

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าวโดยการประมวลผลภาพ

A MACHINE FOR RICE CONTAMINANT DETECTION  
USING IMAGE PROCESSING



โดย

นายจตุรนต์ อยู่สำราญ รหัส 49015185  
นายฉัตรชัย รอดบุญ รหัส 49015187  
นายอัศวิน ปัสโก รหัส 49015221

อาจารย์ที่ปรึกษา

อ.ชินภัทร นันทจิวงกรชัย

เลขหมู่.....  
เลขทะเบียน 103101  
วัน,เดือน,ปี 27 ส.ค. 2552

b. 49015185
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2551

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าวสารโดยการประมวลผลภาพ

ผู้จัดทำ

- |               |           |      |          |
|---------------|-----------|------|----------|
| 1. นายจตุรนต์ | อยู่สำราญ | รหัส | 49015185 |
| 2. นายฉัตรชัย | รอดบุญ    | รหัส | 49015187 |
| 3. นายอัครวิน | ปัสโก     | รหัส | 49015221 |



.....อาจารย์ที่ปรึกษา  
(อาจารย์ชินภัทร นันทจิวงกรชัย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าว

## โดยการประมวลผลภาพ

นายจตุรนต์	อยู่สำราญ	รหัส 49015185
นายฉัตรชัย	รอดบุญ	รหัส 49015187
นายอัศวิน	ปัสโก	รหัส 49015221
อ.ชินภัทร	นันทจิวารชัย	อาจารย์ที่ปรึกษา
ปีการศึกษา 2551		

### บทคัดย่อ

แนวคิดในการทำโครงการ เครื่องคัดแยกสิ่งแปลกปลอมจากเมล็ดข้าวนั้น เกิดจากการที่ทราบดีว่าข้าวเป็นสินค้าที่สำคัญสำหรับประเทศไทย และยังสามารถส่งออกนารายได้เข้ามายังประเทศเป็นจำนวนมากในแต่ละปี แต่มีสิ่งหนึ่งที่จะทำให้ข้าวมีมาตรฐานลดลงจากสิ่งที่ควรจะเป็นก็คือสิ่งแปลกปลอมจะปะปนมากับเมล็ดข้าว ดังนั้นเราจึงต้องทำการคัดแยกสิ่งเหล่านี้ออกจากเมล็ดข้าวให้ได้โดยทำให้เสียเวลาในการคัดแยกสิ่งแปลกปลอมนี้ให้น้อยที่สุด และสาเหตุนี้จึงเป็นเหตุผลที่ทำให้คิดเครื่องคัดแยกสิ่งแปลกปลอมจากเมล็ดข้าวขึ้นมา

โดยหลักการทำงานของเครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าว คือเครื่องจะตรวจจับภาพของเมล็ดข้าวและสิ่งแปลกปลอมที่ถูกส่งผ่านตัวตรวจจับแบบ CCD และนำสัญญาณจาก CCD ไปประมวลผลสัญญาณโดยใช้ IC Signal Processor เบอร์ AD 9822 และนำสัญญาณไปเข้า FPGA ซึ่งภายในได้เขียนโปรแกรมแยกแยะและโปรแกรมนับเพื่อส่งสัญญาณ ไปให้วงจรควบคุมหัวเป่าเพื่อไปควบคุมวาล์วเป่าลมเพื่อเป่าสิ่งแปลกปลอมที่มากับเมล็ดข้าวได้

# A MACHINE FOR RICE CONTAMINATE DETECTION USING IMAGE PROCESSING

Mr. Jaturon Yoosamran ID. 49015185

Mr.Chatchai Rodboon ID. 49015187

Mr.Ausawin Pasko ID. 49015221

Mr.Chinnapat Nantajiwakornchai (Advisor)

Education Year 2008

## Abstract

The idea of inventing rice a machine for rice contaminate detection using image processing originates from the fact that rice is one important product of Thailand. It can be exported, making a lot of money to the country each year. However, one thing that devalues the standard of rice products is the adulteration in rice. It is, thus, needed to sort out such adulteration using the least time in the sorting-out process. With this reason, this detector is devised.

A machine for rice contaminate detection using image processing detects the image of rice kernel and adulteration that are passed through CCD detector. The signal from CCD is then processed by AD 9822 IC Signal Processor. After that, the signal is sent to FPGA, which is programmed to divide and count in order to send the signal to the blower-control circuit so that the blower valve can blow the adulteration that are in the rice kernel.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาวิชาโครงการพิเศษ ซึ่งโครงการนี้ไม่สามารถลุล่วงไปได้ด้วยดีหากไม่ได้รับการช่วยเหลือจากบุคคลต่อไปนี้

ทางผู้จัดทำขอขอบพระคุณอาจารย์ชินภัทร นันทจิวงกรชัย ที่ให้ความรู้ คำแนะนำต่างๆ ที่เกี่ยวกับการทดลอง ข้อเสนอแนะ การตอบข้อซักถามของผู้จัดทำ และช่วยปรับปรุงข้อบกพร่องที่เกิดจากการทดลอง จนกระทั่งโครงการนี้สำเร็จไปได้ด้วยดี

ขอขอบพระคุณเพื่อนที่คอยช่วยเหลือ ให้กำลังใจ และช่วยกันคิดแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการทดลอง ทำให้การทำโครงการนี้เป็นไปได้ด้วยดี

ขอกราบขอบพระคุณบิดา มารดาของผู้จัดทำที่เป็นกำลังใจ และสนับสนุนกำลังใจในการทำโครงการครั้งนี้

ผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงการนี้คงเป็นประโยชน์สำหรับผู้ที่เกี่ยวข้องทางด้านนี้ หากมีข้อผิดพลาดประการใด ผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ด้วย

นายจตุรนต์ อยู่สำราญ  
นายฉัตรชัย รอดบุญ  
นายอัศวิน ปัสโก

## สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
<b>บทที่ 1 บทนำ</b>	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ส่วนประกอบของโครงการ	2
<b>บทที่ 2 หลักการตรวจจับภาพของชุดหัวสแกนเนอร์</b>	3
2.1 ส่วนประกอบของชุดหัวสแกนเนอร์	3
2.2 คุณสมบัติเฉพาะของชุดหัวสแกนเนอร์	5
2.2.1 ความละเอียดของภาพและการแต่งเติม	5
2.2.2 สีและโทนสี	6
2.3 CCD (Charge Coupled Device)	6
2.3.1 การแนะนำ CCD	6
2.3.2 รูปแบบและขนาดของ CCD	6
2.3.3 สถาปัตยกรรมของ CCD (CCD Architectures)	9
2.3.4 CCD พื้นฐาน (CCD Basic)	11
2.3.5 เทคโนโลยีของ CCD	13
<b>บทที่ 3 โครงสร้างและหลักการทำงานของ FPGA</b>	20
3.1 ความเป็นมาของเอซิก (ASIC)	20
3.2 ประเภทของ ASIC	20
3.2.1 Full – custom	21
3.2.2 Semi – custom	21
3.2.3 Programmable	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

เรื่อง	หน้า
3.3 หลักการของ FPGA	24
3.3.1 การเขียนโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ	24
3.3.2 การโปรแกรมโดยใช้หน่วยความจำ	24
3.3.3 โครงสร้างภายในของ FPGA	25
3.3.4 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว	26
3.3.5 การออกแบบวงจร	27
3.3.6 เครื่องมือสำหรับการออกแบบ FPGA	30
3.4 ภาษา VHDL	30
3.4.1 ประวัติความเป็นมาของภาษา VHDL	31
3.4.2 ข้อกำหนด	32
3.4.3 องค์ประกอบพื้นฐานของ VHDL	34
บทที่ 4 การใช้โปรแกรม Xilinx- ISE เบื้องต้น	41
4.1 ขั้นตอนการออกแบบวงจรนับ 4 แบบเชิงโครนัสด้วยวิธีวาดผังวงจร	41
บทที่ 5 การทดลองและออกแบบ	61
5.1 หลักในการออกแบบ	61
5.2 การออกแบบวงจร Drive CCD	62
5.2.1 การออกแบบสัญญาณ $\phi$ ROG	62
5.2.2 การออกแบบสัญญาณ $\phi$ 1 และ $\phi$ 2	65
5.2.3 การออกแบบสัญญาณ $\phi$ RS และ $\phi$ CLP	69
5.3 รูปสัญญาณ Output จาก CCD	77
5.3.1 รูปสัญญาณที่มาจาก CCD โดยตรง	77
5.3.2 รูปสัญญาณ Output R ของ CCD ที่มีการตรวจจับภาพ	78
5.3.3 รูปสัญญาณ Output G ของ CCD ที่มีการตรวจจับภาพ	79
5.3.4 รูปสัญญาณ Output B ของ CCD ที่มีการตรวจจับภาพ	80
5.4 การสร้างสัญญาณ CDSCLK 1 และ CDSCLK 2	84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

เรื่อง	หน้า
5.5 การทำงานของวงจรแยกราจ	83
5.5.1 การทำงานของ Shift Register	84
5.5.2 การทำงานร่วมกันของวงจรแยกราจกับวงจร Shift Register	85
5.5.3 การทำงานของวงจรมับ (COUNTER)	86
5.5.4 วงจรชุดควบคุมหัวเป่า	88
5.6 รูปสัญญาณเมื่อเข้าวงจรแยกราจ	89
5.7 ผลการทดลองถึงความลาดชันของรางที่องศาต่างๆ	95
5.8 ผลการทดลองการคัดแยกเมล็ดข้าวที่ไม่ต้องการโดยใช้ชุดหัวเป่า	97
<b>บทที่ 6 สรุปและวิจารณ์ผลการทดลอง</b>	99
<b>บรรณานุกรม</b>	100
<b>ภาคผนวก</b>	101

## สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวอย่างกระจก 3 ชั้น พร้อมกับเลนส์ที่ประกอบเป็นหัวสแกนเนอร์	4
รูปที่ 2.2 ตัวอย่าง CCD	5
รูปที่ 2.3 Point Scanning	7
รูปที่ 2.4 Line Scanning	8
รูปที่ 2.5 Area Scanning	9
รูปที่ 2.6 Full – Frame Architecture	10
รูปที่ 2.7 Interline Architecture	11
รูปที่ 2.8 Photon Interaction with Silicon	12
รูปที่ 2.9 Potential Well and Barriers	12
รูปที่ 2.10 Floating Diffusion Readout Structure	13
รูปที่ 2.11 Color Sequential capture	14
รูปที่ 2.12 Three – chip Color Capture	15
รูปที่ 2.13 Integral Color Filter Array Patterns	16
รูปที่ 2.14 Normal and Thinned CCD	17
รูปที่ 2.15 Interline CCD Showing Photodiode and Non Sensitive CCD covered by Light Shield	18
รูปที่ 2.16 Interline CCD With Microlenticular Array	18
รูปที่ 3.1 ประเภทของ ASIC	21
รูปที่ 3.2 วงจรพื้นฐานภายในของ PLA	22
รูปที่ 3.3 วงจรพื้นฐานภายในของ PLA	23
รูปที่ 3.4 วงจรพื้นฐานภายในของ PAL	23
รูปที่ 3.5 โครงสร้างภายในของ FPGA ตระกูล MAX7000S	25
รูปที่ 3.6 โครงสร้างภายในของ FPGA ตระกูล FLEX10K	26
รูปที่ 3.7 การโปรแกรมลงชิพ	26
รูปที่ 3.8 การกำหนดการเชื่อมต่อและสถาปัตยกรรม	35
รูปที่ 3.9 บล็อกไดอะแกรมและการบรรยายการเชื่อมต่อของ clock_compoment	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.10 การบรรยายเชิงพฤติกรรมของ clock_component	36
รูปที่ 3.11 โครงสร้างทั่วไปของส่วนการประกาศแพ็คเกจ	37
รูปที่ 3.12 โครงสร้างของบอดีแพ็คเกจ	37
รูปที่ 3.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	38
รูปที่ 3.14 การใช้โพธิ์เจอร์	38
รูปที่ 3.15 การใช้ฟังก์ชัน	39
รูปที่ 3.16 ตัวดำเนินการใน VHDL	40
รูปที่ 4.1 วงจรนับ 4 แบบซิงโครนัส	41
รูปที่ 4.2 หน้าต่าง Xilinx-ISE	42
รูปที่ 4.3 หน้าต่าง New Project Wizard-Create New Project	43
รูปที่ 4.4 หน้าต่าง New Project Wizard-Device Properties	44
รูปที่ 4.5 หน้าต่าง New Project Wizard-Create New Source	44
รูปที่ 4.6 หน้าต่าง New Project Wizard-Select Source Type	45
รูปที่ 4.7 หน้าต่าง New Project Wizard-Add Existing Source	45
รูปที่ 4.8 หน้าต่าง Xilinx-ISE	46
รูปที่ 4.9 รายการอุปกรณ์	47
รูปที่ 4.10 รายการอุปกรณ์ที่เห็นรายละเอียด	47
รูปที่ 4.11 การคลิกเลือกหมวดและชนิดอุปกรณ์ที่อยู่ใน Categories และ Symbols ตามลำดับ	48
รูปที่ 4.12 แสดงขั้นตอนการ Zoom ขยาย	49
รูปที่ 4.13 รูปที่ Zoom ขยายแล้ว	49
รูปที่ 4.14 วางอุปกรณ์เกตแบบอินเวอร์เตอร์และ Vcc	50
รูปที่ 4.15 แสดงการลากเส้นระหว่างจุด 2 จุด	51
รูปที่ 4.16 สายสัญญาณที่ถูกวาดแนวเส้นอย่างอัตโนมัติ	51
รูปที่ 4.17 แสดงต่อสายสัญญาณต่างๆจนครบตามที่ออกแบบไว้	52
รูปที่ 4.18 การใส่ I/O Marker เพื่อกำหนดจุดปลายของสัญญาณ	53
รูปที่ 4.19 ขั้นตอนการแก้ไขชื่อของ Wire หรือ Net ที่ I/O marker	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

	หน้า
รูปที่ 4.20 ขั้นตอนการแก้ไขชื่อของ Wire หรือ Net ที่ I/O marker	54
รูปที่ 4.21 การแก้ไขชื่อของ Wire หรือ Net ที่ I/O Marker จนได้วงจรที่แล้วเสร็จสมบูรณ์	55
รูปที่ 4.22 การตรวจสอบความถูกต้อง	56
รูปที่ 4.23 การสร้างไฟล์ Test Bench Waveform	57
รูปที่ 4.24 หน้าต่าง New Source Wizard-Select Source Type	57
รูปที่ 4.25 หน้าต่าง New Source Wizard-Associate Source	58
รูปที่ 4.26 New Source Wizard-Summary	58
รูปที่ 4.27 หน้าต่าง Intial Timing and Clock Wizard	59
รูปที่ 4.28 สัญญาณที่สร้างขึ้นมา	59
รูปที่ 4.29 หน้าต่าง Sources	60
รูปที่ 4.30 รูปสัญญาณที่ทำการจำลองจากวงจรที่ออกแบบ	60
รูปที่ 5.1 ข้อมูล Datasheet ILX548K	62
รูปที่ 5.2 คำสั่งในการกำหนดขาขึ้นของสัญญาณ $\phi$ ROG	63
รูปที่ 5.3 คำสั่งในการกำหนดขาลงของสัญญาณ $\phi$ ROG	64
รูปที่ 5.4 Block Diagram สัญญาณ $\phi$ ROG	64
รูปที่ 5.5 ผลการ SIMULATOR ของสัญญาณ $\phi$ ROG	65
รูปที่ 5.6 คำสั่งในการกำหนดขาลงของสัญญาณ $\phi$ 1	66
รูปที่ 5.7 คำสั่งในการกำหนดขาลงสัญญาณ $\phi$ 1	67
รูปที่ 5.8 Block Diagram ของสัญญาณ $\phi$ 1 และ $\phi$ 2	68
รูปที่ 5.9 ขอบขาขึ้นของสัญญาณ $\phi$ ROG เทียบกับสัญญาณ $\phi$ 1 และ $\phi$ 2	68
รูปที่ 5.10 ขอบขาลงของสัญญาณ $\phi$ ROG เทียบกับสัญญาณ $\phi$ 1 และ $\phi$ 2	68
รูปที่ 5.11 คำสั่งของวงจร Decoded 3 to 8	69
รูปที่ 5.12 ค่า Din ที่กำหนดในการ SIMULATOR	70
รูปที่ 5.13 ผลการSIMULATOR ของโปรแกรม Decodeder 3 to8	70
รูปที่ 5.14 คำสั่งในการกำหนดขาขึ้นของสัญญาณ RS	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

	หน้า
รูปที่ 5.15 คำสั่งในการกำหนดขาลงของสัญญาณ RS	72
รูปที่ 5.16 Block Diagram $\phi$ RS และ $\phi$ CLP	73
รูปที่ 5.17 ขอบขาขึ้นของสัญญาณ $\phi$ ROG เทียบกับ สัญญาณ $\phi$ RS	73
รูปที่ 5.18 ขอบขาลงของสัญญาณ $\phi$ ROG เทียบกับ สัญญาณ $\phi$ RS	73
รูปที่ 5.19 Counter 4 Bit	74
รูปที่ 5.20 โปรแกรม Divider	75
รูปที่ 5.21 วงจร Drive CCD	76
รูปที่ 5.22 รูปสัญญาณที่ได้มาจากบอร์ด FPGA	76
รูปที่ 5.23 รูปสัญญาณ OUTPUT จาก CCD ขณะที่ยังไม่ได้ทำการตรวจจับวัตถุ	77
รูปที่ 5.24 แถบกระดาษที่มีสีขาว-ดำที่ใช้ในการทดสอบ	77
รูปที่ 5.25 รูปสัญญาณ OUTPUT R ของ CCD ที่มีกรตรวจจับภาพสีขาว	78
รูปที่ 5.26 Block Diagram ภายใน IC Signal Processor เบอร์ AD 9822	78
รูปที่ 5.27 สัญญาณของ 3 ch CDS Mode	79
รูปที่ 5.28 สัญญาณของ 1 ch CDS Mode	79
รูปที่ 5.29 สัญญาณของ 3 Ch SHA Mode	80
รูปที่ 5.30 สัญญาณของ 1 Ch CDS Mode	80
รูปที่ 5.31 รูปแบบของสัญญาณ Register	80
รูปที่ 5.32 ผลของการ Simulation ของสัญญาณ Register ที่ออกแบบไว้	81
รูปที่ 5.33 รูปแบบของสัญญาณ CDSCLK 1 และ CDSCLK 2	81
รูปที่ 5.34 วงจรสร้างสัญญาณ CDSCLK 1 และ CDSCLK 2	82
รูปที่ 5.35 รูปสัญญาณ CDSCLK 1 และ CDSCLK 2	82
รูปที่ 5.36 วงจรแยกรางที่ได้ออกแบบไว้ใน FPGA	83
รูปที่ 5.37 การนำ D FILP FLOP มาจัดเป็นวงจรรับเข้าแบบอนุกรมและส่งออกแบบขนาน	84
รูปที่ 5.38 หลังจากมีสัญญาณ CLOCK เข้ามาครบ 8 ลูก Data ที่ส่งเข้ามาในจังหวะที่สัมพันธ์กับ สัญญาณ CLOCK จะปรากฏออกมาที่ขา QA, QB, QC, QD, QE, QF, QG, และ QH	85
รูปที่ 5.39 การต่อใช้งานของวงจรแยกรางและวงจร Shift Register	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าในรูปแบบใด ๆ ทั้งสิ้น หากมีข้อผิดพลาดประการใด และต้องขออภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

	หน้า
รูปที่ 5.40 วงจรนับ 4 หลัก	87
รูปที่ 5.41 สัญลักษณ์ของวงจรถับ 4 หลัก	87
รูปที่ 5.42 วงจรควบคุมหัวเป่า	88
รูปที่ 5.43 วงจรแยกราง	89
รูปที่ 5.44 สัญลักษณ์ของวงจรถับแยกรางที่ได้จากการ Simulate ใน FPGA	89
รูปที่ 5.45 รูปสัญลักษณ์ที่ผ่านวงจรถับแยกราง	90
รูปที่ 5.46 รูปสัญลักษณ์ขณะที่ยังไม่มีเมล็ดข้าวสีดำผ่าน	91
รูปที่ 5.47 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 1	91
รูปที่ 5.48 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 2	91
รูปที่ 5.49 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 3	92
รูปที่ 5.50 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 4	92
รูปที่ 5.51 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 5	93
รูปที่ 5.52 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 6	93
รูปที่ 5.53 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 7	94
รูปที่ 5.54 รูปสัญลักษณ์ที่มีเมล็ดข้าวสีดำผ่านรางที่ 8	94

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

ผลผลิตทางการเกษตรนั้น จัดได้ว่าเป็นสิ่งที่มีความสำคัญต่อการดำรงชีวิตของคนไทยมาเป็นระยะเวลาอันยาวนานแล้ว และปัจจุบันก็เป็นผลผลิตที่สำคัญต่อเศรษฐกิจของประเทศไทย ซึ่งเป็นประเทศเกษตรกรรมที่มีความสำคัญต่อประเทศไทยมาก โดยที่ผลผลิตแต่ละปีนั้นมียอดออกมาเป็นจำนวนมาก และผลผลิตที่ออกมาก็จำเป็นต้องนำมาทำการคัดแยก ซึ่งขั้นตอนการคัดแยกนี้จะมี ความสำคัญเป็นอย่างมาก เพราะจะทำให้ผลผลิตที่ออกมามีคุณภาพและเป็นการเพิ่มมูลค่าให้แก่ผลผลิต

จากแนวคิดข้างต้นจึงทำให้มีการนำเอาความรู้ด้านดิจิทัลทางอิเล็กทรอนิกส์มาประยุกต์ใช้งานในการทำโครงการพิเศษขึ้นนี้ คือ เครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าว โดยการประมวลผลภาพ ซึ่งลักษณะการตรวจจับนั้นจะทำการนำเอาสีของเมล็ดข้าวมาทำการรับสัญญาณภาพจากตัวซีซีดี (Charge Coupled Device) และตรวจจับภาพแบบซีพ CCD และนำสัญญาณภาพที่ได้จาก CCD มาขยายช่วงสัญญาณให้ได้ขนาดแรงดันตามที่ต้องการ เมื่อได้ระดับแรงดันที่ต้องการแล้ว ก็จะได้เมล็ดข้าวออกเป็น 2 จำพวก คือ เมล็ดข้าวที่เป็นสีอ่อน ซึ่งเป็นเมล็ดข้าวที่ต้องการ โดยที่ไม่แสดงผลออกทาง LED เมื่อมีการตรวจจับ และเมล็ดข้าวที่เป็นสีเข้ม ซึ่งเป็นเมล็ดข้าวที่ต้องการคัดแยกออกไป จะมีการแสดงผลออกทาง LED เมื่อมีการตรวจจับ โดยที่จะมีการนับค่าของเมล็ดข้าวที่ไม่ต้องการออกทาง 7-Segment ทำให้ทราบค่าของสิ่งแปลกปลอมที่ปะปนมากับเมล็ดข้าวเพื่อที่จะได้เข้าสู่ขั้นตอนการแปรรูปเมล็ดข้าวที่ไม่ต้องการต่อไป

### 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาและประยุกต์การใช้งานของสัญญาณที่มาจาก CCD
- 1.2.2 เพื่อศึกษาและออกแบบการแปลงสัญญาณภาพเป็นสัญญาณดิจิทัล
- 1.2.3 เพื่อศึกษาและประยุกต์การใช้งาน FPGA
- 1.2.4 เพื่อศึกษาและออกแบบกลไกการคัดแยกผลผลิตทางการเกษตร

### 1.3 ขอบเขตของโครงการงาน

- 1.3.1 เขียน โปรแกรม FPGA เพื่อสร้างสัญญาณ Drive CCD
- 1.3.2 เขียน โปรแกรม FPGA เพื่อสร้างวงจรนับ
- 1.3.3 วัดสัญญาณที่ได้มาจากการเขียนโปรแกรม FPGA โดยใช้ Logic Analyzer
- 1.3.4 วัดความแตกต่างของสัญญาณ CCD ที่มีสีของภาพที่แตกต่างกันนั้นผ่านตัว CCD
- 1.3.5 เขียน โปรแกรม FPGA เพื่อสร้างสัญญาณให้ IC Signal Processor เบอร์ AD9822
- 1.3.6 วัดสัญญาณที่ได้มาจากการสร้างโปรแกรม FPGA ให้ IC Signal Processor เบอร์ AD9822
- 1.3.7 สร้างวงจรแปลงสัญญาณ A/D ให้ IC Signal Processor เบอร์ AD9822
- 1.3.8 เขียน โปรแกรม FPGA เพื่อทำการแยกสีขาว-ดำจาก CCD
- 1.3.9 สร้างวงจรควบคุมชุดหัวเป่า

### 1.4 ส่วนประกอบของโครงการงาน

- 1.4.1 ชุดหัวสแกนเนอร์
- 1.4.2 บอร์ดทดลอง FPGA
- 1.4.3 Logic Analyzer
- 1.4.4 ชุดหัวเป่าลม

## บทที่ 2

### หลักการตรวจจับภาพของชุดหัวสแกนเนอร์

การตรวจจับภาพเป็นการแปลงภาพ วัตถุ หรือเอกสารให้อยู่ในรูปแบบข้อมูลดิจิทัล โดยที่ชุดหัวสแกนเนอร์จะอ่านภาพหรืออิมเมจที่ประกอบด้วยจุด (หรือพิกเซล) จำนวนมาก จากนั้นซอฟต์แวร์ของการตรวจจับจะแปลงภาพให้เป็นข้อมูลดิจิทัลแล้วนำออกไปใช้งานตามที่ต้องการ โดยไฟล์ภาพที่ถูกตรวจจับขึ้นมาจะถูกเก็บไว้ในฐานข้อมูล

#### 2.1 ส่วนประกอบของชุดหัวสแกนเนอร์

ส่วนประกอบของชุดหัวสแกนเนอร์ตามมาตรฐานก็จะต้องมีอุปกรณ์ต่างๆรวมเข้าไว้ด้วยกัน เช่น หลอดไฟ (Lamp) , เลนส์ (Lens) , กระจกเงา (Mirrors) , แผ่นกระจกใส (Glass Plate) , หัวตรวจจับภาพ (Scan Head) , อุปกรณ์จับประจุไฟฟ้า (CCD) , แผ่นกรองแสง (Filters) , แหล่งจ่ายไฟ (Power Supply) , แผงวงจรรวม (Control Circuitry) , หัวต่อเชื่อม (Interface Port) , และอุปกรณ์อื่นๆ โดยจะอธิบายถึงความสำคัญของอุปกรณ์ดังนี้

##### 2.1.1 หลอดไฟ (Lamp)

หลอดไฟมีจำนวนหนึ่งหลอด จะทำหน้าที่ส่องแสงไปยังเอกสารหรือรูปภาพเพื่อให้เกิดแสงสะท้อนไปยังกระจก หลอดไฟในเครื่องตรวจจับส่วนใหญ่เป็นหลอดฟลูออเรสเซนต์หรือหลอดนีออน

##### 2.1.2 เลนส์ (Lens)

เลนส์จะทำหน้าที่รับแสงสะท้อนจากกระจก เพื่อเพิ่มความคมชัดให้กับข้อมูลจากนั้นแสดงข้อมูลที่ไปยัง CCD

##### 2.1.3 กระจกเงา (Mirrors)

กระจกเงาโดยทั่วไปแล้วจะมีแผ่นในตัวเครื่องจะทำหน้าที่สะท้อนแสงที่หักเหมาจากหลอดไฟ เพื่อให้แสงนั้นผ่านเลนส์แล้วส่งต่อไปยัง CCD

##### 2.1.4 หัวตรวจจับภาพ (Scan Head)

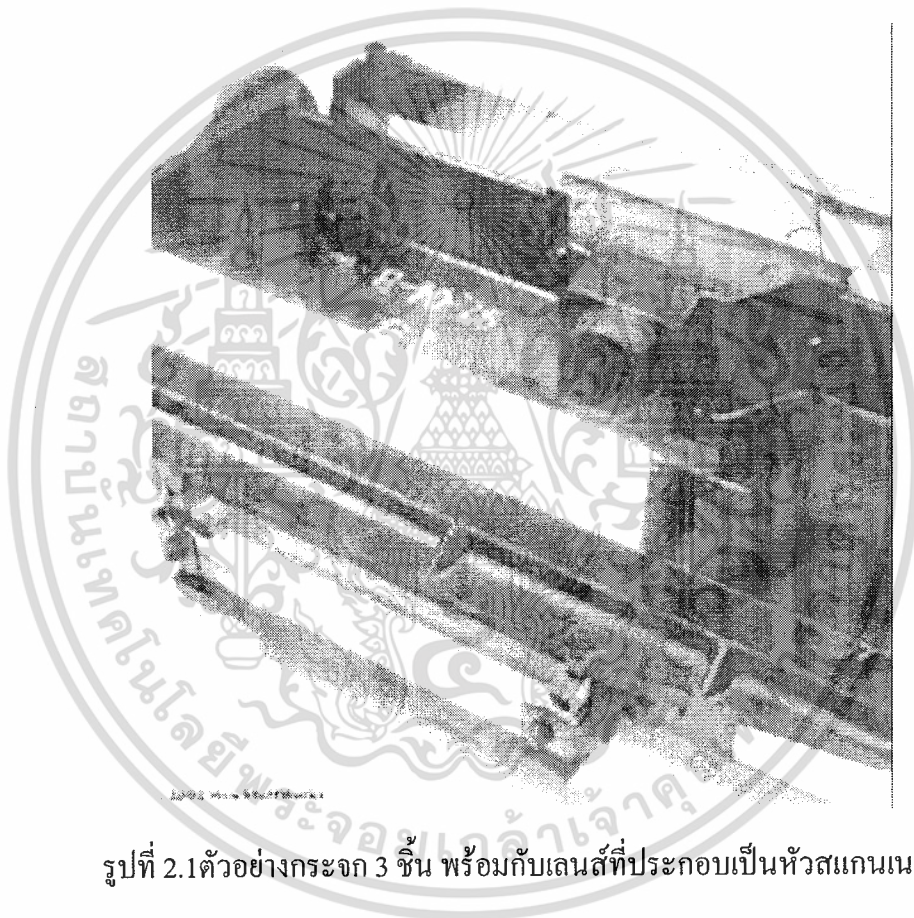
หัวตรวจจับภาพ ประกอบด้วยชุดของกระจก เลนส์ ฟิลเตอร์และ CCD จะทำหน้าที่เคลื่อนผ่านเอกสารหรือรูปภาพอย่างช้าๆ เพื่ออ่านข้อมูลไปประมวลผล

### 2.1.5 แผ่นกรองแสง (Filters)

มีลักษณะเป็นแผ่นแก้วสีขาวโปร่งใสหรือสีอื่นๆ ใช้สวมไว้ต่อจากเลนส์ ทำหน้าที่เป็นฉากกั้นแสงสีบางสี และเพิ่มแสงสีบางสีไปตกกระทบบน CCD นอกจากนี้แผ่นกรองแสงนี้ยังช่วยเปลี่ยนความเข้มของสีวัตถุและสามารถช่วยสร้างภาพให้มีลักษณะพิเศษตามที่ต้องการ

### 2.1.6 แผ่นกระจกใส (Glass Plate)

เป็นแผ่นแก้วหน้าโปร่งใสเป็นส่วนวางเอกสารหรือรูปภาพก่อนเริ่มการตรวจจับภาพ

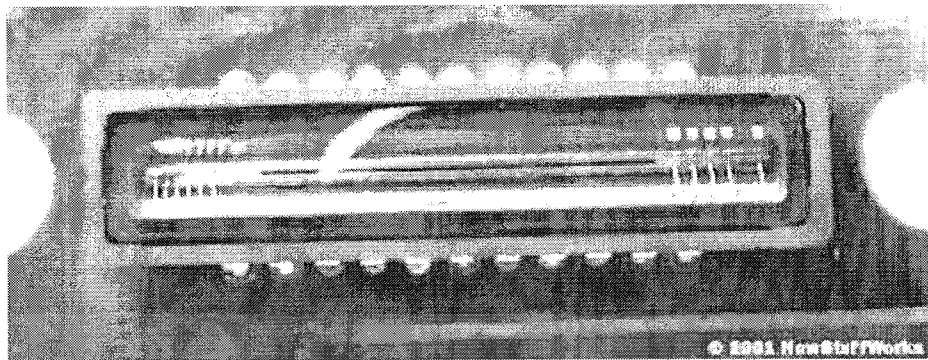


รูปที่ 2.1 ตัวอย่างกระจก 3 ชั้น พร้อมกับเลนส์ที่ประกอบเป็นหัวสแกนเนอร์

### 2.1.7 อุปกรณ์จับประจุไฟฟ้า (CCD)

เป็นอุปกรณ์หัวใจของการตรวจจับภาพ CCD เป็นเทคโนโลยีที่ตรวจจับแสง (Optical sensor) CCD คือ ชุดของไดโอดขนาดเล็กที่มีความไวแสง โดยจะเปลี่ยนโฟตอน (ซึ่งก็คือแสง) ให้เป็นอิเล็กตรอน (ซึ่งคือสัญญาณไฟฟ้า) ไดโอดชนิดนี้จะเรียกว่า Photosites ซึ่งเอกสารที่จะต้องการตรวจจับภาพจะถูกวางครอบบน CCD ซึ่งเทคโนโลยีที่เครื่องตรวจจับภาพที่แปลงข้อมูลอนาล็อกเป็นข้อมูลในรูปแบบดิจิทัล จากนั้นข้อมูลจะถูกส่งไปชีพผ่านระบบออฟดิคัล ซึ่งเป็นเลนส์และระบบกระจก แล้วประมวลผลไปใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ตัวอย่าง CCD

## 2.2 คุณสมบัติเฉพาะของชุดหัวสแกนเนอร์

### 2.2.1 ความละเอียดของภาพและการแต่งเติม

ความละเอียดของภาพเป็นคุณสมบัติแรกที่ผู้ใช้งานสแกนเนอร์ควรทราบ โดยปกติแล้วชุดหัวสแกนเนอร์จะเก็บรายละเอียดของสิ่งที่อ่านและแบ่งออกเป็นจุดเล็กๆหรือที่เรียกว่า พิกเซล ดังนั้นความละเอียดจึงเป็นการวัดจำนวนพิกเซลที่ชุดหัวสแกนเนอร์สามารถเก็บรายละเอียดของเอกสารหรือภาพโดยวัดเป็นจุดภาพต่อตารางนิ้ว หรือ DPI (Dot per Inch) ส่วนใหญ่ชุดหัวของสแกนเนอร์มีความละเอียดเริ่มต้นที่ 300 dpi หรือ  $300 \times 300$  จุดต่อนิ้ว หรือเรียกอีกอย่างหนึ่งว่าชุดหัวสแกนเนอร์สามารถอ่านภาพที่สแกนได้ 90,000 พิกเซล ต่อพื้นที่ 1 ตารางนิ้วในการอ่าน เครื่องตรวจจับภาพจะอ่านภาพ 2 ครั้ง อ่านไปและอ่านย้อนกลับอีกทียวหนึ่ง

บางครั้งความละเอียดของการตรวจจับภาพอาจมีความแตกต่างกันเช่น  $300 \times 600$  จุดต่อนิ้ว หมายความว่า ชุดสแกนเนอร์หยุดเมื่ออ่านค่าในแนวตั้งบ่อยครั้งกว่าอ่านในแนวนอน ซึ่งดูเหมือนว่าเอาท์พุทที่ได้จะมีคุณภาพดีกว่า แต่การพิจารณาควรพิจารณาจากตัวเลขที่น้อยกว่าแทนเพราะเป็นค่าที่บ่งบอกถึงคุณภาพของการตรวจจับที่น่าเชื่อถือ

ชุดหัวสแกนเนอร์รุ่นใหม่ๆจะมีความละเอียดตั้งแต่ 300 600 1,000 และ 1,200 dpi ชุดสแกนเนอร์บางรุ่นมีค่าความละเอียด 2 ประเภท คือ Optical และ Interpolate (หรือ Enhanced) ความละเอียด Optical จะวัดจากจำนวนพิกเซลจริงๆถ้าความละเอียดมีค่าสูง คุณภาพของงานที่ได้จะสูงตามไปด้วย ส่วนความละเอียด Interpolate เป็นค่าเฉลี่ยที่ค่าพิกเซล 2 พิกเซลติดกัน เพื่อสร้างพิกเซลที่สามสี่ของจุดที่สร้างขึ้นใหม่นี้ในระหว่างการตรวจจับภาพเลย สำหรับในการพิจารณาให้ใช้ค่าความละเอียด Optical ซึ่งเป็นค่าที่บอกถึงความละเอียดจริงๆของการตรวจจับภาพและมีความน่าเชื่อถือมากกว่า

## 2.2.2 สีและโทนสี

สีและโทนสีเป็นส่วนที่สำคัญเกี่ยวกับการตรวจจับภาพ ดังนั้นภาพจากชุดหัวสแกนเนอร์จะอยู่ในรูปแบบดิจิทัล ขนาดของไฟล์รูปภาพ จะประกอบด้วยจำนวนพิกเซลเป็นร้อยเป็นพันซึ่งจะถูกบันทึกค่าความเข้มสีของพิกเซลแต่ละพิกเซล ด้วยจำนวน 1 บิต หรือหลายๆบิต และรูปแบบการเก็บข้อมูล มีหลายระบบซึ่งคิดจากบิต เช่น 1 บิต 8 บิต 24 บิต โดยถ้าเป็นข้อมูลแบบ 1 บิต จะใช้สำหรับเก็บข้อมูลพิกเซล 2 สถานะคือ “0” และ “1” ซึ่งจะแสดงสีได้เฉพาะขาวกับดำ แต่ถ้าเป็น 8 บิต จะได้ความแตกต่างของสีถึง 256 ระดับ การรวมแม่สีมีเทคนิคที่เรียกว่า Dithering ซึ่งจะแสดงสีได้ไม่เหมือนกันกับความจริงที่เห็นได้ สำหรับ 24 บิต จะให้ภาพที่มีสีใกล้เคียงจริงมากที่สุด เรียกว่า Photo realistic โดยจะแบ่ง 24 บิต เป็น 3 ส่วนคือ แดง , เขียว , น้ำเงิน ส่วนละ 8 บิต เมื่อรวมทั้ง 3 เข้าด้วยกันจะสามารถแสดงสีได้ถึง 16.7 ล้านสี

## 2.3 CCD (Charge Coupled Device)

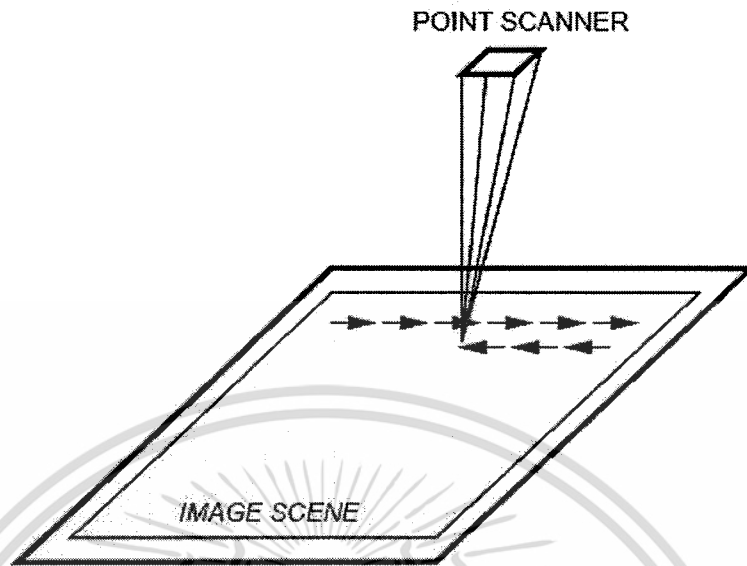
### 2.3.1. การแนะนำ CCD

CCD ถูกประดิษฐ์ครั้งแรกในช่วงปลายทศวรรษ 1960 โดย Bell Labs โดยเบื้องต้นมีจุดมุ่งหมายที่จะพัฒนา Computer memory circuit และในปี 1970 ได้ถูกนำมาออกมาสาคิดการใช้งานอย่างไรก็ตาม CCD ยังสามารถนำมาใช้ในด้านอื่นๆได้อีก รวมถึงการประมวลผลของสัญญาณ (Signal Processing) และการสร้างรูปภาพ (Imaging) ในที่สุด CCD ก็ได้รับการพัฒนาให้มีความสามารถตอบสนองกับแสงที่มีความยาวคลื่นน้อยกว่า 1.1 ไมครอนเมตร (ช่วง Spectrum ของแสงที่สามารถมองเห็นได้ของมนุษย์คือ 0.4 ไมครอนเมตร ถึง 0.7 ไมครอนเมตร) ในปัจจุบันการนำเอา CCD มาใช้ในการสร้างหน่วยความจำจึงลดลง แต่ได้นำมาประยุกต์ใช้ในการตรวจสอบและสร้างภาพ (Imaging Sensor)

### 2.3.2 รูปแบบและขนาดของ CCD

การสร้างรูปแบบ (Image Sensing) สามารถทำได้โดยการใช้เทคนิค 3 เทคนิคดังนี้

1) ตรวจจับภาพแบบจุด (Point Scanning) ดังรูปที่ 2.3 ซึ่งจะมีการอ่านค่าที่จุดเพียงจุดเดียวเรียกว่าพิกเซล (Pixel) โดยรูปจะได้รับการอ่านตามลำดับคู่ลำดับ X Y



รูปที่ 2.3 Point Scanning

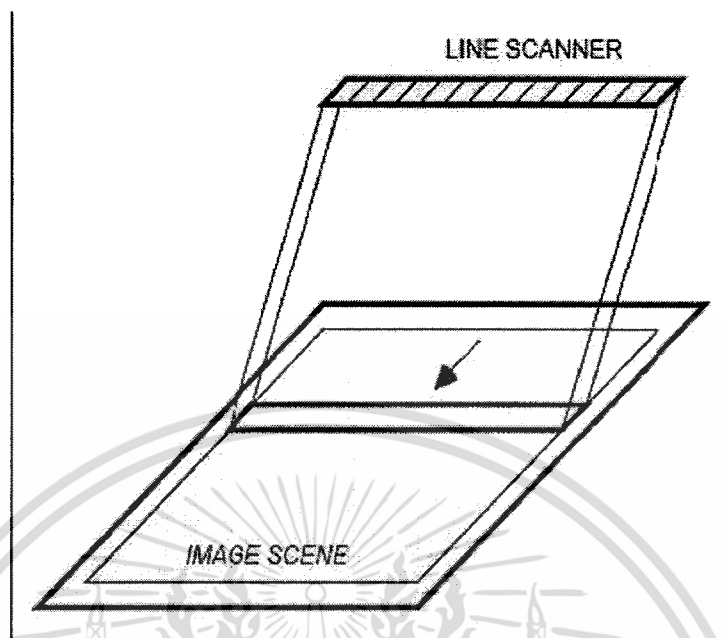
ข้อดี

- คุณภาพของภาพที่ได้สูง
- มีการอ่านค่าแบบ Uniform (ทุกจุดบนรูปแบบอ่านด้วย CCD เดียวกัน)

ข้อเสีย

- อาจมีข้อผิดพลาดจากการเลื่อนตำแหน่งในการอ่านข้อมูล
- ความเร็วต่ำ

2) ตรวจจับภาพแบบเส้น (Line Scanning) ดังรูปที่ 2.4 มีการนำเอา CCD แบบ Point Scanning หลายๆรูปแบบมาประกอบกันในแนวเส้นตรง ทำให้สามารถอ่านค่าได้เป็นแถวเป็นคราวเดียวกัน จากนั้นจึงเลื่อนไปในแถวอื่นๆ



รูปที่ 2.4 Line Scanning

ปัญหา

- จะทำอย่างไรให้ CCD อยู่ชิดกันมากๆเพื่อการประมวลผลของภาพดีที่สุด

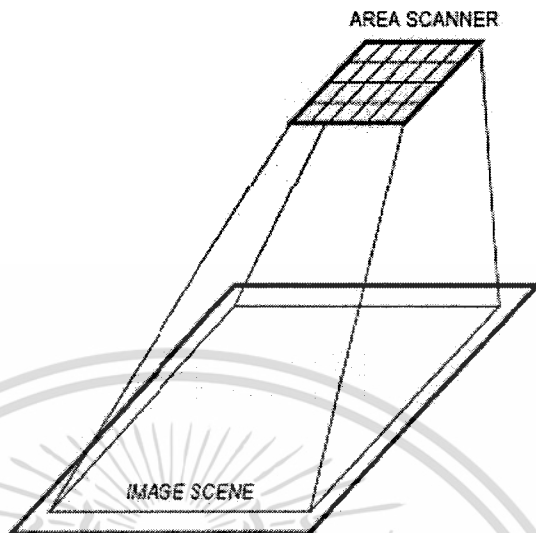
ข้อดี

- เร็วกว่าแบบ Point Scanning

ข้อเสีย

- การอ่านค่า CCD หลายๆตัวทำให้เกิด Error และอ่านค่าได้ไม่เท่ากันในแต่ละจุด
- ราคาสูง

3) ตรวจจับภาพทั่วพื้นที่ (Area Scanning) ดังรูปที่ 2.5 ใช้ Array 2 มิติของ CCD แบบ Point Scanning จะสามารถอ่านข้อมูลได้มากขึ้น ในการตรวจจับ 1 ครั้ง



รูปที่ 2.5 Area Scanning

ข้อดี

- อ่านข้อมูลได้เร็ว
- มีความซับซ้อนต่ำ

ข้อเสีย

- มีอัตราข้อมูลต่อสัญญาณรบกวน (Signal to Noise Ratio) ต่ำ
- ราคา เพราะมีจำนวนของ CCD ที่จะใส่ใน Wafer ต่ำลง

### 2.3.3 สถาปัตยกรรมของ CCD (CCD Architectures)

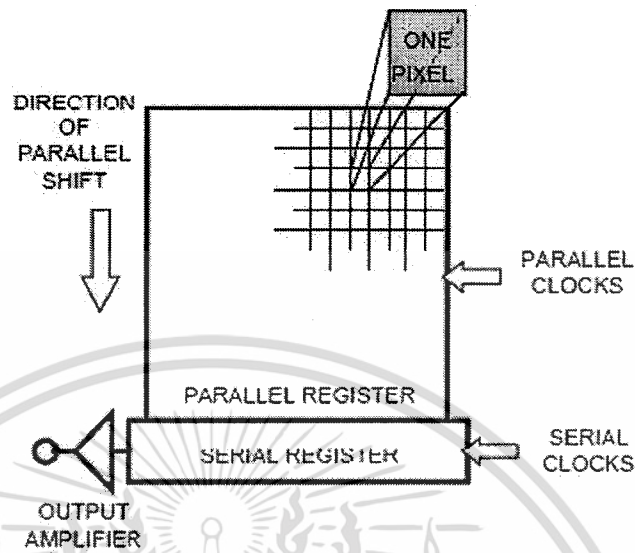
CCD มีหลายรูปแบบสถาปัตยกรรม ซึ่งรูปแบบหลักที่ใช้ในปัจจุบัน คือ Full Frame (FF), Interline (IL)

1) Full Frame (FF) มีรูปแบบการทำงานที่ง่ายที่สุดในการทำงาน วิธีนี้จะประกอบไปด้วย (ดังรูปที่ 2.6)

Parallel CCD Shift Register

Serial CCD Shift Register

Signal Sensing output amplifier

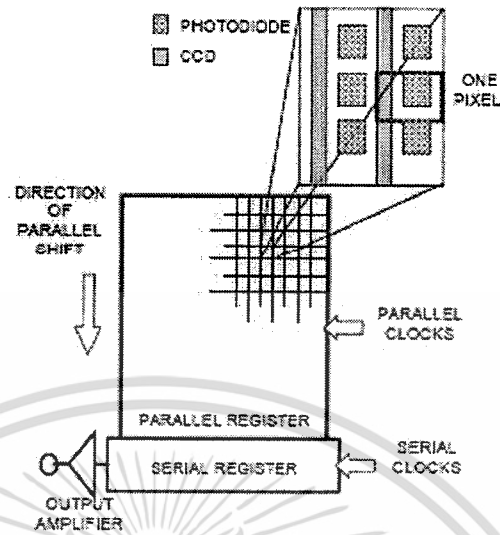


รูปที่ 2.6 Full - Frame Architecture

ข้อดี

- ง่ายในการออกแบบ
- Resolution สูง และ Density สูง

2) Interline (IL) ดังรูปที่ 2.8 มีการแยกหน่วยที่ทำหน้าที่ในการตรวจจับ ออกจากส่วนอ่านค่า โดยการเพิ่มส่วนที่ช่วยกันลำแสง (Light Shielded) เข้าไปในระหว่างแถวของอุปกรณ์ในการตรวจจับภาพซึ่งช่วยให้ทุกๆ พิกเซลที่เก็บข้อมูลได้ถูกส่งไปยังส่วนที่กันลำแสงทันทีและจะได้ผลลัพธ์เช่นเดียวกันแบบ Full Frame แต่แบบ Full Frame นั้นในขณะที่อ่านค่าสามารถประมวลผลภาพถัดไปได้ในอัตราการทำงานที่ต่างกัน



รูปที่ 2.7 Interline Architecture

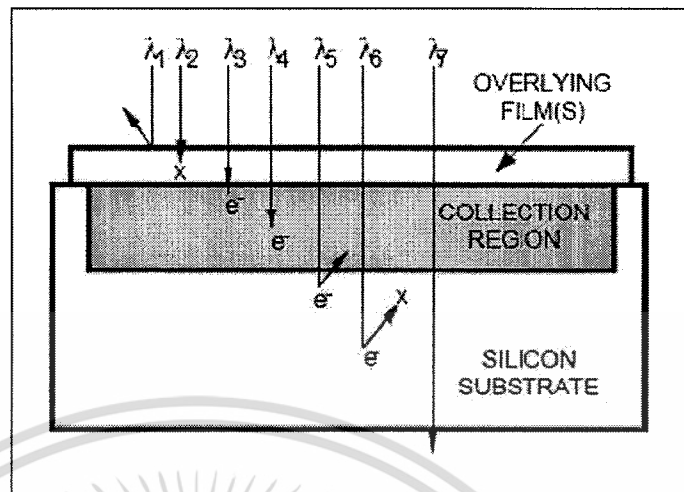
#### ข้อเสีย

- มีความซับซ้อนสูง ทำให้มีราคาสูงตามไปด้วย
- มีความไวต่ำ เพราะมีส่วนที่ไวแสงน้อย ต้องเสียพื้นที่ไปกับส่วนที่กั้นลำแสง
- เกิดความล่าช้าในการเปลี่ยนข้อมูลจาก Photodiode ไปยัง CCD

#### 2.3.4 CCD พื้นฐาน (CCD Basic)

การสร้างภาพของ CCD มี 3 ขั้นตอน

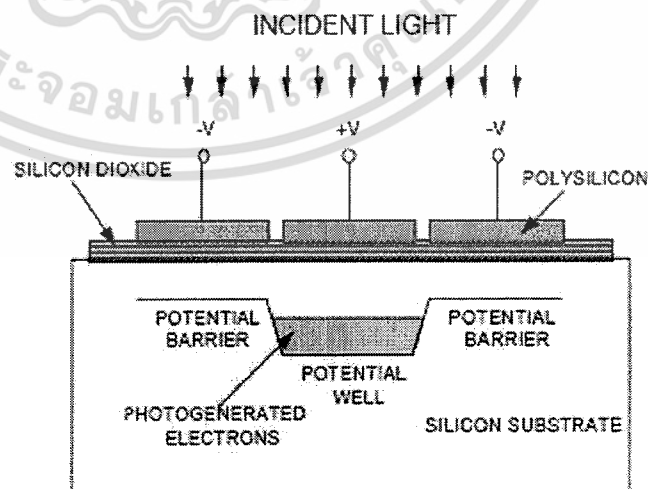
- 1) การเปลี่ยนจากแสง (Photo) เป็นประจุไฟฟ้า การทำงานเริ่มขึ้นเมื่อแสงในรูปแบบของ Photon ให้ตกลงไปยังภาพเป็น Array ของพิกเซล และเกิดการดูดซับ Photo โดยซิลิกอนและทำให้เกิดการตอบโต้ขึ้น เป็นการสร้างหลุมอิเล็กตรอน (Electron Hold) ดังรูปที่ 2.9



รูปที่ 2.8 Photon Interaction with Silicon

จำนวนของอิเล็กตรอน ที่ได้ในแต่ละพิเซกจะเป็นสัดส่วนโดยตรงกับพลังงานแสงและแสงที่ฉายลงไปและเป็น Non – Linearly กับความยาวคลื่น (Wave Length) กับหลายปัจจัยที่ส่งผลต่อการตรวจจับอิเล็กตรอน ดังนั้นจึงมีการฉาบฟิล์มบางๆบนผิวของซิลิกอนเพื่อลดการหักเหของแสงที่อาจเกิดขึ้นภายในตัวเก็บประจุที่เปลี่ยนแปลงตามปริมาณโฟตอนจะถูกดูดซับที่ความลึก (Depth) ต่างๆกันของซิลิกอนตามแต่ความยาวคลื่นของโฟตอนและบางส่วนไม่สามารถตรวจจับได้

