ปริญญานิพนธ์

IMAGE INTERFACE CARD
WITH
RESOLUTION 512*512 PIXELS

เสนอโดย.

1. นายรัชชาร์	ปิฬมพงศา	2-I	326126
2. นายชัยรัตน์	ขาวเอี่ยม	2-O	326306
3. นายศักดา	วิสุทธิวัฒน์	2-O	326335

อาจารย์ที่ปรึกษา.

รศ.ดร.ฐศักดิ์ ทีวสุวิทย์

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2533

เลขหมู่ T 33117 ฉ 2
เลขทะเบียน 027950
วัน, เดือน, ปี 18 ก.ค. 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

027950

ปริญญานิพนธ์ปีการศึกษา 2533

สาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง ชุดการประมวลผลภาพด้วยรายละเอียด 512*512 จุดภาพ

(Image interface card with resolution 512*512 pixels)

ผู้จัดทำ

1. นาย รัชชาร์ ปัทมพงศา 326126
2. นาย ชัยรัตน์ ชาวเอี่ยม 326306
3. นาย ศักดา วิสุทธิวัฒน์ 326335

.....อาจารย์ที่ปรึกษา

(รศ.ดร. ปูฝักดิ์ ชีวสุวิทย์)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์ของการวิจัย	1
1.2 ความรู้ทั่วไป	1
บทที่ 2 พื้นฐานการประมวลผลภาพ	
2.1 ส่วนประกอบพื้นฐานต่างๆในการประมวลผลภาพ	3
2.2 สัญญาณภาพและรายละเอียดของภาพจากกล้องวิดีโอ	9
บทที่ 3 หลักการแปลงสัญญาณอนาล็อคเป็นดิจิตอลและแปลงดิจิตอลเป็นอนาล็อค	
3.1 วงจรปรับค่าความต้านทาน	15
3.2 BINARY LADDER	19
3.3 ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อค (DAC)	22
3.4 ความละเอียดและความเที่ยงตรงของ D To A	28
3.5 ตัวแปลงสัญญาณอนาล็อคเป็นสัญญาณดิจิตอล (ADC)	29
บทที่ 4 หลักการจัด DMA ของระบบ	
4.1 การเกิดขบวนการ DMA ภายใน IBM PC	33
4.2 การใช้ขบวนการ DMA ใน IBM PC	34
4.3 ขั้นตอนในขบวนการ DMA	36
4.4 การอ้างแอดเดรสของ 8237-5	40
บทที่ 5 การดำเนินและการสร้าง	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 อธิบายรายละเอียดการทำงาน

6.1 วงจรแยกสัญญาณภาพ 58

6.2 วงจรการเชื่อมต่อและการขยายสัญญาณภาพ 60

6.3 วงจรการแปลงสัญญาณอนาล็อคเป็นดิจิตอลหรือADC 61

6.4 วงจรการแปลงสัญญาณดิจิตอลเป็นอนาล็อคหรือDAC 64

6.5 วงจรการขยายสัญญาณและการผสมสัญญาณภาพกับสัญญาณซิงค์ 64

6.6 วงจรการเก็บข้อมูลในหน่วยความจำ 66

6.7 วงจรการสร้างแอสเดรสให้กับ Dynamic RAM 69

6.8 วงจรการกำเนิดสัญญาณนาฬิกา 71

6.9 วงจรการกำหนดฟังค์ชั่นของ RAM และการสลับฟังค์ชั่น 71

6.10 วงจรการสร้างสัญญาณซิงค์ทางแนวนอนและแนวตั้ง 74

6.11 วงจรการสร้าง Odd-Even Feed 76

6.12 วงจรการเริ่มต้นการเก็บภาพ 77

6.13 วงจรสร้างสัญญาณควบคุม RAM 79

6.14 วงจร INTERFACE 81

บทที่ 7 หลักการนำ PAL มาใช้งาน

7.1 คุณสมบัติของไอซี PAL 85

7.2 ขั้นตอนการทำ 86

บทที่ 8 การเขียนโปรแกรมและการใช้งานโปรแกรม

8.1 โฟลชาร์ตการเขียนโปรแกรม 88

8.2 การใช้งานโปรแกรม 96

บทที่ 9 บทวิจารณ์และสรุปผล 105

กิตติกรรมประกาศ

หนังสืออ้างอิง

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ชุดประมวลผลภาพด้วยรายละเอียด 512*512 จุดภาพ
นักศึกษา	นายรัชชาร์ ปัทมพงศา
	นายชัยรัตน์ ขาวเอี่ยม
	นายศักดา วิสุทธิวัฒน์
อาจารย์ที่ปรึกษา	รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์
ปีการศึกษา	พ.ศ. 2533

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จะเป็นการนำสัญญาณภาพจากอุปกรณ์ที่สามารถผลิตสัญญาณวิดีโอ ออกมา เช่น กล้อง วิดีโอ,วิดีโอ เทป เป็นต้น นำมาเก็บในหน่วยความจำที่มีขนาด 256 กิโลไบท์ จำนวน 3 ชุด คือ ชุดภาพขาวดำ(Y), ชุดภาพสี R-Y และชุดภาพสี B-Y ซึ่งความละเอียดของสัญญาณภาพขาวดำมี 256 ระดับ ส่วนสัญญาณภาพสีจะมีความละเอียด 64 ระดับ การทำงานจะสามารถแสดงผลภาพเป็นภาพสีหรือภาพขาวดำก็ได้ ขึ้นอยู่กับการควบคุมทางด้านซอฟท์แวร์ ปริญญานิพนธ์นี้ได้ดัดแปลงมาจากของเดิมซึ่งเป็นการประมวลผลภาพขนาด 256*256 จุดภาพ และใช้สแตติกส์-แรมเป็นหน่วยความจำมาเป็นการประมวลผลภาพที่มีขนาด 512*512 จุดภาพ โดยใช้ไดนามิกส์ แรม เป็นหน่วยความจำ ซึ่งทำให้ประหยัดค่าใช้ได้มากกว่า การติดต่อและการควบคุมฟังค์ชั่นต่างๆจะใช้เครื่องคอมพิวเตอร์ IBM PC เป็นตัวสั่งการ โดยจะแบ่งการทำงานได้ 6 ฟังค์ชั่นคือ ฟังค์ชั่นการทำงานปกติระหว่างกล้อง วิดีโอ กับจอภาพมอนิเตอร์ TV พร้อมกับนำข้อมูลไปเก็บในชุดหน่วยความจำ, ฟังค์ชั่นการนำข้อมูลจากหน่วยความจำมาแสดงที่จอมอนิเตอร์ TV, ฟังค์ชั่นการนำข้อมูลภาพจากแฟ้มข้อมูล ไปยังหน่วยความจำ, ฟังค์ชั่นการนำข้อมูลไปเก็บในแฟ้มข้อมูล, ฟังค์ชั่นการนำข้อมูลภาพส่งไปพิมพ์ลงในกระดาษโดยใช้ปริ้นต์เตอร์ และฟังค์ชั่นสำหรับการอ่านชื่อของแฟ้มข้อมูลจากแผ่นดิสต์ ซึ่งทุกฟังค์ชั่นจะแสดงการทำงานเป็นเมนูและภาพประกอบให้เห็นเพื่อสะดวกต่อการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Image interface card with resulotion 512*512 pixels

Name Mr.Ratchar Pathamapongsar
Mr.Chairat Khaoean
Mr.Sakda Visuthiwat

Advisor Assosiate Professer Dr.Fusak Cheevasuvit

Academic Year 1990

ABSTRACT

This thesis is the bringing of Video-signal from Video divice such as Video-camera,Video tape.The Video-signal will be stored in the memory with the capacity of 256 Kbytes per unit for total 3 units. The black-and-white signal (Y),the colour signal R-Y and B-Y. The black-and-white signal(Y) consists of 256 gray levels while each colour signal consists 64 colour levels.A picture can be displayed in colour or in black-and-white by using the software programmed. This thesis has been adapted from the former one which has the resolution of 256*256 pixels and also it requires Static Ram Memory. While in this thesis, image interface card has the resolution of 512*512 pixels and requires Dynamic Ram Memory for theraison of economization. The IBM PC has been used to control all functions of this image interface card. The operation is divided into 6 functions. The first one is digitizing an image from Video-camera for saving into the memory. The second one is displying the digitized image data from memory to onto the monitor. The third is loading the data from a file for saving in the memory. The forth one is saving the digitized image data into a file. The fifth is print out an image by printer. And the last function is listing the data files name from disks. These operation functions will be shown as a menu on the monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาการทำงานของระบบการประมวลผลภาพ และพัฒนาระบบการประมวลผลภาพจากของเก่าให้ดีขึ้น และประหยัดที่สุด

2. เพื่อขยายขอบเขตความละเอียดของจุดภาพให้ละเอียดขึ้น ซึ่งในปริญญานิพนธ์นี้จะมีขนาดความละเอียด 512*512 จุดภาพ

3. เพื่อศึกษาถึงอุปกรณ์ต่างๆที่นำมาใช้กับการประมวลผลภาพ และสามารถนำอุปกรณ์ที่มีราคาถูกมาประยุกต์ใช้งานได้

4. เพื่อศึกษาการลดรูปวงจรโดยใช้หลักการของ Programeble Array Logic (PAL)

5. เพื่อศึกษาการใช้ software ต่างๆ ที่นำมาทำการออกวงจรและออกแบบลายปริ้น เช่น smART WORK, Or CAD/SDT, Or CAD/PCB, P-CAD เป็นต้น

6. เพื่อศึกษาระบบการทำงานของเครื่องคอมพิวเตอร์ IBM PC ระบบไซเคิลการเข้าแอดเดรสของหน่วยความจำ ระบบ DMA (Direct Memory Access) และระบบอื่นๆที่จำเป็น

7. เพื่อศึกษาถึงระบบสัญญาณต่างๆจากอุปกรณ์สร้างสัญญาณ VDO และการนำสัญญาณต่างๆมาใช้งาน

8. เพื่อนำเอาข้อมูลภาพที่ได้ไปบันทึกลงในแผ่นดิสต์เพื่อใช้งานในอนาคตต่อไป

1.2 ความรู้ทั่วไป

การประมวลผลภาพ(Image Processing) ขนาด 512*512 จุดภาพ คือการออกแบบวงจรทางด้านฮาร์ดแวร์ (Hardware) ที่สามารถควบคุมการเก็บข้อมูลภาพเป็นดิจิตอลในหน่วยความจำได้ 256 Kbytes ซึ่งข้อมูลเหล่านี้ได้มาจากกล้องโทรทัศน์, กล้องวีดีโอ, วีดีโอเทป หรืออุปกรณ์ที่สามารถผลิตสัญญาณโทรทัศน์ สัญญาณที่ได้จากอุปกรณ์เหล่านี้จะอยู่ในรูปของสัญญาณอนาล็อค ฉะนั้นต้องเปลี่ยนให้อยู่ในรูปของข้อมูลทางดิจิตอล เพื่อที่จะนำมาเก็บในหน่วยความจำ และสามารถนำไปประมวลผลได้ ซึ่งขบวนการนี้เราเรียกว่า ขบวนการประมวลผลภาพจริง (Realtime imageprocessing system)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่องราวเกี่ยวกับวิธีการประมวลผลภาพจริงนี้ได้รับความสนใจเป็นอย่างยิ่ง ทั้งนี้เนื่องมาจากจินตนาการของมนุษย์ที่สำคัญมี 2 ประการ คือ

1. การปรับปรุงภาพให้ดีขึ้นเพื่อการตีความของมนุษย์เอง เช่น ภาพที่ได้จากดาวเทียม LANDSAT(เป็นดาวเทียมเพื่อการสำรวจทรัพยากร) ภาพที่ส่งมาจะมีลักษณะไม่ชัดเจนยากต่อการตีความว่าพื้นที่ส่วนใด คือ พื้นที่ทางกสิกรรม พื้นที่ส่วนใด คือ ป่าไม้ และ ฯลฯ ดังนั้นมนุษย์จึงได้พยายามหาวิธีการประมวลผลภาพ ซึ่งเป็นวิธีทางซอฟท์แวร์โดยผ่านขบวนการทางคณิตศาสตร์ เพื่อให้ได้ภาพที่สามารถตีความได้ดียิ่งขึ้น

2. ความพยายามที่จะให้เครื่องจักรสามารถเข้าใจถึงภาพที่เห็นได้ เช่น ความพยายามที่จะพัฒนาหุ่นยนต์ให้มีความสามารถใกล้เคียงกับมนุษย์ สามารถแยกแยะวัตถุต่างๆออกจากกันได้ ซึ่งเป็นเรื่องของศาสตร์ทางด้าน Pattern recognition

ในปัจจุบันนี้การประมวลผลภาพได้ประยุกต์ใช้กับงานต่างๆมากมาย เช่น ทำแฟ้มประวัติของบุคคลโดยมีภาพประกอบ ทำเป็นไฟล์ภาพเอ็กซ์เรย์แพทการเก็บฟิล์มจำนวนมากๆ ใช้สร้างเทคนิคภาพเพื่องานโฆษณา และในต่างประเทศได้นำระบบนี้ไปประยุกต์ใช้กับงานทางด้านการออกแบบทรงผมให้เข้ากับใบหน้าของลูกค้า โดยการถ่ายภาพใบหน้าของลูกค้าเก็บเป็นข้อมูลแล้วนำแบบทรงผมต่างๆ ซึ่งเก็บอยู่ในรูปข้อมูลคอมพิวเตอร์ มาทดลองผสมเข้ากับใบหน้าของลูกค้า จนได้ทรงผมที่ถูกใจของลูกค้ามากที่สุด แล้วจึงค่อยตกแต่งทรงผม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

พื้นฐานการประมวลภาพ (BASIC IMAGE PROCESSING)

2.1 ส่วนประกอบพื้นฐานต่างๆในการประมวลผลภาพ

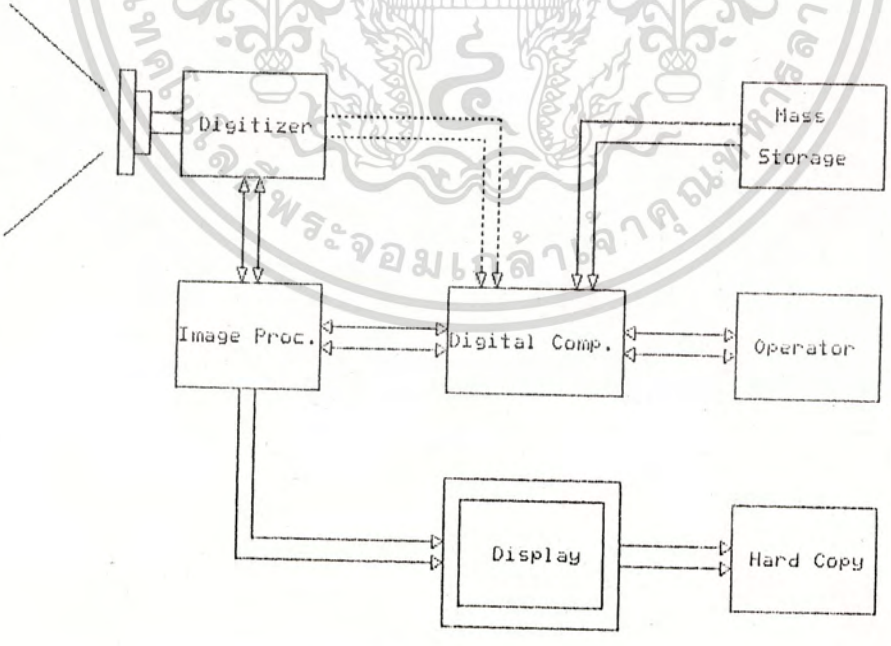
(Element of a image processing system)

ในขบวนการประมวลผลภาพจริง มีส่วนประกอบพื้นฐานที่สำคัญดังรูปที่ 2.1 ซึ่งผลจะอธิบายได้ดังนี้

1. ตัวประมวลผลภาพ (Image Processor)

ตัวประมวลผลภาพถือเป็นหัวใจสำคัญของขบวนการประมวลผลภาพเพราะเป็นส่วนของฮาร์ดแวร์ที่มีหน้าที่สำคัญอยู่ 4 ประการ คือ

1.1 การเก็บภาพ (Image acquisition) เป็นการแปลงสัญญาณโทรทัศน์ ซึ่งอยู่ในรูปของสัญญาณอนาล็อค ให้เป็นข้อมูลทางดิจิตอลเพื่อที่จะนำไปเก็บในหน่วยความจำ ซึ่งตัวประมวลผลภาพสามารถดิจิไตร์ภาพจากสัญญาณโทรทัศน์ได้ภายในช่วงเวลาหนึ่งเฟรม



รูปที่ 2.1 วงจรพื้นฐานในระบบ Image Processing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 การบันทึก (Storage) หรือเรียกอีกอย่างหนึ่งว่า Frame Buffer คือส่วนหน่วยความจำที่ใช้เก็บภาพที่ได้จากการดิจิไตซ์ ซึ่งจะใช้หน่วยความจำมากหรือน้อยขึ้นอยู่กับความละเอียดของจุดภาพ ในโครงงานนี้ให้ความละเอียดของจุดภาพ 512*512 จุดภาพ ซึ่งต้องใช้หน่วยความจำ 256 Kbytes

1.3 การประมวลผลระดับต่ำ (Low-level processing) เป็นส่วนของฮาร์ดแวร์ที่ใช้ในการทำงานทางด้านลอจิกมักนิยมเรียกว่า ALU (Arithmetic Logic Unit) ออกแบบมาเพื่อเพิ่มความเร็วของระบบ

1.4 การแสดงผล (Display) เป็นการนำข้อมูลที่อยู่ใน Memory ออกมาแสดงบนจอมอนิเตอร์ โดยการเปลี่ยนจากสัญญาณข้อมูลมาให้อยู่ในรูปของสัญญาณอนาล็อค เพื่อที่จะทำงานต่อไป

2. ดิจิไตซ์เซอร์ (Digitizer) เป็นอุปกรณ์ที่มีหน้าที่เปลี่ยนสัญญาณภาพที่เป็นสัญญาณอนาล็อคให้เป็นข้อมูลดิจิตอล เพื่อให้สามารถนำไปประมวลผลหรือใช้ได้กับดิจิตอลคอมพิวเตอร์ อุปกรณ์พื้นฐานสำคัญที่รับคือทุกคือ vidicon camera, microdensitometer, flying spot scanner, photosensitive solid-state array เป็นต้น แต่ที่นิยมใช้กันแพร่หลายก็มีอยู่ 2 อย่าง คือ vidicon camera กับ photosensitive solid-state arrays ซึ่งอุปกรณ์สองชนิดนี้เป็นตัวจำแนกภาพที่สามารถเก็บบันทึกภาพที่ต้องการได้ และมีประโยชน์เพิ่มขึ้นที่สามารถดิจิไตซ์ภาพที่มีความเข้มเพียงพอที่จะแบ่งแยกความแตกต่างได้ หลอดวิดิคอน ภาพจะไปตกกระทบที่ผิวหน้าของหลอดโฟโตเซนสิตีฟ ภาพที่ไปปรากฏที่ผิวหน้าของหลอดจะทำให้เกิดรูปแบบการเปลี่ยนแปลงของ conductivity ตามการกระจายของแสงที่สะท้อนมาจากภาพในขณะที่ electrol beam กำลังสแกนบนผิวหน้าของ photoconductivity target, Electrol beam นี้จะสร้างความแตกต่างให้เกิดขึ้นที่ collector ของหลอด ตามความสว่างของแสงที่มากระทบหลอด Digital image ที่ได้ก็คือการเอาสัญญาณที่ออกจาก collector นี้ไป Quantizing ในตำแหน่งที่ตรงกับ scanning หลอดวิดิคอนนี้มักจะอยู่ในรูปของกล้องทีวี และเอาสัญญาณทีวีนี้ไปใช้ในการสร้างส่วนต่างๆ ของ Image

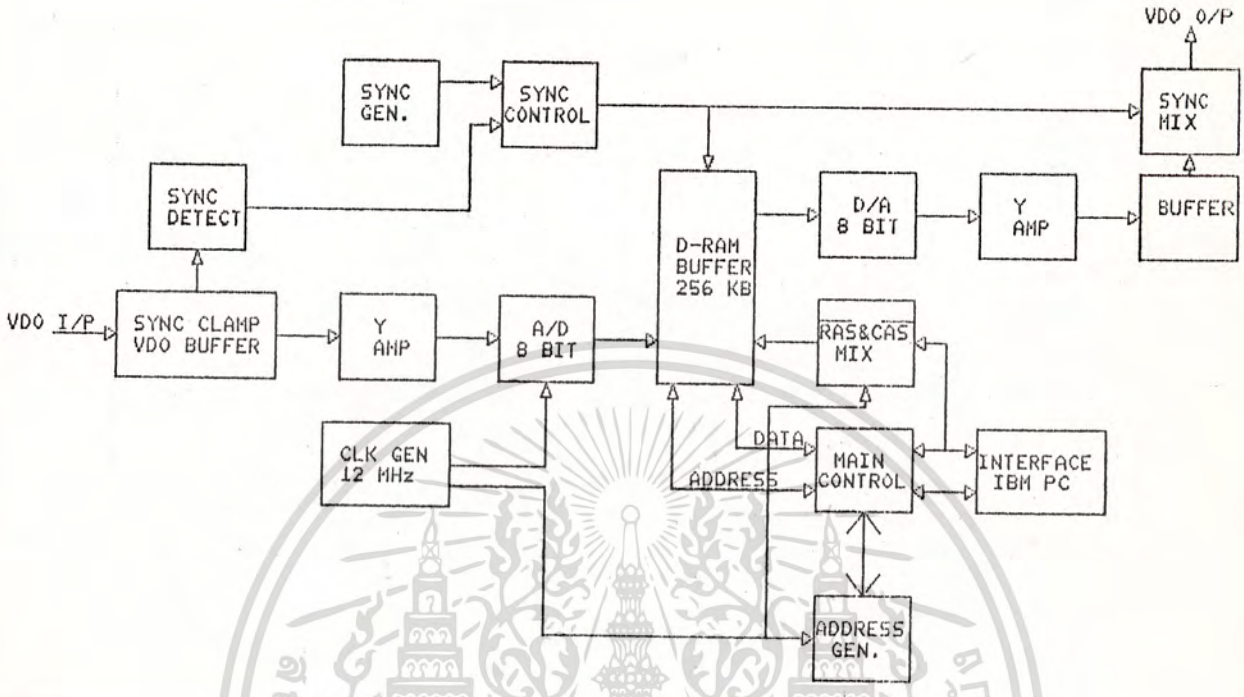
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คอมพิวเตอร์ (Computer) เนื่องจากตัวประมวลผล(Image processor) มีความสามารถเพียงแต่แปลงข้อมูลจากสัญญาณโทรทัศน์ให้มาอยู่ในรูปของข้อมูลดิจิตอลในหน่วยความจำ แต่ในบางครั้งระบบการประมวลผลภาพจำเป็นต้องมีการคำนวณทางคณิตศาสตร์ที่ยุ่งยากหรืออาจจะมีความต้องการผสมอักษรให้เข้ากับรูปภาพ แล้วบันทึกเก็บไว้และอื่นๆอีกมาก เพื่อสนองตอบต่อความต้องการเหล่านี้ เราจึงต้องทำการเชื่อมระบบของตัวประมวลผลภาพให้เข้ากับระบบคอมพิวเตอร์ ซึ่งในโครงงานนี้เราเชื่อมระบบเข้ากับ IBM PC ซึ่งเป็นระบบคอมพิวเตอร์ที่ได้รับความนิยมกันทั่วไป

4. อุปกรณ์ที่ใช้ในการเก็บข้อมูล (Storage Devices) ภาพที่ได้จากการประมวลผลมีความละเอียด 512*512 pixels ซึ่งแต่ละpixel เข้าระหัสเป็น 8 บิท จะต้องใช้หน่วยความจำทั้งหมดถึง 256 Kbyte อุปกรณ์ที่ใช้เก็บข้อมูลที่นิยมกัน ได้แก่ ฟลอปปี้ดิสต์ ซึ่งสามารถเก็บข้อมูลได้ 360 Kbyte, ฮาร์ดดิสต์ ซึ่งสามารถเก็บข้อมูลได้ตั้งแต่ 5 Mbytes ขึ้นไป และแมกเนติกส์ติสต์ เป็นต้น

5. อุปกรณ์แสดงผล (Display and Recorder) เป็นส่วนที่ใช้แสดงภาพที่ได้จากตัวประมวลผลภาพ, ภาพที่เก็บอยู่ในฟลอปปี้ดิสต์ และฯลฯ ซึ่งได้แก่ จอโมโนโครม หรือจอโทรทัศน์ ที่สามารถแสดงภาพที่มีระดับความสว่างแตกต่างกัน มากๆได้ เครื่องพิมพ์(Printer) ก็เป็นอุปกรณ์แห่งที่ใช้ในการแสดงภาพที่ไม่ต้องการความสว่างมากนัก ทั้งนี้เนื่องจากในแต่ละจุดภาพของเครื่องพิมพ์จะให้ระดับความสว่างเพียง 2 ระดับ ดังนั้นจึงต้องสร้างเมตริกส์ของจุดภาพของเครื่องพิมพ์ที่มีขนาด n*n โดย n =2,3,4,.. ซึ่งแล้วแต่ว่าเราจะต้องการภาพที่มีความละเอียดแค่ไหน รวมทั้งความสามารถในการพล็อตเป็นจุดต่อเถว ของเครื่องเองด้วยเราเรียกการแสดงภาพในลักษณะนี้ว่า Half-tone picture นอกจากนี้ยังมีอุปกรณ์ที่ใช้แสดงภาพได้อีก เช่น เลเซอร์ปริ้นเตอร์(Laser Printer) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

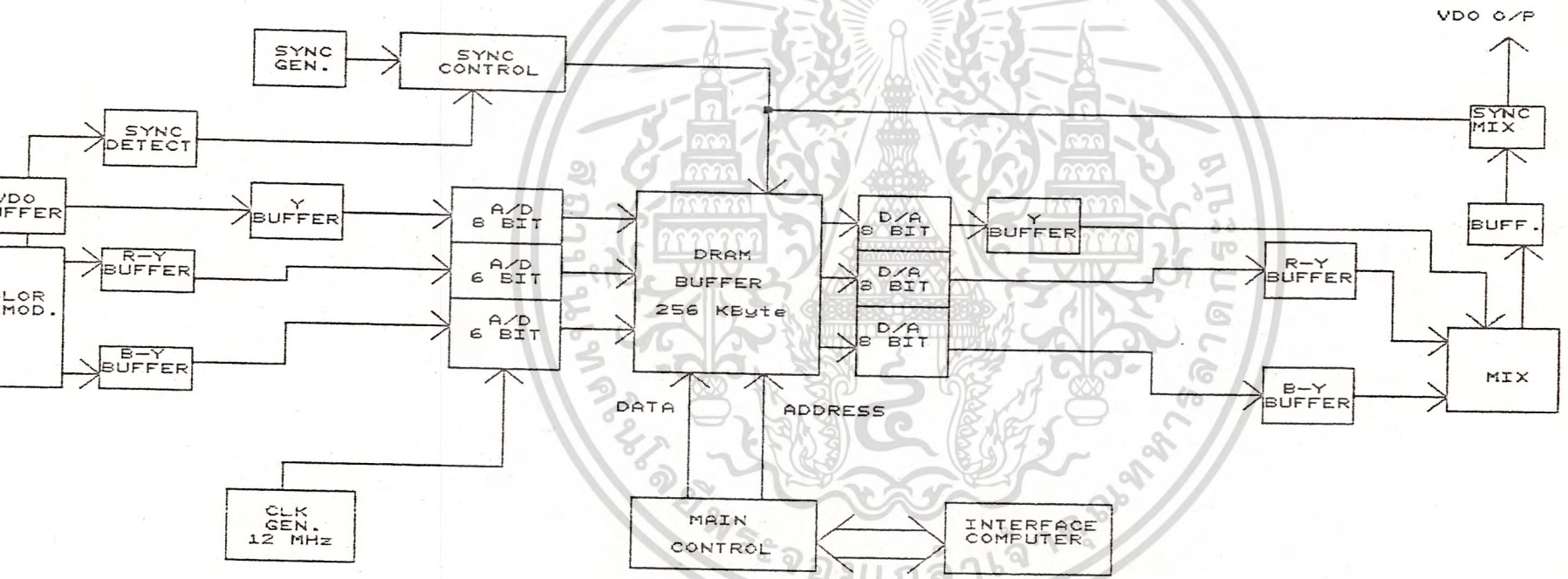


รูปที่ 2.2 Block diagram of image processor ภาพขาวดำ

จาก Block diagram รูปที่ 2.2 จะเห็นได้ว่าเมื่อมีสัญญาณวีดีโอป้อนเข้ามาที่ VDO I/P ซึ่งจะมี Buffer เป็นตัวรองรับสัญญาณอยู่ และการแยกสัญญาณเป็น 2 ส่วน คือสัญญาณภาพขาวดำ(Y) และสัญญาณซิงค์ โดยใช้วงจร Low-pass filter เป็นตัวแยก สัญญาณภาพขาวดำ และใช้วงจร Comparator เป็นตัวแยกสัญญาณซิงค์ ซึ่งสัญญาณซิงค์ที่แยกได้จะต้องประกอบด้วย ซิงค์ทางแนวนอน ซิงค์ทางแนวตั้ง และสัญญาณ O/E F (Odd/Even Field) เพื่อที่จะนำไปควบคุมการทำงานของRAM และอุปกรณ์อื่นๆ

สำหรับสัญญาณที่ผ่าน Low pass filter จะนำมาทำการขยายโดยผ่าน Amplifier จากนั้นนำสัญญาณที่ได้ซึ่งเป็นสัญญาณอนาล็อค ทำการแปลงเป็นข้อมูลดิจิตอลขนาด 8 บิท (มีความเข้มของแสง $2^8 = 256$ ระดับ) เพื่อที่จะนำมาเก็บในหน่วยความจำและทำการประมวลผล โดยจะถูกควบคุมด้วยสัญญาณซิงค์อีกทีหนึ่ง เมื่อต้องการข้อมูลที่อยู่ในหน่วยความจำมาทำการแสดงผล จะต้องทำการแปลงข้อมูลกลับเป็นสัญญาณอนาล็อคและสัญญาณซิงค์อีกทีหนึ่งเพื่อให้สัญญาณเหมือนอุปกรณ์ที่จะแสดงผล

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


SYNC GEN.
SYNC CONTROL
VDO O/P
SYNC DETECT
SYNC MIX
VDO FFER
Y BUFFER
A/D 8 BIT
A/D 6 BIT
A/D 6 BIT
DRAM BUFFER 256 KByte
D/A 8 BIT
D/A 8 BIT
D/A 8 BIT
Y BUFFER
BUFF.
R-Y BUFFER
LOR MOD.
R-Y BUFFER
B-Y BUFFER
B-Y BUFFER
MIX
DATA
ADDRESS
CLK GEN. 12 MHz
MAIN CONTROL
INTERFACE COMPUTER

อธิบายจาก Block Diagram รูปที่ 2.3

จาก Block Diagram สัญญาณจากกล้องวีดีโอจะถูกป้อนเข้าที่ VDO input จากนั้นจะทำการแยกสัญญาณออกเป็นส่วนๆโดยใช้ Comparaterเป็นตัวแยกสัญญาณซิ้งค์ ใช้ Low pass filter เป็นตัวแยกสัญญาณภาพขาวดำ(Y) และสำหรับสัญญาณภาพสี (R-YและB-Y) นั้นจะใช้ Pal decode เป็นตัวแยกออกมา จากนั้นนำสัญญาณภาพขาวดำไปขยายและเข้าวงจรแปลงสัญญาณอนาล็อคเป็นดิจิตอลขนาด 8 บิท แล้วนำข้อมูลที่ได้ไปเก็บในหน่วยความจำ ซึ่งการเก็บข้อมูลดิจิตอลนี้จะถูกควบคุมโดยสัญญาณซิ้งค์อีกทีหนึ่ง เพื่อให้การเก็บข้อมูลเป็นไปอย่างถูกต้อง โดยข้อมูลดิจิตอลนี้ 1 ภาพ จะใช้เนื้อที่ของหน่วยความจำ 256 Kbyte

ส่วนสัญญาณ R-Y และ B-Y นั้นจะถูก Multiplex ก่อน จากนั้นจึงนำสัญญาณที่ถูก Multiplexแล้ว ไปป้อนเข้าวงจรแปลงสัญญาณอนาล็อคเป็นสัญญาณดิจิตอล ซึ่งการที่เราทำการ Multiplex สัญญาณนั้นทำให้เราสามารถประหยัดวงจรแปลงสัญญาณอนาล็อคเป็นดิจิตอลไปได้หนึ่งชุด จากนั้นนำข้อมูลที่ได้จากการแปลงอนาล็อค ซึ่งมีขนาด 6 บิท ไปเก็บลงในหน่วยความจำ โดยการเก็บข้อมูลจะถูกควบคุมด้วยซิ้งค์เหมือนกับสัญญาณภาพขาวดำ ดังนั้นข้อมูลในหน่วยความจำจะเป็นข้อมูลของ R-Y และ B-Y สลับกันไปตามความเร็วของการ Multiplex และอีกส่วนที่สำคัญ คือ การอ้างแอดเดรสของหน่วยความจำในการเก็บข้อมูลของสัญญาณขาวดำ(Y)และสัญญาณสี R-Y,B-Y ซึ่งจะต้องมีความสัมพันธ์กับความเร็วของ Clock Pulse และสัญญาณซิ้งค์

ข้อมูลที่ถูกเก็บอยู่ในหน่วยความจำนี้ เมื่อต้องการแสดงภาพกลับออกมาจะต้องทำการแปลงข้อมูลกลับกันกับตอนที่ทำการเก็บข้อมูล คือต้องทำการแปลงข้อมูลดิจิตอลให้เป็นสัญญาณอนาล็อค ทั้งสัญญาณภาพขาวดำและสัญญาณ R-Y,B-Y ซึ่งสัญญาณ R-Y,B-Y ซึ่งสัญญาณ R-Y,B-Y นี้ตอนที่เราเก็บข้อมูลเราได้ทำการ Multiplex ข้อมูลเอาไว้ ดังนั้นเมื่อต้องการสัญญาณเดิมเราจึงต้องทำการ Demultiplex โดยความเร็วที่ทำการ Demultiplex นี้จะต้องมีความเร็วเท่ากับเวลาในการ Multiplex จากนั้นจึงนำสัญญาณที่ได้จากการแปลงเป็นอนาล็อคมารวมกันโดยเข้าวงจร PAL Encoder จากนั้นจึงนำมารวมกันกับซิ้งค์อีกทีหนึ่งเพื่อให้เป็นสัญญาณ Composite ตามเดิม

นอกจากนี้ข้อมูลที่ถูกเก็บอยู่ในหน่วยความจำ ยังสามารถ Interface เข้ากับไมโครคอมพิวเตอร์ เพื่อดึงข้อมูลมาเก็บในดิสต์และพิมพ์ออกไปยังปริ้นเตอร์ ได้โดยการเชื่อมต่อสัญญาณ ซึ่งอาศัยวงจร Multiplex ช่วยในการนำข้อมูลมาแสดงผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 สัญญาณภาพและรายละเอียดของภาพจากกล้องวีดีโอ

(Video signal and Television signal description)

ภาพที่ตาเรามองเห็นได้นั้น เกิดขึ้นเนื่องจากมีแสงไปกระทบวัตถุนั้น แล้วสะท้อนมาเข้าตาเรา ถ้าหากไม่มีแสงเราจะไม่สามารถเห็นภาพนั้นได้เลย และการที่เรามองเห็นภาพเป็นภาพสีต่างๆ ได้นั้นก็เนื่องจากว่าวัตถุนั้นๆมีความสามารถในการดูดกลืนแสงของสีไม่เท่ากัน จึงทำให้เราสามารถมองเห็นวัตถุนั้นๆ เป็นสีต่างๆ ได้

ปกติภาพที่เราเห็นนั้นก็คือแสงที่สะท้อนออกจากวัตถุนั้นๆ ในการเปลี่ยนความเข้มของแสงที่สะท้อนออกมานั้นให้เป็นสัญญาณทางไฟฟ้า เราเรียกสัญญาณที่ได้จากการเปลี่ยนแสงเป็นไฟฟ้านี้ว่า สัญญาณภาพ อุปกรณ์ที่ทำหน้าที่เปลี่ยนความเข้มของแสงให้เป็นไฟฟ้านั้นมีอยู่ด้วยกันหลายชนิด ดังที่ได้กล่าวมาแล้วในเรื่องการดิจิไตซ์ แต่ที่เรารู้จักกันดี คือ หลอดวิดิคอน(Vidicon tube) ซึ่งเราจะพบอยู่ทั่วไปในรูปของกล้องวีดีโอ หรือกล้องโทรทัศน์

สัญญาณที่ได้มาจากกล้องวีดีโอหรือโทรทัศน์นั้น จะมีลักษณะเป็นสัญญาณต่อเนื่อง (Analog signal) และจะมีลักษณะเป็นสัญญาณรวม (Composite video signal) ซึ่งสัญญาณนี้ประกอบด้วย สัญญาณภาพ, สัญญาณแบล้งค์กิ้งทางด้านแนวนอนและแนวตั้ง, สัญญาณซิ้งค์ทางด้านแนวนอนและแนวตั้ง ซึ่งพอที่จะอธิบายรายละเอียดดังต่อไปนี้

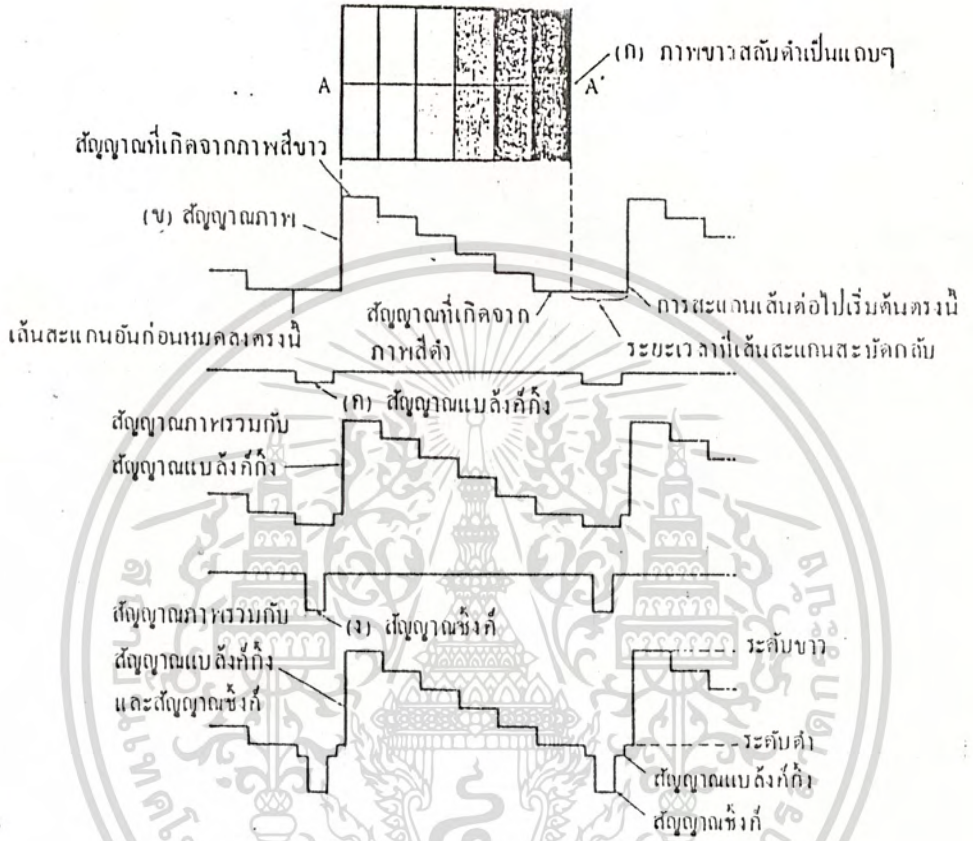
1. สัญญาณแบล้งค์กิ้ง (Blanking Pulse)

เป็นสัญญาณที่ใช้เพื่อลบเส้นสแกนสะบัดทั้งในแนวนอนและแนวตั้ง เพื่อมิให้สังเกตุเห็นได้ชัดทางจอภาพ สำหรับโทรทัศน์จะมีสัญญาณแบล้งค์กิ้งระหว่างเส้นสแกน (แบล้งค์กิ้งทางแนวนอน) ประมาณ 10 ไมโครวินาที ทำนองเดียวกันสัญญาณแบล้งค์กิ้งระหว่างฟิลด์ (แบล้งค์กิ้งทางแนวตั้ง) ขนาดประมาณ 1,250 ไมโครวินาที

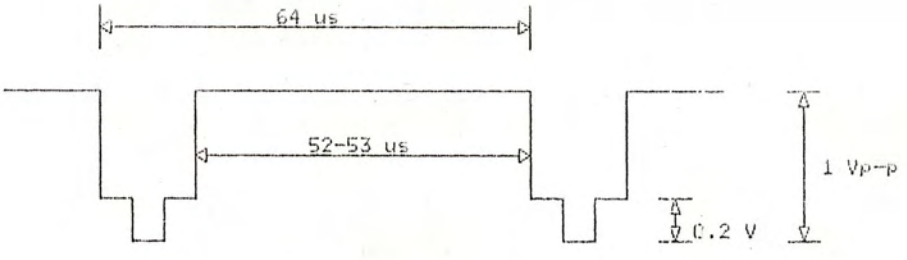
2. สัญญาณซิ้งค์ (Sync Pulse)

เป็นสัญญาณที่ใช้เพื่อช่วยให้วงจรหักเหทางแนวนอนและแนวตั้ง ในเครื่องส่งและเครื่องรับโทรทัศน์มีความถี่ตรงกันตลอดเวลา ซึ่งโทรทัศน์ระบบยุโรปสัญญาณซิ้งค์ทางแนวนอนมีความถี่ 15,625 Hz และสัญญาณซิ้งค์ทางแนวตั้งมีความถี่ 50 Hz เนื่องจากว่าความถี่ของสัญญาณซิ้งค์เท่ากับความถี่ของสัญญาณแบล้งค์กิ้งพอดี จึงจำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
027950



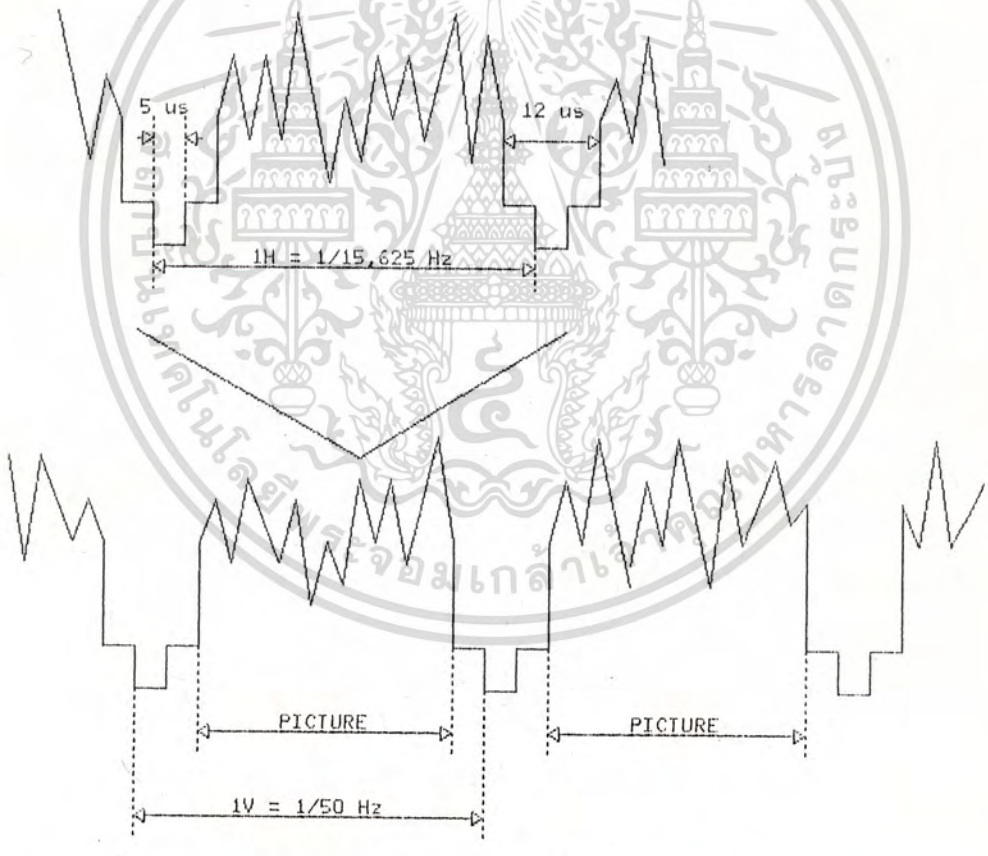
รูปที่ 2.4 รูปร่างของสัญญาณโทรทัศน์ที่เกิดจากภาพขาวสลับดำเป็นแถวๆ



รูปที่ 2.5 ขนาดและช่วงเวลาของสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องป้องกันการรบกวนที่อาจจะเกิดขึ้นโดยการกำหนดขนาดของซิ้งค์พัลซ์ให้น้อยกว่าขนาดของแบล้งค์กิ้งพัลซ์ คือทำให้ซิ้งค์พัลซ์ทางแนวนอน มีขนาดเพียง 2 ไมโครวินาที และซิ้งค์พัลซ์ทางแนวตั้งมีขนาด 190 ไมโครวินาที เท่านั้น นอกจากนี้ยังใช้วิธีการส่งซิ้งค์พัลซ์เหล่านี้บนไปกับแบล้งค์กิ้งพัลซ์อีกด้วย โดยให้ให้ฐานของซิ้งค์พัลซ์อยู่ทับกับขอบบนของแบล้งค์กิ้งพัลซ์ เมื่อจัดขอบความต่างศักย์ให้ระดับสูงสุดของแบล้งค์กิ้งพัลซ์ เป็นระดับดำมืดจนมองไม่เห็นแล้วระดับของซิ้งค์พัลซ์ที่อยู่บนยอดสูงสุดของแบล้งค์กิ้งพัลซ์ก็จะดำมืดสนิทด้วย และไม่ทำให้เกิดการรบกวนภาพที่หน้าจอแต่อย่างใด(ดูรูปที่ 2.4,2.5)



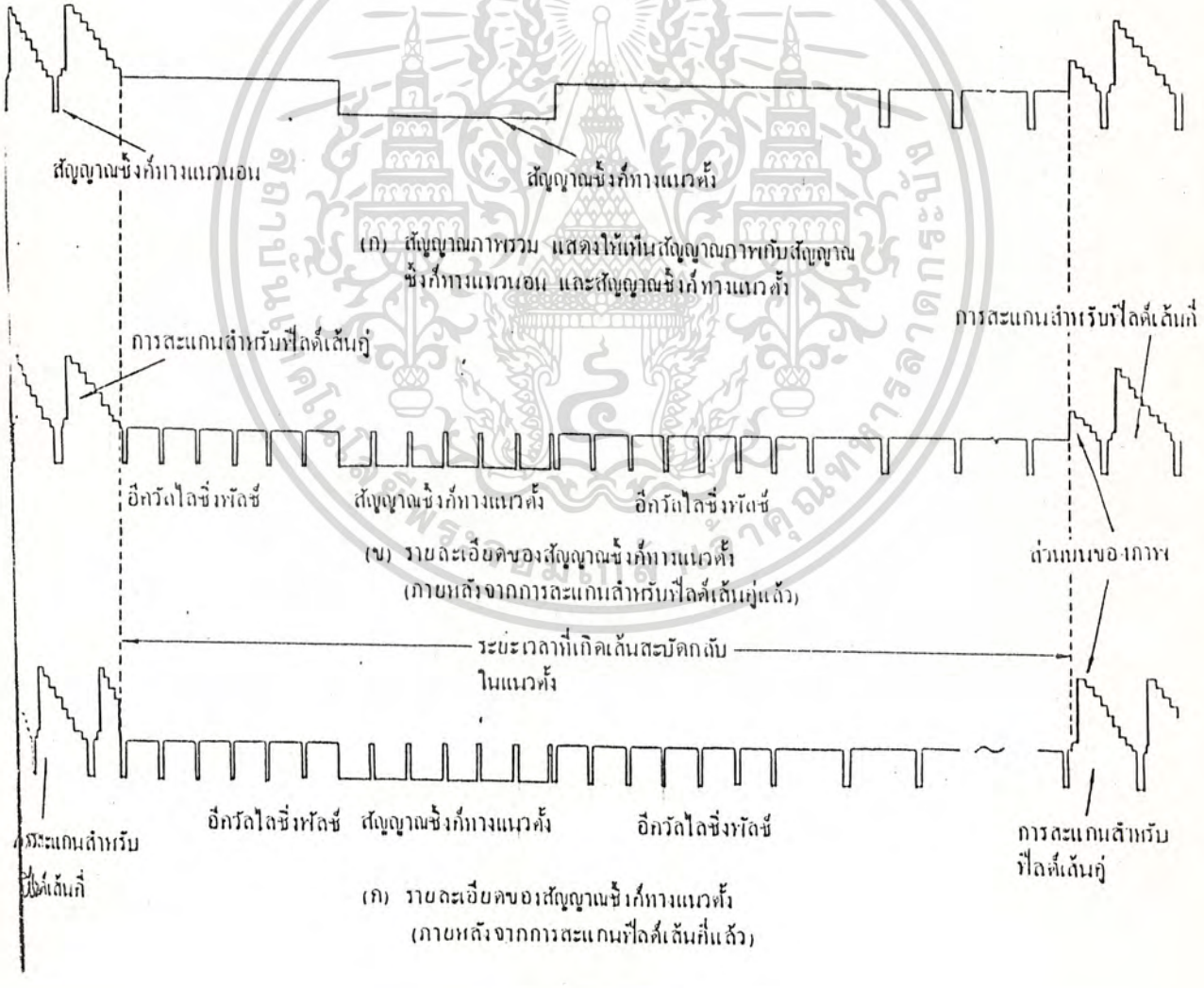
รูปที่ 2.6 แสดงสัญญาณในแต่ละเส้นสแกนไลท์ของแนวตั้งและแนวนอน

3. สัญญาณอีควอไลท์ซิ่ง (Equalizing Pulse)

เป็นสัญญาณที่ใช้เพื่อช่วยให้สัญญาณซิงค์ทางแนวตั้งยังคงมีรูปร่างดีเหมือนเดิมหลังจากแยกออกมาจากสัญญาณซิ้งค์ทางแนวนอนแล้ว นอกจากนี้ช่วยทำให้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สแกนแบบไขว้กันเป็นไปโดยเรียบร้อยสม่ำเสมอ รวมทั้งสัญญาณซิงค์ทางแนวนอนก็ไม่ขาดหายไปในช่วงเวลาของสัญญาณซิงค์ทางแนวตั้งอีกด้วย สัญญาณอีควอไลท์ซิ่งนี้มีความ เป็นสำหรับการสแกนแบบสลับเส้น(Interlaced scanning) เพราะช่วยลดความผิดพลาดที่เกิดขึ้น เนื่องจากการสะบัดกลับผิดตำแหน่ง ในโครงงานนี้การสร้างสัญญาณโทรทัศน์ ในขณะแสดงภาพของข้อมูลที่เก็บอยู่ในหน่วยความจำ จะไม่มีสัญญาณอีควอไลท์ซิ่งทั้งนี้เนื่องจากเป็นการสแกนแบบไปข้างหน้า(Progressive scanning) ซึ่งจะทำให้ระบบวงจรง่ายขึ้น และยังประหยัดค่าใช้จ่ายอีกด้วย



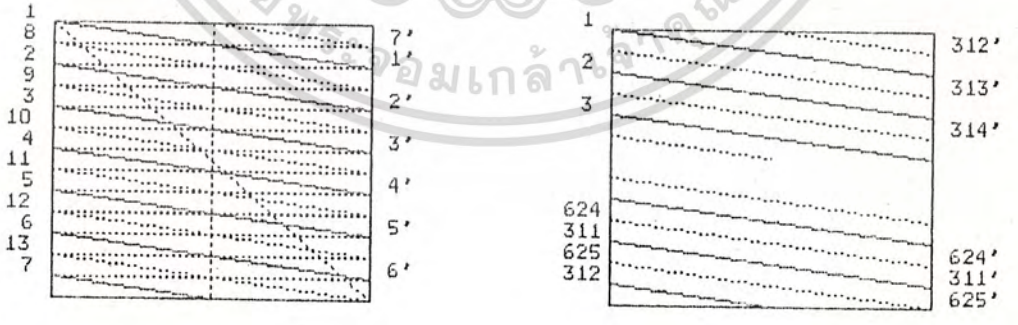
รูปที่ 2.7 สัญญาณภาพรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสแกน (Scanning) คือ การส่งจุดภาพไปยังจอมอนิเตอร์เพื่อที่จะแสดงภาพ ซึ่งจะเป็นจุดขาวดำหรือสีนั้นขึ้นอยู่กับสัญญาณภาพ การสแกนภาพมีอยู่ด้วยกัน 2 วิธี คือ การสแกนแบบไปข้างหน้า(Progressive scanning) และการสแกนแบบสลับเส้น(Interlaced scanning)

