สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจัดการสัญญาณภาพโดยใช้ FPGA

Video Management by Using FPGA

จัดทำโดย

นายกลิ้ง กาญจนสุวรรณ 41014016
นายกิตติ เจริญวนันท์ 41014024
นายกิตติศักดิ์ ชัยเจริญ 41014032
นายจิรวัฒน์ ห่อหุ้มสุวรรณา 41014075

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

เลขหมู่.....
เลขทะเบียน 46204
วัน, เดือน, ปี 21 ธ.ค. 2546

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประจำภาคเรียนที่ 2 ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2544

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจัดการสัญญาณภาพโดยใช้ FPGA

ผู้จัดทำ

1. นายกลิ้ง กาญจนสุวรรณ
2. นายกิตติ เจริญวนันท์
3. นายกิตติศักดิ์ ชัยเจริญ
4. นายจิรวัฒน์ ห่อหุ้มสุวรรณา

.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดการสัญญาณภาพโดยใช้ FPGA

นายกลิ้ง กาญจนสุวรรณ	41014016
นายกิตติ เจริญวนันท์	41014024
นายกิตติศักดิ์ ชัยเจริญ	41014032
นายจิรวัฒน์ ห่อหุ้มสุวรรณา	41014075

ปริญญานิพนธ์ฉบับนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดการสัญญาณภาพโดยใช้ FPGA

นายกลิ้ง กาญจนสุวรรณ
นายกิตติ เจริญวนันท์
นายกิตติศักดิ์ ชัยเจริญ
นายจิรวัฒน์ ห่อหุ้มสุวรรณา
รศ.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

การใช้งานชิพ FPGA ซึ่งเป็นไอซีดิจิตอล ที่เราสามารถโปรแกรมลงไปให้สามารถใช้งานเป็นวงจรใดๆ ก็ได้ที่ต้องการ โดยใช้ซอฟแวร์ MAX+PLUS II ในการออกแบบและจำลองการทำงาน รวมถึงการนำส่วนที่ออกแบบโปรแกรมลงบนตัวชิพ FPGA ซึ่งการใช้งาน FPGA สามารถประยุกต์ใช้ประโยชน์ได้มากมาย สำหรับโครงงานนี้ ได้ทำการประยุกต์ใช้งาน FPGA เพื่อจัดการกับสัญญาณภาพ โดยจะนำสัญญาณภาพ 4 สัญญาณจากแหล่งจ่าย 4 แหล่งมาแสดงผลบนหน้าจอเดียวกัน ซึ่งในการควบคุมสัญญาณภาพจะใช้การเขียนภาษา VHDL ในการควบคุมระบบฮาร์ดแวร์ทั้งหมดร่วมกับการอ่านและเขียนผ่าน RAM ที่ความถี่สูง เพื่อให้ภาพที่ออกมา smooth ให้มากที่สุด ในส่วนของการควบคุมสัญญาณภาพนี้จะแยกสีของภาพทั้ง 4 สัญญาณเป็นสัญญาณสี R, G, B แล้วนำมารวมกัน ซึ่งส่วนต่างๆ ของวงจรประกอบด้วย ส่วนของ A/D, Latch input, RAM, Latch output และ D/A โดยการทำงานทุกส่วนจะสัมพันธ์กันและถูกควบคุมจาก FPGA ทั้งสิ้น

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Video Management by Using FPGA

Mr. Gling Kanchanasuwan.
Mr. Kitti Jaruenwanan.
Mr. Kittisak Chaicharoen.
Mr. Jirawat Houhumsuwanna.
Assoc. Prof. Dr. Manas Sangworasilp(Adviser)
Semester 2,2001

Abstract

Using the digital programmable IC " FPGA " on IC which we can program any logic program into it by using a software called MAX+PLUS II to design and simulate the program that will be burned to the FPGA chip. In this project we use the FPGA for managing video signals, we will use 4 different video source to show up on the same screen at the same time. By controlling the system we use VHDL to control all of the hardware, to read-write the high frequency RAM for getting the smoothest result of the picture on the screen. The part that is used to control the video signal will spilt the screen up to 4 part by mixing the R, G, B signal. The circuits consist of A/D, Latches, RAM, D/A and every part will be controlled by the FPGA.

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญภาพ	V
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของการดำเนินงาน	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
บทที่ 2 เรื่องราวเบื้องต้นของโทรทัศน์	2
2.1 คุณสมบัติของแสงและสี	2
2.2 ความสำคัญสามประการของแสงสีที่มองเห็น	3
2.3 การผสมและการแยกแสงสี	3
2.4 องค์ประกอบของภาพ	4
บทที่ 3 หลักการของการส่งและการรับโทรทัศน์สี	6
3.1 การส่งและการรับโทรทัศน์สีทำได้อย่างไร	6
3.2 การสแกนและเรื่องที่เกี่ยวข้อง	10
3.3 สัญญาณภาพรวม	13
3.4 การผสมสัญญาณโทรทัศน์	17
3.5 ระบบของการผสมสัญญาณ	17
3.6 สัญญาณโทรทัศน์สีประกอบด้วยอะไรบ้าง	20
บทที่ 4 การเขียนภาษา VHDL	24
4.1 การออกแบบในลักษณะโครงสร้างและการบรรยายพฤติกรรม	28
4.2 ชุดคำสั่งลำดับ	30
4.3 ชุดคำสั่งแบบแข่งขนาน	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 หลักการออกแบบ 36
บทที่ 6 การทดลอง 46
6.1 โปรแกรมภาษา VHDL สำหรับควบคุมการทำงานต่างๆ ของวงจร 46
6.2 Module ของโปรแกรมควบคุมการทำงาน 84
6.4 ผลการทดลอง 88
บทที่ 7 วิเคราะห์และสรุปผลการทดลอง 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 3.1 ส่วนประกอบของภาพในรูปซึ่งมีพื้นที่เท่ากัน	7
รูปที่ 3.2 การส่งและการรับภาพทางโทรทัศน์	7
รูปที่ 3.3 การส่งโทรทัศน์สีระบบแสงสีแดง-แสงสีเขียว-แสงสีน้ำเงิน โดยการส่งสัญญาณภาพสีด้วยวิธีการสลับแสงสี	8
รูปที่ 3.4 ระบบการส่งและการรับโทรทัศน์สี	9
รูปที่ 3.5 หลักการส่งและหลักการรับภาพทางโทรทัศน์	10
รูปที่ 3.6 ทฤษฎีการหักเหทางไฟฟ้าสถิตและทางแม่เหล็กไฟฟ้า	10
รูปที่ 3.7 การสแกนจากซ้ายไปขวาและจากบนลงล่าง	12
รูปที่ 3.8 การสแกนไขว้กัน	12
รูปที่ 3.9 รูปร่างของสัญญาณซิงค์ที่ใช้ในการส่งโทรทัศน์	13
รูปที่ 3.10 แสดงระดับสัญญาณขาว-ดำ โดยเกรย์สเกล	14
รูปที่ 3.11 แสดงลักษณะของสัญญาณภาพรวมเบื้องต้น	16
รูปที่ 3.12 สัญญาณภาพรวมที่เครื่องส่งส่งมาซึ่งแสดงให้เห็นระดับต่างๆ ของสัญญาณที่เราคิดเป็นค่าเปอร์เซนต์ ออมาเพื่ออรรถาธิบายถึงความเป็นภาพหน้าจอ	17
รูปที่ 3.13 วิธีการ เอเอ็มดับเบิลไซด์แบนด์	18
รูปที่ 3.14 วิธีการของเวสติเจียลไซด์แบนด์	19
รูปที่ 3.15 แสดงแบนด์วิดธ์เบื้องต้นของสัญญาณภาพ	19
รูปที่ 3.16 สัญญาณส่องสว่าง	22
รูปที่ 3.17 รูปร่างของสัญญาณที่ได้จากการมอดูเลทแบบ เอ.เอ็ม. โดยทิ้งสัญญาณคลื่นพาห์	23
รูปที่ 5.1 การนำสัญญาณมาแสดงผลบนหน้าจอเดียวกัน	36
รูปที่ 5.2 การทำงานของโครงงาน	37
รูปที่ 5.3 การควบคุมการ Sampling สัญญาณ input	38
รูปที่ 5.4 Horizontal Synchronize counter	38
รูปที่ 5.5 Sampling Horizontal wave	39
รูปที่ 5.6 การควบคุมการ Read จาก Latch เข้าสู่ RAM	41
รูปที่ 5.7 การควบคุมการ Read ออกจาก RAM เข้าสู่ Latch	42
รูปที่ 5.8 Control clock A/D & Latch	43
รูปที่ 5.9 การแบ่งแยกกลุ่ม Latch และ Latch สำรอง	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

การออกแบบวงจรในปัจจุบันมีความยุ่งยากซับซ้อนมากขึ้น และวงจรถูกนำมาใช้อย่างแพร่หลายมากขึ้น ไอซีดิจิตอลที่สามารถโปรแกรมลงไปได้กำลังได้รับความนิยมมากขึ้นเพราะสะดวกในการออกแบบ นอกจากนี้ยังมีซอฟแวร์ MAX+PLUS II ช่วยจำลองการทำงานอีกด้วย การออกแบบเพื่อควบคุมสัญญาณภาพนี้จะใช้ภาษา VHDL เป็นตัวควบคุมการสร้างวงจรใน FPGA ซึ่งในโครงงานนี้เราใช้ Flex10K10 และ FPGA นี้จะไปควบคุมการทำงานของส่วนต่างๆ เช่น ควบคุมชุด A/D, ชุด Latch ทั้ง Latch i/p และ Latch o/p, RAM, และชุด D/A เป็นต้น โดยวงจรทุกๆ ส่วนจะควบคุมให้ภาพจากแหล่งจ่าย 4 แหล่ง มาแสดงผลบนหน้าจอเดียวกัน

1.1 วัตถุประสงค์ของโครงงาน

1. สามารถเรียนรู้การทำงาน ตลอดจนการใช้งาน FPGA
2. สามารถเรียนรู้การใช้เครื่องมือที่ใช้ช่วยในการออกแบบวงจรดิจิตอล คือ โปรแกรม MAX+PLUS II เพื่อให้เกิดประสิทธิภาพสูงสุด
3. สามารถควบคุมและออกแบบระบบโดยใช้ภาษา VHDL เพื่อใช้งานกับ FPGA ได้
4. สามารถเข้าใจหลักการทำงานของการ scan ภาพแบบ Interlaced (แบบสลับเส้น) และการ modulate ของสัญญาณภาพทั้งโทรทัศน์และจอมอนิเตอร์
5. นำ FPGA ไปประยุกต์ใช้กับการประมวณผลสัญญาณภาพของโทรทัศน์และจอมอนิเตอร์
6. สามารถออกแบบวงจรดิจิตอลพื้นฐานได้เพื่อนำไปสู่การพัฒนาวงจรขั้นสูงต่อไป

1.2 ขอบเขตของการดำเนินงาน

1. ศึกษาลักษณะการทำการ scan ภาพ ของทั้งมอนิเตอร์และโทรทัศน์
2. ศึกษาพื้นฐานระบบดิจิตอลและการใช้ MAX+PLUS II ในการออกแบบและจำลองการทำงาน เพื่อนำส่วนที่ออกแบบแล้วมาโปรแกรมลงบน FPGA
3. ใช้โปรแกรม VHDL ในการออกแบบการใช้งานกับ FPGA
4. ใช้ FPGA ในการประมวลผลการ scan ของสัญญาณภาพของทั้งมอนิเตอร์และโทรทัศน์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงทฤษฎีต่างๆ ในการออกแบบวงจรดิจิตอล
2. เข้าใจถึงทฤษฎีที่ใช้ในการ scan ภาพ ของทั้งมอนิเตอร์และโทรทัศน์
3. เข้าใจถึงการใช้โปรแกรม MAX+PLUS II ในการออกแบบและจำลองการทำงาน
4. สามารถใช้ VHDL ในการออกแบบ FPGA ได้
5. สามารถใช้ FPGA ในการประมวลผลการ scan ของสัญญาณภาพของทั้งมอนิเตอร์และโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 เรื่องราวเบื้องต้นของโทรทัศน์

โทรทัศน์คือการส่งกระจายเสียงพร้อมทั้งภาพที่เคลื่อนไหวออกไปในรูปของสัญญาณโทรทัศน์ และการรับสัญญาณนี้มาเปลี่ยนให้เป็นเสียงและภาพที่ต้องการ ภาพที่เห็นเคลื่อนไหวติดต่อกันบนจอเครื่องรับโทรทัศน์นั้น ความจริงก็คือผลของการส่งและการรับภาพนิ่งที่มีความแตกต่างกันเพียงเล็กน้อยหลายๆ ภาพต่อหน่วยเวลานั่นเองเพราะสายตาธรรมดาของคนเรานั้น มีลักษณะพิเศษอย่างหนึ่งคือในขณะที่จ้องดูภาพใดภาพหนึ่งแล้วภาพนั้นจางหายไปโดยกะทันหัน ก็จะยังคงรู้สึกเห็นติดตาอยู่อีกในชั่วระยะเวลาอันสั้น ลักษณะพิเศษนี้มีชื่อเรียกกันว่า persistence of vision ด้วยเหตุนี้หากจะมีการฉายภาพนิ่ง ซึ่งแต่ละภาพมีความแตกต่างกันเล็กน้อยติดต่อกันไปเป็นจำนวนหลายๆ ภาพต่อหนึ่งวินาที สายตาของคนเราก็จะเห็นเป็นภาพเคลื่อนไหวต่อเนื่องกันไปโดยไม่หยุดชะงัก สายตาของคนเราจะจับอาการกระพริบของภาพไม่ได้เลย หากจำนวนภาพนิ่งต่อหนึ่งวินาทีนั้นมีเกิน 16 ภาพขึ้นไป การส่งโทรทัศน์ในระบบยุโรป อันเป็นวิธีการของสถานีโทรทัศน์สี ไทยทีวีสีช่อง 3 สถานีโทรทัศน์กองทัพบกช่อง 5 และช่อง 7 กับสถานีโทรทัศน์สี ไทยทีวีสีช่อง 9 ในกรุงเทพมหานคร จะมีจำนวน 25 ภาพต่อวินาที ภาพหนึ่งๆ ก็จะมีการส่ง 2 ครั้ง

การส่งและการรับโทรทัศน์สี มีลักษณะคล้ายคลึงกับการส่งและการรับโทรทัศน์ขาวดำ กล่าวคือ เป็นการส่งกระจายเสียงพร้อมทั้งภาพสีที่เคลื่อนไหวออกไปในรูปของสัญญาณโทรทัศน์สี และการรับสัญญาณนี้มาเปลี่ยนให้เป็นเสียงและภาพที่ต้องการ เนื่องจากระบบการส่งและการรับโทรทัศน์สีได้รับการออกแบบให้สอดคล้องกัน (compatible) กับระบบการส่งและการรับโทรทัศน์ขาวดำ ดังนั้นเครื่องรับโทรทัศน์ขาวดำในเขตบริการจึงสามารถรับสัญญาณจากสถานีโทรทัศน์สีได้เป็นภาพขาวดำ หรืออาจกล่าวได้ว่าเครื่องส่งโทรทัศน์สีก็สามารถส่งสัญญาณโทรทัศน์สีให้เครื่องรับโทรทัศน์สี และส่งสัญญาณโทรทัศน์ขาวดำให้เครื่องรับโทรทัศน์ขาวดำในเขตบริการด้วย

2.1 คุณสมบัติของแสงและสี

แสงโดยทั่วไปมีอยู่ 2 ประเภท คือแสงที่สายตามนุษย์เรามองเห็น (visible rays) กับแสงที่สายตาคนธรรมดาเรามองไม่เห็น ซึ่งได้แก่แสงจำพวก แกมม่า-เรย์ (gramma-rays), เอ็กซ์-เรย์ (X-rays) อุลตราไวโอเล็ต เรย์ (ultraviolet rays) และ อินฟราเรด เรย์ (infrared rays) แสงเหล่านี้ ต่างก็เป็นส่วนหนึ่งของพลังงานแม่เหล็กไฟฟ้า แต่มีความถี่หรือความยาวคลื่นแตกต่างกันออกไป ส่วนของคลื่นแม่เหล็กไฟฟ้าที่น่าสนใจในเรื่องราวของโทรทัศน์สีนี้ ก็คือส่วนของคลื่นแม่เหล็กไฟฟ้าที่เป็นแสงซึ่งมองเป็นด้วยสายตาของคนธรรมดา ซึ่งมีความยาวคลื่นประมาณ 380 นาโนเมตร จนถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมาณ 780 นาโนเมตร แสงที่มองเห็นนี้ จะทำให้สายตาของคนเราได้รับความรู้สึก 2 ประการ ประการแรกคือความรู้สึกว่ามีแสงสว่างมากหรือน้อย (sensation of brightness) ความรู้สึกอีกประการหนึ่ง ก็คือทำให้รู้จักว่าเป็นแสงสีอะไร ความรู้สึกนี้เรียกว่า sensation of color

2.2 ความสำคัญสามประการของแสงสีที่มองเห็น

แสงสีที่มองเห็นจะทำให้เกิดความรู้สึกที่สำคัญอยู่ 3 ประการ คือ เกิดความรู้สึกในเรื่องแสงสี (hue) เกิดความรู้สึกในเรื่องการส่องสว่าง (brightness) และเกิดความรู้สึกในเรื่องแสงสีอิ่มตัว (saturation) โดยความรู้สึกในเรื่องแสงสี (hue) จะทำให้สายตาสามารถแยกแยะออกได้ว่าแสงสีที่มองเห็นเป็นแสงสีแดง หรือแสงสีเขียว หรือแสงสีน้ำเงิน เป็นต้น ส่วนความรู้สึกในเรื่องการส่องสว่าง (brightness or value) จะทำให้สามารถมองเห็นเป็นแสงสีแดงเข้มสว่างหรือมืด เป็นต้น สำหรับความรู้สึกในเรื่องแสงสีอิ่มตัว (saturation or chroma) จะทำให้สามารถรู้ความบริสุทธิ์ของแสงสีได้ว่าเป็นแสงสีน้ำเงินชัดเจน หรือแสงสีน้ำเงินจาง ดังนี้เป็นต้น

2.3 การผสมและการแยกแสงสี

การผสมแสงสีมีลักษณะพิเศษต่างไปจากการผสมสีที่ใช้ในการวาดเขียนภาพสี การผสมสีโดยทั่วไป ซึ่งใช้ในการวาดเขียนหรือการพิมพ์ภาพสี จะทำให้ได้สีผสมที่เข้มข้นกว่า ซึ่งเป็นลักษณะวิธีของ Subtractive mixture ส่วนการผสมแสงสีที่ใช้ในโทรทัศน์สีนั้น จะทำให้ได้แสงสีผสมที่เจือจางกว่าเดิม อันเป็นลักษณะวิธีของ additive mixture การผสมแสงสีที่ใช้ในโทรทัศน์สีนั้นเป็นการผสมแม่สี (primary color) เพื่อทำให้เกิดแสงสีต่างๆ ขึ้น สีที่เรียกว่าแม่สีนี้จะต้องเป็นอิสระและจะต้องไม่สามารถนำสีอื่นๆ ที่มีอยู่มาผสมทำให้เกิดแม่สีนั้นๆ ได้เลย แม่สีในเรื่องของโทรทัศน์สีนี้มีอยู่ 3 สีด้วยกัน คือ แสงสีแดง (red) แสงสีเขียว (green) และแสงสีน้ำเงิน (blue)

แสงสีต่างๆ ที่สายตาสามารถมองเห็นนั้น เกิดจากการผสมสีของแม่สีทั้งสามนี้ทั้งนั้น ซึ่งอาจเกิดจากการนำสีใดสีหนึ่งมาผสมกับอีกสีใดสีหนึ่ง หรืออาจเกิดจากการนำสีใดสีหนึ่งมาหักออกจากแสงสีใดสีหนึ่งก็ได้ ยกตัวอย่างเช่น

แสงสีเหลือง (yellow)	=	แสงสีแดง (red) + แสงสีเขียว (green)
แสงสีเขียวน้ำเงิน (cyan)	=	แสงสีเขียว (green) + แสงสีน้ำเงิน (blue)
แสงสีม่วง (magenta)	=	แสงสีน้ำเงิน (blue) + แสงสีแดง (red)
แสงสีแดง + แสงสีเขียว + แสงสีน้ำเงิน	=	แสงสีขาวหรือไม่มีสี (white)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงสีจากแม่สีทั้งสาม คือ แสงสีแดง แสงสีเขียว และแสงสีน้ำเงินนี้ ได้มีการทดลองหาความยาวคลื่นที่แน่นอนไว้แล้ว คณะกรรมการระหว่างประเทศในเรื่องการส่องสว่าง (International Committee on Illumination หรือ Commission International de I'Eclairage หรือเรียกชื่อสั้นๆ ว่า CIE) ได้กำหนดความยาวคลื่นของแสงจากแม่สีทั้งสามไว้ในตาราง

ตารางแสดงแม่สีทั้งสามในระบบแสงสีแดงแสงสีเขียวแสงสีน้ำเงิน (RGB)

รายการ	ความยาวคลื่น (นาโนเมตร)	หมายเหตุ
สีน้ำเงิน	435.8	เป็นส่วนหนึ่งของสเปคตรัมเชิงเส้นของปรอท
สีเขียว	546.1	เป็นส่วนหนึ่งของสเปคตรัมเชิงเส้นของปรอท
สีแดง	700	ประมาณค่าออกเป็นตัวเลขหยาบๆ

2.4 องค์ประกอบของภาพ

หากเราตัดภาพจากหนังสือพิมพ์สักภาพหนึ่ง แล้วขยายขึ้นด้วยกล้องหรือแว่นขยาย จะพบว่าภาพมีองค์ประกอบมาจากจุดสีขาวและจุดสีดำมากมาย มาเรียงกันประกอบขึ้นเป็นภาพ จุดเหล่านี้เองที่เรียกว่าองค์ประกอบของภาพ หรือพิคเจอร์ อีลีเมนต์ (Picture Element) หรือพิกเซล (Pixel)

ทำนองเดียวกัน ภาพที่ปรากฏทางจอโทรทัศน์ก็เอามาจากหลักการนี้ ภาพที่เกิดขึ้นบนจอโทรทัศน์ประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนเป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิท ส่วนที่จาง และส่วนที่สว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสง (Scan) ความแตกต่างกันบนเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองที่เราจัดว่าเป็นองค์ประกอบภาพ

ระบบสแกน 525 เส้นเราจะแบ่งส่วนทางแนวตั้งได้ 700 เส้น ดังนั้นจึงกล่าวได้ว่าหากจะหาจำนวนขององค์ประกอบภาพในระบบการสแกน 525 เส้นแล้ว จะได้เท่ากับ 525X700 เท่ากับ 367,500 พิกเซล ยิ่งภาพมีจำนวนเส้นมากเท่าไรรายละเอียดภาพยิ่งมากขึ้นเท่านั้น โทรทัศน์ที่มีจำนวนเส้นสแกนมากย่อยได้รายละเอียดหรือความชัดเจนของภาพมากกว่า แต่การออกแบบวงจรจะยากตามไปด้วย เนื่องจากแบนด์วิดธ์ของความถี่จะต้องกว้างขึ้นด้วย จากที่กล่าวมาว่าองค์ประกอบของภาพมีหลายพิกเซล แต่ในความเป็นจริงแล้วเรามองเห็นได้ไม่ครบทุกพิกเซล เนื่องจากมีบางส่วนหายไประหว่างการสแกน จากการทดลองพบว่าภาพที่พอดูได้จะมีองค์ประกอบไม่ต่ำกว่า 200,000 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบที่ใช้อยู่ในประเทศไทย ปัจจุบันเราใช้ระบบการสแกน 625 เส้น 25 ภาพต่อวินาที จึงทำให้แบนด์วิดธ์ต้องกว้างถึง 7 เมกะเฮิรตซ์ ในขณะที่ระบบ 525 เส้น กว้างเพียง 6 เมกะเฮิรตซ์เท่านั้น อย่างไรก็ตามองค์ประกอบภาพจะมีความละเอียดมากขึ้น โดยสามารถหาองค์ประกอบภาพได้จากค่าจำนวนสแกน 625 เส้นคูณกับจำนวนจุดหรือองค์ประกอบตามแนวตั้ง 851 เส้น ได้เท่ากับ 531,875 พิกเซล

ปัจจุบันส่วนที่เรียกว่าองค์ประกอบภาพได้ถูกนำเอาไปใช้งานอย่างเป็นจริงมากขึ้น ในโทรทัศน์หรือเครื่องเล่นวีดีโอคาสเซ็ทเร็คคอร์เดอร์ จะมีการนำเอาพิกเซลเหล่านี้เก็บไว้ในหน่วยความจำ เพราะพิกเซลเท่านั้นที่ระบบดิจิตอลจะจัดการข้อมูลได้ เราจะพบวิธีการนี้ในโทรทัศน์ระบบดิจิตอล, โทรทัศน์ระบบคอมพิวเตอร์, โทรทัศน์จอภาพแอลซีดี, ดิจิตอล วีซีอาร์, โทรทัศน์หรือวีซีอาร์ ระบบภาพซ้อนภาพ ฯลฯ

ในปัจจุบันสำหรับโทรทัศน์ธรรมดาเราจะพบว่ามีการเพิ่มเส้นภาพให้มากขึ้น และนั่นแน่นอนว่าจำนวนพิกเซลย่อมมากขึ้นด้วย อย่างโทรทัศน์จอใหญ่หรือโทรทัศน์ที่ต้องการรายละเอียดสูงอย่าง HDTV อาจจะต้องใช้เส้นสแกนภาพมากกว่า 625 เส้น เช่นที่นิยมใช้กันในปัจจุบันคือ 725 เส้น หรืออย่างเครื่องฉายวีดีโอโปรเจ็คเตอร์จะใช้เส้นภาพ 2,200 เส้นภาพ และหากเป็นจอใหญ่หลายร้อยนิ้วจะต้องเพิ่มรายละเอียดมากขึ้นอีก นั่นคือการเพิ่มพิกเซลอีลีเมนต์นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 หลักการของการส่งและการรับโทรทัศน์สี

โทรทัศน์สีกับโทรทัศน์ขาวดำ มีเรื่องราวและหลักการที่คล้ายคลึงกัน มาตรฐานของโทรทัศน์สีจะต้องเข้ากันได้กับมาตรฐานของโทรทัศน์ขาวดำในระบบเดียวกันด้วยเสมอ ประชาชนทั่วไปที่มีทั้งเครื่องรับโทรทัศน์ทั้งแบบโทรทัศน์ขาวดำและแบบโทรทัศน์สี จึงจะสามารถรับชมโทรทัศน์ได้ กล่าวคือ เมื่อสถานีโทรทัศน์ส่งสัญญาณโทรทัศน์สีออกอากาศ นอกจากเครื่องรับโทรทัศน์สีจะสามารถรับสัญญาณโทรทัศน์สีนี้ได้เป็นภาพสีบนจอหลอดภาพของเครื่องรับโทรทัศน์สีแล้ว ยังจะต้องทำให้เครื่องรับโทรทัศน์ขาวดำในเขตบริการ สามารถรับสัญญาณโทรทัศน์สีนี้ได้เป็นภาพขาวดำบนจอหลอดภาพของเครื่องรับโทรทัศน์ขาวดำด้วย ความเข้ากันได้หรือความสามารถของเครื่องรับโทรทัศน์ขาวดำในการรับสัญญารโทรทัศน์สีนี้ มีชื่อพิเศษเรียกว่า compatibility หรือ correct reception of color TV transmission by B/W TV receivers เนื่องด้วยการเลือกใช้มาตรฐานของโทรทัศน์สี จะต้องเข้ากันได้กับมาตรฐานของโทรทัศน์ขาวดำ ปัญหาต่างๆ ในการรับชมโทรทัศน์จึงไม่เกิดขึ้น

โทรทัศน์สีในโลกปัจจุบันนี้ มีอยู่หลายแบบหลายระบบต่างๆ กัน สถานีโทรทัศน์ไทยทีวีสีช่อง 3 กับสถานีโทรทัศน์สีกองทัพบกช่อง 7 ซึ่งออกอากาศในกรุงเทพมหานคร เลือกส่งโทรทัศน์สีในระบบ PAL ออกอากาศในมาตรฐานยุโรป 625 เส้น ต่อภาพ 25 ภาพ ต่อวินาที แต่สถานีโทรทัศน์ในประเทศสหรัฐอเมริกา ญี่ปุ่น และอีกหลายๆ ประเทศ กลับเลือกส่งโทรทัศน์สีระบบ NTSC ออกอากาศในมาตรฐานอเมริกัน 525 เส้น ต่อภาพ 30 ภาพ ต่อวินาที แตกต่างกันออกไป จึงทำให้จำเป็นต้องมีโทรทัศน์สีระบบต่างๆ ขึ้นเพื่อความเหมาะสม ในปัจจุบันนี้ โทรทัศน์สีมีให้เลือกใช้กันแพร่หลายในโลกมีด้วยกัน 3 ระบบ ดังต่อไปนี้

(1) โทรทัศน์สีระบบ NTSC (National Television System Committee)

(2) โทรทัศน์สีระบบ PAL (Phase Alteration Line)

(3) โทรทัศน์สีระบบ SECAM (Sequential a Memoire)

3.1 การส่งและการรับโทรทัศน์สีทำได้อย่างไร

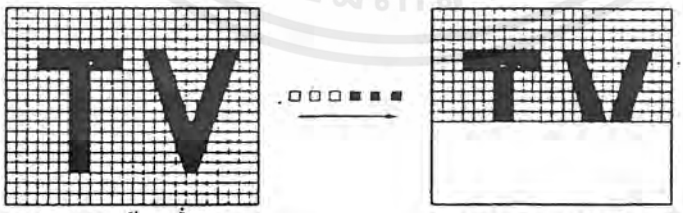
โทรทัศน์สีระบบต่างๆ มีหลักการส่งและหลักการรับโทรทัศน์คล้ายคลึงกับหลักการส่งและหลักการรับโทรทัศน์ขาวดำ กล่าวคือ แทนที่เครื่องส่งโทรทัศน์จะส่งสัญญาณภาพขาวดำไปให้เครื่องรับโทรทัศน์ เครื่องส่งโทรทัศน์สีก็จะส่งสัญญาณภาพสีไปแทน ภาพสีของเครื่องส่งโทรทัศน์สีประกอบด้วยภาพจากสัญญาณของแสงสีแดง – แสงสีเขียว – แสงสีน้ำเงิน เมื่อเครื่องรับโทร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทัศน์สีได้รับสัญญาณภาพสีนี้แล้ว ก็จะทำให้มีภาพสีปรากฏบนจอหลอดภาพของเครื่องรับโทรทัศน์สีขึ้น ภาพที่มองเห็นนี้จะละเอียดหรือหยาบไม่น่าดู ขึ้นอยู่กับสิ่งที่เกี่ยวข้องหลายอย่าง สิ่งที่สำคัญก็คือ จำนวนจุดดำหรือจุดเล็กๆ ที่เป็นส่วนประกอบของภาพ ซึ่งมีชื่อเรียกว่า picture element ภาพเหล่านี้จะประกอบขึ้นด้วยจุดดำหรือจุดสีเล็กๆ จำนวนมาก ซึ่งมีทั้งส่วนที่ดำสนิทและดำจาง หรือส่วนที่มีสีเข้มกับส่วนที่มีสีจาง ขนาดของจุดดำหรือจุดสีในภาพที่มืดสนิทหรือมีสีเข้ม จะมองเห็นใหญ่กว่าขนาดของจุดดำหรือจุดสีในส่วนของภาพสีจาง จำนวนจุดดำหรือจุดสีที่มีมากหรือน้อยนี้ จะมีผลทำให้ภาพมองละเอียดชัดเจนหรือหยาบไม่น่าดูแตกต่างกันด้วย ตามที่แสดงไว้ในรูปที่ 3.1 ในการนี้ ระยะทางที่มองภาพ ก็เข้ามามีส่วนเกี่ยวข้องอยู่ไม่น้อย ภาพที่หยาบ แต่ถ้ามองดูในระยะทางที่ไกลเกินกว่าระยะทางที่ใช้มองดูภาพละเอียด ก็จะรู้สึกว่า พอจะมองดูได้เหมือนกัน จุดเล็กๆ ซึ่งเป็นส่วนประกอบของภาพจะมีวิธีการทางไฟฟ้าทำให้เกิดเป็นสัญญาณภาพ ซึ่งเครื่องส่งโทรทัศน์จะนำออกอากาศ และทำให้เกิดภาพบนจอหลอดภาพในเครื่องรับโทรทัศน์ ตามที่แสดงไว้ในรูปที่ 3.2

(ก) หากมีจำนวนจุดดำมาก ภาพจอมองดูละเอียด (ข) หากมีจำนวนจุดดำน้อย ภาพจะมองดูหยาบ

รูปที่ 3.1 ส่วนประกอบของภาพในรูปซึ่งมีพื้นที่เท่ากัน



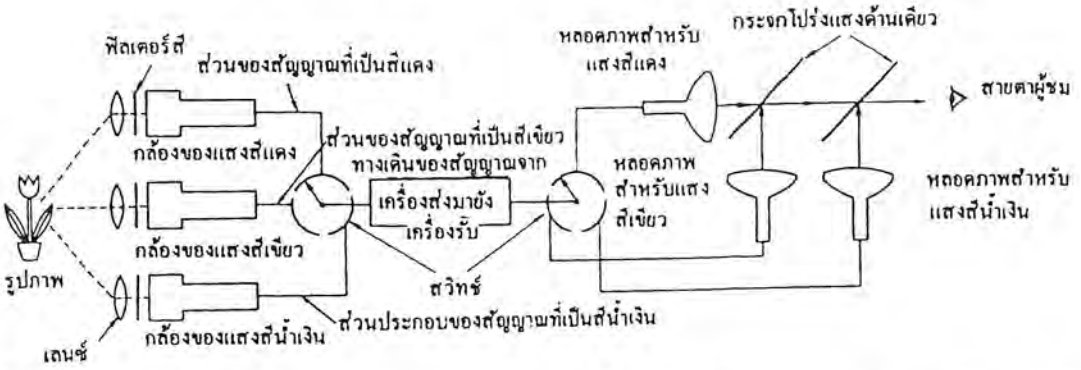
ภาพทางด้านเครื่องส่งโทรทัศน์

กล้องโทรทัศน์ จะเปลี่ยนแปลงภาพในห้องส่ง หรือภาพอื่นใดให้เป็นสัญญาณทางไฟฟ้า ซึ่งเรียกว่าสัญญาณภาพและสัญญาณนี้จะทำให้เกิดภาพบนจอหลอดภาพในเครื่องรับโทรทัศน์ขึ้น

ภาพบนจอหลอดภาพในเครื่องรับโทรทัศน์

รูปที่ 3.2 การส่งและการรับภาพทางโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

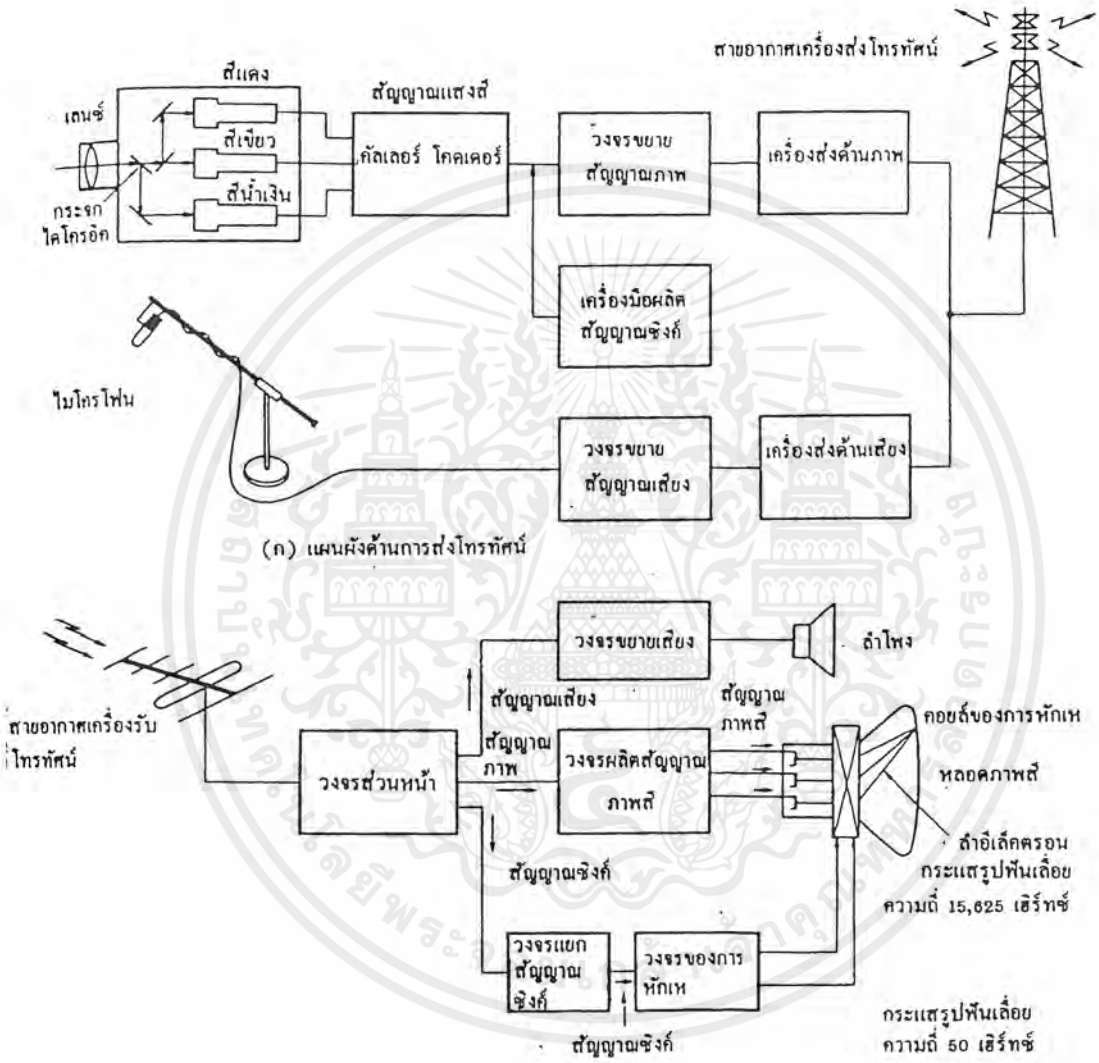


รูปที่ 3.3 การส่งโทรทัศน์สีระบบแสงสีแดง-แสงสีเขียว-แสงสีน้ำเงิน โดยส่งสัญญาณภาพสีด้วยวิธีการสลับแสงสี

รูปที่ 3.3 แสดงให้เห็นหลักการของการส่งโทรทัศน์สีโดยวิธี sequential transmitting system ซึ่งเป็นวิธีการในทางทฤษฎี สำหรับการส่งและการรับโทรทัศน์สีในทางปฏิบัติ เครื่องส่งโทรทัศน์สีและเครื่องรับโทรทัศน์สี จะต้องมีวงจรพิเศษเพื่อทำการส่งสัญญาณโทรทัศน์ขาวดำหรือสัญญาณส่องสว่าง (luminance signal) ปะปนไปกับสัญญาณโทรทัศน์ที่ให้ภาพสี ตามที่แสดงในรูป 3.4 แผนผังที่แสดงในรูปไว้ในรูปที่ 3.4(ก) เป็นแผนผังในด้านการส่งโทรทัศน์สี โดยกล้องโทรทัศน์สีที่ใช้ในห้องส่งโทรทัศน์ จะมีเลนส์และกระจก dichroic ทำการแยกภาพสีให้ปรากฏออกมาในรูปของสัญญาณแสงสีแดง – แสงสีเขียว – แสงสีน้ำเงิน แสงสีทั้งสามนี้ จะส่งผ่าน color coder ซึ่งจะช่วยทำให้เกิดสัญญาณโทรทัศน์ที่ให้ภาพสีปะปนไปกับสัญญาณโทรทัศน์ขาวดำ หรือสัญญาณส่องสว่าง เป็นสัญญาณภาพรวม ตามแต่ระบบของโทรทัศน์สีที่ใช้ เมื่อได้เติมสัญญาณซิ้งค์เพื่อส่งสัญญาณภาพสีไปในจังหวะของสัญญาณที่เหมาะสมแล้ว สัญญาณภาพรวมก็จะส่งผ่านวงจรขยายกำลังสัญญาณภาพ (video amplifier) และผ่านวงจรต่างๆ ในเครื่องส่งโทรทัศน์ด้านภาพ ซึ่งเมื่อได้นำมารวมกับสัญญาณเสียงที่ผ่านเครื่องส่งโทรทัศน์ด้านเสียงแล้ว ก็จะกลายเป็นสัญญาณโทรทัศน์สีซึ่งพร้อมที่จะนำออกอากาศ สายอากาศของเครื่องรับโทรทัศน์สีก็จะรับเอาสัญญาณโทรทัศน์สีที่ส่งมานี้มาทำให้เกิดเสียงและภาพสีทางเครื่องรับโทรทัศน์สีต่อไป แผนผังในรูปที่ 3.4(ข) เป็นแผนผังในด้านของการรับโทรทัศน์สี สัญญาณโทรทัศน์สีที่สายอากาศโทรทัศน์สีรับได้ ก็จะผ่านวงจรส่วนหน้า (tuner or front end) ซึ่งจะแยกสัญญาณเสียงออกจากสัญญาณภาพรวม เพื่อทำให้เกิดเสียงที่ลำโพงเครื่องรับโทรทัศน์สีสำหรับสัญญาณภาพรวม ก็จะแยกออกเป็นส่วนของสัญญาณโทรทัศน์ขาวดำหรือสัญญาณความส่องสว่าง (luminance signal) และสัญญาณโทรทัศน์สีที่ให้ภาพสี (chrominance signal) สัญญาณโทรทัศน์ส่วนที่ให้ภาพสี จะผ่านวงจรผลิตภาพสี (color regenerating circuit) เพื่อแยกออกเป็นสัญญาณแสงสีแดง – แสงสีเขียว – แสงสีน้ำเงิน เพื่อส่งต่อไปให้แคโทดของหลอดภาพโทรทัศน์สีของหลอดภาพโทรทัศน์สี ส่วนสัญญาณโทรทัศน์ขาวดำหรือสัญญาณส่องสว่างนั้น ก็จะผ่านวงจรต่างๆ ตามที่เคยพบเห็นมาแล้วในเครื่องรับโทรทัศน์ขาวดำ ส่วนประกอบต่างๆ ของวงจรเครื่องรับโทรทัศน์สีเหล่านี้ จะช่วยทำให้เกิดเป็นภาพสีขึ้นที่จอหลอด

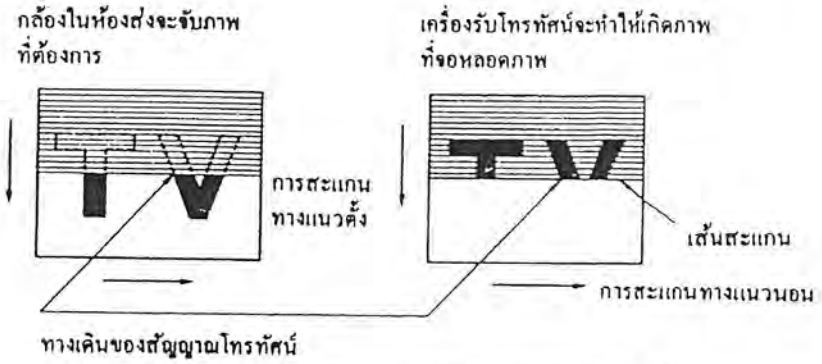
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของเครื่องรับโทรทัศน์สีตามต้องการ การส่งภาพสีนั้น ก็มีวิธีการเช่นเดียวกับการส่งภาพขาวดำ กล่าวคือ กล้องในห้องส่งโทรทัศน์จะมีการสแกนภาพที่ต้องการส่งไปทีละเส้นทีละภาพ โดยจะต้องใช้สัญญาณซิงค์ เพื่อทำให้การสแกนทางกล้องโทรทัศน์ในห้องส่งกับการสแกนทางจอหลอดภาพของเครื่องรับโทรทัศน์เป็นไปในจังหวะเดียวกัน ตามที่ได้แสดงไว้แล้วในรูปที่ 3.4



รูปที่ 3.4 ระบบการส่งและการรับโทรทัศน์สี

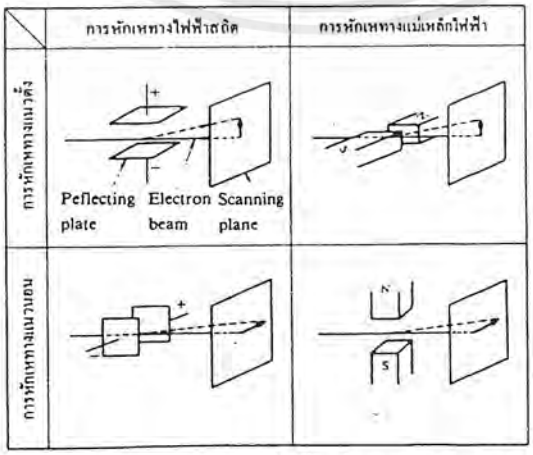
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 หลักการส่งและหลักการรับภาพทางโทรทัศน์

3.2 การสแกนและเรื่องที่เกี่ยวข้อง

ภาพบนจอหลอดภาพของเครื่องรับโทรทัศน์สีโดยทั่วไป จะประกอบด้วยเส้นขวางเล็กๆ ในแนวนอนเป็นจำนวนมาก ซึ่งแต่ละเส้นเหล่านี้ มีทั้งส่วนที่ดำสนิทหรือมีสีเข้ม ส่วนที่ดำจางหรือมีสีจาง และส่วนที่สว่างมากปะปนกันอยู่ เส้นขวางเล็กๆ ในแนวนอนเหล่านี้ มีชื่อเรียกว่า เส้นสแกน เส้นเหล่านี้ประกอบไปด้วยจุดเล็กๆ ซึ่งมีทั้งมืดและสว่างปะปนกันอยู่ ภาพที่ปรากฏบนจอหลอดภาพจึงประกอบด้วยจุดเล็กๆ ที่มีระดับของความสว่างแตกต่างกันเป็นจำนวนมาก จุดเล็กๆ เหล่านี้เรียกว่าส่วนประกอบของภาพ หรือ picture element ซึ่งมีส่วนสัมพันธ์กับความละเอียดของภาพเช่นเดียวกับจุดดำหรือจุดสีเล็กๆ ในรูปภาพของสิ่งตีพิมพ์ ภาพที่เห็นบนจอหลอดภาพจะมองดูละเอียดน่าดู หากจำนวนจุดเล็กๆ หรือจำนวนเส้นสแกนในแนวนอนมากเพียงพอ อย่างไรก็ตาม ภาพที่เห็นบนจอหลอดภาพจะมองละเอียดหรือหยาบไม่น่าดูอย่างไรนั้น ยังขึ้นอยู่กับส่วนประกอบอีกหลายอย่าง เช่น ความสว่างของภาพและระยะทางที่มองดูภาพ เป็นต้น สำหรับโทรทัศน์ระบบอเมริกัน ซึ่งมีจำนวนเส้นสแกนน้อยกว่า แต่ถ้าหากมองดูภาพในระยะห่างประมาณ 4-8 เท่าของความสูงของภาพแล้ว ก็จะรู้สึกว่าเป็นภาพพอใช้ได้เหมือนกัน

