

การออกแบบตัวควบคุมการแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลด้วยวงจรรวม FPGA

A Design of an FPGA-Based Controller for an Image Digitizer



อรวรรณ รุจิราลัย

Orawan Rujiralai

เลขที่	TK 868.05 @ 15 1513 ๗. 2
Bib Key	204728
	1 2 S.A. 2543

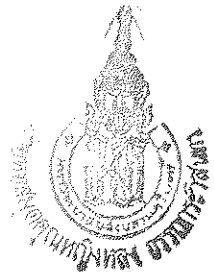
วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยสงขลานครินทร์

Master of Engineering Thesis in Electrical Engineering

Prince of Songkla University

2543



ชื่อวิทยานิพนธ์ การออกแบบตัวควบคุมการแปลงสัญญาณเป็นสัญญาณดิจิทัลด้วยวงจรรวม FPGA
 ผู้เขียน นางสาว อรวรรณ รุจิราลัย
 สาขาวิชา วิศวกรรมไฟฟ้า

คณะกรรมการที่ปรึกษา

คณะกรรมการสอบ

.....ประธานกรรมการ
 (ผู้ช่วยศาสตราจารย์ ดร.เกริกชัย ทองหนู)

.....ประธานกรรมการ
 (ผู้ช่วยศาสตราจารย์ ดร.เกริกชัย ทองหนู)

.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ บุญเจริญ วงศ์กิตติศึกษา)

.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ บุญเจริญ วงศ์กิตติศึกษา)

.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ เลียง คูบุรต์)

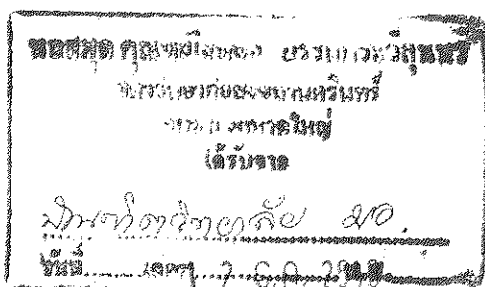
.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ เลียง คูบุรต์)

.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ ดร.สินชัย กมลภักดิ์)

.....กรรมการ
 (ผู้ช่วยศาสตราจารย์ ดร.อาจัน จิรัชีพพัฒนา)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยดำเนินการเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

(รองศาสตราจารย์ ดร.ปิติ ทฤษฎีคุณ)
 คณบดีบัณฑิตวิทยาลัย



ชื่อวิทยานิพนธ์ การออกแบบตัวควบคุมการแปลงสัญญาณเป็นสัญญาณดิจิทัลด้วย
วงจรรวม FPGA
ผู้เขียน นางสาว อรวรรณ รุจิราลัย
สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2543

บทคัดย่อ

งานวิจัยนี้นำเสนอการออกแบบและสร้างวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล (Image digitizer) และตัวควบคุมที่อยู่ในรูปวงจรรวมประเภท FPGA (Field Programmable Gate Array) เบอร์ XC4003E ในตระกูล XC4000E ของบริษัท Xilinx ซึ่งถูกออกแบบโดยใช้ภาษา VHDL (VHSIC Hardware Description Language) เพื่อทำให้เกิดความคล่องตัวทั้งในการปรับเปลี่ยนวงจร การใช้พื้นที่บนบอร์ด PCB ที่น้อยลงเมื่อเปรียบเทียบกับวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลที่จำหน่ายโดยทั่วไป โดยสามารถแปลงสัญญาณภาพจากกล้องวิดีโอ (Video camera) ให้เป็นข้อมูลภาพดิจิทัลขนาด 256×256 จุดต่อภาพด้วยความละเอียด 8 บิตต่อจุดภาพที่ความถี่ในการสุ่มสัญญาณวิดีโอ (Sampling frequency) เท่ากับ 5 เมกกะเฮิรตซ์ ซึ่งจะนำข้อมูลภาพดิจิทัลมาเก็บในหน่วยความจำแรมขนาด 64 กิโลไบต์ และทำการส่งข้อมูลภาพไปยังเครื่องคอมพิวเตอร์โดยผ่านทางพอร์ตขนานในโหมดการทำงานแบบ EPP (Enhanced Parallel Port) เพื่อนำข้อมูลภาพที่ได้ไปประยุกต์ใช้ในงานที่เกี่ยวข้องต่อไป

Thesis Title	A Design of an FPGA-Based Controller for an Image Digitizer
Author	Ms. Orawan Rujiralai
Major Program	Electrical Engineering
Academic Year	2000

Abstract

This paper presents the design and implementation of an image digitizer together with a controller that is designed by using a VHDL (VHSIC Hardware Description Language) and synthesized on a Xilinx XC4003E FPGA chip. Comparing to a commercially available image digitizer, this controller is easily modified to suit particular applications. It also consumes less PCB space. This image digitizer can convert a video signal from a video camera into a digital image with 256×256 resolution at 5 MHz sampling frequency. Subsequently, the image is stored in a 64 kByte RAM and then transferred to a personal computer via a printer port, which is operated in an EPP (Enhanced Parallel Port) mode, for further processing.

กิตติกรรมประกาศ

ขอแสดงคำขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.เกริกชัย ทองหนู ประธานกรรมการที่ปรึกษาที่ได้กรุณาให้การสนับสนุนในด้านต่างๆเป็นอย่างดีไม่ว่าจะเป็นการให้คำปรึกษา การแนะนำความรู้ในด้านต่างๆ เอกสารข้อมูล อุปกรณ์ในการทำวิจัยต่างๆ รวมทั้งกำลังใจในการแก้ปัญหาตลอดจนช่วยตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ บุญเจริญ วงศ์กิตติศึกษา และ ผู้ช่วยศาสตราจารย์ เลียง คูบุรต์ ที่ได้ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัยและการช่วยเหลือในการจัดหาอุปกรณ์ต่างๆสำหรับการทำวิจัยตลอดจนช่วยตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สินชัย กมลภักดิ์และผู้ช่วยศาสตราจารย์ ดร. อาจัน จิรชีพพัฒนา ที่ช่วยตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ คณาจารย์ และ บุคลากรในภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในด้านต่างๆที่สำคัญจนสำเร็จลุล่วง

ขอขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ที่ให้การสนับสนุนทุนในการทำวิจัย

ขอขอบพระคุณ โครงการทุนบัณฑิตศึกษาภายในประเทศ สำนักพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) ที่ได้ให้การสนับสนุนในเรื่องทุนการศึกษาตลอดระยะเวลา 2 ปี การศึกษา

ขอขอบคุณ ส่วนปฏิบัติการวิจัยและพัฒนาเทคโนโลยีไมโครอิเล็กทรอนิกส์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติและอาจารย์วรรณรัช สันติอมรทัต ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความอนุเคราะห์ในเรื่องซอฟต์แวร์ที่ใช้ในงานวิจัย

ขอขอบคุณ เพื่อนและรุ่นพี่นักศึกษาปริญญาโทภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ได้ให้คำแนะนำ คำปรึกษาและกำลังใจเป็นอย่างดีมาโดยตลอด

และที่สำคัญที่สุด ข้าพเจ้าน้อมรำลึกถึงพระคุณของ บิดามารดา และครอบครัวที่ส่งเสริมและสนับสนุนข้าพเจ้าในทุกๆเรื่องตลอดมาจนสำเร็จการศึกษา

อรวรรณ รุจิราลัย

สารบัญ

	หน้า
บทคัดย่อ.....	(3)
Abstract.....	(4)
กิตติกรรมประกาศ.....	(5)
สารบัญ.....	(6)
รายการตาราง.....	(9)
รายการภาพประกอบ.....	(10)
บทที่	
1 บทนำ.....	1
1.1 ความสำคัญและที่มาของหัวข้อวิจัย.....	1
1.2 การตรวจเอกสาร.....	2
1.3 วัตถุประสงค์.....	5
1.4 ขอบเขตของการวิจัย.....	5
1.5 ขั้นตอนและวิธีการวิจัย.....	6
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	6
2 ความรู้ทั่วไปเกี่ยวกับสัญญาณภาพโทรทัศน์.....	7
2.1 ทฤษฎีและหลักการเกิดภาพของเครื่องรับโทรทัศน์.....	7
2.2 ลักษณะของสัญญาณภาพรวม.....	9
2.3 ระบบ Synchronization.....	10
2.3.1 สัญญาณซิงก์ทางแนวนอน.....	11
2.3.2 สัญญาณซิงก์ทางแนวตั้ง.....	12
2.4 รายละเอียดลำดับของเส้นกวาด.....	12
3 การแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	14
3.1 หลักการแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	14
3.1.1 ลักษณะการเก็บข้อมูลภาพขนาด 256×256 จุดต่อภาพ.....	14
3.1.2 ลักษณะการเก็บข้อมูลภาพขนาด 512×512 จุดต่อภาพ.....	16

3.2	หลักการของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	16
3.3	ส่วนประกอบของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	19
3.4	การอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP.....	21
3.4.1	คุณสมบัติทางฮาร์ดแวร์ของโหมดการทำงานแบบ EPP.....	21
3.4.2	พอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP.....	22
3.4.3	หลักการทำ Handshaking ของโหมดการทำงานแบบ EPP.....	23
4	การออกแบบวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	27
4.1	การทำงานของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	27
4.2	ส่วนประกอบของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	28
4.2.1	วงจรแยกสัญญาณซิงก์.....	28
4.2.2	วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล.....	30
4.2.3	วงจรถ่ายสัญญาณนาฬิกา.....	31
4.2.4	ส่วนเก็บข้อมูลภาพหรือหน่วยความจำแรม.....	32
4.2.5	ตัวควบคุมในรูปวงจรรวม FPGA.....	32
4.3	การออกแบบตัวควบคุม.....	32
4.3.1	โครงสร้างภายในของตัวควบคุม.....	33
4.3.1.1	ส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม.....	33
4.3.1.2	ส่วนการอินเตอร์เฟสกับหน่วยความจำแรม.....	37
4.3.1.3	ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนาน.....	39
	ในโหมดการทำงานแบบ EPP	
4.3.2	การออกแบบตัวควบคุมโดยใช้วงจรรวม FPGA.....	39
4.4	การสร้างบอร์ดวงจรสำหรับทดสอบการทำงาน.....	41
5	ผลการทดสอบวงจรและการวิเคราะห์ผล.....	43
5.1	ผลการจำลองการทำงานของตัวควบคุม.....	43
5.2	ผลการทดสอบตัวควบคุม.....	45
5.3	ผลการทดสอบวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล.....	56
6	สรุปผลการวิจัยและข้อเสนอแนะ.....	58
	บรรณานุกรม.....	64
	ภาคผนวก.....	67

ภาคผนวก ก	ขั้นตอนการพัฒนางานออกแบบสำหรับวงจรรวม FPGA.....	67
ภาคผนวก ข	โปรแกรมรหัสต้นฉบับภาษา VHDL สำหรับการออกแบบตัวควบคุม.....	70
ภาคผนวก ค	Schematic diagram ของบอร์ดวงจร PCB1 และ PCB2.....	85
ภาคผนวก ง	ผลการทดสอบและการวิเคราะห์ผลในส่วนวงจรหลักและส่วนหลัก.....	88
	ทั้งสามส่วนของตัวควบคุม	
ภาคผนวก ง1	ผลการทดสอบของวงจรหลักต่างๆ.....	88
ภาคผนวก ง2	ผลการทดสอบของส่วนหลักทั้งสามส่วนของตัวควบคุม.....	96
ภาคผนวก จ	โครงสร้างและสถาปัตยกรรมพื้นฐานของวงจรรวม FPGA.....	120
ภาคผนวก จ1	จำนวนทรัพยากรภายในของวงจรรวม FPGA เบอร์ XC4003E.....	124
	ในตระกูล XC4000E ของบริษัท Xilinx	
ภาคผนวก จ2	Pinout ของวงจรรวม FPGA เบอร์ XC4003E.....	124
ภาคผนวก ฉ	Data sheet ของวงจรแยกสัญญาณซิงค์เบอร์ LM1881.....	127
ภาคผนวก ช	Data sheet ของวงจร Flash ADC เบอร์ MP8780.....	135
ภาคผนวก ซ	โปรแกรมภาษาซีสำหรับควบคุมการทำงานของ Image digitizer.....	145
ประวัติผู้เขียน.....		146

รายการตาราง

ตาราง	หน้า
2-1 แสดงรายละเอียดทางเวลาของเส้นกวาดทางแวนอน 1 เส้น	12
2-2 แสดงรายละเอียดลำดับของเส้นกวาด	13
3-1 แสดงชื่อและหน้าที่ของสัญญาณในโหมดการทำงานแบบ EPP	22
3-2 แสดงพอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP	23
4-1 แสดงการสรุปสัญญาณทั้งหมดที่ใช้ใน Image digitizer กับสัญญาณของตัวควบคุม	41
6-1 แสดงผลการใช้ทรัพยากรภายในวงจรรวม FPGA เบอร์ XC4003E ที่ใช้เป็นตัวควบคุม	59
ก1 แสดงซอฟต์แวร์ที่ใช้ในขั้นตอนต่างๆในงานวิจัย	69
จ1 แสดงจำนวนทรัพยากรภายในวงจรรวม FPGA เบอร์ XC4003E เปรียบเทียบกับเบอร์อื่นในตระกูล XC4000E	124
จ2 แสดง Pinout ของวงจรรวม FPGA เบอร์ XC4003E	124

รายการภาพประกอบ

ภาพประกอบ	หน้า
2-1 แสดงโครงสร้างพื้นฐานของหลอดภาพโทรทัศน์	7
2-2 แสดงการกวาดภาพแบบสอดสลับบนจอเครื่องรับโทรทัศน์	8
2-3 แสดงหลักการสร้างภาพบนจอโทรทัศน์	9
2-4 แสดงลักษณะของสัญญาณภาพรวม	9
2-5 แสดงลักษณะของสัญญาณ Hs ในเส้นกวาดทางแนวนอน 1 เส้น	11
3-1 แสดงลักษณะการเก็บภาพข้อมูลภาพขนาด 256X256 จุดต่อภาพ	15
3-2 แสดงการจัดพื้นที่หน่วยความจำในการเก็บข้อมูลภาพขนาด 256X256 จุดต่อภาพ	15
3-3 แสดงส่วนประกอบหลักของ Image digitizer โดยทั่วไป	18
3-4 แสดงการอ้างถึงตำแหน่งขององค์ประกอบภาพ	20
3-5 แสดงสัญญาณในพอร์ตต่างๆของพอร์ตขนาน	23
3-6 แสดงลักษณะของรอบการเขียนข้อมูล	24
3-7 แสดงลักษณะของรอบการเขียนแอดเดรส	24
3-8 แสดงลักษณะของรอบการอ่านข้อมูล	25
3-9 แสดงลักษณะของรอบการอ่านแอดเดรส	26
4-1 แสดงไดอะแกรมโดยรวมของ Image digitizer	28
4-2 แสดงวงจรรวมแยกสัญญาณซิงก์เบอร์ LM1881	29
4-3 แสดงไดอะแกรมภายในของวงจรรวมแยกสัญญาณซิงก์เบอร์ LM1881	29
4-4 แสดงวงจร Flash ADC และส่วน Signal conditioner	30
4-5 แสดงวงจรกำเนิดสัญญาณนาฬิกาความถี่ 5 เมกกะเฮิร์ตซ์	31
4-6 แสดงจำนวนสัญญาณนาฬิกาที่เกิดขึ้นระหว่างสัญญาณ Hs	32
4-7 แสดงโครงสร้างภายในของตัวควบคุม	33
4-8 แสดงส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม	34
4-9 แสดงส่วนสร้างสัญญาณให้วงจรนับเริ่มการทำงาน	34
4-10 แสดงไดอะแกรมทางเวลาของสัญญาณ vstartdel และ vstartcnt	35
4-11 แสดงไดอะแกรมทางเวลาของสัญญาณ hstartdel และ wstartcnt	36
4-12 แสดงส่วนสร้างสัญญาณรีเซ็ตเพื่อใช้ในส่วน VerAdrCounter	37

รายการภาพประกอบ(ต่อ)

ภาพประกอบ	หน้า
4-13 แสดงสัญญาณควบคุมเมื่ออยู่ในโหมดการเขียนข้อมูลภาพ	38
4-14 แสดงสัญญาณควบคุมเมื่ออยู่ในโหมดการอ่านข้อมูลภาพ	38
4-15 แสดงสัญญาณ Wait เมื่อมีสัญญาณ Write และ Data strobe	39
4-16 แสดงโมดูลในแต่ละลำดับขั้นของการออกแบบตัวควบคุมโดยใช้วงจรรวม FPGA	40
4-17 แสดงบอร์ด PCB1 สำหรับการทดสอบ	42
4-18 แสดงบอร์ด PCB2 สำหรับการทดสอบ	42
5-1 แสดงตัวอย่างผลการจำลองการทำงานของตัวควบคุมในโหมดการเขียน	44
5-2 แสดงตัวอย่างผลการจำลองการทำงานของตัวควบคุมในโหมดการเขียน	44
5-3 แสดงสัญญาณในการทดสอบการทำงานของตัวควบคุม	45
5-4 แสดงสัญญาณ RAMAdr0-7 เมื่ออยู่ในโหมดการเขียน	46
5-5 แสดงสัญญาณ RAMAdr8-15 เมื่ออยู่ในโหมดการเขียน	47
5-6 แสดงสัญญาณ Clk และสัญญาณ RAMWE เมื่ออยู่ในโหมดการเขียน	47
5-7 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการเขียน	48
5-8 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการเขียน	48
5-9 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการเขียน	49
5-10 แสดงสัญญาณ ADCData0 และสัญญาณ RAMData0 เมื่ออยู่ในโหมดการเขียน	49
5-11 แสดงสัญญาณ ADCData7 และสัญญาณ RAMData7 เมื่ออยู่ในโหมดการเขียน	50
5-12 แสดงสัญญาณ RAMAdr0-7 เมื่ออยู่ในโหมดการอ่าน	51
5-13 แสดงสัญญาณ RAMAdr8-15 เมื่ออยู่ในโหมดการอ่าน	52
5-14 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการอ่าน	52
5-15 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการอ่าน	53
5-16 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน	53
5-17 แสดงสัญญาณ RAMData0 และสัญญาณ PCData0 เมื่ออยู่ในโหมดการอ่าน	54
5-18 แสดงสัญญาณ RAMData7 และสัญญาณ PCData7 เมื่ออยู่ในโหมดการอ่าน	54
5-19 แสดงสัญญาณ EPPWrite และสัญญาณ EPPDatastrb จากพอร์ตขนาน	55
5-20 แสดงสัญญาณ EPPDatastrb เมื่อมีสัญญาณ EPPWait ตอรับ	55

รายการภาพประกอบ(ต่อ)

ภาพประกอบ	หน้า
5-21 แสดงภาพตัวอย่างที่ได้จาก Image digitizer Image digitizer (รูปขวามือ) เมื่อเปรียบเทียบกับภาพที่เก็บได้จาก Video capture card (รูปซ้ายมือ)	57
ก1 แสดงขั้นตอนการพัฒนางานออกแบบวงจรรวม FPGA	68
ค1 แสดง Schematic diagram ของบอร์ด PCB1	86
ค2 แสดง Schematic diagram ของบอร์ด PCB2	87
ง1 แสดงไดอะแกรมของวงจรแยกสัญญาณซิงก์	88
ง2 แสดงสัญญาณ Vs และสัญญาณ Hs ที่ได้จากวงจรแยกสัญญาณซิงก์	89
ง3 แสดงไดอะแกรมของวงจรถ่ายสัญญาณนาฬิกาความถี่ 5 เมกกะเฮิรตซ์	89
ง4 แสดงสัญญาณ Hs และสัญญาณ Clk จากวงจรถ่ายสัญญาณนาฬิกา	90
ง5 แสดงไดอะแกรมของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	90
ง6 แสดงสัญญาณภาพรวมและสัญญาณ Vin ของวงจร ADC	91
ง7 แสดงสัญญาณ Vin กับสัญญาณ ADCData0 ของวงจร ADC	91
ง8 แสดงสัญญาณ Vin กับสัญญาณ ADCData7 ของวงจร ADC	92
ง9 แสดงสัญญาณ Vin เมื่อเป็นภาพดำล้วนกับสัญญาณ ADCData0	92
ง10 แสดงสัญญาณ Vin เมื่อเป็นภาพดำล้วนกับสัญญาณ ADCData7	93
ง11 แสดงสัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW ของ Image digitizer	94
ในโหมดการเขียน	
ง12 แสดงสัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW ของ Image digitizer	95
ในโหมดการอ่าน	
ง13 แสดงสัญญาณ Address strobe ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ Reset	95
ง14 แสดงสัญญาณ Vs และสัญญาณ vstartdel เมื่ออยู่ในโหมดการเขียน	96
ง15 แสดงสัญญาณ vstartdel กับสัญญาณ vstartcnt เมื่ออยู่ในโหมดการเขียน	97
ง16 แสดงสัญญาณ Hs และสัญญาณ hstartdel เมื่ออยู่ในโหมดการเขียน	97
ง17 แสดงสัญญาณ hstartdel และสัญญาณ wstartcnt เมื่ออยู่ในโหมดการเขียน	98
ง18 แสดงสัญญาณ wstartcnt และสัญญาณ rstartcnt เมื่ออยู่ในโหมดการเขียน	98
ง19 แสดงสัญญาณ wstartcnt และสัญญาณ startcnt เมื่ออยู่ในโหมดการเขียน	99
ง20 แสดงสัญญาณ wstartcnt และสัญญาณ rstartcnt เมื่ออยู่ในโหมดการอ่าน	100

รายการภาพประกอบ(ต่อ)

ภาพประกอบ	หน้า
ง21 แสดงสัญญาณ wstartcnt และสัญญาณ startcnt เมื่ออยู่ในโหมดการอ่าน	100
ง22 แสดงสัญญาณ CClk และสัญญาณ Clk เมื่ออยู่ในโหมดการเขียน	101
ง23 แสดงสัญญาณ CClk และสัญญาณ Data strobe ที่อินเวอร์ตเมื่ออยู่ในโหมดการอ่าน	102
ง24 แสดงสัญญาณ HAdr0-7 เมื่ออยู่ในโหมดการเขียน	103
ง25 แสดงสัญญาณ HAdr7 และสัญญาณ up_enable เมื่ออยู่ในโหมดการเขียน	103
ง26 แสดงสัญญาณ HAdr0-7 เมื่ออยู่ในโหมดการอ่าน	104
ง27 แสดงสัญญาณ HAdr7 และสัญญาณ up_enable เมื่ออยู่ในโหมดการอ่าน	104
ง28 แสดงสัญญาณ up_enable และสัญญาณ VAdr0 เมื่ออยู่ในโหมดการเขียน	105
ง29 แสดงสัญญาณ VAdr0-7 เมื่ออยู่ในโหมดการเขียน	106
ง30 แสดงสัญญาณ VAdr7 และสัญญาณ TC เมื่ออยู่ในโหมดการเขียน	106
ง31 แสดงสัญญาณ up_enable และสัญญาณ VAdr0 เมื่ออยู่ในโหมดการอ่าน	107
ง32 แสดงสัญญาณ VAdr0-7 เมื่ออยู่ในโหมดการอ่าน	107
ง33 แสดงสัญญาณ VAdr7 และสัญญาณ TC เมื่ออยู่ในโหมดการอ่าน	108
ง34 แสดงสัญญาณ Vs และสัญญาณ VReset เมื่ออยู่ในโหมดการเขียน	108
ง35 แสดงสัญญาณ Vs และสัญญาณ VReset เมื่ออยู่ในโหมดการอ่าน	109
ง36 แสดงสัญญาณ Clk และสัญญาณ RAMWE เมื่ออยู่ในโหมดการเขียน	110
ง37 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการเขียน	110
ง38 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการเขียน	111
ง39 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการเขียน	111
ง40 แสดงสัญญาณ RAMCE1 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการเขียน	112
ง41 แสดงสัญญาณ ADCData0 และสัญญาณ RAMData0 เมื่ออยู่ในโหมดการเขียน	112
ง42 แสดงสัญญาณ ADCData7 และสัญญาณ RAMData7 เมื่ออยู่ในโหมดการเขียน	113
ง43 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการอ่าน	114
ง44 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการอ่าน	114
ง45 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน	115
ง46 แสดงสัญญาณ RAMCE1 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน	115
ง47 แสดงสัญญาณ RAMData0 และสัญญาณ PCData0 เมื่ออยู่ในโหมดการอ่าน	116

รายการภาพประกอบ(ต่อ)

ภาพประกอบ	หน้า
ง48 แสดงสัญญาณ RAMData7 และสัญญาณ PCData7 เมื่ออยู่ในโหมดการอ่าน	116
ง49 แสดงสัญญาณ Write และสัญญาณ Data strobe จากพอร์ตนาน	117
ง50 แสดงสัญญาณ Data strobe เมื่อมีสัญญาณ Wait ตอบรับ	118
ง51 แสดงสัญญาณ Data strobe เมื่อไม่มีสัญญาณ Wait ตอบรับ	118
จ1 แสดงโครงสร้างพื้นฐานภายในของวงจรรวม FPGA ของบริษัท Xilinx	121
จ2 แสดงบล็อกไดอะแกรมอย่างง่ายของ XC4000-Series CLB	122
จ3 แสดงบล็อกไดอะแกรมอย่างง่ายของ Xilinx IOB ในตระกูล XC4000E	123

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของหัวข้อวิจัย

ในปัจจุบันการแปลงสัญญาณภาพจากกล้องวิดีโอ (Video camera) ให้เป็นสัญญาณดิจิทัลเพื่อนำไปใช้งานด้านต่างๆ อาทิเช่น ด้านการประมวลผลภาพ (Image processing) การจดจำรูปแบบ (Pattern recognition) เป็นต้น สามารถทำได้โดยใช้วงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล (Image digitizer) ซึ่งสามารถ digitize สัญญาณภาพให้เป็นสัญญาณดิจิทัลได้โดยทั่วไปมักพบข้อบกพร่อง เช่น ความซับซ้อนของวงจรที่ทำให้ไม่สะดวกทั้งในการสร้างใช้งาน การสิ้นเปลืองค่าใช้จ่ายและกำลังงานไฟฟ้าของวงจร เป็นต้น ปัญหาเหล่านี้ล้วนแล้วแต่ทำให้การ digitize สัญญาณภาพเป็นเรื่องที่ยุ่งยาก สำหรับงานวิจัยนี้มีแนวความคิดเริ่มต้นมาจากการสร้าง Image digitizer ในงานวิจัยเรื่องเครื่องตรวจขวดอัตโนมัติ (Automatic Bottle Inspecting System) (เอกสาร 1.2.1 ในหัวข้อ 1.2 การตรวจเอกสาร) สำหรับใช้ในการเก็บภาพกันขวดแก้วเพื่อตรวจหาสิ่งแปลกปลอมด้วยการประมวลผลของเครื่องคอมพิวเตอร์ โดยพบว่านอกจากข้อบกพร่องตามที่ได้กล่าวมาแล้วยังมีข้อจำกัดของ Image digitizer อีกหลายประการ เช่น ความไม่สะดวกในการใช้งานตลอด 24 ชั่วโมงเนื่องจากจำเป็นต้องมีเครื่องคอมพิวเตอร์สำหรับควบคุมการทำงานของระบบตลอดเวลา เป็นต้น และด้วยความก้าวหน้าทางเทคโนโลยีของวงจรรวมในปัจจุบันทำให้เราสามารถออกแบบอุปกรณ์ทางดิจิทัลฮาร์ดแวร์โดยใช้วงจรรวมประเภท FPGA (Field Programmable Gate Array) ได้ โดยผู้ให้สามารถโปรแกรมการเชื่อมต่อภายในของวงจรรวมเพื่อสร้างฟังก์ชันการทำงานได้ตามต้องการ

ดังนั้นงานวิจัยนี้จึงนำเสนอการออกแบบและสร้าง Image digitizer และตัวควบคุมที่อยู่ในรูปวงจรรวมประเภท FPGA ในตระกูล XC4000E ของบริษัท Xilinx ซึ่งถูกออกแบบโดยใช้ภาษา VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) เพื่อแก้ปัญหาดังที่กล่าวมาแล้วอีกทั้งเพื่อให้สามารถปรับเปลี่ยนการทำงานเพื่อนำไปใช้ในงานด้านต่างๆได้ โดยจะควบคุมทั้งการเขียนข้อมูลภาพลงหน่วยความจำแรมซึ่งอยู่นอกตัวควบคุมและอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่พอร์ตขนานของเครื่องคอมพิวเตอร์ในโหมดการทำงานแบบ EPP (Enhanced Parallel Port) เพื่อประโยชน์ในการนำภาพดิจิทัลไปใช้ในงานต่างๆที่เกี่ยวข้องต่อไป

1.2 การตรวจเอกสาร

1.2.1 เครื่องตรวจขวดอัตโนมัติ (Automatic Bottle Inspecting System) (จรรยา ไชยนิศย์, 2542) งานวิจัยนี้นำเสนอการสร้างระบบสำหรับตรวจสอบสิ่งแปลกปลอมภายในขวดแก้วสำหรับบรรจุผลิตภัณฑ์ต่างๆภายในโรงงานอุตสาหกรรมโดยใช้ Image digitizer ในการเก็บภาพกันขวดเพื่อนำมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดย Image digitizer นี้มีความถี่ในการสุ่มสัญญาณวิดีโอ (Sampling frequency) เท่ากับ 5 เมกกะเฮิร์ตซ์สามารถเปลี่ยนสัญญาณภาพจากกล้อง CCD (Charge-Coupled Device) ให้เป็นสัญญาณดิจิทัลแล้วอ่านเข้าเครื่องคอมพิวเตอร์โดยผ่านทาง สล็อต ISA bus เพื่อนำภาพมาวิเคราะห์ ภาพที่ได้มีความละเอียดขนาด 256×256 จุดต่อภาพและระดับเทา 64 ระดับ โดยพบว่า Image digitizer ไม่สามารถทำงานได้ตลอด 24 ชั่วโมงเนื่องจากจำเป็นต้องมีเครื่องคอมพิวเตอร์สำหรับควบคุมการทำงานของระบบตลอดเวลา ทำให้ไม่สะดวกในการใช้งาน

1.2.2 เครื่องแสดงภาพสีทางดิจิทัลขนาด 512×512 จุดต่อภาพ (วิวัฒน์ ศรีเพ็ง, 2534) งานวิจัยนี้นำเสนอการออกแบบและสร้างเครื่องแสดงภาพสีทางดิจิทัลที่มีความถี่ในการสุ่มสัญญาณวิดีโอเท่ากับ 10 เมกกะเฮิร์ตซ์ โดยนำสัญญาณภาพโทรทัศน์สีมาแปลงให้อยู่ในรูปข้อมูลดิจิทัลแล้วอ่านเข้าเครื่องคอมพิวเตอร์ผ่านทางสล็อต ISA bus หรือแปลงกลับเป็นสัญญาณแอนะล็อกเพื่อแสดงภาพบนจอแสดงผล ภาพสีแต่ละภาพได้จากเส้นกวาดจำนวน 512 เส้น แต่ละเส้นประกอบด้วยจุดภาพจำนวน 512 จุดโดยแต่ละจุดสามารถแสดงระดับสีได้ทั้งหมด 16,777,216 สี ให้ความเร็วความจำความเร็วต่ำเพื่อเก็บข้อมูลในแต่ละภาพขนาด 384 กิโลไบต์และสามารถใช้งานร่วมกับเครื่อง IBM PC/XT/AT หรือที่เข้ากันได้ (Compatible) ผู้วิจัยสรุปว่าสามารถใช้งานได้ดีแต่พบปัญหาในเรื่องของสัญญาณรบกวนเนื่องจากมีส่วนประกอบของอุปกรณ์วงจรรวมจำนวนมากและเครื่องทำงานที่ความเร็วสูง

1.2.3 เครื่องเปลี่ยนสัญญาณภาพแบบเวลาจริง (อรรถสิทธิ์ หล้าสกุล และ วิสุทธิ์ พิบูลวางกูร, 2531) เป็นการนำเสนอการออกแบบและสร้าง Real Time Image Digitizer ที่ใช้กับเครื่อง IBM PC/XT หรือที่เข้ากันได้ มีความถี่ในการสุ่มสัญญาณวิดีโอเท่ากับ 5 เมกกะเฮิร์ตซ์ โดยนำข้อมูลภาพจากกล้องวงจรปิดที่วิ่งไปมาเปลี่ยนให้เป็นสัญญาณดิจิทัลแล้วอ่านเข้าเครื่องคอมพิวเตอร์ทางสล็อต ISA bus เพื่อนำมาวิเคราะห์และบันทึกลงบนแผ่นข้อมูลและแสดงภาพบนจอแสดงผลได้ มีความละเอียดของภาพขนาด 256×256 จุดต่อภาพและระดับเทา 256 ระดับ ตัว Digitizer ประกอบด้วยวงจรต่างๆ เช่น วงจรบัฟเฟอร์เพื่อรับสัญญาณภาพจากกล้องวิดีโอส่งให้วงจรแยก

สัญญาณซิงก์ วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบแฟลช (Flash Analog-to-Digital Converter, Flash ADC) ขนาด 8 บิต วงจรนับ (Counter) ขนาด 8 บิตจำนวน 2 วงจร สำหรับสร้างแอดเดรสเพื่อเก็บข้อมูลดิจิทัลลงหน่วยความจำแรม วงจรกำเนิดสัญญาณนาฬิกา (Oscillator) สำหรับควบคุมกระบวนการต่างๆและหน่วยความจำแรมขนาด 64 กิโลไบต์ เป็นต้น โดยผู้วิจัยสรุปว่าเครื่อง Digitizer ที่ได้มีขนาดวงจรที่เล็กกว่าและสามารถเทียบประสิทธิภาพกับของต่างประเทศได้

1.2.4 Digitizer/Transmitter : Build a Grey Scale Video Digitizer (Steve Caircia, 1987) เป็นบทความที่กล่าวถึงการสร้างเครื่อง ImageWise Digitizer มีความถี่ในการสุ่มสัญญาณวิดีโอเท่ากับ 5 เมกกะเฮิร์ตซ์ โดยสามารถ digitize สัญญาณภาพจากกล้องวิดีโอและส่งข้อมูลออกไปทางพอร์ตอนุกรม (Serial port) เพื่อแสดงภาพบนจอแสดงผลได้ ตัว Digitizer ประกอบด้วยวงจรรย่อย เช่น วงจรแยกสัญญาณซิงก์ วงจร Flash ADC ขนาด 6 บิตซึ่งให้ระดับเทา 64 ระดับ วงจรมัลติเพล็กซ์ วงจรกำเนิดสัญญาณนาฬิกา วงจรนับขนาด 4 บิตจำนวน 2 วงจรและหน่วยความจำแรมขนาด 64 กิโลไบต์ เป็นต้น นอกจากนี้ยังมีส่วนวงจรอื่นๆอีก คือ ส่วนแคลมป์สัญญาณซิงก์ (Sync. clamp) เพื่อกำจัดแรงดันดีซีออฟเซต (DC offset) ของสัญญาณวิดีโอให้เป็นศูนย์ วงจรกรอง (Filter) เพื่อใช้กรองความถี่ของสัญญาณที่ใช้ในการเข้ารหัส (Encoding) สัญญาณภาพสี และมีการนำไมโครคอนโทรลเลอร์ตระกูล 8051 มาใช้เป็นตัวนับจำนวนแถวของแอดเดรสในการเกิดภาพ เป็นต้น

สำหรับ Image digitizer ในงานวิจัยนี้มีหลักการทำงานคล้ายกับหลักการของเอกสารทั้งสี่กล่าวคือใช้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลเพื่อแปลงสัญญาณภาพขาวดำจากกล้องวิดีโอให้เป็นข้อมูลดิจิทัลเก็บไว้ในหน่วยความจำแรมและสามารถอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่เครื่องคอมพิวเตอร์เพื่อแสดงผลได้ แต่มีข้อแตกต่างที่เด่นชัดและเป็นหัวใจของงานวิจัยนี้คือมีการเพิ่มตัวควบคุมที่อยู่ในรูปวงจรรวม FPGA เบอร์ XC4003E ในตระกูล XC4000E ของบริษัท Xilinx สำหรับควบคุมกระบวนการในการเขียนข้อมูลภาพลงหน่วยความจำแรมและอ่านข้อมูลภาพเข้าสู่เครื่องคอมพิวเตอร์ผ่านทางพอร์ตขนานในโหมดการทำงานแบบ EPP แทนการใช้สลอต ISA bus กล่าวคือทำหน้าที่ควบคุม

1. การกำเนิดแอดเดรสให้หน่วยความจำแรมเพื่อเก็บข้อมูลดิจิทัลที่ได้จากการแปลง
2. การอินเตอร์เฟสกับหน่วยความจำแรมเพื่อสร้างสัญญาณควบคุมต่างๆ รวมทั้งส่งข้อมูลและแอดเดรสให้หน่วยความจำแรมทั้งในการเขียนและอ่านข้อมูลภาพ
3. การสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP เพื่อโอนย้ายข้อมูล

ดิจิตอลจากหน่วยความจำแรมเข้าสู่เครื่องคอมพิวเตอร์

สำหรับการนำวงจรรวม FPGA มาใช้เป็นตัวควบคุมของ Image digitizer นี้ยังไม่มีใครเคยมีผู้ใดกระทำมาก่อน โดยประโยชน์ที่ได้จากตัวควบคุมนี้มีหลายประการ เช่น

- สามารถทำงานในลักษณะ Real time operation ได้
- สามารถลดความซับซ้อนของจำนวนอุปกรณ์อิเล็กทรอนิกส์หรือจำนวนของ Discrete components ที่ใช้ใน Image digitizer ลงได้เนื่องจากตัวควบคุมนี้จะทำหน้าที่แทนวงจรต่างๆ เช่น วงจรนับ วงจรมัลติเพล็กซ์เซอร์ เป็นต้น ทำให้พื้นที่ของบอร์ด PCB มีขนาดเล็กและการปรับเปลี่ยนวงจรต่างๆภายใน Image digitizer สามารถทำได้อย่างคล่องตัวมากขึ้น

- สามารถปรับเปลี่ยนการทำงานของตัวควบคุมเพื่อนำ Image digitizer ไปใช้งานด้านต่างๆได้อย่างรวดเร็วเนื่องจากความคล่องตัวในการออกแบบวงจรรวม FPGA อีกทั้งการออกแบบตัวควบคุมในงานวิจัยนี้เป็นการออกแบบในลักษณะที่เรียกว่า Core Design กล่าวคือเป็นการออกแบบที่เน้นคุณสมบัติในการทำงานของวงจรรวมเป็นสำคัญเพื่อให้สามารถนำไปปรับเปลี่ยนในงานต่างๆได้ เช่น อาจปรับเปลี่ยนตัวควบคุมให้ทำงานร่วมกับไมโครคอนโทรลเลอร์เพื่อใช้ในการตรวจวัดก็ได้ เป็นต้น

นอกจากประโยชน์ที่ได้รับแล้ว Image digitizer ที่ได้จากงานวิจัยนี้อาจมีข้อด้อยบางประการเมื่อเปรียบเทียบกับประสิทธิภาพกับ Image digitizer ที่จำหน่ายโดยทั่วไป อาทิเช่น ในเรื่องความละเอียดของภาพที่เลือกใช้ขนาด 256×256 จุดต่อภาพซึ่งเป็นความละเอียดภาพที่ค่อนข้างต่ำ การใช้งานกับสัญญาณภาพขาวดำเท่านั้น เป็นต้น ทั้งนี้เนื่องด้วยข้อจำกัดในหลายๆด้านเช่น ต้นทุน อุปกรณ์ต่างๆที่ใช้ เป็นต้น แต่สิ่งที่ได้จากงานวิจัยนี้คือการนำเอาประสิทธิภาพและความสามารถของวงจรรวม FPGA มาพัฒนาให้เกิดประโยชน์ในด้านที่เกี่ยวข้องกับ Image digitizer และสามารถเป็นแนวทางสำหรับการประยุกต์ใช้วงจรรวม FPGA ในงานด้านอื่นๆต่อไปได้ในอนาคต

สำหรับข้อแตกต่างปลีกย่อยอื่นๆกับเอกสาร 1.2.1 คือ ใช้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลขนาด 8 บิตทำให้สามารถแสดงระดับเทาของภาพดิจิตอลได้ 256 ระดับ ในขณะที่ในเอกสารนี้ใช้วงจรแปลงขนาด 6 บิตทำให้แสดงระดับเทาได้เพียง 64 ระดับ

ข้อแตกต่างจากเอกสาร 1.2.2 คือ

1. ใช้ได้เฉพาะสัญญาณภาพขาวดำ ไม่สามารถใช้ได้กับสัญญาณภาพสี
2. มีความละเอียดภาพขนาด 256×256 จุดต่อภาพไม่ใช่ขนาด 512×512 จุดต่อภาพ
3. ไม่มีการใช้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกเพื่อแสดงภาพบนจอ

แสดงผล

4. มีความถี่ในการสุ่มสัญญาณวิดีโอเท่ากับ 5 เมกกะเฮิร์ตซ์ไม่ใช่ 10 เมกกะเฮิร์ตซ์
ข้อแตกต่างจากเอกสาร 1.2.3 คือไม่มีการใช้วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณ
แอนะล็อกเพื่อแสดงภาพบนจอแสดงผล

และข้อแตกต่างจากเอกสาร 1.2.4 คือ

1. ใช้ได้เฉพาะสัญญาณภาพขาวดำ ไม่สามารถใช้ได้กับสัญญาณภาพสี
2. ส่งข้อมูลไปยังเครื่องคอมพิวเตอร์โดยผ่านทางพอร์ตขนานไม่ได้ผ่านพอร์ตอนุกรม
3. สามารถแสดงระดับเทาของภาพดิจิทัลได้ 256 ระดับ ในขณะที่ในเอกสารนี้แสดง
ระดับเทาได้เพียง 64 ระดับ

1.3 วัตถุประสงค์

1.3.1 เพื่อศึกษาหลักการทำงานของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

1.3.2 เพื่อออกแบบและสร้างวงจรรวมแบบ FPGA สำหรับควบคุมการเปลี่ยนสัญญาณภาพ
จากกล้องวิดีโอให้เป็นสัญญาณดิจิทัล

1.3.3 สามารถนำวงจรรวมที่ได้ออกแบบไปประยุกต์ใช้งานด้านต่างๆต่อไป

1.4 ขอบเขตของการวิจัย

1.4.1 ออกแบบและสร้าง Image digitizer เพื่อเก็บภาพดิจิทัลขนาด 256×256 จุดต่อภาพ
ด้วยความละเอียด 8 บิตต่อจุดภาพที่ความถี่ในการสุ่มสัญญาณวิดีโอเท่ากับ 5 เมกกะเฮิร์ตซ์ รวม
ถึงการออกแบบและสร้างตัวควบคุมที่อยู่ในรูปของวงจรรวม FPGA เบอร์ XC4003E ในตระกูล
XC4000E ของบริษัท Xilinx ซึ่งเป็นส่วนประกอบที่สำคัญสำหรับควบคุมการเปลี่ยนสัญญาณภาพ
ให้เป็นสัญญาณดิจิทัล

1.4.2 นำข้อมูลภาพดิจิทัลมาเก็บในหน่วยความจำแรมขนาด 64 กิโลไบต์ และทำการส่งข้อมูลภาพไปยังเครื่องคอมพิวเตอร์โดยผ่านทางพอร์ตขนานในโหมดการทำงานแบบ EPP เพื่อนำข้อมูลภาพที่ได้ไปใช้ในงานที่เกี่ยวข้องต่อไป

1.5 ขั้นตอนและวิธีการวิจัย

- 1.5.1 ศึกษาหลักการในการสร้างภาพของโทรทัศน์
- 1.5.2 ศึกษาหลักการเปลี่ยนสัญญาณภาพเป็นสัญญาณดิจิทัล
- 1.5.3 ศึกษาโปรแกรมที่เกี่ยวกับการออกแบบวงจรรวม FPGA
- 1.5.4 เขียนโปรแกรมภาษา VHDL เพื่อออกแบบวงจรรวมโดยใช้ชุดออกแบบของบริษัท Xilinx
- 1.5.5 สร้างและทดสอบตัวควบคุมโดยใช้วงจรรวม FPGA ของบริษัท Xilinx
- 1.5.6 สร้างและทดสอบระบบเปลี่ยนสัญญาณภาพเป็นสัญญาณดิจิทัล
- 1.5.7 สรุปและรวบรวมผลการทดสอบ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ได้อุปกรณ์วงจรรวมที่สามารถนำมาใช้ประโยชน์เพื่อการ digitize สัญญาณภาพจากกล้องวิดีโอให้เป็นสัญญาณดิจิทัลได้
- 1.6.2 ได้ศึกษา ค้นคว้า เรียนรู้ และทดสอบสิ่งที่ได้ออกแบบตลอดจนเทคโนโลยีการออกแบบวงจรรวม และสามารถนำความรู้ที่ได้มาใช้ประโยชน์ต่อไปในอนาคตได้
- 1.6.3 สามารถออกแบบวงจรรวมต้นแบบให้สามารถทำงานได้ตามความต้องการ
- 1.6.4 เป็นการสร้างและพัฒนาบุคลากรเพื่อรองรับการเติบโตทางด้านเทคโนโลยีวงจรรวมของประเทศ

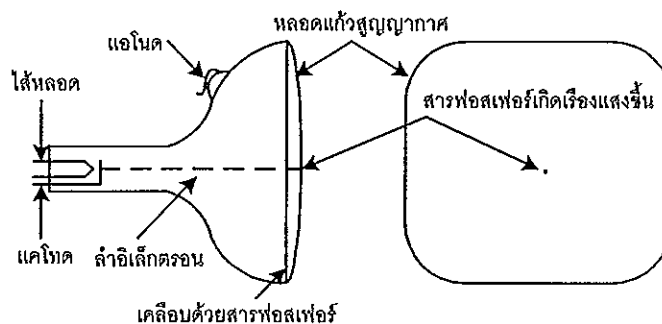
บทที่ 2

ความรู้ทั่วไปเกี่ยวกับสัญญาณภาพโทรทัศน์

ในบทนี้จะเป็นการกล่าวถึงความรู้ทั่วไปเกี่ยวกับสัญญาณภาพไม่ว่าจะเป็นทฤษฎีและหลักการเกิดภาพของเครื่องรับโทรทัศน์ ลักษณะของสัญญาณภาพรวม ระบบ Synchronization และรายละเอียดลำดับของเส้นกวาด ซึ่งจะมีส่วนสำคัญมากในการเป็นข้อพิจารณาสำหรับการออกแบบการทำงานของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลและตัวควบคุมต่อไป

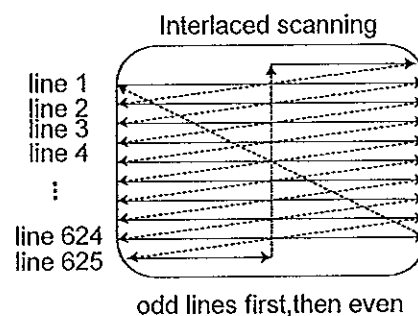
2.1 ทฤษฎีและหลักการเกิดภาพของเครื่องรับโทรทัศน์

โครงสร้างหลอดภาพของจอโทรทัศน์จะมีหลักการทำงานเช่นเดียวกับหลอดสุญญากาศทั่วไปดังภาพประกอบ 2-1 กล่าวคือเมื่อแคโทด (Cathode) ได้รับความร้อนจากการเผาของไส้หลอดจะทำให้แคโทดปล่อยอิเล็กตรอนออกมา อิเล็กตรอนที่ถูกปล่อยออกมาจะถูกดึงให้เข้าไปกระทบกับแอโนด (Anode) ที่หน้าจอภาพเนื่องจากที่แอโนดจะมีแรงดันไฟฟ้าสูงกว่าที่แคโทดมาก โดยที่หน้าจอหลอดภาพจะถูกฉาบไว้ด้วยสารเรืองแสงที่เรียกว่า สารฟอสเฟออร์ (Phosphor) ดังนั้นการกระทบของอิเล็กตรอนกับสารฟอสเฟออร์นี้จะทำให้เกิดจุดสว่างเรืองแสงขึ้น และถ้าหากอิเล็กตรอนดังกล่าวเคลื่อนที่กวาดไปมาในแนวนอนก็จะทำให้เกิดเส้นภาพขึ้น ในขณะที่เดียวกันถ้าหากให้ลำอิเล็กตรอนเคลื่อนที่กวาดไปมาในแนวตั้งด้วยก็จะทำให้เกิดเป็นภาพขึ้นได้ การบังคับให้ลำอิเล็กตรอนกวาดไปมาไม่ว่าในทิศทางใดก็ตามจะใช้สนามแม่เหล็กจากขดลวดสนามแม่เหล็กที่เรียกว่า โยค (Yoke) เป็นตัวควบคุมโดยป้อนกระแสไฟฟ้ารูปฟันเลื่อยเข้าไปซึ่งจะแตกต่างจากหลอดภาพในเครื่องออสซิลอโคปที่ใช้สนามไฟฟ้าเป็นตัวควบคุม



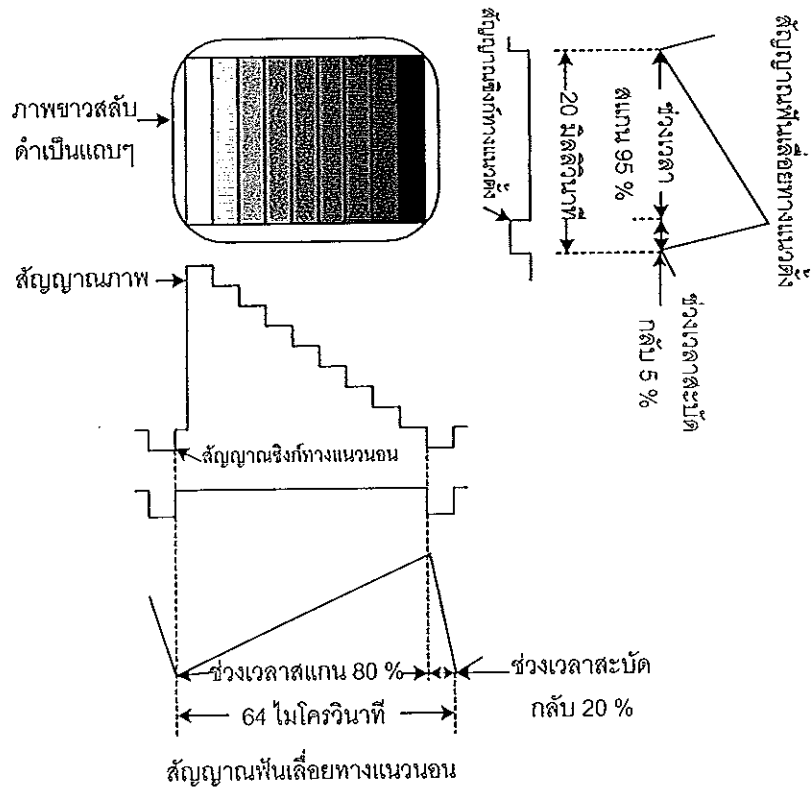
ภาพประกอบ 2-1 แสดงโครงสร้างพื้นฐานของหลอดภาพโทรทัศน์

ภาพที่เกิดบนจอโทรทัศน์นั้นจะประกอบด้วยเส้นภาพเล็กๆเรียงกันอยู่ในแนวนอนมากมายซึ่งเรียกว่า เส้นกวาด (Scanning lines) สำหรับระบบโทรทัศน์ในประเทศไทยซึ่งใช้มาตรฐานระบบ PAL (Phase Alternation Line) นั้นการกวาดภาพทางแนวนอน (Horizontal scanning) มีความถี่เท่ากับ 15,625 เฮิรตซ์และการกวาดภาพทางแนวตั้ง (Vertical scanning) มีความถี่เท่ากับ 50 เฮิรตซ์ การกวาดภาพทางแนวนอนจะใช้เส้นกวาดจำนวน 625 เส้นเรียกว่า 1 ภาพหรือ 1 เฟรม ใน 1 เฟรมจะแบ่งการกวาดออกเป็น 2 ส่วนเรียกว่า ฟิวด์ (Field) โดยแบ่งเป็นฟิวด์เส้นคู่และฟิวด์เส้นคี่ ในแต่ละฟิวด์มีจำนวนเส้นกวาดเท่ากับ 312.5 เส้น การกวาดภาพนั้นจะกระทำที่ฟิวด์เส้นคี่ก่อนโดยเริ่มจากตำแหน่งมุมด้านบนซ้ายสุดของจอภาพแล้วอิเล็กตรอนจะสะบัดกลับ (Retrace) มาด้านบนของจอภาพใหม่และจะทำการกวาดภาพในฟิวด์เส้นคู่จนหมดในลักษณะเดียวกัน การกวาดระหว่างฟิวด์เส้นคี่และฟิวด์เส้นคู่นี้จะทำโดยวางภาพสลับกันซึ่งจะไม่ทับกันเด็ดขาด วิธีการนี้เรียกว่า การกวาดภาพแบบสอดสลับ (Interlaced scanning) ดังแสดงในภาพประกอบ 2-2



ภาพประกอบ 2-2 แสดงการกวาดภาพแบบสอดสลับบนจอเครื่องรับโทรทัศน์

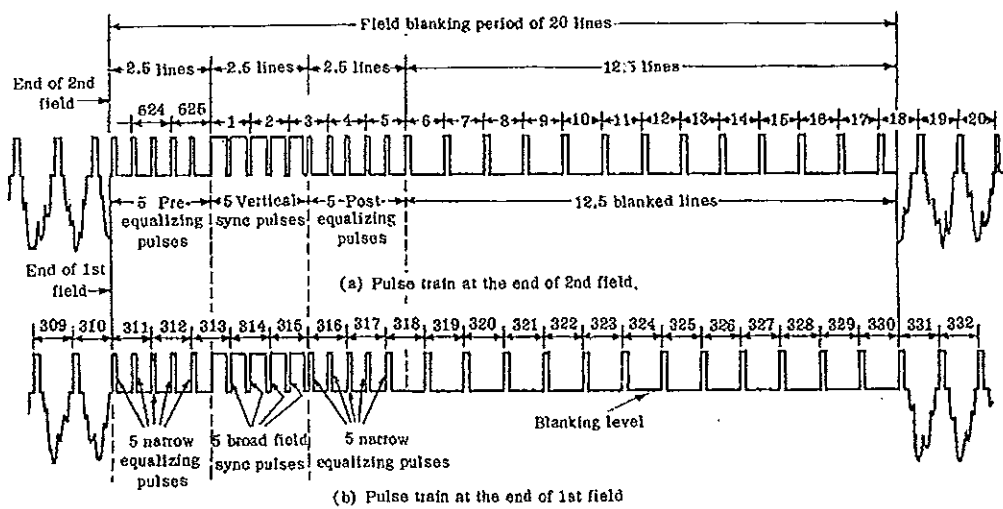
ด้วยหลักการง่าย ๆ ของการสร้างภาพบนจอโทรทัศน์ จากภาพประกอบ 2-3 จะเห็นว่า การสร้างภาพจะต้องประกอบด้วยสัญญาณภาพ สัญญาณซิงค์ทางแนวนอนมีคาบเวลาเท่ากับ 64 ไมโครวินาทีและสัญญาณซิงค์ทางแนวตั้งมีคาบเวลาเท่ากับ 20 มิลลิวินาที การเกิดภาพบนจอภาพจะเกิดเป็นเวลา 80 % ของสัญญาณรูปพื้นเลื่อยที่ใช้กวาดภาพทางแนวนอน ส่วนอีก 20 % จะเป็นช่วงเวลาเส้นกวาดสะบัดกลับเพื่อมาเริ่มต้นการกวาดภาพใหม่ และการกวาดใน 1 ภาพจะเป็นเวลาเท่ากับ 95 % ของสัญญาณรูปพื้นเลื่อยที่ใช้กวาดภาพทางแนวตั้ง ส่วนอีก 5 % จะเป็นช่วงเวลาเส้นกวาดสะบัดกลับเพื่อไปเริ่มต้นกวาดภาพใหม่ต่อไป



ภาพประกอบ 2-3 แสดงหลักการสร้างภาพบนจอโทรทัศน์

2.2 ลักษณะของสัญญาณภาพรวม

ลักษณะของสัญญาณภาพรวม (Composite video signal) ดังแสดงในภาพประกอบ 2-4 ที่สถานีโทรทัศน์ส่งมายังเครื่องรับโทรทัศน์จะประกอบด้วยสัญญาณหลัก 4 สัญญาณ คือ



ภาพประกอบ 2-4 แสดงลักษณะของสัญญาณภาพรวม

(ที่มา : Gulati, R., Monochrome and Colour Television)

2.2.1 สัญญาณภาพและเสียง เป็นสัญญาณที่ส่งไปเพื่อให้เกิดภาพและเสียงขึ้นในเครื่องรับโทรทัศน์

2.2.2 สัญญาณแบลนกกิ่ง (Blanking signal) หรือสัญญาณทำให้ไร้ภาพเป็นสัญญาณที่ส่งมาเพื่อลบเส้นสับตัดกลับอันเกิดจากการกวาดของลำอิเล็กตรอน ซึ่งจะทำให้มองไม่เห็นเส้นกวาดขณะย้อนกลับบนจอภาพมี 2 สัญญาณคือ

2.2.2.1 สัญญาณแบลนกกิ่งทางแนวนอน (Horizontal blanking signal) ทำหน้าที่ลบเส้นสับตัดกลับทางแนวนอนของเส้นกวาดจากด้านขวาของจอภาพให้กลับมารเริ่มต้นกวาดทางด้านซ้ายของจอภาพ โดยมีความถี่เท่ากับเส้นกวาดทางแนวนอนคือ 15,625 เฮิรตซ์

2.2.2.2 สัญญาณแบลนกกิ่งทางแนวตั้ง (Vertical blanking signal) ทำหน้าที่ลบเส้นสับตัดกลับทางแนวตั้งของเส้นกวาดจากด้านล่างของจอภาพให้กลับมารเริ่มต้นกวาดทางด้านบนของจอภาพ มีความถี่เท่ากับเส้นกวาดทางแนวตั้งคือ 50 เฮิรตซ์ โดยจะสับตัดกลับในขณะที่สิ้นสุดการกวาดเส้นคือ 25 ครั้งและสิ้นสุดการกวาดเส้นคู่ 25 ครั้ง

2.2.3 สัญญาณซิงก์ (Sync(hronizing) signal) เป็นสัญญาณที่ควบคุมการกวาดของเครื่องส่งและเครื่องรับโทรทัศน์ให้ทำงานในจังหวะการกวาดที่ตรงกัน ประกอบด้วย 2 สัญญาณคือ

2.2.3.1 สัญญาณซิงก์ทางแนวนอน (Horizontal sync. signal, Hs) เป็นสัญญาณควบคุมการกวาดภาพทางแนวนอน ซึ่งจะขอกล่าวอย่างละเอียดในหัวข้อระบบ Synchronization

2.2.3.2 สัญญาณซิงก์ทางแนวตั้ง (Vertical sync. signal, Vs) เป็นสัญญาณควบคุมการกวาดภาพทางแนวตั้ง ใช้สัญญาณพัลส์จำนวน 5 ลูกระหว่างสัญญาณอีควอลไลซิงซึ่งจะมียู่ในการกวาดทั้งฟิลด์เส้นคู่และเส้นคี่ โดยขอกล่าวอย่างละเอียดในหัวข้อระบบ Synchronization เช่นกัน

2.2.4 สัญญาณอีควอลไลซิง (Equalizing signal) เป็นสัญญาณบังคับรูปร่างของสัญญาณซิงก์ทางแนวตั้งเพื่อให้สามารถคงรูปที่ถูกต้องและช่วยให้การกวาดแบบสลับเส้นเป็นไปอย่างถูกต้องด้วย โดยมีขนาดพัลส์รวมเท่ากับสัญญาณซิงก์ทางแนวตั้งคือ 5 ลูก

