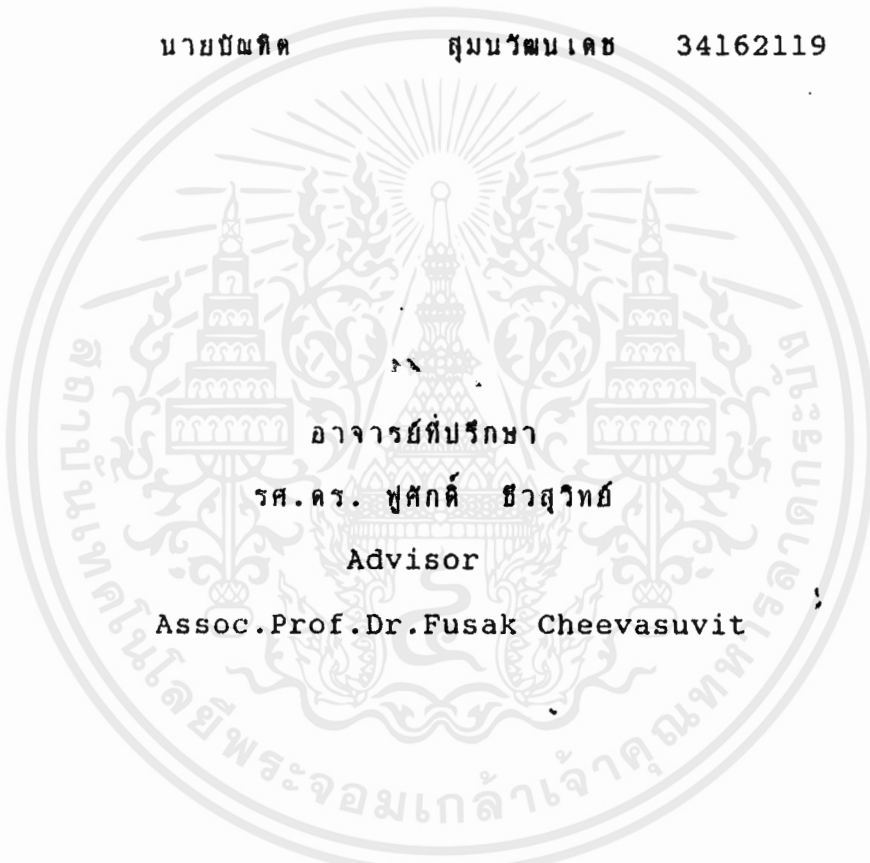




แผนงานจรรยาบรรณภาพ

IMAGE INTERFACE CARD

นายกฤษฎ	วงศ์ศรีทอง	34162101
นายคเชนทร์	แจ่มกมล	34162106
นายบัณฑิต	ศุมนวิวัฒนเดช	34162119



อาจารย์ที่ปรึกษา
ร.ศ.ดร. ฟุสศักดิ์ ชิวสุวิทย์
Advisor
Assoc.Prof.Dr.Fusak Cheevasuvit

ปริญญาโทสำหรับปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
ภาควิชาเทคโนโลยีการวิศวกรรมอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032561

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แผงวงจรควบคุมภาพ

IMAGE INTERFACE CARD

ผู้จัดทำ

1. นายกำจร วงษ์ศรีทอง 34162101
2. นายคเชนทร์ แจ่มกมล 34162106
3. นายบัณฑิต สมนวัตินเดช 34162119

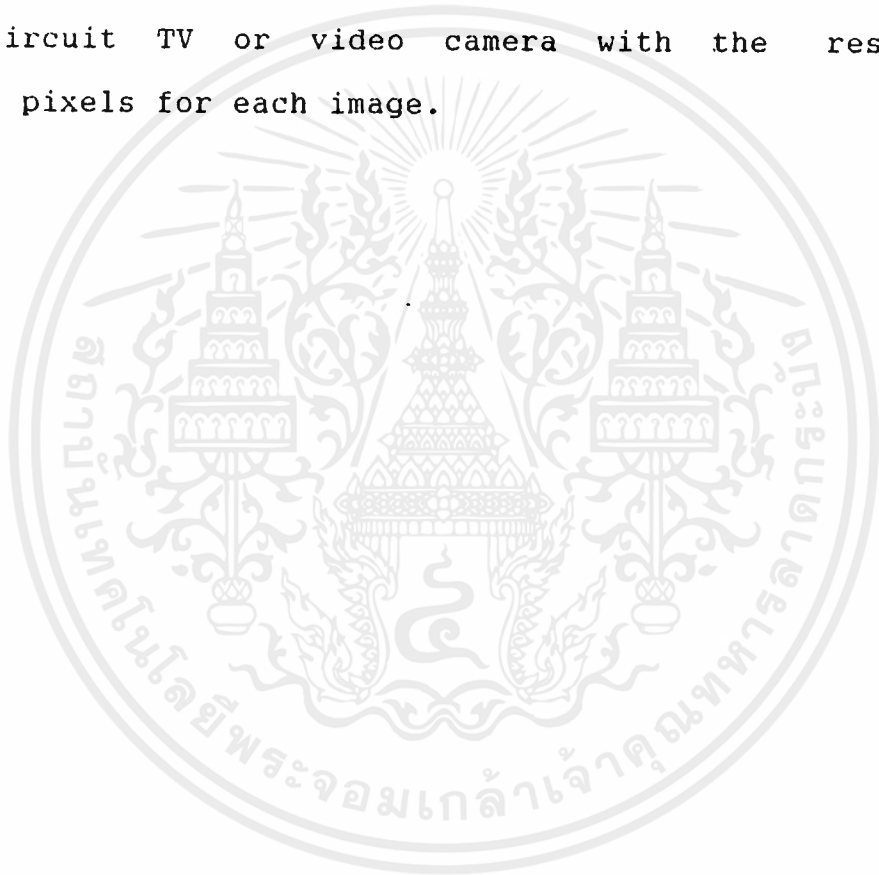
..... อาจารย์ที่ปรึกษา
(รศ.ดร. พุคักดิ์ ชิวสุวิทย์)

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอการออกแบบระบบประมวลผลภาพแบบเวลาจริง (Real time image processing system) โดยสร้างแผงวงจรควบคุมภาพ (Image Interface Card) ที่ประมวลผลโดย IBM PC ซึ่งมีคุณสมบัติที่สำคัญคือ สามารถเก็บภาพแบบเวลาจริง (Real time image acquisition) ที่ได้จากรถสองล้อ, วีดีโอเทป หรืออุปกรณ์ที่สามารถผลิตสัญญาณโทรทัศน์ มาเก็บไว้ในรูปของข้อมูลที่สามารถนำมาประมวลผลโดยคอมพิวเตอร์ต่อไปได้ และยังสามารถแสดงภาพของข้อมูลที่เก็บได้นั้น ออกทางจอมอนิเตอร์ โดยการแปลงกลับให้เป็นสัญญาณโทรทัศน์อีกครั้งหนึ่ง ภาพที่เก็บได้หรือภาพที่แสดงโดยแผงวงจรควบคุมภาพเป็นภาพขาวดำที่มีความละเอียด 256x256 จุด ซึ่งจะต้องใช้หน่วยความจำ 64 Kbytes ในการประมวลผลภาพหนึ่งภาพ

ABSTRACT

The thesis describes the hardware design of real-time image processing card for working on microcomputer IBM PC and also the software design for image processing. Image data can be gotten from close circuit TV or video camera with the resolution of 256x256 pixels for each image.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Abstract

บทนำ	1
บทที่ 1	ส่วนประกอบต่าง ๆ ในระบบประมวลผลภาพ.....	3
บทที่ 2	ระบบสัญญาณโทรทัศน์	5
บทที่ 3	การจัดหน่วยความจำ	15
บทที่ 4	ระบบการทำงาน	19
บทที่ 5	การใช้งานการ์ดและการทำงานของการ์ด.....	31
บทที่ 6	สรุปผลการทดลอง.....	33
เอกสารอ้างอิง		
กิตติกรรมประกาศ		
ภาคผนวก		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ในปัจจุบันเทคโนโลยีทางด้าน Image Processing ได้แพร่หลายในเมืองไทย มาก และเริ่มมีแนวโน้มในการนำเอาเทคโนโลยีทางด้านนี้มาประยุกต์ใช้กับงานต่าง ๆ อีก มากมาย ซึ่งโครงการนี้เป็นการศึกษาและทดลองเกี่ยวกับ Image Processing ซึ่งแบ่ง ออกเป็นสองส่วนใหญ่ ๆ คือ ส่วนการศึกษาและทดลองทางด้าน Hardware และส่วนการ ศึกษาและทดลองทางด้าน Software ซึ่งในส่วนของการศึกษาและทดลองทางด้าน Hard ware จะทำให้สำเร็จออกมาในรูปแบบของ Image Interface Card และในส่วนของ Soft ware จะศึกษาในด้านการประมวลผลภาพด้วยโปรแกรม

อย่างไรก็ตามการศึกษาและทดลองของคณะผู้จัดทำ สำหรับในขั้นตอนนี้ยังเป็นแนวทาง ของศึกษาและทดลอง ในส่วนของ Hardware เท่านั้น

1.1 วัตถุประสงค์ของปริญาณิพนธ์

ในโครงการที่นำเสนอนี้ ได้ทำการพัฒนาระบบประมวลผลภาพ ที่มีความละเอียด ขนาด 256x256 จุดต่อภาพ ซึ่งมีระดับความสว่างในการแสดงภาพ 256 ระดับที่สามารถ ติดต่อและควบคุมได้โดย Personal Computer ทัว ๆ ซึ่งจะสามารถนำไปประยุกต์ใช้ใน งานด้านต่าง ๆ

1.2 ขอบเขตของปริญาพนธ์

ในส่วนของปริญาพนธ์จะ เป็นการแสดงการออกแบบและการสร้าง แผลวงจรรวม คุมการค ที่มีความสามารถทำภารกิจใค้ภาพให้อยู่ในรูปของข้อมูล เก็บลงสู่หน่วยความจำ บนการค และแสดงภาพที่อยู่ในหน่วยความจำบนการคออกสู่จอมอนิเตอร์ได้ ซึ่งในปริญา พนธ์นี้จะเป็นแผลวงจรรวมภาพที่มีความละเอียด 256 x 256 จุดต่อหนึ่งภาพสามารถ ความคมได้ด้วยคอมพิวเตอร์ เนื้อหารายละเอียดต่าง ๆ ดังนี้

- | | |
|--------|---|
| บทนำ | เป็นการกล่าวถึงวัตถุประสงค์ต่าง ๆ และขอบเขตของปริญาพนธ์ |
| บทที่1 | กล่าวถึงส่วนประกอบต่าง ๆ ที่ใช้ในระบบประมวลผลภาพ |
| บทที่2 | เนื้อหาเกี่ยวกับระบบสัญญาณทรานส์มิชชั่น |
| บทที่3 | การจัดการหน่วยความจำ |
| บทที่4 | การออกแบบและการสร้างแผลวงจรรวม |
| บทที่5 | อธิบายถึงวิธีการใช้งานการค |
| บทที่6 | บทสรุปผลที่ได้จากการทดลอง |

บทที่ 1

ส่วนประกอบต่าง ๆ ในระบบประมวลผลภาพ (Element of a image processing system)

ในระบบประมวลผลภาพประกอบไปด้วยส่วนต่าง ๆ ดังนี้

1.1 ตัวประมวลผลภาพ (Image processor)

ตัวประมวลผลภาพนับได้ว่า เป็นหัวใจสำคัญของระบบประมวลผลภาพเพราะเป็นส่วนของฮาร์ดแวร์ที่มีหน้าที่สำคัญอยู่ 4 ประการ คือ

1.1.1 การเก็บภาพ (Image acquisition) เป็นการแปลงสัญญาณโทรทัศน์ ซึ่งเป็นสัญญาณอนาล็อกให้เป็นข้อมูลดิจิทัลเพื่อเก็บในหน่วยความจำ โดยทั่วไปตัวประมวลผลภาพสามารถจับภาพจากสัญญาณโทรทัศน์ได้ภายในช่วงเวลาหนึ่งเฟรม และในโครงการที่ทำอยู่ดิจิทัลเพียงฟิล์มเดียวในหนึ่งเฟรม

1.1.2 การบันทึก (Storage) ก็คือส่วนของหน่วยความจำที่ใส่เก็บภาพที่ได้จากการดิจิทัล ซึ่งจะใช้หน่วยความจำมากหรือน้อยขึ้นอยู่กับความละเอียดของจุดภาพ ในโครงการนี้ให้ความละเอียดของจุดภาพ 256x256 จุดต่อภาพ ซึ่งต้องใช้หน่วยความจำ 64 กิโลไบต์ เราเรียกส่วนนี้ว่าเฟรมบัฟเฟอร์ (Frame buffer)

1.1.3 การประมวลผลในระดับต่ำ (Low level processing) เป็นส่วนของฮาร์ดแวร์ที่ใช้ในการทำงานทางด้านลอจิกมักนิยมเรียกว่า ALU (Arithmetic Logic Unit) ออกแบบมาเพื่อใช้เพิ่มความเร็วของระบบ

1.1.4 การแสดงผล (Display) เป็นการอ่านข้อมูลภาพที่อยู่ในหน่วยความจำเพื่อเปลี่ยนให้เป็นสัญญาณโทรทัศน์ต่อไป

1.2 ดิจิไทเซอร์ (Digitizer)

เป็นส่วนที่แปลงสัญญาณภาพให้เป็นข้อมูลที่เหมาะสมทางด้าน NUMERICAL สำหรับการใช้ในการประมวลผลต่อไป

1.3 คอมพิวเตอร์ (Computer)

เนื่องจากตัวประมวลผลภาพมีความสามารถเพียงแปลงข้อมูลจากสัญญาณโทรทัศน์ให้มาอยู่ในรูปของสัญญาณดิจิทัลในหน่วยความจำ แต่ในบางครั้งระบบการประมวลผลภาพจำเป็นต้องมีการคำนวณทางคณิตศาสตร์ที่ยุ้งยากหรืออาจจะมีความต้องการผสมตัวอักษรให้เข้ากับรูปภาพแล้วบันทึกเก็บไว้และอื่น ๆ อีกมากมาย เพื่อสนองตอบต่อความต้องการเหล่านี้เราจึงต้องทำการเชื่อมระบบของตัวประมวลผลภาพให้เข้ากับระบบคอมพิวเตอร์ทั่วไป ในโครงการงานนี้เราได้ทำการเชื่อมระบบเข้ากับ IBM PC ซึ่งเป็นระบบคอมพิวเตอร์ที่ได้รับความนิยมทั่วโลก

1.4 อุปกรณ์ที่ใช้ในการเก็บข้อมูล (Storage device)

ภาพที่ได้จากตัวประมวลผลภาพมีความละเอียด 256x256 จุดต่อภาพ แต่ละจุดมีระดับความสว่าง 256 ระดับ (8 บิต/จุด) ซึ่งต้องใช้อุปกรณ์ที่เก็บข้อมูลอย่างน้อย 64 กิโลไบต์ อุปกรณ์ที่ใช้เก็บข้อมูลที่นิยมกันได้แก่ ฟลอปปีดิสก์, ฮาร์ดดิสก์