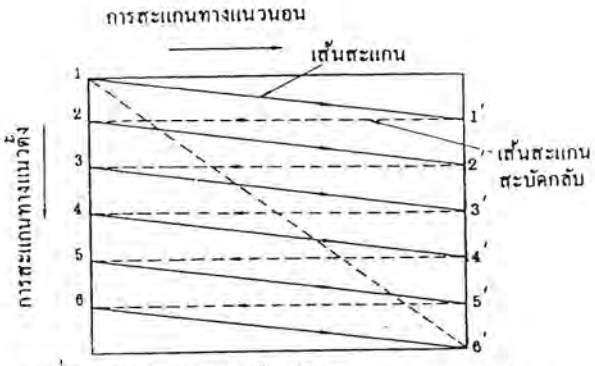


รูปที่ 3.6 ทฤษฎีของการหักเหทางไฟฟ้าสถิต และทางแม่เหล็กไฟฟ้า

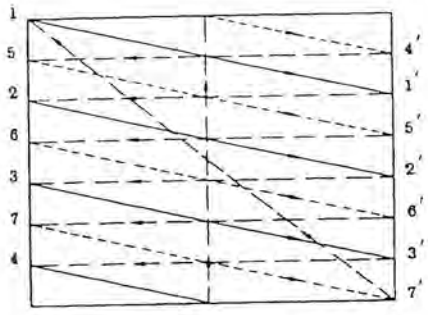
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่เห็นสว่างในจอหลอดภาพของเครื่องรับโทรทัศน์ เกิดขึ้นเพราะอิเล็กตรอนที่หลุดออกไปจากแคโทด และถูกดึงดูดให้วิ่งเป็นลำไปกระทบแอโนดหรือจอหลอดภาพ ซึ่งฉาบวัสดุเรืองแสงบางชนิดเอาไว้ จุดที่มีการกระทบกัน ก็จะมองเห็นเป็นจุดสว่างขึ้นที่จอ การสแกนก็คือ การทำให้จุดสว่างนี้เคลื่อนที่ไปในจังหวะที่ต้องการ ซึ่งในเรื่องของโทรทัศน์ ก็ต้องการให้จุดสว่างนี้เคลื่อนที่ไปในแนวนอนและแนวตั้ง โดยอาศัยความเข้มของสนามแม่เหล็กเข้าช่วยเหลือ ทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอน ในหลักการ การทำให้เกิดการดึงดูดหรือการผลักกันกับอิเล็กตรอนนี้ อาจทำได้โดยวิธีการหักเหของไฟฟ้าสถิต (electrostatic deflection) หรือวิธีการหักเหของแม่เหล็กไฟฟ้า (electromagnetic deflection) ตามที่แสดงไว้ในรูปที่ 3.6 ซึ่งวิธีการหลังนี้เป็นที่นิยมกันมากในทางปฏิบัติ สนามแม่เหล็กนี้เกิดขึ้นโดยการปล่อยกระแสไฟฟ้ารูปฟันเลื่อยให้ไหลผ่านขดลวดของการหักเห (deflection coil) ที่พันอยู่รอบๆ คอหลอดภาพ ซึ่งมีอยู่ด้วยกันสองชุดคือ ขดลวดที่พันอยู่รอบคอหลอดภาพในแนวนอนชุดหนึ่ง และขดลวดที่พันอยู่รอบคอหลอดภาพในแนวตั้งอีกชุดหนึ่ง สำหรับโทรทัศน์ระบบยุโรป ความถี่ของกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวนอน จะมีค่า 16,625 เฮิร์ทซ์ ส่วนกระแสรูปฟันเลื่อยที่ไหลผ่านขดลวดของการหักเหในแนวตั้ง จะมีค่าเพียง 50 เฮิรทซ์ เท่านั้น โดยปกติการสแกนจะเริ่มต้นขึ้นโดยการทำให้จุดสว่างบนจอหลอดภาพเคลื่อนที่จากซ้ายมือด้านบนของจอไปขวามือในแนวนอน ซึ่งเมื่อถึงตำแหน่งขวามือสุด ก็จะถูกเบนต่ำลงเล็กน้อย อันเป็นผลจากการที่มีกระแสรูปฟันเลื่อยไหลผ่านขดลวดของการหักเหในแนวตั้ง แล้วก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือเพื่อเคลื่อนที่มาทางขวามือในแนวนอนอีก เป็นอยู่เช่นนี้เรื่อยๆ จนกระทั่งจุดสว่างนั้นไปถึงตำแหน่งขวามือข้างล่างสุดของจอหลอดภาพ จึงเป็นอันเสร็จสิ้นการสแกนภาพนิ่งภาพหนึ่ง หรือเรียกกันว่า เฟรมหนึ่ง ตามที่แสดงไว้ในรูป 3.7 หลังจากนั้น ลำอิเล็กตรอนก็จะกลับไปตั้งต้นใหม่ทางซ้ายมือด้านบนสุดของจอหลอดภาพอีกเพื่อสแกนภาพอันดับถัดไป อย่างไรก็ตามเพื่อลดอาการกระพริบของการสแกนภาพนิ่งแต่ละภาพ มักนิยมจัดทำสองครั้งในแบบของการสแกนไขว้กัน ซึ่งเรียกว่า interlace scanning ตามที่แสดงไว้ในรูปที่ 3.8 โดยกำหนดให้ภาพนิ่งหนึ่งเฟรม (frame) ประกอบด้วยภาพนิ่งสองฟิลด์ (field) และเริ่มต้นด้วยการสแกนภาพนิ่งฟิลด์เส้นคี่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งขวามือล่างสุดของจอหลอดภาพแล้ว จึงกลับไปตั้งต้นใหม่ทางซ้ายมือด้านบนสุด แล้วเริ่มสแกนภาพนิ่งฟิลด์เส้นคู่ต่อไปจนถึงตำแหน่งขวามือล่างสุด หลังจากนั้นจึงจะเริ่มต้นสแกนภาพนิ่งอันดับอื่นต่อไปใหม่ ดังนั้นภาพนิ่งหนึ่งภาพหรือภาพนิ่งหนึ่งเฟรม จึงประกอบด้วยฟิลด์เส้นสแกนเส้นคี่และฟิลด์เส้นสแกนเส้นคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



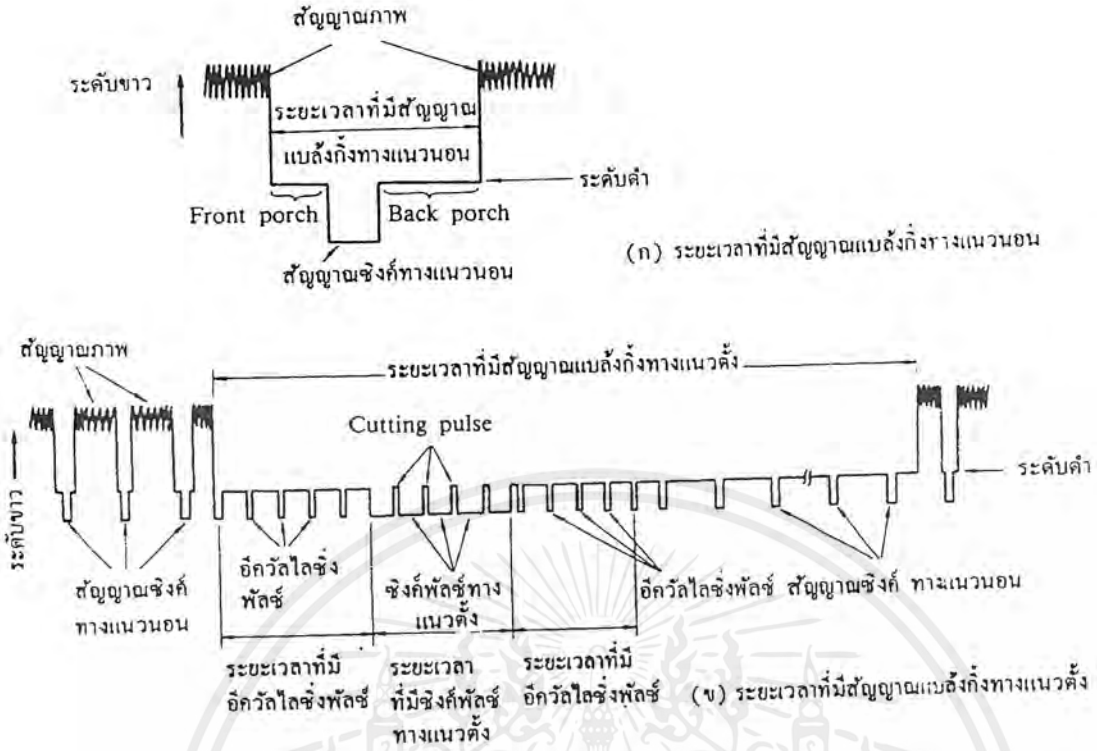
รูปที่ 3.7 การสแกนจากซ้ายไปขวาและจากบนลงล่าง



รูปที่ 3.8 การสแกนไขว้กัน

เรื่องที่สำคัญอีกอย่างหนึ่งของการส่งและการรับโทรทัศน์ก็คือ จะต้องสามารถหาวิธีการซึ่งทำให้การสแกนของภาพที่เกิดขึ้นในกล้องโทรทัศน์นั้น เกิดขึ้นพร้อมกันกับการสแกนของภาพที่จอหลอดภาพของเครื่องรับโทรทัศน์ หรือทำให้ความถี่ของกระแสรูปฟันเลื่อยของวงจรหักเหทางแนวนอนและแนวตั้งทางกล้องโทรทัศน์เท่ากันตลอดเวลา กับความถี่ของวงจรหักเหทางแนวนอนและแนวตั้งทางจอหลอดภาพของเครื่องรับโทรทัศน์ หากความถี่ของกระแสรูปฟันเลื่อยในวงจรทางเครื่องส่งโทรทัศน์ไม่เท่ากันตลอดเวลากับความถี่ของกระแสรูปฟันเลื่อยในวงจรทางเครื่องรับโทรทัศน์ ก็จะพบว่า ภาพจะล้มหรือไม่มีเครื่องรับโทรทัศน์ การทำให้ความถี่ของกระแสรูปฟันเลื่อยทางด้านเครื่องส่งโทรทัศน์เท่ากันตลอดเวลา กับความถี่ของกระแสรูปฟันเลื่อยทางด้านเครื่องรับโทรทัศน์นี้ เรียกว่า เกิดการเข้าจังหวะ (synchronization) ขึ้น ในทางปฏิบัติสถานีโทรทัศน์จะต้องส่งสัญญาณชนิดหนึ่งเรียกว่าสัญญาณซิงค์ (synchronizing signal หรือ sync pulse signal) ไปพร้อมกับสัญญาณภาพ รูป 3.9 สัญญาณซิงค์นี้จะประกอบด้วยสัญญาณซิงค์ทางแนวนอน (horizontal synchronizing signal) ซึ่งมีความถี่ 15,625 เฮิร์ทซ์ หรือจะมี sync pulse ครั้งหนึ่งในทุกๆ ครั้งที่มีเส้นสแกนในแนวนอนกับสัญญาณซิงค์ทางแนวตั้ง (vertical synchronizing signal) ซึ่งมีความถี่ 50 เฮิร์ทซ์ หรือจะมี sync pulse ครั้งหนึ่งในขณะที่มีการสแกนฟิลด์เส้นคี่หรือฟิลด์เส้นคู่เสร็จสิ้นลง สัญญาณซิงค์เหล่านี้จะส่งไปพร้อมๆ กับสัญญาณภาพ ในช่วงระยะเวลาของเส้นสแกนสะบัดกับ หรือช่วงระยะเวลาที่เส้นสแกนกำลังหันกลับไปเริ่มต้นใหม่ (flyback period)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 รูปร่างของสัญญาณซิงค์ที่ใช้ในการส่งโทรทัศน์

3.3 สัญญาณภาพรวม

หากจะถามว่าเครื่องส่งทำการส่งสัญญาณอะไรมาให้เครื่องรับรับบ้าง หากตอบกันง่ายๆ ก็ต้องตอบว่าส่งสัญญาณภาพรวม (Composite Video Signal) ซึ่งการที่เราจะทำให้เครื่องรับโทรทัศน์บรรลุวัตถุประสงค์ได้นั้นต้องให้สถานีโทรทัศน์ส่งสัญญาณต่างๆ ดังนี้

1. สัญญาณเสียง
2. สัญญาณภาพ
3. สัญญาณแบลงกิ้ง
4. สัญญาณซิงโครไนซ์
5. สัญญาณอีควอไลซิ่ง

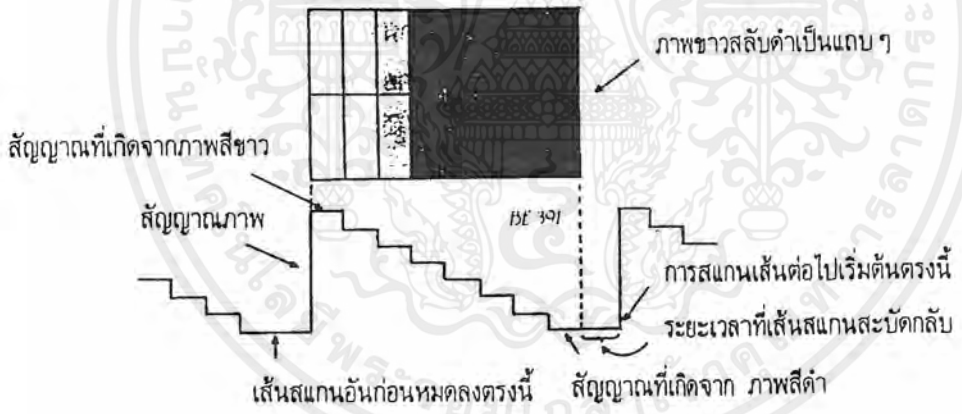
ในส่วนของระบบสัญญาณเสียงเราจะใช้คลื่นพาห์ (Carrier) เฉพาะ เพราะทราบกันบ้างแล้วว่าระบบเสียงในโทรทัศน์เป็นระบบเอฟเอ็ม ส่วนสัญญาณภาพและอื่นๆ ที่เหลือนั้นเราจะส่งเป็นสัญญาณภาพรวมหรือคอมโพสิท วีดีโอ ซิกแนล (Composite Video Signal) แล้วใช้คลื่นพาห์ภาพส่งออกไป การที่เราต้องส่งสัญญาณทั้ง 5 ตัว ออกอากาศแพรคลื่นออกไปเพื่อวัตถุประสงค์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **สัญญาณภาพและสัญญาณเสียง** เป็นสัญญาณที่ส่งไปเพื่อให้เกิดภาพและเสียงขึ้นในเครื่องรับโทรทัศน์
2. **สัญญาณแบลงกิ้ง** เป็นสัญญาณที่ส่งเพื่อให้ลบเส้นสะบัดกลับทั้งในแนวตั้งและแนวนอน
3. **สัญญาณซิงโครไนซ์** เป็นสัญญาณที่ส่งมาเพื่อช่วยให้วงจรหักเหทางแนวตั้งและวงจรหักเหทางแนวนอน เพื่อให้เครื่องส่งกับเครื่องรับทำงานสอดคล้องตรงกัน
4. **สัญญาณอีควอไลซิ่ง** เป็นสัญญาณที่ช่วยให้สัญญาณซิงโครไนซ์ทั้งแนวตั้งและแนวนอนยังคงรูปเดิมอยู่ได้ แม้ว่าจะเป็นการสแกนสลับเส้นก็ตาม

ก. สัญญาณภาพขาว - ดำ

สมมุติว่าเราจะดูระดับสัญญาณขาว – ดำ กรณีที่เรากล่าวถึงสัญญาณขาว – ดำ หรือสัญญาณโมโนโครมได้ดีที่สุดต้องกล่าวว่าภาพที่เป็นสีขาวคือภาพที่มีความสว่างมากที่สุด และภาพที่เป็นสีดำคือ ภาพที่ไม่มีความสว่างเลย ภาพจำลองที่ดีที่สุดของกรณีก็คือแถบภาพที่มีความแตกต่างของระดับขาวดำทีละน้อย ซึ่งเราเรียกว่าระดับเกรย์สเกลนั่นเอง



รูปที่ 3.10 แสดงระดับสัญญาณขาว-ดำ โดยเกรย์ สเกล

จากภาพแสดงให้เห็นระดับของเกรย์สเกล ในกรณีที่ระดับสัญญาณภาพเป็นสัญญาณสีขาวเข้ามา นั่นหมายความว่าระดับความแรงของสัญญาณภาพมากที่สุดจึงให้ความส่องสว่างที่หน้าจอสว่างมากที่สุด และเมื่อระดับสัญญาณสีขาวลดลงเป็นสีมัว, เทา และดำนั้น ระดับสัญญาณจะลดลงมาเรื่อยๆ นั่นหมายความว่าเมื่อสัญญาณมีความแรงน้อยลงความส่องสว่างจะน้อยลงตามไปด้วย

อันที่จริงแล้วภาพแต่ละภาพเป็นสัญญาณความถี่ทางไฟฟ้าที่มีความถี่สูงต่ำไม่เท่ากัน โดยความถี่สูงสุดจะไม่เกิน 4 เมกะเฮิรตซ์ในระบบเอฟซีซี และไม่เกิน 5 เมกะเฮิรตซ์ในระบบซีซีไออาร์ ภาพที่เกิดจากความถี่สูงย่อมมีความละเอียดกว่าความถี่ต่ำ (มีจำนวนจุดดำมากกว่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. สัญญาณแบลงกิ้ง

ทราบแล้วว่าเมื่อมีการสแกนลำอิเล็กตรอนที่หน้าจอจะเกิดเส้นรีเทรสหรือเส้นสะบัดกลับ ซึ่งเป็นเส้นภาพที่เราไม่ต้องการ เครื่องส่งจึงต้องส่งสัญญาณแบลงกิ้ง เพื่อบังคับให้เครื่องรับสามารถลบเส้นสะบัดกลับได้ สัญญาณแบลงกิ้งส่วนหนึ่งเครื่องรับจะต้องสร้างขึ้นเหมือนการสร้างสัญญาณซิงโครไนซ์แต่มันจะสัมพันธ์กับเครื่องส่งได้อย่างไร จึงต้องมีการส่งสัญญาณแบลงกิ้งมาจากเครื่องส่งเพื่อลบเส้นสะบัดกลับในเครื่องรับ สัญญาณแบลงกิ้งมีอยู่ 2 อย่างคือ เวอร์ติคอลแบลงกิ้ง กับฮอริซอนตอลแบลงกิ้ง

ค. สัญญาณซิงโครไนซ์

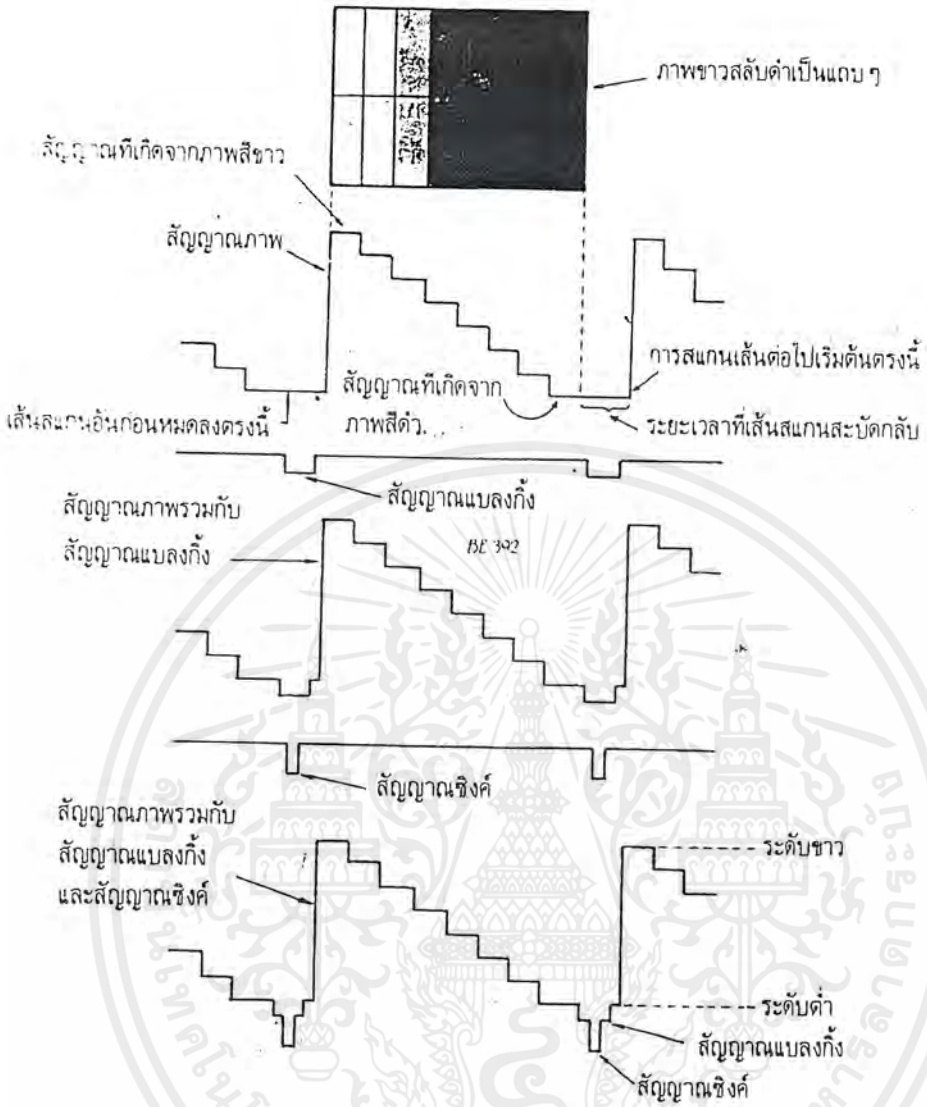
เป็นสัญญาณเพื่อให้การสแกนเป็นไปอย่างถูกต้อง ทั้งการสแกนแนวตั้งและแนวนอน โดยสัญญาณซิงโครไนซ์มีอยู่ 2 สัญญาณ คือ

1. ฮอริซอนตอล ซิงโครไนซ์ (Horizontal Synchronize) เป็นสัญญาณซิงโครไนซ์ทางแนวนอน ซึ่งมีความถี่ 15,625 เฮิรตซ์ (ในระบบ CCIR) หรือ 15,750 เฮิรตซ์ (ในระบบ FCC) ถ้าไม่มีสัญญาณส่วนนี้ส่งมาจะทำให้ภาพเกิดการล้มได้

2. เวอร์ติคอล ซิงโครไนซ์ (Vertical Synchronize) เป็นสัญญาณซิงโครไนซ์ทางแนวตั้ง ซึ่งมีความถี่ 50 เฮิรตซ์ (ในระบบ CCIR) หรือ 60 เฮิรตซ์ (ในระบบ FCC) ถ้าไม่มีการส่งสัญญาณส่วนนี้มาจะทำให้ภาพเลื่อน

เนื่องจากสัญญาณซิงโครไนซ์กับสัญญาณแบลงกิ้งไม่ว่าจะเป็นแนวนอนหรือแนวตั้ง จะมีความถี่เท่ากัน เวลาส่งจึงต้องกำหนดตำแหน่งของการส่งให้ถูกต้อง มิฉะนั้นแล้วจะเกิดกวนกันได้ ในทางปฏิบัติจึงให้สัญญาณซิงค์พัลซ์มีขนาดความกว้างน้อยกว่าแบลงกิ้งพัลซ์ แล้วใช้วิธีการส่งแบบผสม กล่าวคือส่งซิงค์พัลซ์กับแบลงกิ้งพัลซ์ไปด้วยกัน ให้แบลงกิ้งพัลซ์เป็นฐานของสัญญาณซิงค์พัลซ์ เมื่อมีการจัดระดับของสัญญาณส่วนนี้เทียบกับเกรย์สเกล ระดับของแบลงกิ้งจะอยู่ที่ระดับต่ำกว่าดำ ส่วนซิงค์จะเป็นระดับต่ำกว่าระดับดำมากกว่าลงไปอีก สัญญาณเหล่านี้จึงไม่มีการส่งผลต่อการมองเห็น (หรือไม่กวนสัญญาณภาพ) ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงลักษณะของสัญญาณภาพรวมเบื้องต้น

ง. สัญญาณอีควอไลซิ่ง

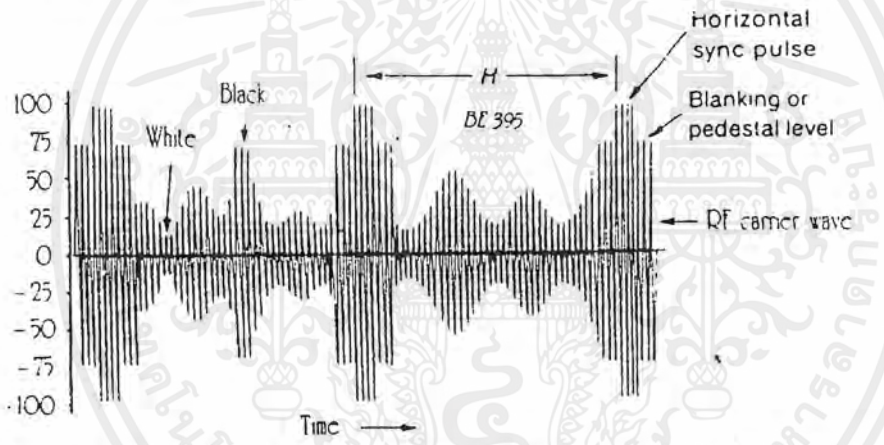
เป็นสัญญาณบังคับรูปร่างของสัญญาณซิงโครไนซ์ทางแนวตั้ง เพื่อให้สามารถคงรูปถูกต้อง แล้วยังช่วยให้การสแกนแบบสลับเส้นเป็นไปได้ถูกต้องด้วย สามารถขึ้นเส้นคู่เส้นคี่ในตำแหน่งที่ถูกต้องได้ ทั้งยังส่งผลทางอ้อมให้สัญญาณซิงโครไนซ์ทางแนวนอนไม่ขาดช่วงหายไปในระหว่างการส่งสัญญาณทางแนวตั้งด้วย สัญญาณตัวนี้จะมีขนาดของพัลซ์รวมเท่ากับเวอร์ติคอลซิงโครไนซ์ พัลซ์ สัญญาณส่วนนี้จะถูกแบ่งออกเป็น 6 ลูกเล็กๆ ในระบบ 525 เส้น และถูกแบ่งเป็น 5 ลูกในระบบ 625 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.4 การผสมสัญญาณโทรทัศน์

สัญญาณโทรทัศน์ประกอบด้วยสัญญาณใหญ่ๆ 2 สัญญาณคือ สัญญาณภาพและสัญญาณเสียง ดังนี้ระบบการผสมความถี่หรือการมอดูเลชั่น (Modulation System) จึงค่อนข้างยุ่งยากพอสมควร ประการแรกก็คือสัญญาณภาพเป็นสัญญาณที่มีตัวประกอบหลายตัว ส่วนสัญญาณเสียงเป็นสัญญาณธรรมดาที่ไม่มีความยุ่งยากมากนัก สัญญาณภาพจะถูกมอดูเลชั่นในระบบเอเอ็ม (Amplitude Modulation) ทั้งนี้ในระบบเอเอ็ม การผสมสัญญาณมิได้ทำให้เกิดการแปรความถี่เป็นความถี่ใหม่ตามสัญญาณความถี่วิทยุแต่อย่างใด สัญญาณภาพเป็นสัญญาณที่มีสัญญาณซิงโครไนซ์, แบลงกิ้ง, อีควอไลซิ่ง ฯลฯ รวมอยู่ด้วย จึงทำให้แบนด์วิดธ์ในการมอดูเลชั่นสูงกว่าที่ควรจะเป็น ในเครื่องรับโทรทัศน์จึงค่อนข้างจะยุ่งยาก เมื่อเข้าสู่วิธีการเกี่ยวกับระบบการแปรสภาพสัญญาณให้เป็นภาพที่หน้าจอ แต่หากเรามอดูเลชั่นในระบบเอเอ็ม เราจะพบว่าความยุ่งยากจะลดน้อยลงหากพิจารณาจากหลักการเบื้องต้นที่ผ่านมา ดังรูป



รูปที่ 3.12 สัญญาณภาพรวมที่เครื่องส่งส่งมาซึ่งแสดงให้เห็นระดับต่างๆ ของสัญญาณที่เราคิดเป็นค่าเปอร์เซนต์ออกมาเพื่ออรรถาธิบายถึงความเป็นภาพหน้าจอ

3.5 ระบบของการผสมสัญญาณ

ระบบของการผสมสัญญาณหรือการมอดูเลชั่น อันเป็นวิธีการที่เราเอาสัญญาณความถี่วิทยุมาผสมกับข้อมูลข่าวสารที่เราต้องการสื่อสารออกไป มีระบบต่างๆ ที่สำคัญดังนี้คือ

1. แอมปลิจูด มอดูเลชั่น หรือระบบเอเอ็ม (Amplitude Modulation) เป็นวิธีการผสมสัญญาณ แล้วคลื่นพาห์หรือแคร์เรียร์มีแอมปลิจูดเปลี่ยนแปลงไปตามข้อมูลที่เราต้องการ

2. ฟรีเควนซี่ มอดูเลชั่น หรือเอฟเอ็ม (Frequency Modulation) เป็นวิธีการผสมสัญญาณ แล้วทำให้คลื่นพาห์หรือแคร์เรียร์มีการเปลี่ยนแปลงทางด้านความถี่ แปรตามสัญญาณข้อมูลที่ต้องการ

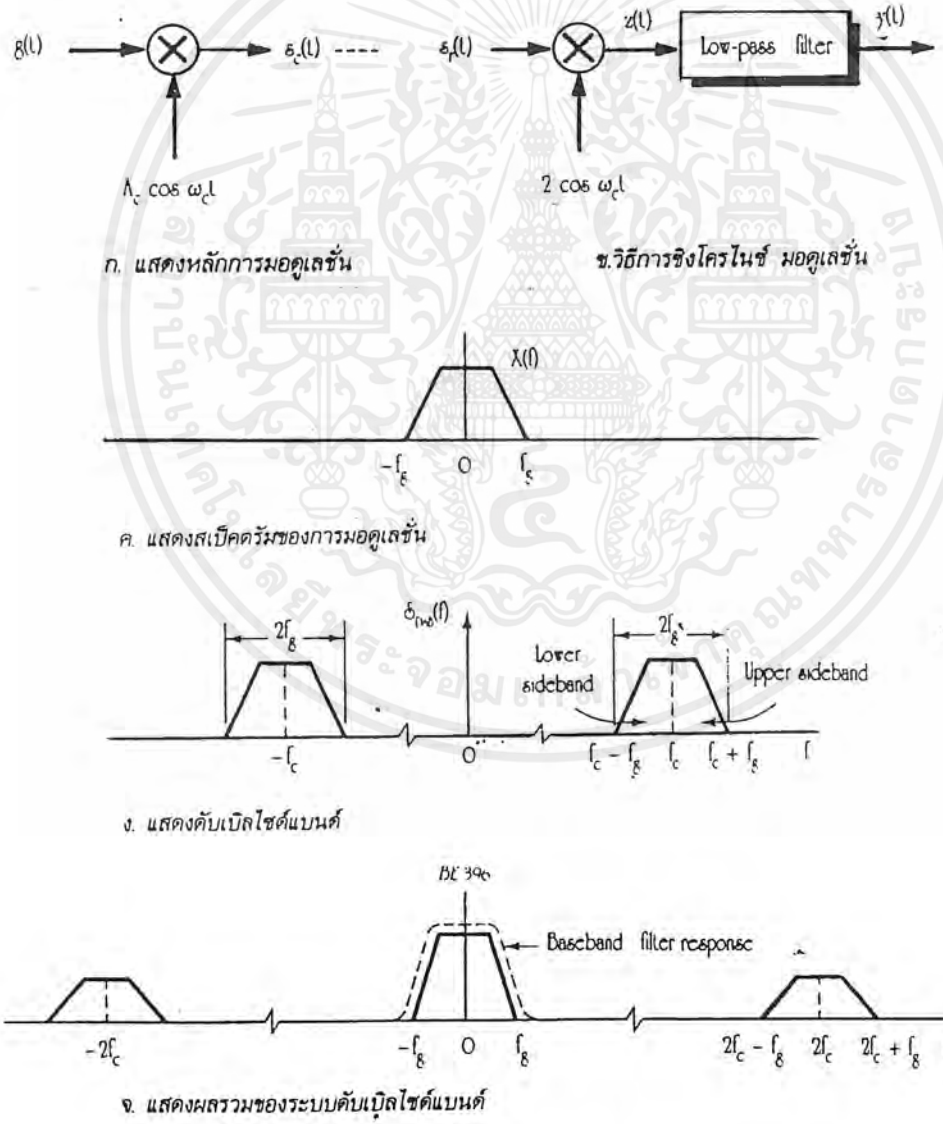
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46204

3. เฟสมอดูเลชั่น หรือระบบพีเอ็ม (Phase Modulation) เป็นวิธีการเปลี่ยนแปลงค่าพารามิเตอร์หรือค่าคุณสมบัติของข้อมูลให้เป็นไปตามเฟสของคลื่นพาห์

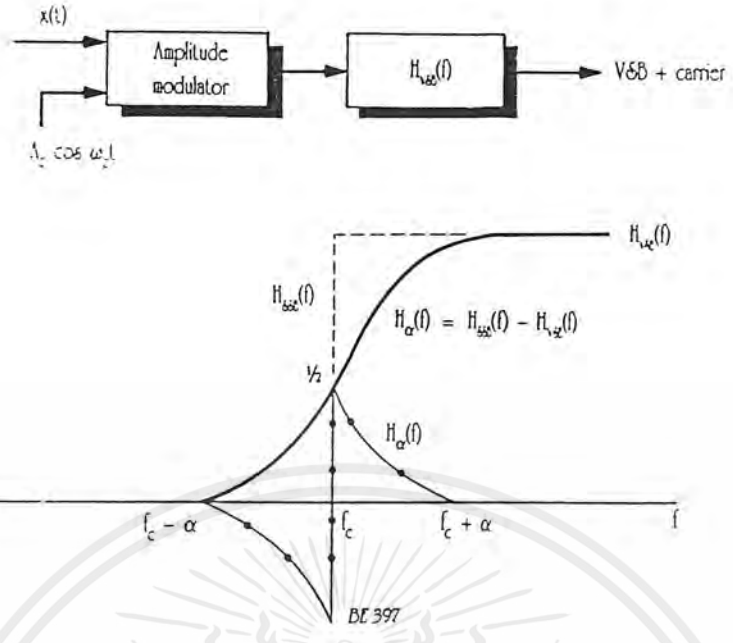
4. พัลซ์มอดูเลชั่น หรือระบบพีซีเอ็ม (Pulse Code Modulation) เป็นลักษณะของการผสมความถี่ให้เป็นรหัสสัญญาณพัลซ์ ซึ่งปัจจุบันที่มีใช้อยู่อาจจะเป็นระบบพีเอเอ็ม (PAM : Pulse Amplitude Modulation) ระบบพีดีเอ็ม หรือพีดับเบิลยูเอ็ม (PDM, PWM : Pulse Duration (Width) Modulation) และระบบพีพีเอ็ม (PPM : Pulse Position Modulation)

สัญญาณภาพจะถูกมอดูเลชั่นในระบบเอเอ็ม ส่วนสัญญาณเสียงสำหรับโทรทัศน์เรามอดูเลชั่นในระบบเอฟเอ็ม โดยมีข้อกำหนดว่าสัญญาณทั้ง 2 อย่างนี้จะต้องมีความห่างกันไม่มากเกินไปนัก ไม่นั้นแล้วแบนด์วิดธ์ของการส่งจะกว้างมาก ทำให้เป็นเรื่องยากของระบบการส่งและระบบของการรับ



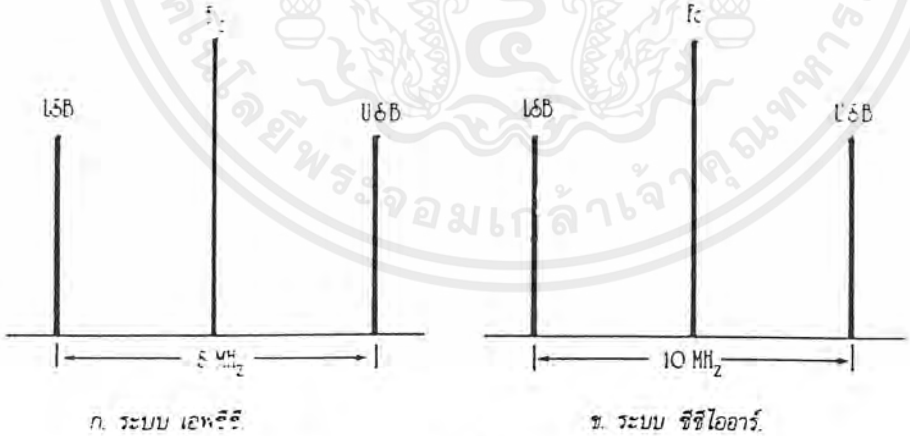
รูปที่ 3.13 วิธีการ เอเอ็มดับเบิลไซด์แบนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 วิธีการของเวสติเจียลไซด์แบนด์

สัญญาณภาพเมื่อรวมเอาสัญญาณที่เรียกรวมว่าสัญญาณภาพรวมเข้ามาด้วยแล้ว จะได้ความกว้างของย่านความถี่เท่ากับ 5 เมกะเฮิรตซ์ ในระบบ 625 เส้น และ 4 เมกะเฮิรตซ์ ในระบบ 525 เส้น เมื่อมอดูเลชั่นแล้วจะได้แบนด์วิดธ์ในระบบ 525 เส้น เท่ากับ 8 เมกะเฮิรตซ์ และ 10 เมกะเฮิรตซ์ ในระบบ 625 เส้น ดังภาพ



ก. ระบบ เอ็ม/ซี ข. ระบบ ซีซีไออาร์

รูปที่ 3.15 แสดงแบนด์วิดธ์เบื้องต้นของสัญญาณภาพ

การส่งแบบเอเอ็มธรรมดา จะพบว่าแบนด์วิดธ์กว้างมาก ทำให้ย่านความถี่วีเอชเอฟ ส่งคลื่นสถานีโทรทัศน์ได้น้อยช่อง ดังนั้นวิธีการของเอเอ็ม ที่เรียกว่าระบบเวสติเจียล ไซด์แบนด์ จึงถูกหยิบยกออกมาใช้งาน วิธีการนี้จะมีไซด์แบนด์แตกต่างไปจากระบบเอเอ็มธรรมดาตรงที่ว่าหาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรามอดูเลชั่นในระบบเอเอ็มธรรมดา จะพบว่าจะเกิดแถบความถี่ข้างเคียงคือโลเวอร์ไซแบนด์ กับอัปเปอร์ไซด์แบนด์ ซึ่งไซด์แบนด์รวมทั้งหมดหากเป็นสัญญาณภาพก็ยังคงได้เท่ากับ 10 เมกะเฮิรตซ์ตามที่กล่าวมาแล้ว หากเราจะลดแบนด์วิดธ์ลงสามารถทำได้โดยการมอดูเลชั่นแบบเอาไซด์แบนด์ออกด้านเดียว หรือระบบซิงเกิลไซแบนด์ (Single Sideband) ซึ่งระบบนี้จะทำการกดแคร์เรียร์ทิ้ง (Suppress Carrier) ดังนั้นการดีโค้ดเดอร์หรือดีมอดูเลชั่นในเครื่องรับจะมีความซับซ้อนมากขึ้น อันอาจจะไม่สะดวกต่อการสร้างวงจร และนอกจากนั้นวิธีการดีมอดูเลชั่น หรือการดีเทคเตอร์จะต้องสร้างความถี่แคร์เรียร์ขึ้นมาเอง เหมือนวิทยุสนามในวงการทหาร เมื่อเป็นเช่นนั้นเราจะดีเทคเตอร์สัญญาณภาพด้วยวิธีง่ายๆ ได้อย่างไร ข้อมูลภาพเป็นลักษณะข้อมูลบรอดคาสติ้ง มิใช่ข้อมูลลับของทางราชการแต่อย่างใด จึงมีการนำเอาระบบเวสติเจียลไซด์แบนด์ ดังแสดงกรรมวิธีไว้ในรูป

วิธีการนี้ยอมให้อัปเปอร์ไซด์แบนด์ออกไปได้เต็มที่ ในขณะที่โลเวอร์ไซด์แบนด์ออกได้เพียงเล็กน้อย นั่นหมายความว่าวิธีการนี้มิได้กดแคร์เรียทิ้งเหมือนระบบไซด์แบนด์ทั่วไป จึงทำให้วงจรดีเทคเตอร์ในภาครับสามารถดีเทคเตอร์ได้ด้วยวงจรธรรมดา

ในขณะเดียวกันสัญญาณเสียงจะถูกมอดูเลชั่นในระบบเอฟเอ็ม ซึ่งถูกกำหนดให้คลื่นพาห์เสียงห่างจากคลื่นพาห์ภาพอยู่ 4.5 เมกะเฮิรตซ์ในระบบ 525 เส้น และ 5.5 เมกะเฮิรตซ์ในระบบ 625 เส้น เฉพาะสัญญาณเสียงโดยทั่วไปแล้วเราให้ความกว้างของความถี่เสียงประมาณ 12 กิโลเฮิรตซ์ก็พอแล้ว ในเครื่องรับโทรทัศน์เราให้ย่านความถี่ของเสียงสูงสุดได้ถึง 25 กิโลเฮิรตซ์ และแบนด์วิดธ์เสียงรวมแล้วประมาณ 0.5 เมกะเฮิรตซ์ จึงทำให้แชตแนลของระบบ 525 เส้น มีแบนด์วิดธ์ของแชนแนลเท่ากับ 6 เมกะเฮิรตซ์ และระบบ 625 เส้นต้องมีแบนด์วิดธ์ 7 เมกะเฮิรตซ์

3.6 สัญญาณโทรทัศน์สีประกอบด้วยอะไรบ้าง

ความต้องการในเรื่องการให้บริการโทรทัศน์สีในเขตบริการที่มีเครื่องรับโทรทัศน์ขาวดำอยู่ด้วย มีอยู่ด้วยกัน 3 ประการ ประการแรกคือ เรื่องของ compatibility กับ reverse compatibility ประการที่สองคือ สัญญาณโทรทัศน์สีที่ส่งออกอากาศ จะต้องประกอบด้วยส่วนที่เป็นสัญญาณส่องสว่าง (luminance signal) กับส่วนที่เป็นสัญญาณโทรทัศน์สีที่ให้ภาพสี (chrominance signal) ประการสุดท้ายก็คือ ความถี่ของสัญญาณโทรทัศน์ที่ให้ภาพสี จะต้องมีขอบเขตอยู่ในความถี่เดียวกันกับสัญญาณส่องสว่าง ในทางปฏิบัติ กล้องโทรทัศน์สีจะช่วยเหลือทำให้เกิดสัญญาณแสงสีแดง – แสงสีเขียว – แสงสีน้ำเงิน ขึ้น ซึ่งหากจะทำการส่งสัญญาณแสงสีเหล่านี้ไปยังเครื่องรับโทรทัศน์โดยตรง จะเป็นการยุ่งยาก จึงใช้วิธีการนำสัญญาณแสงสีทั้งสามเหล่านี้มาผสมกันในวงจรพิเศษ ซึ่งเรียกว่า วงจรแมททริกซ์ (matrix) เพื่อทำให้เกิดเป็นสัญญาณใหม่สองสัญญาณ คือ สัญญาณโทร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทัศน์ขาวดำ หรือสัญญาณส่องสว่าง (luminance signal) และสัญญาณโทรทัศน์ที่ให้ภาพสี (chrominance signal) เมื่อสถานีโทรทัศน์สีส่งสัญญาณต่างๆ เหล่านี้ออกอากาศ เครื่องรับโทรทัศน์ขาวดำก็จะรับแต่เฉพาะสัญญาณโทรทัศน์ขาวดำ หรือส่วนที่เป็นสัญญาณส่องสว่าง ส่วนเครื่องรับโทรทัศน์สี ก็จะรับสัญญาณโทรทัศน์ทั้งหมด แล้วทำให้เกิดเป็นภาพสีขึ้นบนจอหลอดภาพต่อไป

กล่าวโดยสรุป สถานีโทรทัศน์สีจะต้องส่งสัญญาณต่างๆ ออกอากาศไปให้เครื่องรับโทรทัศน์เพื่อทำให้เกิดภาพสีขึ้นที่จอหลอดภาพของเครื่องรับโทรทัศน์สี และภาพขาวดำที่จอหลอดภาพของเครื่องรับโทรทัศน์ขาวดำ กับมีเสียงที่ลำโพงของเครื่องรับโทรทัศน์ ดังต่อไปนี้

- สัญญาณเสียง (sound signal)
- สัญญาณโทรทัศน์ขาวดำ หรือสัญญาณส่องสว่าง (luminance signal)
- สัญญาณโทรทัศน์ที่ให้ภาพสี (chrominance signal)
- สัญญาณซิงค์, แบล้งกิ้ง และอีควัลไลซิ่ง (synchronizing, blanking and equalizing signal)
- สัญญาณซิงค์ของภาพสี (color sync signal)

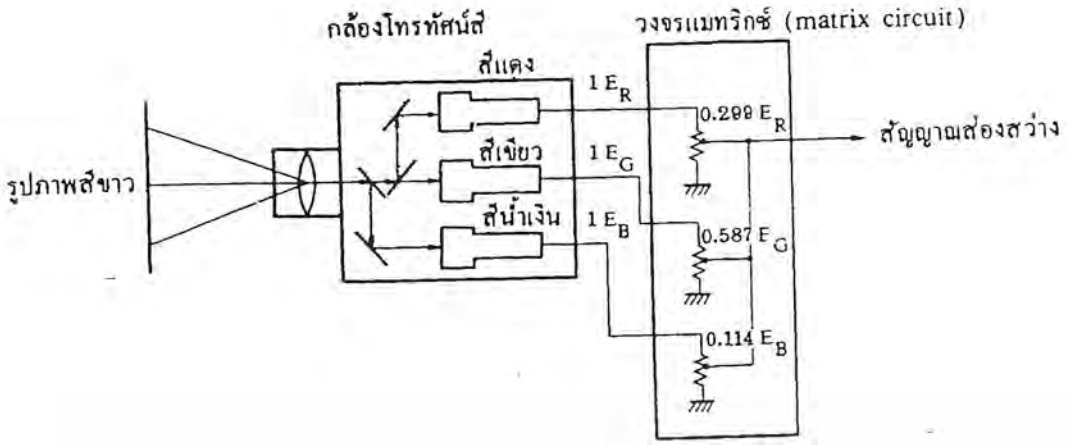
เหตุผลและความจำเป็นในการที่ต้องส่งสัญญาณต่างๆ เหล่านี้ มีดังต่อไปนี้

ก. สัญญาณเสียง (sound signal) เพื่อทำให้เกิดเสียงที่ลำโพงเครื่องรับโทรทัศน์ตามต้องการ

ข. สัญญาณโทรทัศน์ขาวดำ หรือสัญญาณส่องสว่าง (luminance signal) สัญญาณนี้ คือ สัญญาณภาพ (video signal or brightness signal) ในเรื่องของโทรทัศน์ขาวดำนั่นเอง กล้องโทรทัศน์สีในห้องส่งโทรทัศน์ จะช่วยทำให้เกิดสัญญาณแสงสีแดง – แสงสีเขียว – แสงสีน้ำเงิน ขึ้น ซึ่งเราอาจใช้สัญลักษณ์ E_R , E_G , E_B แทนค่าสัญญาณทางไฟฟ้าเป็นโวลต์ที่ได้จากหลอดสีแดง – หลอดสีเขียว – หลอดสีน้ำเงิน ตามลำดับ สัญญาณแสงสีทั้งสามนี้ จะผ่านวงจรพิเศษ เรียกว่า วงจรแมททริซ์ ตามที่แสดงไว้ในรูปที่ 3.16 เพื่อทำให้เกิดสัญญาณส่องสว่าง (brightness or video signal, or luminance signal) E_Y โดยมีส่วนผสมของแสงสีทั้งสามในอัตราส่วนที่แน่นอน ดังต่อไปนี้

$$E_Y = 0.299E_R + 0.587E_G + 0.114E_B$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