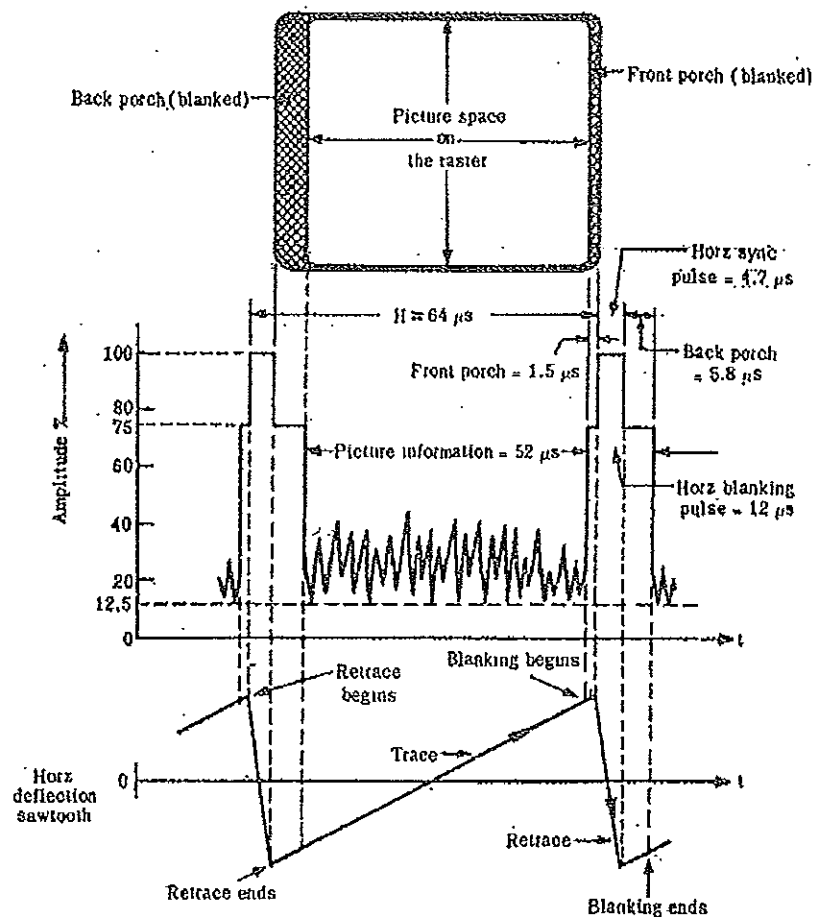
2.3 ระบบ Synchronization

ระบบ Synchronization หมายถึงระบบที่ทำให้เครื่องส่งและเครื่องรับโทรทัศน์เกิดเหตุการณ์ที่สอดคล้องและตรงกันในการส่งและรับสัญญาณภาพกล่าวคือไม่ว่าทางด้านเครื่องส่งโทรทัศน์จะส่งสัญญาณภาพมา มีลักษณะเช่นไรทางด้านเครื่องรับก็จะปรากฏสัญญาณภาพเช่นเดียวกับเครื่องส่งด้วย ซึ่งรวมถึงการส่งสัญญาณภาพจากกล้องวิดีโอมาแสดงผลบนจอมอนิเตอร์ด้วย

เช่นกัน หากในกรณีที่เครื่องส่งและเครื่องรับโทรทัศน์ทำงานไม่ synchronize กันก็จะทำให้เครื่องรับไม่สามารถรับภาพได้หรือสัญญาณที่รับได้อาจมองไม่เป็นภาพก็ได้ ในระบบ Synchronization นี้จะแบ่งสัญญาณออกเป็น 2 ชนิดคือ

2.3.1 สัญญาณซิงค์ทางแนวนอน

โดยปกติในระบบ PAL การทำให้เกิดเป็นภาพที่ต่อเนื่องกันในเวลา 1 วินาทีที่ต้องเกิดภาพเป็นจำนวน 25 ภาพ ดังนั้นเวลาที่ใช้ในการเกิดภาพ 1 ภาพจึงใช้เวลาเพียง $1/25$ วินาที และเนื่องจากในระบบ CCIR (International Radio Consultative Committee) กำหนดให้ใน 1 ภาพประกอบด้วยเส้นกวาดทางแนวนอนทั้งหมด 625 เส้นหรือเกิดสัญญาณ Hs ทั้งหมด 625 ลูก ดังนั้นช่วงเวลาของเส้นกวาดทางแนวนอน 1 เส้น (1 H) จะใช้เวลาเท่ากับ 64 ไมโครวินาที ($1/(25 \times 625)$) หรือคิดเป็นความถี่เท่ากับ 15,625 เฮิรตซ์ จากภาพประกอบ 2-5 แสดงลักษณะของสัญญาณ Hs ในเส้นกวาดทางแนวนอน 1 เส้นโดยมีรายละเอียดทางเวลาในตาราง 2-1



ภาพประกอบ 2-5 แสดงลักษณะของสัญญาณ Hs ในเส้นกวาดทางแนวนอน 1 เส้น
(ที่มา : Gulati, R., Monochrome and Colour Television)

ตาราง 2-1 แสดงรายละเอียดทางเวลาของเส้นกวาดทางแนวนอน 1 เส้น

ช่วงเวลา (Period)	เวลา (ไมโครวินาที)
เส้นกวาดทางแนวนอน 1 เส้น (H)	64
สัญญาณแบล็กกิ้งทางแนวนอน	12 ± 0.3
ความกว้างพัลส์ของสัญญาณ Hs	4.7 ± 0.2
ช่วง Front porch	1.5 ± 0.3
ช่วง Back porch	5.8 ± 0.3
ช่วงที่มองเห็นเส้น (Visible line time)	52

(ที่มา : Gulati, R., Monochrome and Colour Television)

โดยในช่วงเวลาของเส้นกวาดทางแนวนอน 1 เส้นจะประกอบด้วยส่วนต่างๆดังนี้

2.3.1.1 Front porch มีช่วงเวลาเท่ากับ 1.5 ไมโครวินาทีเป็นสัญญาณที่อยู่ระหว่างช่วงสิ้นสุดข้อมูลภาพในแต่ละเส้นกวาดและช่วงเริ่มของสัญญาณ Hs ซึ่งจะทำให้ส่วนของวงจรที่เครื่องรับทำการปรับระดับแรงดันให้อยู่ในระดับการไวภาพ (Blanking level) เมื่อสิ้นสุดข้อมูลภาพของเส้นกวาดเส้นก่อนและก่อนเข้าสู่ช่วงของสัญญาณ Line sync. pulse

2.3.1.2 Line sync. pulse หรือพัลส์ของสัญญาณ Hs มีช่วงเวลาเท่ากับ 4.7 ไมโครวินาทีซึ่งจะทำให้การกวาดภาพในแต่ละเส้นระหว่างเครื่องส่งและเครื่องรับเกิดการ synchronize กันนั่นเอง

2.3.1.3 Back porch มีช่วงเวลาเท่ากับ 5.8 ไมโครวินาทีเป็นช่วงเวลาที่ทำให้เส้นสลับกลับมีเวลามากพอในการสลับกลับก่อนที่จะเริ่มเข้าสู่ข้อมูลภาพ (Picture information) ของเส้นกวาดซึ่งเป็นส่วนที่อยู่ถัดไป

2.3.2 สัญญาณซิงค์ทางแนวตั้ง

ลักษณะของสัญญาณ Vs สามารถดูได้จากภาพประกอบ 2-4 โดยสัญญาณ Vs มีความถี่เท่ากับ 50 เฮิรตซ์หรือคิดเป็นคาบเวลาเท่ากับ 20 มิลลิวินาที

2.4 รายละเอียดลำดับของเส้นกวาด

ใน 1 ภาพที่เกิดจากการกวาดของลำอิเล็กตรอนจะประกอบด้วยเส้นกวาดที่เกิดจากการเบี่ยงเบนทางแนวนอนจำนวน 625 เส้นโดยแบ่งออกเป็น 2 ฟิลด์คือฟิลด์เส้นคี่และฟิลด์เส้นคู่ฟิลด์

ละ 312.5 เส้น รายละเอียดและลำดับการกวาดในแต่ละฟิลด์แสดงในตาราง 2-2

ตาราง 2-2 แสดงรายละเอียดลำดับของเส้นกวาด

เส้นกวาดในฟิลด์เส้นคี่	จำนวน (เส้น)	สัญญาณ
- เส้นที่ 1,2 และครึ่งแรกของเส้นที่ 3	2.5	- สัญญาณซิงก์ทางแนวตั้ง
- ครึ่งหลังของเส้นที่ 3 และเส้นที่ 4,5	2.5	- สัญญาณอีควอไลซิงช่วงหลังของสัญญาณซิงก์ทางแนวตั้ง
- เส้นที่ 6 ถึง 17 และครึ่งแรกของเส้นที่ 18	12.5	- สัญญาณแบลงกิงหรือเส้นสะบัดกลับ
- ครึ่งหลังของเส้นที่ 18 ถึงเส้นที่ 310	292.5	- สัญญาณภาพ
- เส้นที่ 311,312 และครึ่งแรกของเส้นที่ 313	2.5	- สัญญาณอีควอไลซิงช่วงแรกของสัญญาณซิงก์ทางแนวตั้งสำหรับฟิลด์เส้นคู่
เส้นกวาดในฟิลด์เส้นคู่	จำนวน (เส้น)	สัญญาณ
- ครึ่งหลังของเส้นที่ 313 และเส้นที่ 314, 315	2.5	- สัญญาณซิงก์ทางแนวตั้ง
- เส้นที่ 316,317 และครึ่งแรกของเส้นที่ 318	2.5	- สัญญาณอีควอไลซิงช่วงหลังของสัญญาณซิงก์ทางแนวตั้ง
- ครึ่งหลังของเส้นที่ 318 ถึงเส้นที่ 330	12.5	- สัญญาณแบลงกิงหรือเส้นสะบัดกลับ
- เส้นที่ 331 ถึงครึ่งแรกของเส้นที่ 623	292.5	- สัญญาณภาพ
- ครึ่งหลังเส้นที่ 623 และเส้นที่ 624, 625	2.5	- สัญญาณอีควอไลซิงช่วงแรกของสัญญาณซิงก์ทางแนวตั้งสำหรับฟิลด์เส้นคี่

บทที่ 3

การแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

ในบทนี้จะเป็นการกล่าวถึงหลักการแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลรวมทั้งหลักการและส่วนประกอบหลักของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล (Image digitizer) สำหรับภาพขาวดำและการอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP (Enhanced Parallel Port) เพื่อใช้เป็นแนวทางในการออกแบบ Image digitizer สำหรับงานวิจัยนี้ในบทต่อไป

3.1 หลักการแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

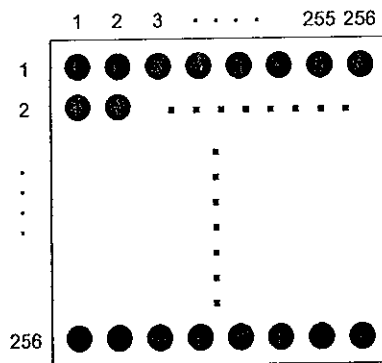
เนื่องจากสัญญาณภาพที่ได้จากกล้องวิดีโอ (Video camera) เป็นสัญญาณแบบแอนะล็อก (Analog signal) ดังนั้นหากเราต้องการเก็บสัญญาณภาพนี้จึงจำเป็นต้องทำการแปลงสัญญาณภาพนี้ให้อยู่ในรูปสัญญาณดิจิทัล (Digital signal) หรือข้อมูลทางดิจิทัลเสียก่อนแล้วจึงสามารถนำข้อมูลดิจิทัลนี้ไปเก็บในหน่วยความจำแรมหรือวิดีโอแรม (Video RAM) ได้ การแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล (Analog-to-Digital conversion) นี้จะใช้วงจรที่เรียกว่าวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (Analog-to-Digital Converter, ADC) แต่เนื่องจากสัญญาณภาพเป็นสัญญาณที่มีความถี่สูงคือมีความถี่เท่ากับ 15.625 กิโลเฮิรตซ์และเวลาที่ใช้ในการกวาดภาพในแต่ละฟิล์มมีค่าประมาณ 20 มิลลิวินาทีซึ่งเป็นความเร็วที่สูงมาก และในกรณีที่เราต้องการเก็บข้อมูลภาพให้ได้ความละเอียดของภาพสูงจึงต้องใช้สัญญาณนาฬิกาในการสุ่มข้อมูลภาพที่มีความถี่สูงด้วยเช่นกัน ดังนั้นวงจร ADC ที่ใช้ในการแปลงสัญญาณภาพจึงเลือกใช้วงจร ADC แบบแฟลช (Flash ADC) ซึ่งมีความเร็วในการแปลงสัญญาณที่เพียงพอสำหรับการกวาดภาพในแต่ละฟิล์ม และด้วยความก้าวหน้าทางเทคโนโลยีทางด้านการผลิตอุปกรณ์ทางอิเล็กทรอนิกส์ในปัจจุบันทำให้วงจร Flash ADC ที่มีจำหน่ายในท้องตลาดมีราคาที่ไม่สูงนักและมีความเร็วสูงมากซึ่งทำให้งานทางด้านการเก็บภาพมีประสิทธิภาพมากขึ้น

3.1.1 ลักษณะการเก็บข้อมูลภาพดิจิทัลขนาด 256×256 จุดต่อภาพ

การเก็บข้อมูลภาพนั้นเริ่มพัฒนามาจากการเก็บข้อมูลภาพขนาด 64×64 จุดต่อภาพซึ่งเป็นความละเอียดที่ค่อนข้างต่ำทั้งนี้มิใช่สาเหตุมาจากข้อจำกัดทางด้านความเร็วและราคาของ

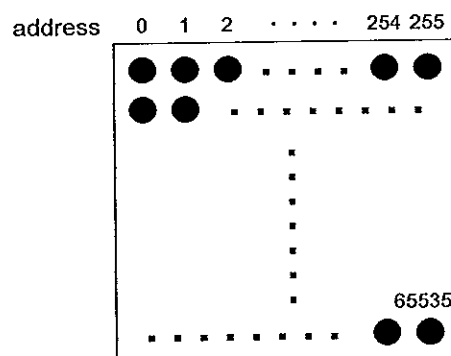
อุปกรณ์ที่ใช้ในสมัยนั้น ต่อมาได้มีการพัฒนาให้การเก็บข้อมูลภาพมีความละเอียดสูงมากขึ้นจนกลายมาเป็นการเก็บภาพขนาด 256×256 จุดต่อภาพ

สำหรับการเก็บภาพดิจิทัลขนาด 256×256 จุดต่อภาพก็คือการเก็บข้อมูลภาพในแต่ละเส้นกวาดจำนวน 256 จุดและเก็บข้อมูลภาพจำนวน 256 เส้นดังในภาพประกอบ 3-1 ซึ่งความละเอียดขนาดนี้จะเหมาะสำหรับการนำข้อมูลภาพไปใช้ในงานที่ไม่ต้องการความละเอียดของภาพมากนักและหน่วยความจำที่ใช้จะมีขนาดเท่ากับ 256×256 หรือ 65536 ไบต์หรือ 64 กิโลไบต์



ภาพประกอบ 3-1 แสดงลักษณะการเก็บภาพข้อมูลภาพขนาด 256×256 จุดต่อภาพ

การจัดพื้นที่หน่วยความจำในการเก็บภาพจะเริ่มจากตำแหน่งมุมบนซ้ายมือสุดเป็นแอดเดรสที่ 0 ของหน่วยความจำและนับไปเรื่อยๆทางขวามือจนถึงตำแหน่งมุมขวาสุดซึ่งตรงกับแอดเดรสที่ 255 จากนั้นก็เริ่มเก็บภาพในเส้นต่อไปในตำแหน่งซ้ายมือสุดอีกซึ่งตรงกับแอดเดรสที่ 256 โดยทำการเก็บภาพไปเรื่อยๆในลักษณะนี้จนถึงตำแหน่งมุมขวาล่างสุดซึ่งจะตรงกับแอดเดรสที่ 65535 พอดีดังในภาพประกอบ 3-2



ภาพประกอบ 3-2 แสดงการจัดพื้นที่หน่วยความจำในการเก็บข้อมูลภาพ

ขนาด 256×256 จุดต่อภาพ

3.1.2 ลักษณะการเก็บข้อมูลภาพดิจิทัลขนาด 512×512 จุดต่อภาพ

การเก็บภาพดิจิทัลขนาด 512×512 จุดต่อภาพนี้ก็คือการเก็บข้อมูลภาพในแต่ละเส้นกวาดจำนวน 512 จุดและเก็บข้อมูลภาพจำนวน 512 เส้น ซึ่งจะให้ภาพที่มีความละเอียดมากขึ้น จึงเหมาะสำหรับภาพที่ต้องการใช้ในงานที่ต้องการความละเอียดของภาพสูง และหน่วยความจำที่ใช้ในการเก็บภาพจะมีขนาดมากกว่าการเก็บภาพขนาด 256×256 จุดต่อภาพถึง 4 เท่าคือใช้หน่วยความจำขนาด 256 กิโลไบต์

สำหรับงานวิจัยนี้เลือกลักษณะการเก็บภาพดิจิทัลขนาด 256×256 จุดต่อภาพซึ่งเพียงพอสำหรับการประยุกต์ใช้งานในระบบการตรวจสอบอัตโนมัติเชิงเวลาจริงโดยใช้กล้องวิดีโอในงานวิจัยเรื่องเครื่องตรวจขวดอัตโนมัติ (จรรยา ไชยนิศย์, 2542) และหากต้องการความละเอียดของภาพที่สูงขึ้นก็สามารถปรับเปลี่ยนได้โดยการเพิ่มความถี่ของสัญญาณนาฬิกาและขนาดหน่วยความจำที่ใช้

3.2 หลักการของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

ในปัจจุบันการแปลงสัญญาณภาพที่ได้จากกล้องวิดีโอหรือเครื่องรับโทรทัศน์ให้เป็นสัญญาณดิจิทัลหรือภาพดิจิทัลที่สามารถนำไปใช้ประโยชน์ในด้านต่างๆ เช่น ด้านการประมวลผลภาพ (Image processing) หรือการจดจำรูปแบบ (Pattern recognition) เป็นต้น สามารถทำได้โดยใช้วงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลหรือที่เรียกว่า Image digitizer แต่ก่อนจะเข้าสู่รายละเอียดของ Image digitizer ขอกล่าวถึงคำศัพท์หรือความหมายของคำที่เกี่ยวข้องกับ Image digitizer ก่อนดังนี้

3.2.1 Digitizer หรืออาจมีชื่อเรียกที่แตกต่างกันออกไปแต่ยังคงทำหน้าที่หลักเหมือนกัน เช่น Frame grabber, Video digitizer หรือ Video capture เป็นต้น เป็นอุปกรณ์ที่สามารถ digitize สัญญาณภาพวิดีโอให้เป็นสัญญาณดิจิทัลได้ตามเวลาจริง (Real time) ซึ่งสามารถ digitize สัญญาณภาพได้ทั้งสองฟิลด์ ยกเว้น Digitizer ชนิด Field grabber ที่สามารถ digitize ภาพได้เพียงฟิลด์ใดฟิลด์หนึ่งของสัญญาณภาพเท่านั้นและทำให้ความละเอียดของภาพลดลงครึ่งหนึ่ง การแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลแบบเวลาจริง (Real time digitization) นี้จะยอมให้วัตถุที่กล้องวิดีโอจับภาพอยู่สามารถเคลื่อนที่ได้โดยไม่มีผลกระทบต่อภาพที่ถูก digitized แต่มีข้อแม้ว่าเวลาที่ใช้ในการ digitize สัญญาณภาพใน 1 เฟรมจะต้องน้อยกว่า 1/30 วินาที (ในระบบ NTSC) หรือน้อยกว่า 1/25 วินาที (ในระบบ PAL) และวัตถุจะต้องไม่เคลื่อนไหวในระหว่างที่

Digitizer กำลังเก็บภาพอยู่ ดังนั้น Real Time digitizer จึงเหมาะสำหรับสัญญาณภาพที่มาจากเครื่องรับโทรทัศน์หรือเครื่องบันทึกภาพแบบเทปตลับ (Video Cassette Tape Recorder, VCR) เนื่องจากสามารถเก็บภาพได้ทันที นอกจากนี้ยังมี Digitizer อีกชนิดหนึ่งคือ Still-frame digitizer โดยภาพที่ต้องการเก็บด้วย digitizer ชนิดนี้จะต้องไม่เคลื่อนไหวในระหว่างขั้นตอนการเก็บภาพ เนื่องจากการเคลื่อนไหวของวัตถุในภาพจะส่งผลต่อความเพี้ยน (Distortion) ของภาพนั่นเอง

3.2.2 ความละเอียดของภาพ (Resolution) เป็นตัวเลขที่ใช้ระบุจำนวนจุดภาพ (Pixels หรือ Picture elements) ของภาพที่ Digitizer สามารถเก็บได้ โดยทั่วไปสำหรับ Real time digitizer มักใช้ตัวเลขแบบเมตริกซ์จัตุรัส เช่น ความละเอียดขนาด 256×256 จุดต่อภาพก็จะมี ความหมายว่าใน 1 ภาพจะมีจำนวนจุดภาพทางแนวนอนเท่ากับ 256 จุดและจำนวนเส้นภาพทางแนวตั้งเท่ากับ 256 เส้น เป็นต้น สำหรับมิติของเมตริกซ์ (Matrix dimensions) จะใช้จำนวนฐานสอง (Binary number) หรือจำนวนเลขยกกำลังของสองเพื่อให้ง่ายต่อการออกแบบทางด้านฮาร์ดแวร์ของวงจร สำหรับ Still-frame digitizer บางตัวอาจไม่ระบุความละเอียดของภาพเป็นเลขยกกำลังของสองเนื่องจากใน Digitizer เองไม่มีการใช้หน่วยความจำสำหรับการเก็บภาพแต่จะใช้หน่วยความจำที่อยู่ในเครื่องคอมพิวเตอร์สำหรับเก็บภาพแทน

โดยปกติแล้ว Digitizer ก็เหมือนกับอะแดปเตอร์สำหรับแสดงผล (Display adapter) ทั่วไปคือมีการระบุถึงจำนวนบิตที่แน่นอนในแต่ละจุดภาพเพื่อแสดงถึงจำนวนการสุ่มภาพในแต่ละจุด ซึ่งจำนวนการสุ่มภาพในแต่ละจุดนี้จะแทนความเข้ม (Intensity) ของภาพ ณ ตำแหน่งที่จุดภาพนั้นถูก digitized ได้ โดยจำนวนบิตยังมีค่ามากเท่าไรหรือความถูกต้องของการแสดงภาพดิจิทัลนั้นก็ยิ่งมากขึ้นเท่านั้นกล่าวคือภาพดิจิทัลที่ได้จะมีความใกล้เคียงกับภาพเดิมนั่นเอง ดังนั้นจำนวนบิตนี้จึงสามารถเป็นตัวบ่งบอกถึงราคาของ Digitizer แต่ละตัวได้เนื่องจากจำนวนบิตที่มีค่ามากย่อมส่งผลให้ขนาดหรือจำนวนบิตของวงจร ADC ที่ใช้มากขึ้นตามไปด้วยเพราะวงจร ADC ที่มีจำนวนบิตมากจะใช้จำนวนของวงจรเปรียบเทียบ (Comparator) ภายในมากกว่านั่นเอง โดยทั่วไปจำนวนบิตที่ใช้อาจมีค่าตั้งแต่ 4 6 8 9 10 ถึง 11 บิตทำให้ได้ความเข้มในแต่ละจุดภาพเท่ากับ 16 64 256 512 1024 2048 ระดับตามลำดับซึ่งแน่นอนว่าความเข้ม 2048 ระดับย่อมสามารถแสดงภาพเดิมได้ใกล้เคียงมากกว่าความเข้มเพียงแค่ 16 ระดับ แต่จำนวนบิตที่ให้ความเข้มของภาพที่เหมาะสมและเป็นที่ยอมรับกันก็คือจำนวน 8 บิตต่อจุดภาพทั้งนี้เนื่องจากสายตาคนเราโดยปกติแล้วจะไม่สามารถแยกความแตกต่างระหว่างภาพที่ใช้จำนวนบิตต่อจุดภาพตั้งแต่ 9-11 บิตกับภาพที่ใช้จำนวนบิตต่อจุดภาพเท่ากับ 8 บิตได้เท่าไรนักจึงไม่จำเป็นต้องใช้จำนวนบิตที่สูงเกินไป และเหตุผลที่สำคัญอีกประการหนึ่งก็คือ Digitizer ที่ใช้จำนวนบิตต่อจุดภาพตั้งแต่ 9-11 บิตจะมี

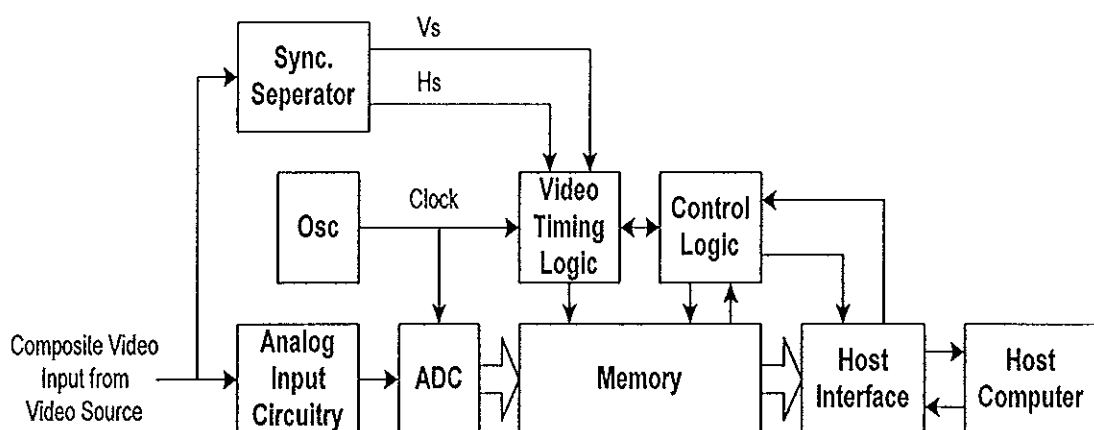
ราคาสูงกว่า Digitizer ที่ใช้จำนวนบิตต่อจุดภาพเท่ากับ 8 บิตมากเช่น วงจร ADC ขนาด 11 บิตจะใช้จำนวนวงจรเปรียบเทียบเท่ากับ 2048 ตัวในขณะที่วงจร ADC ขนาด 8 บิตจะใช้จำนวนวงจรเปรียบเทียบเพียง 256 ตัวซึ่งต่างกันถึง 8 เท่าทำให้ราคาของวงจร ADC สูงกว่ามาก

นอกจากนี้ Digitizer ที่ใช้งานในปัจจุบันยังมีทั้งชนิดที่สามารถใช้งานได้กับภาพขาวดำและชนิดที่ใช้งานได้กับภาพสี โดย Digitizer ชนิดหลังนี้จะประกอบด้วยวงจร ADC และหน่วยความจำที่ใช้อย่างละ 3 ชุดสำหรับการเก็บภาพในแต่ละสีคือ สีแดง สีเขียวและสีน้ำเงิน และการ digitize ในแต่ละจุดภาพจะเกิดขึ้น 3 ครั้งเช่นกันโดยอาจเกิดขึ้นพร้อมกันแบบขนานหรือเกิดขึ้นทีละครั้งแบบอนุกรมก็ได้ แต่สำหรับ Digitizer ชนิดแรกนี้จะใช้วงจร ADC และหน่วยความจำสำหรับเก็บภาพอย่างละ 1 ชุดเท่านั้น โดยภาพที่ได้จะเป็นภาพแบบ Continuous-tone image เนื่องจากค่าความเข้มของจุดภาพ (Pixel-intensity value) ที่เป็นไปได้จะถูกแทนด้วยระดับเทา (Shades of gray หรือ Gray level) ที่ต่างกันตั้งแต่สีดำไปจนถึงสีขาวซึ่งมีความสัมพันธ์กับจำนวนบิตต่อจุดภาพตามที่ได้กล่าวไปแล้ว

สำหรับในงานวิจัยนี้เป็นการสร้าง Digitizer สำหรับภาพขาวดำดังนั้นจึงเลือกจำนวนบิตต่อจุดภาพมีค่าเท่ากับ 8 บิตเพื่อให้สามารถแสดงระดับเทาในแต่ละจุดภาพได้ทั้งหมด 256 ระดับ

3.3 ส่วนประกอบของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

Image digitizer โดยทั่วไปจะมีส่วนประกอบหลักดังแสดงในภาพประกอบ 3-3 โดยรายละเอียดของแต่ละส่วนมีดังนี้



ภาพประกอบ 3-3 แสดงส่วนประกอบหลักของ Image digitizer โดยทั่วไป

3.3.1 ส่วนวงจรสัญญาณอินพุต (Analog Input Circuitry)

ทำหน้าที่รับสัญญาณภาพรวม (Composite video signal) จากอุปกรณ์กำเนิดสัญญาณภาพต่างๆ เช่น กล้องวิดีโอ เครื่องรับโทรทัศน์หรือเครื่องบันทึกภาพแบบเทปตลับ เป็นต้น เพื่อเป็นสัญญาณอินพุตให้กับ Digitizer โดยในส่วนี้อาจประกอบด้วยส่วนปรับสัญญาณ (Signal conditioner) เพื่อทำหน้าที่ปรับแรงดันของสัญญาณภาพให้มีขนาดที่เหมาะสมก่อนเข้าสู่ส่วนวงจร ADC ต่อไป

3.3.2 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (Analog-to-Digital Converter, ADC)

ทำหน้าที่แปลงสัญญาณภาพรวมซึ่งเป็นสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล วงจร ADC ที่ใช้ใน digitizer มักเป็นชนิด Flash ADC เนื่องจากใช้เวลาในการแปลงสัญญาณ (Conversion time) ที่เร็วเพียงพอต่อการกวาดภาพในแต่ละฟิลด์ ซึ่งอัตราในการแปลงสัญญาณภาพนี้จะถูกควบคุมด้วยสัญญาณนาฬิกาจากส่วนวงจรกำเนิดสัญญาณนาฬิกาในส่วนถัดไป นอกจากนี้จำนวนบิตของวงจร ADC จะมีผลต่อระดับความเข้มของภาพในแต่ละจุดภาพและราคาของ Digitizer ตามที่ได้กล่าวไว้ในหัวข้อ 3.2

3.3.3 วงจรกำเนิดสัญญาณนาฬิกา (Oscillator, Osc)

ทำหน้าที่กำเนิดสัญญาณนาฬิกา (Clock) เพื่อกำหนดอัตราในการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลให้กับวงจร ADC รวมทั้งทำหน้าที่ควบคุมกระบวนการต่างๆที่เกิดขึ้นภายใน Digitizer ทั้งในกระบวนการเขียนและอ่านข้อมูลดิจิทัลจากหน่วยความจำ

3.3.4 วงจรแยกสัญญาณซิงก์ (Sync. Separator)

เนื่องจากการส่งสัญญาณภาพที่ได้จากกล้องวิดีโอเพื่อให้ไปปรากฏบนจอแสดงผลนั้น จะเป็นการส่งข้อมูลแบบอสังเวลาและความถี่ระหว่างข้อมูลที่ส่งมากับข้อมูลที่รับได้ ในทำนองเดียวกันการเก็บภาพดิจิทัลลงในหน่วยความจำก็จำเป็นต้องให้หน่วยความจำสามารถเก็บภาพที่ถูกต้องตามลำดับของเส้นกวาดด้วยเช่นกัน ดังนั้นเพื่อให้เกิดเหตุการณ์ที่สอดคล้องกันดังกล่าวจึงจำเป็นต้องใช้สัญญาณซิงก์ในการกำหนดตำแหน่งการเกิดภาพ วงจรแยกสัญญาณซิงก์จึงถูกใช้เพื่อทำหน้าที่แยกสัญญาณซิงก์ทางแนวนอน (Horizontal sync. signal, Hs) และสัญญาณซิงก์ทางแนวตั้ง (Vertical sync. signal, Vs) ออกจากสัญญาณภาพรวมเพื่อใช้สร้างแอดเดรสทางแถวและคอลัมน์ให้หน่วยความจำซึ่งอยู่ในส่วนควบคุมการเก็บภาพดิจิทัล

3.3.5 วงจรควบคุมการเก็บภาพดิจิทัล (Video Timing Logic)

เป็นส่วนที่ควบคุมกระบวนการในการเก็บข้อมูลดิจิทัลจากวงจร ADC ลงหน่วยความจำให้ตรงกับส่วนที่เป็นข้อมูลภาพ (Picture information) ของเส้นกวาดแต่ละเส้น โดยอาจเป็น

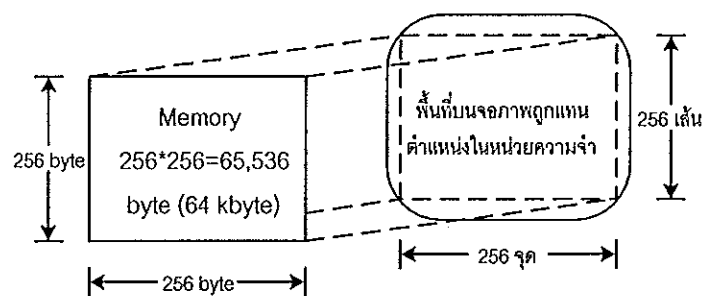
วงจรถอดจิกเกต (Logic gates) ทั่วไปเช่น วงจรนับ (Counter) เพื่อทำหน้าที่สร้างแอดเดรสทางแถวและคอลัมน์ให้หน่วยความจำซึ่งจะถูกควบคุมด้วยสัญญาณซิงก์ทั้งสองสัญญาณจากส่วนแยกสัญญาณซิงก์และสัญญาณ clock จากส่วนวงจรถอดสัญญาณนาฬิกาอีกทีหนึ่ง

3.3.6 วงจรควบคุม (Control Logic)

ทำหน้าที่ควบคุมการติดต่อระหว่างหน่วยความจำกับส่วนอินเทอร์เฟซกับเครื่องคอมพิวเตอร์หลักเพื่อให้ Digitizer สามารถทำกระบวนการทั้งในการเขียนข้อมูลภาพลงหน่วยความจำหรืออ่านข้อมูลภาพเข้าสู่ส่วนอินเทอร์เฟซกับเครื่องคอมพิวเตอร์หลักโดยเกี่ยวข้องกับส่วนควบคุมเวลาด้วยเช่นกัน

3.3.7 หน่วยความจำ (Memory)

ทำหน้าที่เก็บสัญญาณดิจิทัลที่ได้จากวงจร ADC เพื่อให้สามารถแทนตำแหน่งขององค์ประกอบภาพในแต่ละตำแหน่งได้ กล่าวคือหากสมมติว่าต้องการความละเอียดของภาพขนาด 256×256 จุดต่อภาพคือกำหนดให้แสดงเส้นภาพทางแนวนอนได้ 256 เส้นและในแต่ละเส้นภาพจะมีรายละเอียดของจุดภาพจำนวน 256 จุด ดังนั้นหน่วยความจำที่ใช้จึงมีขนาดเท่ากับ 64 กิโลไบต์ (ให้องค์ประกอบภาพ 1 ตำแหน่งเท่ากับหน่วยความจำแรม 1 ไบต์) ดังแสดงในภาพประกอบ 3-4 ซึ่งทำให้สามารถอ้างถึงตำแหน่งขององค์ประกอบภาพได้ทุกตำแหน่ง ณ จุดใดก็ได้โดยอาศัยการอ้างตำแหน่งของหน่วยความจำในตำแหน่งนั้นและทำให้สามารถส่งข้อมูลดิจิทัลนี้ไปยังเครื่องคอมพิวเตอร์เพื่อการใช้งานต่อไปได้



ภาพประกอบ 3-4 แสดงการอ้างถึงตำแหน่งขององค์ประกอบภาพ

3.3.8 ส่วนอินเทอร์เฟซกับเครื่องคอมพิวเตอร์หลัก (Host Interface)

ทำหน้าที่เชื่อมต่อวงจรภายนอกกับเครื่องคอมพิวเตอร์เพื่อโอนย้ายข้อมูลดิจิทัลที่เก็บไว้ในหน่วยความจำ การติดต่อกับเครื่องคอมพิวเตอร์นี้สามารถทำได้หลายลักษณะเช่น การติดต่อผ่านทางสล๊อต ISA bus, สล๊อต PCI และทางพอร์ตนาน (Parallel port) หรือพอร์ตปริ้นต์เตอร์ (Printer port) เป็นต้น สำหรับงานวิจัยนี้เลือกการอินเทอร์เฟซกับเครื่องคอมพิวเตอร์โดยผ่านทาง

พอร์ตขนานเนื่องจากเครื่องคอมพิวเตอร์ในปัจจุบันมักมีพอร์ตขนานเพื่อให้ผู้ใช้สามารถเชื่อมต่อกับอุปกรณ์ภายนอกหรืออุปกรณ์รอบนอก (Peripheral device) อื่นๆได้อย่างสะดวก อีกทั้งการใช้งานสล็อต ISA bus ก็ไม่เป็นที่นิยมแล้วในปัจจุบัน และส่วนการใช้งานสล็อต PCI ก็มีความยุ่งยากมากกว่าเมื่อเทียบกับการใช้งานพอร์ตขนาน การอินเตอร์เฟสกับพอร์ตขนานนี้จะเลือกใช้โหมดการทำงานแบบ EPP ซึ่งจะขอล่าวอย่างละเอียดในหัวข้อต่อไป

3.3.9 เครื่องคอมพิวเตอร์หลัก (Host Computer)

ทำหน้าที่รับนำภาพที่เก็บได้มาประมวลผลเพื่อให้ประโยชน์ในงานที่เกี่ยวข้องกับภาพต่อไป

3.4 การอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP

ปัจจุบันการใช้พอร์ตขนานของเครื่องคอมพิวเตอร์ในการเชื่อมต่อกับอุปกรณ์ภายนอกที่ไม่ใช่อุปกรณ์รอบนอกจำพวก เครื่องพิมพ์ พล็อตเตอร์ กำลังเป็นที่นิยมมากขึ้นเนื่องจากเครื่องคอมพิวเตอร์สมัยใหม่มีขีดความสามารถที่สูงขึ้น มีอัตราการถ่ายโอน (Transfer rate) ข้อมูลที่เร็วมากขึ้น ในขณะที่ขาคสัญญาณข้อมูลของพอร์ตขนานรุ่นใหม่สามารถถ่ายโอนข้อมูลได้สองทิศทาง ทำให้มีการนำพอร์ตขนานมาใช้ประโยชน์มากขึ้น สำหรับงานวิจัยนี้เลือกการอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP ซึ่งมีรายละเอียดดังต่อไปนี้

โหมดการทำงานแบบ EPP เป็นโหมดการทำงานแบบหนึ่งของพอร์ตขนานซึ่งถูกออกแบบโดยความร่วมมือกันระหว่างบริษัท Intel, Xircom และ Zenith Data Systems เริ่มแรกโหมดการทำงานแบบ EPP จะถูกกำหนดเป็นมาตรฐาน 1.7 (EPP 1.7 Standard) ต่อมาในปี 1994 โหมดการทำงานแบบ EPP ก็ได้ถูกรวมในมาตรฐานของ IEEE 1284 ดังนั้นโหมดการทำงานแบบ EPP จึงมี 2 มาตรฐานคือโหมดการทำงานแบบ EPP มาตรฐาน 1.7 และโหมดการทำงานแบบ EPP มาตรฐาน 1.9 (EPP 1.9 Standard) ซึ่งมีข้อแตกต่างที่ส่งผลต่อการทำงานของอุปกรณ์รอบนอกโดยปกติโหมดการทำงานแบบ EPP จะมีอัตราการถ่ายโอนข้อมูลประมาณ 500 กิโลไบต์ต่อวินาที (KB/s) ถึง 1.5 หรือ 2 เมกกะไบต์ต่อวินาที (MB/s) ซึ่งสามารถทำได้โดยฮาร์ดแวร์ที่มีอยู่ในพอร์ตขนานเพื่อสร้างการ Handshaking หรือการ Strobing มากกว่าจะใช้ซอฟต์แวร์เป็นผู้ทำเหมือนในกรณีโหมดการทำงานแบบ Centronics

3.4.1 คุณสมบัติทางฮาร์ดแวร์ของโหมดการทำงานแบบ EPP

จากตาราง 3-1 แสดงชื่อและหน้าที่ของสัญญาณในโหมดการทำงานแบบ EPP โดยเปรียบเทียบกับโหมดการทำงานแบบมาตรฐาน (Standard Parallel Port, SPP)

ตาราง 3-1 แสดงชื่อและหน้าที่ของสัญญาณในโหมดการทำงานแบบ EPP

ขา (Pin)	SPP signal	EPP signal	In/Out	หน้าที่ (Function)
1	Strobe	Write	Out	ลอจิก '0' แสดงโหมดการเขียน ลอจิก '1' แสดงโหมดการอ่าน
2-9	Data 0-7	Data 0-7	In-Out	บัสข้อมูลแบบ 2 ทิศทาง (Bidirection)
10	Ack	Interrupt	In	เส้นอินเทอร์รัปต์ โดยเกิดอินเทอร์รัปต์ที่ ขอบขาขึ้นของสัญญาณ (Rising edge)
11	Busy	Wait	In	ใช้สำหรับการทำ Handshaking โดย รอบการทำงานของโหมด EPP จะเริ่ม เมื่อสัญญาณเป็นลอจิก '0' และสิ้นสุด เมื่อสัญญาณเป็นลอจิก '1'
12	Paper out/end	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
13	Select	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
14	Auto linefeed	Data strobe	Out	ลอจิก '0' แสดงว่าเกิดการโอนย้ายข้อมูล
15	Error/Fault	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
16	Initialize	Reset	Out	ลอจิก '0' เป็นสัญญาณรีเซ็ต
17	Select printer	Address strobe	Out	ลอจิก '0' แสดงว่าเกิดการโอนย้ายแอด เดรส
18-25	Ground	Ground	GND	กราวด์

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

หมายเหตุ สัญญาณ Paper out, Select printer และ Error ไม่ได้ใช้ในการทำ Handshaking ดังนั้นผู้ใช้จึงสามารถใช้งานได้โดยอิสระ ส่วนสถานะของสัญญาณเหล่านี้สามารถดูได้ที่รีจิสเตอร์หรือพอร์ตแสดงสถานะ (Status port) ของโหมดการทำงานแบบมาตรฐาน

3.4.2 พอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP

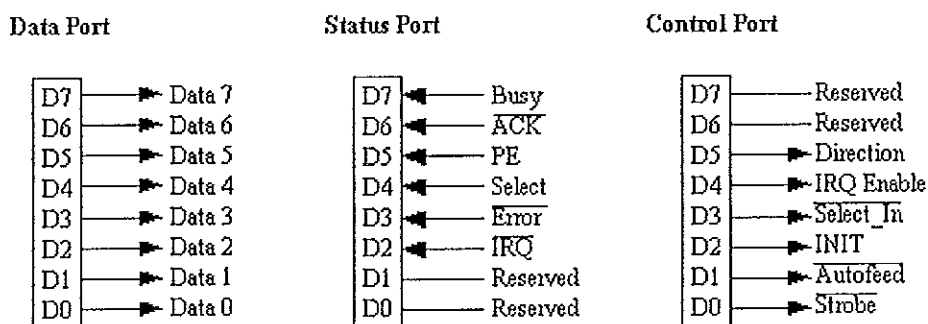
โหมดการทำงานแบบ EPP จะมีพอร์ตที่เพิ่มเติมจากโหมดการทำงานแบบมาตรฐานดังแสดงในตาราง 3-2 และสัญญาณในพอร์ตต่างๆแสดงในภาพประกอบ 3-5

ตาราง 3-2 แสดงพอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP

แอดเดรส (Address)	ชื่อพอร์ต (Port name)	อ่าน/เขียน (Read/Write)
Base address + 0	พอร์ตข้อมูล (Data port) (SPP)	Write
Base address + 1	พอร์ตสถานะ (Status port) (SPP)	Read
Base address + 2	พอร์ตควบคุม (Control port) (SPP)	Write
Base address + 3	พอร์ตแอดเดรส (Address port) (EPP)	Read/Write
Base address + 4	พอร์ตข้อมูล (Data port) (EPP)	Read/Write
Base address + 5	ไม่ระบุ (16/32 bit Transfer)	-
Base address + 6	ไม่ระบุ (16/32 bit Transfer)	-
Base address + 7	ไม่ระบุ (16/32 bit Transfer)	-

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

หมายเหตุ Base address เป็นแอดเดรสเริ่มต้นสำหรับการติดต่อกับพอร์ตขนาน โดยปกติมักอยู่ที่แอดเดรส 0x378

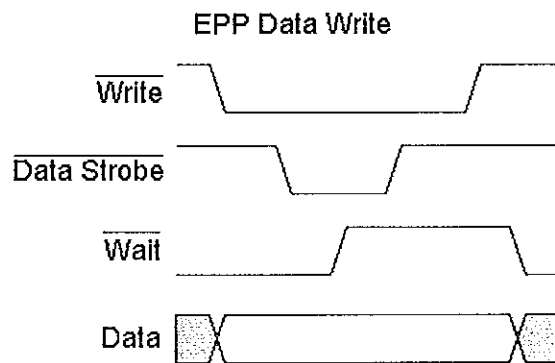


ภาพประกอบ 3-5 แสดงสัญญาณในพอร์ตต่างๆของพอร์ตขนาน

3.4.3 หลักการทำ Handshaking ของโหมดการทำงานแบบ EPP

การใช้งานพอร์ตขนานในโหมดการทำงานแบบ EPP เพื่อโอนย้ายข้อมูลนั้นต้องปฏิบัติตามหลักการ Handshaking ซึ่งต้องการเพียงแค่ให้ฮาร์ดแวร์เป็นตัวจัดการ โดยสัญญาณที่ใช้ในการทำ Handshaking จะมี 3 สัญญาณคือสัญญาณ Write, Data strobe และ Wait ซึ่งการโอนย้ายนี้จะแบ่งรอบการทำงานออกเป็น 2 ลักษณะคือรอบการเขียนข้อมูล (Data) หรือแอดเดรส (Address) และรอบการอ่านข้อมูลหรือแอดเดรส ดังรายละเอียดต่อไปนี้

3.4.3.1 รอบการเขียนข้อมูล (Data write cycle)



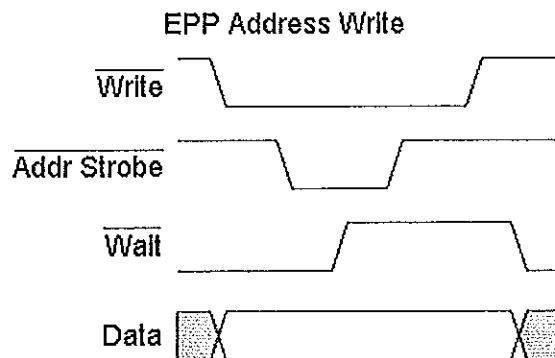
ภาพประกอบ 3-6 แสดงลักษณะของรอบการเขียนข้อมูล

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 3-6 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Write จะมีค่าเป็นลอจิก '0' (แสดงว่าอยู่ในโหมดการเขียน)
- ข้อมูลพร้อมอยู่ที่เส้นสัญญาณข้อมูลทั้ง 7 เส้น (Data lines 0-7)
- สัญญาณ Data strobe จะถูกส่งออกมา (เป็นลอจิก '0') ถ้าสัญญาณ Wait เป็นลอจิก '0' (เริ่มรอบการเขียนข้อมูล)
- Host จะรอการตอบรับ (Acknowledgement) โดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก '1' (เพื่อหยุดรอบการทำงาน)
- สัญญาณ Data strobe จะกลับเป็นลอจิก '1'
- สิ้นสุดรอบการเขียนข้อมูล

3.4.3.2 รอบการเขียนแอดเดรส (Address write cycle)



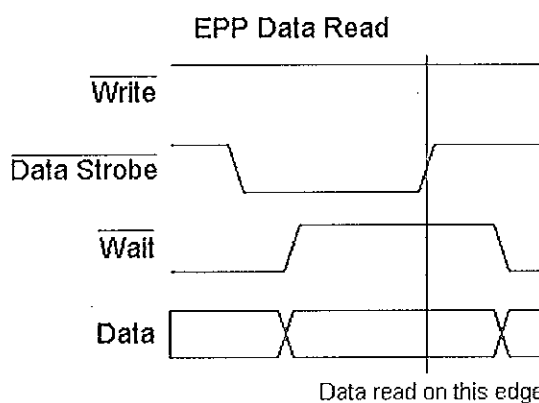
ภาพประกอบ 3-7 แสดงลักษณะของรอบการเขียนแอดเดรส

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 3-7 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตแอดเดรสที่ Base address+3
- สัญญาณ Write จะมีค่าเป็นลอจิก '0' (แสดงว่าอยู่ในโหมดการเขียน)
- แอดเดรสพร้อมอยู่ที่เส้นสัญญาณข้อมูลทั้ง 7 เส้น
- สัญญาณ Address strobe จะถูกส่งออกมา (เป็นลอจิก '0') ถ้าสัญญาณ Wait เป็นลอจิก '0' (เพื่อเริ่มรอบการเขียนแอดเดรส)
- Host จะรอการตอบรับโดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก '1' (เพื่อหยุดรอบการทำงาน)
- สัญญาณ Address strobe จะกลับเป็นลอจิก '1'
- สิ้นสุดรอบการเขียนแอดเดรส

3.4.3.3 รอบการอ่านข้อมูล (Data read cycle)



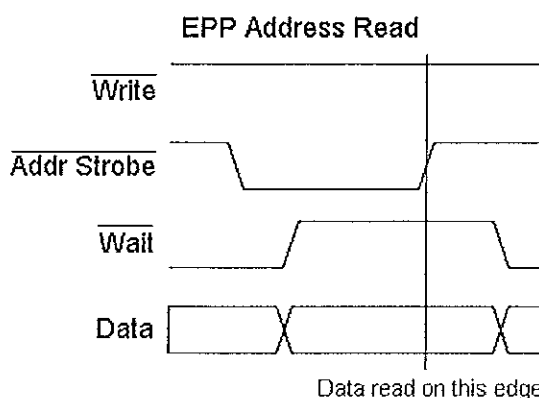
ภาพประกอบ 3-8 แสดงลักษณะของรอบการอ่านข้อมูล

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 3-8 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Data strobe จะถูกส่งออกมา (เป็นลอจิก '0') ถ้าสัญญาณ Wait เป็นลอจิก '0' (เพื่อเริ่มรอบการอ่านข้อมูล)
- Host จะรอการตอบรับโดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก '1'
- ข้อมูลจะถูกอ่านจากขาข้อมูล
- สัญญาณ Data strobe จะเป็นลอจิก '1'
- สิ้นสุดรอบการอ่านข้อมูล

3.4.3.4 การอ่านแอดเดรส (Address read cycle)



ภาพประกอบ 3-9 แสดงลักษณะของรอบการอ่านแอดเดรส

(ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 3-9 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตแอดเดรสที่ Base address+3
- สัญญาณ Address strobe จะถูกส่งออกมา (เป็นลอจิก '0') ถ้าสัญญาณ Wait เป็นลอจิก '0'
- Host จะรอการตอบรับโดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก '1'
- แอดเดรสจะถูกอ่านจากขาข้อมูล
- สัญญาณ Address strobe จะกลับเป็นลอจิก '1'
- สิ้นสุดรอบการอ่านแอดเดรส

สำหรับโหมดการทำงานแบบ EPP มาตรฐาน 1.7 สัญญาณ Data strobe และ Address strobe สามารถถูกส่งออกมาโดยไม่จำเป็นต้องรอให้สัญญาณ Wait เป็นลอจิก '0' แต่ทั้งโหมดการทำงานแบบ EPP มาตรฐาน 1.7 และโหมดการทำงานแบบ EPP มาตรฐาน 1.9 จำเป็นต้องรอจนกว่าสัญญาณ Wait เป็นลอจิก '1' เพื่อสิ้นสุดรอบการทำงาน

ในงานวิจัยนี้จะใช้พอร์ตขนานในโหมดการทำงานแบบ EPP ในการอินเตอร์เฟสกับเครื่องคอมพิวเตอร์เพื่อโอนย้ายข้อมูลดิจิทัลที่เก็บในหน่วยความจำแรม ดังนั้นจึงใช้เพียงรอบการทำงานในการอ่านข้อมูลเข้าพอร์ตขนานเท่านั้น

บทที่ 4

การออกแบบวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

ในบทนี้จะกล่าวถึงการออกแบบวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัลหรือ Image digitizer และตัวควบคุมที่อยู่ในรูปวงจรรวม FPGA (Field Programmable Gate Array) สำหรับแปลงสัญญาณภาพจากกล้องวิดีโอซึ่งเป็นสัญญาณภาพขาวดำให้เป็นภาพดิจิทัลขนาด 256×256 จุดต่อภาพเก็บไว้ในหน่วยความจำแรมและสามารถอ่านภาพดิจิทัลนี้โดยเครื่องคอมพิวเตอร์ผ่านทางพอร์ตขนานในโหมดการทำงานแบบ EPP (Enhanced Parallel Port) เพื่อนำภาพไปใช้ในงานต่างๆที่เกี่ยวข้อง โดย Image digitizer ที่ออกแบบและสร้างในงานวิจัยนี้จะมีหลักการทำงานและส่วนประกอบหลักเหมือนกับ Image digitizer ที่ใช้สำหรับสัญญาณภาพขาวดำโดยทั่วไปแต่มีส่วนที่เพิ่มเติมคือมีวงจรรวม FPGA สำหรับเป็นตัวควบคุมในกระบวนการเขียนภาพดิจิทัลลงหน่วยความจำแรมและกระบวนการอ่านภาพดิจิทัลจากหน่วยความจำแรมเข้าสู่เครื่องคอมพิวเตอร์ ซึ่งตัวควบคุมนี้จะช่วยทำให้การปรับเปลี่ยนวงจรสามารถทำได้ด้วยความคล่องตัวรวมทั้งการใช้พื้นที่หรือขนาดของแผงวงจรพิมพ์ (Printed Circuit Board) ที่น้อยลงเมื่อเปรียบเทียบกับ Image digitizer ที่จำหน่ายโดยทั่วไป ดังรายละเอียดที่จะได้กล่าวถึงต่อไป

4.1 การทำงานของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

Image digitizer ในงานวิจัยนี้จะถูกออกแบบให้สามารถเก็บภาพดิจิทัลจากสัญญาณภาพขาวดำที่มีความละเอียดขนาด 256×256 จุดต่อภาพและสามารถนำภาพดิจิทัลนี้มาแสดงบนเครื่องคอมพิวเตอร์ในรูปแบบไฟล์ข้อมูลภาพชนิด BMP (Bit Map File) โดยมีการทำงานทั้งหมด 2 โหมดคือ

4.1.1 โหมดการเขียนข้อมูลภาพ

โหมดการทำงานนี้เป็นขั้นตอนในการเขียนข้อมูลภาพลงหน่วยความจำแรม โดย Image digitizer จะเริ่ม digitize สัญญาณภาพเมื่อเข้าสู่เส้นกวาดของสัญญาณภาพรวมเส้นที่ 32 หรือสัญญาณ Hs ลุกที่ 32 เพื่อให้มั่นใจว่าเข้าสู่การกวาดภาพในแต่ละฟิลด์ของสัญญาณภาพรวมและจะเสร็จสิ้นการทำงานในโหมดนี้เมื่อครบจำนวนเส้นกวาดทั้งหมด 256 เส้นตามที่ต้องการ

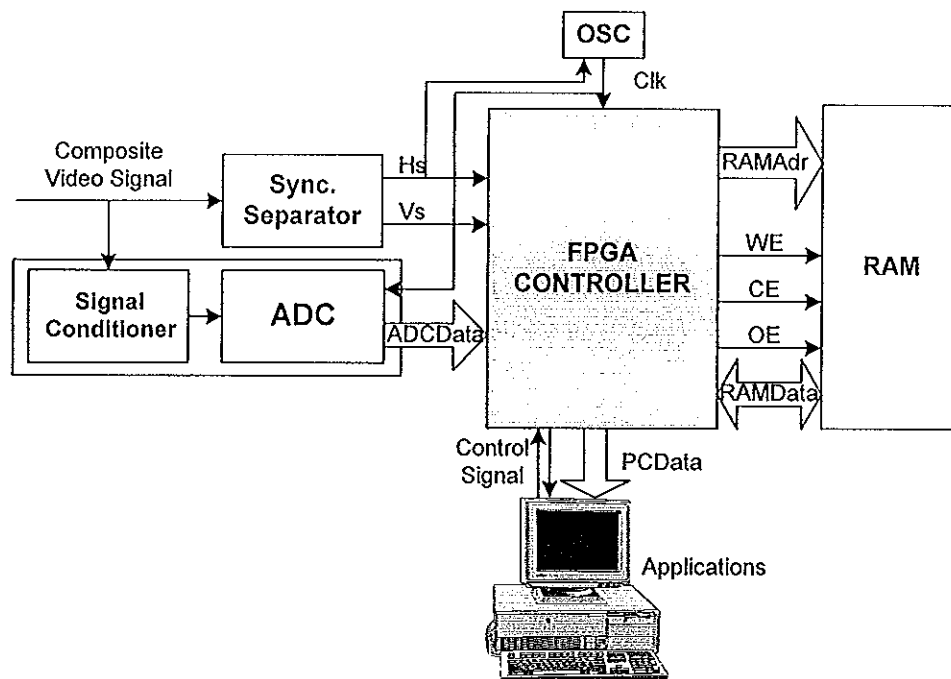
4.1.2 โหมดการอ่านข้อมูลภาพ

โหมดการทำงานนี้เป็นขั้นตอนในการอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่เครื่อง

คอมพิวเตอร์ผ่านทางพอร์ตขนานในโหมดการทำงานแบบ EPP โดยข้อมูลภาพจะถูกเก็บเป็นไฟล์ข้อมูลชนิดไบนารี (Binary file) แล้วจึงนำมาแปลงเป็นไฟล์ภาพชนิด BMP เพื่อประโยชน์ในการใช้งานภาพต่อไป สาเหตุที่เลือกบันทึกข้อมูลภาพเป็นชนิด BMP นี้เนื่องจากเป็นรูปแบบที่นิยมใช้กันอย่างแพร่หลายโดยเฉพาะในการใช้งานบน Windows และในปัจจุบันมีโปรแกรมที่สามารถแปลงไฟล์ข้อมูลชนิดไบนารีเป็นไฟล์ภาพชนิด BMP ได้อย่างสะดวกและรวดเร็วโดยโปรแกรมที่ใช้ในงานนี้มีชื่อว่า bin2bmp.exe

4.2 ส่วนประกอบของวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

Image digitizer ที่ออกแบบจะประกอบด้วยส่วนหลักทั้งหมด 5 ส่วนดังแสดงโดยอะแกรมโดยรวมในภาพประกอบ 4-1 ซึ่งสามารถอธิบายถึงหน้าที่และการทำงานในแต่ละส่วนได้ดังนี้

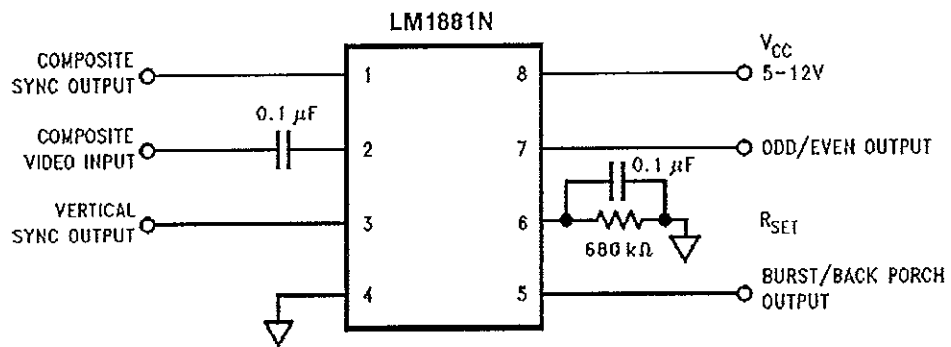


ภาพประกอบ 4-1 แสดงโดยอะแกรมโดยรวมของ Image digitizer

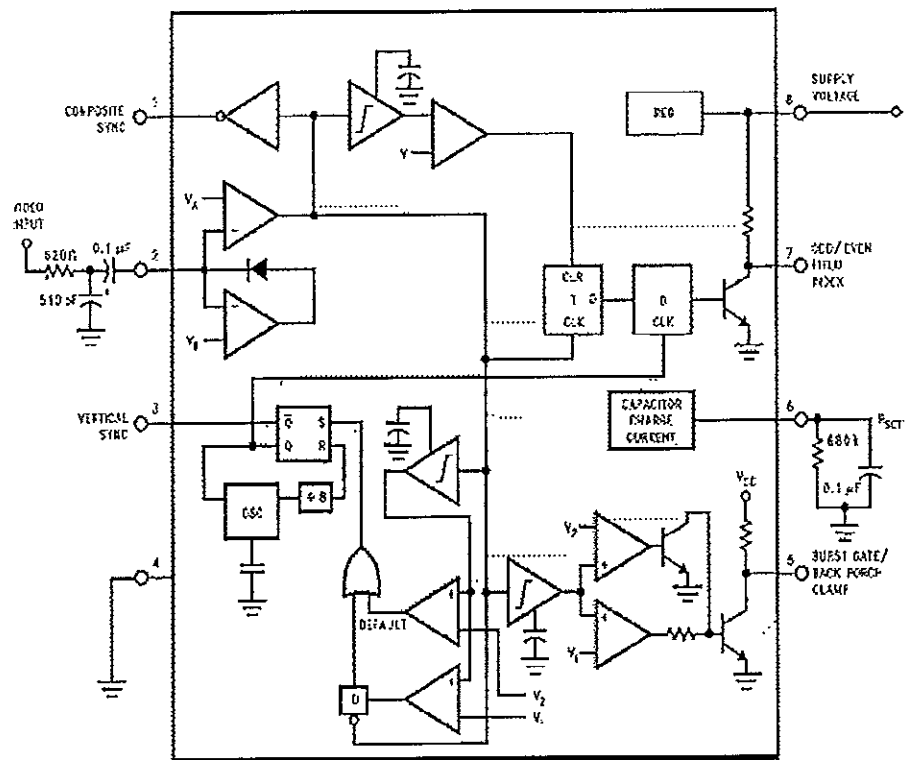
4.2.1 วงจรแยกสัญญาณซิงก์ (Sync. Separator)

