-1]^.DataX[X+1] := 255;
            DataOut^.Dat Y[Y]^.DataX[X+1] := 255;
            DataOut^.DataY[Y+1]^.DataX[X+1] := 255;
          end;
        if SubData^.DataY[Y]^.DataX[X]<>
           SubData^.DataY[Y+2]^.DataX[X] then
          begin
            DataOut^.DataY[Y+1]^.DataX[X-1] := 255;
            DataOut^.DataY[Y+1]^.DataX[X] := 255;
            DataOut^.DataY[Y+1]^.DataX[X+1] := 255;
          end;
      end;
  end;
  DisplayScreen(DataOut,MaxElementX,MaxElementY);

  Wait_Message;
  for Y := 3 to MaxElementY-3 do
  begin
    gotoxy(20,25); write(Y:3);                          (*-4-*)
    for X := 3 to MaxElementX-3 do
      if odd(x) and odd(y) then
      begin
        for j := -1 to 1 do
         for i := -1 to 1 do
           if  (i<>0) or (j<>0) then
             if DataOut^.DataY[Y+j]^.DataX[X+i]=255 then
               SubData^.DataY[Y+j]^.DataX[X+i] := 255
             else
               SubData^.DataY[Y+j]^.DataX[X+i] :=
               SubData^.DataY[Y]^.DataX[X];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
  end;


  Wait_Message;
  MoveData(SubData,DataOut);
  for Y := 3 to MaxElementY-3 do
  begin
   gotoxy(20,25); write(Y:3);                          (*-5-*)
   for X := 2 to MaxElementX-4 do
   begin
    if (not odd(x)) and odd(y) then
     if DataOut^.DataY[Y]^.DataX[X]=255 then
      if (abs(DataOut^.DataY[Y]^.DataX[X-1]-DataOut^.DataY[Y]^.
          DataX[X+1])<=1) then
      begin
       i := x;
       j := y;
       g := DataOut^.DataY[Y]^.DataX[X-1];
       di := 1;
       di1 := 0;
       repeat
        repeat
         case di of
       1: begin
            if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
             begin
              if (DataOut^.DataY[j+2]^.DataX[i+1]=g) then
               begin
                di := 4;
                i := i+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        j := j+1;
       end
      else
       goto exit2;
     end
    else
     if (DataOut^.DataY[j+2]^.DataX[i]=255) then
      begin
       if (DataOut^.DataY[j+2]^.DataX[i-1]=g) then
        j := j+2
       else
        goto exit2;
      end
     else
      if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
      begin
       if (DataOut^.DataY[j]^.DataX[i-1]=g) then
        begin
         di := 2;
         i := i-1;
         j := j+1;
        end
       else
        goto exit2;
      end;
   end;
2: begin
    if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
     begin
      if (DataOut^.DataY[j+1]^.DataX[i-2]=g) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          j := j+1;
         end
        else
         goto exit2;
       end
      else
       if (DataOut^.DataY[j+2]^.DataX[i]=255) then
        begin
         if (DataOut^.DataY[j+2]^.DataX[i-1]=g) then
          j := j+2
         else
          goto exit2;
        end
       else
        if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
        begin
         if (DataOut^.DataY[j]^.DataX[i-1]=g) then
          begin
           di := 2;
           i := i-1;
           j := j+1;
          end
         else
          goto exit2;
         end;
    end;
2: begin
     if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
      begin
       if (DataOut^.DataY[j+1]^.DataX[i-2]=g) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          begin
           di := 1;
           i := i-1;
           j := j+1;
          end
         else
          goto exit2;
        end
       else
        if (DataOut^.DataY[j]^.DataX[i-2]=255) then
         begin
          if (DataOut^.DataY[j-1]^.DataX[i-2]=g) then
           i := i-2
          else
           goto exit2;
         end
        else
        if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
         begin
          if (DataOut^.DataY[j-1]^.DataX[i]=g) then
           begin
            di := 3;
            i := i-1;
            j := j-1;
           end
          else
           goto exit2;
         end;
      end;
  3: begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
 begin
  if (DataOut^.DataY[j-2]^.DataX[i-1]=g) then
   begin
    di := 2;
    i := i-1;
    j := j-1;
   end
  else
   goto exit2;
 end
else
 if (DataOut^.DataY[j-2]^.DataX[i]=255) then
  begin
   if (DataOut^.DataY[j-2]^.DataX[i+1]=g) then
    j := j-2
   else
    goto exit2;
  end
 else
  if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
   begin
    if (DataOut^.DataY[j]^.DataX[i+1]=g) then
     begin
      di := 4;
      i := i+1;
      j := j-1;
     end
    else
     goto exit2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;
    end;
4:  begin
     if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
      begin
       if (DataOut^.DataY[j-1]^.DataX[i+2]=g) then
        begin
         di := 3;
         i := i+1;
         j := j-1;
        end
       else
        goto exit2;
      end
     else
      if (DataOut^.DataY[j]^.DataX[i+2]=255) then
      begin
       if (DataOut^.DataY[j+1]^.DataX[i+2]=g) then
        i := i+2
       else
        goto exit2;
      end
     else
      if (DataQut^.DataY[j+1]^.DataX[i+1]=255) then
       begin
        if (DataOut^.DataY[j+1]^.DataX[i]=g) then
         begin
          di := 1;
          i := i+1;
          j := j+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            end
           else
            goto exit2;
          end;
       end;
     end;
     if di1=1 then
      case di of
     1:begin
        DataOut^.DataY[j]^.DataX[i] := g;
        DataOut^.DataY[j+1]^.DataX[i] := g;
       end;
     2:begin
        DataOut^.DataY[j]^.DataX[i] := g;
        DataOut^.DataY[j]^.DataX[i-1] := g;
       end;
     3:begin
        DataOut^.DataY[j]^.DataX[i] := g;
        DataOut^.DataY[j-1]^.DataX[i] := g;
       end;
     4:begin
        DataOut^.DataY[j]^.DataX[i] := g;
        DataOut^.DataY[j]^.DataX[i+1] := g;
       end;
      end;
    until (x=i) and (y=j);
    di1 := di1+1;
   until di1=2;
  end;
exit2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 end;
end;

for Y := 4 to MaxElementY-4 do
begin
 gotoxy(20,25); write(Y:3);                              (*-6-*)
 for X := 4 to MaxElementX-4 do
  if (not odd(x)) and (not odd(y)) then
   if DataOut^.DataY[Y]^.DataX[X]<>255 then
    if (DataOut^.DataY[Y-1]^.DataX[X]=255) or
       (DataOut^.DataY[Y]^.DataX[X-1]=255) or
       (DataOut^.DataY[Y]^.DataX[X+1]=255) or
       (DataOut^.DataY[Y+1]^.DataX[X]=255) then
        DataOut^.DataY[Y]^.DataX[X] := 255;
end;
DisplayScreen(DataOut,MaxElementX,MaxElementY);

SubProcess(0);                                           (*-7-*)
DisplayScreen(DataOut,MaxElementX,MaxElementY);