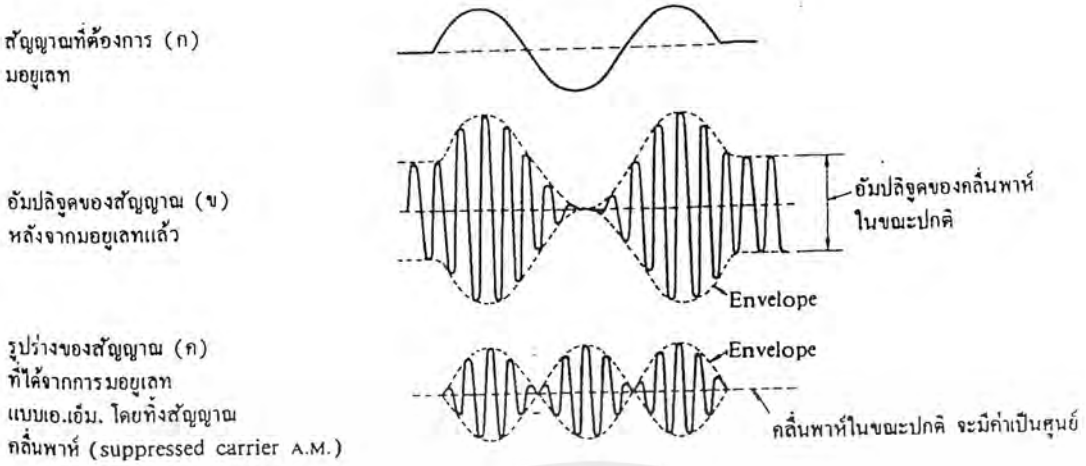
รูปที่ 3.16 สัญญาณส่องสว่าง

ค. สัญญาณโทรทัศน์ที่ให้ภาพสี (chrominance signal) สัญญาณนี้เป็นสัญญาณที่เครื่องส่งโทรทัศน์สีได้ทำการส่งไปยังเครื่องรับโทรทัศน์ เพื่อช่วยเหลือทำให้เกิดภาพสีขึ้นทางจอหลอดภาพของเครื่องรับโทรทัศน์สี คุณลักษณะของสัญญาณนี้ จะขึ้นอยู่กับระบบของโทรทัศน์สี ซึ่งมีอยู่ 3 ระบบ คือ โทรทัศน์สีระบบ NTSC โทรทัศน์สีระบบ PAL และโทรทัศน์สีระบบ SECAM วิธีการส่งสัญญาณโทรทัศน์ที่ให้ภาพสีรวมไปกับสัญญาณโทรทัศน์ขาวดำหรือสัญญาณส่องสว่าง โดยการใช้คลื่นพาห์ของสัญญาณสีนี้เป็นวิธีการพิเศษในทางไฟฟ้า ซึ่งเรียกว่า multiplex transmission

ง. สัญญาณซิงค์, แบล้งกิ้ง และอีควัลไลซิ่ง (synchronizing, blanking and equalizing signal) สัญญาณดังกล่าวนี้ มีลักษณะเช่นเดียวกันกับสัญญาณซิงค์ สัญญาณแบล้งกิ้ง และสัญญาณอีควัลไลซิ่ง ในเรื่องของโทรทัศน์ขาวดำทุกประการ

จ. สัญญาณซิงค์ของภาพสี (color sync signal) เนื่องจากสัญญาณโทรทัศน์ที่ให้ภาพสี อยู่ในรูปของ amplitude-modulated signal (suppressed carrier) ตามที่แสดงไว้ในรูปที่ 3.19 ซึ่งเมื่อเครื่องรับโทรทัศน์สีได้รับแล้ว ก็จำเป็นต้องใช้คลื่นพาห์ของภาพสี หรือคัลเลอร์ซับแคริเออร์ (color subcarrier) ที่เหมือนกันกับที่ใช้ในเครื่องส่งโทรทัศน์ด้วย ดังนั้น เครื่องรับโทรทัศน์สีจึงจำเป็นต้องมีวงจรผลิตคลื่นพาห์ของภาพสี หรือคัลเลอร์ซับแคริเออร์ที่ต้องการขึ้น เพื่อทำให้คลื่นพาห์ของภาพสีที่ใช้ในด้านเครื่องส่งโทรทัศน์กับด้านเครื่องรับโทรทัศน์สี มีความถี่และเฟสที่ถูกต้องตรงกัน เครื่องส่งโทรทัศน์จึงจำเป็นต้องส่งสัญญาณซิงค์ของภาพสี (color sync signal) ไปให้เครื่องรับโทรทัศน์สี ซึ่งเรียกสัญญาณซิงค์ของภาพสีนี้ว่า คัลเลอร์เบิร์สต์ (color burst)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 รูปร่างของสัญญาณที่ได้จากการมอดูเลทแบบ เอ.เอ็ม. โดยทิ้งสัญญาณคลื่นพาห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การเขียนภาษา VHDL

Terminology and Conventions

การเขียนรูปแบบของระบบดิจิตอลด้วยภาษา VHDL นั้น จะมีศัพท์เทคนิคเฉพาะ ฉะนั้นในส่วนนี้จะเป็นการบรรยาย และอธิบายศัพท์บางคำที่จะต้องพบในรายงานชุดนี้

- ลักษณะของรูปแบบ (model styles) : ลักษณะของการเขียนรูปแบบ (model) ด้วยภาษา VHDL สามารถแบ่งได้เป็น

- Behavioral Model : หรือที่เรียกอีกอย่างว่า algorithmic description เป็นรูปแบบที่บรรยายพฤติกรรมของระบบดิจิตอล ในส่วนที่บรรยายมีโครงสร้างคล้ายกับภาษาชั้นสูง (high level language) ทั่วๆ ไป เช่น PASCAL หรือ C เป็นต้น ในการจำลองการทำงาน (simulation) คำสั่งแต่ละคำสั่ง (statement) จะถูกประเมินผลเป็นไปตามลำดับ (sequential) จากบนลงล่าง ยกเว้นในกรณีของคำสั่ง LOOP หรือการเรียกใช้โปรแกรมย่อย รูปแบบลักษณะนี้จะไม่ให้รายละเอียดที่เกี่ยวกับการผลิต หรือโครงสร้างของ Hardware แต่ในทางตรงข้ามที่จะให้รายละเอียดเกี่ยวกับความสัมพันธ์ระหว่าง input กับ output ที่ดี

- Dataflow Model : เรียกอีกอย่างหนึ่งได้ว่า "Register Transfer Level" (RTL) เป็นรูปแบบที่ถูกเขียนขึ้น เพื่อจุดประสงค์ที่จะใช้เป็นเครื่องมือสำหรับสังเคราะห์วงจรอัตโนมัติ รูปแบบลักษณะนี้ส่วนใหญ่จะเป็น procedural constructs และ functional operators

- Structural Model : เป็นรูปแบบที่แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ต่างๆ ที่ประกอบกันขึ้นเป็นวงจรหรือระบบดิจิตอล และสามารถเรียกอีกอย่างได้ว่า "netlist representation" เป็นการเขียนที่แสดงให้เห็นโครงสร้างของ hardware

- Mixed-Level Model : จากคุณสมบัติที่อ่อนตัวของภาษา VHDL จึงสามารถที่จะเขียนรูปแบบ โดยใช้ลักษณะต่างๆ บรรยายวงจรหรือระบบดิจิตอลเดียวกันได้ ฉะนั้นรูปแบบเช่นนี้จึงมีการเขียนแบบผสม

- **Concurrency :** ในภาษา VHDL นั้น ชุดคำสั่งแต่ละชุดคำสั่งจะทำงานในเวลาเดียวกันและอิสระต่อกัน ลักษณะเช่นนี้เป็นคุณสมบัติที่เป็นความจริงทางฟิสิกส์ของวงจรอิเล็กทรอนิกส์ ชุดคำสั่งนี้เรียกว่า "concurrent stratement" และจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงค่าของสัญญาณ

- **Sequential :** นอกจากความสามารถที่ชุดคำสั่งจะทำงานแบบ concurrent แล้ว บางครั้งการเขียนรูปแบบในลักษณะที่บรรยายพฤติกรรมของวงจร มีความจำเป็นที่จะต้องให้ชุดคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สั่งทำงานเป็นลำดับขั้นเรียงกันจากบนลงล่าง อย่างเช่นการเขียนแบบ behavioral model เป็นต้น ชุดคำสั่งที่เป็น sequential นี้จะใช้ในโปรแกรมย่อย (subprogram) และ process statement

- **Driver :** สัญญาณต่างๆ (signal) ใน VHDL นั้นจะถูกควบคุมด้วยตัวขับหรือ "drever" สัญญาณเหล่านี้จะรับค่าใหม่ (ระดับของสัญญาณ) ได้ด้วยตัวขับนี้เอง
- **Transaction :** การเกิด transaction กับ signal นั้นจะเกิดขึ้นเมื่อมีการกำหนดค่าๆ หนึ่งให้กับ signal นั้น ค่าใหม่ที่ signal ได้รับอาจจะมีผลหรือไม่มีผลทำให้เกิดการเปลี่ยนแปลงของระดับสัญญาณ (event) เช่นการเปลี่ยนจากค่า Logic '0' เป็นค่า Logic '0' เป็นต้น
- **Event :** คือการเปลี่ยนระดับค่าของ SIGNAL จากระดับหนึ่งไปสู่ระดับอื่น อย่างเช่น ในระบบดิจิตอลการเปลี่ยนจาก Logic '0' เป็นค่า Logic '1' หรือในทางตรงกันข้ามถือว่า SIGNAL นั้นเกิด "event" ฉะนั้นจะเห็นได้ว่า การที่จะเกิด event ได้นั้นจะต้องเกิด transaction ด้วย แต่ในทางตรงข้ามการเกิด transaction ไม่จำเป็นต้องเกิด event ทุกครั้ง
- **Sensitivity List :** คือรายชื่อของ signal ต่างๆ ที่มีผลทำให้เกิดการทำงานของ concurrent statement เมื่อเกิด event ขึ้นกับ signal ตัวใดตัวหนึ่งหรือหลายตัวพร้อมกันในรายชื่อนั้น
- **Objects :** ในภาษา VHDL นั้นคำว่า object ใช้เขียนเพื่อบ่งบอกถึงองค์ประกอบส่วนหนึ่งของรูปแบบ ซึ่งเปรียบได้เหมือนกับภาชนะที่มีไว้สำหรับบรรจุค่าต่างๆ สามารถแบ่งออกได้เป็นสามชั้น (class) ด้วยกันคือ

- CONSTANT: ได้แก่ object ประเภทหนึ่งที่เมื่อกำหนดค่าเริ่มต้นให้แล้วจะคงค่านั้นไว้ตลอด ไม่สามารถดัดแปลงหรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศต่างๆ ของรูปแบบ (model)

- SIGNAL : หมายถึง object ประเภทหนึ่งที่สามารถกำหนดค่าที่สัมพันธ์กับเวลาให้ได้นั้น หมายความว่า SIGNAL สามารถรับค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง SIGNAL จะรับค่าๆ หนึ่งได้จากตัวขับสัญญาณหรือ driver ซึ่งตัวขับนี้อาจจะเก็บค่าในอนาคตสำหรับ SIGNAL ไว้ด้วย SIGNAL สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้น ดังนั้น SIGNAL จึงสามารถถูกนำไปใช้ตลอดโครงสร้างของรูปแบบหรือที่เรียกว่า global object

- VARIABLE : หรือตัวแปรได้แก่ object ที่สามารถกำหนดค่าใดๆ ให้ได้และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดการจำลองการทำงาน แต่จะเก็บค่าเพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง เนื่องจาก VARIABLE สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้นอันได้แก่ส่วนประกอบของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น VARIABLE จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้เท่านั้น (local object)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประเภทของ object ที่กำหนดไว้แล้ว (predefined type): ได้แก่ TYPE ที่กำหนดไว้ใน package ชื่อ STANDARD และกำหนดโดย IEEE ว่าจะต้องมีในระบบที่ใช้พัฒนา VHDL ฉะนั้นจึงไม่จำเป็นต้องประกาศใช้ในทุกรูปแบบที่เขียนขึ้น TYPE ประเภทนี้ได้แก่

1) BOOLEAN คือกลุ่มของค่า FALSE และ TURE
2) BIT คือกลุ่มของค่า '0' และ '1'
3) INTEGER คือกลุ่มของค่า –214748347 ถึง 214748347
4) REAL คือกลุ่มของค่า –1.0E38 ถึง 1.05E38
5) CHARACTER คือกลุ่มของค่าพยัญชนะ 'A' - 'Z', 'a' - 'z' อักษรหรือเครื่องหมายพิเศษ และตัวอักษรควบคุม
6) TIME ได้แก่หน่วยเวลาที่มีค่าพื้นฐานเป็นวินาที (second ย่อด้วย s หรือ S)
7) SEVERITY LEVEL คือกลุ่มของค่า NOTE, WARNING, ERROR, FAILURE

ส่วนต่างๆ ในการเขียน VHDL

การเขียนรูปแบบหรือ modeling ด้วยภาษา VHDL มีความจำเป็นที่จะต้องแนะนำให้รู้จักกับส่วนต่างๆ ของแบบ (design units) ที่ใช้ภาษาเสียก่อน และนี่ก็เป็นขั้นตอนแรกที่สำคัญที่สุดของการศึกษาเรียนรู้การใช้ภาษา VHDL เขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะ Top-Down Desing นอกจากนั้นการที่จะเข้าใจในกฎเกณฑ์ได้นั้น จะต้องทำความเข้าใจในเรื่องของโครงสร้าง และส่วนต่างๆ ของรูปแบบ VHDL ให้ถูกต้องเสียก่อน

ภาษา VHDL นั้นประกอบด้วยส่วนต่างๆ ที่สำคัญและเป็นพื้นฐานของการเขียนรูปแบบระบบดิจิตอลที่สำคัญ 4 หน่วยคือ

1. Entity Design Unit
2. Architecture Design Unit
3. Package Design Unit
4. Configuration Design Unit

1. Entity Design Unit

หน่วยของแบบ (design unit) ส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบ (model) ที่จะเขียนขึ้น ส่วนนี้เรียกว่า "entity design unit" ในส่วนนี้ใช้กำหนดจุดต่อ (connection point) ของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ (mode) และประเภทของค่า (type of value) ที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ (PORT) ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Architecture Design Unit

คือส่วนที่ใช้เขียนบรรยายกำหนดพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงาน (simulation) พฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้า – ออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ (ports and generics) ที่กำหนดใน entity design unit

ส่วนของ Architecture Design Unit นั้น เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า architecture นั้นบรรยาย entity design unit ใด ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นส่วนประกาศกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายใน architecture นั้นได้ อาทิเช่น ประเภท (type), สัญญาณ (SIGNAL), ตัวคงที่ (CONSTANT) โปรแกรมย่อยต่างๆ

3. Package Design Unit

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย (subprogram) ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนที่เรียกว่า package ได้ และข้อมูลเหล่านี้สามารถนำไปใช้ได้โดย entity design unit, architecture design unit หรือจาก package design unit อื่นๆ ด้วยชุดคำสั่ง USE statement นอกจากนั้นสิ่งที่นิยมทำกันมากคือรูปแบบ (model) ต่างๆ อาทิเช่น standard components จะถูกเก็บไว้ใน package ที่ทุกคนสามารถเข้าถึง และนำไปใช้ได้ โดยปกติ package จะแบ่งเป็น 2 ส่วน คือ package declaration และ package body

3.1 Package declaration

เป็นส่วนที่มีความสำคัญที่สุดของ package เพราะจะเป็นส่วนที่กำหนดชื่อ (identifier) ของสิ่งที่ประกาศอยู่ภายใน package สำหรับนำไปใช้ภายนอกตัวของ package เอง ถ้าสิ่งใดๆ ถูกประกาศในส่วนของ package body แต่ไม่ถูกประกาศใน package declaration จะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้จากส่วนนอกได้ ฉะนั้นโดยทั่วไปแล้ว package สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วน body และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่นใช้สำหรับประกาศ TYPE หรือ SIGNAL เช่นเดียวกันกับ package body ที่ไม่จำเป็นต้องมี Package declaration แต่ package นั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบ (model) อื่นได้

3.2 Package body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ (sequential statemment) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของ package declatation แล้ว จะถูกเก็บไว้ใน package body ทั้งนี้รวมทั้ง deferred constants

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Configuration Design Unit

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมี entity design unit ได้เพียงหน่วยเดียวเท่านั้น แต่ในขณะที่ entity design unit หนึ่งหน่วยนี้อาจจะมี architecture ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจึงเกิดคำถามขึ้นว่า ในการจำลองการทำงานของรูปแบบ (model) นั้น simulator จะนำ architecture อันไหนไปจำลอง คำตอบของคำถามนี้คือ ต้องบอกให้ simulator ทราบ และในรูปแบบ VHDL นั้นการบอกหรือกำหนดคือการใช้ CONFIGURATION ประกอบ entity กับ architecture design unit ที่ต้องการเข้าด้วยกัน

4.1 การออกแบบในลักษณะโครงสร้าง และการบรรยายพฤติกรรม (Structural Design & Behavioral Design)

ภาษา VHDL เป็นเครื่องมือสำหรับช่วยสร้างรูปแบบ (model) ของระบบดิจิตอลที่ซับซ้อน โดยอาศัยขบวนการของ Top-Down Design ขบวนการดังกล่าวคือ การบรรยายระบบดิจิตอลในระดับบนสุด (top-level) ในรูปของแนวความคิดฟังก์ชันการทำงานอย่างสังเขป เขียนรูปแบบและจำลองกรทำงาน เพื่อตรวจสอบความถูกต้อง หลังจากที่ผ่านการตรวจสอบแล้ว แนวความคิดอย่างสังเขปนี้จะถูกแบ่ง (partition) ให้เป็นส่วนย่อยๆ และลำดับชั้น (hierarchical) ตามกลุ่มของฟังก์ชันการทำงาน และเช่นเดียวกับส่วนย่อยๆ ที่สร้างขึ้นเหล่านั้น จะถูกจำลองการทำงาน (simulation) ตรวจสอบความถูกต้อง (test and verification) เป็นเช่นนี้ไปเรื่อยๆ ของวงรอบการทำงาน ระดับสุดท้ายคือระดับล่างสุด (gate-level) สามารถที่จะนำไปเปรียบเทียบกับอุปกรณ์ digital hardware ต่างๆ ได้ อาทิเช่น microprocessor, RAM, ROM, PLD และ FPGA โดยผ่านขั้นตอนของการสังเคราะห์วงจร (circuit synthesis)

ฉะนั้นการเขียนรูปแบบในลักษณะของ structural description จึงเป็นการบรรยายที่แสดงให้เห็นโครงสร้างของระบบในรูปของอุปกรณ์ต่างๆ และการเชื่อมต่อสัญญาณระหว่างกัน อุปกรณ์แต่ละตัวอาจจะถูกบรรยายพฤติกรรมในลักษณะ behavioral description สำหรับการทำงานของตัวเอง หรืออาจจะบรรยายด้วยอุปกรณ์ระดับล่างลงไปอีกเช่น gates หรือ transistor description คือ behavioral description จะเป็นตัวอย่างที่ดีของรูปแบบอย่างสังเขป (abstract model) รูปแบบในลักษณะนี้จะไม่ชี้ให้เห็นชัดว่า วงจรจะมีรูปร่างและโครงสร้างเป็นอย่างไร ส่วนรูปแบบลักษณะ structural description นั้นจะเป็นรูปแบบที่สามารถมองเห็นรูปร่างของวงจรได้ชัดเจน เช่นวงจรประกอบด้วยอุปกรณ์อะไรบ้าง และแต่ละประเภทมีจำนวนเท่าไร มีการเชื่อมต่อกัน และลำดับชั้นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้จะเป็นการขยายความของหลักการ behavioral description และ structural description พร้อมกับแสดงให้เห็นว่า โครงสร้างทั้งสองนี้มีความสำคัญอย่างไรกับวิธีการออกแบบในลักษณะของ Top-Down-Design

1. Behavioral Design

การเขียนรูปแบบลักษณะของ behavioral description ของระบบดิจิตอลด้วยภาษา VHDL นั้น ถูกจัดให้อยู่ในประเภทของการบรรยายที่ไม่ต้องมีการอ้างถึงรูปแบบย่อย (submodel) ภายใน architecture นั้นอีก ทั้งนี้จะไม่รวมถึงการเรียกโปรแกรมย่อยที่สามารถเกิดขึ้นได้เสมอในรูปแบบลักษณะนี้ แต่จะหมายถึงการอ้างถึงอุปกรณ์อื่นที่ถูกกำหนดด้วย VHDL ก่อนแล้ว

ลักษณะของ behavioral description โดยทั่วไปแล้วจะเป็นขั้นตอนของการบรรยายที่จะกำหนดฟังก์ชันการทำงานของแบบ ส่วนที่กำหนดการติดต่อกับโลกภายนอกคือ entity declaration (คำสั่ง ENTITY)

รูปแบบ VHDL ที่ใช้บรรยาย multiply accumulate function มีลักษณะคล้ายกับการเขียนโปรแกรมด้วยภาษา C หรือ PASCAL มากกว่าที่จะเป็นโครงสร้างของ hardware ดังนั้น ด้วยเหตุผลนี้เอง จึงไม่สามารถที่จะหาความสัมพันธ์ของการบรรยายในรูปของฟังก์ชันการทำงาน กับโครงสร้างทางฟิสิกส์ในรูปของอุปกรณ์ดิจิตอลได้ รูปแบบ VHDL เช่นนี้จึงจัดให้อยู่ในประเภท behavioral ข้อดีของการบรรยายแบบ behavioral ที่ว่าวิศวกรออกแบบสามารถที่จะสร้างรูปแบบของระบบดิจิตอล โดยไม่ต้องคำนึงถึงรายละเอียดของการสร้างวงจรจริง จึงสามารถที่จะเขียนรูปแบบให้เข้าสู่จุดประสงค์ของงาน (specification) โดยไม่ต้องเบี่ยงเบนความคิดว่าจะใช้อุปกรณ์อะไรมาสร้างให้ได้ฟังก์ชันตามต้องการ

2. Structural Design

การบรรยายในลักษณะโครงสร้างด้วยภาษา VHDL จัดอยู่ในประเภทการแสดงด้วยการแทนที่โดยอุปกรณ์ (ในที่นี้หมายถึงอุปกรณ์ที่อยู่ในรูปแบบของ VHDL คือ entity และ architecture design unit) และการเชื่อมต่อภายในระหว่างอุปกรณ์เหล่านั้นด้วยโครงสร้าง VHDL การบรรยายเช่นนี้จะอยู่ในรูปที่เรียกว่า VHDL netlist ในส่วนของ architecture declaration จะมีการประกาศกำหนดอุปกรณ์ด้วย component statement ส่วนที่ใช้บรรยาย architecture เองนั้นจะใช้อุปกรณ์เท่าที่ได้ประกาศกำหนดไว้แล้ว

3. Mixed Level Modeling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุผลที่ชุดคำสั่งแบบแข่งขนาน (concurrent statement) สามารถที่จะเขียนลงในตำแหน่งใดๆ ภายในส่วนของ architecture ได้นี่เอง ภาษา VHDL จึงเปิดโอกาสให้วิศวกรออกแบบค่อยๆ แปลงรูปแบบจาก behavioral model ไปสู่ structural model ได้

Concurrent statement สามารถมองให้เป็นส่วนย่อยๆ ของแบบ (subdesign) ที่เป็นอิสระในการทำงาน และจากการที่ concurrent statement ทั้งหลายเป็นอิสระต่อกัน ดังนั้นภายใน architecture เดียวกันสามารถที่จะมีชุดคำสั่งประเภท concurrent statement ได้หลายๆ ชุด จาก concurrent statement หลายๆ อันมาใช้ด้วยกันแสดงให้เห็นหลักการของการเขียนรูปแบบ (model) ในลักษณะของ mixed level modeling

4.2 ชุดคำสั่งลำดับ (Sequential Statements)

ภาษา VHDL สามารถใช้เขียนรูปแบบ (modeling) บรรยายระบบดิจิตอลในลักษณะของ behavioral description ที่โครงสร้างภายในประกอบด้วย sequential statement การศึกษาในรายละเอียดของโครงสร้างดังกล่าว สำหรับ solfware engineering ที่มีความคุ้นเคยกับการเขียนโปรแกรมด้วยภาษาชั้นสูง อาทิเช่น C หรือ PASCAL อยู่ก่อนแล้ว จะสามารถทำความเข้าใจโครงสร้างแบบ sequential ได้ง่าย เพียงแต่ต้องทำความเข้าใจเกี่ยวกับลักษณะการทำงานของ hardware เพิ่มเติม ในภาษา VHDL มีคำสั่งที่เป็น sequential statement ดังต่อไปนี้

- WAIT statement
- VARIABLE assignment
- Signal assignment
- IF-THEN-ELSE statement
- CASE statement
- Loops
- NEXT statement
- EXIT statement
- RETURN statement
- NULL statement
- Procedure call
- ASSERTION statement

จะกล่าวเฉพาะ statement ที่สำคัญๆ เพื่อความเข้าใจในการทำงานแบบ sequential เท่านั้น ตามที่เคยกล่าวมาแล้วว่าภาษา VHDL เป็นภาษาที่มีคุณสมบัติเป็นแบบแข่งขนานนั้น คือ ชุดคำสั่งภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวโครงสร้างจะเป็นชุดคำสั่งแบบแข่งขนาน เช่นเดียวกับภาษา ADA ชุดคำสั่งลำดับหรือ sequential statement ที่สร้างขึ้นจะถูกบรรจุอยู่ภายในเปลือกของส่วนที่เป็น concurrent หรือที่เรียกกันว่า "concurrent shell" อันได้แก่ process statement

1. Process Statement

หัวใจสำคัญของ concurrent shell ที่ทำให้สามารถเขียน VHDL model เพื่อบรรยายพฤติกรรมของระบบดิจิตอลอิเล็กทรอนิกส์ในลักษณะ behavioral description ได้แก่ชุดคำสั่ง process ที่โครงสร้างภายในจะประกอบด้วยชุดคำสั่งแบบลำดับเท่านั้น ชุดคำสั่งเหล่านี้จะทำงานเป็นลำดับจากบนลงล่าง เมื่อ PROCESS ถูกกระตุ้นให้ทำงาน

คำ PROCESS ในบรรทัดแรกของโครงสร้าง แสดงถึงจุดเริ่มต้นของชุดคำสั่ง process ในบางกรณีใน architecture หนึ่งอาจจะมีโครงสร้างของชุดคำสั่ง process หลายชุดได้ คำ END PROCESS บอกถึงจุดสิ้นสุดของชุดคำสั่ง process เมื่อคำสั่ง END PROCESS ถูกปฏิบัติแล้วชุดคำสั่ง process จะหยุดการทำงานลงชั่วขณะ (แต่ยังคง active อยู่ตลอดเวลา) จนกว่าจะมีสัญญาณอย่างน้อยตัวใดตัวหนึ่งใน sensitivity list เกิด event ขึ้นอีก ชุดคำสั่ง process เป็นชุดคำสั่งแบบแข่งขนาน นั้นหมายความว่า โดยปกติแล้วชุดคำสั่ง process จะทำงานตลอดเวลา การที่ชุดคำสั่งทั้งหลายที่อยู่ภายในบล็อกของชุดคำสั่ง process เป็นชุดคำสั่งแบบลำดับ นั้นถ้าชุดคำสั่ง process ใดที่ไม่มี sensitivity list เป็นตัวควบคุมการทำงาน จะทำให้เกิดการทำงานที่เปรียบเสมือนว่าเป็นวงรอบ (loop) ที่ไม่รู้จบขึ้น วิธีการที่จะป้องกันการเกิดเหตุการณ์เช่นนี้ คือการเติม wait statement ลงในส่วนของชุดคำสั่ง process

2. Wait Statement

ชุดคำสั่ง proess สามารถมี sensitivity list ได้เพียงอันเดียว หมายความว่าชุดคำสั่ง process จะถูกกระตุ้นได้จากการที่สัญญาณใดสัญญาณหนึ่งในรายชื่อที่เกิด event ขึ้นเท่านั้น หลังจากที่ถูกกระตุ้นแล้วคำสั่งทั้งหลายที่อยู่ภายในจะทำงานแบบลำดับลงมาจนกระทั่งหมด และชุดคำสั่ง process จะหยุดการทำงานชั่วคราวจนกว่าจะมี event เกิดขึ้นอีกกับสัญญาณตัวใดตัวหนึ่งในรายชื่อนั้นอีก ถ้ากรณีที่ภายในรายชื่อประกอบด้วยสัญญาณหลายตัว และเกิด event ขึ้นในเวลาเดียวกัน จะมีสัญญาณเพียงตัวเดียวจากทั้งหมดเท่านั้น ที่กระตุ้นการทำงานของชุดคำสั่ง process ซึ่งไม่สามารถที่จะบอกได้ว่าเป็นตัวใด ดังนั้นการใช้ชุดคำสั่ง process ร่วมกับ sensitivity list จึงมีขีดจำกัดอยู่มาก ในภาษา VHDL มีวิธีการหลีกเลี่ยงปัญหาเช่นนี้โดยใช้ชุดคำสั่ง wait statement

คำสั่ง wait statement นั้นให้ความคล่องตัวกว่า เพราะ wait statement มีข้อดีคือ ประการแรก สามารถที่จะกำหนดลงตรงตำแหน่งใดๆ ภายในโครงสร้างของชุดคำสั่งลำดับได้ เพื่อระงับการทำงานภายในชุดคำสั่ง process ตรงตำแหน่งที่ wait statement อยู่ ส่วนการใช้ sensitivity list

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวควบคุมการทำงานชุดคำสั่ง process จะหยุดตรงตำแหน่งสุดท้ายของชุดคำสั่ง process เท่านั้น ประการที่สอง สามารถที่จะกำหนด wait statement ได้หลายๆ อันในแต่ละตำแหน่งตามความต้องการโดยไม่จำกัด ส่วนประการสุดท้าย wait statement มีหลายรูปแบบ นั่นหมายความว่าสามารถที่จะใช้ควบคุม PROCESS ได้หลายลักษณะ แต่มีสิ่งที่ต้องจำไว้ว่า wait statement ไม่สามารถที่จะใช้ร่วมกับ sensitivity list ภายใน PROCESS เดียวกันได้ ฉะนั้นผู้เขียนรูปแบบจึงต้องตัดสินใจก่อนว่าจะใช้อะไร ทั้งนี้ขึ้นอยู่กับระบบดิจิตอลที่จะเขียนบรรยายและประสบการณ์ในการเขียน

ในภาษา VHDL สามารถใช้ wait statement ได้ 4 แบบคือ

- WAIT ON signal_list; -- signal sensitivity
- WAIT UNTIL condition; -- condition
- WAIT FOR time; -- timeout
- WAIT; --forever

แต่ละอย่างมีความหมายและวิธีใช้ที่แตกต่างกันตามตัวอย่างต่อไปนี้

- **WAIT ON clock, clear, preset, d;**

คำสั่งนี้จะหยุดการทำงานของชุดคำสั่งลำดับ ไว้จนกว่าจะเกิด event ขึ้นที่สัญญาณ clock หรือ clear หรือ preset หรือ d

- **WAIT UNTIL (clock = "1");**

การทำงานของชุดคำสั่งลำดับจะหยุดที่ตำแหน่งนี้ และจะทำงานต่อไปเมื่อสัญญาณ clock เกิด event และ boolean expression เป็น TRUE ทุกครั้งเมื่อลำดับการทำงานถูกหยุดตรงจุดนี้ และก่อนที่จะทำต่อไปได้นั้นต้องตรวจสอบว่ามี event เกิดขึ้นบนสัญญาณ clock หรือเปล่า และสัญญาณ clock มีค่าเป็น "1" ฉะนั้น boolean expression จะต้องเป็นจริง

- **WAIT FOR 10 NS;**

คำสั่งนี้เป็นการรอหรือหยุดในขณะที่จำลองการทำงาน (simulation) จนกว่าเวลาการจำลองจะล่วงเลยไปแล้ว 10 ns. จะมีผลทำให้ PROCESS เริ่มทำงานต่อไปได้

- **WAIT;**

เป็นคำสั่งให้ PROCESS หยุดทำงานตลอดไป โดยไม่มีการเริ่มต้นการทำงานใหม่ บางครั้งการเขียนรูปแบบบรรยายการทำงานของระบบดิจิตอล มีความจำเป็นที่ต้องใช้คำสั่งนี้ เพื่อหยุดการทำงานของ PROCESS อย่างถาวร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

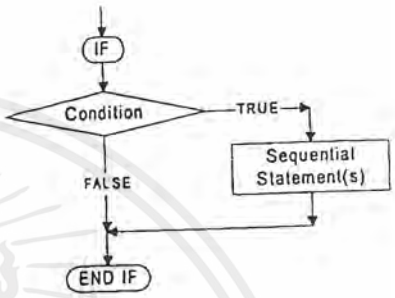
นอกจากนั้นยังสามารถใช้ควบคุม (ON, UNTIL และ FOR) ร่วมกันใน wait statement ได้ ส่วนใหญ่ของ VHDL analyzer จะตรวจสอบเพื่อให้เป็นที่แน่ใจว่าใน PROCESS มี sensitivity list หรือ wait statement อย่างใดอย่างหนึ่ง เพื่อป้องกันปัญหาอันอาจจะเกิดขึ้นได้

3. IF-THEN-ELSE statement

เป็นชุดคำสั่งพื้นฐานที่ใช้สอบถาม เพื่อการตัดสินใจกระทำอะไรบางสิ่งบางอย่าง

IF statement

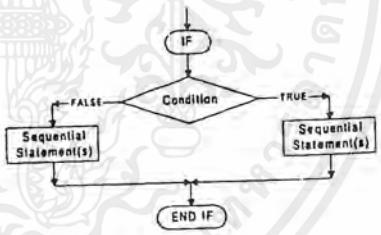
```
Format: IF condition THEN
            sequential_statement(s)
        END IF;
```




```
ตัวอย่าง:   IF a = '1' THEN
              count := count + 1;
            END IF;
```

IF-ELSE construct

```
Format: IF condition THEN
            sequential_statement(s)
        ELSE
            sequential_statement(s)
        END IF;
```