1) Potential Wells and Barriers, CCD ถูกผลิตขึ้นตามหลักของ MOS (Metal Oxide Semiconductor) โครงสร้าง CCD จึงประกอบไปด้วยชั้นต่างๆซ้อนๆกันตามแนวตั้ง ดังรูปที่ 2.10 โดยจะมีชั้นของตัวนำวางอยู่บนสารกึ่งตัวนำซึ่งแยกออกจากกันโดยใช้ Silicon dioxide

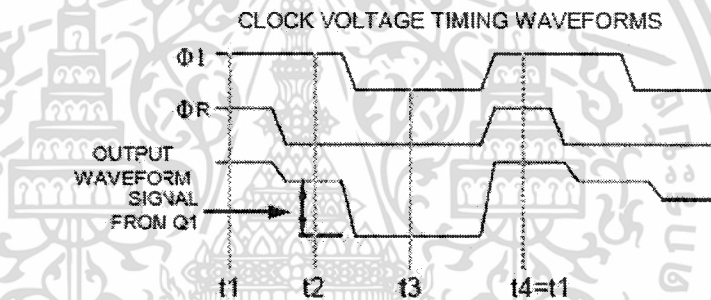


รูปที่ 2.9 Potential Well and Barriers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจ่ายแรงดันให้กับ Polysilicon gate จะทำให้ความต่างศักย์ของไฟฟ้าสถิตของ Silicon เกิดจากการเปลี่ยนแปลงทำให้เกิดความต่างศักย์ที่สามารถตรวจจับได้เมื่อมีการสร้างอิเล็กตรอนจากแสงที่ตกกระทบ (Incident light) ซึ่งจะเกิดขึ้นที่ gate เหล่านั้นถูกต้องไว้

2) Readout Techniques ประจุที่ได้จะถูกส่งเป็นผลลัพท์ไปยังอุปกรณ์ ซึ่งอิเล็กตรอนจะถูกเปลี่ยนเป็นแรงดันซึ่งจะง่ายขึ้นในการทำงานของ Chip ซึ่งเทคนิคในการเปลี่ยนนี้ ก็คือการใช้ floating diffusion sense node ตามด้วยการปรับเปลี่ยนแรงดันตามแหล่งจ่ายของวงจร ในการทำงานเริ่มจากการเคลียอุปกรณ์ผ่านทาง reset gate และ reset drain เพื่อเป็นการคายประจุจากนั้นแรงดันที่จ่ายเข้าไปใหม่จะไปทำการชาร์ตประจุอีกครั้งและทำการเปรียบเทียบกับแรงดันอ้างอิง และส่งไปยัง output ของอุปกรณ์ และหลักการทำงานนี้จะถูกเปลี่ยน Phase ไปยังสุดท้ายของ CCD ดังรูปที่ 2.11

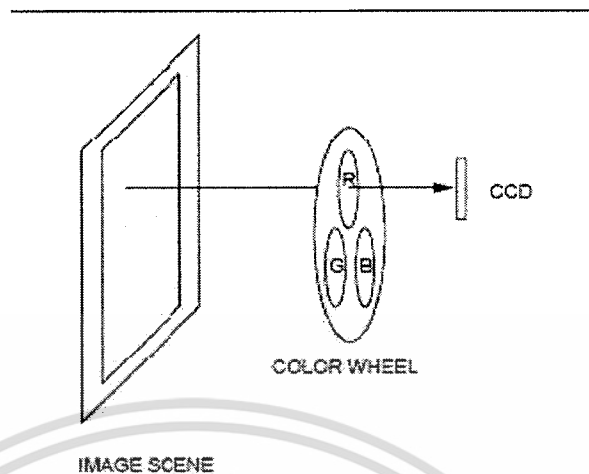


รูปที่ 2.10 Floating Diffusion Readout Structure

### 2.3.5 เทคโนโลยีของ CCD

2.3.5.1 การใช้ CCD ในการสร้างภาพ Monochrome (สีเดียว) ทำให้ไม่เป็นธรรมชาติในการที่จะระบุสีต่างๆ ได้แก่ แดง เขียว น้ำเงิน (RGB) ในแต่ละพิกเซลมี 3 วิธีการ ในการที่จะแยกสีต่างๆ ออกได้ แต่ปัญหาหลักๆก็คือ ต้องมีปริมาณข้อมูลเพิ่มมากขึ้นถึง 3 เท่า

1) Color Sequential ดังรูปที่ 2.12 วิธีนี้จะทำให้การฉายแสงสลับกันไปทีละสี โดยใช้ตัวกรองแสงตามแต่ละสีที่ต้องการ และนำข้อมูลที่ได้มาสร้างเป็นรูปใหม่



รูปที่ 2.11 Color Sequential capture

ข้อดี

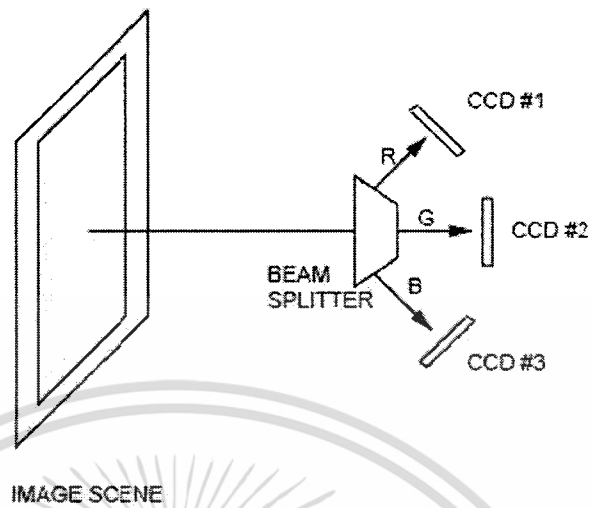
- คุณภาพยังคงเหมือนเดิมกับการใช้ CCD สีเดียว

ข้อเสีย

- การใช้แหล่งกำเนิดแสงสามแหล่งทำให้ลดจำนวนภาพในแต่ละหน่วยเวลาลงถึง 3 เท่า
- ตารางเพิ่มกลไกการสลับ Filter ของแสง เป็นการทำให้ระบบซับซ้อนขึ้น

2) Three-chip Color ดังรูปที่ 2.13 จะเป็นการใช้การสลับสี โดยใช้วงจร Filter การใช้ Three-chip Color จะใช้แสงในการแยกภาพออกมาเป็น 3 ภาพย่อย (ตามสีที่แยกได้) และให้ CCD ในแต่ละช่วงสีซึ่งวางไว้ในตำแหน่งที่เหมาะสม และอ่านค่าพร้อมๆกัน

วิธีที่ภาพจะถูกสร้างขึ้นโดยการ Synchronizing Output ระหว่าง CCD ถึง 3 แสงนำมาสร้างเป็นภาพเดียวกัน



รูปที่ 2.12 Three – chip Color Capture

ข้อดี

- อัตราการทำงานเท่ากับการใช้ CCD ชุดเดียว

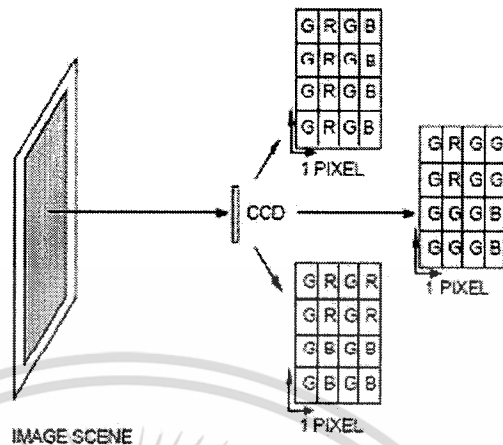
ข้อเสีย

- ความซับซ้อนสูงมาก

- การปรับค่าของ CCD แต่ละหน่วยเป็นไปได้ยาก

- Effective data rate เป็น 3 เท่า

3) Integral Data Filter Array (CFA) ดังรูปที่ 2.13 แทนที่จะใช้ในการสร้าง chip ซึ่งแยกสีต่างๆออกเป็น 3 สี วิธีนี้จะมีการสร้างคุณสมบัติพิเศษบน chip ให้สามารถตอบสนองต่อสีทั้ง 3 ได้ด้วยตนเองในแต่ละพิกเซล โดยการใช้ Photo resists ในรูปแบบต่างๆแต่การตรวจรับจะได้สีในรูปแบบ CMY (Cyan, Magenta, Yellow)

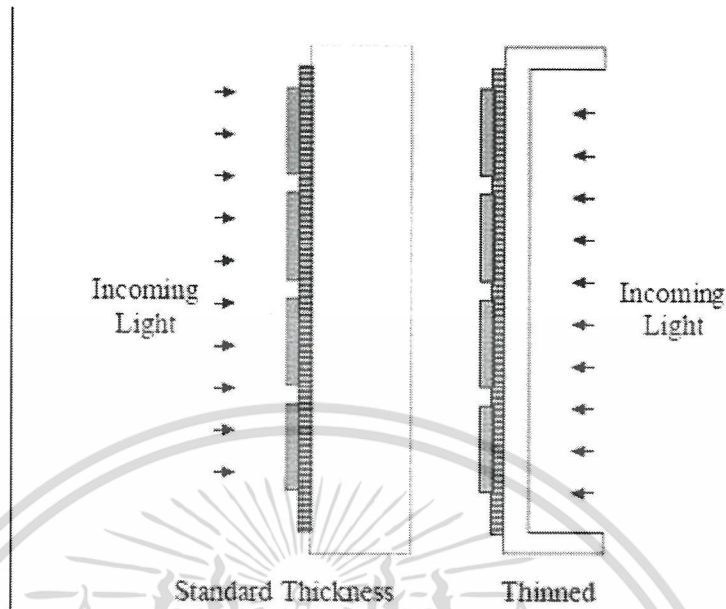


รูปที่ 2.13 Integral Color Filter Array Patterns

#### ข้อเสีย

- ไม่ได้สีเป็น RGB ทำให้ได้รูปไม่เป็นไปตามธรรมชาติ
- ข้อมูลสูญหายทำให้ Resolution ลดลงต้องเพิ่ม Sampling
- ต้องมีการสร้างพิกเซลที่มีข้อมูลของสีที่หายไปขึ้นมาใหม่

2.3.5.2) การทำให้ซิลิกอนบางลง (Silicon Thinning) ดังที่กล่าวมาแล้ว พิล์มบางๆจะช่วยในการดูดซับการสะท้อนแสงตามความยาวคลื่น แล้วอิเล็กตรอนจะถูกสร้างขึ้นด้านบนของพื้นผิว (ปกติจะเป็นแสงอุตราไวโอเลต และแสงสีน้ำเงิน) อิเล็กตรอนเหล่านี้จะเกิดการสูญหายไปโดยการทำปฏิกิริยากับพื้นผิวซิลิกอน ในการเพิ่มการตอบสนองให้กับ CCD นั้นจำเป็นจะต้องทำให้ด้านหลังของแผ่นเวเฟอร์บางลงซึ่งมีความหนาประมาณ 10 – 15 ไมโครเมตร ดังรูปที่ 2.15



รูปที่ 2.14 Normal and Thinned CCD

ข้อดี

- การตอบสนองของแสงสีน้ำเงินและอุตราไวโอเลตดีขึ้น

ข้อเสีย

- ราคาแพง
- การหยิบจับยาก

2.3.5.3) UV Enhancement Coating เนื่องจากการทำให้ซิลิกอนบางลงได้ยาก เพราะฉะนั้นจะมีการเพิ่ม Phosphors ลงไปที่ด้านบน CCD Phosphors นี้จะมีลักษณะใสและหนา ประมาณ 0.45 ไมโคร

ข้อดี

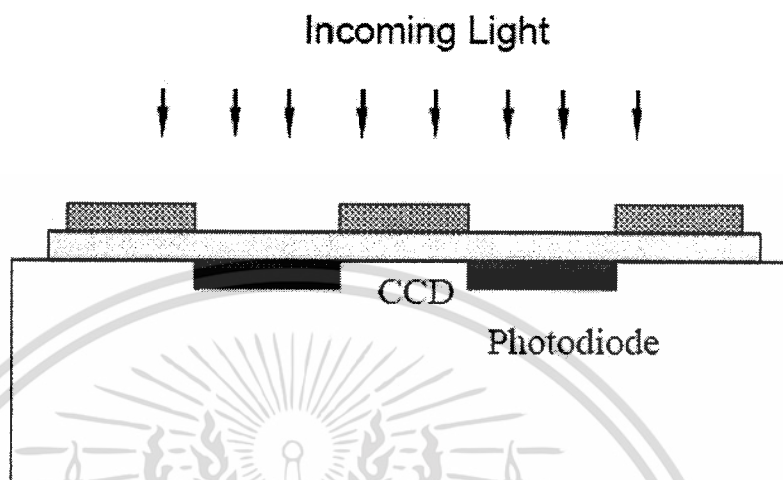
- จะเป็นตัวดูดซับ UV และแสงสีน้ำเงินให้มีช่วงความยาวคลื่นสูงขึ้น

ข้อเสีย

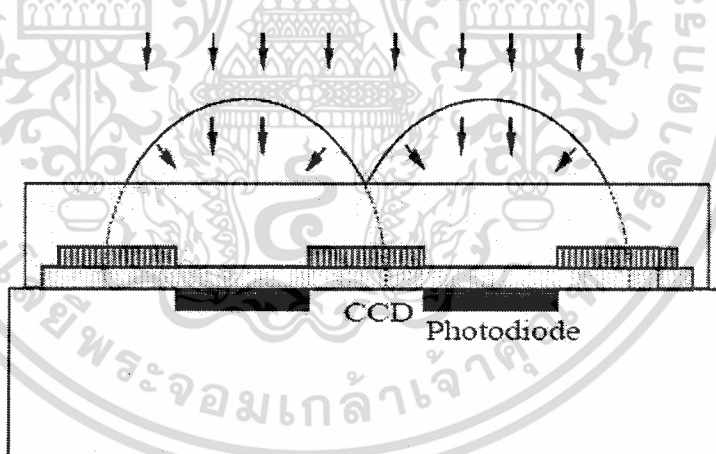
- ในกรณีที่เกิดการกระเจิงของแสงจะทำให้ข้อมูลสูญหายได้

2.3.5.4) Microlens Array เนื่องจากการอ่านโดยใช้เทคนิค Inline (IL) จะทำให้ความไวต่ำเนื่องจากการต้องมีการเพิ่มแถวของโฟโตไดโอดขึ้น ดังนั้นเพื่อปรับปรุงความไวแสงให้มากขึ้นจึงได้มีการสร้างเลนส์เล็กๆกร่อมขึ้นในแต่ละพิกเซล ซึ่งเลนส์เหล่านี้จะทำการรวมแสงที่ส่งไปยัง

อุปกรณ์ซึ่งไม่ไวต่อแสง ให้ส่งต่อไปยังบริเวณที่ไวต่อแสง (Photo Diode) เพียงอย่างเดียวดังรูปที่ 2.16 และ 2.17



รูปที่ 2.15 Interline CCD Showing Photodiode and Non Sensitive CCD covered by Light Shield



รูปที่ 2.16 Interline CCD with Microlenticular Array

ข้อดี

- มีความไวแสงเพิ่มขึ้นเป็น 3 เท่า

ข้อเสีย

- การบรรจุอุปกรณ์ทำได้ยาก
- การออกแบบเลนส์ทำได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5.5) CCD ความเร็วสูง (High Speed CCD) ในการพัฒนา CCD ให้สามารถรองรับอัตราการทำงานได้มากขึ้นนั้นจำเป็นต้องพัฒนา CCD ให้มีความไวสูงขึ้นตามไปด้วย เช่น

- 1) การสร้างวงจรถ่ายบนชิพ CCD ซึ่งจะต้องระมัดระวังไม่ให้ใช้พลังงานมากเกินไป เพราะจะทำให้คุณภาพของการอ่านข้อมูลลดลง
- 2) การทำให้ CCD มีหลายๆ Output โดยการแบ่งกลุ่มของ CCD ออกเป็นบล็อกแล้วสามารถทำให้การอ่านข้อมูลทำพร้อมๆกันได้ แต่จะต้องไม่มีจำนวน Output มากเกินไปจะทำให้การประมวลผลยุ่งยาก



## บทที่ 3

# โครงสร้างและหลักการทำงานของ FPGA

### 3.1 ความเป็นมาของเอซิก (ASIC)

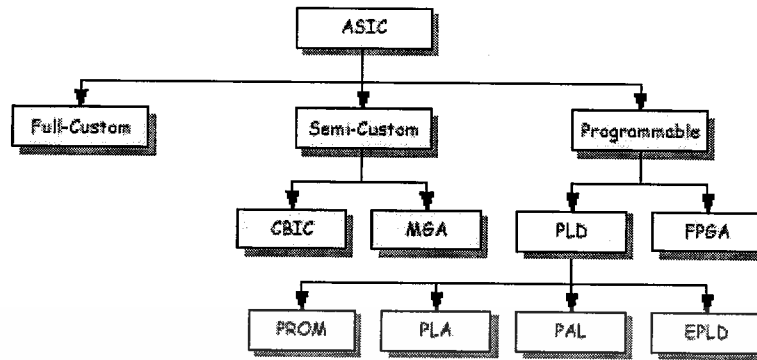
ในช่วงก่อนทศวรรษ 1970 อุตสาหกรรมเซมิคอนดักเตอร์ได้ถูกปลุกให้ตื่นตัวขึ้นหลังจากที่มีการประดิษฐ์วงจรรวมหรือไอซี ตัวแรกสำเร็จ ในยุคแรกนั้น ไอซีขนาดเล็กหรือ SSI (Small-Scale Integration) ประกอบไปด้วยเกทดิจิทัลจำนวนไม่มากนัก (ประมาณ 1 ถึง 10 ตัว) ต่อมาได้มีการเพิ่มปริมาณของเกทดิจิทัลและฟังก์ชันทางลอจิกให้มากขึ้นจนเป็น MSI (Medium-Scale Integration) การพัฒนาไอซีเป็นไปอย่างต่อเนื่องจนมาถึงยุคของ LSI (Large-Scale Integration) ซึ่งเป็นยุคที่มีการสร้างไมโครโปรเซสเซอร์ตัวแรกขึ้นและในปัจจุบันเป็นยุคของ VLSI (Very Large-Scale Integration) ซึ่งเทคโนโลยีในการสร้างไอซีรุ่นหน้างานสามารถสร้างไมโครโปรเซสเซอร์ขนาด 64 บิต ที่มีหน่วยความจำเท่ากับหน่วยคำนวณทางคณิตศาสตร์ ของโฟลตติงพอยน์ (Floating-Point Arithmetic Units) รวมอยู่ภายในตัวมันและเนื่องจากการปรับปรุงเทคโนโลยีของกระบวนการสร้างชิปที่มีมาอย่างต่อเนื่องทำให้ขนาดของทรานซิสเตอร์ที่บรรจุอยู่ในไอซีมีขนาดเล็กลงเรื่อยๆ จนบางคน โดยเฉพาะ ในญี่ปุ่นใช้คำว่า ULSI (Ultra large Scale Integration) เพื่อใช้เรียกระดับของไอซีในปัจจุบันแต่คนส่วนมาก ยังมักนิยมเรียกเพียงแค่ VLSI

จากการปรากฏตัวของ VLSI ในช่วงทศวรรษ 1980 ทำให้วิศวกรเริ่มมีการออกแบบไอซีตามความต้องการของลูกค้าซึ่งใช้ใน ระบบที่เจาะจงนอกเหนือจากการใช้ไอซีมาตรฐานเพียงอย่างเดียว โดยไอซีเหล่านี้มีชื่อเรียกว่า ASIC: Application-Specific Integrated Circuit (ออกเสียงว่า เอ-ซิก) ซึ่งตัวอย่างของ ASIC ได้แก่ชิปไอซีที่ใช้ สำหรับตุ๊กตาของเล่นพูดได้ ดาวเทียม และ ชิพที่ภายในบรรจุด้วยไมโครโปรเซสเซอร์กับอุปกรณ์ทางลอจิกอื่นๆ

### 3.2 ประเภทของ ASIC

ASIC แบ่งเป็น 3 ประเภทใหญ่ๆ คือ Full-custom, Semi-custom และ Programmable ดังรูป

#### ที่ 3.1



รูปที่ 3.1 ประเภทของ ASIC

### 3.2.1 Full-custom

ASIC ประเภทนี้ลูกค้าจะเป็นผู้ออกแบบเซลล์ลอจิก (เช่น แอนด์เกต ออร์เกต มัลติเพล็กซ์ และฟลิปฟล็อป) และลักษณะ การจัดวางอุปกรณ์บนตัวไอซีรวมถึงหน้ากาสำหรับควบคุม การเจือและสร้างชั้นสาร (Mask) ต่างๆ ที่ ใช้ในการทำไอซีเอง ดังนั้นค่าใช้จ่ายในการออกแบบและ การผลิตจะสูงมาก

### 3.2.2 Semi-custom

ASIC ประเภทนี้เซลล์ลอจิกจะถูกออกแบบเอาไว้ก่อนแล้วในรูปแบบของไลบรารีและ ลูกค้าจะเป็นผู้ออกแบบ Mask ต่างๆเอง ตัวอย่างของไอซีประเภทนี้ได้แก่ Standard-Cell-Based ASIC และ Masked Gate-Array-Based ASIC

#### -Standard-Cell-BasedASIC

ไอซีประเภทนี้จะมีพื้นที่สำหรับจัดวางเซลล์ลอจิกมาตรฐานซึ่งถูกออกแบบเอาไว้แล้ว ในบางครั้ง เซลล์ มาตรฐานเหล่านี้จะถูกนำมาประกอบกันเป็นเซลล์ที่มีขนาดใหญ่ขึ้นเรียกว่า Mega cell สำหรับการ ออกแบบนั้นผู้ออกแบบจะทำเพียงแค่กำหนดตำแหน่งของเซลล์มาตรฐานและการเชื่อมต่อ ภายในของแต่ละเซลล์เท่านั้นแต่อย่างไรก็ดีเซลล์ต่างๆ เหล่านี้สามารถวางที่ตำแหน่งใดๆ ก็ได้บน แผ่นเวเฟอร์ซิลิกอน นั่นก็หมายความว่าชั้น Mask จะถูกจัดวางตามความต้องการของผู้ออกแบบ

#### -MaskedGate-Array-BasedASIC

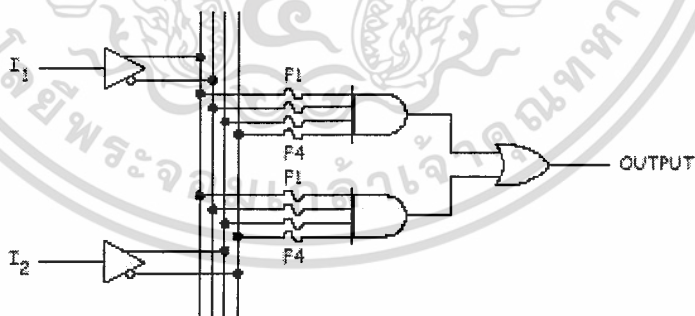
ไอซีชนิดนี้จะมีทรานซิสเตอร์หรือเกตถูกสร้างมาในลักษณะของอะเรย์สองมิติบนแผ่นเวเฟอร์ ซิลิกอน และผู้ออกแบบจะทำการออกแบบ Mask เพื่อใช้สำหรับกำหนดการต่อเชื่อมของ ทรานซิสเตอร์แต่ละตัว

### 3.2.3 Programmable

ASIC ประเภทนี้เซลล์ลอจิกจะถูกออกแบบไว้ก่อนเช่นเดียวกับ Semi-Custom แต่ชั้นของ Mask จะไม่สามารถเปลี่ยนแปลงได้ ตามความต้องการของผู้ออกแบบ ไอซีประเภทนี้ยังแบ่งออกเป็น 2 ชนิด คือ Programmable Logic Device (PLD) และ Field Programmable Gate Array (FPGA)

#### 1) Programmable Logic Device (PLD)

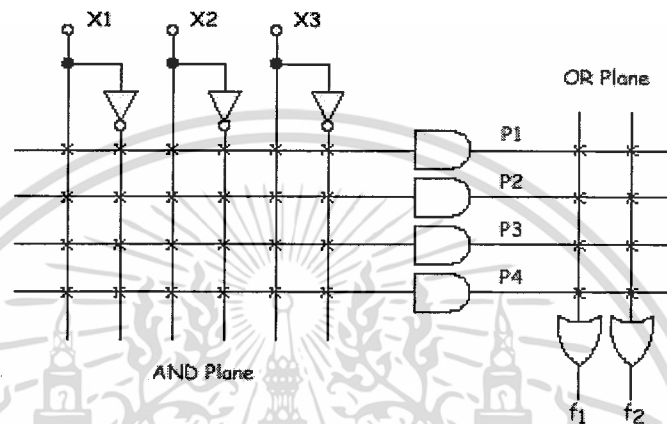
มีโครงสร้างภายในเป็นวงจรพื้นฐานทางด้านลอจิกต่อกันอยู่เป็นกลุ่มซึ่งมีทั้งวงจรคอมบิเนชัน(Combination) และซีควเอนเชียล (Sequential) สำหรับเทคโนโลยีของวงจรที่ใช้สร้าง PLD จะมีทั้ง TTL, ECL และ CMOS ตามความเหมาะสมของแต่ละระบบ ไอซี PLD ทุกชนิดมีหลักการพื้นฐานของวงจรภายในที่เหมือนกันโดยมี วงจรคอมบิเนชันที่เป็นผลคูณร่วม บวก (Sum of product) ประกอบไปด้วยชุดของแอนด์เกตต่อร่วมกับออร์ เกทและในการ โปรแกรมจะเป็นการเลือกว่าอินพุทภายในของแอนด์เกตกับสัญญาณอินพุทใดบ้างที่จะต้อง ต่อ ถึงกันซึ่งมีทั้งจากภายนอกและสัญญาณป้อนกลับจากเอาต์พุทภายในเอง เช่น การติดต่ออินพุทของออร์ เกทกับเอาต์พุทของแอนด์เกตตัวต่างๆ สำหรับการ โปรแกรมทางกายภาพนั้นอินพุท ต่างๆ ของอุปกรณ์ทุกตัว จะถูกต่อผ่านฟิวส์เข้ากับแหล่งสัญญาณ ซึ่งถ้าไม่ต้องการใช้สัญญาณ ใดก็จะตัดฟิวส์ตัวนั้นทิ้งทำให้สามารถ โปรแกรมได้เพียงครั้งเดียว ไอซี PLD บางชนิดใช้ มอสทรานซิสเตอร์แทนฟิวส์ทำให้สามารถโปรแกรมโดยใช้กระแสไฟฟ้าและสามารถลบแล้ว โปรแกรมเข้าไปใหม่ได้อีก สำหรับ ไอซีในตระกูล PLD ได้แก่ PROM, PAL, PLA และ EPLD



รูปที่ 3.2 วงจรพื้นฐานภายในของ PLA

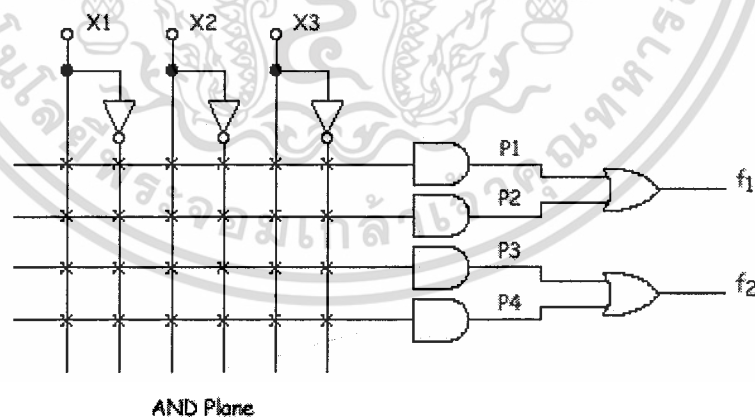
PROM (Programmable Read Only Memory) PROM คือหน่วยความจำประเภท ROM ซึ่งนับว่าเป็นไอซี PLD ชนิดหนึ่งซึ่งวงจรภายใน ของ PROM ประกอบไปด้วยอะเรย์ของแอนด์และออร์เกต (And - Or Array) ผลลัพธ์ที่ขา ดาต้าเอาต์พุทสามารถแสดงได้ในสมการของฟังก์ชันผลคูณร่วมบวก (Sum of product) ของสัญญาณอินพุทที่ขาแอดเดรส

PLA (Programmable Logic Array) ลักษณะเด่นของ PLA คือสามารถโปรแกรมการเชื่อมต่อได้ทั้งทางด้าน AND และด้าน OR ทำให้มีความยืดหยุ่นในการใช้งานมาก แต่อย่างไรก็ดีข้อเสียที่เห็นได้อย่างชัดเจน ของ PLA คือความยุ่งยากในการสร้างและคุณสมบัติทางด้านความเร็วที่ลดลงเนื่องจาก สัญญาณจะต้องวิ่งผ่านอะเรย์ของ AND และ OR



รูปที่ 3.3 วงจรพื้นฐานภายในของ PLA

PAL (Programmable Array Logic) PAL มีลักษณะโครงสร้างที่ใกล้เคียงกับ PROM และ PLA มาก แต่การ โปรแกรม PAL จะสามารถทำได้เพียงด้าน AND เท่านั้น



รูปที่ 3.4 วงจรพื้นฐานภายในของ PAL

EPLD (Erasable Programmable Logic Device) EPLD เป็นอุปกรณ์ที่สามารถทำการโปรแกรมได้หลายครั้งซึ่งเหมาะสำหรับการ ทำวงจรถนัแบบ สำหรับเทคโนโลยีที่ใช้ในการสร้างจะเหมือนกับ CMOS EPROM คือ ใช้มอสทรานซิสเตอร์เชื่อมต่อระหว่างสัญญาณอินพุตกับจุดที่

ต้องการแทนการ ใช้ฟิวส์แบบเดิมทำให้สามารถโปรแกรมการต่อวงจรภายในอุปกรณ์ด้วยการจ่าย ไฟฟ้าตามขนาดที่กำหนดไว้และลบได้โดยใช้แสงอัลตราไวโอเลตฉาย ผ่านช่อง หน้าต่างกระจก ของตัวชิพ

## 2) Field-Programmable Gate Array (FPGA)

เป็นอุปกรณ์ที่มีความซับซ้อนมากกว่า PLD ไปอีกระดับหนึ่ง ซึ่งในความเป็นจริงแล้ว PLD และ FPGA แตก ต่างกันน้อยมาก สำหรับ FPGA แล้วนับว่าเป็นอุปกรณ์ตัวใหม่ในตระกูลของ ASIC ซึ่งมีการเจริญเติบโตอย่างรวดเร็วและมีบทบาทที่สำคัญในการเข้ามาแทนที่ระบบ อิเล็กทรอนิกส์ที่ใช้ TTL โครงสร้าง ภายในของ FPGA ประกอบไปด้วยอะเรย์ของลอจิกเกตต่างๆ มากมาย ซึ่งในปัจจุบันความจุเกตภายใน ตัวชิพ FPGA ได้เพิ่มขึ้น จากระดับไม่กี่พันตัวจนถึงระดับ ล้านตัวซึ่งสามารถรองรับวงจรดิจิทัลที่มีความ สลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการ ออกแบบพัฒนาและทดสอบก็ทำได้ง่ายซึ่งในปัจจุบัน การออกแบบวงจร โดยใช้ FPGA กำลังเป็นที่ นิยมและมีแนวโน้มที่จะนำมาใช้งานมากขึ้นเรื่อยๆ

### 3.3 หลักการของFPGA

ในปัจจุบันมี FPGA อยู่ 4 ชนิดที่วางขายอยู่ในท้องตลาดได้แก่ Symmetrical Array, Row-Based, Hierarchical PLD และ Sea-of-Gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการ โปรแกรม ที่แตกต่างกันไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ ใช้ในการโปรแกรม ซึ่งมีอยู่ 2 แบบคือ การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทาง กายภาพของตัวชิพ และการ โปรแกรม โดยการใช้หน่วยความจำ

#### 3.3.1 การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

- Fuse เป็นวิธีการ โปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากที่โปรแกรมแล้วจุด เชื่อมต่อจะขาดจากกัน

- Anti Fuse เป็นวิธีการ โปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการ โปรแกรม แล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

#### 3.3.2 การโปรแกรมโดยใช้หน่วยความจำ

##### - EEPROM Based FPGA

FPGA ที่ใช้การ โปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกตต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกต แต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการ โปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิต ซึ่งการ โปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

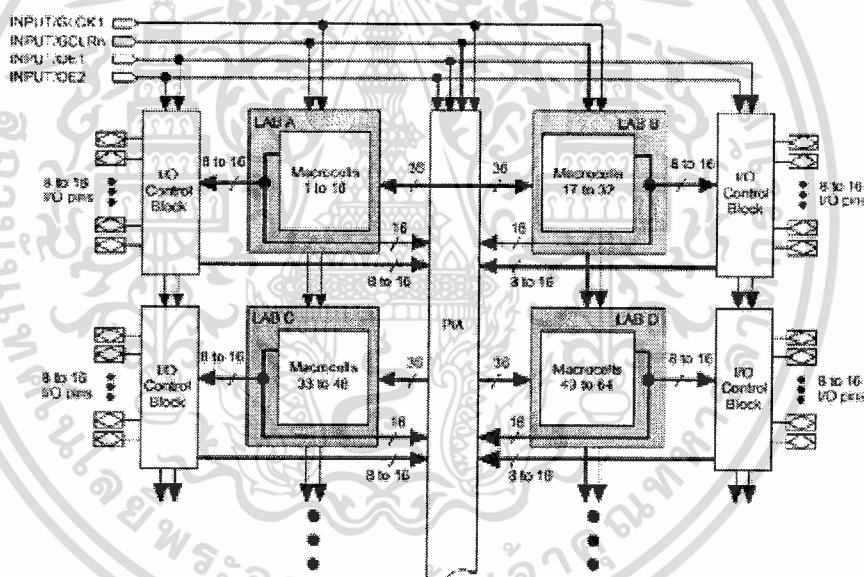
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - SRAM Based FPGA

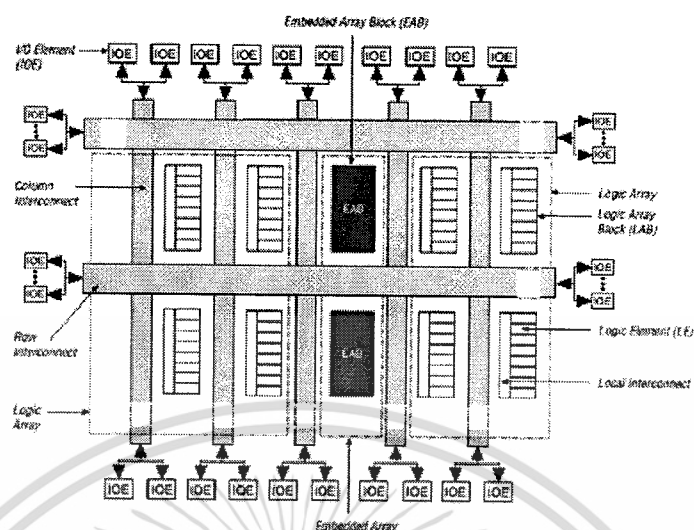
FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถโปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกตในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกต) ซึ่งข้อดีของ SRAM Based FPGA คือใช้เวลาในการโปรแกรมน้อย (ระดับ nsec) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิปในขณะที่เริ่มต้นใช้งาน

### 3.3.3 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอะเรย์ของบล็อกลอจิกที่สามารถทำการโปรแกรมได้ดังรูปที่ 3.5 และ 3.6



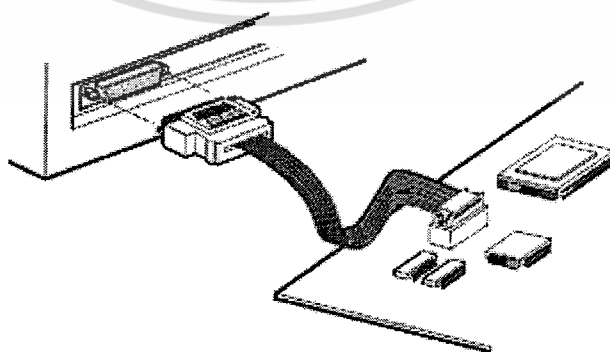
รูปที่ 3.5 โครงสร้างภายในของ FPGA ตระกูล MAX7000S



รูปที่ 3.6 โครงสร้างภายในของ FPGA ตระกูล FLEX10K

### 3.3.4 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว

- ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพ เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว
- มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจร หรือ HDL (Hardware Description Language) เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and Optimize ให้ทั้งหมด
- การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของ คอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้



รูปที่ 3.7 การโปรแกรมลงชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5 การออกแบบวงจร

การออกแบบวงจรด้วย FPGA โดยการใช้ Schematic หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำให้สะดวกกว่า เนื่องจากวิธีการนี้ผู้ออกแบบไม่จำเป็นต้องคำนึงถึงเทคโนโลยีที่จะใช้สร้างไอซีและที่สำคัญ การออกแบบโดยวิธีนี้สามารถแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยีสำหรับภาษาที่ใช้ สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL, AHDL และ Verilog เป็นต้น ส่วนรายละเอียดของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

#### 1.) การสังเคราะห์วงจร (Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรมของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์ นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น FPGA ของบริษัท XILINX และบริษัท ALTERA จะมีซอฟต์แวร์หลายตัวที่สามารถใช้ได้ เช่น Max Plus II ในขั้นตอนนี้ ซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตาม เทคโนโลยีที่เลือกใช้ในการสังเคราะห์วงจรนั้นวงจรระดับเกต (Gate Level) จะไม่เหมาะสมกับโครงสร้างที่มีอยู่ในอุปกรณ์ FPGA ดังนั้นในการ Optimize ซอฟต์แวร์สังเคราะห์วงจร จะต้องทำการ Optimize ให้ได้เป็นวงจรที่ประกอบด้วยกลุ่มของลอจิกที่เหมาะสมกับอุปกรณ์ FPGA นั่นๆจึงทำให้ผล ที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดล แต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจร ที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับ เทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอฟต์แวร์การสังเคราะห์วงจรก็จะมีรายงานผลว่าโมเดลที่ออกแบบไปนั้น เป็นอย่างไร เช่น มีค่าความหน่วง (Delay) เท่าใด ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

#### 2.) การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในคอนทำการเชื่อม ต่อสัญญาณ (routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำโดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกท (gate), ฟลิป-ฟลอป (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGA หลังจากทำขั้นตอน นี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงภายในแต่ละส่วนเท่านั้น หรือที่เรียกว่าความ หน่วงลอจิก(logic delay) ส่วนซอฟต์แวร์จะรวมเอาซอฟต์แวร์ย่อยอื่นๆ อีก เพื่อให้การทำ PPR (Partitioning Placement & Routing) เป็นไปอย่างต่อเนื่อง

### 3.) การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นทางเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควร จะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กัน เพื่อจะ ได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมี ความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณ ได้ไม่หมด การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆ ให้อยู่ใกล้กัน โดยเฉพาะส่วน ที่มีการเชื่อมต่อสัญญาณด้วยกันนอกจากนั้นการกำหนดตำแหน่งขา I/O (I/O pin) ตามตำแหน่งขา I/O ของ FPGA บนแผ่น PCB ก็จะมีผลโดยตรงเลยคือซอฟต์แวร์จะวาง I/O ลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่ง บางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา I/O ควรกำหนดตำแหน่งให้เหมาะสม หรือ ไม่ก็ให้ซอฟต์แวร์จัดการเอง

### 4.) การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็น การเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA ขั้นตอนนี้จะทำ ต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณ ได้ไม่หมด (เนื่องจากจำนวนทรัพยากรสำหรับเชื่อมต่อสัญญาณนั้นมีอยู่จำกัด) หรือเกิดความหน่วงเกิน ค่าที่กำหนดในข้อบังคับ ผู้ออกแบบสามารถทำ

ขั้นตอนนี้ได้โดยใช้ซอฟต์แวร์หรือผู้ออกแบบจะทำการ เชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับ ทางเวลา จะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

### 5.) ความหน่วงด้านเวลา (Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วง ที่เกิดขึ้นนั้นแยกได้เป็นสองประเภทคือ

- ความหน่วงลอจิก (Logic delay) เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เอง
- ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ (Routing delay)

โดยปกติแล้ว ค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณมักจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับบางเวลาเพื่อให้ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น

### 6) การจำลองการทำงานของวงจร (Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการทำงานของโมเดลว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้ จะมีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจร ควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดล เกิดขึ้นตรงไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้เกิดข้อผิดพลาด นั่นคือการทำจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจ สอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจร แล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตาม ข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้อง

ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์ การเชื่อมต่อสัญญาณ (Post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจร ในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้อง ตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวม

สามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงาน ของวงจรควรรู้ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่างๆ

#### 7) การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็นข้อมูลวงจร (configuration data) ซึ่งอยู่ในรูปของบิตสตรีม (bit stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการ ซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับอุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือ ในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้น ในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ serial PROM ด้วยเพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานโมเดลครั้งต่อไปไม่ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยใช้วิธี EPROM หรือ Anti fuse ก็ไม่จำเป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

#### 3.3.6 เครื่องมือสำหรับการออกแบบ FPGA

จะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้เวลาน้อยกว่ามากด้วย ส่วน สำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่เขียนโค้ดอธิบายฮาร์ดแวร์ จนกระทั่งดาวน์โหลดลงใน อุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็น ซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้ สำหรับซอฟต์แวร์ที่ใช้ทำการ จำลองการทำงานของวงจรมัน ต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบ เพราะโมเดลที่ได้จาก การทำขั้นตอนต่างๆ (ด้วยซอฟต์แวร์ต่างๆ ต้องเอามาจำลองการทำงานได้ และในการจำลองการทำงานของ วงจรควรรู้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่ จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์คสเตชัน (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ ที่ใช้บนพีซี (PC) มากขึ้นซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

### 3.4 ภาษา VHDL

ความซับซ้อนและขนาดของระบบดิจิทัลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะ ส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยใน การออกแบบหรือ CAD มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นกัน อีกทั้งอุปกรณ์และวิธีการ ออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ (HDL : Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่อง เพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิทัลเป็นไปอย่างมีประสิทธิภาพ

### 3.4.1 ประวัติความเป็นมาของภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิทัล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และสามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษา VHDL เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปที่ละชั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกร ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น VHDL ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิทัลที่มีความซับซ้อนได้ทั้งหมด ดังนั้น VHDL จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกา หรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้น ประกอบกับเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้ จากการนำวงจรดิจิทัลหลายๆ วงจรมาทำการผลิตอยู่บนแผ่นซิลิกอนที่มีพื้นที่เพียง 1 - 2 ตารางเซนติเมตรเท่านั้น ซึ่ง เป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือในการทำงานและความคงทนต่อสภาพ แวกล่อมสูง แต่เนื่องจากในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกร และเวลาในดำเนินการมาก ฉะนั้นทาง DoD จึงจัดตั้งโครงการขึ้นมาเพื่อศึกษาวิธีการที่ช่วยในการพัฒนา วงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิทัล ให้สามารถนำไปผลิตได้เร็วขึ้น ซึ่งโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ VHSIC โดยในระยะแรกนั้น โครงการนี้ถือเป็นความลับทาง ด้านความมั่นคงของประเทศ และอยู่ภายใต้ความควบคุมดูแลของ United States International Traffic and Arms Regulations (ITAR)

สำหรับมาตรฐานของภาษาที่ใช้บรรยาย พฤติกรรมวงจรหรือฮาร์ดแวร์ของระบบ สำหรับโครงการ VHSIC ที่ DoD ได้ให้ไว้สามารถสรุปได้ดังนี้

- 1) ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิทัล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่อง คอมพิวเตอร์ โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
  - 2) สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
  - 3) ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร
- ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือ ภาษาซี ซึ่งในทางวิศวกรรม ภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า "Hardware Description Language" หรือ HDL

ในตอนเริ่มแรกนั้น DoD ได้มอบหมายให้บริษัท ไอบีเอ็ม เท็กซัสอินสตุเมนต์ และอินเตอร์เมทริกซ์ เป็นผู้ศึกษาและพัฒนา โครงการ ซึ่งการดำเนินงานเป็นไปอย่างต่อเนื่อง จนกระทั่งในปี ค.ศ.1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอด เทคโนโลยีทางทหารออกจากโครงการนี้ ดังนั้นภาษา VHDL จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป และประมาณปี ค.ศ. 1987 IEEE ได้ทำการกำหนดมาตรฐานของภาษานี้เป็น IEEE 1076-1987 และมีชื่อเรียกว่า VHDL ซึ่งมาตรฐานนี้ได้รับการปรับปรุงจนเป็นมาตรฐาน IEEE 1076-1993 หรือ VHDL 1993

เนื่องจากในขณะนั้น DoD เป็นลูกค้ารายใหญ่ ของอุตสาหกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ ดังนั้นจึงมีผู้รับโครงการต่างๆ จาก DoD ไปดำเนินการวิจัยและพัฒนา เป็นจำนวนมาก และเพื่อให้ทุกโครงการอยู่ในมาตรฐานเดียวกันหมด ดังนั้นทาง DoD จึงได้กำหนดว่า ทุกๆ โครงการต้อง เขียนอยู่ในรูปของภาษา VHDL เท่านั้น ซึ่งทำให้ DoD สามารถนำโครงการเหล่านี้ไปจำลองกับเครื่องคอมพิวเตอร์ได้ หลายๆระบบ

### 3.4.2 ข้อกำหนด

DoD ได้ตั้งข้อกำหนดสำหรับภาษา VHDL ในเดือนมกราคมปี ค.ศ.1983 ไว้ดังนี้

#### 1.) ลักษณะทั่วไป

DoD ได้กำหนดให้ VHDL เป็นภาษาสำหรับการออกแบบและบรรยายของฮาร์ดแวร์ ซึ่งหมายถึงความสามารถ ในการอธิบายและออกแบบในระดับสูง การจำลอง (Simulation) การสังเคราะห์ (Synthesis) และการทดสอบ (Testing) นอกจากนั้น VHDL ยังถูกกำหนดไว้สำหรับการบรรยายฮาร์ดแวร์ตั้งแต่ระดับบนซึ่งก็คือระบบจนถึง ระดับเกทอีกด้วย

เนื่องจากการทำงานของระบบดิจิทัลนั้น ทุกๆ องค์ประกอบภายในระบบไม่ว่าเล็กหรือใหญ่ จะทำงานไปพร้อมๆ กัน ซึ่งในเรื่องของความพร้อมเพรียงในการทำงานนี้ก็ถือเป็นข้อกำหนดที่สำคัญอย่างหนึ่งของ VHDL ด้วยเช่นกัน (สำหรับในภาษาที่ใช้ในการบรรยายฮาร์ดแวร์นั้นความพร้อมเพรียงจะหมายถึงทุกๆ คำสั่ง องค์ประกอบ เกทหรือวงจรลอจิกจะถูกนำมาปฏิบัติทั้งหมด ดังนั้นในที่สุดแล้วก็จะดูเหมือนว่าได้มีการปฏิบัติไป พร้อมๆ กัน)

## 2.) สนับสนุนการออกแบบลำดับชั้น

การออกแบบแบบลำดับชั้นเป็นลักษณะที่สำคัญอย่างหนึ่งสำหรับการออกแบบระบบที่มีหลายๆ ระดับ โดยในการ ออกแบบจะประกอบด้วยส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายหน้าที่การทำงาน ซึ่งหน้าที่การทำงาน ของระบบสามารถกำหนดได้ด้วยตัวเอง หรืออาจถูกกำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลง ไปได้เช่นกัน แต่ที่ระดับล่างสุด องค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวมันเอง และไม่สามารถกำหนด การทำงานโดยลักษณะแบบโครงสร้างได้

## 3.) ไลบรารี

VHDL ได้สนับสนุนการมีไลบรารีเพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของ อุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูก ต้องควรจัดเก็บไว้ในไลบรารีหลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้วเพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไป ใช้ได้ด้วย

## 4.) ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่งหรือกระบวนการ โดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ VHDL ก็ตาม ตัวภาษา เองก็ยังมี การจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่ง ไว้ให้ด้วย เมื่อผู้ออกแบบได้กำหนดหน้าที่และองค์ประกอบ ที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออกแบบยังสามารถบรรยายหน้าที่การทำงานซึ่งเป็นรายละเอียดภายใน ของแต่ละองค์ประกอบได้ในลักษณะเดียวกับการเขียนโปรแกรมที่ประกอบด้วยโครงสร้างแบบ case, if - then - else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์ กระทำได้ สะดวกและง่ายขึ้น อย่างไรก็ตาม โครงสร้างทั้งหมดของ VHDL ก็ยังคงเป็นการทำงานแบบพร้อมเพรียงกันเช่นเดิม

## 5.) การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุตและเอาต์พุตแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของ อุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อมและลักษณะทางกายภาพของอุปกรณ์ นั้นๆ ด้วย ซึ่งภาษาสำหรับการออกแบบที่ดีควร ให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลา โหลด และเงื่อนไขทางสภาพแวดล้อมอื่นๆ ซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา VHDL ด้วยเช่นกัน

#### 6.) ชนิดของข้อมูล

VHDL สามารถกำหนดชนิดของข้อมูลไม่เพียงแต่ชนิด BIT และ BOOLEAN เท่านั้น แต่ยังสามารถกำหนดชนิดของข้อมูลเป็นจำนวนเต็ม จำนวนจริง จุดทศนิยม และชนิดลำดับการนับ (Enumerate Type) หรือแม้แต่ชนิดของ ข้อมูลที่ผู้ออกแบบกำหนดขึ้นมาเองก็ได้

#### 7.) โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชันและ โพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน VHDL ซึ่งผู้ออกแบบ สามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การกำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการได้เช่นเดียวกับการเขียนโปรแกรมทั่วไป

#### 8.) การควบคุมเวลา

VHDL อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการ การตรวจสอบ การออกแบบเกทหรือการหน่วงเวลาก็สามารถกระทำได้โดยการกำหนดช่วงเวลาที่เหมาะสมหรือกำหนดให้มีการรอคอย เหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

#### 9.) การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบ โดยการกำหนดโครงสร้างขององค์ประกอบรวมที่เกิดจากองค์ประกอบย่อยซึ่งแตกต่างกันหรือ เหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ VHDL เช่นกัน

### 3.4.3 องค์ประกอบพื้นฐานของ VHDL

รูปแบบพื้นฐานที่ใช้ในการบรรยายถึงองค์ประกอบของ VHDL จะประกอบไปด้วยส่วนกำหนดการเชื่อมต่อ (Interface) และส่วนกำหนดลักษณะเชิงสถาปัตยกรรม (Architecture) ดังแสดงในรูปที่ 5.3 โดยในการบรรยายการเชื่อมต่อจะขึ้น ต้นด้วยคำว่า ENTITY แล้วตามด้วยชื่อขององค์ประกอบจากนั้นตามด้วยคำว่า IS และถัดมาจะเป็นการบรรยายถึงพอร์ต การติดต่อ อินพุต - เอาท์พุท ขององค์ประกอบ ส่วนลักษณะภายนอกอื่น ๆ เช่น เวลา อุณหภูมิก็สามารถรวมเข้าไปในส่วนนี้ได้เช่นกัน ในส่วนของการกำหนดลักษณะเชิงสถาปัตยกรรมจะขึ้นต้นด้วยคำว่า ARCHITECTURE ซึ่งเป็นส่วนที่ใช้ บรรยายหน้าที่การทำงานขององค์ประกอบ โดยหน้าที่การทำงานนี้จะขึ้นอยู่กับสัญญาณอินพุต เอาท์พุทและพารามิเตอร์ อื่นๆ ที่ได้กำหนดไว้ในส่วนของการเชื่อมต่อดังรูปที่ 3.8 และสำหรับการบรรยายหน้าที่ขององค์ประกอบจะเริ่มต้นหลังจาก คำว่า BEGIN เป็นต้นไป

```

ENTITY component_name IS
    Input and output ports
    Physical and other parameters
END [component_name] ;

ARCHITECTURE identifier OF component_name IS
    [declaration]
BEGIN
    specification of the functionality of the component
    in terms of its input lines and as influenced
    by physical and other parameters
END [identifier];

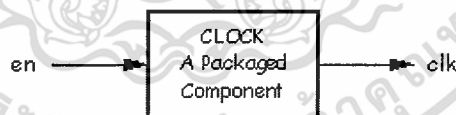
```

รูปที่ 3.8 การกำหนดการเชื่อมต่อและสถาปัตยกรรม

### 1.) การกำหนดการเชื่อมต่อ

การกำหนดการเชื่อมต่อเป็นระดับบนสุดของการออกแบบ โดยในระดับนี้ต้องกำหนดพอร์ตสำหรับการติดต่อกับองค์ประกอบ ภายนอกอื่นๆ ดังตัวอย่างในรูปที่ 5.4 ซึ่งเป็นบล็อกไดอะแกรม และการบรรยายการเชื่อมต่อขององค์ประกอบสำหรับตัวจ่าย สัญญาณนาฬิกา