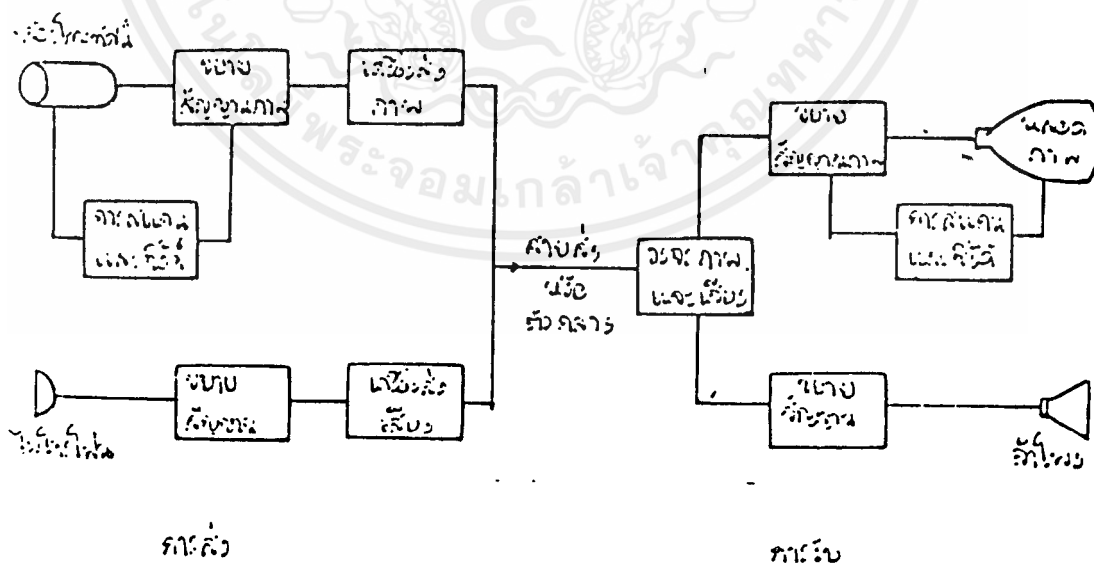
1.5 ส่วนแสดงภาพ (Display)

เป็นส่วนที่ใช้แสดงภาพที่ได้จากตัวประมวลผลภาพ, ภาพที่เก็บอยู่ในดิสก์และ ฯลฯ ซึ่งได้แก่ จอโมโนโครมหรือจอโทรทัศน์ที่สามารถแสดงภาพที่มีระดับความสว่างแตกต่างกันมาก ๆ ได้ เครื่องพิมพ์ (Printer) ก็เป็นอุปกรณ์ชนิดหนึ่งที่ใช้ในการแสดงภาพที่ไม่ต้องการระดับความสว่างมากนักทั้งนี้ เนื่องจากในแต่ละจุดภาพของเครื่องพิมพ์จะให้ระดับความสว่างเพียง 2 ระดับเราจึงต้องมาสร้างเมตริกซ์ของจุดภาพที่มีขนาด $n \times n$ โดย $n = 1, 2, \dots$ ซึ่งแล้วแต่ที่เราจะต้องการภาพที่มีความละเอียดแค่ไหน รวมทั้งความสามารถในการพล็อต (Plot) จุด/แถว ของเครื่องพิมพ์เองด้วย เราเรียกการแสดงผลภาพในลักษณะนี้ว่า half-tone picture นอกจากนี้ยังมีอุปกรณ์ที่ใช้ในการแสดงผลภาพได้อีก เช่น เลเซอร์พริ้นเตอร์ เป็นต้น

บทที่ 2

ระบบสัญญาณโทรทัศน์ (composite signal system)

ในระบบโทรทัศน์นั้นจะต้องประกอบไปด้วย กล้องโทรทัศน์ทำหน้าที่ผลิตสัญญาณภาพ (Video Signal) และไมโครโฟนทำหน้าที่ผลิตสัญญาณเสียง (Audio Signal) โดยมีเครื่องมือแสดงผล (Monitor) ทำหน้าที่แสดงภาพและเสียงที่สร้างได้จากกล้องโทรทัศน์ และไมโครโฟนอีกครั้งหนึ่งดัง รูป 2.1 และเพื่อให้การส่งและการรับสัญญาณภาพได้อย่างถูกต้องจำเป็นต้องมีสัญญาณซิงค์ (Synchronizing Pulses) สำหรับควบคุมการทำงานให้ดำเนินไปอย่างถูกต้อง ทั้งนี้ยังมีสัญญาณอื่น ๆ ที่ช่วยให้เกิดภาพที่สมบูรณ์อีกโดยจะกล่าวละเอียดในส่วต่อไป ในกรณีที่ต้องการส่งสัญญาณออกไปในระยะทางไกลๆ สัญญาณเหล่านี้จะถูกมอดูเลต (Modulate) ด้วยคลื่นพาหะ (Carrier) ก่อน แล้วจึงทำการส่งไปได้ ในโครงงานนี้เพียงแต่นำสัญญาณโทรทัศน์ที่ได้ จากอุปกรณ์กำเนิดสัญญาณโทรทัศน์ เช่น กล้องโทรทัศน์ , วีดีโอเทป ฯลฯ มาใช้ในขบวนการเก็บภาพเพียงอย่างเดียวโดยไม่มีการมอดูเลต ดังนั้น เนื้อหาที่จะกล่าวต่อไปนี้จะเน้นเฉพาะส่วนของสัญญาณโทรทัศน์เท่านั้น



รูปที่ 2.1 ระบบการส่ง-รับโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ส่วนประกอบของภาพ (Picture Elements)

ภาพขาว-ดำแต่ละภาพถ้าลองขยายให้ใหญ่ขึ้น จะพบว่าภาพประกอบด้วยจุดขาวและจุดดำมากมาย ดังเช่นภาพจากหนังสือพิมพ์ การจัดเรียงตัวของจุดขาวและจุดดำทำให้เกิดภาพขึ้นได้ จุดเหล่านี้เรียกว่า ส่วนประกอบของภาพ สำหรับในพื้นที่เท่ากันแล้วภาพใดที่มีจำนวนระดับของความสว่าง (Level of Brightness) มากกว่าจะเป็นภาพที่คมชัดและกลมกลืนมากกว่า สิ่งนี้เป็นสาเหตุหนึ่งที่ทำให้เราไม่สามารถใช้จอภาพโมโนโครมของ IBM PC/XT แสดงผลได้ เพราะมีจำนวนระดับของความสว่างเพียง 2 ระดับ คือ ขาว กับ ดำ เท่านั้น



รูปที่ 2.2 เปรียบเทียบภาพที่มีพื้นที่เท่ากัน

2.2 การสแกน (Scanning)

จากที่ได้กล่าวมาแล้วข้างต้นว่าภาพประกอบด้วยจำนวนส่วนประกอบภาพมากมาย ซึ่งแต่ละจุดภาพ ที่ส่งไปจะประกอบว่าเป็นจุดขาวหรือดำก็แสดง โดยสัญญาณภาพ ทางด้านส่งจะส่งทีละจุดเป็นลำดับแยกกันไป ทางด้านรับก็จะนำจุดต่างวเหล่านี้มาเรียบเรียงกันใหม่ให้เป็นภาพขึ้นมีเป็นเอวธีนี้เรียกว่าสแกนเนอร์งานเส้นที่ประกอบกันเป็นภาพนี้จึงใช้โปรแกรมที่ควบคุมการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

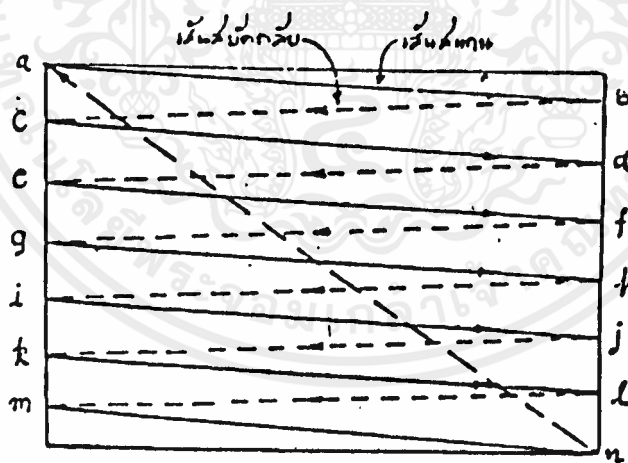
สแกน ปัจจุบันในโลกเรามีโทรทัศน์ใช้กันอยู่ 2 ระบบ คือ

2.2.1 โทรทัศน์ระบบอเมริกัน มีจำนวนเส้นสแกน 525 เส้น/ภาพและทำการส่ง 30 ภาพ/วินาที ดังนั้นความถี่ที่ใช้ในการสแกนเท่ากับ $(525) \times (30) = 15,750 \text{ Hz}$

2.2.2 โทรทัศน์ระบบยุโรป มีจำนวนเส้นสแกน 625 เส้น/ภาพ และทำการส่ง 25 ภาพ/วินาที ดังนั้นความถี่ที่ใช้ในการสแกนเท่ากับ $(625) \times (25) = 15,625 \text{ Hz}$

การสแกนมีด้วยกัน 2 วิธี คือการสแกนแบบก้าวหน้า (Progressive Scanning) และวิธีสแกนแบบสลับเส้น (Interlaced Scanning)

จากรูป 2.3 จะเห็นได้ว่าการสแกนเริ่มจาก a--->b , c--->d , e--->f จนถึงสุดท้าย k--->l ซึ่งเป็นการสแกนตามลำดับจากซ้ายไปขวาและจากข้างบนลงข้างล่าง เหมือนการอ่านหนังสือหรือการพิมพ์ตีคั่นเอง เป็นการสแกนแบบที่ใช้ข้อสซิโลสโคป เรียก การสแกนแบบก้าวหน้า



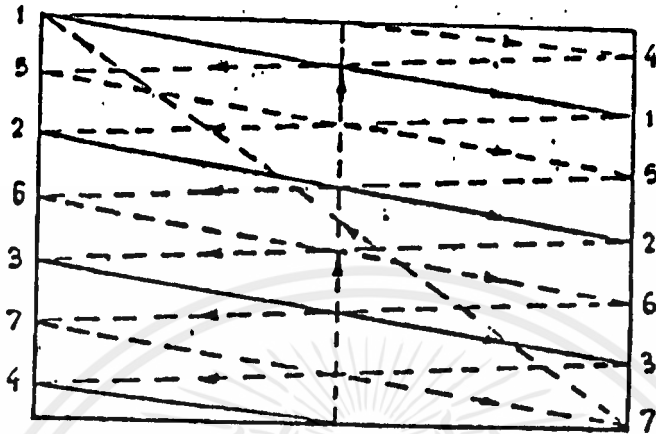
รูปที่ 2.3 การสแกนแบบก้าวหน้า

จากรูป 2.4 (ก) จะเห็นว่าการสแกนเริ่มจาก 1,2,3,4 เรียกว่าฟิลด์คี่ (Odd Field) และในระหว่างเส้นคือเส้นก็จะเว้นช่องว่างให้ทอสแกนได้อีกครั้งหนึ่ง จากนั้นก็จะเริ่มที่ 5,6,7 ใหม่อีกครั้งหนึ่ง เรียก ว่าฟิลด์คู่ (Even Field) เป็นการสแกนเส้นเว้นเส้น ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

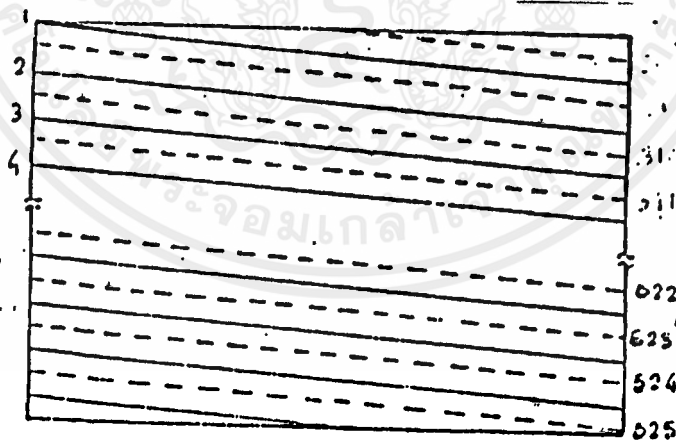
ซึ่งต้องใช้เวลาสแกนในแนวตั้ง 2 ครั้ง ดังนั้นถ้าต้องการส่งภาพ 25 ภาพ/วินาที ก็ต้องส่ง 50 ครั้ง (เป็นแบบระบบยุโรป) นั่นคือการส่งภาพแบบทยาย ๆ ไป 50 ภาพนั้นเองวิธีนี้เรา เรียกว่า วิธีการสแกนแบบสลับเส้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 2.4 การสแกนแบบสลับเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสแกนแบบสลับเส้นนี้ต้องใช้เวลาสแกนแนวตั้ง 2 ครั้ง การสแกนแนวตั้ง 1 ครั้งเรียกว่า การสแกน 1 ฟิลด์ และการสแกน 2 ฟิลด์ เรียกว่า 1 เฟรม (Frame) ในระบบ 625 เส้น จากรูป 2.4 (ข) การสแกน 1 ฟิลด์ มี 312.5 เส้น เนื่องจากตาของมนุษย์มีคุณสมบัติในการคงอยู่ของภาพ (Persistence of Image) และระยะเวลาในการเรืองแสงของฟอสเฟอร์ที่จอภาพ ทำให้ภาพของฟิลด์ที่หนึ่งยังคงอยู่ในขณะที่ฟิลด์สแกนเสร็จแล้ว ภาพที่มองเห็นจึงมีจำนวน 625 เส้น ข้อดีของวิธีนี้ทำให้ลดการกระพริบของภาพ (Flicker) ได้ถึงเท่าตัว

ในการสแกนทั้งแบบก้าวหน้าและแบบสลับเส้น เมื่อสแกนไปสุดของแต่ละเส้นแล้ว ต้องรีบกลับไปยังเส้นใหม่ ทั้งแนวตั้งและแนวนอน ระยะเวลาในการวิ่งกลับมาเริ่มใหม่น้อยกว่าหรือยิ่งดี จากรูป 2.3 เส้นประจาก b--->c , d--->e เส้นนี้เรียกว่าเส้นสแกนกลับ (Retrace or Flyback) เส้นนี้ไม่มีความจำเป็นในการประกอบเป็นภาพจึงมีวงจรควบคุมไม่ให้ปรากฏที่จอ

ในโครงการนี้จะทำการดิจิไทซ์ (Digitize) เพียงฟิลด์เดียว จากสัญญาณที่ส่งมาโดย วิธีการแบบสลับเส้น ดังนั้นก่อนที่จะทำการเก็บภาพ ภาพที่แสดงบนจอโทรทัศน์จะเป็นภาพของข้อมูลที่เก็บบนหน่วยความจำ และแสดงผลโดยวิธีการสแกนแบบก้าวหน้า

