สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสร้างชุดทดลองเอฟพีจีเอ
FPGA DEVELOPMENT BOARD

นายนเรศ มาลัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 37041
วัน, เดือน, ปี 30 ธ.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างชุดทดลองเอฟพีจีเอ
FPGA DEVELOPMENT BOARD

โดย

นายนเรศ มาลัย

อาจารย์ที่ปรึกษา

รศ.สมศักดิ์ มิตะถา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสร้างชุดทดลองเอฟพีจีเอ

FPGA DEVELOPMENT BOARD

ผู้จัดทำ

1.นายนเรศ มาลัย รหัสประจำตัว 38014230

_____ อาจารย์ที่ปรึกษา

(รศ. สมศักดิ์ มิตะถา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างชุดทดลองเอฟพีจีเอ

นายนเรศ มาลัย รหัสประจำตัว 38014230
รศ. สมศักดิ์ มิตะถา อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ในปัจจุบันนี้ประเทศไทยได้มีการเริ่มมีการออกแบบวงจรรวมด้วยภาษาวีเอชดีแอล(VHDL: VHSIC (Very High Speed Intergreated Circuit) Hardware Description Language) มากขึ้นแต่ผู้ที่ออกแบบไม่ได้รับความสะดวกในการออกแบบเท่าที่ควรเพราะไม่มีบอร์ดทดลองให้มาทดสอบโปรแกรมที่ออกแบบมาและเป็นการยากต่อผู้ที่ทำการเริ่มศึกษาการออกแบบวงจรรวมด้วยภาษาวีเอชดีแอลเพราะว่าต้องทำวงจรขึ้นมาทดสอบเองซึ่งต้องเสียเวลามากดังนั้นโครงงานนี้จึงได้ทำการศึกษาและทำชุดทดลองเอฟพีจีเอขึ้นมาโดยมีจุดประสงค์เพื่อให้ผู้ที่ต้องการศึกษาการออกแบบวงจรโดยใช้ภาษาวีเอชดีแอล ได้ทดลองออกแบบและสามารถเรียนรู้ได้ด้วยตัวเอง

ในการออกแบบโครงงานนี้ตัวบอร์ดทดลองจะมีการแบ่งเป็นโมดูลต่างๆ เพื่อให้สะดวกแก่การใช้งานและให้ผู้ที่ใช้งานเข้าใจได้ง่ายโดยจะมีโมดูลต่างๆ คือ โมดูลเอฟพีจีเอ, โมดูลหน่วยความจำ, โมดูลแสดงผลแบบ LED 7 ส่วน(7-Segment Led Display), LCD โมดูล, โมดูลแสดงผลแบบ Dot Matrix ,โมดูล การติดต่อกับคีย์บอร์ด,โมดูลสวิทช์, โมดูลการแสดงผลแบบ LED, โมดูลการติดต่อของ RS-232 port, โมดูลการติดต่อของ Printer port, Steping Motor โมดูล และ DC-Motor โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA Development Board

Nareth Malai

Assoc.Prof. Somsak Mitatha Advisor

1999

Abstract

Nowadays, Thailand had begun design and development Integrated Circuit(IC) by using VHDL(VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) language. But designer and developer rough because they don't have experiment board for test his program. Design and development experiment board for only one purpose is consume more time and money. So the purposes of this project are design and develop FPGA Development Board for anyone who wants to study and design circuit by self.

In FPGA Development Board had partition into several modules for comfort and easy to learning. This board include FGPA, Memory, 7-Segment, LCD, Dot Matrix, Keyboard, Switch, LED, RS-232, Printer Port, Stepping Motor, and DC-Motor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ด้วยดีก็คือ อาจารย์สมศักดิ์ มิตะถา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก บุคคลที่สองที่จะขาดไปเสียมิได้คือนายปัญญยศ ไชยกาพ เป็นที่ปรึกษาและให้คำแนะนำต่างๆในการเขียนโปรแกรมทดสอบด้วยภาษาวีเอชดีแอลข้าพเจ้าขอขอบคุณมา ณ ที่นี้ด้วย

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอ กราบขอบพระคุณมา ณ ที่นี้

นเรศ มาลัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของงานโครงงาน	1
1.3 ขอบเขตของงานวิจัย	2
1.4 รายละเอียดของวิทยาโครงงาน	2
บทที่ 2 ภาษาวีเอชดีแอล	3
2.1 ประวัติความเป็นมาของภาษาวีเอชดีแอล	3
2.2 ส่วนประกอบต่างๆ ของภาษาวีเอชดีแอล	4
2.2.1 หน่วยการออกแบบเอนทิตี	5
2.2.1.1 port ชนิดต่างๆ	5
2.2.1.2 การตั้งชื่อ	5
2.2.1.3 VHDL Operators	6
2.2.1.3.1 Logical Operator	6
2.2.1.3.2 Relational Operator	7
2.2.1.3.3 Arithmetic Operator	7
2.2.1.3.4 Concatenation Operator	8
2.2.1.4 Data Objects	8
2.2.1.4.1 การประกาศค่า constant	8
2.2.1.4.2 การประกาศ variable	9
2.2.1.4.3 การประกาศ signal	9
2.2.1.5 ชนิดของข้อมูล	10
2.2.2 หน่วยการออกแบบสถาปัตยกรรม	12
2.2.3 หน่วยการออกแบบแพ็กเก็ต	14
2.2.3.1 Package Declaration	15
2.2.3.2 Package Body	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.2.4 หน่วยการออกแบบโครงแบบ 16

