ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

Digital Training System Using VHDL with FPGA and CPLD

ชื่อนักศึกษา

1. นายคณาพจน์	ลิ่มทอง	รหัสประจำตัว	44035316
2. นายกิตติพงษ์	บุญธนฤทธิ์	รหัสประจำตัว	44035357
3. นายธันว์	บัวเนียม	รหัสประจำตัว	44035370
4. นายไมตรี	นาคประสิทธิ์	รหัสประจำตัว	44035379

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์วรวิทย์ สมหา

อาจารย์ที่ปรึกษาร่วม อาจารย์กิติพงค์ มะโน

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์วรวิทย์	สมหา	.....
2. อาจารย์กิติพงค์	มะโน	.....
3. อาจารย์พีระวุฒิ	สุวรรณจันทร์	.....
4. อาจารย์ไพบูลย์	พวงวงศ์ตระกูล	.....
5. อาจารย์โกศล	ตราชู	.....

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 พฤศจิกายน พ.ศ. 2545 เวลา 14.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 30 เดือน เมษายน พ.ศ. 2546

<BT4502102>

ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

DIGITAL TRAINING SYSTEM USING VHDL WITH FPGA AND CPLD

T 048327

นายคณาพจน์ ลิ่มทอง
นายคิตติพงษ์ บุญธนฤทธิ์
นายธันว์ บัวเนียม
นายไมตรี นาคประสิทธิ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 48327
วัน, เดือน, ปี 15 ต.ค. 2546

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

Digital Training System Using VHDL With FPGA And CPLD

วัตถุประสงค์

1) เพื่อศึกษาหลักการทำงานโปรแกรม Xilinx Foundation F2.1I , อุปกรณ์โปรแกรมได้และการประยุกต์ใช้ของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

2) เพื่อออกแบบระบบดิจิตอลในชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

3) เพื่อสร้างชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

4) เพื่อทดลองใช้และหาคุณภาพของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

5) เพื่อนำชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี ไปใช้ออกแบบและทดสอบระบบดิจิตอล

ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ทราบหลักการทำงานของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

2) ได้วงจรในชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

3) ได้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี ที่สมบูรณ์

4) ได้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี ที่ผ่านการทดสอบและหาคุณภาพได้อย่างสมบูรณ์

5) ได้นำชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดีเป็นสื่อการเรียนรู้สำหรับผู้ที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี
นักศึกษา	นายคณาพจน์ ลิ่มทอง
	นายกิตติพงษ์ บุญธนฤทธิ์
	นายธันว์ บัวเนียม
	นายไมตรี นาคประสิทธิ์
อาจารย์ที่ปรึกษา	อาจารย์วรวิทย์ สมหา
อาจารย์ที่ปรึกษาร่วม	อาจารย์กิติพงค์ มะโน
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการออกแบบและการสร้างชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี ชุดปฏิบัติการจะแบ่งแผงวงจรออกเป็น 3 ส่วน คือส่วนที่ 1 แผงวงจรเอฟพีจีเอ ส่วนที่ 2 แผงวงจรซีพีแอลดี ส่วนที่ 3 แผงวงจรอินพุตเอาต์พุต ชุดปฏิบัติการนี้ช่วยในการศึกษาโปรแกรมภาษาวีเอชดีแอล รวมทั้งหลักการทำงานของอุปกรณ์เอฟพีจีเอและซีพีแอลดีและสามารถนำไปประยุกต์ใช้งานกับวงจรต่าง ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Digital Training System Using VHDL with FPGA and CPLD
Students	Mr. Kanapot Limtong
	Mr. Kittipong Bunthanarit
	Mr. Than Buaniam
	Mr. Mitree Nakparsit
Advisor	Mr. Worawit Somha
Co-Advisor	Mr. Kitipong Mano
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	2002

ABSTRACT

This thesis presents design and implemetation of Digital Training System Using VHDL with FPGA and CPLD Again, three board and previded. First bord is FPGA board. Second board is CPLD board. The last board is input/output. It is help to study VHDL Language, principled FPGA and CPLD. Con be applied to any circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีจากความร่วมมือของ สมาชิกภายในกลุ่มทุกท่าน นอกจากนี้ยังได้รับความกรุณาจาก ท่านอาจารย์กิติพงค์ มะโน ซึ่งเป็นอาจารย์ที่ปรึกษาประจำโครงงาน และอาจารย์อำพล ทองระอา ซึ่งเป็นอาจารย์ที่ปรึกษาร่วม ซึ่งได้ให้ความความช่วยเหลือในด้านต่าง ๆ ตลอดจนให้โอกาสในการทำโครงงาน อย่างเต็มที่ทั้งเวลา สถานที่และอุปกรณ์ต่าง ๆ และอาจารย์สุรพงษ์ สิริพงษ์ดี ซึ่งได้ให้คำปรึกษาแนะนำในด้าน ฮาร์ดแวร์ อีกทั้งคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ขอขอบคุณเพื่อน ๆ ที่ได้ให้คำปรึกษาในด้านการใช้ซอฟต์แวร์ต่างๆ ด้วยดีเสมอมา สุดท้ายที่ควรระลึกถึงอย่างยิ่ง บิดา และมารดา ที่เป็นผู้ให้กำเนิด ให้โอกาสในการศึกษาตลอดจนสนับสนุนและให้กำลังใจในการทำโครงงานนี้ ด้วยดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.4 เนื้อหาโดยสังเขป	3
บทที่ 2 ทฤษฎี และหลักการ	4
2.1 กล่าวนำ	4
2.2 ประวัติความเป็นมาของภาษาวีเอชดีแอล	4
2.3 ส่วนประกอบต่าง ๆ ของภาษาวีเอชดีแอล	6
2.4 ภาษาวีเอชดีแอลเพื่อการสังเคราะห์	18
2.5 การออกแบบจากบนลงล่าง	21
2.6 เอฟพีจีเอ (FPGA : Field Programmable Gate Array)	23
2.7 การโปรแกรมเอฟพีจีเอตระกูล XC 4000	29
2.8 การใช้ความสามารถของแรมในเอฟพีจีเอตระกูล XC 4000	33
2.9 ซีพีแอลดี (CPLD : Complex Programmable Logic Device)	35
2.10 อุปกรณ์การดาวน์โหลด Xchecker Cable	38
บทที่ 3 การออกแบบการสร้าง และการทำงาน	43
3.1 โมดูลต่าง ๆ ของชุดทดลอง	43
3.2 วงจรและการทำงานโดยสังเขป	43
3.2.1 โมดูลแสดงผลเจ็ดส่วน	44
3.2.2 โมดูลลำโพง	45
3.2.3 โมดูลดิฟสวิตช์ 8บิต	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.2.4 โมดูลแอลอีดี 40 ดวง	46
3.2.5 โมดูลสี่แยกไฟแดง	47
3.2.6 โมดูลคีย์บอร์ดเมตริกซ์4 x 4	47
3.2.7 โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก	48
3.2.8 โมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล	49
3.2.9 โมดูลมอเตอร์ไฟฟ้ากระแสตรง	50
3.2.10 โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว	51
3.2.11 โมดูลวงจรกำเนิดสัญญาณนาฬิกา	52
3.2.12 โมดูลอีพรอม 256 กิโลไบต์	53
3.2.13 โมดูล MCS 8051	56
3.2.14 โมดูลสเตปปิ้งมอเตอร์	60
3.2.15 โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์, 9 โวลต์ และ12 โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย	61
3.2.16 โมดูล แรม 128 กิโลไบต์	62
3.2.17 แผงวงจรเอฟพีจีเอ	64
3.2.18 แผงวงจรซีพีแอลดี	70
บทที่ 4 การทดลอง และผลการทดลอง	74
4.1 แผงวงจรซีพีแอลดี	74
4.2 แผงวงจรเอฟพีจีเอ	75
4.3 แผงวงจรอินพุต/เอาต์พุต	75
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	90
5.1 บทสรุป	90
5.2 ปัญหา และแนวทางการแก้ไข	90
5.3 แนวทางการพัฒนา	90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ก เครื่องต้นแบบ	91
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	94
ภาคผนวก ค ใบงานการทดลอง	108
ภาคผนวก ง เฉลยใบงานการทดลอง	176
ภาคผนวก จ คู่มือการใช้งาน	220
บรรณานุกรม	249
ประวัติผู้แต่ง	250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การทำโอเปอรเรชั่นทางคณิตศาสตร์	9
ตารางที่ 2.2 ลักษณะการทำงานของเกตพื้นฐานต่าง ๆ	19
ตารางที่ 2.2 (ต่อ) ลักษณะการทำงานของเกตพื้นฐานต่าง	20
ตารางที่ 2.3 ลักษณะการทำงานของฟลิปฟลอปพื้นฐานต่างๆ ด้วยการเขียนโปรแกรมวีเอชดีแอล	20
ตารางที่ 2.3 (ต่อ) ลักษณะการทำงานของฟลิปฟลอปพื้นฐานต่างๆ ด้วยการเขียนโปรแกรมวีเอชดีแอล	21
ตารางที่ 2.4 รายละเอียดของอุปกรณ์ภายในเอฟพีจีเอตระกูล XC4000	29
ตารางที่ 2.5 รูปแบบของโหมดต่าง ๆ ในการโปรแกรมเอฟพีจีเอตระกูล XC4000	30
ตารางที่ 2.6 จำนวนของแรมภายในเอฟพีจีเอตระกูล XC4000	33
ตารางที่ 2.7 รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010	34
ตารางที่ 2.7 (ต่อ)รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010	35
ตารางที่ 2.8 คุณสมบัติของไอซีตระกูล XC9500XL/XV	38
ตารางที่ 3.1 ตำแหน่งขา IC LM741	49
ตารางที่ 3.2 ตำแหน่งขาของ IC ADC0804	50
ตารางที่ 3.3 ตำแหน่งขาของ IC 7402	52
ตารางที่ 3.3 (ต่อ) ตำแหน่งขาของ IC 7402	53
ตารางที่ 3.4 ตำแหน่งขาของ IC M27C2001	54
ตารางที่ 3.4 (ต่อ) ตำแหน่งของ IC M27C2001	55
ตารางที่ 3.5 โหมดการทำงานของ IC M27C2001	55
ตารางที่ 3.6 ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L	57
ตารางที่ 3.6 (ต่อ) ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L	58
ตารางที่ 3.7 การทำงานของ ULN2003	60
ตารางที่ 3.8 ตารางขาของ IC K6T1008C2E	63
ตารางที่ 3.9 หน้าที่การทำงานของ XC4000E	67
ตารางที่ 3.9 (ต่อ) หน้าที่การทำงานของ XC4000E แต่ละขา	68
ตารางที่ 3.10 ชุดคาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตาราง	หน้า
ตารางที่ 3.10 (ต่อ) ชุดดาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด	69
ตารางที่ 3.11 หน้าที่การทำงานของ XC9500 แต่ละขา	71
ตารางที่ 3.11 (ต่อ) หน้าที่การทำงานของ XC9500 แต่ละขา	72
ตารางที่ 3.12 ชุดดาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด	73
ตารางที่ 4.1 ผลการทดลองโมดูลแสดงผลเจ็ดส่วน	76
ตารางที่ 4.2 ผลการทดลองโมดูลดิพสวิตช์ 8 บิต	77
ตารางที่ 4.3 ผลการทดลองโมดูลคีย์บอร์ดเมตริกซ์ 4x4	78
ตารางที่ 4.4 ผลการทดลองโมดูลสเตปปิ้งมอเตอร์	79
ตารางที่ 4.5 ผลการทดลองโมดูลแอลอีดี 40 ดวง	80
ตารางที่ 4.6 ผลการทดลองโมดูลแปลงสัญญาณดิจิตอลเป็น แอนะลอก	81
ตารางที่ 4.7 ผลการทดลองโมดูลแปลงสัญญาณแอนะลอกเป็น ดิจิตอล	82
ตารางที่ 4.8 ผลการทดลองโมดูลมอเตอร์ไฟฟ้ากระแสตรง	83
ตารางที่ 4.9 ผลการทดลองโมดูลลำโพง	83
ตารางที่ 4.10 ผลการทดลองการเขียนข้อมูลของโมดูลแรม 128 กิโลไบต์	84
ตารางที่ 4.11 ผลการทดลองการอ่านข้อมูลของโมดูลแรม 128 กิโลไบต์	84
ตารางที่ 4.12 ผลการทดลองโมดูลอีพรอม 256 กิโลไบต์	85
ตารางที่ 4.13 ผลการทดลองของ โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5โวลต์ ,9 โวลต์ ,12 โวลต์	86
ตารางที่ 4.14 ผลการทดลองโมดูลสวิตช์กดติดปล่อยดับ 16 ตัว	87
ตารางที่ 4.15 ผลการทดลองโมดูลวงจรสี่แยกไฟแดง	88
ตารางที่ ค.1 บันทึกผลการทดลองที่ 1	129
ตารางที่ ค.2 บันทึกผลการทดลองที่ 2	131
ตารางที่ ค.3 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	142
ตารางที่ ค.4 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	143
ตารางที่ ค.5 ตารางความจริงของวงจรเข้ารหัส	144
ตารางที่ ค.6 เอแอลยู 4 บิต 15 คำสั่ง	145
ตารางที่ ค.6 (ต่อ) เอแอลยู 4 บิต 15 คำสั่ง	146

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตาราง	หน้า
ตารางที่ ค.7 การเกิดสถานะของเอาต์พุตปัจจุบัน	152
ตารางที่ ค.8 ตารางความจริงของนอร์แลทซ์	152
ตารางที่ ค.9 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	158
ตารางที่ ค.10 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	159
ตารางที่ ค.11 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	160
ตารางที่ ค.12 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	160
ตารางที่ ค.13 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	163
ตารางที่ ค.14 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	164
ตารางที่ ค.15 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	165
ตารางที่ ค.16 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	166
ตารางที่ ค.17 กำหนดขาเพื่อใช้งานเอฟพีจีเอ	167
ตารางที่ ค.18 ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L	171
ตารางที่ ง.1 ผลการทดลองที่ 1 ลักษณะการหมุนของมอเตอร์ไฟฟ้ากระแสตรง	179
ตารางที่ ง.2 ตารางบันทึกผลการทดลอง	179
ตารางที่ ง.2 (ต่อ) ตารางบันทึกผลการทดลอง	180
ตารางที่ ง.3 ตารางความจริง	186
ตารางที่ ง.4 ผลการทดลองที่ 1	189
ตารางที่ ง.5 ผลการทดลองที่ 1	192
ตารางที่ ง.6 ผลการทดลองที่ 3	194
ตารางที่ ง.7 ผลการทดลองที่ 4	195
ตารางที่ ง.8 เฉลยตารางความจริง	196

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี	6
รูปที่ 2.2 ตัวอย่างของการประกาศค่า Constant	9
รูปที่ 2.3 ตัวอย่างของการประกาศ Variable	10
รูปที่ 2.4 ตัวอย่างของการประกาศ Signal	10
รูปที่ 2.5 Character มีค่าที่เป็นไปได้	11
รูปที่ 2.6 รูปแบบของมัลติเพล็กซ์ (ก) หน่วยการออกแบบเอนทิตี (ข) มุมมองของตัวเชื่อมประสาน (Interfacing)	12
รูปที่ 2.7 รูปแบบของมัลติเพล็กซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย (ก) หน่วยการออกแบบเอนทิตีในรูปของ (ข) มุมมองของตัวเชื่อมประสาน (Interfacing)	12
รูปที่ 2.8 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก	13
รูปที่ 2.9 โครงสร้างโดยทั่วของหน่วยการออกแบบสถาปัตยกรรม	13
รูปที่ 2.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ตามฟังก์ชั่นบูลีน	14
รูปที่ 2.11 โครงสร้างภายในสถาปัตยกรรมของมัลติเพล็กซ์	15
รูปที่ 2.12 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทโครงสร้าง	15
รูปที่ 2.13 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทพฤติกรรม	16
รูปที่ 2.14 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็คเก็ต	17
รูปที่ 2.15 โครงสร้างของบอดี้แพ็กเก็ต	18
รูปที่ 2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	18
รูปที่ 2.17 ขั้นตอนการออกแบบจากบนลงล่าง	22
รูปที่ 2.18 โครงสร้างภายในของเอฟพีจีเอของบริษัทไซลิงซ์ (XILINX Inc.)	24
รูปที่ 2.19 ผังวงจรภายในของไอซีแอลบีของเอฟพีจีเอ ตระกูล XC4000	25
รูปที่ 2.20 ผังวงจรของไอโอบีภายในเอฟพีจีเอตระกูล XC4000	26
รูปที่ 2.21 เส้นทางการเชื่อมต่อระหว่างไอโอบีกับซีแอลบีเอของเอฟพีจีเอตระกูล XC4000	27
รูปที่ 2.22 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดหลักแบบขนาน	31
รูปที่ 2.23 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดเพอริเฟอรัลแบบอะซิงโครนัส	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.24 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดรองแบบอนุกรม	33
รูปที่ 2.25 โครงสร้างของ XC9500 CPLDs	36
รูปที่ 2.26 การพิจารณความหมายต่างๆ ที่ด้านบนของตัวถังไอซีในตระกูล XC9500 CPLDs	37
รูปที่ 2.27 สาย X-Checker พร้อมแถบสายต่อไปแผงวงจรทดลอง	39
รูปที่ 2.28 ส่วนหัวของสาย X-Checker	40
รูปที่ 2.29 การเชื่อมระหว่าง X-Checker กับแผงวงจรทดลอง	41
รูปที่ 2.30 หน้าต่างโปรแกรม Hardware Debugger	41
รูปที่ 2.31 หน้าต่างโปรแกรม Hardware Debugger ในการเปิด Design project	42
รูปที่ 2.32 หน้าต่าง Communication Setup	42
รูปที่ 3.1 วงจรแสดงผลเจ็ดส่วน	44
รูปที่ 3.2 วงจงลำโพง	45
รูปที่ 3.3 วงจรดิฟสวิตซ์ 8บิต2 ตัว	45
รูปที่ 3.4 วงจงวงจรแอลอีดี 40 ดวง	46
รูปที่ 3.5 วงจร 4 แยกไฟแดง	47
รูปที่ 3.6 วงจรคีย์บอร์ดเมตริกซ์4 x 4	48
รูปที่ 3.7 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก	48
รูปที่ 3.8 วงจรวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล	49
รูปที่ 3.9 วงจรมอเตอร์ไฟฟ้ากระแสตรง	51
รูปที่ 3.10 วงจรสวิตซ์กดติดปล่อยดับ 16 ตัว	51
รูปที่ 3.11 วงจรกำเนิดสัญญาณนาฬิกา	52
รูปที่ 3.12 วงจรอีพรอม 256 กิโลไบต์	53
รูปที่ 3.13 วงจร MCS 8051	56
รูปที่ 3.14 วงจรสเตปปิ้งมอเตอร์	60
รูปที่ 3.15 วงจรแหล่งจ่ายไฟฟ้ากระแสตรง 5โวลต์, 9โวลต์ และ 12โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย	61
รูปที่ 3.16 วงจร แรม 128 กิโลไบต์	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.17 วงจรแผงวงจรเอฟพีจีเอ	64
รูปที่ 3.17 (ต่อ) วงจรแผงวงจรเอฟพีจีเอ	65
รูปที่ 3.17 (ต่อ) วงจรแผงวงจรเอฟพีจีเอ	66
รูปที่ 3.18 วงจรแผงวงจร ซีพีแอลดี	70
รูปที่ ก.1 แผงวงจรซีพีแอลดี	92
รูปที่ ก.2 แผงวงจรเอฟพีจีเอ	92
รูปที่ ก.3 แผงวงจรอินพุต/เอาต์พุต	93
รูปที่ ข.1 วงจรรักษาระดับแรงดันไฟฟ้า 5 โวลต์ ของแผงวงจรเอฟพีจีเอ	95
รูปที่ ข.2 วงจรของแผงวงจรเอฟพีจีเอส่วนที่ 1	95
รูปที่ ข.3 วงจรของแผงวงจรเอฟพีจีเอส่วนที่ 2	96
รูปที่ ข.4 แผ่นวงจรพิมพ์ ด้านล่างของแผงวงจรเอฟพีจีเอ	97
รูปที่ ข.5 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรเอฟพีจีเอ	98
รูปที่ ข.6 แผ่นวงจรพิมพ์ของจุดการลงอุปกรณ์ของแผงวงจรเอฟพีจีเอ	99
รูปที่ ข.7 ตำแหน่งการวางอุปกรณ์ของแผงวงจรเอฟพีจีเอ	99
รูปที่ ข.8 วงจรของแผงวงจรซีพีแอลดี	100
รูปที่ ข.9 แผ่นวงจรพิมพ์ด้านล่างของแผงวงจรซีพีแอลดี	101
รูปที่ ข.10 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรซีพีแอลดี	102
รูปที่ ข.11 แผ่นวงจรพิมพ์ของจุดการลงอุปกรณ์ของแผงวงจรซีพีแอลดี	103
รูปที่ ข.12 ของตำแหน่งการวางอุปกรณ์ของแผงวงจรซีพีแอลดี	104
รูปที่ ข.13 แผ่นวงจรพิมพ์ด้านล่างของแผงวงจรอินพุต/เอาต์พุต	105
รูปที่ ข.14 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรอินพุต/เอาต์พุต	106
รูปที่ ข.15 ตำแหน่งอุปกรณ์ของแผงวงจรอินพุต/เอาต์พุต	107
รูปที่ ค.1 Icon Project Manager	110
รูปที่ ค.2 หน้าต่าง Getting Started	110
รูปที่ ค.3 หน้าต่าง New Project	111
รูปที่ ค.4 หน้าต่าง Project Manager	111
รูปที่ ค.5 HDL Editor	112
รูปที่ ค.6 หน้าต่าง HDL Editor	112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ค.7 หน้าต่าง Design Wizard	113
รูปที่ ค.8 หน้าต่าง Design Wizard – Language	113
รูปที่ ค.9 หน้าต่าง Design Wizard - Name	114
รูปที่ ค.10 หน้าต่าง Design Wizard - Ports	114
รูปที่ ค.11 หน้าต่าง Design Wizard - Ports	115
รูปที่ ค.12 หน้าต่าง HDL Editor	115
รูปที่ ค.13 หน้าต่าง HDL Editor	116
รูปที่ ค.14 Menu Add to Project	116
รูปที่ ค.15 Menu Check Syntax	117
รูปที่ ค.16 หน้าต่าง Check Successful	117
รูปที่ ค.17 หน้าต่าง Project Manager	118
รูปที่ ค.18 Synthesis	118
รูปที่ ค.19 หน้าต่าง Synthesis/Implementation settings	119
รูปที่ ค.20 หน้าต่าง Constraints	119
รูปที่ ค.21 Implementation	120
รูปที่ ค.22 หน้าต่าง Synthesis/Implementation settings	120
รูปที่ ค.23 Flow Engine	121
รูปที่ ค.24 Device Programming	121
รูปที่ ค.25 หน้าต่าง Select Program	122
รูปที่ ค.26 หน้าต่าง Hardware Debugger	122
รูปที่ ค.27 หน้าต่าง Logic Simulator	123
รูปที่ ค.28 หน้าต่าง Component Selection for Wavefrom Viewer	124
รูปที่ ค.29 หน้าต่าง Waveform Viewer	124
รูปที่ ค.30 หน้าต่าง Logic Simulator	125
รูปที่ ค.31 โปรแกรมภาษาวีเอชดีแอลบรรยาย แอนเกต	126
รูปที่ ค.32 วงจรการทดลองที่ 1 การใช้แผงวงจรซีพีแอลดี	128
รูปที่ ค.33 โปรแกรมภาษา วีเอชดีแอล บรรยาย มอเตอร์ไฟฟ้ากระแสตรง	129
รูปที่ ค.34 โปรแกรมภาษาวีเอชดีแอลบรรยาย Decoder to 7 – Segment	130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ค.35 Behavioral Style	133
รูปที่ ค.36 แบบ Data Flow Style	133
รูปที่ ค.37 แบบ Structure Style	134
รูปที่ ค.38 โปรแกรมภาษาวีเอชดีแอลบรรยายออร์เกตแบบ Data Flow Style	135
รูปที่ ค.39 โปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style	136
รูปที่ ค.40 วงจรการทดลองที่ 6	138
รูปที่ ค.41 วงจรคอมบิเนชั่น	140
รูปที่ ค.42 วงจรบวกครึ่งตัวบวกครึ่ง	141
รูปที่ ค.43 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกครึ่ง	141
รูปที่ ค.44 แผนผังการต่อแผงวงจรเอฟพีจีเอ	142
รูปที่ ค.45 ตัวบวกเต็ม (Full Adder) สร้างจากวงจรตัวบวกครึ่ง (Half Adder)	142
รูปที่ ค.46 ตัวบวกเต็ม (Full Adder)	143
รูปที่ ค.47 แผนผังการต่อแผงวงจรเอฟพีจีเอ	145
รูปที่ ค.48 การกำหนดตัวแปร	146
รูปที่ ค.49 แผนผังการต่อแผงวงจรเอฟพีจีเอ	147
รูปที่ ค.50 แผนผังการออกแบบเอแอลยู	147
รูปที่ ค.51 โครงสร้างของอุปกรณ์เก็บข้อมูล	150
รูปที่ ค.52 ฟลิบฟลอบอย่างง่าย	151
รูปที่ ค.53 วงจร SR Latch ชนิด NOR Gate หรือ NOR Latch	151
รูปที่ ค.54 รีจิสเตอร์ขนาด 4 บิต	153
รูปที่ ค.55 ชิพรีจิสเตอร์	154
รูปที่ ค.56 ชิพรีจิสเตอร์ขนาด 4 บิต	155
รูปที่ ค.57 การโหลดข้อมูลแบบอนุกรม (Serial)	156
รูปที่ ค.58 การทำงานหมุนเป็นวงกลมของรีจิสเตอร์	156
รูปที่ ค.59 โปรแกรมภาษาวีเอชดีแอลบรรยายดีฟลิปฟลอป	157
รูปที่ ค.60 แผนผังการต่อแผงวงจรเอฟพีจีเอ	158
รูปที่ ค.61 4-Bit Shift Register	159

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ค.62 การกำหนดพอร์ต	161
รูปที่ ค.63 แผนผังการต่อแผงวงจรเอฟพีจีเอ	163
รูปที่ ค.64 แผนผังการต่อแผงวงจรเอฟพีจีเอ	164
รูปที่ ค.65 แผนผังการต่อแผงวงจรเอฟพีจีเอ	165
รูปที่ ค.66 แผนผังการต่อแผงวงจรเอฟพีจีเอ	166
รูปที่ ค.67 แผนผังการต่อแผงวงจรเอฟพีจีเอ	167
รูปที่ ค.68 วงจร MCS 8051	170
รูปที่ ค.69 แผนผังการต่อแผงวงจรเอฟพีจีเอ	174
รูปที่ ค.70 แผนผังการต่อแผงวงจรเอฟพีจีเอ	174
รูปที่ ค.71 แผนผังการต่อแผงวงจรเอฟพีจีเอ	175
รูปที่ ง.1 โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต	177
รูปที่ ง.2 ผลของการจำลองผลการทำงาน	181
รูปที่ ง.3 เฉลยโปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style	181
รูปที่ ง.3 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style	182
รูปที่ ง.4 ผลของการจำลองผลการทำงาน	182
รูปที่ ง.5 เฉลยโปรแกรมภาษาวีเอชดีแอล แบบ Data Flow Style	182
รูปที่ ง.5 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอล แบบ Data Flow Style	183
รูปที่ ง.6 ผลของการจำลองผลการทำงาน	183
รูปที่ ง.7 ภาษาวีเอชดีแอลบรรยายแบบ Behavioral วงจรแนนเกต 3 อินพุต	183
รูปที่ ง.7 (ต่อ) ภาษาวีเอชดีแอลบรรยายแบบ Behavioral วงจรแนนเกต 3 อินพุต	184
รูปที่ ง.8 ผลของการจำลองผลการทำงาน	184
รูปที่ ง.9 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายจาก Behavioral เป็น Data Flow	184

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ง.9 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายจาก Behavioral เป็น Data Flow	185
รูปที่ ง.10 ผลของการจำลองผลการทำงาน	185
รูปที่ ง.11 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral	186
รูปที่ ง.11 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral	187
รูปที่ง.12 ผลของการจำลองผลการทำงาน	187
รูปที่ ง.13 โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต 3 อินพุต	187
รูปที่ ง.13 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต 3 อินพุต	188
รูปที่ ง.14 ผลของการจำลองผลการทำงาน	189
รูปที่ ง.15 โปรแกรมภาษาวีเอชดีแอลบรรยาย ออร์เกต	190
รูปที่ ง.16 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็ม (Half Adder)	190
รูปที่ ง.17 โปรแกรมภาษาวีเอชดีแอลบรรยาย Full Adder	191
รูปที่ ง.18 ผลของการจำลองผลการทำงาน	192
รูปที่ ง.19 เฉลยโปรมแกรมภาษาวีเอชดีแอล บรรยายตัวบวกเต็ม	193
รูปที่ ง.20 เฉลยโปรมแกรมภาษาวีเอชดีแอล บรรยายตัวบวกเต็ม	193
รูปที่ ง.21 เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายวงจรเข้ารหัส จากตารางความจริง	194
รูปที่ ง.21 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายวงจรเข้ารหัส จากตารางความจริง	195
รูปที่ ง.22 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยาย BCD to 7 Segment (1 – 9)	196
รูปที่ ง.22 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยาย BCD to 7 Segment (1 – 9)	197
รูปที่ ง.23 โปรแกรมภาษา VHDL บรรยาย ALU 15 คำสั่งเบื้องต้น	198
รูปที่ ง.23 (ต่อ) โปรแกรมภาษา VHDL บรรยาย ALU 15 คำสั่งเบื้องต้น	199

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ง.24 โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง	200
รูปที่ ง.25 ผลของการจำลองผลการทำงาน	201
รูปที่ ง.26 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวเลื่อนข้อมูล	202
รูปที่ ง.27 เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายตัวนับขึ้น 4 บิต(Counter)	203
รูปที่ ง.28 เฉลยโปรแกรมภาษา วีเอชดีแอลบรรยาย Counter 5 บิต	204
รูปที่ ง.29 โปรแกรมภาษาวีเอชดีแอลนับขึ้นทีละ 1 พร้อมทั้ง แสดงผลออก 7- Segment	205
รูปที่ ง.29 (ต่อ) โปรแกรมภาษาวีเอชดีแอล นับขึ้นทีละ 1 พร้อมทั้ง แสดงผลออก 7- Segment	206
รูปที่ ง.30 โปรแกรมภาษาวีเอชดีแอลบรรยายการทำงานของระบบ ควบคุมสี่แยกไฟแดง	207
รูปที่ ง.30 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายการทำงานของระบบ ควบคุมสี่แยกไฟแดง	208
รูปที่ ง.31 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายระบบ การควบคุมสเต็ปปิ้งมอเตอร์	208
รูปที่ ง.31 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายระบบ การควบคุมสเต็ปปิ้งมอเตอร์	209
รูปที่ ง.31 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายระบบ การควบคุมสเต็ปปิ้งมอเตอร์	210
รูปที่ ง.32 โปรแกรมภาษาวีเอชดีแอลโดยรับค่าจากวงจรแปลงสัญญาณแอนะลอก เป็นดิจิตอล มาเก็บไว้ในRAM 15 ตำแหน่ง	211
รูปที่ ง.33 โปรแกรมภาษาวีเอชดีแอลบรรยายวงจรคูณเลขฐาน 2(4 บิต) โดยดึงค่าตัวเลขใน อีพรอม	212
รูปที่ ง.34 เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายควบคุมการทำงานของวงจร แปลงสัญญาณดิจิตอลเป็นแอนะลอกโดยให้รับ ค่าจากสวิตช์แมททริกซ์	213
รูปที่ ง.35 โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง	214
รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง	215

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง	216
รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง	217
รูปที่ ง.36 โปรแกรมภาษาวีเอชดีแอล ติดต่อกับ MCS 8051	218
รูปที่ ง.37 โปรแกรมภาษาวีเอชดีแอลบรรยายการรับค่าเคาเตอร์ 0-9 จาก MCS 8051 มาแสดงผลออก 7 - Segment	219
รูปที่ จ.1 แผงวงจรเอฟพีจีเอ	222
รูปที่ จ.2 แผงวงจรซีพีแอลดี	223
รูปที่ จ.3 โมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล	224
รูปที่ จ.4 โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก	225
รูปที่ จ.5 โมดูลวงจรกำเนิดสัญญาณนาฬิกา	226
รูปที่ จ.6 โมดูลมอเตอร์ไฟฟ้ากระแสตรง	227
รูปที่ จ.7 โมดูลดิพสวิตช์ 8 บิต	228
รูปที่ จ.8 โมดูลอีพรอม 256 กิโลไบต์	229
รูปที่ จ.9 โปรแกรมที่ใช้เขียนลงในอีพรอม	238
รูปที่ จ.10 โมดูลคีย์บอร์ดเมตริกซ์ 4 x 4	239
รูปที่ จ.11 โมดูลวงจรสี่แยกไฟแดง	240
รูปที่ จ.12 โมดูลแอลอีดี 40 ดวง	241
รูปที่ จ.13 โมดูล MCS 8051	242
รูปที่ จ.14 โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์, 9 โวลต์ และ12 โวลต์	243
รูปที่ จ.15 โมดูลแรม 128 กิโลไบต์	244
รูปที่ จ.16 โมดูลแสดงผลเจ็ดส่วน	245
รูปที่ จ.17 โมดูลลำโพง	246
รูปที่ จ.18 โมดูลสเตปปิ้งมอเตอร์	247
รูปที่ จ.19 โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว	248

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

เนื่องจากหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรมบัณฑิต มีวิชาการทดลองปฏิบัติทางอิเล็กทรอนิกส์และคอมพิวเตอร์ 2 และในวิชาดังกล่าวยังมีหัวข้อที่เกี่ยวกับการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอล และอุปกรณ์เอฟพีจีเอและซีพีแอลดี แต่ทั้งนี้ภาควิชาครุศาสตร์วิศวกรรม ยังมีชุดทดลองที่ไม่เพียงพอและยังขาดอุปกรณ์เชื่อมโยงไปยังภายนอก ซึ่งทำให้นักศึกษาไม่สามารถฝึกทักษะได้อย่างสมบูรณ์ รวมทั้งไม่สามารถที่จะเรียนรู้และเข้าใจหลักการทำงานของระบบได้อย่างแท้จริงเพียงแต่ได้เรียนรู้หลักกระบวนการเบื้องต้น จึงขาดทักษะการปฏิบัติในการทำงานของระบบดิจิตอล ด้วยเหตุนี้จึงนำความรู้ไปประยุกต์ใช้งานได้อย่างจำกัด นอกจากนี้ต้องสั่งซื้อชุดปฏิบัติการดังกล่าวจากผู้แทนจำหน่ายซึ่งมีราคาแพงโดยเฉพาะอย่างยิ่งสายดาวน์โหลดสำหรับโปรแกรมของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

ด้วยเหตุนี้จึง ได้สร้างชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี ซึ่งจากชุดปฏิบัติการทั้งหมดดังกล่าวสามารถที่จะนำมาใช้งานได้จริง พร้อมทั้งมีประสิทธิภาพตามความต้องการ และมีราคาไม่แพง ฉะนั้นจึงสามารถที่จะสร้างชุดปฏิบัติการดังกล่าวให้มีจำนวนที่เพียงพอกับความต้องการของผู้เรียนได้

1.2 ขีดความสามารถของโครงงาน

1) ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ ประกอบด้วยแผงวงจรการทดลอง 3 แผงวงจร คือ แผงวงจรเอฟพีจีเอ , แผงวงจรซีพีแอลดีและแผงวงจรอินพุตเอาต์พุต

2) แผงวงจรเอฟพีจีเอ ใช้กับชิพของบริษัท Xilinx ตระกูล XC4000 หรือ XC5200 ขนาด 84 ขา โดยใช้แรงดันไฟฟ้ากระแสตรงภายนอก 9 โวลท์ , มีพอร์ตสำหรับการโปรแกรม Configuration ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถต่อกับสายดาวน์โหลดแบบขนานได้, มีพอร์ตเชื่อมต่อไปยังภายนอกได้, ใช้ซอฟต์แวร์ Xilinx Foundation F2.1I ในการออกแบบ

3) แผงวงจร ซีพีแอลดี ใช้ชิพของบริษัท Xilinx ตระกูล 9500 ขนาด 44 ขา โดยใช้แรงดันไฟฟ้ากระแสตรงภายนอก 9 โวลท์, มีวงจร JTAG Programmer สำหรับโปรแกรมตัวชิพ ซีพีแอลดี ได้โดยผ่านพอร์ตแบบขนาน, มีพอร์ตเชื่อมต่อไปยังภายนอกได้, ใช้ซอฟต์แวร์ Xilinx Foundation F2.1I ในการออกแบบ

4) แผงวงจรอินพุตเอาต์พุต มีตัวแสดงผลเจ็ดส่วน, ดิฟสวิตช์ 8 บิต, คีย์บอร์ดเมตริกซ์4 x 4 , สเตปปิ้งมอเตอร์, แอลอีดี 40 ดวง, แปลงสัญญาณดิจิตอลเป็นแอนะลอก, แปลงสัญญาณแอนะลอกเป็นดิจิตอล, มอเตอร์ไฟฟ้ากระแสตรง, ลำโพง, แรม 128 กิโลไบต์, อีพรอม 256 กิโลไบต์, แหล่งจ่ายไฟตรง 5 โวลต์, 9 โวลต์ และ 12 โวลต์จ่ายไฟเลี้ยงแผงวงจรย่อย, สวิตช์กดติดปล่อยดับ 16 ตัว , ชุดทดลองสี่แยกไฟแดง, วงจรกำเนิดสัญญาณนาฬิกา, ชุด MCS 8051

5) มีใบงานประกอบการทดลองไม่น้อยกว่า 6 ใบงาน

5.1) ใบงานที่ 1 เรื่องการศึกษาโปรแกรม Xilinx Foundation F2.1I

5.2) ใบงานที่ 2 เรื่องการใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษา วีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

5.3) ใบงานที่ 3 เรื่องการเขียนภาษาวีเอชดีแอล เบื้องต้น และการเลียนแบบการทำงาน

5.4) ใบงานที่ 4 เรื่องการออกแบบวงจรคอมบิเนชั่นด้วยภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

5.5) ใบงานที่ 5 เรื่องการออกแบบวงจรซีเควนเชียลด้วยภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

5.6) ใบงานที่ 6 เรื่องการออกแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

5.7) ใบงานที่ 7 เรื่องการออกแบบระบบดิจิตอลร่วมกับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

ในปริญญานิพนธ์ฉบับนี้จะมีเนื้อหาซึ่งประกอบด้วยรายละเอียดดังนี้

บทที่ 2 ทฤษฎีและหลักการ ซึ่งกล่าวถึง รายละเอียดเบื้องต้นเกี่ยวกับภาษาวีเอชดีแอล และรายละเอียดเกี่ยวกับการใช้อุปกรณ์เอฟพีจีเอและซีพีแอลดี และขั้นตอนการใช้โปรแกรม Xilinx Foundation F2.1I ของบริษัทไซลิงค์ (Xilinx)

บทที่ 3 การออกแบบการสร้าง และการทำงาน ขั้นตอนการออกแบบและการสร้างชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

บทที่ 4 การทดลองและผลการทดลอง ทำการทดสอบการใช้งานของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา เพื่อนำไปประยุกต์ใช้ต่อภายในอนาคตให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์

ภาคผนวก ค ใบงานการทดลอง

ภาคผนวก ง เฉลยใบงานการทดลอง

ภาคผนวก จ คู่มือการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

วีเอชดีแอล ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาของโปรแกรมในระดับสูง (High Level Language) ที่ใช้ในการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ลักษณะของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปแบบของลำดับชั้น (Hierarchy) ได้และสามารถที่จะเขียนได้หลายรูปแบบซึ่งจะกล่าวต่อไป จึงทำให้ภาษา วีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุดคือแนวความคิดที่จะแก้ปัญหาลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาสามารถเปิดโอกาสให้วิศวกรได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชั่นการทำงานของวงจรอย่างสังเขปโดยที่ยังไม่ต้องไปคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น วีเอชดีแอล ยังเป็นภาษาที่สนับสนุนลักษณะต่าง ๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งาน

2.2 ประวัติความเป็นมาของภาษาวีเอชดีแอล

วิวัฒนาการของภาษาวีเอชดีแอล นั้นเริ่มต้นประมาณปี พ.ศ.2534 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ ดีโอดี (DoD : Department of Defense) มองเห็นว่าอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร เป็นอุปกรณ์ที่ได้รับการพัฒนามาเมื่อประมาณ 20 ปีก่อน เพราะเทคโนโลยีในขณะนั้นทำให้การพัฒนาอุปกรณ์อิเล็กทรอนิกส์เป็นไปอย่างล่าช้า ซึ่งเป็นสภาพที่ไม่อาจยอมรับได้ในปัจจุบัน เพราะเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์ ได้รับการพัฒนาไปอย่างรวดเร็ว ดังที่จะเห็นได้ว่ามีวงจรดิจิตอลอิเล็กทรอนิกส์หลายวงจร ที่แต่เดิมถูกสร้างขึ้นมาจากอุปกรณ์หลายชิ้นที่ถูกนำมาประกอบกันอยู่บนแผงวงจรไฟฟ้า ที่มีขนาดใหญ่ แต่ในปัจจุบันสามารถที่จะใช้เทคโนโลยีการออกแบบและผลิตวงจรรวมขนาดใหญ่มาก (VLSI : Very Large Scale Integration) รวมอุปกรณ์ต่าง ๆ เหล่านั้นให้อยู่บนชิ้นอุปกรณ์สารกึ่งตัวนำ ที่มีขนาดประมาณ 1 – 2 ตร.ซม. ได้เป็นผลให้การทำงานของวงจรสูงขึ้นมีประสิทธิภาพสูงขึ้น (ความเร็วในการทำงานของวงจร) ตลอดจนความน่าเชื่อถือในการทำงาน และความคงทนต่อสภาพแวดล้อมสูง ขณะเดียวกันนั้นในวงการทหารได้มีการนำระบบคอมพิวเตอร์และอิเล็กทรอนิกส์ มาใช้ในระบบอาวุธอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพร่หลาย ดังนั้นอุปกรณ์ที่มีใช้อยู่จึงไม่เหมาะสมกับเทคโนโลยีด้านอาวุธของประเทศคู่แข่งขัน การที่จะเปลี่ยนอุปกรณ์ใหม่เป็นสิ่งที่ต้องใช้งบประมาณมาก และก็จะประสบกับปัญหาเช่นเดิมคือ อุปกรณ์ใหม่ได้รับการพัฒนามานานแล้วเช่นกัน เพราะในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกรและเวลาสำหรับดำเนินการมาก ฉะนั้นทาง ดีโอดีจึงตั้งโครงการขึ้นมาเพื่อศึกษา วิธีการที่จะช่วยพัฒนาวงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น และโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ วีเอชเอสไอซี (VHSIC) ในระยะแรกนั้นโครงการเป็นความลับทางด้านความมั่นคงของประเทศ และอยู่ในความดูแลควบคุมของ United States International Traffic and Arms Regulation หรือ ไอทีเออาร์ (ITAR) ในปี ค.ศ. 1983 ตามคำแนะนำของคณะทำงาน (Woods Hole Workshop) ทาง ดีโอดีนั้นได้ออกความต้องการมาตรฐานของภาษาที่ใช้สำหรับบรรยายพฤติกรรมของวงจรหรือฮาร์ดแวร์ของระบบสำหรับโครงการวีเอชเอสไอซี ซึ่งมีสาระสำคัญพอสรุปได้ดังนี้

1) ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถจะเข้าใจได้ทั้งคนและเครื่องโดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก

2) สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้

3) ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้น ภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรมการออกแบบฮาร์ดแวร์เรียกว่า "Hardware Description Language" หรือ เอชดีแอล (HDL) เริ่มต้นโครงการดีโอดีได้มอบหมายให้บริษัทไอบีเอ็มและบริษัทเท็กซัสอินสตูลเมนท์และบริษัทอินเตอร์เมทริกซ์เป็นผู้ศึกษาและพัฒนา การดำเนินการได้กระทำอย่างต่อเนื่องและได้ผลเป็นที่น่าพอใจ จนกระทั่งปี ค.ศ. 1985 ทางไอทีเออาร์ได้ยกเลิกข้อจำกัดในการถ่ายทอดเทคโนโลยีทางทหารออกจากโครงการนี้ ดังนั้นภาษา วีเอชดีแอล จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป จนกระทั่งทางไออีอีอี (IEEE) จึงได้รับภาษานี้เข้ามาศึกษาและในประมาณปี ค.ศ.1987 ได้ยอมรับกำหนดมาตรฐานของภาษาโดยให้ชื่อว่า IEEE 1076-1993 หรือ วีเอชดีแอล 1993 การที่ทาง ดีโอดีในขณะนั้นเป็นลูกค้ารายใหญ่ของอุตสาหกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ จึงมีผู้รับโครงการจาก ดีโอดี ไปดำเนินการด้านวิจัยและพัฒนามาก เพื่อที่จะให้เป็นมาตรฐานเดียวกันทั้งหมด ทางดีโอดี จึงกำหนดว่า ในการส่งโครงการนั้นจะต้องเขียนอยู่ในรูปของภาษา วีเอชดีแอล เท่านั้น ซึ่งทำให้เกิดข้อดีต่อ DOD เองที่เป็นมาตรฐานเดียวกัน สามารถนำไปจำลองกับเครื่องคอมพิวเตอร์ได้หลาย ๆ ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนประกอบต่าง ๆ ของภาษาวีเอชดีแอล

ในการเขียนแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบในลักษณะบนลงล่างจะต้องทำความเข้าใจในเรื่องของโครงสร้างและระบบต่าง ๆ ของรูปแบบภาษา วีแอวดีแอล เสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

2.3.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้น ที่เรียกว่าหน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบกำหนดทิศทางการไหลของสัญญาณและประเภทของคำที่สามารถกำหนดให้กับสัญญาณตามจุดต่าง ๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.1 แสดงให้เห็นโครงสร้างอย่างง่าย ๆ ของหน่วยการออกแบบเอนทิตี

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

รูปที่ 2.1 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

ส่วนนี้ขึ้นต้นด้วยคำว่า ENTITY และ IS ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (Component_Name) สำหรับการตั้งชื่อนั้นต้องเป็นไปตามกฎเกณฑ์ของภาษา หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (Input-Output) รวมทั้งค่าพารามิเตอร์อื่นๆ ส่วนนี้เรียกว่าส่วนหัว (Entity Header) และที่สำคัญนั้นคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า END และเครื่องหมายอัฒภาคเสมอ (;)

1) port ชนิดต่างๆ

พอร์ตที่ใช้กันส่วนใหญ่ในภาษา VHDL มีดังนี้

1) in : รับข้อมูลไหลเข้าวงจรได้อย่างเดียว

2) out : ส่งข้อมูลไหลออกจากวงจรได้อย่างเดียว

3) inout : ข้อมูลสามารถไหลเข้าและไหลออกจากพอร์ตนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การตั้งชื่อ (Identifier)

ในการเขียนโปรแกรมด้วยภาษาอะไรก็แล้วแต่ จะต้องมีการตั้งชื่อเสมอ เช่นในการเขียนภาษาปาสคาลจะต้องมีการตั้งชื่อโปรแกรม ชื่อตัวแปร ชื่อโพซีเยอร์ ชื่อฟังก์ชั่น เป็นต้น ในภาษาวีเอชดีแอล ก็เช่นเดียวกัน จะต้องมีการตั้งชื่อของส่วน Entity ชื่อของ Signal ชื่อของส่วน Architecture ชื่อของตัวแปร ชื่อของคอมโพเนนท์ที่ใช้งาน เป็นต้น

ชื่อ (Identifier) ที่จะตั้งในภาษา วีเอชดีแอล จะต้องประกอบด้วยตัวของอักขระตั้งแต่ 1 ตัวขึ้นไปโดยมีความยาวของชื่อได้ไม่จำกัด ค่าที่เป็นไปได้คือ ตัวอักษรพิมพ์ใหญ่ (A...Z) ตัวอักษรพิมพ์เล็ก (a...z) ตัวเลข (0...9) และเครื่องหมาย อันเดอร์สคอร์ (_) โดยมีกฎอยู่ว่า Identifier จะต้องขึ้นต้นด้วยตัวอักษร (A...Z) (a...z) และต้องไม่ลงท้ายด้วยเครื่องหมาย Underscore โดยในภาษา วีเอชดีแอล นั้นจะถือว่าตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ไม่มีความแตกต่างกัน (Non-Case Sensitive) ยกตัวอย่างเช่น ADDER , Adder , Adder และ Adder จะถือว่าเป็นตัวเดียวกัน ในการใช้เครื่องหมาย underscore นั้นจะต้องไม่ใช้ติดกัน นอกจากนี้แล้ว ในการตั้งชื่อจะต้องไม่ใช้คำสงวน (Reserved Word) ของภาษา วีเอชดีแอล ด้วย ตัวอย่างของการตั้งชื่อที่ถูกต้องได้แก่

RESET_SIGNAL DATA_Bus SET_OUT_LOW

CONST34_99 c3po AndGate

ตัวอย่างการตั้งชื่อที่ผิด ได้แก่

2bus_high	ผิด เพราะตัวขึ้นต้นด้วยตัวเลข
_Jkflipflop	ผิด เพราะขึ้นต้นด้วยเครื่องหมาย underscore
Data_bus_	ผิด เพราะลงท้ายด้วยเครื่องหมาย underscore
Memory_address	ผิด เพราะใช้เครื่องหมาย underscore 2 ตัวติดกัน

3) VHDL Opertors

Operator ที่ถูกกำหนดไว้แล้ว (predefined operator) ในภาษา VHDL นั้นสามารถแบ่งออกเป็น 4 ประเภท ดังต่อไปนี้

3.1) Logical operators

ใช้ในการทำโอเปอเรชั่นทางลอจิก ซึ่งมีอยู่ 6 ตัวด้วยกัน อันได้แก่

And Or Nand Nor Xor Not

โอเปอเรเตอร์ดังกล่าวใช้ได้สำหรับข้อมูลชนิด Bit, Bit_vector และBoolean โดยที่โอเปอเรชั่น Not จะมีลำดับความสำคัญ (Precedence) สูงสุด ส่วนโอเปอเรชั่น And , Or , Nand , Nor และ Xor จะมีลำดับความสำคัญเท่ากัน และมีลำดับความสำคัญน้อยกว่าโอเปอเรชั่น Not ในการทำโอเปอเรชั่นทางลอจิกนั้นข้อมูลที่นำมาทำโอเปอเรชั่นกันจะต้องเป็นข้อมูลชนิดเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และผลลัพธ์จากการทำโอเปอเรชั่นให้เอาต์พุตชนิดเดียวกันกับอินพุต ตัวอย่างของการทำโอเปอเรชั่นทางลอจิก ได้แก่

A and B

Not (CLK orRESET)

ADDREESS_BUS(2 downto 0) Nor “010”

3.2) Relational Operators

ใช้ในการเปรียบเทียบโอเปอแรนด์ชนิดใดก็ได้ยกเว้น File Type ซึ่งผลลัพธ์จากการเปรียบเทียบจะมีชนิดเป็น Boolean โดยที่โอเปอเรเตอร์แบบ relational มีดังนี้

=	เท่ากับ
/=	ไม่เท่ากับ
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
>	มากกว่า
<	น้อยกว่า

ในการเปรียบเทียบออบเจกต์ที่เป็นชนิด Bit_Vector นั้นออบเจกต์ที่นำมาเปรียบเทียบกันไม่จำเป็นต้องมีจำนวนบิตที่เท่ากันก็ได้ ดังตัวอย่าง

1 0 1 0 = A

1 1 1 = B

ผลลัพธ์จากการเปรียบเทียบ A กับ B ก็คือ B มีมากกว่า A เพราะจำนวนบิตของการเปรียบเทียบจะเท่ากับจำนวนบิตของโอเปอแรนด์ตัวที่มีจำนวนบิตน้อยที่สุด โดยเริ่มเปรียบเทียบจาก MSB ลงมาจนกระทั่งถึง LSB ดังนั้นค่า “111” จึงถูกเปรียบเทียบกับค่า “101” จึงให้ผลออกมาว่า B มีค่ามากกว่า A

3.3) Arithmetic Operators

ใช้ในการทำโอเปอเรชั่นทางคณิตศาสตร์ ซึ่งแสดงดังตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การทำโอเปอเรชั่นทางคณิตศาสตร์

Operator	Operation
+	บวก
-	ลบ
*	คูณ
/	หาร
mod	modulus
abs	หาค่าสมบูรณ์
**	ยกกำลัง

3.4) Concatenation operators

ใช้ในการเชื่อมต่อข้อมูลตัวเดียว ๆ หรือข้อมูลชนิดอะเรย์ 1 มิติเข้าด้วยกัน โดยให้ผลลัพธ์ออกมาเป็นข้อมูลชนิดอะเรย์เสมอ ซึ่งสัญลักษณ์ที่ใช้คือเครื่องหมาย & ตัวอย่างเช่น

'0' & '1' จะได้ผลลัพธ์เท่ากับ "01"

'F' & 'A' & 'T' จะได้ผลลัพธ์เท่ากับ "FAT"

"FRO" & "NT" จะได้ผลลัพธ์เท่ากับ "FRONT"

4) Data Objects

Data Object มีอยู่ 3 ชนิด อันได้แก่

4.1) การประกาศค่า constant (constant declaration)

ตัวอย่างของการประกาศค่า Constant ได้แก่

```
constant CLOCK_PERIOD : time := 10 ns;
constant BUS_WIDTH : integer : 32;
```

รูปที่ 2.2 ตัวอย่างของการประกาศค่า Constant

รูปที่ 2.2 เป็นการประกาศออบเจกต์ชื่อ CLOCK_PERIOD ให้มีชนิดของออบเจกต์เป็นชนิด time (เป็น Type มาตรฐานตัวหนึ่งในภาษา VHDL) และมีค่าเท่ากับ 10 ns รูปที่ 2.2 เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการประกาศออบเจกต์ชื่อ BUS_WIDTH ให้มีชนิดของออบเจกต์เป็นชนิด integer ดังนั้น ค่าที่ ออบเจกต์ BUS_WIDTH จะเก็บได้ก็คือตัวเลขจำนวนเต็ม ซึ่งมีค่าเท่ากับ 8

4.2) การประกาศ Variable

ตัวอย่างของการประกาศ Variable ได้แก่

```
variable A_LT_B , DONE : boolean;
variable REMAINDER : integer range 0 to 127 := 10;
variable INT_SIGNAL :  bit_vector (10 downto 0) :;
```

รูปที่ 2.3 ตัวอย่างของการประกาศ Variable

รูปที่ 2.3 เป็นการประกาศ Variable ชื่อ A_LT_B และ variable ชื่อ DONE ให้มีชนิดเป็น Boolean ซึ่งค่าที่สามารถเก็บได้จะมีแค่ค่า True กับ False เท่านั้น ตัวอย่างที่ 2 เป็นการประกาศ Variable ชื่อ REMAINDER ซึ่งมีชนิดเป็น Integer โดยมีการกำหนดขอบเขตให้ค่าที่ REMAINDER จัดเก็บได้นี้จำกัดอยู่ที่ค่าในช่วง 0 ถึง 127 เท่านั้นและค่าเริ่มต้น (Initial Value) ของ Variable มีค่าเท่ากับ 10 ตัวอย่างที่ 3 เป็นการประกาศ Variable ชื่อ INT_SIGNAL ให้มีชนิดเป็น Bit_Vector ซึ่งในการประกาศไม่มีการกำหนดค่าเริ่มต้นเอาไว้

4.3) การประกาศ Signal

ตัวอย่างของการประกาศ Signal ได้แก่

```
Signal RESET : bit := '0';
Signal SUM : bit_vector (31 downto 0);
Signal DIVIDEND : integer;
```

รูปที่ 2.4 ตัวอย่างของการประกาศ Signal

รูปที่ 2.4 เป็นการประกาศ Signal ชื่อ RESET ให้มีชนิดเป็น bit และมีค่าเริ่มต้นเป็นลอจิก '1' ตัวอย่างที่ 2 เป็นการประกาศ Signal ชื่อ SUM ให้มีชนิดเป็นอะเรย์ของข้อมูลชนิด Bit ขนาด 32 บิต ตัวอย่างสุดท้ายเป็นการประกาศ Signal ชื่อ DIVIDEND ให้มีชนิดเป็น integer

5) ชนิดของข้อมูล (Data Type)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Type ที่มีอยู่แล้วในภาษาวีเอชดีแอล (Predefined Type) ได้แก่

1) CHARACTER มีค่าที่เป็นไปได้ดังในรูปที่ 2.5

2) BIT ที่ค่าที่เป็นไปได้ คือ '0' และ '1'

3) BIT_VECTOR เป็นอะเรย์ของข้อมูลชนิดบิต

4) BOOLEAN มีค่าที่เป็นไปได้ คือ true และ false

5) INTEGER มีค่าที่เป็นได้คือ ตั้งแต่ $-(2^{31} - 1)$ ไปจนถึง $+(2^{31} - 1)$

6) REAL ซึ่งมีค่าได้ตั้งแต่ $-(1.0 * 10^{38})$ ไปจนถึง $+ (1.0 * 10^{38})$ ค่าที่สามารถรับได้จะมีความละเอียดของเลขทศนิยมไม่เกิน 6 หลัก

7) STRING เป็นอะเรย์ของข้อมูลชนิด CHARACTER

nul, soh, stx, etx, eot, enp, ack, bel,

bs, ht, lf, vt, ff, cr, so, si,

dle, dc1, dc2, dc3, dc4, nak, syn, etb,

can, em, sub, esc, fsp, gsp, rsp, usp,

' ', '!', '"', '#', '$', '%', '&', ''',

'(', ')', '*', '+', ',', '-', '.', '/',

'0', '1', '2', '3', '4', '5', '6', '7',

'8', '9', ':', ';', '<', '=', '>', '?',

'@', 'A', 'B', 'C', 'D', 'E', 'F', 'G',

'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',

'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W',

'X', 'Y', 'Z', '[', '\', ']', '^', '_',

'`', 'a', 'b', 'c', 'd', 'e', 'f', 'g',

'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',

'p', 'q', 'r', 's', 't', 'u', 'v', 'w',

'x', 'y', 'z', '{', '|', '}', '~', del

รูปที่ 2.5 Character มีค่าที่เป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6 เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพล็กซ์ หรือ MUX ในส่วนหัวของเอนทิตี มีการกำหนดจุดต่อ 4 จุดภายใต้ชุดคำสั่ง PORT โดยที่ 3 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ in0, in1, sel ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลเข้าของข้อมูล (IN) ที่แสดงด้วยรูปสี่เหลี่ยมโปร่งในรูปที่ 2.6 ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (OUT) ที่แสดงด้วยรูปสี่เหลี่ยมทึบในรูปที่ 2.6 ส่วนประเภทของข้อมูลที่จะไหลเข้า และออกนั้น เป็นประเภทบิต ที่สามารถมีค่าได้เพียงสองค่าคือ '0' และ '1' เท่านั้น

(ก) หน่วยการออกแบบเอนทิตี (ข) มุมมองของตัวเชื่อมประสาน (Interfacing)

รูปที่ 2.6 รูปแบบของมัลติเพล็กซ์

นอกจากนั้น ผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ที่เป็นข้อมูลเพิ่มเติมอื่นลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์อันได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างในรูปที่ 2.7

(ก) หน่วยการออกแบบเอนทิตี (ข) มุมมองของตัวเชื่อมประสาน (Interfacing)

รูปที่ 2.7 รูปแบบมัลติเพล็กซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบางกรณีสามารถที่จะใช้ภาษา วีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหลเข้า และออกของข้อมูลได้ดังรูปที่ 2.8 ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบสำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่งคือ วีเอชดีแอล สำหรับการทดสอบเปรียบเทียบ (Test Bench)

```
ENTITY test_bench IS
END test_bench;
```

รูปที่ 2.8 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

2.3.2 หน่วยการออกแบบสถาปัตยกรรม

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงาน พฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้า และออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดในหน่วยการออกแบบเอนทิตี รูปที่ 2.9 แสดงให้เห็นถึงโครงสร้างอย่างง่ายของหน่วยการออกแบบสถาปัตยกรรม

```
ARCHITECTURE indentifier 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 2.9 โครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (Identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า ARCHITECTURE นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OF < Entity Design Unit > IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และBEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Architecture Declarative Area) ที่เป็นเพียงส่วนเผื่อเลือก (Option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ในสถาปัตยกรรมนั้นได้ อาทิเช่น ประเภท (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 หน่วยการออกแบบสถาปัตยกรรม จะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้นๆ ที่เป็นส่วนเผื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิตอลด้วยภาษา วีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

1) ประเภทการไหลของข้อมูล (Dataflow Description)
2) ประเภทพฤติกรรม (Behavioral Description)
3) ประเภทโครงสร้าง (Structure Description)
4) ประเภทผสม (Mixed Model Description)

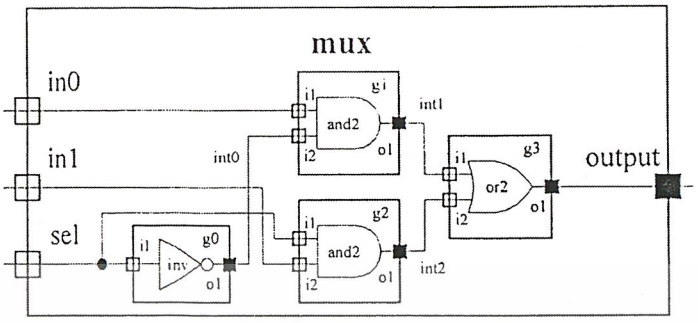
```
ARCHITECTURE data_flow OF mux IS
BEGIN
        Output <= ((NOT sel) AND in0) OR (sel AND in 1);
END data_flow ;
```

รูปที่ 2.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ตามฟังก์ชันบูลีน

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

รูปที่ 2.10 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (In0, In1) กับข้อมูลที่ไหลออก (Output) ประกอบด้วยชุดคำสั่งแบบแข่งขนานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพล็กซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL : Register Transfer Level)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 โครงสร้างภายในสถาปัตยกรรมของมัลติเพล็กซ์

รูปที่ 2.11 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทโครงสร้าง โดยใช้อินเวอร์เตอร์ (Inv ที่ตำแหน่ง G0) , แอนด์เกต 2 อินพุตจำนวน 2 ตัว (And2 ที่ตำแหน่ง G1 และ G2) และ ออร์เกต 2 อินพุต (Or2 ที่ตำแหน่ง G3) มาสร้างตามฟังก์ชันบูลีนของรูปที่ 2.11

```
ARCHITECTURE struc OF mux IS
        COMPONENT inv
        PORT (i1 : IN BIT ; o1 : OUT BIT);
        COMPONENT and2
        PROT (i1, i2 : IN BIT ; o3 : OUT BIT);
        COMPONENT or2
        PORT (i1,i2 : IN BIT ; 01:OUT BIT);
END COMPONENT;
        SIGNAL int0,int1,int2 : BIT;
BEGIN
        g0 : inv    PORT MAP (i1 => sel , o1 => int0);
        g1 : and2  PORT MAP (i1 => int0 , i2 => int0 , o1 => int1);
        g3 : or2    PORT MAP(i1 => int1 , i2 => int2 , o1 => output);
END struc;
```

รูปที่ 2.12 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.13 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทพฤติกรรม การเขียนจะมีลักษณะที่แตกต่างกับแบบโครงสร้างแต่การทำงานเหมือนกัน