Wait_Message;
for Y := 3 to MaxElementY-3 do
begin
 gotoxy(20,25); write(Y:3);                              (*-8-*)
 for X := 3 to MaxElementX-3 do
 begin
  if (X=3) or (X=MaxElementX-3) or
     (Y=3) or (Y=MaxElementX-3) then
   if DataOut^.DataY[Y]^.DataX[X]=255 then
   begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
i := x;
j := y;
di1 := 0;
if X=3 then di := 4
else if X=MaxElementX-3 then di := 2
     else if Y=3 then di := 1
          else if Y=MaxElementX-3 then di := 3;
repeat
 case di of
1: begin
   DataOut^.DataY[j]^.DataX[i] := 0;
   DataOut^.DataY[j+1]^.DataX[i] := 0;
   if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
   begin
    if (DataOut^.DataY[j+2]^.DataX[i]=255) or
       (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
        di1 := di1+1;
    di := 4;
     i := i+1;
     j := j+1;
     end
    else
     if (DataOut^.DataY[j+2]^.DataX[i]=255) then
      begin
       if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
          di1 := di1+1;
       j := j+2;
      end
     else
      if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
         di := 2;
         i := i-1;
         j := j+1;
        end
       else
        di1 := di1+2;
   end;
2: begin
    DataOut^.DataY[j]^.DataX[i] := 0;
    DataOut^.DataY[j]^.DataX[i-1] := 0;
    if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
     begin
      if (DataOut^.DataY[j]^.DataX[i-2]=255) or
        (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
          di1 := di1+1;
      di := 1;
      i := i-1;
      j := j+1;
     end
    else
     if (DataOut^.DataY[j]^.DataX[i-2]=255) then
      begin
       if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
          di1 := di1+1;
       i := i-2;
      end
     else
      if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
       begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            di := 3;
            i := i-1;
            j := j-1;
           end
          else
           di1 := di1+2;
      end;
  3: begin
      DataOut^.DataY[j]^.DataX[i] := 0;
      DataOut^.DataY[j-1]^.DataX[i] := 0;
      if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
       begin
        if (DataOut^.DataY[j-2]^.DataX[i]=255) or
           (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
            di1 := di1+1;
        di := 2;
        i := i-1;
        j := j-1;
       end
      else
       if (DataOut^.DataY[j-2]^.DataX[i]=255) then
        begin
         if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
             di1 := di1+1;
         j := j-2;
        end
       else
        if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
         begin
          di := 4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          i := i+1;
          j := j-1;
        end
       else
        di1 := di1+2;
    end;
4: begin
    DataOut^.DataY[j]^.DataX[i] := 0;
    DataOut^.DataY[j]^.DataX[i+1] := 0;
    if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
     begin
      if (DataOut^.DataY[j]^.DataX[i+2]=255) or
         (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
          di1 := di1+1;
      di := 3;
      i := i+1;
      j := j-1;
     end
    else
     if (DataOut^.DataY[j]^.DataX[i+2]=255) then
      begin
       if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
           di1 := di1+1;
       i := i+2;
      end
     else
      if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
       begin
        di := 1;
        i := i+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            j := j+1;
           end
          else
           di1 := di1+2;
       end;
     end;
    until di1>0;
   end;
 end;
end;

SubProcess(50);                                    (*-9-*)
DisplayScreen(DataOut,MaxElementX,MaxElementY);

HiRes;
Wait_Message;
DefineWorld(0,255,255,0);
for Y := 3 to MaxElementY-3 do
begin
 gotoxy(20,25); write(Y:3);                        (*-10-*)
 for X := 3 to MaxElementX-3 do
  if (not odd(x)) and odd(y) then
   if DataOut^.DataY[Y]^.DataX[X]=255 then
   begin
     i := x;
     j := y;
     di := 1;
     di1 := 0;
     di2 := 0;
     c := 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sum := 0;
ex := 0;
ey := 0;
r1 := 0;
r2 := 0;
Fillchar(Ri,SizeOf(Ri),0);
start := true;
repeat
 case di of
1: begin
   if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
    begin
    if (DataOut^.DataY[j+2]^.DataX[i]=255) or
       (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
        di1 := di1+1;
    di := 4;
    i := i+1;
    j := j+1;
    end
   else
    if (DataOut^.DataY[j+2]^.DataX[i]=255) then
     begin
      if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
          di1 := di1+1;
      j := j+2;
     end
    else
     if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
      begin
       di := 2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          i := i-1;
          j := j+1;
        end;
    end;
2: begin
     if (DataOut^.DataY[j+1]^.DataX[i-1]=255) then
      begin
       if (DataOut^.DataY[j]^.DataX[i-2]=255) or
          (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
           di1 := di1+1;
      di := 1;
      i := i-1;
      j := j+1;
     end
    else
     if (DataOut^.DataY[j]^.DataX[i-2]=255) then
      begin
       if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
          di1 := di1+1;
       i := i-2;
      end
     else
      if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
       begin
        di := 3;
        i := i-1;
        j := j-1;
       end;
    end;
3: begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (DataOut^.DataY[j-1]^.DataX[i-1]=255) then
     begin
      if (DataOut^.DataY[j-2]^.DataX[i]=255) or
         (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
          di1 := di1+1;
      di := 2;
      i := i-1;
      j := j-1;
     end
    else
     if (DataOut^.DataY[j-2]^.DataX[i]=255) then
      begin
       if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
           di1 := di1+1;
       j := j-2;
      end
     else
      if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
       begin
        di := 4;
        i := i+1;
        j := j-1;
       end;
   end;
4: begin
    if (DataOut^.DataY[j-1]^.DataX[i+1]=255) then
     begin
      if (DataOut^.DataY[j]^.DataX[i+2]=255) or
         (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
          di1 := di1+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      di := 3;
      i := i+1;
      j := j-1;
     end
    else
     if (DataOut^.DataY[j]^.DataX[i+2]=255) then
      begin
       if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
           di1 := di1+1;
       i := i+2;
      end
     else
      if (DataOut^.DataY[j+1]^.DataX[i+1]=255) then
       begin
        di := 1;
        i := i+1;
        j := j+1;
       end;
  end;
end;
if (ex=i) and (ey=j) then
begin
 if sum>num-50 then di2 := di2+2;
 if (di1=4) and (di2=1) then
 begin
  for a := 1 to di1-1 do
  if Ri[a]/sum>=0.45 then
  begin
   r1 := a;
   case r1 of
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     1: r2 := round(Ri[a]*Ri[2]/(Ri[2]+Ri[3]));
     2: r2 := round(Ri[a]*Ri[3]/(Ri[3]+Ri[1]));
     3: r2 := round(Ri[a]*Ri[1]/(Ri[1]+Ri[2]));
    end;
   end;
  end;
  di2 := di2+1;
  di1 := 1;
  sum := 0;
  Fillchar(Ri,SizeOf(Ri),0);
 end;
 if (di1=1) and (di2=0) then
 begin
  ex := i;
  ey := j;
  sum := 0;
  Fillchar(Ri,SizeOf(Ri),0);
  di2 := di2+1;
 end;
 sum := sum+1;
 Ri[di1] := Ri[di1]+1;
 if di2>1 then
  if (di1<>c) or ((di1=r1) and (Ri[di1]=r2)) then
  begin
   if di1<>c then c := di1;
   case di of
    1: if start then pset(i,j-1)
                else line(i,j-1);
    2: if start then pset(i+1,j)
                else line(i+1,j);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          3: if start then pset(i,j+1)
                      else line(i,j+1);
          4: if start then pset(i-1,j)
                      else line(i-1,j);
         end;
         start := false;
        end;
      until di2>2;
    end;
  end;
  gotoXY(1,25); ClrEol;
  Tone(2000,300);
  WaitForReadKey(ch);
  textmode(3);
{! 17. The^ Textmode procedure requires a parameter (Mode:Word) in Turbo 5.0.