2.3 ภาษาวีเอชดีแอลเพื่อการสังเคราะห์ 16

2.3.1 ตัวอย่างรูปแบบการเขียนเกตพื้นฐาน 16

2.3.2 ตัวอย่างรูปแบบการเขียนฟลิปฟลอปพื้นฐาน 17

2.4 การออกแบบจากบนลงล่าง 18

บทที่ 3 เอฟพีจีเอ 21

3.1 Field Programmable 21

3.1.1 พีแอลดี (PLD: Programmable Logic Device) 21

3.1.2 พรอม (PROM: Programmable Read Only Memory) 22

3.1.3 พีเอแอล (PAL: Programmable Array Logic) 23

3.1.4 พีแอลเอ (PLA: Programmable Logic Array) 23

3.1.5 แอลซีเอ (LCA: Logic Cell Array) 24

3.1.6 อีพีแอลดี (EPLD: Erasable Programmable Logic Device) 25

3.2 Mask programmable 25

3.2.1 เกตอาเรย์ (Gate Array) 26

3.2.2 เซลล์มาตรฐาน (Standard Cell) 26

3.2.3 ฟูลคัสตัม (Full Custom) 27

3.3 เอฟพีจีเอ (FPGA: Field Programmable Gate Array) 27

3.3.1 สถาปัตยกรรมภายในของเอฟพีจีเอตระกูล XC4000 28

3.3.1.1 ซีแอลบี (CLB: Configuration Logic Block) 29

3.3.1.2 ไอโอบี (IOB: Input Output Block) 29

3.3.1.3 อินเตอร์คอนเน็ค (Interconnect) 30

3.3.2 คุณสมบัติโดยทั่วไปของเอฟพีจีเอตระกูล XC4000 31

3.4 การโปรแกรมเอฟพีจีเอตระกูล XC4000 32

3.4.1 โหมดการโปรแกรม 33

3.4.1.1 โหมดหลัก (Master modes) 33

3.4.1.2 เพอริเฟอรัลโหมด (Peripheral modes) 34

3.4.1.3 โหมดรองแบบอนุกรม (Slave Series modes) 35

3.5 การใช้ความสามารถของแรมในเอฟพีจีเอตระกูล XC4000 36

บทที่ 4 การใช้ซอฟแวร์เพื่อการโปรแกรมลงเอฟพีจีเอ 38

4.1 การอิมพลิเมนท์ (Implement) 39

4.2 การใช้ FPGA Editor เพื่อการตรวจสอบวงจรภายใน FPGA 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.3 การจำลองฐานเวลา (Simulation) ของวงจรหลังจากทำขั้นตอน Place&Route	45
4.4 การทดสอบวงจรกับฮาร์ดแวร์บนบอร์ดทดลองที่ใช้เอฟพีจีเอ	49
4.5 การคอนฟิกก์พรอม (Confige)	52
4.6 สรุป	54
บทที่ 5 การออกแบบฮาร์ดแวร์และซอฟแวร์	55
5.1 โมดูลต่างๆ ของบอร์ดทดลอง	56
5.1.1 เอฟพีจีเอโมดูล	56
5.1.2 เมโมรีโมดูล	58
5.1.3 7-Segment โมดูล	59
5.1.4 LCD โมดูล	60
5.1.5 Dot Matrix โมดูล	61
5.1.6 Keyboard โมดูล	62
5.1.7 Switch โมดูล	63
5.1.8 LED โมดูล	64
5.1.9 RS-232 Port โมดูล	65
5.1.10 Printer Port โมดูล	66
5.1.12 DC Motor โมดูล	69
5.2 วงจรรวมของบอร์ดทดลอง	70
บทที่ 6 การทดลองและผลการทดลอง	74
6.1 การทดลองที่ 1	74
6.2 การทดลองที่ 2	74
6.3 การทดลองที่ 3	75
6.4 การทดลองที่ 4	75
6.5 การทดลองที่ 5	75
6.6 การทดลองที่ 6	76
6.7 การทดลองที่ 7	76
6.8 การทดลองที่ 8	77
6.9 การทดลองที่ 9	77
6.10 การทดลองที่ 10	78
บทที่ 7สรุปผลและบทวิจารณ์	79
7.1 วิธีการดำเนินงาน	79
7.2 การออกแบบและทดสอบ	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
7.3 ปัญหาและอุปสรรคในการดำเนินงาน	80
7.4 คำวิจารณ์	80
7.5 สิ่งที่ต้องพัฒนาต่อไป	80
ภาคผนวก	81
วงจรรวม	82
วงจร PCB	84
ตารางขาที่ใช้งานของบอร์ดทั้งหมด	86
บรรณานุกรม	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี	5
2.2 ตัวอย่างของการประกาศค่า constant	8
2.3 ตัวอย่างของการประกาศ variable	9
2.4 ตัวอย่างของการประกาศ signal	9
2.5 Character ที่มีค่าเป็นไปได้	10
2.6 แสดงรูปแบบของมัลติเพลกซ์	11
2.7 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย	11
2.8 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก	12
2.9 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม	12
2.10 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชั่นบูลีน $output = (\overline{sel}.in0) + (sel.in1)$	13
2.11 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์	13
2.12 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง	14
2.13 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม	14
2.14 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต	15
2.15 โครงสร้างของบอดี้แพ็กเก็ต	15
2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	16
2.17 ขั้นตอนการออกแบบจากบนลงล่าง	19
3.1 ผังแสดงการแบ่งกลุ่มของวงจรรวม ASIC	21
3.2 แสดงวงจรพื้นฐานของอุปกรณ์พีแอลดีซึ่งอยู่ในรูปผลคูณร่วมบวก	22
3.3 แสดงลักษณะของพรอมเมื่อเปรียบเทียบเป็นวงจรในรูปผลคูณร่วมบวก	23
3.4 แสดงวงจรพื้นฐานภายในของพีแอลเอ	24
3.5 แสดงการจัดเรียงเป็นเมทริกซ์ของลอจิกเซลล์ของ LCA	24
3.6 ลักษณะของพีแอลดีในอนุกรมอีพีของบริษัทอัลเทอร่า	25
3.7 แสดงโครงสร้างของเซลล์มาตรฐาน	26
3.8 โครงสร้างภายในของ FPGA ของบริษัทไซริวซ์ (XILINX Inc.)	28
3.9 แสดงผังวงจรภายในของซีแอลบีของเอฟพีจีเอตระกูล XC4000	29
3.10 แสดงผังวงจรของไอโอบีภายในเอฟพีจีเอตระกูล XC4000	30
3.11 แสดงเส้นทางการเชื่อมต่อระหว่างไอโอบีกับซีแอลบีของเอฟพีจีเอตระกูล XC4000	30
3.12 แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดหลักแบบขนาน	34
3.13 แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดเพอริเฟอรัลแบบอะซิงโครนัส	35
3.14 แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดรองแบบอนุกรม	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.1 แสดงขั้นตอนการออกแบบวงจรโดยใช้ FPGA	38
4.2 หน้าต่างของ Design Manager	39
4.3 หน้าต่างของ New Project เมื่อสร้างโปรเจคใหม่	40
4.4 หน้าต่างของการกำหนด Version และเบอร์ของ FPGA ที่ใช้	40
4.5 หน้าต่าง Project ที่ทำการ New Version complete	41
4.6 หน้าต่าง Set Option	41
4.7 หน้าต่าง Set Option ที่ถูกกำหนดค่าต่างๆ	42
4.8 หน้าต่าง Flow Engine	42
4.9 หน้าต่าง Flow Engine ที่ทำครบทุกขั้นตอน	44
4.10 หน้าต่างสำหรับเลือกแสดงผลการทำงานในขั้นตอนต่างๆ	44
4.11 หน้าต่าง FPGA Editor และ Layer ของวงจรภายใน FPGA	45
4.12 หน้าต่าง Load Design ภายในโปรแกรม Modelsim	46
4.13 หน้าต่างสำหรับเลือกไฟล์ SDF ในการซิมมูเลชั่น	46
4.14 หน้าต่างตัวอย่างการเลือกไฟล์ SDF ในการซิมมูเลชั่น	47
4.15 หน้าต่างสำหรับเลือก Delay Selection เป็น MAX ในการซิมมูลเลชั่น	47
4.16 หน้าต่างที่แสดงรายละเอียดสำหรับการเลือกไฟล์ SDF ในการซิมมูลเลชั่น	48
4.17 หน้าต่างในการ Load Test bench พร้อมกับ SDF เรียบร้อย	48
4.18 สาย X-Checker พร้อมแถบสายต่อไปยังบอร์ดทดสอบ	49
4.19 ส่วนหัวของสาย X-Checker	50
4.20 การเชื่อมระหว่าง X-Checker กับบอร์ดทดสอบ	50
4.21 หน้าต่างโปรแกรม Hardware Debugger	51
4.22 หน้าต่างโปรแกรม Hardware Debugger ในการเปิด Design project	51
4.23 หน้าต่าง Communication Setup	52
4.24 หน้าต่างแสดงการ Download วงจรลงสู่ FPGA	52
4.25 หน้าต่างแสดงการ Download วงจรเสร็จสมบูรณ์	52
4.26 หน้าต่างโปรแกรม PROM File Formatter	53
4.27 หน้าต่าง PROM Properties	53
4.28 หน้าต่าง PROM Properties ในการเริ่มเขียนแบบ UP Direction	54
5.1 แสดงถึงโมดูลต่างๆ	55
5.2 วงจรของเอฟพีจีเอโมดูล	56
5.3 วงจรของเมโมรีโมดูล	58
5.4วงจรของ 7-Segment โมดูล	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
5.5 วงจรของ LCD โมดูล	60
5.6 วงจรของ Dot Matrix โมดูล	61
5.7 วงจรของ Keyboard โมดูล	62
5.8 วงจรของ Switch โมดูล	63
5.9 วงจรของ LED โมดูล	64
5.10 วงจรของ RS-232 Port โมดูล	65
5.11 วงจรของ Printer Port โมดูล	66
5.12 วงจรของ Steping Motor โมดูล	68
5.13 วงจรของ DC Motor โมดูล	69
5.14 บอร์ดทดลองจริง	70
5.15 การจัดวางโมดูลต่างๆ	70
5.16 วงจร Schematic รวมของบอร์ดทดลอง	71
5.17 ลาย PCBด้านบนของบอร์ดทดลอง	72
5.18 ลาย PCB ด้านล่างของบอร์ดทดลอง	72
5.19 รูของขาอุปกรณ์ทั้งหมดที่ใช้ในบอร์ดทดลอง	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การทำโอเปอเรชั่นทางคณิตศาสตร์	7
3.1 แสดงรายละเอียดของอุปกรณ์ภายในเอฟพีจีเอตระกูล XC4000	32
3.2 แสดงรูปแบบของโหมดต่างๆ ในการโปรแกรมเอฟพีจีเอตระกูล XC4000	33
3.3 จำนวนของแรมภายในเอฟพีจีเอตระกูล XC4000	36
3.4 รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010	36
5.1 แสดงถึงขาของเอฟพีจีเอที่ใช้งาน	57
5.2 แสดงขาของ SRAM ที่ต่อเข้ากับขาต่างๆ ของเอฟพีจีเอ	58
5.3 แสดงขาของ 7-Segment ที่ต่อใช้งานกับเอฟพีจีเอ	59
5.4 แสดงขาของ LEDที่ต่อใช้งานกับเอฟพีจีเอ	60
5.5แสดงขาของ Dot Matrix ที่ต่อใช้งานกับเอฟพีจีเอ	61
5.6 แสดงขาของ Keyboard ที่ต่อใช้งานกับเอฟพีจีเอ	62
5.7 แสดงขาของ Switchที่ต่อใช้งานกับเอฟพีจีเอ	63
5.8 แสดงขาของ LED ที่ต่อใช้งานกับเอฟพีจีเอ	64
5.9 แสดงขาของ RS-232 ทั้ง 2ตัวที่ต่อใช้งานกับเอฟพีจีเอ	65
5.10 แสดงขาของ Printer Port ทั้ง 2 ตัวที่ต่อใช้งานกับเอฟพีจีเอ	67
5.11 แสดงขาของ Steping Motor ที่ต่อใช้งานกับเอฟพีจีเอ	68
5.12 แสดงขาของ DC Motor โมดูลที่ต่อใช้งานกับเอฟพีจีเอ	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันถึงแม้ว่าอุตสาหกรรมอิเลกทรอนิกส์และเครื่องใช้ไฟฟ้าจะเป็นอุตสาหกรรมใหญ่ของประเทศ ที่มีการนำเข้า ส่งออกและการบริโภคภายในประเทศสูงก็ตาม แต่ในเชิงเทคโนโลยีแล้ว ประเทศไทยยังไม่สามารถเข้าไปมีส่วนร่วมในส่วนของการออกแบบและการผลิตชิ้นส่วนในระดับสูงที่เป็นหัวใจของผลิตภัณฑ์ได้ ทำให้มูลค่าเพิ่มที่เกิดจากอุตสาหกรรมไมโครอิเล็กทรอนิกส์ในประเทศอยู่ในระดับต่ำและต้องพึ่งพาเทคโนโลยีจากต่างประเทศเป็นหลัก

ดังนั้นเพื่อเป็นการแก้ปัญหาดังกล่าวเราจึงต้องอาศัยการออกแบวงจรรวม(IC Design) ซึ่งมีจุดเด่นคือใช้คนเป็นหลัก, ไม่ต้องลงทุนสูงมาก แต่เนื่องจากยังมีบุคลเป็นจำนวนมากที่ไม่สามารถออกแบบหรือสร้างวงจรเพื่อนำมาใช้กับผลิตภัณฑ์ของตนเอง สาเหตุนี้เนื่องมาจากขาดเครื่องมือและความเข้าใจในการออกแบบนั่นเอง หรือเกิดจากความคิดที่ว่าการออกแบบระบบฮาร์ดแวร์ดิจิตอลเป็นเรื่องที่ยากและไม่สามารถที่จะออกแบบได้ ดังนั้นจึงเป็นที่มาของโครงงานนี้ ซึ่งได้ออกแบบบอร์ดทดลองที่ใช้เอฟพีจีเอ (FPGA: Filed Programmable Gate Array) เพื่อให้ผู้ที่ต้องการศึกษาการออกแบบวงจรโดยใช้ภาษาวีเอชดีแอล (VHDL: VHSIC (Very High Speed Intergreated Circuit) Hardware Description Language) ได้ทดลองออกแบบและสามารถเรียนรู้ได้ด้วยตัวเอง

1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงงาน

ในโครงงานเรื่องการออกแบบบอร์ดทดลองเอฟพีจีเอนี้ได้ทำการศึกษาการออกแบบบอร์ดทดลองและสร้างต้นแบบชุดทดสอบ/ทดลองการใช้งานเอฟพีจีเอ เพื่อใช้ในการเรียนการสอนของภาควิชาต่อไปในอนาคต โดยชุดทดลองจะประกอบไปด้วยส่วนอินพุต/เอ้าท์พุตต่างๆ เพื่อใช้ในการทดสอบ FPGA รวมทั้งเป็นชุดทดลองที่ให้นักศึกษาสามารถทดลองเรียนรู้ได้ด้วยตัวเอง โดยจะมีอุปกรณ์อินพุต/เอ้าท์พุตเช่น LED 8 bit, Switch 8 bit, LCD 4 x 20, Keyboard 4 x 4, Stepping Motor, DC Motor, Speed Sensor &DC Motor, 7-Segment 4 หลัก เป็นต้น