ทำหน้าที่แยกสัญญาณซิงก์ทางแนวนอน (Horizontal sync. signal, Hs) และสัญญาณซิงก์ทางแนวตั้ง (Vertical sync. signal, Vs) ออกจากสัญญาณภาพรวมโดยใช้สัญญาณซิงก์ทั้งสองสัญญาณนี้ในการสร้างแอดเดรสทางแถวและคอลัมน์ให้หน่วยความจำแรมเนื่องจากการเก็บ

ข้อมูลดิจิทัลของหน่วยความจำแรมจำเป็นต้องให้หน่วยความจำแรมสามารถเก็บภาพที่ถูกต้องตามลำดับการกวาดเส้นสัญญาณภาพ ในงานวิจัยนี้เลือกใช้วงจรรวมเบอร์ LM1881 ของบริษัท National Semiconductor เป็นวงจรแยกสัญญาณซิงก์ซึ่งเป็นที่ยอมรับใช้กันแพร่หลายและมีราคาถูก ดังแสดงวงจรในภาพประกอบ 4-2 และไดอะแกรมภายในของวงจรรวมในภาพประกอบ 4-3 ซึ่งสัญญาณซิงก์ที่ได้จะมี 2 สัญญาณคือสัญญาณ Vs และสัญญาณ Composite sync. ซึ่งจะประกอบด้วยสัญญาณ Vs สัญญาณอ็อกวไลซิงและสัญญาณ Hs ตามที่ต้องการ



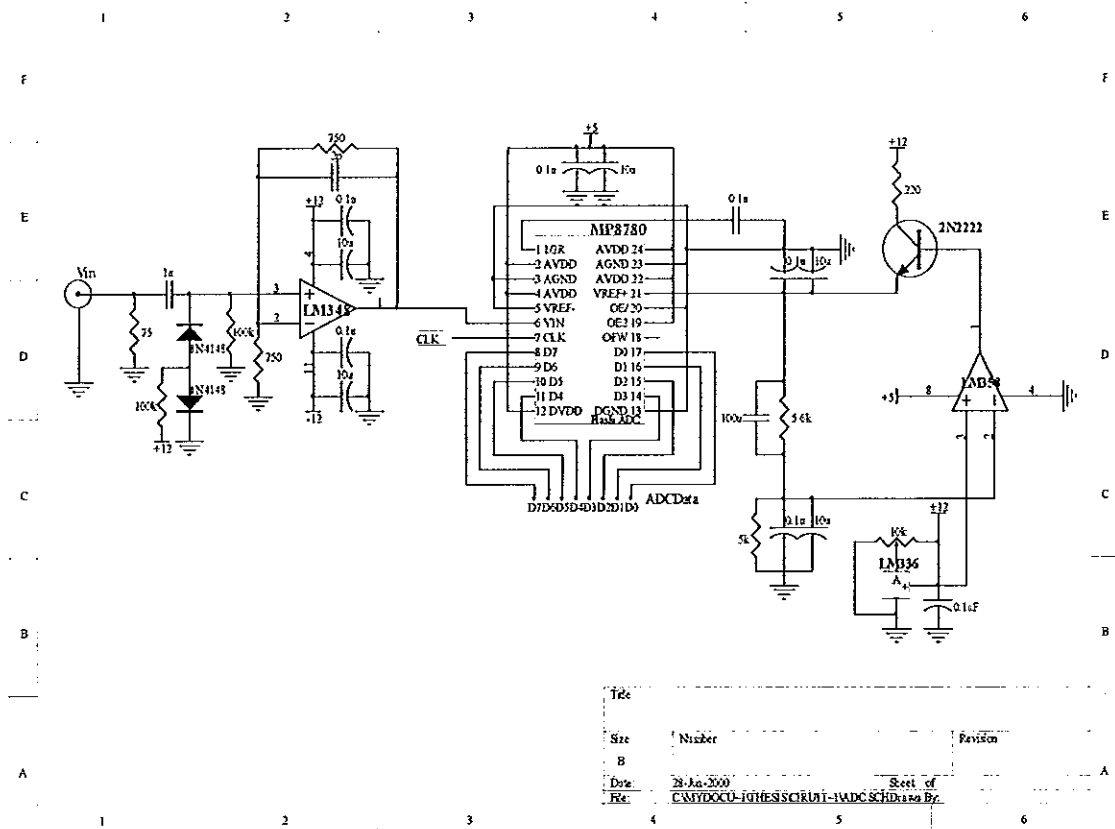
ภาพประกอบ 4-2 แสดงวงจรรวมแยกสัญญาณซิงก์เบอร์ LM1881



ภาพประกอบ 4-3 แสดงไดอะแกรมภายในของวงจรรวมแยกสัญญาณซิงก์เบอร์ LM1881

4.2.2 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (Analog-to-Digital Converter, ADC)

ทำหน้าที่แปลงสัญญาณภาพรวมซึ่งเป็นสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล เพื่อให้อยู่ในรูปของข้อมูลดิจิทัลที่สามารถเก็บไว้ในหน่วยความจำแรมได้ เนื่องจากสัญญาณภาพเป็นข้อมูลที่มีความถี่สูงคือมีความถี่เท่ากับ 15.625 กิโลเฮิรตซ์จึงจำเป็นต้องใช้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบแฟลช (Flash ADC) เพื่อให้เวลาที่ใช้ในการแปลงสัญญาณ (Conversion time) เร็วที่สุด และต้องการเก็บภาพดิจิทัลให้มีระดับเทา 256 ระดับซึ่งเป็นความละเอียดที่เพียงพอสำหรับภาพขาวดำ ดังนั้นจึงเลือกใช้วงจรรวม Flash ADC ของบริษัท EXAR เบอร์ MP8780 ขนาด 8 บิตและมีความถี่ในการทำงานสูงสุด 20 เมกกะเฮิรตซ์ โดยสัญญาณภาพรวมที่เป็นสัญญาณอินพุตให้วงจร Flash ADC ต้องนำมาผ่านส่วนวงจรปรับสัญญาณ (Signal conditioner) ซึ่งมีอัตราขยาย (Gain) เท่ากับ 2 เท่าโดยเลือกใช้วงจรเปรียบเทียบเบอร์ LM348 เพื่อปรับสัญญาณภาพรวมให้มีขนาดแรงดันที่เหมาะสมก่อนดังแสดงวงจรในภาพประกอบ 4-4

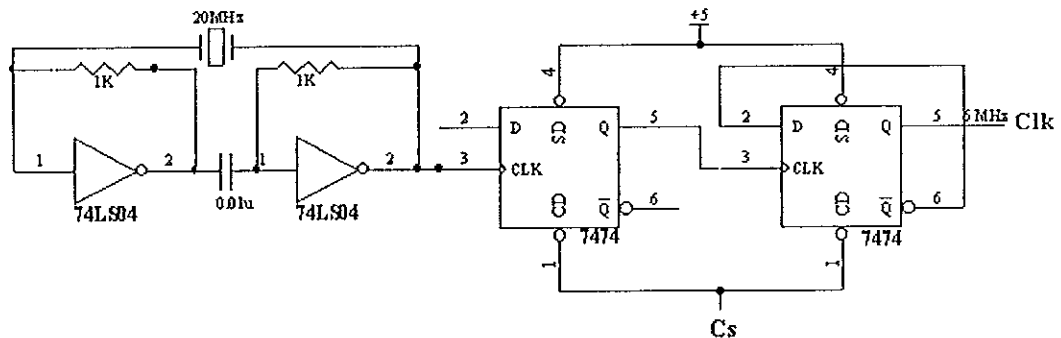


ภาพประกอบ 4-4 แสดงวงจร Flash ADC และส่วน Signal conditioner

สัญญาณภาพรวมที่ผ่านส่วนวงจรปรับสัญญาณแล้วจะเข้ามาทางขาสัญญาณ Vin ส่วนสัญญาณนาฬิกาความถี่ 5 เมกกะเฮิร์ตซ์ถูกป้อนเข้าที่ขาสัญญาณ CLK เพื่อกำหนดอัตราการแปลงสัญญาณ โดยวงจรจะทำงานเมื่อสัญญาณควบคุมที่ขาสัญญาณ OE1 เป็นลอจิก '0' และขาสัญญาณ OE2 เป็นลอจิก '1' ส่วนขาสัญญาณ Vref+ ซึ่งเป็นแรงดันอ้างอิงมีค่าประมาณ 2.964 โวลต์และขาสัญญาณ Vref- ให้มีค่าเป็นศูนย์โวลต์

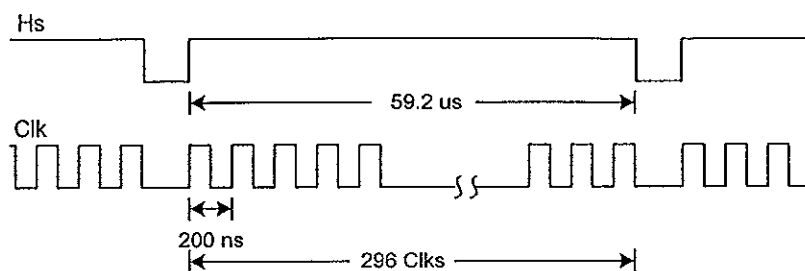
4.2.3 วงจรกำเนิดสัญญาณนาฬิกา (Oscillator)

ทำหน้าที่กำเนิดสัญญาณนาฬิกา (Clock, Clk) เพื่อกำหนดอัตราในการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลให้กับวงจร Flash ADC รวมทั้งควบคุมกระบวนการต่างๆที่เกิดขึ้นภายใน Image digitizer ทั้งในโหมดการเขียนและอ่านข้อมูลภาพ ทั้งนี้เนื่องจากส่วนที่เป็นข้อมูลภาพ (Picture information) ในสัญญาณ Hs แต่ละลูกจะมีช่วงเวลาเท่ากับ 51.2 ไมโครวินาทีและผู้ใช้จำเป็นต้องการเก็บจำนวนข้อมูลภาพในแต่ละเส้นภาพเท่ากับ 256 จุดภาพ ดังนั้นสัญญาณ Clk ที่ใช้เป็นความถี่ในการสุ่มสัญญาณวิดีโอ (Sampling frequency) จึงต้องมีความถี่เท่ากับ 5 เมกกะเฮิร์ตซ์ (256 จุดภาพ/51.2 ไมโครวินาที) หรือมีคาบเวลาเท่ากับ 200 นาโนวินาที ซึ่งจะสร้างวงจรส่วนนี้โดยอาศัยคริสตอลความถี่ 20 เมกกะเฮิร์ตซ์จากนั้นนำมาผ่านวงจรหารสี่โดยใช้วงจรรวมเบอร์ 74F74 และเพื่อทำให้เริ่มการสุ่มในส่วนที่เป็นข้อมูลภาพของสัญญาณ Hs แต่ละสัญญาณจึงใช้สัญญาณ Hs ในการควบคุมวงจรหารสี่ด้วยดังวงจรในภาพประกอบ 4-5



ภาพประกอบ 4-5 แสดงวงจรกำเนิดสัญญาณนาฬิกาความถี่ 5 เมกกะเฮิร์ตซ์

เนื่องจากช่วงเวลาระหว่างสัญญาณ Hs 2 ลูกจะมีค่าเท่ากับ 59.2 ไมโครวินาทีดังนั้นจึงเกิดสัญญาณ Clk ทั้งหมด 296 ลูก (59.2 ไมโครวินาที/200 นาโนวินาที) ดังภาพประกอบ 4-6



ภาพประกอบ 4-6 แสดงจำนวนสัญญาณนาฬิกาที่เกิดขึ้นระหว่างสัญญาณ Hs

4.2.4 ส่วนเก็บข้อมูลภาพหรือหน่วยความจำแรม (RAM)

ทำหน้าที่เก็บสัญญาณดิจิทัลที่ได้จากวงจร ADC เพื่อแทนตำแหน่งขององค์ประกอบภาพในแต่ละตำแหน่งได้และทำให้สามารถโอนย้ายข้อมูลเข้าสู่เครื่องคอมพิวเตอร์ได้ สำหรับการเลือกใช้หน่วยความจำต้องพิจารณาถึงขนาดของหน่วยความจำที่มีความสัมพันธ์กับความละเอียดของภาพตามที่ได้กล่าวไปแล้วในบทที่แล้ว สำหรับงานวิจัยนี้ต้องการเก็บภาพให้มีความละเอียดของภาพ 256×256 จุดต่อภาพดังนั้นหน่วยความจำที่ใช้จึงมีขนาด 64 กิโลไบต์ นอกจากขนาดหน่วยความจำแล้วยังต้องคำนึงค่าเวลาเข้าถึง (Access time) อีกด้วยซึ่งต้องเพียงพอสำหรับการเก็บภาพในแต่ละฟิลด์ด้วยเช่นกัน โดยหน่วยความจำที่ใช้อาจเป็นชนิด Static RAM (SRAM) หรือชนิด Dynamic RAM (DRAM) ก็ได้ ซึ่งชนิด SRAM จะมีความเร็วมากกว่าแต่มีราคาสูงกว่าชนิด DRAM

ในงานวิจัยนี้เลือกใช้หน่วยความจำแรมแบบ SRAM เบอร์ 62256 ขนาด 32 กิโลไบต์มีเวลาเข้าถึงเท่ากับ 70 นาโนวินาทีจำนวน 2 ตัวซึ่งมีความเร็วเพียงพอสำหรับการเขียนอ่านข้อมูลภาพและมีราคาไม่สูงนัก สำหรับกระบวนการเก็บข้อมูลภาพที่สอดคล้องกับตำแหน่งของภาพนั้นๆ จะถูกควบคุมโดยตัวควบคุมที่จะกล่าวถึงต่อไป

4.2.5 ตัวควบคุมในรูปวงจรรวม FPGA (FPGA Controller)

ทำหน้าที่ควบคุมกระบวนการทั้งในการเขียนข้อมูลภาพลงหน่วยความจำแรมและอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่พอร์ตขนานของเครื่องคอมพิวเตอร์ในโหมดการทำงานแบบ EPP ซึ่งจะได้กล่าวถึงการทำงานและการออกแบบตัวควบคุมนี้อย่างละเอียดในหัวข้อต่อไป

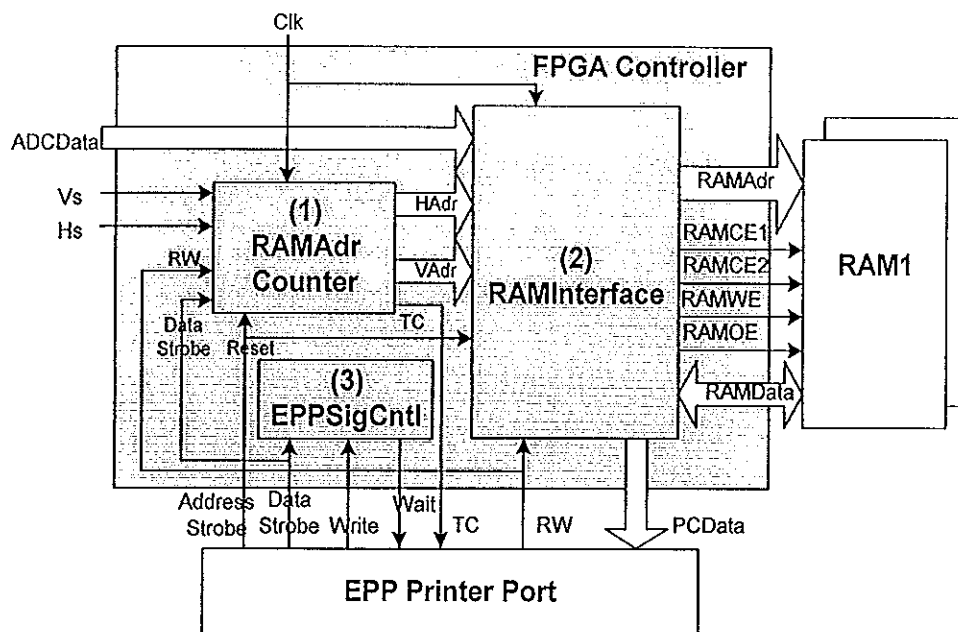
4.3 การออกแบบตัวควบคุม

ส่วนประกอบที่มีความสำคัญที่สุดในการควบคุมการทำงานของ Image digitizer นี้คือตัวควบคุม (Controller) ซึ่งจะทำหน้าที่ควบคุมทั้งกระบวนการในโหมดการเขียนข้อมูลภาพและ

กระบวนการในโหมดการอ่านข้อมูลภาพ โดยจะถูกออกแบบให้ทำงานร่วมกับส่วนวงจรหลักต่างๆ ดังที่ได้กล่าวมาแล้วคือเมื่ออยู่ในโหมดการเขียนข้อมูลภาพตัวควบคุมจะใช้สัญญาณ Hs และ Vs จากวงจรแยกสัญญาณซิงก์เพื่อสร้างแอดเดรสให้หน่วยความจำแรม และเมื่ออยู่ในโหมดการอ่านข้อมูลภาพตัวควบคุมจะใช้สัญญาณ Data strobe จากพอร์ตขนานในโหมดการทำงานแบบ EPP ในการสร้างแอดเดรสแทน กล่าวโดยสรุปคือการสร้างแอดเดรสให้หน่วยความจำแรม การติดต่อกับหน่วยความจำแรมและการติดต่อกับพอร์ตขนานเพื่อนำภาพดิจิทัลออกไปใช้งานจะเป็นหน้าที่ของตัวควบคุมนั่นเอง

4.3.1 โครงสร้างภายในของตัวควบคุม

โครงสร้างภายในของตัวควบคุมที่ได้ออกแบบดังแสดงในภาพประกอบ 4-7 จะประกอบด้วยส่วนหลักทั้งหมด 3 ส่วนคือส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม ส่วนการอินเตอร์เฟสกับหน่วยความจำแรมและส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP สำหรับหน้าที่และการทำงานของแต่ละส่วนหลักภายในตัวควบคุมมีดังนี้

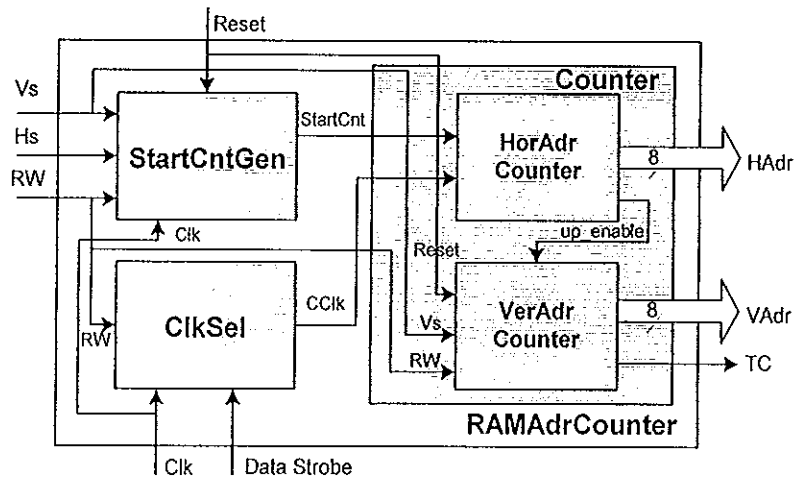


ภาพประกอบ 4-7 แสดงโครงสร้างภายในของตัวควบคุม

4.3.1.1 ส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม (RAM Address Counter, RAMAdrCounter)

ทำหน้าที่สร้างแอดเดรสจำนวน 16 บิตให้หน่วยความจำแรมโดยเมื่ออยู่ในโหมดการเขียนข้อมูลภาพตัวควบคุมจะใช้สัญญาณ Hs และ Vs จากวงจรแยกสัญญาณซิงก์และสัญญาณ

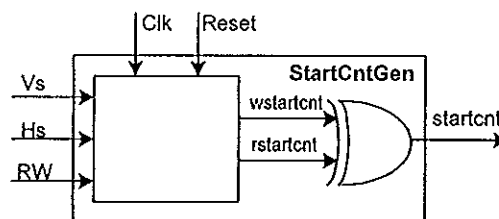
Clk จากวงจรกำเนิดสัญญาณนาฬิกาในการสร้างแอดเดรส แต่เมื่ออยู่ในโหมดการอ่านข้อมูลภาพ ตัวควบคุมจะใช้สัญญาณ Data strobe จากพอร์ตขนานในการสร้างแอดเดรส ซึ่งส่วนนี้จะประกอบด้วยส่วนย่อย 3 ส่วนดังภาพประกอบ 4-8 คือ



ภาพประกอบ 4-8 แสดงส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม

ก. ส่วนสร้างสัญญาณให้วงจรนับเริ่มการทำงาน (Start Count Generator, StartCntGen)

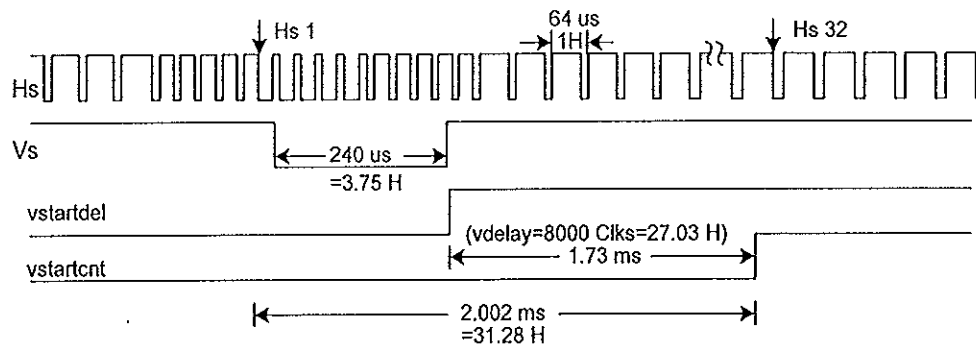
ทำหน้าที่สร้างสัญญาณ startcnt (start count) เพื่อให้ส่วนวงจรถับ (Counter) เริ่มทำงานทั้งในโหมดการเขียนและอ่านข้อมูลภาพดังภาพประกอบ 4-9 โดยมีสัญญาณ RW (Read/Write) ซึ่งเป็นสัญญาณที่ส่งมาจากพอร์ตขนานในโหมดการทำงานแบบ EPP ของเครื่องคอมพิวเตอร์เป็นตัวเลือกกล่าวคือ



ภาพประกอบ 4-9 แสดงส่วนสร้างสัญญาณให้วงจรถับเริ่มการทำงาน

- เมื่อสัญญาณ RW เป็นลอจิก '1' แสดงว่า Image digitizer ทำงานอยู่ในโหมดการเขียนข้อมูลภาพ สัญญาณ wstartcnt (write start count) จะมีค่าเป็นลอจิก '1' ก็ต่อเมื่อมี

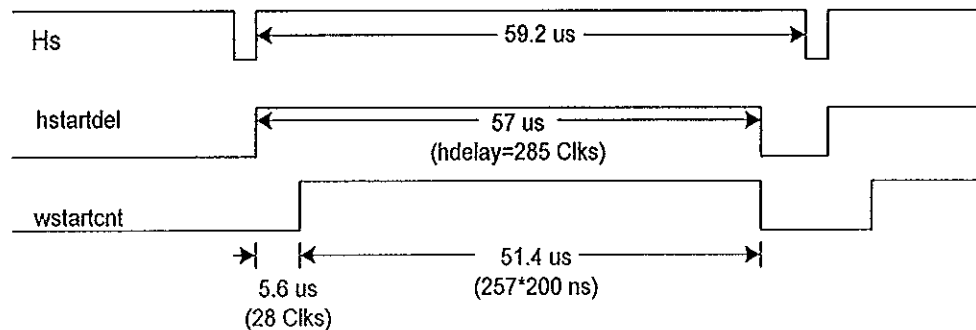
สัญญาณ Vs ในแต่ละฟิลด์ของสัญญาณภาพรวมเข้ามาและเข้าสู่ส่วนที่เป็นข้อมูลภาพของสัญญาณ Hs ลูกที่ 32 กล่าวคือเมื่อมีสัญญาณ Vs เข้ามาจะทำให้เกิดสัญญาณ vstartdel (vertical start delay) ขึ้นเพื่อทำให้เกิดการหน่วงเวลาไป 1.73 มิลลิวินาที (จากภาคผนวก ข โปรแกรมส่วน StartCntGen ตัวแปร vdelay (vertical delay) จะเพิ่มค่าตามจำนวนของสัญญาณ Clk จนครบ 8000 ลูกและเนื่องจากในคาบเวลาของสัญญาณ Hs 1 ลูก (1 H = 64 ไมโครวินาที) จะมีสัญญาณ Clk ทั้งหมด 296 ลูกดังนั้นเวลาที่หน่วงไปจึงเท่ากับ Clk 8000 ลูก/Clk 296 ลูก = 27.03 H = 1.73 มิลลิวินาที) เมื่อหน่วงเวลาครบจะเกิดสัญญาณ vstartcnt (vertical start count) ขึ้นเพื่อรอสัญญาณ Hs เส้นต่อไปดังแสดงไต่อะแกรมทางเวลา (Timing diagram) ในภาพประกอบ 4-10



ภาพประกอบ 4-10 แสดงไต่อะแกรมทางเวลาของสัญญาณ vstartdel และ vstartcnt

เมื่อมีสัญญาณ Hs ลูกที่ 32 เข้ามาก็จะเกิดสัญญาณ hstartdel (horizontal start delay) เพื่อทำให้เกิดการหน่วงเวลาต่อไปอีก 5.8 ไมโครวินาทีจึงจะเริ่มการส่มส่วนที่เป็นข้อมูลภาพจริงๆเนื่องจากช่วงเวลานี้จะเป็นส่วนที่เรียกว่า Back porch ของสัญญาณภาพดังรายละเอียดของสัญญาณภาพรวมที่ได้กล่าวไว้ก่อนหน้าแล้ว (จากภาคผนวก ข โปรแกรมส่วน StartCntGen ตัวแปร hdelay (horizontal delay) จะเพิ่มค่าตามจำนวนสัญญาณ Clk จนครบ 28 ลูก (28×200 นาโนวินาที = 5.6 ไมโครวินาที) ทำให้วงจรมับเริ่มนับที่สัญญาณ Clk ลูกที่ 29) เมื่อหน่วงเวลาครบก็จะทำให้สัญญาณ wstartcnt (write start count) มีค่าเป็นลอจิก '1' ส่วนสัญญาณ rstartcnt (read start count) จะมีค่าเป็นลอจิก '0' เมื่อทั้งสองสัญญาณผ่านเกตเอ็กซ์คลูซีฟ (Exclusive OR gate) ก็จะได้สัญญาณ startcnt มีค่าเป็นลอจิก '1' ส่งให้ส่วนวงจรมับเริ่มการส่มส่วนที่เป็นข้อมูลภาพหรือสร้างแอดเดรสตามขอบขาขึ้น (Rising edge) ของสัญญาณ Clk เป็นจำนวน 256 จุดภาพ เมื่อส่มครบจำนวนแล้วสัญญาณ wstartcnt จะเป็นลอจิก '0' ทำให้สัญญาณ startcnt

เป็นลอจิก '0' เพื่อเริ่มการสุ่มส่วนข้อมูลภาพของสัญญาณ Hs ลุกต่อไปจนกว่าจะครบจำนวน 256 เส้นภาพดังแสดงโดยอะแกรมทางเวลาในภาพประกอบ 4-11



ภาพประกอบ 4-11 แสดงโดยอะแกรมทางเวลาของสัญญาณ hstartdel และ wstartnt

- เมื่อสัญญาณ RW เป็นลอจิก '0' แสดงว่า Image digitizer ทำงานอยู่ในโหมดการอ่านข้อมูลภาพ สัญญาณ wstartnt จะมีค่าเป็นลอจิก '0' ส่วนสัญญาณ rstartnt จะมีค่าเป็นลอจิก '1' เมื่อทั้งสองสัญญาณผ่านเกตเอ็กซ์คลูซีฟออร์ก์จะได้สัญญาณ startcnt ที่มีค่าเป็นลอจิก '1' ส่งให้ส่วนวงจรนับต่อไป

ข. ส่วนเลือกสัญญาณนาฬิกาให้ส่วนวงจรนับ (Clock Selector, ClkSel)

ทำหน้าที่เลือกสัญญาณ Clk ให้ส่วนวงจรนับ (Counter Clock, CClk) โดยมีสัญญาณ RW เป็นตัวเลือกเช่นเดียวกัน กล่าวคือ

- เมื่อสัญญาณ RW เป็นลอจิก '1' ส่วนนี้จะใช้สัญญาณ Clk จากวงจรกำเนิดสัญญาณนาฬิกาเป็นสัญญาณ CClk (CClk=Clk)

- เมื่อสัญญาณ RW เป็นลอจิก '0' ส่วนนี้จะใช้สัญญาณ Data strobe จากพอร์ตนานเป็นสัญญาณ CClk โดยให้สัญญาณ CClk มีค่าเท่ากับสัญญาณ Data strobe ที่อินเวอร์ทแล้ว (CClk=not Data strobe) เนื่องจากวงจรนับจะนับตามขอบขาขึ้นของสัญญาณ CClk

ค. ส่วนวงจรนับ (Counter)

ทำหน้าที่สร้างแอดเดรสจำนวน 16 บิตให้หน่วยความจำแรมเมื่อสัญญาณ startcnt มีค่าเป็นลอจิก '1' โดยจะประกอบด้วย 2 ส่วน คือ

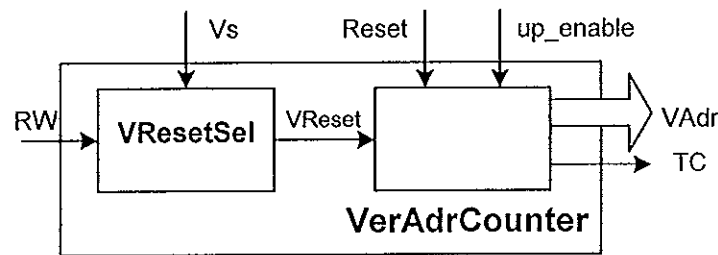
(1) วงจรนับเพื่อกำเนิดแอดเดรสทางแนวนอน (Horizontal Address Counter, HorAdrCounter)

ทำหน้าที่สร้างแอดเดรสทางแนวนอน (Horizontal Address, HAdr) จำนวน 8 บิตเมื่อ

สัญญาณ startcnt มีค่าเป็นลอจิก '1' ซึ่งเมื่อสร้างแอดเดรสจนครบหรือสุ่มสัญญาณในส่วนที่เป็นข้อมูลภาพตามสัญญาณ CClk ที่ได้จากส่วน CkSel จนครบ 256 จุดภาพต่อ 1 เส้นภาพแล้วก็จะรีเซ็ตการนับเพื่อเริ่มการนับครั้งใหม่และจะส่งสัญญาณ up_enable ให้กับส่วนวงจรนับเพื่อกำเนิดแอดเดรสทางแนวตั้งต่อไป

(2) วงจรนับเพื่อกำเนิดแอดเดรสทางแนวตั้ง (Vertical Address Counter, VerAdrCounter)

ทำหน้าที่สร้างแอดเดรสทางแนวตั้ง (Vertical Address, VAdr) จำนวน 8 บิต โดยจะเพิ่มค่าทีละหนึ่งตามสัญญาณ up_enable เมื่อสร้างแอดเดรสจนครบหรือนับจำนวนสัญญาณ Hs ครบ 256 เส้นแล้วก็จะส่งสัญญาณ TC (Terminal Count) ซึ่งมีค่าลอจิก '1' ออกมาเพื่อแสดงสถานะว่าเสร็จสิ้นการสร้างแอดเดรสให้หน่วยความจำแรมในหนึ่งฟิลด์ของสัญญาณภาพแล้วและจะรีเซ็ตค่าเพื่อเริ่มการนับสัญญาณ Hs ในฟิลด์ต่อไป โดยภายในส่วนของ VerAdrCounter นี้จะมีส่วนเลือกสัญญาณรีเซ็ต (VResetSelector, VResetSel) เพิ่มเติมขึ้นมาดังภาพประกอบ 4-12 ซึ่งทำหน้าที่สร้างสัญญาณ VReset เพื่อรีเซ็ตการนับ โดยเมื่อ



ภาพประกอบ 4-12 แสดงส่วนสร้างสัญญาณรีเซ็ตเพื่อใช้ในส่วน VerAdrCounter

- สัญญาณ RW เป็นลอจิก '1' สัญญาณ VReset จะมีค่าเท่ากับสัญญาณ Vs เพื่อให้เกิดการรีเซ็ตการนับเมื่อมีสัญญาณ Vs ในแต่ละฟิลด์เข้ามา
- สัญญาณ RW เป็นลอจิก '0' สัญญาณ VReset จะมีค่าเป็นลอจิก '1' เพื่อให้เกิดการรีเซ็ตการนับตามสัญญาณ Reset จากภายนอกเท่านั้น

4.3.1.2 ส่วนการอินเตอร์เฟสกับหน่วยความจำแรม (RAM Interface) ทำหน้าที่

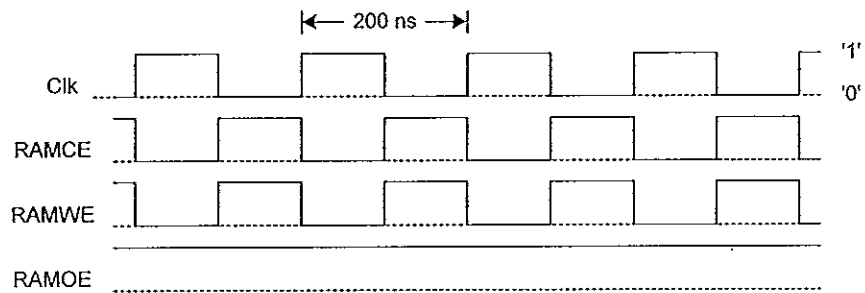
ก. ส่งแอดเดรส (RAMAddress, RAMAdr) จำนวน 16 บิตที่ได้จากส่วน RAMAdrCounter ให้หน่วยความจำแรมทั้งในโหมดการเขียนและอ่านข้อมูลภาพ

ข. ส่งข้อมูลดิจิทัลที่ได้จากวงจร ADC (ADCData) จำนวน 8 บิตให้หน่วยความจำแรมเมื่ออยู่ในโหมดการเขียนข้อมูลภาพ

ค. รับข้อมูลดิจิทัลจากหน่วยความจำแรม (RAMData) จำนวน 8 บิตส่งให้เครื่องคอมพิวเตอร์ในโหมดการอ่านข้อมูลภาพ

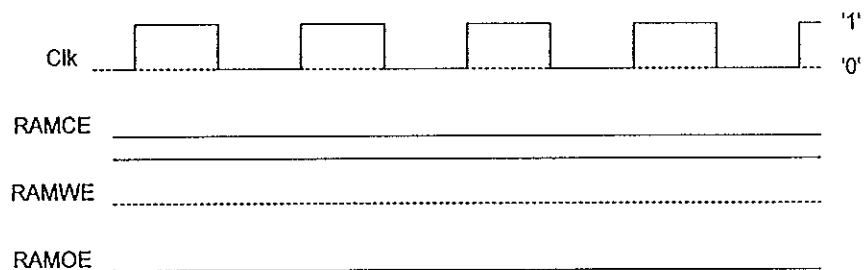
ง. สร้างสัญญาณควบคุมต่างๆให้หน่วยความจำแรม เช่น สัญญาณ RAMCE (RAM Chip Enable), RAMWE (RAM Write Enable) และ RAMOE (RAM Output Enable) โดยเมื่ออยู่ใน

(1) โหมดการเขียนข้อมูลภาพ สัญญาณควบคุมจะมีลักษณะดังภาพประกอบ 4-13



ภาพประกอบ 4-13 แสดงสัญญาณควบคุมเมื่ออยู่ในโหมดการเขียนข้อมูลภาพ

(2) โหมดการอ่านข้อมูลภาพ สัญญาณควบคุมจะมีลักษณะดังภาพประกอบ 4-14

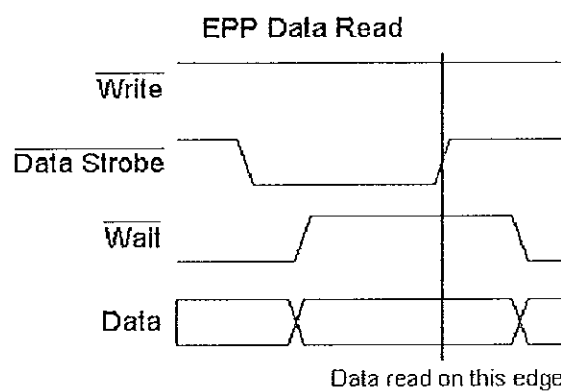


ภาพประกอบ 4-14 แสดงสัญญาณควบคุมเมื่ออยู่ในโหมดการอ่านข้อมูลภาพ

เนื่องจากในงานวิจัยนี้ใช้หน่วยความจำแรมขนาด 32 กิโลไบต์จำนวน 2 ตัวจึงต้องสร้างสัญญาณ CE จำนวน 2 สัญญาณซึ่งก็คือ RAMCE1 และ RAMCE2 เพื่อให้หน่วยความจำแรมทั้งสองตัวสลับกันทำงานโดยสัญญาณ RAMCE1 จะแอกทีฟเมื่อบิตนัยสำคัญสูงสุด (Most Significant Bit, MSB) ของ RAMAdr (RAMAdr15) มีค่าเป็นลอจิก '0' เพื่อทำให้หน่วยความจำแรมตัวที่ 1 ทำงานและสัญญาณ RAMCE2 จะเป็นแอกทีฟเมื่อบิต MSB ของ RAMAdr มีค่าเป็นลอจิก '1' เพื่อทำให้หน่วยความจำแรมตัวที่ 2 ทำงาน ส่วนสัญญาณอื่นสามารถใช้ร่วมกันได้ตามปกติ

4.3.1.3 ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP (EPP Signal Control, EPPSigCntl)

เนื่องจากในการใช้สัญญาณ Data strobe จากพอร์ตขนานเป็นสัญญาณ CClk เพื่อสร้างแอดเดรสให้หน่วยความจำแรมในโหมดการอ่านข้อมูลภาพนั้นจำเป็นต้องมีสัญญาณ Wait จากภายนอกเป็นสัญญาณตอบรับเพื่อให้เกิดการ Handshaking ตามข้อตกลงของพอร์ตขนานในโหมดการทำงานแบบ EPP ซึ่งรายละเอียดเกี่ยวกับการทำงานของพอร์ตขนานในโหมดการทำงานแบบ EPP ได้กล่าวไว้ก่อนหน้านี้แล้ว ดังนั้นส่วนนี้จึงทำหน้าที่สร้างสัญญาณ Wait ตอบรับเมื่อมีสัญญาณ Write และ Data strobe จากพอร์ตขนานดังภาพประกอบ 4-15

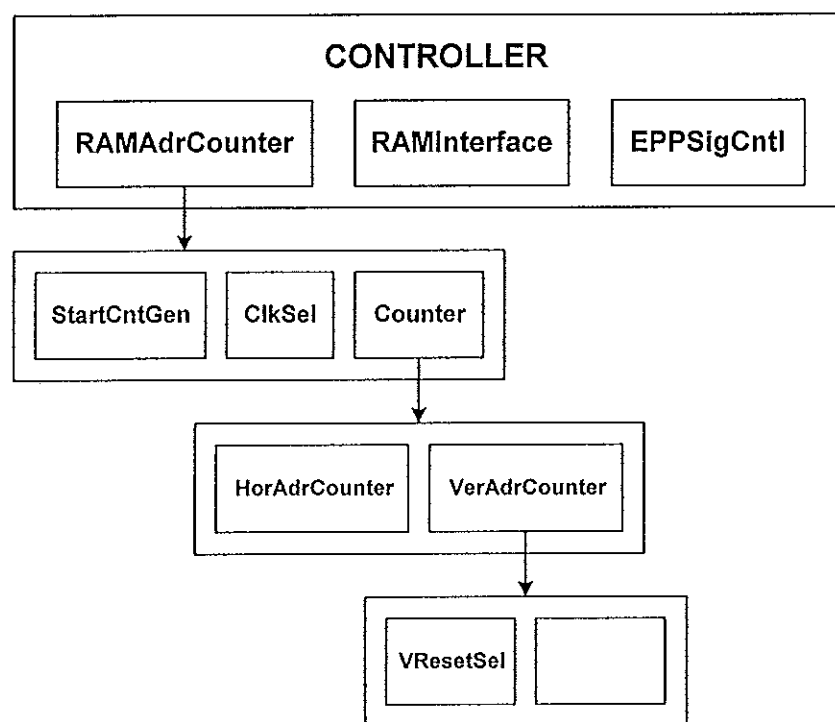


ภาพประกอบ 4-15 แสดงสัญญาณ Wait เมื่อมีสัญญาณ Write และ Data strobe (ที่มา : Craig Peacock, Interfacing the Enhanced Parallel Port version 1.0)

4.3.2 การออกแบบตัวควบคุมโดยใช้วงจรรวม FPGA

ในงานวิจัยนี้ตัวควบคุมจะอยู่ในรูปวงจรรวมประเภท FPGA โดยเลือกใช้วงจรรวม FPGA เบอร์ XC4003EPC84 ในตระกูล XC4000E ของบริษัท Xilinx (รายละเอียดเกี่ยวกับสถาปัตยกรรมของวงจรรวม FPGA สามารถดูได้ในภาคผนวก จ) ซึ่งเป็นที่นิยมแพร่หลายและสามารถยืมชีพเพื่อใช้ในการทดสอบได้จากส่วนปฏิบัติการวิจัยและพัฒนาเทคโนโลยีไมโครอิเล็กทรอนิกส์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติหรือเนคเทค สำหรับการเขียนโปรแกรมเพื่อออกแบบวงจรรวมจะใช้ภาษา VHDL (VHSIC Hardware Description Language) และหลักการออกแบบจากสูงสูดต่ำ (Top-down design) ซึ่งรายละเอียดเกี่ยวกับขั้นตอนการพัฒนาออกแบบสำหรับวงจรรวม FPGA รวมทั้งรหัสต้นฉบับ (Source code) ของโปรแกรมได้แสดงในภาคผนวก ก และ ข ตามลำดับ โดยสามารถแสดงโมดูลในแต่ละลำดับชั้น (Hierarchy) ของการออกแบบตัวควบคุมซึ่งได้กล่าวถึงรายละเอียดของแต่ละโมดูลไปแล้วในหัวข้อ

4.3.1 ในภาพประกอบ 4-16



ภาพประกอบ 4-16 แสดงโมดูลในแต่ละลำดับชั้นของการออกแบบตัวควบคุม
โดยใช้วงจรรวม FPGA

จากการออกแบบตัวควบคุมโดยใช้วงจรรวม FPGA ร่วมกับส่วนของวงจรหลักต่างๆตามที่ได้กล่าวมาทั้งหมดนั้น สัญญาณบางสัญญาณที่ใช้ในตัวควบคุมจะเป็นสัญญาณที่ได้มาจากพอร์ตขนานในโหมดการทำงานแบบ EPP เนื่องจากสัญญาณเหล่านี้ผู้ใช้สามารถใช้งานได้เป็นอย่างดีและเป็นการใช้ประโยชน์จากพอร์ตขนานให้มากที่สุด ดังนั้นจึงขอสรุปชื่อสัญญาณทั้งหมดที่ใช้ใน Image digitizer กับสัญญาณที่ใช้กับตัวควบคุมดังตาราง 4-1 ดังนี้

ตาราง 4-1 แสดงการสรุปสัญญาณทั้งหมดที่ใช้ใน Image digitizer กับสัญญาณของตัวควบคุม

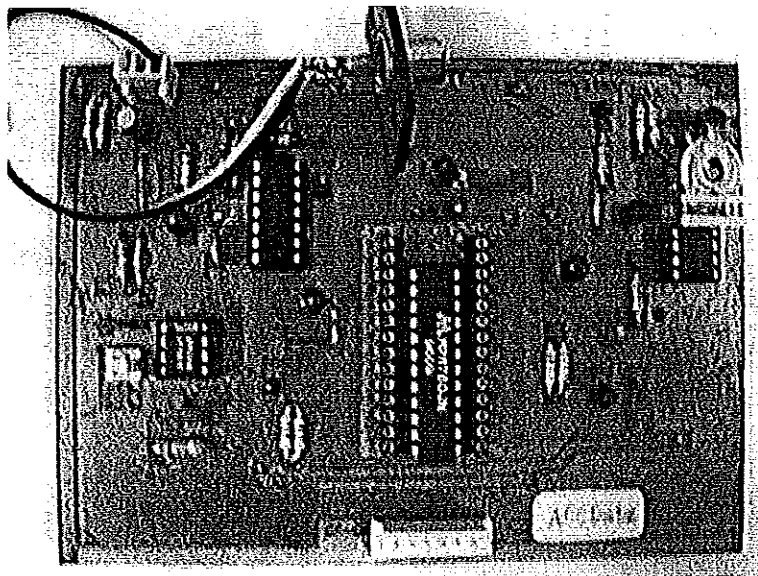
ส่วน Image Digitizer		ส่วนตัวควบคุม XC40003EPC84	
ส่วน	ชื่อสัญญาณ	ชื่อสัญญาณ	ขา (Pin)
1. วงจรแยกสัญญาณซิงก์	Vs	Vs	35
	Cs	Cs	51
2. วงจร ADC	ADCDat0-7	ADCDat0-7	36,47,38,39,40,44,45,46
3. วงจรกำเนิดสัญญาณนาฬิกา	Clk	Clk	13
4. หน่วยความจำแรม	RAMAdr0-15	RAMAdr0-15	10,9,8,7,6,5,4,3, 84,83,82,81,80,79,78,77
	RAMWE	RAMWE	68
	RAMOE	RAMOE	67
	RAMCE1	RAMCE1	70
	RAMCE2	RAMCE2	69
	RAMADC0-7	RAMADC0-7	57,58,59,60,61,62,65,66
5. พอร์ตขนานในโหมดการทำงานแบบ EPP	Write	EPPWrite	15
	D0-7	PCData0-7	20,23,24,25,26,27,28,19
	Wait	EPPWait	17
	Spare	TC	18
	Data strobe	EPPDatastrb	16
	Reset	RW	14
	Address strobe	Reset	29

4.4 การสร้างบอร์ดวงจรสำหรับทดสอบการทำงาน

ผู้วิจัยได้สร้างแผงวงจรพิมพ์จำนวน 2 บอร์ดสำหรับทดสอบการทำงานของตัวควบคุมและระบบรวมของ Image digitizer ที่ได้ออกแบบไว้ดังนี้

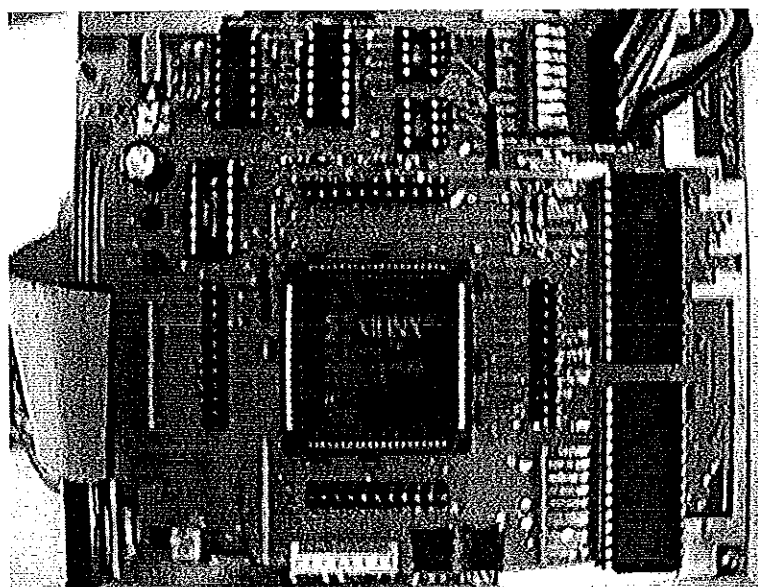
- บอร์ด PCB1 ประกอบด้วยวงจร Flash ADC และวงจรแยกสัญญาณซิงก์ดังแสดงใน

ภาพประกอบ 4-17 โดยสามารถดู Schematic diagram ของวงจรได้ในภาพประกอบ ค1 จากภาคผนวก ค



ภาพประกอบ 4-17 แสดงบอร์ด PCB1 สำหรับการทดสอบ

- บอร์ด PCB2 ประกอบด้วยส่วนของวงจรรวม FPGA วงจรกำเนิดสัญญาณนาฬิกา หน่วยความจำแรมและส่วนของพอร์ตขนานดังแสดงในภาพประกอบ 4-18 โดยสามารถดู Schematic diagram ของวงจรได้ในภาพประกอบ ค2 จากภาคผนวก ค



ภาพประกอบ 4-18 แสดงบอร์ด PCB2 สำหรับการทดสอบ

บทที่ 5

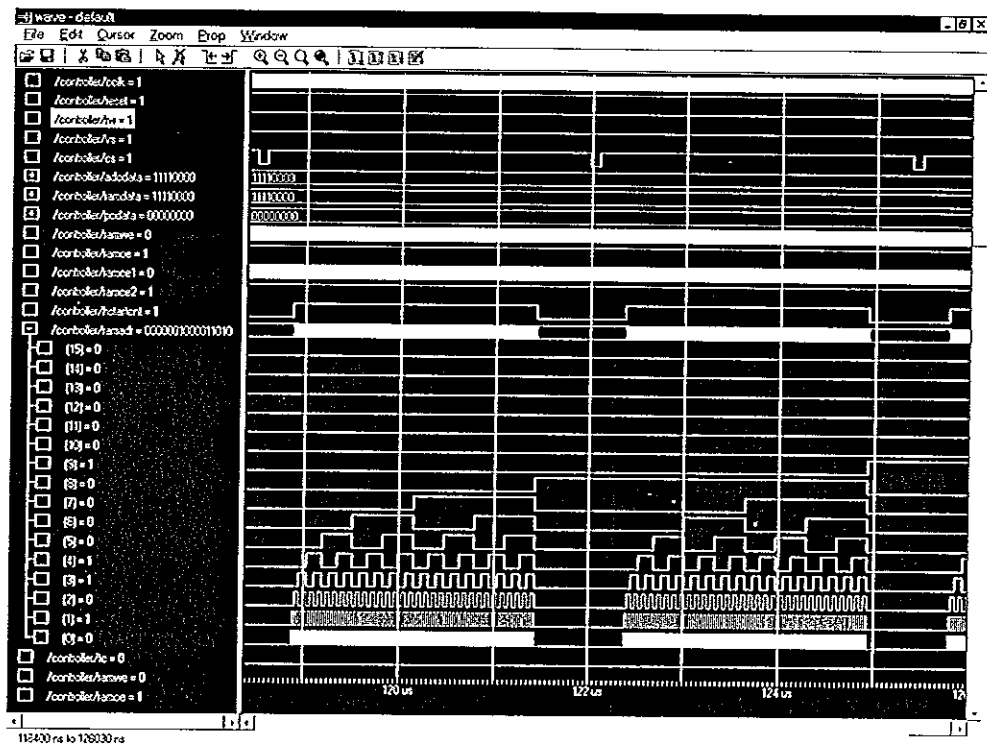
ผลการทดสอบและการวิเคราะห์ผล

ในบทที่แล้วเป็นการกล่าวถึงรายละเอียดหน้าที่การทำงานและการออกแบบในส่วนประกอบต่างๆของ Image digitizer และชิพตัวควบคุมรวมทั้งการสร้างแผงวงจรพิมพ์ทั้งบอร์ด PCB1 และ PCB2 สำหรับทดสอบการทำงาน สำหรับบทนี้จะเป็นการกล่าวถึงผลการจำลองการทำงาน ผลการทดสอบการทำงานจริงของตัวควบคุมและ Image digitizer แต่เนื่องจากสัญญาณทั้งหมดที่ได้จากการทดสอบมีจำนวนค่อนข้างมากจึงขอ นำผลการทดสอบและการวิเคราะห์ผลในส่วนของวงจรหลักและสัญญาณต่างๆในส่วนหลักทั้งสามส่วนภายในตัวควบคุมไปแสดงอย่างละเอียดในภาคผนวก ง โดยในบทนี้จะแสดงเฉพาะผลของสัญญาณหลักของตัวควบคุมที่ได้ออกแบบไว้และผลการเก็บภาพจาก Image digitizer โดยแสดงภาพที่ได้ในรูปแบบไฟล์ชนิด BMP

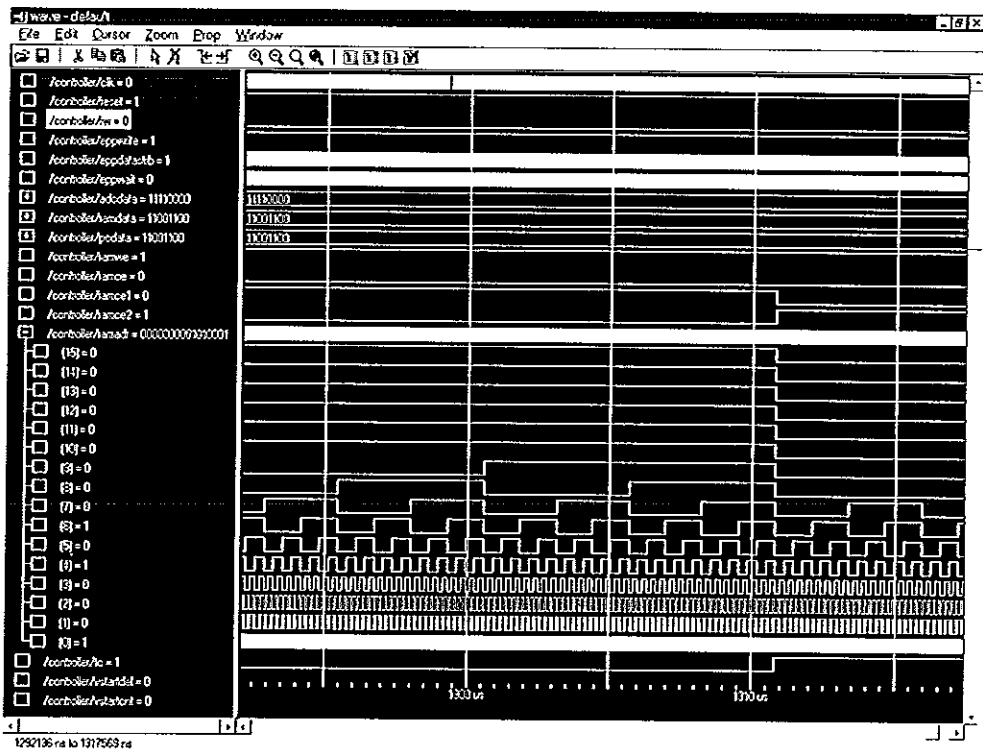
5.1 ผลการจำลองการทำงานของตัวควบคุม

เมื่อได้ทำการออกแบบและเขียนโปรแกรมภาษา VHDL เพื่อสร้างฟังก์ชันการทำงานของส่วนหลักทั้งสามส่วนภายในโครงสร้างของตัวควบคุมตามที่ต้องการแล้ว ก่อนเข้าสู่ขั้นตอนการทดสอบการทำงานจริงเราสามารถจำลองการทำงานของตัวควบคุมโดยใช้ซอฟต์แวร์สำหรับการจำลองการทำงาน (Simulation software) เพื่อยืนยันการทำงานในเบื้องต้นก่อนได้ ซึ่งในงานวิจัยนี้เลือกใช้ซอฟต์แวร์ชื่อ ModelSim EE/Plus 5.2e beta-3 ของบริษัท Model Technology

สำหรับผลการจำลองการทำงานของตัวควบคุมในทุกส่วนพบว่าตัวควบคุมสามารถทำงานได้ตามหน้าที่ที่ออกแบบไว้ โดยตัวอย่างผลการจำลองการทำงานของตัวควบคุมทั้งในโหมดการเขียนและอ่านข้อมูลภาพได้แสดงในภาพประกอบ 5-1 และ 5-2 ตามลำดับ



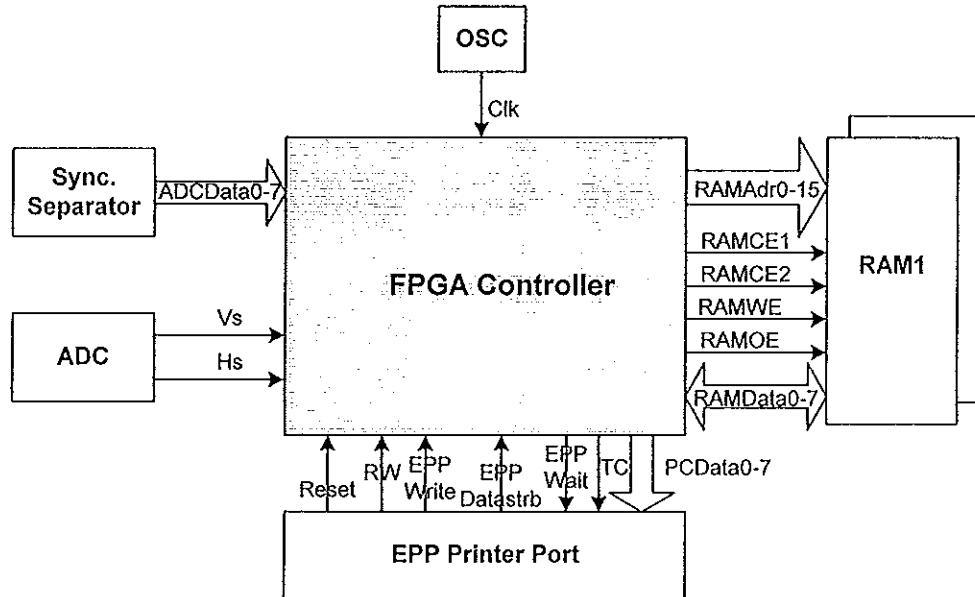
ภาพประกอบ 5-1 แสดงตัวอย่างผลการจำลองการทำงานของตัวควบคุมในโหมดการเขียน



ภาพประกอบ 5-2 แสดงตัวอย่างผลการจำลองการทำงานของตัวควบคุมในโหมดการอ่าน

5.2 ผลการทดสอบตัวควบคุม

สำหรับการทดสอบตัวควบคุมนี้ขอพิจารณาเฉพาะผลของสัญญาณต่างๆที่เกี่ยวข้องกับการทำงานตามหน้าที่หลักของตัวควบคุมที่ได้ออกแบบไว้ดังแสดงในภาพประกอบ 5-3



ภาพประกอบ 5-3 แสดงสัญญาณในการทดสอบการทำงานของตัวควบคุม

โดยสัญญาณทั้งหมดที่ใช้ในการทดสอบตัวควบคุมมีดังนี้

- สัญญาณ Vs และ Hs จากวงจรแยกสัญญาณซิงค์
- สัญญาณ Clk ความถี่ 5 เมกกะเฮิรตซ์จากวงจรกำเนิดสัญญาณนาฬิกา
- สัญญาณ ADCData จากวงจร ADC
- สัญญาณ RW ใช้สัญญาณ Reset จากพอร์ตขนาน
- สัญญาณ Reset ใช้สัญญาณ Address strobe จากพอร์ตขนาน
- สัญญาณ EPPWrite และ EPPDatastrb จากพอร์ตขนาน

การทดสอบการทำงานจะทดสอบทั้งในโหมดการเขียนข้อมูลภาพลงหน่วยความจำแรม และโหมดการอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่เครื่องคอมพิวเตอร์ในโหมดการทำงานแบบ EPP ดังนี้

5.2.1 ผลการทดสอบตัวควบคุมในโหมดการเขียนข้อมูลภาพ

ในโหมดการเขียนข้อมูลภาพหรือเมื่อสัญญาณ RW จากพอร์ตขนานมีค่าเป็นลอจิก '1'