```
ARCHITECTURE behav OF mux IS
BEGIN
        PROCESS (in0, in1, sel)
                IF (sel = '0') THEN output <= in0;
                ELSE output <= in1;
                END IF;
        END PROCESS;
END behav;
```

รูปที่ 2.13 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพล็กซ์ประเภทพฤติกรรม

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพล็กซ์ในลักษณะของ ประเภทพฤติกรรมประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกันและจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกันซึ่งนี่ก็เป็นข้อดีของภาษา วีเอชดีแอล

2.3.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อยที่สามารถเป็นประโยชน์ต่อการเขียนในรูปแบบบรรยายระบบดิจิตอลสามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้งานได้โดย หน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือ จากหน่วยการออกแบบแพ็กเก็ตอื่น ๆ นอกจากนั้นสิ่งที่นิยมทำกันมากคือรูปแบบมาตรฐานต่าง ๆ เช่น อุปกรณ์มาตรฐาน (เช่น IC ตระกูล 74XX เป็นต้น) จะถูกเก็บไว้ในแพ็กเก็ตที่ทุกคนเข้าถึงโดยปกติแล้ว แพ็กเก็ตจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กเก็ต และ ส่วนของบอดี้แพ็กเก็ตเนื่องจากแพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

1) Package Declaration

ส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (ถ้ามองในแง่การนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อ ของส่วนที่ประกาศอยู่ภายในแพ็กเก็ต สำหรับการนำไปใช้ภายนอก ตัวของแพ็กเก็ตของสิ่งใด ๆ ถูกประกาศในส่วนของส่วนบอดี้แพ็กเก็ตแต่ไม่ถูกประกาศในส่วนการประกาศแพ็กเก็ต จะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตี้คือ จุดเชื่อมต่อ หรือพอร์ตที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้ว แพ็กเก็ตสามารถสร้างขึ้นได้โดยไม่ต้องมีส่วนบอดี้ และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศชนิด (TYPE) หรือสัญญาณ เช่นเดียวกันกับ ส่วนบอดี้แพ็กเก็ตที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นไม่สมารถนำไปใช้จากรูปแบบอื่นได้ ดังรูป 2.14

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

รูปที่ 2.14 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

2) Package Body

โครงสร้างที่ประกอบด้วยส่วนต่าง ๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชั่นการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ตแล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ต ทั้งนี้รวมทั้งการกำหนดค่าคงที่ต่างอันได้แก่การตั้งคงที่ ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ตแต่ถูกกำหนดในส่วนของบอดี้แพ็กเก็ต ฉะนั้นส่วนบอดี้แพ็กเก็ตจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเก็ตไม่มีการประกาศชื่อ ที่เป็นโปรแกรมย่อยหรือค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูป 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
48327

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

รูปที่ 2.15 โครงสร้างของบอดี้แพ็กเก็ต

2.3.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตีหนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกันดังรูปที่ 2.16

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

รูปที่ 2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.4 ภาษาวีเอชดีแอล เพื่อการสังเคราะห์

ภาษาวีเอชดีแอลเป็นภาษาที่เขียนเพื่อจำลองการทำงานของวงจร ซึ่งในบางรูปแบบการเขียนไม่สามารถที่จะนำไปสังเคราะห์ได้ทั้งหมด ดังนั้นถ้าต้องการเขียนเพื่อนำไปสังเคราะห์ ควรหลีกเลี่ยงรูปแบบต่าง ๆไม่สามารถนำไปสังเคราะห์ได้ ในที่นี้ขึ้นอยู่กับความสามารถของโปรแกรมที่ใช้สังเคราะห์แต่ละโปรแกรม ดังนั้นในหัวข้อนี้จะแสดงตัวอย่างของการเขียนโมดูลในรูปแบบต่างๆ ที่สามารถนำไปสังเคราะห์ซึ่งยึดหลักการเขียนตาม View Synthesis User's Guide ของโปรแกรม Viewlogic ซึ่งเป็นโปรแกรมที่ใช้ในการสังเคราะห์วงจรทั้งหมด ในการออกแบบไมโครคอนโทรลเลอร์ โดยแบ่งออกเป็น 2 กลุ่ม คือ เกตและฟลิปฟลอปประเภทต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ตัวอย่างรูปแบบการเขียนเกตพื้นฐาน

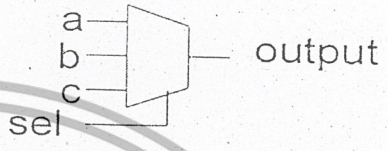
```
SIGNAL a, b, c, d, input, output : vlbit_1d(3 DOWNTO 0);
SIGNAL sel   : vlbit_1d(1 DOWNTO 0);
SIGNAL enb : vlbit;
```

ตารางที่ 2.2 ลักษณะการทำงานของเกตพื้นฐานต่าง ๆ

สมการ	สัญลักษณ์
output <= a AND b;	
… output <= a OR b;	
… output <= a XOR b;	
… output <= NOT input;	
… output <= input WHEN enb='1' ELSE "ZZ";	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

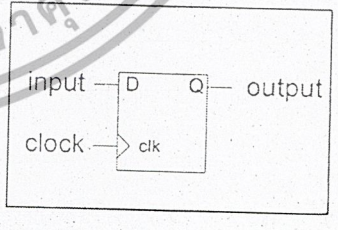
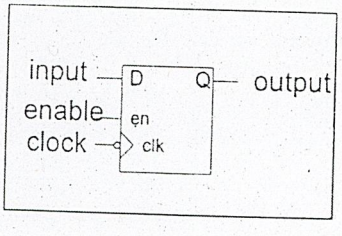
ตารางที่ 2.2 (ต่อ) ลักษณะการทำงานของเกตพื้นฐานต่าง

สมการ	สัญลักษณ์
... WITH sel SELECT Output <= a WHEN "00" ELSE b WHEN "01" ELSE c;	

2.4.2 ตัวอย่างรูปแบบการเขียนฟลิปฟลอปพื้นฐาน