2.3. สัญญาณโทรทัศน์ (Composite Signal)

เพื่อจะทำให้เกิดภาพที่สมบูรณ์ สัญญาณโทรทัศน์จะต้องประกอบด้วยสัญญาณต่าง ๆ ดังนี้

2.3.1 สัญญาณภาพและสัญญาณเสียง (Video And Audio Signal)

เป็นสัญญาณที่ใช้เพื่อทำให้เกิดภาพและเสียงตามต้องการ ในโครงการนี้เราใช้เพียงสัญญาณภาพเพียงอย่างเดียว

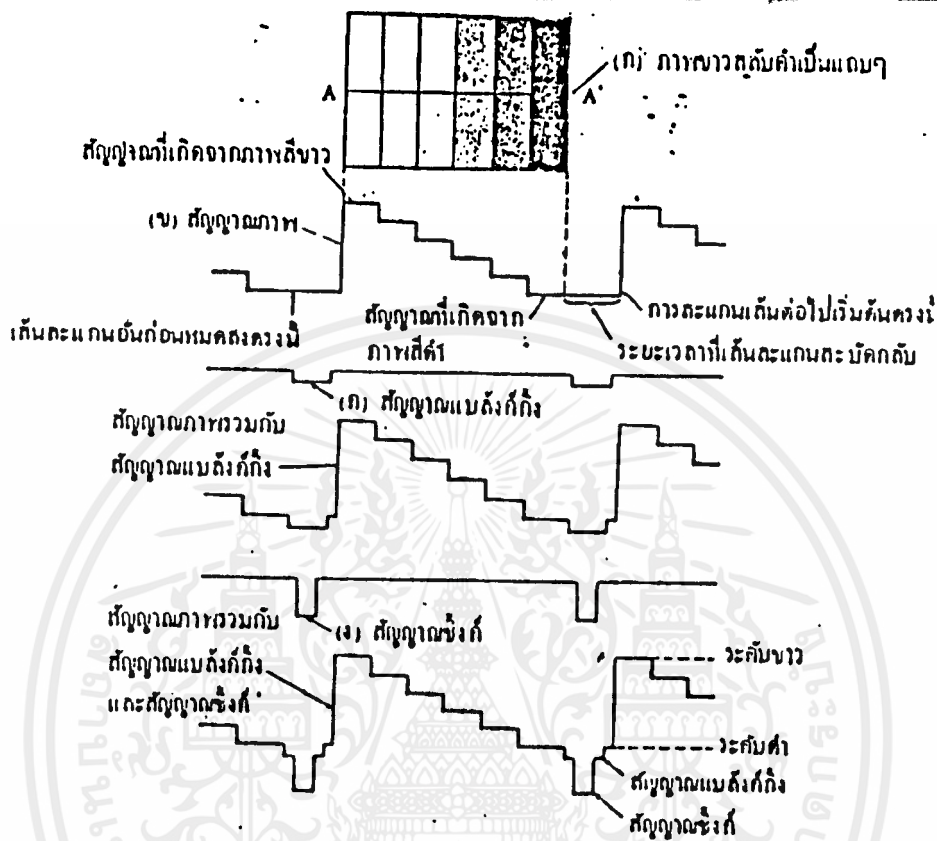
2.3.2 สัญญาณเบสลับ (Blanking Pulse)

เป็นสัญญาณที่ใช้เพื่อลบการสแกนสลับทั้งในแนวนอนและแนวตั้ง เพื่อมิให้สังเกตเห็นได้ชัดทางจอภาพ สำหรับโทรทัศน์ระบบอเมริกัน สัญญาณเบสลับระหว่างเส้นสแกน (เบสลับทางแนวนอน) จะมีขนาดประมาณ 10 ไมโครวินาที ทานองแนวเดียวกันสัญญาณเบสลับระหว่างฟิลด์ (เบสลับทางแนวตั้ง) จะมีขนาดประมาณ 1,250 ไมโครวินาที ส่วนโทรทัศน์ระบบยุโรปก็มีค่าประมาณนี้ (ดูรูปที่ 2.5)

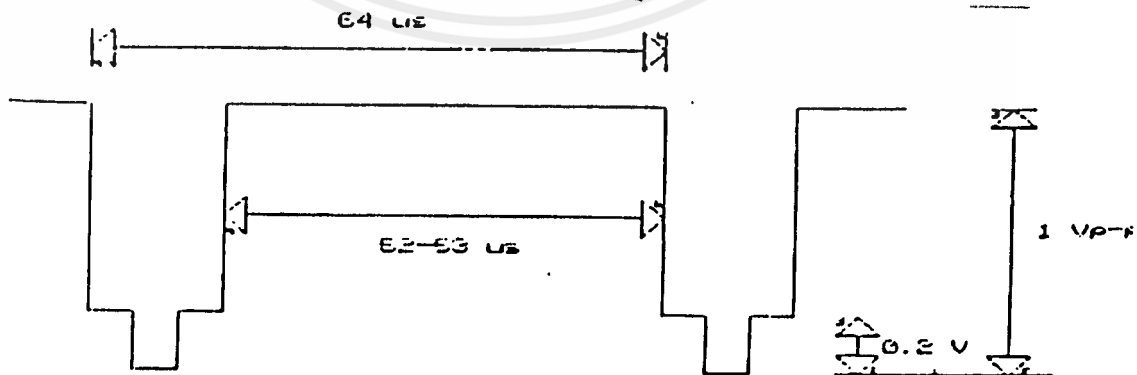
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 สัญญาณซิงค์

เป็นสัญญาณที่ใช้เพื่อทำให้วงจรถักเททางแนวนอนและแนวตั้ง (เรื่องราวเกี่ยวกับวงจรถักเทสามารถค้นคว้าได้จากตำราโทรทัศน์ทั่วไป) ในเครื่องส่งและเครื่องรับโทรทัศน์มีความถี่ตรงกันตลอดเวลา ในระบบยุโรปสัญญาณซิงค์ทางแนวนอนมีความถี่ 15,625 Hz ซึ่งเท่ากับความถี่ของวงจรถักเททางแนวนอน และสัญญาณซิงค์ทางแนวตั้งมีความถี่ 50 Hz ซึ่งเท่ากับความถี่ของวงจรถักเททางแนวตั้งเหมือนกัน เนื่องจากว่าความถี่ของสัญญาณซิงค์มีความเท่ากับความถี่ของสัญญาณเบสแก๊งค์พอสิต จึงจำเป็นต้องป้องกันการรบกวนที่อาจเกิดขึ้นโดยการกำหนดขนาดของซิงค์พัลส์ให้น้อยกว่าขนาดของเบสแก๊งค์พัลส์คือทำให้ซิงค์พัลส์ทางแนวนอนมีขนาดเพียง 5 ไมโครวินาที และซิงค์ทางแนวตั้งมีขนาดเพียง 190 ไมโครวินาที เท่านั้น นอกจากนี้ยังใช้วิธีส่งซิงค์เหล่านี้ไปกับเบสแก๊งค์พัลส์ด้วย โดยพื้นฐานของซิงค์พัลส์อยู่ที่ขอบบนของเบสแก๊งค์พัลส์อีกชั้นหนึ่ง เมื่อจัดขอบเขตความต่างศักย์ให้ระดับสูงสุดของเบสแก๊งค์พัลส์เป็นระดับต่ำจนมองไม่เห็นแล้ว ระดับของซิงค์พัลส์ที่อยู่บนยอดสูงสุดของเบสแก๊งค์พัลส์ก็จะควมอดสนิทไปด้วยและไม่ทำให้เกิดการรบกวนที่จอภาพแต่อย่างใด (ดูรูป 2.5, 2.6, 2.7)

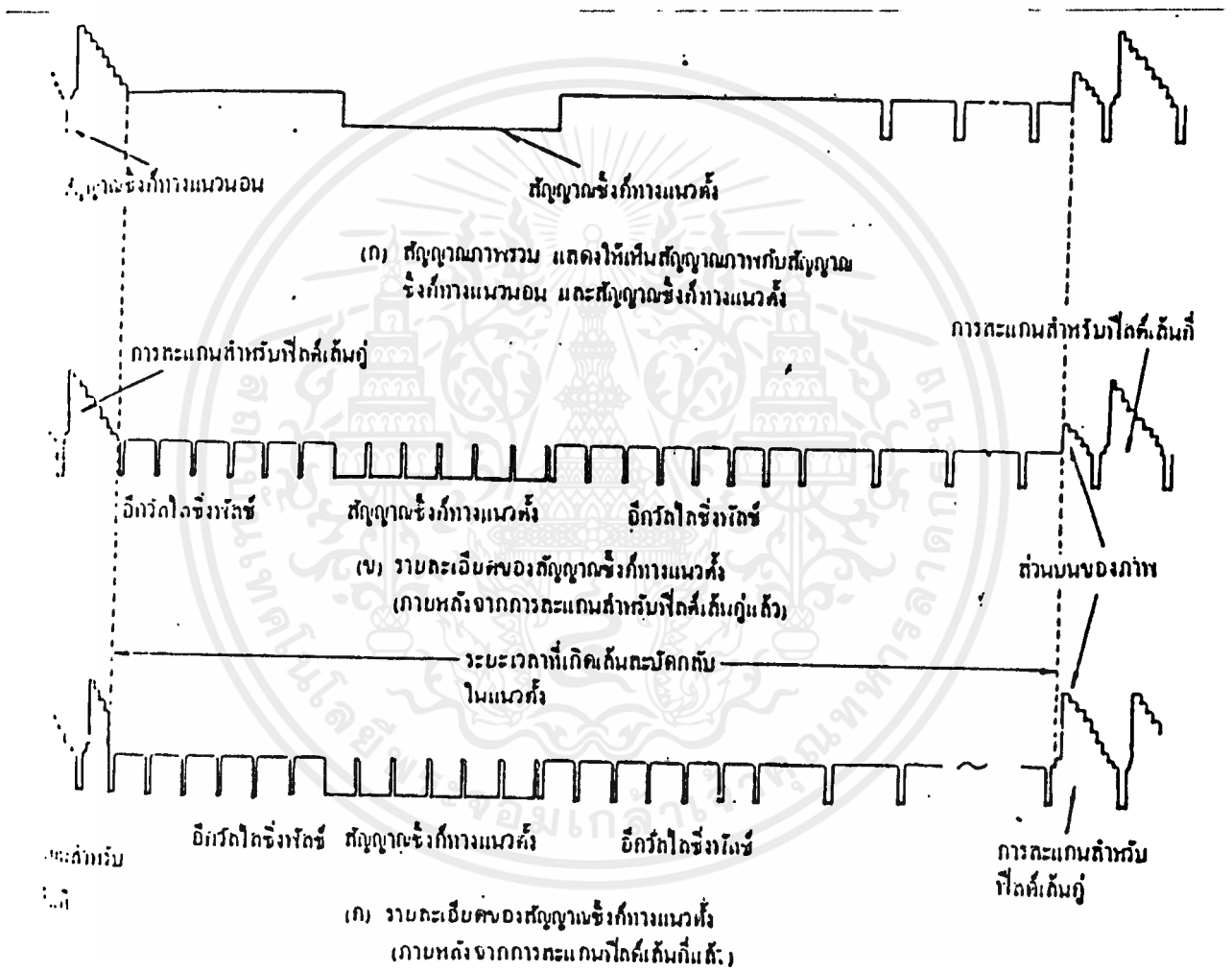


รูปที่ 2.5 รูปร่างของสัญญาณโทรทัศน์ที่เกิดจากภาพขาวสลับดำเป็นแถบ ๆ



รูปที่ 2.6 ขนาดและช่วงเวลาของสัญญาณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 สัญญาณภาพรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 สัญญาณอีควอลไลซิง (Equalizing Pluses)

เป็นสัญญาณที่ใช้เพื่อทำให้สัญญาณเชิงคทางแนวตั้ง ยังคงมีรูปร่างดีเหมือนเดิมหลังจากแยกออกจากสัญญาณเชิงคทางแนวนอนแล้ว นอกจากนี้ยังช่วยทำให้การสแกนไขว้กันเป็นไปโดยเรียบร้อยเสมอ รวมทั้งสัญญาณเชิงคทางแนวนอนก็ไม่เสียหายก็ไปในช่วงเวลาของสัญญาณเชิงคทางแนวตั้งอีกด้วย สัญญาณอีควอลไลซิงนี้มีความจำเป็นสำหรับการสแกนแบบสลับเส้น เพราะช่วยลดความผิดพลาดที่เกิดขึ้นจากการสับคสลับผิดตำแหน่ง ในโครงการนี้การสร้างสัญญาณโทรทัศน์ในขณะที่แสดงภาพของข้อมูลที่เก็บอยู่ในหน่วยความจำ (ภาพที่ดจิดจได้) จะไม่มีสัญญาณอีควอลไลซิงคนี้เนื่องจากเป็น วิธีการสแกนแบบก้าวหน้า ซึ่งจะทำให้ระบบวงจรง่ายขึ้น และยังประหยัดค่าใช้จ่ายอีกด้วย

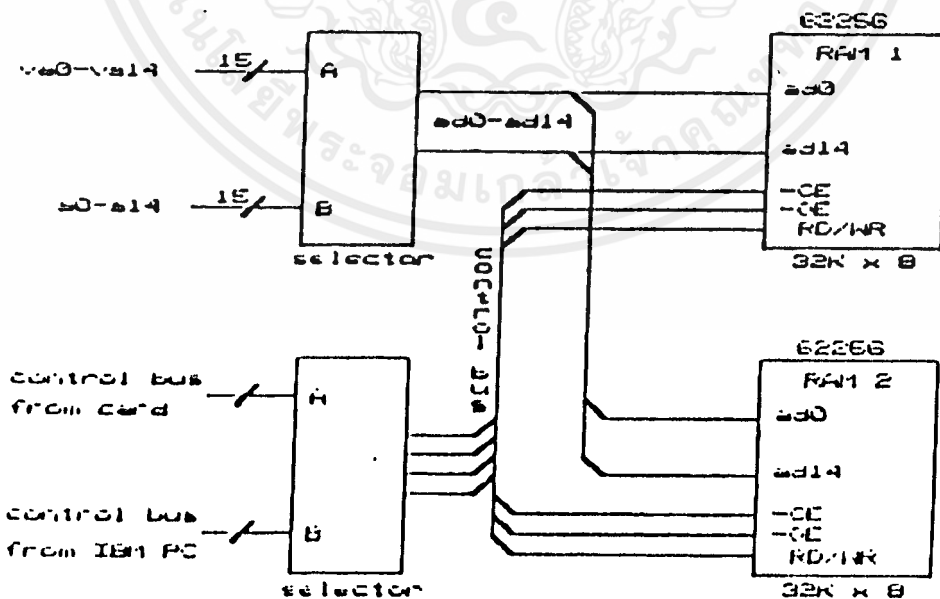
สัญญาณทั้งหมดที่กล่าวมานี้ เมื่อนำมารวมกันเราเรียกว่า สัญญาณโทรทัศน์ (Composite Signal)