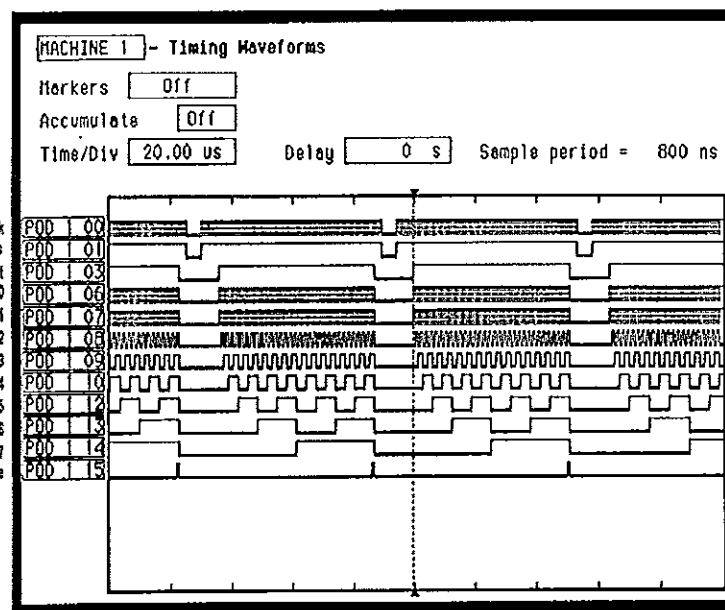
จะดูผลของสัญญาณต่างๆดังนี้

- สัญญาณ RAMAdr0-15 หรือแอดเดรสขนาด 16 บิตที่สร้างให้หน่วยความจำแรม
- สัญญาณควบคุม เช่น RAMWE, RAMOE, RAMCE1 และ RAMCE2
- สัญญาณ RAMData0-7 ซึ่งต้องมีค่าเท่ากับสัญญาณ ADCData0-7 จากวงจร ADC เพื่อเขียนข้อมูลภาพลงหน่วยความจำแรม โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ RAMData0 กับ RAMData7 และ ADCData0 กับ ADCData7 ตามลำดับ

- สัญญาณ TC เมื่อสิ้นสุดการสร้างแอดเดรสในโหมดการเขียน

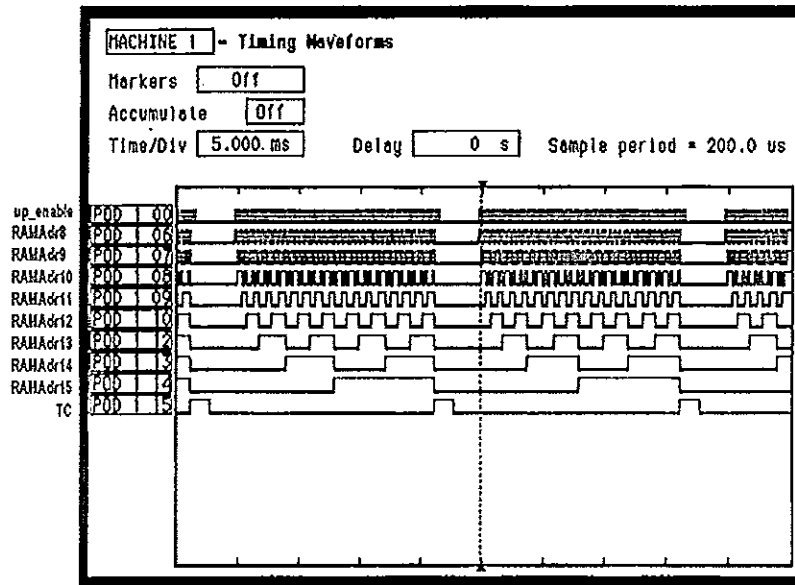
สำหรับผลของสัญญาณ RAMAdr0-15 ที่ได้จากการทดสอบจริงสามารถเทียบกับผลการจำลองการทำงานที่แสดงเป็นตัวอย่างในภาพประกอบ 5-1 ได้ซึ่งจะเห็นว่าให้ผลการทำงานเหมือนกัน ในทำนองเดียวกันกับผลการทำงานของส่วนอื่นๆก็ให้ผลตรงกับผลการจำลองการทำงานในส่วนนั้นเช่นเดียวกัน โดยผลการทดสอบที่ได้มีดังนี้

- สัญญาณ RAMAdr0-7



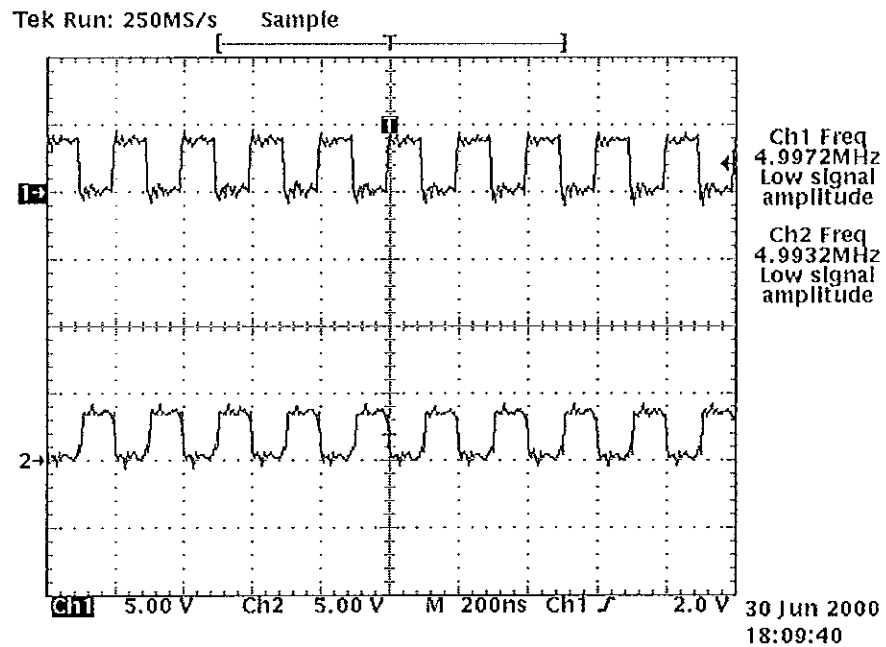
ภาพประกอบ 5-4 แสดงสัญญาณ RAMAdr0-7 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ RAMAdr8-15



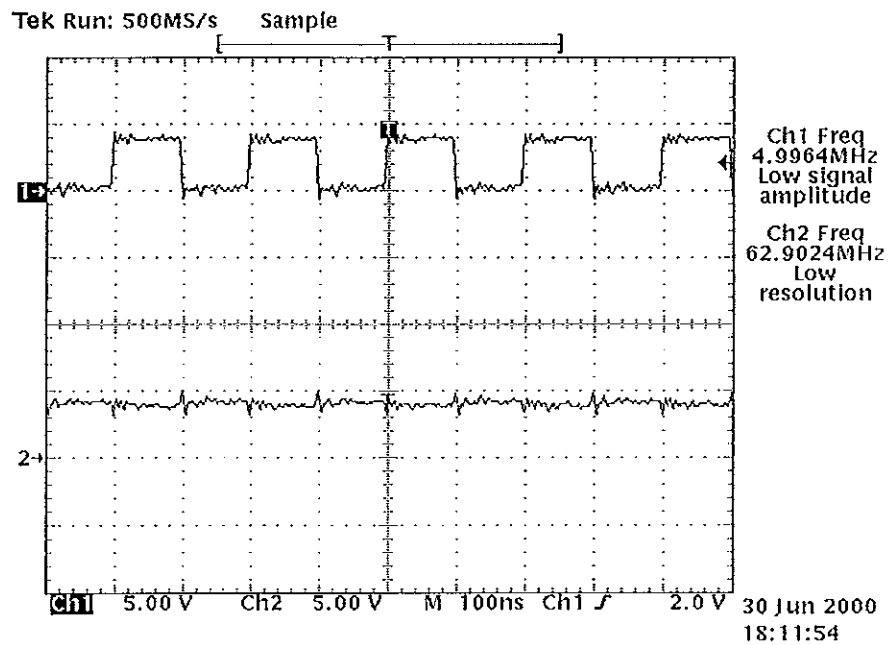
ภาพประกอบ 5-5 แสดงสัญญาณ RAMAdr8-15 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ Clk (Ch1) และสัญญาณ RAMWE (Ch2)



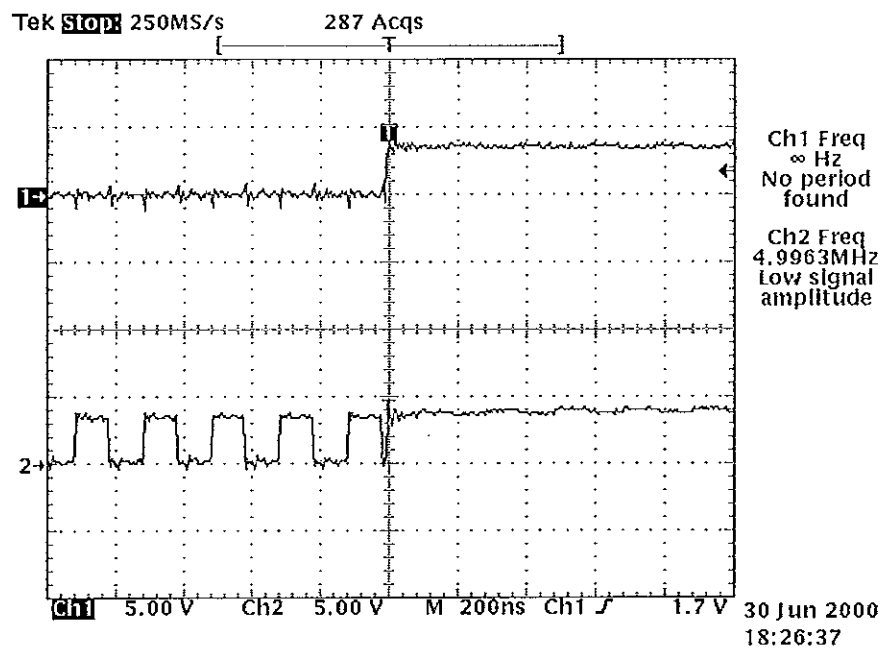
ภาพประกอบ 5-6 แสดงสัญญาณ Clk และสัญญาณ RAMWE เมื่ออยู่ในโหมดการเขียน

- สัญญาณ RAMWE (Ch1) และสัญญาณ RAMOE (Ch2)



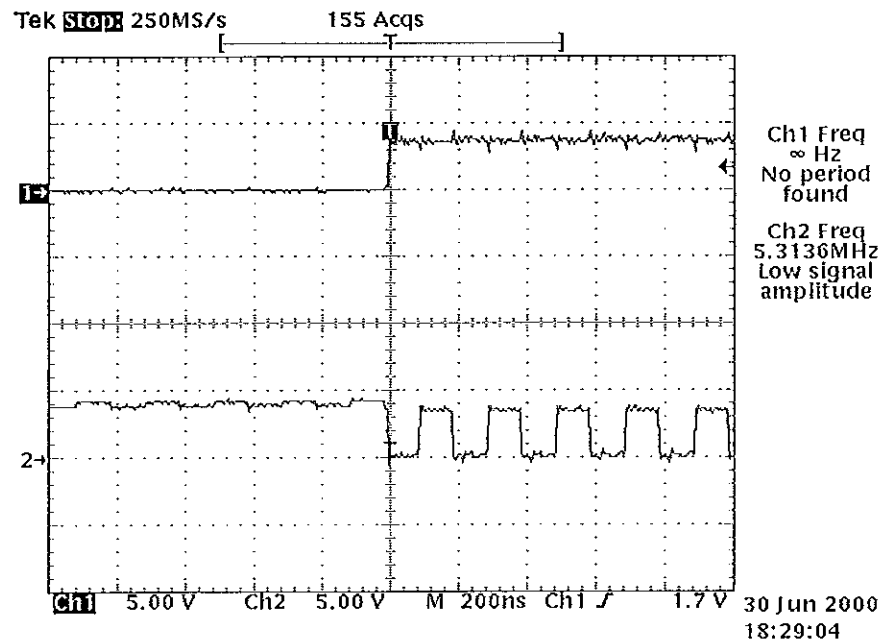
ภาพประกอบ 5-7 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการเขียน

- สัญญาณ RAMAdr15 (Ch1) และสัญญาณ RAMCE1 (Ch2)



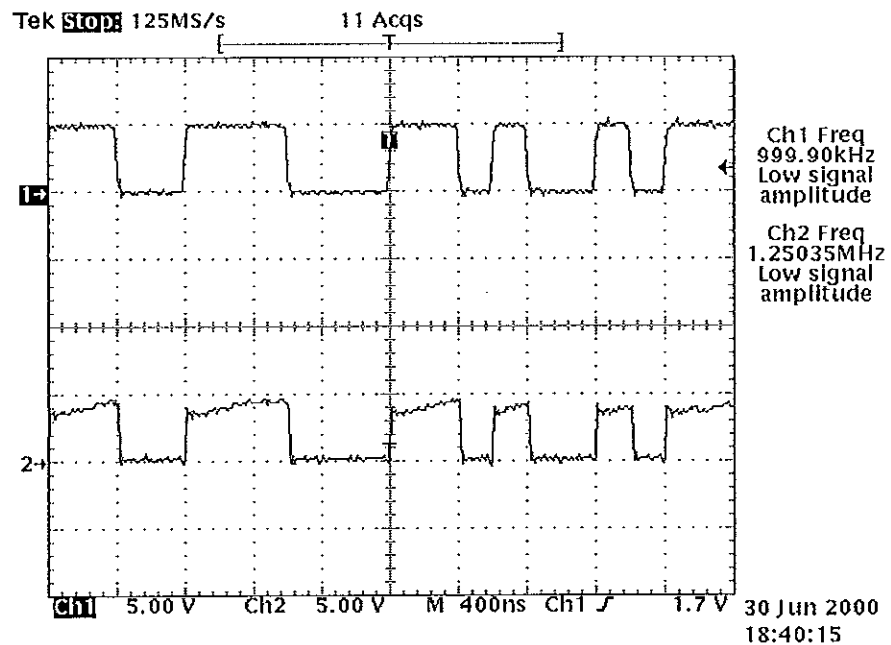
ภาพประกอบ 5-8 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ RAMAdr15 (Ch1) และสัญญาณ RAMCE2 (Ch2)



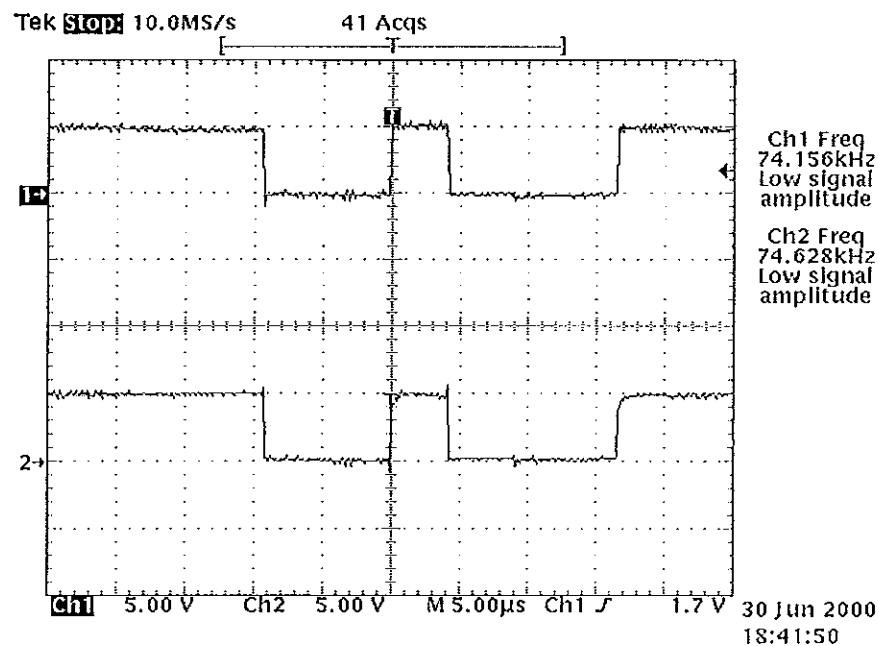
ภาพประกอบ 5-9 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE2 เมื่ออยู่ใน
โหมดการเขียน

- สัญญาณ ADCData0 (Ch1) และสัญญาณ RAMData0 (Ch2)



ภาพประกอบ 5-10 แสดงสัญญาณ ADCData0 และสัญญาณ RAMData0 เมื่ออยู่ใน
โหมดการเขียน

- สัญญาณ ADCData7 (Ch1) และสัญญาณ RAMData7 (Ch2)



ภาพประกอบ 5-11 แสดงสัญญาณ ADCData7 และสัญญาณ RAMData7 เมื่ออยู่ในโหมดการเขียน

ผลการวิเคราะห์

- จากภาพประกอบ 5-4 และ 5-5 แสดงสัญญาณ RAMAdr0-7 และ RAMAdr8-15 พบว่าตัวควบคุมสามารถสร้างแอดเดรสทั้ง 16 บิตและส่งสัญญาณ TC เมื่อสร้างแอดเดรสจนครบตามที่ออกแบบไว้

- จากภาพประกอบ 5-6 และ 5-7 แสดงสัญญาณ RAMWE และ RAMOE โดยพบว่าสัญญาณ RAMWE จะมีค่าเท่ากับสัญญาณ Clk ที่อินเวอร์ตในขณะที่สัญญาณ RAMOE มีค่าเป็นลอจิก '1' ส่วนในภาพประกอบ 5-8 และ 5-9 แสดงสัญญาณ RAMCE1 และ RAMCE2 โดยสัญญาณทั้งสองจะเกิดขึ้นสลับกันคือสัญญาณ RAMCE1 จะแอกทีฟ (มีค่าเท่ากับสัญญาณ Clk ที่อินเวอร์ต) เมื่อสัญญาณ RAMAdr15 มีค่าลอจิก '0' ส่วนสัญญาณ RAMCE2 จะแอกทีฟเมื่อสัญญาณ RAMAdr15 มีค่าลอจิก '1' ตามที่ออกแบบไว้

- จากภาพประกอบ 5-10 และ 5-11 แสดงผลการทดสอบในการส่งข้อมูล ADCData0-7 ให้หน่วยความจำแรมผ่านสัญญาณ RAMData0-7 โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ ADCData0 กับ ADCData7 และ RAMData0 กับ RAMData7 ตามลำดับ พบว่าสัญญาณทั้งสองมีค่าเหมือนกันตรงตามที่ออกแบบไว้เช่นกัน

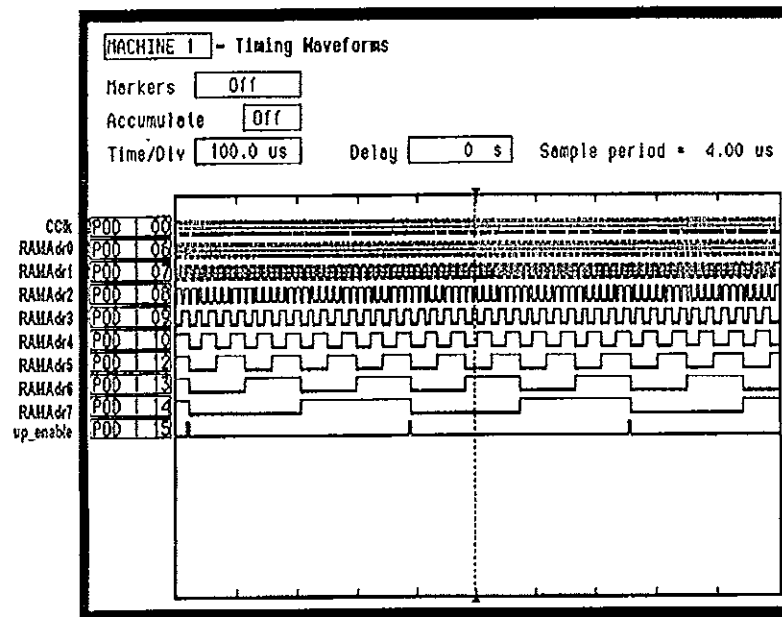
5.2.2 ผลการทดสอบตัวควบคุมในโหมดการอ่านข้อมูลภาพ

ในโหมดการอ่านข้อมูลภาพหรือเมื่อสัญญาณ RW จากพอร์ตนานมีค่าเป็นลอจิก '0' จะดูผลของสัญญาณต่างๆดังนี้

- สัญญาณ RAMAdr0-15 หรือแอดเดรสขนาด 16 บิตที่สร้างให้หน่วยความจำแรม
- สัญญาณควบคุม เช่น RAMWE, RAMOE, RAMCE1 และ RAMCE2
- สัญญาณ PCData0-7 ซึ่งต้องมีค่าเท่ากับสัญญาณ RAMData0-7 จากหน่วยความจำแรมเพื่ออ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่พอร์ตนาน โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ PCData0 กับ PCData7 และ RAMData0 กับ RAMData7 ตามลำดับ
- สัญญาณ TC เมื่อสิ้นสุดการสร้างแอดเดรสในโหมดการอ่าน
- สัญญาณ EPPWait ตอบรับเมื่อมีสัญญาณ EPPWrite และ EPPDatastrb จากพอร์ตนาน

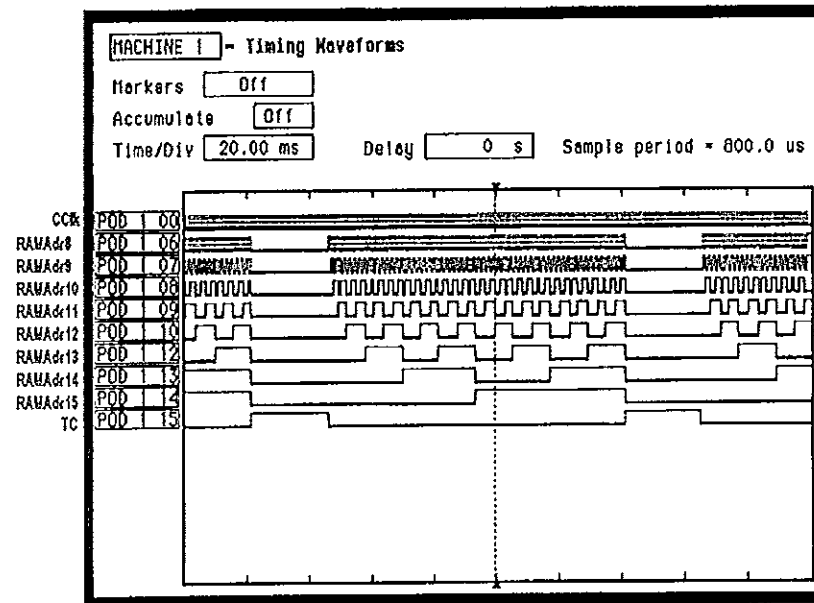
ผลการทดสอบที่ได้มีดังนี้

- สัญญาณ RAMAdr0-7



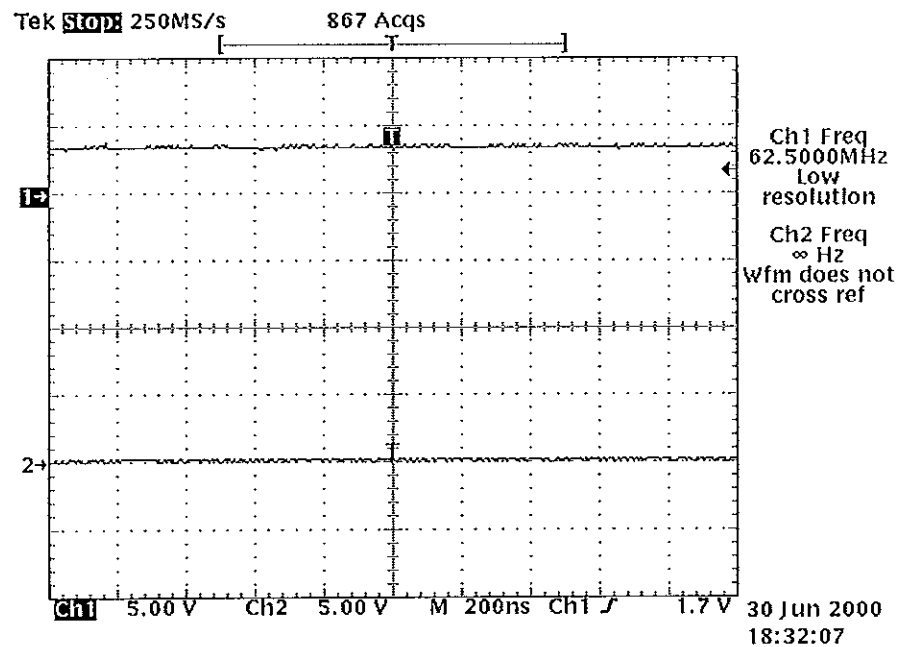
ภาพประกอบ 5-12 แสดงสัญญาณ RAMAdr0-7 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMAdr8-15



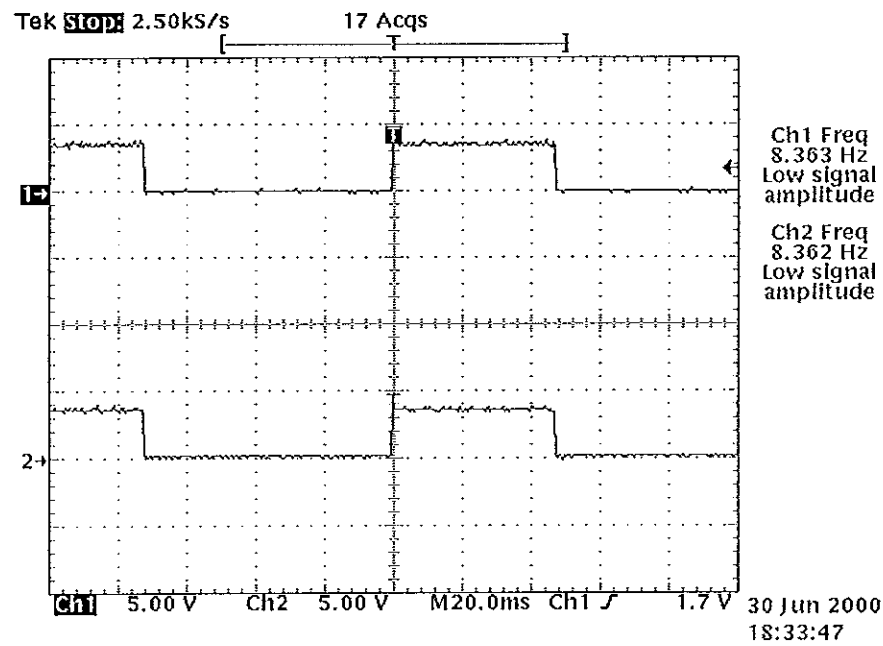
ภาพประกอบ 5-13 แสดงสัญญาณ RAMAdr8-15 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMWE (Ch1) และสัญญาณ RAMOE (Ch2)



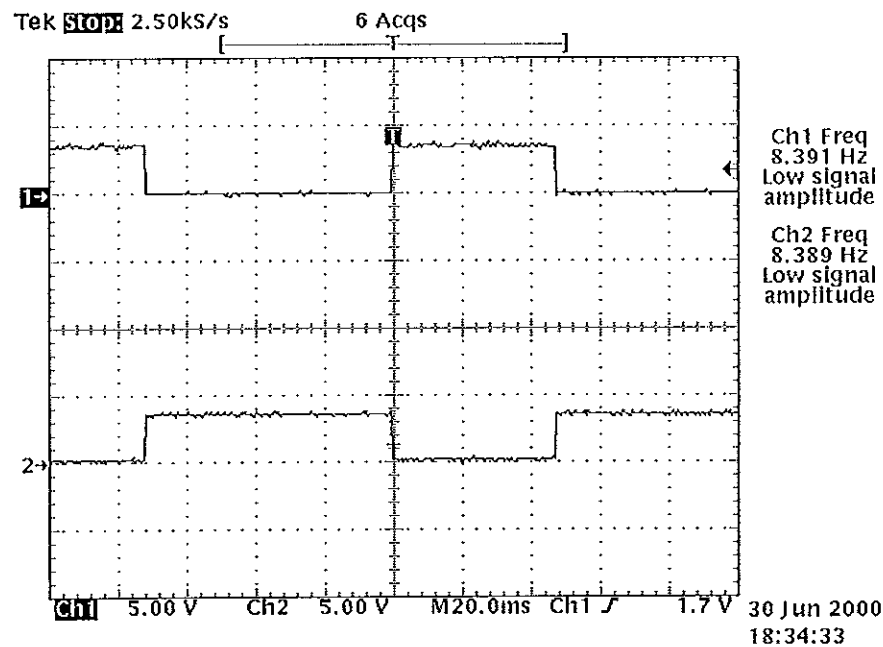
ภาพประกอบ 5-14 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMAdr15 (Ch1) และสัญญาณ RAMCE1 (Ch2)



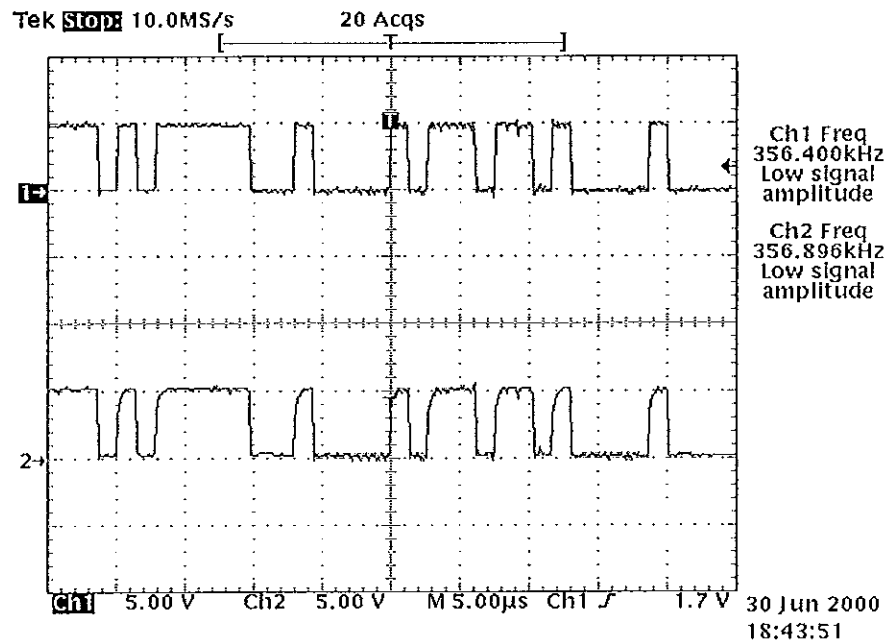
ภาพประกอบ 5-15 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMAdr15 (Ch1) และสัญญาณ RAMCE2 (Ch2)



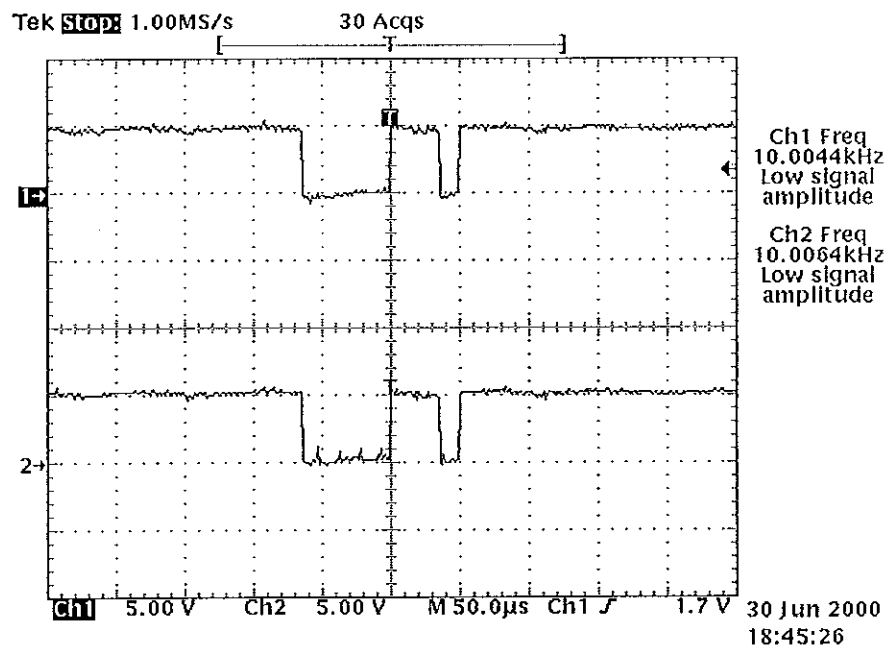
ภาพประกอบ 5-16 แสดงสัญญาณ RAMAdr15 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMData0 (Ch1) และสัญญาณ PCData0 (Ch2)



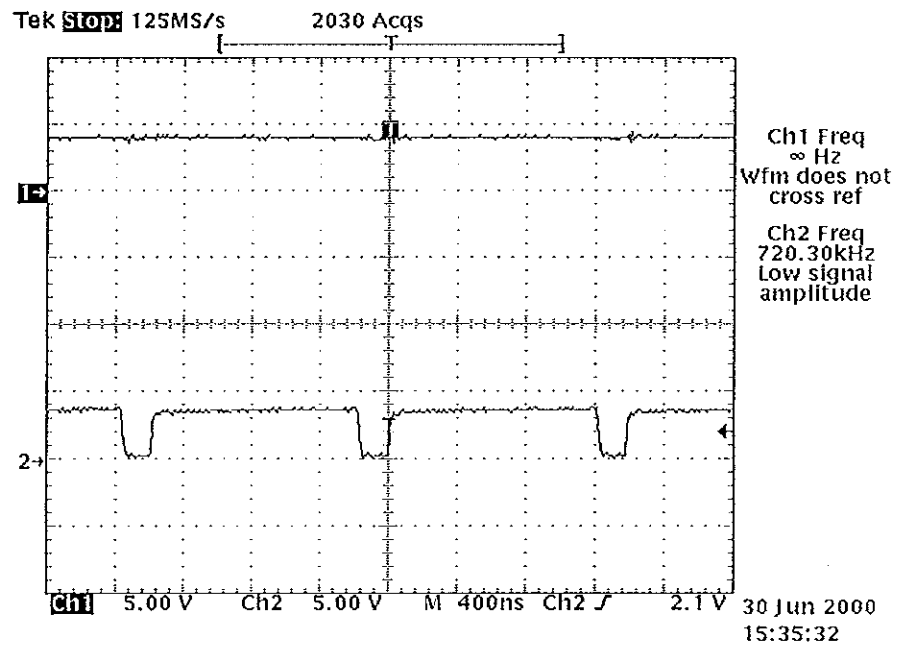
ภาพประกอบ 5-17 แสดงสัญญาณ RAMData0 และสัญญาณ PCData0 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMData7 (Ch1) และสัญญาณ PCData7 (Ch2)



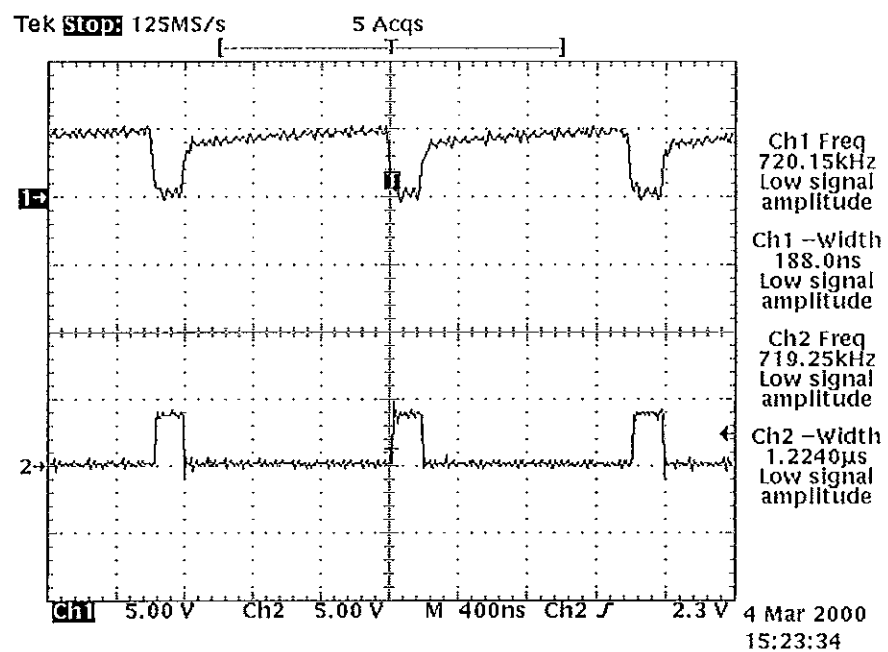
ภาพประกอบ 5-18 แสดงสัญญาณ RAMData7 และสัญญาณ PCData7 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ EPPWrite (Ch1) และสัญญาณ EPPDatastrb (Ch2) จากพอร์ตขนาน



ภาพประกอบ 5-19 แสดงสัญญาณ EPPWrite และสัญญาณ EPPDatastrb จากพอร์ตขนาน

- สัญญาณ EPPDatastrb (Ch1) เมื่อมีสัญญาณ EPPWait ตอบรับ (Ch2)



ภาพประกอบ 5-20 แสดงสัญญาณ EPPDatastrb เมื่อมีสัญญาณ EPPWait ตอบรับ

ผลการวิเคราะห์

- จากภาพประกอบ 5-12 และ 5-13 แสดงสัญญาณ RAMAdr0-7, RAMAdr8-15 และ สัญญาณ TC ในการสร้างแอดเดรสของตัวควบคุมตามที่ออกแบบ

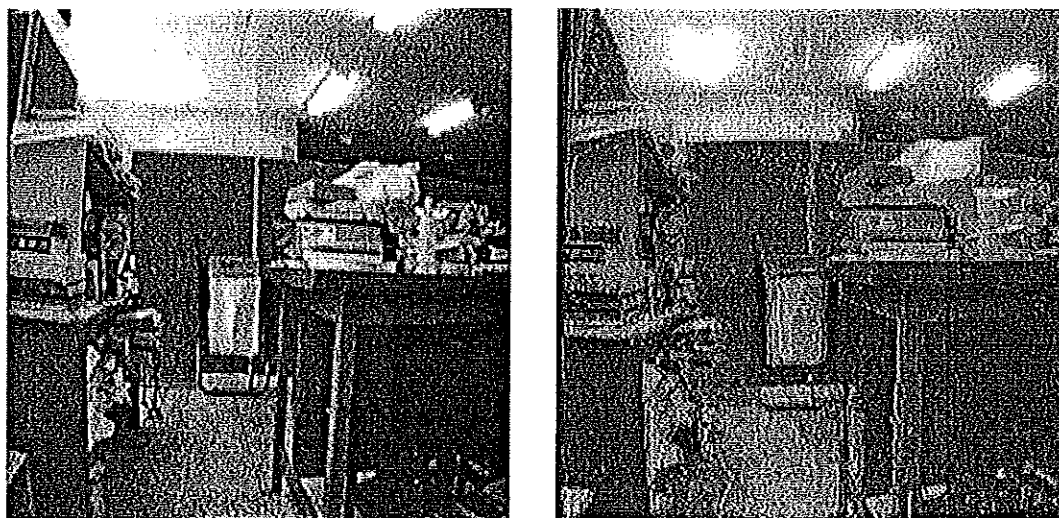
- จากภาพประกอบ 5-14 แสดงสัญญาณ RAMWE และ RAMOE พบว่าสัญญาณ RAMWE จะมีค่าลอจิก '1' ขณะที่สัญญาณ RAMOE มีค่าลอจิก '0' และจากภาพประกอบ 5-15 และ 5-16 แสดงสัญญาณ RAMCE1 และ RAMCE2 พบว่าสัญญาณทั้งสองจะแอกทีฟ (มีค่าเป็นลอจิก '0') หรือเกิดขึ้นสลับกันเช่นเดียวกับในโหมดการเขียนซึ่งเป็นไปตามที่ออกแบบ

- จากภาพประกอบ 5-17 ถึง 5-18 แสดงผลการทดสอบในการรับข้อมูลจากหน่วยความจำแรมคือสัญญาณ RAMData0-7 ส่งให้เครื่องคอมพิวเตอร์ผ่านทางพอร์ตขนานซึ่งคือสัญญาณ PCData0-7 โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ RAMData0 กับ RAMData7 และ PCData0 กับ PCData7 ตามลำดับ ซึ่งพบว่าสัญญาณทั้งสองมีค่าเหมือนกันตรงตามที่ออกแบบเช่นกัน

- และจากภาพประกอบ 5-19 ถึง 5-20 แสดงสัญญาณทั้ง 3 สัญญาณที่ใช้ในการทำ Handshaking คือสัญญาณ EPPWrite, EPPDatastrb และ EPPWait พบว่าสามารถสร้างสัญญาณ EPPWait ได้ตามที่ออกแบบ

5.3 ผลการทดสอบวงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล

ภาพที่ได้จาก Image digitizer จะถูกบันทึกให้อยู่ในรูปแบบไฟล์ภาพชนิด BMP สำหรับผลการทดสอบจะแสดงภาพตัวอย่างที่เก็บได้จริงซึ่งมีความละเอียดของภาพขนาด 256×256 จุดต่อภาพโดยเปรียบเทียบกับภาพที่เก็บได้จาก Video capture card ยี่ห้อ Tekram รุ่น M205 Pro ซึ่งมีความละเอียดของภาพขนาด 384×288 จุดต่อภาพดังแสดงในภาพประกอบ 5-21



ภาพประกอบ 5-21 แสดงภาพตัวอย่างที่ได้จาก Image digitizer (รูปขวามือ) เมื่อเปรียบเทียบกับภาพที่เก็บได้จาก Video capture card (รูปซ้ายมือ)

ผลการวิเคราะห์

ผลการทดสอบจะเห็นว่าภาพที่เก็บได้จาก Image digitizer ในงานวิจัยซึ่งมีความละเอียดภาพขนาด 256×256 จุดต่อภาพสามารถให้รายละเอียดของวัตถุต่างๆ โดยเฉพาะวัตถุที่มีขนาดใหญ่ได้ใกล้เคียงกับภาพที่เก็บได้จาก Video capture card ที่จำหน่ายโดยทั่วไปซึ่งในที่นี้มีความละเอียดภาพขนาด 384×288 จุดต่อภาพ โดยภาพทั้งสองมีจุดเริ่มต้นของภาพที่ต่างกันเนื่องจาก Video capture card โดยปกติจะเริ่มเก็บภาพเมื่อผ่านช่วงที่เป็นสัญญาณแบลนคิงก็คือเข้าสู่เส้นกวาดเส้นที่ 19 (ดังรายละเอียดของสัญญาณภาพรวมในบทที่ 2) ในขณะที่งานวิจัยนี้ออกแบบให้ Image digitizer เริ่มทำงานเมื่อเข้าสู่เส้นกวาดเส้นที่ 32 ตามที่ได้กล่าวไปแล้ว นอกจากนี้การแสดงผลโดยเฉพาะส่วนที่เป็นสีดำและสีขาวของภาพจาก Image digitizer ในงานวิจัยจะไม่ชัดเจนเมื่อเทียบกับภาพจาก Video capture card ทำให้ภาพที่ได้มีความคมชัดน้อยกว่าทั้งนี้ขึ้นกับการเลือกค่าระดับแรงดันอ้างอิงของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลในงานวิจัยที่มีช่วงการเปลี่ยนแปลงที่น้อยกว่าซึ่งหากต้องการความคมชัดที่มากขึ้นก็สามารถปรับเปลี่ยนค่าแรงดันอ้างอิงนี้ได้ตามต้องการและการเลือกจำนวนบิตต่อจุดภาพเท่ากับ 8 บิตทำให้สามารถแสดงระดับเทาในแต่ละจุดภาพได้เพียง 256 ระดับตามที่กล่าวไปแล้วในบทที่ 4 แต่ถึงอย่างไรก็ตามจากผลของภาพที่ได้แสดงให้เห็นว่าระบบรวมของ Image digitizer และตัวควบคุมที่ออกแบบไว้สามารถทำงานและให้ผลตรงตามวัตถุประสงค์ของงานวิจัย

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

จากงานวิจัยการออกแบบและสร้าง Image digitizer ซึ่งรวมถึงตัวควบคุมที่อยู่ในรูปวงจรรวม FPGA สำหรับควบคุมกระบวนการเขียนข้อมูลภาพลงหน่วยความจำแรมและกระบวนการอ่านข้อมูลภาพจากหน่วยความจำแรมเข้าสู่พอร์ตนานของเครื่องคอมพิวเตอร์ในโหมดการทำงานแบบ EPP โดยสามารถสรุปคุณสมบัติทางด้านฮาร์ดแวร์และซอฟต์แวร์ของ Image digitizer ได้ดังนี้

คุณสมบัติทางด้านฮาร์ดแวร์

1. สามารถเก็บภาพแบบเฟรมเดียว (Single frame acquisition) จากกล้องวิดีโอซึ่งเป็นสัญญาณภาพในระบบ PAL และมีการกวาดภาพแบบสอดสลับ (Interlaced scanning)
2. มีความละเอียดภาพขนาด 256X256 จุดต่อภาพ
3. ให้ระดับเทาในแต่ละจุดภาพ 256 ระดับ (จำนวน 8 บิตต่อจุดภาพ)
4. ใช้ได้เฉพาะสัญญาณภาพขาวดำ
5. มีความถี่ในการสุ่มสัญญาณวิดีโอ (Sampling frequency) เท่ากับ 5 เมกกะเฮิร์ตซ์
6. สามารถโอนย้ายข้อมูลดิจิทัลเข้าสู่เครื่องคอมพิวเตอร์โดยผ่านทางพอร์ตนานในโหมดการทำงานแบบ EPP (Enhanced Parallel Port)
7. ไม่มีส่วน Video output เนื่องจากภาพจะถูกแสดงบนเครื่องคอมพิวเตอร์เท่านั้น
8. ใช้วงจรรวม FPGA (Field Programmable Gate Array) เบอร์ XC4003E ในตระกูล XC4000E ของบริษัท Xilinx เป็นตัวควบคุมการทำงานทั้งหมด

คุณสมบัติทางด้านซอฟต์แวร์

สามารถบันทึกภาพที่ถูก digitized ในรูปแบบไฟล์ภาพชนิด BMP ได้

สำหรับการทดสอบการทำงานของตัวควบคุมพบว่าสามารถทำงานได้ตามที่ออกแบบไว้ทั้งสามส่วนหลัก โดยสรุปผลการทำงานในแต่ละส่วนได้ดังนี้

1. ส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม (RAMAdrCounter) สามารถสร้างแอดเดรสจำนวน 16 บิตให้หน่วยความจำแรม คือเมื่ออยู่ในโหมดการเขียนข้อมูลภาพตัวควบคุมจะใช้สัญญาณ Hs และ Vs จากวงจรแยกสัญญาณซิงก์และสัญญาณ Clk จากวงจรกำเนิดสัญญาณนาฬิกาในการ

สร้างแอดเดรส โดยเริ่มสร้างแอดเดรสเมื่อเกิดสัญญาณ Vs และสัญญาณ Hs ลูกที่ 32 แต่เมื่ออยู่ในโหมดการอ่านข้อมูลภาพตัวควบคุมจะใช้สัญญาณ Data strobe ที่อินเวอร์ตจากพอร์ตขนานในการสร้างแอดเดรสแทนและสร้างสัญญาณ TC เมื่อสร้างแอดเดรสครบในทั้งสองโหมดการทำงานซึ่งตรงตามทีออกแบบไว้

2. ส่วนการอินเทอร์เฟสกับหน่วยความจำแรม (RAM Interface) สามารถส่งแอดเดรสจำนวน 16 บิตให้หน่วยความจำแรมทั้งในโหมดการเขียนและอ่านข้อมูลภาพ ส่งข้อมูลดิจิทัลที่ได้จากวงจร ADC จำนวน 8 บิตให้หน่วยความจำแรมเมื่ออยู่ในโหมดการเขียนข้อมูลภาพ รับข้อมูลดิจิทัลจากหน่วยความจำแรมจำนวน 8 บิตส่งให้เครื่องคอมพิวเตอร์ในโหมดการอ่านข้อมูลภาพ และสร้างสัญญาณควบคุมให้หน่วยความจำแรมในทั้งสองโหมดการทำงานได้ซึ่งเป็นไปตามทีออกแบบ

3. ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP (EPPSig-Cntl) สามารถสร้างสัญญาณ Wait ตอรับเมื่อมีสัญญาณ Write และ Data strobe จากพอร์ตขนานเพื่อทำให้เกิดการ Handshaking ในโหมดการทำงานแบบ EPP ซึ่งตรงตามทีออกแบบไว้เช่นกัน

โดยสามารถสรุปผลการใช้ทรัพยากร (Resources) ต่างๆภายในวงจรรวม FPGA เบอร์ XC4003E ที่ใช้เป็นตัวควบคุมดังตาราง 6-1

ตาราง 6-1 แสดงผลการใช้ทรัพยากรภายในวงจรรวม FPGA เบอร์ XC4003E ที่ใช้เป็นตัวควบคุม

ทรัพยากร (Resources)	จำนวนที่ใช้ (Used number)	จำนวนที่มีอยู่ (Available number)	ปริมาณที่ใช้ไป (%) (Utilization)
IOBs	53	61	86.89
FG Function Generators	134	200	67
H Function Generators	18	100	18
CLB Flip Flops	56	360	15.56

โดยพบว่า

- จำนวนของ IOBs หรือ Input Output Blocks ซึ่งแสดงถึงขา (Pin) ของสัญญาณอินพุตและสัญญาณเอาต์พุตที่ใช้ในตัวควบคุมนี้มีจำนวนทั้งสิ้น 53 ขาจากจำนวนขาที่มีอยู่ทั้งหมด 61 ขา

หรือคิดเป็น 86.89 เปอร์เซ็นต์โดยเขาที่เหลืออีกจำนวน 8 เขาสามารถใช้เป็นเขาสำหรับคูณผลของสัญญาณต่างๆภายในตัวควบคุมได้ตามต้องการ

- จำนวนของ Function Generators ชนิด F และ G ใช้ไปจำนวน 134 ตัวจากจำนวนทั้งหมด 200 ตัวคิดเป็น 67 เปอร์เซ็นต์และจำนวนของ Function Generators ชนิด H ใช้ไปจำนวน 18 ตัวจากจำนวนทั้งหมด 100 ตัวคิดเป็น 18 เปอร์เซ็นต์ ในขณะที่จำนวนของ CLB (Configurable Logic Block) Flip Flops ใช้ไปจำนวน 56 ตัวจากจำนวนทั้งหมด 360 ตัวคิดเป็น 15.67 เปอร์เซ็นต์เท่านั้น

สำหรับปริมาณการใช้ทรัพยากรแต่ละชนิดนี้จะสะท้อนให้เห็นถึงผลที่ได้จากการเลือกรูปแบบการเขียนโปรแกรมภาษา VHDL (VHDL coding style) และผลที่ได้จากการทำงานของซอฟต์แวร์ในกระบวนการต่างๆ โดยจากตาราง 6-4 แสดงให้เห็นว่ารูปแบบการเขียนโปรแกรมภาษา VHDL ของการออกแบบตัวควบคุมในงานวิจัยนี้ให้ผลการใช้ทรัพยากรในปริมาณที่ไม่มากนัก โดยเฉพาะจำนวนของ CLB Flip Flops ที่ใช้เพียง 15.67 เปอร์เซ็นต์ ดังนั้นปริมาณที่เหลืออีก 84.33 เปอร์เซ็นต์ทำให้เราสามารถปรับเปลี่ยนรูปแบบการเขียนโปรแกรมภาษา VHDL หรือโครงสร้างของส่วนต่างๆภายในตัวควบคุมได้ตามต้องการ

สำหรับผลการทดสอบการทำงานของวงจรหลักต่างๆสามารถทำงานได้ตามต้องการและในส่วนผลการทำงานของ Image digitizer พบว่าสามารถเก็บภาพให้มีความละเอียดภาพขนาด 256×256 จุดต่อภาพได้ตามวัตถุประสงค์ของงานวิจัย

บทวิจารณ์และข้อเสนอแนะ

1. การนำสัญญาณ Clk ความถี่ 5 เมกกะเฮิร์ตซ์ของวงจรถูกกำเนิดสัญญาณนาฬิกาจากบอร์ด PCB2 เพื่อนำมาใช้เป็นอัตราในการแปลงสัญญาณภาพของวงจร ADC ในบอร์ด PCB1 นั้น ในเบื้องต้นเมื่อใช้วงจรรวมเบอร์ 74LS74 เป็นวงจรถูกจะเกิดปัญหาคือแรงดันของสัญญาณ Clk จะมีขนาดลดลงจากเดิม 3.8 โวลต์เหลือ 2 โวลต์เพราะวงจรรวมเบอร์ 74LS74 ไม่สามารถขับโหลด (Load) คือวงจร ADC ได้ซึ่งอาจไม่เพียงพอที่จะทำให้วงจร ADC ทำงานได้ จึงได้แก้ไขปัญหานี้โดยใช้วงจรรวมเบอร์ 74F74 แทนซึ่งจะมีค่ากระแสเอาต์พุตสูงสุดเมื่อเป็นลอจิกต่ำ (Maximum Output Current-Low, $I_{OL(max)}$) เท่ากับ 20 มิลลิแอมแปร์และมีค่าเวลาประวิงการแพร่กระจายสูงสุด (Maximum Propagation Delay, $t_{PLH(max)}$) เท่ากับ 6.8 นาโนวินาที ในขณะที่เบอร์ 74LS74 จะมีค่า $I_{OL(max)}$ เท่ากับ 8 มิลลิ

และความยาวของสายสัญญาณทั้งหมดที่ใช้ในทุกวงจร และเพิ่มตัวต้านทานขนาด 10 กิโลโห์มมา pull up สัญญาณต่างๆที่เกิดปัญหาพร้อมทั้งข้อมูลทั้ง 8 บิตในทั้ง 2 จุดไว้ โดยพบว่าสามารถลดสัญญาณรบกวนที่เกิดขึ้นได้ในระดับที่น่าพอใจ

ส่วนการแก้ปัญหาในเรื่องของการเกิด Cross talk ระหว่างข้อมูล 8 บิตของพอร์ตขนานตามที่ได้สันนิษฐานไว้นั้น ผู้วิจัยได้ทำการดัดแปลงสายแพที่ใช้เป็นสายสัญญาณในการนำข้อมูลเข้าพอร์ตขนานโดยใช้ตัวต้านทานขนาด 150 โหห์มมาต่ออนุกรมกับสัญญาณข้อมูล PCData ทั้ง 8 บิต (PCData0-7) ไว้ สำหรับการแก้ปัญหาเช่นนี้ได้พิจารณาในเรื่องการลดการเปลี่ยนแปลงของค่า dV/dt ของสัญญาณ เนื่องจากสายสัญญาณโดยปกติจะถูกพิจารณาว่าเปรียบเสมือนมีตัวเก็บประจุค่าหนึ่งต่อลงกราวด์อยู่ที่ปลายสาย การต่อตัวต้านทานอนุกรมกับสายสัญญาณนี้จะทำให้ค่าการหน่วงเวลาของสัญญาณมีค่ามากขึ้นทำให้ความชันทั้งขอบขาขึ้นและขอบขาลงของสัญญาณมีค่าลดลง ส่งผลให้การเปลี่ยนแปลงค่า dV/dt มีค่าลดลงเช่นกันทำให้สามารถแก้ปัญหการเกิด Cross talk ได้

สำหรับผลที่ได้จากการแก้ปัญหาดังกล่าวมาทั้งหมดนี้ทำให้ Image digitizer และตัวควบคุมสามารถทำงานได้ตามปกติ

3. ความนิ่งของกล้อง CCD หรือกล้องวิดีโอที่มีส่วนสำคัญมากที่จะทำให้ภาพที่เก็บได้มีความต่อเนื่องไม่ขาดตอน ซึ่งหากกล้องเกิดการสั่นหรือเคลื่อนไหวระหว่างการเก็บภาพก็จะทำให้ได้ภาพที่ไม่ต่อเนื่อง ดังนั้นจึงควรใช้แคลมป์หนีบเพื่อยึดกล้องให้อยู่กับที่และวางในตำแหน่งที่มีระนาบเรียบ

4. ความสว่างของวัตถุหรือฉากที่ต้องการเก็บภาพก็มีผลต่อความคมชัดของภาพด้วยเช่นกัน โดยหากต้องการภาพที่คมชัดควรให้แสงหรือความสว่างกับวัตถุและฉากให้มากพอ

5. ภาพที่เก็บได้จาก Image digitizer นี้ในบางครั้งอาจมีปัญหาในเรื่องการเอียงของภาพได้ ทั้งนี้เนื่องจากการเลือกใช้วิธีการสร้างสัญญาณนาฬิกาความถี่ 5 เมกะเฮิรตซ์ซึ่งมีการ synchronize กับสัญญาณ Hs ในลักษณะ Digital synchronization อาจไม่มีความแม่นยำมากนักและไม่ใช่วิธีการที่ดีที่สุดเพราะไม่สามารถควบคุมสัญญาณ Hs ได้ วิธีการที่ดีที่สุดจึงควรเลือกการสร้างสัญญาณนาฬิกาในลักษณะ Analog synchronization โดยใช้เฟสล็อกลูป (Phase-locked loop) เพื่อให้เกิดความแม่นยำมากขึ้น สำหรับวัตถุประสงค์ของงานวิจัยนี้เป็นการออกแบบและสร้างวงจรรวม FPGA เพื่อเป็นตัวควบคุมใน Image digitizer ดังนั้นข้อเสนอแนะนี้จึงอาจใช้เป็นข้อพิจารณาในเรื่องการออกแบบวงจรเพื่อปรับปรุงและพัฒนาการทำงาน Image digitizer ต่อไป

6. เนื่องจากในงานวิจัยนี้ได้เลือกความละเอียดของภาพให้มีขนาด 256×256 จุดต่อภาพซึ่งอาจเหมาะสมสำหรับการนำภาพไปใช้ในงานที่ไม่ต้องการความละเอียดของภาพมากนัก ดังนั้นเพื่อให้สามารถปรับเปลี่ยนความละเอียดของภาพให้สูงขึ้นและเป็นแนวทางในการวิจัยเพิ่มเติมต่อไปจึงควรพิจารณาตัวแปรที่สำคัญ เช่น ความถี่ของสัญญาณนาฬิกาที่ใช้ ขนาดหน่วยความจำแรม หรือจำนวนบิตของวงจรนับในตัวควบคุม เป็นต้น ตามที่ผู้วิจัยได้กล่าวไปแล้วในบทที่ 3

7. นอกจากข้อเสนอแนะในเรื่องการปรับปรุงการออกแบบวงจรของ Image digitizer แล้วในเรื่องการพัฒนาการทำงานของตัวควบคุมก็สามารถทำได้ตามความต้องการเช่นกันเนื่องจากการออกแบบวงจรรวม FPGA ในลักษณะ Semi-custom นี้ทำให้ผู้ออกแบบสามารถปรับเปลี่ยนรูปแบบการเขียนโปรแกรม VHDL เพื่อสร้างฟังก์ชันการทำงานของวงจรรวมได้อย่างสะดวกและรวดเร็ว แต่มีข้อควรพิจารณาที่สำคัญคือเมื่อโปรแกรมมีความซับซ้อนมากขึ้นทำให้ขนาดของวงจรรวมที่ใช้ย่อมมากขึ้นด้วย ซึ่งอาจส่งผลในเรื่องค่าใช้จ่ายที่เกี่ยวกับวงจรรวมที่เพิ่มขึ้นเช่นกัน

บรรณานุกรม

- จรูญ ไชยนิทย์. 2542. "เครื่องตรวจขวดอัตโนมัติ (Automatic Bottle Inspecting Machine)".
วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรม
ศาสตร์ มหาวิทยาลัยสงขลานครินทร์. (สำเนา)
- ณรงค์ อย่างสกุล. 2537. "เทคนิคการออกแบบวงจรดิจิทัลสร้างภาพบนจอโทรทัศน์ตอน 1
และ 2", เซมิคอนดักเตอร์อิเล็กทรอนิกส์. 143(ต.ค. 2537) และ 144(พ.ย. 2537),
71-76 และ 65-71.
- บวร ปภัสราทร และ คณะ. 2533. เทคโนโลยีการออกแบบวงจรรวม. กรุงเทพฯ :สมาคม
ส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- บุญถึง แน่นหนา. 2520. ตำราโทรทัศน์ภาคทฤษฎีและปฏิบัติ. กรุงเทพฯ : สำนักพิมพ์
โอเดียนสโตร์.
- บัณฑิต สุมณวัฒน์เดช. 2539. "แผงวงจรเก็บข้อมูลภาพที่ประกอบด้วยฟังก์ชันหน่วยความจำ
(A Built-in Functions Memory Image Data Acquisition Card)", วิทยานิพนธ์
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้าคุณทหารลาดกระบัง. (สำเนา)
- จิวัฒน์ ศรีเพ็ง. 2534. "เครื่องแสดงภาพสีทางดิจิทัลขนาด 512×512 จุดต่อภาพ (512×512
Pixels of Color Image Display)", วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขา
วิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้าคุณ
ทหารลาดกระบัง. (สำเนา)
- สุรเชษฐ์ ศรีพลกั้ง และ บรรจง ปิยะธรรมวง. 2538. "การออกแบบระบบดิจิทัลโดยใช้ VHDL
(Digital System Design Using VHDL)", วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้า

คุณทหารลาดกระบัง. (สำเนา)

อรรถสิทธิ์ หล้าสกุล และ วิสุทธิ์ พิบูลวรารังกูร. 2531. "เครื่องเปลี่ยนสัญญาณภาพแบบเวลาจริง (Real Time Video Digitizer)", วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้าคุณทหาร ลาดกระบัง. (สำเนา)

Bush, G. and Smith, M.J. 1994. "Frame Grabber for Sequential Real Time Video Image Digitization and Transfer to Microcomputer System", Medical & Biological Engineering & Computing. (July 1994), 476-478.