  for Y := MaxElementY Downto 1 do DisPose(SubData^.DataY[Y]);
  DisPose(SubData);
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure Tone(Note,Time : Integer);
{
begin
  Sound(Note);
  Delay(Time);
  NoSound;
end;
};

procedure WaitForReadKey(var Ch : char);
{
begin
  ch := ReadKey;
  ch:=Upcase(ch);
  Tone(3000,30);
end;
};
procedure graphicsVGA;
{
 var GrDriver,GrMode : Integer;
 begin
   GrDriver := Detect;
   InitGraph(GrDriver,GrMode,' ');
 end;
};
procedure BarGraph;
{
 begin
   SetFillStyle(7,1);
   Bar(0,350,640,480);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  end;
{
}
procedure DefineWorld(X1W,Y2W,X2W,Y1W: integer);
{
var X_1,Y_1,X_2,Y_2  : integer;
begin
  X_1 := 0;
  Y_1 := 0;
  X_2 := 639;
  Y_2 := 199;
  BxGlb := (X_2-X_1)/(X2W-X1W);
  ByGlb := (Y_2-Y_1)/(Y2W-Y1W);
  AxGlb := X_1-X1W*BxGlb;
  AyGlb := Y_1-Y1W*ByGlb;
end;
{
}
procedure pset(x,y: integer);
{
begin
  x1 := Round(AxGlb+BxGlb*x);
  y1 := Round(AyGlb+ByGlb*y);
end;
{
}
procedure Line(x,y: integer);
{
var x2,y2 : integer;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    x2 := Round(AxGlb+BxGlb*x);
    y2 := Round(AyGlb+ByGlb*y);
    Draw(x1,y1,x2,y2,1);
    x1 := x2;,
    y1 := y2;
  end;
  ;

  procedure DrawLine(x1,y1,x2,y2: integer);
  ;
  begin
    pset(x1,y1);
    Line(x2,y2);
  end;
  ;
  ;
  procedure TextClrEol;
  ;
  begin
    ClrEol;
  end;
  ;

  procedure ClrEol;
  ;
  var i    : byte;
  begin
  ;
    regs.ah := 15;
    intr($10,Dos.Registers(regs));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if regs.al in [4..6] then
    begin
     regs.ah := 10;
     regs.bh := 0;
     regs.cx := 80;
     regs.al := $20;
     regs.bl := $70;
     intr($10,Dos.Registers(regs));
    end
   else TextClrEol;
end;
;
:
procedure Wait_Message;
{
begin
  gotoXY(1,25); ClrEol;
  write('  Please Wait ');
end;
;
}
procedure Monitor;
{
Begin
  Clrscr;
  TextBackGround(1);
  gotoXY(22,13); ClrEol;
  writeln('Uses   [1]:Analog Monitor ');
  gotoXY(22,15); ClrEol;r
  writeln('       [2]:Color/Graphics Monitor');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  repeat
   WaitForReadKey(disp);
  until (disp='1') or (disp='2');
end;
;

procedure Initialize;
{
var Y : integer;
begin
  New(DataIn);
  for Y := 1 to YMaxElement do New(DataIn^.DataY[Y]);
  New(DataOut);
  for Y := 1 to YMaxElement do New(DataOut^.DataY[Y]);
  New(IntOut);
  for Y := 1 to YMaxElement do New(IntOut^.DataY[Y]);
{
end;
;

procedure ClearData(var Data : DataPointerB);
{
var Y  : integer;
begin
  for Y := 1 to YMaxElement do
    Fillchar(Data^.DataY[Y]^,XMaxElement,0);
end;
;
}
procedure MoveData(    Data1 : DataPointerB;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    var Data2 : DataPointerB);

var Y  : integer;

begin
  for Y := 1 to YMaxElement do
    Data2^.DataY[Y]^ := Data1^.DataY[Y]^;
end;

;

;

procedure SaveDataPicture(DataSave    : DataPointerB;
                          MaxElementX : integer;
                          MaxElementY : integer);


var  CreateFile,ok     : boolean;
     ch1               : char;
     x,y,y2,Rec1,Rec2  : integer;
     i,d               : real;
 begin
 ch1:= ' ';
  repeat
   FileName[1] := ' ';
    repeat
     ClrEOL ;
     BarGraph;
     OuttextXY(30,400,' Enter Filename ' );
     read(filename);F
     Tone(2000,50);
   until FileName[1]<>' ';
   Assign(DataFile,FileName);
   {$I-} Reset(DataFile); {$I+}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  CreateFile := IOresult = 0;
  if NOT CreateFile then
    begin
      rewrite(DataFile);
      ok := true;
    end
  else
    begin
      BarGraph;
      OuttextXY(30,400,'Overwrite old FileName  (Y/N) ');
      repeat
        WaitForReadKey(ch1);
      until (ch1='Y') or (ch1='N');
      if ch1='Y' then
        begin
          rewrite(DataFile);
          ok := true;
        end
      else
        ok := False;
    end;
until ok;
if FileName[2]=':' then
  regs.dl := Ord(Upcase(FileName[1]))-$40
else
  regs.dl := 0;
regs.ah := $36;
MsDos(Dos.Registers(regs));
d := 1.0*regs.bx*regs.cx*regs.ax;
i := 1.0*MaxElementX*MaxElementY;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gotoXY(1,25); ClrEOL;;
if d<i then
 begin;
  Tone(2000,200);
  BarGraph;
  OuttextXY(30,400,' Disk Full ');
  Close(DataFile);
  Erase(DataFile);
  WaitForReadKey(ch1);
 end
else
 begin
  write(' Saving  ',FileName);
  New(TemData1);
  if i>32768.0 then
   begin
     y2 := Trunc((i-32768.0) / MaxElementX);
     MaxElementY := Trunc(32768.0 / MaxElementX);
   end;
  for Y := 1 to MaxElementY do
   move(DataSave^.DataY[Y]^,TemData1^[(Y-1)*MaxElementX],MaxElementX);
  Rec1 := Trunc(1.0*MaxElementX*MaxElementY/128);;
  if Frac(1.0*MaxElementX*MaxElementY/128)>0 then Rec1 := Rec1+1;
  Blockwrite(DataFile,TemData1^[0],Rec1);
  if i>32768.0 then
   begin
     for Y := 1 to y2 do
       move(DataSave^.DataY[Y+MaxElementY]^,TemData1^[(Y-1)
       *MaxElementX],MaxElementX);
     Rec2 := Trunc(1.0*MaxElementX*y2/128);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if Frac(1.0*MaxElementX*y2/128)>0 then Rec2 := Rec2+1;
        Blockwrite(DataFile,TemData1^[0],Rec2);
      end;t
     DisPose(TemData1);
     close(DataFile);
     Tone(2000,300);
    end;
  gotoxy(1,whereY); clreol;
end;
;
}
procedure PrintScreen(DataP      : DataPointerB;
                      MaxElementX : integer;
                      MaxElementY : integer);