จากรูปที่ 2.8 จะเห็นได้ว่าการสแกนเริ่มจาก a-->b, c-->d จนกระทั่งสุดท้าย m-->n ซึ่งเป็นการสแกนจากซ้ายไปขวาและจากข้างบนลงข้างล่างเหมือนกับการอ่านหนังสือพิมพ์หรือการพิมพ์ดีด เรียกการสแกนแบบนี้ว่า การสแกนแบบไปข้างหน้า

รูปที่ 2.8 การสแกนแบบไปข้างหน้า



(ก) (ข)

รูปที่ 2.9 การสแกนแบบสลับเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9(ก) จะเห็นว่าการสแกนจะเริ่มจาก 1,2,..,7 เรียกว่า ฟิลด์คี่ (Odd Field) และในระหว่างเส้นต่อเส้นก็จะเว้นช่องว่างให้พอสแกนได้อีกครั้งหนึ่ง จากนั้นก็เริ่มที่ 8,9,..,13 ใหม่อีกครั้งหนึ่งเรียกว่า ฟิลด์คู่(Even Field) เป็นการสแกนแบบเส้นเว้นเส้นซึ่งต้องใช้การสแกนในแนวตั้ง 2 ครั้ง ดังนั้นถ้าต้องการภาพ 25 ภาพต่อวินาที ก็ต้องส่ง 50 ครั้ง ซึ่งวิธีนี้เราเรียกว่าการสแกนแบบสลับเส้น

ในการสแกนสลับเส้นนี้ต้องใช้การสแกนแนวตั้ง 2 ครั้ง การสแกนแนวตั้ง 1 ครั้งเรียกว่า การสแกน 1 ฟิลด์ การสแกน 2 ฟิลด์ เรียกว่า 1 เฟรม(frame) ในระบบ 625 เส้น จากรูปที่ 2.9(ข) การสแกน 1 ฟิลด์ มี 312.5 เส้น เนื่องจากตาของมนุษย์มีคุณสมบัติในการคงอยู่ของภาพ(Presistance of image) และระยะเวลาในการเรืองแสงของเฟอร์เฟอร์ที่จอภาพ ทำให้ภาพของฟิลด์หนึ่งยังคงอยู่ในขณะฟิลด์ที่สองสแกนเสร็จแล้ว ภาพที่มองเห็นจึงมีจำนวน 625 เส้น ข้อดีของวิธีนี้คือ ลดการกระพริบของของภาพ(flicker)

ในการสแกนทั้งแบบไปข้างหน้าและแบบสลับเส้น เมื่อสแกนไปสุดแต่ละเส้นแล้วต้องรีบกลับมาเริ่มเส้นใหม่ ทั้งแนวตั้งและแนวนอน ระยะเวลาในการวิ่งกลับมาเริ่มใหม่ยิ่งน้อยเท่าไรยิ่งดี จากรูปที่ 2.8 คือเส้นประจาก b-->c,d-->e เส้นนี้เรียกว่า เส้นสแกนกลับ (Retrace or Flyback) เส้นนี้ไม่มีความจำเป็นในการประกอบเป็นภาพ จึงมีวงจรควบคุมไม่ให้ปรากฏที่จอภาพ

ในโครงงานนี้จะทำการดิจิไตซ์เพียงฟิลด์เดียว จากสัญญาณที่ส่งมาโดยวิธีการสแกนแบบสลับเส้น ก่อนที่จะทำการเก็บภาพ จะเห็นว่าภาพที่แสดงบนจอโทรทัศน์จะเป็นการสแกนแบบสลับเส้น เมื่อทำการเก็บภาพแล้วภาพที่แสดงบนจอโทรทัศน์ จะเป็นภาพของข้อมูลที่เก็บอยู่ในหน่วยความจำ และแสดงผลโดยวิธีการสแกนแบบไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการแปลงสัญญาณอนาล็อกเป็นดิจิตอลและแปลงดิจิตอลเป็นสัญญาณอนาล็อก

ANALOG(A) TO DIGITAL(D) AND DIGITAL(D) TO ANALOG(A) CONVERTION

การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก เป็นการแปลงข้อมูลหรือข่าวสารทางดิจิตอลให้เป็นข้อมูลในทางอนาล็อก เช่นเอาท์พุทของระบบดิจิตอลจะต้องเปลี่ยนเป็นอนาล็อกเสียก่อนในการที่จะใช้ขับปากกาเรคคอร์ดเดอร์(recoder) โดยใช้ D to A converter หรือสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลเป็นสิ่งจำเป็นมาก สำหรับการประมวลผลทางดิจิตอล

ขบวนการของการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลนั้น ทำได้โดยใช้ A to D converter ซึ่งถูกใช้สำหรับแปลงสัญญาณอนาล็อกจากเอาท์พุท ของทรานสดิวเซอร์ (transducer) ไปเป็นสัญญาณดิจิตอลเพื่อใช้ป้อนเข้ายังระบบดิจิตอล การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก เป็นขบวนการที่ทำได้โดยตรง และทำได้ง่ายกว่าการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล

3.1 วงจรปรับค่าความต้านทาน (VARIABLE RESISTOR NETWORK)

ปัญหาพื้นฐานในการแปลงสัญญาณดิจิตอลไปเป็นสัญญาณอนาล็อกคือ การเปลี่ยนระดับของดิจิตอลโวลท์เตจ (digital voltage) ค่าใดๆไปเป็นอนาล็อกโวลท์เตจ (analog voltage)ที่เหมาะสม ซึ่งเราแก้ไขปัญหานี้ได้โดยการออกแบบวงจรปรับความต้านทาน ที่สามารถเปลี่ยนแต่ละระดับของสัญญาณดิจิตอลไปเป็น binary weight -equivalent voltage ตัวอย่างในการพิจารณาอะไรคือความหมายของequivalent binary weight ให้พิจารณาได้จากตารางความจริง(truth table) สำหรับสัญญาณไบนารี่ 3 บิท ดังรูปที่ 3.1 สมมุติว่าเราต้องการเปลี่ยนสัญญาณดิจิตอลทั้ง 8 ค่าดังในรูปเป็นอนาล็อกโลท์เตจ จำนวนน้อยที่สุดที่แสดงไว้ในรูปคือ "000"ที่ดิจิตอลค่านี้เราต้องการแปลงให้เป็นอนาล็อกโวลท์เตจเท่ากับ "0"โวลท์และจำนวนที่มากที่สุดคือ "111"ต้องการเปลี่ยนเป็นอนาล็อกเท่ากับ"7"โวลท์ การทำเช่นนี้เป็นการกำหนดย่านของสัญญาณอนาล็อกที่เราต้องการ จากตารางที่ 3.1 เราจะเห็นว่าระหว่างค่า"000"กับ"111" นั้นสามารถแบ่งออกเป็น 7 ระดับด้วยกัน ดังนั้นสัญญาณอนาล็อกที่ได้จะถูกแบ่งออกเป็น 7 ระดับเช่นกัน การเปลี่ยนแปลงที่เกิดขึ้นน้อยที่สุดของสัญญาณดิจิตอลอยู่ที่บิทต่ำสุด(LSB) ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทนี้จะทำให้การเปลี่ยนแปลงของอนาล็อคเอาท์พุทเท่ากับ 1/7 ของค่าเต็มสเกล (FULL SCALE) R DIVIDER ถูกกำหนดให้เป็น "1" ในตำแหน่ง 2^0 ซึ่งจะทำให้อนาล็อคเอาท์พุทเท่ากับ 7*1/7 = 1 โวลท์

2^2	2^1	2^0	DEC
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	0	1	7

รูปที่ 3.1 ตารางแสดงค่าไบนารี 3 บิท

พิจารณาที่บิท2^1เราจะเห็นได้ว่าที่บิท2^1มีค่าเป็นสองเท่าของบิท 2^0 ดังนั้น บิท 2^1 จะทำให้การเปลี่ยนแปลงของอนาล็อคเอาท์พุทโวลท์เตจเป็นสองเท่าของ LSB และค่า resistive divider จะทำให้อนาล็อคเอาท์พุทโวลท์เตจเท่ากับ 7*2/7 = 2 โวลท์ ในการพิจารณาที่บิทอื่นก็ใช้หลักการเดียวกัน

จากที่กล่าวมาแล้วข้างต้นจะเห็นว่าบิทต่ำสุดจะให้ weight เท่ากับ 1/7 บิท 2^1 เท่ากับ 2/7 ผลรวมของ weight ทั้งหมดจะเท่ากับ 1 โดยทั่วไปแล้ว binary weight ที่บิทต่ำสุดจะเขียนเป็นสมการได้คือ $1/2^{n-1}$ เมื่อ n คือจำนวนบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเราจะหาค่าของ resistive divider สำหรับดิจิตอล 3 บิท ดังรูปที่3.2

BIT	WEIGHT	BIT	WEIGHT
2^0	1/7	2^0	1/15
2^1	2/7	2^1	2/15
2^2	4/7	2^2	4/15
		2^3	8/15

รูปที่ 3.2 BINARY EQUIVALENT WEIGHTS

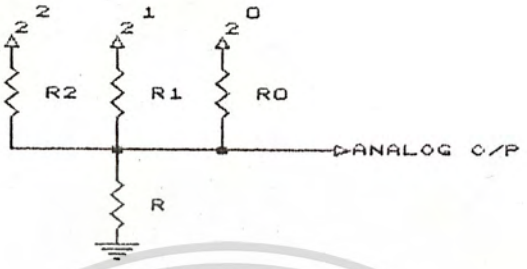
สมมุติดิจิตอลอินพุทมีค่า "000" ต้องการอนาล็อคเอาท์พุทเท่ากับ "0" โวลท์และ "111" เท่ากับ "7"เพราะฉะนั้นสำหรับอินพุท "001" จะได้เอาท์พุทเท่ากับ"1"โวลท์ ในทำนองเดียวกันเมื่ออินพุทเท่ากับ "010" จะได้อนาล็อคเอาท์พุทเท่ากับ "2" โวลท์ สำหรับค่าอื่นๆ ก็เช่นกันจะได้อนาล็อคเอาท์พุทตามตารางรูปที่ 3.2 resistive divider จะต้องทำหน้าที่สองอย่างด้วยกันเพื่อจะเปลี่ยนดิจิตอลอินพุทไปเป็นอนาล็อคเอาท์พุทโวลท์เตจ คือ

1.บิท 2 จะต้องเปลี่ยนเป็น 1 โวลท์ บิท 2^1 เท่ากับ 2 โวลท์ และบิท 2^2 เท่ากับ 4 โวลท์

2.โวลท์เตจทั้งสามค่านี้ที่แสดงดิจิตอลบิท เมื่อรวมกันแล้วจะได้อนาล็อคเอาท์พุทโวลท์เตจ

รูปที่ 3.3 แสดงถึง resistive divider ที่สมบูรณ์ของตัวอย่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

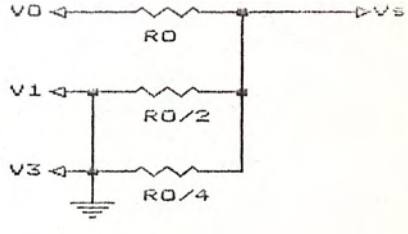


รูปที่ 3.3 RESISTIVE DIVIDER

R0 R1 และ R2 ต่อแบบ divider network ความต้านทาน R1 เป็นโหลดสมมุติว่าดิจิตอลอินพุทเท่ากับ"001" เราสามารถเขียนวงจรสมมูลย์ (equivalent circuit) ได้ดังรูปที่ 3.4 อนาล็อคเอาท์พุทโวลท์เตจ สามารถหาได้โดยใช้MILLMAN 's theorem ดังสมการต่อไปนี้

$$V = \frac{V1/R1 + V2/R2 + V3/R3 + \cdots\cdots\cdots\cdots\cdots}{1/R1 + 1/R2 + 1/R3 + \cdots\cdots\cdots\cdots\cdots}$$

โดยการใช้ MILLMAN 's theorem จากรูปที่ 3.4 จะได้



$$V = \frac{V0/R0 + V1/(R0/2) + V3/(R0/4)}{1/R0 + 1/(R0/2) + 1/(R0/4)}$$

สรุป resistive divider สามารถสร้างขึ้นเพื่อแปลงดิจิตอลโวลท์เตจไปอนาล็อคโวลท์เตจ โดยใช้กฎเกณฑ์ต่อไปนี้

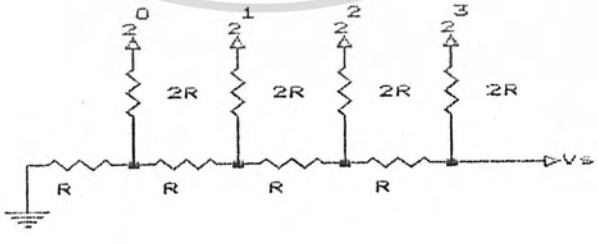
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ต้องมีความต้านทานอินพุทหนึ่งตัวสำหรับแต่ละบิทของดิจิตอล
2. เริ่มต้นที่บิทต่ำสุดค่าความต้านในบิทต่อมา มีค่าเป็นครึ่งหนึ่งของความต้านทานของบิทก่อนหน้านั้น
3. ค่าเต็มสเกลของเอาท์พุทโวลท์เตจ มีค่าเท่ากับผลบวกของสัญญายดิจิตอลอินพุท
4. บิทต่ำสุดจะมี weight เท่ากับ $1/2^{n-1}$ เมื่อ n คือจำนวนบิท
5. การเปลี่ยนแปลงของอินพุทโวลท์เตจ ต่อการเปลี่ยนแปลงของบิทต่ำสุดเท่ากับ $V/2^{n-1}$
6. เอาท์พุทโวลท์เตจ สามารถหาได้โดยใช้ MILLMAN 's theorem

$$V_a = V0.2^0 + V1.2^1 + V2.2^2 + \ldots\ldots\ldots Vn-1.2^{n-1}$$

3.2 BINARY LADDER

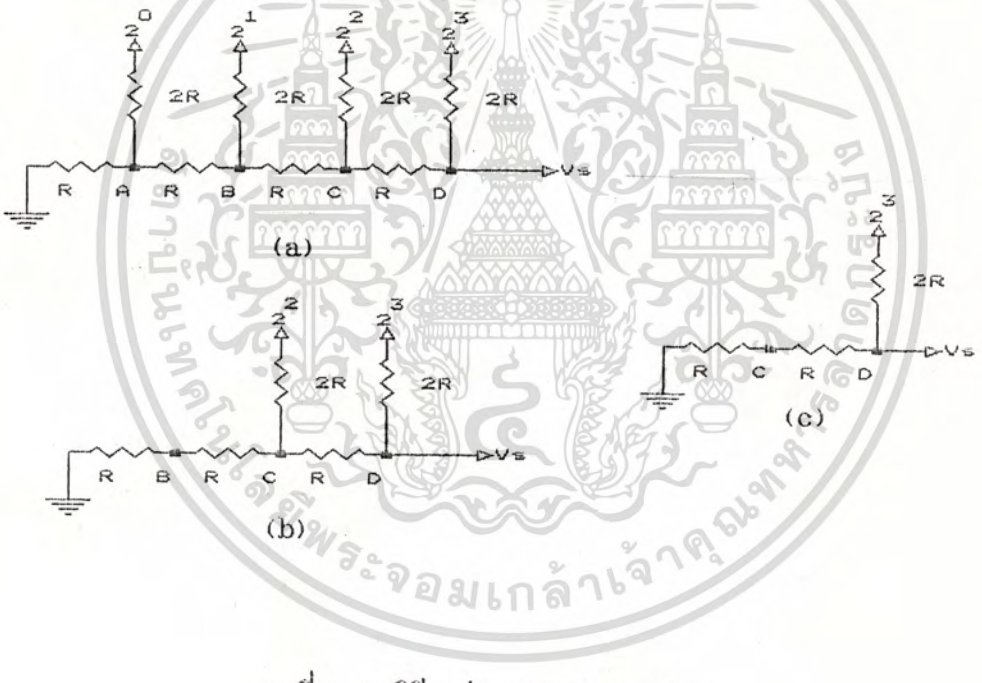
binary ladder เป็น network ที่ output voltage เป็นผลรวม weight ของดิจิตอลอินพุทเราลองมาพิจารณาladderขนาด 4 bit ดังรูปที่ 3.5



รูปที่ 3.5 BINARY LADDER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

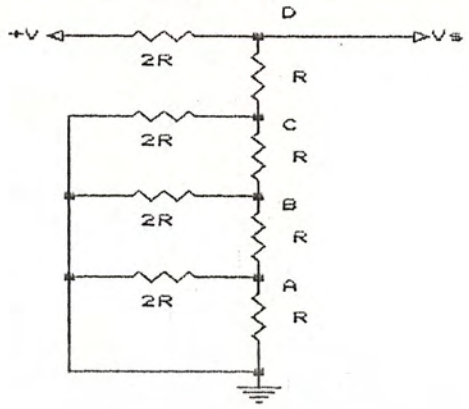
จากรูปจะเห็นว่าใช้ resistor เพียงสองค่า สมมุติว่าดิจิตอลทั้งหมดถูกต่อลงกราวด์ เริ่มต้นที่จุด A ค่าความต้านทานที่มองที่จุด A ไปยัง resistor ตัวซ้ายสุดคือ 2R ค่าความต้านทานที่มองไปยังบิท 2^0 มีค่าเป็น 2R เช่นกัน ซึ่งค่าความต้านทาน 2R ทั้งสองตัวนี้จะขนานกันอยู่ ด้วยเหตุนี้เราจึงสามารถแทนด้วยความต้านสมมูลย์ ซึ่งมีค่าเท่ากับ R ดังรูป 3.6 (a) ต่อไปเราพิจารณาที่จุด B จะเห็นว่าค่าความต้านทานที่จุด B นี้มีค่าเป็น 2R และค่าความต้านทานของบิท 2^1 เท่ากับ 2R เช่นกัน ซึ่งสามารถแทนด้วยความต้านทานสมมูลย์ R ดังรูปที่ 3.6(b)



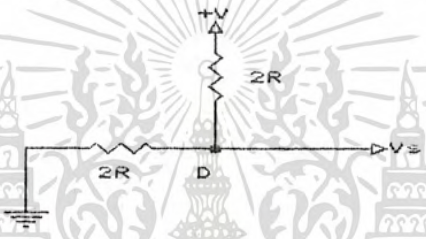
รูปที่ 3.6 วิธีหาค่า BINARY LADDER

เราสามารถใช้คุณสมบัติของ ladder เพื่อกำหนดเอาท์พุทของโวลท์เตจสำหรับดิจิตอลเอาท์พุทค่าต่างๆได้ สมมุติดิจิตอลอินพุทเป็น 1000 binary ladder สามารถเขียนได้ดังรูปที่ 3.7 (a) เพราะฉะนั้น network ที่อยู่ด้านซ้ายของจุด D สามารถแทนด้วยความต้าน 2R ดังรูป 3.7(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 3.7 (a) BINARY LADDER WITH A DIGITAL INPUT 1000

(b) EQUIVALENT CIRCUIT FOR DIGITAL INPUT 1000

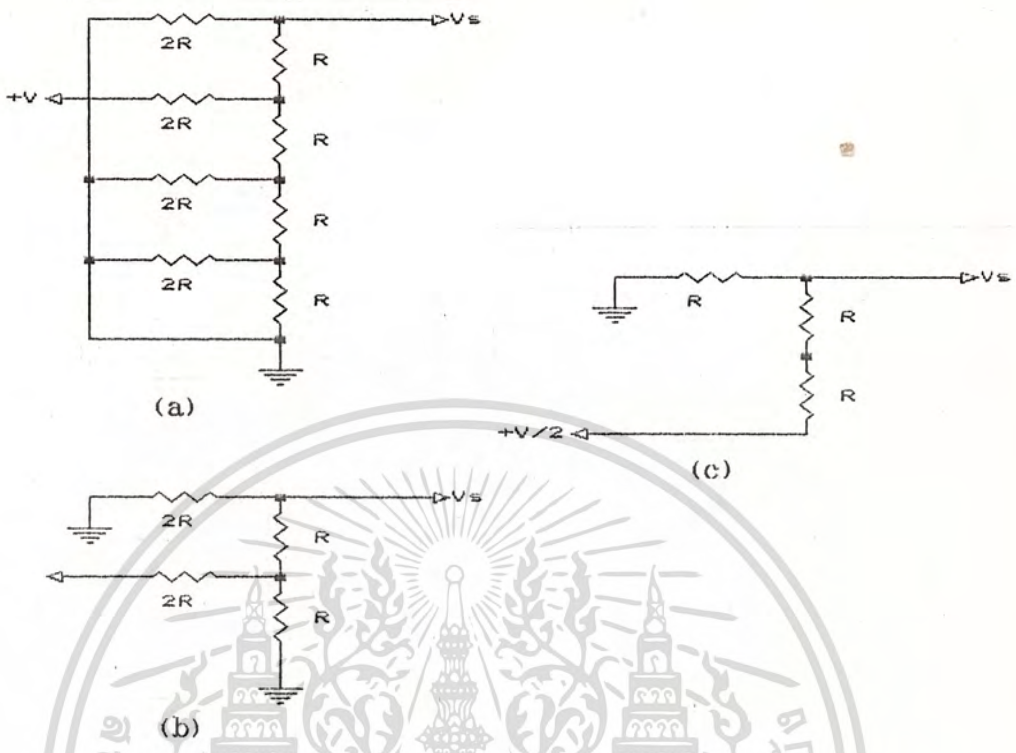
จากรูปเราจะได้

$$V_o = V.[2R/(2R+2R)]$$

$$= V/2$$

นั่นคือถ้าบิทสูงสุด(MSB) มีค่าลอจิก(logic)เป็น 1 จะได้ $V_o = V/2$ โวลท์ ในการหาเอาท์พุทโวลท์เตจเมื่ออินพุทเป็น 0100 สามารถทำได้ดังแสดงในรูปที่ 3.8(a) จะเห็นว่าไม่มีโวลท์เตจทางด้านซ้ายมือของจุด c เลย เพราะฉะนั้นทางด้านซ้ายมือของจุด c สามารถแทนด้วยความต้าน 2R ดังรูป 3.8 (b) ต่อไปเราสามารถแทนที่วงจรทางด้านซ้ายของจุด c โดยใช้ทฤษฎีของTHEVENIN's theoremโดยการตัดวงจรดังรูป 3.8 (b) จะได้วงจรใหม่ดังรูป 3.8 (c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

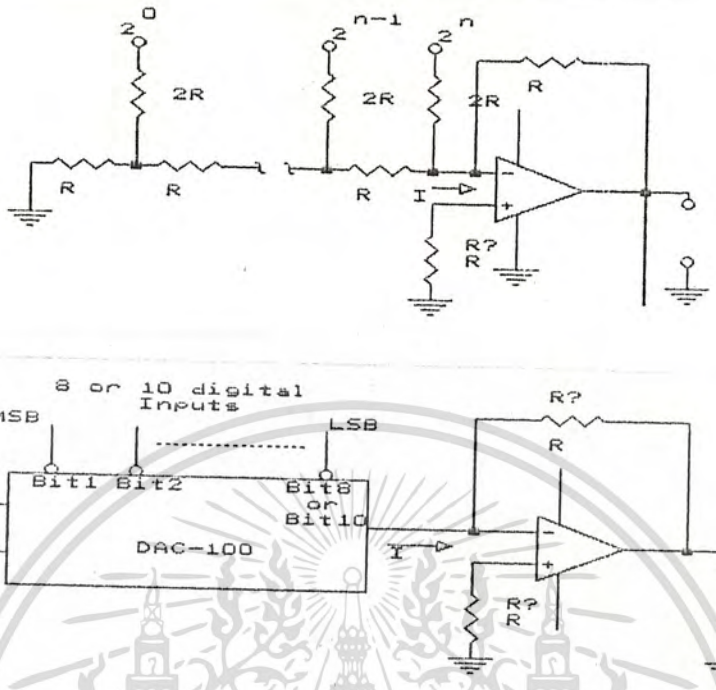


รูปที่ 3.8 การใช้ THEVENIN 'S THEOREM เข้าช่วยแก้ไขวงจร

3.3 ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อค (D TO A CONVERTER)

เราสามารถใช้วงจรพื้นฐานของวงจร resistive divider or ladderมาทำเป็นDAC ได้หรืออาจกล่าวได้ว่า DAC มีส่วนประกอบพื้นฐานหรือโครงสร้างที่ประกอบขึ้นจาก resistive dividernetwork หรือ ladder อย่างไรก็ตาม DAC ต้องการส่วนปลีกย่อยอื่นๆอีกเพื่อที่จะทำให้ DAC ที่สมบูรณ์ขึ้นคือ DAC จะต้องมีส่วนของรีจีสเตอร์(register)ที่สามารถใช้เก็บข้อมูลทางดิจิตอล register ที่ง่ายที่สุดจะใช้ฟลิบฟลอบ(flip-flop)ซึ่งสำหรับ1บิทจะใช้ฟลิบฟลอบ 1 ตัว DAC ต้องมี level amplifier อยู่ระหว่าง register กับ resistive network เพื่อให้แน่ใจได้ว่าสัญญาณดิจิตอลที่ส่งไปยัง resistive network เป็นสัญญาณที่คงที่อยู่ในระดับเดียวกัน ส่วนสุดท้ายของ DACจะต้องมีส่วนอินพุทเกทเพื่อทำการ gatingอินพุทของ register ฟลิบฟลอบสามารถถูกเซ็ทให้มีค่าข้อมูลที่เหมาะสมกับข้อมูลดิจิตอลที่เข้ามา DAC ที่สมบูรณ์แสดงดัง block diagram ดังรูปที่ 3.10 (a)

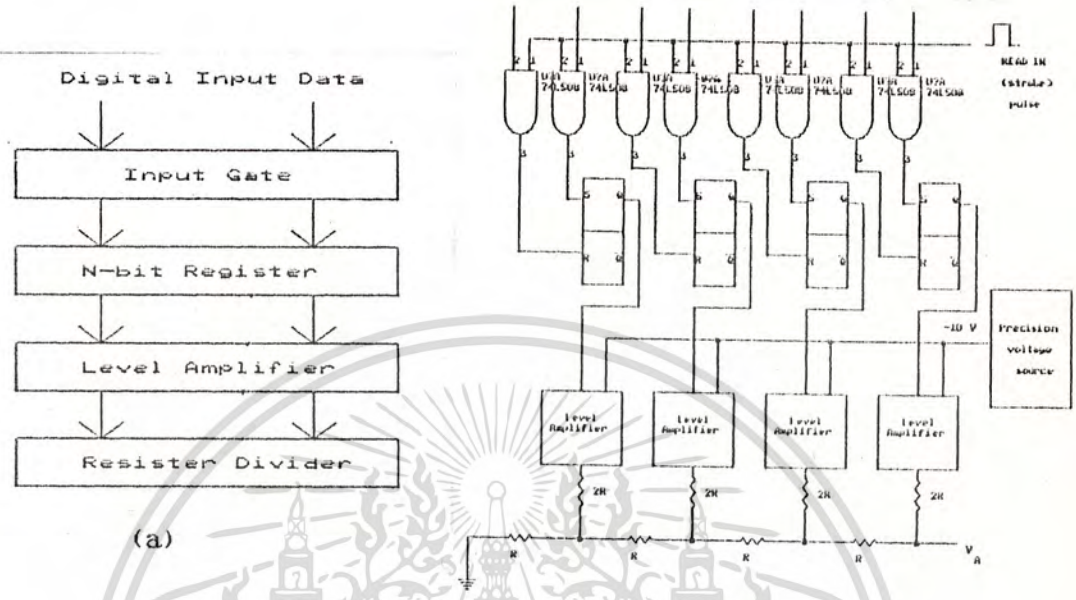
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 D TO A CONVERTER

อาศัยหลักการที่แสดงโดย block diagram เราสามารถเขียนวงจร D to A ที่สมบูรณ์ขนาด 4 บิทได้ดังรูปที่ 3.10(b)จากรูปจะเห็น level amplifier แต่ละตัวจะมีอินพุท 2อินพุทคือ 10 โวลท์ จาก precision voltage sourceและจากเอาท์พุทของฟลิบฟลอบ level amplifier จะทำงานดังนี้คือ ถ้าเอาท์พุทจากฟลิบฟลอบเป็น hight level amplifier จะให้เอาท์พุทเป็น 1 แต่ถ้าเอาท์พุทจากฟลิบฟลอบเป็น low เอาท์พุทจะเป็น 0 ฟลิบฟลอบทั้ง 4 ที่ใช้เป็น register จะใช้สำหรับเก็บข้อมูลทางดิจิตอลฟลิบฟลิบที่อยู่ทางขวามือเป็น MSB และตัวซ้ายสุดเป็น LSB ฟลิบฟลอบแต่ละตัว จะเป็น RS-latch และต้องการระดับสัญญาณที่เป็นบวก(positive level) ที่อินพุท R หรือ S เพื่อที่จะไปเซ็ทหรือรีเซ็ทเอาท์พุทของฟลิบฟลอบ ส่วนของอินพุทเกทใช้สำหรับข้อมูลเข้ามายัง register เมื่อสาย read-in เป็น high จะมีเกทเพียงตัวเดียวเท่านั้นที่มีเอาท์พุทเป็น high ส่งไปยังอินพุท R หรือ S ฟลิบฟลอบจึงถูกเซ็ทโดยขึ้นอยู่กับค่าลอจิกของ R หรือ S ข้อมูลจะถูกส่งเข้าไปยัง register ในแต่ละครั้งที่มีพัลซ์(pulse)ป้อนเข้ามาที่สาย read-in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 (a)BLOCK DIAGRAM

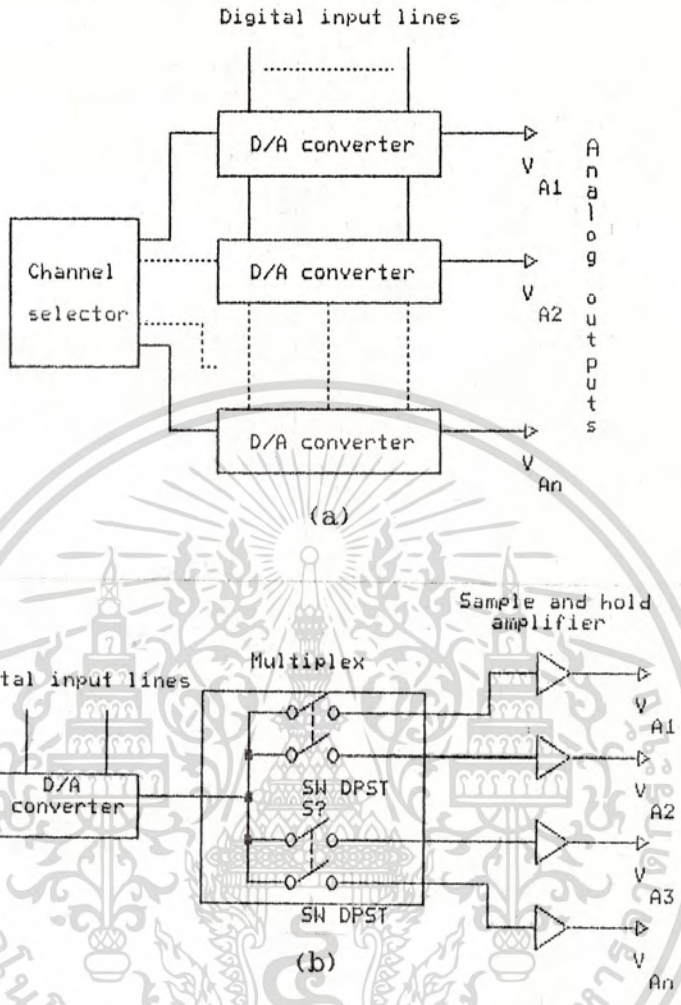
(b)4-BIT D TO A CONVERTER

บ่อยครั้งที่เดียวที่เราจำเป็นต้อง decode สัญญาณอินพุทที่มีหลายๆ อินพุทเช่น แกน X แกน Y สำหรับ plotting board ในกรณีเช่นนี้จะมี 2 วิธีในการ decode สัญญาณคือ

1. โดยใช้DtoA converter สำหรับแต่ละสัญญาณอินพุทดังแสดงไว้ดังรูปที่ 3.11(a)ข้อดีของวิธีนี้คือแต่ละสัญญาณที่จะ decode จะถูกเก็บไว้ใน register ของมันเอง และอนาล็อคเอาท์พุทจะถูก hold ไว้ให้มีค่าคงที่ สายดิจิตอลอินพุทถูกต่อขนานเข้ากับแต่ละconverter และconverter ตัวที่เหมาะสมจะถูกเลือกสำหรับ decode โดยสาย select (select line)

2.เป็นการ multiplex แสดงดังรูปที่ 3.11(b) ข้อเสียของวิธีนี้คือ อนาล็อคเอาท์พุทจะถูกholdในระหว่างของการ sampling และเอาท์พุทจะต้องต่อกับ sample-hold amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

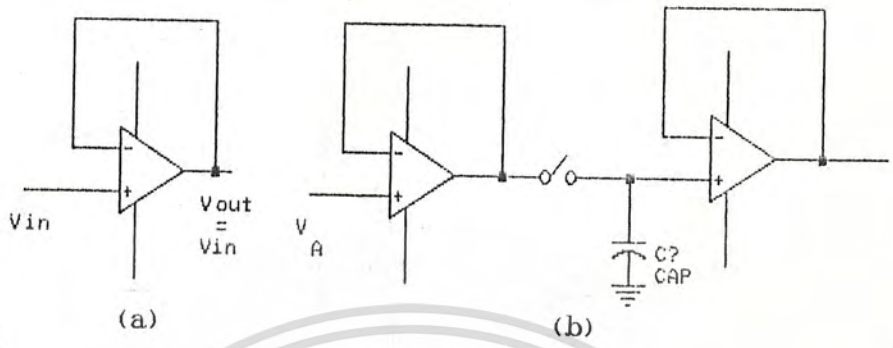


รูปที่ 3.11(a) CHANNEL SELECT METHOD

(b) MULTIPLEX METHOD

จากรูปที่ 3.12(a) เป็นแบบ unity gain noninverting voltage amplifier ซึ่งจะได้ Vo= Vi และถ้านำเอาออปแอมป์แบบนี้มาต่อกันดังรูปที่ 3.12(b) ก็จะได้เป็น sample-hold amplifier เมื่อสวิทช์ปิดวงจร คาปาซิเตอร์จะชาร์จประจุอนาล็อคโวลท์เตจที่ได้จาก D to A ไว้ และเมื่อสวิทช์เปิดวงจร คาปาซิเตอร์ hold ระดับโวลท์เตจไว้ จนกระทั่งถึงเวลาของการ sampling ใหม่และเพราะว่าออปแอมป์มีอินพุทอิมพีแดนซ์สูงมาก ดังนั้นจะเห็นได้ว่าคาปาซิเตอร์จะได้ดีสชาร์จ (discharge) และในเวลาเดียวกันก็จะให้ gain เพื่อขับวงจรภายนอกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 (a)UNITY GAIN AMPLIFIER
(b)SAMPLE-HOLD CIRCUIT

เมื่อ D/A ถูกต่อเข้ากับ multiplexer เราจะต้องคำนึงถึงอัตราสูงสุดที่converter จะสามารถทำงานได้ด้วย ในแต่ละครั้งที่ข้อมูลถูกส่งไปยังregister จะเกิด transient ขึ้นที่เอาท์พุทของตัว converter ทั้งนี้เพราะว่าในความเป็นจริงแล้วฟลิบฟลอบแต่ละตัวมี rise และ fall time ที่แตกต่างกันดังนั้นจะต้องมี setting time ระหว่างเวลาที่ข้อมูลถูกส่งไปยัง register กับเวลาที่อนาล็อคโวลท์เตจถูกอ่านออกมา setting time นี้เป็นตัวประกอบหลักในการกำหนดอัตราสูงสุดของการ multiplex กรณีเช่นนี้จะเกิดขึ้นเมื่อทุกบิทมีการเปลี่ยนแปลงเช่นจาก1000 เป็น 0111 โดยปกติแล้วคาปาซิเตอร์ของวงจร sample-hold amplifierจะไม่มีความสามารถในการ hold voltage ที่แน่นอน ดังนั้นอัตราการ sampling (SAMPLING RATE)จะต้องเพียงพอเพื่อให้มั่นใจได้ว่าโวลท์เตจไม่ถูกลดทอนหรือหายไปในระหว่างการ sampling อัตราการ sampling คือความถี่ที่คาปาซิเตอร์ชาร์จ (charge)ค่าของสัญญานอนาล็อคที่ออกจากเอาท์พุทของ converter

มาถึงจุดนี้แล้วเราอาจจะอยากรู้ว่าสัญญาณจะต้องถูกsampling เท่าไรจึงจะเพียงพอโดยอาศัยความรู้สึก(common sense)เราจะรู้ว่าจำนวนครั้งหรือความถี่ในการ sampling จะต้องมากกว่าความถี่ของสัญญาณที่ถูก sampling และความถี่ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

samplingที่น้อยความถี่ของสัญญาณที่ถูก sampling จะทำให้สัญญาณถูกลดทอนไปหรือ อาจจะกล่าวได้ว่าถ้าความถี่ของสัญญาณที่ถูก sampling น้อยเกินไปสัญญาณจะลดลงมาก เพราะเวลาของการ discharge ของ sampling-hold capacitor มากเกินไป และอาจจะทำให้ข้อมูลของสัญญาณสูญหายไป เรามีวิธีการสองอย่างในการกำหนดอัตรา การ sampling คือ

1.ถ้าสัญญาณของเราเป็นรูปซายน์ (sinusoidal) ต้องทำการ samplingเป็นสองเท่าของความถี่ของสัญญาณที่เราจะทำการ sampling เช่นถ้าสัญญานเป็นsinewave 5KHz เราต้อง sampling ด้วยอัตราที่มากกว่าหรือเท่ากับ 10KHz หรือทุกๆ 1/1000 = 100 uS

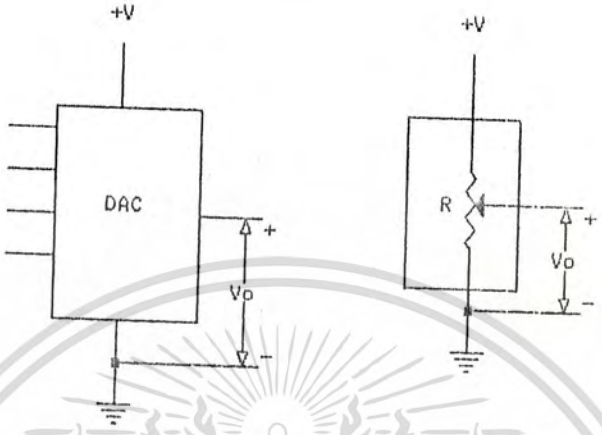
อย่างไรก็ตามถ้าสัญญาณเป็นแบบ sine เราจะทำอย่างไร สัญญาณใดๆก็ตามที่เป็นสัญญาณรายคาบ (periodic) ที่สามารถแสดงได้โดยผลรวมของเทอมโคซายน์ (cosine) ในกรณีนี้เราจะต้อง sampling ด้วยอัตราสองเท่าของความถี่สูงสุดที่ปรากฏอยู่ในเทอมของโคซายน์

2. โดยวิธีการทดสอบซึ่งจะตรวจสอบการทำงานที่เหมาะสมของD/Aได้ซึ่ง แบ่งออกเป็น steady-state accuracy test และ monotomicity test

การทดสอบแบบ seatdy-state เป็นการป้อนค่าดิจิตอลอินพุทเป็นอินพุท register ของ D/A จากนั้นก็จะทำการวัดอนาล็อคเอาท์พุท ที่ได้แล้วเปรียบเทียบค่าที่วัดได้จากการคำนวณว่าตรงกันหรือไม่

การทดสอบแบบ monotomicity หมายถึงการตรวจสอบค่าเอาท์พุทโวลท์เตจ (analog output voltage) ว่าเพิ่มขึ้นตามค่าดิจิตอลและสังเกตอนาล็อคเอาท์พุทจากออสซิลโลสโคป(oscilloscope) สำหรับ monotomicity ที่เหมาะสม สัญญาณเอาท์พุทจะต้องมีรูปร่างดังรูปที่ 3.13 ขั้นต่างๆของสัญญาณรูปขั้นบันไดจะต้องมีช่องว่างและแอมปลิจูด(amplitude)เท่ากัน ขั้นที่ไม่ถูกต้องเช่น ขั้นที่มีแอมปลิจูดต่างกันหรือขั้นที่ตกลงมาข้างล่างจะแสดงถึง malfunction การทดสอบแบบ monotomicity นี้ไม่ได้ตรวจสอบความเที่ยงตรง(accuracy)แต่ถ้าผ่านการทดสอบแบบนี้แล้วก็เป็นการประกันได้ว่าค่าผิดพลาด (error)ในการแปลงสัญญาณ(convert.)จะต่ำกว่า 1 LSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 CORRECT OUTPUT VOLTAGE WAVEFORM FOR MONOTOMICITY TEST

3.4 ความละเอียดและความเที่ยงตรงของ D TO A

ความเที่ยงตรง(accuracy)ของ D/A คอนเวอร์เตอร์ขึ้นอยู่กับเที่ยงตรงของ precision resistor ที่ใช้เป็นแลดเดอร์และความแม่นยำ(precision)ของแรงดันอ้างอิง(reference voltage supply)ที่ใช้ความเที่ยงตรงเป็นการวัดค่าเอาท์พุทที่ได้ในทางปฏิบัติว่ามีค่าใกล้เคียงกับค่าทฤษฎีเพียงใด เช่นสมมุติว่าเอาท์พุททางทฤษฎีมีค่าเท่ากับ 10 V.ความเที่ยงตรง 10% จะหมายถึงเอาท์พุทที่ได้ในทางปฏิบัติอยู่ระหว่าง 9.9 V.และ 10.1 V

ความละเอียด (resolution) หรือจะกล่าวอีกอย่างหนึ่งคือการหาโวลท์เตจเพิ่มขึ้นอย่างน้อยที่สุดที่สังเกตได้ ความละเอียดจะขึ้นอยู่กับจำนวนบิทของดิจิตอลอินพุทนั่นคือเอาท์พุทโวลท์เตจเพิ่มขึ้นอย่างน้อยที่สุดถูกกำหนดโดย LSB สมมุติว่าในระบบ 4 บิท ที่ใช้แลดเดอร์ที่ LSB มี weight เป็น 1/16 ของอินพุทโวลท์เตจและถ้าสมมุติว่าอินพุทโวลท์เตจของเรามีค่าเป็น 16 ดังนั้นในการเปลี่ยนแปลงของ LSB จะทำให้เอาท์พุทโวลท์เตจเปลี่ยนแปลงเป็น 16*1/16 = 1 V. ดังนั้นเอาท์พุทโวลท์เตจจะเปลี่ยนแปลงเป็นขั้นๆ หรือเพิ่มขึ้นทีละ 1 V.ซึ่งจะได้เอาท์พุทโวลท์เตจจะเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงเป็นรูปขั้นบันไดดังรูปที่ 3.13 และจะเปลี่ยนแปลงจาก 0-15 V โดยเพิ่มขึ้นทีละ1 V. คอนเวอร์เตอร์ อย่างไรก็ตามมันไม่สามารถใช้กับโวลท์เตจที่เพิ่มขึ้นน้อยกว่า 1โวลท์ เช่นถ้าเราต้องการเอาท์พุทโวลท์เตจเท่ากับ 4.2 V. โดยใช้เอาท์พุทเท่ากับ 7.8 V. เราจะได้เอาท์พุทเป็น 8 V. ทั้งนี้เพราะว่า คอนเวอร์เตอร์ของเราเพิ่มค่าขึ้นขั้นละ 1 V. เท่านั้นกล่าวได้ว่า คอนเวอร์เตอร์ของเรามีค่าความละเอียดแค่ 1 โวลท์ เท่านั้น

ถ้าเราต้องการเอาท์พุทที่มีการเปลี่ยนแปลงน้อยกว่าที่กล่าวมาแล้ว เราต้องใช้คอนเวอร์เตอร์ที่มีอินพุทมากว่านี้ เช่นคอนเวอร์เตอร์ขนาด 10 บิท จะมี LSB weight เท่ากับ $1/2^{10} = 1/1024$ ดังนั้นค่าที่เปลี่ยนแปลงที่น้อยที่สุดประมาณ 1/1000 และถ้าคอนเวอร์เตอร์มีค่าเต็มสเกลเท่ากับ 10 V. เราจะได้การเปลี่ยนแปลงของเอาท์พุทขั้นละ 10/1000 = 10 มิลลิโวลท์

3.5 <u>ตัวแปลงสัญญาณอนาล็อคเป็นสัญญาณดิจิตอล</u> (A TO D CONVERTER)

ขบวนการในการแปลงสัญญาณอนาล็อคเป็นดิจิตอลเราเรียกว่า analog to digital conversion (ADC) ซึ่งจะมีบางสิ่งบางอย่างที่ซับซ้อนกว่าการแปลงสัญญาณดิจิตอลไปเป็นอนาล็อค ซึ่งจะมีวิธีการต่างๆอยู่หลายวิธี วิธีที่ง่ายที่สุดคือ

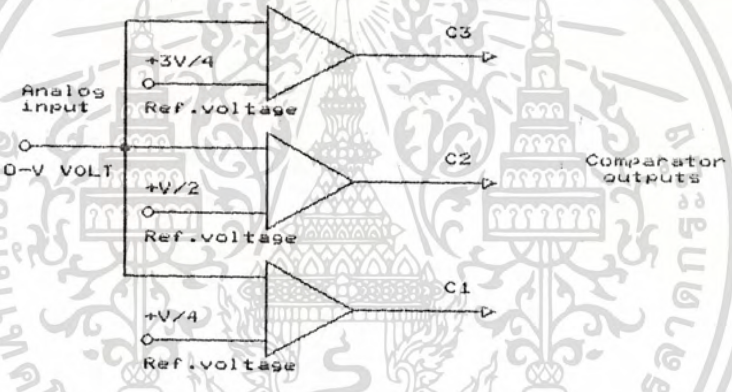
1. วิธีการของ simultaneous จะได้วงจรเปรียบเทียบ (comparator circuit) เช่นดังรูปที่ 3.14 จะใช้คอมพาราเตอร์ 3 ตัว อนาล็อคอินพุทที่เราต้องการแปลงเป็นดิจิตอล จะถูกป้อนเข้าที่อินพุทของคอมพาราเตอร์แต่ละตัว และขาอินพุทของคอมพาราเตอร์แต่ละตัวที่เหลือ ถูกต่อเข้ากับแรงดันอ้างอิงมาตรฐาน(standard reference voltage) แรงดันอ้างอิงที่เหลือที่ใช้จะเป็น V/4 ,V/2 และ 3V/4 ระบบนี้จะใช้สำหรับอนาล็อคอินพุทระหว่าง 0-V โวลท์

ถ้าอนาล็อคอินพุทมีค่าเกินแรงดันอ้างอิงของคอมพาราเตอร์ตัวใดๆ ก็ตามจะทำให้คอมพาราเตอร์ตัวนั้นทำงาน(TURN ON) ในการทำงานของคอมพาราเตอร์สมมุติให้เอาท์พุทเป็น HIGH ถ้าคอมพาราเตอร์ทุกตัวไม่ทำงานแสดงว่าอนาล็อคอินพุทมีค่าอยู่ระหว่าง 0-V/4 ถ้า C_1 เป็น high (comparator C_1 on)และ C_2 กับ C_3 เป็น low อินพุทจะมีค่าอยู่ระหว่าง V/2 กับ V/4 ถ้า C_1 กับ C_2 เป็น HIGH ส่วน C_3 เป็น low อินพุทมีค่าอยู่ระหว่าง V/2 กับ 3V/4 และสุดท้ายถ้าเอาท์พุทของคอมพาราเตอร์เป็น high หมด สัญญาณอินพุทอยู่ระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3V/4 กับ V สำหรับค่าอินพุทของคอมพาราเตอร์ในย่านต่างๆของอินพุทจะเป็นไปตามรูปที่ 3.14

ตัวอย่างที่แสดงให้เห็นดังรูปที่ 3.14 นั้นจะมีอยู่ด้วยกัน 4 ย่าน ที่คอนเวอร์เตอร์ตัวนี้สามารถตรวจจับ(detect)ได้ ซึ่งทั้ง 4 ย่านนี้สามารถแทนได้ด้วยไบนารี 2 บิท คอมพาราเตอร์ทั้งสามตัวจะต้องส่งเอาท์พุทไปยัง coding network เพื่อกำหนด 2 บิทที่สมบูรณ์กับอนาล็อคอินพุทโวลท์เตจ บิทต่างๆของ coding network จะถูกส่งไปฟลิปฟลอบรีจีสเตอร์เพื่อเก็บค่าไว้ block diagram ที่สมบูรณ์สำหรับตัวอย่างของ A/D ที่ได้กล่าวมานี้แสดงดังรูปที่ 3.15

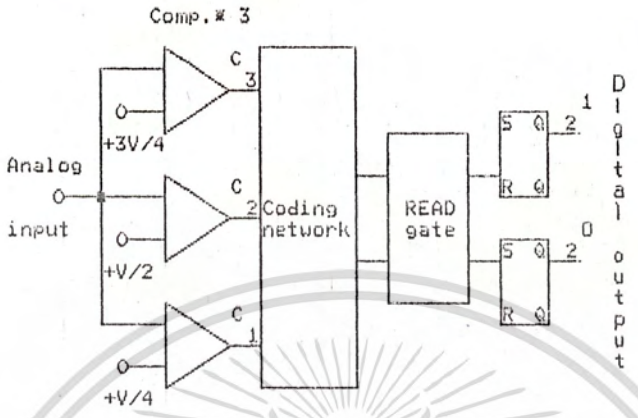


รูปที่ 3.14 SIMULTANEOUS A/D CONVERSION

INPUT VOLTAGE	COMPARATOR OUTPUT		
	C_1	C_2	C_3
0 - V/4	LOW	LOW	LOW
V/4 - V/2	HIGH	LOW	LOW
V/2 - 3V/4	HIGH	HIGH	LOW
3V/4 - V	HIGH	HIGH	HIGH

รูปที่ 3.14 COMPARATOR OUTPUTS FOR INPUT VOLTAGE RAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 2-BIT SIMULTANEOUS A/D CONVERTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

หลักการจัด DMA ของระบบ

ภายในระบบคอมพิวเตอร์ที่มีการเชื่อมต่อกับวงจรหรืออุปกรณ์ภายนอกนั้น มักจะมีความจำเป็นต้องการรับ/ส่งข้อมูลระหว่างวงจรหรืออุปกรณ์ภายนอกกับหน่วยความจำของระบบด้วยอัตราความเร็วที่สูงเกินกว่าที่จะใช้วิธีการทางด้านซอฟท์แวร์ได้ (โดยทั่วไปทางซอฟท์แวร์จะใช้โปรแกรมที่มีลักษณะเป็น Loop ซึ่งใช้คำสั่ง IN และ OUT ในการอ่านและเขียนข้อมูลบนพอร์ท) เช่น การรับส่งข้อมูลระหว่างคอมพิวเตอร์กับดิสค์ไดรฟ์ ซึ่งอัตราการรับ/ส่งข้อมูลของดิสค์ไดรฟ์นั้น จะมีอัตราเร็วที่สูงมากจนในบางครั้งทำให้ CPU ไม่สามารถที่จะทำหน้าที่ในการส่งหรือรับข้อมูลจากดิสค์ไดรฟ์ได้ทัน โดยเฉพาะกรณีที่ CPU จำเป็นต้องคอยตอบสนองต่ออุปกรณ์อื่นของระบบเช่น คีย์บอร์ดอยู่ด้วย