ในบรรทัดแรกของการบรรยายการเชื่อมต่อเป็นการกำหนดชื่อขององค์ประกอบซึ่งกำหนดเป็น Clock\_component ตามด้วยคำว่า PORT และชื่อของพอร์ตที่อยู่ในวงเล็บ ส่วน IN และ OUT เป็นการกำหนด โหนดของสัญญาณให้เป็นอินพุทหรือเอาต์พุท และ BIT เป็นการแสดงชนิดของข้อมูล



```

ENTITY clock_component IS
    PORT (en : IN BIT; ck : OUT BIT)
END clock_name;

```

รูปที่ 3.9 บล็อกไดอะแกรมและการบรรยายการเชื่อมต่อของ clock component

### 2.) การกำหนดรูปแบบการบรรยาย

หน้าที่การทำงานขององค์ประกอบจะถูกบรรยายภายในส่วนนี้ ซึ่งในการบรรยายสามารถกำหนดค่าของสัญญาณ เอาต์พุทในเทอมของอินพุทหรือในรูปขององค์ประกอบอื่นๆ หรือทั้งสอง

อย่างรวมกันก็ได้ ดังตัวอย่างการบรรยายของ Clock\_component ในรูปที่ 5.5 ซึ่งเป็นการบรรยายในเชิงพฤติกรรม โดยมี en เป็นอินพุทและ ck เป็นเอาต์พุท

PROCESS เป็นคำที่ใช้ในการเริ่มต้นสำหรับการบรรยายในเชิงพฤติกรรม และภายในโปรเซสกำหนดให้ periodic เป็นตัวแปรที่มีค่าเริ่มต้นเป็น "0" ถ้าสัญญาณ en มีค่าเป็น "1" จะทำให้ตัวแปร periodic ถูกคอมพลิเมนต์ (complement) และส่งค่าให้กับ ck ซึ่งเป็นสัญญาณเอาต์พุท และสำหรับคำสั่ง WAIT จะเป็นการกำหนดให้สัญญาณมีคาบเวลาเท่ากับ 1 ไมโครวินาที

```

ARCHITECTURE behavioral OF clock_component IS
BEGIN
  PROCESS
    VARIABLE periodic : BIT := '0';
  BEGIN
    IF en='1' THEN
      periodic := Not periodic;
    END IF;
    ck <= periodic;
    WAIT FOR 1 US;
  END PROCESS;
ENDbehavioral;
  
```

รูปที่ 3.10 การบรรยายเชิงพฤติกรรมของ clock component

### 3.) หน่วยการออกแบบแพ็คเกจ

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบการบรรยายระบบดิจิทัล สามารถเก็บไว้ใน ส่วนของแพ็คเกจ ซึ่งหน่วยการออกแบบต่างๆ เช่น หน่วยการออกแบบ Entity หน่วยการออกแบบสถาปัตยกรรมหรือ หน่วยการออกแบบแพ็คเกจอื่นๆ สามารถเรียกข้อมูลเหล่านี้ไปใช้ได้ นอกจากนั้นสิ่งที่นิยมทำกันมากคือการนำรูปแบบ มาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล 74XX เป็นต้น) มาเก็บไว้ในรูปของแพ็คเกจ ที่ทุกคนสามารถเข้าถึงได้

ตามปกติแล้วแพ็คเกจจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็คเกจ (Package declaration) และ ส่วนของบอดี้แพ็คเกจ (Package body) เนื่องจาก แพ็คเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่ กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็คเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถ กระทำได้ด้วยชุดคำสั่ง

**USE PACKAGE DECLARATION**

ส่วนที่มีความสำคัญที่สุดของแพ็คเกจ (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็คเกจ เนื่องจากเป็นส่วนที่ใช้กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็คเกจ สำหรับนำไปใช้ภายนอกตัวของแพ็คเกจเอง ถ้ามีการประกาศสิ่งใดๆ ในส่วนของส่วนบอดีแพ็คเกจ แต่ไม่ถูกประกาศในส่วนการประกาศแพ็คเกจจะทำให้ค่าและพฤติกรรมไม่สามารถนำไปใช้งานในส่วนนอกได้ซึ่งเปรียบเทียบกับได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศ Entity คือ จุดเชื่อมต่อหรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้น โดยทั่วไปแล้วแพ็คเกจสามารถสร้างขึ้นได้โดยไม่จำเป็น ต้องมีส่วนบอดี และสามารถนำไปใช้งานจากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศ ชนิด (Type) หรือสัญชาตญาณ เช่นเดียวกับ ส่วนบอดีแพ็คเกจที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็คเกจ แต่แพ็คเกจนั้นจะไม่สามารถนำไปใช้งานจากรูปแบบอื่นได้

```
PACKAGE package_name IS
    Package_declarative_part
END package_name;
```

รูปที่ 3.11 โครงสร้างทั่วไปของส่วนการประกาศแพ็คเกจ

#### - PACKAGE BODY

โครงสร้างซึ่งประกอบด้วยลำดับคำสั่งที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลาย ซึ่งชื่อของโปรแกรมย่อยนั้นๆ ได้ถูกประกาศไปแล้วในส่วนของการประกาศแพ็คเกจ จะถูกเก็บไว้ในส่วนของบอดีแพ็คเกจ ทั้งนี้รวมถึง การกำหนดค่าคงที่ต่างๆ อันได้แก่ค่าคงที่ที่ถูกประกาศชื่อไว้ก่อนในส่วนของการประกาศแพ็คเกจ และถูกกำหนดค่าใน ส่วนของบอดีแพ็คเกจ ฉะนั้นในส่วนของบอดีแพ็คเกจจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็คเกจไม่มีการประกาศชื่อที่เป็น โปรแกรมย่อย หรือค่าคงที่ การเขียนบอดีแพ็คเกจนั้นจะเป็นไปตามกฎเกณฑ์ดังแสดงในรูปที่ 3.12

```
PACKAGE BODY package_name IS
    declarative part
END package_name;
```

รูปที่ 3.12 โครงสร้างของบอดีแพ็คเกจ

#### 4.) หน่วยการออกแบบ Configuration

ดังที่ทราบกันแล้วว่าระบบดิจิทัลรูปแบบหนึ่งไม่ว่าจะเป็นอะไรก็ตาม จะสามารถมีหน่วยการออกแบบ Entity ได้ เพียงหนึ่งเดียวเท่านั้น ซึ่งในหน่วยการออกแบบ Entity หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลาย หน่วย ดังนั้นจะต้องมีหน่วยการออกแบบ Configuration มาเพื่อกำหนดการใช้ Configuration ของการประกอบ Entity กับหน่วยการออกแบบสถาปัตยกรรมหน่วยใดๆ เข้าด้วยกัน

```
CONFIGURATION identifier OF entity_name IS
    Configuration_declarative_part
END ;
```

รูปที่ 3.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงสร้าง

#### 5.) โปรแกรมย่อย

การใช้ฟังก์ชันและโพรซีเจอร์ใน VHDL เปรียบได้กับการใช้โปรแกรมย่อยในการเขียนโปรแกรมภาษาขั้นสูงต่างๆ ไปค่าที่ถูกส่งกลับหรือถูกเปลี่ยนแปลงโดยโปรแกรมย่อยอาจจะมีหรือไม่มีผลต่อฮาร์ดแวร์โดยตรงก็ได้ เช่นถ้าใช้ฟังก์ชัน แทนการกระทำในสมการบูลีนก็จะมีผลต่อวงจรลอจิกจริงๆ ในขณะที่ถ้าใช้โปรแกรมย่อยในการเปลี่ยนชนิดของข้อมูล หรือในการคำนวณค่าการหน่วงเวลาแล้วก็จะไม่มีผลต่อโครงสร้างของฮาร์ดแวร์ รูปที่ 3.14 แสดงการใช้โพรซีเจอร์ เพื่อเปลี่ยนข้อมูลชนิด 8 บิตเป็นค่าจำนวนเต็ม และรูปที่ 3.15 แสดงการใช้ฟังก์ชันโดยกำหนดให้ X เป็นตัวแปรชนิด บิตแทนการกระทำในสมการบูลีน

```
TYPE byte IS ARRAY (7 DOWNTO 0) OF BIT;
...
PROCEDURE byte_to_integer (ib : IN byte; oi : OUT INTEGER) IS
    VARIABLE result: INTEGER := 0;
BEGIN
    FOR i IN 0 TO 7 LOOP
        IF ib(i) = '1' THEN
            result := result + 2**i;
        END IF;
    END LOOP;
    oi := result;
END byte_to_integer
```