label exit,ex1;
var x,y,i,j,Line,row,col,
    dot,col1,Lines,bi,li,
    SubLine,min,max          : integer;
    s                        : real;
    Data                     : array [1..4] of byte;
    Data2                    : array [1..6,1..3] of byte;
    a,b                      : byte;
    ch0,ch1,ch2,ch3,ch4,ch5  : char;
    space                    : boolean;
begin
j
  BarGraph;
  OuttextXY(30,400,' Uses [A]: 8 , [B]:24   Number of Pins');
  repeat,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  WaitForReadKey(ch0);
until (ch0='A') or (ch0='B');
ch5 := ' ';
if ch0='B' then
begin
  write('Print  [A]:Small size ,  [B]:Normal Size');
  BarGraph;
   OuttextXY(30,400,' Print [A]:Small size , [B]:Normal size');
  repeat
    WaitForReadKey(ch5);
  until (ch5='A') or (ch5='B');
  if ch5='A' then ch0 := 'A';
end;
write(' Negative (Y/N) ? ');
BarGraph;
OuttextXY(30,400,' Negative (Y/N) ? ');
repeat
  WaitForReadKey(ch1);
until (ch1='Y') or (ch1='N');
BarGraph;
OuttextXY(30,400,' SideWays (Y/N) ? ');
repeat
  WaitForReadKey(ch3);
until (ch3='Y') or (ch3='N');
if ch3='Y' then'
 begin
   row := MaxElementX;
   col := MaxElementY;
 end
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    row := MaxElementY;
    col := MaxElementX;
  end;
 col1 := 1;
 if (col>240) and (ch0='A') then
  begin
   BarGraph;
   OuttextXY(30,400,' Paper width : Greater 8 inches (Y/N) ? ');
   repeat
     WaitForReadKey(ch4);
   until (ch4='Y') or (ch4='N');
   if ch4='N' then
    begin
     col1 := Round((col-240)/2);
     col := col1+240;
    end;
  end;
 max := 0;
 min := 255;
 for y := 1 to MaxElementY do
   for x := 1 to MaxElementX do
     begin
       if DataP^.DataY[Y]^.DataX[X]>max then max :=
          DataP^.DataY[Y]^.DataX[X];
       if DataP^.DataY[Y]^.DataX[X]<min then min :=
          DataP^.DataY[Y]^.DataX[X];
     end;
 Bargraph;
 OuttextXY(30,400,' Minimum Gray Level = '); writeln(min:3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OuttextXY(30,420,' Maximum Gray Level = '); Writeln(max:3);
if (min<>0) or (max<>255) then
  begin
    BarGraph;
    OuttextXY(30,400,' Scaling (Y/N) ? ');
    repeat
      WaitForReadKey(ch2);
    until (ch2='Y') or (ch2='N');
    if ch2='Y' then
    begin
     BarGraph;
     OuttextXY(30,400,' Scaling ... ');
      s := 255/(max-min);
    end;
  end
else
  ch2 := 'N';
Tone(3000,300);
BarGraph;
OuttextXY(30,400,' Press ESC to Stop Printing ');
Writeln(Lst,#27,'3',#12);
if ch0='A' then
 begin
  bi := 2;
  li := 4;
  Lines := (row shr 2)-1;
  if Frac(row/4)>0 then
   begin
    Lines := Lines+1;
    space := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end
   else
     space := false;
  end
 else
  begin
   bi := 1;
   li := 3;
   Lines := Trunc(row/3)-1;
   if Frac(row/3)>0 then
    begin
     Lines := Lines+1;
     space := true;
    end
   else
     space := false;
  end;
 for Line := 0 to Lines do
  for a := 1 to bi do
  begin
   if ch0='A' then
     Write(Lst,#27,'*',#1,Chr(Lo((col-col1+1) shl 2)),
          Chr(Hi((col-col1+1) shl 2)))
   else
     Write(Lst,#27,'*',#33,Chr(Lo(col*6)),Chr(Hi(col*6)));
   for dot := col1 to col do
   begin
     if ch0='A' then Fillchar(Data,SizeOf(Data),0)
                else Fillchar(Data2,SizeOf(Data2),0);
     for SubLine := 1 to li do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin:
      if ch0='B' then
       begin
        if ch3='Y' then
          b := DataP^.DataY[col-dot+1]^.DataX[Line*3+Subline]
        else
          b := DataP^.DataY[Line*3+Subline]^.DataX[dot];
if ch2='Y' then b := round((b-min)*s);
if ch1='N' then b := not b;
if space then
if Line=Lines then
if SubLine>Frac(row/3)*3 then b := 0;
if b>5   then Data2[3,SubLine] := Data2[3,SubLine] or $10 else goto ex1;
if b>10  then Data2[3,SubLine] := Data2[3,SubLine] or $20 else goto ex1;
if b>15  then Data2[3,SubLine] := Data2[3,SubLine] or $40 else goto ex1;
if b>20  then Data2[3,SubLine] := Data2[3,SubLine] or $80 else goto ex1;
if b>26  then Data2[4,SubLine] := Data2[4,SubLine] or $08 else goto ex1;
if b>31  then Data2[4,SubLine] := Data2[4,SubLine] or $04 else goto ex1;
if b>36  then Data2[4,SubLine] := Data2[4,SubLine] or $02 else goto ex1;
if b>41  then Data2[4,SubLine] := Data2[4,SubLine] or $01 else goto ex1;
if b>46  then Data2[2,SubLine] := Data2[2,SubLine] or $08 else goto ex1;
if b>52  then Data2[2,SubLine] := Data2[2,SubLine] or $04 else goto ex1;
if b>57  then Data2[2,SubLine] := Data2[2,SubLine] or $02 else goto ex1;
if b>62  then Data2[2,SubLine] := Data2[2,SubLine] or $01 else goto ex1;
if b>67  then Data2[5,SubLine] := Data2[5,SubLine] or $10 else goto ex1;
if b>72  then Data2[5,SubLine] := Data2[5,SubLine] or $20 else goto ex1;
if b>78  then Data2[5,SubLine] := Data2[5,SubLine] or $40 else goto ex1;
if b>83  then Data2[5,SubLine] := Data2[5,SubLine] or $80 else goto ex1;
if b>88  then Data2[1,SubLine] := Data2[1,SubLine] or $10 else goto ex1;
if b>93  then Data2[1,SubLine] := Data2[1,SubLine] or $20 else goto ex1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if b>98  then Data2[1,SubLine] := Data2[1,SubLine] or $40 else goto ex1;
if b>104 then Data2[1,SubLine] := Data2[1,SubLine] or $80 else goto ex1;
if b>109 then Data2[6,SubLine] := Data2[6,SubLine] or $08 else goto ex1;
if b>114 then Data2[6,SubLine] := Data2[6,SubLine] or $04 else goto ex1;
if b>119 then Data2[6,SubLine] := Data2[6,SubLine] or $02 else goto ex1;
if b>124 then Data2[6,SubLine] := Data2[6,SubLine] or $01 else goto ex1;
if b>130 then Data2[3,SubLine] := Data2[3,SubLine] or $08 else goto ex1;