บทที่ 3

การจัดการหน่วยความจำ (Memory management)

บนการ์ดจะมีหน่วยความจำ 64 กิโลไบต์ แยกออกจาก Mainboard ของ IBM PC/XT โดยใช้ RAM เบอร์ 62256 ซึ่งเป็น 32 K*8 Static RAM จำนวน 2 ตัว เหตุที่ต้องใช้หน่วยความจำถึง 64 กิโลไบต์ เนื่องจากต้องการเก็บจุดภาพ (Pixel) 256*256*8 บิตในหนึ่งฟิลด์ ในปัจจุบันราคาของ Static RAM ใกล้เคียงกับราคาของ Dynamic RAM ไม่ต่างกันมากนัก และข้อดีอีกข้อหนึ่งที่เลือกใช้ Static RAM คือไม่ต้องทำการ Refresh หน่วยความจำ การอ้างตำแหน่งก็ไม่ยุ่งยาก ทำให้ง่ายและไม่ซับซ้อน

การอ้างตำแหน่งของหน่วยความจำจะต้องทำการ Multiplex ในการดึงใช้ IC เบอร์ 74LS157 และ 74LS257 ซึ่งเป็น Selector ทำการเลือกสัญญาณจาก Address Bus และสัญญาณควบคุม (Control Signal) จากการ์ดหรือ IBM PC/XT ออกไปที่ RAM ทั้งสองตามรูป 3.1



3.1 การอ้างตำแหน่งแอดเดรสของ IBM PC/XT

ในการอ้างตำแหน่งแอดเดรสของ IBM PC/XT จะต้องบอกค่าสองค่าคือ เซกเมนต์ และออฟเซต (Segment And Offset) โดยเขียนในรูป Segment:Offset ทั้งเซกเมนต์และออฟเซตเป็นข้อมูลขนาด 16 บิต การคำนวณตำแหน่งที่แท้จริงทำได้โดยขยับทางซ้าย (Shift left) 4 บิต แล้วนำมารวมกับค่าออฟเซตจะได้ค่าแอดเดรสขนาด 20 บิต ดังตัวอย่าง

ตัวอย่าง การคำนวณตำแหน่งแอดเดรสของหน่วยความจำ

D0000

+

0FFFF

DFFFF

3.2 การตีโค้ด (Decode) ตำแหน่งแอดเดรสของหน่วยความจำ

การตีโค้ดตำแหน่งแอดเดรสของการ์ดนี้ใช้เพียงค่าเซกเมนต์ เนื่องจากหน่วยความจำที่ต้องการตีโค้ดเป็น 64 กิโลไบต์ ค่าเซกเมนต์ที่ใช้คือ D0000 H ซึ่งเป็นเซกเมนต์ที่วางอยู่ (IBM PC/XT Technical Reference) ให้ RAM 1 เป็นหน่วยความจำ 32 กิโลไบต์ แรกและ RAM 2 เป็นหน่วยความจำ 32 กิโลไบต์ถัดมา เพราะฉะนั้น RAM1 มีค่าแอดเดรสอยู่ช่วง 0000H-7FFFH และ RAM2 จะมีออฟเซตแอดเดรสอยู่ในช่วง 8000H-FFFFH ดังนี้

Address	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAM1	1	1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
RAM2	1	1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x : don't care

เมื่อเราต้องการติดต่อกับ RAM ตัวใดตัวหนึ่งจะต้องให้แอดเดรสและต้องทำการ Enable ขาสัญญาณควบคุม 3 ขาคือ CE (Chip Enable) ซึ่งใช้สัญญาณจากการตีโค้ดแอดเดรส, ขา OE (Output Enable), ขา RD/WR (Read Or Write)

3.3 การตีโค้ดตำแหน่งแอดเดรสของอินพุทเอาต์พุทพอร์ท

ในการอ้างตำแหน่งแอดเดรสของพอร์ทต่างๆ บน IBM PC/XT จะใช้บิตแอดเดรสเพียง 10 บิต เท่านั้นคือ A0-A9 จะได้แอดเดรสของพอร์ทสูงสุดทั้งหมด 1,024 พอร์ท ซึ่งจะแบ่งออกเป็นกลุ่มสองกลุ่มคือ

3.3.1 เมื่อบิต A9 เป็น 0 จะเป็นพอร์ทที่อยู่บน Mainboard เท่านั้น ได้แก่พอร์ท 0-1FFH (512พอร์ท)

3.3.2 เมื่อบิต A9 เป็น 1 จะเป็นพอร์ทที่อยู่บนการ์ดต่างๆ ได้แก่พอร์ท 200H-3FFH (512 พอร์ท เช่นกัน)

บนการ์ดเราใช้พอร์ท 300H-31FH ซึ่งเป็นพอร์ทที่ IBM PC/XT เตรียมไว้ให้การ์ดต่างๆ ใช้ได้แต่จริงแล้วต้องการใช้เพียงแอดเดรสเดียว แต่เพื่อความประหยัดของอุปกรณ์ที่ใช้ในการตีโค้ดแอดเดรสจึงต้องทำเช่นนี้ ในการใช้งานร่วมกับการ์ดอื่นๆ จึงควรระวังกรณีพอร์ทซ้ำกันด้วย

Address	9	8	7	6	5	4	3	2	1	0
300 H	1	1	0	0	0	0	0	0	0	0
31F H	1	1	0	0	0	1	1	1	1	1

3.4 PAL (Programmable Array Logic)

สำหรับโครงการนี้ได้นำเอา IC PAL มาใช้แทนวงจรถูกต่างๆ ในวงจรถูก ซึ่ง IC PAL ที่ใช้ในโครงการนี้ใช้เบอร์ PAL 16L8 การใช้ IC PAL เข้ามาแทนวงจรถูกต่างๆในวงจรถูกก็เพื่อเป็นการประหยัดค่าใช้จ่าย และลดความยุ่งยากในการออกแบบ และในโครงการนี้สามารถใช้ IC PAL เพียงตัวเดียวแทนเกททั้งหมดได้ โดยต้องนำเอาฟังก์ชันของวงจรถูกที่ออกแบบไว้ มาทำการก๊อปปี้ลงบนตัว IC PAL โดยการใส่เครื่องมือช่วยซึ่งในโครงการนี้ได้ทำการออกแบบตามฟังก์ชันต่อไปนี้

$$/Q1 = IOW + Y1$$

$$/Q2 = IOR + Y1$$

$$/Q3 = MEMR * Q2$$

$$/Q4 = Q1 * Q2 * YA6 * YA7$$

$$/Q5 = MEMW * MEMR$$

$$/Q6 = /P + YA6 * YA7$$

$$/Q7 = /P * VBL * HBL$$

16L8.JED

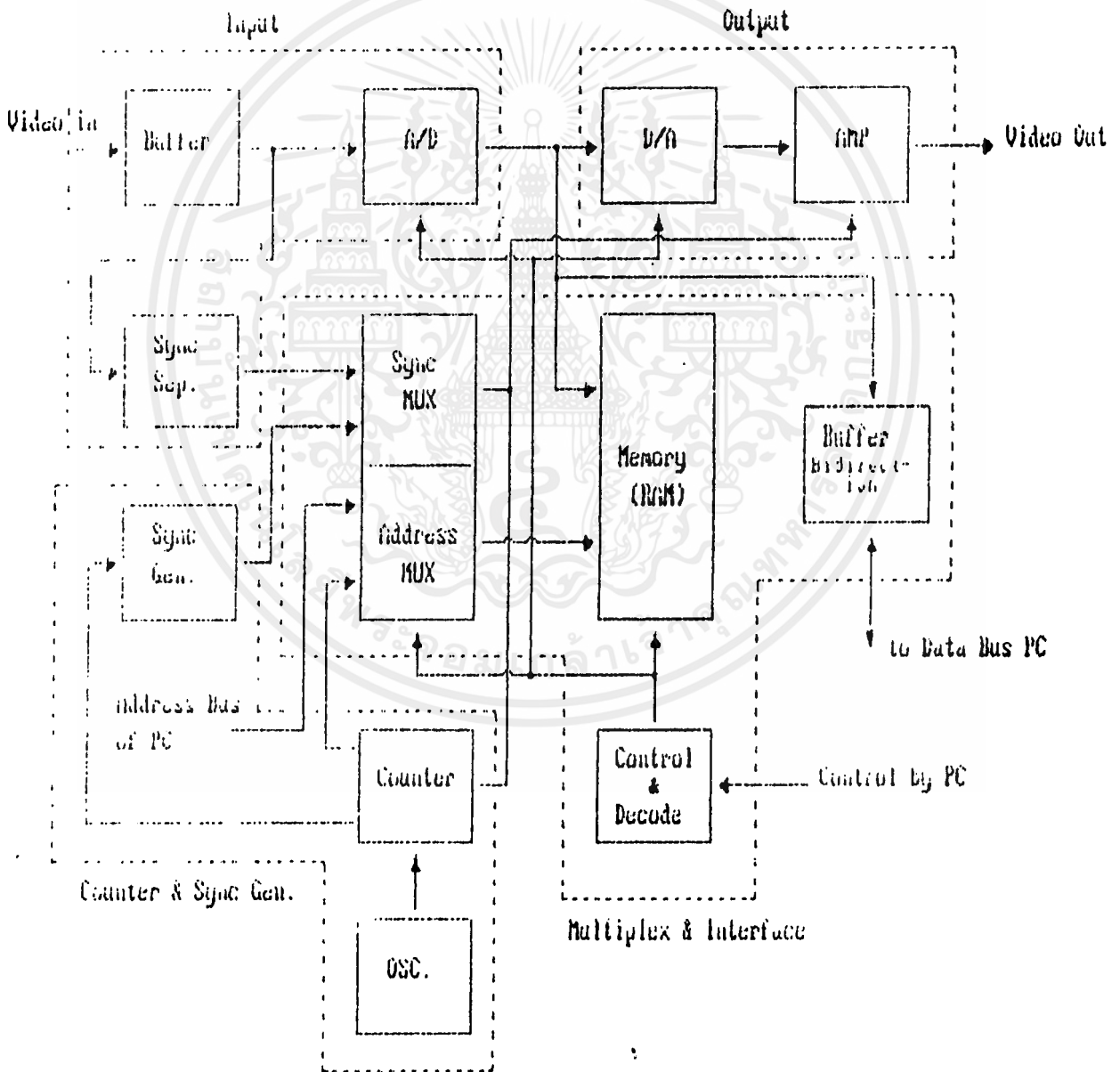
IOW---	1	20	---VCC
Y1---	2	19	---Q5
IOR---	3	18	---
MEMR---	4	17	---Q6
YA6---	5	16	---Q2
YA7---	6	15	---Q1
MEMW---	7	14	---Q4
P---	8	13	---Q3
VAL---	9	12	---Q7
GND---	10	11	---HBL

PAL16L8

บทที่ 4

ขั้นตอนการออกแบบและทดลอง

1. บล็อกไดอะแกรม ของ Image Interface Card



Block Diagram of Image Interface Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อคอโคเดแกรมที่ได้ เราก็นำมาทำการออกแบบ และสร้างเป็นวงจรใช้งาน โดยจะแยกออกเป็นส่วน ๆ ได้ 4 ส่วน คือ

1. ส่วนที่เป็นภาคอินพุทจะประกอบด้วย Buffer, Sync Seperator และส่วนของ A/D converter

2. ส่วนที่เป็นตัวนับและสร้างสัญญาณซิงค์ (Counter and Sync Generator)

3. ส่วนของภาคเอาต์พุท

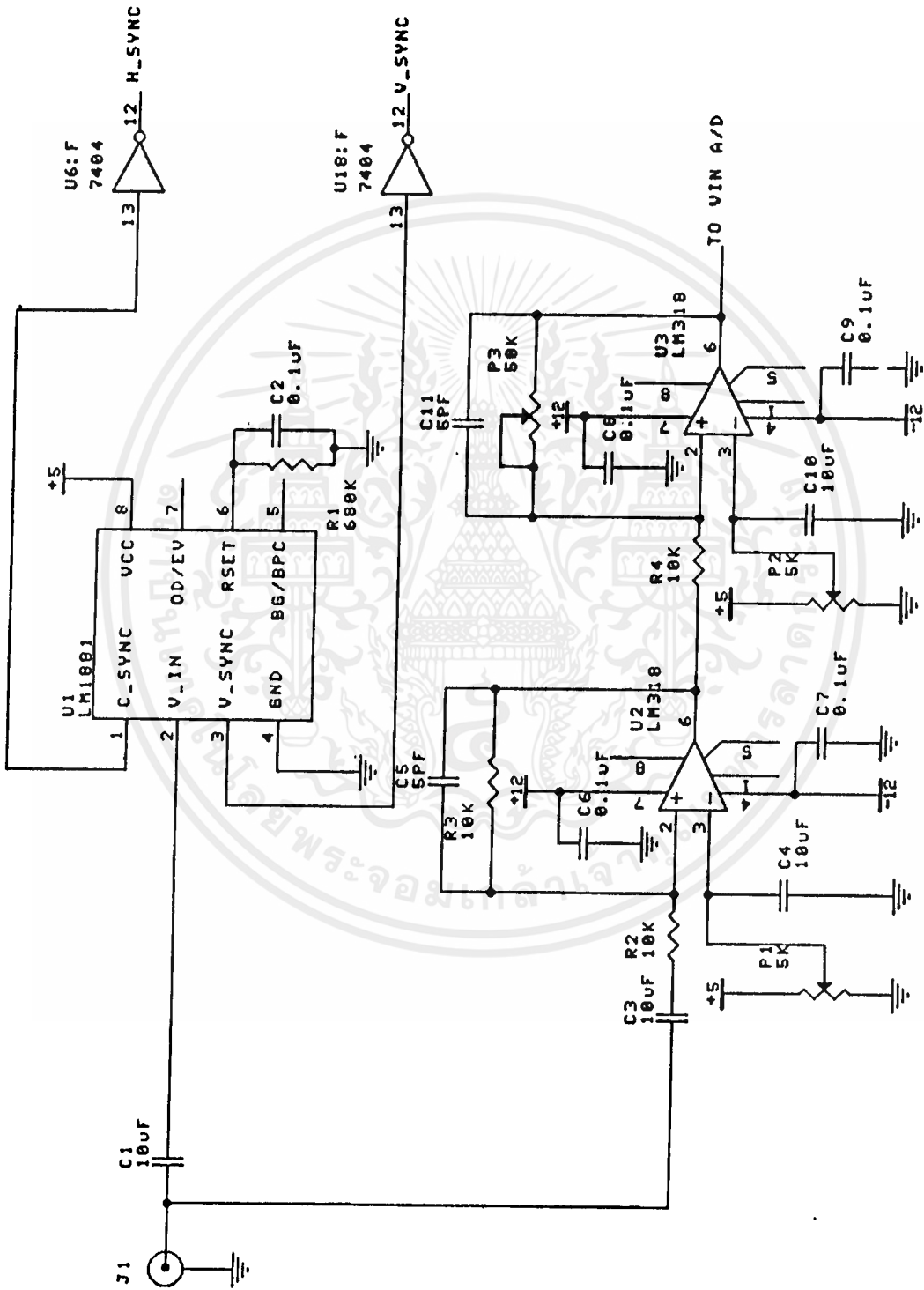
4. ส่วนของการผลิตเพสิสกีชและการอินเตอร์เฟส