```
SIGNAL Input, Output : Vlbit_1d(3 DOWNTO 0);
SIGNAL Clock, Enable, Reset : Vlbit;
```

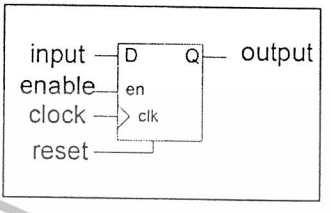
ตารางที่ 2.3 ลักษณะการทำงานของเกตพื้นฐานต่างด้วยการเขียนโปรแกรมวีเอชดีแอล

สมการ	สัญลักษณ์
... PROCESS BEGIN WAIT UNTIL PRISING (clock); Output <= input; END PROCESS;	
PROCESS BEGIN WAIT UNTIL PFALLING (clock); output <= input; END PROCESS;	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 (ต่อ) ลักษณะการทำงานของเกตพื้นฐานต่างด้วยการเขียนโปรแกรมวีเอชดีแอล

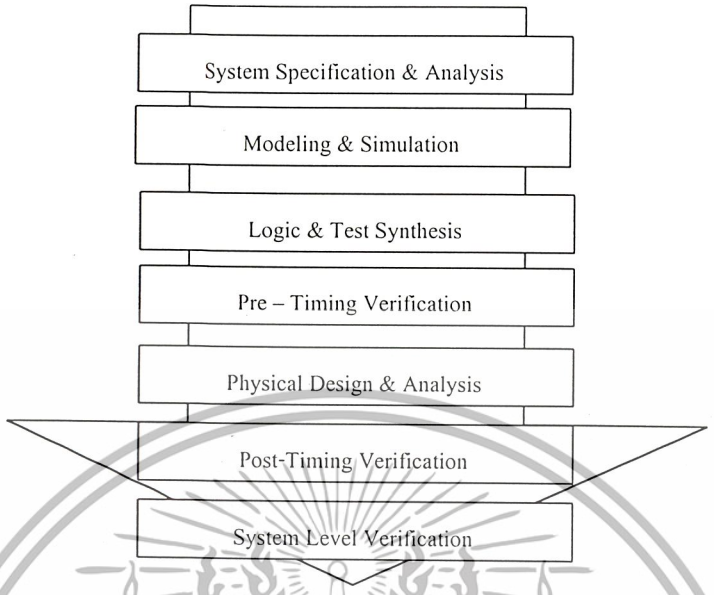
สมการ	สัญลักษณ์

PROCESS BEGIN
WAIT UNTIL PRISING (clock) OR (reset='1');
IF (reset = '1') THEN output <= "00";
ELSIF (enable = '1') THEN output <= input;
END IF;
END PROCESS;
	

2.5 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน เช่น วงจรรวม (ASIC : Application Specific Integrated Circuit) วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของ บล็อกไดอะแกรมเสียก่อน ก่อนที่จะวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามที่ต้องการ และเพิ่มเติมในรายละเอียดทีละชั้นนี้คือหลักการออกแบบจากบนลงล่าง (Top-Down Design) ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom-up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาในการออกแบบมากกว่า 90% เพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่าง ๆ (Schematic Capture) ที่ประกอบเข้ากันเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามากและถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากให้การออกแบบในลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอล กับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรออกแบบที่จะสามารถออกแบบและพัฒนาวงจรที่มีความซับซ้อนได้มากขึ้น และช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.17 แสดงให้เห็นถึงขั้นตอนของการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลาย ๆ 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 หรือภาษาเอชดีแอล อื่น ๆ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3) หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจร หรือสังเคราะห์ ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกต และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสท์ (Netlist) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4) หลังจากการสังเคราะห์วงจรให้อยู่ในรูประดับเกตหรือเน็ทลิสท์แล้ว ข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงาน ในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี เวลาหน่วงของการแพร่กระจาย (Propagation Delay Time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ นาโนวินาที (10^{-9} นาที) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่าง ๆ จำนวน 10,000 เกต ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5) ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวม (ASIC)

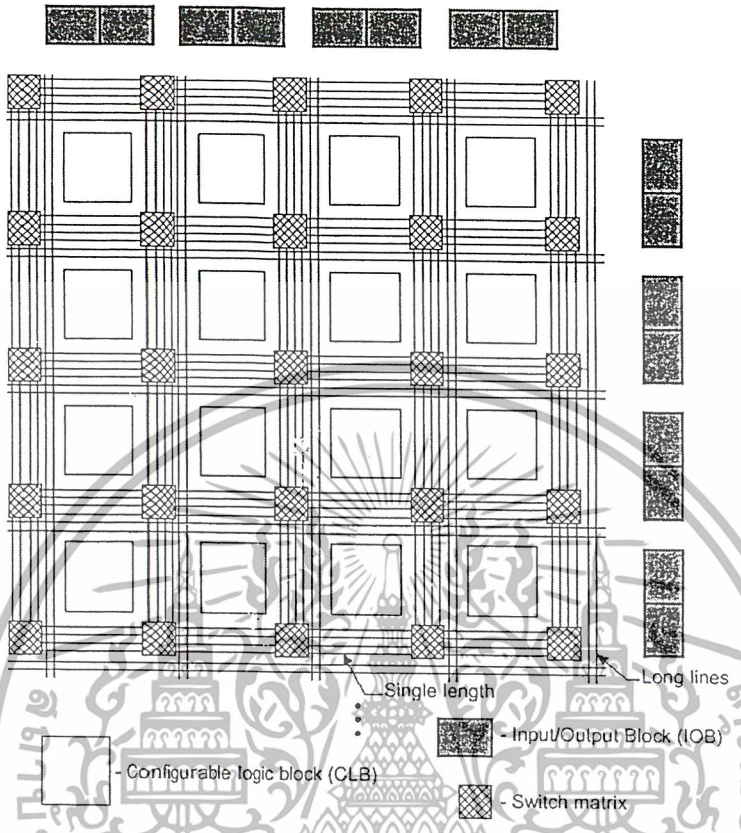
6) หลังจากที่ได้วงจรจริงมาแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วยเพื่อความถูกต้องของวงจรครั้งสุดท้าย ก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วยอินพุตและเอาต์พุตแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7) หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆอีกครั้งเป็นการควบคุมคุณภาพของ ผลิตภัณฑ์

2.6 เอฟพีจีเอ (FPGA : Field Programmable Gate Array)

เป็นอุปกรณ์ที่ถูกพัฒนาต่อจากอุปกรณ์แอลซีเอของบริษัทไซลิงซ์ (XILINX Inc.) โดยมีประสิทธิภาพในการทำงานและมีปริมาณความหนาแน่นของเกตสูง สามารถจะกำหนดฟังก์ชั่นการทำงานได้ตามความต้องการของผู้ใช้โดยผ่านการโปรแกรมเอฟพีจีเอ มีข้อดี ได้แก่ การออกแบบการผลิต, ลดเวลาที่จะส่งตัวผลิตภัณฑ์ออกตลาด ซึ่งเป็นประโยชน์ต่อการผลิตวงจรเป็นอย่างมาก นักออกแบบเพียงกำหนดฟังก์ชั่นการทำงานของวงจร ดังนั้นการออกแบบโดยใช้เอฟพีจีเอ สามารถออกแบบและทดสอบภายในเวลาเพียง 2 — 3 วัน เท่านั้น ตรงกันข้ามกับการออกแบบวงจรโดยใช้เกตอาเรย์ ซึ่งใช้เวลาหลายอาทิตย์ การเปลี่ยนแปลงแก้ไขแบบก็เช่นเดียวกัน จากประโยชน์ของเอฟพีจีเอ ดังกล่าวมาทำให้เกิดการประหยัดค่าใช้จ่ายและเวลาเป็นอย่างมาก โครงสร้างภายในของเอฟพีจีเอ จะเป็นดังรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 โครงสร้างภายในของเอฟพีจีเอของบริษัทไซลิงซ์ (XILINX Inc.)

2.6.1 สถาปัตยกรรมภายในของเอฟพีจีเอตระกูล XC 4000

สถาปัตยกรรมภายในคล้าย ๆ กับเกตอาเรย์โดยทั่วไปจะมีส่วนประกอบที่สำคัญดังนี้

1) ซีแอลบี (CLB : Configuration Logic Block)

2) ไอโอบี (IOB : Input Output Block)

3) อินเตอร์คอนเน็ค (Interconnection)

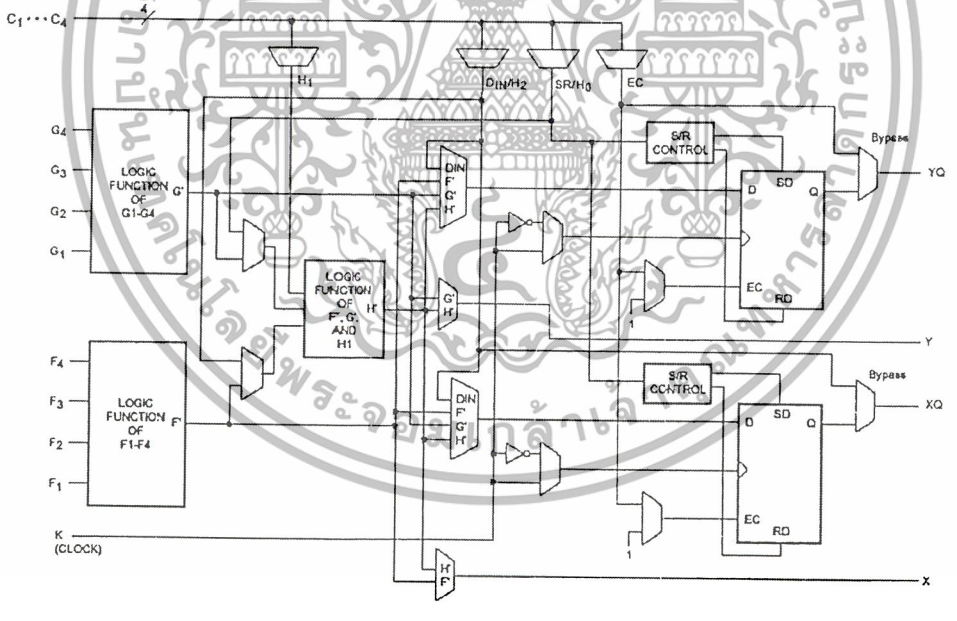
ภายในมีลักษณะเป็นเมทริกซ์ของลอจิกบล็อก (Logic block) และล้อมรอบไปด้วยบล็อกการเชื่อมต่อของไอโอ (I/O Interface block) การเชื่อมต่อระหว่างซีแอลบี (CLB : Configuration Logic Block) และไอโอบี (IOB : Input Output Block) ทำได้โดยผ่านช่องที่ว่างพาดผ่านระหว่างแถว (Row) และคอลัมน์ (Column) มีการทำงาน เหมือนกันไมโครโพรเซสเซอร์ ตัวแอลซีเอจะทำงานได้ต้องใช้ Program - Driven Logic Device หน้าที่ของซีแอลบีและไอโอบีแต่ละตัว การเชื่อมต่อภายใน (Interconnection) ถูกกำหนดไว้ในโปรแกรมคอนฟิกคูเรชัน (Configuration Program) หรือเก็บไว้ในอีพรอม ภายในแอลซีเอโปรแกรมจะถูกโหลดเข้าสู่แอลซีเอ เมื่อมีการจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Power Up)โดยทางคำสั่ง(Command) ซึ่งเป็นส่วนหนึ่งของการเริ่มต้นระบบ (System Initialization) ประสิทธิภาพของแอลซีเอกำหนดโดยความเร็วของลอจิกส่วนประกอบหน่วยความจำและการโปรแกรมการเชื่อมต่อต่างๆ ความเร็วของอัตราของระบบสัญญาณนาฬิกา (System Clock Rate) ถูกกำหนดด้วยทอกเกิลฟลิปฟลอป สำหรับการประยุกต์ใช้โดยทั่วไปจะอยู่ที่ประมาณ 1/3 ถึง 1/2 ค่าสูงสุดของทอกเกิลเกต (Maximum Toggle Gate)

1) ซีแอลบี (CLB : Configuration Logic Block)

ภายในแอลซีเอ คือเมทริกซ์ของซีแอลบีแต่ละตัวประกอบด้วยหน่วยของคอมบิเนชั่นลอจิกที่สามารถโปรแกรมได้ (Programmable Combination Logic) และส่วนของรีจิสเตอร์เก็บข้อมูล (Storage Register) ส่วนของวงจรคอมบิเนชั่นลอจิกสามารถใช้สร้างวงจรทางด้านฟังก์ชั่นบูลีนของอินพุต ส่วนรีจิสเตอร์รับค่าจากส่วนคอมบิเนชั่นหรือ โดยตรงจากเอาต์พุตของซีแอลบี สามารถขับวงจรคอมบิเนชั่นลอจิกโดยตรงผ่านเส้นทางเดินย้อนกลับ (Feedback Path) ดังรูปที่ 2.19

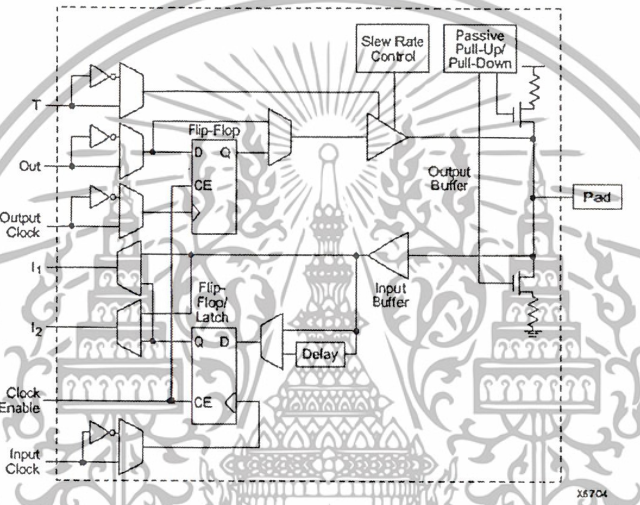


รูปที่ 2.19 ผังวงจรภายในของซีแอลบีของเอฟพีจีเอตระกูล XC4000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ไอโอบี (IOB : Input Output Block)

เป็นส่วนติดต่อกับวงจรภายนอกของแอลซีเอที่สร้างมาจากส่วนของอุปกรณ์ อินพุต/เอาต์พุตที่สามารถโปรแกรมได้ (Programmable Input/Output Device) แต่ละตัวสามารถโปรแกรมได้อย่างอิสระโดยจะให้เป็นอินพุต/เอาต์พุตแบบ3สถานะหรือไอโอแบบสองทิศทางก็ได้โดยอินพุตสามารถโปรแกรมให้รู้จักทั้งระดับสัญญาณทีทีแอลและซีมอสเทรดโฮลของไอโอบี แต่ละตัวมีฟลิปฟลอป ที่สามารถใช้เป็นบัฟเฟอร์สำหรับอินพุตและเอาต์พุต ดังรูปที่ 2.20

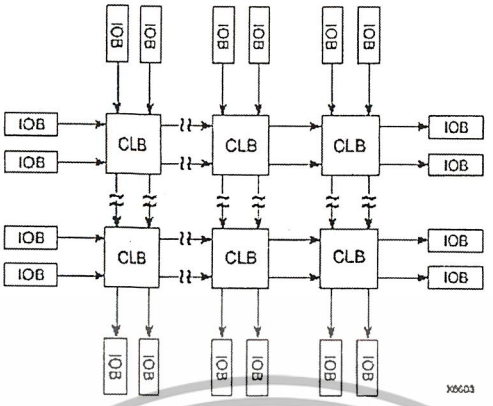


รูปที่ 2.20 ผังวงจรของไอโอบีภายในเอฟพีจีเอตระกูล XC4000

3) อินเตอร์คอนเน็ค (Interconnect)

ความยืดหยุ่นของการใช้แอลซีเอมาทำเป็นอุปกรณ์ขึ้นอยู่กับการโปรแกรม ทรัพยากรต่าง ๆ ที่อยู่ภายในเข้าด้วยกันการที่จะควบคุมการเชื่อมต่อระหว่างจุดสองจุดภายในชิปเหมือนกับเกตอาเรย์ทั่ว ๆ ไป การเชื่อมต่อภายในแอลซีเอประกอบด้วยเน็ตเวริ์ค 2 ทิศทางคือทาง แถวและคอลัมน์ซึ่งวางอยู่ระหว่าง CLB Programmable Switch จะทำการเชื่อมต่ออินพุตและเอาต์พุตของไอโอบีและซีแอลบีที่จุดต่อระหว่างแถวกับคอลัมน์สามารถสลับสัญญาณจากเส้นทางไปยังส่วนต่างๆดังรูป2.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 เส้นทางการเชื่อมต่อระหว่างไอโอบีกับซีแอลบีของเอฟพีจีเอตระกูล XC4000

2.6.2 คุณสมบัติโดยทั่วไปของเอฟพีจีเอตระกูล XC4000

1) เป็นอุปกรณ์รุ่นที่สามของเอฟพีจีเอ

1.1) มีฟลิปฟลอปเป็นจำนวนมาก

1.2) ในการผลิตฟังก์ชั่นของการทำงานมีความยืดหยุ่นสูง

1.3) มีจำนวนเกตภายในจำนวน 2,000 — 10,000 เกต

1.4) เพิ่มความสามารถพิเศษของรีจิสเตอร์และอินพุต/เอาต์พุต

1.5) มีค่าแฟนเอาต์ (Fan-Out) สูง

1.6) มีบัสภายใน 3 สถานะ

1.7) ทำงานกับสัญญาณทีทีแอลและซีมอส

1.8) มีออสซิเลเตอร์แอมปลิฟลายเออร์ภายใน

1.9) มีแรมภายในความเร็วสูง (< 25 Ns)

1.10)ใช้กับงานที่ต้องการความเร็วสูง (ใช้งานได้ที่ความถี่ 70/100/125 MHz)

1.11) มี Wide Edge Decoder

1.12) เส้นทางการเชื่อมต่อ (Interconnect Line) เป็นแบบลำดับชั้น

1.13) มีการกระจายกำลังงานของสัญญาณต่ำ

2) มีสถาปัตยกรรมภายในที่ยืดหยุ่น

2.1) มีลอจิกบล็อคและไอโอบล็อคที่สามารถโปรแกรมได้

2.2) มีอินเตอร์คอนเน็คและ Wide decoder ที่โปรแกรมได้

3) ทำกระบวนการซับไมครอนชนิดซีมอสได้

3.1) มีลอจิกและอินเตอร์คอนเน็คที่มีความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) ใช้กำลังงานต่ำ

4) คุณลักษณะทาง System - Oriented

4.1) รองรับมาตราฐาน IEEE 1149.1 ในการทำ boundary - scan logic

4.2) สามารถโปรแกรมให้อินพุตมีลักษณะพูลอัพ (Pull-Up) หรือ พลูดาวน์ (Pull-Down) รีจิสเตอร์ได้

4.3) ให้กระแสเอาต์พุตได้ตั้งแต่ 12-24 มิลลิแอมป์ (ขึ้นอยู่แต่ละรุ่น)

5) ทำการโหลดเอาแฟ้มข้อมูลประเภทไบนารี่

5.1) ไม่จำกัดจำนวนครั้งในการโปรแกรมซ้ำ

5.2) มีโหมดในการโปรแกรมให้เลือก 6 โหมด

6) มีโปรแกรมช่วยพัฒนาได้แก่ XACT Development System ที่ทำงานบนคอมพิวเตอร์รุ่นต่าง ๆ เช่น 486/Pentiums, NEC PC, Apollo, Sun-4, HP700

6.1) สามารถติดต่อกับโปรแกรมอื่นๆ ได้ เช่น Viewlogic, Mentor Graphic และ OrCAD เป็นต้น

6.2) มีโปรแกรมการวางและเชื่อมโยงอุปกรณ์ภายในแบบอัตโนมัติ (Automatic Place And routing) ที่ครบสมบูรณ์

6.3) มี Interactive Design Editor ที่ใช้สำหรับการทำ Optimization

6.4) มี 288 มาโคร 34 ฮาร์ดมาโคร และ แรม/รอมคอมพายเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 รายละเอียดของอุปกรณ์ภายในเอฟพีจีเอตระกูล XC4000

Device	MAX Logic gate (No RAM)	MAX RAM Bits (No Logic)	Typical Gate Range (logic and RAM)*	CLB Matrix	Total logic Blocks	Number of Flip-Flops	Max Decode Input per side	Max User I/O
XC4003E	3,000	3,200	2,000-5,000	10x10	100	360	30	80
XC4005E/L	5,000	6,272	3,000-9,000	14x14	196	616	42	112
XC4006E	6,000	8,192	4,000-12,000	16x16	256	768	48	128
XC4008E	8,000	10,368	6,000-15,000	18x18	324	936	54	144
XC4010E/L	10,000	12,800	7,000-20,000	20x20	400	1,120	60	160
XC4013E/L	13,000	18,432	10,000-30,000	24x24	576	1,536	72	192
XC4020E	20,000	25,088	13,000-40,000	28x28	784	2,016	84	224
XC4025E	25,000	32,768	15,000-45,000	43x32	1,024	2,560	96	256
XC4028EX/XL	28,000	32,768	18,000-50,000	32X32	1,024	2,560	96	256
XC4036EX/XL	36,000	41,472	22,000-65,000	36x36	1,296	3,168	108	288
XC4044EX/XL	44,000	51,200	27,000-80,000	40x40	1,600	3,840	120	320
XC4052XL	52,000	61,952	33,000-100,000	44x44	1,936	4,576	132	352
XC4062XL	62,000	73,728	40,000-130,000	48x48	2,304	5,376	144	384
Larger Devices Available in the First Half of 1997								

*Max value of Typical Gate Range include 20 – 30% of CLBs Used as RAM

2.7 การโปรแกรมเอฟพีจีเอตระกูล XC4000

คือกระบวนการในการโหลดข้อมูลในโปรแกรมไปยังแอลซีเอ เพื่อกำหนดหน้าที่ของการทำงานในแต่ละบล็อกภายใน และการเชื่อมต่อ ซึ่งเอฟพีจีเอตระกูล XC4000 จะต้องใช้ข้อมูลเกี่ยวกับการโปรแกรมประมาณ 350 บิตต่อซีแอลบี โดยแต่ละบิตจะบอกถึงสถานะของหน่วยความจำสเตติกที่ควบคุมบิตในการควบคุมตารางฟังก์ชั่น (function table bit) มัลติเพล็กซ์อินพุตหรือการเชื่อมต่อกันระหว่างทรานซิสเตอร์

2.7.1 โหมดการโปรแกรม

เอฟพีจีเอในตระกูล XC4000 มีโหมดการโปรแกรม 6 โหมด ซึ่งแต่ละโหมดจะถูกกำหนดโดยจากบิตโหมด ได้แก่ บิต M0, M1 และ M2 โดยมีโหมดการโปรแกรมดังตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

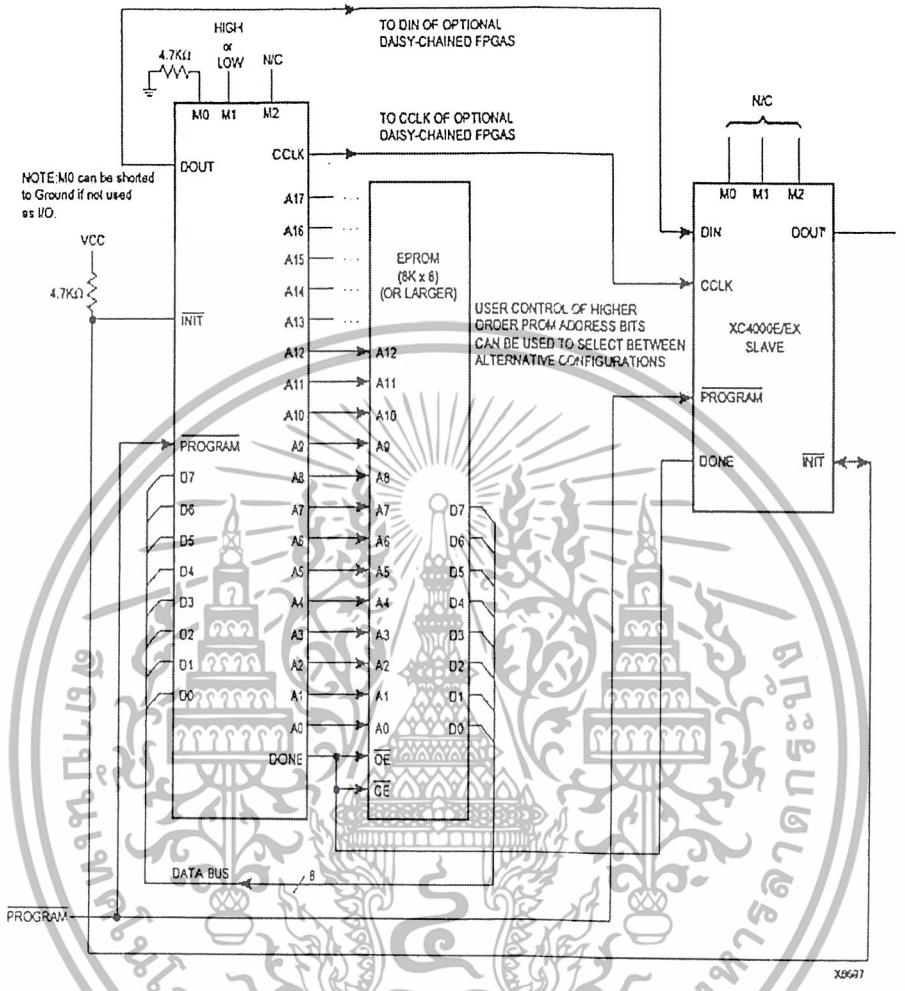
ตารางที่ 2.5 รูปแบบของโหมดต่าง ๆ ในการโปรแกรมเอฟพีจีเอตระกูล XC4000

Mode	M2	M1	M0	Clock	Data
Master Serial	0	0	0	Output	Bit-serial
Slave Serial	1	1	1	Input	Bit-Serial
Master parallel up	1	0	0	Output	Byte-Wide,00000
Master parallel down	1	1	0	Output	Byte-Wide,3FFFF
Peripheral Synch.	0	1	1	Input	Byte-Wide
Peripheral Asynch.	1	0	1	Output	Byte-Wide

1) โหมดหลัก (Master Modes)

ในการทำงานในโหมดนี้ตัวแอลซีเอจะถูกโหลดข้อมูลโครงแบบ (Configuration Data) จากหน่วยความจำภายนอกเข้ามาโดยอัตโนมัติ มีโหมดที่แตกต่างกัน 8 โหมดโดยใช้ ช่วงเวลาภายในจ่ายให้ซีคล็อก (CCLK : Configuration Clock) เพื่อที่จะเป็นฐานเวลาในการนำข้อมูลที่เข้ามาทางโหมดหลักแบบอนุกรม (Serial Master Mode) และรับข้อมูลโครงแบบเข้ามาทางขาสัญญาณดีอิน (DIN : Data In) จากแหล่งสัญญาณซิงโครนัส เช่น Xilinx Serial Configuration PROM, Parallel Master Low Master High Mode โดยจะรับข้อมูลแบบขนานมาจากบิต D0-D7 โดยสัมพันธ์กับแอดเดรสที่กำหนดโดยแอลซีเอดังรูปที่ 2.22 จะเป็นการแสดงตัวอย่างการเชื่อมต่อในโหลดหลักแบบขนาน (Master Parallel Mode) โดยเริ่มต้นที่แอดเดรส 0000 ไบต์ข้อมูล (Data Byte) จะถูกอ่านเข้ามาแบบขนานทุก ๆ สัญญาณอาร์คล็อก (RCLK : Read Clock) และส่งเข้าไปภายในแบบอนุกรมโดยอาศัยสัญญาณนาฬิกาโครงแบบ (Configuration Clock)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



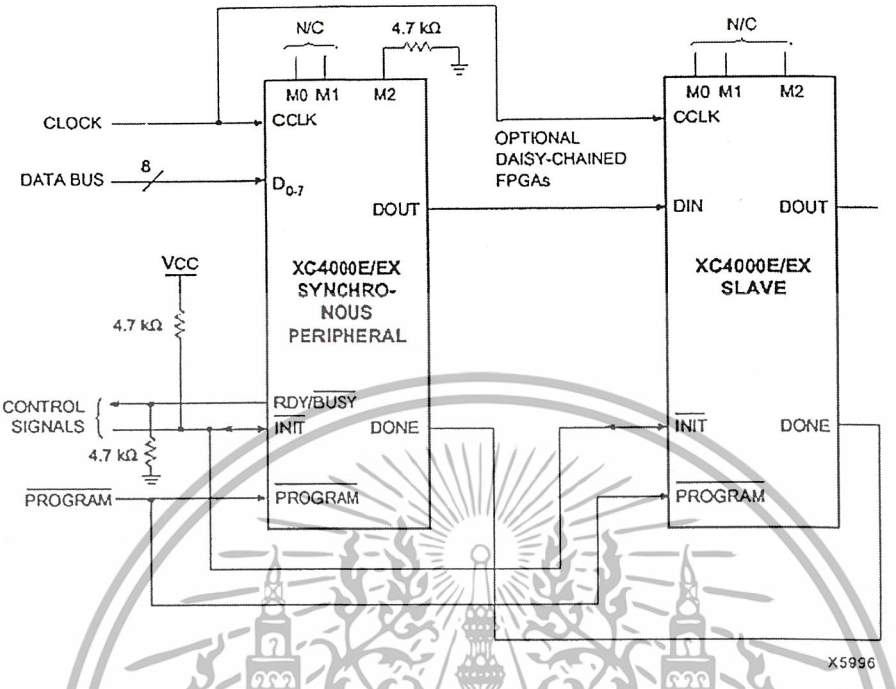
รูปที่ 2.22 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดหลักแบบขนาน

2) เพอริเฟอรัลโหมด (Peripheral Modes)

โหมดนี้จะรับข้อมูลเป็นไบต์จากบัสสถานะพร้อม (Ready) / กำลังทำงาน (Busy) จะมีอยู่เพื่อทำให้สัญญาณต่าง ๆ สื่อสารกันได้อย่างมีประสิทธิภาพมากขึ้น

ในอะซิงโครนัสโหมดนั้นออสซิสเลเตอร์ภายในจะสร้างสัญญาณ CCLK เพื่อรับข้อมูลที่เป็นไบต์ให้อยู่ในรูปแบบที่เหมาะสม และในซิงโครนัสโหมดนั้นสัญญาณนาฬิกาภายนอกจะเป็นตัวควบคุมการรับข้อมูลที่เป็นไบต์ดังรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

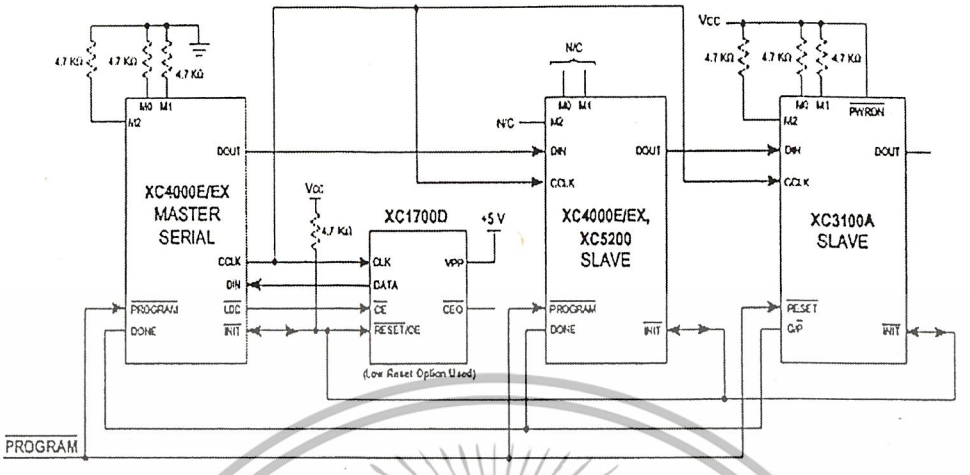


รูปที่ 2.23 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดเพอริเฟอรัลแบบอะซิงโครนัส (Asynchronous Peripheral mode)

3) โหมดรองแบบอนุกรม (Slave Serial Modes)

ในโหมดนี้แอลซีเอจะรับข้อมูลโครงแบบที่เป็นอนุกรมในขาขึ้นของสัญญาณ CCLK และหลังจากรับข้อมูลมาแล้วจะส่งข้อมูลเพิ่มเติมออกไปด้วยและแอลซีเอก็จะถูกทำการซิงโครไนซ์ในขาลงถัดไปของสัญญาณ CCLK ดังรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดรองแบบอนุกรม

2.8 การใช้ความสามารถของแรมในเอฟพีจีเอตระกูล XC4000

แอลซีเอทำงานโดยใช้ตารางการค้นหา (Look-up table) ซึ่งจะทำการเก็บตารางที่ว่านี้ในสแตติกแรม ซึ่งจะถูกเขียนในระหว่างการโปรแกรมโครงแบบลงบนแอลซีเอจะถูกอ่านในการโอเปอร์เรชั่น ดังนั้นแรมภายในจึงควรรวมไว้ในการออกแบบของผู้ใช้ด้วย

หน้าที่ของแรมในเอฟพีจีเอตระกูล XC4000 มีหน้าที่คล้ายแรมโดยทั่ว ๆ ไป เช่น เอฟไอเอฟโอ (FIFO : First In First Out) แอลไอเอฟโอ (LIFO : Last In First Out) รีจิสเตอร์ไฟล์รวมทั้งแอปพลิเคชั่นบางอย่าง เช่น รีจิสเตอร์เลื่อนข้อมูล (Shift register) แรมของเอฟพีจีเอตระกูล XC4000 มีความเร็วสูงเหมือนเอสแรม (SRAM) จึงไม่จำเป็นต้องคำนึงถึงเวลาหน่วงของการเชื่อมต่อ (Interconnection Delay)

ตารางที่ 2.6 จำนวนของแรมภายในเอฟพีจีเอตระกูล XC4000

RAM Module	Equivalent Logic	XC4003	XC4005	XC4010
16x1	4 — input Function Generator (F or G)	200	392	800
32x1	Two — 4 — input Function Generators and One 3 — input Function Generators (F+G+H)	100	196	400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010

Pad Name	Pin Number	Pad Name	Pin Number
GND	P1	I/O,TDI	P15
VCC	P2	I/O,TCK	P16
I/O(A8)	P3	I/O,TMS	P17
I/O(A9)	P4	I/O	P18
I/O(A10)	P5	I/O	P19
I/O(A11)	P6	I/O	P20
I/O(A12)	P7	GND	P21
I/O(A13)	P8	VCC	P22
I/O(A14)	P9	I/O	P23
I/O,SGCK1(A15)	P10	I/O	P24
I/O,PGCK1(A16)	P13	I/O	P27
I/O(A17)	P14	I/O	P28
I/O,SCGK2	P29	I/O,PGCK3	P57
O(M1)	P30	I/O(D6)	P58
GND	P31	I/O(D5)	P59
I(M0)	P32	I/O($\overline{CS0}$)	P60
VCC	P33	I/O(D4)	P61
I(M2)	P34	I/O	P62
I/O,PGCK2	P35	VCC	P63
I/O($\overline{HDC}$)	P36	GND	P64
I/O(LDC)	P37	I/O(D3)	P65
I/O	P38	I/O(RS)	P66
I/O	P39	I/O(D2)	P67
I/O	P40	I/O	P68
I/O($\overline{INIT}$)	P41	I/O(D1)	P69
VCC	P42	I/O($\overline{RCLK}$,RDY/$\overline{BUSY}$)	P70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 (ต่อ)รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010

Pad Name	Pin Number	Pad Name	Pin Number
GND	P43	I/O(D0,DIN)	P71
I/O	P44	I/O,SGLK4(DOUT)	P72
I/O	P45	CCLK	P73
I/O	P46	VCC	P74
I/O	P47	O,TDO	P75
I/O	P48	GND	P76
I/O	P49	I/O(A0,WS)	P77
I/O	P50	I/O,PGCK4(A1)	P78
I/O,SGCK3	P51	I/O(CS,A2)	P79
GND	P52	I/O(A3)	P80
DONE	P53	I/O(A4)	P81
VCC	P54	I/O(A5)	P82
$\overline{\text{PROGRAM}}$	P55	I/O(A6)	P83
I/O (D7)	P56	I/O(A7)	P84

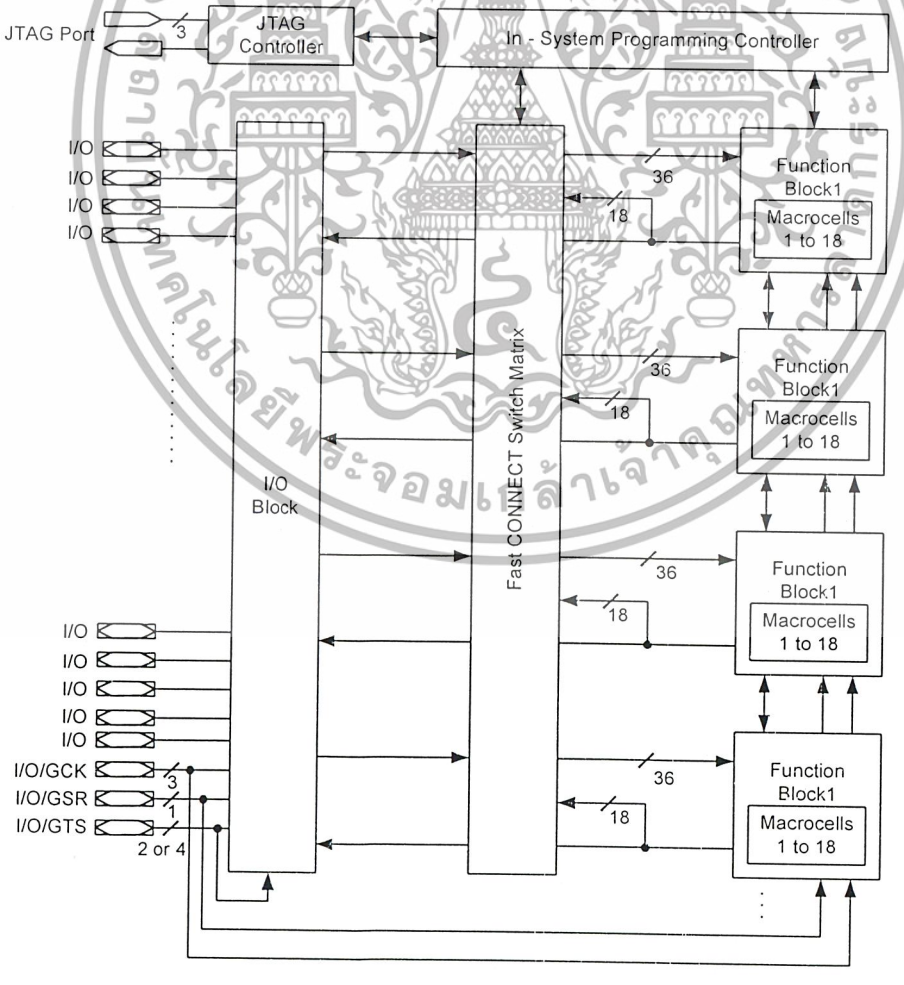
2.9 ซีพีแอลดี (CPLD : Complex Programmable Logic Device)

ในอดีตการออกแบบวงจรดิจิตอลจะต้องอาศัยการออกแบบและทดลองต่อวงจรกับอุปกรณ์ประเภทไอซีเกตซึ่งเป็นตัวไอซีนำมาต่อวงจรเข้าด้วยกัน การต่อวงจรไอซีด้วยไอซีหลายๆ ตัว ผู้ออกแบบต้องมีการสำรองและจัดเก็บไอซีไว้ให้เลือกใช้ทดลองต่อวงจรเป็นจำนวนมาก และต้องทำการต่อวงจรไอซีทีละตัวเข้าด้วยกันบนบอร์ดทดลอง ทำให้การออกแบบเกิดความล่าช้า ยากต่อการตรวจสอบและแก้ไขวงจรในกรณีที่มีข้อผิดพลาดเกิดขึ้น ต้นทุนในการออกแบบวงจรสูงมาก เป็นต้น ในปัจจุบันการออกแบบวงจรดิจิตอลสามารถออกแบบวงจรดิจิตอลหลายๆ วงจร หรือวงจรที่มีขนาดใหญ่ด้วยเครื่องคอมพิวเตอร์หลังจากการออกแบบสามารถทำการจำลองการทำงานของวงจรที่ออกแบบว่าวงจรนั้นทำงานได้ถูกต้องหรือไม่ เมื่อวงจรที่ออกแบบทำงานได้ถูกต้องและตรงตามความต้องการของผู้ออกแบบ สามารถทำการดาวน์โหลดวงจรเหล่านั้นเก็บไว้ในไอซีเพียงตัวเดียวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

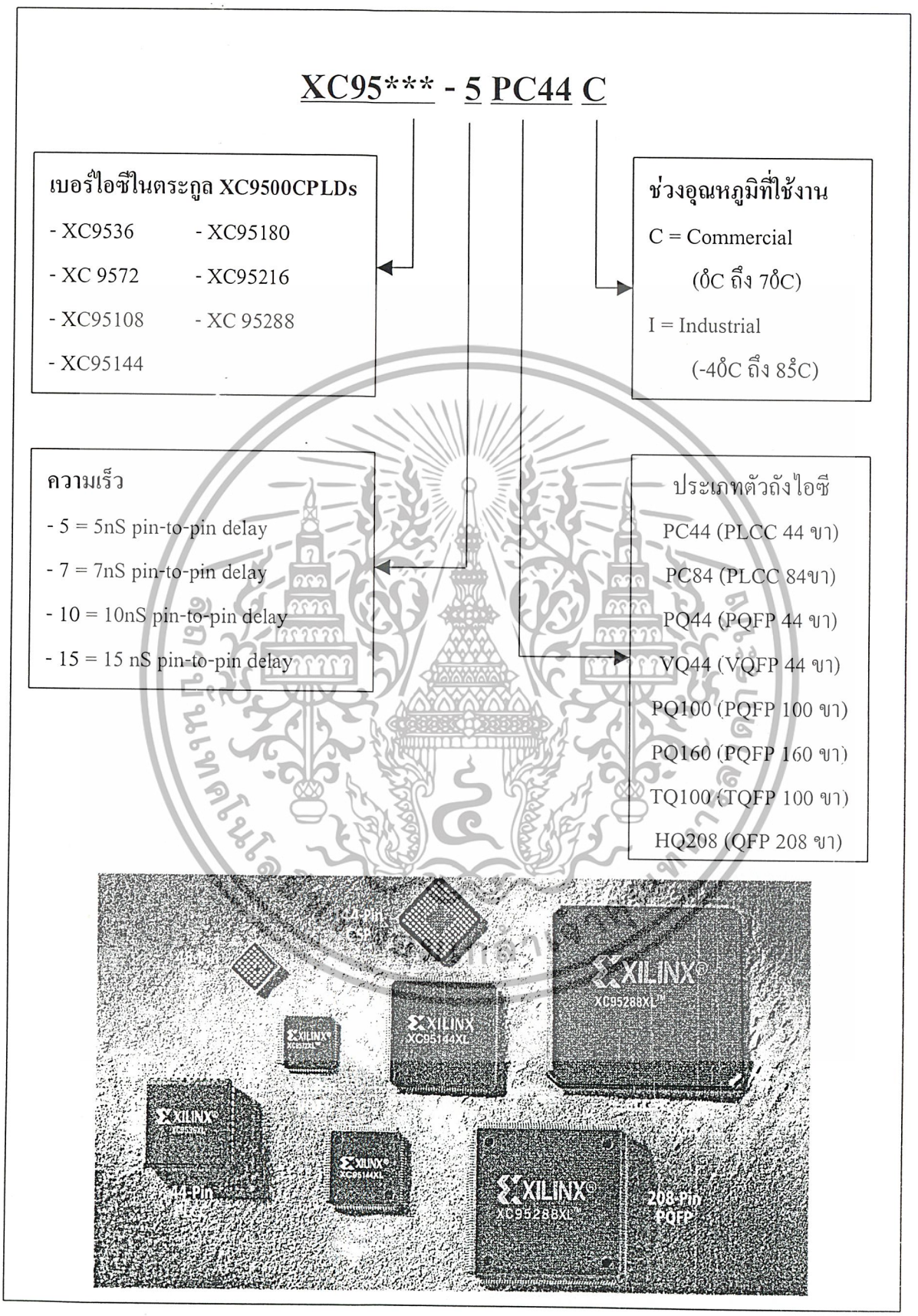
การออกแบบวงจรดิจิตอลให้อยู่ภายในไอซีเพียงตัวเดียว โดยใช้ไอซีในตระกูล XC9500 CPLDs (XC9500 Complex Programmable Logic Device) เป็นตระกูลไอซีที่อยู่ในกลุ่มที่ผู้ใช้สามารถออกแบบวงจรรวมดิจิตอล หรือกำหนดการทำงานรวมของวงจรรวมดิจิตอลภายในไอซีได้ โดยใช้ระบบซอฟต์แวร์ที่พัฒนาขึ้นมาโดยบริษัท Xilinx

ภายในตระกูล XC9500 CPLDs มีเบอร์ไอซีให้สามารถเลือกใช้งานได้หลายเบอร์ขึ้นอยู่กับจำนวนมาโครเซลภายในไอซี ถ้าจำนวนมาโครเซลภายในไอซีมีจำนวนมากจะทำให้สามารถออกแบบวงจรที่มีจำนวนลอจิกเกตภายในได้มากขึ้นเช่นกัน การดาวน์โหลดวงจรลงในไอซีสามารถทำการดาวน์โหลดวงจรผ่านทางพอร์ตขนานของเครื่องคอมพิวเตอร์ในลักษณะ ISP (In System Programming) ตรงตามมาตรฐาน IEEE 1149.1 BOUNDARY Scan (JTAG) ซึ่งจะสามารถดาวน์โหลดวงจรลงในไอซีได้โดยไม่ต้องถอดไอซีออกจากวงจร การดาวน์โหลดวงจรหรือกระบวนการอ่านหรือเขียนวงจรลงในไอซีสามารถทำได้ประมาณ 10,000 ครั้ง



รูปที่ 2.25 โครงสร้างของ XC9500 ซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 การพิจารณาความหมายต่าง ๆ ที่ด้านบนของตัวถังไอซีในตระกูล XC9500 ซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาที่ด้านบนของไอซี จะปรากฏข้อความบอกข้อมูลต่าง ๆ ที่เกี่ยวข้องกับไอซี เมื่อพิจารณาแล้วสามารถตีความหมายได้ดังนี้

1) สมมติเป็นไอซี XC9572

2) ตัวถังเป็นประมาณ PC44 (PLCC44 มีทั้งหมด 44 ขา)

3) มีค่าเวลาหน่วงระหว่างขาถึงขา (Pin-to-pin delay) 15 nS

4) ช่วงอุณหภูมิที่ใช้งานพิจารณาจากตัวอักษร C หลังเลข 15 บนไอซี (C = Commercial) จะใช้งานในช่วงอุณหภูมิ 0° C ถึง 70° C

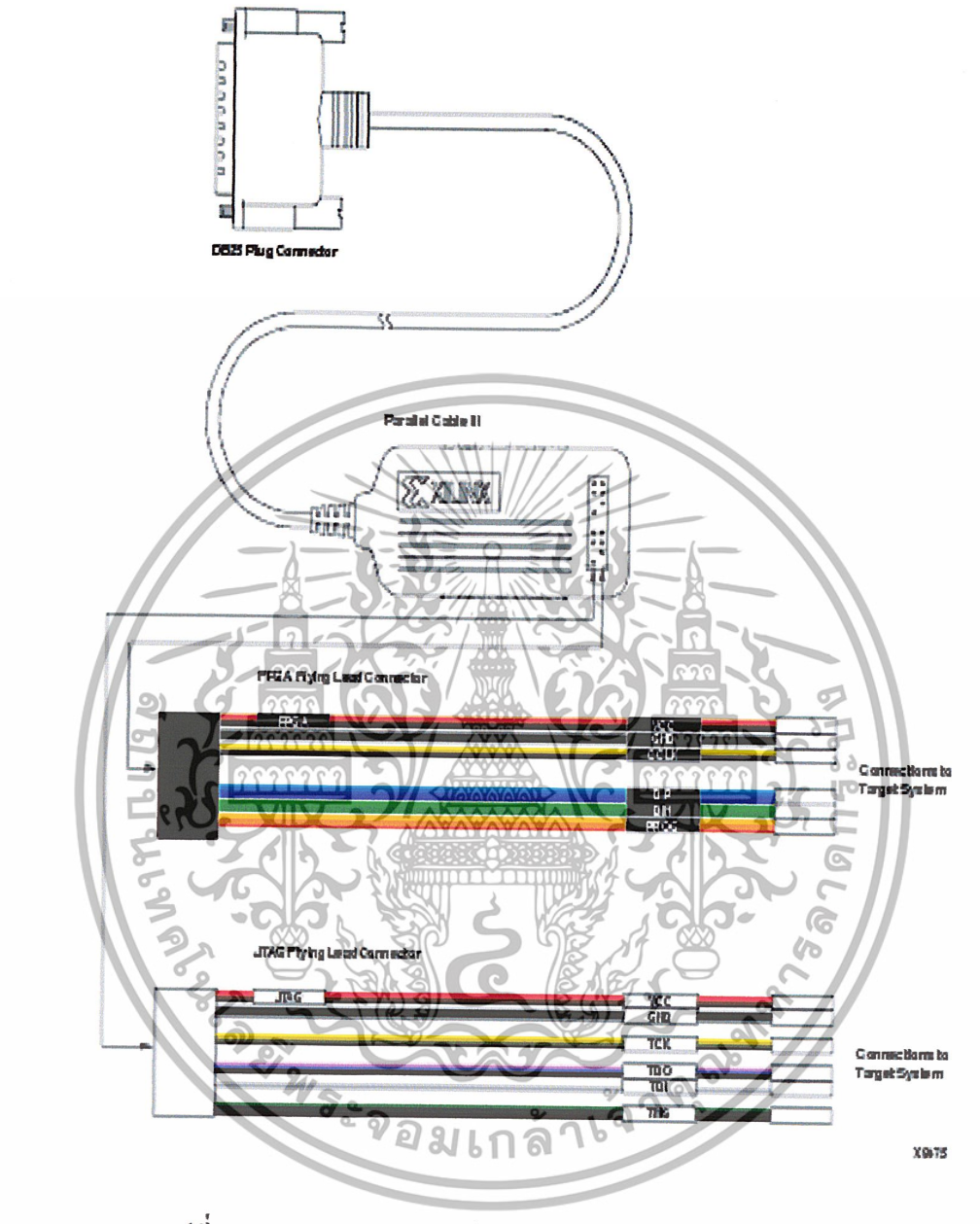
ตารางที่ 2.8 คุณสมบัติของไอซีตระกูล XC9500XL/XV

	XC9536XL/XV	XC9572XL/XV	XC95144XL/XV	XC95288XL/XV
Macrocells	36	72	144	288
Usable Gates	800	1600	3200	6400
Registers	36	72	144	288
t_{pD} (ns)				
XC9500XL	5	5	5	6
XC9500XV	3.5	4	4	5
f_{system} (MHz)				
XC9500XL	222	222	222	208
XC9500XV	278	250	250	222
Packages	PC44 (34)	PC44 (34)	TQ100 (81)	TQ144 (117)
Max. User I/O	VQ44 (34)	VQ44 (34)	CS144 (117)	PQ208 (168)
	VQ64 (34)	VQ64 (34)	TQ144 (117)	FG256 (192)
	CS48 (36)	CS48 (38)		BG256(192)
		TQ 100 (72)		CS280(192)

2.10 อุปกรณ์การดาวน์โหลด Xchecker Cable

เมื่อผ่านขั้นตอนต่าง ๆ จากการ Flow Enging แล้วผู้ใช้สามารถนำวงจรที่ได้สร้างขึ้นสำหรับ FPGA และผ่านการทดสอบด้วยซอฟต์แวร์แล้วมาทดสอบกับฮาร์ดแวร์จริงได้โดยการ Download ผ่านสาย Download ดังรูปที่ 2.27 ซึ่งจะใช้โปรแกรม Hardware Debugger โดยคลิกปุ่ม Programming ในโปรแกรม Design Manager เพื่อ Download ข้อมูลลงในแผงทดลองเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 สาย X-Checker พร้อมแถบสายต่อไปบอร์ดทดลอง

ส่วนประกอบของสาย X-Checker ดังรูปที่ 2.28 จะประกอบด้วยขาต่าง ๆ เพื่อใช้ต่อกับแถบสายต่อไปยังบอร์ดทดลอง มีดังนี้

1) VCC = ไฟเลี้ยงเอฟพีจีเอ 5 โวลท์ , 3.3 โวลท์ หรือ 2.5 โวลท์

2) GNG = กราวด์ของบอร์ดเอฟพีจีเอ

3) CCLK = ต่อกับขา CCLK ของเอฟพีจีเอ

4) D/P = ต่อกับขา DONE ของเอฟพีจีเอ

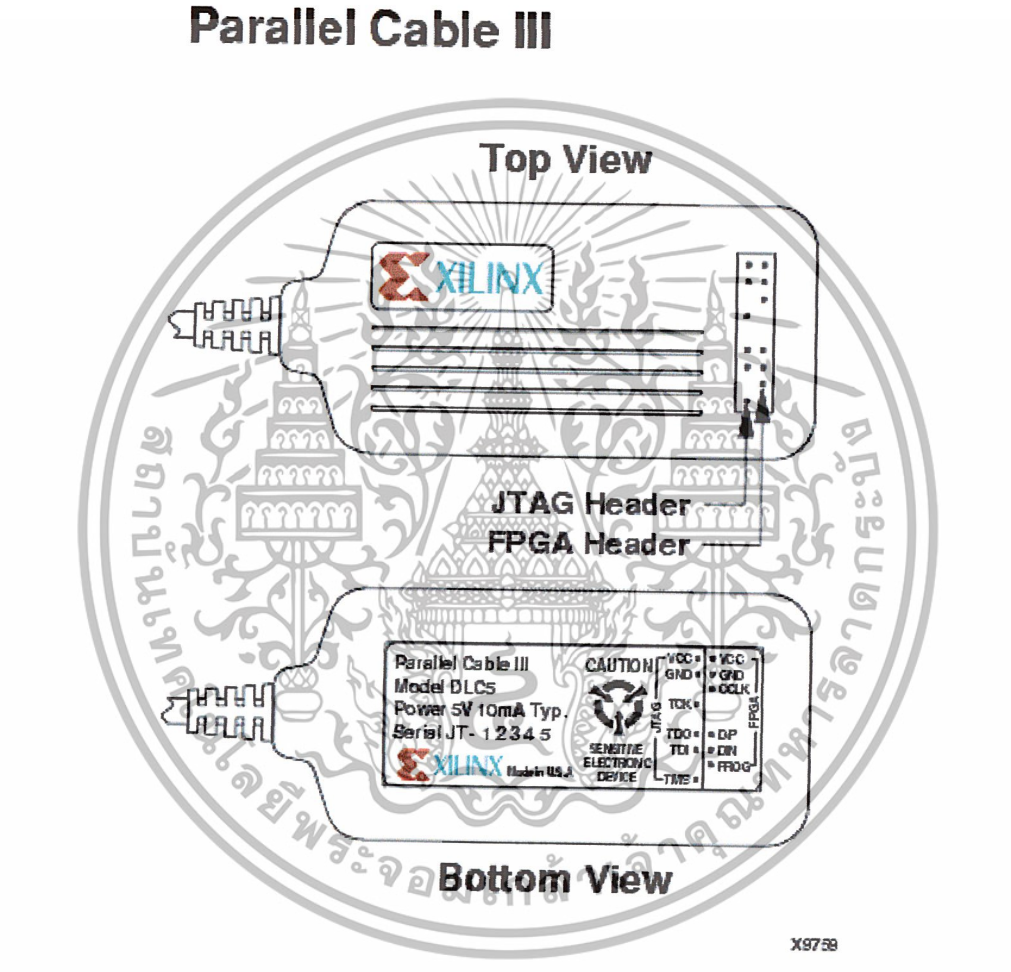
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) DIN = ต่อกับขา DIN ของเอฟพีจีเอ

6) PROG = ต่อกับขา PROGRAM ของเอฟพีจีเอ

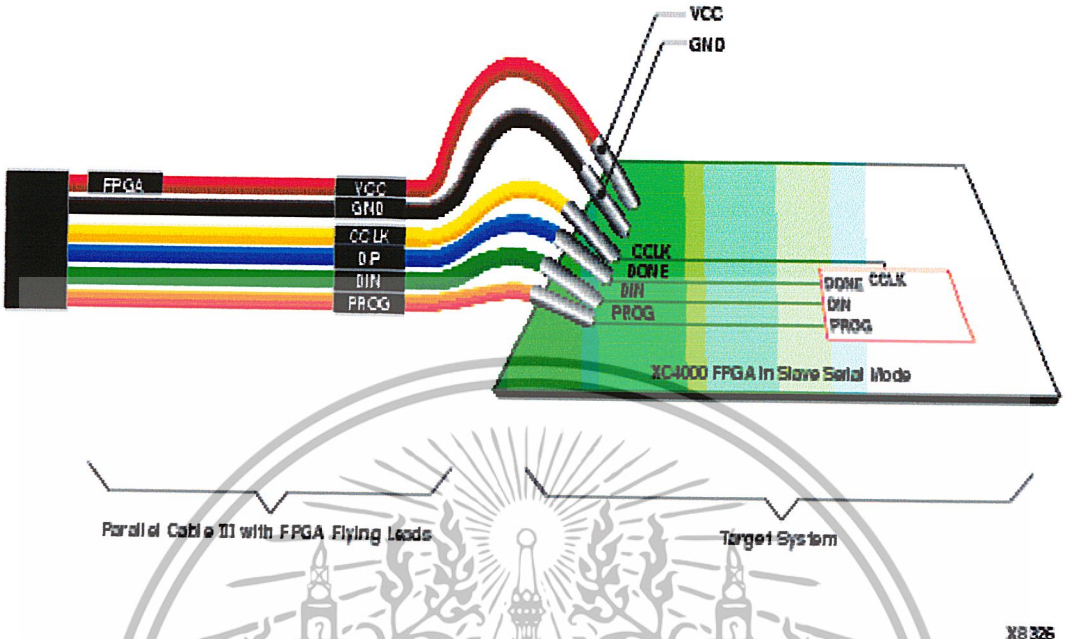
7) INIT = ต่อกับขา INIT ของเอฟพีจีเอ

8) RST = ไม่ได้ใช้

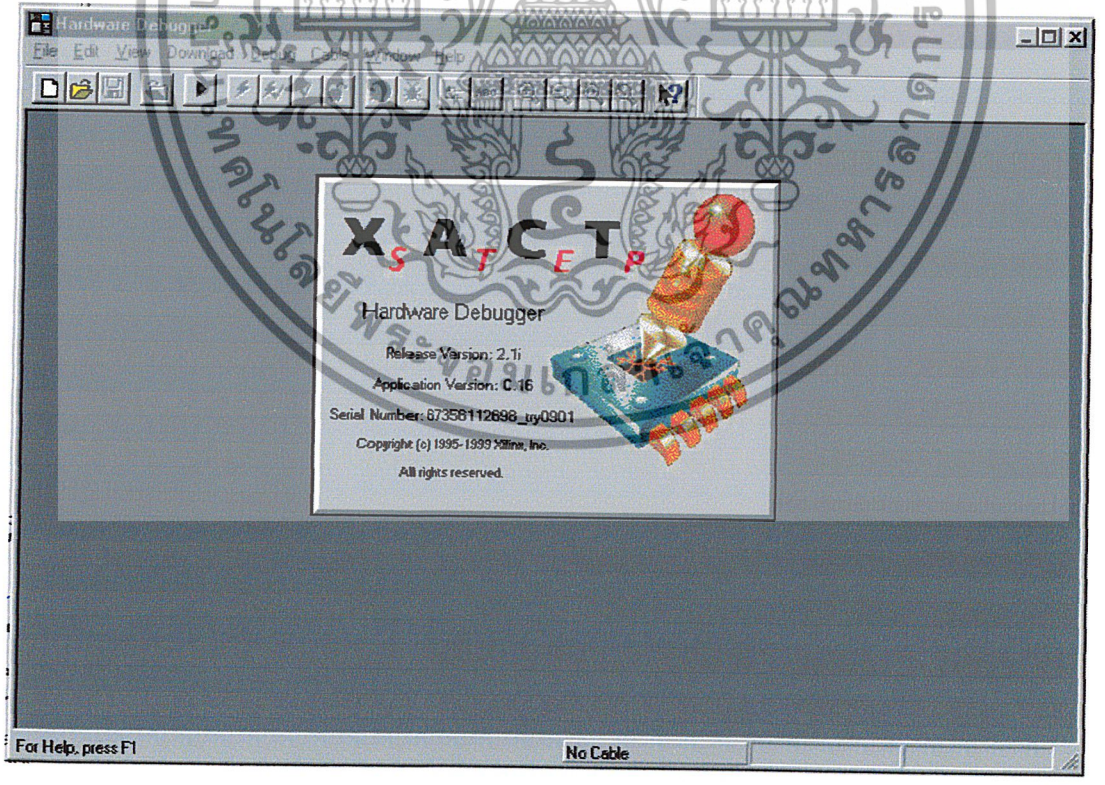


รูปที่ 2.28 ส่วนหัวของสาย X-Checker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



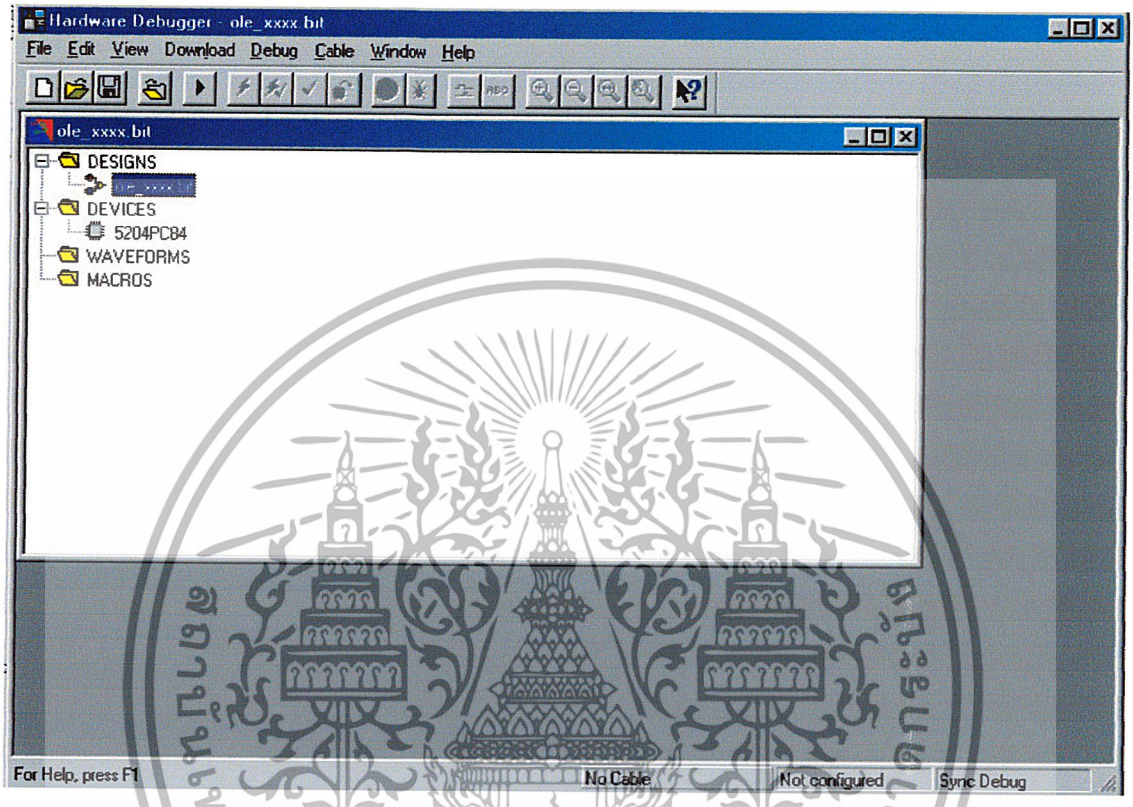
รูปที่ 2.29 การเชื่อมต่อระหว่าง X – Checker กับแผงทดลอง



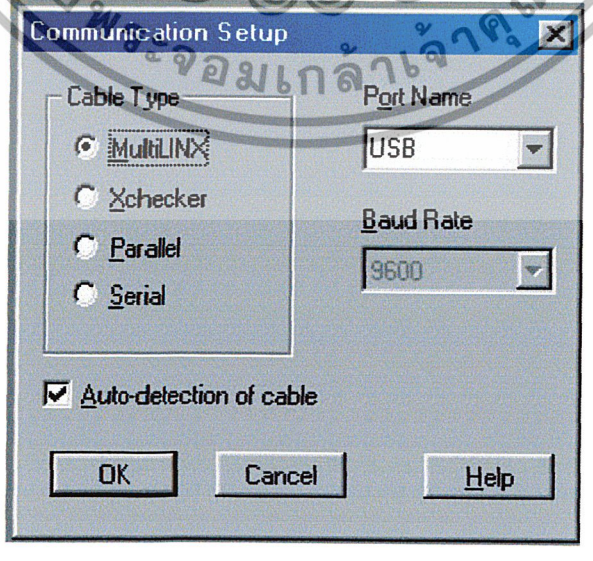
รูปที่ 2.30 หน้าต่างโปรแกรม Hardware Debugger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกเมนู Cable > Communications เพื่อการ Setup การเชื่อมต่อของ X-Checker cable กับเครื่องคอมพิวเตอร์ที่ใช้



รูปที่ 2.31 หน้าต่างโปรแกรม Hardware Debugger ในการเปิด Design project



รูปที่ 2.32 หน้าต่าง Communication Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบการสร้าง และการทำงาน

3.1 โมดูลต่าง ๆ ของชุดทดลอง

ในชุดทดลองนี้ก็จะประกอบไปด้วยโมดูลต่าง ๆ โดยจะแบ่งออกเป็น 3 ส่วน คือ แผงวงจรเอฟพีจีเอ, แผงวงจรซีพีแอลดี และแผงวงจรอินพุต/เอาต์พุต

โดยในแผงวงจรอินพุต/เอาต์พุตก็จะมีส่วนของแผงวงจรย่อย ๆ ดังนี้

1) แสดงผลเจ็ดส่วน
2) ดิฟสวิตช์ 8 บิต
3) คีย์บอร์ดเมตริกซ์ 4 x 4
4) สเตปปิ้งมอเตอร์
5) แอลอีดี 40 ดวง
6) แปลงสัญญาณดิจิตอลเป็นแอนะลอก
7) แปลงสัญญาณแอนะลอกเป็นดิจิตอล
8) มอเตอร์ไฟฟ้ากระแสตรง
9) ลำโพง
10) แรม 128 กิโลไบต์
11) อีพรอม 256 กิโลไบต์
12) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์, 9 โวลต์ และ 12 โวลต์จ่ายไฟเลี้ยงแผงวงจรย่อย
13) สวิตช์กดติดปล่อยดับ 16 ตัว
