



การพัฒนาระบบเก็บภาพลายนิ้วมือแบบดิจิทัล

DIGITAL FINGER-PRINT ACQUISITION SYSTEM



โดย

นาย กิตติ อาการส

นาย สุชาติ แซ่อุ้น

วัน เดือน ปี	24.คค.2541
เลขทะเบียน	039155
เลขเรียกหนังสือ	ท.4๐๓๗4 ก๖๗1ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039155

การพัฒนาระบบเก็บภาพลายนิ้วมือแบบดิจิทัล

DIGITAL FINGER-PRINT ACQUISITION SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

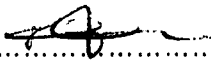

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์เรื่อง การพัฒนาระบบเก็บภาพลายนิ้วมือแบบดิจิทัล
DIGITAL FINGER-PRINT ACQUISITION SYSTEM
จัดทำโดย นาย กิตติ อาการส 38013176
 นาย สุชาติ แซ่ฮุ้น 38013214
อาจารย์ที่ปรึกษา ดร. กิติพล ชิตสกุล



ปริญญาานิพนธ์นี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ..........อาจารย์ที่ปรึกษา
()

วันที่ ๒๐, ๑๑, ๖๑.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดี ทั้งนี้เนื่องมาจากการได้รับความช่วยเหลือและการแนะนำจากบุคคลหลายฝ่าย ทางกลุ่มจึงขอถือโอกาสนี้ขอกราบขอขอบพระคุณบุคคลต่างๆ ดังต่อไปนี้ ขอขอบพระคุณท่านอาจารย์ ดร. กิตติพล ชิตสกุล ที่คอยให้คำปรึกษาแนะนำให้ไอซี A/D เบอร์ TDA8708A และให้ความเอื้อเฟื้อทางค่านเครื่องมือในการวัดผล ห้องในการทดลอง ขอขอบพระคุณท่านอาจารย์ ประภากร สุวรรณะ ที่ให้ยืม Pettern Generator ขอขอบคุณพี่อ็อคและสุวัฒน์ วิสวเซวง ที่ให้คำแนะนำในการเขียนโปรแกรมภาษาซี ตลอดจนเพื่อนทุกคนที่ให้ความสนใจเสมอมา

กิตติ อการส

(นาย กิตติ อการส)

สุชาติ แซ่อุ้น

(นาย สุชาติ แซ่อุ้น)

ผู้จัดทำ

การพัฒนาระบบเก็บภาพถ่ายนิ้วมือแบบดิจิทัล

นาย กิตติ อาการส

นาย สุชาติ แซ่ฮุ้น

ดร. กิตติพล ชิตสกุล (อาจารย์ที่ปรึกษา)

ภาคเรียนที่ 2 ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้กล่าวถึงการปรับปรุงพัฒนาระบบเก็บข้อมูลภาพถ่ายนิ้วมือระบบดิจิทัล ขนาด 512×512 จุดภาพ ข้อมูลที่เก็บเป็นลักษณะภาพระดับเทาที่มีความแตกต่างระดับขาวดำ 256 ระดับ ซึ่งข้อมูลขนาด 8 บิตที่ถูกแปลงเป็นดิจิทัลแล้วถูกเก็บลงหน่วยความจำขนาด 128 กิโลไบต์ 2 ตัวต่อหนึ่งภาพในลักษณะไบต์อาร์เรย์ ข้อมูลภาพจะถูกส่งต่อไปยังไมโครคอมพิวเตอร์โดยทำการส่งข้อมูลครั้งละ 4 บิต 2 ครั้งในหนึ่งไบต์ผ่านทางพริ้นเตอร์พอร์ต การทำงานของระบบถูกควบคุมภายใต้คอนโทรลเลอร์และรายละเอียดของโปรแกรมภาษาซีบนไมโครคอมพิวเตอร์ที่รับข้อมูลมาแสดงผลก็ได้รายงานไว้เช่นกัน

DIGITAL FINGER-PRINT ACQUISITION SYSTEM

Mr. Kitti Arkaros

Mr. Suchart Sae - Oun

Dr. Kitiphol Chitsakul (Advisor)

2nd Semester, Education Year 1997

Abstract

This thesis reports a development of finger - print image acquisition system. A finger-print digital image of 256 digital gray level with resolution of 512×512 pixel is obtained by quantizing the video signal. Once an image has been quantized, it is stored in two memory units of 128 kbytes per frame in byte array format. The data in memories are then send to a microcomputer via a printer port in 4 bits 2 times per byte. These operations are controlled by a controller. Moreover the program in C for display the image on PC is detailed.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 หลักการเก็บข้อมูลภาพ	1
1.3 โครงร่างของปริญญานิพนธ์	1
บทที่ 2 ทฤษฎีการสแกนและสัญญาณภาพ	3
2.1 องค์ประกอบภาพ	3
2.2 การสแกน	5
2.3 การหักเหลำอเล็กตรอน	8
2.4 สัญญาณภาพ	9
2.4.1 สัญญาณภาพขาว - ดำ	9
2.4.2 สัญญาณแบบลงกิ่ง	10
2.4.3 สัญญาณเชิงสี	10
2.4.4 สัญญาณเอ็ควอไลซิ่ง	12
บทที่ 3 การแปลงสัญญาณภาพเป็นข้อมูลทางดิจิทัลและการอินเทอร์เฟซ	14
3.1 การแปลงสัญญาณอนาลอกเป็นดิจิทัล	14
3.2 การแปลงสัญญาณดิจิทัลเป็นอนาลอก	15
3.2.1 วงจร D/A แบบใช้ความต้านทานหลายค่า	15
3.2.2 แบบใช้ไอซี	16
3.3 การอินเทอร์เฟซแบบขนาน	18
บทที่ 4 การสร้างและรายละเอียดของวงจรส่วนต่างๆ	20
4.1 วงจรและการทำงานของภาคอินพุท	22
4.2 วงจรและการทำงานของภาคเอาท์พุท	24
4.3 วงจรและการทำงานของภาคผลิตความถี่	24
4.4 วงจรและการทำงานของภาคแยกเชิงสี	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

4.5	วงจรและการทำงานของภาควงจรม้วน	29
4.6	วงจรและการทำงานของภาคเลือกแอดเดรส	31
4.7	วงจรและการทำงานของภาคควบคุมการเขียนข้อมูลภาพ	33
4.8	วงจรและการทำงานของภาคหน่วยความจำภาพ	36
4.9	วงจรและการทำงานของภาคควบคุมและการอินเตอร์เฟส	39
4.10	การเขียนโปรแกรม	41
บทที่ 5	การทดลองและผลการทดลอง	45
5.1	การทดลองภาค A/D และ D/A	45
5.2	การทดลองภาคควบคุมการเขียนข้อมูลภาพ	46
5.3	การทดลองภาคแยกซิงค์	48
5.4	การทดสอบโดยการป้อนสัญญาณจาก Pattern Generator	50
5.5	การทดสอบโดยการป้อนสัญญาณจากกล้อง ซีซีดี	52
บทที่ 6	สรุปและวิจารณ์	55
ภาคผนวก		
	ภาคผนวก ก. โปรแกรมภาษาซี	
	ภาคผนวก ข. โปรแกรมแอสเซมบลี 89C51	
บรรณานุกรม		

สารบัญรูป

	หน้า
บทที่ 2 ทฤษฎีการสแกนภาพและสัญญาณภาพ	3
รูปที่ 2.1 องค์ประกอบภาพที่เอาเฉพาะพิกเซลไปใช้งาน	4
รูปที่ 2.2 การเอาพิกเซลไปเก็บ ในหน่วยความจำระบบดิจิทัล	4
รูปที่ 2.3 แสดงวิธีการเบื้องต้นของการสแกนหน้า	5
รูปที่ 2.4 รายละเอียดของการสแกนแบบสลับเส้นหรือแบบสอดแทรก	6
รูปที่ 2.5 การสแกนแบบสลับเส้นหรือแบบสอดแทรกในระบบ CCIR	7
รูปที่ 2.6 การหักของลำอิเล็กตรอนเมื่ออยู่ในสนามไฟฟ้าและสนามแม่เหล็ก	9
รูปที่ 2.7 แสดงระดับสัญญาณขาว-ดำ โดยเกรย์สเกล	10
รูปที่ 2.8 ลักษณะของสัญญาณรวมเบื้องต้น	11
รูปที่ 2.9 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง	12
รูปที่ 2.10 ระบบพัลส์ในฟิลชต์ของการซิงโครไนซ์ของระบบโทรทัศน์	13
บทที่ 3 การแปลงสัญญาณภาพเป็นข้อมูลทางดิจิทัลและอินเทอร์เฟส	14
รูปที่ 3.1 วงจรพื้นฐานของการแปลงสัญญาณอนาลอกเป็นดิจิทัล	15
รูปที่ 3.2 วงจร D/A แบบใช้ความต้านทานหลายค่า	16
รูปที่ 3.3 แสดงโครงสร้างและการต่อใช้งานของ D/A เบอร์ MC 1480	17
รูปที่ 3.4 แสดงรูปร่างของ 89C51	18
รูปที่ 3.5 แสดงวงจรของพรีนเตอร์พอร์ท	19
บทที่ 4 การสร้างและรายละเอียดส่วนต่างๆ	20
รูปที่ 4.1 แสดงบล็อกไดอะแกรมของวงจรต่างๆ	21
รูปที่ 4.2 แสดงวงจรภาคอินพุท	23
รูปที่ 4.3 แสดงวงจรภาคเอาท์พุท	25
รูปที่ 4.4 แสดงวงจรผลัดความถี่	26
รูปที่ 4.5 แสดงวงจรภาคแยกซิงค์	28
รูปที่ 4.6 แสดงวงจรภาควงจรนับ	30
รูปที่ 4.7 แสดงวงจรภาคเลือกแอสแตเรส	32
รูปที่ 4.8 แสดงวงจรภาคควบคุมการเขียนข้อมูลภาพ	35
รูปที่ 4.9 แสดงวงจรภาคหน่วยความจำภาพ	38
รูปที่ 4.10 แสดงวงจรภาคควบคุมและการอินเทอร์เฟส	40

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.11 แสดงโพลวชาร์ตของโปรแกรมแอสเซมบลี 89C51 ที่ควบคุมระบบ	43
รูปที่ 4.12 แสดงโพลวชาร์ตของโปรแกรมภาษาซี	44
บทที่ 5 การทดลองและผลการทดลอง	45
รูปที่ 5.1 แสดงรูปสัญญาณอินพุท (CH2) ที่ป้อนกับสัญญาณแอ้าท์พุท (CH1) จากภาค D/A	45
รูปที่ 5.2 แสดงสัญญาณที่ภาคควบคุมการเขียนข้อมูลภาพ	47
รูปที่ 5.3 สัญญาณทดสอบที่ภาคแยกซิงค์	49
รูปที่ 5.4 แสดงรูปที่ได้จากจอคอมพิวเตอร์โดยการป้อนสัญญาณจาก Pettern Generator	50
รูปที่ 5.5 แสดงรูปที่ได้จากจอคอมพิวเตอร์โดยการป้อนสัญญาณจากPettern Geneerator	51
รูปที่ 5.6 แสดงรูปที่ได้จากจอคอมพิวเตอร์โดยการป้อนสัญญาณจากกล้องซีซีดี	52
รูปที่ 5.7 แสดงรูปที่ได้จากจอคอมพิวเตอร์โดยการป้อนสัญญาณจากกล้องซีซีดี	53
รูปที่ 5.8 แสดงตัวเครื่องด้านหน้าและด้านหลัง	54

สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงค่าของเข้าที่พหุเปลี่ยนแปลงไปตามช่วงแรงคั่นอนาลอกอินพุท	15
ตารางที่ 4.1 แสดงสัญญาณควบคุมในการทำงาน	39



บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

โครงการนี้เป็นการปรับปรุงโครงการเก่าซึ่งระบบเก็บข้อมูลลายนิ้วมือจากกล้อง CCD [3] ในส่วนของการคัดเปลี่ยนข้อมูลเป็นดิจิทัล เนื่องจากโครงการเก่าได้ใช้การอินเทอร์เฟสผ่าน Slot ของเครื่องคอมพิวเตอร์โดยตรง การทำงานต่างๆจึงถูกควบคุมโดยไมโครคอมพิวเตอร์ โครงการนี้จึงได้ออกแบบให้การทำงานของการ์ดเป็นอิสระจากไมโครคอมพิวเตอร์ และใช้การถ่ายโอนข้อมูลตามพอร์ตมาตรฐานที่มีอยู่ในเครื่องคอมพิวเตอร์โดยทั่วไป ในการออกแบบเบื้องต้นของโครงการจะใช้พอร์ตเซ็นโทรนิกส์ในการอินเทอร์เฟสระหว่างอุปกรณ์ภายนอกกับคอมพิวเตอร์ เนื่องจากคอมพิวเตอร์ทุกเครื่องก็จะมีพอร์ตเซ็นโทรนิกส์ให้ใช้สำหรับการรับส่งข้อมูล ซึ่งจะมีความสะดวกในการใช้งานยิ่งขึ้น

โครงการที่จัดทำขึ้นนี้ใช้ MCS-51 เป็นตัวควบคุมการทำงานของระบบทั้งหมดแทนเครื่องคอมพิวเตอร์โดยการตั้งโปรแกรมการทำงานไว้ ในการตรวจสอบการวางนิ้วมือให้กล้องทำงานเพื่อทำการสแกนลายนิ้วมือ อย่างไรก็ตามก็มีการเขียนข้อมูลภาพลงหน่วยความจำจะไม่ใช้ MCS-51 เนื่องจากความเร็วของการทำงานของ MCS-51 ไม่เพียงพอกับการ สุ่ม(Sampling) ข้อมูลของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล (Analog to Digital Convertor) ที่ความถี่ 10 MHz ซึ่งอาจทำให้ข้อมูลที่ได้อาจมีความผิดพลาดได้ แต่จะใช้วงจรภายนอกควบคุมการเขียนข้อมูลภายใต้การควบคุมของ MCS-51 นอกจากนี้จะใช้ MCS-51 ในการอ่านข้อมูลจากหน่วยความจำและส่งข้อมูลให้กับคอมพิวเตอร์เพื่อแสดงผล ซึ่งขณะนี้ไมโครคอมพิวเตอร์จะทำหน้าที่เพียงการรับข้อมูลภาพและแสดงผลออกทางหน้าจอโดยการเขียนโปรแกรมควบคุม รายงานฉบับนี้จะนำเสนอรายละเอียดในการออกแบบตามแนวความคิดเบื้องต้นดังกล่าวมานี้

1.2 หลักการเก็บข้อมูลภาพ

หลักการพื้นฐานในการเก็บข้อมูลเนื่องจากสัญญาณภาพทั่วไปประกอบไปด้วยพิกเซลคู่และพิกเซลคู่ซึ่งใช้หน่วยความจำขนาด 128 กิโลไบต์ จำนวน 2 ตัวในการเก็บข้อมูลพิกเซลคู่และพิกเซลคู่ ข้อมูลที่จัดเก็บมีขนาด 512×512 จุดภาพ ดังนั้นจึงจำเป็นต้องใช้ความถี่ในการสุ่มขนาด 10 MHz

1.3 โครงร่างของปริิณญาณิพนธ์

ปริิณญาณิพนธ์นี้เป็นการรวบรวมข้อมูลที่ผู้เขียนได้ศึกษาและทดลองตลอดระยะเวลา 2 ภาคเรียน โดยแบ่งเนื้อหาออกเป็นส่วนๆดังนี้

บทที่ 1 เป็นบทนำจะกล่าวถึงความจำเป็นมาของโครงการและ โครงร่างของปริิณญาณิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 เป็นการกล่าวถึงทฤษฎีการสแกนภาพเบื้องต้นและรายละเอียดของสัญญาณภาพ ในส่วนขององค์ประกอบภาพ ซึ่งมีความจำเป็นต้องมีความเข้าใจเป็นพื้นฐานก่อน

บทที่ 3 เป็นการกล่าวถึงพื้นฐานการแปลงสัญญาณภาพเป็นข้อมูลทางดิจิทัล แบบขนาน หรืออาจเรียกอีกอย่างได้ว่าแบบ “แฟลช” เนื่องจากการทำงานที่เร็วนั่นเอง ซึ่งในโครงการนี้จะใช้ไอซีสำเร็จรูปเบอร์ TDA8708A การทำงานและวงจรการใช้งานของไอซีเบอร์นี้จะได้กล่าวไว้ในบทที่ 4

บทที่ 4 เป็นการกล่าวถึงการสร้างของโครงการ รายละเอียดของการทำงานของวงจรในส่วนต่างๆ

บทที่ 5 เป็นการทดลองวัดรูปคลื่นของสัญญาณ ในส่วนของภาคอินพุท ภาคเอาท์พุท ภาคแยกสัญญาณซิงค์ ซึ่งสัญญาณซิงค์ต้องใช้ในการควบคุมการทำงานของวงจรมับ ภาคควบคุมการเขียนข้อมูลภาพ และผลที่ได้จากการแสดงข้อมูลที่มอนิเตอร์และคอมพิวเตอร์

บทที่ 6 เป็นการกล่าวสรุปตลอดจนปัญหาต่างๆ ในการดำเนินงาน



บทที่ 2

ทฤษฎีการสแกนภาพและสัญญาณภาพ

2.1 องค์ประกอบภาพ

ในภาพหนึ่งภาพประกอบไปด้วยจุดขาวและจุดดำเป็นจำนวนมากมาเรียงกันประกอบขึ้นเป็นภาพ ซึ่งจุดเหล่านี้เรียกว่าองค์ประกอบภาพหรือพิกเจอร์ อีลีเมนต์ (Picture Element) หรือพิกเซล (Pixel)

ภาพที่เกิดขึ้นที่จอโทรทัศน์ประกอบด้วยเส้นขวางเล็กๆในแนวนอนจำนวนมาก แต่ละเส้นมีทั้งส่วนที่ดำและส่วนที่ขาว เส้นเหล่านี้ได้จากการกวาดลำแสง (Scan)

ระบบการสแกน 525 เส้นเราจะแบ่งส่วนทางแนวดิ่ง 700 เส้น ดังนั้นถ้าจะหาจำนวนองค์ประกอบภาพในระบบการสแกน 525 เส้นได้เท่ากับ 525×700 เท่ากับ 367,500 พิกเซล ภาพที่มีจำนวนเส้นสแกนมากจะมีรายละเอียดภาพมาก โทรทัศน์ที่มีจำนวนเส้นสแกนมากจะได้รายละเอียดหรือความชัดเจนของภาพมาก แต่การออกแบบวงจรจะยากตามไปด้วยเนื่องจากแบนด์วิดธ์ของความถี่จะต้องกว้างด้วย [1]

$$f_{\max} = \frac{1}{2} K n^2 f_p (b/h)(Y/X) \quad (2.1)$$

เมื่อ f_{\max} คือความถี่สูงสุด

K คือ ค่าคงที่ประมาณ 0.64 ถึง 0.7

n คือ จำนวนเส้นสแกน

f_p คือ จำนวนภาพต่อวินาที

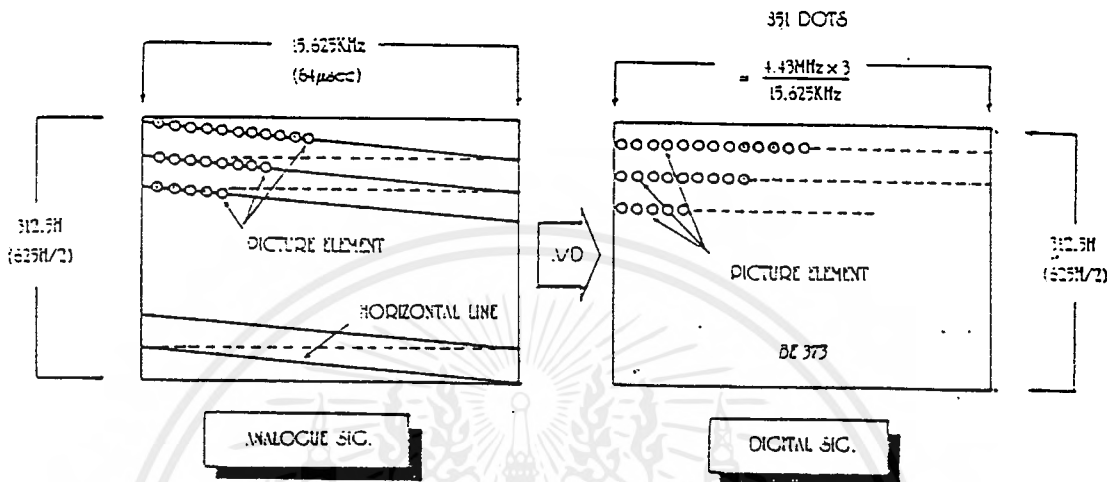
b/h คือ อัตราส่วนแอสเป็คท์ซึ่งทั่วไปใช้ 4 ต่อ 3

X/Y คือ ค่าออฟเพ็คทิฟแฟกเตอร์ มีค่า 0.95/0.84

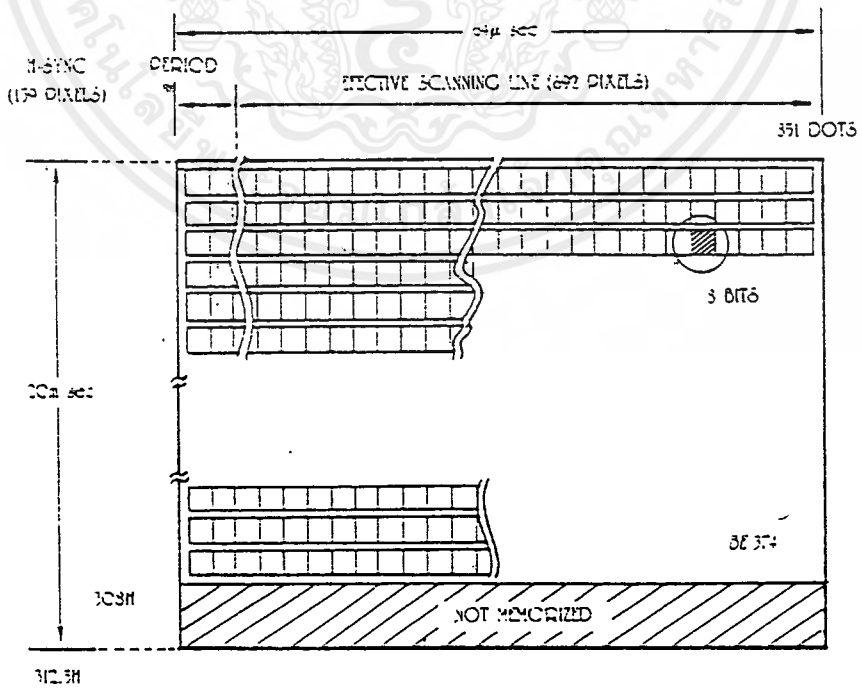
จากสมการจะเห็นว่าเมื่อเส้นสแกนมากขึ้น ความถี่จะเพิ่มตามด้วย จากที่กล่าวไว้ว่าระบบ 525 เส้นมีองค์ประกอบภาพ 367,500 พิกเซลแต่ความเป็นจริงเรามองไม่เห็นครบทุกพิกเซล เนื่องจากมีบางส่วนหายไประหว่างที่มีการสแกน ภาพที่สามารถดูได้จะมีองค์ประกอบภาพไม่ต่ำกว่า 200,000 พิกเซล

ระบบที่ใช้ในประเทศไทยใช้ระบบสแกน 625 เส้น 25 ภาพต่อวินาทีซึ่งมีองค์ประกอบภาพเท่ากับ 851×625 ได้เท่ากับ 531,875 ปัจจุบันระบบดิจิทัลได้เข้าไปมีบทบาทเป็นอย่างมากในระบบของโทรทัศน์ ซึ่งการเก็บภาพก็จะทำแบบดิจิทัลโดยการนำพิกเซลต่างๆ ไปเก็บไว้ในหน่วยความจำเพราะว่าพิกเซลเท่านั้นที่ระบบดิจิทัลจะจัดการข้อมูลได้

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้ในวงการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 องค์ประกอบภาพที่เอาเฉพาะพิกเซลไปใช้งาน



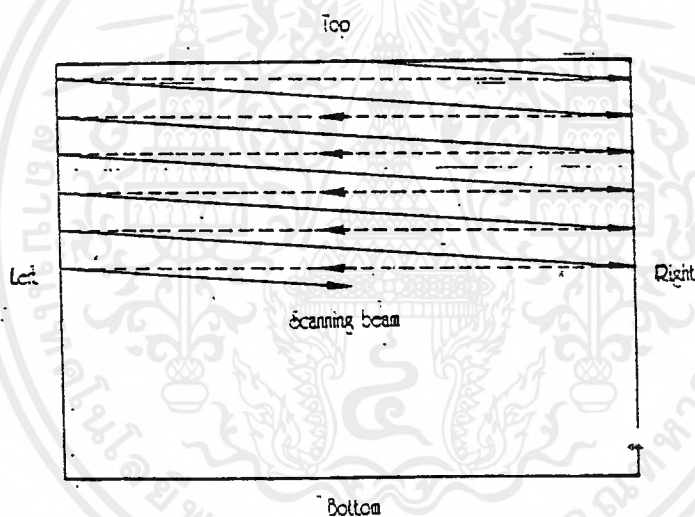
รูปที่ 2.2 การเอาพิกเซลไปเก็บในหน่วยความจำระบบดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่ออนุญาตให้ผู้อื่นใช้โดยไม่ได้รับอนุญาตเป็นการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การสแกน

จากองค์ประกอบภาพที่ได้กล่าวมาแล้ว เครื่องส่งจะรับภาพแล้วเปลี่ยนพลังแสงเป็นพลังงานไฟฟ้าส่งเป็นจุดขาวดำมา เครื่องรับจะรับเอาสัญญาณเหล่านี้มาเรียงกันใหม่ โดยการกวาดเป็นเส้นภาพขึ้นที่หนึ่ง จอวิธีการนี้เราเรียกว่าการสแกน (Scanning) ตัวที่มีหน้าที่สำคัญคือจอภาพหรือหลอดภาพ หลอดภาพมีโครงสร้างคล้ายกับหลอดสุญญากาศที่ปล่อยอิเล็กตรอนออกมาจากแคโทดแล้วจะมีการดึงลำอิเล็กตรอนให้วิ่งไปเป็นลำกระทบเข้ากับอนิโอดหรือหน้าจอ ตรงหน้าจอนั้นมีการฉาบสารเรืองแสงไว้เพื่อให้เกิดการเรืองแสงของจอ

การสแกนมี 2 วิธี คือการสแกนแบบก้าวหน้า (Progressive Scanning) กับการสแกนแบบสลับเส้น (Interlace Scanning)



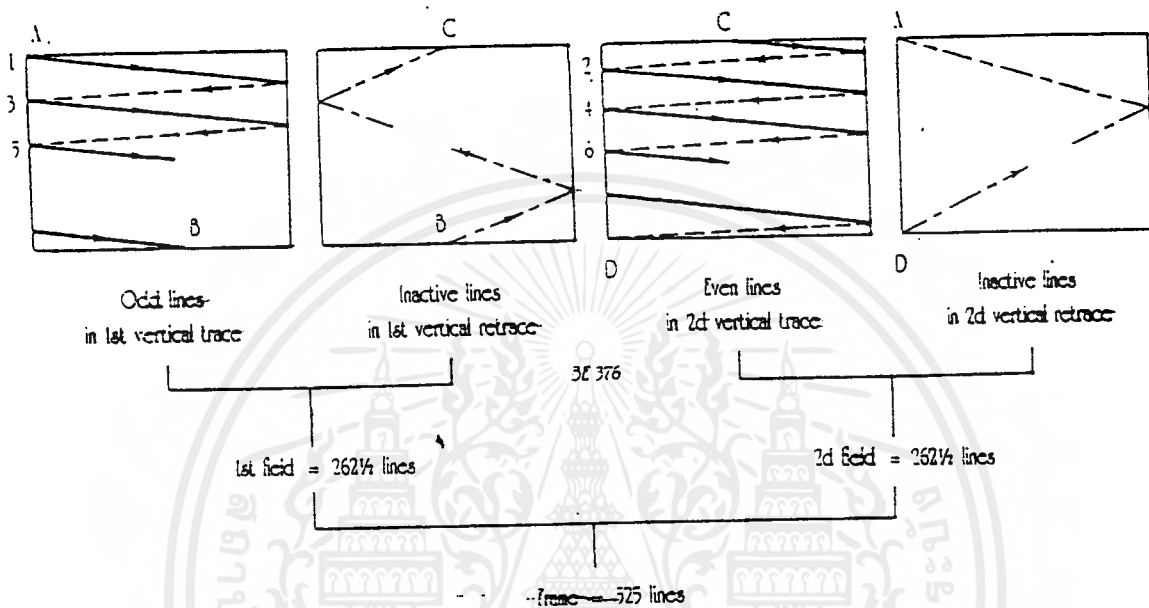
รูปที่ 2.3 แสดงวิธีการเบื้องต้นของการสแกน

การที่จะทำให้การสแกนมีความต่อเนื่องขององค์ประกอบภาพดังที่กล่าวมาแล้ว จะต้องคำนึงถึงหลัก 3 ประการคือ

2.2.1 ลำอิเล็กตรอนที่กวาดไปทางแนวนอน (Horizontal Scanning) ในแต่ละครั้ง จะต้องสามารถครอบคลุมองค์ประกอบภาพทั้งหมดของเส้นนั้นๆ

2.2.2 ในแต่ละเส้นของการสแกนลำอิเล็กตรอน ลำแสงต้องกวาดกลับด้วยความเร็วสูงไปยังด้านซ้ายเพื่อเริ่มต้นเส้นภาพทางแนวนอนลำดับต่อไป เวลาของการสับกลับเรียกว่า “รีเทรซ” (Retrace) หรือ ฟลายแบ็ค (Flyback) ในกรณีดังกล่าวจะต้องไม่มีข้อมูลภาพใดๆ เพราะว่่าทั้งกล้องถ่ายและหลอดภาพจะเกิดการแบล็กเอาท์ (Blank Out) ในขณะนั้น

2.2.3 ในขณะที่เส้นสแกนสลับกลับมาเพื่อเริ่มต้นทางซ้ายใหม่ ตำแหน่งทางแนวตั้งต้องต่ำกว่าตำแหน่งเดิม เพื่อให้การสแกนเส้นต่อไปไม่ทับกัน ทั้งนี้โดยการควบคุมของสัญญาณทางแนวตั้ง (Vertical Scanning)



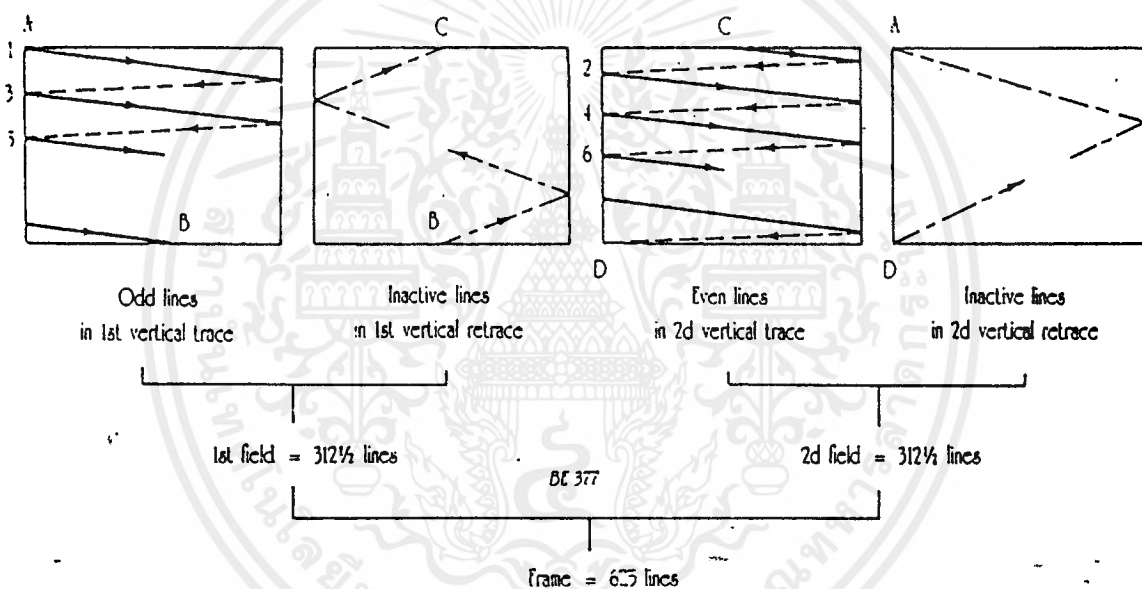
รูปที่ 2.4 รายละเอียดของการสแกนแบบสลับเส้นหรือแบบสอดแทรก

