สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บอร์ดแสดงผล LED ขั้นสูง

ADVANCE LED DISPLAY BOARD

นาย สุเมธี ตรีมงคล
นาย สุรกิจ ศรีวรากร
นาย เสกสรร พรหมศิริไพบูลย์

วพ.
ส843ป
2549

เลขหมู่.....
เลขทะเบียน 72761
วัน.เดือน.ปี 22 มิ.ย. 2550

.b 11272335
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษาที่ 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดแสดงผล LED ขั้นสูง

ADVANCE LED DISPLAY BOARD

นาย สุเมธี ตรีมงคล เลขประจำตัว 46010864

นาย สุรกิจ ศรีวรากร เลขประจำตัว 46010866

นาย เสกสรร พรหมศิริไพบูลย์ เลขประจำตัว 46010892

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษาที่ 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงาน (ภาษาไทย) บอร์ดแสดงผล LED ขั้นสูง

(ภาษาอังกฤษ) ADVANCE LED DISPLAY BOARD

ชื่อและเลขประจำตัว นาย สุเมธี ตรีมงคล เลขประจำตัว 46010864
นาย สุรกิจ ศรีวรากร เลขประจำตัว 46010866
นาย เสกสรร พรหมศิริไพบูลย์ เลขประจำตัว 46010892

รายงานฉบับนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา
(อาจารย์ ชินภัทร นันทจิวากรชัย)

วันที่ ...9.../...ม.ค.../...50...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดแสดงผล LED ขั้นสูง

นาย สุเมธี ตรีมงคล รหัส 46010864
นาย สุรกิจ ศรีวรากร รหัส 46010866
นาย เสกสรร พรหมศิริไพบูลย์ รหัส 46010892
อ. ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา
ภาคเรียนที่ 2 ปีการศึกษา 2549

บทคัดย่อ

ในรายงานฉบับนี้ เป็นการศึกษาเกี่ยวกับการแสดงผลสัญญาณภาพ ผ่านทาง บอร์ดแสดงผล LED โดยพัฒนารูปแบบของการแสดงข้อมูลบนบอร์ดแสดงผลแบบจุด (Dot Matrix Display Board) ซึ่งใช้ LED ชนิด Dot Matrix ขนาด 8 × 8 สองสี คือ สีแดง และสีเขียว มาใช้ในการสร้างบอร์ดแสดงผลซึ่งมีความละเอียด 176 × 256 จุด โดยใช้คอมพิวเตอร์ส่วนบุคคลส่งข้อมูลภาพที่ต้องการแสดงผลผ่านทางสายสัญญาณ VGA โดยสัญญาณภาพจากที่ได้รับมาจากคอมพิวเตอร์ต้องผ่านวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ก่อนที่จะนำข้อมูลมาประมวลผลซึ่งใช้ FPGA (Field Programmable Gate Array) ในการจัดเรียงข้อมูล โดยข้อมูลที่ได้จากการจัดเรียงแล้วจะส่งไปยังบอร์ดแสดงผล LED ซึ่งบอร์ดแสดงผลจะประกอบด้วยวงจรต่างๆ เช่น วงจรพักข้อมูล วงจรขับกระแส เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADVANCE LED DISPLAY BOARD

Sumetee Tremongkol ID. 46010864
Surakit Srivarakorn ID. 46010866
Seksun Promsiripaibool ID. 46010892
Chinnapat Nantajivagornchai Advisor
Second Semester,Educational Year 2006

ABSTRACT

This report is to study and develop graphic signal for led display board. Led display board consists of led dot Matrix 8 × 8 two colors red and green, it has dimension 176 × 256 dot matrix. The FPGA is used to receive graphic data from a personal computer, which graphic signal is analog there for the data are must to convert by A/D converter before sent to FPGA. This FPGA process in order to create digital data for sent to led display board. The LED's Board circuit consists of many parts for example such as latch buffer circuit, drive circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่อง ADVANCE LED DISPLAY BOARD จะประสบความสำเร็จลงไม่ได้เลย หากขาดผู้ที่มีอุปการะคุณทุกท่านที่ให้คำแนะนำและคำปรึกษาที่ดีมาโดยตลอด ทางคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงโดยสามารถเรียงลำดับบุคคลที่มีพระคุณดังต่อไปนี้

-อาจารย์ ชินภัทร นันทจิวากรชัย ที่ให้คำปรึกษาในทุกๆเรื่องที่เกี่ยวกับการทำโครงงานชิ้นนี้

-อาจารย์ พลผดุง พดุงกุล ที่ให้คำปรึกษาในเรื่องสัญญาณวีดีโอและการสแกน

-อาจารย์ ประภากร สุวรรณะ ที่ให้คำปรึกษาในเรื่องการทำความสว่างหลายระดับ

-อาจารย์ ยุทธนา คิดใจเดียว ที่ให้คำปรึกษาในเรื่องการตอบสนองของดวงตาต่อแสงสว่างภายนอก

-คุณพ่อ ประมาณ ดรีมงคล ที่ช่วยในด้านการประกอบโครงเหล็กฉากเป็นจอแสดงผล LED ขนาดใหญ่

-พี่อ้อ พี่หมู พี่ยันต์ ที่ช่วยในด้านเอกสารของโครงงานและโปรแกรมต่างๆ

.....

(นาย สุเมธี ตรีมงคล)

.....

(นาย สุรกิจ ศรีวรากร)

.....

(นาย เสกสรร พรหมศิริไพบูลย์)

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อ	viii
ABSTRACT	ix
กิตติกรรมประกาศ	x
บทที่ 1 บทนำ	1
-ความมุ่งหมายของโครงงาน	1
-วัตถุประสงค์ของโครงงาน	2
-ขอบเขตของโครงงาน	2
-ประโยชน์ที่ได้รับจากโครงงาน	2
บทที่ 2 ทฤษฎี	3
2.1 ไดโอดเปล่งแสง (Light Emitting Diode: LED)	3
-การขับ LED ด้วยวิธีแบบสแตติกและแบบมัลติเพล็กซ์	8
2.2 ทฤษฎีเบื้องต้น เกี่ยวกับการทำงานของจอ VGA	9
2.3 การจัดเก็บสัญญาณภาพ	10
2.4 การสุ่ม (Sampling)	12
2.5 การควอนไตซ์ (Quantizatoin)	13
2.6 ลักษณะหน่วยความจำที่ใช้ในการเก็บข้อมูล	14
2.7 องค์ประกอบของภาพ	15
2.8 การสแกน (Scanning)	16
-หลักของการสแกน	17
2.9 ชิฟรีจิสเตอร์(Shift register)	18
2.10 อุปกรณ์หน่วยความจำ	19
2.11 Block RAM ในชิพ FPGA	20
2.12 การสื่อสารแบบอนุกรม (Serial Communication)	22
บทที่ 3 ภาษา VHDL	30
3.1 การออกแบบระบบดิจิตอล	30
3.2 ข้อกำหนดของภาษา VHDL	31
3.3 ส่วนประกอบต่างๆของภาษา VHDL	33
-หน่วยการออกแบบเอนทิตี	34
-หน่วยการออกแบบสถาปัตยกรรม	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
-หน่วยการออกแบบแพ็กเก็ต	40
-หน่วยการออกแบบโครงแบบ	42
3.4 การออกแบบจากบนลงล่าง	42
บทที่ 4 การออกแบบและการสร้าง	44
4.1 ลักษณะการเชื่อมต่อโดยรวมของระบบ	44
4.2 SCANNING	45
4.3 วงจรขับกระแส	46
4.4 วงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์ และ รีจิสเตอร์	46
4.5 ระบบแสดงภาพนิ่ง	47
4.5.1 โปรแกรมในส่วนของการติดต่อผู้ใช้งาน (User interface program)	48
4.5.2 วงจรภายใน FPGA	49
4.6 ระบบแสดงภาพเคลื่อนไหว	54
4.6.1 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (A/D)	55
4.6.2 วงจรภายใน FPGA	58
บทที่ 5 ผลการทดลอง	64
5.1 ส่วนของวงจรแปลงสัญญาณวีดีโอให้เป็นสัญญาณดิจิตอล	64
5.1.1 ผลการทดลองจากการ Simulation	64
5.1.2 ผลการทดลองจากการทดลองทางฮาร์ดแวร์	66
5.2 ส่วนของการแสดงภาพนิ่งผ่านทางโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6	68
5.2.1 ผลการ Simulation	68
5.2.2 ผลการทดลองกับฮาร์ดแวร์	71
บทที่ 6 สรุปและวิจารณ์ผลการทดลอง	74
-สรุปผลการทดลอง	74
-วิจารณ์ผลการทดลอง	74
-อุปสรรคที่พบในการทำโครงงาน	75
-ประโยชน์ที่ได้รับจากโครงงานนี้	75
หนังสืออ้างอิง	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2.1 แสดงการปลดปล่อยพลังงานออกมาในรูปของโฟตอน	4
รูปที่ 2.2 แสดงลักษณะของ LED ในรูปแบบต่างๆ	4
รูปที่ 2.3 แสดงโครงสร้างภายในของ LED	5
รูปที่ 2.4 แสดงลักษณะการเปล่งแสงของ LED	5
รูปที่ 2.5 แสดงแถบพลังงานของ LED (a) เมื่อไม่มีการป้อนศักดาไฟฟ้าให้ LED (b) เมื่อมีการจ่ายศักดาไฟฟ้าให้ LED	6
รูปที่ 2.6 แสดงการมองเห็นแสงของ LED ในย่านความยาวคลื่นต่างๆ	7
รูปที่ 2.7 แสดงรูป DOT MATRIX LED ที่นำมาใช้เป็น Display Board ในโครงงาน	8
รูปที่ 2.8 VGA connector ตัวผู้ที่มองออกมาจากสายเคเบิลของจอมอนิเตอร์	9
รูปที่ 2.9 แสดงการทำงานของจอ VGA โดยใช้สัญญาณ HS และ VS	10
รูปที่ 2.10 แสดงฟังก์ชันของการสุ่ม	12
รูปที่ 2.11 แสดงวิธีการควอนไตซ์สัญญาณ	13
รูปที่ 2.12 BLOCK DIAGRAM SRAM IS61C64AH 8K × 8 HIGH-SPEED CMOS STATIC RAM	15
รูปที่ 2.13 ชิฟท์รีจิสเตอร์ซีเรียลอิน/ซีเรียลเอาท์ (serial-in, serial-out shift register) (a)ชิฟท์รีจิสเตอร์แบบเซลล์มาสเตอร์-สเลฟชิฟรีจิสเตอร์ (shift register with master-slave cell) (b) ชิฟท์รีจิสเตอร์ อาร์เอส ฟลิปฟลอปทำงานที่ขอบของสัญญาณนาฬิกา (shift register with edge-triggered RS flip flop)	19
รูปที่ 2.14 แผนภาพเวลาของแลตช์ และ ฟลิปฟลอป (a) แลตช์ถูกกระตุ้นด้วยอินพุตเท่านั้น (b) ฟลิปฟลอปถูกกระตุ้นด้วยอินพุตและสัญญาณนาฬิกา	20
รูปที่ 2.15 ตาราง Block RAM ที่มีใช้ใน FPGA ตระกูล Spartan-3 และ Spartan-3L	21
รูปที่ 2.16 แสดงลักษณะโครงสร้างของ Block RAM	21
รูปที่ 2.17 การส่งข้อมูลแบบอนุกรม	22
รูปที่ 2.18 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมแบบ DB-9	23
รูปที่ 2.19 การต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
รูปที่ 2.20 รูปแบบการสื่อสารแบบอนุกรม	24
รูปที่ 2.21 วงจรแปลงระดับแรงดันของการสื่อสารแบบอนุกรม	24
รูปที่ 2.22 การแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม	25
รูปที่ 2.23 รูปแบบการส่งข้อมูลแบบอนุกรม	26
รูปที่ 2.24 State Diagram ของการส่งข้อมูลแบบอนุกรม	27
รูปที่ 2.25 รูปแบบการรับข้อมูลแบบอนุกรม	28
รูปที่ 2.26 State Diagram ของการรับข้อมูลแบบอนุกรม	29
รูปที่ 3.1 แสดงขั้นตอนการออกแบบวงจรดิจิตอล	30
รูปที่ 3.2 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี	34
รูปที่ 3.3 แสดงรูปแบบของมัลติเพลกซ์	35
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล	
(b) มุมมองของตัวเชื่อมประสาน (interfacing)	
รูปที่ 3.4 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย	35
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล	
(b) มุมมองของตัวเชื่อมประสาน (interfacing)	
รูปที่ 3.5 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก	36
รูปที่ 3.6 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม	36
รูปที่ 3.7 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน	37
รูปที่ 3.8 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์	38
รูปที่ 3.9 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง	39
รูปที่ 3.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม	39
รูปที่ 3.11 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต	40
รูปที่ 3.12 โครงสร้างของบอดี้แพ็กเก็ต	41
รูปที่ 3.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	41
รูปที่ 3.14 ขั้นตอนการออกแบบจากบนลงล่าง	42
รูปที่ 4.1 ลักษณะการเชื่อมต่อของระบบทั้งหมด	44
รูปที่ 4.2 แสดงค่าทางกายภาพและคุณสมบัติทางไฟฟ้าของสาย Pair	45
รูปที่ 4.3 วงจรขับกระแส	46
รูปที่ 4.4 แสดงส่วนของวงจรรักษาระดับข้อมูล	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 4.5 บล็อกไดอะแกรมโดยรวมของระบบแสดงภาพนิ่ง 48
รูปที่4.6 ลักษณะหน้าต่างที่ใช้ติดต่อกับผู้ใช้งาน 49
รูปที่ 4.7 วงจรภายใน FPGA ระบบแสดงภาพนิ่ง 50
รูปที่ 4.8 ส่วนควบคุมความถี่ 51
รูปที่ 4.9 ส่วนรับข้อมูลจากพอร์ตอนุกรม 51
รูปที่ 4.10 ส่วนส่งข้อมูลออก Data Bus 52
รูปที่ 4.11 ส่วนควบคุมการเขียน SRAM 52
รูปที่ 4.12 ส่วนควบคุมการอ่าน SRAM และการแสดงผล 53
รูปที่ 4.13 ส่วนควบคุมบัส 54
รูปที่ 4.14 บล็อกไดอะแกรมโดยรวมของระบบทั้งหมด 55
รูปที่ 4.15 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter) 56
รูปที่ 4.16 โครงสร้างภายใน TDA8708A 57
รูปที่ 4.17 วงจรภายใน FPGA ส่วนต่างๆ 58
รูปที่ 4.18 ส่วนสร้างสัญญาณควบคุมการเชื่อมต่อบัส 59
รูปที่ 4.19 ส่วนสร้างสัญญาณ Sampling และเขียนข้อมูลวีดีโอลงหน่วยความจำ SRAM 60
รูปที่ 4.20 แสดง Block Ram ที่ใช้ในการออกแบบ 61
รูปที่ 4.21 ส่วนอ่านข้อมูลและควบคุมการแสดงผล 61
รูปที่ 4.22 ส่วนเชื่อมต่อบัสภายในกับบัสภายนอก 63

รูปที่ 5.1 ผลการ Simulation จากโปรแกรม VHDL 64
รูปที่ 5.2 แสดงขนาดความกว้างของสัญญาณต่างๆ 65
รูปที่ 5.3 แสดงส่วนของสัญญาณ Hor_Sync และ สัญญาณ Sampling 66
รูปที่ 5.4 แสดงส่วน Blanking ก่อนที่จะมีสัญญาณ Sampling 66
รูปที่ 5.5 แสดงส่วนของสัญญาณ Sampling 67
รูปที่ 5.6 แสดงส่วนของสัญญาณข้อมูลที่ได้จากการ Sampling 67
รูปที่ 5.7 แสดงส่วนขยายของสัญญาณ Sampling และสัญญาณข้อมูลที่ได้ทำการแปลงเป็นสัญญาณดิจิตอล 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 5.8 แสดง Timming ของการเขียนข้อมูลลง SRAM แถวที่ 1 69

รูปที่ 5.9 แสดง Timming ของการเขียนข้อมูลลง SRAM แถวที่ 1 – 11 69

รูปที่ 5.10 แสดง Timming ของการอ่านข้อมูลSRAM แถวที่ 1 และทำการ Shift ข้อมูลไปที่บอร์ดแสดงผล 70

รูปที่ 5.11 แสดง Timming ของการอ่านข้อมูลSRAM แถวที่ 1 กับ 2 และทำการ Shift ข้อมูลไปที่บอร์ดแสดงผล 70

รูปที่ 5.12 แสดง Timming ของการ Scan หลังจากการ Shift ข้อมูลครบทุกบอร์ด 71

รูปที่ 5.13 ตัวอย่างการแสดงผลการทดลองกับ Hardware จริง 72

รูปที่ 5.14 แสดง Timing ของการ Scan 72

รูปที่ 5.15 แสดง Timing ของการ Scan 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันการติดต่อสื่อสารทำได้ง่ายโดยอาศัยเทคโนโลยีของอุปกรณ์สื่อสารที่พัฒนาขึ้นเรื่อยๆ เพื่ออำนวยความสะดวกในการติดต่อสื่อสาร สิ่งเหล่านี้ได้เข้ามามีบทบาทในชีวิตประจำวันของผู้คนในปัจจุบันมากขึ้น โดยเฉพาะด้านธุรกิจโฆษณา ที่ต้องการสื่อที่สามารถแสดงข่าวสารได้อย่างครบถ้วน และมีรูปแบบที่ดึงดูดความสนใจพร้อมทั้งสื่อสารข้อมูลให้ทุกคนเข้าใจได้อย่างชัดเจน

แผ่นป้ายแสดงข่าวสาร (display board) เป็นอีกสื่อหนึ่งที่ได้รับความนิยมใช้กันมากในปัจจุบัน ซึ่งได้พัฒนาจากระบบการแสดงข้อมูลแบบเป็นเซเวนเซ็กเม้น (7 – Segment) มาเป็นการแสดงข้อมูลแบบจุดแสดงผล (Dot matrix display) ซึ่งให้ความละเอียดของภาพมากกว่าเดิม ซึ่งสามารถแสดงผลได้ทั้งตัวอักษรและรูปภาพ อีกทั้งสามารถปรับเปลี่ยนข้อมูลที่จะแสดงผลได้ตลอดเวลาซึ่งป้ายแสดงข่าวสารนี้จะมีขนาดเล็กหรือใหญ่ได้ตามความต้องการ สามารถเห็นได้อย่างชัดเจน จึงเป็นที่นิยมใช้ในปัจจุบัน ดังจะเห็นได้จากแหล่งชุมชน, ย่านธุรกิจ ,ศูนย์การค้า และห้างสรรพสินค้า

ความมุ่งหมายของโครงงาน

ในโครงงานนี้จะเป็นการพัฒนารูปแบบของการแสดงข้อมูลบนบอร์ดแสดงผลแบบจุด (Dot Matrix Display Board) ซึ่งใช้ LED ชนิด Dot Matrix ขนาด 8x8 มาใช้ในการสร้างบอร์ดแสดงผล ซึ่งมีขนาดความละเอียด 176 × 256 จุด การแสดงผลบนบอร์ดจะใช้ คอมพิวเตอร์ส่วนบุคคล (PC) ส่งข้อมูลภาพในรูปแบบของสัญญาณ อนาล็อก ส่งผ่านทางสาย VGA ของคอมพิวเตอร์ โดยข้อมูลภาพที่ส่งไปจะมีลักษณะเป็น สัญญาณ ANALOG ดังนั้นเพื่อที่จะให้ได้ข้อมูลแบบ DIGITAL เราจึงต้องทำการแปลงสัญญาณ ผ่านA/D เพื่อให้ได้สัญญาณ ดิจิตอลตามต้องการ จากนั้นจึงใช้ FPGA (Fields Programmable Gate Array) ในการควบคุมการทำงานส่วนต่างๆ ที่เป็นดิจิตอล ของวงจรแปลงสัญญาณภาพ และวงจรจัดเรียงข้อมูลภาพ โดยการใช้ภาษา VHDL เขียนอธิบายโปรแกรมการทำงาน แล้วจึงทำการโปรแกรมลงภายในชิพ FPGA ซึ่งในการสร้างวงจรที่ใช้ FPGA สามารถที่จะลดอุปกรณ์ลงได้มาก ทำให้ต้นทุนในการออกแบบต่ำ

ส่วนประกอบทางฮาร์ดแวร์ที่สำคัญของโครงงานนี้ ได้แก่

ส่วนแสดงผล (Display Board)

ส่วนตัวขับกระแส (Driver Board)

ส่วนควบคุมและพักข้อมูล (Control / Latch Data Board)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ของโครงงาน

1. เพื่อให้มีความรู้เกี่ยวกับการแสดงผลโดยใช้ LED
2. สามารถเรียนรู้การทำงาน ตลอดจนการใช้งาน FPGA
3. สามารถควบคุมการออกแบบ โดยใช้ภาษา VHDL เพื่อใช้งานกับ FPGA ได้
4. สามารถนำข้อมูลภาพมาแสดงออกทาง LED DISPLAY BOARD
5. สามารถออกแบบวงจรดิจิตอลพื้นฐาน เพื่อนำไปสู่การพัฒนาขั้นสูงต่อไป

ขอบเขตของโครงงาน

1. ศึกษาลักษณะสัญญาณภาพที่ใช้ในการจัดเก็บ เพื่อใช้ในการออกแบบ
2. ใช้ FPGA ในการควบคุมการทำงานของวงจรจับสัญญาณภาพ
3. ใช้ FPGA ในการควบคุมการทำงานในส่วนของการแสดงภาพบน LED DISPLAY BOARD
4. ใช้ภาษา VHDL ในการออกแบบการบรรยายเชิงพฤติกรรมการทำงานของวงจรดิจิตอล
5. ทำการจัดเก็บข้อมูลภาพที่ได้จาก FPGA มาเก็บไว้ใน SRAM ก่อนที่ส่งข้อมูลไปยัง LED DISPLAY BOARD

ประโยชน์ที่ได้รับจากโครงงาน

1. สามารถพัฒนาความรู้ที่ได้จากการเรียนไปประยุกต์ใช้งานจริงในการประดิษฐ์บอร์ดแสดงผล LED
2. สามารถใช้งานบอร์ดแสดงผล LED ในการสื่อสารได้จริง
3. เข้าใจถึงทฤษฎีต่างๆในการออกแบบวงจรดิจิตอล
4. เข้าใจลักษณะสัญญาณของภาพ และวิธีการจัดเก็บสัญญาณภาพ
5. เข้าใจการเขียนโปรแกรม VHDL ในการออกแบบ FPGA
6. เข้าใจหลักการออกแบบแผ่น PCB (print circuit board) เพื่อที่จะให้วงจรที่ออกแบบมีสัญญาณรบกวนน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

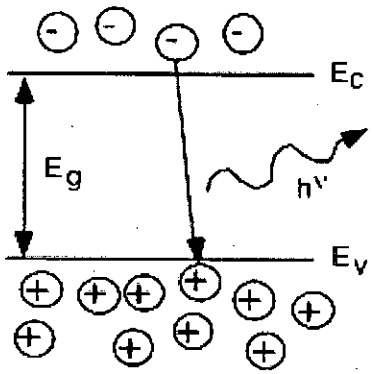
ในโครงงานนี้ประกอบด้วย ส่วนควบคุมการแสดงผล และส่วนการแสดงผล โดยส่วนควบคุมการแสดงผลจะมีชิพ FPGA เป็นส่วนประกอบที่สำคัญและส่วนการแสดงผลจะมี LED เป็นส่วนประกอบสำคัญ ดังนั้นจึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติและการทำงานต่างๆดังนี้

2.1 ไดโอดเปล่งแสง (Light Emitting Diode: LED)

LED ย่อจาก Light emitting diodes มีให้เห็นได้ทั่วไปในอุปกรณ์อิเล็กทรอนิกส์ บางครั้งคุณเห็นได้ในนาฬิกาดิจิตอล รีโมทคอนโทรล หน้าปัดอุปกรณ์ไฟฟ้า โทรทัศน์จัมโบ้ หรือแม้แต่ไฟจราจรตามสี่แยกเป็นต้น ที่จริงแล้วหลอด LED คือหลอดไฟขนาดเล็ก แต่มีหลักการทำงานแตกต่างจากหลอดไฟมีไส้ เพราะว่าไม่มีการเผาไส้หลอด ดังนั้น หลอด LED จึงไม่เกิดความร้อน แสงสว่างเกิดขึ้นจากการเคลื่อนของอิเล็กตรอนภายในสารกึ่งตัวนำ ซึ่งเป็นวัสดุแบบเดียวกับที่ใช้ในการทำทรานซิสเตอร์

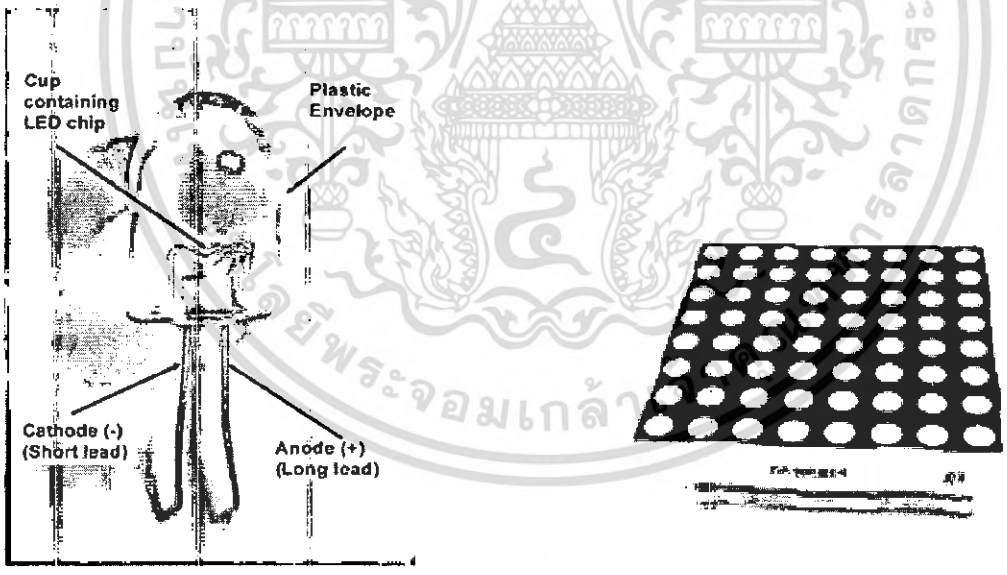
ไดโอดเปล่งแสงเป็นสิ่งประดิษฐ์สารกึ่งตัวนำชนิดหนึ่ง ที่สามารถแปลงพลังงานไฟฟ้าเป็นพลังงานแสง การเปล่งแสงเกิดขึ้นเมื่อพาหะดายหรือปลดปล่อยพลังงานออกมาเพื่อลดระดับลงไปอยู่ที่ระดับพลังงานต่ำกว่า การปลดปล่อยพลังงานนี้เรียกว่า "Band to band Transition" จะมีโอกาสเกิดได้สูงในกรณีที่สารกึ่งตัวนำนั้นเป็น สารกึ่งตัวนำชนิด "Direct band gap semiconductor" เช่น GaAs แสงเกิดขึ้นจากพลังงานที่ปลดปล่อยจากอะตอม แสงเป็นโฟตอนที่มีพลังงานและโมเมนตัม ดังนั้นจึงเป็นอนุภาคชนิดหนึ่ง ภายในอะตอม อิเล็กตรอนโคจรรอบนิวเคลียส และมีวงโคจรหลายวง แต่ละวงมีพลังงานแตกต่างกัน วงนอกมีพลังงานมากกว่าวงใน ถ้าอะตอมได้รับพลังงานจากภายนอก อิเล็กตรอนจะกระโดดจากวงโคจรในออกสู่วงโคจรนอก ในทางกลับกัน ถ้าอิเล็กตรอนกระโดดจากวงโคจรนอกเข้าสู่วงโคจรใน มันจะปลดปล่อยพลังงานออกมา และพลังงานนี้ก็คือแสงนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



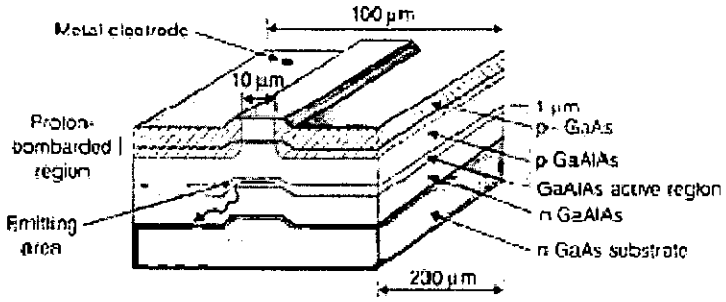
รูปที่ 2.1 แสดงการปลดปล่อยพลังงานออกมาในรูปของโฟตอน

ขณะที่อิเล็กตรอนเคลื่อนที่ผ่านรอยต่อไปที่โฮลของสาร P อิเล็กตรอนจะตกจากวงโคจรสูงหรือแถบนำไฟฟ้า ไปสู่วงโคจรต่ำหรือแถบวาเลนซ์ มันจะปลดปล่อยพลังงานออกมาในรูปของโฟตรอน ปรากฏการณ์นี้เกิดขึ้นกับไดโอดทุกชนิด แต่ที่คนสามารถเห็นแสงได้ก็ต่อเมื่อ ความถี่ของพลังงานอยู่ในช่วงความถี่ที่ตามองเห็นได้ ดังเช่นไดโอดที่ทำจากซิลิคอน ซึ่งมีช่วงของแถบพลังงานแคบ ทำให้ได้โฟตอนความถี่ต่ำ เป็นความถี่ที่ตามองเห็นได้ อย่างไรก็ตาม ความถี่ที่ตามองไม่เห็นก็มีประโยชน์ไม่น้อย ยกตัวอย่างเช่น ช่วงอินฟราเรด สามารถนำไปใช้ในเครื่องควบคุมระยะไกลหรือรีโมทคอนโทรล เป็นต้น

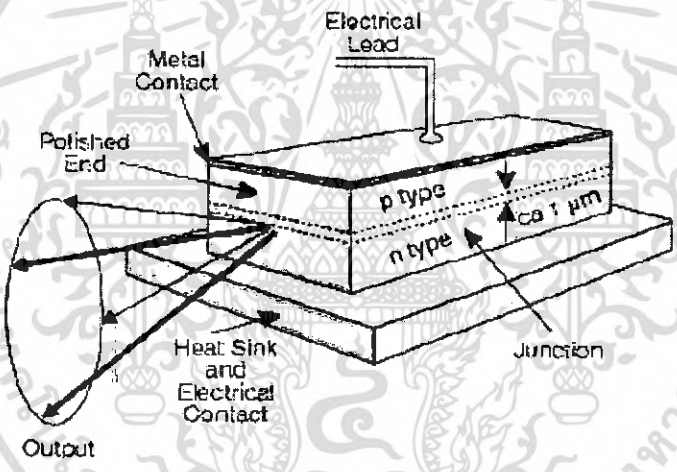


รูปที่ 2.2 แสดงลักษณะของ LED ในรูปแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