14) ชุดทดลอง สี่แยกไฟแดง
15) วงจรกำเนิดสัญญาณนาฬิกา
16) MCS 8051

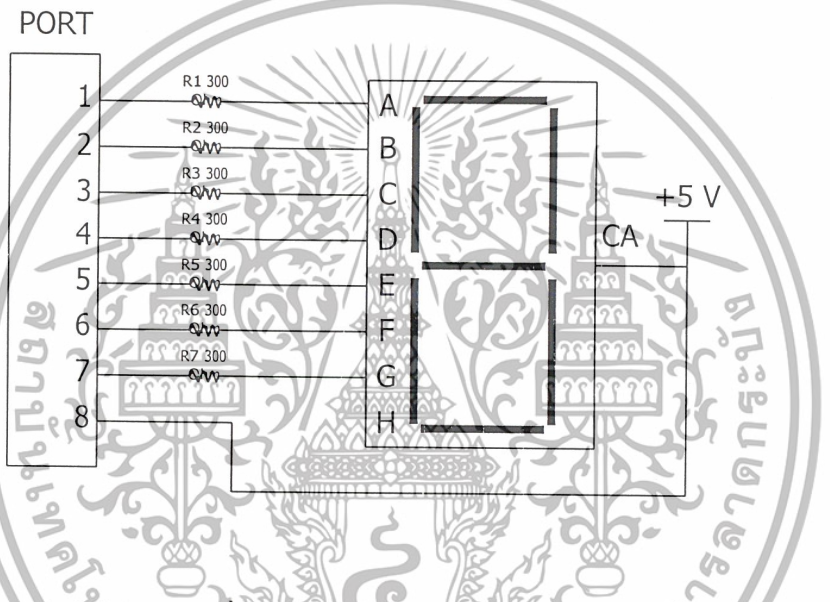
3.2 วงจรและการทำงานโดยสังเขป

ในโครงงานนี้ได้แบ่งโมดูลต่าง ๆ ออกเป็นแต่ละส่วนซึ่งจะได้อธิบายการทำงานของแต่ละโมดูลโดยสังเขปดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 โมดูลแสดงผลเจ็ดส่วน

โมดูลแสดงผลเจ็ดส่วนดังแสดงในรูปที่ 3.1 ประกอบด้วย ตัวแสดงผลเจ็ดส่วนทั้งหมด 8 หลัก โดยแต่ละหลักนั้นจะแยกออกจากกันโดยอิสระเพื่อสะดวกในการควบคุม และจะต่อขา A – G ออกมาเป็นพอร์ตใช้งาน โดยมีตัวต้านทานดร็อป (Drop) ไว้ทุกขาเพื่อจำกัดกระแสที่จะเข้าไปยังแสดงผลเจ็ดส่วนเนื่องจาก แสดงผลเจ็ดส่วนเป็นชนิด คอมม่อนเอโหนด จึงมีพอร์ตเพื่อต่อเป็นคอมม่อน ให้ 5 โวลต์

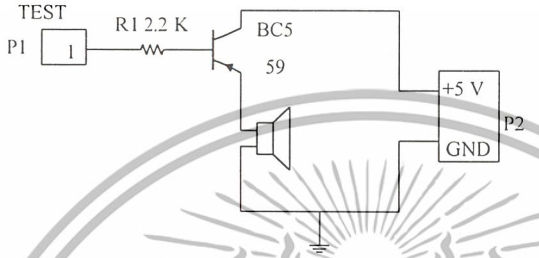


รูปที่ 3.1 วงจรแสดงผลเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โมดูลลำโพง

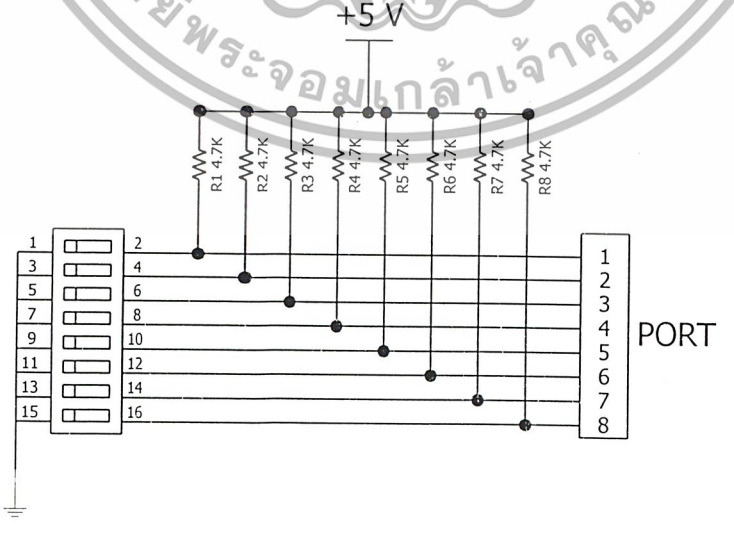
โมดูลลำโพงดังแสดงในรูปที่ 3.2 ประกอบด้วย ทรานซิสเตอร์ เบอร์ BC557 รีซิสเตอร์ 2.2 KΩ 1 ตัว ลำโพง 1 ตัว หลักการทำงานคือเมื่อมีการ Bias ที่ขา B ซึ่งจะจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ของทรานซิสเตอร์เพื่อเป็นการทริกซ์สัญญาณจะทำให้วงจร ขับให้ลำโพงดังขึ้นมา



รูปที่ 3.2 วงจรลำโพง

3.2.3 โมดูลดิฟสวิตช์ 8 บิต

โมดูลดิฟสวิตช์ 8 บิตดังแสดงในรูปที่ 3.3 จะประกอบไปด้วย ดิฟสวิตช์ 8 บิตจำนวน 2 ชุด โดยจะมีตัวต้านทาน 4.7 KΩ 16 ตัว และ Header 8 ตัว 2 ชุด ในโมดูลนี้จะเป็นวงจรที่จ่ายไฟให้ตลอดเวลาแต่จะควบคุมการเปิด-ปิด ด้วยการต่อกราวน์ด้วยดิฟสวิตช์



รูปที่ 3.3 วงจรดิฟสวิตช์ 8บิต2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โมดูลแอลอีดี 40 ดวง

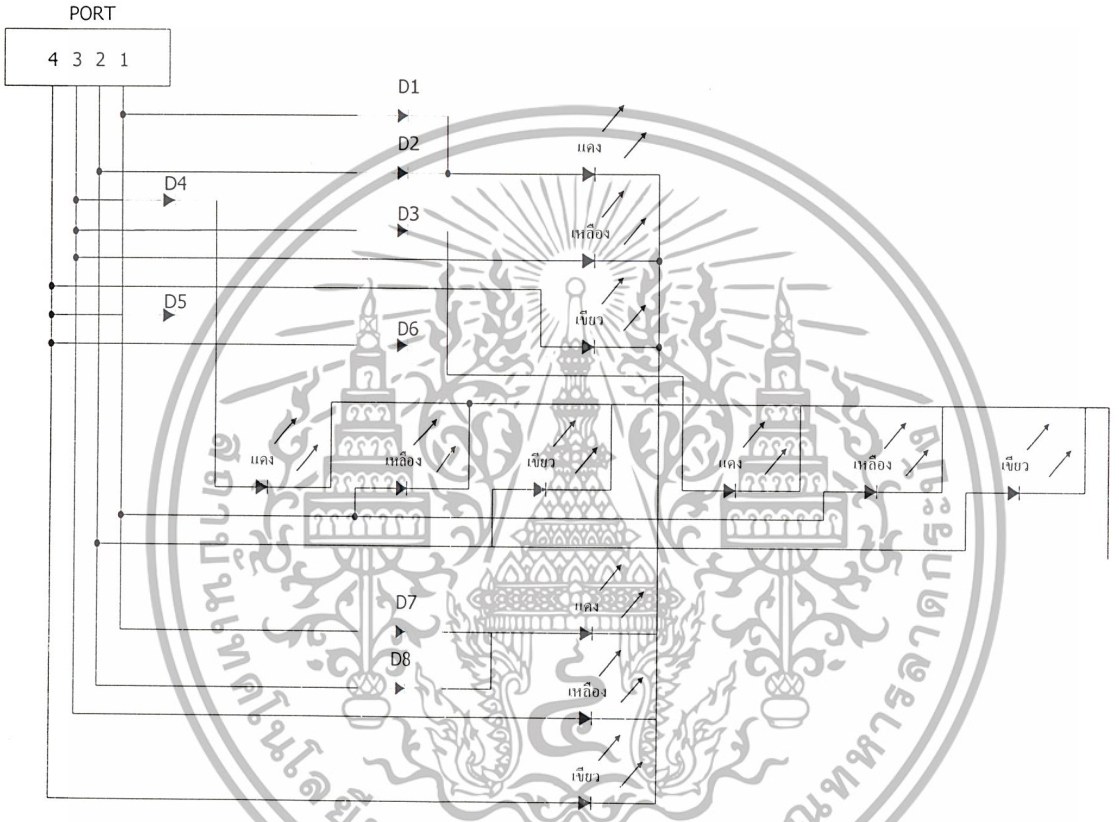
โมดูลแอลอีดี 40 ดวงดังแสดงในรูปที่ 3.4 ซึ่งจะประกอบด้วยวงจรดังกล่าวทั้งหมดจำนวน 5 ชุด และมีพอร์ตสำหรับต่อใช้งาน 5 พอร์ต คือ P1 – P5 พอร์ตละ 8 บิต ส่วนพอร์ตที่ 6 จะต่อเข้ากับกราวด์

รูปที่ 3.4 วงจรแอลอีดี 40 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โมดูลสี่แยกไฟแดง

โมดูลสี่แยกไฟแดงดังแสดงในรูปที่ 3.5 ประกอบด้วยแอลอีดี สีแดง 4 ดวง, แอลอีดี สีเหลือง 4 ดวง, แอลอีดี สีเขียว 4 ดวง ต่อเข้ากับ ไดโอด จำนวน 8 ตัว โดยวงจรนี้จะบังคับด้วยซอฟต์แวร์โดยจะป้อนรหัสควบคุมทั้งหมด 4 รหัส ผ่านพอร์ต ดังนี้ 0001, 0011, 0100, 1100

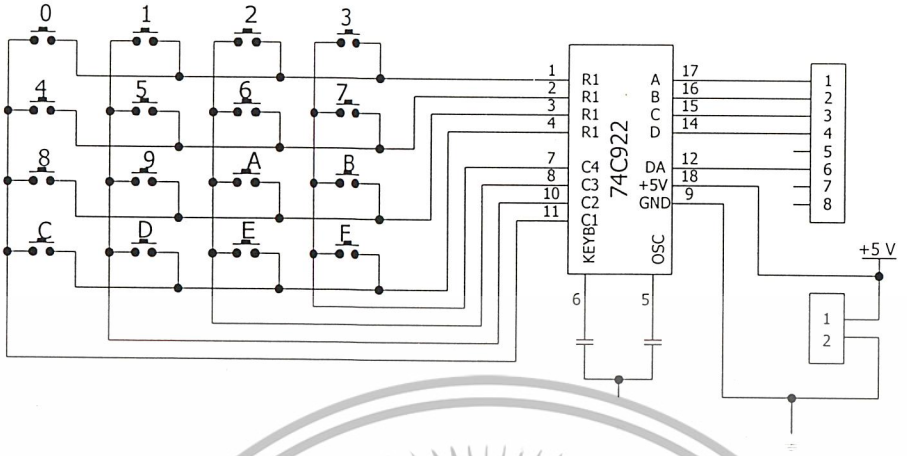


รูปที่ 3.5 วงจรสี่แยกไฟแดง

3.2.6 โมดูลคีย์บอร์ดเมตริกซ์4 x 4

โมดูลคีย์บอร์ดเมตริกซ์4 x 4 ดังแสดงในรูปที่ 3.6 เป็นวงจรที่จะทำหน้าที่ในการ ถอดรหัส โดยในแต่ละขาของ IC 74C922 ที่เป็นอินพุตนั้นจะทำหน้าที่รับสัญญาณอินพุตเข้ามาแล้ว ถอดรหัสส่งออกที่เอาต์พุตเป็นรหัสและยังทำหน้าที่เป็นตัว ค้างสถานะของสัญญาณไว้ด้วย สัญญาณที่ออกนั้นมี 4 บิต

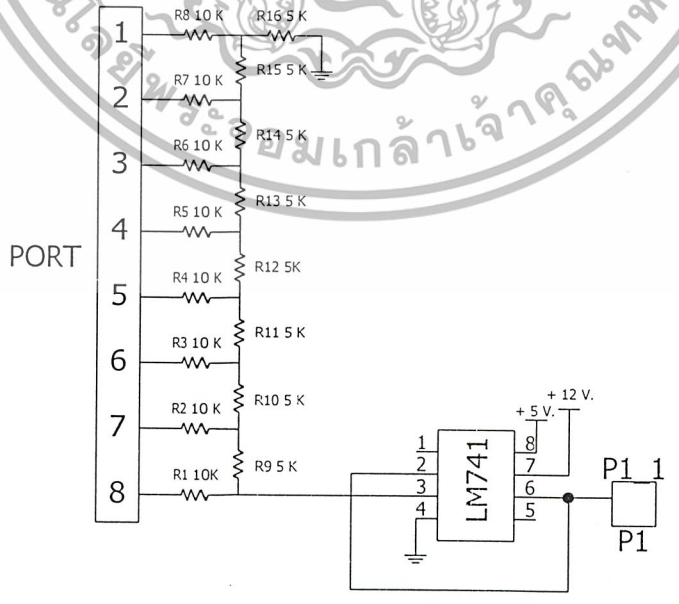
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรคีย์บอร์ดเมตริกซ์4 x 4

3.2.7 โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก

โมดูล แปลงสัญญาณดิจิตอลเป็นแอนะลอก ดังแสดงในรูปที่ 3.7 เป็นวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก โดยมีอินพุต 8 บิต และ เอาต์พุต 1 บิต ผ่านตัว IC LM741 เป็นตัวสร้างสัญญาณ โดยประมวลผลจากเอาต์พุต 8 บิต เข้าที่ขา 3 ของ IC และทำการประมวลผลเป็นระดับแรงดัน ส่งออกทางเอาต์พุตที่ขา 6 ซึ่งต่ออยู่กับขา 2 และส่งออกเอาต์พุต ตำแหน่งและหน้าที่การทำงานของขา IC LM741 จะแสดงในตารางที่ 3.1



รูปที่ 3.7 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

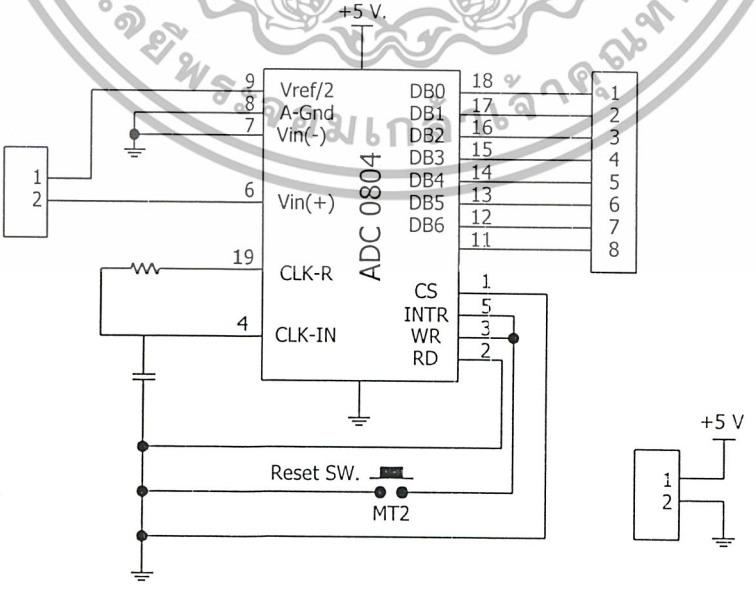
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ตำแหน่งขาของ IC LM741

ขาที่	คุณสมบัติ
1	No Use
2	Output
3	Input
4	GND
5	No Use
6	Output
7	Vcc (+12 โวลต์)
8	Vcc (+5 โวลต์)

3.2.8 โมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

โมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ดังแสดงในรูปที่ 3.8 เป็นวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล ด้วย IC ADC0804 เป็นตัวแปลง โดยจะรับสัญญาณอินพุตเข้ามา 1 ขา แล้วทำการประมวลผลแปลงเป็นสัญญาณดิจิตอล 8 บิต อีก 8 ขา ส่วนตำแหน่งการทำงานของขา IC ADC08040 จะแสดงอยู่ในตารางที่ 3.2



รูปที่ 3.8 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

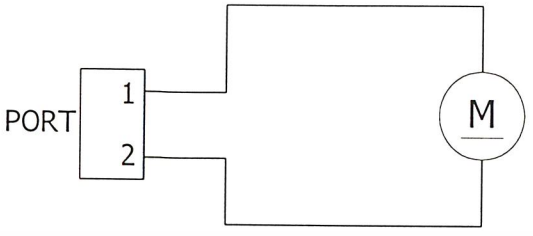
ตารางที่ 3.2 ตำแหน่งขาของ IC ADC0804

ขาที่	คุณสมบัติ
1	CS
2	RD
3	WR
4	CLK_IN
5	INTR
6	VIN(+)
7	GND
8	GND
9	Vref/2
10	GND
11	DB7
12	DB6
13	DB5
14	DB4
15	DB3
16	DB2
17	DB1
18	DB0

3.2.9 โมดูลมอเตอร์ไฟฟ้ากระแสตรง

โมดูลมอเตอร์ไฟฟ้ากระแสตรง ดังแสดงในรูปที่ 3.9 เป็นวงจรที่มีเพียงมอเตอร์ไฟฟ้ากระแสตรง 5โวลต์ 1 ตัว โดยจะมีพอร์ตสำหรับต่อเพื่อควบคุมการหมุนของมอเตอร์ โดยถ้าหากป้อนแบบ Forward มอเตอร์ก็จะหมุนไปทางขวา และถ้าหากป้อนแบบ Reverse มอเตอร์จะหมุนไปทางซ้าย

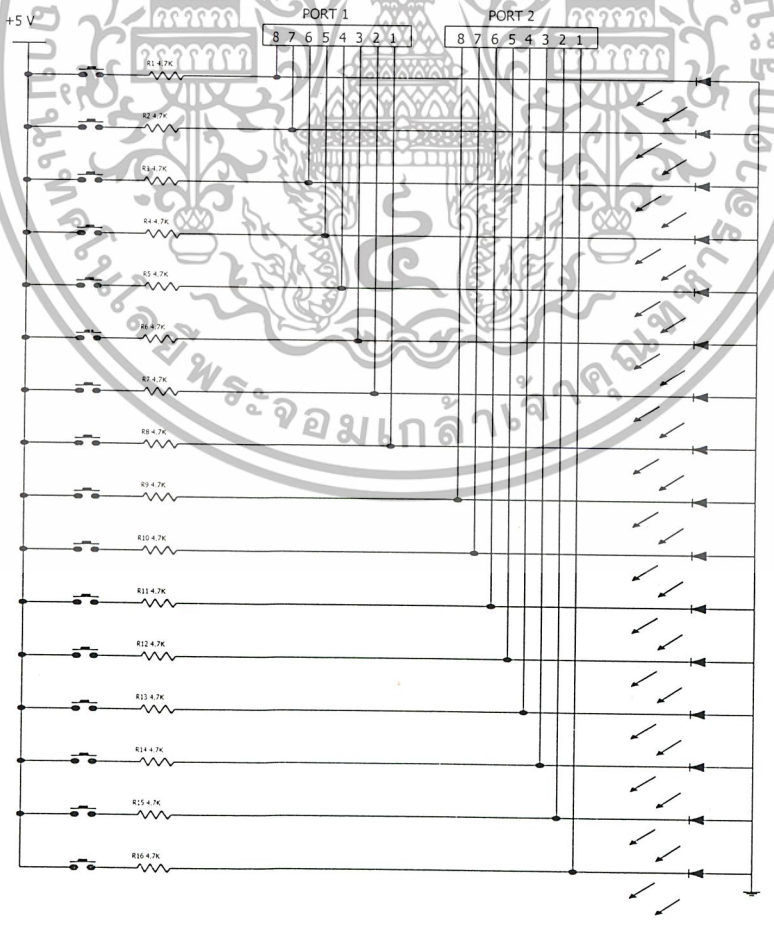
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรมอเตอร์ไฟฟ้ากระแสตรง

3.2.10 โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว

โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว ดังแสดงในรูปที่ 3.10 จะประกอบด้วย สวิตช์ที่เป็นลักษณะกดติดปล่อยดับ ทั้งหมด 16 ตัว โดยจะทำงานที่ "1" ซึ่งจะสามารถจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ได้ตลอดเวลาที่กดสวิตช์ และจะแสดงผลการทำงานด้วยแอลอีดี ในสวิตช์แต่ละดวง

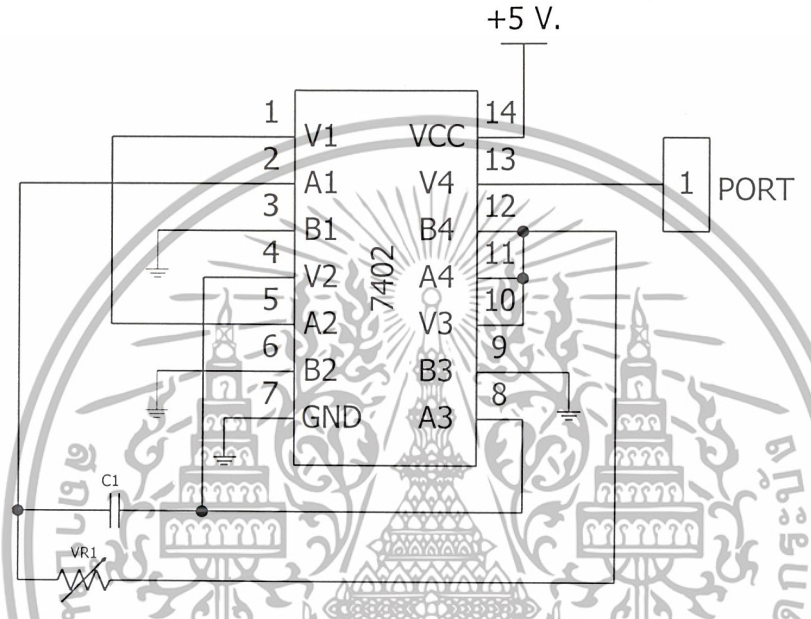


รูปที่ 3.10 วงจรสวิตช์กดติดปล่อยดับ 16 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.11 โมดูลวงจรกำเนิดสัญญาณนาฬิกา

โมดูลวงจรกำเนิดสัญญาณนาฬิกาดังแสดงในรูปที่ 3.11 เป็นวงจรกำเนิดสัญญาณนาฬิกา (Clock) โดยสามารถผลิตความถี่ได้ตั้งแต่ 1 – 1000 Hz ผลิตสัญญาณด้วย IC 7402 ปรับระดับความถี่ด้วย C1 และ VR1 ตามต้องการ ส่วนของตำแหน่งขา IC 7402 ดังแสดงในตารางที่ 3.3



รูปที่ 3.11 วงจรกำเนิดสัญญาณนาฬิกา

ตารางที่ 3.3 ตำแหน่งขาของ IC 7402

ขาที่	คุณสมบัติ
1	V1
2	A1
3	B1
4	V2
5	A2
6	B2
7	GND
8	A3
9	B3

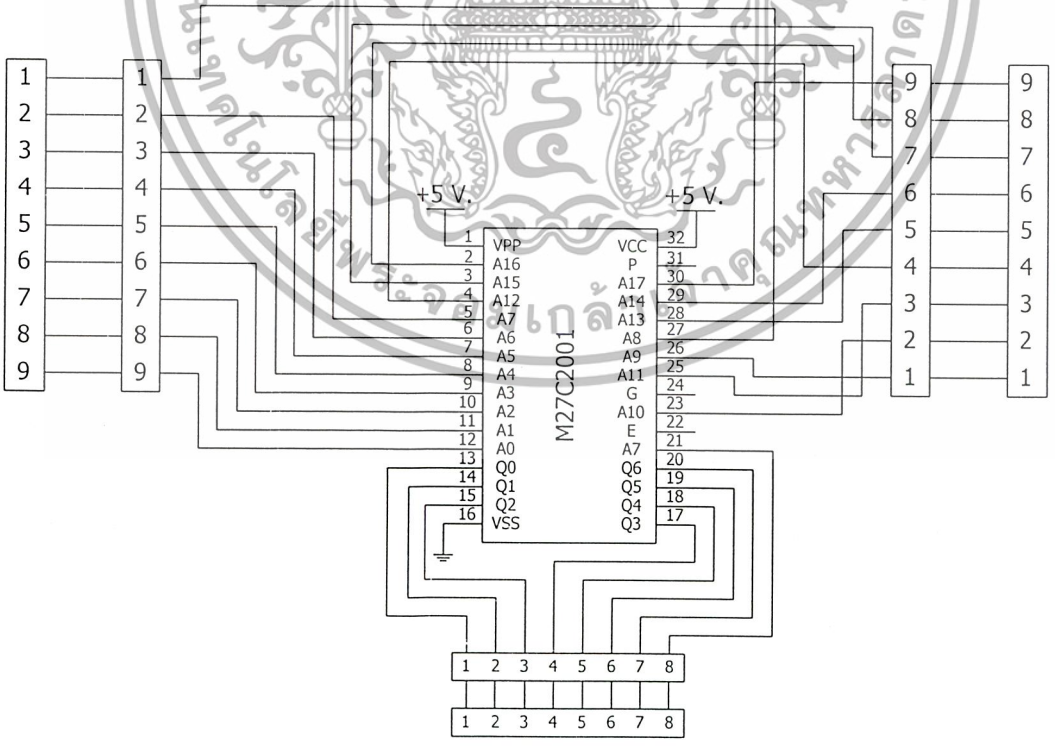
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 (ต่อ) ตำแหน่งขาของ IC 7402

ขาที่	คุณสมบัติ
10	V3
11	A4
12	B4
13	V4
14	VCC (+5 โวลต์)

3.2.12 โมดูลอีพรอม 256 กิโลไบต์

โมดูลอีพรอม 256 กิโลไบต์ ดังแสดงในรูปที่ 3.12 โดยใช้ IC เบอร์ M27C2001 มีจำนวนขาทั้งหมด 32 ขา สามารถจุข้อมูลได้ 256 KB โดยมีการอ้างถึงตำแหน่งได้ 17 บิต คือ A0 – A16 และมีพอร์ตข้อมูล 8 บิต คือ Q0 – Q7 ส่วนตำแหน่งของขา IC M27C2001 จะแสดงในตารางที่ 3.4 และโหมดการทำงาน ดังแสดงในตารางที่ 3.5



รูปที่ 3.12 วงจรอีพรอม 256 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 ตำแหน่งขาของ IC M27C2001

ขาที่	คุณสมบัติ
1	Vpp (Program Supply)
2	Address Input
3	Address Input
4	Address Input
5	Address Input
6	Address Input
7	Address Input
8	Address Input
9	Address Input
10	Address Input
11	Address Input
12	Address Input
13	Data Output
14	Data Output
15	Data Output
16	Ground
17	Data Output
18	Data Output
19	Data Output
20	Data Output
21	Data Output
22	Chip Enable
23	Address Input
24	Output Enable
25	Address Input
26	Address Input
27	Address Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (ต่อ) ตำแหน่งขาของ IC M27C2001

ขาที่	คุณสมบัติ
28	Address Input
29	Address Input
30	Address Input
31	Program
32	VCC

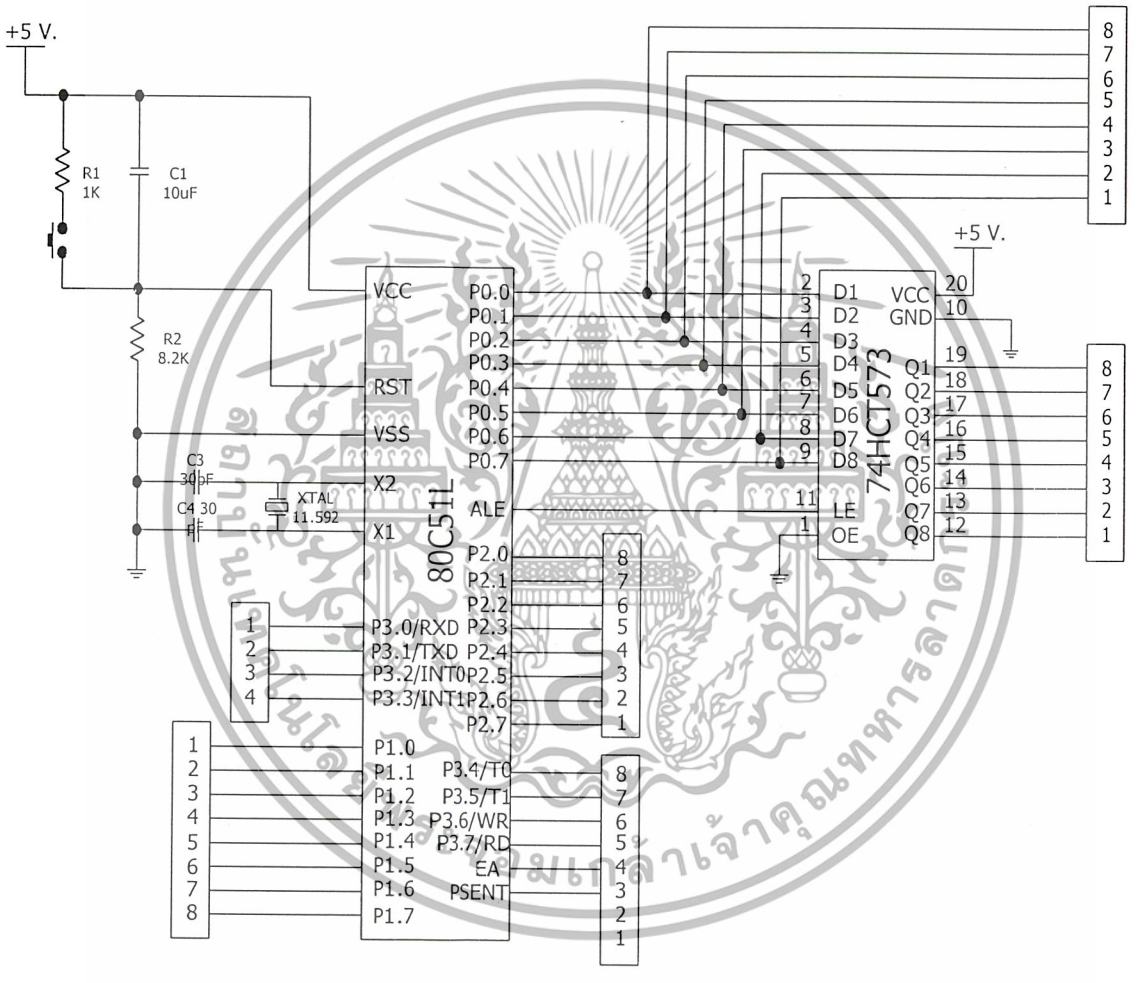
ตารางที่ 3.5 โหมดการทำงานของ IC M27C2001

Mode	$\overline{E}$	$\overline{G}$	$\overline{P}$	A9	Vpp	Q0 - Q7
Read	ViL	ViL	X	X	Vcc Or Vss	Data Out
Output Disable	ViL	ViH	X	X	Vcc Or Vss	Hi – z
Program	ViL	ViH	ViL Pulse	X	Vpp	Data In
Verify	ViL	ViH	ViH	X	Vpp	Data Out
Program Inhibit	ViH	X	X	X	Vpp	Hi – z
Stand By	ViH	X	X	X	Vcc Or Vss	Hi – z
Electronic Signature	Vih	ViL	ViH	ViD	Vcc	codes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.13 โมดูล MCS 8051

โมดูล MCS 8051 ดังแสดงในรูปที่ 3.13 เป็นไมโครคอนโทรลเลอร์ เบอร์ 80C51L สามารถเขียนโปรแกรมควบคุมโดยใช้ภาษา Assembly ซึ่งจะต่อ MCS 8051 แบบเบื้องต้นร่วมกับ IC 74HCT573 ซึ่งจะใช้ Latch ข้อมูล ส่วนของตำแหน่งขาของ MCS 80C51L จะแสดงในตารางที่ 3.6



รูปที่ 3.13 วงจร MCS 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L

ขาที่	คุณสมบัติ
1	Port 1.0
2	Port 1.1
3	Port 1.2
4	Port 1.3
5	Port 1.4
6	Port 1.5
7	Port 1.6
8	Port 1.7
9	Reset (Rst)
10	Port 3.0 RXD
11	Port 3.1 TXD
12	Port 3.2 $\overline{\text{INT 0}}$
13	Port 3.3 $\overline{\text{INT 1}}$
14	Port 3.4 T 0
15	Port 3.5 T 1
16	Port 3.6 $\overline{\text{WR}}$
17	Port 3.7 $\overline{\text{RD}}$
18	XTAL 2
19	XTAL 1
20	Ground
21	Port 2.0
22	Port 2.1
23	Port 2.2
24	Port 2.3
25	Port 2.4
26	Port 2.5
27	Port 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 (ต่อ) ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L

ขาที่	คุณสมบัติ
28	Port 2.7
29	$\overline{\text{PSEN}}$
30	ALE
31	$\overline{\text{EA}}$
32	Port 0.7
33	Port 0.6
34	Port 0.5
35	Port 0.4
36	Port 0.3
37	Port 0.2
38	Port 0.1
39	Port 0.0
40	VCC

พอร์ตของ 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีรายละเอียดต่าง ๆ ดังนี้

1) Vcc (ขา 40) ต่อกับ +5 โวลต์

2) Vss (ขา 20) เป็นขา GND

3) พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7–P0.0) มีโครงสร้างแบบ Open-Drain Bi-Direction

4) พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ 2 หน้าที่ คือ แอดเดรสบัส และดาต้าบัสเมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ต ถ้าต้องการ ให้ทำงานเป็นอินพุตพอร์ตต้องส่งลอจิก "1" ไปยังพอร์ตนี้ จะมีผลให้ $\overline{Q}$ ของ D-FF เป็น "0" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุตพอร์ตแลทซ์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (Pin) จะใช้สัญญาณ Read(Pin)

5) พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน Internal Pullup Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) พอร์ต 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.7-P2.0) มีโครงสร้างคล้าย พอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal Pull up) พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ทใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุตพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FET อยู่ในสถานะ off

7) พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ ขา (P3.7-P3.0) มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ตถ้าจะโปรแกรมให้เป็นอินพุตพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมาและรับสัญญาณเข้าไปโดยจะมีสัญญาณต่าง ๆ ดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)

P3.1/TXD (Serial Output Port)เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)

P3.2/$\overline{\text{INT0}}$ (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0

P3.3/$\overline{\text{INT1}}$ (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

P3.4/T0 (Counter 0 External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 0 (เป็นอินพุตโหมดเคาน์เตอร์)

P3.5/T1 (Counter 1 External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 1 (เป็นอินพุตโหมดเคาน์เตอร์)

P3.6/$\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียน ข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.7/$\overline{\text{RD}}$ (External Data Menory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

8) ALE (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตซ์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอตทีฟทุก ๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล

9) $\overline{\text{PSEN}}$ (ขา 29) เป็นขาสโตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้

10) $\overline{\text{EA}}$ (ขา 31) ใช้เลือกหน่วยความจำโปรแกรมภายนอก

ป้อน "0" จะอ่านโปรแกรมจากภายนอกชิพ

ป้อน "1" จะอ่านโปรแกรมจากภายในชิพ

11) RST (ขา 9) ขารีเซ็ท จะรีเซ็ทได้ก็ต่อเมื่อป้อนลอจิก "1" เข้าที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12) XTAL1 (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิสเลเตอร์ภายใน

13) XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาต์พุตของวงจรออสซิสเตอร์ภายใน

3.2.14 โมดูลสเตปปิ้งมอเตอร์

โมดูลสเตปปิ้งมอเตอร์ ดังแสดงในรูปที่ 3.14 เป็นวงจรที่ใช้สำหรับขับ Stepper Motor ขนาด 12โวลต์ให้ทำงานตามที่ต้องการ โดยวงจรประกอบด้วย IC ULN2003 เป็นตัวขับ Stepper Motor ขนาด 12โวลต์โดยดูรายละเอียดของขาในตารางที่ 3.7

รูปที่ 3.14 วงจรสเตปปิ้งมอเตอร์

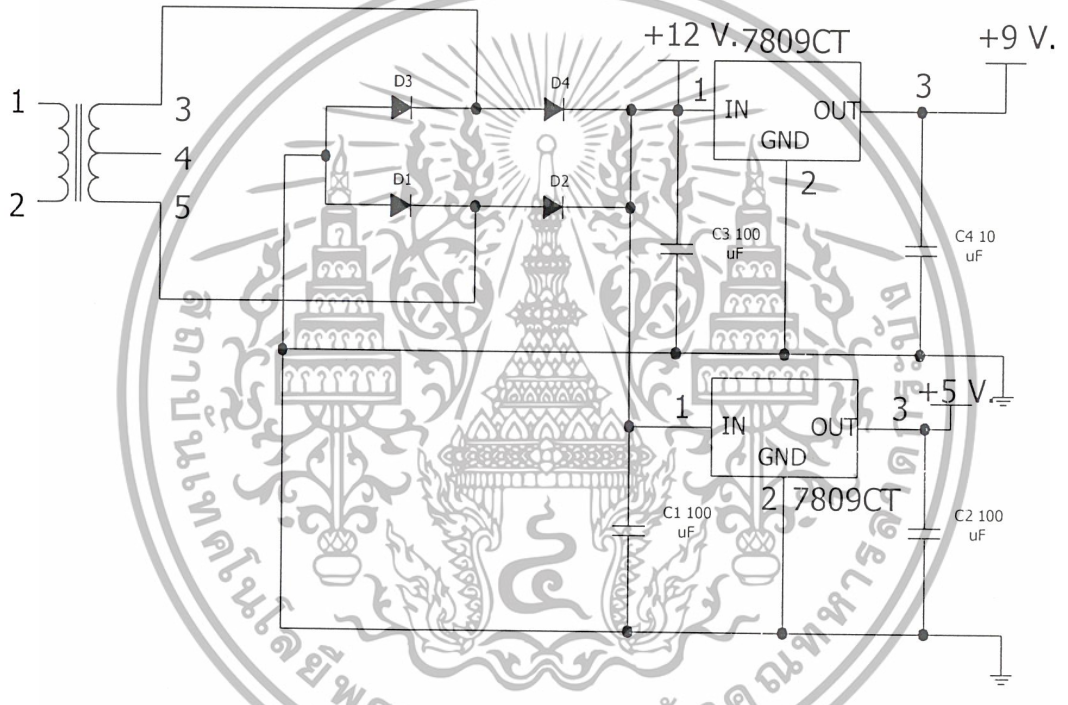
ตารางที่ 3.7 การทำงานของ ULN 2003

ขาที่	คุณสมบัติ
1	Input
2	Input
3	Input
4	Input
13	Output
14	Output
15	Output
16	Output
8	GND
9	+12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.15 โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5โวลต์, 9โวลต์ และ12โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย

โมดูล แหล่งจ่ายไฟฟ้ากระแสตรง 5โวลต์, 9โวลต์ และ 12โวลต์ ดังแสดงในรูปที่3.15 สามารถจ่ายไฟได้ถึง 3 ระดับ ดังนี้ +5โวลต์ , +9โวลต์ และ 12โวลต์ จะได้จากหม้อแปลงโดยตรง ส่วน 5โวลต์ และ 9โวลต์ จะได้จากวงจรเร็กกูเลเตอร์ของ IC7805 และ IC7809 ตามลำดับโดยแต่ละระดับแรงดันทั้ง 3 ของแหล่งจ่ายมีพอร์ตอย่างละ 10 พอร์ต

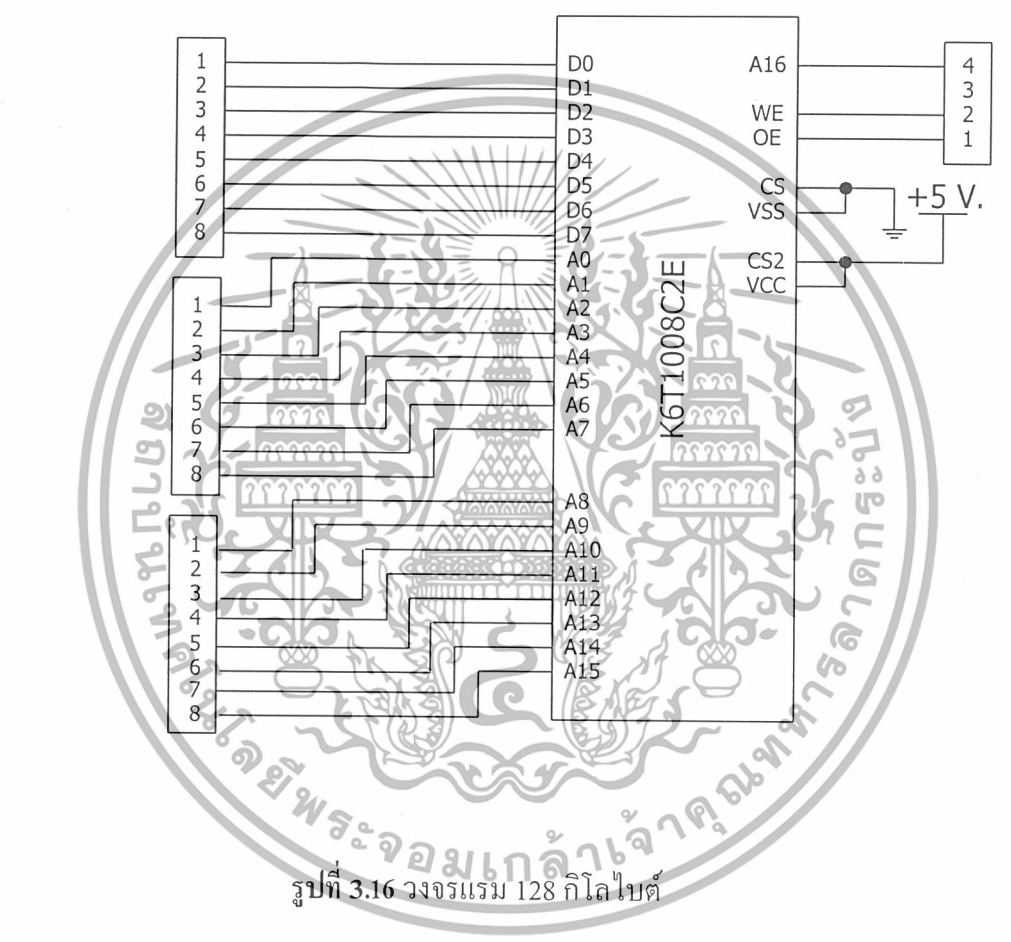


รูปที่ 3.15 วงจรแหล่งจ่ายไฟฟ้ากระแสตรง 5โวลต์, 9โวลต์ และ 12โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.16 โมดูลแรม 128 กิโลไบต์

โมดูลแรม 128 กิโลไบต์ ดังแสดงในรูปที่ 3.16 ใช้ IC K6T1008C2E เป็นหลัก ทำหน้าที่เป็นหน่วยความจำโดยสามารถที่จะอ่านหรือเขียนข้อมูลลงไปก็ได้ ส่วนตำแหน่งขาของ K6T1008C2E จะแสดงในตารางที่ 3.8



รูปที่ 3.16 วงจรแรม 128 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ตำแหน่งขาของ IC K6T1008C2E

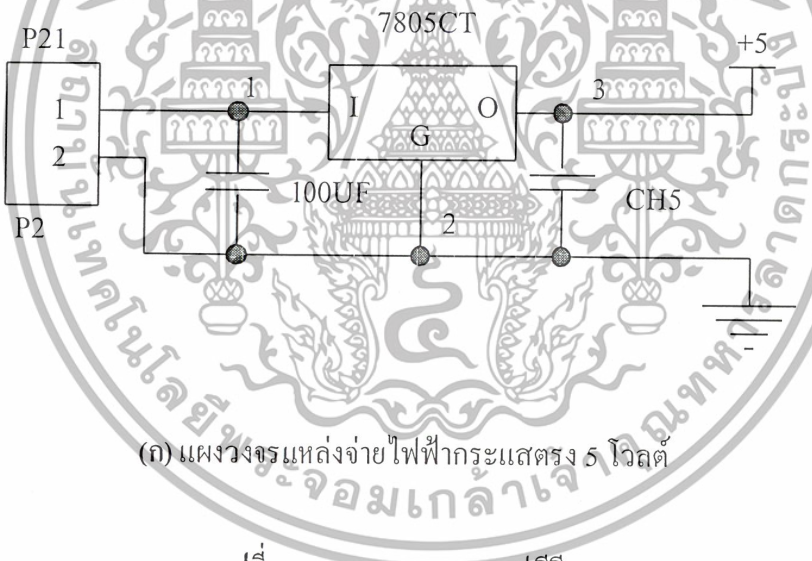
ขาที่	คุณสมบัติ
1	N.C.
2 - 4	A16 ,A14,A12
5 - 12	A7 – A0
13 - 15	$I/O_1 – I/O_3$
16	VSS
17 - 21	$I/O_4 – I/O_8$
22	$\overline{ES1}$
23	AO
24	$\overline{OE}$
25	A11
26	A9
27	A8
28	A13
29	$\overline{WE}$
30	CS2
31	A15
32	VCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.17 แผงวงจรเอฟพีจีเอ

แผงวงจรเอฟพีจีเอ ดังแสดงในรูปที่ 3.17 ออกแบบมาเพื่อใช้ในการศึกษาการออกแบบวงจรดิจิตอลด้วยอุปกรณ์เอฟพีจีเอ (Field Programmable Gate Arrays) ของบริษัท Xilinx ตระกูล XC 4000E โดยแผงวงจรเอฟพีจีเอ นี้สามารถโปรแกรมการทำงาน (configuration) ผ่านทางชุดสายดาวน์โหลดโดยหน้าที่การทำงานของ XC4000E แต่ละขาจะแสดงในตารางที่ 3.9 และในแผงวงจรทดลองประกอบด้วย

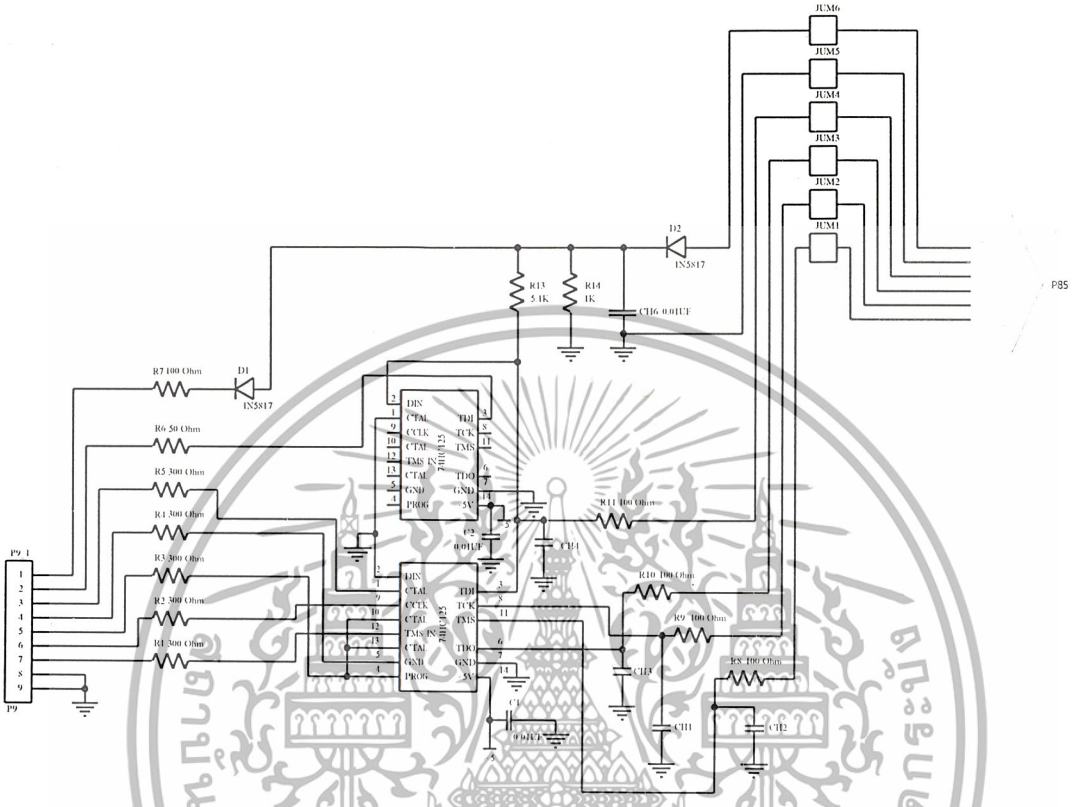
1) เอฟพีจีเอ ตระกูล XC4000E ขนาด 84 ขา
2) ชุดสายดาวน์โหลดแบบขนาน
3) คอนเน็คเตอร์สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกขนาด 8 บิต (P81,P82,P83,P84)
4) ชุดแหล่งจ่ายไฟ +5 V



(ก) แผงวงจรแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

รูปที่ 3.17 วงจรแผงวงจรเอฟพีจีเอ

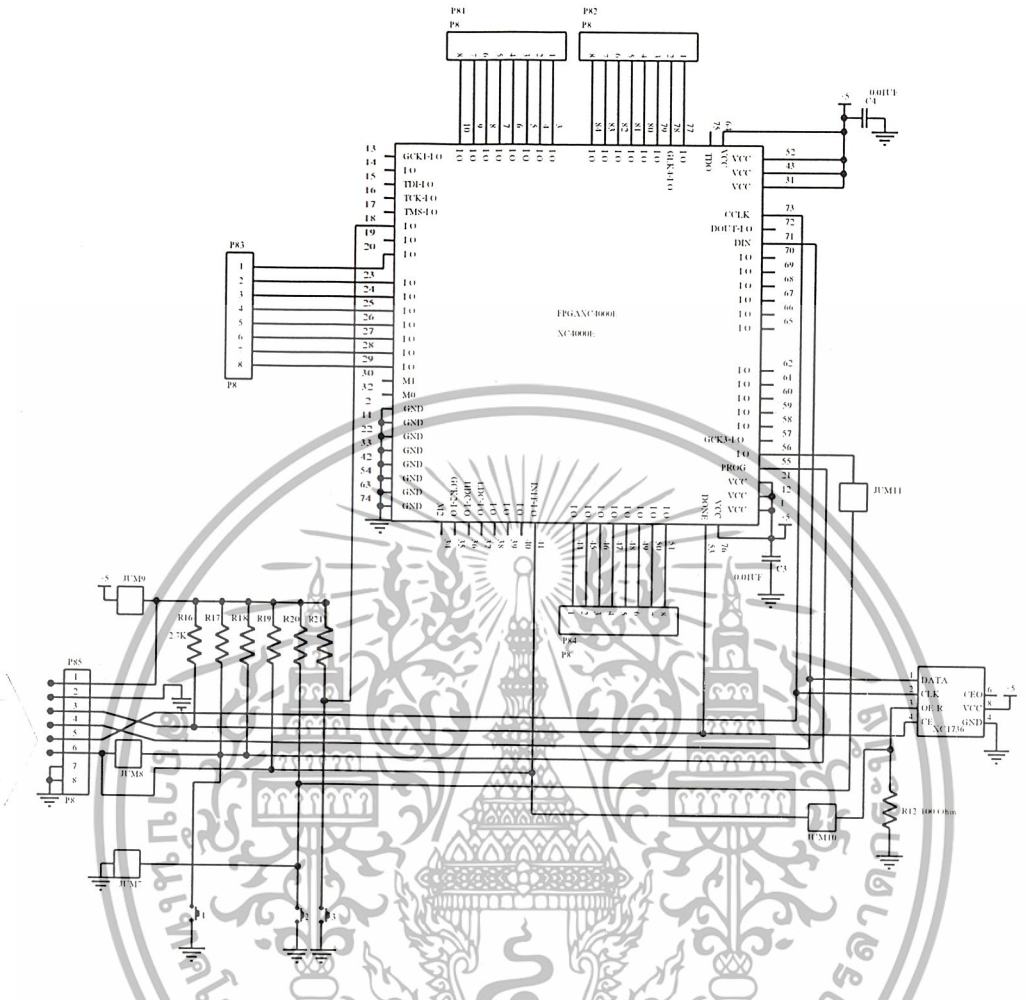
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) ส่วนที่ 1

รูปที่ 3.17 (ต่อ) วงจรแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค) ส่วนที่ 2

รูปที่ 3.17 (ต่อ) วงจรแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 หน้าที่การทำงานของ XC4000E แต่ละขา

ขาที่	คุณสมบัติ
1	GND
2	VCC
3 – 10	I/O
11	VCC
12	GND
13	GCK1 – I/O
14	I/O
15	TDI – I/O
16	TCK – I/O
17	TMS – I/O
18 – 20	I/O
21	GND
22	VC
23 – 29	I/O
30	M1
31	GND
32	M0
33	VCC
34	M2
35	GCK2 – I/O
36	HDC – I/O
37	LDC – I/O
38	I/O
39 - 40	I/O
41	INIT – I/O
42	VCC
43	HND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 (ต่อ) หน้าที่การทำงานของ XC4000E แต่ละขา

ขาที่	คุณสมบัติ
44 – 51	I/O
52	GND
53	DONE
54	VCC
55	PROG
56	I/O
57	GCK3 – I/O
58 - 62	I/O
63	VCC
64	GND
65 - 70	I/O
71	DIN
72	DOUT – I/O
73	CCLK
74	VCC
75	TD0
77 - 84	I/O

ตารางที่ 3.10 ชุดดาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด

ชื่อ	คุณสมบัติ
+5 โวลต์	ไฟเลี้ยง VDC (VCC)
GND	ระดับสัญญาณอ้างอิง
CCLK	สัญญาณนาฬิกาที่ใช้ในโปรแกรมและอ่านข้อมูลกลับ เอฟพีจีเอ ต่อเข้าที่ขา
DONE	73
DIN	ใช้ตรวจสอบสภาวะการโปรแกรมข้อมูลต่อเข้าที่ขา 53 ของ เอฟพีจีเอ
	ส่งข้อมูลที่โปรแกรมเข้าที่ขา 71 ของ เอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 (ต่อ) ชุดดาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด

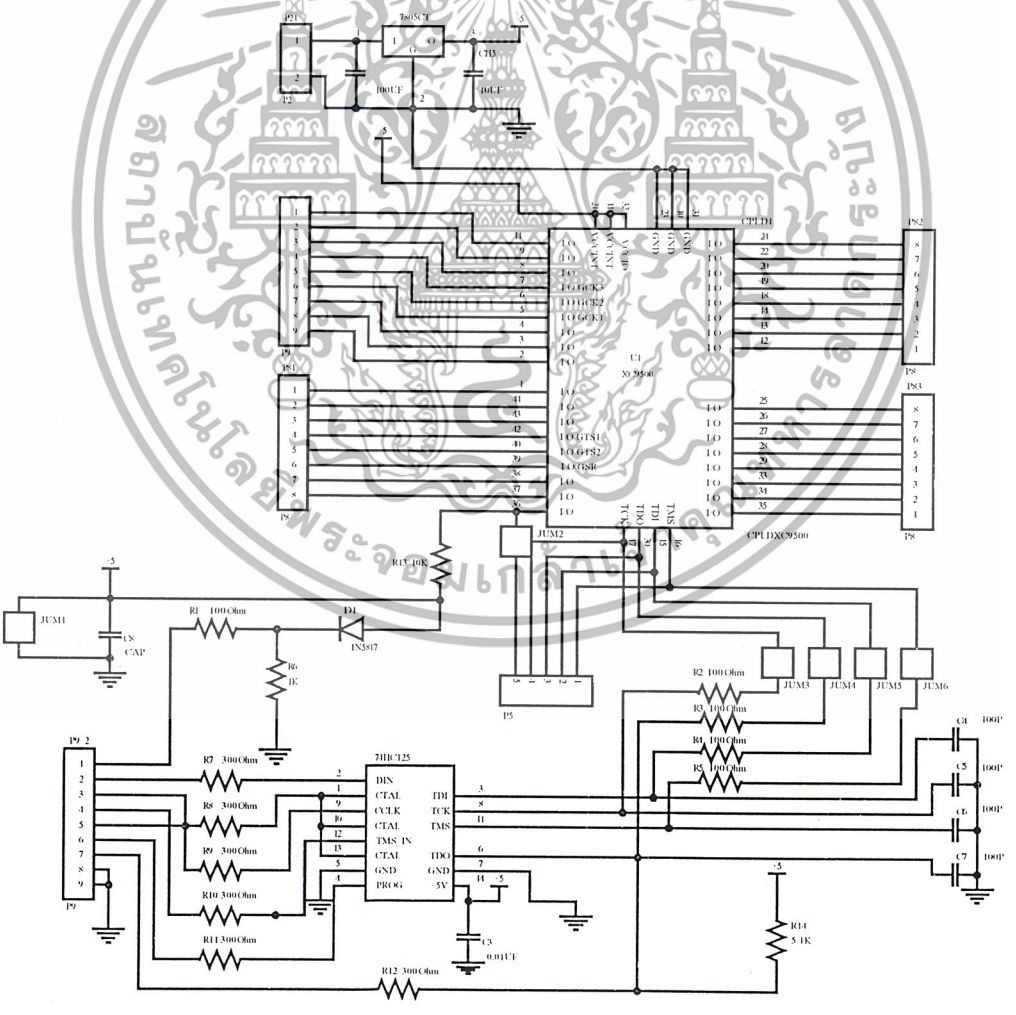
ชื่อ	คุณสมบัติ
PROG	ขานี้จะส่งสัญญาณพัลซ์มาเมื่อสั่งให้ เอฟพีจีเอ เข้าสู่การโปรแกรมข้อมูล ต่อเข้าที่ขา 55 ของเอฟพีจีเอ
INIT	ขานี้จะเป็นลอจิก “0” ถ้าข้อมูลที่โปรแกรมมีการผิดพลาด (CRC ERROR) ต่อเข้าที่ขา 41
RST	ต่อเข้าที่ JUM 5 ซึ่งถ้ามีการต่อเอาไว้ Xchecker จะสามารถส่งสัญญาณรีเซ็ทมายัง เอฟพีจีเอ (เหมือนการกดปุ่ม Reset)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.18 แผงวงจรซีพีแอลดี

แผงวงจรซีพีแอลดี ดังแสดงในรูปที่ 3.18 ออกแบบขึ้นมาเพื่อใช้ในการศึกษาการออกแบบวงจรดิจิตอลด้วยอุปกรณ์ซีพีแอลดี (Complex Programmable Logic Devices) ยี่ห้อ Xilinx ตระกูล XC 9500 โดยหน้าที่การทำงานของแต่ละขาจะแสดงในตารางที่ 3.11 และสามารถโปรแกรมการทำงาน (Configuration) ผ่านทางพอร์ตขนานของเครื่องคอมพิวเตอร์ (In – System Programmable หรือ ISP) ซึ่งประกอบด้วย

1) ซีพีแอลดี ตระกูล XC9500 ขนาด 44 ขา
2) ชุดดาวน์โหลดแบบพอร์ตขนาน
3) คอนเน็คเตอร์สำหรับเชื่อมต่ออุปกรณ์ภายนอก 34 อินพุต/เอาต์พุต (P81,P82,P83,P91)
4) ชุดจ่ายไฟ +5 โวลต์



รูปที่ 3.18 วงจรแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 หน้าที่การทำงานของ XC9500 แต่ละขา

ขาที่	คุณสมบัติ
1	I/O
2	I/O
3	I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	I/O
9	I/O
10	GND
11	I/O
12	I/O
13	I/O
14	I/O
15	TDI
16	TMS
17	TCK
18	I/O
19	I/O
20	I/O
21	VCC
22	I/O
23	GND
24	I/O
25	I/O
26	I/O
27	I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 (ต่อ) หน้าที่การทำงานของ XC9500 แต่ละขา

ขาที่	คุณสมบัติ
28	I/O
29	I/O
30	TDO
31	GND
32	VCC
33	I/O
34	I/O
35	I/O
36	I/O
37	I/O
38	I/O
39	I/O /GSR
40	I/O /GTS2
41	VCC
42	I/O /GTS1
43	I/O
44	I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 ชุดดาวน์โหลดโปรแกรมผ่านทางสายดาวน์โหลด

ชื่อ	คุณสมบัติ
VCC_SENS	ตรวจสอบสภาวะไฟเลี้ยงบนชุดซีพีแอลดี
CCLK	ส่งข้อมูลที่โปรแกรมเข้าที่ขา TD1(ขา 15) ของ (XC9500)
TMS_IN	สัญญาณนาฬิกาที่ใช้ในการโปรแกรม XC9500 ที่ขา TMS (ขา 16)
CTRL	ควบคุมสัญญาณต่าง ๆ ที่ติดต่อกับ XC9500
PROG	ใช้ในการตรวจสอบการทำงานของชุดดาวน์โหลดเคเบิ้ล
DONE	อ่านข้อมูลจากตัว XC9500 ที่ขา TD0 (ขา 30) เพื่อตรวจสอบความถูกต้องขณะทำการโปรแกรม
GND	ระดับสัญญาณอ้างอิงของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

โครงงานนี้ได้มีการออกแบบชุดปฏิบัติการไว้ทั้งหมด 3 ส่วนด้วยกันคือ แผงวงจรซีพีแอลดี แผงวงจรเอฟพีจีเอ และแผงวงจรอินพุต/เอาต์พุตซึ่งจะช่วยให้ผู้ใช้ได้ฝึกและทดลองออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดีเพื่อให้มีทักษะมากขึ้น ซึ่งได้ทำการทดลองส่วนต่าง ๆ ของชุดปฏิบัติการที่ได้ออกแบบ ดังนี้

4.1 แผงวงจรซีพีแอลดี

4.1.1 อุปกรณ์ที่ใช้

1) แผงวงจรซีพีแอลดี
2) ระบบคอมพิวเตอร์
3) โปรแกรม Xilinx Foundation F2.11
4) สายดาวน์โหลดแบบขนาน
5) โปรแกรมภาษาวีเอชดีแอล บรรยายวงจรนับ 0 – 9
6) โมดูล แสดงผลเจ็ดส่วน

4.1.2 ขั้นตอนการทดลอง

1) ทำการต่อเครื่องคอมพิวเตอร์กับ แผงวงจรซีพีแอลดี
2) ทำการดาวน์โหลดโปรแกรมนับ 0 – 9 ลงแผงวงจรซีพีแอลดี
3) สังเกตการทำงานของโปรแกรมด้วย แสดงผลเจ็ดส่วน

4.1.3 ผลการทดลอง

แสดงผลเจ็ดส่วน เริ่มทำงานโดยแสดงผลตั้งแต่ 0 – 9

4.1.4 สรุป

จากผลการทดลอง แสดงผลเจ็ดส่วน สามารถแสดงผลได้ตามที่โปรแกรมเขียนไว้นั้นแสดงให้เห็นว่าแผงวงจรซีพีแอลดี ทำงานได้โดยสามารถดาวน์โหลดและแสดงผลได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 แผงวงจรเอฟพีจีเอ

4.2.1 อุปกรณ์ที่ใช้

1) แผงวงจรเอฟพีจีเอ

2) ระบบคอมพิวเตอร์

3) โปรแกรม Xilinx Foundation F2.1I

4) สายดาวน์โหลดแบบขนาน

5) โปรแกรมภาษาวีเอชดีแอล บรรยายวงจรนับ 0 – 9

6) โมดูลแสดงผลเจ็ดส่วน Common Anode

4.2.2 ขั้นตอนการทดลอง

1) ทำการต่อเครื่องคอมพิวเตอร์กับแผงวงจรเอฟพีจีเอ

2) ทำการดาวน์โหลดโปรแกรมนับ 0 – 9 ลงแผงวงจรเอฟพีจีเอ

3) สังเกตุการทำงานของโปรแกรมด้วย แสดงผลเจ็ดส่วน

4.2.3 ผลการทดลอง

แสดงผลเจ็ดส่วน เริ่มทำงานโดยแสดงผลตั้งแต่ 0 – 9

4.2.4 สรุป

จากที่ แสดงผลเจ็ดส่วน เริ่มทำงานโดย แสดงผลตามต้องการแสดงให้เห็นว่าแผงวงจร เอฟพีจีเอ สามารถทำงานได้ตามที่ต้องการ โดยสามารถดาวน์โหลดโปรแกรมและแสดงผลออกได้ตามที่ต้องการ

4.3 แผงวงจรอินพุต/เอาต์พุต

ในส่วนของแผงวงจรอินพุต/เอาต์พุต จะแบ่งออกเป็นโมดูลย่อย ๆ ทั้งหมด 16 โมดูล ดังนี้

4.3.1 โมดูลแสดงผลเจ็ดส่วน

1) อุปกรณ์ที่ใช้

1.1) โมดูลแสดงผลเจ็ดส่วน 8 หลัก แบบ คอมม่อนเอโหนด

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) จ่ายไฟบวกให้กับขา คอมม่อน ของ แสดงผลเจ็ดส่วน ทุกตัว

2.2) จ่ายลบให้กับขา a,b,c,d,e,f,g ของ แสดงผลเจ็ดส่วน ทุกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ผลการทดลอง

ตารางที่ 4.1 ผลการทดลองโมดูลแสดงผลเจ็ดส่วน

แสดงผลเจ็ดส่วน (ตัวที่)	a	b	c	d	e	f	g
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1

4) สรุป

วงจรแสดงผลเจ็ดส่วน ทำงานได้ทุกตัว

4.3.2 โมดูลดิฟสวิตช์ 8 บิต

1) อุปกรณ์ที่ใช้

1.1) โมดูลดิฟสวิตช์ 8 บิต 2 ตัว

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.3) แอลอีดี 16 ดวง

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟเข้ากับโมดูลดิฟสวิตช์ 8 บิต

2.2) ต่อเอาต์พุตเข้ากับ แอลอีดี 16 ดวง

2.3) ทดลองเลื่อน ดิฟสวิตช์แล้วบันทึกผลการทดลองลงในตารางที่ 4.2

3) ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดลองโมดูลดิฟสวิตช์ 8 บิต

อินพุต		เอาต์พุต (แอลอีดี)															
P I	P 8	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16																
1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามทำสำเนาหรือดัดแปลงส่วนใดส่วนหนึ่งของวิทยานิพนธ์นี้ก่อนได้รับอนุญาตเป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์ ผู้ที่ฝ่าฝืนจะถูกดำเนินคดีตามกฎหมาย

4) สรุป

ดิฟสวิตช์สามารถทำงานได้ทุกแชลแนล

4.3.3 โมดูลคีย์บอร์ดเมตริกซ์ 4x4

1) อุปกรณ์ที่ใช้

1.1) โมดูลคีย์บอร์ดเมตริกซ์ 4x4

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.3) แอลอีดี 4 ดวง

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟเข้ากับ โมดูลคีย์บอร์ดเมตริกซ์ 4x4

2.2) ต่อเอาต์พุตของโมดูลคีย์บอร์ดเมตริกซ์ 4x4 เข้ากับ แอลอีดี ทั้งหมด 4 ดวง