โดยการทำโครงงานนี้ได้กำหนดจุดประสงค์ไว้ดังนี้

- เพื่อศึกษารูปแบบการเขียนภาษาวีเอชดีแอลในการออกแบบระบบฮาร์ดแวร์ดิจิตอล ซึ่งเริ่มตั้งแต่การออกแบบแก้ไขตรวจสอบ จำลองการทำงาน จนถึงขั้นผลิตวงจรหรือสังเคราะห์วงจร (Synthesis)
- เพื่อศึกษาลักษณะการทำงาน,การควบคุมและฟังก์ชั่นการใช้งานต่างๆของเอฟพีจีเอ
- เพื่อศึกษาการใช้งานของโปรแกรมต่างๆที่ใช้ในการออกแบบฮาร์ดแวร์ดิจิตอล เช่น V-system, Examplar Logic Design และ XILINX Foundation 2.1i
- สามารถสร้างบอร์ดทดลองมาใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

โครงงานนี้เริ่มจากการศึกษาการออกแบบวงจรสำหรับบอร์ดทดลองเอฟพีจีเอและได้สร้างบอร์ดทดลองให้สามารถใช้งานได้จริงโดยที่บอร์ดทดลองสามารถรองรับการออกแบบฮาร์ดแวร์ดิจิตอลด้วยภาษาวีเฮชดีแอลได้

1.4 รายละเอียดของโครงงาน

ในโครงงานนี้ ได้แบ่งเนื้อหาออกเป็นบทได้ทั้งหมด 8 บทโดย

บทที่ 1 จะเป็นการกล่าวนำถึง แนวความคิดและวัตถุประสงค์ในการทำโครงงาน และได้กล่าวถึงเนื้อหาโดยย่อของแต่ละบท

บทที่ 2 กล่าวถึงรูปแบบการเขียนภาษาวีเอชดีแอล และวิธีการออกแบบจากบนลงล่าง

บทที่ 3 กล่าวถึงรายละเอียดของอุปกรณ์เอฟพีจีเอที่นำมาใช้และการใช้งาน

บทที่ 4 กล่าวถึงการใช้ software เพื่อการโปรแกรมลง FPGA

บทที่ 5 กล่าวถึงการออกแบบฮาร์ดแวร์หรือตัวบอร์ดทดลอง

บทที่ 6 กล่าวถึงการทดลองและผลการทดลอง

บทที่ 7 สรุปและวิจารณ์

ภาคผนวก วงจรรวมของบอร์ดทดลอง,วงจรPCBและตารางขาการใช้งานของอุปกรณ์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ภาษาวีเอชดีแอล

วีเอชดีแอล ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (high level language) ที่ใช้ในการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (hierarchy) ได้ และสามารถที่จะเขียนได้หลายรูปแบบซึ่งจะกล่าวต่อไป จึงทำให้ภาษาวีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาสามารถเปิดโอกาสให้วิศวกรได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยที่ยังไม่ต้องไปคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้นวีเอชดีแอลยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งาน

2.1 ประวัติความเป็นมาของภาษาวีเอชดีแอล

วิวัฒนาการของภาษาวีเอชดีแอล นั้นเริ่มต้นประมาณปี ค.ศ. 1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ ดีโอดี (DoD: Department of Defense) มองเห็นว่าอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร เป็นอุปกรณ์ที่ได้รับการพัฒนามาเมื่อประมาณ 20 ปีก่อน เพราะเทคโนโลยีในขณะนั้นทำให้การพัฒนาอุปกรณ์อิเล็กทรอนิกส์เป็นไปอย่างล่าช้า ซึ่งเป็นสภาพที่ไม่อาจยอมรับได้ในปัจจุบัน เพราะเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์ ได้รับการพัฒนาไปอย่างรวดเร็ว ดังที่จะเห็นได้ว่ามีวงจรดิจิตอลอิเล็กทรอนิกส์หลายวงจร ที่แต่เดิมถูกสร้างขึ้นมาจากชิ้น ถูกนำประกอบกันอยู่บนแผงวงจรไฟฟ้า ที่มีขนาดใหญ่ แต่ในปัจจุบันสามารถที่จะใช้เทคโนโลยีการออกแบบและผลิตวงจรรวมขนาดใหญ่มาก (VLSI: Very Large Scale Integration) รวมอุปกรณ์ต่างๆ เหล่านั้นให้อยู่บนชิ้นอุปกรณ์สารกึ่งตัวนำ ที่มีขนาดประมาณ 1-2 ตร.ซม. ได้ ซึ่งเป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้น (ความเร็วในการทำงานของวงจร) ตลอดจนความน่าเชื่อถือในการทำงาน และความคงทนต่อสภาพแวดล้อมสูง ขณะเดียวกันนั้นในวงการทหารได้มีการนำระบบคอมพิวเตอร์และอิเล็กทรอนิกส์ มาใช้ในระบบอาวุธอย่างแพร่หลาย ดังนั้นอุปกรณ์ที่มีใช้อยู่จึงไม่เหมาะสมกับเทคโนโลยีด้านอาวุธของประเทศคู่แข่งขัน การที่จะเปลี่ยนอุปกรณ์ใหม่เป็นสิ่งที่ต้องใช้งบประมาณมาก และก็จะประสบกับปัญหาเช่นเดิมคือ อุปกรณ์ใหม่ได้รับการพัฒนามานานแล้วเช่นกัน เพราะในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกรและเวลาสำหรับดำเนินการมาก ฉะนั้นทาง ดีโอดีจึงตั้งโครงการขึ้นมาเพื่อศึกษา วิธีการที่จะช่วยพัฒนาวงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น และโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ วีเอชเอสไอซี (VHSIC) ในระยะแรกนั้นโครงการเป็นความลับทางด้านความมั่งคงของประเทศ และอยู่ในความดูแลควบคุมของ United States International Traffic and Arms Regulations หรือ ไอทีเออาร์ (ITAR) ในปี ค.ศ. 1983 ตามคำแนะนำของคณะทำงาน ("Woods Hole"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

workshop) ทาง ดีโอดีได้ออกความต้องการมาตรฐานของภาษาที่ใช้สำหรับบรรยายพฤติกรรมของวงจรหรือฮาร์ดแวร์ของระบบสำหรับโครงการวีเอชเอสไอซี ซึ่งมีสาระสำคัญพอสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถจะเข้าใจได้ทั้งคนและเครื่องโดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรมการออกแบบฮาร์ดแวร์เรียกว่า "Hardware Description Language" หรือ เอชดีแอล (HDL) เริ่มต้นโครงการดีโอดีได้มอบหมายให้บริษัทไอบีเอ็มและบริษัทเท็กซัสอินสตูลเมนท์และบริษัทอินเตอร์เมทริกซ์เป็นผู้ศึกษาและพัฒนา การดำเนินการได้กระทำไปอย่างต่อเนื่อง และได้ผลเป็นที่น่าพอใจ จนกระทั่งปี ค.ศ. 1985 ทางไอทีเออาร์ได้ยกเลิกข้อจำกัดในการถ่ายทอดเทคโนโลยีทางทหาร ออกจากโครงการนี้ ดังนั้นภาษาวีเอชดีแอล จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป จนกระทั่งทางไออีอีอี (IEEE) จึงได้รับภาษานี้เข้ามาศึกษาและประมาณปี ค.ศ. 1987 ได้ยอมรับกำหนดมาตรฐานของภาษา โดยให้ชื่อว่า IEEE 1076-1987 และมีชื่อเรียกว่าวีเอชดีแอลมาตรฐานนี้ก็ได้รับการปรับปรุงจนปัจจุบัน ได้ชื่อว่า IEEE 1076-1993 หรือ วีเอชดีแอล 1993 การที่ทางดีโอดีในขณะนั้น เป็นลูกค้ารายใหญ่ของอุตสาหกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ จึงมีผู้รับโครงการต่างๆ จากดีโอดี ไปดำเนินการค้นวิจัยและพัฒนามาก เพื่อที่จะให้เป็นมาตรฐานเดียวกันหมด ทางดีโอดี จึงกำหนดว่า ในการส่งโครงการนั้นจะต้องเขียนอยู่ในรูปของภาษาวีเอชดีแอลเท่านั้น ซึ่งทำให้เกิดข้อดีต่อดีโอดีเองที่เป็นมาตรฐานเดียวกัน สามารถนำไปจำลองกับเครื่องคอมพิวเตอร์ได้หลายๆ ระบบ

2.2 ส่วนประกอบต่างๆ ของภาษาวีเอชดีแอล

ในการเขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเอชดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

1. หน่วยการออกแบบเอนทิตี (Entity Design Unit)
2. หน่วยการออกแบบสถาปัตยกรรม (Architecture Design Unit)
3. หน่วยการออกแบบแพ็กเก็ต (Package Design Unit)
4. หน่วยการออกแบบโครงแบบ (Configuration Design Unit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้น ที่เรียกว่า หน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.1 แสดงให้เห็นโครงสร้างอย่างง่ายๆ ของ หน่วยการออกแบบเอนทิตี

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

รูปที่ 2.1 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

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

2.2.1.1 port ชนิดต่างๆ

พอร์ตที่ใช้กันส่วนใหญ่ในภาษา VHDL มีดังนี้

1. *in :* รับข้อมูลไหลเข้าวงจรได้อย่างเดียว
2. *out :* ส่งข้อมูลไหลออกจากวงจรได้อย่างเดียว
3. *inout :* ข้อมูลสามารถไหลเข้าและออกจากพอร์ตนี้ได้

2.2.1.2 การตั้งชื่อ (Identifier)

ในการเขียนโปรแกรมด้วยภาษาอะไรก็แล้วแต่ จะต้องมีการตั้งชื่อเสมอ เช่นในการเขียนโปรแกรมภาษาปาสคาลจะต้องมีการตั้งชื่อโปรแกรม ชื่อตัวแปร ชื่อโพรซีเยอร์ ชื่อฟังก์ชั่น เป็นต้น ในภาษา VHDL ก็เช่นเดียวกัน จะต้องมีการตั้งชื่อของส่วน entity ชื่อของ signal ชื่อของส่วน architecture ชื่อของตัวแปร ชื่อของคอมโพเนนท์ที่ใช้งาน เป็นต้น