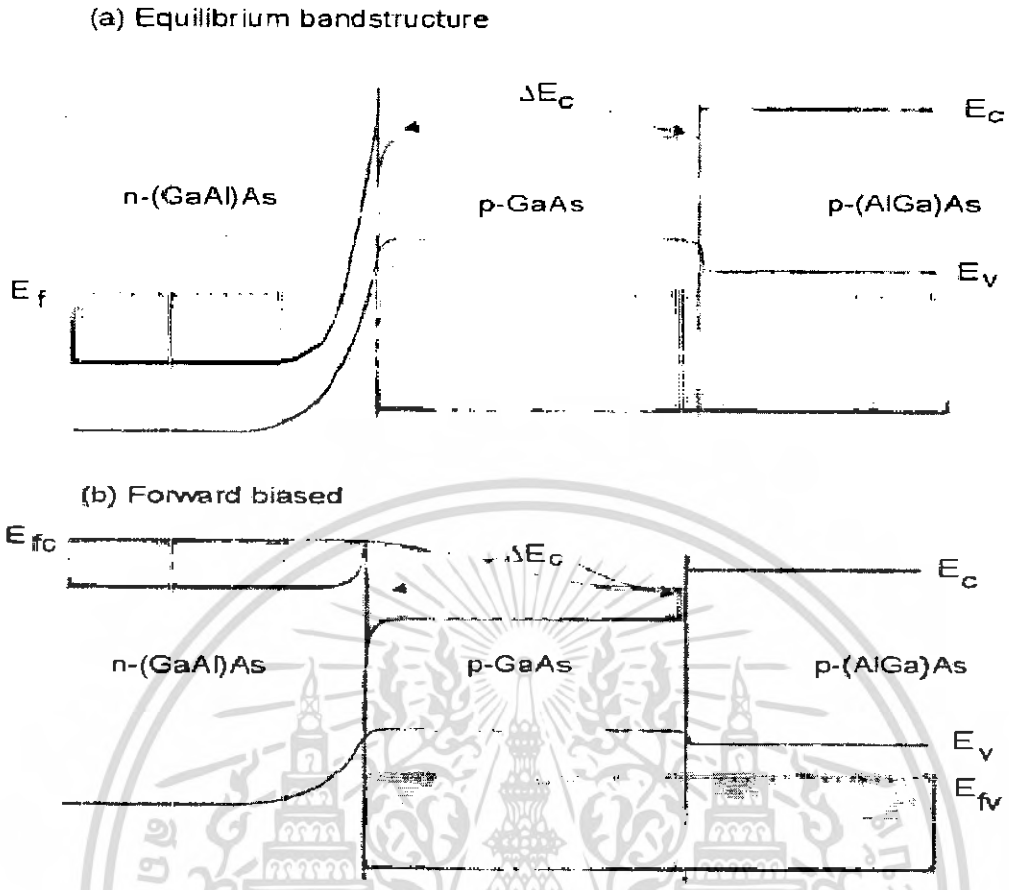


รูปที่ 2.3 แสดงโครงสร้างภายในของ LED



รูปที่ 2.4 แสดงลักษณะการแปล่งแสงของ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงแถบพลังงานของ LED (a) เมื่อไม่มีการป้อนศักดาไฟฟ้าให้ LED (b) เมื่อมีการจ่ายศักดาไฟฟ้าให้ LED

เงื่อนไขที่สำคัญของสารกึ่งตัวนำที่จะมาทำเป็น LED ที่เปล่งแสงในย่านที่ตามองเห็นมี 3 ประการ คือ

1.) ต้องเป็นสารกึ่งตัวนำประเภท Direct band gap

2.) ขนาดของ E_G ต้องอยู่ระหว่าง 1.77eV - 3.10eV

3.) สามารถทำเป็นโครงสร้างของรอยต่อ P-N ได้

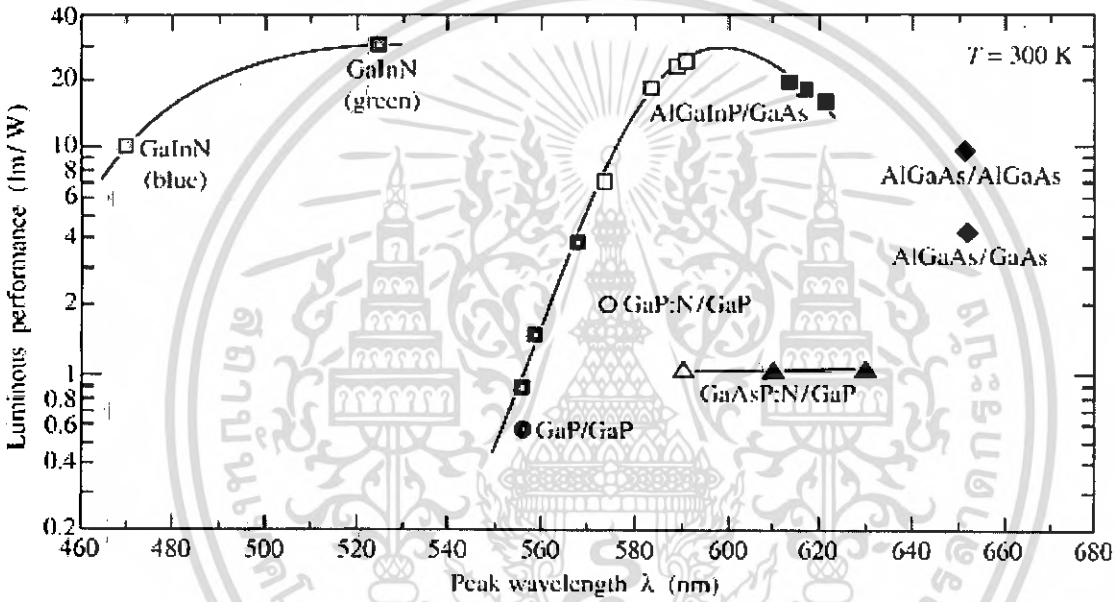
จากเงื่อนไขข้างต้น LED ที่ผลิตขายกันมักสร้างมาจากสารกึ่งตัวนำประเภทอัลลอย ประสิทธิภาพของไดโอดเปล่งแสงด้วยค่า η ซึ่งเรียกว่า "External efficiency" ถูกนิยามด้วยอัตราส่วนระหว่าง กำลังงานของแสงด้านเอาท์พุตกับกำลังงานด้านอินพุต

การวัดความเข้มแสงแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ แบบเรดิโอมีทรี (Radiometry) และแบบโฟโต้มีทรี (Photometry) โดยการวัดแบบเรดิโอมีทรีเป็นการวัดพลังงานที่เปล่งออกมาทุกความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยาวคลื่นแสงในขณะที่การวัดแบบโฟโต้มีทรีจะวัดความสว่างที่ปรากฏที่มนุษย์สามารถมองเห็นได้ด้วยตาเท่านั้น โดยทั่วไปดาคนเราจะมองเห็นได้ดีในช่วงความยาวคลื่นแสง (light wavelengths) จาก 380nm – 740nm

หน่วยวัดความเข้มแสงนั้นในปัจจุบันมีอยู่ด้วยกันหลายหน่วย แต่ที่พบมากคือ หน่วยลูเมนส์ (Lumens) โดยหน่วยนี้จะวัดคลื่นแสงทุกความยาวคลื่น อีกหน่วยคือ หน่วยแคนเตลา (candelas) หรือ cd หน่วยนี้วัดเฉพาะความเข้มแสงที่ตามองเห็นได้เท่านั้น ดังนั้นโดยทั่วไปสำหรับการวัดความเข้มแสงของจอแสดงผลใดๆจึงมักใช้หน่วย cd หรือ mcd ในการบอกถึงปริมาณของแสงที่ผู้ใช้หรือผู้มองภาพจะสามารถมองเห็น



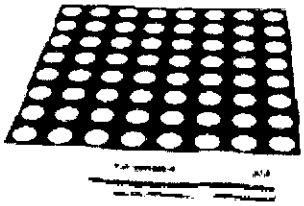
รูปที่ 2.6 แสดงการมองเห็นแสงของ LED ในย่านความยาวคลื่นต่างๆ

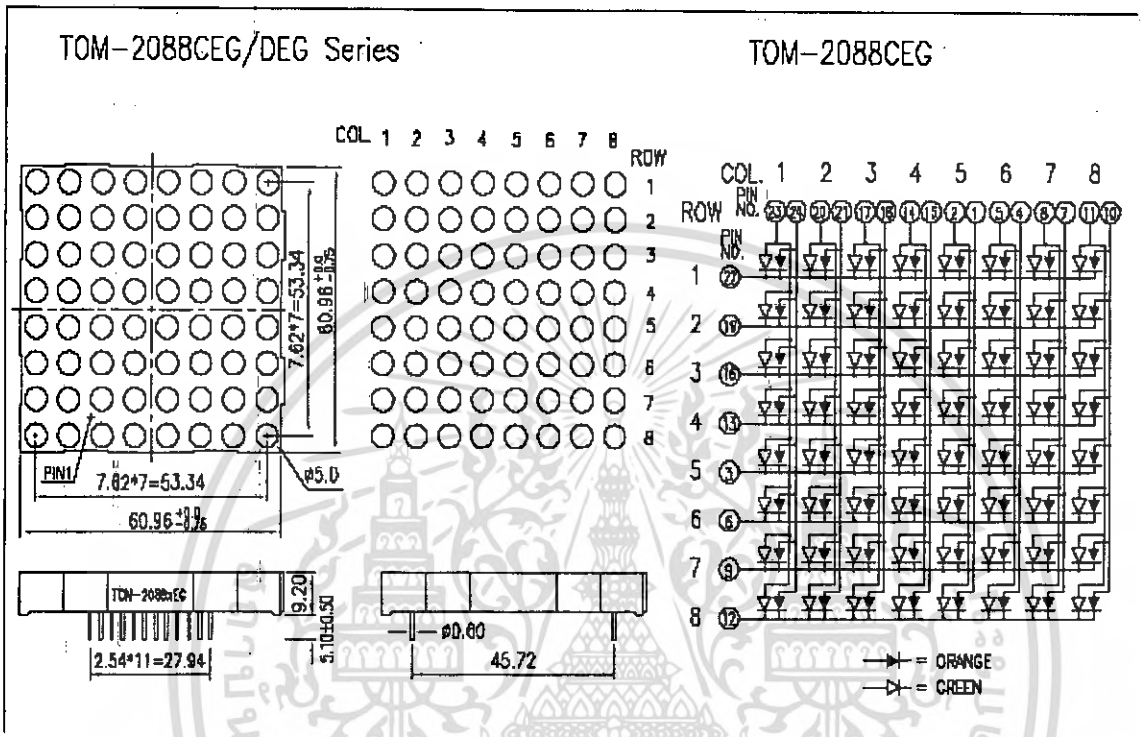
ในการผลิตโมดูล LED เหล่านี้ LED ชิปหลายๆตัวจะถูกนำมาติดตั้งบนลีดเฟรมเดียวกัน หรือไม่ก็ใช้เป็นวงจรพิมพ์ (PCB) โดย LED แบบโมดูลที่ผลิตขึ้นมามี 2 ประเภทให้เลือกใช้งาน คือ แบบ CA (common Anode) หรือแบบ CC (common Cathode)

สำหรับในโครงงานนี้จะใช้ LED ชนิด DOT MATRIX LED แบบ 3 สี ซึ่งมีคุณสมบัติพาะดังนี้

- เป็นหลอด LED ที่มีความสว่างน้อย ประมาณ 20 mcd.(มิลลิแคนเดลลา)
- เป็นหลอด LED ขนาด 8 จุด × 8 จุด ใน 1 ก้อน มีเส้นผ่าศูนย์กลาง 5 มม. และมีระยะห่างระหว่างดวงไฟ 7.62 มม.
- เปล่งแสงได้ 3 สี ได้แก่ สีแดง, สีเขียว และสีส้ม
- มีมุมมองของแสงมากกว่า 120 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.7 แสดงรูป DOT MATRIX LED ที่นำมาใช้เป็น Display Board ในโครงงาน

การขับ LED ด้วยวิธีแบบสแตติกและแบบมัลติเพล็กซ์

ในการขับ LED พร้อมกันหลายๆตัว อย่างเช่น โมดูล LED มี 2 แบบ คือ

1.) แบบสแตติก (Static drive) เป็นวิธีที่ง่ายโดยการขับ LED แต่ละดวงจะแยกอิสระจากกันโดยการต่อตัวต้านทานตัวหนึ่งเข้ากับแหล่งจ่ายแรงดันหรือแหล่งจ่ายกระแสคงที่ มาทำการป้อนฟอร์เวิร์ดไบแอสให้กับ LED LED ก็จะติดสว่างได้ตามต้องการ เราเรียกว่า static เพราะว่ามีกระแสไหลผ่าน LED อย่างต่อเนื่องตลอดเวลา เหมาะสำหรับการขับ LED ไม่กี่ตัว LED ทั่วไปมีความสว่างปกติต้องการกระแสประมาณ 2 mA ซึ่งเอาท์พุตของไมโครคอนโทรลเลอร์ส่วนใหญ่จะสามารถขับกระแสนี้ให้กับ LED ได้โดยตรง

2.) แบบมัลติเพล็กซ์ (Multiplex drive) หรือการขับแบบพัลส์ (pulse drive) เป็นการลดความยุ่งยากของวงจรโดยทำการป้อนกระแสไฟไบแอสให้กับ LED ในแต่ละเซกเมนต์เพียงกลุ่มหนึ่ง (เพื่อให้แสดงผลสมบูรณ์ในหน่วยนั้นๆ) ในแต่ละช่วงเวลาสั้นๆ แล้วจึงค่อยเลื่อนไปไบแอสให้กับ

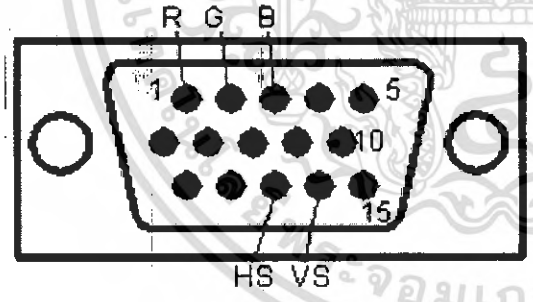
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED เซกเมนต์ในกลุ่มต่อไป อย่างไรก็ตามการไบแอสแบบให้ดวงไฟติดสว่างในระยะเวลาสั้นๆ ต้องมีคาบในการติดสว่างซ้ำ (repetition rate) ที่รวดเร็วเพียงพอจนสายตามนุษย์มองเห็นเหมือนแสงสว่างนั้นติดตลอดเวลา เนื่องจากตาคนเรานั้นไม่สามารถแยกแยะการกระพริบของไฟที่รวดเร็วมากๆได้ อีกประการหนึ่งคือตาคนเรามักจะอ่านและจำในตำแหน่งที่หลอดไฟดวงนั้นสว่างมากที่สุด ดังนั้นถ้ามีการกระพริบที่เร็วพอเราจะเห็นความสว่างของหลอดไฟนั้นติดต่อเนื่อง โดยมาจากการเฉลี่ยค่าความสว่าง นั่นหมายความว่าถ้าเราป้อนพัลส์ที่มีค่าดิวตี้ไซเกิล (duty cycle) ต่ำ แต่มีความเข้มแสงสูงเมื่อทำการมองเห็นหลอดไฟติดสว่างมากกว่า เมื่อเทียบกับการป้อนกระแสตรง ดังนั้นประโยชน์อีกข้อหนึ่งของการขับ LED แบบมัลติเพล็กซ์ คือ การปรับปรุเรื่องความเข้มแสงในการแสดงผลของจอแสดงผลในขณะที่สิ้นเปลืองพลังงานโดยเฉลี่ยต่ำกว่าหรือเทียบเท่ากัน

2.2 ทฤษฎีเบื้องต้น เกี่ยวกับการทำงานของจอ VGA

สำหรับจอ VGA นั้นต้องการสัญญาณ 5 สัญญาณเพื่อที่จะนำมาควบคุมการแสดงผลดังนี้คือ

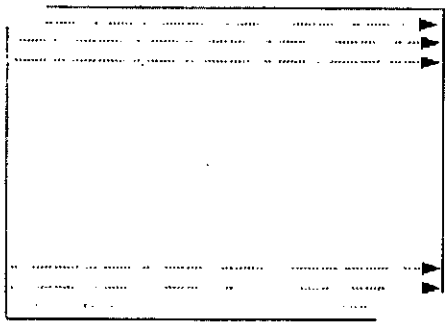
1. สัญญาณสีต่าง ๆ 3 สีด้วยกันคือ Red (R) Green (G) และ Blue (B)
2. สัญญาณควบคุมการแสดงผล 2 สัญญาณ ได้แก่สัญญาณซิงค์ในด้านแนวนอน HS (Horizontal Synchronization) และสัญญาณซิงค์ในด้านแนวตั้ง VS (Vertical Synchronization) โดยสัญญาณต่าง ๆ เหล่านี้จะอยู่ในตำแหน่งที่ VGA connector ต่าง ๆ กันดัง**รูปที่ 2.8**

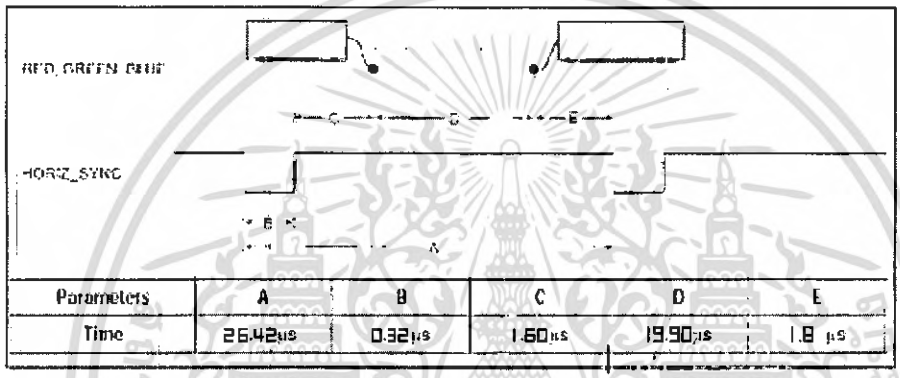


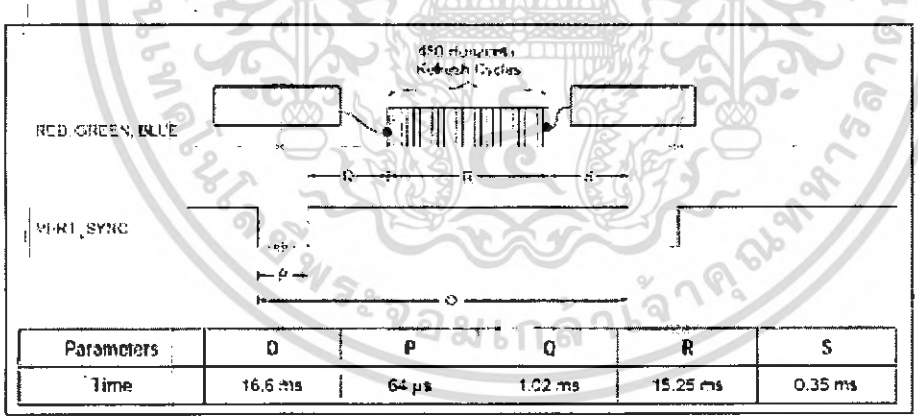
รูปที่ 2.8 VGA connector ตัวผู้ที่มองออกมาจากสายเคเบิลของจอมอนิเตอร์

ในการแสดงผลของจอมอนิเตอร์นั้นจะเป็นมาตรฐานเดียวกันคือ มีการแสดงทีละเส้น line-by-line และจากบนลงล่าง top-to-bottom ในแต่ละเส้นนั้นจะมีการกวาดจากซ้ายไปขวา left-to-right ดังนั้นเราจึงสามารถกำหนดการแสดงในแต่ละช่วงของการกวาดให้เป็นลักษณะของ coordinate x-y โดยใช้สัญญาณ Horizontal และ vertical synchronization (HS และ VS) เป็นตัวกำหนด (reference) โดยที่สัญญาณ HS จะเป็นตัวบอกช่วงที่สัญญาณทำการกวาดในแนวนอน (x-coordinate) และ VS จะเป็นตัวบอกว่าตอนนี้ทำการกวาดในเส้นที่เท่าไหร่ (y-coordinate) ดู**รูปที่ 2.9**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







รูปที่ 2.9 แสดงการทำงานของจอ VGA โดยใช้สัญญาณ HS และ VS

2.3 การจัดเก็บสัญญาณภาพ

สัญญาณภาพจากจากสัญญาณสี จะมีลักษณะเป็นสัญญาณอนาล็อกเข้าไปเก็บไว้ในหน่วยความจำ ซึ่งมีความจำเป็นอย่างยิ่งที่จะต้องทำการแปลงสัญญาณภาพดังกล่าว จากสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล เพื่อให้สามารถที่จะจัดเก็บเข้าไปยังหน่วยความจำภายใน FPGA ได้ ในการแปลงสัญญาณอนาล็อกนั้นจะต้องประกอบด้วย ส่วนของการสุ่มตัวอย่างสัญญาณ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควอนไตซ์สัญญาณ (Quantization) หรือการจัดระดับสัญญาณ ซึ่งการจัดระดับสัญญาณนั้นกระทำโดยการนำสัญญาณที่ผ่านการสุ่มสัญญาณแล้ว นำมาทำการแปลงให้เป็นสัญญาณดิจิตอล โดยสัญญาณภาพที่ถูกเปลี่ยนแปลงให้เป็นสัญญาณดิจิตอลแล้ว สามารถที่จัดเก็บลงในหน่วยความจำได้ แต่ในการสแกนทางแนวนอนของสัญญาณภาพนั้น จะใช้เวลาในการสแกนที่สั้นมาก ดังนั้นจึงทำให้เกิดปัญหาในขั้นตอนของการเปลี่ยนสัญญาณภาพจากสัญญาณอนาลอก ให้เป็นสัญญาณดิจิตอล (Analog to Digital Converter) รวมทั้งการเขียนข้อมูลเข้าไปเก็บยังหน่วยความจำ

ปัญหาแรกคือ ปัญหาในเรื่องการแปลงสัญญาณอนาลอกเป็นดิจิตอล การที่สัญญาณภาพมีเวลาที่ใช้ในการสแกนทางแนวนอนที่สั้นมาก ในการจัดเก็บเพื่อให้ความละเอียดของภาพคงเดิม ดังเช่นที่แสดงทางหน้าจอมอนิเตอร์ จำเป็นจะต้องใช้วงจรแปลงสัญญาณ อนาลอกเป็นดิจิตอลที่มีอัตราในการสุ่มที่สูงเพียงพอกับสัญญาณดังกล่าว

ปัญหาที่สองนั้น เกิดขึ้นจากข้อมูลที่ถูกแปลงแล้วจะนำไปเขียนในหน่วยความจำซึ่งในการเขียนข้อมูลดังกล่าวนี้จะต้องมีการจัดเวลาอย่างเหมาะสมเพื่อไม่ให้เกิดปัญหาขึ้นในขั้นตอนของการเขียนข้อมูลลงหน่วยความจำ ในการจัดเก็บข้อมูลลงในหน่วยความจำนั้นจะอาศัยหลักการจัดเก็บภาพลงสู่หน่วยความจำแบบแอดเดรสเป็นแบบต่อเนื่อง

ถ้าหากเป็นสัญญาณภาพที่ได้รับจากสัญญาณ VGA ขนาด 600 × 800 จุดจะพบว่าใน 1 เส้นสแกนทางแนวนอนจะใช้เวลาทั้งสิ้น 26.6 ไมโครวินาที แต่เนื่องจากส่วนที่เป็นสัญญาณภาพจริงๆ จะประมาณ 22 ไมโครวินาที ส่วนที่เหลือจะเป็นสัญญาณในช่วงแบล้งกิ้ง และถ้าหากต้องการที่จะทำการจัดเก็บสัญญาณในแต่ละเส้นสแกนทางแนวนอนให้มีจุดภาพทั้งสิ้น 256 จุดภาพ ก็จำเป็นต้องใช้เวลาในการเขียนข้อมูลภาพแต่ละจุดลงหน่วยความจำ เท่ากับ

เวลาในการเขียนข้อมูลแต่ละจุดภาพ = 22 μs/256

= 80 ns

เวลาดังกล่าวเป็นค่า ACESS TIME ของหน่วยความจำที่นำมาใช้ โดยค่าความจุของหน่วยความจำที่ใช้จะมีค่าเท่ากับผลคูณของจำนวนเส้นสแกนต่อหนึ่งเฟรมกับจำนวนจุดภาพในหนึ่งเส้นสแกนทางแนวนอน คือ

ความจุของหน่วยความจำ = จำนวนเส้นสแกนในแต่ละบอร์ด × จำนวนจุดภาพในหนึ่งเส้นสแกนทางแนวนอน

= 16×256

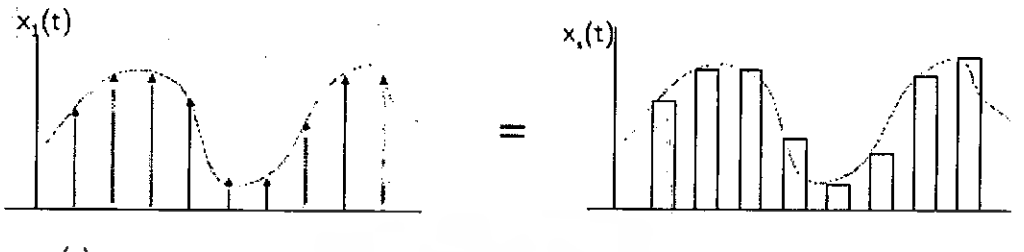
= 4096 Byte

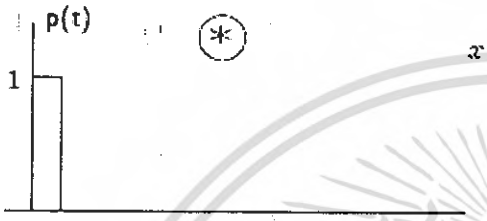
แต่ในกรณีที่มีสัญญาณสีสองสีเข้ามาพร้อมกัน ความจุของหน่วยความจำจะต้องเพิ่มเป็น 2 เท่า ซึ่งในที่นี้ก็คือ 8 KByte

สำหรับเวลาในการ Sampling สัญญาณวีดีโอจะใช้เวลาในการ Sampling เท่ากับเวลาในการเขียนข้อมูลลงในหน่วยความจำ นั่นคือ 80 ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสุ่ม (Sampling)





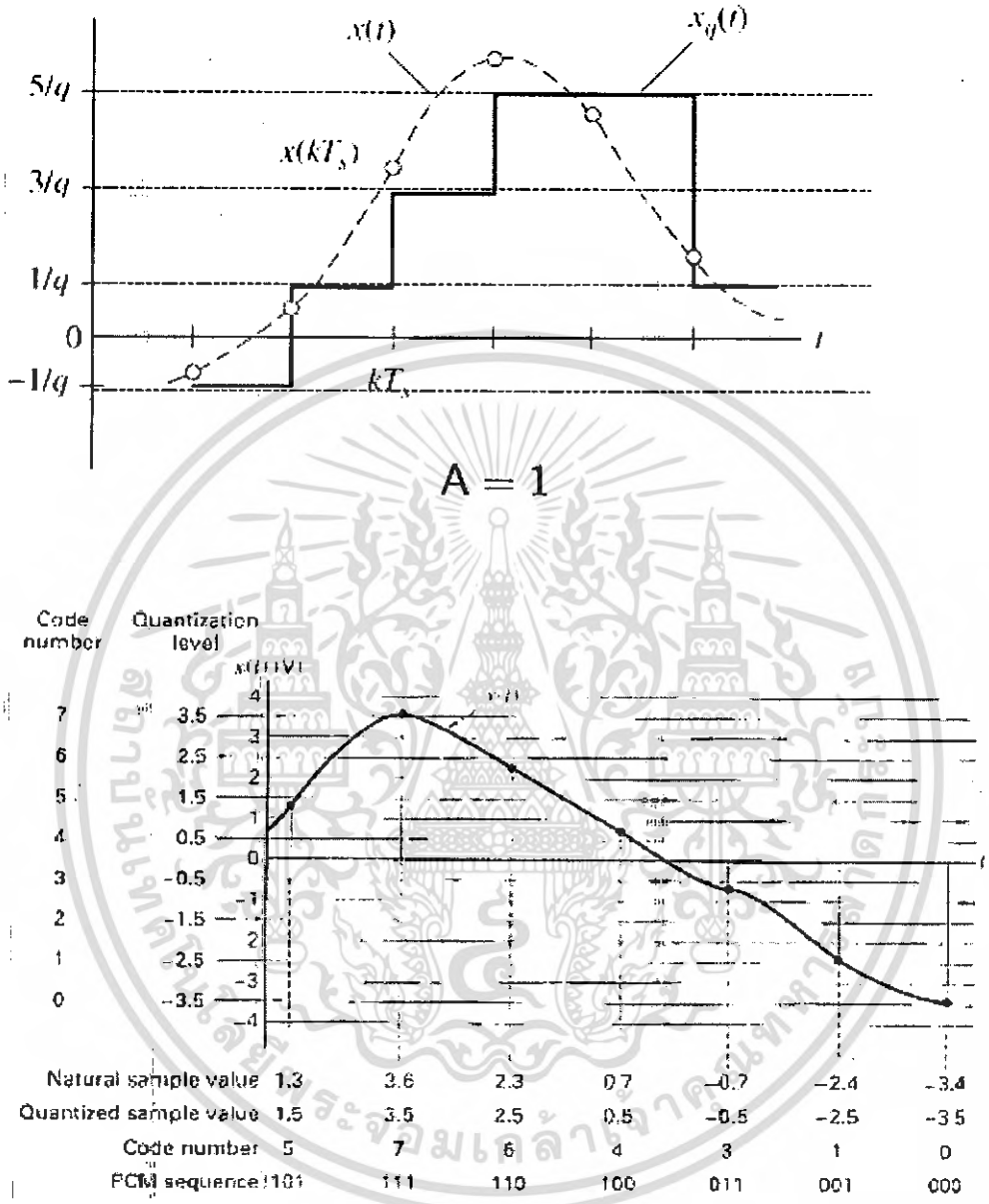
$$
\begin{aligned}
x_s(t) &= p(t) * \delta_1(t) \\
&= p(t) * x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \\
&= p(t) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \\
&= \sum_{n=-\infty}^{\infty} x(nT_s)p(t - nT_s)
\end{aligned}
$$

รูปที่ 2.10 แสดงฟังก์ชันของการสุ่ม

ความถี่ที่ใช้ในการสุ่มในระบบภาพนั้นจะเป็นตัวบ่งบอกถึงขนาดภาพ ภาพที่ผ่านการสุ่มด้วยความถี่สูงๆก็จะได้จุดภาพที่ละเอียดมากขึ้น สามารถเก็บรายละเอียดได้มากขึ้นในกรณีที่จุดภาพมีค่าน้อยจะทำให้เกิดผลอย่างหนึ่งคือ การเกิดซ้ำกันของจุดภาพ (Pixel Replication) ทำให้เห็นภาพเป็นบล็อกๆ (Checker-Board Effect)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การควอนไตซ์ (Quantization)



รูปที่ 2.11 แสดงวิธีการควอนไตซ์สัญญาณ

การควอนไตซ์เป็นการเข้ารหัสของระดับที่ผ่านการสุ่ม เพื่อจัดเข้าระดับที่เป็นมาตรฐานหรือเป็นไปตามที่ต้องการ แต่ในทางด้านการประมวลผลภาพการควอนไตซ์เป็นการจัดระดับของสัญญาณภาพที่ผ่านการสุ่มให้อยู่ในระดับเทา จำนวนระดับเทาที่ใช้นั้นเท่ากับสองยกกำลังตามจำนวนบิต แสดงตามสมการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = 2^m$$

เมื่อ G เท่ากับจำนวนระดับเทา และ m เป็นจำนวนบิตข้อมูลที่ใช้ ตัวอย่างเช่น ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลที่ให้ข้อมูลจากการแปลงแล้ว 8 บิต ทำให้ได้ระดับเทาที่แตกต่างกัน 256 ระดับ ระดับการควอนไตซ์นั้นจะมีผลต่อภาพที่เก็บ ถ้าใช้ระดับการควอนไตซ์ที่มีจำนวนระดับความแตกต่างน้อยหรือกล่าวอีกนัยหนึ่ง คือจำนวนบิตข้อมูลดิจิตอลที่น้อยกว่าปกตินั้นจะทำให้เกิดความผิดพลาดของข้อมูลสูง สาเหตุที่เป็นเช่นนี้เพราะว่าความห่างของระดับนั้นมีมาก เวลาทำการควอนไตซ์จะเกิดการปรับค่าที่ได้จากการสุ่มให้เข้าสู่ระดับที่กำหนด ถ้าข้อมูลที่ได้จากการสุ่มห่างจากระดับที่กำหนดมากเท่าใด ก็จะทำให้เกิดความผิดพลาดมากขึ้นเท่านั้นหรือกล่าวอีกนัยหนึ่ง คือเราไม่มีระดับเทาที่แทนค่าของระดับความเข้มของภาพได้ทั้งหมด ส่วนจำนวนระดับเทาหรือบิตของข้อมูลภาพที่ใช้นั้นปกติไม่ควรต่ำกว่า 64 ระดับเทา หรือจำนวนบิตไม่ควรต่ำกว่า 6 บิตจึงเหมาะสมกับสายตาของคนเรที่จะไม่รู้สึกว่ากิดความคลาดเคลื่อนขึ้นกับภาพ แต่ถ้าใช้จำนวนบิตที่ต่ำกว่านี้จะทำให้เกิดผลอย่างหนึ่งที่เรียกว่า "ขอบเทียม" (false contour) แม้มีการใช้บิดของจุดภาพที่น้อยลง แต่สายตาเราก็ยังไม่สามารถตรวจจับความแตกต่างของภาพได้ แต่ถ้ามีการลดจำนวนบิตของจุดภาพลงไปอีก จะทำให้เราสามารถจับความผิดเพี้ยนของภาพที่เกิดขึ้นได้