สำหรับเครื่อง IBM/PC นั้น ก็มีปัญหาในลักษณะเดียวกับที่ได้กล่าวมาแล้วเช่นกัน ซึ่งทำให้ภายใน IBM/PC ต้องอาศัยเทคนิคการทำ DMA หรือ Driect Memory Access เข้าช่วยในการแก้ปัญหาที่เกี่ยวกับอัตราเร็วในการรับส่งข้อมูลกับอุปกรณ์ภายนอก การทำ DMA นี้จะเป็นลักษณะของการรับ/ส่งข้อมูลระหว่างหน่วยความจำของระบบกับอุปกรณ์ภายนอกโดยตรง กล่าวคือ เป็นการรับ/ส่งข้อมูลที่ไม่จำเป็นต้องอาศัย CPU ของระบบเลย ซึ่งถ้าหากเป็นการรับ/ส่งข้อมูลตามปกติแล้ว จะเป็นการกระทำโดย CPU ทั้งหมด คือ ถ้าเป็นขบวนการส่งข้อมูลจากหน่วยความจำให้กับอุปกรณ์ภายนอก CPU จะทำการอ่านข้อมูลจากหน่วยความจำเข้ามาเก็บไว้ก่อนจากนั้นจึงทำการส่งออกไปให้กับอุปกรณ์ภายนอก และในทางกลับกันถ้าเป็นการส่งข้อมูลจากอุปกรณ์ภายนอกให้กับหน่วยความจำ CPU ก็จะอ่านข้อมูลจากอุปกรณ์ภายนอกเข้ามาเก็บไว้ก่อนจากนั้นจึงทำการส่งออกไปให้กับหน่วยความจำของระบบ จะเห็นได้ว่าในการส่งผ่านข้อมูลแต่ละไบท์นั้น จะต้องมีการรับ/ส่งข้อมูลถึง 2 ครั้ง (จากต้นทางไปยัง CPU และจาก CPU ไปยังปลายทาง) ในขณะที่ขบวนการ DMA นั้นใช้การส่งข้อมูลจากต้นทางไปปลายทางโดยตรง ซึ่งจะประหยัดเวลาลงได้มาก

สำหรับการทำ DMA ของ IBM/PC นั้น จะใช้ชิพซัพพอร์ทที่ทำหน้าที่ในการสร้างขบวนการ DMA โดยตรง คือ DMA Controller เบอร์ 8237A-5 ดังนั้นจะกล่าวถึงการโปรแกรมและใช้งาน 8237A-5 ภายใน IBM/PC โดยละเอียด

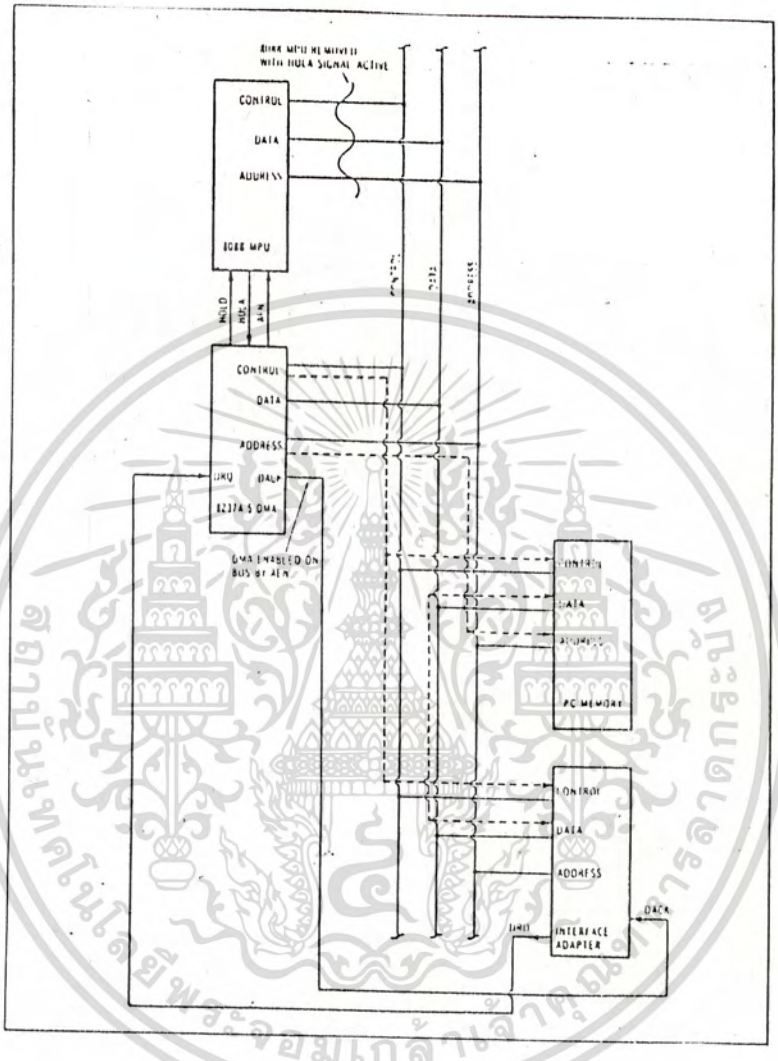
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การเกิดขบวนการ DMA ภายใน IBM/PC

ในการทำงานโดยปกตินั้น 8088 จะทำการควบคุมบัสต่าง ๆ ของระบบเองโดยการส่งแอดเดรสของหน่วยความจำหรือพอร์ทต่าง ๆ ที่ 8088 ต้องการจะติดต่อด้วยออกมาบนบัสแอดเดรส, ส่งสัญญาณควบคุมต่าง ๆ ที่จำเป็นออกมาบนบัสควบคุม (Control Bus) และรับหรือส่งข้อมูลให้กับหน่วยความจำหรือพอร์ทต่าง ๆ โดยผ่านทางบัสข้อมูล เมื่อวงจรหรืออุปกรณ์ภายนอกมีความต้องการที่จะส่งผ่านข้อมูลโดยใช้ขบวนการ DMA อุปกรณ์หรือวงจรนั้นก็จะส่งสัญญาณให้กับ DMA Controller เพื่อขอใช้ขบวนการ DMA หลังจากได้รับสัญญาณนี้แล้ว DMA Controller ก็จะตรวจสอบลำดับความสำคัญของสัญญาณที่ได้รับจากอุปกรณ์ หรือวงจรนี้กับสัญญาณที่ได้รับจากอุปกรณ์หรือวงจรที่ทำการขอ DMA ในแชนแนลอื่นของ DMA-Controller จากนั้นจึงส่งสัญญาณให้กับ 8088 เพื่อขอให้บัสในการส่งผ่านข้อมูล (โดยความจริงสัญญาณนี้ไม่ได้ถูกส่งให้กับ 8088 โดยตรง แต่จะถูกส่งให้กับวงจรสร้าง Wait State ซึ่งจะกล่าวถึงในภายหลัง) เมื่อ 8088 ได้รับสัญญาณแล้วก็จะปล่อยบัสต่าง ๆ ภายในระบบ (หยุดการควบคุมสัญญาณในบัสต่าง ๆ ไว้ชั่วคราว เพื่อให้วงจรหรืออุปกรณ์ที่ทำขบวนการ DMA ซึ่งในที่นี้คือ DMA Controller เข้าควบคุมสัญญาณในบัสเหล่านี้แทน) คือ บัสแอดเดรส, บัสข้อมูล และบัสควบคุมบางเส้น (ทำให้เอาท์พุทที่บัสต่าง ๆ เหล่านี้เป็น Hi-Impedance) ในช่วงท้ายของบัสไซเคิลที่ 8088 กำลังทำงานอยู่ และจะส่งสัญญาณ HOLD Acknowledge กลับไปให้กับ DMA Controller เพื่อแสดงให้ทราบว่า 8088 ได้ปล่อยบัสต่าง ๆ ในระบบแล้วจากนั้น DMA Controller จะเข้าควบคุมบัสต่าง ๆ เหล่านี้แทน 8088 และทำการสร้างบัสไซเคิลเพื่อที่ใช้สำหรับการส่งผ่านข้อมูลระหว่างอุปกรณ์ หรือวงจรที่ขอใช้ขบวนการ DMA กับหน่วยความจำของระบบในช่วงเวลานี้ DMA Controller จะส่งสัญญาณ DMA Acknowledge ให้กับอุปกรณ์หรือวงจรที่ขอ DMA เพื่อให้วงจรหรืออุปกรณ์นั้นทราบว่า DMA Controller พร้อมที่จะเข้าสู่ขบวนการ DMA แล้ว

ในระหว่างขบวนการ DMA นั้น DMA Controller จะทำหน้าที่เพียงแต่ควบคุมจังหวะการส่งผ่านข้อมูลระหว่างวงจรหรืออุปกรณ์ที่ขอ DMA กับหน่วยความจำเท่านั้น กล่าวคือ DMA Controller จะเพียงแต่สร้างสัญญาณควบคุมต่าง ๆ ที่จำเป็นต้องใช้ในการรับส่งข้อมูลเท่านั้น โดยที่ข้อมูลจะถูกรับหรือส่งจากวงจรหรืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



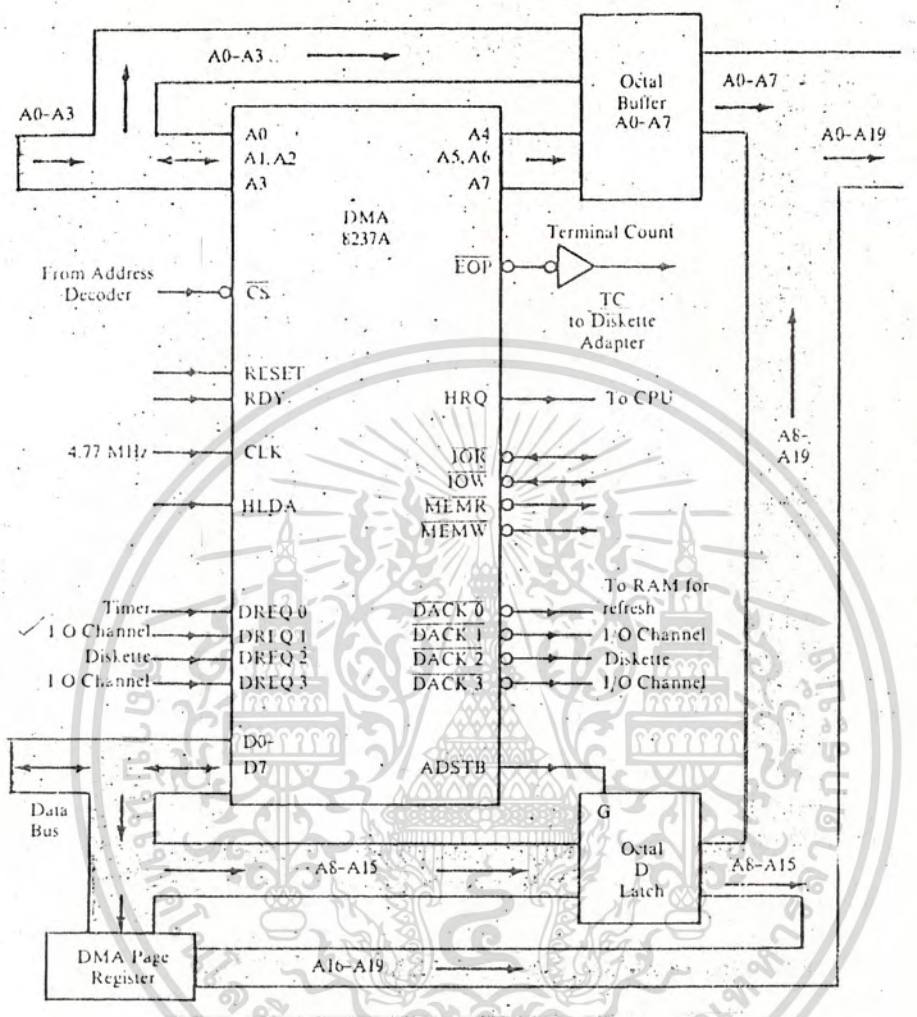
รูปที่ 4.1 ไดอะแกรมแสดงทิศทางของสัญญาณต่าง ๆ ในขบวนการ DMA

ที่ขอ DMA ไปยังหน่วยความจำของระบบโดยตรง (ข้อมูลไม่ผ่าน DMA Controller) ดังกล่าวในไดอะแกรมรูปที่ 4.1

4.2 <u>การใช้ขบวนการ DMA ใน IBM/PC</u>

ภายใน IBM/PC นั้น ใช้ 8237A-5 เป็น DMA Controller ซึ่ง 8237-5 นี้จะมีแชนแนลสำหรับการขอ DMA อยู่ 4 แชนแนล คือ แชนแนลที่ 0-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



DMA controller 8237A and associated circuits.

Address	Register	Number of Bits	Read/Write
00H	Ch 0 current address	16	Read/Write
01H	Ch 0 current word count	16	Read/Write
02H	Ch 1 current address	16	Read/Write
03H	Ch 1 current word count	16	Read/Write
04H	Ch 2 current address	16	Read/Write
05H	Ch 2 current word count	16	Read/Write
06H	Ch 3 current address	16	Read/Write
07H	Ch 3 current word count	16	Read/Write
08H	Status register	8	Read only
08H	Command register	8	Write only
09H	Request register	8	Write only
0AH	Mask register (1 bit)	8	Write only
0BH	Mode register	8	Write only
0CH	Clear first/last FF	1	Write only
0DH	Temporary register	8	Read only
0DH	Master clear	0	Write only
0FH	Mask register (all bits)	8	Write only

Valid port numbers of the 8237 DMA controller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับภายใน IBM/PC เองจะใช้งานแชนแนลเหล่านี้ 2 แชนแนล คือ แชนแนลที่ 0 ซึ่งใช้สำหรับการรีเฟรชหน่วยความจำที่เป็น DRAM (Dynamic RAM) ของระบบ และแชนแนลที่ 2 ซึ่งใช้ในการส่งผ่านข้อมูลระหว่าง Diskette Drive Controller กับหน่วยความจำของระบบ ส่วนแชนแนลที่ 1 และ 3 นั้นยังไม่ถูกใช้งาน

เนื่องจากการที่ IBM/PC ใช้แชนแนลที่ 0 ในการรีเฟรชหน่วยความจำนี้จึงทำให้วงจรภายนอกสามารถใช้งานได้เฉพาะแชนแนลที่ 1-3 เท่านั้น (ถึงแม้แชนแนลที่ 1 จะถูกกำหนดให้ใช้สำหรับ Diskette Drive Controller ก็ตาม แต่ Diskette Drive Controller นี้ก็เป็นแต่เพียงวงจรภายนอกที่ต้องนำมาเพิ่มเติมในภายหลังเช่นกัน) โดยบนสล็อตของ IBM/PC จะมีเฉพาะขาสัญญาณสำหรับการขอ DMA ในแชนแนลที่ 1-3 เท่านั้นสำหรับ BIOS ของ IBM/PC จะทำการโปรแกรม 8237A-5 ให้แชนแนลที่ 0 มีลำดับความสำคัญสูงสุด และแชนแนลที่ 3 มีลำดับความสำคัญต่ำที่สุด ตารางข้างล่างนี้จะแสดงลำดับความสำคัญ (0 แทนลำดับความสำคัญสูงสุด และ 3 แทนลำดับความสำคัญต่ำสุด) และการใช้งานแชนแนลทั้งสี่ของ DMA Contorller ภายใน IBM/PC

แชนแนล	ลำดับความสำคัญ	การใช้งาน
0	0	รีเฟรชหน่วยความจำ
1	1	----
2	2	Disk Drive Controller
3	3	----

4.3 <u>ขั้นตอนในขบวนการ DMA</u> ขั้นตอนในขบวนการ DMA ซึ่งสามารถจะแบ่งกล่าวได้เป็นข้อ ๆ ดังนี้

4.3.1 ก่อนที่ 8237A-5 จะทำงานตามที่เราต้องการได้นั้น เราจะต้องทำการโปรแกรมรูปแบบการทำงานที่ต้องการให้กับ 8237A-5 เสียก่อน สำหรับสิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ ที่ได้แสดงไว้ข้างล่างนี้ จะเป็นสิ่งที่จะต้องทำการโปรแกรมหรือกำหนดไว้ก่อนที่ 8237A-5 จะทำงานได้ตามที่เราต้องการ

- กำหนดรูปแบบการส่งผ่านข้อมูลในขบวนการ DMA ว่าเป็นการเขียน (DMA Write) หรือ อ่าน (DMA Read) ข้อมูลจากหน่วยความจำ
- กำหนดลักษณะการส่งผ่านข้อมูลว่าเป็นการส่งผ่านข้อมูลเพียงไบท์เดียวหรือหลายไบท์ต่อการเกิดขบวนการ DMA 1 ครั้ง
- จำนวนไบท์ทั้งหมดของข้อมูลที่ต้องการจะรับหรือส่ง โดยใช้ขบวนการ DMA
- กำหนดลำดับความสำคัญของแต่ละแชนแนลใน 8237A-5
- ตำแหน่งแอดเดรสเริ่มต้น (ของหน่วยความจำ) ของการส่งผ่านข้อมูลในขบวนการ DMA
- อีนาเบิลการขอ DMA ของแต่ละแชนแนลใน 8237A-5

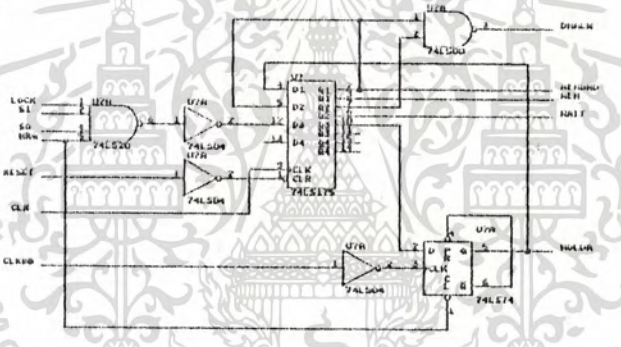
สำหรับการโปรแกรม 8237A-5 นี้ จะทำได้โดยใช้วิธีการในลักษณะเดียวกับการโปรแกรม 8259A กล่าวคือ ใช้คำสั่ง OUT ส่งคำสั่งต่าง ๆ ที่ใช้ในการโปรแกรม 8237A-5 ให้กับแอดเดรสของพอร์ทที่เป็น 8237A-5 นี้

4.3.2 เมื่ออุปกรณ์หรือวงจรใดต้องการที่จะใช้ขบวนการ DMA อุปกรณ์นั้นก็จะส่งสัญญาณให้กับขา DREQ ขาใดขาหนึ่งของ 8237A-5 เพื่อขอใช้ขบวนการ DMA (ขา DREQ นี้บนสล็อตของ IBM/PC จะใช้สัญญลักษณ์เป็น DRQ ดังนั้นเพื่อความสะดวกต่อไปจะเรียกขา DREQ ของ 8237A-5 เป็นขา DRQ แทน) สำหรับขา DREQ หรือ DRQ นี้จะมีอยู่ 3 ขา หรือ 3 แชนแนลคือ แชนแนลที่ 1, 2 และ 3 (ขา DRQ1, 2, 3) ส่วนแชนแนลที่ 0 หรือ DREQ0 นั้น จะถูกใช้งานในการรีเฟรชหน่วยความจำของระบบ ดังนั้นวงจรภายนอกจึงไม่สามารถที่จะขอให้ขบวนการ DMA ผ่านทางแชนแนลที่ 0 นี้ได้

4.3.3 แชนแนลทั้ง 3 ของ 8237A-5 นั้น จะมีลำดับความสำคัญต่างกัน คือ แชนแนลที่ 1 มีความสำคัญสูงสุด และแชนแนลที่ 3 มีลำดับความสำคัญต่ำสุด (เป็นกรณีของ IBM/PC ถ้าเป็นการใช้งาน 8237A-5 ในกรณีอื่นก็สามารถจะทำการจัดลำดับความสำคัญเป็นลักษณะอื่นได้ โดยการโปรแกรม 8237A-5 ให้ทำงานตามที่เราต้องการ สำหรับการจัดลำดับความสำคัญในลักษณะของ IBM/PC นี้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการโปรแกรม 8237A-5 โดย BIOS ในตอนเริ่มต้นการทำงานของระบบ) แต่อย่างไรก็ตามแชนแนลที่มีลำดับความสำคัญสูงสุดโดยแท้จริงนั้นจะเป็นแชนแนลที่ 0 หรือ DREQ0 ดังนั้นถ้ามีการขอ DMA พร้อมกันมากกว่า 1 แชนแนล 8237A-5 ก็จะเปรียบเทียบลำดับความสำคัญของแชนแนลต่าง ๆ ที่ขอ DMA นั้น และจะทำขบวนการ DMA ให้กับแชนแนลที่มีลำดับความสำคัญสูงสุดก่อน โดยการส่งสัญญาณ HRQ ให้กับวงจรสร้าง Wait - State ซึ่งวงจรนี้จะทำให้ 8088 หยุดการทำงานชั่วขณะจนกว่าขบวนการ DMA จะสิ้นสุดลง



รูปที่ 4.3 รูปไดอะแกรมของวงจรสร้าง Wait-State

4.3.4 จากรูปเมื่อมีการขอ DMA (สัญญาณ HRQ มีลอจิกเป็น "1") เข้ามายังวงจรสร้าง Wait-State วงจรนี้ก็จะรอจนกว่า 8088 อยู่ในสภาวะ Passive (ระดับลอจิกที่ขา S1 และ S0 ของ 8088 เป็น "1" ทั้งคู่ ซึ่งอยู่ในประมาณช่วงเวลา T3 ของบัสไซเคิล)และ 8088 ไม่ได้อยู่ในระหว่างเอ็กซีคิวท์คำสั่งที่มีคำสั่ง Lock เป็น Prefix (ขา Lock มีระดับลอจิกเป็น "1") แล้ววงจรสร้าง Wait-State ก็จะส่งสัญญาณ HLDA ให้กับ 8237A-5, ดิสเอเบิลบัฟเฟอร์ของบัสแอดเดรส, บัสข้อมูล และดิสเอเบิล 8288 ซึ่งจะมีผลให้บัสแอดเดรส,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัสข้อมูล และ บัสควบคุมบางเส้นมีสถานะเป็น Hi-Impedance รวมทั้งส่งสัญญาณ WAIT ให้กับขา READY ของ 8284 เพื่อสร้างสัญญาณ WAIT ให้กับ 8088 ด้วย (8284 จะทำให้สัญญาณที่ขา READY ของ 8088 ไม่แอคทีฟ คือ มีระดับลอจิกเป็น "0") นอกจากนี้วงจรสร้าง Wait-State ยังทำการส่งสัญญาณ HOLD (Hold Acknowledge) กลับไปให้กับ 8237A-5 ด้วย

4.3.5 เมื่อ 8237A-5 ได้รับสัญญาณ HOLDY แล้ว ก็จะส่งสัญญาณ DACK ไปให้กับอุปกรณ์ที่ขอ DMA กล่าวคือ ถ้า 8237A-5 ทำขบวนการ DMA ให้กับแชนแนลใดก็จะส่งสัญญาณ DACK กลับให้กับแชนแนลนั้น เช่น ถ้า 8237A-5 ทำขบวนการ DMA ให้กับแชนแนลที่ 1 (DRQ1) 8237A-5 ก็จะส่งสัญญาณ DACK ให้กับแชนแนลที่ 1 (สัญญาณที่ขา DACK1 แอคทีฟ) เป็นต้น ซึ่งการส่งสัญญาณ DACK ให้กับอุปกรณ์ที่ขอ DMA นี้ ก็เพื่อให้อุปกรณ์นั้น ๆ ทราบว่า 8237A-5 พร้อมที่จะทำขบวนการ DMA ให้กับอุปกรณ์นั้นแล้ว

สำหรับแชนแนลที่ 0 นั้นถึงแม้วงจรภายนอกจะไม่สามารถใช้งานได้ แต่สัญญาณ DACK0 จะถูกต่อออกมายังสล็อตของ IBM/PC ด้วย ทั้งนี้ก็เพื่อให้วงจรหรืออุปกรณ์ภายนอกสามารถใช้ช่วงเวลานี้ในการรีเฟรชหน่วยความจำได้ด้วย

4.3.6 จากนั้น 8237A-5 จะเข้าควบคุมบัสต่าง ๆ ภายในระแบบแทน 8088 โดยการส่งค่าแอดเดรสให้กับหน่วยความจำที่อุปกรณ์ที่ขอ DMA นั้นต้องการจะรับหรือส่งข้อมูลด้วย และจะส่งสัญญาณควบคุมต่าง ๆ ที่จำเป็นต้องใช้ในการส่งผ่านข้อมูล (สัญญาณ MEMR, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ และ $\overline{\text{IOW}}$) ออกมายังบัสควบคุมด้วยทั้งนี้ 8237A-5 จะสร้างบัสไซเคิลที่เกี่ยวกับการส่งผ่านข้อมูลขึ้นได้

4.3.7 หลังจากวงจรหรืออุปกรณ์ภายนอกนั้นได้รับสัญญาณ DACK แล้ว ก็จะยกเลิกสัญญาณ DRQ (ทำให้กลับเป็นลอจิก "0") ที่ส่งให้กับ 8237A-5 จากนั้น เมื่อ 8237A-5 เสร็จจากการทำงานในขบวนการ DMA แล้ว ก็จะยกเลิกสัญญาณ HRQ (ทำให้กลับเป็นลอจิก "0") ที่ส่งให้กับวงจรสร้าง Wait-State ซึ่งจะทำให้วงจรสร้าง Wait-State ยกเลิก HOLDY ที่ส่งให้กับ 8237A-5 เช่นกัน และวงจรสร้าง Wait-State จะยกเลิกสัญญาณ WAIT ที่ส่งให้กับ 8284 รวมทั้งการอีนาเบิลบัฟเฟอร์ของบัสแอดเดรส, บัสข้อมูล และ 8288 ด้วยซึ่งจะเป็นผลให้ 8088 กลับเข้าสู่การทำงานตามปกติต่อจากที่ได้ทำไว้ก่อนที่จะเข้าสู่สภาวะ Wait อย่างไรก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามในช่วงที่เริ่มต้นเข้าสู่การทำงานในบัสไซเคิลของ 8088 นั้น จะมีการหน่วงเวลาออกไปเท่ากับช่วงเวลาของคล็อก 2 ลูก เพื่อให้การทำงานในบัสไซเคิลต่อจากที่ทำค้างอยู่นั้นสามารถเกิดขึ้นได้ทัน

ขั้นตอนต่าง ๆ ที่ได้กล่าวมานี้เป็นขั้นตอนการทำงานที่เกิดขึ้นในการทำขบวนการ DMA ใน IBM/PC ซึ่งจะสังเกตุได้ว่าภายใน IBM/PC นั้น ใช้วงจรสร้าง Wait-State ในการทำขบวนการ DMA ทั้งนี้เนื่องจากภายใน IBM/PC จะใช้งาน 8088 ในโหมด Maximum ซึ่งไม่มีขา HOLD และ HOLDA และยังจำเป็นต้องใช้ขา $\overline{RQ}/GT$ สำหรับการใช้งานร่วมกับ Co-Processor ด้วย

4.4 การอ้างแอดเดรสของ 8237A-5

จากลักษณะการจัดเรียงขาของ 8237A-5 ที่ได้กล่าวถึงในตอนต้น จะเห็นว่าจำนวนขาที่เป็นบัสแอดเดรสของ 8237A-5 จะมีเพียง 8 ขา คือ A0-A7 โดยที่ขา A0-A7 เป็นแบบ Bi-Directional ซึ่งจะใช้เป็นอินพุทเมื่อ CPU ต้องการจะติดต่อกับรีจิสเตอร์ภายในของ 8237A-5 แต่ในระหว่างที่ 8237A-5 สร้างขบวนการ DMA เพื่อช่วยในการส่งผ่านข้อมูลระหว่างหน่วยความจำกับอุปกรณ์ภายนอก (หรือระหว่างหน่วยความจำ) อยู่นั้น ขา A0-A3 จะถูกใช้เป็นเอาท์พุทร่วมกับ A4-A7 ในการอ้างแอดเดรสของหน่วยความจำที่เกี่ยวข้อง ซึ่งจะเห็นได้ว่าไม่เพียงพอที่จะอ้างแอดเดรสได้ครบทั้ง 64 Kbyte ดังนั้น 8237A-5 จึงอาศัยวิธีการมัลติเพล็กซ์ค่าแอดเดรส 8 บิทบนที่เหลือ คือ A8-A15 ออกมาบนขา DB0-DB7 ของ 8237A-5 โดย 8237A-5 จะใช้ช่วงเวลาของคล็อก S1 ในแต่ละบัสไซเคิลในการส่งค่าแอดเดรส 8 บิทบนออกมาบนขา DB0-DB7 เพื่อส่งให้กับอุปกรณ์ที่ทำหน้าที่แลทช์ค่าแอดเดรส 8 บิทบนนี้ออกมาบนบัสแอดเดรสของระบบ โดยขอบขาลงของสัญญาณจากขา ADSTB (ขา Address Strobe; ขา 8 ของ 8237A-5) จะถูกใช้สำหรับควบคุมให้อุปกรณ์นี้ทำการแลทช์ค่าแอดเดรส A8-A15 ซึ่ง 8237A-5 ส่งออกมาบนขา DB0-DB7 นั้นออกมาบนบัสแอดเดรสของระบบ ด้วยวิธีการเช่นนี้จะเห็นว่าค่าแอดเดรสทั้ง 16 บิทคือ A0-A15 จะถูกส่งออกมาบนบัสแอดเดรสได้ โดยไม่จำเป็นต้องเพิ่มจำนวนของ 8237A-5 เลย

สำหรับในกรณีที่ 8237A-5 ทำงานในแบบ Block หรือ Demand Transfer ซึ่งมีการส่งผ่านข้อมูลเข้าหรือออกจากหน่วยความจำเป็นจำนวนมากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8237A-5 จะต้องทำการส่งค่าแอดเดรสของหน่วยความจำที่เกี่ยวข้องออกมาบนบัสแอดเดรสอย่างต่อเนื่องกัน ซึ่งโดยทั่วไปแล้วค่าแอดเดรสในส่วนที่อยู่ใน 8 บิทบนคือ A8-A15 นั้น จะไม่เปลี่ยนแปลงมากนัก กล่าวคือ ค่าแอดเดรสในบิท A8-A15 จะถูกเปลี่ยนแปลงเฉพาะกรณีที่การเปลี่ยนแปลงค่าแอดเดรสนั้นมีการทด (Carry) หรือยืม (Borrow) จากแอดเดรสบิท A7 ให้กับแอดเดรสบิท A8 เช่นกรณีที่ค่าแอดเดรสเปลี่ยนจาก 00FFH เป็น 0100H เท่านั้น ดังนั้นเพื่อเป็นการลดเวลาที่ใช้ในแต่ละบัสไซเคิลที่เกี่ยวกับการส่งผ่านข้อมูลในกรณีของ Block หรือ Demand Transfer ลง 8237A-5 จึงจะทำการสร้างบัสไซเคิลในช่วงของคล็อก S1 (ช่วงเวลาที่ 8237A-5 ส่งข้อมูลคือ แอดเดรสบิท A8-A15 ให้กับวงจรแลทช์ภายนอกเฉพาะในกรณีที่มีการเปลี่ยนแปลงค่าแอดเดรสนั้นมีผลกระทบต่อแอดเดรสบิท A8-A15 เท่านั้น ซึ่งจะเห็นได้ว่าคล็อก S1 จะถูกสร้างขึ้นเมื่อมีการส่งผ่านข้อมูลทุก ๆ 256 ไบท์แทนที่จะถูกสร้างขึ้นในการส่งผ่านข้อมูลทุก ๆ ไบท์ ดังนั้นจึงสามารถที่จะลดจำนวนคล็อกลงได้ 255 ลูกในการส่งผ่านข้อมูลด้วยวิธี Block หรือ Demand Transfer ทุก ๆ 256 ไบท์ แต่ดังที่ได้กล่าวไว้ข้างต้นแล้วว่าในระบบของ IBM/PC นั้น ไม่สามารถที่จะโปรแกรมให้ 8237A-5 ทำการส่งผ่านข้อมูลหลาย ๆ ไบท์ในครั้งเดียวได้ ดังนั้นในการส่งผ่านข้อมูลโดยการใช้ขบวนการ DMA ในทุกไบท์จะมีช่วงเวลาของคล็อก S1 ถูกสร้างขึ้นด้วยเสมอ

ภายในระบบของ IBM/PC นั้นใช้ CPU เบอร์ 8088 ที่อ้างแอดเดรสของหน่วยความจำได้ถึง 1 Mbyte (Megabyte) ซึ่งมากเกินกว่าที่ 8237A-5 จะอ้างถึงได้ ดังนั้นภายใน IBM/PC จึงอาศัยวิธีการทางฮาร์ดแวร์เข้าช่วยในการสร้างแอดเดรส 4 บิทบนนี้เหลือ คือ A16-A19 โดยเรียกว่า Page Register ซึ่งการเขียนหรืออ่านข้อมูล (แอดเดรส 4 บิท บน) จาก Page Register นี้ จะทำได้โดยการเขียนหรืออ่านข้อมูลผ่านทางพอร์ท I/O ของ IBM /PC 3 พอร์ท สำหรับแชนแนลทั้ง 3 แชนแนล (แชนแนล 1, 2, 3) ดังในรูปข้างบน

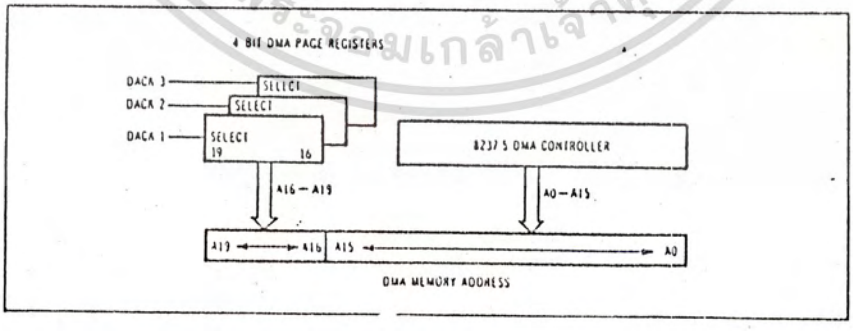
ข้อมูลที่เก็บอยู่ใน Page Register นี้จะถูกส่งออกมายังเส้นแอดเดรส 8 บิทบนของบัสแอดเดรส เมื่อ 8237A-5 ตอบสนองต่อการขอ DMA (สัญญาณ DACK แอคทีฟ) ที่เกิดขึ้น โดยถ้า 8237A-5 ทำการตอบสนองให้กับการขอ DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แชนแนล	แอดเดรสของพอร์ท I/O
1	0083H
2	0081H
3	0082H

รูปที่ 4.16

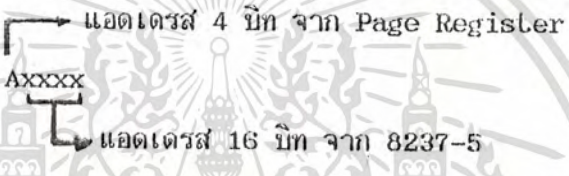
ในแชนแนลใด Page Register ก็จะส่งแอดเดรส 4 บิทแรกของแชนแนลนั้นออกมาบนบัสแอดเดรส เมื่อประกอบกับแอดเดรส 16 บิทที่ 8237A-5 ส่งออกมาแล้ว ก็จะครบทั้ง 20 บิทพอดี



รูปที่ 4.17 แสดงถึงการประกอบกันของแอดเดรส 4 บิทจาก Page Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่นถ้าเราโปรแกรม Page Register โดยการใช้คำสั่ง OUT ส่งข้อมูล OAH ให้กับพอร์ท 0081H (แชนแนล 2) เมื่อ 8237A-5 ทำการตอบสนองต่อการขอ DMA ให้กับแชนแนลที่ 2 นั้น สัญญาณ DACK2 จะแอคทีฟซึ่งจะทำให้ Page Register ส่งค่าแอดเดรส 4 บิทที่เราโปรแกรมไว้ ซึ่งในกรณีนี้คือ OA ในฐาน 16 ออกมาบนเส้นแอดเดรส A16-A19 ของบัสแอดเดรส เมื่อประกอบกับแอดเดรส 16 บิทที่ 8237A-5 ส่งออกมาบนบัสแอดเดรสแล้ว จะได้เป็นแอดเดรส 20 บิท ดังนี้



อย่างไรก็ตามสิ่งสำคัญสิ่งหนึ่งที่จะต้องคำนึงถึงในการโปรแกรม Page Register ก็คือ Page Register นี้เป็นรีจิสเตอร์ขนาด 4 บิท ซึ่งจะทำให้ข้อมูลที่ส่งให้กับ Page Register ถูกใช้งานเพียง 4 บิทคือ บิท D0-D3 เท่านั้น ดังนั้นการส่งข้อมูล OAH, 1AH, 2AH, ... นั้นจะให้ผลเหมือนกัน คือ แอดเดรส 4 บิทที่ Page Register ส่งออกมาบนบัสแอดเดรสจะเป็น OAH ในฐานสิบหก

4.5 ประสิทธิภาพในขบวนการ DMA

อัตราในการส่งผ่านข้อมูลในขบวนการ DMA และ ช่วงเวลาที่สูญเสียไปก่อนที่ 8237A-5 จะเริ่มต้นการส่งผ่านข้อมูลในขบวนการ DMA ซึ่งเป็นสิ่งที่เราจะต้องคำนึงถึงเสมอในการออกแบบวงจรอินเทอร์เฟสที่จำเป็นต้องใช้ขบวนการ DMA และต้องการความรวดเร็วในการตอบสนองต่อการขอ DMA รวมทั้งอัตราเร็วในการส่งผ่านข้อมูลที่สูงพอด้วย

4.6 อัตราสูงสุดในการส่งผ่านข้อมูลโดยใช้ DMA

โดยทั่วไปแล้วในแต่ละบัสไซเคิลที่ถูกสร้างขึ้นในขบวนการ DMA นั้น จะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อกจำนวน 5 ลูก คือ S0-S4 แต่สำหรับในระบบของ IBM PC นั้นจะเพิ่มจำนวนคล็อกขึ้นอีก 1 ลูกโดยอัตโนมัติ ซึ่งคล็อกที่เพิ่มเข้ามานี้ จะแทรกอยู่ระหว่างคล็อก S3 และ S4 โดยจะเรียกว่า Sw สำหรับ Sw นี้ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเพิ่มเข้ามาเพื่อเพิ่มช่วงเวลาในการรับ/ส่งข้อมูลระหว่างหน่วยความจำกับอุปกรณ์ภายนอกให้มากขึ้น ทั้งนี้ก็เพื่อป้องกันปัญหาที่อาจจะเกิดขึ้นได้จากการที่หน่วยความจำหรืออุปกรณ์ภายนอกทำงานไม่ทัน ดังนั้นสำหรับใน IBM/PC จะใช้ช่วงเวลาในบัสไซเคิลในขบวนการ DMA เท่ากับช่วงเวลาของคล็อก 6 ลูก คือ S0-S3, Sw และ S4 และเนื่องจากคาบเวลาของคล็อก 1 ลูก จะประมาณ 210 nanosec. ดังนั้นในแต่ละบัสไซเคิลในขบวนการ DMA ของ IBM PC จะใช้เวลาประมาณ 1.26 usec.

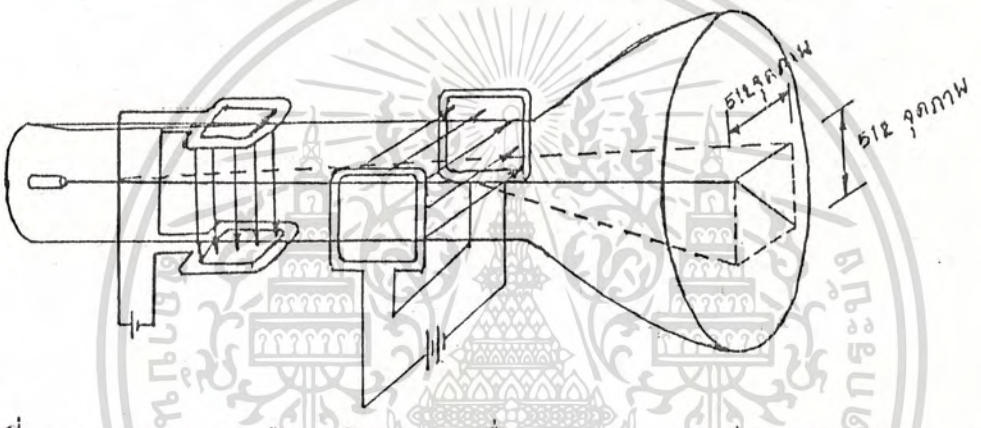
ภายใน IBM PC จะทำการโปรแกรมให้ 8237A-5 ทำงานในโหมด Single Transfer ซึ่งในระหว่างบัสไซเคิลของขบวนการ DMA จะถูกแทรกด้วยบัสไซเคิลของ 8088 1 ไซเคิล โดยที่บัสไซเคิลของ 8088 จะใช้ช่วงเวลาโดยทั่วไปเท่ากับช่วงเวลาของคล็อก 4 ลูก หรือประมาณ 840 nanosec. ดังนั้นจึงจำเป็นต้องรวมช่วงเวลาที่เสียไปสำหรับบัสไซเคิลของ 8088 นี้กับช่วงเวลาของบัสไซเคิล (ในขบวนการ DMA) ในการส่งผ่านข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

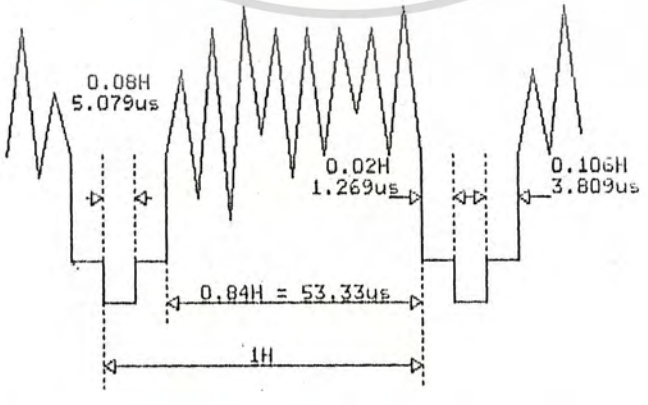
การคำนวณและการสร้าง

เมื่อเราต้องการให้จอมอนิเตอร์แสดงภาพในแนวนอนและแนวตั้งเท่ากับ 512*512 จุดภาพ(pixel) ซึ่งโดยปกติแล้วการสแกนภาพของโทรทัศน์ระบบยุโรปจะสแกนในแนวนอน 625 line แต่ในโครงงานนี้ต้องการแค่เพียง 512 line เท่านั้น ซึ่งจะเห็นได้ว่าภาพที่ได้จากการสแกนนี้จึงไม่เต็มจอ ฉะนั้นเราจึงมีหลักการในการคำนวณและสร้างวงจรเพื่อที่จะให้การสแกนผลภาพอยู่กลางจอภาพดังนี้



รูปที่ 5.1 การสแกนของลำแสงอิเลคตรอนเนื่องจากขดลวดแม่เหล็ก 512*512 pixel

1. วัดสัญญาณจากกล้องวีดีโอ แล้วเทียบกับสัญญาณมาตราฐาน ในโครงงานนี้ใช้กล้องวีดีโอ ยี่ห้อ SONY รุ่น CCD-M8E ซึ่งสามารถวัดสัญญาณภาพโดยใช้ Oscilloscope ได้ดังรูปที่ 10



รูปที่ 5.2 สัญญาณทางด้านแนวนอนหรือสัญญาณใน 1 line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. นำความถี่ทางด้านแนวนอน(Horizotal Frequency) มาทำการคำนวณเพื่อหาค่าความถี่ Oscillate ซึ่งจากทฤษฎีแล้วความถี่ Oscillate ประมาณ 8 MHz (15,265*512)

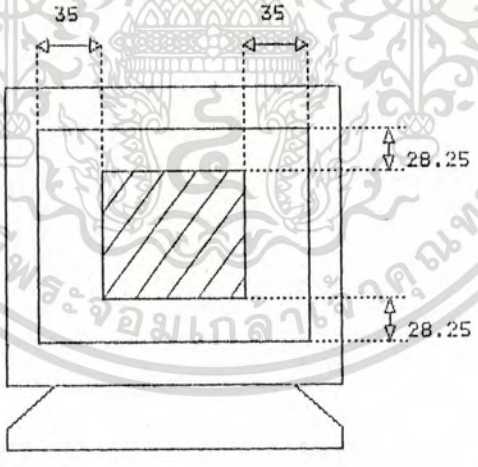
จากสัญญาณภาพที่วัดได้ในโครงงานนี้ 1 line หรือความถี่ทางแนวนอน มีค่าเท่ากับ 18,751.17 Hz (1/53.33Microsec) ฉะนั้นถ้าสแกน 512 line จะมีความถี่เท่ากับ 9.6 MHz

โครงงานนี้ใช้ Crystal ขนาด 21.792 MHz เป็นตัวกำเนิดความถี่ เมื่อนำไปเข้า Counter หาร 2 แล้ว จะได้ความถี่ที่นำไปใช้เท่ากับ 10.896 MHz

จากรูปที่ 9 เมื่อความถี่ 18,751.17 Hz คือการสแกนภาพ 1 line

ถ้าความถี่ที่ใช้มีค่า 10.892 MHz จะต้องมีการสแกนถึง 581.08 line

จากค่าที่ได้จะเห็นว่าเมื่อใช้ความถี่ 10.892 MHz จะทำให้การสแกนทางแนวนอนเกินจากค่าที่กำหนดประมาณ 70 line (581.08-512) ซึ่งสามารถเว้นขอบซ้ายขวาได้ข้างละ 35 line หรือจุดภาพ(ดูรูปที่ 5.3)

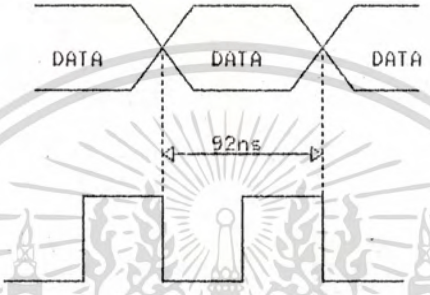


รูปที่ 5.3 ภาพการสแกนหน้าจอโทรทัศน์

ในการสแกนภาพของโทรทัศน์ระบบยุโรป มีเส้นสแกน 625 เส้น/ภาพ(2 ฟิลด์) แต่ที่เราต้องการคือ 512 เส้น/ภาพ เพราะฉะนั้นจะมีพื้นที่ว่าง 113 เส้น สำหรับการนำข้อมูลที่อยู่ในหน่วยความจำมาสแกนภาพ สามารถสแกนได้ฟิลด์เดียว ฉะนั้นจะมีพื้นที่ที่ไม่ใช้ในการสแกนเท่ากับ 56.5 เส้น ซึ่งสามารถทำการเว้นขอบบนและขอบล่างได้ข้างละ 28.25 เส้น(ดูรูปที่ 5.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อใช้ความถี่ Oscillate ขนาด 10.896 MHz การรับส่งข้อมูลแต่ละครั้งจะต้องใช้เวลา ประมาณ 92 nanosec (1/10.896MHz) ดังนั้นการกำหนดค่า RAM นั้น จะต้องกำหนดโดยค่าเวลาที่ใช้ในการเขียนและอ่านต้องไม่ต่ำกว่า 92 nanosec ซึ่งในโครงงานนี้เราใช้ RAM เบอร์ 4464 ซึ่งใช้เวลาในการเขียนและอ่าน 200 nanosec



รูปที่ 3.4 การเขียนหรืออ่านข้อมูล 1 cycle

ในการออกแบบการเขียนข้อมูลลงไปใน RAM ขนาด 8 บิท (RAM # 4464 มีขนาด 4 บิท จะต้องใช้ 2 ตัว เพื่อให้ได้ 8 บิท) จำนวน 4 ชุด จะต้องใช้ Latch เพื่อช่วยในการคงสภาวะของข้อมูลไว้จำนวน 4 ชุดด้วย ในโครงงานนี้เราใช้ Latch # 374 ช่วยในการเขียนข้อมูลเข้าไปใน RAM และใช้เบอร์ 373 เพื่อช่วยในการนำข้อมูลออกจาก RAM เพื่อเข้าสู่ Data Bus และจะเป็น Buffer ให้กับ RAM ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบ การใช้ RAM ชนิด Static และ Dynamic

RAM ชนิด Static

จากราคาในท้องตลาดแล้ว RAM ชนิด Static ที่มีความจุขนาด 32 Kbyte จะมีราคาขาย 500 บาทต่อตัว ซึ่งถ้า RAM ที่มีขนาดความจุสูงขึ้น ราคาขายก็จะสูงขึ้นเช่นกัน เมื่อต้องการ RAM ที่มีหน่วยความจำขนาด 256 Kbyte ก็จะต้องใช้ RAM ถึง 8 ตัว ดังนั้นต้องใช้ต้นทุนถึง 4,000 บาท และ RAM ชนิดนี้ จะต้องใช้ Address line อย่างน้อย 16 เส้น ในการเขียนและอ่านแต่ละครั้ง ใช้ Access time 60 nanosec

RAM ชนิด Dynamic

จากราคาขายในท้องตลาดแล้ว RAM ชนิด Dynamic ที่มีขนาดความจุ 64 Kbyte 4 Bit จะมีราคาขายตัวละ 100 บาท ซึ่งถ้าต้องการหน่วยความจำที่มีขนาด 256 Kbyte 8 Bitแล้ว จะต้องใช้ RAM 8 ตัว ดังนั้นต้องใช้ต้นทุนเพียงแค่ 800 บาทเท่านั้น และRAM ชนิดนี้ จะต้องใช้ Address line อย่างน้อย 8 เส้น ในการอ่านและเขียนแต่ละครั้ง ใช้ Access time 200 nanosec

ข้อดีข้อการใช้ RAM ชนิด Dynamic

- ความหนาแน่นต่อการเก็บข้อมูลมีมากกว่า
- ขนาดของRAMชนิดนี้จะเล็กกว่าแบบ Static ในขณะเมื่อมีความจุเท่ากัน
- ราคาถูกกว่า ในขณะที่ความจุเท่ากัน

ข้อเสียของ RAM ชนิด Dynamic

- การนำไปใช้งานจะมีวงจรที่ยุ่งยากกว่า
- ความเร็วในการอ่านและเขียนข้อมูลจะช้ากว่า
- ต้องมีการRefesh ข้อมูลอยู่ตลอดเวลา คือในตำแหน่งของ Row Address จะต้องมีการ Refash ทุกๆ 4 ms

ข้อที่เหมาะสมกับการใช้ Dynamic RAM

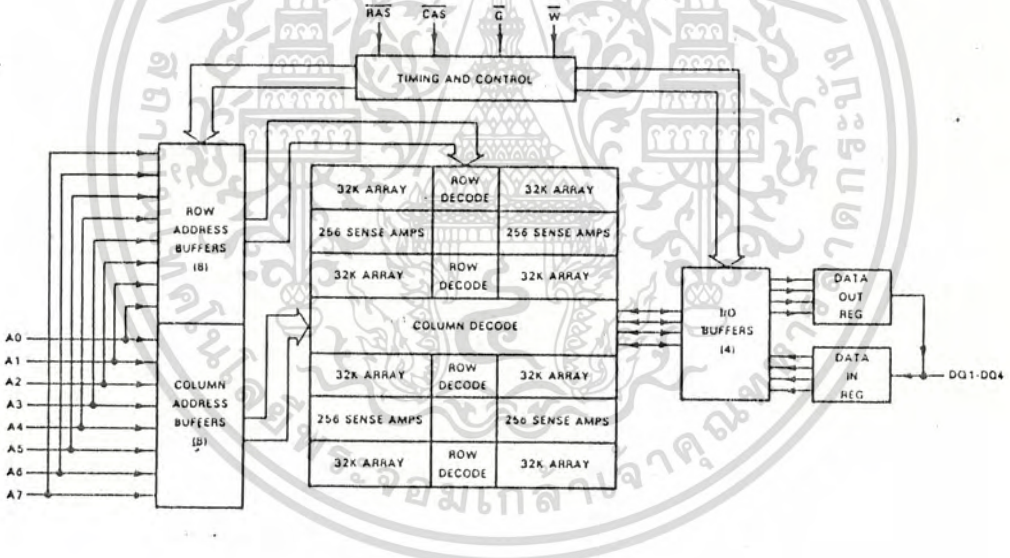
- Dynamic RAM เหมาะกับงานที่ใช้ Memory Area ขนาดใหญ่ เช่น Main Board Computer เป็นต้น
- ตัว RAM ชนิดนี้จะมีขนาดเล็ก ฉะนั้นจึงใช้พื้นที่ในการวาง RAM สะดวกและเล็กว่า Static RAM ในขณะที่มีหน่วยความจำเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Dynamic RAM 4464