การสแกนที่ใช้ในเครื่องรับโทรทัศน์ถึงแม้เราจะพบว่าหากให้มีการเรียงภาพเกินกว่า 16 ภาพต่อวินาทีแล้ว สายตาจะเห็นเป็นภาพที่ต่อเนื่อง จากการทดลองสแกนภาพพบว่าแม้ภาพที่เกิดขึ้นจะเกิดขึ้น 24 ภาพต่อวินาทีแล้วก็ตาม ยังมีการกระพริบเกิดขึ้น เนื่องจากในขณะที่การสแกนเริ่มจากขอบบนลงมาด้านล่าง เมื่อเส้นสแกนลงมาถึงขอบด้านล่างแสงทางด้านบนในความรู้สึกของมนุษย์เริ่มมีลดลงกว่าด้านล่าง เวลาที่ลำแสงการสแกนวกกลับไปด้านบนด้านล่างก็เกิดปัญหาเช่นเดียวกัน ความรู้สึกต่อกรณีนี้ก็คือเกิดแสงกระพริบหรือวูบวาบขึ้นจึงต้องใช้การสแกนแบบสลับเส้นหรือบางคนเรียกว่าการสแกนแบบสอดแทรก (Interlace Scanning) โดยครั้งแรกจะสแกนฟิลด์คี่ (Odd line Trace) และครั้งต่อไปจะสแกนฟิลด์คู่ (Even line Trace) เป็นการสแกนแบบเส้นเว้นเส้น นั้นหมายความว่า การได้ภาพ 1 ภาพหรือที่เรียกว่าภาพ 1 เฟรม (Frame) ต้องใช้การสแกนแนวตั้งถึง 2 ครั้ง หรือ 2 ฟิลด์(Field)

ตามมาตรฐาน เอฟซีซี (FCC) ใช้เส้นสแกน 525 เส้นต่อ 1 ภาพ และใช้ภาพ 30 ภาพต่อวินาที ดังนั้นใน 1 ฟิลด์จะมีเส้นสแกน 262.5 เส้น ภาพหนึ่งแต่ละภาพซึ่งเป็นส่วนขององค์ประกอบภาพจะเกิดขึ้นภายใน 1/30 วินาที ความถี่ที่ใช้เพื่อหักเหลำอิเล็กตรอนในแนวนอนจึงได้จาก

จำนวนเส้นภาพคูณกับจำนวนภาพในแต่ละเฟรม จึงได้ 525×30 เท่ากับ 15,750 HZ ส่วนความถี่
 บังคับการหักเหในแนวตั้งใน 1 ฟิลด์จะใช้เวลาเพียง 1/60 วินาทีเท่านั้นความถี่จึงเท่ากับ 60 HZ

มาตรฐานซีซีไออาร์ (CCIR) ใช้เส้นสแกน 625 เส้นต่อหนึ่งภาพ และใช้ภาพ 25 ภาพต่อ
 วินาที ดังนั้นใน 1 ฟิลด์จะมีเส้นสแกน 312.5 เส้น ภาพหนึ่งแต่ละภาพซึ่งเป็นองค์ประกอบภาพจะ
 เกิดขึ้นภายใน 1/25 วินาที ความถี่ที่ใช้เพื่อการหักกล้อเล็กทรอนิกส์ในแนวนอนจึงได้จากจำนวนเส้น
 ภาพ 625 เส้นคูณกับจำนวนภาพในแต่ละเฟรม ดังนั้นเราจึงสามารถหาความถี่ได้จาก 625×25 เท่า
 กับ 15,625 HZ ความถี่หักเหทางแนวตั้งจึงเท่ากับ 50 HZ



รูปที่ 2.5 การสแกนแบบสลับเส้นหรือแบบสอดแทรกในระบบ CCIR

จากรูปที่ 2.4 และ 2.5 ได้แสดงวิธีการของการสแกนแบบสอดแทรกของระบบโทรทัศน์ทั้งสองระบบใหญ่ที่ใช้กันอยู่ในปัจจุบัน โดยเมื่อเริ่มดำเนินการสแกนสมมุติว่าการเริ่มสแกนในกรณีนี้เริ่มจากการสแกนจากเฟรมที่เป็นเส้นสแกนที่ โดยเริ่มจาก A ซึ่งอยู่ทางด้านซ้ายแล้วกวาดไปทางขวานับเป็นเส้นสแกนเส้นที่ 1 แล้วจึงสแกนเส้นที่ 3, 5, 7, 9 และต่อไปจนกระทั่งได้เส้นสแกน 262.5 เส้นในระบบ FCC หรือ 312.5 เส้นในระบบ CCIR ซึ่งก็คือสแกนมาถึงจุด B ดังในรูปที่ 2.5 และ 2.6 ณ จุดนี้การสแกนจะถูกความถี่หักเหทางแนวตั้งซึ่งเรียกว่าเวอร์ติคอลล รีเทรซ (Vertical Retrace) หรือสัญญาณฟลายแบ็ค (Flyback) ดึงกลับไปยังตำแหน่งในจุด C เพื่อเริ่มต้นการสแกนเส้นคู่ต่อไป

เวลาของการรีเทรซทั้งการรีเทรซทางเวอร์ติคอลลและฮอร์ริซอนตอลลเป็นเวลาสั้นๆถึงอย่างไรก็ตามเราไม่ต้องการให้เส้นสแกนที่เป็นช่วงของการสลับกลับ นี้เข้ามารบกวนการทำให้เกิดสัญญาณไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลบางประการและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ ในส่วนนี้จึงต้องทำการลบเส้นสับคกลับ ก่อนที่จะถึงจุดที่ว่าน้ำที่เราลองมาดูรายละเอียดของ การสับคกลับว่าใน ส่วนของการกวาดลำแสงหรือการกวาดลำแสงในทางแนวนอน เวลาของการ รีเทรชจะได้ประมาณ 10 - 16 เปอร์เซ็นต์ของเวลาทั้งหมด ในระบบ FCC เราใช้เวลาในการสแกน ทางแนวนอนเท่ากับ 63.5 μ S ดังนั้นเวลาของการรีเทรชเท่ากับ 6.35 μ S ส่วนระบบ CCIR เราใช้ เวลาในการสแกนเท่ากับ 64 μ S ดังนั้นเวลาของการรีเทรชจะได้ประมาณ 6.4 μ S ในส่วนของ ทางด้านความถี่หักเหทางแนวตั้งเราใช้เวลาในการรีเทรชไม่เกิน

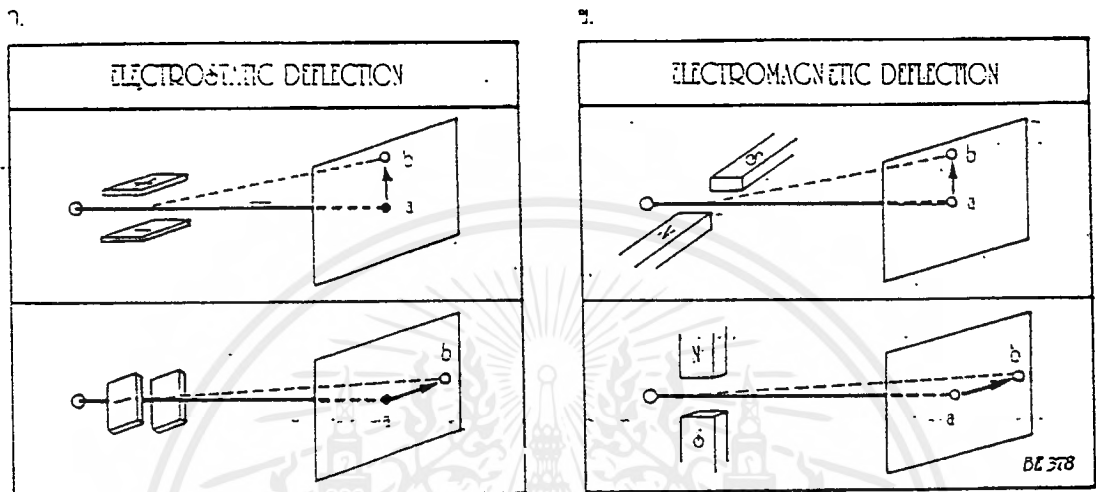
5 - 8 เปอร์เซ็นต์ อย่างเช่นเราใช้เวลาในส่วนนี้เท่ากับ 3 เปอร์เซ็นต์ของเวลา 1/50 หรือ 1/60 ของวินาที ดังนั้นเวลาของการรีเทรชจึงเท่ากับ 600 μ S และ 500 μ S ตามลำดับ หมายความว่าใน ช่วงของการรีเทรชทางแนวตั้งกินเวลานานกว่าการสแกนทางแนวนอนประมาณ 8 - 10 เส้นภาพ

จากหลักการดังกล่าวสามารถสรุปกฎเกณฑ์บางอย่างได้ว่า ตามความเป็นจริงแล้วในเส้นภาพ 525 เส้นหรือ 625 เส้น เราไม่อาจจะเห็นได้ครบทุกเส้นอย่างน้อยๆในกรณีที่เกิดเวรติคอลลรีเราจะ กินเวลาของการสแกนทางแนวนอน ไปด้วย แต่จะกิน ไปก็เส้นนั้นขึ้นอยู่กับสัญญาณบังคับฟลายแบ็ค ซึ่งในตัวรับเรียกสัญญาณนี้ว่าสัญญาณแบลลจิง

จากหลักการดังกล่าวนี้โทรทัศน์ระบบดิจิตอลจึงเก็บเส้นสัญญาณภาพเข้าสู่หน่วยความจำของ ระบบเพียงครั้งละ 308 เส้นเท่านั้น เพราะจากมาตรฐานโทรทัศน์ในปัจจุบันที่กำหนดมาตรฐาน สัญญาณในกรณีรีเทรชทางแนวตั้งเท่ากับ 3 เปอร์เซ็นต์จึงได้เส้นภาพประมาณ 625 - 10 (615 เส้น ภาพ) เมื่อแบ่งเฟรมแล้วจะได้ค่าเส้นภาพ 308 เส้นภาพ

2.3 การหักเหลำอิเล็กตรอน

เมื่อก้าวถึงการสแกนเพื่อเปลี่ยนสัญญาณภาพให้เป็นสัญญาณไฟฟ้าในเครื่องส่ง หรือ สแกนเพื่อเปลี่ยนสัญญาณไฟฟ้าให้เป็นสัญญาณภาพบนจอของหลอดภาพในเครื่องรับโทรทัศน์ ล้วน ต้องใช้ลำอิเล็กตรอนเป็นตัวสำคัญ โดยเฉพาะในเครื่องรับ ภาคแสดงผลของสัญญาณภาพคือหลอด ภาพ ซึ่งโครงสร้างของหลอดภาพเบื้องต้นอาศัยการยิงลำอิเล็กตรอนจากปืนอิเล็กตรอน โดยมีไฟ แรงสูงเป็นตัวคอยดึงให้ลำอิเล็กตรอนวิ่งเข้าชนสารเรืองแสงที่หน้าจอ กรณีเช่นนี้จะทำให้เกิดแสง เพียงจุดเดียวกลางจอ จะทำให้ได้เส้นภาพอย่างที่กล่าวมาแล้วสามารถกระทำได้ โดยการเบี่ยงเบน หรือหักเหลำอิเล็กตรอนให้เกิดการกวาดตามหลักการที่กล่าวมา ลักษณะของลำอิเล็กตรอนเมื่อวิ่ง เข้าไปยังสนามไฟฟ้าสถิตย์หรือสนามแม่เหล็กจะสามารถเปลี่ยนทิศทางได้ดังแสดงไว้ในรูปที่ 2.6



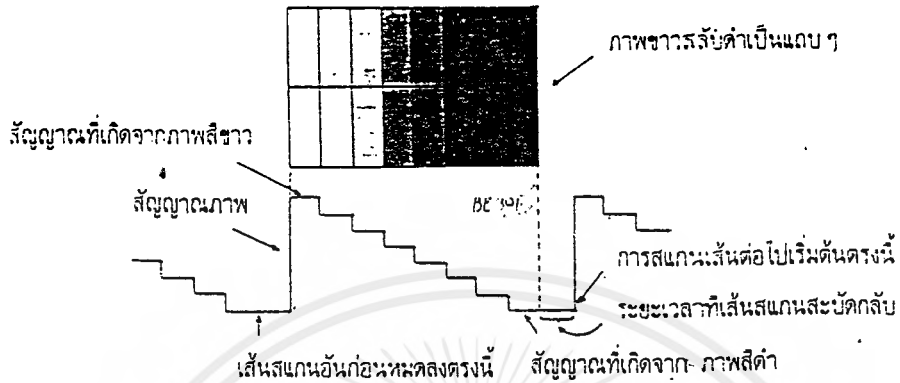
รูปที่ 2.6 การหักของลำอิเล็กตรอนเมื่ออยู่ในสนามไฟฟ้าและสนามแม่เหล็ก

2.4 สัญญาณภาพ

เพื่อความเข้าใจที่ดีในการเก็บภาพทางดิจิทัล เราควรจะทำความเข้าใจเกี่ยวกับรายละเอียดของสัญญาณภาพดังต่อไปนี้คือ

2.4.1 สัญญาณภาพขาว - ดำ

กรณีที่เรากล่าวถึงสัญญาณขาว-ดำ หรือสัญญาณโมโนโครมได้ดีที่สุดต้องกล่าวว่าภาพที่เป็นสีขาวคือภาพที่มีความสว่างมากที่สุด และภาพที่เป็นสีดำคือภาพที่ไม่มีความสว่างเลยภาพจำลองที่ดีที่สุดก็คือแถบภาพที่มีความแตกต่างของระดับขาวดำที่ละน้อย ซึ่งเราเรียกว่าระดับเกรย์สเกล (Gray Scale)



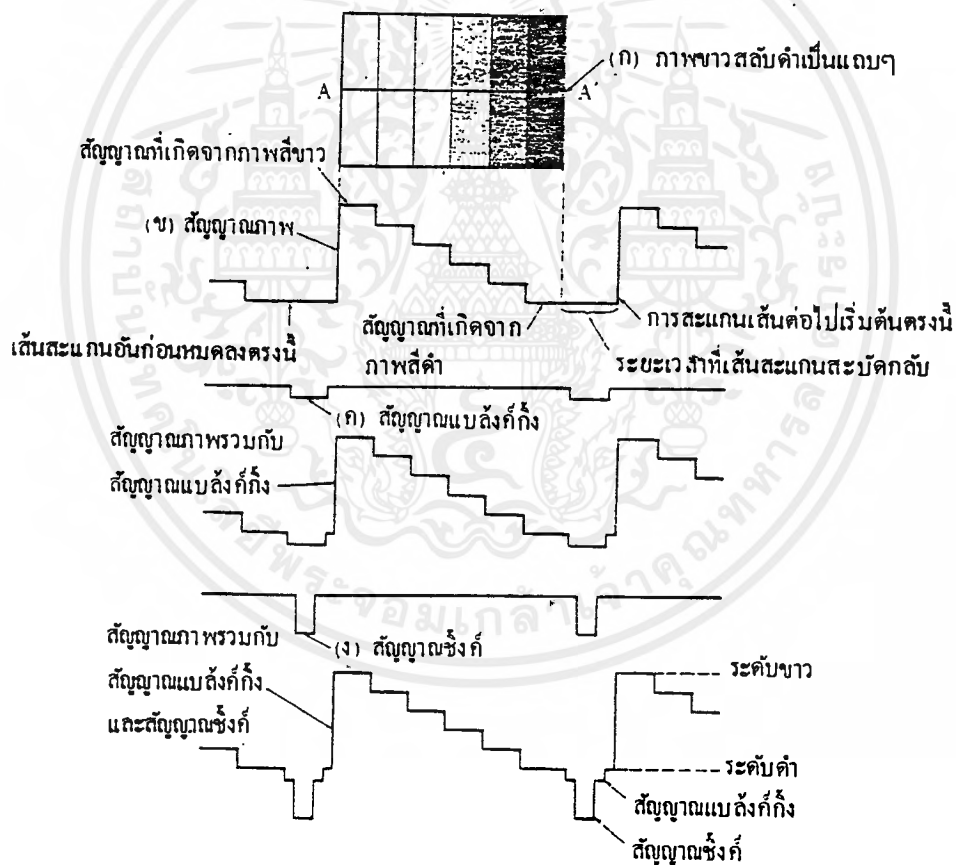
รูปที่ 2.7 แสดงระดับสัญญาณขาว - ดำ โดยเกรย์สเกล

จากภาพที่ 2.7 แสดงให้เห็นถึงระดับของเกรย์สเกลในกรณีที่ระดับของสัญญาณภาพเป็นสัญญาณสีขาวเข้ามา นั่นหมายความว่าระดับความแรงของสัญญาณภาพมากที่สุดจึงให้ความส่องสว่างที่หน้าจอสว่างมากที่สุด และเมื่อระดับสัญญาณสีขาวลดลงเป็นสีม่วง, เทา และ ดำนั้น ระดับสัญญาณจะลดลงมาเรื่อยๆ หมายความว่าเมื่อสัญญาณมีความแรงน้อยลงความส่องสว่างจะน้อยลงไปด้วยจริงๆ แล้วสัญญาณภาพเป็นสัญญาณความถี่ทางไฟฟ้าที่มีความถี่สูงต่ำไม่เท่ากัน โดยความถี่สูงสุดจะไม่เกิน 4 MHz ในระบบ FCC และไม่เกิน 5 MHz ในระบบ CCIR ภาพที่เกิดจากความถี่สูงย่อมมีความละเอียดดีกว่าความถี่ต่ำ

2.4.2 สัญญาณแบลงกิ้ง เป็นสัญญาณที่ใช้เพื่อลบเส้นสแกนสะบัดกลับทั้งในแนวนอนและแนวตั้งเพื่อไม่ให้สังเกตเห็นได้ชัดทางจอหลอดคา สำหรับโทรทัศน์อเมริกัน วงจรของการหักเหทางแนวนอนมีความถี่ 15,750 HZ ฉะนั้นในระยะเวลา 1/15,750 วินาที หรือ 63.5 μ S ที่จะต้องเกิดเส้นสแกนสะบัดกลับครั้งหนึ่งจึงต้องใช้แบลงกิ้งพัลส์ทางแนวนอนหนึ่งครั้ง โดยมีขนาดประมาณ 10 μ S ในทำนองเดียวกันทุกๆ ระยะเวลา 1/60 วินาที หรือ 16.667 μ S ก็ต้องใช้แบลงกิ้งพัลส์ทางแนวตั้งครั้งหนึ่ง โดยมีขนาดประมาณ 1,250 μ S

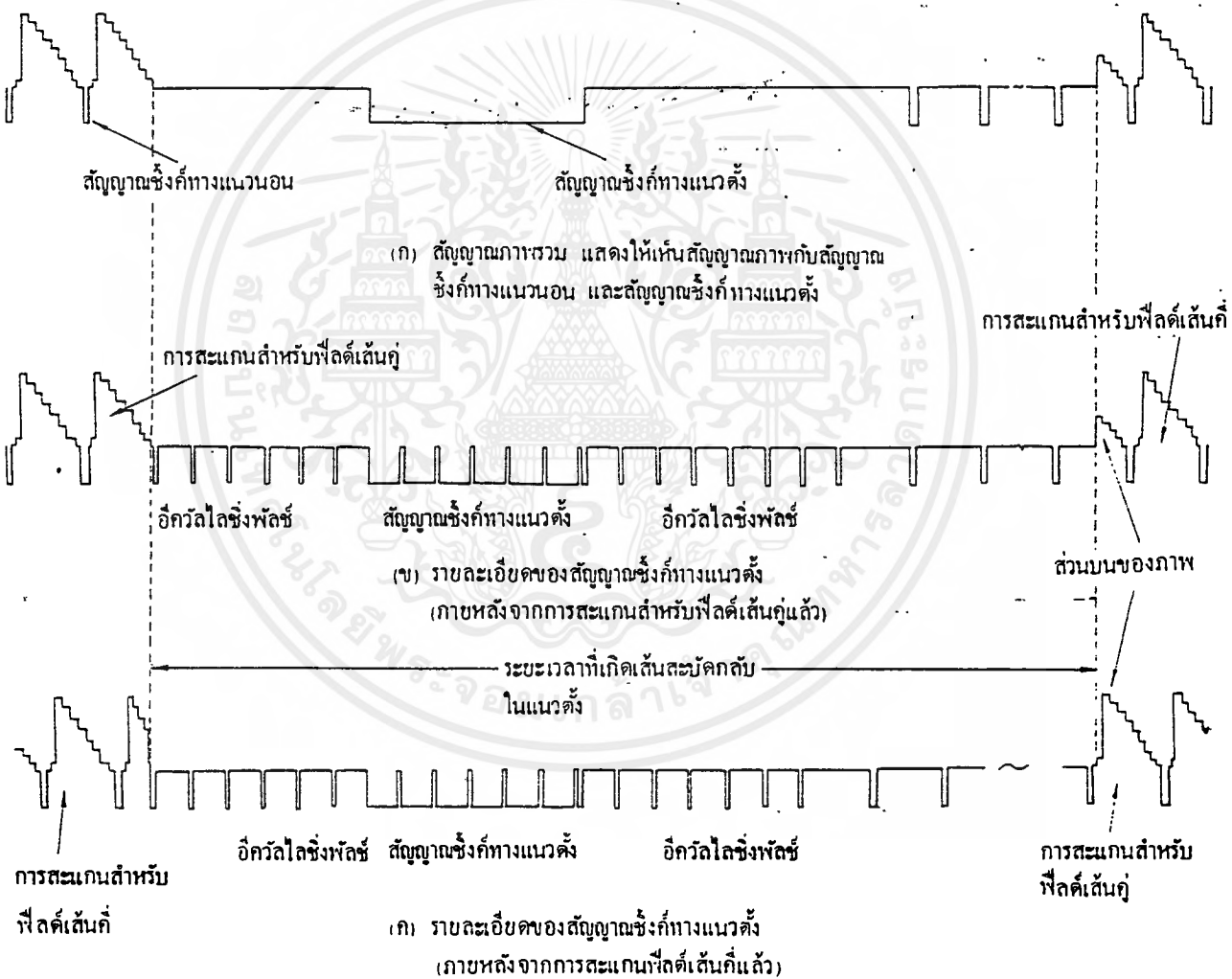
2.4.3 สัญญาณซิงค์ เป็นสัญญาณที่ใช้เพื่อช่วยทำให้วงจรหักเหทางแนวนอน และวงจรหักเหทางแนวตั้งในเครื่องส่งและเครื่องรับโทรทัศน์มีความถี่ตรงกันตลอดเวลา สัญญาณซิงค์ทางแนวนอนมีความถี่ 15,750 HZ ในระบบ FCC และ 15,625 HZ ในระบบ CCIR ซึ่งเท่ากับความถี่ของวงจรหักเหทางแนวนอน และสัญญาณซิงค์ทางแนวตั้งซึ่งมีความถี่ 60 HZ ในระบบ FCC และ 50 HZ ในระบบ CCIR ซึ่งเท่ากับความถี่ของวงจรหักเหทางแนวตั้งเหมือนกัน เนื่องจากว่าความถี่ของสัญญาณซิงค์มีค่าเท่ากับความถี่ของสัญญาณแบลงกิ้งพอดี จึงจำเป็นต้องป้องกันการรบกวนที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจเกิดขึ้น โดยจำเป็นต้องกำหนดขนาดของซิงค์พัลส์ให้น้อยกว่าขนาดของซิงค์พัลส์ กล่าวคือทำให้ซิงค์พัลส์ทางแนวนอนมีขนาดเพียง 5 μ S และซิงค์พัลส์ทางแนวตั้งมีขนาดเพียง 190 μ S นอกจากนี้ยังใช้วิธีส่งซิงค์พัลส์เหล่านี้ปนกับแบลงกิ้งพัลส์ โดยให้ฐานของซิงค์พัลส์อยู่ที่ขอบบนของแบลงกิ้งพัลส์ เมื่อจัดขอบเขตความต่างศักย์ให้ระดับสูงสุดของแบลงกิ้งพัลส์เป็นระดับค่ามีคจนมองไม่เห็นแล้วระดับของซิงค์พัลส์ที่อยู่บนยอดสูงสุดของแบลงกิ้งพัลส์ก็จะป็นระดับค่ามีคสนิท และไม่ทำให้เกิดการรบกวนที่จอหลอดภาพ

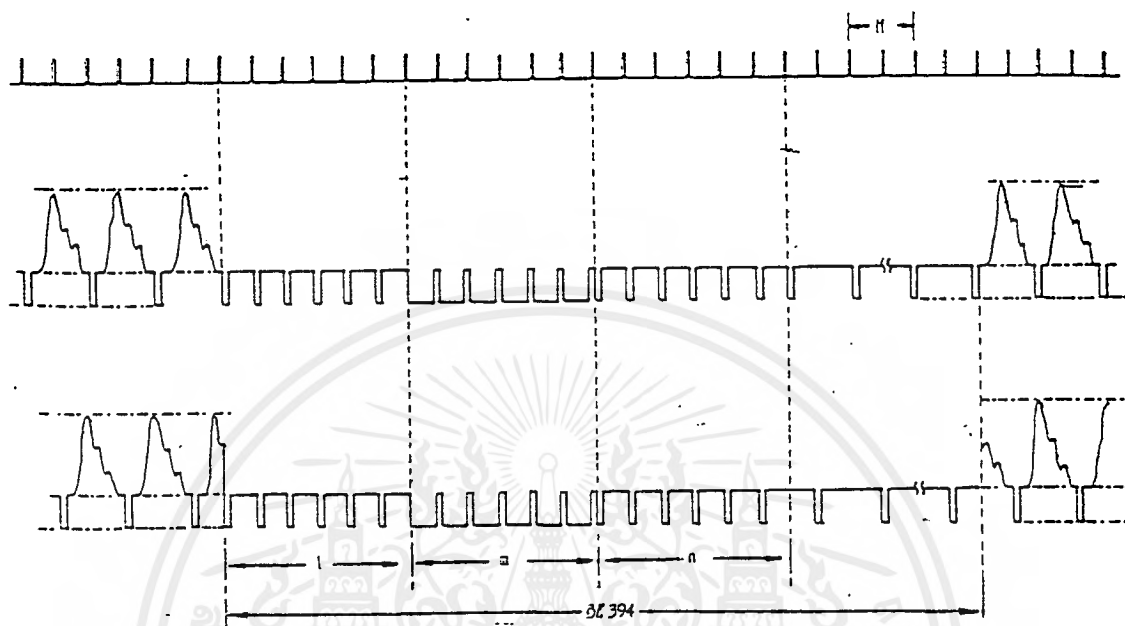


รูปที่ 2.8 ลักษณะของสัญญาณภาพรวมเบื้องต้น

2.4.4 สัญญาณอ็ควอลโลจิ่ง เป็นสัญญาณบังคับรูปร่างของสัญญาณเชิงโครโมโซมทางแนวตั้ง เพื่อให้สามารถคงรูปถูกต้อง แล้วยังช่วยให้การสแกนแบบสลับเส้นเป็นไปได้อย่างถูกต้องด้วย สามารถขึ้นเส้นคู่เส้นคี่ในตำแหน่งที่ต้องการได้ ทั้งยังส่งผลทางอ้อมให้สัญญาณเชิงโครโมโซมทางแนวนอนไม่ขาดช่วงหายไปในช่วงการส่งสัญญาณทางแนวตั้งด้วย สัญญาณตัวนี้จะมีความยาวของพัลส์รวมเท่ากับเวอริคัลของสัญญาณพัลส์ สัญญาณส่วนนี้จะถูกแบ่งเป็น 6 ลูกเล็กๆ ในระบบ 525 เส้น และถูกแบ่งเป็น 5 ลูกในระบบ 625 เส้น ดังแสดงไว้ในรูปที่ 2.9



รูปที่ 2.9 แสดงสัญญาณภาพรวมที่มีรายละเอียดทุกอย่าง



รูปที่ 2.10 ระบบพัลส์ในฟิล์มของการซิงโครไนซ์ของระบบโทรทัศน์

บทที่ 3

การแปลงสัญญาณภาพเป็นข้อมูลทางดิจิทัลและอินเทอร์เฟส

3.1 การแปลงสัญญาณอนาลอกเป็นดิจิทัล (Analog to Digital Converter :A/D)

สัญญาณภาพโดยทั่วไปจะเป็นสัญญาณทางอนาลอก แต่กระบวนการต่างๆที่คอมพิวเตอร์สามารถกระทำได้ต้องอยู่ในรูปของสัญญาณดิจิทัล และการเก็บสัญญาณภาพเข้าสู่หน่วยความจำสามารถทำได้ในระดับของพิกเซลเท่านั้นจึงมีความจำเป็นที่ต้องเปลี่ยนสัญญาณภาพเป็นสัญญาณทางดิจิทัลก่อน และเนื่องจากภาพเป็นสัญญาณความถี่สูงดังนั้น IC ที่ทำหน้าที่แปลงสัญญาณอนาลอกเป็นดิจิทัลต้องมีความเร็วสูงในการทำงาน

การแปลงสัญญาณอนาลอกเป็นดิจิทัล ที่ใช้ในโครงงานนี้เลือกใช้ไอซีเบอร์ TDA 8708A ซึ่งรายละเอียดแสดงไว้ในภาคผนวก ที่มีพื้นฐานมาจากวงจรถ่ายเอชดี แบบใช้วงจรถ่ายแบบขนานหรือแบบ Flash ซึ่งหลักการอย่างง่ายของการแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบขนานแสดงดังรูปที่ 3.1 ใช้ตัวเปรียบเทียบ 3 ตัวต่อในลักษณะ ขนานกัน มีตัวต้านทานต่อแบ่งแรงดันจากแรงดันอ้างอิง ใ้กำหนดแรงดันต่ำสุดที่ตัวเปรียบเทียบทั้ง 3 ตัวยังสามารถทำงานได้ แรงดันอ้างอิงนี้อาจจะมีค่าเท่ากับค่าแรงดันสูงสุดของสัญญาณอินพุตที่เป็นอนาลอกก็ได้

จากตารางที่ 3.1 จะเห็นว่าตัวเปรียบเทียบแต่ละตัวจะให้เอาต์พุตเป็น “1” ก็ต่อเมื่อแรงดันอินพุตมีค่าสูงกว่าแรงดันอ้างอิงของตัวเปรียบเทียบแต่ละตัว ซึ่งมีค่าแตกต่างกัน และถ้าแรงดันอินพุตมีค่าอยู่ในช่วง 3 - 4 โวลต์ (แรงดันอ้างอิง +4 โวลต์) จะทำให้ตัวเปรียบเทียบทั้ง 3 ตัวให้เอาต์พุตเป็น “1” หมด เอาต์พุตของตัวเปรียบเทียบทั้งหมดส่งเข้าไปที่วงจรถ่ายรหัสเพื่อทำให้เป็นสัญญาณดิจิทัลในระบบเลขฐานสองต่อไป

รูปวงจรถ่ายที่ 3.1 นี้ตอบสนองต่อแรงดันอินพุต (อนาลอก) 4 ระดับและแต่ละระดับมีความแตกต่าง 1 โวลต์ ดังนั้นความละเอียด (Resolution) ของวงจรถ่ายนี้มีขนาด 2 บิต เราสามารถหาความละเอียดของวงจรถ่ายได้จากจำนวนของตัวเปรียบเทียบคือ

$$\text{จำนวนตัวเปรียบเทียบ} = 2^n - 1 \quad (3.1)$$

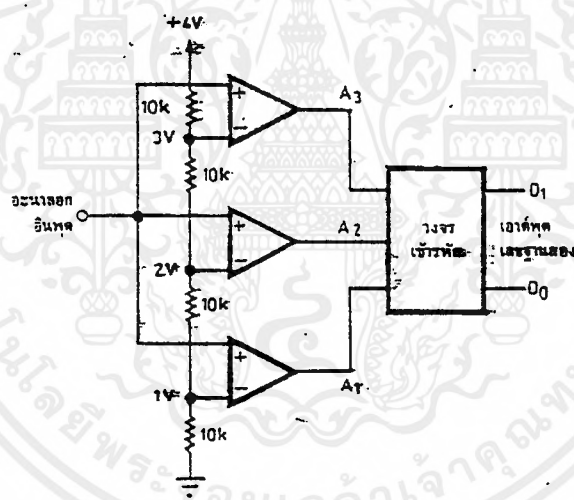
เมื่อ n = ความละเอียด

เช่นต้องการความละเอียดขนาด 8 บิต จะต้องใช้ตัวเปรียบเทียบ 255 ตัว จากลักษณะการต่อตัวเปรียบเทียบให้ขนานกันเพื่อให้รับสัญญาณอินพุตได้พร้อมๆ กันเราจึงเรียกวงจรถ่ายนี้ว่า วงจรถ่ายแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบขนาน (Parallel A/D Converter) และเนื่องจากสามารถตอบสนองต่อสัญญาณอนาลอกเป็นดิจิทัลได้อย่างรวดเร็วมากจึงเรียกได้อีกอย่างว่าวงจรถ่ายแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบ “แฟลช” (Flash A/D Converter)

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนาล็อกอินพุต (โวลต์)	เอาต์พุตตัวเปรียบเทียบ			เอาต์พุต เลขฐานสอง	
	A ₁	A ₂	A ₃	D ₁	D ₀
0-1	0	0	0	0	0
1-2	1	0	0	0	1
2-3	1	1	0	1	0
3-4	1	1	1	1	1

ตารางที่ 3.1 แสดงค่าของเอาต์พุตเปลี่ยนแปลงไปตามช่วงแรงดันอนาล็อกอินพุต



รูปที่ 3.1 วงจรพื้นฐานของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

3.2 การแปลงสัญญาณดิจิทัลเป็นอนาล็อก (Digital to Analog Converter :D/A)