ในการกำหนดขนาดของภาพ และระดับเทาของภาพสำหรับวงจรเก็บข้อมูลภาพจะต้องพิจารณาให้เหมาะสมกับงานที่จะใช้ ซึ่งโดยรวมเป็นการกำหนดรายละเอียดของภาพ (Resolution of Image) ถ้ากำหนดภาพที่มีรายละเอียดสูงๆ ก็จะได้คุณภาพที่ดีแต่ทำให้ต้องใช้หน่วยความจำในการเก็บข้อมูลที่มีขนาดใหญ่มาก หรือถ้ากำหนดรายละเอียดของภาพต่ำก็จะสามารถใช้หน่วยความจำขนาดเล็กแต่อาจไม่ได้รายละเอียดเท่าที่ควร ฉะนั้นการออกแบบวงจรเก็บข้อมูลภาพควรให้มีความเหมาะสมในเรื่องคุณสมบัติของภาพที่ต้องการจัดเก็บด้วย

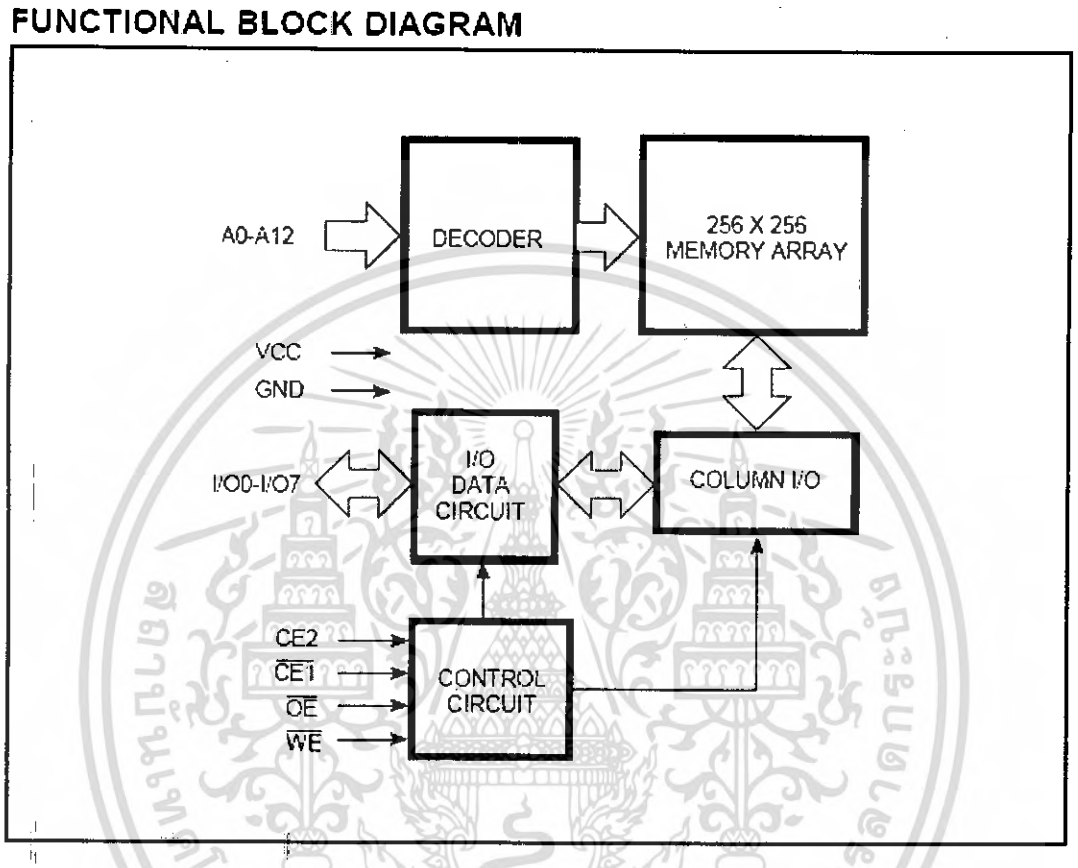
2.6 ลักษณะหน่วยความจำที่ใช้ในการเก็บข้อมูล

การจัดเก็บข้อมูลของสัญญาณภาพ ปัญหาที่พบทั้วสองปัญหา คือในเรื่องของ A/D และหน่วยความจำที่ใช้ในการเก็บข้อมูล สามารถทำการแก้ไขได้ คือการแปลงสัญญาณอนาลอกเป็นดิจิตอลควรจะใช้ความถี่ในการสุ่มตัวอย่าง (Sampling) ที่มีความถี่สูงเพื่อที่จะให้ความละเอียดของภาพสูง ดังนั้นในส่วนของ A/D จึงสามารถแก้ไขได้ โดยการใช้ชิพ A/D เบอร์ TDA8708A

ส่วนปัญหาในเรื่องของหน่วยความจำที่จะนำมาใช้ในการจัดเก็บข้อมูลภาพนั้นจะใช้หน่วยความจำแบบ Static RAM เบอร์ IS61C64AH ซึ่งเป็นหน่วยความจำที่มีค่า Access Time เท่ากับ 20 ns มีความจุเท่ากับ 8 KByte และเป็นการจัดเก็บข้อมูลลงหน่วยความจำแบบโดยตรง วิธีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นวิธีที่สะดวกและง่ายที่สุด เพราะจะใช้อุปกรณ์ที่ประกอบในวงจรน้อยที่สุดทำให้ง่ายต่อการออกแบบแก้ไข



รูปที่ 2.12 BLOCK DIAGRAM SRAM IS61C64AH

8K × 8 HIGH-SPEED CMOS STATIC RAM

2.7 องค์ประกอบของภาพ

หากเราตัดภาพจากหนังสือพิมพ์สักภาพหนึ่ง แล้วขยายขึ้นด้วยกล้อง หรือแว่นขยายจะพบว่าภาพมีองค์ประกอบมาจากจุดสีขาวและจุดสีดำมากมาย มาเรียงกันประกอบขึ้นเป็นภาพจุดเหล่านี้เองที่เรียกว่าองค์ประกอบของภาพ หรือพิคเจอร์ อีลีเมนต์ (Picture Element) หรือ พิกเซล (Pixel)

ทำนองเดียวกัน ภาพที่ปรากฏทางจอภาพก็เอามาจากหลักการนี้ ภาพที่เกิดขึ้นบนจอภาพหรือโทรทัศน์ประกอบด้วยเส้นขวางเล็กๆในแนวนอนเป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิท ส่วนที่จาง และส่วนที่สว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสงสแกนความแตกต่างกันบนเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองที่เราจัดว่าเป็นองค์ประกอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันส่วนที่เรียกว่าองค์ประกอบภาพได้ถูกนำไปใช้งานอย่างจริงจังมากขึ้น ในส่วนโทรทัศน์ หรือเครื่องเล่นวีดีโอคาสเซ็ทเร็คคอร์ดเดอร์ ซึ่งจะมีการนำพิกเซลเหล่านี้เก็บไว้ในหน่วยความจำ เพราะข้อมูลที่เป็นพิกเซลเท่านั้นที่ระบบดิจิตอลจะจัดการข้อมูลได้ เราจะพบวิธีการนี้ในโทรทัศน์ระบบดิจิตอล โทรทัศน์ระบบคอมพิวเตอร์ โทรทัศน์จอภาพแอลซีดี วีซีอาร์ ระบบภาพซ้อนภาพ เป็นต้น

ในปัจจุบันสำหรับโทรทัศน์ธรรมดาเราจะพบว่ามีการเพิ่มเส้นสแกนภาพให้มากขึ้น และแน่นอนว่าจำนวนพิกเซลย่อมมากขึ้นด้วย อย่างโทรทัศน์จอใหญ่หรือโทรทัศน์ที่ต้องการรายละเอียดสูง HDTV อาจจะต้องใช้เส้นสแกนภาพมากกว่า 625 เส้น เช่นที่นิยมใช้ในปัจจุบันคือ 725 เส้น หรืออย่างเครื่องฉายวีดีโอโปรเจกเตอร์จะต้องใช้เส้นภาพ 2,200 เส้นภาพ และหากเป็นจอใหญ่หลายร้อยนิ้วจะต้องเพิ่มรายละเอียดมากขึ้นอีก นั่นคือการเพิ่มพิกเซลอีลีเมนต์นั่นเอง

2.8 การสแกน (Scanning)

ภาพที่เราเห็นหรือตัวอักษรต่างๆในป้ายโฆษณานั้น จะประกอบด้วยจุดเล็กๆจำนวนหนึ่งที่ทำให้เกิดขึ้นโดยเส้นแนวนอนและเส้นแนวตั้ง โดยจะแบ่งออกเป็นส่วนๆส่วนละเท่าๆกัน จุดเหล่านี้ก็คือ LED หนึ่งดวงนั่นเอง การที่เราจะมองเห็นว่าเป็นภาพได้ ก็คือการควบคุมให้ LED สว่างติดตามต้องการ แต่เราไม่สามารถทำให้ LED เหล่านี้สว่างติดตลอดเวลาได้ ดังนั้นต้องใช้เทคนิคในการสแกน ปัญหาที่ยุ่งยากอันหนึ่งคือเราจำเป็นต้องควบคุมการดิดดับของ LED ให้สามารถมองเห็นเป็นภาพนิ่งหรือภาพที่เคลื่อนไหวได้ โดยปกติแล้ว LED ที่ใช้กันอยู่จะมีความเข้มของการส่องสว่างในระดับ มิลลิคาเลนดา (mcd) ซึ่งเป็นหน่วยการวัดความเข้มของการส่องสว่าง เช่น LED สีแดงให้ความส่องสว่าง 20 mcd ความส่องสว่างเอาต์พุตเมื่อป้อนกระแสตรง 10 มิลลิแอมป์(mA) จะได้เพียง 0.7 mcd ในขณะที่ป้อนเป็นพัลส์ที่มีกระแสสูงสุด 100mA ดิวตี้ไซเคิล 1/T = 1/10 วิ่งก็ได้กระแสเฉลี่ย 10 mA เช่นกันจะได้ความสว่างเอาต์พุตเฉลี่ย 2 mcd ซึ่งจะเห็นได้ว่าถ้าใช้กระแสพัลส์ จะได้กำลังแสงเอาต์พุตเพิ่มขึ้นเป็นสามเท่า เมื่อใช่กระแสเท่ากัน และการทำให้เห็นเป็นแสงต่อเนื่องได้จะต้องป้อนกระแสพัลส์ที่มีความถี่มากกว่า 30 Hz จึงสามารถทำให้ตาคนเรามองเห็นแสงอาต์พุต LED เป็นแสงต่อเนื่องได้ แต่ถ้าเราต้องการให้เป็นภาพต่อเนื่องหรือตัวอักษรวิ่งที่ไม่สามารถสังเกตเห็นการกระพริบได้เลย เราควรป้อนพัลส์ที่มีความถี่ประมาณ 50-60 Hz จึงจะไม่เห็นภาพเกิดการพริ้ว และหน้าที่ของการสแกนก็คือการเลือก LED ที่สว่างติดให้ได้ภาพที่ชัดเจนตามต้องการ การสแกนหมายถึง จำนวนเส้นการสแกนต่อหนึ่งภาพ และจำนวนภาพที่ส่งออกไปต่อวินาทีมากมายเท่าไร การกระพริบของภาพก็จะลดลงเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หลักของการสแกน

ในการทำให้ LED ติดดับ มีหลักการอยู่ที่การสแกน ซึ่งการสแกนสามารถทำได้ 2 วิธี คือการสแกนทางคอลัมน์ (Column) และ การสแกนทางโรว์ (Row)

1. การสแกนทางคอลัมน์

การสแกนทางคอลัมน์จะทำการข้อมูลออกไปทางโรว์โดยส่งข้อมูลตัวที่ 1 ออกไปแล้วให้คอลัมน์ที่ 1 แอกทีฟ (Active) จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไป แล้วให้คอลัมน์ที่ 2 แอกทีฟ ทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกคอลัมน์ก็จะเป็นการสแกนครบ 1 รอบ ดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษรมีหลายหลัก วิธีนี้จึงไม่เหมาะสมนักที่จะนำมาใช้งาน เพราะว่าเมื่อให้ LED ในคอลัมน์ที่ 1 ติด กว่าที่ LED ที่คอลัมน์สุดท้ายจะติดต้องใช้เวลานาน

2. การสแกนทางโรว์

ส่วนการสแกนทางโรว์จะทำการส่งข้อมูลออกไปครบทุกหลักก่อนแล้วให้โรว์ที่ 1แอกทีฟ จากนั้นก็ทำการส่งข้อมูลชุดถัดไปจนครบหมดทุกหลัก แล้วให้โรว์ที่ 2 แอกทีฟ ทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบทุกโรว์ ก็จะปืนการสแกนครบ 1 รอบ วิธีนี้มีข้อดีคือ สามารถ แสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลักและถ้าจัดเวลาให้เหมาะสมแล้ว เวลาทำการสแกนก็จะไม่เกิดอาการพลิ้ว แต่มีข้อเสีย คือ การเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำได้ยากกว่าการ สแกนทางคอลัมน์

สำหรับการสแกนในโครงงานนี้ จะใช้การสแกนทางโรว์ โดยการส่งข้อมูลออกไปจนครบทุกคอลัมน์ ซึ่งมีทั้งหมด 32 คอลัมน์ จากนั้นจึงทำการสแกนทางโรว์ จนครบ 16 แถว ก็จะครบ 1 รอบในการสแกน

2.9 ชิฟรีจิสเตอร์ (Shift register)

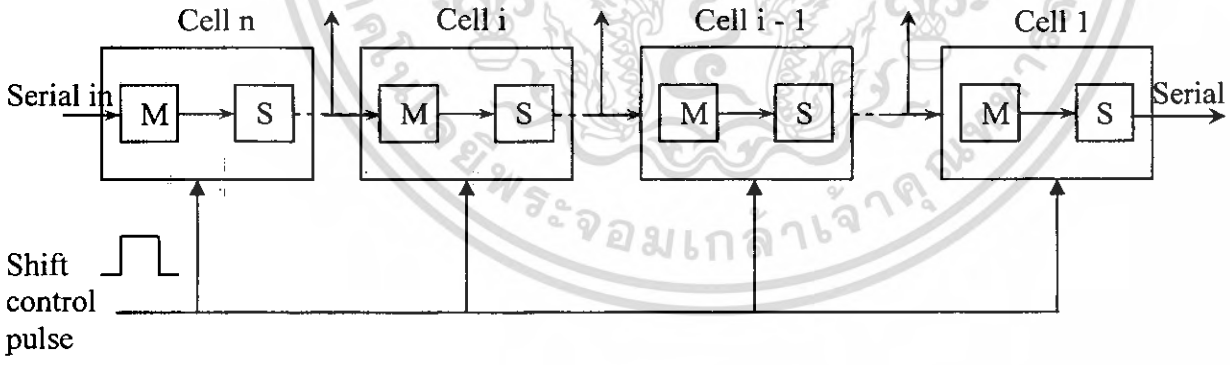
ชิฟรีจิสเตอร์(shift register)เป็นส่วนหนึ่งของวงจรซีเควิ่นเชียลลอจิก(sequential logic circuits) ที่สร้างมาจากฟลิปฟลอปที่เคลื่อนย้าย(manipulate) ตำแหน่งบิตของข้อมูลไบนารี(binary data) โดยการเลื่อนบิตข้อมูลไปทางซ้ายหรือขวา ชิฟรีจิสเตอร์ทั่วไปแสดงใน**รูปที่ 2.13** ชิฟท์รีจิสเตอร์ขนาด n บิต ใน**รูปที่ 2.13a** รับ n บิตของข้อมูลไบนารีและ ถูกสร้างโดยใช้การออกแบบมาสเตอร์สเลฟฟลิปฟลอป(master-slave flip flop design) และการทำงานมาสเตอร์สเลฟฟลิปฟลอปในแต่ละมาสเตอร์สเลฟฟ์ลิปฟลอปจะเป็น 1 เซลล์(cell)ของ ชิฟรีจิสเตอร์แต่ละเซลล์มีมาสเตอร์แลตช์(master latch) และ สเลฟแลตช์(slave latch) และ รับ 1 บิตของสัญญาณนาฬิกา เป็นจังหวะต่ำสูงต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72761

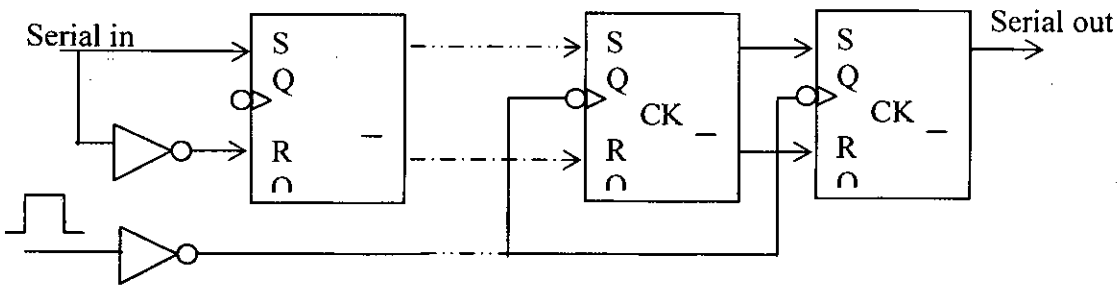
(0->1->0) ในการเปลี่ยนตำแหน่งหรือเลื่อนบิตข้อมูล ไปทางขวา 1 ตำแหน่ง ข้อมูลไบนารี ปกติแล้วข้อมูลจะอยู่ในสเลฟแลตช์ ในขอบขาขึ้น (0->1) ของสัญญาณนาฬิกา จากสเลฟแลตช์(slave latch)ในแต่ละเซลล์ถูกเปลี่ยนไปยังมาสเตอร์แลตช์ในเซลล์ถัดไปทางขวาข้อมูล ในเซลล์ เดี่ยว ในขณะนั้นจะประกอบด้วย ค่าเก่า อยู่ใน สเลฟแลตช์ (slave latch) และ ค่าใหม่ (ค่าเอาท์พุท หลังจากเสร็จสิ้นการเลื่อนสัญญาณนาฬิกา) อยู่ในมาสเตอร์แลตช์ในขอบขาลง (1->0) ของสัญญาณนาฬิกา ในแต่ละเซลล์ ถูกเคลื่อนไปยัง สเลฟแลตช์แสดง ค่าใหม่ไปที่ขาเอาท์พุทดังนั้นหลังจากผ่านทั้งขาขึ้น และขาลงของ สัญญาณนาฬิกา แล้วบิตข้อมูลในเซลล์ X_i ได้ถูก เคลื่อนไป เซลล์ X_{i-1} หรือเลขไบนารีในชิฟรีจิสเตอร์ได้ถูกย้ายไปทางขวา 1 ตำแหน่ง เราเรียกรีจีสเตอร์นี้ว่าชิฟท์รีจิสเตอร์ซีเรียลอิน/ซีเรียลเอาท์ (serial in , serial-out) ซึ่งแสดงไว้ใน**รูปที่ 2.13a**

รูปที่ 2.13 b แสดงรูปของชิฟท์รีจิสเตอร์ซีเรียลอิน/ซีเรียลเอาท์ (serial-in, serial-out shift register) โดยใช้เอจทริกเกอร์อาร์-เอส ฟลิปฟลอบ (edge-triggered SR flip flop)ใช้สัญลักษณ์ลอจิก (logic symbol)แทนสำหรับฟลิปฟลอป (flip flop) และ CK แทนสัญญาณนาฬิกาสามเหลี่ยมแทนว่ามันคือฟลิปฟลอปทำงานที่ขอบของสัญญาณนาฬิกา (edge-trigger input) และวงกลม (bubble) ข้างหน้า แทนว่ามันจะทำงานเมื่อ เป็นขอบขาลง (1->0) ดังนั้นเราเรียกว่าอาร์-เอสฟลิปฟลอปแบบทำงานที่ขอบขาลง (negative-edge triggered SR flip flop) ส่วนชิฟท์คอนโทรลพัลส์ (shift control pulse) จะถูกกลับโดยใช้เกต NOT และ ใช้กับ CK และฟลิปฟลอป(flip flop)นี้จะเปลี่ยนเมื่อเปลี่ยนจากต่ำ 0ไปสูง 1 (0->1) ในทรานซิทชัน (การเปลี่ยน) นี้ข้อมูลในฟลิปฟลอป X_i (flip flop X_i)จะถูกย้ายไปยังฟลิปฟลอป X_{i-1} (flip flop X_{i-1}) ซึ่ง จะแสดงใน**รูปที่ 2.13**



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b)

รูปที่ 2.13 ชิฟท์รีจิสเตอร์ซีเรียลอิน/ซีเรียลเอาท์(serial-in, serial-out shift register)
(a)ชิฟท์รีจิสเตอร์แบบเซลล์มาสเตอร์-สเลฟชิฟรีจิสเตอร์(shift register with master-slave cell)
(b) ชิฟท์รีจิสเตอร์ อาร์เอส ฟลิปฟลอปทำงานที่ขอบของสัญญาณนาฬิกา
(shift register with edge-triggered RS flip flop)

2.10 อุปกรณ์หน่วยความจำ

เนื่องจากการทำงานของวงจรแบบซีเควนเชียลจะต้องนำอินพุตที่ได้มาประมวลผลร่วมกับสภาวะปัจจุบันของระบบ ดังนั้น จึงจำเป็นต้องมีอุปกรณ์สำหรับเก็บสภาวะของระบบเอาไว้ อุปกรณ์นั้นคือ อุปกรณ์หน่วยความจำ

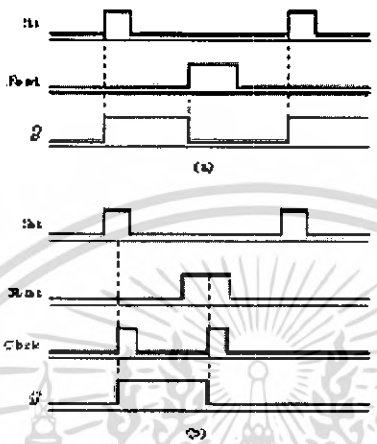
อุปกรณ์หน่วยความจำในวงจรดิจิตอลเป็นแบบไบสเตเบิล (Bistable) หมายถึง สามารถมีสภาวะคงที่ได้ 2 สภาวะ คือ สภาวะที่เป็น 0 หรือ สภาวะที่เป็น 1 โดยเอาต์พุตของอุปกรณ์แทนด้วยสัญลักษณ์ Q

อุปกรณ์หน่วยความจำแบ่งเป็น 2 ประเภทคือแลตช์ (Latch) และ ฟลิปฟลอป (Flip-Flop) โดยแลตช์เป็นอุปกรณ์หน่วยความจำที่การเปลี่ยนสภาวะของอุปกรณ์จะถูกกระตุ้นด้วยอินพุตเท่านั้น หากสัญญาณที่เข้ามากระตุ้นการทำงานส่งผลให้สภาวะของแลตช์เปลี่ยนเป็น '1' เรียกว่าการเซ็ตแลตช์ (Set latch) หากมีสัญญาณที่เข้ามากระตุ้นการทำงานส่งผลให้สภาวะของแลตช์เปลี่ยนเป็น '0' เรียกว่าการรีเซ็ตแลตช์ การทำงานของแลตช์ แสดงเป็นแผนภาพเวลาได้ดัง**รูปที่ 2.14a**

การแลตช์ คือ การค้างของสภาวะข้อมูลของเอาท์พุตเอาไว้ ถึงแม้ไม่มีข้อมูลทางอินพุทเข้ามาข้อมูลเอาท์พุทครั้งก่อนหน้าก็จะยังคงแสดงสภาวะอยู่จนกระทั่งมีการเลือก (Chip Select) ข้อมูลครั้งใหม่เข้ามาเมื่อข้อมูลใหม่เข้ามาก็จะถูกแลตช์ข้อมูลทางเอาท์พุตไว้เช่นกัน ส่วนการทำงานฟลิปฟลอปนั้นคล้ายกับแลตช์ แต่มีส่วนที่แตกต่างกันคือ การเปลี่ยนสภาวะของฟลิปฟลอปจะถูกควบคุมด้วยสัญญาณนาฬิกาอีกต่อหนึ่ง ดังตัวอย่างใน**รูปที่ 2.14b** จากรูปจะสังเกตได้ว่า ถึงแม้อินพุตของฟลิปฟลอปจะมีการเปลี่ยนแปลงแล้ว แต่หากว่าสัญญาณนาฬิกา (Clock) ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น '0' อยู่ สภาวะของฟลิปฟลอปก็จะยังคงเดิม จำเป็นต้องรอให้สัญญานาฬิกาเป็น '1' เสียก่อนฟลิปฟลอปจึงจะมีการทำงานตามอินพุตที่ใส่เข้ามา การที่สามารถควบคุมการทำงานของฟลิปฟลอปโดยสัญญาณนาฬิกานั้น มีข้อดีหลายประการ ยกตัวอย่างเช่น หากภายในวงจรมี ฟลิปฟลอปหลายตัว เราสามารถควบคุมการทำงานให้ฟลิปฟลอปทั้งหมดทำงานพร้อมกันได้ ทำให้ง่ายต่อการกำหนดสภาวะของระบบ เป็นต้น



รูปที่ 2.14 แผนภาพเวลาของแลตช์ และ ฟลิปฟลอป
(a) แลตช์ถูกกระตุ้นด้วยอินพุตเท่านั้น (b) ฟลิปฟลอปถูกกระตุ้นด้วยอินพุตและสัญญาณนาฬิกา

2.11 Block RAM ในชิพ FPGA

เนื่องจากในการเขียนภาษา VHDL ที่ใช้ในการออกแบบวงจรดิจิตอลนั้น หากมีการเขียนเพื่อสร้างหน่วยความจำในชิพที่ใช้เก็บข้อมูลในปริมาณที่มากๆ จะทำให้เมื่อนำโค้ดนั้นมาสังเคราะห์วงจรด้วยซอฟต์แวร์สังเคราะห์วงจร วงจรที่ผ่านการสังเคราะห์แล้วจะใช้เกตเป็นจำนวนมากซึ่งอาจใช้มากเกินกว่าจำนวนเกตที่มีอยู่ภายในชิพ FPGA ตัวนั้นๆได้ ปัญหาดังกล่าวนั้นจะหมดไป โดยเราจะมาสร้างหน่วยความจำที่ใช้เทคโนโลยีที่ถูกสร้างขึ้นมาตั้งแต่ต้นกระบวนการผลิตตัวชิพ FPGA นั่นคือ Block RAM

สำหรับ Block RAM ที่สร้างขึ้นมานั้น ในโครงงานชิ้นนี้เราใช้ชิพ FPGA ตระกูล Spartan-3 เบอร์ XC3S200 ซึ่งเมื่อดูจากตาราง Block RAM เราสามารถสร้าง Block RAM เพื่อเก็บข้อมูลได้สูงสุด 216 Kbits

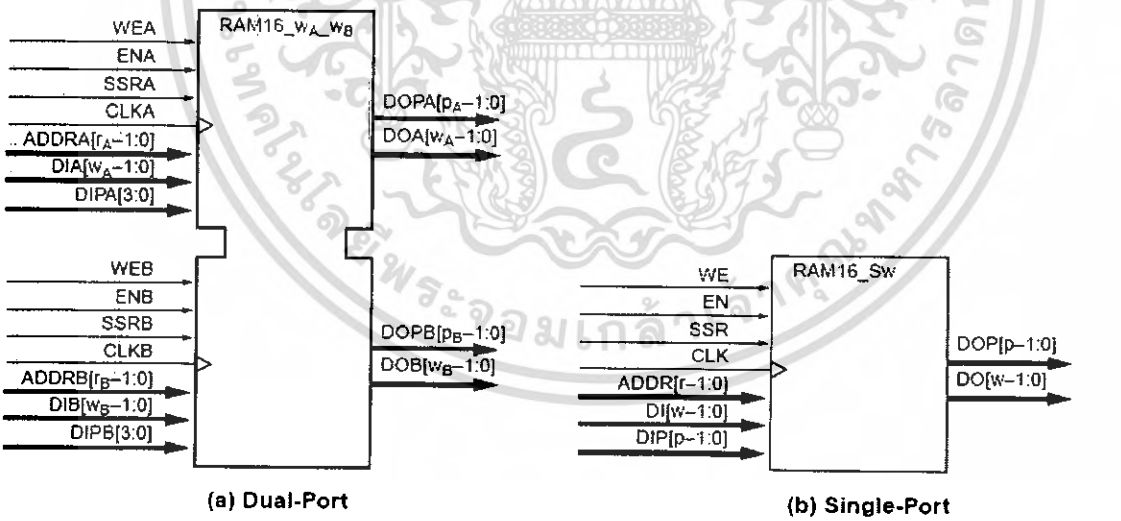
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Device	RAM Columns	RAM Blocks Per Column	Total RAM Blocks	Total RAM Bits	Total RAM Kbits
XC3S50	1	4	4	73,728	72K
XC3S200	2	6	12	221,184	216K
XC3S400	2	8	16	294,912	288K
XC3S1000/L	2	12	24	442,368	432K
XC3S1500/L	2	16	32	589,824	576K
XC3S2000	2	20	40	737,280	720K
XC3S4000/L	4	24	96	1,769,472	1,728K
XC3S5000	4	26	104	1,916,928	1,872K

Notes:
1. 1Kbit = 1,024 bits, per memory conventions.