ชื่อ(Identifier) ที่จะตั้งในภาษา VHDL จะต้องประกอบด้วยลำดับของตัวอักขระตั้งแต่ 1 ตัวขึ้นไป โดยมีความยาวของชื่อได้ไม่จำกัด ค่าที่เป็นไปได้คือ ตัวอักษรพิมพ์ใหญ่ (A...Z) ตัวอักษรพิมพ์เล็ก (a...z) ตัวเลข (0...9) และเครื่องหมาย underscore (_) โดยมีกฎอยู่ว่า Identifier จะต้องขึ้นต้นด้วยตัวอักษร(A...Z) a...z) และต้องไม่ลงท้ายด้วยเครื่องหมาย underscore โดยในภาษา VHDL นั้นจะถือว่าตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ไม่มีความแตกต่างกัน (non-case sensitive) ยกตัวอย่างเช่น ADDER, adder, Adder และ AddER จะถือว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวเดียวกัน ในการใช้เครื่องหมาย underscore นั้นจะต้องไม่ใช้ติดกัน นอกจากนี้แล้ว ในการตั้งชื่อจะต้องไม่ใช้คำสงวน (reserved word) ของภาษา VHDL ด้วย ตัวอย่างของ การตั้งชื่อที่ถูกต้องได้แก่

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

2.2.1.3 VHDL Operators

Operator ที่ถูกกำหนดไว้แล้ว (predefined operator) ในภาษาวีเอชดีแอลนั้นสามารถแบ่งออกเป็น 4 ประเภท ดังต่อไปนี้

1. Logical operators
2. Relational operators
3. Arithmetic operators
4. Concatenation operator

2.2.1.3.1 Logical Operator

ใช้ในการทำโอเปอเรชั่นทางลอจิก ซึ่งมีอยู่ 6 ตัวด้วยกัน อันได้แก่

and or nand nor xor not

โอเปอเรเตอร์ดังกล่าวใช้ได้สำหรับข้อมูลชนิด bit, bit_vector และ boolean โดยที่โอเปอเรชั่น not จะมีลำดับความสำคัญ(precedence) สูงสุด ส่วนโอเปอเรชั่น and, or, nand, nor และ xor จะมีความลำดับความสำคัญเท่ากัน และมีลำดับความสำคัญน้อยกว่าโอเปอเรชั่น not

ในการทำโอเปอเรชั่นทางลอจิกนั้นข้อมูลที่นำมาทำโอเปอเรชั่นกันจะต้องเป็นข้อมูลชนิดเดียวกัน และผลลัพธ์จากการทำโอเปอเรชั่นให้เอาต์พุตชนิดเดียวกันกับอินพุต ตัวอย่างของการทำโอเปอเรชั่นทางลอจิกได้แก่

```
A and B
not (CLK or RESET)
ADDREESS_BUS(2 downto 0)  nor "010"
```

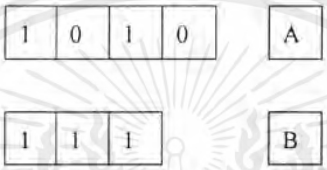
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3.2 Relational operators

ใช้ในการเปรียบเทียบโอเปอแรนด์ชนิดอะไรก็ได้ยกเว้น file type ซึ่งผลลัพธ์จากการเปรียบเทียบจะมีชนิดเป็น boolean โดยที่โอเปอเรเตอร์แบบ relational มีดังนี้

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

ในการเปรียบเทียบออบเจกต์ที่เป็นชนิด bit_vector นั้น ออบเจกต์ที่นำมาเปรียบเทียบกันไม่จำเป็นจะต้องมีจำนวนบิทเท่ากันก็ได้ ดังตัวอย่าง



ผลลัพธ์จากการเปรียบเทียบ A กับ B ก็คือ B มีค่ามากกว่า A เพราะจำนวนบิทของการการเปรียบเทียบจะเท่ากับจำนวนบิทของโอเปอแรนด์ตัวที่มีจำนวนบิทน้อยที่สุด โดยเริ่มเปรียบเทียบจาก MSB ลงมาจนกระทั่งถึง LSB ดังนั้น ค่า "111" จึงถูกเปรียบเทียบกับค่า "101" จึงให้ผลออกมาว่า B มีค่ามากกว่า A

2.2.1.3.3 Arithmetic operators

ใช้ในการทำโอเปอเรชั่นทางคณิตศาสตร์ ซึ่งแสดงดังตารางที่ 2.1

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

ตารางที่ 2.1 การทำโอเปอเรชั่นทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3.4 Concatenation operators

ใช้ในการเชื่อมต่อข้อมูลตัวเดี่ยวๆ หรือข้อมูลชนิดอะเรย์ 1 มิติเข้าด้วยกันโดยให้ผลลัพธ์ออกมาเป็นข้อมูลชนิดอะเรย์เสมอ ซึ่งสัญญลักษณ์ที่ใช้คือเครื่องหมาย & ตัวอย่างเช่น

'0' & '1'	จะได้ผลลัพธ์เท่ากับ "01"
'F' & 'A' & 'T'	จะได้ผลลัพธ์เท่ากับ "FAT"
"FRO" & "NT"	จะได้ผลลัพธ์เท่ากับ "FRONT"

2.2.1.4 Data Objects

Data Object มีอยู่ 3 ชนิด อันได้แก่

1.Constant ออบเจ็กต์ที่เป็นชนิด constant(ค่าคงที่) จะสามารถเก็บค่าของข้อมูลชนิดที่ระบุให้กับออบเจ็กต์ได้แค่หนึ่งค่าเท่านั้น โดยค่าดังกล่าวจะถูกกำหนดให้กับออบเจ็กต์ก่อนที่จะมีการ simulation เกิดขึ้นและค่าของมันจะไม่เปลี่ยนแปลงเลยตลอดการ simulation

2. Variable ออบเจ็กต์ชนิด variable(ตัวแปร) นี้สามารถเก็บค่าของข้อมูลชนิดทีระบุให้กับออบเจ็กต์นี้ได้ แต่ออบเจ็กต์ชนิดนี้สามารถรับค่าอื่นเข้าไปเก็บไว้ในภายหลังได้ด้วยการใช้ variable assignment statement (ซึ่งก็คือเครื่องหมาย := นั่นเอง)

3. Signal ออบเจ็กต์ชนิด signal(สัญญาณ) นี้จะเป็นของสัญญาณซึ่งจะมีค่าของเดิมของสัญญาณ, ค่าปัจจุบัน และเซทของค่าในอนาคต โดยที่ ค่าในอนาคตของสัญญาณนี้จะถูกกำหนดให้โดยใช้ signal assignment statement (ซึ่งก็คือเครื่องหมาย <= นั่นเอง)

ออบเจ็กต์ที่เป็นแบบ signal จะเปรียบเหมือนสายไฟที่ใช้เชื่อมต่อภายในวงจร ในขณะที่ variable นั้นจะเปรียบเหมือนกับตัวแปรที่ใช้กันในการเขียนโปรแกรมภาษาระดับสูงเช่นภาษาซีหรือภาษาปาสคาลโดยทั่วไป signal จะใช้ในการสร้างสายไฟหรือฟลิปฟลอปภายในวงจรในขณะที่ variable และ constant จะใช้ในการบรรยายพฤติกรรมของวงจร

การประกาศออบเจ็กต์(object declaration) จะใช้ในการประกาศตัวออบเจ็กต์และชนิดของออบเจ็กต์(object type) ตลอดจนถึง ประเภทของมันด้วย(ว่าเป็นออบเจ็กต์ประเภท constant, variable หรือ signal กันแน่) และในบางครั้งการประกาศออบเจ็กต์อาจจะมีการกำหนดค่าเริ่มต้น(initial value) ของออบเจ็กต์นั้นอีกด้วย

2.2.1.4.1 การประกาศค่า constant (constant declaration)

ตัวอย่างของการประกาศค่า constant ได้แก่

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

รูปที่ 2.2 ตัวอย่างของการประกาศค่า constant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 เป็นการประกาศออบเจ็กต์ชื่อ CLOCK_PERIOD ให้มีชนิดของออบเจ็กต์เป็นชนิด time (เป็น type มาตรฐานตัวหนึ่งในภาษาวีเอชดีแอล) และมีค่าเท่ากับ10 ns ตัวอย่างที่สองเป็นการประกาศออบเจ็กต์ชื่อ BUS_WIDTH ให้มีชนิดของออบเจ็กต์เป็นชนิด integer ดังนั้นค่าที่ ออบเจ็กต์ BUS_WIDTH จะเก็บได้ก็คือตัวเลขจำนวนเต็ม ซึ่งมีค่าเท่ากับ 8

2.2.1.4.2 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 ตัวอย่างที่สองเป็นการประกาศ variable ชื่อ REMAINDER ซึ่งมีชนิดเป็น integer โดยมีการกำหนดขอบเขตให้ค่าที่ REMAINDER จัดเก็บได้นี้จำกัดอยู่ที่ค่าในช่วง 0 ถึง 127 เท่านั้นและค่าเริ่มต้น(initial value) ของ variableมีค่าเท่ากับ 10
ตัวอย่างที่ 3 เป็นการประกาศ variable ชื่อ INT_SIGNAL ให้มีชนิดเป็น bit_vector ซึ่งในการประกาศไม่มีการกำหนดค่าเริ่มต้นเอาไว้

2.2.1.4.3 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' ตัวอย่างที่สองเป็นการประกาศ signal ชื่อ SUM ให้มีชนิดเป็นอะเรย์ของข้อมูลชนิด bit ขนาด 32 บิท ตัวอย่างสุดท้ายเป็นการประกาศ signal ชื่อ DIVIDEND ให้มีชนิดเป็น integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.5 ชนิดของข้อมูล (Data type)