RAM # 4464 เป็น RAM ขนาด 4 บิท ที่มีความเร็วในการทำงานสูง และสามารถเก็บข้อมูลได้ 65,536 * 4 Bit ซึ่งใน $\overline{RAS}$ จะมี Access Time สูงสุดถึง 120,150และ200 nanosec และจะมีค่า Power ในการทำงานเท่ากับ 125 mw และในขณะ Standbyเท่ากับ 17.5 mw

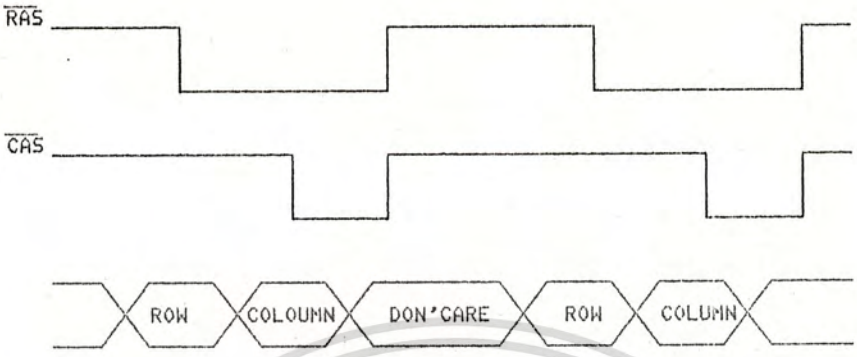
ใน Address line จะมี Row Address 8 line ซึ่งสามารถอ้างตำแหน่งได้ 256 ตำแหน่ง และใน Column Address จะมี 8 line เช่นกัน ดังนั้นสามารถอ้าง Address รวมได้ 256*256 = 65,536 Address ซึ่งจะมี Data input 4 line สามารถเก็บข้อมูลครั้งละ 4 Bit



รูปที่ 5.5 functional block diagram of dynamic ram # 4464

ในการอ้าง Address จะใช้สัญญาณ $\overline{RAS}$ (Row Address Stobe) เป็นสัญญาณในการ Stobe Row Address และสัญญาณ $\overline{CAS}$ (Column Address Stobe) จะเป็นสัญญาณในการ Stobe column Address การ Stobe จะใช้ Pulse ขาลง ของ $\overline{RAS}$ และ $\overline{CAS}$ ซึ่งจะทำงานสัมพันธ์กับ Address ใน Address line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 การ Stobe Address

$\overline{W}$ เป็นสัญญาณควบคุมการอ่านหรือเขียน ซึ่งถ้ามีค่าลอจิกเป็น"1"จะเป็นการอ่าน และถ้าลอจิกเป็น"0"ก็จะเป็นการเขียน

$\overline{G}$ เป็นสัญญาณควบคุม Out Put Buffer ของ RAM ซึ่งในขณะเขียนจะให้เป็นลอจิก"1" และในขณะอ่านจะให้เป็นลอจิก"0"

<u>การอ่านและการเขียนเข้าไปใน RAM</u>

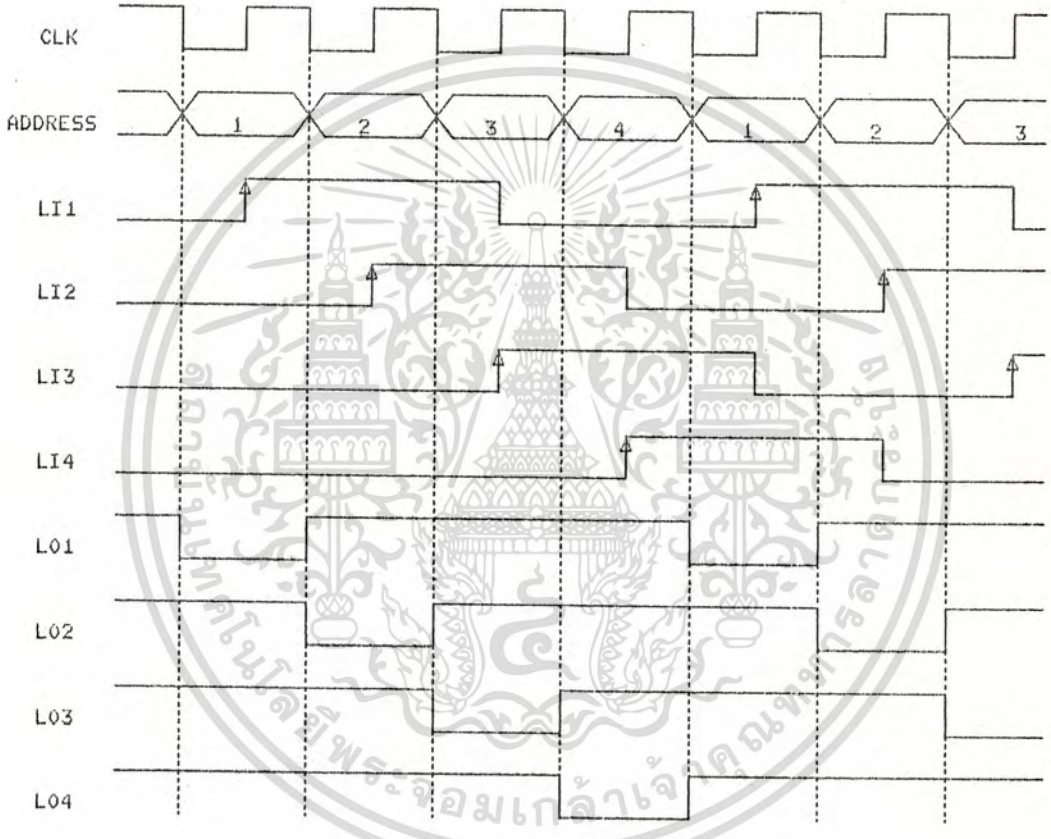
จากข้อมูลดิจิตอลที่ได้ เมื่อต้องการจะเก็บข้อมูลเข้าไปใน RAM จะต้องผ่านการคงข้อมูลไว้ โดยใช้ Latch จากข้อมูลใน Data line จะทำการแบ่งออกเป็น 4 ชุด โดยข้อมูลชุดที่ 1 จะเก็บใน RAM ชุดที่ 1 ข้อมูลชุดที่ 2 จะเก็บใน RAM ชุดที่ 2 ทำเช่นนี้จนครบ 4 ชุด แล้วเริ่มต้นใหม่ (ดูรูปที่ 5.7)

ผลที่ได้จากการเขียนแบบนี้ จะทำให้ลดความถี่ในการอ่านและเขียน RAM ซึ่งจะได้มีเวลาที่จะทำการอ่านและเขียนได้มากพอสมควร (ค่าคาบเวลา จะมีค่ามากกว่า Access time ของ RAM) ข้อดีของการใช้วิธีนี้ คือ

- สามารถขยายขนาดของการ Sampling Data ได้มากขึ้น
- สามารถใช้ RAM ความเร็วต่ำได้
- มีเวลามากพอในการเขียนและอ่าน RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Latch จะต้องสอดคล้องกับ Clock Pulse ขนาด 10.892 MHz โดยLatch จะทำงานเมื่อ Clock Pulse อยู่ในช่วงขาขึ้นและ oc อยู่ในช่วงลอจิก "0" ซึ่ง Latch ทั้ง 4 ตัวจะเปลี่ยนการทำงานตาม Clock Pulse ที่เข้ามา ดูจากรูป Timing Diagram (รูปที่ 5.7)



รูปที่ 5.7 Timing Diagram of Latch in

<u>การสร้าง Address line</u> ในการสร้าง Address line ให้กับหน่วยความจำนั้น ถ้าหน่วยความจำที่ใช้กับข้อมูลเพียงอย่างเดียวแล้ว เราสามารถสร้างได้โดยไม่ต้องคำนึงค่าอื่นๆจากภายนอกมากนัก แต่หน่วยความจำที่ใช้เกี่ยวกับงานการเก็บข้อมูลภาพ ซึ่งเกิดจากการสแกนของกล้องวีดีโอแล้ว ในการสร้าง Address line จะต้องคำนึงถึงส่วนต่างๆหลายอย่างคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สัญญาณภาพ ที่ประกอบด้วย สัญญาณซิ้งค์ทางแนวนอน ซิ้งค์ทางแนวตั้ง สัญญาณแบล้งค์กิ้ง

2. ภาพหนึ่งเฟรมมีสองฟิลด์ ฟิลด์หนึ่งมีซิ้งค์แนวนอน 312.5 ลูก และเมื่อหมดหนึ่งฟิลด์จะมีซิ้งค์แนวตั้งหนึ่งลูก

3. ในการเก็บสัญญาณภาพที่เป็นดิจิตอล จะไม่เก็บสัญญาณส่วนที่เป็นสัญญาณแบล้งค์กิ้งและสัญญาณซิ้งค์

4. จำนวนจุดที่จะใช้ในการการ Samping สัญญาณในหนึ่งเส้นทางแนวนอนจะใช้กี่จุด

5. จำนวนเส้นที่จะใช้ทางแนวนอนกี่เส้นในหนึ่งฟิลด์

ความถี่ที่เป็น Clock ให้กับ Counter นั้นได้มาจากวงจร Oscillater 21.792 MHz แล้วผ่านวงจรหารสอง ได้ความถี่ 10.892 MHz ซึ่งความถี่นี้ป้อนให้กับ A/D ด้วย การคำนวณค่าต่างๆ ที่ใช้ในการสร้าง Counter Address line มีดังนี้

- สัญญาณซิ้งทางแนวตั้งที่มีความถี่ 50 Hz หรือมีคาบเวลาเท่ากับ 20 ms
- สัญญาณซิ้งค์ทางแนวนอนที่มีความถี่ 15,625 Hz หรือมีคาบเวลาเท่ากับ 64 microsec

ส่วนที่เป็นสัญญาณภาพจริงๆในหนึ่งไลน์นั้น เป็นเวลาประมาณ 52-54 microsec เพราะต้องลบเอาเวลาของ แบล้งค์กิ้งและซิ้งค์ออกด้วย ในที่นี้จะใช้เวลา 53.33 microsec ในการเก็บจุดภาพ 512 จุด ดังนั้นจะคำนวณความถี่ของสัญญาณที่นำมาเป็น Clock ให้กับ Couter ดังนี้

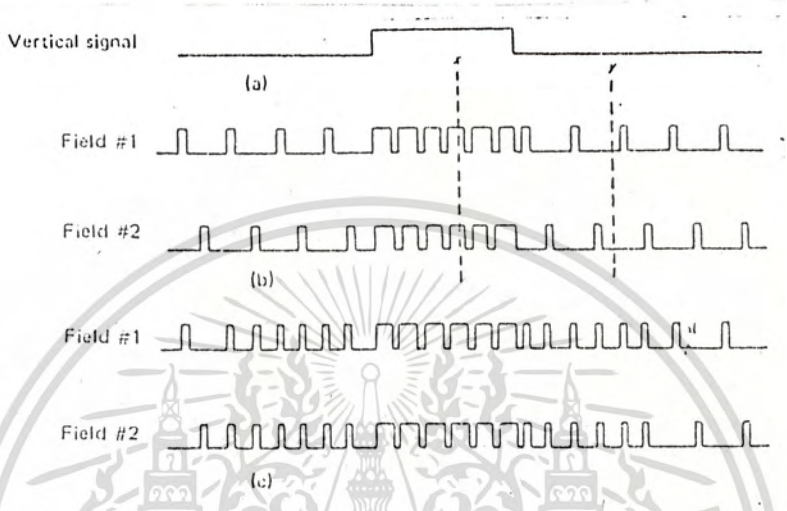
512 ครั้งใช้เวลา 53.33 microsec

1 ครั้งใช้เวลา 53.33/512 = 0.104 microsec

คิดเป็นความถี่ได้ $1/0.104*10^{-6}$ = 9.6 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-การแยก Odd - Even Field

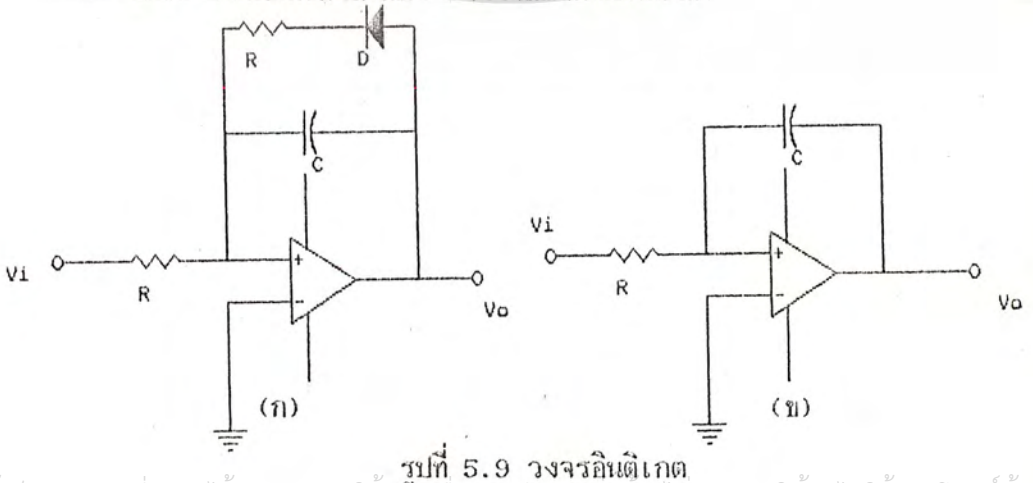


รูปที่ 5.8 แสดงส่วนของสัญญาณวีดีวอไลท์ชิ่ง

การแยกจะใช้การ toggle ผลที่ได้จะอยู่ที่ pulse สุดท้าย ซึ่งการ latch จะบอกว่าเป็น odd - even field และจะใช้ V sync เป็นสัญญาณการ latch

การแยกสัญญาณ V-sync

จะใช้วงจรอินทิเกรต และ สามารถคำนวณได้ดังนี้



รูปที่ 5.9 วงจรอินติเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากค่าแอมปริจูด(Vin)ของสัญญาณซิ้งคี้ทางแนวนอนที่ได้จากกล้อง VDO เมื่อนำมาผ่านวงจร Multideplex (คูณ -1) จะได้เป็นวงจรอินทริเกต ดังรูปที่ 5.9(ก)

จากสัญญาณ H sync จะเห็นว่ามีลักษณะเป็น Square wave จากรูป 5.9 (ข) จะเห็นว่าในขณะที่ Vin น้อยกว่า 0 นั่นก็คือ -Vmax

$I_{in} = -I_f$

$V(+) = V(-) = 0$

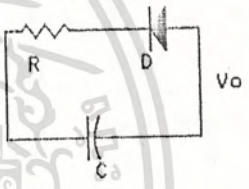
$I_{in} = V_{in}/R1$

$V_f = (1/c)\int I_f\, dt$ เมื่อ $V_c = V_o - 0 = V_o$

$I_f = c\, dV_o/dt$

$V_{in}/R1 = -cdV_o/dt$

ดังนั้นจะได้ $V_o = (-1/Rc)\int V_{in}\, dt + V_{co}$



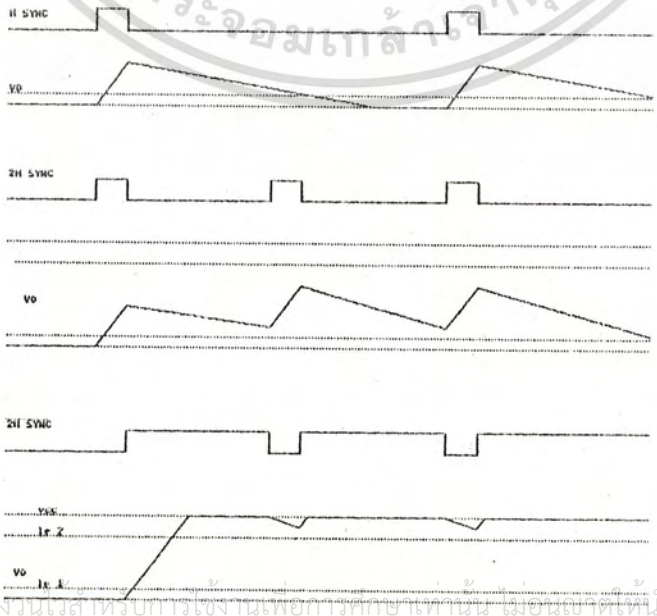
$V_{in} = V_{max}$

$V_o = -V_{mt}/Rc$

ในขณะที่ $V_{in} = 0$

จะได้ $V_o = V_{cm}(1-e^{-t/R2C})$

จากสมการนี้นำมาเขียนเป็นรูป Timming Diagram ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างสัญญาณซิ้งค์ (Sync generater)

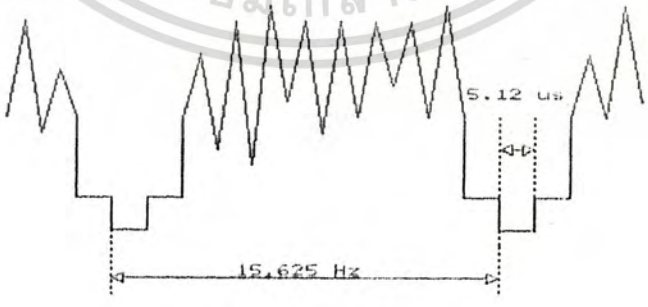
เมื่อต้องการจะอ่านข้อมูลจาก RAM ในขณะที่เราไม่ได้นำสัญญาณจากกล้องวีดีโอ ซึ่งเราอาจจะนำเอาข้อมูลจากฟลอบปี้ดิสต์ หรือข้อมูลจากฮาร์ดดิสต์ เพื่อนำไปแสดงผลที่จอโทรทัศน์ ดังนั้นเราจำเป็นจะต้องสร้างสัญญาณซิ้งค์ขึ้นมาช่วยในการสแกนภาพและนำไปควบคุมการทำงานของ RAM

ซึ่งการสร้างสัญญาณซิ้งค์นั้น ในโครงงานนี้จะใช้ความถี่ 10.892 MHz โดยผ่านวงจร Counter(IC#74LS393) ซึ่งมีหลักการคำนวณดังนี้

จากความถี่มาตราฐาน 15,625 Hz (H sync)จะได้จุดภาพ 1 จุดภาพ

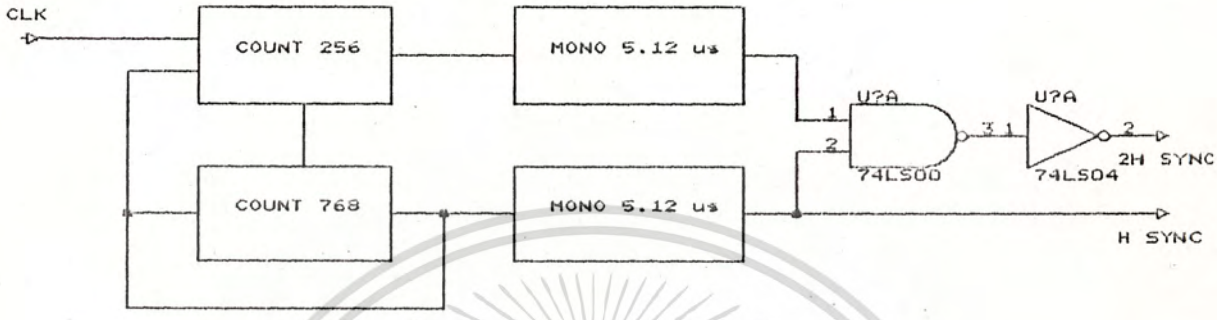
เมื่อใช้ความถี่ Oscillate ขนาด 10.892 MHz จะได้จุดภาพ 697 จุดภาพ

ซึ่งหมายความว่า Counter จะต้องนับถึง 697 ครั้งในการกำเนิดสัญญาณทางแนวนอน (H sync) 1 pulse เราใช้ Monostable (IC#74LS221) เป็นตัวสร้าง Pulse ซึ่งความกว้างของ Sync pulse จากสัญญาณมาตราฐาน มีค่าเท่ากับ 5.12 microsec (ดูรูปที่ 5.10) ดังนั้นเราจะต้องทำการสร้าง Sync pulse ให้มีคาบเวลาเท่ากับคาบเวลามาตราฐานโดยการใช้R และ C ช่วยในการสร้าง (ดูรูปที่ 5.11)



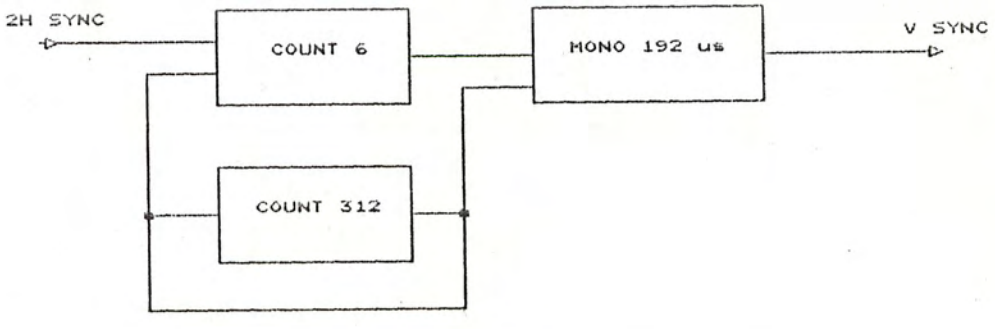
รูปที่ 5.10 สัญญาณมาตราฐานทางด้านแนวนอน(H sync)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 วงจรสร้างสัญญาณซิ้งค์ทางแนวนอน(H sync)

สำหรับการสร้างซิ้งค์ทางด้านแนวตั้ง(V sync) ก็เช่นกัน จะต้องเทียบกับสัญญาณมาตราฐานที่ความถี่ 50 Hz (2 ฟิลด์ต่อภาพ) จะมีการสแกน 625 line แต่ในโครงงานนี้มีการสแกน 1 ฟิลด์ต่อภาพ ดังนั้นต้องใช้ความถี่ 25 Hz และมีการสแกน 312.5 line การออกแบบก็จะต้องให้ Counter นับ 312 ครั้ง ในการสร้าง Pulse 1 pulse (รูปที่ 5.12)



รูปที่ 5.12 วงจรสร้างสัญญาณซิ้งค์ทางแนวตั้ง(V sync)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการสร้างสัญญาณอีควอไลซิ่ง(Equalizing)นั้น จะต้องให้มีความถี่เป็นสองเท่าของความถี่ทางด้านแนวนอน(H sync) เราสามารถสร้างได้โดยใช้ Counter นับ 348 ครั้ง เพื่อสร้าง pulse และนำสัญญาณนี้ผสมกับสัญญาณทางด้าน H sync อีกทีหนึ่ง(ดูรูปที่ 5.11) เพื่อนำสัญญาณไปใช้ในการสะบัดกลับของ H sync และแยก Odd/Even Field

<u>การออกแบบ Function การทำงาน</u>

เมื่อเราทำการอินเตอร์เฟสของระบบเข้ากับ IBM PC ดังนั้นจะต้องทำการ program หรือเลือกฟังค์ชั่นในการทำงาน ซึ่งจากรูปที่ 21 สามารถอธิบายการทำงานได้ดังนี้

Port 300A จะเป็นการติดต่อระหว่าง RAM ที่เก็บข้อมูลภาพขาวดำ กับ Data Bus โดยสามารถนำข้อมูลเข้าออกได้

Port 300B จะเป็นการติดต่อระหว่าง RAM ที่เก็บข้อมูลภาพสี กับ Data Bus โดยสามารถนำข้อมูลเข้าหรือออกได้เช่นกัน

Port 301 จะเป็นการติดต่อระหว่าง Data Bus กับ IBM PC เพื่อที่จะจัดจอภาพ

Port 302 จะเป็นการนำ Data Bus มาทำการ Decode เพื่อที่จะเลือกโหมดการทำงาน ซึ่งกำหนดโหมดการทำงานดังนี้

โหมด WN คือการนำสัญญาณภาพมาแสดงภาพที่จอโทรทัศน์และนำข้อมูลภาพเข้าไปเก็บใน RAM

โหมด RN คือการอ่าน RAM หรือการนำข้อมูลที่อยู่ใน RAM ไปแสดงภาพที่จอโทรทัศน์

โหมด WF คือการนำข้อมูลที่อยู่ในฟลอบปี้ดิสต์ หรือฮาร์ดดิสต์ เข้าไปเก็บใน RAM

โหมด RF คือการนำข้อมูลที่อยู่ใน RAM ไปเก็บในฟลอบปี้ดิสต์ หรือฮาร์ดดิสต์

โหมด Program คือ การ Set ค่าต่างๆเพื่อที่จะควบคุมจอโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

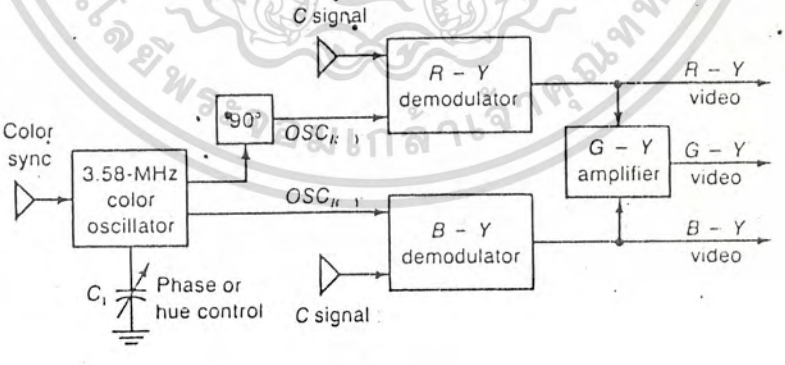
บทที่ 6

อธิบายรายละเอียดการทำงาน

จากหลักการทำงานขั้นพื้นฐานขั้นต้น และการคำนวณการสร้างของส่วนต่างๆ เมื่อเรานำออกแบบและทำการทดลองในวงจรแต่ละส่วนแล้ว ก็นำมาผสมผสานเข้าด้วยกัน ซึ่งในบทนี้จะอธิบายการทำงานในแต่ละส่วนย่อยอย่างละเอียด โดยที่จะอธิบายหน้าที่และฟังค์ชั่นต่างๆของอุปกรณ์ในแต่ละส่วน

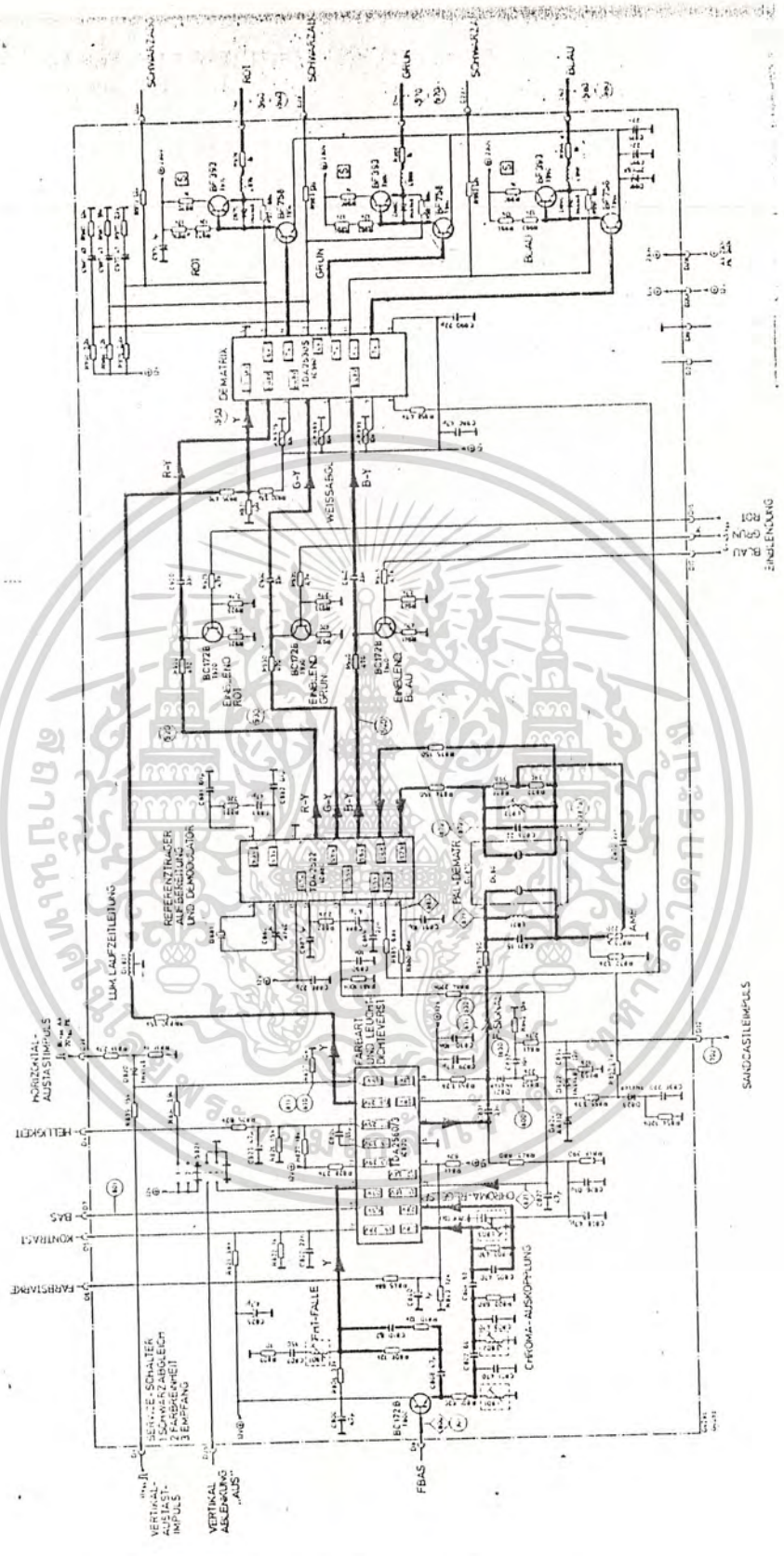
6.1 วงจรแยกสัญญาณภาพ (Separate Signal Circuit)

เมื่อเรานำสัญญาณภาพจากกล้อง VDO หรือจากอุปกรณ์กำเนิดสัญญาณ มาเพื่อที่จะนำมาแปลงสัญญาณจากสัญญาณอนาล็อคมาเป็นดิจิตอล สัญญาณที่ได้จากกล้อง VDO จะเป็นสัญญาณรวม ดังนั้นจึงจำเป็นต้องแยกสัญญาณแต่ละประเภทออกมา ซึ่งสัญญาณสามารถแยกออกมาได้ 3 ประเภท คือ สัญญาณภาพขาวดำ(Y) สัญญาณภาพสีR-Y และสัญญาณภาพสี B-Y วงจรที่แยกสัญญาณ จะเป็นวงจรที่ใช้ IC สำเร็จรูป ซึ่งเห็นได้จากวงจรโทรทัศน์สีในรูปที่ 6.1



รูปที่ 6.1 Block Diagram การแยกสัญญาณภาพ

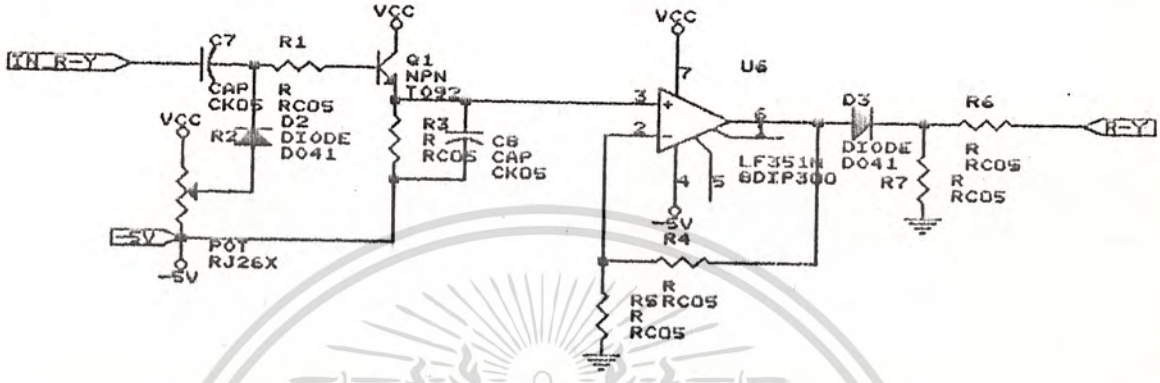
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 วงจรแยกสัญญาณภาพในวงจรโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

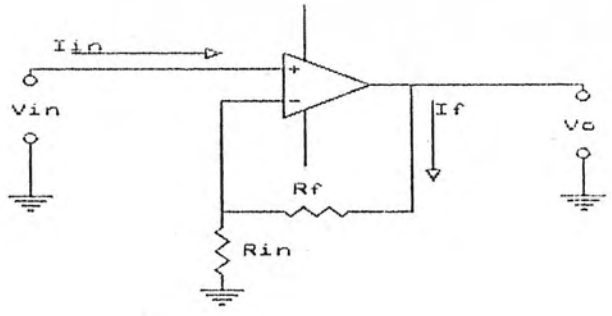
6.2 วงจรการเชื่อมต่อและขยายสัญญาณภาพ (Coupling and Signal Amplifier)



รูปที่ 6.3 วงจร Coupling และวงจรขยายสัญญาณ

ก่อนการนำสัญญาณภาพเข้าไปแปลงสัญญาณให้เป็นข้อมูล เราต้องนำสัญญาณที่ได้มาจากการแยกมาทำการCoupling เพื่อเชื่อมต่อสัญญาณเข้าด้วยกันโดยใช้ C_7 พร้อมกันนั้นจะต้อง Clamping ด้วย D_2 เพื่อที่จะกำจัดสัญญาณ H-Sync ดังรูป 6.3

หลังจากนั้นก็นำสัญญาณที่ได้มาทำการขยายโดยใช้ Op-Amp ซึ่งจะขยายแบบ Non-Inverter สำหรับอัตราการขยายจะเห็นได้จากการคำนวณข้างล่าง จากนั้นจะส่งสัญญาณไปยัง ADC



รูปที่ 6.4 วงจร Non-Inverting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_{in} = -I_f$$

$$I_{in} = V_{in}/R_{in}$$

$$I_f = -(V_{in} - V_o)/R_f$$

$$= (V_o - V_{in})/R_f$$

เมื่อ $V+ = V-$

ฉะนั้น $V_{in}/R_{in} = (V_o - V_{in})/R_f$

$$V_o = V_{in}(1 + R_f/R_{in})$$

ดังนั้น อัตราขยาย $V_o/V_{in} = (1 + R_f/R_{in})$

6.3 วงจรการแปลงสัญญาณอนาล็อคเป็นดิจิตอล หรือ ADC (Analog to Digital Converter)

สัญญาณภาพเมื่อต้องการนำมาเก็บในหน่วยความจำจะต้องนำสัญญาณภาพที่เป็นอนาล็อคแปลงเป็นข้อมูลทางดิจิตอล เนื่องจากสัญญาณ VDO มีค่าเวลาที่ใช้ในการสแกนภาพเท่ากับ 64 us ต่อ หนึ่งLine ซึ่งเราต้องการให้ได้ความละเอียดของจุดภาพเท่ากับ 512 จุด * 512 จุด ดังนั้นเราจำเป็นต้องแบ่งช่วงเวลาในการเก็บข้อมูลแต่ละครั้งดังการคำนวณข้างล่างนี้

การสแกน 1 จุดภาพจะใช้เวลา > 64 us/512 จุดภาพ

> $1.25 * 10^{-07}$ sec/pixel

ดังนั้นจึงต้องใช้ความถี่ในการ Sampling > $1/1.25 * 10^{-07}$ Hz

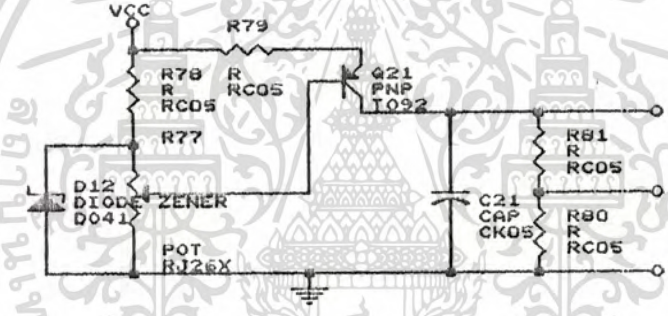
> 8 MHz

จากการคำนวณจะเห็นได้ว่า เราจะต้องใช้ ADC ที่มีความเร็วสูงมากกว่า 8 MHz ซึ่งการคำนวณอย่างละเอียดเราแสดงให้เห็นในบทที่ 3 สำหรับบทนี้เราจะแสดงให้เห็นเฉพาะการหา ADC ที่มีความเร็วอยู่ในย่านที่สามารถตอบสนองการย้อนกลับ (Convert.) ได้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการศึกษาเราพบว่า ADC แบบ Flat จะให้ความเร็วในการแปลงสัญญาณ(Convert) อยู่ในย่านที่ต่ำกว่า 20 MHz (ความถี่ในช่วงนี้จะเป็นอัตรา (Rate)สูงสุดในการ Run)

สำหรับการสร้างแรงดัน Refference ให้กับ ADC นั้น เราจะสร้างในลักษณะที่ปรับระดับแรงดัน Refference ได้ตั้งแต่ 0 ถึง 4 Volt โดยการจัด Bias ให้กับ PNP Transistor ซึ่งเราสามารถแบ่งช่วงแรงดันได้โดยใช้หลักการ Divider จะเห็นได้จากวงจรรูปที่ 6.5



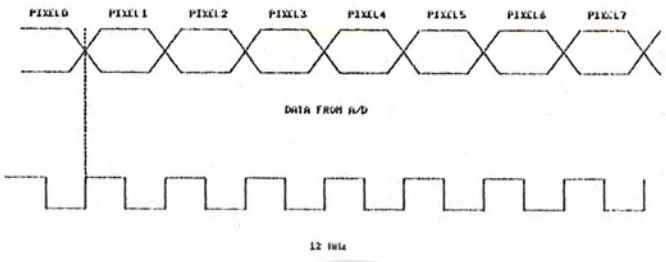
รูปที่ 6.5 วงจรแบ่งแรงดัน(Divider)

ในการ sampling ภาพ จะใช้สัญญาณนาฬิกา (Clock Pulse) 12 MHz สัญญาณนี้จะ sampling ภาพโดยตรง โดยการ sampling ของ DAC ทั้ง 3 ชุด จะทำงานพร้อมๆกัน ซึ่งจะเห็นได้จาก Timing Diagram รูปที่ 6.6

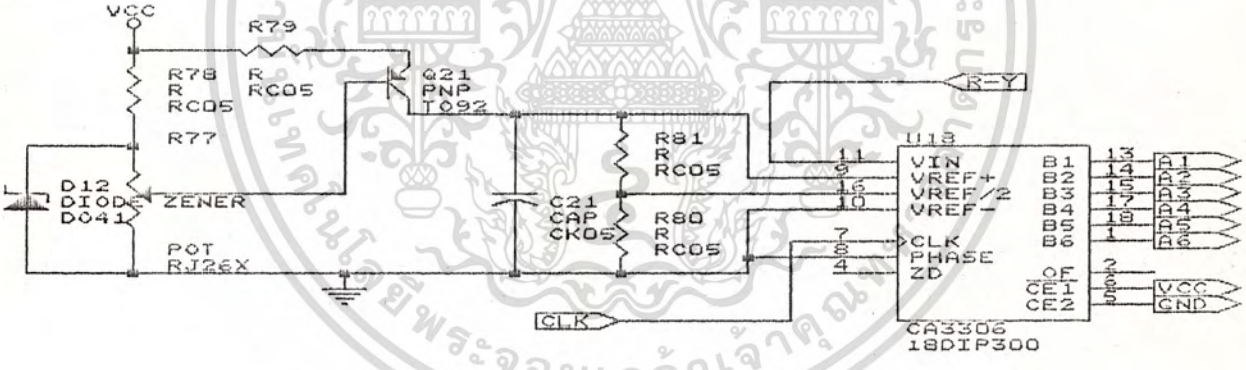
จากสัญญาณภาพที่ได้ทั้ง 3 ชุด และจากการคำนวณเราจะใช้ ADC แปลงสัญญาณภาพขาวดำ(Y) ขนาด 8 บิท เบอร์ 3318 ซึ่งจะให้ระดับของสัญญาณทางดิจิตอล $2^8 = 256$ ระดับ

ส่วนสัญญาณภาพสีR-Y และB-Y จะผ่าน ADC ขนาด 6 บิท เบอร์ 3306 ซึ่งจะให้ระดับของสัญญาณทางดิจิตอล $2^6 = 64$ ระดับ แต่ Bus ของข้อมูลจะรับรองไว้ 8 บิท เพื่อที่จะปรับปรุงใช้ในอนาคตข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 Timing Diagram



รูปที่ 6.7 วงจรแปลงสัญญาณอนาล็อคเป็นดิจิตอล (ADC)

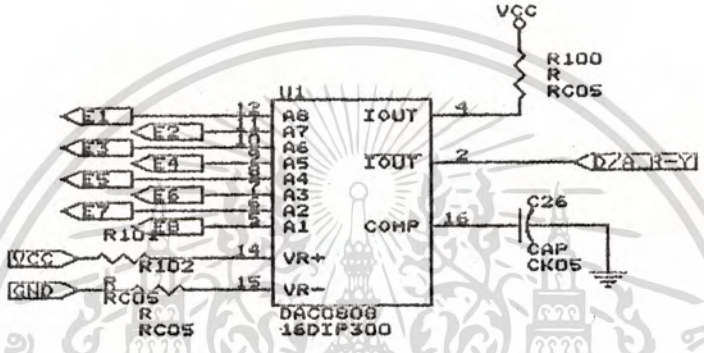
จากรูปที่ 6.7 เมื่อสัญญาณภาพถูกแปลงเป็นข้อมูลทางดิจิตอลแล้ว ข้อมูลจะถูกส่งผ่านไปยัง Buffer ซึ่งแต่ละชุดจะแยกจากกันอย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 วงจรการแปลงข้อมูลดิจิตอลเป็นสัญญาณอนาล็อค หรือ DAC

(Digital to Analog Converter)

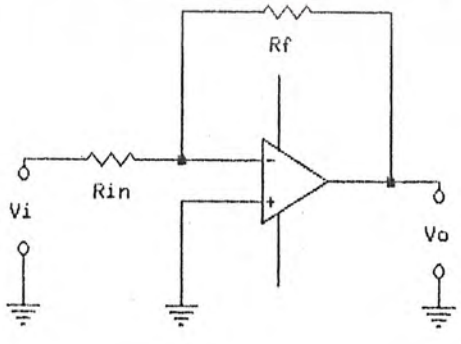
การนำข้อมูลจากหน่วยความจำ หรือจากแป้นข้อมูลมาแสดงบนจอภาพ จะต้องแปลงข้อมูลทางดิจิตอลเป็นสัญญาณอนาล็อค (DAC) โดยกำหนด DAC ขนาด 8 บิท เบอร์ 0808 ทั้ง 3 ชุด ซึ่งก่อนนำข้อมูลเข้า DAC จะต้องผ่าน Buffer



รูปที่ 6.8 วงจรแปลงข้อมูลดิจิตอลเป็นสัญญาณอนาล็อค (DAC)

6.5 การขยายสัญญาณและการผสมสัญญาณภาพกับสัญญาณซิงค์

เนื่องจากสัญญาณที่ออกมาจาก DAC จะเป็นสัญญาณแบบกลับ(Invert) ดังนั้นจึงต้องผ่าน วงจรขยายสัญญาณแบบ Inverter ซึ่งจะมีอัตราการขยายดังการคำนวณข้างล่างนี้



รูปที่ 6.9 วงจร Inverting

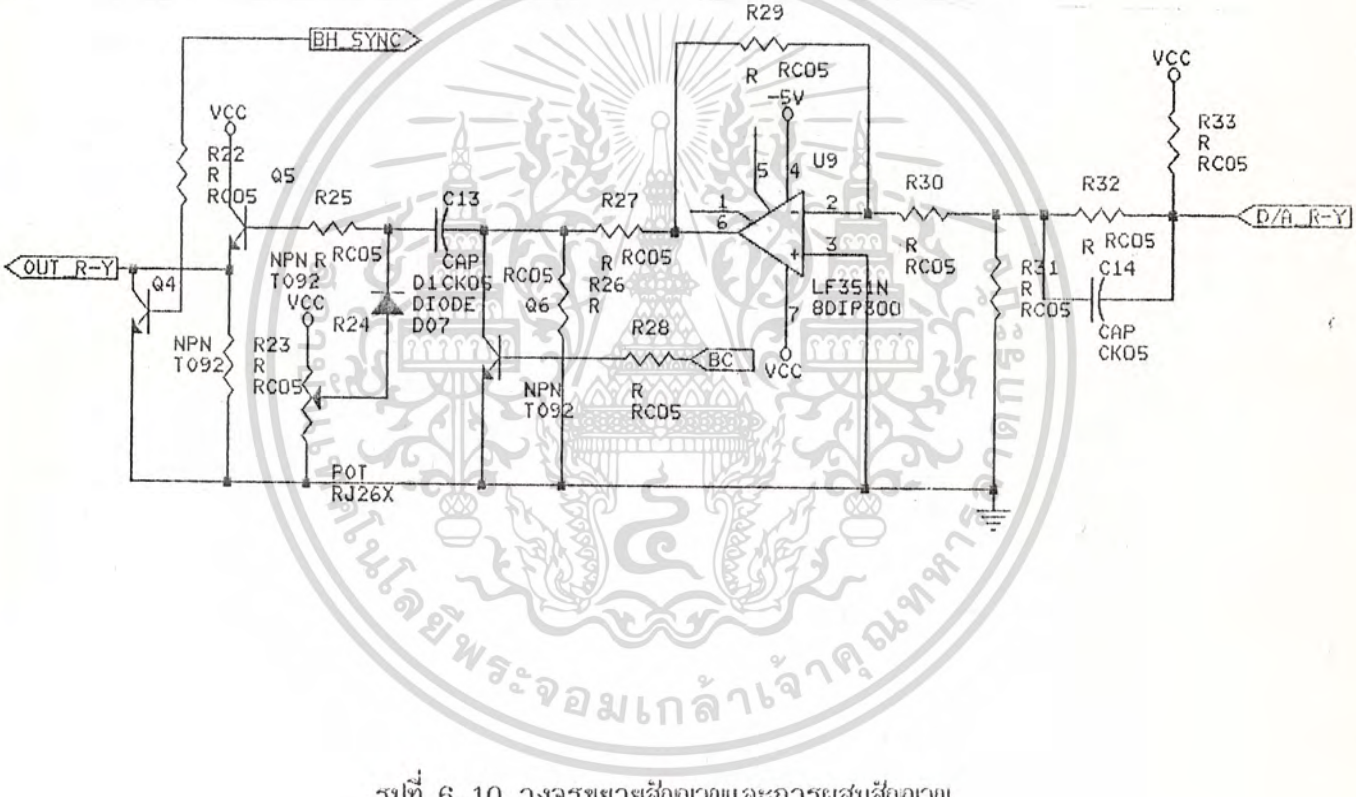
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป $I_{in} = -If$

ดังนั้น $V_{in}/R_{in} = -Vo/R_f$

ฉะนั้น $V_o = -V_{in}(R_f/R_{in})$ Volt

อัตราการขยาย$(V_o/V_{in}) = -R_f/R_{in}$



รูปที่ 6.10 วงจรขยายสัญญาณและการผสมสัญญาณ

จากการคำนวณจะเห็นว่า อัตราการขยายของสัญญาณจะขึ้นอยู่กับความต้านทาน R_{29} และ R_{27} โดยที่จะมี R_{32} และ C_{14} เป็นชุด Filter เพื่อกรอง Noise ที่เกิดขึ้น และเมื่อได้สัญญาณ Out put ของ Op-Amp แล้วจะทำการ Coupling โดยใช้ C_{13} และจะยกระดับสัญญาณขึ้นโดย RC05 และ D_1 จากนั้นก็นำสัญญาณไปผสมกับสัญญาณ H Sync เพื่อที่จะนำไปออกสู่จอทีวี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 วงจรการเก็บข้อมูลในหน่วยความจำ (Memory Circuit)

จากข้อมูลใน Data Bus เมื่อต้องการทำการ Digitizer หรือต้องการนำข้อมูลไปเก็บในหน่วยความจำ(Memory) เราสามารถทำได้โดยการกำหนดทางด้าน Soft ware เพื่อที่จะไปทำการปิด-เปิดทางผ่านของข้อมูลและการ Control RAM ในการเขียนหรืออ่านข้อมูล

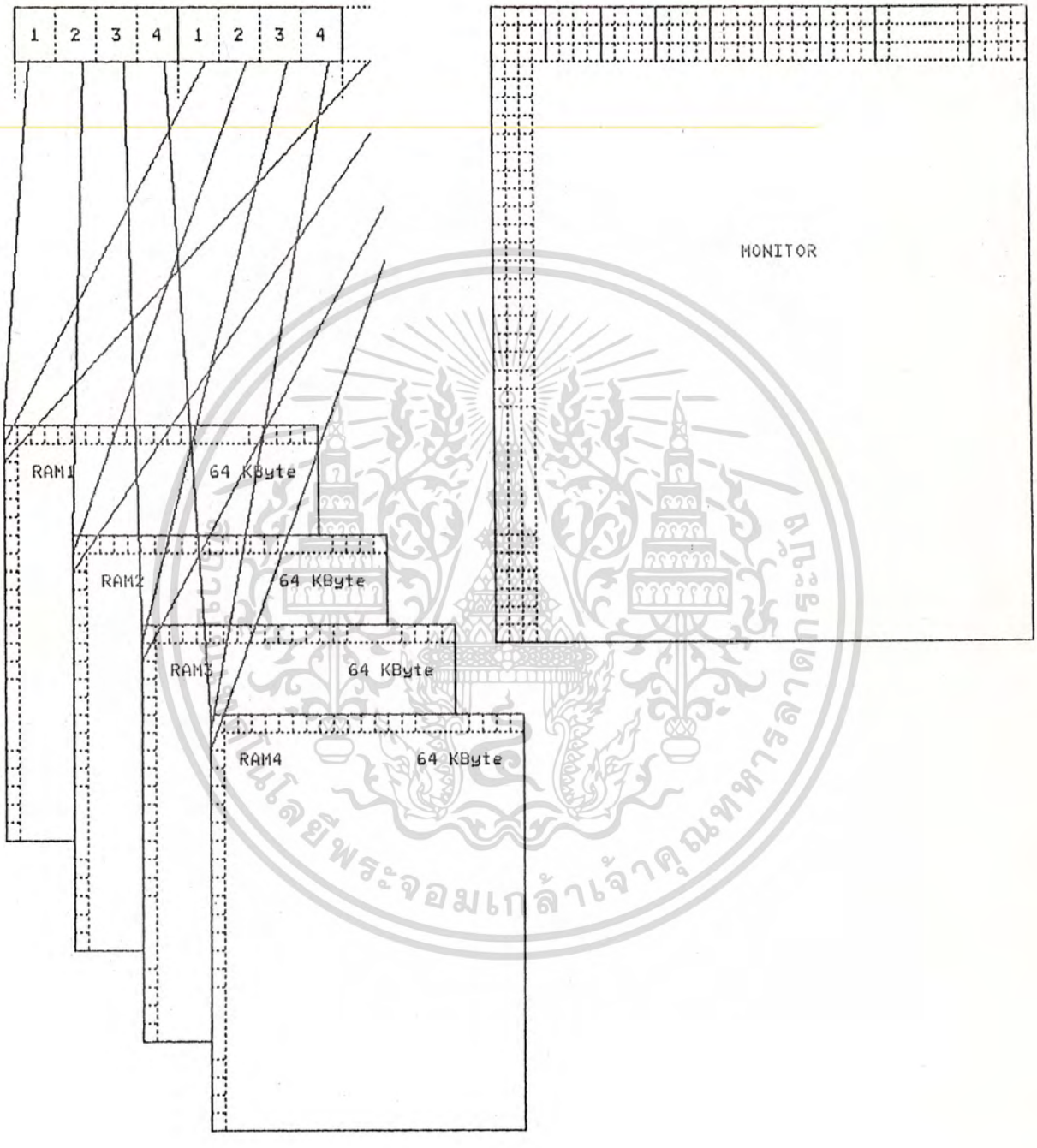
หน่วยความจำที่ประกอบด้วย RAM 3 ชุดนั้นแต่ละชุดจะมีความสามารถเก็บข้อมูลหรือมีหน่วยความจำ 256 Kbytes การสร้างหน่วยความจำนี้ จะใช้ RAM ชนิด Dynamic(D-RAM) ซึ่งคุณสมบัติต่างๆก็ได้อธิบายมาแล้วในตอนต้น