รูปที่ 2.15 ตาราง Block RAM ที่มีใช้ใน FPGA ตระกูล Spartan-3 และ Spartan-3L

Block RAM มีโครงสร้างอยู่ 2 ลักษณะ คือ (a) Dual-Port (b) Single-Port ตามรูปด้านล่าง ซึ่งในโครงงานชิ้นนี้เราใช้แบบ Single-Port จำนวน 2 ตัว ทำหน้าที่สลับกันทำงาน กล่าวคือ ถ้าตัวที่หนึ่งเขียนข้อมูลลง RAM ตัวที่สองจะอ่านข้อมูลออกมาจากRAM และถ้าตัวที่หนึ่งอ่านข้อมูลออกมาจาก RAM ตัวที่สองจะเขียนข้อมูลลง RAM

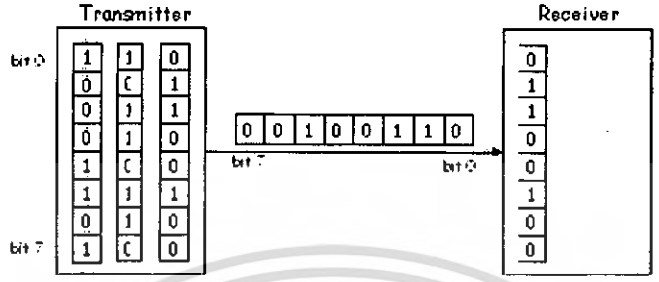


รูปที่ 2.16 แสดงลักษณะโครงสร้างของ Block RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 การสื่อสารแบบอนุกรม (Serial Communication)

เป็นการสื่อสารโดยการส่งข้อมูลทีละบิต ผ่านสายสัญญาณเส้นเดียวจนครบทั้ง 8 bits หรือ 1 ไบท์ โดยจะส่งบิตต่ำ (LSB) ออกไปก่อน สามารถแสดงให้เห็นหลักการส่งข้อมูล แบบอนุกรมได้ ดัง**รูปที่ 2.17**



รูปที่ 2.17 การส่งข้อมูลแบบอนุกรม

การอินเตอร์เฟสตามมาตรฐาน RS-232

มาตรฐาน RS-232 เป็นมาตรฐานที่ได้รับการพัฒนามานานและถูกใช้งานกันอย่างแพร่หลาย เราใช้ RS-232 เชื่อมต่อ DTE (Data Terminal Equipment) เข้ากับ DCE (Data Communication Equipment) เช่น การต่อเทอร์มินัลเข้ากับโมเด็ม มาตรฐาน RS-232 กล่าวถึงลักษณะทางกล , ลักษณะของสัญญาณไฟฟ้าและลักษณะการทำงานที่ใช้การอินเทอร์เฟส ตัวอย่างของอุปกรณ์ที่ใช้ในการอินเทอร์เฟสตามมาตรฐาน RS-232 ได้แก่ เทอร์มินัล , พล็อตเตอร์ , ลอจิกอนาไลเซอร์ (Logic Analyzer) และเครื่องพิมพ์ ถ้าการประยุกต์ใช้งานของเราต้องการทำอินเทอร์เฟสอุปกรณ์ เข้ากับอินเทอร์เฟสมาตรฐาน RS-232 เราจำเป็นต้องแปลงระดับสัญญาณ TTL ให้เป็นระดับสัญญาณแบบอื่น ซึ่งรายละเอียดของระดับสัญญาณที่ใช้สำหรับ RS-232 จะได้กล่าวต่อไป

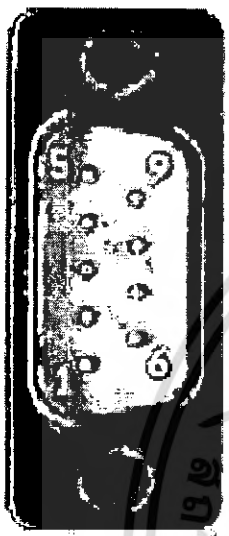
ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส

มาตรฐาน RS-232 ใช้สัญญาณเพียงเส้นเดียวในการส่งสัญญาณ โดยสัญญาณจะส่งไปในทิศทางเดียวกัน ในกรณีที่อัตราเร็วในการส่งข้อมูลมีค่าเท่ากับ 20 kbps (กิโลบิดต่อวินาที) ซึ่งค่านี้เป็นค่าสูงสุดที่ใช้ในการสื่อสารข้อมูล (ในปัจจุบันพัฒนาให้สามารถส่งข้อมูลได้มากกว่านี้) ระยะทางในการส่งข้อมูลไม่ควรเกิน 50 ฟุต (ตามข้อกำหนดในมาตรฐาน) สำหรับการแทนแรงดันของระดับสัญญาณจะแทนระดับสัญญาณของลอจิก “0” ด้วยค่าแรงดัน +3 โวลท์ ถึง +12 โวลท์ ส่วนลอจิก “1” จะแทนระดับสัญญาณด้วยค่าแรงดันระหว่าง –3 โวลท์ ถึง –12โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

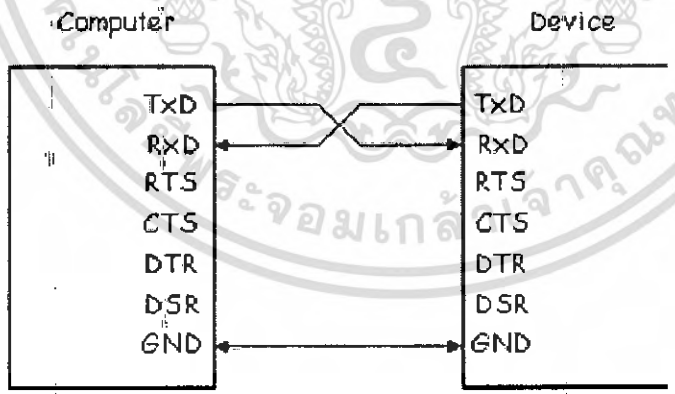
การออกแบบตัวแปลงสัญญาณ

การเชื่อมต่อกับพอร์ทสื่อสารของคอมพิวเตอร์ส่วนบุคคลจะเลือกใช้พอร์ตสื่อสารแบบ อนุกรม 9 ขา (DB-9) ซึ่งสามารถทำการรับส่งข้อมูลได้แบบอนุกรม โดยลักษณะของสัญญาณจะเป็นไปตามมาตรฐาน RS-232 โดยลักษณะของการเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9 สามารถแสดงให้เห็นได้ดัง**รูปที่ 2.18** และ**รูปที่ 2.19**



ตำแหน่งขา DB-9	สัญญาณ
1	Data Carrier Detect : DCD
2	Received Data : RxD
3	Transmitted Data : TxD
4	Data Terminal Ready : DTR
5	Signal Ground : GND
6	Data Set Ready : DSR
7	Request To Send : RTS
8	Clear To Send : CTS
9	Ring Indicator : RI

รูปที่ 2.18 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมแบบ DB-9

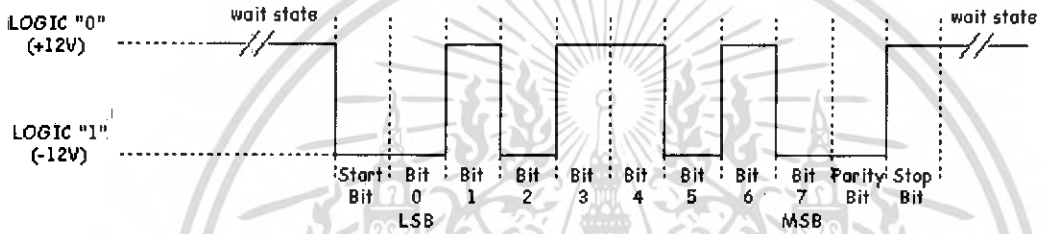


รูปที่ 2.19 การต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น

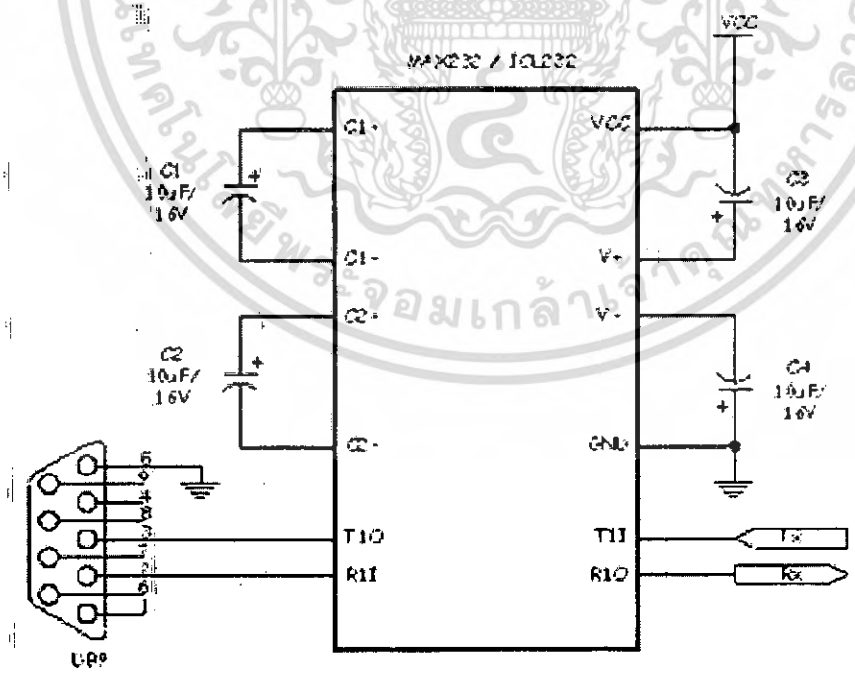
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรอแคปเตอร์

จากหัวข้อที่ผ่านมาตอนนี้เราทราบแล้วว่าในการสื่อสารแบบอนุกรมนั้น ในระดับของลอจิก "0" จะแทนระดับลอจิกด้วยระดับแรงดันระหว่าง +3 โวลท์ ถึง +12 โวลท์และในระดับของลอจิก "1" จะแทนระดับลอจิกด้วยระดับแรงดันระหว่าง –3 โวลท์ ถึง –12 โวลท์ เพราะฉะนั้นต้องทำการแปลง ระดับของลอจิก "1" และ "0" ให้เป็นระดับแรงดันดังกล่าว ซึ่งจะต้องใช้วงจรในการแปลงระดับแรงดัน สามารถแสดงลักษณะการส่งข้อมูลแบบอนุกรมที่ผ่านตัวแปลงแรงดันได้ดัง**รูปที่ 2.20** ในส่วนของวงจรแปลงระดับแรงดันสำหรับการสื่อสาร RS-232 นั้นจะใช้ไอซีเบอร์ MAX232 หรือ ICL232 ดัง**รูปที่ 2.21**



รูปที่ 2.20 รูปแบบการสื่อสารแบบอนุกรม



รูปที่ 2.21 วงจรแปลงระดับแรงดันของการสื่อสารแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

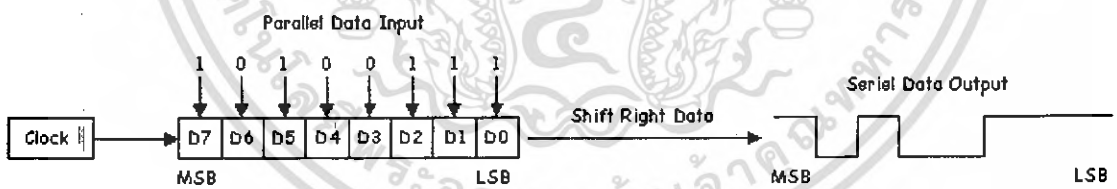
รูปแบบของการส่งข้อมูลแบบอนุกรมและอัตราบอดในการสื่อสาร

อัตราบอด (Baud Rate) คือความเร็วในการรับ - ส่งข้อมูลอนุกรมมีหน่วยเป็นบิตต่อวินาที ซึ่งจะบอกจำนวนบิตที่รับ - ส่งในเวลา 1 วินาที เช่น ส่งข้อมูลด้วยอัตรา 9600 บิตต่อวินาที ก็คือการส่งข้อมูลตัวอักขระขนาด 10 บิต (บิต Start 1 บิต บิตข้อมูล 8 บิต บิต Stop 1 บิต) ได้ 960 ตัวอักขระใน 1 วินาที ซึ่งตารางที่ 1 แสดงอัตราบอต

ของ UART ที่ใช้กันทั่ว ตารางที่ 1 เป็นการแสดงอัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

ตารางที่ 1 อัตราบอดที่ใช้กันทั่วไป

อัตราบอด	ช่วงเวลาของแต่ละบิต
110	9.91 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	0.833 ms
2400	0.417 ms
4800	0.208 ms
9600	0.104 ms
19200	0.052 ms



รูปที่ 2.22 การแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม

ใน**รูปที่ 2.22** เป็นการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยเริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ใน Shift Register หลังจากนั้นจะใช้สัญญาณนาฬิกาในการเลื่อนค่าในรีจิสเตอร์ออกมาทีละบิต (โดยการเลื่อนค่าไปทางขวามือ) โดยบิตแรกที่ถูกเลื่อนออกมาคือบิต LSB ของข้อมูลและบิตที่สองที่ถูกเลื่อนออกมาก็คือบิตที่อยู่ถัดจากบิต LSB และบิตต่อๆไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาก็คือบิต MSB ของข้อมูล

การแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลแบบขนานนั้นจะมีขั้นตอนตรงกันข้ามกับที่กล่าวมานั่นคือข้อมูลแบบอนุกรมจะถูกเคลื่อนเข้าไปใน Shift Register โดยใช้สัญญาณนาฬิกาเป็นตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุม และหลังจากที่มีการเคลื่อนข้อมูลทุกบิตเข้าไปใน Shift Register ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้ จะถูกนำออกมาแบบขนานเพื่อนำไปใช้งานต่อไป

อุปกรณ์ที่ทำหน้าที่แปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนานและแปลงข้อมูลแบบขนานเป็นข้อมูลแบบ

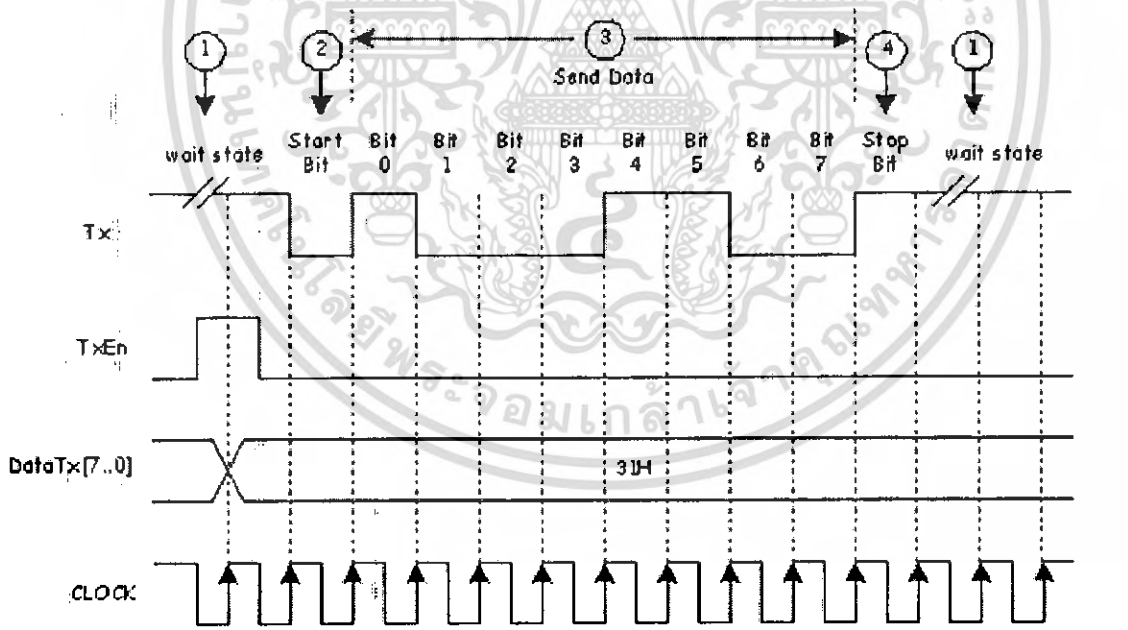
อนุกรมเรียกว่า UART (Universal Asynchronous Receiver-Transmitter) ซึ่งนอกจากจะมีหน้าที่ในการแปลงข้อมูลแล้ว UART ยังมีหน่วยควบคุมและหน่วยตรวจสอบการทำงานอีกด้วย ในการส่งข้อมูลขนาด 8 บิต แบบอนุกรมนี้จะต้องมีบิตเริ่มต้น (Start Bit) และบิตหยุด (Stop Bit) เพิ่มขึ้นมา ซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงๆนั้นมีขนาด 10 บิต

หลักการออกแบบวงจรรับ – ส่งข้อมูลแบบอนุกรมด้วย FPGA

สำหรับวงจรสื่อสารข้อมูลแบบอนุกรมที่เรากำลังจะออกแบบกันนั้น จะมีคุณสมบัติดังนี้ Baud Rate = 9600 Bits/Sec, Data = 8 Bits, Start Bit = 1 Bit, Stop Bit = 1 Bit, Parity = none

ภาคส่งข้อมูลแบบอนุกรม

ในส่วนนี้เป็นการออกแบบ FPGA ให้ทำหน้าที่เป็นตัวส่งข้อมูลแบบอนุกรมให้แก่อุปกรณ์ต่างๆ เช่น คอมพิวเตอร์, ไมโครคอนโทรลเลอร์ เป็นต้น รูปแบบการส่งข้อมูลแบบอนุกรม แสดงดัง**รูปที่ 2.23**



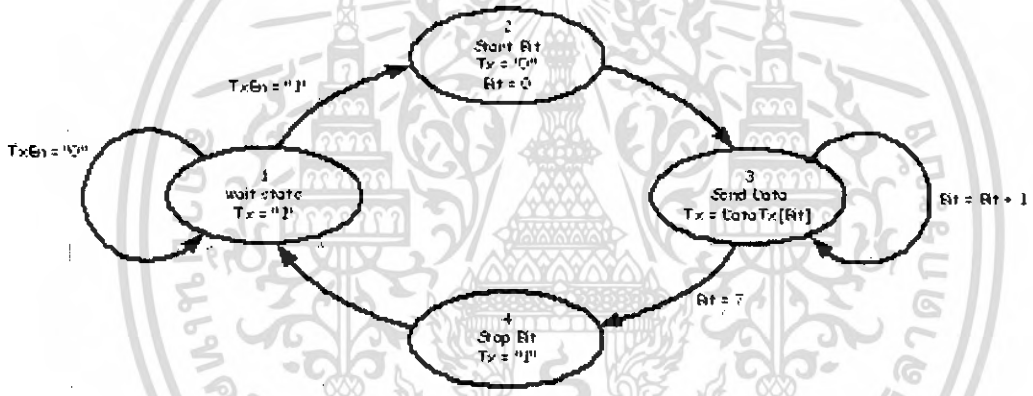
รูปที่ 2.23 รูปแบบการส่งข้อมูลแบบอนุกรม

จาก**รูปที่ 2.23** สัญญาณ Tx เป็นสัญญาณขนาด 1 บิต ใช้สำหรับส่งข้อมูลแบบอนุกรมออกไป ในการออกแบบวงจรส่งข้อมูลแบบอนุกรมจะต้องมีสัญญาณ CLOCK เป็นสัญญาณอ้างอิง เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดความเร็วในการส่งข้อมูลซึ่งจะต้องตรงกับทางภาครับด้วย ความเร็วในการส่งข้อมูลนี้ก็คือ Baud Rate นั้นเอง เช่น ต้องการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที ก็จะต้องกำหนดให้สัญญาณ CLOCK มีค่าเท่ากับ 9600 Hz เมื่อสัญญาณ TxEn มีค่าเป็นลอจิก "0" แสดงว่ายังไม่ต้องการส่งข้อมูล จะทำให้สัญญาณ Tx มีสถานะเป็นลอจิก "1" จนกว่าสัญญาณ TxEn จะมีสถานะเป็นลอจิก "1" แสดงว่าต้องการส่งข้อมูลออกไป จะทำให้สัญญาณ Tx มีค่าเป็นลอจิก "0" เพื่อเป็นการบอกทางภาครับว่าจะเริ่มต้นส่งข้อมูลออกไป

หลังจากนั้นจะทำการส่งข้อมูลออกไปDataTx[7..0] เป็นข้อมูลที่ต้องการส่งแบบอนุกรม โดยส่งบิตที่มีความสำคัญต่ำสุด (Bit 0) ออกไปก่อน จะส่งข้อมูลออกไปที่ละบิตจนครบทั้ง 8 บิต จากนั้นสัญญาณ Tx จึงมีสถานะเป็นลอจิก "1" เพื่อบอกทางภาครับว่าเป็นการสิ้นสุดการส่งข้อมูล เราสามารถนำวิธีการส่งข้อมูลแบบอนุกรมที่ได้กล่าวมาแล้วมาเขียนเป็น State Diagram ได้ดัง**รูปที่ 2.24**



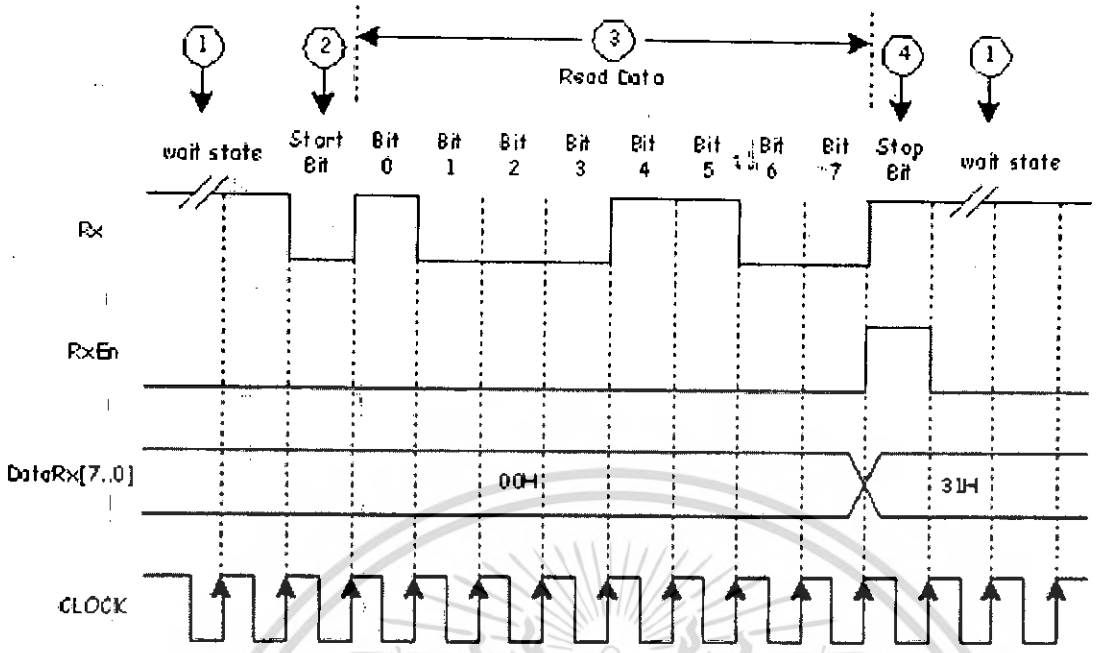
รูปที่ 2.24 State Diagram ของการส่งข้อมูลแบบอนุกรม

จาก State Diagram ใน**รูปที่ 2.24** เมื่อ TxEn มีค่าเป็นลอจิก "0" จะทำให้ Tx มีค่าเท่ากับลอจิก "1" และจะยังอยู่ใน State 1 จนกว่า TxEn จะมีค่าเป็นลอจิก "1" จึงจะเข้าสู่ State 2 ใน State 2 สัญญาณ Tx จะมีค่าเป็นลอจิก "0" เป็นการกำหนดบิตเริ่มต้นการส่งข้อมูล หลังจากนั้นจะเข้าสู่ State 3 ซึ่งจะเป็นการส่งข้อมูลออกไปที่ละบิต เริ่มต้นจะส่งข้อมูลบิตที่ 0 ออกไปก่อนและวนส่งข้อมูลออกไปจนครบ 8 บิต จึงจะหลุดเข้าสู่ State 4 จะกำหนดให้ Tx มีค่าเป็นลอจิก "1" เป็นการกำหนดบิตสิ้นสุดการส่งข้อมูล หลังจากนั้นจึงกลับเข้าสู่ State 1 ใหม่อีกครั้ง

ภาครับข้อมูลแบบอนุกรม

สำหรับภาครับข้อมูลแบบอนุกรมการทำงานจะคล้ายๆ กับทางภาคส่ง ข้อมูลแบบอนุกรมจะถูกส่งมาจากตัวส่ง เช่น คอมพิวเตอร์ , ไมโครคอนโทรลเลอร์ เป็นต้น ข้อมูลแบบอนุกรมที่ส่งมาจะมีลักษณะดัง**รูปที่ 2.25**

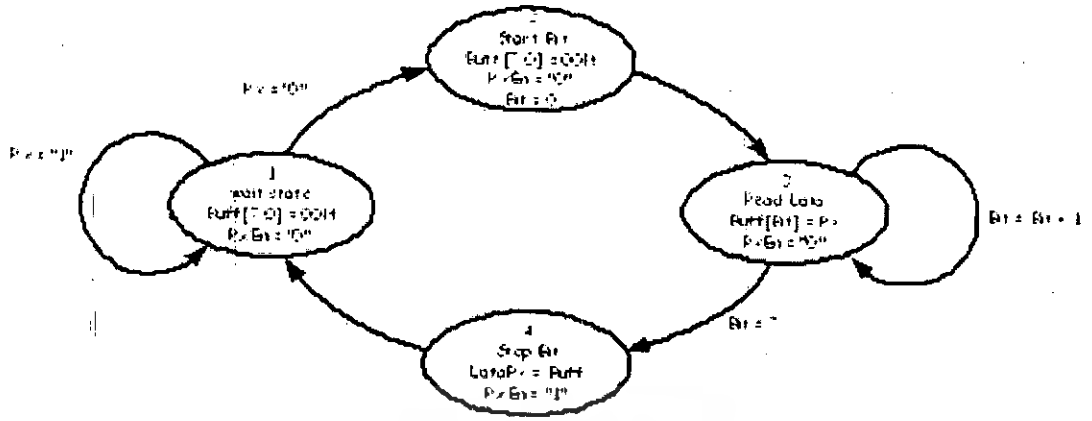
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 รูปแบบการรับข้อมูลแบบอนุกรม

จาก**รูปที่ 2.25** สัญญาณ Rx เป็นสัญญาณขนาด 1 บิตใช้สำหรับรับข้อมูลแบบอนุกรมที่ส่งมาจากตัวส่ง ในการออกแบบภาครับข้อมูลจะต้องมีสัญญาณ CLOCK เป็นสัญญาณอ้างอิง เพื่อกำหนดความเร็วในการรับข้อมูลซึ่งจะต้องสอดคล้องกับทางภาคส่ง เช่นภาคส่งใช้อัตราเร็วในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ทางภาครับก็จะต้องกำหนดให้มีอัตราเร็วในการรับข้อมูลเป็น 9600 บิตต่อวินาทีเช่นกัน เพราะฉะนั้นจะต้องกำหนดให้สัญญาณ CLOCK มีความถี่เท่ากับ 9600 Hz เมื่อสัญญาณ Rx มีค่าเป็นลอจิก "1" แสดงว่ายังไม่มีการส่งข้อมูลออกมา จะรอจนกว่า Rx มีค่าเป็นลอจิก "0" แสดงว่าทางภาคส่งจะเริ่มส่งข้อมูลมาแล้ว หลังจากนั้นจะทำการอ่านข้อมูลเข้ามาเก็บไว้ที่ละบิดจนครบทั้ง 8 บิต และตรวจสอบสัญญาณ Rx ว่าเป็นลอจิก "1" หรือไม่ หากเป็นลอจิก "1" แสดงว่าสิ้นสุดการส่งข้อมูล และกำหนดให้สัญญาณ DataRx [7..0] มีค่าเท่ากับสัญญาณที่รับมาได้ และกำหนดให้สัญญาณ RxEn มีค่าเป็นลอจิก "1" เพื่อเป็นสัญญาณกระตุ้นให้วงจรต่อไปนำข้อมูล DataRx[7..0] ไปใช้งาน เราสามารถนำวิธีการรับข้อมูลแบบอนุกรมที่ได้กล่าวมาแล้วมาเขียนเป็น State Diagram ได้ดัง**รูปที่ 2.26**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 State Diagram ของการรับข้อมูลแบบอนุกรม

จาก State Diagram ใน**รูปที่ 2.26** เริ่มต้นจะอยู่ใน State 1 หากสัญญาณ Rx มีค่าเป็นลอจิก "1" จะไม่มีการเปลี่ยน Stateจะให้ค่า RxEn เป็นลอจิก "0" และ Buff[7..0] ซึ่งเป็น Buffer สำหรับเก็บข้อมูลที่อ่านมาได้จากสัญญาณ Rx มีค่าเป็น "00000000" เมื่อสัญญาณ Rx มีค่าเป็นลอจิก "0" แสดงว่าเป็นบิตเริ่มต้นของข้อมูลจะเปลี่ยนเป็น State 2 เมื่อเข้ามาที่ State 2 แสดงว่าได้มีการส่งบิตเริ่มต้นข้อมูลมาแล้ว ใน State นี้จะยังคงกำหนดให้ Buff[7..0] มีค่าเท่ากับ "00000000" , RxEn เป็นลอจิก "0" และ Bit มีค่าเท่ากับ 0 หลังจากนั้นจะเข้าสู่ State 3 โดยอัตโนมัติ ภายใน State 3 จะมีการอ่านข้อมูลจาก Rx เข้ามาเก็บไว้ใน Buff จะอ่านข้อมูลจนครบทั้ง 8 บิต นำข้อมูลที่ได้ทั้งหมดเก็บไว้ใน Buff และกำหนดให้สัญญาณ RxEn มีค่าเป็นลอจิก "0" เมื่ออ่านข้อมูลจนครบทั้ง 8 บิตแล้วก็เปลี่ยน State เป็น State 4 ภายใน State 4 จะมีการโอนย้ายข้อมูลจาก Buff มาเก็บไว้ใน DataRx ซึ่งเป็นข้อมูลขนาด 8 บิตทั้งหมดที่รับมาได้ และกำหนดให้ RxEn มีค่าเป็นลอจิก "1" เพื่อเป็นสัญญาณกระตุ้นให้ภาคอื่นๆ รับทราบว่าได้รับข้อมูลมาครบ 8 บิตแล้ว หลังจากนั้นก็จะกลับมา State 1 เพื่อรอรับข้อมูลอีกครั้ง

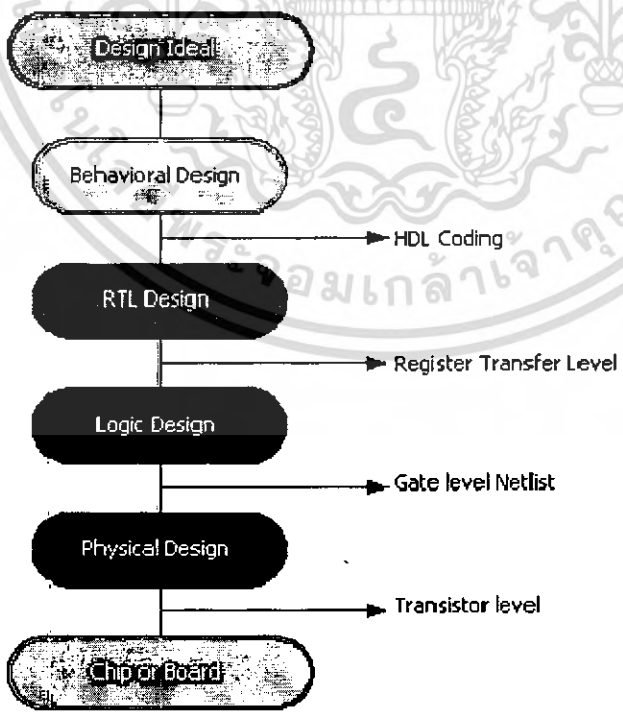
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 ภาษา VHDL

ขนาดซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบหรือ CAD มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกันอีกทั้งอุปกรณ์และวิธีการออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นอีกด้วย สำหรับภาษาบรรยายพฤติกรรมฮาร์ดแวร์ (HDL: Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่รับการพัฒนามาอย่างต่อเนื่องเพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

3.1 การออกแบบระบบดิจิตอล

ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำหนดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ ที่ใช้งานได้จะต้องผ่านขั้นตอนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้น ก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป **รูปที่ 3.1** แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิตอลทั่วไป ขั้น แรกผู้ออกแบบจะกำหนดแนวความคิดในการออกแบบแล้วทำการพัฒนาให้สามารถนำมาใช้ได้อย่างสมบรูณ์ ซึ่งภาย ในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบในเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะเป็นผังงานแสดงแบบหรือ รหัสคำสั่งเทียม (Pseudo code) ก็ได้



รูปที่ 3.1 แสดงขั้นตอนการออกแบบระบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิก ที่จำเป็นทั้งหมดเพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสอง ทิศทาง (Unidirectional or Bidirectional Bus) ส่วนกระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่ ขั้นดอนถัดมาเป็นการออกแบบวงจรลอจิก ซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐานและฟลิปฟลอป (flip-flop) มาประกอบเป็นอุปกรณ์ย่อยต่างๆ เช่น รีจิสเตอร์เก็บข้อมูล บัสวงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ ที่ได้ในขั้นดอนนี้จะเป็นเครือข่ายของการโยงใยระหว่างเกทและ ฟลิปฟลอปนั่นเองการออกแบบในขั้นตอนถัดไป เป็นการเปลี่ยนเครือข่ายการโยงใยที่ได้จากขั้นตอนที่แล้วให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ Layout ซึ่งขั้นดอนนี้จะเกี่ยวข้องโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์ เพื่อ แทนเกทและฟลิปฟลอปต่างๆและในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือสารที่โรงงานเพื่อผลิตออกมาเป็น วงจรรวมในที่สุด