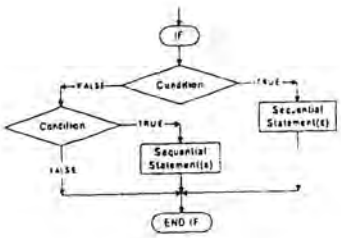



```
ตัวอย่าง:   IF a = '1' THEN
              one_count := count + 1;
            ELSE
              two_count := count + 2;
            END IF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IF - ELSIF construct

Format: IF condition THEN
sequential_statement(s)
ELSIF condition THEN
sequential_statement(s)
END IF;



ตัวอย่าง:
```
IF a = 'X'    THEN
   b := b + 1;
ELSIF a = '0'   THEN
   c := c + 1;
ELSIF a = '1'   THEN
   d := d+1;
ELSIF a = 'Z'   THEN
   e := e + 1;
E   N   D     I   F   ;
```

4. CASE Statement

โครงสร้างอีกอันหนึ่งที่เป็นลำดับ และมีความคล้ายคลึงกับ IF-THEN-ELSE คือ CASE statement เพราะใช้เป็นคำสั่งเลือกหนทางปฏิบัติ ตามข้อแม้ (condition) ที่กำหนดให้

คำสั่ง CASE และ END CASE กำหนดจุดเริ่มต้นและจุดสิ้นสุด คำสั่ง WHEN ใช้สำหรับกำหนดตัวเลือกที่จะนำมาเปรียบเทียบกับ expression ตัวเลือกใดเป็นไปตาม expression ที่กำหนด PROCESS จะเริ่มต้นทำงานที่ชุดคำสั่งลำดับที่ตามมาจนกระทั่งคำสั่งสุดท้ายของตัวเลือกนั้นๆ และจะออกจาก CASE statement โดยไม่ทำหนทางเลือกอื่นๆ ที่ยังคงเหลืออยู่

5. LOOP Statement

คือการทำงานในลักษณะที่เป็นวงรอบ (loop) สามารถสร้าง loop ได้ 3 ลักษณะ คือ LOOP ธรรมดา WHILE-LOOP และ FOR-LOOP

FOR-LOOP:

กลุ่มคำสั่งลำดับใน FOR-LOOP จะทำงานเป็นวงรอบตราบเท่าที่ค่าของพารามิเตอร์ยังอยู่ในของข่าย (range) ที่กำหนด ตัว loop_parameter เป็นพารามิเตอร์ที่แฝงอยู่ในโครงสร้าง ที่ไม่ต้องประกาศหรือกำหนดขึ้น

WHILE-LOOP:

กลุ่มคำสั่งลำดับใน WHILE-LOOP จะทำงานเป็นวงรอบไปเรื่อยๆ จนตราบเท่าที่ข้อแม้ของ boolean expression ใน WHILE-LOOP ยังคงมีค่าเป็น TRUE จำนวนครั้ง (รอบ) ที่ทำจะถูกควบคุมจากภายในตัวของ loop เอง นั่นคือการจะออกจากวงรอบประเภทนี้ได้ ก็ต่อเมื่อผลลัพธ์ของ boolean expression ได้ค่าเป็น FALSE ข้อแม้ที่ใช้ควบคุมการทำงานนี้ จะถูกตรวจสอบทุกครั้งก่อนการทำงานต่อไปภายในวงรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ชุดคำสั่งแบบแข่งขนาน (Concurrent Statement)

ภาษา VHDL เป็นภาษาที่มีการทำงานในลักษณะแข่งขนาน concurrency หรือสามารถที่จะมองชุดคำสั่งแบบแข่งขนานแต่ละอันเป็น PROCESS ที่เชื่อมต่อกันด้วย signal แต่ละ PROCESS ทำงานอิสระไม่ขึ้นต่อกัน ที่เรียกว่า asynchronous ดังนั้น concurrent statement ส่วนใหญ่จึงสามารถเขียนแทนได้ด้วย PROCESS statement

ชุดคำสั่งแบบ sequential ไม่สามารถที่จะใช้ในรูปของชุดคำสั่งแบบแข่งขนานได้ มีบางคำที่สามารถใช้ใน VHDL ได้ทั้งสองรูปแบบ เช่น signal assignment เป็นต้น ชุดคำสั่งที่ใช้ในโครงสร้างแบบ concurrent ซึ่งมีทั้งหมดได้แก่

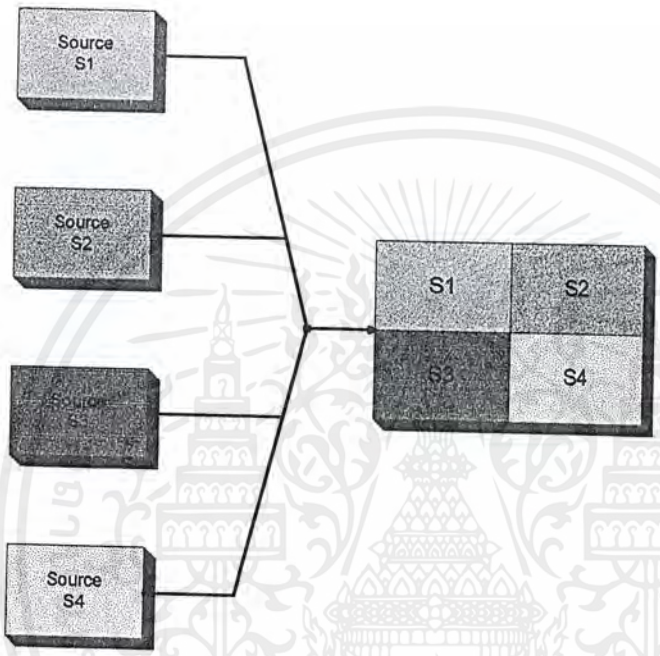
1) Signal assignment statement
2) Component instantiation statement
3) Assert statement
4) Generate statement
5) Process statement
6) Procedure statement
7) Block statement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 หลักการออกแบบ

องค์ประกอบและการควบคุมการทำงาน

เป้าหมายของโครงงานชิ้นนี้

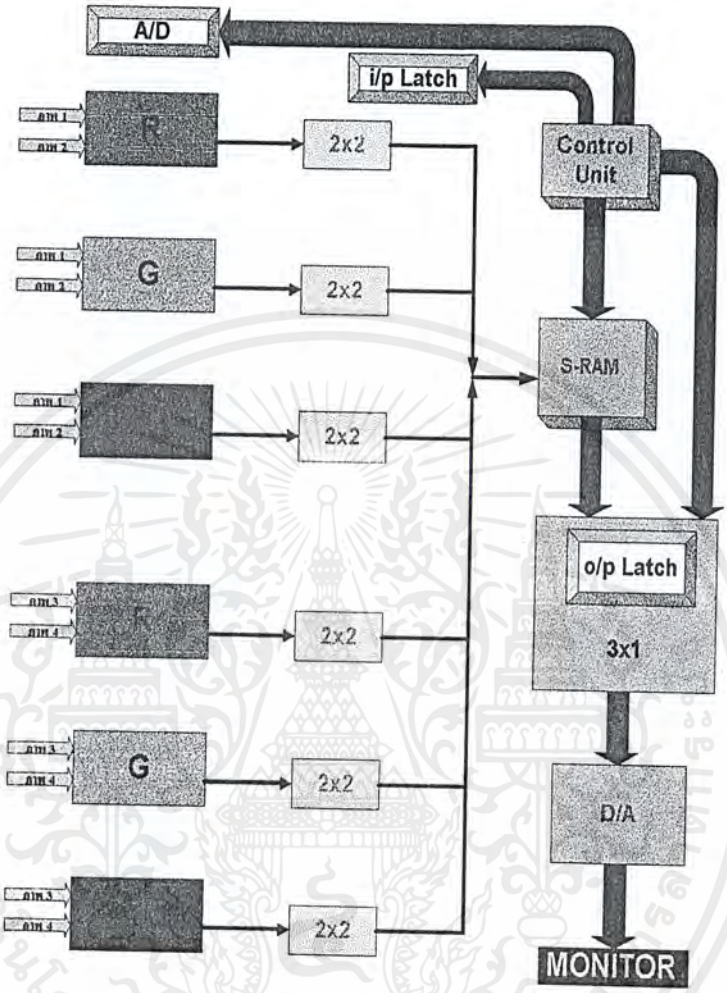


รูปที่ 5.1 การนำสัญญาณมาแสดงผลบนหน้าจอเดียวกัน

การใช้ FPGA ควบคุมการจัดการกับสัญญาณภาพโดยในโครงงานนี้ จะนำสัญญาณภาพจาก 4 แหล่ง อาจเป็นสัญญาณจากกล้องวงจรปิด หรือ สัญญาณวีดีโอก็ได้ แต่ในโครงงานนี้จะใช้สัญญาณจาก computer สัญญาณจากแหล่งจ่ายทั้ง 4 แหล่งจะถูกส่งไปแสดงผลบนหน้าจอเดียวกัน คือ แหล่งจ่ายที่1 (S1) นำสัญญาณไปแสดงผลที่หน้าจอทางด้านซ้ายซีกบนของหน้าจอ แหล่งจ่ายที่2 (S2) นำสัญญาณไปแสดงผลที่หน้าจอทางด้านขวาซีกบนของหน้าจอ แหล่งจ่ายที่ 3 (S3) นำสัญญาณไปแสดงผลที่หน้าจอทางด้านซ้ายซีกล่างของหน้าจอ และแหล่งจ่ายที่ 4 (S4) นำสัญญาณไปแสดงผลที่หน้าจอทางด้านขวาซีกล่างของหน้าจอ การนำสัญญาณทั้ง 4 แหล่งจ่ายมาแสดงผลบนหน้าจอเดียวกัน แสดงดังรูปที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบของโครงงาน

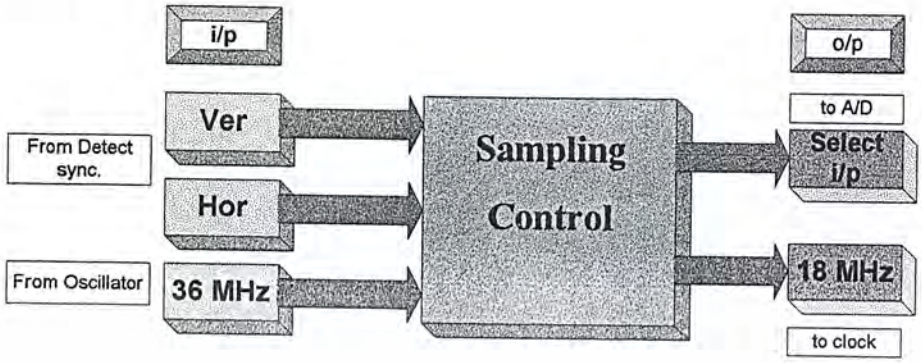


รูปที่ 5.2 การทำงานของโครงงาน

เริ่มจากสัญญาณภาพทั้ง 4 ภาพซึ่งผ่าน A/D เพื่อแปลงจาก analog เป็นสัญญาณ digital แล้วจะแบ่งแยกเป็น 2 ส่วนคือส่วนแรกจะประกอบด้วยภาพที่ 1 และภาพที่ 2 ส่วนที่สองจะประกอบด้วยภาพที่ 3 และ ภาพที่ 4 ซึ่งทั้งสองส่วนจะแบ่งออกเป็นสัญญาณสี 3 สี คือ สีแดง (R) สีเขียว (G) สีน้ำเงิน (B) ซึ่งภาพทั้งสองส่วนจะถูกส่งไปพักข้อมูลที่ Latch input ซึ่งใช้ Latch 24 ชุด คือ ในแต่ละภาพแต่ละสี จะใช้ Latch 1 ชุด และสำรอง 1 ชุด หลังจากนั้นข้อมูลจะถูกส่งไปเก็บที่ RAM ก่อนที่ข้อมูลจะถูกส่งไปแสงผลที่จอ จะนำไปพักข้อมูลที่ Latch output ซึ่งใช้ Latch 3 ชุด คือ แต่ละสีจะใช้ Latch 1 ชุด สุดท้ายจะส่งข้อมูลผ่าน D/A เพื่อเปลี่ยนเป็นสัญญาณ analog ก่อนที่จะแสดงผลบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุม Video input Selection



รูปที่ 5.3 การควบคุมการ Sampling สัญญาณ input

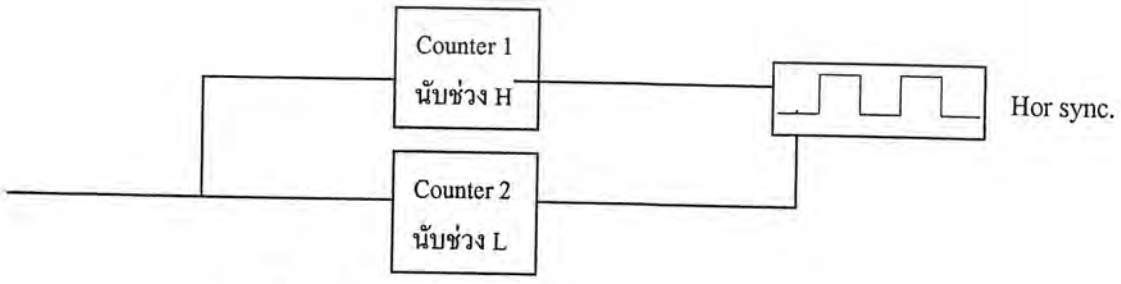
ปัจจัยที่มีผลต่อการเลือก input

- Vertical Synchronize
- Horizontal Synchronize
- Clock

Vertical Synchronize โดยจะยอมเสียเวลาไป 1 Vertical Synchronize เพื่อทำการจำช่วง high ของ Synchronize ว่าต้องใช้ clock กี่ลูกเพื่อป้องกันเวลาเกิด noise หรือเวลามี overshoot ของสัญญาณ

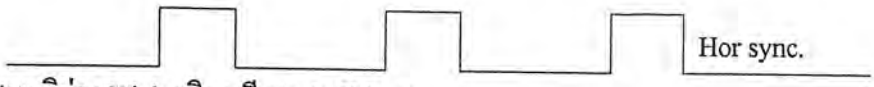
วิธี detect Horizontal Synchronize

1. จับระดับ High/Low
2. เมื่อมีการเปลี่ยนของสัญญาณ ให้ทำการเริ่มนับ clock ไปเรื่อยจนกว่าจะพบการเปลี่ยนของสัญญาณ โดยถ้าการเปลี่ยนของสัญญาณนั้นมีค่าน้อยกว่า 2 ลูก ของ clock 80 MHz ก็ให้ทำการนับต่อไป แต่ถ้าการเปลี่ยนของสัญญาณมีค่ามากกว่า ให้ทำการ reset counter ตัวหนึ่งแต่ counter อีกตัวให้ทำการนับต่อไป



รูปที่ 5.4 Horizontal Synchronize counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สมมุติช่วง High จริงๆ มี clock 100 ลูก

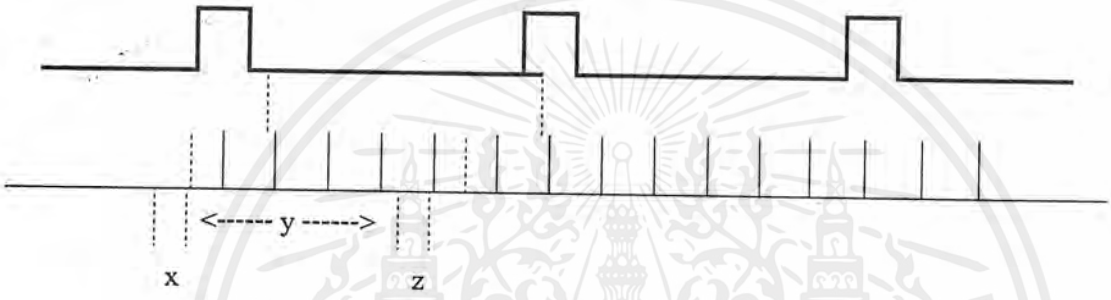
จาก High ไป Low counter 1 จะนับได้ 102

ดังนั้นจะต้องนำค่า ที่ counter 1 นับได้ลบไป 2 เท่ากับ 100 จึงเป็นจำนวน clock ที่แท้จริง

สมมุติช่วง Low จริงๆ มี clock 200 ลูก

จาก Low ไป High counter 2 จะนับได้ 202

ดังนั้นจะต้องนำค่า ที่ counter 2 นับได้ลบไป 2 เท่ากับ 200 จึงเป็นจำนวน clock ที่แท้จริง



รูปที่ 5.5 Sampling Horizontal wave

ช่วง x นับจำนวน clock เมื่อ Vertical Synchronize เปลี่ยนจาก High ไป Low ไปจนพบ Horizontal Synchronize ลูกแรกไว้ สมมุติได้เท่ากับ A

ช่วง y นับจำนวน Horizontal Synchronize ทั้งหมดที่ครบคาบ x (จำนวน clock ช่วง Low บวกจำนวน clock ช่วง High)

ช่วง z นับจำนวน clock ช่วง High บวกจำนวน clock ช่วง Low ที่ไม่ครบคาบไปจนเจอ Vertical Synchronize ลูกใหม่ จำนวน clock ในช่วง Vertical Synchronize เท่ากับ x + y +z

หมายเหตุ ให้ทำการเช็คสถานะของ Vertical Synchronize อยู่ด้วยตลอดเวลาจาก Low ไป High

การควบคุม input selector

สิ่งที่ควรรู้การจะเลือก input selector ต้องรู้ว่า input selector มีสัญญาณมาหรือไม่

สิ่งที่แสดงว่า input selector พร้อม

- Vertical Synchronize ให้มี counter เช็ค Vertical Synchronize ที่ต้องการว่าเป็น 0 หรือ 1 เพื่อการ scan Vertical Synchronize เว้น Vertical Synchronize

1) ถ้าพบ Vertical Synchronize ที่ต้อง sample นับเป็น 0 หรือ 1 แล้วแต่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เช็คสถานะ Vertical Synchronize ว่า Vertical Synchronize เข้ามาหรือยัง (เป็น 0 หรือ 1) ใช้ตอนเริ่มเท่านั้น

- ทาง Horizontal Synchronize ให้มี counter นับว่าเป็น Horizontal Synchronize ลูกที่เท่าไร
  1) ให้เช็คเมื่อสิ้นสุด Horizontal Synchronize pulse เพื่อให้กำหนดช่วงการsample
  2) ให้เช็คเพื่อรับรู้สถานะ high หรือ low ของ Horizontal Synchronize pulse เพื่อให้รู้ช่วงเวลา x และ y ตลอดเวลา

ทำการกำหนดให้แต่ละ input มีcode เฉพาะ เช่น 00, 01,10,11เช่น

- เมื่อเลือก input 1 มี code 00

เมื่อ Vertical Synchronize ลูกแรกพร้อม จะ set FF1 เป็น 1

เมื่อ Horizontal Synchronize ลูกแรกพร้อม จะ set FF2 เป็น 1

เลือก input clock เว้น clock (clock ที่ใช้คุมการเลือก input) โดยมีเงื่อนไขคือ input ต้องมารอก่อน clock sample อย่างน้อย 2 ns

หมายเหตุ ให้ input มารอก่อนที่จะทำการ sample

- เมื่อเลือก input 2 มี code 01

เมื่อ Vertical Synchronize ของ input 2 พร้อม จะ set FF3 เป็น 1

เมื่อ Horizontal Synchronize ของ input 2 พร้อม จะ set FF4 เป็น 1

เลือก input clock เว้น clock (clock ที่ใช้คุมการเลือก input) โดยมีเงื่อนไขคือ input ต้องมารอก่อน clock sample อย่างน้อย 2 ns

Latch Control

Output ของ latch มี 3 สถานะ คือ High, Low, High impedance

นำสัญญาณ Output ที่ได้จาก Analog to Digital Converter (A/D) ไปเก็บไว้ใน Latch

- นำค่าที่ได้จากการควบคุม input selector มาเก็บไว้ใน FF ก่อน เพราะว่า Output ออกจาก A/D มีการ Delay จะได้รู้ว่า Output เป็นของ input ตัวไหนแล้วทำการเก็บ
- นับจำนวน clock ที่ sample เพื่อจะได้รู้เวลาที่จะถ่ายข้อมูลจาก Latch ไป RAM
- การถ่ายข้อมูลจาก Latch ไป RAM โดย Latch แต่ละตัวจะมี code ของมัน ใน Latch แต่ละตัวที่ทำการเก็บนั้นจะต้องรู้ว่าเป็นข้อมูลของเส้นสแกนที่เท่าไร sample ที่เท่าไรในเส้น scan นั้น เพื่อดูว่า Latch แต่ละตัวจะต้องเก็บใน Address ใด

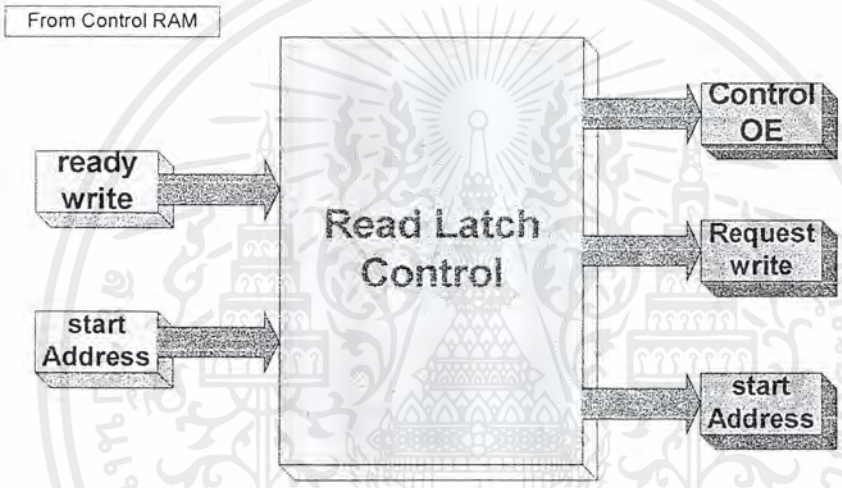
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control RAM

หลักการพิจารณาการเก็บข้อมูลว่า จะทำการเก็บใน Address ใด

- จะต้องรู้ว่าเส้น Horizontal Synchronize เส้นที่เท่าไร
- ต้องรู้ว่าเป็นการ sample ครั้งที่เท่าใดในเส้น scan นั้น
- จัดพื้นที่ RAM โดยใช้จำนวน Sample เป็นตัวพิจารณา

การถ่ายข้อมูลจากLatch เข้าสู่ RAM

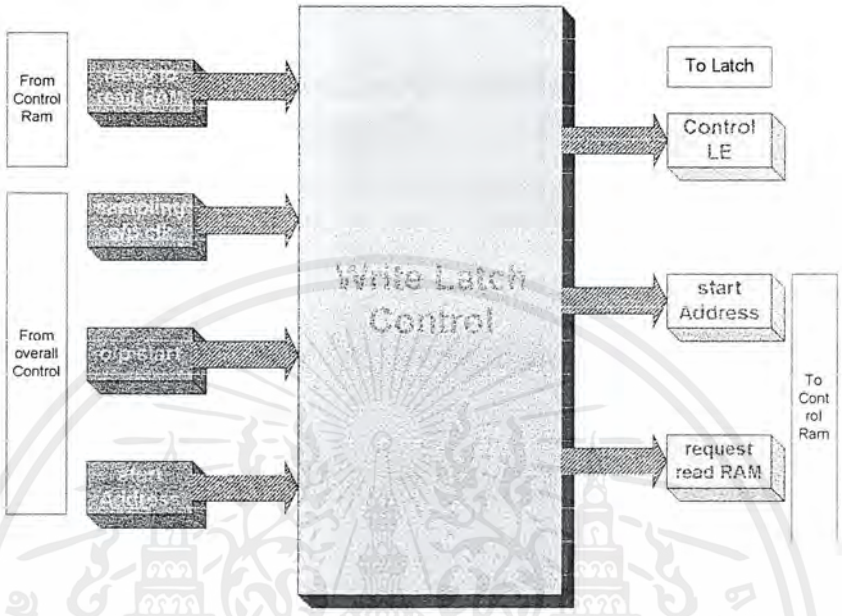


รูปที่ 5.6 การควบคุมการ Read จาก Latch เข้าสู่ RAM

ต้องมี Address counter แต่ละ input เพื่อที่จะทำการเก็บข้อมูลที่ได้ทำการ sample ได้ถูกต้องตามแต่ละสัญญาณ input โดยช่วงใดที่ไม่ได้ sample ก็ให้เว้นช่วง address นั้นด้วยโดยใช้จำนวนเคาะการ sample เป็นตัวกำหนด โดยการใช้ counter สำหรับ latch ของแต่ละ input เพื่อจะได้กำหนด address ถูกต้องในการเก็บข้อมูลแต่ละ input ลงใน RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถ่ายข้อมูลจาก RAM เข้าสู่ Latch



รูปที่ 5.7 การควบคุมการ Read ออกจาก RAM เข้าสู่ Latch

ใช้การสร้าง Synchronize เป็นตัวกำหนด Address counter แต่จังหวะในการ Read จาก RAM (ช่วงที่มี Synchronize และ Synchronize เป็น 0) ให้ output Latch เป็น Z แล้วป้อน 0 ให้ D/A ในช่องนั้นโดยใช้ counter นับจำนวน clock ที่ sampling ออกเพื่อจะได้รู้ว่าถึงเวลาที่จะดึงข้อมูลจาก Latch

หลักการในการสร้างสัญญาณ Synchronize

ต้องสัมพันธ์กับการอ่าน-การเขียน RAM ไม่ให้เวลา Read-Write ซ้อนทับกัน (ในการใช้คำสั่ง Device จะใช้ร่วมกับ Auto precharge)

Control Latch

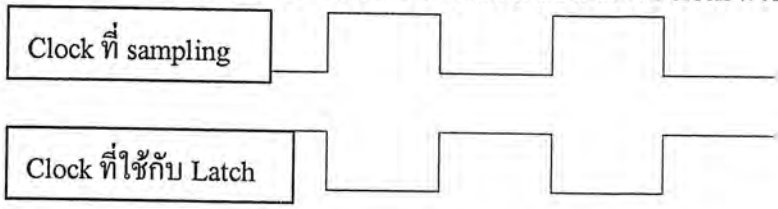
ลักษณะของ Latch ที่ใช้

1. ต้องมีขา LE (Latch Enable) ในการ control ว่าจะทำการรับ input หรือไม่
2. output ต้องเป็นแบบ Tri-state

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

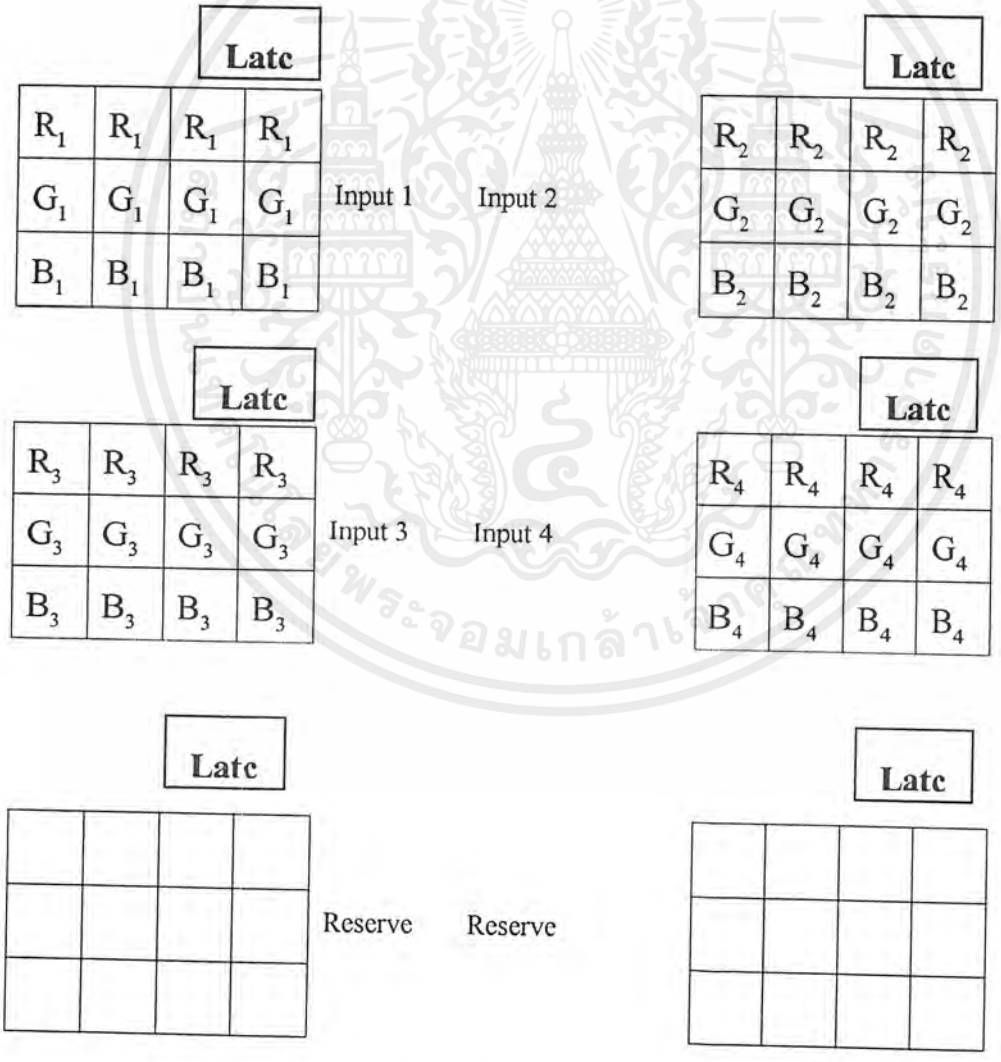
การ Control Latch

1. ใช้ clock ความถี่ 10 MHz โดยมีลักษณะเทียบกับ clock ที่ใช้ sampling ดังรูป



รูปที่ 5.8 Control clock A/D & Latch

2. กำหนดรหัสของ Latch แต่ละชุดดังนี้ โดยการกำหนดจะแยกเป็น 2 ชุด ที่อิสระต่อกันชุดละ 8 ตัว ดังรูป (Latch ที่ใช้จะเป็นขนาด 24 bit โดย R=8, G=8, B=8)



รูปที่ 5.11 การแบ่งแยกกลุ่ม Latch และ Latch สำรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โดยมีวงจรควบคุมขา Latch Enable input ดังนี้

หมายเหตุ ต้องทำการเช็คค่า delay time ของ decoder ของ AND gate ที่นำมาใช้ต้องมีค่าน้อยมากๆ

วิธีการควบคุม Decoder (A0,A1)

ส่วนที่มีอิทธิพลต่อขา A0, A1 ว่าจะทำการเลือก Latch ตัวใด

- Vertical Synchronize
- Horizontal Synchronize
- จังหวะ clock

การใช้ Synchronize ควบคุมการ Multiplex สัญญาณ

1. กำหนดรหัสของแต่ละช่องสัญญาณ เช่น 00, 01, 10, 11
2. ทำการ Detect Vertical Synchronize เป็น Pulse 4-5 ลูกติดกัน

   - Detect Horizontal Synchronize หลังจาก Detect Vertical Synchronize สำเร็จแล้วเมื่อ Horizontal Synchronize เปลี่ยน High เป็น Low, Low เป็น High แล้วให้เริ่มทำการ sampling ทันที
   - แล้วทำการนับ clock ในขณะ sample เทียบกับ clock ที่เก็บไว้ใน Register และจับการเปลี่ยนแปลง Horizontal Synchronize ด้วยแล้วนำมาเทียบกัน โดยจำนวนข้อมูลในช่วงระหว่าง Synchronize แต่ละช่วงมีค่าตายตัว
3. การ Multiplex ทำโดย check ช่องสัญญาณที่จะ Multiplex กันว่าพร้อมหรือไม่ สามารถเกิดได้คือ ถ้าช่องใดช่องหนึ่งพร้อมแต่อีกช่องไม่พร้อมก็ให้ทำการ Sample ช่องที่พร้อม ส่วนช่องที่ยังไม่พร้อมกำหนดให้ข้อมูลเป็น 0 ทั้งหมดหรือเป็น 1 ทั้งหมด

การจัดการกับข้อมูลที่ sample มาได้

- เมื่อ sample ได้ข้อมูลแล้วให้นำข้อมูลที่ได้ไปเก็บใน Buffer ก่อน
- เมื่อเก็บข้อมูลได้ครบ 4 ชุดแล้วให้ทำการขนย้ายไปยัง S-RAM โดยขนตามลำดับการ sample ก่อนหลัง
- ถ้าต้องการ Refresh ก็ให้ทำตามที่ RAM กำลังว่าง ไม่ได้ทำอะไรอยู่

การอ่านข้อมูลจาก RAM ไปใช้ใน DAC

- ทำการอ่านข้อมูลแบบ Barst ไปยัง Buffer ก่อน (การใช้คำสั่ง Read จะเสียเวลาไป 2 clock)
- เมื่อ DAC รับข้อมูลจาก Buffer ไปแล้ว จนถึงตอนกำลัง หรือหลังจากนำข้อมูลไปทั้งหมดแล้ว Buffer ก็ให้เริ่มปล่อยคำสั่ง Read ไปที่ RAM พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดเก็บข้อมูลของ RAM

- มี Source ทั้งหมด 4 source และต้องการแสดงออกที่หน้าจอ out ดังนี้
- ต้องรู้ว่าใน 1 เส้น scan ต้องใช้ RAM กี่ Address (กี่ sample)
- ทำการแบ่งให้เป็นที่เก็บ sample ให้ตายตัว (Address ของการ sample ตายตัว) โดยการจัดเก็บจะมีลักษณะดังนี้

หมายเหตุ กำหนดให้ 1 Block แทนจำนวน sample ครึ่งเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 การทดลอง

```
ENTITY  gsyn  IS
PORT   ( clk    :       IN      BIT;
         nout   :       OUT     INTEGER         RANGE 0         TO      1000000;
                nnout   :       OUT     INTEGER         RANGE 0         TO      1500);
END gsyn;

ARCHITECTURE  be  OF gsyn  IS
BEGIN
        PROCESS(clk)
                VARIABLE        n       :       INTEGER         RANGE 0 TO 1000000;
                VARIABLE        nn      :       INTEGER         RANGE 0 TO 1500;
        BEGIN
                IF      (clk'EVENT      AND clk         = '1')  THEN
                        n       :=      n+1;
                        nn      :=      nn+1;
                END IF;
                IF      n       =       943170  THEN
                        n       :=      0;
                END IF;
                IF      nn      =       1490    THEN
                        nn      :=      0;
                END IF;
                nout    <=      n;
                nnout   <=      nn;
        END PROCESS;
END be;

        ..........

ENTITY   gvsyn IS
PORT   ( nin    :       IN      INTEGER         RANGE 0         TO      1000000;
         vsyn   :       OUT     BIT);
END gvsyn;

ARCHITECTURE            be      OF      gvsyn   IS
BEGIN
        PROCESS(nin)
                VARIABLE        vcheck  :       INTEGER         RANGE 0 TO 1;
        BEGIN
                IF      (nin > 0  AND   nin < 4480)     THEN
                        vcheck  :=      1;
                END IF;
                IF      (nin >= 4484 AND nin < 943170)  THEN
                        vcheck  :=      0;
                END IF;
                IF      nin     =       0       THEN
                        vcheck  :=      1;
                END IF;
                IF      vcheck  =       1       THEN
                        vsyn    <=      '1';
                ELSE    vsyn    <=      '0';
                END IF;
        END PROCESS;
END be;

        ..........
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY   ghsyn        IS
PORT  ( nnin   :      IN     INTEGER      RANGE 0      TO     1500;
        hsyn   :      OUT    BIT);
END ghsyn;

ARCHITECTURE      be      OF      ghsyn   IS
BEGIN
      PROCESS(nnin)
            VARIABLE      hcheck  :       INTEGER      RANGE 0 TO 1;
      BEGIN
            IF      (nnin > 0       AND     nnin < 91)      THEN
                    hcheck  :=      1;
            END IF;
            IF      (nnin >= 91 AND nnin < 1490)    THEN
                    hcheck  :=      0;
            END IF;
            IF      nnin    =       0       THEN
                    hcheck  :=      1;
            END IF;
            IF      hcheck  =       1       THEN
                    hsyn    <=      '1';
            ELSE    hsyn    <=      '0';
            END IF;
      END PROCESS;
END be;
```

.....

```
ENTITY      gagc    IS
PORT  ( nnin   :    IN     INTEGER      RANGE 0      TO     1500;
        hsyn   :    OUT    BIT);
END gagc;

ARCHITECTURE      be      OF      gagc    IS
BEGIN
      PROCESS(nnin)
            VARIABLE      hcheck  :       INTEGER      RANGE 0 TO 1;
      BEGIN
            IF      (nnin > 0       AND     nnin < 181)     THEN
                    hcheck  :=      1;
            END IF;
            IF      (nnin >= 181 AND nnin < 1490)   THEN
                    hcheck  :=      0;
            END IF;
            IF      nnin    =       0       THEN
                    hcheck  :=      1;
            END IF;
            IF      hcheck  =       1       THEN
                    hsyn    <=      '1';
            ELSE    hsyn    <=      '0';
            END IF;
      END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY testsamout       IS
PORT  (         vsyn    :       IN BIT;
                hsyn    :       IN BIT;
                clk     :       IN BIT;
                innv    :       OUT     INTEGER         RANGE 0 TO 1000;
                p       :       OUT BIT);
END testsamout;

ARCHITECTURE        be      OF      testsamout      IS
BEGIN
        PROCESS(hsyn,vsyn,clk)
                VARIABLE    vcheck  :       INTEGER     RANGE  0 TO 1;
                VARIABLE    hcheck  :       INTEGER     RANGE  0 TO 1;
                VARIABLE    vpass   :       INTEGER     RANGE  0 TO 1;
                VARIABLE    hpass   :       INTEGER     RANGE  0 TO 1;
                VARIABLE    nh      :       INTEGER     RANGE  0 TO 1000;
                VARIABLE    hdelay  :       INTEGER     RANGE  0 TO 1500;
                VARIABLE    np      :       BIT;
        BEGIN
                IF      (vsyn = '0')        THEN
                        vcheck  :=      1;
                END IF;
                IF      (hsyn'EVENT AND hsyn = '1')     THEN
                        IF      vcheck  =       1       THEN
                                nh      :=      nh+1;
                        END IF;
                END IF;
                IF      (nh     >=      27 AND nh <= 627)       THEN
                        vpass   :=      1;
                ELSE    vpass   :=      0;
                END IF;

                IF      (hsyn = '0')        THEN
                        hcheck  :=      1;
                END IF;
                IF      (clk'EVENT AND clk = '1')       THEN
                        IF      vpass   =       1       THEN
                                IF      hcheck  =       1       THEN
                                        hdelay  :=      hdelay+1;
                                END IF;
                        END IF;
                END IF;
                IF      (hdelay >=      216     AND hdelay <= 1354)     THEN
                        hpass   :=      1;
                ELSE    hpass   :=      0;
                END IF;

                IF      (vsyn = '1')        THEN
                        vcheck  :=      0;
                        nh      :=      0;
                END IF;
                IF      (hsyn = '1')        THEN
                        hcheck  :=      0;
                        hdelay  :=      0;
                END IF;

                IF      hpass   =       1       THEN
                        np      :=      '1';
                ELSE    np      :=      '0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                END IF;
                p       <=      np;
                innv    <=      nh;
        END PROCESS;
END be;

                ..........

ENTITY          potfield IS
PORT    (       p       :       IN      BIT;
                adclk   :       IN      BIT;
                hsyn    :       IN      BIT;
                pot     :       OUT     INTEGER         RANGE 0 TO 300);
END     potfield;

ARCHITECTURE            be      OF      potfield IS
BEGIN
        PROCESS(p,adclk,hsyn)
                VARIABLE        npot    :       INTEGER         RANGE 0 TO 300;
        BEGIN
                IF      (adclk'EVENT AND adclk = '1')   THEN
                        IF      p       =       '1'     THEN
                                npot    :=      npot+1;
                        END     IF;
                END IF;

                IF      (hsyn = '1')    THEN
                        npot    :=      0;
                END IF;
                pot     <=      npot;
        END     PROCESS;
END     be;

                ..........

ENTITY          adlclk   IS
PORT    (       clk     :       IN BIT;
                nout    :       OUT INTEGER RANGE 0 TO 5);
END adlclk;

ARCHITECTURE            be      OF      adlclk  IS
BEGIN
        PROCESS(clk)
                VARIABLE        n       :       INTEGER RANGE 0 TO 5;
        BEGIN
                IF      ( clk'EVENT     AND clk = '1' )   THEN
                        n       :=      n+1;
                END IF;
                IF      n       =       4       THEN
                        n       :=      0;
                END IF;
                nout    <=      n;
        END PROCESS;
END be;

                ..........
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY alclk    IS
PORT    (       nin     :       IN      INTEGER         RANGE 0 TO    4;
                adclk   :       OUT     BIT;
                sam     :       OUT     BIT;
                oeout   :       OUT     BIT);
END alclk;

ARCHITECTURE        be      OF      alclk    IS
BEGIN
PROCESS (nin)
BEGIN
        CASE    nin     IS
        WHEN 0 => sam <= '0';
                        ADCLK       <=      '0';
                        oeout       <=      '1';
        WHEN 1 => sam <= '1';
                        ADCLK       <=      '0';
                        oeout       <=      '0';
        WHEN 2 => sam <= '1';
                        ADCLK       <=      '1';
                        oeout       <=      '0';
        WHEN 3 => sam <= '0';
                        ADCLK       <=      '1';
                        oeout       <=      '1';
  WHEN 4 => sam <= '0';
                        ADCLK       <=      '0';
                        oeout       <=      '1';
        WHEN OTHERS => sam <= '0';
                        ADCLK       <=      '1';
                        Oeout <=    '1';
  END CASE;
END PROCESS;
end be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      checkreadram    IS
PORT  (     p1      :       IN      BIT;
            p2      :       IN      BIT;
            p3      :       IN      BIT;
            p4      :       IN      BIT;
            vsyn    :       IN      BIT;
            pass1   :       OUT     BIT;
            pass2   :       OUT     BIT;
            pass3   :       OUT     BIT;
            pass4   :       OUT     BIT);
END   checkreadram;

ARCHITECTURE        be      OF      checkreadram    IS
BEGIN
      PROCESS(p1,p2,p3,p4,vsyn)
            VARIABLE    mp1     :       BIT;
            VARIABLE    mp2     :       BIT;
            VARIABLE    mp3     :       BIT;
            VARIABLE    mp4     :       BIT;
      BEGIN
            IF      (vsyn'EVENT AND vsyn = '1')     THEN
                  IF    p1      =       '1'     THEN
                        mp1     :=      '1';
                  ELSE  mp1     :=      '0';
                  END   IF;
                  IF    p2      =       '1'     THEN
                        mp2     :=      '1';
                  ELSE  mp2     :=      '0';
                  END   IF;
                  IF    p3      =       '1'     THEN
                        mp3     :=      '1';
                  ELSE  mp3     :=      '0';
                  END   IF;
                  IF    p4      =       '1'     THEN
                        mp4     :=      '1';
                  ELSE  mp4     :=      '0';
                  END   IF;
            END IF;
            pass1   <=      mp1;
            pass2   <=      mp2;
            pass3   <=      mp3;
            pass4   <=      mp4;
      END   PROCESS;
END   be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY      controlread     IS
PORT  (     innh    :       IN      INTEGER         RANGE 0 TO 300;
            innv    :       IN      INTEGER         RANGE 0 TO 1000;
            p       :       IN      BIT;
            p1      :       IN      BIT;
            p2      :       IN      BIT;
            p3      :       IN      BIT;
            p4      :       IN      BIT;
            adclk   :       IN      BIT;
            dataA   :       INOUT   STD_LOGIC_VECTOR (7 DOWNTO 0);
            dataB   :       INOUT   STD_LOGIC_VECTOR (7 DOWNTO 0);
            dataC   :       INOUT   STD_LOGIC_VECTOR (7 DOWNTO 0);
                le              :       OUT     BIT;
                read1           :       OUT     BIT;
                read2           :       OUT     BIT;
                read3           :       OUT     BIT;
                read4           :       OUT     BIT);
END     controlread;

ARCHITECTURE        be      OF      controlread     IS
BEGIN
        PROCESS(innh,innv,p,p1,p2,p3,p4,adclk)
        BEGIN
                IF      p       =       '0'     THEN
                        IF      adclk   =       '1'     THEN
                                dataA   <=      "00000000";
                                dataB   <=      "00000000";
                                dataC   <=      "00000000";
                                read1   <=      '0';
                                read2   <=      '0';
                                read3   <=      '0';
                                read4   <=      '0';
                        END     IF;
                        IF      adclk   =       '0'     THEN
                                dataA   <=      "ZZZZZZZZ";
                                dataB   <=      "ZZZZZZZZ";
                                dataC   <=      "ZZZZZZZZ";
                                read1   <=      '0';
                                read2   <=      '0';
                                read3   <=      '0';
                                read4   <=      '0';
                        END     IF;
                END IF;
                IF      adclk   =       '1'     THEN
                        le      <=      '1';
                ELSIF   adclk   =       '0'     THEN
                        le      <=      '0';
                END IF;
                IF      p       =       '1'     THEN
                        IF      (innv >= 27 AND innv < 327)     THEN
                                IF      (innh >= 0 AND innh < 142)      THEN
                                        IF      p1      =       '1'     THEN
                                                read1   <=      '1';
                                        ELSE    read1   <=      '0';
                                        END     IF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                IF      p1      =       '0'     THEN
                        IF      adclk   =       '1'     THEN
                                dataA   <=      "00000000";
                                dataB   <=      "00000000";
                                dataC   <=      "00000000";
                        END     IF;
                        IF      adclk   =       '0'     THEN
                                dataA   <=      "ZZZZZZZZ";
                                dataB   <=      "ZZZZZZZZ";
                                dataC   <=      "ZZZZZZZZ";
                        END     IF;
                END IF;
        END IF;
        IF      (innh >= 0 AND innh < 284)      THEN
                IF      p2      =       '1'     THEN
                        read2   <=      '1';
                ELSE    read2   <=      '0';
                END     IF;
                IF      p2      =       '0'     THEN
                        IF      adclk   =       '1'     THEN
                                dataA   <=      "00000000";
                                dataB   <=      "00000000";
                                dataC   <=      "00000000";
                        END     IF;
                        IF      adclk   =       '0'     THEN
                                dataA   <=      "ZZZZZZZZ";
                                dataB   <=      "ZZZZZZZZ";
                                dataC   <=      "ZZZZZZZZ";
                        END     IF;
                END IF;
        END IF;
END IF;
IF      (innv >= 327 AND innv < 627)    THEN
        IF      (innh >= 0 AND innh < 142)      THEN
                IF      p3      =       '1'     THEN
                        read3   <=      '1';
                ELSE    read3   <=      '0';
                END     IF;
                IF      p3      =       '0'     THEN
                        IF      adclk   =       '1'     THEN
                                dataA   <=      "00000000";
                                dataB   <=      "00000000";
                                dataC   <=      "00000000";
                        END     IF;
                        IF      adclk   =       '0'     THEN
                                dataA   <=      "ZZZZZZZZ";
                                dataB   <=      "ZZZZZZZZ";
                                dataC   <=      "ZZZZZZZZ";
                        END     IF;
                END IF;
        END IF;
        IF      (innh >= 0 AND innh < 284)      THEN
                IF      p4      =       '1'     THEN
                        read4   <=      '1';
                ELSE    read4   <=      '0';
                END     IF;
                IF      p4      =       '0'     THEN
                        IF      adclk   =       '1'     THEN
                                dataA   <=      "00000000";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        dataB   <=      "00000000";
                                        dataC   <=      "00000000";
                                END     IF;
                                IF      adclk   =       '0'     THEN
                                        dataA   <=      "ZZZZZZZZ";
                                        dataB   <=      "ZZZZZZZZ";
                                        dataC   <=      "ZZZZZZZZ";
                                END     IF;
                        END IF;
                END IF;
        END IF;
    END PROCESS;
END be;
```

.....

```
ENTITY  shsynIS
PORT    (       hsyn    :       IN      BIT;
                hout    :       OUT     INTEGER         RANGE 0 TO 2);
END     shsyn;

ARCHITECTURE        be      OF      shsyn   IS
BEGIN
        PROCESS (hsyn)
                VARIABLE    nh      :       INTEGER         RANGE 0 TO 2;
        BEGIN
                IF      (hsyn'EVENT     AND     hsyn = '1')     THEN
                        nh      :=      nh+1;
                END IF;
                IF      nh      =       2       THEN
                        nh      :=      0;
                END IF;
                hout    <=      nh;
        END PROCESS;
END     be;
```

.....

```
ENTITY          shsyn   IS
PORT    (       hsyn    :       IN      BIT;
                hout    :       OUT     INTEGER         RANGE 0 TO 2);
END     shsyn;

ARCHITECTURE        be      OF      shsyn   IS
BEGIN
        PROCESS(hsyn)
                VARIABLE    nh      :       INTEGER         RANGE 0 TO 2;
        BEGIN
                IF      (hsyn'EVENT     AND     hsyn = '1')     THEN
                        nh      :=      nh+1;
                END IF;
                IF      nh      =       2       THEN
                        nh      :=      0;
                END IF;
                hout    <=      nh;
        END PROCESS;
END     be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY testsam IS
PORT   (        pass    :       IN      INTEGER RANGE 0 TO 1;
                vsyn    :       IN      BIT;
                hsyn    :       IN      BIT;
                clk     :       IN      BIT;
                nhin    :       IN      INTEGER RANGE 0 TO 3;
                p       :       OUT     BIT);
END testsam;

ARCHITECTURE    be      OF      testsam  IS
BEGIN
        PROCESS(hsyn,vsyn,clk)
                VARIABLE        vcheck  :       INTEGER         RANGE 0 TO 1;
                VARIABLE        hcheck  :       INTEGER         RANGE 0 TO 1;
                VARIABLE        vpass   :       INTEGER         RANGE 0 TO 1;
                VARIABLE        hpass   :       INTEGER         RANGE 0 TO 1;
                VARIABLE        nh      :       INTEGER         RANGE 0 TO 700;
                VARIABLE        hdelay  :       INTEGER         RANGE       0 TO1500;
        BEGIN
                IF      (vsyn = '0')    THEN
                        vcheck  :=      1;
                END IF;
                IF      (hsyn'EVENT AND hsyn = '1')     THEN
                        IF      vcheck  =       1       THEN
                                nh      :=      nh+1;
                        END IF;
                END IF;
                IF      (nh     >=      27 AND nh <= 627)       THEN
                        vpass   :=      1;
                ELSE    vpass   :=      0;
                END IF;

                IF      (hsyn = '0')    THEN
                        hcheck  :=      1;
                END IF;
                IF      (clk'EVENT AND clk = '1')               THEN
                        IF      vpass   =       1       THEN
                                IF      hcheck  =       1       THEN
                                        hdelay  :=      hdelay+1;
                                END IF;
                        END IF;
                END IF;
                IF      (hdelay >=      216     AND hdelay <= 1354)     THEN
                        hpass   :=      1;
                ELSE    hpass   :=      0;
                END IF;

                IF      (vsyn = '1')    THEN
                        vcheck  :=      0;
                        nh      :=      0;
                END IF;
                IF      (hsyn = '1')    THEN
                        hcheck  :=      0;
                        hdelay  :=      0;
                END IF;

                IF      (nhin =  1 AND pass = 1) THEN
                        IF      hpass   =       1       THEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    p       <=      '1';
                ELSE p      <=      '0';
                END IF;
            ELSE p      <=      '0';
            END IF;
        END PROCESS;
END be;
```

.....

```
ENTITY testsam IS
PORT    (       pass    :       IN INTEGER RANGE 0 TO 1;
                vsyn    :       IN BIT;
                hsyn    :       IN BIT;
                clk     :       IN BIT;
                nhin    :       IN INTEGER RANGE 0 TO 3;
                p       :       OUT BIT);
END testsam;

ARCHITECTURE    be      OF      testsam  IS
BEGIN
        PROCESS(hsyn,vsyn,clk)
                VARIABLE    vcheck  :       INTEGER     RANGE 0 TO 1;
                VARIABLE    hcheck  :       INTEGER     RANGE 0 TO 1;
                VARIABLE    vpass   :       INTEGER     RANGE 0 TO 1;
                VARIABLE    hpass   :       INTEGER     RANGE 0 TO 1;
                VARIABLE    nh      :       INTEGER     RANGE 0 TO 700;
                VARIABLE    hdelay  :       INTEGER     RANGE       0 TO1500;
        BEGIN
                IF      (vsyn = '0')    THEN
                        vcheck  :=      1;
                END IF;
                IF      (hsyn'EVENT AND hsyn = '1')     THEN
                        IF      vcheck  =       1       THEN
                                nh      :=      nh+1;
                        END IF;
                END IF;
                IF      (nh     >=      27 AND nh <= 627)       THEN
                        vpass   :=      1;
                ELSE    vpass   :=      0;
                END IF;

                IF      (hsyn = '0')    THEN
                        hcheck  :=      1;
                END IF;
                IF      (clk'EVENT AND clk = '1')               THEN
                        IF      vpass   =       1       THEN
                                IF      hcheck  =       1       THEN
                                        hdelay  :=      hdelay+1;
                                END IF;
                        END IF;
                END IF;
                IF      (hdelay >=      216     AND hdelay <= 1354)     THEN
                        hpass   :=      1;
                ELSE    hpass   :=      0;
                END IF;

                IF      (vsyn = '1')    THEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                vcheck  :=      0;
                nh      :=      0;
        END IF;
        IF      (hsyn = '1')    THEN
                hcheck  :=      0;
                hdelay  :=      0;
        END IF;

        IF      (nhin =  1 AND pass = 1) THEN
                IF      hpass   =       1       THEN
                        p       <=      '1';
                ELSE    p       <=      '0';
                END IF;
        ELSE    p       <=      '0';
        END IF;
    END PROCESS;
END be;
```

.....

```
ENTITY          checklimit      IS
PORT    (       nin     :       IN      INTEGER         RANGE 0 TO 150;
                hsyn    :       IN      BIT;
                pass    :       OUT     INTEGER         RANGE 0 TO 1);
END     checklimit;

ARCHITECTURE    be      OF      checklimit      IS
BEGIN
    PROCESS(nin,hsyn)
        VARIABLE        p       :       INTEGER RANGE 0 TO 1;
    BEGIN
        IF      nin     =       142     THEN
                p       :=      0;
        END     IF;
        IF      ( hsyn = '1')   THEN
                p       :=      1;
        END IF;
        pass    <=      p;
    END     PROCESS;
END     be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      checklimit      IS
PORT  (     nin     :       IN      INTEGER     RANGE 0 TO 150;
            hsyn    :       IN      BIT;
            pass    :       OUT     INTEGER     RANGE 0 TO 1);
END   checklimit;

ARCHITECTURE        be      OF      checklimit      IS
BEGIN
      PROCESS(nin,hsyn)
            VARIABLE    p       :       INTEGER RANGE 0 TO 1;
      BEGIN
            IF      nin     =       142     THEN
                    p       :=      0;
            END     IF;
            IF      ( hsyn = '1')   THEN
                    p       :=      1;
            END IF;
            pass    <=      p;
      END   PROCESS;
END   be;
```

.....

```
ENTITY      checksam        IS
PORT  (     sam     :       IN      BIT;
            nout    :       OUT     INTEGER     RANGE 0 TO 3);
END checksam;

ARCHITECTURE        be      OF      checksam        IS
BEGIN
      PROCESS(sam)
            VARIABLE    n       :       INTEGER RANGE 0 TO 3;
      BEGIN
            IF      (sam'EVENT AND sam = '1')   THEN
                    n       :=      n+1;
            END IF;
            IF      n       =       2       THEN
                    n       :=      0;
            END IF;
            nout    <=      n;
      END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      gensam  IS
PORT  (     n       :       IN      INTEGER RANGE 0 TO 3;
            p1      :       OUT     BIT;
            p2      :       OUT     BIT);
END   gensam;

ARCHITECTURE        be      OF      gensam  IS
BEGIN
      PROCESS (n)
      BEGIN
            CASE n IS
                  WHEN 0 =>     p1 <= '1';
                                p2 <= '0';
                  WHEN  1 =>    p1 <= '0';
                                p2 <= '1';
                  WHEN  2 =>    p1 <= '1';
                                p2 <= '0';
                  WHEN OTHERS=>       p1      <=      '0';
                                      p2      <=      '0';
            END CASE;
      END PROCESS;
END be;
```

.....

```
ENTITY      limitsam IS
PORT  (     le      :       IN      BIT;
            ler     :       IN      BIT;
            hsyn    :       IN      BIT;
            nout    :       OUT     INTEGER RANGE 0 TO 150);
END   limitsam;

ARCHITECTURE        be      OF      limitsam IS
BEGIN
      PROCESS(le,ler,hsyn)
            VARIABLE    n       :       INTEGER RANGE 0 TO 150;
            VARIABLE    nn      :       INTEGER RANGE 0 TO 150;
            VARIABLE    np      :       INTEGER RANGE 0 TO 150;
      BEGIN
            IF      (le'EVENT AND le = '0')   THEN
                    n       :=      n+1;
            END     IF;
            IF      (ler'EVENT AND ler = '0')         THEN
                    nn      :=      nn+1;
            END     IF;
            IF      (hsyn = '1')      THEN
                    nn      :=      0;
                    n       :=      0;
            END     IF;
            nout    <=      (nn+n);
            END     PROCESS;
END   be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      controllatch    IS
PORT  (     p1              :   IN    INTEGER     RANGE 0 TO 1;
            p2              :   IN    INTEGER     RANGE 0 TO 1;
            pp1             :   IN    INTEGER     RANGE 0 TO 1;
            pp2             :   IN    INTEGER     RANGE 0 TO 1;
            lcheck1  :      IN    INTEGER     RANGE 0 TO 1;
            lcheck2  :      IN    INTEGER     RANGE 0 TO 1;
            le1             :   OUT   BIT;
            le1r            :   OUT   BIT;
            le2             :   OUT   BIT;
            le2r            :   OUT   BIT;
            ad              :   OUT   INTEGER RANGE 0 TO 3);
END   controllatch;

ARCHITECTURE      be      OF      controllatch      IS
BEGIN
      PROCESS (P1,P2,PP1,PP2,lcheck1,lcheck2)
            VARIABLE    MPP1,MPP2           :   INTEGER RANGE 0 TO 1;
            VARIABLE    Mlcheck1,Mlcheck2   :   INTEGER RANGE 0 TO 1;
      BEGIN
            MPP1  :=    PP1;
            MPP2  :=    PP2;
            Mlcheck1    :=      lcheck1;
            Mlcheck2    :=      lcheck2;
            IF P1 = 0 AND P2 = 0 THEN
                  IF MPP1 = 1 AND Mlcheck1 = 0 THEN
                        LE1     <=      '1';
                        LE1R    <=      '0';
                        AD      <=      1;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
                  IF MPP1 = 1 AND Mlcheck1 = 1 THEN
                        LE1     <=      '0';
                        LE1R    <=      '1';
                        AD      <=      1;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
                  IF MPP2 = 1 THEN
                        LE1     <=      '0';
                        LE1R    <=      '0';
                        AD      <=      0;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
            END IF;
            IF P1 = 0 AND P2 = 1 THEN
                  IF MPP2 = 1 AND Mlcheck2 = 0 THEN
                        LE2     <=      '1';
                        LE2R    <=      '0';
                        AD      <=      2;
                        LE1     <=      '0';
                        LE1R    <=      '0';
                  END IF;
                  IF MPP2 = 1 AND Mlcheck2 = 1 THEN
                        LE2     <=      '0';
                        LE2R    <=      '1';
                        AD      <=      2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LE1     <=      '0';
                LE1R    <=      '0';
            END IF;
            IF MPP1 = 1 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      0;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
        END IF;
        IF P1 = 1 AND P2 = 1 THEN
            IF MPP1 = 1 AND Mlcheck1 = 0 THEN
                LE1     <=      '1';
                LE1R    <=      '0';
                AD      <=      1;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
            IF MPP1 = 1 AND Mlcheck1 = 1 THEN
                LE1     <=      '0';
                LE1R    <=      '1';
                AD      <=      1;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
            IF MPP2 = 1 AND Mlcheck2 = 0 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      2;
                LE2     <=      '1';
                LE2R    <=      '0';
            END IF;
            IF MPP2 = 1 AND Mlcheck1 = 0 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      2;
                LE2     <=      '0';
                LE2R    <=      '1';
            END IF;
        END IF;
        IF P1 = 0 AND P2 = 0 THEN
            LE1     <=      '0';
            LE1R    <=      '0';
            AD      <=      0;
            LE2     <=      '0';
            LE2R    <=      '0';
        END IF;
    END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      limitsam IS
PORT  (     le      :     IN      BIT;
            ler     :     IN      BIT;
            hsyn    :     IN      BIT;
            nout    :     OUT     INTEGER RANGE 0 TO 150);
END   limitsam;

ARCHITECTURE      be      OF      limitsam IS
BEGIN
      PROCESS(le,ler,hsyn)
            VARIABLE    n     :     INTEGER RANGE 0 TO 150;
            VARIABLE    nn    :     INTEGER RANGE 0 TO 150;
            VARIABLE    np    :     INTEGER RANGE 0 TO 150;
      BEGIN
            IF    (le'EVENT AND le = '0')  THEN
                  n     :=    n+1;
            END   IF;
            IF    (ler'EVENT AND ler = '0')      THEN
                  nn    :=    nn+1;
            END   IF;
            IF    (hsyn = '1')     THEN
                  nn    :=    0;
                  n     :=    0;
            END   IF;
            nout  <=    (nn+n);
            END   PROCESS;
END   be;

      ..........................................................................