เพื่อให้เกิดความเข้าใจมากยิ่งขึ้นเกี่ยวกับการทำงานของชุด RAM ดังนั้นจึงข้ออธิบายเกี่ยวกับการเก็บและ Display ออกมา ดังในรูปที่ 6.11 ซึ่งสัญญาณภาพทางแนวนอนจะแบ่งออกเป็น 128 ชุด ชุดละ 4 pixels ต่อๆกันไป การวางข้อมูลในหน่วยความจำ จะแบ่ง RAM ออกเป็น 4 ชุดเช่นกัน แต่ละชุดจะมี RAM 2 ตัว ซึ่ง RAM แต่ละชุดจะรับข้อมูลในแต่ละ pixels

จากวงจรรูปที่ 6.12 เราใช้ D-RAM เบอร์ 4464 ซึ่งแต่ละตัวจะมีความจุ 64Kbytes 4 บิท แต่ข้อมูลที่ได้จาก ADC มีขนาด 8 บิท ดังนั้นเราจึงใช้ D-RAM2 ตัว เพื่อให้ขนาด 8 บิท และมีความจุ 64 Kbytes จะมี Address Line 16 เส้นในการเก็บภาพแต่ละชุดจะประกอบด้วย D-RAM 8 ตัว สำหรับการทำงานของ D-RAM ก็ได้อธิบายในบทขั้นต้นแล้ว แต่ในบทนี้จะอธิบายการทำงานของวงจร ในการอ่านและเขียนข้อมูลภาพกับหน่วยความจำ

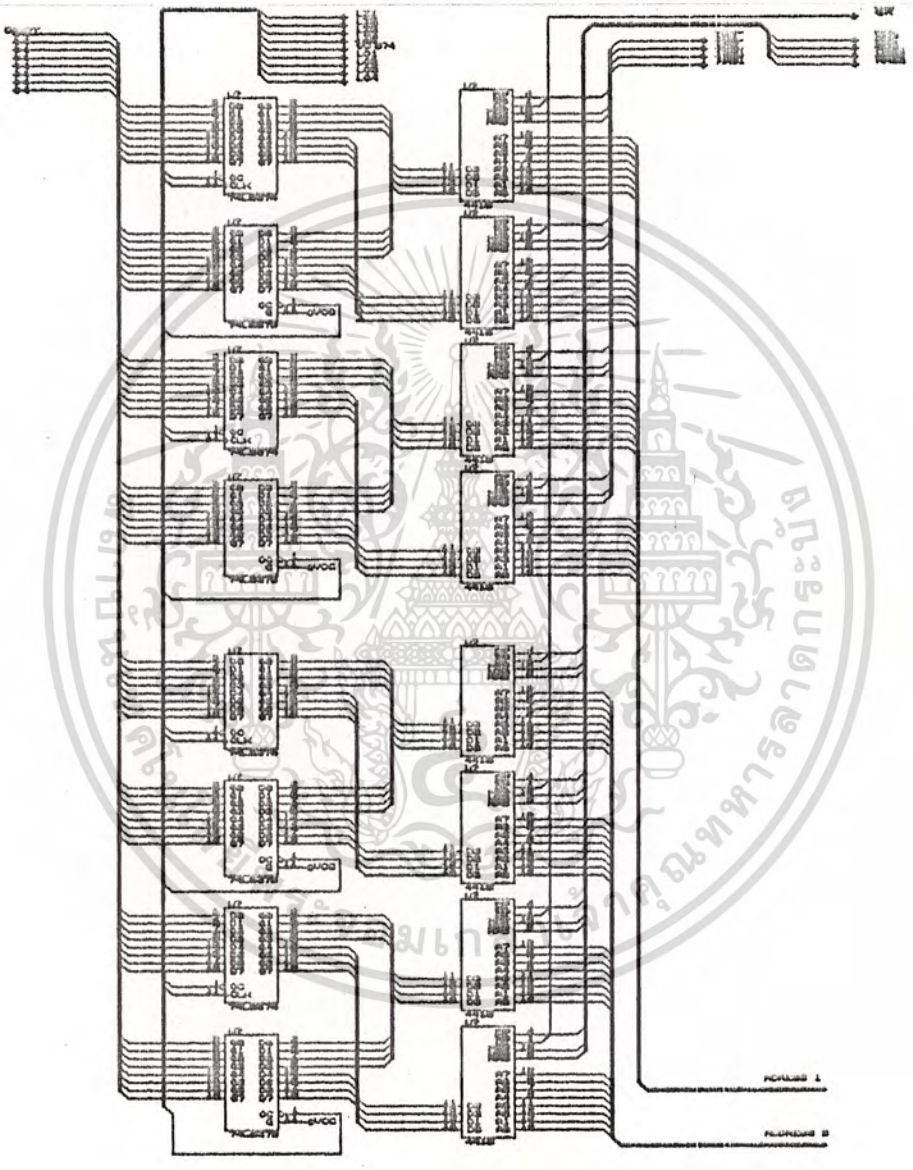
จะเห็นได้ว่า Address Line มี 16 เส้น เพื่อที่จะสร้าง Row Address และ Column Address ซึ่งแยกออกเป็นชุดละ 8 เส้น ต่อ D-RAM 1 ตัว เมื่อต้องการเขียนข้อมูลลงไปใน RAM เราต้องให้ RAS และ CAS อยู่ในลักษณะ Pulse ขาลง ให้ WR เป็นลอจิก "0" และให้ GR เป็นลอจิก "1" ในการเขียนข้อมูลลงไปใน RAM จะไม่สามารถเขียนพร้อมกันทุกตัวได้เพราะว่าถ้าเขียนพร้อมกันจะทำให้ข้อมูลซ้ำซ้อน และAddress ไม่ตรงกัน ดังนั้นเราจึงใช้ Latch ซึ่งเป็นตัวคงค่าข้อมูล เป็นตัวกำหนดการทำงานของ RAM แต่ละตัว โดยหลักการทำงานได้แสดงไว้แล้วในบทขั้นต้น จะเห็นได้จากรูปที่ Timing Diagrame ของ Latch In และ Latch Out

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.11 แบบจำลองของการเก็บภาพในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

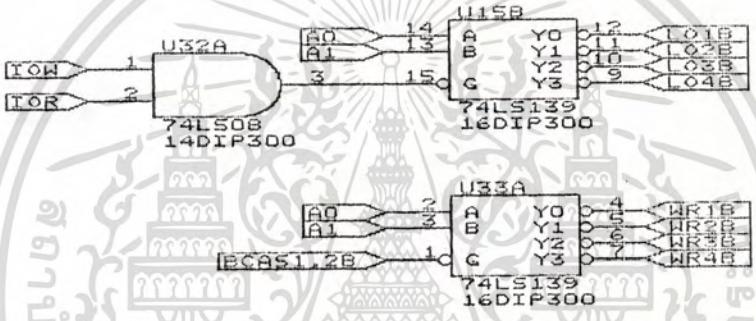


รูปที่ 6.12 วงจร Memory ที่ใช้ D-RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 การสร้าง Address ให้กับ D-RAM

ตามปกติแล้วเมื่อ Memory มีขนาดความจุ 256 Kbytes จะต้องใช้ Address Line ไม่ต่ำกว่า 18 เส้น แต่เรานำเข้าใน D-RAM 16 เส้น ดังนั้นอีก 2 เส้น (A_0 และ A_1) เราจะนำมาทำการดีโค้ต (Decode) เพื่อที่จะไปทำการควบคุมฟังค์ชั่นต่างๆ เช่น ควบคุมการทำงานของ Latch In, Latch Out และการเขียนข้อมูลลงไปใน RAM ซึ่งวงจรการ Decode เราจะเห็นได้จาก รูปที่ 6.13



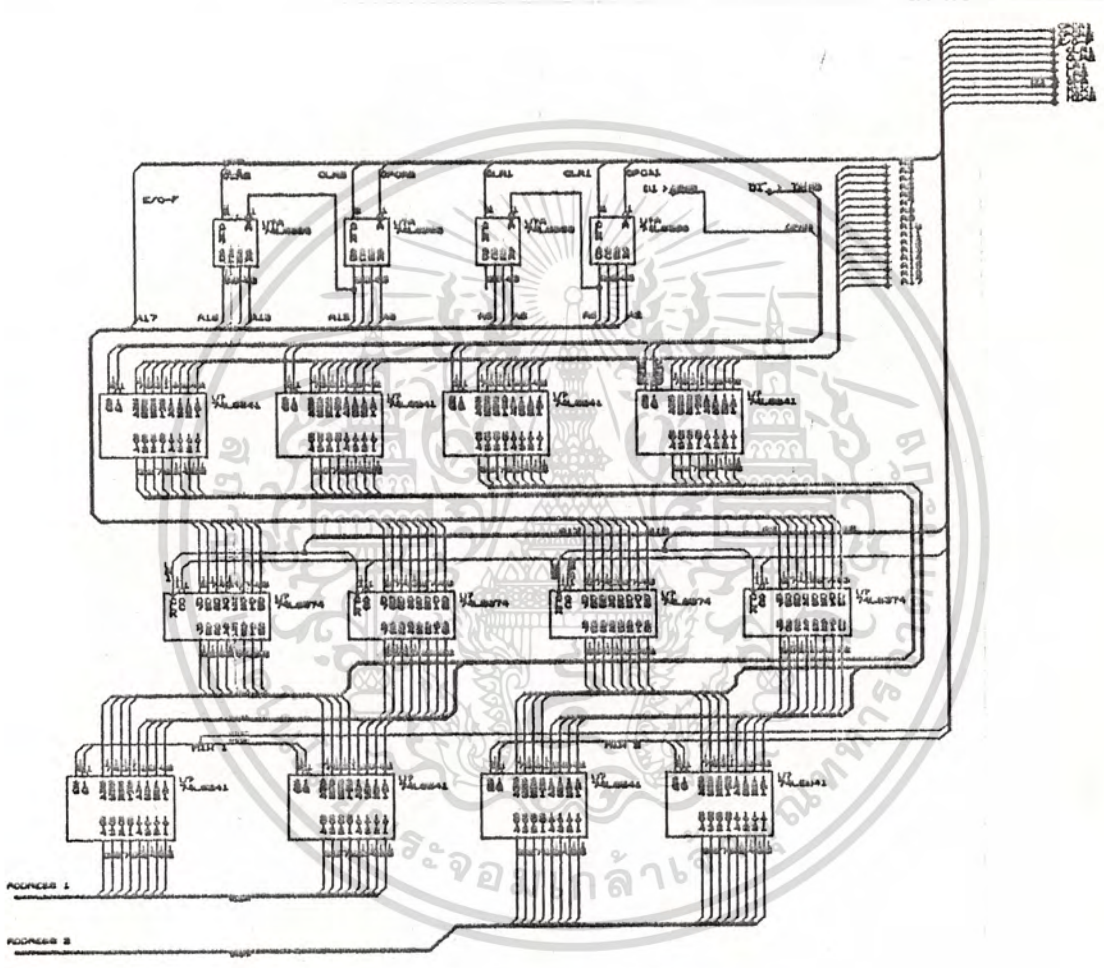
รูปที่ 6.13 วงจรการ Decode

การสร้าง Address ของ D-RAM สามารถสร้างได้ 2 สภาวะ คือ การสร้างโดยใช้วงจรภายนอก(สร้างขึ้นเอง) และการสร้างโดยการเชื่อมต่อกับ Bus Address ของComputer

กรณีการสร้างAddress จากการสร้างขึ้นเอง เราสามารถสร้างได้โดยการใช้Counterเป็นตัวนับ ซึ่งจะใช้เบอร์ 74LS393 จากวงจรรูปที่ 6.14 โดยสัญญาณที่นำมากระตุ้น U_{22} และ U_{20} คือ CPCA1 และ CPCA2

การกำหนด Address Line ที่สร้างขึ้นจาก Counter จะเริ่มที่ A_2 ถึง A_{17} สำหรับ Pluseที่ใช้ในการสร้าง Address นำมาจากสัญญาณ MIX2 และสัญญาณ H Sync และที่ A_{17} จะเป็นสัญญาณ E/O-F เพื่อที่จะใช้ในการติดต่อ RAM ใน Page ของ Odd Feed หรือ Even Feed จากรูปที่ 6.14 Address Line จะต่อเชื่อมกับ Latch โดยที่จะค้างข้อมูลไว้เพื่อรอการ Mix Address ที่ Buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



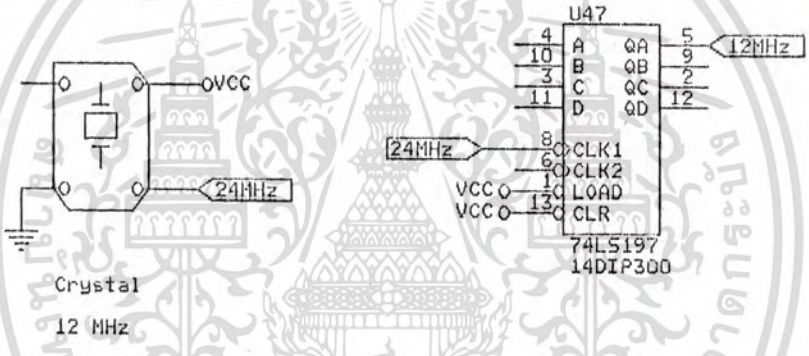
รูปที่ 6.14 การสร้าง Address Line ให้กับ D-RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน A_0 และ A_1 จะไม่มีการสร้างขึ้นมา แต่จะใช้การ Shift Register ในการควบคุม Latch in และ Latch out โดยใช้สัญญาณ clock pulse 12 MHz ในการควบคุม Shift Register

สำหรับ Address Line ที่ต่อเชื่อมกับ Bus ของ Computerนั้น จะมี Buffer เป็นตัวปิด-เปิดก่อนนำไป Mix กับ Address ที่สร้างขึ้นเอง

6.8 วงจรการกำเนิดสัญญาณนาฬิกา (Clock Pluse Generation)



รูปที่ 6.15 วงจรกำเนิดสัญญาณนาฬิกา

โครงงานนี้เราใช้ Crystal เป็นตัวกำเนิดสัญญาณนาฬิกา ซึ่งคุณสมบัติของ Crystal ก็คือเมื่อป้อนแรงดันค่าคงที่แร่ Crystal จะเกิดการโก่งตัวงอไป-งอมา ทำให้เกิดการแกว่งของสัญญาณ จากรูป 6.15 เราใช้ Crystal ขนาด 24.00 MHz แต่ที่ใช้จริงประมาณ 12 MHz ดังนั้นจึงใช้ Counter เบอร์ 74LS167 เป็นตัวหาร ในที่นี้จะทำการหารสอง ดังนั้นจะมีความถี่ 12.0 MHz ดังแสดงในรูป 6.15

6.9 การกำหนดฟังค์ชั่นของ RAM และการสลับฟังค์ชั่น

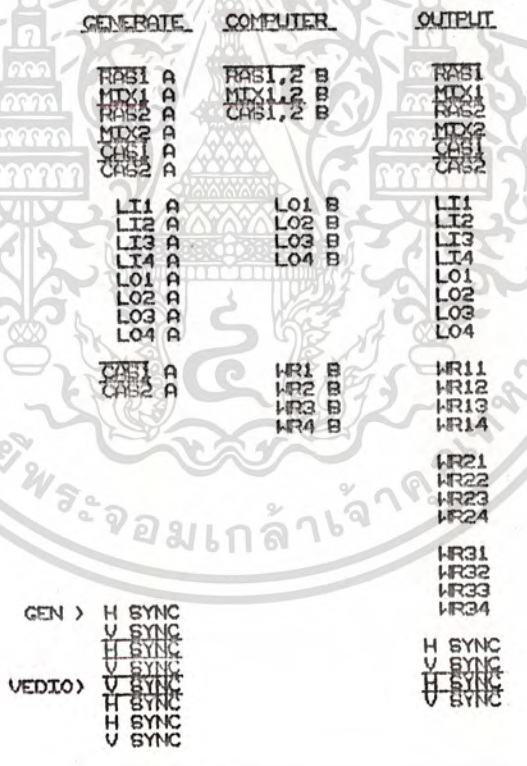
ในระบบการทำงานจากที่เคยอธิบายมาแล้วว่า การจัดการทำงานของ Image interface control สามารถแบ่งได้เป็น 4 ฟังค์ชั่น คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแสดงภาพหรือการอ่านข้อมูลจาก RAM (RN)
- การเก็บภาพจากกล้องหรือการเขียนข้อมูลลงไปใน RAM (WN)
- การโหลดภาพหรือการอ่านข้อมูลจาก File (RF)
- การเก็บภาพหรือเขียนข้อมูลลงไปใน File (WF)

ในการควบคุมการทำงานของฟังค์ชั่นของRAM สัญญาณควบคุมจะแบ่งออกเป็น 2 ส่วน คือ

ส่วนแรก สัญญาณควบคุมจะสร้างจากสัญญาณ Sync ที่สร้างขึ้นและที่แยกออกมาจากกล้องวิดีโอ เพื่อที่จะมาควบคุมการทำงานของ RAM และสร้าง Address ให้แก่ RAM ด้วย ซึ่งการสร้างสัญญาณควบคุมดูได้จากสัญญาณอินพุทของวงจรสลับฟังค์ชั่นในรูปที่ 6.16

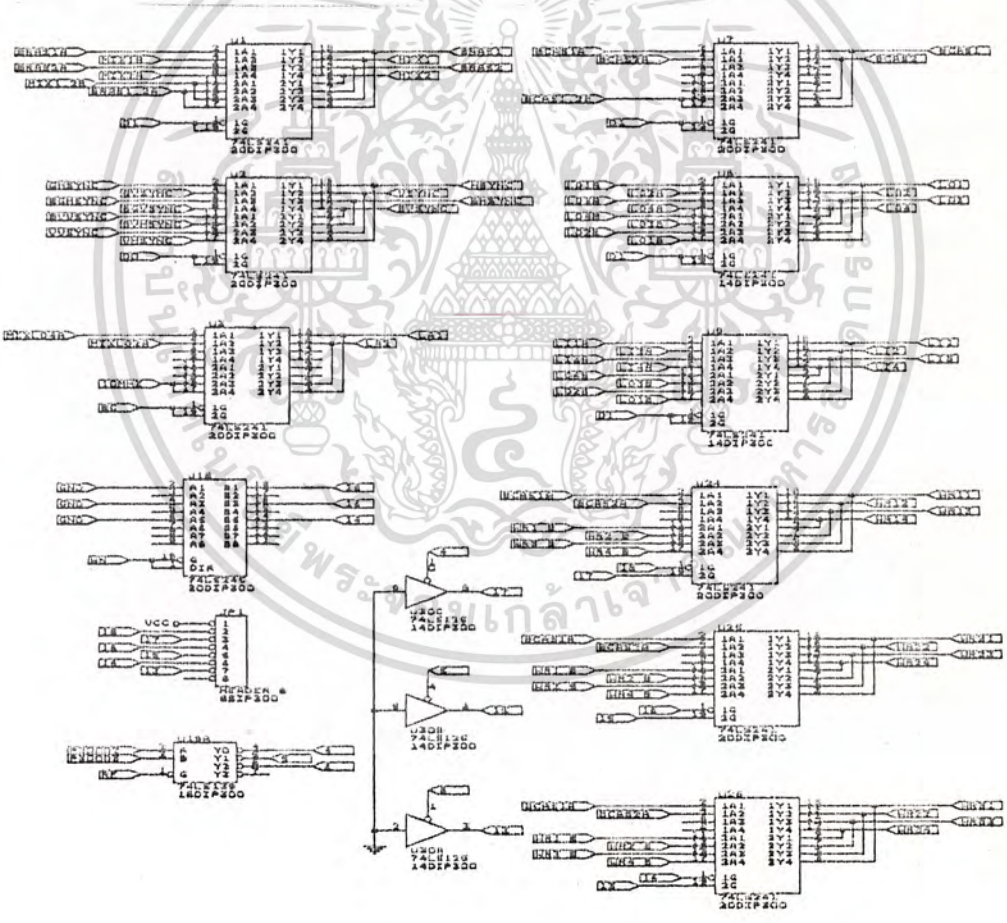


รูปที่ 6.16 สัญญาณอินพุทของวงจรสลับฟังค์ชั่น

สำหรับการใช้งานในส่วนนี้จะขึ้นอยู่กับฟังค์ชั่นการเก็บภาพจากกล้องลงไปใน RAM (WR)และ การแสดงภาพบนจอ TV หรือการอ่านข้อมูลจาก RAM (RN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สอง จะนำสัญญาณจาก Computer คือสัญญาณ I_{OR} และ I_{OW} เพื่อที่จะมาสร้างสัญญาณควบคุม การทำงานของ RAM และควบคุมการสร้าง Address ที่ส่งจาก computer ไปยัง D-RAM ในช่วงนี้จะใช้งานเมื่ออยู่ในฟังค์ชั่นการเก็บภาพไว้ใน File (WF) หรืออ่านข้อมูลจาก File ลงไปเก็บไว้ใน RAM (RF)ซึ่งจะใช้งานเมื่ออยู่ในฟังค์ชั่นการเก็บภาพไว้ใน File (WF) หรือ การอ่านข้อมูลภาพจาก File ลงเก็บไว้ใน RAM (RF)

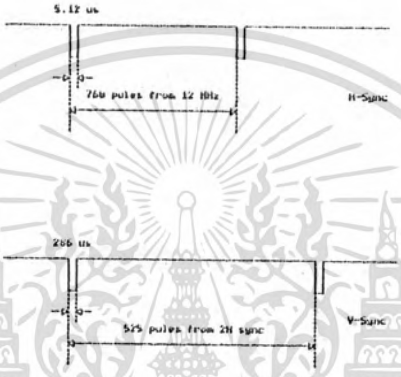


รูปที่ 6.17 วงจรควบคุมในการสลับฟังค์ชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.10 การสร้างสัญญาณซิ้งค์ทางแนวนอน (H-Sync)และแนวตั้ง(V-Sync)

ในการกำเนิดสัญญาณ Sync ทั้ง H-Sync และ V-Sync ที่ใช้ในฟังค์ชั่นการอ่านข้อมูลภาพจาก RAM (RN) ซึ่งกำเนิดโดยวิธีนับ (Count) การนับ จะใช้สัญญาณนาฬิกา 12 MHz จากรูปที่ 6.18 จะมีวิธีการนับคือ



รูปที่ 6.18 Timing แสดงการกำเนิดสัญญาณ Sync

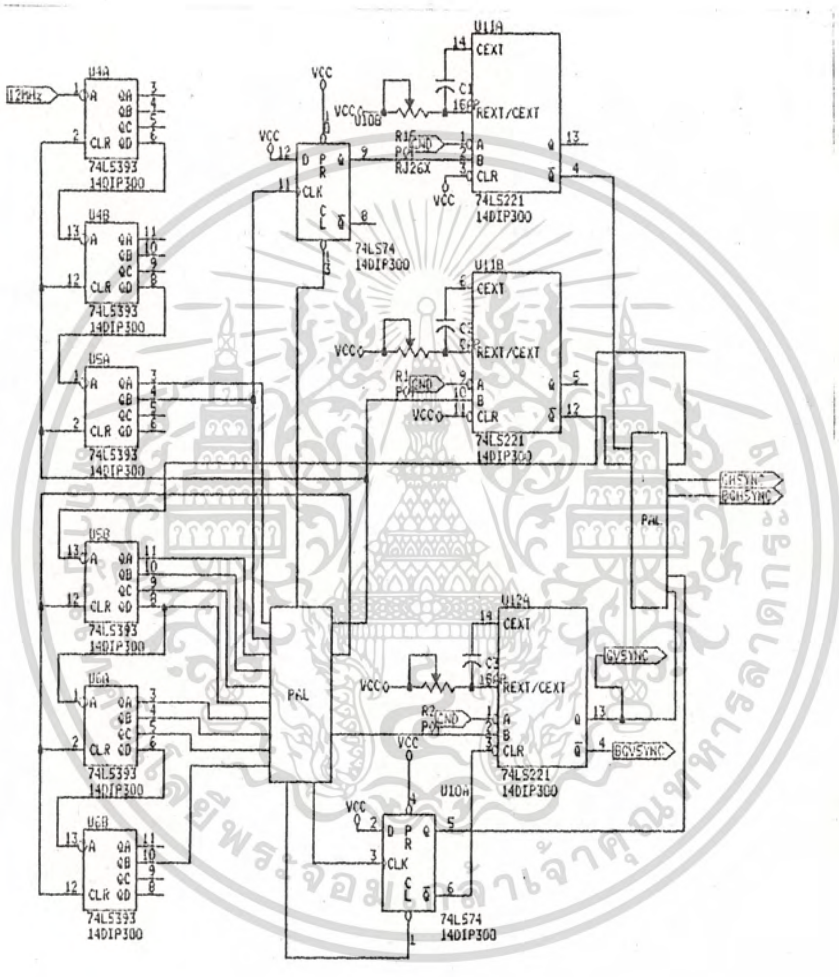
จาก Timing จะเห็นว่าการนับจะแบ่งได้เป็น 2 ช่วง คือ ช่วงแรกจะเป็นการนับเพื่อกำเนิดสัญญาณ H-Sync และช่วงที่ 2 จะเป็นการนับเพื่อกำเนิดสัญญาณ V-Sync

ในการนับของ Counter เพื่อใช้ในการกำเนิดสัญญาณ H-Sync นั้นจะนับ 2 ครั้งเพื่อที่จะได้สัญญาณ H-Sync 1 Pluse ซึ่งค่าแรกจะนับ 200H เพื่อนำไปใช้ในการสร้างสัญญาณอีควอไลท์ซิ่ง(Equalizing Pluse) และการนับครั้งที่สอง ซึ่งจะนับ 300H เพื่อที่จะสร้าง H-Sync ใช้ในวงจร

การสร้าง V-Sync จะเริ่มที่การนับจากสัญญาณความถี่2 เท่าของ H-Sync ซึ่งจะมีการนับ 625 ครั้ง จึงจะได้สัญญาณ V-Sync 1 Pluse

สัญญาณซิ้งค์ที่ใช้งานจริงจะถูกสร้างจากวงจร PAL โดยการนำสัญญาณ H-Sync และV-Sync ที่สร้างขึ้นมาประกอบกัน

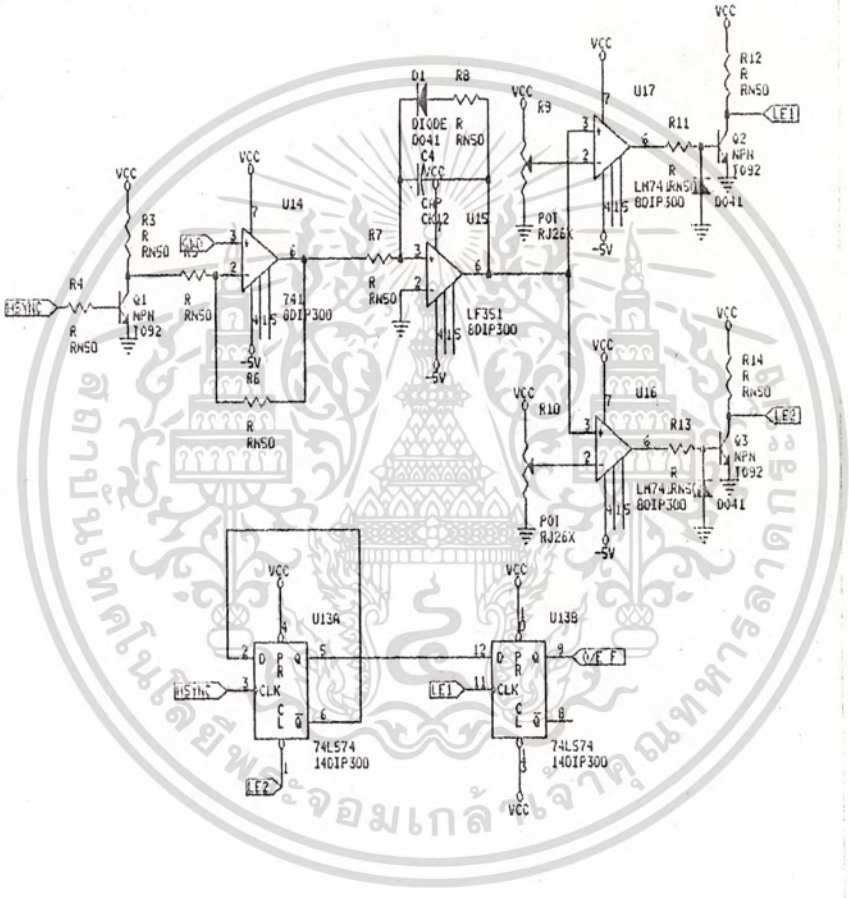
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.19 วงจรการสร้างสัญญาณซิงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.11 วงจรการสร้างสัญญาณ Odd-Even Feed



รูปที่ 6.20 วงจรการสร้างสัญญาณ Odd-Even Feed

จากรูปที่ 6.20 จะใช้ H-Sync ที่ได้จากกล้อง VDO หรือจากการสร้างขึ้น ซึ่งเราจะนำมาเช็คก่อนว่าเป็นสัญญาณ Odd หรือว่าเป็นสัญญาณ Even Feed ซึ่งจะทำการเช็คอยู่ตลอดเวลา ซึ่งการแยก Odd-Even Feed จะใช้หลักการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนับสัญญาณ H-Sync ในช่วงเริ่มต้นของสัญญาณอิควอไลท์ซิ่ง คือถ้านับได้ 6 ครั้ง จะได้Feed ต่อไปเป็น Even

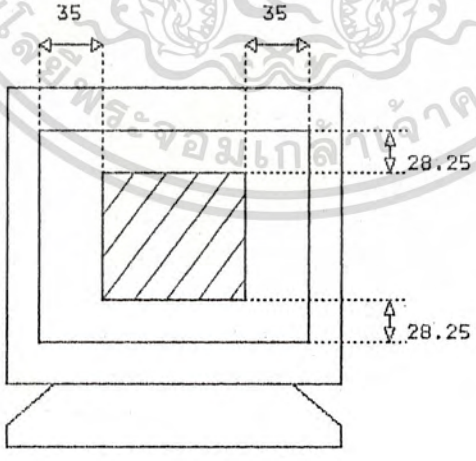
การตรวจจับการนับ(Detect Count)จะใช้วงจร Integrat โดยการนำ H-Sync มาเข้าวงจร Inverter และเข้าวงจร Integrat จากนั้น Out Put ที่ได้ก็จะนำมาทำการตรวจเช็ค 2 ระดับ จากรูปจะเห็นว่าเราจะใช้ Op-Amp 2 ตัวเป็นตัวตรวจเช็ค ซึ่ง Out Put ที่ได้จะนำไปควบคุมการนับของ D-Fiflop ทั้ง 2 ตัว Out Put จากวงจรนี้จะเป็นสัญญาณ O-E/F ซึ่งจะนำไปใช้เป็น Address Line สูงสุดของชุด RAM (A_{17})

6.12 วงจรกำหนดการเริ่มต้นการเก็บภาพ (Cover)

การเริ่มต้นการเก็บสัญญาณภาพ(Cover)มี 2 ลักษณะด้วยกัน คือ

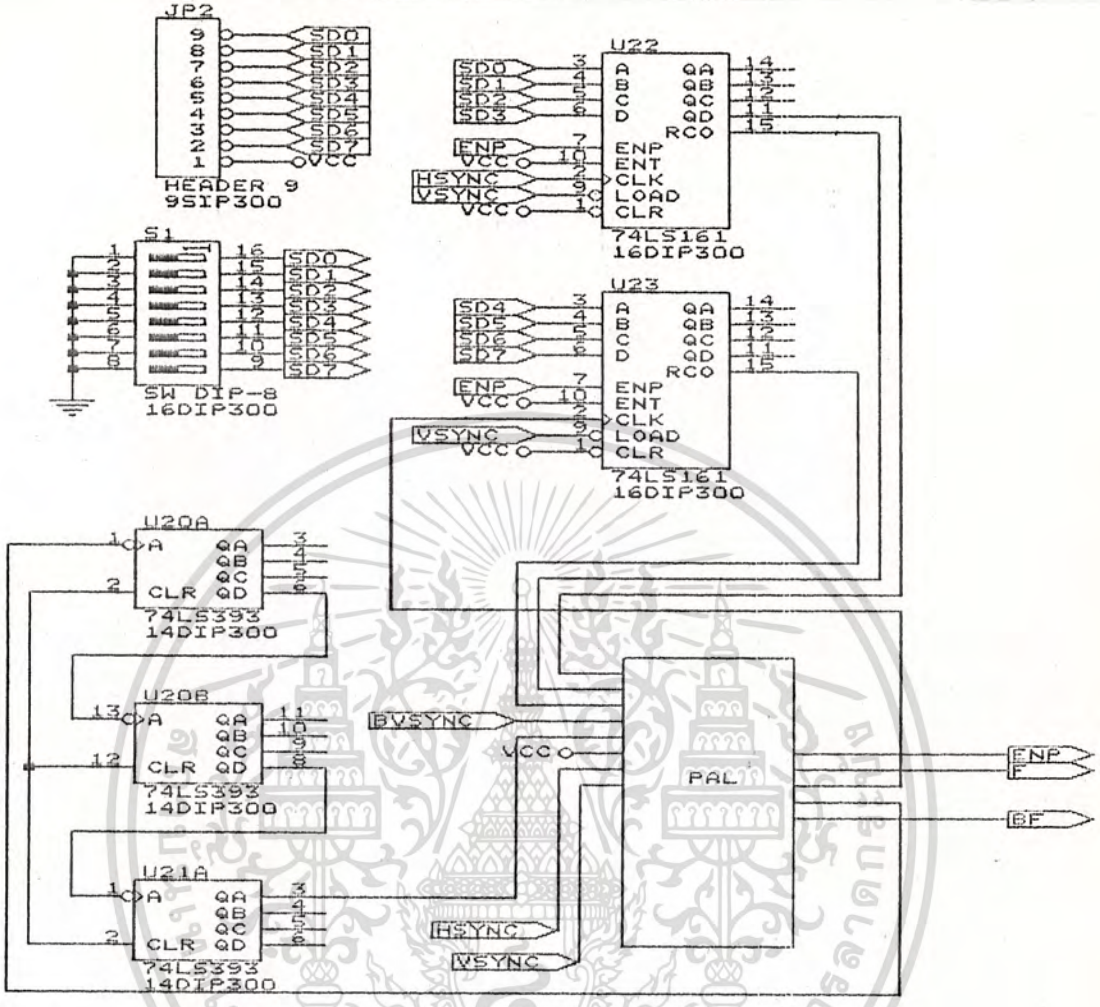
- Cover Line (Horizotal)
- Cover Feed (Vertical)

ทั้ง 2 ลักษณะนี้สามารถที่จะโปรแกรมค่าเริ่มต้นได้โดย Dip Switch เพื่อที่จะไปควบคุมตำแหน่งต่างๆบนจอภาพ



รูปที่ 6.21 การกำหนดค่าเริ่มต้นของสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.22 วงจรการกำหนดค่าเริ่มต้นของสัญญาณภาพ

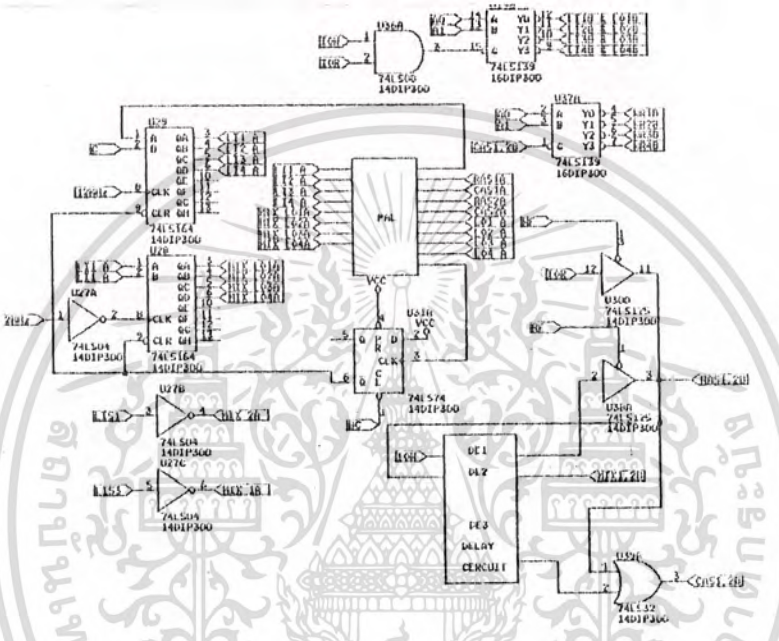
จากรูปที่ 6.21 วงจรเราจะทำการ Cover แบบการนับลง (n,..2,1,0) โดยจะใช้ Counter เป็นตัวนับ ซึ่งจากรูปที่ 6.21 เราสามารถกำหนดค่าเริ่มต้นได้โดยการควบคุมระยะ x โดยใช้ Dip Switch เป็นตัวกำหนด

การเริ่มต้นนับจะเริ่มที่ H Sync ซึ่งจะใช้ความถี่ในการนับ 12 MHz ในส่วนนี้จะเป็นการเริ่มในแนวนอน ส่วนการนับทางแนวตั้งจะนับจำนวน H Sync ว่ามีจำนวนที่ Pluse จากรูป เมื่อนับค่าเริ่มต้นแล้ว (ระยะ x) ก็จะทำการนับต่อ 512 ครั้ง ซึ่งจะได้จุดภาพ 512 จุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.13 วงจรสร้างสัญญาณควบคุม RAM

สัญญาณที่ใช้ในการติดต่อ RAM ได้แก่ สัญญาณ RAS, CAS, MIX สัญญาณเหล่านี้จะเป็นส่วนสำคัญมากในการติดต่อ RAM ไม่ว่าจะเป็นการอ่าน (Read) หรือการเขียน (Write) เข้าไปใน RAM

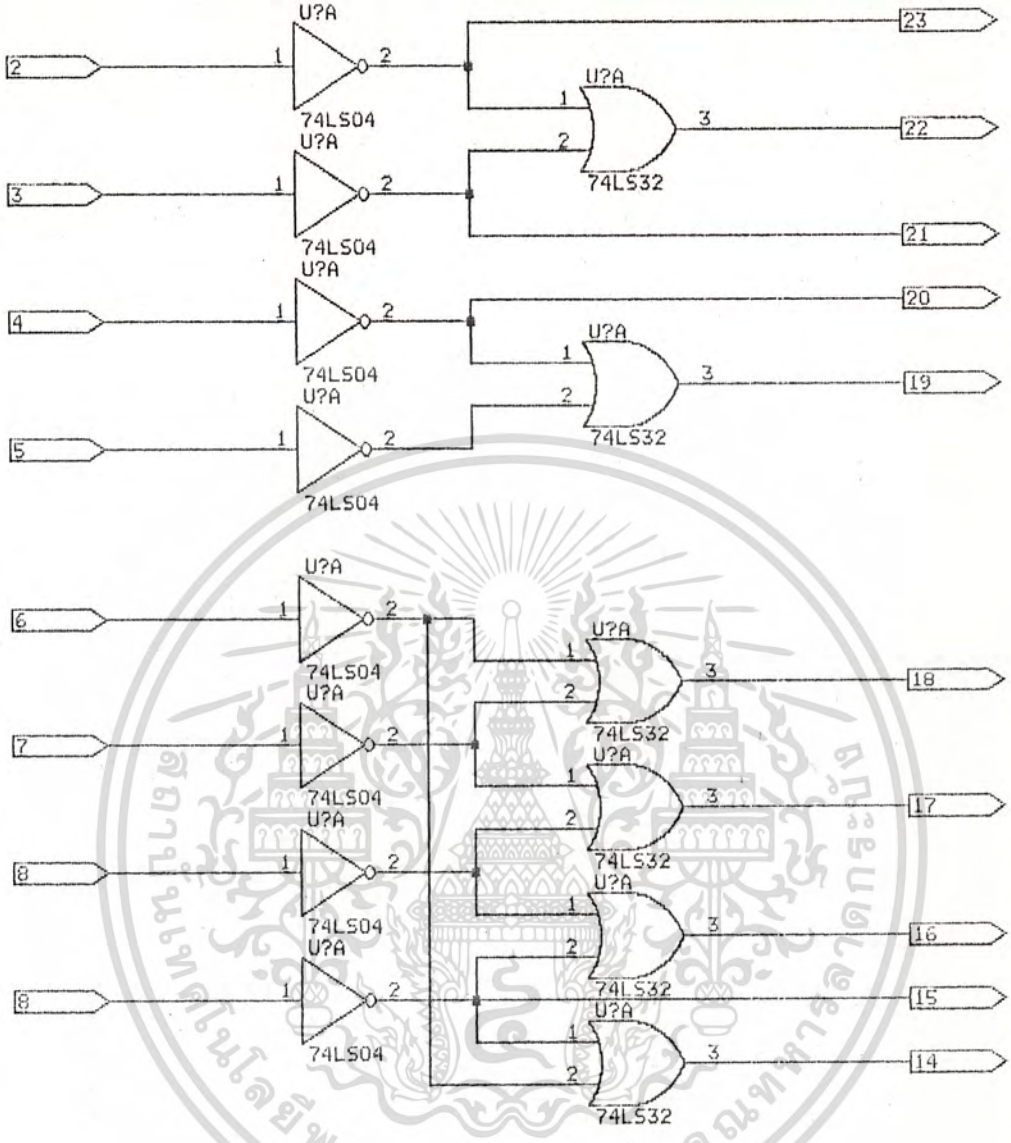


รูปที่ 6.23 วงจรสร้างสัญญาณควบคุม RAM

ในการสร้างสัญญาณควบคุม RAM จะมีลักษณะการติดต่อ RAM 2 ลักษณะ คือ

1. ติดต่อโดยใช้สัญญาณซิ้งค์ที่มาจากกล้องหรือมาจากวงจรสร้างสัญญาณซิ้งค์ เพื่อที่จะนำสัญญาณเหล่านี้มาใช้ในการกำเนิดสัญญาณควบคุม Out Put ที่ได้จากวงจรกำหนดสัญญาณเริ่มต้น (Cover) จะเป็นตัวเริ่มต้นในการกำเนิดสัญญาณ ซึ่งการทำงานจะใช้ Shift-Register ดังแสดงในรูปที่ 6.23 สัญญาณในการ Shift จะใช้สัญญาณนาฬิกา 12 MHz และข้อมูลในการ Shift จะใช้ 2 BIT ที่ตามกัน (โดยใช้ Out Put Q_2 นำมาเป็น In Put ของ Shift-Register สัญญาณที่ได้จะมีความถี่ 1/4 เท่าของสัญญาณนาฬิกา 12 MHz) ซึ่ง Out Put ที่ใช้จะมีอยู่ 4 line คือ Q_1, Q_2, Q_3, Q_4 ซึ่งจะนำมาเข้าวงจร PAL ตามรูปที่ 6.24 ซึ่งก็จะได้สัญญาณ Out Put เป็นสัญญาณ RAS และ CAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



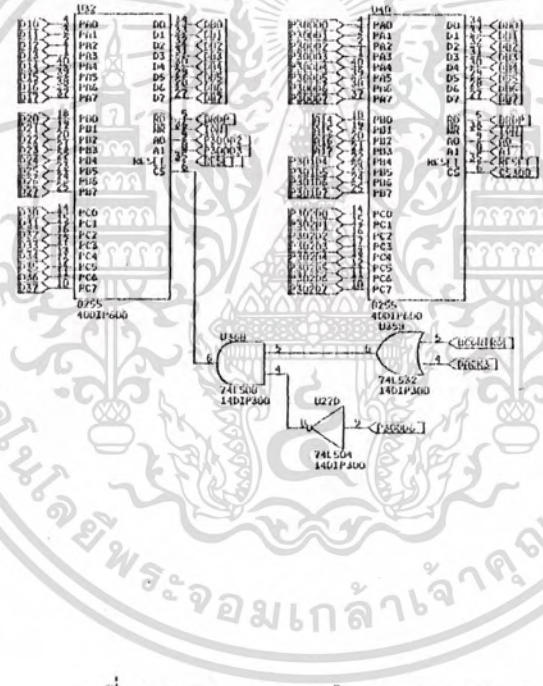
รูปที่ 6.24 วงจรลอจิกภายใน PAL

2. การติดต่อ RAM โดยใช้สัญญาณจาก Slot ของ IBM PC มาสร้างสัญญาณควบคุม (ใช้สัญญาณ IOR และ IOW รวมทั้ง Address A_0 และ A_1) ซึ่งจะใช้ สัญญาณ IOR และ IOW ผ่านวงจร Delay 2 ถึง 3 ครั้ง เพื่อที่จะสร้าง สัญญาณ RAS, CAS และสัญญาณ MIX สัญญาณ A_0และA_1 จะนำมาผ่านวงจร Decoder เพื่อสร้างสัญญาณในการติดต่อ RAM สัญญาณ Out Put ที่ได้ก็จะเหมือนกับส่วนแรก จากนั้นนำสัญญาณนี้ไปควบคุมการเลือกใช้ในวงจรเปลี่ยนฟังค์ชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.14 วงจร INTERFACE

วงจรนี้จะใช้กับ Port 300,301,302 และ303 ซึ่งจากวงจรรูปที่ 6.25 จะใช้ไอซี เบอร์ 74LS688 ในการ Decode Port หมายเลข300 ถึง303 โดยจะใช้ Dip Switch เป็นตัวโปรแกรมหมายเลข Port ว่าจะใช้ Port หมายเลขอะไร การใช้ Port จะใช้ตัวไอซี 2 ตัว ซึ่งตัวแรกจะใช้งานในการส่งผ่าน DATA ระหว่าง DATA BUS ของ Computer กับ 3 DATA BUS ของระบบ



รูปที่ 6.25 วงจรการทำงานของ Port

สำหรับ Port ตัวที่ 2 จะใช้ในงานควบคุมระบบ ซึ่ง Port A จะ Detect หมายเลข 300 การใช้งาน 2 บิทแรก(D_0และD_1) จะใช้สำหรับการควบคุมฟังค์ชั่น 2 บิทถัดมา(D_2และD_3) จะใช้ในการเลือกชุด RAM ที่จะติดต่อซึ่งจะมี R-Y,B-Yและ Y สำหรับ D_4 จะใช้เมื่อเริ่มต้นที่จะใช้ Image Card นี้ สำหรับ D_5 จะใช้เป็นตัว Reset System โดยการใช้ Software และสำหรับ D_6 จะใช้ควบคุม Port 8255 ตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Port ตัวที่ 2 (Port 300) จะเห็นได้จากฟังค์ชั่นข้างล่างนี้

D_1	D_0	Function
0	0	RN
0	1	WN
1	0	RF
1	1	WF

****

D_3	D_2	Select-Port
0	0	R-Y
0	1	B-Y
1	0	Y
1	1	Write Control Port

****

D_4 Count Control System

****

D_5 Control second Port

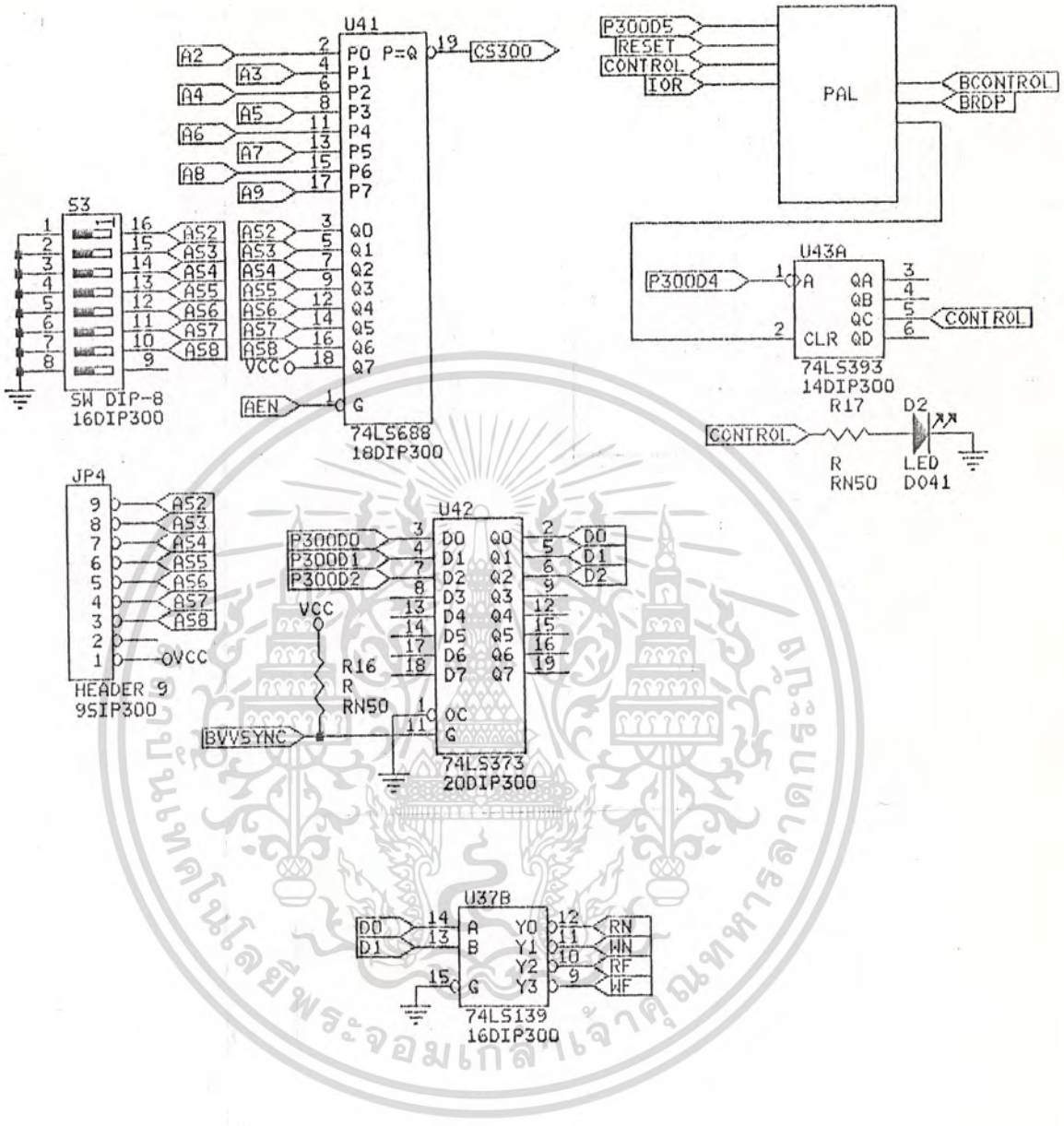
****

D_6 Enable second Port

****

สำหรับ PortB หมายเลข 301 จะสร้าง Address เพิ่ม A_{14} ถึง A_{17} เพื่อใช้แทน Address ที่มาจาก Slot IBM PC สำหรับสัญญาณที่ใช้ในการ Interface กับ Slot มี DACK 3, IOW, IOR, AEN, RESET, A_0-A_{13} และ D_0-D_7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.26 วงจรการ INTERFACE