Caircia, Steve. 1987. "Part 2 : Digitizer/Transmitter Build a Gray-Scale Video Digitizer", BYTE. (May 1987), 129-138.

Dhake, Arvind M. 1988. Television Engineering. 1st ed. New Delhi : TataMcGraw-Hill.

Fulcher, John. 1989. An Introduction to Microcomputer Systems : Architecture and Interfacing. 1st ed. Singapore : Addison-Wesley Publishing Company.

Grob, Bernard. 1964. Basic Television : Principle and Serving. 3rd ed. New York : McGraw-Hill Book Company.

Gulati, R. 1984. Monochrome and Colour Television. 2nd ed. New York : Wiley Eastern Limited.

Model Technology. 1998. HDL Simulation FPGA Design Methodology. October 1998 Edition.

Hopkins, Richard. 1997. "Video on the PC/104 Bus", Circuit Cellar. 85(Aug 1997), 49-54.

Lindley, Craig. 1991. Practical Image Processing In C. 1st ed. Canada : John Wiley & Sons. Inc.

Peacock, Craig. 1997. "Interfacing the Enhanced Parallel Port version 1.0", <http://www.senet.com.au/~cpeacock>.

Perry, Douglas L. 1995. VHDL. 2nd ed. Singapore : McGraw-Hill Inc.

Premia Corporation. 1997. Codewright for Windows : User's Guide. January 1997 Edition.

Roll, Edward. 1955. TV for Radiomen. 1st ed. New York : Macmillan Company.

Sjoholm, Stefab and Lindh, Lennart. 1997. VHDL For Designers. 1st ed. Prentice Hall Europe.

Xilinx, Inc. 1997. Design Manager/Flow Engine /User Guide. 1997 Edition.

Xilinx, Inc. 1997. Development System User Guide. 1997 Edition.

Xilinx, Inc. 1997. Hardware Debugger Reference/User Guide. 1997 Edition.

ภาคผนวก

ภาคผนวก ก ขั้นตอนการพัฒนางานออกแบบสำหรับวงจรรวม FPGA

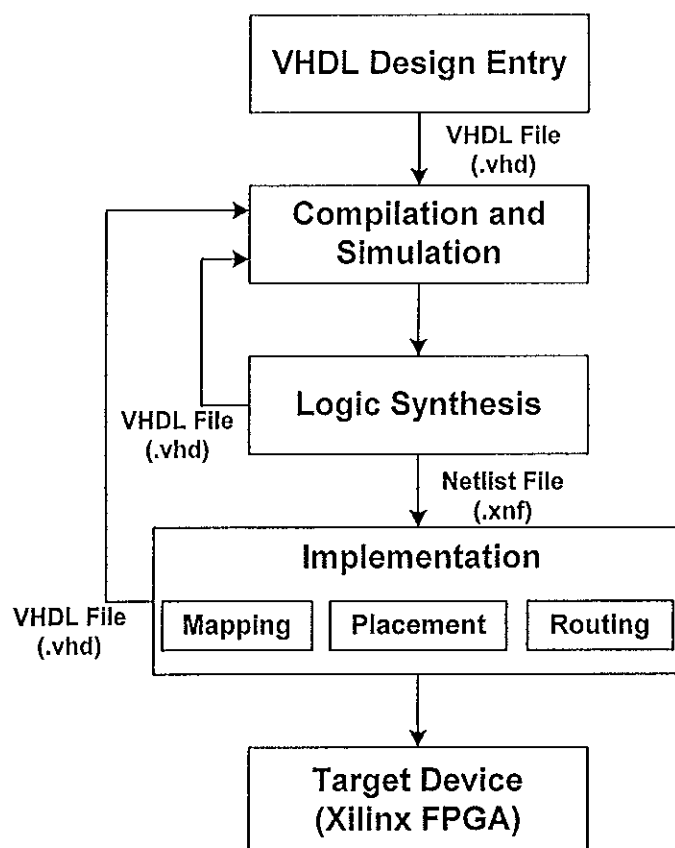
สำหรับการออกแบบวงจรรวม FPGA จะใช้ภาษาบรรยายฮาร์ดแวร์หรือภาษา VHDL (VHSIC Hardware Description Language) ซึ่งเป็นภาษาระดับสูงสำหรับการออกแบบวงจรรวมในระดับ Semi-custom โดยสามารถบรรยายการทำงานของดิจิทัลลอจิกเกตใดๆได้หลายลักษณะเช่นการบรรยายเชิงโครงสร้าง (Structural description) หรือการบรรยายเชิงพฤติกรรม (Behavioral description) สำหรับการออกแบบตัวควบคุมในงานวิจัยนี้เลือกลักษณะการบรรยายเชิงพฤติกรรมหรืออธิบายความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต โดยผู้ออกแบบไม่จำเป็นต้องรู้ถึงรายละเอียดของวงจรที่ออกแบบว่าประกอบด้วยลอจิกเกตชนิดใดเพียงแต่รู้หน้าที่การทำงานและความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตของวงจรที่ต้องการออกแบบ การเขียนโปรแกรมสำหรับวงจรรวมขนาดใหญ่ที่มีความซับซ้อนมักใช้หลักการออกแบบจากสูงสูดต่ำ (Top-down design) กล่าวคือ การออกแบบในลักษณะการแตกย่อยส่วนต่างๆออกเป็นลำดับขั้น (Hierarchy) จนได้รายละเอียดในระดับโมดูล (Module) ที่คิดว่าเหมาะสม จากนั้นจึงเขียนเป็นโปรแกรมแทนการทำงานของแต่ละโมดูลด้วยภาษา VHDL แล้วทำการคอมไพล์ (Compile) จำลองการทำงาน (Simulate) เพื่อตรวจสอบและยืนยันการทำงานว่าตรงตามที่ต้องการหรือไม่ ถ้าต้องมีการแก้ไขก็เพียงแค่แก้ที่รหัสต้นฉบับ (Source code) เท่านั้นแล้วทำขั้นตอนต่างๆซ้ำๆจนกว่าจะได้ผลการทำงานตามต้องการ หลังจากที่ได้โปรแกรมแทนแต่ละโมดูลตามต้องการแล้วก็ป็นขั้นตอนการนำโปรแกรมไปสังเคราะห์ (Synthesis) เพื่อให้ได้เป็นวงจรขึ้นมาจริงๆ โดยสรุปแล้วการพัฒนางานออกแบบวงจรรวม FPGA จะประกอบด้วย 4 ขั้นตอนหลักดังภาพประกอบ ก1 ได้ดังนี้

1. การป้อนแบบภาษา VHDL (VHDL Design Entry) ขั้นตอนนี้เริ่มจากการกำหนดรายละเอียดหน้าที่การทำงานของวงจรที่ต้องการ (Functional specification) แล้วจึงออกแบบในลักษณะ Top-down โดยแตกรายละเอียดทั้งหมดออกเป็นโมดูลย่อยๆ จากนั้นก็เขียนบรรยายแต่ละโมดูลด้วยภาษา VHDL จนครบทุกโมดูล

2. การคอมไพล์และจำลองการทำงาน (Compilation and Simulation) เป็นขั้นตอนการใช้ซอฟต์แวร์หรือตัวคอมไพเลอร์ (Compiler) ทำการตรวจสอบไวยากรณ์ของรหัสต้นฉบับภาษา VHDL ในแต่ละโมดูลและจำลองฟังก์ชันการทำงานด้วยซอฟต์แวร์สำหรับจำลองการทำงาน (Simulator) โดยสามารถสร้างชุดสัญญาณทดสอบ (Test vector) สำหรับทดสอบวงจรให้ครบทุกสถานะการทำงานเพื่อให้มั่นใจว่าผลลัพธ์ที่ได้เป็นไปตามที่กำหนดไว้

3. การสังเคราะห์ลอจิก (Logic Synthesis) ขั้นตอนนี้เป็นการใช้โปรแกรมในการสังเคราะห์จากโปรแกรมภาษา VHDL ไปเป็นวงจรจริงที่ประกอบด้วยอุปกรณ์ดิจิทัลพื้นฐาน เช่น เกต ฟลิปฟลอป เป็นต้น ซึ่งจะเป็นวงจรที่ทำงานได้ตามที่ได้เขียนบรรยายด้วยภาษา VHDL โดยสามารถทดสอบวงจรทั้งหมดที่ได้อีกครั้งด้วยชุดสัญญาณทดสอบเดิมเพื่อให้แน่ใจว่าวงจรยังคงทำงานให้ผลลัพธ์เป็นไปตามที่ต้องการ

4. การสร้างงานออกแบบ (Implementation) เป็นขั้นตอนในการนำวงจรที่สังเคราะห์ได้มาสร้างการทำงานจริงให้วงจรรวม FPGA ด้วยซอฟต์แวร์สำหรับ Implementation โดยซอฟต์แวร์จะทำการ Mapping วงจรแต่ละส่วนลงบนเซลล์ต่างๆภายในวงจรรวมแล้วทำการวาง (Placement) แต่ละส่วนลงในวงจรรวมเพื่อให้ผลลัพธ์ที่ดีที่สุดและทำการเชื่อมต่อ (Routing) ระหว่างเซลล์ที่เกี่ยวข้องเข้าด้วยกัน หลังจากนั้นก็สามารถดาวน์โหลดวงจรที่ได้ลงบนอุปกรณ์เป้าหมาย (Target device) ซึ่งก็คือชิพ FPGA เพื่อโปรแกรมการทำงาน (Configuration) ซึ่งสามารถนำชิพไปใช้งานตามต้องการได้



ภาพประกอบ ก1 แสดงขั้นตอนการพัฒนางานออกแบบวงจรรวม FPGA

นอกจากนี้เรายังสามารถยืนยันฟังก์ชันการทำงานเมื่อผ่านขั้นตอนที่ 3 และ 4 ได้อีกตาม
 ต้องการโดยกำหนดให้เอาต์พุตที่ได้หลังจากขั้นตอนทั้งสองเป็นไฟล์นามสกุล .vhd เพื่อให้สามารถ
 นำกลับไปทำขั้นตอนที่ 2 ได้

สำหรับซอฟต์แวร์ที่ใช้ในขั้นตอนต่างๆในงานวิจัยนี้แสดงในตาราง ก1 ดังนี้

ตาราง ก1 แสดงซอฟต์แวร์ที่ใช้ในขั้นตอนต่างๆในงานวิจัย

ขั้นตอน	ชื่อซอฟต์แวร์ที่ใช้	บริษัทผู้ผลิต
VHDL Design Entry	Codewright V.5.0c	Premia Corp.
Compilation and Simulation	ModelSim EE/Plus 5.2e beta-3	Model Technology
Logic Synthesis	Leonardo Spectrum Level3 v1998.2c	Exemplar Logic
Implementation	Xilinx Foundation F1.5	Xilinx

ภาคผนวก ข โปรแกรมรหัสต้นฉบับภาษา VHDL สำหรับการออกแบบตัวควบคุม

ลำดับการแสดงผลโปรแกรมรหัสต้นฉบับขอเริ่มจากส่วนโมดูลบนสุดลงไปยังส่วนโมดูลล่างสุดของงานออกแบบตามที่ได้แสดงในภาพประกอบ 4-16 ตามลำดับต่อไปนี้

1. ส่วนกำเนิดแอดเดรสให้หน่วยความจำแรม (RAMAdrCounter)
 - 1.1 ส่วนสร้างสัญญาณให้วงจรมับเริ่มการทำงาน (StartCntGen)
 - 1.2 ส่วนเลือกสัญญาณนาฬิกาให้ส่วนวงจรมับ (ClkSel)
 - 1.3 ส่วนวงจรมับ (Counter)
 - 1.3.1 ส่วนกำเนิดแอดเดรสทางแนวนอน (HorAdrCounter)
 - 1.3.2 ส่วนกำเนิดแอดเดรสทางแนวนอน (VerAdrCounter)และส่วน VResetSel
2. ส่วนการอินเตอร์เฟสกับหน่วยความจำแรม (RAM Interface)
3. ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP (EPPSigCntl)

```

-- ***** Controller Part ***** --
-- Function      : FPGA Controller for Image digitizer
-- Inputs        : Clk, Reset, RW, Vs, Cs, EPPWrite, EPPDataStrb and ADCData
-- Outputs       : RAMCE1, RAMCE2, RAMWE, RAMOE, EPPWait, TC, RAMAdr and PCData
-- Input/Output : RAMData

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity Controller is
  port (Clk,Reset,RW          : in    std_logic;
        Vs,Cs                : in    std_logic;
        EPPWrite,EPPDataStrb : in    std_logic;
        ADCData              : in    std_logic_vector(7 downto 0);
        RAMData              : inout  std_logic_vector(7 downto 0);
        RAMCE1,RAMCE2,EPPWait : out   std_logic;
        RAMWE,RAMOE,TC       : out   std_logic;
        RAMAdr               : out   std_logic_vector(15 downto 0);
        PCData               : out   std_logic_vector(7 downto 0));
end Controller;

architecture Struct of Controller is
  component RAMAdrCounter
    port (Clk,Reset,RW          : in    std_logic;
          EPPDataStrb,Vs,Cs    : in    std_logic;
          HAdr,VAdr           : out   std_logic_vector(7 downto 0);
          TC                   : out   std_logic);
  end component;

  component RAMInterface
    port (Clk,Reset,RW          : in    std_logic;
          HAdr,VAdr,ADCData    : in    std_logic_vector(7 downto 0);
          RAMData              : inout  std_logic_vector(7 downto 0);
          RAMCE1,RAMCE2       : out   std_logic;
          RAMWE,RAMOE         : out   std_logic;
          RAMAdr               : out   std_logic_vector(15 downto 0);
          PCData               : out   std_logic_vector(7 downto 0));
  end component;

  component EPPSigCntl
    port (EPPWrite,EPPDataStrb : in    std_logic;
          EPPWait              : out   std_logic);
  end component;

  signal hadr_sig  : std_logic_vector(7 downto 0);
  signal vadr_sig  : std_logic_vector(7 downto 0);

begin

  Unit1:RAMAdrCounter -- Main Part I
  port map( Clk      => Clk,
            Reset    => Reset,
            RW       => RW,
            EPPDataStrb => EPPDataStrb,
            Vs       => Vs,
            Cs       => Cs,
            HAdr     => hadr_sig,

```

```
VAdr    => vadr_sig,  
TC      => TC);
```

```
Unit2:RAMInterface -- Main Part II
```

```
port map( Clk      => Clk,  
          Reset    => Reset,  
          RW       => RW,  
          HAdr     => hadr_sig,  
          VAdr     => vadr_sig,  
          ADCData  => ADCData,  
          RAMData  => RAMData,  
          RAMCE1   => RAMCE1,  
          RAMCE2   => RAMCE2,  
          RAMWE    => RAMWE,  
          RAMOE    => RAMOE,  
          RAMAdr   => RAMAdr,  
          PCData   => PCData);
```

```
Unit3:EPPSigCntl -- Main Part III
```

```
port map( EPPWrite => EPPWrite,  
          EPPDataStrb => EPPDataStrb,  
          EPPWait    => EPPWait);
```

```
end Struct;
```

```
configuration CFG_CONTROLLER of CONTROLLER is  
for STRUCT  
end for;  
end CFG_CONTROLLER;
```

```
-- ***** I. RAMAdrCounter ***** --
-- Function   : Generating 16 bit Address to RAM and TC Signal for Monitoring Controller's Status in
--             Both Operation Modes
-- Inputs     : Clk, Reset, RW, EPPDatastrb, Vs and Cs
-- Outputs    : HAdr, VAdr and TC
```

```
Library IEEE;
use IEEE.Std_Logic_1164.all;
```

```
entity RAMAdrCounter is
  port ( Clk,Reset,RW      : in   std_logic;
         EPPDatastrb,Vs,Cs : in   std_logic;
         HAdr,VAdr        : out  std_logic_vector(7 downto 0);
         TC                : out  std_logic);
end RAMAdrCounter;
```

```
architecture Struct of RAMAdrCounter is
```

```
  component StartCntGen
    port (Reset,Clk,RW      : in   std_logic;
          Vs,Cs            : in   std_logic;
          Startcnt         : out  std_logic);
  end component;
```

```
  component ClkSel
    port (RW,Clk           : in   std_logic;
          EPPDataStrb     : in   std_logic;
          CClk            : out  std_logic);
  end component;
```

```
  component Counter
    port (CClk,Reset,Vs,RW : in   std_logic;
          StartCnt        : in   std_logic;
          HAdr,VAdr       : out  std_logic_vector(7 downto 0);
          TC               : out  std_logic);
  end component;
```

```
  signal startcnt_sig : std_logic;
  signal cclk_sig     : std_logic;
```

```
begin
```

```
  Unit1:StartCntGen
    port map( Reset   => Reset,
             Clk     => Clk,
             RW      => RW,
             Vs      => Vs,
             Cs      => Cs,
             Startcnt => startcnt_sig);
```

```
  Unit2:ClkSel
    port map(RW      => RW,
             Clk     => Clk,
             EPPDataStrb => EPPDataStrb,
             CClk    => cclk_sig);
```

```
Unit3:Counter
  port map( CClk    => cclk_sig,
           Reset   => Reset,
           Vs      => Vs,
           RW      => RW,
           StartCnt => startcnt_sig,
           HAdr    => HAdr,
           VAdr    => VAdr,
           TC      => TC);

end Struct;

configuration CFG_RAMAdrCounter of RAMAdrCounter is
  for STRUCT
  end for;
end CFG_RAMAdrCounter;
```

```

-- ***** I.I StartCntGen Part ***** --
-- Function   : Generating Startcnt Signal to Counter Part
-- Inputs     : Reset, Clk, RW, Vs and Cs
-- Output     : Startcnt

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity StartCntGen is
  port (Reset,Clk,RW      : in   std_logic;
        Vs,Cs           : in   std_logic;
        Startcnt        : out  std_logic);
end StartCntGen;

architecture Behav of StartCntGen is

  signal hs_sig      : std_logic;
  signal vstartdel_sig : std_logic;--vertical start delay signal
  signal vstartcnt_sig : std_logic;--vertical start counter signal
  signal hstartdel_sig : std_logic;--horizontal start delay signal
  signal wstartcnt_sig : std_logic;--write start count signal
  signal rstartcnt_sig : std_logic;--read start count signal
  signal vdel_sig     : integer range 0 to 87001;--vertical delay
  signal hdel_sig     : integer range 0 to 285;--horizontal delay

begin

  hs_sig <= not Cs; -- Converting Cs into Hs for Using in Counter

  --Generating rstartcnt Signal in Both Operation Modes
  proc1:process(Reset,RW)
  begin
    if (Reset='0') then
      rstartcnt_sig <='0';
    else
      if (RW='0') then
        rstartcnt_sig <='1';
      else
        rstartcnt_sig <='0';
      end if;
    end if;
  end process;

  --Generating vstartdel Signal after Occurring Rising Edge of Vs in Each Field
  proc2:process(Reset,RW,vdel_sig,Vs)
  begin
    if (Reset='0') then
      vstartdel_sig <='0';
    elsif (RW='0') then
      vstartdel_sig <='0';
    elsif (vdel_sig>87000) then
      vstartdel_sig <='0';
    elsif rising_edge(Vs) then
      if (RW='1') then
        vstartdel_sig <='1';
      end if;
    end if;
  end if;
end if;

```



```
end process;
```

```
--Generating vstartcnt Signal after Counting for 8000 Clks (delay about 1.73 ms)
```

```
proc3:process(vstartdel_sig,Clk)
variable vdelay : integer range 0 to 87001;
begin
if (vstartdel_sig='1') then
if rising_edge(Clk) then
if (vdelay<=87000) then
vdelay := vdel_sig+1;
vstartcnt_sig <='0';
if (vdelay>8000) then
vstartcnt_sig <='1';
end if;
else
vdelay :=0;
vstartcnt_sig <='0';
end if;
end if;
else
vdelay :=0;
vstartcnt_sig <='0';
end if;
vdel_sig <= vdelay;
end process;
```

```
--Generating hstartdel Signal after Occurring 32th Hs
```

```
proc4:process(vstartcnt_sig,hdel_sig,hs_sig)
begin
if (vstartcnt_sig='0') then
hstartdel_sig <='0';
elsif (hdel_sig=285) then
hstartdel_sig <='0';
elsif falling_edge(hs_sig) then
hstartdel_sig <='1';
end if;
end process;
```

```
--Generating hstartcnt Signal after Counting for 28 Clks (delay about 5.6 us)
```

```
proc5:process(hstartdel_sig,Clk)
variable hdelay : integer range 0 to 285;
begin
if (hstartdel_sig='1') then
if rising_edge(Clk) then
if (hdelay/=285) then
hdelay := hdelay+1;
wstartcnt_sig <='0';
if (hdelay>28) then
wstartcnt_sig <='1';
end if;
else
hdelay := 0;
wstartcnt_sig <='0';
end if;
end if;
end if;
else
```

```

        wstartcnt_sig <='0';
        hdelay := 0;
    end if;
    hdel_sig <= hdelay;
end process;

-- Generating startcnt Signal from wstartcnt xor rstartcnt
Startcnt <= wstartcnt_sig xor rstartcnt_sig;

end Behav;

-----

-- ***** I.II ClkSel Part ***** --
-- Function   : Selecting Counter Clk (CCLK) from Clk (Write Mode) and Data Strobe (Read Mode)
-- Inputs    : RW, Clk, and EPPDatastrb
-- Output    : CCLK

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity ClkSel is
    port (RW,Clk      : in   std_logic;
          EPPDatastrb : in   std_logic;
          CCLK       : out  std_logic);
end ClkSel;

architecture Behav of ClkSel is
begin

    process(RW,Clk,EPPDatastrb)
    begin
        if (RW='1') then          -- Write Mode
            CCLK <= Clk;          -- CCLK equals Clk
        else                       -- Read Mode
            CCLK <= not EPPDatastrb; -- CCLK equals inverted EPPDatastrb
        end if;
    end process;

end Behav;

-----

```

```

-- ***** I.III.I HorAdrCounter ***** --
-- Function   : Generating 8 bit Horizontal Address (HAdr) to RAM and up_enable Signal to
--              VerAdrCounter
-- Inputs     : Reset, CClk and Startcnt
-- Outputs    : HAdr and up_enable

```

```

Library IEEE;
use IEEE.Std_Logic_1164.all;

```

```

entity HorAdrCounter is
  port ( Reset,CClk   : in   std_logic;
        StartCnt     : in   std_logic;
        HAdr          : out  std_logic_vector(7 downto 0);
        up_enable     : out  std_logic);
end HorAdrCounter;

```

```

architecture Behav of HorAdrCounter is

```

```

  function increment(val:std_logic_vector) return std_logic_vector is
    -- normalize the indexing
    alias input : std_logic_vector(val'length downto 1) is val;
    variable result : std_logic_vector(input'range) := input;
    variable carry : std_logic := '1';
  begin
    for i in input'low to input'high loop
      result(i) := input(i) xor carry;
      carry := input(i) and carry;
      exit when carry = '0';
    end loop;
    return result;
  end increment;

```

```

begin

```

```

  -- Generating HAdr and up_enable after Occurring startcnt='1'and Rising Edge of CClk
  process(Reset,StartCnt,CClk)
    variable HAdr_out : std_logic_vector(7 downto 0);
    variable upen     : std_logic;
  begin
    if (Reset='0') then
      HAdr_out := (others=>'0');
      upen := '0';
    else
      if (Startcnt='1') then
        if rising_edge(CClk) then
          if (HAdr_out<"1111111") then
            HAdr_out := increment(HAdr_out);
            upen := '0';
            if (HAdr_out=="1111111") then
              upen :='1';
            end if;
          else
            HAdr_out := (others=>'0');
            upen := '0';
          end if;
        end if;
      end if;
    else

```

```

        HAdr_out := (others=>'0');
        upen := '0';
    end if;
end if;
HAdr    <= HAdr_out;
up_enable <= upen;
end process;

end Behav;

-----

-- ***** I.III.I VerAdrCounter ***** --
-- Function   : Generating 8 bit Vertical Address (VAdr) to RAM and TC Signal to Printer Port
-- Inputs     : Reset, RW, Vs and up_enable
-- Outputs    : VAdr and TC

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity VerAdrCounter is
    port ( Reset,RW,Vs : in    std_logic;
          up_enable   : in    std_logic;
          VAdr        : out   std_logic_vector(7 downto 0);
          TC          : out   std_logic);
end VerAdrCounter;

architecture Behav of VerAdrCounter is

    function increment(val:std_logic_vector) return std_logic_vector is
        -- normalize the indexing
        alias input : std_logic_vector(val'length downto 1) is val;
        variable result : std_logic_vector(input'range) := input;
        variable carry : std_logic := '1';
    begin
        for i in input'low to input'high loop
            result(i) := input(i) xor carry;
            carry := input(i) and carry;
            exit when carry = '0';
        end loop;
        return result;
    end increment;

    signal Vcount_sig : integer range 0 to 320;
    signal vreset_sig : std_logic;

begin

    -- VResetSel Part for Generating VReset signal
    VResetSel:process(RW,Vs)
    begin
        if (RW='1') then
            vreset_sig <= Vs ;-- VReset Equals Vs When Controller Operates in Write Mode
        else
            vreset_sig <= '1';-- VReset Equals '1' When Controller Operates in Read Mode
        end if;
    end process;
end Behav;

```

```

end if;
end process;

```

```

-- Generating VAdr after Occuring Rising Edge of up_enable

```

```

proc1: process(Reset,vreset_sig,up_enable)
variable VAdr_out : std_logic_vector(7 downto 0);
variable Vcount   : integer range 0 to 320;
begin
  if (Reset='0') then
    VAdr_out := (others=>'0');
    Vcount   := 0;
  else
    if (vreset_sig='1') then
      if falling_edge(up_enable) then
        if (Vcount/=320) then
          if (Vcount<255) then
            Vcount := Vcount+1;
            if (VAdr_out<"1111111") then
              VAdr_out := increment(VAdr_out);
            else
              VAdr_out := (others=>'0');
            end if;
          else
            Vcount := Vcount+1;
            VAdr_out := (others=>'0');
          end if;
        else
          Vcount :=0;
          VAdr_out := (others=>'0');
        end if;
      end if;
    else
      VAdr_out := (others=>'0');
      Vcount :=0;
    end if;
  end if;
  VAdr   <= VAdr_out;
  Vcount_sig <= Vcount;
end process;

```

```

-- Generating TC when vcount_sig>255

```

```

proc2: process(Vcount_sig)
variable termcnt : std_logic;
begin
  if (Vcount_sig>255) then
    termcnt := '1';
  else
    termcnt := '0';
  end if;
  TC<=termcnt;
end process;

```

```

end Behav;

```

```

-- ***** I.III Counter ***** --
-- Funtiong   : Generating HAdr (from HorAdrCounter), VAdr (from VerAdrCounter) to RAM
--             and TC to Printer Port
-- Inputs     : Reset, CClk, Vs, RW and StartCnt
-- Outputs    : HAdr, VAdr and TC

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity Counter is
  port ( Reset,CClk,Vs,RW : in   std_logic;
        StartCnt         : in   std_logic;
        HAdr,VAdr        : out  std_logic_vector(7 downto 0);
        TC               : out  std_logic);
end Counter;

architecture Struct of Counter is

  component HorAdrCounter
    port (Reset,CClk : in   std_logic;
          StartCnt  : in   std_logic;
          HAdr      : out  std_logic_vector(7 downto 0);
          up_enable  : out  std_logic);
  end component;

  component VerAdrCounter
    port (Reset,RW,Vs : in   std_logic;
          up_enable   : in   std_logic;
          VAdr        : out  std_logic_vector(7 downto 0);
          TC          : out  std_logic);
  end component;

  signal upen_sig : std_logic;

begin

  Unit1:HorAdrCounter
    port map( Reset    => Reset,
              CClk     => CClk,
              StartCnt => StartCnt,
              HAdr     => HAdr,
              up_enable => upen_sig);

  Unit2:VerAdrCounter
    port map( Reset    => Reset,
              RW       => RW,
              Vs       => Vs,
              up_enable => upen_sig,
              VAdr     => VAdr,
              TC       => TC);

end Struct;

```

```

-- ***** II. RAMInterface ***** --
-- Function : Interfacing RAM, i.e.
--           - Sending 8 bit Data from ADC (ADCData) in Write Mode
--           - Sending 16 bit Address (RAMAdr) from RAMAdrCounter and Control Signal
--             (RAMCE1, RAMCE2, RAMWE and RAMOE) in Both Operation
--           - Receiving 8 bit Data from RAM (RAMData) to Printer Port, Operated in EPP Mode
-- Inputs    : Clk, Reset, RW, HAdr, VAdr and ADCData
-- Outputs   : RAMCE1, RAMCE2, RAMWE, RAMOE, RAMAdr and PCData
-- Input/Output : RAMData

Library IEEE;
use IEEE.Std_Logic_1164.all;

entity RAMInterface is
port ( Clk,Reset,RW           : in          std_logic;
      HAdr,VAdr,ADCData      : in          std_logic_vector(7 downto 0);
      RAMData                : inout       std_logic_vector(7 downto 0);
      RAMCE1,RAMCE2         : out          std_logic;
      RAMWE,RAMOE           : out          std_logic;
      RAMAdr                 : out          std_logic_vector(15 downto 0);
      PCData                 : out          std_logic_vector(7 downto 0) );
end RAMInterface;

architecture Behav of RAMInterface is
signal ce_sig : std_logic;
begin

-- Generating RAMAdr from HAdr and VAdr
RAMAdr(7 downto 0)  <= HAdr;
RAMAdr(15 downto 8) <= VAdr;

-- Sending ADCData to RAMData when Controller Operates in Write Mode
WrProc:process(Reset,RW,Clk,ADCData)
begin
  RAMData <= (others=>'0');
  if (Reset='0') then
    RAMData <= (others=>'0');
  else
    if (RW='1') then
      RAMData <= ADCData ;
    else
      RAMData <= (others=>'Z');
    end if;
  end if;
end process;

-- Receiving RAMData to PCData when Controller Operates in Read Mode
PCData <=RAMData when (RW='0' and Reset='1') else (others=>'0');

-- Generating RAMCE1 and RAMCE2
ce_sig <= VAdr(7);
CEent:process(Reset,RW,ce_sig,Clk)
begin
  RAMCE1 <='0';
  RAMCE2 <='0';
  if (Reset='0') then
    RAMCE1 <='1';
  end if;
end process;

```

```

        RAMCE2 <='1';
    else
        if (RW='1') then -- Write Mode
            if (ce_sig='0') then
                RAMCE1 <=not Clk; -- RAM1 works
                RAMCE2 <='1';
            else
                RAMCE2 <=not clk; -- RAM2 works
                RAMCE1 <='1';
            end if;
        else -- Read Mode
            if (ce_sig='0') then
                RAMCE1 <='0'; -- RAM1 works
                RAMCE2 <='1';
            else
                RAMCE2 <='0'; -- RAM2 works
                RAMCE1 <='1';
            end if;
        end if;
    end if;
end process;

-- Generating RAMWE and RAMOE
WEOECnt:process(Reset,RW,Clk)
begin
    RAMWE <='0';
    RAMOE <='0';
    if (Reset='0') then
        RAMWE <='0';
        RAMOE <='0';
    else
        if (RW='1') then -- Write Mode
            RAMWE <= not clk;
            RAMOE <='1';
        else -- Read Mode
            RAMOE <='0';
            RAMWE <='1';
        end if;
    end if;
end process;

end Behav;

```

-- ***** III. EPPSigCntl Part ***** --

```

-- Function   : Controlling Wait Signal Generation to Acknowledge EPP-Mode Printer Port
-- Inputs    : EPPWrite and EPPDataStrb
-- Output    : EPPWait

```

```

Library IEEE;
use IEEE.Std_Logic_1164.all;

```

```

entity EPPSigCntl is
    port (EPPWrite,EPPDataStrb : in  std_logic;
          EPPWait              : out std_logic);

```



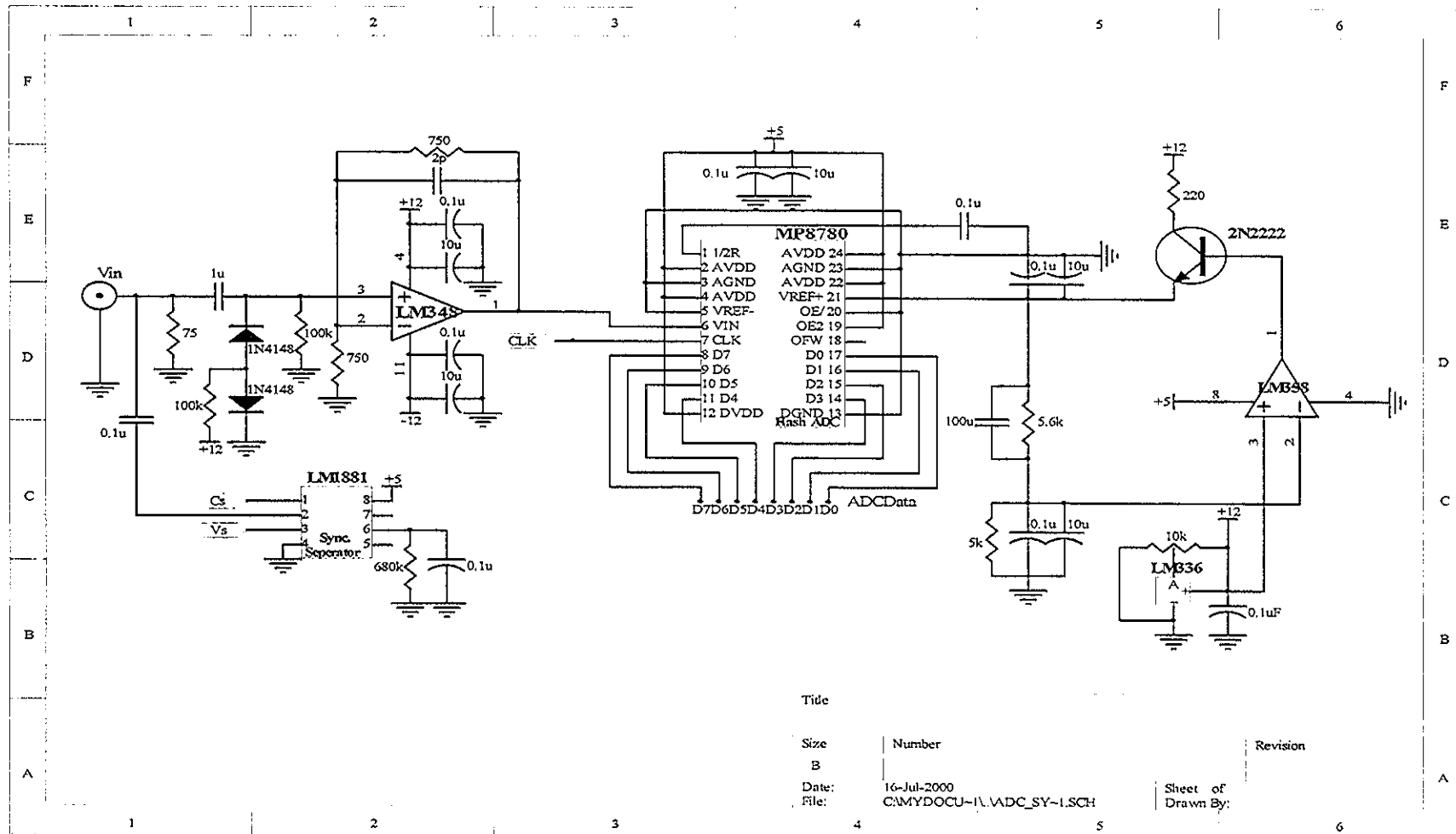
```
end EPPSigCntl;

architecture Behav of EPPSigCntl is
  signal eppwait_sig : std_logic;
begin

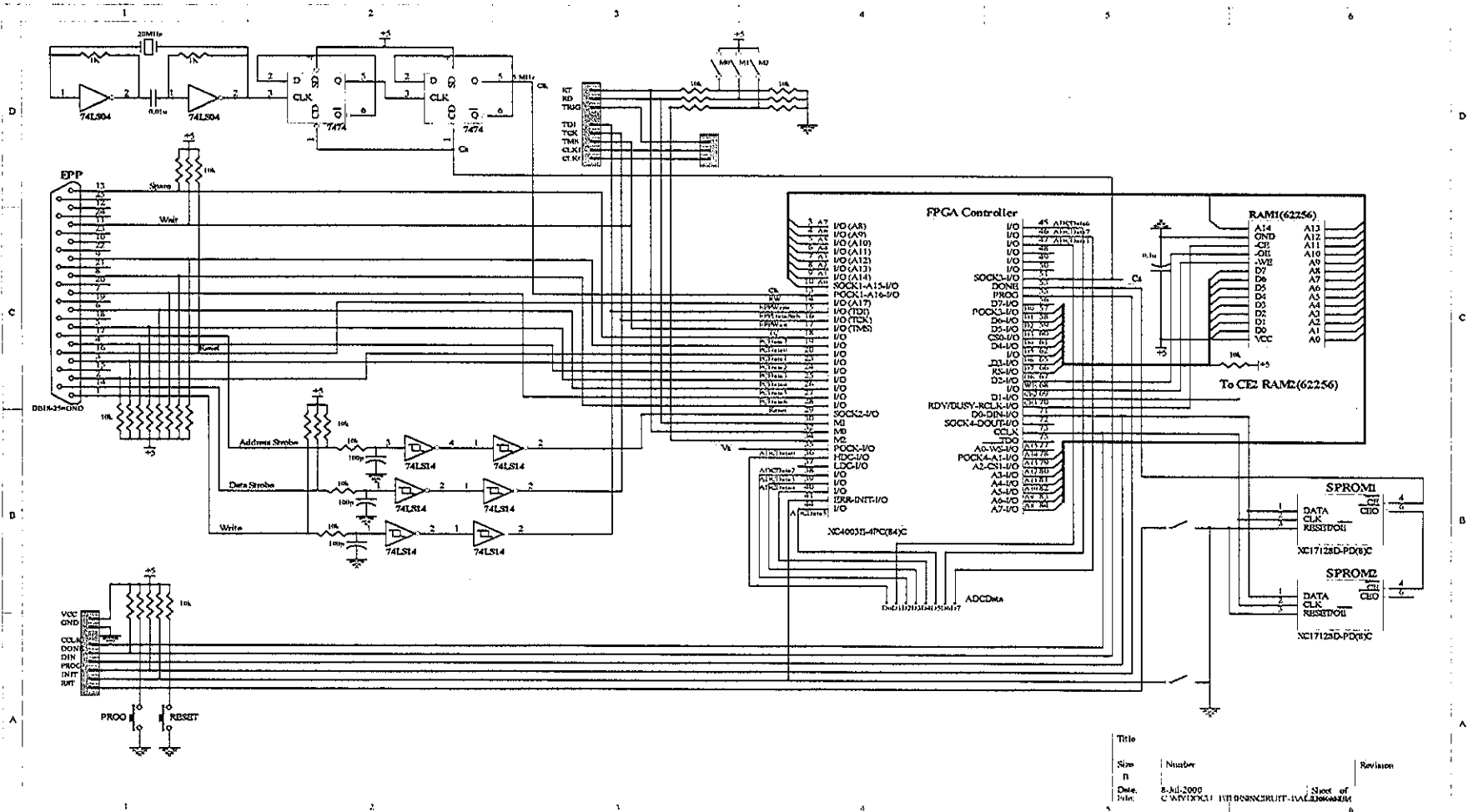
  eppwait_sig <= EPPWrite nand (not EPPDataStrb);
  EPPWait    <= not eppwait_sig;

end Behav;
```

ภาคผนวก ค Schematic diagram ของบอร์ด PCB1 และ PCB2



ภาพประกอบ ค1 แสดง Schematic diagram ของบอร์ด PCB1



ภาพประกอบ ค2 แสดง Schematic diagram ของบอร์ด PCB2

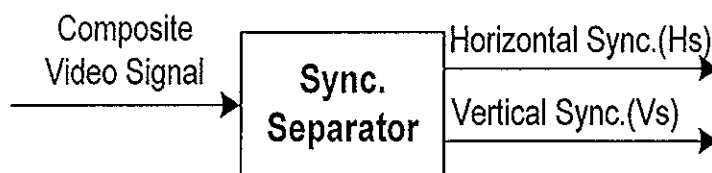
ภาคผนวก ง ผลการทดสอบและการวิเคราะห์ผลในส่วนวงจรหลักและส่วนหลักทั้งสาม ส่วนของตัวควบคุม

จากบทที่ 5 ได้แสดงผลการทดสอบเฉพาะในส่วนของสัญญาณที่ได้จากการทดสอบตัวควบคุมและภาพเก็บที่ได้จาก Image digitizer สำหรับผลการทดสอบการทำงานของส่วนวงจรหลักต่างๆภายใน Image digitizer และส่วนหลักทั้งสามภายในตัวควบคุมได้ยกมาแสดงในภาคผนวกนี้ดังนี้

ภาคผนวก ง1 ผลการทดสอบของวงจรหลักต่างๆ

1. วงจรแยกสัญญาณซิงก์

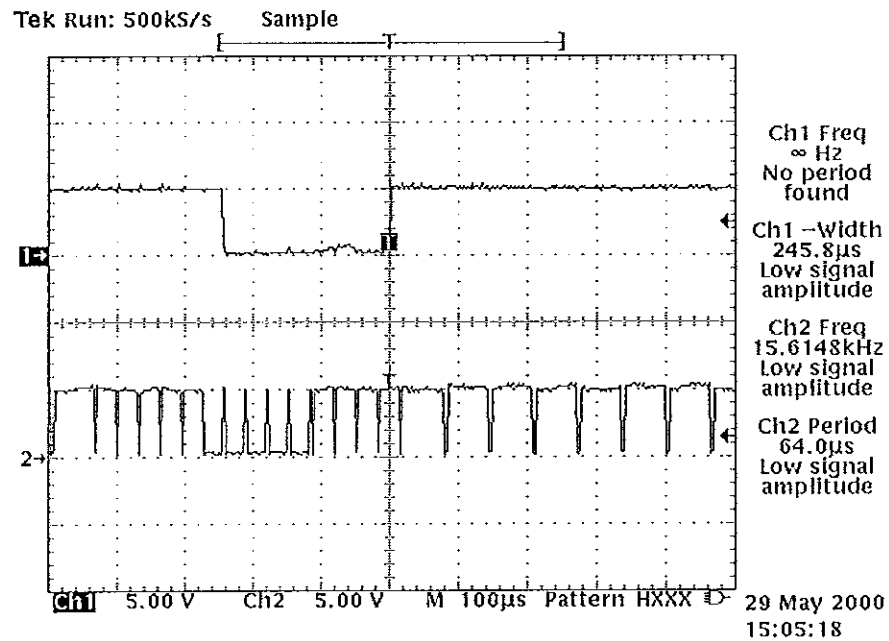
การทดสอบวงจรแยกสัญญาณซิงก์โดยการนำสัญญาณภาพรวมมาผ่านวงจรแยกสัญญาณซิงก์ซึ่งใช้วงจรรวมเบอร์ LM1881 ของบริษัท National Semiconductor สัญญาณเอาต์พุตที่ได้จะเป็นสัญญาณซิงก์ 2 สัญญาณคือสัญญาณ Vs และสัญญาณ Composite sync. ซึ่งจะประกอบด้วยสัญญาณ Vs สัญญาณฮอริซอนทัลซิงก์และสัญญาณ Hs ตามที่ต้องการดังภาพประกอบ ง1



ภาพประกอบ ง1 แสดงไดอะแกรมของวงจรแยกสัญญาณซิงก์

ผลที่ได้จากการทดสอบวงจรแยกสัญญาณซิงก์แสดงในภาพประกอบ ง2

- สัญญาณ Vs (Ch1) และ Hs (Ch2)



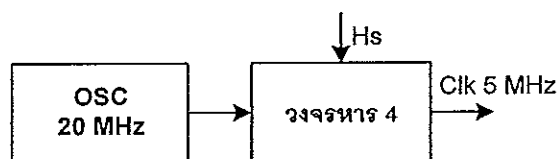
ภาพประกอบ ง2 แสดงสัญญาณ Vs และสัญญาณ Hs ที่ได้จากวงจรแยกสัญญาณซิงก์

ผลการวิเคราะห์

จากสัญญาณเอาต์พุตทั้งสองสัญญาณที่ได้พบว่าให้ผลของสัญญาณที่ถูกต้องกล่าวคือ สัญญาณ Vs จะมีช่วงเวลาที่สัญญาณพัลส์มีค่าเป็นลอจิก '0' ประมาณ 245.8 ไมโครวินาที ส่วนสัญญาณ Cs ซึ่งประกอบด้วยสัญญาณ Hs ที่ต้องการนั้นจะมีช่วงเวลาที่สัญญาณพัลส์มีค่าเป็นลอจิก '0' ประมาณ 4.78 ไมโครวินาที มีคาบเวลาเท่ากับ 64 ไมโครวินาทีและมีความถี่ประมาณ 15.6148 กิโลเฮิร์ตซ์ซึ่งใกล้เคียงกับความถี่จริงคือ 15.625 กิโลเฮิร์ตซ์

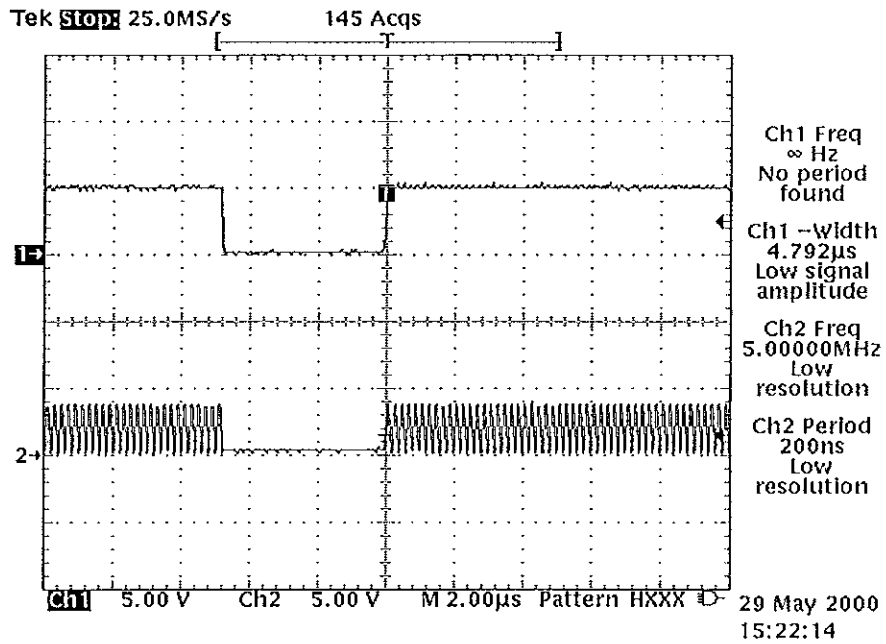
2. วงจรกำเนิดสัญญาณนาฬิกา

การทดสอบวงจรกำเนิดสัญญาณนาฬิกาเพื่อสร้างสัญญาณนาฬิกา (Clk) ความถี่ 5 เมกกะเฮิร์ตซ์โดยการนำคริสตอลความถี่ 20 เมกกะเฮิร์ตซ์จากนั้นนำมาผ่านวงจรหารสี่ซึ่งถูกควบคุมด้วยสัญญาณ Hs ที่ได้จากวงจรแยกสัญญาณซิงก์ดังไดอะแกรมในภาพประกอบ ง3 และผลการทดสอบที่ได้ในภาพประกอบ ง4



ภาพประกอบ ง3 แสดงไดอะแกรมของวงจรกำเนิดสัญญาณนาฬิกาความถี่ 5 เมกกะเฮิร์ตซ์

- สัญญาณ Hs (Ch1) และสัญญาณ Clk ความถี่ 5 เมกกะเฮิร์ตซ์ (Ch2)



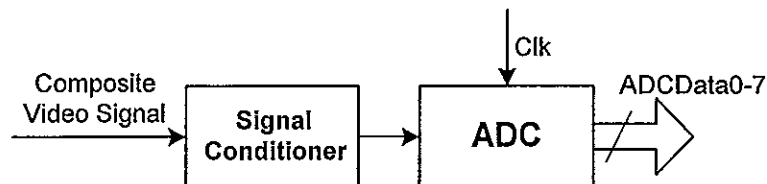
ภาพประกอบ ง4 แสดงสัญญาณ Hs และสัญญาณ Clk จากวงจรกำเนิดสัญญาณนาฬิกา

ผลการวิเคราะห์

พบว่าสัญญาณ Clk ที่ได้จะเกิดขึ้นและสิ้นสุดทุกครั้งในช่วงของสัญญาณ Hs ในแต่ละลูกและมีความถี่เท่ากับ 5 เมกกะเฮิร์ตซ์หรือมีคาบเวลาเท่ากับ 200 นาโนวินาทีตามที่ออกแบบไว้

3. วงจรแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล (ADC)

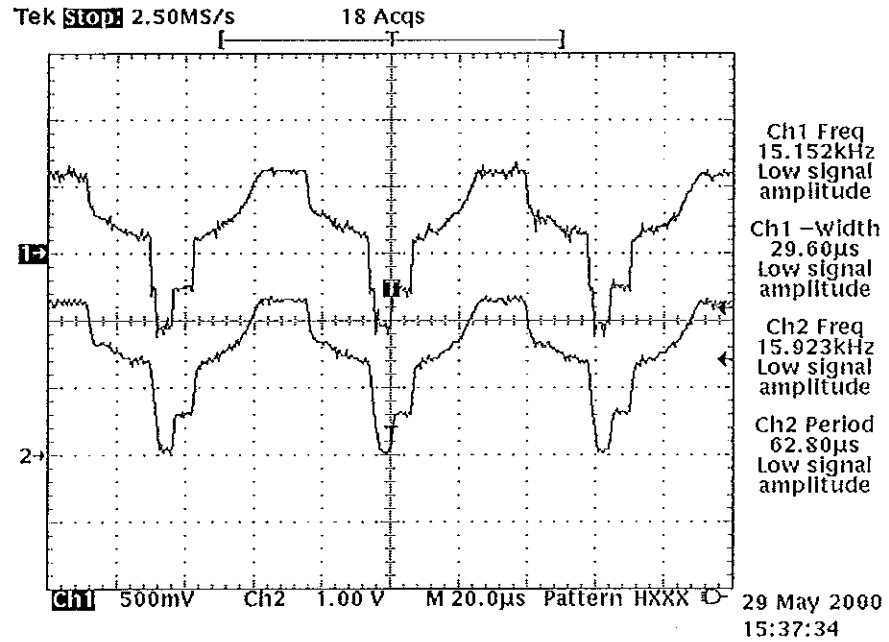
วงจร ADC ที่ทดสอบประกอบด้วย 2 ส่วน โดยส่วนแรกเป็นส่วนที่เรียกว่า Signal conditioner ทำหน้าที่ปรับขนาดแรงดันของสัญญาณภาพรวมให้เพิ่มขึ้นจากเดิม 2 เท่าเพื่อเป็นสัญญาณ Vin ให้วงจร Flash ADC ซึ่งเป็นส่วนที่สอง โดยวงจร Flash ADC นี้จะถูกกำหนดอัตราในการแปลงสัญญาณด้วยสัญญาณ Clk ที่ได้จากวงจรกำเนิดสัญญาณนาฬิกาและเอาต์พุตที่ได้จะเป็นข้อมูลดิจิทัลขนาด 8 บิต (ADCData0-7) ดังไดอะแกรมในภาพประกอบ ง5



ภาพประกอบ ง5 แสดงไดอะแกรมของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

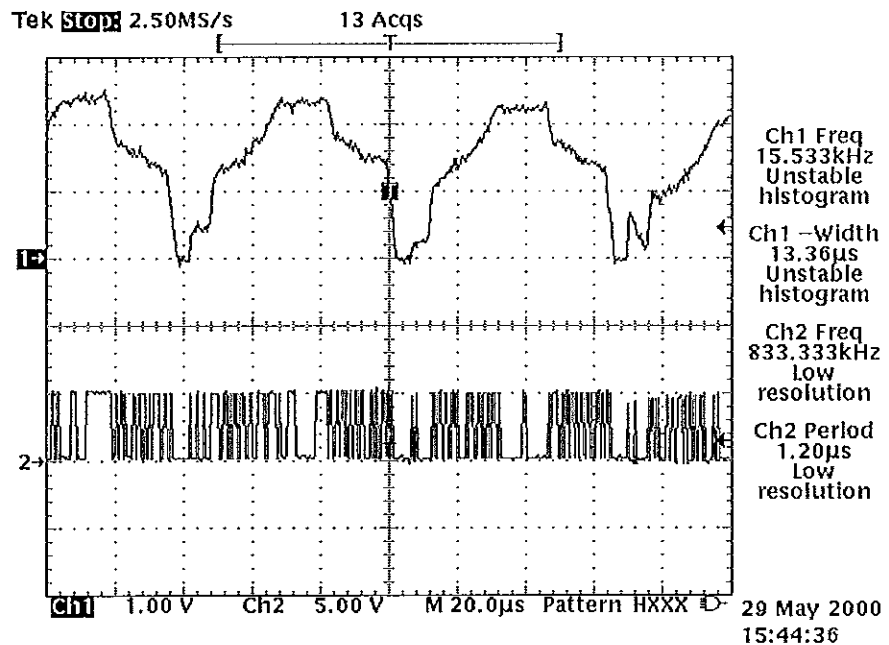
ผลการทดสอบที่ได้มีดังนี้

- สัญญาณภาพรวม (Ch1) และสัญญาณ Vin ของวงจร ADC (Ch2)



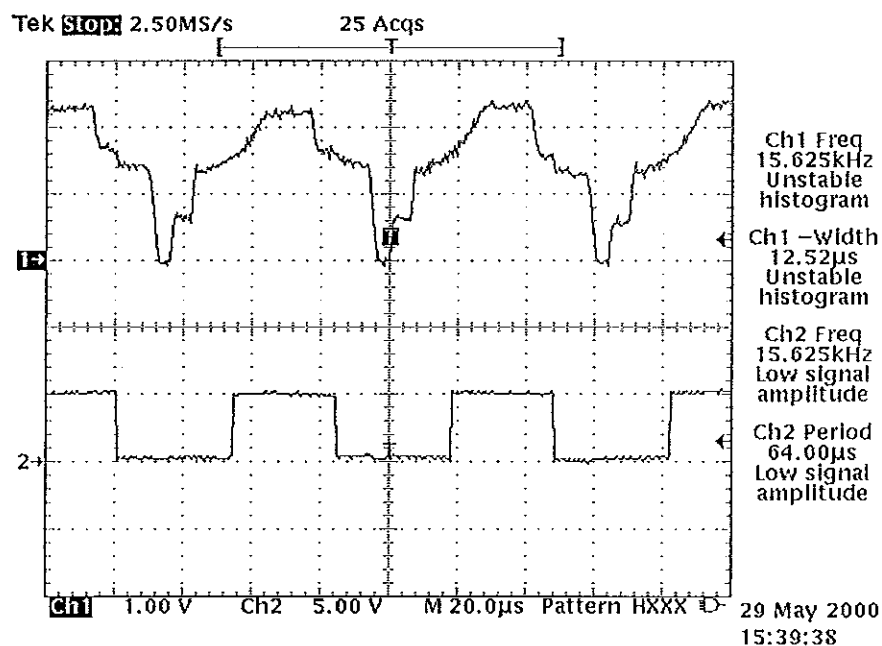
ภาพประกอบ ง6 แสดงสัญญาณภาพรวมและสัญญาณ Vin ของวงจร ADC

- สัญญาณ Vin (Ch1) กับสัญญาณ ADCData บิตที่ 1 (ADCData0) (Ch2)



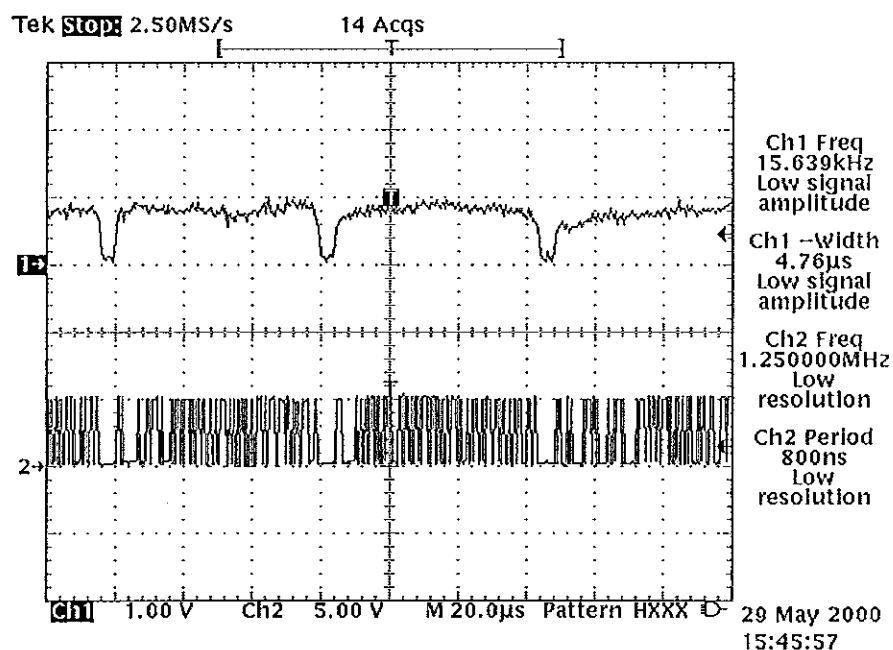
ภาพประกอบ ง7 แสดงสัญญาณ Vin กับสัญญาณ ADCData0 ของวงจร ADC

- สัญญาณ Vin (Ch1) กับสัญญาณ ADCData บิตที่ 8 (ADCData7) (Ch2)



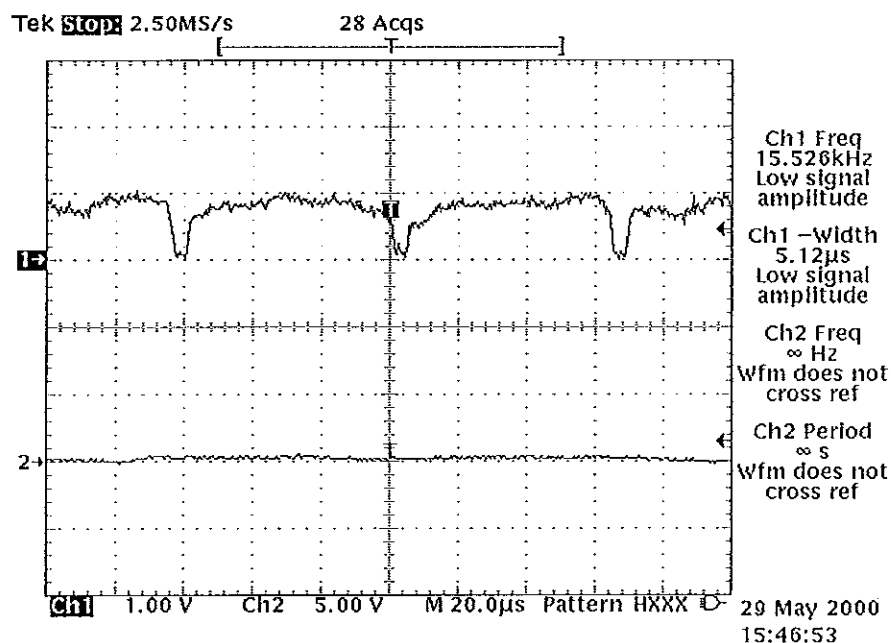
ภาพประกอบ ง8 แสดงสัญญาณ Vin กับสัญญาณ ADCData7 ของวงจร ADC

- สัญญาณ Vin เมื่อเป็นภาพดำนว (Ch1) กับสัญญาณ ADCData0 (Ch2)



ภาพประกอบ ง9 แสดงสัญญาณ Vin เมื่อเป็นภาพดำล้วนกับสัญญาณ ADCData0

- สัญญาณ Vin เมื่อเป็นภาพดำล้วน (Ch1) กับสัญญาณ ADCData7 (Ch2)