ENTITY      lcheckno      IS
PORT  (     le            :     IN      BIT;
            ler           :     IN      BIT;
            oe            :     IN      BIT;
            oer           :     IN      BIT;
            lcheck        :     OUT     INTEGER     RANGE 0 TO 1;
            lcheckr       :     OUT     INTEGER     RANGE 0 TO 1;
            no            :     OUT     INTEGER     RANGE 0 TO 3);
END   lcheckno;

ARCHITECTURE      be      OF      lcheckno      IS
BEGIN
      PROCESS(le,ler,oe,oer)
            VARIABLE    check   :     INTEGER     RANGE 0 TO 1;
            VARIABLE    checkr  :     INTEGER     RANGE 0 TO 1;
            VARIABLE    n       :     INTEGER     RANGE 0 TO 3;
      BEGIN
            IF    (le'EVENT AND le = '0')  THEN
                  check   :=    1;
            END   IF;
            IF    (oe = '0')THEN
                  check   :=    0;
            END   IF;
            IF    (ler'EVENT AND ler = '0')      THEN
                  checkr  :=    1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                IF      check   =       1       THEN
                        n       :=      2;
                END IF;
                IF      check   =       0       THEN
                        n       :=      1;
                END     IF;
        END     IF;
        IF      (oer = '0')     THEN
                checkr  :=      0;
        END     IF;
        lcheck  <=      check;
        lcheckr <=      checkr;
        no      <=      n;
    END PROCESS;
END be;
```

.....

```
ENTITY      lcheckno        IS
PORT    (   le              :       IN      BIT;
            ler             :       IN      BIT;
            oe              :       IN      BIT;
            oer             :       IN      BIT;
            lcheck          :       OUT     INTEGER     RANGE 0 TO 1;
            lcheckr         :       OUT     INTEGER     RANGE 0 TO 1;
            no              :       OUT     INTEGER     RANGE 0 TO 3);
END     lcheckno;

ARCHITECTURE        be      OF      lcheckno        IS
BEGIN
    PROCESS(le,ler,oe,oer)
        VARIABLE    check   :       INTEGER     RANGE 0 TO 1;
        VARIABLE    checkr  :       INTEGER     RANGE 0 TO 1;
        VARIABLE    n       :       INTEGER     RANGE 0 TO 3;
    BEGIN
        IF      (le'EVENT AND le = '0')  THEN
                check   :=      1;
        END     IF;
        IF      (oe = '0')THEN
                check   :=      0;
        END     IF;
        IF      (ler'EVENT AND ler = '0')       THEN
                checkr  :=      1;
                IF      check   =       1       THEN
                        n       :=      2;
                END IF;
                IF      check   =       0       THEN
                        n       :=      1;
                END     IF;
        END     IF;
        IF      (oer = '0')     THEN
                checkr  :=      0;
        END     IF;
        lcheck  <=      check;
        lcheckr <=      checkr;
        no      <=      n;
    END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      statewrite      IS
PORT  (     lcheck  :       IN      INTEGER     RANGE 0 TO 1;
            lcheckr :       IN      INTEGER     RANGE 0 TO 1;
            no      :       IN      INTEGER     RANGE 0 TO 3;
            write   :       OUT     BIT;
            writer  :       OUT     BIT);
END   statewrite;

ARCHITECTURE      be      OF      statewrite      IS
BEGIN
      PROCESS(lcheck,lcheckr,no)
      BEGIN
            IF    (lcheck  = 1    AND lcheckr = 0)      THEN
                  write    <=     '1';
                  writer   <=     '0';
            END IF;
            IF    (lcheck  = 0    AND lcheckr = 1)      THEN
                  write    <=     '0';
                  writer   <=     '1';
            END IF;
            IF    (lcheck  = 1    AND lcheckr = 1)      THEN
                  IF     no       =       1       THEN
                         write    <=      '0';
                         write    <=      '1';
                  ELSE   write    <=      '1';
                         writer   <=      '0';
                  END IF;
            END IF;
            IF    (lcheck  = 0    AND lcheckr = 0)      THEN
                  write    <=     '0';
                  writer   <=     '0';
            END IF;
      END PROCESS;
END be;
```

.....

```
ENTITY      statewrite      IS
PORT  (     lcheck  :       IN      INTEGER     RANGE 0 TO 1;
            lcheckr :       IN      INTEGER     RANGE 0 TO 1;
            no      :       IN      INTEGER     RANGE 0 TO 3;
            write   :       OUT     BIT;
            writer  :       OUT     BIT);
END   statewrite;

ARCHITECTURE      be      OF      statewrite      IS
BEGIN
      PROCESS(lcheck,lcheckr,no)
      BEGIN
            IF    (lcheck  = 1    AND lcheckr = 0)      THEN
                  write    <=     '1';
                  writer   <=     '0';
            END IF;
            IF    (lcheck  = 0    AND lcheckr = 1)      THEN
                  write    <=     '0';
                  writer   <=     '1';
            END IF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                IF      (lcheck  = 1      AND lcheckr = 1)          THEN
                        IF      no      =       1       THEN
                                write   <=      '0';
                                write   <=      '1';
                        ELSE    write   <=      '1';
                                writer  <=      '0';
                        END IF;
                END IF;
                IF      (lcheck  = 0      AND lcheckr = 0)          THEN
                        write   <=      '0';
                        writer  <=      '0';
                END IF;
        END PROCESS;
END be;
                .....

ENTITY          shsyn   IS
PORT    (       hsyn    :       IN      BIT;
                hout    :       OUT     INTEGER         RANGE 0 TO 2);
END     shsyn;

ARCHITECTURE            be      OF      shsyn   IS
BEGIN
        PROCESS(hsyn)
                VARIABLE        nh      :       INTEGER         RANGE 0 TO 2;
        BEGIN
                IF      (hsyn'EVENT     AND     hsyn = '1')     THEN
                        nh      :=      nh+1;
                END IF;
                IF      nh      =       2       THEN
                        nh      :=      0;
                END IF;
                hout    <=      nh;
        END PROCESS;
END     be;
                .....

ENTITY          shsyn   IS
PORT    (       hsyn    :       IN      BIT;
                hout    :       OUT     INTEGER         RANGE 0 TO 2);
END     shsyn;

ARCHITECTURE            be      OF      shsyn   IS
BEGIN
        PROCESS(hsyn)
                VARIABLE        nh      :       INTEGER         RANGE 0 TO 2;
        BEGIN
                IF      (hsyn'EVENT     AND     hsyn = '1')     THEN
                        nh      :=      nh+1;
                END IF;
                IF      nh      =       2       THEN
                        nh      :=      0;
                END IF;
                hout    <=      nh;
        END PROCESS;
END     be;
                .....
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY testsam IS
PORT  (       pass    :       IN      INTEGER RANGE 0 TO 1;
              vsyn    :       IN      BIT;
              hsyn    :       IN      BIT;
              clk     :       IN      BIT;
              nhin    :       IN      INTEGER RANGE 0 TO 3;
              p       :       OUT     BIT);
END testsam;

ARCHITECTURE      be      OF      testsam  IS
BEGIN
      PROCESS(hsyn,vsyn,clk)
            VARIABLE    vcheck  :       INTEGER     RANGE 0 TO 1;
            VARIABLE    hcheck  :       INTEGER     RANGE 0 TO 1;
            VARIABLE    vpass   :       INTEGER     RANGE 0 TO 1;
            VARIABLE    hpass   :       INTEGER     RANGE 0 TO 1;
            VARIABLE    nh      :       INTEGER     RANGE 0 TO 700;
            VARIABLE    hdelay  :       INTEGER     RANGE       0 TO1500;
      BEGIN
            IF    (vsyn = '0')      THEN
                  vcheck  :=      1;
            END IF;
            IF    (hsyn'EVENT AND hsyn = '1')     THEN
                  IF      vcheck  =       1       THEN
                          nh      :=      nh+1;
                  END IF;
            END IF;
            IF    (nh     >=      27 AND nh <= 627)       THEN
                  vpass   :=      1;
            ELSE  vpass   :=      0;
            END IF;

            IF    (hsyn = '0')      THEN
                  hcheck  :=      1;
            END IF;
            IF    (clk'EVENT AND clk = '1')       THEN
                  IF      vpass   =       1       THEN
                          IF      hcheck  =       1       THEN
                                  hdelay  :=      hdelay+1;
                          END IF;
                  END IF;
            END IF;
            IF    (hdelay  >=     216     AND hdelay <= 1354)     THEN
                  hpass   :=      1;
            ELSE  hpass   :=      0;
            END IF;

            IF    (vsyn = '1')      THEN
                  vcheck  :=      0;
                  nh      :=      0;
            END IF;
            IF    (hsyn = '1')      THEN
                  hcheck  :=      0;
                  hdelay  :=      0;
            END IF;
            IF    (nhin =  1 AND pass = 1)  THEN
                  IF      hpass   =       1       THEN
                          p       <=      '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ELSE    p       <=      '0';
                END IF;
        ELSE    p       <=      '0';
        END IF;
    END PROCESS;
END be;

.....

ENTITY testsam IS
PORT    (       pass    :       IN INTEGER RANGE 0 TO 1;
                vsyn    :       IN BIT;
                hsyn    :       IN BIT;
                clk     :       IN BIT;
                nhin    :       IN INTEGER RANGE 0 TO 3;
                p       :       OUT BIT);
END testsam;

ARCHITECTURE    be      OF      testsam  IS
BEGIN
    PROCESS(hsyn,vsyn,clk)
        VARIABLE    vcheck  :   INTEGER     RANGE 0 TO 1;
        VARIABLE    hcheck  :   INTEGER     RANGE 0 TO 1;
        VARIABLE    vpass   :   INTEGER     RANGE 0 TO 1;
        VARIABLE    hpass   :   INTEGER     RANGE 0 TO 1;
        VARIABLE    nh      :   INTEGER     RANGE 0 TO 700;
        VARIABLE    hdelay  :   INTEGER     RANGE       0 TO1500;
    BEGIN
        IF      (vsyn = '0')    THEN
                vcheck  :=      1;
        END IF;
        IF      (hsyn'EVENT AND hsyn = '1')     THEN
                IF      vcheck  =       1       THEN
                        nh      :=      nh+1;
                END IF;
        END IF;
        IF      (nh     >=      27 AND nh <= 627)       THEN
                vpass   :=      1;
        ELSE    vpass   :=      0;
        END IF;

        IF      (hsyn = '0')    THEN
                hcheck  :=      1;
        END IF;
        IF      (clk'EVENT AND clk = '1')       THEN
                IF      vpass   =       1       THEN
                        IF      hcheck  =       1       THEN
                                hdelay  :=      hdelay+1;
                        END IF;
                END IF;
        END IF;
        IF      (hdelay >=      216     AND hdelay <= 1354)     THEN
                hpass   :=      1;
        ELSE    hpass   :=      0;
        END IF;

        IF      (vsyn = '1')    THEN
                vcheck  :=      0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                nh      :=      0;
        END IF;
        IF      (hsyn = '1')    THEN
                hcheck  :=      0;
                hdelay  :=      0;
        END IF;

        IF      (nhin =  1 AND pass = 1) THEN
                IF      hpass   =       1       THEN
                        p       <=      '1';
                ELSE    p       <=      '0';
                END IF;
        ELSE    p       <=      '0';
        END IF;
    END PROCESS;
END be;
```

.....

```
ENTITY          checklimit      IS
PORT    (       nin     :       IN      INTEGER         RANGE 0 TO 150;
                hsyn    :       IN      BIT;
                pass    :       OUT     INTEGER         RANGE 0 TO 1);
END     checklimit;

ARCHITECTURE    be      OF      checklimit      IS
BEGIN
        PROCESS(nin,hsyn)
                VARIABLE        p       :       INTEGER RANGE 0 TO 1;
        BEGIN
                IF      nin     =       142     THEN
                        p       :=      0;
                END     IF;
                IF      ( hsyn  =       '1')    THEN
                        p       :=      1;
                END IF;
                pass    <=      p;
        END     PROCESS;
END     be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      checklimit      IS
PORT  (     nin     :       IN      INTEGER     RANGE 0 TO 150;
            hsyn    :       IN      BIT;
            pass    :       OUT     INTEGER     RANGE 0 TO 1);
END   checklimit;

ARCHITECTURE        be      OF      checklimit      IS
BEGIN
      PROCESS(nin,hsyn)
            VARIABLE        p       :       INTEGER RANGE 0 TO 1;
      BEGIN
            IF      nin     =       142     THEN
                    p       :=      0;
            END     IF;
            IF      ( hsyn = '1')   THEN
                    p       :=      1;
            END IF;
            pass    <=      p;
      END   PROCESS;
END   be;
```

.....

```
ENTITY      checksam        IS
PORT  (     sam     :       IN      BIT;
            nout    :       OUT     INTEGER     RANGE 0 TO 3);
END checksam;

ARCHITECTURE        be      OF      checksam        IS
BEGIN
      PROCESS(sam)
            VARIABLE        n       :       INTEGER RANGE 0 TO 3;
      BEGIN
            IF      (sam'EVENT AND sam = '1')       THEN
                    n       :=      n+1;
            END IF;
            IF      n       =       2       THEN
                    n       :=      0;
            END IF;
            nout    <=      n;
      END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      gensam  IS
PORT  (     n       :       IN      INTEGER RANGE 0 TO 3;
            p1      :       OUT     BIT;
            p2      :       OUT     BIT);
END   gensam;

ARCHITECTURE        be      OF      gensam  IS
BEGIN
      PROCESS (n)
      BEGIN
            CASE n IS
                  WHEN 0 =>     p1 <= '1';
                                p2 <= '0';
                  WHEN  1 =>    p1 <= '0';
                                p2 <= '1';
                  WHEN  2 =>    p1 <= '1';
                                p2 <= '0';
                  WHEN OTHERS=>       p1      <=      '0';
                                      p2      <=      '0';
            END CASE;
      END PROCESS;
END be;
```

.....

```
ENTITY      limitsam IS
PORT  (     le      :       IN      BIT;
            ler     :       IN      BIT;
            hsyn    :       IN      BIT;
            nout    :       OUT     INTEGER RANGE 0 TO 150);
END   limitsam;

ARCHITECTURE        be      OF      limitsam IS
BEGIN
      PROCESS(le,ler,hsyn)
            VARIABLE    n       :       INTEGER RANGE 0 TO 150;
            VARIABLE    nn      :       INTEGER RANGE 0 TO 150;
            VARIABLE    np      :       INTEGER RANGE 0 TO 150;
      BEGIN
            IF    (le'EVENT AND le = '0')   THEN
                  n       :=      n+1;
            END   IF;
            IF    (ler'EVENT AND ler = '0')         THEN
                  nn      :=      nn+1;
            END   IF;
            IF    (hsyn = '1')      THEN
                  nn      :=      0;
                  n       :=      0;
            END   IF;
            nout  <=      (nn+n);
            END   PROCESS;
END   be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      controllatch    IS
PORT  (     p1              :     IN      INTEGER       RANGE 0 TO 1;
            p2              :     IN      INTEGER       RANGE 0 TO 1;
            pp1             :     IN      INTEGER       RANGE 0 TO 1;
            pp2             :     IN      INTEGER       RANGE 0 TO 1;
            lcheck1         :     IN      INTEGER       RANGE 0 TO 1;
            lcheck2         :     IN      INTEGER       RANGE 0 TO 1;
            le1             :     OUT     BIT;
            le1r            :     OUT     BIT;
            le2             :     OUT     BIT;
            le2r            :     OUT     BIT;
            ad              :     OUT     INTEGER RANGE 0 TO 3);
END   controllatch;

ARCHITECTURE      be      OF      controllatch      IS
BEGIN
      PROCESS (P1,P2,PP1,PP2,lcheck1,lcheck2)
            VARIABLE    MPP1,MPP2           :   INTEGER RANGE 0 TO 1;
            VARIABLE    Mlcheck1,Mlcheck2   :   INTEGER RANGE 0 TO 1;
      BEGIN
            MPP1        :=      PP1;
            MPP2        :=      PP2;
            Mlcheck1    :=      lcheck1;
            Mlcheck2    :=      lcheck2;
            IF P1 = 0 AND P2 = 0 THEN
                  IF MPP1 = 1 AND Mlcheck1 = 0 THEN
                        LE1     <=      '1';
                        LE1R    <=      '0';
                        AD      <=      1;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
                  IF MPP1 = 1 AND Mlcheck1 = 1 THEN
                        LE1     <=      '0';
                        LE1R    <=      '1';
                        AD      <=      1;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
                  IF MPP2 = 1 THEN
                        LE1     <=      '0';
                        LE1R    <=      '0';
                        AD      <=      0;
                        LE2     <=      '0';
                        LE2R    <=      '0';
                  END IF;
            END IF;
            IF P1 = 0 AND P2 = 1 THEN
                  IF MPP2 = 1 AND Mlcheck2 = 0 THEN
                        LE2     <=      '1';
                        LE2R    <=      '0';
                        AD      <=      2;
                        LE1     <=      '0';
                        LE1R    <=      '0';
                  END IF;
                  IF MPP2 = 1 AND Mlcheck2 = 1 THEN
                        LE2     <=      '0';
                        LE2R    <=      '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                AD      <=      2;
                LE1     <=      '0';
                LE1R    <=      '0';
            END IF;
            IF MPP1 = 1 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      0;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
        END IF;
        IF P1 = 1 AND P2 = 1 THEN
            IF MPP1 = 1 AND Mlcheck1 = 0 THEN
                LE1     <=      '1';
                LE1R    <=      '0';
                AD      <=      1;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
            IF MPP1 = 1 AND Mlcheck1 = 1 THEN
                LE1     <=      '0';
                LE1R    <=      '1';
                AD      <=      1;
                LE2     <=      '0';
                LE2R    <=      '0';
            END IF;
            IF MPP2 = 1 AND Mlcheck2 = 0 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      2;
                LE2     <=      '1';
                LE2R    <=      '0';
            END IF;
            IF MPP2 = 1 AND Mlcheck1 = 0 THEN
                LE1     <=      '0';
                LE1R    <=      '0';
                AD      <=      2;
                LE2     <=      '0';
                LE2R    <=      '1';
            END IF;
        END IF;
        IF P1 = 0 AND P2 = 0 THEN
            LE1     <=      '0';
            LE1R    <=      '0';
            AD      <=      0;
            LE2     <=      '0';
            LE2R    <=      '0';
        END IF;
    END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      limitsam IS
PORT  (     le      :       IN      BIT;
            ler     :       IN      BIT;
            hsyn    :       IN      BIT;
            nout    :       OUT     INTEGER RANGE 0 TO 150);
END   limitsam;

ARCHITECTURE      be      OF      limitsam IS
BEGIN
      PROCESS(le,ler,hsyn)
            VARIABLE    n       :       INTEGER RANGE 0 TO 150;
            VARIABLE    nn      :       INTEGER RANGE 0 TO 150;
            VARIABLE    np      :       INTEGER RANGE 0 TO 150;
      BEGIN
            IF    (le'EVENT AND le = '0')  THEN
                  n       :=      n+1;
            END   IF;
            IF    (ler'EVENT AND ler = '0')        THEN
                  nn      :=      nn+1;
            END   IF;
            IF    (hsyn = '1')      THEN
                  nn      :=      0;
                  n       :=      0;
            END   IF;
            nout  <=      (nn+n);
            END   PROCESS;
END   be;
```

.....

```
ENTITY      lcheckno      IS
PORT  (     le                :       IN      BIT;
            ler               :       IN      BIT;
            oe                :       IN      BIT;
            oer               :       IN      BIT;
            lcheck            :       OUT     INTEGER     RANGE 0 TO 1;
            lcheckr           :       OUT     INTEGER     RANGE 0 TO 1;
            no                :       OUT     INTEGER     RANGE 0 TO 3);
END   lcheckno;

ARCHITECTURE      be      OF      lcheckno      IS
BEGIN
      PROCESS(le,ler,oe,oer)
            VARIABLE    check   :       INTEGER     RANGE 0 TO 1;
            VARIABLE    checkr  :       INTEGER     RANGE 0 TO 1;
            VARIABLE    n       :       INTEGER     RANGE 0 TO 3;
      BEGIN
            IF    (le'EVENT AND le = '0')  THEN
                  check   :=      1;
            END   IF;
            IF    (oe = '0')THEN
                  check   :=      0;
            END   IF;
            IF    (ler'EVENT AND ler = '0')        THEN
                  checkr  :=      1;
                  IF      check   =       1       THEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    n       :=      2;
                END IF;
                IF      check   =       0       THEN
                        n       :=      1;
                END     IF;
        END     IF;
        IF      (oer = '0')     THEN
                checkr  :=      0;
        END     IF;
        lcheck  <=      check;
        lcheckr <=      checkr;
        no      <=      n;
    END PROCESS;
END be;
```

.....

```
ENTITY      lcheckno        IS
PORT    (   le              :       IN      BIT;
            ler             :       IN      BIT;
            oe              :       IN      BIT;
            oer             :       IN      BIT;
            lcheck          :       OUT     INTEGER     RANGE 0 TO 1;
            lcheckr         :       OUT     INTEGER     RANGE 0 TO 1;
            no              :       OUT     INTEGER     RANGE 0 TO 3);
END     lcheckno;

ARCHITECTURE        be      OF      lcheckno        IS
BEGIN
    PROCESS(le,ler,oe,oer)
        VARIABLE    check   :       INTEGER     RANGE 0 TO 1;
        VARIABLE    checkr  :       INTEGER     RANGE 0 TO 1;
        VARIABLE    n       :       INTEGER     RANGE 0 TO 3;
    BEGIN
        IF      (le'EVENT AND le = '0')  THEN
                check   :=      1;
        END     IF;
        IF      (oe = '0')THEN
                check   :=      0;
        END     IF;
        IF      (ler'EVENT AND ler = '0')       THEN
                checkr  :=      1;
                IF      check   =       1       THEN
                        n       :=      2;
                END IF;
                IF      check   =       0       THEN
                        n       :=      1;
                END     IF;
        END     IF;
        IF      (oer = '0')     THEN
                checkr  :=      0;
        END     IF;
        lcheck  <=      check;
        lcheckr <=      checkr;
        no      <=      n;
    END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      statewrite      IS
PORT  (     lcheck   :      IN      INTEGER      RANGE 0 TO 1;
            lcheckr  :      IN      INTEGER      RANGE 0 TO 1;
            no       :      IN      INTEGER      RANGE 0 TO 3;
            write    :      OUT     BIT;
            writer   :      OUT     BIT);
END   statewrite;

ARCHITECTURE      be      OF      statewrite      IS
BEGIN
      PROCESS(lcheck,lcheckr,no)
      BEGIN
            IF    (lcheck  = 1     AND lcheckr = 0)      THEN
                  write    <=      '1';
                  writer   <=      '0';
            END IF;
            IF    (lcheck  = 0     AND lcheckr = 1)      THEN
                  write    <=      '0';
                  writer   <=      '1';
            END IF;
            IF    (lcheck  = 1     AND lcheckr = 1)      THEN
                  IF       no      =       1       THEN
                           write   <=      '0';
                           write   <=      '1';
                  ELSE     write   <=      '1';
                           writer  <=      '0';
                  END IF;
            END IF;
            IF    (lcheck  = 0     AND lcheckr = 0)      THEN
                  write    <=      '0';
                  writer   <=      '0';
            END IF;
      END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      statewrite      IS
PORT  (     lcheck  :       IN      INTEGER     RANGE 0 TO 1;
            lcheckr :       IN      INTEGER     RANGE 0 TO 1;
            no      :       IN      INTEGER     RANGE 0 TO 3;
            write   :       OUT     BIT;
            writer  :       OUT     BIT);
END   statewrite;

ARCHITECTURE      be      OF      statewrite      IS
BEGIN
      PROCESS(lcheck,lcheckr,no)
      BEGIN
            IF    (lcheck  = 1    AND lcheckr = 0)      THEN
                  write    <=     '1';
                  writer   <=     '0';
            END IF;
            IF    (lcheck  = 0    AND lcheckr = 1)      THEN
                  write    <=     '0';
                  writer   <=     '1';
            END IF;
            IF    (lcheck  = 1    AND lcheckr = 1)      THEN
                  IF       no     =       1       THEN
                           write  <=      '0';
                           write  <=      '1';
                  ELSE     write  <=      '1';
                           writer <=      '0';
                  END IF;
            END IF;
            IF    (lcheck  = 0    AND lcheckr = 0)      THEN
                  write    <=     '0';
                  writer   <=     '0';
            END IF;
      END PROCESS;
END be;

            ..........

ENTITY      controlram      IS
PORT  (     write           :       IN      BIT;
            writer          :       IN      BIT;
            read            :       IN      BIT;
            adw             :       IN      INTEGER     RANGE 0 TO 100000;
            adr             :       IN      INTEGER     RANGE 0 TO 100000;
            adclk           :       IN      BIT;
            ad              :       OUT     INTEGER     RANGE 0 TO 100000;
            oel             :       OUT     BIT;
            oelr            :       OUT     BIT;
            ce1             :       OUT     BIT;
            ce2             :       OUT     BIT;
            we              :       OUT     BIT;
            oe              :       OUT     BIT);
END   controlram;

ARCHITECTURE      be      OF      controlram      IS
BEGIN
      PROCESS(write,writer,read,adclk,adw,adr)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        VARIABLE    check   :       INTEGER     RANGE 0 TO 1;
BEGIN
        IF      adclk   =       '1'     THEN
                check   :=      1;
        ELSE    check   :=      0;
        END IF;
        IF      (write = '1' AND writer = '0' AND read = '0')THEN
                IF      check   =       1       THEN
                        ad              <=      adw;
                        oel             <=      '0';
                        oelr            <=      '1';
                        ce1             <=      '0';
                        ce2             <=      '1';
                        we              <=      '0';
                        oe              <=      '1';
                ELSE    oel             <=      '1';
                        oelr            <=      '1';
                        ce1             <=      '1';
                        ce2             <=      '0';
                        we              <=      '1';
                        oe              <=      '1';
                END IF;
        END IF;
        IF      (write = '0' AND writer = '1' AND read = '0')THEN
                IF      check   =       0       THEN
                        ad              <=      adw;
                        oel             <=      '1';
                        oelr            <=      '0';
                        ce1             <=      '0';
                        ce2             <=      '1';
                        we              <=      '0';
                        oe              <=      '1';
                ELSE    oel             <=      '1';
                        oelr            <=      '1';
                        ce1             <=      '1';
                        ce2             <=      '0';
                        we              <=      '1';
                        oe              <=      '1';
                END IF;
        END IF;
        IF      (write = '0' AND writer = '0' AND read = '1')THEN
                IF      check   =       0       THEN
                        ad              <=      adr;
                        oel             <=      '1';
                        oelr            <=      '1';
                        ce1             <=      '0';
                        ce2             <=      '1';
                        we              <=      '1';
                        oe              <=      '0';
                ELSE    oel             <=      '1';
                        oelr            <=      '1';
                        ce1             <=      '1';
                        ce2             <=      '0';
                        we              <=      '1';
                        oe              <=      '1';
                END IF;
        END IF;
        IF      (write = '1' AND writer = '0' AND read = '1')THEN
                IF      check   =       1       THEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ad          <=      adr;
                    oel         <=      '1';
                    oelr        <=      '1';
                    ce1         <=      '0';
                    ce2         <=      '1';
                    we          <=      '1';
                    oe          <=      '0';
            END IF;
            IF      check   =       1       THEN
                    ad          <=      adw;
                    oel         <=      '0';
                    oelr        <=      '1';
                    ce1         <=      '0';
                    ce2         <=      '1';
                    we          <=      '0';
                    oe          <=      '1';
            END IF;
        END IF;
        IF      (write = '0' AND writer = '1' AND read = '1') THEN
            IF      check   =       0       THEN
                    ad          <=      adr;
                    oel         <=      '1';
                    oelr        <=      '1';
                    ce1         <=      '0';
                    ce2         <=      '1';
                    we          <=      '1';
                    oe          <=      '0';
            END IF;
            IF      check   =       1       THEN
                    ad          <=      adw;
                    oel         <=      '1';
                    oelr        <=      '0';
                    ce1         <=      '0';
                    ce2         <=      '1';
                    we          <=      '0';
                    oe          <=      '1';
            END IF;
        END IF;
        IF      (write = '0' AND writer = '0' AND read = '0') THEN
                oel             <=      '1';
                oelr            <=      '1';
                ce1             <=      '1';
                ce2             <=      '0';
                we              <=      '1';
                oe              <=      '1';
        END IF;
    END PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      addressOE   IS
PORT  (     oe      :   IN      BIT;
            nadd    :   OUT     INTEGER     RANGE 0 TO 100000);
END   addressOE;

ARCHITECTURE        be      OF      addressOE       IS
BEGIN
      PROCESS(oe)
            VARIABLE    n       :       INTEGER     RANGE 0 TO 100000;
      BEGIN
            IF      (oe'EVENT AND oe = '1') THEN
                    n       :=      n+1;
            END IF;
            IF      n       =       85200   THEN
                    n       :=      0;
            END IF;
            nadd    <=      n;
      END PROCESS;
END be;
```

.....

```
ENTITY      setpic  IS
PORT  (     adin    :   IN      INTEGER     RANGE 0 TO 100000;
            adout   :   IN      INTEGER     RANGE 0 TO 100000;
            pout    :   OUT     INTEGER     RANGE 0 TO 1);
END   setpic;
ARCHITECTURE        be      OF      setpic  IS
BEGIN
      PROCESS(adin,adout)
            VARIABLE    p1      :       BIT;
            VARIABLE    p2      :       BIT;
      BEGIN
            IF      adin    =       42599   THEN
                    p1      :=      '1';
            END IF;
            IF      adin    =       85199   THEN
                    p2      :=      '1';
            END     IF;
            IF      adout   =       42599   THEN
                    p1      :=      '0';
            END     IF;
            IF      adout   =       85199   THEN
                    p2      :=      '0';
            END IF;
            IF      (p1 = '1' OR p2 = '1')      THEN
                    pout    <=      1;
            ELSE    pout    <=      0;
            END     IF;
      END   PROCESS;
END be;
```

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ENTITY      controlram      IS
PORT  (     write           :     IN      BIT;
            writer          :     IN      BIT;
            read            :     IN      BIT;
            adw             :     IN      INTEGER     RANGE 0 TO 100000;
            adr             :     IN      INTEGER     RANGE 0 TO 100000;
            adclk           :     IN      BIT;
            ad              :     OUT     INTEGER     RANGE 0 TO 100000;
            oel             :     OUT     BIT;
            oelr            :     OUT     BIT;
            ce1             :     OUT     BIT;
            ce2             :     OUT     BIT;
            we              :     OUT     BIT;
            oe              :     OUT     BIT);
END   controlram;

ARCHITECTURE      be      OF      controlram      IS
BEGIN
      PROCESS(write,writer,read,adclk,adw,adr)
            VARIABLE      check   :     INTEGER     RANGE 0 TO 1;
      BEGIN
            IF      adclk   =     '1'     THEN
                    check   :=    1;
            ELSE    check   :=    0;
            END IF;
            IF      (write = '1' AND writer = '0' AND read = '0')THEN
                    IF      check   =     1       THEN
                            ad              <=    adw;
                            oel             <=    '0';
                            oelr            <=    '1';
                            ce1             <=    '0';
                            ce2             <=    '1';
                            we              <=    '0';
                            oe              <=    '1';
                    ELSE    oel             <=    '1';
                            oelr            <=    '1';
                            ce1             <=    '1';
                            ce2             <=    '0';
                            we              <=    '1';
                            oe              <=    '1';
                    END IF;
            END IF;
            IF      (write = '0' AND writer = '1' AND read = '0')THEN
                    IF      check   =     1       THEN
                            ad              <=    adw;
                            oel             <=    '1';
                            oelr            <=    '0';
                            ce1             <=    '0';
                            ce2             <=    '1';
                            we              <=    '0';
                            oe              <=    '1';
                    ELSE    oel             <=    '1';
                            oelr            <=    '1';
                            ce1             <=    '1';
                            ce2             <=    '0';
                            we              <=    '1';
                            oe              <=    '1';
                    END IF;
            END IF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF      (write = '0' AND writer = '0' AND read = '1')THEN
        IF      check   =       0       THEN
                ad              <=      adr;
                oel             <=      '1';
                oelr            <=      '1';
                ce1             <=      '0';
                ce2             <=      '1';
                we              <=      '1';
                oe              <=      '0';
        ELSE    oel             <=      '1';
                oelr            <=      '1';
                ce1             <=      '1';
                ce2             <=      '0';
                we              <=      '1';
                oe              <=      '1';
        END IF;
END IF;
IF      (write = '1' AND writer = '0' AND read = '1')THEN
        IF      check   =       0       THEN
                ad              <=      adr;
                oel             <=      '1';
                oelr            <=      '1';
                ce1             <=      '0';
                ce2             <=      '1';
                we              <=      '1';
                oe              <=      '0';
        END IF;
        IF      check   =       1       THEN
                ad              <=      adw;
                oel             <=      '0';
                oelr            <=      '1';
                ce1             <=      '0';
                ce2             <=      '1';
                we              <=      '0';
                oe              <=      '1';
        END IF;
END IF;
IF      (write = '0' AND writer = '1' AND read = '1')THEN
        IF      check   =       0       THEN
                ad              <=      adr;
                oel             <=      '1';
                oelr            <=      '1';
                ce1             <=      '0';
                ce2             <=      '1';
                we              <=      '1';
                oe              <=      '0';
        END IF;
        IF      check   =       1       THEN
                ad              <=      adw;
                oel             <=      '1';
                oelr            <=      '0';
                ce1             <=      '0';
                ce2             <=      '1';
                we              <=      '0';
                oe              <=      '1';
        END IF;
END IF;
IF      (write = '0' AND writer = '0' AND read = '0')THEN
        oel             <=      '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    oelr        <=      '1';
                    ce1         <=      '1';
                    ce2         <=      '0';
                    we          <=      '1';
                    oe          <=      '1';
                END IF;
        END PROCESS;
END be;
```

.....

```
ENTITY      addressOE   IS
PORT    (   oe      :   IN      BIT;
            nadd    :   OUT     INTEGER     RANGE 0 TO 100000);
END     addressOE;

ARCHITECTURE        be      OF      addressOE       IS
BEGIN
        PROCESS(oe)
                VARIABLE    n       :       INTEGER     RANGE 0 TO 100000;
        BEGIN
                IF      (oe'EVENT AND oe = '1') THEN
                        n       :=      n+1;
                END IF;
                IF      n       =       85200   THEN
                        n       :=      0;
                END IF;
                nadd    <=      n;
        END PROCESS;
END be;
```

.....

```
ENTITY      setpic  IS
PORT    (   adin    :   IN      INTEGER     RANGE 0 TO 100000;
            adout   :   IN      INTEGER     RANGE 0 TO 100000;
            pout    :   OUT     INTEGER     RANGE 0 TO 1);
END     setpic;
ARCHITECTURE        be      OF      setpic  IS
BEGIN
        PROCESS(adin,adout)
                VARIABLE    p1      :       BIT;
                VARIABLE    p2      :       BIT;
        BEGIN
                IF      adin    =       42599   THEN
                        p1      :=      '1';
                END IF;
                IF      adin    =       85199   THEN
                        p2      :=      '1';
                END     IF;
                IF      adout   =       42599   THEN
                        p1      :=      '0';
                END     IF;
                IF      adout   =       85199   THEN
                        p2      :=      '0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                END IF;
                IF      (p1 = '1' OR p2 = '1')      THEN
                        pout    <=      1;
                ELSE    pout    <=      0;
                END     IF;
        END     PROCESS;
END be;
```

.....

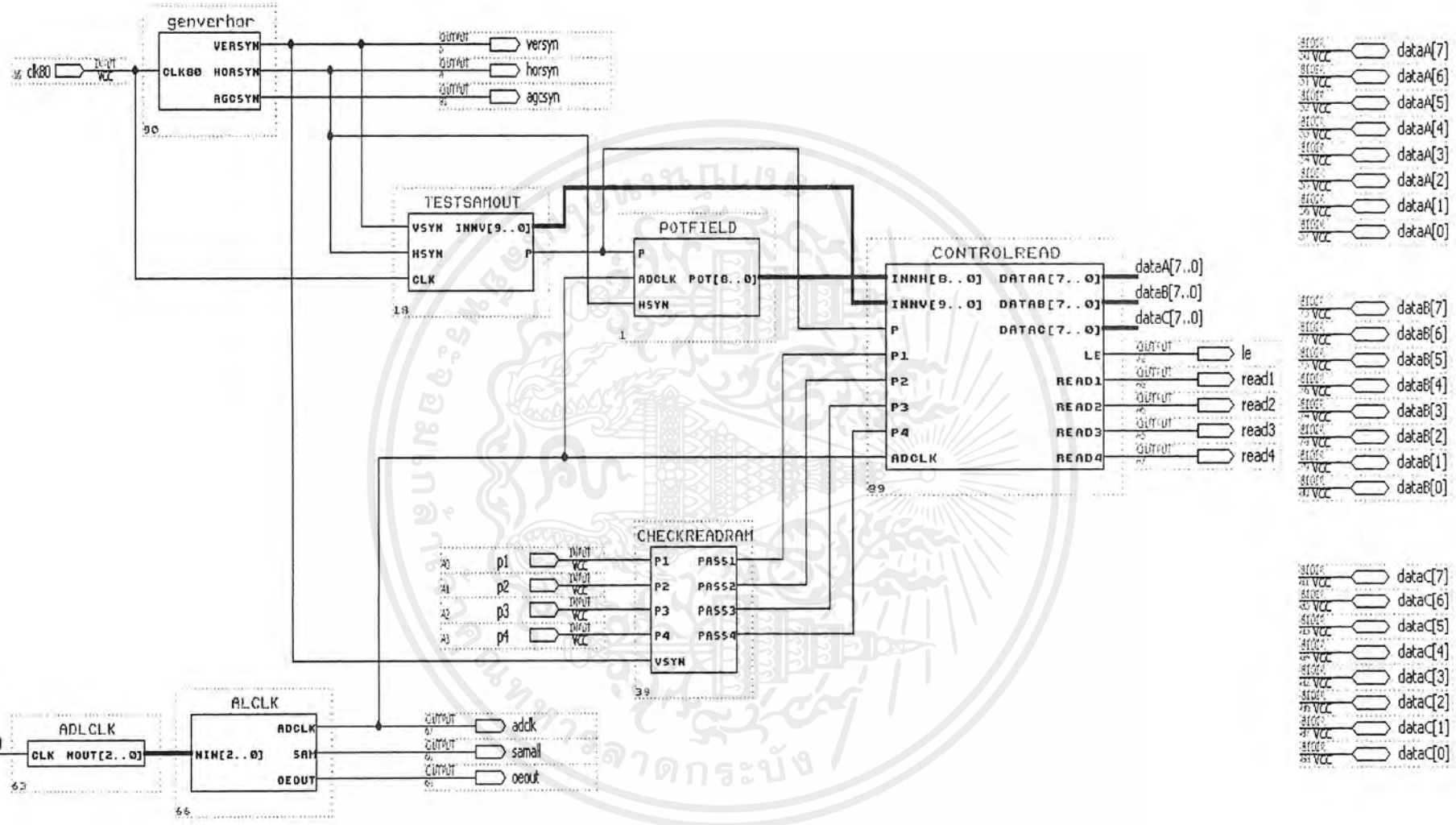
```
ENTITY          addressrtl      IS
PORT    (       we      :       IN      BIT;
                nadd    :       OUT     INTEGER         RANGE 0 TO 100000);
END     addressrtl;

ARCHITECTURE            be      OF      addressrtl      IS
BEGIN
        PROCESS(we)
                VARIABLE        n       :       INTEGER         RANGE 0 TO 100000;
        BEGIN
                IF      (we'EVENT AND we = '1')         THEN
                        n       :=      n+1;
                END IF;
                IF      n       =       85200   THEN
                        n       :=      0;
                END IF;
                nadd    <=      n;
        END PROCESS;
END be;
```

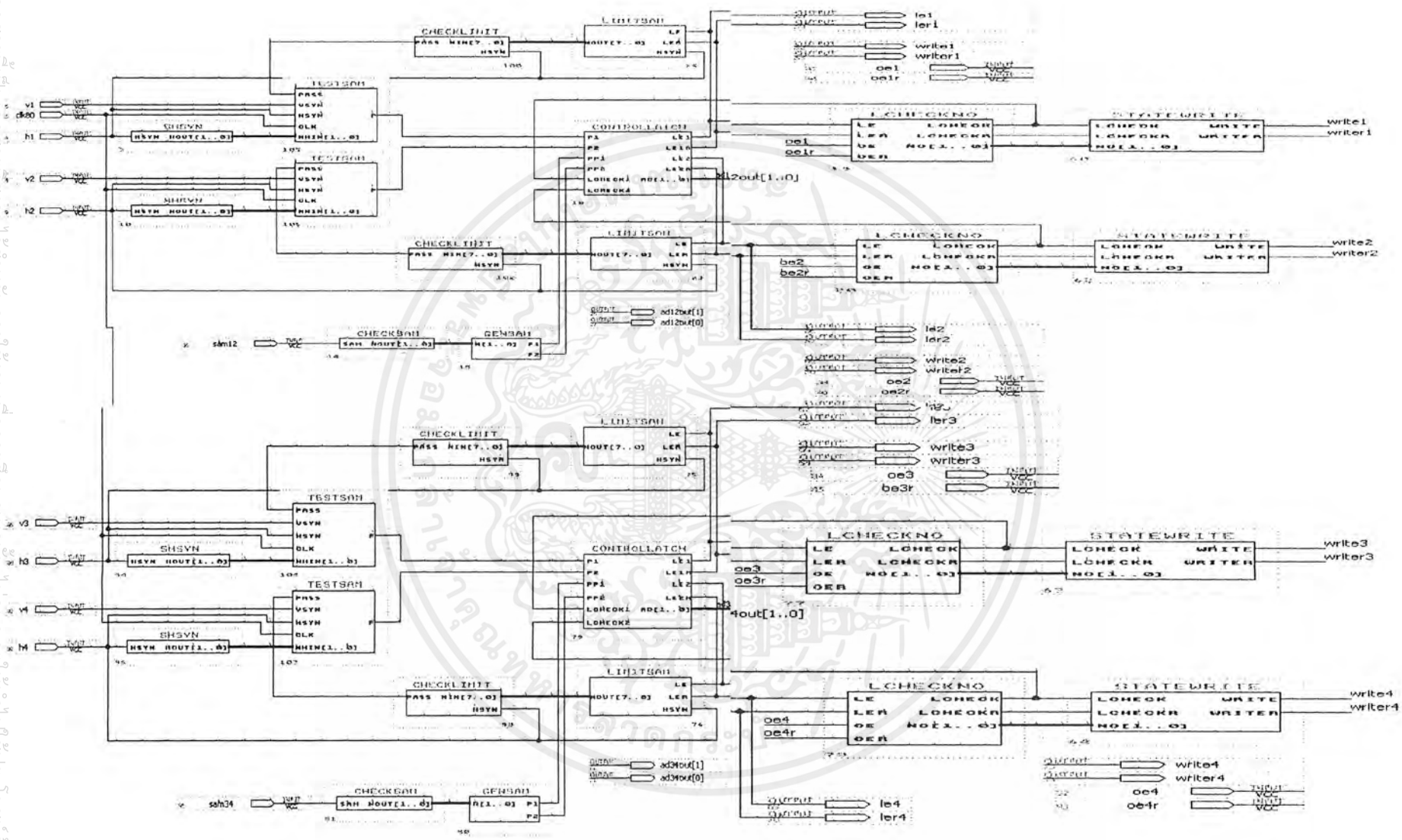
.....