if b>135 then Data2[3,SubLine] := Data2[3,SubLine] or $04 else goto ex1;
if b>140 then Data2[3,SubLine] := Data2[3,SubLine] or $02 else goto ex1;
if b>145 then Data2[3,SubLine] := Data2[3,SubLine] or $01 else goto ex1;
if b>150 then Data2[4,SubLine] := Data2[4,SubLine] or $10 else goto ex1;
if b>156 then Data2[4,SubLine] := Data2[4,SubLine] or $20 else goto ex1;
if b>161 then Data2[4,SubLine] := Data2[4,SubLine] or $40 else goto ex1;
if b>166 then Data2[4,SubLine] := Data2[4,SubLine] or $80 else goto ex1;
if b>171 then Data2[2,SubLine] := Data2[2,SubLine] or $10 else goto ex1;
if b>176 then Data2[2,SubLine] := Data2[2,SubLine] or $20 else goto ex1;
if b>182 then Data2[2,SubLine] := Data2[2,SubLine] or $40 else goto ex1;
if b>187 then Data2[2,SubLine] := Data2[2,SubLine] or $80 else goto ex1;
if b>192 then Data2[5,SubLine] := Data2[5,SubLine] or $08 else goto ex1;
if b>197 then Data2[5,SubLine] := Data2[5,SubLine] or $04 else goto ex1;
if b>202 then Data2[5,SubLine] := Data2[5,SubLine] or $02 else goto ex1;
if b>208 then Data2[5,SubLine] := Data2[5,SubLine] or $01 else goto ex1;
if b>213 then Data2[1,SubLine] := Data2[1,SubLine] or $08 else goto ex1;
if b>218 then Data2[1,SubLine] := Data2[1,SubLine] or $04 else goto ex1;
if b>223 then Data2[1,SubLine] := Data2[1,SubLine] or $02 else goto ex1;
if b>228 then Data2[1,SubLine] := Data2[1,SubLine] or $01 else goto ex1;
if b>234 then Data2[6,SubLine] := Data2[6,SubLine] or $10 else goto ex1;
if b>239 then Data2[6,SubLine] := Data2[6,SubLine] or $20 else goto ex1;
if b>244 then Data2[6,SubLine] := Data2[6,SubLine] or $40 else goto ex1;
if b>249 then Data2[6,SubLine] := Data2[6,SubLine] or $80;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ex1:  end;
      else
       begin
        if ch3='Y' then
         b := DataP^.DataY[col-dot+col1]^.DataX[(Line shl 2)+Subline]
        else
          b := DataP^.DataY[(Line shl 2)+Subline]^.DataX[dot];
        if ch2='Y' then b := round((b-min)*s);
        if ch1='N' then b := not b;
        if space then
          if Line=Lines then
            if SubLine>Frac(row/4)*4 then b := 0;
        for i := 1 to 4 do
          Data[i] := Data[i] shl 2;
        if a=1 then
         case b of
   31..45  : begin
               Data[1] := Data[1] or $00;  Data[2] := Data[2] or $02;
               Data[3] := Data[3] or $00;  Data[4] := Data[4] or $00;
             end;
   46..75  : begin
               Data[1] := Data[1] or $00;  Data[2] := Data[2] or $02;
               Data[3] := Data[3] or $01;  Data[4] := Data[4] or $00;
             end;
  76..120  : begin
               Data[1] := Data[1] or $01;  Data[2] := Data[2] or $02;
               Data[3] := Data[3] or $01;  Data[4] := Data[4] or $00;
             end;
 121..135 : begin
               Data[1] := Data[1] or $01;  Data[2] := Data[2] or $02;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $02;
          end;a
  136..180 : begin
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $02;
          end;a
  181..210 : begin
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $02;
          end;a
  211..225 : begin
            Data[1] := Data[1] or $03;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $02;
          end;a
  226..255 : begin
            Data[1] := Data[1] or $03;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $03;
          end;a
      end
    else
      case b of
  16..60 : begin
            Data[1] := Data[1] or $00;  Data[2] := Data[2] or $02;
            Data[3] := Data[3] or $00;  Data[4] := Data[4] or $00;
          end;
  61..90 : begin
            Data[1] := Data[1] or $00;  Data[2] := Data[2] or $02;
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $00;
          end;
  91..105 : begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $02;
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $00;
          end;
106..150 : begin
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $02;
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $02;
          end;
151..165 : begin
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $01;  Data[4] := Data[4] or $02;
          end;
166..195 : begin
            Data[1] := Data[1] or $01;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $02;
          end;
196..240 : begin
            Data[1] := Data[1] or $03;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $02;
          end;
241..255 : begin
            Data[1] := Data[1] or $03;  Data[2] := Data[2] or $03;
            Data[3] := Data[3] or $03;  Data[4] := Data[4] or $03;
          end;
      end;
    end;
  end;
  if ch0='A' then
   begin
     for i := 1 to 4 do
       Write(Lst,Chr(Data[i]));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      end;
     else
      for i := 1 to 6 do
        for j := 1 to 3 do
          Write(Lst,Chr(Data2[i,j]));
   end;
   if ch0='B' then
    Writeln(Lst,#27,'3',#24)
   else
    if ch5='A' then
     begin
      if a=1 then
       Writeln(Lst,#28,'3',#3)
      else
       Writeln(Lst,#28,'3',#45);
     end
    else
     if a=1 then
      Writeln(Lst,#27,'3',#1)
     else
      Writeln(Lst,#27,'3',#23);
   regs.ah := 6;,
   regs.dl := $ff;
   MsDos(Dos.Registers(Regs));
   if (regs.al=27) then goto exit;
  end;
 exit:
 Write(Lst,#27,'2');
{gotoxy(1,whereY); clreol;}
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure DisplayScreen(Data        : DataPointerB;
                        MaxElementX : integer;
                        MaxElementY : integer);