การออกแบบภาคอินพุท

1. Buffer จะเป็นส่วนที่รับสัญญาณ composite video เข้ามาประกอบด้วย Q1 C1, C2, R3 และ R4 โดยจะมี R1, R2 และ D1 เป็นตัวต่อรวมช่วยยกระดับสัญญาณภาพให้ไว้ที่ขั้วอินพุทในระดับไฟตีส 0.2 V เอาท์พุทที่ออกจากขา E ของ Q1 จะส่งต่อไปยังส่วนของตัวแยกซิงค์และส่วนของ A/D converter

2. ส่วนของการแยกซิงค์ (Sync Seperator) เราใช้ IC เบอร์ LM 1881N เป็นตัวแยกสัญญาณซึ่งออกจากสัญญาณภาพ สัญญาณที่ออกจากขา 1 ของ U1 เป็นสัญญาณซิงค์รวม (Composit Sync) และสัญญาณที่ออกจากขา 3 ของ U1 จะเป็นสัญญาณซิงค์ทางแนวตั้ง (Vertical Sync)

3. A/D converter ในส่วนนี้จะประกอบไปด้วย ตัวแปลงสัญญาณจากอนาล็อกเป็นสัญญาณดิจิทัลโดยเราจะต้องใช้ IC ที่มีความเร็วสูงพอที่จะทำการแปลงสัญญาณในการสแกนของสัญญาณภาพจะใช้เวลา 51.2 μ S ให้ได้ 256 จุด ซึ่งการแปลงแต่ละครั้งจะใช้เวลา 51.2μ S/256 = 0.2 μ S มีความถี่เท่ากับ 5 MHz ซึ่งเราจะใช้ IC CA 3006 UB ซึ่งจะมีขนาด 6 บิต (Resolution = 64 ระดับ) ส่วนแรงดัน reference นั้นเราเราจะใช้วงจรดิคโคเนชันประกอบด้วย Q1, Q2, R11, 12, 13, 14, 15 โดยมี VR8, VR9 เป็นตัวปรับระดับอ้างอิงตามที่ต้องการ โดยปกติจะไว้ที่ขั้วอินพุท 0.2 ถึง 2.5 V สัญญาณดิจิทัลที่ได้จากภาคนี้จะส่งเข้าสู่หน่วยความจำบนการ์ด และอีกส่วนหนึ่งจะถูกส่งออกไปภาคเอาต์พุทเพื่อทำการแสดงผลไปพร้อม ๆ กัน



Title		Input and Sync_Sep
Size	A4	Revision
Date:	19-MAR 1993	Sheet
File:	A:\RIMG1/1	Drawn By:

รูป 6.2 INPUT AND SYNC_SEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

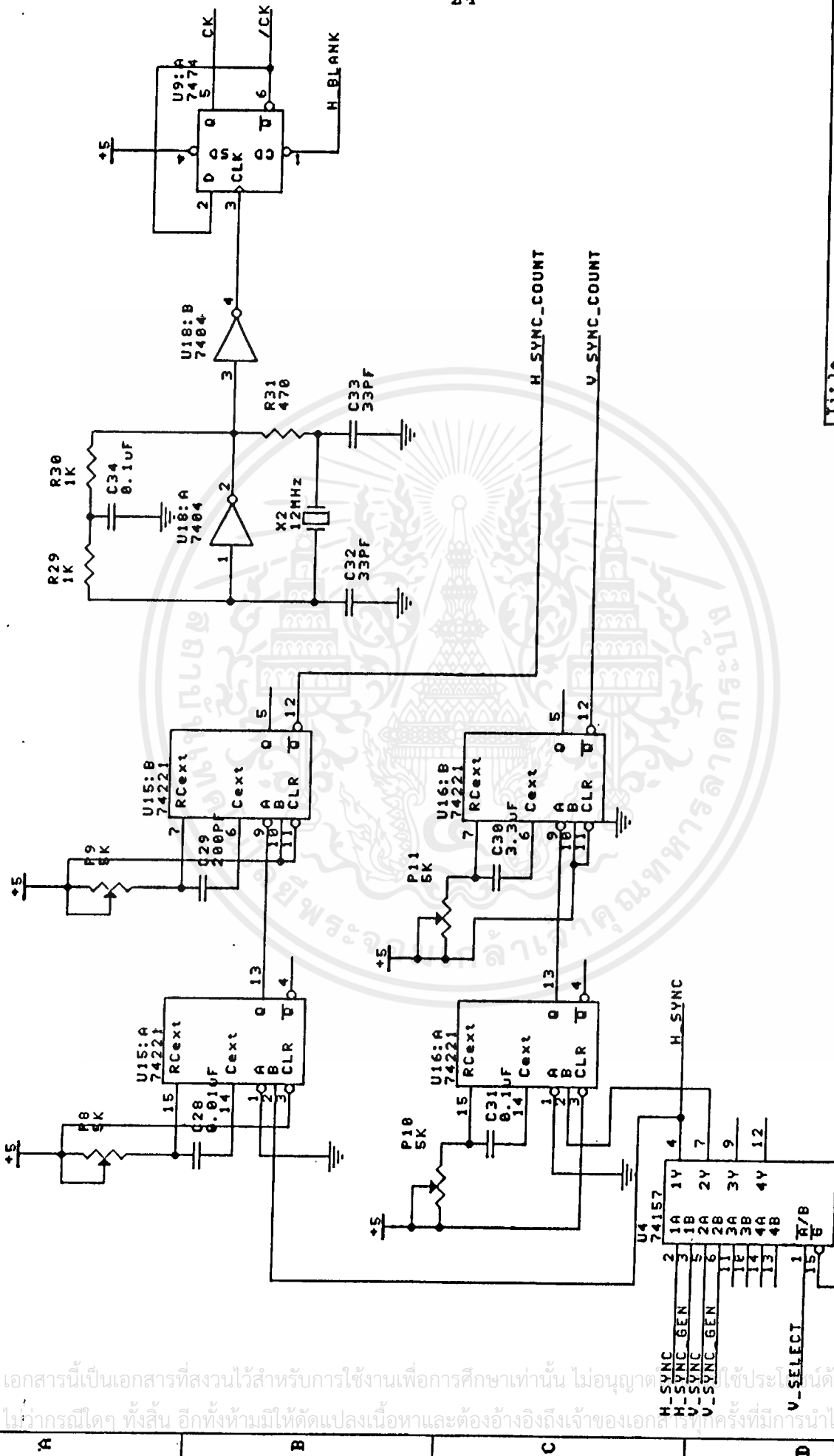
การออกแบบตัวนับและตัวสร้างซิงค์ (Counter and Sync Generater)

1. Counter เราจะออกแบบให้เป็นตัวนับขนาด 16 บิตที่จะสามารถอ้างหน่วยความจำได้ 64 Kbytes ซึ่งจะใช้ในช่องของการดิจิทัล ซึ่งเราจะใช้ความถี่ในการนับ 6 MHz โดยจะมีซิงค์มาควบคุมการนับให้ได้ตำแหน่งของการเก็บ Pixel ลงในหน่วยความจำได้ถูกต้อง ในที่นี้เราใช้ IC 74393 U21,U22 ซึ่งสัญญาณ Va0-Va15 จะนำไปใช้ในการอ้างแอดเดรสของแรมและสัญญาณ VA7 กับ VA15 ของตัวนับ U21 และ U22 จะถูกส่งไปให้ตัวผลิตสัญญาณซิงค์และสัญญาณเบลนคิง ซึ่งการนับจะถูกควบคุมจากสัญญาณซิงค์ที่ออกจากตัวมัลติเพล็กซ์

2. Sync Generater ในส่วนนี้จะผลิตสัญญาณซิงค์โดยใช้ IC เบอร์ ZNA 234E เป็น U5 ซึ่งจะทำการผลิตสัญญาณซิงค์รวม (Composit Sync) ออกมาที่ขา 3 ซึ่งสัญญาณซิงค์รวม (Composit Sync) ที่สร้างจาก IC U5 นี้ จะนำไปผ่านวงจรกรองความถี่ต่ำ ผ่านเพื่อกรองเอาสัญญาณซิงค์ทางแนวตั้ง (Vertical Sync) และสัญญาณทั้งสองจะถูกส่งไปยังภาค Multiplex

การออกแบบส่วนเอาต์พุต (Output)

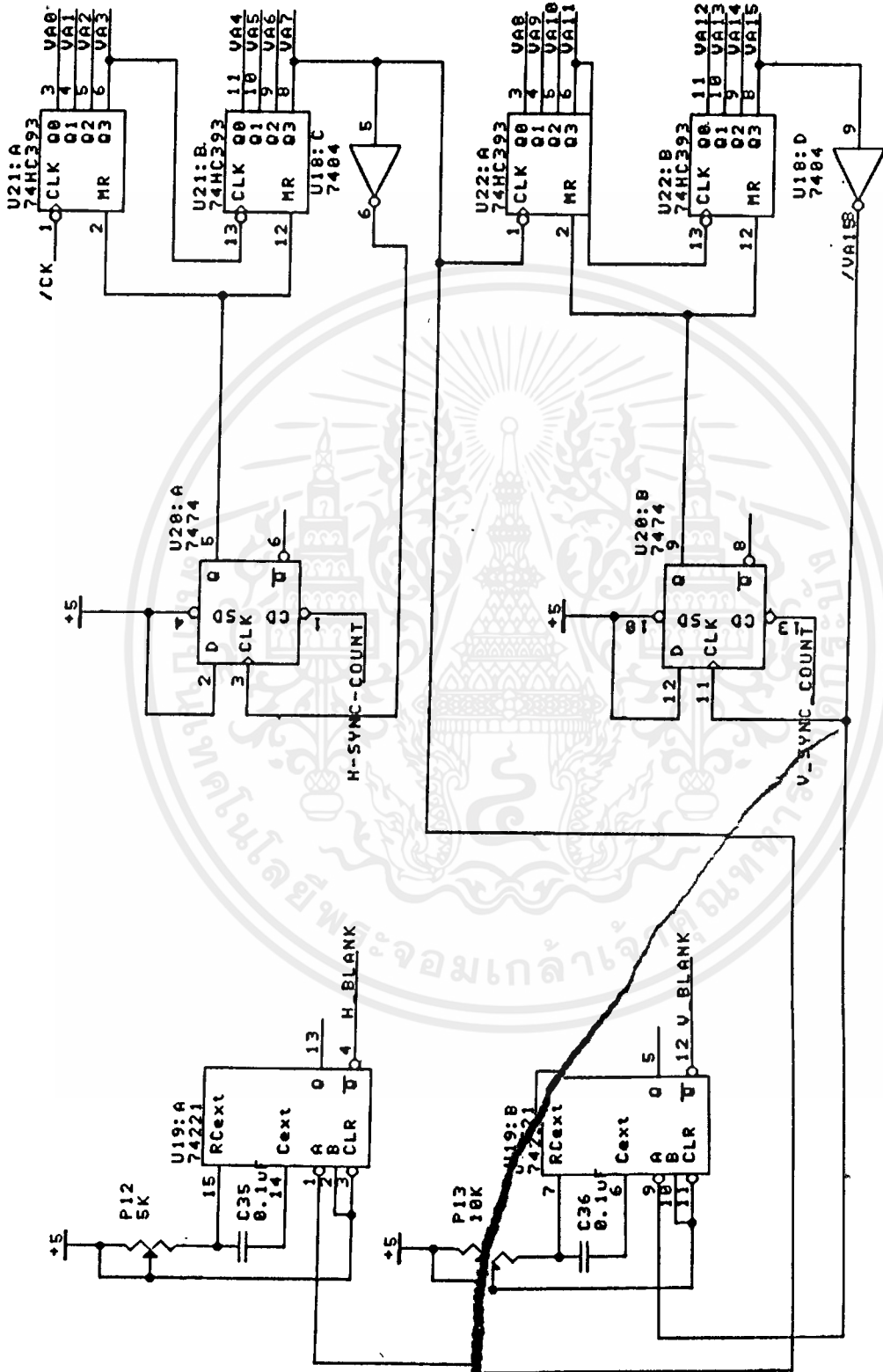
ในส่วนนี้จะประกอบไปด้วยตัวแปลงข้อมูล 74273 U11 จะทำการแปลงข้อมูลดิจิทัลที่ส่งออกมาจากหน่วยความจำ จากนั้นก็จะทำการแปลงเป็นสัญญาณอนาล็อกโดยใช้ IC DAC 0800 U12 ซึ่งจะได้ออกมาเป็นรูปของกระแส I_{out} ออกมาที่ขา 4 ซึ่งเราก็จะทำการแปลงให้เป็นแรงดันโดยผ่านวงจร V to I ในที่นี้เราใช้ LM318 U13 ต่อร่วมกับ VR6 ซึ่งเราจะได้สัญญาณภาพออกมาแล้วนำไปรวมกับสัญญาณซิงค์ที่เข้ามาที่ B ของ Q4 กลายเป็นสัญญาณ composite ออกมาที่ขา E ของ Q3



รูป 6.4 CONTROL COUNTER AND CLOCK

Title Control counter and Clock	
Size A4	Revision
Date: 19-MAR-1993	Sheet of
File: R:\RIMG3\1	Drawn By:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