การแปลงสัญญาณอนาล็อกเป็นดิจิทัลมีหลายวิธีดังนี้คือ

3.2.1 วงจร D/A แบบใช้ความต้านทานหลายค่า

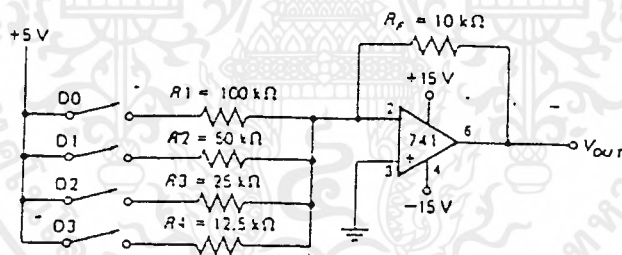
วงจร D/A แบบนี้ใช้ตัวต้านทานและออปแอมป์เพื่อเปลี่ยนระดับลอจิก 2 ระดับเป็นแรงดันที่ได้สัดส่วนกันดังรูปที่ 3.2 แสดงวงจรเปลี่ยนสัญญาณดิจิทัลขนาด 4 บิต ออปแอมป์ที่ใช้มีเกนซ์สูงมาก มีความต้านทานเอาต์พุตต่ำความต้านทานอินพุตสูงมากที่สำคัญคือสัญญาณที่เอาต์พุตถูกป้อนกลับมายังอินพุตแบบไม่กลับเฟสเอาต์พุตของออปแอมป์จะเป็นตัวจ่ายหรือรับกระแสเพื่อให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันที่เปรียบเทียบกับนั้นมีค่าเดียว วงจรในรูปที่ 3.2 ต่อขาไม่กลับเฟสของกราวด์คั้งนั้นที่ขากลับเฟสก็จะมีแรงดัน 0 โวลต์ด้วยโดยไม่ได้ต่อลงกราวด์ตรงเรียกว่า Virtual Ground

เมื่อปิดสวิตช์ D0. ตัวต้านทาน R1 ค่า 100K จะมีแรงดัน 5 โวลต์ที่ปลายข้างหนึ่งและที่ปลายอีกข้างหนึ่งจะเป็น 0 จากกฎของโอห์มจะมีแรงดันตกคร่อม 5 โวลต์ กระแสไหลผ่าน 0.05 มิลลิแอมป์ กระแสนี้ไม่อาจเข้าไปยังอินพุทของออปแอมป์ได้คั้งนั้นจึงไหลไปยังเอาต์พุทโดยผ่านความต้านทาน Rf เท่ากับ 10K จะได้แรงดันเอาต์พุทเท่ากับ $10K * (-0.05)$ เท่ากับ -0.5 โวลต์

เมื่อเปิดสวิตช์ D0 และปิดวงจรที่สวิตช์ D1 (R2 เป็นครึ่งหนึ่งของ R1) กระแสจะเพิ่มขึ้นเป็น 2 เท่าหรือ 0.5 มิลลิแอมป์ไหลผ่าน Rf และ R2 ทำให้มีแรงดันที่เอาต์พุทเป็น -1 โวลต์ต่อไปปิดวงจร D0, D1 จะได้กระแส 0.05 มิลลิแอมป์ไหลผ่าน R1 และ 0.1 มิลลิแอมป์ผ่าน R2 รวมการกระแสเป็น 0.15 มิลลิแอมป์ได้แรงดันเอาต์พุท -0.5 โวลต์

แรงดันเอาต์พุทจะเพิ่มขึ้นเรื่อยๆ เหมือนชั้นบันไดคั้งนั้น 4 บิตจะได้ 15 ระดับแต่ละระดับต่างกัน -0.5 โวลต์



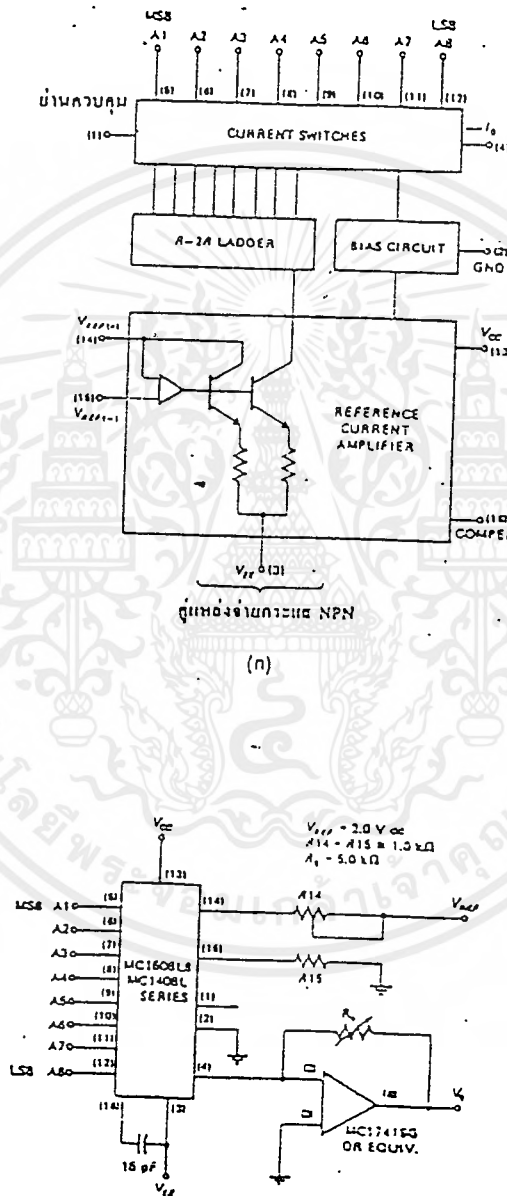
รูปที่ 3.2 วงจร D/A แบบใช้ความต้านทานหลายค่า

3.2.2 แบบใช้ไอซี (Monolithic and hybride D/A converter)

ตัวอย่างของ ไอซีเปลี่ยนสัญญาณดิจิตอลเป็นอนาลอกโมโนลิทิกขนาด 8 บิต MC 1408 มีผังการทำงานคั้งรูป 2.13 ซึ่งเป็นไอซี 16 ขาทำงานที่แรงดัน Vcc +5 โวลต์ และ VEE -5 โวลต์ ถึง -15 โวลต์ใน MC1408 R/2R แลคเตอร์แบ่งกระแสที่ได้จากภาคขยายเป็น 8 ระดับขึ้นอยู่กัค่าทางเลขฐานสอง ทรานซิสเตอร์แบบไบโพลาร์จะสวิตช์ให้กระแสที่สอดคล้องกับอินพุท A1 ถึง A8 การเรียงจากบิตที่มีนัยสำคัญต่ำสุดจะกลับกันกับวงจรนับต่างๆไป แต่วงจรเปลี่ยนสัญญาณดิจิตอลเป็นอนาลอกบางตัวก็ไม่ได้เรียงอย่างนี้ขึ้นอยู่กัแต่ละเบอร์ที่ผลิตมา MC 1408L มีกระแสเอาต์พุทที่

สามารถเปลี่ยนเป็นแรงดันได้ด้วยออปแอมป์และตัวต้านทานดังรูปที่ 3.3 แรงดันเข้าที่พหุสามารถคำนวณได้จาก (3.2)

$$V_{out} = V_{ref} / R_{14} \times R_o(A_1/2 - A_2/4 + \dots + A_6/64 + A_7/128 + A_8/256) \quad (3.2)$$



รูปที่ 3.3 แสดงโครงสร้างและการต่อใช้งานของ D/A เบอร์ MC 1408

ในส่วนของการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกโครงงานนี้ได้ใช้ไอซีเบอร์ DAC 0800

ซึ่งวงจรและรายละเอียดการใช้งานจะได้กล่าวในบทที่ 4

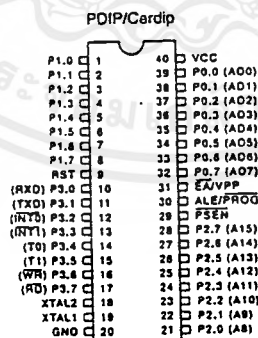
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การอินเทอร์เฟซแบบขนาน

การอินเทอร์เฟซแบบขนานโดยทั่วไปในเครื่องคอมพิวเตอร์จะกระทำผ่านทางพอร์ทัลพรีนเตอร์ซึ่งสามารถนำมาใช้งานเป็นทางเข้าของข้อมูล (input port) และทางส่งข้อมูลออก (output port) ซึ่งรายละเอียดของวงจรพรีนเตอร์พอร์ทัลแสดงดังรูปที่ 3.5

จากวงจรข้างต้นจะพบว่า ในการส่งข้อมูลออก ข้อมูลขนาด 8 บิตจะถูกแลตซ์ผ่านโดย IC 74LS374 ทางขา 2,3,4,5,6,7,8,9 ของหัวต่อ 25 ขา ตามลำดับ ซึ่งสามารถอ่านสถานะบัพเฟอร์ข้อมูลนั้นได้โดย IC 74LS244 แอดเดรสที่ถูกตีโค้ดไว้สำหรับบอร์คอะแดปเตอร์เครื่องพิมพ์คือหมายเลข 378H ในการนำข้อมูลเข้าขนาด 5 บิต สามารถนำเข้าได้ทางขา 10 (Ack), 11 (Busy), 12 (PE), 13 (Select), 15 (Error) โดย IC 74LS240 และ 74LS125 แอดเดรสที่ถูกตีโค้ดไว้คือหมายเลข 389H

ในส่วนของการอินเทอร์เฟซนี้จะใช้ตัวคอนโทรลเลอร์ (89C51) เป็นตัวในการรับส่งข้อมูลกับคอมพิวเตอร์ด้วยโดยจะคอยทำการ Hand Checking ในการรับส่งข้อมูลแต่ละไบต์จนครบ 256 กิโลไบต์ ตัวของ 89C51 นี้เป็นคอนโทรลเลอร์ที่มีรอมภายในขนาด 4 กิโลไบต์มีความสะดวกในการใช้งานโดยไม่จำเป็นต้องต่อรอมภายนอกในการเขียนโปรแกรมสามารถลบและเขียนทับได้ 1000 ครั้ง ซึ่งรูปร่างลักษณะแสดงดังรูปที่ 3.4 และรายละเอียดแสดงในภาคผนวก



รูปที่ 3.4 แสดงรูปร่างของ 89C51

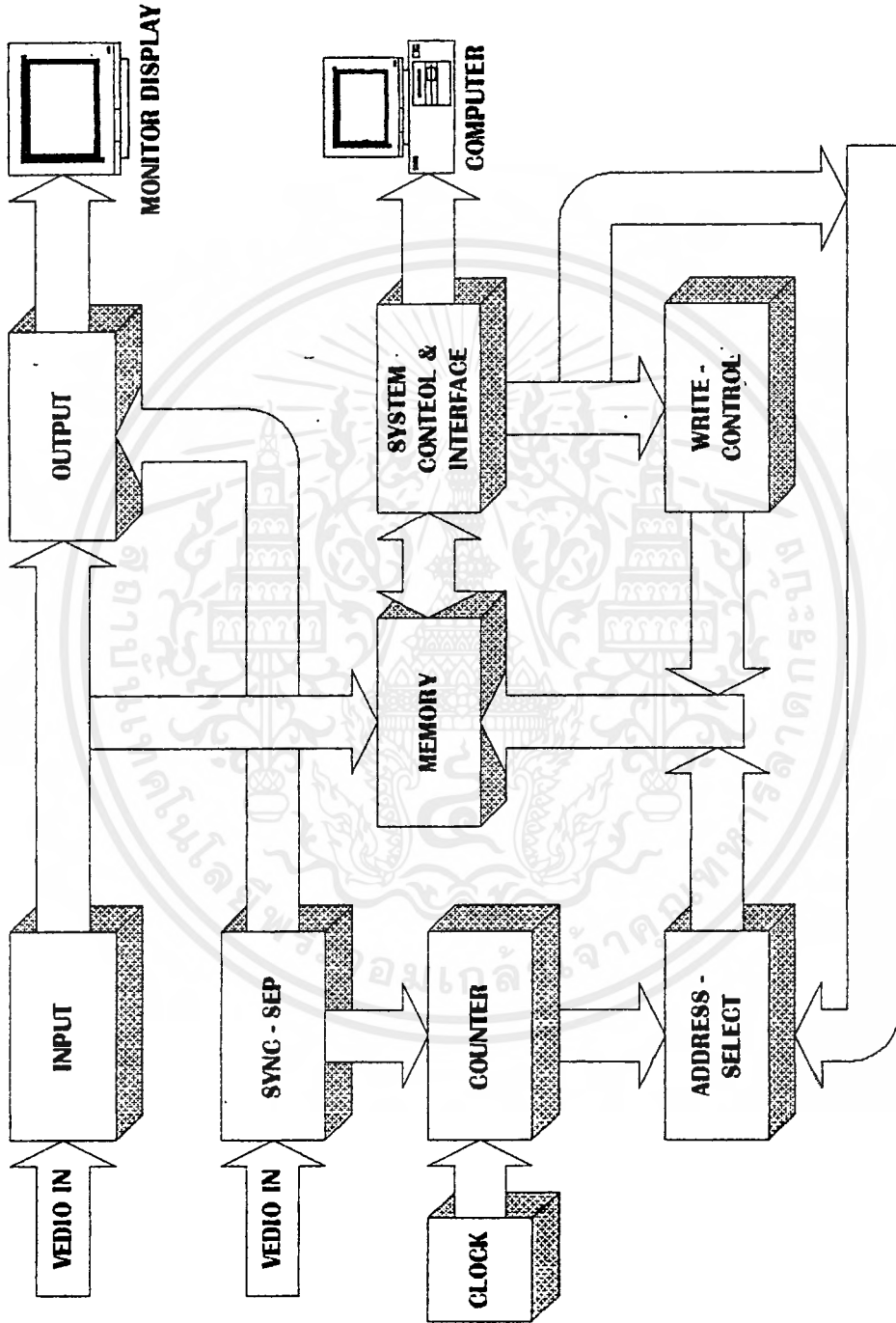
บทที่ 4

การสร้างและรายละเอียดของวงจรส่วนต่างๆ

โครงสร้างของโครงการนี้ จะใช้หลักการรับสัญญาณภาพเข้ามาแล้วทำการแปลงเป็นสัญญาณทางดิจิทัลเพื่อนำไปเก็บไว้ในหน่วยความจำเพื่อนำไปประมวลผลต่อไป ส่วนหนึ่งจะนำไปแยกสัญญาณซิงค์และนำสัญญาณซิงค์ที่ได้ไปควบคุมการแสดงผลทางมอนิเตอร์และควบคุมการทำงานของวงจรมัลติเพลกซ์การควบคุมการเขียน รูปที่ 4.1 เป็นบล็อกไดอะแกรมของวงจรในภาคต่างๆ ซึ่งประกอบด้วยส่วนต่างๆดังนี้

1. ภาคอินพุท
2. ภาคเอาต์พุท
3. ภาคผลิตความถี่
4. ภาคแยกซิงค์
5. ภาควงจรมัลติเพลกซ์
6. ภาคเลือกแอดเดรส
7. ภาคควบคุมการเขียนข้อมูลภาพ
8. ภาคหน่วยความจำ
9. ภาคควบคุมระบบและอินเตอร์เฟส

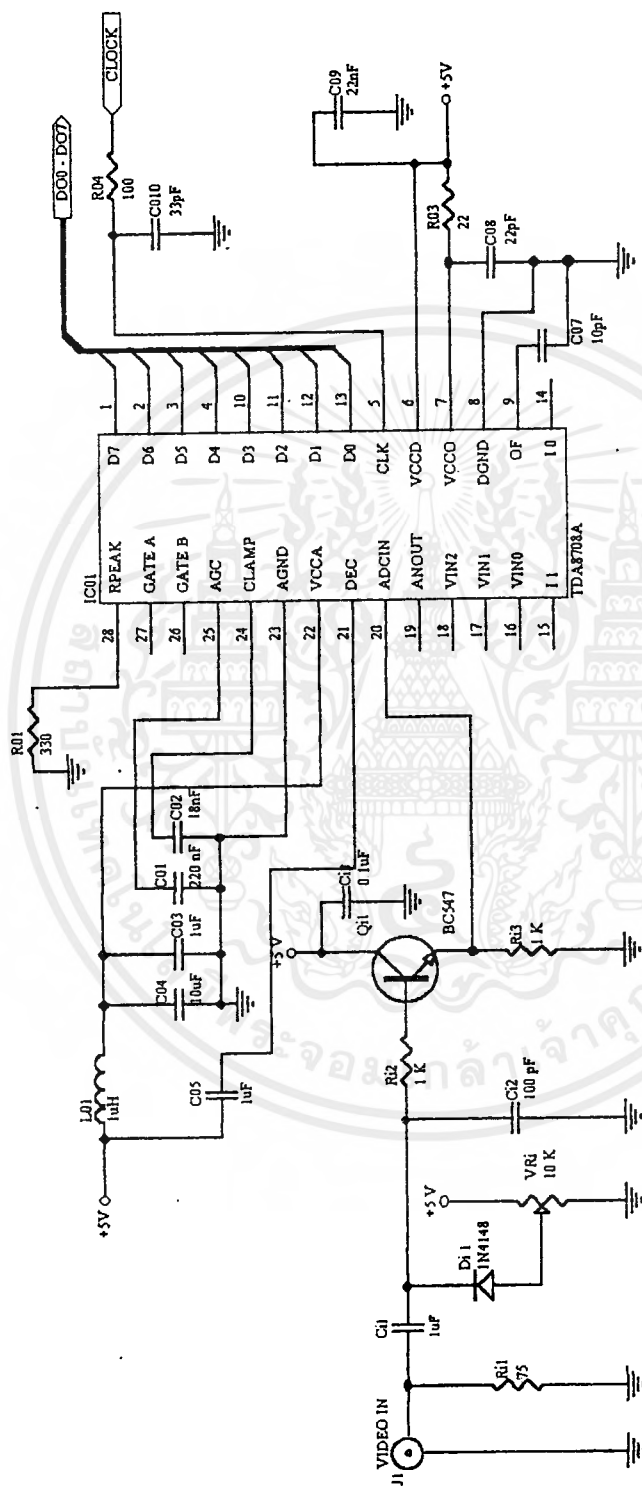
ซึ่งบล็อก ไดอะแกรมแสดงได้ดังรูปที่ 4.1 และการทำงานของภาคต่างๆมีดังต่อไปนี้



รูปที่ 4.1 แสดงบล็อกไดอะแกรมของวงจรภาคต่างๆ

4.1 วงจรและการทำงานของภาคอินพุท

ในภาคอินพุทนี้ได้เลือกใช้ไอซีเบอร์ TDA8708A ในการแปลงสัญญาณภาพซึ่งเป็นสัญญาณอนาลอกให้เป็นสัญญาณทางดิจิทัล โดยในไอซีจะประกอบด้วยภาค Input Selection ที่สามารถเลือกแหล่งสัญญาณได้ 3 แหล่ง ควบคุมอินพุทด้วยการเข้ารหัสเลือกที่ขา 14 และ 15 โดยอินพุทสามารถป้อนเข้าที่ขา 16 , 17 , และ 18 แต่ในการออกแบบเบื้องต้นนี้จะใช้ขา 20 เป็นอินพุท โดยเมื่อสัญญาณภาพที่ได้จากกล้อง วิดีโอชนิด ซีซีดี ถูกส่งเข้าไปยังวงจรที่ทำหน้าที่ป้องกันความผิดพลาดของสัญญาณ ประกอบด้วย Ri 1 ทำหน้าที่ปรับอิมพีแดนซ์ของวงจรให้เหมาะสมกับอิมพีแดนซ์ของกล้องคือ 75 โอห์ม สัญญาณภาพจะถูกส่งเข้า Ci 1 ที่ทำหน้าที่แยกไฟตรงเพื่อให้เฉพาะสัญญาณกระแสสลับผ่านเท่านั้น สัญญาณจาก Ci1 ถูกยกระดับด้วย Di1 และ VR i1 เพื่อให้มีสัญญาณรบกวนน้อยที่สุดก่อนที่จะถูกขยายด้วย Qi1 สัญญาณจาก Qi1 ถูกส่งไปยังขา 20 ของ TDA 8708A ซึ่งสัญญาณจะถูกขยายด้วยภาคขยายภายในออปแอมป์ อีกครั้งแล้วส่งให้วงจรแชนเปลิ่งสัญญาณด้วยความถี่สัญญาณนาฬิกา 10 MHz โดยป้อนเข้าที่ขา 5 สัญญาณดิจิทัลเข้าที่พอร์ทมีขนาด 8 บิตถูกขับออกที่ขา 1, 2, 3, 4, 10, 11, 12, และ 13 โดยขา 13 จะเป็นสัญญาณดิจิทัลบิตต่ำสุด ซึ่งจะมีระดับของสัญญาณ 256 ระดับ สัญญาณดิจิทัลที่ได้จะถูกส่งให้วงจร TTL Output เพื่อส่งสัญญาณไปยังภายนอก และขาที่ 9 ที่เรียกว่า Out Format หรือ Chip Enable จะใช้ในการควบคุมเข้าที่พอร์ทดิจิทัลให้สามารถมีเข้าที่พอร์ทแบบไบนารี , แบบ Two ' Complement หรือให้ขาของสัญญาณเข้าที่พอร์ทมีสถานะเป็นไฮอิมพีแดนซ์ จากวงจรที่ใช้จะเลือกใช้รูปแบบข้อมูลเข้าที่พอร์ทแบบไบนารี ในส่วนของไฟเลี้ยงไอซีจะจ่ายเข้ามาที่ขา 6 , 22 และขา 7 มีขนาด 5 โวลท์

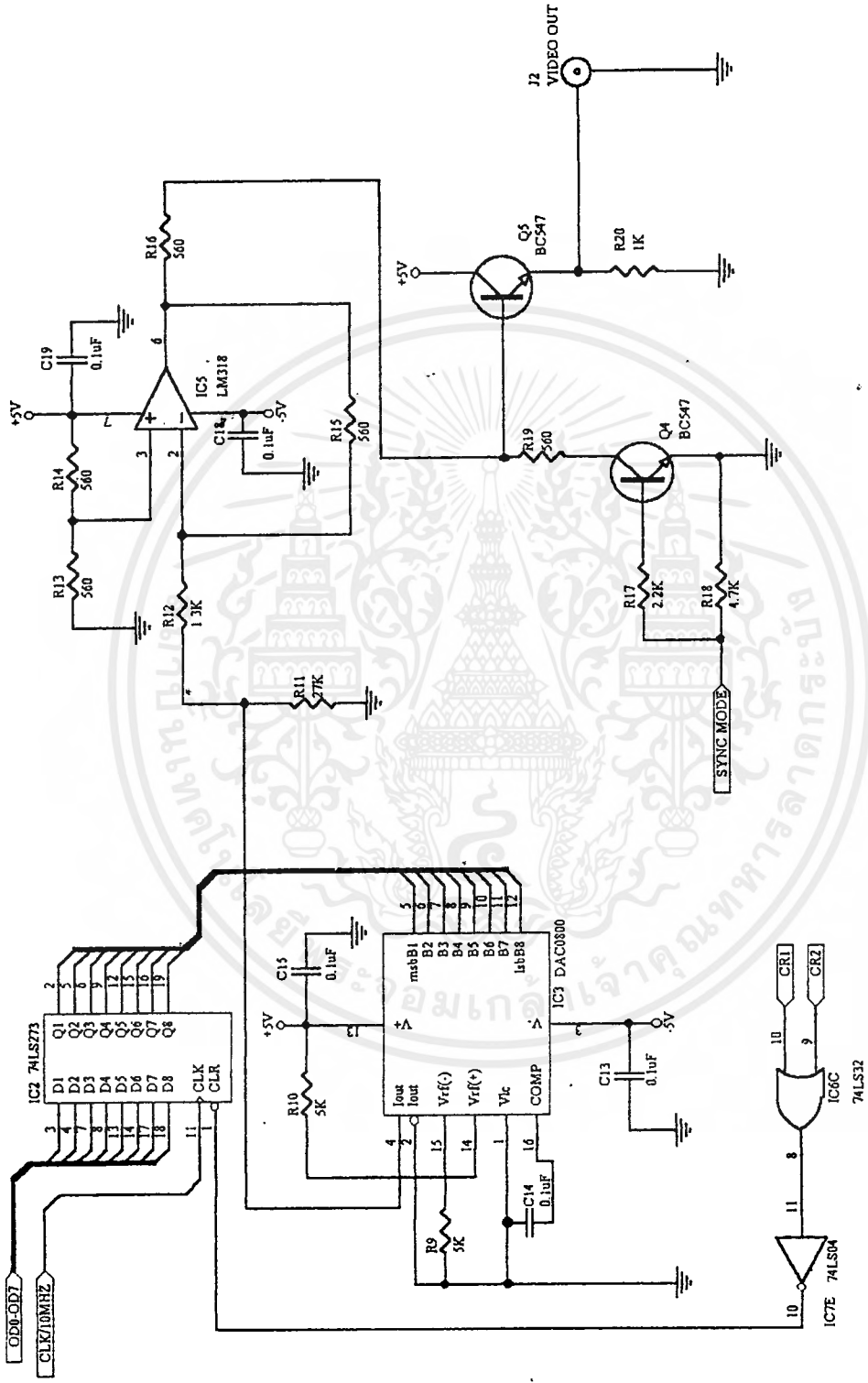


รูปที่ 4.2 แสดงวงจรภาคอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจรและการทำงานภาคเอาต์พุต

สัญญาณดิจิตอลจากภาคอินพุตส่วนหนึ่งจะถูกส่งเข้าสู่วงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกซึ่งสัญญาณดิจิตอลนี้ถูกส่งผ่าน IC2 (74LS273) ซึ่งทำหน้าที่เป็นบัฟเฟอร์ข้อมูลเอาต์พุตเพื่อส่งข้อมูลไปยัง IC3 (DAC 0800) ซึ่งทำหน้าที่แปลงสัญญาณดิจิตอลเป็นอนาล็อก IC2 ทำงานโดยอาศัยสัญญาณนาฬิกา 10MHZ และสัญญาณควบคุมการส่งข้อมูล โดยควบคุมที่ขาเรีเซ็ท (ขา 1) IC2 จะให้เอาต์พุตตามสัญญาณนาฬิกา คือทุกขอบขาขึ้นของสัญญาณนาฬิกาและสัญญาณควบคุมการส่งข้อมูลจะต้องเป็น “1” สัญญาณควบคุมนี้ได้จากการรวมสัญญาณการนับทางแนวนอน (CR1) และสัญญาณควบคุมการนับทางแนวตั้ง (CR2) แบบนอร์เกต ผ่านทาง IC6: C และ IC7: E จะเห็นว่าสัญญาณการส่งข้อมูลนี้จะเป็น “1” คอเมื่อสัญญาณควบคุมการนับทางแนวนอนและแนวตั้งต้องเป็น “0” ถ้านอกเหนือจากนี้จะทำให้สัญญาณควบคุมการส่งข้อมูลเป็น “0” (สัญญาณควบคุมการนับทั้งทางแนวตั้งและแนวนอน ถ้ามีสัญญาณเป็น “0” หมายความว่ากำลังมีการนับเกิดขึ้น แต่ถ้ามีสัญญาณเป็น “1” จะไม่มีการนับเกิดขึ้น) สัญญาณดิจิตอลที่ออกจาก IC2 จะถูกส่งเข้า IC3 ซึ่ง IC3 จะทำงานได้นั้นต้องอาศัยระดับแรงดันอ้างอิงทางด้านบวกและแรงดันอ้างอิงด้านลบเป็นตัวกำหนดค่าแรงดันสูงสุดและต่ำสุดที่จะได้ทางเอาต์พุตหลังการเปลี่ยนแปลงข้อมูลเป็นสัญญาณอนาล็อกแล้ว ระดับแรงดันอ้างอิงด้านบวกที่ให้กับ IC3 จะต่อกับไฟเลี้ยงวงจรที่มีแรงดัน +5V ผ่าน R10 ส่วนระดับแรงดันอ้างอิงด้านลบจะต่อลงกราวด์ผ่าน R9 เอาต์พุตที่ได้จาก IC3 จะเป็นกระแส จึงต้องต่อค่าความต้านทาน R11 เพื่อทำให้เป็นแรงดันแรงดันที่ได้จะถูกส่งไปยัง IC5 เพื่อทำการขยายสัญญาณให้แรงขึ้นก่อนที่จะส่งออกแสดงที่มอนิเตอร์ โดยอัตราการขยายถูกกำหนดด้วย R12 และ R15 และก่อนที่จะส่งสัญญาณแสดงออกมอนิเตอร์จำเป็นต้องรวมสัญญาณภาพเข้ากับสัญญาณซิงค์ก่อน ซึ่งได้จาก IC10 (LM 1881) ซึ่งทำหน้าที่แยกซิงค์ออกจากสัญญาณภาพรวม เมื่อทำการรวมสัญญาณภาพกับซิงค์แล้วก็จะทำการขยายที่วงจรขยายซึ่งประกอบด้วย R16, R17, R18, R19, R20, Q4 และ Q5 แล้วส่งไปแสดงผลยังมอนิเตอร์ต่อไป

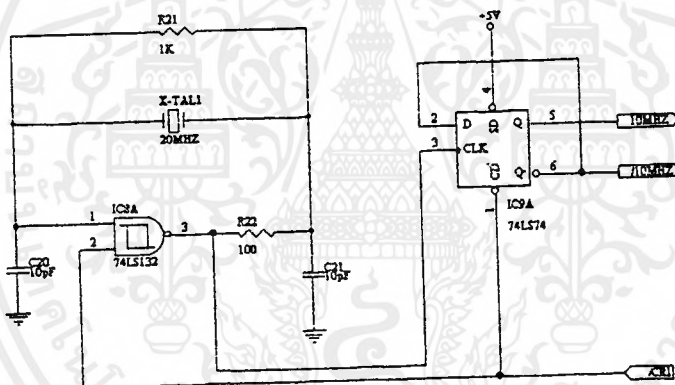


รูปที่ 4.3 แสดงวงจรภาคเข้าที่พุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรและการทำงานภาคผลิตความถี่

การทำงานของภาคผลิตความถี่ประกอบด้วย X-TAL1 ,R21 ,R22 ,C20 ,C21 ,IC8:A ที่เป็นตัวกำหนดสัญญาณนาฬิกา และ IC9:A ซึ่งทำหน้าที่หารความถี่การทำงานของภาคผลิตความถี่สัญญาณนาฬิกาที่ผลิตได้จะถูกควบคุมให้ทำงานเป็นจังหวะๆ สัญญาณที่ใช้ควบคุมวงจรผลิตความถี่คือ /CR1 เหตุที่ต้องมีสัญญาณควบคุมก็เพื่อให้สัญญาณนาฬิกาที่ใช้ในการส่งข้อมูลใน เอชดีคอนเวอร์เตอร์สัมพันธ์กับสัญญาณภาพที่ทำการเก็บ และสัญญาณที่ได้จากวงจรผลิตความถี่จะมีขนาด 10 MHZ