var x,y,MaxY  : integer;
    ch1       : char;
    GrDriver,GrMode : Integer;
begin
  case disp of
  '1': begin
       gotoXY(1,25); ClrEol;
       Port[$300] := $61;
       for Y := 1 to YMaxElement do
         move(Data^.DataY[Y]^,mem[$D000:(Y-1)*XMaxElement],XMaxElement);
       port[$300] := $80;0
      end;
  '2': begin
       GraphicsVGA;
       SetFillStyle(1,1);
       Bar(0,0,640,480);
       if MaxElementY>200 then MaxY := 250
                          else MaxY := MaxElementY;
       for y := 1 to MaxY do
        for x := 1 to MaxElementX do
          begin
            case Data^.DataY[Y]^.DataX[X] of

              0..15  : PutPixel (x+30,y+10,Black);
             16..31  : PutPixel (x+30,y+10,Blue);
             32..47  : PutPixel (x+30,y+10,Green);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                48..63  : PutPixel (x+30,y+10,Cyan);
                64..79  : PutPixel (x+30,y+10,Red);
                80..95  : PutPixel (x+30,y+10,Magenta);
               96..111  : PutPixel (x+30,y+10,Brown);
              112..127  : PutPixel (x+30,y+10,LightGray);
              128..143  : PutPixel (x+30,y+10,DarkGray);
              144..159  : PutPixel (x+30,y+10,LightBlue);
              160..175  : PutPixel (x+30,y+10,LightGreen);
              176..191  : PutPixel (x+30,y+10,LightCyan);
              192..207  : PutPixel (x+30,y+10,LightRed);
              208..223  : PutPixel (x+30,y+10,LightMagenta);
              224..239  : PutPixel (x+30,y+10,Yellow);
              240..255  : PutPixel (x+30,y+10,White);
            end;
          end;
      end;
  end;
  Tone(3000,300);
  repeat
   BarGraph;
   OutTextXY(30,400,' Save (Y/N) ? ');
   repeat,
     WaitForReadKey(ch1);
   until (ch1='Y') or (ch1='N');
   if ch1='Y' then SaveDataPicture(Data,MaxElementX,MaxElementY);
  until (ch1='N');
  repeat
    BarGraph;
    OutTextXY(30,400,' Print (Y/N) ? ');
  repeat:
```

นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      WaitForReadKey(ch1);
    until (ch1='Y') or (ch1='N');
    if ch1='Y' then PrintScreen(Data,MaxElementX,MaxElementY);
   until (ch1='N');
   TextMode(3);
end;
;
}
 procedure ReadDataPicture(var DataPic   : DataPointerB;
                               Xin       : integer;
                               Yin       : integer;
                               Histogram : Boolean);

var nk                  : array [0..MaxGrayLevel] of real;
    nk1                 : array [0..MaxGrayLevel] of integer;
    Ok,ch               : char;
    filename,a          : string[40];
    Created,Minimum     : boolean;
    tem                 : byte;
    x,y,y1,y2,min,max,
    c,k,Rec1,Rec2       : integer;
    s,i,num             : real;
begin
  Created := false;
  repeat
    gotoXY(1,5);
    ClrEol;
    write('Please enter data filename : ');
    read (filename);
    writeln;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write(' Data filename  ',filename);
readln;
Assign(DataFile,FileName);
{$I-} reset(DataFile); {$I+}
Created := IOresult = 0 ;
if Created then
  begin
   gotoXY(1,5);
    ClrEol;
    write(' Loading  ',FileName);
    Tone(2000,50);
    Dispose(DataIn);
    Dispose(DataOut);
    New(TemData1);
    New(TemData2);
    ClearData(DataPic);
    i := 1.0*Xin*Yin;
    y1 := Yin;
    if i>32768.0 then
     begin
      y2 := Trunc((i-32768.0) / Xin);
      y1 := Trunc(32768.0 / Xin);
     end;
    Rec1 := Trunc(1.0*Xin*y1/128);
    if Frac(1.0*Xin*y1/128)>0 Then Rec1 := Rec1+1;
    Blockread(DataFile,TemData1^[0],Rec1);
    if i>32768.0 then
     begin
      Rec2 := Trunc(1.0*Xin*y2/128);
      if Frac(1.0*Xin*y2/128)>0 then Rec2:= Rec2+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Blockread(DataFile,TemData2^[0],Rec2);
   end;
  for Y := 1 to y1 do
    move(TemData1^[(Y-1)*Xin],DataPic^.DataY[Y]^,Xin);
  if Histogram then
  begin
    Wait_Message;
    Fillchar(nk1,SizeOf(nk1),0);
    k := Trunc(1.0*y1*Xin - 1);
    for c := 0 to k do
      nk1[TemData1^[c]] := nk1[TemData1^[c]]+1;
    for c := 0 to MaxGrayLevel do
      if nk1[c]>=0 then nk[c] := nk1[c]
                   else nk[c] := 65536.0 + nk1[c];
  end;
  if i>32768.0 then
  begin
    for Y := 1 to y2 do
      move(TemData2^[(Y-1)*Xin],DataPic^.DataY[Y+Y1]^,Xin);
    if Histogram then
    begin
      Fillchar(nk1,SizeOf(nk1),0);
      k := Trunc(1.0*y2*Xin - 1);
      for c := 0 to k do
        nk1[TemData2^[c]] := nk1[TemData2^[c]]+1;
      for c := 0 to MaxGrayLevel do
       if nk1[c]>=0 then nk[c] := nk[c] + nk1[c]
                    else nk[c] := nk[c] + 65536.0 + nk1[c];
      end;
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DisPose(TemData2);
DisPose(TemData1);
close(DataFile);
Tone(2000,50);
if Histogram then
begin
  HiRes;
  max := 0;
  min := 255;
  num := 0;
  for c := 0 to MaxGrayLevel do
    if nk[c]>num then num := nk[c];
  Minimum := true;k
  for c := 0 to MaxGrayLevel do
   begink
     if Minimum then
       if nk[c]>0 then
       begin
         Minimum := false;
         min := c;
       end;
     if nk[c]>0 then max := c;
     Draw(60+c*2,177,60+c*2,Round(177-(nk[c]/num)*175),1);
   end;?
  gotoXY(9,24);  write('Minimum Gray Level = ',min:3);
  gotoXY(48,24); write('Maximum Gray Level = ',max:3);
  if (min<>0) or (max<>255) thenx
   begin
    gotoXY(10,25);  write('Scaling (Y/N) ? ');
    repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          WaitForReadKey(ch);
          until (ch='Y') or (ch='N');
          gotoXY(1,25); ClrEOL;
          if ch='Y' then
          begin
            gotoXY(10,25);
            write('Scaling');
            s := 255/(max-min);
            for Y := 1 to Yin do
            begin
              gotoxy(25,25); write(Y:3);
              for X := 1 to Xin do
                DataPic^.DataY[Y]^.DataX[X] :=
                round((DataPic^.DataY[Y]^.DataX[X]-min)*s);
            end;
            Tone(2000,300);
          end;
        end
       else
        WaitForReadKey(ch);
       TextMode(3);
     end;
     DisplayScreen(DataPic,Xin,Yin);
   end
  else
  begin
   Tone(2000,300);
   gotoXY(1,25); ClrEol;
   write('file ',FileName,' not found...    Anykey to again
         ESC : Exit ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      WaitForReadKey(OK);
        gotoXY(1,25); ClrEol;
       end;
   until (Created) or (Ok=#27);
 end;

 procedure MaxMin(    Data : DataPointerI;
                  var max  : integer;
                  var min  : Integer);

 var x,y  : integer;
 begin
   max := 0;
   min := 255;
   for Y := 2 to MaxElementY-1 do
     for X := 2 to MaxElementX-1 do
     begin
       if Data^.DataY[Y]^.DataX[X]>max then max :=
          Data^.DataY[Y]^.DataX[X];
       if Data^.DataY[Y]^.DataX[X]<min then min :=
          Data^.DataY[Y]^.DataX[X];
     end;
   gotoxy(1,whereY); clreol;