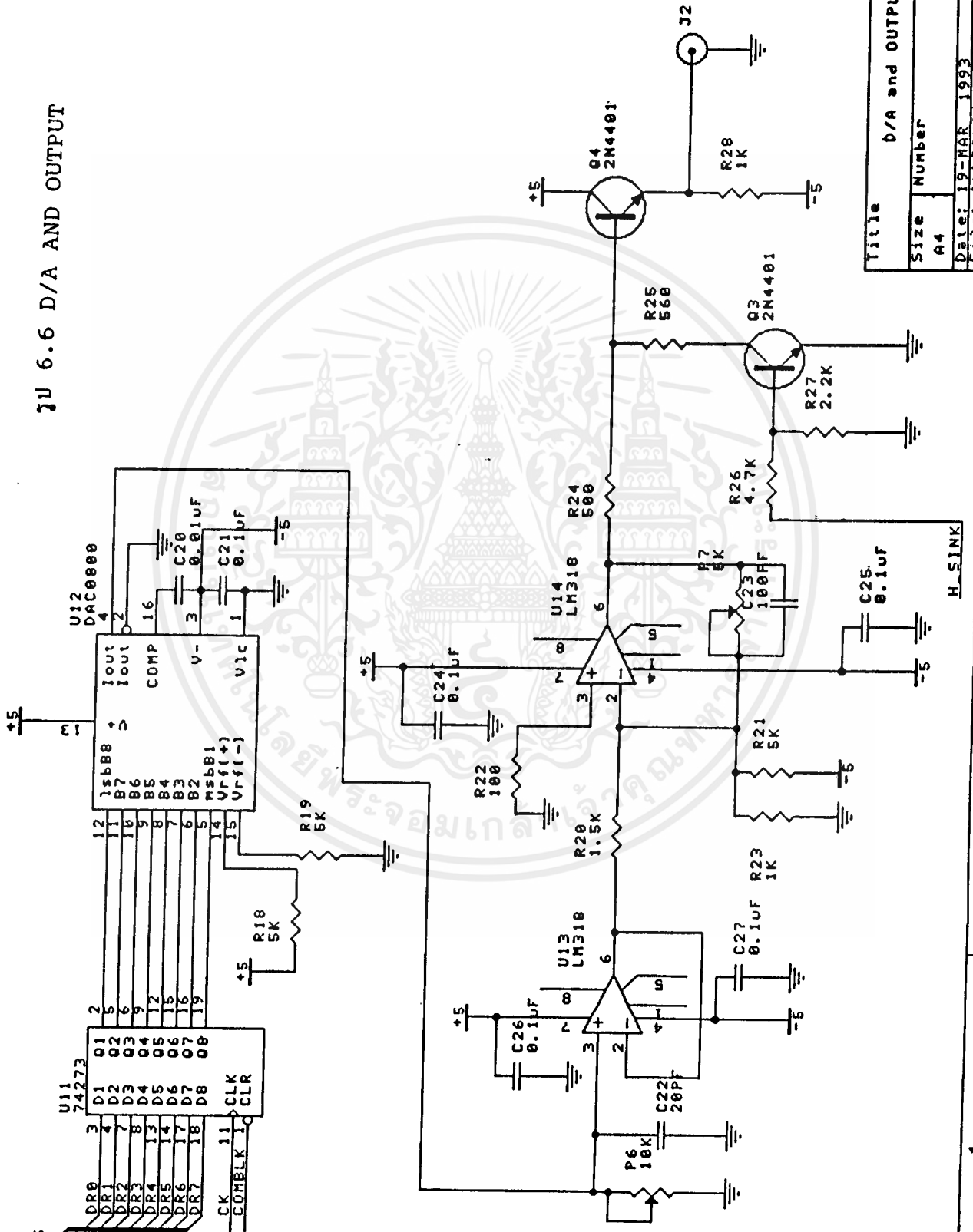


3U 6.5 COUNTER AND BALNK GEN

Title		Counter and Balnk Gen	
Size	Number	Revision	
A4			
Date:	19-MAR-1993	Sheet	of
File:	A:\RINGS\1	Drawn	By:

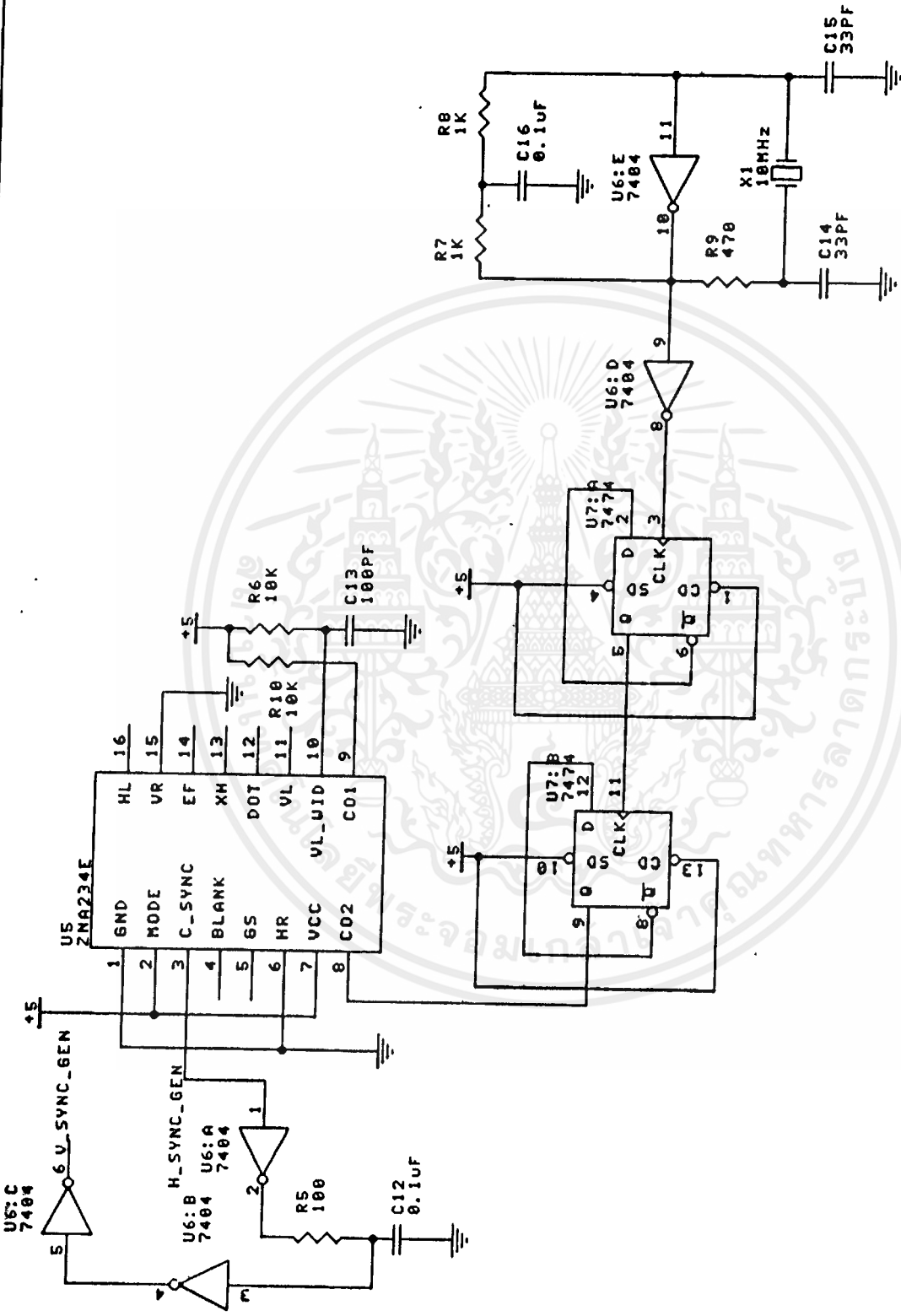
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7U 6.6 D/A AND OUTPUT



Title		D/A and OUTPUT	
Size	Number	Revision	
A4			
Date:	19-MAR-1993	Sheet	of
File:	A:\RIMG66/1	Drawn	By:
			4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
 การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมายและต้องอ้างถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 6.7 SYNC_GENERATOR

Title: Sync_Generator	
Size: A4	Revision:
Date: 19-MAR-1993	Sheet: of
File: A:RING771	Drawn By: 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบส่วนมัลติเพล็กซ์และส่วนอินเตอร์เฟส (Multiplex and Interface)

1. ส่วนมัลติเพล็กซ์ จะแบ่งออกเป็น 3 ส่วน คือ

1.1 ส่วนมัลติเพล็กซ์สัญญาณเชิงคีย์โดยจะเลือกการมัลติเพล็กซ์ว่าจะใช้เชิงคีย์จากสัญญาณ composite จากอินพุตหรือเชิงคีย์จากภาคกำเนิดเชิงคีย์ ในกรณีของการตีจ๊อบเราจะรับเชิงคีย์จากอินพุต และในกรณีของการแสดงภาพจากหน่วยความจำเราจะใช้เชิงคีย์จากภาคกำเนิดเชิงคีย์ ซึ่งในการออกแบบเราใช้ IC 74157 U4 เป็นตัวมัลติเพล็กซ์โดยมีสัญญาณ "select" เป็นตัวควบคุมการมัลติเพล็กซ์

1.2 ส่วนมัลติเพล็กซ์แอดเดรส จะเป็นการมัลติเพล็กซ์เลือกระหว่างแอดเดรสที่จะติดต่อกับหน่วยความจำนการ์คว่าจะเป็นการติดต่อกับคอมพิวเตอร์หรือติดต่อกับแอดเดรสของตัวนับ ถ้าเป็นช่วงการตีจ๊อบเราจะเลือกการติดต่อกับแอดเดรสของตัวนับคือ VA0-VA15 โดยมีสัญญาณ "P_control" เป็นตัวควบคุมการมัลติเพล็กซ์ ใช้ IC 74157 U23,24 25 และ U26 เป็นตัวมัลติเพล็กซ์

1.3 ส่วนมัลติเพล็กซ์สัญญาณควบคุม โดยส่วนนี้จะเป็นการเลือกสัญญาณควบคุมการอ่านเขียนหน่วยความจำ ใช้ IC 74257 U17 เป็นตัวมัลติเพล็กซ์

2. ส่วนอินเตอร์เฟส ประกอบด้วยส่วนของการตีจ๊อบ 2 ส่วน คือ

2.1 ส่วนของการตีจ๊อบตำแหน่งของพอร์อินพุตเอาต์พุตที่จะใช้ควบคุมการตีจ๊อบพอร์อินพุต, เอาต์พุตของการตีจ๊อบจะมีตำแหน่งอยู่ ระหว่าง 310H - 31FH

Adress	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
310H-31FH	1	1	0	0	0	0	X	X	X	X

2.2 ส่วนของการตีจ๊อบตำแหน่งของหน่วยความจำ หน่วยนี้จะเลือกให้อยู่ในช่วงที่นอกเหนือจากหน่วยความจำของระบบ คือเหนือจากช่วง 640 K ขึ้นไป โดยเลือกตำแหน่งของหน่วยความจำนการ์คอยู่ที่ D0000H - DFFFFH (64 K)

Address	A15	A14	A13	A12	A11	.	.	.	A1	A0
RAM1	1	1	0	1	0	X	X	X	X	X
RAM2	1	1	0	1	1	X	X	X	X	X

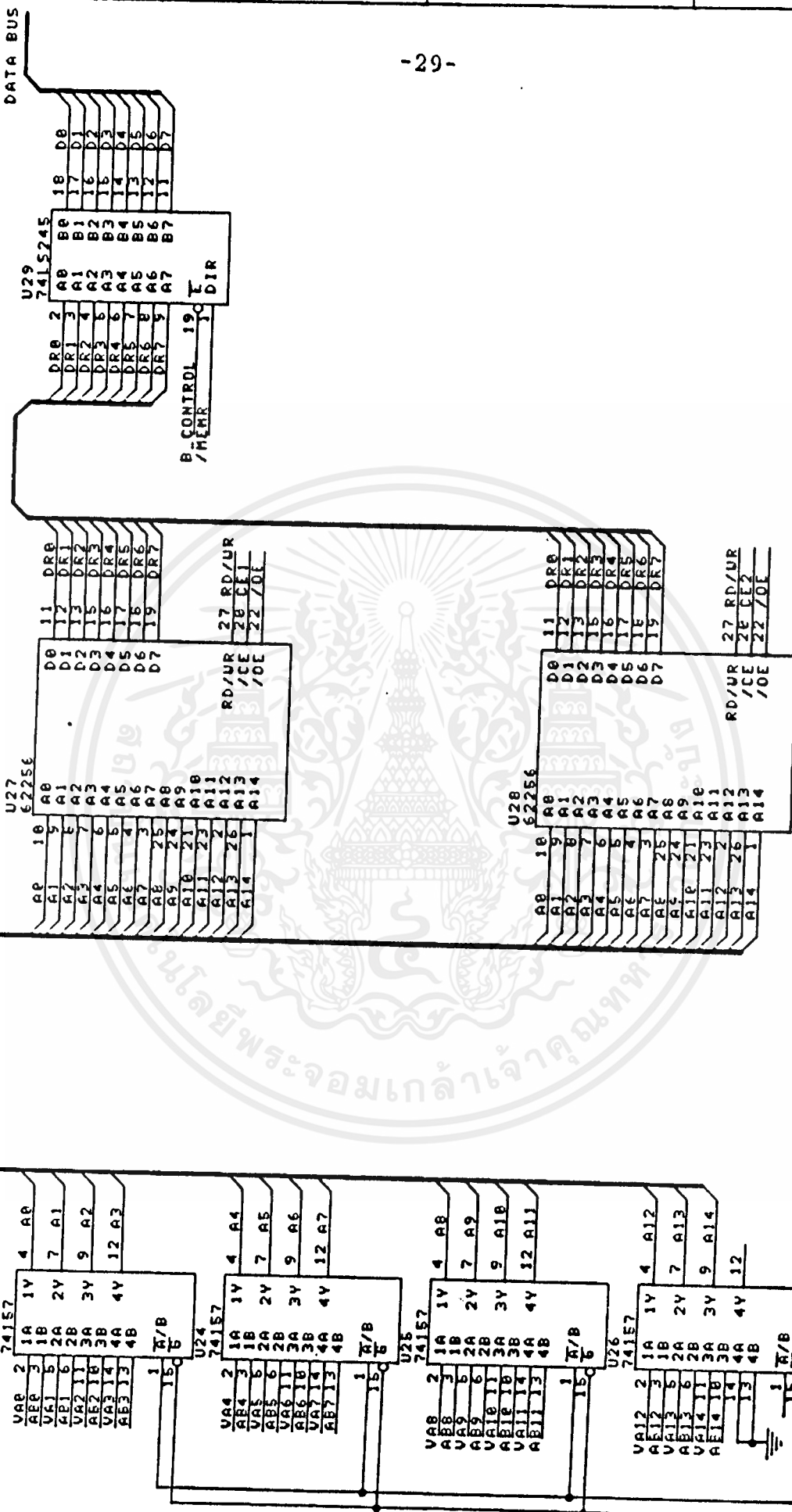
ซึ่งหน่วยความจำนการ์คนี้เราจะใช้ Static Ram ขนาด 8x32K 2 ตัว RAM1 จะมีตำแหน่งตั้งแต่ D0000H - D7FFFH RAM2 ตำแหน่ง D8000H - DFFFFH ด้านการคำ