2.3) ทดลองกดปุ่มตั้งแต่ 0 - F

3) ผลการทดลอง

ตารางที่ 4.3 ผลการทดลองโมดูลคีย์บอร์ดเมตริกซ์ 4x4

อินพุต	เอาต์พุต (แอลอีดี)	อินพุต	เอาต์พุต (แอลอีดี)
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

4) สรุป

โมดูลคีย์บอร์ดเมตริกซ์ 4x4 ทำงานได้ถูกต้องตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 โมดูลสเตปปิ้งมอเตอร์

1) อุปกรณ์ที่ใช้

1.1) โมดูลสเตปปิ้งมอเตอร์

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 12 โวลต์

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟฟ้ากระแสตรง 12 โวลต์ เข้ากับโมดูลสเตปปิ้งมอเตอร์

2.2) ทำการจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ให้พอร์ตอินพุตของสเตปปิ้งมอเตอร์

3) ผลการทดลอง

ตารางที่ 4.4 ผลการทดลอง โมดูลสเตปปิ้งมอเตอร์

พอร์ต				สเตปปิ้งมอเตอร์
1	2	3	4	
0	0	0	1	ขวา
0	0	1	0	ขวา
0	1	0	0	ขวา
1	0	0	0	ขวา
1	0	0	0	เซต
0	1	0	0	ซ้าย
0	0	1	0	ซ้าย
0	0	0	1	ซ้าย

4) สรุป

วงจรทำงานได้ผลตามที่ต้องการ

4.3.5 โมดูลแอลอีดี 40 ดวง

1) อุปกรณ์ที่ใช้

1.1) โมดูลแอลอีดี 40 ดวง

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) ต่อกราวด์ เข้ากับพอร์ตกราวด์ของโมดูลแอลอีดี 40 ดวง

2.2) จ่ายไฟบวกกระแสตรง 5 โวลต์ ให้กับ แอลอีดี ทั้งหมดโดยผ่านพอร์ต 1 – 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ผลการทดลอง

ตารางที่ 4.5 ผลการทดลองโมดูลแอลอีดี 40 ดวง

พอร์ต	แอลอีดี							
	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1

4) สรุป

โมดูลแอลอีดี สีแดง 40 ดวง 5 แถว 8 หลัก ทำงานได้ถูกต้องตามต้องการ

4.3.6 โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก

1) อุปกรณ์ที่ใช้

1.1) โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก

1.2) โวลต์มิเตอร์

1.3) แหล่งจ่ายไฟฟ้ากระแสตรง 12 โวลต์

1.4) แอลอีดี 8 ดวง

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟฟ้ากระแสตรง 12 โวลต์ ให้กับโมดูลแปลงสัญญาณดิจิตอล เป็นแอนะลอก

2.2) ป้อนสัญญาณไฟฟ้าแบบไบนารี่ 8 บิต เข้าพอร์ตอินพุตของ โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก

2.3) วัดค่าเอาต์พุตที่แสดงผล

3) ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ผลการทดลองโมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก

อินพุต (แอลอีดี)								เอาต์พุต
1	2	3	4	5	6	7	8	(โวลต์มิเตอร์)
0	0	0	0	0	0	0	0	1 โวลต์
0	1	0	1	0	0	0	0	2 โวลต์
1	0	0	1	0	0	0	0	3 โวลต์
1	1	0	0	1	0	0	1	4 โวลต์
1	1	1	1	1	1	1	1	5 โวลต์

4) สรุป

จากการทดลองป้อนสัญญาณไฟฟ้าแบบไบนารี่ 8 บิต เข้าพอร์ตอินพุตของโมดูล แปลงสัญญาณดิจิตอลเป็นแอนะลอก แล้ววัดค่าเอาต์พุตที่ได้เอาต์ตามรหัสที่ป้อนเข้าไป แสดงว่าวงจรทำงานได้ตามที่ต้องการ

4.3.7 โมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล

1) อุปกรณ์ที่ใช้

1.1) โมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล

1.2) โวลต์มิเตอร์

1.3) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.4) แอลอีดี 8 ดวง

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ให้กับโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล

2.2) จ่ายแรงดันไฟที่ต่างกันดังตารางที่ 4.7

2.3) วัดค่าเอาต์พุตแล้วบันทึกผลในตารางที่ 4.7

3) ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ผลการทดลองโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล

อินพุต (โวลต์มิเตอร์)	เอาต์พุต (แอลอีดี)							
	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	1
2	0	1	1	0	1	0	0	1
3	1	0	0	1	1	1	1	0
4	1	0	1	0	0	0	1	1
5	1	1	1	1	1	0	1	0

4) สรุป

ระดับอินพุตที่เข้ามาต่างกันมีผลทำให้ได้เอาต์พุตที่ต่างกันดังนั้นแสดงว่าวงจรทำงานได้ถูกต้อง

4.3.8 โมดูลมอเตอร์ไฟฟ้ากระแสตรง

1) อุปกรณ์ที่ใช้

1.1) โมดูลมอเตอร์ไฟฟ้ากระแสตรง

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) ต่อแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์โดยจ่ายไฟบวกให้เข้าขั้วบวกของ มอเตอร์ไฟฟ้ากระแสตรงและไฟลบให้เข้าขั้วลบของ มอเตอร์ไฟฟ้ากระแสตรง แล้วบันทึกผล

2.2) ต่อแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์โดยจ่ายไฟลบให้เข้าขั้วบวกของ มอเตอร์ไฟฟ้ากระแสตรงและไฟบวกให้เข้าขั้วลบของ มอเตอร์ไฟฟ้ากระแสตรง แล้วบันทึกผลในตารางที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ผลการทดลอง

ตารางที่ 4.8 ผลการทดลองโมดูลมอเตอร์ไฟฟ้ากระแสตรง

อินพุต		ลักษณะการหมุน
ขั้วบวก	ขั้วลบ	
ไฟบวก	ไฟลบ	หมุนขวา
ไฟลบ	ไฟบวก	หมุนซ้าย

4) สรุป

วงจรมอเตอร์ไฟฟ้ากระแสตรง สามารถทำงานได้ตามที่กำหนด

4.3.9 โมดูลลำโพง

1) อุปกรณ์ที่ใช้

1.1) โมดูลลำโพง

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) ป้อนแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ให้กับอินพุตของวงจรแล้วบันทึกผลในตารางที่ 4.9

3) ผลการทดลอง

ตารางที่ 4.9 ผลการทดลอง โมดูลลำโพง

อินพุต (โวลต์)	เอาต์พุต
1	มีเสียง
0	ไม่มีเสียง

4) สรุป

วงจรลำโพง สามารถทำงานได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.10 โมดูลแรม 128 กิโลไบต์

1) อุปกรณ์ที่ใช้

1.1) โมดูลแรม 128 กิโลไบต์

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.3) แอลอีดี 8 ดวง

2) ขั้นตอนการทดลอง

2.1) ป้อนแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ให้กับโมดูลแรม 128 กิโลไบต์

2.2) ต่อแอลอีดี 8 ดวงเข้าที่ พอร์ตอินพุต/เอาต์พุต 1-8

2.3) ทำการอ้างแอดเดรส และทำการเขียนข้อมูล

2.4) ทำการอ้างแอดเดรส และทำการอ่านข้อมูล

3) ผลการทดลอง

ตารางที่ 4.10 ผลการทดลองการเขียนข้อมูลของโมดูลแรม 128 กิโลไบต์

แอดเดรส								ดาต้า (เข้า)								
A7	A6	A5	A4	A3	A2	A1	A0	I/O8	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	$\overline{WE}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0

ตารางที่ 4.11 ผลการทดลองการอ่านข้อมูลของโมดูลแรม 128 กิโลไบต์

แอดเดรส								ดาต้า (เข้า)								
A7	A6	A5	A4	A3	A2	A1	A0	I/O8	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	$\overline{OE}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) สรุป

จากผลการทดลองสามารถที่จะเขียนข้อมูลและอ่านข้อมูลจากแรม ได้อย่าง ถูกต้อง

4.3.11 โมดูลอีพรอม 256 กิโลไบต์

1) อุปกรณ์ที่ใช้

1.1) โมดูลอีพรอม 256 กิโลไบต์

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.3) แอลอีดี 8 ดวง

2) ขั้นตอนการทดลอง

2.1) ป้อนแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ให้กับโมดูลอีพรอม 256 กิโลไบต์

2.2) ต่อแอลอีดี เข้ากับพอร์ต Q0 – Q7

2.3) ทดลองอ่านข้อมูลด้วยการอ้างค่าแอดเดรสและทริกที่ขาเอาต์พุต Enable($\overline{G}$)

3) ผลการทดลอง

ตารางที่ 4.12 ผลการทดลองโมดูลอีพรอม 256 กิโลไบต์

Read	แอดเดรส																	ดาต้า							
$\overline{G}$	A 17	A 16	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

4) สรุป

สามารถอ่านข้อมูลที่ทำการบันทึกไว้ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.12 โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ,9 โวลต์ ,12 โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย

1) อุปกรณ์ที่ใช้

1.1) โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ,9 โวลต์ ,12 โวลต์ จ่ายไฟเลี้ยงแผงวงจรย่อย

1.2) โวลต์มิเตอร์

2) ขั้นตอนการทดลอง

2.1) จ่ายไฟ 220 โวลต์ ให้กับโมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ,9 โวลต์12 โวลต์

2.2) ทำการวัดค่าเอาต์พุตทั้ง 3 ระดับ

3) ผลการทดลอง

ตารางที่ 4.13 ผลการทดลองของ โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ,9 โวลต์, 12 โวลต์

ระดับแรงดัน	ค่าที่วัดได้ (โวลต์)
+5 โวลต์	+5โวลต์
+9 โวลต์	+9.5 โวลต์
+12โวลต์	+12 โวลต์

4) สรุป

ผลการวัดค่าแรงดันทางเอาต์พุตของวงจรที่จะจ่ายไฟทั้งหมดเป็นไปตามที่ต้องการถึงแม้ว่าในย่าน 9 โวลต์ จะออกมาถึง 9.5โวลต์ ก็ตามซึ่งยังถือว่าทำงานได้จริง

4.3.13 โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว

1) อุปกรณ์ที่ใช้

1.1) โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) ป้อนแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์ ให้กับโมดูลสวิตช์กดติดปล่อยดับ 16 ตัว

2.2) ทดลองกดสวิตช์และบันทึกผลในตารางที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ผลการทดลอง

ตารางที่ 4.14 ผลการทดลองของ โมดูลสวิตช์กดติดปล่อยดับ 16 ตัว

สวิตช์	แอลอีดี															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

4) สรุป

จากผลการทดลองจะได้ผลที่ถูกต้องตามที่ต้องการ

3.4.14 โมดูลวงจรสี่แยกไฟแดง

1) อุปกรณ์ที่ใช้

1.1) โมดูลวงจรสี่แยกไฟแดง

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

2) ขั้นตอนการทดลอง

2.1) ทำการป้อนค่าอินพุตตามตารางที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ผลการทดลอง

ตารางที่ 4.15 ผลการทดลองโมดูลวงจรสี่แยกไฟแดง

อินพุต				เอาต์พุต (แอลอีดี)											
A	B	C	D	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0
0	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1
0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0
1	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0

4) สรุป

วงจรสี่แยกไฟแดง สามารถที่จะทำงานตามฟังก์ชั่นที่กำหนดได้

3.4.15 โมดูลวงจรกำเนิดสัญญาณนาฬิกา

1) อุปกรณ์ที่ใช้

1.1) โมดูลวงจรกำเนิดสัญญาณนาฬิกา

1.2) แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

1.3) แอลอีดี 1 ดวง

2) ขั้นตอนการทดลอง

2.1) ป้อนแหล่งจ่าย ไฟฟ้ากระแสตรง 5 โวลต์ ให้กับโมดูลวงจรกำเนิดสัญญาณนาฬิกา

2.2) ต่อแอลอีดี เข้าที่พอร์ตเอาต์พุตของโมดูลวงจรกำเนิดสัญญาณนาฬิกา

2.3) ทดลองปรับค่า ความต้านทานปรับค่าได้แล้วบันทึกผลการทดลอง

3) ผลการทดลอง

จากการทดลอง เมื่อทำการลดค่าของความต้านทานปรับค่าได้ ให้มีค่าความต้านทานต่ำ จะทำให้แอลอีดีกระพริบช้า ๆ แต่เมื่อทำการเพิ่มค่าของความต้านทานปรับค่าได้ ให้มีความต้านทานสูงขึ้น แอลอีดีจะกระพริบเร็วขึ้นและเมื่อปรับค่าของความต้านทานปรับค่าได้จนค่าความต้านทานมีค่าสูงสุด แอลอีดีจะสว่างจนมองไม่เห็นการกระพริบของแอลอีดี

4) สรุป

วงจรกำเนิดสัญญาณนาฬิกาสามารถใช้งานได้จริง โดยสังเกตผลการกระพริบของแอลอีดี และได้จากการคำนวณค่าความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.16 โมดูล MCS 8051

1) อุปกรณ์ที่ใช้

1.1) โมดูล MCS 8051 1 ชุด

1.2) โมดูลแสดงผลเจ็ดส่วน

2) ขั้นตอนการทดลอง

2.1) ทดลองเขียนโปรแกรม นับ 0 – 9

2.2) นำโปรแกรมที่ได้ออกแบบเขียนลงใน MCS 8051

2.3) ต่อแสดงผลเจ็ดส่วน ที่พอร์ตเอาต์พุตของโมดูล MCS 8051

2.4) ทำการรันโปรแกรมแล้วสังเกตการทำงานของวงจร MCS 8051

3) ผลการทดลอง

โปรแกรมที่ได้ออกแบบสามารถทำงานได้จริง โดยสังเกตจาก แสดงผลเจ็ดส่วน ที่เริ่มทำงานตามโปรแกรมโดยเปลี่ยนแปลงจาก 0 – 9

4) สรุป

จากการทดลองพบว่า วงจร MCS 8051 ที่ได้ทำการออกแบบสามารถติดต่อกับแผงวงจรเอฟพีจีเอ ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

5.1 บทสรุป

จากการศึกษาค้นคว้าหาข้อมูลต่างๆ เกี่ยวกับการเขียนโปรแกรมภาษาวีเอชดีแอล การออกแบบวงจรของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับเอฟพีจีเอ และซีพีแอลดี ทำให้ชุดทดลองดังกล่าวสำเร็จลุล่วงไปได้ด้วยดี การทดลองตามใบงานต่างๆ ที่ได้ออกแบบมานั้น สามารถที่จะนำไปใช้งานได้จริง ซึ่งระดับของการใช้งานของชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับเอฟพีจีเอ และซีพีแอลดี ได้บรรลุตามวัตถุประสงค์ที่ตั้งไว้ ซึ่งผู้ใช้สามารถที่จะนำชุดปฏิบัติการดังกล่าวไปใช้งานในการฝึกปฏิบัติได้จริง

5.2 ปัญหา และแนวทางการแก้ไข

1) ปัญหา เครื่องคอมพิวเตอร์ที่ใช้ในการทำปริญญานิพนธ์มีประสิทธิภาพในการทำงานต่ำ
แนวทางการแก้ไข จัดหาเครื่องคอมพิวเตอร์เพิ่มขึ้นทั้งจำนวนและประสิทธิภาพในการทำงาน

2) ปัญหา โปรแกรม Potel 99SE ไม่ถูกลิขสิทธิ์
แนวทางการแก้ไข จัดหาโปรแกรมที่ถูกลิขสิทธิ์มาใช้งาน

3) ปัญหา โปรแกรม Xilinx Foundation 2.1I ไม่สมบูรณ์
แนวทางการแก้ไข ติดต่ออาจารย์ที่ปรึกษาเพื่อขอโปรแกรมที่สมบูรณ์มาใช้

5.3 แนวทางการพัฒนา

1) ในปริญญานิพนธ์นี้ยังมีส่วนที่จะพัฒนาต่อไปได้ก็คือในส่วนของฮาร์ดแวร์ซึ่งยังสามารถเพิ่มโมดูลการทำงานให้หลากหลายเพิ่มได้อีก

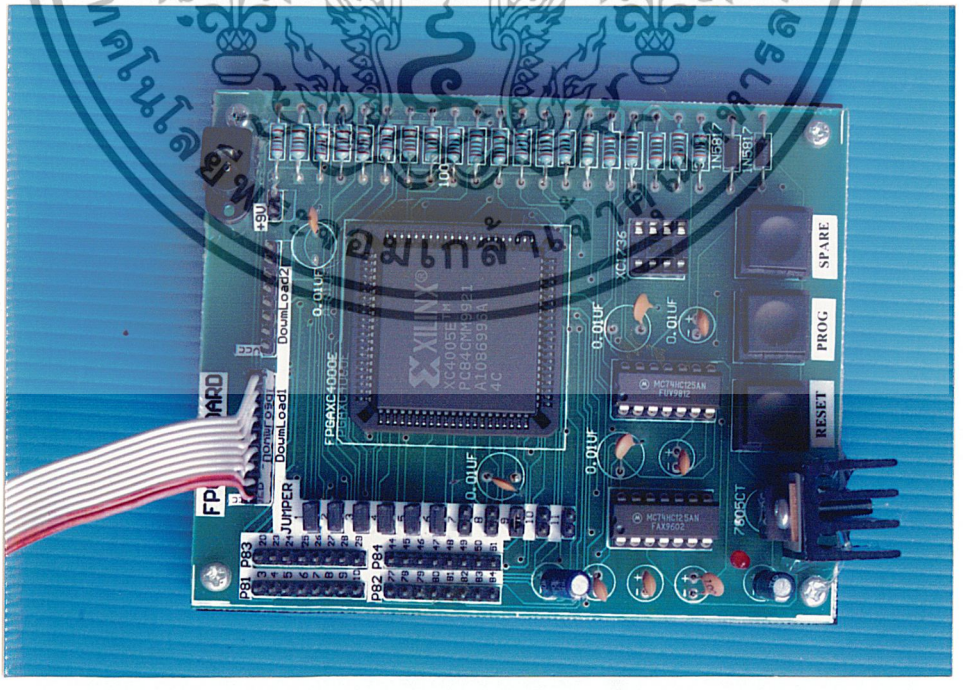
2) ในส่วนของใบงานซึ่งซอฟต์แวร์หรือโปรแกรมภาษาวีเอชดีแอลนั้นสามารถที่จะออกแบบให้มีความซับซ้อนมากขึ้นเพื่อให้เกิดการพัฒนาทางด้านทักษะต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เครื่องต้นแบบ

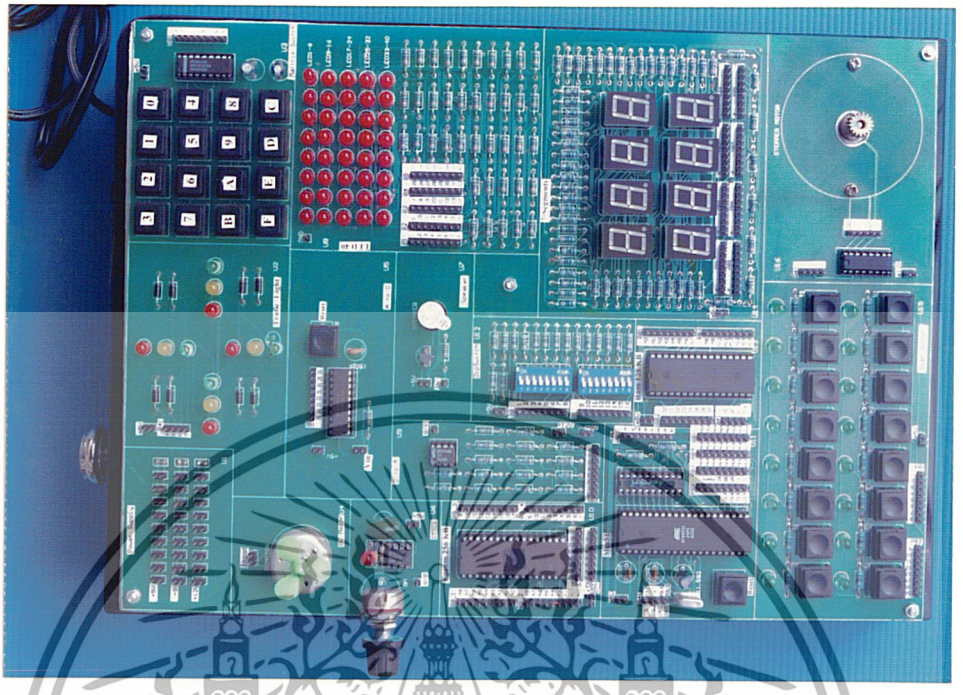
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.1 แผงวงจรซีพีแอลดี



รูปที่ ก.2 แผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 แผงวงจรอินพุต/เอาต์พุต

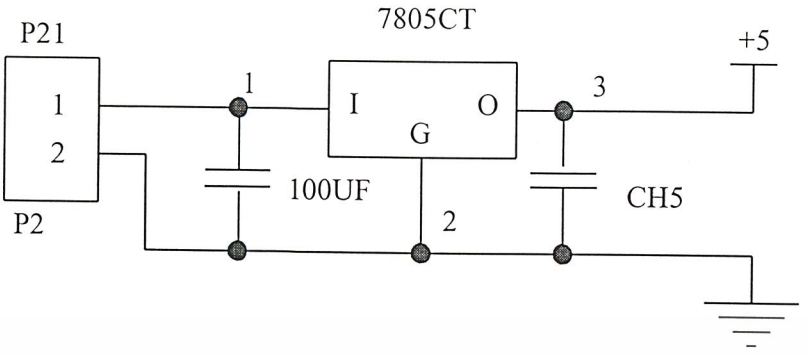
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

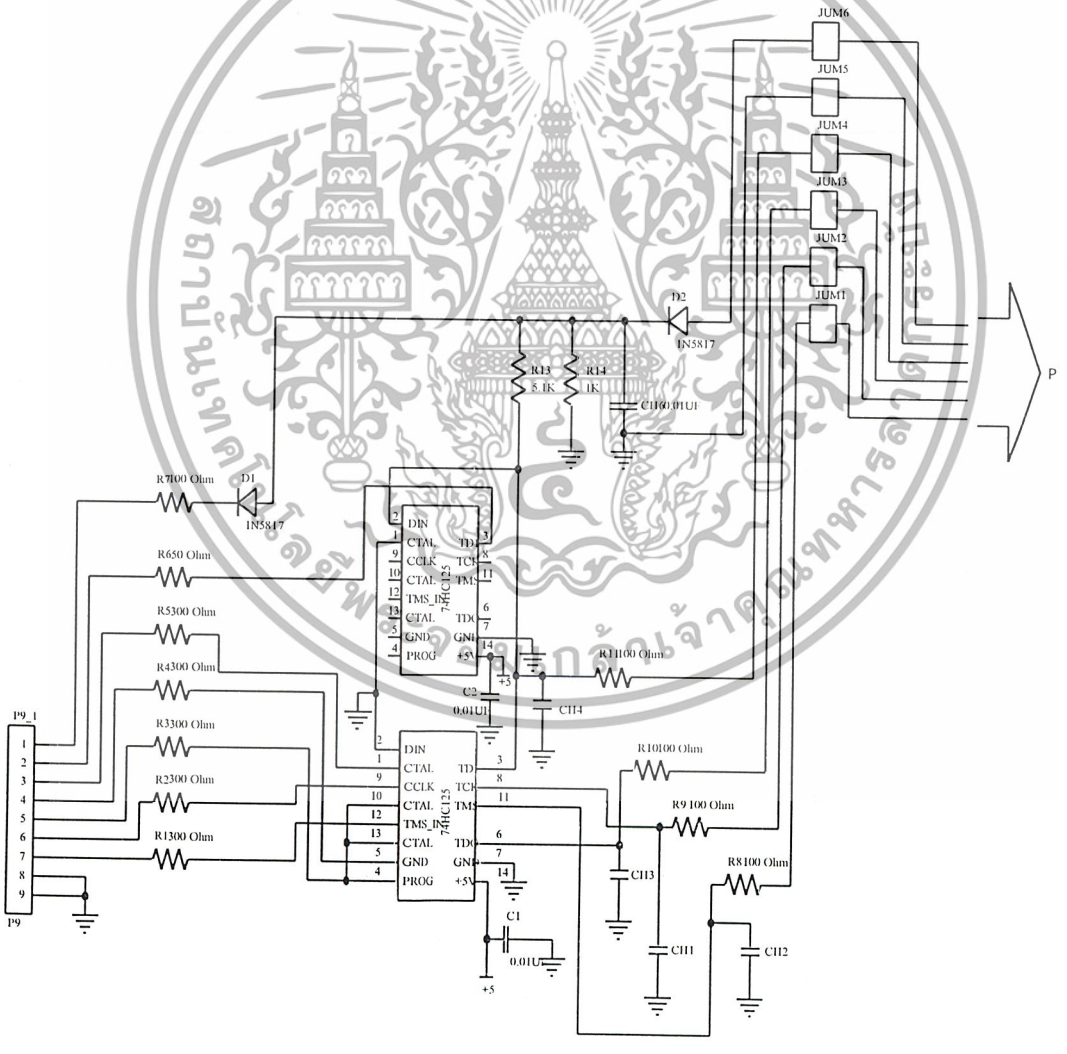
ภาคผนวก ข

วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

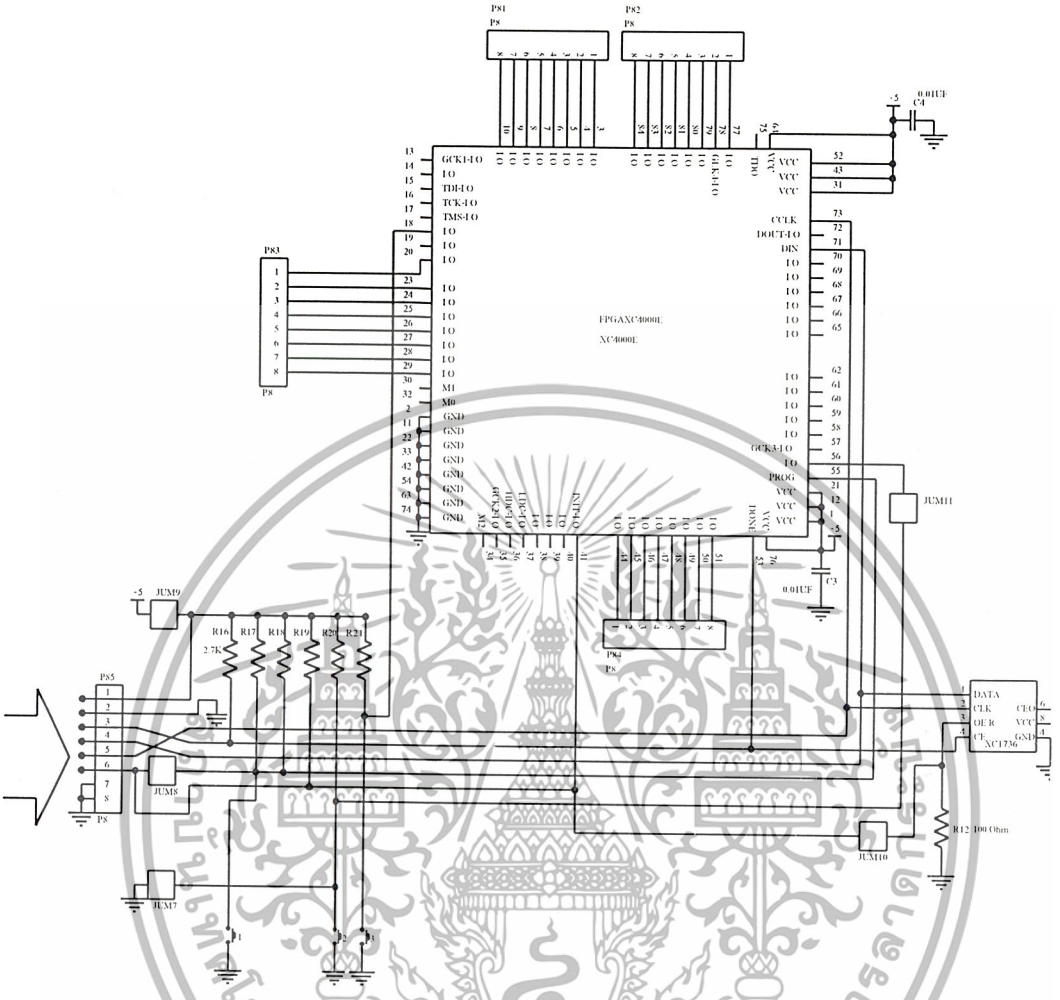


รูปที่ ข.1 วงจรรักษาระดับแรงดันไฟฟ้า 5 โวลต์ ของแผงวงจรเอฟพีจีเอ



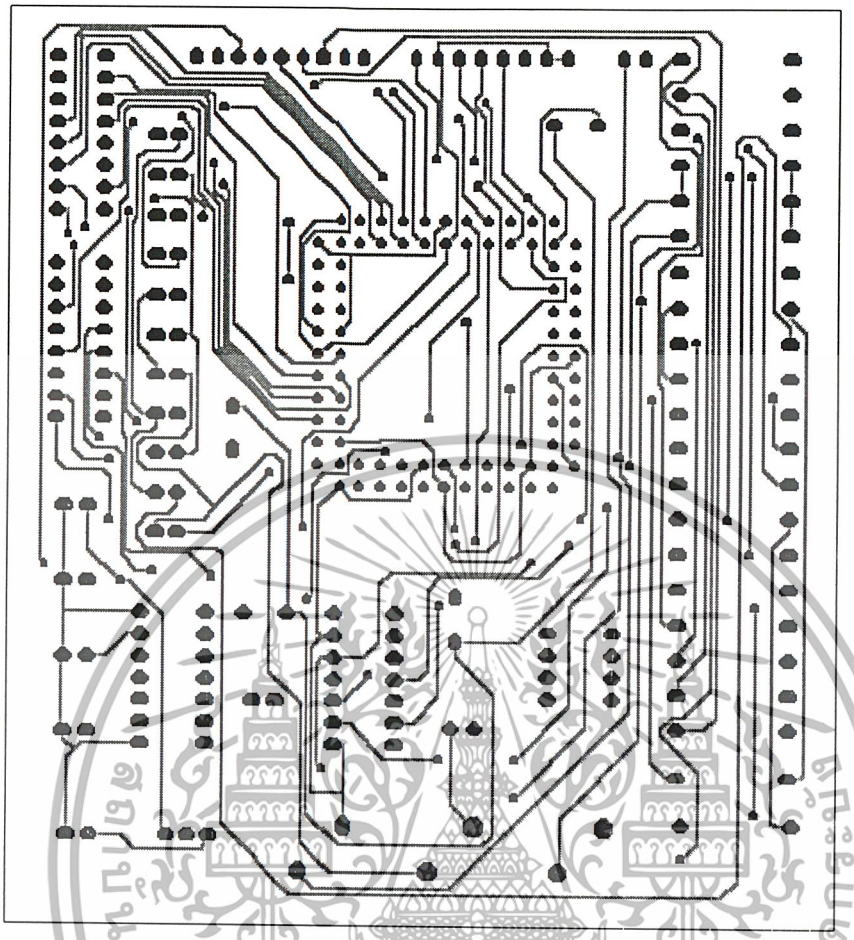
รูปที่ ข.2 วงจรของแผงวงจรเอฟพีจีเอ ส่วนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



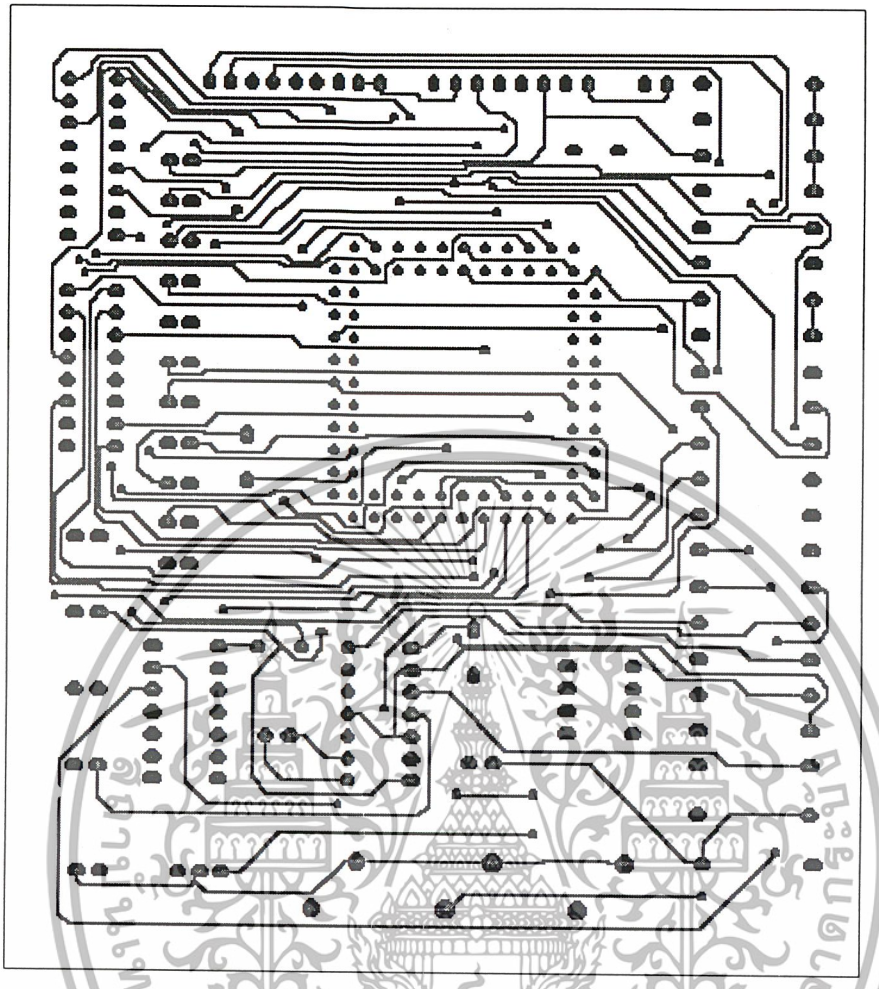
รูปที่ ข.3 วงจรของแผงวงจรเอฟพีจีเอส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



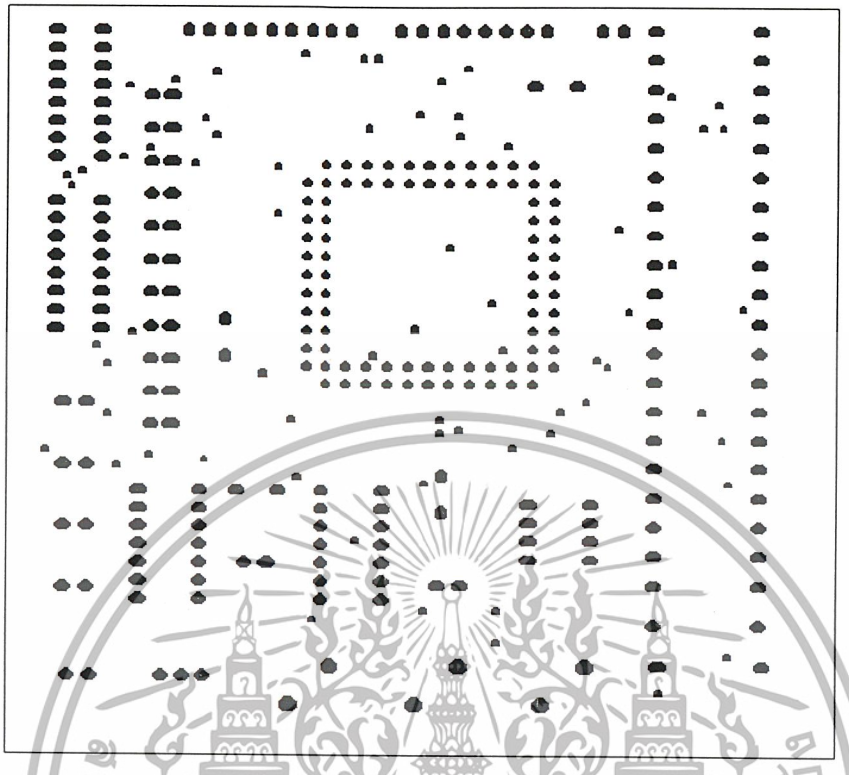
รูปที่ ข.4 แผ่นวงจรพิมพ์ ด้านล่างของแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

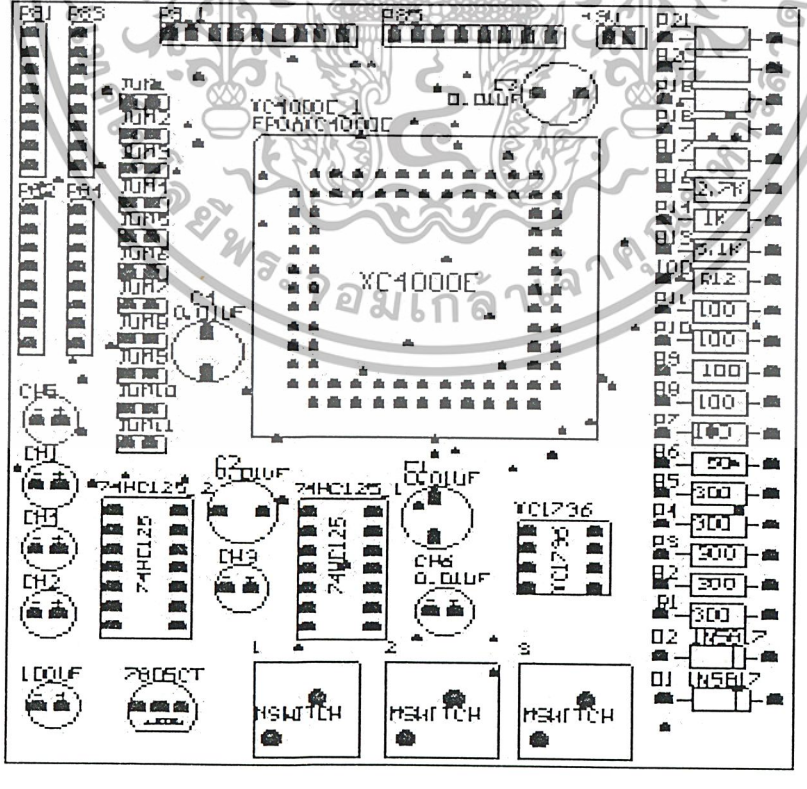


รูปที่ ข.5 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

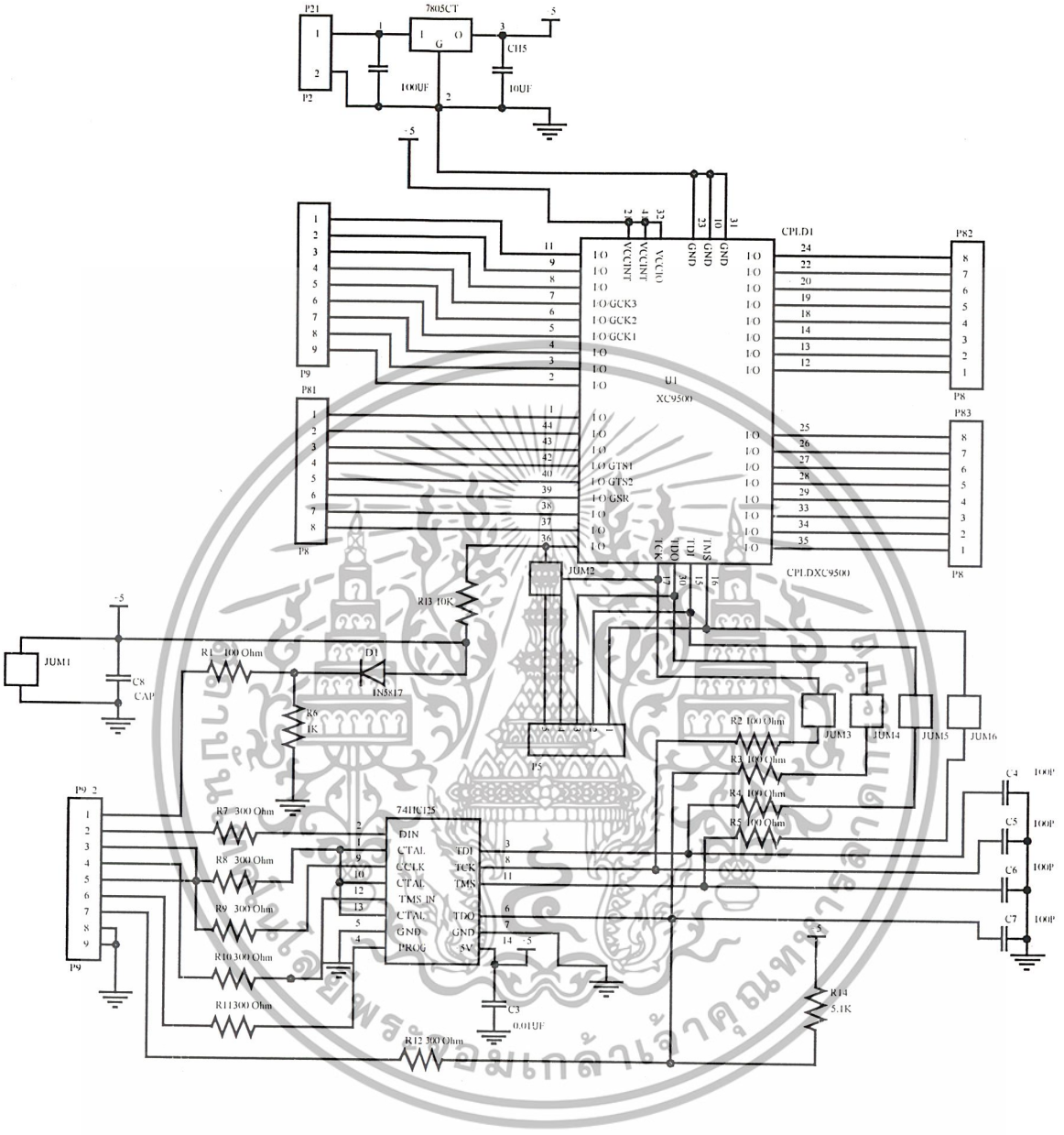


รูปที่ ข.6 แผ่นวงจรพิมพ์ของจุดการลงอุปกรณ์ของแผงวงจรเอฟพีจีเอ



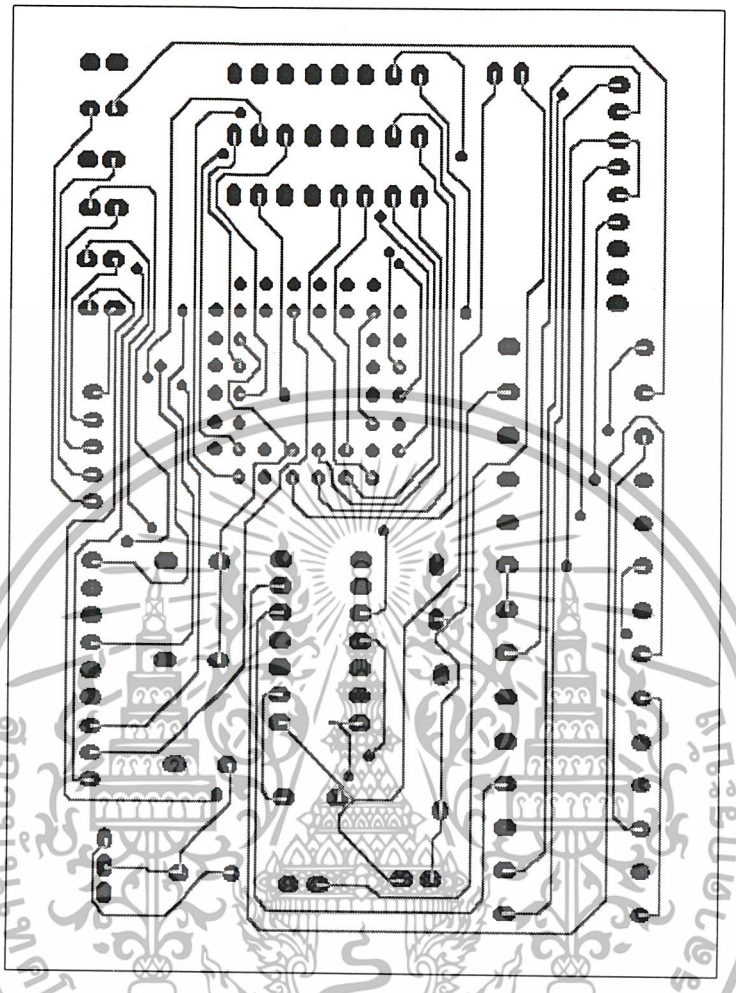
รูปที่ ข.7 ตำแหน่งการวางอุปกรณ์ของแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



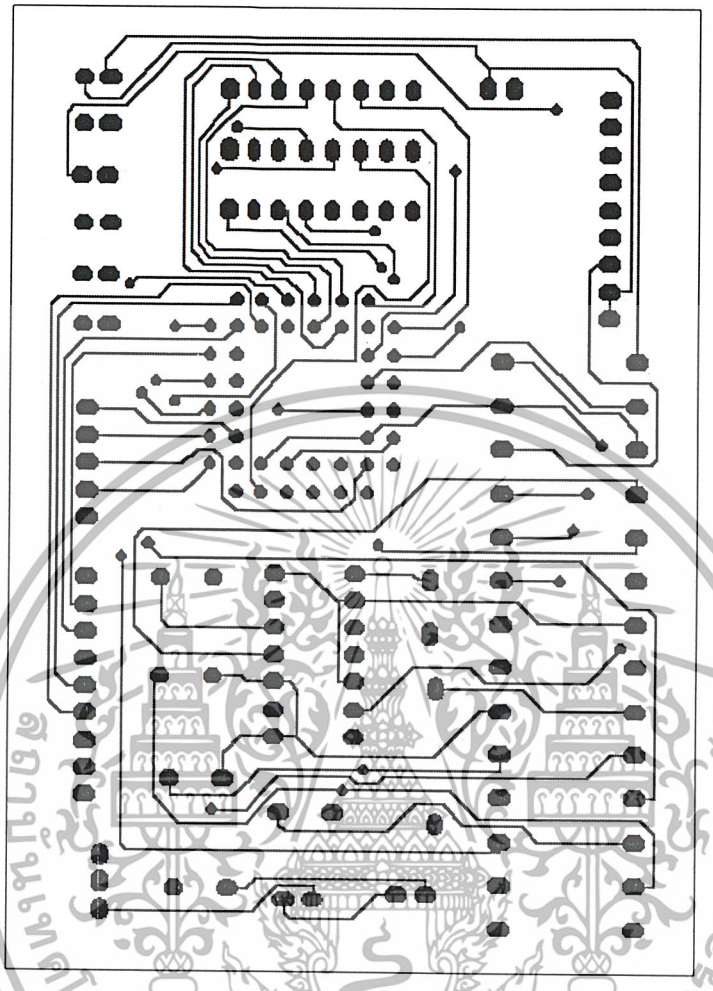
รูปที่ ข.8 วงจรของแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



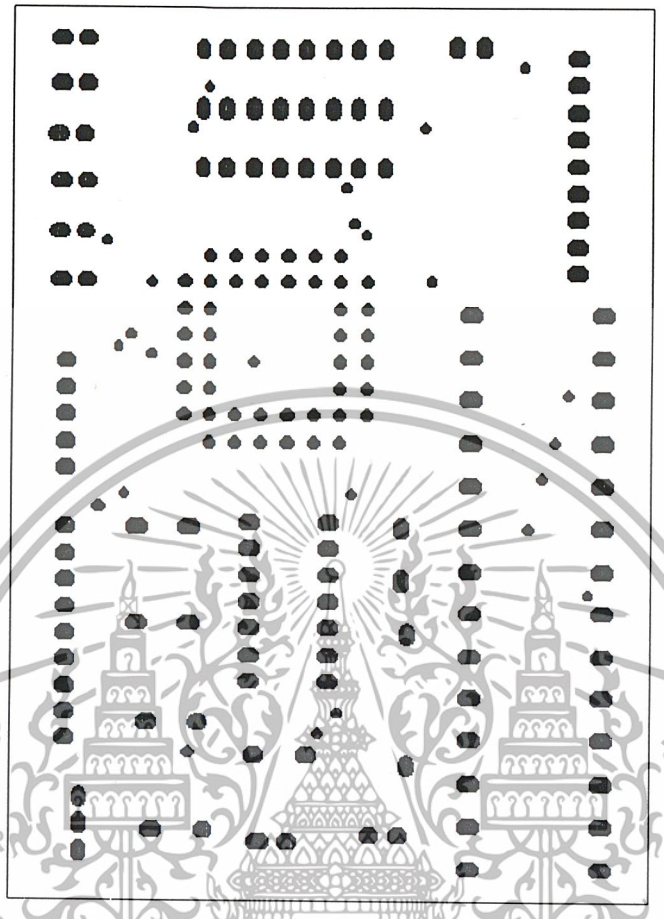
รูปที่ ข.9 แผ่นวงจรพิมพ์ด้านล่างของแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



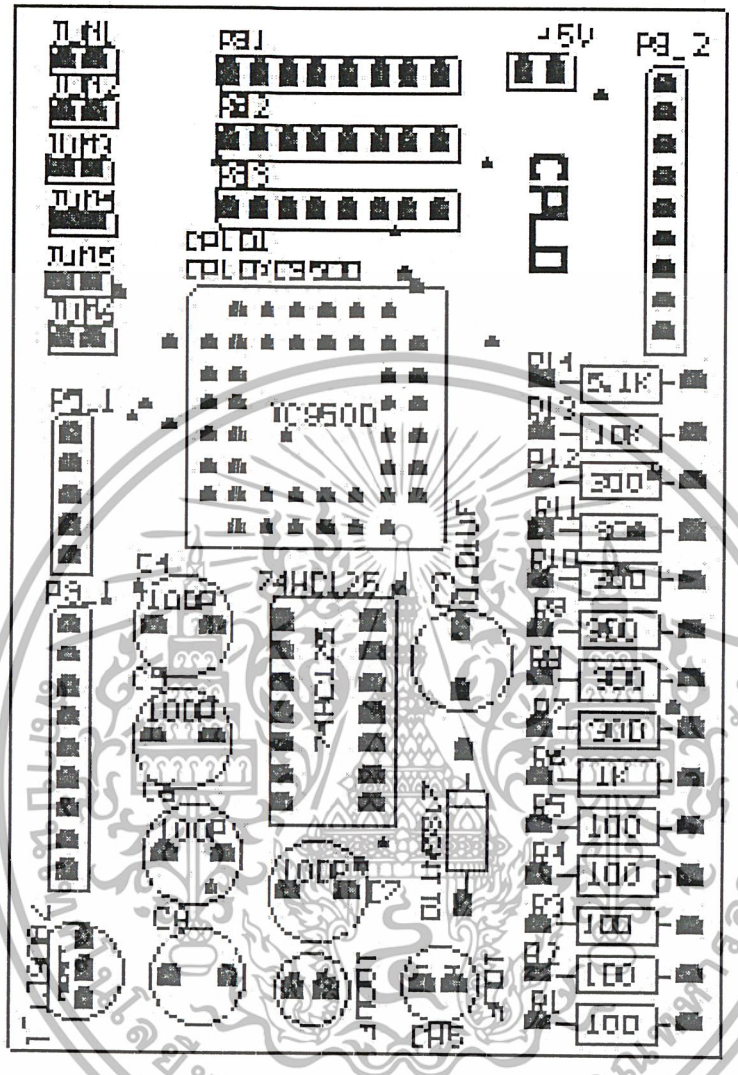
รูปที่ ข.10 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



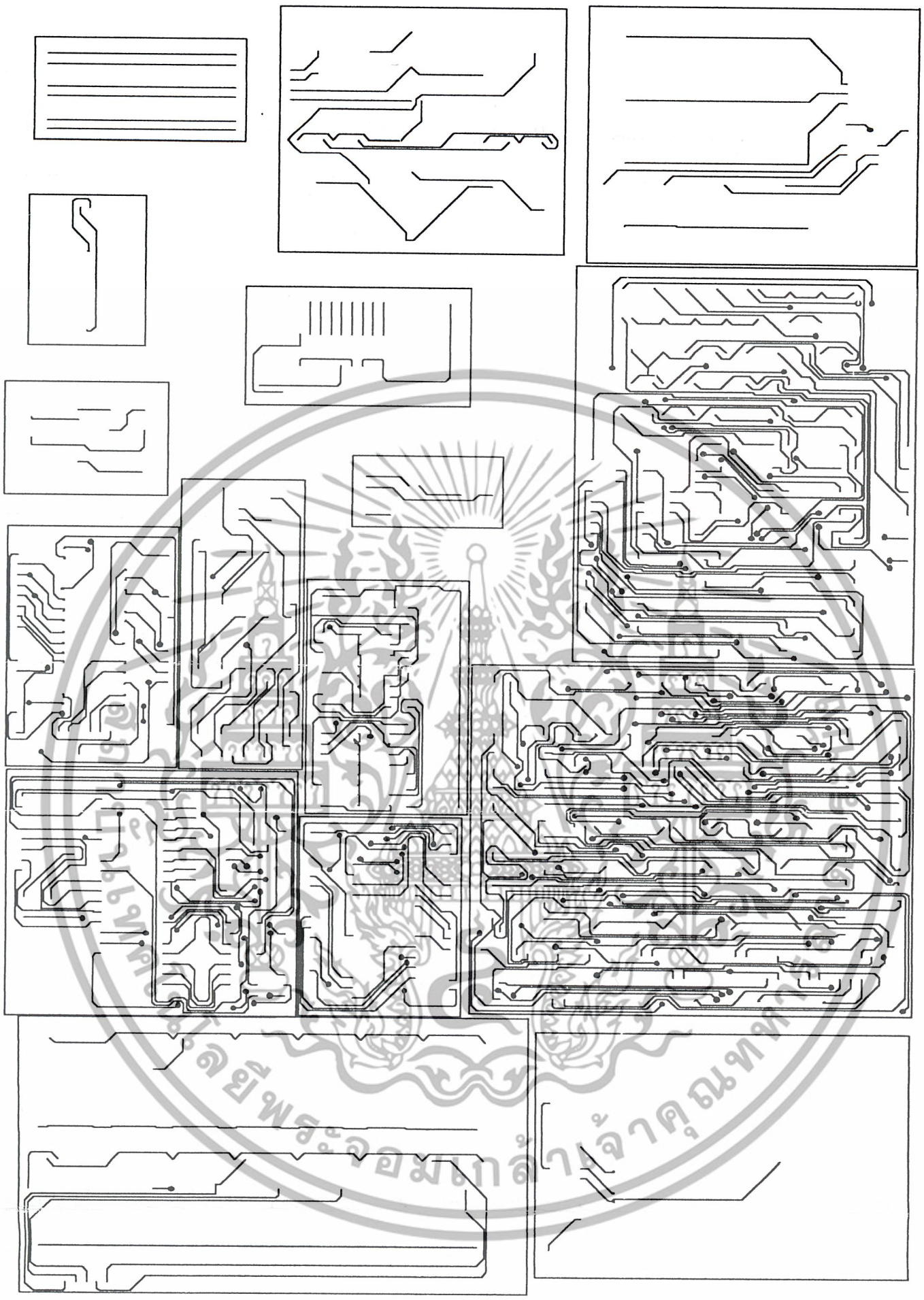
รูปที่ ข.11 แผ่นวงจรพิมพ์ของจุดการลงอุปกรณ์ของแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



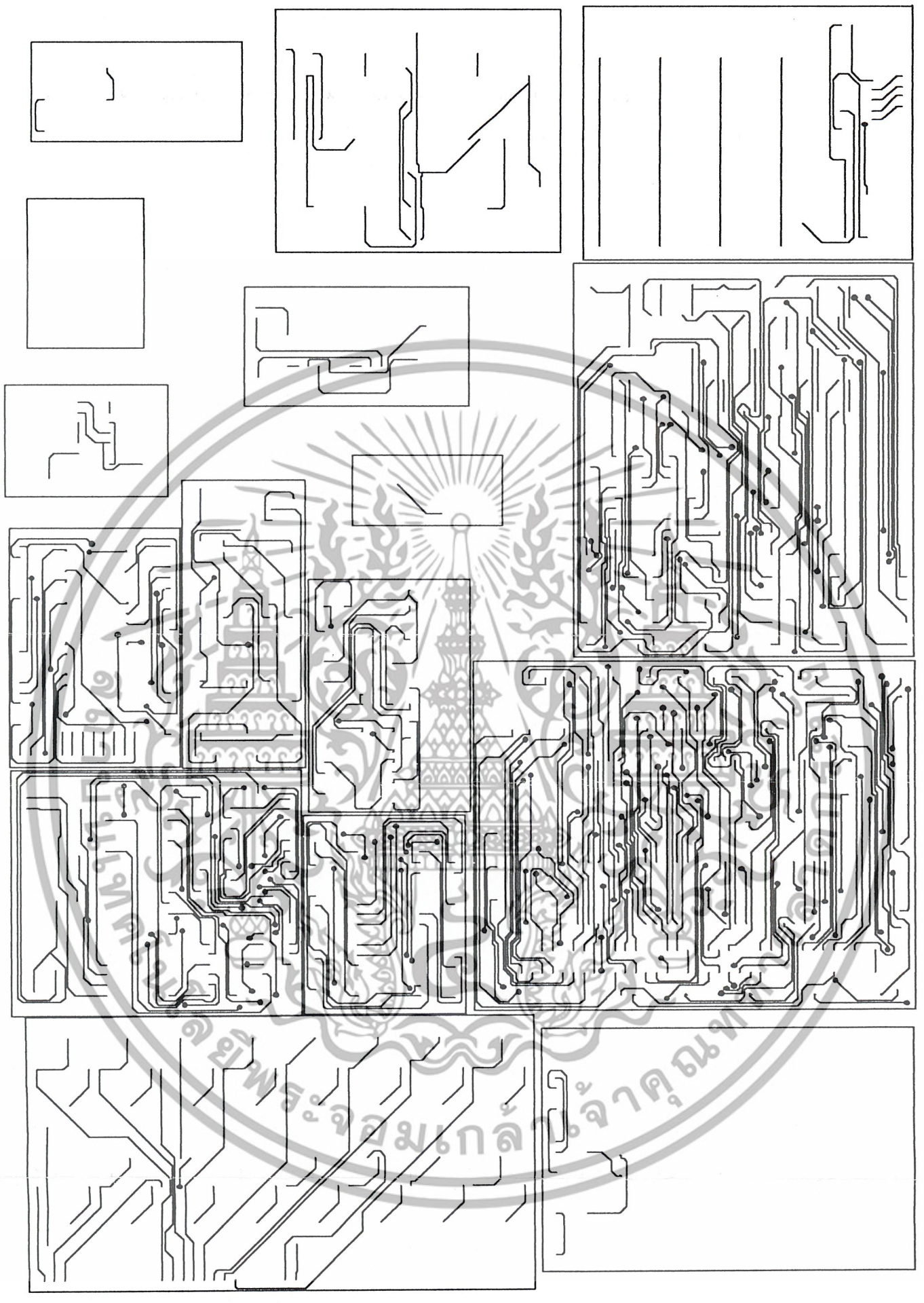
รูปที่ ข.12 ตำแหน่งการวางอุปกรณ์ของแผงวงจรซีพีแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



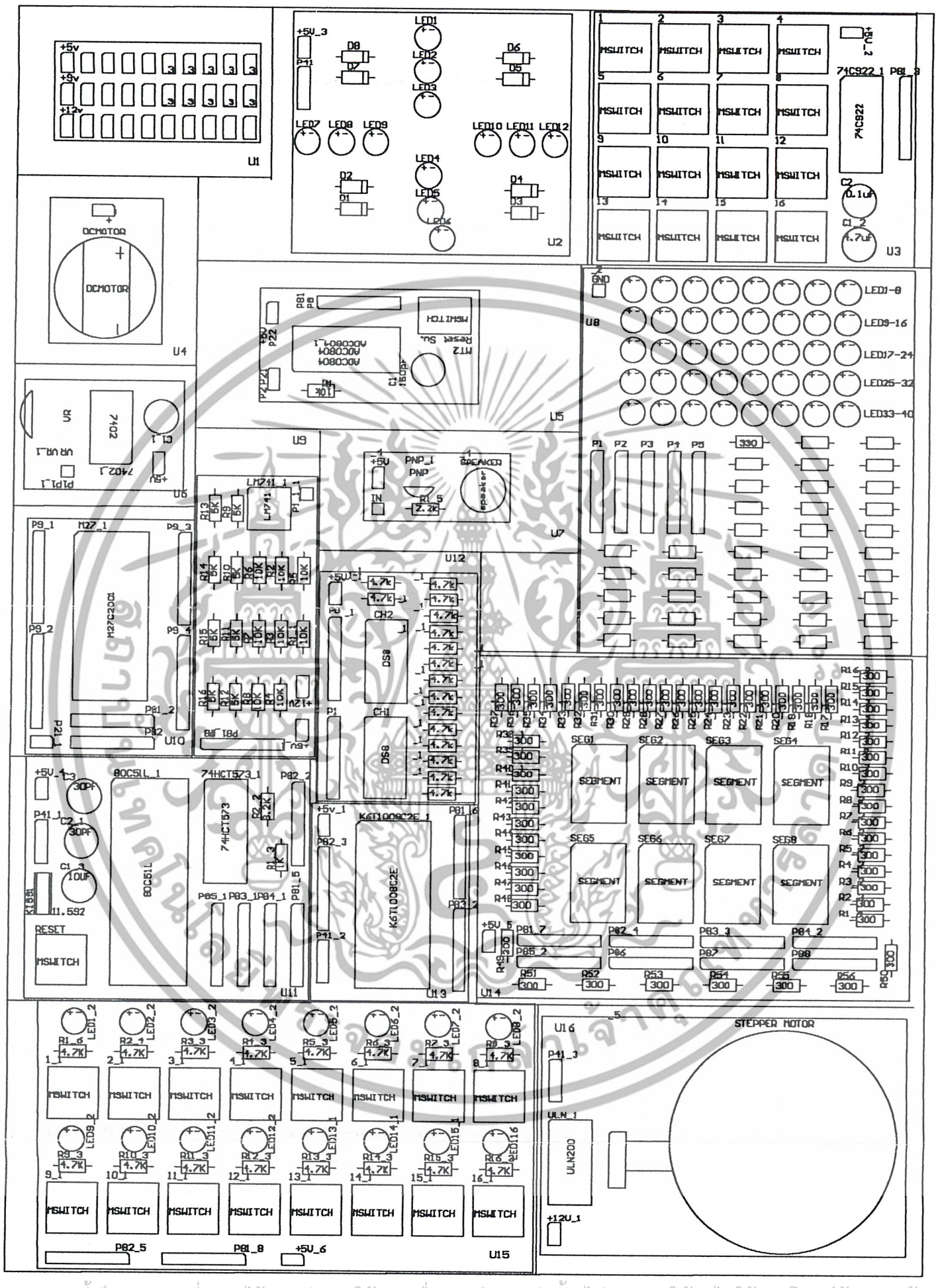
รูปที่ ข.13 แผ่นวงจรพิมพ์ด้านบนของแผงวงจรอินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.13 แผ่นวงจรพิมพ์ด้านล่างของแผงวงจรอินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.14 ตำแหน่งอุปกรณ์ของแผงวงจรอินพุต/เอาต์พุต

ภาคผนวก ค
ใบงานประกอบการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1

การศึกษาโปรแกรม Xilinx Foundation F2.1I

วัตถุประสงค์

1) เพื่อให้ผู้เรียนสามารถอธิบายถึงการสร้างโปรเจคของโปรแกรม Xilinx Foundation F2.1I ได้

2) เพื่อให้ผู้เรียนสามารถอธิบายการใช้ HDL Editor ได้

3) เพื่อให้ผู้เรียนสามารถอธิบายการส่งไฟล์เข้าสู่โปรเจคได้

4) เพื่อให้ผู้เรียนสามารถอธิบายการ Check Syntax ได้

5) เพื่อให้ผู้เรียนสามารถอธิบายการ Synthesis ได้

6) เพื่อให้ผู้เรียนสามารถอธิบายการ Implementation ได้

7) เพื่อให้ผู้เรียนสามารถอธิบายการ Programming ได้

8) เพื่อให้ผู้เรียนสามารถอธิบายการ Simulator ได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์

2) โปรแกรม Xilinx Foundation F2.1I ที่ติดตั้งเรียบร้อยแล้ว

3) แผงวงจรเอฟพีจีเอ

4) สายดาวน์โหลด

5) สายต่อวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

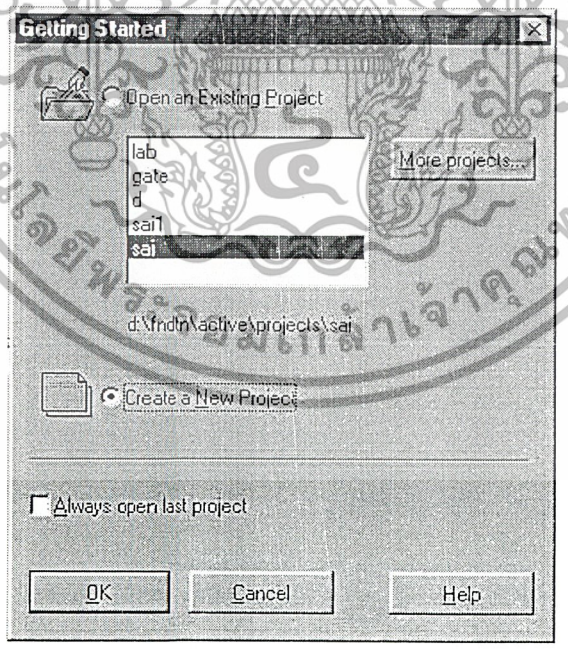
ทฤษฎีเบื้องต้น

การใช้โปรแกรม Foundation F2.1I ของบริษัทไซลิงค์ Xilinx มีขั้นตอนดังนี้

ขั้นที่ 1 ดับเบิ้ลคลิกที่ไอคอนของโปรแกรม Project Manager ดังรูปที่ ค.1

รูปที่ ค.1 Icon Project Manager

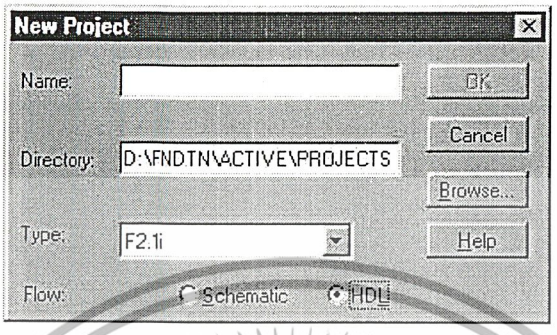
ขั้นที่ 2 เมื่อเข้ามาสู่โปแกรมเรียบร้อยแล้ว โปรแกรมก็จะแสดงหน้าต่างดังรูปที่ ค.2 ขึ้นมาโดยมีให้เลือก Open an Existing Project คือเปิดโปรเจคเดิมที่มีอยู่ในช่องรายการ หรือ Create a New Project คือสร้างโปรเจคใหม่ เมื่อเลือกแล้วให้คลิกปุ่ม OK



รูปที่ ค.2 หน้าต่าง Getting Started

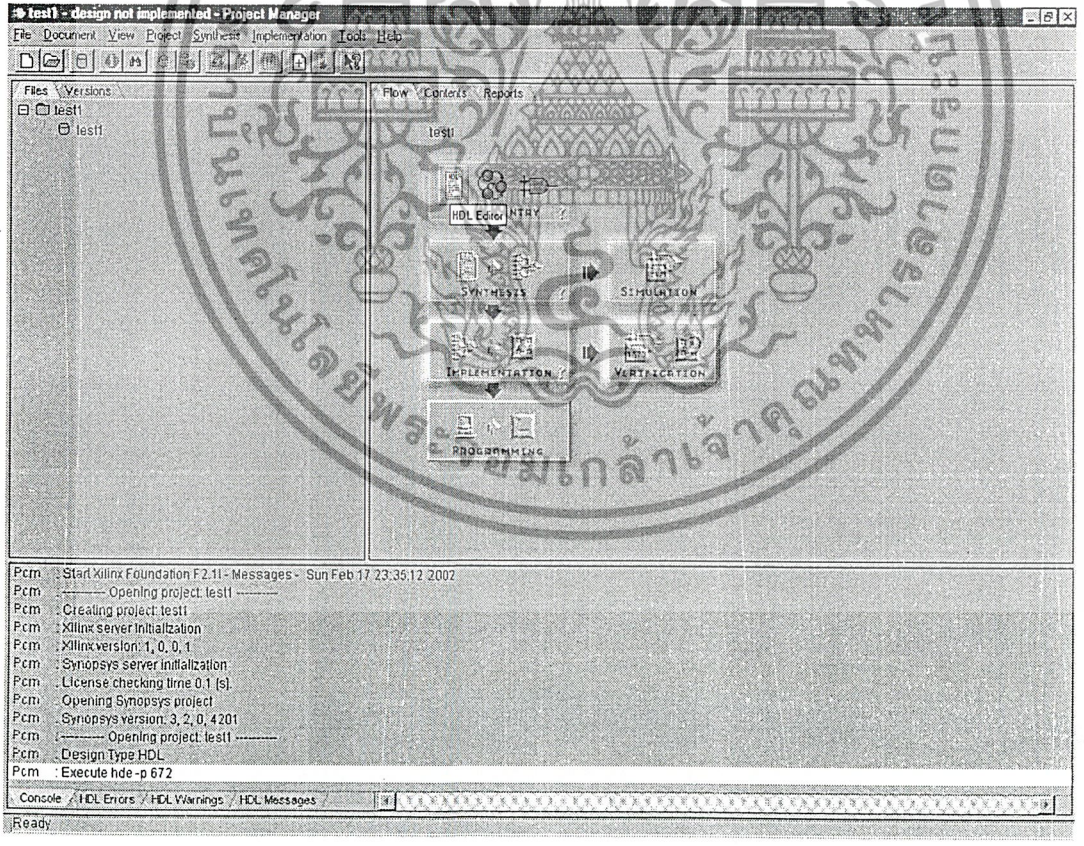
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 จากนั้นโปรแกรมก็จะให้ตั้งชื่อของโปรเจคให้พิมพ์ชื่อลงไปในช่อง Name พร้อมกับเลือก Flow ไว้ที่ HDL แล้วคลิกปุ่ม OK



รูปที่ ค.3 หน้าต่าง New Project

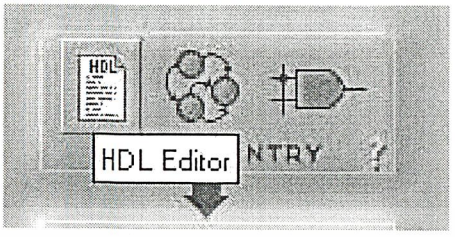
ขั้นที่ 4 โปรแกรมก็จะเข้ามาสู่หน้าต่างของการ Design ดังรูปที่ ค.4



รูปที่ ค.4 หน้าต่าง Project Manager

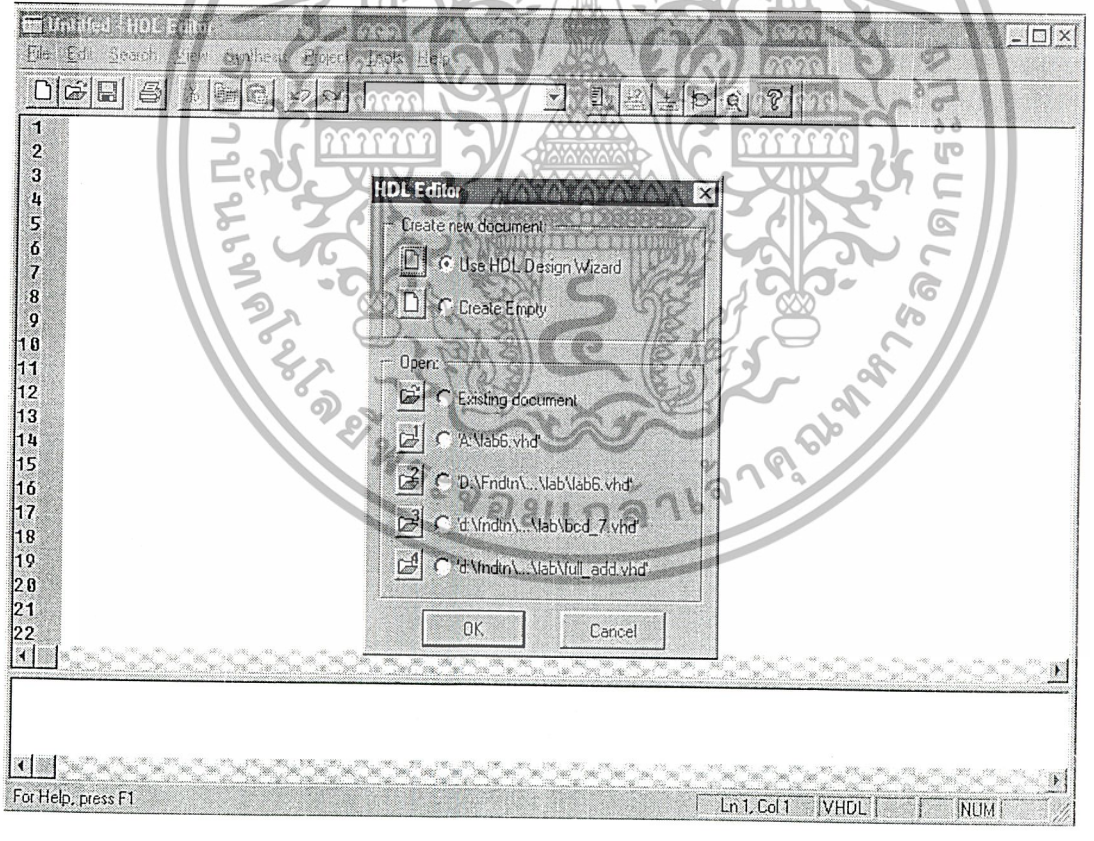
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5 จากรูปที่ ค.5 ให้คลิกที่ Design Entry แบบ HDL Editor



รูปที่ ค.5 HDL Editor

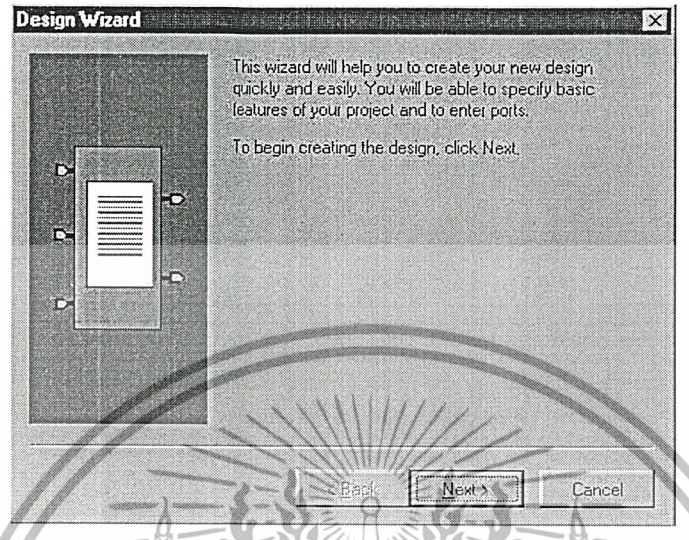
ขั้นที่ 6 โปรแกรมก็จะเข้าสู่ HDL Editor ดังรูปที่ ค.6 ให้เลือก Use HDL Design Wizard แล้วคลิกปุ่ม OK



รูปที่ ค.6 หน้าต่าง HDL Editor

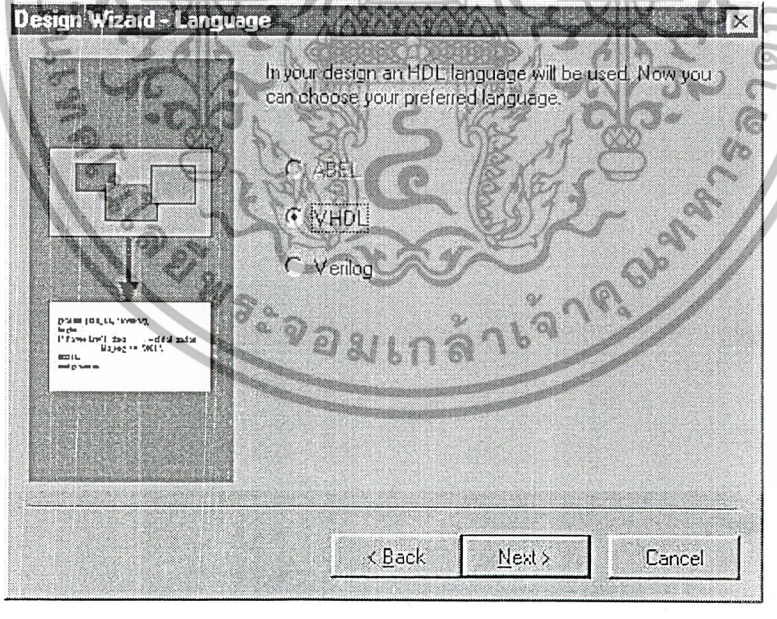
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 7 โปรแกรมก็จะเข้าสู่ Design Wizard ดังรูปที่ ค.7 ให้คลิก Next



รูปที่ ค.7 หน้าต่าง Design Wizard

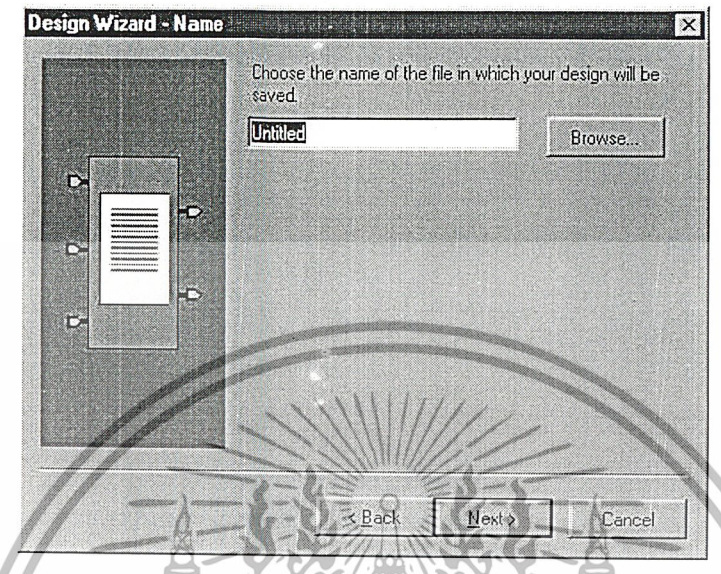
ขั้นที่ 8 ให้คลิกเลือกที่ VHDL แล้ว คลิก Next ต่อดังรูปที่ ค.8



รูปที่ ค.8 หน้าต่าง Design Wizard - Language

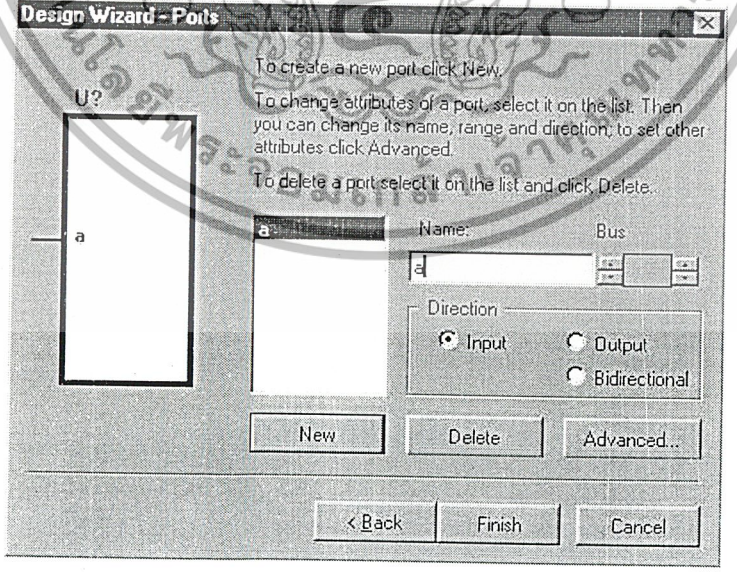
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 9 ให้พิมพ์ชื่อของตัวอุปกรณ์ที่จะสร้างขึ้นแล้วคลิก Next



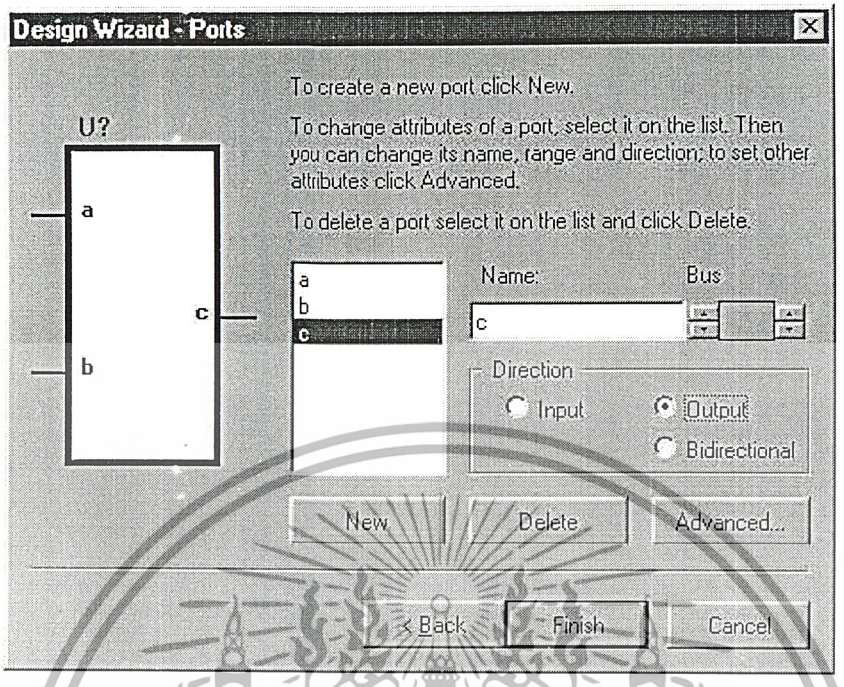
รูปที่ ค.9 หน้าต่าง Design Wizard - Name

ขั้นที่ 10 กำหนดขาของตัวอุปกรณ์โดยคลิก New แล้วเลือก Direction โดยจะให้ขาเป็น Input,Output หรือ Bidirection ได้ตามต้องการ และถ้าหากจะเพิ่มขาต่อไปก็คลิก New เมื่อกำหนดขาครบตามต้องการแล้วให้คลิก Finish จากรูปที่ ค.11 จะให้ A,B เป็นขา Input และ C เป็น Output



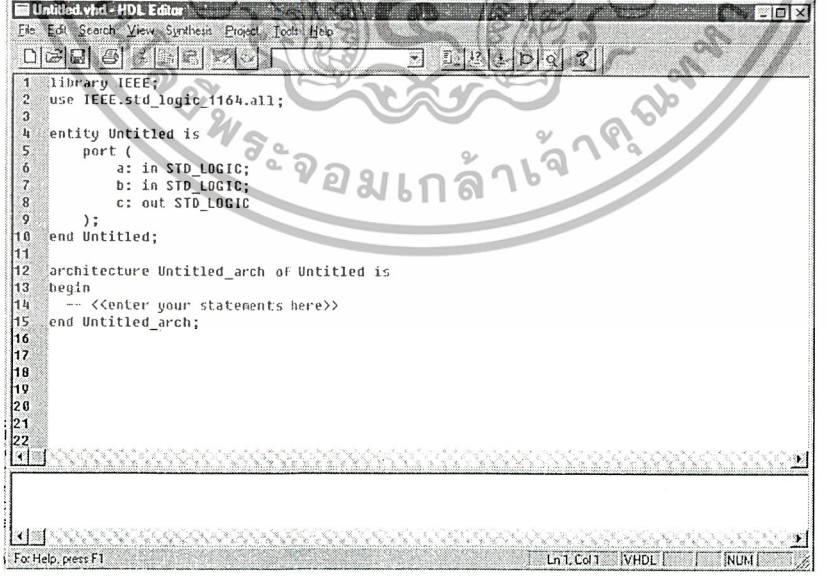
รูปที่ ค.10 หน้าต่าง Design Wizard - Ports

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



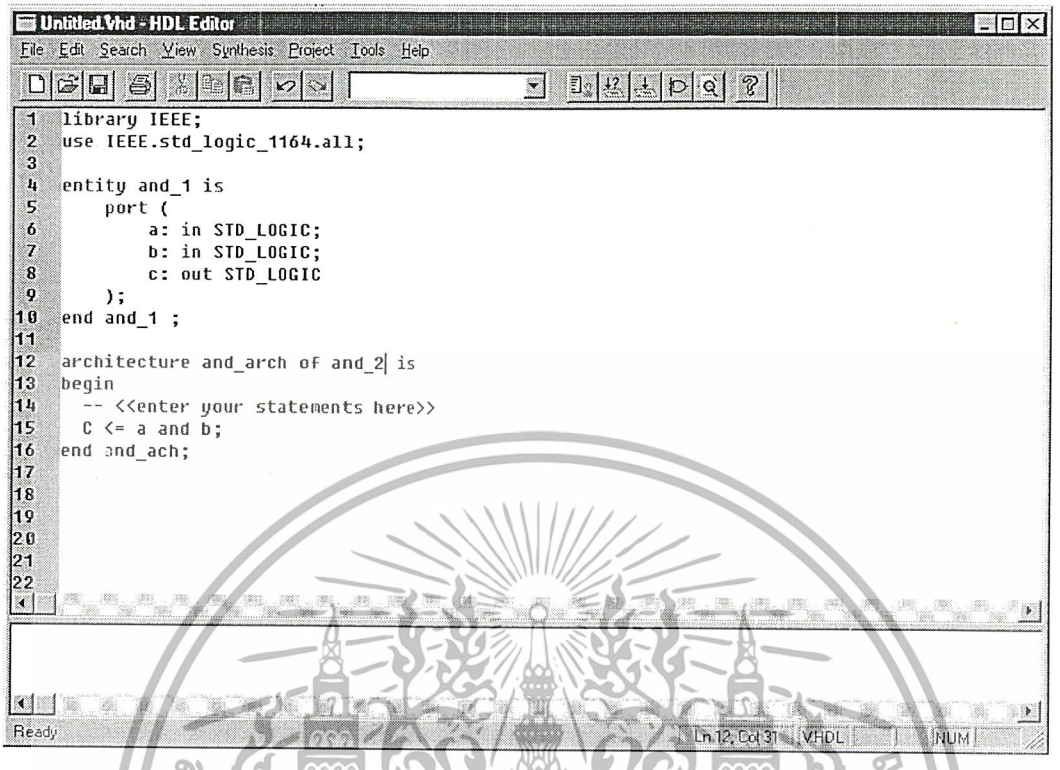
รูปที่ ค.11 หน้าต่าง Design Wizard - Ports

ขั้นที่ 11 โปรแกรมก็จะเข้ามาที่ HDL Editor ซึ่งใน Comment "--<<enter your statements here>>" จะเป็นส่วนที่เราจะเขียนสมการของการทำงานอุปกรณ์ที่เราจะสร้าง จากรูปที่ ค.13 จะเป็นตัวอย่างของ AND GATE



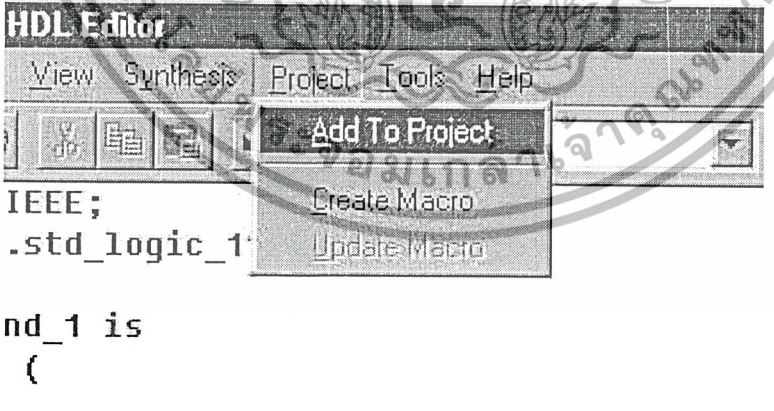
รูปที่ ค.12 หน้าต่าง HDL Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.13 หน้าต่าง HDL Editor

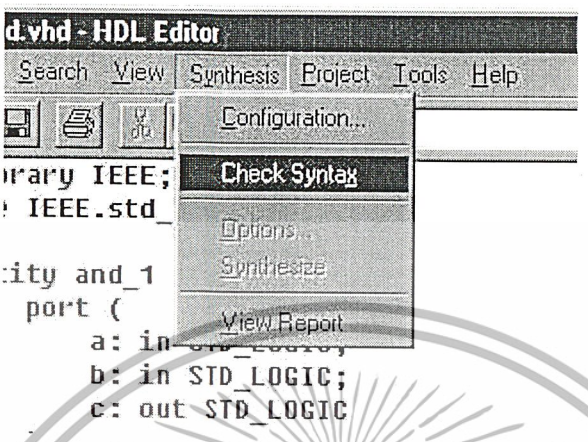
ขั้นที่ 12 ส่งไฟล์เข้าสู่ Project คลิกที่ Menu Project แล้วเลือกที่ Add To Project ดังรูปที่ ค.14



รูปที่ ค.14 Menu Add to Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 13 การ Check Syntax คลิก Menu Synthesis แล้วเลือก Check Syntax ดังรูปที่ ค.15

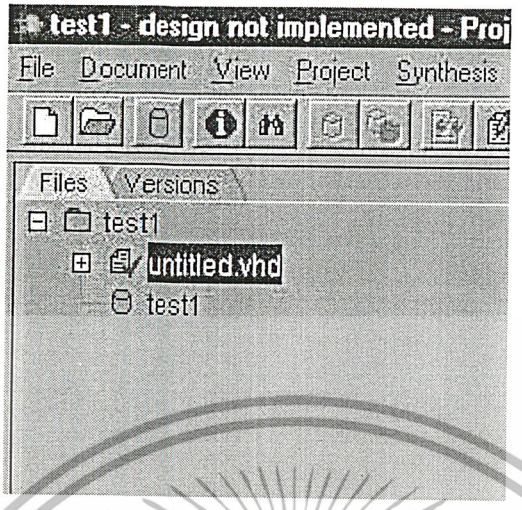


รูปที่ ค.15 Menu Check Syntax

ขั้นที่ 14 ถ้าหากว่า Check Syntax แล้วถูกต้องหมดไม่มี Error ก็จะแสดงข้อความดังรูปที่ ค.16 เมื่อดูที่ไฟล์ก็จะมีเครื่องหมายถูกเช็คอยู่ที่ไฟล์นั้น ดังรูปที่ ค.17

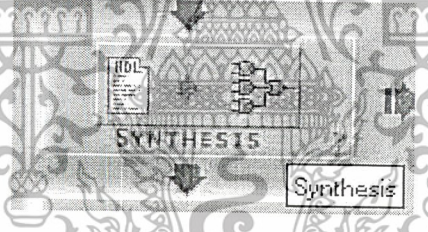
รูปที่ ค.16 หน้าต่าง Check Successful

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



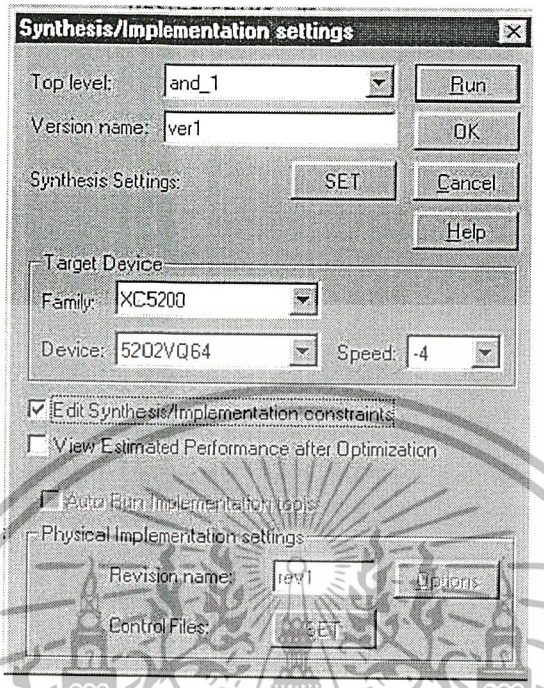
รูปที่ ค.17 หน้าต่าง Project Manager

ขั้นที่ 15 การ Synthesis คลิกที่ SYNTHESIS ดังรูปที่ ค.18 จากนั้นให้กำหนดเบอร์ เอฟพีจีเอ หรือ ซีพีเอลดี และคลิกเลือกที่ Edit Synthesis/Implementation Constraints ดังรูปที่ ค.19



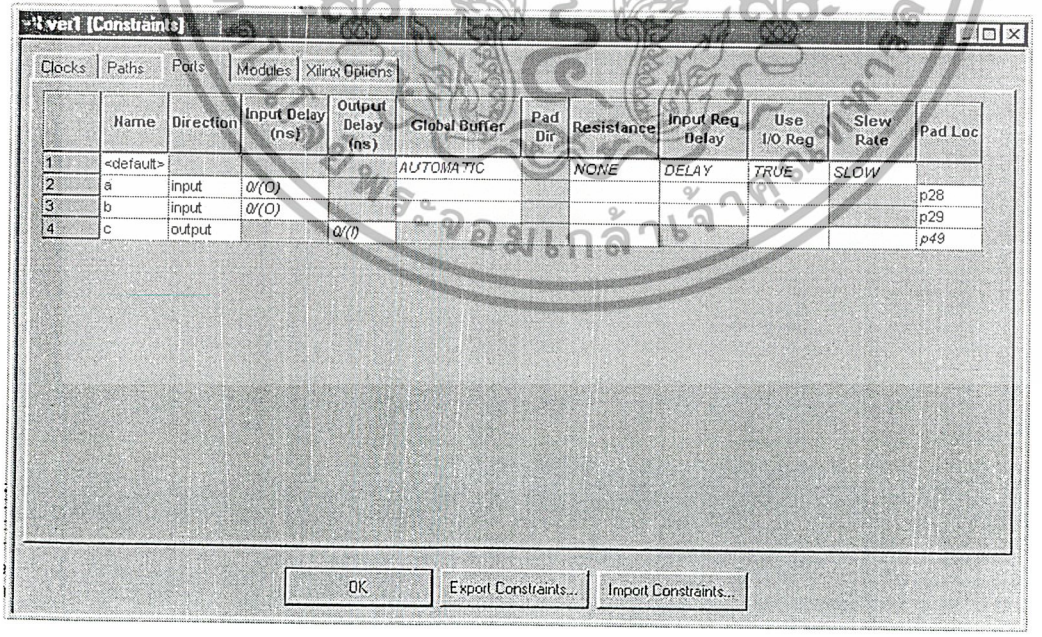
รูปที่ ค.18 Synthesis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.19 หน้าต่าง Synthesis/Implementation settings

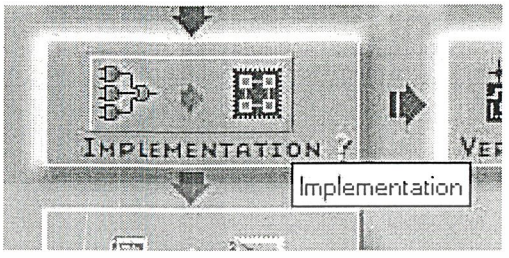
ขั้นที่ 16 การกำหนดขาที่ช่อง Pad Loc โดยจะต้องนำด้วยตัวอักษร P เช่น P28



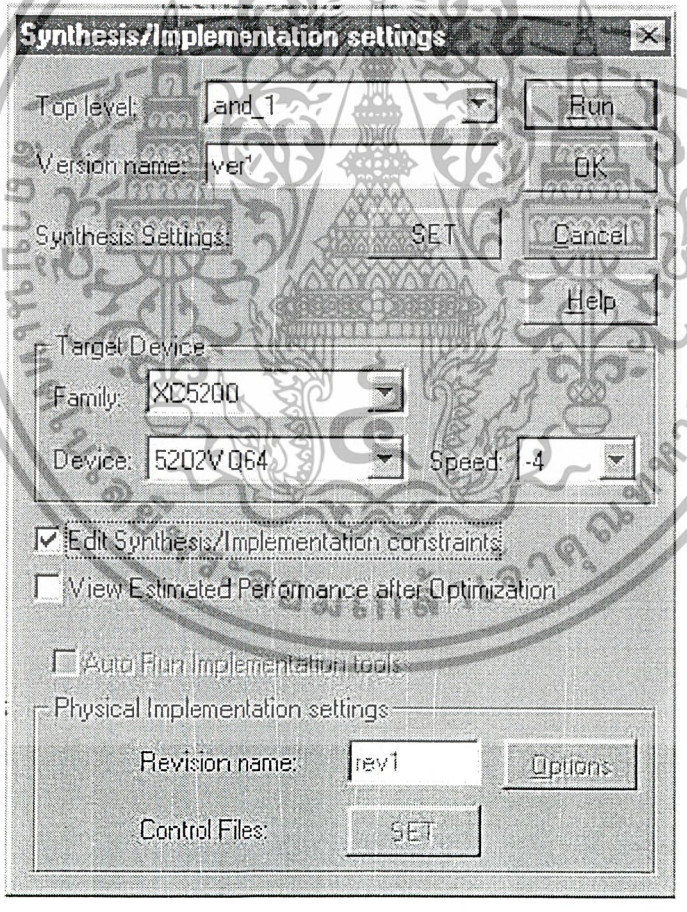
รูปที่ ค.20 หน้าต่าง Constraints

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 17 การ Implementation คลิกที่ IMPLEMENTATIION จากนั้นก็กำหนดเบอร์ของ เอฟพีจีเอ หรือ ซีพีแอลดี แล้วคลิก RUN ดังรูปที่ ค.22



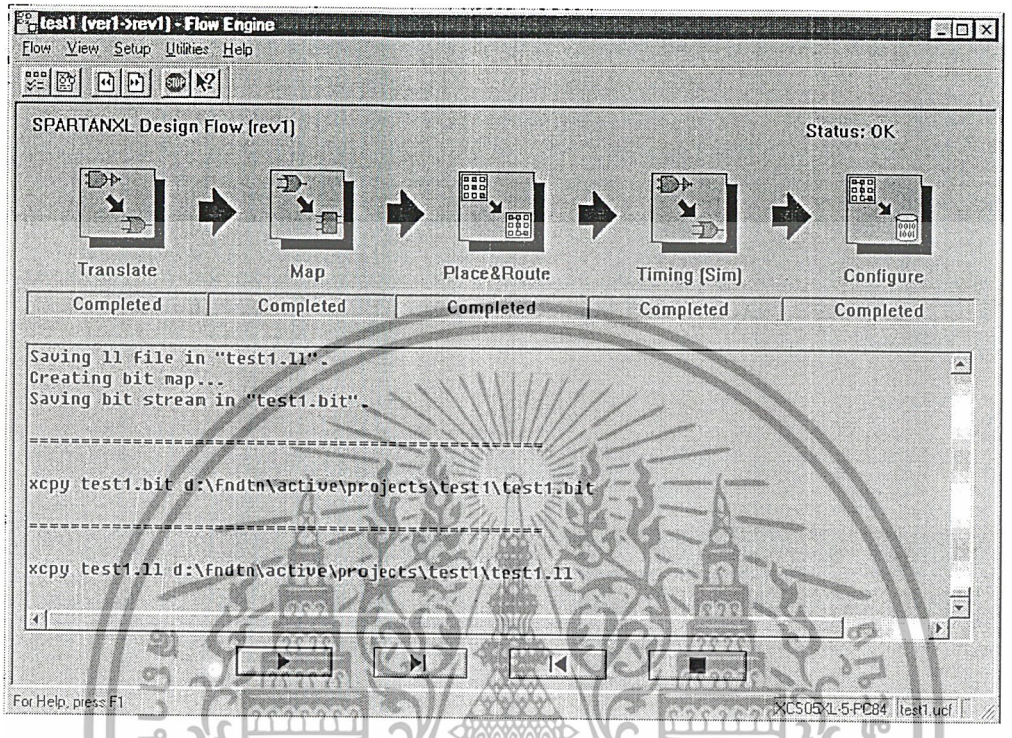
รูปที่ ค.21 Implementation



รูปที่ ค.22 หน้าต่าง Synthesis/Implementation settings

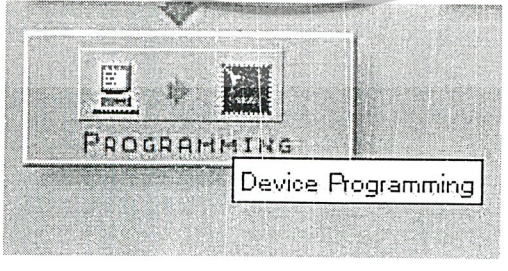
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 18 เครื่องจะทำการ Impelmentation ตามลำดับ ดังรูปที่ ค.23



รูปที่ ค.23 Flow Engine

ขั้นที่ 19 การดาว์นโหลดโดยสาย จะต้องต่อสายดาว์นโหลดเข้ากับพอร์ตของเครื่องคอมพิวเตอร์ แล้วแต่ว่าจะให้พอร์ตขนาน, อนุกรม หรือยูเอสบีเมื่อต่อสายดาว์นโหลดเรียบร้อยแล้วให้คลิกที่ Programming แล้วเลือก Hardware Debugger ดังรูปที่ ค.24,ค.25 จากรูปที่ ค.26 จะเป็น Hardware Debugger



รูปที่ ค.24 Device Programming

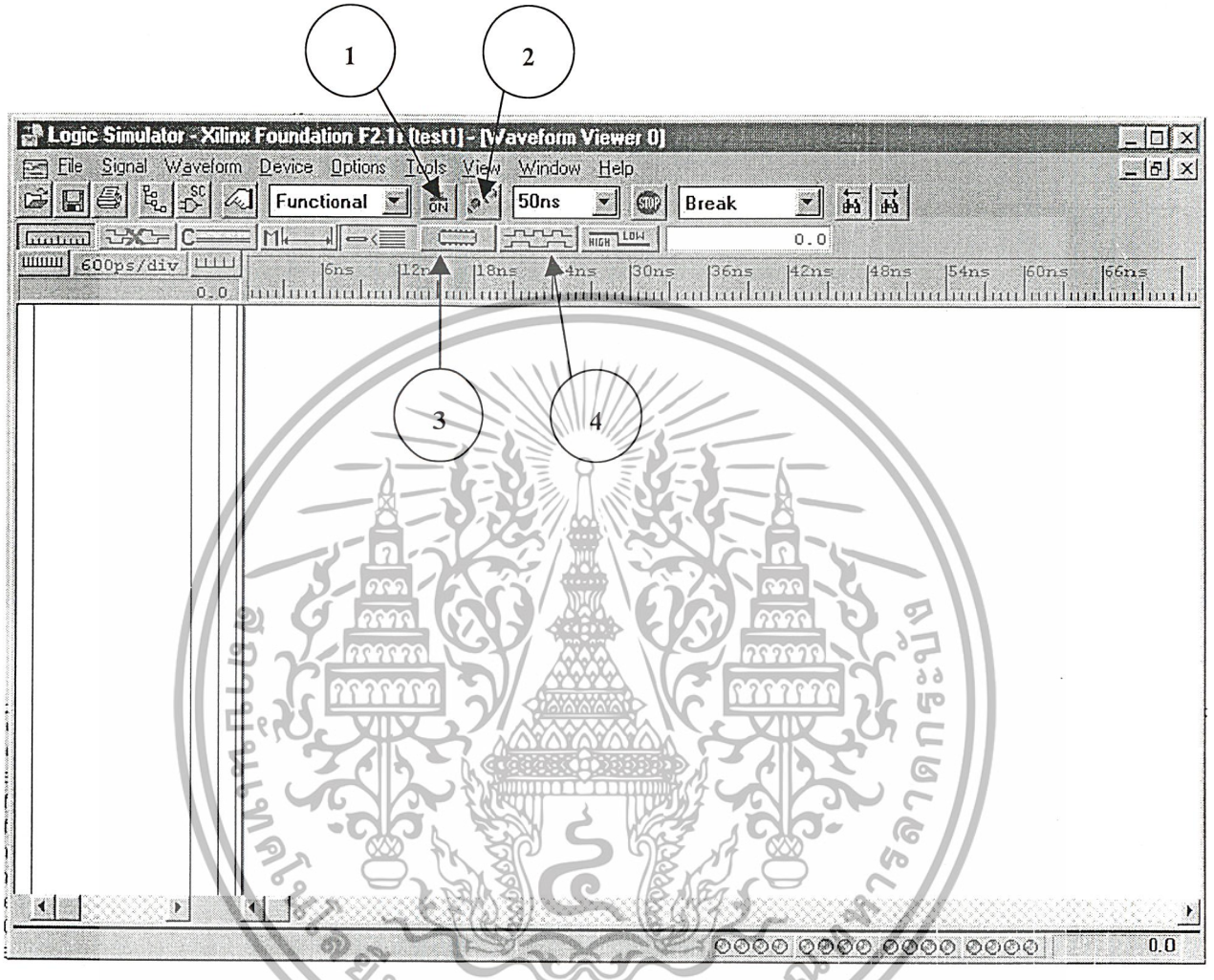
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.25 หน้าต่าง Select Program

รูปที่ ค.26 หน้าต่าง Hardware Debugger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

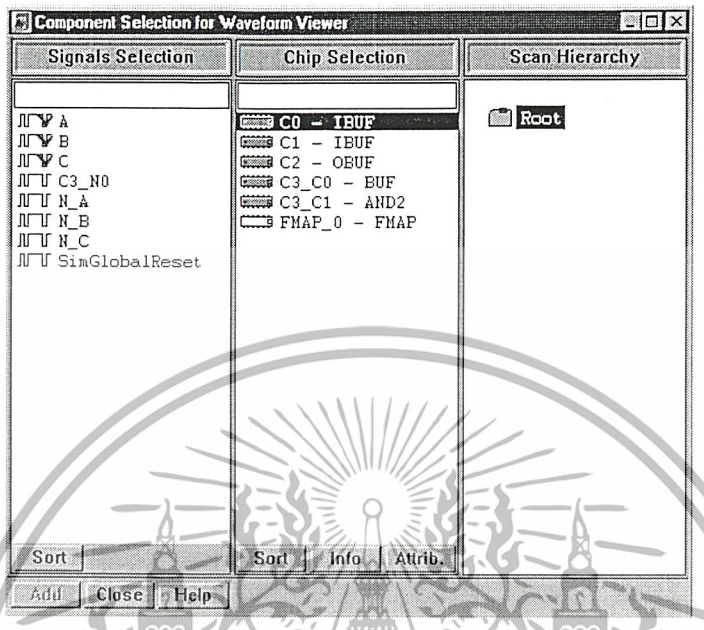
ขั้นที่ 20 การ Simulator โดยทั่วไปการเลียนแบบการทำงานจะทำหลังจากการ Synthesis



รูปที่ ค.27 หน้าต่าง Logic Simulator

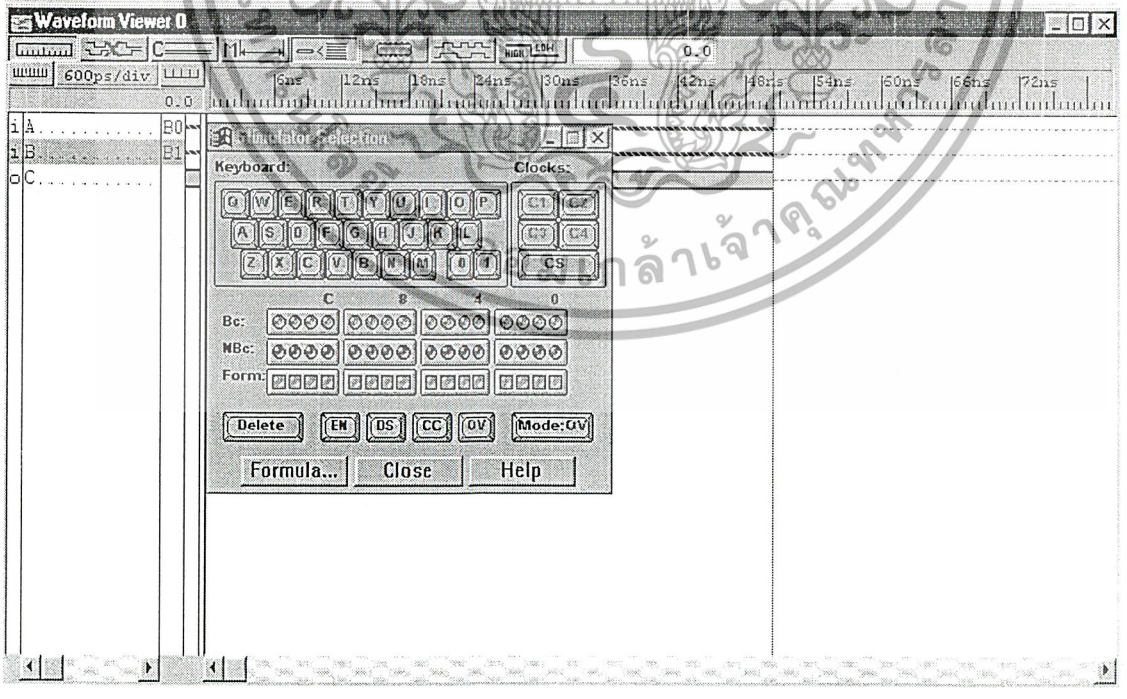
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 21 ให้คลิกที่ 3 เพื่อกำหนดขาที่จะทำการ Simutate สัญญาณในรูปที่ ค.28 จะเลือกขา A,B,C



รูปที่ ค.28 หน้าต่าง Component Selection for Wavefrom Viewer

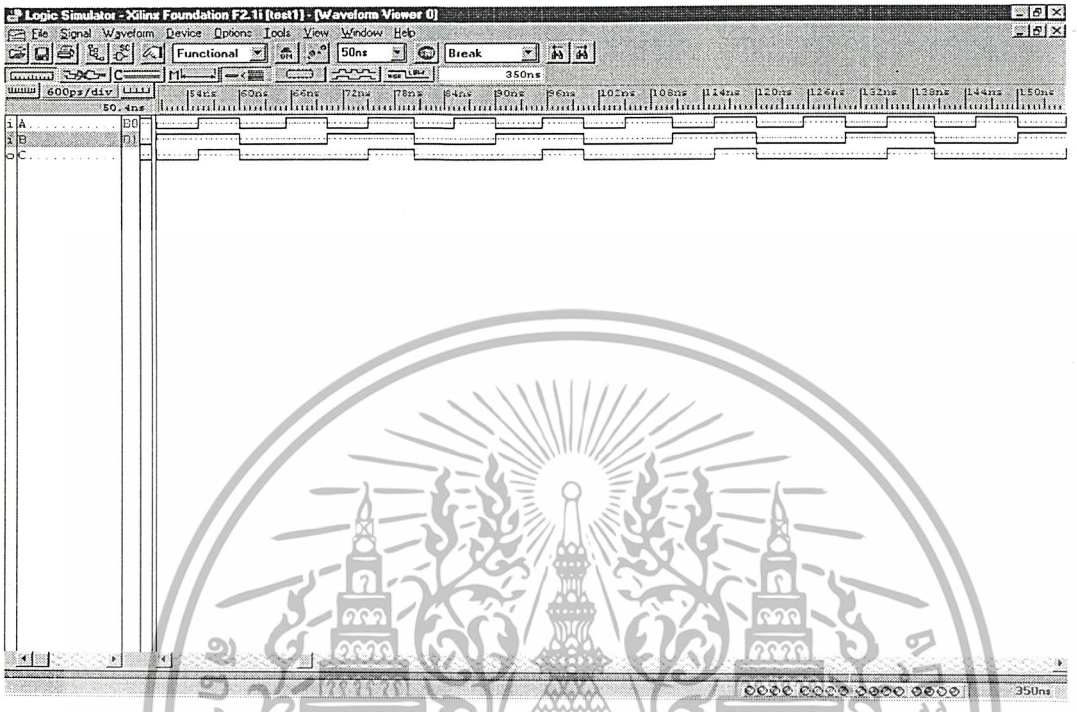
ขั้นที่ 22 คลิกที่ 4 เพื่อกำหนดสัญญาณ Clock ให้กับ ขา Input A,B โดยเลือก Bc บิทที่ 0 , 1



รูปที่ ค.29 หน้าต่าง Waveform Viewer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 23 คลิก 1,2 เพื่อ Simulate สัญญาณ Clock ก็จะแสดงผลดังรูปที่ ค.30



รูปที่ ค.30 หน้าต่าง Logic Simulator

ลำดับขั้นการทดลอง

1) เปิดโปรแกรม Xilinx Foundation F2.1I โดยคลิกที่ไอคอน Project Manager

2) สร้างโปรเจคใหม่ และตั้งชื่อโปรเจคว่า lab1

3) คลิกเลือก Flow แบบ HDL แล้วคลิก OK

4) เลือกการออกแบบ Design Entry แบบ HDL Editor

5) ให้ออกแบบวงจรแอนเกตดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library IEEE;
use IEEE.std_logic_1164.all;
entity and_1 is
port (a: in STD_LOGIC;
b: in STD_LOGIC;
c: out STD_LOGIC);
end and_1;
architecture and_arch of and_1 is
begin
c <= a and b;
end and_arch;
```