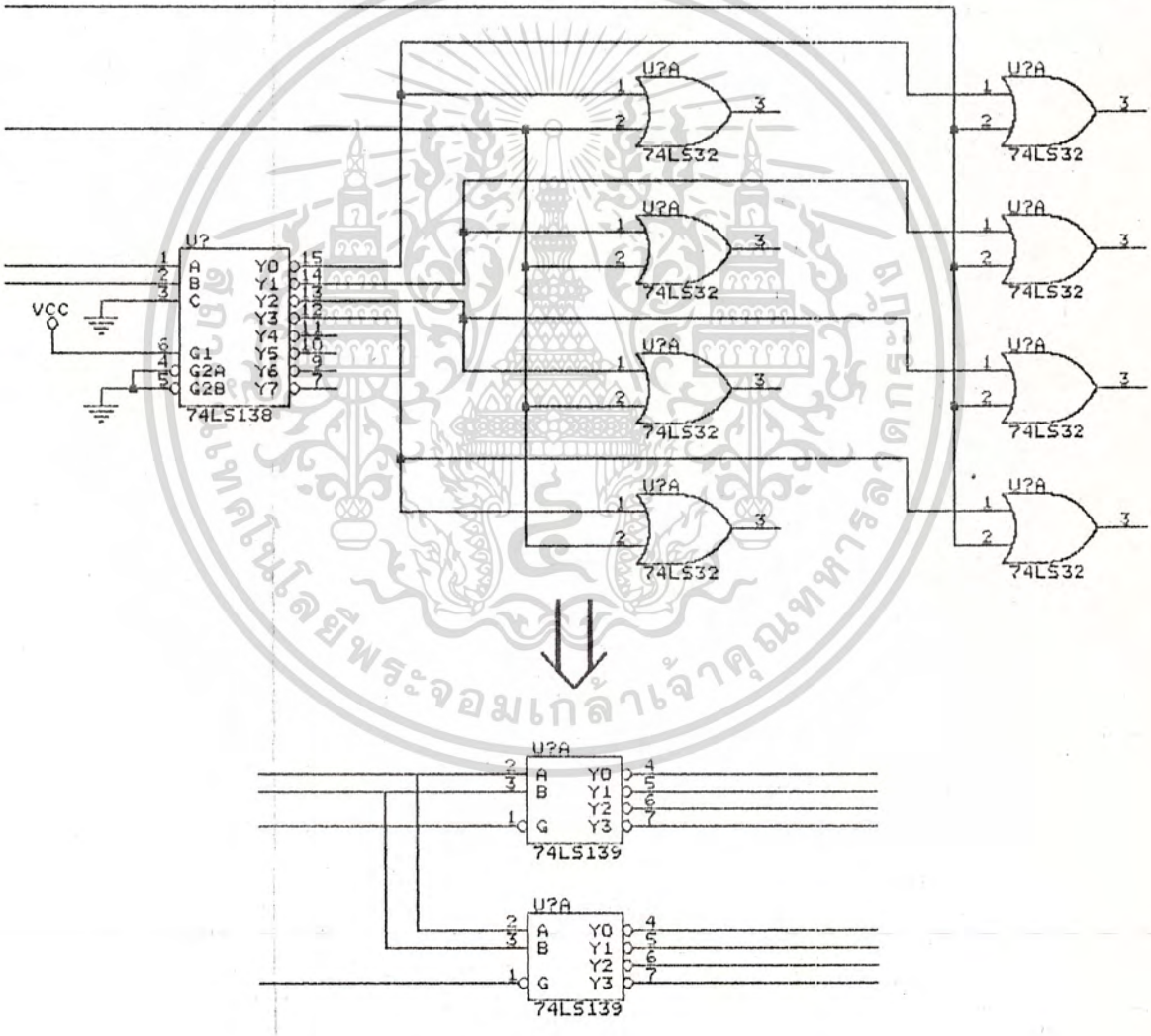
การเข้าถึงระบบ(Initial System) จะใช้วิธีการ Reset ด้วย D_5 ของ Port หมายเลข 301 แล้วส่งสัญญาณไปนับที่ D_4 จำนวน 8 ครั้ง ระบบก็สามารถเริ่มการทำงานได้ การอ่านสภาวะการทำงานจะอ่านที่ Port 302 D_1 เพื่อที่จะตรวจเช็คว่าระบบทำงานหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

หลักการนำ PAL (Programmable Array Logic) มาใช้งาน

เนื่องจากการออกแบบวงจรทางด้าน Logic ของ Project ใช้อุปกรณ์มากทำให้การออกแบบ PCB (Print Circuit Board) มีพื้นที่มาก และเป็นการสิ้นเปลือง จึงต้องมีการหาวิธีการลดจำนวนอุปกรณ์ลง ซึ่งวิธีแรกที่ใช้คือการ Simplifly Logic Gate ดังรูปที่ 7.1



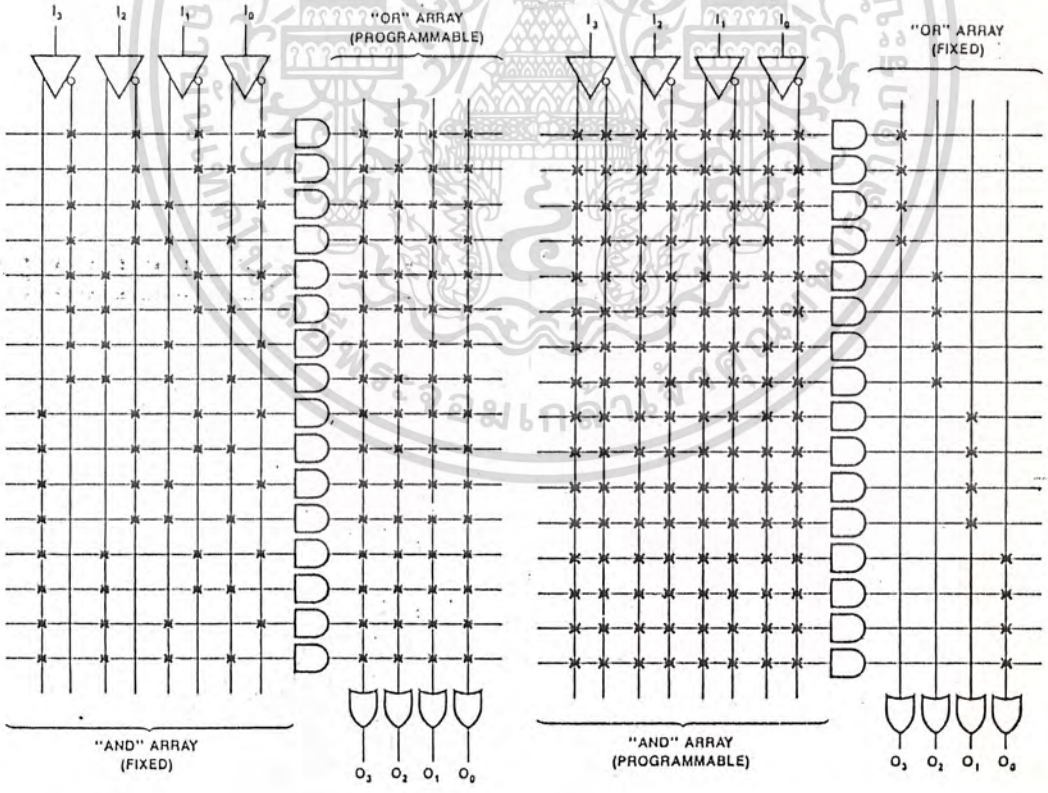
รูปที่ 7.1 การ Simplifly Logic Gate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรเป็นการออกแบบวงจร Control Read Write ที่ Memoryของวงจรซึ่งเป็นการออกแบบครั้งแรก จะเห็นว่าเป็นการนำ Logic Gate มาใช้ถึง 3 ตัว จึงเป็นการสิ้นเปลืองอย่างมาก ส่วนวงจรที่ 2 เป็นการออกแบบใหม่โดยการใช้ Logic Gate เพียง 1 ตัวแต่วิธีนี้ใช้ได้กับบางวงจรที่ออกแบบไว้แล้วเท่านั้น แต่ก็ยังเหลืออีกบางวงจรในวงจร Main ที่ไม่สามารถกระทำได้ ดังนั้นจึงต้องหาวิธีการที่ใช้ในการลดจำนวนอุปกรณ์ ซึ่งวิธีการที่ใช้ต่อไปคือการใช้ไอซี PAL (Programmable Array Logic)

7.1 คุณสมบัติของไอซี PAL

PAL มีลักษณะโครงสร้างภายในตรงกันข้ามกับ PROM คืออินพุตของ AND เป็นแบบโปรแกรมได้ ส่วนอินพุตของ OR จะคงที่ดังตัวอย่างในรูปที่ 7.2



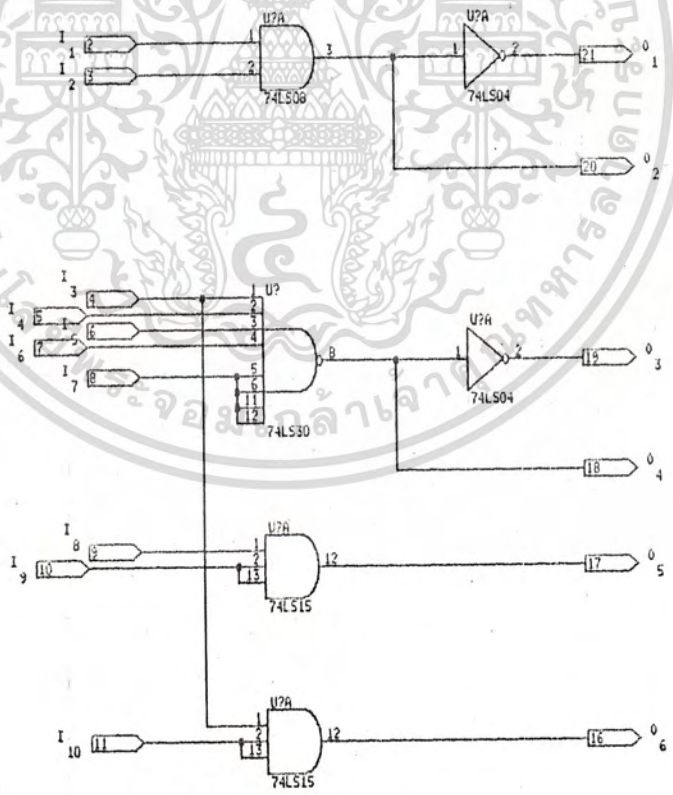
รูปที่ 7.2 PROM and PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแรกแสดงโครงสร้างภายในของ PAL ขนาด 4 อินพุต 4 เอาต์พุต จากตัวอย่างนี้สังเกตว่าสัญญาณอินพุต I_0-I_3 จะส่งผ่านบัฟเฟอร์และอินเวอร์เตอร์ ออกมาเป็นสัญญาณ 8 เส้น (I_0-I_3,I_0-I_3) ป้อนเข้า AND ทุกตัว ซึ่ง AND ทุกตัวจะต้องมีอินพุตอยู่ 8 ขาโดยที่สามารถโปรแกรมได้ว่าจะต่อเข้ากับ I_0-I_3,I_0-I_3 เส้นใด ส่วนเอาต์พุตของ AND จะถูกจัดแบ่งเป็น 4 ชุด ป้อนเข้า OR อย่างตายตัว คือเอาต์พุต 4 ตัวบนจะป้อนเข้า OR ตัวแรก ถัดลงมา 4 ตัวจะป้อนเข้า OR ตัวที่ 2 เป็นเช่นนี้ลงมาเรื่อย ๆ ในกรณีนี้หมายถึงว่า OR จะต้องมีอินพุตอยู่ 4 ขาเท่านั้น

7.2 ขั้นตอนการทำ

วงจรที่ใช้ในการทำ PAL มีทั้งหมด 5 วงจร โดยวงจรแรกเป็นวงจร Sync Generater 1



รูปที่ 7.3 Sync Generater Circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรถ้าใช้ Logic Gate ธรรมดามาต่อกันต้องใช้จำนวนไม่น้อยกว่า 3 ตัว แต่เราจะออกแบบและใช้ไอซี PAL เพียง 1 ตัวเท่านั้นมาทำงานแทน

เริ่มต้นต้องเลือกเบอร์หรือชนิดของ PAL ก่อน โดยพิจารณาดูที่เอาต์พุตของวงจร จะเห็นว่าเอาต์พุตสมมุติให้เป็น O_1-O_6 มีทั้งหมด 6 เอาต์พุต ส่วนมากเป็นแอคทีฟที่ Logic "1" และจำนวนอินพุตสมมุติให้เป็น I_1-I_{10} มีทั้งหมด 10 อินพุต เมื่อพิจารณาดูแล้ว เบอร์ที่น่าจะใช้คือ PAL 10H8 กับ 10L8 แต่ตามท้องตลาดไม่มีจึงต้องใช้เบอร์อื่นแทนคือ PAL 12L10 แต่เมื่อพิจารณาดูโครงสร้างของ PAL จะเห็นว่า Out Put เป็น NOR ถึง 10 ตัว จึงต้องพิจารณาเลือก Out Put ที่เหมาะสม

หลังจากที่เลือกเบอร์ที่ต้องการแล้วจึงนำมาทำการคำนวณความสัมพันธ์ของ In Put และ Out Put ด้วยทฤษฎีของดีมอร์แกน ดังนี้

$$/O_1 = I_1 * I_2$$

$$/O_2 = /I_1 + /I_2$$

$$/O_3 = /I_3 + /I_4 + /I_5 + /I_6 + /I_7$$

$$/O_4 = I_3 * I_4 * I_5 * I_6 * I_7$$

$$/O_5 = /I_8 + /I_9$$

$$/O_6 = /I_3 + /I_{10}$$

แล้วนำสมการที่ได้ไปเขียนเป็นโปรแกรม ไปทำการ Simulate โดยใช้ Software PALASM หาค่า Fuse map และนำค่านี้ไปเข้าเครื่อง Burn PAL แล้วนำ PAL ที่ได้มาใช้งานในโครงงานนี้ ส่วนอีก 4 วงจร ก็ทำในลักษณะเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

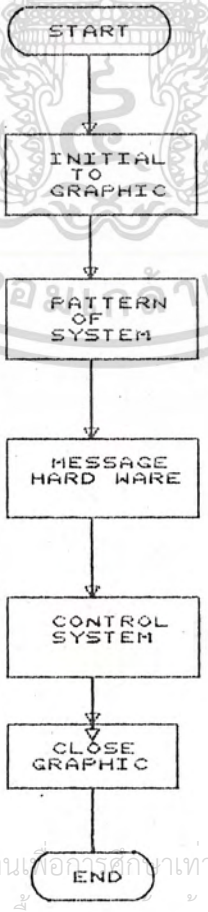
บทที่ 8

การเขียนโปรแกรมและการใช้งานโปรแกรม

การติดต่อการทำงานของทุกฟังค์ชั่น เราใช้เครื่องคอมพิวเตอร์ IBM PC เป็นตัวสั่งการ ดังนั้นจึงต้องเขียนโปรแกรมเพื่อไปควบคุมการทำงานของฟังค์ชั่น ซึ่งการเขียนโปรแกรมคอมพิวเตอร์ (Software) นี้จะใช้ภาษาซีในการเขียน การกำหนดฟังค์ชั่นต่างๆ สามารถอธิบายเป็น โฟลชาร์ต (Flowchart)

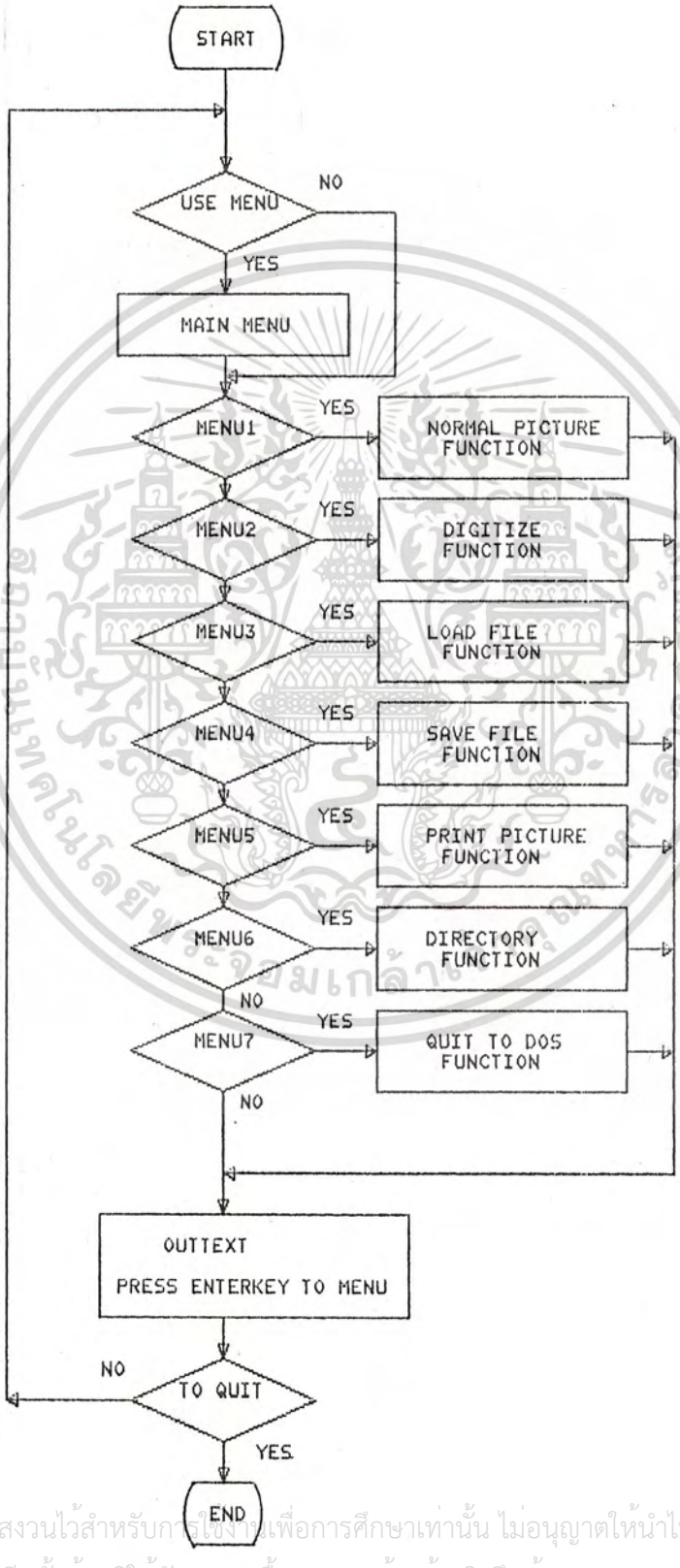
8.1 โฟลชาร์ตฟังค์ชั่นหลักของเมนู

- การเข้าสู่กราฟฟิกโหมด
- การเข้าสู่ระบบการทำงาน
- อธิบายข้อความส่วนของฮาร์ดแวร์
- การควบคุมการทำงานของระบบ
- การเลิกใช้กราฟฟิกโหมด



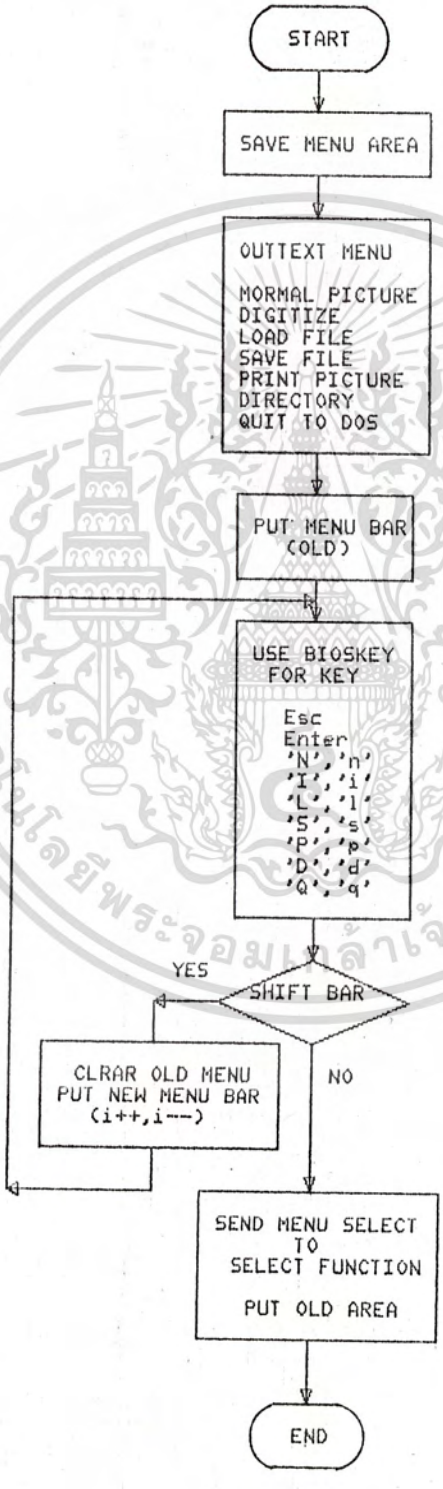
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 <u>โฟลชาร์ตการทำงานของระบบควบคุม</u> (Control System)



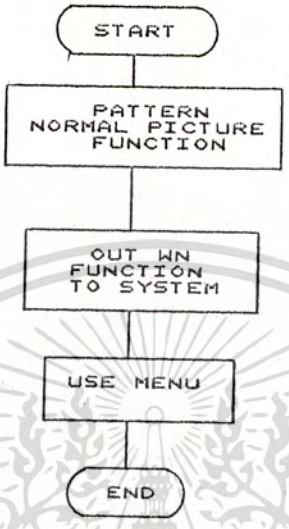
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 โฟลชาร์ตการแสดงเมนู (Mian Menu)

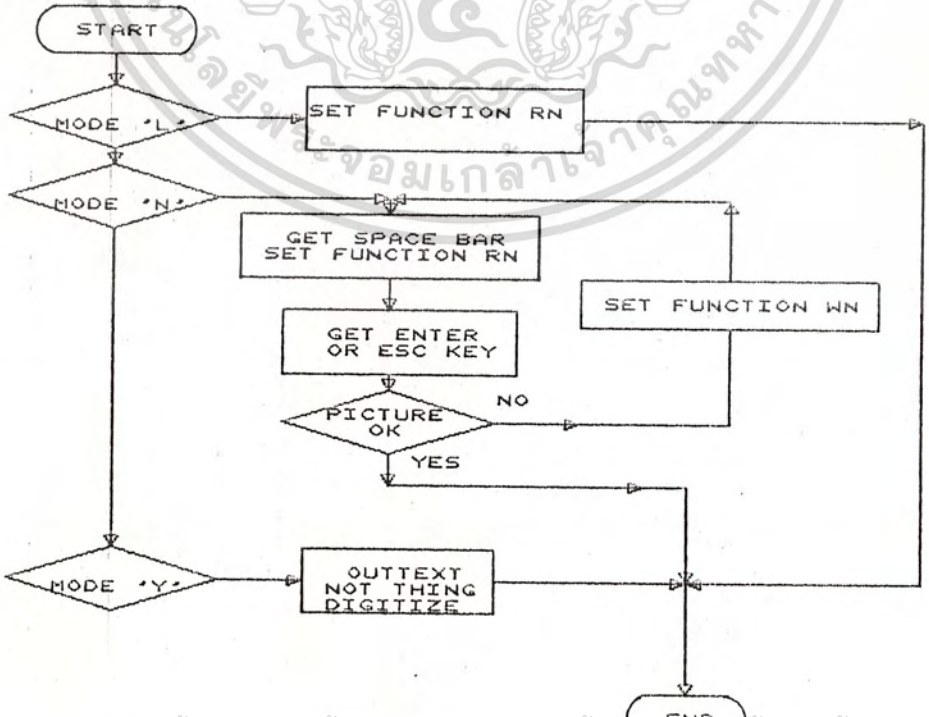


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.1 โฟลชาร์ตของฟังค์ชั่นการทำงานปกติในการแสดงภาพ (Normal Picture)

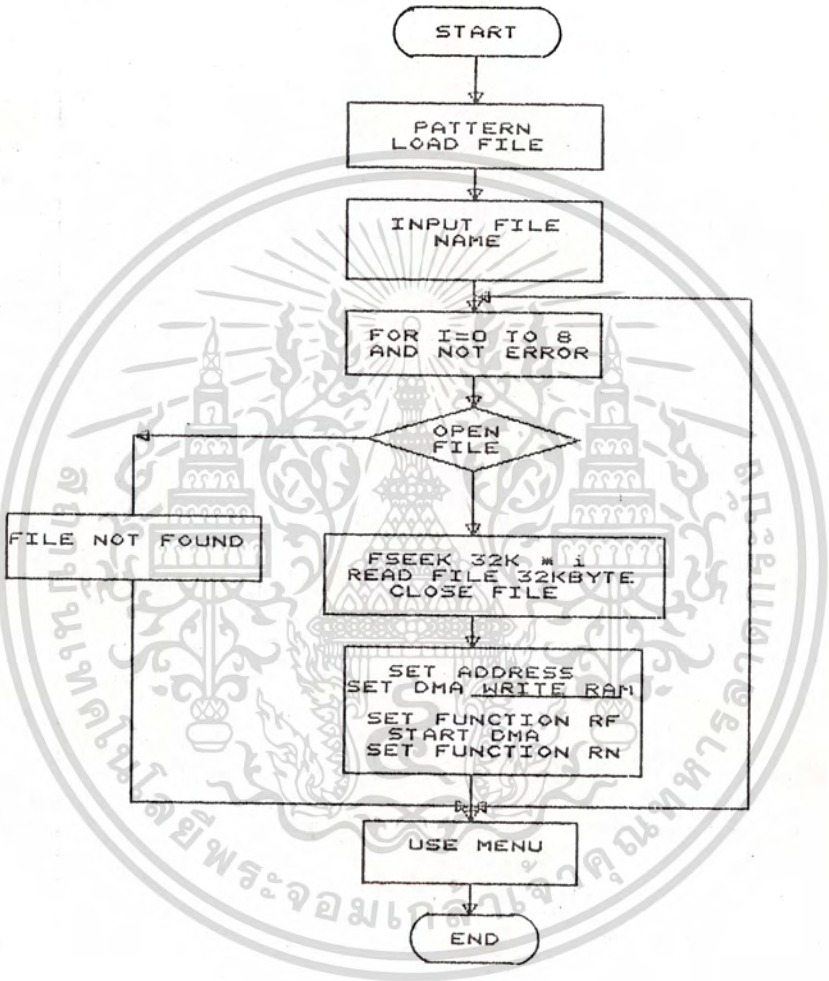


8.3.2 โฟลชาร์ตของการอ่านข้อมูลจากหน่วยความจำไปแสดงผล (Digitize)



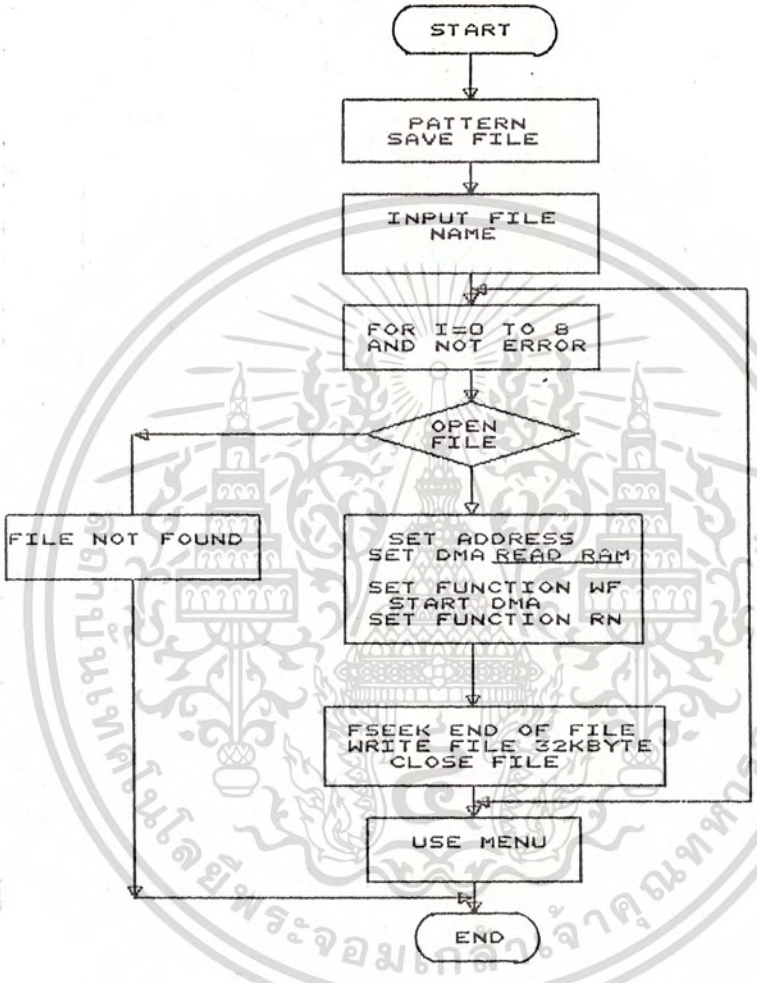
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.3 <u>โฟลชาร์ตของการอ่านข้อมูลจากแฟ้มข้อมูล</u> (Load File)



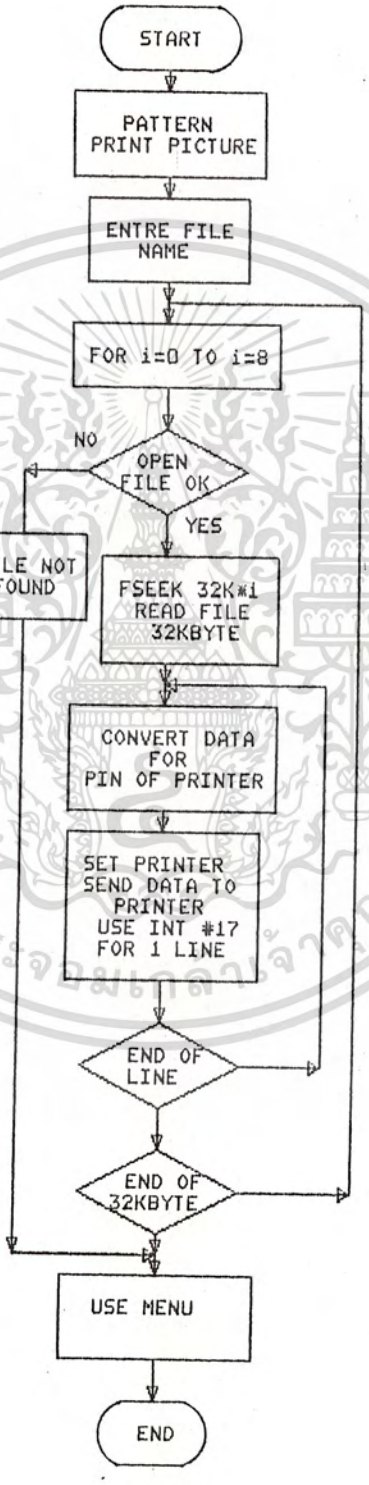
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.4 โฟลชาร์ตการนำข้อมูลไปเก็บในแฟ้มข้อมูล (Save File)



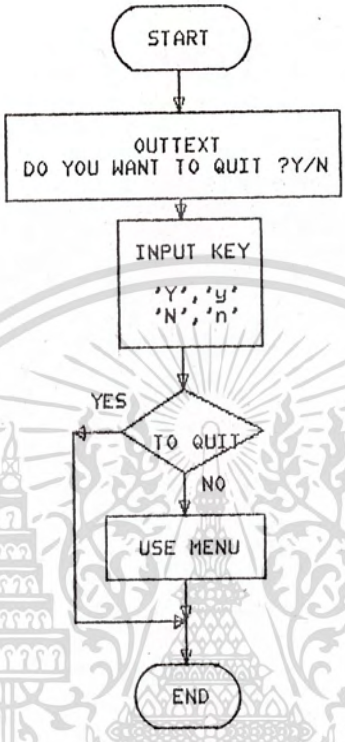
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.5 <u>โฟลชาร์ตการนำข้อมูลออกไปพิมพ์</u> (Print Picture)

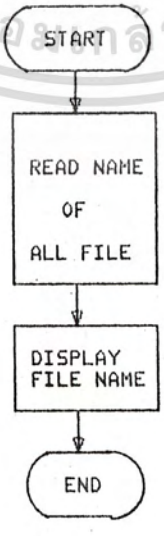


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3.6 โฟลชาร์ตการออกจากระบบไปสู่ DOS (Quit to Dos)



8.3.7 โฟลชาร์ตการอ่านชื่อแฟ้มข้อมูลจากแฟ้มข้อมูล (Directory)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรม

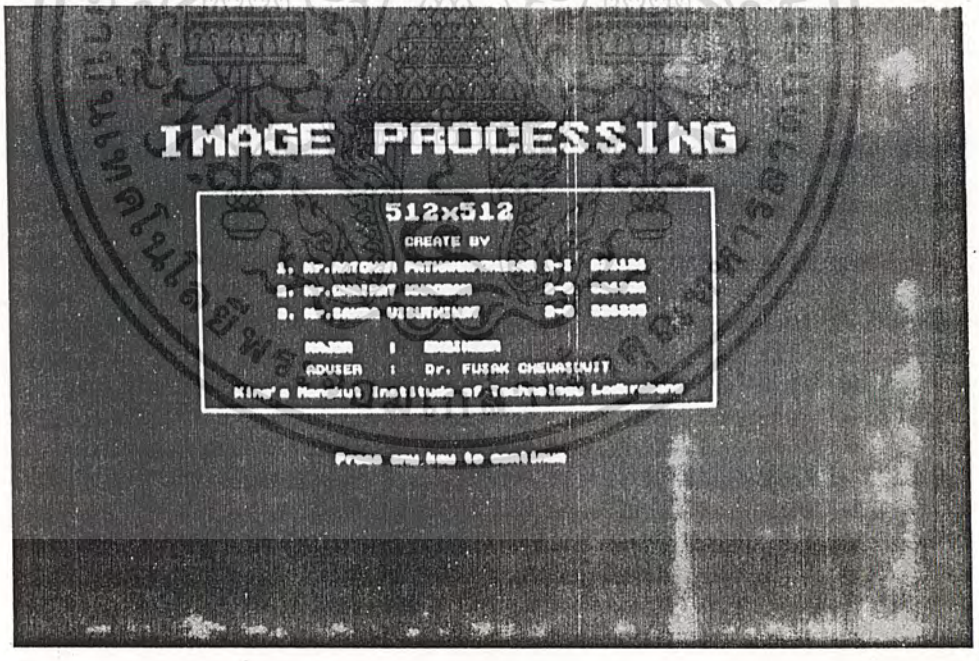
เมื่อทำการต่ออุปกรณ์ต่างๆเรียบร้อยแล้ว และต้องการติดต่อหรือให้อุปกรณ์ดังกล่าวทำงาน จะมีขั้นตอนการใช้โปรแกรมดังนี้

1. ทำการ Bost DOS โดยใช้แผ่น DOS Vertion อะไรก็ได้

2. หลังที่ courser กระพริบหลัง A Prome (A>>) ให้นำแผ่นดิสต์ที่มี File IMAGE512.EXE ใส่ใน Drive A แล้วพิมพ์

A>>IMAGE512 <ENTER>

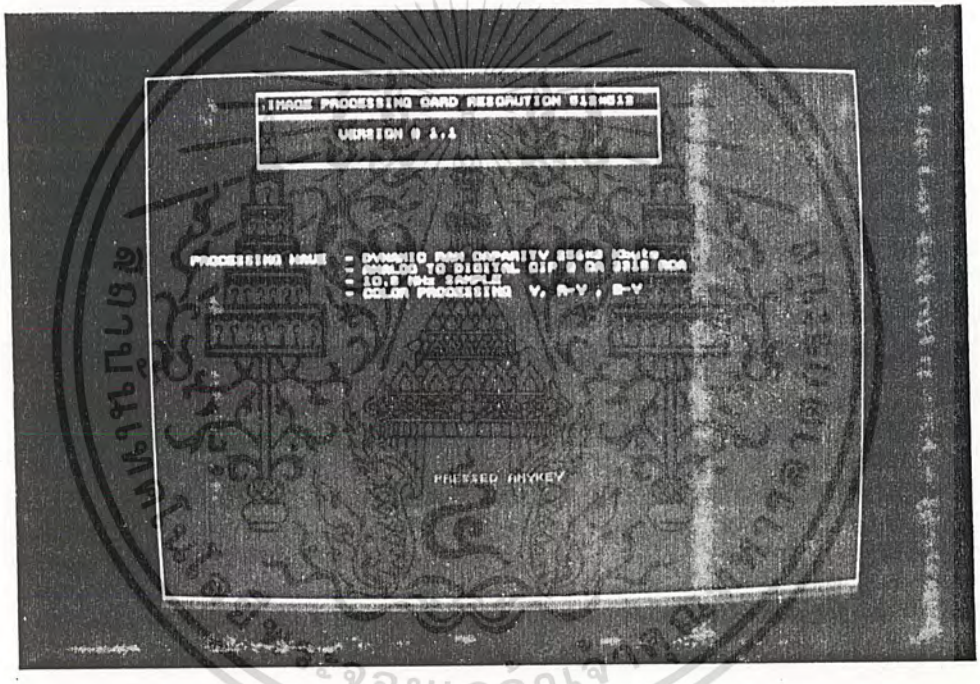
จะเห็นว่าเมื่อกด Enter แล้วจอภาพจะแสดงเป็น Pattern ให้ดูดังในรูปที่ 8.1



รูปที่ 8.1 ภาพแสดง Pattern บนจอมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

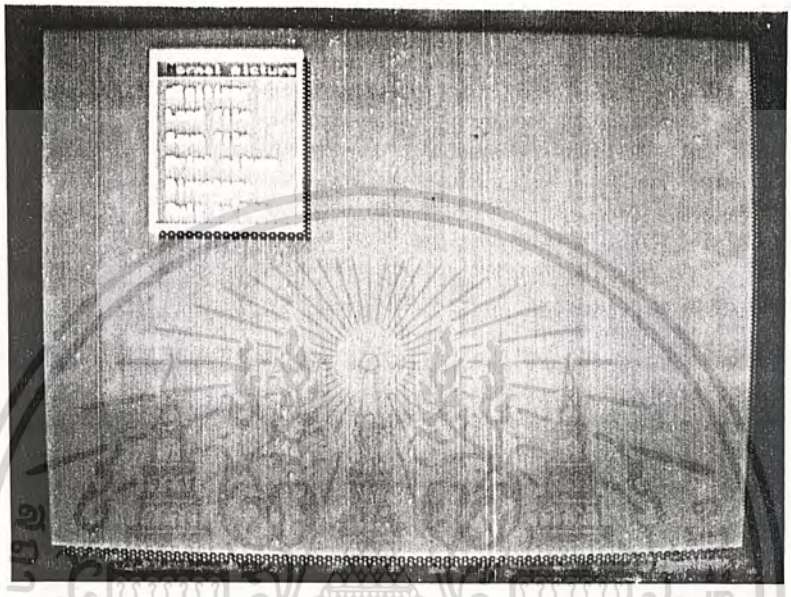
4. ที่หน้าจอจะแสดงอักษรคำ "Press any key" ซึ่งหมายความว่าเรา สามารถกด Key board อะไรก็ได้ จอมอนิเตอร์จะแสดงรายละเอียด(Message) ของฮาร์ดแวร์(Hardware) ดังแสดงในรูปที่ 8.2



รูปที่ 8.2 ภาพแสดงข้อความ รายละเอียดของHardware

5. หลังจากนั้นให้กด Key board คีย์ใดคีย์หนึ่งเพื่อลบหน้าจอแล้วจะขึ้นMenu ซึ่ง Menu การทำงานจะเห็นได้จาก รูปที่ 8.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.3 ภาพแสดง Menu การทำงาน

การเลือก Menu สามารถเลือกได้โดยการกดตัวอักษรตัวใหญ่ใน Menu หรือจะใช้วิธีการเลือก Menu Bar แล้วกด Enter เพื่อเลือก Menu ก็ได้

กรณีต้องการยกเลิก Menu ก็สามารถกด ESC ได้เลย

6. การเลือก Menu

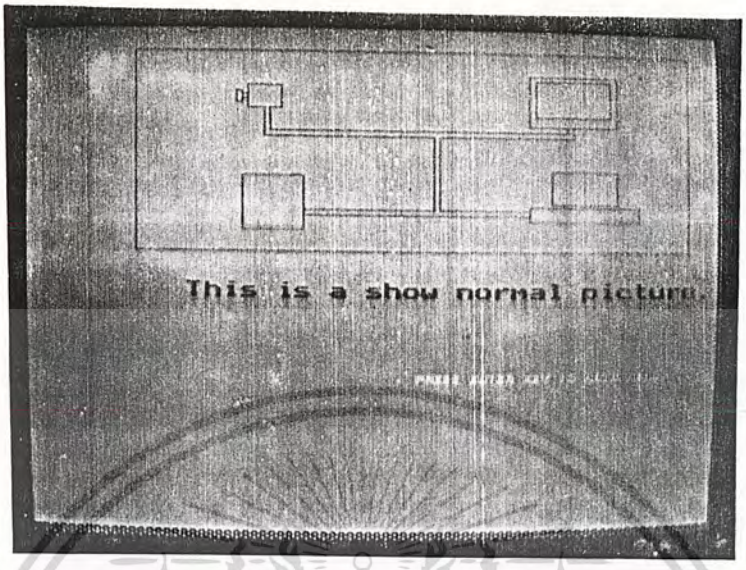
6.1 ถ้าเลือก Menu ที่ 1 Normal Picture ก็จะขึ้นหน้าจอภาพดังแสดงในรูปที่ 8.4

สำหรับฟังค์ชั่นการทำงานแบบ Normal Picture นี้จะไปสั่งให้ Hardware ทำการเปลี่ยนฟังค์ชั่นให้เป็นการแสดงภาพจากกล้องวิดีโอ โดยตรง

การกลับไปที่ Menu ก็จะทำได้โดยการกด Enter

การยกเลิก Menu ทำได้โดยการกด ESC

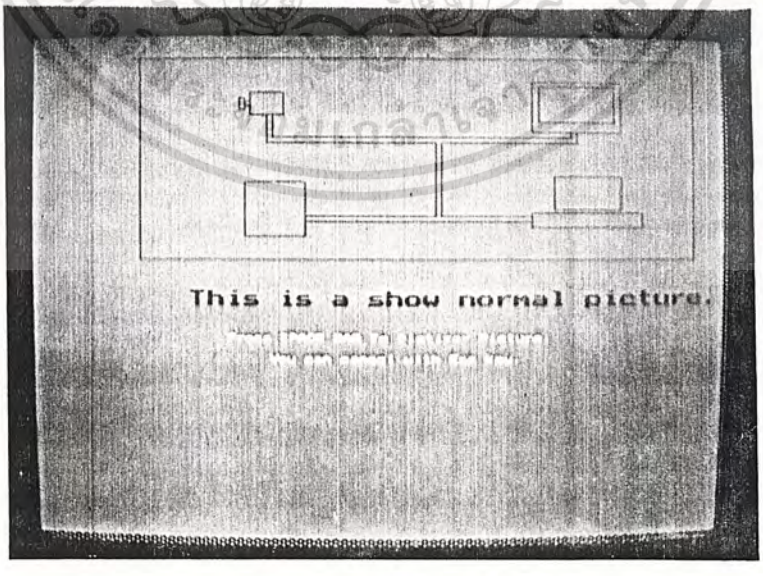
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.4 ภาพแสดงการทำงานแบบ Normal Picture

6.2 ถ้าเลือก Menu ที่ 2 Digitize ก็จะขึ้นหน้าจอมอนิเตอร์ ซึ่งการ Digitize มีอยู่ 3 กรณี คือ

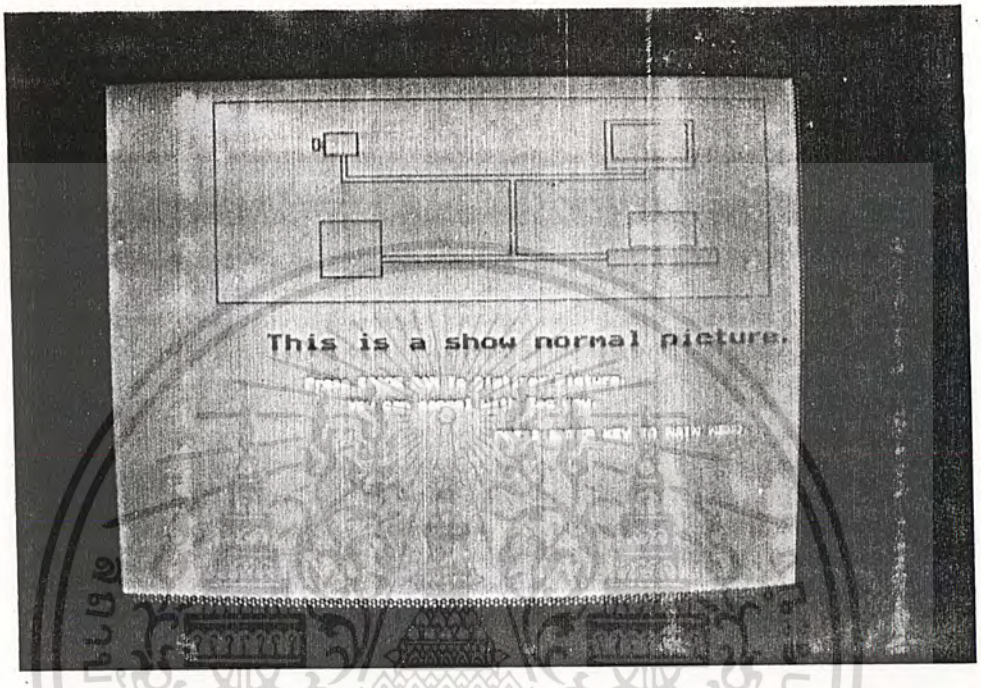
กรณีที่ 1 ถ้าไม่มีการใช้ Menu อื่นมาก่อนก็จะขึ้นข้อความดังแสดงใน รูปที่ 8.5



รูปที่ 8.5 การแสดงการ Digitize ในกรณีที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

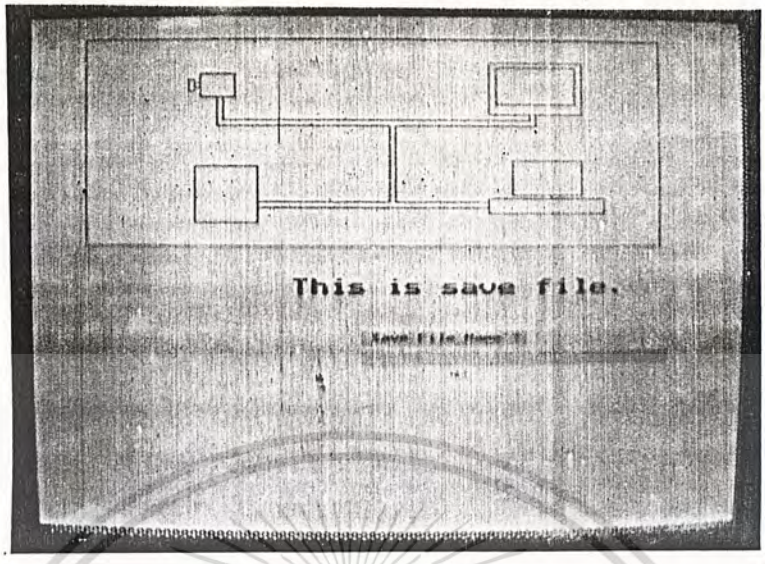
กรณีที่ 2 การเลือก Digitize โดยผ่าน Menu Normal Picture ซึ่งจะสามารถเข้ามา Digitize ภาพนั้นได้โดยจะขึ้นข้อความดังแสดงในรูปที่ 8.6



รูปที่ 8.6 การแสดงการ Digitize ในกรณีที่ 2

จากรูปที่ 8.6 ข้อความที่ว่าให้รอการกด Space Bar เมื่อทำการกด Space Bar แล้วก็จะต้องตัดสินใจอีกว่าจะยกเลิกภาพหรือไม่ยกเลิก เพื่อที่จะได้ทำการเก็บภาพใหม่ และถ้าทำการกด Enter ภาพที่ได้ก็เป็นภาพที่ Digitize ไว้ ภาพนี้จะสามารถนำไปเก็บลงใน File ได้โดยการผ่านขึ้นไปที่ Menu โดยการ Enter Key จอภาพก็จะแสดง Menu หลังจากนั้นถ้าต้องการเก็บภาพก็เลื่อน Menu ไปที่ Save File แล้วกด Enter จอภาพก็จะแสดง Pattern ของการ Save ภาพ แล้วโปรแกรมก็จรอชื่อ File ซึ่งข้อความบนหน้าจอมอนิเตอร์จะปรากฏดังรูปที่ 8.7

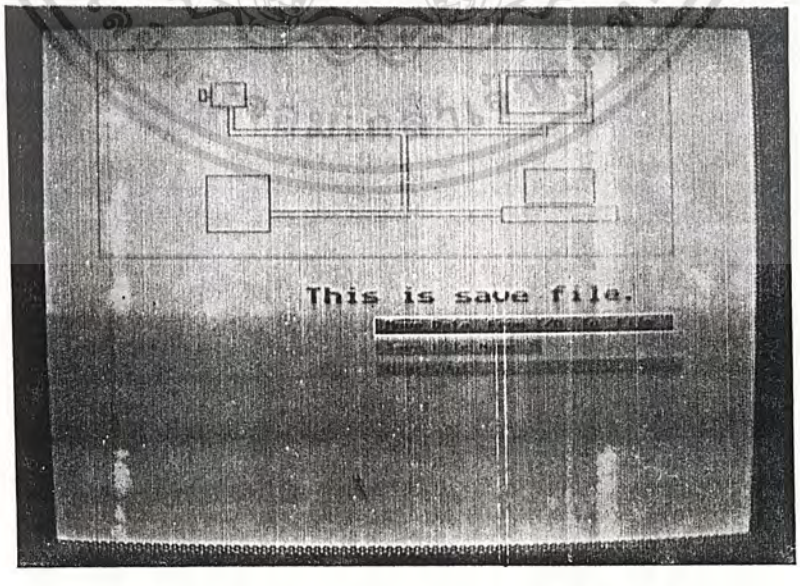
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.7 ภาพแสดงการ Saving

หลังจากป้อนชื่อ File แล้วกด Enter โปรแกรมก็จะนำข้อมูลจาก Memory ไปเก็บใน File ซึ่งมีขนาด 256 Kbytes หลังจากการเก็บภาพแล้ว โปรแกรมก็จะเข้าสู่โหมด Digitize ในกรณีที่ 3 ทันที

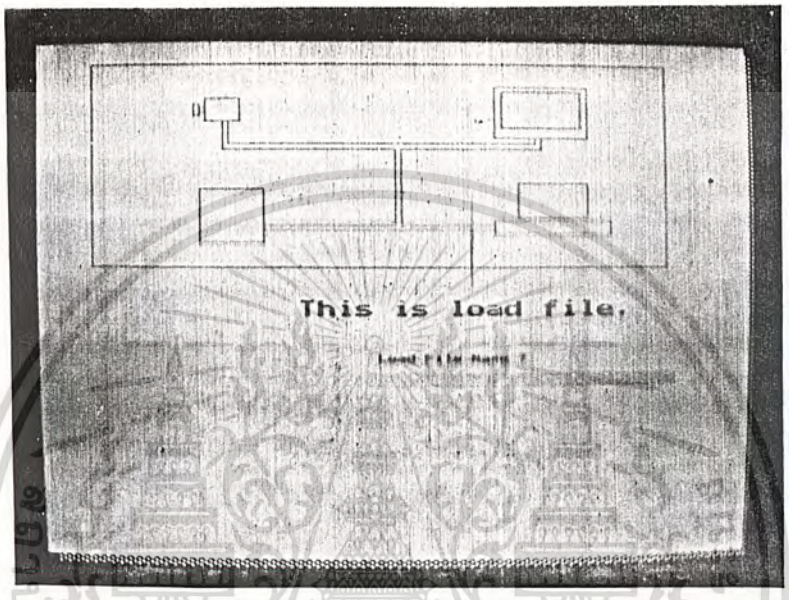
กรณีที่ 3 หลังจาก Load หรือ Save ข้อมูลเรียบร้อยแล้ว โปรแกรมจะเข้าสู่การ Digitize โดยตรง ซึ่งข้อความแสดงดังรูปที่ 8.8



รูปที่ 8.8 ภาพแสดงการ Digitize ในกรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 ถ้าเลือก Menu ที่ 3 Load File ก็จะปรากฏหน้าจอมอนิเตอร์ โดยมีข้อความดังแสดงในรูปที่ 8.9



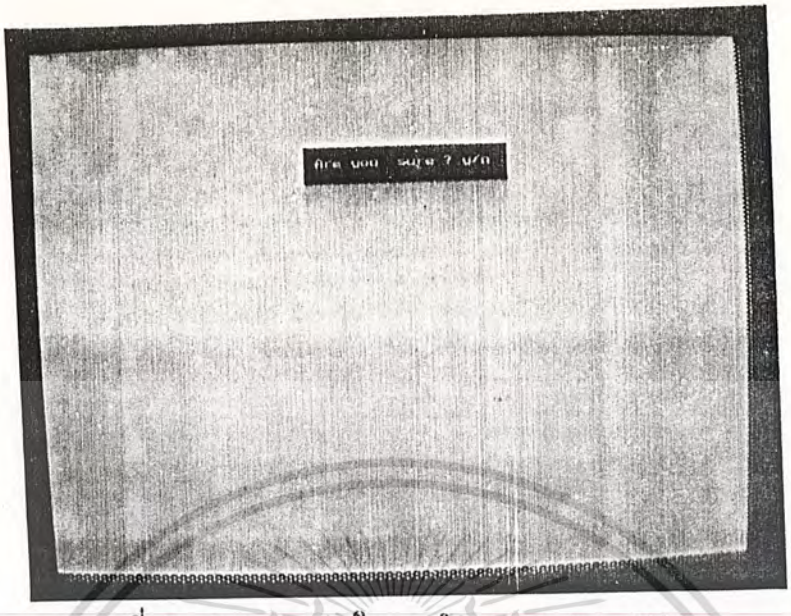
รูปที่ 8.9 ภาพแสดงการมี Load File

จากรูปที่ 8.9 เป็น Pattern ของ Load File Function โดยโปรแกรมจะรอชื่อ File ที่จะทำการ Load เมื่อใส่ชื่อ File และกด Enter Key โปรแกรมก็จะทำการ Load ข้อมูลจาก File