1

2

3

4



รูป 6.8 ADDRESS MULTIPLEX

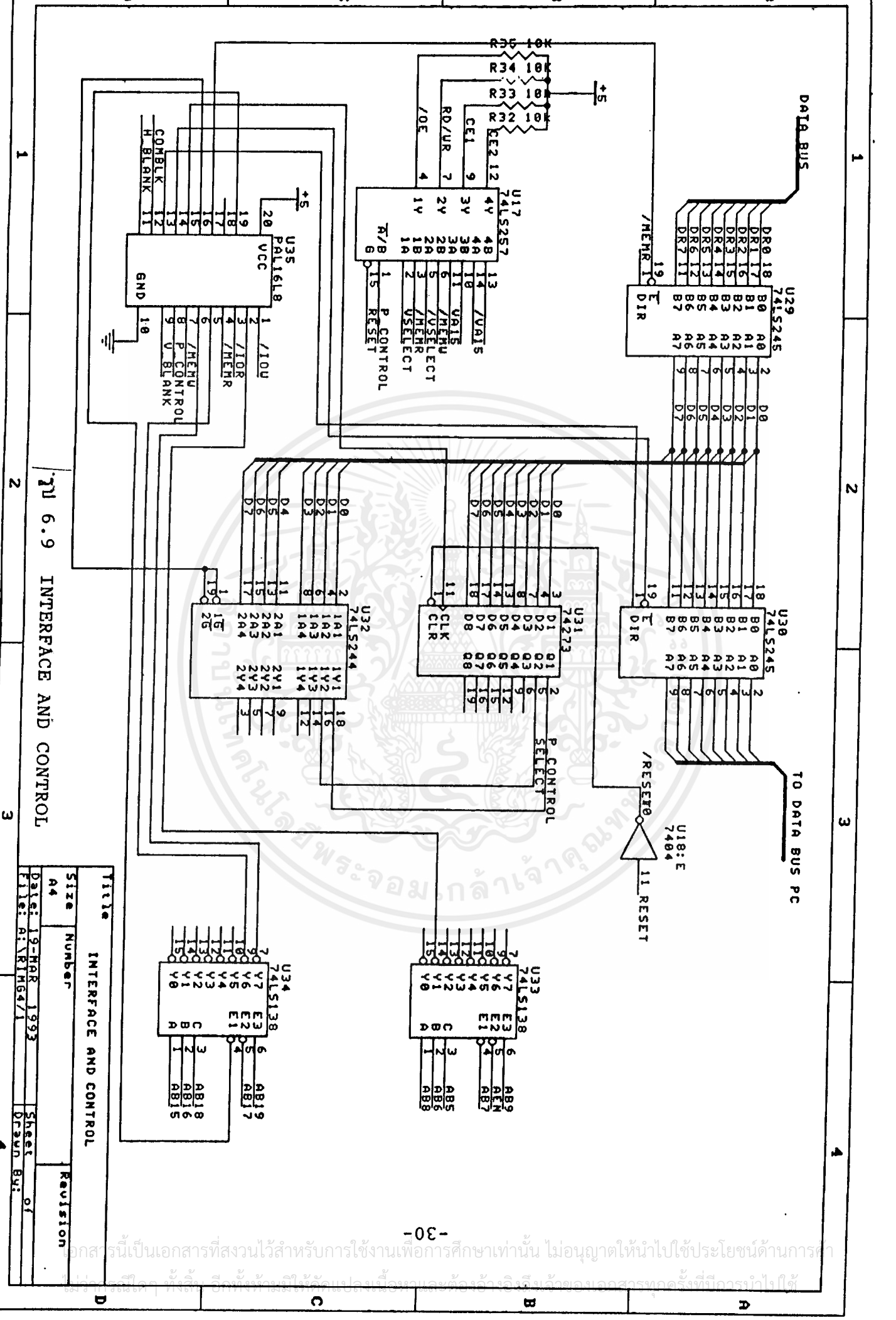
Title		ADDRESS MULTIPLEX	
Size	Number	Revision	
A4			
Date:	19-MAR-1993	Sheet	of
File:	ARXIMG671	Drawn	By:

1

2

3

4



7416.9 INTERFACE AND CONTROL

7416.9 INTERFACE AND CONTROL

Title	7416.9 INTERFACE AND CONTROL	Revision
Size	A4	Sheet
Number		of
Date:	19-MAR 1993	Drawn By:
File:	A:\KIMG4\1	

บทที่ 5

การใช้งานการ์ดและการทำงานของการ์ด

5.1 ส่วน HARDWARE

การใช้งานการ์ดแบ่งออกเป็น 2 อย่าง

5.1.1 การดึงวิดีโอภาพเป็นข้อมูลเก็บลงในหน่วยความจำ ชั้นแรกต้องทำการส่งค่า 00H ออกสู่พอร์ต 310H เป็นการสั่งให้การ์ดทำการดึงวิดีโอ การ์ดก็จะทำการรับสัญญาณ composite video เข้ามาทำการแยกซิงค์และแปลงเป็นสัญญาณอนาล็อก เก็บไว้ในหน่วยความจำบนการ์ดโดยใช้ VA0-VA15 ของตัวนี้ทำการย้ายแอดเดรสในการเก็บข้อมูล โดยข้อมูลดิจิทัลส่วนหนึ่งจะส่งออกไปที่ภาคเอาต์พุตแสดงภาพออกสู่จอมอนิเตอร์ ซึ่งเมื่อได้ภาพที่ต้องการก็ให้ทำการส่งข้อมูล 02H ออกสู่พอร์ต 310H ซึ่งเป็นการสั่งให้หยุดทำการดึงวิดีโอและการ์ดก็จะทำการแสดงภาพที่ได้จากการดึงวิดีโอ ซึ่งจะแสดงออกมาเป็นภาพนิ่ง

5.1.2 การแสดงภาพที่อยู่ในดิสก์และการเก็บภาพ เมื่อต้องแสดงภาพที่อยู่ในดิสก์ของออกสู่จอมอนิเตอร์ ชั้นแรกต้องทำการส่งข้อมูล 03H ออกสู่พอร์ตเอาต์พุต 310H เป็นการควบคุมให้การ์ดทำการติดต่อกับคอมพิวเตอร์ โดยทำการต่อแอดเดรสของหน่วยความจำเข้ากับแอดเดรสของคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ทำการเขียนข้อมูลภาพที่จะแสดงสู่หน่วยความจำบนการ์ด เมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็ทำการส่งข้อมูล 02H ออกมาที่พอร์ตเอาต์พุต เป็นการสั่งให้การ์ดแสดงภาพจากหน่วยความจำ

ในกรณีของการอ่านข้อมูลภาพจากการ์ดเข้าไปเก็บไว้ในดิสก์จะทำได้ในลักษณะเดียวกับการแสดงภาพในดิสก์ คือ ต้องทำการส่งข้อมูล 03H ออกไปที่เอาต์พุตพอร์ต จากนั้นก็ทำการอ่านข้อมูลจากหน่วยความจำบนการ์ดเก็บลงสู่ดิสก์

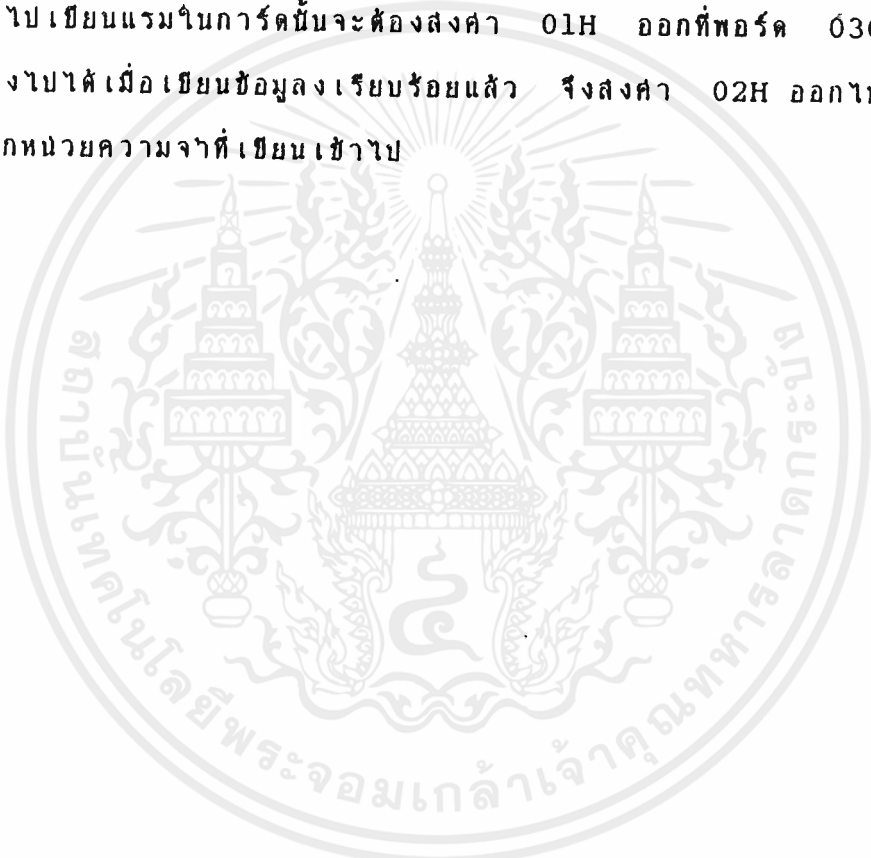
5.2 ส่วน SOFTWARE

ในส่วนของซอฟต์แวร์ ประกอบไปด้วยโปรแกรมที่เขียนขึ้นมาเพื่อทดลองใช้งานกับการ์ดนี้ มีทั้งหมด 2 โปรแกรม ซึ่งทั้งหมดเขียนด้วยภาษาซี (เทอร์โบซี) ดังมีรายละเอียดดังนี้

5.1.1 โปรแกรมถ่ายภาพ เป็นโปรแกรมที่ใช้เก็บภาพจากกล้อง VDO ลงใน File ารได้จะมีภาพที่รวบจับได้ว่ามีการกดคีย์บอร์ดหรือไม่ ถ้ามีก็จะทำการส่งค่า 02H ออกไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปที่พอร์ค 0300H เพื่อแสดงภาพที่ทำการ Digitize ได้ ซึ่งเป็นภาพนิ่งจากแรม และเมื่อต้องการเก็บภาพที่จะส่งค่า 03H ออกมาที่พอร์ค 0300H แล้วจึงทำการอ่านข้อมูลจากแรมลงเก็บในFile

5.1.2 โปรแกรมแสดงผลภาพ เป็นโปรแกรมแสดงผลภาพจาก File ที่ได้จากการ Digitize ซึ่งเป็น File ขนาด 256*256 จุดต่อภาพ หรือ 64 กิโลไบต์ ในการที่จะเข้าไปเขียนแรมในการ์ดนั้นจะต้องส่งค่า 01H ออกที่พอร์ค 0300H แล้วจึงจะเขียนข้อมูลลงไปได้เมื่อเขียนข้อมูลลงเรียบร้อยแล้ว จึงส่งค่า 02H ออกไปที่ 0300H เพื่อแสดงภาพจากหน่วยความจำที่เขียนเข้าไป



ตารางแสดงการควบคุมการทำงานของการ์ด

Select	P_control	ค่า	การทำงาน
0	0	00H	ทำการดึงไอดีซ์ข้อมูล
1	0	02H	แสดงข้อมูลในหน่วยความจำ
0	1	01H	Not Used
1	1	03H	ทำการติดต่อกับคอมพิวเตอร์

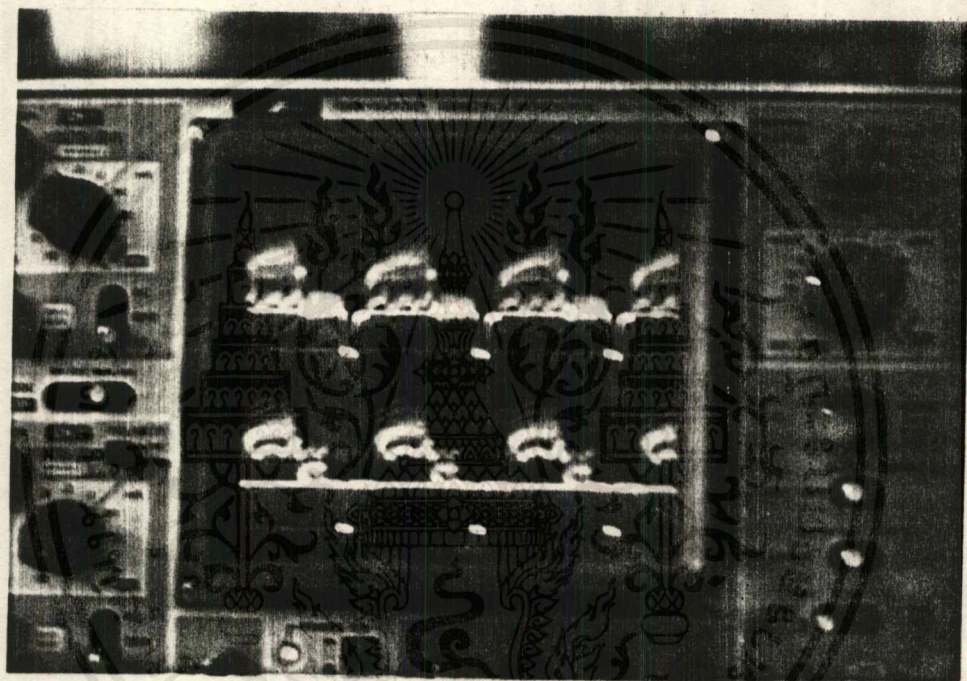
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

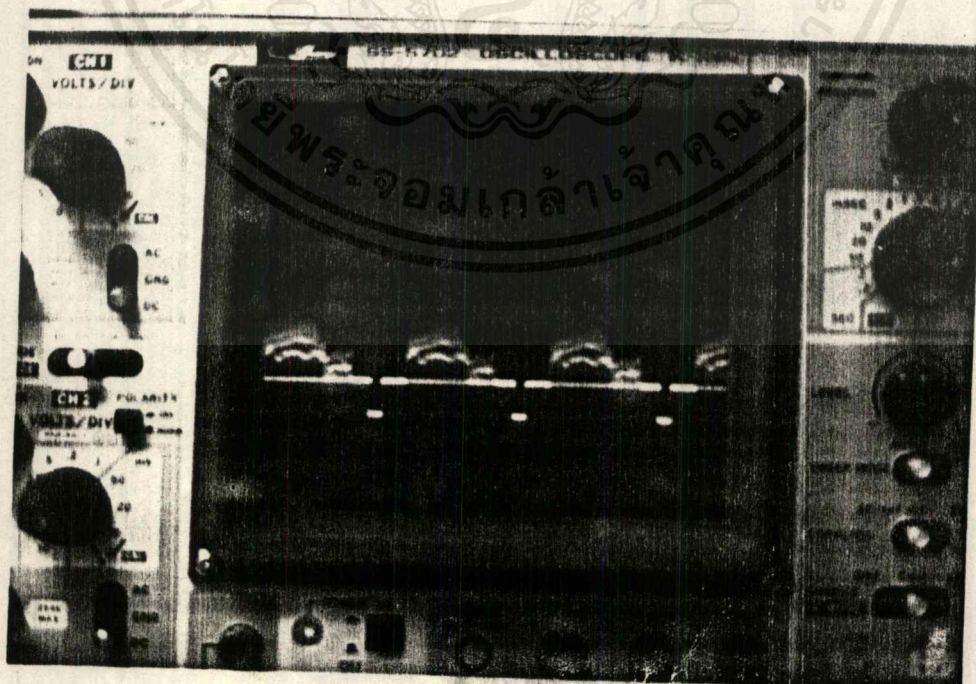
6.1 สรุปผลการทดลอง

ผลการทดลองจะแสดงได้ในภาพ



รูปที่ 1 แสดงการเปรียบเทียบ ระหว่างสัญญาณภาพจากกล้องเทียบกับสัญญาณภาพที่ผ่านขบวนการดิจิตอล