รูปที่ 4.4 วงจรผลิตความถี่

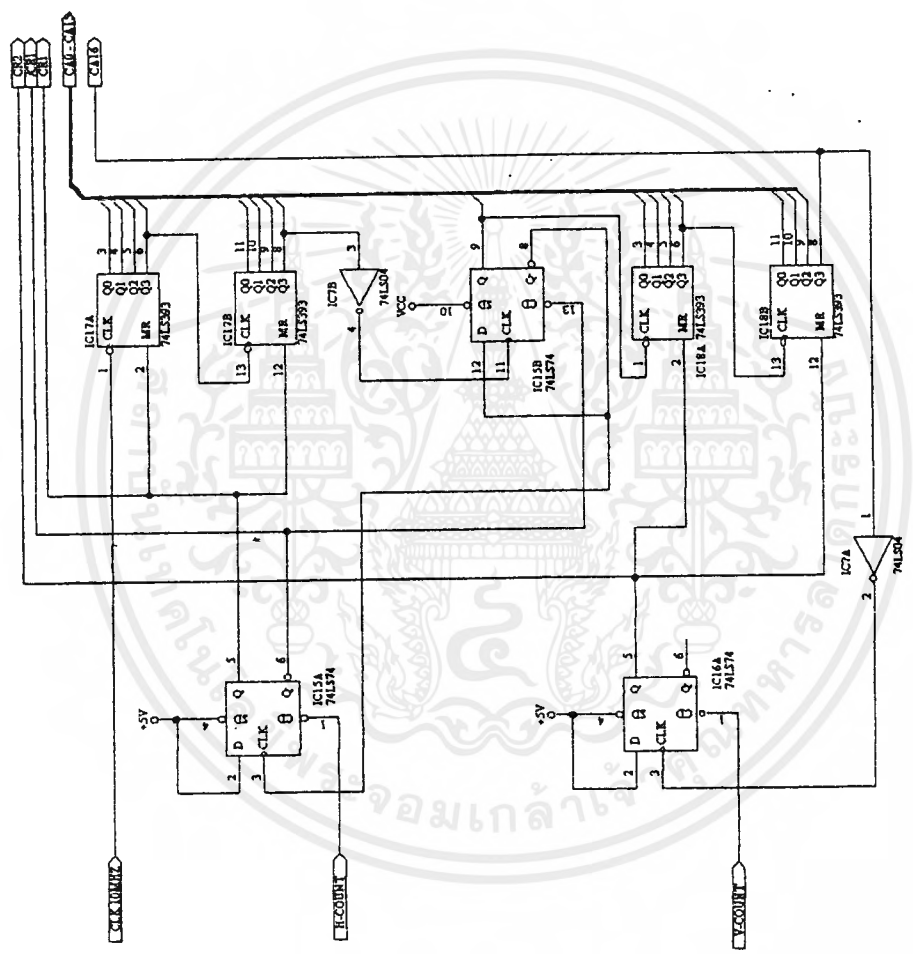
4.4 วงจรและการงานภาคแยกซิงค์

การทำงานภาคแยกซิงค์ จะนำสัญญาณภาพออกเป็นจากกล้องวิดีโอชนิดซีซีดี หรือแหล่งกำเนิดสัญญาณภาพอื่น ๆ ส่วนหนึ่งเข้ามายังอินพุตที่ขา V_{IN} (ขา 2) ของ IC10 (LM 1881) ซึ่งเป็นไอซีทำหน้าที่แยกสัญญาณซิงค์ออกจากสัญญาณภาพ โดยมี C23 ซึ่งทำหน้าที่คัปปลิงสัญญาณกันกลางระหว่างขา 2 ของ IC10 และสัญญาณภาพที่จะส่งเข้าไป นอกจากนี้ IC10 จะทำงานได้นั้นต้องต่อร่วมกับตัวต้านทานและตัวเก็บภาวนอกคือ R19, C24 เพื่อกำหนดช่วงเวลาของวงจรผลิตความถี่ภายใน เอาท์พุทที่ได้จาก IC10 จะแบ่งออกเป็นสัญญาณซิงค์รวม (ขา 1) สัญญาณซิงค์ทางแนวตั้ง (ขา 3) และสัญญาณฟิลด์คู่ฟิลด์คี่ (ขา 7) สัญญาณซิงค์รวมจะถูกส่งไปยังชุดหน่วยเวลาทางด้านแนวอนซึ่งประกอบด้วย R24, VR4, C25, IC11:B, R25, C26 และ IC 11:A การทำงานของชุดหน่วยเวลาทางด้านแนวอนคือจะทำการหน่วงซิงค์รวมประมาณ 9 μ S ด้วย R24, VR4, C25, IC11:B เมื่อการหน่วงสิ้นสุดลงจะเกิดสัญญาณพัลส์ช่วงสั้น ๆ ประมาณ 140 nS สัญญาณพัลส์นี้ได้จาก R25, C26, IC11:A ซึ่งสัญญาณพัลส์ที่ได้นี้จะถูกส่งไปยังวงจรควบคุมการนับทางแนวอนเพื่อทำการรีเซ็ตให้วงจรนับทางแนวอนเริ่มทำการนับได้ วัตถุประสงค์ของวงจรหน่วงเวลาก็เพื่อให้วงจรนับทางแนวอน เริ่มทำการนับ ณ ตำแหน่งเริ่มต้นของการสะแกนสัญญาณภาพที่เท่ากันทุกเส้นสะแกน เพราะถ้าเราเอาสัญญาณซิงค์รวมไปรีเซ็ตวงจรควบคุมการนับทางแนวอนโดยไม่มีหน่วยเวลาเลย จะทำให้การนับทางแนวอนแต่ละครั้งไม่เริ่มต้นที่ตำแหน่งของการสะแกนแต่จะเริ่มคั่นก่อนหน้าการสะแกนซึ่งเป็นการไม่ถูกต้อง

นอกจากนี้สัญญาณซิงค์รวมจาก IC10 จะถูกส่งไปยังวงจรนับสัญญาณซิงค์ด้วย เพื่อหน่วยเวลาให้กับสัญญาณซิงค์ทางแนวตั้ง วงจรนับสัญญาณซิงค์นี้ประกอบด้วย IC9:B, IC12:A และ IC12:B เป็นไบนารีเค้านเตอร์การทำงานคือเมื่อมีสัญญาณซิงค์ทางแนวตั้ง (มีค่าเป็น "0") เข้ามาจะทำให้เกิดการรีเซ็ตชุดควบคุมการนับสัญญาณซิงค์รวมเพราะสัญญาณซิงค์ทางแนวตั้งต่อเข้ากับขารีเซ็ตของชุดควบคุมการนับซิงค์รวม ทำให้เอาท์พุท Q ของ IC9:B มีค่าเป็น "0" ผลก็คือทำให้ IC12:A, IC12:B ทำการนับสัญญาณซิงค์รวมนั้น จากวงจรจะเป็นวงจรนับ 32 คือ ถ้านับสัญญาณซิงค์รวมได้ครบ 32 ลูก ก็จะทำให้เกิดสัญญาณพัลส์เล็ก ๆ ประมาณ 140 nS โดยที่สัญญาณพัลส์ดังกล่าวเกิดจาก R26, C27, และ IC13:A โดยส่งไปรีเซ็ตวงจรการนับทางแนวตั้ง เพื่อให้เกิดการนับทางแนวตั้งต่อไป ซึ่งการนับซิงค์รวมนี้จะเริ่มทำการนับอีกครั้งภายหลังจากมีสัญญาณซิงค์ทางแนวตั้งลูกใหม่เข้ามา

4.5 วงจรและการทำงานของภาควงจรมับ

วงจรมับและวงจรควบคุมการนับทั้งแนวนอนและแนวตั้งประกอบด้วย IC17 :A, IC17 :B, IC18 :A, IC18 :B, IC15 :A, IC15 :B, IC16 :A, IC7 :A, IC7 :B การทำงานคือสัญญาณนาฬิกาขนาด 10 MHz จากวงจรกำเนิดความถี่ถูกส่งเข้าที่ขา 1 ของ IC17 :A ซึ่งเป็นไอซีไบนารีอะซิงโครนัสเคาร์เตอร์ขนาด 4 บิต และเอาต์พุต CA3 (ขา 6) ถูกต่อเข้าสู่อินพุตของ IC17 :B ที่ขา 13 และเอาต์พุต CA7 (ขา 8) ของ IC17 :B ก็ต่อเข้ากับอินพุตของ IC15 :B ผ่าน IC7 :B เอาต์พุต Q ขา 9 ของ IC15 :B ถูกต่อเป็นอินพุตให้กับ IC18 :A (ขา 1) และเอาต์พุต CA12 (ขา 16) ของ IC18 :A ต่อเป็นอินพุตให้กับ IC18 :B การต่อเช่นนี้ก็เพื่อให้ได้สัญญาณเอาต์พุตของวงจรมับรวมทั้งหมด 17 เส้น เพื่อใช้ในการอ้างตำแหน่งของข้อมูลที่จะเก็บในหน่วยความจำ ซึ่งการเก็บสัญญาณภาพ 1 เฟรม ใช้หน่วยความจำ 128 กิโลไบต์ ซึ่งต้องอ้างตำแหน่งถึง 17 เส้น การอ้างตำแหน่งนี้แบ่งออกเป็น 2 ส่วนคือ IC17 :A, IC17 :B และ IC15 :B จะให้เอาต์พุตของการนับ 9 เส้นเพื่อใช้อ้างตำแหน่งหน่วยความจำภาพทางแนวนอน (เก็บข้อมูล 1 เส้นสแกนภาพ) มีขนาด 512 จุดภาพ ส่วนที่ 2 ประกอบด้วย IC18 :A , IC18 :B ให้เอาต์พุตการนับ 8 เส้นใช้ในการกำหนดตำแหน่งของหน่วยความจำภาพทางแนวตั้ง (ใช้นับเส้นสแกนภาพที่ทำการเก็บข้อมูล) ซึ่งการอ้างตำแหน่งนี้อ้างได้ 256 เส้น การอ้างตำแหน่งหน่วยความจำภาพที่แบ่งออกเป็นสองส่วนนี้จะต้องมีชุดควบคุมการนับ 2 ชุด คือ IC15 :A ใช้ควบคุมการนับทางแนวนอนและ IC16 :A ใช้ควบคุมการนับทางแนวตั้งวงจรมับทางแนวนอนจะมีการนับก็ต่อเมื่อสัญญาณควบคุมการนับ (CR1) มีค่าเป็น "0" ให้ IC17 :A และ IC17 :B และเป็น "1" ให้ IC15 :B ส่วนวงจรการนับทางแนวตั้งจะทำการนับได้ก็ต่อเมื่อสัญญาณควบคุมการนับ (CR2) ต้องมีสัญญาณ "0" ให้กับ IC18 :A, IC18 :B การที่วงจรควบคุมการนับจะให้วงจรควบคุมการนับเป็น "0" หรือ "1" ขึ้นอยู่กับพัลส์รีเซ็ตวงจรควบคุมการนับ (เป็นสัญญาณที่เกิดหลังจากมีการหน่วงสัญญาณซิงค์แล้ว) สัญญาณพัลส์รีเซ็ตวงจรควบคุมการนับทางแนวนอนคือสัญญาณ H - COUNT ส่วนสัญญาณพัลส์รีเซ็ตวงจรควบคุมการนับทางแนวตั้งคือสัญญาณ V - COUNT ถ้าสัญญาณนี้มีค่าเป็น "0" จะทำให้เกิดการนับได้ สัญญาณรีเซ็ตนี้จะ เป็นเพียงสัญญาณพัลส์เล็ก ๆ วัตถุประสงค์ก็เพียงเพื่อรีเซ็ตวงจรควบคุมให้ได้เท่านั้น หลังจากที่เอาต์พุตของสัญญาณควบคุมเป็น "0" แล้ว รีเซ็ตพัลส์นี้จะมีค่าเป็น "1" การที่เอาต์พุตของวงจรควบคุมการนับทางแนวนอนหรือทางแนวตั้งจะมีค่าเป็น "1" ได้นั้นก็เกิดจากการที่วงจรมับทางแนวนอนนับครบ 512 จุดภาพ หรือวงจรมับทางแนวตั้งนับครบ 256 เส้น กล่าวคือวงจรมับจะเริ่มนับเมื่อสัญญาณพัลส์รีเซ็ตเข้ามาสู่วงจรควบคุมการนับ และจะหยุดนับเมื่อนับครบ 512 (ทางด้านแนวนอน) หรือ 256 (ทางด้านแนวตั้ง) และจะหยุดนับจนกว่าจะมีสัญญาณพัลส์รีเซ็ตถูกต่อไปเข้ามา



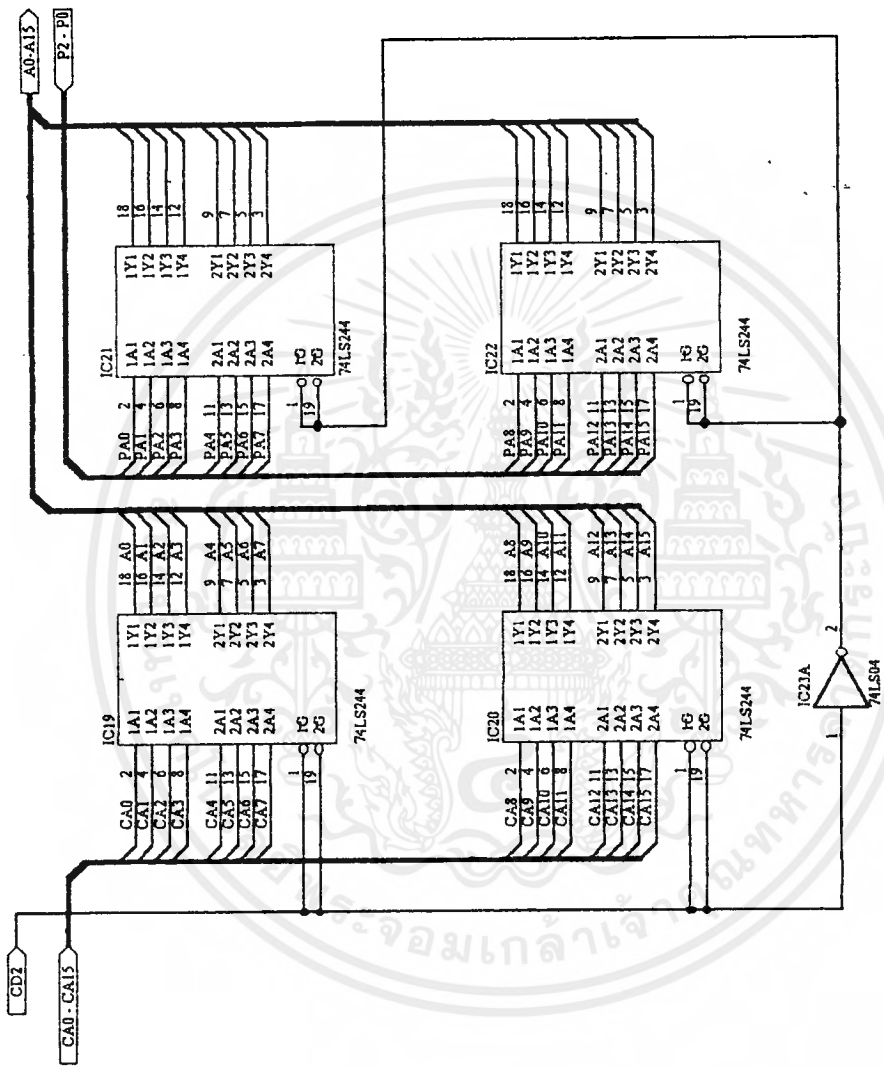
รูปที่ 4.6 แสดงวงจรภาควงนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 วงจรและการทำงานภาคเลือกแอดเดรส

ภาคเลือกแอดเดรสมีหน้าที่เลือกแอดเดรสที่มาจากวงจรรับ (กรณีเก็บข้อมูลภาพเข้าหน่วยความจำ) หรือเลือกสัญญาณแอดเดรสที่มาจาก MCS - 51 (กรณีอ่านข้อมูลส่งไปยังคอมพิวเตอร์) IC19, IC20, IC21, IC22 ทำหน้าที่เป็นบัฟเฟอร์ IC19, IC20 จะติดต่อกับแอดเดรสที่มาจากวงจรรับ และ IC21, IC22 จะติดต่อกับแอดเดรสที่มาจาก MCS - 51 สัญญาณ CD2 และ /CD2 เป็นสัญญาณที่กำหนดว่าจะเลือกแอดเดรสที่มาจากวงจรรับหรือ MCS - 51 สัญญาณทั้งสองจะมีเฟสต่างกัน ดังนั้น CD2 จะต่อเข้ากับ IC19, IC20 ที่ขาอินเวิร์ต (ขา 1 และ ขา 19) และ /CD2 จะถูกต่อเข้ากับ IC21, IC22 (ที่ขา 1 และ ขา 19) เมื่อสัญญาณอินเวิร์ตของ IC19, IC20 หรือ IC21, IC22 ชูคใดชุดหนึ่งเป็น "0" จะทำให้มีข้อมูลเข้าที่ชุดที่ชุดนั้น แต่ถ้าชุดใดชุดหนึ่งมีขาอินเวิร์ตเป็น "1"

ชุดนั้นจะมีเข้าที่ชุดเป็นไฮอิมพีแดนซ์ จากรูปวงจรภาคเลือกแอดเดรสจะเห็นว่า IC19, IC20 มีอินพุตต่ออยู่กับวงจรรับทำให้แอดเดรสของหน่วยความจำภาพถูกกำหนดด้วยวงจรรับ ส่วน IC21, IC22 มีอินพุตต่ออยู่กับแอดเดรสของ MCS - 51 ดังนั้นแอดเดรสของหน่วยความจำภาพจึงถูกกำหนดด้วยแอดเดรสของ MCS - 51 การเลือกแอดเดรสจะทำได้เพียง 16 เส้น ดังนั้นสัญญาณที่เหลืออีก 1 เส้นจะถูกกำหนดโดยตรงจาก MCS - 51



รูปที่ 4.7 แสดงวงจรภาคเลือกแอสตเรส

4.7 วงจรและการทำงานของภาคควบคุมการเขียนข้อมูลภาพ

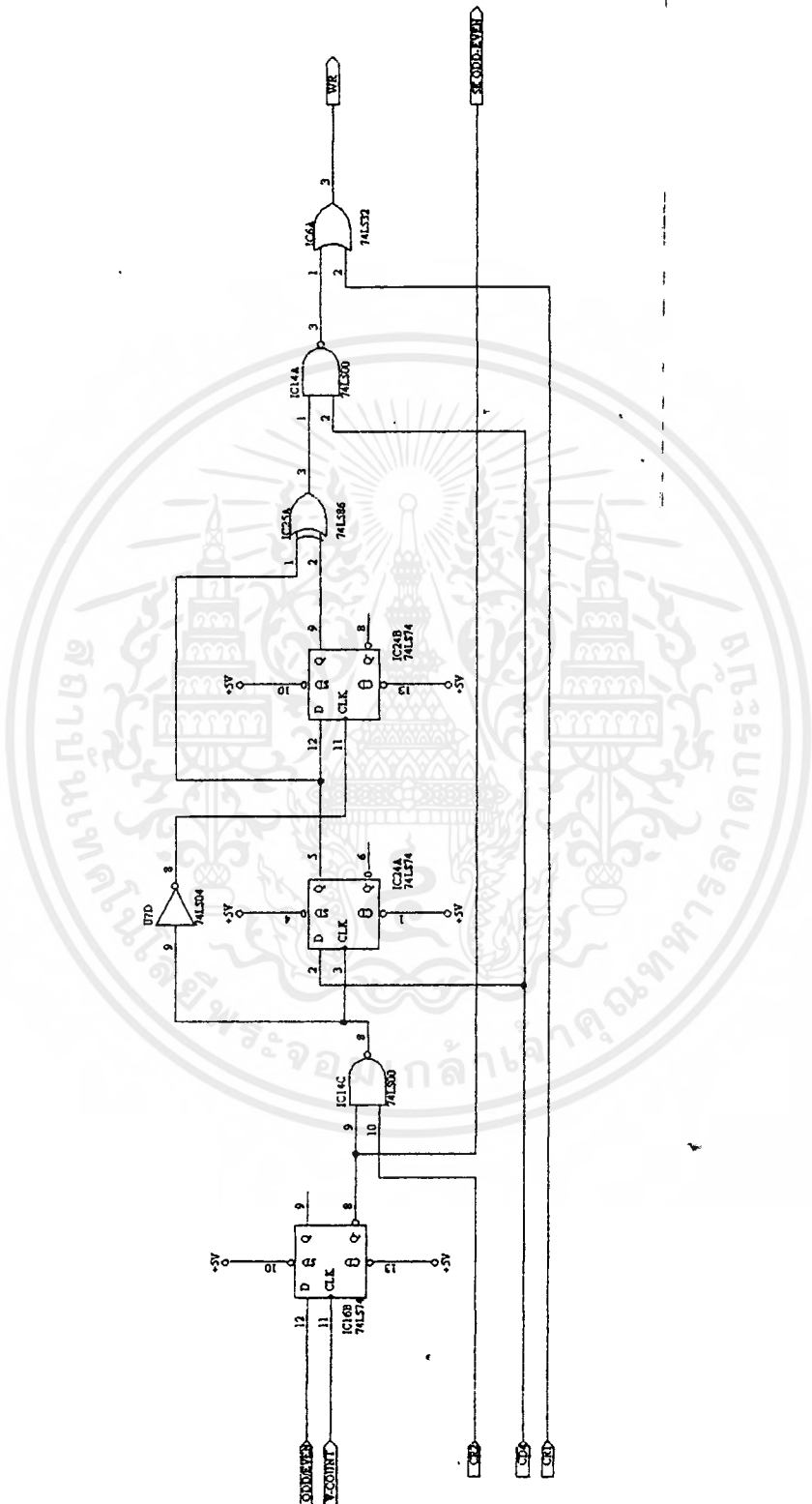
การทำงานคือ เมื่อสัญญาณอินพุทของ IC24:A เปลี่ยนสถานะ (จากการใช้งานจะเปลี่ยนจาก “0” เป็น “1”) IC24:A จะเกิดการเปลี่ยนแปลงเข้าที่พุทที่ต่อเมื่อสัญญาณที่ต่อกับขานาฬิกา (ขา 3) เปลี่ยนสถานะจาก “0” เป็น “1” สัญญาณนี้ได้จากการแนบคั่นระหว่างสัญญาณควบคุมการนับทางแนวตั้ง (CR2) กับสัญญาณกำหนดฟิลด์คู่ฟิลด์คี่ กล่าวคือ สัญญาณนี้จะ เป็น “1” ก็ต่อเมื่อสัญญาณ (CR2) เป็น “0” และ/หรือ สัญญาณกำหนดฟิลด์คู่หรือฟิลด์คี่ เป็น 000“0” และสัญญาณนี้จะ เป็น “0” ก็ต่อเมื่อ สัญญาณ CR2 เป็น “1” และสัญญาณกำหนดฟิลด์คี่หรือฟิลด์คู่เป็น “1” เท่านั้น ด้วยเหตุนี้เราให้มีการเปลี่ยนแปลงของข้อมูลทางด้านอินพุทของ IC24:A (จาก “0” เป็น “1”) เข้าที่พุทของ IC24:A จะเกิดการเปลี่ยนแปลงตามอินพุท เมื่อสัญญาณควบคุมการนับแนวตั้ง (CR2) เป็น “0” จากวงจรนั้นเข้าที่พุทของ IC24:A จะถูกต่อเป็นอินพุทให้กับ IC24:B และเข้าที่พุทของ IC24:B จะเกิดการเปลี่ยนแปลงก็ต่อเมื่อสัญญาณที่ขานาฬิกาเปลี่ยนสถานะจาก “0” เป็น “1” แต่เนื่องจากสัญญาณที่ขานาฬิกาของ IC24:B จะต่อกลับเฟสกับสัญญาณที่ขานาฬิกาของ IC24:A โดยใช้ IC7:D เป็นตัวกลับเฟส

ในช่วงที่สัญญาณ CR2 เป็น “0” หมายถึง ช่วงที่วงจรนับทางแนวตั้งกำลังทำการนับเพื่อ กำหนดแอดเดรสให้กับหน่วยความจำภาพ จังหวะที่สัญญาณ CR2 จะเปลี่ยนสถานะจาก “1” เป็น “0” ทำให้สัญญาณที่นาฬิกาของ IC24:A เปลี่ยนสถานะเช่นกันคือ จาก “0” เป็น “1” นั้นหมายถึง ถ้าอินพุทของ IC24:A มีการเปลี่ยนแปลงสถานะก่อนหน้าและคงค่าไว้ จะทำให้เข้าที่พุทของ IC24:A เกิดการเปลี่ยนแปลงสถานะตามอินพุทในจังหวะนี้ จากที่ได้กล่าวแล้วว่าเข้าที่พุทของ IC24:A จะต่อเข้ากับอินพุทของ IC24:B ดังนั้น IC24:B จะมีการเปลี่ยนแปลงทางเข้าที่พุทตามอินพุทที่ต่อเมื่อ สัญญาณ CR2 เปลี่ยนสถานะจาก “0” เป็น “1” และสัญญาณกำหนดฟิลด์คู่หรือฟิลด์คี่เป็น “1” จากการเปลี่ยนแปลงนี้เราจะนำไปควบคุมการเขียนข้อมูลภาพเข้าสู่หน่วยความจำภาพ (ใช้เวลาการเขียน 2 ฟิลด์) โดยต่อเข้าที่พุทของ IC24:A และเข้าที่พุทของ IC24:B เข้ากับ IC25:A เพื่อเปรียบเทียบข้อมูลกันคือ ถ้าเข้าที่พุทของ IC24:A และ IC24:B เหมือนกัน IC25:A จะให้เข้าที่พุทเป็น “0” แต่ถ้าข้อมูลเข้าที่พุทของ IC24:A และ IC24:B ต่างกัน IC25:A จะให้เข้าที่พุทเป็น “1” เข้าที่พุท IC24:A จะต่อเป็นอินพุทให้กับ IC14:A โดยอินพุทข้างหนึ่งของ IC14:A ถูกต่อเข้ากับสัญญาณควบคุม CD4 (ส่งมาจาก MCS - 51) IC14:A จะให้ค่าเข้าที่พุทเปลี่ยนแปลงตามอินพุทที่ได้จาก IC25:A หรือ ไม่ก็ขึ้นอยู่กับสัญญาณควบคุม CD4 คือถ้า CD4 เป็น “1” สัญญาณเข้าที่พุทของ IC14:A จะเปลี่ยนแปลงตามอินพุทที่ต่อกับ IC25:A แต่ถ้า CD4 เป็น “0” เข้าที่พุท IC14:A จะมีสถานะเป็น “0” ได้เมื่อสัญญาณ CD4 มีค่าเป็น “1” และเข้าที่พุทของ IC25:A เป็น “1” ซึ่งจะเกิดในช่วงที่ข้อมูลอินพุทของ IC24:A ต่อเข้าเป็นอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ IC6 :A และอินพุตอีกขาหนึ่งของ IC6 :A ถูกต่อกับสัญญาณควบคุมการนับทางแนวนอน (CR1) ซึ่งเอาต์พุตของ IC6 :A จะเป็น “0” ได้ก็ต่อเมื่อสัญญาณ CR1 เป็น “0” (หมายถึงการนับเพื่อกำหนดแอดเดรสให้กับหน่วยความจำภาพทางแนวนอน) และสัญญาณเอาต์พุตของ IC14 :A เป็น “0” เท่านั้น





รูปที่ 4.8 แสดงวงจรภาคควบคุมการเขียนข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 วงจรและการทำงานของภาคหน่วยความจำภาพ

วงจรหน่วยความจำภาพประกอบด้วย IC26, IC27, IC28 และ IC29 โดยที่ IC26 จะเป็นบัฟเฟอร์ข้อมูลระหว่างวงจรแปลงสัญญาณกับหน่วยความจำภาพ (IC28, IC29) ในส่วนของ IC27 จะเป็นบัฟเฟอร์ข้อมูลภาพระหว่างหน่วยความจำภาพกับ CPU ในการใช้งานเราจะเลือกติดต่อระหว่างหน่วยความจำภาพกับวงจรแปลงสัญญาณ หรือ CPU อย่างใดอย่างหนึ่ง หมายความว่า IC26 กับ IC27 จะไม่สามารถทำงานพร้อมกันได้ การควบคุม IC26 ให้ทำงานต้องอาศัยสัญญาณ CD2 ซึ่งต่อกับขาอินพุต (ขา 19) คือถ้า สัญญาณ CD2 เป็น "0" IC26 จะสามารถส่งผ่านข้อมูลได้ แต่ถ้า CD2 เป็น "1" ทำให้ขาข้อมูลของ IC26 จะเป็นไฮอิมพีแดนซ์ การที่ IC26 จะทำการส่งข้อมูลจากวงจรแปลงสัญญาณเข้าสู่หน่วยความจำภาพได้นั้นต้องควบคุมให้สัญญาณ CD0 เป็น "0" และในส่วนของ IC27 จะทำงานในช่วงของการอ่านข้อมูลส่งไปยังคอมพิวเตอร์โดย CPU ซึ่งจะต้องควบคุมให้สัญญาณ CD2 (ขา 19) มีค่าเป็น "0" และสัญญาณ CD3 (ขา 1) มีค่าเป็น "1" ซึ่งสามารถส่งได้จาก CPU

IC28 และ IC29 เป็นหน่วยความจำภาพขนาด 128 กิโลไบต์ ใช้เก็บข้อมูลภาพในแต่ละฟิลด์คือ IC29 จะเก็บข้อมูลภาพในฟิลด์คู่ และ IC28 จะเก็บข้อมูลภาพในฟิลด์คี่เนื่องจาก IC28, IC29 เป็นหน่วยความจำขนาด 128 กิโลไบต์จึงต้องใช้แอดเดรสในการอ้างตำแหน่ง 17 เส้น คือ A0 - A16 สัญญาณแอดเดรสนี้ได้จากวงจรนับภายในระบบ หรือ CPU นอกจากนี้ IC28, IC29 จะทำงานได้นั้นต้องมีสัญญาณอื่น ๆ อีก คือ สัญญาณควบคุมไอซี 1 (/CS1), สัญญาณควบคุมไอซี 2 (CS2), สัญญาณควบคุมการเขียน (/WE), สัญญาณควบคุมเอาต์พุต (/OE) การต่อใช้งานของสัญญาณต่าง ๆ มีดังนี้คือ IC29 (เก็บข้อมูลภาพฟิลด์คี่) เราจะต่อ CS2 กับ +Vcc, /CS1 ต่อเข้ากับสัญญาณควบคุมการเลือกหน่วยความจำ (ได้จากขา 7 ของ IC30) สัญญาณ /WE ต่อเข้ากับสัญญาณควบคุมการเขียน และสัญญาณ /OE ต่อเข้ากับสัญญาณควบคุมเอาต์พุตหน่วยความจำ ส่วน IC28 (เก็บข้อมูลภาพฟิลด์คู่) สัญญาณ /CS1 ต่อเข้ากับกราวด์, สัญญาณ CS2 ต่อเข้ากับสัญญาณควบคุมการเลือกหน่วยความจำ, สัญญาณ /WE ต่อเข้ากับสัญญาณควบคุมการเขียน และสัญญาณ /OE ต่อเข้ากับสัญญาณควบคุมเอาต์พุตหน่วยความจำเช่นเดียวกับ IC29

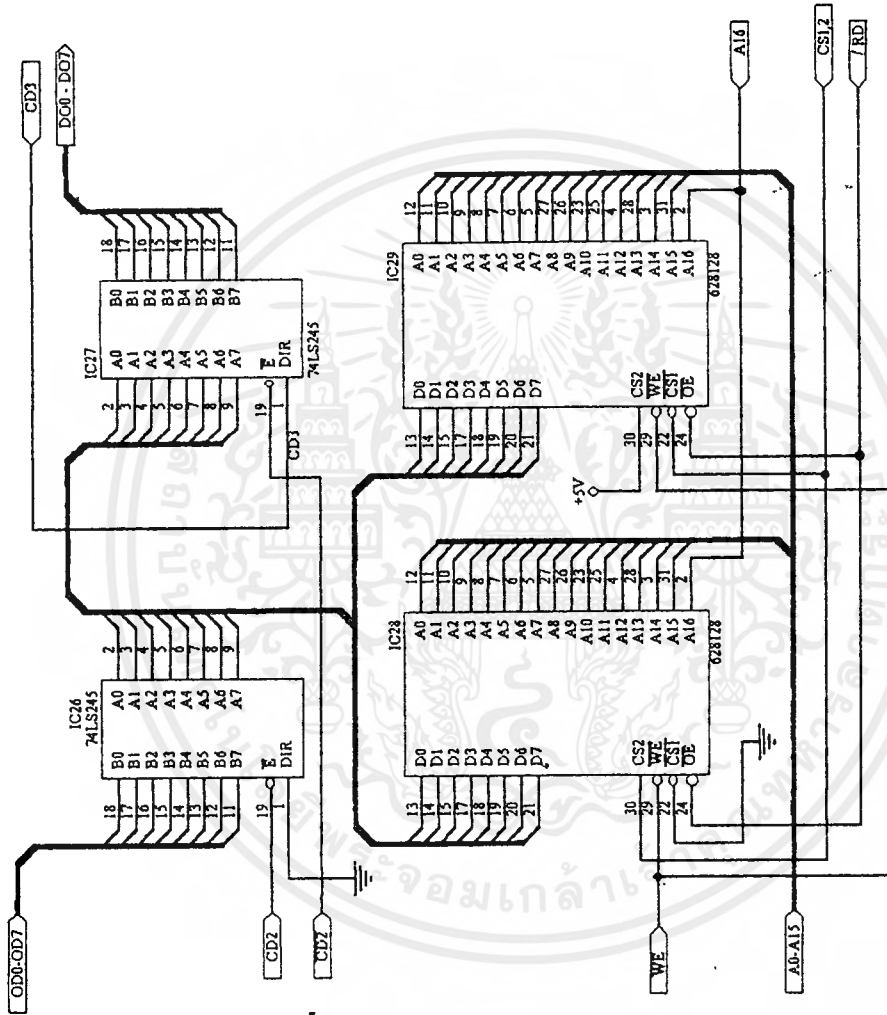
ถ้าเราต้องการให้ IC29 ทำงานจะต้องให้สัญญาณควบคุมการเลือกหน่วยความจำเป็น "0" แต่ถ้าสัญญาณควบคุมนี้เป็น "1" จะเป็นการกำหนดให้ IC28 ทำงานแทน จะเห็นว่า IC28, IC29 ไม่สามารถทำงานพร้อมกันได้และต้องทำงานสลับกันตลอดเวลา เมื่อตัวใดหยุดทำงานขาสัญญาณอินพุต / เอาต์พุตของตัวนั้นจะเป็นไฮอิมพีแดนซ์ ในกรณีที่ต้องการเขียนหรืออ่านข้อมูลสู่หรือจากหน่วยความจำภาพนั้นทำได้โดยควบคุมสัญญาณ /WE, /OE (ซึ่งจะเหมือนกันทั้ง IC28, IC29) คือถ้าต้องการอ่านข้อมูลสัญญาณ /WE ต้องเป็น "1" และ /OE ต้องเป็น "0" แต่ถ้าต้องการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลภาพ /WE ต้องเป็น “0” และ สัญญาณ /OE ต้องเป็น “1” การอ่านหรือการเขียนข้อมูลนั้นจะต้องทำร่วมกับการกำหนดตำแหน่งของข้อมูลบนหน่วยความจำภาพซึ่งก็คือ การกำหนดค่าแอดเดรสนั่นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงวงจรมหาหน่วยความจำภาพ