Data type ที่มีอยู่แล้วในภาษาวีเอชดีแอล (predefined type) ได้แก่

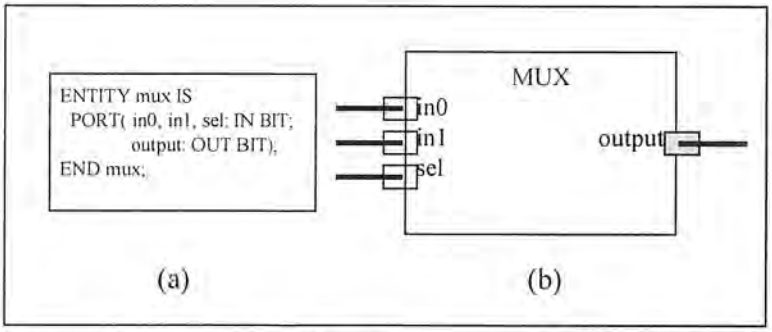
1. CHARACTER มีค่าที่เป็นไปได้ดังนี้

```
nul, soh, stx, etx, eot, enq, 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

2. BIT มีค่าที่เป็นไปได้ คือ '0' และ '1'
3. BIT_VECTOR เป็นอะเรย์ของข้อมูลขนิดบิท
4. BOOLEAN มีค่าที่เป็นไปได้คือ true และ false
5. INTEGER มีค่าที่เป็นไปได้คือ 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

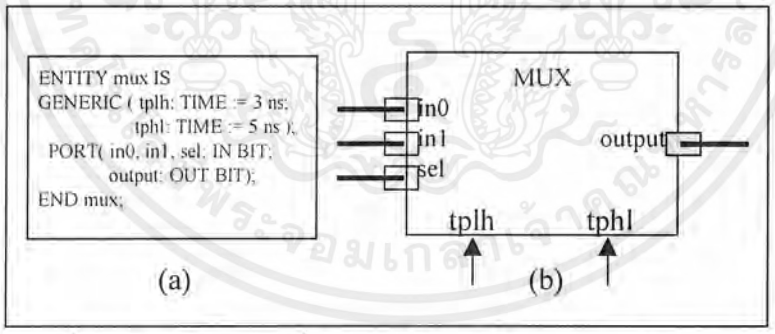
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงรูปแบบของมัลติเพลกซ์ (a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล (b) มุมมองของตัวเชื่อมประสาน (Interfacing)

ในรูปที่ 2.6 เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพลกซ์ 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 ส่วนประเภทของข้อมูลที่จะไหล เข้าและออก นั้นเป็นประเภท BIT ที่สามารถมีค่าได้เพียงสองค่าคือ '0' และ '1' เท่านั้น

นอกจากนั้นผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ที่เป็นข้อมูลเพิ่มเติมอื่นๆ ลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์ อันได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างในรูปที่ 2.7



รูปที่ 2.7 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย (a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล (b) มุมมองของตัวเชื่อมประสาน

ในบางกรณีสามารถที่จะใช้ภาษาวีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหล เข้าและออกของข้อมูลได้ดังรูปที่ 2.8 ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบ สำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่ง คือ วีเอชดีแอลสำหรับการทดสอบเปรียบเทียบ (Test bench)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.8 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

2.2.2 หน่วยการออกแบบสถาปัตยกรรม

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงาน พฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดใน หน่วยการออกแบบเอนทิตี รูปที่ 2.9 แสดงให้เห็นถึงโครงสร้างอย่างง่ายๆ ของหน่วยการออกแบบสถาปัตยกรรม

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

รูปที่ 2.9 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า ARCHITECTURE นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OF <entity design unit> IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิ เช่นประเภท (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 ที่เป็นส่วนเผื่อเลือกโดยทั่วไป การเขียนรูปแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

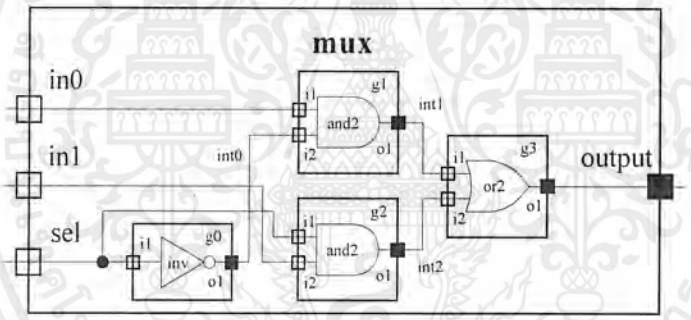
1. ประเภทการไหลของข้อมูล (Dataflow description)
2. ประเภทพฤติกรรม (Behavioral description)
3. ประเภทโครงสร้าง (Structure description)
4. ประเภทผสม (Mixed model description)

```
ARCHITECTURE data_flow OF mux IS
BEGIN
    output <= ((NOT sel) AND in0) OR (sel AND in1);
END data_flow;
```

รูปที่ 2.10 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน

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

รูปที่ 2.10 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (in0, in1) กับข้อมูลที่ไหลออก (output) ประกอบด้วยชุดคำสั่งแบบแข่งขนานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพลกซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างเรจิสเตอร์ (RTL: Register Transfer Level)



รูปที่ 2.11 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์

รูปที่ 2.11 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง โดยใช้ อินเวอร์เตอร์ (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
  PORT ( i1, i2 : IN BIT ; o3 : OUT BIT );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    COMPONENT or2
    PORT ( i1, i2 : IN BIT ; o1 : OUT BIT );
END COMPONENT;
    SIGNAL int0, int1, int2 : BIT;
BEGIN
    g0 : inv   PORT MAP (i1 => sel, o1 => int0);
    g1 : and2 PORT MAP (i1 => in0, i2 => int0, o1 => int1);
    g3 : or2   PORT MAP (i1 => int1, i2 => int2, o1 => ouput);
END struc;
```

รูปที่ 2.12 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง

จากรูปที่ 2.13 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรมซึ่งการเขียนจะมีลักษณแตกต่างกับแบบโครงสร้างแต่การทำงานเหมือนกัน

```
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 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพลกซ์ในลักษณะของ ประเภทพฤติกรรม ประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งนี้ก็เป็นข้อดีของภาษาวีเอชดีแอล

2.2.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดย หน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือ จากหน่วยการออกแบบแพ็กเก็ตอื่นๆ นอกจากนั้นสิ่งที่นิยมทำกันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.2.3.1 Package Declaration

ส่วนที่มีความสำคัญที่สุดของแพ็กแก็ต (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กแก็ต เพราะจะเป็นส่วนที่กำหนดชื่อ ของสิ่งที่ประกาศอยู่ภายในแพ็กแก็ต สำหรับนำไปใช้ภายนอกตัวของแพ็กแก็ตเอง สิ่งใดๆ ถูกประกาศในส่วนของ ส่วนบอดี้แพ็กแก็ต แต่ไม่ถูกประกาศในส่วนการประกาศแพ็กแก็ต จะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือ จุดเชื่อมต่อ หรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้ว แพ็กแก็ต สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศ ชนิด (Type) หรือ สัญญาณ เช่นเดียวกันกับ ส่วนบอดี้แพ็กแก็ต ที่ไม่จำเป็นต้องมี ส่วนของการประกาศแพ็กแก็ต แต่แพ็กแก็ต นั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้ ดังรูปที่ 2.14

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

รูปที่ 2.14 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กแก็ต

2.2.3.2 Package Body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กแก็ต แล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กแก็ต ทั้งนี้รวมทั้ง การกำหนดค่าคงที่ต่างๆ อันได้แก่ตังคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กแก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กแก็ต) ฉะนั้นส่วนบอดี้แพ็กแก็ต จึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กแก็ต ไม่มีการประกาศชื่อ ที่เป็นโปรแกรมย่อยหรือ ค่าคงที่ การเขียนบอดี้แพ็กแก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.15

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

รูปที่ 2.15 โครงสร้างของบอดี้แพ็กแก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตี หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกันดังรูปที่ 2.16

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

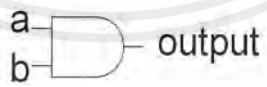
รูปที่ 2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.3 ภาษาวีเอชดีแอลเพื่อการสังเคราะห์

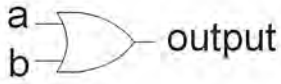
ภาษาวีเอชดีแอล เป็นภาษาที่เขียนเพื่อจำลองการทำงานของวงจร ซึ่งในบางรูปแบบการเขียนไม่สามารถที่จะนำไปสังเคราะห์ได้ทั้งหมด ดังนั้นถ้าต้องการเขียนเพื่อนำไปสังเคราะห์ ควรหลีกเลี่ยงรูปแบบต่างๆ ที่ไม่สามารถนำไปสังเคราะห์ได้ ในที่นี้ขึ้นอยู่กับความสามารถของโปรแกรมที่ใช้สังเคราะห์แต่ละโปรแกรม ดังนั้นในหัวข้อนี้จะแสดงตัวอย่างของการเขียนโมเดลในรูปแบบต่างๆ ที่สามารถนำไปสังเคราะห์ ซึ่งยึดหลักการเขียนตาม ViewSynthesis User's Guide ของโปรแกรม Viewlogic ซึ่งเป็นโปรแกรมที่ใช้ในสังเคราะห์วงจรทั้งหมดในการออกแบบไมโครคอนโทรลเลอร์ โดยแบ่งออกเป็น 2 กลุ่ม คือ เกตและฟลิปฟลอปประเภทต่างๆ

2.3.1 ตัวอย่างรูปแบบการเขียนเกตพื้นฐาน

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



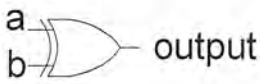

```
ouput <= a AND b;
...
```



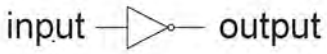

```
output <= a OR b;
...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

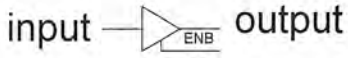
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



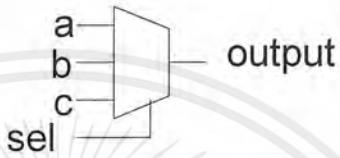

```
output <= a XOR b;
...
```




```
output <= NOT input;
...
```



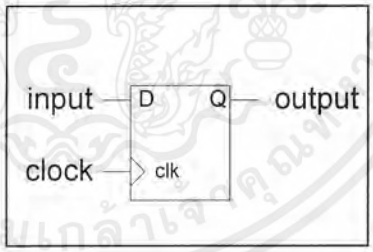

```
output <= input WHEN enb='1'  ELSE "ZZ";
...
```




```
WITH sel SELECT
        Output  <=    a WHEN "00" ELSE
                      b WHEN "01" ELSE
                      c;
```

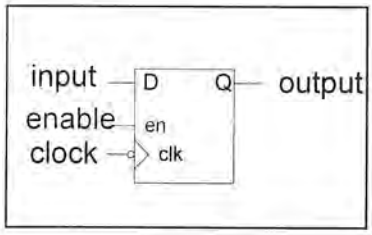
2.3.2 ตัวอย่างรูปแบบการเขียนฟลิปฟลอปพื้นฐาน

```
SIGNAL input, output : vlbit_1d(3 DOWNTO 0);
SIGNAL clock, enable, reset : vlbit;
...
```