3.2 ข้อกำหนดของภาษา VHDL

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

2.) สนับสนุนการออกแบบแบบลำดับขั้น

การออกแบบแบบลำดับขั้นเป็นลักษณะที่สำคัญอย่างหนึ่งสำหรับการออกแบบระบบที่มีหลายๆ ระดับ โดยในการ ออกแบบจะประกอบด้วยส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายหน้าที่การทำงาน ซึ่งหน้าที่การทำงาน ของระบบสามารถกำหนดได้ด้วยตัวเอง หรืออาจถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลงไปได้เช่นกัน แต่ที่ระดับล่างสุด องค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวมันเอง และไม่สามารถกำหนด การทำงานโดยลักษณะแบบโครงสร้างได้

3.) ไลบรารี

VHDL ได้สนับสนุนการมีไลบรารีเพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของ อุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูก ต้องควรจัดเก็บไว้ในไลบรารีหลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้วเพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไป ใช้ได้ด้วย

4.) ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่งหรือกระบวนการโดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ VHDL ก็ตาม ตัวภาษา เองก็ยังมีการจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่งไว้ให้ด้วย เมื่อผู้ออกแบบได้กำหนดหน้าที่และองค์ประกอบ ที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออกแบบยังสามารถบรรยายหน้าที่การทำงานซึ่งเป็นรายละเอียดภายใน ของแต่ละองค์ประกอบได้ในลักษณะเดียวกับการเขียนโปรแกรมที่ประกอบด้วยโครงสร้างแบบ case, if - then - else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์กระทำได้ สะดวกและง่ายขึ้น อย่างไรก็ตามโครงสร้างทั้งหมดของ VHDL ก็ยังคงเป็นการทำงานแบบพร้อมเพรียงกันเช่นเดิม

5.) การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุทและเอาท์พุทแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของอุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อมและลักษณะทางกายภาพของอุปกรณ์นั้นๆ ด้วย ซึ่งภาษาสำหรับการออกแบบที่ดีควร ให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลา โหลด และเงื่อนไขทางสภาพแวดล้อมอื่นๆ ซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา VHDL ด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.) ชนิดของข้อมูล

VHDL สามารถกำหนดชนิดของข้อมูลไม่เพียงแค่ชนิด BIT และ BOOLEAN เท่านั้น แต่ยังสามารถกำหนดชนิด ของข้อมูลเป็นจำนวนเต็ม จำนวนจริง จุดทศนิยม และชนิดลำดับการนับ (Enumerate Type) หรือแม้แต่ชนิดของ ข้อมูลที่ผู้ออกแบบกำหนดขึ้นมาเองก็ได้

7.) โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชันและโพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน VHDL ซึ่งผู้ออกแบบ สามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การกำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการได้เช่นเดียวกับการเขียนโปรแกรมทั่วไป

8.) การควบคุมเวลา

VHDL อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการ การตรวจสอบ การออกแบบเกทหรือการหน่วงเวลาก็สามารถกระทำได้โดยการกำหนดช่วงเวลาที่แน่นอนหรือกำหนดให้มีการรอคอย เหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

9.) การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบ โดยการกำหนดโครง สร้างขององค์ประกอบร่วมที่เกิดจากองค์ประกอบย่อยซึ่งแตกต่างกันหรือเหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ VHDL เช่นกัน

3.3 ส่วนประกอบต่างๆ ของภาษา VHDL

ในการเขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเอชดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

- หน่วยการออกแบบเอนทิตี (Entity Design Unit)

- หน่วยการออกแบบสถาปัตยกรรม (Architecture Design Unit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยการออกแบบแพ็กเก็ต (Package Design Unit)

- หน่วยการออกแบบโครงแบบ (Configuration Design Unit)

หน่วยการออกแบบเอนทิตี

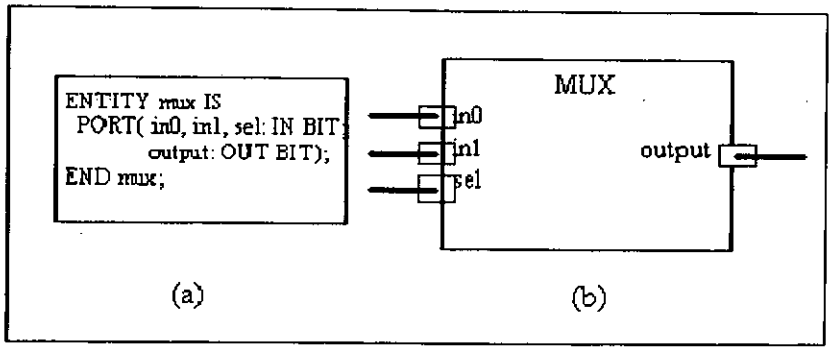
หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้นที่เรียกว่า หน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหล่ผ่านจุดต่อเหล่านั้น **รูปที่ 3.2** แสดงให้เห็นโครงสร้างอย่างง่ายๆ ของ หน่วยการออกแบบเอนทิตี

```
ENTITY component_name IS

    Input and output ports

    Physical and other parameters

END [component_name];
```

รูปที่ 3.2 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

ส่วนนี้จะขึ้นต้นด้วยคำ ENTITY และ IS ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component_name) สำหรับการตั้งชื่อนั้นต้องเป็นไปตามกฎเกณฑ์ของภาษา หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (input-output) รวมทั้งพารามิเตอร์อื่นๆ ส่วนนี้เรียกว่าส่วนหัว (entity header) และที่สำคัญคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า END และเครื่องหมายอัฒภาคเสมอ (;)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

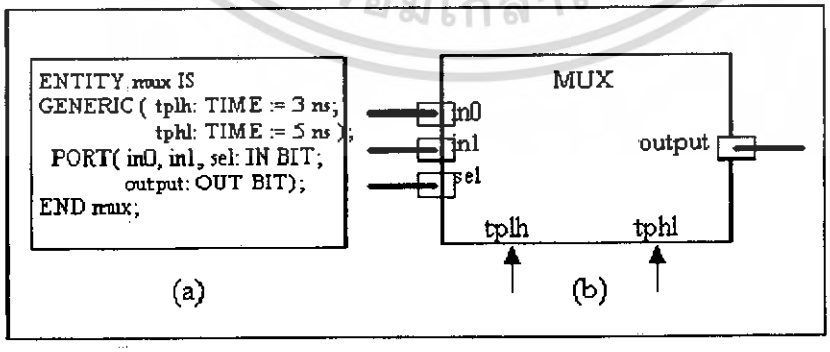


รูปที่ 3.3 แสดงรูปแบบของมัลติเพลกซ์ (a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล

(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

ใน**รูปที่ 3.3** เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพลกซ์ หรือ MUX ในส่วนหัวของเอนทิตี มีการกำหนดจุดต่อ 4 จุดภายใต้ชุดคำสั่ง PORT โดยที่ 3 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ in0, in1, sel ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลเข้าของข้อมูล (IN) ที่แสดงด้วยรูปสี่เหลี่ยมโปร่งใน**รูปที่ 3.3(a)** ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (OUT) ที่แสดงด้วยรูปสี่เหลี่ยมทึบใน**รูปที่ 3.3 (b)** ส่วนประเภทของข้อมูลที่จะไหล เข้าและออก นั้นเป็นประเภท BIT ที่สามารถมีค่าได้เพียงสองค่าคือ '0' และ '1' เท่านั้น

นอกจากนั้นผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ที่เป็นข้อมูลเพิ่มเติมอื่นๆ ลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์ อันได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างใน**รูปที่ 3.4**



รูปที่ 3.4 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย

(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล (b) มุมมองของตัวเชื่อมประสาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบางกรณีสามารถที่จะใช้ภาษาวีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหล เข้าและออกของข้อมูล ได้ ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบ สำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่ง คือ วีเอชดีแอลสำหรับการทดสอบเปรียบเทียบ (Test bench)

```
ENTITY test_bench IS

END test_bench;
```

รูปที่ 3.5 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

หน่วยการออกแบบสถาปัตยกรรม

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงานพฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางคลอดจนพารามิเตอร์ต่างๆ ที่กำหนดใน หน่วยการออกแบบเอนทิตี **รูปที่ 3.6** แสดงให้เห็นถึงโครงสร้างอย่างง่ายๆ ของหน่วยการออกแบบสถาปัตยกรรม

```
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 3.6 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OF <entity design unit> IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในสถาปัตยกรรมนั้นได้ อาทิเช่นประเภท (type) ต่างๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ตัวคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า และไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง PORT) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า BEGIN กับ END ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (concurrent statement) เท่านั้น หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้นๆ ที่เป็นส่วนเผื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

- ประเภทการไหลของข้อมูล (Dataflow description)

- ประเภทพฤติกรรม (Behavioral description)

- ประเภทโครงสร้าง (Structure description)

- ประเภทผสม (Mixed model description)

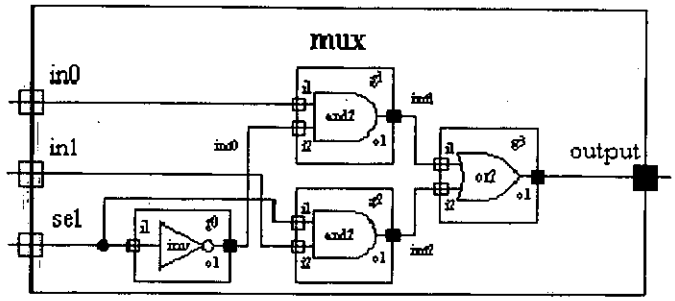
```
ARCHITECTURE data_flow  OF mux IS

BEGIN

   output <= ((NOT sel) AND in0) OR (sel AND in1);

END data_flow;
```

รูปที่ 3.7 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ดามฟังก์ชันบูลีน

$$output = (\overline{sel}.in0) + (sel.in1)$$

รูปที่ 3.7 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (*in0, in1*) กับข้อมูลที่ไหลออก (*output*) ประกอบด้วยชุดคำสั่งแบบแข่งขนานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพลกซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL: Register Transfer Level)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์

รูปที่ 3.8 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง โดยใช้อินเวอร์เตอร์ (inv ที่ตำแหน่ง g0), แอนด์เกต 2 อินพุตจำนวน 2 ตัว (and2 ที่ตำแหน่ง g1 และ g2) และ ออร์เกต 2 อินพุต (or2 ที่ตำแหน่ง g3) มาสร้างตามฟังก์ชันบูลีนของ**รูปที่ 3.7**

```
ARCHITECTURE struc OF mux IS

  COMPONENT inv

  PORT ( i1 : IN BIT ; o1 : OUT BIT );

  COMPONENT and2

  PORT ( i1, i2 : IN BIT ; o3 : OUT BIT );

  COMPONENT or2

  PORT ( i1, i2 : IN BIT ; o1 : OUT BIT );

END COMPONENT;

  SIGNAL int0, int1, int2 : BIT;

BEGIN

  g0 : inv   PORT MAP (i1 => sel, o1 => int0);

  g1 : and2 PORT MAP (i1 => in0, i2 => int0, o1 => int1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
g3 : or2   PORT MAP (i1 => int1, i2 => int2, o1 => ouput);

END struc;
```

รูปที่ 3.9 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง

```
ARCHITECTURE behav OF mux IS

BEGIN

    PROCESS (in0, in1, sel)

    BEGIN

        IF (scl = '0') THEN output <= in0;

        ELSE output <= in1;

        END IF;

    END PROCESS;

END behav;
```

รูปที่ 3.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพลกซ์ในลักษณะของ ประเภทพฤติกรรม ประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จาการจำลองการทำงานที่เหมือนกัน ซึ่งนี้ก็เป็นข้อดีของภาษาวีเอชดีแอล

หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดย หน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือ จากหน่วยการออกแบบแพ็กเก็ตอื่นๆ นอกจากนั้นสิ่งที่นิยมทำกันมากคือรูปแบบมาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74XX เป็นต้น) จะถูกเก็บไว้ในแพ็กแก็ต ที่ทุกคนสามารถเข้าถึง โดยปกติแล้ว แพ็กแก็ตจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กแก็ต (Package declaration) และ ส่วนของบอดี้แพ็กแก็ต (Package body) เนื่องจาก แพ็กแก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กแก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอล สามารถกระทำได้ด้วยชุดคำสั่ง USE

- Package declaration

ส่วนที่มีความสำคัญที่สุดของแพ็กแก็ต (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กแก็ต เพราะจะเป็นส่วนที่กำหนดชื่อ ของสิ่งที่ประกาศอยู่ภายในแพ็กแก็ต สำหรับนำไปใช้ภายนอกตัวของแพ็กแก็ตเอง สิ่งใดๆ ถูกประกาศในส่วนของ ส่วนบอดี้แพ็กแก็ต แต่ไม่ถูกประกาศในส่วนการประกาศแพ็กแก็ต จะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือ จุดเชื่อมต่อ หรือพอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้ว แพ็กแก็ต สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศชนิด (Type) หรือ สัญญาณ เช่นเดียวกันกับ ส่วนบอดี้แพ็กแก็ต ที่ไม่จำเป็นต้องมี ส่วนของการประกาศแพ็กแก็ต แต่แพ็กแก็ต นั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
PACKAGE package_name IS

    Package_declarative_part

END package_name;
```

รูปที่ 3.11 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กแก็ต

- Package body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กแก็ต แล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กแก็ต ทั้งนี้รวมทั้ง การกำหนดค่าคงที่ต่างๆ อันได้แก่ดังคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กแก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กแก็ต) ฉะนั้นส่วนบอดี้แพ็กแก็ต จึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก็ต ไม่มีการประกาศชื่อ ที่เป็นโปรแกรมย่อย หรือ ค่าคงที่ การเขียนบอดี้แพ็กแก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงใน**รูปที่ 3.12**

```
PACKAGE BODY package_name IS

    declarative part

END package_name;
```

รูปที่ 3.12 โครงสร้างของบอดี้แพ็กแก็ต

หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตี หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

```
CONFIGURATION identifier OF entity_name IS

    Configuration_declarative_part

END ;
```

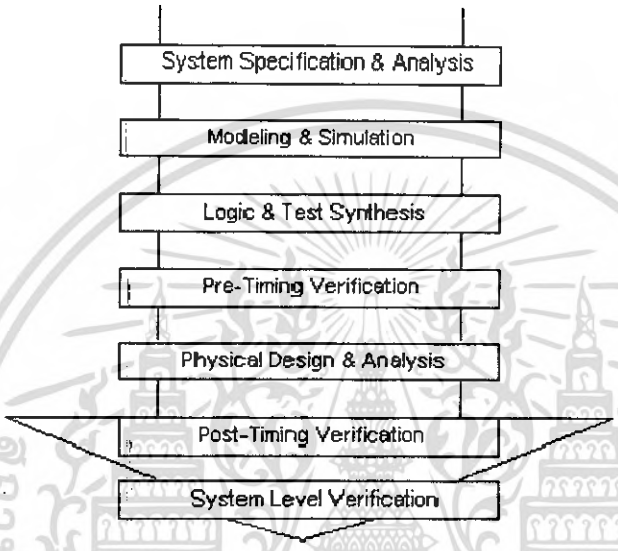
รูปที่ 3.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

3.4 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน เช่น วงจรรวม (ASIC: Application Specific Integrated Circuit) วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของของบล็อกไดอะแกรมเสียก่อน ก่อนที่จะวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาติให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามที่ต้องการ และเพิ่มเติมในรายละเอียดที่ละขั้นนี้คือหลักการออกแบบจากบนลงล่าง (Top-Down Design) ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Botom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่า 90% เพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากให้การออกแบบในลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอลกับหลักการออกแบบจากบนลงล่าง จึงเป็นทางออกให้กับวิศวกรออกแบบที่จะสามารถออกแบบและพัฒนา วงจรที่มีซับซ้อนได้มากขึ้น และช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ



รูปที่ 3.14 ขั้นตอนการออกแบบจากบนลงล่าง

จาก**รูปที่ 3.14** แสดงให้ถึงขั้นตอนของการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายๆ เทคโนโลยี เช่น พีแอลดี (PLD: Programmable Logic Device) อันได้แก่ พีแอลเอ (PLA: Programmable Logic Array), เอฟพีจีเอ (FPGA: Filed Programmable Gate Array), ซีพีแอลดี (CPLD: Cell Programmable Logic Device) เป็นต้น นอกนั้นยังมี เซมิคัลตัมไอซี (Semi-Custom IC) ได้แก่ เกตอะเรย์ (Gate array), เซลล์มาตรฐาน (Standard Cell) ขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียดดังนี้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษาวีเอชดีแอล หรือ ภาษาเอชดีแอลอื่นๆ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจร หรือสังเคราะห์ ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกต และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสท์ (Netlist) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในรูประดับเกตหรือเน็ทลิสท์แล้ว ข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงาน ในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี เวลาหน่วงของการแพร่กระจาย (Propagation delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ นาโนวินาที (10^{-9} วินาที) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่างๆ จำนวน 10,000 เกต ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรรวมทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวม (ASIC)

6. หลังจากที่ได้วงจรจริงมาแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วยอินพุตและเอาต์พุตแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

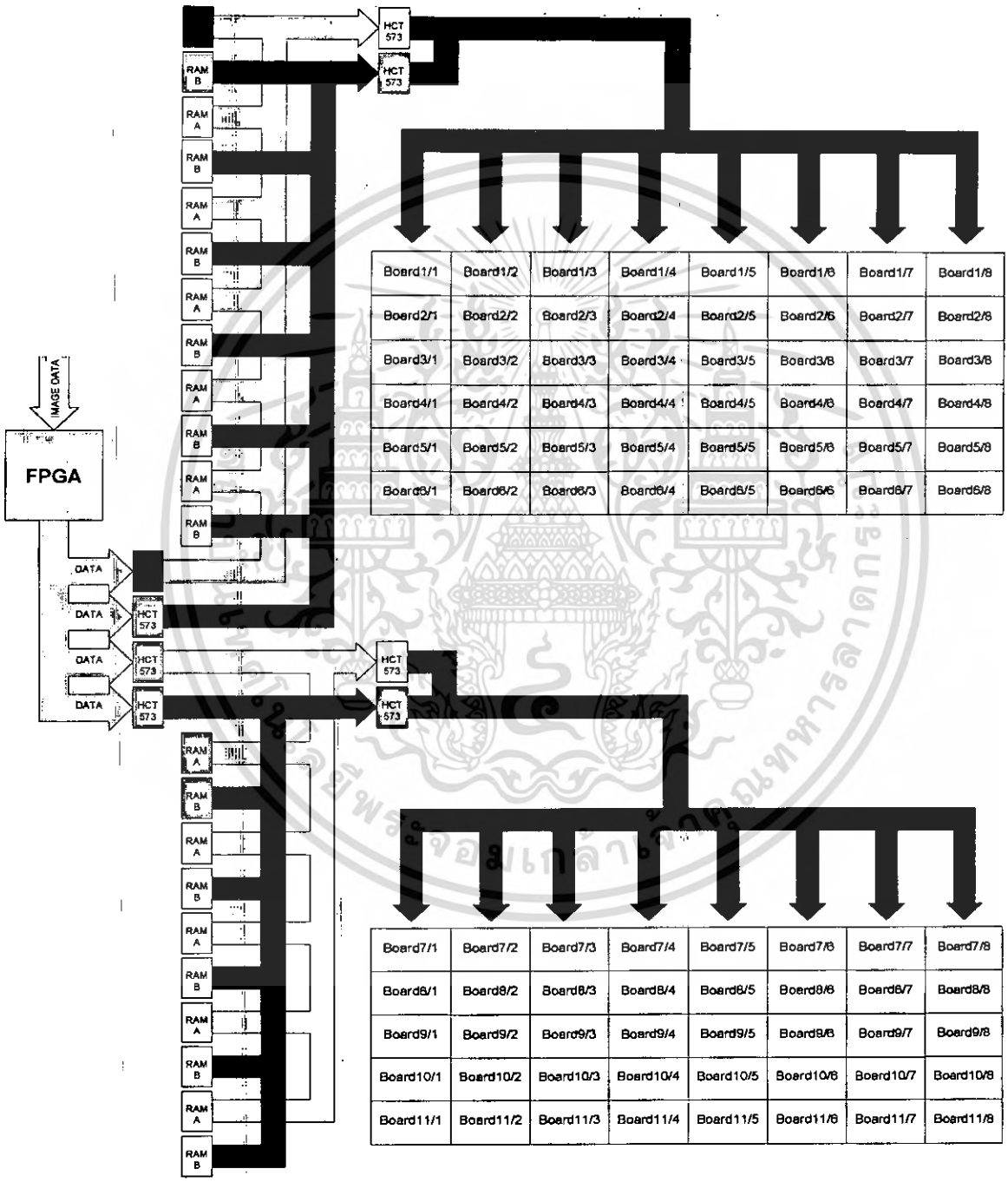
7. หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้ง เป็นการควบคุมคุณภาพของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบและการสร้าง

4.1 ลักษณะการเชื่อมต่อโดยรวมของระบบ



รูปที่ 4.1 ลักษณะการเชื่อมต่อของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก **รูปที่ 4.1** เป็นภาพรวมของเส้นทางของข้อมูล (Data Bus), เส้นทางของบัสตำแหน่ง (Address Bus) และเส้นทางของขาสแกน (Scan Bus) เราจะใช้บอร์ดแสดงผลเรียงกันดังรูปมีขนาด 11 × 8 บอร์ด ซึ่งแต่ละบอร์ดมีทั้งหมด 16 × 32 จุด ดังนั้นจอแสดงผลจะมีความละเอียด 45,056 จุด ในแต่ละจุดเราสามารถควบคุมได้ 2 สี คือ สีแดง กับสีเขียว โดยในแต่ละสีจะควบคุมความสว่างได้ 8 ระดับ ซึ่งเป็นการปรับความสว่างที่ Software อย่างเดียว

จากระบบที่แสดงข้างต้นจะเห็นได้ว่าจะแยกส่วนการสแกนและการส่งข้อมูลไปที่บอร์ดแสดงผลเป็น 2 ส่วน คือ ส่วนบนและส่วนล่างเพื่อความรวดเร็วในการสแกนแสดงผล ทำให้ความถี่ของการสแกนเร็วขึ้น และยังทำให้เราไม่ต้องใช้ความถี่ในการส่งข้อมูลที่สูงจนอาจทำให้เกิดปัญหาเรื่องความผิดเพี้ยนของสัญญาณและปัญหาสัญญาณรบกวน

เนื่องจากระบบต้องเป็นระบบแบบเวลาจริงตามอัตราการส่งข้อมูลของสัญญาณวีดีโอ ทำให้ในขณะที่เราแสดงผลอยู่นั้นเราต้องรับข้อมูลภาพถัดไปด้วยในเวลาเดียวกัน จึงเป็นเหตุผลให้ต้องมีการแยกระบบบัสข้อมูล และบัสตำแหน่ง ออกเป็นอีกชุดนึง เพื่อทำให้เราสามารถอ่านและเขียนหน่วยความจำแรมได้พร้อมกันโดยแรมก็จะมี 2 ชุดเช่นเดียวกัน ซึ่งในขณะที่ชุดนึงเป็นชุดเขียนอีกชุดนึงก็จะเป็นชุดอ่าน เมื่อมีสัญญาณเริ่มต้นภาพใหม่มาก็จะทำการสลับกันทำงาน โดยเอาภาพที่เก็บไว้มาแสดงผลแทน และจะเก็บภาพถัดไปเข้ามาอีกเหมือนเดิม

การออกแบบ จอ LED DISPLAY BOARD	
ขนาดจอภาพ	สูง 134.112 cm ยาว 195.072 cm (52.8" X 76.8")
จำนวน board	88 board สูง 11 board ยาว 8 board
อัตราส่วน จอภาพ	256 X 172 จุด (4 : 2.6875)
จำนวน จุดทั้งหมด	44032 จุด
จำนวน connector	176 ตัว
จำนวน สาย pair	176 เส้น ความยาวประมาณเส้นละ 1.42 ฟุต
electronic Characteristics	
Spark test	2500V
Conductor resistance	273 ohm /km (0.6m/ 0.1422 ohm)
Insulation resisance	1Gohm/m
Capacity	40PF/m
Inductance	1.45 microH/m
Impedance	100 ohm
Propagation Delay T	4.2 ns/m

รูปที่ 4.2 แสดงค่าทางกายภาพและคุณสมบัติทางไฟฟ้าของสาย Pair

4.2 SCANNING

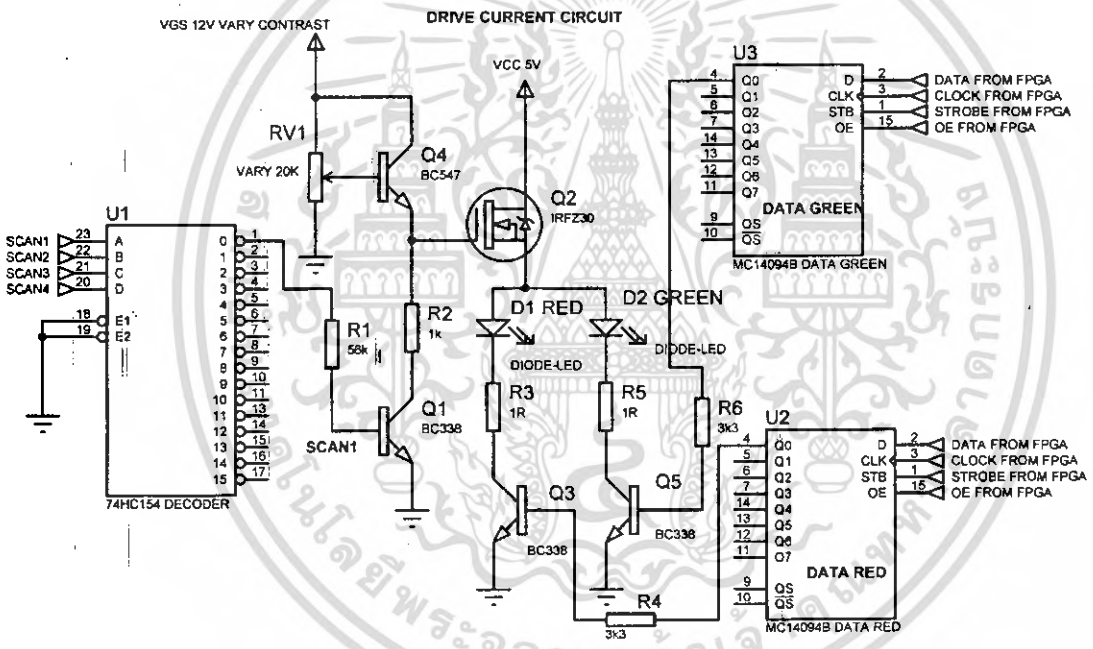
สำหรับส่วนที่ใช้ควบคุมการสแกนจะใช้ FPGA ควบคุมการสแกนในแนวแถว(Row scan) โดย FPGA จะส่งข้อมูลภาพที่เก็บไว้ไปยังไอซี โดยไอซีจะมีรีจิสเตอร์ ทำหน้าที่เก็บข้อมูลที่ส่งมาจาก FPGA และวงจรแลทช์ ทำหน้าที่พักข้อมูลก่อนส่งออกไปที่จอแสดงผล สำหรับการสแกนในแนว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอลัมน์นี้ จะต้องชิฟท์ข้อมูลครบทั้ง 32 คอลัมน์เนื่องจากรับข้อมูลแบบอนุกรม โดยข้อมูลทั้งหมดจะถูกเก็บไว้ภายในไอซี MC14094B เมื่อข้อมูลถูกส่งมาครบทั้ง 32 คอลัมน์ จากนั้นการสแกนก็จะเริ่มขึ้น โดยFPGA ก็จะส่งสัญญาณ OUTPUT ENABLE เพื่อให้ข้อมูลที่ถูกเก็บไว้ถูกส่งไปที่ บอร์ด LEDโดยจะเป็นการส่งข้อมูลแบบขนาน โดยจะทำการสแกนจากแถวที่ 1- 16 ตามลำดับ ซึ่งจะใช้ Decodeder 4 to 16 แล้วกลับมาเริ่มทำการสแกนใหม่อีกครั้ง

เนื่องจากการที่จะทำให้หลอด LED ติดสว่างนั้น จะต้องใช้วงจรขับกระแส เนื่องจากสัญญาณข้อมูลที่ออกมาจากวงจรบัฟเฟอร์ไม่สามารถที่จะจ่ายกระแสจำนวนมากได้ เราจึงจำเป็นที่จะต้องนำเอาสัญญาณเอาต์พุตที่ได้นี้ไปต่อร่วมกับวงจรขับกระแส

4.3 วงจรขับกระแส

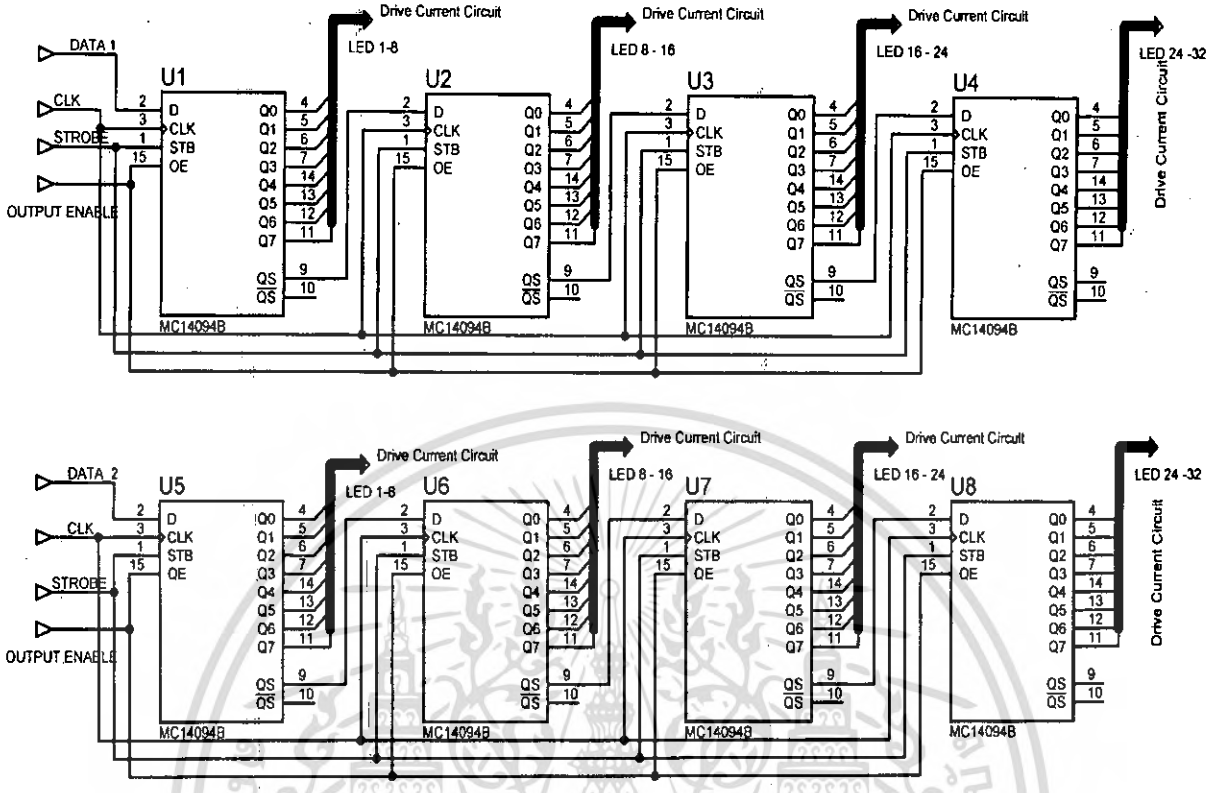


รูปที่ 4.3 วงจรขับกระแส

4.4 วงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์ และ รีจิสเตอร์

ในการแสดงภาพแต่ละภาพ ถ้าเราใช้ตัวเก็บข้อมูล หรือรักษาข้อมูลทุกจุด จะเป็นการสิ้นเปลืองอุปกรณ์มาก เราจึงใช้ไอซีรีจิสเตอร์ แบบมีแลทช์และบัฟเฟอร์ (8 –Stage Shift/Store register with Three- stage Outputs)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงส่วนของวงจรรักษาระดับข้อมูล