รูปที่ 3.14 การใช้โพรซีเจอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FUNCTION f (a, b, c:BIT) RETURN BIT IS
    VARIABLE x:BIT;
BEGIN
    x := ((NOT a) AND (NOT b) AND c);
    RETURN x;
END f;

```

รูปที่ 3.15 การใช้ฟังก์ชัน

## 6.) โอเพอเรเตอร์

การบรรยายเชิงพฤติกรรมในภาษา VHDL มีตัวดำเนินการหรือโอเพอเรเตอร์ทางลอจิกและคณิตศาสตร์เช่นเดียวกับภาษาซอฟต์แวร์ทั่วไปดังรูปที่ 3.16

PREDEFIND OPERATORS	
LOGICAL OPERATORS :	NOT AND OR NAND NORXOR
OPERAND TYPE :	BIT BOOLEAN
RESULT TYPE :	BIT BOOLEAN
RELATIONAL OPERATORS :	= /= < <= > >=
OPERAND TYPE :	any type
RESULT TYPE :	Boolean
ARITHMETIC OPERATORS :	+ - * / ** MOD REM ABS
OPERAND TYPE :	INTEGER REAL Physical
RESULT TYPE :	INTEGER REAL Physical
CONCANTENATION OPERATOR :	&
OPERAND TYPE :	ARRAY of any type
RESULT TYPE :	array of any type
RESULT TYPE :	array of any type

รูปที่ 3.16 ตัวดำเนินการใน VHDL

## 7.) เวลาและความพร้อมเพรียง

ในวงจรอิเล็กทรอนิกส์อุปกรณ์ทุกๆ ตัวจะอยู่ในสภาวะเตรียมพร้อมเสมอ (Always Active) และจะมีเรื่องของเวลาเข้ามาเกี่ยวข้องในหลายๆเหตุการณ์ที่เกิดขึ้นเสมอ

VHDL เป็นภาษาที่ได้รับการออกแบบมาเพื่อให้สามารถบรรยายรูปแบบและการพ้องกันของเวลาสำหรับการทำงานของอุปกรณ์ได้อย่างถูกต้อง การบรรยายการทำงานที่อยู่ภายในส่วน ของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบรรยายสถาปัตยกรรม จะมีการทำงานที่พร้อมเพรียงกันเสมอ หรือแม้แต่โปรเซสซึ่งมีการทำงานภายในเป็น แบบลำดับคำสั่งก็ตาม ซึ่งหากมีหลายๆ โปรเซสอยู่ภายในโครงสร้างเดียวกัน ทุกๆ โปรเซสก็จะทำงานไปพร้อมๆ กัน ด้วย

#### 8.) สัญญาณและตัวแปร

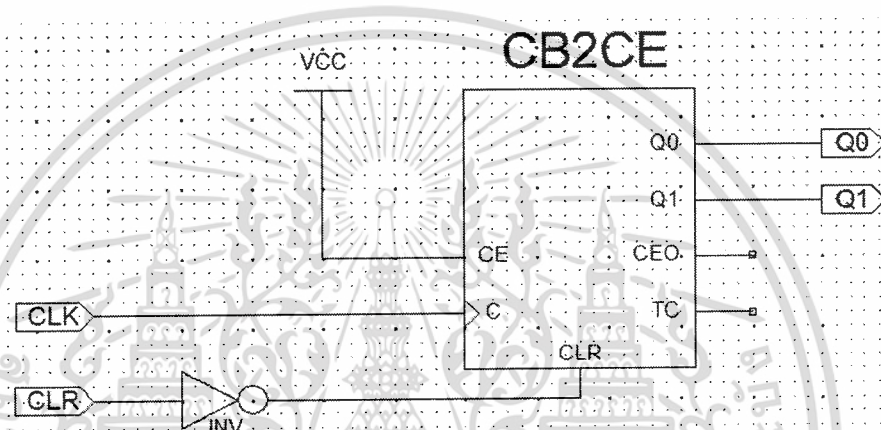
สัญญาณมีลักษณะเป็นเสมือนตัวกลางฮาร์ดแวร์ที่ใช้ในการส่งผ่านข้อมูลและมีเรื่องของเวลาเข้ามาเกี่ยวข้องด้วย การกำหนดค่าให้กับสัญญาณจะใช้สัญลักษณ์  $\leq$  ในการส่งค่าและสามารถใช้คำสั่ง AFTER เพื่อกำหนดช่วงเวลาในการ ส่งผ่านค่าของสัญญาณ เช่น  $w \leq a$  AFTER 12 NS หมายถึงการกำหนดค่าสัญญาณ  $a$  ให้กับ  $w$  หลังจากเวลา ผ่านไป 12 นาโนวินาที

ในทางตรงข้ามตัวแปรมีลักษณะเป็นเสมือนตัวกลางที่ใช้ในการส่งผ่านข้อมูลและไม่มีเรื่องของ เวลาเข้ามาเกี่ยวข้องด้วย ซึ่งตัวแปรจะถูกใช้ในส่วนที่มีการทำงานเป็นแบบลำดับคำสั่งเช่นใน ฟังก์ชัน โปรซีเจอร์ และ โปรเซส สำหรับการกำหนดค่าให้กับตัวแปรจะใช้สัญลักษณ์: =


## บทที่ 4

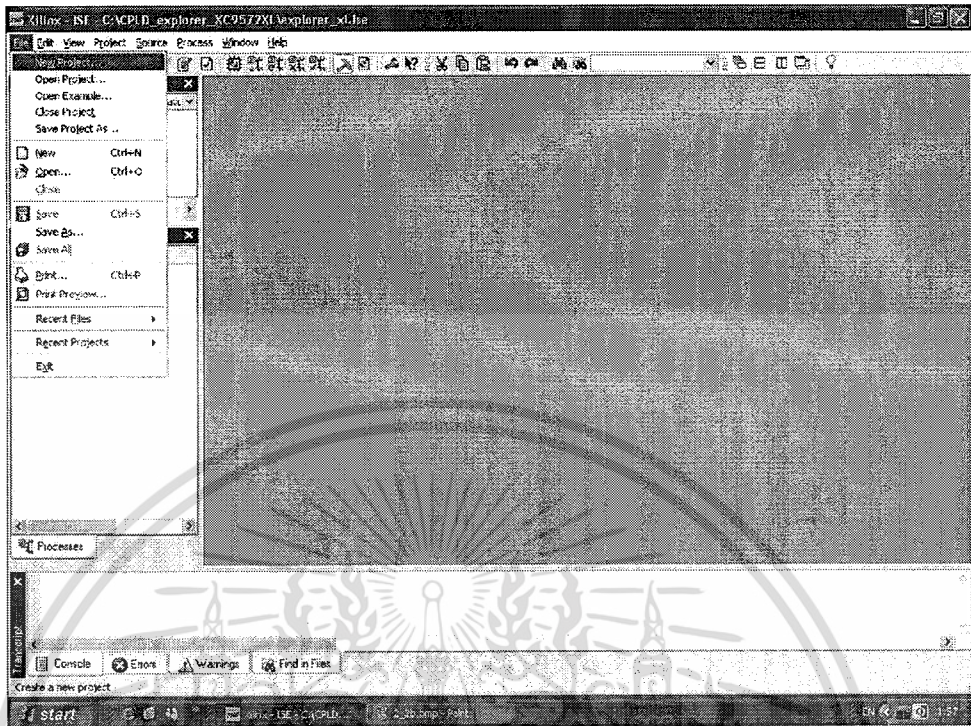
### การใช้โปรแกรม Xilinx-ISE เบื้องต้น

#### 4.1 ขั้นตอนการออกแบบวงจรนับ 4 แบบซิงโครนัสด้วยวิธีวาดผังวงจร

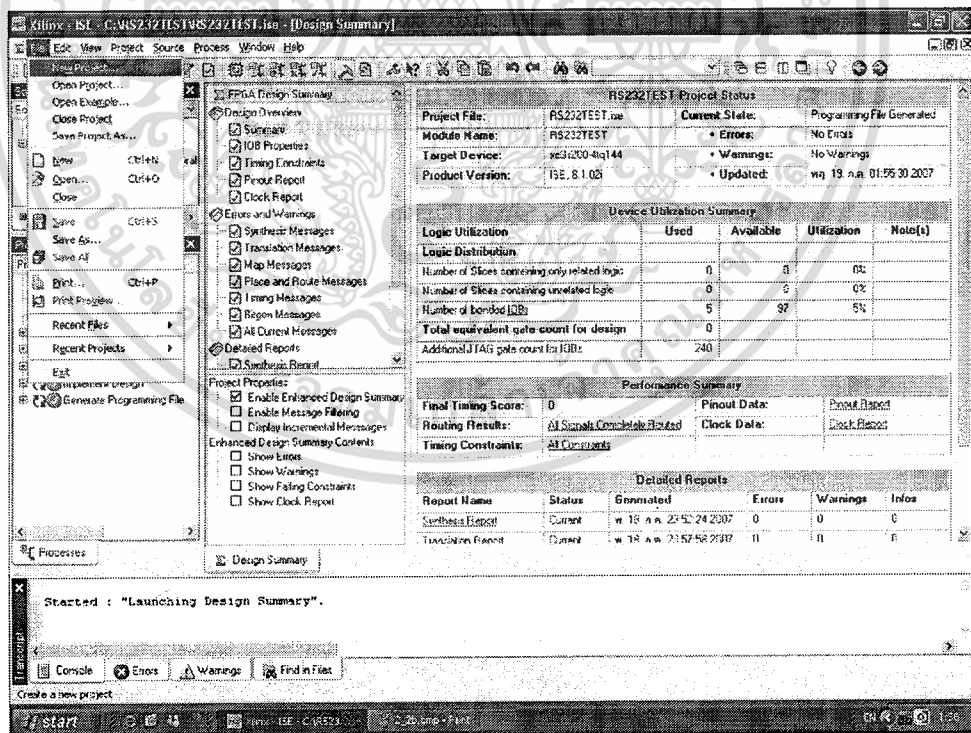


รูปที่ 4.1 วงจรนับ 4 แบบซิงโครนัส

4.1.1) ก่อนจะใช้โปรแกรม ISE Web pack ให้สร้าง Folder ใหม่ชื่อ ch2sch (หรือชื่ออื่นก็ได้) ไว้ในไดรฟ์ C เสร็จแล้วจึงเริ่มใช้งาน ISE Web pack โดยที่หน้าจอกอมพิวเตอร์ คลิกปุ่ม Start->Program->Xilinx 9.2i->Project Navigator หรือดับเบิลคลิกที่  แล้วจะได้หน้าต่าง Xilinx-ISE จากนั้นคลิก File-> New Project แล้วจะได้หน้าต่าง ดังรูป 4.1 (หรือ Dialog Box) New Project Wizard



a)

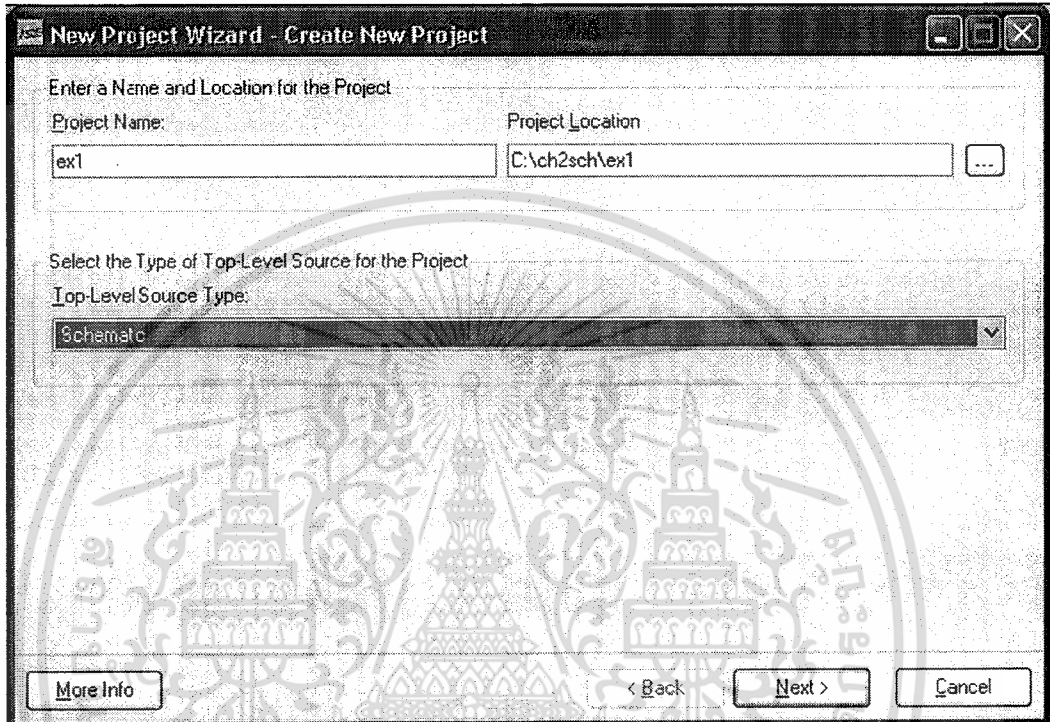


b)

### รูปที่ 4.2 หน้าต่าง Xilinx-ISE

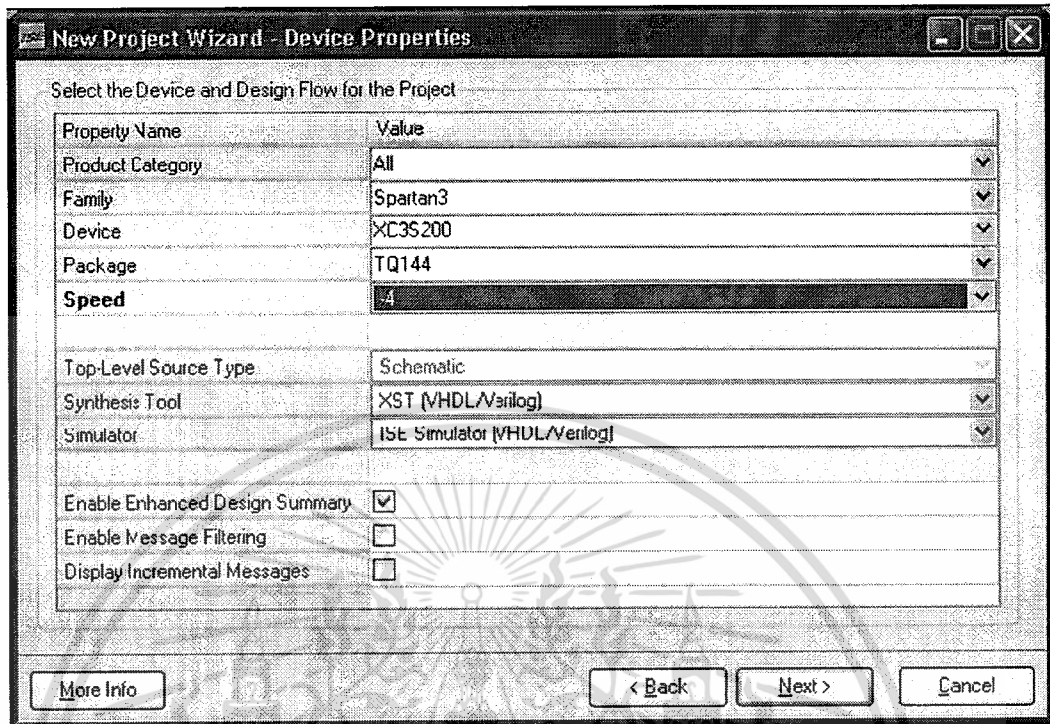
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2) ที่หน้าต่าง New Project Wizard-Create New Project ให้พิมพ์ชื่อ C:\ch2sch ไว้ในช่อง Project Location แล้วพิมพ์ชื่อ ex1 ที่ช่อง Project Name จากนั้นคลิกที่ Top-Level Source Type เป็น Schematic แสดงดังรูปที่4.2.คลิก Next แล้วจะได้หน้าต่างถัดไป

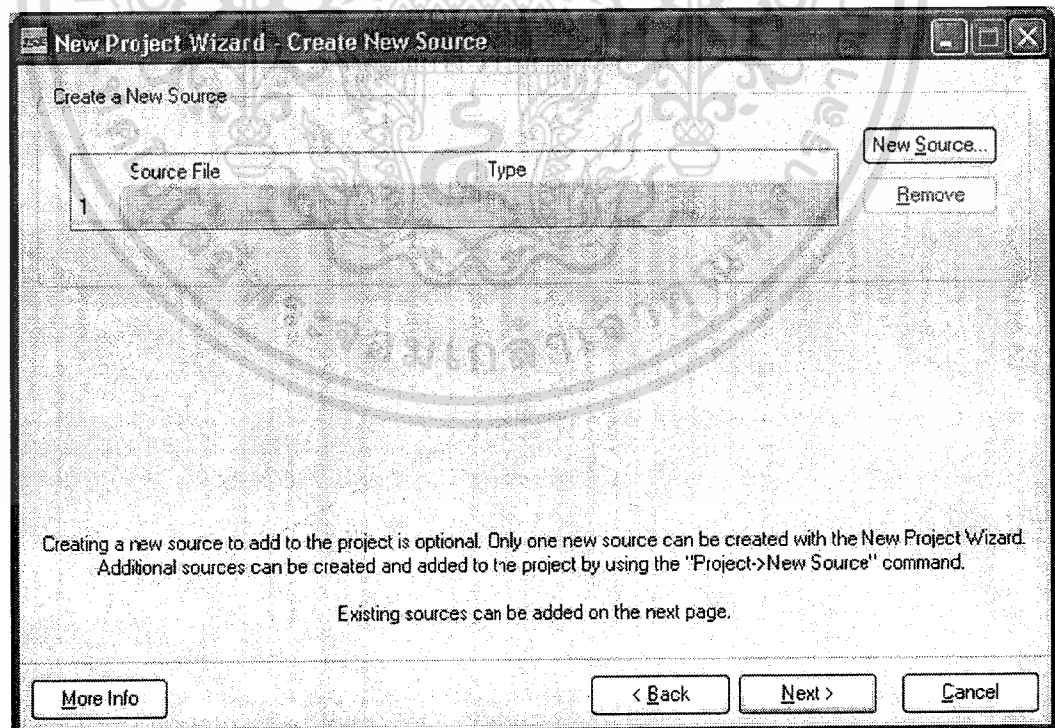


รูปที่ 4.3 หน้าต่าง New Project Wizard-Create New Project

4.1.3) ที่หน้าต่าง New Project Wizard-Device Properties ให้คลิกดังรูปที่ 4.4 ซึ่งในการออกแบบนี้จะใช้ FPGA เบอร์ XC3S200-4TQ144C ที่ใช้บนบอร์ดFPGA Discovery-III XCS200F ดังนั้นต้องเลือกตระกูล (Family) Spartan3 เบอร์ (Device) XC3S200 ที่มีPackage แบบ ๕TQ144 ขา (Package: TQ144) และ Speed Grade: -4 จากนั้นคลิกเลือกที่ Synthesis Tool เป็น XST (VHDL/Verilog) และ Simulator เป็น ISE Simulator (VHDL/Verilog) คลิก Next และจะได้หน้าต่างดังรูปที่ 4.3



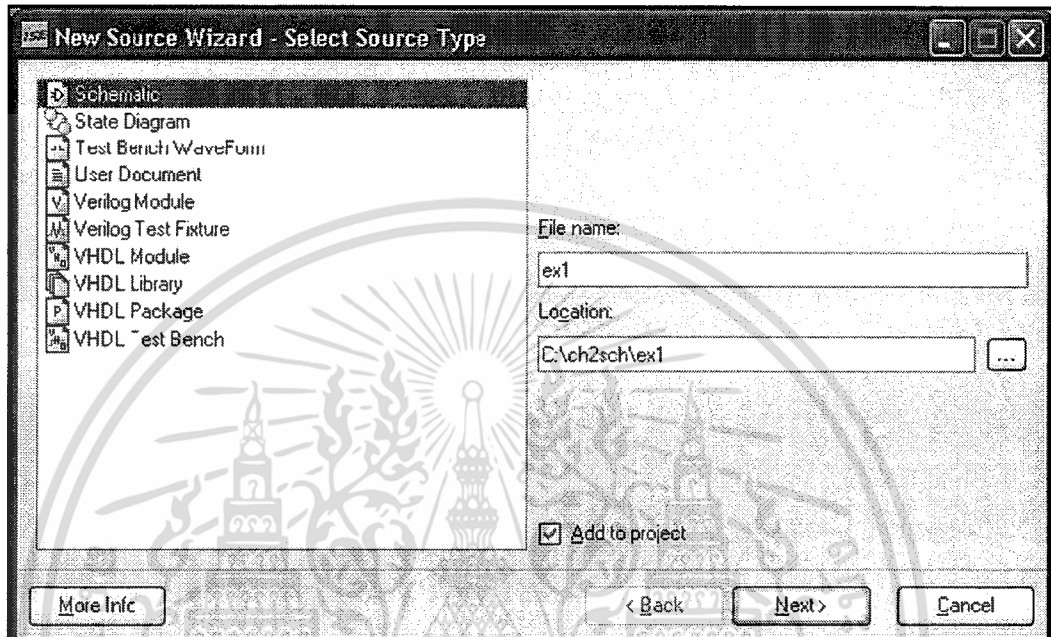
รูปที่ 4.4 หน้าต่าง New Project Wizard-Device Properties



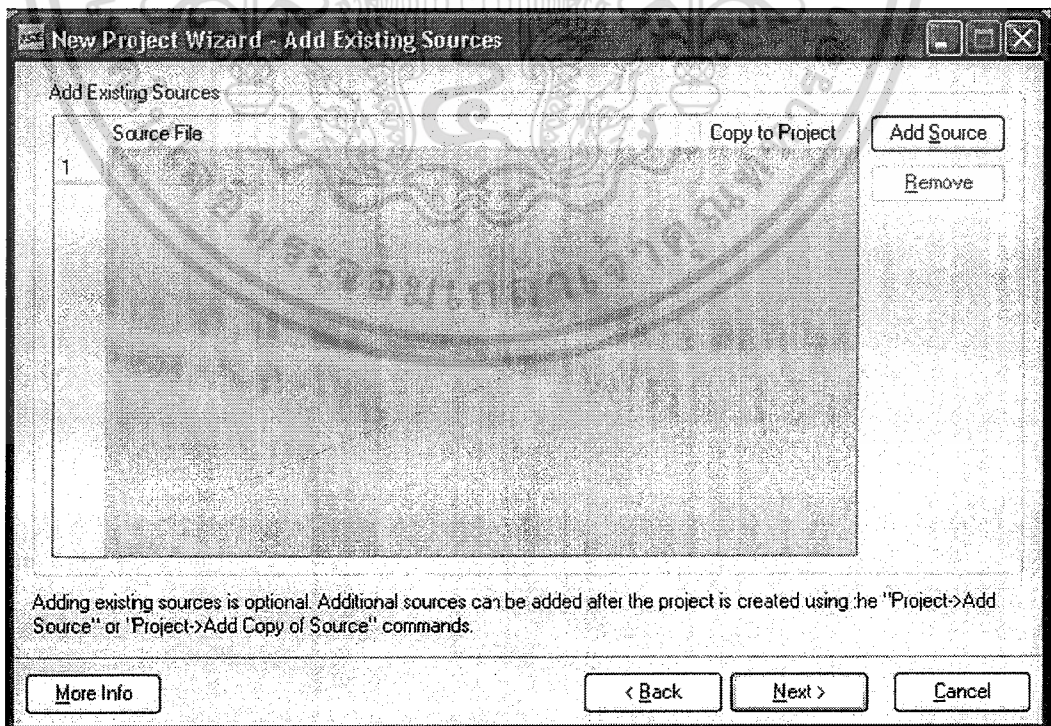
รูปที่ 4.5 หน้าต่าง New Project Wizard-Create New Source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4) จากรูป 4.5 คลิกปุ่ม New Source แล้วจะได้หน้าต่างถัดไป พิมพ์ชื่อ ex1 ลงในช่อง File Name แล้วคลิกที่ Schematic ดังรูปที่ 4.6 คลิก Next อีกครั้งจะได้หน้าต่างดังรูปที่ 4.7 .เสร็จแล้วคลิก Next และคลิก Finish แล้วจะได้หน้าต่าง Xilinx-ISE ที่ใช้สำหรับวาดผังวงจรดังรูปที่ 4.8

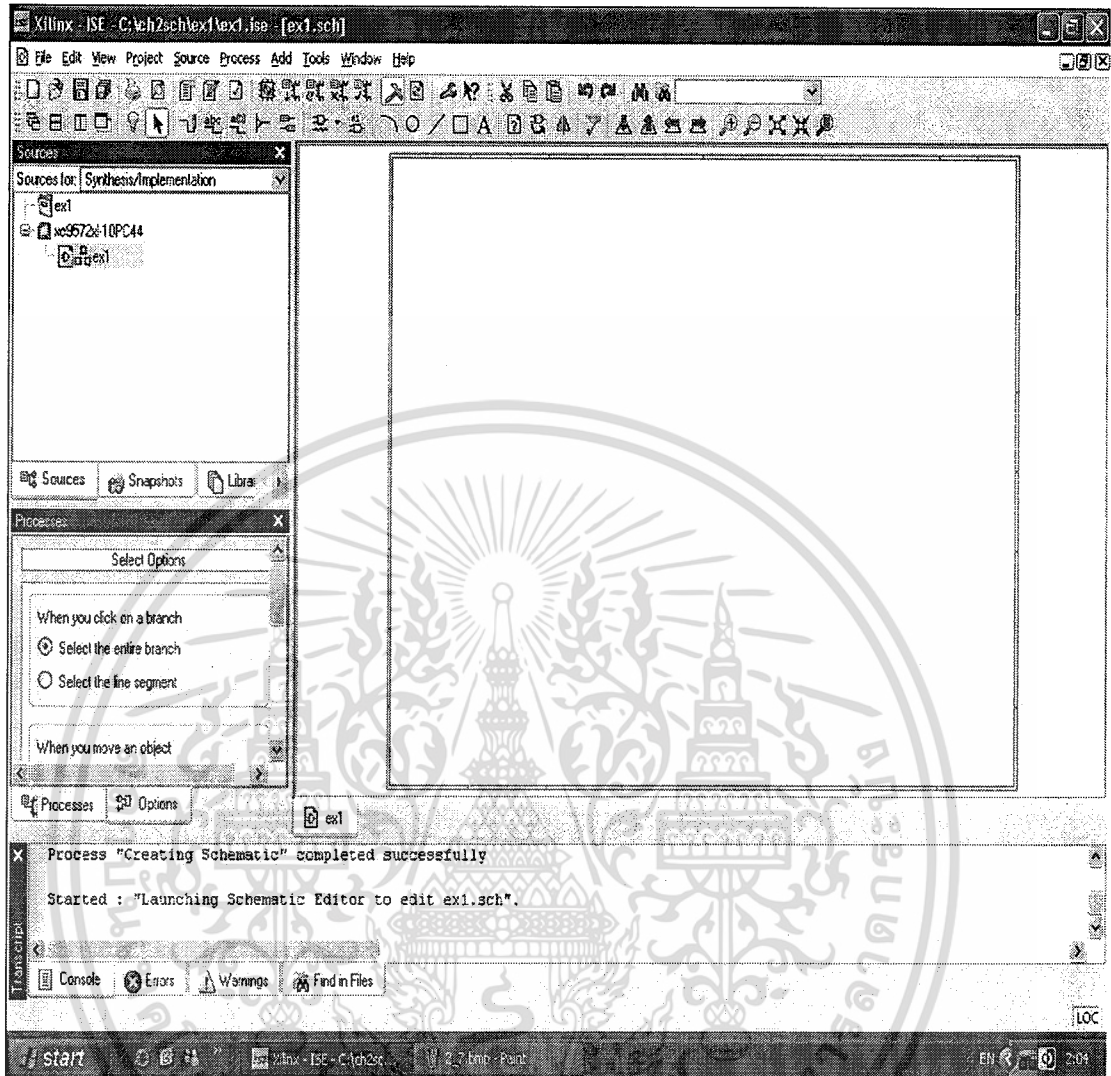


รูปที่ 4.6 หน้าต่าง New Project Wizard-Select Source Type



รูปที่ 4.7 หน้าต่าง New Project Wizard-Add Existing Source

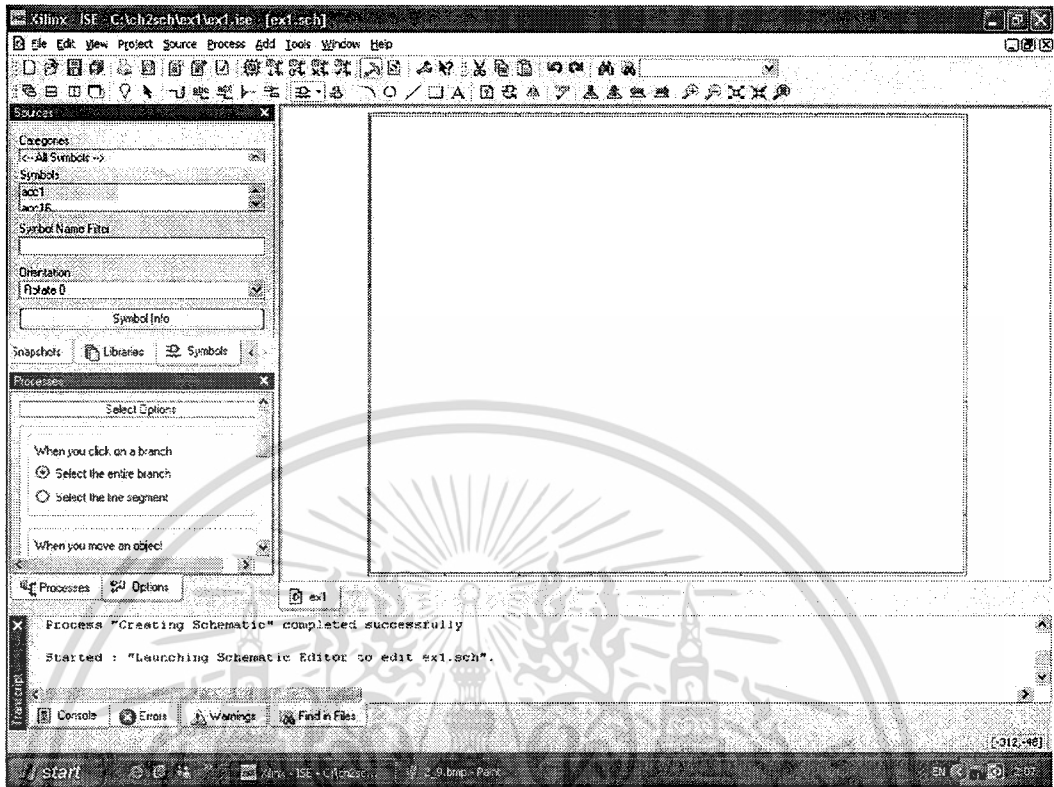
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



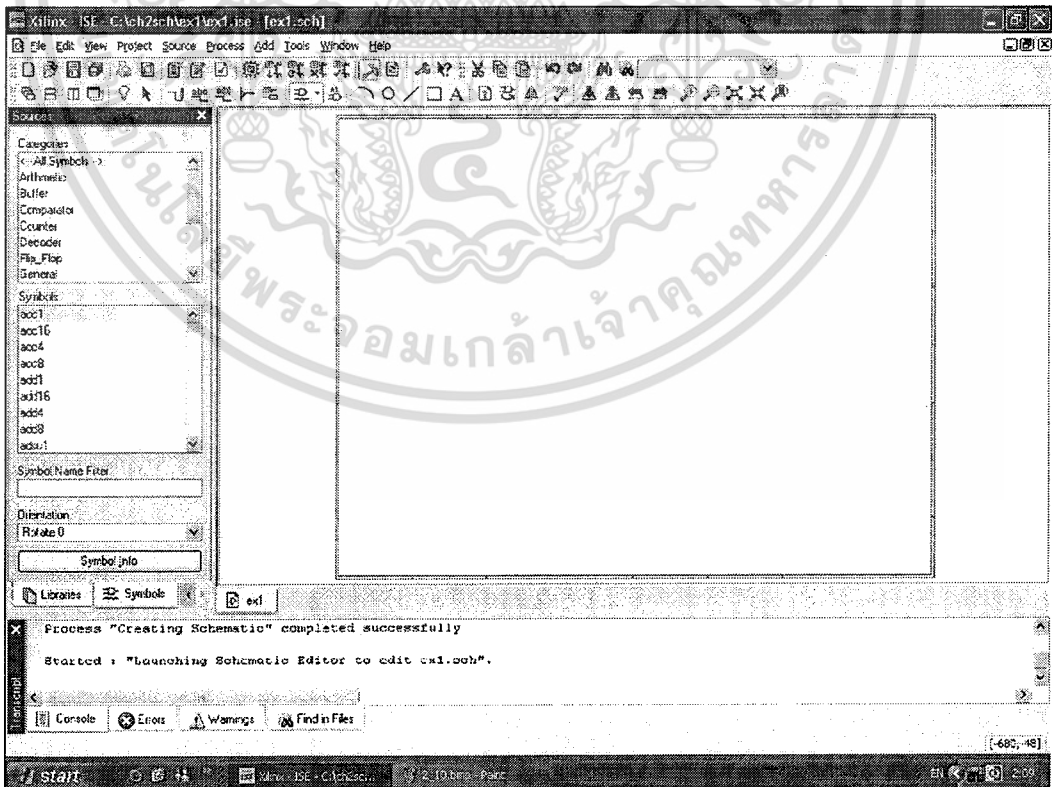
รูปที่ 4.8 หน้าต่าง Xilinx-ISE

4.1.5) การวาดผังวงจรคลิก  จากนั้นคลิก  ด้านบนขวาของหน้าต่าง Processes ในรูปที่ 4.9 แล้วจะได้ดังรูปที่ 4.10 ซึ่งจะปรากฏรายการของหมวดอุปกรณ์ใน Categories และชนิดอุปกรณ์ใน Symbols อย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้






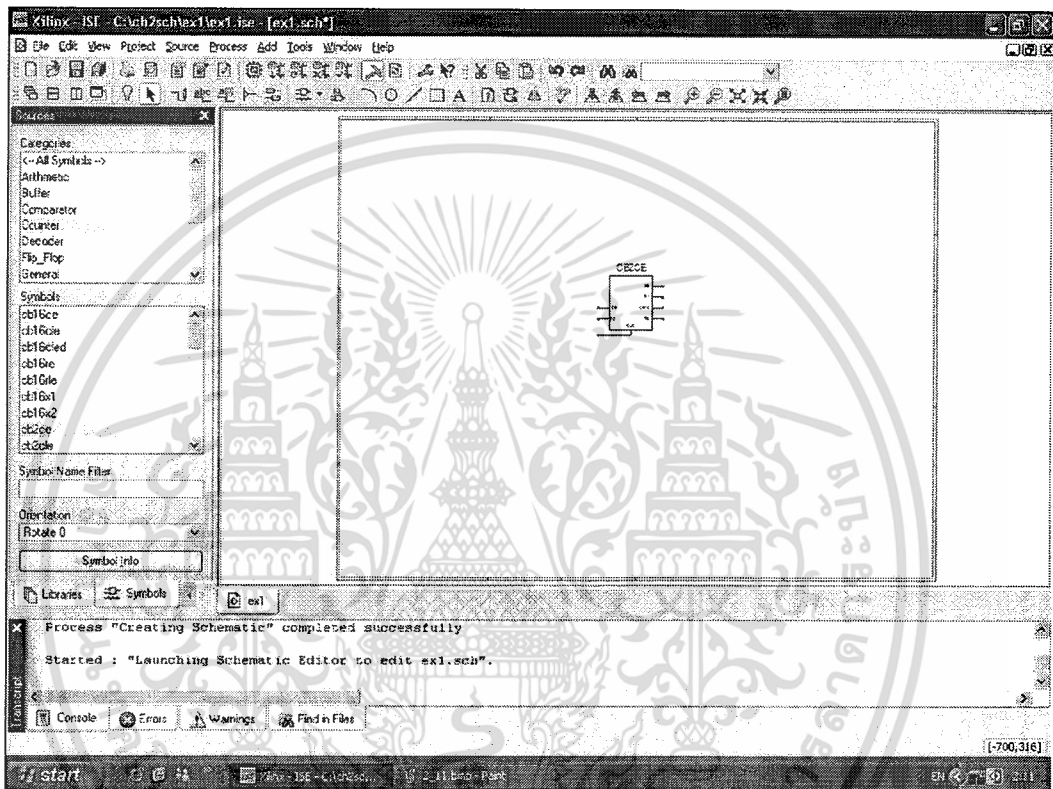
รูปที่ 4.9 รายการอุปกรณ์



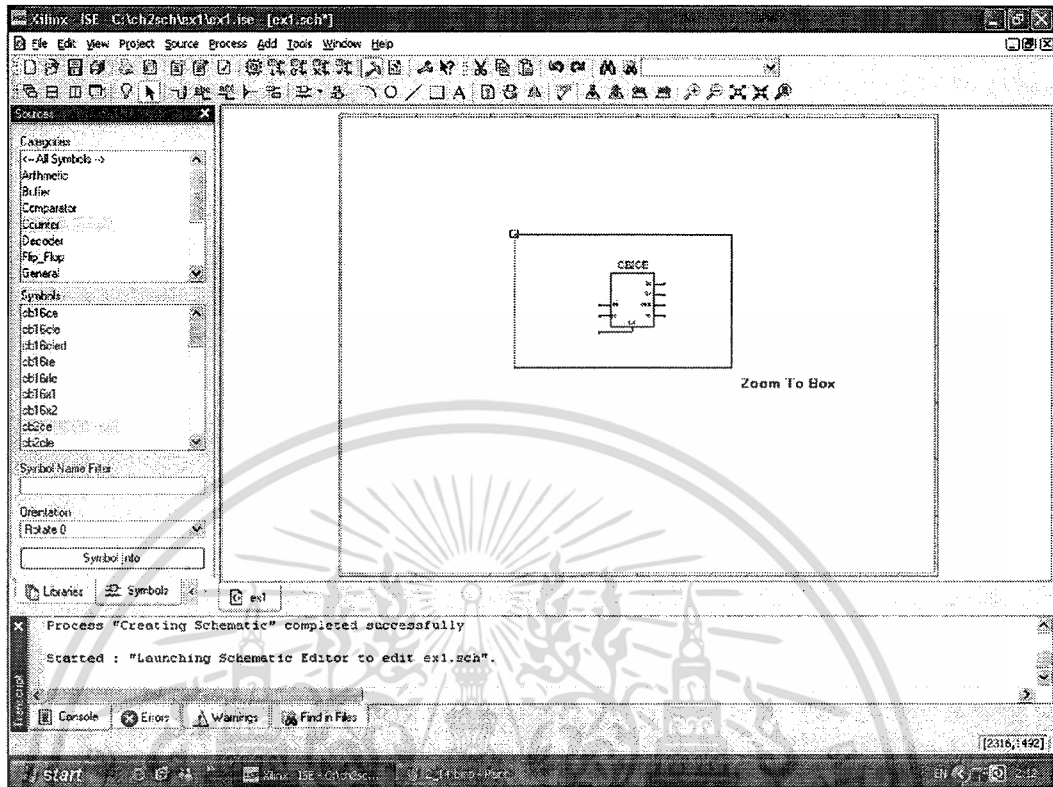
รูปที่ 4.10 รายการอุปกรณ์ที่เห็นรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

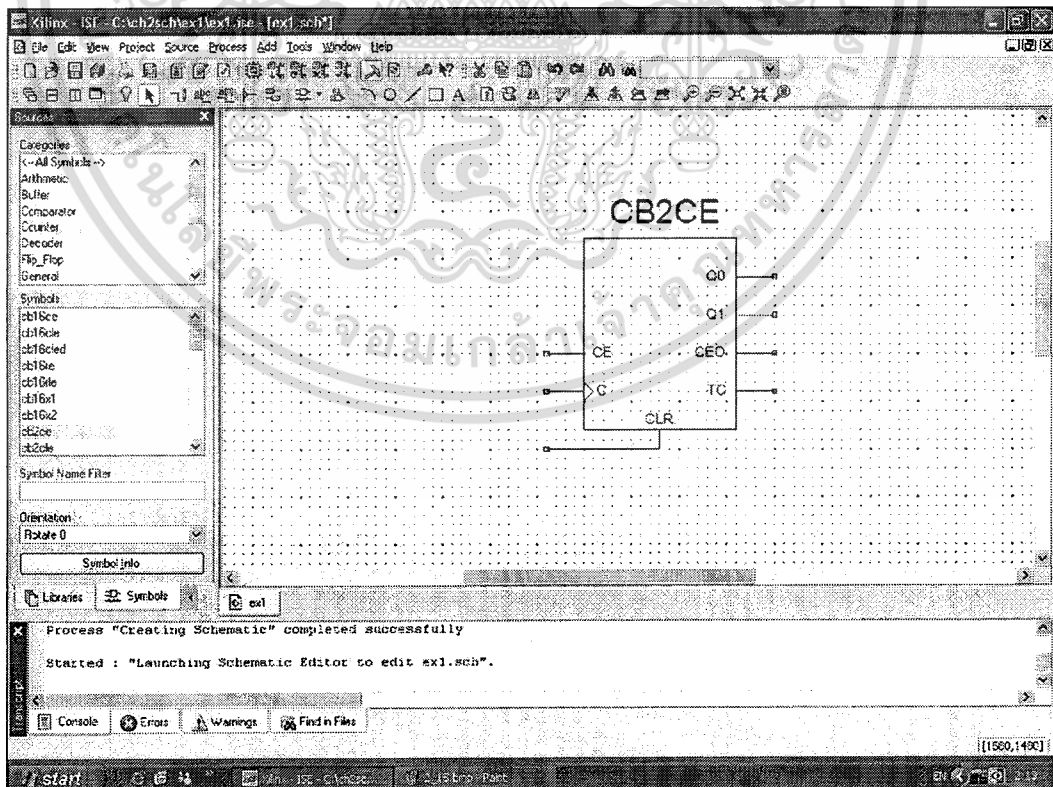
4.1.6) การวาดผังวงจร คลิกที่ไอคอน  จากนั้นคลิกที่ Counter และ cb2ce ในหน้าต่าง Categories และ Symbols ตามลำดับ เลื่อนเมาส์มา ณ ตำแหน่งที่ต้องการแล้วคลิกเมาส์วาง cb2ce ดังรูปที่ 4.11 คลิก  เพื่อหยุดการขยายรูปให้คลิก  แล้วเลื่อนเมาส์มาที่พื้นที่ที่ต้องการ คลิกซ้ายค้างไว้พร้อมลากทแยงจากซ้ายมาขวาดังรูปที่ 4.12 เมื่อปล่อยเมาส์จะได้รูปที่ขยายขึ้นดังรูปที่ 4.13



รูปที่ 4.11 การคลิกเลือกหมวดและชนิดอุปกรณ์ที่อยู่ใน Categories และ Symbols ตามลำดับ






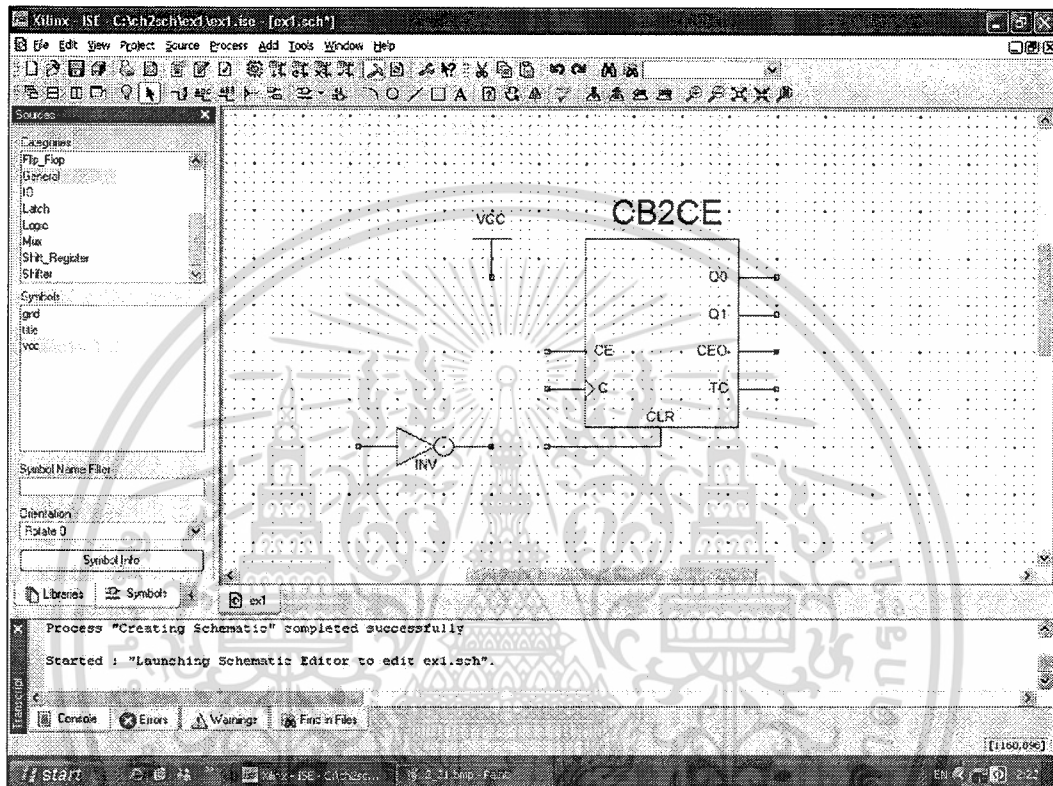
รูปที่ 4.12 แสดงขั้นตอนการ Zoom ขยาย



รูปที่ 4.13 รูปที่ Zoom ขยายแล้ว

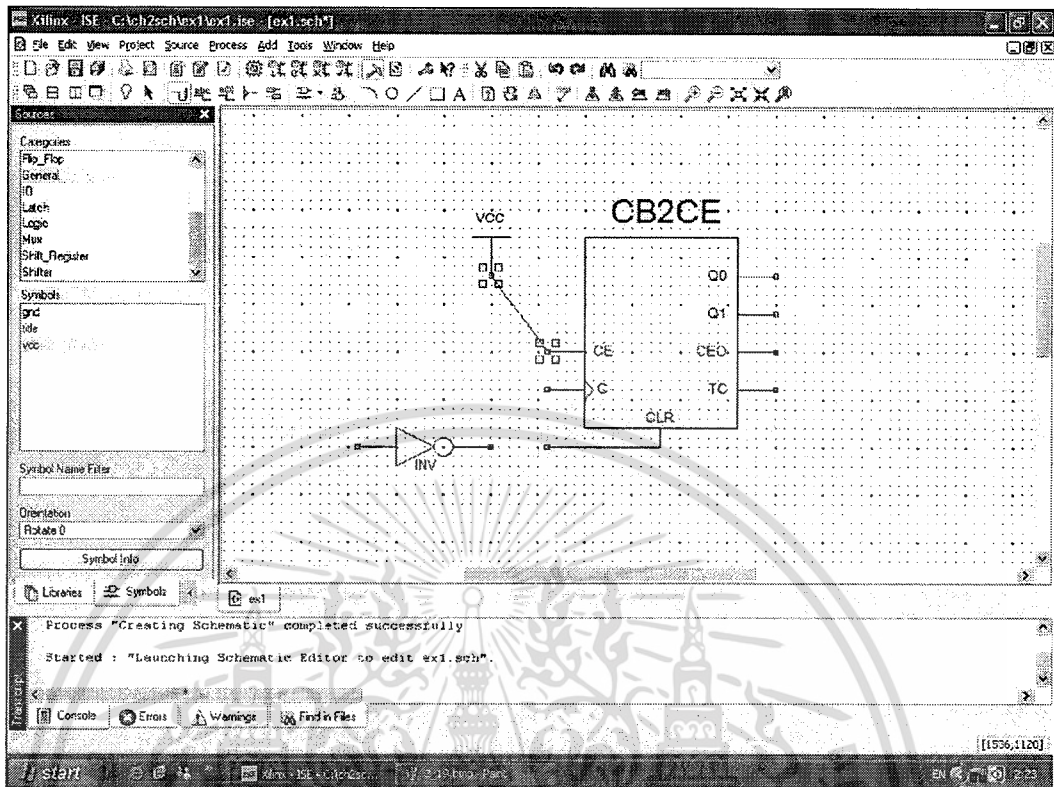
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.7) คลิกที่ไอคอน  แล้วคลิกที่ Logic และ inv ใน Categories และ Symbols ตามลำดับ เลื่อนเมาส์มา ณ ที่ตำแหน่งที่ต้องการคลิกเมาส์วาง inv แล้วคลิก  จากนั้นวาง Vcc โดยคลิกเลือกที่ General และ Vcc แล้วนำมาวางดังรูป 4.14 คลิก 

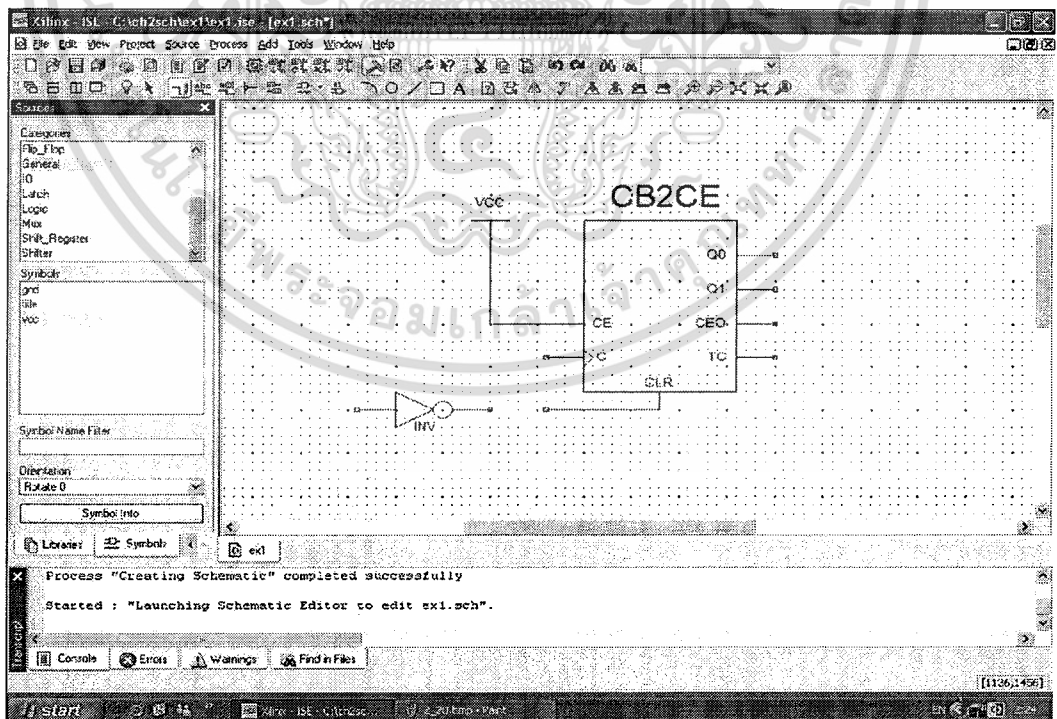


รูปที่ 4.14 วางอุปกรณ์เกตแบบอินเวอร์เตอร์และ Vcc

4.1.8) การลากเส้น คลิก  แล้วเลื่อนเมาส์ไปที่จุดปลาย Symbol คลิกเมาส์ค้างไว้แล้วลากไปที่จุดปลายอีกด้านหนึ่งดังรูปที่ 4.15 เมื่อหยุดคลิกเมาส์แล้วเส้นจะถูกวาดเองอัตโนมัติดังรูปที่ 4.16 จากนั้นลากสายสัญญาณเส้นต่อไปจนครบทุกเส้นดังรูปที่ 4.17

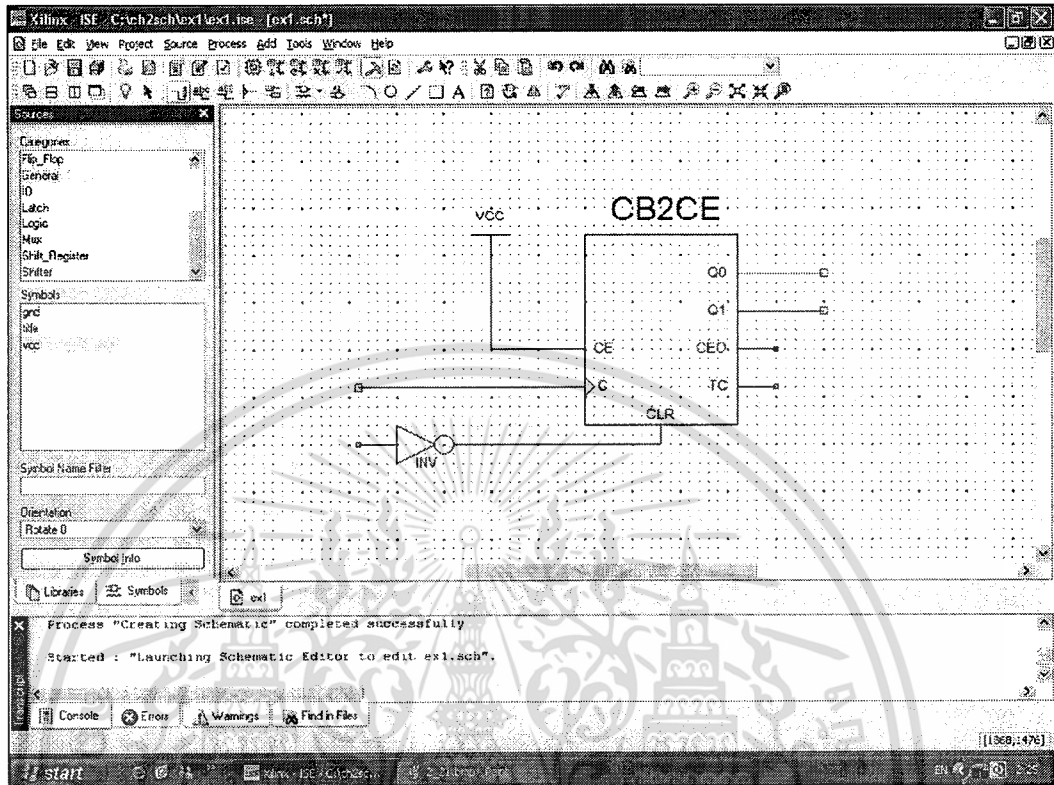


รูปที่ 4.15 แสดงการลากเส้นระหว่างจุด 2 จุด





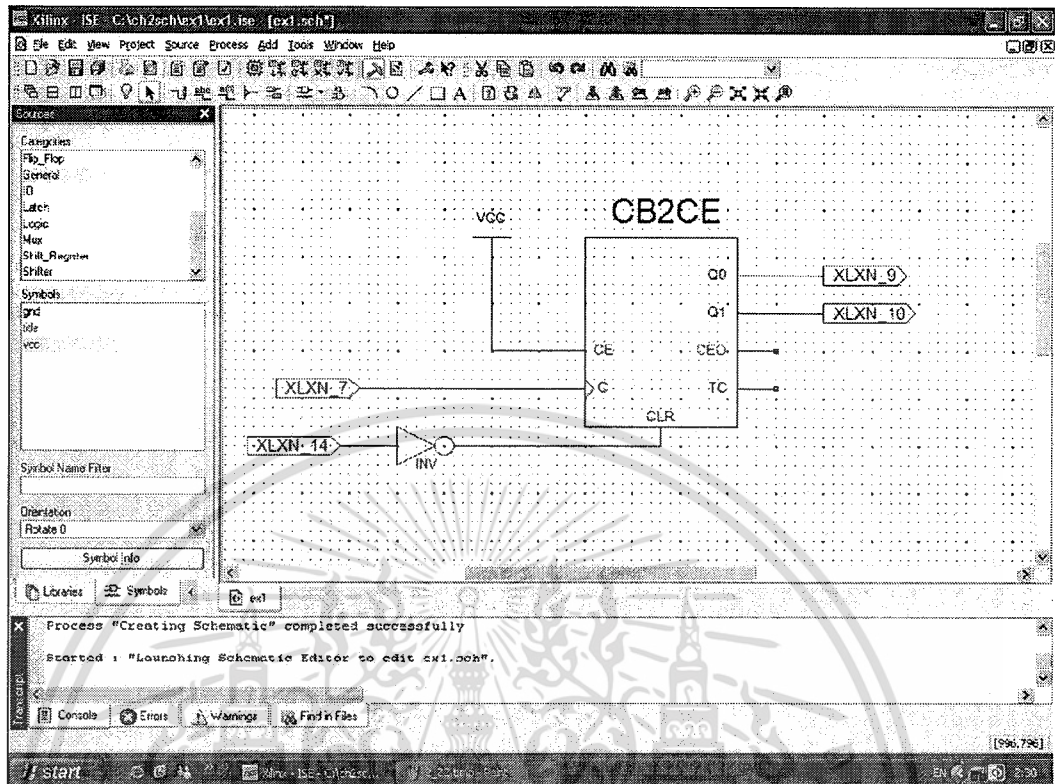
รูปที่ 4.16 สายสัญญาณที่ถูกวาดแนวเส้นอย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



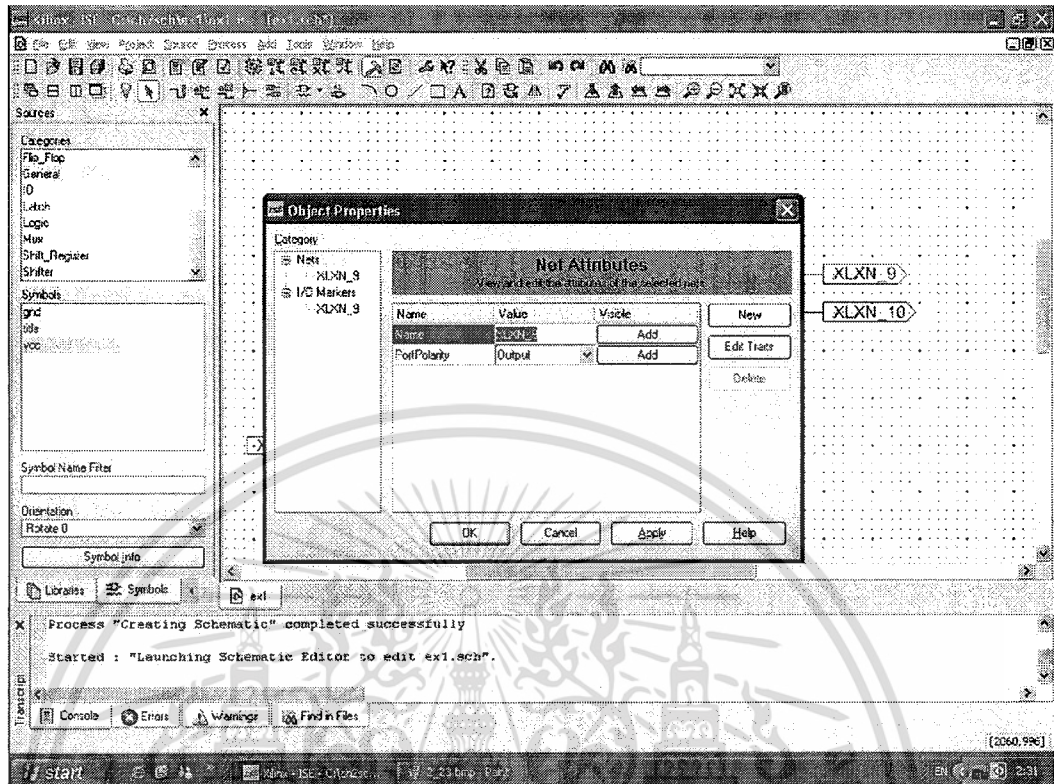
รูปที่ 4.17 แสดงต่อสายสัญญาณต่างๆจนครบตามที่ยอกแบบไว้

4.1.8) การใส่ I/O Marker เพื่อกำหนดจุดปลายสายสัญญาณ คลิก  จากนั้นเลื่อนเมาส์ไปที่ขั้วอินพุตหรือเอาต์พุต แล้วคลิกเมาส์วาง I/O Marker ตามที่ต้องการ เมื่อครบแล้วให้หยุดโดยคลิก  จะได้ดังรูปที่ 4.18 (กรณีนี้ CEO และ TC ไม่ใช้จึงไม่ต้องการ I/O Marker แต่ถ้าเป็นขาอินพุตห้ามปล่อยขาลอยไว้เด็ดขาด ซึ่งอาจต่อขาอินพุตเข้ากับ Vcc หรือ Gnd )

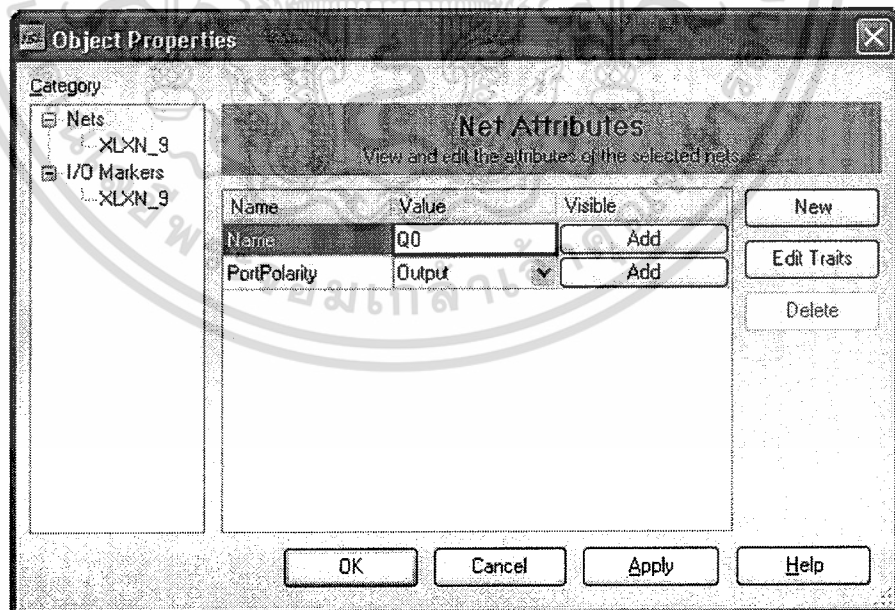


รูปที่ 4.18 การใส่ I/O Marker เพื่อกำหนดจุดปลายของสัญญาณ

4.1.9) จากรูปที่ 4.18 การแก้ไขชื่อของสายสัญญาณที่ I/O Marker ทำได้โดยดับเบิลคลิกที่ I/O Marker ที่ต้องการแล้วจะ ได้ดังรูปที่ 4.19 จากนั้นทำการแก้ไขที่คอลัมน์ Value แถว Name เป็นชื่อ Q0 ส่วนแถว Port Polarity นั้นเป็น Output ถูกต้องแล้ว ดังรูปที่ 4.20 คลิก OK จากนั้นแก้ไข I/O Marker จนเสร็จดังรูปที่ 4.21

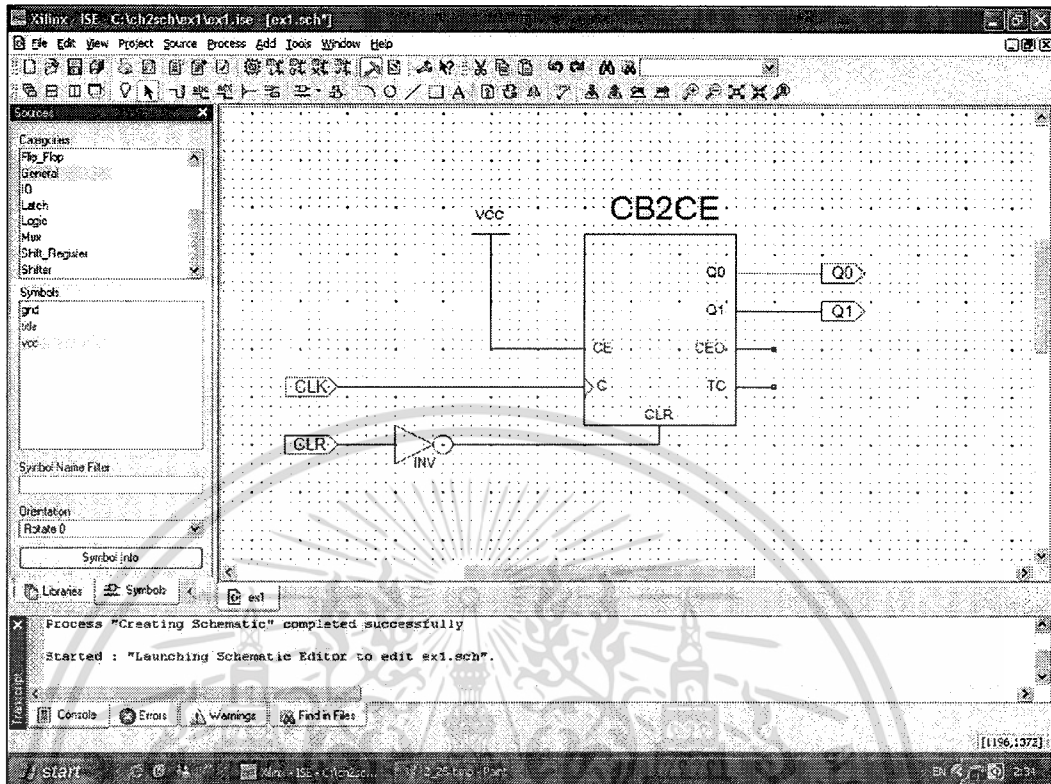


รูปที่ 4.19 ขั้นตอนการแก้ไขชื่อของ Wire หรือ Net ที่ I/O marker





รูปที่ 4.20 ขั้นตอนการแก้ไขชื่อของ Wire หรือ Net ที่ I/O marker

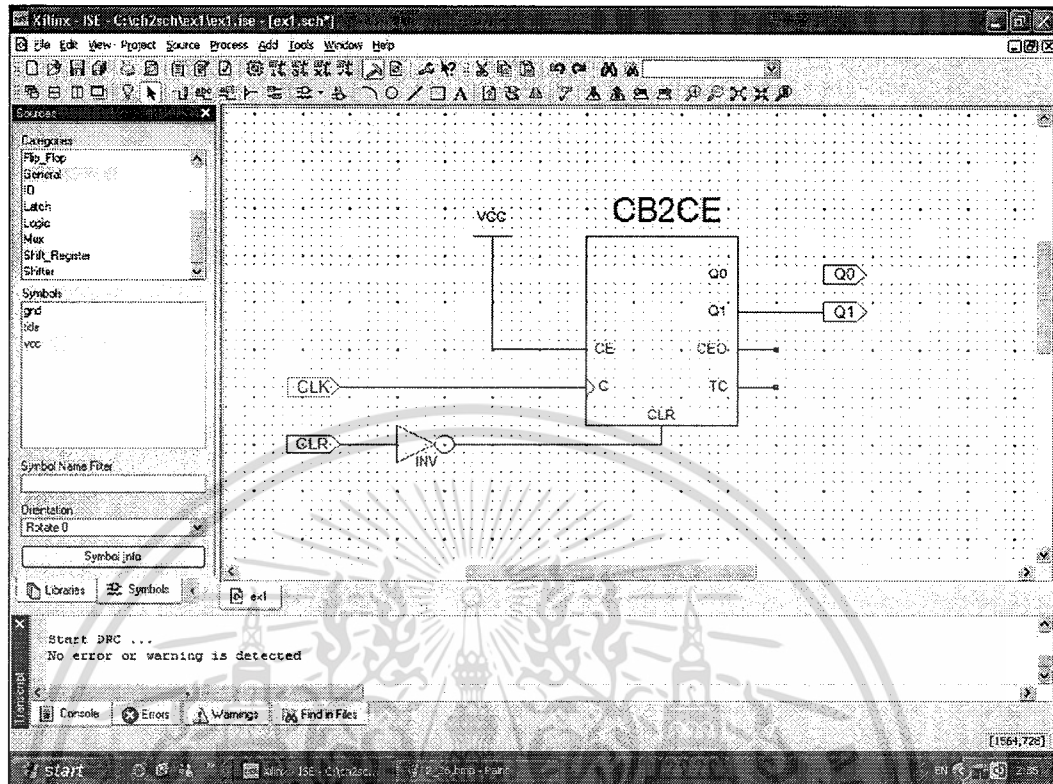
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 การแก้ไขชื่อของ Wire หรือ Net ที่ I/O Marker จนได้วงจรที่แล้วเสร็จสมบูรณ์

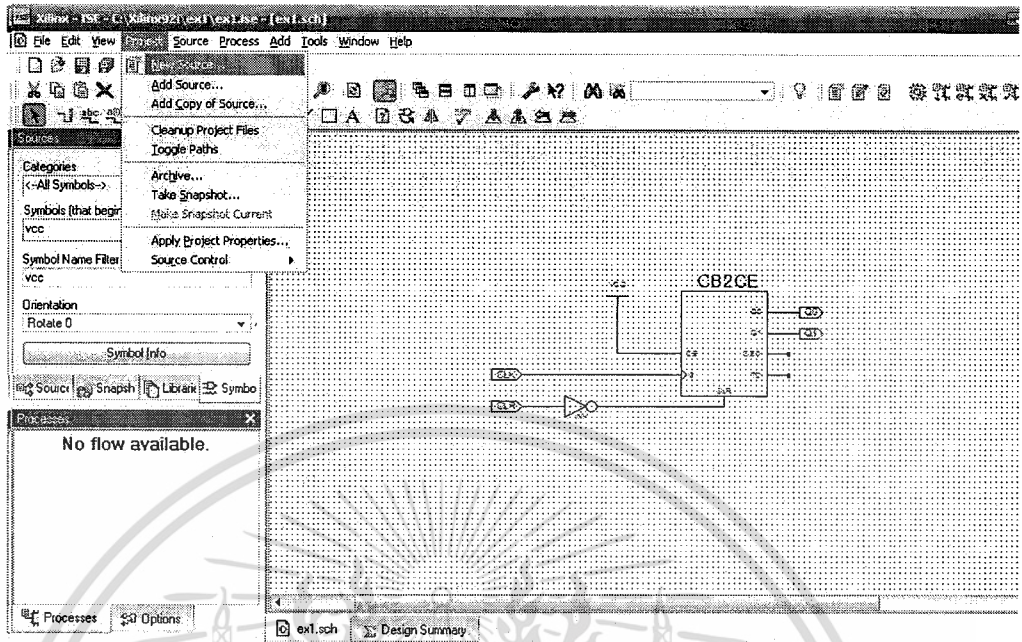
4.1.10) การตรวจสอบความถูกต้อง (Syntax) ทำได้โดยคลิก  ถ้าไม่มีข้อผิดพลาดจะปรากฏข้อความ No error or warning is detected ดังรูปที่ 4.22 จากนั้นบันทึกไฟล์โดยคลิก  ในขั้นตอนนี้ถือว่าการออกแบบวงจรเสร็จสมบูรณ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

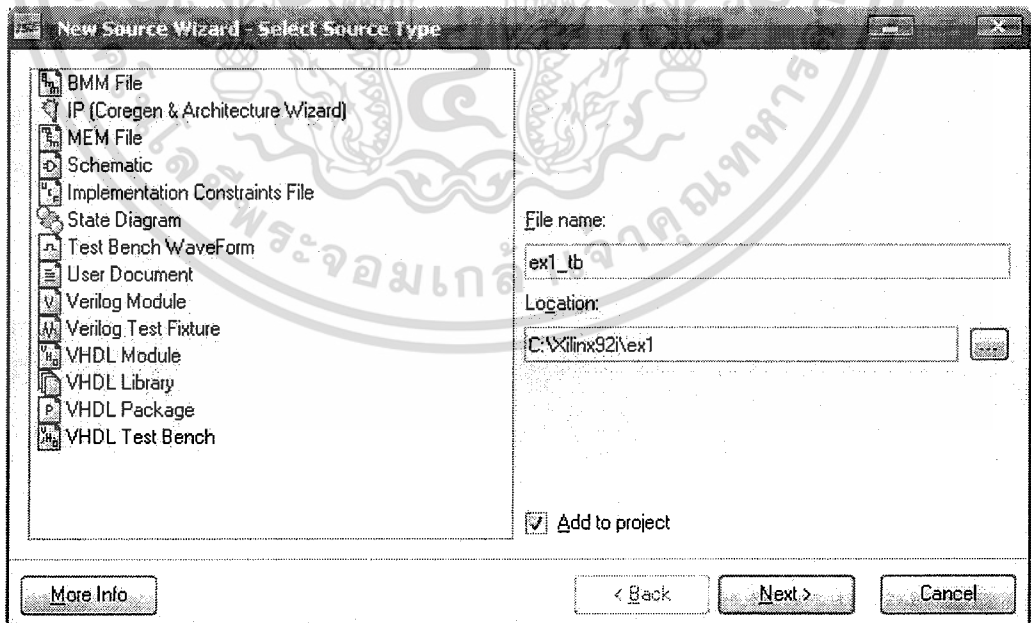


รูปที่ 4.22 การตรวจสอบความถูกต้อง

4.1.11) เราจะทำการสร้างไฟล์ทดสอบที่เรียกว่า Testbench เพื่อใช้จำลองการทำงานของ Schematic ที่เราออกแบบจากภาษา VHDL ในระดับฟังก์ชัน (Functional Simulation) จากโปรแกรมช่วยในการสร้างไฟล์ทดสอบ (Test Bench Waveform) โดยเริ่มจาก คลิก Project ที่เมนูบาร์ เลือก New Source > Test Bench Waveform โดยกำหนด File Name: ex1\_tb Location: C:\ch2\ex1 แล้วคลิก Next ดังรูปที่ 4.24 โปรแกรมจะให้เราทำการเลือก Source ในการสร้างไฟล์ทดสอบ ซึ่งในที่นี้คือ ex1 จากนั้นคลิก Next โปรแกรมจะทำการสร้าง Test Bench Waveform สำหรับวงจร ex1 ที่เราออกแบบขึ้นดังรูปที่ 4.26 จากนั้นคลิก Finish

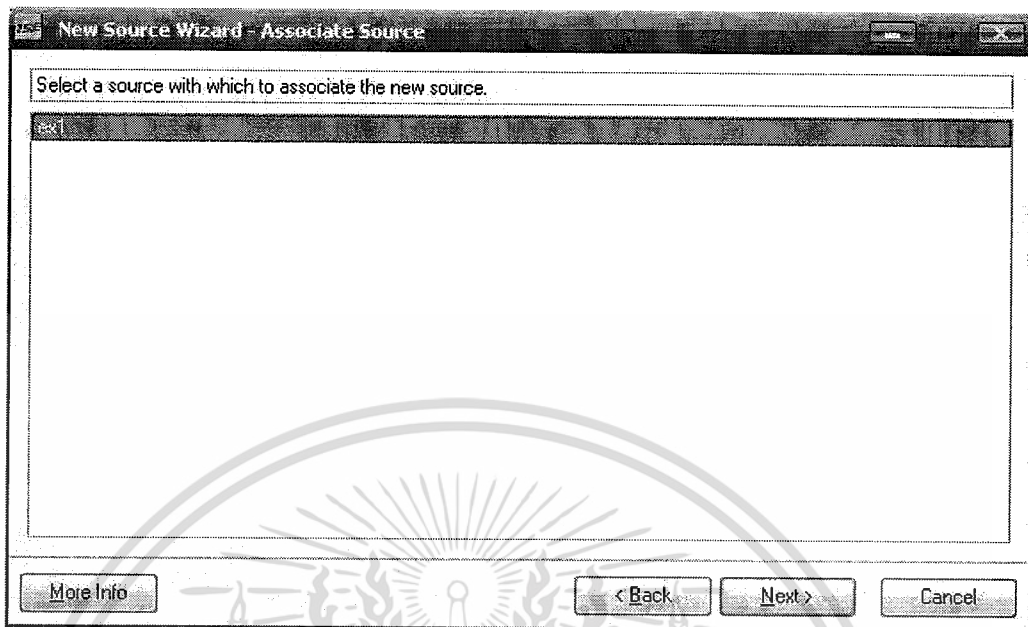


รูปที่ 4.23 การสร้างไฟล์ Test Bench Waveform

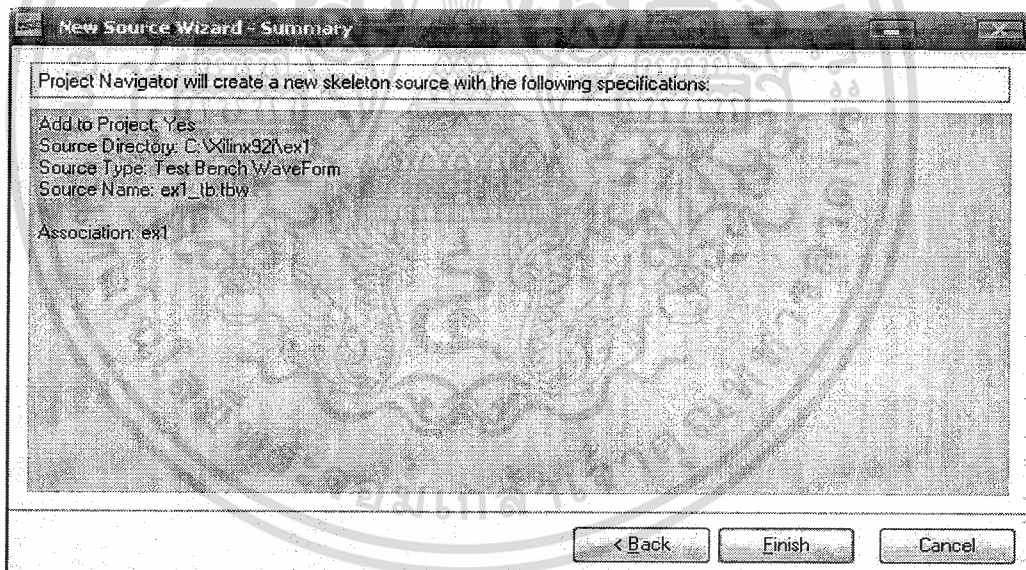


รูปที่ 4.24 หน้าต่าง New Source Wizard-Select Source Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 หน้าต่าง New Source Wizard-Associate Source



รูปที่ 4.26 New Source Wizard-Summary

โปรแกรมให้เรากำหนดค่าของเวลาที่ใช้ในการจำลองการทำงาน โดยค่าต่างๆ กำหนดดังนี้

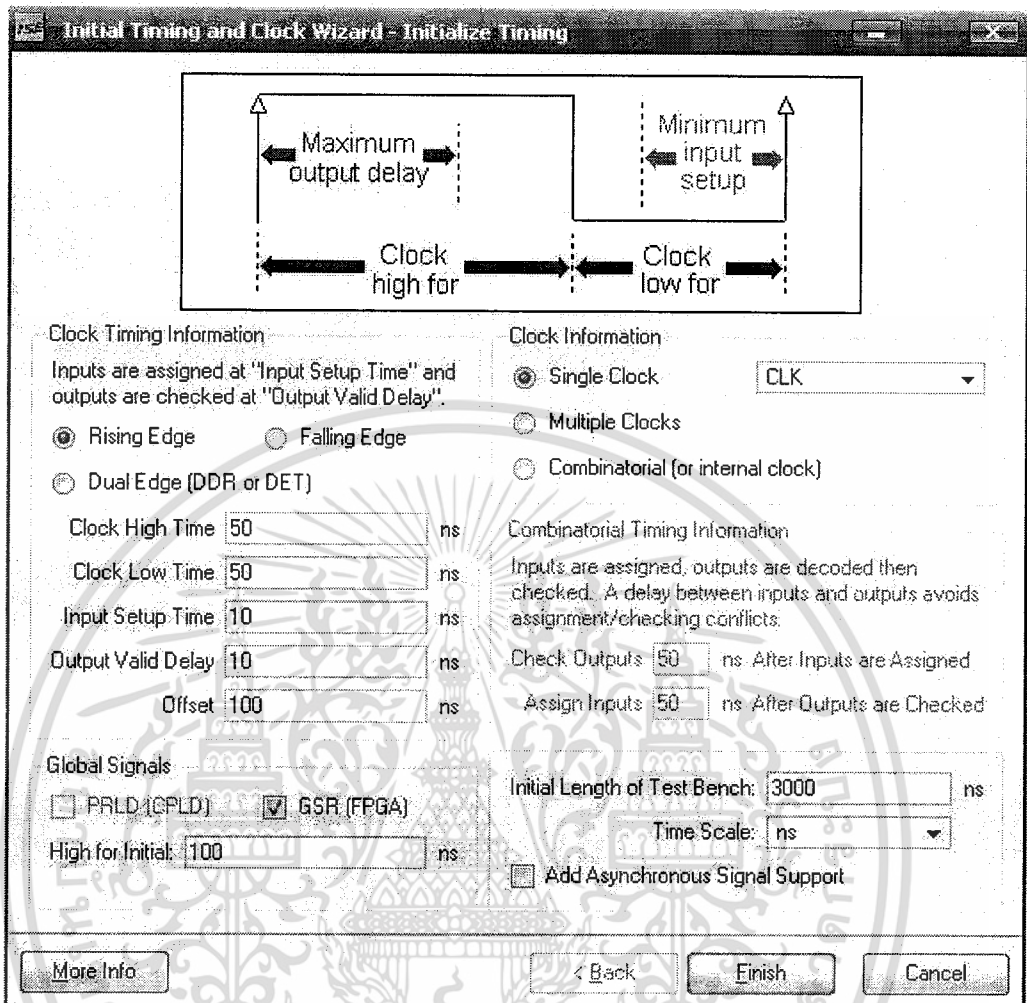
Clock High Time: 50 ns

Clock Low Time: 50 ns

(Clock freq. Is 1/100 ns = 10 MHz)

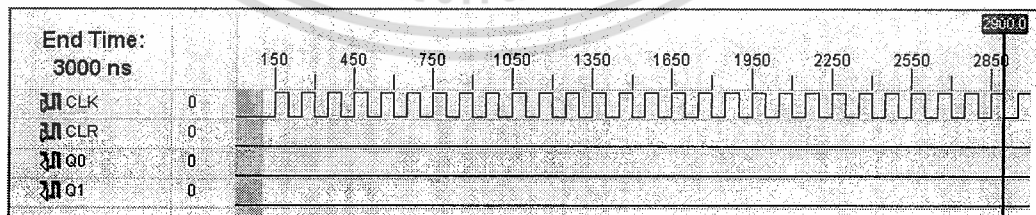
Input Setup Time: 10 ns Output Setup Time: 10 ns Initial Length of Test Bench: 3000 ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 หน้าต่าง Intial Timing and Clock Wizard

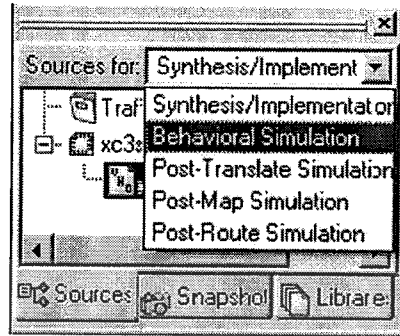
โปรแกรม HDL Bencher จะท การสร้าง Waveform ขึ้นมาดังรูปที่ 4.28



รูปที่ 4.28 สัญญาณที่สร้างขึ้นมา

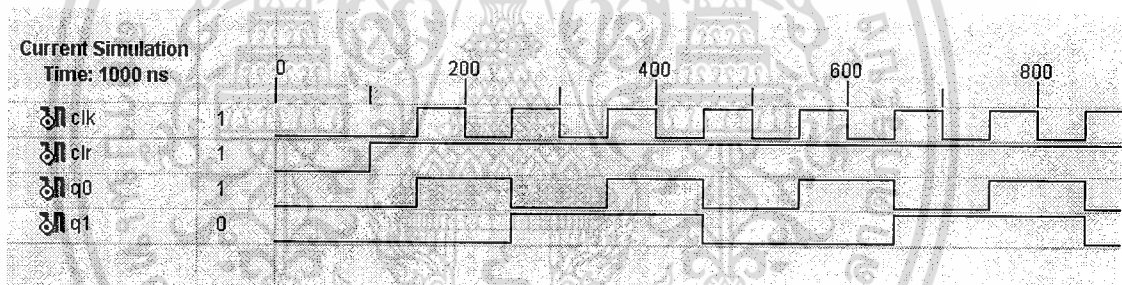
จากนั้นทำการ Save Waveform โดยคลิกปุ่ม  จากนั้นที่หน้าต่าง Sources เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในช่อง Sources for ให้เปลี่ยนเป็น Behavioral Model ดังรูปที่ 4.29



รูปที่ 4.29 หน้าต่าง Sources

ที่หน้าต่าง Processes ให้ดับเบิลคลิกที่ Simulate Behavioral Model โปรแกรมจะทำการเรียกโปรแกรม ISE Simulator และทำการ Compile และ Simulate วงจรเคาน์เตอร์ 4 บิตที่ออกแบบโดยใช้ไฟล์ทดสอบจาก ex1.tbw ดังรูปที่ 4.30



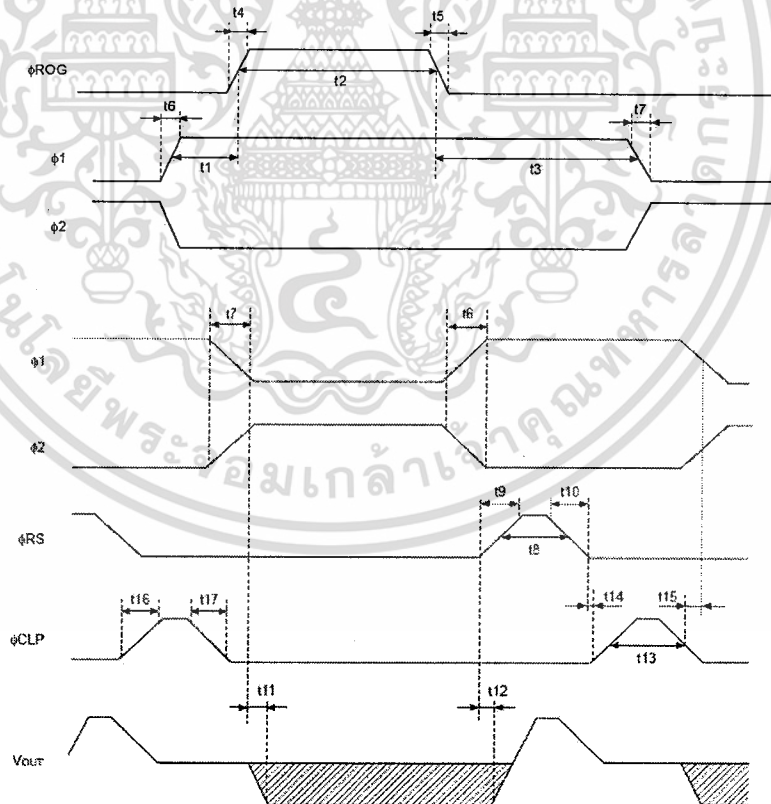
รูปที่ 4.30 รูปสัญญาณที่ทำการจำลองจากวงจรที่ออกแบบ

# บทที่ 5

## การออกแบบและทดลอง

### 5.1 หลักในการออกแบบ

ในการออกแบบวงจร Drive CCD เราได้ทำการออกแบบไว้ใน FPGA โดยใช้สัญญาณนาฬิกา Clock มีค่าเท่ากับ 25 MHz (ติดตามในบอร์ดFPGA) แล้วนำไปผ่านวงจรหารความถี่ให้เหลือ 5 MHz เพื่อที่จะนำไปสร้างสัญญาณให้กับ CCD ซึ่งสัญญาณที่ได้นี้ต้องมีลักษณะและมีคุณสมบัติตาม Data Sheet ของ CCD ตัวนั้น โดยในการทำเครื่องตรวจสอบสิ่งแปลกปลอมในเมล็ดข้าวนี้ได้ใช้ CCD เบอร์ ILX548K เพราะฉะนั้นเราต้องออกแบบสัญญาณให้กับ CCD ตรงตาม Data Sheet ซึ่งมีลักษณะของสัญญาณดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Item	Symbol	Min.	Typ.	Max.	Unit
$\phi$ ROG, $\phi$ 1 pulse timing	t1	50	100	—	ns
$\phi$ ROG pulse high level period	t2	3	5	—	$\mu$ s
$\phi$ ROG, $\phi$ 1 pulse timing	t3	1	2	—	$\mu$ s
$\phi$ ROG pulse rise time	t4	0	5	—	ns
$\phi$ ROG pulse fall time	t5	0	5	—	ns
$\phi$ 1 pulse rise time/ $\phi$ 2 pulse fall time	t6	0	20	—	ns
$\phi$ 1 pulse fall time/ $\phi$ 2 pulse rise time	t7	0	20	—	ns
$\phi$ RS pulse high level period	t8	30	50*1	—	ns
$\phi$ RS pulse rise time	t9	0	20	—	ns
$\phi$ RS pulse fall time	t10	0	20	—	ns
Signal output delay time	t11	—	50	—	ns
	t12	—	20	—	ns
$\phi$ CLP pulse high level period	t13	40	100	—	ns
$\phi$ CLP pulse timing	t14	40	100	—	ns
	t15	10	50	—	ns
$\phi$ CLP pulse rise time	t16	0	20	—	ns
$\phi$ CLP pulse fall time	t17	0	20	—	ns

\*1 These timing is the recommended condition under  $f\phi$  RS = 1MHz.

### รูปที่ 5.1 ข้อมูล Datasheet ILX548K

## 5.2 การออกแบบวงจร Drive CCD

### 5.2.1 การออกแบบสัญญาณ $\phi$ ROG

จากข้อมูล Datasheet เราจะพบว่าช่วงเวลาที่สัญญาณ  $\phi$  ROG มีสถานะเป็น High ว่ามีช่วงเวลาดังแต่ 50 us โดยถ้าสังเกตลักษณะอื่นๆจะพบว่าช่วงเวลาที่  $\phi$  ROG มีสถานะ High นั้นคาบของสัญญาณ  $\phi$  1 ,  $\phi$  2 ,  $\phi$  RS ,  $\phi$  CLP จะมีการเปลี่ยนแปลงตามไปด้วยเราจึงต้องการออกแบบสัญญาณ  $\phi$  ROG เป็นอันดับแรกเพราะสัญญาณ  $\phi$  1 ,  $\phi$  2 ,  $\phi$  RS ,  $\phi$  CLP ในข้อมูล Datasheet จะเทียบกับสัญญาณ  $\phi$  ROG

5.2.1.1) ทำการกำหนดช่วงขาขึ้นของสัญญาณ  $\phi$  ROG โดยเราจะทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity pp is
  Port ( S_CLK : in  STD_LOGIC:= '0';
        S_CLK1 : out STD_LOGIC);
end pp;

architecture Behavioral of pp is
  signal count : integer range 0 to 5500 :=0;
begin
  process (s_clk,count)
  begin
    if (s_clk' event and s_clk='1') then
      if count<6 then
        count<=count+1;
        s_clk1<='1';
      elsif count=5500 then
        s_clk1<='0';
        count<=0;
      else
        s_clk1<='0';
        count<=count+1;
      end if;
    end if;
  end process;
end architecture Behavioral of pp;

```

รูปที่ 5.2 คำสั่งในการกำหนดขาขึ้นของสัญญาณ  $\phi$  ROG

5.2.1.2) จากนั้นทำการกำหนดช่วงขาลงของสัญญาณ  $\phi$  ROG โดยเราจะทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

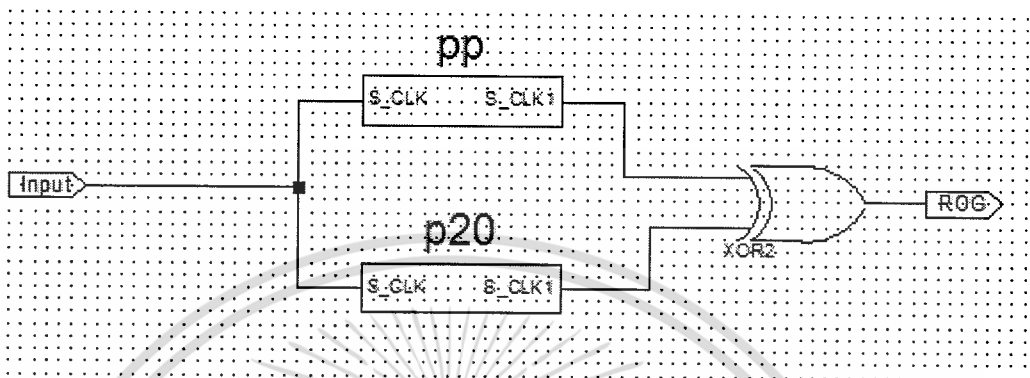
entity p20 is
    Port ( S_CLK : in  STD_LOGIC:= '0';
          S_CLK1 : out STD_LOGIC);
end p20;

architecture Behavioral of p20 is
    signal count : integer range 0 to 5500 :=0;
begin
    process (s_clk,count)
    begin
        if (s_clk' event and s_clk='1') then
            if count<22 then
                count<=count+1;
                s_clk1<='1';
            elsif count=5500 then
                s_clk1<='0';
                count<=0;
            else
                s_clk1<='0';
                count<=count+1;
            end if;
        end if;
    end process;
end architecture Behavioral of p20;

```

รูปที่ 5.3 คำสั่งในการกำหนดขาลงของสัญญาณ  $\phi$  ROG

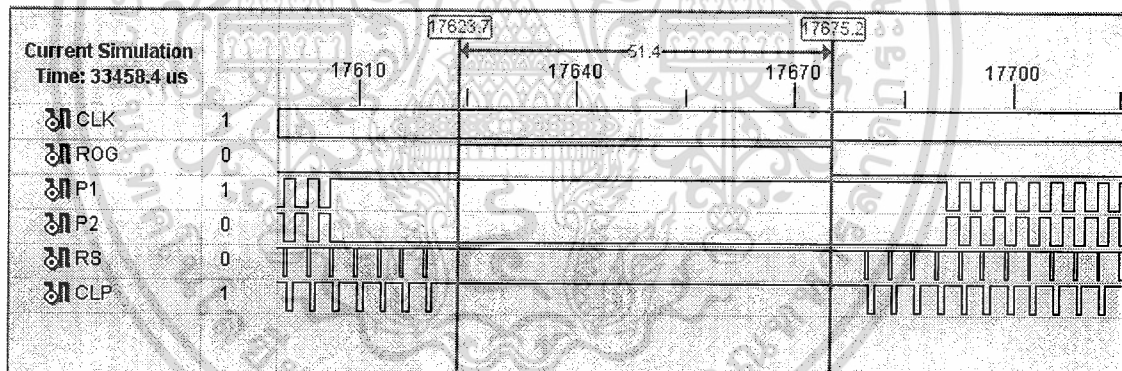
5.2.1.3) เมื่อทำการกำหนดขอบขาขึ้นแล้วขอบขาลงของสัญญาณ  $\phi$  ROG ได้แล้วเราจะนำสัญญาณที่ได้ผ่าน Logic Gate เพื่อให้ได้สัญญาณตาม Datasheet



รูปที่ 5.4 Block Diagram สัญญาณ  $\phi$  ROG

โดยใน Block Diagram PP และ P20 จะบรรจุคำสั่ง VHDL ดังรูปที่ 5.2 และ 5.3

5.2.1.4) ทำการ Simulator วงจรสร้างสัญญาณ  $\phi$  ROG ที่ออกแบบ



รูปที่ 5.5 ผลการ SIMULATOR ของสัญญาณ  $\phi$  ROG

เมื่อเทียบสัญญาณใน Datasheet  $t_2 = 5$  us ขึ้นไป โดยคาบเวลาที่ SIMULATOR ได้นั้นมีค่ามากกว่า 5 us ซึ่งเป็นไปตามข้อมูลใน Datasheet โดยไม่ต่ำกว่าค่า Min

## 5.2.2 การออกแบบสัญญาณ $\phi$ 1, $\phi$ 2

จากข้อมูล Datasheet สัญญาณของ  $\phi$  1 และ  $\phi$  2 คาบเวลาจะเหมือนกันทุกประการจะต่างกันแค่สัญญาณมีการกลับเฟสกันเท่านั้น โดยหลักการจะเหมือนกันกับการออกแบบสัญญาณ  $\phi$  ROG

5.2.2.1) ทำการกำหนดช่วงขาขึ้นของสัญญาณ  $\phi 1$  และ  $\phi 2$  โดยเราจะทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity p25 is
    Port ( s_clk : in  STD_LOGIC:= '0';
          s_clk1 : out STD_LOGIC);
end p25;

architecture Behavioral of p25 is
    signal count : integer range 0 to 5500 :=0;
begin
    process (s_clk,count)
    begin
        if (s_clk' event and s_clk='1') then
            if count<1 then
                count<=count+1;
                s_clk1<='1';
            elsif count=5500 then
                s_clk1<='0';
                count<=0;
            else
                s_clk1<='0';
                count<=count+1;
            end if;
        end if;
    end process;

end Behavioral;