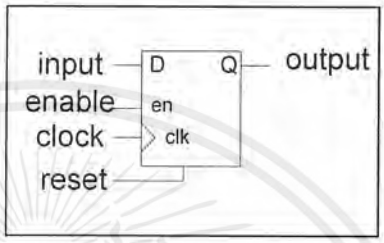



```
PROCESS BEGIN
        WAIT UNTIL PRISING(clock);
                output <= input;
END PROCESS;
...
```

37041
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




```
PROCESS BEGIN
        WAIT UNTIL PFALLING(clock);
                output <= input;
END PROCESS;
    ...
```

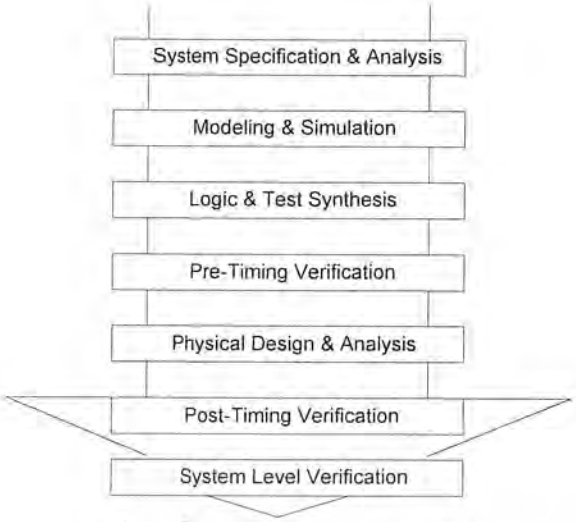



```
PROCESS BEGIN
        WAIT UNTIL PRISING(clock) OR (reset='1');
        IF (reset = '1') THEN output <= "00";
        ELSIF (enble = '1') THEN ouput <= input;
        END IF;
END PROCESS;
```

2.4 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน เช่น วงจรรวม (ASIC: Application Specific Integrated Circuit) วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของของ บล็อกไดอะแกรมเสียก่อน ก่อนที่จะวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามที่ต้องการ และเพิ่มเติมในรายละเอียดที่ละชั้นนี้คือ หลักการออกแบบจากบนลงล่าง (Top-Down Design) ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Botom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากให้การออกแบบในลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอลกับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรออกแบบที่จะสามารถออกแบบและพัฒนาวงจรที่มีซับซ้อนได้มากขึ้น และช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.17 แสดงให้ถึงขั้นตอนของการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายๆ เทคโนโลยี เช่น พีแอลดี (PLD: Programmable Logic Device) อันได้แก่ พีแอลเอ (PLA: Programmable Logic Array), เอฟพีจีเอ (FPGA: Filed Programmable Gate Array), ซีพีแอลดี (CPLD: Cell Programmable Logic Device) เป็นต้น นอกนั้นยังมี เซมิคัสตัมไอซี (Semi-Custom IC) ได้แก่ เกตอะเรย์ (Gate array), เซลล์มาตรฐาน (Standard Cell) ขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียดดังนี้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา
2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษาวีเอชดีแอล หรือ ภาษาเอชดีแอลอื่นๆ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด
3. หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจร หรือสังเคราะห์ ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกต และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสท์ (Netlist) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้
4. หลังจากการสังเคราะห์วงจรให้อยู่ในรูประดับเกตหรือเน็ทลิสท์แล้ว ข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงาน ในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี เวลาหน่วงของการแพร่กระจาย (Propagation delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ นาโนวินาที (10^{-9} นาที) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

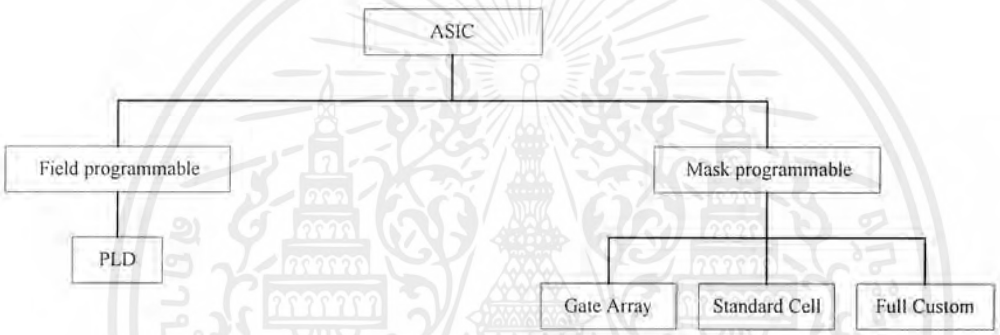
ต่างๆ จำนวน 10,000 เกต ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรรวมทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวม (ASIC)
6. หลังจากที่ได้วงจรจริงมาแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วยอินพุตและเอาต์พุตแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก
7. หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้ง เป็นการควบคุมคุณภาพของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
เอฟพีจีเอ

ความก้าวหน้าของอุตสาหกรรมอิเล็คทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่าย การสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโพรเซสเซอร์และหน่วยความจำปัจจุบัน ทุกๆ ครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้น ในการพัฒนาเพิ่มความหนาแน่นและจำนวน ฟังก์ชั่นลอจิก ที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิตอลได้พิจารณาถึงการผลิตให้ขนาดมากๆ และการผลิตวงจรรวม (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวม จะแบ่งตามการสร้างออกเป็น 2 กลุ่ม คือ Field programmable และ Mask programmable ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ผังแสดงการแบ่งกลุ่มของวงจรรวม ASIC

3.1 Field Programmable

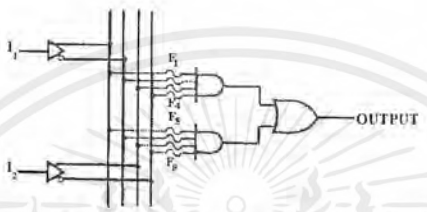
อุปกรณ์วงจรรวม ASIC แบบ field programmable มีอยู่มากมายหลายชนิด แต่มีลักษณะการสร้างหรือกำหนดการทำงานของวงจรที่เหมือนกัน กล่าวคือ ผู้ใช้งานสามารถออกแบบและสร้างวงจรที่ต้องการใช้ลงในตัวอุปกรณ์ได้เองโดยไม่ต้องไปโรงงานเพื่อผลิต โดยเฉพาะอย่างยิ่งในปัจจุบันนี้มีเครื่องมือที่ใช้ช่วยในการออกแบบ และสร้างวงจรร่วมกับไมโครคอมพิวเตอร์ที่มีความสามารถสูงในการพัฒนาตั้งแต่ขั้นการออกแบบ การจำลองการทำงาน จนถึงจัดสร้างวงจรลงในอุปกรณ์ รวมทั้งอุปกรณ์ field programmable เหล่านี้สามารถหาซื้อได้ง่ายทำให้การสร้างวงจรอิเล็กทรอนิกส์จนถึงระบบไมโครโพรเซสเซอร์หันมาใช้อุปกรณ์จำพวกนี้ เป็นอุปกรณ์ประกอบในวงจรแทนอุปกรณ์ย่อยๆ แยกชิ้น (Discrete component) มาก

3.1.1 พีแอลดี (PLD: Programmable Logic Device)

ภายในอุปกรณ์พีแอลดีถูกเตรียมเป็นวงจรพื้นฐานทางด้านลอจิกต่อกันอยู่เป็นกลุ่มมีทั้งวงจรคอมบิเนชั่น (Combination) และซีเควลเชียล (Sequential) ซึ่งมีส่วนประกอบเป็นวงจรภายใน เทคโนโลยีของวงจรที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้สร้างพีแอลดีมีทั้ง ทีทีแอล (TTL) อีซีแอล (ECL) และ ซีมอส (CMOS) ตามความเหมาะสมของแต่ละระบบ อุปกรณ์พีแอลดีทุกชนิดมีหลักการพื้นฐานของวงจรภายในที่เหมือนกันโดยมีวงจรหลักเป็นวงจรคอมบิเนชั่นที่ให้ผลเป็นผลคูณร่วมบวก (Sum of product) ดังรูปที่ 3.2 ประกอบไปด้วยชุดของแอนด์เกตที่ต่อร่วมกับออเกต การโปรแกรมคือ การเลือกว่าจะจะให้มีการต่ออินพุตภายในของแอนด์เกตกับสัญญาณอินพุตใดบ้างซึ่งมีทั้งจากภายนอกและสัญญาณป้อนกลับจากเอาต์พุตภายในเอง การติดต่ออินพุตของออเกตกับเอาต์พุตของ แอนด์เกต ตัวต่างๆ วิธีการเลือกหรือการโปรแกรมทางกายภาพ อินพุตต่างๆ ของอุปกรณ์ทุกตัวจะถูกต่อผ่านฟิวส์เข้ากับแหล่งสัญญาณ ซึ่งถ้าไม่ต้องการใช้สัญญาณใดจะตัดฟิวส์ทำให้สามารถโปรแกรมได้ครั้งเดียว อุปกรณ์พีแอลดีบางชนิดใช้มอสทรานซิสเตอร์แทนฟิวส์ทำให้สามารถโปรแกรมโดยใช้กระแสไฟฟ้า และสามารถลบและโปรแกรมใหม่เข้าไปได้อีก

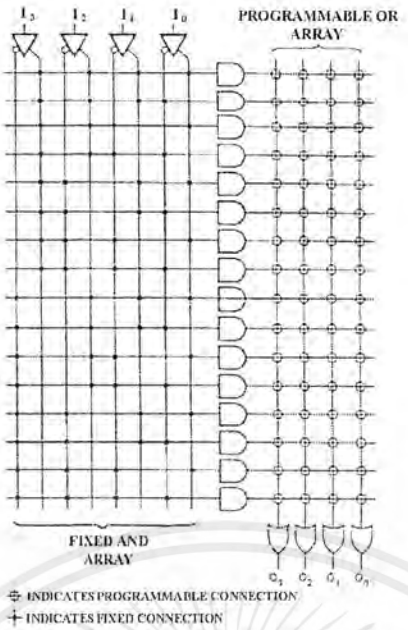


รูปที่ 3.2 แสดงวงจรพื้นฐานของอุปกรณ์พีแอลดีซึ่งอยู่ในรูปผลคูณร่วมบวก

3.1.2 พรอม (PROM: Programmable Read Only Memory)