เนื่องจากข้อมูลที่จะนำมาแสดงผลนั้น มี 2 แบบ คือ

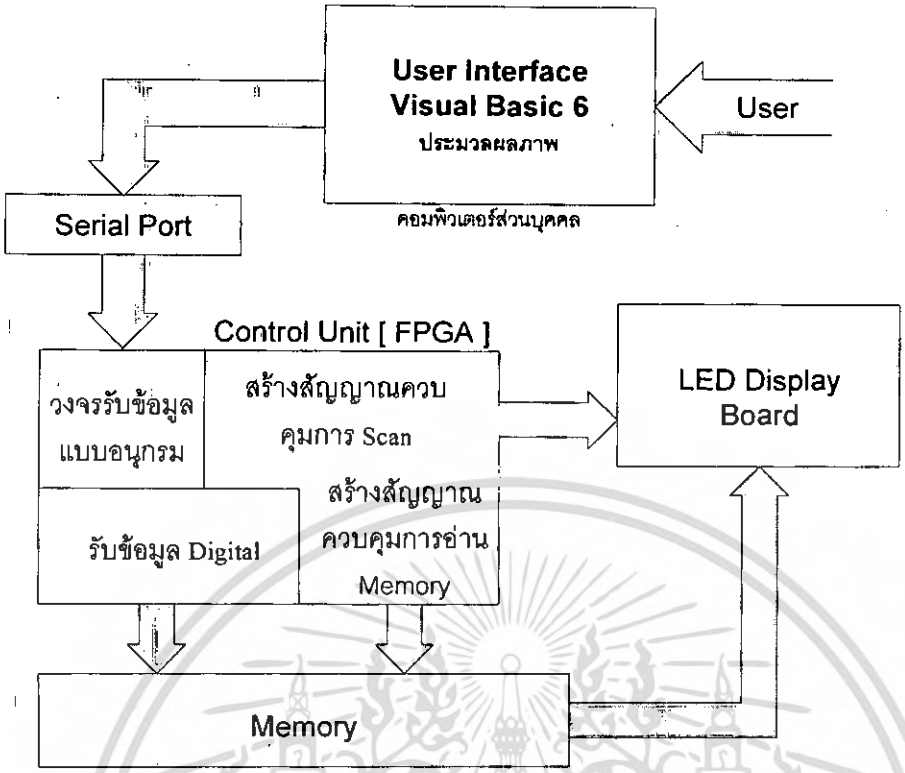
1. จากการประมวลผลภาพด้วยโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6

2. จากสัญญาณ VGA ที่แปลงเป็นสัญญาณดิจิตอล

4.5 ระบบแสดงภาพนิ่ง

ข้อมูลภาพนิ่งได้มาจากโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6 โดยจะทำหน้าที่ติดต่อกับผู้ใช้งาน (User interface) แล้วส่งข้อมูลผ่านทางพอร์ตอนุกรมโดยใช้ Baudrate 19200 บิตต่อวินาที ไปที่ FPGA ซึ่งจะทำหน้าที่เป็นตัวควบคุมการทำงาน โดยจะทำหน้าที่จัดการเก็บข้อมูลภาพลงหน่วยความจำแรม และสร้างสัญญาณควบคุมในการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 บล็อกไดอะแกรมโดยรวมของระบบแสดงภาพนิ่ง

4.5.1 โปรแกรมในส่วนของการติดต่อผู้ใช้งาน (User interface program)

โปรแกรมในส่วนนี้จะทำหน้าที่ในการติดต่อกับผู้ใช้งาน โดยผู้ใช้งานจะเป็นคนเลือกรูปภาพที่ต้องการแสดงผลเข้ามาในโปรแกรมเมื่อกดปุ่ม Compute โปรแกรมจะทำการประมวลผลภาพโดยการลดจำนวนสีให้เท่ากับจำนวนสีที่บอร์ดแสดงผล LED สามารถแสดงได้ ในที่นี้คือ 64 สี ประกอบด้วย สีแดง 8 ระดับ และสีเขียว 8 ระดับ เมื่อทำการลดจำนวนสีเรียบร้อยแล้วก็จะทำการจัดเรียงข้อมูลให้เหมือนกับข้อมูลที่จะเก็บเข้าไปในหน่วยความจำ SRAM เพื่อให้ FPGA อ่านออกมาแสดงผลต่อไป จากนั้นเมื่อต้องการจะให้ภาพแสดงผลที่ Board ให้กดปุ่ม Display โปรแกรมจะทำการส่งข้อมูลที่ประมวลผลแล้วผ่านทาง พอร์ตอนุกรมไปที่ FPGA

จาก **รูปที่ 4.6** จะเห็นได้ว่าสามารถใส่ภาพได้ 9 ภาพโดยการโหลดรูปเข้าที่หน้าต่างช่องต่างๆโดยการเลือกที่ Option Button ที่อยู่ทางด้านซ้ายมือ แล้วเลือก Open รูปภาพออกมาโดยชนิดของรูปภาพเป็นชนิด Bitmap File โดยการแสดงผลเราสามารถเลือกได้ว่าต้องการส่งภาพที่เท่าไหร่ถึงภาพที่เท่าไหร่ทำได้โดยป้อนตัวเลขเข้าไปที่ Text Box จำนวนรอบในการแสดงผลสามารถเลือกได้ในช่อง Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่4.6 ลักษณะหน้าต่างที่ใช้ติดต่อกับผู้ใช้งาน

4.5.2 วงจรภายใน FPGA

วงจรภายใน FPGA แบ่งออกเป็น 6 ส่วนดังนี้

1.) ส่วนของการควบคุมความถี่

2.) ส่วนของการรับข้อมูลจากพอร์ตอนุกรม

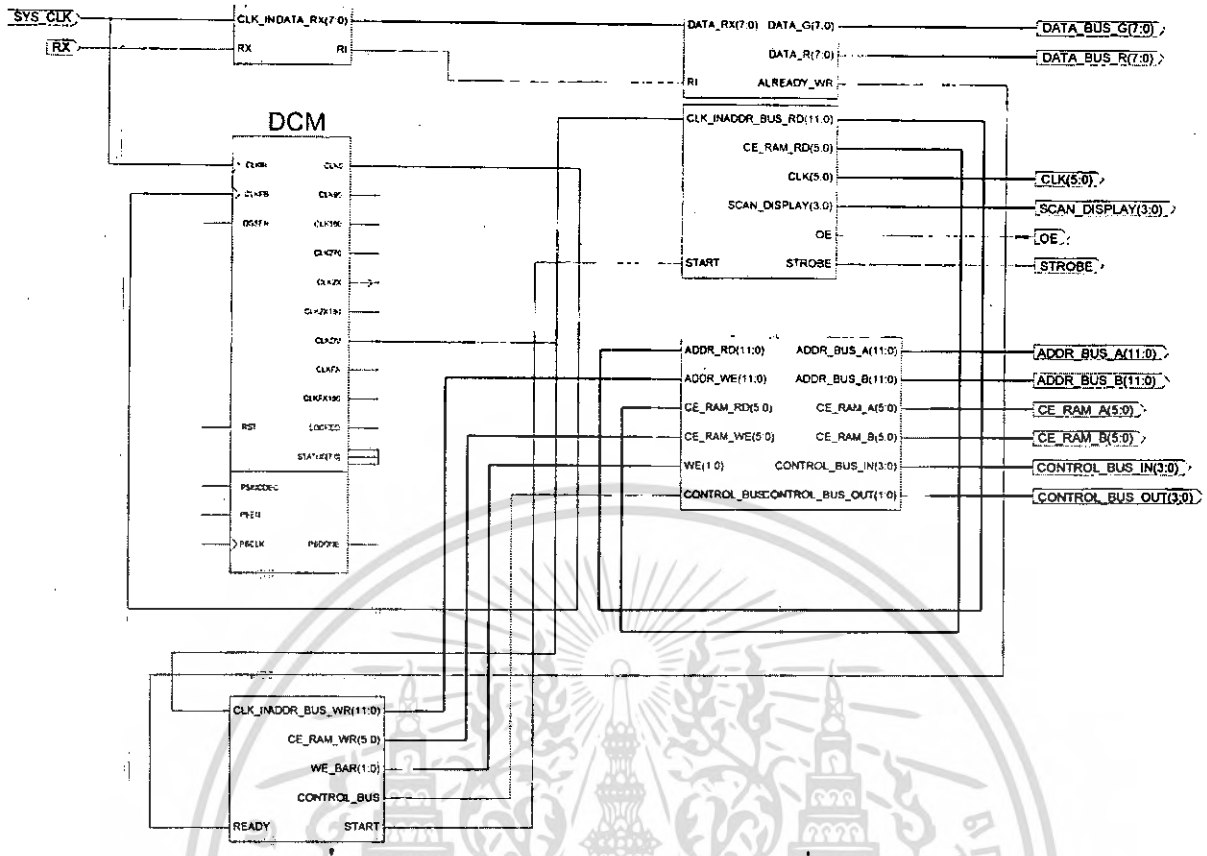
3.) ส่วนของการนำข้อมูลที่รับได้จากพอร์ตส่งออกไปที่ Data Bus

4.) ส่วนของการควบคุมการเขียน SRAM

5.) ส่วนของการควบคุมการอ่าน SRAM และการแสดงผล

6.) ส่วนของการควบคุมบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

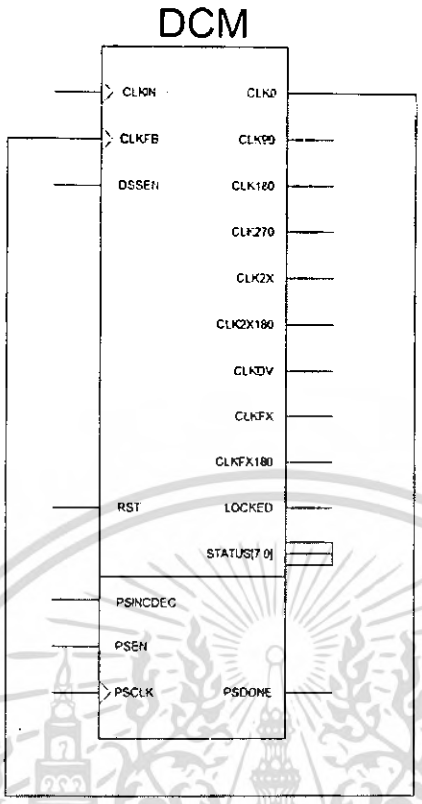


รูปที่ 4.7 วงจรภายใน FPGA ระบบแสดงภาพนิ่ง

รายละเอียดของวงจรแต่ละส่วนมีดังนี้

1.) ส่วนของการควบคุมความถี่

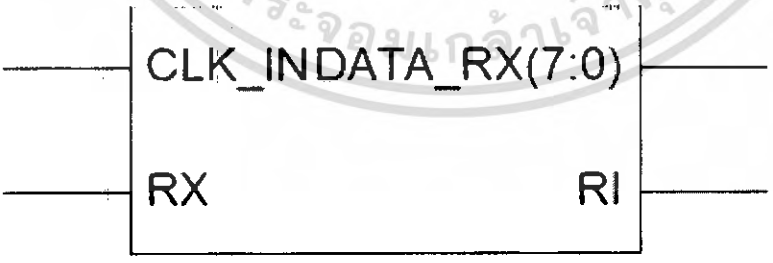
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ส่วนควบคุมความถี่

วงจรนี้ทำหน้าที่ในการหารความถี่ ซึ่งภายใน FPGA รุ่น Spartan3 นี้มีตัวจัดการกับความถี่อยู่ซึ่งมันสามารถหารความถี่ได้ โดยความถี่ที่หารได้นี้จะนำไปใช้ป้อนให้กับวงจรในส่วนของการควบคุมการอ่าน-เขียน SRAM และการแสดงผล เพื่อเป็นการปรับความถี่ของระบบการอ่าน-เขียน SRAM และการแสดงผลเพื่อไม่ให้ความถี่ที่ออกไปภายนอก FPGA สูงเกินไป

2.) ส่วนของการรับข้อมูลจากพอร์ตอนุกรม



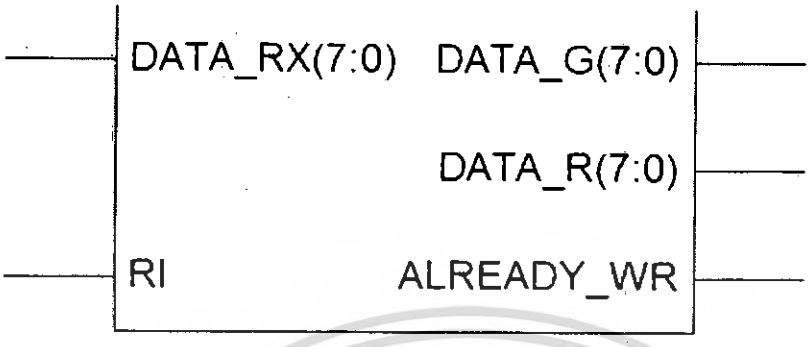
รูปที่ 4.9 ส่วนรับข้อมูลจากพอร์ตอนุกรม

วงจรนี้ทำหน้าที่ในการรับข้อมูลอนุกรม โดยข้อมูลอนุกรมจะเข้ามาทางขา Rx เมื่อข้อมูลเข้ามาครบ 1 Byte ก็จะทำให้ขาสัญญาณ RI มีสถานะเป็นขอบขาลง เพื่อเป็นการบอกวงจรส่วนถัดไปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำหน้าที่รับข้อมูลออกไปเก็บที่ SRAM รู้ว่าข้อมูลที่รับเข้ามาครบแล้ว ก็จะส่งข้อมูลออกไปทางขา DATA_RX (7:0) ซึ่งเป็นบัสขนาด 8 Bit

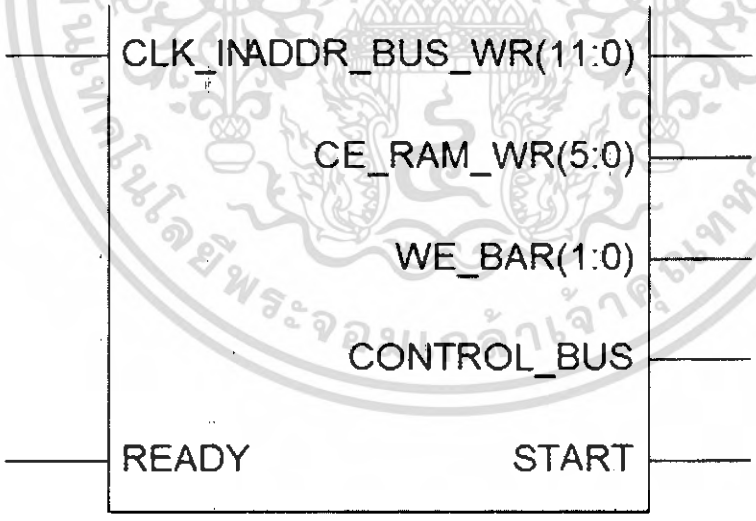
3.) ส่วนของการนำข้อมูลที่รับได้จากพอร์ตส่งออกไปที่ Data Bus



รูปที่ 4.10 ส่วนส่งข้อมูลออก Data Bus

วงจรในส่วนนี้ทำหน้าที่ในการเก็บข้อมูลที่ส่งมาจากส่วนของการรับข้อมูลจากพอร์ตอนุกรม เมื่อข้อมูลไบท์แรกที่รับได้จะเป็นข้อมูลของสีแดง ไบท์ที่สองเป็นสีเขียววงจรนี้จะส่งข้อมูลสีนี้ออกไปทาง Data Bus ของทั้ง 2 สี เมื่อข้อมูลออกไปที่ Data Bus เรียบร้อยแล้วก็จะส่งสัญญาณพร้อมเพื่อบอกวงจรส่วนควบคุมการเขียน รู้ว่ามีข้อมูลพร้อมที่จะเขียนลงสู่ SRAM แล้วโดยสัญญาณนี้เป็นสัญญาณขอบขาลงออกไปที่ขา ALREADY_WR

4.) ส่วนของการควบคุมการเขียน SRAM



รูปที่ 4.11 ส่วนควบคุมการเขียน SRAM

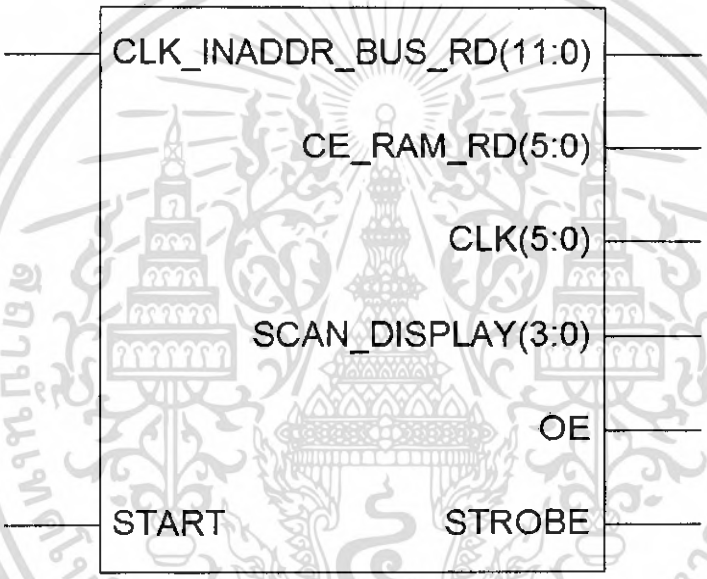
วงจรส่วนนี้จะทำหน้าที่ในการเขียนข้อมูลภาพลง SRAM โดยการทำงานเริ่มต้นเมื่อได้รับสัญญาณขอบขาลงที่ขา READY ก็จะเป็นจุดเริ่มต้นของการเขียนข้อมูล สัญญาณที่ใช้ในการเขียนข้อมูลคือ ADDRESS, CE และWE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการอ่านและการเขียนอิสระต่อกันและจะทำพร้อมกันได้แต่ในจังหวะแรกที่เริ่มต้นโหลดข้อมูลภาพลงไปเป็นภาพแรก ส่วนของการอ่านต้องยังไม่ทำงานจนกว่าภาพแรกจะโหลดเสร็จ โดยเมื่อภาพแรกโหลดเสร็จวงจรในส่วนนี้จะส่งสัญญาณ START ไปให้ส่วนของการอ่านข้อมูลเพื่อบอกจุดเริ่มต้นของการอ่านนั่นเอง

จากการที่กล่าวไว้เบื้องต้นแล้วว่าภายนอกมีการแยกระบบบัสออกเป็น 2 ส่วน เพื่อให้การอ่านและการเขียนแยกอิสระต่อกันวงจรในส่วนนี้จะส่งสัญญาณ CONTROL BUS เพื่อทำการสลับบัสในการอ่าน-เขียน SRAM โดยสัญญาณจะสลับไปมาตามจังหวะของการรับข้อมูลภาพ 1 ภาพเสร็จ

5.) ส่วนของการควบคุมการอ่าน SRAM และการแสดงผล

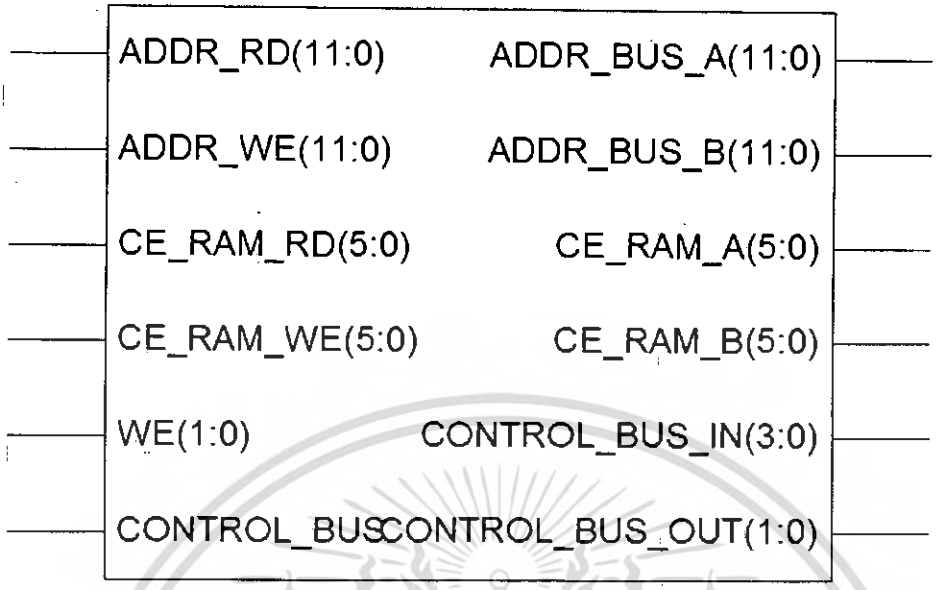


รูปที่ 4.12 ส่วนควบคุมการอ่าน SRAM และการแสดงผล

วงจรส่วนนี้ทำหน้าที่ในการอ่านข้อมูลภาพออกมาจาก SRAM และสร้างสัญญาณควบคุมการแสดงผล โดยวงจรส่วนนี้จะทำงานได้ก็ต่อเมื่อได้รับสัญญาณ START

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.) ส่วนของการควบคุมบัส



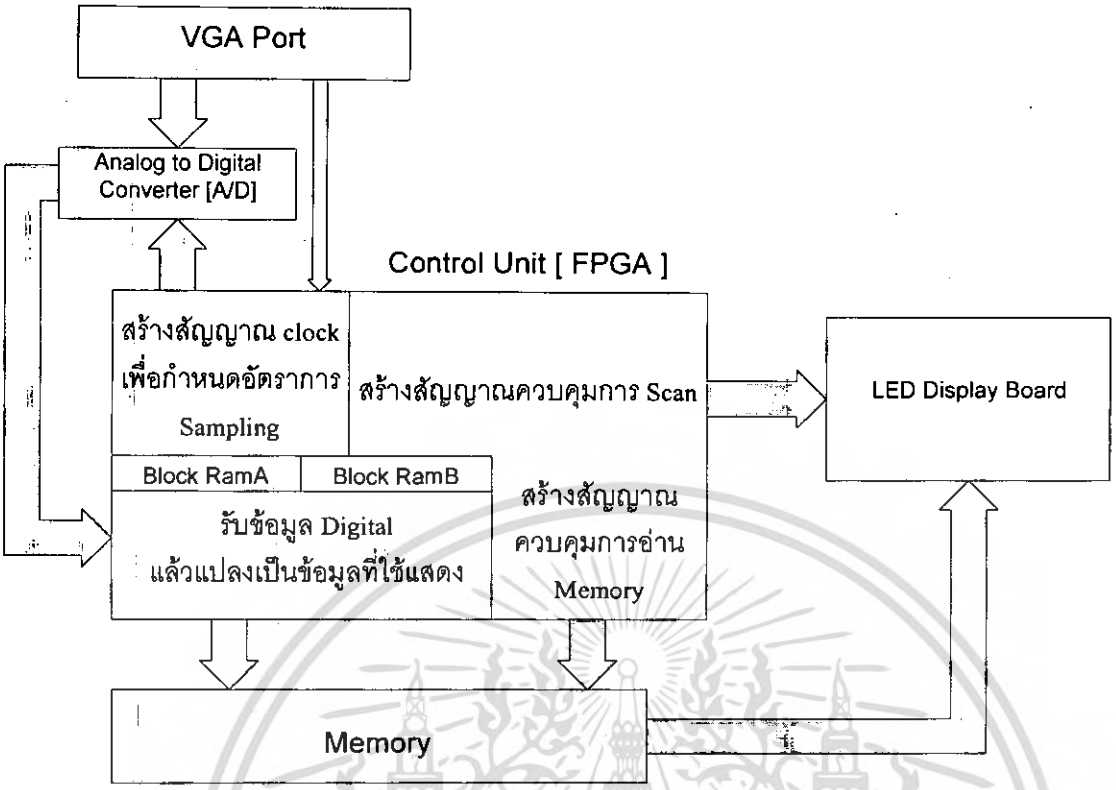
รูปที่ 4.13 ส่วนควบคุมบัส

วงจรในส่วนนี้ทำหน้าที่ในการจัดการเชื่อมต่อระบบบัสภายในกับภายนอกเข้าด้วยกัน โดยการเชื่อมต่อนี้จะขึ้นอยู่กับสัญญาณที่ส่งมาจากส่วนการเขียน SRAM เข้ามาที่ขา CONTROL BUS

4.6 ระบบแสดงภาพเคลื่อนไหว

สัญญาณภาพที่รับมาจาก VGA Port ของ Computer จะถูกแปลงเป็นสัญญาณดิจิตอล และป้อนเข้าสู่ FPGA จากนั้น FPGA จะทำการประมวลผลแปลงข้อมูลให้เหมาะสมกับ บอร์ดแสดงผล จากนั้นจะทำการเก็บข้อมูลลงไปในหน่วยความจำ เมื่อข้อมูลภาพเข้ามาจนเต็มภาพก็จะโหลดข้อมูลไปที่ บอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



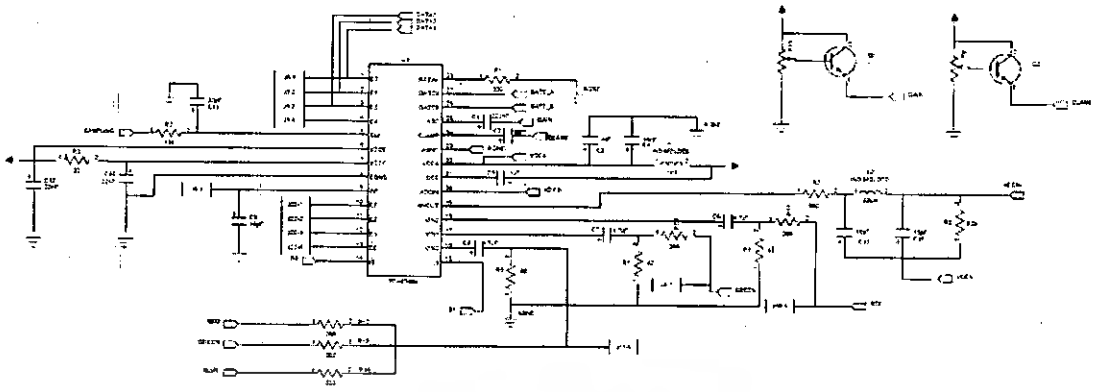
รูปที่ 4.14 บล็อกไดอะแกรมโดยรวมของระบบทั้งหมด

จากบล็อกไดอะแกรมดัง **รูปที่ 4.14** เป็นบล็อกไดอะแกรมโดยรวมทั้งหมดของระบบการแสดงผล ซึ่งประกอบไปด้วยส่วนของ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter), หน่วยความจำ (Memory), ส่วนแสดงผลด้วย LED (LED Display Board) และส่วนควบคุม (Control Unit) ซึ่งใช้ FPGA ซึ่งภายในเราจะแบ่งเป็นส่วนๆ โดยแต่ละส่วนจะทำหน้าที่แตกต่างกัน

4.6.1 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (A/D)

ส่วนของ Analog to Digital Converter (A/D) ส่วนนี้จะทำหน้าที่ในการแปลงสัญญาณอนาล็อกที่อยู่ในรูปของสัญญาณ VGA จากคอมพิวเตอร์ โดยใช้การสุ่ม (Sampling) ข้อมูลจากสัญญาณอนาล็อกให้กลายเป็นสัญญาณดิจิตอล เพื่อที่จะสามารถนำมาประมวลผลและเก็บข้อมูลลงหน่วยความจำ (Memory) ก่อนที่จะจัดส่งข้อมูลไปยังส่วนแสดงผล วงจรในส่วน A/D จะเป็นดัง **รูปที่ 4.15**

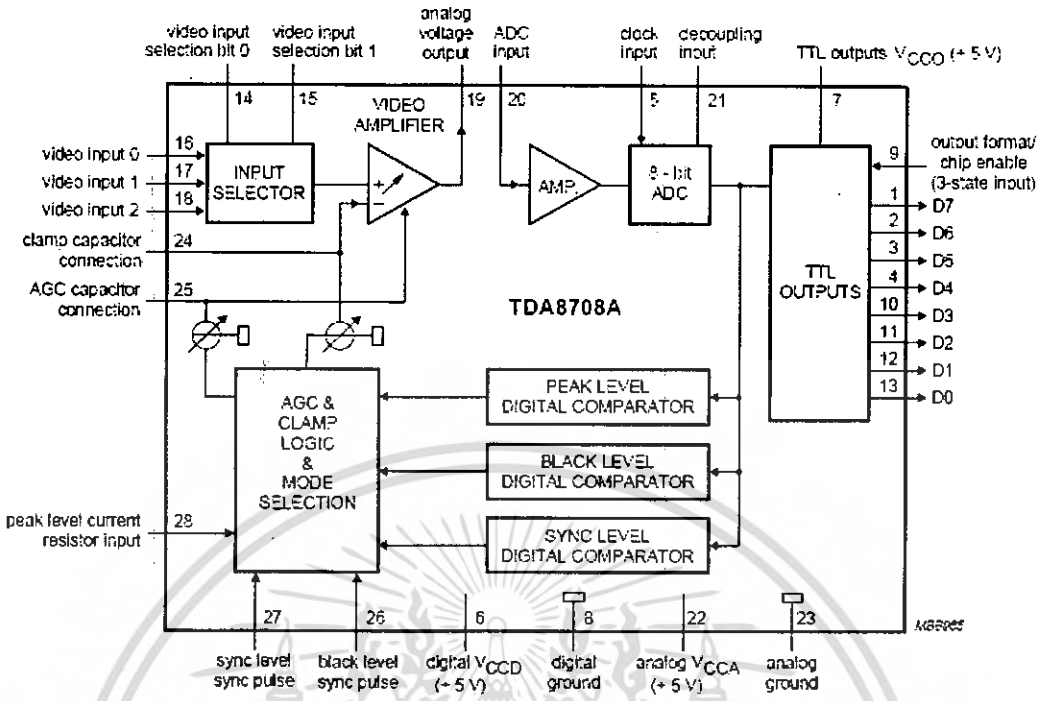
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (Analog to Digital Converter)

A/D ที่ใช้จะเป็นเบอร์ TDA8708A เป็นไอซีของบริษัทฟิลิปส์ ทำหน้าที่ในการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล ซึ่งโครงสร้างภายในของ TDA8708A ดังใน **รูปที่ 4.16** เหตุผลที่เลือกใช้ไอซีเบอร์นี้เพราะว่าเป็นไอซีที่ผลิตมาเพื่อใช้แปลงสัญญาณอนาล็อกที่เป็นสัญญาณวีดีโอโดยเฉพาะ โดยที่ภายใน TDA8708A จะประกอบด้วยส่วนของการเลือกสัญญาณอินพุทจากอินพุทช่องที่ 1,2 หรือ 3 สัญญาณที่ได้จะนำไปขยายโดยภาค Video Amplifier พร้อมทั้งทำการยกระดับของสัญญาณ และควบคุมเกณฑ์การขยายด้วยภาค Clamp and Gain Control หลังจากนั้นจึงนำสัญญาณไปแปลงจากสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล ไอซีเบอร์นี้ยังมีอัตราการสุ่มข้อมูลค่อนข้างสูงคืออยู่ที่ 32 MHz ซึ่งทำให้ได้ข้อมูลที่ละเอียดมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 โครงสร้างภายใน TDA8708A