รูปที่ ค.31 โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต

1) ให้ Check Syntax ตรวจสอบความถูกต้องของโปรแกรม

2) เมื่อ Check Syntax ถูกต้องเรียบร้อยแล้ว ให้ส่งไฟล์เข้าสู่โปรเจคโดยคลิกที่เมนู Project แล้วเลือก Add To Project

3) ทำการ Synthesis โดยกำหนดเบอร์ไอซีตามที่ให้มาบนแผงวงจรทดลอง

4) ทำการ Simulation เพื่อตรวจสอบการเลียนแบบการทำงาน

5) ทำการ Implementation

6) ทำการProgramming ลงในไอซีตามเบอร์ที่ให้มาบนแผงวงจรทดลอง

คำถามท้ายการทดลอง

1) การออกแบบวงจรรวมเกทโดยใช้อุปกรณ์เอฟพีจีเอ และซีพีแอลดีนั้นสามารถออกแบบได้กี่แบบ ?

2) จงอธิบายขั้นตอนการออกแบบวงจรรวมเกทที่ได้ทำการทดลองมาเป็นข้อ ๆ ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2

การใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอล ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

วัตถุประสงค์

1) เพื่อให้นักศึกษาสามารถใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี ได้

2) เพื่อให้นักศึกษาสามารถอธิบายขั้นตอนการใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี ได้

3) เพื่อให้นักศึกษาสามารถดาวน์โหลดโปรแกรมลงแผงวงจรเอฟพีจีเอ และซีพีแอลดีได้

4) เพื่อให้นักศึกษาสามารถต่อสายดาวน์โหลดได้

5) เพื่อให้นักศึกษาสามารถต่อพอร์ตอินพุต/เอาต์พุตได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์

2) สายดาวน์โหลดเอฟพีจีเอ และซีพีแอลดี

3) โปรแกรม Xilinx Foundation F2.1I ที่ติดตั้งเรียบร้อยแล้ว

4) แผงวงจรซีพีแอลดี

5) แผงวงจรเอฟพีจีเอ

6) แผงวงจรอินพุต/เอาต์พุต โมดูลมอเตอร์ไฟฟ้ากระแสตรง

7) อะแด็ปเตอร์

8) สายต่อวงจร

ทฤษฎีเบื้องต้น

ขั้นตอนการใช้งานชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

1) ต่อสายดาวน์โหลดเข้าที่ขั้วต่อ ของชุดปฏิบัติการฯ หลังจากนั้นทำการต่อปลายสายดาวน์โหลดอีกข้างหนึ่งเข้าที่พอร์ตขนานของเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

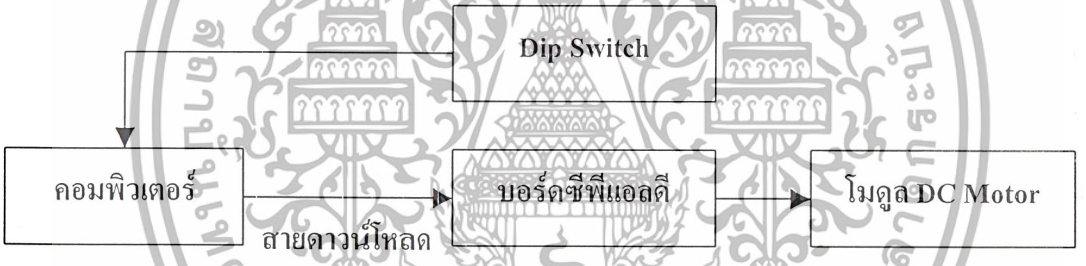
2) เสียบปลั๊กของอะแด็ปเตอร์เข้ากับแหล่งจ่ายไฟฟ้า 220 โวลต์ หลังจากนั้นให้เปิดโปรแกรม Xilinx Foundation F2.1I ในเครื่องคอมพิวเตอร์ และทำการ ดาวน์โหลด โปรแกรมที่ได้ทำการออกแบบไว้สมบูรณ์แล้ว

3) เมื่อทำการ ดาวน์โหลด เสร็จเรียบร้อยแล้ว สามารถที่จะนำไปทดลองกับแผงวงจรอินพุตเอาต์พุตในโมดูลต่างๆ ได้ตามต้องการ

4) หลังจากการใช้งานชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอและซีพีแอลดี เสร็จเรียบร้อยแล้ว ทำการปิดเครื่องคอมพิวเตอร์ และดึงปลั๊กของอะแด็ปเตอร์ ออกจากแหล่งจ่ายไฟฟ้ากระแสสลับ 220 โวลต์

การทดลองที่ 1

1) ต่อแผงวงจรทดลองดังรูปที่ ค.32



รูปที่ ค.32 วงจรการทดลองที่ 1 การใช้แผงวงจรซีพีแอลดี

2) ต่อพอร์ตจากโมดูล Dip Switch ไปเข้าที่พอร์ต P42 และ P43 ของแผงวงจรซีพีแอลดี

3) ต่อสายดาวน์โหลดจากพอร์ตขนานของเครื่องคอมพิวเตอร์เข้ากับแผงวงจรซีพีแอลดี โดยหันขั้วของสายดาวน์โหลด (แถบสีแดง) ให้ตรงกับพอร์ต VCC

4) ต่อพอร์ต 1 และ 44 จากแผงวงจรซีพีแอลดี ไปเข้าพอร์ต อินพุตของมอเตอร์ไฟฟ้ากระแสตรง

5) ทำการดาวน์โหลดโปรแกรมมอเตอร์ไฟฟ้ากระแสตรง ตามรูปที่ ค.33

6) ทดลองรันโปรแกรมแล้วบันทึกผลการทดลองลงในตารางที่ ค.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library IEEE;
use IEEE.std_logic_1164.all;
entity dc_motor is
    port (
        left_right: in STD_LOGIC;
        o_vcc: out STD_LOGIC;
        o_gnd: out STD_LOGIC
    );
end dc_motor;
architecture dc_motor_arch of dc_motor is
begin
 process(left_right)
   begin
  if(left_right='1') then
     o_vcc<='1';
     o_gnd<='0';

  elsif(left_right='0' ) then
     o_vcc<='0';
     o_gnd<='1';
  else o_vcc<='0';
       o_gnd<='0';
  end if;
  end process;
end dc_motor_arch;
```

รูปที่ ค.33 โปรแกรมภาษาวีเอชดีแอลบรรยายมอเตอร์ไฟฟ้ากระแสตรง

ตารางที่ ค.1 บันทึกผลการทดลองที่ 1

ดิฟสวิตช์		ลักษณะการหมุนของ
1	44	มอเตอร์ไฟฟ้ากระแสตรง
0	1	
1	0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

1) ต่อสายดาวน์โหลดจากพอร์ตขนานของเครื่องคอมพิวเตอร์เข้ากับแผงวงจรเอฟพีจีเอโดยหันขั้วของสายดาวน์โหลด (แถบสีแดง) ให้ตรงกับพอร์ต VCC

2) ต่อพอร์ต P81 ของโมดูลคีย์บอร์ดเมตริกซ์ เข้าที่ขา 3, 4, 5 และ 6 ของพอร์ต P81

3) ต่อพอร์ต P83 ของแผงวงจรเอฟพีจีเอ ขาที่ 20, 23, 24, 25, 26, 27, 28, 29 ไปยังพอร์ต P81ของ โมดูลแสดงผลเจ็ดส่วน

4) ทำการดาวน์โหลดโปรแกรม Decoder to 7 – Segment

5) ทดลองกดแป้นคีย์บอร์ดเมตริกซ์ตั้งแต่ 0 – F แล้วบันทึกผลการทดลองลงในตารางที่ ค.2

```
library IEEE;
use IEEE.std_logic_1164.all;
entity BCD_7 is
    port (
        a:in STD_LOGIC_VECTOR (3 downto 0);
        o:out STD_LOGIC_VECTOR (6 downto 0)
 );
end BCD_7;
architecture BCD_7_arch of BCD_7 is
begin
  With a select
  o<="1111001"when "0001",  --1
  "0100100" when "0010",  --2
  "0110000" when "0011",  --3
  "0011001" when "0100",  --4
  "0010010" when "0101",  --5
  "0000010" when "0110",  --6
  "1111000" when "0111",  --7
  "0000000" when "1000",  --8
  "0010000" when "1001",  --9
  "1111111" when others;
end BCD_7_arch;
```

รูปที่ ค.34 โปรแกรมภาษาวีเอชดีแอลบรรยาย Decoder to 7 – Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 บันทึกผลการทดลองที่ 2

อินพุต (คีย์บอร์ดเมตริกซ์)	เอาต์พุต (แสดงผลเจ็ดส่วน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
การเขียนภาษาวีเอชดีแอลเบื้องต้น และการเลียนแบบการทำงาน

วัตถุประสงค์

1) เพื่อให้นักศึกษาอธิบายโครงสร้างของภาษาวีเอชดีแอลได้
2) เพื่อให้นักศึกษาเลียนแบบการทำงานของวงจรโดยใช้ภาษาวีเอชดีแอลได้
3) เพื่อให้นักศึกษาใช้เครื่องมือสังเคราะห์วงจรจากภาษาวีเอชดีแอลได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์
2) โปรแกรม Xilinx Froundation 2.1I ที่ติดตั้งเรียบร้อยแล้ว

ทฤษฎีเบื้องต้น

โครงสร้างของภาษา VHDL

1) Entity การบรรยายการเชื่อมต่อกับโลกภายนอกหรือใช้กำหนดทางเข้าออก (Port) มาพร้อมบอกชนิด (Type)
2) Architecture เป็นส่วนอธิบายสถาปัตยกรรมที่อยู่ภายใน Entity
3) Process ใช้กำหนดการทำงานแบบลำดับ
4) Configuration ประกาศเพื่อเลือกเชื่อมโยง Architecture กับ Entity
5) Package ใช้บรรจุข้อมูลที่มีการใช้ซ้ำๆ กันเป็นประจำ
6) Library เป็นส่วนที่เก็บค่ามาตรฐานต่างๆ

สไตล์การเขียนภาษาวีเอชดีแอล

1) Behavioral Style
2) Data flow Style
3) Structure Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1) Behavioral Style
      library IEEE ;
      use IEEE.std_logic_1164.all ;
      entity logic Fen is port (
      a,b,c : in std_logic ;
            y : out std_logic ) ;
      end logic Fen ;
      architecture behavioral of logic Fen is
      begin
      Fen : process (a,b,c) begin
      if (a = '0' and b = '0') then
      r <= '1' ;
      elsit c= '1' then
      r <= '1' ;
      else
      y <= '0' ;
      end if ;
      end process
      end behavioral ;
```

รูปที่ ค.35 Behavioral Style

```
2) Data flow Style
      library IEEE ;
      use IEEE.std_logic_1164.all ;
      entity logic Fen is port (
      a,b,c : in std_logic ;
      y : out std_logic ) ;
       architecture Data flow of logic Fen is
      begin
      y <= '1'when(a = '0'and b = '0') or (c ='1')
      else '0' ;
      end Data flow ;
```

รูปที่ ค.36 แบบ Data Flow Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
3) Structure Style
      library IEEE ;
      use IEEE.std_logic_1164.all ;
      use work primitire.all ;
      entity logic Fen is port (
      a,b,c : in std_logic ;
            y : out std_logic ) ;
      end logic Fen ;
      architecture structural of logic Fen is
      signal not a, not b, and signal : std_logic ;
      begin
      i1 : inverter port map (i => a, 0 => not a) ;
      i2 : inverter port map (i => b, 0 => not b) ;
      a1 : and 2 port map (i1 => not a, i2 => not b,
             y => and signal) ;
      o1 : or 2 port map (i1 => and signal, i2 => c,
             y => y) ;
      end structural ;
```

รูปที่ ก.37 แบบ Structure Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB3_1

3) ออกแบบวงจรออร์เกตโดยเขียนโปรแกรมภาษาวีเอชดีแอล ดังรูปที่ ค.38

```
library IEEE ;
use IEEE.std_logic_1164.all ;
entity OR_Gate is port
(a,b : in std_logic ;
                    o : out std_logic ) ;
end OR_Gate ;
architecture OR_Gate of OR_Gate is
begin
o <= a or b ;
end OR_arc;
```

รูปที่ ค.38 โปรแกรมภาษาวีเอชดีแอลบรรยายออร์เกตแบบ Data Flow Style

4) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB3_2

3) จากการทดลองที่ 1 ให้เปลี่ยนการเขียนโปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style

```
library IEEE;
use IEEE.std_logic_1164.all;
 OR_Gate is port
(a,b : in std_logic ;
                        o : out std_logic ) ;
end OR_Gate ;
architecture OR_arc of OR_Gate is
begin
process (a,b)
begin
if (a ='0' and b = '0') then 0 <= '0' ;
els(0 <= '1');
end if;
end process ;
end OR_arc ;
```

รูปที่ ค.39 โปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style

4) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB3_3

3) จงออกแบบวงจรรวม แอนเก9 2 อินพุต โดยใช้รูปแบบการเขียนภาษาวีเอชดีแอลแบบ Data Flow Style

4) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

การทดลองที่ 4

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB3_4

3) จงออกแบบวงจรแนนเกต 3 อินพุต โดยใช้ภาษาวีเอชดีแอลบรรยายแบบ Behavioral พร้อมทั้งบันทึกผลการทดลอง

4) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

การทดลองที่ 5

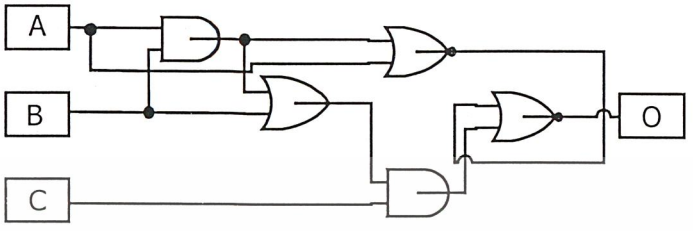
1) จากการทดลองที่ 4 จงเปลี่ยนการเขียนภาษาวีเอชดีแอลจาก Behavioral เป็น Data Flow

2) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

1) จากรูปที่ ค.40 จงเขียนให้เป็นโปรแกรมภาษาวีเอชดีแอล



รูปที่ ค.40 วงจรการทดลองที่ 6

2) ทำการจำลองการทำงาน สังเกตการทำงานของโปรแกรมที่สร้างขึ้นและบันทึกผลการทดลอง

คำถามท้ายการทดลอง

1) ให้ออกแบบ แอนเกต 3 อินพุต และเขียนโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral Style และ Data Flow Style ?

2) ท่านคิดว่าการเขียนโปรแกรมภาษาวีเอชดีแอลแบบ Behavioral Style และ Data Flow Style แตกต่างกันอย่างไร ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
การออกแบบวงจรคอมบิเนชั่นด้วยภาษาวีเอชดีแอล ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

วัตถุประสงค์

1) เพื่อให้นักศึกษาสามารถอธิบายวงจรคอมบิเนชั่นได้
2) เพื่อให้นักศึกษาสามารถออกแบบวงจรคอมบิเนชั่นได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์
2) สายดาวน์โหลด
3) แผงวงจรเอฟพีจีเอ
4) แผงวงจรซีพีแอลดี
5) แผงวงจรอินพุต/เอาต์พุต

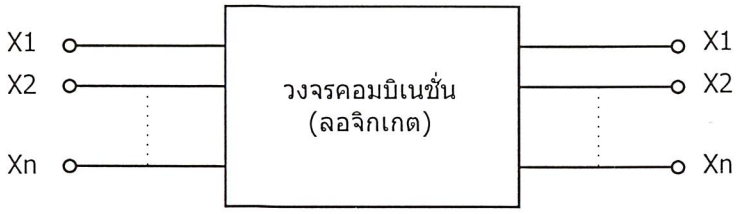
ทฤษฎีเบื้องต้น

วงจรคอมบิเนชั่น ซึ่งเป็นวงจรดิจิตอลที่ได้จากการออกแบบวงจรลอจิก โดยการประยุกต์มาจากสมการพีชคณิตบูลีนในรูปของมินเทอมหรือแม็กเทอม ซึ่งใช้ลอจิกเกตพื้นฐานมารวมกันเพื่อสร้างวงจรลอจิก ส่วนวิธีที่ใช้ในการออกแบบวงจรลอจิก สามารถทำได้หลายลักษณะ เช่น จากฟังก์ชั่นของสมการพีชคณิตบูลีน, จากตารางความจริงและจากไดอะแกรมเวลา (Timing Diagram) ไม่ว่าจะเป็นการออกแบบวงจรลอจิกวิธีใดก็ตาม สิ่งสำคัญที่ต้องทำคือการลดรูปฟังก์ชั่นของสมการให้เหลือน้อยที่สุดก่อน

หลักการออกแบบวงจรคอมบิเนชั่น
(Designing Combination Circuit)

วงจรคอมบิเนชั่น ประกอบด้วยตัวแปรอินพุต (Input Variable) ลอจิกเกตและตัวแปรเอาต์พุต (Output Variables) ดังแสดงในรูปที่ ค.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.41 วงจรคอมบิเนชั่น

วงจรคอมบิเนชั่น คือ วงจรลอจิกที่มีค่าระดับลอจิกของตัวแปรทางเอาต์พุตขึ้นอยู่กับค่าปัจจุบัน (ค่าเวลาในขณะนั้น) ของการจัดหมู่ระดับลอจิกของอินพุตเท่านั้น ค่าของเอาต์พุตจะไม่ขึ้นอยู่กับค่าของอินพุตในอดีตที่ผ่านมาแต่อย่างใด นั่นคือ ระบบไม่มีส่วนความจำและไม่มีการป้อนเอาต์พุตกลับไปเป็นอินพุตอีก ความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตของวงจรคอมบิเนชั่นนี้สามารถเขียนเป็นสมการได้ดังนี้

$$Z_1 = f_1(X_1, X_2 \ldots, X_n)$$
$$Z_2 = f_2(X_1, X_2 \ldots, X_n)$$
$$Z_3 = f_3(X_1, X_2 \ldots, X_n)$$
$$Z_4 = f_4(X_1, X_2 \ldots, X_n)$$

สำหรับขั้นตอนการออกแบบวงจรลอจิก อาจสรุปได้เป็นขั้นตอนดังนี้

1) ทำความเข้าใจปัญหาและขอบเขตของปัญหาให้ชัดเจน

2) ดูจำนวนตัวแปรด้านอินพุตและตัวแปรด้านเอาต์พุตตามลักษณะของปัญหา

3) กำหนดตัวอักษรและระดับลอจิกแทนตัวแปรด้านอินพุตและเอาต์พุต

4) เขียนสมการลอจิก หรือฟังก์ชั่นสวิทชิ่งจากตารางความจริงและจัดให้อยู่ในรูปสมการอย่างง่าย

5) เขียนวงจรลอจิก

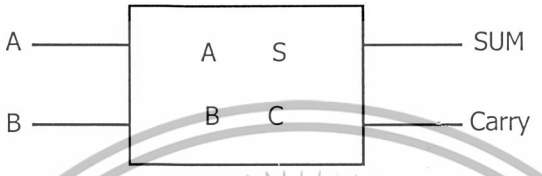
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_1

3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกครึ่ง (Half Adder) ดังรูปที่ ค.42



รูปที่ ค.42 ตัวบวกครึ่ง Half Adder

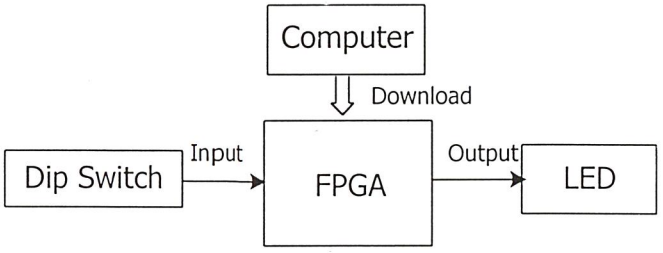
```
Library   IEEE;
Use   IEEE .Std_logic_1164.all;
Entity  halfadd is
Port  (     A, B  : in Std_Logic  ;
            S, C : Out Std_Logic );
End half_add
Architecture  halfadd_arc of halfadd is
Begin
            S   <=  a Xor  b;
            C   <=  a and  b;
End  halfadd_arg  ;
```

รูปที่ ค.43 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกครึ่ง

3) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.3

4) ทดลองดาวน์โหลดลงแผงวงจรเอฟพีจีเอพร้อมตรวจสอบการทำงาน โดยต่อวงจร ดังรูปที่ ค.44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.44 แผนผังการต่อแผงวงจรเอฟพีจีเอ

ตารางที่ ค.3 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input A	Port 3
Input B	Port 4
Output S	Port 5
Output C	Port 6

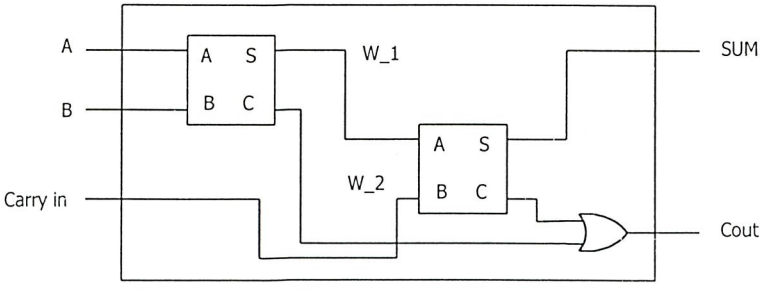
5) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 2

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_2

3) เขียนโปรแกรมภาษาวีเอชดีแอล บรรยายตัวบวกเต็มตัว (Full Adder) โดยเขียนแบบโครงสร้างโดยสร้างจากวงจรตัวบวกครึ่ง (Half Adder) ดังรูปที่ ค.45



รูปที่ ค.45 ตัวบวกเต็ม (Full Adder) สร้างจากวงจร ตัวบวกครึ่ง (Half Adder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทดลองดาวน์โหลดโปรแกรมลงแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.44

5) กำหนดขาเพื่อใช้งานของเอฟพีจีเอดังตารางที่ ค.4

ตารางที่ ค.4 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input A	Port 3
Input B	Port 4
Carry In	Port 5
Output Sum	Port 6
Output Cout	Port 7

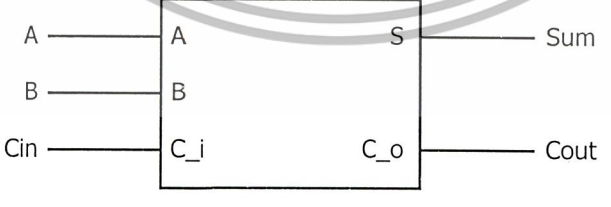
6) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 3

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_3

3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็ม (Full Adder) ดังรูปที่ ค.46 โดยกำหนดขาการทำงานดังตารางที่ ค.46



รูปที่ ค.46 ตัวบวกเต็ม (Full Adder)

4) ทดลองดาวน์โหลดลงแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.44

5) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_4

3) เขียนโปรแกรมภาษาวีเอชดีแอล บรรยายวงจรเข้ารหัสจากตารางความจริงในตารางที่ ค.5

4) นำโปรแกรมที่เขียนได้ดาวน์โหลดลงแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.44

ตารางที่ ค.5 ตารางความจริงของวงจรเข้ารหัส

	Input									Output			
	S1	S2	S3	S4	S5	S6	S7	S8	S9	Q3	Q2	Q1	Q0
1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	1	0	0	0	0	1	1
8	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	1	1	0	0	1

5) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 5

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_5

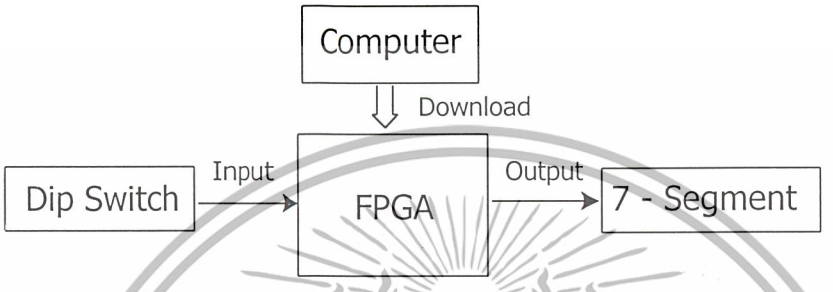
3) เขียนโปรแกรม BCD to 7 Segment (1 – 9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เขียนตารางความจริงของ BCD to 7 Segment (1 – 9)

5) เขียนโปรแกรมภาษาวีเอชดีแอล พร้อมตรวจสอบความถูกต้องโดยการ Synthesis

6) ทำการดาวน์โหลดโปรแกรมลงแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.47



รูปที่ ค.47 แผนผังการต่อแผงวงจรเอฟพีจีเอ

7) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 6

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB4_6

3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายเอแอลยู 4 บิต 15 คำสั่งเบื้องต้นดังตารางที่ ค.6 และกำหนดตัวแปรดังรูปที่ ค.48

ตารางที่ ค.6 เอแอลยู 4 บิต 15 คำสั่ง

S4	S3	S2	S1	S0	Cin	Operation	Function	Implementation block
0	0	0	0	0	0	Y<=A	Transfer A	Arithmetic Unit
0	0	0	0	0	1	Y<=A+1	Increment A	Arithmetic Unit
0	0	0	0	1	0	Y <= A+B	Addition	Arithmetic Unit
0	0	0	0	1	1	Y <= A+B+1	Add with carry	Arithmetic Unit
0	0	0	1	0	0	Y <= A+Bbar	A plus 1's complement of B	Arithmetic Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.6 (ต่อ) เอแอลยู 4 บิต 15 คำสั่ง

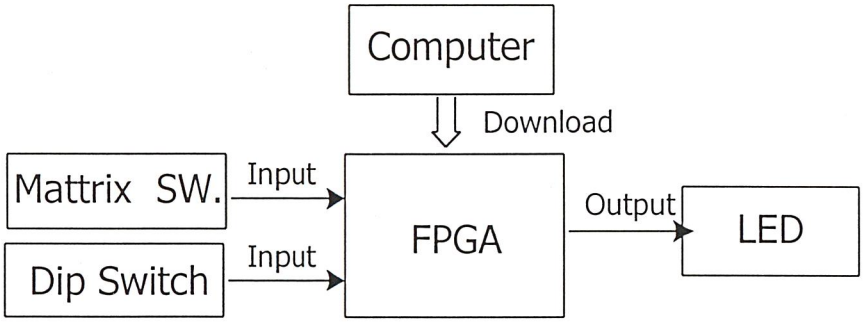
S4	S3	S2	S1	S0	Cin	Operation	Function	Implementation block
0	0	0	1	0	1	Y <= A+Bbar+1	Subratraction	Arithmetic Unit
0	0	0	1	1	0	Y <= A-1	Decrement A	Arithmetic Unit
0	0	0	1	1	1	Y <= A	Transfer A	Arithmetic Unit
0	0	1	0	0	0	Y <= A and B	AND	Logic Unit
0	0	1	0	1	0	Y <= A or B	OR	Logic Unit
0	0	1	1	0	0	Y <= A xor B	XOR	Logic Unit
0	0	1	1	1	0	Y <= Abar	Complement A	Logic Unit
0	0	0	0	0	0	Y <= A	Transfer A	Shifter Unit
0	1	0	0	0	0	Y <= shl A	Shift left A	Shifter Unit
1	0	0	0	0	0	Y <= shr A	Shift right A	Shifter Unit
1	1	0	0	0	0	Y <= 0	Transfer 0's	Shifter Unit

```
Library IEEE;
Use IEEE.STD_logic_1164.all,IEEE.nUmeric_STD.all;
Entity ALU is
 Port (
      Sel:  in    unsigned(4 downto 0);
      Carry in    std_logic;
      A,B   in    Unsiged(7 downto 0);
      Y     out   unsigned(7 downto 0);
End entity ALU;
```

รูปที่ ค.48 การกำหนดตัวแปร

4) ทำการดาวน์โหลดโปรแกรมลงแผงวงจรเอฟพีจีเอพร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.49 แผนผังการต่อแผงวงจรเอฟพีจีเอ

5) บันทึกผลการจำลองผลการทำงาน

คำถามท้ายการทดลอง

1) จงออกแบบเอแอลยูดังรูปที่ ค.50

รูปที่ ค.50 แผนผังการออกแบบเอแอลยู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเอแอลยูจะเขียนข้อมูลแทนได้ดังนี้

A or B

A and B

Not A

Not B

A nor B

A nand B

A xor B

A xnor B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5
การออกแบบวงจรซีเควนเชียลด้วยภาษาวีเอชดีแอล ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

วัตถุประสงค์

1) เพื่อให้นักศึกษาสามารถอธิบายวงจรซีเควนเชียลได้
2) เพื่อให้นักศึกษาสามารถออกแบบวงจรซีเควนเชียลได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์
2) สายดาวน์โหลด
3) แผงวงจรเอฟพีจีเอ
4) แผงวงจรซีพีแอลดี
5) แผงวงจรอินพุต/เอาต์พุต

ทฤษฎีเบื้องต้น

วงจรซีเควนเชียลแบ่งได้เป็น 2 แบบ โดยใช้สัญญาณหรือคาบเวลาของสัญญาณ ที่ใช้ควบคุมการทำงานเป็นหลัก ในการพิจารณาได้แก่ วงจรซีเควนเชียลแบบอะซิงโครนัส (Asyschronous) และแบบซิงโครนัส (Syschronous)

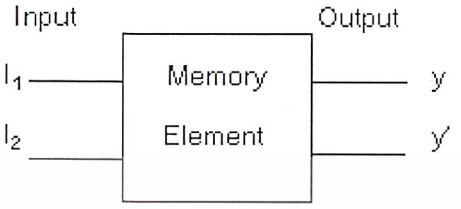
วงจรซีเควนเชียลแบบอะซิงโครนัส เป็นวงจรที่เอาต์พุตของวงจร สามารถเปลี่ยนแปลงสภาวะได้ทุกเวลาที่มีการเปลี่ยนแปลงอินพุต (State Update At Any Time)

วงจรซีเควนเชียลแบบซิงโครนัส เป็นวงจรที่เอาต์พุตของวงจร จะเปลี่ยนแปลงสภาวะได้เมื่อมีสัญญาณ (Clock Pulse) มากระตุ้นและมีอินพุต ที่เหมาะสม หรือเป็นวงจรที่ทำงานแบบเข้าจังหวะ (State Update At Discrete Time)

รูปที่ ค.51 เป็นรูปแสดงอุปกรณ์เก็บข้อมูลพื้นฐาน ที่ประกอบด้วยอินพุต I1 และ I2 มีเอาต์พุต y และ y' ในบล็อกประกอบด้วย ชุดของรีเลย์ (Relay) หรือ ฟลิบฟลอบ (Flip-Flop) หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์อื่นที่สามารถคงสถานะของเอาต์พุตได้ไม่ว่าอินพุตเดิมยังคงอยู่หรือไม่ก็ตาม แต่ถ้าต้องการให้เอาต์พุตเปลี่ยนสถานะก็จะต้องจ่ายอินพุตใหม่ที่เหมาะสมเข้าไป



รูปที่ ค.51 โครงสร้างของอุปกรณ์เก็บข้อมูล

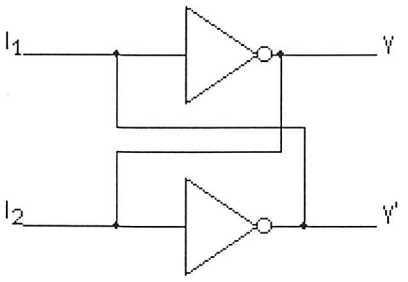
อุปกรณ์เก็บข้อมูลหรืออุปกรณ์หน่วยความจำที่ทำจาก วงจรซีเควนเชียลนี้ เป็นวงจรไบสเตเบิลมัลติไวเบรเตอร์ (Bistable Multivibrator) จะมีเอาต์พุต 2 เอาต์พุต มีค่าลอจิกตรงข้ามกัน กรณีที่อุปกรณ์หน่วยความจำเก็บข้อมูลที่มีค่าลอจิกเป็น "1" นั่นคือ y="1" และ y'="0" หรือถ้าเก็บข้อมูล ที่มีค่าลอจิกเป็น "0" นั่นคือ y="0" และ y'="1" อุปกรณ์เก็บข้อมูลนี้เรียกว่า แลทช์ (Latch)หรือ ฟลิบฟลอบ (Flip-Flop) แบ่งได้เป็น 2 แบบ คือ

Direct Flip-Flop เป็นแบบป้อนข้อมูลหรือค่าลอจิกเข้าโดยตรงทางอินพุต โดยไม่ต้องมีสัญญาณควบคุมใด ๆ คือเป็น วงจรซีเควนเชียลแบบอะซิงโครนัสนั่นเอง ได้แก่ RS-Latch, NOR Latch, NAND Latch เป็นต้น

Triggering Flip-Flop เป็นแบบที่จะต้องมีสัญญาณควบคุมมากระตุ้นเพื่อให้ Flip-Flop ทำงาน โดยทั่วไปสัญญาณที่มากระตุ้นนี้ เรียกว่า Clock (Clk หรือ Ck) หรือ บางทีขาที่ใช้ควบคุมนี้เรียกว่าขา อีนาเบิล (Enable) คือเป็น วงจรซีเควนเชียลแบบซิงโครนัส (วงจรที่ทำงานแบบเข้าจังหวะ) นั่นเอง ได้แก่ RS Flip-Flop, JK Flip-Flop, D Flip-Flop, T Flip-Flop เป็นต้น

ฟลิบฟลอบ เป็นวงจรไบสเตเบิลที่ประกอบด้วยลอจิกเกตต่าง ๆ เช่น นอร์เกต, แนนเกตมาต่อกัน โดยเอาต์พุตของเกตตัวหนึ่งต่อย้อนกลับ (Feed Back) เข้ามายังอินพุตอีกตัวหนึ่ง อีกตัวหนึ่งก็ทำเช่นเดียวกัน ดังตัวอย่างในรูปที่ ค.52 ฟลิบฟลอบ เป็นชื่อของวงจรประเภทหนึ่งซึ่งมีสถานะทางเอาต์พุตที่เสถียรสองสถานะ และการเปลี่ยนแปลงเอาต์พุตเกิดจากการกระตุ้นทางอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



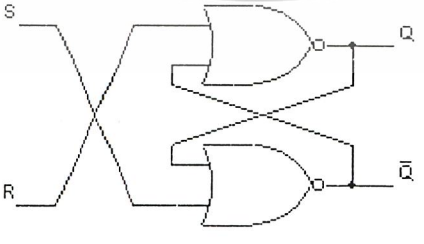
รูปที่ ค.52 ฟลิบฟลอบอย่างง่าย

ฟลิบฟลอบ เป็นอุปกรณ์เก็บข้อมูลหรืออุปกรณ์หน่วยความจำเพราะว่า เมื่อฟลิบฟลอบได้รับอินพุตที่เหมาะสม เอาต์พุตของฟลิบฟลอบจะยังคงสถานะเช่นนั้นตลอดไป ไม่ว่าอินพุตเดิมจะยังคงอยู่หรือไม่ก็ตาม แต่ถ้าต้องการที่จะให้เอาต์พุตของฟลิบฟลอบเปลี่ยนสถานะก็จะต้องจ่ายอินพุตใหม่ที่เหมาะสมเข้าไป

ฟลิบฟลอบ มีชื่อเรียกหลายชื่อ เช่น แลทซ์ (Latch) และ ไบสเตเบิลมัลติไวเบรเตอร์ (Bistable Multivibrator) แลทซ์ (Latch) จะใช้เรียกฟลิบฟลอบ ที่เป็นแบบป้อนข้อมูลหรือค่าลอจิกเข้าโดยตรงทางอินพุต หรือ ไดเร็กฟลิบฟลอบ (Direct FF) และฟลิบฟลอบที่มีขาอีนาเบิล (Enable) ถูกกระตุ้นให้ทำงานด้วยลอจิก "1" ไบสเตเบิลมัลติไวเบรเตอร์ (Bistable Multivibrator) เป็นชื่อทางเทคนิค

นอร์เกจแลทซ์ (NOR Gate Latch)

เป็น เอสอาร์แลทซ์ (SR Latch) ที่ทำจากนอร์เกตขาอินพุต S คือ Set ส่วนขาอินพุต R คือขา Reset และมีเอาต์พุต 2 เอาต์พุต คือ Q และ $\overline{Q}$ ดังแสดงในรูปที่ ค.53



รูปที่ ค.53 วงจร SR Latch ชนิด NOR gate หรือ NOR Latch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.7 การเกิดสถานะของเอาต์พุตปัจจุบัน

Input		Output		Action
S	R	Q	$\overline{Q}$	
1	0	1	0	Q = 1 : Set State
0	0	1	0	
0	1	0	1	Q = 0 : Reset State
0	0	0	1	
1	1	0	0	Undefined หรือสถานะไม่พึงประสงค์

ตารางที่ ค.8 ตารางความจริงของนอร์แลช์ (NOR Latch)

S	R	Q	Action
0	0	NC	No Change
0	1	0	Reset
1	0	1	Set
1	1	-	Undefined

อธิบายการทำงานจากตารางที่ ค.7

1. ในขณะที่ Q และ $\overline{Q}$ มีสถานะเดิมเป็นอะไรก็ได้ เมื่อให้ S=1 และ R=0 จะทำให้ Q=1และ $\overline{Q}$ = 0 ทันที ในขณะเดียวกันถ้าเปลี่ยนให้ S=0 และ R ยังเป็น 0 เหมือนเดิม เอาต์พุตยังคงเหมือนเดิม ไม่เปลี่ยนแปลง คือ Q=1 และ $\overline{Q}$ =0 นั่นคือการเก็บข้อมูลเดิม (Store) นั่นเอง และอินพุตที่ทำให้ Q=1 ก็คือ S=1 และ R=0 สถานะนี้เรียกว่าสถานะ SET คือการทำให้ Q = 1

2. ในขณะที่ Q และ $\overline{Q}$ มีสถานะเดิมเป็นอะไรก็ได้ เมื่อให้ S=0 และ R=1 จะทำให้ Q=0 และ $\overline{Q}$ =1 ทันที ในขณะเดียวกันถ้าเปลี่ยนให้ R=0 และ S ยังเป็น 0 เหมือนเดิม เอาต์พุตยังคงเหมือนไม่เปลี่ยนแปลง คือ Q=0 และ $\overline{Q}$ =1 นั่นคือการเก็บข้อมูลเดิม (Store) นั่นเอง และอินพุตที่ทำให้ Q=0 ก็คือ S=0 และ R=1 สถานะนี้เรียกว่าสถานะ RESET คือการทำให้ Q=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในขณะที่ Q และ $\overline{Q}$ มีสถานะเดิมเป็นอะไรก็ได้ เมื่อให้ S=1 และ R=1 จะทำให้ Q=0 และ $\overline{Q}$ =0 ทันที สถานะดังกล่าวไม่ใช่คุณสมบัติของ ฟลิบฟลอบ ดังกล่าวมาแล้ว (ฟลิบฟลอบมีค่าลอจิกเอาต์พุตตรงข้ามกัน) สถานะนี้จึงเป็นสถานะที่ไม่พึงประสงค์ หรือห้ามใช้

รีจิสเตอร์(Register)

รีจิสเตอร์เป็นอุปกรณ์สำหรับจัดเก็บข้อมูล ซึ่งมีความสามารถเก็บข้อมูลที่อยู่ในรูปของเลข 0 และ 1 (Binary Data)ได้ โดยที่รีจิสเตอร์ประกอบไปด้วยฟลิปฟลอป (Flip Flop) หลาย ๆ ตัว นั่นคือถ้ากล่าวถึงรีจิสเตอร์ขนาด n บิต ก็จะประกอบไปด้วยฟลิปฟลอปจำนวน n ตัวด้วย ตัวอย่างเช่น รีจิสเตอร์ 4 บิต จะประกอบด้วยฟลิปฟลอปจำนวน 4 ตัว เพื่อสร้างเป็นริจิสเตอร์ 4 บิต

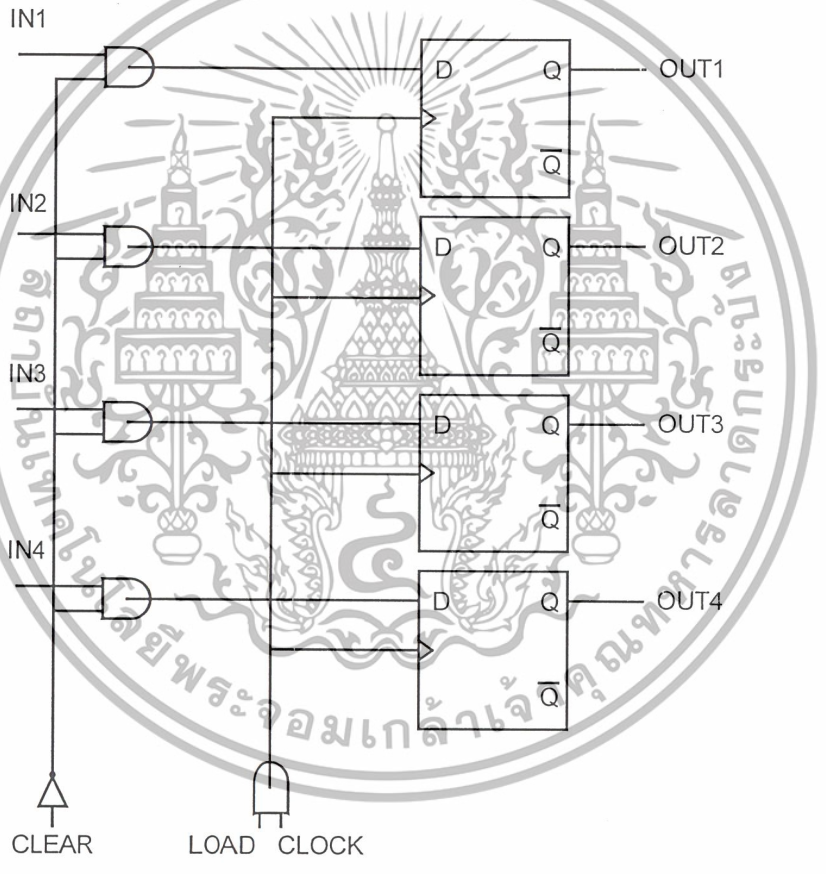
รูปที่ ค.54 รีจิสเตอร์ขนาด 4 บิต

จากรูปที่ ค.54 คือรีจิสเตอร์ขนาด 4 บิต ดีฟลิปฟลอปจำนวน 4 ตัว มีข้อมูลเข้าคือ IN1, IN2, IN3, IN4 โดยแต่ละเส้นจะถูกเชื่อมต่อเข้ากับ ดีของฟลิปฟลอปแต่ละตัว โดยมีข้อมูลออกคือ OUT1, OUT2, OUT3, OUT4 จาก Q

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสัญญาณนาฬิกา (Clock Pluse) เกิดขึ้นในแต่ละครั้ง ข้อมูลจะถูกนำจากเส้น IN1, IN2, IN3, IN4 เข้าสู่ รีจิสเตอร์ ซึ่งจะเรียกการนำข้อมูลเข้าสู่รีจิสเตอร์ว่า “โหลด(Load)” โดยการเข้ามาของข้อมูลและข้อมูลออก จะเป็นลักษณะแบบขนาน (Parallel) โดยที่ข้อมูลเข้าทั้ง 4 ตัวเข้ามาสู่รีจิสเตอร์ จากรูปคือ ดี และได้ข้อมูลออกมาจาก Q สู่ OUT1, OUT2, OUT3, OUT4

สัญญาณนาฬิกาที่เกิดขึ้นแต่ละครั้ง เมื่อมีข้อมูลเข้าสู่รีจิสเตอร์ (IN1…IN4) และจะยังเก็บอยู่ในรีจิสเตอร์ และจะมีการเปลี่ยนแปลงอีกครั้งเมื่อมีการเปลี่ยนสัญญาณนาฬิกาในครั้งต่อไป ซึ่งจะพบว่าสัญญาณนาฬิการจะเป็นตัวควบคุมทำให้เกิดการโหลดข้อมูลสู่รีจิสเตอร์



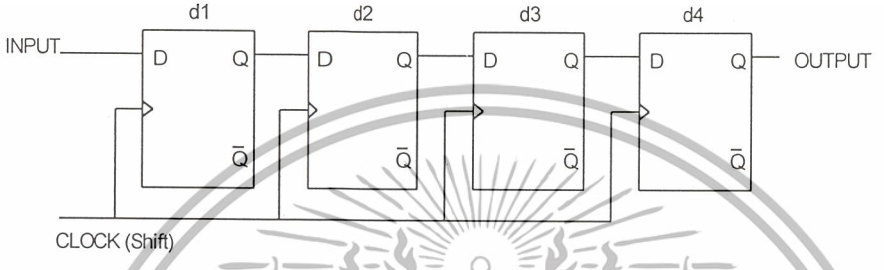
รูปที่ ค.55 ชิพรีจิสเตอร์

จากรูปที่ ค.55 ค่าโหลด(LOAD) จะต้องมีค่าเท่ากับ 1 สำหรับเป็นการทำให้เกิดการนำเอาข้อมูลเข้าสู่รีจิสเตอร์ หรือถ้าในกรณีการเคลียร์ค่ารีจิสเตอร์ ตัวเคลียร์จะถูกเซ็ทค่า คือเท่ากับ 1 และเมื่อผ่านน็อทเกต (Not Gate) ก็จะเป็น 0 ทำให้มีการนำเอาค่า 0 ทั้งหมดเข้าสู่รีจิสเตอร์ สำหรับการเคลียร์ที่เกิดขึ้น จะถือได้ว่าเป็นงานทั่วๆ ไป ที่เกิดขึ้นในวงจรทั้งหลาย ส่วนงานด้านอื่นๆในวงจรคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การชิพ (Shift) ซึ่งจะเป็นการเลื่อนค่าไปด้านซ้ายหรือด้านขวา ขึ้นอยู่กับการกำหนด โดยทั่วไปแล้ว การชิพซ้ายจะเป็นการนำเอาค่าในรีจิสเตอร์คูณด้วย 2 และการชิพขวาก็จะเป็นการนำเอาค่าในรีจิสเตอร์ไปหารด้วย 2

ชิพรีจิสเตอร์ (Shift Register)



รูปที่ ค.56 ชิพรีจิสเตอร์ขนาด 4 บิต

จากรูปที่ ค.56 จะแสดงถึงชิพรีจิสเตอร์ขนาด 4 บิตที่สร้างจาก ดีฟลิปฟลอป จำนวน 4 ตัว โดยใช้เอาต์พุตของ d1, d2 และ d3 ต่อเข้าเป็นอินพุตของ d2, d3, d4 ตามลำดับ จนได้ออกมาเป็นข้อมูลสุดท้ายที่ d4

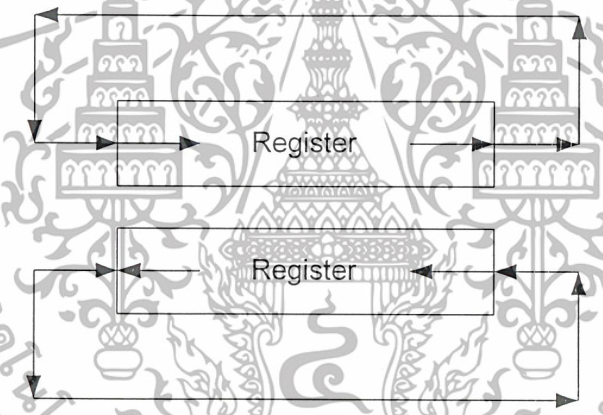
เมื่อพิจารณาที่คล็อก (Clock) ซึ่งเป็นสัญญาณนาฬิกาที่กำหนดให้มีการชิพเกิดขึ้น และในแต่ละครั้งของสัญญาณนาฬิกา จะทำให้ข้อมูลถูกส่งจาก d1 ไปยัง d2 ไปยัง d3 และ d4 ตามลำดับ และเกิดขึ้นในเวลาเดียวกันทำให้มีการเลื่อนขยับตำแหน่งของข้อมูลในรีจิสเตอร์

โครงสร้างของชิพริจิสเตอร์แบบ 4 บิต ไม่ว่าจะใช้ฟลิปฟลอปใดๆมาสร้างก็ตาม การทำงานจะต้องสามารถทำการโหลดข้อมูลเข้าสู่รีจิสเตอร์ได้ โดยจะต้องมีการกำหนดค่าของข้อมูลเข้าที่สอดคล้องกับความต้องการในการชิพแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.57 การโหลดข้อมูลแบบอนุกรม (Serial)

จากรูปที่ ค.57 ในกรณีของชิพรีจิสเตอร์ขนาด n บิต ก็จะสามารถโหลดข้อมูลในรูปแบบอนุกรม(Serial) ได้ตามจำนวนครั้งของสัญญาณนาฬิกาที่เปลี่ยนแปลงไป n ครั้ง และข้อมูลออกจากรีจิสเตอร์ ตามสัญญาณนาฬิกาได้จำนวน n ครั้งเช่นกัน เมื่อมีการชิพไปแล้ว n ครั้งข้อมูลที่อยู่ในรีจิสเตอร์จะกลายเป็น 0 ทั้งหมดหรือ 1 ทั้งหมดก็ได้ โดยจะขึ้นอยู่กับว่าข้อมูลเข้าเป็น 0 หรือ 1



รูปที่ ค.58 การทำงานหมุนเป็นวงกลมของรีจิสเตอร์

จากรูปที่ ค.58 ถ้าเรานำเอาข้อมูลออกของรีจิสเตอร์มาต่อเชื่อมกับเส้นทางข้อมูลเข้า ก็จะกลายเป็นการหมุนเป็นวงกลม (Circulate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB5_1

3) เขียนโปรแกรมวีเอชดีแอล บรรยายดีฟลิปฟลอปดังรูปที่ ค.59

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.9

5) ทดลองโดยการดาวน์โหลดลงแผงวงจรเอฟพีจีเอ พร้อมทดสอบการทำงานโดยต่อวงจรดังรูปที่ ค.60

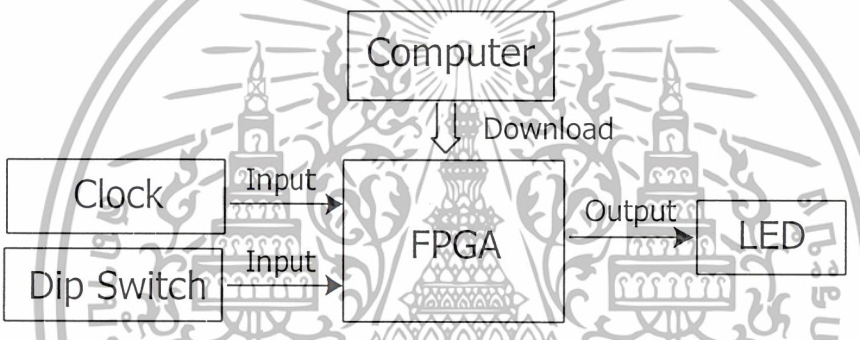
```
VHDL Process Description
library ieee;
use ieee.std_logic_1164.all;
entity dff is
     port (CLK, RESET, D: in std_logic;
           Q, Q_n : out std_logic;
end dff;
architecture pet_pr of dff is
-- Implements positive edge-triggered bit state
storage
-- with asynchronous reset.
     signal state : std_logic;
begin
     Q <= state;
     Q_n <= not state;
     process (CLK, RESET)
     begin
               if (RESET = '1') then
          state <= '0';
          else
               if (CLK'event and CLK = '1') then
                    state <= D;
               end if;
          end if;
     end process;
end pet_pr;
```

รูปที่ ค.59 โปรแกรมภาษาวีเอชดีแอลบรรยายดีฟลิปฟลอป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.9 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input CLK	Port 78
Input Reset	Port 77
Output Q	Port 3
Output Q_n	Port 4



รูปที่ ค.60 แผนผังการต่อแผงวงจรเอฟพีจีเอ

การทดลองที่ 2

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

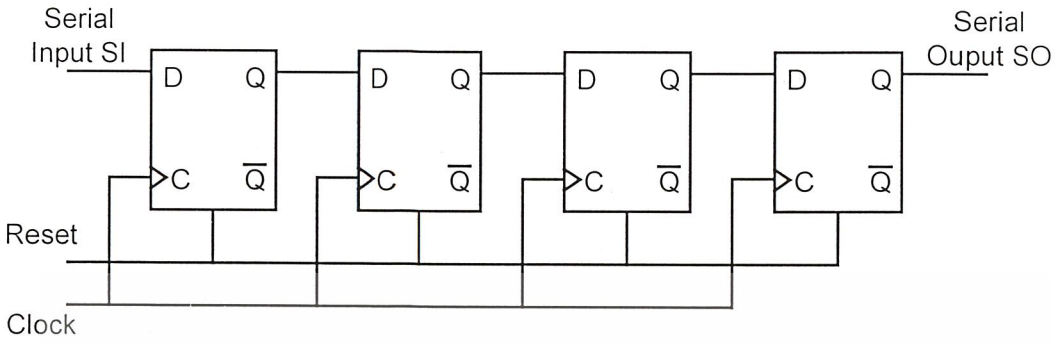
2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB5_2

3) เขียนโปรแกรมภาษาวีเอชดีแอล บรรยายตัวเลื่อนข้อมูลดังรูปที่ ค.61

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.10

5) ทดลองโดยการดาวน์โหลดลงแผงวงจรเอฟพีจีเอ พร้อมทดสอบการทำงานโดยต่อวงจรดังรูปที่ ค.60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.61 4-Bit Shift Register

ตารางที่ ค.10 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input CLK	Port 78
Input Reset	Port 79
Input SI	Port 80
Output Q_0-Q_3	Port 3-6
Output SO	Port 7

6) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 3

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB5_3

3) เขียนโปรแกรมภาษาวีเอชดีแอล บรรยายตัวนับ 4 บิต(Counter)

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.11

5) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.11 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input CLK	Port 78
Input Reset	Port 79
Output Q_0-Q_3	Port 3-6

6) บันทึกผลการจำลองผลการทำงาน

การทดลองที่ 4

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB5_4

3) เขียนวงจร นับ 5 บิต (Counter) โดยให้นับขึ้นทีละหนึ่ง นับลงทีละสองซึ่งกำหนดตัวแปรดังรูปที่ ค.62

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.12

5) ทดลอง โดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงาน โดยต่อวงจรดังรูปที่ ค.60

ตารางที่ ค.12 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input CLK	Port 78
Input Reset	Port 3
Input Up	Port 4
Input Down	Port 5
Output Q_0-Q_4	Port 44-48

6) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Library IEEE;
      Use IEEE.STD_Logic_1164.all,
IEEE.Numeric_STD.all ;
      Entity CNT_UP1_DOWN2 is
        Port (Clock, Reset, Up, Down : in std_logic;
            Count: out unsigned (4 downto 0));
      end entity CNT_UP1_DOWN2 ;
```

รูปที่ ค.62 การกำหนดพอร์ต

คำถามท้ายการทดลอง

1) จงออกแบบวงจรนับขึ้นทีละ 1 พร้อมทั้งแสดงผลออกโมดูลแสดงผลเจ็ดส่วนตั้งแต่ 0-9?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 6
การออกแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

วัตถุประสงค์

1) เพื่อให้นักศึกษาสามารถอธิบายระบบดิจิตอลได้
2) เพื่อให้นักศึกษาสามารถออกแบบระบบดิจิตอลได้
3) เพื่อให้นักศึกษาสามารถเขียนโปรแกรมภาษาวีเอชดีแอลบรรยายระบบดิจิตอลได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์
2) สายดาวน์โหลด
3) แผงวงจรเอฟพีจีเอ
4) แผงวงจรซีพีแอลดี
5) แผงวงจรอินพุต/เอาต์พุต

ทฤษฎีเบื้องต้น

1) กำหนดข้อกำหนดของระบบที่ต้องการในระดับสูง
2) แปรความต้องการเป็นไดอะแกรมสถานะ หรือ ตารางสถานะในระดับสูง
3) ลดความซ้ำซ้อนของฐานะ (ถ้ามี)
4) กำหนดรหัสไบนารีให้สถานะ (ถ้าทราบจำนวนสถานะสามารถทราบจำนวนฟลิปฟลอป)
5) เขียนตารางสถานะ (สถานะปัจจุบัน, ถัดไปและเอาต์พุต) โดยแทนสถานะด้วยรหัสไบนารีที่กำหนดในข้อ 4
6) เขียนแผนภูมิ K-map เพื่อหาสมการการกระตุ้น พิจารณาจากตาราง ในข้อ 5 จาก Q (t) ที่เปลี่ยนสถานะเป็น Q(t + 1) หรือ Q(next) และใช้ตารางสถานะของฟลิปฟลอปที่ต้องการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนสถานะถัดไปของแต่ละขาของการกระตุ้นบน K-map และเขียน K-map ของเอาต์พุต จะได้สมการการกระตุ้นที่อินพุตฟลิปฟลอปและสมการเอาต์พุต

7) นำสมการในข้อ 6 มาออกแบบวงจร

การทดลองที่ 1

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB6_1

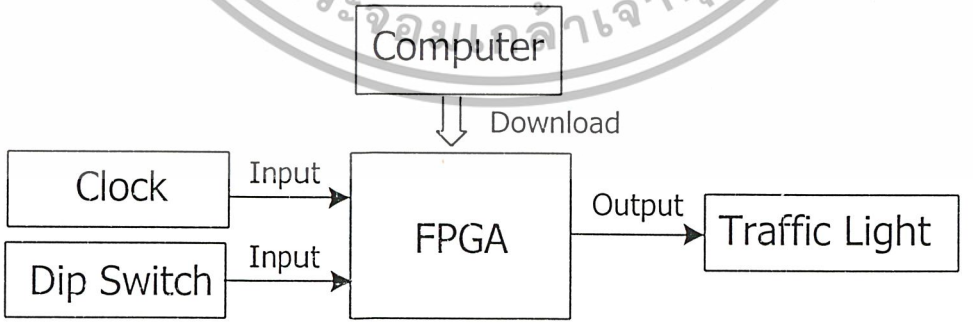
3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายการทำงานของระบบควบคุมสี่แยกไฟแดง

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.13

5) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.63

ตารางที่ ค.13 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input Clock	Port 78
Input On/Off	Port 10
Output O0 – O3	Port 3 - 6



รูปที่ ค.63 แผนผังการต่อแผงวงจรเอฟพีจีเอ

6) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB6_2

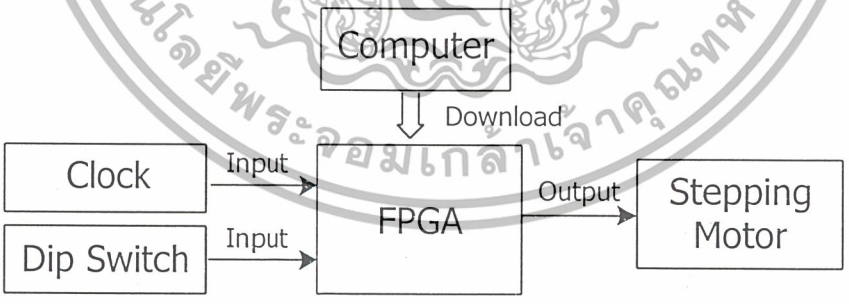
3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายระบบการควบคุม Stepping Motor พร้อมทั้งแสดงจำนวนรอบและไฟเพื่อให้ทราบว่าขณะนี้หมุนซ้ายหรือขวา

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.14

5) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.64

ตารางที่ ค.14 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input Clock	Port 78
Input Enable	Port 10
Lr	Port 9
Output $O_0 - O_3$	Port 44 - 47



รูปที่ ค.64 แผนผังการต่อแผงวงจรเอฟพีจีเอ

6) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB6_3

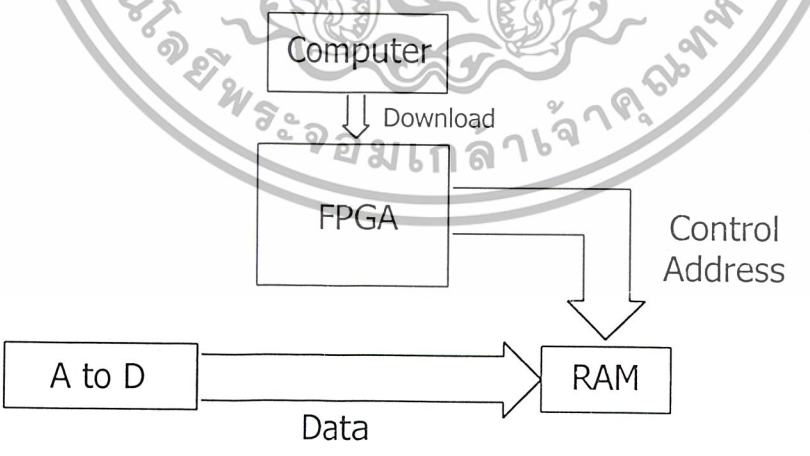
3) เขียนโปรแกรมภาษาวีเอชดีแอลโดยรับค่าจากวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลมาเก็บไว้ในแรม 15 ตำแหน่ง

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.15

5) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.65

ตารางที่ ค.15 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input Enable	Port 3
Input Address	Port 4 - 7
Output WR	Port 8
Output OE	Port 9



รูปที่ ค.65 แผนผังการต่อแผงวงจรเอฟพีจีเอ

6) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB6_4

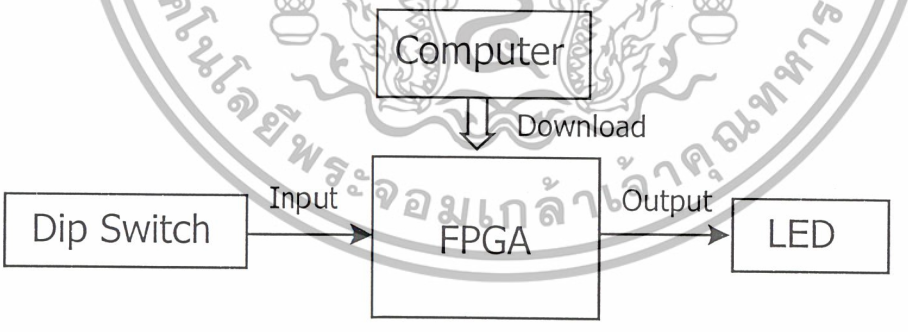
3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรคูณเลขฐาน 2 (4 บิต) โดย ดึงค่าตัวเลขในอีพรอมที่เก็บไว้เป็นตัวเลขตั้งแต่ 1 ถึง 255

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.16

5) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.66

ตารางที่ ค.16 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input A_{0-3}	Port 3 – 6
Input B_{0-3}	Port 7 – 10
Output O_{0-8}	Port 77 – 84 , 44



รูปที่ ค.66 แผนผังการต่อแผงวงจรเอฟพีจีเอ

6) บันทึกผลการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

1) เรียกโปรแกรม Xilinx Foundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB6_5

3) เขียนโปรแกรมภาษาวีเอชดีแอลควบคุมการทำงานของวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก โดยให้รับค่าจาก สวิตซ์แมททริกซ์ดังนี้

ถ้า กด 1 ให้ สัญญาณ จากวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกมีค่าต่ำสุด คือ ประมาณ 1 .8 V

กด 2 ให้ สัญญาณ จากวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกมีค่าต่ำสุด คือ ประมาณ 2 V

กด 3 ให้ สัญญาณ จากวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกมีค่าต่ำสุด คือ ประมาณ 3 V

กด 4 ให้ สัญญาณ จากวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกมีค่าต่ำสุด คือ ประมาณ 4 V

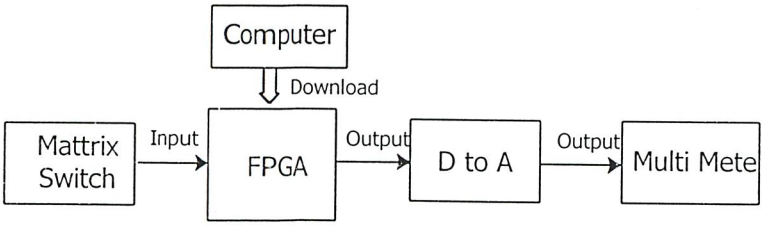
กด 5 ให้ สัญญาณ จากวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกมีค่าต่ำสุด คือ ประมาณ 5 V

4) กำหนดขาการทำงานของเอฟพีจีเอดังตารางที่ ค.17

5) ทดลอง โดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงาน โดยต่อวงจรดังรูปที่ ค.67

ตารางที่ ค.17 กำหนดขาเพื่อใช้งานเอฟพีจีเอ

ชื่อขา	พอร์ต
Input A_{0-3}	Port 23 – 26
Output O_{0-7}	Port 3 – 10



รูปที่ ค.67 แผนผังการต่อแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) บันทึกผลการจำลองผลการทำงาน

คำถามท้ายการทดลอง

1) สร้างนาฬิกาดิจิตอลโดยใช้โปรแกรมภาษาวีเอชดีแอล ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 7
การติดต่อระหว่างเอฟพีจีเอ กับ MCS 51

วัตถุประสงค์

1) เพื่อให้นักศึกษาสามารถอธิบายการติดต่อระหว่างเอฟพีจีเอ กับ MCS 51 ได้

2) เพื่อให้นักศึกษาสามารถออกแบบวงจรดิจิตอลเพื่อการติดต่อระหว่างเอฟพีจีเอ กับ MCS 51 ได้

3) เพื่อให้นักศึกษาสามารถเขียนโปรแกรมภาษาวีเอชดีแอล บรรยายวงจรดิจิตอลเพื่อการติดต่อระหว่างเอฟพีจีเอ กับ MCS 51 ได้