4.9 วงจรและการทำงานของภาคควบคุมและการอินเตอร์เฟส

การทำงานของภาคนี้จะอยู่ที่ IC31 ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล MCS - 51 เบอร์ 89C51 ซึ่งเป็นไอซีที่มีหน่วยความจำภายในสามารถใช้งานได้สะดวก ในการพัฒนาระบบสามารถลบและเขียนโปรแกรมได้ 1000 ครั้ง การทำงานจะเริ่มจาก IC31 จะทำการตรวจเช็คลอจิก "0" ที่ขา 6 (P1.5) โดยจะตรวจจับระดับลอจิก "0" ถ้าเกิดระดับลอจิกไม่เท่ากับ "0" ไมโครคอนโทรลเลอร์จะรอรับจนมีค่าเป็น "0" และเมื่อไอซีที่ระดับลอจิกที่ ขา 6 มีค่าเป็น "0" IC31 ก็จะส่งสัญญาณควบคุมออกมาที่ P1.0 - P1.4 ภายใต้อินพุต CD1 - CD4 มาทำการสั่งให้ระบบทำงานซึ่งแสดงได้ดังตารางต่อไปนี้

การทำงาน	CD1	CD2	CD3	CD4
เก็บข้อมูลเข้าหน่วยความจำ	0	0	0	1
อ่านข้อมูลฟลัดจ์	0	1	1	0
อ่านข้อมูลฟลัดจ์	1	1	1	0

ตารางที่ 4.1 แสดงสัญญาณควบคุมและการทำงาน

ในการเขียนข้อมูลภาพ IC31 จะไม่ทำการเขียนโดยตรงแต่จะควบคุมให้วงจรภายนอกเขียน ซึ่งก็คือวงจรควบคุมการเขียนข้อมูลภาพในช่วงของการอ่านข้อมูล IC31 จะทำการอ้างแอดเดรสไบต์ค่าออกที่ P0.0 - P0.7 (ขา 32 - 39) แล้วทำการแลทไว้โดย IC32 จากนั้นก็จะทำการอ้างแอดเดรสไบต์สูงออกที่ P2.0 - P2.7 จากนั้นขา 17 (RD) ถูกสั่งให้เอาคิฟ ข้อมูลก็จะถูกดึงออกจากหน่วยความจำรอตที่ IC32 ซึ่งทำหน้าที่เป็นตัวแลทข้อมูลไว้โดยไมโครคอนโทรลเลอร์จะเป็นตัวส่งแลท จากนั้นไมโครคอนโทรลเลอร์จะส่งระดับลอจิก "0" ออกจากขา P1.0 เพื่อบอกให้คอมพิวเตอร์อ่านข้อมูลเข้าทางพอร์ทพรีนเตอร์ที่ขา 10, 11, 12, 13 ซึ่งระบบการอินเตอร์เฟสทางพรีนเตอร์พอร์ทได้ออกแบบเอาไว้ในระบบคอมพิวเตอร์ ฉะนั้นทางคอมพิวเตอร์จะต้องอ่านข้อมูล 2 ครั้ง ๆ ละ 4 บิต โดยส่งสัญญาณมาควบคุมที่ขา 1 และ 19 ของ IC 34 เมื่ออ่านครบ 8 บิตแล้วคอมพิวเตอร์จะส่งสัญญาณลอจิก "0" มาที่ขา P1.7 ของไมโครคอนโทรลเลอร์เพื่อให้ส่งตำแหน่งต่อไปจนครบ 256 กิโลไบต์ เนื่องจากในการอ้างตำแหน่งต้องทำถึง 17 เส้น แต่การอ้างของ IC31 สามารถทำได้เพียง 16 เส้นจึงต้องเซ็ทบิต P1.6 อีก 1 บิตเพื่อที่จะได้ครบ 17 เส้น โดยผ่านทาง IC30 ในการที่จะอ่านฟลัดจ์หรือจะขึ้นอยู่กับ CD1 ดังตาราง โดยที่ถ้าค่า CD1 มีค่าเป็น "0" หน่วยความจำฟลัดจ์จะถูกอ่าน เมื่อ CD1 มีค่าเป็น "1" หน่วยความจำฟลัดจ์จะถูกอ่าน SW1 ใช้ในการรีเซ็ท 89C51

4.10 การเขียนโปรแกรม

การทำงานของระบบทั้งหมดถูกควบคุมด้วยโปรแกรมภาษาแอสเซมบลีของคอนโทรลเลอร์ และโปรแกรมภาษาซีในการแสดงผลบนคอมพิวเตอร์ ดังนั้นโปรแกรมจึงแบ่งเป็น 2 ส่วนดังนี้

แอสเซมบลี 89C51

การทำงานจะเริ่มจากการตรวจสอบสัญญาณจากคอมพิวเตอร์ "0" ที่ขา P1.5 ของ 89C51 เพื่อเริ่มการเก็บข้อมูล หลังจากนั้นจะทำการส่งคำสั่ง ไปให้วงจรควบคุมการเขียนข้อมูลทำการเขียนข้อมูลลงหน่วยความจำโดยชุดของคำสั่งที่ 89C51 ส่งออกมาจะส่งออกมาทาง P1.1-P1.4 โดยเมื่อเขียนข้อมูลลงหน่วยความจำเสร็จก็จะทำการหน่วงเวลาไว้ประมาณ 120 ns เพื่อให้การเขียนข้อมูลเสร็จอย่างสมบูรณ์ หลังจากนั้นจะส่งคำสั่งในการอ่านข้อมูลจากหน่วยความจำเพื่อทำการส่งข้อมูลให้คอมพิวเตอร์ โดยครั้งแรกจะอ่านข้อมูลจากหน่วยความจำพีดส์ก่อนแล้วอ่านข้อมูลจากหน่วยความจำพีดส์คู่ ข้อมูลแต่ละตำแหน่งในหน่วยความจำจะถูกอ่านออกมาโดยการชี้ของ Register DPTR ภายในของ 89C51 ซึ่ง DPTR เป็น Register ขนาด 16 บิตดังนั้นโดยลำพังแล้วจะอ้างตำแหน่งข้อมูลได้ FFFF H ตำแหน่ง หรือประมาณ 64 กิโลไบต์แต่ข้อมูลในหน่วยความจำแต่ละตัวมีขนาด 128 กิโลไบต์ดังนั้นเพื่อสามารถอ่านข้อมูลครบ 128 กิโลไบต์จึงต้องเซ็ทบิตของ 89C51 อีกหนึ่งบิตก็จะได้เป็น 1FFFF H ก็สามารถอ้างข้อมูลได้ครบ 128 กิโลไบต์เมื่ออ่านข้อมูลออกมาแล้วข้อมูลจะถูกแลตซ์เอาไว้เพื่อรอให้คอมพิวเตอร์ทำการอ่านข้อมูลเข้าไปจนครบทั้งพีดส์และพีดส์คู่ เมื่อทำการส่งข้อมูลให้คอมพิวเตอร์เสร็จก็จะกลับไปตรวจสอบสัญญาณที่ขา P1.5 ของ 89C51 เพื่อเริ่มการทำงานรอบใหม่

ภาษาซี

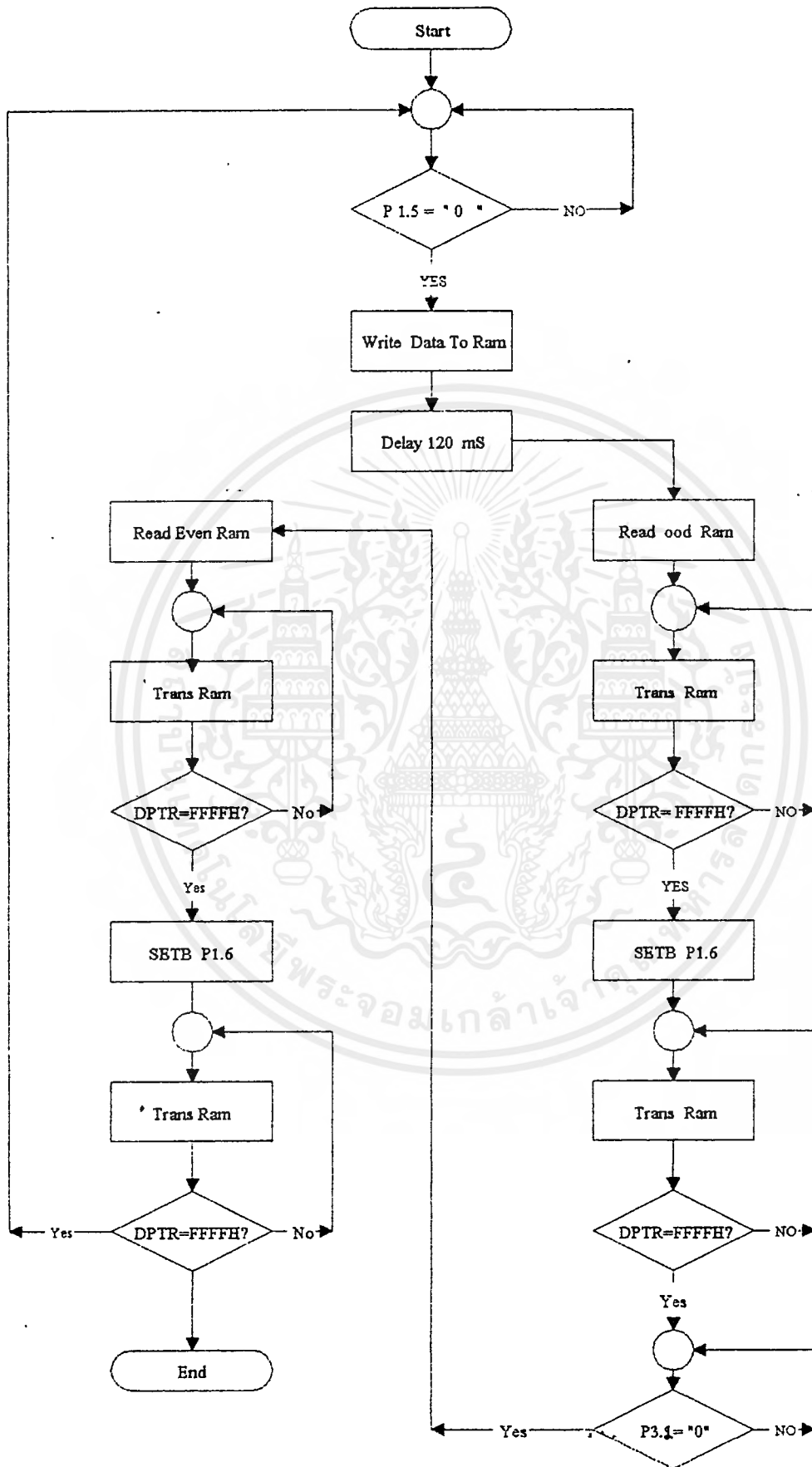
การทำงานจะเริ่มจากการจองพื้นที่ขนาด 512×512 จุดเพื่อเอาจุดภาพเข้ามาเก็บ หลังจากนั้นก็จะทำการส่งสัญญาณไปให้คอนโทรลเลอร์ (89C51) เริ่มทำงานแล้วก็จะรอสัญญาณให้เริ่มอ่านข้อมูลจาก 89C51 เมื่อได้รับสัญญาณแล้วก็จะอ่านข้อมูลเข้าที่พอร์ทพรีนเตอร์ที่หมายเลข 389 H โดยอ่านเข้ามาครั้งละ 4 บิตดังนั้นโปรแกรมภาษาซีจึงต้องเขียนให้อ่านข้อมูล 2 ครั้งในหนึ่งไบต์ โดยครั้งแรกจะทำการอ่าน 4 บิตต่ำก่อนแล้วครั้งที่ 2 จะทำการอ่าน 4 บิตสูงแล้วจึงนำข้อมูลทั้งสองชุดมาทำการออร์กันก็จะได้ข้อมูลครบ 8 บิตเมื่ออ่านพีดส์แรกคือพีดส์คู่เสร็จก็จะส่งสัญญาณไปบอกให้คอนโทรลเลอร์ทำการอ่านพีดส์ที่สองคือพีดส์คู่โดยกระบวนการในการนำข้อมูลเข้าเครื่องคอมพิวเตอร์เหมือนกับในกระบวนการของพีดส์คู่ทุกอย่าง ข้อมูลที่อ่านจากหน่วยความจำจะเก็บลงพื้นที่ที่จองไว้ครั้งละหนึ่งเส้นภาพ (512 ไบต์) ในลักษณะการสลับเส้นตามลักษณะพีดส์และพีดส์คู่

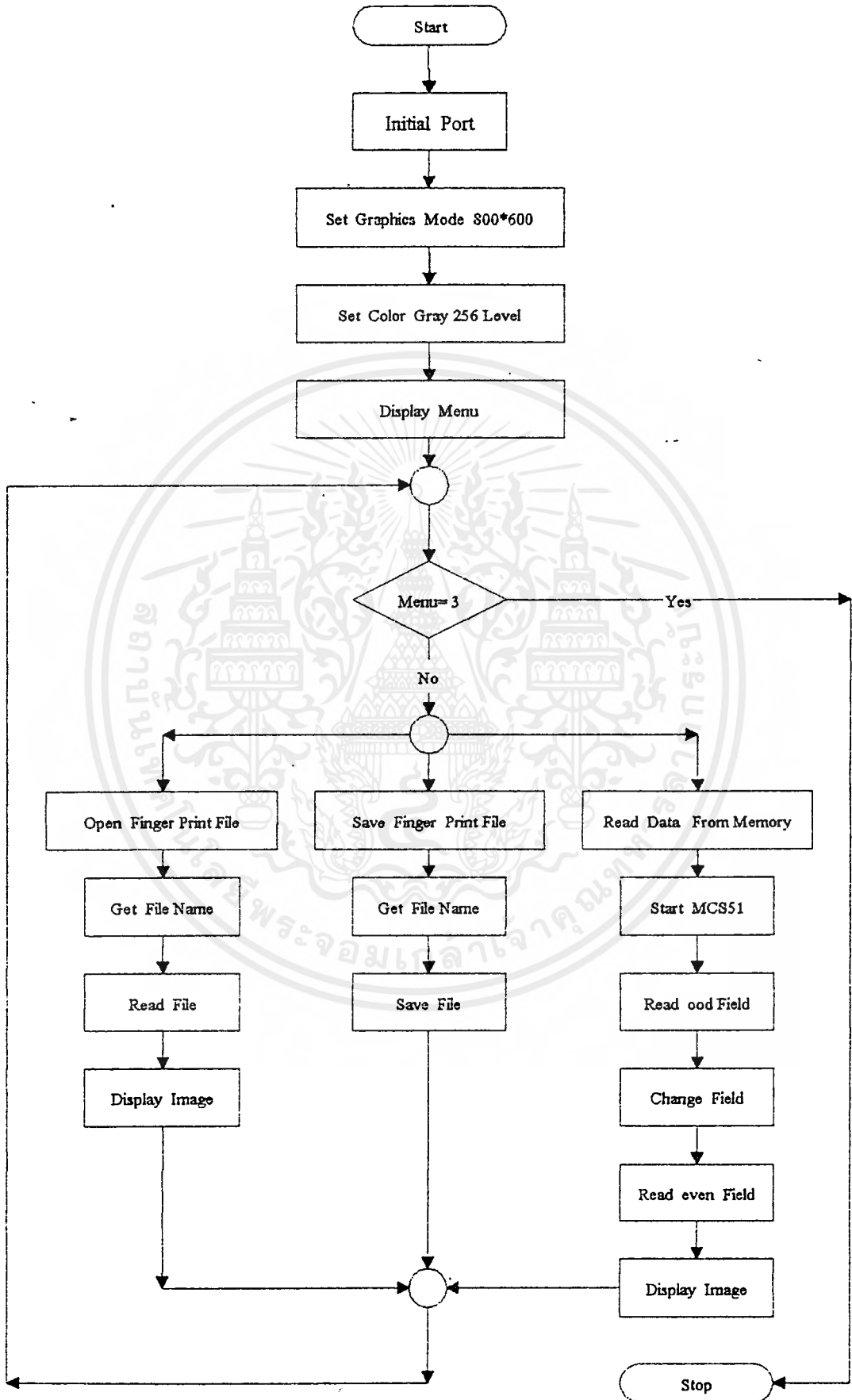
ในส่วนของการแสดงภาพ จะกำหนดให้แสดงในโหมดภาพ 800×600 256 สี ซึ่งเป็น

โหมดของภาพกราฟฟิคที่มีอยู่ในการ์ดแสดงผลซูเปอร์วีจีเอ (Super VGA) มีลักษณะการแสดงสีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆ เป็นการผสมแม่สีเข้าด้วยกัน คือ สีแดง สีเขียว สีน้ำเงิน เนื่องจากข้อมูลที่เก็บไว้ในบัฟเฟอร์ มีขนาด 8 บิต ซึ่งจะทำให้มีความละเอียด 256 ดังนั้นจึงจำเป็นต้องทำการแสดงออกทางหน้าจอ คอมพิวเตอร์ 256 ระดับด้วยโดยการใช้ Driver SVGA256 เข้ามาช่วยในการแสดงระดับเกรย์ 256 ระดับ ในโหมดของ 800×600 เพื่อให้สามารถแสดงภาพขนาด 512×512 จุดภาพได้ ซึ่งถ้าเซ็ทค่า โหมดให้ต่ำกว่านี้จะทำให้ไม่สามารถแสดงรายละเอียดได้ครบ ซึ่งการที่จะสามารถแสดงระดับเกรย์ ให้ได้ 256 ระดับเราจำเป็นต้องทำการปรับตารางสีให้เป็นระดับเกรย์ 256 ระดับ โดยการเข้าไปเซ็ท ค่าในรีจิสเตอร์ที่เก็บค่าแม่สี R G B ไว้ในอัตราส่วนที่เหมาะสมโดยข้อมูลที่ปรับระดับแล้วจะถูก Plot ลงบนหน้าจอคอมพิวเตอร์ตามตำแหน่งที่กำหนดโดยคำสั่ง Put Pixel

การเก็บข้อมูลภาพจะเก็บในลักษณะไบต์อาร์เรย์ (byte array) ซึ่งเป็นการเก็บในลักษณะ เรียงข้อมูลที่ละตำแหน่งจากตำแหน่งเริ่มต้น ที่ด้านบนซ้ายของภาพจนถึงตำแหน่งสุดท้ายที่ด้านล่าง ขวาของภาพ การเก็บข้อมูลภาพสามารถเก็บลงฮาร์ดดิสต์หรือเก็บลงบนแผ่นดิสต์และสามารถ กำหนด PART ได้ตามความต้องการ





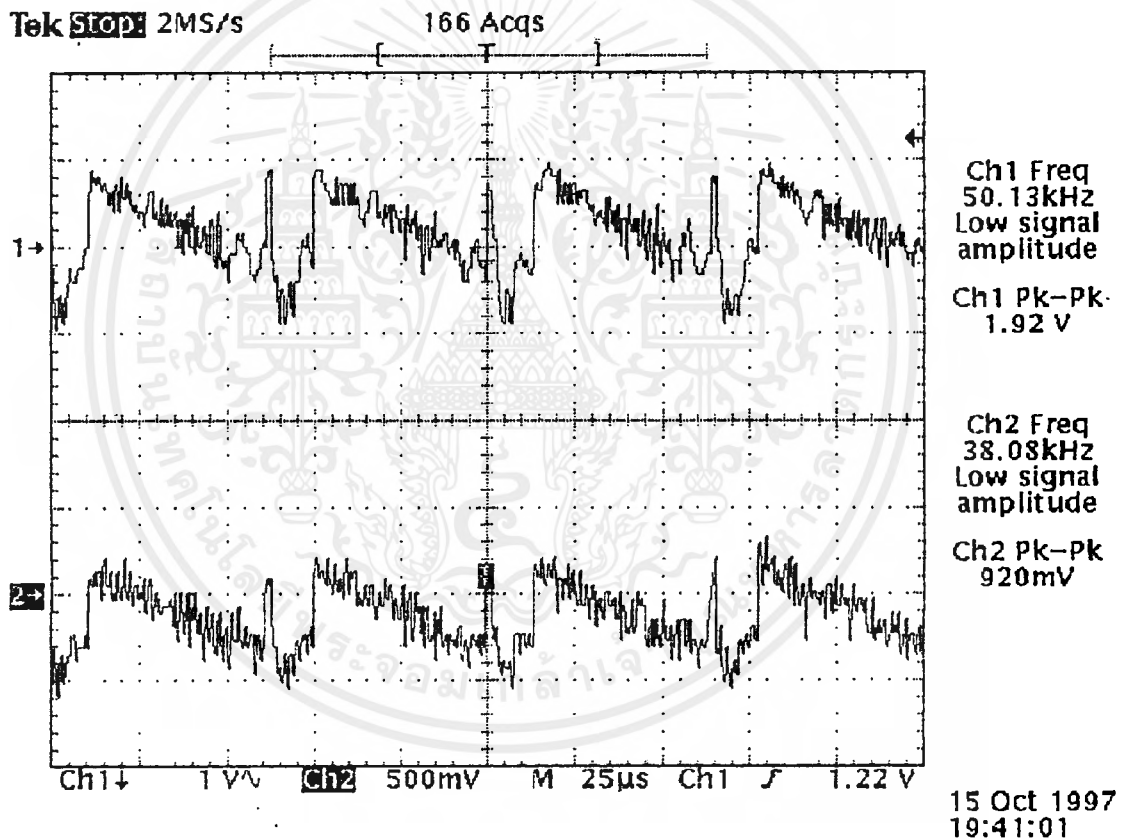
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรของคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.12 แสดงโฟลวชาร์ตของโปรแกรมภาษาซี
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

5.1 การทดลองภาค A/D และ D/A

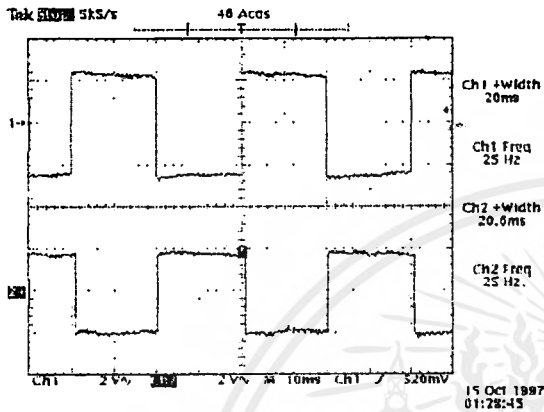
การทดลองในส่วนนี้จะทำการป้อนสัญญาณภาพเข้าที่ภาคอินพุทเพื่อทำการแปลงเป็นสัญญาณดิจิทัลแล้วทำการแปลงกลับเป็นสัญญาณอนาลอก เพื่อที่จะตรวจสอบความถูกต้องซึ่งรูปคลื่นที่ได้เป็นดังรูปที่ 5.1



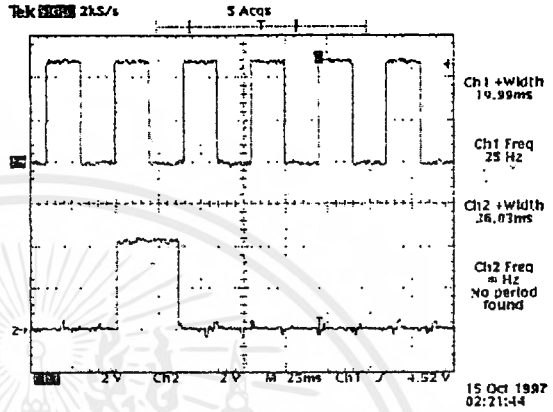
รูปที่ 5.1 แสดงรูปสัญญาณอินพุท (CH2) ที่ป้อนกับสัญญาณแอมป์พุท (CH1) จากภาค D/A

5.2 การทดสอบภาคควบคุมการเขียนข้อมูลภาพ

ในการวัดรูปคลื่นจะทำการป้อนสัญญาณควบคุมจากภาคแยกซิงค์ และทำการกำหนดสัญญาณควบคุมการเขียน (CD4) ให้วงจรข้อมูลภาพ ซึ่งสามารถแสดงได้ดังรูปที่ 5.2



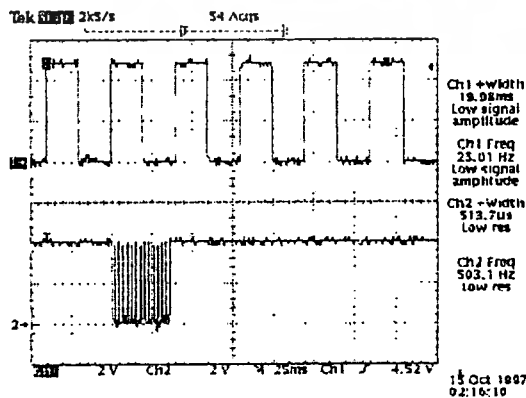
(ก) แสดงสัญญาณ ODD/EVEN (CH2)



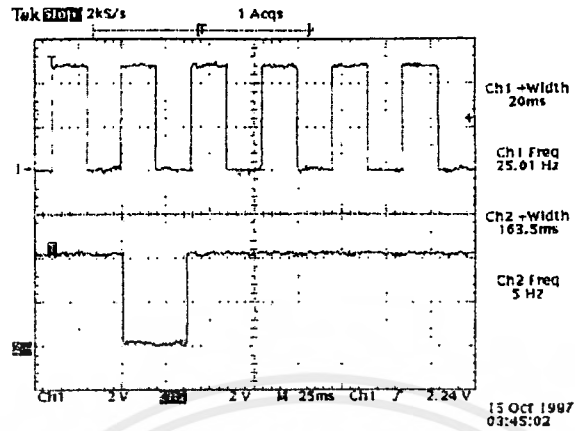
(ข) แสดงสัญญาณขา 3 IC 25:A (CH2)



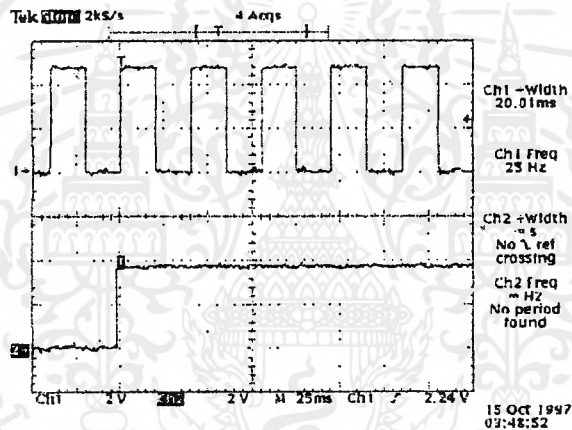
(ค) แสดงสัญญาณ CR2 (CH2)



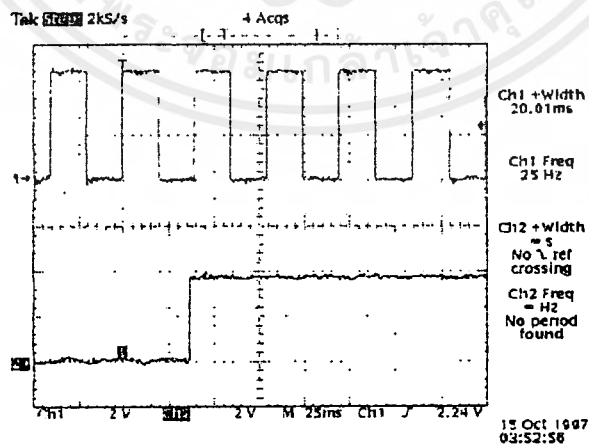
(ง) แสดงสัญญาณ WR (CH2)



(จ) แสดงสัญญาณ ขา 3 IC 14:A (CH2)



(ฉ) แสดงสัญญาณขา 1 IC 25:A (CH2)



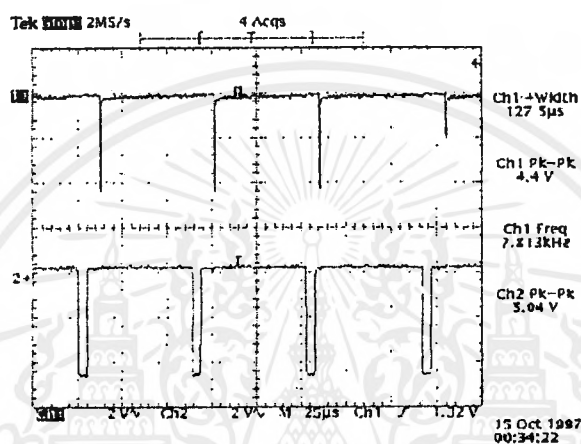
(ช) แสดงสัญญาณขา 2 IC 25 :A (CH2)

รูปที่ 5.2 แสดงสัญญาณที่ภาคควบคุมการเขียนข้อมูลภาพ

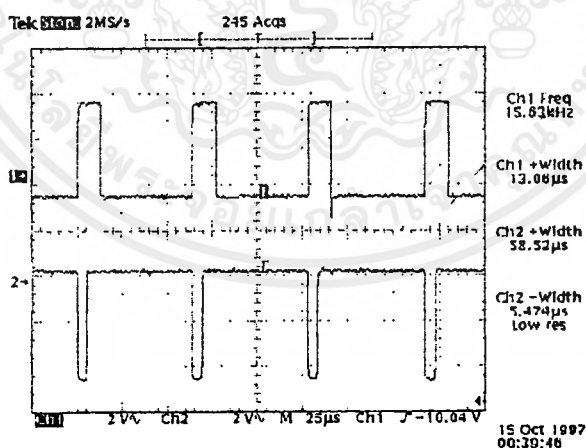
5.3 การทดสอบภาคแยกซิงค์

สามารถทำได้โดยการป้อนสัญญาณภาพให้กับไอซี LM 1881 แล้วทำการวัดสัญญาณซิงค์รวมที่ขา 1 และสัญญาณซิงค์ทางแนวตั้งที่ ขา 3 และสัญญาณที่ใช้ในการควบคุมการนับแสดงดังรูปที่

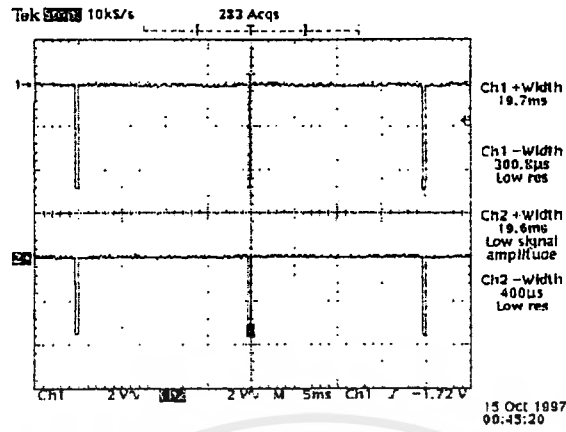
5.3



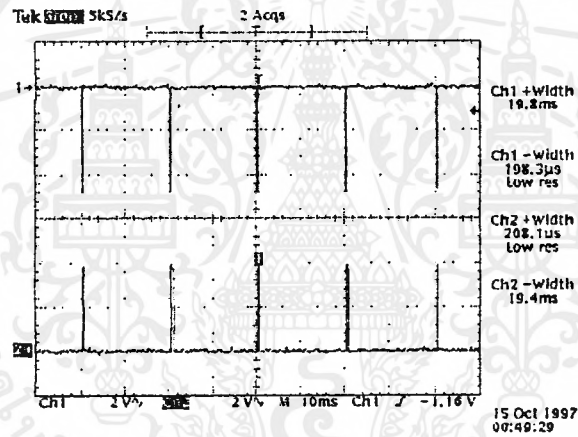
(ก) H - Sync Count (CH1) , Composite Sync (CH2)



(ข) สัญญาณขา 5 IC 11 :B (CH1)



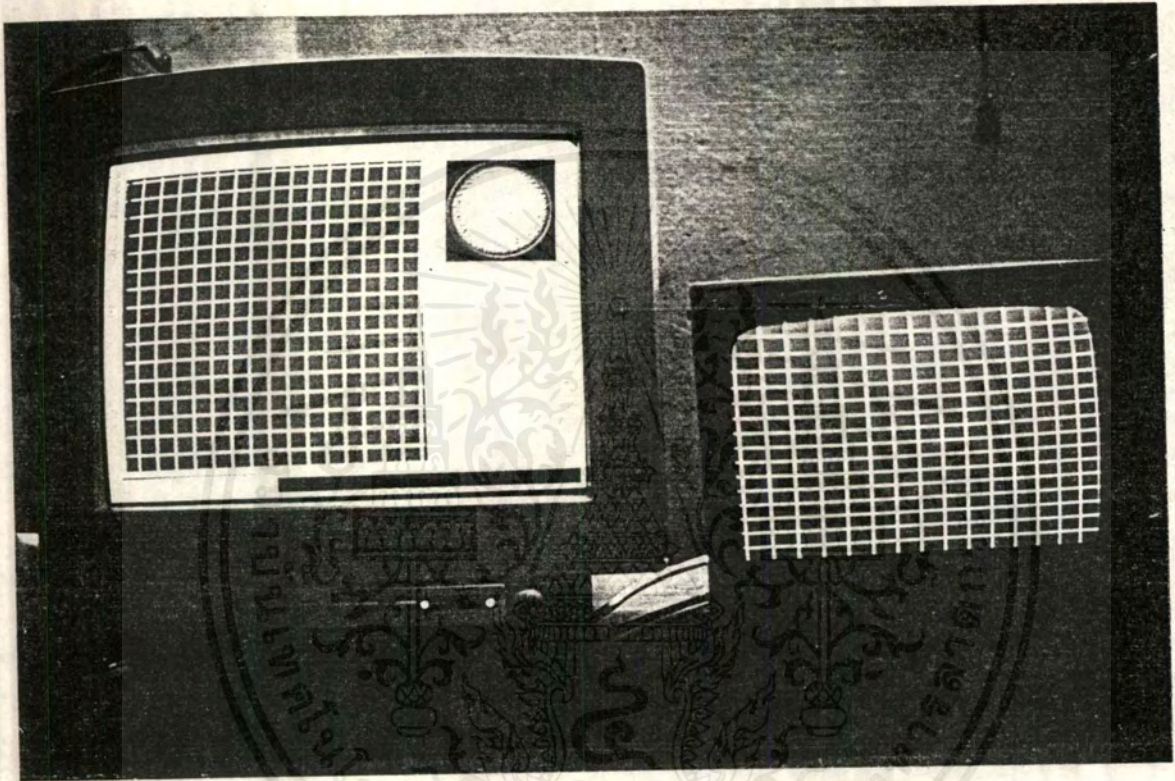
(ก) V - Sync (CH 1) , สัญญาณขา 9 IC 9:B (CH 2)