พรอมคือหน่วยความจำรอม (ROM) ที่โปรแกรมได้ ซึ่งนับว่าเป็นอุปกรณ์พีแอลดี ชนิดหนึ่งซึ่งวงจรภายในของพรอมเสมือนกับประกอบไปด้วยแถวลำดับของแอนด์เกตและออเกต (And – Or Array) ผลเอาต์พุตที่ขาดาต้าเอาต์พุตสามารถแสดงในสมการของฟังก์ชั่นผลคูณร่วมบวก (Sum of product) ของสัญญาณอินพุตที่ขาแอดเดรส รูปที่ 3.3 แสดงถึงลักษณะการต่อเป็นแถวลำดับของแอนด์เกตและออเกตของพรอมขนาด 16x4 บิต วงจรทางด้านซ้ายบนสุดเป็น แอนด์เกตที่ให้ผลเป็นผลคูณ (Product) ของกรณีอินพุตเป็น 0000 แอนด์เกตที่อยู่ถัดลงมาเป็นผลคูณของกรณีที่อินพุต เป็น 0001, 0010, ... จนถึงตัวล่างสุดคือผลคูณในกรณีที่อินพุตเป็น 1111 ที่เอาต์พุตแต่ละบิตของหน่วยความจำ สามารถเลือกได้ว่าจะให้เป็น 1 ในกรณีที่อินพุตจากแอดเดรสเป็นอย่างไรบ้างเหมือนกันเป็นจากนำเอาต์พุตจากผลคูณที่ต้องการให้เอาต์พุตแต่ละบิตเป็น 1 ไปออกันจึงเปรียบเหมือนกับว่าในพรอมมีจำนวนแอนด์เกตเท่ากับจำนวนตำแหน่งความจำและมีออเกตจำนวนเท่ากับจำนวนบิตของสัญญาณข้อมูลออก (Data output) อินพุตของออเกตทุกตัวสามารถต่อเข้ากับแอนด์เกตตัวใดก็ได้ทุกตัว ซึ่งอาจเรียกได้ว่าเป็นพีแอลดีแบบ fixed ND / programm -able OR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงลักษณะของพรอมเมื่อเปรียบเทียบเป็นวงจรในรูปผลคูณร่วมบวก

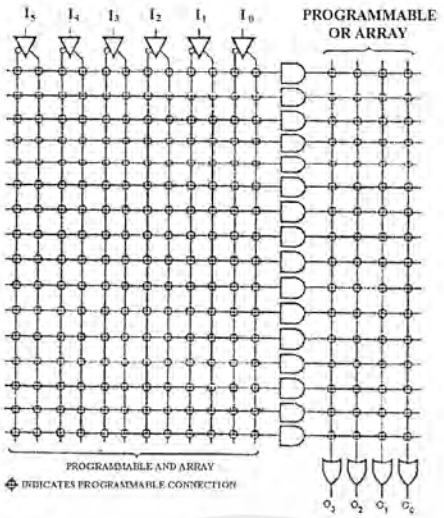
3.1.3 พีเอแอล (PAL: Programmable Array Logic)

ในช่วงกลางปี ค.ศ. 1970 บริษัทเอ็มเอ็มไอ (MMI: Monolithic Memory) ในประเทศสหรัฐอเมริกา ได้พัฒนาอุปกรณ์พีเอแอล เป็นพีแอลดีชนิดใหม่โดยใช้เทคโนโลยีแบบแอลเอสไอ (LSI: Large Scale Integration) สามารถโปรแกรมเลือกวงจรภายใน โดยใช้ฟิวส์ที่เชื่อมต่ออยู่ระหว่างสัญญาณอินพุตภายนอก และการป้อนกลับจากภายในกับแอนด์เกตที่ต่อเป็นฟังก์ชั่นผลคูณ (Product) อยู่ในตัววงจรรวม

3.1.4 พีแอลเอ (PLA: Programmable Logic Array)

อุปกรณ์ที่สามารถโปรแกรมได้แบบพีแอลเอเกิดขึ้นเมื่อปี ค.ศ. 1975 โดยบริษัทซิกเนทริกส์ (Signetics) สหรัฐอเมริกา ซึ่งเป็นบริษัทผู้ผลิตวงจรรวมรายใหญ่รายหนึ่ง ผลิตและนำเสนออุปกรณ์โดยใช้ชื่อว่า เอฟพีแอลเอ (FPLA: Field Programmable Logic Array) สามารถโปรแกรมการต่อลอจิกทั้งทางด้านแอนด์เกตและออเกตได้ และยังเลือกเอาต์พุตเป็น active high หรือ active low โดยต่อผ่านเอ็กคูลสีบออเกต ให้ทำหน้าที่เป็นนอนอินเวอร์เตอร์หรือเป็น อินเวอร์เตอร์แล้วแต่ภายในของพีแอลเอต่อมาปี ค.ศ. 1979 บริษัทซิกเนทริกส์ ได้สร้างเอฟพีแอลเอใหม่ที่มีรีจิสเตอร์ต่ออยู่ภายในวงจรเพิ่มขึ้นรวมทั้งสามารถเลือกสัญญาณอินพุตที่มาจากการป้อนกลับจากรีจิสเตอร์ได้ด้วยดังรูปที่ 3.4 ทำให้สามารถใช้อุปกรณ์พีแอลเอใหม่นี้สร้างวงจร State machine ได้ อุปกรณ์ใหม่ที่มีรีจิสเตอร์ อยู่ด้วยนี้ถูกตั้งชื่อใหม่เป็น เอฟพีแอลเอส (FPLS: Field Programmable Logic Sequencer) มีทั้งที่เป็นทีทีแอลและซีมอส

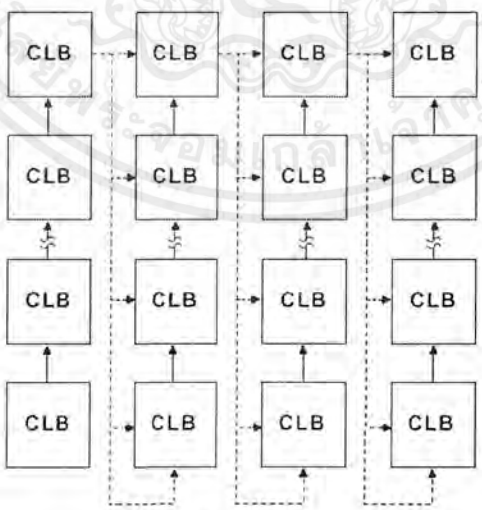
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงวงจรพื้นฐานภายในของพีแอลเอ

3.1.5 แอลซีเอ (LCA: Logic Cell Array)

อุปกรณ์ชนิดนี้ถูกสร้างขึ้นเมื่อประมาณปี ค.ศ. 1986 โดยบริษัทไซริ้ง (XILINX Inc.) ซึ่งเป็นบริษัทที่ร่วมทำการค้นคว้ากับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นอาเรย์ที่ประกอบด้วยเกตจำนวน 1,200-1,800 เกต มีลักษณะของสถาปัตยกรรมที่ใกล้เคียงกับเกตอาเรย์ (Gate array) โดยโปรเซสแบบซีมอส 1.6 ไมครอนขึ้นโลหะคู่ (CMOS 1.6 microns double-layer metal) สามารถโปรแกรมและลบได้โดยใช้กระแสไฟฟ้า (Static RAM based) ภายในจัดเรียงเป็นเมทริกซ์ของลอจิกเซลล์ดังรูปที่ 3.5 และล้อมรอบภายนอกด้วยอินพุต เอาต์พุตเซลล์ อุปกรณ์แอลซีเอตัวแรกของบริษัทไซริ้งคือ แอลซีเอเบอร์ 2064 ประกอบด้วยเซลล์เรียงเป็นเมทริกซ์มีจำนวน 64 เซลล์ แต่ละเซลล์เรียกว่า ซีแอลบี (CLB: Configurable Logic Block) แต่ในปัจจุบันได้พัฒนามาอยู่ในรูปของ เอฟพีจีเอ (FPGA: Field Programmable Gate Array) ซึ่งมีประสิทธิภาพ ความจุของเกตสูงมากขึ้น โดยสร้างออกมาเป็นอนุกรม (Series) ต่างๆ เช่น ตระกูล XC3000 และ ตระกูล XC4000 เป็นต้น

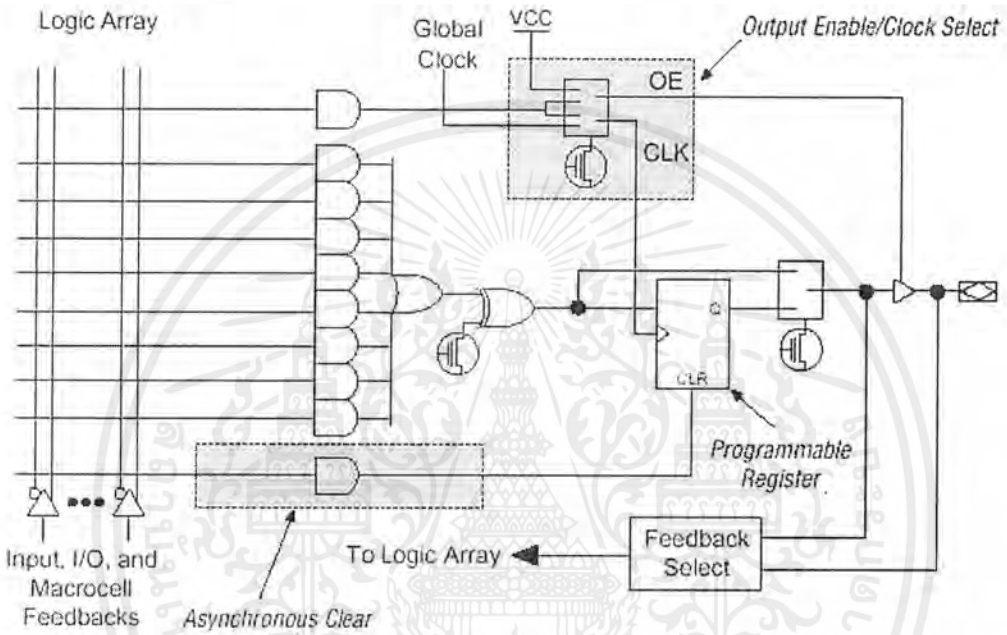


รูปที่ 3.5 แสดงการจัดเรียงเป็นเมทริกซ์ของลอจิกเซลล์ของ LCA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 อีพีแอลดี (EPLD: Erasable Programmable Logic Device)