```

รูปที่ 5.6 คำสั่งในการกำหนดขาลงของสัญญาณ  $\phi 1$

5.2.2.2) ทำการกำหนดช่วงขาลงของสัญญาณ  $\phi 1$  และ  $\phi 2$  โดยเราจะทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity p26 is
    Port ( s_clk : in  STD_LOGIC:= '0';
          s_clk1 : out STD_LOGIC);
end p26;

architecture Behavioral of p26 is

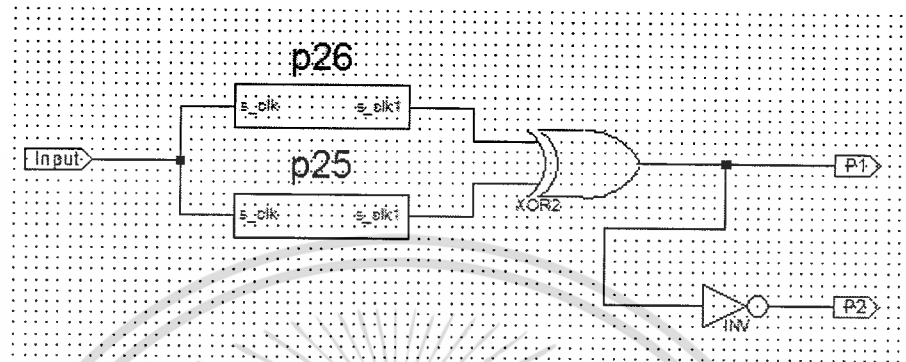
    signal count : integer range 0 to 5500 :=0;
begin
    process (s_clk,count)
    begin
        if (s_clk' event and s_clk='1') then
            if count<27 then
                count<=count+1;
                s_clk1<='1';
            elsif count=5500 then
                s_clk1<='0';
                count<=0;
            else
                s_clk1<='0';
                count<=count+1;
            end if;
        end if;
    end process;

end Behavioral;

```

รูปที่ 5.7 คำสั่งในการกำหนดขาลงสัญญาณ  $\phi 1$

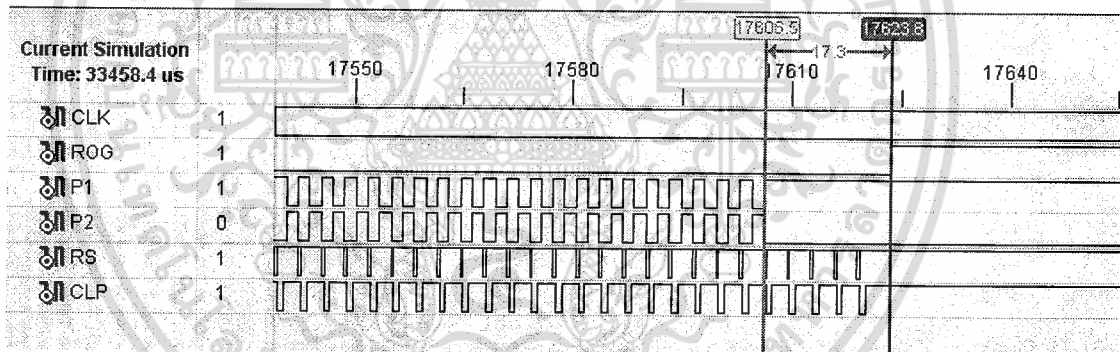
5.2.2.3) เมื่อทำการขอขาขึ้นแล้วขอขาตลงของสัญญาณ  $\phi 1$  และ  $\phi 2$  ได้แล้วจะนำสัญญาณที่ได้ผ่าน Logic Gate เพื่อให้สัญญาณตาม Datasheet



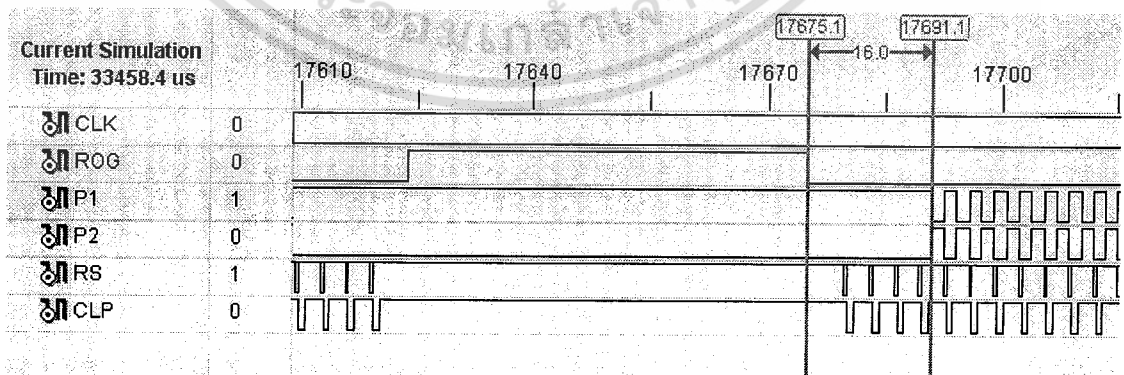
รูปที่ 5.8 Block Diagram ของสัญญาณ  $\phi 1$  และ  $\phi 2$

โดยใน Block Diagram P26 และ P25 จะบรรจุคำสั่ง VHDL ดังรูปที่ 5.6 และ 5.7

5.2.2.4) ทำการ Simulator วงจรสร้างสัญญาณ  $\phi 1$  และ  $\phi 2$  ที่ออกแบบ



รูปที่ 5.9 ขอขาขึ้นของสัญญาณ  $\phi$  ROG เทียบกับสัญญาณ  $\phi 1$  และ  $\phi 2$



รูปที่ 5.10 ขอขาลงของสัญญาณ  $\phi$  ROG เทียบกับสัญญาณ  $\phi 1$  และ  $\phi 2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเทียบกับสัญญาณใน Datasheet t1, t3 มีค่า Min อยู่ที่ 50 ns และ 1us แต่คาบสัญญาณที่ได้จากการSIMULATOR นั้นมีค่ามากกว่าค่า Min ซึ่งเป็นไปตามข้อมูลใน Datasheet โดยไม่ต่ำกว่าค่า Min

### 5.2.3 การออกแบบสัญญาณ $\phi$ RS และ $\phi$ CLP

5.2.3.1) ก่อนที่จะทำการสร้างสัญญาณ  $\phi$  RS และ  $\phi$  CLP จะต้องทำการออกแบบวงจร Decodeder เพื่อทำการ Decodeder สัญญาณเข้า 3 ช่องแล้วออกมา 8 ช่องโดยการออกแบบได้ใช้ภาษา VHDL ในการเขียนโดยมีคำสั่งดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Decodeder is
    Port ( Din : in  STD_LOGIC_VECTOR (2 downto 0);
          Dout : out STD_LOGIC_VECTOR (7 downto 0));
end Decodeder;

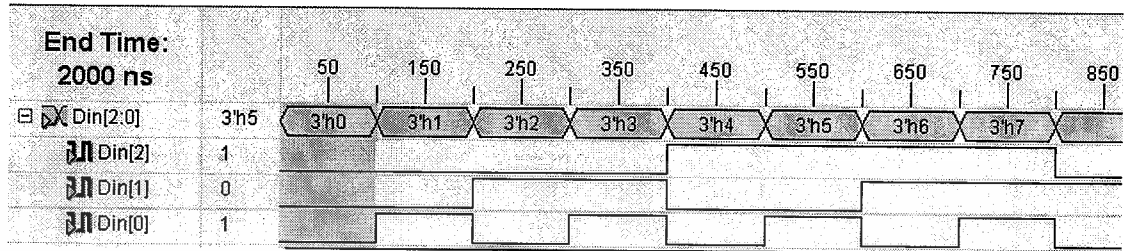
architecture Behavioral of Decodeder is
    signal tmpout :STD_LOGIC_VECTOR (7 downto 0);
begin
    process (Din)
    begin
        tmpout<= "11111111" ;
        case Din is
            when "000" => tmpout (0) <= '0' ;
            when "001" => tmpout (1) <= '0' ;
            when "010" => tmpout (2) <= '0' ;
            when "011" => tmpout (3) <= '0' ;
            when "100" => tmpout (4) <= '0' ;
            when "101" => tmpout (5) <= '0' ;
            when "110" => tmpout (6) <= '0' ;
            when others => tmpout (7) <= '0' ;
        end case;
    end process;
    Dout<=tmpout;
end Behavioral;

```

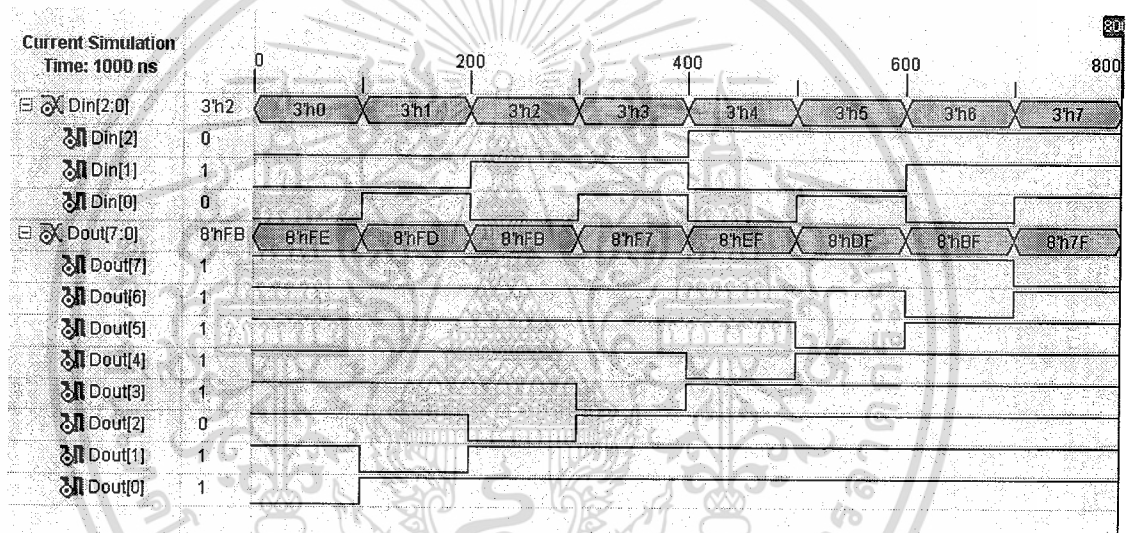
รูปที่ 5.11 คำสั่งของวงจร Decodeder 3 to 8

### 5.2.3.2) ทำการ SIMULATOR โปรแกรม Decoder 3 to 8 โดยกำหนดค่า Din ตาม

มีค่า



รูปที่ 5.12 ค่า Din ที่กำหนดในการ SIMULATOR



รูปที่ 5.13 ผลการ SIMULATOR ของโปรแกรม Decoder 3 to 8

5.2.3.3) เมื่อเขียนโปรแกรม Decodeder 3 to 8 แล้วต่อไปก็ทำการกำหนดช่วง สัญญาณขาขึ้นของสัญญาณ  $\phi$  RS โดยทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity p3 is
    Port ( s_clk : in  STD_LOGIC:= '0';
          s_clk1 : out STD_LOGIC);
end p3;

architecture Behavioral of p3 is
    signal count : integer range 0 to 5500 :=0;
begin
    process (s_clk,count)
    begin
        if (s_clk' event and s_clk='1') then
            if count<5 then
                count<=count+1;
                s_clk1<='1';
            elsif count=5500 then
                s_clk1<='0';
                count<=0;
            else
                s_clk1<='0';
                count<=count+1;
            end if;
        end if;
    end process;
end Behavioral;

```

รูปที่ 5.14 คำสั่งในการกำหนดขาขึ้นของสัญญาณ  $\phi$  RS

5.2.3.4) ทำการกำหนดช่วงสัญญาณขาของสัญญาณ  $\phi$ RS โดยทำการเขียนโปรแกรมโดยใช้ภาษา VHDL ดังนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity p23 is
    Port ( s_clk : in  STD_LOGIC:= '0';
          s_clk1 : out STD_LOGIC);
end p23;

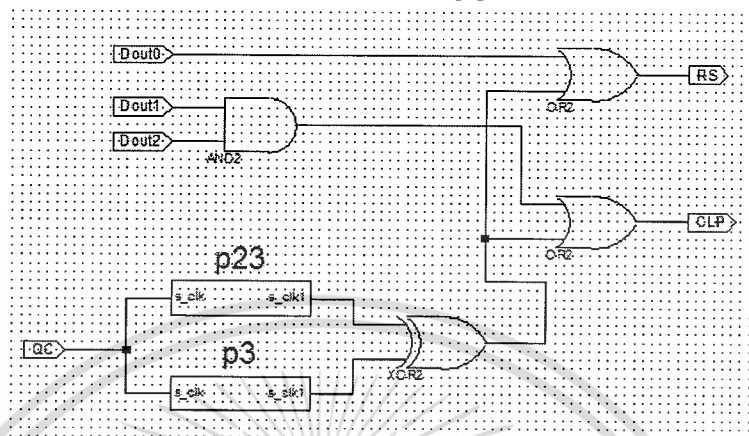
architecture Behavioral of p23 is
    signal count : integer range 0 to 5500 :=0;
begin
    process (s_clk,count)
    begin
        if (s_clk' event and s_clk='1') then
            if count<23 then
                count<=count+1;
                s_clk1<='1';
            elsif count=5500 then
                s_clk1<='0';
                count<=0;
            else
                s_clk1<='0';
                count<=count+1;
            end if;
        end if;
    end process;

end Behavioral;

```

รูปที่ 5.15 คำสั่งในการกำหนดขาลงของสัญญาณ  $\phi$ RS

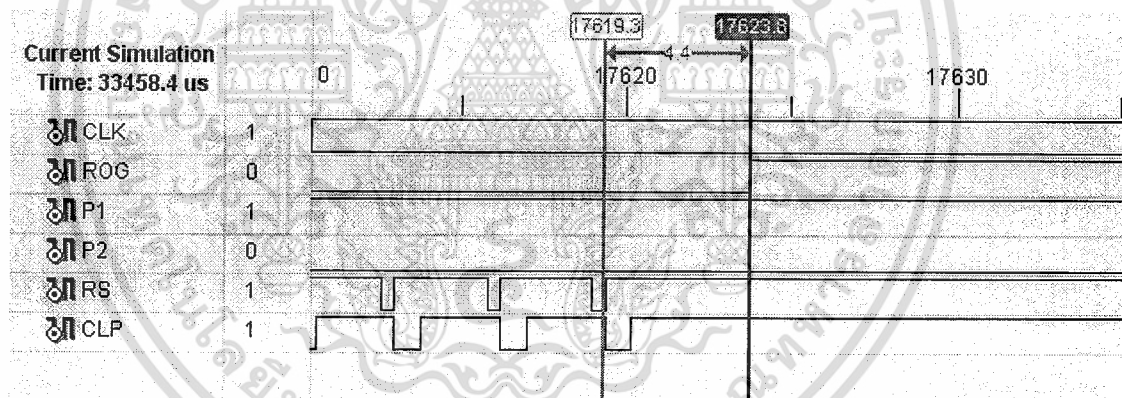
5.2.3.5) เมื่อทำการกำหนดขอบขาขึ้นและขอบขาลงของสัญญาณ  $\phi$  RS และ  $\phi$  CLP ได้แล้วจะนำสัญญาณที่ได้ผ่าน Logic Gate เพื่อให้ได้สัญญาณตาม Datasheet



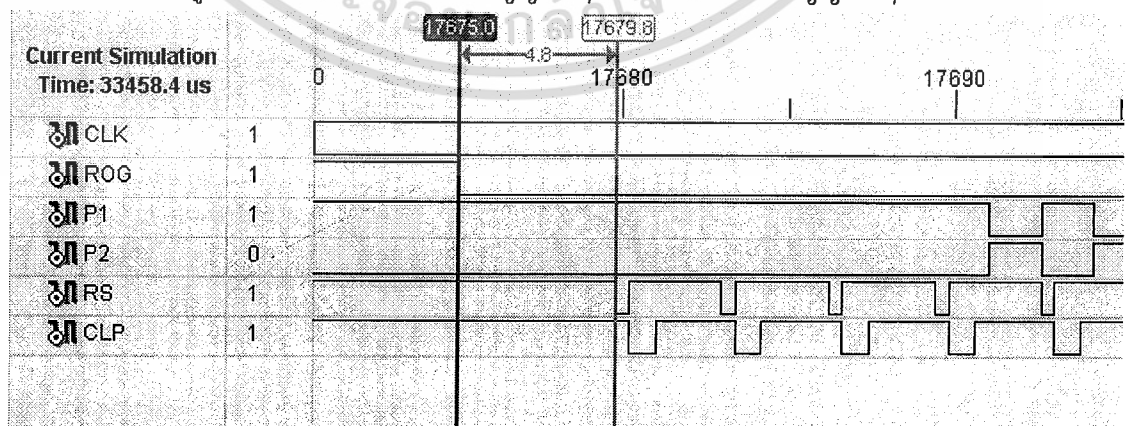
รูปที่ 5.16 Block Diagram  $\phi$  RS และ  $\phi$  CLP

โดยใน Block Diagram P23 และ P3 จะบรรจุคำสั่ง VHDL ดังรูปที่ 5.14 และ 5.15

5.2.3.6) ทำการ Simulator วงจรสร้างสัญญาณ  $\phi$  RS และ  $\phi$  CLP ที่ออกแบบ



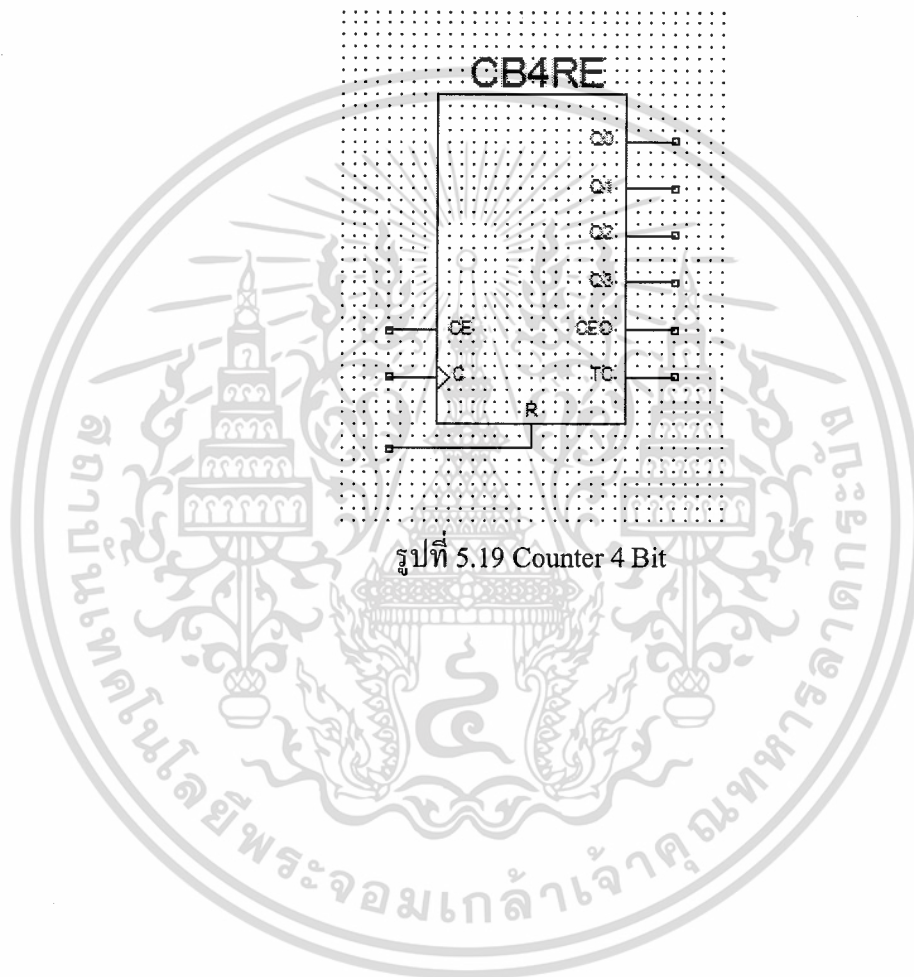
รูปที่ 5.17 ขอบขาขึ้นของสัญญาณ  $\phi$  ROG เทียบกับ สัญญาณ  $\phi$  RS



รูปที่ 5.18 ขอบขาลงของสัญญาณ  $\phi$  ROG เทียบกับ สัญญาณ  $\phi$  RS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3.6) นำวงจรที่ออกแบบมาทั้งหมดมาต่อเข้าด้วยกันเพื่อให้ได้สัญญาณตาม Datasheet ซึ่งจะต้องให้วงจร Counter 4 Bit ดังรูปที่ 5.19 ซึ่งมีมาให้แล้วในโปรแกรม เพื่อเข้าวงจร Decoder 3 to 8 และก่อนที่จะนำสัญญาณมาใช้งานต้องทำการหารความถี่ให้เหลือประมาณ 5 MHz ซึ่งโปรแกรม Divider สามารถเขียนโดยภาษา VHDL ดังรูปที่ 5.20 จากนั้นนำวงจรที่ออกแบบมาทั้งหมดต่อเข้าด้วยกันตามรูปที่ 5.21



รูปที่ 5.19 Counter 4 Bit

```

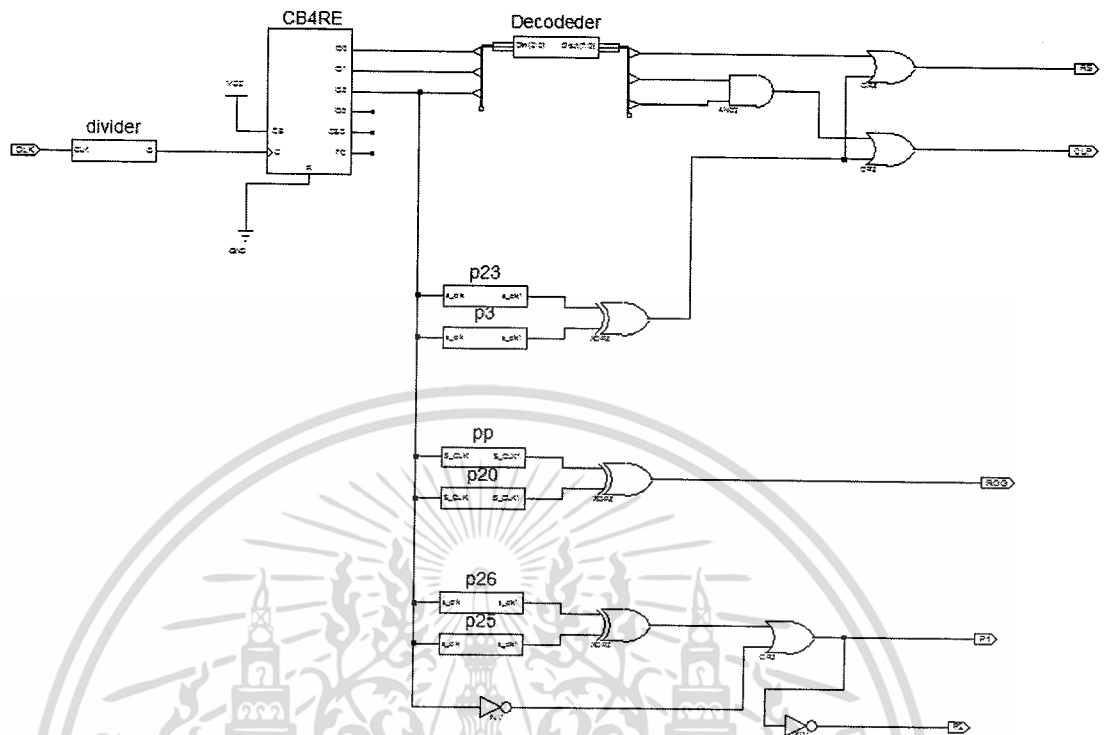
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity divider is
generic (fin: integer := 25000000;
        fout: integer := 500000000);
Port ( CLK : in  STD_LOGIC;
      Q : out  STD_LOGIC);
end divider;

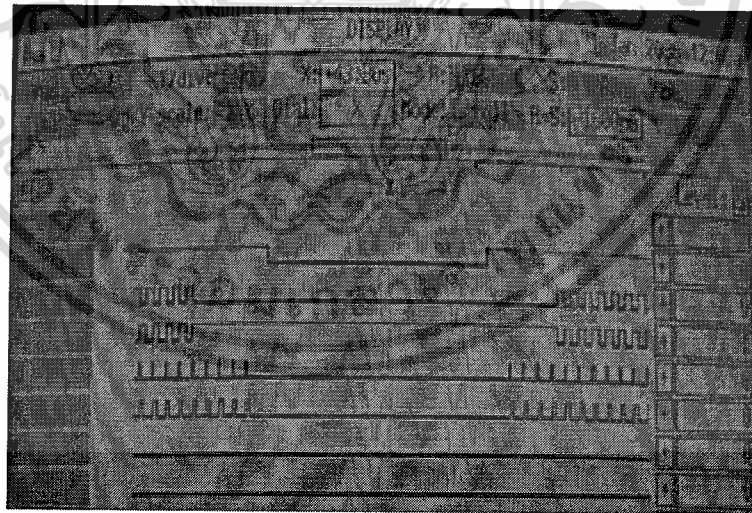
architecture Behavioral of divider is
signal COUNT : integer range 0 to (fin/(2*fout)) ;
signal qs : std_logic := '0';
begin
process (CLK)
begin
if CLK'event and CLK = '1' then
if (COUNT >= (fin/(2*fout)-1)) then
COUNT <= 0;
qs <= not(qs);
else
COUNT <= COUNT +1;
end if;
end if;
end process;
Q <= qs ;
end Behavioral;

```

รูปที่ 5.20 โปรแกรม Divider



รูปที่ 5.21 วงจร Drive CCD



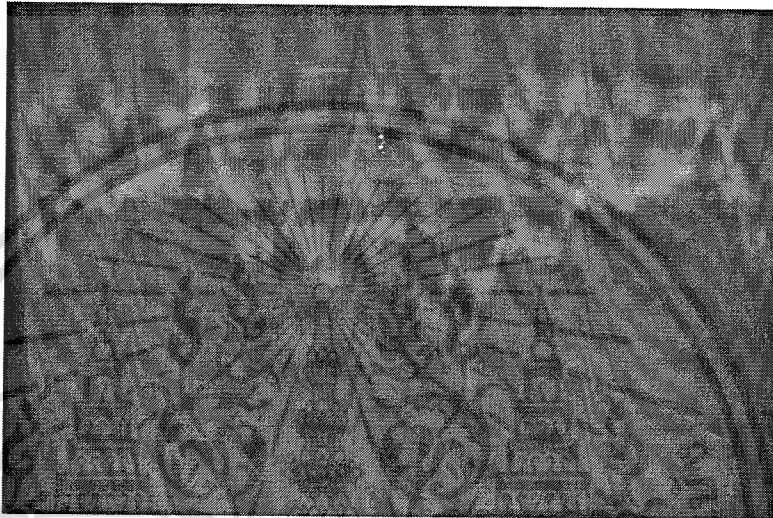
รูปที่ 5.22 รูปสัญญาณที่ได้มาจากบอร์ด FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

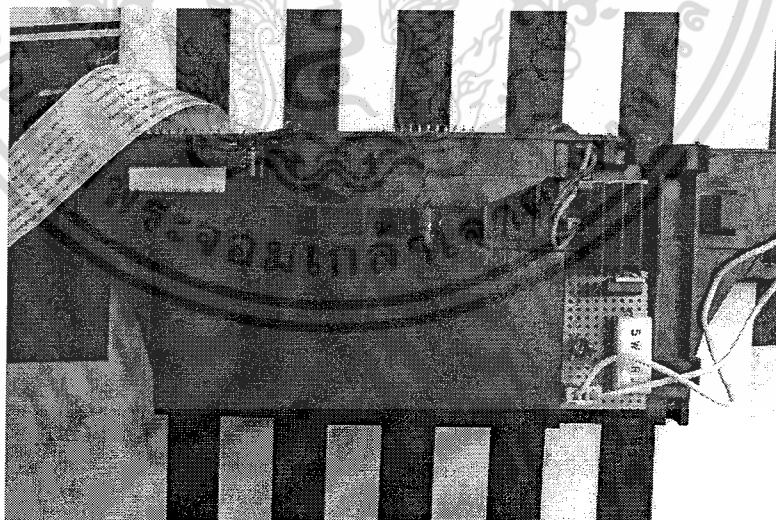
### 5.3 รูปสัญญาณ OUTPUT จาก CCD

#### 5.3.1) รูปสัญญาณที่มาจาก CCD โดยตรง

รูปสัญญาณที่ได้จาก CCD ขณะที่ยังไม่มีการตรวจจับวัตถุ โดยได้ทำการขยายให้เห็นช่วงของสัญญาณชัดเจนขึ้น



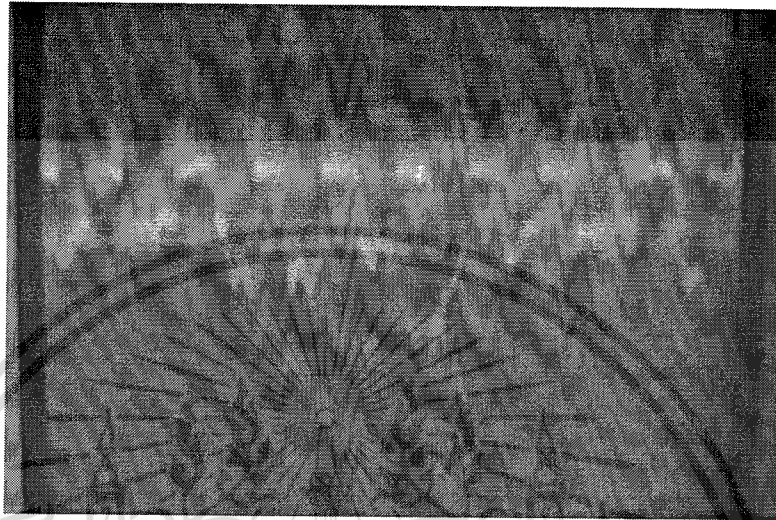
รูปที่ 5.23 รูปสัญญาณ OUTPUT จาก CCD ขณะที่ยังไม่ได้ทำการตรวจจับวัตถุ



รูปที่ 5.24 แถบกระดาศที่มีสี่ขา - คำที่ใช้ในการทดสอบ

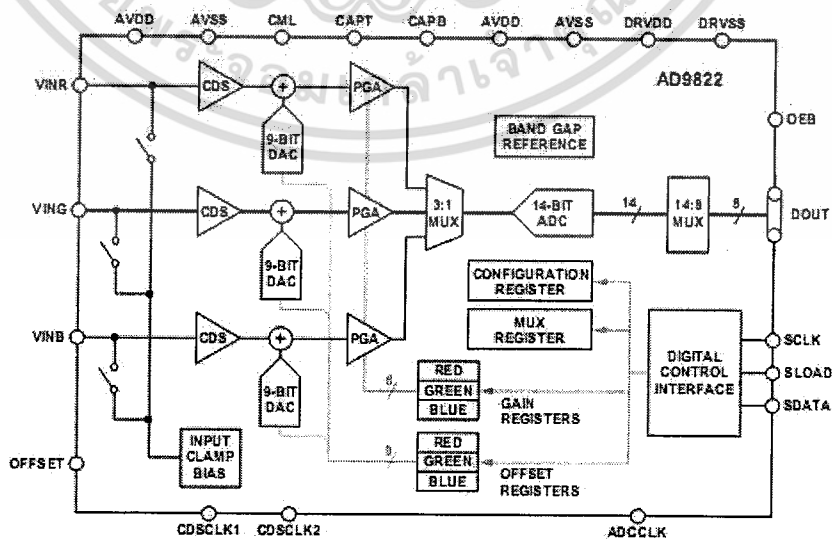
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2) รูปสัญญาณ OUTPUT R ของ CCD ที่มีการตรวจจับภาพ



รูปที่ 5.25 รูปสัญญาณ OUTPUT R ของ CCD ที่มีการตรวจจับภาพสีขาว

เมื่อทำการสร้างสัญญาณ Drive CCD เสร็จแล้ว ก็นำสัญญาณจากตัว CCD ไปเข้าวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลโดยใช้ IC สำเร็จรูปเบอร์ AD9822 ซึ่งเป็น IC ที่ใช้สำหรับแปลงสัญญาณจากตัว CCD โดยเฉพาะ โดยจะให้เอาท์พุทออกมาเป็นข้อมูลทางดิจิทัลขนาด 8 Bit ซึ่งภายในของ IC Signal Processor เบอร์ AD9822 จะเป็นดังรูปที่ 5.30

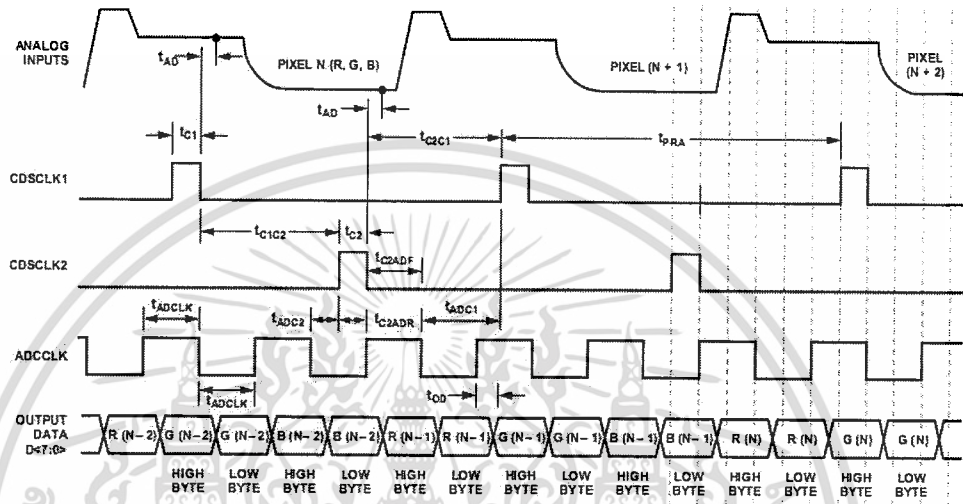


รูปที่ 5.26 Block Diagram ภายในของ IC Signal Processor เบอร์ AD 9822

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

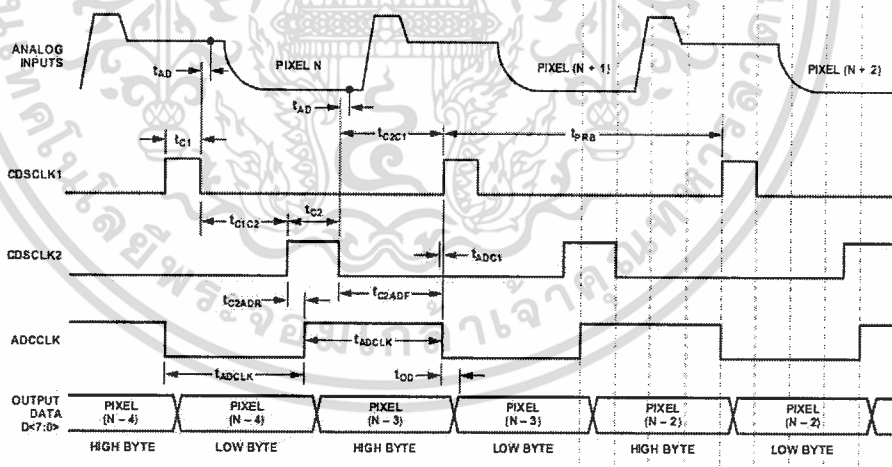
IC Signal Processor เบอร์ AD 9822 จะมีโหมดการทำงานหลักอยู่ 4 โหมด  
ซึ่งการทำงานของแต่ละ โหมดจะมีรูปแบบสัญญาณที่แตกต่างกันไปในแต่ละโหมดดังนี้

### 1.) 3 Ch CDS Mode



รูปที่ 5.27 สัญญาณของ 3 Ch CDS Mode

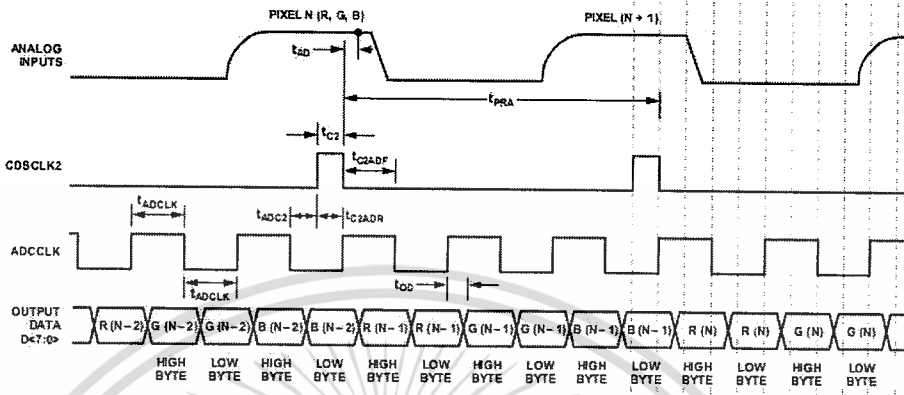
### 2.) 1 Ch CDS Mode



รูปที่ 5.28 สัญญาณของ 1 Ch CDS Mode

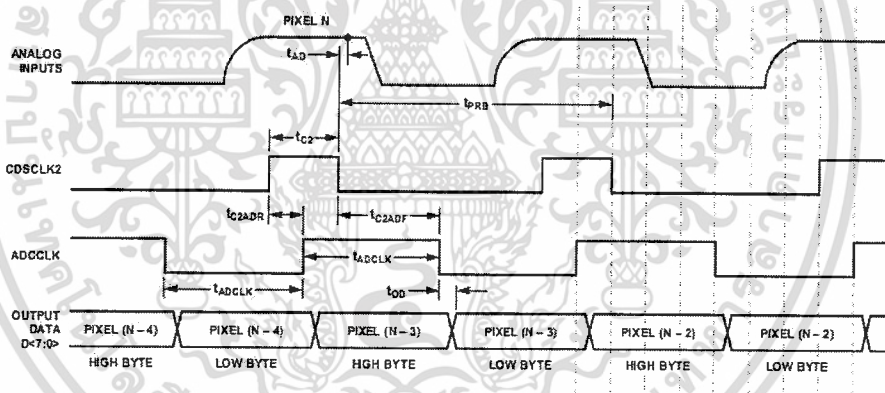
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) 3 Ch SHA Modes



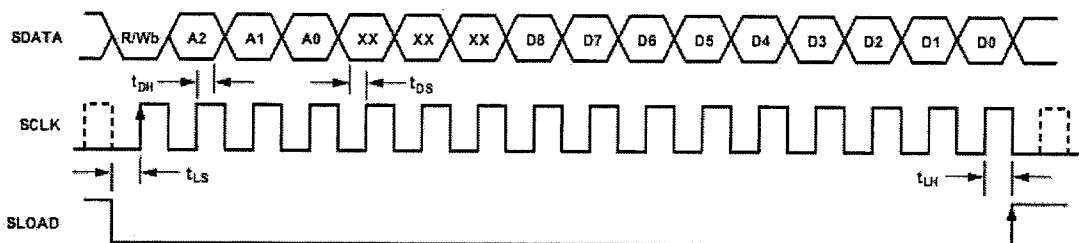
รูปที่ 5.29 สัญญาณของ 3 Ch SHA Mode

4.) 1 Ch SHA Mode



รูปที่ 5.30 สัญญาณของ 1 Ch CDS Mode

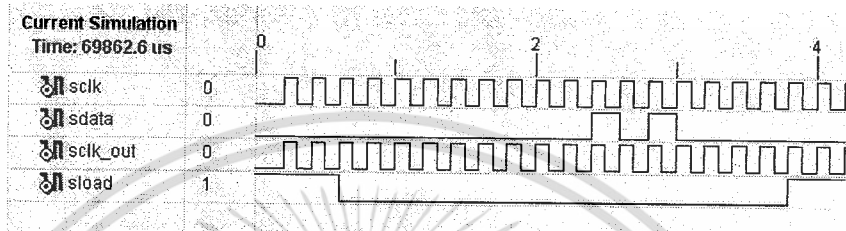
IC Signal Processor AD 9822 สามารถกำหนดโหมดการทำงานได้โดยการส่งสัญญาณ Register โดยรูปแบบของสัญญาณ Register นั้นจะเป็นคังรูปที่ 5.35



รูปที่ 5.31 รูปแบบของสัญญาณ Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

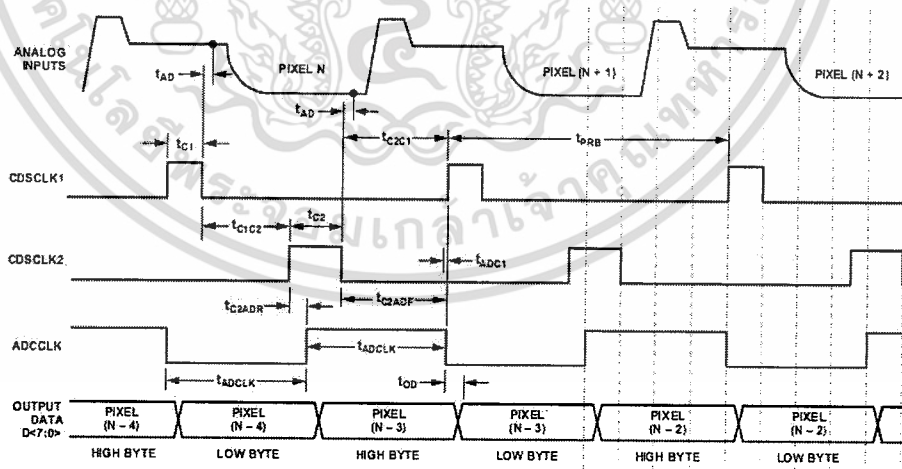
โดยค่าของสัญญาณ Register นั้นสามารถดูได้ใน Data sheet ในส่งของภาคผนวก ในการสร้างสัญญาณ Register จะใช้ภาษา VHDL ในการออกแบบ โดยกำหนดให้ทำงานในโหมด 1 Ch CDS Mode โดยโค้ดของภาษา VHDL ดูได้จากภาคผนวก เมื่อออกแบบโค้ดภาษา VHDL เสร็จก็ทำการ Simulation โค้ดที่เขียนไว้ว่าสัญญาณจะออกมาตรงที่ต้องการหรือไม่ ซึ่งรูปสัญญาณ Register ที่ทำการ Simulation ได้ดังรูปที่ 5.36



รูปที่ 5.32 ผลของการ Simulation ของสัญญาณ Register ที่ออกแบบไว้

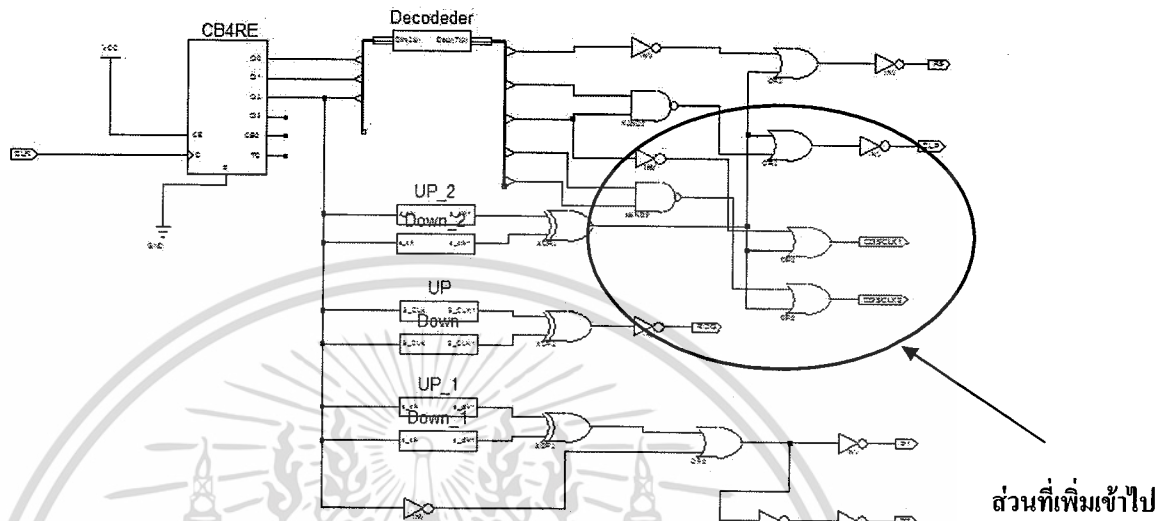
#### 5.4 การสร้างสัญญาณ CDSCLK1 และ CDSCLK2

การทำงานของ IC Signal Processor จะนำสัญญาณที่มาจากตัว CCD มาทำการตรวจสอบโดยใช้สัญญาณ CDSCLK 1 และ CDSCLK 2 ในการหาจุด Reference และจุดของ Data Level ซึ่งรูปแบบของสัญญาณ CDSCLK 1 และ CDSCLK 2 จะเป็นดังรูปที่ 5.37

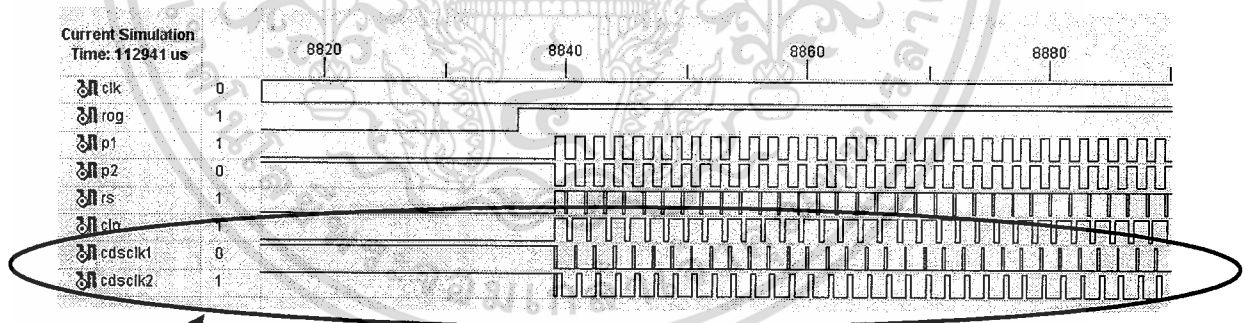


รูปที่ 5.33 รูปแบบของสัญญาณ CDSCLK 1 และ CDSCLK 2

ในการสร้างสัญญาณ CDSCLK 1 และ CDSCLK 2 จะใช้วงจร Drive CCD ที่ออกแบบไว้แล้ว โดยจะเพิ่ม Logic Gate เข้าไปดังรูปที่ 5.38



รูปที่ 5.34 วงจรสร้างสัญญาณ CDSCLK 1 และ CDSCLK 2  
ทำการ Simulation วงจรที่ออกแบบได้รูปแบบของสัญญาณดังรูปที่ 5.40

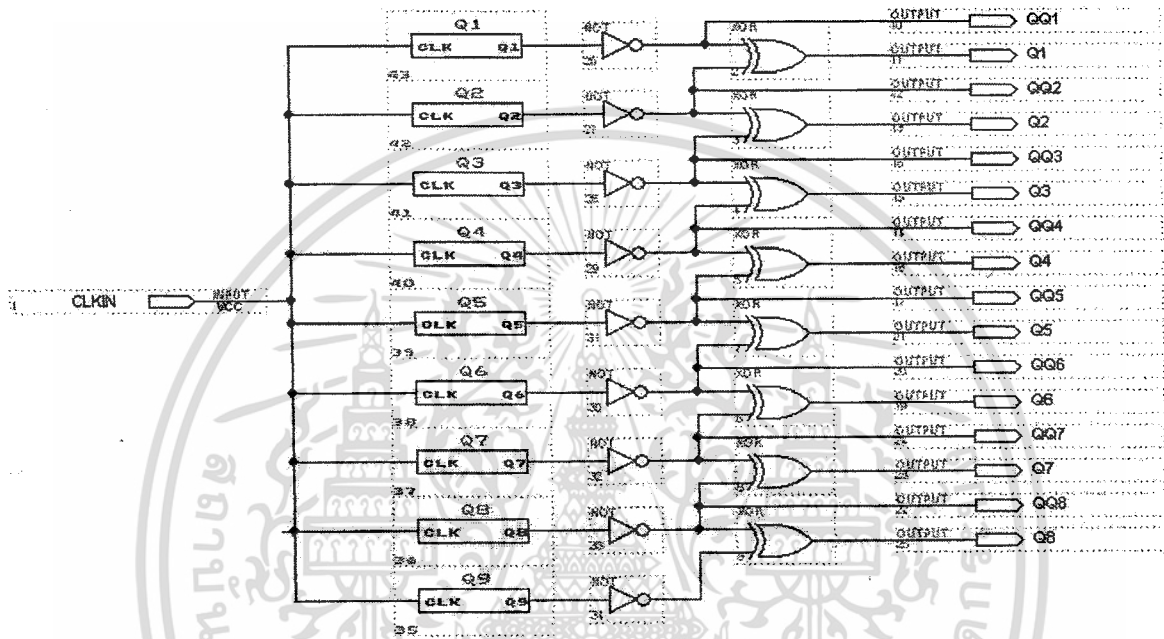


รูปที่ 5.35 รูปสัญญาณ CDSCLK 1 และ CDSCLK 2

สัญญาณ CDCCLK1 และ 2

## 5.5 การทำงานของวงจรแยกราง

เมื่อได้สัญญาณข้อมูลจาก CCD เรานำสัญญาณที่ได้มาประมวลผลดูว่าสัญญาณนั้นหรือ เมล็ดข้าวอยู่ในรางไหน โดยอาศัยวงจรแยกรางที่เราได้ทำการออกแบบไว้ในตัว FPGA ซึ่งมีรูปวงจรถูกบดดังนี้



รูปที่ 5.36 วงจรแยกรางที่ได้ออกแบบไว้ใน FPGA

วงจรแยกรางนี้จะเป็นตัวสร้างสัญญาณเพื่อเช็คว่าสัญญาณที่ได้จาก CCD นั้นอยู่ที่รางไหน โดยภายในวงจรเราได้ทำการแบ่งจุดภาพบนเส้นภาพว่าแต่ละจุดจะอยู่ในรางใดและเมื่อมีเมล็ดข้าวสีดำผ่านเราก็จะนำข้อมูลที่ได้มาประมวลผลทางดิจิทัลกับวงจรแยกรางนี้ จึงทำให้เราทราบได้ว่ามีสิ่งแปลกปลอมเข้ามาในรางใด

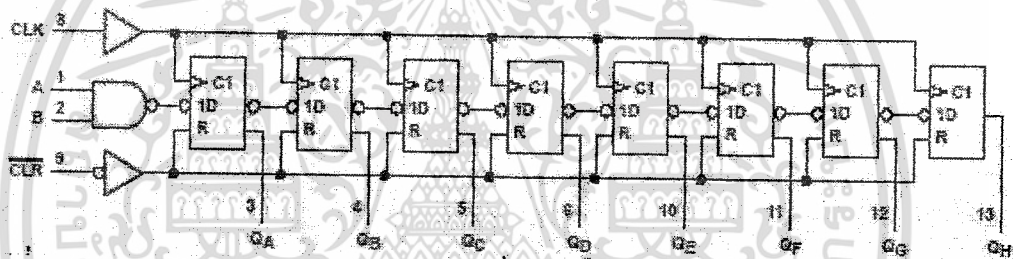
### 5.5.1) การทำงานของ Shift Register

จะเป็นการแปลงข้อมูลแบบส่งเข้าอนุกรม เปลี่ยนให้ส่งออกเป็นแบบขนาน เพื่อจะนำข้อมูลที่ได้ออกไปใช้งานต่อ

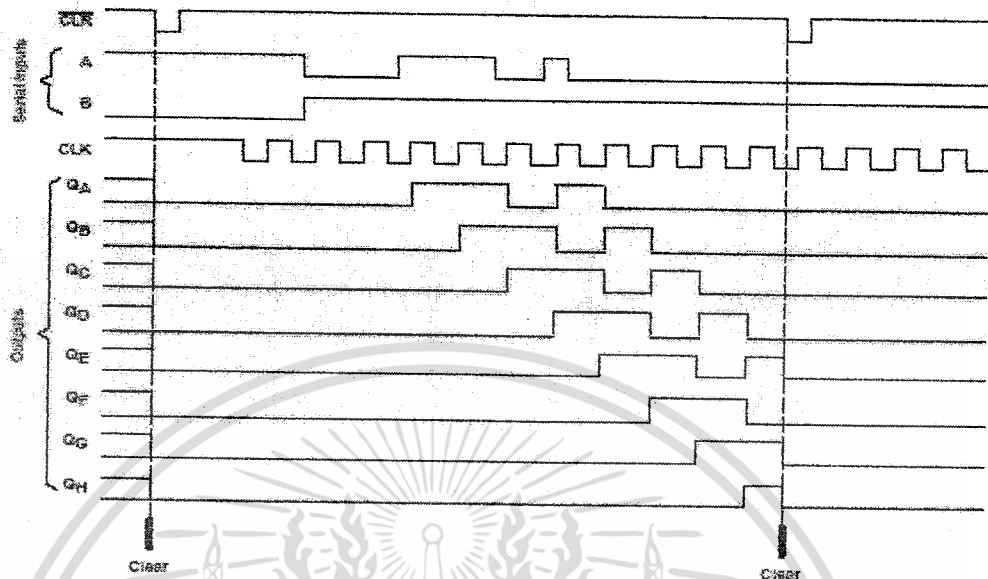
ตัวเลื่อนข้อมูลแบบ SIPO สามารถนำเอาที่พืทมาใช้ได้ 2 ลักษณะ คือ

- นำเอาที่พืทขนานมาใช้ คือ OUTPUT QA, QB, QC, และ QD
- นำเอาที่พืทอนุกรมมาใช้ คือ จาก QA มาใช้ การนำเอาที่พืทขนานมาใช้ จะต้องอาศัยการเลื่อน 4 จังหวะ จึงจะได้ข้อมูลครบถ้วน ส่วนข้อมูลอนุกรมจะได้ครบก็ต่อเมื่อต้องอาศัยการเลื่อน 8 จังหวะ