ภาพประกอบ ง10 แสดงสัญญาณ Vin เมื่อเป็นภาพดำล้วนกับสัญญาณ ADCData7

ผลการวิเคราะห์

- จากภาพประกอบ ง6 สัญญาณ Vin หรือสัญญาณอินพุตของวงจร Flash ADC ซึ่งผ่าน ส่วน Signal conditioner จะถูกปรับขนาดแรงดันจากเดิมมีค่าเป็น -0.52 โวลต์ถึง 0.5 โวลต์ให้มีขนาดแรงดันเป็น 0.2 โวลต์ถึง 2.3 โวลต์หรือแรงดันถูกขยายให้มีขนาดเพิ่มจากเดิมเป็น 2 เท่า

- สำหรับภาพประกอบ ง7 ถึง ง10 เป็นตัวอย่างของผลที่ได้จากการแปลงสัญญาณภาพ ให้เป็นสัญญาณดิจิทัลซึ่งเลือกแสดงเฉพาะสัญญาณ ADCData บิตที่ 1 (ADCData0) และ สัญญาณ ADCData บิตที่ 8 (ADCData7) โดยพบว่าเมื่อเป็นภาพปกติหรือภาพดำล้วนสัญญาณ ADCData0 ให้ผลการแปลงตลอดเวลา แต่สำหรับสัญญาณ ADCData7 จะให้ผลการแปลงเป็นลอจิก '0' เมื่อเป็นภาพดำล้วนและให้ผลการแปลงเป็นลอจิก '1' เมื่อภาพมีบางส่วนที่เป็นสีขาว

4. การสร้างสัญญาณจากพอร์ตขนานในโหมดการทำงานแบบ EPP

การทดสอบการสร้างสัญญาณจากพอร์ตขนานในโหมดการทำงานแบบ EPP เพื่อใช้ใน Image digitizer ซึ่งมีสัญญาณทั้งหมด 3 สัญญาณ สามารถทำได้ดังนี้

4.1 สัญญาณ Reset เป็นขาสัญญาณที่ 16 หรือบิตที่ 3 ในรีจิสเตอร์ควบคุม (Base address+2) ของพอร์ตขนานใช้เป็นสัญญาณ RW (Read/Write) เพื่อกำหนดโหมดการทำงาน ของ Image digitizer โดยเมื่อต้องการให้มีค่าเป็น

- ลอจิก '1' เป็นการสั่งให้ Image digitizer ทำงานในโหมดการเขียนข้อมูลภาพ โดยใช้คำสั่ง outportb() เขียนค่า 0xC4 ไปที่รีจิสเตอร์ควบคุม

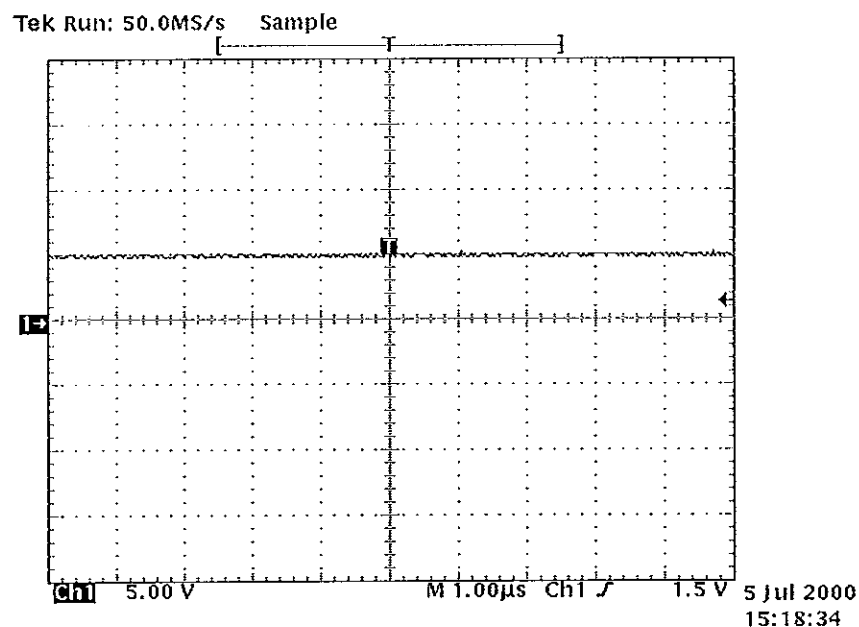
- ลอจิก '0' เป็นการสั่งให้ Image digitizer ทำงานในโหมดการอ่านข้อมูลภาพ โดยใช้คำสั่ง outportb() เขียนค่า 0xC0 ไปที่รีจิสเตอร์ควบคุม

4.2 สัญญาณ Address strobe เป็นขาสัญญาณที่ 17 ของพอร์ตขนานใช้เป็นสัญญาณ Reset สำหรับรีเซ็ตการทำงานของ Image digitizer สามารถสั่งให้ทำงาน (สัญญาณแอกทีฟที่ลอจิก '0') โดยใช้คำสั่ง inportb() เขียนไปที่รีจิสเตอร์แอดเดรส (Base address+3) ในการทดสอบจะใช้คำสั่งวนรูปแบบไม่รู้จบเพื่อสร้างสัญญาณอย่างต่อเนื่อง

4.3 สัญญาณ Data strobe เป็นขาสัญญาณที่ 16 ของพอร์ตขนานใช้เป็นสัญญาณ CClk ให้ส่วน RAMAdrCounter ในโหมดการอ่านข้อมูลภาพ สามารถสั่งให้ทำงาน (สัญญาณแอกทีฟที่ลอจิก '0') โดยใช้คำสั่ง inportb() เขียนไปที่รีจิสเตอร์ข้อมูล (Base address+4) สำหรับผลการทดสอบขอยกไปกล่าวอีกครั้งในหัวข้อผลการทดสอบตัวควบคุมในส่วนของ EPPSigCntl

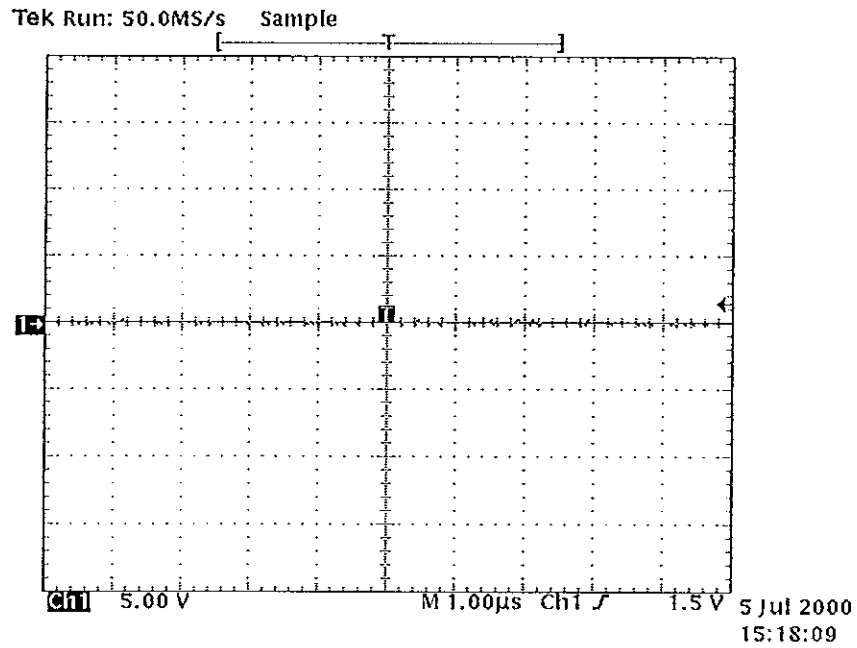
โดยผลของสัญญาณต่างๆมีดังนี้

- สัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW เมื่ออยู่ในโหมดการเขียนข้อมูลภาพ



ภาพประกอบ ง11 แสดงสัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW ของ Image digitizer ในโหมดการเขียน

- สัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW เมื่ออยู่ในโหมดการอ่าน

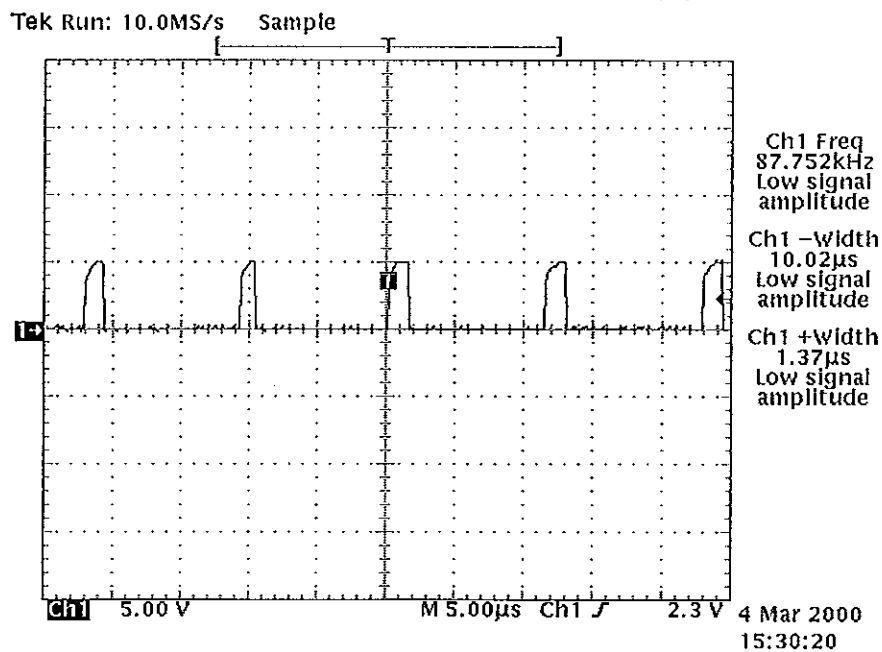


ภาพประกอบ ง12 แสดงสัญญาณ Reset ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ RW ของ Image digitizer ในโหมดการอ่าน

ผลการวิเคราะห์

จากภาพประกอบ ง11 และ ง12 ให้ผลการสร้างสัญญาณ RW ในโหมดการเขียนและการอ่านเป็นลอจิก '1' และลอจิก '0' ตามลำดับซึ่งตรงตามที่ต้องการ

- สัญญาณ Address strobe ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ Reset



ภาพประกอบ ง13 แสดงสัญญาณ Address strobe ของพอร์ตขนานซึ่งใช้เป็นสัญญาณ Reset

ผลการวิเคราะห์

จากการทดสอบพบว่าสัญญาณ Address strobe มีช่วงเวลาที่สัญญาณพัลส์เป็นลอจิก '0' เท่ากับ 10 ไมโครวินาที ดังนั้นสัญญาณ Reset ที่ใช้ใน Image digitizer ก็จะมีช่วงเวลาที่สัญญาณพัลส์เป็นลอจิก '0' เท่ากับ 10 ไมโครวินาทีด้วยเช่นกัน

ภาคผนวก ง2 ผลการทดสอบของส่วนหลักทั้งสามส่วนของตัวควบคุม

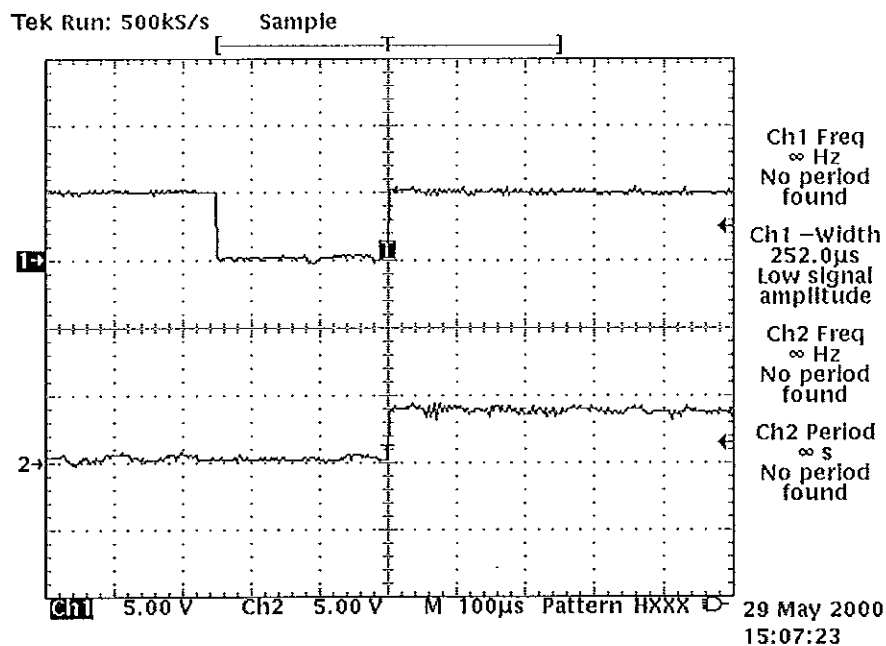
การทดสอบตัวควบคุมจะทดสอบการทำงานทั้งในโหมดการเขียนและโหมดการอ่านข้อมูลภาพ โดยแยกตามจำนวนของส่วนหลักทั้งสามตามที่ยกแบบไว้ดังต่อไปนี้

1. ส่วนกำเนิดแอดเดรสสำหรับหน่วยความจำแรม (RAM Address Counter, RAM-AdrCounter) โดยแบ่งการทดสอบตามจำนวนส่วนย่อยทั้ง 3 ส่วนคือ

1.1 ส่วนสร้างสัญญาณให้วงจรนับเริ่มการทำงาน (Start Count Generator, StartCntGen)

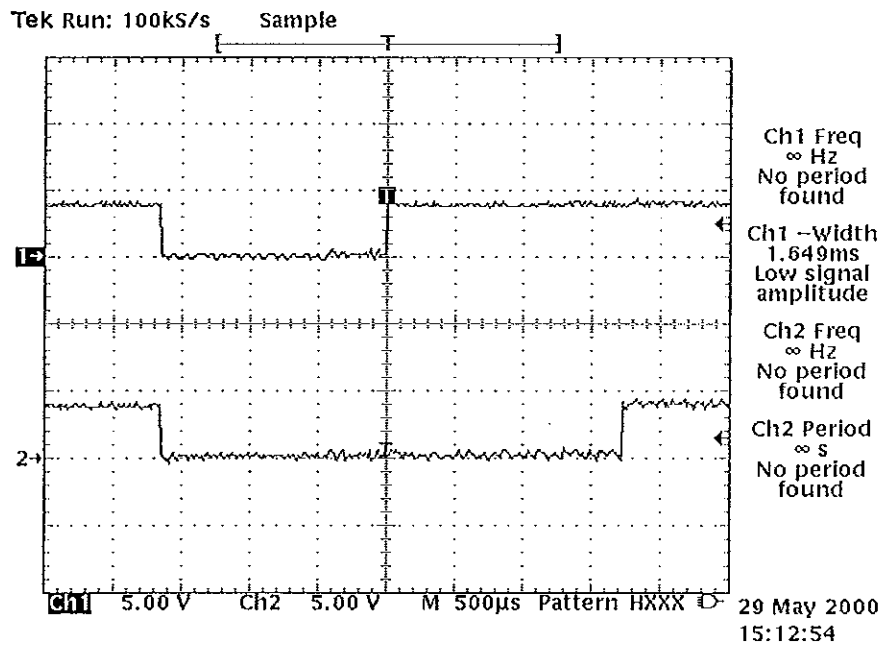
1.1.1 โหมดการเขียนข้อมูลภาพ

- สัญญาณ Vs (Ch1) กับสัญญาณ vstartdel (Ch2)



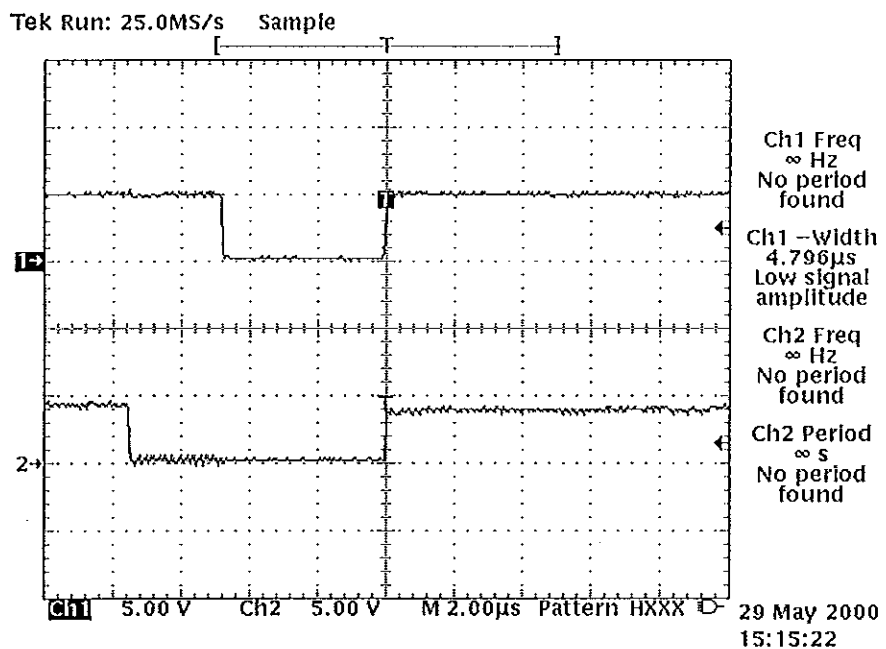
ภาพประกอบ ง14 แสดงสัญญาณ Vs และสัญญาณ vstartdel เมื่ออยู่ในโหมดการเขียน

- สัญญาณ vstartdel (Ch1) กับสัญญาณ vstartcnt (Ch2)



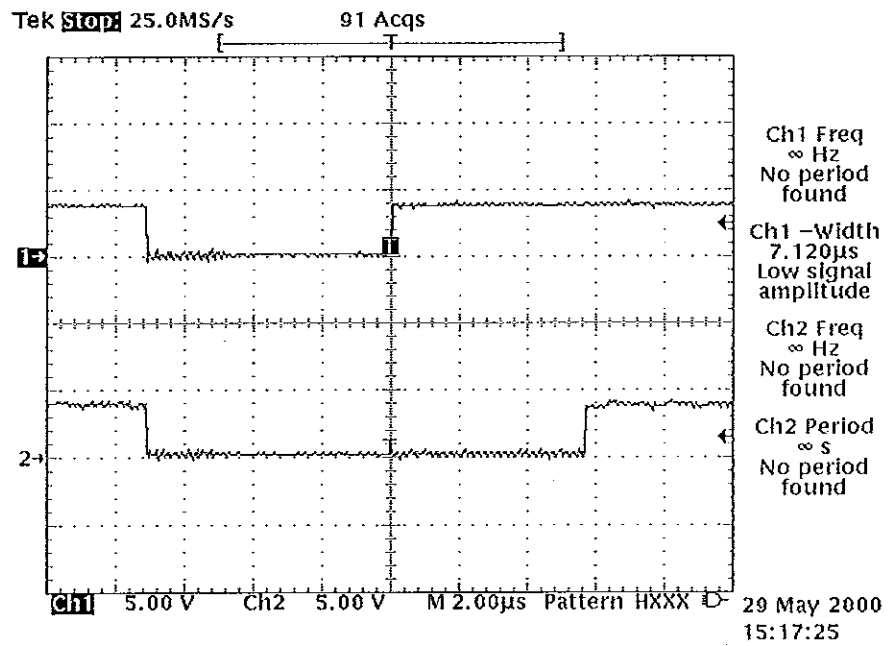
ภาพประกอบ ง15 แสดงสัญญาณ vstartdel กับสัญญาณ vstartcnt เมื่ออยู่ในโหมดการเขียน

- สัญญาณ Hs (Ch1) และสัญญาณ hstartdel (Ch2)



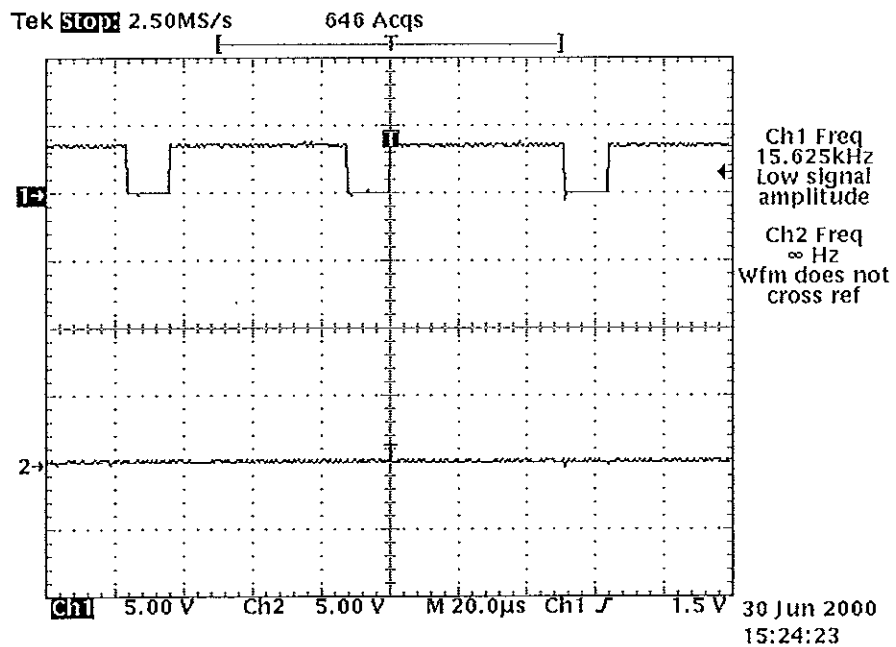
ภาพประกอบ ง16 แสดงสัญญาณ Hs และสัญญาณ hstartdel เมื่ออยู่ในโหมดการเขียน

- สัญญาณ hstartdel (Ch1) และสัญญาณ wstartcnt (Ch2)



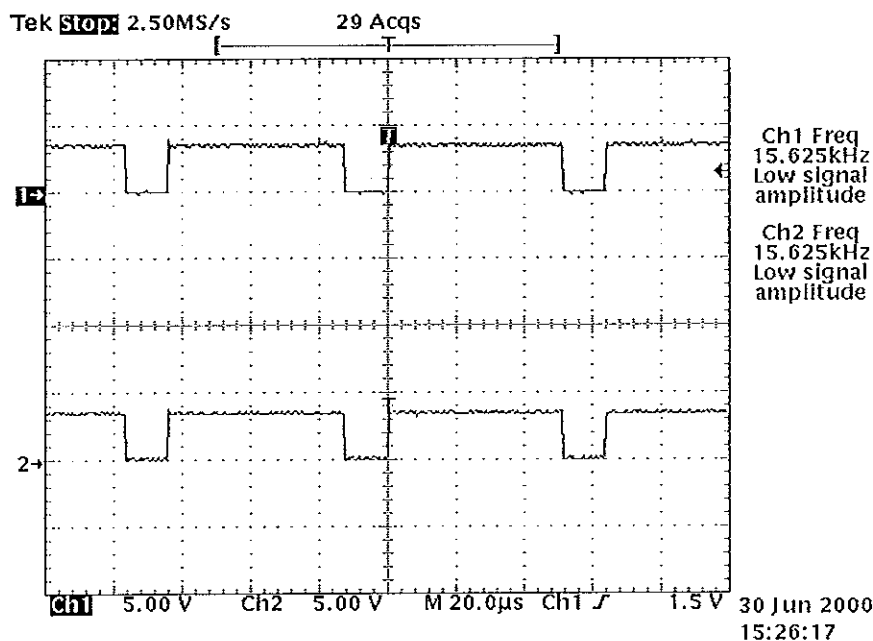
ภาพประกอบ ง17 แสดงสัญญาณ hstartdel และสัญญาณ wstartcnt เมื่ออยู่ในโหมดการเขียน

- สัญญาณ wstartcnt (Ch1) และสัญญาณ rstartcnt (Ch2)



ภาพประกอบ ง18 แสดงสัญญาณ wstartcnt และสัญญาณ rstartcnt เมื่ออยู่ในโหมดการเขียน

- สัญญาณ wstartcnt (Ch1) และสัญญาณ startcnt (Ch2)



ภาพประกอบ ง19 แสดงสัญญาณ wstartcnt และสัญญาณ startcnt เมื่ออยู่ในโหมดการเขียน

ผลการวิเคราะห์

ผลการทดสอบของส่วน StartCntGen เมื่ออยู่ในโหมดการเขียนข้อมูลภาพให้ผลดังนี้

- จากภาพประกอบ ง14 ถึง ง15 แสดงการเกิดของสัญญาณ vstartdel และสัญญาณ vstartcnt พบว่าเมื่อมีสัญญาณ Vs เข้ามาจะเกิดสัญญาณ vstartdel ที่ขอบขาขึ้นของสัญญาณ Vs และเกิดการหน่วงเวลาไปประมาณ 1.73 มิลลิวินาทีหรือผ่านสัญญาณ Hs ไป 31 ลูกจึงเกิดสัญญาณ vstartcnt เพื่อรอสัญญาณ Hs ลูกที่ 32 ต่อไปตามที่ออกแบบไว้

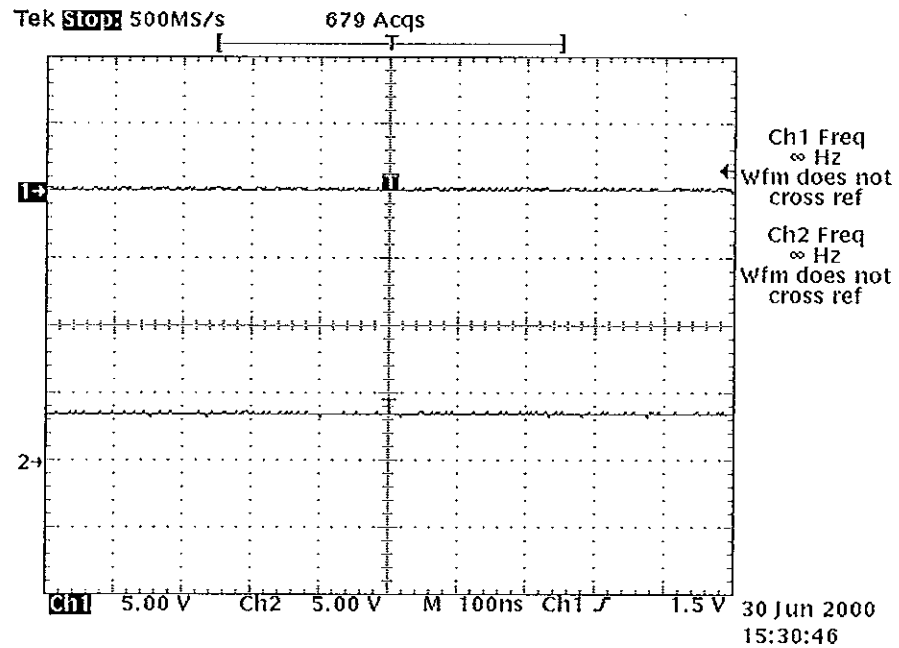
- จากภาพประกอบ ง16 ถึง ง17 แสดงการเกิดของสัญญาณ hstartdel และสัญญาณ wstartcnt พบว่าเมื่อเกิดสัญญาณ vstartcnt ขึ้นและมีสัญญาณ Hs เข้ามาจะเกิดสัญญาณ hstartdel ตามขอบขาขึ้นของสัญญาณ Hs และเกิดการหน่วงเวลาไปประมาณ 5.6 ไมโครวินาทีจึงเกิดสัญญาณ wstartcnt ตามที่ออกแบบไว้

- จากภาพประกอบ ง18 พบว่าสัญญาณ wstartcnt จะเกิดขึ้นตามสัญญาณ Vs และสัญญาณ Hs ที่เข้ามาในขณะที่สัญญาณ rstartcnt จะมีค่าเป็นลอจิก '0' ตลอดเวลาซึ่งเมื่อทั้งสองสัญญาณผ่านเกตเอ็กซ์คลูซีฟออร์ทำให้ได้สัญญาณ startcnt ที่มีลักษณะเหมือนสัญญาณ wstartcnt ดังในภาพประกอบ ง19

1.1.2 โหมตการอ่านข้อมูลภาพ

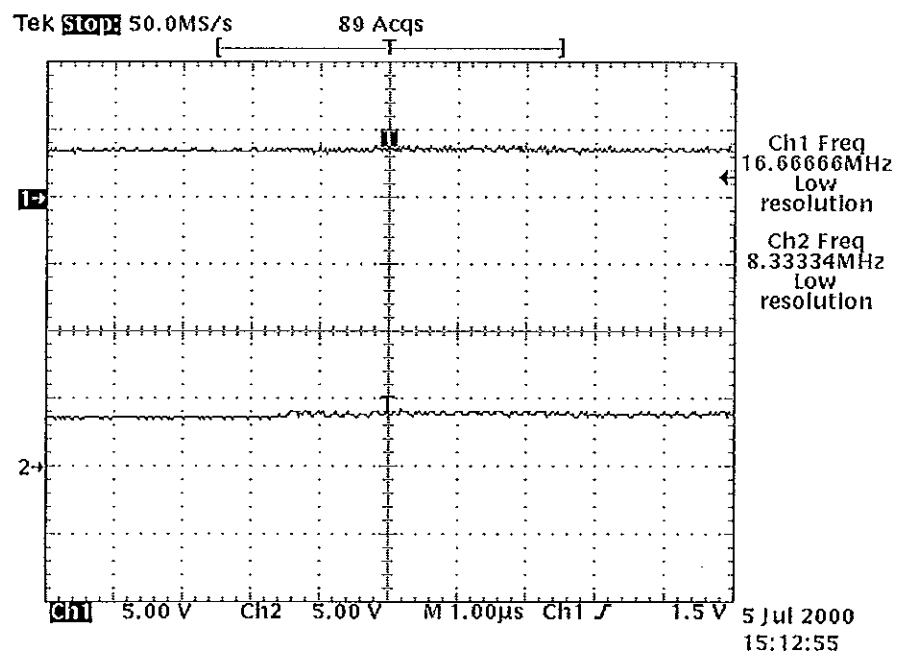
ในโหมตการอ่านนี้สัญญาณ wstartcnt จะมีค่าเป็นลอจิก '0' ส่วนสัญญาณ rstartcnt จะมีค่าเป็นลอจิก '1' ดังผลต่อไปนี้

- สัญญาณ wstartcnt (Ch1) และสัญญาณ rstartcnt (Ch2)



ภาพประกอบ ง20 แสดงสัญญาณ wstartcnt และสัญญาณ rstartcnt เมื่ออยู่ในโหมตการอ่าน

- สัญญาณ rstartcnt (Ch1) และสัญญาณ startcnt (Ch2)



ภาพประกอบ ง21 แสดงสัญญาณ wstartcnt และสัญญาณ startcnt เมื่ออยู่ในโหมตการอ่าน

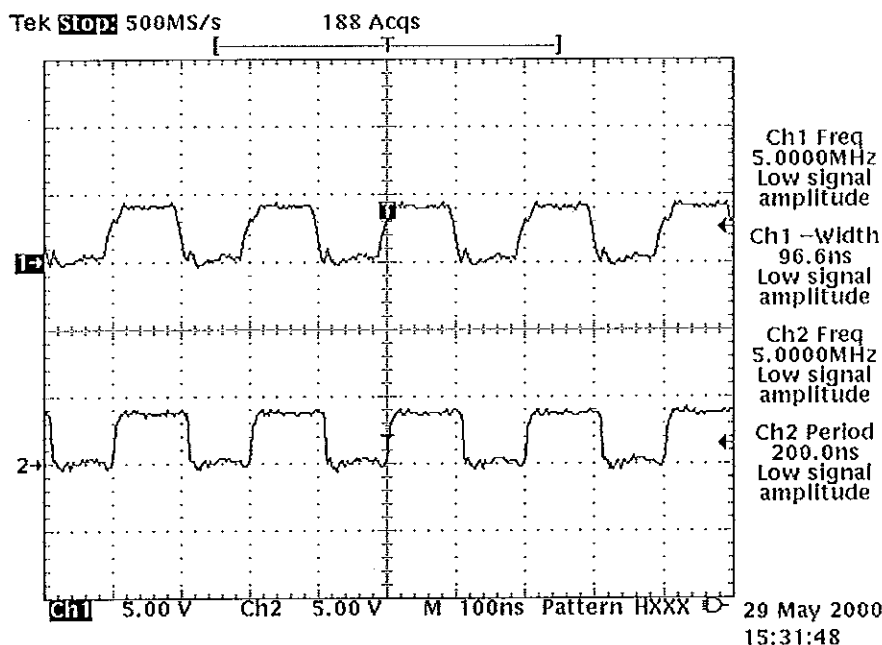
ผลการวิเคราะห์

ผลการทดสอบของส่วน StartCntGen เมื่ออยู่ในโหมดการอ่านข้อมูลภาพคือจากภาพประกอบ ง20 พบว่าสัญญาณ wstartcnt จะมีค่าเป็นลอจิก '0' ขณะที่สัญญาณ rstartcnt จะมีค่าเป็นลอจิก '1' จริงซึ่งเมื่อทั้งสองสัญญาณผ่านเกตเอ็กซ์คลูซีฟออร์ทำให้ได้สัญญาณ startcnt ที่มีค่าเป็นลอจิก '1' เหมือนสัญญาณ rstartcnt ดังภาพประกอบ ง21

1.2 ส่วนเลือกสัญญาณนาฬิกาให้ส่วนวงจรรีบ (Clock Selector, ClkSel)

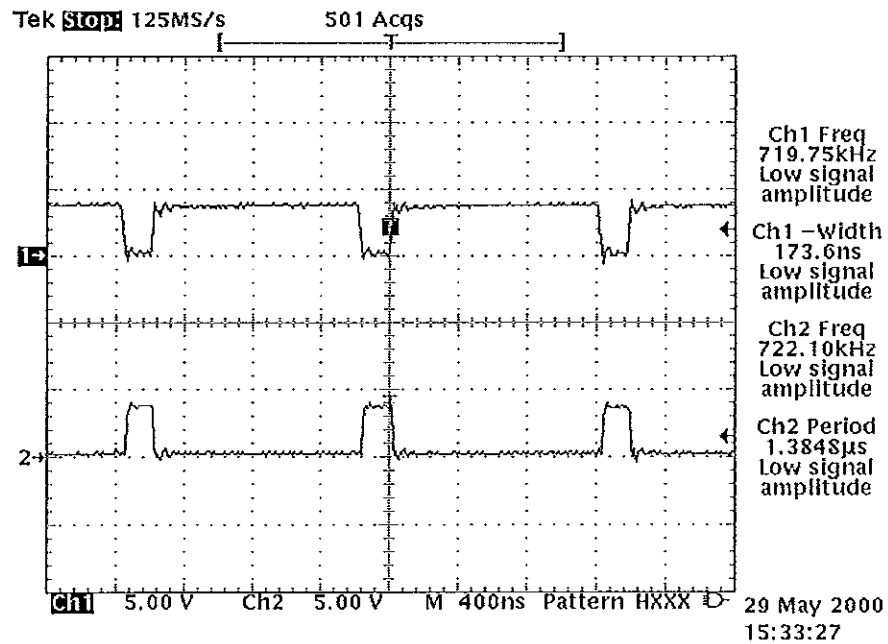
ผลการสร้างสัญญาณ CClk มีดังนี้

- สัญญาณ CClk (Ch1) และสัญญาณ Clk (Ch2) เมื่ออยู่ในโหมดการเขียนข้อมูลภาพ



ภาพประกอบ ง22 แสดงสัญญาณ CClk และสัญญาณ Clk เมื่ออยู่ในโหมดการเขียน

- สัญญาณ CClk (Ch1) และสัญญาณ Data strobe ที่อินเวอร์ต (Ch2) เมื่ออยู่ในโหมดการอ่าน



ภาพประกอบ ง23 แสดงสัญญาณ CClk และสัญญาณ Data strobe ที่อินเวอร์ตเมื่ออยู่ในโหมดการอ่าน

ผลการวิเคราะห์

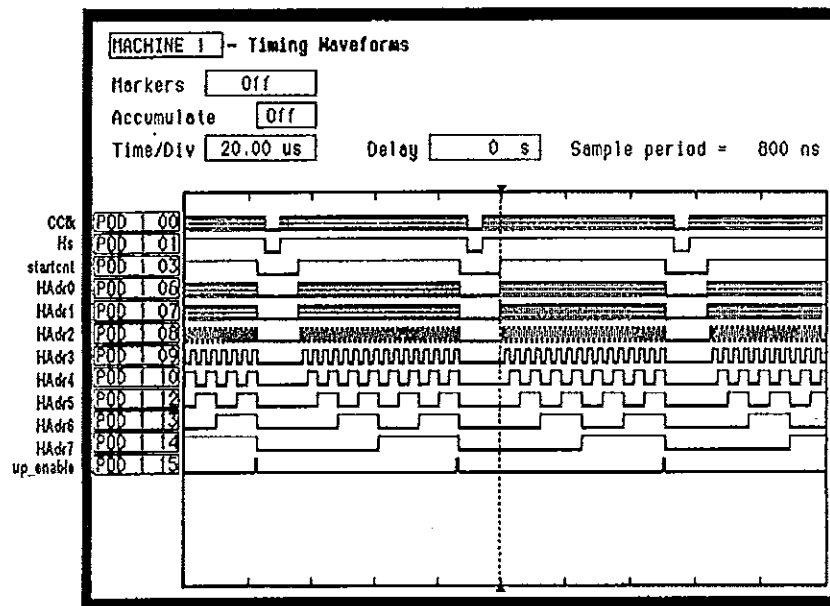
จากผลการทดสอบของส่วน ClkSel พบว่าเมื่ออยู่ในโหมดการเขียนข้อมูลภาพคือสัญญาณ RW เป็นลอจิก '1' สัญญาณ CClk จะมีค่าเท่ากับสัญญาณ Clk ดังภาพประกอบ ง22 และเมื่ออยู่ในโหมดการอ่านคือสัญญาณ RW เป็นลอจิก '0' สัญญาณ CClk จะมีค่าเท่ากับสัญญาณ Data strobe ที่อินเวอร์ตแล้วดังภาพประกอบ ง23 ซึ่งเป็นไปตามที่ออกแบบไว้

1.3 ส่วนวงจรรนับ (Counter) โดยแบ่งออกเป็น 2 ส่วนคือ

1.3.1 วงจรรนับเพื่อกำหนดแอดเดรสทางแนวนอน (Horizontal Address Counter, HorAdrCounter) ให้ผลการทดสอบทั้งสองโหมดดังนี้

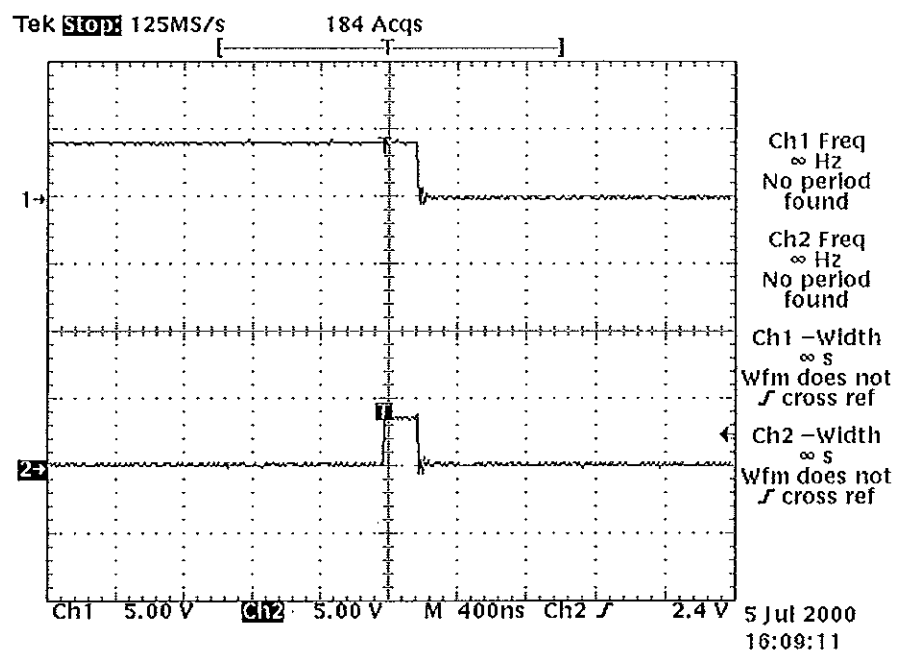
1.3.1.1 โหมดการเขียนข้อมูลภาพ

- สัญญาณ HAdr0-7



ภาพประกอบ ง24 แสดงสัญญาณ HAdr0-7 เมื่ออยู่ในโหมดการเขียน

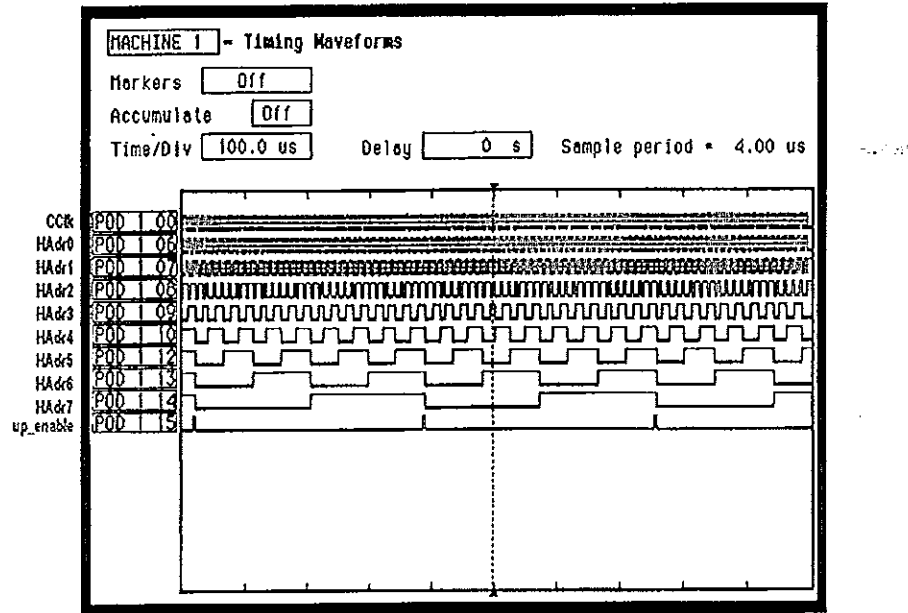
- สัญญาณ HAdr7 (Ch1) และสัญญาณ up_enable (Ch2)



ภาพประกอบ ง25 แสดงสัญญาณ HAdr7 และสัญญาณ up_enable เมื่ออยู่ในโหมดการเขียน

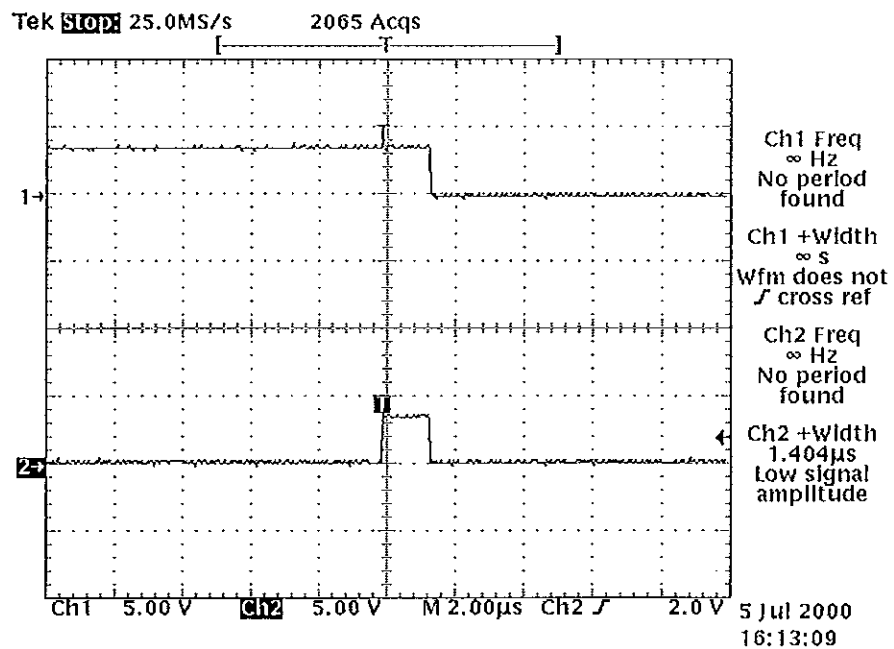
1.3.1.2 โหมดการอ่านข้อมูลภาพ

- สัญญาณ HAdr0-7



ภาพประกอบ ง26 แสดงสัญญาณ HAdr0-7 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ HAdr7 (Ch1) และสัญญาณ up_enable (Ch2)



ภาพประกอบ ง27 แสดงสัญญาณ HAdr7 และสัญญาณ up_enable เมื่ออยู่ในโหมดการอ่าน

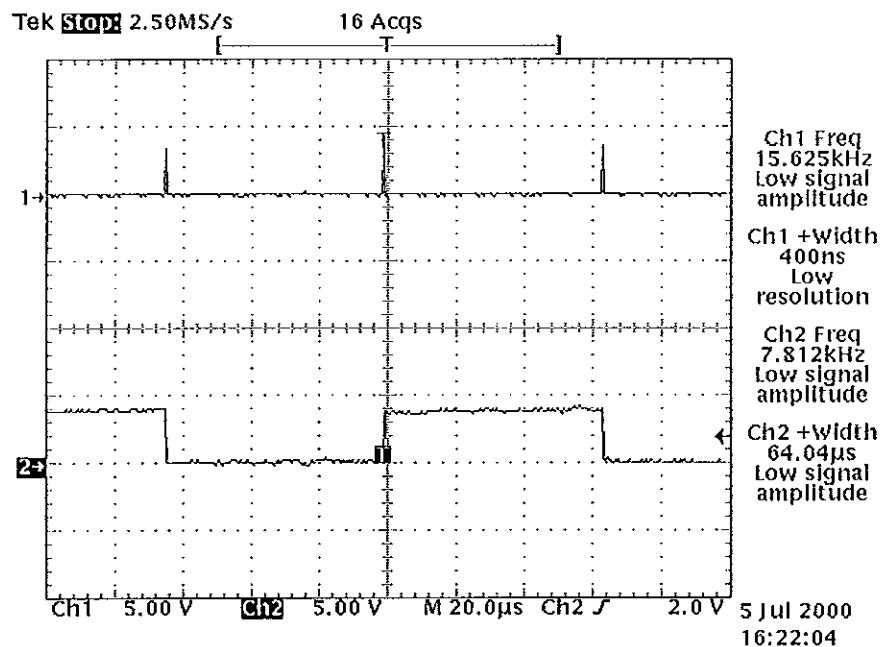
ผลการวิเคราะห์

จากภาพประกอบ ง24 ถึง ง27 พบว่าในส่วน HorAdrCounter สามารถทำงานได้ตามที่ออกแบบไว้ทั้งในโหมดการเขียนและอ่านข้อมูลภาพคือเมื่อสัญญาณ startcnt มีค่าลอจิก '1' HorAdrCounter ก็จะนับจำนวนจุดภาพหรือสร้างแอดเดรสทางแนวนอน 8 บิตตามสัญญาณ CClk และสร้างสัญญาณ up_enable เมื่อนับสัญญาณ CClk ครบ 256 ลูก

1.3.2 วงจรนับเพื่อกำเนิดแอดเดรสทางแนวดิ่ง (Vertical Address Counter, VerAdrCounter) ผลการทำงานทั้งสองโหมดดังนี้

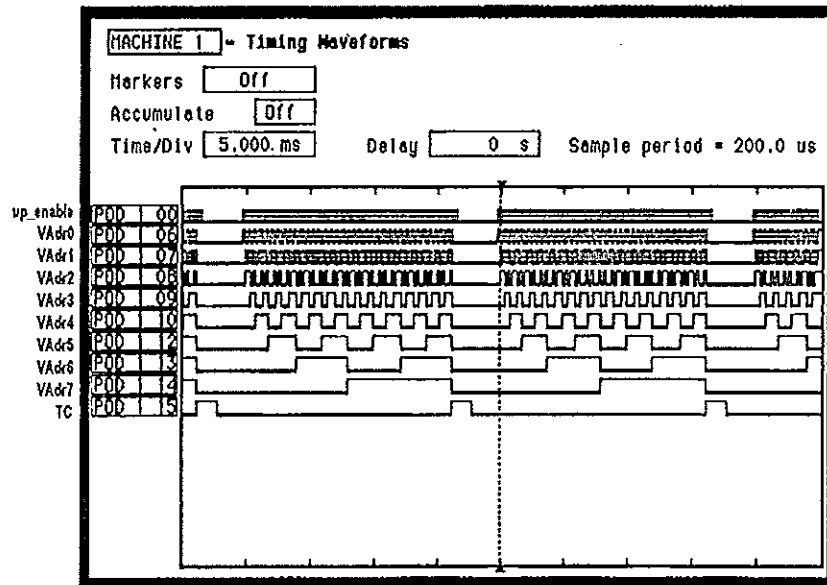
1.3.2.1 โหมดการเขียนข้อมูลภาพ

- สัญญาณ up_enable (Ch1) และสัญญาณ VAdr0 (Ch2)



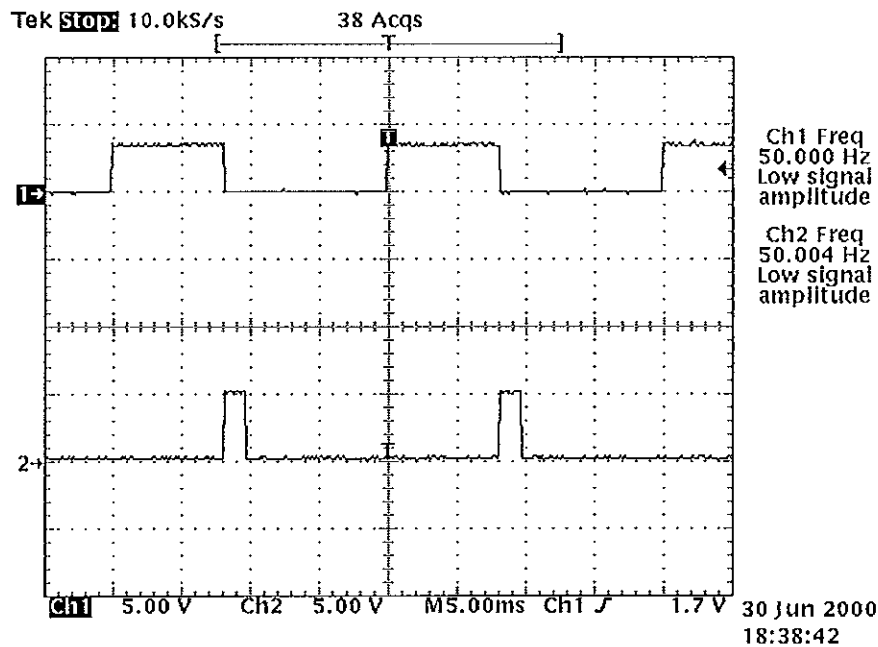
ภาพประกอบ ง28 แสดงสัญญาณ up_enable และสัญญาณ VAdr0 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ VAdr0-7



ภาพประกอบ ง29 แสดงสัญญาณ VAdr0-7 เมื่ออยู่ในโหมดการเขียน

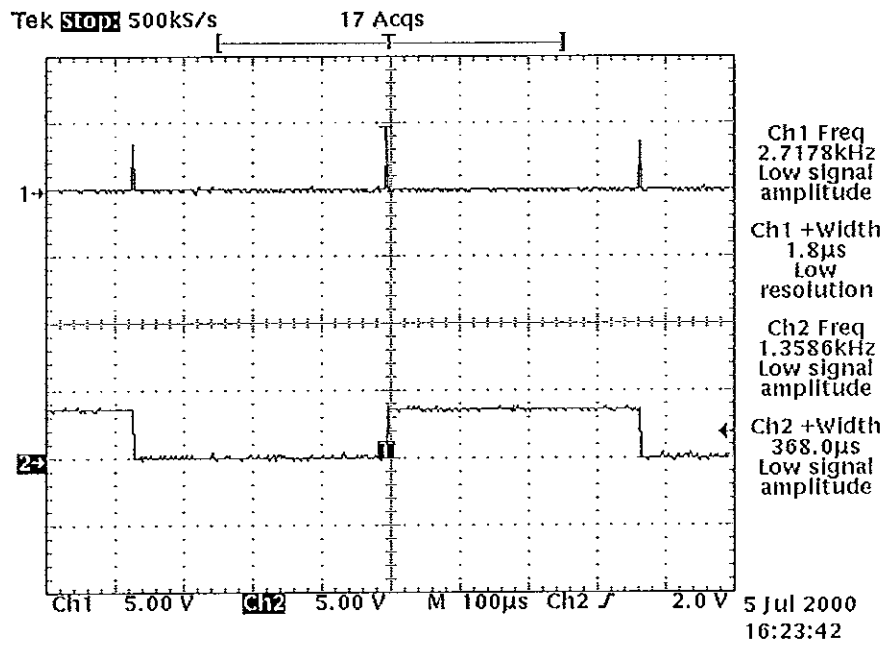
- สัญญาณ VAdr7 (Ch1) และสัญญาณ TC (Ch2)



ภาพประกอบ ง30 แสดงสัญญาณ VAdr7 และสัญญาณ TC เมื่ออยู่ในโหมดการเขียน

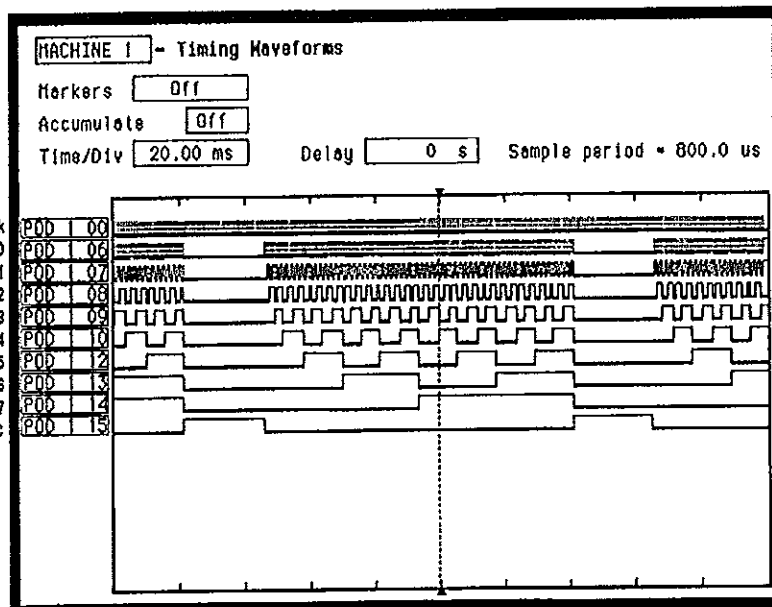
1.3.2.2 โหมดการอ่านข้อมูลภาพ

- สัญญาณ up_enable (Ch1) และสัญญาณ VAdr0 (Ch2)



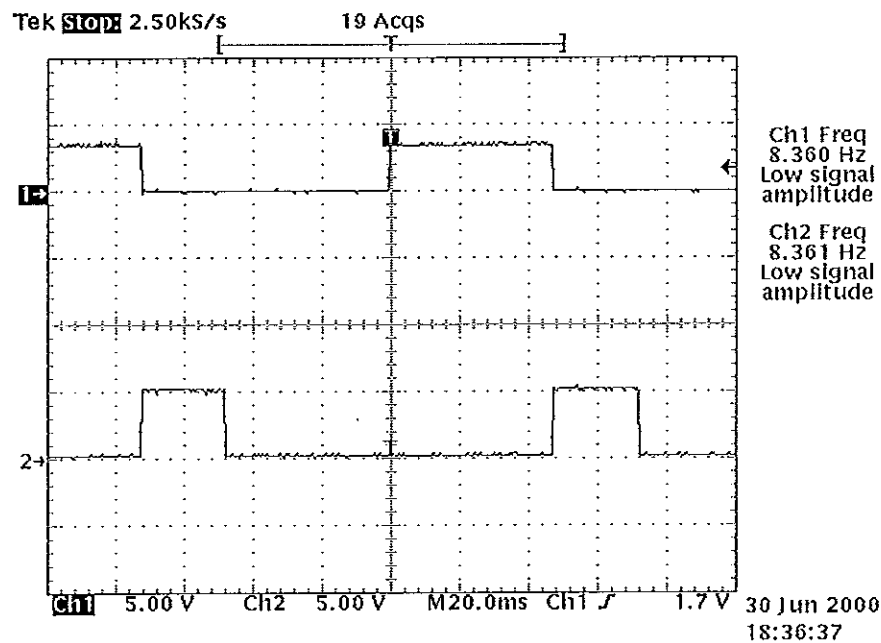
ภาพประกอบ ง31 แสดงสัญญาณ up_enable และสัญญาณ VAdr0 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ VAdr0-7



ภาพประกอบ ง32 แสดงสัญญาณ VAdr0-7 เมื่ออยู่ในโหมดการอ่าน

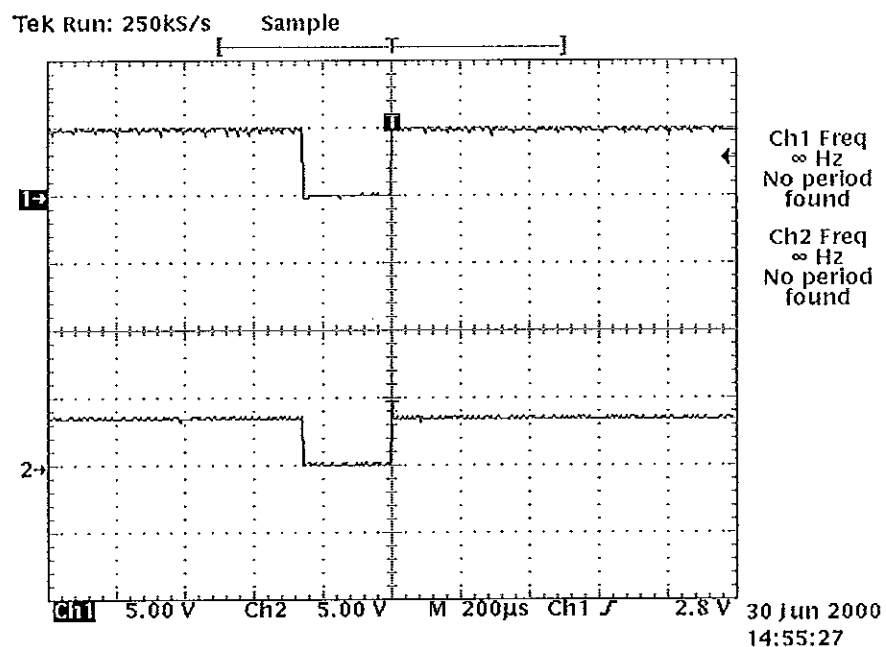
- สัญญาณ VAdr7 (Ch1) และสัญญาณ TC (Ch2)