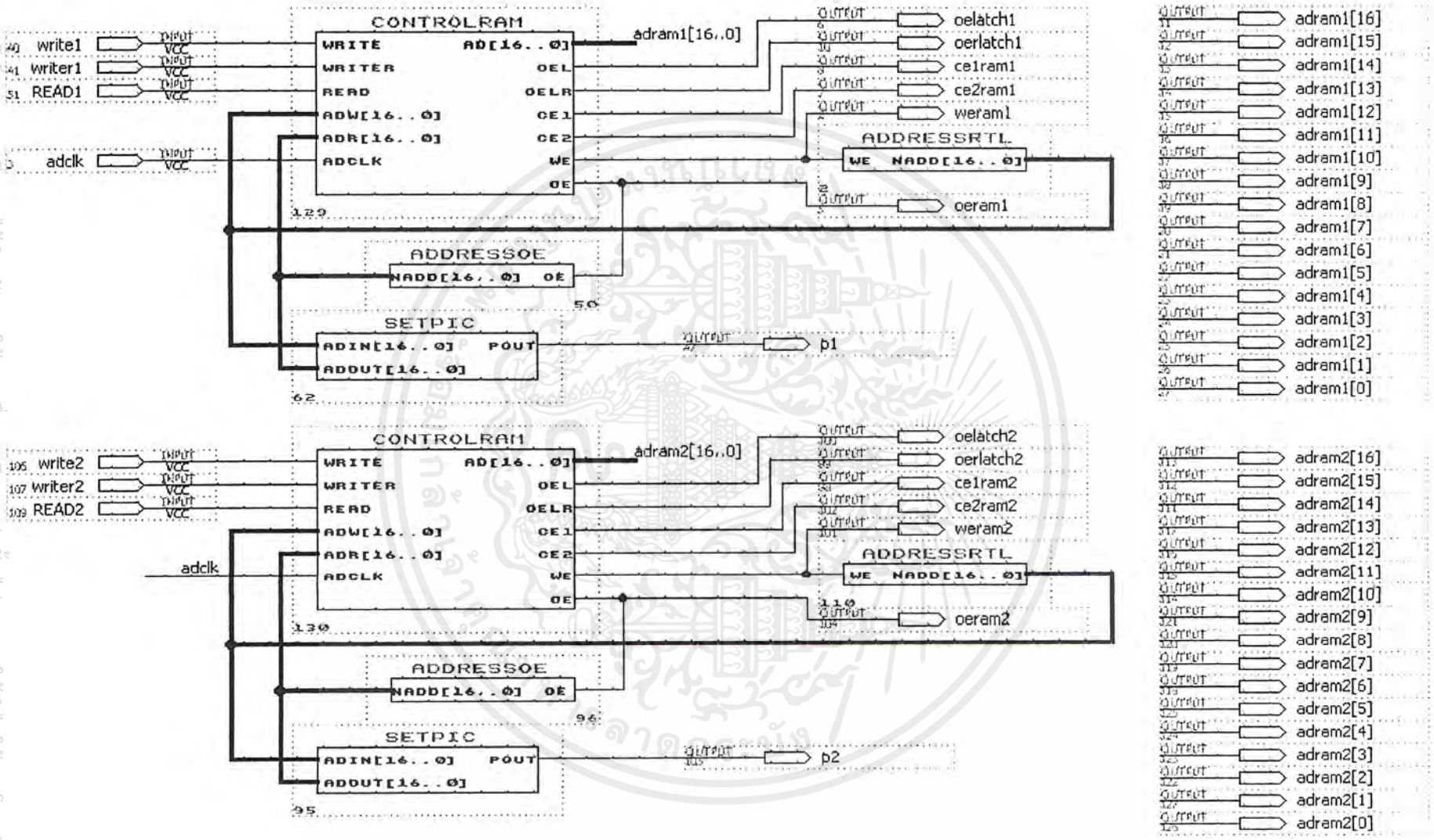
```
ENTITY          addressrtl      IS
PORT    (       we      :       IN      BIT;
                nadd    :       OUT     INTEGER         RANGE 0 TO 100000);
END     addressrtl;

ARCHITECTURE            be      OF      addressrtl      IS
BEGIN
        PROCESS(we)
                VARIABLE        n       :       INTEGER         RANGE 0 TO 100000;
        BEGIN
                IF      (we'EVENT AND we = '1')         THEN
                        n       :=      n+1;
                END IF;
                IF      n       =       85200   THEN
                        n       :=      0;
                END IF;
                nadd    <=      n;
        END PROCESS;
END be;
```

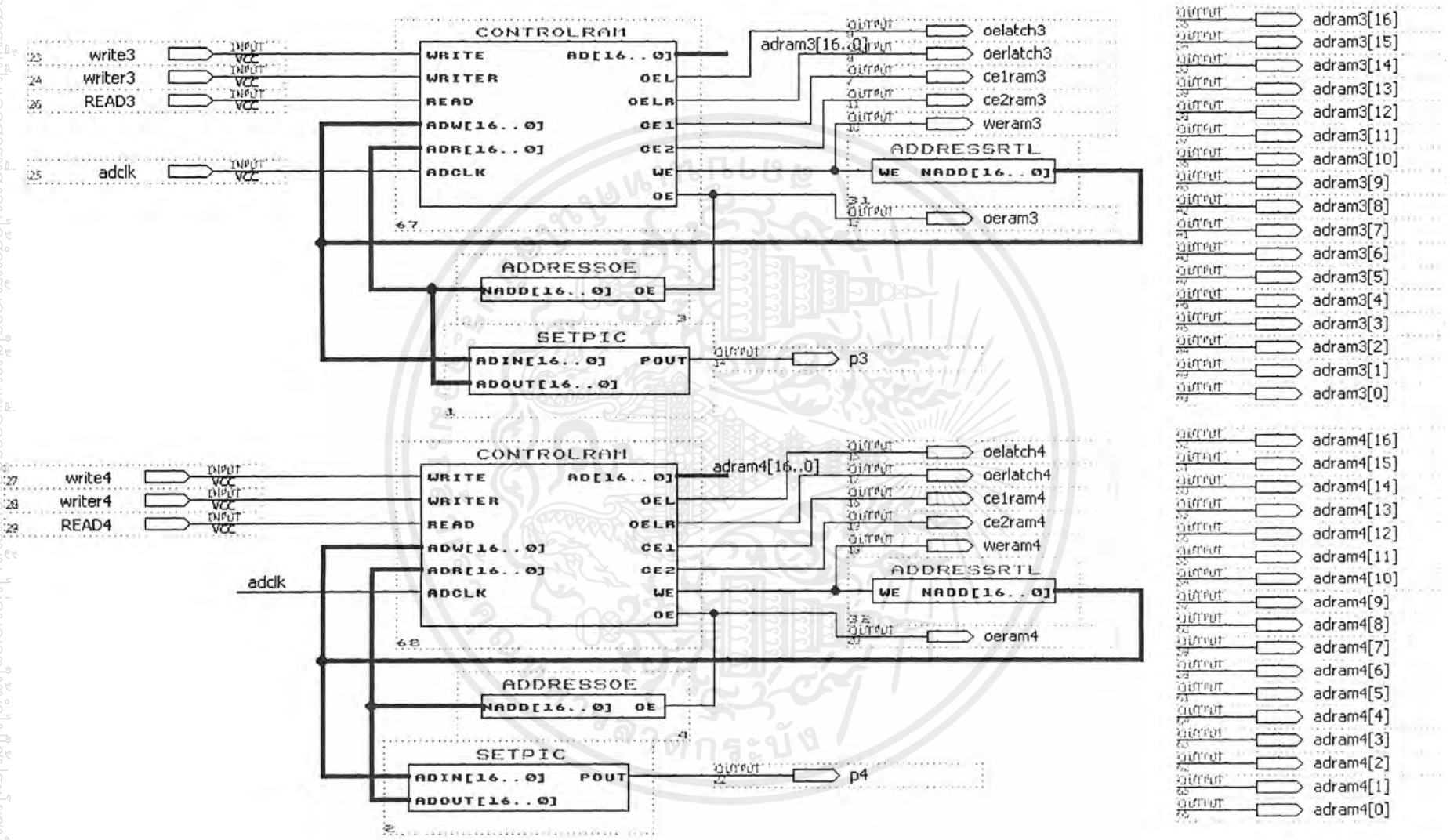
.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

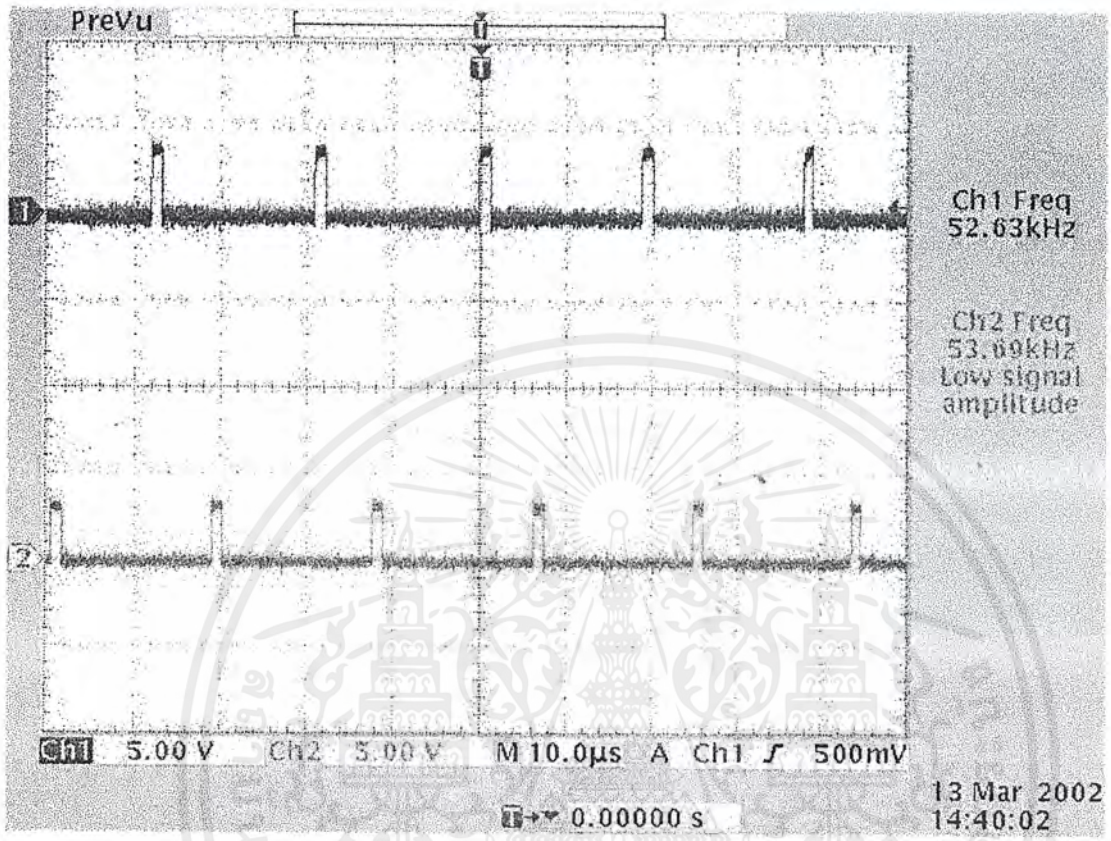

genverhor
VERSYN
CLK80
HORSYN
AGCSYN
90
ck80
versyn
horsyn
agcsyn
TESTSAMOUT
VSYN
INNV[9..0]
HSYN
P
CLK
18
POTFIELD
P
ADCLK
POT[8..0]
HSYN
1
CONTROLREAD
INNH[8..0]
INNV[9..0]
P
P1
P2
P3
P4
ADCLK
DATAA[7..0]
DATAB[7..0]
DATAC[7..0]
LE
READ1
READ2
READ3
READ4
89
dataA[7..0]
dataB[7..0]
dataC[7..0]
le
read1
read2
read3
read4
dataA[7]
dataA[6]
dataA[5]
dataA[4]
dataA[3]
dataA[2]
dataA[1]
dataA[0]
dataB[7]
dataB[6]
dataB[5]
dataB[4]
dataB[3]
dataB[2]
dataB[1]
dataB[0]
dataC[7]
dataC[6]
dataC[5]
dataC[4]
dataC[3]
dataC[2]
dataC[1]
dataC[0]
CHECKREADRAM
p1
p2
p3
p4
P1
P2
P3
P4
VSYN
PASS1
PASS2
PASS3
PASS4
39
ADLCLK
CLK
NOUT[2..0]
63
ALCLK
NIN[2..0]
ADCLK
SAM
OEOUT
66
addk
samall
oeout


CHECKLIMIT
LIMITSAM
TESTSAM
SHSYN
CONTROLLATCH
LCHECKNO
STATEWRITE
CHECKSAM
GENSAM
v1
clk80
h1
v2
h2
v3
h3
v4
h4
sam12
sam34
ld1
ler1
write1
writer1
oe1
oe1r
ad12out[1..0]
ld2
ler2
write2
writer2
oe2
oe2r
be2
be2r
ld3
ler3
write3
writer3
oe3
be3r
ad34out[1..0]
ld4
ler4
write4
writer4
oe4
oe4r


write1
writer1
READ1
adclk
CONTROLRAM1
WRITE
WRITER
READ
ADW[16..0]
ADR[16..0]
ADCLK
AD[16..0]
OEL
OELR
CE1
CE2
WE
OE
adram1[16..0]
oelatch1
oerlatch1
ce1ram1
ce2ram1
weram1
ADDRESSRTL
WE NADD[16..0]
oeram1
ADDRESSOE
NADD[16..0] OE
SETPIC
ADIN[16..0] POUT
ADDUT[16..0]
p1
write2
writer2
READ2
adclk
adram2[16..0]
oelatch2
oerlatch2
ce1ram2
ce2ram2
weram2
oeram2
p2
adram1[16]
adram1[15]
adram1[14]
adram1[13]
adram1[12]
adram1[11]
adram1[10]
adram1[9]
adram1[8]
adram1[7]
adram1[6]
adram1[5]
adram1[4]
adram1[3]
adram1[2]
adram1[1]
adram1[0]
adram2[16]
adram2[15]
adram2[14]
adram2[13]
adram2[12]
adram2[11]
adram2[10]
adram2[9]
adram2[8]
adram2[7]
adram2[6]
adram2[5]
adram2[4]
adram2[3]
adram2[2]
adram2[1]
adram2[0]


write3
writer3
READ3
adclk
CONTROLRAM
WRITE
WRITER
READ
ADW[16..0]
ADR[16..0]
ADCLK
AD[16..0]
OEL
OELR
CE1
CE2
WE
OE
adram3[16..0]
oelatch3
oerlatch3
ce1ram3
ce2ram3
weram3
ADDRESSRTL
WE NADD[16..0]
oeram3
ADDRESSOE
NADD[16..0] OE
SETPIC
ADIN[16..0]
ADOUT[16..0]
POUT
p3
write4
writer4
READ4
adclk
adram4[16..0]
oelatch4
oerlatch4
ce1ram4
ce2ram4
weram4
oeram4
p4
adram3[16]
adram3[15]
adram3[14]
adram3[13]
adram3[12]
adram3[11]
adram3[10]
adram3[9]
adram3[8]
adram3[7]
adram3[6]
adram3[5]
adram3[4]
adram3[3]
adram3[2]
adram3[1]
adram3[0]
adram4[16]
adram4[15]
adram4[14]
adram4[13]
adram4[12]
adram4[11]
adram4[10]
adram4[9]
adram4[8]
adram4[7]
adram4[6]
adram4[5]
adram4[4]
adram4[3]
adram4[2]
adram4[1]
adram4[0]

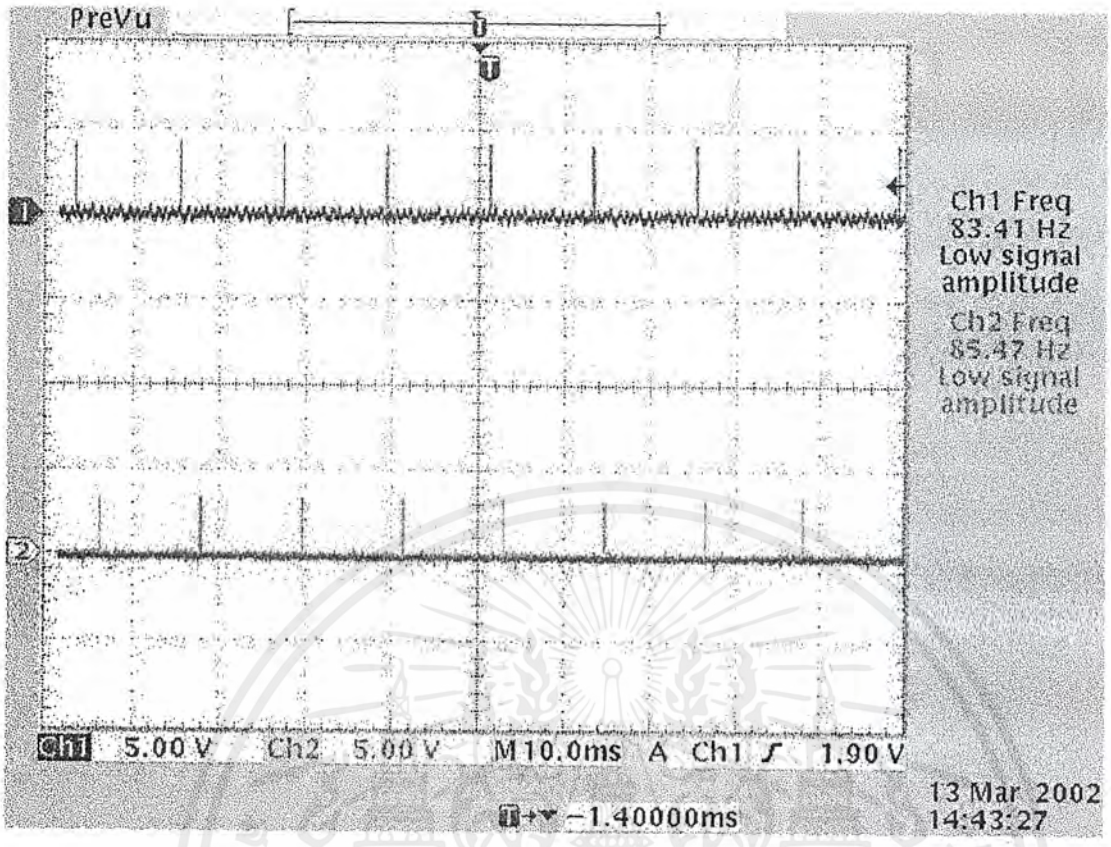
ผลการทดลอง



กราฟเส้นบน : Horizontal Synchronize Output

กราฟเส้นล่าง : Horizontal Synchronize Input

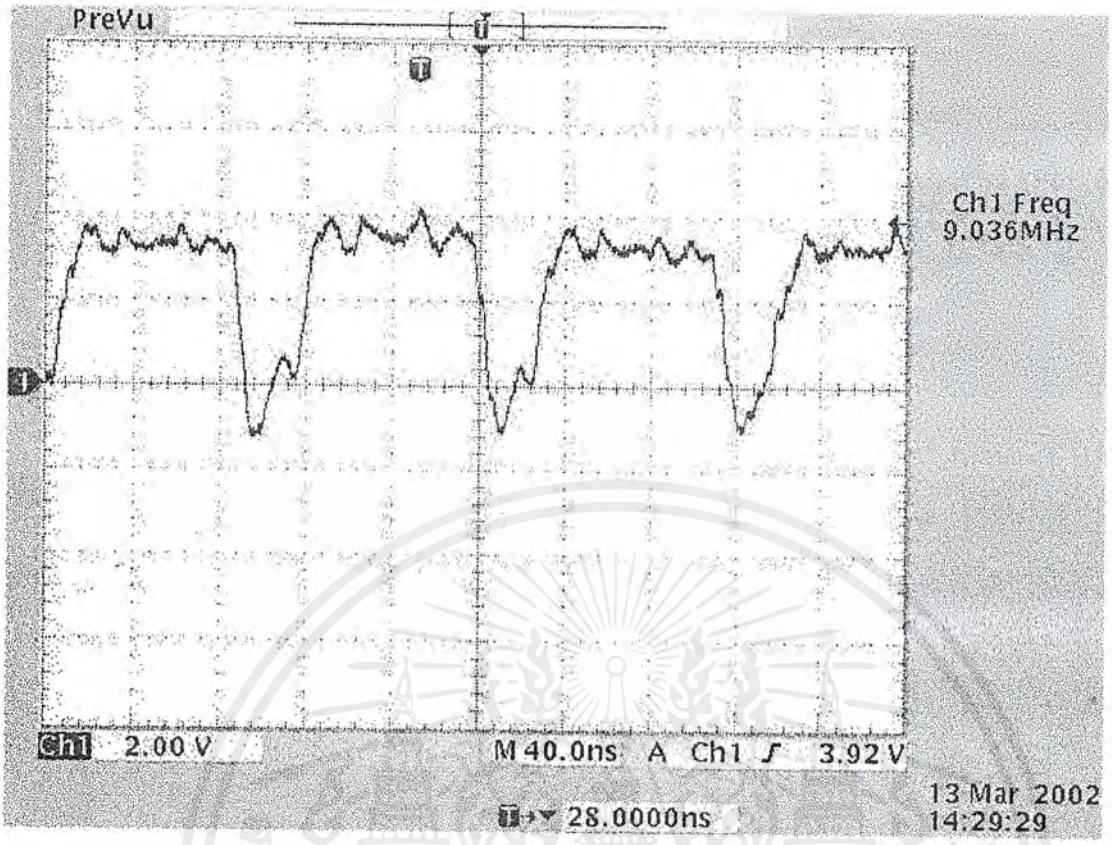
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟเส้นบน : Vertical Synchronize Output

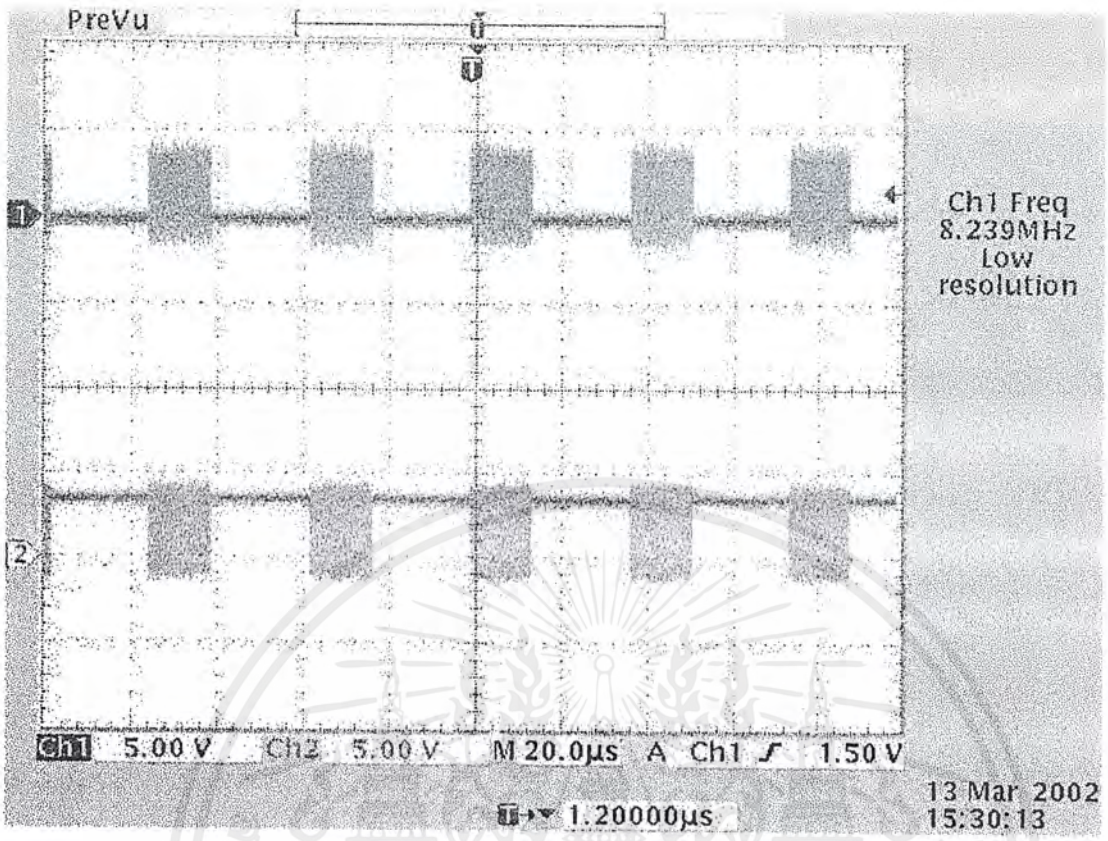
กราฟเส้นล่าง : Vertical Synchronize Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Control Enable ของ RAM

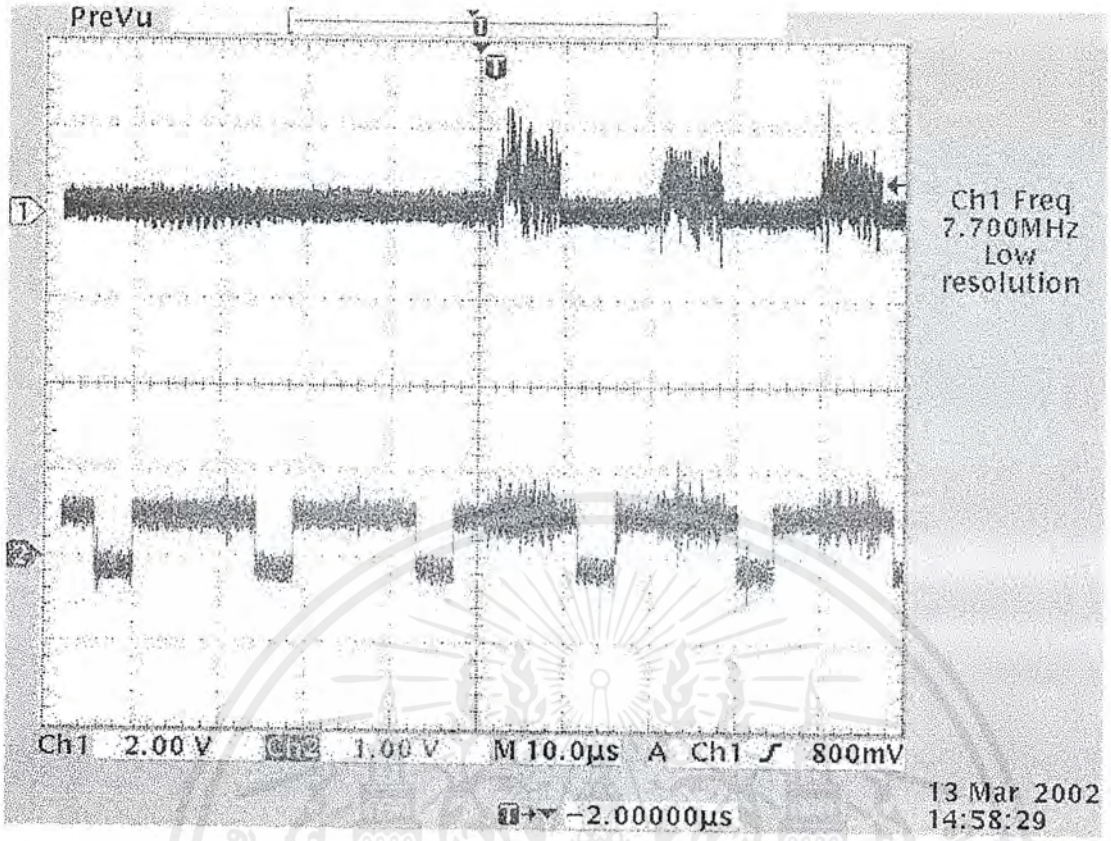
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟเส้นบน : OE (Output Enable)

กราฟเส้นล่าง : LE (Latch Enable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟเส้นบน : สัญญาณภาพ O/P จาก D/A ก่อนเข้า Filter

กราฟเส้นล่าง : สัญญาณภาพ I/P จากพอร์ต VGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 วิเคราะห์และสรุปผลการทดลอง

จากการใช้ VHDL ในการออกแบบ Detector Synchronize ทั้ง Horizontal Synchronize และ Vertical Synchronize พบว่าเมื่อ simulate ด้วยโปรแกรม MAX+PLUS II แล้ว ลักษณะของ Horizontal Synchronize และ Vertical Synchronize ที่ถูกสร้างขึ้นมีลักษณะใกล้เคียงกับ Horizontal Synchronize และ Vertical Synchronize ในการ control A/D ให้ทำการ sampling ภาพแรกสลับกับภาพที่สองไปเรื่อยๆ ได้ตรงตามที่เราออกแบบไว้

จากการใช้ VHDL ในการ control ตัว Latch โดยให้มีการรับข้อมูลของตัว Latch จาก A/D มา เพื่อพักข้อมูลไว้ก่อนจะถูกส่งไปที่ RAM โดย Latch จะมีการทำการเปลี่ยนไปเรื่อยๆ ซ้ำกันไป และยังมีการส่งข้อมูลจาก RAM ไปยัง Latch เพื่อพักข้อมูลก่อนที่จะส่งไปยัง D/A ซึ่งตรงตามที่เราออกแบบเอาไว้

จากการทดลองวัดสัญญาณที่ตำแหน่งต่างๆ ในวงจร ได้สัญญาณรูปกราฟที่ตำแหน่งต่างๆ ดังรูปที่แสดงไว้แล้ว พบว่า มีสัญญาณที่ตำแหน่งต่างๆ ตามที่วัด แสดงว่า มีการส่งสัญญาณได้จริงจากแหล่งจ่ายถึง มอนิเตอร์ที่ใช้แสดงผล แต่จากกราฟนั้นบางจุดมีสัญญาณรบกวนอยู่ ทำให้ผลออกมาไม่เรียบเท่าที่ควร และจากการทดลองโปรแกรมลงบนชิพ FPGA ให้สามารถแสดงผลได้ 4 ภาพในหน้าจอเดียวกันนั้น พบว่า มีการแบ่งหน้าจอเพื่อแสดงผลออกเป็น 4 ส่วนตามที่โปรแกรมไว้ แต่สัญญาณที่แสดงออกมาเป็นสัญญาณที่ไม่มีภาพเกิดขึ้น อาจเนื่องมาจากสัญญาณรบกวนเพราะใช้งานที่ความถี่สูง รวมถึงการเดินสายและการวางลายปริ้นถ้ามีความยาวมากก็จะเกิดสัญญาณรบกวนอีกเช่นกัน

ปัญหาและอุปสรรคในการทำโครงงานครั้งนี้

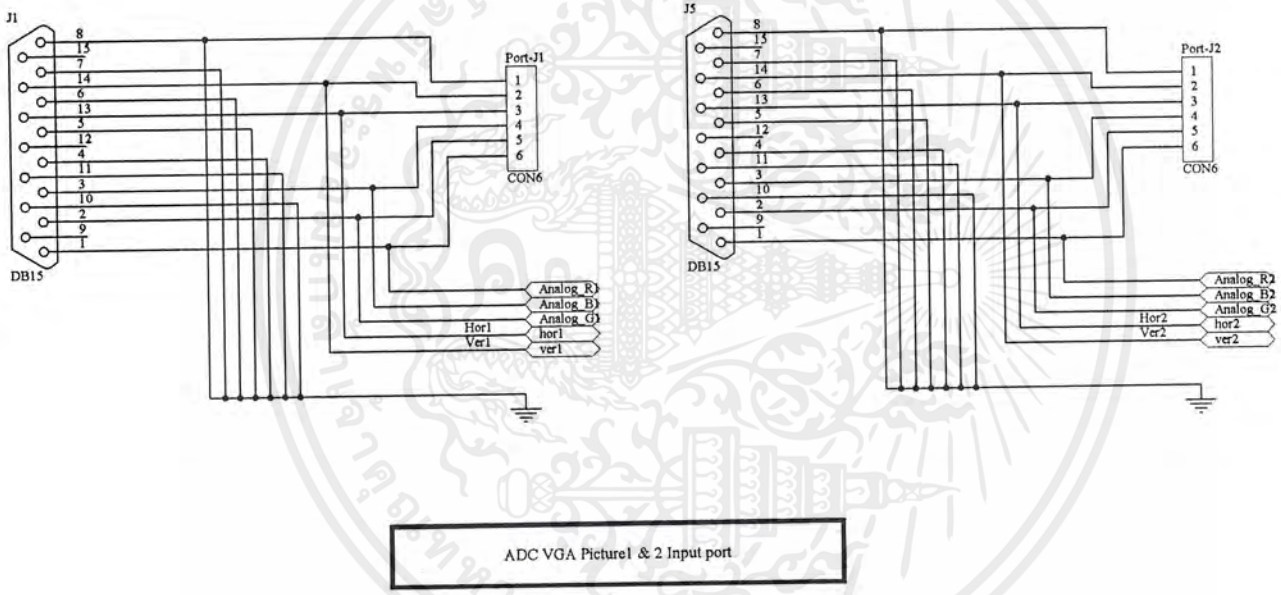
1. โครงงานนี้เป็นเรื่องที่ยังไม่แพร่หลายมากนัก ดังนั้น การค้นคว้าหรือหาข้อมูลเกี่ยวกับรายละเอียดของโครงงานทำได้ยาก
2. อุปกรณ์บางชนิดที่ใช้หาซื้อได้ยาก และใช้เวลานานพอสมควรในการจัดซื้อ
3. เนื่องจากเป็นการใช้งานที่ความถี่สูง จึงมีสัญญาณรบกวนมาก ดังนั้น จึงต้องระวังเรื่องสัญญาณรบกวนมากเป็นพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางในการแก้ปัญหา

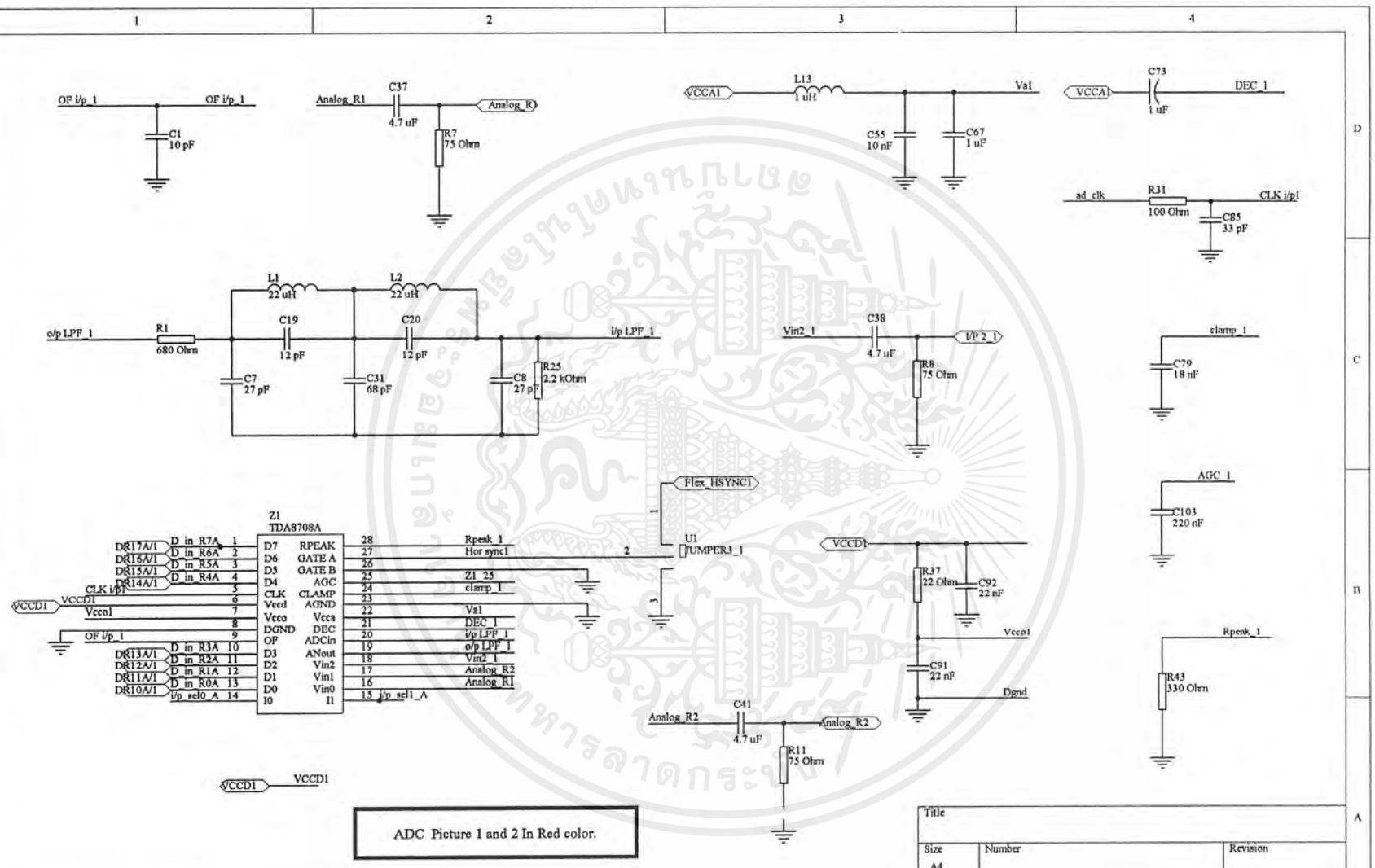
พยายามศึกษาค้นคว้าจากแหล่งข้อมูลให้มากที่สุด และเมื่อมีปัญหาควรปรึกษาผู้รู้ สำหรับเรื่องอุปกรณ์ พยายามออกแบบ และเลือกใช้อุปกรณ์ที่สามารถหาซื้อได้ง่าย และการออกแบบวงจรใช้งานที่ความถี่สูงควรพยายามเดินสายหรือลายปริ้นให้สั้นที่สุดเท่าที่จะทำได้ เพื่อป้องกันสัญญาณรบกวนที่อาจเกิดขึ้นเนื่องจากความยาวสาย

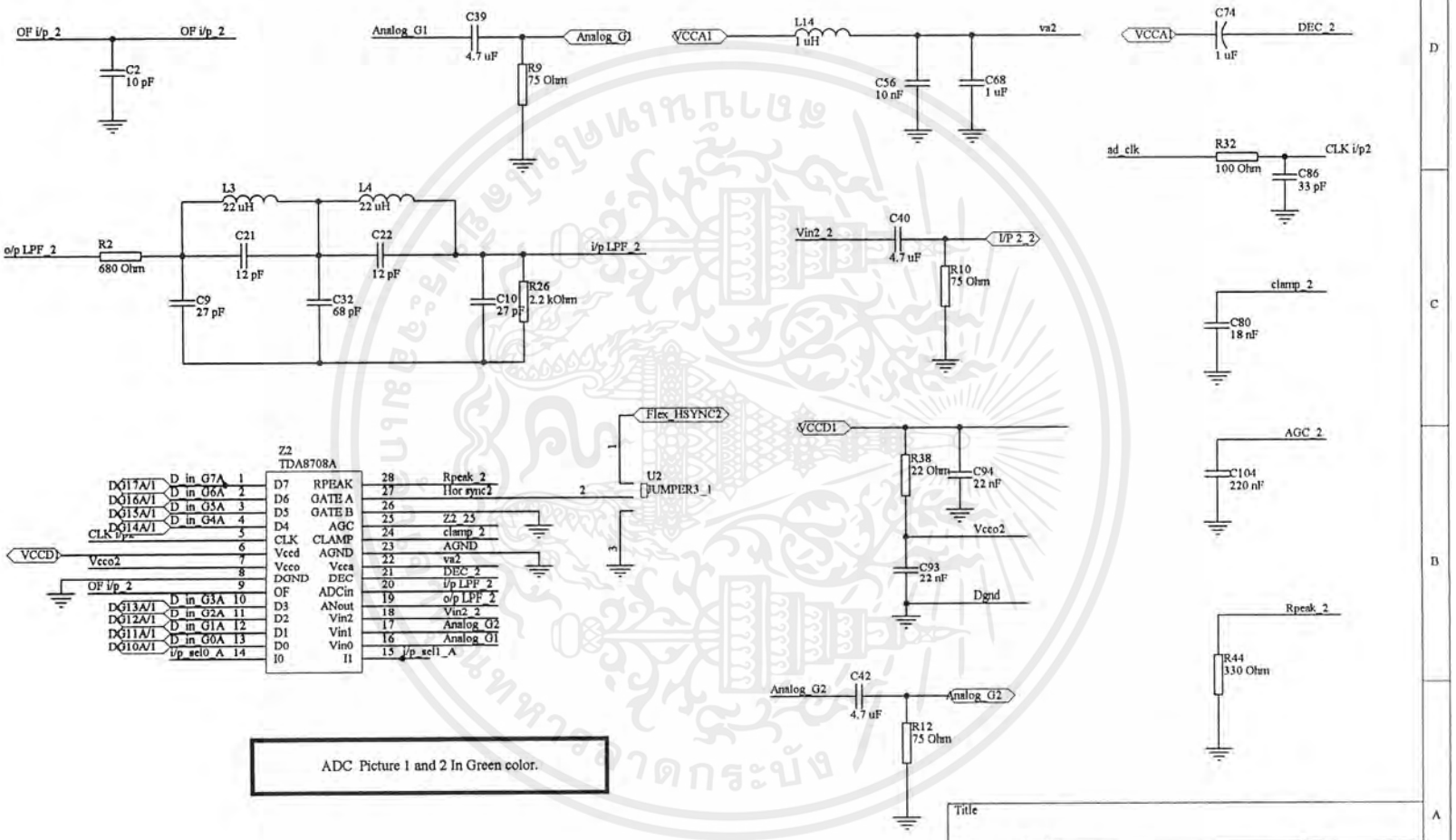
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

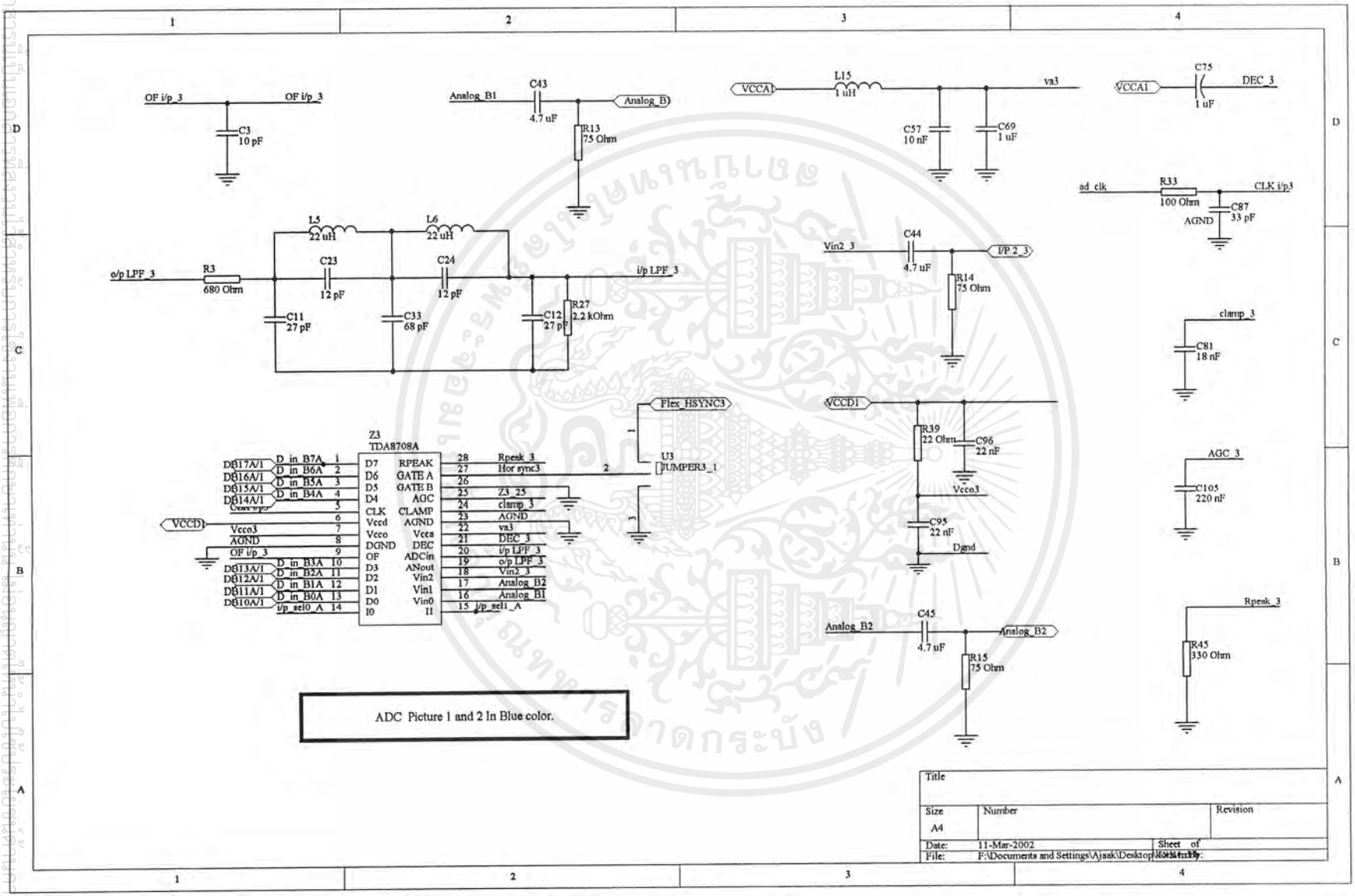


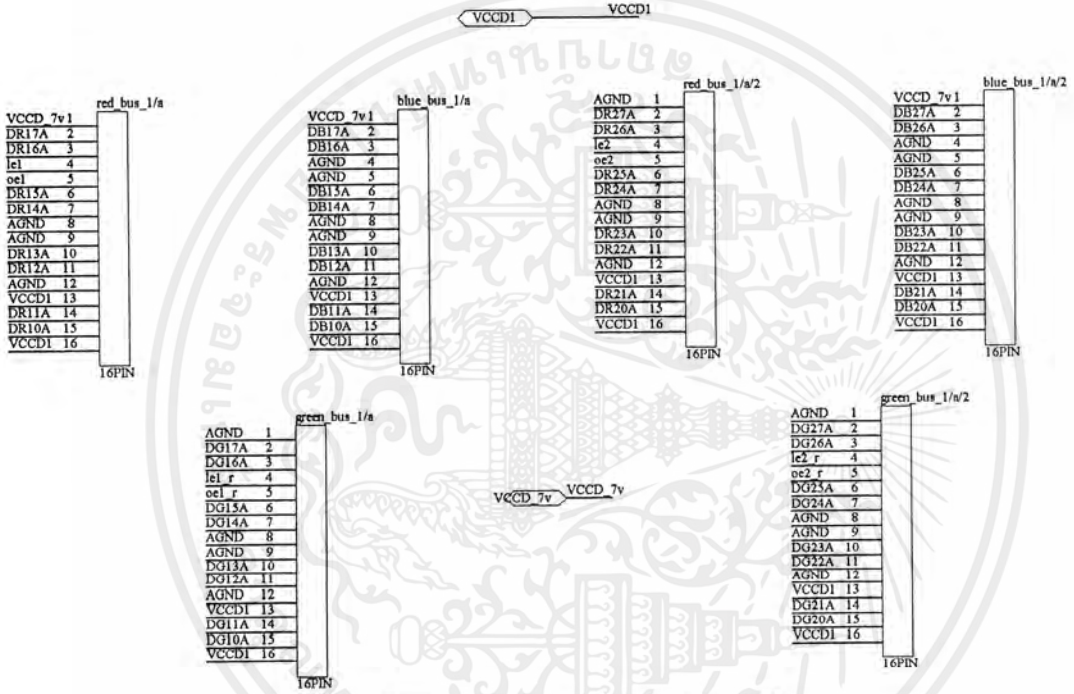
ADC VGA Picture1 & 2 Input port

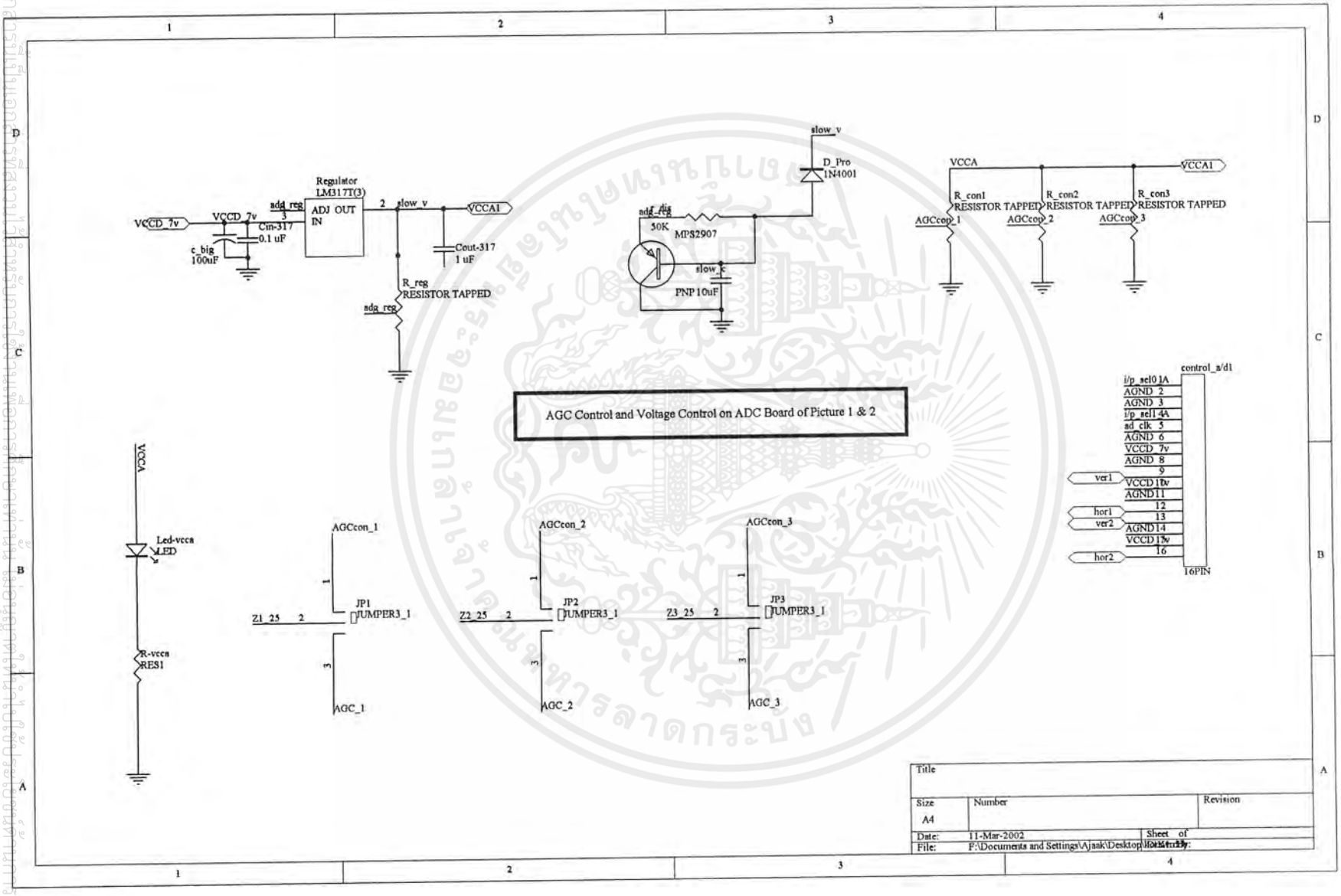
Title		
Size A4	Number	Revision
Date: 11-Mar-2002		Sheet of
File: F:\Documents and Settings\Ajank\Desktop\Horizontal.ddb		Drawn By:

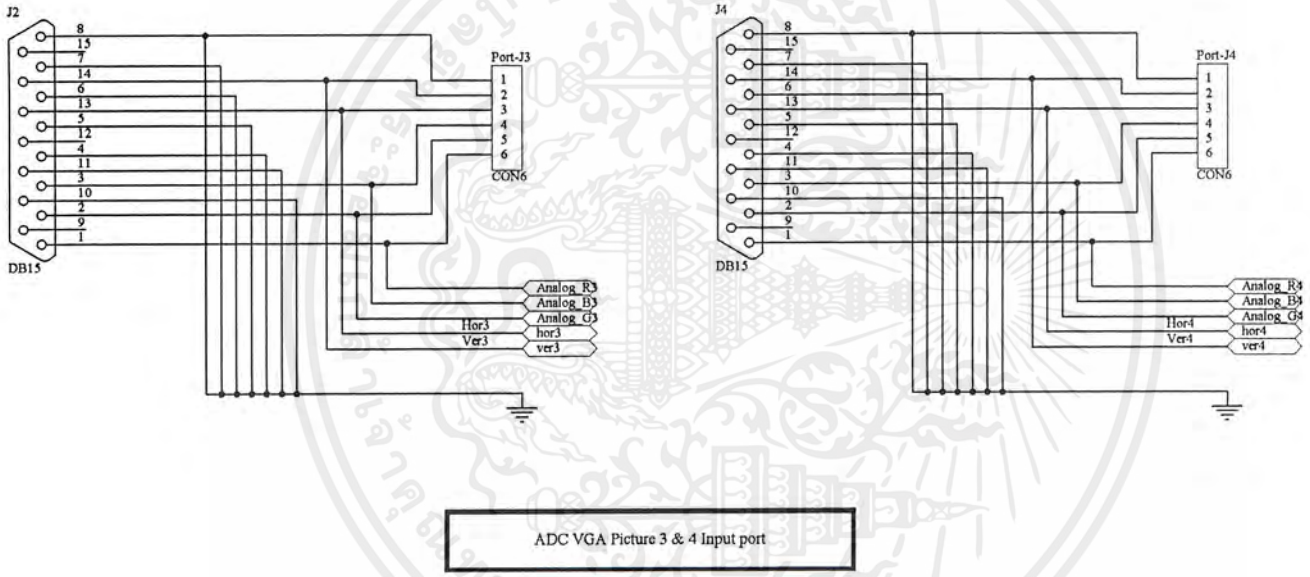

OF i/p_1
C1
10 pF
Analog_R1
C37
4.7 uF
Analog_R1
R7
75 Ohm
VCCA1
L13
1 uH
Val
C55
10 nF
C67
1 uF
VCCA1
C73
1 uF
DEC_1
ad_clk
R31
100 Ohm
CLK i/p1
C85
33 pF
o/p LPF_1
R1
680 Ohm
L1
22 uH
L2
22 uH
C19
12 pF
C20
12 pF
C7
27 pF
C31
68 pF
C8
27 pF
R25
2.2 kOhm
i/p LPF_1
Vin2_1
C38
4.7 uF
I/P 2_1
R8
75 Ohm
clamp_1
C79
18 nF
AGC_1
C103
220 nF
Flex_HSYNC1
U1
JUMPER3_1
Z1
TDA8708A
D_in_R7A
D_in_R6A
D_in_R5A
D_in_R4A
D_in_R3A
D_in_R2A
D_in_R1A
D_in_R0A
D7 RPEAK
D6 GATE A
D5 GATE B
D4 AGC
CLK CLAMP
Vced AGND
Vcco Vccs
DGND DEC
OF ADCin
D3 ANout
D2 Vin2
D1 Vin1
D0 Vin0
I0 I1
Rpeak_1
Hor sync1
Z1_25
clamp_1
Val
DEC_1
i/p LPF_1
o/p LPF_1
Vin2_1
Analog_R2
Analog_R1
i/p_sel1_A
CLK i/p1
VCCD1
Vcco1
OF i/p_1
VCCD1
VCCD1
R37
22 Ohm
C92
22 nF
Vcco1
C91
22 nF
Dgnd
Rpeak_1
R43
330 Ohm
Analog_R2
C41
4.7 uF
Analog_R2
R11
75 Ohm
ADC Picture 1 and 2 In Red color.
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\Untitled.