ต่อไปนี้เป็นตัวอย่างการใช้ฟลิปฟลอปชนิด D ในการเลื่อนข้อมูลแบบรับเข้าอนุกรมและส่งออกขนาน



รูปที่ 5.37 การนำ D FLIPFLOP มาจัดเป็นวงจรรับเข้าแบบอนุกรมและส่งออกแบบขนาน

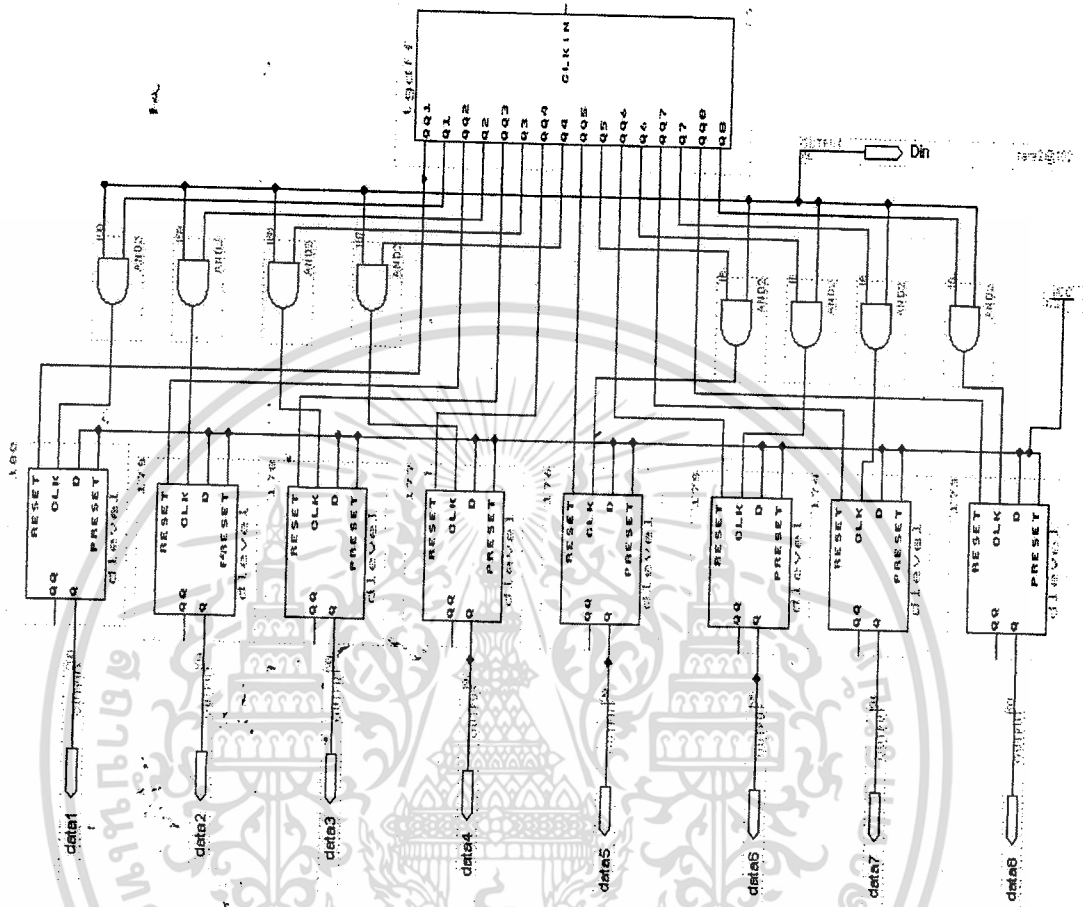


รูปที่ 5.38 หลังจากมีสัญญาณ CLOCK เข้ามาครบ 8 ลูก Data ที่ส่งเข้ามาในจังหวะที่สัมพันธ์กับสัญญาณ CLOCK จะปรากฏออกมาที่ขา QA, QB, QC, QD, QE, QF, QG, และ QH

การรับข้อมูลแบบอนุกรมในลักษณะนี้สัญญาณ CLOCK ของฝ่ายรับจะต้องมีคาบเวลาเท่ากับสัญญาณ CLOCK ของฝ่ายส่ง (แต่ไม่จำเป็นต้องเป็นสัญญาณเดียวกัน) ข้อมูลที่รับได้จึงจะเป็นข้อมูลที่ถูกต้อง

#### 5.5.2) การทำงานร่วมกันของวงจรแยกรางกับวงจร Shift Register

เมื่อเราทำการออกแบบวงจรแยกรางกับวงจร Shift Register เสร็จเรียบร้อยแล้ว เราก็จะนำวงจรทั้งสองนั้นมาต่อใช้งานร่วมกันจะได้วงจรแสดงในรูปที่ 5.39

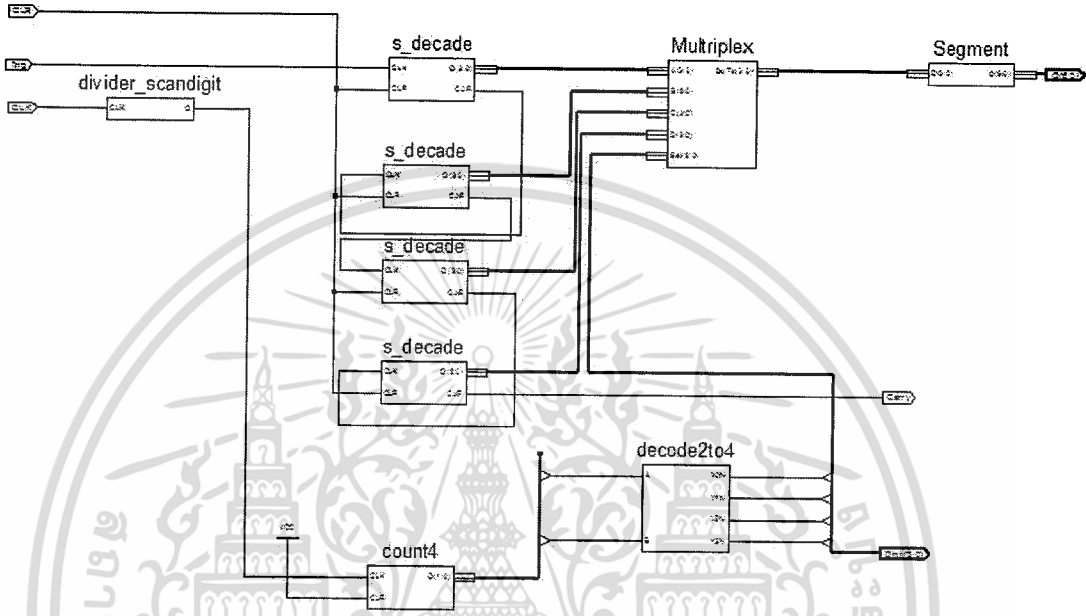


รูปที่ 5.39 การใช้งานของวงจรรวมแยกและวงจรรวม Shift Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

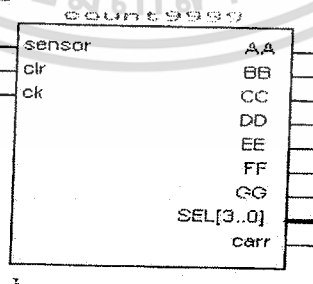
### 5.5.3) การทำงานของวงจรรนับ (COUNTER)

วงจรรนับที่เราใช้นั้นเป็นวงจรรนับ 4 หลัก ซึ่งเราได้ทำการออกแบบไว้ใน FPGA มีวงจรรนับ ดังรูปที่ 5.40



รูปที่ 5.40 วงจรรนับ 4 หลัก

เมื่อเราออกแบบวงจรรนับเสร็จแล้วจะเห็นได้ว่าวงจรรนับมีขนาดใหญ่จึงทำให้ตอนที่เรานำไปต่อร่วมกับวงจรรทั้งหมดนั้นเกิดความยุ่งยากและเปลืองเนื้อที่ เราจึงได้ทำการออกแบบสัญลักษณ์ใน FPGA เพื่อได้ง่ายต่อการนำไปใช้งานซึ่งมีลักษณะดังนี้



รูปที่ 5.41 สัญลักษณ์ของวงจรรนับ 4 หลัก

หลักการการทำงานโดยรวมของวงจรร เริ่มจากวงจรร S\_DECAD E เป็นวงจรรที่นับที่จะนับ 0-9 และเมื่อนับถึง 9 แล้วก็นำไปทดให้กับวงจรร S\_DECAD E หลักต่อไป จนครบ 4 หลักส่วนวงจรร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MULTIPLEX จะเป็นตัวรวมข้อมูลของวงจร S\_DECADE แต่ละหลักแปลงเป็นสัญญาณข้อมูลและ ส่งไปยัง BCD TO 7-SEGMENT เพื่อส่ง ไปแสดงผลยัง 7-SEGMENT

#### 5.5.4) วงจรชุดควบคุมหัวเป่า

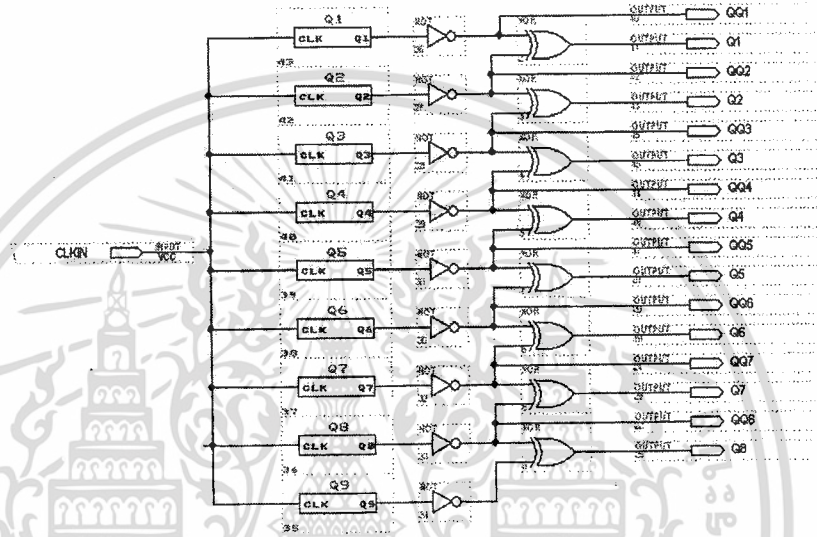
เมื่อวงจรได้รับสัญญาณที่ผ่านวงจรแยกรางก็นำเข้าวงจร Shift Register แล้วนำสัญญาณที่ได้มาเป็นสัญญาณ Trig ให้กับวงจร โมโนสเตเบิล มัลติไวเบเรเตอร์ เพื่อทำการขยายความกว้างของสัญญาณพัลส์แล้วได้สัญญาณพัลส์ที่จะไป Trig ตัว OPTO TRANSISTER (PC 817) หลังจากนั้นจะได้สัญญาณไป Drive ตัวทรานซิสเตอร์ BD 139 ซึ่งเป็นทรานซิสเตอร์ที่ใช้เป็นตัวสวิตซ์ในการตัดต่อแรงดันไฟ 24 โวลต์ ที่ใช้ป้อนให้กับชุดหัวเป่าซึ่งมีรูปวงจรทั้งหมดดังนี้



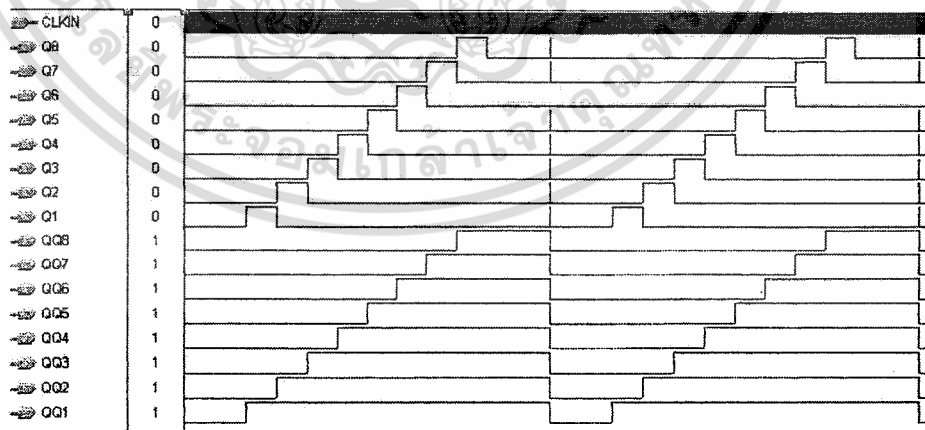
รูปที่ 5.42 วงจรควบคุมหัวเป่า

### 5.6 รูปสัญญาณเมื่อเข้าวงจรแยกแวง

เมื่อเราได้สัญญาณที่ผ่าน D- Flip-Flop แล้วนั้นก็ทำการป้อนให้กับวงจรถ่ายแวงซึ่งได้ทำการออกแบบไว้ใน FPGA



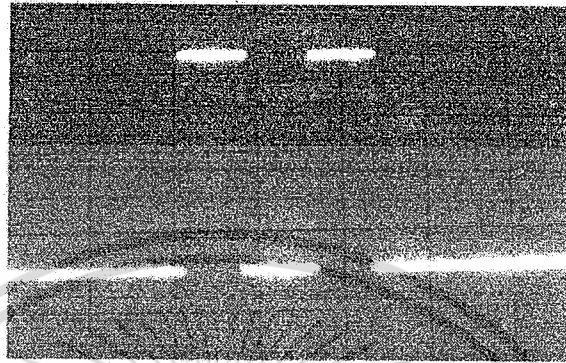
รูปที่ 5.43 วงจรถ่ายแวง



รูปที่ 5.44 สัญญาณของวงจรถ่ายแวงที่ได้จากการ Simulate ใน FPGA

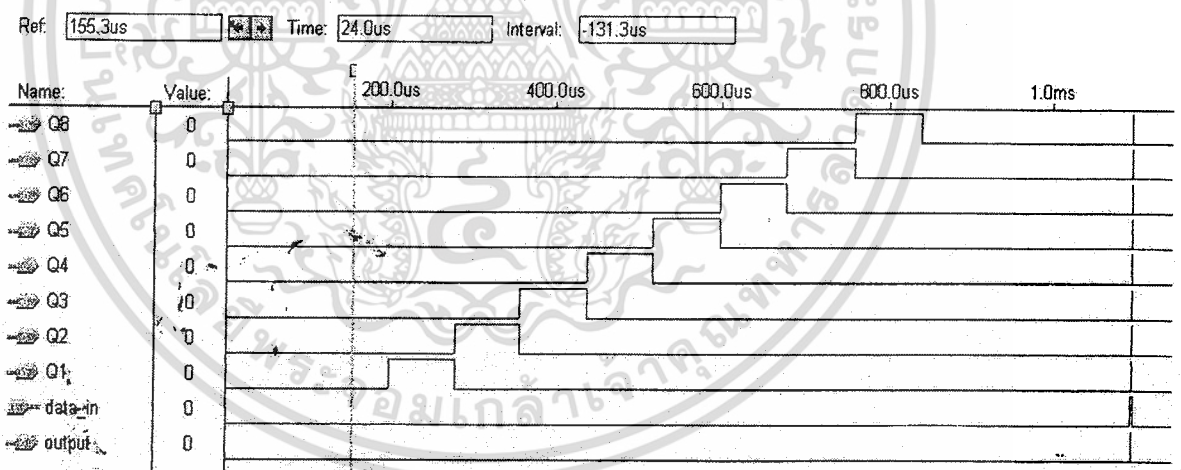
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีสัญญาณข้อมูลเข้ามาในวงจรแยกราง จากนั้นจะนำสัญญาณที่เข้ามาไปผ่าน AND GATE เพื่อทำการ AND กับสัญญาณของวงจรแยกรางด้านบนจะได้รูปสัญญาณดังนี้



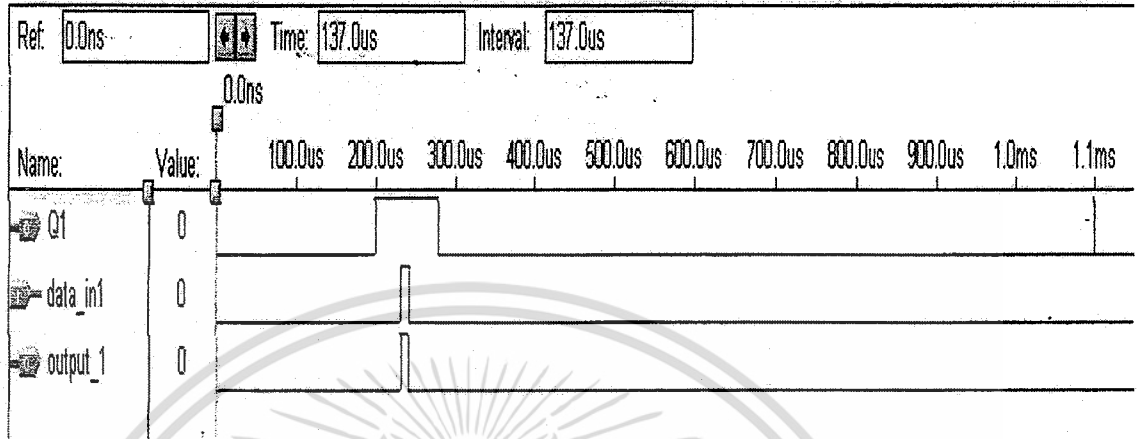
รูปที่ 5.45 รูปสัญญาณที่ผ่านวงจรแยกราง

1) รูปสัญญาณขณะที่ยังไม่มีเมตริกซ์ค่าผ่าน



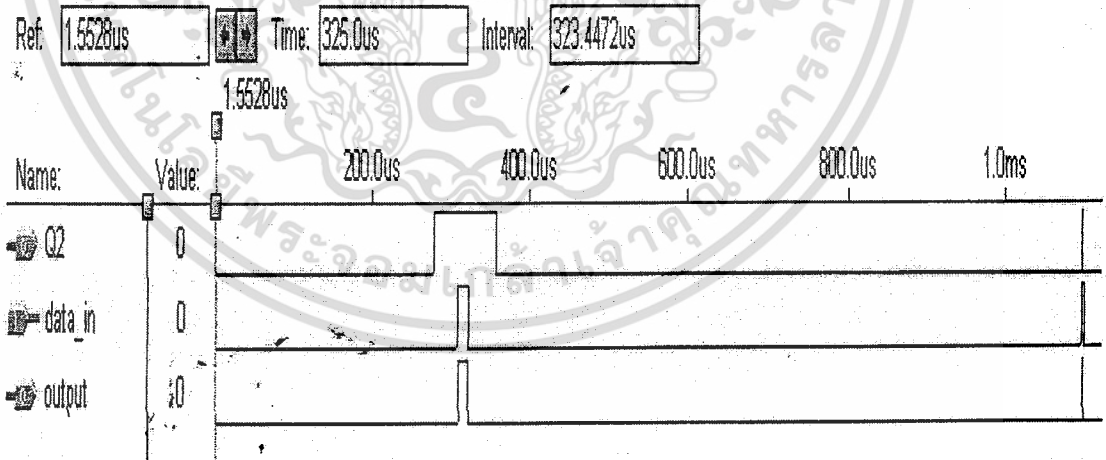
รูปที่ 5.46 รูปสัญญาณขณะที่ยังไม่มีเมตริกซ์ค่าผ่าน

2) รูปสัญญาณที่มีเมสจิดข่าวสดำผ่านรงที่1



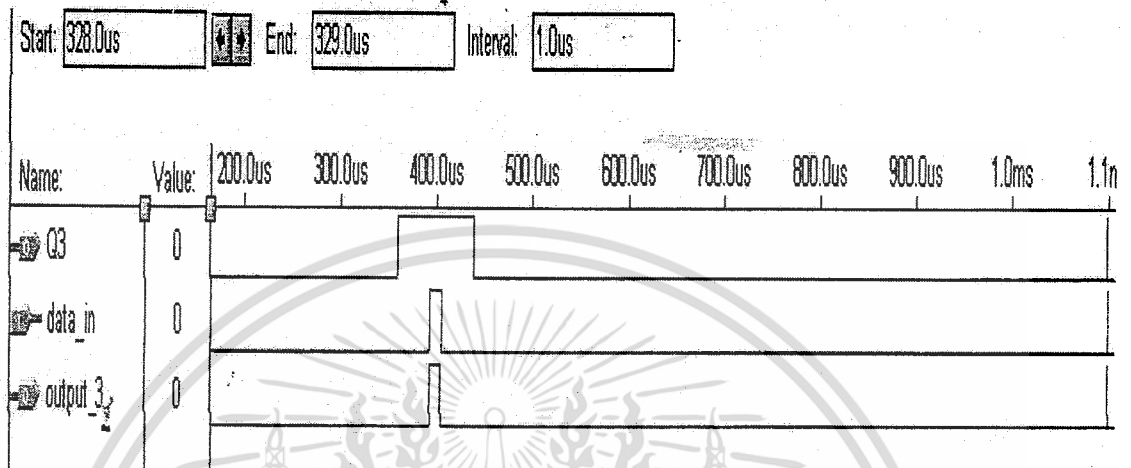
รูปที่ 5.47 รูปสัญญาณที่มีเมสจิดข่าวสดำผ่านรงที่ 1

3) รูปสัญญาณที่มีเมสจิดข่าวสดำผ่านรงที่2



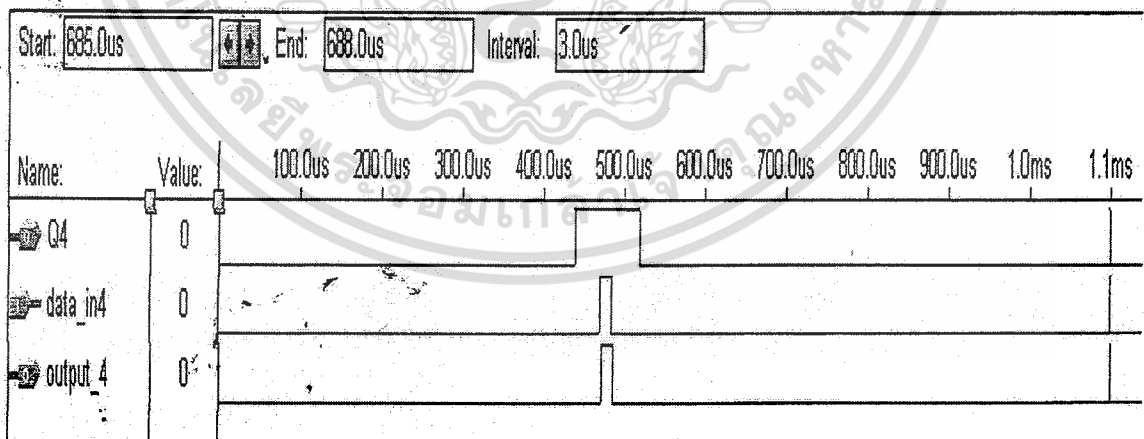
รูปที่ 5.48 รูปสัญญาณที่มีเมสจิดข่าวสดำผ่านรงที่ 2

## 4) รูปสัญญาณที่มีเมล็ดข้าวลีดำผ่านรางที่ 3



รูปที่ 5.49 รูปสัญญาณที่มีเมล็ดข้าวลีดำผ่านรางที่ 3

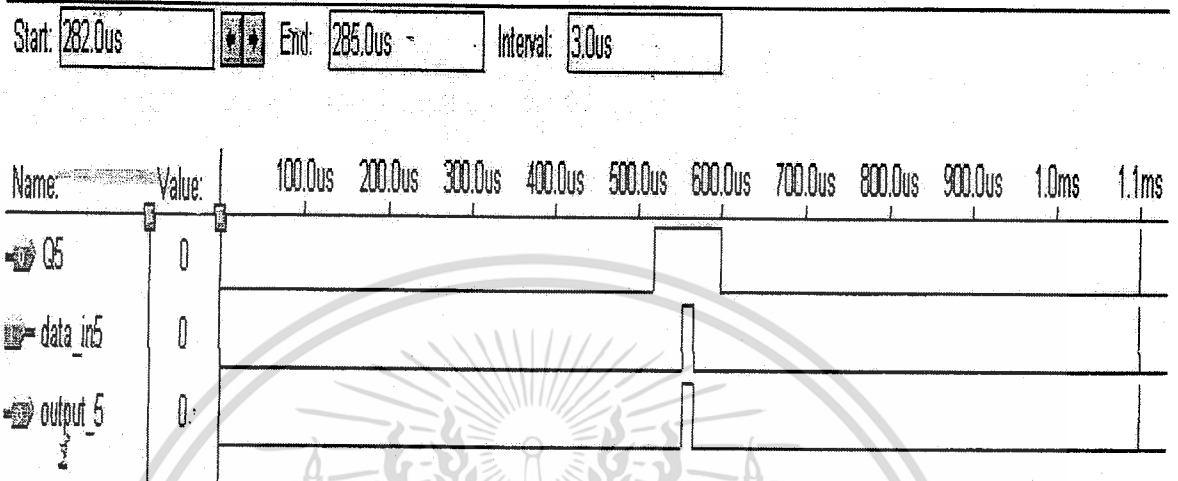
## 5) รูปสัญญาณที่มีเมล็ดข้าวลีดำผ่านรางที่ 4



รูปที่ 5.50 รูปสัญญาณที่มีเมล็ดข้าวลีดำผ่านรางที่ 4

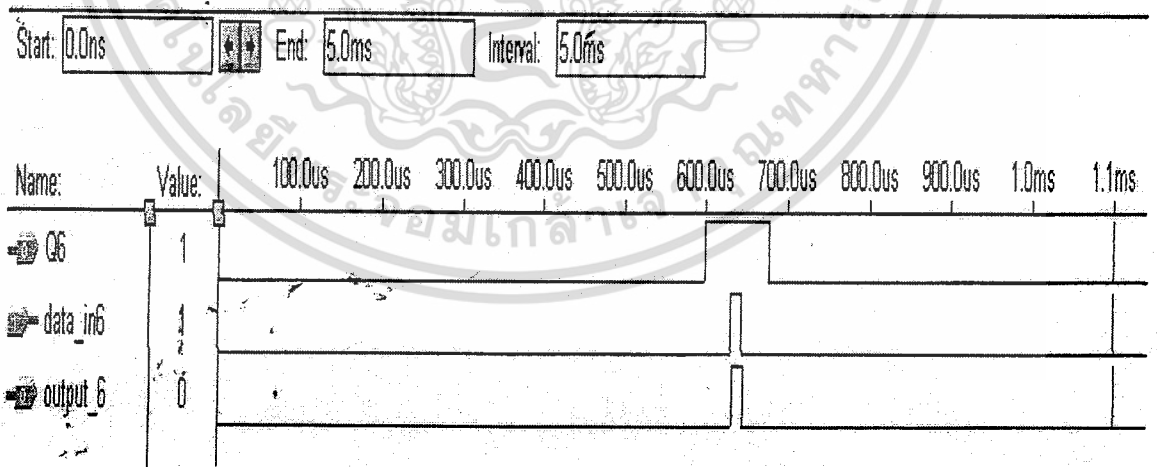
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) รูปสัญญาณที่มีเมสจิดเข้าสาค่าผ่านรางที่ 5



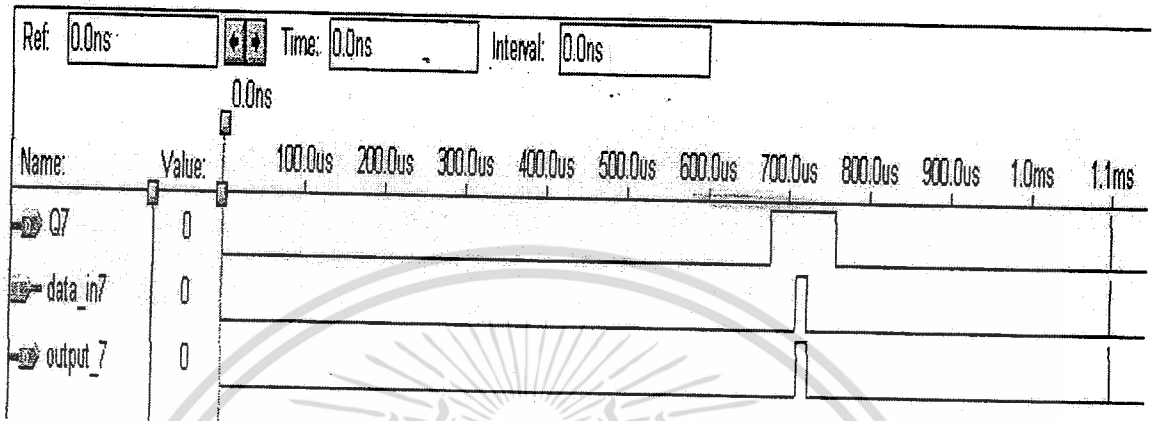
รูปที่ 5.51 รูปสัญญาณที่มีเมสจิดเข้าสาค่าผ่านรางที่ 5

7) รูปสัญญาณที่มีเมสจิดเข้าสาค่าผ่านรางที่ 6



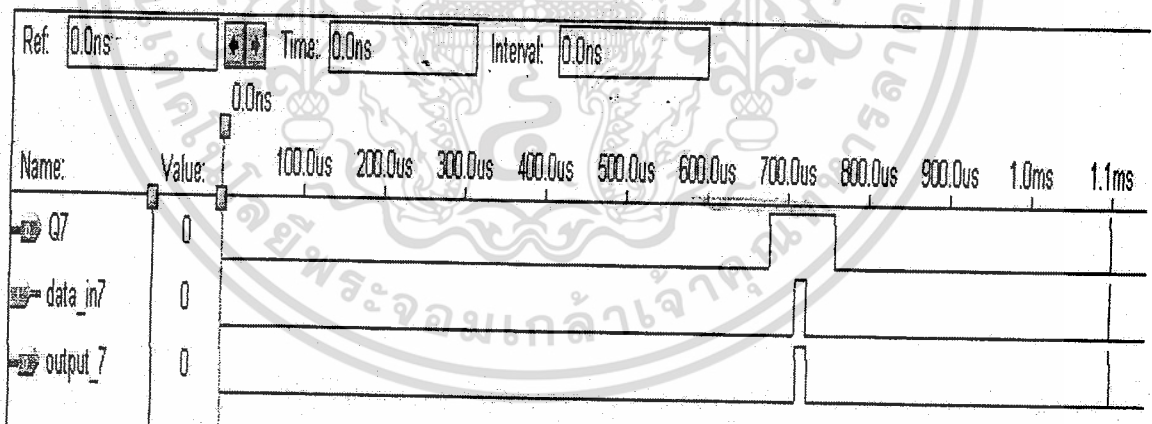
รูปที่ 5.52 รูปสัญญาณที่มีเมสจิดเข้าสาค่าผ่านรางที่ 6

## 8) รูปสัญญาณที่มีเมสจิดข่าวลิตำผ่านรางที่7



รูปที่ 5.53 รูปสัญญาณที่มีเมสจิดข่าวลิตำผ่านรางที่7

## 9) รูปสัญญาณที่มีเมสจิดข่าวลิตำผ่านรางที่ 8



รูปที่ 5.54 รูปสัญญาณที่มีเมสจิดข่าวลิตำผ่านรางที่8

## 5.7 ผลการทดลองถึงความลาดชันของรางที่องศาต่างๆ

- 1.) เตรียมผลผลิตที่ต้องการเป็นวัตถุนาขนาดเล็กสีขาว เช่น ข้าว และของเสียที่ไม่ต้องการเป็น วัตถุสีดำ จำนวน 50 เมล็ด
- 2.) ปล่อยผลผลิตให้ไหลตามรางที่ 3 ที่ความลาดชัน 30 องศา
- 3.) ทำการตรวจจับวัตถุสีดำที่ไม่ต้องการ โดยแสดงผลออกทาง 7 – Segment แล้วทำการ บันทึกผลลงในตารางที่ 5.1
- 4.) ทำการทดลองตามข้อ 2. และข้อ 3. อีกครั้งแต่ทำการเปลี่ยนความลาดชันเป็น 45 องศา และทำการบันทึกผลลงในตารางที่ 5.1
- 5.) ทดสอบปล่อยข้าวในรางที่ 4, 5 ที่ความชัน 30 องศา แล้วทำตามข้อที่ 3,4 จากนั้นบันทึกผลลงในตารางที่ 5.2 และ 5.3 ตามลำดับ

### ผลการทดลอง

ตารางที่ 5.1 จำนวนของเสียที่นับได้ของรางที่ 3 ในลักษณะต่างๆ

ครั้งที่	ความชันของราง(องศา)	
	30 องศา	45 องศา
1	35 เมล็ด	31 เมล็ด
2	34 เมล็ด	30 เมล็ด
3	35 เมล็ด	30 เมล็ด
เฉลี่ย	35 เมล็ด	30 เมล็ด

ตารางที่ 5.2 จำนวนของเสียที่นับได้ของรางที่ 4 ในลักษณะต่างๆ

ครั้งที่	ความชื้นของราง(องศา)	
	30 องศา	45 องศา
1	36 เมล็ด	33 เมล็ด
2	36 เมล็ด	32 เมล็ด
3	36 เมล็ด	32 เมล็ด
เฉลี่ย	36 เมล็ด	32 เมล็ด

ตารางที่ 5.3 จำนวนของเสียที่นับได้ของรางที่ 5 ในลักษณะต่างๆ

ครั้งที่	ความชื้นของราง(องศา)	
	30 องศา	45 องศา
1	37 เมล็ด	34 เมล็ด
2	38 เมล็ด	34 เมล็ด
3	37 เมล็ด	34 เมล็ด
เฉลี่ย	37 เมล็ด	34 เมล็ด

### 5.8 ผลการทดลองการคัดแยกเมล็ดข้าวที่ไม่ต้องการโดยใช้ชุดหัวเป่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำสัญญาณที่ได้จากวงจร Shift Register แยกออกเป็น 2 ทางซึ่งทางหนึ่งไปเข้าวงจรนับ และอีกทางหนึ่งนำสัญญาณไป Trig ให้อุปกรณ์ โมโนสเตเบิลเพื่อทำการควบคุมชุดหัวเป่าที่จะทำการแยกสิ่งแปลกปลอม (วัตถุสีเข้ม)

จากการที่เราทำการนับของเสียในรางต่างๆแล้วนั้น จากนั้นชุดหัวเป่าก็จะทำการคัดแยกของเสียนั้นออกไป

ผลการทดลอง(จำนวนที่ใช้ในการทดสอบเมล็ดข้าวจำนวน 50 เมล็ด)

ตารางที่ 5.4 จำนวนของเสียที่ชุดหัวเป่าคัดแยกออกมาได้ในรางที่ 3 ที่องศาต่างๆ

ครั้งที่	ความชื้นของราง(องศา)	
	30 องศา	45 องศา
1	32 เมล็ด	29 เมล็ด
2	32 เมล็ด	30 เมล็ด
3	33 เมล็ด	30 เมล็ด
เฉลี่ย	32 เมล็ด	30 เมล็ด

ตารางที่ 5.5 จำนวนของเสียที่ชุดหัวเป่าคัดแยกออกมาได้ในรางที่ 4 ที่องศาต่างๆ

ครั้งที่	ความชื้นของราง(องศา)	
	30 องศา	45 องศา
1	30 เมล็ด	29 เมล็ด
2	30 เมล็ด	28 เมล็ด
3	29 เมล็ด	29 เมล็ด
เฉลี่ย	30 เมล็ด	29 เมล็ด

ตารางที่ 5.6 จำนวนของเสียที่ชุดหัวเป่าคัดแยกออกมาได้ในรางที่ 5 ที่องศาต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่	ความชันของราง(องศา)	
	30 องศา	45 องศา
1	28 เมตต์	27 เมตต์
2	27 เมตต์	26 เมตต์
3	28 เมตต์	27 เมตต์
เฉลี่ย	28 เมตต์	27 เมตต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### สรุปและวิจารณ์ผลการทดลอง

ในการทำเครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าวโดยการประมวลภาพนั้น เริ่มต้นด้วยการศึกษาคุณสมบัติของ CCD ซึ่งจะต้องหา CCD ให้ได้คุณสมบัติตามที่ต้องการ และเริ่มต้นศึกษาการเขียนโปรแกรมด้วยภาษา VHDL โดยการใช้งานโปรแกรม Xilinx-ISE เพื่อที่จะได้นำมาเป็นเครื่องมือในการออกแบบวงจรที่อยู่ใน FPGA เพื่อที่จะให้เกิดความสะดวกและถ้ามีการออกแบบที่ผิดพลาดก็สามารถที่จะแก้ไขได้สะดวก จากนั้นเริ่มออกแบบภาค Drive CCD เพื่อสร้างสัญญาณ ROG, P1, P2, RS, CLP ซึ่งเป็นสัญญาณที่ใช้ในการควบคุมการทำงานของ CCD และเมื่อ CCD ได้รับสัญญาณจากภาค Drive CCD แล้วเราก็จะทำการวัดสัญญาณ OUTPUT ที่ได้จาก CCD โดยจะใช้พื้นกระดาษสีขาวเป็นฉากรับแสงเพื่อสะท้อนกลับให้ CCD และเราจะใช้วัสดุสีดำเพื่อทดลองในการตรวจจับภาพ

ในขณะที่วัตถุไม่มีวัสดุสีดำผ่านสัญญาณที่ได้จาก CCD จะมีแรงดันที่คงที่อยู่ระดับหนึ่ง และเมื่อวัตถุสีดำผ่านระดับแรงดันก็จะมีค่าเปลี่ยนแปลงไปในทางที่สูงขึ้นกว่าระดับแรงดันที่ยังไม่มีวัตถุสีดำผ่าน ก็หมายถึงตัว CCD นั้นสามารถตรวจจับวัตถุและแยกสีของวัตถุได้ คือถ้าวัตถุสีอ่อนใกล้เคียงระดับสีขาวระดับค่าแรงดันก็จะมีเปลี่ยนแปลงอย่างเห็นได้ชัดเจน เมื่อได้สัญญาณก็นำมาผ่านชิป ADC เพื่อทำการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล และนำสัญญาณไปเข้า FPGA ที่ทำการออกแบบวงจรแยกแยะไว้ เพื่อทำการตรวจดูว่าวัตถุที่มีสีเข้มนั้นอยู่ในรางใด และนำสัญญาณไปควบคุมวงจรควบคุมหัวเป่าเพื่อเป่าเมล็ดข้าวที่อยู่รางนั้น และนำสัญญาณไปเข้าวงจร Counter เพื่อนับจำนวนเมล็ดข้าวที่เป่าออกแสดงทาง 7 Segment

ในการทดสอบเครื่องตรวจสอบสิ่งแปลกปลอมจากเมล็ดข้าว ไม่สามารถที่จะคัดแยกเมล็ดข้าวที่มีสีดำได้ทุกเมล็ดที่ผ่านตัว CCD เนื่องจากความเร็วของตัว CCD นั้นยังไม่สามารถตรวจจับวัตถุได้เร็วพอ และสัญญาณที่ออกมาจากตัว CCD นั้นมีสัญญาณรบกวนสูงจึงส่งผลให้การแปลงสัญญาณจากอนาล็อกเป็นดิจิทัลนั้นไม่มีความถูกต้อง ทำให้ผลที่ได้ออกมาไม่มีความผิดพลาดเกิดขึ้น วิธีแก้คือหาตัว CCD ที่มีความเร็วในการสแกนสูง และหาวิธีกำจัดสัญญาณรบกวนให้มีค่าน้อยที่สุด

## บรรณานุกรม

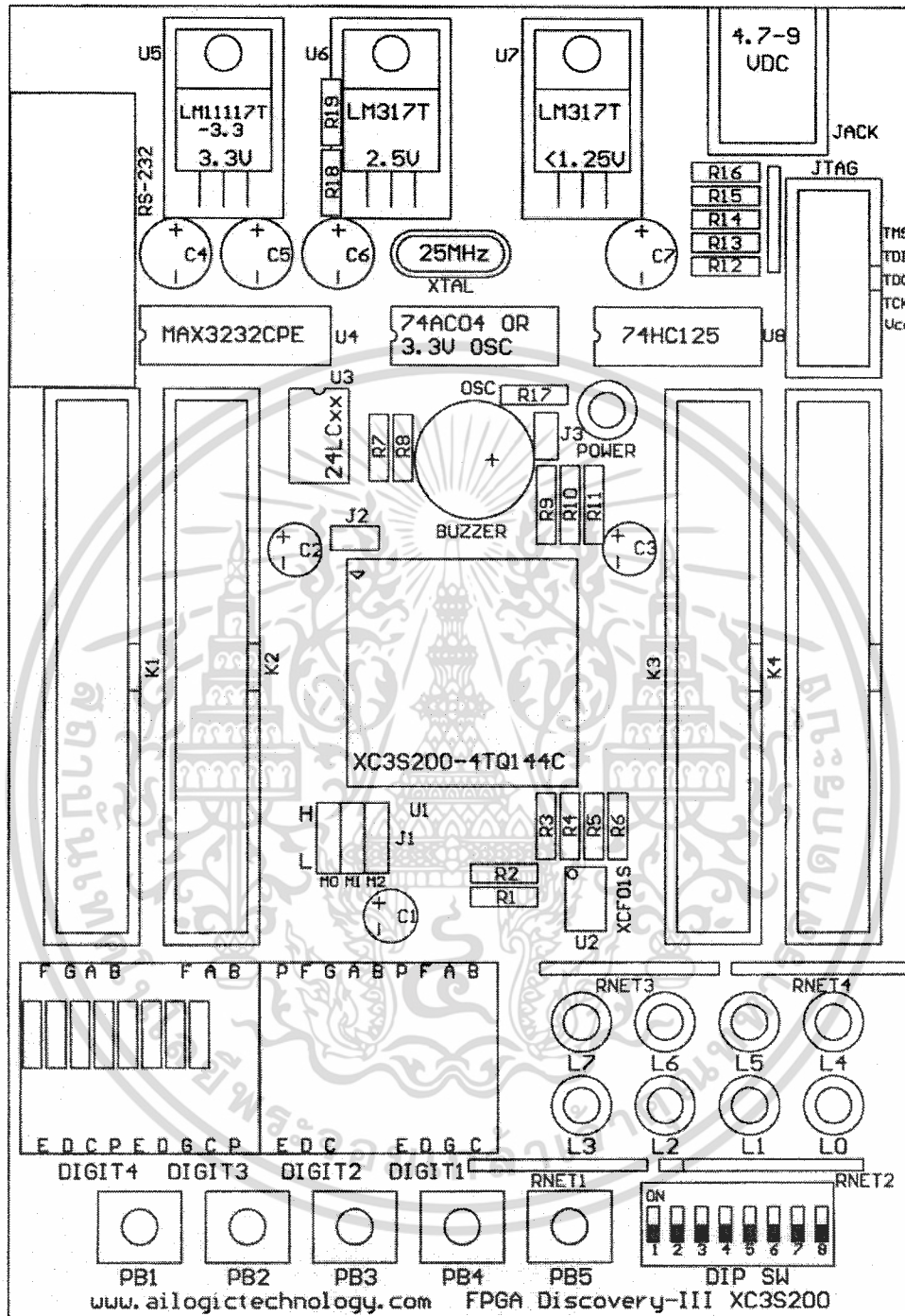
- [1] ชำนาญ ปัญญาใส และ วิศวกร หนูทอง . 2547. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
- [2] รัชชยัต พึ่งแสงโชติช่วง และคณะ. Digital System Design การออกแบบระบบดิจิทัล มหาวิทยาลัยเทคโนโลยีมหานคร
- [3] นอ.ชาติชาย ดิษฐกุล. เอกสารประกอบการเรียน ภาษา VHDL. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [4] [www.astronlogic.com](http://www.astronlogic.com)



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## FPGA Discovery-III XC3S200 Board Manual



### 1. Connector and Jumper

#### 1.1 Expansion connector (K1 - K4)

เป็นหัวต่อที่ใช้เชื่อมสัญญาณ I/O จาก FPGA ไปยังบอร์ดหรืออุปกรณ์ภายนอกที่มี I/O เป็น 3.3V โดยจะต่ออยู่กับขา CPLD ดังตารางด้านล่าง ในกรณีที่ I/O ของ FPGA เป็น Output สามารถต่อออกจากบอร์ดไปจับ Input เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของอุปกรณ์ที่เป็นระบบ 3.3V และ 5V ได้โดยตรง แต่ถ้า I/O ของ FPGA เป็น Input นั้นจะรับได้เฉพาะ Input ที่เป็นระบบ 3.3V เท่านั้น ( ถ้ารับมาจากระบบ 2.5V ต้องใช้ตัวความต้านทาน ( R ) มา Pull up ) แต่ถ้า Input เป็นระบบ 5V จะต้องใช้บัฟเฟอร์ที่เป็นระบบ 3.3V มากั้นเพื่อป้องกันไม่ให้ I/O ของ FPGA ได้รับความเสียหาย บัฟเฟอร์ที่เป็นระบบ 3.3V อาจใช้ไอซีตระกูล 74HCxx หรือ 74ACxx ( เร็วกว่า ) มากินไฟเลี้ยง (Vcc) 3.3VDC และต้องต่อความต้านทาน 200 โอห์มที่ Input ของบัฟเฟอร์ทุกตัว ( เพื่อจำกัดกระแสไม่ให้ Input ของบัฟเฟอร์เสียหาย คือ <math>< 10 \text{ mA}</math> )

## 1.2 JTAG connector

เป็นหัวต่อที่ใช้ต่อกับสายที่ใช้โปรแกรมข้อมูลลงตัว FPGA และ PROM โดยผ่านทาง JTAG Cable

## 1.3 J1

จัมเปอร์ J1 ประกอบด้วย M0 , M1 , M2 โดยปกติให้เซทไว้ที่ลอจิก “ L “ หรืออยู่ในโหมด Master serial เนื่องจากเราสามารถโปรแกรม PROM หรือ FPGA โดยใช้สาย JTAG ได้อยู่แล้วโดยไม่ต้องสนใจตำแหน่งของจัมเปอร์แต่อย่างใด

## 1.4 J2

จัมเปอร์ที่ใช้ควบคุมให้ FPGA ทำการ Pull up I/O ของ FPGA ทุกขาเมื่อใส่จัมเปอร์ และจะเป็น Hi Impedance เมื่อถอดจัมเปอร์ออก

## 1.5 J3

จัมเปอร์ที่ใช้ตัด BUZZER ออกจาก FPGA เมื่อถอดจัมเปอร์ออก

## 2. Input

### 2.1 DIP switch ( DIP SW )

เป็นชุดของสวิตช์เลื่อนขนาดเล็กที่ใช้ป้อนข้อมูลเข้าสู่ FPGA โดยถ้าเลื่อนลง ( Off ) จะเป็น “1”



ถ้าเลื่อนขึ้น ( On ) จะเป็น “0” โดยเชื่อมต่อกับขาของ FPGA ดังตารางด้านท้าย DIP SW ทุกตัวจึงทำงานแบบ Active Low



### 2.2 Push button switch ( PB1 – PB5 )

เป็นสวิตช์กดติดปล่อยดับที่ให้สัญญาณเข้าที่พุทเป็นระดับลอจิก “0” เมื่อกดสวิตช์ และเป็นระดับลอจิก “1” เมื่อปล่อยสวิตช์ โดยจะต่ออยู่กับขา FPGA ดังตารางด้านท้าย Push button switch ทุกตัวจึงทำงานแบบ Active Low

### 2.3 Changeable oscillator ( OSC )

เป็นตัวกำเนิดสัญญาณนาฬิกาที่สามารถเปลี่ยนค่าความถี่ที่ต้องการได้ โดยการถอดเปลี่ยน ออสซิลเลเตอร์เดิม ( 3.3V ) ที่ให้มาบนบอร์ดออก แล้วใส่ตัวใหม่ ( 3.3V ) เข้าไปแทนที่ที่ซอกเก็ตไอซี เบอร์ 74AC04 โดยที่เอาท์พุทของ OSC จะต่ออยู่กับขา FPGA ดังตารางด้านท้าย ซึ่งเป็นขา Global clock เหมาะสำหรับวงจรที่ต้องการความถี่ในการทำงานสูงๆ

### 3. Output

#### 3.1 7-Segment ( DIGIT1 – DIGIT4 )

เป็นตัวแสดงผลเจ็ดส่วนจำนวน 4 หลักที่สามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับตัวแสดงผลเจ็ดส่วนทั้ง 4 หลัก) โดยเรียงจากซ้ายไปขวาคือ DIGIT4, DIGIT3, DIGIT2 และ DIGIT1 โดยตัวที่ 2 และ 1 จะทำการกลับตัวแสดงผลเจ็ดส่วนให้เพื่อใช้จุด (Dot) ในการทำนาฬิกาหรือ แสดงองศาในการวัดอุณหภูมิ เช่น 11:39 หรือ 20°

ตัวแสดงผลเจ็ดส่วนทั้งหมดจะต่อขา cathode เข้าด้วยกัน โดยมีขาไฟร่วม ( Common cathode ) แยกกันสี่ขา ดังนั้นผู้ใช้จึงจำเป็นต้องใช้เทคนิคในการสแกน ( Scan ) เพื่อให้ตัวแสดงผลทั้งเจ็ดส่วนสามารถแสดงผลพร้อมกันได้ทั้งหมด และตัวแสดงผลทั้งหมดเป็นแบบไฟร่วม ( Common cathode ) โดยจะต่ออยู่กับขา FPGA ดังตารางด้านล่าง

#### 3.2 LED แสดงผล

LED แสดงผล L0 – L7 จะต่อแชร์กับ I/O ของ Connector K3 และ K4 โดยที่ L2 , L3 , L6 และ L7 จะต่อแชร์กับ I/O ของ Connector K3 โดยมี Resistor “RNET3” 8P4R 470 Ohm จำกัดกระแส และ L0 , L1 , L4 และ L5 จะต่อแชร์กับ I/O ของ Connector K4 โดยมี Resistor “RNET3” 8P4R 470 Ohm จำกัดกระแส

#### 3.3 Buzzer

เป็นออกความถี่เสียง ( Buzzer ) โดยที่จะมีเสียงดังเมื่อป้อนสัญญาณเป็น High “1” โดยจะต่ออยู่กับขา FPGA ดังตารางด้านล่าง กรณีที่ต้องการใช้ I/O ของ Connector K4 ที่แชร์อยู่กับออกหรือไม่ต้องการใช้ออกให้ถอดจัมเปอร์ J3 ออก

### 4. Misc

#### 4.1 Jack สำหรับ DC Adaptor

เป็นหัวต่อไฟเลี้ยงเพื่อป้อนให้แก่บอร์ดในการทำงาน ต่ออยู่กับแอดปเตอร์ที่มีไฟออกมาเป็น 4.7V – 9V โดยมีขั้วด้านในเป็น บวก “+” ด้านนอกเป็น ลบ “-”

#### 4.2 Power LED ( POWER )

เป็นไดโอดเปล่งแสงว่าในขณะที่นั้นๆ มีไฟเลี้ยงบอร์ดอยู่หรือไม่

#### 4.3 RS-232C Port

เป็นพอร์ต RS-232C ซึ่งหากไม่ต้องการใช้พอร์ต RS-232C แต่ต้องการใช้เป็น I/O ที่ Connector K1 และ K2 ( ส่วนที่แชร์ I/O อยู่กับพอร์ต RS-232C ) ให้ถอดไอซี MAX3232CPE ออกจาก Socket

**หมายเหตุ** ขา 15 ของไอซี MAX3232CPE ต้องบัดกรีลงกราวนด์ด้วย

#### 4.4 Platform Flash PROM เบอร์ XCF01S

เป็น Serial PROM ที่สามารถโปรแกรมได้โดยตรงผ่านทางสาย JTAG สามารถโปรแกรมซ้ำได้ประมาณ 20,000 ครั้ง

#### 4.5 I2C Socket

เป็น Socket สำหรับใส่ไอซีแบบ I2C Serial EEPROM เบอร์ 24LCxx เช่น 24LC256 เป็นต้น ซึ่งสามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับ I2C Serial EEPROM ) I/O 2 ขานี้จะมี Pull up resistor 4.7kOhm ต่ออยู่

#### 5. ตาราง I/O ของ FPGA

ดูจากไฟล์ K1 , K2 , K3 , K4 และ 7-Segment การดูขาของ Connector K1-K4 ให้สังเกตด้านรอยบาก ลึกรูปสี่เหลี่ยมจะเป็นแถวของขาที่เป็นเลขที่ทั้งหมดเริ่มจาก 1 ถึง 39 โดยที่ด้านซ้ายสุดเป็นขา 1 ซึ่งจะมี Mark รูปสามเหลี่ยมปรากฏอยู่ด้านข้าง

**หมายเหตุ** ข้อมูล UPDATE ต่างๆ สอบถามหรือดูรายละเอียดได้ที่บริษัท เอเพก อินสตรูเมนต์ จำกัด 77/9 ซอยลาดพร้าว1 แขวงลาดยาว เขตจตุจักร กทม 10900 โทร. 0 – 293 – 2084 หรือ 0 – 1833 – 7279 หรือดาวน์โหลดได้ที่ <http://www.aiogictechnology.com>