(ง) สัญญาณขา 1 IC 13 :A (CH 2)

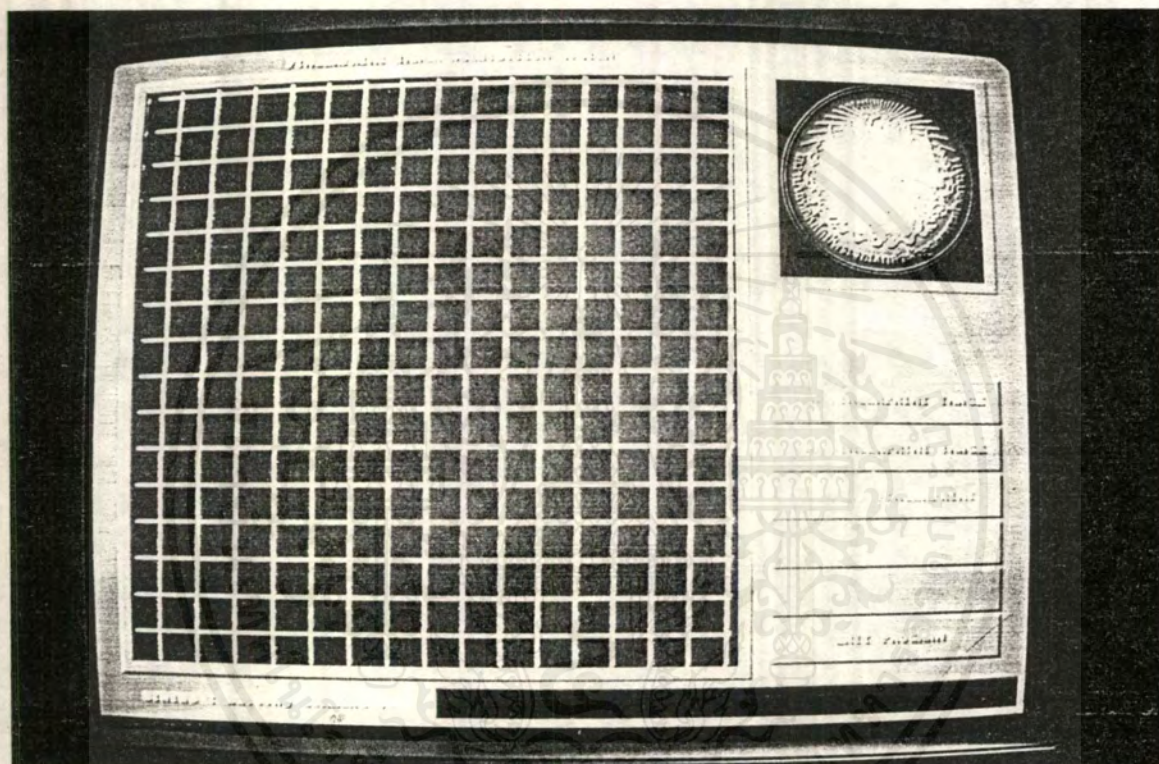
รูปที่ 5.3 สัญญาณทดสอบที่ภาคแยกซิงค์

5.4 การทดสอบโดยการป้อนสัญญาณจาก Pettern Generator



รูปที่ 5.4 แสดงรูปที่ได้จากจอมอนิเตอร์เทียบกับคอมพิวเตอร์

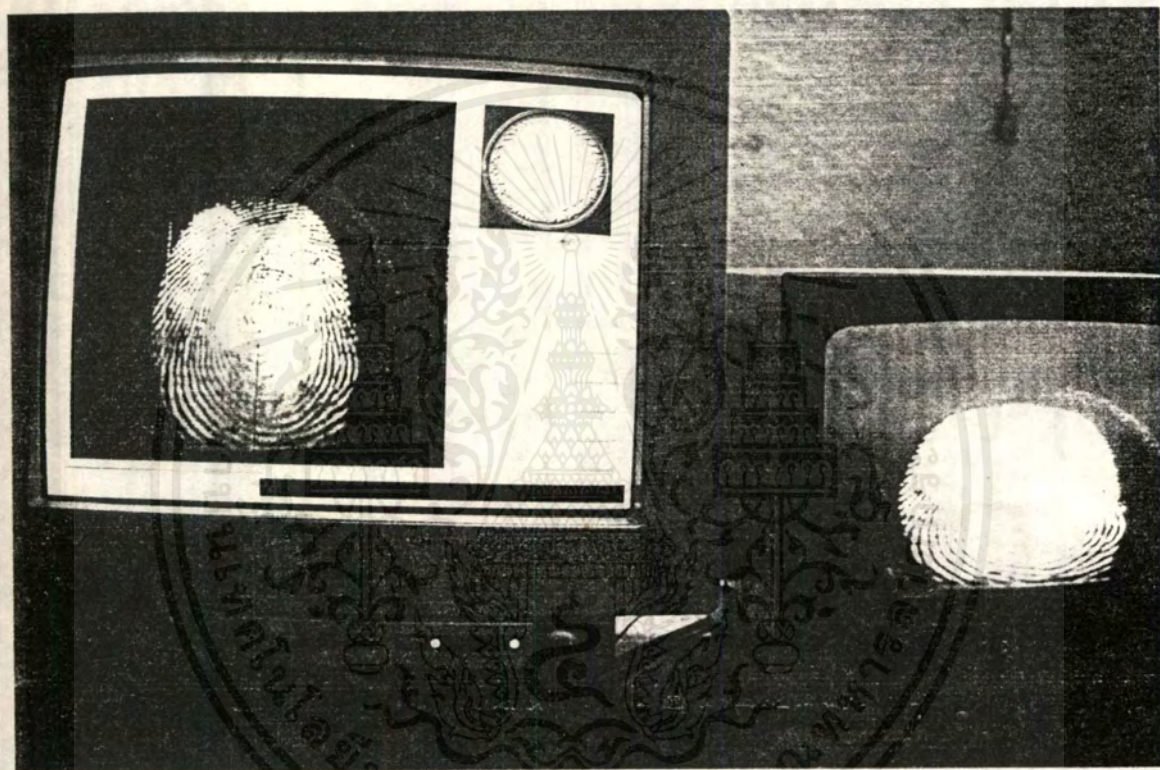
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงรูปที่ได้จากจอคอมพิวเตอร์

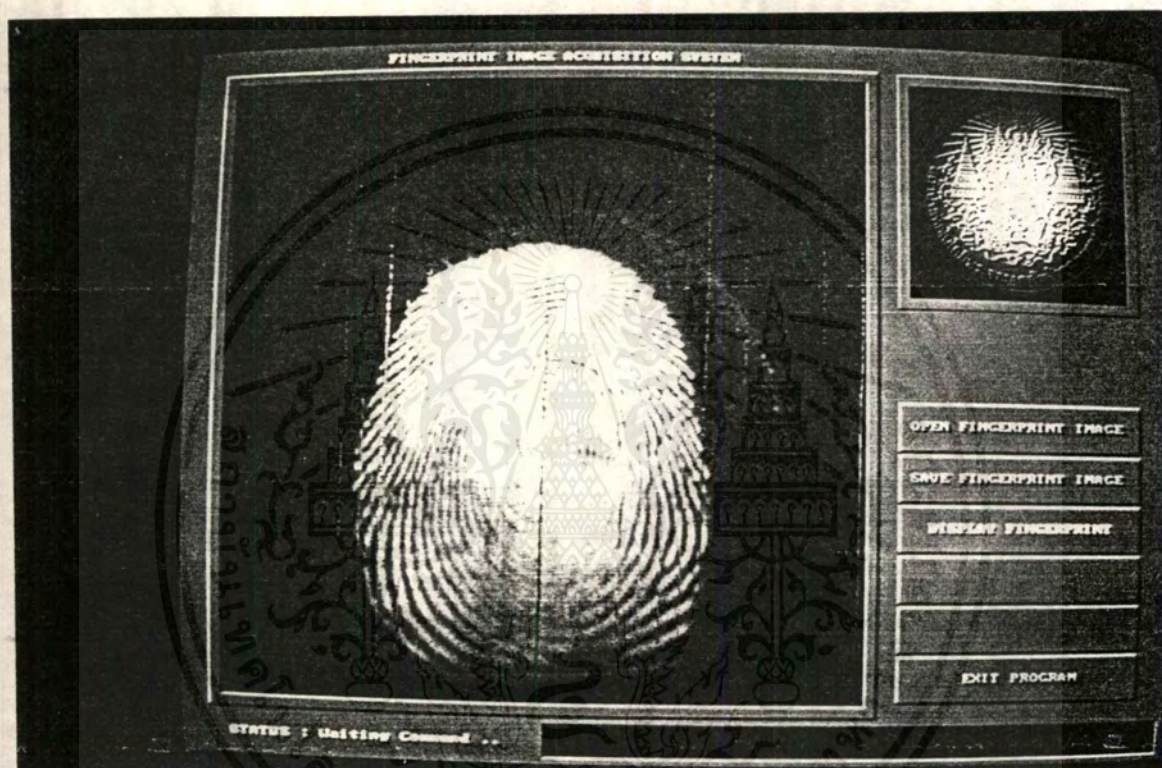
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การทดสอบโดยการป้อนสัญญาณจากกล้อง ซีซีดี



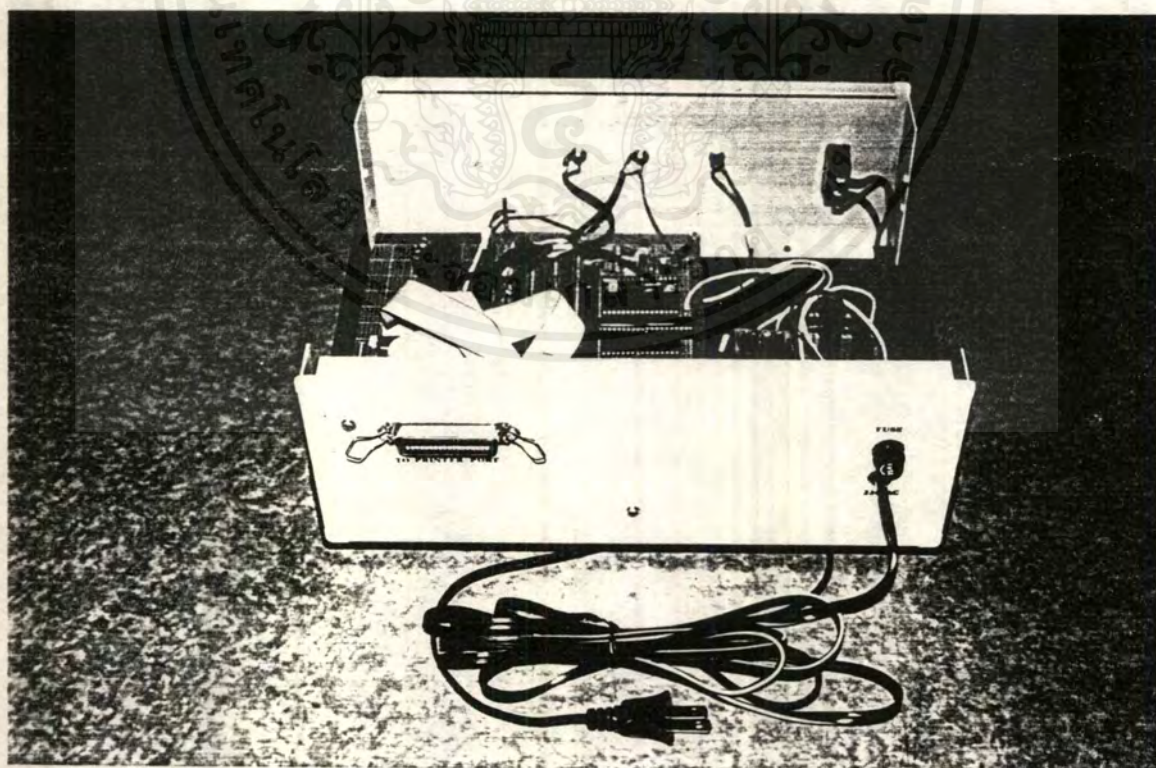
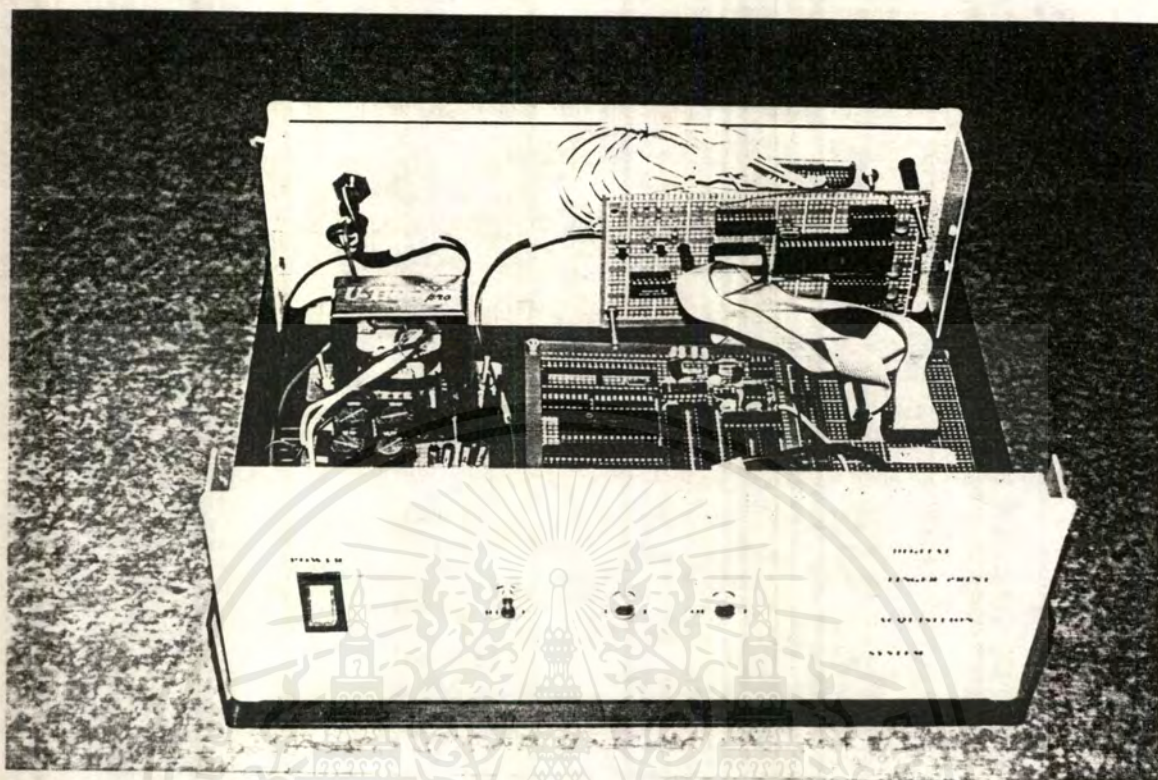
รูปที่ 5.6 แสดงรูปที่ได้จากมอนิเตอร์เทียบกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงรูปที่ได้จากจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 5.8 ไม่แสดงตัวเครื่องด้านหน้าและด้านหลังให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

จากการทดลองตลอดระยะเวลา 2 ภาคการเรียนซึ่งพอจะสรุปงานได้ดังนี้ ในส่วนของภาค A/D นั้นเราใช้ IC สำเร็จรูปเบอร์ TDA 8708A ในการแปลงข้อมูลภาพทางอนาลอกให้เป็นสัญญาณทางดิจิทัล ซึ่งจะให้อัตราข้อมูลได้ 8 บิต ซึ่งทำให้ระดับแอมพลิจูดมีความแตกต่าง 256 ระดับ สัญญาณที่ได้ออกจากภาค A/D จะถูกป้อนให้กับภาค D/A เพื่อทำการแปลงสัญญาณกลับไปแสดงที่ มอนิเตอร์โดยใช้จอคอมพิวเตอร์ซึ่งจากการทดลองในการกำหนดอัตราการขยายต้องมีความเหมาะสม และต้องทำงานถึงสัญญาณรบกวนที่เกิดขึ้นจากการต่อกราวด์ ซึ่งมีผลต่อสัญญาณทางแอมพลิจูด

ในส่วนของภาคแยกซึ่งก็เมื่อเราทำการป้อนสัญญาณภาพเข้าไปโดยไอซีเบอร์ LM 1881 ซึ่งเป็นไอซีสำเร็จรูปจะทำการแยกสัญญาณที่เป็นองค์ประกอบของสัญญาณภาพ ออกมาได้เป็น สัญญาณ COMPOSITE SYNC ที่ขา 1 ,สัญญาณ VERTICAL SYNC ที่ขา 3 และ สัญญาณ ODD / EVEN ที่ขา 7 ซึ่งสัญญาณ จะมีการเปลี่ยนแปลงตามรูปแบบของสัญญาณที่ป้อนเข้าไป และสัญญาณเหล่านี้จะถูกนำไปใช้ในการควบคุมการเขียนข้อมูลภาพ และควบคุมวงจรมัน

ในส่วนของวงจรมันสัญญาณสัญญาณนาฬิกาที่ใช้ในการทำงานของระบบก็สามารถสร้าง ได้ที่ความถี่ 10 MHZ

ในส่วนของวงจรมันซึ่งต้องใช้ในการอ้างตำแหน่งหน่วยความจำขณะทำการเก็บข้อมูลจาก ภาค A/D สามารถสร้างได้โดยใช้ไอซีไบนารีอะซิงโครนัสเคานเตอร์ขนาด 4 บิต รวมกับ D-Flip Flop เพื่อให้ได้ 17 บิต สามารถอ้างหน่วยความจำได้ 256 กิโลไบต์ ซึ่งจะใช้หน่วยความจำขนาด 128 กิโลไบต์ จำนวน 2 ตัวในการจัดเก็บข้อมูลฟิลด์คู่และฟิลด์คี่

ส่วนของภาคควบคุมการเขียนข้อมูลภาพและการอินเทอร์เฟสจะนำเอาสัญญาณจากภาคแยก ซึ่งก็และภาคควบคุมการมัน มาสร้างสัญญาณในการเขียนข้อมูลภาพภายใต้การควบคุมของไมโคร คอนโทรลเลอร์ 89C51 โดยการเขียนโปรแกรมควบคุมการเขียนและอ่านข้อมูลจากหน่วยความจำ เพื่อส่งข้อมูลให้กับคอมพิวเตอร์ผ่านทางพริ้นเตอร์พอร์ทโดยจะทำการส่งข้อมูลครั้งละ 4 บิต ไปยัง คอมพิวเตอร์เพื่อทำการแสดงผลภายใต้โปรแกรมภาษาซี

ปัญหาและอุปสรรค

1. ในส่วนของภาค A/D จากการทดลองปัญหาที่พบคือ เมื่อป้อนสัญญาณอินพุตที่ขา 16 ตามคู่มือแล้วจะเกิดการเพี้ยนของสัญญาณไม่มีสัญญาณดิจิทัลออกมา ซึ่งการแก้ไขเบื้องต้นคือป้อน อินพุตที่ขา 20 โดยไม่ผ่าน ภาค VIDEO AMP ภายในไอซี โดยการจัดการไบอัสที่ขา 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยได้ทำการเพิ่มชุดการปรับสัญญาณรบกวนเข้าไปด้วยปรากฏว่าสามารถทำการแปลงสัญญาณแสดงออกที่มอนิเตอร์ได้และสามารถปรับความเข้มที่หน้าจอได้ด้วย

2. ส่วนของโปรแกรมทั้ง ภาษาซี และ แอสเซมบลีเสร็จเมื่อนำทั้งสองส่วนมาทดสอบพบว่า ครั้งแรกเมื่อทดลองป้อนสัญญาณจาก Pattern Generator พบว่ารูปที่ออกมมอนิเตอร์ค่อนข้างสมบูรณ์แต่พอออกที่จอคอมพิวเตอร์ภาพเกิดลึ้ม ซึ่งแก้ไขเบื้องต้นคือการตรวจโปรแกรมส่วนของภาษาซีพบว่ามีความผิดพลาดตรงส่วนของตำแหน่งเริ่มต้นของการอ่านข้อมูลเมื่อทำการแก้แล้วรูปที่ได้จึงปรกติ

3. อีกปัญหาคือแต่ละรูปแบบพบว่าบางรูปแบบเมื่อแสดงออกมมอนิเตอร์รูปที่ได้ค่อนข้างสมบูรณ์แต่พอแสดงบนจอคอมพิวเตอร์ภาพที่ได้ไม่ค่อยสมบูรณ์เท่าที่ควรเช่นรูป Gray Scale และเมื่อทำการป้อนสัญญาณลายนิ้วมือที่ได้จากกล้อง ซีซีดี พบว่ามีลักษณะผลที่ได้คล้ายกับ Gray Scale ก็จะมีเส้นที่ไม่ต้องการปนออกมาในขณะที่รูปแบบอื่นจาก Pattern Generator เช่น รูปสี่ขาว รูปสี่ดำ รูปลายเส้นไม่พบปัญหาดังกล่าวเมื่อแสดงออกจอคอมพิวเตอร์ และเมื่อทำการลงกล่องแล้วทำการทดลองดูอีกครั้งพบว่าเมื่อบัญญาณเรบกานมีลักษณะเป็นเส้นจัน ซึ่งแนวทาง ในการแก้ไขจะต้องทำการออกแบบวงจร ให้ดีและการลงอุปกรณ์ที่ดี

แนวทางในการปรับปรุงและพัฒนา

เนื่องจากโครงการที่จัดทำนี้ได้ประสบความสำเร็จมา ไครคอลลโทรลเลอร์มาใช้และระบบการอินเทอร์เฟซสามารถใช้งานได้แล้วดังนั้นแนวทางที่จะพัฒนาต่อไปก็อาจจะให้มีการทำงานที่มีการตรวจเช็คอัตโนมัติ โดยอาจเมื่อมีการการวางนิ้วมือระบบสามารถทำงานได้ทันทีเป็นต้น ตลอดจนจะต้องมีการออกแบบวงจรบางส่วนใหม่เพื่อให้มีการรบกวนน้อยที่สุดเพราะจากการทดลองที่ผ่านมาผลที่ได้ยังไม่ดีเท่าที่ควร

วิธีการใช้งานโปรแกรม

ในส่วนของคอมพิวเตอร์ภายใต้โปรแกรมภาษาซี เมื่อเริ่มต้นการทำงานจะแสดงโลโก้ของสถาบัน กรอบหน้าต่างในการแสดงรูป และ เมนูในการใช้งานซึ่งประกอบไปด้วย

OPEN FINGERPRINT IMAGE จะทำหน้าที่ในการเปิดแฟ้มข้อมูลภาพเท่าที่ได้ทำการเก็บเอาไว้ซึ่งอาจจะเก็บอยู่ในฮาร์ดดิสก์ หรือ อาจจะเปิดจากแผ่นดิสก์ทั่วไปที่ใช้สำหรับเก็บข้อมูล โดยจะต้องทำการระบุแหล่งของข้อมูลและพาร์ทให้ถูกต้อง

SAVE FINGERPRINT IMAGE จะทำหน้าที่ในการจัดเก็บข้อมูลที่แสดงอยู่บนหน้าจอคอมพิวเตอร์ลงบนฮาร์ดดิสก์หรืออาจจะเก็บลงบนแผ่นดิสก์ในรูปแบบของไบต์อาร์เรย์ ซึ่งเราจะต้องทำการกำหนดชื่อ และพาร์ทที่จะจัดเก็บตามความเหมาะสม

DISPLAY FINGERPRINT จะทำหน้าที่ในการนำข้อมูลของรูปภาพใหม่มาแสดงที่หน้าจอคอมพิวเตอร์โดยจะทับข้อมูลเดิมทิ้งไป

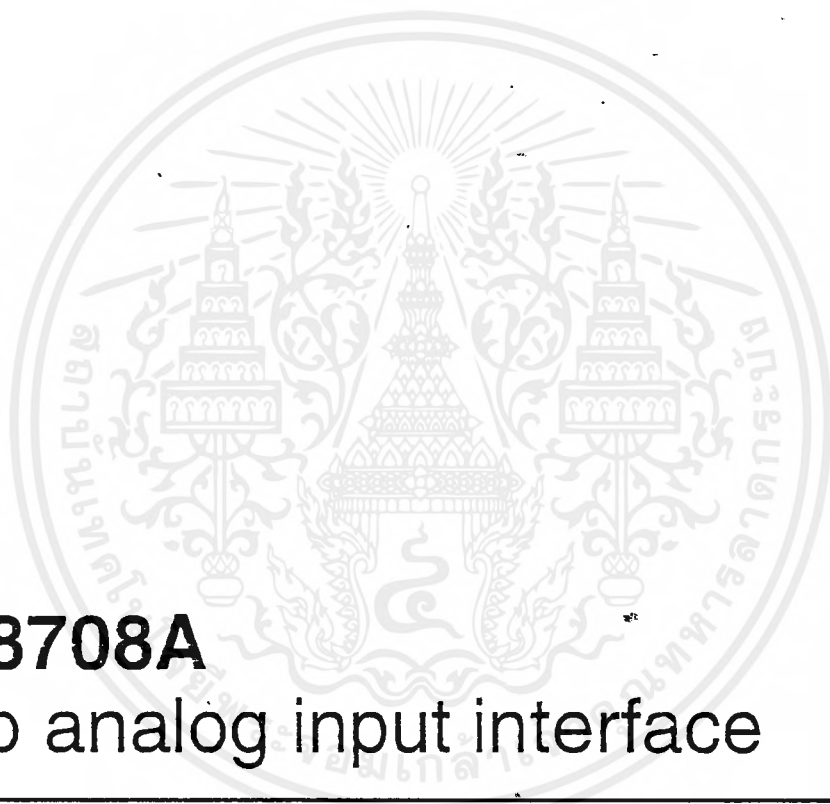
EXIT จะทำหน้าที่ในการออกจากโปรแกรมที่ทำงานอยู่ไปสู่คอส





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET



TDA8708A

Video analog input interface

Product specification

June 1994

Supersedes data of April 1993

File under Integrated Circuits, IC02



PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Video analog input interface

TDA8708A

FEATURES

- 8-bit resolution
- Sampling rate up to 32 MHz
- Binary or two's complement 3-state TTL outputs
- TTL-compatible digital inputs and outputs
- Internal reference voltage regulator
- Power dissipation of 365 mW (typical)
- Input selector circuit (one out of three video inputs)
- Clamp and Automatic Gain Control (AGC) functions for CVBS and Y signals
- No sample-and-hold circuit required.
- The TDA8708A has white peak control in modes 1 and 2 whereas the TDA8708B has control in mode 1 only.

APPLICATIONS

- Video signal decoding
- Scrambled TV (encoding and decoding)
- Digital picture processing
- Frame grabbing.

GENERAL DESCRIPTION

The TDA8708A is an analog input interface for video signal processing. It includes a video amplifier with clamp and gain control, an 8-bit analog-to-digital converter (ADC) with a sampling rate of 32 MHz and an input selector.

QUICK REFERENCE DATA

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
V_{CCA}	analog supply voltage	4.5	5.0	5.5	V
V_{CCD}	digital supply voltage	4.5	5.0	5.5	V
V_{CCO}	TTL output supply voltage	4.2	5.0	5.5	V
I_{CCA}	analog supply current	–	37	45	mA
I_{CCD}	digital supply current	–	24	30	mA
I_{CCO}	TTL output supply current	–	12	16	mA
ILE	DC integral linearity error	–	–	± 1	LSB
DLE	DC differential linearity error	–	–	± 0.5	LSB
$f_{clk(max)}$	maximum clock frequency	30	32	–	MHz
B	maximum –3 dB bandwidth (AGC amplifier)	12	18	–	MHz
P_{tot}	total power dissipation	–	365	500	mW

ORDERING INFORMATION

TYPE NUMBER	PACKAGE			
	PINS	PIN POSITION	MATERIAL	CODE
TDA8708A	28	DIP	plastic	SOT117-1
TDA8708AT	28	SO28L	plastic	SOT136-1

Video analog input interface

TDA8708A

BLOCK DIAGRAM

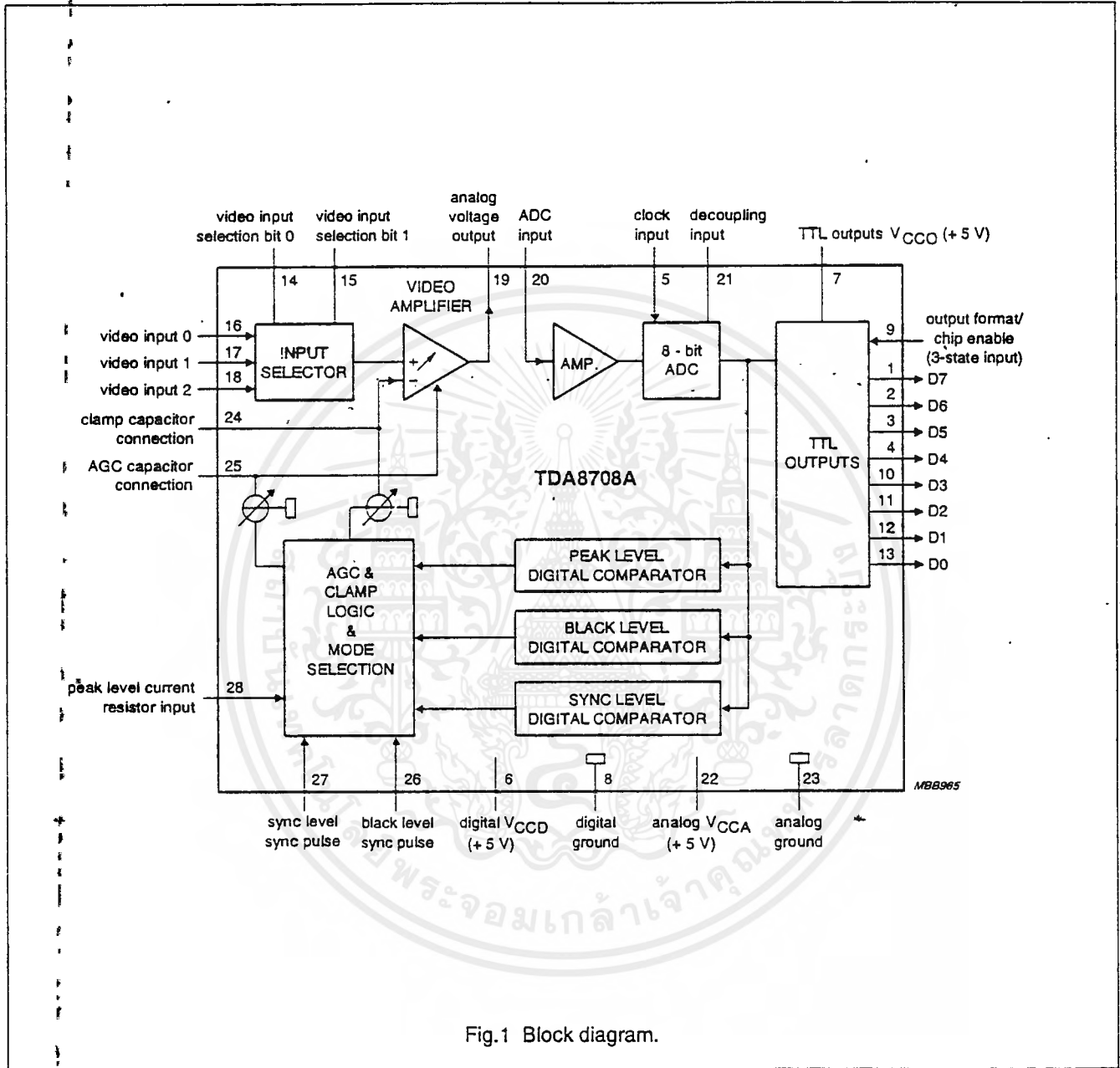


Fig.1 Block diagram.