เมื่อ Load ภาพ เสร็จเรียบร้อย ก็จะเข้าสู่ Digitize Function ที่แท้

6.4 ถ้าเลือก Menu 5 Printing ก็จะปรากฏที่หน้าจอมอนิเตอร์ ซึ่ง Pattern แสดงในรูปที่ 8.10 โปรแกรมจะรอรับชื่อ File ที่จะทำการ Print

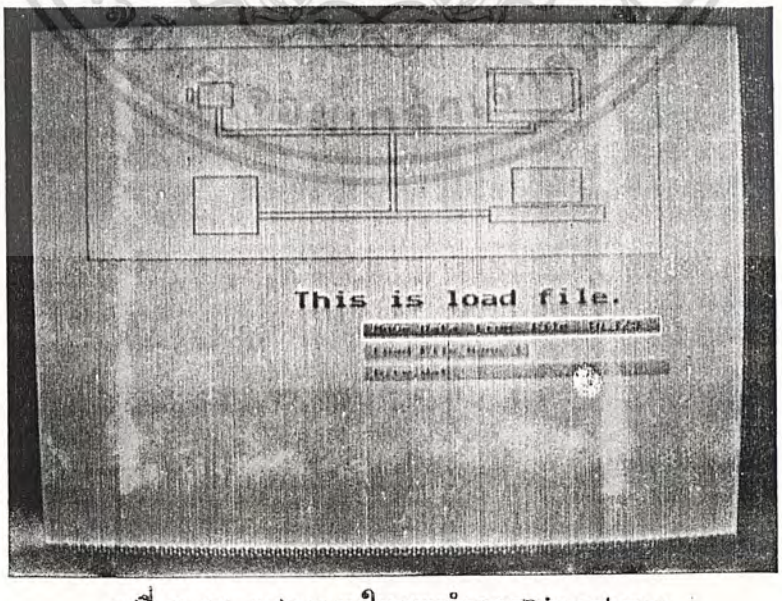
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.10 แสดงภาพในขณะทำการ Print File

หลังจากใส่ชื่อ File ที่ต้องการ Print แล้วกด Enter โปรแกรมก็จะจัดการส่งค่าข้อมูลออกมา Print

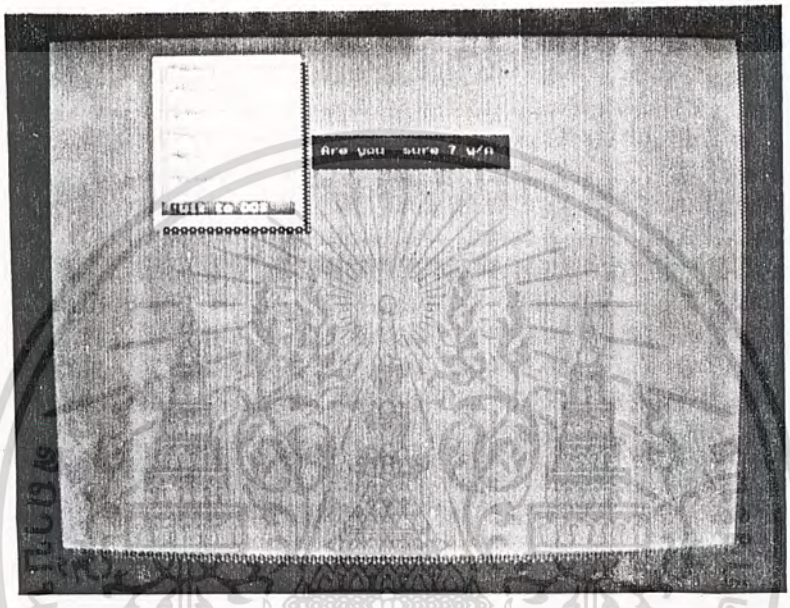
6.5 ถ้าเลือก Menu 6 Directory ก็จะปรากฏที่หน้าจอมอนิเตอร์ ซึ่ง Pattern แสดงในรูปที่ 8.11 โปรแกรมจะถามว่าต้องการ Load ชื่อ File ใน Drive ไหน และเมื่อกด Enter Key โปรแกรมก็จะอ่าน ชื่อ File ทั้งหมดขึ้นมาให้ดูบนจอมอนิเตอร์



รูปที่ 8.11 แสดงภาพในขณะทำการDirectory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 ถ้าเลือก Menu 7 Quit to Dos จะปรากฏหน้าจอมอนิเตอร์ ซึ่ง Pattern แสดงในรูปที่ 8.12 ซึ่งโปรแกรมจะถามความแน่นอนของเราอีกทีหนึ่ง โดยการให้เลือกตอบใช่ (Yes) หรือไม่ใช่ (No) ถ้าเกิดตอบใช่(Yes) โปรแกรมก็จะออกไปสู่ Dos



รูปที่ 8.12 แสดงภาพในขณะทำการ Quit to Dos

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

บทวิจารณ์และสรุปผล

ในการสร้าง Image interface card with resolution 512*512 pixels นั้นผู้สร้าง ได้แบ่งขั้นตอนในการสร้างดังต่อไปนี้

- การหาข้อมูลเกี่ยวกับหลักการขั้นพื้นฐาน และข้อมูลจากปริญญานิพนธ์และวิทยานิพนธ์เก่าที่มีผู้ทำไว้แล้วในอดีต พร้อมทั้งทฤษฎีของระบบสัญญาณวิดีโอทั้งภาพขาวดำและภาพสี

- ขั้นตอนการเลือกอุปกรณ์ โดยการพิจารณาเลือกอุปกรณ์ที่ใช้นั้นต้องดูความเหมาะสมในการใช้งาน ราคา และความมีประสิทธิภาพในการใช้งานด้วย

- ขั้นตอนการออกแบบ โดยในขั้นแรกจะใช้อุปกรณ์พื้นฐานก่อนเพื่อง่ายต่อความเข้าใจและไม่สับสน ในขั้นนี้เมื่อออกแบบเสร็จจะได้วงจรที่มีขนาดใหญ่มาก ซึ่งไม่เหมาะสมกับพื้นที่ในการติดตั้ง และยุ่งยากในการเชื่อมโยงสายต่อ

- ขั้นตอนการทดลอง เพื่อให้ผลของการออกแบบเป็นจริงดังนั้นจึงนำเอาวงจรที่ได้มาจากการออกแบบมาทำการทดลอง

- ขั้นตอนการลดรูปวงจร ในขั้นตอนนี้จะทำการปรับปรุงวงจรที่สร้างขึ้นให้มีขนาดเล็กลงเพื่อให้เหมาะสมกับพื้นที่ในการติดตั้ง และเหมาะสมที่สุดในการออกแบบและการสร้างออกมาเป็นชิ้นงาน ซึ่งรวมถึงการใช้ PAL เพื่อทำการยุบวงจรในเล็กที่สุด

- ขั้นตอนการออกแบบลายวงจร ขั้นตอนนี้จะใช้ Software Or-CAD.SDT และ Or-CAD.PCB เพื่อช่วยในการออกแบบลายวงจร เมื่อได้ลายวงจรแล้วก็นำไปถ่ายฟิมล์ และทำปริ้นต์

- ขั้นตอนการตรวจสอบลายวงจรในแผ่นปริ้นต์ที่ทำขึ้นมา โดยจะตรวจสอบดูทุกจุดบนแผ่นปริ้นต์ เพื่อหาจุดขาดหรือจุดที่ Short กัน และลายทองแดงที่ไม่ถูกต้องตามวงจร

- ขั้นตอนการลงอุปกรณ์ ซึ่งจะต้องลงอุปกรณ์ตามตำแหน่งต่างๆที่ออกแบบไว้ จากนั้นก็นำมาตรวจเช็คหาจุดขาด หรือ จุด Short อีกทีหนึ่ง

- ขั้นตอนการทดสอบและทดลอง เพื่อให้ได้ค่าที่ถูกต้อง

- ขั้นตอนการเขียน Software เพื่อใช้ในการควบคุมการทำงานของฟังค์ชั่น

- ขั้นตอนการนำ Hardware และ Software มาใช้ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผล

ในระบบการทำงานของ Image interface card นั้น เนื่องจากวงจรที่สร้างขึ้นมาเป็นวงจรที่มีขนาดใหญ่ ดังนั้นจึงต้องประสบกับปัญหาต่างๆมากมายในการทดลอง ไม่ว่าจะเป็นการทดลองในโฟโตบอร์ดหรือกระทั่งบนแผ่นเปริ้นต์ ซึ่งปัญหานี้อาจจะเกิดจากการต่อสายยาวเกินไป กระแสไหลลงกราวด์ไม่ดี เป็นต้น แนวทางการแก้ปัญหาก็โดยการจัดแยกระบบออกเป็นส่วนย่อยๆ ซึ่งแต่ละส่วนย่อยจะเป็นง่ายต่อการตรวจสอบ การทดสอบจะเป็นการทดสอบค่าของอินพุทกับเอาท์พุท สมมติว่าทำการตรวจสอบแล้วค่าอินพุทถูกต้อง แต่ค่าเอาท์พุทไม่ถูกต้องก็แสดงว่าภายในของวงจรย่อยนี้ไม่ดีหรือไม่ถูกต้อง ซึ่งตรวจเช็คและแก้ไขได้ง่าย สำหรับค่า Noise ในสายไฟอาจจะมีบางแต่ก็ไม่มากเหมือนกับการตรวจสอบทั้งระบบใหญ่ เมื่อวงจรในแต่ละส่วนย่อยทำงานแล้วเราจึงค่อยนำมาต่อรวมเป็นระบบใหญ่ ซึ่งถ้าวงจรยังไม่ทำงานก็สามารถรู้สาเหตุได้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

(ACKNOWLEDGMENT)

ปริญญานิพนธ์ชุดนี้จะสำเร็จลงไม่ได้ถ้าขาดอาจารย์ รศ.ดร.ฐศักดิ์ ช่วงสุวิทย์ อาจารย์เกษตร ศิริสีนติสัมฤทธิ์ และอาจารย์วิทยา ทิพย์สุวรรณพร ผู้ซึ่งประสิทธิ์ประสาทวิชาความรู้ ตลอดจนให้คำปรึกษา แนะนำแนวทางและวิธีการแก้ไขตลอดมา

ขอขอบพระคุณอาจารย์พิษณุ เฉลิมวัฒน์ และคุณสุชาติ ฉิมประดิษฐ์ ที่ให้การสนับสนุนและให้ข้อคิดในด้านต่างๆตลอดมา

และขอขอบพระคุณอาจารย์ทุกท่านในภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม เพื่อนๆน้องๆทุกคนที่ให้ความช่วยเหลือในสิ่งต่างๆและกำลังใจ

คณะผู้จัดทำ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ปริญญานิพนธ์ เรื่องแผงวงจรควบคุมภาพ(IMAGE INTERFACE CARD), สุพรชัย โชติพุทธิกุล, สุพัฒน์ เอี่ยมวิวัฒน์, อนุวัฒน์ พฤษ์พิจารณ์, ปีการศึกษา 2331

2. ปริญญานิพนธ์ เรื่องแปลงสัญญาณภาพสีเป็นดิจิตอล, วชิรพันธ์ ภัทรมงคลฤทธิ์, เฉลิมพล ปุญญาภิรมณ์, เกรียงไกร จันทรา, ปีการศึกษา 2532

3. ปริญญาณนิพนธ์ เรื่องการแปลงสัญญาณเอนกประสงค์, ชัยโชค ภัทรโกวิท, ณรงค์ จันทร์ฉาย และเพื่อนๆ, ปีการศึกษา 2531

4. ดร.ธวัช เมฆสวรรค์, เทคนิคการซ่อมเครื่องรับโทรทัศน์, สมาคมส่งเสริมความรู้ระหว่างประเทศ, พ.ศ. 2531

5. ประกิจ ตังตีสานนท์, หลักฐานโทรทัศน์, คณะวิศวกรรมศาสตร์ สจล., พ.ศ. 2529

6. ทินกร ดุ๊ก, การอินเตอร์เฟส IBM PC, ฟิสิกส์เซนต์เตอร์ การพิมพ์

7. TTL Dalabook, Texas Instrument Incorporated

8. Technical Referent, IBM

9. Brian W. Kernighan and Dennis M.Ritchie, "The C Programming Language", Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632

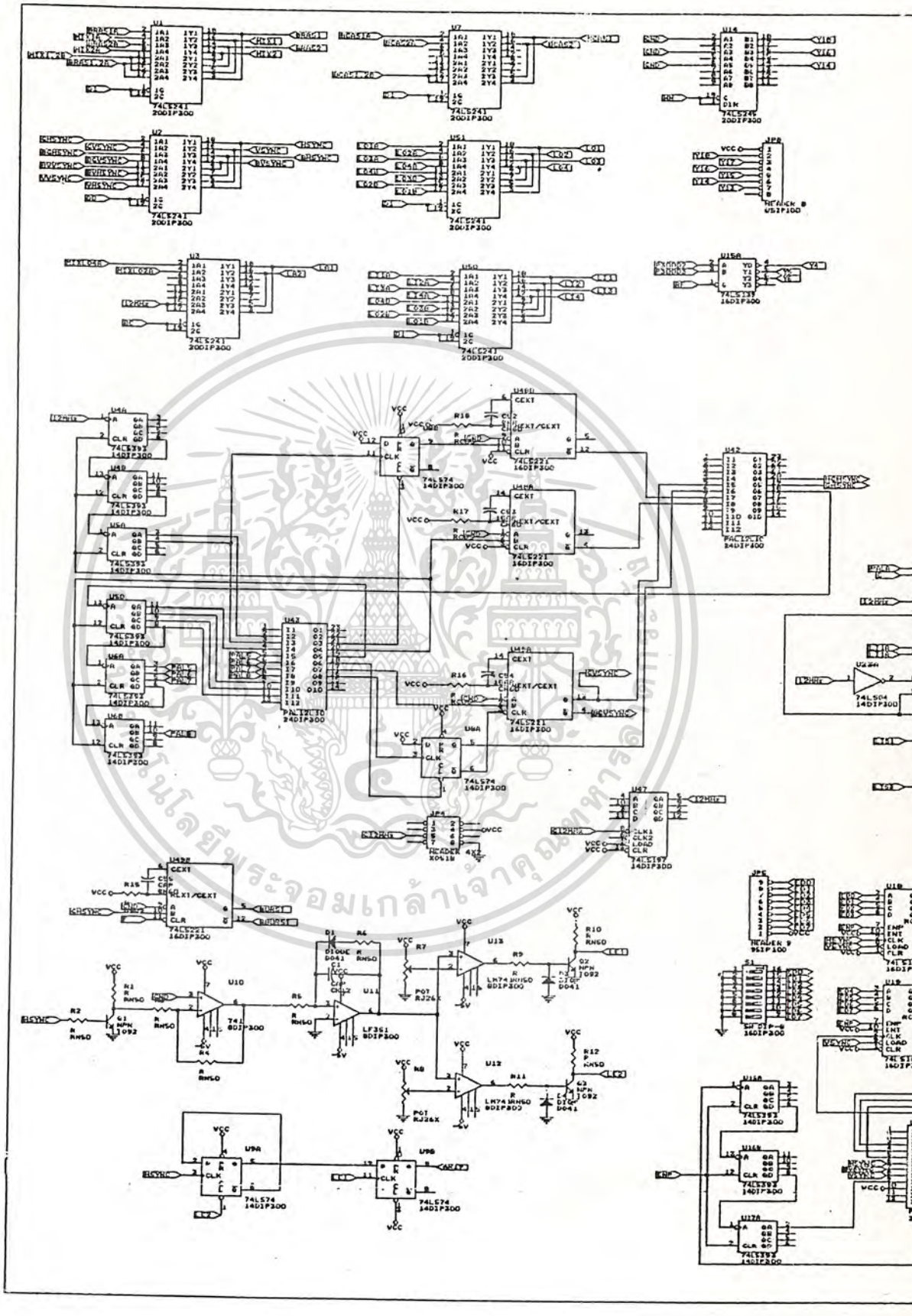
10. "Turbo C", Borland International, Inc., 4585 Scotts Valley Drive Scotts Valley, California 95066

11. Rafael C.Gonzalez and Paul Wintz, "Digital Image Processing", Addison-Wesley Publishig Company, Inc.

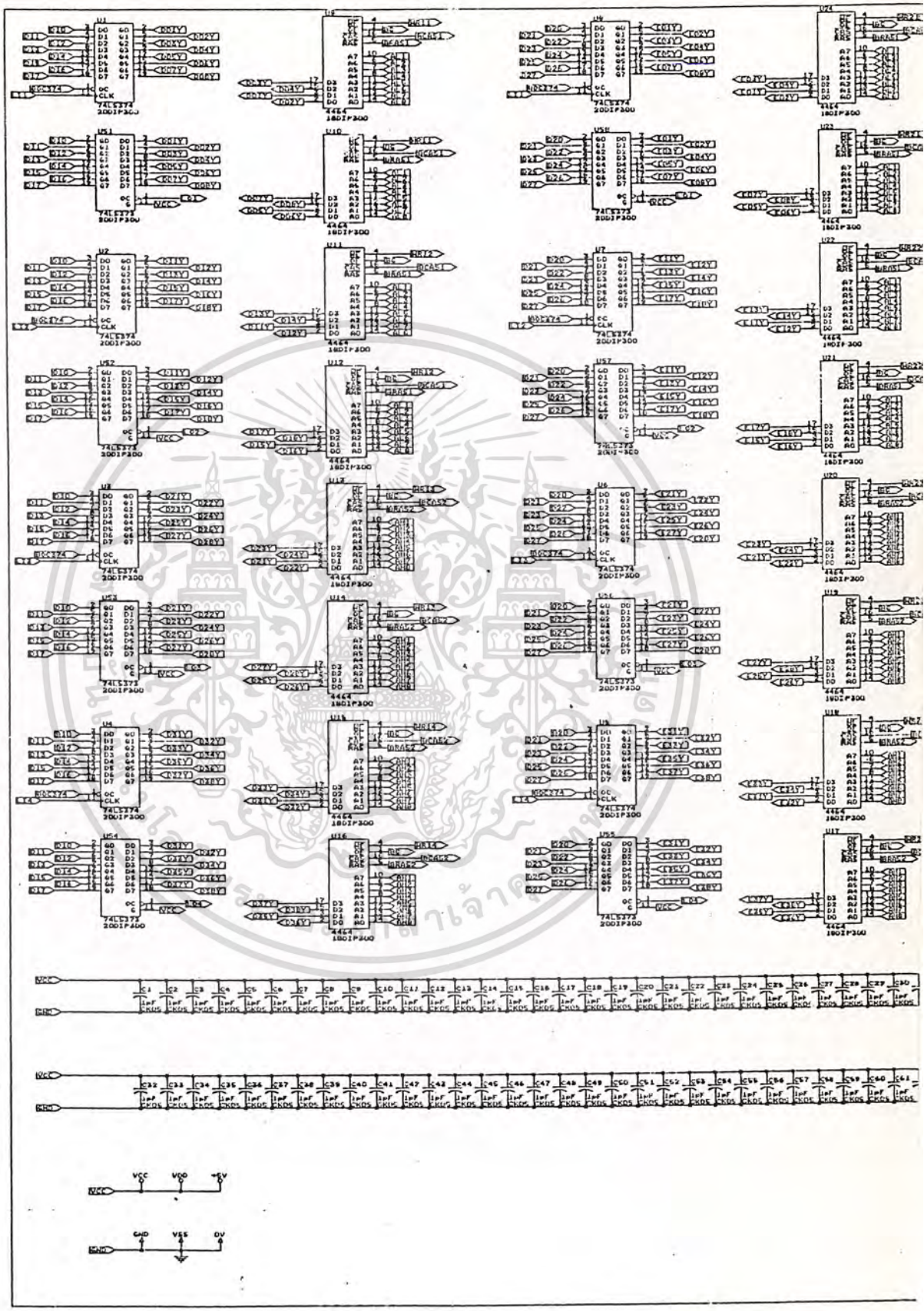
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

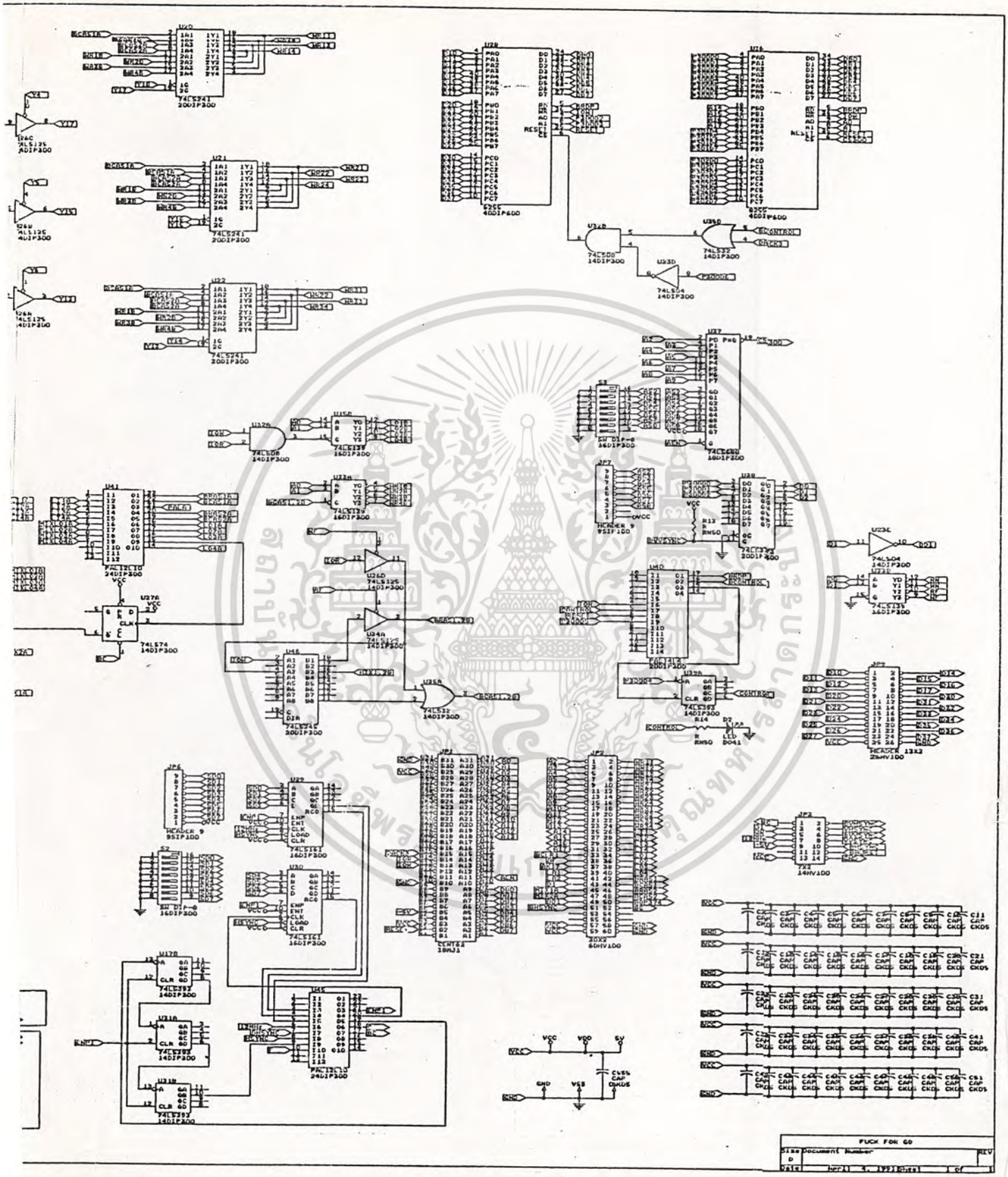
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



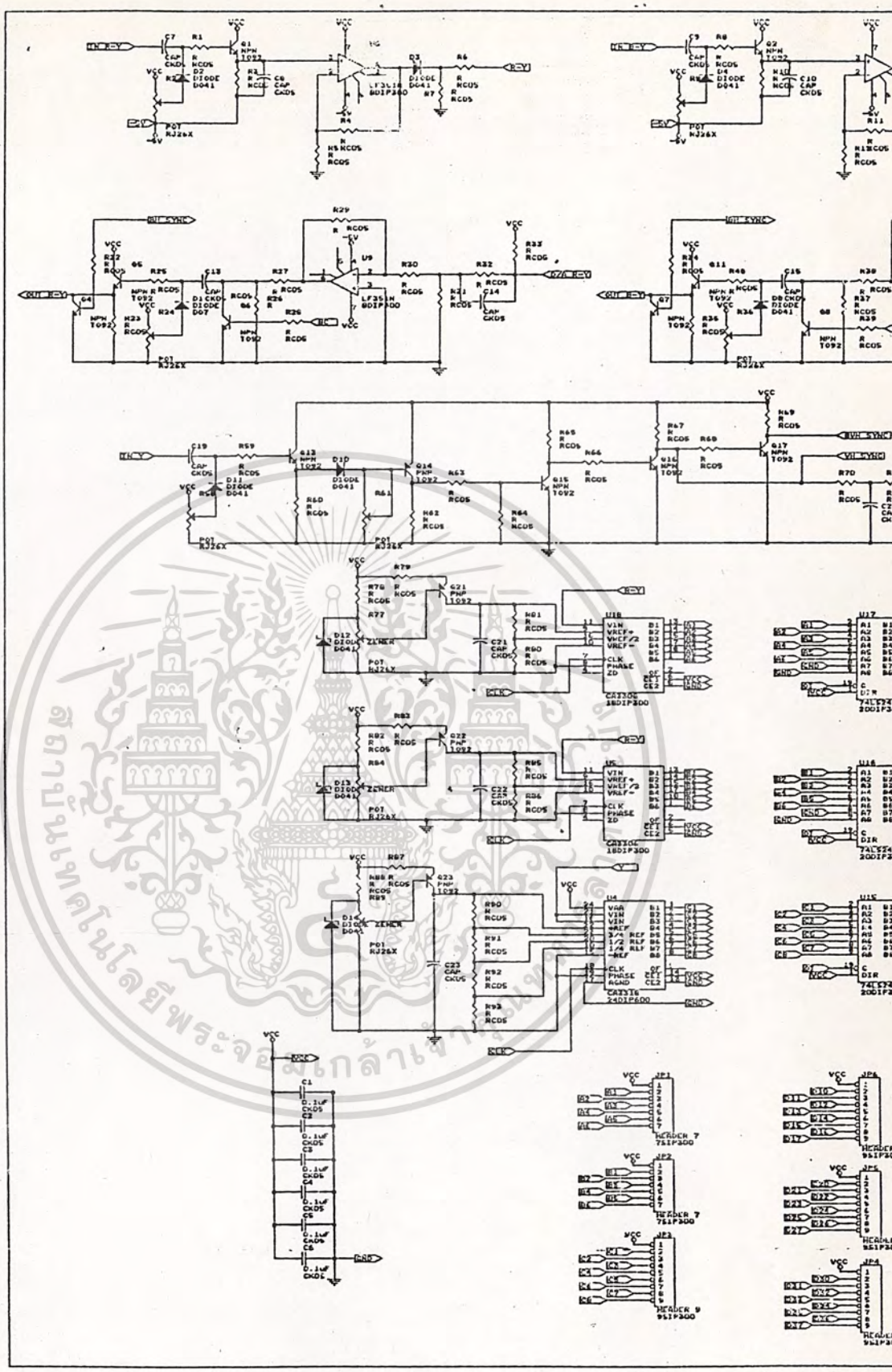
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



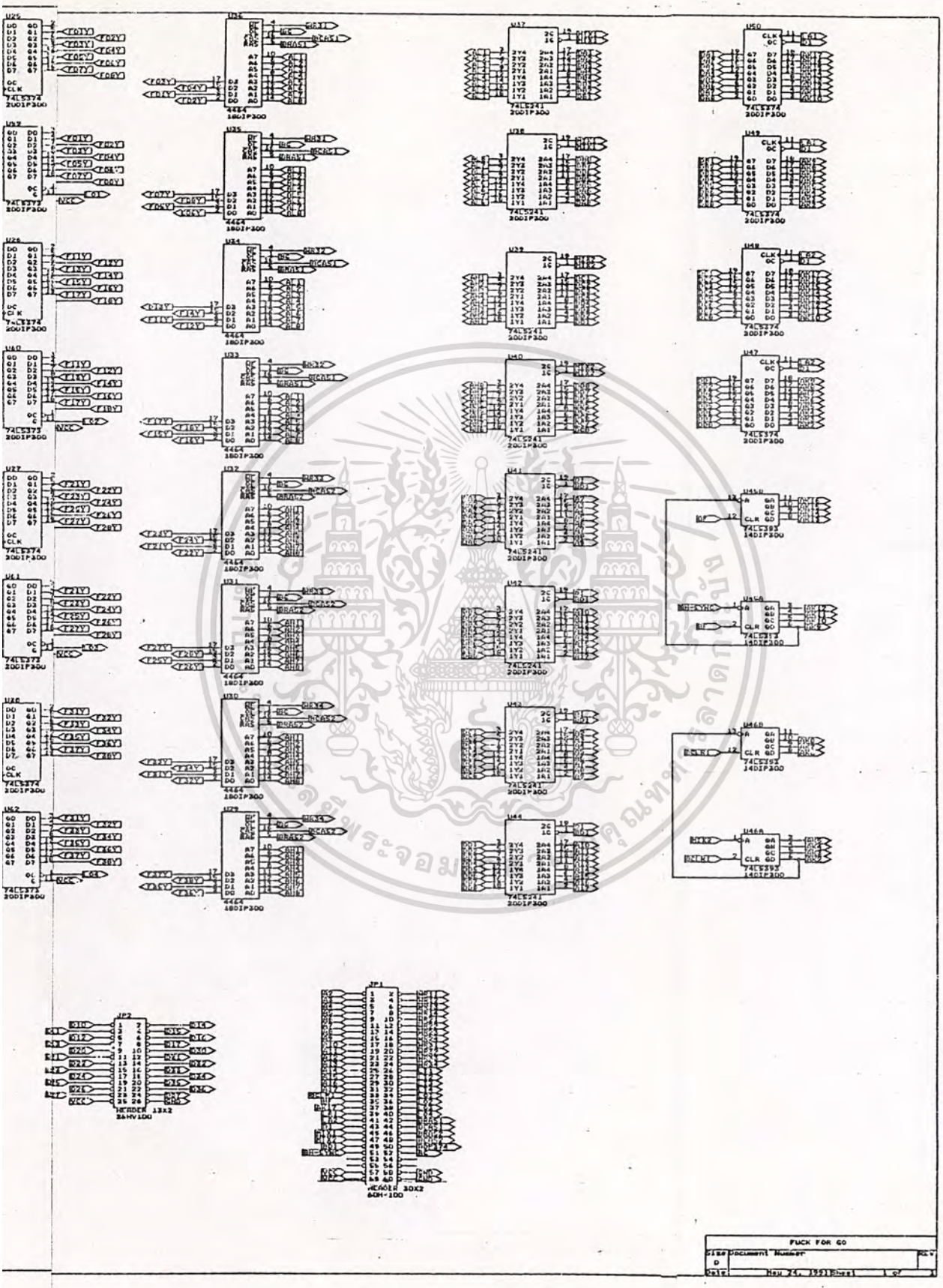
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



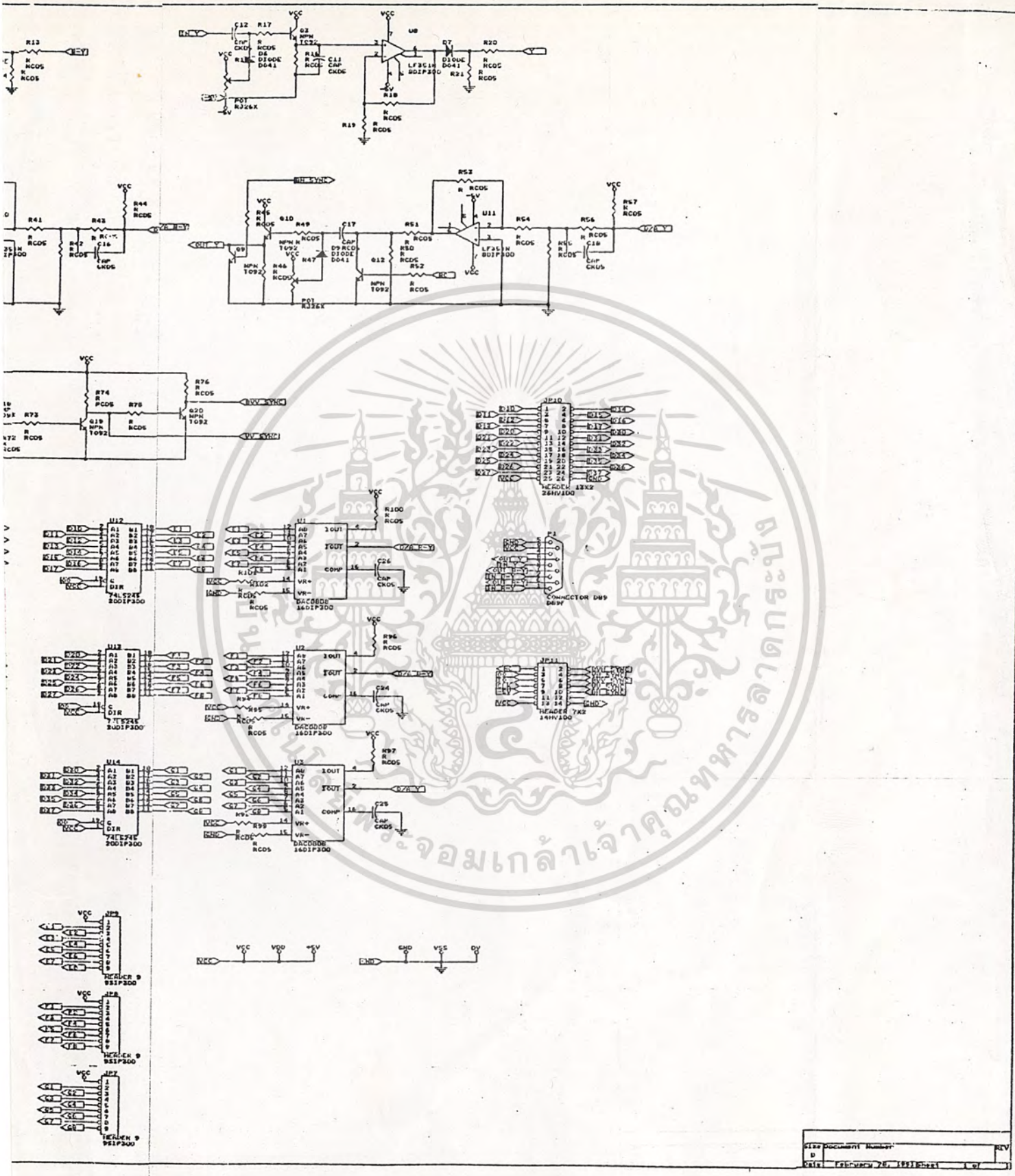
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



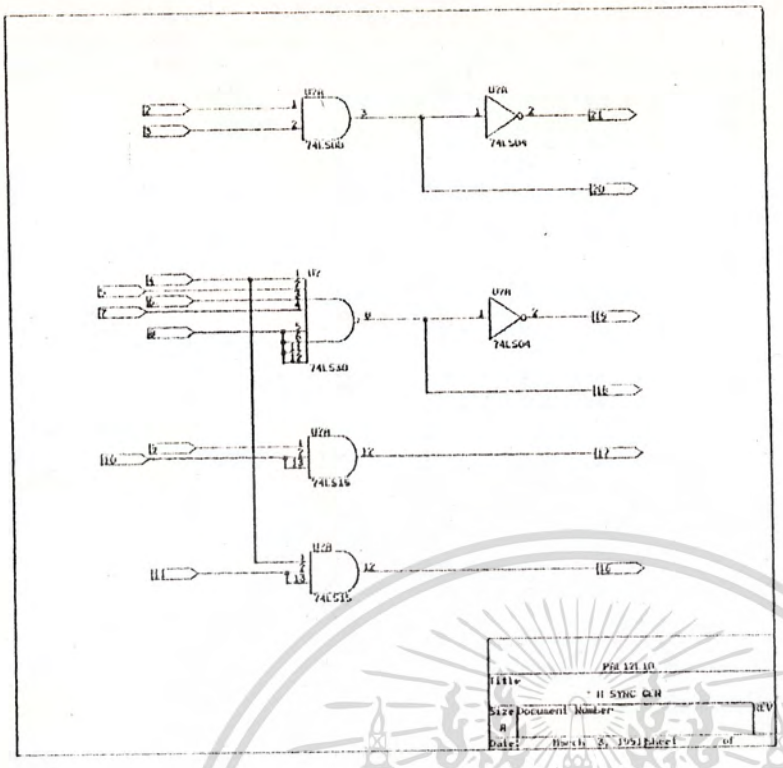
FUCK FOR GO

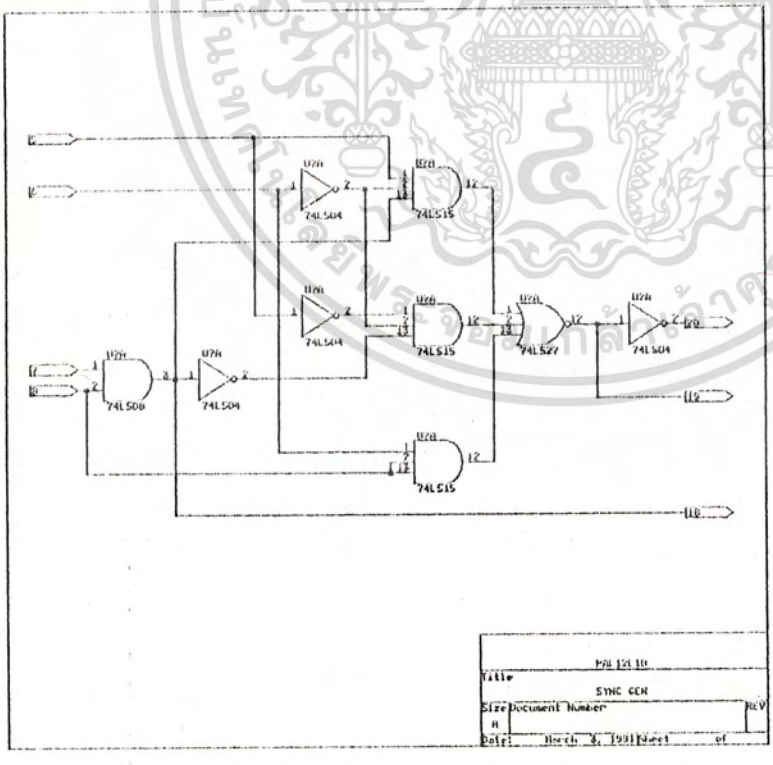
Size: D | Document Number: | Rev:

Date: Nov 24, 1991 | Sheet: | of:

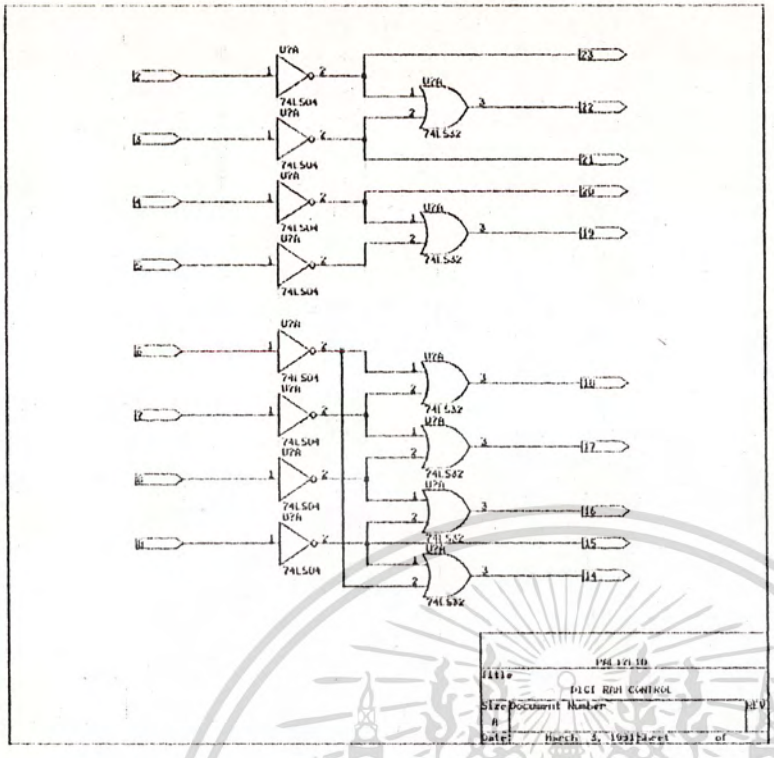


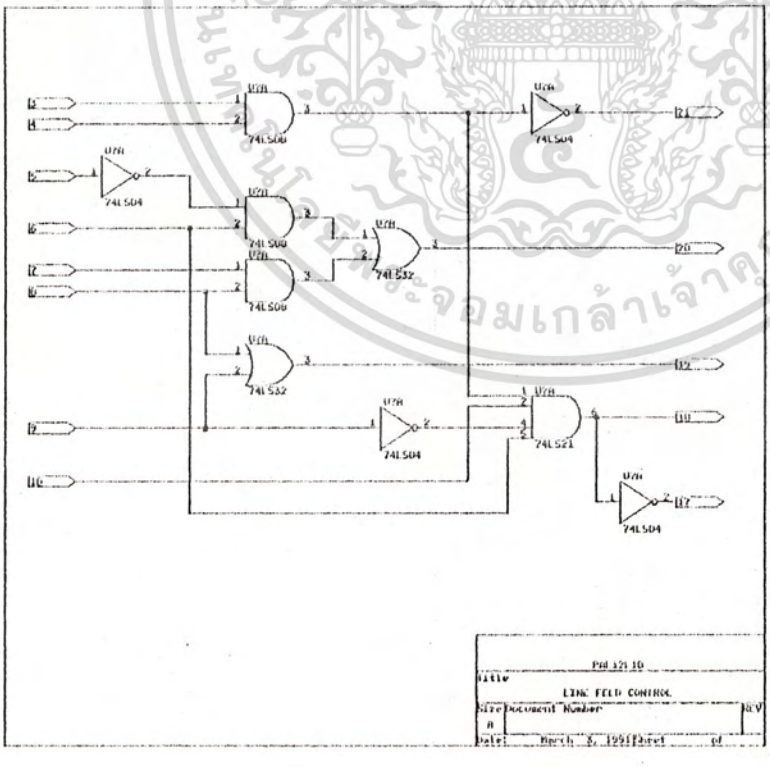
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



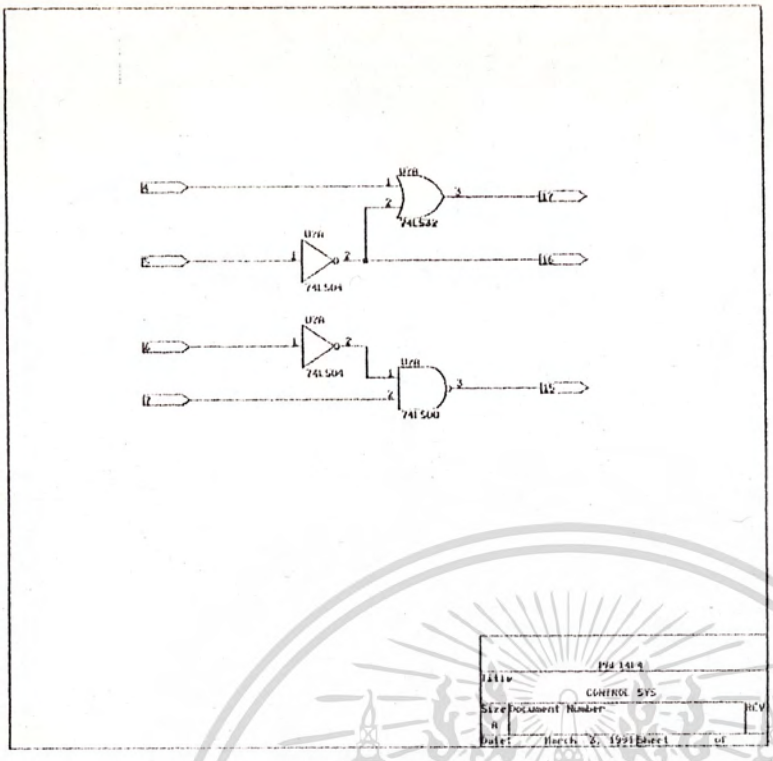


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* THIS PROGRAM USE ONLY IMAGE PROJECT .    */
/* IT WRITE BY Mr. CHAIRAT KHAOEAM .        */
/* IF YOU NEVER USE ... YOU CAN DELETE IT .*/
/*  THANK  YOU  */

# define CA     0x0006      /* # ch3 Base And Current Address */
# define CWC    0x0007      /* # ch3 Base And Current Word Count */
# define RSR    0x0008      /* Write Command Register */
# define WRR    0x0009      /* Write Request Register */
# define WSMRB 0x000A       /* Write Single Mask Register Bit */
# define WMR    0x000B      /* Write Mode Register */
# define CBPFF 0x000C       /* Clear Byte Pointer Flip-Flop */
# define RTR    0x000D      /* Write Master Clear */
# define WAMRB 0x000F       /* Write All Mask Register Bits */
# define PR     0x82        /* # ch3 Page Register */
# define CCP1   0x300
          /* Command Crad Port1 Write Mode,Selectport */
          /*  Count Control System Soft Ware Reset */
          /*  And control Sub-Port */
# define CCP2   0x301       /* Command Crad Port2 */
# define CCP3   0x302       /* Command Crad Port3 */
# define CCP4   0x303       /* Command Crad Port4 */
# define DMAUSE  0x7000     /* Location For DMA Use */
# define USEPAG  0x7        /* Set Value Pag Register */
# define DATAMOVE  16384
# define LOWAC       0xff
# define HIAC        0x3f
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*=============================================================

    Mr. CHAIRAT     KHAOEAM   (HEADER)
    Mr. SAKDA       VISUTHIWAT
    Mr. RATCHAR     PATHAMAPONGSAR

=============================================================
USE RUN ON IBMPC /XT/AT 286  MONOCHROM
USE DMA #3 FOR TRANSFER DATA
*/

  # include <stdlib.h>
  # include <stdio.h>
  # include <conio.h>
  # include <chairat.h>
  # include <dos.h>
  # include <dir.h>
  # include <mem.h>


  int i,j,size;
  char databuffer[DATAMOVE];
  FILE *fload,*fsave;
  char namefload[30],namefsave[30];
  char ch,errorh = 'n';

  void set_port(sub_mode,band,pagera)
   char sub_mode;
   short band,pagera;
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
switch(sub_mode)
  {
   case 'p' :  outportb(CCP4,0x88);break;
                                /* command port 0x300-0x303 */
   case 'i' :  outportb(CCP1,0x80);
                                     /* clear  system */
               outportb(CCP1,0xa0);
                                    /* not clear system */
             for(i=0;i<4;i++)
              {
               outportb(CCP1,0xb0);
               outportb(CCP1,0xa0);     /* count #4 clock */
              }
             break;
   case 'n' :  outportb(CCP1,0x01);break;
   case 'd' :  outportb(CCP1,0x00);break;
   case 'b' :  outportb(CCP1,0x80);break;
   case 't' :  outportb(CCP1,0x6c);
                                /* to command word sub port. */
               outportb(CCP3,0x9b);
                                /*  all input port. */
               outportb(CCP2,pagera);
               outportb(CCP1,0x00);
                break;
   case 's' :  outportb(CCP1,0xec);
                                /* to command word sub port */
               outportb(CCP3,0x9b);
                                /*  all input port */
               outportb(CCP2,pagera);
               outportb(CCP1,0xa3+((band-1)*0x4));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        /* to mode WF */
                    break;

        case 'l' :  outportb(CCP1,0xec);  /* select band port */
                    switch(band){
                      case 3  : outportb(CCP3,0x92);
                                        /*  out port C */
                                break;
                      case 2  : outportb(CCP3,0x99);
                                        /*  out port B */
                                break;
                      case 1  : outportb(CCP3,0x8b);
                                        /*  out port A */
                                break;
                    }
                    outportb(CCP2,pagera);
                    outportb(CCP1,0xa2+((band-1)*0x4));
                                        /* to mode RF */
                    break;
    }
  }


void oksystem()
 {
   int loop1;
   for(loop1=0;(loop1<4)&(inport(CCP2) & 0x10);loop1++)
    {
      set_port('p',0,0);
      set_port('i',0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
   set_port('d',0,0);
   if (loop1== 3)
    {
      printf("\n\n              Error on Hard ware !");
      errorh = 'y';
    }
   else
    {
      printf("\n\n                    Hard ware  OK ");
      errorh = 'n';
    }
}


void set.dma(dma_mode)
  char dma_mode;
  {
   gotoxy(40,20);
   outportb(CBPFF,0x67);          /* first/last flip-flop */
   if (dma_mode=='s')
    {
      printf("Move data from I/O to FILE ");
      outportb(WMR,0x97);          /* 10010111 USE MODE AUTO*/
    }
   if (dma_mode=='l')
    {
      printf("Move data from FILE to I/O ");
      outportb(WMR,0x9b);           /* 10011011 USE MODE AUTO*/
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    outportb(PR,USEPAG);          /* set page address */
    outportb(CA,0x00);            /* low byte current adrress */
    outportb(CA,0x00);            /* hi byte current address */
    outportb(CWC,LOWAC);          /* low byte current word count */
    outportb(CWC,HIAC);           /* hi byte current word count */
    outportb(WSMRB,0x03);         /* clrscr  single mask bit */
}


void menu_1(short *band2,char sele)
  {
    char band3,chband;
    clrscr();
    gotoxy(20,4);
    switch(sele)
    {
     case '1' : printf("  ****    LOAD  FUNCTION   ****");
                printf("\n\n        Enter   file  name ? - ");
                scanf("%s",namefload);break;
      case 's' : printf("  ****    SAVE  FUNCTION   ****");
                printf("\n\n        Enter   file  name ? - ");
                scanf("%s",namefsave);break;
     }
    *band2 =  1;
   }

void menu(char *usemenu1)
 {
   char c1;
   do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      c1 = 'n';
      clrscr();
      printf("              MAIN  MENU \n");
      printf("              1 : Digitized..\n");
      printf("              2 : Ram to TV..\n");
      printf("              3 : Load file..\n");
      printf("              4 : Save file..\n");
      printf("              5 : Directory..\n");
      printf("              6 : Quit..\n\n\n");
      printf("       ENTER  MENU  SELECT 1..6  ?  ");
      *usemenu1 = getche();
      switch(*usemenu1)
       {
        case '1' : c1 = 'y'; break;
        case '2' : c1 = 'y'; break;
        case '3' : c1 = 'y'; break;
        case '4' : c1 = 'y'; break;
        case '5' : c1 = 'y'; break;
        case '6' : c1 = 'y'; break;
       }
     }
   while(c1 != 'y');
}

void digitize()
 {
   clrscr();
   printf(" ****    DIGITIZED  FUNCTION   ****");
   set_port('n',0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


void ram_to_tv()
{
   clrscr();
   printf("  ****     RAM TO TV  FUNCTION    ****");
   set_port('d',0,0);
}


void load_file ()
  {
    int loop,m;
    char sys= 'y';
    short band;
    menu_1(&band,'1');
    oksystem(&sys);ch = 'y';
    if ((fload=fopen(namefload,"rb"))==NULL)
     {
       gotoxy(40,1);
       printf("Cant'n open file ");
       printf(namefload);
       getch();
       ch = 'n';
     }
    else
       fclose(fload);
    if (ch == 'y' & errorh == 'n')
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setdma('l');
        for(loop=0;(loop<16);loop++)
         {
           gotoxy(46,22);printf("%3d  Kbyte",(1+loop)*16);
           fload=fopen(namefload,"rb");
           for(m=0;m<loop;m++)
           fseek(fload,DATAMOVE,1);
           fread(&databuffer,DATAMOVE,1,fload);
           fclose(fload);
           movedata(_DS,(unsigned)databuffer,DMAUSE,0,DATAMOVE);
           set_port('i',0,0);                    /* into system */
           set_port('l',band,loop);              /* set mode RF */
           outportb (WRR,0x0f);  /* set request register start dma */
           set_port('b',0,0);        /* Command Card1 To RN MODE */
         }
       set_port('d',0,0);          /* Command Card1 To RN MODE */
       set_port('t',0,0);          /* set seconed port to read all */
    }
  }

void save_file ()
  {
   int loop;
   short band;
   menu_1(&band,'s');
   ch = 'y';
   if ((fsave=fopen(namefsave,"wb"))==NULL)
    {
      gotoxy(40,1);
      printf("Cant'n open file ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf(namefsave);
    ch = 'n';
    getch();
  }
 else
    fclose(fsave);
 if(ch == 'y' & errorh == 'n')
   {
    oksystem();
    setdma('s');
    for(loop=0;loop<16;loop++)
     {
     gotoxy(46,22);printf("%3d  Kbyte",(1+loop)*16);
     set_port('i',0,0);
     set_port('s',band,loop);
                    /* Command Card1 To Write File (WF) MODE */
     outportb (WRR,0x0f);
                    /*   set request register start dma */
     set_port('b',0,0);
                    /* Command Card1 To RN MODE */
     movedata(DMAUSE,0,_DS,(unsigned)databuffer,DATAMOVE);
     fsave=fopen(namefsave,"ab");
     fseek(fsave,0L,2);
     fwrite(&databuffer,DATAMOVE,1,fsave);
     fclose(fsave);
    }
   set_port('d',0,0);          /* Command Card1 To RN MODE */
   set_port('t',0,0);       /* set seconed port to read all */
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void directory()
 {
  struct ffblk ffblk;
  int done;
  char namesc[20];
  clrscr();
  printf("\ndirectory list of ");
  scanf("%s",namesc);
  printf("\n");
  done = findfirst(namesc,&ffblk,0);
  i=0;
  while(!done)
   {
    if(i ==23)
    {
     i=0;
      printf("Strike a key when ready . . .\n");
      getch();
    }
   printf("  %15s   %#10lu\n",ffblk.ff_name,ffblk.ff_fsize);
   done = findnext(&ffblk);
   i++;
  }
 printf("\n\n Press anykey to exit");
 getch();
}

 main()
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char usemenu1,c0;
    set_port('p',0,0);              /* inti port control */
    set_port('t',0,0);          /* set seconed port to read all */
    do
    {
      menu(&usemenu1);
      switch (usemenu1)
       {
        case '1' :  digitize();break;
        case '2' :  ram_to_tv();break;
        case '3' :  load_file ();break;
        case '4' :  save_file ();break;
        case '5' :  directory();break;
       }
     if (usemenu1 != '6')
      {
       gotoxy(20,25);
       printf("                    USE  KEY  'm'  TO  MAIN  MENU ");
       do
        c0 = getch();
       while(c0 != 'm');
      }
    }
  while(usemenu1 != '6');
  clrscr();
    set_port('d',0,0);
  printf(" THANK  YOU ....");
  delay(300);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Product Preview