OF i/p_2
C2
10 pF
Analog_G1
C39
4.7 uF
R9
75 Ohm
VCCA1
L14
1 uH
va2
C56
10 nF
C68
1 uF
C74
1 uF
DEC_2
ad_clk
R32
100 Ohm
CLK i/p2
C86
33 pF
L3
22 uH
L4
22 uH
o/p LPF_2
R2
680 Ohm
C21
12 pF
C22
12 pF
i/p LPF_2
C9
27 pF
C32
68 pF
C10
27 pF
R26
2.2 kOhm
Vin2_2
C40
4.7 uF
I/P 2_2
R10
75 Ohm
clamp_2
C80
18 nF
Z2
TDA8708A
D7
D6
D5
D4
CLK
Vccd
Vcco
DGND
OF
D3
D2
D1
D0
I0
RPEAK
GATE A
GATE B
AGC
CLAMP
AGND
Vcca
DEC
ADCin
ANout
Vin2
Vin1
Vin0
I1
Rpeak_2
Hor sync2
Z2_25
clamp_2
AGND
va2
DEC_2
i/p LPF_2
o/p LPF_2
Vin2_2
Analog_G2
Analog_G1
i/p_sel1_A
D_in_G7A
D_in_G6A
D_in_G5A
D_in_G4A
CLK i/p2
VCCD
Vcco2
OF i/p_2
D_in_G3A
D_in_G2A
D_in_G1A
D_in_G0A
i/p_sel0_A
Flex_HSYNC2
U2
JUMPER3_1
VCCD1
R38
22 Ohm
C94
22 nF
Vcco2
C93
22 nF
Dgnd
AGC_2
C104
220 nF
Rpeak_2
R44
330 Ohm
Analog_G2
C42
4.7 uF
R12
75 Ohm
ADC Picture 1 and 2 In Green color.
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\


OF i/p_3
C3
10 pF
Analog_B1
C43
4.7 uF
Analog_B
R13
75 Ohm
VCCA1
L15
1 uH
va3
C57
10 nF
C69
1 uF
VCCA1
C75
1 uF
DEC_3
ad_clk
R33
100 Ohm
CLK i/p3
AGND
C87
33 pF
L5
22 uH
L6
22 uH
o/p LPF_3
R3
680 Ohm
C23
12 pF
C24
12 pF
i/p LPF_3
C11
27 pF
C33
68 pF
C12
27 pF
R27
2.2 kOhm
Vin2_3
C44
4.7 uF
I/P 2_3
R14
75 Ohm
clamp_3
C81
18 nF
Flex_HSYNC3
U3
JUMPER3_1
VCCD1
R39
22 Ohm
C96
22 nF
Vcco3
C95
22 nF
Dgnd
AGC_3
C105
220 nF
Z3
TDA8708A
D_in_B7A
D_in_B6A
D_in_B5A
D_in_B4A
D_in_B3A
D_in_B2A
D_in_B1A
D_in_B0A
i/p_sel0_A
i/p_sel1_A
VCCD
Vcco3
AGND
OF i/p_3
D7
D6
D5
D4
CLK
Vccd
Vcco
DGND
OF
D3
D2
D1
D0
I0
RPEAK
GATE A
GATE B
AGC
CLAMP
AGND
Vcca
DEC
ADCin
ANout
Vin2
Vin1
Vin0
I1
Rpeak_3
Hor sync3
Z3_25
clamp_3
AGND
va3
DEC_3
i/p LPF_3
o/p LPF_3
Vin2_3
Analog_B2
Analog_B1
i/p_sel1_A
Analog_B2
C45
4.7 uF
Analog_B2
R15
75 Ohm
Rpeak_3
R45
330 Ohm
ADC Picture 1 and 2 In Blue color.
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajsak\Desktop\Work\Z3.ddb
Drawn By:

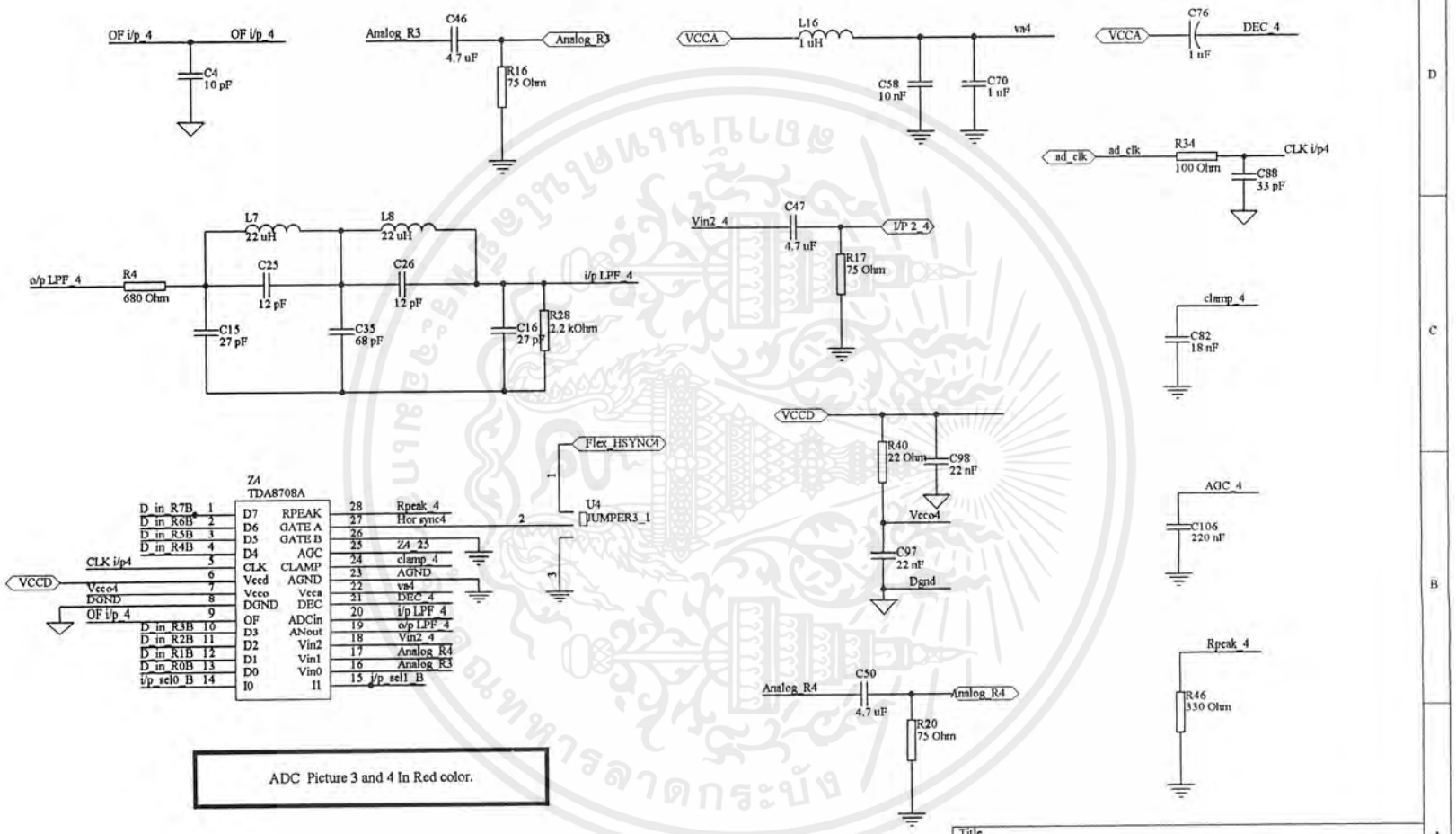

VCCD1
VCCD1
red_bus_1/a
VCCD_7v 1
DR17A 2
DR16A 3
le1 4
oe1 5
DR15A 6
DR14A 7
AGND 8
AGND 9
DR13A 10
DR12A 11
AGND 12
VCCD1 13
DR11A 14
DR10A 15
VCCD1 16
16PIN
blue_bus_1/a
VCCD_7v 1
DB17A 2
DB16A 3
AGND 4
AGND 5
DB15A 6
DB14A 7
AGND 8
AGND 9
DB13A 10
DB12A 11
AGND 12
VCCD1 13
DB11A 14
DB10A 15
VCCD1 16
16PIN
red_bus_1/a/2
AGND 1
DR27A 2
DR26A 3
le2 4
oe2 5
DR25A 6
DR24A 7
AGND 8
AGND 9
DR23A 10
DR22A 11
AGND 12
VCCD1 13
DR21A 14
DR20A 15
VCCD1 16
16PIN
blue_bus_1/a/2
VCCD_7v 1
DB27A 2
DB26A 3
AGND 4
AGND 5
DB25A 6
DB24A 7
AGND 8
AGND 9
DB23A 10
DB22A 11
AGND 12
VCCD1 13
DB21A 14
DB20A 15
VCCD1 16
16PIN
green_bus_1/a
AGND 1
DG17A 2
DG16A 3
le1_r 4
oe1_r 5
DG15A 6
DG14A 7
AGND 8
AGND 9
DG13A 10
DG12A 11
AGND 12
VCCD1 13
DG11A 14
DG10A 15
VCCD1 16
16PIN
VCCD_7v
VCCD_7v
green_bus_1/a/2
AGND 1
DG27A 2
DG26A 3
le2_r 4
oe2_r 5
DG25A 6
DG24A 7
AGND 8
AGND 9
DG23A 10
DG22A 11
AGND 12
VCCD1 13
DG21A 14
DG20A 15
VCCD1 16
16PIN
ADC Connector of Picture 1 & 2 to Control Board
Title
Size A4
Number
Revision
Date: 11-Mar-2002
Sheet of
File: F:\Documents and Settings\Ajaak\Desktop\Untitled
Drawn By:

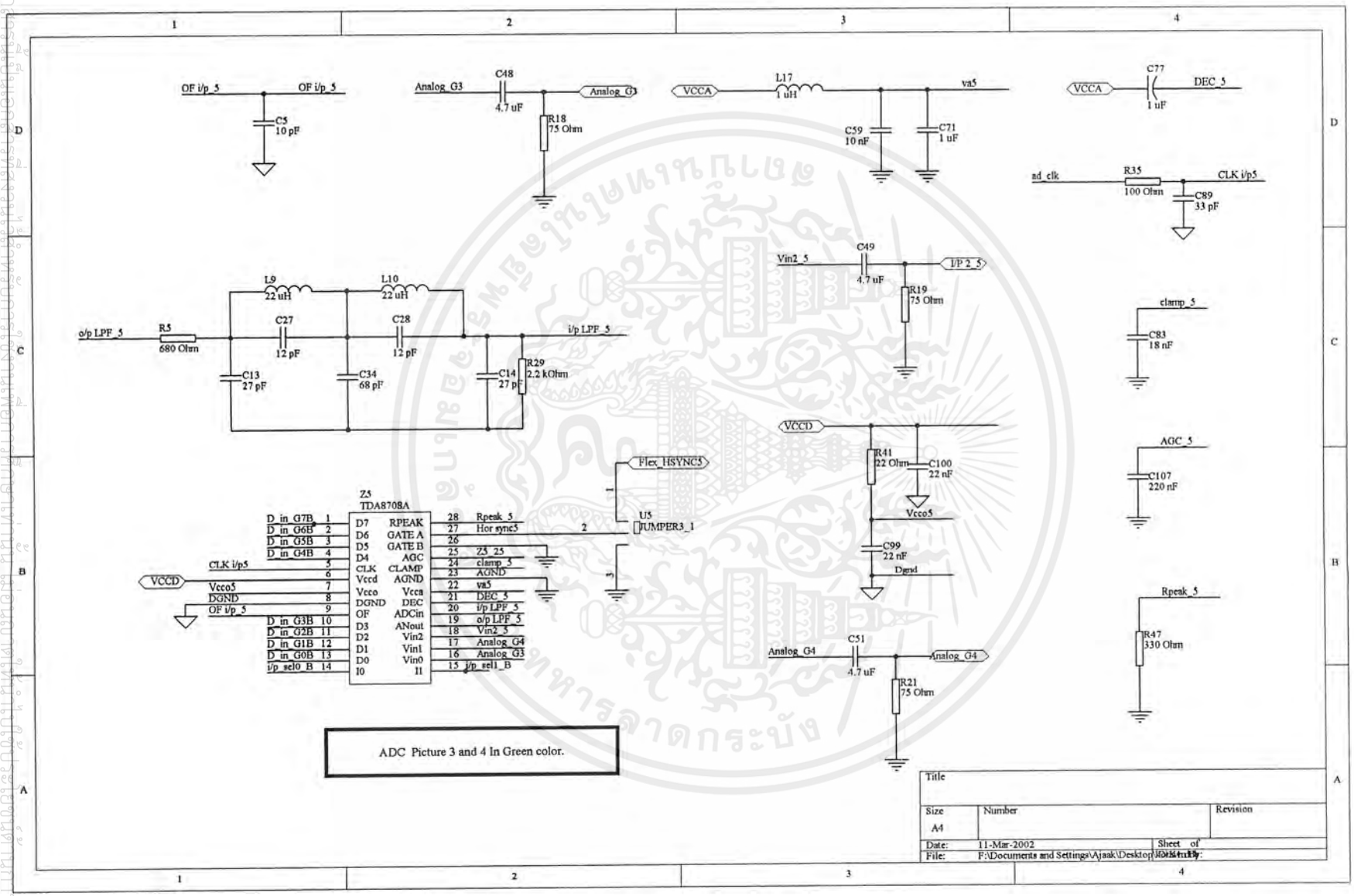

Regulator
LM317T(3)
ADJ OUT
IN
adg_reg
VCCD_7v
Cin-317
0.1 uF
c_big
100uF
slow_v
VCCA1
Cout-317
1 uF
R_reg
RESISTOR TAPPED
adg_reg
D_Pro
1N4001
slow_v
adg_reg
50K
MPS2907
slow_c
PNP 10uF
VCCA
VCCA1
R_con1
RESISTOR TAPPED
R_con2
RESISTOR TAPPED
R_con3
RESISTOR TAPPED
AGCcon_1
AGCcon_2
AGCcon_3
AGC Control and Voltage Control on ADC Board of Picture 1 & 2
control_a/d1
i/p_sel0 1A
AGND 2
AGND 3
i/p_sel1 4A
ad_clk 5
AGND 6
VCCD 7v
AGND 8
9
ver1
VCCD 10v
AGND 11
hor1
12
ver2
13
AGND 14
VCCD 15v
hor2
16
16PIN
VCCA
Led-vcca
LED
R-vcca
RES1
JP1
JUMPER3_1
JP2
JUMPER3_1
JP3
JUMPER3_1
Z1_25
Z2_25
Z3_25
AGC_1
AGC_2
AGC_3
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\Horizontal.ddb
Drawn By:

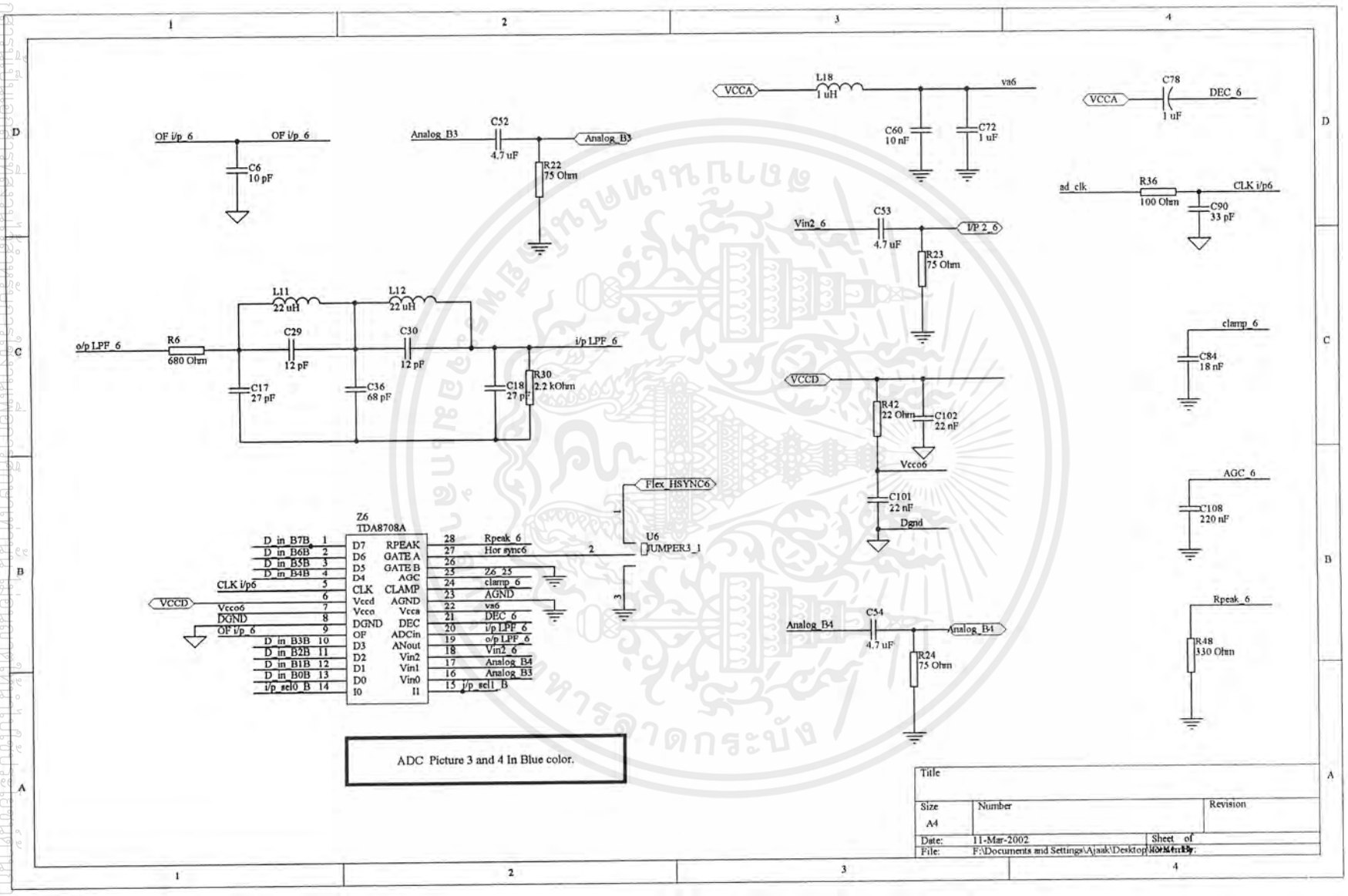


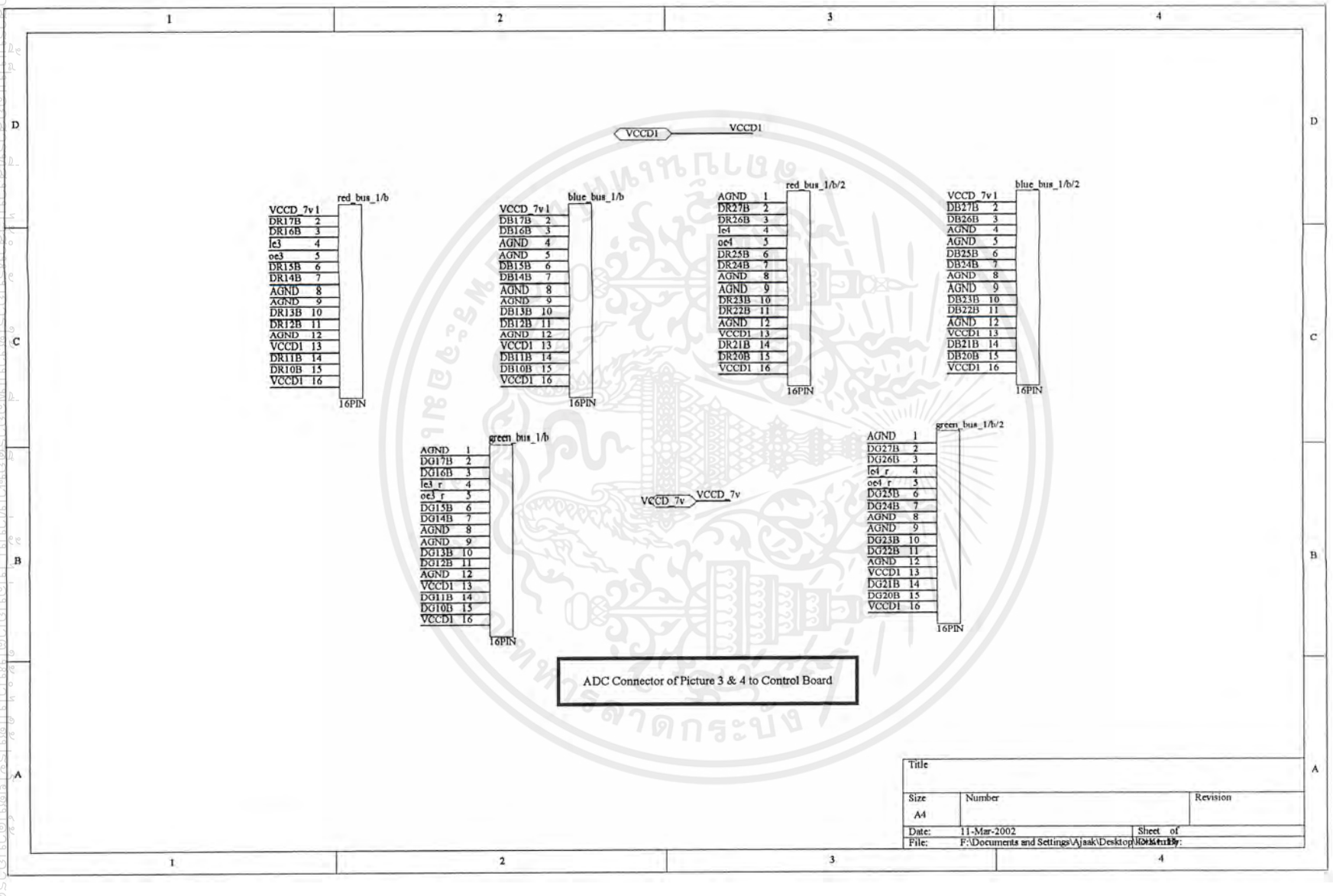
ADC VGA Picture 3 & 4 Input port

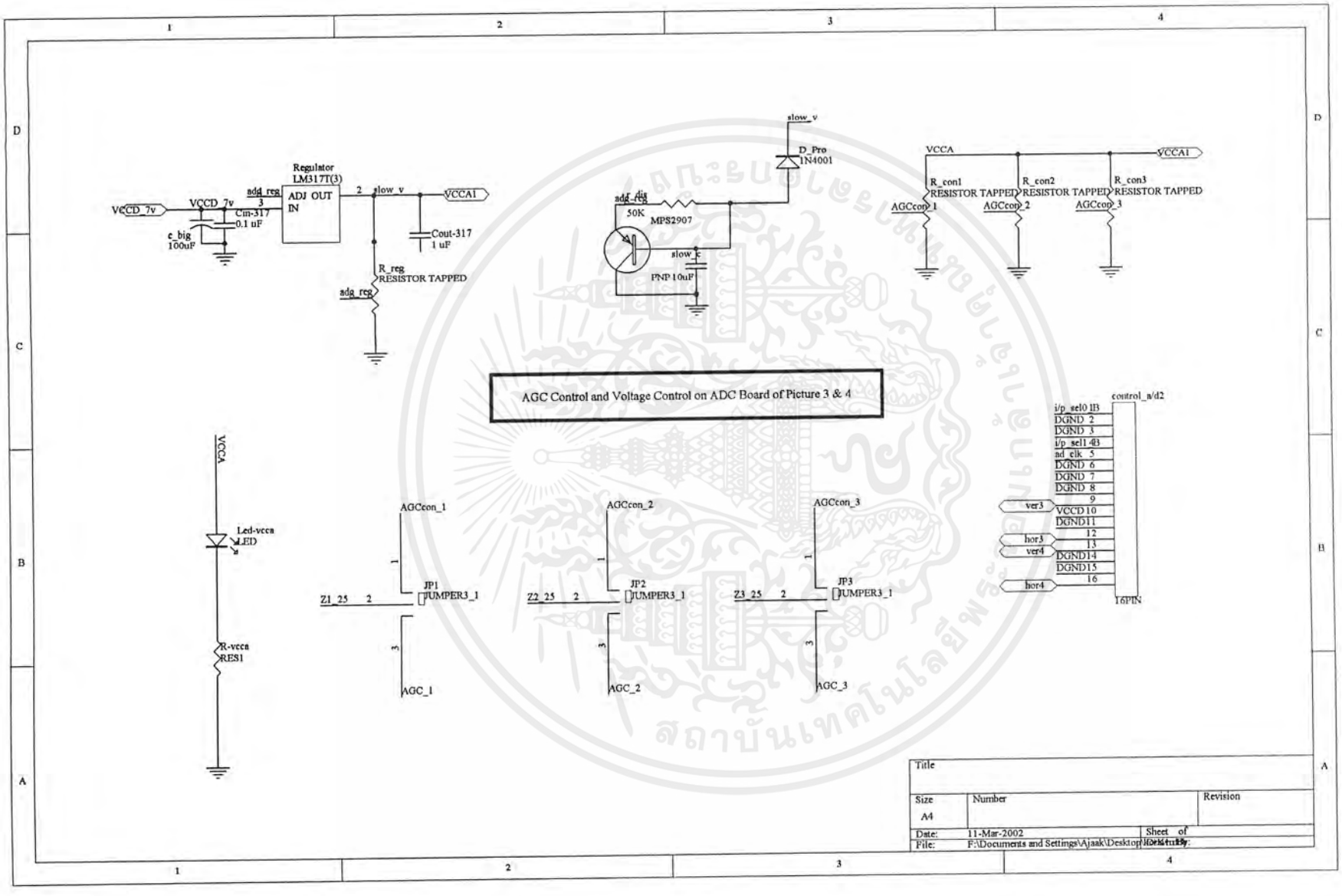
Title		
Size A4	Number	Revision
Date: 11-Mar-2002	Sheet of	
File: F:\Documents and Settings\Ajaak\Desktop\[illegible]	Drawn By:	


OF i/p_4
OF i/p_4
C4
10 pF
Analog_R3
C46
4.7 uF
Analog_R3
R16
75 Ohm
VCCA
L16
1 uH
va4
C58
10 nF
C70
1 uF
VCCA
C76
1 uF
DEC_4
ad_clk
ad_clk
R34
100 Ohm
CLK i/p4
C88
33 pF
L7
22 uH
L8
22 uH
o/p LPF_4
R4
680 Ohm
C25
12 pF
C26
12 pF
i/p LPF_4
C15
27 pF
C35
68 pF
C16
27 pF
R28
2.2 kOhm
Vin2_4
C47
4.7 uF
I/P 2_4
R17
75 Ohm
clamp_4
C82
18 nF
VCCD
R40
22 Ohm
C98
22 nF
Vcco4
C97
22 nF
Dgnd
AGC_4
C106
220 nF
Flex_HSYNC4
U4
JUMPER3_1
Z4
TDA8708A
D_in_R7B
D_in_R6B
D_in_R5B
D_in_R4B
CLK i/p4
VCCD
Vcco4
DGND
OF i/p_4
D_in_R3B
D_in_R2B
D_in_R1B
D_in_R0B
i/p_sel0_B
D7 RPEAK
D6 GATE A
D5 GATE B
D4 AGC
CLK CLAMP
Vccd AGND
Vcco Vcca
DGND DEC
OF ADCin
D3 ANout
D2 Vin2
D1 Vin1
D0 Vin0
I0 I1
Rpeak_4
Hor sync4
Z4_25
clamp_4
AGND
va4
DEC_4
i/p LPF_4
o/p LPF_4
Vin2_4
Analog_R4
Analog_R3
i/p_sel1_B
Analog_R4
C50
4.7 uF
Analog_R4
R20
75 Ohm
Rpeak_4
R46
330 Ohm
ADC Picture 3 and 4 In Red color.
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\Untitled

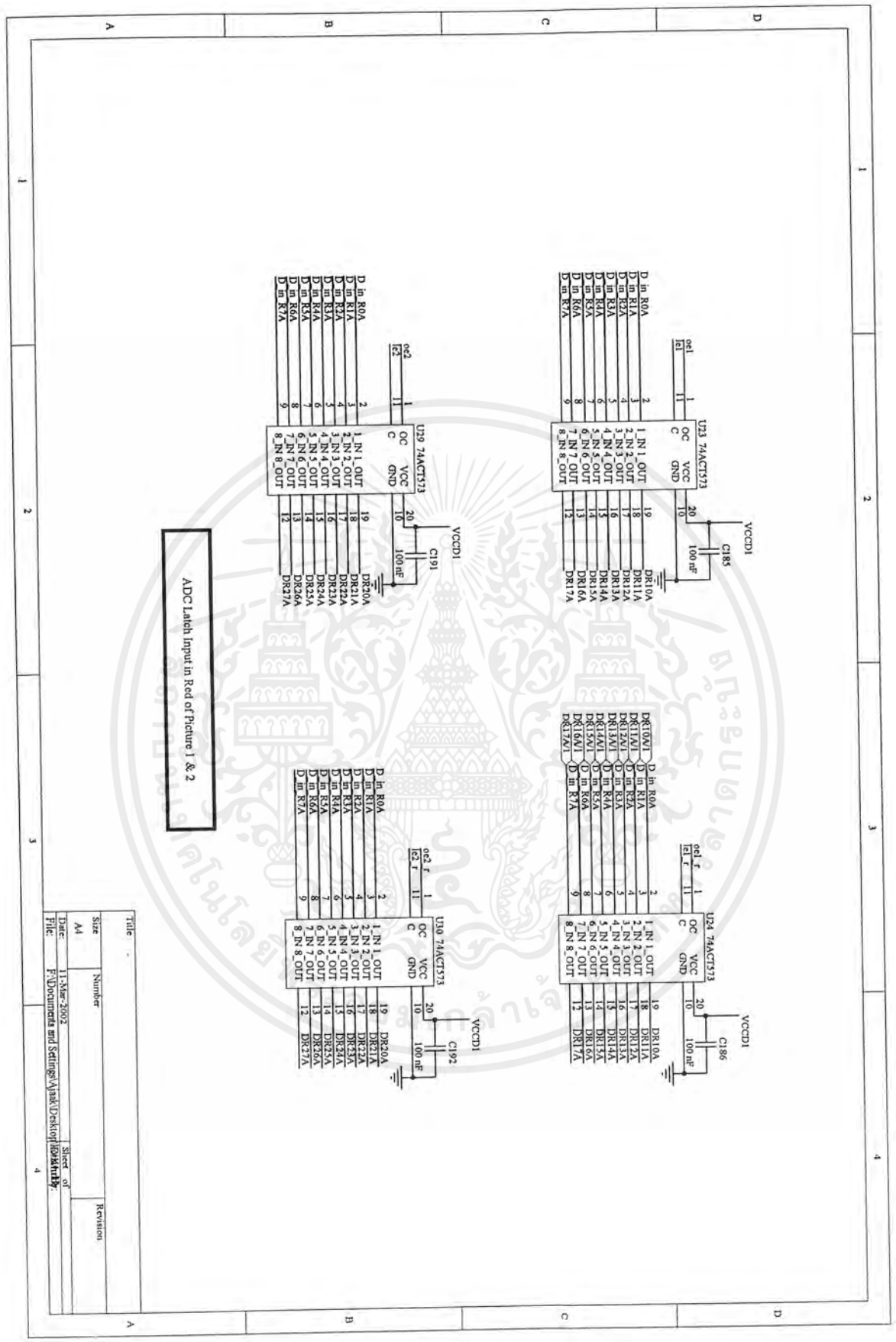

OF i/p_5
OF i/p_5
C5
10 pF
Analog_G3
C48
4.7 uF
Analog_G3
R18
75 Ohm
VCCA
L17
1 uH
va5
C59
10 nF
C71
1 uF
VCCA
C77
1 uF
DEC_5
ad_clk
R35
100 Ohm
CLK i/p5
C89
33 pF
L9
22 uH
L10
22 uH
o/p LPF_5
R5
680 Ohm
C27
12 pF
C28
12 pF
i/p LPF_5
C13
27 pF
C34
68 pF
C14
27 pF
R29
2.2 kOhm
Vin2_5
C49
4.7 uF
I/P 2_5
R19
75 Ohm
clamp_5
C83
18 nF
VCCD
R41
22 Ohm
C100
22 nF
Vcco5
C99
22 nF
Dgnd
AGC_5
C107
220 nF
Flex_HSYNC5
U5
JUMPER3_1
Z5
TDA8708A
D_in_G7B 1
D_in_G6B 2
D_in_G5B 3
D_in_G4B 4
CLK i/p5 5
6
VCCD
Vcco5 7
DGND 8
OF i/p_5 9
D_in_G3B 10
D_in_G2B 11
D_in_G1B 12
D_in_G0B 13
i/p_sel0_B 14
D7 RPEAK
D6 GATE A
D5 GATE B
D4 AGC
CLK CLAMP
Vccd AGND
Vcco Vcca
DGND DEC
OF ADCin
D3 ANout
D2 Vin2
D1 Vin1
D0 Vin0
I0 I1
28 Rpeak_5
27 Hor sync5
26
25 Z5_25
24 clamp_5
23 AGND
22 va5
21 DEC_5
20 i/p LPF_5
19 o/p LPF_5
18 Vin2_5
17 Analog_G4
16 Analog_G3
15 i/p_sel1_B
Rpeak_5
R47
330 Ohm
Analog_G4
C51
4.7 uF
Analog_G4
R21
75 Ohm
ADC Picture 3 and 4 In Green color.
Title
Size
A4
Number
Revision
Date: 11-Mar-2002
Sheet of
File: F:\Documents and Settings\Ajaak\Desktop\


VCCA
L18
1 uH
va6
C60
10 nF
C72
1 uF
C78
DEC_6
1 uF
OF i/p_6
C6
10 pF
Analog_B3
C52
4.7 uF
R22
75 Ohm
ad_clk
R36
100 Ohm
CLK i/p6
C90
33 pF
Vin2_6
C53
4.7 uF
I/P 2_6
R23
75 Ohm
L11
22 uH
L12
22 uH
o/p LPF_6
R6
680 Ohm
C29
12 pF
C30
12 pF
i/p LPF_6
C17
27 pF
C36
68 pF
C18
27 pF
R30
2.2 kOhm
clamp_6
C84
18 nF
VCCD
R42
22 Ohm
C102
22 nF
Vcco6
C101
22 nF
Dgnd
AGC_6
C108
220 nF
Flex_HSYNC6
U6
JUMPER3_1
Z6
TDA8708A
D_in_B7B
D_in_B6B
D_in_B5B
D_in_B4B
CLK i/p6
VCCD
Vcco6
DGND
OF i/p_6
D_in_B3B
D_in_B2B
D_in_B1B
D_in_B0B
i/p_sel0_B
D7 RPEAK
D6 GATE A
D5 GATE B
D4 AGC
CLK CLAMP
Vccd AGND
Vcca Vcca
DGND DEC
OF ADCin
D3 ANout
D2 Vin2
D1 Vin1
D0 Vin0
I0 I1
Rpeak_6
Hor sync6
Z6_25
clamp_6
AGND
va6
DEC_6
i/p LPF_6
o/p LPF_6
Vin2_6
Analog_B4
Analog_B3
i/p_sel1_B
Analog_B4
C54
4.7 uF
R24
75 Ohm
Rpeak_6
R48
330 Ohm
ADC Picture 3 and 4 In Blue color.
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\ADC.ddb
Drawn By:

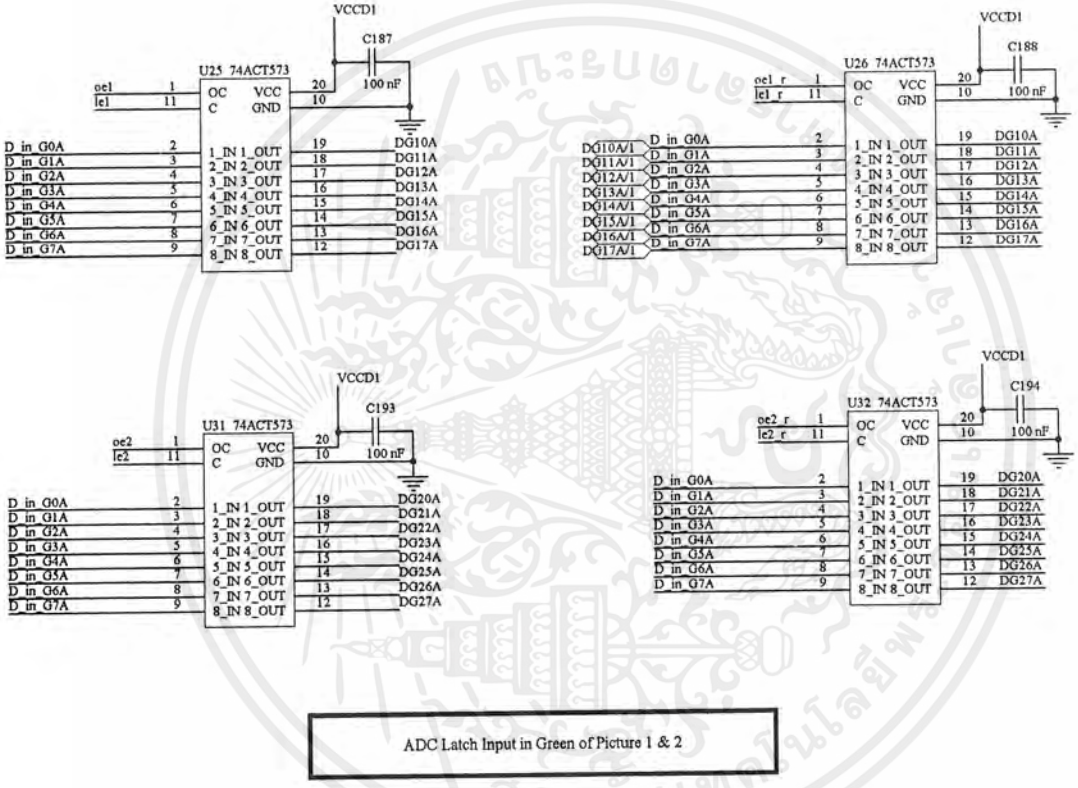

VCCD1
VCCD1
red_bus_1/b
VCCD_7v 1
DR17B 2
DR16B 3
le3 4
oe3 5
DR15B 6
DR14B 7
AGND 8
AGND 9
DR13B 10
DR12B 11
AGND 12
VCCD1 13
DR11B 14
DR10B 15
VCCD1 16
16PIN
blue_bus_1/b
VCCD_7v 1
DB17B 2
DB16B 3
AGND 4
AGND 5
DB15B 6
DB14B 7
AGND 8
AGND 9
DB13B 10
DB12B 11
AGND 12
VCCD1 13
DB11B 14
DB10B 15
VCCD1 16
16PIN
red_bus_1/b/2
AGND 1
DR27B 2
DR26B 3
le4 4
oe4 5
DR25B 6
DR24B 7
AGND 8
AGND 9
DR23B 10
DR22B 11
AGND 12
VCCD1 13
DR21B 14
DR20B 15
VCCD1 16
16PIN
blue_bus_1/b/2
VCCD_7v 1
DB27B 2
DB26B 3
AGND 4
AGND 5
DB25B 6
DB24B 7
AGND 8
AGND 9
DB23B 10
DB22B 11
AGND 12
VCCD1 13
DB21B 14
DB20B 15
VCCD1 16
16PIN
green_bus_1/b
AGND 1
DG17B 2
DG16B 3
le3_r 4
oe3_r 5
DG15B 6
DG14B 7
AGND 8
AGND 9
DG13B 10
DG12B 11
AGND 12
VCCD1 13
DG11B 14
DG10B 15
VCCD1 16
16PIN
VCCD_7v
VCCD_7v
green_bus_1/b/2
AGND 1
DG27B 2
DG26B 3
le4_r 4
oe4_r 5
DG25B 6
DG24B 7
AGND 8
AGND 9
DG23B 10
DG22B 11
AGND 12
VCCD1 13
DG21B 14
DG20B 15
VCCD1 16
16PIN
ADC Connector of Picture 3 & 4 to Control Board
Title
Size A4
Number
Revision
Date: 11-Mar-2002
Sheet of
File: F:\Documents and Settings\Ajaak\Desktop\Drawing1.ddb
Drawn By:



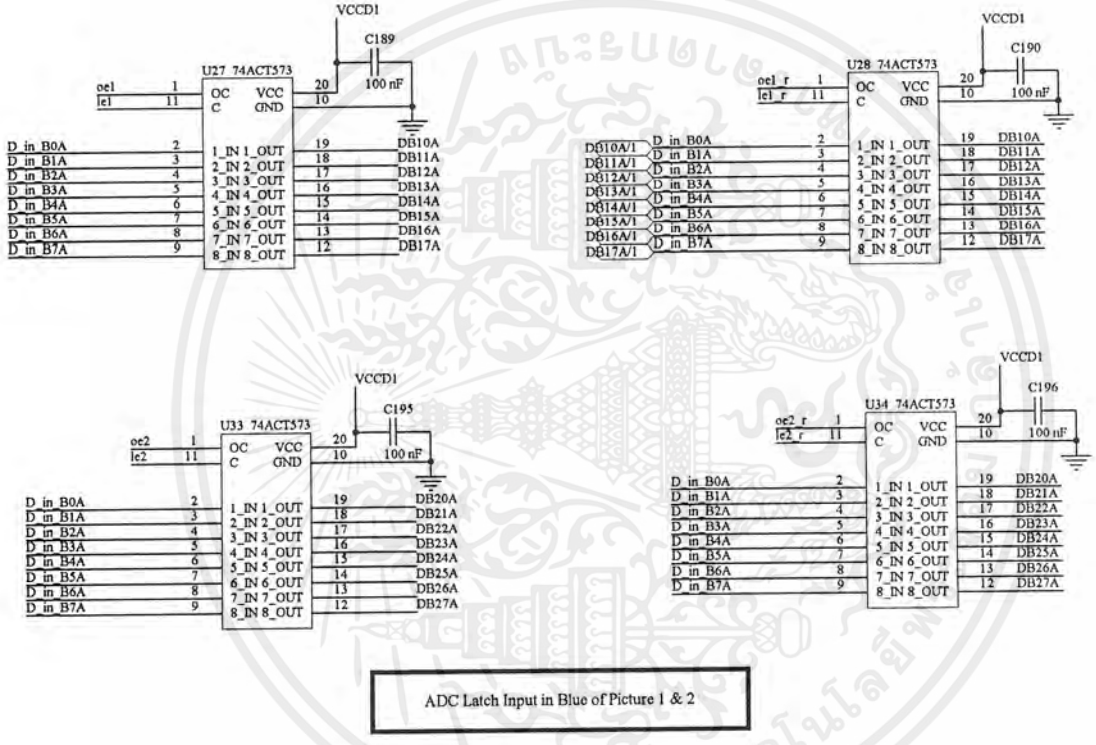
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