Video analog input interface

TDA8708A

PINNING

SYMBOL	PIN	DESCRIPTION
D7	1	data output; bit 7 (MSB)
D6	2	data output; bit 6
D5	3	data output; bit 5
D4	4	data output; bit 4
CLK	5	clock input
V _{CCD}	6	digital supply voltage (+5 V)
V _{CCD}	7	TTL outputs supply voltage (+5 V)
DGND	8	digital ground
OF	9	output format/chip enable (3-state input)
D3	10	data output; bit 3
D2	11	data output; bit 2
D1	12	data output; bit 1
D0	13	data output; bit 0 (LSB)
I0	14	video input selection bit 0
I1	15	video input selection bit 1
VIN0	16	video input 0
VIN1	17	video input 1
VIN2	18	video input 2
ANOUT	19	analog voltage output
ADCIN	20	analog-to-digital converter input
DEC	21	decoupling input
V _{CCA}	22	analog supply voltage (+5 V)
AGND	23	analog ground
CLAMP	24	clamp capacitor connection
AGC	25	AGC capacitor connection
GATE B	26	black level synchronization pulse
GATE A	27	sync level synchronization pulse
RPEAK	28	peak level current resistor input

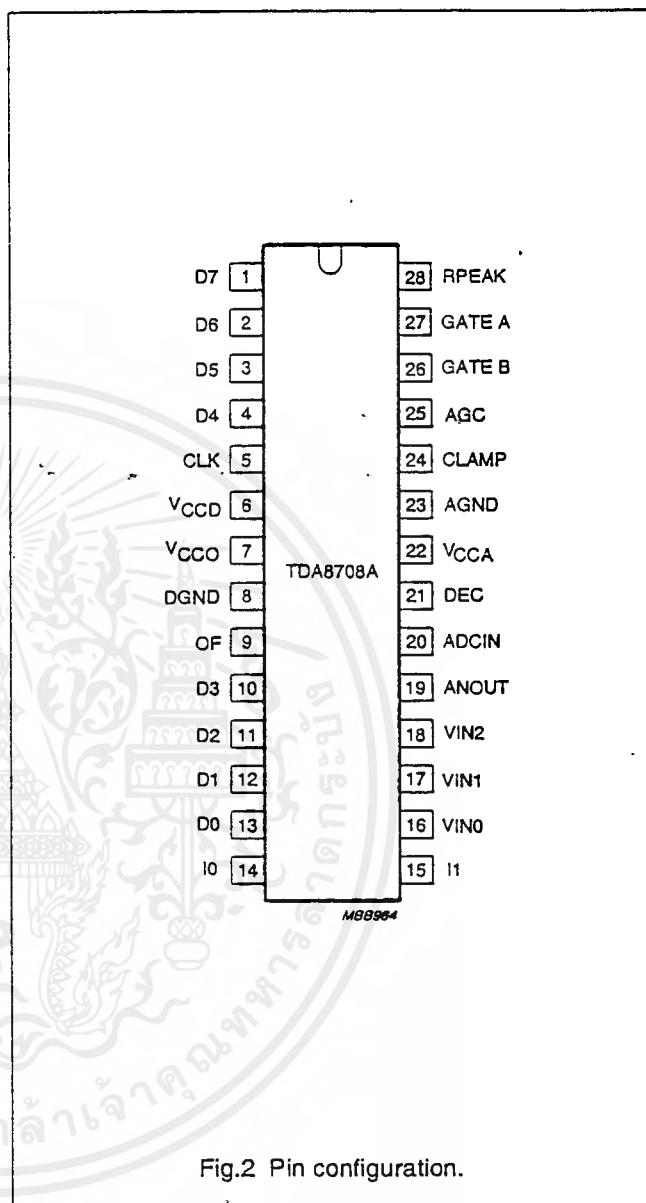


Fig.2 Pin configuration.

Video analog input interface

TDA8708A

FUNCTIONAL DESCRIPTION

The TDA8708A provides a simple interface for decoding video signals.

The TDA8708A operates in configuration mode 1 (see Fig.4) when the video signals are weak (i.e. when the gain of the AGC amplifier has not yet reached its optimum value). This enables a fast recovery of the synchronization pulses in the decoder circuit. When the pulses at the GATE A and GATE B inputs become distinct (GATE A and GATE B pulses are synchronization pulses occurring during the sync period and rear porch respectively) the TDA8708A automatically switches to configuration mode 2 (see Fig.5).

When the TDA8708A is in configuration mode 1, the gain of the AGC amplifier will be roughly adjusted (sync level to a digital output level of 0 and the peak-level to a digital output level of 255).

In configuration mode 2 the digital output of the ADC is compared to internal digital reference levels. The resultant outputs control the charge or discharge current of a capacitor connected to the AGC pin. The voltage across this capacitor controls the gain of the video amplifier. This is the gain control loop.

The sync level comparator is active during a positive-going pulse at the GATE A input. This means that the sync pulse of the composite video signal is used as an amplitude reference. The bottom of the sync pulse is adjusted to obtain a digital output of logic 0 at the converter output. As the black level is at digital level 64, the sync pulse will have a digital amplitude of 64 LSBs.

The peak-white control loop is always active. If the video signal tends to exceed the digital code of 248, the gain will be limited to avoid any over-range of the converter.

The use of nominal signals will prevent the output from exceeding a digital code of 213 and the peak-white control loop will be non-active.

The clamp level control is accomplished by using the same techniques as used for the gain control. The black-level digital comparator is active during a positive-going pulse at the GATE B input. The clamp capacitor will be charged or discharged to adjust the digital output to code 64.

LIMITING VALUES

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

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{CCA}	analog supply voltage	-0.3	+7.0	V
V_{CCD}	digital supply voltage	-0.3	+7.0	V
V_{CCO}	output supply voltage	-0.3	+7.0	V
ΔV_{CC}	supply voltage difference between V_{CCA} and V_{CCD}	-1.0	+1.0	V
	supply voltage difference between V_{CCO} and V_{CCD}	-1.0	+1.0	V
	supply voltage difference between V_{CCA} and V_{CCO}	-1.0	+1.0	V
V_I	input voltage	-0.3	V_{CCA}	V
I_O	output current	0	+10	mA
T_{stg}	storage temperature	-55	+150	°C
T_{amb}	operating ambient temperature	0	+70	°C
T_j	junction temperature	0	+125	°C

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	VALUE	UNIT
R_{th-j-a}	thermal resistance from junction to ambient in free air		
	SOT117-1	55	K/W
	SOT136-1	70	K/W

Video analog input interface

TDA8708A

CHARACTERISTICS

$V_{CCA} = V_{22}$ to $V_{23} = 4.5$ to 5.5 V; $V_{CCD} = V_6$ to $V_8 = 4.5$ to 5.5 V; $V_{CCO} = V_7$ to $V_8 = 4.2$ to 5.5 V; AGND and DGND shorted together; V_{CCA} to $V_{CCD} = -0.5$ to $+0.5$ V; V_{CCO} to $V_{CCD} = -0.5$ to $+0.5$ V; V_{CCA} to $V_{CCO} = -0.5$ to $+0.5$ V; $T_{amb} = 0$ to $+70$ °C; typical readings taken at $V_{CCA} = V_{CCD} = V_{CCO} = 5$ V and $T_{amb} = 25$ °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supplies						
V_{CCA}	analog supply voltage		4.5	5.0	5.5	V
V_{CCD}	digital supply voltage		4.5	5.0	5.5	V
V_{CCO}	TTL output supply voltage		4.2	5.0	5.5	V
I_{CCA}	analog supply current		–	37	45	mA
I_{CCD}	digital supply current		–	24	30	mA
I_{CCO}	TTL output supply current	TTL load (see Fig.8)	–	12	16	mA
Video amplifier inputs						
VIN(0 TO 2) INPUTS						
$V_{I(p-p)}$	input voltage (peak-to-peak value)	AGC load with external capacitor; note 1	0.6	–	1.5	V
$ Z_i $	input impedance	$f_i = 6$ MHz	10	20	–	k Ω
C_i	input capacitance	$f_i = 6$ MHz	–	1	–	pF
IO AND I1 TTL INPUTS (SEE TABLE 1)						
V_{IL}	LOW level input voltage		0	–	0.8	V
V_{IH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{IL}	LOW level input current	$V_i = 0.4$ V	–400	–	–	μ A
I_{IH}	HIGH level input current	$V_i = 2.7$ V	–	–	20	μ A
GATE A AND GATE B TTL INPUTS (SEE FIGS 4 AND 5)						
V_{IL}	LOW level input voltage		0	–	0.8	V
V_{IH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{IL}	LOW level input current	$V_i = 0.4$ V	–400	–	–	μ A
I_{IH}	HIGH level input current	$V_i = 2.7$ V	–	–	20	μ A
t_w	pulse width	see Fig.5	2	–	–	μ s
RPEAK INPUT (PIN 28)						
$I_{28(min)}$	minimum peak level current	$R_{28} = 0$ Ω	–	80	150	μ A
AGC INPUT (PIN 25)						
$V_{25(min)}$	AGC voltage for minimum gain		–	2.8	–	V
$V_{25(max)}$	AGC voltage for maximum gain		–	4.0	–	V
	AGC output current		see Table 2			
CLAMP INPUT (PIN 24)						
V_{24}	clamp voltage for code 128 output		–	3.5	–	V
I_{24}	clamp output current		see Table 3			

Video analog input interface

TDA8708A

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Video amplifier outputs						
ANOUT OUTPUT (PIN 19)						
$V_{19(p-p)}$	AC output voltage (peak-to-peak value)	$V_{VIN} = 1.33 \text{ V (p-p)}$; $V_{25} = 3.6 \text{ V}$	–	1.33	–	V
I_{19}	internal current source	$R_L = \infty$	2.0	2.5	–	mA
$I_{O(p-p)}$	output current driven by the load	$V_{ANOUT} = 1.33 \text{ V (p-p)}$; note 2	–	–	1.0	mA
V_{19}	DC output voltage for black level	note 3	–	$V_{CCA} - 2.24$	–	V
Z_{19}	output impedance		–	20	–	Ω
Video amplifier dynamic characteristics						
α_{ct}	crosstalk between VIN inputs	$V_{CCA} = 4.75 \text{ to } 5.25 \text{ V}$	–	–50	–45	dB
G_{diff}	differential gain	$V_{VIN} = 1.33 \text{ V (p-p)}$; $V_{25} = 3.6 \text{ V}$	–	2	–	%
φ_{diff}	differential phase	$V_{VIN} = 1.33 \text{ V (p-p)}$; $V_{25} = 3.6 \text{ V}$	–	0.8	–	deg
B	–3 dB bandwidth		12	–	–	MHz
S/N	signal-to-noise ratio	note 4	60	–	–	dB
SVRR1	supply voltage ripple rejection	note 5	–	45	–	dB
ΔG	gain range	see Fig.10	–4.5	–	+6.0	dB
G_{stab}	gain stability as a function of supply voltage and temperature	see Fig.10	–	–	5	%
Analog-to-digital converter inputs						
CLK INPUT (PIN 5)						
V_{IL}	LOW level input voltage		0	–	0.8	V
V_{IH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{IL}	LOW level input current	$V_{clk} = 0.4 \text{ V}$	–400	–	–	μA
I_{IH}	HIGH level input current	$V_{clk} = 2.7 \text{ V}$	–	–	100	μA
$ Z_i $	input impedance	$f_{clk} = 10 \text{ MHz}$	–	4	–	k Ω
C_i	input capacitance	$f_{clk} = 10 \text{ MHz}$	–	4.5	–	pF
OF INPUT (3-STATE; SEE TABLE 4)						
V_{IL}	LOW level input voltage		0	–	0.2	V
V_{IH}	HIGH level input voltage		2.6	–	V_{CCD}	V
V_g	input voltage in high impedance state		–	1.15	–	V
I_{IL}	LOW level input current		–370	–300	–	μA
I_{IH}	HIGH level input current		–	300	450	μA

Video analog input interface

TDA8708A

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
ADC IN INPUT (PIN 20; SEE TABLE 5)						
V_{20}	input voltage	digital output = 00	–	$V_{CCA} - 2.42$	–	V
V_{20}	input voltage	digital output = 255	–	$V_{CCA} - 1.41$	–	V
$V_{20(p-p)}$	input voltage amplitude (peak-to-peak value)		–	1.0	–	V
I_{20}	input current		–	1.0	10	μ A
$ Z_i $	input impedance	$f_i = 6$ MHz	–	50	–	M Ω
C_i	input capacitance	$f_i = 6$ MHz	–	1	–	pF
Analog-to-digital converter outputs						
DIGITAL OUTPUTS D0 TO D7						
V_{OL}	LOW level output voltage	$I_{OL} = 2$ mA	0	–	0.6	V
V_{OH}	HIGH level output voltage	$I_{OL} = -0.4$ mA	2.4	–	V_{CCD}	V
I_{OZ}	output current in 3-state mode	0.4 V < V_O < V_{CCD}	-20	–	+20	μ A
Switching characteristics						
$f_{clk(max)}$	maximum clock input frequency	see Fig.6; note 6	30	32	–	MHz
Analog signal processing ($f_{clk} = 32$ MHz; see Fig.8)						
G_{diff}	differential gain	$V_{20} = 1.0$ V (p-p); see Fig.3; note 7	–	2	–	%
φ_{diff}	differential phase	see Fig.3; note 7	–	2	–	deg
f_1	fundamental harmonics (full-scale)	$f_i = 4.43$ MHz; note 7	–	–	0	dB
f_{all}	harmonics (full-scale); all components	$f_i = 4.43$ MHz; note 7	–	-55	–	dB
SVRR2	supply voltage ripple rejection	note 8	–	1	5	%/V
Transfer function (see Fig.8)						
ILE	DC integral linearity error		–	–	± 1	LSB
DLE	DC differential linearity error		–	–	± 0.5	LSB
ILE	AC integral linearity error	note 9	–	–	± 2	LSB
Timing ($f_{clk} = 32$ MHz; see Figs 6, 7 and 8)						
DIGITAL OUTPUTS ($C_L = 15$ pF; $I_{OL} = 2$ mA; $R_L = 2$ kΩ)						
t_{ds}	sampling delay time		–	2	–	ns
t_h	output hold time		6	8	–	ns
t_d	output delay time		–	16	20	ns
t_{dEZ}	3-state delay time; output enable		–	19	25	ns
t_{dDZ}	3-state delay time; output disable		–	14	20	ns

Video analog input interface

TDA8708A

Notes

- 0 dB is obtained at the AGC amplifier when applying $V_{i(p-p)} = 1.33$ V.
- The output current at pin 19 should not exceed 1 mA. The load impedance R_L should be referenced to V_{CCA} and defined as:
 - AC impedance ≥ 1 k Ω and the DC impedance > 2.7 k Ω .
 - The load impedance should be coupled directly to the output of the amplifier so that the DC voltage supplied by the clamp is not disturbed.
- Control mode 2 is selected.
- Signal-to-noise ratio measured with 5 MHz bandwidth:

$$\frac{S}{N} = 20 \log \frac{V_{ANOUTC(p-p)}}{V_{ANOUTY(RMS\ noise)}} \text{ at } B = 5 \text{ MHz.}$$

- The voltage ratio is expressed as:

$$SVRR1 = 20 \log \frac{\Delta V_{CCA}}{V_{CCA}} \times \frac{G}{\Delta G} \text{ for } V_i = 1 \text{ V (p-p), gain at 100 kHz} = 1 \text{ and 1 V supply variation.}$$

- It is recommended that the rise and fall times of the clock are ≥ 2 ns. In addition, a 'good layout' for the digital and analog grounds is recommended.
- These measurements are realized on analog signals after a digital-to-analog conversion (TDA8702 is used).
- The supply voltage rejection is the relative variation of the analog signal (full-scale signal at input) for 1 V of supply variation:

$$SVRR2 = \frac{\Delta(V_{I(00)} - V_{I(FF)}) + (V_{I(00)} - V_{I(FF)})}{\Delta V_{CCA}}$$

- Full-scale sine wave ($f_i = 4.4$ MHz; $f_{clk} = 27$ MHz).

Video analog input interface

TDA8708A

Table 1 Video input selection (CVBS).

I1	I0	SELECTED INPUT
0	0	VIN0
0	1	VIN1
1	0	VIN2
1	1	VIN2

Table 2 AGC output current.

GATE A	GATE B	DIGITAL OUTPUT	I _{AGC}	MODE ⁽²⁾
1	1	output < 255	-2.5 μ A	1
		output > 255	I _{AGCM}	1
0	X ⁽¹⁾	output < 248	0 μ A	2
		output > 248	I _{AGCM}	2
1	0	output < 0	+2.5 μ A	2
		0 < output < 248	-2.5 μ A	2
		output > 248	I _{AGCM}	2

Note

- X⁽¹⁾ = don't care.
- Mode 2 can only be initialized with successive pulses on GATE A and GATE B (see Fig.5).

Table 5 Output coding and input voltage (typical values).

STEP	V _{ADCIN}	BINARY OUTPUTS								TWO'S COMPLEMENT							
		D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Underflow	-	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	V _{CCA} - 2.41 V	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	-	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
-	-
-	-
254	-	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0
255	V _{CCA} - 1.41 V	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Overflow	-	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Table 3 CLAMP output current.

GATE A	GATE B	DIGITAL OUTPUT	I _{CLAMP}	MODE
1	1	output < 0	I _{CLAMPM}	1
		output > 0	-2.5 μ A	1
X ⁽¹⁾	0	X ⁽¹⁾	0 μ A	2
0	1	output < 64	+50 μ A	2
		64 < output	-50 μ A	2

Note

- X = don't care.

Table 4 OF input coding.

OF	D0 TO D7
0	active, two's complement
1	high impedance
open circuit ⁽¹⁾	active, binary

Note

- Use C \geq 10 pF to DGND.

Video analog input interface

TDA8708A

INTERNAL PIN CIRCUITRY

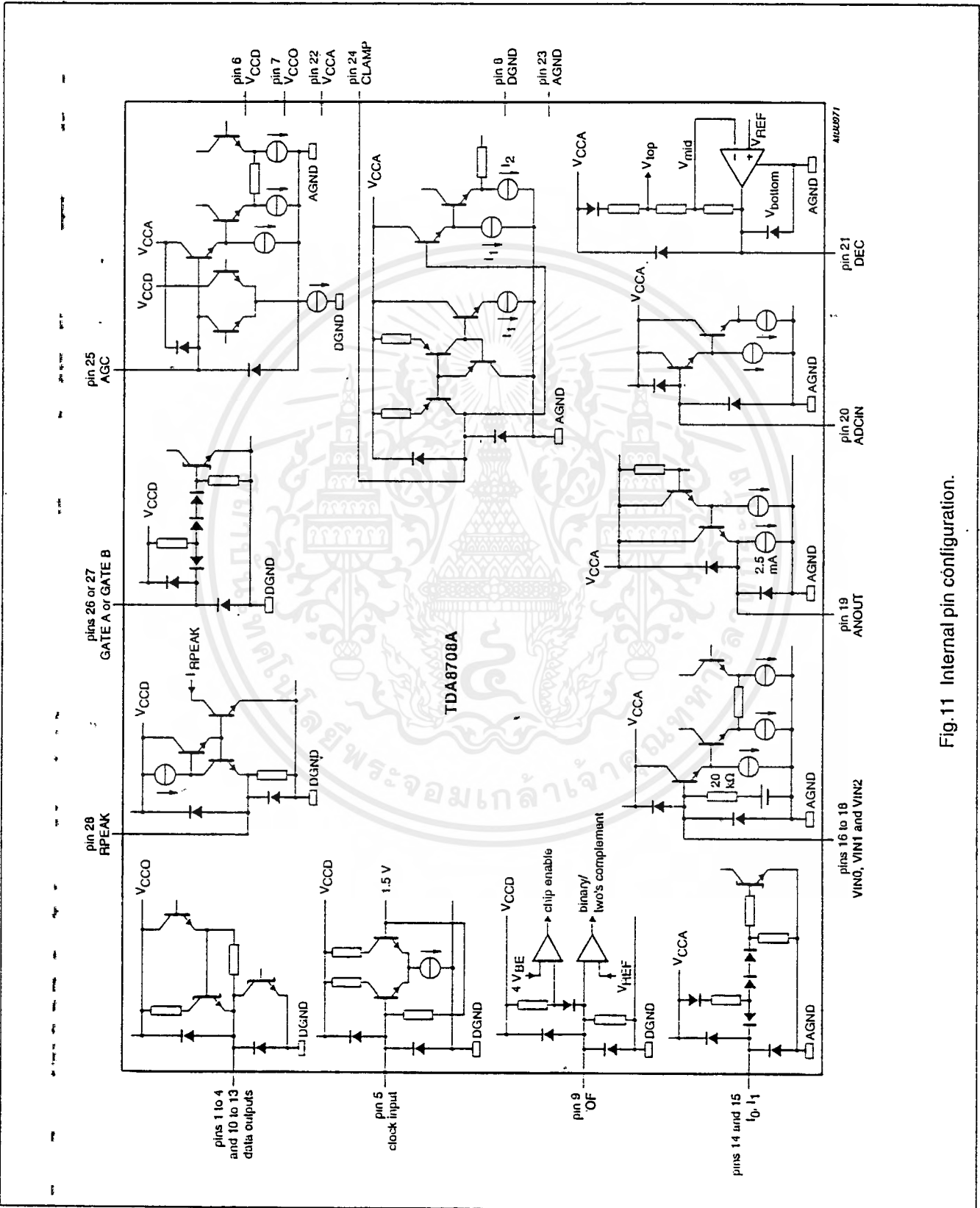


Fig.11 Internal pin configuration.

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
Data Retention: 10 Years
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

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

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five source two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is

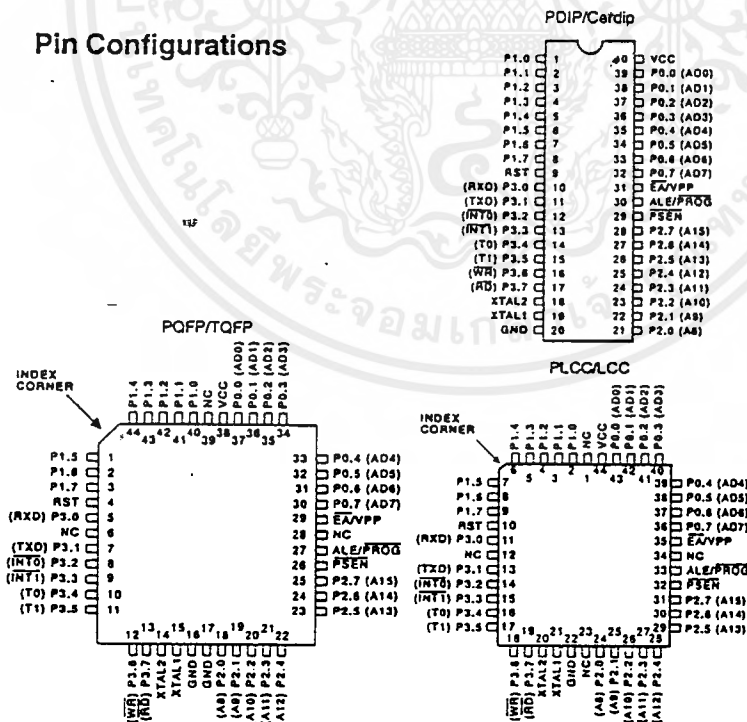
continued



8-Bit
Microcontroller
with 4 Kbytes
Flash

AT89C51

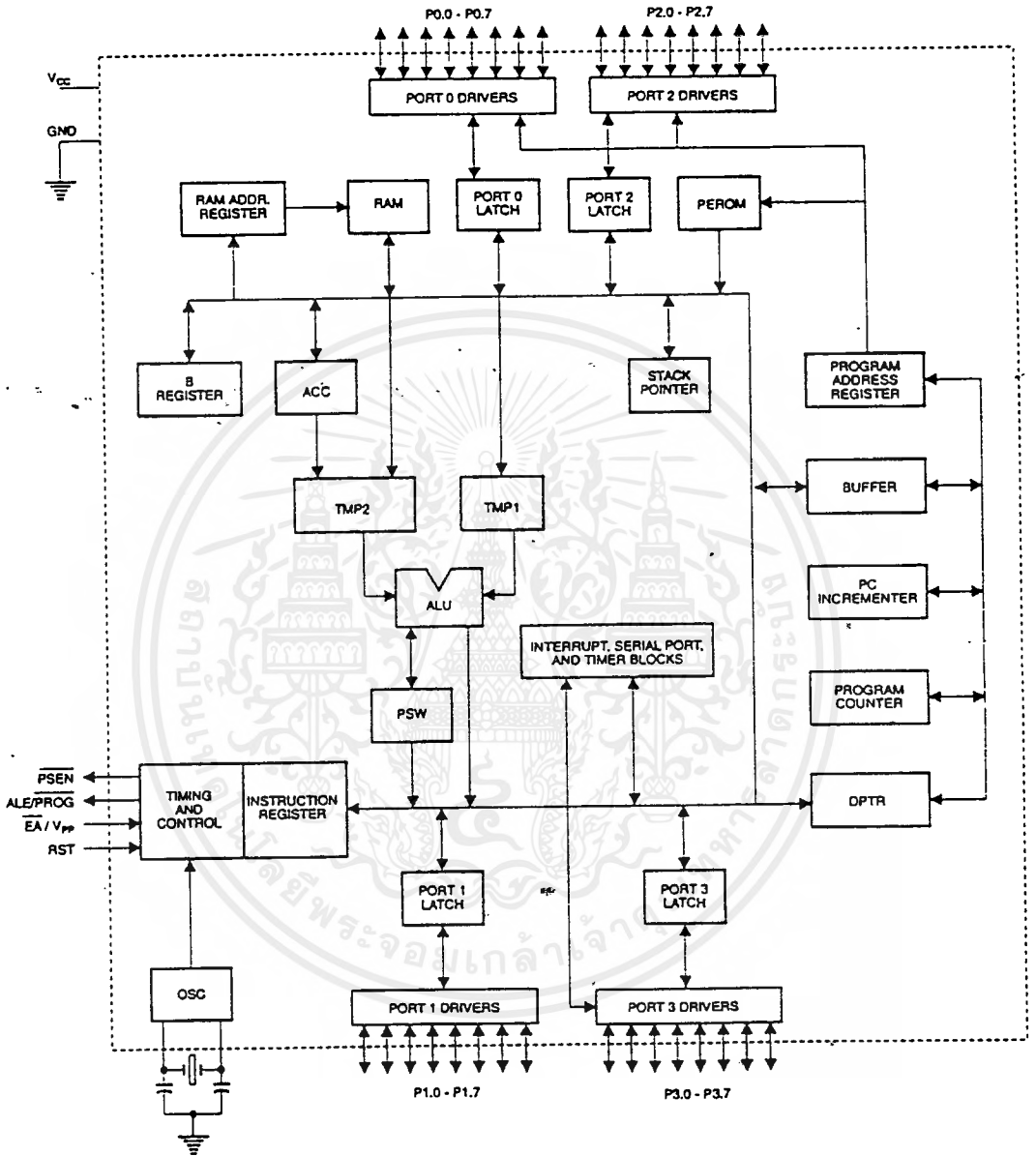
Pin Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

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

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

Port 2

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

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

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

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s

are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

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

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

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

RST

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

ALE/ $\overline{\text{PROG}}$

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

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

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

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

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

$\overline{\text{EA/Vpp}}$

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

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

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

continued



Pin Description (Continued)

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

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

Idle Mode

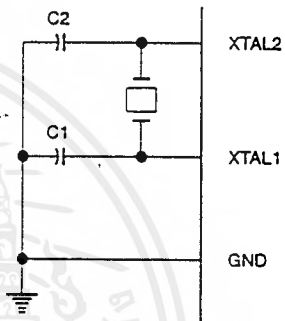
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

Power Down Mode

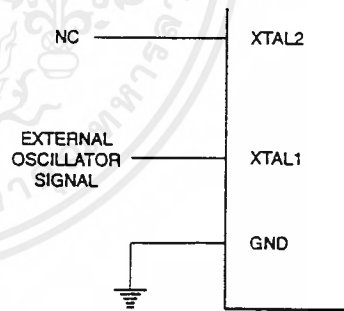
In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines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

Figure 1. Oscillator Connections



Notes: $C_1, C_2 = 30 \text{ pF} \pm 10 \text{ pF}$ for Crystals
 $= 40 \text{ pF} \pm 10 \text{ pF}$ for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

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

Program Memory Lock Bits

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

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up

without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of \overline{EA} be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

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

Programming the Flash

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

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

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

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

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

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.

4. Raise \overline{EA}/V_{pp} to 12 V for the high-voltage programming mode.
5. Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

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

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

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

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

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





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

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

Programming Interface

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

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

Flash Programming Modes

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

Notes: 1. The signature byte at location 032H designates whether V_{pp} = 12 V or V_{pp} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

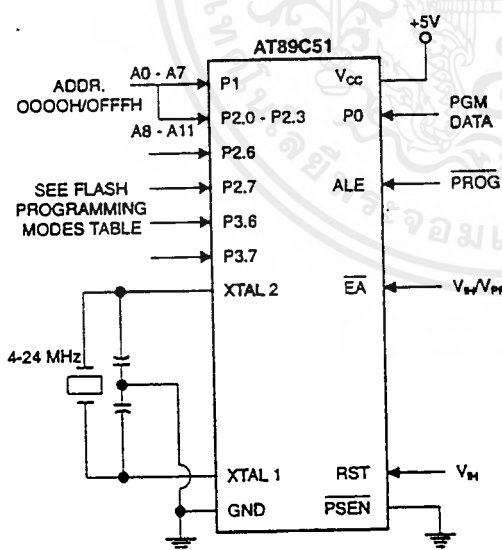
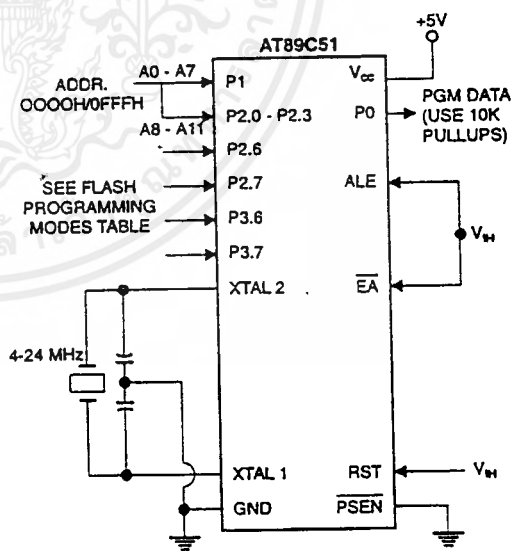


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

TA = 21°C to 27°C, VCC = 5.0 ± 10%

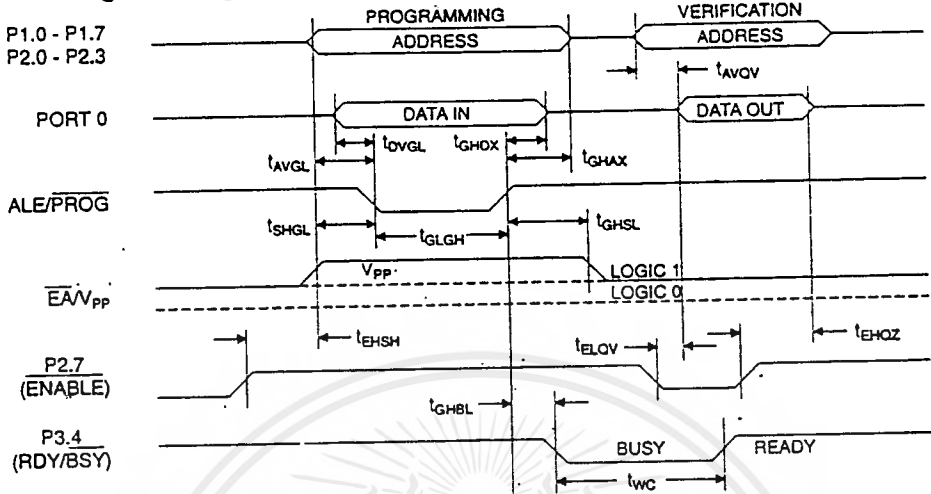
Symbol	Parameter	Min	Max	Units
VPP ⁽¹⁾	Programming Enable Voltage	11.5	12.5	V
IPP ⁽¹⁾	Programming Enable Current		1.0	mA
1/tCLCL	Oscillator Frequency	4	24	MHz
tAVGL	Address Setup to PROG Low	48tCLCL		
tGHAX	Address Hold After PROG	48tCLCL		
tOVGL	Data Setup to PROG Low	48tCLCL		
tGHDX	Data Hold After PROG	48tCLCL		
tEHS	P2.7 (ENABLE) High to VPP	48tCLCL		
tSHGL	VPP Setup to PROG Low	10		µs
tGHSL ⁽¹⁾	VPP Hold After PROG	10		µs
tGLGH	PROG Width	1	110	µs
tAVQV	Address to Data Valid		48tCLCL	
tELOV	ENABLE Low to Data Valid		48tCLCL	
tEHQV	Data Float After ENABLE	0	48tCLCL	
tGHBL	PROG High to BUSY Low		1.0	µs
tWC	Byte Write Cycle Time		2.0	ms

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

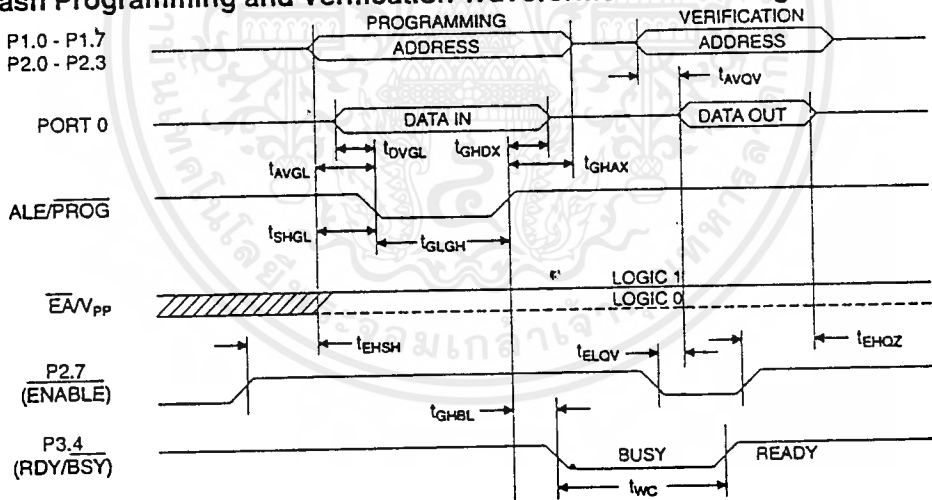




Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

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

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

D.C. Characteristics

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

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

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8-bit port:
 Port 0: 26 mA
 Ports 1, 2, 3: 15 mA
 Maximum total IOL for all output pins: 71 mA

If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
 2. Minimum VCC for Power Down is 2 V.