7-Segment	FPGA Pinout	Descriptions
a	p40	a
b	p35	b
c	p32	c
d	p30	d
e	p27	e
f	p25	f
g	p23	g
dp	p20	Decimal Point
DG1	p31	DIGIT1 , COMMON CATHODE
DG2	p33	DIGIT2 , COMMON CATHODE
DG3	p36	DIGIT3 , COMMON CATHODE
DG4	p41	DIGIT4 , COMMON CATHODE

Push Botton	FPGA Pinout	Descriptions
PB1	p44	Push Botton No. 1
PB2	p46	Push Botton No. 2
PB3	p47	Push Botton No. 3
PB4	p50	Push Botton No. 4
PB5	p51	Push Botton No. 5

EEPROM	FPGA Pinout	Descriptions
I2C-SCL	p128	24LCXX
I2C-SDA	p129	24LCXX

RS-232	FPGA Pinout	Descriptions
TX	p131	MAX3232CPE
RX	p132	MAX3232CPE

LED	FPGA Pinout	Descriptions
L0	p70	L0
L1	p77	L1
L2	p69	L2
L3	p76	L3
L4	p74	L4
L5	p79	L5
L6	p73	L6
L7	p78	L7

Dip SW	FPGA Pinout	Description
1	p52	Dip Switch No.1
2	p53	Dip Switch No.2
3	p55	Dip Switch No.3
4	p56	Dip Switch No.4
5	p59	Dip Switch No.5
6	p60	Dip Switch No.6
7	p63	Dip Switch No.7
8	p68	Dip Switch No.8

Oscillator	FPGA Pinout	Descriptions
OSC	p127	25MHz , GCLK6

BUZZER	FPGA Pinout	Descriptions
BUZZER	p124	BUZZER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K1 Pinout	FPGA Pinout	Descriptions
1	p40	I/O , a
2		3.3 V
3	p35	I/O , b
4		GND
5	p32	I/O , c
6		GND
7	p30	I/O , d
8		GND
9	p27	I/O , e
10		GND
11	p25	I/O , f
12		GND
13	p23	I/O , g
14		GND
15	p20	I/O , dp
16		GND
17	p17	I/O
18		GND
19	p14	I/O
20		GND

k1 Pinout	FPGA Pinout	Descriptions
21	p12	I/O
22		GND
23	p10	I/O
24		GND
25	p7	I/O
26		GND
27	p5	I/O
28		GND
29	p2	I/O
30		GND
31	p141	I/O
32		GND
33	p137	I/O
34		GND
35	p132	I/O, RS-232 (RX)
36		GND
37	p130	I/O
38		GND
39	p128	I/O , GCLK7 , I2C-SCL
40		GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K2 Pinout	FPGA Pinout	Descriptions
1	p41	I/O , DIGIT4
2		3.3 V
3	p36	I/O , DIGIT3
4		GND
5	p33	I/O , DIGIT2
6		GND
7	p31	I/O , DIGIT1
8		GND
9	p28	I/O
10		GND
11	p26	I/O
12		GND
13	p24	I/O
14		GND
15	p21	I/O
16		GND
17	p18	I/O
18		GND
19	p15	I/O
20		GND

k2 Pinout	FPGA Pinout	Descriptions
21	p13	I/O
22		GND
23	p11	I/O
24		GND
25	p8	I/O
26		GND
27	p6	I/O
28		GND
29	p4	I/O
30		GND
31	p1	I/O
32		GND
33	p140	I/O
34		GND
35	p135	I/O
36		GND
37	p131	I/O , RS-232 (TX)
38		GND
39	p129	I/O , I2C-SDA
40		GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K3 Pinout	FPGA Pinout	Descriptions
1		3.3 V
2	p69	I/O , L2
3		GND
4	p73	I/O , L6
5		GND
6	p76	I/O , L3
7		GND
8	p78	I/O , L7
9		GND
10	p80	I/O
11		GND
12	p83	I/O
13		GND
14	p85	I/O
15		GND
16	p87	I/O
17		GND
18	p90	I/O
19		GND
20	p93	I/O

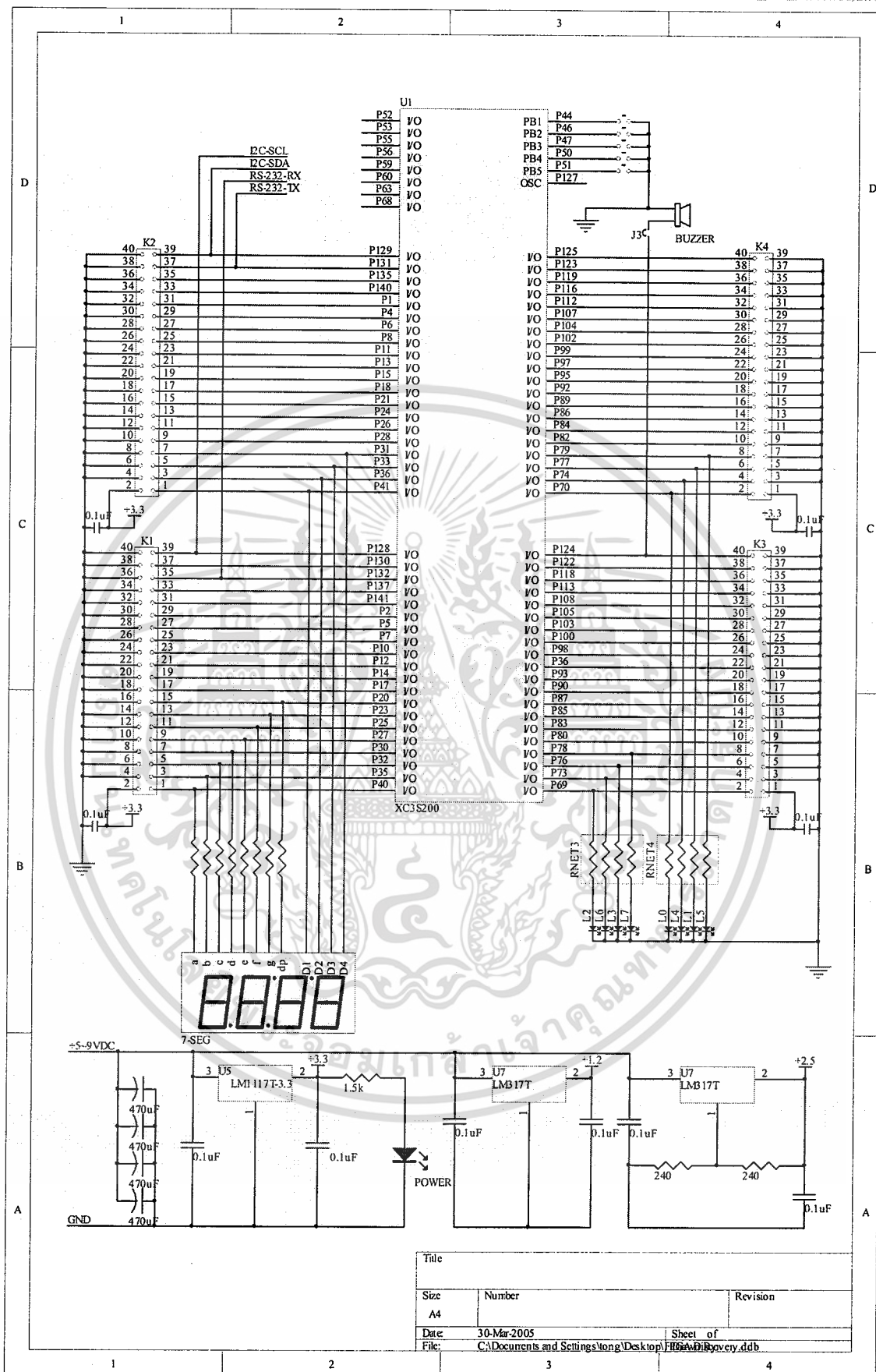
k3 Pinout	FPGA Pinout	Descriptions
21		GND
22	p96	I/O
23		GND
24	p98	I/O
25		GND
26	p100	I/O
27		GND
28	p103	I/O
29		GND
30	p105	I/O
31		GND
32	p108	I/O
33		GND
34	p113	I/O
35		GND
36	p118	I/O
37		GND
38	p122	I/O
39		GND
40	p124	I/O , GCLK4 , BUZZER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K4 Pinout	FPGA Pinout	Descriptions
1		3.3 V
2	p70	I/O , L0
3		GND
4	p74	I/O , L4
5		GND
6	p77	I/O , L1
7		GND
8	p79	I/O , L5
9		GND
10	p82	I/O
11		GND
12	p84	I/O
13		GND
14	p86	I/O
15		GND
16	p89	I/O
17		GND
18	p92	I/O
19		GND
20	p95	I/O

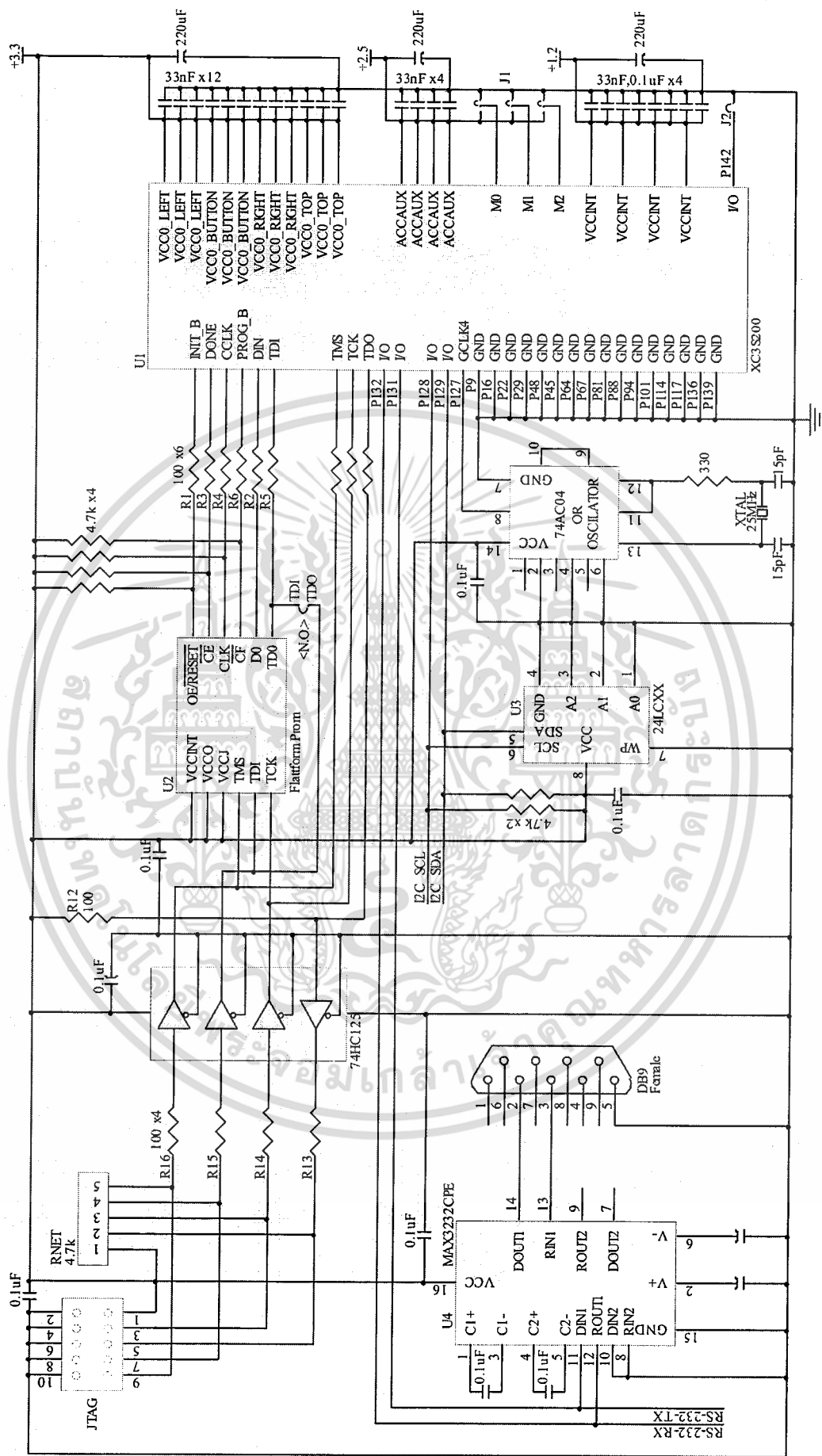
k4 Pinout	FPGA Pinout	Descriptions
21		GND
22	p97	I/O
23		GND
24	p99	I/O
25		GND
26	p102	I/O
27		GND
28	p104	I/O
29		GND
30	p107	I/O
31		GND
32	p112	I/O
33		GND
34	p116	I/O
35		GND
36	p119	I/O
37		GND
38	p123	I/O
39		GND
40	p125	I/O , GCLK5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**หมายเหตุ ขา 36 ของ FPGA ที่ต่อกับ ขา 22 ของ Connector K3 แก้เป็นขา 96**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5340 pixel × 3 line CCD Linear Sensor (Color)

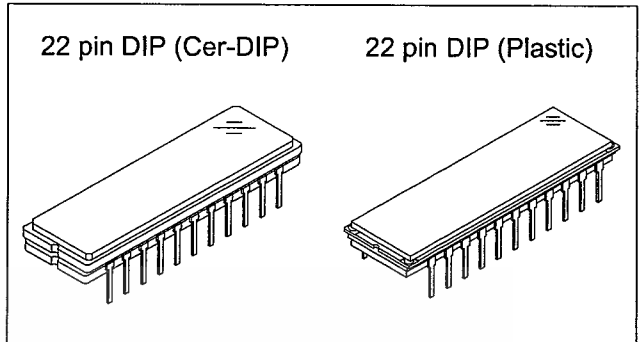
For the availability of this product, please contact the sales office.

Description

The ILX548K is a reduction type CCD linear sensor developed for color image scanner. This sensor reads A4-size documents at a density of 600DPI.

Features

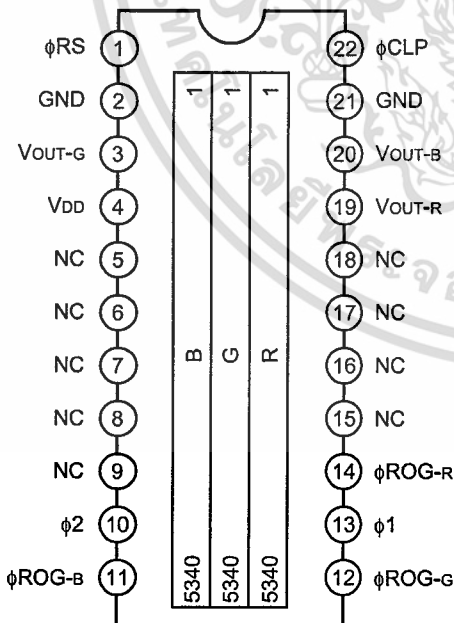
- Number of effective pixels: 16020 pixels (5340 pixels × 3)
- Pixel size: 4μm × 4μm (4μm pitch)
- Distance between line: 32μm (8 lines)
- Single-sided readout
- Ultra low lag/High sensitivity
- Single 12V power supply
- Maximum data rate: 5MHz/Color
- Input clock pulse: CMOS 5V drive
- Number of output: 3 (R, G, B)
- Package: 22 pin Cer-DIP (400mil)  
22 pin Plastic DIP (400mil)



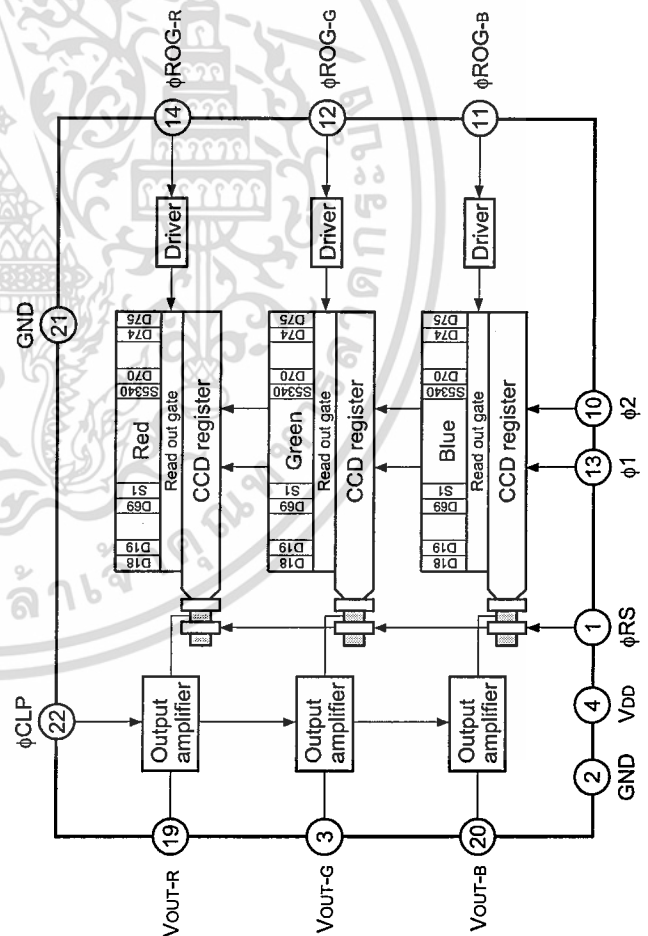
Absolute Maximum Ratings

- Supply voltage V<sub>DD</sub> 15 V
- Operating temperature -10 to +55 °C

Pin Configuration (Top View)



Block Diagram



Sony reserves the right to change products and specifications without prior notice. This information does not convey any license by any implication or otherwise under any patents or other right. Application circuits shown, if any, are typical examples illustrating the operation of the devices. Sony cannot assume responsibility for any problems arising out of the use of these circuits.

## Pin Description

Pin No.	Symbol	Description	Pin No.	Symbol	Description
1	$\phi$ RS	Clock pulse input	12	$\phi$ ROG-G	Clock pulse input
2	GND	GND	13	$\phi$ 1	Clock pulse input
3	V <sub>OUT-G</sub>	Signal output (green)	14	$\phi$ ROG-R	Clock pulse input
4	V <sub>DD</sub>	12V power supply	15	NC	NC
5	NC	NC	16	NC	NC
6	NC	NC	17	NC	NC
7	NC	NC	18	NC	NC
8	NC	NC	19	V <sub>OUT-R</sub>	Signal output (red)
9	NC	NC	20	V <sub>OUT-B</sub>	Signal output (blue)
10	$\phi$ 2	Clock pulse input	21	GND	GND
11	$\phi$ ROG-B	Clock pulse input	22	$\phi$ CLP	Clock pulse input

## Recommended Supply Voltage

Item	Min.	Typ.	Max.	Unit
V <sub>DD</sub>	11.4	12	12.6	V

## Clock Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit
Input capacity of $\phi$ 1, $\phi$ 2	C $\phi$ 1, C $\phi$ 2	—	500	—	pF
Input capacity of $\phi$ RS	C $\phi$ RS	—	10	—	pF
Input capacity of $\phi$ CLP	C $\phi$ CLP	—	10	—	pF
Input capacity of $\phi$ ROG	C $\phi$ ROG	—	10	—	pF

## Clock Frequency

Item	Symbol	Min.	Typ.	Max.	Unit
$\phi$ 1, $\phi$ 2, $\phi$ RS, $\phi$ CLP	f $\phi$ 1, f $\phi$ 2, f $\phi$ RS, f $\phi$ CLP	—	1	5	MHz

## Input Clock Pulse Voltage Condition

Item		Min.	Typ.	Max.	Unit
$\phi$ 1, $\phi$ 2, $\phi$ RS, $\phi$ CLP, $\phi$ ROG pulse voltage	High level	4.75	5.0	5.25	V
	Low level	—	0	0.1	V

**Electrooptical Characteristics (Note 1)**

(Ta = 25°C, VDD = 12V, f<sub>DRS</sub> = 1MHz, Input clock = 5Vp-p, Light source = 3200K, IR cut filter CM-500S (t = 1.0mm))

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks	
Sensitivity	Red	R <sub>R</sub>	1.8	2.7	3.6	V/(lx · s)	Note 2
	Green	R <sub>G</sub>	2.1	3.3	4.5		
	Blue	R <sub>B</sub>	1.7	2.6	3.5		
Sensitivity nonuniformity	PRNU	—	4	20	%	Note 3	
Saturation output voltage	V <sub>SAT</sub>	2	2.5	—	V	Note 4	
Saturation exposure	Red	SE <sub>R</sub>	0.56	0.93	—	lx · s	Note 5
	Green	SE <sub>G</sub>	0.44	0.76	—		
	Blue	SE <sub>B</sub>	0.57	0.96	—		
Dark voltage average	V <sub>DRK</sub>	—	2	5	mV	Note 6	
Dark signal nonuniformity	DSNU	—	4	12	mV		
Image lag	IL	—	0.02	—	%	Note 7	
Supply current	I <sub>VDD</sub>	—	25	50	mA	—	
Total transfer efficiency	TTE	92	98	—	%	—	
Output impedance	Z <sub>o</sub>	—	450	—	Ω	—	
Offset level	V <sub>OS</sub>	—	7.3	—	V	Note 8	

**Notes)**

- In accordance with the given electrooptical characteristics, the black level is defined as the average value of D18, D19 to D67.
- For the sensitivity test light is applied with a uniform intensity of illumination.
- PRNU is defined as indicated below. Ray incidence conditions are the same as for Note 2.

$$V_{OUT-G} = 500\text{mV (Typ.)}$$

$$PRNU = \frac{(V_{MAX} - V_{MIN})/2}{V_{AVE}} \times 100 [\%]$$

Where the 5340 pixels are divided into blocks of 100, the maximum output of each block is set to V<sub>MAX</sub>, the minimum output to V<sub>MIN</sub> and the average output to V<sub>AVE</sub>.

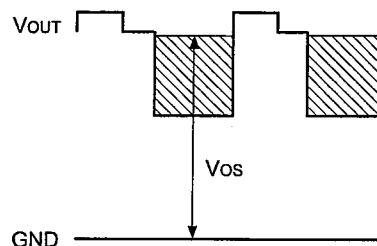
- Use below the minimum value of the saturation output voltage.
- Saturation exposure is defined as follows.

$$SE = \frac{V_{SAT}}{R}$$

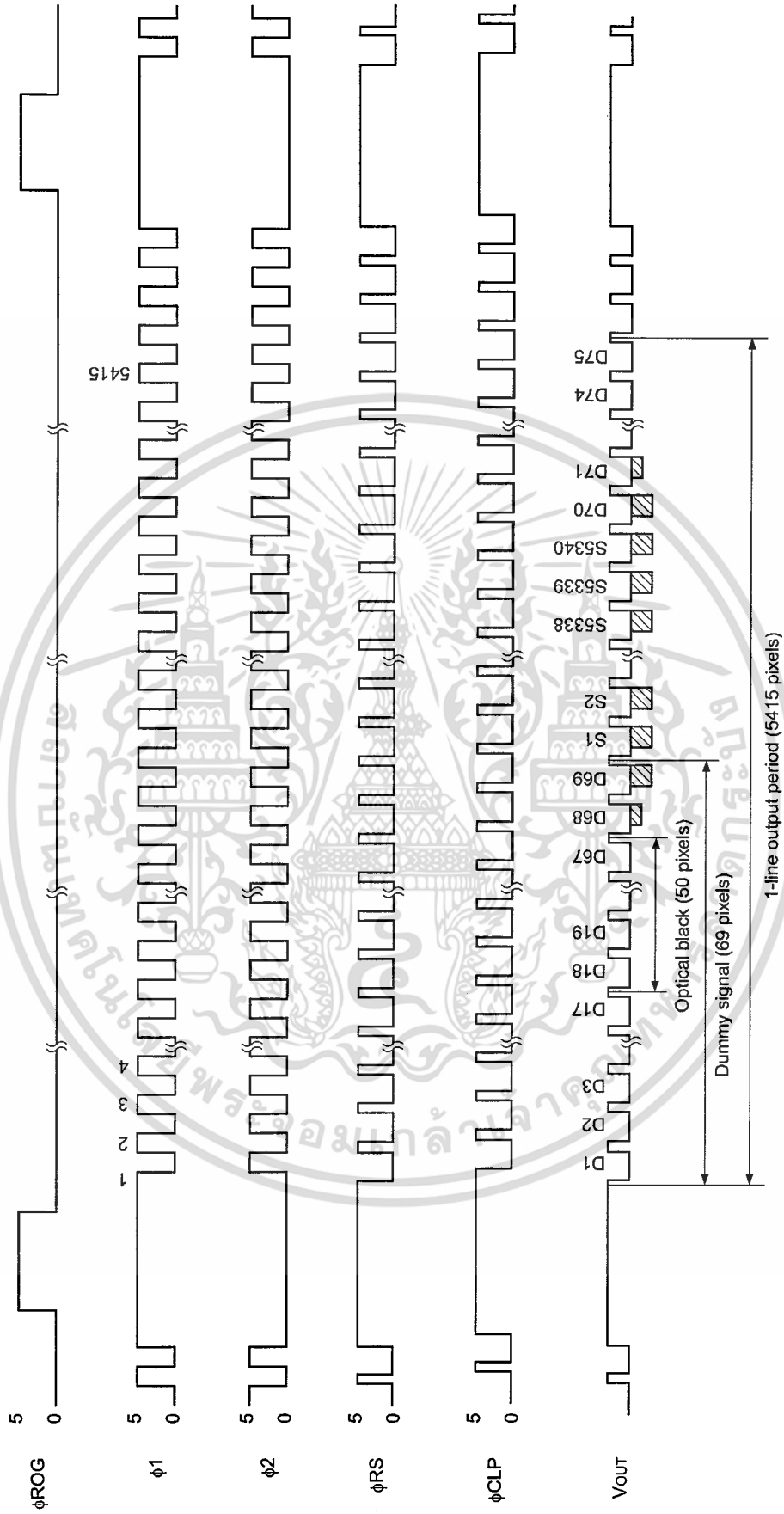
Where R indicates R<sub>R</sub>, R<sub>G</sub>, R<sub>B</sub>, and SE indicates SE<sub>R</sub>, SE<sub>G</sub>, SE<sub>B</sub>.

- Optical signal accumulated time τ<sub>int</sub> stands at 5.5ms.
- V<sub>OUT-G</sub> = 500mV (Typ.)
- V<sub>OS</sub> is defined as indicated below.

V<sub>OUT</sub> indicates V<sub>OUT-R</sub>, V<sub>OUT-G</sub> and V<sub>OUT-B</sub>.

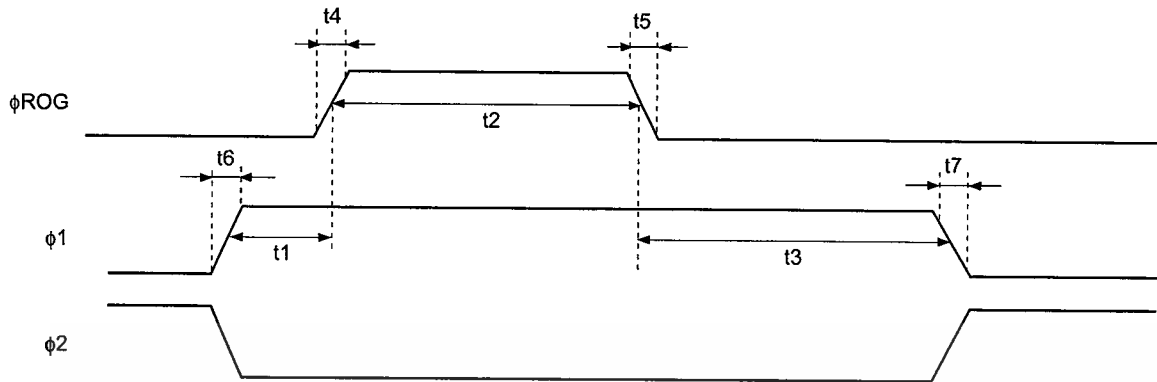


Clock Timing Chart 1

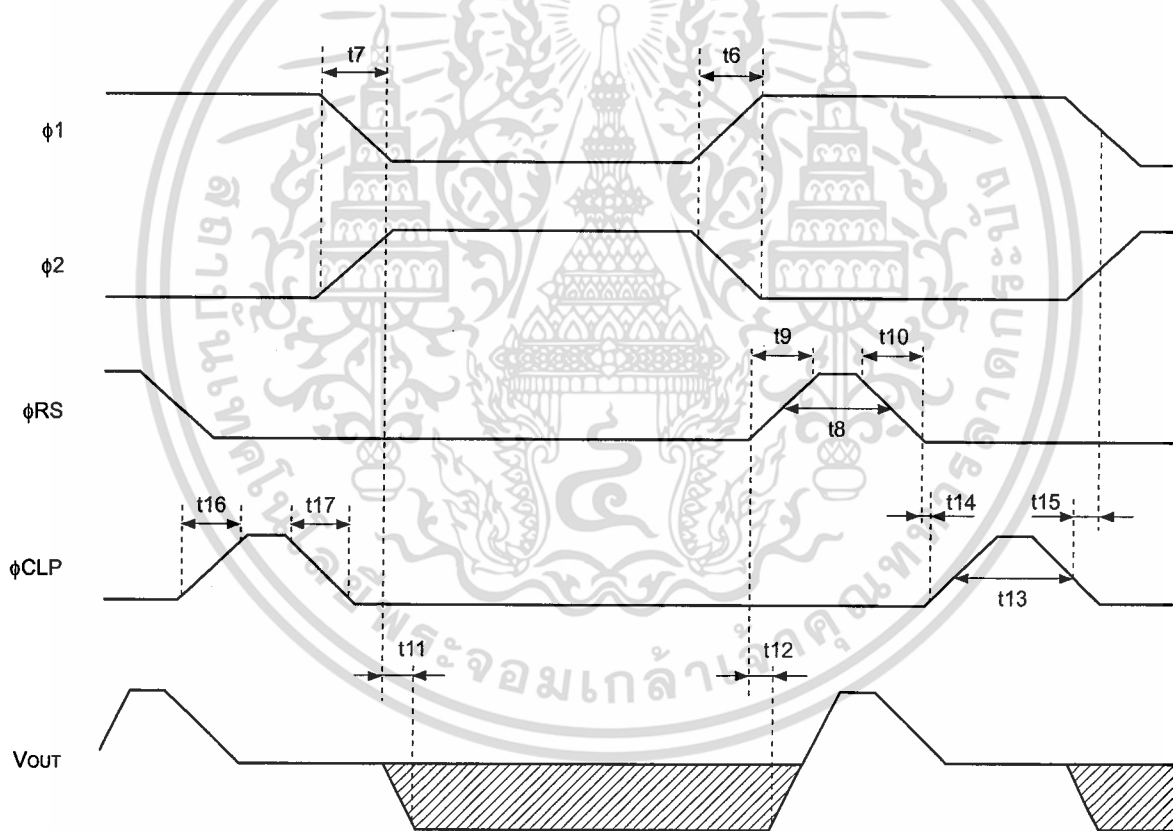


**Note)** The transfer pulses ( $\phi 1, \phi 2$ ) must have more than 5415 cycles.  
 Vout indicates Vout-R, Vout-G, Vout-B.  
 $\phi ROG$  indicates  $\phi ROG-R, \phi ROG-G, \phi ROG-B$ .

**Clock Timing Chart 2**



**Clock Timing Chart 3**

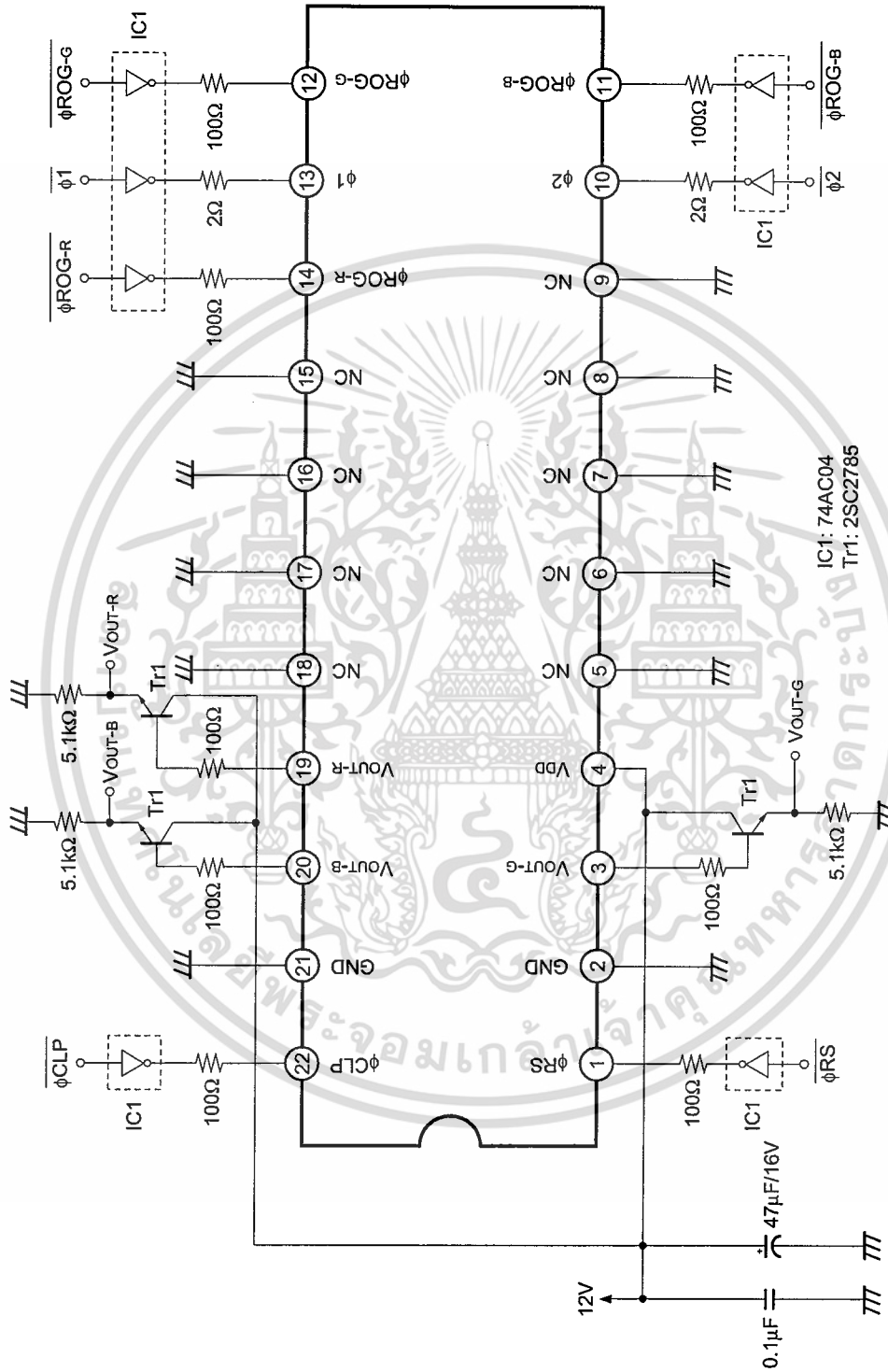


## Clock Pulse Recommended Timing

Item	Symbol	Min.	Typ.	Max.	Unit
$\phi$ ROG, $\phi$ 1 pulse timing	t1	50	100	—	ns
$\phi$ ROG pulse high level period	t2	3	5	—	$\mu$ s
$\phi$ ROG, $\phi$ 1 pulse timing	t3	1	2	—	$\mu$ s
$\phi$ ROG pulse rise time	t4	0	5	—	ns
$\phi$ ROG pulse fall time	t5	0	5	—	ns
$\phi$ 1 pulse rise time/ $\phi$ 2 pulse fall time	t6	0	20	—	ns
$\phi$ 1 pulse fall time/ $\phi$ 2 pulse rise time	t7	0	20	—	ns
$\phi$ RS pulse high level period	t8	30	50*1	—	ns
$\phi$ RS pulse rise time	t9	0	20	—	ns
$\phi$ RS pulse fall time	t10	0	20	—	ns
Signal output delay time	t11	—	50	—	ns
	t12	—	20	—	ns
$\phi$ CLP pulse high level period	t13	40	100	—	ns
$\phi$ CLP pulse timing	t14	40	100	—	ns
	t15	10	50	—	ns
	t16	0	20	—	ns
$\phi$ CLP pulse rise time	t16	0	20	—	ns
$\phi$ CLP pulse fall time	t17	0	20	—	ns

\*1 These timing is the recommended condition under  $f_{\phi RS} = 1\text{MHz}$ .

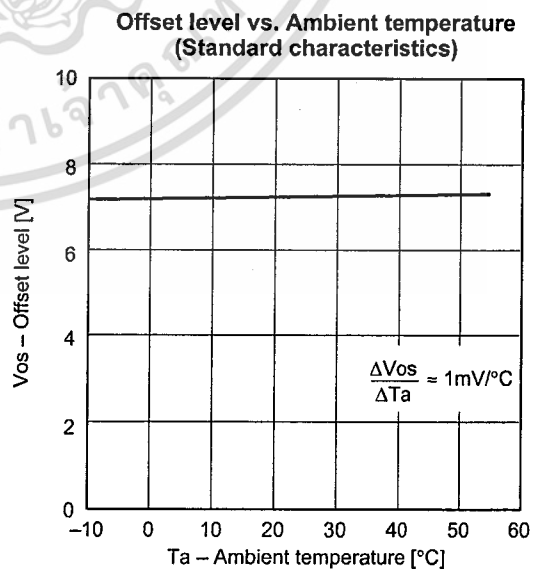
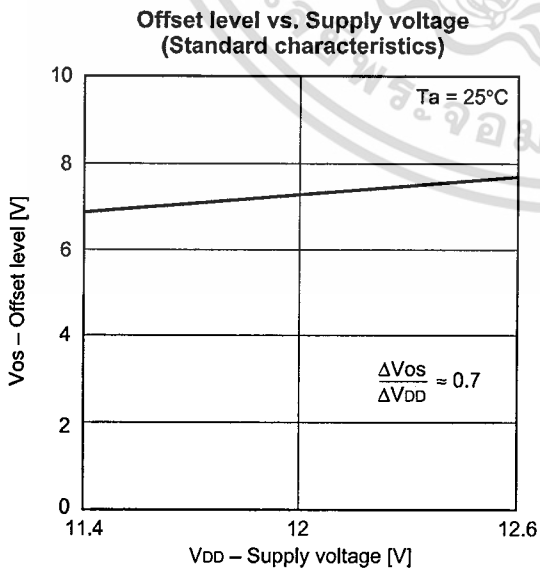
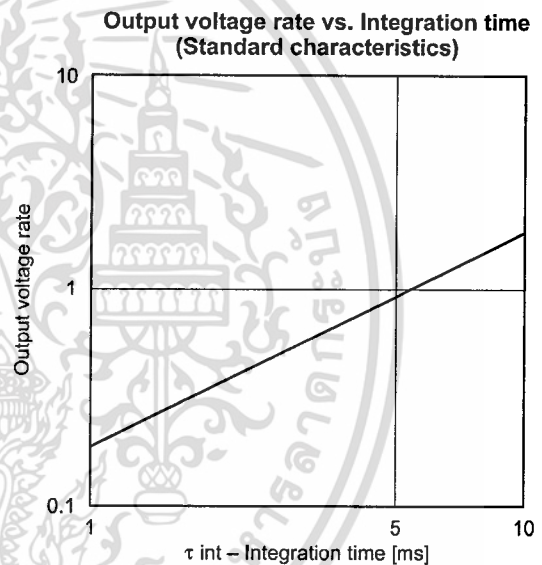
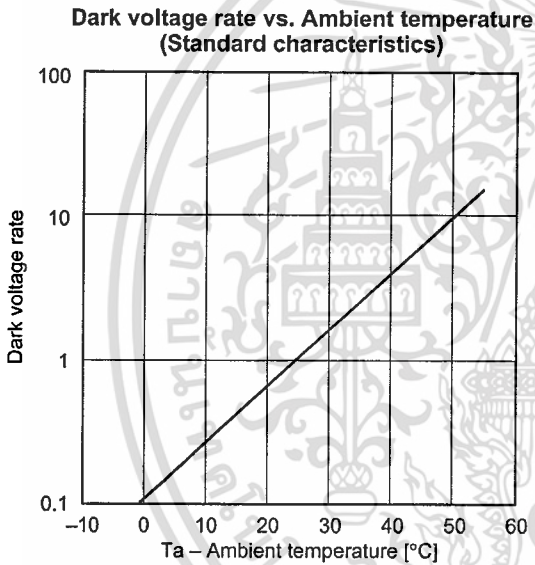
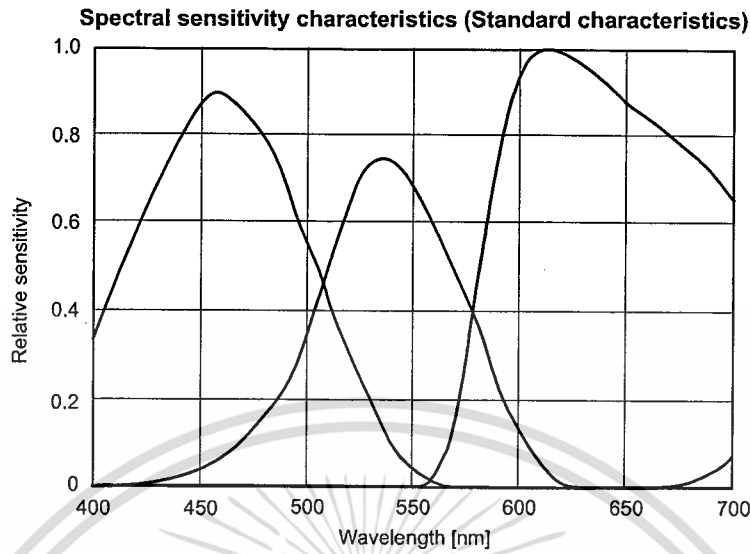
Application Circuit\*



\* Data rate  $f_{\phi RS} = 1\text{MHz}$

Application circuits shown are typical examples illustrating the operation of the devices. Sony cannot assume responsibility for any problems arising out of the use of these circuits or for any infringement of third party patent and other right due to same.

Example of Representative Characteristics ( $V_{DD} = 12V$ ,  $T_a = 25^\circ C$ )



## Notes of Handling

### 1) Static charge prevention

CCD image sensors are easily damaged by static discharge. Before handling be sure to take the following protective measures.

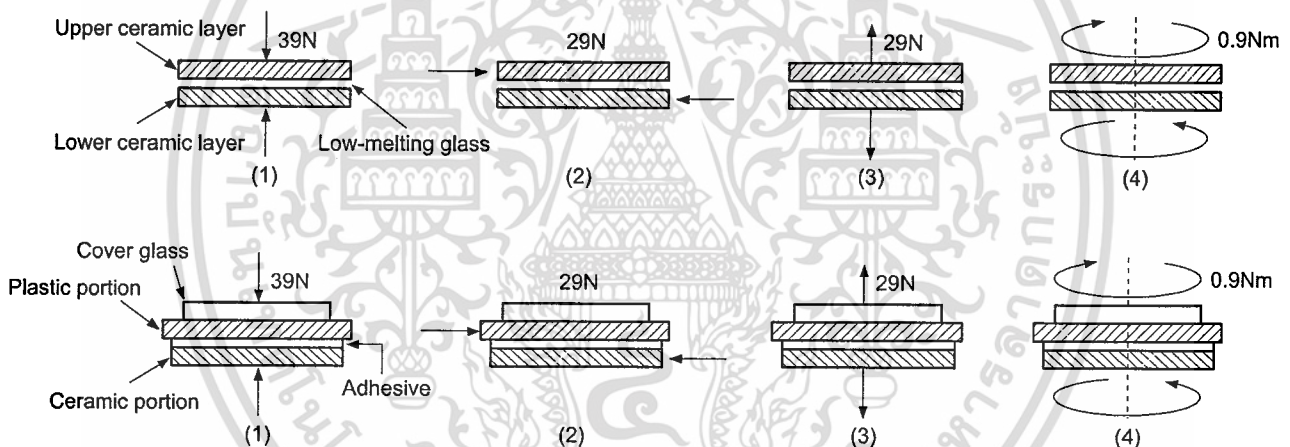
- Either handle bare handed or use non chargeable gloves, clothes or material.  
Also use conductive shoes.
- When handling directly use an earth band.
- Install a conductive mat on the floor or working table to prevent the generation of static electricity.
- Ionized air is recommended for discharge when handling CCD image sensor.
- For the shipment of mounted substrates, use boxes treated for prevention of static charges.

### 2) Notes on Handling CCD Packages

The following points should be observed when handling and installing packages.

#### a) Remain within the following limits when applying static load to the package:

- Compressive strength: 39N/surface (Do not apply load more than 0.7mm inside the outer perimeter of the glass portion.)
- Shearing strength: 29N/surface
- Tensile strength: 29N/surface
- Torsional strength: 0.9Nm



b) In addition, if a load is applied to the entire surface by a hard component, bending stress may be generated and the package may fracture, etc., depending on the flatness of the ceramic portion. Therefore, for installation, either use an elastic load, such as a spring plate, or an adhesive.

c) Be aware that any of the following can cause the package to crack or dust to be generated.

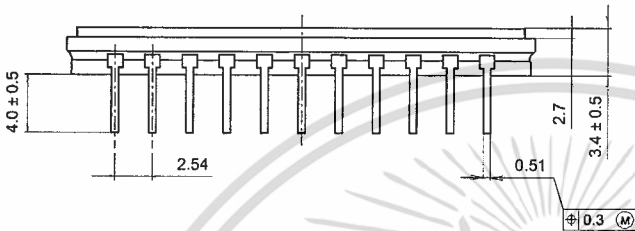
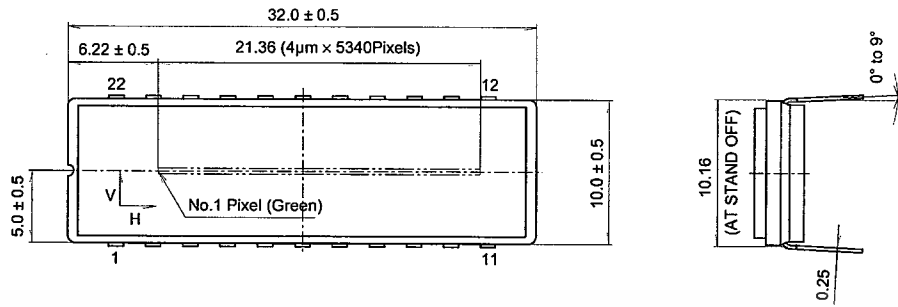
- Applying repetitive bending stress to the external leads.
- Applying heat to the external leads for an extended period of time with soldering iron.
- Rapid cooling or heating.
- Rapid cooling or impact to a limited portion of the low-melting glass with a small-tipped tool such as tweezers.
- Prying the upper or lower ceramic layers away at a support point of the low-melting glass.
- Prying the plastic portion and ceramic portion away at a support point of the adhesive layer.
- Applying the metal a crash or a rub against the plastic portion.

Note that the preceding notes should also be observed when removing a component from a board after it has already been soldered.

- 3) Soldering
  - a) Make sure the package temperature does not exceed 80°C.
  - b) Solder dipping in a mounting furnace causes damage to the glass and other defects. Use a grounded 30W soldering iron and solder each pin in less than 2 seconds. For repairs and remount, cool sufficiently.
  - c) To dismount an imaging device, do not use a solder suction equipment. When using an electric desoldering tool, ground the controller. For the control system, use a zero cross type.
  
- 4) Dust and dirt protection
  - a) Operate in clean environments.
  - b) Do not either touch glass plates by hand or have any object come in contact with glass surfaces. Should dirt stick to a glass surface, blow it off with an air blower. (For dirt stuck through static electricity ionized air is recommended.)
  - c) Clean with a cotton bud and ethyl alcohol if the glass surface is grease stained. Be careful not to scratch the glass.
  - d) Keep in a case to protect from dust and dirt. To prevent dew condensation, preheat or precool when moving to a room with great temperature differences.
  
- 5) Exposure to high temperatures or humidity will affect the characteristics. Accordingly avoid storage or usage in such conditions.
  
- 6) CCD image sensors are precise optical equipment that should not be subject to mechanical shocks.

Package Outline Unit: mm

22pin DIP (400mil)

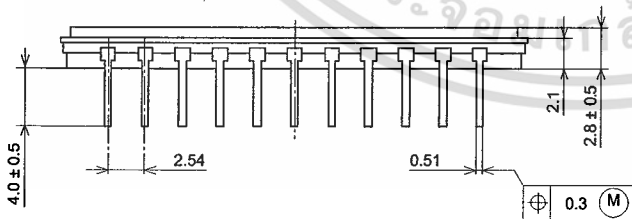
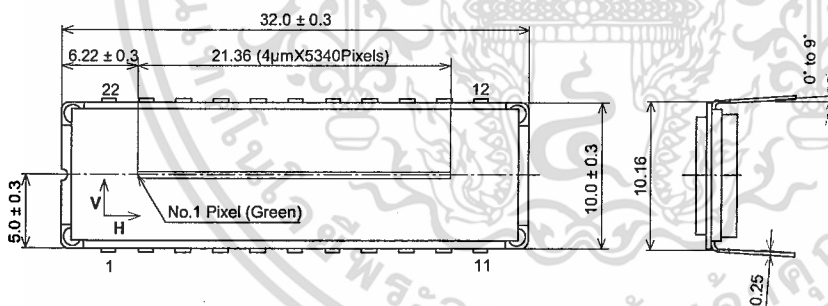


1. The height from the bottom to the sensor surface is  $1.61 \pm 0.3$ mm.
2. The thickness of the cover glass is 0.7mm, and the refractive index is 1.5.

PACKAGE STRUCTURE

PACKAGE MATERIAL	Cer-DIP
LEAD TREATMENT	TIN PLATING
LEAD MATERIAL	42 ALLOY
PACKAGE MASS	3.0g
DRAWING NUMBER	LS-D15(E)

22pin DIP (400mil)



1. The height from the bottom to the sensor surface is  $1.61 \pm 0.3$ mm.
2. The thickness of the cover glass is 0.7mm, and the refractive index is 1.5.

PACKAGE STRUCTURE

PACKAGE MATERIAL	Plastic,Ceramic
LEAD TREATMENT	GOLD PLATING
LEAD MATERIAL	42ALLOY
PACKAGE MASS	2.21g
DRAWING NUMBER	LS-D13-01(E)

This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### FEATURES

- 14-bit 15 MSPS ADC
- No missing codes guaranteed
- 3-channel operation up to 15 MSPS
- 1-channel operation up to 12.5 MSPS
- Correlated double sampling
- 1–6× programmable gain
- ±350 mV programmable offset
- Input clamp circuitry
- Internal voltage reference
- Multiplexed byte-wide output (8 + 6 format)
- 3-wire serial digital interface
- ±1.5 V digital I/O compatibility
- 28-Lead SOIC or SSOP
- Low power CMOS: 385 mW (typ)
- Power-down mode: <1 mW

### APPLICATIONS

- Flatbed document scanners
- Film scanners
- Digital color copiers
- Multifunction peripherals

### GENERAL DESCRIPTION

The AD9822 is a complete analog signal processor for CCD imaging applications. It features a 3-channel architecture designed to sample and condition the outputs of trilinear color CCD arrays. Each channel consists of an input clamp, correlated double sampler (CDS), offset DAC, and programmable gain amplifier (PGA) multiplexed to a high performance 14-bit ADC.

The CDS amplifiers may be disabled for use with sensors such as contact image sensors (CIS) and CMOS active pixel sensors, which do not require CDS.

The 14-bit digital output is multiplexed into an 8-bit output word that is accessed using two read cycles. The internal registers are programmed through a 3-wire serial interface and provide adjustment of the gain, offset, and operating mode.

The AD9822 operates from a single 5 V power supply, consumes 385 mW of power typically, and is packaged in a 28-lead SOIC or SSOP.

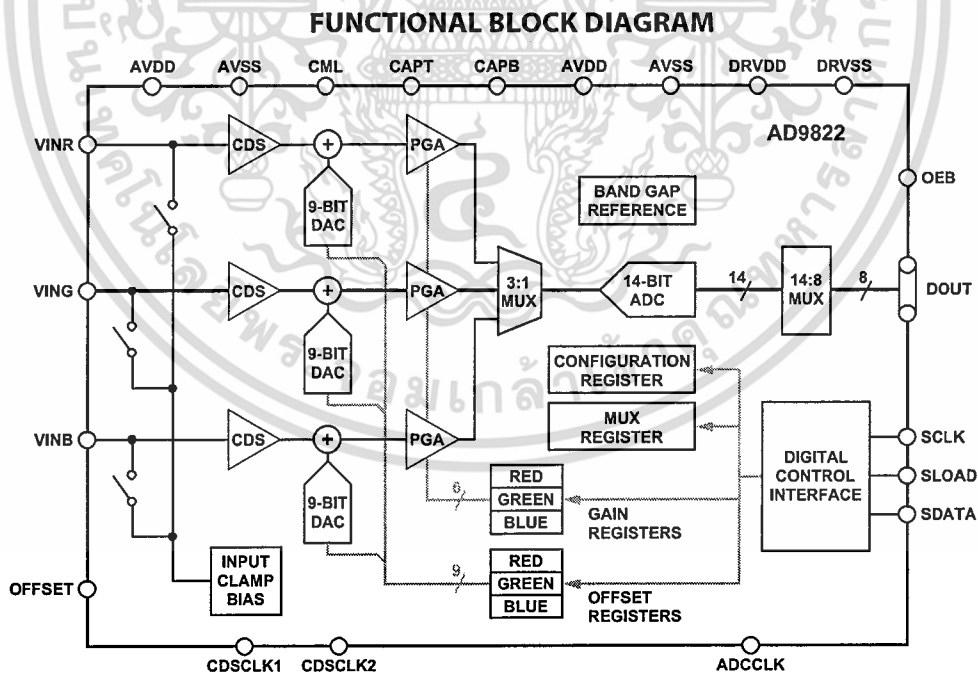


Figure 1.