เครื่องมือและอุปกรณ์

1) ระบบไมโครคอมพิวเตอร์

2) สายดาวน์โหลด

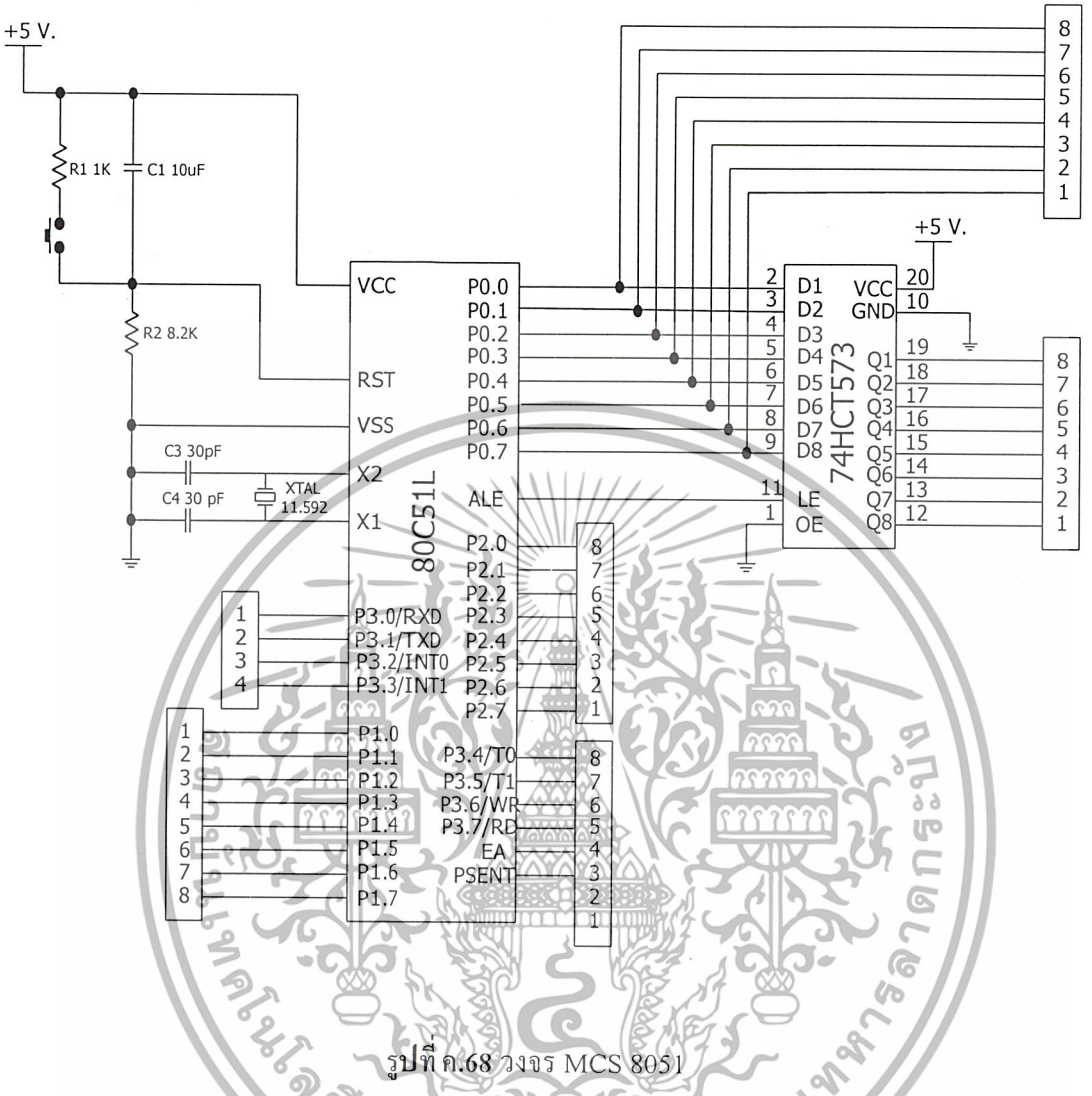
3) แผงวงจรเอฟพีจีเอ

4) ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

ทฤษฎีเบื้องต้น

โมดูล MCS 8051 ดังแสดงในรูปที่ ค.68 เป็นไมโครคอนโทรลเลอร์ เบอร์ 80C51L สามารถเขียนโปรแกรมควบคุมโดยใช้ภาษา Assembly ซึ่งจะต่อ MCS 8051 แบบเบื้องต้นร่วมกับ IC 74HCT573 ใช้ Latch ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.68 วงจร MCS 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.18 ตำแหน่งขาของ MCS 8051 เบอร์ 80C51L

ขาที่	คุณสมบัติ	ขาที่	คุณสมบัติ
1	Port 1.0	28	Port 2.7
2	Port 1.1	29	$\overline{\text{PSEN}}$
3	Port 1.2	30	ALE
4	Port 1.3	31	$\overline{\text{EA}}$
5	Port 1.4	32	Port 0.7
6	Port 1.5	33	Port 0.6
7	Port 1.6	34	Port 0.5
8	Port 1.7	35	Port 0.4
9	Reset (Rst)	36	Port 0.3
10	Port 3.0 RXD	37	Port 0.2
11	Port 3.1 TXD	38	Port 0.1
12	Port 3.2 $\overline{\text{INT 0}}$	39	Port 0.0
13	Port 3.3 $\overline{\text{INT 1}}$	40	VCC
14	Port 3.4 T 0		
15	Port 3.5 T 1		
16	Port 3.6 $\overline{\text{WR}}$		
17	Port $\overline{\text{RD}}$		
18	XTAL 2		
19	XTAL 1		
20	Ground		
21	Port 2.0		
22	Port 2.1		
23	Port 2.2		
24	Port 2.3		
25	Port 2.4		
26	Port 2.5		
27	Port 2.6		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตของ 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมารายละเอียดต่าง ๆ ดังนี้

1) Vcc (ขา 40) ต่อกับ +5V

2) Vss (ขา 20) เป็นขา GND

3) พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7 – P0.0) มีโครงสร้างแบบ Open-Drain Bi-direction

4)พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ 2 หน้าที่ คือ แอดเดรสบัสและดาต้าบัสเมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ต ถ้าต้องการให้ทำงานเป็นอินพุตพอร์ตต้องส่งลอจิก "1" ไปยังพอร์ตนี้ จะมีผลให้ Q ของ D-$\overline{FF}$ เป็น "0" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุตพอร์ตแลทช์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (pin) จะใช้สัญญาณ Read(pin)

5) พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน Internal Pull up Register

6) พอร์ต 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.7-P2.0) มีโครงสร้างคล้าย พอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal pull up) พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ตใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุตพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FET อยู่ในสถานะ off

7) พอร์ตขา (10-17) มีทั้งหมด 8 บิต คือ ขา (P3.7-P3.0) มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ตถ้าจะโปรแกรมให้เป็นอินพุตพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมาและรับสัญญาณเข้าไปสัญญาณต่าง ๆ ดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)

P3.2/$\overline{INT0}$ (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0

P3.3/$\overline{INT1}$ (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

P3.4/T0 (Counter 0 External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 0 (เป็นอินพุตโหมดเคาน์เตอร์)

P3.5/T1 (Counter 1 External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 1 (เป็นอินพุตโหมดเคาน์เตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.6/$\overline{WR}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.7/$\overline{RD}$ (External Data Menory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

8) ALE (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตซ์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอตทีฟทุก ๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล

9) $\overline{PSEN}$ (ขา 29) เป็นขาสโตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภาย 11นอกสัญญาณนี้จะส่งออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้

10) $\overline{EA}$ (ขา 30) ใช้เลือกหน่วยความจำโปรแกรมภายนอก

ป้อน "0" จะอ่านโปรแกรมจากภายนอกชิพ

ป้อน "1" จะอ่านโปรแกรมจากภายในชิพ

11.) RST (ขา 9) ขารีเซ็ท จะรีเซ็ทได้ก็ต่อเมื่อป้อนลอจิก "1" เข้าที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล

12) XTAL1 (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิสเลเตอร์ภายใน

13) XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาต์พุตของวงจรออสซิสเลเตอร์ภายใน

การทดลองที่ 1

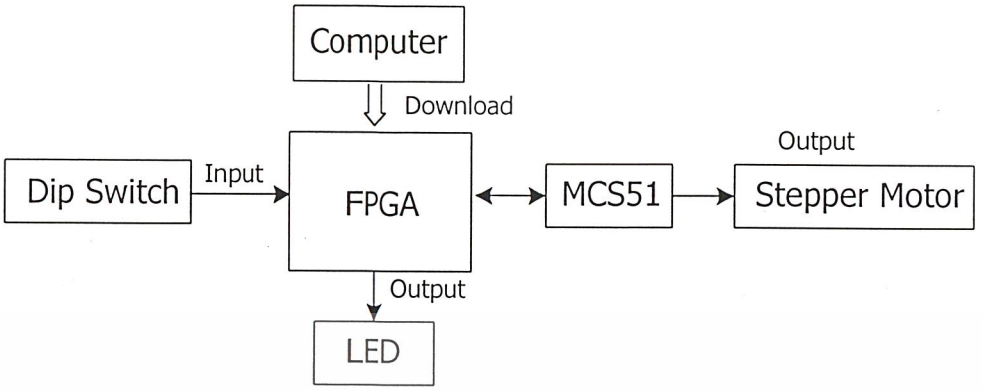
1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจกใหม่ และตั้งชื่อว่า LAB7_1

3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายการติดต่อกับ MCS51 โดยใน MCS51 จะเขียนโปรแกรมควบคุม Stepper motor โดย จะกำหนดให้ กำหนดหมุนไปทางซ้ายขวาได้ พร้อมทั้งรับค่าจาก MCS เข้า เอฟพีจีเอ แล้วส่งไปแสดงผลไฟวิ่ง

4) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.69 แผนผังการต่อแผงวงจรเอฟพีจีเอ

5) บันทึกผลการจำลองผลการทำงาน

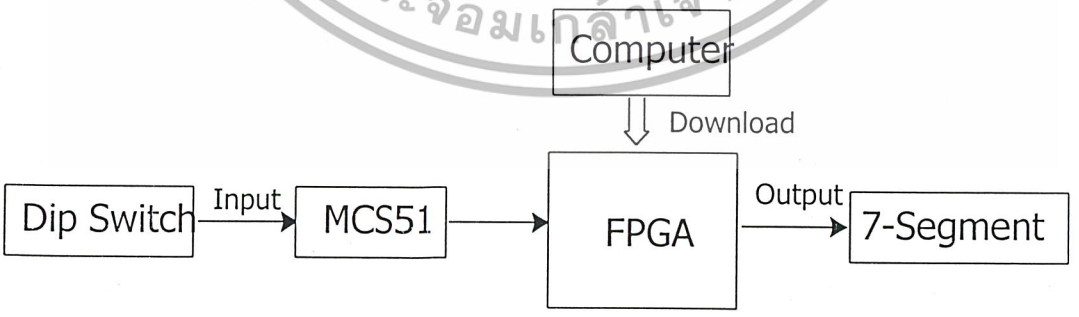
การทดลองที่ 2

1) เรียกโปรแกรม Xilinx Froundation 2.1I ขึ้นมา

2) สร้างโปรเจคใหม่ และตั้งชื่อว่า LAB7_2

3) เขียนโปรแกรมภาษาวีเอชดีแอลบรรยายการรับค่า เคาเตอร์ 0-9 จาก MCS51 มาแสดงผลออก โมดูลแสดงผลเจ็ดส่วน

4) ทดลองโดยการดาวน์โหลดลงบนแผงวงจรเอฟพีจีเอ พร้อมตรวจสอบการทำงานโดยต่อวงจรดังรูปที่ ค.70

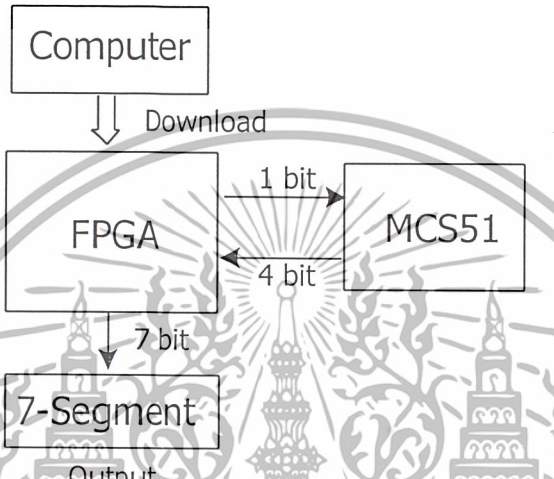


รูปที่ ค.70 แผนผังการต่อแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1) จงเขียนโปรแกรมภาษาวีเอชดีแอลบรรยายการรับส่งค่าระหว่าง เอฟพีจีเอ กับ MCS51 โดยเอฟพีจีเอส่งสัญญาณให้ MCS 51 แล้ว MCS 51 ส่งค่าเคาน์เตอร์ 0 – 9 เข้ามาที่เอฟพีจีเอแล้ว เอฟพีจีเอ แสดงผลออก 7 – Segment โดยวงจรดังรูปที่ ค.71



รูปที่ ค.71 แผนผังการต่อแผงวงจรเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
เฉลยใบงานประกอบการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1

การศึกษาโปรแกรม Xilinx Foundation F2.1I

เฉลยคำถามท้ายการทดลอง

1) สามารถออกแบบได้ 2 วิธี

1.1) ออกแบบโดยใช้ภาษาวีเอชดีแอล (VHDL)

1.2) ออกแบบโดยใช้ภาษาเวอร์ริล็อก (VERILOG)

2) ขั้นตอนการออกแบบวงจรรวมเกทที่ได้ทำการทดลอง มีดังนี้

2.1) เปิดโปรแกรม Xilinx Foundation F2.1I โดยคลิกที่ไอคอน Project Manager

2.2) สร้างโปรเจคใหม่ และตั้งชื่อโปรเจคว่า lab1

2.3) คลิกเลือก Flow แบบ HDL แล้วคลิก OK

2.4) เลือกการออกแบบ Design Entry แบบ HDL Editor

2.5) เขียนโปรแกรมโดยใช้ภาษาวีเอชดีแอลออกแบบแอนเกต ดังรูปที่ ง.1

```
library IEEE;
use IEEE.std_logic_1164.all;
entity and_1 is
    port (a: in STD_LOGIC;
          b: in STD_LOGIC;
          c: out STD_LOGIC);
end and_1;

architecture and_arch of and_1 is
begin
  c <= a and b;
end and_arch;
```

รูปที่ ง.1 โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต

7) Check Syntax ตรวจสอบความถูกต้องของโปรแกรม

8) เมื่อ Check Syntax ถูกต้องเรียบร้อยแล้ว ให้ส่งไฟล์เข้าสู่โปรเจคโดยคลิกที่เมนู Project แล้วเลือก Add To Project

9) ทำการ Synthesis โดยกำหนดเบอร์ไอซีตามที่ให้มาบนแผงวงจรทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) ทำการ Simulation เพื่อตรวจสอบการเลียนแบบการทำงาน

11) ทำการ Implementation

12) ทำการ Programming ลงในไอซีตามเบอร์ที่ให้มาบนแผงวงจรทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2

การใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอล ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

ผลการทดลองที่ 1

ผลการทดลองที่ 1 ดังแสดงในตารางที่ ง.1

ตารางที่ ง.1 ผลการทดลองที่ 1 ลักษณะการหมุนของมอเตอร์ไฟฟ้ากระแสตรง

Dipswith		ลักษณะการหมุนของ
1	44	มอเตอร์ไฟฟ้ากระแสตรง
0	1	หมุนขวา
1	0	หมุนซ้าย

ผลการทดลองที่ 2

ผลการทดลองที่ 2 ดังแสดงในตารางที่ ง.2

ตารางที่ ง.2 ตารางบันทึกผลการทดลอง

อินพุตคีย์บอร์ดเมตริกซ์	เอาต์พุตแสดงผลเจ็ดส่วน
0	0
1	1
2	2
3	3
4	4
5	5
6	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.2 (ต่อ) ตารางบันทึกผลการทดลอง

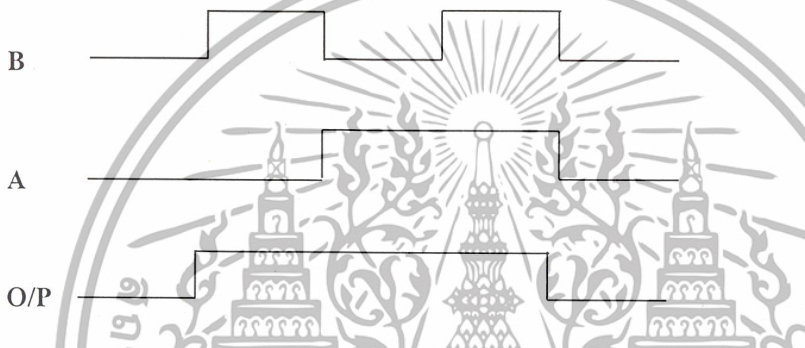
อินพุตคีย์บอร์ดเมตริกซ์	เอาต์พุตแสดงผลเจ็ดส่วน
7	7
8	8
9	9
a	a
b	b
c	c
d	d
e	e
f	f

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
การเขียนภาษาเอชดีแอลเบื้องต้น และการเลียนแบบการทำงาน

ผลการทดลองที่ 1

ผลของการจำลองผลการทำงานการทดลองที่ 1 ดังรูปที่ ง.2



รูปที่ ง.2 ผลของการจำลองผลการทำงาน

การทดลองที่ 2

เฉลยโปรแกรมภาษาวีเอชดีแอลแบบ Behavioral ดังรูปที่ ง.3

```
library IEEE;
use IEEE.std_logic_1164.all;
entity OR_Gate is port
(a,b : in std_logic ;
      o : out std_logic ) ;
end OR_Gate ;
architecture OR_arc of OR_Gate is
begin
process (a,b)
begin
if (a ='0' and b = '0') then
o <= '0' ;
```

รูปที่ ง.3 เฉลยโปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style

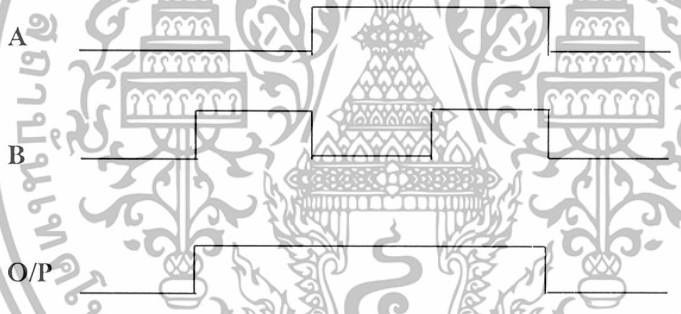
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elsif (a = '0' and b = '1') then
o <= '1' ;
elsif (a = '1' and b = '0') then
o <= '1' ;
elsif (a = '1' and b = '1') then
o <= '1' ;
end if;
end process ;
end OR_arc ;
```

รูปที่ ง.3 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลจาก Data Flow Style เป็น Behavioral Style

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 2 ดังรูปที่ ง.4



รูปที่ ง.4 ผลของการจำลองผลการทำงาน

การทดลองที่ 3

เฉลยโปรแกรมภาษาวีเอชดีแอล แบบ Data Flow บรรยายแอนเกต 2 อินพุต ดังรูปที่ ง.5

```
library IEEE ;
use IEEE.std_logic_1164.all ;
entity AND_Gate is
port (
     a,b : in std_logic ;
```

รูปที่ ง.5 เฉลยโปรแกรมภาษาวีเอชดีแอล แบบ Data Flow Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

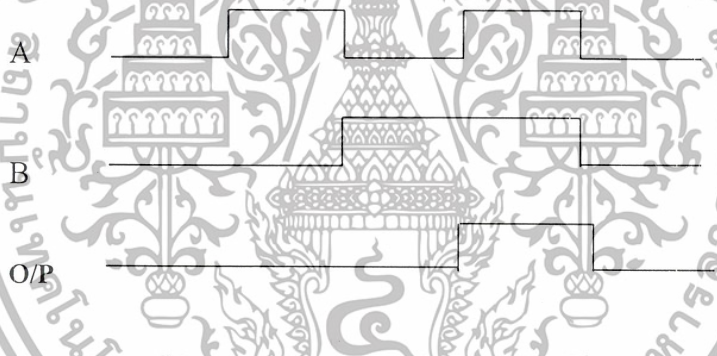
```
    o : out std_logic
 ) ;
end AND_Gate ;

architecture AND_arc of AND_Gate is
begin
o <= a and b ;
end AND_arc;
```

รูปที่ ง.5 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลแบบ Data Flow Style

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 3 ดังรูปที่ ง.6



รูปที่ ง.6 ผลของการจำลองผลการทำงาน

การทดลองที่ 4

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral วงจรแนนเกต 3 อินพุต ดังรูปที่ ง.7

```
library IEEE;
use IEEE.std_logic_1164.all;
entity nand3 is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
```

รูปที่ ง.7 ภาษาวีเอชดีแอลบรรยายแบบ Behavioral วงจรแนนเกต 3 อินพุต

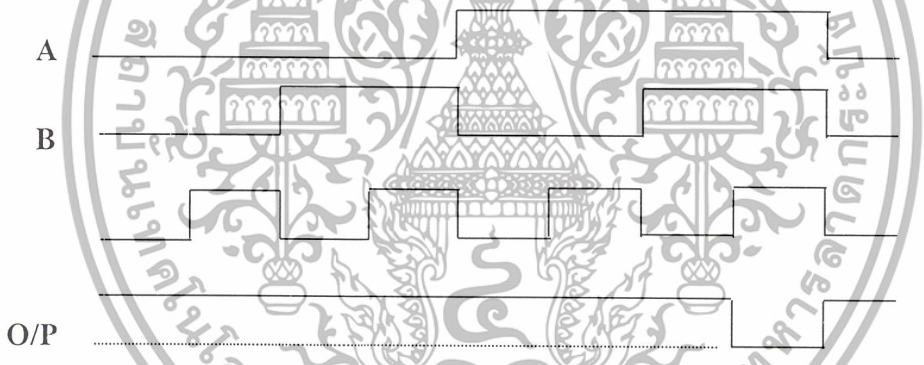
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
c: in STD_LOGIC;
        o: out STD_LOGIC
    );
end nand3;
architecture nand3_arch of nand3 is
begin
  o<= not (a and b and c);
end nand3_arch;
```

รูปที่ ง.7 (ต่อ) ภาษาวีเอชดีแอลบรรยายแบบ Behavioral วงจรแนนเกต 3 อินพุต

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 4 ดังรูปที่ ง.8



รูปที่ ง.8 ผลของการจำลองผลการทำงาน

การทดลองที่ 5

ภาษาวีเอชดีแอลบรรยายแบบ Data Flow ดังรูปที่ ง.9

```
library IEEE;
use IEEE.std_logic_1164.all;
entity nand3 is
    port (
        a: in STD_LOGIC;
```

รูปที่ ง.9 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายจาก Behavioral เป็น Data Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

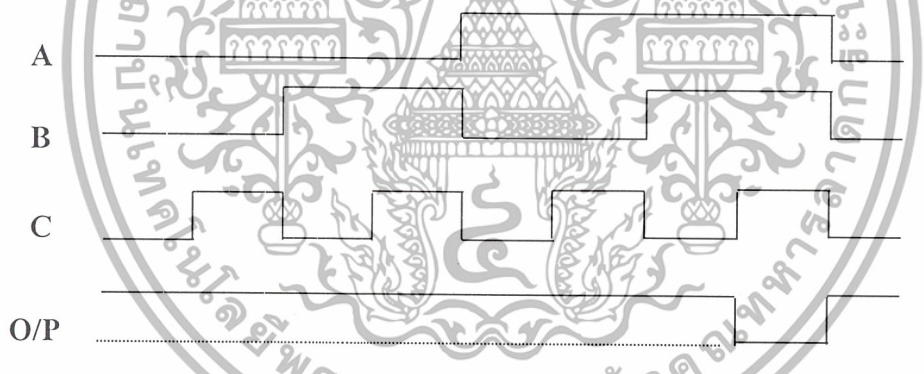
```
b: in STD_LOGIC;
        c: in STD_LOGIC;
        o: out STD_LOGIC
    );
end nand3;

architecture nand3_arch of nand3 is
begin
      process (a,b,c)
      begin
      if(a='1'and b='1' and c='1') then
```

รูปที่ ง.9 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายจาก Behavioral เป็น Data Flow

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 5 ดังรูปที่ ง.10



รูปที่ ง.10 ผลของการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

เฉลยตารางความจริงของโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral

ตารางที่ ง.3 ตารางความจริง

A	B	C	AB	$\overline{AB+A}$	AB+B	(AB+B)C	$\overline{(AB+A)}$+(AB+B)C
0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	1	0	0
1	1	1	1	0	1	1	1

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral ดังรูปที่ ง.11

```
library IEEE;
use IEEE.std_logic_1164.all;
entity mix is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        c: in STD_LOGIC;
        o: out STD_LOGIC
    );
end mix;

architecture mix_arch of mix is
begin
  process(a,b,c)
  begin
  if (a='1'and b='0' and c='0')then
  o<='0';
```

รูปที่ ง.11 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral

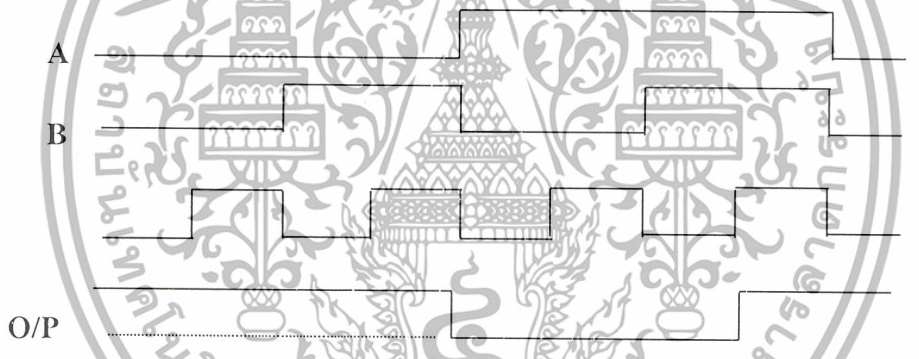
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elsif (a='1'and b='0' and c='1')then
  o<='0';
  elsif (a='1'and b='1' and c='0')then
  o<='0';
  else o<='1';
  end if;
  end process;
end mix_arch;
```

รูปที่ ง.11 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายแบบ Behavioral

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 6 ดังรูปที่ ง.12



รูปที่ ง.12 ผลของการจำลองผลการทำงาน

เฉลยคำถามท้ายการทดลอง

โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต 3 อินพุต ดังรูปที่ ง.13

```
Data flow  Style
      library IEEE;
      use IEEE.std_logic_1164.all;
      entity and3 is
          port (
```

รูปที่ ง.13 โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต 3 อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      a: in STD_LOGIC;
              b: in STD_LOGIC;
              c: in STD_LOGIC;
              o: out STD_LOGIC
          );
      end and3;
      architecture and3_arch of and3 is
      begin
        o<=  (a and b and c);
      end and3_arch;

Behavioral Style
      library IEEE;
      use IEEE.std_logic_1164.all;
      entity and3 is
          port (
              a: in STD_LOGIC;
              b: in STD_LOGIC;
              c: in STD_LOGIC;
              o: out STD_LOGIC
          );
      end and3;
      architecture and3_arch of and3 is
      begin
      procss (a,b,c)
              if (a='1'and b='1' and c='1')then
       o<= '1' ;
      else o<='0' ;
      end and3_arch;
```

รูปที่ ง.13 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายแอนเกต 3 อินพุต

2. การเขียนโปรแกรมแบบ Behavioral จะใช้การบรรยายการทำงานโดยการกำหนดว่าเมื่ออินพุตเข้ามาแบบนี้แล้วให้เอาต์พุตออกค่าไหน ส่วนแบบ Data Flow ใช้การบรรยายโดยการนำอินพุตมากระทำทางดิจิตอล (ผ่านคำสั่งพีชคณิตบูลีน) แล้วผลการกระทำก็จะเป็นเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
การออกแบบวงจรคอมบิเนชั่นด้วยภาษาวีเอชดีแอล ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

การทดลองที่ 1

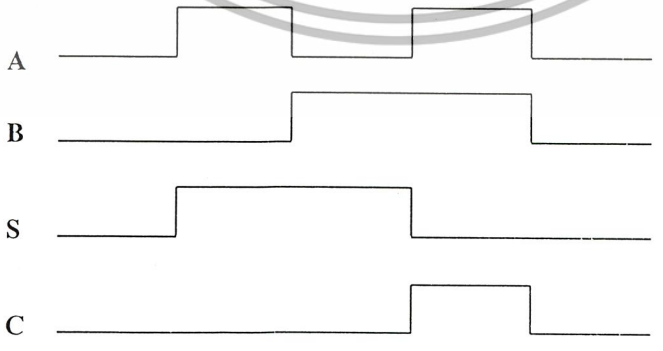
ผลการทดลองที่ 1 ดังแสดงในตารางที่ ง.4

ตารางที่ ง.4 ผลการทดลองที่ 1

อินพุต		เอาต์พุต	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ผลของการจำลองผลการทำงาน

ผลของการจำลองผลการทำงาน การทดลองที่ 1 ดังแสดงในรูปที่ ง.14



รูปที่ ง.14 ผลของการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็มโดยเขียนแบบโครงสร้างโดยสร้างจากวงจรตัวบวกครึ่ง (Half Adder) ดังรูปที่ ง.15

```
-- OR GATE
library IEEE;
use
IEEE.std_logic_1164.all;
entity or_1 is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        o: out STD_LOGIC
    );
end or_1;
architecture or_1_arch of
or_1 is
begin
 o<=a or b;
end or_1_arch;
```

รูปที่ ง.15 โปรแกรมภาษาวีเอชดีแอลบรรยายออร์เกต

```
-- Half Adder
library IEEE;
use IEEE.std_logic_1164.all;
entity ha is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        s: out STD_LOGIC;
        c: out STD_LOGIC
    );
end ha;
architecture ha_arch of ha is
begin
  s<= a xor b;
  c<= a and b;
end ha_arch;
```

รูปที่ ง.16 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกครึ่ง (Half Adder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library IEEE;
use IEEE.std_logic_1164.all;
entity fa_pm is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        c_in: in STD_LOGIC;
        sum: out STD_LOGIC;
        c_out: out STD_LOGIC
    );
end fa_pm;
architecture fa_pm_arch of fa_pm is
component ha
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        s: out STD_LOGIC;
        c: out STD_LOGIC
    );
end component;
component or_1
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        o: out STD_LOGIC
    );
end component;
signal w_1,w_2,w_3:STD_LOGIC;
begin
  u1: ha port map (a,b,w_1,w_2);
  u2: ha port map
(w_1,c_in,sum,w_3);
  u3: or_1 port map
(w_3,w_2,c_out);
end fa_pm_arch;
```

รูปที่ ง.17 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็ม (Full Adder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

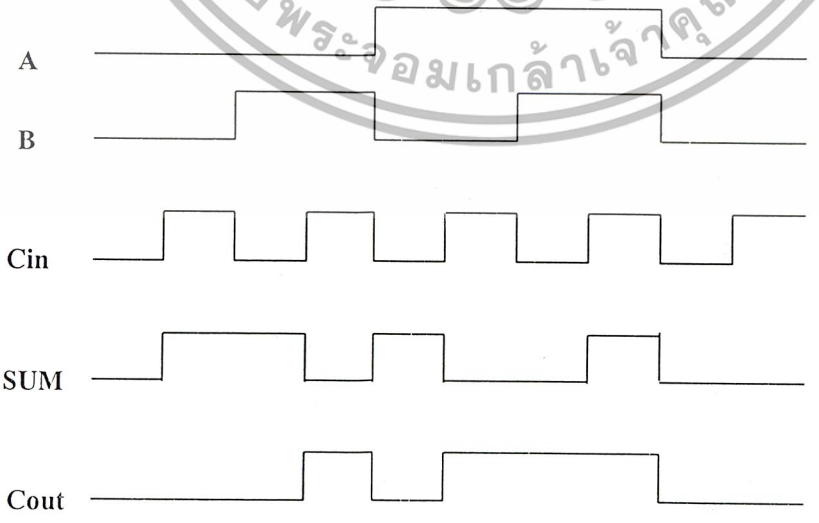
ผลการทดลองที่ 1 ดังแสดงในตารางที่ ง.18

ตารางที่ ง.5 ผลการทดลองที่ 1

อินพุต			เอาต์พุต	
A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ผลการจำลองผลการทำงาน

ผลของการจำลองผลการทำงานการทดลองที่ 1 ดังแสดงในรูปที่ ง.18



รูปที่ ง.18 ผลของการจำลองผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

เฉลยโปรมแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็มดังรูปที่ ง.19

```
library IEEE;
use IEEE.std_logic_1164.all;
entity FA is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        c_i: in STD_LOGIC;
        s: out STD_LOGIC;
        c_o: out STD_LOGIC
    );
end FA;
architecture FA_arch of FA is
begin
     s<=a xor b xor c_i;
  c_o<=((a xor b) and c_i) or (a and b);
 end FA_arch;
```

รูปที่ ง.19 เฉลยโปรมแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็ม

ผลการเลียนแบบการทำงาน

ผลของการจำลองผลการทำงาน การทดลองที่ 3 ดังแสดงในรูปที่ ง.20

รูปที่ ง.20 เฉลยโปรมแกรมภาษาวีเอชดีแอลบรรยายตัวบวกเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองที่ 3 ดังแสดงในตารางที่ ง.6

ตารางที่ ง.6 ผลการทดลองที่ 3

อินพุต			เอาต์พุต	
A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

การทดลองที่ 4

```
library IEEE;
use IEEE.std_logic_1164.all;
entity encode is
    port (
        a: in STD_LOGIC_VECTOR (0 to 8);
        o: out STD_LOGIC_VECTOR (0 to 3)
    );
end encode;
architecture encode_arch of encode is
begin
  with a select
  o<="0001" when "100000000",
     "0010" when "010000000",
     "0011" when "001000000",
     "0100" when "000100000",
```

รูปที่ ง.21 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรเข้ารหัสจากตารางความจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      "0101" when "000010000",
      "0110" when "000001000",
      "0111" when "000000100",
      "1000" when "000000010",
      "1001" when "000000001",
      "1111" when others;
end encode_arch;
```

รูปที่ ง.21 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรเข้ารหัสจากตารางความจริง

ผลการทดลองที่ 4 ดังแสดงในตารางที่ ง.7

ตารางที่ ง.7 ผลการทดลองที่ 4

	Input									Output			
	S1	S2	S3	S4	S5	S6	S7	S8	S9	Q3	Q2	Q1	Q0
1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	1	0	0	0	0	1	1
8	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	1	1	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

ผลการทดลองที่ 5 ดังแสดงในตารางที่ ง.8

ตารางที่ ง.8 เฉลยตารางความจริง

	A	B	C	D	g	f	e	d	c	b	a
1	0	0	0	1	1	1	1	1	0	0	1
2	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0	0	0	0
4	0	1	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	0	1	0	0	1	0
6	0	1	1	0	0	0	0	0	0	1	0
7	0	1	1	1	1	1	1	1	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	0	0	0	0

```
library IEEE;
use IEEE.std_logic_1164.all;
entity BCD_7 is
    port (
        a: in STD_LOGIC_VECTOR (3 downto
0);
        o: out STD_LOGIC_VECTOR (6 downto
0)
    );
end BCD_7;
architecture BCD_7_arch of BCD_7 is
begin
  With a select
  o<= "1111001" when "0001", --1
      "0100100" when "0010", --2
      "0110000" when "0011", --3
```

รูปที่ ง.22 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยาย BCD to 7 Segment (1 – 9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        "0011001" when "0100", --4
        "0010010" when "0101", --5
        "0000010" when "0110", --6
        "1111000" when "0111", --7
        "0000000" when "1000", --8
        "0010000" when "1001", --9
        "1111111" when others;
end BCD_7_arch;
```

รูปที่ ง.22 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยาย BCD to 7 Segment (1 – 9)

ผลการทดลอง

เมื่อทำการดาวน์โหลดโปรแกรมลงแผงวงจรเอฟพีจีเอและทำการทดลองเรียบร้อยแล้วก็จะได้ผลดังนี้ เมื่อ

Switch เป็น 0001 จะแสดงผลออกเป็น 1

Switch เป็น 0010 จะแสดงผลออกเป็น 2

Switch เป็น 0011 จะแสดงผลออกเป็น 3

Switch เป็น 0100 จะแสดงผลออกเป็น 4

Switch เป็น 0101 จะแสดงผลออกเป็น 5

Switch เป็น 0110 จะแสดงผลออกเป็น 6

Switch เป็น 0111 จะแสดงผลออกเป็น 7

Switch เป็น 1000 จะแสดงผลออกเป็น 8

Switch เป็น 1001 จะแสดงผลออกเป็น 9

นอกเหนือจากนี้จะกำหนดให้ไม่แสดงผลใด ๆ เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยาย ALU 15 คำสั่ง ดังรูปที่ ง.23

```
Library IEEE;
Use IEEE.STD _LOGIC_1164.all, IEEE.NUMERIC_STD.all;
Entity ALU is
      Port (Sel: in    unsigned (4 downto 0);
            Carryln:   in    std_logic;
                A, B: in    unsigned (7 downto 0);
                   Y: out   unsigned (7 downto 0));
End entity ALU:
Architecture COND_DATA_FLOW of ALU is
Begin
       ALU_AND_SHIFT:
       Process (Sel, A, B, Carryln)
      Variable Sel0_1_Carryln: unsigned (2 downto 0);
      Variable LogicUnit, ArithUnit,
           ALU_NoShift: unsigned (7 downto 0);
       Begin
      ---------------
- Logic Unit
      ---------------
      LOGIC_UNIT: case Sel (1 downto 0) is
             when "00"  =>  LogicUnit  :=  A and B;
             when "01"  =>  LogicUnit  :=  A or B;
             when "10"  =>  LogicUnit  :=  A xor B;
             when "11"  =>  LogicUnit  :=  not A;
             when others => LogicUnit  :=  (others => 'x');
      end case LOGIC_UNIT;
      ---------------------
- Arithmetic Unit
      ---------------------
      Sel0_1_Carryln :=  Sel(1 downto 0) & Carryln;
      ARITH_UNIT: case Sel0_1_Carryln in
             when "000"     =>    ArithUnit  :=  A;
             when "001"     =>    ArithUnit  :=  A + 1;
             when "010"     =>    ArithUnit  :=  A + B;
             when "011"     =>    ArithUnit  :=  A + B + 1;
             when "100"     =>    ArithUnit  :=  A + not + B;
             when "101"     =>    ArithUnit  :=  A - B;
             when "110"     =>    ArithUnit  :=  A - 1;
             when "111"     =>    ArithUnit  :=  A;
             when others    =>    LogicUnit  :=(others=>'x');
      end case ARITH_UNIT;
```

รูปที่ ง.23 โปรแกรมภาษาวีเอชดีแอล บรรยาย ALU 15 คำสั่งเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-------------------------------------------------------
- Multiplex between Login & Arithmetic Units
     -------------------------------------------------------
--
     LA_MUX: if (Sel(2) = '1') then
            ALU_NoShift := LogicUnit;
     else
            ALU_Noshift := ArithUnit;
     end if LA_MUX;

     ---------------------
- Shift operations
     ---------------------
     SHIFT : Sel(4 downto 3) is
            when "00"      =>   Y   <=      ALU_NoShift;
            when "01"      => Y   <=
     Shift_left(ALU_NoShift, 1);
            when "10"      => Y   <=
     Shift_right(ALU_NoShift, 1);
            when "11"      => Y   <=  (others => '0');
            when others    => Y   <=  (others => 'x');
     end case SHIFT;
    end process ALU_AND_SHIFT;
end architecture CONE_DATA_FLOW;
```

รูปที่ ง.23 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายเอแอลยู15 คำสั่งเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยคำถามท้ายการทดลอง

```
library IEEE;
use IEEE.std_logic_1164.all;

entity digitol_gate is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        s: in STD_LOGIC_VECTOR (2 downto 0);
        o: out STD_LOGIC
    );
end digitol_gate;
architecture digitol_gate_arch of digitol_gate is
begin
     process(s)
      begin
      case s is
            when "000"      =>    o<= a or b;
            when "001"      =>    o<= a and  b;
            when "010"      =>    o<=    not  a;
            when "011"      =>    o<=    not  b;
            when "100"      =>    o<= a nor  b;
            when "101"      =>    o<= a nand b;
            when "110"      =>    o<= a xor  b;
            when "111"      =>    o<= a xnor b;
            when others     =>    o<= '0';
      end case ;
       end process;
-- <<enter your statements here>>
end digitol_gate_arch;
```

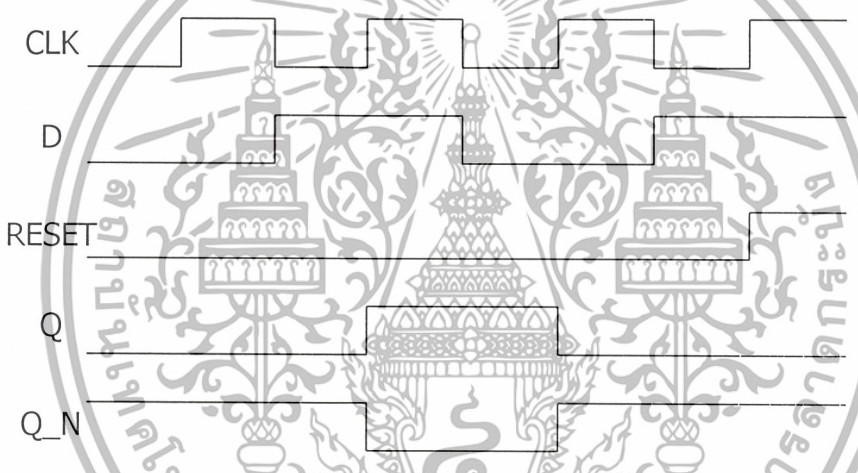
รูปที่ ง.24 โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่5
การออกแบบวงจรซีเควนเชียลด้วยภาษาวีเอชดีแอล
ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

การทดลองที่ 1

ผลการจำลองการทำงาน



รูปที่ ง.25 ผลของการจำลองผลการทำงาน

ผลการจำลองผลการทำงานวงจรทำงานเมื่อมี Clock เข้ามาและเอาต์พุตจะขึ้นอยู่กับอินพุท ดี เมื่อ Reset เป็น “1” O จะเป็น “0” ลักษณะการทำงานจะเหมือนกับการเลียนแบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายตัวเลื่อนข้อมูลตามรูปที่ ง.26

```
-- 4 - bit Shift Register with Reset
library ieee;
use ieee.std_logic_1164.all;
entity srg_4_r is
     Port(CLK, RESET, SI : in std_logic;
          Q : out std_logic_vector (3 downto 0);
         SO : out std_logic
         );
end srg_4_r ;
architecture bh of srg_4_r is
signal shift : std_logic_vector  (3 downto 0);
begin
        if(RESET = '1') then
        shift <= "0000";
      elsif(CLK'event and CLK = '1') then
        shift <=shift(2 downto 0 ) & SI ;
     end if;
end process;
Q <= shift;
SO <= shift(3);
End bh;

process(RESET,CLK)
```

รูปที่ ง.26 โปรแกรมภาษาวีเอชดีแอลบรรยายตัวเลื่อนข้อมูล

ผลการทดลอง

ผลการทำงานของวงจรจะทำการเลื่อนบิตที่ เอาต์พุตจาก Q0 ไปที่ Q1 จาก Q1 ไป Q2 จาก Q2 ไปที่ Q3 โดย S1 จะเป็นอินพุตของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายตัวนับขึ้น 4 บิต (Counter) ดังแสดงในรูปที่ ง.27

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.Numeric_STD.all;
entity counter4b is
    port (
        ck : in  STD_LOGIC;
        reset : in  STD_LOGIC;
            o : out INTEGER range 0 to 15
    );
end counter4b;

architecture counter4b_arch of counter4b is
signal    count : integer range 0 to 15;
begin
     process(ck,reset)
        begin

            if (reset = '1' ) then
                count <= 0;
            elsif( ck ='1'and ck'event ) then
                count <= count+1;
            end if;
            if (count>15) then count <=0;
            end if;
     end process;
        o<= count;
end counter4b_arch;
```

รูปที่ ง.27 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายตัวนับขึ้น 4 บิต (Counter)

ผลการทดลอง

จากการต่อวงจรเมื่อป้อนสัญญานนาฬิกาให้กับวงจรเอาพุตก็จะนับขึ้นตั้งแต่ “0000” จน ถึง “1111” และวนมานับ “0000” ใหม่อีกครั้งวนไปเลื่อย ๆ เมื่อกด Reset เอาต์พุตจะเป็น “0000”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรนับ 5 บิต (Counter)โดยให้นับขึ้นทีละหนึ่ง นับลงทีละสอง ดังแสดงในรูปที่ ง.28

```
Library IEEE;
Use IEEE.STD_Logic_1164.all, IEEE.Numeric_STD.all ;

Entity CNT_UP1_DOWN2 is
     Port (Clock, Reset, Up, Down : in std_logic;
     Count: out unsigned (4 downto 0));
end entity CNT_UP1_DOWN2 ;


architecture RTL of CNT_UP_DOWN2 is
begin
process UpDown: unsigned (1 downto 0)
     variable UpDown : unsigned(1 downto 0);
     variable count_v : unsigned(4 downto 0);
     begin
     UpDown := Up&Down;
     ef rising_edge(Clock) then
          if (Reset = '1') then
          Count_v := "0000";
           else
          case UpDown is
               when "00" =>   Count_v  :=  Count_v;
               when "01" =>   Count_v  :=  Count_v + 1;
               when "10" =>   Count_v  :=  Count_v - 2;
               when others => Count_v  :=  Count_v;
          end case;
            end if;
     end if;
     Count  <=  Count_v
       end process;
end architecture RTL;
```

รูปที่ ง.28 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรนับ 5 บิต (Counter)

ผลการทดลอง

เมื่อจ่ายสัญญาณนาฬิกาให้กับวงจรถ้าเลือกให้ Up=1 เอาท์พุตจะนับขึ้นที่ละ 1 จนถึง "1111" ถ้า Down เป็น 1 ก็นับที่ละ 2จนถึง "0000" และถ้าให้ Reset = '1' เอาท์พุตจะ เท่ากับ '0'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยคำถามท้ายการทดลอง

1) เฉลยโปรแกรมภาษาวีเอชดีแอล นับขึ้นทีละ 1 พร้อมทั้งแสดงผลออกโมดูลแสดงผลเจ็ดส่วน ตั้งแต่ 0-9 ตามรูปที่ ง.29

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.Numeric_STD.all;
entity counter4b is
    port (
        ck : in  STD_LOGIC;
        reset : in  STD_LOGIC;
            o : out INTEGER range 0 to 15
    );
end counter4b;

architecture counter4b_arch of counter4b is
signal    count : integer range 0 to 9;
begin
     process(ck,reset)
       begin

          if (reset ='1') then
             count <= 0;
          elsif( ck ='1'and ck'event ) then
               count <= count+1 ;
          end if;
          if (count>9) then count <=0;
          end if;
     end process;
       o<= count;
end counter4b_arch;

library IEEE;
use IEEE.std_logic_1164.all;

entity BCD_7 is
    port (
        a: in  integer range 0 to 9;
        o: out STD_LOGIC_VECTOR (6 downto 0)
    );
end BCD_7;
```

รูปที่ ง.29 โปรแกรมภาษาวีเอชดีแอลนับขึ้นทีละ 1 พร้อมทั้งแสดงผลออกโมดูลแสดงผลเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
architecture BCD_7_arch of BCD_7 is
begin
  With a select
  o<= "1111001" when 1, --1
      "0100100" when 2, --2
      "0110000" when 3, --3
      "0011001" when 4, --4
      "0010010" when 5, --5
      "0000010" when 6, --6
      "1111000" when 7, --7
      "0000000" when 8, --8
      "0010000" when 9, --9
      "1000000" when 0, --0
      "1111111" when others;
end BCD_7_arch;
library IEEE;
use IEEE.std_logic_1164.all;
entity count_to_seg is
    port (
        clk: in STD_LOGIC;
        rst: in STD_LOGIC;
        o_7seg:out STD_LOGIC_VECTOR (6 downto 0)
    );
end count_to_seg;
architecture count_to_seg_arch of count_to_seg is
signal l:integer range 0 to 9 ;
component  counter4b
    port (
        ck : in  STD_LOGIC;
        reset : in  STD_LOGIC;
            o : out INTEGER range 0 to 9
    );
end component ;

component BCD_7
    port (
        a: in  integer range 0 to 9;
        o: out STD_LOGIC_VECTOR (6 downto 0)
    );
end component;
begin
  -- <<enter your statements here>>
  counter_i_p  :counter4b port map( clk,rst,l);
  decode_to_seg:BCD_7 port map(l,o_7seg);
end count_to_seg_arch;
```

รูปที่ ง.29 (ต่อ) โปรแกรมภาษาวีเอชดีแอลนับขึ้นทีละ1พร้อมทั้งแสดงผลออกโมดูลแสดงผลเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 6
การออกแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอล
ร่วมกับอุปกรณ์เอฟพีจีเอ และซีพีแอลดี

การทดลองที่ 1

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยาย การทำงานของระบบควบคุมสี่แยกไฟแดง ดังรูปที่ ง.30

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.Numeric_STD.all;
entity counter4b is
    port (
        ck : in  STD_LOGIC;
      reset : in  STD_LOGIC;
          o : out integer range 0 to 8;
          c : out integer range 0 to 512
    );
end counter4b;

architecture counter4b_arch of counter4b is

signal       op: integer range 0 to 8;
signal    count : integer range 0 to 512;
begin
     process(ck,reset)
          begin
              op<=0;
              if (reset = '1' ) then
                  count <= 0;
                  op<=0;
              elsif( ck ='1'and ck'event ) then
                   count <= count+1;
              end if;
```

รูปที่ ง.30 โปรแกรมภาษาวีเอชดีแอลบรรยายการทำงานของระบบควบคุมสี่แยกไฟแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (count>512) then
                count <=0;
               end if;
               if (count < 128) then
                  op <= 8;
               end if;

            if (count >128  and count<256) then
                  op <=4;
               end if;
                 if (count>256 and count<384) then
                  op <= 2;
               end if;
                if (count >384 and count<512) then
                  op <= 1;
               end if;
      end process;
         o<= op;
         c<=count;
end counter4b_arch;
```

รูปที่ ง.30 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายการทำงานของระบบควบคุมสี่แยกไฟแดง

ผลการทดลอง

เมื่อทำการต่อวงจรและดาวน์โหลดโปรแกรมสี่แยกไฟแดงจะทำงานตามปกติซึ่งจะสามารถ ปรับความเร็วของการเปลี่ยนสัญญาณไฟได้โดยการปรับสัญญาณนาฬิกา

การทดลองที่ 2

เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายระบบการควบคุมสเต็ปปิ้งมอเตอร์ ดังรูปที่ ง.31

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.Numeric_STD.all;
entity counter4b is
    port (
        ck    : in  STD_LOGIC;
        enable: in  STD_LOGIC;
        lr    : in  STD_LOGIC;
```

รูปที่ ง.31 เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายระบบการควบคุมสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            o : out integer range 0 to 8;
            c : out integer range 0 to 512
    );
end counter4b;
architecture counter4b_arch of counter4b is

signal        op: integer range 0 to 8;
signal    count : integer range 0 to 512;
begin
     process(ck,enable)
begin
               op<=0;
               if (enable = '0') then
                   count <= 0;
                   op<=1;
               elsif(ck ='1'and ck'event  ) then
                    count <= count+1;
               end if;

               if (count>512 ) then
                   count <=0;
               end if;
               if (count < 128 and lr='1') then
                  op <= 8;
               end if;
if (count >128  and count<256 and lr='1') then
                  op <=4;
               end if;

                if (count>256 and count<384 and lr='1') then
                  op <= 2;
               end if;
                if (count >384 and count<512 and lr='1') then
                  op <= 1;
               end if;
               if (count < 128 and lr='0') then
                  op <= 1;
               end if;
               if (count >128  and count<256 and lr='0') then
                  op <=2;
               end if;
                 if (count>256 and count<384 and lr='0') then
                  op <= 4;
               end if;
                if (count >384 and count<512 and lr='0') then
                  op <= 8;
               end if;
```

รูปที่ ง.31 (ต่อ) เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายระบบการควบคุมสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (count>256 and count<384 and lr='1') then
               op <= 2;
            end if;
            if (count >384 and count<512 and lr='1') then
               op <= 1;
            end if;
            if (count < 128 and lr='0') then
               op <= 1;
            end if;
            if (count >128  and count<256 and lr='0') then
               op <=2;
            end if;
              if (count>256 and count<384 and lr='0') then
               op <= 4;
            end if;
             if (count >384 and count<512 and lr='0') then
               op <= 8;
            end if;
      end process;
         o<= op;
         c<=count;
```

รูปที่ ง.31 (ต่อ) โปรแกรมภาษาวีเอชดีแอลบรรยายระบบการควบคุมสเต็ปปิ้งมอเตอร์

ผลการทดลอง

เมื่อทำการต่องวงจรดาวน์โหลดโปรแกรม และป้อนสัญญาณนาฬิกาให้กับวงจรเมื่อขา Enable = 1 สเต็ปปิ้งมอเตอร์ก็จะหมุนตามสัญญาณที่ควบคุมขา lr ซึ่ง

lr เป็น 0 จะหมุนซ้าย

lr เป็น 1 จะหมุนขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

เฉลยโปรแกรมภาษาวีเอชดีแอลโดยรับค่าจากวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล มาเก็บไว้ใน แรม 15 ตำแหน่งดังรูปที่ ง.32

```
library IEEE;
use IEEE.std_logic_1164.all;
entity a_to_ram is
    port (
        enable: in STD_LOGIC;
        address_o: out integer range 0 to 15 ;
        wr: out STD_LOGIC;
        oe: out STD_LOGIC
    );
end a_to_ram;

architecture a_to_ram_arch of a_to_ram is
signal  address:integer range 0 to 15 ;
begin
process(enable)
  begin
     if(enable='0') then
        oe<='0';
        address <= 0;
        wr<='1';
     else
     if(enable='1') then
        oe<='1';
        address <= address + 1;
        wr<='0';
     end if;
     end if;
     if(address > 15 ) then
        wr<='1';
        oe<='0';
        address <= 0;
     end if;
     address_o <= address;
end process;

  -- <<enter your statements here>>
end a_to_ram_arch;
```

รูปที่ ง.32 โปรแกรมภาษาวีเอชดีแอลโดยรับค่าจากวงจร แปลงสัญญานแอนะลอกเป็นดิจิตอล มาเก็บไว้ใน แรม 15 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

ในแรมจะมีข้อมูลที่รับจากแอนะลอกเป็นดิจิตอล อยู่ 15 ตำแหน่ง ซึ่งเมื่อนำข้อมูลจากแรมมาแสดงผลก็จะมีค่าตรงกับที่อยู่ใน แอนะลอกเป็นดิจิตอล

การทดลองที่ 4

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายวงจรคูณเลขฐาน 2(4 บิต) โดย ดึงค่าตัวเลขในอีพรอมดังแสดงในรูปที่ ง.33

```
Library IEEE
Use IEEE.std_logic.all
Entity mul is
(a,b:in INTEGER range 0 to 15;
o: out INTEGER range 0 to 255
    );
end mul;

architecture mul_arch of mul is
begin
a<=a*b;
end mul_arch;
```

รูปที่ ง.33 โปรแกรมภาษาวีเอชดีแอลบรรยายวงจรคูณเลขฐาน 2(4 บิต) โดย ดึงค่าตัวเลขในอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายควบคุมการทำงานของวงจรแปลงสัญญานดิจิตอลเป็นแอนะลอก โดยให้รับค่าจากสวิตช์เมททริกซ์ ดังแสดงในรูปที่ ง.34

```
library IEEE;
use IEEE.std_logic_1164.all;

entity d_to_a is
    port (
        input: in STD_LOGIC_VECTOR (3 downto 0);
        o: out STD_LOGIC_VECTOR (7 downto 0)
    );
end d_to_a;

architecture d_to_a_arch of d_to_a is
     begin
     process(input)
        begin
            if(input="0001") then o<="00000000";
          elsif(input="0010") then o<="01010000";

          elsif(input="0011") then o<="10010000";
          elsif(input="0100") then o<="11001001";

          elsif(input="0101") then o<="11111111";
          else o<="00000000";
             end if;
     end process;
end d_to_a_arch;
```

รูปที่ ง.34 เฉลยโปรแกรมภาษาวีเอชดีแอล บรรยายควบคุมการทำงานของ วงจรแปลงสัญญานดิจิตอลเป็นแอนะลอก โดยให้รับค่าจากสวิตช์เมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยคำถามท้ายการทดลอง

เฉลยคำถามท้ายการทดลองดังแสดงในรูปที่ ง.35

```
      package AM_PM_Package is
           type AM_PM_type is (AM,PM);
           function "not" (Value: AM_PM_type) return
AM_PM_type;
      and;

      package body AM_PM_Package is
          function "not" (Value: AM_PM_type) return
AM_PM_type is
          begin
            if value = AM then
                 return PM;
            else
                 return AM;
            end if;
          end "not";
      end AM_PM_Package;
library IEEE;
use IEEE.STD_Logic_1164.all,IEEE.Numeric_STD.all;
use work.AM_PM_Package.all;

entity ALARM_CLOCK is
    port ( Clock_1sec:          in    std_logic;
      Reset:                    in    std_logic;
      LoadTime:           in    std_logic;
      SetHours:           in    integer range 0 to 11;
      SetMins, Setsecs:   in    integer range 0 to 59;
      Set_AM_PM:          in    AM_PM_type;
      LoadAlm:            in    std_logic;
      AlarmHoursin:             in    integer range 0 to 11;
      AlarmMinsin:              in    integer range 0 to 59;
      Alarm_AM_PM_in:     in    AM_PM_type;
      AlarmEnable               in    std_logic;
Hours:                  out   integer range 0 to 11;
Mins, Secs:             out   integer range 0 to 59;
AM_PM;            out AM_PM_type;
Flashing:         out std_logic;
Alarm:                  out std_logic );
      End entity ALARM_CLOCK;
```

รูปที่ ง.35 โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Architecture RTL of ALARM_CLOCK is
    Signal Hours_s:          integer range 0 to 11;
  Signal Mins_s,Secs_s:      integer range 0 to 59;
  Signal AM_PM_s:            AM_PM_type;
  Signal Flashing_s:         std_logic;

Begin
    --**************************************
    --Calculate the next value of time:
    --        Secs Mins, Hours, AM_PM & Flashing.
  --**************************************
  TIMER: process (Clock_1sec)
  Begin
      If rising_edge(clock_1sec) then
      -- Synchronous Reset
      if (Reset = '1') then
            Hours_s       <=0;
            Mins_s     <=0;
            Sec_s         <=0;
            AM_PM_s <= AM;
            Flashing_s <= '1';
      -- Set the time
      elseif (LoadTime = '1') then
            Hours_s       <= SetHours;
            Mins_s     <= SetMins;
            Sec_s         <= SetSecs;
            AM_PM_s   <= Set_AM_PM;
            Flashing_s   <= '0';
      -- Increment time
      ----------------------
      else
          Flashing_s <= Flashing_s;
         If (Secs_s = 59) Then
             Sec_s <= 0;
             If (Mins_s = 59) then
            Mins_s <= 0;
            If (Hours_s = 11) then
                 Hours_s <= 0;
                 AM_PM_s <= not AM_PM_s;
            else
                 Hours_s <= Hours_s + 1;
                AM_PM_s <= AM_PM_s;
               end if;
                   else
                Mins_s  <= Mins_s +1;
                Hours_s <= Hours_s;
                AM_PM_s <= AM_PM_s;
               end if;
```

รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
                Secs_s <= Secs_s + 1;
Mins_s <= Mins_s;
                Hours_s <= Hours_s;
                AM_PM_s <= AM_PM_s;
              end if;
           end if;
      end if;
      Hours      <= Hours_s;
      Mins  <= Mins_s;
      Secs  <= Secs_s;
      AM_PM   <= AM_PM_s;
end process TIMER;
--*****************************************
--Store set alarm time when "LoadALM" active and
--compare current time with set alarm time.
--*****************************************

      ALARM_LOAD_AND_TEST:
      Process (Clock_1sec)
           Variable AlarmMins: integer range 0 to 59;
           Variable AlarmHours: integer range 0 to 11;
           Variable Alarm_AM_PM : AM_PM_type;
      Begin
           -- store set alarm time when "LoadAlm" active.
           If rising_edge(Clock_1sec) then
              If (reset = '1') then
                   AlarmMins := 0;
                   AlarmHours := 0;
                   Alarm_AM_PM := AM;
              elseif (LoadAlm = '1') then
                   AlarmMins := AlarmMinsin;
                   AlarmHours := AlarmHoursin;
                   Alarm_AM_PM := Alarm_AM_PM_In;
              else
                   AlarmMins := AlarmMins;
                   AlarmHours := AlarmHours;
                   Alarm_AM_PM := Alarm_AM_PM;
              end if;
           end if;
           --compare current time with set the alarm time.
           --Sets alarm for 1 minute (ignores seconds)
           if (Min_s = AlarmMins and
                Hours_s = AlarmHours and
                AM_PM_s = Alarm_AM_PM and
                AlarmEnable = '1') then
                Alarm <= '1';
```

รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
                Alarm <= '0';
            end if;
end process ALARM_LOAD_AND_TEST;
end architecture RTL;
```

รูปที่ ง.35 (ต่อ) โปรแกรมภาษาวีเอชดีแอลเฉลยคำถามท้ายการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 7

การติดต่อระหว่าง FPGA กับ MCS 51

การทดลองที่ 1

เฉลยโปรแกรมภาษาวีเอชดีแอลติดต่อกับ MCS 8051 ดังรูปที่ ง.36

```
library IEEE;
use IEEE.std_logic_1164.all;

entity step_motor is
    port (
        on_off     : in bit;
        left_right: in bit;
        in_step    : in bit_vector(3 downto 0);
        o          : out bit_vector(1 downto 0);
        out_step   : out bit_vector(3 downto 0)

    );
end step_motor;

architecture step_motor_arch of step_motor is
begin
     process(on_off,left_right)
      begin
         if(on_off ='1'and left_right='1')   then
                    o<= "10";
                out_step <=  in_step;
         elsif(on_off ='1'and left_right='0')   then
                  o<="01";
                out_step <=  in_step;
         else
              o<="00";
              out_step<="0000" ;
         end if;
end process;

end step_motor_arch;
```

รูปที่ ง.36 โปรแกรมภาษาวีเอชดีแอลติดต่อกับ MCS 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

เฉลยโปรแกรมภาษาวีเอชดีแอลบรรยายการรับค่าเคาน์เตอร์ 0-9 จาก MCS 8051 มาแสดงผลออกโมดูลแสดงผลเจ็ดส่วน ดังแสดงในรูปที่ ง.37