ภาพแสดงผลการทดลองการนำข้อมูลที่ได้จากการดิจิทัล ออกแสดงสู่จอมอนิเตอร์ และสัญญาณภาพจากจอออสซิลโลสโคป



เอกสารนี้เป็นเอกสารที่สงวนไว้ส่วนหนึ่งการใช้งานเพื่อการศึกษาเท่านั้นไปมอบภาคให้นำไปใช้ประโยชน์ด้านการค้า
รูป 6.2 แสดงภาพการดิจิทัล และ รูปสัญญาณ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

ในที่นี้จะกล่าวถึงปัญหาสำคัญ ที่พบในการทดลอง

จากการทดลองพบปัญหาที่เกิดจากสัญญาณรบกวน (noise) โดยมากเกิดจากการ wire สายบนบอร์ด และสัญญาณรบกวนที่เกิดจากการ ทำงานของอุปกรณ์จำพวกคิวดอลซึ่งสามารถทำให้ลดลงได้ด้วยการต่อ Capacitor ค่า 0.1 ไมโครครอมระหว่างขา VCC กับ GND ของไอซี ซึ่งสัญญาณรบกวนต่าง ๆ ที่กล่าวมาแล้วนั้นจะมีผลต่อการคิวดอลนำข้อมูลไปไว้ในหน่วยความจำทำให้ได้ภาพที่แสดงออกทางจอจอเตอร์ไม่คมชัดเท่าที่ควร

ปัญหาที่เกิดจากสัญญาณซิงค์ที่สร้างมีการคาดเคลื่อนซึ่ง จะทำให้ภาพที่แสดงที่จอจอเตอร์สั่นในบางครั้งแต่สามารถแก้ไขได้โดยการปรับ VR ในวงจรถ่ายสัญญาณซิงค์

6.2.1 ข้อเสนอแนะ จากการทดลองทำการคิวดอลนี้จะพบว่าในการคิวดอลจะใช้อุปกรณ์ต่อรวมกันเป็นจำนวนมากทำให้เกิดการ เบี่ยงเนื้อที่และค่าใช้จ่ายที่สูงในการทำปรินท์ และยังส่งผลให้เกิดปัญหาพวกสัญญาณรบกวน, การทำงานที่ไม่แน่นอน ผู้ออกแบบสามารถนำไอซีสำเร็จรูปมาทำงานแทนวงจรเหล่านี้ หรือในพวกวงจร เกทพื้นฐานอาจจะยุบได้โดยการใช้พวก PAL ได้ ซึ่งจะทำให้ง่ายต่อการทำปรินท์ เซอร์คิตบอร์ด และสัญญาณรบกวนต่าง ๆ ที่เป็นปัญหาใหญ่ก็จะลดลงตามไปด้วย

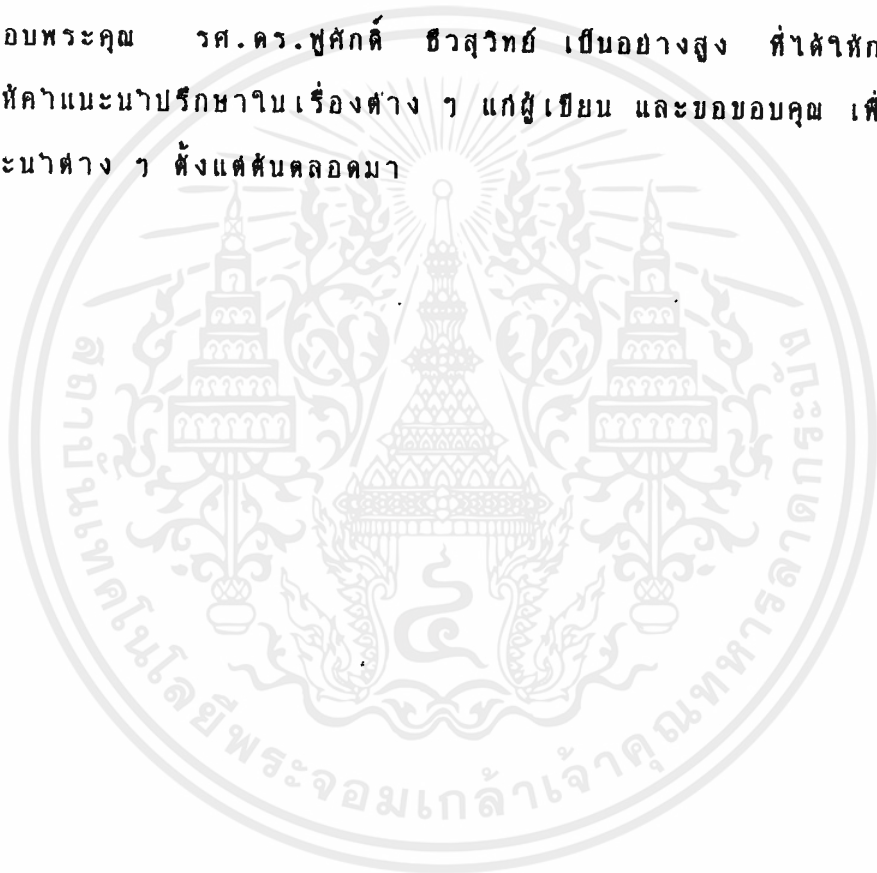
ส่วนแนวทางการพัฒนาต่อไปนั้นผู้ออกแบบจะทำการ์ด ให้มีความละเอียดสูงขึ้นเป็นขนาด 512x256 จุดต่อภาพ ระดับความสว่าง 256 ระดับ

เอกสารอ้างอิง

- [1] วารสาร Byte ฉบับประจำเดือน พฤษภาคม - สิงหาคม ปี 1987
- [2] กิกร ศุก , การอินเตอร์เฟส IBM PC , คลิกส์เซ็นเตอร์ การพิมพ์
- [3] ดร. ธวัช เมฆสุวรรณค์ , เทคนิคการซ่อมเครื่องรับโทรศัพท์ , สมาคมส่งเสริมความรู้ระหว่างประเทศ , พ.ศ. 2531
- [4] TTL Databook , Texas Instrument incorporated
- [5] Technical Reference , IBM
- [6] "Turbo C", Broland International, Inc. <4585 Scotts Valley Drive Scotts Valley , California 95066

กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.ดร.ชูศักดิ์ ชิวสุวathy เป็นอย่างสูง ที่ได้ให้การประสิทธิ์ประสาทวิชา ให้คำแนะนำปรึกษาในเรื่องต่าง ๆ แก่ผู้เขียน และขอขอบคุณ เพื่อน ๆ ที่ได้ช่วย และให้คำแนะนำต่าง ๆ ตั้งแต่ต้นตลอดมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* DIGITIZE 256 x 256 */
```

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
#include <dir.h>
```

```
#include <conio.h>
```

```
#include <string.h>
```

```
#include <io.h>
```

```
#define port 0x300
```

```
unsigned char fname[40];
```

```
char f[MAXFILE];
```

```
char d[MAXDRIVE];
```

```
char dir[MAXDIR];
```

```
char ex[MAXEXT];
```

```
main()
```

```
{
```

```
int spf = 12;
```

```
int lop = 0;
```

```
char ch;
```

```
do{
```

```
clrscr();
```

```
digitize();
```

```
do{
```

```
getname(&spf) ;
```

```
if(spf==1234){
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยศูนย์บริการคอมพิวเตอร์และสารสนเทศ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    getch();

    lop = 1;
}

if(spf==12){
    printf("SAVING  %s..",fname);
    if(Wrfile(fname))
printf("\nWrite error !");
    lop = 0;
}

}while(lop);
outportb(port,0x02);
puts("\nEnd program");
printf("\nDIGITIZE AGAIN (Y/N) : ");
}while(toupper(getche())=='Y');
outportb(port,0x00);
}

getname(int *sp_fl)
{
    struct ffblk ffblk;
    struct dfree dfreep;
    long sp;
    int i,flag,drive,pres;
    int fl = 0;
    int wr = 0;
    char tff[60];

    clrscr();

    *sp_fl = 12;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    printf("Enter file name..");
    scanf("%s",fname);
    flag = fnsplit(fname,d,dir,f,ex);
    if(flag&DRIVE){
        if(d[0]>='a' && d[0]<='z') drive = d[0]-'a'+1;
        else if(d[0]>='A' && d[0]<='Z') drive = d[0]-'A'+1;
        else{
            puts("Illegal drive name");
            fl = NULL;
        }
    }
    else{
        drive = 0;
        strcpy(d,"X:");
        d[0]='A'+getdisk();
        strcpy(tf,d);
        strcpy(tf+2,"\\");
        strcat(tf+3,fname,strlen(fname));
        strncpy(fname,tf,strlen(fname)+3);
    }

    pres = findfirst(fname,&ffblk,0);
    if(!pres){
        printf("%s ALREADY EXIST.. OVERWRITE (Y/N) : ",fname);
        if(toupper(getche())=='Y') wr = 1;
    }
    else wr=0;
    printf("\n");
}

}while(!pres&&!wr);

getdfree(drive,&dfree);

```

```

sp=(long)dfreep.df_avail*(long)dfreep.df_bsec*
    (long)dfreep.df_sclus;
sp+=(long)(wr*ffblk.ff_fsize);
printf("\nDRIVE %s %10ld bytes available.",d,sp);
if(sp<0x10000){
    printf("\nWARNING Insufficient space for IMAGE file.");
    *sp_fl = 1234;
}
return(fl);
}

```

```
digitize()
```

```

{
    int dig,lp;

    printf("\nDIGITIZE (Y/N) : ");
    if(toupper(getche())=='Y') dig=1;
        else dig = 0;
    if(dig){
        do{
            clrscr();
            outportb(port,0x00);
            printf("\nPress any key to get PICTURE...");
            getch();
            outportb(port,0x02);
            printf("\nSave Picture (Y/N) : ");
            if (toupper(getche())=='Y') lp=0;
        else lp=1;
            if(lp==1){

```

printf("\nDIGITIZE AGAIN (Y/N) : ");
 เปรียบเทียบการศึกษานี้กับที่การศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(toupper(getche())!= 'Y') {
    outportb(port,0x00);
    exit();
}
}

}while(lp);
}

else {
    outportb(port,0x00);
    exit();
}

return(0);
}

```

```

Wrfileread(char *file)

```

```

{
    struct SREGS sregs;
    union REGS regs;
    unsigned int byte;
    int ret=0;
    int handle,i;

    regs.h.ah = 0x3c;
    regs.x.cx = 0;
    sregs.ds = _DS;
    regs.x.dx = (unsigned)file;

    handle = intdosx(&regs,&regs,&sregs);
    if(regs.x.cflag){

```

เอกสารนี้เป็นลิขสิทธิ์สงวนไว้สำหรับใช้ภายในเท่านั้น การเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(NULL);
}

outportb(port,0x03);
for(i=0;i<2;i++){

    regs.h.ah = 0x40;
    regs.x.bx = handle;
    regs.x.cx = 0x8000;
    regs.x.dx = i*0x8000;
    sregs.ds = 0xd000;

    byte = intdosx(&regs,&regs,&sregs);

    if(regs.x.cflag){
        ret=345;
        break;
    }
    else if(byte!=0x8000){
ret=345;
break;
}
    else;
}

    regs.h.ah = 0x3e;
    regs.x.bx = handle;
    intdosx(&regs,&regs,&sregs);
    if(regs.x.cflag) ret=345;
    return(ret) ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* DISPLAY IMAGE 256 x 256 */
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <fcntl.h>
#include <float.h>
#include <dos.h>
#include <io.h>

#define port 0x300

int flag = 1310;

extern int _fmode;
typedef struct data{
    unsigned char far *pxy[256];
}dataxy;
typedef dataxy *dpointer;
typedef unsigned int table[256];

main()
{
    _fmode = O_BINARY;
    do{
        clrscr();
        begin();
        printf("\n\tDISPLAY AGAIN (Y/N) : ");
        )while(toupper(getche())=='Y');
        outport(port,0x00);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin()
{
    unsigned char lut[256];
    char fname[60];
    dataxy dst;
    int c;
    int read = 1;
    int i;
    char *errmsg[] = {"\n Not enough memory.",
        "\n Read error !",
        "\n Write error ! ",
        "\n Do not REPEAT any disk during calculation !"
    };
    if((c=malloc(&dst))!=flag){
        puts(errmsg[0]);
        free(&dst,c);
        return(1);
    }
    b1:getname(fname,read);
    printf("\nLoading...");
    if(iloop(&dst,fname)!=flag){
        puts(errmsg[1]);
        getch();
        goto b1;
    }
    printf("\nDisplay SOURCE image...");
    disp(&dst);
    free(&dst,256);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
;
s()
{
    sound(800);
    delay(200);
    nosound();
}

```

```

imalloc(dpointer dptr)
{
    int i;
    for(i=0;i<256;i++){
        if((dptr->pxyfil = (unsigned char *)farmalloc(256))==NULL)
return(i);
    }
    return(flag);
}
;

```

```

ifree(dpointer dptr,int num)
{
    int i;
    for(i=0;i<num;i++)
        farfree(dptr->pxyfil);
}
;

```

```

iload(dpointer dp,char *fname)
{
    int handle,i,mode_rd;

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mode_rd = O_RDONLY|O_BINARY;

if((handle=_open(fname,mode_rd))==EOF){
    fprintf(stderr,"Can't open file %s",fname);
    return(NULL);
}

for(i=0;i<256;i++){
    if(_read(handle,dp->pxy[i],256)!=256){
        ok = EOF;
        break;
    }
}

close(handle);
return(ok);
}

disp(dpointer dp)
{
    unsigned int offset = 0;
    int i,j;
    unsigned char huge *d;
    outportb(port,0x03);
    for(j=0;j<256;j++){
        d=dp->pxy[j];
        for(i=0;i<256;i++) pokeb(0xd000,offset+(j*256)+i,*(d+i));
    }
    outportb(port,0x02);
}

getname(char *fname,int mode)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int lp = 0;
do{
    clrscr();
    printf("Enter file name <drive:file.ext>");
    scanf("%s",fname);
    lp = !file_exist(fname,mode);
    if(mode){
        if(lp){
            printf("Permission denied OR file not found.\n\n <Press any key>");
            bioskëy(0);
        }
    }
else{
    if(!lp){
        printf("File already exist overwrite Y/N ");
        if(toupper(getche()) == 'Y') lp=0;
        else lp=1;
    }
    else lp=0;
}
}while(lp);
}

```

```

file_exist (char *fname,int mode)
{
    int acs;
    acs = (!mode)*06;
    return(access(fname,acs)==0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้