### Level B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700  
Fax: 781.326.8703

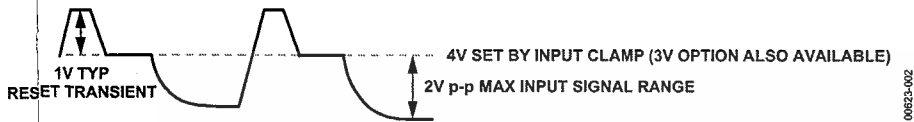
www.analog.com  
© 2005 Analog Devices, Inc. All rights reserved.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AD9822

Parameter	Min	Typ	Max	Unit
<b>POWER DISSIPATION</b>				
3-Channel Mode		385	450	mW
3-Channel Mode @ 6 MHz		335	410	mW
1-Channel Mode		300		mW
1-Channel Mode @ 6 MHz		250		mW

<sup>1</sup> Linear input signal range is from 2 V to 4 V when the CCD's reference level is clamped to 4 V by the AD9822's input clamp.



<sup>2</sup> The PGA gain is approximately linear-in-dB and follows the equation:  $Gain = \left[ \frac{5.7}{1 + 4.7 \left[ \frac{63 - G}{63} \right]} \right]$  where G is the register value. See Figure 15.

## DIGITAL SPECIFICATIONS

$T_{MIN}$  to  $T_{MAX}$ , AVDD = 5 V, DRVDD = 5 V, CDS mode,  $f_{ADCCLK} = 15$  MHz,  $f_{CDSCLK1} = f_{CDSCLK2} = 5$  MHz,  $C_L = 10$  pF, unless otherwise noted.

Table 2.

Parameter	Symbol	Min	Typ	Max	Unit
<b>LOGIC INPUTS</b>					
High Level Input Voltage	$V_{IH}$	2.0			V
Low Level Input Voltage	$V_{IL}$			0.8	V
High Level Input Current	$I_{IH}$		10		$\mu$ A
Low Level Input Current	$I_{IL}$		10		$\mu$ A
Input Capacitance	$C_{IN}$		10		pF
<b>LOGIC OUTPUTS</b>					
High Level Output Voltage	$V_{OH}$	4.5			V
Low Level Output Voltage	$V_{OL}$			0.1	V
High Level Output Current	$I_{OH}$		50		$\mu$ A
Low Level Output Current	$I_{OL}$		50		$\mu$ A

## TABLE OF CONTENTS

Specifications.....	3	1-Channel CDS Mode .....	12
Analog Specifications.....	3	1-Channel SHA Mode .....	12
Digital Specifications .....	4	Internal Register Descriptions.....	13
Timing Specifications .....	5	Circuit Operation .....	15
Absolute Maximum Ratings.....	6	Analog Inputs—CDS Mode.....	15
Thermal Characteristics .....	6	External Input Coupling Capacitors.....	15
Important Caution.....	6	Analog Inputs—SHA Mode.....	16
Configuration and Function Descriptions.....	7	Programmable Gain AmplifierS (PGA).....	16
Pinology .....	8	Applications.....	17
Functional Description .....	12	Circuit and Layout Recommendations .....	17
1-Channel CDS Mode.....	12	Outline Dimensions.....	18
1-Channel SHA Mode.....	12	Ordering Guide .....	18

## REVISION HISTORY

—Rev. A to Rev. B	
Changes to Format .....	Universal
Changes to Ordering Guide .....	18
Updated Outline Dimensions.....	18
—Rev. 0 to Rev. A	

## TIMING SPECIFICATIONS

MIN to T<sub>MAX</sub>, AVDD = 5 V, DRVDD = 5 V.

Table 3.

Parameter	Symbol	Min	Typ	Max	Unit
<b>LOCK PARAMETERS</b>					
3-Channel Pixel Rate	t <sub>PRA</sub>	67			ns
1-Channel Pixel Rate	t <sub>PRB</sub>	80			ns
ADCCLK Pulse Width	t <sub>ADCLK</sub>	30			ns
CDSCLK1 Pulse Width	t <sub>C1</sub>	10			ns
CDSCLK2 Pulse Width	t <sub>C2</sub>	10			ns
CDSCLK1 Falling to CDSCLK2 Rising	t <sub>C1C2</sub>	0			ns
ADCCLK Falling to CDSCLK2 Rising	t <sub>ADC2</sub>	0			ns
CDSCLK2 Rising to ADCCLK Rising	t <sub>C2ADR</sub>	0			ns
CDSCLK2 Falling to ADCCLK Falling	t <sub>C2ADF</sub>	30	40		ns
CDSCLK2 Falling to CDSCLK1 Rising	t <sub>C2C1</sub>	30	40		ns
ADCCLK Falling to CDSCLK1 Rising	t <sub>ADC1</sub>	0			ns
Aperture Delay for CDS Clocks	t <sub>AD</sub>		2		ns
<b>SERIAL INTERFACE</b>					
Maximum SCLK Frequency	f <sub>SCLK</sub>	10			MHz
SLOAD to SCLK Setup Time	t <sub>LS</sub>	10			ns
SCLK to SLOAD Hold Time	t <sub>LH</sub>	10			ns
SDATA to SCLK Rising Setup Time	t <sub>DS</sub>	10			ns
SCLK Rising to SDATA Hold Time	t <sub>DH</sub>	10			ns
SCLK Falling to SDATA Valid	t <sub>RDV</sub>	10			ns
<b>DATA OUTPUT</b>					
Output Delay	t <sub>OD</sub>		8		ns
Three-State to Data Valid	t <sub>DV</sub>		10		ns
Output Enable High to Three-State	t <sub>HZ</sub>		10		ns
Latency (Pipeline Delay)			3 (Fixed)		Cycles

## ABSOLUTE MAXIMUM RATINGS

Table 4.

Parameter	With Respect To	Min	Max	Unit
V <sub>IN</sub> , CAPT, CAPB	AVSS	-0.3	AVDD + 0.3	V
Digital Inputs	AVSS	-0.3	AVDD + 0.3	V
AVDD	AVSS	-0.5	+6.5	V
DRVDD	DRVSS	-0.5	+6.5	V
AVSS	DRVSS	-0.3	+0.3	V
Digital Outputs	DRVSS	-0.3	DRVDD + 0.3	V
Junction Temperature			150	°C
Storage Temperature		-65	+150	°C
Lead Temperature (10 sec)			300	°C

### THERMAL CHARACTERISTICS

#### 28-Lead 300 Mil SOIC

$$\theta_{JA} = 71.4^{\circ}\text{C/W}$$

$$\theta_{JC} = 23^{\circ}\text{C/W}$$

#### 28-Lead 5.3 mm SSOP

$$\theta_{JA} = 109^{\circ}\text{C/W}$$

$$\theta_{JC} = 39^{\circ}\text{C/W}$$

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

### ESD CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

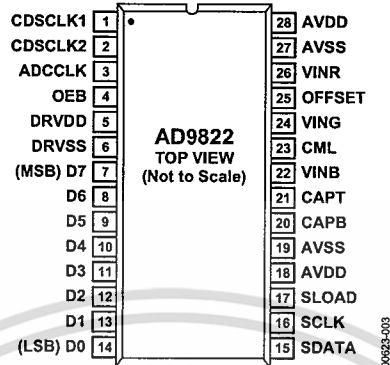


Figure 2. Pin Configuration

Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Type <sup>1</sup>	Description
1	CDSCLK1	DI	CDS Reference Level Sampling Clock.
2	CDSCLK2	DI	CDS Data Level Sampling Clock.
3	ADCCLK	DI	ADC Sampling Clock.
4	OEB	DI	Output Enable, Active Low.
5	DRVDD	P	Digital Output Driver Supply.
6	DRVSS	P	Digital Output Driver Ground.
7	D7 (MSB)	DO	Data Output MSB. ADC DB13 High Byte, ADC DB5 Low Byte.
8	D6	DO	Data Output. ADC DB12 High Byte, ADC DB4 Low Byte.
9	D5	DO	Data Output. ADC DB11 High Byte, ADC DB3 Low Byte.
10	D4	DO	Data Output. ADC DB10 High Byte, ADC DB2 Low Byte.
11	D3	DO	Data Output. ADC DB9 High Byte, ADC DB1 Low Byte.
12	D2	DO	Data Output. ADC DB8 High Byte, ADC DB0 Low Byte.
13	D1	DO	Data Output. ADC DB7 High Byte, Don't Care Low Byte.
14	D0 (LSB)	DO	Data Output LSB. ADC DB6 High Byte, Don't Care Low Byte.
15	SDATA	DI/DO	Serial Interface Data Input/Output.
16	SCLK	DI	Serial Interface Clock Input.
17	SLOAD	DI	Serial Interface Load Pulse.
18	AVDD	P	5 V Analog Supply.
19	AVSS	P	Analog Ground.
20	CAPB	AO	ADC Bottom Reference Voltage Decoupling.
21	CAPT	AO	ADC Top Reference Voltage Decoupling.
22	VINB	AI	Analog Input, Blue Channel.
23	CML	AO	Internal Bias Level Decoupling.
24	VING	AI	Analog Input, Green Channel.
25	OFFSET	AO	Clamp Bias Level Decoupling.
26	VINR	AI	Analog Input, Red Channel.
27	AVSS	P	Analog Ground.
28	AVDD	P	5 V Analog Supply.

<sup>1</sup>Type: AI = Analog Input, AO = Analog Output, DI = Digital Input, DO = Digital Output, P = Power.

## SPECIFICATIONS

### ANALOG SPECIFICATIONS

$T_{MIN}$  to  $T_{MAX}$ , AVDD = 5 V, DRVDD = 5 V, CDS mode,  $f_{ADCLK} = 15$  MHz,  $f_{CDCLK1} = f_{CDCLK2} = 5$  MHz, PGA gain = 1, unless otherwise noted.

Table 1.

Parameter	Min	Typ	Max	Unit
<b>MAXIMUM CONVERSION RATE</b>				
3-Channel Mode with CDS	15			MSPS
1-Channel Mode with CDS	12.5			MSPS
<b>ACCURACY (ENTIRE SIGNAL PATH)</b>				
ADC Resolution		14		Bits
Integral Nonlinearity (INL)		-17.0/+3.5		LSB
INL @ 6 MHz		-10.5/+1.5		LSB
Differential Nonlinearity (DNL)		-0.65/+0.75		LSB
DNL @ 6 MHz	-1.0	-0.6/+0.65	+1.1	LSB
No Missing Codes		14		Bits
No Missing Codes @ 6 MHz	14			Bits
Offset Error	-240	-19	+200	mV
Gain Error	-1.4	+3.5	+6.9	% FSR
<b>ANALOG INPUTS</b>				
Input Signal Range <sup>1</sup>		2.0		V p-p
Allowable Reset Transient <sup>1</sup>		1.0		V
Input Limits <sup>2</sup>	AVSS - 0.3		AVDD + 0.3	V
Input Capacitance		10		pF
Input Bias Current		10		nA
<b>AMPLIFIERS</b>				
PGA Gain at Minimum		1		V/V
PGA Gain at Maximum		5.7		V/V
PGA Gain Resolution <sup>2</sup>		64		Steps
PGA Gain Monotonicity		Guaranteed		
Programmable Offset at Minimum		-350		mV
Programmable Offset at Maximum		+350		mV
Programmable Offset Resolution		512		Steps
Programmable Offset Monotonicity		Guaranteed		
<b>NOISE AND CROSSTALK</b>				
Total Output Noise @ PGA Minimum		1.5		LSB rms
Total Output Noise @ PGA Maximum		6.0		LSB rms
Channel-to-Channel Crosstalk @ 6 MHz		<1		LSB
<b>POWER SUPPLY REJECTION</b>				
AVDD = 5 V ± 0.25 V		0.063	0.9	% FSR
<b>DIFFERENTIAL VREF (@ 25°C)</b>				
CAPT to CAPB (2 V ADC Full-Scale Range)	0.94	1.0	1.06	V
<b>TEMPERATURE RANGE</b>				
Operating	0		+70	°C
Storage	-65		+150	°C
<b>POWER SUPPLIES</b>				
AVDD	4.75	5.0	5.25	V
DRVDD	3.0	5.0	5.25	V
<b>OPERATING CURRENT</b>				
AVDD		73		mA
DRVDD		4		mA
Power-Down Mode Current		150		µA

## TERMINOLOGY

### Integral Nonlinearity (INL)

Integral nonlinearity error refers to the deviation of each individual code from a line drawn from zero scale through positive full scale. The point used as zero scale occurs  $\frac{1}{2}$  LSB before the first code transition. Positive full scale is defined as a level  $1\frac{1}{2}$  LSB beyond the last code transition. The deviation is measured from the middle of each particular code to the true straight line.

### Differential Nonlinearity (DNL)

An ideal ADC exhibits code transitions that are exactly 1 LSB apart. DNL is the deviation from this ideal value; therefore, every code must have a finite width. No missing codes guaranteed to 14-bit resolution indicates that all 16384 codes, respectively, must be present over all operating ranges.

### Offset Error

The first ADC code transition should occur at a level  $\frac{1}{2}$  LSB above the nominal zero-scale voltage. The offset error is the deviation of the actual first code transition level from the ideal level.

### Gain Error

The last code transition should occur for an analog value  $1\frac{1}{2}$  LSB below the nominal full-scale voltage. Gain error is the deviation of the actual difference between the first and last code transitions and the ideal difference between the first and last code transitions.

### Input Referred Noise

The rms output noise is measured using histogram techniques. The ADC output codes' standard deviation is calculated in LSB and converted to an equivalent voltage, using the relationship  $1 \text{ LSB} = 4 \text{ V}/16384 = 244 \text{ mV}$ . The noise is then referred to the input of the AD9822 by dividing by the PGA gain.

### Channel-to-Channel Crosstalk

In an ideal 3-channel system, the signal in one channel will not influence the signal level of another channel. The channel-to-channel crosstalk specification is a measure of the change that occurs in one channel as the other two channels are varied. In the AD9822, one channel is grounded and the other two channels are exercised with full-scale input signals. The change in the output codes from the first channel is measured and compared with the result when all three channels are grounded. The difference is the channel-to-channel crosstalk, stated in LSB.

### Aperture Delay

The time delay that occurs from when a sampling edge is applied to the AD9822 until the actual sample of the input signal is held. Both CDSCLK1 and CDSCLK2 sample the input signal during the transition from high to low; therefore, the aperture delay is measured from each clock's falling edge to the instant the actual internal sample is taken.

### Power Supply Rejection

It specifies the maximum full-scale change that occurs from the initial value when the supplies are varied over the specified limits.

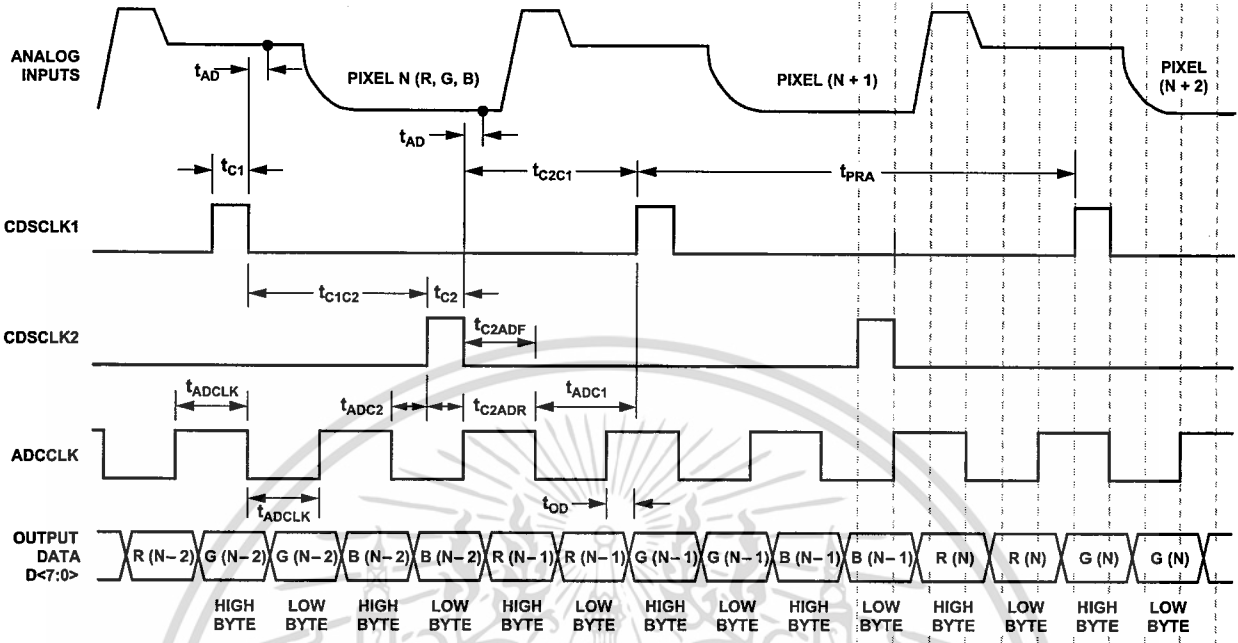


Figure 3. 3-Channel CDS Mode Timing

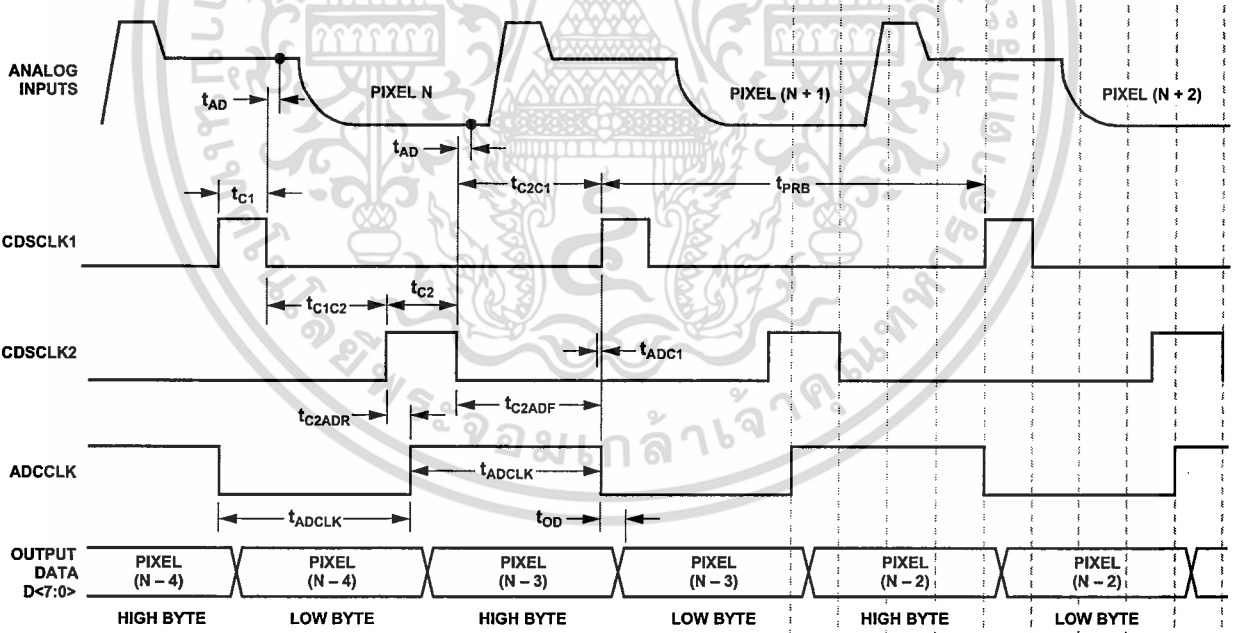


Figure 4. 1-Channel CDS Mode Timing

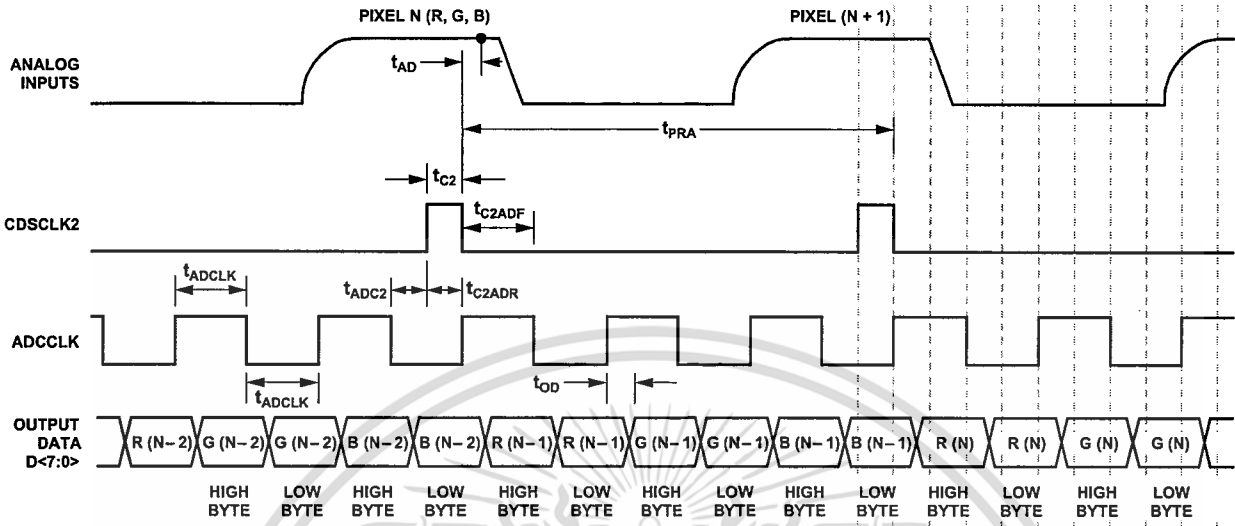


Figure 5. 3-Channel SHA Mode Timing

00823-006

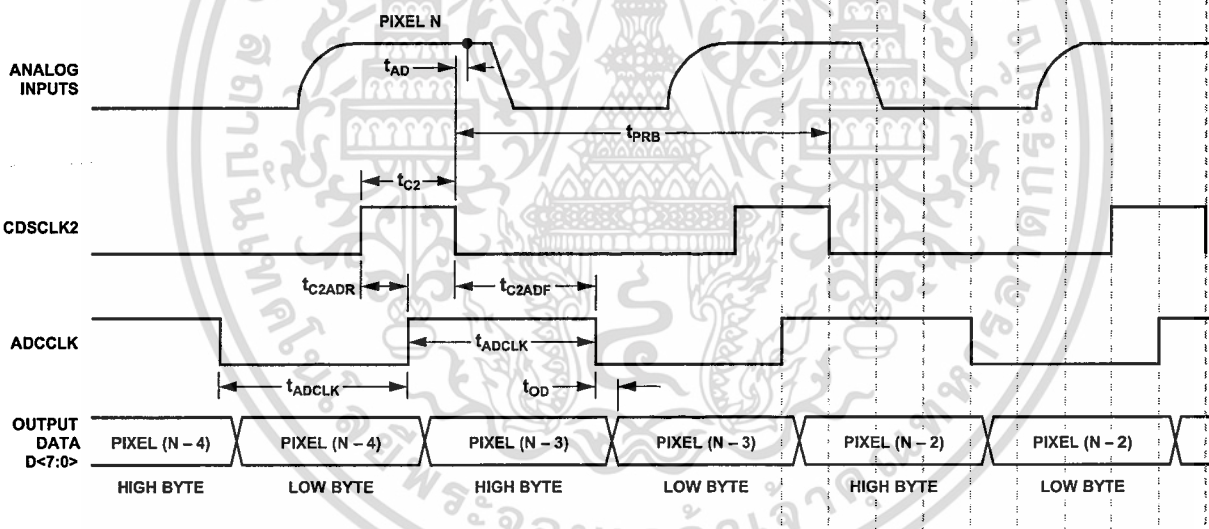


Figure 6. 1-Channel SHA Mode Timing

00823-007

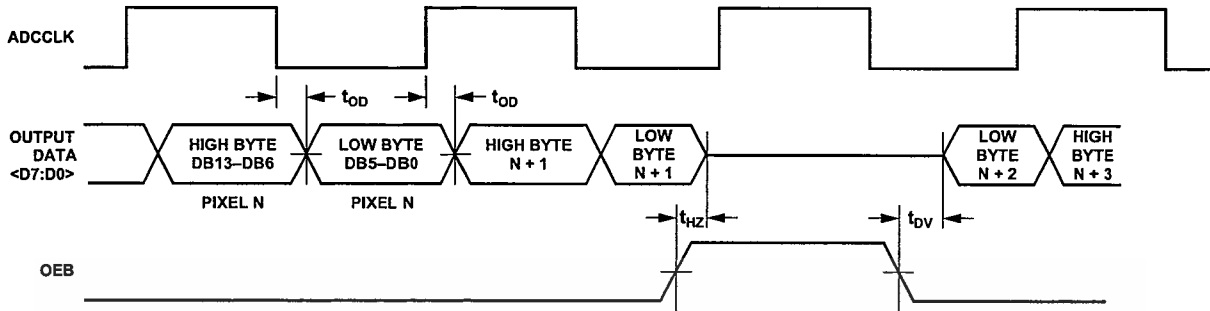


Figure 7. Digital Output Data Timing

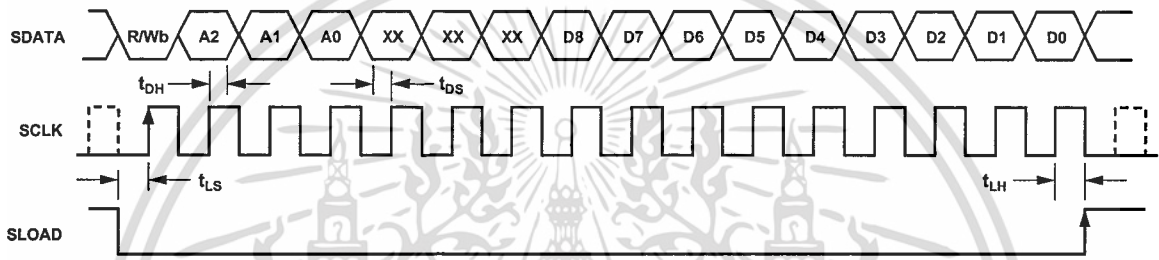


Figure 8. Serial Write Operation Timing

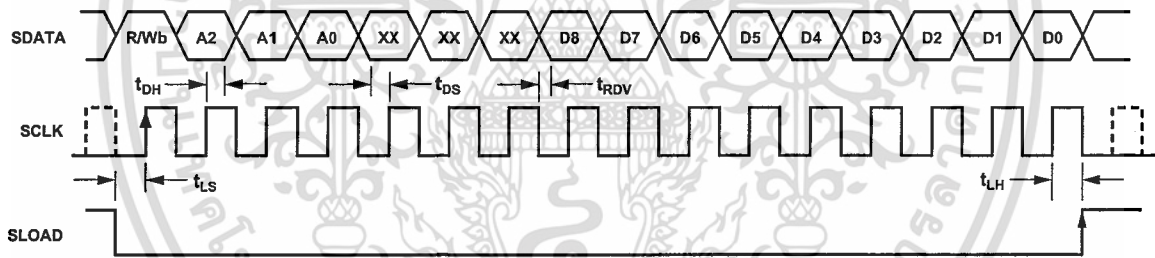


Figure 9. Serial Read Operation Timing

## FUNCTIONAL DESCRIPTION

The AD9822 can be operated in four different modes: 3-channel CDS mode, 3-channel SHA mode, 1-channel CDS mode, and 1-channel SHA mode. Each mode is selected by programming the configuration register through the serial interface. For more information on CDS or SHA mode operation, see the Circuit Operation section.

### 3-CHANNEL CDS MODE

In 3-channel CDS mode, the AD9822 simultaneously samples the red, green, and blue input voltages from the CCD outputs. The sampling points for each CDS are controlled by CDSCLK1 and CDSCLK2 (see Figure 10 and Figure 11). CDSCLK1's falling edge samples the reference level of the CCD waveform, and CDSCLK2's falling edge samples the data level of the CCD waveform. Each CDS amplifier outputs the difference between the CCD's reference and data levels. The output voltage of each CDS amplifier is then level-shifted by an offset DAC. The voltages are scaled by the three PGAs before being multiplexed through the 14-bit ADC. The ADC sequentially samples the PGA outputs on the falling edges of ADCCLK.

The offset and gain values for the red, green, and blue channels are programmed using the serial interface. The order in which the channels are switched through the multiplexer is selected by programming the MUX register.

Timing for this mode is shown in Figure 3. It is recommended that the falling edge of CDSCLK2 occur coincident with or before the rising edge of ADCCLK. However, this is not required to satisfy the minimum timing constraints. The rising edge of CDSCLK2 should not occur before the previous falling edge of ADCCLK, as shown by  $t_{ADC2}$ . The output data latency is three clock cycles.

### 3-CHANNEL SHA MODE

In 3-channel SHA mode, the AD9822 simultaneously samples the red, green, and blue input voltages. The sampling point is controlled by CDSCLK2. CDSCLK2's falling edge samples the input waveforms on each channel. The output voltages from the three SHAs are modified by the offset DACs and then scaled by the three PGAs. The outputs of the PGAs are then multiplexed through the 14-bit ADC. The ADC sequentially samples the PGA outputs on the falling edges of ADCCLK.

The input signal is sampled with respect to the voltage applied to the OFFSET pin (see Figure 12). With the OFFSET pin grounded, a 0 V input corresponds to the ADC's zero-scale output. The OFFSET pin may also be used as a coarse offset adjust pin. A voltage applied to this pin is subtracted from the voltages applied to the red, green, and blue inputs in the first amplifier stage of the AD9822. The input clamp is disabled in this mode. For more information, see the Circuit Operation section.

Timing for this mode is shown in Figure 5. CDSCLK1 should be grounded in this mode. Although not required, it is recommended that the falling edge of CDSCLK2 occur coincident with or before the rising edge of ADCCLK. The rising edge of CDSCLK2 should not occur before the previous falling edge of ADCCLK, as shown by  $t_{ADC2}$ . The output data latency is three ADCCLK cycles.

The offset and gain values for the red, green, and blue channels are programmed using the serial interface. The order in which the channels are switched through the multiplexer is selected by programming the MUX register.

### 1-CHANNEL CDS MODE

This mode operates in the same way as the 3-channel CDS mode. The difference is that the multiplexer remains fixed in this mode; therefore, only the channel specified in the MUX register is processed.

Timing for this mode is shown in Figure 4.

### 1-CHANNEL SHA MODE

This mode operates in the same way as the 3-channel SHA mode, except the multiplexer remains stationary. Only the channel specified in the MUX register is processed.

The input signal is sampled with respect to the voltage applied to the OFFSET pin. With the OFFSET pin grounded, a 0 V input corresponds to the ADC's zero-scale output. The OFFSET pin may also be used as a coarse offset adjust pin. A voltage applied to this pin is subtracted from the voltages applied to the red, green, and blue inputs in the first amplifier stage of the AD9822. The input clamp is disabled in this mode. For more information, see the Circuit Operation section.

Timing for this mode is shown in Figure 6. CDSCLK1 should be grounded in this mode of operation.

**INTERNAL REGISTER DESCRIPTIONS**

**Table 6. Internal Register Map**

Register Name	Address			Data Bits								
	A2	A1	A0	D8	D7	D6	D5	D4	D3	D2	D1	D0
Configuration	0	0	0	0	0	VREF	3Ch/1Ch	CDS On	Clamp	Pwr Dn	0	0
MUX	0	0	1	0	RGB/BGR	Red	Green	Blue	0	0	0	0
Red PGA	0	1	0	0	0	0	MSB					LSB
Green PGA	0	1	1	0	0	0	MSB					LSB
Blue PGA	1	0	0	0	0	0	MSB					LSB
Red Offset	1	0	1	MSB								LSB
Green Offset	1	1	0	MSB								LSB
Blue Offset	1	1	1	MSB								LSB

**Configuration Register**

The Configuration Register controls the AD9822's operating mode and bias levels. Bits D8, D1, and D0 should always be set low. Bit D7 sets the full-scale voltage range of the AD9822's ADC to either 4 V (high) or 2 V (low). Bit D6 controls the internal voltage reference. If the AD9822's internal voltage reference is used, this bit is set high. Setting Bit D6 low disables the internal voltage reference, allowing an external voltage reference to be used. Bit D5 configures the AD9822 for either the 3-channel (high) or 1-channel (low) mode of operation. Setting Bit D4 high enables the CDS mode of operation and setting this bit low enables the SHA mode of operation. Bit D3 sets the dc bias level of the AD9822's input clamp. This bit should always be set high for the 4 V clamp bias, unless a CCD with a reset feedthrough transient exceeding 2 V is used. If the 3 V clamp bias level is used, the peak-to-peak input signal range to the AD9822 is reduced to 3 V maximum. Bit D2 controls the power-down mode. Setting Bit D2 high places the AD9822 into a very low power "sleep" mode. All register contents are retained while the AD9822 is in the power-down state.

**Table 7. Configuration Register Settings**

D8	D7	D6	D5	D4	D3	D2	D1	D0
Set to 0	Set to 0	Internal VREF	No. of Channels	CDS Operation	Input Clamp Bias	Power-Down	Set to 0	Set to 0
		1 = Enabled <sup>1</sup> 0 = Disabled	1 = 3-Ch Mode <sup>1</sup> 0 = 1-Ch Mode	1 = CDS Mode <sup>1</sup> 0 = SHA Mode	1 = 4 V <sup>1</sup> 0 = 3 V	1 = On 0 = Off (Normal) <sup>1</sup>		

<sup>1</sup> Power-on default value.

**MUX Register**

The MUX register controls the sampling channel order in the AD9822. Bits D8, D3, D2, D1, and D0 should always be set low. Bit D7 is used when operating in 3-channel mode. Setting Bit D7 high sequences the MUX to sample the red channel first, then the green channel, and then the blue channel. When in this mode, the CDSCLK2 pulse always resets the MUX to sample the red channel first (see Figure 3). When Bit D7 is set low, the channel order is reversed to blue first, green second, and red third. The CDSCLK2 pulse always resets the MUX to sample the blue channel first. Bits D6, D5, and D4 are used when operating in 1-channel mode. Bit D6 is set high to sample the red channel. Bit D5 is set high to sample the green channel. Bit D4 is set high to sample the blue channel. The MUX remains stationary during 1-channel mode.

**Table 8. MUX Register Settings**

D8	D7	D6	D5	D4	D3	D2	D1	D0
Set to 0	3-Channel Select	1-Channel Select	1-Channel Select	1-Channel Select	Set to 0	Set to 0	Set to 0	Set to 0
	1 = R-G-B <sup>1</sup> 0 = B-G-R	1 = RED <sup>1</sup> 0 = Off	1 = GREEN 0 = Off <sup>1</sup>	1 = BLUE 0 = Off <sup>1</sup>				

<sup>1</sup> Power-on default value.

# AD9822

## PGA Gain Registers

There are three PGA registers for individually programming the gain in the red, green, and blue channels. Bits D8, D7, and D6 in each register must be set low, and Bits D5 through D0 control the gain range in 64 increments. See Figure 15 for the PGA gain vs. the PGA register code. The coding for the PGA registers is straight binary, with an all 0s word corresponding to the minimum gain setting (1×) and an all 1s word corresponding to the maximum gain setting (5.7×).

Table 9. PGA Gain Register Settings

D8	D7	D6	D5	D4	D3	D2	D1	D0	Gain (V/V)	Gain (dB)
Set to 0	Set to 0	Set to 0	MSB					LSB		
0	0	0	0	0	0	0	0	0 <sup>1</sup>	1.0	0.0
0	0	0	0	0	0	0	0	1	1.013	0.12
									•	•
									•	•
									•	•
0	0	0	1	1	1	1	1	0	5.4	14.6
0	0	0	1	1	1	1	1	1	5.7	15.1

<sup>1</sup> Power-on default value.

## Offset Registers

There are three PGA registers for individually programming the offset in the red, green, and blue channels. Bits D8 through D0 control the offset range from -350 mV to +350 mV in 512 increments. The coding for the offset registers is sign magnitude, with D8 as the sign bit. Table 10 shows the offset range as a function of the Bits D8 through D0.

Table 10. Offset Register Settings

D8 (MSB)	D7	D6	D5	D4	D3	D2	D1	D0 (LSB)	Offset (mV)
0	0	0	0	0	0	0	0	0 <sup>1</sup>	0
0	0	0	0	0	0	0	0	1	+1.2
									•
									•
									•
0	1	1	1	1	1	1	1	1	+350
1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	-1.2
									•
									•
									•
1	1	1	1	1	1	1	1	1	-350

<sup>1</sup> Power-on default value.

# CIRCUIT OPERATION

## ANALOG INPUTS—CDS MODE

Figure 10 shows the analog input configuration for the CDS mode of operation. Figure 11 shows the internal timing for the sampling switches. The CCD reference level is sampled when CDSCLK1 transitions from high to low, opening S1. The CCD data level is sampled when CDSCLK2 transitions from high to low, opening S2. S3 is then closed, generating a differential input voltage representing the difference between the two sampled levels.

The input clamp is controlled by CDSCLK1. When CDSCLK1 is high, S4 closes and the internal bias voltage is connected to the analog input. The bias voltage charges the external 0.1 μF input capacitor, level-shifting the CCD signal into the AD9822's input common-mode range. The time constant of the input clamp is determined by the internal 5 kΩ resistance and the external 0.1 μF input capacitance.

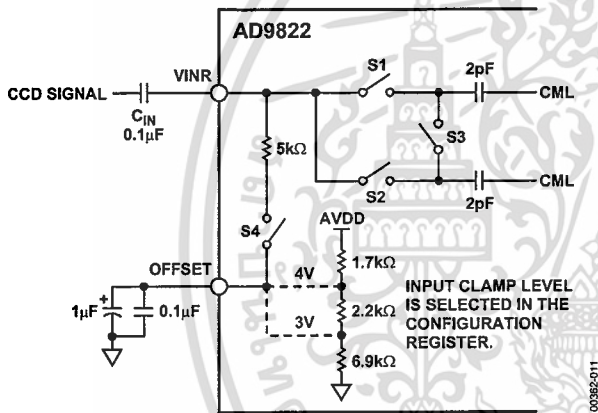


Figure 10. CDS Mode Input Configuration (All Three Channels are Identical)

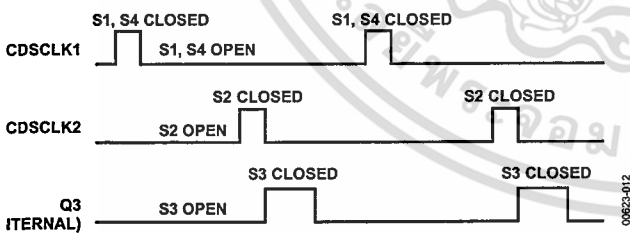


Figure 11. CDS Mode Internal Switch Timing

## EXTERNAL INPUT COUPLING CAPACITORS

The recommended value for the input coupling capacitors is 0.1 μF. While it is possible to use a smaller capacitor, this larger value is chosen for several reasons:

- **Signal Attenuation:** The input coupling capacitor creates a capacitive divider with a CMOS integrated circuit's input capacitance, attenuating the CCD signal level. CIN should be large relative to the IC's 10 pF input capacitance in order to minimize this effect.
- **Linearity:** Some of the input capacitance of a CMOS IC is junction capacitance, which varies nonlinearly with applied voltage. If the input coupling capacitor is too small, the attenuation of the CCD signal varies nonlinearly with signal level. This degrades the system linearity performance.
- **Sampling Errors:** The internal 2 pF sample capacitors have a "memory" of the previously sampled pixel. There is a charge redistribution error between CIN and the internal sample capacitors for larger pixel-to-pixel voltage swings. As the value of CIN is reduced, the resulting error in the sampled voltage increases. With a CIN value of 0.1 μF, the charge redistribution error is less than 1 LSB for a full-scale, pixel-to-pixel voltage swing.

## ANALOG INPUTS—SHA MODE

Figure 12 shows the analog input configuration for the SHA mode of operation. Figure 13 shows the internal timing for the sampling switches. The input signal is sampled when CDSCLK2 transitions from high to low, opening S1. The voltage on the OFFSET pin is also sampled on the falling edge of CDSCLK2, when S2 opens. S3 is then closed, generating a differential output voltage representing the difference between the sampled input voltage and the OFFSET voltage. The input clamp is disabled during SHA mode operation.

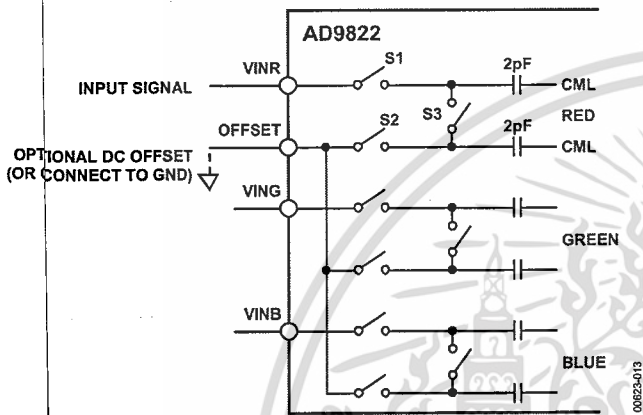


Figure 12. SHA Mode Input Configuration (All Three Channels are Identical)

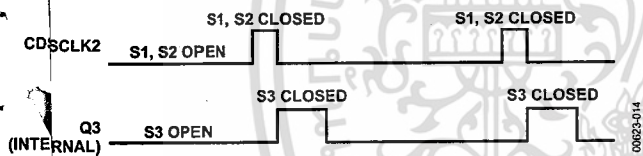


Figure 13. SHA Mode Internal Switch Timing

Figure 14 shows how the OFFSET pin may be used in a CIS application for coarse offset adjustment. Many CIS signals have dc offsets ranging from several hundred millivolts to more than 1 V. By connecting the appropriate dc voltage to the OFFSET pin, the CIS signal is restored to 0. After the large dc offset is removed, the signal can be scaled using the PGA to maximize the ADC's dynamic range.

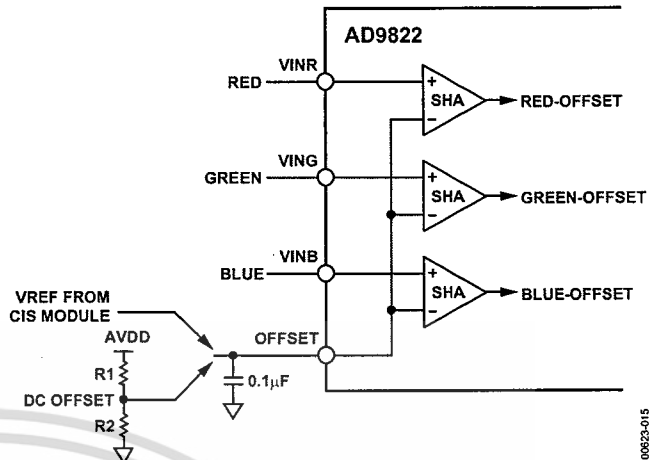


Figure 14. SHA Mode Used with External DC Offset

## PROGRAMMABLE GAIN AMPLIFIERS (PGA)

The AD9822 uses one PGA for each channel. Each PGA has a gain range from  $1\times$  (0 dB) to  $5.8\times$  (15.5 dB), adjustable in 64 steps. Figure 15 shows the PGA gain as a function of the PGA register code. Although the gain curve is approximately linear-in-dB, the gain in V/V varies nonlinearly with register code, following the equation

$$\text{Gain} = \frac{5.7}{1 + 4.7 \left[ \frac{63 - G}{63} \right]}$$

where G is the decimal value of the gain register contents and varies from 0 to 63.

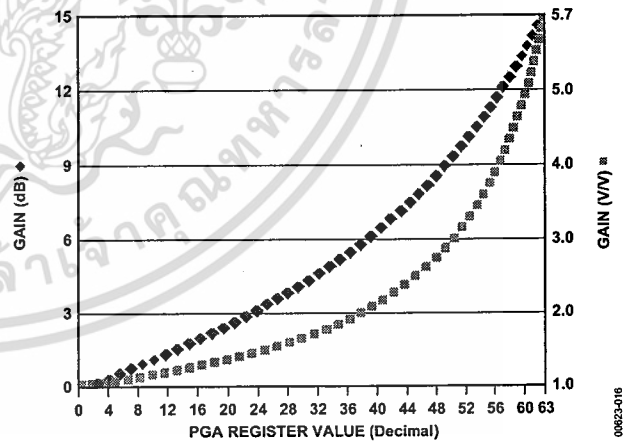


Figure 15. PGA Gain Transfer Function

# APPLICATIONS

## CIRCUIT AND LAYOUT RECOMMENDATIONS

Figure 16 shows the recommended circuit configuration for channel CDS mode operation. The recommended input coupling capacitor value is 0.1  $\mu\text{F}$  (see the Circuit Operation section). A single ground plane is recommended for the AD9822. A separate power supply may be used for DRVDD, the digital driver supply, but this supply pin should still be decoupled to the same ground plane as the rest of the AD9822. The loading of the digital outputs should be minimized, either by using short traces to the digital ASIC or by using external digital buffers. To minimize the effect of digital transients during major output code transitions, the falling edge of

CDSCLK2 should occur coincident with or before the rising edge of ADCCLK (see Figure 3 through Figure 6 for timing). All 0.1  $\mu\text{F}$  decoupling capacitors should be located as close as possible to the AD9822 pins. When operating in single-channel mode, the unused analog inputs should be grounded.

Figure 17 shows the recommended circuit configuration for 3-channel SHA mode. All of the above considerations also apply for this configuration, except that the analog input signals are directly connected to the AD9822 without the use of coupling capacitors. The analog input signals must already be dc-biased between 0 V and 2 V (see the Circuit Operation section).

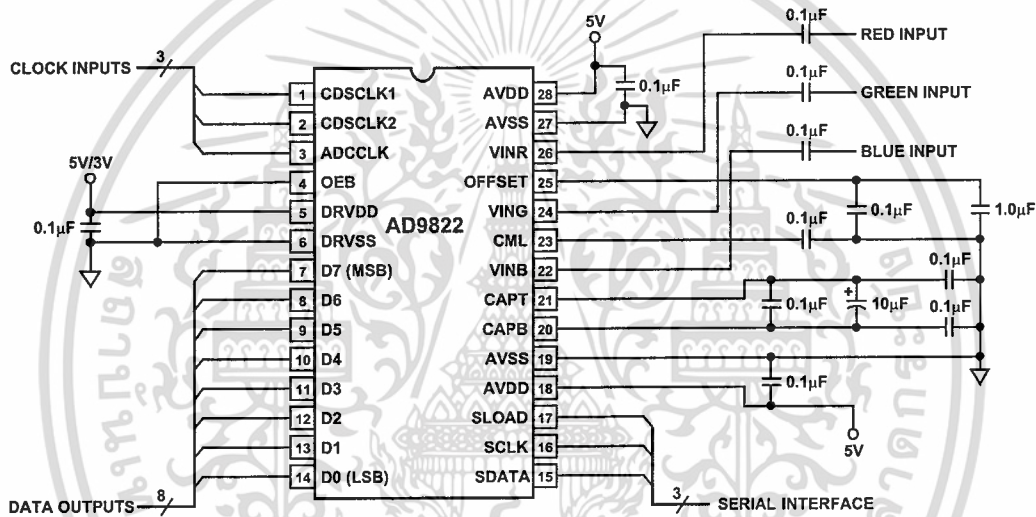


Figure 16. Recommended Circuit Configuration, 3-Channel CDS Mode

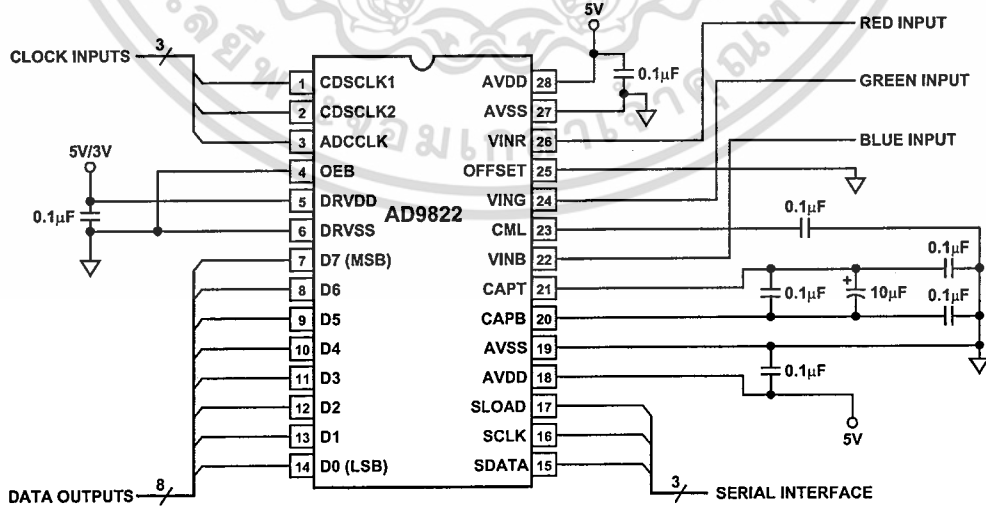
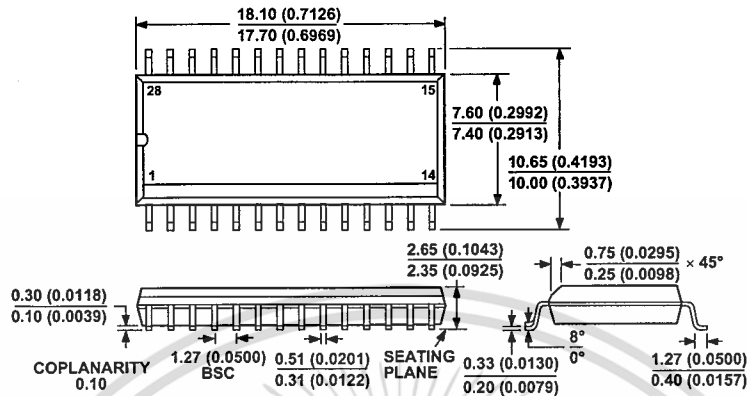


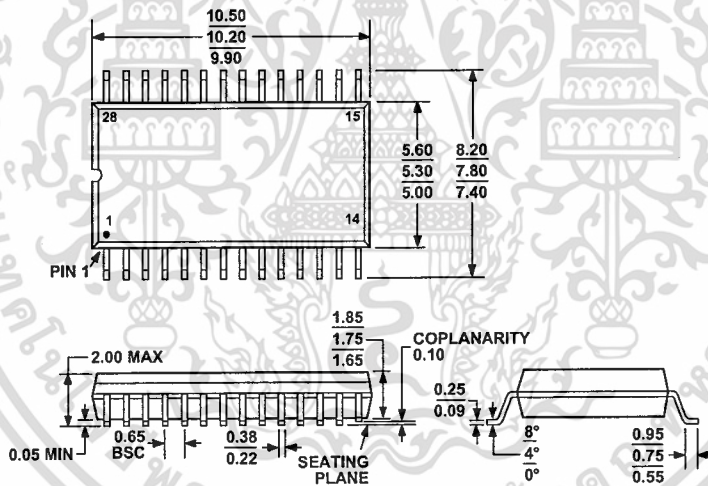
Figure 17. Recommended Circuit Configuration, 3-Channel SHA Mode (Analog Inputs Sampled with Respect to Ground)

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MS-013AE  
 CONTROLLING DIMENSIONS ARE IN MILLIMETERS; INCH DIMENSIONS  
 (IN PARENTHESES) ARE ROUNDED-OFF MILLIMETER EQUIVALENTS FOR  
 REFERENCE ONLY AND ARE NOT APPROPRIATE FOR USE IN DESIGN

Figure 18. 28-Lead Standard Small Outline Package [SOIC]  
 Wide Body (R-28)  
 Dimensions shown in millimeters and (inches)



COMPLIANT TO JEDEC STANDARDS MO-150AH

Figure 19. 28-Lead Shrink Small Outline Package [SSOP]  
 (RS-28)  
 Dimensions shown in millimeters

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Options
AD9822JR	0°C to 70°C	28-Lead SOIC	R-28
AD9822JRRL	0°C to 70°C	28-Lead SOIC	R-28
AD9822JRS	0°C to 70°C	28-Lead SSOP	RS-28
AD9822JRSRL	0°C to 70°C	28-Lead SSOP	RS-28
AD9822JRSZ <sup>1</sup>	0°C to 70°C	28-Lead SSOP	RS-28
AD9822JRSZRL <sup>1</sup>	0°C to 70°C	28-Lead SSOP	RS-28

<sup>1</sup> Z = Pb-free part.

## NOTES



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
Rev. B | Page 19 of 20  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