ภาพประกอบ ง33 แสดงสัญญาณ VAdr7 และสัญญาณ TC เมื่ออยู่ในโหมดการอ่าน

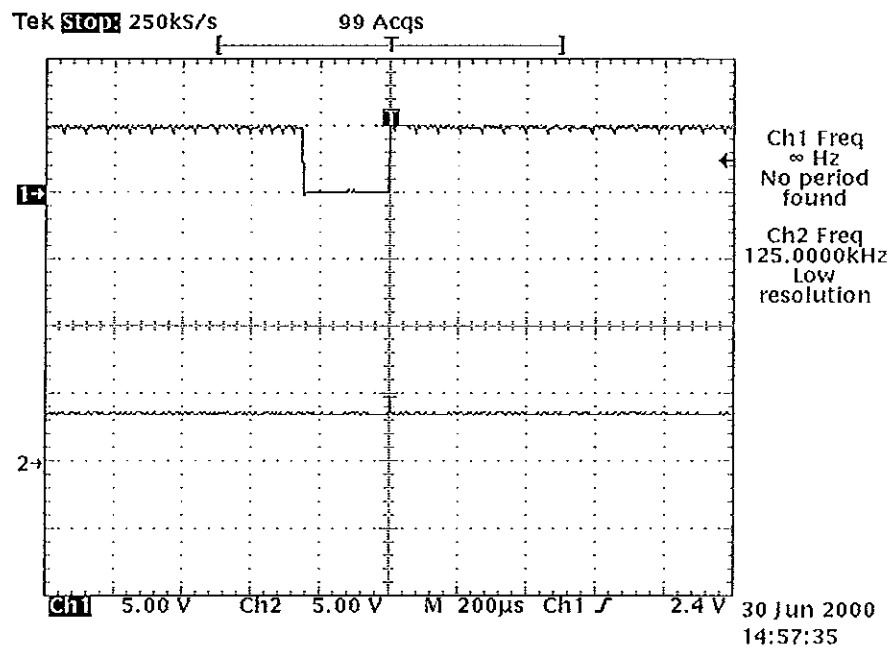
ผลการทดสอบในส่วน VerAdrCounter ดังนี้

- สัญญาณ Vs (Ch1) และสัญญาณ VReset (Ch2) เมื่ออยู่ในโหมดการเขียน



ภาพประกอบ ง34 แสดงสัญญาณ Vs และสัญญาณ VReset เมื่ออยู่ในโหมดการเขียน

- สัญญาณ Vs (Ch1) และสัญญาณ VReset (Ch2) เมื่ออยู่ในโหมดการอ่าน



ภาพประกอบ ง35 แสดงสัญญาณ Vs และสัญญาณ VReset เมื่ออยู่ในโหมดการอ่าน

ผลการวิเคราะห์

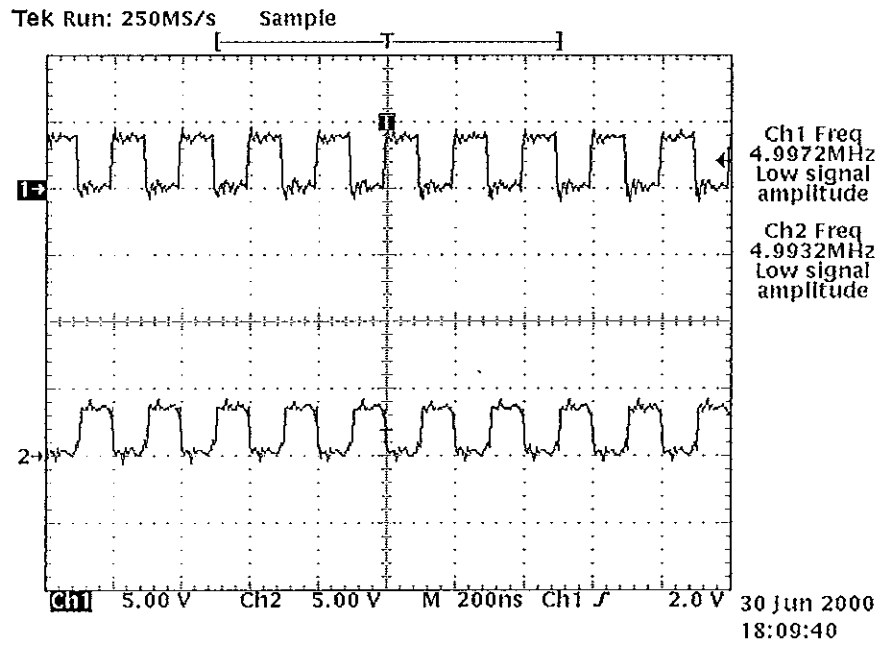
จากภาพประกอบ ง28 ถึง ง33 พบว่าในส่วน VerAdrCounter สามารถทำงานได้ทั้งสองโหมดการทำงานที่ออกแบบเช่นกันคือส่วน VorAdrCounter จะนับจำนวนเส้นภาพหรือสร้างแอดเดรสทางแนวตั้ง 8 บิตตามขอบขาขึ้นของสัญญาณ up_enable และสร้างสัญญาณ TC เมื่อนับครบ 256 ลูกและสำหรับส่วน VResetSel ก็ให้ผลตามต้องการคือเมื่ออยู่ในโหมดการเขียนสัญญาณ VReset จะมีค่าเท่ากับสัญญาณ Vs แต่เมื่ออยู่ในโหมดการอ่านสัญญาณ VReset จะมีค่าเป็นลอจิก '1' ดังภาพประกอบ ง34 และ ง35 ตามลำดับ

2. ส่วนการอินเตอร์เฟสกับหน่วยความจำแรม (RAM Interface)

2.1 โหมดการเขียนข้อมูลภาพ

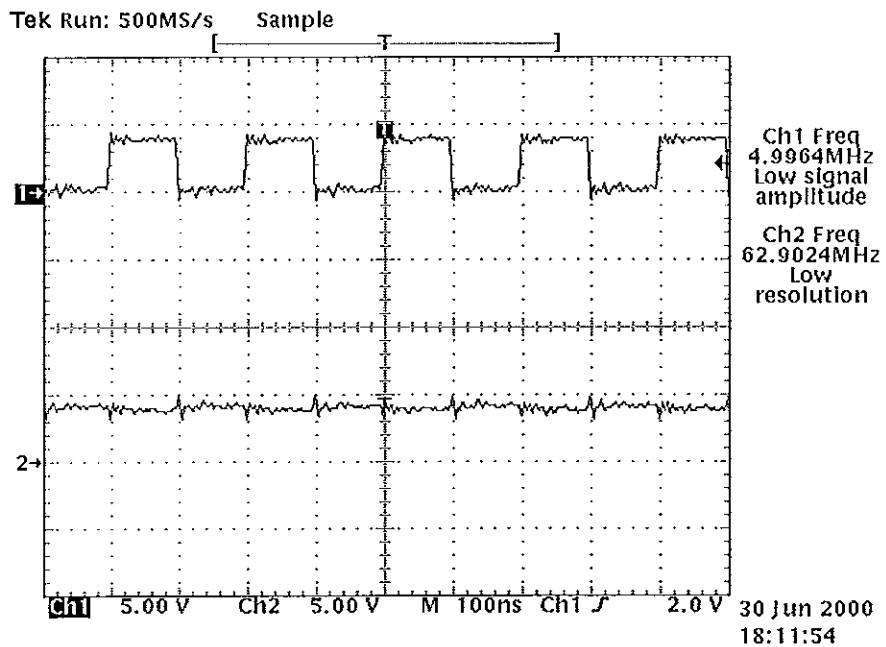
2.1.1 สัญญาณควบคุมต่างๆ

- สัญญาณ Clk (Ch1) และสัญญาณ RAMWE (Ch2)



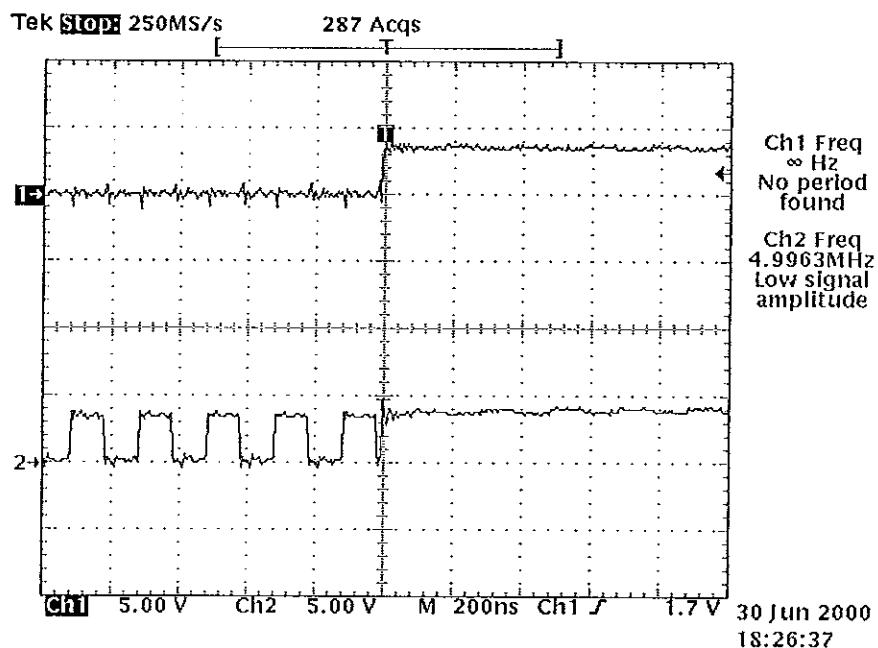
ภาพประกอบ ง36 แสดงสัญญาณ Clk และสัญญาณ RAMWE เมื่ออยู่ในโหมดการเขียน

- สัญญาณ RAMWE (Ch1) และสัญญาณ RAMOE (Ch2)



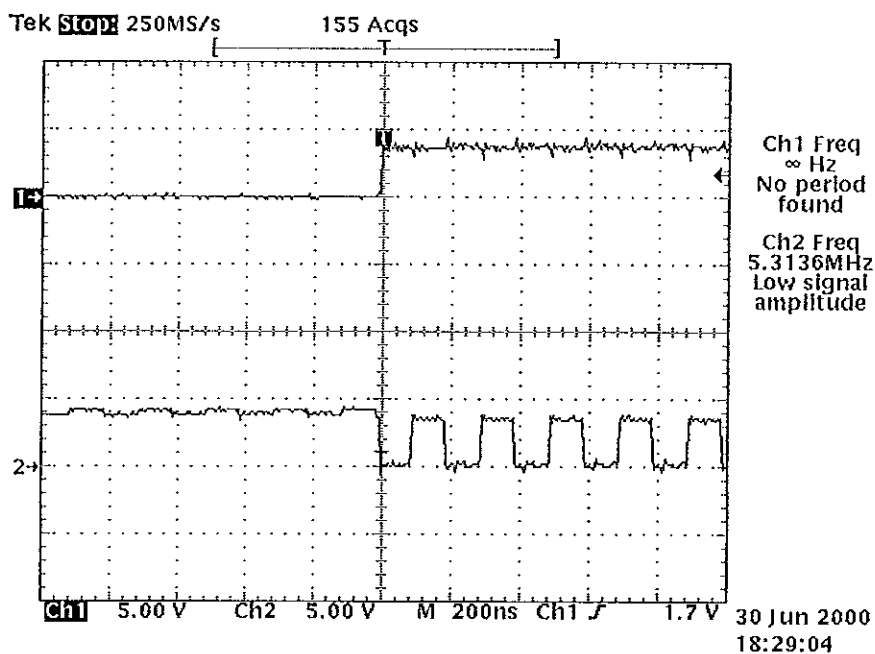
ภาพประกอบ ง37 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการเขียน

- สัญญาณ VAdr7 (Ch1) และสัญญาณ RAMCE1 (Ch2)



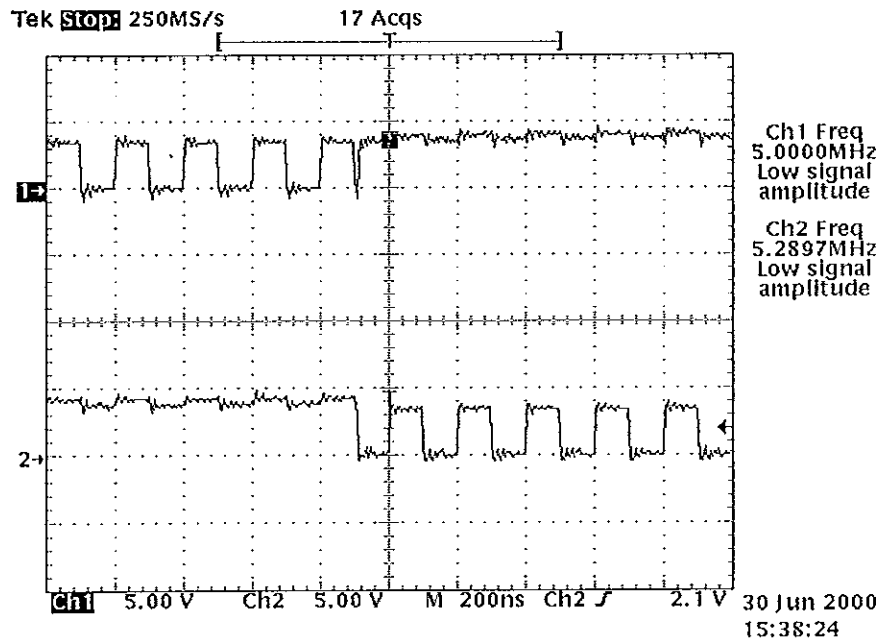
ภาพประกอบ ง38 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE1 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ VAdr7 (Ch1) และสัญญาณ RAMCE2 (Ch2)



ภาพประกอบ ง39 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการเขียน

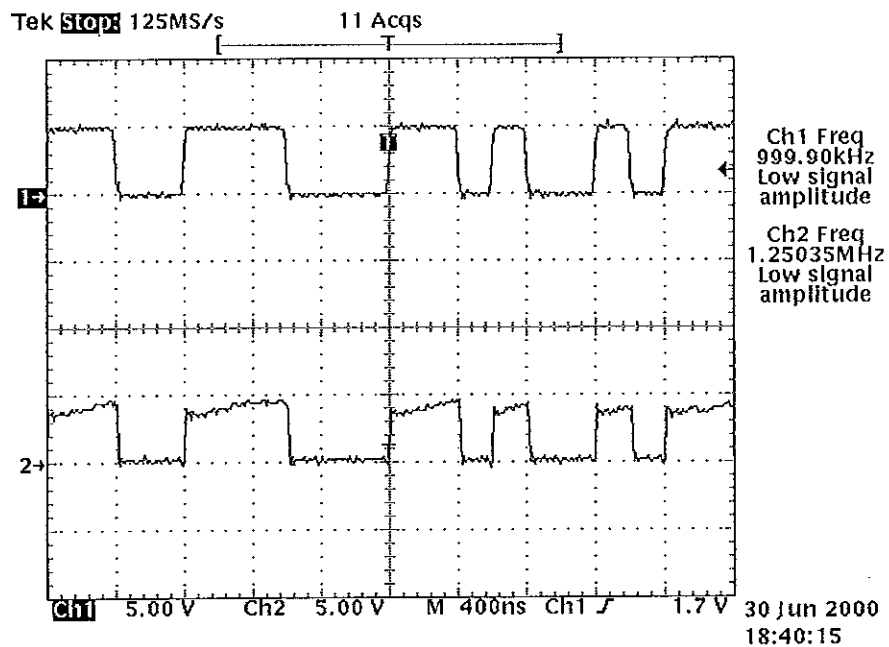
- สัญญาณ RAMCE1 (Ch1) และสัญญาณ RAMCE2 (Ch2)



ภาพประกอบ ง40 แสดงสัญญาณ RAMCE1 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการเขียน

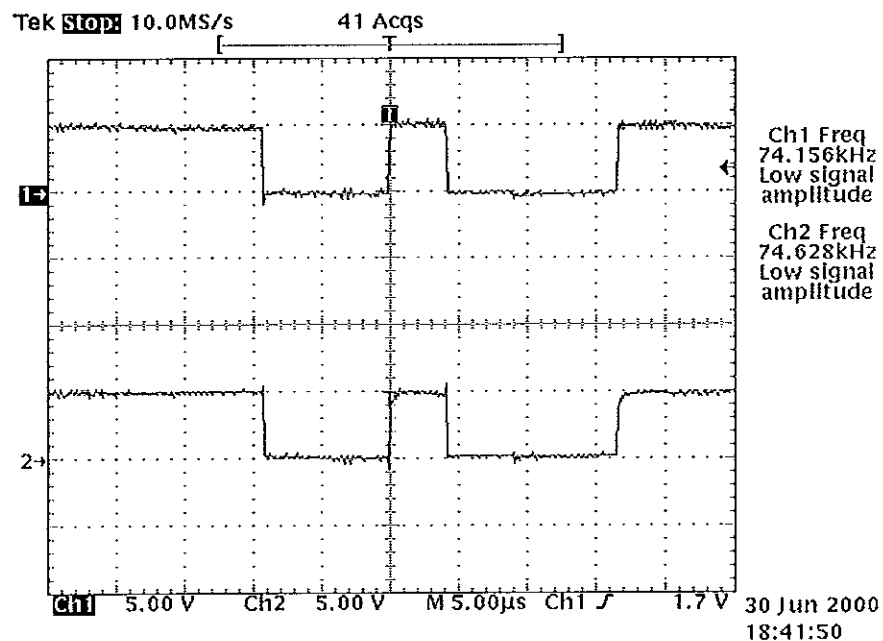
2.1.2 สัญญาณ ADCData กับสัญญาณ RAMData

- สัญญาณ ADCData0 (Ch1) และสัญญาณ RAMData0 (Ch2)



ภาพประกอบ ง41 แสดงสัญญาณ ADCData0 และสัญญาณ RAMData0 เมื่ออยู่ในโหมดการเขียน

- สัญญาณ ADCData7 (Ch1) และสัญญาณ RAMData7 (Ch2)



ภาพประกอบ ง42 แสดงสัญญาณ ADCData7 และสัญญาณ RAMData7 เมื่ออยู่ในโหมดการเขียน

ผลการวิเคราะห์

ผลการทดสอบส่วน RAMInterface เมื่ออยู่ในโหมดการเขียนข้อมูลภาพให้ผลดังนี้

- จากภาพประกอบ ง36 ถึง ง37 แสดงสัญญาณ RAMWE และ RAMOE พบว่าสัญญาณ RAMWE จะมีค่าเท่ากับสัญญาณ Clk ที่อินเวอร์ตในขณะที่สัญญาณ RAMOE มีค่าเป็นลอจิก '1' ตามที่ออกแบบไว้

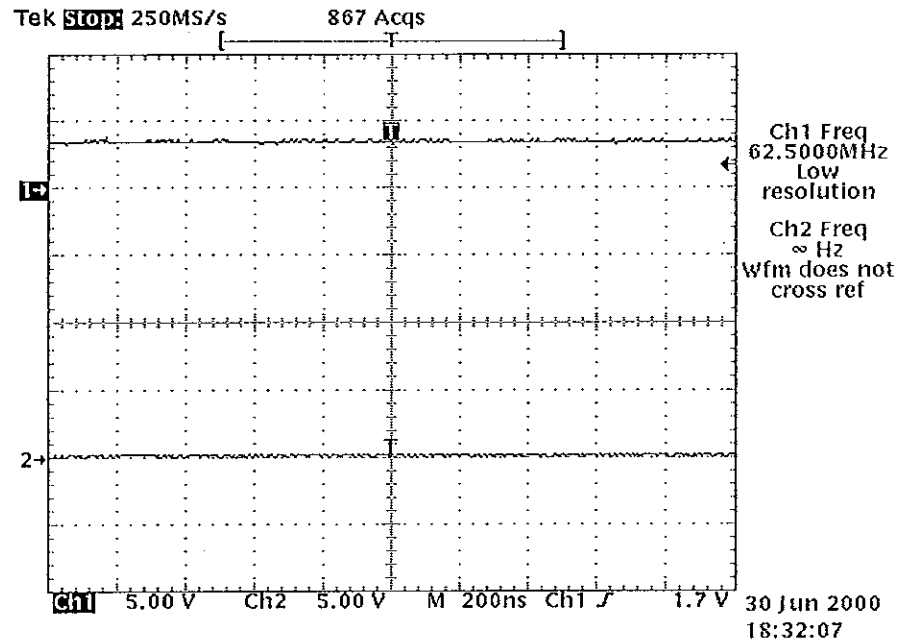
- จากภาพประกอบ ง38 ถึง ง40 แสดงสัญญาณ RAMCE1 และ RAMCE2 พบว่าเกิดสัญญาณทั้งสองสลับกันคือสัญญาณ RAMCE1 จะแอกทีฟ (มีค่าเท่ากับสัญญาณ Clk ที่อินเวอร์ต) เมื่อสัญญาณ VAdr7 มีค่าลอจิก '0' ส่วนสัญญาณ RAMCE2 จะแอกทีฟเมื่อสัญญาณ VAdr7 มีค่าลอจิก '1' ตามที่ออกแบบ

- จากภาพประกอบ ง41 ถึง ง42 แสดงผลการทดสอบในการส่งข้อมูล ADCData0-7 ให้หน่วยความจำแรมผ่านสัญญาณ RAMData0-7 โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ ADCData0 กับ ADCData7 และ RAMData0 กับ RAMData7 ตามลำดับ พบว่าสัญญาณทั้งสองมีค่าเหมือนกันตรงตามที่ออกแบบไว้เช่นกัน

2.2 โหมดการอ่านข้อมูลภาพ

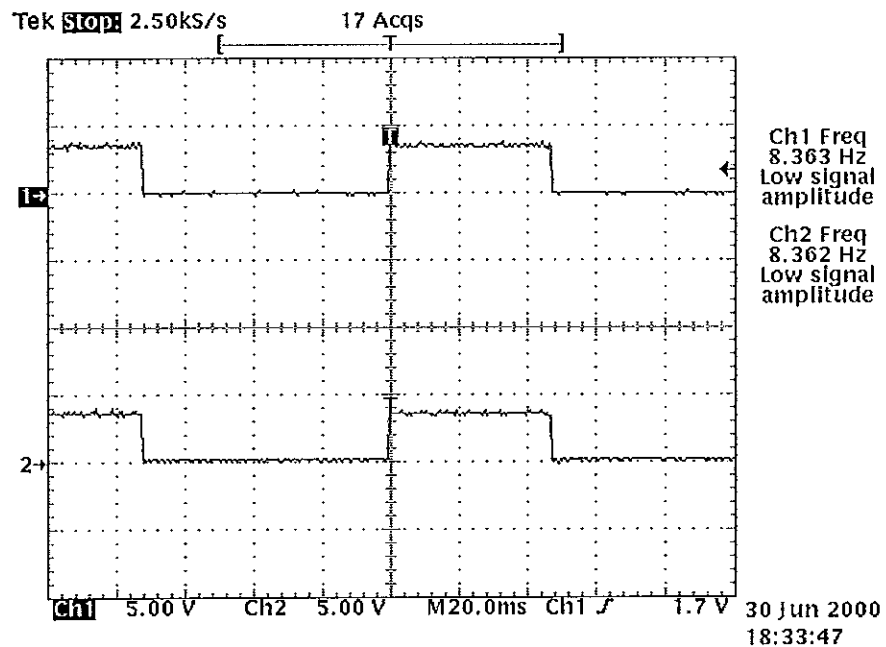
2.2.1 สัญญาณควบคุมต่างๆ

- สัญญาณ RAMWE (Ch1) และสัญญาณ RAMOE (Ch2)



ภาพประกอบ ง43 แสดงสัญญาณ RAMWE และสัญญาณ RAMOE เมื่ออยู่ในโหมดการอ่าน

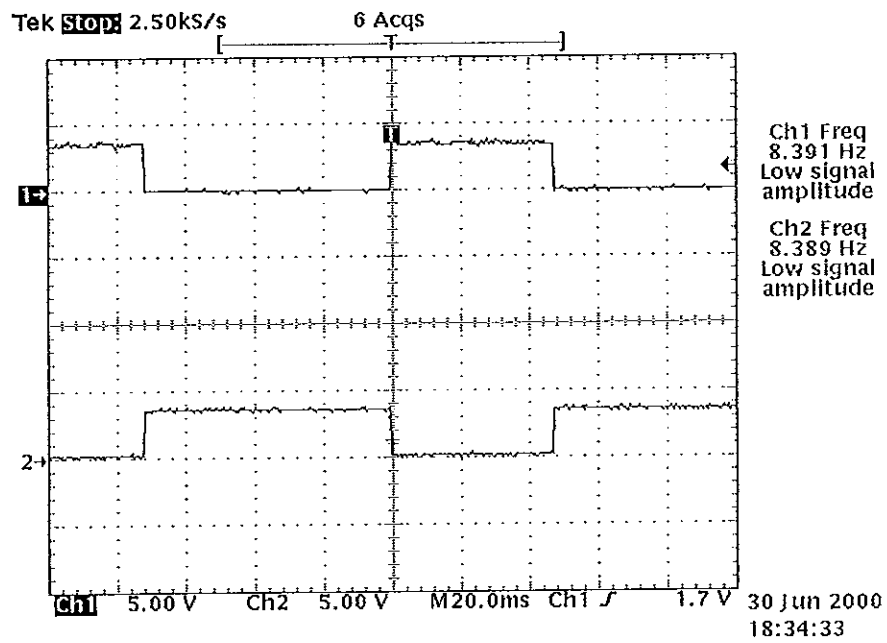
- สัญญาณ VAdr7 (Ch1) และสัญญาณ RAMCE1 (Ch2)



ภาพประกอบ ง44 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE1 เมื่ออยู่ใน

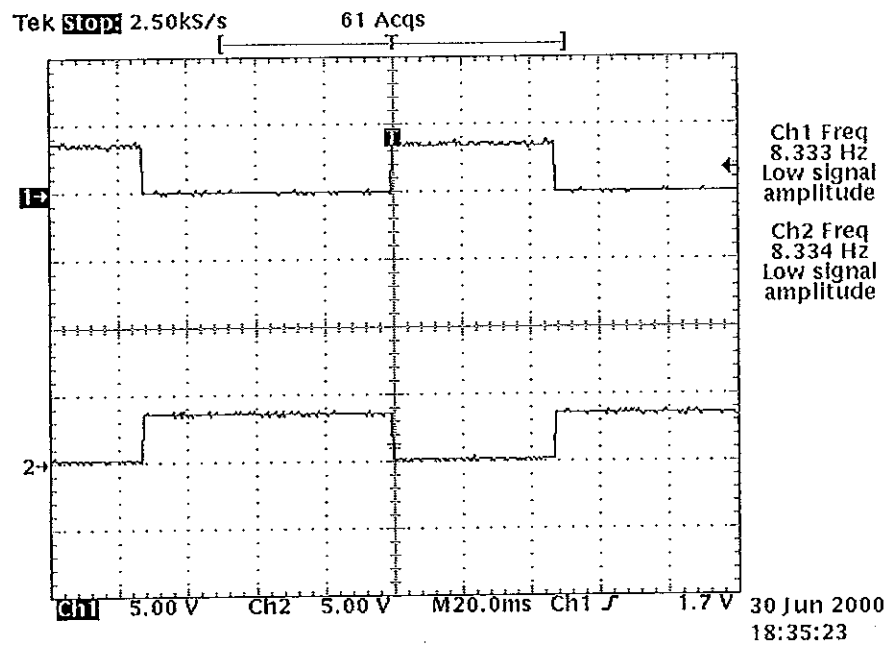
โหมดการอ่าน

- สัญญาณ VAdr7 (Ch1) และสัญญาณ RAMCE2 (Ch2)



ภาพประกอบ ง45 แสดงสัญญาณ VAdr7 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน

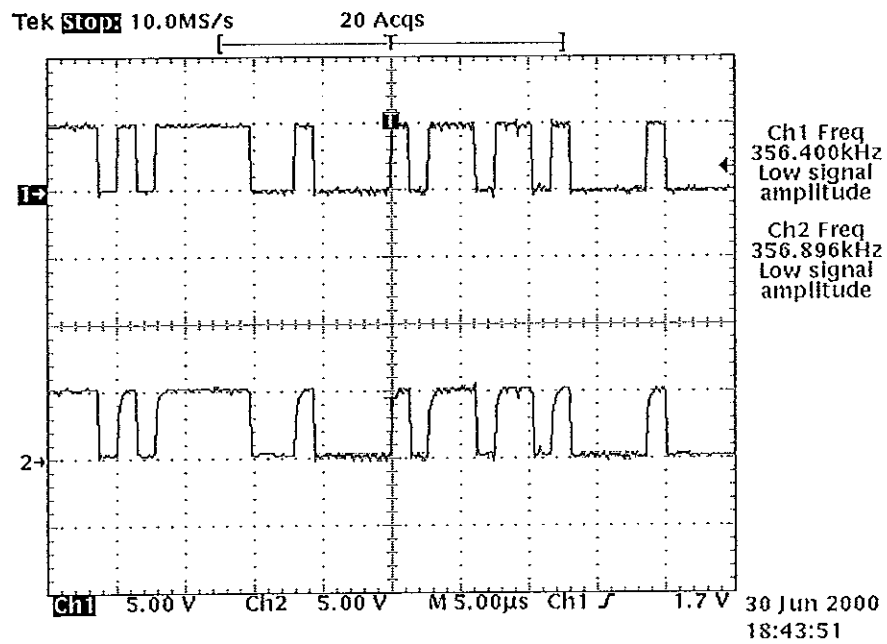
- สัญญาณ RAMCE1 (Ch1) และสัญญาณ RAMCE2 (Ch2)



ภาพประกอบ ง46 แสดงสัญญาณ RAMCE1 และสัญญาณ RAMCE2 เมื่ออยู่ในโหมดการอ่าน

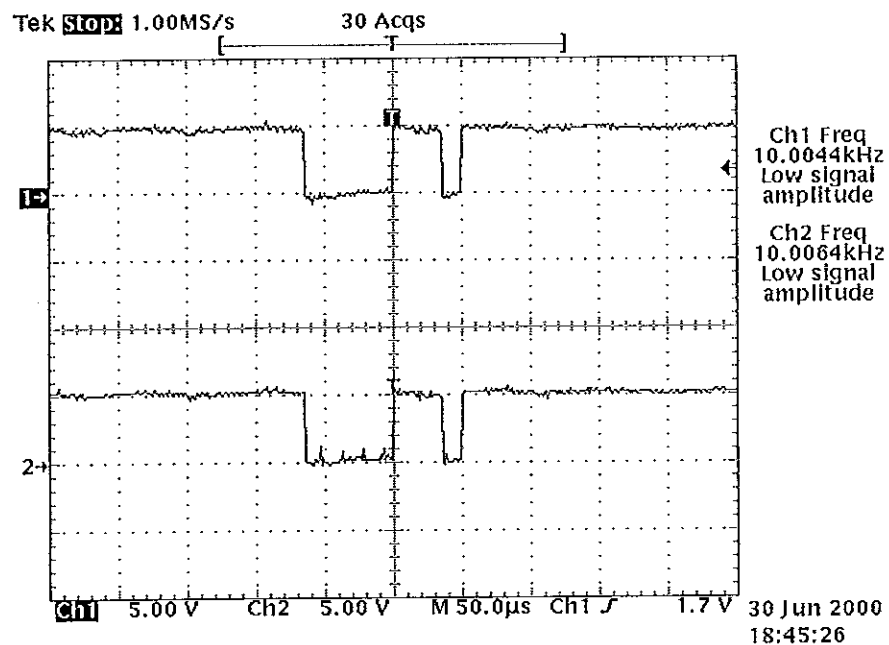
2.2.2 สัญญาณ RAMData กับสัญญาณ PCData

- สัญญาณ RAMData0 (Ch1) และสัญญาณ PCData0 (Ch2)



ภาพประกอบ ๔๗ แสดงสัญญาณ RAMData0 และสัญญาณ PCData0 เมื่ออยู่ในโหมดการอ่าน

- สัญญาณ RAMData7 (Ch1) และสัญญาณ PCData7 (Ch2)



ภาพประกอบ ๔๘ แสดงสัญญาณ RAMData7 และสัญญาณ PCData7 เมื่ออยู่ในโหมดการอ่าน

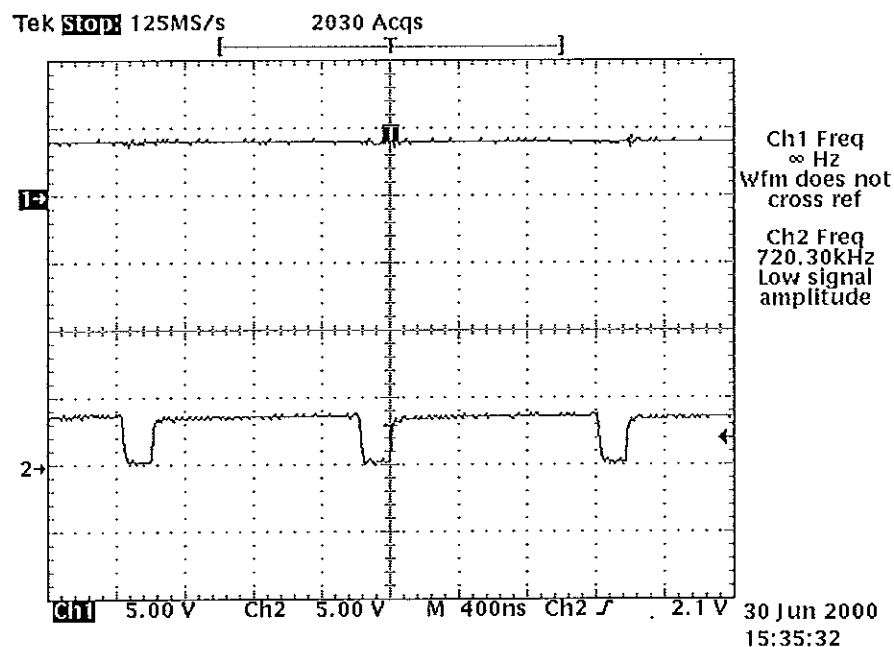
ผลการวิเคราะห์

ผลการทดสอบของส่วน RAMInterface เมื่ออยู่ในโหมดการอ่านข้อมูลภาพให้ผลดังนี้

- จากภาพประกอบ ง41 แสดงสัญญาณ RAMWE และ RAMOE พบว่าสัญญาณ RAMWE จะมีค่าลอจิก '1' ขณะที่สัญญาณ RAMOE มีค่าลอจิก '0' ตรงตามที่ยอกแบบ
- จากภาพประกอบ ง44 ถึง ง46 แสดงสัญญาณ RAMCE1 และ RAMCE2 พบว่าเกิดสัญญาณทั้งสองสลับกันคือสัญญาณ RAMCE1 จะแอกทีฟ (มีค่าเป็นลอจิก '0') เมื่อสัญญาณ VAdr7 มีค่าลอจิก '0' ส่วนสัญญาณ RAMCE2 จะแอกทีฟเมื่อสัญญาณ VAdr7 มีค่าลอจิก '1'
- จากภาพประกอบ ง47 ถึง ง48 แสดงผลการทดสอบในการรับข้อมูลจากหน่วยความจำแรมคือสัญญาณ RAMData0-7 ส่งให้เครื่องคอมพิวเตอร์ผ่านทางพอร์ตขนานซึ่งคือสัญญาณ PCData0-7 โดยเลือกแสดงสัญญาณทั้งสองเฉพาะบิตที่ 1 และ 8 คือ RAMData0 กับ RAMData7 และ PCData0 กับ PCData7 ตามลำดับ ซึ่งพบว่าสัญญาณทั้งสองมีค่าเหมือนกัน ตรงตามที่ยอกแบบเช่นกัน

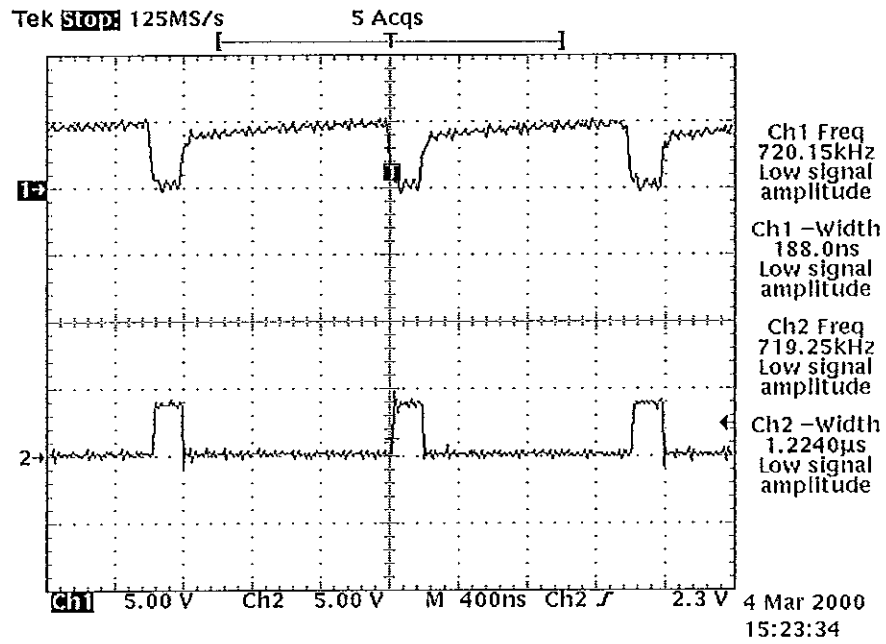
3. ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP (EPP Signal Control, EPPSigCntl)

- สัญญาณ Write (Ch1) และสัญญาณ Data strobe (Ch2) จากพอร์ตขนาน



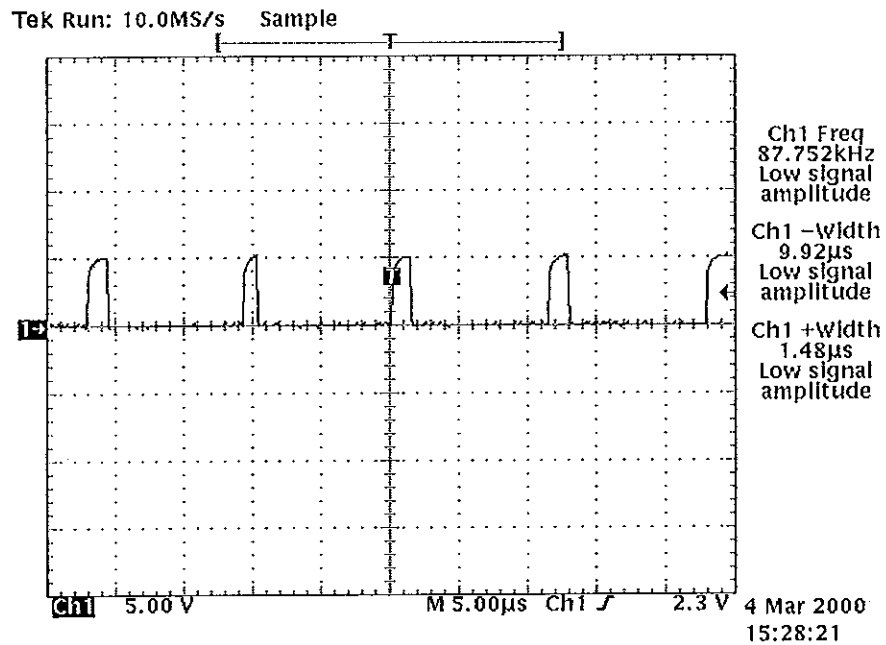
ภาพประกอบ ง49 แสดงสัญญาณ Write และสัญญาณ Data strobe จากพอร์ตขนาน

- สัญญาณ Data strobe (Ch1) เมื่อมีสัญญาณ Wait ตอรับ (Ch2)



ภาพประกอบ ง50 แสดงสัญญาณ Data strobe เมื่อมีสัญญาณ Wait ตอรับ

- สัญญาณ Data strobe (Ch1) เมื่อไม่มีสัญญาณ Wait ตอรับ



ภาพประกอบ ง51 แสดงสัญญาณ Data strobe เมื่อไม่มีสัญญาณ Wait ตอรับ

ผลการวิเคราะห์

จากภาพประกอบ ง49 ถึง ง51 แสดงสัญญาณทั้ง 3 สัญญาณที่ใช้ในการทำ Handshaking คือสัญญาณ Write, Data strobe และ Wait พบว่าสามารถสร้างสัญญาณ Wait ได้ตามที่ออกแบบและมีข้อสังเกตคือความถี่ของสัญญาณ Data strobe ที่ไม่มีสัญญาณ Wait ตอบรับในภาพประกอบ ง51 ซึ่งมีค่าเท่ากับ 87.752 กิโลเฮิร์ตซ์จะมีค่าน้อยกว่าความถี่ของสัญญาณ Data strobe ที่มีสัญญาณ Wait ตอบรับในภาพประกอบ ง50 ซึ่งมีค่าเท่ากับ 720.15 กิโลเฮิร์ตซ์และมีช่วงเวลาที่สัญญาณพัลส์เป็นลอจิก '0' มากกว่าคือเท่ากับ 10 ไมโครวินาที ทั้งนี้เนื่องจากในรีจิสเตอร์สถานะของพอร์ตนานในโหมดการทำงานแบบ EPP จะมีบิตที่มีชื่อว่า Time-out bit ซึ่งจะถูกเซ็ตค่าเป็นลอจิก '1' เมื่อไม่มีสัญญาณ Wait ตอบรับภายในเวลา 10 ไมโครวินาที

ภาคผนวก จ โครงสร้างและสถาปัตยกรรมพื้นฐานของวงจรรวม FPGA

PLD (Programmable Logic Device) เป็นอุปกรณ์ที่สามารถเปลี่ยนแปลงคุณลักษณะทางลอจิก (Logic characteristics) ของอุปกรณ์ได้ตามความต้องการ ตัวอย่างที่พื้นฐานที่สุดของอุปกรณ์ประเภทนี้คืออุปกรณ์พวก PAL (Programmable Array Logic) โดยภายในอุปกรณ์ประกอบด้วยแถวลำดับของเกตแบบแอนด์ (Arrays of AND gates) ที่สามารถโปรแกรมได้และแถวลำดับของเกตแบบออร์ (Arrays of OR gates) ที่ไม่สามารถโปรแกรมได้ เมตริกซ์ของสวิตช์ (Switch matrix) จะเป็นตัวเลือกอินพุตของ PAL ให้กับอินพุตของแถวลำดับของเกตแบบแอนด์ซึ่งจะต่อกับเมตริกซ์ของเกตแบบออร์อีกทีหนึ่ง นอกจากนี้ยังมีอุปกรณ์อื่นอีกคืออุปกรณ์พวก CPLD (Complex Programmable Logic Device) โดยประกอบด้วยแมโครเซลล์ (Macro cells) ขนาดเล็กที่สามารถโปรแกรมการเชื่อมต่อระหว่างกันได้ ปัจจุบันมีผู้ผลิตอุปกรณ์พวก CPLD หลายบริษัทเช่น AMD, Lattice, Xilinx เป็นต้น

สำหรับอุปกรณ์จำพวกวงจรรวม FPGA (Field Programmable Gate Array) จะประกอบด้วยแถวลำดับของลอจิกบล็อก (Arrays of logic blocks) ที่สามารถเชื่อมต่อกันเพื่อสร้างการทำงานของลอจิกที่ซับซ้อนและมีความยืดหยุ่นกว่าอุปกรณ์พวก PAL และ CPLD โดยวงจรรวม FPGA สามารถแบ่งออกเป็น 2 ประเภทคือ

1. Fine Grained FPGA ประกอบด้วยทรานซิสเตอร์หรือแมโครเซลล์ขนาดเล็กจำนวนมาก

2. Coarse Grained FPGA ประกอบด้วยแมโครเซลล์ขนาดใหญ่กว่า

สำหรับภายใน Macro Cells จะประกอบด้วยชุดของฟลิปฟลอป (Flip Flops) และตาราง LUTs (Look Up Tables) เพื่อสร้างฟังก์ชันการทำงานแบบคอมบินเนชันลอจิกได้ (Combinatorial logic function) และจะมีสวิตช์ (Switches) หรือมัลติเพล็กซ์เซอร์ (Multiplexers) ที่ทำให้สามารถใช้งานแมโครเซลล์ได้หลากหลายรูปแบบ โดยแมโครเซลล์แต่ละเซลล์จะถูกเชื่อมต่อเข้าด้วยกันด้วยการรวมกัน (Combination) ของเมตริกซ์สวิตช์และเมตริกซ์ของ Metal lines ที่สามารถใช้งานโดยผ่านทรานซิสเตอร์, Fuses/Antifuses หรือมัลติเพล็กซ์เซอร์

นอกจากนี้โครงสร้างภายในของวงจรรวม FPGA สามารถแบ่งออกเป็นประเภทต่างๆ ตามลักษณะเทคโนโลยีที่ใช้ดังนี้

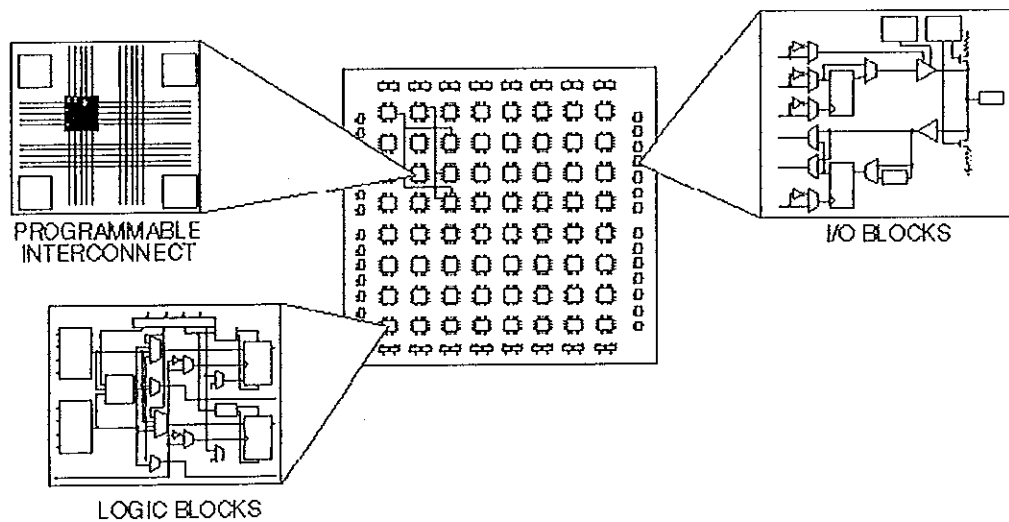
1. ประเภท PROM/EPROM/EEPROM/FLASH based FPGA วงจรรวม FPGA ประเภทนี้วงจรรวมภายในจะถูกโปรแกรมทั้งหมดและอาจสามารถโปรแกรมใหม่ได้โดยขึ้นกับชนิดของ

โครงสร้างว่าเป็นชนิดใด เช่น ชนิด E (Erasable) คือสามารถลบได้ ชนิด EE (Electrically Erasable) คือลบด้วยไฟฟ้า หรือชนิด PROM คือโปรแกรมได้เพียงครั้งเดียว

2. ประเภท Anti-Fuse/Fuse based FPGA โดย Fuses จะถูกวางตำแหน่งอย่างถาวร มีลักษณะคล้าย PROM คือสามารถโปรแกรมได้เพียงครั้งเดียว แต่มีข้อดีในด้านขนาดและราคาที่ถูกกว่า

3. ประเภท Static RAM based FPGA โดย SRAM cells จะถูกกำหนดให้เป็นตัวกำเนิดฟังก์ชัน (Function generator) เพื่อจำลองเป็น Combinatorial logic และจะถูกใช้ควบคุม มัลติเพล็กซ์และทรัพยากรในการเชื่อมต่อ (Routing resources) ซึ่งวงจรรวม FPGA ประเภทนี้ จะเป็นที่นิยมมากที่สุด

สำหรับในงานวิจัยนี้ใช้วงจรรวม FPGA ประเภท Static RAM based FPGA ของบริษัท Xilinx โดยมีโครงสร้างพื้นฐานภายในของวงจรรวม FPGA ของบริษัท Xilinx ดังในภาพประกอบ ๑1 และมีส่วนประกอบที่สำคัญดังนี้

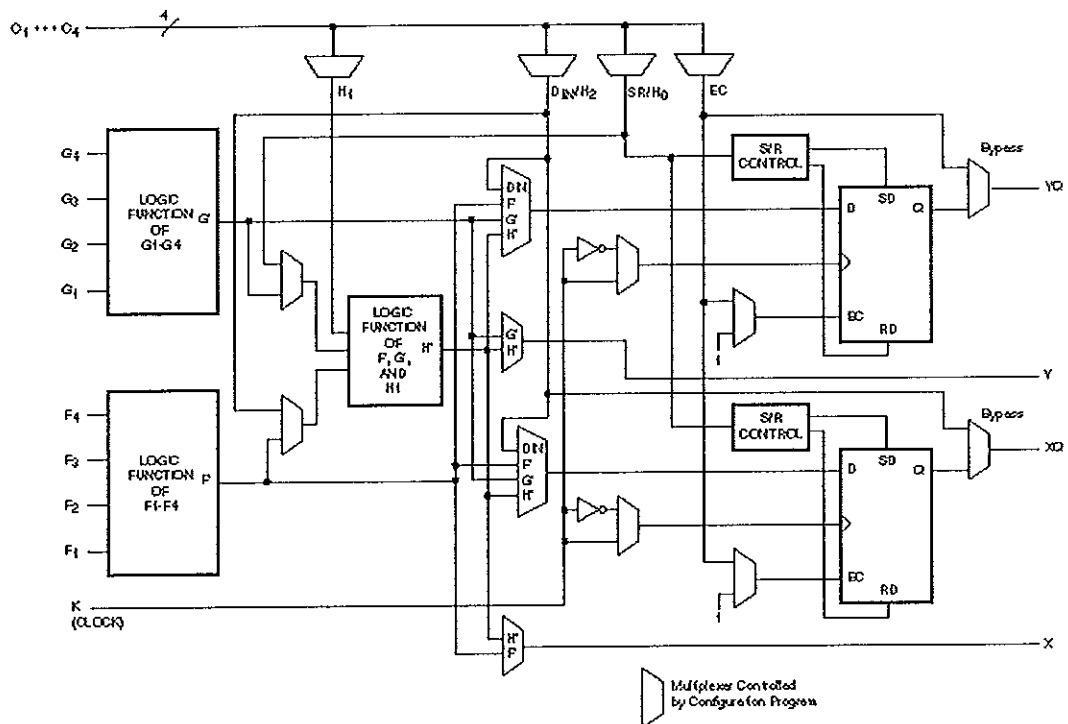


ภาพประกอบ ๑1 แสดงโครงสร้างพื้นฐานภายในของวงจรรวม FPGA ของบริษัท Xilinx

1. Configurable Logic Block (CLB)

CLB หรือเซลล์พื้นฐานของวงจรรวม FPGA ของบริษัท Xilinx ในภาพประกอบ ๑2 จะประกอบด้วยตัวกำเนิดฟังก์ชันจำนวน 3 ตัวคือ F, G, H Function generator โดยทั้ง F และ G Function generator จะเป็นเพียงบล็อกของแรม (RAM blocks) ที่มีแอดเดรสขนาด 4 บิตและบิตข้อมูล 1 บิต (16X1 RAM) ซึ่งจะเก็บพฤติกรรมทาง Combinatorial (Combinatorial behavior) ของอินพุต ส่วน H Function generator จะมีอินพุตจำนวน 3 ตัว (มาจากเอาต์พุตของทั้ง F และ

G Function generator และจากอินพุตภายนอก (จิกบลิค) โดย CLB สามารถสร้างฟังก์ชันการทำงานตามการจัดเรียงของอินพุตทั้ง 9 ตัว นอกจากนี้ยังมีฟลิปฟลอปจำนวน 2 ตัวสำหรับเก็บเอาต์พุตของตัวกำเนิดฟังก์ชันซึ่งสามารถใช้งานได้สองสถานะ (Din) รวมทั้งมีส่วนวงจรสำหรับเซต/รีเซต (Set/Reset circuitry) ด้วย สำหรับเอาต์พุตของ CLB จะมี 4 ตัวคือ Y, YQ, X และ XQ โดย X สามารถเป็นเอาต์พุตของ F หรือ H Function generator และ Y สามารถเป็นเอาต์พุตของ G หรือ H Function generator ได้



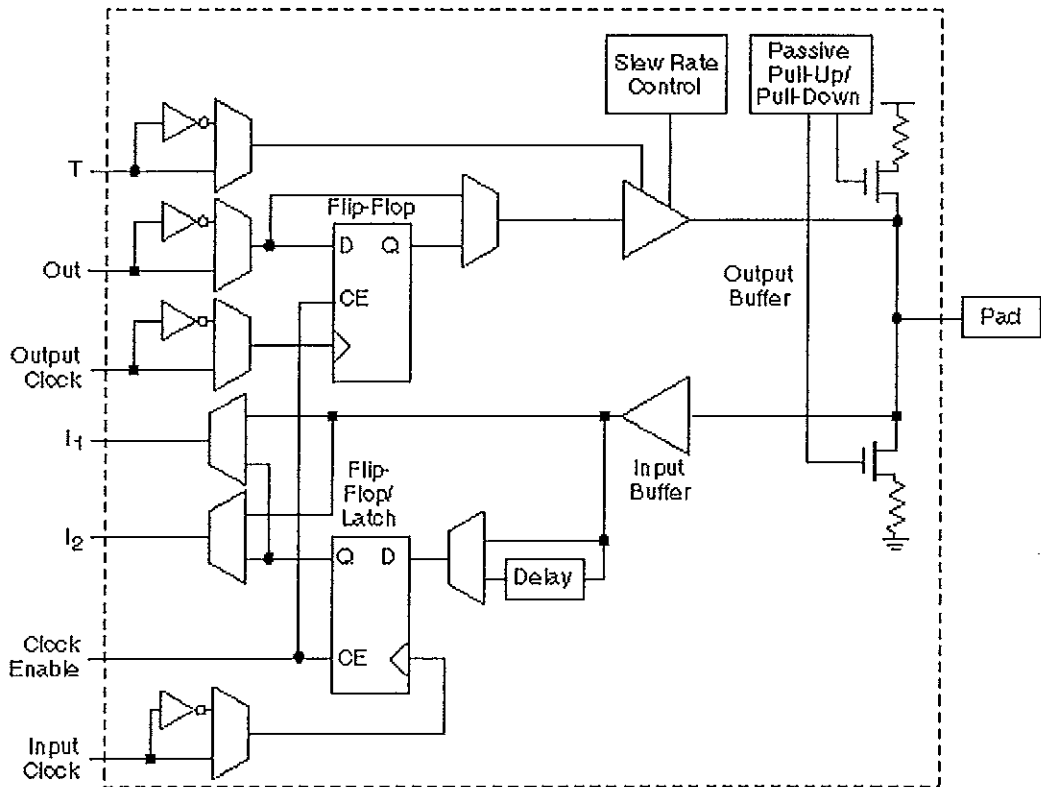
ภาพประกอบ ๑2 แสดงบล็อกไดอะแกรมอย่างง่ายของ XC4000-Series CLB

2. Input Output Block (IOB)

ส่วน IOB นี้จะเป็นส่วนที่ทำให้วงจรรวม FPGA สามารถติดต่อกับขา (Pin) หรือ Pad ได้ จากภาพประกอบ ๑3 แสดงบล็อกไดอะแกรมอย่างง่ายของ Xilinx IOB ในตระกูล XC4000E จะเห็นว่า I1 และ I2 สามารถต่อกับสัญญาณที่อยู่ภายนอกวงจรรวมได้ ในแต่ละ IOB จะมี Flip Flops ที่เป็นอินพุตเอาต์พุตและบัฟเฟอร์ที่เป็นอินพุตเอาต์พุต สำหรับลักษณะอื่นๆของ IOB ที่สำคัญเช่น

- อินพุตสามารถหน่วงเวลาได้หลายนาโนวินาทีเพื่อชดเชยการหน่วงเวลาของสัญญาณนาฬิกา (Clock delays)

- เอาต์พุตสามารถถูกอินเวอร์ตได้
- บัฟเฟอร์สามารถถูกบายพาสเพื่อกำจัดการหน่วงเวลาได้



ภาพประกอบ ๑3 แสดงบล็อกไดอะแกรมอย่างง่ายของ Xilinx IOB ในตระกูล XC4000E

3. ส่วนเชื่อมต่อที่สามารถโปรแกรมได้ (Programmable Interconnects)

ส่วนเชื่อมต่อภายในทุกส่วนจะมาจาก Metal segment ซึ่งมีจุดสวิตช์ซึ่งที่โปรแกรมได้ (Programmable switching points) โดยมีลักษณะการเชื่อมต่ออยู่ 3 ชนิดหลักดังนี้

- Single length lines เป็นเส้นกริดทางแนวนอนและแนวตั้งซึ่งจะตัดผ่านกันตรงเมตริกซ์สวิตช์ที่อยู่ระหว่าง CLB แต่ละบล็อก
- Double length lines เป็นเส้นกริดทางแนวนอนและแนวตั้งซึ่งจะตัดผ่านกันตรงเมตริกซ์สวิตช์ที่อยู่ระหว่าง CLB จำนวน 2 บล็อก
- Long lines เป็นเส้นที่ใช้สร้างเส้นกริดของ Metal interconnection segment ที่ผ่านความยาวและความกว้างทั้งหมดของชุดลำดับ โดยที่ Global buffer สามารถขับเส้นทางแนวตั้งที่เพิ่มเติมเพื่อกระจายสัญญาณนาฬิกาและสัญญาณ high fanout อื่นๆ

ภาคผนวก จ1 จำนวนทรัพยากรภายในของวงจรรวม FPGA เบอร์ XC4003E
ในตระกูล XC4000E ของบริษัท Xilinx

ตาราง จ1 แสดงจำนวนทรัพยากรภายในวงจรรวม FPGA เบอร์ XC4003E เปรียบเทียบกับเบอร์
อื่นในตระกูล XC4000E

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4003E	238	3,000	3,200	2,000 - 5,000	10 x 10	100	360	80
XC4005E/XL	466	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112
XC4006E	608	6,000	8,192	4,000 - 12,000	16 x 16	256	768	128
XC4008E	770	8,000	10,368	6,000 - 15,000	18 x 18	324	936	144
XC4010E/XL	950	10,000	12,800	7,000 - 20,000	20 x 20	400	1,120	160
XC4013E/XL	1368	13,000	18,432	10,000 - 30,000	24 x 24	576	1,536	192
XC4020E/XL	1862	20,000	25,088	13,000 - 40,000	28 x 28	784	2,016	224
XC4025E	2432	25,000	32,768	15,000 - 45,000	32 x 32	1,024	2,560	256
XC4028E/XL	2432	28,000	32,768	18,000 - 50,000	32 x 32	1,024	2,560	256
XC4036E/XL	3078	36,000	41,472	22,000 - 65,000	36 x 36	1,296	3,168	288
XC4044XL	3800	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320
XC4052XL	4598	52,000	61,952	33,000 - 100,000	44 x 44	1,936	4,576	352
XC4062XL	5472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384
XC4085XL	7448	85,000	100,352	55,000 - 180,000	56 x 56	3,136	7,168	448

* Max values of Typical Gate Range include 20-30% of CLBs used as RAM.

ภาคผนวก จ2 Pinout ของวงจรรวม FPGA เบอร์ XC4003E ในตระกูล
XC4000E ของบริษัท Xilinx

ตาราง จ2 แสดง Pinout ของวงจรรวม FPGA เบอร์ XC4003E

Pad Name	PC84	Pad Name	PC84
VCC	P2	VCC	P11
I/O (A8)	P3	GND	P12
I/O (A9)	P4	I/O, PGCK1 (A16)	P13
I/O (A10)	P5	I/O (A17)	P14
I/O (A11)	P6	I/O, TDI	P15
I/O (A12)	P7	I/O, TCK	P16
I/O (A13)	P8	I/O, TMS	P17
I/O (A14)	P9	I/O	P18
I/O, SGCK1 (A15)	P10	I/O	P19

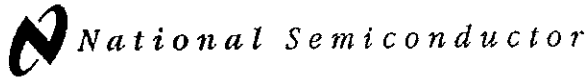
ตาราง ๑2 แสดง Pinout ของวงจรรวม FPGA เบอร์ XC4003E (ต่อ)

Pad Name	PC84	Pad Name	PC84
I/O	P20	I/O	P47
GND	P21	I/O	P48
VCC	P22	I/O	P49
I/O	P23	I/O	P50
I/O	P24	I/O, SGCK3	P51
I/O	P25	GND	P52
I/O	P26	DONE	P53
I/O	P27	VCC	P54
I/O	P28	PROGRAM	P55
I/O, SGCK2	P29	I/O (D7)	P56
O (M1)	P30	I/O, PGCK3	P57
GND	P31	I/O (D6)	P58
I (M0)	P32	I/O (D5)	P59
VCC	P33	I/O (CS0)	P60
I (M2)	P34	I/O (A10)	P61
I/O, PGCK2	P35	I/O (D4)	P62
I/O (HDC)	P36	VCC	P63
I/O (LDC)	P37	GND	P64
I/O	P38	I/O (D3)	P65
I/O	P39	I/O (RS)	P66
I/O	P40	I/O (D2)	P67
I/O (INIT)	P41	I/O	P68
VCC	P42	I/O	P69
GND	P43	I/O (RCLK), RDY/BUSY	P70
I/O	P44	I/O (D0,DIN)	P71
I/O	P45	I/O, SGCK4 (DOUT)	P72
I/O	P46	CCLK	P73

ตาราง ๑2 แสดง Pinout ของวงจรรวม FPGA เบอร์ XC4003E (ต่อ)

Pad Name	PC84	Pad Name	PC84
VCC	P74	I/O (A3)	P80
O, TDO	P75	I/O (A4)	P81
GND	P76	I/O (A5)	P82
I/O (A0, WS)	P77	I/O (A6)	P83
I/O, PGCK4 (A1)	P78	I/O (A7)	P84
I/O (CS1, A2)	P79		

ภาคผนวก จ Data sheet ของวงจรแยกสัญญาณซิงค์เบอร์ LM1881



February 1995

LM1881 Video Sync Separator

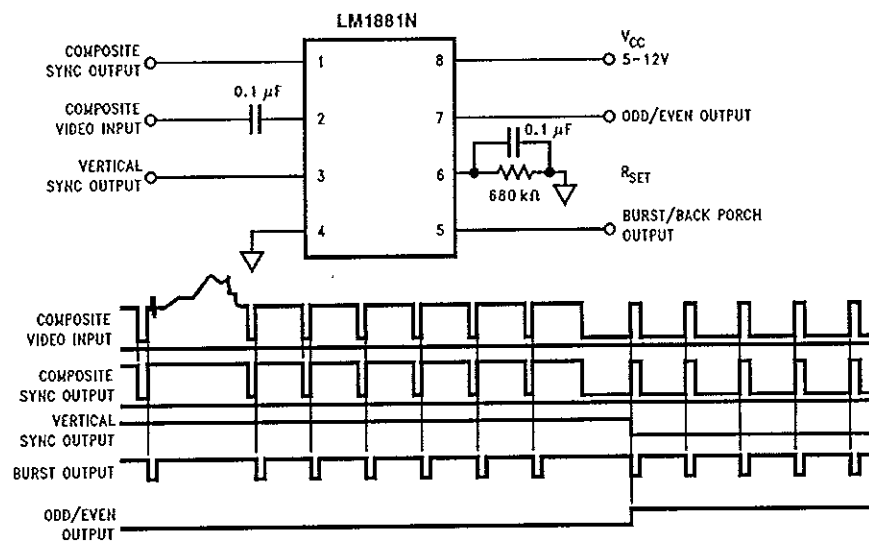
General Description

The LM1881 Video sync separator extracts timing information including composite and vertical sync, burst/back porch timing, and odd/even field information from standard negative going sync NTSC, PAL*, and SECAM video signals with amplitude from 0.5V to 2V p-p. The integrated circuit is also capable of providing sync separation for non-standard, faster horizontal rate video signals. The vertical output is produced on the rising edge of the first serration in the vertical sync period. A default vertical output is produced after a time delay if the rising edge mentioned above does not occur within the externally set delay period, such as might be the case for a non-standard video signal.