   writeln(' Minimum Gray Level = ',min:4);
   writeln(' Maximum Gray Level = ',max:4);
   writeln;
   Tone(2000,300);
 end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure Scaling(    DataI : DataPointerI;
                  var DataB : DataPointerB;
                      max   : integer;
                      min   : integer);

var s    : real;
    x,y  : integer;
begin
  gotoxy(5,whereY); write('Scaling');
  s := 255/(max-min);
  for Y := 2 to MaxElementY-1 do
  begin
    gotoxy(20,whereY); write(Y:3);
    for X := 2 to MaxElementX-1 do
      DataB^.DataY[Y]^.DataX[X] :=
      round((DataI^.DataY[Y]^.DataX[X]-min)*s);
  end;
  Tone(2000,300);
end;

procedure GetSubCondition(var xmax : integer;
                          var ymax : integer);

begin
  xmax := XMaxElement;
  ymax := YMaxElement;
  gotoXY(1,2);
  writeln('Image size not greater than 256 x 256');
  writeln;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Tone(2000,70);
  gotoXY(1,whereY);
  ClrEol;
  write('Number of pixels in horizontal direction = 256');
  xmax := 256;
  MaxElementX := 256;
  writeln;
  Tone(2000,70);
  gotoXY(1,whereY);
  ClrEol;
  write('Number of pixels in vertical direction   = 256');
  ymax := 256;
  MaxElementY := 256;
  writeln;
  delay(800);
  clrscr;
end;
;

procedure GetCondition;
;
begin
  case Choose of
  '1'..'6': begin
              gotoXY(1,22);
              write('Uses [O]:Old , [N]:New or [L]:Load Data for Process');
            end;
  repeat
    WaitForReadKey(ch);
  until Ch in ['O','N','L'];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  gotoXY(1,25); ClrEol;
  case ch of
   'N' : begin
           Wait_Message;
           MoveData(DataOut,DataIn);
           gotoXY(1,25); ClrEol;
         end;
   'L' : begin
           GetSubCondition(MaxElementX,MaxElementY);
           ReadDataPicture(DataIn,MaxElementX,MaxElementY,true);
         end;
  end;
  ClrScr;
end;

procedure Histogram_Equalization;

type level = array [0..MaxGrayLevel] of real;
var  nk,Pr,Sk  : level;
     n,E       : real;
     X,Y,i     : integer;
begin
  TextMode(3);
  TextBackGround(1);
  Wait_Message;
  Fillchar(nk,SizeOf(nk),0);
  n := 1.0*MaxElementX*MaxElementY;
  for Y := 1 to MaxElementY do
  begin
    gotoxy(20,25); write(Y:3);                    (*-1-*)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for X := 1 to MaxElementX do
    nk[DataIn^.DataY[Y]^.DataX[X]] := nk[DataIn^.DataY[Y]^.DataX[X]]+1;
end;
HiRes;
E := 0;
for i := 0 to MaxGrayLevel do
begin
  Pr[i] := nk[i]/n;
  if Pr[i]>E then E := Pr[i];
end;
for i := 0 to MaxGrayLevel do
  Draw(30+i,185,30+i,Round(185-(Pr[i]/E)*170),1);
Wait_Message;
Sk[0] := Pr[0];
for i := 1 to MaxGrayLevel do
  Sk[i] := Sk[i-1] + Pr[i];
Fillchar(nk,SizeOf(nk),0);
for Y := 1 to MaxElementY do
begin
  gotoxy(20,25); write(Y:3);                    (*-2-*)
  for X := 1 to MaxElementX do
  begin
    DataOut^.DataY[Y]^.DataX[X] := Round(Sk[DataIn^.DataY[Y]
          ^.DataX[X]]*255);
    nk[DataOut^.DataY[Y]^.DataX[X]] := nk[DataOut^.DataY[Y]
          ^.DataX[X]]+1;
  end;.
end;
for i := 0 to MaxGrayLevel do
  Draw(350+i,185,350+i,Round(185-((nk[i]/n)/E)*170),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gotoXY(1,25); ClrEol;
    write('Original histogram':29,'Equalized histogram':41);
    Tone(2000,300);
    WaitForReadKey(ch);
    Textmode(3);
    DisplayScreen(DataOut,MaxElementX,MaxElementY);
  end;

procedure NeighborhoodCommand;

begin
   GraphicsVGA;
   SetFillStyle(7,1);
   Bar(0,0,640,480);
   OuttextXY(200,50,' Neighborhood Averaging');

   OuttextXY(30,100,'       Mask_1          Mask_2        3*3 image region');
   OuttextXY(30,110,'                                                      ');
   OuttextXY(30,120,'    1    1    1      1    2    1      x1   x2   x3    ');
   OuttextXY(30,130,'                                                      ');
   OuttextXY(30,140,'    1    1    1      2    4    2      x4   x5   x6    ');
   OuttextXY(30,150,'                                                      ');
   OuttextXY(30,160,'    1    1    1      1    2    1      x7   x8   x9    ');
   OuttextXY(30,170,'                                                      ');

   OuttextXY(30,250,' N[f(x,y)] = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9          ...(1)');
   OuttextXY(30,260,' N[f(x,y)] = x1 + 2x2 + x3 + 2x4 + 4x5 + 2x6 + x7 + 2x8 + x9    ...(2)');

   OuttextXY(30,340,'Select Type of Computation [1] or [2]');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  repeat,
    WaitForReadKey(ch);
  until ch in ['1','2'];
end;
;

procedure RobertCommand;
{
begin
  SetFillStyle(7,1);
  Bar(0,0,640,480);
  OuttextXY(30,50,' Roberts gradient ');
{
  OuttextXY(30,100,'                                    ');
  OuttextXY(30,110,'                                    ');
  OuttextXY(30,120,'      f(x,y)      f(x+1,y)          ');
  OuttextXY(30,130,'                                    ');
  OuttextXY(30,140,'                                    ');
  OuttextXY(30,150,'                                    ');
  OuttextXY(30,160,'      f(x,y+1)    f(x+1,y+1)        ');
  OuttextXY(30,170,'                                    ');
  OuttextXY(30,180,'                                    ');
{
  OuttextXY(30,220,' G[f(x,y)] = [ [f(x,y)-f(x+1,y+1)]^2 + [f(x,y+1)-f(x+1,y)]^2 ]^1/2   ...(1)');
  OuttextXY(30,240,' G[f(x,y)] = @f(x,y)-f(x+1,y+1)@ + @f(x,y+1)-f(x+1,y)@               ...(2)');
{
  OuttextXY(30,280,'       Type of Computation in Edge Magnitude ');
  OuttextXY(45,290,'            Type = 1 : Using Square Root ');
  OuttextXY(45,300,'            Type = 2 : Using Absolute ');
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  OuttextXY(70,340,'Select Type of Computation [1] or [2]');
  repeat,
    WaitForReadKey(ch);
  until ch in ['1','2'];
end;
;

procedure SobelCommand;
{
begin
  SetFillStyle(7,1);
  Bar(0,0,640,480);
  OuttextXY(30,50,' Sobel Operators');
{
  OuttextXY(30,100,'    compute Gx      compute Gy      3*3 image region');
  OuttextXY(30,110,'                                                    ');
  OuttextXY(30,120,'  -1   -2   -1      -1    0    1      x1   x2   x3  ');
  OuttextXY(30,130,'                                                    ');
  OuttextXY(30,140,'   0    0    0      -2    0    2      x4   x5   x6  ');
  OuttextXY(30,150,'                                                    ');
  OuttextXY(30,160,'   1    2    1      -1    0    1      x7   x8   x9  ');
  OuttextXY(30,170,'                                                    ');
{
  OuttextXY(30,210,' Gx = (x7 + 2x8 + x9) - (x1 + 2x2 + x3)');
  OuttextXY(30,220,' Gy = (x3 + 2x6 + x9) - (x1 + 2x4 + x7)');
{
  OuttextXY(30,240,' G[f(x,y)] = [Gx^2 + Gy^2]^1/2              ...(1)');
  OuttextXY(30,250,' G[f(x,y)] = |Gx| + |Gy|                    ...(2)');