แสดงการคำนวณหาอัตราสุ่ม (Sampling Rate) ของ Analog to Digital (A/D) สัญญาณ VGA ขนาดจอภาพ 600 × 800 มีความถี่สแกนทางแนวนอน (Horizontal Frequency) = 37.5 KHz

คาบเวลา (Time) ของ Hor.sync. = 1/ (Hor.Freq.) = 1/37.5 KHz = 26.6 μS

สัญญาณภาพจะมีคาบเวลา = 20 μS และมีสัญญาณ Blanking ก่อนที่จะมีสัญญาณภาพ 1.5 μS

ซึ่งใน 1 เส้นแนวนอนจะมีจำนวน Pixels ของภาพจำนวน 256 pixels ดังนั้น

คาบเวลา 1 Pixels จะหาได้จากData_Times/Data_Pixels =20μS/256pixels = 78.125 nS/1pixels

จะได้ความถี่ที่ใช้ในการสุ่ม (Sampling Rate) = 1/78.125nS = 12.8 MHz

ซึ่งสามารถใช้ Analog to Digital (A/D) ที่มี Sampling Rate 12.8 MHz ขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในค่ามีใช้งานการออกแบบใช้งานจริงจะใช้ A/D เบอร์ TDA8708A ซึ่งมีอัตราการสุ่มสัญญาณ (Sampling Rate) สูงสุด 32 MHz มีขนาด 8 บิด ซึ่งสามารถจัดระดับสัญญาณได้ 256 ระดับ เพิ่มที่สามารถใช้กับความถี่ Hor.Sync. ที่สูงๆได้ เช่นที่ Hor.Sync. 62.5 KHz ที่มีจำนวน Pixels 512 Pixels และ Hor.Sync. 31.25 KHz ที่จำนวน Pixels 1024 Pixels ซึ่งเป็นค่าสูงสุด (Maximum) ที่ A/D จะสามารถทำงานได้ และไอซีเบอร์ TDA8708A เหมาะสำหรับการแปลงสัญญาณอนาล็อกที่เป็นสัญญาณวีดีโอโดยเฉพาะ

4.6.2 วงจรภายใน FPGA

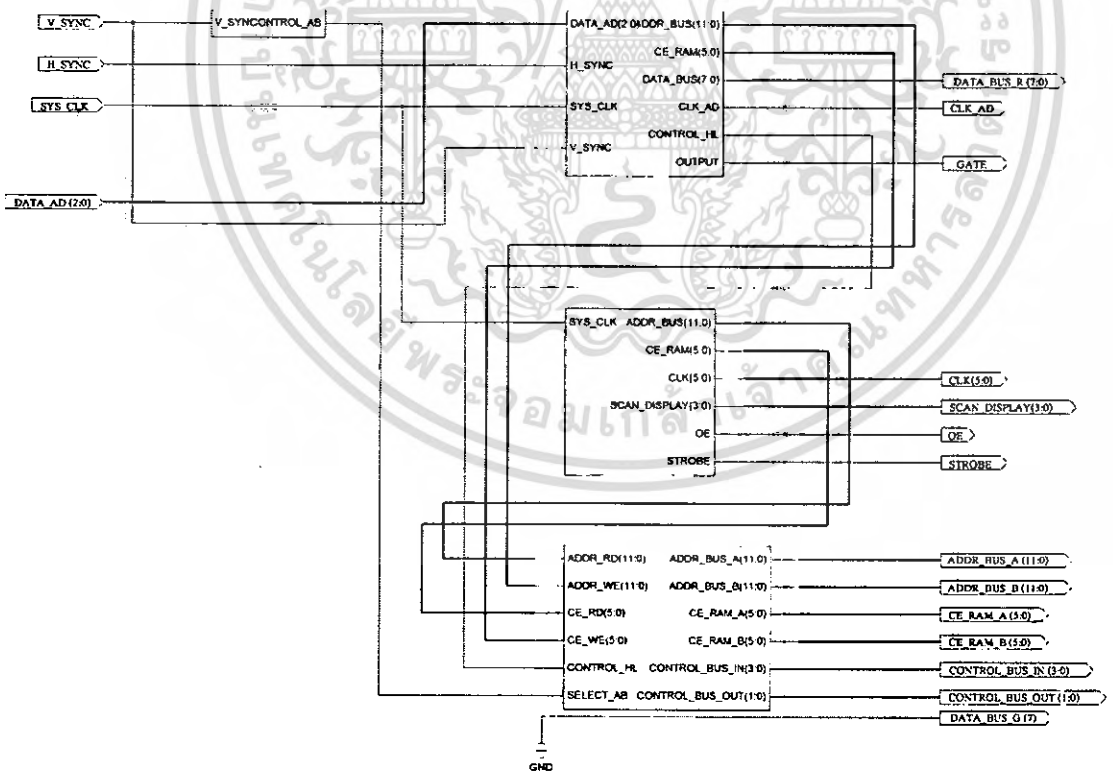
วงจรภายใน FPGA แบ่งออกเป็น 4 ส่วนดังนี้

1.) ส่วนสร้างสัญญาณควบคุมการเชื่อมต่อบัส

2.) ส่วนสร้างสัญญาณ Sampling และเขียนข้อมูลวีดีโอลงหน่วยความจำ SRAM

3.) ส่วนอ่านข้อมูลและควบคุมการแสดงผล

4.) ส่วนเชื่อมต่อบัสภายในกับบัสภายนอก



รูปที่ 4.17 วงจรภายใน FPGA ส่วนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อข้อมูลดิจิตอลออกมาเราจะต้องทำการรับข้อมูล ตามจังหวะการส่งข้อมูล โดยขั้นแรกเราจะต้องรอสัญญาณเริ่มต้นภาพหรือสัญญาณซิงก์ในด้านแนวตั้ง VS (Vertical Synchronization) เพื่อให้รู้จุดเริ่มต้นของข้อมูลภาพ จากนั้นจะมีสัญญาณแนวนอน HS (Horizontal Synchronization) ซึ่งจะเป็นตัวบอกจุดเริ่มต้นของเส้นภาพในแต่ละแถว หลังจากนั้นข้อมูลภาพดิจิตอลจะออกมาตามจังหวะของสัญญาณนาฬิกา (Clock) หรือตัวที่กำหนดจังหวะการสุ่ม (Sampling) ของสัญญาณวีดีโอ เนื่องจากตัว A/D จะสุ่มสัญญาณในขอบขาขึ้นของสัญญาณนาฬิกาแต่การทำงานจะมีช่วงเวลาหน่วง (Delay Time) ดังนั้นข้อมูลจะปรากฏอย่างถูกต้องที่ขอบขาลงของสัญญาณนาฬิกา ดังนั้นเราจะเก็บข้อมูลดิจิตอลที่เวลานี้

เนื่องจากความสว่างที่ต้องการทำมี 8 ระดับดังนั้นเราจะใช้ข้อมูลดิจิตอลจำนวน 3 บิตโดยเลือกจากบิตที่มีความสำคัญสูง(MSB) จากนั้นจะมีการแปลงเป็น 7 บิต เพื่อใช้ในการแสดงผล 7 รอบเพื่อควบคุมความสว่างของ LED ให้ได้ 8 ระดับ โดยครั้งที่มีความสว่างมากที่สุดจะมีรหัสดิจิตอลเป็น “1111111” รองลงมาคือ “0111111” และความสว่างน้อยที่สุดเป็น “0000000” เมื่อบิตเป็น “1” คือ ให้ LED ดิด ส่วน”0” ให้ LED ดับจะเห็นได้ว่ากรณีที่ รหัสดิจิตอลเป็น “1111111” LED จะติดทั้งหมด 7 ครั้งใน 1 รอบการสแกนทำให้มีความสว่างสูงที่สุด

จากที่กล่าวมาแล้วในเรื่องรหัสของความสว่าง 7 บิต จะขอเรียกว่า เป็น 7 ชั้น (Layer) โดยบิตที่ 0 จะเรียกว่า Layer 1 การแสดงผลใน 1 เส้นภาพหรือ 1 แถว จะแสดงทีละ Layer โดยเริ่มที่ Layer1 จนครบจากนั้นจึงเปลี่ยนแถวสแกนถัดไป

รายละเอียดของวงจรแต่ละส่วนมีดังนี้

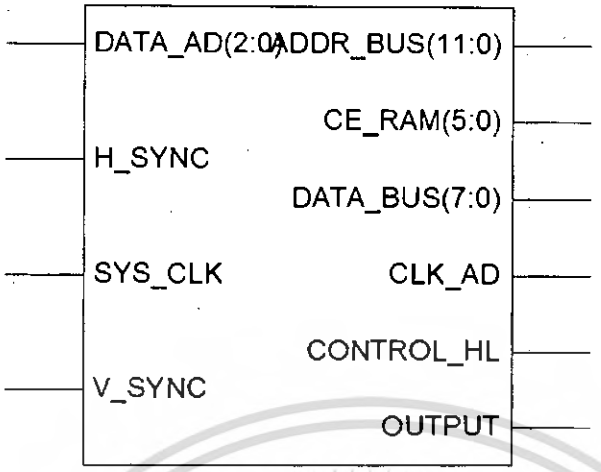
1.) ส่วนสร้างสัญญาณควบคุมการเชื่อมต่อบัส

รูปที่ 4.18 ส่วนสร้างสัญญาณควบคุมการเชื่อมต่อบัส

วงจรส่วนนี้ทำหน้าที่กำเนิดสัญญาณควบคุมการเปลี่ยนบัส อ่าน-เขียน SRAM ภายนอก โดย Data Bus ภายนอกมี 2 ชุดดังที่กล่าวมาแล้ว สัญญาณตัวนี้จะส่งต่อไปยังส่วนเชื่อมต่อบัสภายในกับบัสภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ส่วนสร้างสัญญาณ Sampling และเขียนข้อมูลวีดีโอลงหน่วยความจำ SRAM



รูปที่ 4.19 ส่วนสร้างสัญญาณ Sampling และเขียนข้อมูลวีดีโอลงหน่วยความจำ SRAM

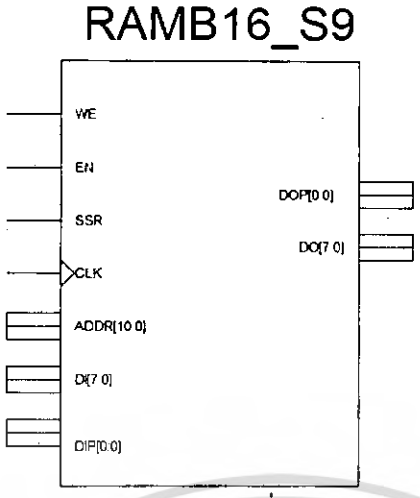
วงจรส่วนนี้ทำหน้าที่ในการสร้างสัญญาณ Sampling และสัญญาณควบคุมเพื่อทำให้ A/D ส่งข้อมูล ดิจิตอล Video ออกมา และส่วนนี้ยังจัดการในการรับข้อมูล ดิจิตอล Video แล้วทำการเรียงข้อมูลที่ได้มาแล้วเขียนลง SRAM ด้วย

จากการทำงานของจอแสดงผลเป็นลักษณะของเวลาจริง(Real time) ดังนั้นข้อมูลจะถูกส่งเข้ามาอย่างต่อเนื่องตลอดเวลา ในขณะเดียวกันนั้นข้อมูลก็จะต้องถูกแสดงไปที่จอนั่นหมายความว่าในเวลาหนึ่งๆ จะต้องมีการรับข้อมูลใหม่เข้ามาพร้อมๆกับมีการ ส่งข้อมูลภาพไปแสดงที่จอ ดังนั้นในระบบการเก็บข้อมูลจะต้องแบ่งออกเป็น 2 ส่วน

ภายใน FPGA จะต้องมีหน่วยความจำ 2 ส่วนเพื่อรับข้อมูลเส้นภาพถัดไป และข้อมูลที่จะเขียนลง SRAM โดยในการออกแบบนี้ ภายใน FPGA มีหน่วยความจำขนาดใหญ่ที่เรียกว่า Block Ram **(รูปที่ 4.20)** อยู่ถ้าเราใช้หน่วยความจำชนิดนี้มีข้อดีคือเราจะไม่สิ้นเปลืองเกตที่อยู่ภายใน โดยแรมชนิดนี้สามารถใช้ความถี่ได้สูงสุด 200 MHz โดยหน่วยความจำสองส่วนนี้จะออกแบบระบบบัสคล้ายกับภายนอก คือ เป็นบัสอ่าน – เขียนสลับกันไปมา โดยจะใช้ Data Selector เป็นตัวเลือกการเชื่อมต่อบัสภายใน การสลับบัสจะเป็นไปตามจังหวะสัญญาณ H_SYNC แทน

เนื่องจากข้อมูลที่รับมาจะต้องมีการการเรียงข้อมูล 8 บิต ลงใน SRAM จะเรียงตามวงจรที่ออกแบบไว้ซึ่งจะกล่าวต่อไปในส่วนของการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดง Block Ram ที่ใช้ในการออกแบบ

การคำนวณขนาดของ SRAM

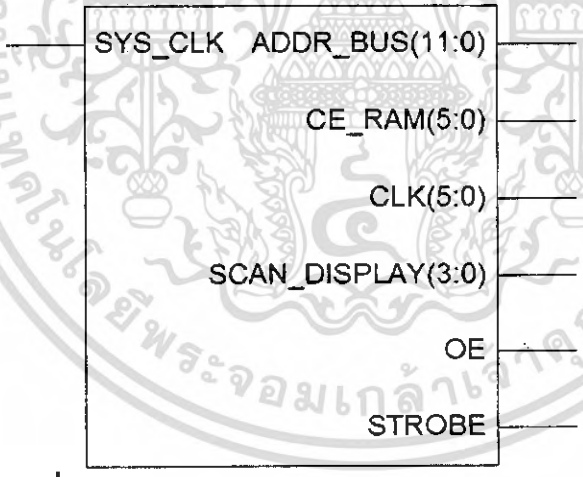
SRAM 1 ตัวเก็บข้อมูล 8 บอร์ด

1 บอร์ดแสดงผลมีทั้งหมด 512 จุดภาพ

1 จุดภาพ ต้องใช้หน่วยความจำ 7 บิต

ดังนั้น SRAM จะต้องมีขนาด = 8 × 512 ×7 = 28672 bits = 3.5 Kbytes

3.) ส่วนอ่านข้อมูลและควบคุมการแสดงผล



รูปที่ 4.21 ส่วนอ่านข้อมูลและควบคุมการแสดงผล

ในส่วนนี้จะทำการอ่านข้อมูลจาก SRAM ไปที่บอร์ดแสดงผลซึ่งการอ่านข้อมูลจะเป็นการอ่านข้อมูลเรียง Address เนื่องจากมีการจัดข้อมูลให้พร้อมที่จะแสดงผลอยู่แล้ว ข้อมูลที่จะเข้าไปแสดงผลนั้นจะมีการเลื่อนของข้อมูลเรียงต่อกันในแต่ละบอร์ดแสดงผลจำนวน 32 จุดภาพ ดังนั้นข้อมูลแรกที่จะเข้าไปก็คือ จุดที่ 1 ของภาพ

ในแต่ละจุดภาพจะมีการแสดงผลทั้งหมด 7 ครั้ง ให้ได้ความสว่างที่แตกต่างกัน 8 ระดับ โดยข้อมูลแรกในการแสดงคือ Layer 1 ดังนั้นลำดับของข้อมูลจะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่1 Layer 1 - จุดที่2 Layer 1 - … - จุดที่256 Layer 1

จุดที่1 Layer 2 - จุดที่2 Layer 2 - … - จุดที่256 Layer 2

.

.

.

จุดที่1 Layer 7 - จุดที่2 Layer 7 - … - จุดที่256 Layer 7

นั่นเป็นลำดับของแต่ละบอร์ดแสดงผล แต่ข้อมูลที่ออกจากบัสข้อมูลของ SRAM จะต้องเข้าไปในบอร์ด 8 บอร์ดตามแนวนอนดังนั้นถ้ามองภาพรวมของภาพ เราจะพบว่า SRAM จะส่งข้อมูล ตามลำดับดังนี้

จุดที่1 Layer 1 – จุดที่33 Layer 1 - … - จุดที่225 Layer 1

จุดที่2 Layer 1 – จุดที่34 Layer 1 - … - จุดที่226 Layer 1

.

.

.

จุดที่1 Layer 2 – จุดที่33 Layer 2 - … - จุดที่225 Layer 2

จุดที่2 Layer 2 – จุดที่34 Layer 2 - … - จุดที่226 Layer 2

.

.

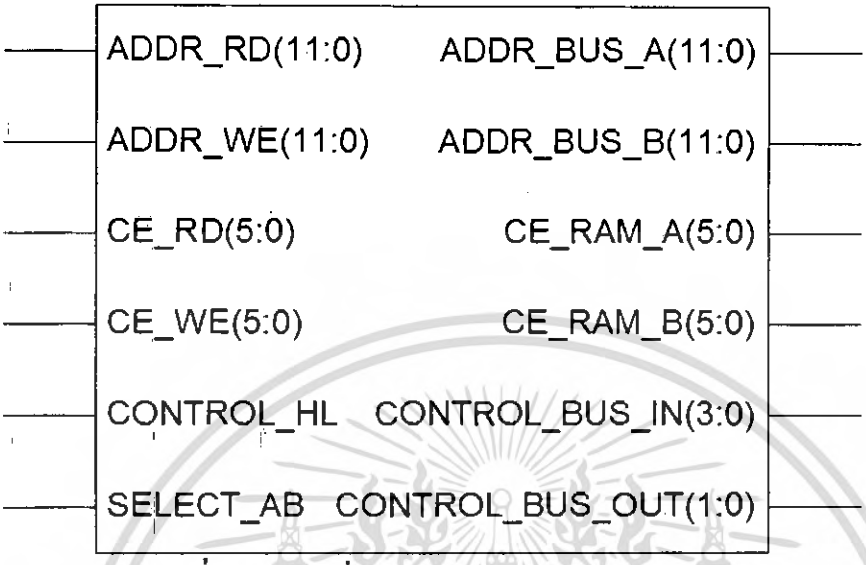
.

จุดที่1 Layer 7 – จุดที่33 Layer 7 - … - จุดที่225 Layer 7

จุดที่2 Layer 7 – จุดที่34 Layer 7 - … - จุดที่226 Layer 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) ส่วนเชื่อมต่อบัสภายในกับบัสภายนอก



รูปที่ 4.22 ส่วนเชื่อมต่อบัสภายในกับบัสภายนอก

วงจรส่วนนี้จะทำหน้าที่ในการเลือกการเชื่อมต่อ Data Bus, Address Bus และ CE ที่ใช้เลือก SRAM โดยจะใช้สัญญาณจากส่วนสร้างสัญญาณควบคุมการเชื่อมต่อบัสมาทำการสลับการเชื่อมต่อ กับบัสภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

จากการออกแบบแบ่งออกเป็น 2 ระบบคือ

1. ระบบแสดงภาพนิ่ง

2. ระบบแสดงภาพเคลื่อนไหว

เนื่องจากคณะผู้จัดทำมีเวลาจำกัดทำให้ไม่สามารถทำระบบภาพเคลื่อนไหวได้สำเร็จแต่สามารถแปลงสัญญาณวีดีโอให้เป็นสัญญาณดิจิตอลได้ส่วนของระบบแสดงภาพนิ่งผ่านทางโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6 นั้นทำสำเร็จจึงแบ่งผลการทดลองออกเป็น 2 ส่วนคือ

1. ส่วนของวงจรแปลงสัญญาณวีดีโอให้เป็นสัญญาณดิจิตอล

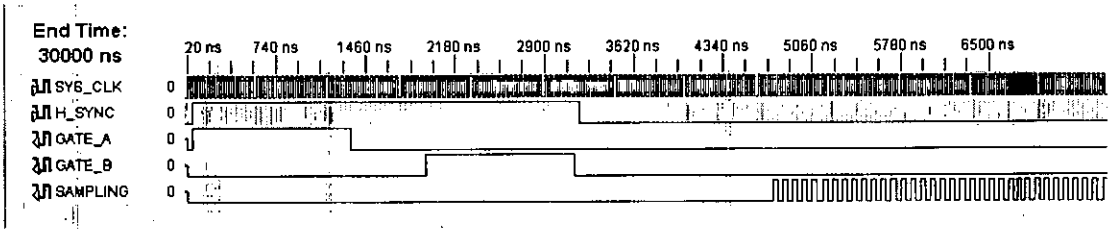
2. ส่วนของการแสดงภาพนิ่งผ่านทางโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6

โดยการทดลองนั้นจะทำ 2 ส่วน คือ ส่วนของการจำลองการทำงานในโปรแกรม ISE WebPack ของ Xilinx ซึ่งทางคณะผู้จัดทำได้ออกแบบโดยการเขียนภาษา VHDL แล้วทำการ Simulation ดูผลการทำงานของวงจรในโปรแกรมก่อน เมื่อผลการทำงานตรงตามที่ได้ออกแบบไว้จึงจะทำการสังเคราะห์วงจรแล้วโปรแกรมลงบนชิพ FPGA จริง และในส่วนที่ 2 คือส่วนของการทดลองจริง เป็นการทดลองกับฮาร์ดแวร์ที่ได้ออกแบบไว้

5.1 ส่วนของวงจรแปลงสัญญาณวีดีโอให้เป็นสัญญาณดิจิตอล

5.1.1 ผลการทดลองจากการ Simulation

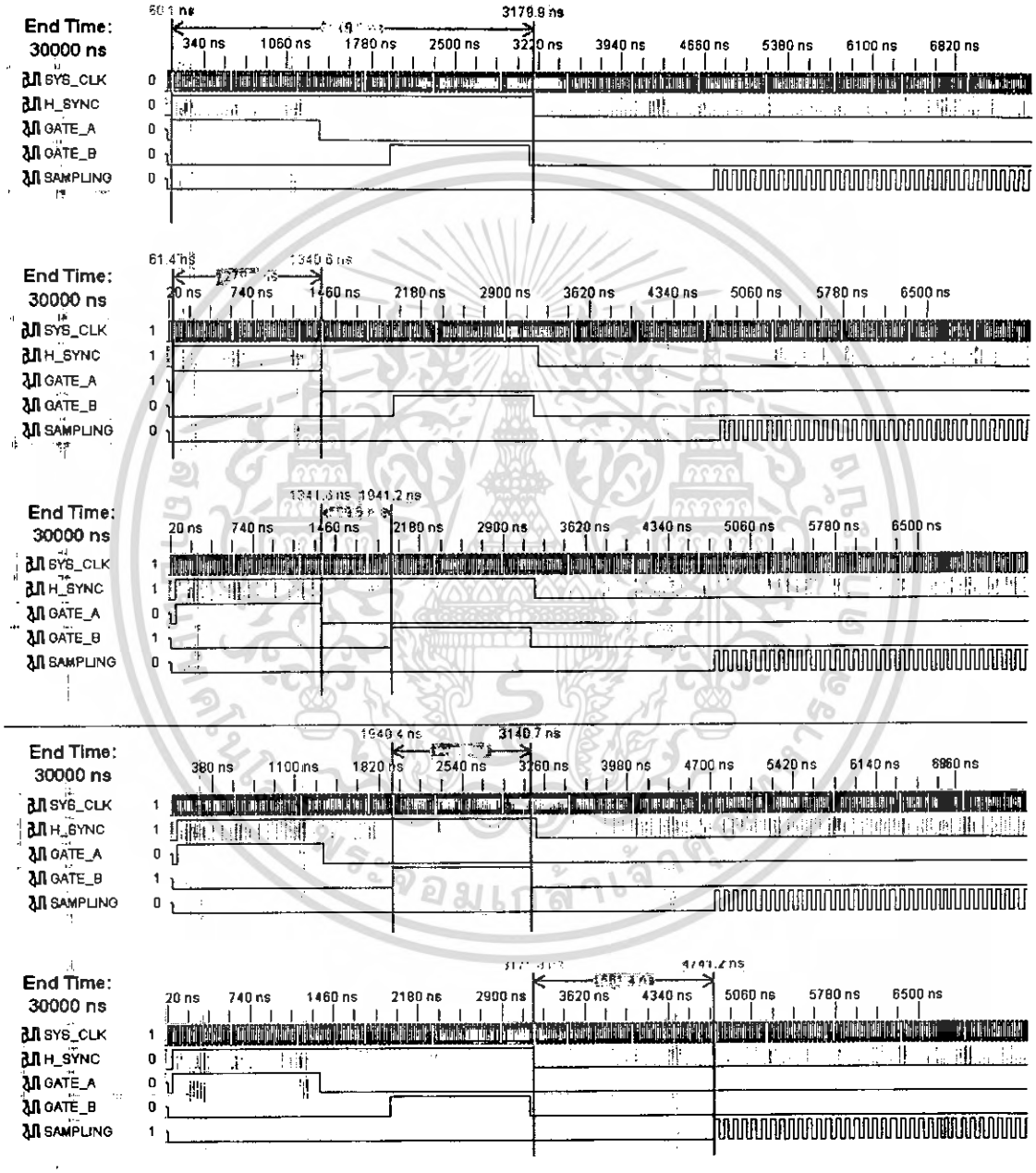
การ Simulation นี้ เป็นการ Simulation แบบดู Timing ของสัญญาณเมื่อ FPGA ได้รับสัญญาณวีดีโอ และสร้างสัญญาณควบคุมให้กับ A/D



รูปที่ 5.1 ผลการ Simulation จากโปรแกรม ISE WebPack ของ Xilinx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก**รูปที่ 5.1** เป็นผลจากการ Simulation ของวงจรสร้างสัญญาณที่ใช้ในการแปลงสัญญาณวีดีโอซึ่งเป็นสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลโดยในรูปแสดงสัญญาณ H_SYNC ซึ่งเป็นสัญญาณอินพุทที่รับเข้ามาจากสาย VGA แล้วมีสัญญาณ GATE_A, GATE_B และSAMPLING เป็นสัญญาณเอาต์พุทออกไปยังวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลที่อยู่ภายนอกต่อไป



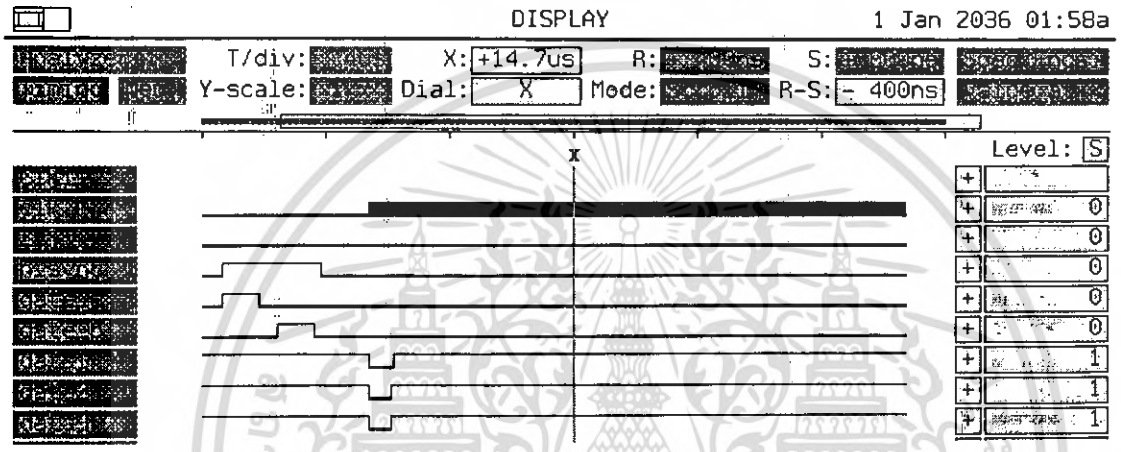
รูปที่ 5.2 แสดงขนาดความกว้างของสัญญาณต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก**รูปที่ 5.2** เป็นการแสดงขนาดความกว้างของสัญญาณ H_SYNC, GATE_A, ช่วงระยะห่างระหว่างสัญญาณ GATE_A และ GATE_B, GATE_B และช่วงระยะห่างของการเกิดสัญญาณ SAMPLING หลังจากที่สัญญาณ H_SYNC เป็นศูนย์

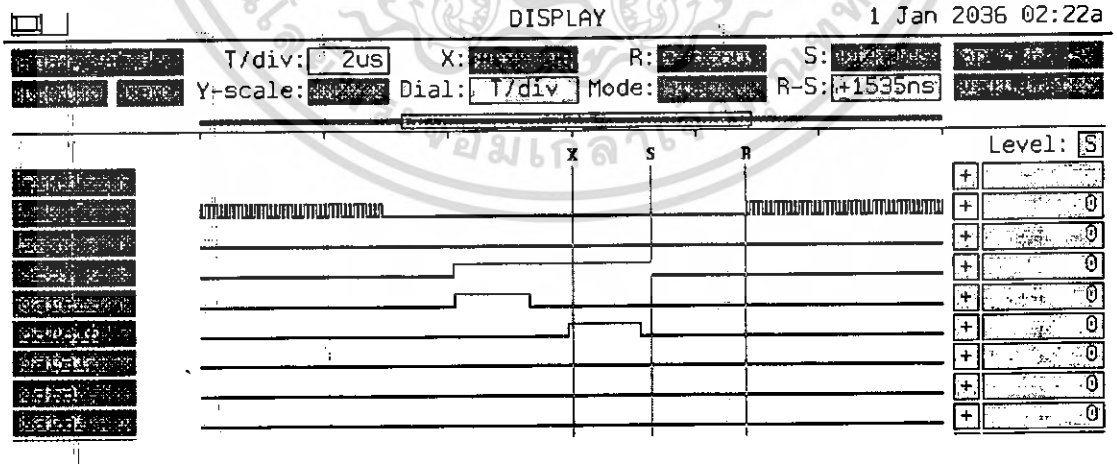
5.1.2 ผลการทดลองจากการทดลองทางฮาร์ดแวร์

จากการทดลองโดยใช้ TDA8708A ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลจะได้รูปสัญญาณดัง**รูปที่ 5.3**



รูปที่ 5.3 แสดงส่วนของสัญญาณ Hor_Sync และ สัญญาณ Sampling

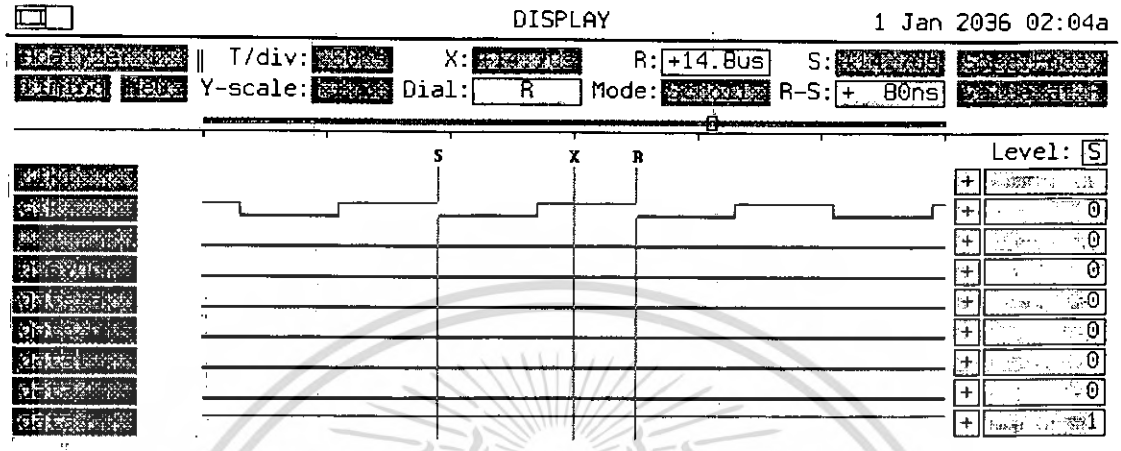
จาก**รูปที่ 5.3** สัญญาณ Hor_Sync ที่ได้จากการวัดจะมีคาบของสัญญาณเท่ากับ 26.5 µS หรือ 37.5 KHz