Features

- AC coupled composite input signal
- >10 k Ω input resistance
- <10 mA power supply drain current
- Composite sync and vertical outputs
- Odd/even field output
- Burst gate/back porch output
- Horizontal scan rates to 150 kHz
- Edge triggered vertical output
- Default triggered vertical output for non-standard video signal (video games-home computers)

Connection Diagram



Order Number LM1881M or LM1881N
See NS Package Number M08A or N08E

TL/H/9150-1

*PAL in this datasheet refers to European broadcast TV standard "Phase Alternating Line", and not to Programmable Array Logic.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	13.2V
Input Voltage	3 Vpp ($V_{CC} = 5V$) 6 Vpp ($V_{CC} \geq 8V$)
Output Sink Currents; Pins 1, 3, 5	5 mA
Output Sink Current; Pin 7	2 mA
Package Dissipation (Note 1)	1100 mW
Operating Temperature Range	0°C – 70°C

Storage Temperature Range	-65°C to +150°C
ESD Susceptibility (Note 2)	2 kV
Soldering Information	
Dual-In-Line Package (10 sec.)	260°C
Small Outline Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

See AN-450 "Surface Mounting Methods and their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics

$V_{CC} = 5V$; $R_{set} = 680\text{ k}\Omega$; $T_A = 25^\circ\text{C}$; Unless otherwise specified

Parameter	Conditions	Typ	Tested Limit (Note 3)	Design Limit (Note 4)	Units (Limits)
Supply Current	Outputs at Logic 1 $V_{CC} = 5V$ $V_{CC} = 12V$	5.2	10		mAmax
		5.5	12		mAmax
DC Input Voltage	Pin 2	1.5	1.3 1.8		Vmin Vmax
Input Threshold Voltage	Note 5	70	55 85		mVmin mVmax
Input Discharge Current	Pin 2; $V_{IN} = 2V$	11	6 16		μAmin μAmax
Input Clamp Charge Current	Pin 2; $V_{IN} = 1V$	0.8	0.2		mAmin
R_{SET} Pin Reference Voltage	Pin 6; Note 6	1.22	1.10 1.35		Vmin Vmax
Composite Sync. & Vertical Outputs	$I_{OUT} = 40\ \mu\text{A}$; Logic 1 $I_{OUT} = 1.6\ \text{mA}$; Logic 1	$V_{CC} = 5V$ $V_{CC} = 12V$	4.5	4.0 11.0	Vmin Vmin
		$V_{CC} = 5V$ $V_{CC} = 12V$	3.6	2.4 10.0	Vmin Vmin
Burst Gate & Odd/Even Outputs	$I_{OUT} = 40\ \mu\text{A}$; Logic 1	4.5	4.0 11.0		Vmin Vmin
Composite Sync. Output	$I_{OUT} = -1.6\ \text{mA}$; Logic 0; Pin 1	0.2	0.8		Vmax
Vertical Sync. Output	$I_{OUT} = -1.6\ \text{mA}$; Logic 0; Pin 3	0.2	0.8		Vmax
Burst Gate Output	$I_{OUT} = -1.6\ \text{mA}$; Logic 0; Pin 5	0.2	0.8		Vmax
Odd/Even Output	$I_{OUT} = -1.6\ \text{mA}$; Logic 0; Pin 7	0.2	0.8		Vmax
Vertical Sync Width		230	190 300		μsmin μsmax
Burst Gate Width	2.7 k Ω from Pin 5 to V_{CC}	4	2.5 4.7		μsmin μsmax
Vertical Default Time	Note 7	65	32 90		μsmin μsmax

Note 1: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and a package thermal resistance of 110° C/W, junction to ambient.

Note 2: ESD susceptibility test uses the "human body model, 100 pF discharged through a 1.5 k Ω resistor".

Note 3: Typicals are at $T_J = 25^\circ\text{C}$ and represent the most likely parametric norm.

Note 4: Tested Limits are guaranteed to National's AOQL (Average Outgoing Quality Level).

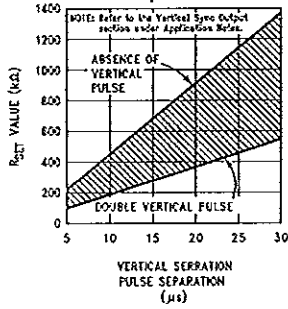
Note 5: Relative difference between the input clamp voltage and the minimum input voltage which produces a horizontal output pulse.

Note 6: Careful attention should be made to prevent parasitic capacitance coupling from any output pin (Pins 1, 3, 5, and 7) to the R_{SET} pin (Pin 6).

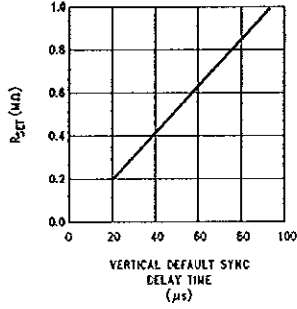
Note 7: Delay time between the start of vertical sync (at input) and the vertical output pulse.

Typical Performance Characteristics

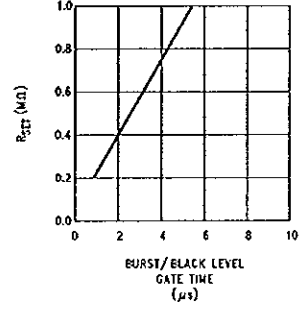
R_{set} Value Selection vs Vertical Serration Pulse Separation



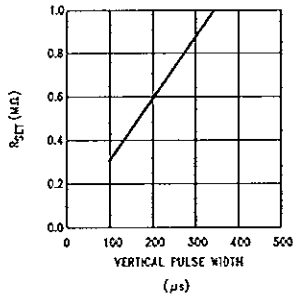
Vertical Default Sync Delay Time vs Rset



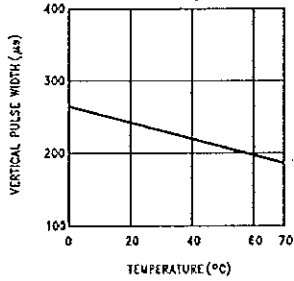
Burst/Black Level Gate Time vs Rset



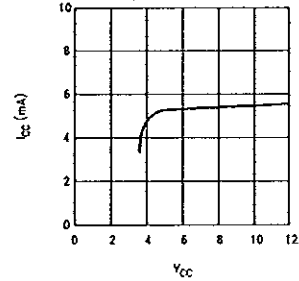
Vertical Pulse Width vs Rset



Vertical Pulse Width vs Temperature



Supply Current vs Supply Voltage



TL/H/9150-2

Application Notes

The LM1881 is designed to strip the synchronization signals from composite video sources that are in, or similar to, the N.T.S.C. format. Input signals with positive polarity video (increasing signal voltage signifies increasing scene brightness) from 0.5V (p-p) to 2V (p-p) can be accommodated. The LM1881 operates from a single supply voltage between 5V DC and 12V DC. The only required external components beside power supply and set current decoupling are the input coupling capacitor and a single resistor that sets internal current levels, allowing the LM1881 to be adjusted for source signals with line scan frequencies differing from 15.734 kHz. Four major sync signals are available from the I/C: composite sync including both horizontal and vertical scan timing information; a vertical sync pulse; a burst gate or back porch clamp pulse; and an odd/even output. The odd/even output level identifies which video field of an interlaced video source is present at the input. The outputs from the LM1881 can be used to gen-lock video camera/VTR signals with graphics sources, provide identification of video fields for memory storage, recover suppressed or contaminated sync signals, and provide timing references for the extraction of coded or uncoded data on specific video scan lines.

To better understand the LM1881 timing information and the type of signals that are used, refer to *Figure 2(a-e)* which shows a portion of the composite video signal from the end of one field through the beginning of the next field.

COMPOSITE SYNC OUTPUT

The composite sync output, *Figure 2(b)*, is simply a reproduction of the signal waveform below the composite video black level, with the video completely removed. This is obtained by clamping the video signal sync tips to 1.5V DC at Pin 2 and using a comparator threshold set just above this voltage to strip the sync signal, which is then buffered out to Pin 1. The threshold separation from the clamped sync tip is nominally 70 mV which means that for the minimum input level of 0.5V (p-p), the clipping level is close to the halfway point on the sync pulse amplitude (shown by the dashed line on *Figure 2(a)*). This threshold separation is independent of the signal amplitude, therefore, for a 2V (p-p) input the clipping level occurs at 11% of the sync pulse amplitude. The charging current for the input coupling capacitor is 0.8 mA, whereas the discharge current is only 11 μ A, typically. This allows relatively small capacitor values to be used—0.1 μ F is generally recommended.

Normally the signal source for the LM1881 is assumed to be clean and relatively noise-free, but some sources may have excessive video peaking, causing high frequency video and chroma components to extend below the black level reference. Some video discs keep the chroma burst pulse present throughout the vertical blanking period so that the burst actually appears on the sync tips for three line periods instead of at black level. A clean composite sync signal can be generated from these sources by filtering the input signal. When the source impedance is low, typically 75 Ω , a 620 Ω resistor in series with the source and a 510 pF capacitor to ground will form a low pass filter with a corner frequency of 500 kHz. This bandwidth is more than sufficient to pass the sync pulse portion of the waveform; however, any subcarrier content in the signal will be attenuated by almost 18 dB, effectively taking it below the comparator threshold. Filtering will also help if the source is contaminated with thermal noise. The output waveforms will become delayed

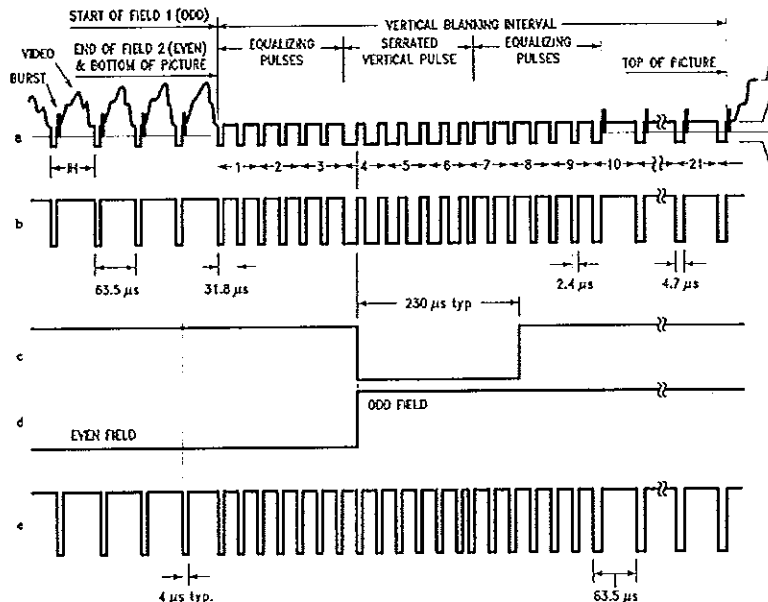
from between 40 ns to as much as 200 ns due to this filter. This much delay will not usually be significant but it does contribute to the sync delay produced by any additional signal processing. Since the original video may also undergo processing, the need for time delay correction will depend on the total system, not just the sync stripper.

VERTICAL SYNC OUTPUT

A vertical sync output is derived by internally integrating the composite sync waveform (*Figure 3*). To understand the generation of the vertical sync pulse, refer to the lower left hand section *Figure 3*. Note that there are two comparators in the section. One comparator has an internally generated voltage reference called V_1 going to one of its inputs. The other comparator has an internally generated voltage reference called V_2 going to one of its inputs. Both comparators have a common input at their noninverting input coming from the internal integrator. The internal integrator is used for integrating the composite sync signal. This signal comes from the input side of the composite sync buffer and are positive going sync pulses. The capacitor to the integrator is internal to the LM1881. The capacitor charge current is set by the value of the external resistor R_{set} . The output of the integrator is going to be at a low voltage during the normal horizontal lines because the integrator has a very short time to charge the capacitor, which is during the horizontal sync period. The equalization pulses will keep the output voltage of the integrator at about the same level, below the V_1 . During the vertical sync period the narrow going positive pulses shown in *Figure 2* is called the serration pulse. The wide negative portion of the vertical sync period is called the vertical sync pulse. At the start of the vertical sync period, before the first Serration pulse occurs, the integrator now charges the capacitor to a much higher voltage. At the first serration pulse the integrator output should be between V_1 and V_2 . This would give a high level at the output of the comparator with V_1 as one of its inputs. This high is clocked into the "D" flip-flop by the falling edge of the serration pulse (remember the sync signal is inverted in this section of the LM1881). The "Q" output of the "D" flip-flop goes through the OR gate, and sets the R/S flip-flop. The output of the R/S flip-flop enables the internal oscillator and also clocks the ODD/EVEN "D" flip-flop. The ODD/EVEN field pulse operation is covered in the next section. The output of the oscillator goes to a divide by 8 circuit, thus resetting the R/S flip-flop after 8 cycles of the oscillator. The frequency of the oscillator is established by the internal capacitor going to the oscillator and the external R_{set} . The "Q" output of the R/S flip-flop goes to pin 3 and is the actual vertical sync output of the LM1881. By clocking the "D" flip-flop at the start of the first serration pulse means that the vertical sync output pulse starts at this point in time and lasts for eight cycles of the internal oscillator as shown in *Figure 2*.

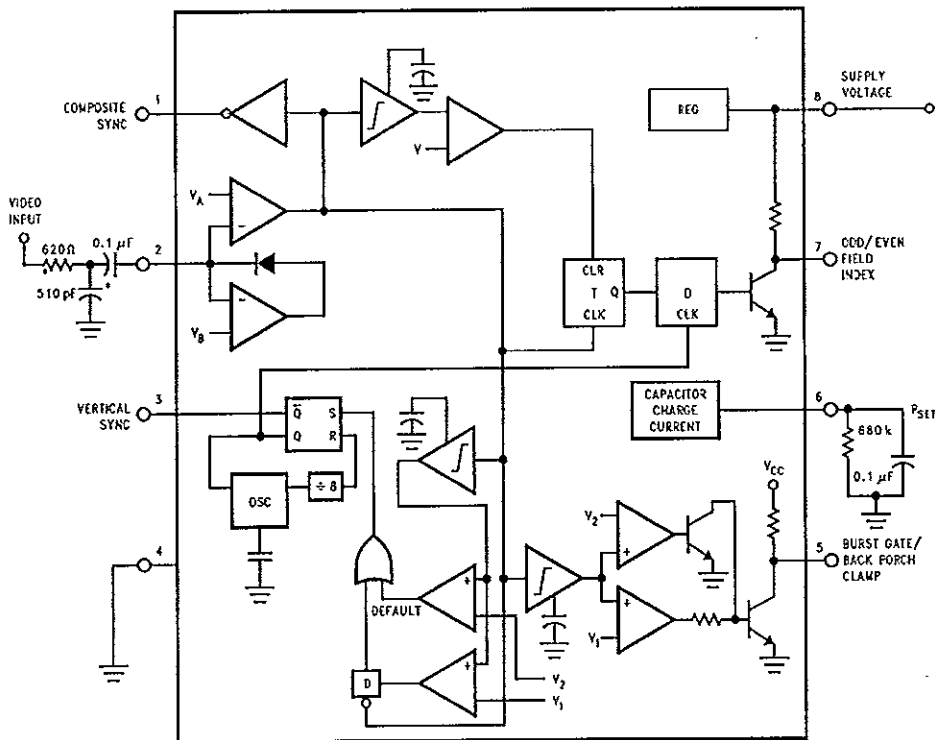
How R_{set} affects the integrator and the internal oscillator is shown under the Typical Performance Characteristics. The first graph is " R_{set} Value Selection vs Vertical Serration Pulse Separation". For this graph to be valid, the vertical sync pulse should last for at least 85% of the horizontal half line (47% of a full horizontal line). A vertical sync pulse from any standard should meet this requirement; both NTSC and PAL do meet this requirement (the serration pulse is the remainder of the period, 10% to 15% of the horizontal

Application Notes (Continued)



TL/H/9150-3

FIGURE 2. (a) Composite Video; (b) Composite Sync; (c) Vertical Output Pulse; (d) Odd/Even Field Index; (e) Burst Gate/Back Porch Clamp



*Components Optional,
See Text

TL/H/9150-4

FIGURE 3

Application Notes (Continued)

half line). Remember this pulse is a positive pulse at the integrator but negative in *Figure 2*. This graph shows how long it takes the integrator to charge its internal capacitor above V_1 .

WITH R_{set} too large the charging current of the integrator will be too small to charge the capacitor above V_1 , thus there will be no vertical sync output pulse. As mentioned above, R_{set} also sets the frequency of the internal oscillator. If the oscillator runs too fast its eight cycles will be shorter than the vertical sync portion of the composite sync. Under this condition another vertical sync pulse can be generated on one of the later serration pulses after the divide by 8 circuit resets the R/S flip-flop. The first graph also shows the minimum R_{set} necessary to prevent a double vertical pulse, assuming that the serration pulses last for only three full horizontal line periods (six serration pulses for NTSC). The actual pulse width of the vertical sync pulse is shown in the "Vertical Pulse Width vs R_{set} " graph. Using NTSC as an example, lets see how these two graphs relate to each other. The Horizontal line is 64 μs long, or 32 μs for a horizontal half line. Now round this off to 30 μs . In the " R_{set} Value Selection vs Vertical Serration Pulse Separation" graph the minimum resistor value for 30 μs serration pulse separation is about 550 k Ω . Going to the "Vertical Pulse Width vs R_{set} " graph one can see that 550 k Ω gives a vertical pulse width of about 180 μs , the total time for the vertical sync period of NTSC (3 horizontal lines). A 550 k Ω will set the internal oscillator to a frequency such that eight cycles gives a time of 180 μs , just long enough to prevent a double vertical sync pulse at the vertical sync output of the LM1881.

The LM1881 also generates a default vertical sync pulse when the vertical sync period is unusually long and has no serration pulses. With a very long vertical sync time the integrator has time to charge its internal capacitor above the voltage level V_2 . Since there is no falling edge at the end of a serration pulse to clock the "D" flip-flop, the only high signal going to the OR gate is from the default comparator when output of the integrator reaches V_2 . At this time the R/S flip-flop is toggled by the default comparator, starting the vertical sync pulse at pin 3 of the LM1881. If the default vertical sync period ends before the end of the input vertical sync period, then the falling edge of the vertical sync (positive pulse at the "D" flip-flop) will clock the high output from the comparator with V_1 as a reference input. This will retrigger the oscillator, generating a second vertical sync output pulse. The "Vertical Default Sync Delay Time vs R_{set} " graph shows the relationship between the R_{set} value and the delay time from the start of the vertical sync period before the default vertical sync pulse is generated. Using the NTSC example again the smallest resistor for R_{set} is 500 k Ω . The vertical default time delay is about 50 μs , much longer than the 30 μs serration pulse spacing.

A common question is how can one calculate the required R_{set} with a video timing standard that has no serration pulses during the vertical blanking. If the default vertical sync is to be used this is a very easy task. Use the "Vertical Default

Sync Delay Time vs R_{set} " graph to select the necessary R_{set} to give the desired delay time for the vertical sync output signal. If a second pulse is undesirable, then check the "Vertical Pulse Width vs R_{set} " graph to make sure the vertical output pulse will extend beyond the end of the input vertical sync period. In most systems the end of the vertical sync period may be very accurate. In this case the preferred design may be to start the vertical sync pulse at the end of the vertical sync period, similar to starting the vertical sync pulse after the first serration pulse. A VGA standard is to be used as an example to show how this is done. In this standard a horizontal line is 32 μs long. The vertical sync period is two horizontal lines long, or 64 μs . The vertical default sync delay time must be longer than the vertical sync period of 64 μs . In this case R_{set} must be larger than 680 k Ω . R_{set} must still be small enough for the output of the integrator to reach V_1 before the end of the vertical period of the input pulse. The first graph can be used to confirm that R_{set} is small enough for the integrator. Instead of using the vertical serration pulse separation, use the actual pulse width of the vertical sync period, or 64 μs in this example. This graph is linear, meaning that a value as large as 2.7 M Ω can be used for R_{set} (twice the value as the maximum at 30 μs). Due to leakage currents it is advisable to keep the value of R_{set} under 2.0 M Ω . In this example a value of 1.0 M Ω is selected, well above the minimum of 680 k Ω . With this value for R_{set} the pulse width of the vertical sync output pulse of the LM1881 is about 340 μs .

ODD/EVEN FIELD PULSE

An unusual feature of LM1881 is an output level from Pin 7 that identifies the video field present at the input to the LM1881. This can be useful in frame memory storage applications or in extracting test signals that occur only in alternate fields. For a composite video signal that is interlaced, one of the two fields that make up each video frame or picture must have a half horizontal scan line period at the end of the vertical scan—i.e., at the bottom of the picture. This is called the "odd field" or "field 1". The "even field" or "field 2" has a complete horizontal scan line at the end of the field. An odd field starts on the leading edge of the first equalizing pulse, whereas the even field starts on the leading edge of the second equalizing pulse of the vertical retrace interval. *Figure 2(a)* shows the end of the even field and the start of the odd field.

To detect the odd/even fields the LM1881 again integrates the composite sync waveform (*Figure 3*). A capacitor is charged during the period between sync pulses and discharged when the sync pulse is present. The period between normal horizontal sync pulses is enough to allow the capacitor voltage to reach a threshold level of a comparator that clears a flipflop which is also being clocked by the sync waveform. When the vertical interval is reached, the shorter integration time between equalizing pulses prevents this

Application Notes (Continued)

threshold from being reached and the Q output of the flip-flop is toggled with each equalizing pulse. Since the half line period at the end of the odd field will have the same effect as an equalizing pulse period, the Q output will have a different polarity on successive fields. Thus by comparing the Q polarity with the vertical output pulse, an odd/even field index is generated. Pin 7 remains low during the even field and high during the odd field.

BURST/BACKPORCH OUTPUT PULSE

In a composite video signal, the chroma burst is located on the backporch of the horizontal blanking period. This period, approximately 4.8 μ s long, is also the black level reference for the subsequent video scan line. The LM1881 generates a pulse at Pin 5 that can be used either to retrieve the chroma burst from the composite video signal (thus providing a subcarrier synchronizing signal) or as a clamp for the DC restoration of the video waveform. This output is obtained simply by charging an internal capacitor starting on the trailing edge of the horizontal sync pulses. Simultaneously the output of Pin 5 is pulled low and held until the capacitor charge circuit times out—4 μ s later. A shorter output burst gate pulse can be derived by differentiating the burst output using a series C-R network. This may be necessary in applications which require high horizontal scan rates in combination with normal (60–120 Hz) vertical scan rates.

APPLICATIONS

Apart from extracting a composite sync signal free of video information, the LM1881 outputs allow a number of interesting applications to be developed. As mentioned above, the burst gate/backporch clamp pulse allows DC restoration of the original video waveform for display or remodulation on an R.F. carrier, and retrieval of the color burst for color synchronization and decoding into R.G.B. components. For frame memory storage applications, the odd/even field level allows identification of the appropriate field ensuring the correct read or write sequence. The vertical pulse output is particularly useful since it begins at a precise time—the rising edge of the first vertical serration in the sync waveform. This means that individual lines within the vertical blanking period (or anywhere in the active scan line period) can easily be extracted by counting the required number of transitions in the composite sync waveform following the start of the vertical output pulse.

The vertical blanking interval is proving popular as a means to transmit data which will not appear on a normal T.V. receiver screen. Data can be inserted beginning with line 10 (the first horizontal scan line on which the color burst appears) through to line 21. Usually lines 10 through 13 are not used which leaves lines 14 through 21 for inserting signals, which may be different from field to field. In the U.S., line 19 is normally reserved for a vertical interval reference

signal (VIRS) and line 21 is reserved for closed caption data for the hearing impaired. The remaining lines are used in a number of ways. Lines 17 and 18 are frequently used during studio processing to add and delete vertical interval test signals (VITS) while lines 14 through 18 and line 20 can be used for Videotex/Teletext data. Several institutions are proposing to transmit financial data on line 17 and cable systems use the available lines in the vertical interval to send decoding data for descrambler terminals.

Since the vertical output pulse from the LM1881 coincides with the leading edge of the first vertical serration, sixteen positive or negative transitions later will be the start of line 14 in either field. At this point simple counters can be used to select the desired line(s) for insertion or deletion of data.

VIDEO LINE SELECTOR

The circuit in *Figure 4* puts out a single video line according to the binary coded information applied to line select bits b0–b7. A line is selected by adding two to the desired line number, converting to a binary equivalent and applying the result to the line select inputs. The falling edge of the LM1881's vertical pulse is used to load the appropriate number into the counters (MM74C193N) and to set a start count latch using two NAND gates. Composite sync transitions are counted using the borrow out of the desired number of counters. The final borrow out pulse is used to turn on the analog switch (CD4066BC) during the desired line. The falling edge of this signal also resets the start count latch, thereby terminating the counting.

The circuit, as shown, will provide a single line output for each field in an interlaced video system (television) or a single line output in each frame for a non-interlaced video system (computer monitor). When a particular line in only one field of an interlaced video signal is desired, the odd/even field index output must be used instead of the vertical output pulse (invert the field index output to select the odd field). A single counter is needed for selecting lines 3 to 14; two counters are needed for selecting lines 15 to 253; and three counters will work for up to 2046 lines. An output buffer is required to drive low impedance loads.

MULTIPLE CONTIGUOUS VIDEO LINE SELECTOR WITH BLACK LEVEL RESTORATION

The circuit in *Figure 5* will select a number of adjoining lines starting with the line selected as in the previous example. Additional counters can be added as described previously for either higher starting line numbers or an increased number of contiguous output lines. The back porch pulse output of the LM1881 is used to gate the video input's black level through a low pass filter (10 k Ω , 10 μ F) providing black level restoration at the video output when the output selected line(s) is not being gated through.

Typical Applications

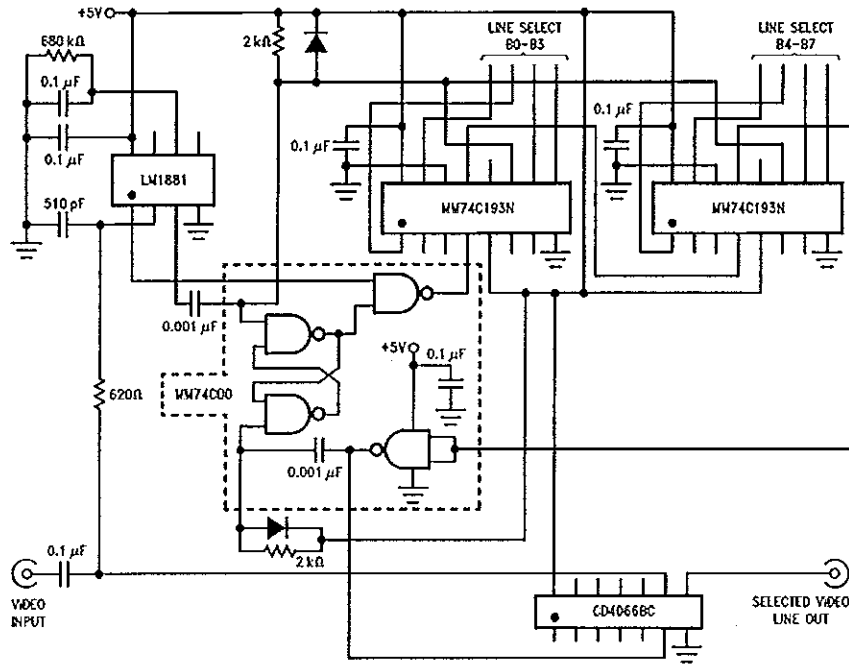


FIGURE 4. Video Line Selector

TL/H/9150-5

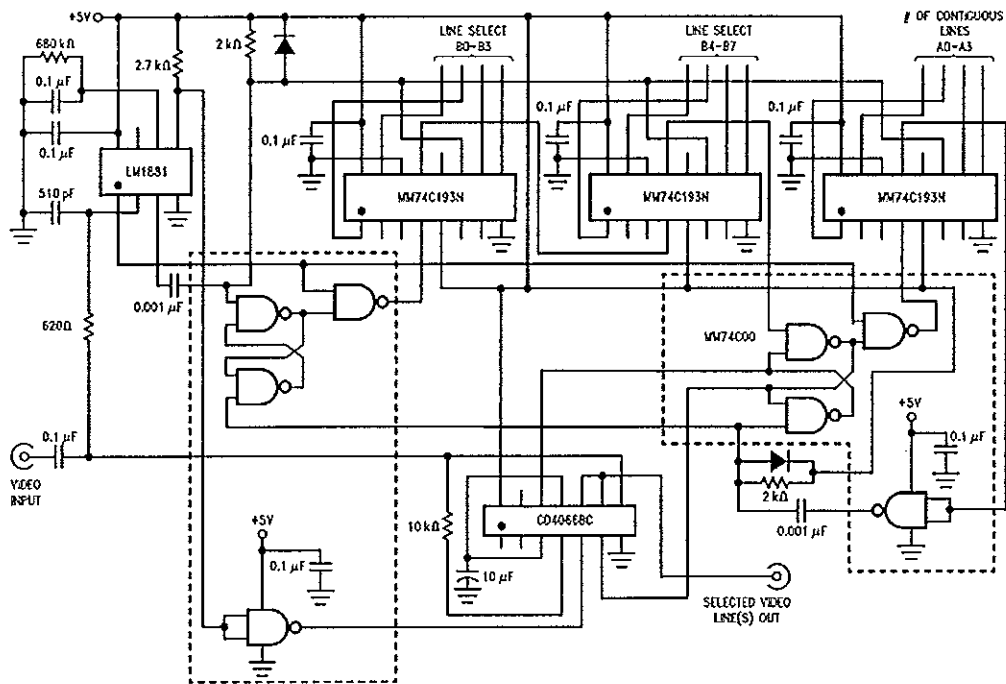


FIGURE 5. Multiple Contiguous Video Line Selector With Black Level Restoration

TL/H/9150-6



MP8780
 CMOS
 8-Bit Video
 Analog-to-Digital Converter

FEATURES

- 15 MHz Input Bandwidth (-0.3 dB)
- SNR > 44 dB @ Fin 2.4 MHz
- 1.2 to 5.0 Volts (Peak to Peak) Input Range
- 1/2 LSB Dynamic DNL at 14.4 MHz
- 3/4 LSB Dynamic DNL at 17.7 MHz
- Monotonic, No Missing Codes
- Latch Up Free CMOS Technology
- High ESD Protector - 2000 Volts Minimum

BENEFITS

- Optimized Combination of Performance, Power, Packaging and Cost for Video Digitizing Applications
- Excellent Video Digitizing Performance
- Smaller Board Space
- Lower System Power

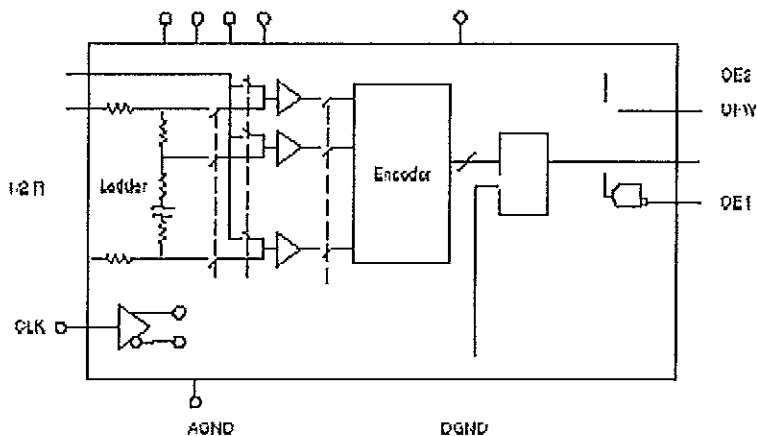
GENERAL DESCRIPTION

The MP8780 is a CMOS 8-bit high speed Analog-to-Digital Converter designed and specified for applications in imaging and video digitizing. With Signal to Noise Ratio greater than 44 dB in the Video Input Bandwidth and encode rates up to 20 MHz, the MP8780 easily meets the requirements needed to digitize standard American and European video signals.

A fast digital interface simplifies connection to most modern DSP and CPU chips.

Careful design and layout have reduced static sensitivity, while our proprietary latch-up free process virtually eliminates the need for many of the diode protection schemes used in the past.

SIMPLIFIED BLOCK AND TIMING DIAGRAM



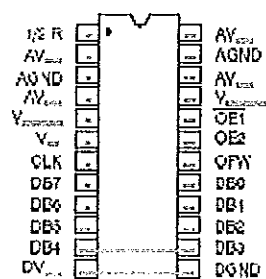
MP8780



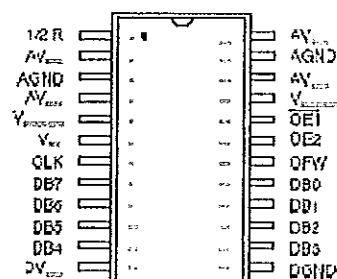
ORDERING INFORMATION

Package Type	Temperature Range	Part No.	DNL (LSB)	INL (LSB)
PDIP	-40 to +65°C	MP8780JN	±1	1/2
SOIC	-40 to +65°C	MP8780JS	±1	1/2

PIN CONFIGURATIONS *See Packaging Section for Package Dimensions*



24 Pin PDIP (0.300")



24 Pin SOIC (Jedec, 0.300")

PIN OUT DEFINITIONS

Pin No.	Pin Name	Pin Description
1	1/2R	50% Point of Reference on Resistance Ladder
2	AV _{cc}	Analog Power Supply Voltage
3	AGND	Analog Ground
4	AV _{cc}	Analog Power Supply Voltage
5	V _{ref}	Lower Reference Voltage Input
6	V _{in}	Analog Input Voltage
7	CLK	Sampling Clock Input
8	DB7	Data Output Bit 7 (MSB)
9	DB6	Data Output Bit 6
10	DB5	Data Output Bit 5
11	DB4	Data Output Bit 4
12	DV _{cc}	Digital Power Supply Voltage

Pin No.	Pin Name	Pin Description
13	DGND	Digital Ground
14	DB0	Data Output Bit 0
15	DB2	Data Output Bit 2
16	DB1	Data Output Bit 1
17	DB0	Data Output Bit 0 (LSB)
18	OFW	Overflow flag
19	OE2	Output Enable Control Pin
20	OE1	Output Enable Control Pin
21	V _{ref}	Upper Reference Voltage Input
22	AV _{cc}	Analog Power Supply Voltage
23	AGND	Analog Ground
24	AV _{cc}	Analog Power Supply Voltage



MP8780

ELECTRICAL CHARACTERISTICS

Specified: $AV_{DD} = DV_{DD} = 5\text{ V}$, $F_s = 14.4\text{ MHz}$ (Duty Cycle: 1/3 Sample & 2/3 Balance)
 $V_{REF(+)} = 2.5\text{ V}$, $V_{REF(-)} = \text{GND}$, Temp = 25°C

		MP8780J			
RESOLUTION		9		Bits	
Maximum Sampling Rate	FS	14.4		MHz	
ACCURACY:					
Differential Non-Linearity	DNL	1/2		LSB	
Integral Non-Linearity	INL	1		LSB	
Zero Scale Error	LS	-27		mV	
Full Scale Error	FS	12		mV	
DNL ¹	DNL	1/4		LSB	
INL ¹	INL	3/4		LSB	
				$F_{IN} = 100\text{ kHz}$ $F_{IN} = 10\text{ MHz}$	
DYNAMIC ACCURACY:					
(Histogram Test)					
Differential Non-Linearity		1/2		LSB	
Differential Non-Linearity		1/2		LSB	
Differential Non-Linearity		3/4		LSB	
Differential Non-Linearity		3/4		LSB	
				F_{IN}	F_{IN}
				100kHz	10MHz
				2.0	2.0
				2.5	2.5
				4.4	4.4
				2.0	2.5
REFERENCE VOLTAGES					
Positive Ref. Voltage	$V_{REF(+)}$	$V_{REF(+)}$		V	
Negative Ref. Voltage	$V_{REF(-)}$	GND		V	
Differential Ref. Voltage	$V_{REF(DIFF)}$	20		mV	
Ladder Resistance	R_{LDR}	100		Ω	
Ladder Temp. Coefficient ¹	R_{LDR}	225		ppm/°C	
ANALOG INPUT:					
Input Voltage Range	V_{IN}	$V_{REF(+)}$		V	
Input Impedance (See Figure 4)	Z_{IN}	50		Ω	
Aperture					
Aperture Delay	t_{APD}	10		ns	
Aperture Uncertainty	t_{APU}	45		ps	
Clock Slewback Pulse		10		pA	
Input Bandwidth (-0.5dB)	BW	20		MHz	
				$V_{REF(+)} = 2.5\text{ V}$ $V_{REF(-)} = 4.0\text{ V}$	
DIGITAL INPUTS: (Tmin to Tmax)					
Logical '1' Voltage	V_{OH}	20		V	
Logical '0' Voltage	V_{OL}	0.6		V	
Current (CLK)		-100		μA	
Current (VDET: Res to GND)		-5		μA	
Current (OE2: Res to V_{DD})		-30		μA	
Input Capacitance ¹	C_{IN}	5		pF	
Clock Timing (See Figure 1):					
Rise & Fall Time ¹	$t_{R/F}$	6		ns	
'High' Time (Autozero/Autobalance)	t_H	20		ns	
'Low' Time (Sampling)	t_L	20		ns	

MP8780



ELECTRICAL CHARACTERISTICS TABLE, (CONT'D)

Description	Symbol	Min	MP8780J Typ	Max	Units	Conditions
DIGITAL OUTPUTS						
Logic '1' Voltage	V_{OH}	$V_{OH} - 0.5$			V	$C_{load} = 15 \text{ pF}$ $V_{OH} = 0.1 \times V_{DD}$
Logic '1' Source Current	I_{OH}	2			mA	
Logic '0' Voltage	V_{OL}			0.4	V	
Logic '0' Sink Current	I_{OL}	4			mA	
Quiescent Leakage	I_{LQ}	-10		10	μA	
Min Data Hold Time (See Figure 1)*	t_{HD}	12	15		ns	
Max Data Valid Delay*	t_{VLD}		30	33	ns	
Data Enable Delay (See Figure 2)*	t_{DE}			20	ns	
Data Tristate Delay*	t_{DT}			20	ns	
POWER SUPPLIES						
Operating Voltage	V_{DD}	4		6	V	
Operating Current	I_{DD}		30	50	mA	
AC PARAMETERS*						
Differential Gain Error			2		%	$F_{max} = 2 \times \text{HTSC}$
Differential Phase Error			1		Degrees	
Signal Noise Ratio (RMS, RMS)	SNR				dB	$F_{max} = \frac{F_{min}}{100 \text{ Hz}}$ $F_{min} = 10 \text{ Hz}$
			46		dB	
			44		dB	17.7

- * Linearity (DNL, INL) is a function of clock frequency. INL is specified as the best straight line fit. See characterization chart.
- ** See V_{OH} Input Equivalence Circuit (Figure 4). Switched capacitor analog input requires input buffer with lowest output resistance possible.
- ** All inputs have current sources to V_{DD} and GND. OES has pull-up transistors. OET has pull-down transistors. These DC currents will not exceed the specified values for any input voltage between 0 and V_{DD} .
- ** CLK input spec to meet MP8780 operating specifications. Actual differential timing can be less stringent with no loss of accuracy.
- ** Specified values guarantee functional device. Refer to other parameters for accuracy.
- ** Values are typical. Not guaranteed tested.

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS (TA = +25°C unless otherwise noted)^{1, 2}

V_{DD} to GND	±7 V	Storage Temperature	-65 to +150°C
$V_{REF(1)}$ & $V_{REF(2)}$	GND -0.5 to V_{DD} 10.5 V	Lead Temperature (Soldering 10 seconds)	1000°C
V_{IN}	GND -0.5 to V_{DD} 10.5 V	Package Power Dissipation Rating to 75°C	
All Inputs	GND -0.5 to V_{DD} 10.5 V	PD P, SO C	1000mW
All Outputs	GND -0.5 to V_{DD} 10.5 V	Derates above 75°C	10mW/°C

Notes:

- ¹ Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation at or above this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
- ² Any input pin which can see a voltage outside the absolute maximum ratings should be protected by Schottky diode clamps (1P0452-2630) from input pin to the supplies. All inputs have protection diodes which will protect the device from short transients outside the supplies of less than 100mA for less than 100µs.



MP8780

TIMING DIAGRAMS

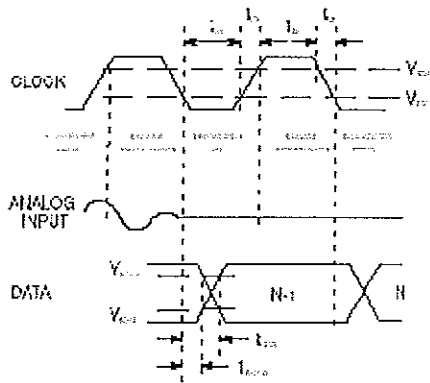


Figure 1. MP8780 Timing Diagram

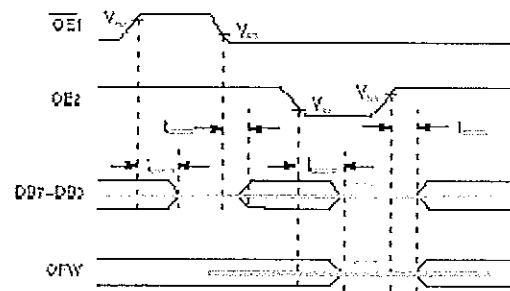


Figure 2. Output Enable/Disable Timing Diagram

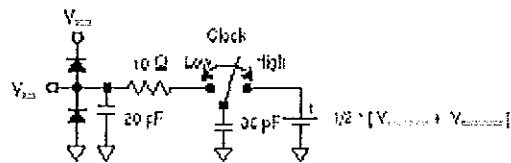


Figure 3. Analog Input Equivalent Circuit

MP8780



THEORY OF OPERATION

The MP8780 converts analog voltages into 256 digital codes by encoding the outputs of 255 comparators. A 256th comparator is used to generate the overflow bit. The conversion is synchronous with the sample clock. A complete conversion cycle is accomplished in 1.5 cycles. Data is transferred from the comparator latches to the output register each cycle at the same time the input is sampled.

The clock signal generates the two internal phases, ϕ_{1a} (CLK 1 q 1 = 0) and ϕ_{1b} (CLK 0 w = 0). Phase B connects the comparators to the reference (a) points. Phase S connects the comparators to the analog input voltage.

The reference resistance ladder is a series of 257 resistors. The first and the last resistor of the ladder are half the value of the others so that the following relations apply:

$$R_{128} = 256 \times R \quad V_{REF} = V_{REF(1)} = V_{REF(2)} = 256 \times \text{LSB}$$

The ideal transfer function for MP8780 is shown in Figure 4.

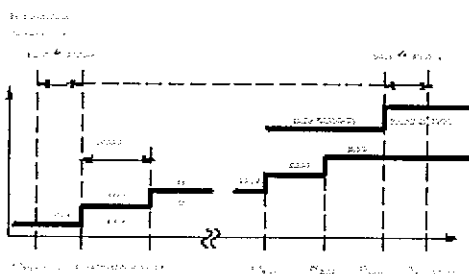


Figure 4. Ideal A/D Transfer Function

The overflow transition (V_{OFF}) takes place at:

$$V_{IH} = V_{OFF} = V_{REF(2)} = 0.5 \text{ LSB}$$

Thus the first and the last transitions for the data bits take place at:

$$V_{IH} = V_{01} = V_{REF(1)} + 0.5 \times \text{LSB}$$

$$V_{IL} = V_{FF} = V_{REF(1)} - 1.5 \times \text{LSB}$$

$$\text{LSB} = V_{REF}/256 = (V_{FF} - V_{01})/254$$

Note that the overflow transition is a lag and has no impact on the data bits.

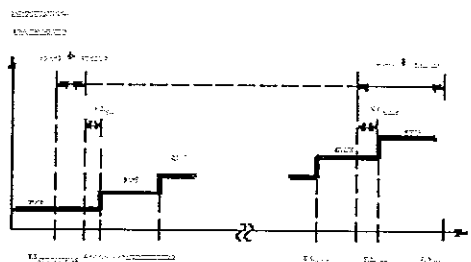


Figure 5. Real A/D Transfer Curve

The formulas define the various error relationships for Differential Non-linearity (DNL), Integral Non-Linearity (INL) and zero and full scale errors (E_{ZS} , E_{FS}).

$$\text{DNL}(i) = V_{02} - V_{01} - \text{LSB}$$

...

$$\text{DNL}(FE) = V_{FF} - V_{FE} - \text{LSB}$$

$$E_{FS} \text{ (full scale error)} = V_{FF} - [V_{REF(1)} - 1.5 \times \text{LSB}]$$

$$E_{ZS} \text{ (zero scale error)} = V_{01} - [V_{REF(1)} + 0.5 \times \text{LSB}]$$

$$\text{INL}(i) = \sum \text{DNL}(i)$$

Figure 5 shows the effect of the zero scale and full scale error terms.

Systems that equalize the V_{REF} voltages only increase the DNL accuracy at the two extreme points. In the MP8780, such adjustments have little impact at frequencies lower than 10 MHz. Refer to the characterization data for frequency dependence.

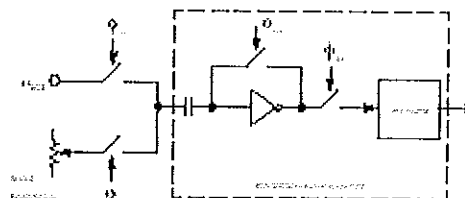


Figure 6. MP8780 Comparator

OE1	OE2	OPW	DB0-DB7
X	0	High Z	High Z
1	1	Valid	High Z
0	1	Valid	Valid

Table 1. Output Enable Logic

The MP8780 uses the balance phase (t_{BAL}) to charge one plate of the capacitors to the reference ladder tap point and the other to the zero/comparator trigger point. During the sample phase (t_{SAMP}), one plate of the capacitors switches to V_{IN} . The change in voltage ($V_{IN} - V_{REF}$) transfers across the capacitor and forces the inverter into one of the two possible logic states. A latch (connected to the comparator during t_{SAMP}) captures and propagates the digital level to the decode logic.

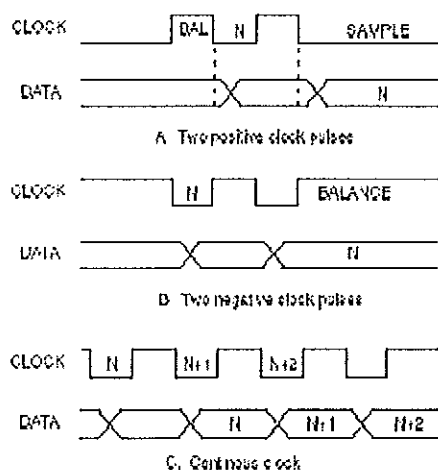


Figure 7. Relationship of Data to Clock

A system can clock the MP8780 continuously or it may choose to give clock pulses intermittently when a conversion is desired. The timing of Figure 7B, keeps the MP8780 comparators in balance and ready to sample the analog input. This mode draws the most current from V_{CC} . The timing of Figure 7A leaves the comparator inputs floating (and AC coupled to the V_{IN} input) and a balance phase is needed before a valid sampling phase. In this

mode, the rising edge of the CLK input marks the end of the sampling phase (t_{SAMP}). Internal delay of the clock circuitry will delay the actual instant when V_{IN} disconnects the latch from the comparator. This delay is called aperture delay (t_{AP}).

The aperture delay is not constant but changes from one cycle to the next. Internal thermal noise, power supply noise and slow input clock edges are major contributors to this variation. The aperture jitter (t_{AJ}) is the variation of the aperture delay distribution.

This uncertainty shows as digital code errors if the input slew rate multiplied by t_{AP} is of the same order of magnitude as the LSB. That is, if $(dV/dt) \cdot t_{AP} > V_{LSB}/255$ an internal 1 LSB error results.

The code encodes the 255 bits into a binary code and latches the code in a D-type flip-flop output. The inputs OE1 and OE2 control the output buffers in an asynchronous mode.

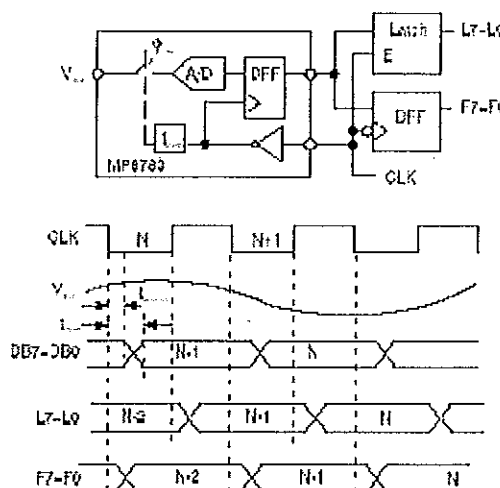


Figure 6. MP8780 Functional Equivalent Circuit and Interface Timing

mode, V_{IN} varies because of the floating comparator inputs.

This will work at any frequency. If the system must latch with the positive going edge, then care must be taken to avoid the overlap of the clock edge with the changing outputs. A slash follows the A/D; the positive half of the clock uses as enable e.g., guarantees stable output at the end of the enable pulse.

MP8780



CHARACTERIZATION CHARTS

Unless Otherwise Specified: $V_{DD} = 5\text{ V}$, $F_S = 14.4\text{ MHz}$ (50% Duty Cycle)

$V_{REF(+)} = 2.5\text{ V}$, $V_{REF(-)} = \text{GND}$, $T_{\text{emp}} = 25\text{ }^\circ\text{C}$

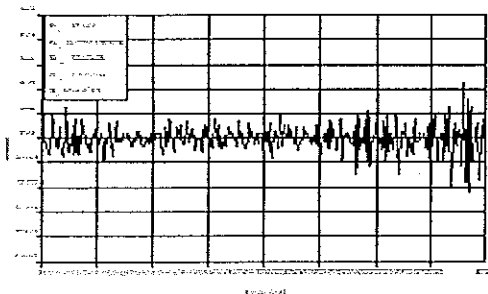


Figure 9. DNL ERROR PLOT

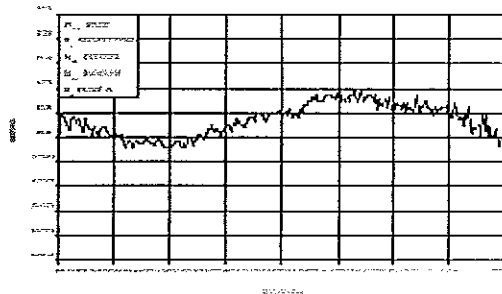


Figure 10. INL ERROR PLOT

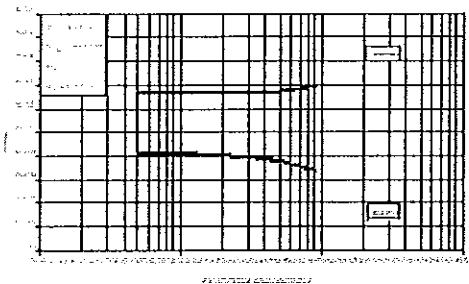


Figure 11. DNL vs FREQUENCY

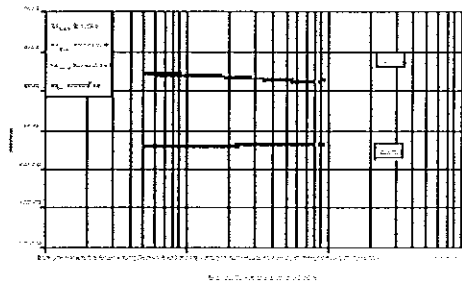


Figure 12. INL vs FREQUENCY

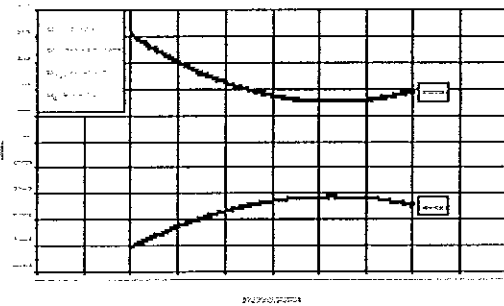


Figure 13. DNL vs VREF

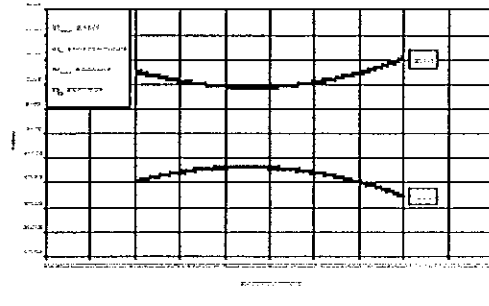


Figure 14. INL vs VREF

MP8780



APPLICATION NOTES

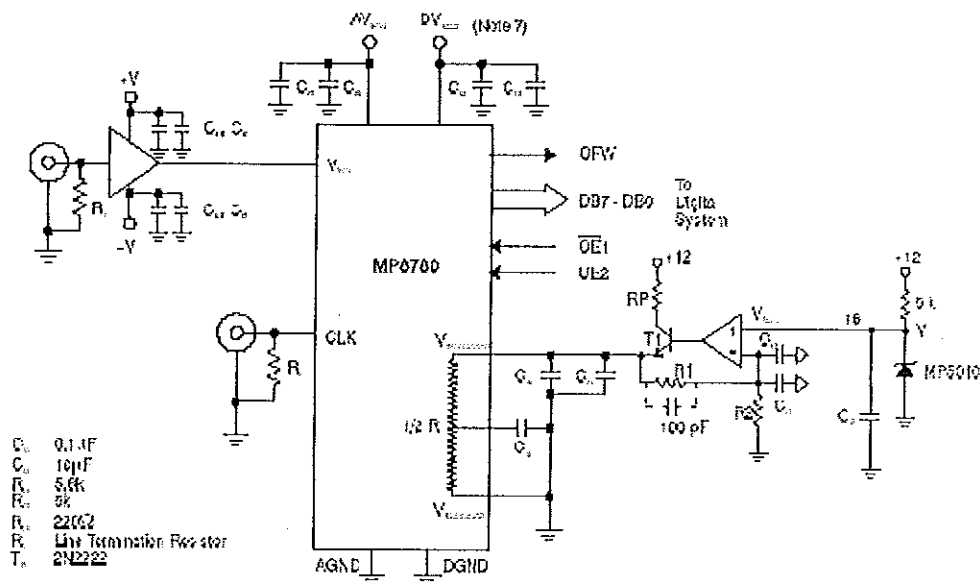


Figure 20. Typical Circuit Connections

The following information will be useful in maximizing the performance of the MP8780.

1. No signals should exceed $V_{cc} + 0.5$ V or $GND - 0.5$ V.
2. Any input pin which can see a signal below GND or above V_{cc} should be protected by diode clamps (1N4148 or 1P5062-2635) from input pins to the supplies. All MP8780 inputs have input protection diodes which will protect the device from short transients outside the supply range.
3. The design of a PCB board will affect the accuracy of MP8780. Use of wire wrap is not recommended.
4. The analog input signal (V_{in}) is quite sensitive and should be properly routed and terminated. It should be shielded from the clock and digital outputs so as to minimize cross coupling and noise pickup.
5. The analog input should be driven by a low impedance (less than 60 Ω).
6. Analog and digital ground planes should be substantial and common at one point only. The ground plane should act as a shield for parallels and not a return path for signals. Separate low impedance ground paths will reduce noise levels.
7. DV_{cc} should not be shared with other digital circuitry to avoid conversion errors caused by digital supply transients. In case a separate DV_{cc} for the MP8780 cannot be provided, then DV_{cc} should be connected to AV_{cc} next to the MP8780.
8. Each power supply and reference voltage pin should be decoupled with a ceramic (0.1 μ F) and a tantalum (10 μ F) capacitor as close to the device as possible.
9. The digital output should not be driving long wires as the capacitive coupling and reflection will contribute noise to the conversion. When driving distant loads, buffers should be used.

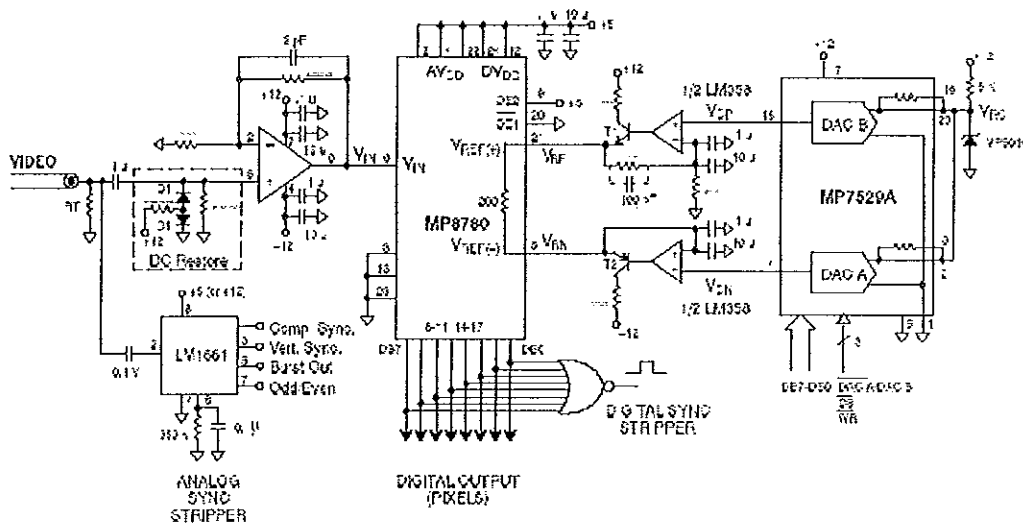


Figure 21. Video Digitizer

- D: 1N4148 (any diode would do)
- T: 2N1933, 2N1934
- T: 2N1933, 2N1934
- U: EL2050 (others under test)
- R: 75Ω
- R_{AV}: 750Ω
- R_D: 750Ω
- R_{DC1}: 100Ω
- R_{DC2}: 100Ω
- R₁: 5.6k
- R₂: 5k
- R₃: 220Ω
- R₄: 350Ω

ภาคผนวก ซ โปรแกรมภาษาซีสำหรับควบคุมการทำงานของ Image digitizer

```

#include <stdio.h>
#include <dos.h> /* Required for Delay Function */

#define PORTBASE 0x0378 /* for the PC LPT1 */
#define STATUS   PORTBASE+1
#define CONTROL  PORTBASE+2
#define EPPADR   PORTBASE+3
#define EPPDATA  PORTBASE+4

main(void)
{
    FILE *outfilep;
    clrscr();
    // Save PCData into data.bin //
    outfilep=fopen("c:\\bin2bmp\\data.bin","w");
    if (outfilep==NULL) printf("Cannot open file\n");

    /* Write Process */
    outportb(CONTROL,0xC4); /* RW=1, Start to Write Process */
    inportb(EPPADR); /* Reset Active */
    printf("\nStart to Grab Image");
    while (((inportb(STATUS))&0x10)==0x00); // Wait Until TC=1 //
    inportb(EPPADR); /* Reset Active */

    /* Read Process and Saving Data */
    outportb(CONTROL,0xC0); /* RW=0, Start to Read Process */
    inportb(EPPADR); /* Reset Active */
    while (((inportb(STATUS))&0x10)==0x00) // Wait Until TC=1 //
    {
        putc((char)inportb(EPPDATA), outfilep);
    };
    inportb(EPPADR); /* Reset Active */
    fclose(outfilep);
    printf("\nThe End");
    return(0);
}

```

ประวัติผู้เขียน

ชื่อ นางสาวอรรรณ รุจิราลัย

วัน เดือน ปี เกิด 15 มีนาคม 2519

วุฒิการศึกษา

วุฒิ ชื่อสถาบัน ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้าสื่อสาร) มหาวิทยาลัยสงขลานครินทร์ 2541

ทุนการศึกษาที่ได้รับระหว่างศึกษา

โครงการทุนบัณฑิตศึกษาภายในประเทศ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.)