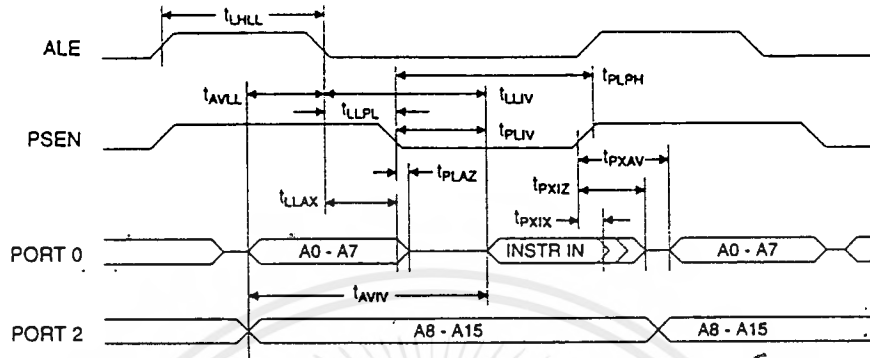
A.C. Characteristics

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

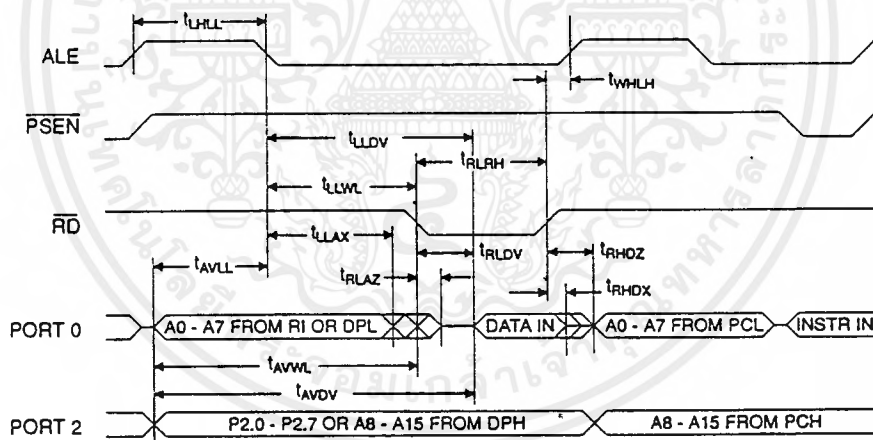
External Program and Data Memory Characteristics

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

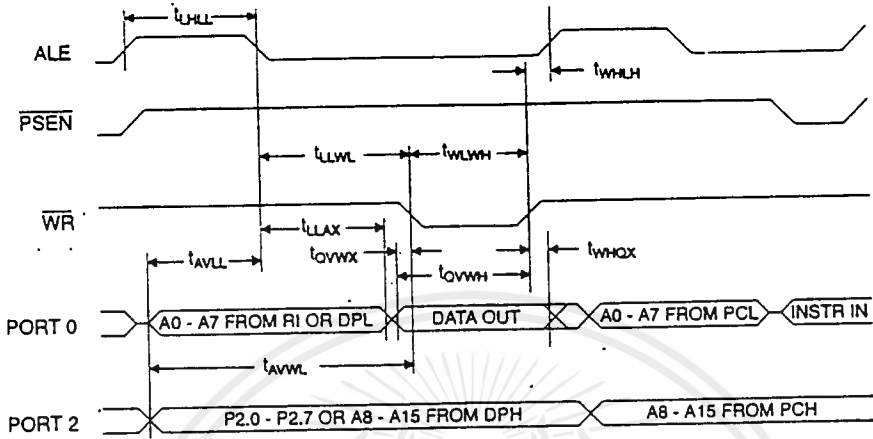
External Program Memory Read Cycle



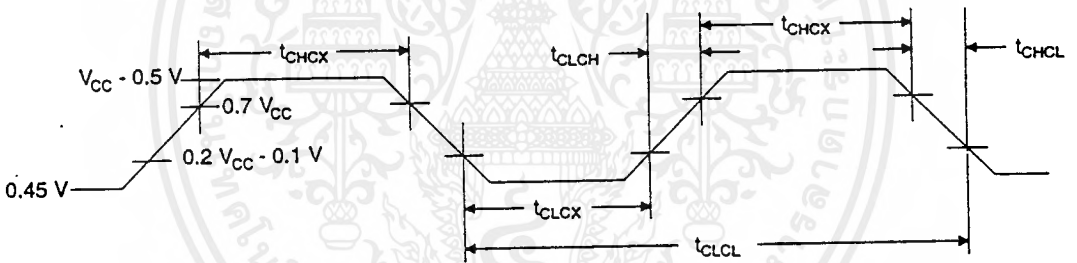
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

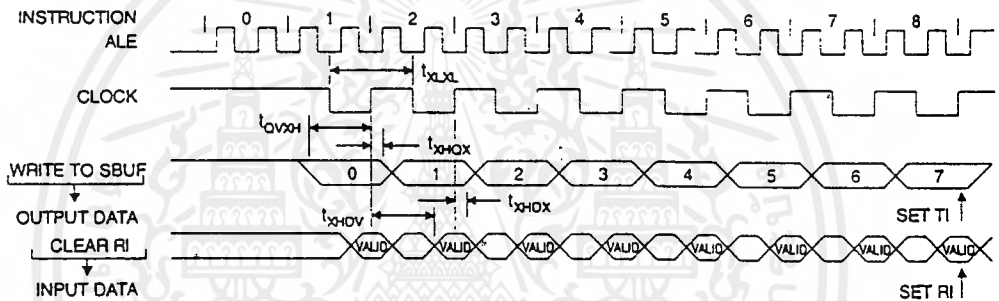
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

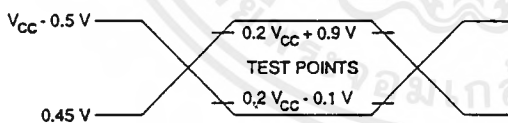
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t _{xLXL}	Serial Port Clock Cycle Time	1.0		12t _{CLCL}		μs
t _{OVXH}	Output Data Setup to Clock Rising Edge	700		10t _{CLCL} -133		ns
t _{XHOX}	Output Data Hold After Clock Rising Edge	50		2t _{CLCL} -33		ns
t _{XHOX}	Input Data Hold After Clock Rising Edge	0		0		ns
t _{XHDV}	Clock Rising Edge to Input Data Valid		700	10t _{CLCL} -133		ns

Shift Register Mode Timing Waveforms

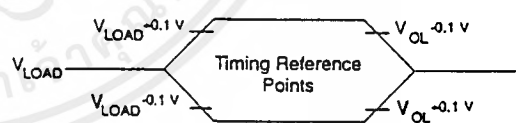


AC Testing Input/Output Waveforms ⁽¹⁾



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

Float Waveforms ⁽¹⁾



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





Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V \pm 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
	5 V \pm 10%	AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
	5 V \pm 10%	AT89C51-12AA	44A	Automotive (-40°C to 125°C)
		AT89C51-12JA	44J	
5 V \pm 10%	AT89C51-12PA	40P6	Military (-55°C to 125°C)	
	AT89C51-12QA	44Q		
5 V \pm 10%	AT89C51-12DM	40D6	Military/B83C Class B, Fully Compliant (-55°C to 125°C)	
	AT89C51-12LM/883	44L		
16	5 V \pm 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
	5 V \pm 20%	AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
	5 V \pm 20%	AT89C51-16AA	44A	Automotive (-40°C to 125°C)
		AT89C51-16JA	44J	
AT89C51-16PA		40P6		
AT89C51-16QA		44Q		
20	5 V \pm 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
	5 V \pm 20%	AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
5 V \pm 20%	AT89C51-20PI	40P6		
	AT89C51-20QI	44Q		
24	5 V \pm 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
	5 V \pm 20%	AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Finger_m.c */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include <conio.h>
#include <alloc.h>
#include <graphics.h>
#include <bios.h>

#define X_SIZE 512
#define Y_SIZE 512
#define Output_Sbit 0x378
#define Input_Sbit 0x379

int check;
unsigned char huge data[512][512];
unsigned char huge logo[180][180];
char filename[80];

#include "init256.c"
#include "screen.c"
#include "status.c"
#include "control.c"
#include "getfile.c"
#include "readfile.c"
#include "savefile.c"
#include "menu.c"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main()
{
  initsvga_gray256();
  Initial_port();
  screen();
  readlogo();
  menu();
  graphdefaults();
  closegraph();
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Control.c */
```

```
/* Control Hardware */
```

```
void Initial_port(void)
```

```
{  
    outport(Output_8bit,0xff);  
}
```

```
void Receive(void)
```

```
{  
    /* Wait recive data from MCS-51 pin 32 (bit3) = 0 */
```

```
do{  
    check =inport(Input_5bit);  
}while(((check&0x0f)>>3)!=0);  
}
```

```
void Start(void)
```

```
{  
    /* Send data=0 to MSC 51 (Acknowledge) */
```

```
    outport(Output_8bit,0xfe);  
}
```

```
void Read_data_odd(void)
```

```
{  
    int i,j,pt,n=0;  
    for(j=0; j<Y_SIZE ; j=j+2)  
    {  
        for( i=0; i<X_SIZE ; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    /* Wait to Acknowledge data READY by pin 32 */
    Receive();
    /* Select 4 low bit use pin 3 */
    outport(Output_8bit,0xfd);
    /* Read first column and 4 low bit */
    data[i][j] = (inport(Input_5bit)>>4)&0x000f;
    /* Select 4 high bit use pin 4 control 74244*/
    outport(Output_8bit,0xfb);
    /* Full 8 bit */
    data[i][j] =((inport(Input_5bit)&0x00f0)|data[i][j]);
    /* Read already by pin 6*/
    outport(Output_8bit,0xef);
}
}
}

```

```

void Read_data_even(void)

```

```

{
    int i,j,pt,n=0;
    for(j=1; j<Y_SIZE; j=j+2)
    {
        for( i=0; i<X_SIZE; i++)
        {
            /* Wait to Acknowledge read data to next address by pin 32 */
            Receive();
            /* Select 4 low bit use pin 3 */
            outport(Output_8bit,0xfd);
            /* Read first column and 4 low bit */
            data[i][j] = (inport(Input_5bit)>>4)&0x000f;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    /* Select 4 high bit use pin 4 */
    outport(Output_8bit,0xfb);
    /* Full 8 bit */
    data[i][j] =((inport(Input_5bit)&0x00f0)|data[i][j]);
    /* Read already by pin 6 */
    outport(Output_8bit,0xef);
}
}
}

void Change_field(void)
{
    outport(Output_8bit,0xfb);
}

void Data_hardware(void)
{
    buttom3(575,384,775,423,60,150,100,602,220,60,"DISPLAY FINGERPRINT");
    delay(100);
    buttom3(575,384,775,423,150,60,100,600,220,60,"DISPLAY FINGERPRINT");
    clearstatus();
    buttom2(32,560,289,595,100,100,100); // clear status command
    buttom3(33,561,100,594,100,100,100,40,200,60,"STATUS : Display FINGERPRINT .."); // status
    bar

    Start();
    Read_data_odd();
    Change_field();
    Read_data_even();
    displayimage();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
button2(32,560,289,595,100,100,100); // clear status command
```

```
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Getfile.c */
```

```
void getfilename()
```

```
{
```

```
char ch=10;
```

```
int x=300,i=0,j=0,n;
```

```
for(n=0;n<80;n++) //clear filename..
```

```
filename[n]=NULL;
```

```
settextstyle(7,0,1);
```

```
setcolor(200);
```

```
while(ch!=13)
```

```
{
```

```
setcolor(200);
```

```
outtextxy(x+i,578,"_");
```

```
ch=getch();
```

```
if(ch==0)
```

```
ch=getch();
```

```
else
```

```
{
```

```
if(ch==8)
```

```
{
```

```
j--;
```

```
i=i-8;
```

```
if(j >= 0)
```

```
{
```

```
filename[j]=NULL;
```

```
clearstatus();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

setcolor(200);
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(x,576,filename);
    outtextxy(x+i,578,"_");
}
else
{
    j++;
    i=i+8;
} // end if j >= 0
} //end if ch=8
else if(ch != 13)
{
    filename[j]=ch;
    setcolor(0);
    outtextxy(x+i,578,"_");
    setcolor(200);
    outtextxy(x,576,filename);
    i=i+8;
    j++;
} //end else if
} // end if ch=0 special key
} // end loop while
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program for Initial 800x600 256 color (Gray scale)*/
```

```
/* Init256.c */
```

```
int huge DetectVGA256(void);
```

```
void initsvga_gray256();
```

```
void setgraycolor(void);
```

```
int huge DetectVGA256()
```

```
{  
    int Vid=3;  
    return Vid;  
}
```

```
void initsvga_gray256()
```

```
{  
    int Gd=DETECT,Gm;  
    int xasp,yasp;
```

```
    installuserdriver("Svga256",DetectVGA256);
```

```
    imitgraph(&Gd,&Gm,"");
```

```
    getaspectratio(&xasp,&yasp);
```

```
    setaspectratio(xasp,yasp);
```

```
    setgraycolor();  
}
```

```
void setgraycolor()
```

```
{  
    int i,j,n=0;
```

```
    int R,G,B;
```

```
    float weighcolor[256];  
    เมื่อก่อนเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
    เมื่อก่อนนี้ทุกสิ่งทุกอย่างก็ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```

int newcolor[256][3];
int buff1[3];
float buff2;
union REGS regs;

settextstyle(2,0,1);
setusercharsize(1,2,1,1);
outtextxy(0,0,"Please Wait !");
for(i=0;i<=63;i++)
for(j=0;j<=3;j++)
{
regs.h.ah = 0x10;
regs.h.al = 0x10;
regs.x.bx = n;
switch (j) {
case 0 : regs.h.dh = i;
regs.h.ch = i;
regs.h.cl = i;
break;
case 1 : regs.h.dh = i;
regs.h.ch = i;
regs.h.cl = i+1;
break;
case 2 : regs.h.dh = i+1;
regs.h.ch = i;
regs.h.cl = i;
break;
case 3 : regs.h.dh = i;
regs.h.ch = i+1;
regs.h.cl = i;
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
    int86(0x10,&regs,&regs);
    n++;
}

for(i=0;i<=255;i++)
{
    regs.h.ah=0x10;
    regs.h.al=0x15;
    regs.x.bx=i;
    int86(0x10,&regs,&regs);
    newcolor[i][0]=regs.h.dh;
    newcolor[i][1]=regs.h.ch;
    newcolor[i][2]=regs.h.cl;
    weighcolor[i]=(0.3*newcolor[i][0])+(0.59*newcolor[i][1])
        +(0.11*newcolor[i][2]);
}
for(i=0;i<=255;i++)
for(j=0;j<=254;j++)
{
    if (weighcolor[j+1]< weighcolor[j])
    {
        buff1[0]=newcolor[j][0];
        buff1[1]=newcolor[j][1];
        buff1[2]=newcolor[j][2];
        newcolor[j][0]=newcolor[j+1][0];
        newcolor[j][1]=newcolor[j+1][1];
        newcolor[j][2]=newcolor[j+1][2];
        newcolor[j+1][0]=buff1[0];
        newcolor[j+1][1]=buff1[1];
        newcolor[j+1][2]=buff1[2];
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม้ว่ากรณีนี้ที่สงวนไว้ให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

newcolor[j+1][2]=buff1[2];
buff2=weighcolor[j];
weighcolor[j]=weighcolor[j+1];
weighcolor[j+1]=buff2;
}
}
for(i=0;i<=255;i++)
{
regs.h.ah=0x10;
regs.h.al=0x10;
regs.x.bx=i;
regs.h.dh=newcolor[i][0];
regs.h.ch=newcolor[i][1];
regs.h.cl=newcolor[i][2];
int86(0x10,&regs,&regs);
}

regs.h.ah = 0x10;
regs.h.al = 0x1b;
regs.x.bx = 0;
regs.x.cx = 256;
int86(0x10,&regs,&regs);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Menu.c */
```

```
void menu()
```

```
{
```

```
int mn=0;
```

```
char ch=10;
```

```
while(mn!=3) // check for exit program if mn=3 will quit program
```

```
{
```

```
while(ch!=13) // check key Enter for select menu
```

```
{
```

```
ch=getch();
```

```
if(ch==0) // check special key if ch=0 >> special key
```

```
{
```

```
ch=getch();
```

```
if(ch==72 || ch==80) // 72 = Up-arrow , 80 = Down-arrow
```

```
{
```

```
if(ch==72)
```

```
{
```

```
if(mn==0)
```

```
mn=4;
```

```
mn=mn-1;
```

```
switch(mn)
```

```
{
```

```
case 0 : buttom3(575,301,775,341,150,60,100,585,220,60,"OPEN  
FINGERPRINT IMAGE");
```

```
buttom3(575,343,775,382,150,60,100,585,150,60,"SAVE  
FINGERPRINT IMAGE");
```

```
break;
```

```
case 1 : buttom3(575,343,775,382,150,60,100,585,220,60,"SAVE  
FINGERPRINT IMAGE");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        button3(575,384,775,423,150,60,100,600,150,60,"DISPLAY
                FINGERPRINT");

        break;

    case 2 : button3(575,384,775,423,150,60,100,600,220,60,"DISPLAY
                FINGERPRINT");

        button3(575,507,775,546,150,60,100,630,150,60,"EXIT
                PROGRAM");

        break;

    case 3 : button3(575,507,775,546,150,60,100,630,220,60,"EXIT
                PROGRAM");

        button3(575,301,775,341,150,60,100,585,150,60,"OPEN
                FINGERPRINT IMAGE");

        break;
} // end switch mn
}
else
{
    if(mn==3)
        mn=-1;
    mn=mn+1;
    switch(mn)
    {
        case 0 : button3(575,301,775,341,150,60,100,585,220,60,"OPEN
                FINGERPRINT IMAGE");

        button3(575,507,775,546,150,60,100,630,150,60,"EXIT
                PROGRAM");

        break;

        case 1 : button3(575,343,775,382,150,60,100,585,220,60,"SAVE
                FINGERPRINT IMAGE");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ button3(575,301,775,341,150,60,100,585,150,60,"OPEN
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก FINGERPRINT IMAGE"); ถ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;

    case 2 : buttom3(575,384,775,423,150,60,100,600,220,60,"DISPLAY
                FINGERPRINT");

        buttom3(575,343,775,382,150,60,100,585,150,60,"SAVE
                FINGERPRINT IMAGE");

        break;

    case 3 : buttom3(575,507,775,546,150,60,100,630,220,60,"EXIT
                PROGRAM");

        buttom3(575,384,775,423,150,60,100,600,150,60,"DISPLAY
                FINGERPRINT");

        break;
    } // end switch mn
} // end if ch=72
} // end if ch = 72 or 80
} // end if ch=0 check special key
} // end.loop while ch!=13

switch(mn)
{
    case 0 : readfile();

        mn=0;

        ch=10;

        break;

    case 1 : savefile();

        mn=1;

        ch=10;

        break;

    case 2 : Data_hardware();

        mn=2;

        ch=10;

        break;

    case 3 : buttom3(575,507,775,546,60,150,100,632,220,60,"EXIT PROGRAM");

```

```
delay(100);  
buttom3(575,507,775,546,150,60,100,630,220,60,"EXIT PROGRAM");  
clearstatus();  
buttom2(32,560,289,595,100,100,100); // clear status command  
buttom3(33,561,100,594,100,100,100,40,200,60,"STATUS : Quit Program ..");  
// status bar  
  
cleardisplay();  
mn=3;  
ch=10;  
break;  
} // end switch mn  
} // end loop while mn!=3  
}
```



```
/* Readfile.c */
```

```
void readfile()
```

```
{
```

```
int i;
```

```
FILE *imagefile;
```

```
button3(575,301,775,341,60,150,100,587,220,60,"OPEN FINGERPRINT IMAGE"); // button  
command
```

```
delay(100);
```

```
button3(575,301,775,341,150,60,100,585,220,60,"OPEN FINGERPRINT IMAGE"); // button  
command
```

```
clearstatus();
```

```
button2(32,560,289,595,100,100,100); // clear status command
```

```
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Open Image .."); // status bar
```

```
getfilename();
```

```
if(filename[0] != NULL)
```

```
{
```

```
if((imagefile = fopen(filename,"rb"))==NULL)
```

```
{
```

```
clearstatus();
```

```
outtextxy(300,578,"No file");
```

```
outtextxy(381,578,filename);
```

```
delay(1000);
```

```
clearstatus();
```

```
button2(32,560,289,595,100,100,100); // clear status command
```

```
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
```

```
}  
else
```

```
ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
|
```

```

{
fread(data,512,512,imagefile);
for(i=0;i<100;i++){
    statusbar(i);
    delay(5);
}
clearstatus();
fclose(imagefile);
displayimage();
button2(32,560,289,595,100,100,100); // clear status command
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
}
} // end if filename != NULL.
else
{
clearstatus();
button2(32,560,289,595,100,100,100); // clear status command
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
} // end else filename[0] != NULL.
}

```

```
void readlogo()
```

```

{
int i;
FILE *imagefile;

if((imagefile = fopen("logo.raw", "rb"))==NULL)
    printf("No file %s\n", "logo.raw");
else

```

{
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
fread(logo,180,180,imagefile);
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
fclose(imagefile);  
displaylogo();  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Savefile.c */
```

```
void savefile()
```

```
{
```

```
int i;
```

```
FILE *imagefile;
```

```
button3(575,343,775,382,60,150,100,587,220,60,"SAVE FINGERPRINT IMAGE"); // button  
command
```

```
delay(100);
```

```
button3(575,343,775,382,150,60,100,585,220,60,"SAVE FINGERPRINT IMAGE"); // button  
command
```

```
button2(32,560,289,595,100,100,100); // clear status command
```

```
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Save Image .."); // status bar
```

```
getfilename();
```

```
if((imagefile = fopen(filename,"wb"))==NULL)
```

```
printf("No file %s\n",filename);
```

```
else
```

```
{
```

```
fwrite(data,512,512,imagefile);
```

```
for(i=0;i<100;i++){
```

```
statusbar(i);
```

```
delay(5);
```

```
}
```

```
clearstatus();
```

```
fclose(imagefile);
```

```
button2(32,560,289,595,100,100,100); // clear status command
```

```
button3(33,561,100,594,100,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Screen.c */
```

```
void screen(void);
```

```
void button(int x1,int y1,int x2,int y2,int ul,int lr,int face);
```

```
void button2(int x1,int y1,int x2,int y2,int ul,int lr,int face);
```

```
void button3(int x1,int y1,int x2,int y2,int ul,int lr,int face,int xt,int etc,int cuc,char tc[30]);
```

```
void screen()
```

```
{
```

```
int x,y;
```

```
cleardevice();
```

```
button(0,0,799,599,100,100,100);
```

```
button3(33,561,100,594,100,100,40,200,60,"STATUS : Waiting Command .."); // status bar
```

```
button2(290,565,794,594,60,150,0); // status bar
```

```
button2(23,23,559,559,150,60,100);
```

```
button2(33,33,549,549,60,150,0); // for display image
```

```
button(575,25,775,224,150,60,100); // for logo of program
```

```
button(585,35,765,214,60,150,0);
```

```
button3(575,301,775,341,150,60,100,585,220,60,"OPEN FINGERPRINT IMAGE"); // button  
command
```

```
button3(575,343,775,382,150,60,100,585,150,60,"SAVE FINGERPRINT IMAGE"); // button  
command
```

```
button3(575,384,775,423,150,60,100,600,150,60,"DISPLAY FINGERPRINT"); // button  
command
```

```
button2(575,425,775,464,150,60,100); // button command
```

```
button2(575,466,775,505,150,60,100); // button command
```

```
button3(575,507,775,546,150,60,100,630,150,60,"EXIT PROGRAM"); // button command
```

```
settextstyle(7,0,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
setcolor(60); ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextxy(130,12,"DIGITAL FINGER-PRINT ACQUISITION SYSTEM");
setcolor(220);
outtextxy(128,10,"DIGITAL FINGER-PRINT ACQUISITION SYSTEM");
}
```

```
void buttom(int x1,int y1,int x2,int y2,int ul,int lr,int face)
```

```
{
setcolor(ul);
line(x1,y1,x2,y1);
line(x1,y1,x1,y2);
setcolor(lr);
line(x1,y2,x2,y2);
line(x2,y1,x2,y2);
setfillstyle(1,face);
bar(x1+1,y1+1,x2-1,y2-1);
}
```

```
void buttom2(int x1,int y1,int x2,int y2,int ul,int lr,int face)
```

```
{
setcolor(ul);
line(x1,y1,x2,y1);
line(x1,y1,x1,y2);
line(x1+1,y1+1,x2-1,y1+1);
line(x1+1,y1+1,x1+1,y2-1);
setcolor(lr);
line(x1,y2,x2,y2);
line(x2,y1,x2,y2);
line(x1+1,y2-1,x2-1,y2-1);
line(x2-1,y1+1,x2-1,y2-1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
setfillstyle(1,face);
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bar(x1+2,y1+2,x2-2,y2-2);
```

```
}
```

```
void button3(int x1,int y1,int x2,int y2,int ul,int lr,int face,int xt,int ctc,int cuc,char tc[30])
```

```
{
```

```
setcolor(ul);
```

```
line(x1,y1,x2,y1);
```

```
line(x1,y1,x1,y2);
```

```
line(x1+1,y1+1,x2-1,y1+1);
```

```
line(x1+1,y1+1,x1+1,y2-1);
```

```
setcolor(lr);
```

```
line(x1,y2,x2,y2);
```

```
line(x2,y1,x2,y2);
```

```
line(x1+1,y2-1,x2-1,y2-1);
```

```
line(x2-1,y1+1,x2-1,y2-1);
```

```
setfillstyle(1,face);
```

```
bar(x1+2,y1+2,x2-2,y2-2);
```

```
settextstyle(7,0,1);
```

```
setcolor(cuc);
```

```
outtextxy(xt+2,y1+17,tc);
```

```
setcolor(ctc);
```

```
outtextxy(xt,y1+15,tc);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} /* Status.c */

void statusbar(int pct)
{
    int a;
    a=pct*5;
    setfillstyle(1,170);
    bar(294,569,294+a,590);
}

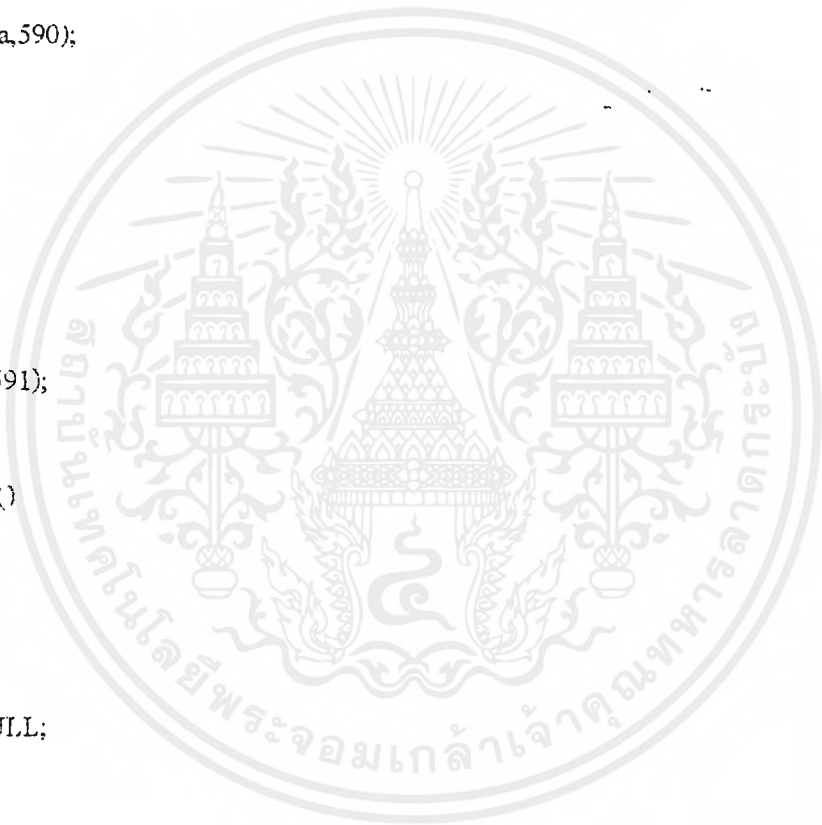
void clearstatus()
{
    setfillstyle(1,0);
    bar(292,568,792,591);
}

void clearfilename()
{
    int i;
    for(i=0;i<80;i++)
        filename[i]=NULL;
}

void displayimage()
{
    int i,j;

    for(i=0;i<512;i++)
        for(j=0;j<512;j++){
            putpixel(35+j,35+i,data[j][i]);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void displaylogo()
{
    int i,j;

    for(i=0;i<180;i++)
        for(j=0;j<180;j++){
            putpixel(585+i,35+j,logo[j][i]);
        }
}

void cleardisplay()
{
    int i,j;

    setfillstyle(1,0);
    for(i=0;i<256;i++){

        bar(290-i,290-i,291+i,291+i);

    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

โปรแกรมเอสเซมบลี 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ORG 0000H  
COMMAND1 EQU 010H ;WRITE DATA TO RAM  
COMMAND2 EQU 00CH ;READ DATA ODD  
COMMAND3 EQU 00EH ;READ DATA EVEN
```

```
;***** START PROGRAM *****
```

```
MAIN: CLR P1.4  
JB P1.5,$  
ANL P1,#0E1H ;MARK BIT  
ORL P1,#COMMAND1 ;WRITE DATA  
CALL DELAY ;DELAY 120mS  
ANL P1,#0E1H  
ORL P1,#COMMAND2 ;READ DATA ODD  
CLR P1.6  
CALL TRANS_RAM  
SETB P1.6  
CALL TRANS_RAM  
JB P3.2,$  
ANL P1,#0E1H  
ORL P1,#COMMAND3 ;READ DATA EVEN  
CLR P1.6  
CALL TRANS_RAM  
SETB P1.6  
CALL TRANS_RAM  
JMP MAIN
```

```
;*****DELAY 120mS SUB*****
```

```
;DELAY SUBROUTINE
```

```
DELAY: MOV R2,#6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEL1: MOV R3,#100
DEL2: MOV R4,#100
      DJNZ R4,$
      DJNZ R3,DEL2
      DJNZ R2,DEL1
      RET
```

```
;*****TRANS RAM SUB*****
```

```
TRANS_RAM: MOV DPTR,#0000H
LOOP1:    MOVX A,@DPTR
          MOVX @DPTR,A
          CLR P1.0
          JB  P1.7,$
          SETB P1.0
          MOV A,#00H
          MOV A,DPL
          CJNE A,#0FFH,EXIT
          MOV A,DPH
          CJNE A,#0FFH,EXIT
          JMP FINISH
EXIT:    INC DPTR
          JMP LOOP1
FINISH:  RET
          END
```



1. เจน สงสมพันธุ์, นิคม อนันต์ทิพย์, เทคโนโลยีโทรทัศน์, สถาบันอิเล็กทรอนิกส์กรุงเทพฯ, หน้า 24 - 49.
2. ไอฉีน่าสน CA3318, เซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 95, ปี 2532, หน้า 274 - 277.
3. อนันต์ ตั้งเกียรติตรง, จักร์สม ลิ้มปี่ธีระกุล และ เทพฤทธิ์ ปิยพันธุ์วรจกร, ระบบเก็บข้อมูลลายนิ้วมือ, ณะวิศวกรรมศาสตร์ สจล., ปี 2539.
4. จิติ หนูแก้ว, เทคนิคการเชื่อมต่อ IBM PC, บริษัท ซีเอ็ดยูเคชั่น จำกัด, หน้า 65 - 67.
5. ชันวา ศรีประมง, การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม, พิมพ์ครั้งที่ 4 (เพิ่มเติม), มหาวิทยาลัยเทคโนโลยีมหานคร, ปี 2539.
6. อรรถสิทธิ์ หล้าสกุล, การเก็บภาพขนาด 512 x 256 จุด โดยใช้หน่วยความจำความเร็วต่ำ, ณะวิศวกรรมศาสตร์ สจล., ปี 2532.