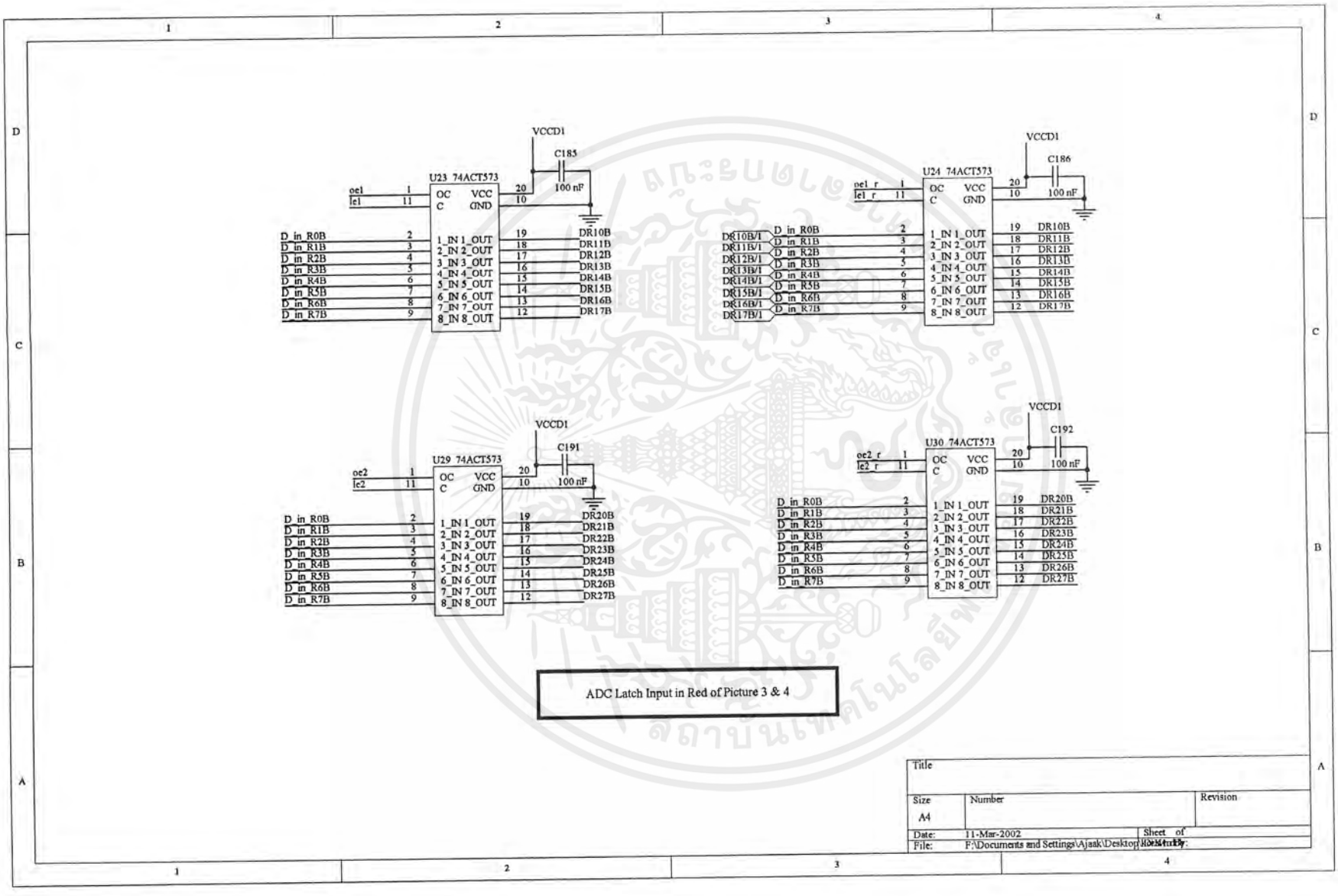
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



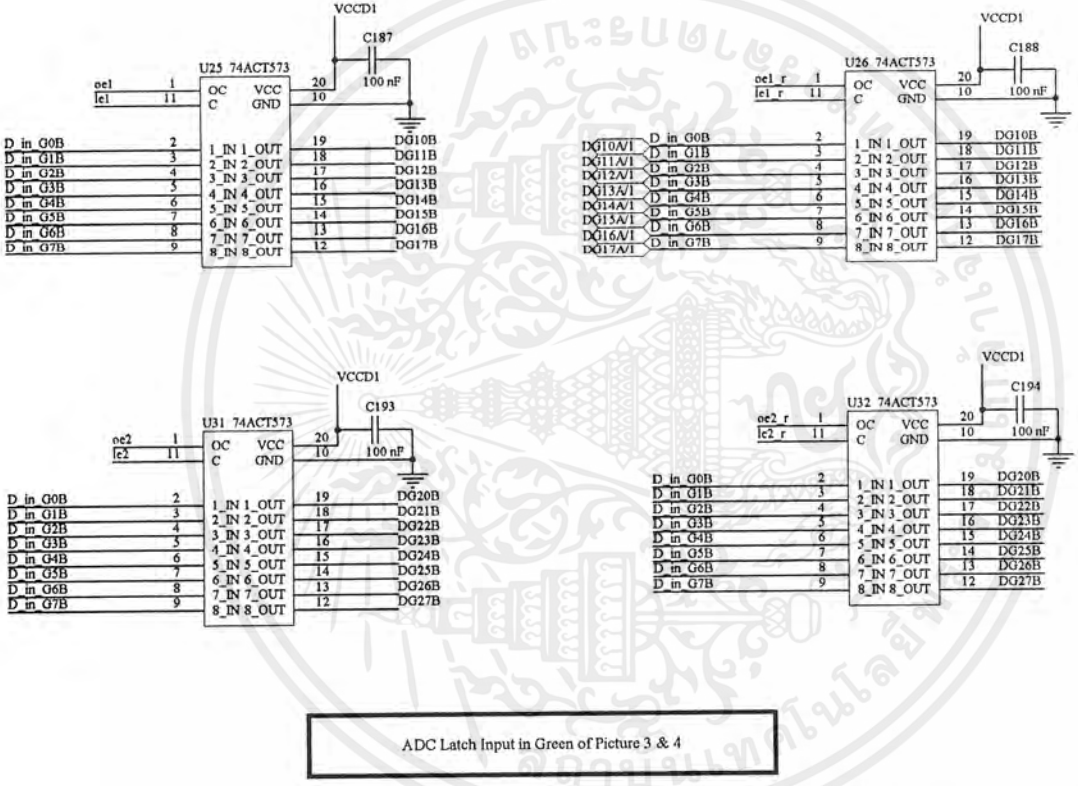
ADC Latch Input in Blue of Picture 1 & 2

Title		
Size: A4	Number	Revision
Date: 11-Mar-2002	Sheet of	
File: F:\Documents and Settings\Ajaak\Desktop\Mosaic.ddb	Drawn By:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



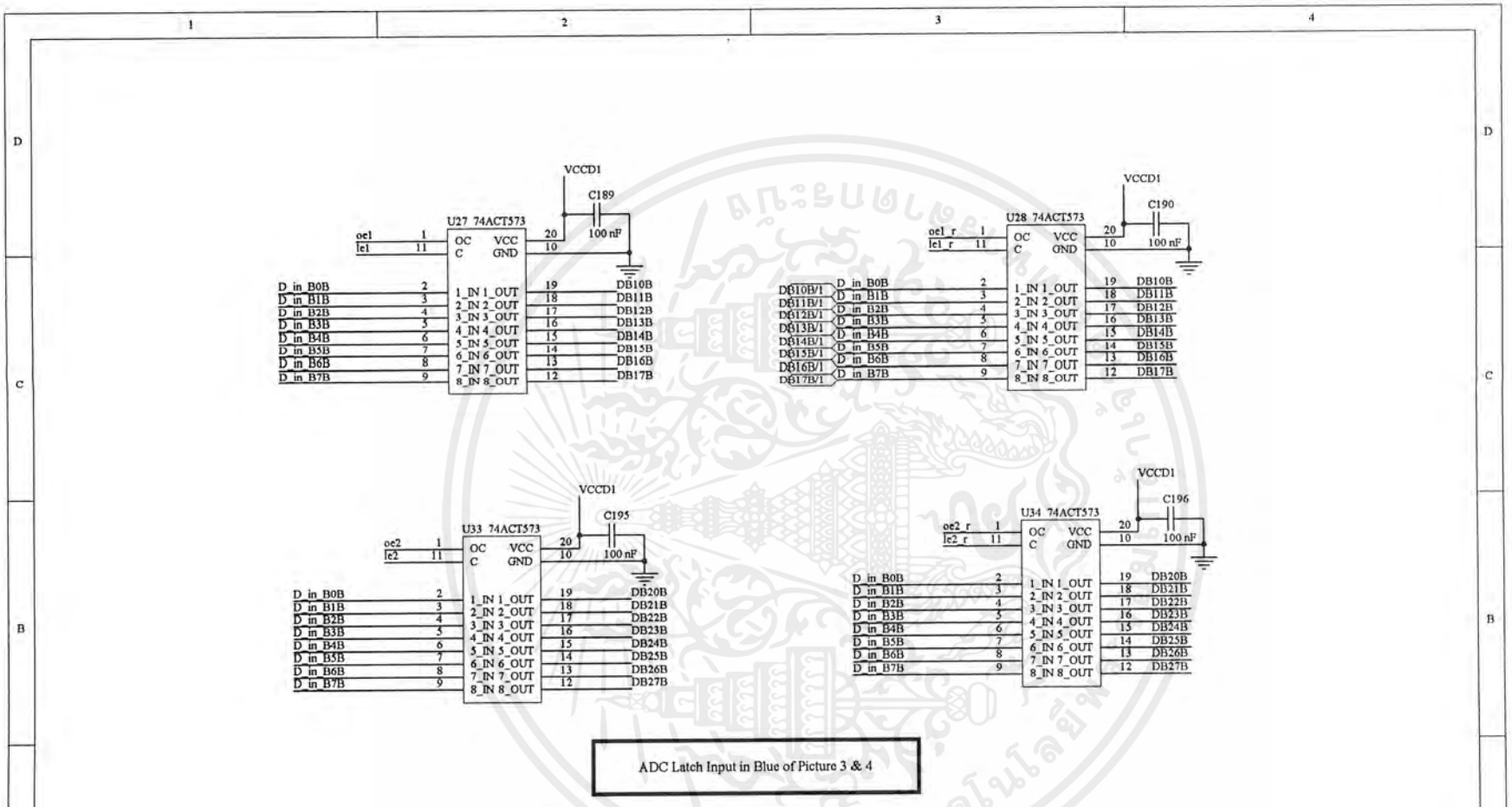
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



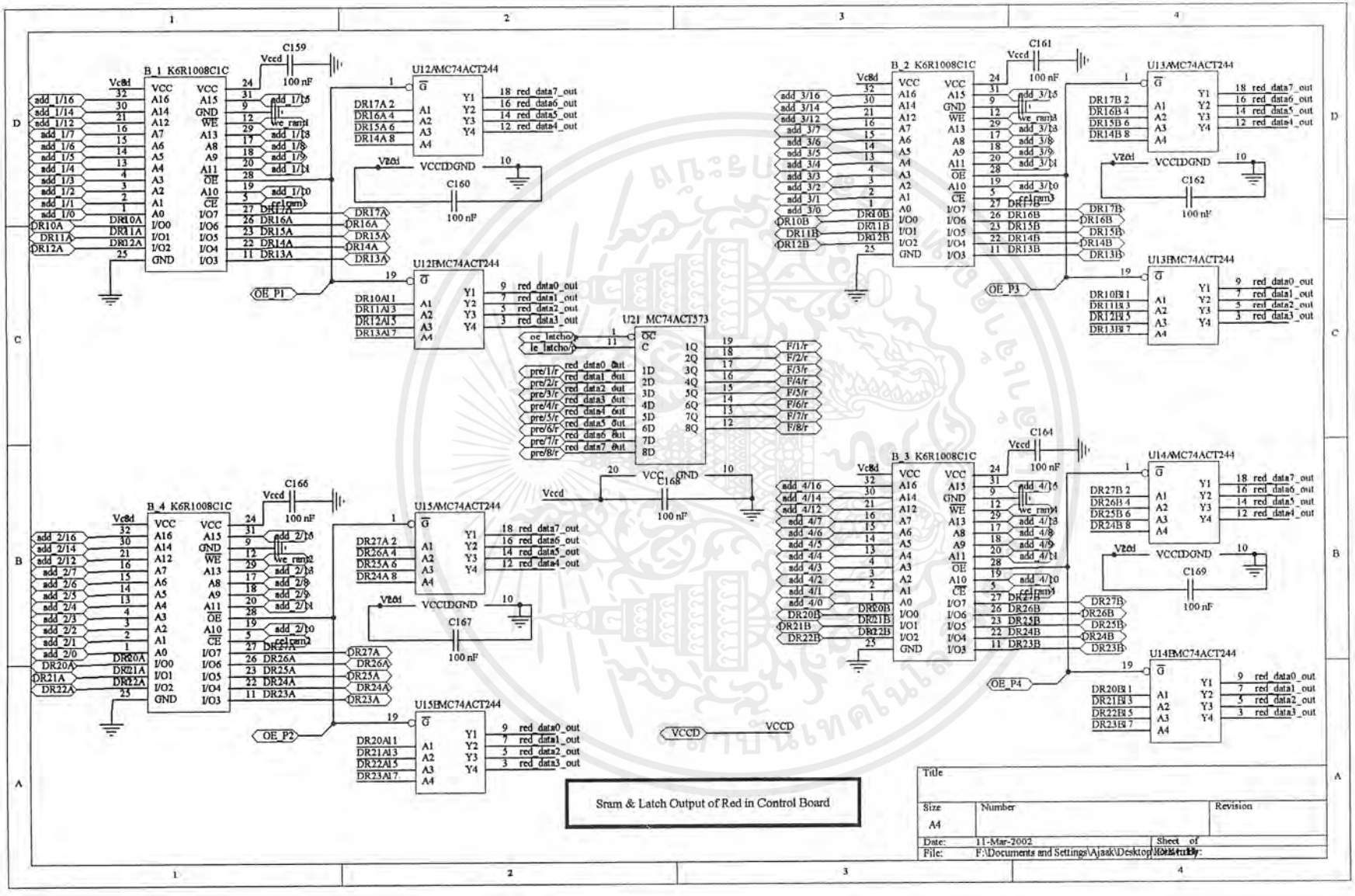
ADC Latch Input in Green of Picture 3 & 4

Title		
Size: A4	Number	Revision
Date: 11-Mar-2002	Sheet of	
File: F:\Documents and Settings\Ajaak\Desktop\Untitled.ddb	Drawn By:	

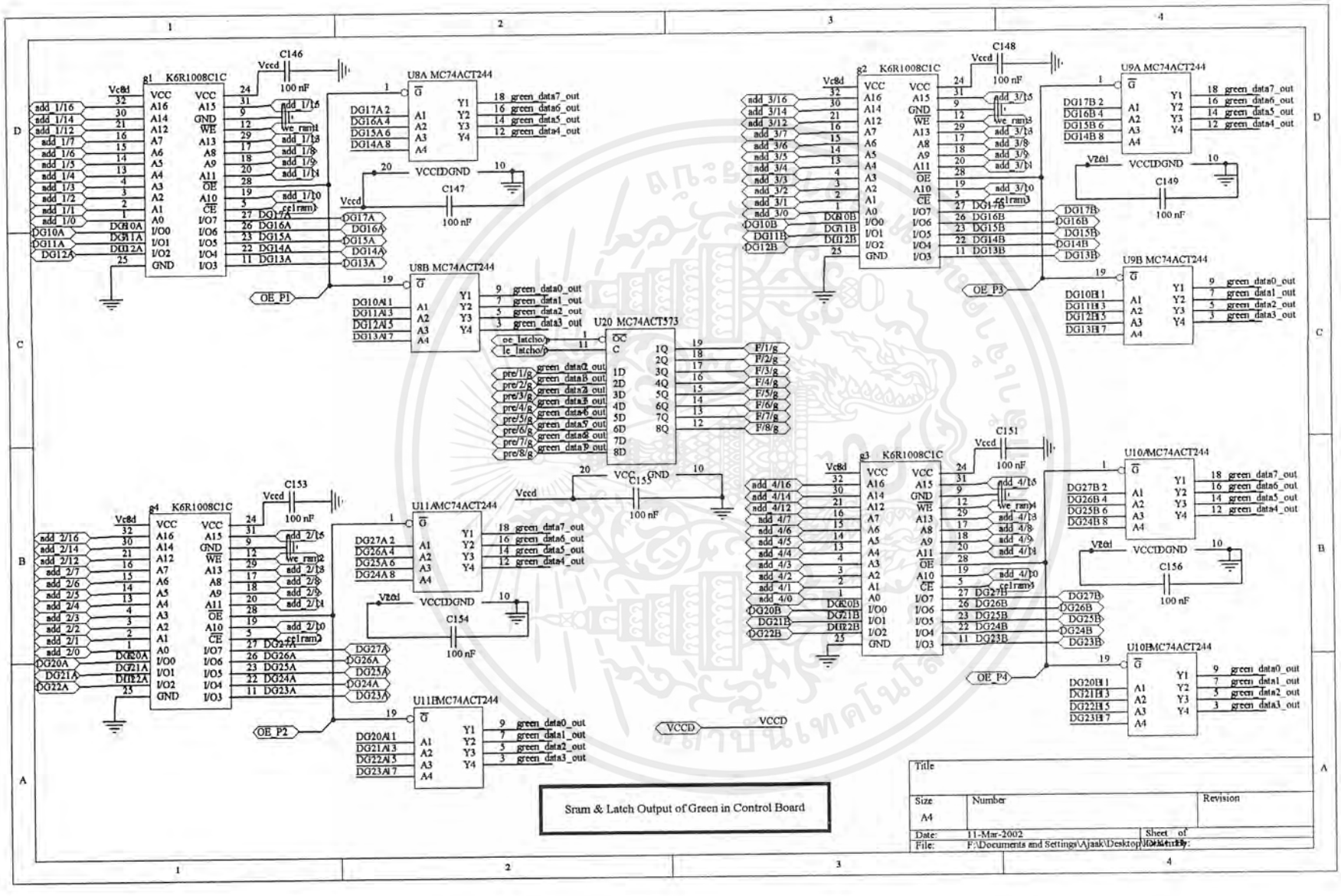
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



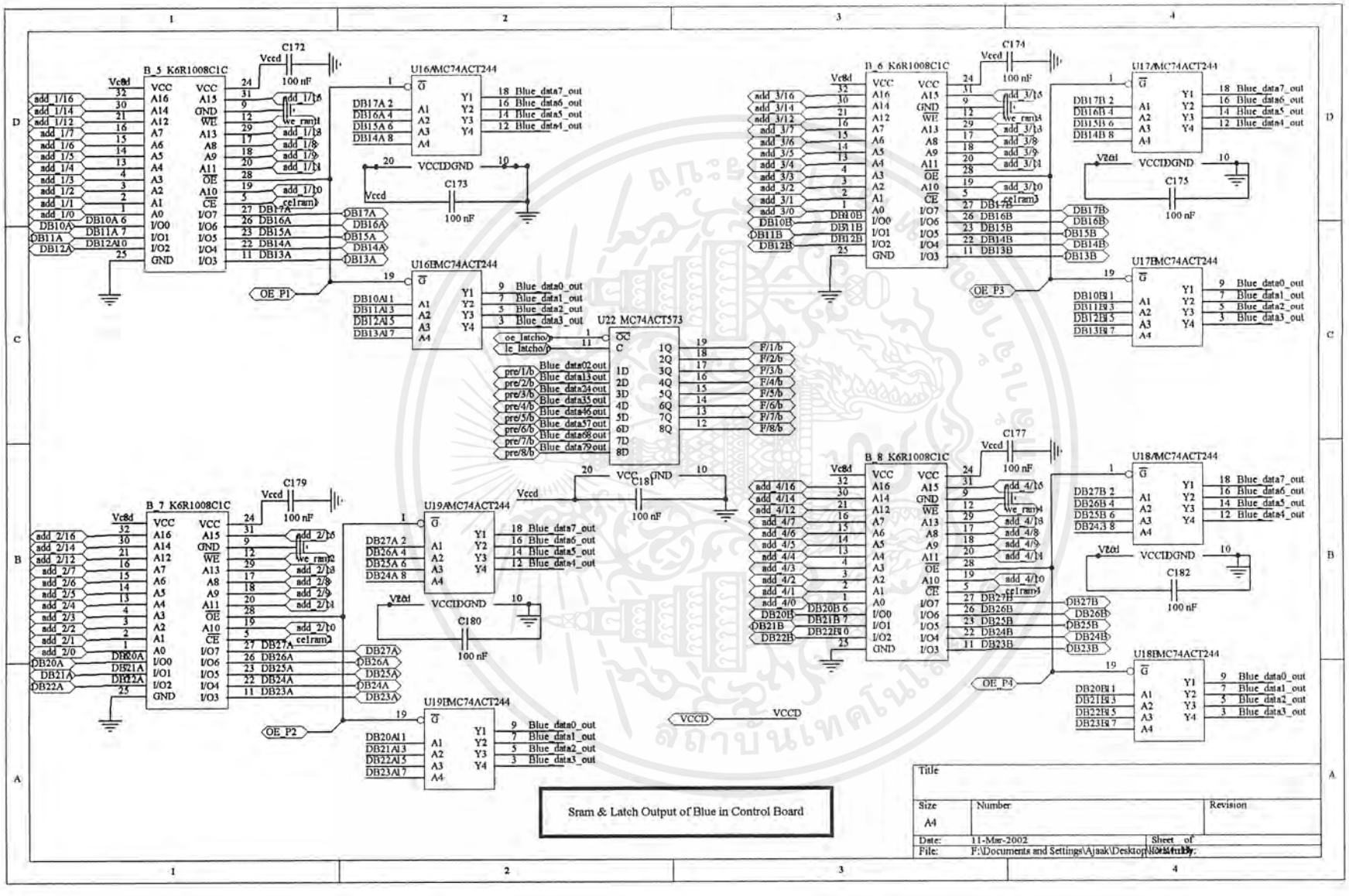
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



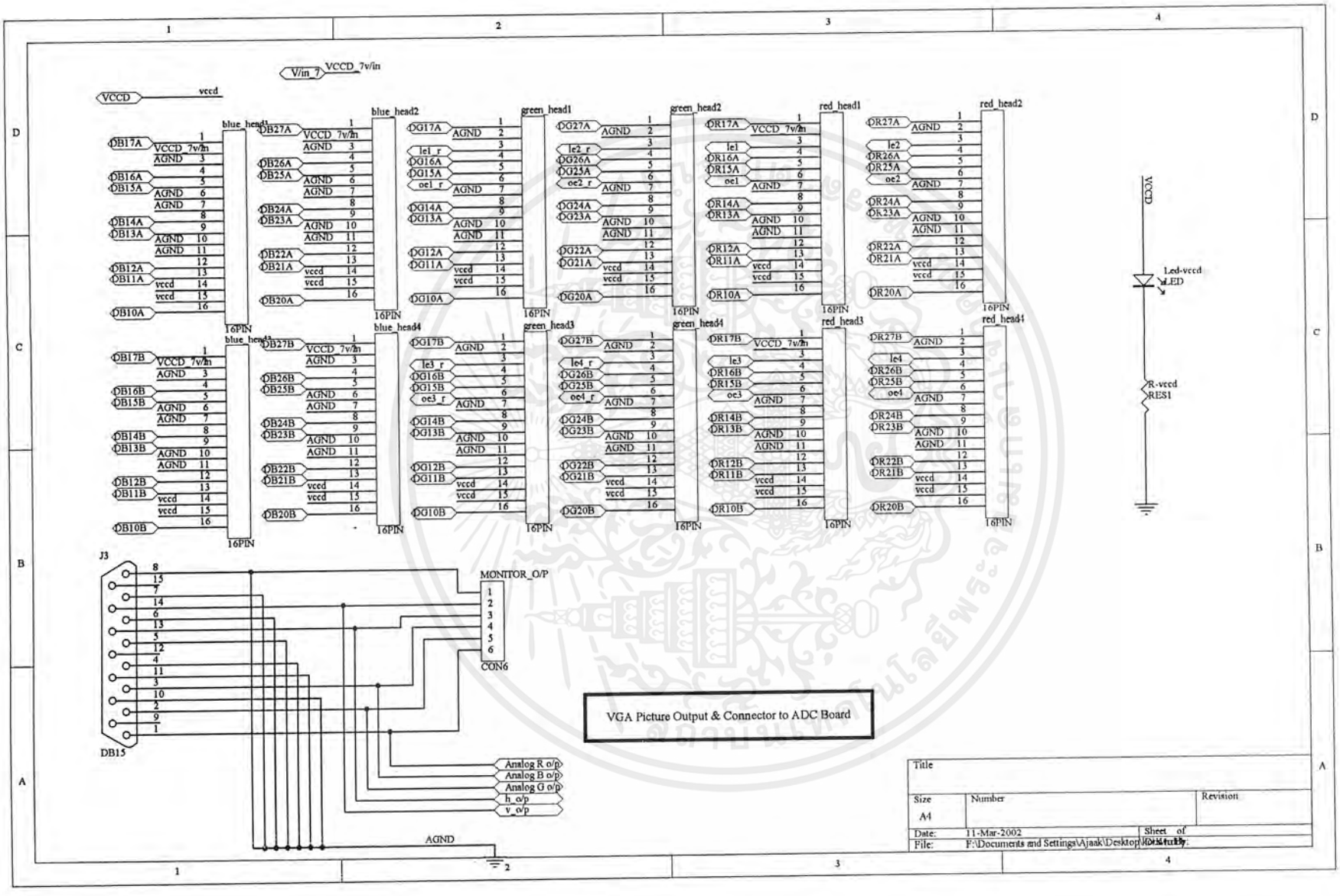
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

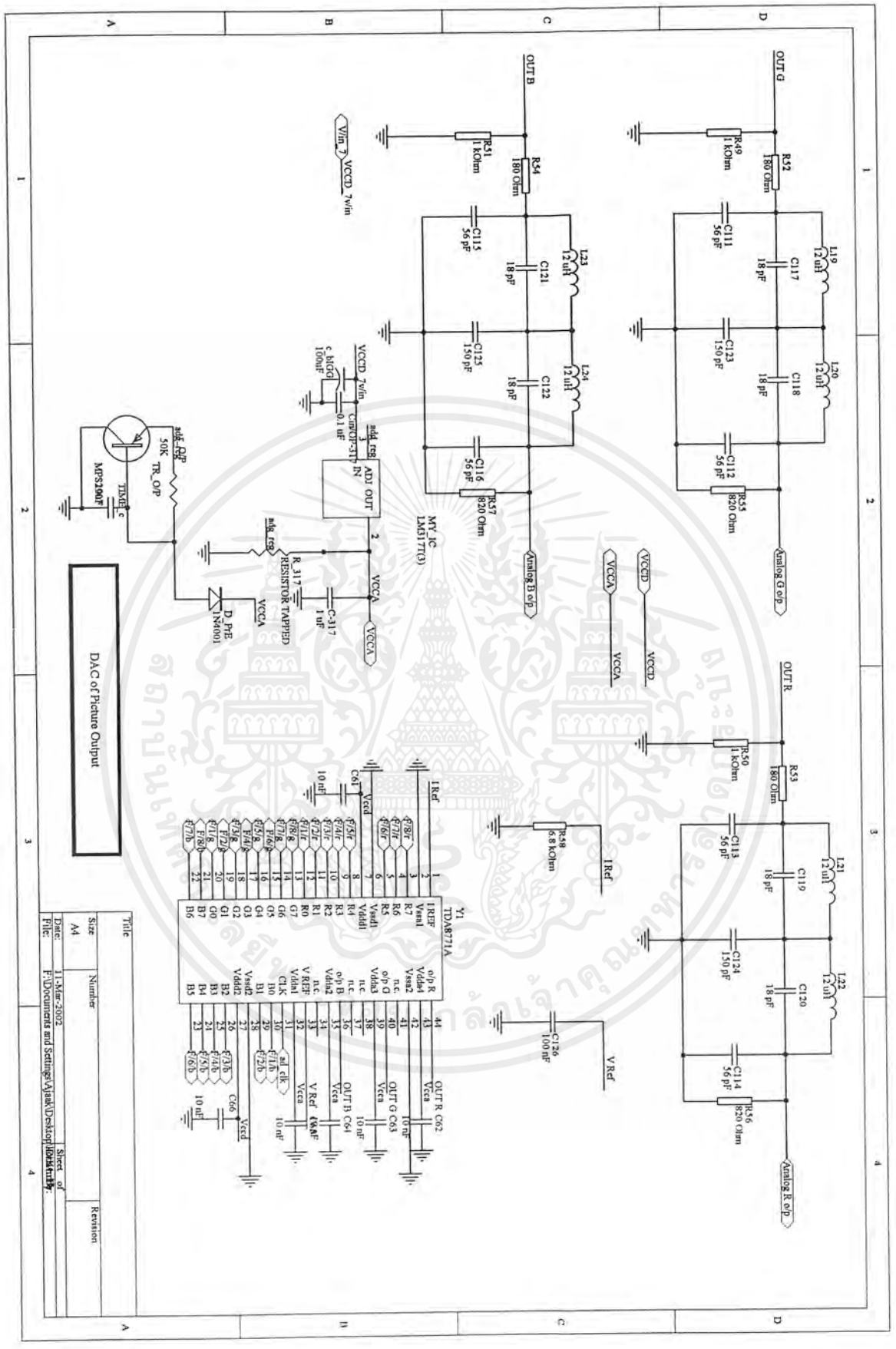


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

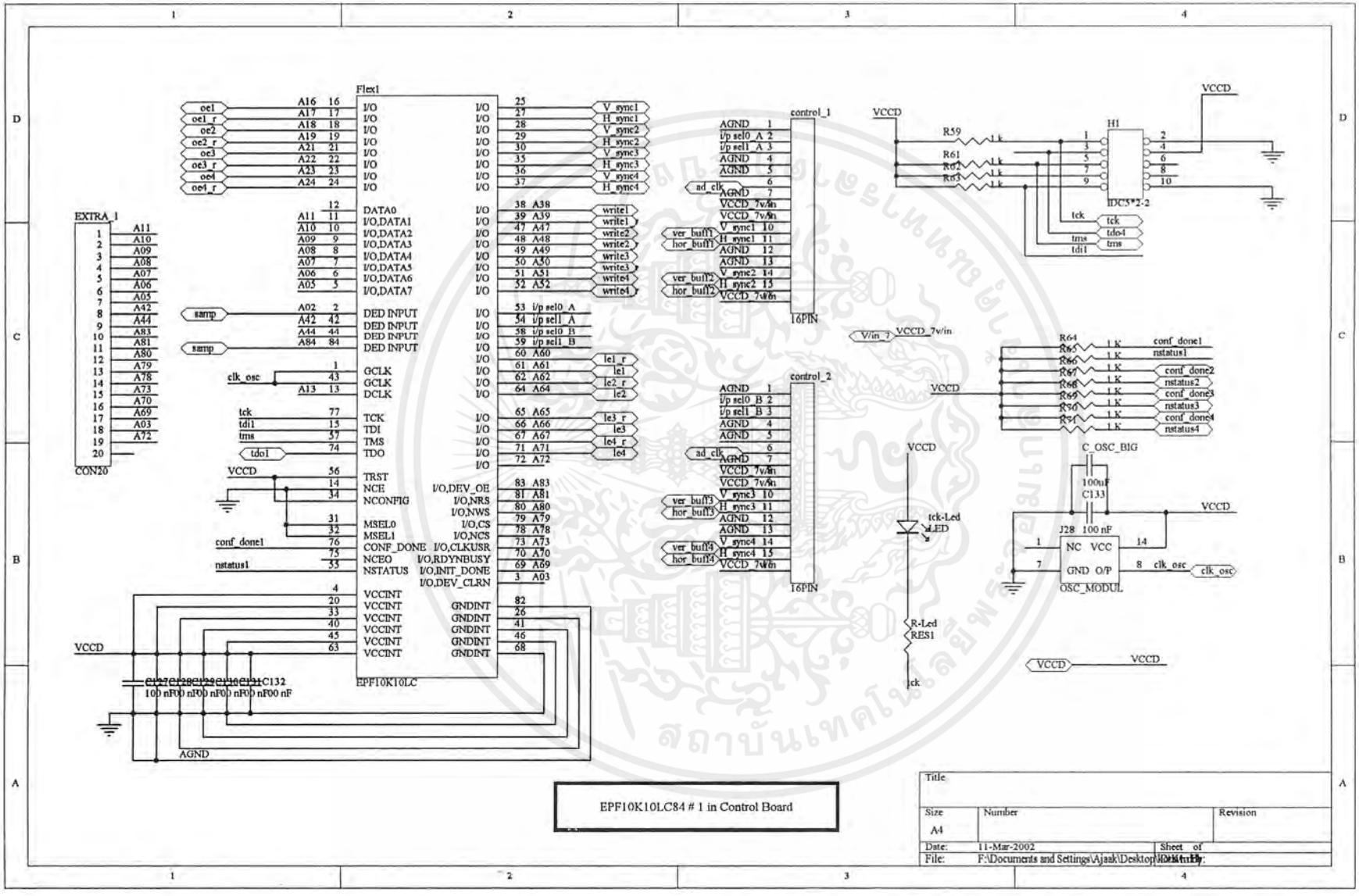


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

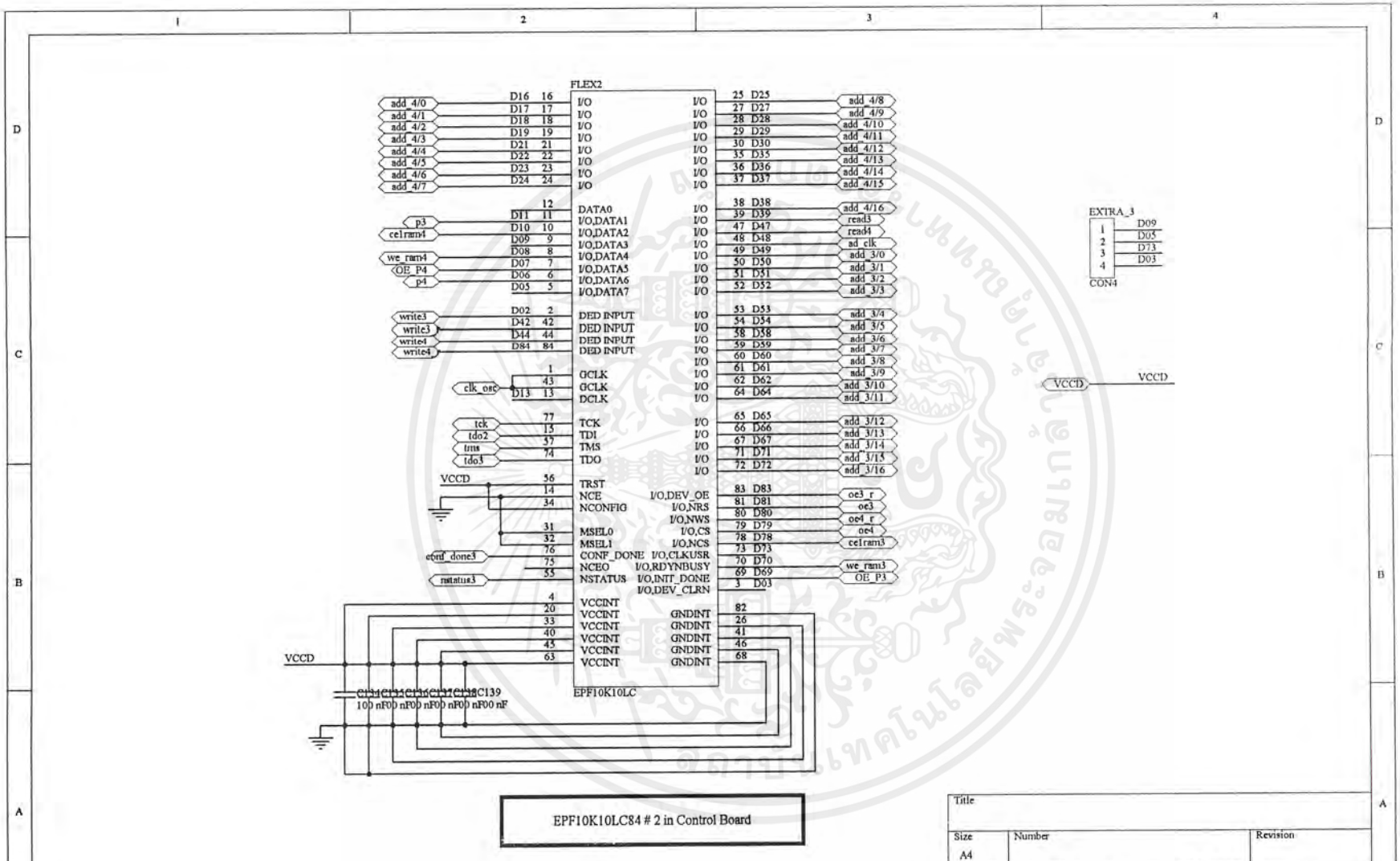

VCCD
vccd
V/in_7
VCCD_7v/in
blue_head1
blue_head2
green_head1
green_head2
red_head1
red_head2
blue_head3
blue_head4
green_head3
green_head4
red_head3
red_head4
16PIN
Led-vccd
LED
R-vccd
RES1
J3
DB15
MONITOR_O/P
CON6
Analog R o/p
Analog B o/p
Analog G o/p
h_o/p
v_o/p
AGND
VGA Picture Output & Connector to ADC Board
Title
Size
A4
Number
Revision
Date:
11-Mar-2002
Sheet of
File:
F:\Documents and Settings\Ajaak\Desktop\Desktop\



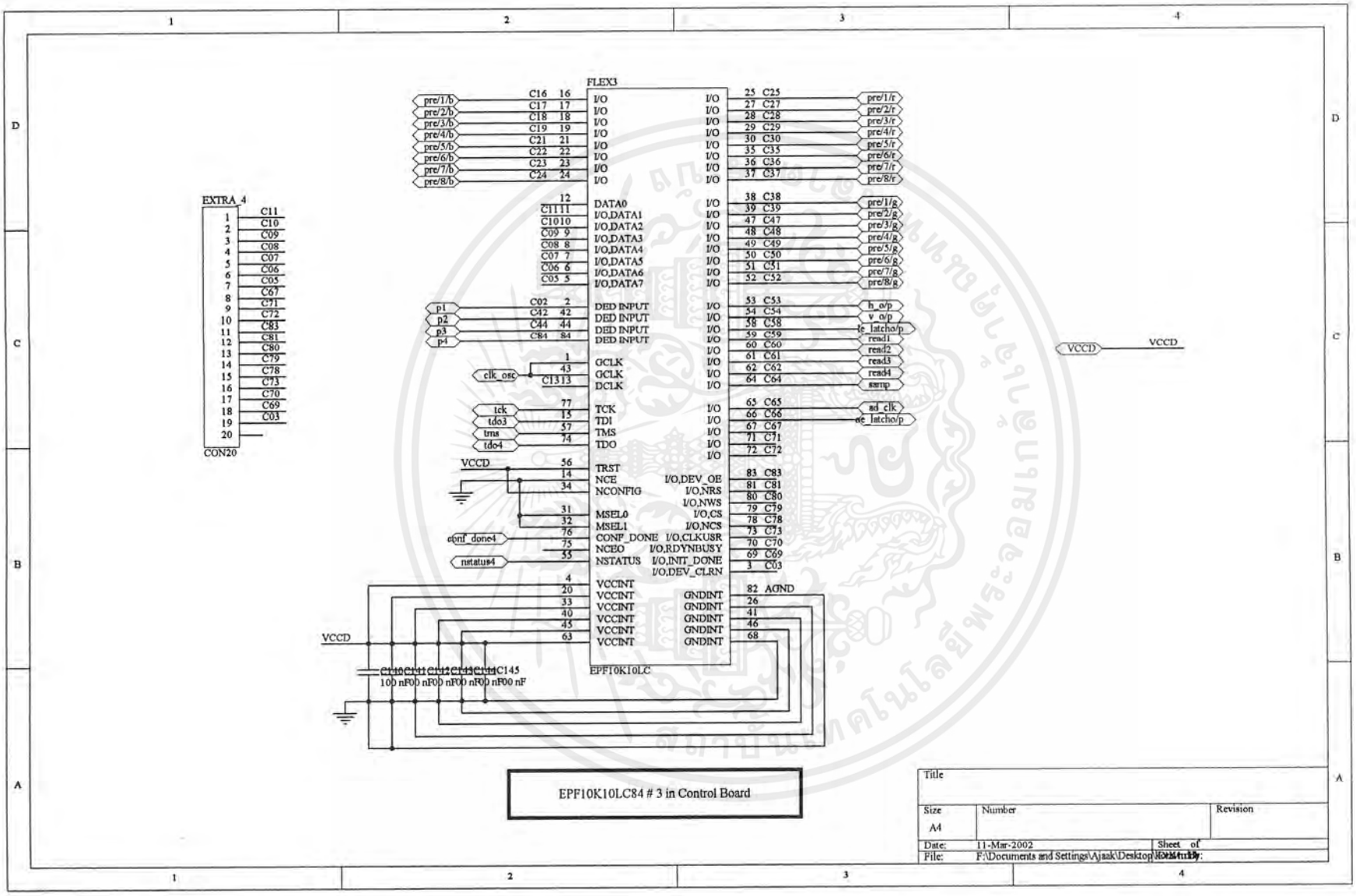
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



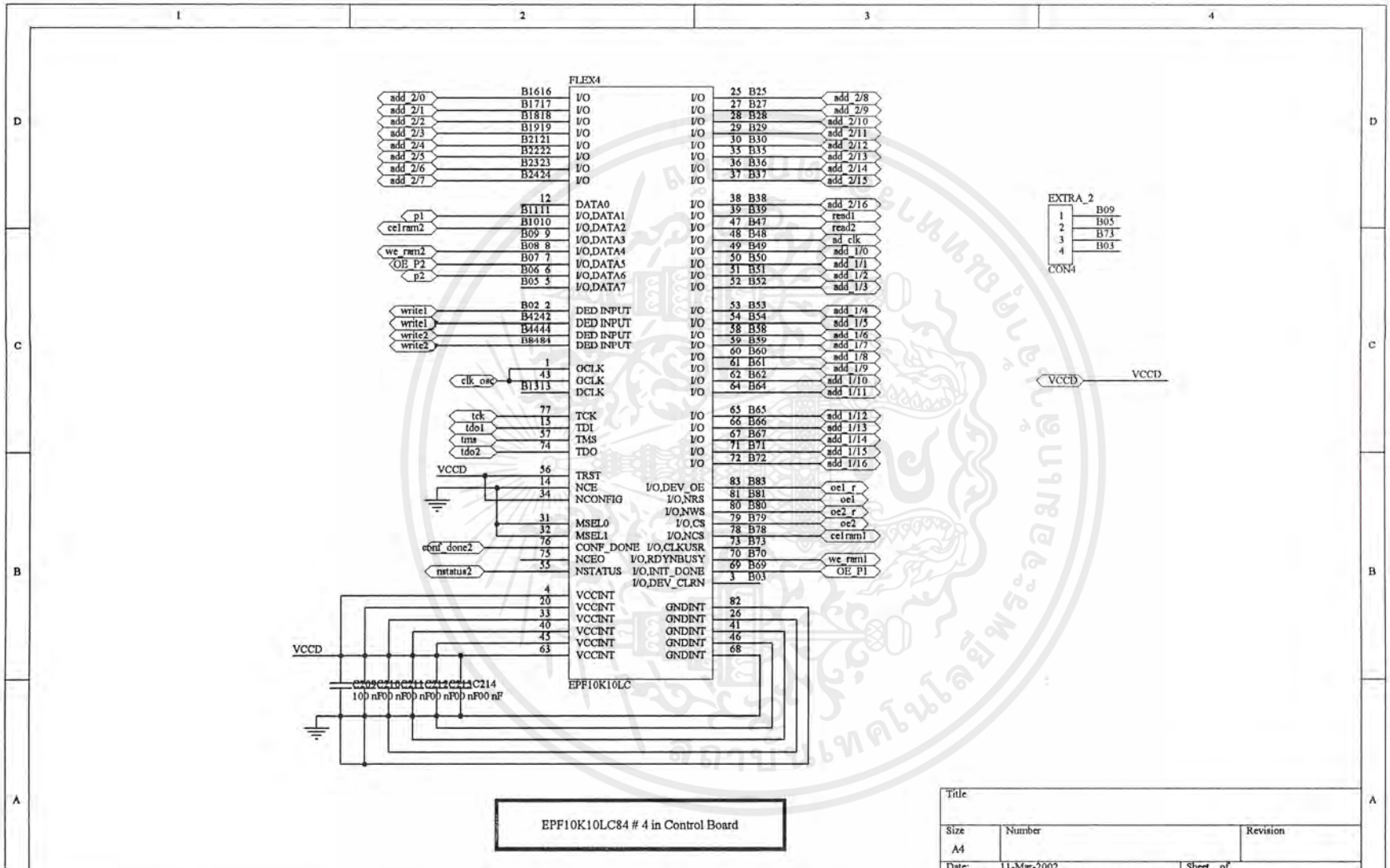
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



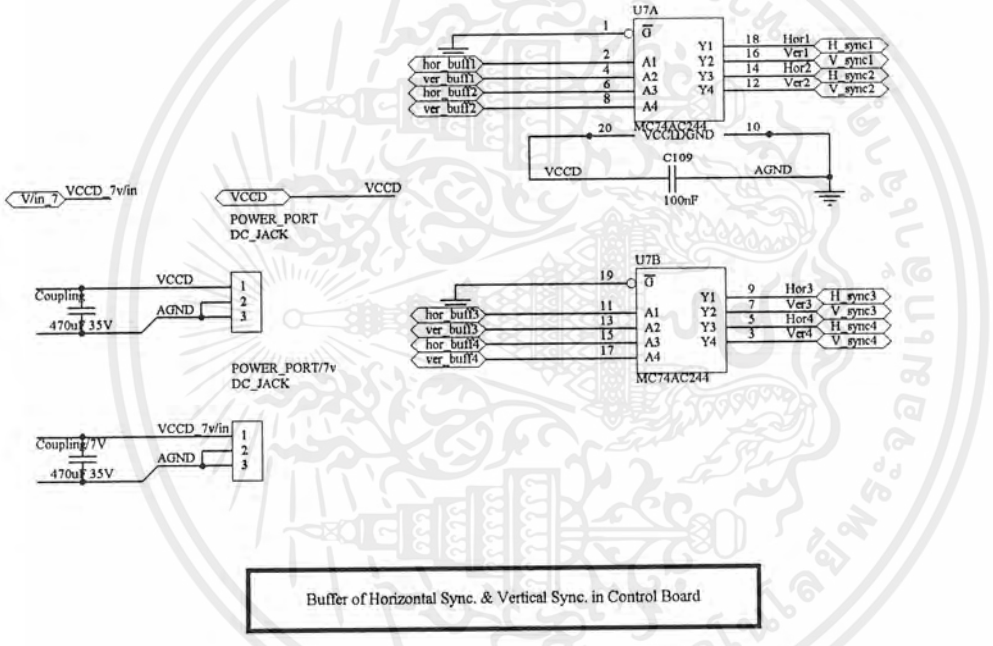
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Buffer of Horizontal Sync. & Vertical Sync. in Control Board

Title		
Size: A4	Number	Revision
Date: 11-Mar-2002	Sheet of	
File: F:\Documents and Settings\Ajask\Desktop\[illegible]	Drawn By:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้