รูปที่ 5.4 แสดงส่วน Blanking ก่อนที่จะมีสัญญาณ Sampling

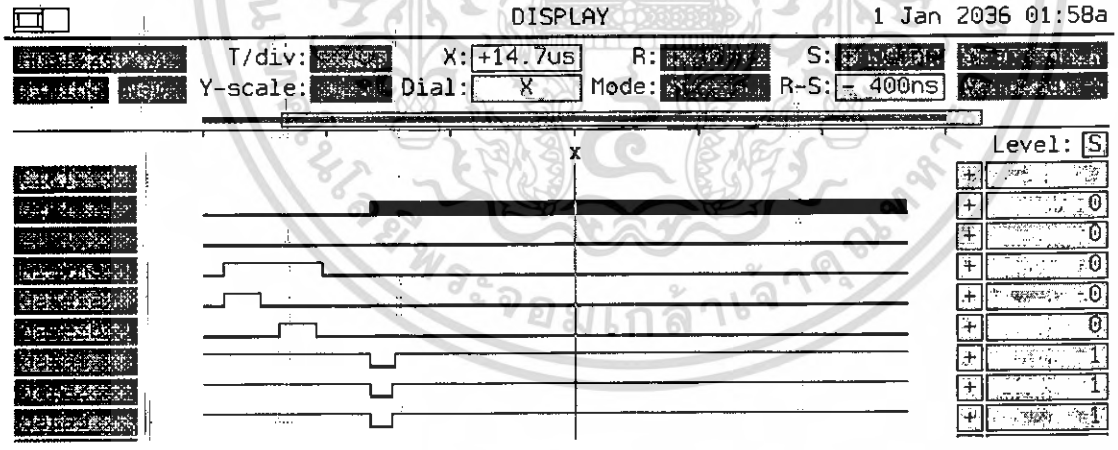
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก**รูปที่ 5.4** ช่วง Blanking ก่อนที่จะมีสัญญาณ Sampling จะมีช่วงเวลา 1.535 µS โดยช่วง Blanking นี้จะช่วยกำหนดจุดเริ่มต้นของเฟรมภาพในแต่ละแนวเส้นสแกนซึ่งเส้นสแกนภาพใน โครงงานนี้จะใช้จำนวนเส้นสแกนทั้งหมด 176 เส้นภาพใน 1 ภาพ



รูปที่ 5.5 แสดงส่วนของสัญญาณ Sampling

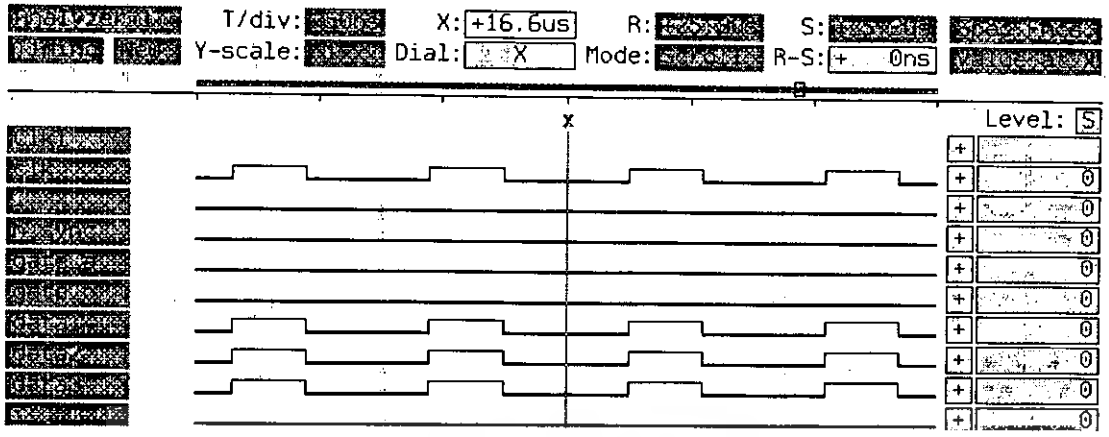
จาก**รูปที่ 5.5** แสดงสัญญาณ Sampling โดยกำหนดความถี่ในการ Sampling เท่ากับ 12.5 MHz หรือใช้เวลาในแต่ละคาบของสัญญาณเท่ากับ 80 nS โดยสัญญาณ Sampling ที่ได้จะนำไปใช้กับ Analog to Digital Converter ซึ่งในโครงงานนี้จะใช้ IC เบอร์ TDA8708A



รูปที่ 5.6 แสดงส่วนของสัญญาณ Digital ที่ได้จากการ Sampling

จาก**รูปที่ 5.6** แสดงสัญญาณ Digital 3 บิตโดยใช้อัตราการ Sampling 12.5 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงส่วนขยายของสัญญาณ Sampling และสัญญาณข้อมูลที่ได้ทำการแปลงเป็นสัญญาณดิจิตอล

5.2 ส่วนของการแสดงภาพนิ่งผ่านทางโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6

ส่วนของการแสดงผลนี้จะมีการเขียน และการอ่านข้อมูลออกมาแสดงผล โดยการทดลองแบ่งออกเป็น 2 ส่วน คือ ส่วนของการ Simulation และส่วนของการทดลองจริงกับ Hardware

5.2.1 ผลการ Simulation

จุดประสงค์ของการทำ Simulation นี้คือ ทำให้การทำงานระดับ Function ถูกต้องตามที่เราได้ออกแบบไว้โดยใช้การ Simulation แบบดู Timing เนื่องจาก FPGA เป็นการออกแบบ Hardware ไม่ใช่แบบ Software บางครั้งถ้าเราต้องการดูผลทั้งหมดอาจเป็นไปไม่ได้เนื่องจากการประมวลผลของ Computer มีจำกัดเราจึงจำเป็นต้องลดจำนวน Loop ในการสร้างสัญญาณควบคุมต่างๆ เพื่อให้เราสามารถดูผลของมันได้กว้างขึ้น

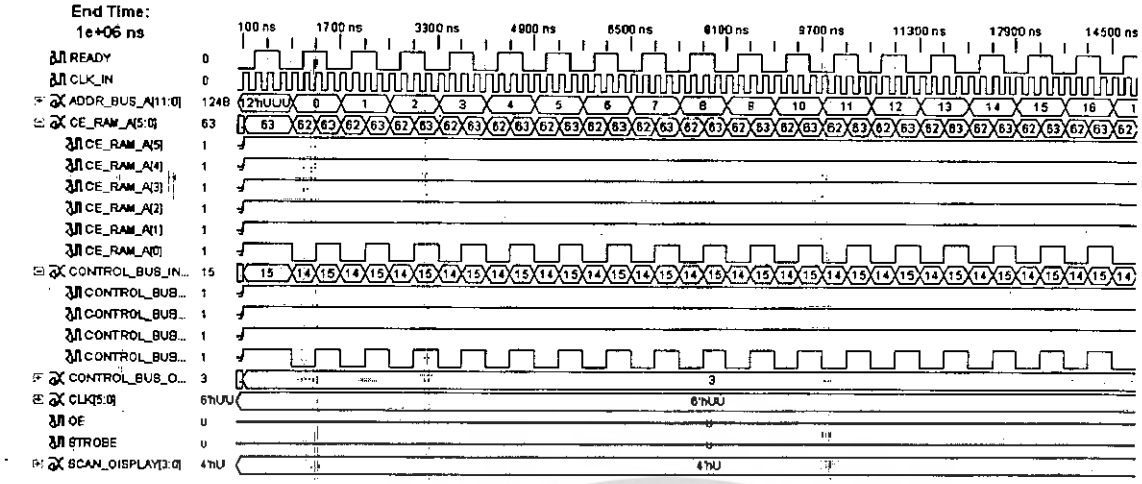
การเขียนข้อมูลภาพ

การเขียนข้อมูลภาพลง SRAM มีสัญญาณที่เกี่ยวข้องดังต่อไปนี้

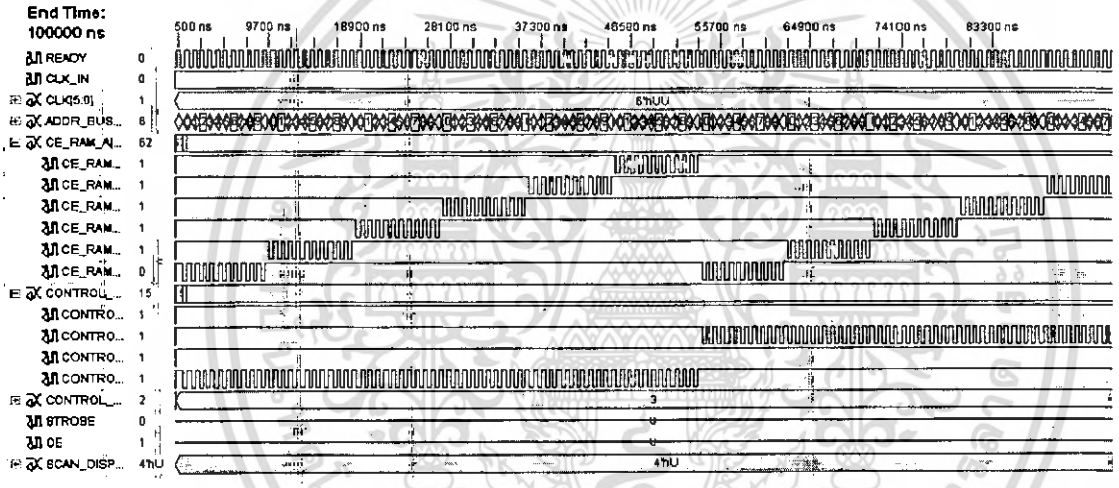
- Address ใช้อ้างอิงตำแหน่งของข้อมูลที่เราต้องการจะเขียน
- Chip Enable (CE) ใช้ Enable แรมตัวที่เราต้องการจะเขียน
- WE ใช้เลือกโหมดการอ่าน – เขียน โดยในที่นี้จะเชื่อมต่อกับสัญญาณ CONTROL_BUS_IN ซึ่งเป็นสัญญาณที่ใช้ควบคุม Buffer ให้ Data Bus ของ FPGA เชื่อมต่อไปยัง Data Bus ของ SRAM

สัญญาณเหล่านี้เราต้องควบคุมให้เป็นลำดับ และ Timing ของสัญญาณต่างๆต้องอยู่ในช่วงที่ Data Sheet กำหนดไว้ โดยผลจากการ Simulation แสดงดัง **รูปที่5.8** และ **รูปที่5.9**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดง Timming ของการเขียนข้อมูลลง SRAM แถวที่ 1

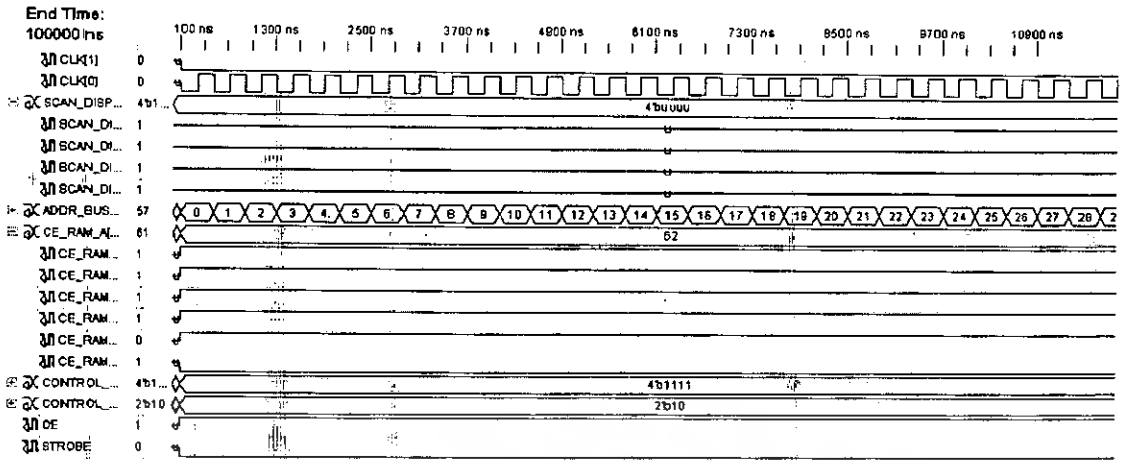


รูปที่ 5.9 แสดง Timming ของการเขียนข้อมูลลง SRAM แถวที่ 1 – 11

การอ่านข้อมูลภาพ และการควบคุมการแสดงผล

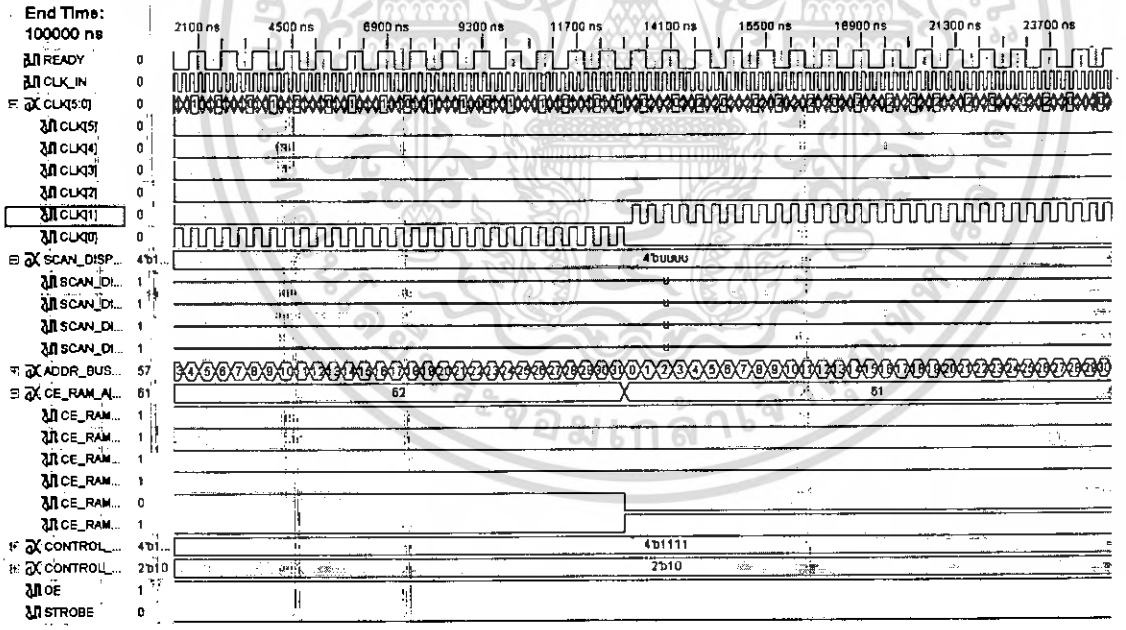
ในการแสดงผลเริ่มต้นด้วยการอ่านข้อมูลภาพจาก SRAM โดยการอ้าง Address เมื่อข้อมูลออกมาที่ Data Bus ของบอร์ดเรียบร้อยแล้วก็จะทำการสร้างสัญญาณ Clock เพื่อ Shift ข้อมูลเข้าสู่บอร์ดแสดงผล โดยจะทำการ Shift ข้อมูล 32 Column จากการที่ออกแบบระบบแบบ บน- ล่าง ทำให้สามารถ Shift ข้อมูลส่วนบนและส่วนล่างพร้อมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดง Timming ของการอ่านข้อมูลSRAM แถวที่ 1 และทำการ Shift ข้อมูลไปที่บอร์ดแสดงผล

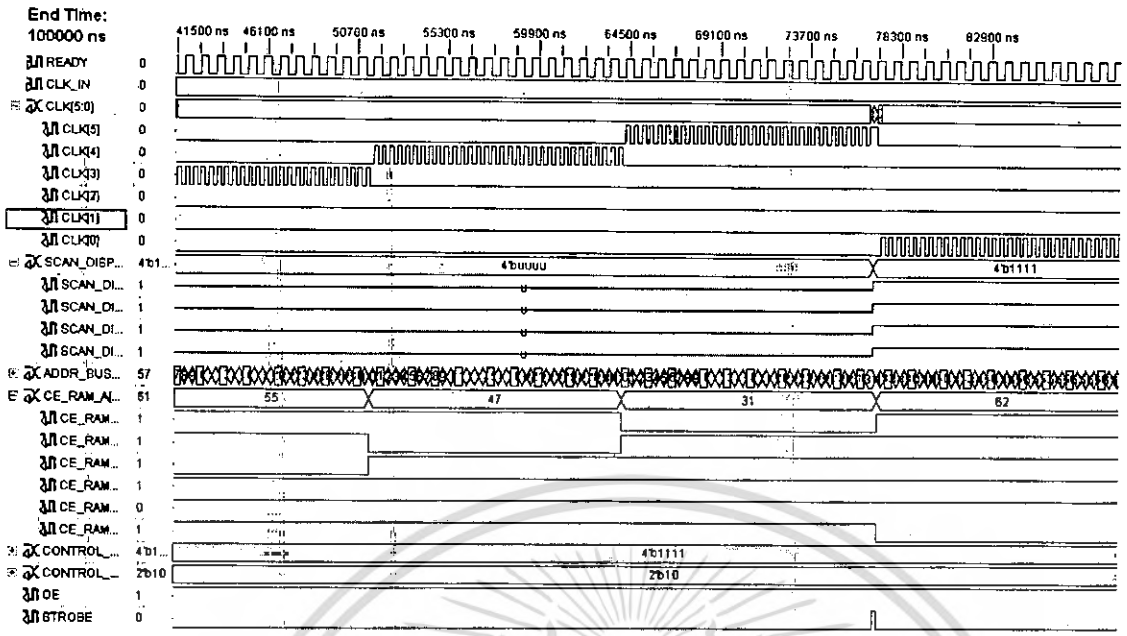
เมื่อ Shift ข้อมูลครบ 32 Column แล้วก็จะเปลี่ยนแถวของบอร์ด ก็จะส่งสัญญาณ CE_RAM ในการเปลี่ยนแถว จะเห็นได้ว่ามีสัญญาณ CE_RAM 6 เส้นก็คือจำนวน แถวของบอร์ดในส่วนบนนั่นเอง



รูปที่ 5.11 แสดง Timming ของการอ่านข้อมูลSRAM แถวที่ 1 กับ 2 และทำการ Shift ข้อมูลไปที่บอร์ดแสดงผล

เมื่อทำการ Shift ข้อมูลจนครบทุกบอร์ดแล้ว ก็จะทำการสร้างสัญญาณควบคุมการ Scan ก็คือจะเปิด STROBE ให้ข้อมูลส่งผ่าน Buffer ออกไปได้ พร้อมกันนั้นก็จะเปิดเส้น Scan เพื่อให้ข้อมูลแสดงผลออกมาที่ LED Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 แสดง Timming ของการ Scan หลังจากการ Shift ข้อมูลครบทุกบอร์ด

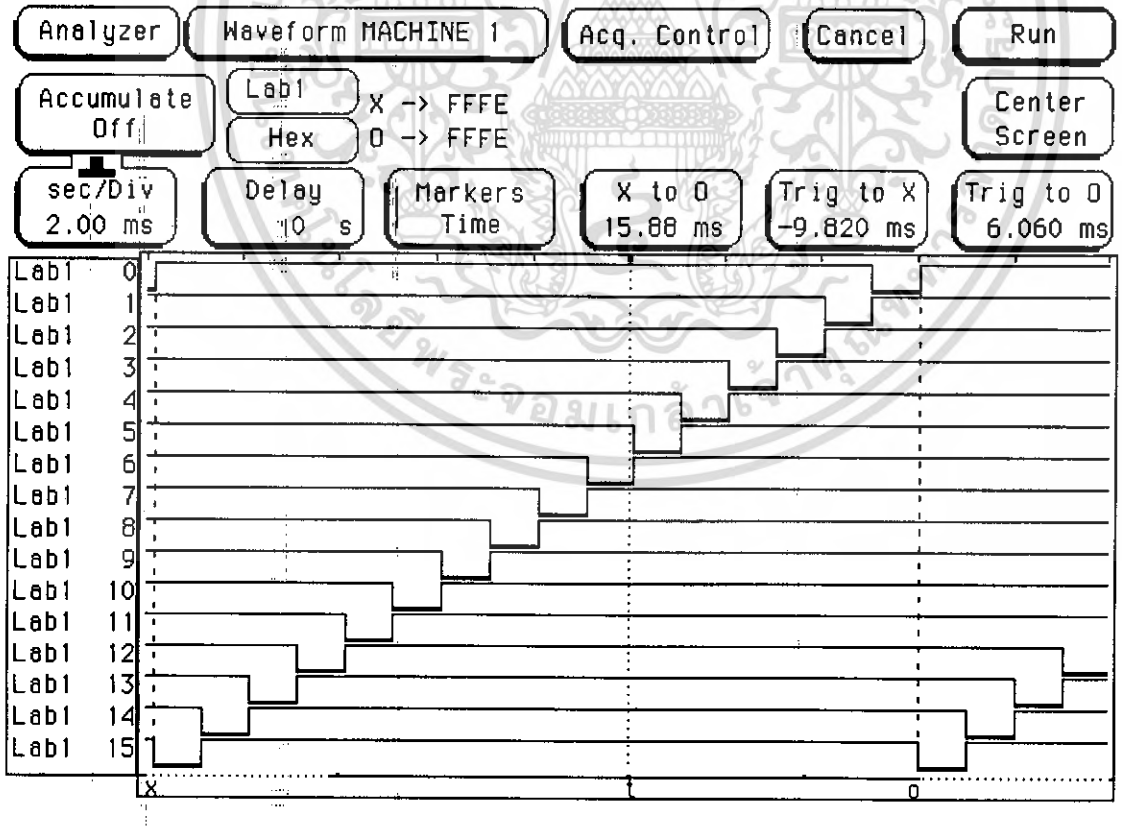
5.2.2 ผลการทดลองกับ Hardware

การทดลองทาง Hardware เป็นการทดสอบระบบทั้งหมดรวมทั้งโปรแกรม Visual Basic 6 ด้วย โดยการทดลองใส่ภาพที่ต้องการแล้วส่งข้อมูลไปที่ FPGA เพื่อทำการแสดงผล

ตัวอย่างผลการทดลองเป็นดัง **รูปที่ 5.13** จากรูปจะเห็นว่าการทำงานระดับ Function ถูกต้อง เมื่อวัดความถี่ในการ Scan จะได้ Timing ดัง **รูปที่ 5.14** และ **รูปที่ 5.15** จากรูปจะเห็นได้ว่าใช้เวลาในการ Scan 1 รอบ เท่ากับ 15.88 mS ก็คือความถี่ 62.97 Hz

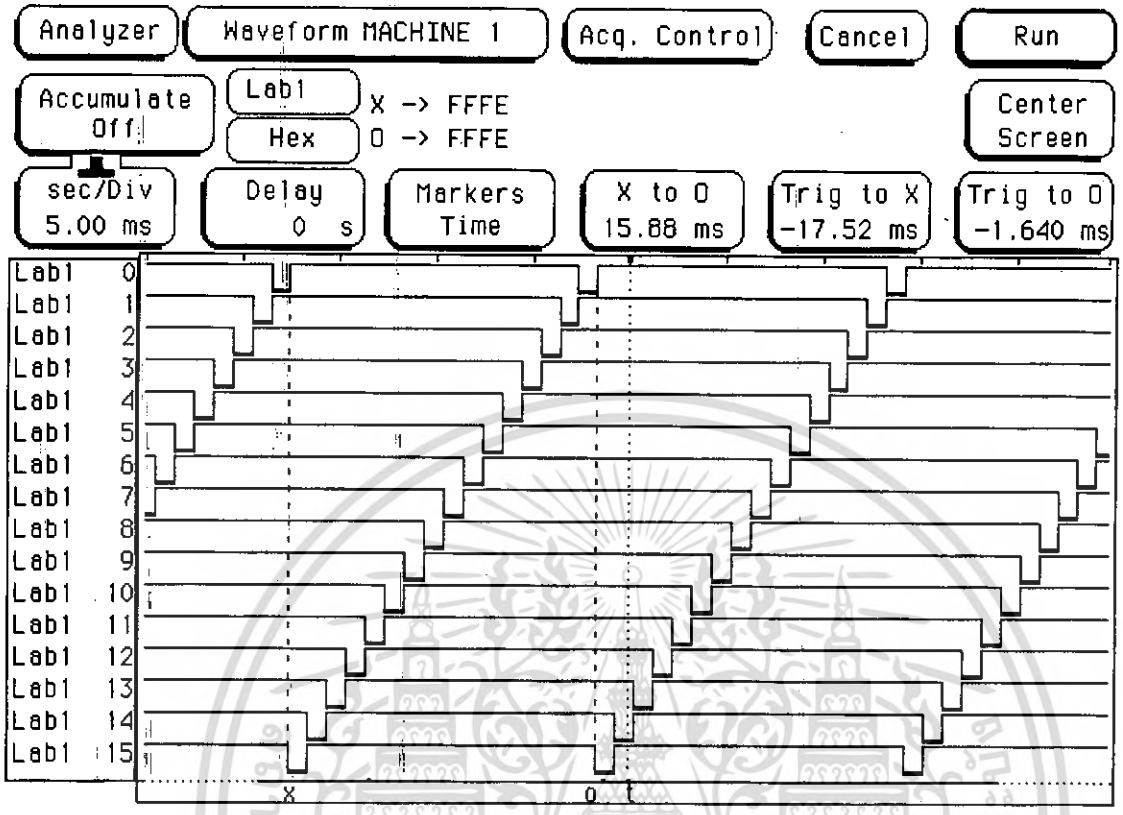
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.13 ตัวอย่างการแสดงผลการทดลองกับ Hardware จริง



รูปที่ 5.14 แสดง Timing ของการ Scan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.15 แสดง Timing ของการ Scan

เมื่อวัดกำลังงานที่ใช้ไปในแต่ละ BOARD LED จะได้ข้อมูลดังตาราง

REAL CURRENT AND POWER DISSIPATION					
	NO.	W	TOTAL(W)	AMPARE	Volts
control board	1	1.5	1.5	0.3	5
box LED	1	25	6	1.2	5
VGS : box LED	1	0.48	0.48	0.04	12
		TOTAL	7.98	1.596	5

เมื่อวัดกำลังงานทั้งหมดของ LED DISPLAY BOARD จะได้ข้อมูลดังตาราง

REAL CURRENT AND POWER DISSIPATION ALL BOARD					
	NO.	W	TOTAL(W)	AMPARE	Volts
control board	1	1.5	1.5	0.3	5
All box LED	88	25	528	1.2	5
All VGS : box LED	88	0.48	42.24	0.04	12
		TOTAL	580.2	116.04	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์ผลการทดลอง

สรุปผลการทดลอง

จากการทดลองนี้ทำให้ทางคณะผู้จัดทำมีความรู้ความเข้าใจเกี่ยวกับส่วนต่างๆของบอร์ดแสดงผล LED การใช้มอสเฟทในการขับกระแสและทำให้เข้าใจถึงหลักการสแกนภาพเพื่อลดกำลังงานที่จ่ายให้แก่โหลดและช่วยเพิ่มความสว่างให้แก่ LED การออกแบบวงจรดิจิตอลด้วยภาษา VHDL การแปลงสัญญาณ VGA ซึ่งเป็นสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอล (A/D) การนำสัญญาณดิจิตอลที่ได้ไปแปลงเป็นระดับความสว่าง 8 ระดับบนจอแสดงผล LED การใช้บอร์ดทดลอง FPGA (Field Programmable Gate Arrays) คอนโทรลการทำงานในทุกส่วนที่เป็นสัญญาณดิจิตอลทั้งหมดในโครงงานชิ้นนี้

วิจารณ์ผลการทดลอง

1. ในส่วนของการแปลงสัญญาณ VGA (สัญญาณอนาล็อก) ไปเป็นสัญญาณดิจิตอล

เนื่องจากสัญญาณวีดีโอที่ใช้ในโครงงานนี้เราได้นำสัญญาณ VGA จากคอมพิวเตอร์ส่วนบุคคลซึ่งสัญญาณวีดีโอที่ได้นั้น ได้มีการแยกสัญญาณสีออกเป็น 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน นอกจากนี้ยังแยกสัญญาณ Hor_Sync และสัญญาณ Ver_Sync แล้วดังนั้นสัญญาณที่จะนำมาแปลงเป็นสัญญาณดิจิตอลจึงมีสามสี โดยในการทดลองครั้งนี้ได้ทำการทดลองเฉพาะสัญญาณสีเขียว

เนื่องจากระบบ Automatic Gain Control ภายใน TDA8708A จะใช้งานได้นั้น เราต้องควบคุมสัญญาณในการควบคุม โดยการนำเอาสัญญาณ Gate A และ Gate B ที่สร้างมาจาก FPGA มาควบคุม ระบบ Automatic Gain Control และใช้การปรับแรงดันที่ขา 24 CLAMP Input ที่ 3.5 โวลต์ และปรับแรงดันที่ขา 25 ACG Input ที่ 2.8-4.0 โวลต์ ทั้งนี้เพื่อทำให้สัญญาณวีดีโอที่ได้จากการแปลงเป็นสัญญาณดิจิตอลมีค่าที่ถูกต้อง

2. ส่วนของการแสดงภาพนิ่งผ่านทางโปรแกรมที่เขียนขึ้นด้วยภาษา Visual Basic 6

ความถี่ในการ Scan เท่ากับ 62.97 Hz เมื่อดูจากภาพที่ปรากฏจะพบว่ายังเห็นการ Scan อยู่ ต้องเพิ่มความถี่ในการ Scan มากกว่านี้ แต่จะทำให้เกิดปัญหา การ Shift ข้อมูลขึ้น แนวทางการแก้ปัญหานี้ทำได้โดยแยกความถี่ในการ Scan และการ Shift ข้อมูลออกจากกัน เนื่องจากการออกแบบนี้ไม่ได้แยกออกจากกันทำให้เกิดปัญหานี้ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปสรรคที่พบในการทำโครงงาน

1. ผลการทดลองที่ได้บางส่วนยังไม่เป็นตามทฤษฎี หรือที่ออกแบบไว้โดยตรง ทั้งนี้อาจเกิดจากตัวแปรหลายอย่าง เช่น สัญญาณรบกวน การเชื่อมต่ออุปกรณ์ เครื่องมือ หรือความผิดพลาดของผู้ทดลอง
2. เนื่องจากบอร์ดทดลอง FPGA ของ Xilinx ที่ใช้ในการทำโครงงานชิ้นนี้มีปัญหาในส่วนของการโปรแกรมลง PROM โดยเมื่อโปรแกรมแล้วในบางครั้งการทำงานของวงจรที่ออกแบบอาจไม่สมบูรณ์ร้อยเปอร์เซ็นต์ หากเป็นเช่นนี้แนวทางแก้ไขจึงควรที่จะลบข้อมูลที่มีอยู่ใน PROM เดิมออกให้หมดเสียก่อนแล้วจึงโปรแกรมลงไปใหม่ การทำงานของวงจรก็จะให้ผลออกมาสมบูรณ์ตามที่ได้ออกแบบไว้
3. จากปัญหาเรื่องความสว่างของ LED ในแต่ละบอร์ดที่ไม่เท่ากันทำให้ภาพที่ได้ไม่ดีจากเทอมที่แล้ว สำหรับเทอมนี้ได้มีการแก้ปัญหานั้นโดยการแยกการปรับแรงดันของส่วนควบคุมความสว่างของแต่ละบอร์ดออกจากกันทำให้แก้ปัญหาได้ส่วนหนึ่ง แต่ว่าเราไม่สามารถแยกปรับความสว่างของแต่ละสีได้ เนื่องจากข้อจำกัดของวงจรขับกระแส ทำให้ความสว่างยังไม่ค่อยเท่ากันอยู่ดี

ประโยชน์ที่ได้รับจากโครงงานนี้

1. ใช้ภาษา VHDL ในการออกแบบส่วนประมวลผลต่างๆ และพัฒนาให้มีประสิทธิภาพดีขึ้น
2. สามารถที่จะออกแบบ และประยุกต์ใช้งานอุปกรณ์ต่างๆในวงจรได้ดี
3. สามารถเข้าใจการออกแบบเป็นระบบขั้นตอน
4. ได้ความรู้ที่จะนำไปพัฒนาและประยุกต์ใช้งานในระดับที่สูงขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ชำนาญ ปัญญาใส และ วัชรากร หนูทอง,ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล, "บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)", ปี 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้