{
  OuttextXY(30,280,' Type of Computation in Edge Magnitude ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   OuttextXY(45,290,'      Type = 1 : Using Square Root ');
   OuttextXY(45,300,'      Type = 2 : Using Absolute ');
{
   OuttextXY(30,340,'Select Type of Computation [1] or [2]');
   repeat,
     WaitForReadKey(ch);
   until ch in ['1','2'];
   clrscr;'
end;
;
'
procedure LaplacianCommand;
{
begin
  SetFillStyle(7,1);
  Bar(0,0,640,480);
  OuttextXY(30,50,' Laplacian Operator');
{
OuttextXY(30,110,'     Mask_1        Mask_2        Mask_3   3*3 image region');
OuttextXY(30,120,'                                                          ');
OuttextXY(30,130,'  0  -1   0    -1  -1  -1     1  -2   1    x1  x2  x3    ');
OuttextXY(30,140,'                                                          ');
OuttextXY(30,150,' -1   4  -1    -1   8  -1    -2   4  -2    x4  x5  x6    ');
OuttextXY(30,160,'                                                          ');
OuttextXY(30,170,'  0  -1   0    -1  -1  -1     1  -2   1    x7  x8  x9    ');
OuttextXY(30,180,'                                                          ');
{
OuttextXY(30,220,' L[f(x,y)] = 4x5 - x2 - x4 - x6 - x8          ...(1)');
OuttextXY(30,230,' L[f(x,y)] = 8x5 - x1 - x2 - x3 - x4 - x6
                             - x7 - x8 - x9   ...(2)');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OuttextXY(30,240,' L[f(x,y)] = x1 + x3 + 4x5 + x7 + x9 - 2x2
                              - 2x4 - 2x6 - 2x8    ...(3)');


  QuttextXY(30,340,'Select Type of Computation [1] or [2] or [3]');
  repeat,
    WaitForReadKey(ch);
  until ch in ['1','2','3'];
  clrscr;'
end;
;
procedure Laplacian_Operator;
type  k = array[-1..1,-1..1] of integer;
const Laplacian_kernel1 : k = (( 0,-1, 0),(-1, 4,-1),( 0,-1, 0));
      Laplacian_kernel2 : k = ((-1,-1,-1),(-1, 8,-1),(-1,-1,-1));
      Laplacian_kernel3 : k = (( 1,-2, 1),(-2, 4,-2),( 1,-2, 1));
var   kernel,sum,max,min,X,Y,i,j  : integer;
begin
  GraphicsVGA;
  LaplacianCommand;
  TextMode(3);
  TextBackGround(1);
  Wait_Message;
  ClearData(DataOut);
  for Y := 2 to MaxElementY-1 do
  begin
   gotoxy(20,25); write(Y:3);
   for X := 2 to MaxElementX-1 do
   begin
    sum := 0;
    for j := -1 to 1 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      for i := -1 to 1 do
       begin
        case ch of
         '1' : kernel := Laplacian_kernel1[j,i];
         '2' : kernel := Laplacian_kernel2[j,i];
         '3' : kernel := Laplacian_kernel3[j,i];
        end;
        sum := sum+DataIn^.DataY[Y+j]^.DataX[X+i]*kernel;
       end;
     if sum<0 then sum := 0;
     IntOut^.DataY[Y]^.DataX[X] := sum;
    end;
   end;
   Tone(2000,300);
   MaxMin(IntOut,max,min);
   Scaling(IntOut,DataOut,max,min);
   DisplayScreen(DataOut,MaxElementX,MaxElementY);
end;

procedure Sobel_Operators;
var Gx,Gy,magnitude : real;
    max,min,X,Y     : integer;
begin
  GraphicsVGA;
  SobelCommand;
  TextMode(3);
  TextBackGround(1);
  Wait_Message;
  ClearData(DataOut);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for Y := 2 to MaxElementY-1 do
    begin
     gotoxy(20,25); write(Y:3);
     for X := 2 to MaxElementX-1 do
   begin
     Gx := (DataIn^.DataY[Y+1]^.DataX[X-1] + 2*DataIn^.DataY[Y+1]^.DataX[X]
          + DataIn^.DataY[Y+1]^.DataX[X+1])
          - (DataIn^.DataY[Y-1]^.DataX[X-1] + 2*DataIn^.DataY[Y-1]^.DataX[X]
          + DataIn^.DataY[Y-1]^.DataX[X+1]);
     Gy := (DataIn^.DataY[Y-1]^.DataX[X+1] + 2*DataIn^.DataY[Y]^.DataX[X+1]
          + DataIn^.DataY[Y+1]^.DataX[X+1])
          - (DataIn^.DataY[Y-1]^.DataX[X-1] + 2*DataIn^.DataY[Y]^.DataX[X-1]
          + DataIn^.DataY[Y+1]^.DataX[X-1]);
      case ch of
       '1' : magnitude := sqrt(sqr(Gx)+sqr(Gy));
       '2' : magnitude := abs(Gx)+abs(Gy);
      end;
      IntOut^.DataY[Y]^.DataX[X] := round(magnitude);
    end;
   end;
   Tone(2000,300);
   MaxMin(IntOut,max,min);
   Scaling(IntOut,DataOut,max,min);
   DisplayScreen(DataOut,MaxElementX,MaxElementY);
end;

procedure Neighborhood_Averaging;

type  k = array[-1..1,-1..1] of integer;
const Neighborhood1 : k = (( 1, 1, 1),( 1, 1, 1),( 1, 1, 1));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Neighborhood2 : k = (( 1, 2, 1),( 2, 4, 2),( 1, 2, 1));
var   kernel,sum,max,min,X,Y,i,j  : integer;
begin
  GraphicsVGA;
  NeighborhoodCommand;
  TextMode(3);
  TextBackGround(1);
  Wait_Message;
  ClearData(DataOut);
  for Y := 2 to MaxElementY-1 do
  begin
   gotoxy(20,25); write(Y:3);
   for X := 2 to MaxElementX-1 do
    begin
    sum:=0;
   for j := -1 to 1 do
    for i := -1 to 1 do
    begin
      case ch of
       '1' : kernel := Neighborhood1[j,i];
       '2' : kernel := Neighborhood2[j,i];
      end;
      sum:=sum+DataIn^.DataY[Y+j]^.DataX[X+i]*kernel;
    end;
   case ch of
    '1' : IntOut^.DataY[Y]^.DataX[X] := Round(sum/9);
    '2' : IntOut^.DataY[Y]^.DataX[X] := Round(sum/16);
   end;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Tone(2000,300);
  MaxMin(IntOut,max,min);
  Scaling(IntOut,DataOut,max,min);
  DisplayScreen(DataOut,MaxElementX,MaxElementY);
end;
;
procedure Roberts_Gradient;
var m1,m2,m3,m4,magnitude : real;
    max,min,X,Y           : integer;

begin
  GraphicsVGA;
  RobertCommand;
  TextMode(3);
  TextBackGround(1);
  Wait_Message;
  ClearData(DataOut);
  for Y := 1 to MaxElementY-1 do
  begin
   gotoxy(20,25); write(Y:3);
   for X := 1 to MaxElementX-1 do
   begin
    m1 := DataIn^.DataY[Y  ]^.DataX[X  ];
    m2 := DataIn^.DataY[Y  ]^.DataX[X+1];
    m3 := DataIn^.DataY[Y+1]^.DataX[X  ];
    m4 := DataIn^.DataY[Y+1]^.DataX[X+1];
    case ch of
     '1' : magnitude:=(sqrt(sqr(m1-m4)+sqr(m3-m2)));
     '2' : magnitude:=(abs(m1-m4)+abs(m3-m2));
    end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      IntOut^.DataY[Y]^.DataX[X]:=round(magnitude);
    end;)
  end;
  Tone(2000,300);
  MaxMin(IntOut,max,min);
  Scaling(IntOut,DataOut,max,min);
  DisplayScreen(DataOut,MaxElementX,MaxElementY);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

กลุ่มผู้จัดทำโครงงาน IMAGE CARD DISPLAY ขอขอบพระคุณท่านอาจารย์ไพศาล สิทธิโยภาสกุล อาจารย์ที่ปรึกษาที่ได้ช่วยให้คำแนะนำปรึกษาด้านต่าง ๆ มาโดยตลอด

สุดท้ายนี้ กลุ่มผู้จัดทำขอขอบพระคุณท่านอาจารย์ทุกท่านที่ช่วยอบรมสั่งสอนวิชาการต่าง ๆ ทั้งมวลให้ ทำให้โครงงานนี้สำเร็จลุล่วงไปด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. เทคนิคการซ่อมเครื่องรับโทรทัศน์สี
   โดย ดร.ธวัช เมฆสวรรค์ นายโยชิคะซี ซาวามูระ.

2. อรรถสิทธิ์ หล่าสกุล "การเก็บภาพขนาด 512*256 จุด โดยใช้หน่วยความจำที่มีความเร็วต่ำ" วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2532.

3. เอกภพ นาคเสรีย์,ณรงค์ วัฒนะกิจ,สุชาติ ฉิมประดิษฐ์"วีดีโอดิจิไตเซอร์" วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์มหาบัณฑิต ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2532.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้