```
library IEEE;
use IEEE.std_logic_1164.all;

entity BCD_7 is
    port (
        a: in STD_LOGIC_VECTOR (3 downto 0);
        o: out STD_LOGIC_VECTOR (6 downto 0)
    );
end BCD_7;

architecture BCD_7_arch of BCD_7 is
begin
  With a select
  o<= "1111001" when "0001", --1
      "0100100" when "0010", --2
      "0110000" when "0011", --3
      "0011001" when "0100", --4
      "0010010" when "0101", --5
      "0000010" when "0110", --6
      "1111000" when "0111", --7
      "0000000" when "1000", --8
      "0010000" when "1001", --9
      "1111111" when others;
end BCD_7_arch;
```

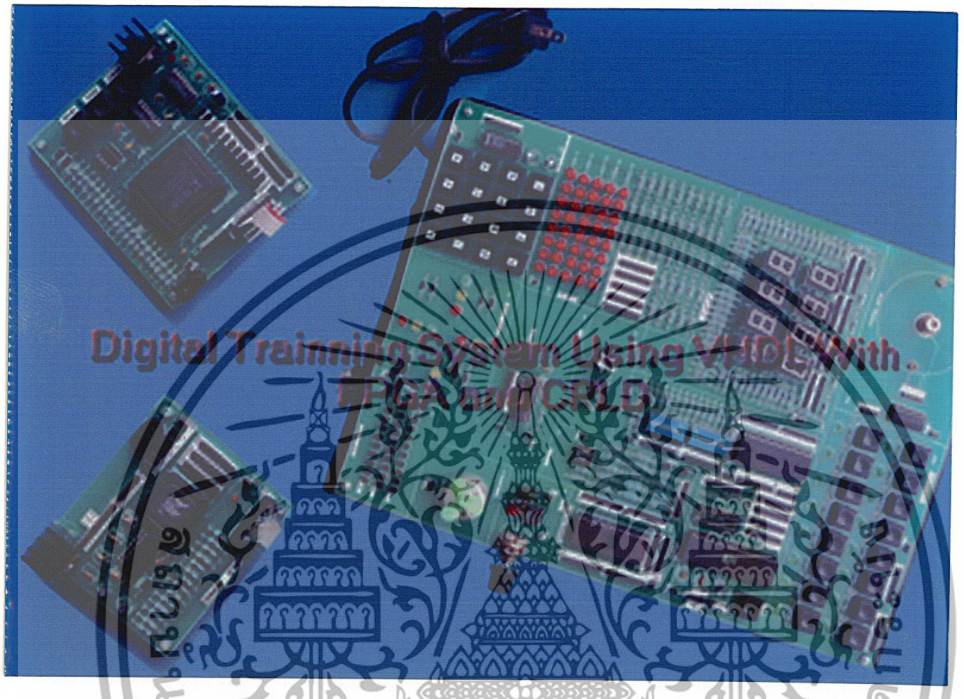
รูปที่ ง.37 โปรแกรมภาษาวีเอชดีแอลบรรยายการรับค่าเคาเตอร์ 0-9 จาก MCS 8051 มาแสดงผลออกโมดูลแสดงผลเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้ชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับเอฟพีจีเอและซีพีแอลดี



กล่าวนำ

การใช้งานชุดปฏิบัติการออกแบบระบบดิจิตอลโดยใช้ภาษาวีเอชดีแอลร่วมกับอุปกรณ์เอฟพีจีเอนั้น สามารถที่จะต่อใช้งานโดยผ่านพอร์ตขนานของเครื่องคอมพิวเตอร์เพื่อดาวน์โหลดโปรแกรมที่ต้องการออกแบบ ลงไปในตัวเอฟพีจีเอ และซีพีแอลดีได้ ซึ่งสามารถที่ต่อแผงวงจรเอฟพีจีเอ และแผงวงจรซีพีแอลดีเข้ากับแผงวงจรอินพุต/เอาต์พุตเพื่อที่จะแสดงผลได้ตามต้องการ ซึ่งผู้ใช้จะต้องระมัดระวังในการต่อแผงวงจรต่าง ๆ เข้าด้วยกัน โดยเฉพาะแหล่งจ่ายไฟฟ้ากระแสตรงที่ป้อนเข้าไปเลี้ยงแผงวงจรดังกล่าว ถ้าหากผู้ใช้ไม่ตรวจสอบให้ดีก่อนที่จะใช้งานอาจทำความเสียหายให้กับตัวอุปกรณ์ต่างๆ ได้

อุปกรณ์ทั้งหมด

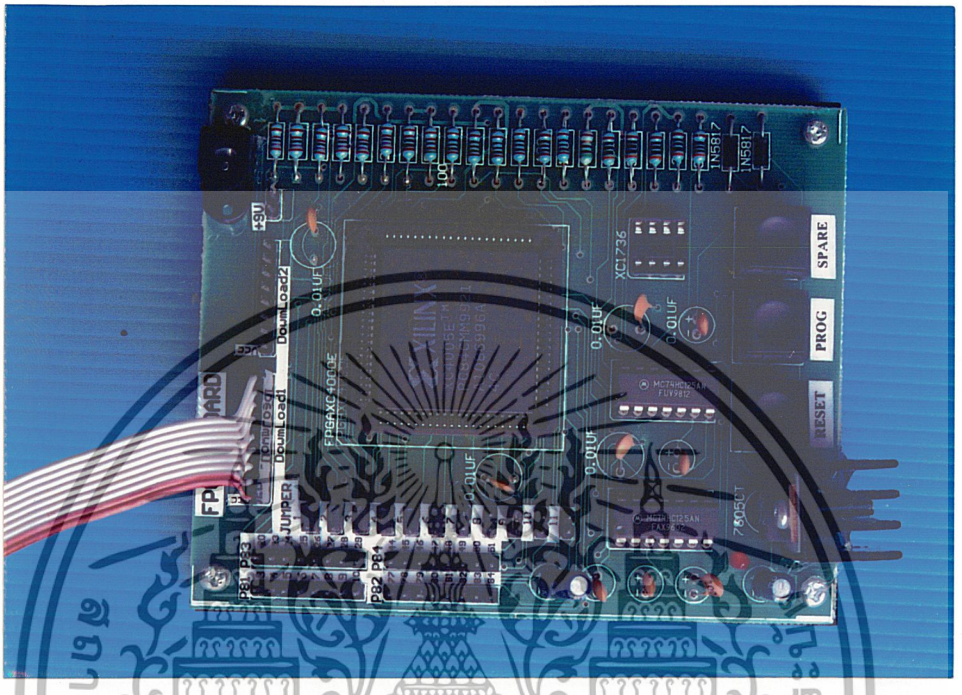
1) แผงวงจรเอฟพีจีเอ

2) แผงวงจรซีพีแอลดี

3) แผงวงจรอินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แผงวงจรเอฟพีจีเอ



รูปที่ จ.1 แผงวงจรเอฟพีจีเอ

P9_1 เป็น พอร์ตที่ใช้ดาวน์โหลดโปรแกรมจากเครื่องคอมพิวเตอร์

P+9V เป็นพอร์ตสำหรับจ่ายไฟ 9V ให้กับวงจร

Jum1 – Jum6 ต่อไว้เมื่อต้องการใช้วงจรดาวน์โหลดของพอร์ต ปลดออกเมื่อต้องการใช้วงจรดาวน์โหลดชนิดอื่นหรือในกรณีที่วงจรดาวน์โหลดของแผงวงจรมีปัญหา

Jum7 ต่อไว้หากต้องการใช้ปุ่ม Reset

P81 เป็นพอร์ต I/O โดยต่อที่ขา 3-10 ของ XC4000E

P82 เป็นพอร์ต I/O โดยต่อที่ขา 77-84 ของ XC4000E

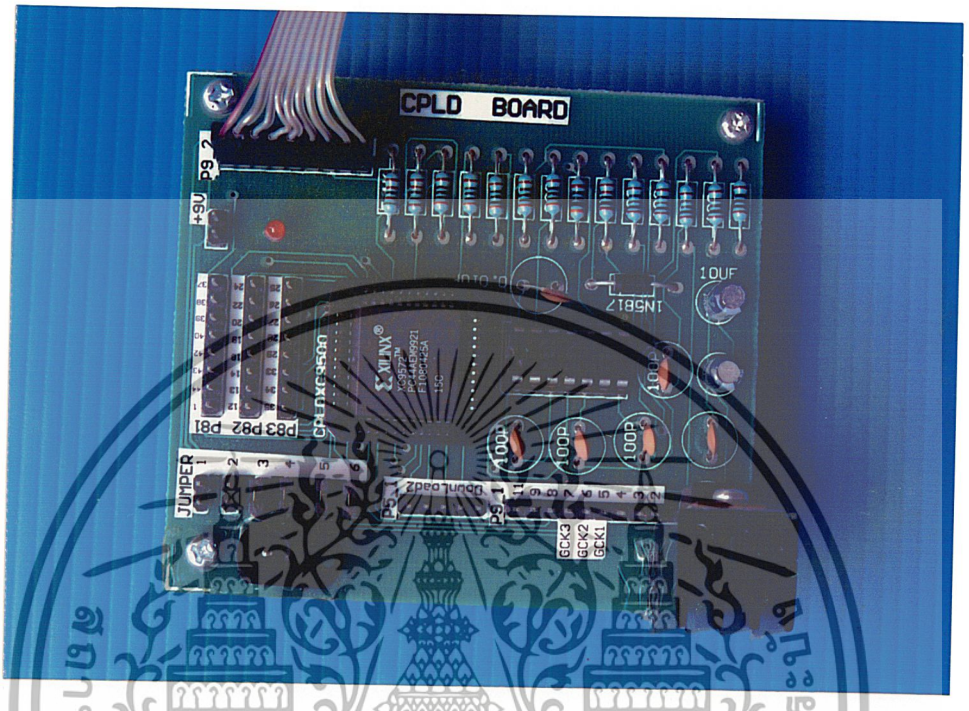
P83 เป็นพอร์ต I/O โดยต่อที่ขา 20 และ 23-29 ของ XC4000E

P84 เป็นพอร์ต I/O โดยต่อที่ขา 44-51 ของ XC4000E

P85 เป็นพอร์ตสำรองที่ทำไว้เพื่อในกรณีที่แผงวงจรทดลองนี้ชุดดาวน์โหลดมีปัญหาให้ใช้ชุดดาวน์โหลดอื่นสามารถต่อได้ที่นี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แผงวงจรซีพีแอลดี



รูปที่ จ.2 แผงวงจรซีพีแอลดี

P_1 เป็นพอร์ตสำหรับเชื่อมต่อกับคอมพิวเตอร์

P5_1 เป็นพอร์ตสำหรับเมื่อต้องการใช้ชุดดาวน์โหลดจากที่อื่น

P9_1 เป็นพอร์ต I/O ต่ออยู่กับขา 2-11 ของ XC9500

P81 เป็นพอร์ต I/O ต่ออยู่กับขา 31-40, 42-44 และขา 1 ของ XC9500

P82 เป็นพอร์ต I/O ต่ออยู่กับขา 12-14, 18-20 และ 22 กับ 24 ของ XC9500

P83 เป็นพอร์ต I/O ต่ออยู่กับขา 25-29 และ 33-35 ของ XC9500

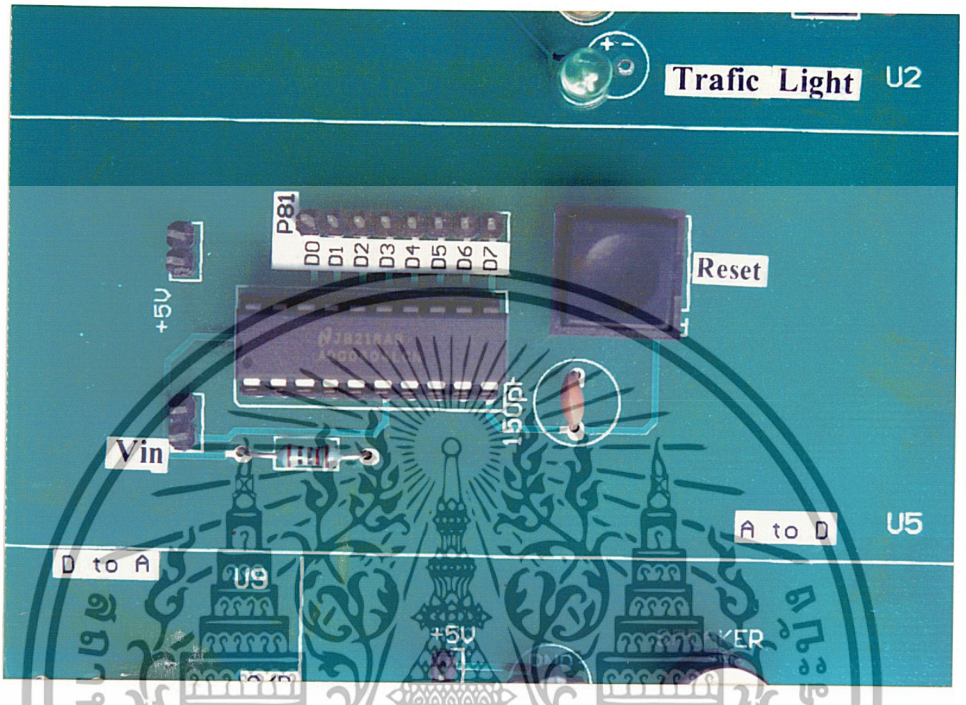
Jum1 ทำการต่อเมื่อไม่ต้องการให้ไฟเข้าขา 36

Jum2 ต่อไว้เมื่อมีการใช้ P5_1

Jum3-6 ต่อไว้เมื่อต้องการใช้ชุดดาวน์โหลดของแผงวงจร ปลดออกเมื่อต้องการใช้ชุดดาวน์โหลดจากที่อื่น (เมื่อใช้ P5_1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล



รูปที่ จ.3 โมดูล แปลงสัญญาณแอนะลอกเป็นดิจิตอล

P2_1 เป็น Vin

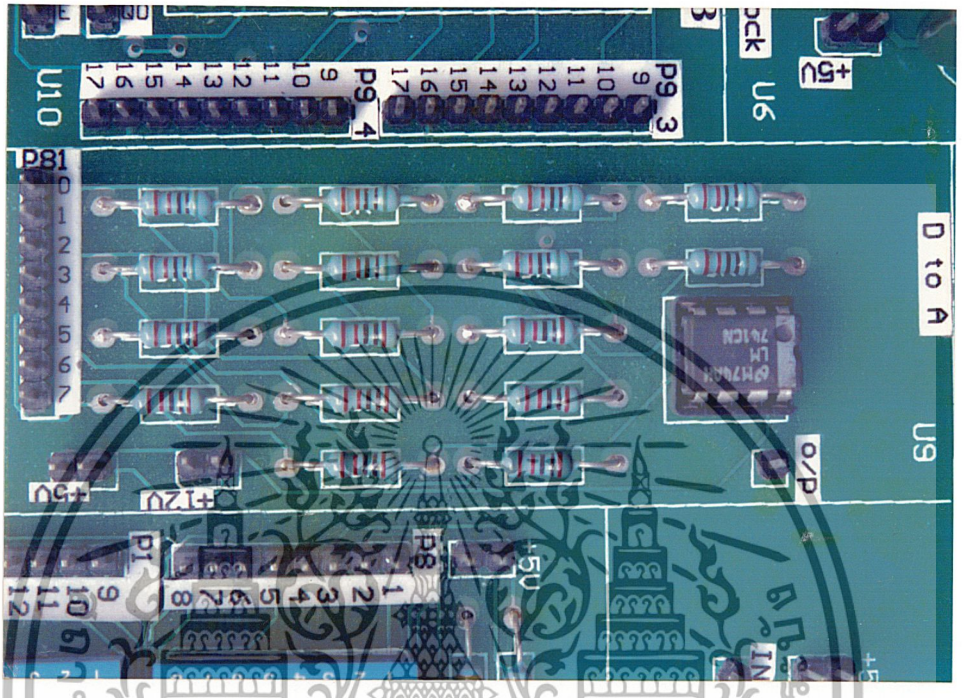
Sw1 เป็น Switch ใช้ในการรีเซ็ทค่าที่จะส่งออก

P81 เป็นพอร์ตเอาต์พุต 8 บิต

P+5V เป็นจุดต่อไฟกระแสตรง 5V ให้กับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โมดูลแปลงสัญญาณดิจิตอลเป็นแอนะลอก



รูปที่ จ.4 โมดูล แปลงสัญญาณดิจิตอลเป็นแอนะลอก

P+5V สำหรับต่อไฟ +5V ให้กับวงจร

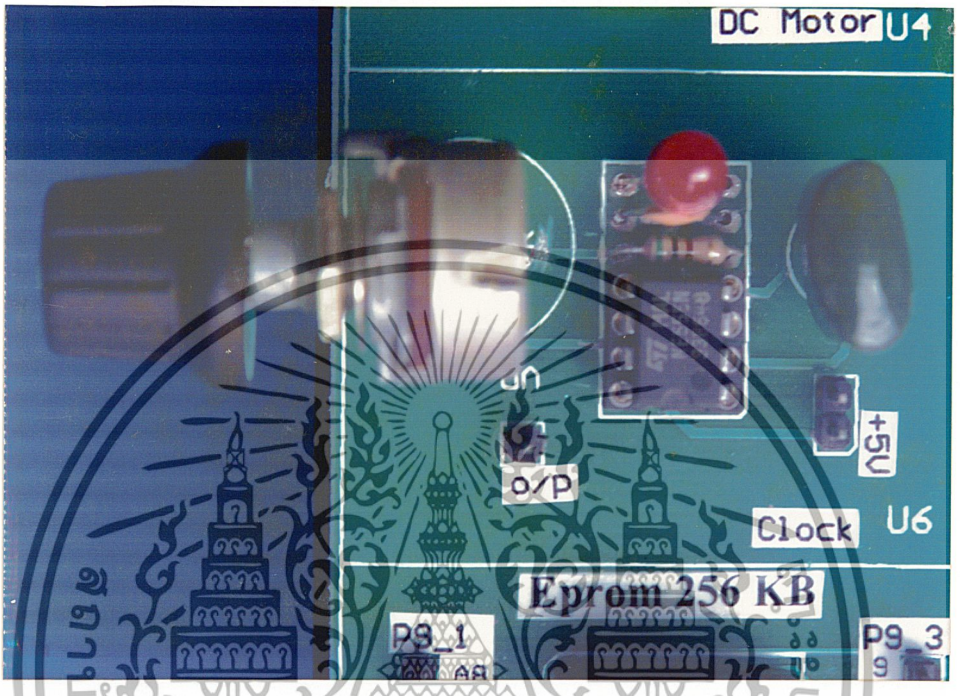
P+12V สำหรับต่อไฟ +12V ให้กับวงจร

P81 เป็นพอร์ต I/P 8 bit

P1_1 เป็นพอร์ต O/P ของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โมดูลวงจรกำเนิดสัญญาณนาฬิกา



รูปที่ จ.5 โมดูล วงจรกำเนิดสัญญาณนาฬิกา

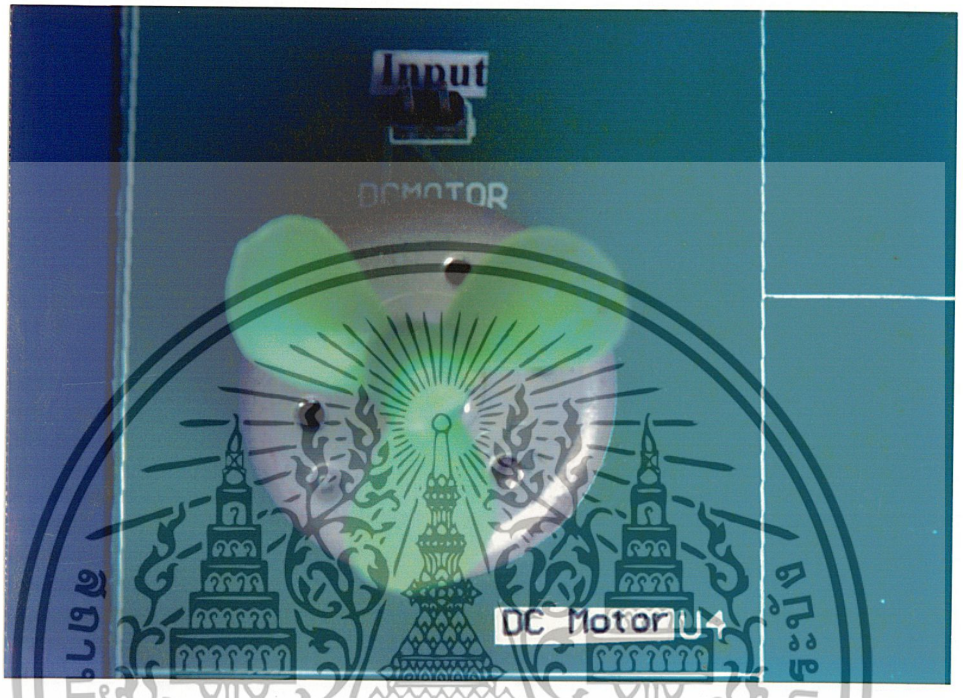
P+5V ใช้สำหรับต่อไฟ +5V ให้กับวงจร

VR_1 ปรับระดับของความถี่

P1_1 เป็นพอร์ตเอาต์พุตของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. โมดูลมอเตอร์ไฟฟ้ากระแสตรง

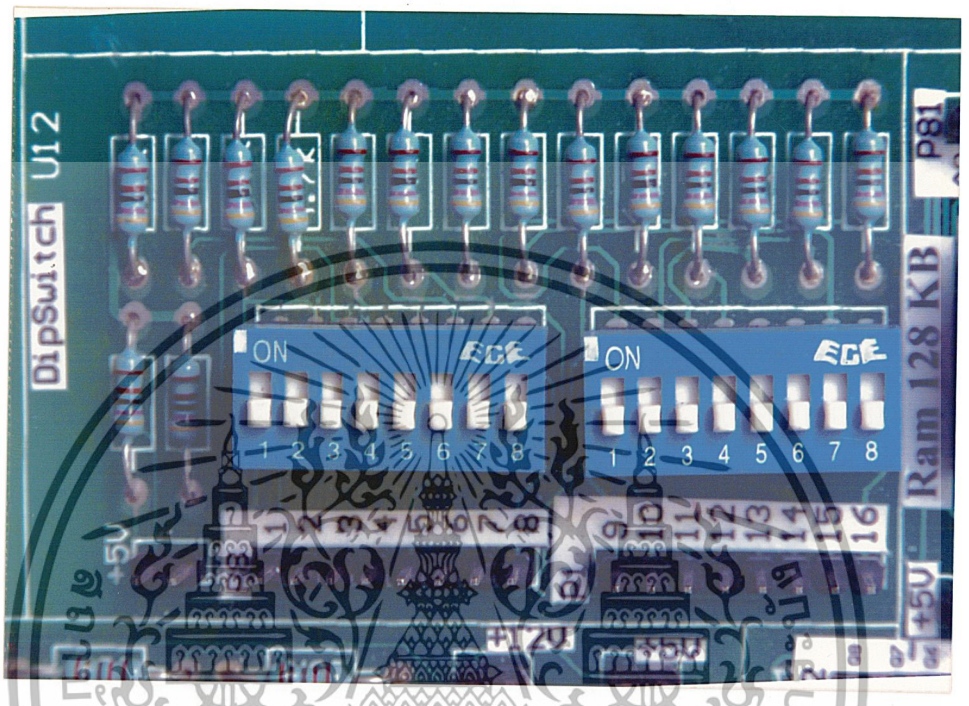


รูปที่ จ.6 โมดูล มอเตอร์ไฟฟ้ากระแสตรง

P2_1 เป็นพอร์ตอินพุตของวงจรใช้สำหรับควบคุมการหมุนของมอเตอร์ไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. โมดูลดิฟสวิตช์ 8 บิต



รูปที่ จ.7 โมดูล ดิฟสวิตช์ 8 บิต

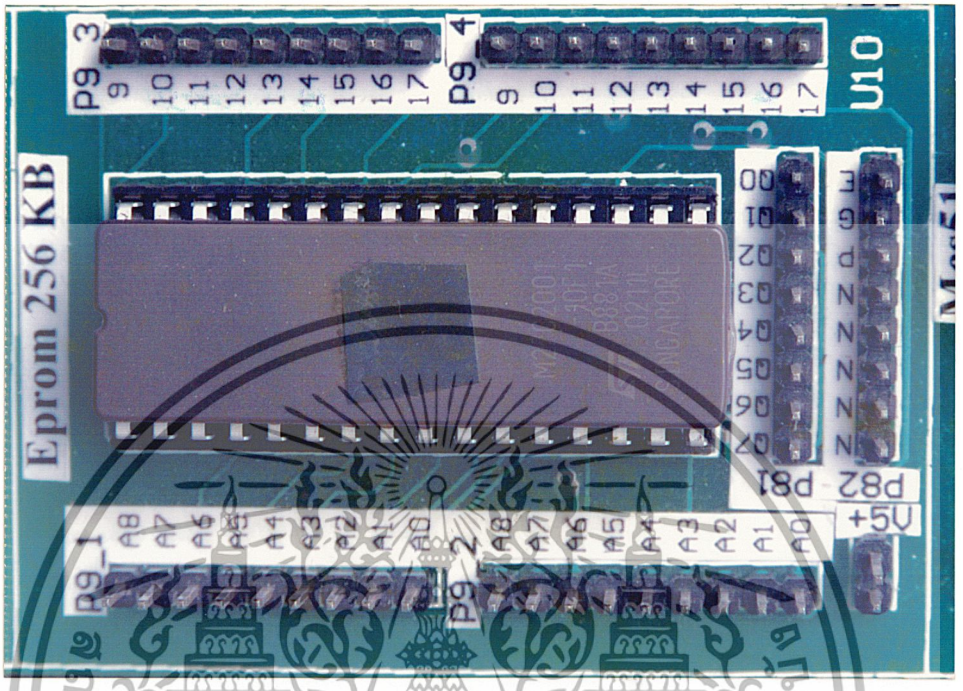
P+5V เป็นพอร์ตสำหรับจ่ายไฟ +5V ให้วงจร

P1 เป็นพอร์ตเอาต์พุต ของ ดิฟสวิตช์ CH1

P2 เป็นพอร์ตเอาต์พุต ของ ดิฟสวิตช์ CH2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. โมดูลอีพรอม 256 กิโลไบต์



รูปที่ จ.8 โมดูลอีพรอม 256 กิโลไบต์

P+5V เป็นพอร์ตสำหรับจ่ายไฟเลี้ยงวงจร

P9_1 และ P9_2 เป็นพอร์ตแอดเดรส A0 – A8

P9_3 และ P9_4 เป็นพอร์ตแอดเดรส A9 – A17

P8_1 และ P8_2 เป็นพอร์ตดาต้า Q0 – Q7

Pg เป็นพอร์ตที่ใช้สำหรับอ่านค่าจากอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ORG 0000H
table equ 0000h
DB 00H
DB 01H
DB 02H
DB 03H
DB 04H
DB 05H
DB 06H
DB 07H
DB 08H
DB 09H
DB 0AH
DB 0BH
DB 0CH
DB 0DH
DB 0EH
DB 0FH
DB 10H
DB 11H
DB 12H
DB 13H
DB 14H
DB 15H
DB 16H
DB 17H
DB 18H
DB 19H
DB 38H
DB 1AH
DB 1BH
DB 1CH
DB 1DH
DB 1EH
DB 1FH
DB 20H
DB 21H
DB 22H
DB 23H
DB 24H
DB 25H
DB 26H
DB 27H
DB 28H
DB 29H
DB 2AH
DB 2BH
DB 2CH
DB 2DH
DB 2EH
DB 2FH
DB 30H
DB 31H
DB 32H
DB 33H
DB 34H
DB 35H
DB 36H
DB 37H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    39H
DB    3AH
DB    3BH
DB    3CH
DB    3DH
DB    3EH
DB    3FH
DB    40H
DB    41H
DB    42H
DB    43H
DB    44H
DB    45H
DB    46H
DB    47H
DB    48H
DB    49H
DB    4AH
DB    4BH
DB    4CH
DB    4DH
DB    4EH
DB    4FH
DB    50H
DB    51H
DB    52H
DB    53H
DB    54H
DB    55H
DB    56H
DB    57H
DB    58H
DB    59H
DB    5AH
DB    5BH
DB    5CH
DB    5DH
DB    5EH
DB    5FH
DB    60H
DB    61H
DB    62H
DB    63H
DB    64H
DB    65H
DB    66H
DB    67H
DB    68H
DB    69H
DB    6AH
DB    6BH
DB    6CH
DB    6DH
DB    6EH
DB    6FH
DB    70H
DB    71H
DB    72H
DB    73H
DB    74H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB   75H
DB   76H
DB   77H
DB   78H
DB   79H
DB   7AH
DB   7BH
DB   7CH
DB   7DH
DB   7EH
DB   7FH
DB   80H
DB   81H
DB   82H
DB   83H
DB   84H
DB   85H
DB   86H
DB   87H
DB   88H
DB   89H
DB   8AH
DB   8BH
DB   8CH
DB   8DH
DB   8EH
DB   8FH
DB   90H
DB   91H
DB   92H
DB   93H
DB   94H
DB   95H
DB   96H
DB   97H
DB   98H
DB   99H
DB   9AH
DB   9BH
DB   9CH
DB   9DH
DB   9EH
DB   9FH
DB   0A0H
DB   0A1H
DB   0A2H
DB   0A3H
DB   0A4H
DB   0A5H
DB   0A6H
DB   0A7H
DB   0A8H
DB   0A9H
DB   0AAH
DB   0ABH
DB   0ACH
DB   0ADH
DB   0AEH
DB   0AFH
DB   0B0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    0B1H
DB    0B2H
DB    0B3H
DB    0B4H
DB    0B5H
DB    0B6H
DB    0B7H
DB    0B8H
DB    0B9H
DB    0BAH
DB    0BBH
DB    0BCH
DB    0BDH
DB    0BEH
DB    0BFH
DB    0C0H
DB    0C1H
DB    0C2H
DB    0C3H
DB    0C4H
DB    0C5H
DB    0C6H
DB    0C7H
DB    0C8H
DB    0C9H
DB    0CAH
DB    0CBH
DB    0CCH
DB    0CDH
DB    0CEH
DB    0CFH
DB    0D0H
DB    0D1H
DB    0D2H
DB    0D3H
DB    0D4H
DB    0D5H
DB    0D6H
DB    0D7H
DB    0D8H
DB    0D9H
DB    0DAH
DB    0DBH
DB    0DCH
DB    0DDH
DB    0DEH
DB    0DFH
DB    0E0H
DB    0E1H
DB    0E2H
DB    0E3H
DB    0E4H
DB    0E5H
DB    0E6H
DB    0E7H
DB    0E8H
DB    0E9H
DB    0EAH
DB    0EBH
DB    0ECH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    0EDH
DB    0EEH
DB    0EFH
DB    0F0H
DB    0F1H
DB    0F2H
DB    0F3H
DB    0F4H
DB    0F5H
DB    0F6H
DB    0F7H
DB    0F8H
DB    0F9H
DB    0FAH
DB    0FBH
DB    0FCH
DB    0FDH
DB    0FEH
DB    0FFH
DB    00000000B  ;0
DB    00000001B  ;1
DB    00000011B  ;2
DB    00000010B  ;3
DB    00000110B  ;4
DB    00000111B  ;5
DB    00000101B  ;6
DB    00000100B  ;7
DB    00001100B  ;8
DB    00001101B  ;9
DB    00001111B  ;10
DB    00001110B  ;11
DB    00001010B  ;12
DB    00001011B  ;13
DB    00001001B  ;14
DB    00001000B  ;15
DB    00011000B  ;16
DB    00011001B  ;17
DB    00011011B  ;18
DB    00011010B  ;19
DB    00011110B  ;20
DB    00011111B  ;21
DB    00011101B  ;22
DB    00011100B  ;23
DB    00010100B  ;24
DB    00010101B  ;25
DB    00010111B  ;26
DB    00010110B  ;27
DB    00010010B  ;28
DB    00010011B  ;29
DB    00010001B  ;30
DB    00010000B  ;31
DB    00110000B  ;32
DB    00110001B  ;33
DB    00110011B  ;34
DB    00110010B  ;35
DB    00110110B  ;36
DB    00110111B  ;37
DB    00110101B  ;38
DB    00110100B  ;39
DB    00111100B  ;40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    00111101B  ;41
DB    00111111B  ;42
DB    00111110B  ;43
DB    00111010B  ;44
DB    00111011B  ;45
DB    00111001B  ;46
DB    00111000B  ;47
DB    00101000B  ;48
DB    00101001B  ;49
DB    00101011B  ;50
DB    00101010B  ;51
DB    00101110B  ;52
DB    00101111B  ;53
DB    00101101B  ;54
DB    00101100B  ;55
DB    00100100B  ;56
DB    00100101B  ;57
DB    00100111B  ;58
DB    00100110B  ;59
DB    00100010B  ;60
DB    00100011B  ;61
DB    00100001B  ;62
DB    00100000B  ;63
DB    01100000B  ;64
DB    01100001B  ;65
DB    01100011B  ;66
DB    01100010B  ;67
DB    01100110B  ;68
DB    01100111B  ;69
DB    01100101B  ;70
DB    01100100B  ;71
DB    01101100B  ;72
DB    01101101B  ;73
DB    01101111B  ;74
DB    01101110B  ;75
DB    01101010B  ;76
DB    01101011B  ;77
DB    01101001B  ;78
DB    01101000B  ;79
DB    01111000B  ;80
DB    01111001B  ;81
DB    01111011B  ;82
DB    01111010B  ;83
DB    01111110B  ;84
DB    01111111B  ;85
DB    01111101B  ;86
DB    01111100B  ;87
DB    01110100B  ;88
DB    01110101B  ;89
DB    01110111B  ;90
DB    01110110B  ;91
DB    01110010B  ;92
DB    01110011B  ;93
DB    01110001B  ;94
DB    01110000B  ;95
DB    01010000B  ;96
DB    01010001B  ;97
DB    01010011B  ;98
DB    01010010B  ;99
DB    01010110B  ;100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    01010111B  ;101
DB    01010101B  ;102
DB    01010100B  ;103
DB    01011100B  ;104
DB    01011001B  ;105
DB    01011111B  ;106
DB    01011110B  ;107
DB    01011010B  ;108
DB    01011011B  ;109
DB    01011001B  ;110
DB    01011000B  ;111
DB    01001000B  ;112
DB    01001001B  ;113
DB    01001011B  ;114
DB    01001010B  ;115
DB    01001110B  ;116
DB    01001111B  ;117
DB    01001101B  ;118
DB    01001100B  ;119
DB    01000100B  ;120
DB    01000101B  ;121
DB    01000111B  ;122
DB    01000110B  ;123
DB    01000010B  ;124
DB    01000011B  ;125
DB    01000001B  ;126
DB    01000000B  ;127
DB    11000000B  ;128
DB    11000001B  ;129
DB    11000011B  ;130
DB    11000010B  ;131
DB    11000110B  ;132
DB    11000111B  ;133
DB    11000101B  ;134
DB    11000100B  ;135
DB    11001100B  ;136
DB    11001101B  ;137
DB    11001111B  ;138
DB    11001110B  ;139
DB    11001010B  ;140
DB    11001011B  ;141
DB    11001001B  ;142
DB    11001000B  ;143
DB    11011000B  ;144
DB    11011001B  ;145
DB    11011011B  ;146
DB    11011010B  ;147
DB    11011110B  ;148
DB    11011111B  ;149
DB    11011101B  ;150
DB    11011100B  ;151
DB    11010100B  ;152
DB    11010101B  ;153
DB    11010111B  ;154
DB    11010110B  ;155
DB    11010010B  ;156
DB    11010011B  ;157
DB    11010001B  ;158
DB    11010000B  ;159
DB    11110000B  ;160
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    11110001B  ;161
DB    11110011B  ;162
DB    11110010B  ;163
DB    11110110B  ;164
DB    11110111B  ;165
DB    11110101B  ;166
DB    11110100B  ;167
DB    11111100B  ;168
DB    11111101B  ;169
DB    11111111B  ;170
DB    11111110B  ;171
DB    11111010B  ;172
DB    11111011B  ;173
DB    11111001B  ;174
DB    11111000B  ;175
DB    11101000B  ;176
DB    11101001B  ;177
DB    11101011B  ;178
DB    11101010B  ;179
DB    11101110B  ;180
DB    11101111B  ;181
DB    11101101B  ;182
DB    11101100B  ;183
DB    11100100B  ;184
DB    11100101B  ;185
DB    11100111B  ;186
DB    11100110B  ;187
DB    11100010B  ;188
DB    11100011B  ;189
DB    11100001B  ;190
DB    11100000B  ;191
DB    10100000B  ;192
DB    10100001B  ;193
DB    10100011B  ;194
DB    10100010B  ;195
DB    10100110B  ;196
DB    10100111B  ;197
DB    10100101B  ;198
DB    10100100B  ;199
DB    10101100B  ;200
DB    10101101B  ;201
DB    10101111B  ;202
DB    10101110B  ;203
DB    10101010B  ;204
DB    10101011B  ;205
DB    10101001B  ;206
DB    10101000B  ;207
DB    10111000B  ;208
DB    10111001B  ;209
DB    10111011B  ;210
DB    10111010B  ;211
DB    10111110B  ;212
DB    10111111B  ;213
DB    10111101B  ;214
DB    10111100B  ;215
DB    10110100B  ;216
DB    10110101B  ;217
DB    10110111B  ;218
DB    10110110B  ;219
DB    10110010B  ;220
```

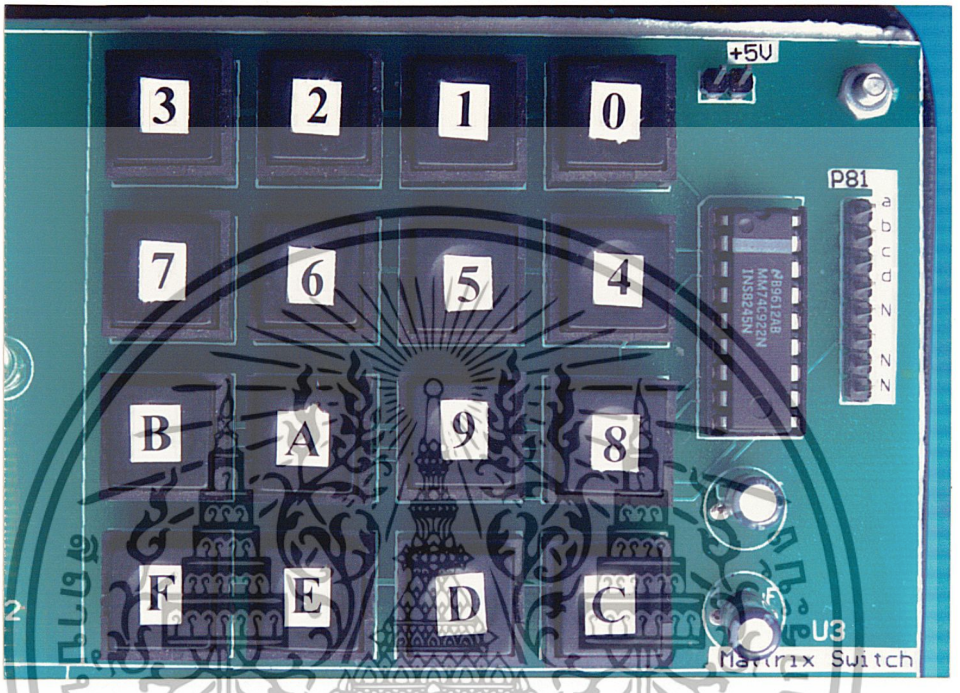
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB    10110011B  ;221
DB    10110001B  ;222
DB    10110000B  ;223
DB    10010000B  ;224
DB    10010001B  ;225
DB    10010011B  ;226
DB    10010010B  ;227
DB    10010110B  ;228
DB    10010111B  ;229
DB    10010101B  ;230
DB    10010100B  ;231
DB    10011100B  ;232
DB    10011101B  ;233
DB    10011111B  ;234
DB    10011110B  ;235
DB    10011010B  ;236
DB    10011011B  ;237
DB    10011001B  ;238
DB    10011000B  ;239
DB    10001000B  ;240
DB    10001001B  ;241
DB    10001011B  ;242
DB    10001010B  ;243
DB    10001110B  ;244
DB    10001111B  ;245
DB    10001101B  ;246
DB    10001100B  ;247
DB    10000100B  ;248
DB    10000101B  ;249
DB    10000111B  ;250
DB    10000110B  ;251
DB    10000010B  ;252
DB    10000011B  ;253
DB    10000001B  ;254
DB    10000000B  ;255
END
```

รูปที่ จ.9 โปรแกรมที่ใช้เขียนลงในอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. โมดูลคีย์บอร์ดเมตริกซ์ 4 x 4

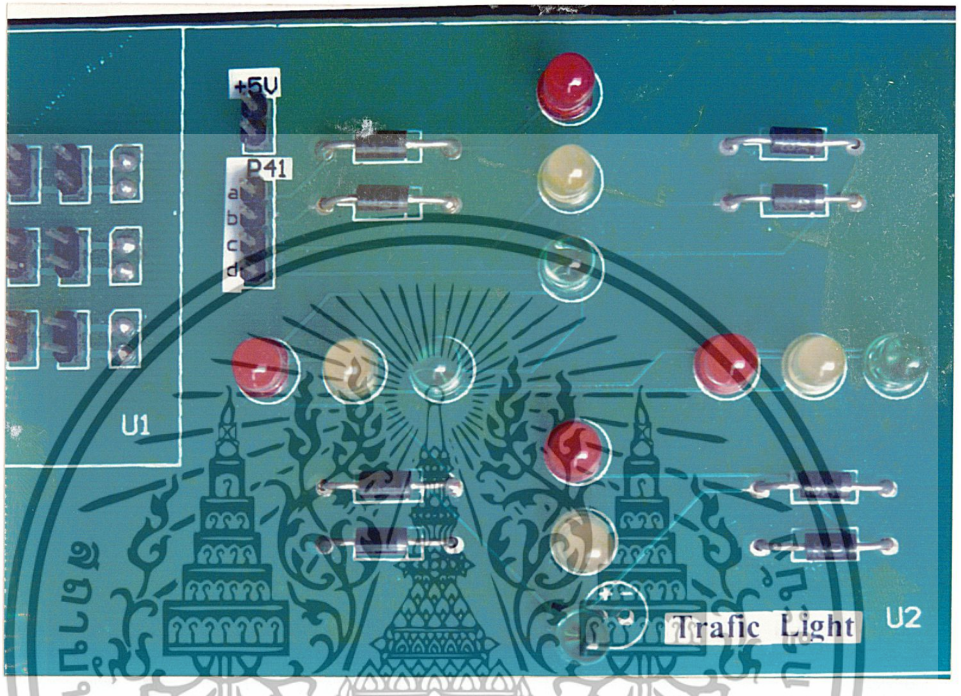


รูปที่ จ.10 โมดูลคีย์บอร์ดเมตริกซ์ 4 x 4

P+5V	เป็นพอร์ตสำหรับต่อไฟเลี้ยงให้กับวงจร
$SW_1 - SW_{16}$	เป็นสวิตช์สำหรับส่งค่า ตั้งแต่ 0 – F ออกพอร์ตเอาต์พุต
P81	เป็นพอร์ตเอาต์พุต 4 bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. โมดูลวงจรสี่แยกไฟแดง



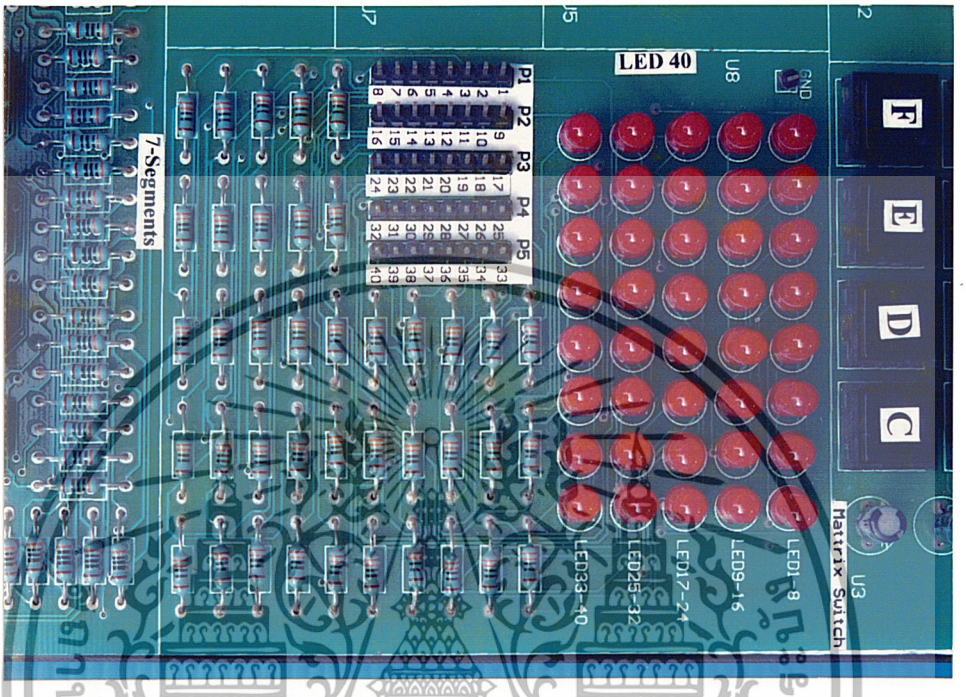
รูปที่ จ.11 โมดูล วงจรสี่แยกไฟแดง

P+5V เป็นพอร์ตสำหรับจ่ายไฟให้กับวงจร

P4.1 เป็นพอร์ตสำหรับป้อนอินพุตควบคุมสี่แยกไฟแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. โมดูลแอลอีดี 40 ดวง



รูปที่ จ.12 โมดูล แอลอีดี 40 ดวง

P81 เป็นพอร์ตควบคุม LED 1-8

P82 เป็นพอร์ตควบคุม LED 9-16

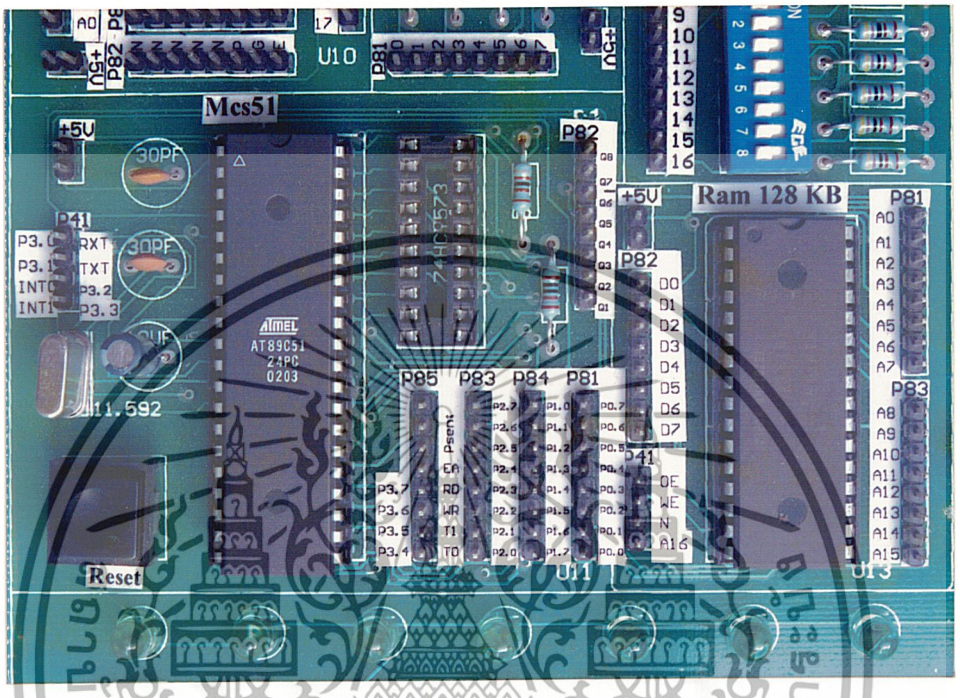
P83 เป็นพอร์ตควบคุม LED 17-24

P84 เป็นพอร์ตควบคุม LED 25-32

P85 เป็นพอร์ตควบคุม LED 33-40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. โมดูลMCS 8051



รูปที่ จ.13 โมดูล MCS 8051

P+5V เป็นพอร์ตสำหรับจ่ายไฟเลี้ยงวงจร

P81 เป็นพอร์ต I/O 8 bit ต่อกับ P0.0 – P0.7 ของ MCS 8051

P82 เป็นพอร์ต เป็นพอร์ตที่ต่ออยู่กับวงจรแล็ดเดอร์ของ 74 HCT579 ที่ขา Q1-Q8

P83 เป็นพอร์ต I/O 8 bit ต่อกับ P2.0 – P2.7 ของ MCS 8051

P84 เป็นพอร์ต I/O 8 bit ต่อกับ P1.0 – P1.7 ของ MCS 8051

P85 เป็นพอร์ต I/O และพอร์ตควบคุมดังนี้ที่ P85-4 เป็นพอร์ต P3.4-P3.7 ส่วน P85_5 และ P85_6 เป็นขา EA และ PSENT ตามลำดับ

P41 เป็นพอร์ต I/O 4 bit ต่อที่ขา P3.0 – P3.3 ของ MCS 8051

ภายในตัว IC ได้เขียนโปรแกรมนี้ลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. โมดูลแหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์, 9 โวลต์ และ12 โวลต์

รูปที่ จ.14 โมดูล แหล่งจ่ายไฟฟ้ากระและตรง 5 โวลต์, 9 โวลต์ และ12 โวลต์

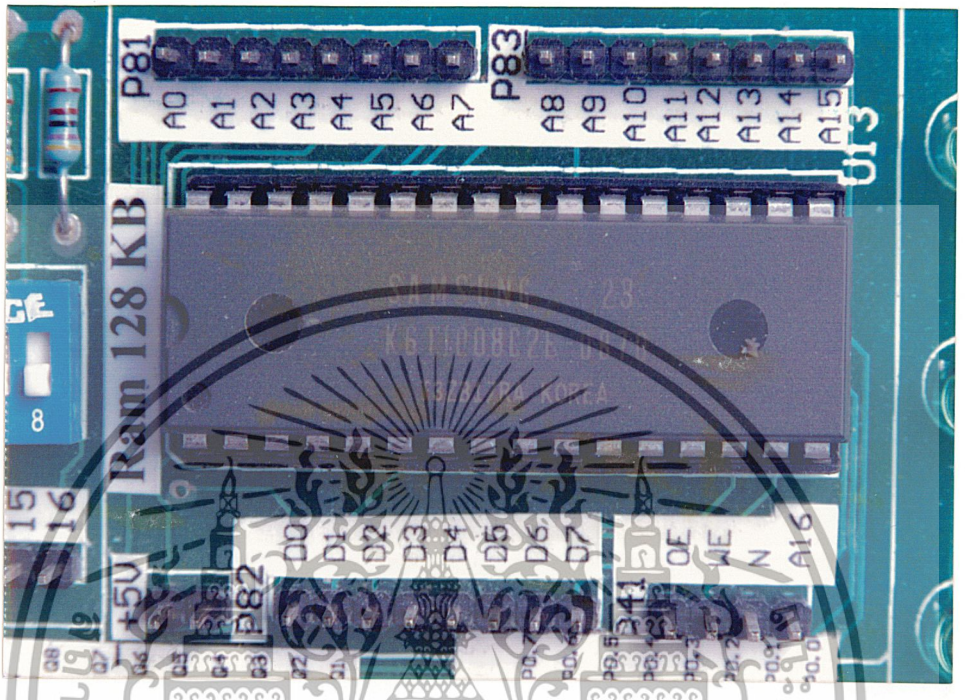
P+5V จำนวน 10 พอร์ต สำหรับจ่ายไฟ 5 V

P+9V จำนวน 10 พอร์ต สำหรับจ่ายไฟ 9 V

P+12V จำนวน 10 พอร์ต สำหรับจ่ายไฟ 12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. โมดูลแรม 128 กิโลไบต์



รูปที่ จ.15 โมดูล แรม 128 กิโลไบต์

P+5V เป็นพอร์ตจ่ายไฟเลี้ยงให้กับวงจร

P8_2 เป็นพอร์ต Data ตั้งแต่ D0-D7

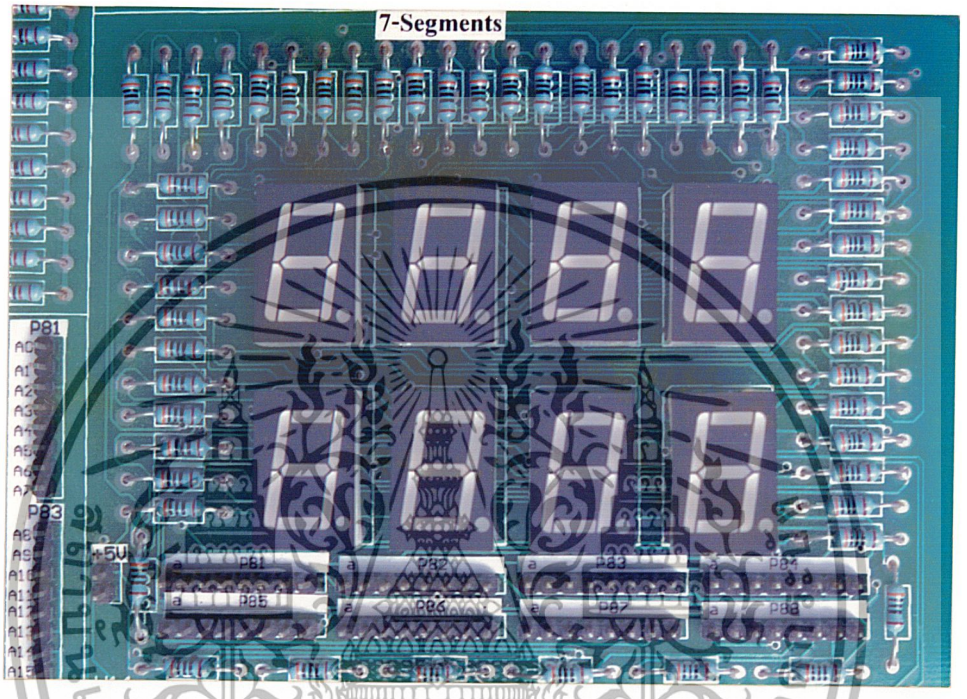
P8_1 เป็นพอร์ต Address ตั้งแต่ A0-A7

P8_3 เป็นพอร์ต Address ตั้งแต่ A8-A15

P4_1 เป็นพอร์ตควบคุม โดยที่ P41_3 และ P41_4 เป็นขา WE และ OE ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. โมดูลแสดงผลเจ็ดส่วน



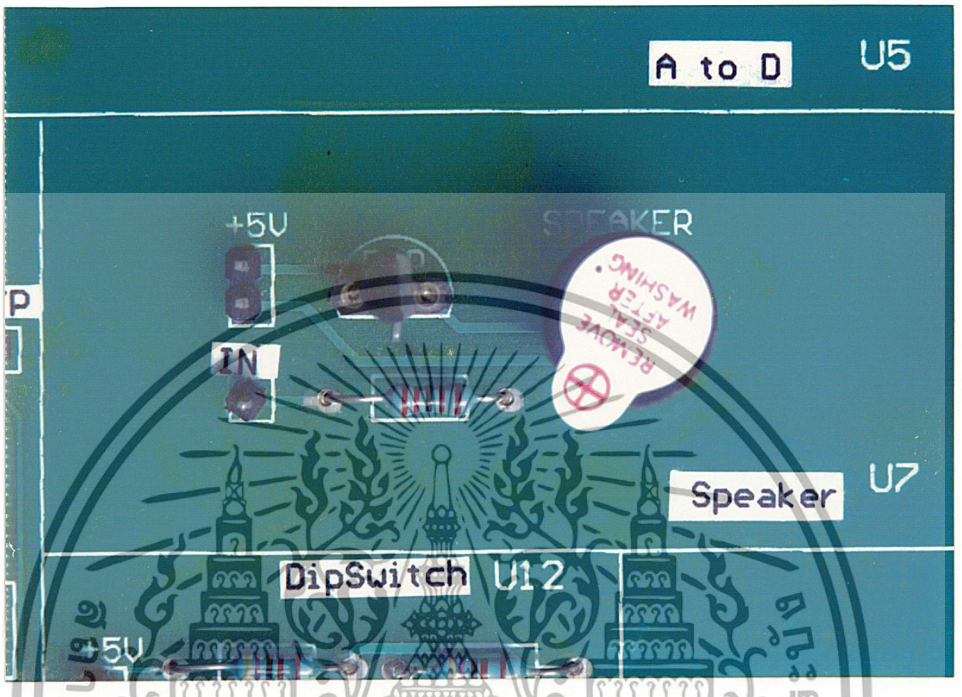
รูปที่ จ.16 โมดูล แสดงผลเจ็ดส่วน

P81-P88 เป็นพอร์ตควบคุม 7-Segment ตั้งแต่หลักที่ 1 ถึง หลักที่ 8

P+5V เป็นพอร์ตสำหรับจ่ายไฟ 5V ให้กับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. โมดูลลำโพง



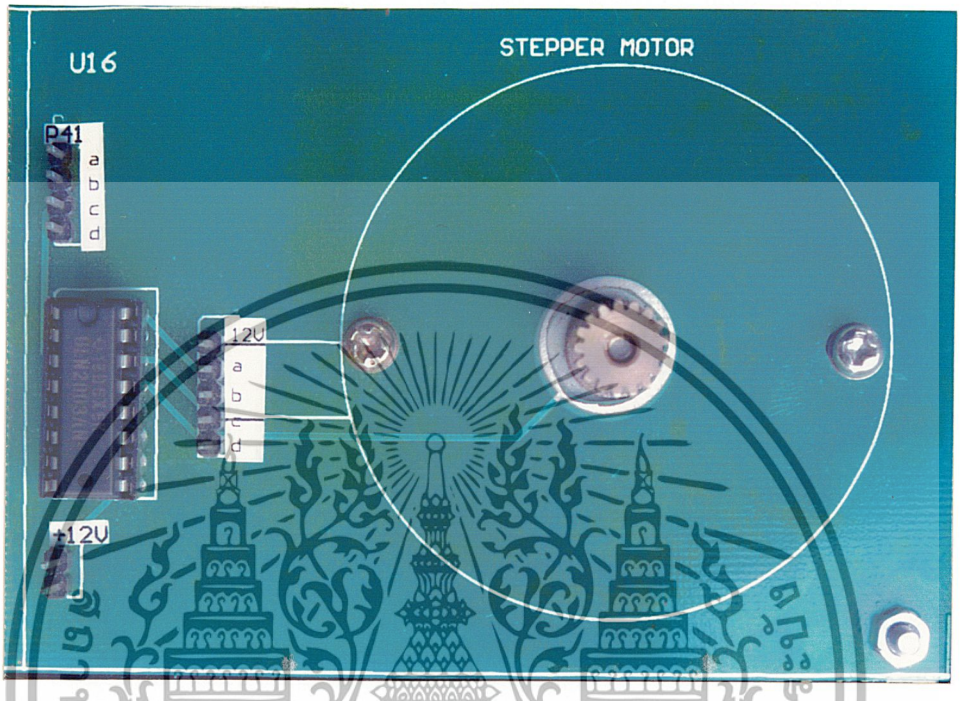
รูปที่ จ.17 โมดูล ลำโพง

P1_1 เป็น I/P ของวงจร

P+5V เป็นพอร์ตสำหรับจ่ายไฟเลี้ยงวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. โมดูลสเตปปิ้งมอเตอร์



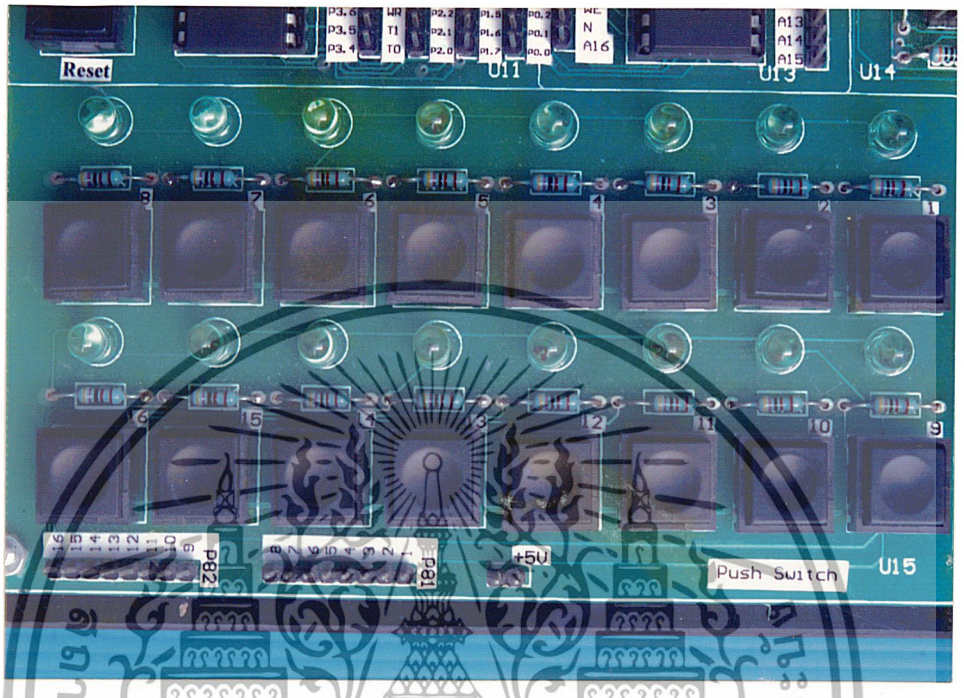
รูปที่ จ.18 โมดูล สเตปปิ้งมอเตอร์

P+12V เป็นพอร์ตสำหรับจ่ายไฟเลี้ยง +12V ให้กับวงจร

P41 เป็นอินพุตสำหรับควบคุม สเตปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. โมดูลสวิตซ์กดติดปล่อยดับ 16 ตัว



รูปที่ จ.19 โมดูลสวิตซ์กดติดปล่อยดับ 16 ตัว

P+5V	เป็นพอร์ตสำหรับจ่ายไฟ +5V ให้กับวงจร
$SW_1 - SW_{16}$	เป็นสวิตซ์สำหรับจ่ายไฟ 5V ให้ออกเอาต์พุตตามต้องการ
P81	เป็นเอาต์พุตที่ควบคุมโดย SW1- SW8
P82	เป็นเอาต์พุตที่ควบคุมโดย SW9 – SW16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ธวัชชัย เลื่อนฉวี,น.ต. และ อนุรักษ์ เถื่อนศิริ,พ.ท. **ดิจิตอลเทคนิค เล่ม 1,2.** พิมพ์ครั้งที่4 .กรุงเทพฯ: ภาพพิมพ์. 2532.

บัณฑิต บัวบูชา และ คณะ **ทฤษฎีและการออกแบบวงจรดิจิตอล เล่ม2** .กรุงเทพฯ : ฟิสิกส์เซ็นเตอร์.ม.ป.ป.

สมบูรณ์ เนียมกล่ำ.**เทคนิคการออกแบบวงจรลอจิกด้วยชิพ CPLD และ การใช้โปรแกรมออกแบบวงจรลอจิกFoundation F2.1i.** กรุงเทพฯ:ม.ป.ท.2544

สมยศ จุณณะปิยะ,รศ. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51.** พิมพ์ครั้งที่3. กรุงเทพฯ:สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2543

Douglas J Smith. **HDL Chip Desigh.** 2nd ed. USA : Doone. 2000

Ulrich Heinkel. **THE VHDL REFERENCE.** England : John Wiley & Sons. 2000

Xilinx Inc. **The Programmable Logic Data Book 1999.** USA : 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายคณาพจน์ ลิ่มทอง
วัน เดือน ปี เกิด	27 มิถุนายน พ.ศ. 2523
สถานที่เกิด	จังหวัดกระบี่
ภูมิลำเนา	111/2 หมู่ 2 ตำบลอ่าวลึกใต้ อำเภออ่าวลึก จังหวัดกระบี่ 81110
ที่อยู่ปัจจุบัน	111/26 แขวงลาดกระบัง เขตลาดกระบัง จังหวัดกรุงเทพฯ 10520
โทรศัพท์	0-9025-9500

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอ่าวลึก จังหวัดกระบี่
มัธยมศึกษาตอนต้น	โรงเรียนอำมาตย์พานิชนุกูล จังหวัดกระบี่
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคกระบี่ จังหวัดกระบี่
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตพระนครเหนือ จังหวัดกรุงเทพฯ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	เรียนรู้ลึกล้ำทำในสิ่งที่คิดว่าดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายกิตติพงษ์ บุญธนฤทธิ์
วัน เดือน ปี เกิด	22 ตุลาคม พ.ศ.2523
สถานที่เกิด	จังหวัดลพบุรี
ภูมิลำเนา	598 หมู่ 8 ตำบลลำนารายณ์
	อำเภอชัยบาดาล จังหวัดลพบุรี 15130
ที่อยู่ปัจจุบัน	187-189 หมู่ 3 แขวงลำประทิว เขตลาดกระบัง
	จังหวัดกรุงเทพฯ 10520
โทรศัพท์	0-1337-7720

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนนารายณ์วิทยา จังหวัดลพบุรี
มัธยมศึกษาตอนต้น	โรงเรียนชัยบาดาลวิทยา จังหวัดลพบุรี
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคลพบุรี จังหวัดลพบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคท่าหลวงซิเมนต์ไทยอนุสรณ์
	จังหวัดสระบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
	ลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	ทำวันนี้ให้ดีที่สุด เพื่ออนาคตที่สดใส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายธันว์ บัวเนียม
วัน เดือน ปี เกิด	15 ธันวาคม พ.ศ. 2523
สถานที่เกิด	จังหวัดยะลา
ภูมิลำเนา	60/1 ถ.ผังเมือง 1 ซ.1 ตำบลสะเตง อำเภอเมือง จังหวัดยะลา 95000
ที่อยู่ปัจจุบัน	260/17 ถ.อ่อนนุช แขวงลาดกระบัง เขตลาดกระบัง จังหวัดกรุงเทพฯ 10520
โทรศัพท์	0-9670-1176

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลยะลา จังหวัดยะลา
มัธยมศึกษาตอนต้น	โรงเรียนสตรียะลา จังหวัดยะลา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคยะลา จังหวัดยะลา
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตภาคใต้
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	ตั้งใจ จริงใจ และใส่ใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายไมตรี นาคประสิทธิ์
วัน เดือน ปี เกิด	7 สิงหาคม 2523
สถานที่เกิด	จังหวัดพิษณุโลก
ภูมิลำเนา	103 หมู่ 5 ตำบลคลองคะเชนทร์ อำเภอเมือง จังหวัดพิจิตร 66000
ที่อยู่ปัจจุบัน	111/26 แขวงลาดกระบัง เขตลาดกระบัง จังหวัดกรุงเทพฯ 10520
โทรศัพท์	0-1323-7471

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนชุมชนวัดคลองคะเชนทร์จังหวัดพิจิตร
มัธยมศึกษาตอนต้น	โรงเรียนพิจิตรพิทยาคม2 จังหวัดพิจิตร
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคพิจิตร จังหวัดพิจิตร
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตพระนครเหนือ จังหวัดกรุงเทพฯ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	ทำวันนี้ให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้