CA3318, CA3318C

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

Features:

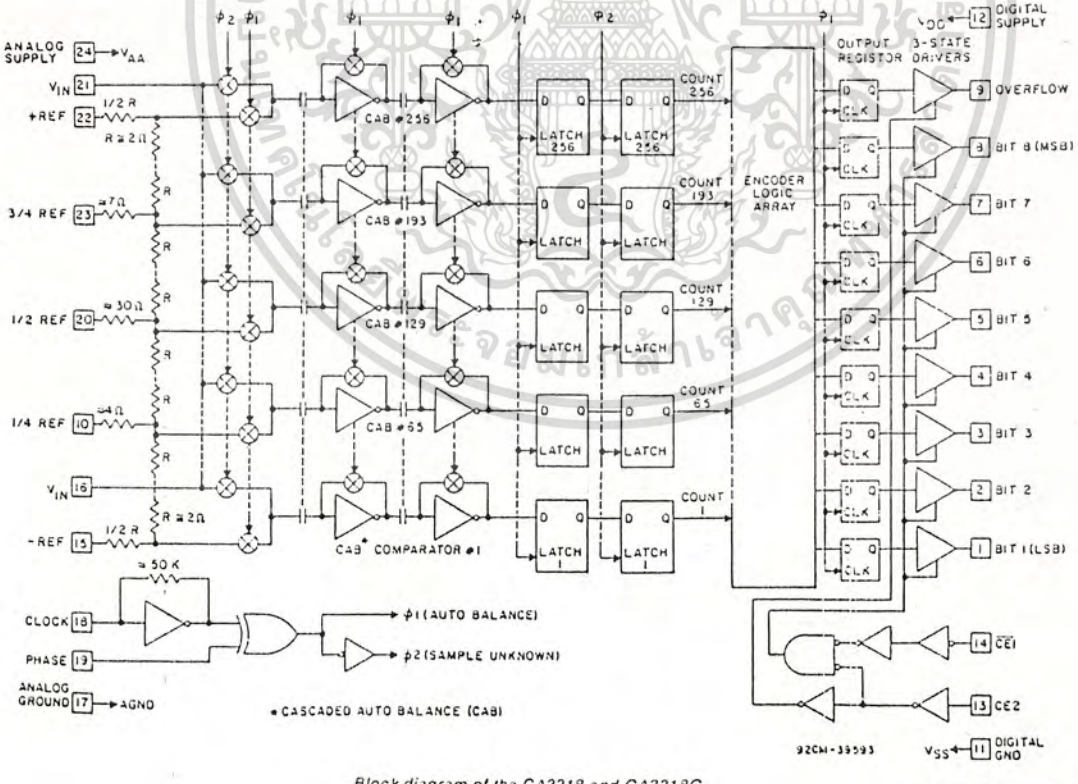
- *Pin compatible with 41051/CA3308*
- *CMOS/SOS low power*
- *Flash (Parallel) conversion technique*
- *15 MSPS conversion rate at 5 V (CA3318C)*
- *20 MSPS conversion rate at 5 V (CA3318)*
- *1 LSB differential linearity*
- *1.5 LSB integral linearity*
- *Single 4 to 6.5 V supply*
- *8 latched bit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

Applications:

- *Especially suited for high-speed conversion 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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Product Preview

CA3306, CA3306A, CA3306C

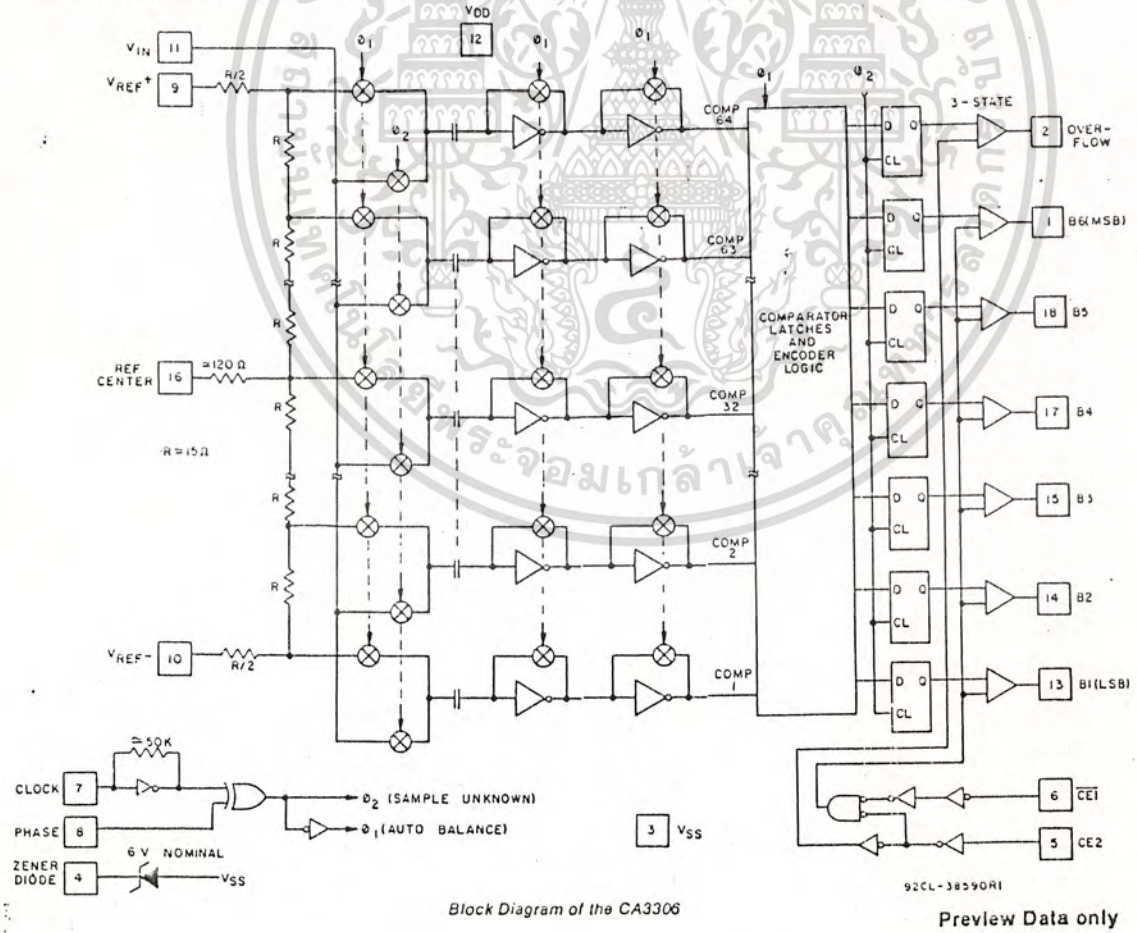
CMOS High-Speed 6-Bit Flash A/D Converter

Features:

- *Improved pin-for-pin retrofits for CA3300*
- *CMOS/SOS low power*
- *Flash (Parallel) conversion technique*
- *15 MSPS conversion rate at 5 V*
- *1/4 LSB accuracy*
- *Single 3 to 6 V supply*
- *6 latched-bit outputs plus overflow*
- *May be stacked for higher resolution*
- *May be parallel for double speed*

The CA3306 family members are pin-for-pin retrofits for the CA3300 (File 1316), but offering improved speed and linearity. All functions of the CA3300 are carried over: the ability to stack devices for higher resolution, parallel devices for doubled speed, and the availability of a built-in zener reference. Accurate digitizing at video speeds is now possible with only a single 5 volt supply (8 volts required for CA3300), and a tighter linearity is guaranteed at a lower reference (full scale) range.

The CA3306-series devices are supplied in 18-lead dual-in-line plastic packages (E suffix) and in 18-lead dual-in-line ceramic packages (D suffix).



Block Diagram of the CA3306

Preview Data only

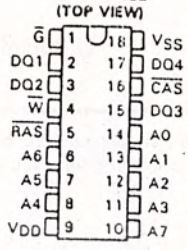
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 65,536 X 4 Organization
- Single 5-V Supply (10% Tolerance)
- JEDEC Standardized Pinout
- Pinout Identical to TMS4416 (16K X 4 Dynamic RAM)
- Performance Ranges:

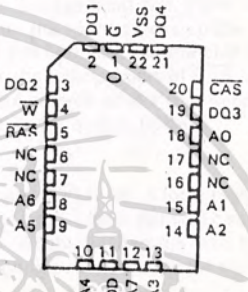
	ACCESS TIME ROW ADDRESS (MAX)	ACCESS TIME COLUMN ADDRESS (MAX)	READ OR WRITE CYCLE (MIN)	READ-MODIFY-WRITE CYCLE (MIN)
TMS4464-12	120 ns	60 ns	230 ns	310 ns
TMS4464-15	150 ns	75 ns	260 ns	345 ns
TMS4464-20	200 ns	100 ns	330 ns	435 ns

- Long Refresh Period . . . 4 ms (Max)
- Low Refresh Overhead Time . . . As Low As 1.3% of Total Refresh Period
- On-Chip Substrate Bias Generator
- All Inputs, Outputs, and Clocks Fully TTL Compatible
- 3-State Unlatched Output
- Early Write or $\overline{G}$ to Control Output Buffer Impedance
- Page-Mode Operation for Faster Access
- Power Dissipation As Low As:
  - Operating . . . 275 mW (Typ)
  - Standby . . . 12.5 mW (Typ)
- $\overline{RAS}$-Only Refresh Mode
- $\overline{CAS}$-Before-$\overline{RAS}$ Refresh Mode

N PACKAGE
(TOP VIEW)



FM PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A7	Address Inputs
$\overline{CAS}$	Column-Address Strobe
DQ1-DQ4	Data In/Data Out
$\overline{G}$	Output Enable
NC	No Connection
$\overline{RAS}$	Row-Address Strobe
V_{DD}	5-V Supply
V_{SS}	Ground
$\overline{W}$	Write Enable

description

The TMS4464 is a high-speed, 262,144-bit dynamic random-access memory, organized as 65,536 words of four bits each. It employs state-of-the-art SMOS (scaled MOS) N-channel double-level polysilicon/polycide gate technology for very high performance combined with low cost and improved reliability.

This device features maximum $\overline{RAS}$ access times of 120 ns, 150 ns, or 200 ns. Typical power dissipation is as low as 275 mW operating and 12.5 mW standby.

New SMOS technology permits operation from a single 5-V supply, reducing system power supply and decoupling requirements, and easing board layout. I_{DD} peaks are 125 mA typical, and a −1-V input voltage undershoot can be tolerated, minimizing system noise considerations.

All inputs and outputs, including clocks, are compatible with Series 74 TTL. All address and data-in lines are latched on chip to simplify system design. Data out is unlatched to allow greater system flexibility.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1985 Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAS-before-RAS refresh

The CAS-before-RAS refresh is utilized by bringing CAS low earlier than RAS (see parameter t_{CLRL}) and and holding it low after RAS falls (see parameter t_{RLCHR}). For successive CAS-before-RAS refresh cycles, CAS can remain low while cycling RAS. The external address is ignored and the refresh address is generated internally.

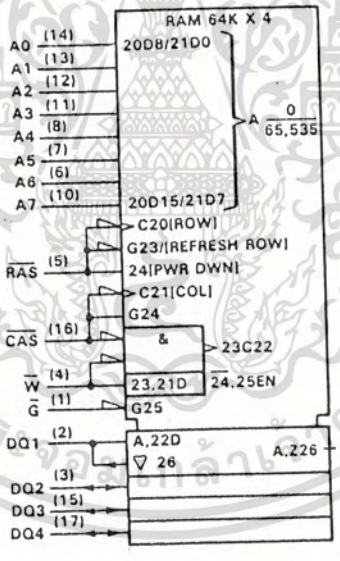
page mode

Page-mode operation allows effectively faster memory access by keeping the same row address and strobing random column addresses onto the chip. Thus, the time required to setup and strobe sequential row addresses for the same page is eliminated. The maximum number of columns that can be addressed is determined by $t_{w(RL)}$, the maximum RAS low pulse duration.

power up

To achieve proper device operation, an initial pause of 200 μs is required after power up followed by a minimum of eight initialization cycles.

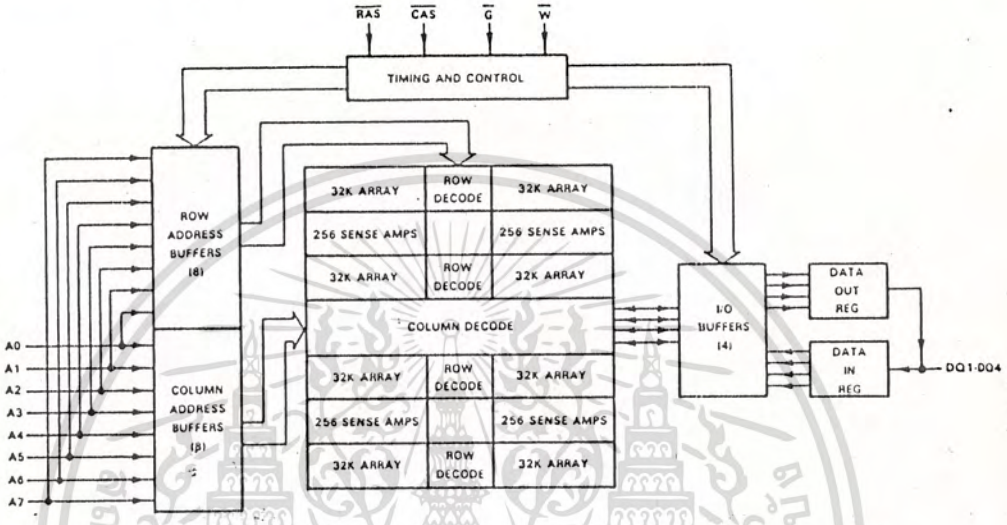
logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for the dual-in-line package.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

functional block diagram



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Voltage on any pin including V_{DD} supply (see Note 1) . . . . . −1 V to 7 V
Short circuit output current . . . . . 50 mA
Power dissipation . . . . . 1 W
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

† Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values in this data sheet are with respect to V_{SS}.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{DD}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	High-level input voltage	2.4		$V_{DD}+1$	V
V_{IL}	Low-level input voltage (see Note 2)	−1		0.8	V
T_A	Operating free-air temperature	0		70	°C

NOTE 2: The algebraic convention, where the more negative (less positive) limit is designated as maximum, is used in this data sheet for logic voltage levels only.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics over full ranges of recommended operating conditions (unless otherwise noted)

PARAMETER		TEST CONDITIONS	TMS4464-12			UNIT
			MIN	TYP†	MAX	
V_{OH}	High-level output voltage	I_{OH} = −5 mA	2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 4.2 mA			0.4	V
I_I	Input current (leakage)	V_I = 0 V to 6.5 V, V_{DD} = 5 V, All other pins = 0 V to 6.5 V			±10	μA
I_O	Output current (leakage)	V_O = 0 V to 5.5 V, V_{DD} = 5 V, $\overline{CAS}$ high, All outputs open			±10	μA
I_{DD1}	Average operating current during read or write cycle	t_c = minimum cycle, All outputs open		65	80	mA
I_{DD2}	Standby current	After 1 memory cycle, DQ1-DQ4 held at > 0 V, $\overline{RAS}$ and $\overline{CAS}$ high, All outputs open		2.5	5	mA
I_{DD3}	Average refresh current	t_c = minimum cycle, $\overline{RAS}$ low, $\overline{CAS}$ high, All outputs open		50	60	mA
I_{DD4}	Average page-mode current	$t_{c(P)}$ = minimum cycle, $\overline{RAS}$ low, $\overline{CAS}$ cycling, All outputs open		45	55	mA

PARAMETER		TEST CONDITIONS	TMS4464-15			TMS4464-20			UNIT
			MIN	TYP†	MAX	MIN	TYP†	MAX	
V_{OH}	High-level output voltage	I_{OH} = −5 mA	2.4			2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 4.2 mA			0.4			0.4	V
I_I	Input current (leakage)	V_I = 0 V to 6.5 V, V_{DD} = 5 V, All other pins = 0 V to 6.5 V			±10			±10	μA
I_O	Output current (leakage)	V_O = 0 V to 5.5 V, V_{DD} = 5 V, $\overline{CAS}$ high All outputs open			±10			±10	μA
I_{DD1}	Average operating current during read or write cycle	t_c = minimum cycle All outputs open		55	70		50	60	mA
I_{DD2}	Standby current	After 1 memory cycle, DQ1-DQ4 held at > 0 V, $\overline{RAS}$ and $\overline{CAS}$ high, All outputs open		2.5	5		2.5	5	mA
I_{DD3}	Average refresh current	t_c = minimum cycle, $\overline{RAS}$ low, $\overline{CAS}$ high, All outputs open		45	55		40	50	mA
I_{DD4}	Average page-mode current	$t_{c(P)}$ = minimum cycle, $\overline{RAS}$ low, $\overline{CAS}$ cycling, All outputs open		40	50		30	40	mA

†All typical values are at T_A = 25°C and nominal supply voltages.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

capacitance over recommended supply voltage range and operating free-air temperature range, f = 1 MHz

PARAMETER		TMS4464		UNIT
		TYP†	MAX	
$C_{i(A)}$	Input capacitance, address inputs	4	7	pF
$C_{i(RC)}$	Input capacitance strobe inputs	8	10	pF
$C_{i(W)}$	Input capacitance, write enable input	8	10	pF
$C_{i/o}$	Output capacitance	8	10	pF

† All typical values are at T_A = 25 °C and nominal supply voltages.

switching characteristics over recommended supply voltage range and operating free-air temperature range

PARAMETER		TEST CONDITIONS	ALT. SYMBOL	TMS4464-12		UNIT
				MIN	MAX	
$t_{a(C)}$	Access time from $\overline{CAS}$	$t_{RLCL} \geq$ MAX, C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{CAC}		60	ns
$t_{a(R)}$	Access time from $\overline{RAS}$	t_{RLCL} = MAX, C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{RAC}		120	ns
$t_{a(G)}$‡	Access time after $\overline{G}$ low	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{GAC}		35	ns
$t_{dis(CH)}$	Output disable time after $\overline{CAS}$ high	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{OFF}	0	30	ns
$t_{dis(G)}$	Output disable time after $\overline{G}$ high	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{GOFF}	0	30	ns

switching characteristics over recommended supply voltage range and operating free-air temperature range

PARAMETER		TEST CONDITIONS	ALT. SYMBOL	TMS4464-15		TMS4464-20		UNIT
				MIN	MAX	MIN	MAX	
$t_{a(C)}$	Access time from $\overline{CAS}$	$t_{RLCL} \geq$ MAX, C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{CAC}		75		100	ns
$t_{a(R)}$	Access time from $\overline{RAS}$	t_{RLCL} = MAX, C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{RAC}		150		200	ns
$t_{a(G)}$‡	Access time after $\overline{G}$ low	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{GAC}		45		55	ns
$t_{dis(CH)}$	Output disable time after $\overline{CAS}$ high	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{OFF}	0	30	0	35	ns
$t_{dis(G)}$	Output disable time after $\overline{G}$ high	C_L = 100 pF, Load = 2 Series 74 TTL gates	t_{GOFF}	0	30	0	35	ns

‡ $t_{a(C)}$ and $t_{a(R)}$ must be satisfied to guarantee $t_{a(G)}$.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing requirements over recommended supply voltage range and operating free-air temperature range

		ALT. SYMBOL	TMS4464-12		UNIT
			MIN	MAX	
$t_{c(P)}$	Page-mode cycle time	t_{PC}	120		ns
$t_{c(PM)}$	Page-mode cycle time (read-modify-write cycle)	t_{PCM}	200		ns
$t_{c(rd)}$	Read cycle time†	t_{RC}	230		ns
$t_{c(W)}$	Write cycle time	t_{WC}	230		ns
$t_{c(rdW)}$	Read-write/read-modify-write cycle time	t_{RWC}	310		ns
$t_{w(CH)P}$	Pulse duration, $\overline{CAS}$ high (page mode)	t_{CP}	50		ns
$t_{w(CH)}$	Pulse duration, $\overline{CAS}$ high (non-page mode)	t_{CPN}	50		ns
$t_{w(CL)}$	Pulse duration, $\overline{CAS}$ low‡	t_{CAS}	60	10,000	ns
$t_{w(RH)}$	Pulse duration, $\overline{RAS}$ high	t_{RP}	100		ns
$t_{w(RL)}$	Pulse duration, $\overline{RAS}$ low§	t_{RAS}	120	10,000	ns
$t_{w(W)}$	Write pulse duration	t_{WP}	40		ns
t_t	Transition times (rise and fall) for $\overline{RAS}$ and $\overline{CAS}$	t_T	3	50	ns
$t_{su(CA)}$	Column-address setup time	t_{ASC}	0		ns
$t_{su(RA)}$	Row-address setup time	t_{ASR}	0		ns
$t_{su(D)}$	Data setup time	t_{DS}	0		ns
$t_{su(rd)}$	Read-command setup time	t_{RCS}	0		ns
$t_{su(WCL)}$	Early-write command setup time before $\overline{CAS}$ low	t_{WCS}	0		ns
$t_{su(WCH)}$	Write-command setup time before $\overline{CAS}$ high	t_{CWL}	40		ns
$t_{su(WRH)}$	Write-command setup time before $\overline{RAS}$ high	t_{RWL}	40		ns
$t_{h(CLCA)}$	Column-address hold time after $\overline{CAS}$ low	t_{CAH}	20		ns
$t_{h(RA)}$	Row-address hold time	t_{RAH}	15		ns
$t_{h(RLCA)}$	Column-address hold time after $\overline{RAS}$ low	t_{AR}	80		ns
$t_{h(CLD)}$	Data hold time after $\overline{CAS}$ low	t_{DH}	35		ns
$t_{h(RLD)}$	Data hold time after $\overline{RAS}$ low	t_{DHR}	95		ns
$t_{h(WLD)}$	Data hold time after $\overline{W}$ low	t_{DH}	35		ns
$t_{h(CHrd)}$	Read-command hold time after $\overline{CAS}$ high	t_{RCH}	0		ns
$t_{h(RHrd)}$	Read-command hold time after $\overline{RAS}$ high	t_{RRH}	10		ns
$t_{h(CLW)}$	Write-command hold time after $\overline{CAS}$ low	t_{WCH}	35		ns
$t_{h(RLW)}$	Write-command hold time after $\overline{RAS}$ low	t_{WCR}	95		ns

Continued next page.

†All cycle times assume t_t = 5 ns.

‡In a read-modify-write cycle, t_{CLWL} and $t_{su(WCH)}$ must be observed. Depending on the user's transition times, this may require additional $\overline{CAS}$ low time ($t_{w(CL)}$).

§In a read-modify-write cycle, t_{RLWL} and $t_{su(WRH)}$ must be observed. Depending on the user's transition times, this may require additional $\overline{RAS}$ low time ($t_{w(RL)}$).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing requirements over recommended supply voltage range and operating free-air temperature range (continued)

		ALT. SYMBOL	TMS4464-12 MIN	TMS4464-12 MAX	UNIT
t_{RLCHR}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high[1]	t_{CHR}	25		ns
t_{RLCH}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high	t_{CSH}	120		ns
t_{CHRL}	Delay time, $\overline{CAS}$ high to $\overline{RAS}$ low	t_{CRP}	0		ns
t_{RHCL}	Delay time, $\overline{RAS}$ high to $\overline{CAS}$ low[1]	t_{RCP}	0		ns
t_{CLRH}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ high	t_{RSH}	60		ns
t_{CLWL}	Delay time, $\overline{CAS}$ low to $\overline{W}$ low (read-modify-write cycle only)[#]	t_{CWD}	95		ns
t_{CLRL}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ low[1]	t_{CSR}	25		ns
t_{RLCL}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ low (maximum value specified only to guarantee access time)	t_{RCD}	25	60	ns
t_{RLWL}	Delay time, $\overline{RAS}$ low to $\overline{W}$ low (read-modify-write cycle only)[#]	t_{RWD}	155		ns
t_{GHD}	Delay time, $\overline{G}$ high before data applied at DQ	t_{GDD}	30		ns
t_{rf}	Refresh time interval	t_{REF}		4	ms

[1] $\overline{CAS}$-before-$\overline{RAS}$ refresh option only.
[#] $\overline{G}$ must disable the output buffers prior to applying data to the device.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing requirements over recommended supply voltage range and operating free-air temperature range (continued)

		ALT. SYMBOL	TMS4464-15		TMS4464-20		UNIT
			MIN	MAX	MIN	MAX	
$t_{c(P)}$	Page-mode cycle time	t_{PC}	145		190		ns
$t_{c(PM)}$	Page-mode cycle time (read-modify-write cycle)	t_{PCM}	230		295		ns
$t_{c(rd)}$	Read cycle time†	t_{RC}	260		330		ns
$t_{c(W)}$	Write cycle time	t_{WC}	260		330		ns
$t_{c(rdW)}$	Read-write/read-modify-write cycle time	t_{RWC}	345		435		ns
$t_{w(CH)P}$	Pulse duration, $\overline{CAS}$ high (page mode)	t_{CP}	60		80		ns
$t_{w(CH)}$	Pulse duration, $\overline{CAS}$ high (non-page mode)	t_{CPN}	60		80		ns
$t_{w(CL)}$	Pulse duration, $\overline{CAS}$ low‡	t_{CAS}	75	10,000	100	10,000	ns
$t_{w(RH)}$	Pulse duration, $\overline{RAS}$ high	t_{RP}	100		120		ns
$t_{w(RL)}$	Pulse duration, $\overline{RAS}$ low§	t_{RAS}	150	10,000	200	10,000	ns
$t_{w(W)}$	Write pulse duration	t_{WP}	45		55		ns
t_t	Transition times (rise and fall) for $\overline{RAS}$ and $\overline{CAS}$	t_T	3	50	3	50	ns
$t_{su(CA)}$	Column-address setup time	t_{ASC}	0		0		ns
$t_{su(RA)}$	Row-address setup time	t_{ASR}	0		0		ns
$t_{su(D)}$	Data setup time	t_{DS}	0		0		ns
$t_{su(rd)}$	Read-command setup time	t_{RCS}	0		0		ns
$t_{su(WCL)}$	Early-write command setup time before $\overline{CAS}$ low	t_{WCS}	0		0		ns
$t_{su(WCH)}$	Write-command setup time before $\overline{CAS}$ high	t_{CWL}	45		60		ns
$t_{su(WRH)}$	Write-command setup time before $\overline{RAS}$ high	t_{RWL}	45		60		ns
$t_{h(CLCA)}$	Column-address hold time after $\overline{CAS}$ low	t_{CAH}	25		45		ns
$t_{h(RA)}$	Row-address hold time	t_{RAH}	15		20		ns
$t_{h(RLCA)}$	Column-address hold time after $\overline{RAS}$ low	t_{AR}	100		145		ns
$t_{h(CLD)}$	Data hold time after $\overline{CAS}$ low	t_{DH}	45		55		ns
$t_{h(RLD)}$	Data hold time after $\overline{RAS}$ low	t_{DHR}	120		155		ns
$t_{h(WLD)}$	Data hold time after $\overline{W}$ low	t_{DH}	45		55		ns
$t_{h(CHrd)}$	Read-command hold time after $\overline{CAS}$ high	t_{RCH}	0		0		ns
$t_{h(RHrd)}$	Read-command hold time after $\overline{RAS}$ high	t_{RRH}	10		15		ns
$t_{h(CLW)}$	Write-command hold time after $\overline{CAS}$ low	t_{WCH}	45		55		ns
$t_{h(RLW)}$	Write-command hold time after $\overline{RAS}$ low	t_{WCR}	120		155		ns

Continued next page.

†All cycle times assume t_t = 5 ns.

‡In a read-modify-write cycle, t_{CLWL} and $t_{su(WCH)}$ must be observed. Depending on the user's transition times, this may require additional $\overline{CAS}$ low time ($t_{w(CL)}$).

§In a read-modify-write cycle, t_{RLWL} and $t_{su(WRH)}$ must be observed. Depending on the user's transition times, this may require additional $\overline{RAS}$ low time ($t_{w(RL)}$).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timing requirements over recommended supply voltage range and operating free-air temperature range (concluded)

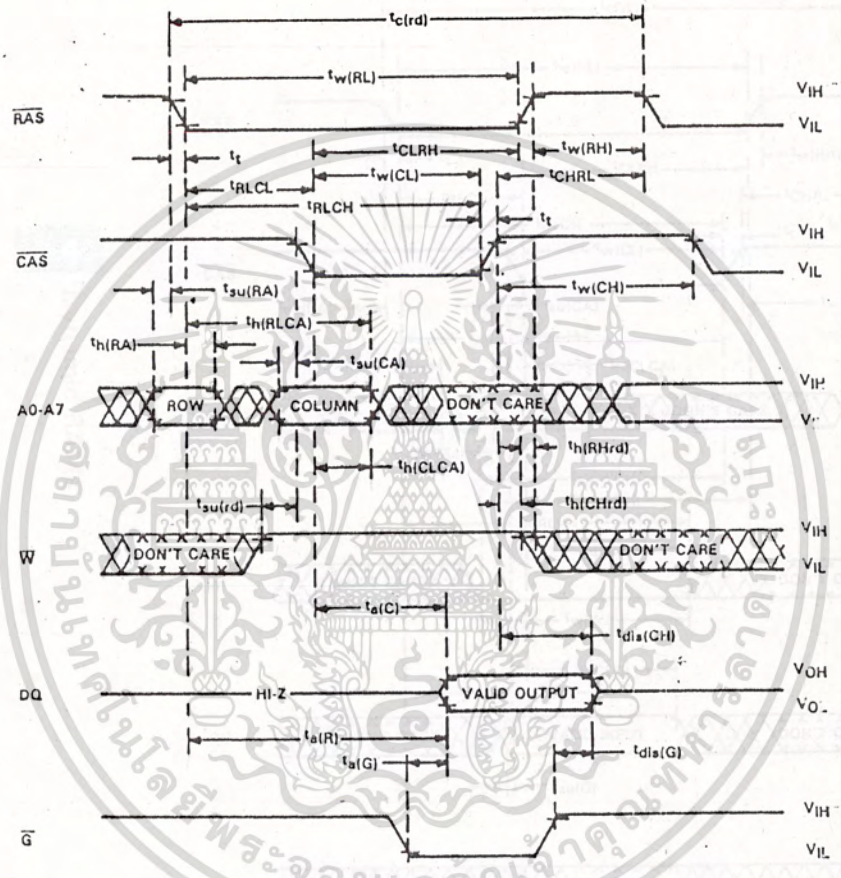
		ALT. SYMBOL	TMS4464-15		TMS4464-20		UNIT
			MIN	MAX	MIN	MAX	
t_{RLCHR}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high†	t_{CHR}	30		35		ns
t_{RLCH}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high	t_{CSH}	150		200		ns
t_{CHRL}	Delay time, $\overline{CAS}$ high to $\overline{RAS}$ low	t_{CRP}	0		0		ns
t_{RHCL}	Delay time, $\overline{RAS}$ high to $\overline{CAS}$ low†	t_{RCP}	0		0		ns
t_{CLRH}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ high	t_{RSH}	75		100		ns
t_{CLWL}	Delay time, $\overline{CAS}$ low to $\overline{W}$ low (read-modify-write cycle only)#	t_{CWD}	110		140		ns
t_{CLRL}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ low†	t_{CSR}	30		35		ns
t_{RLCL}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ low (maximum value specified only to guarantee access time)	t_{RCD}	25	75	30	100	ns
t_{RLWL}	Delay time, $\overline{RAS}$ low to $\overline{W}$ low (read-modify-write cycle only)#	t_{RWD}	185		240		ns
t_{GHD}	Delay time, $\overline{G}$ high before data applied at DQ	t_{GDD}	30		35		ns
t_{rf}	Refresh time interval	t_{REF}		4		4	ms

†$\overline{CAS}$-before-$\overline{RAS}$ refresh option only.
#$\overline{G}$ must disable the output buffers prior to applying data to the device.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

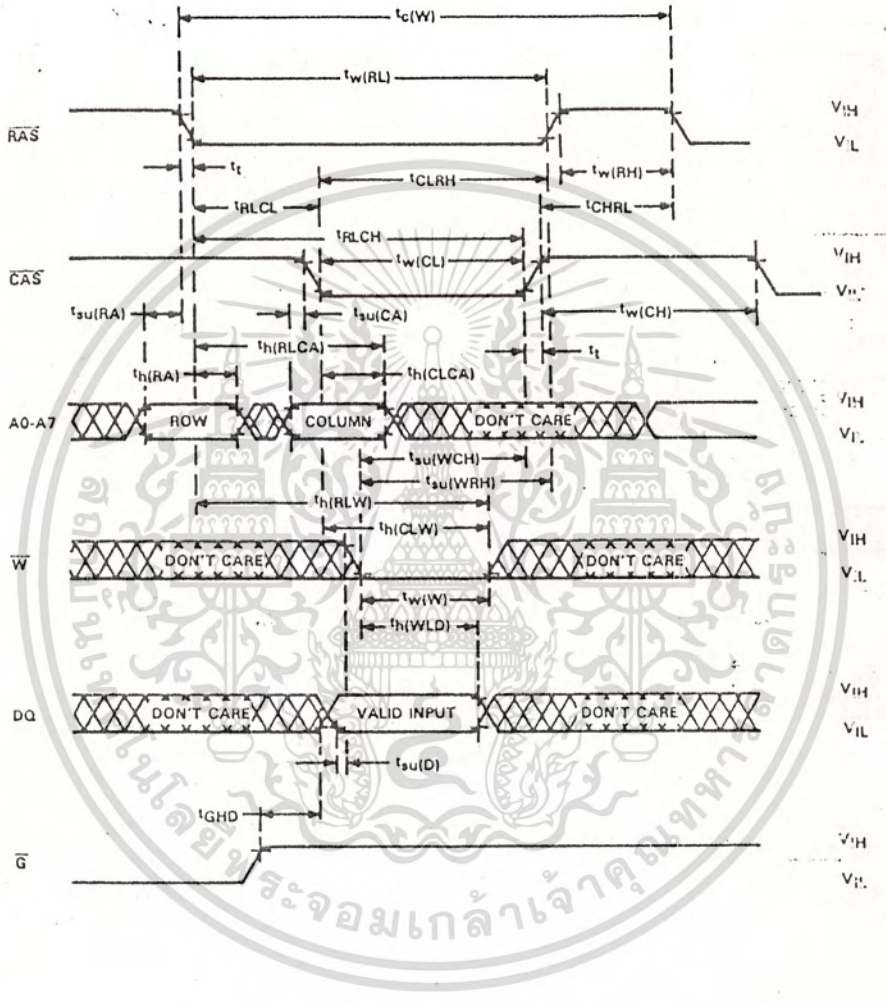
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

read cycle timing



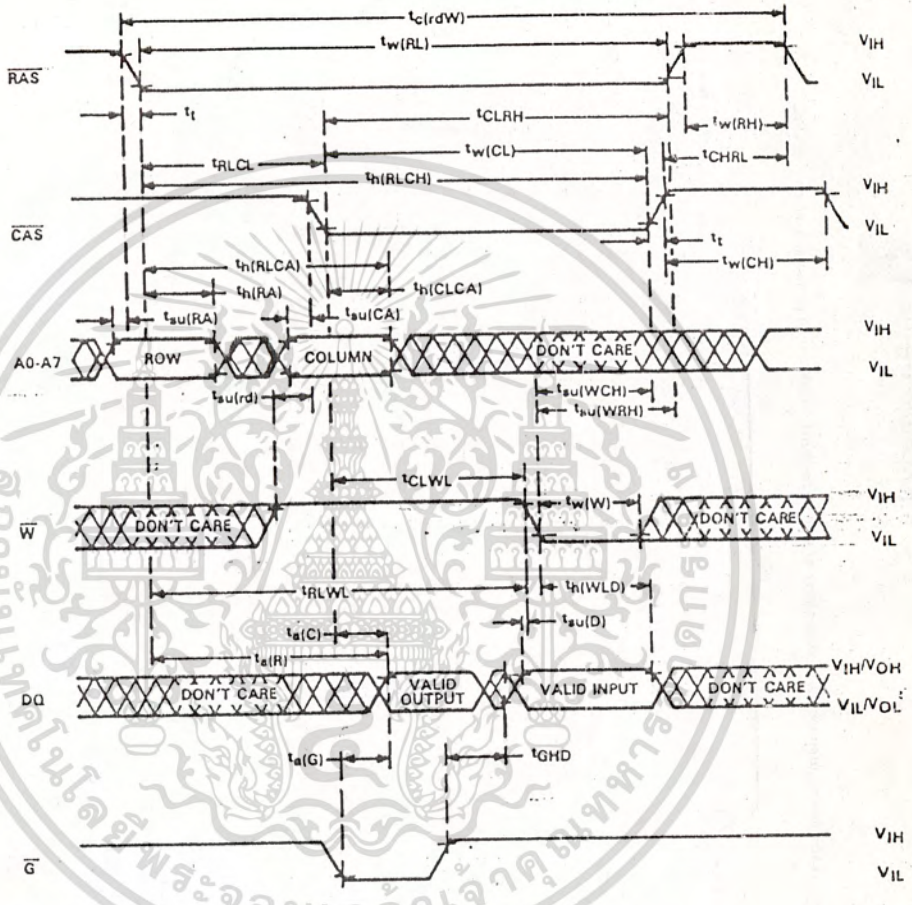
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

write cycle timing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

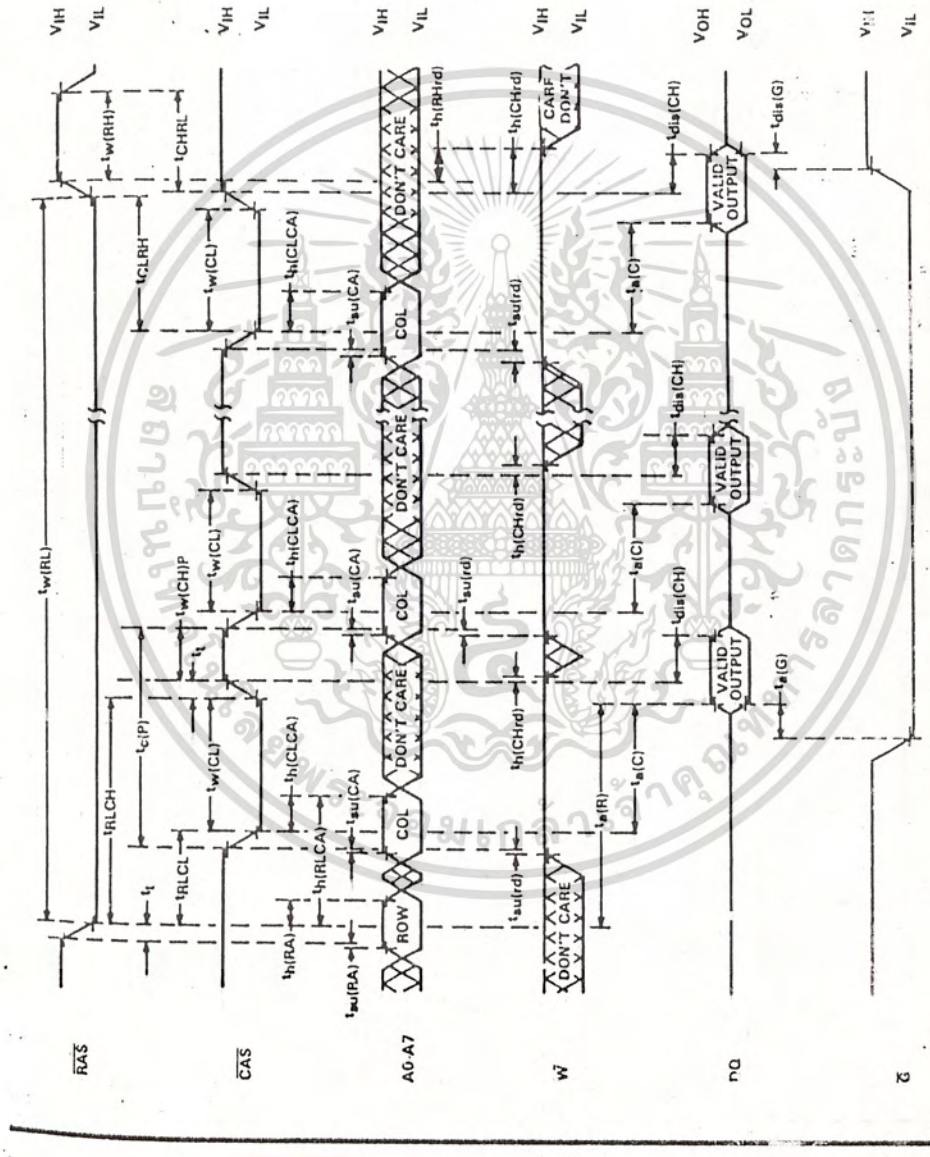
read-write/read-modify-write cycle timing



TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

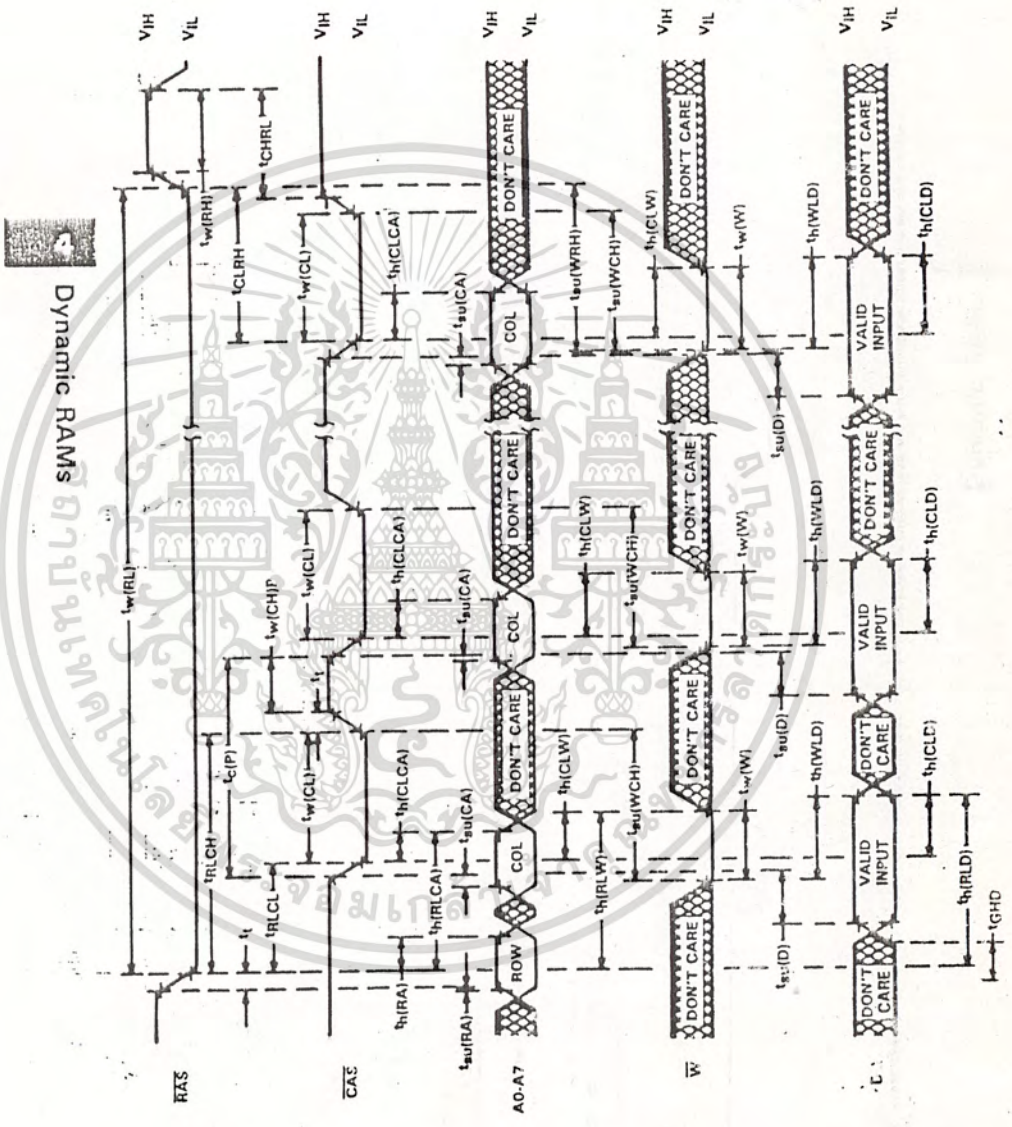
page-mode read cycle timing



NOTE 3: A write cycle or read-modify-write cycle can be intermixed with read cycles as long as the write and read-modify-write timing specifications are not violated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

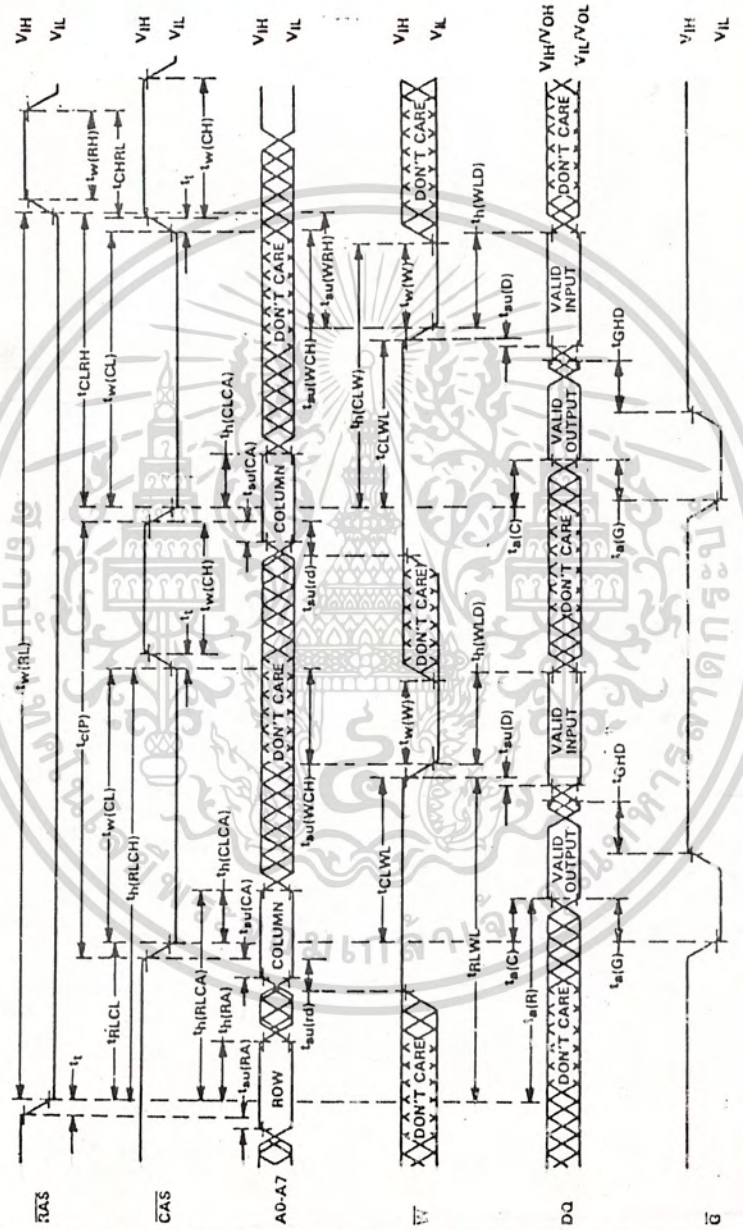
page-mode write cycle timing



TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

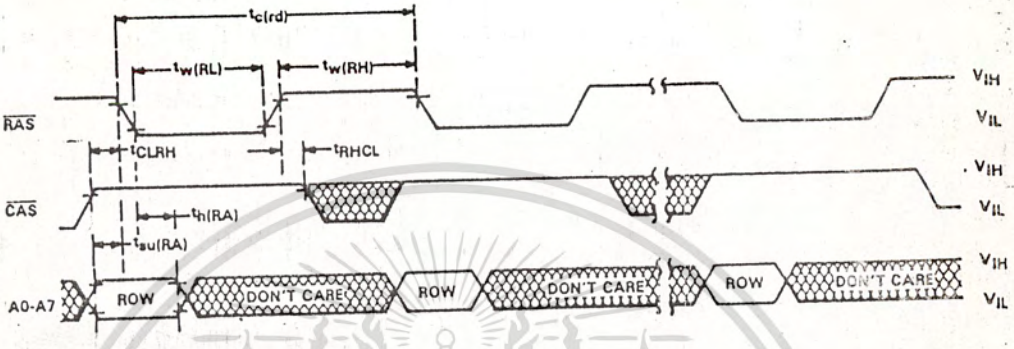
page-mode read-modify-write cycle timing



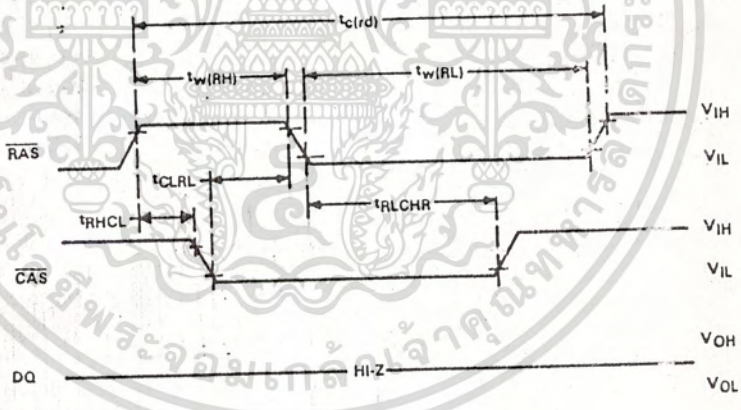
NOTE : A read cycle or a write cycle can be intermixed with read-modify-write cycles as long as read and write timing specifications are not violated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{RAS}$-only refresh cycle timing



$\overline{CAS}$-before-$\overline{RAS}$ refresh cycle timing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้