อีพีแอลดีเป็นอุปกรณ์สามารถโปรแกรมได้ที่สามารถลบและกระทำการโปรแกรมใหม่ได้ เพื่อใช้ทำวงจรต้นแบบ ตัวอย่างได้แก่ พีแอลดี ในอนุกรมอีพี (EP series) ของบริษัทอัลเทอร่า (Altera Inc.) ประเทศสหรัฐอเมริกาดังรูปที่ 3.6 ซึ่งเป็นบริษัทที่ผลิตอีพีแอลดี เป็นรายแรกโดยเริ่มเมื่อปี ค.ศ. 1984 เป็น พีแอลดี ที่ใช้โปรเซสเหมือนกับซีมอสอีพรอม (CMOS EPROM) คือใช้มอสทรานซิสเตอร์เชื่อมต่อระหว่างสัญญาณอินพุตกับจุดที่ต้องการแทนการการใช้ฟิวส์ดังเดิม ทำให้สามารถโปรแกรมการต่อวงจรภายในอุปกรณ์ด้วยการจ่ายไฟฟ้าตามขนาดที่กำหนดเข้าไปยังตัวอุปกรณ์ และลบได้โดยใช้แสงอุลตร้าไวโอเลทฉายผ่านช่องหน้าต่างกระจกเข้าไปตกกระทบในตัวชิปของอุปกรณ์



รูปที่ 3.6 ลักษณะของพีแอลดีในอนุกรมอีพีของบริษัทอัลเทอร่า

3.2 Mask Programmable

การใช้งานวงจรรวม ASIC ในเชิงพาณิชย์ จำเป็นต้องใช้วงจรรวม ASIC แบบ Mask programmable เนื่องจากต้นทุนต่อหนึ่งตัวต่ำกว่าวงจรแบบ Field programmable ASIC ในกรณีที่ปริมาณการผลิตสูงนับพันพันนับหมื่นตัวขึ้นไป ตัวอย่างเช่น วงจรอีพีแอลดี ตัวหนึ่งอาจสูงถึงหนึ่งพันบาท ในขณะที่ถ้าผลิตวงจรรวมที่มีคุณสมบัติเหมือนกันทุกประการโดยใช้ Mask programmable แล้ว ราคาตัวหนึ่งจะลดลงเหลือเพียงไม่ถึงหนึ่งร้อยบาท การใช้วงจรรวมแบบ Mask programmable จึงมีบทบาทสำคัญในการผลิตสินค้าอิเล็กทรอนิกส์ในเชิงพานิชย์ในปัจจุบัน

วงจรรวมประเภทนี้ หลังจากผู้ใช้ออกแบบวงจรและตรวจสอบการทำงานจนเป็นที่น่าพอใจแล้ว ต้องส่งให้ผู้ผลิตทำการเจือสาร ไม่สามารถโปรแกรมได้ด้วยตนเองเหมือนกับวงจรรวมแบบ Field programmable ช่วงเวลาการผลิตออกใช้งานจึงใช้เวลานับเดือนและมีค่าใช้จ่ายเบื้องต้นในการเจือสารสูง วงจรรวมแบบ mask programmable ASIC ในปัจจุบันได้แก่ เกตอาเรย์, เซลล์มาตรฐานและฟูลัสตัม (full custom)

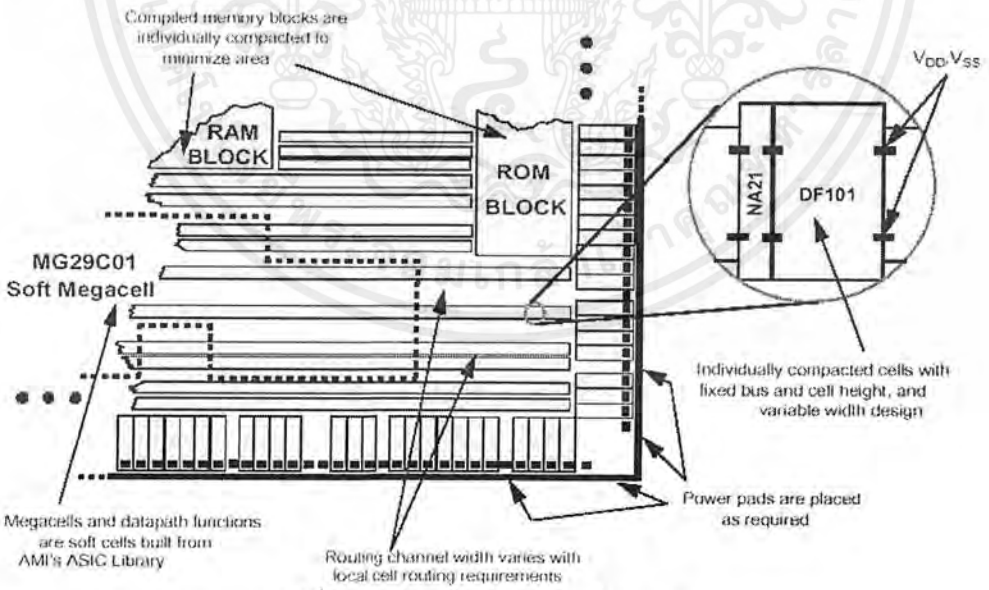
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 เกตอาเรย์ (Gate Array)

วงจรรวมนี้ประกอบด้วยแถวลำดับของวงจรเกต ซึ่งอาจจะเป็นวงจรเกตประเภทเดียวกัน หรือต่างชนิดกันก็ได้ กับขั้วต่อสายไฟ (Pad) สำหรับต่อกับวงจรภายนอก ผู้ใช้มีหน้าที่ออกแบบการเชื่อมโยงทางไฟฟ้าระหว่างวงจรเกตแต่ละตัวและขั้วต่อสายไฟเพื่อให้ทำหน้าที่ตามต้องการ แล้วจึงส่งผลการออกแบบนั้นไปยังโรงงานผู้ผลิตวงจรรวมเกตอาเรย์ นั้นไปทำการเจือสารต่อไป โดยทั่วไปแล้ว ปรากฏในการใช้วงจรรวมเกตอาเรย์สร้างวงจรที่ต้องการใช้งานทั่วไปจะมีเพียง 25-30% ของพื้นที่ซิลิคอนที่ใช้สำหรับวงจรเกต ส่วนพื้นที่ที่เหลือจะใช้กับการเชื่อมโยงวงจรเกตเข้าด้วยกันเองและเข้ากับขั้วต่อสายไฟ พื้นที่ซิลิคอนจะใช้ประโยชน์สูงถึง 75% สำหรับวงจรเชิงเลขที่มีลักษณะสม่ำเสมอ เช่น หน่วยความจำ เป็นต้น

3.2.2 เซลล์มาตรฐาน (Standard Cell)

ปัญหาการใช้พื้นที่ซิลิคอนอย่างมากมายสำหรับการเชื่อมโยงวงจรของวงจรรวม เกตอาเรย์ทำให้เกิดขีดจำกัดในความซับซ้อนของวงจรที่ใช้ ประกอบกับผู้ใช้จำนวนมากต้องการรวบรวมวงจรรวมมาตราฐาน อาทิ วงจรตระกูล 7400 ,วงจรตระกูล 4000 ได้รวมจำนวนหลายตัวเข้าเป็นวงจรรวม ASIC เพียงตัวเดียว ทำให้เกิดวงจรรวม ASIC แบบเซลล์มาตรฐานขึ้น ดังรูปที่ 3.7 วงจรรวม เซลล์มาตรฐานนี้ ผู้ใช้เป็นผู้เลือกกลุ่มวงจรทำหน้าที่ต่างๆ เช่น เกต,ฟลิบฟลอป,ตัวนับ, ตัวเลื่อน ,หน่วยความจำ หรือกระทั่งไมโครโปรเซสเซอร์จากแฟ้มข้อมูล (Library) ของผู้ผลิต ซึ่งอาจจะเป็นแฟ้มข้อมูลคอมพิวเตอร์ เหมาะสมสำหรับวงจรเชิงเลขที่มีความซับซ้อน การผลิตวงจรรวม เซลล์มาตรฐานจะมีต้นสูงกว่าวงจรรวมเกตอาเรย์และใช้เวลาในการออกแบบและเจือสารยาวกว่าสองถึงสามเท่าตัว วงจรรวมเซลล์มาตรฐานจึงเหมาะสมกับการใช้งานเชิงพานิชย์ที่มีปริมาณการผลิตนับหมื่นตัวขึ้นไป บทบาทวงจรรวมเซลล์มาตรฐานจะมีเพิ่มมากขึ้นในอนาคต เมื่อต้นทุนการเจือสารลดลง



รูปที่ 3.7 แสดงโครงสร้างของเซลล์มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ฟูลคัสตัม (Full Custom)

วงจรรวมฟูลคัสตัม นี้ผู้ใช้เป็นผู้ออกแบบเองทั้งหมด ตั้งแต่ระดับวงจรเชิงเลข (Layout) จนกระทั่งถึงระดับกายภาพ แล้วจึงส่งข้อมูลการออกแบบไปให้ผู้ผลิตเจือสารในรูปแฟ้มข้อมูลมาตรฐาน เช่น GDS II, CIF การใช้ฟูลคัสตัม ASIC นี้เท่าที่แพร่หลายอยู่ในปัจจุบัน จะเป็นไปเพื่อการศึกษาและการวิจัยและเพื่อการผลิตจำนวนน้อย ส่วนใหญ่จะเป็นการเจือสารในลักษณะที่ใช้ค่าใช้จ่ายร่วมกัน กล่าวคือ รวบรวมการออกแบบหลายวงจรลงบนแผ่นซิลิคอนเดียวกันเพื่อประหยัดค่าเจือสาร

รูปแบบการเจือสารวงจรรวมฟูลคัสตัมที่ประกอบด้วยการออกแบบหลายวงจรรวมอยู่บนแผ่นซิลิ-คอนเดียวกัน มีกระทำอยู่ 3 รูปแบบดังนี้

1. Multi-project wafer (MPW) การเจือสารวงจรรวมฟูลคัสตัมแบบนี้ แผ่นเวเฟอร์ (Wafer) จะแบ่งเป็นส่วนๆ ที่เรียกว่าไค (Die) และไดแต่ละชุดจะเจือสารวงจรต่างกัน ผู้ผลิตที่ให้บริการแบบนี้ได้แก่ โมซิส (MOSIS) และ ออบิต (ORBIT) ในสหรัฐอเมริกา

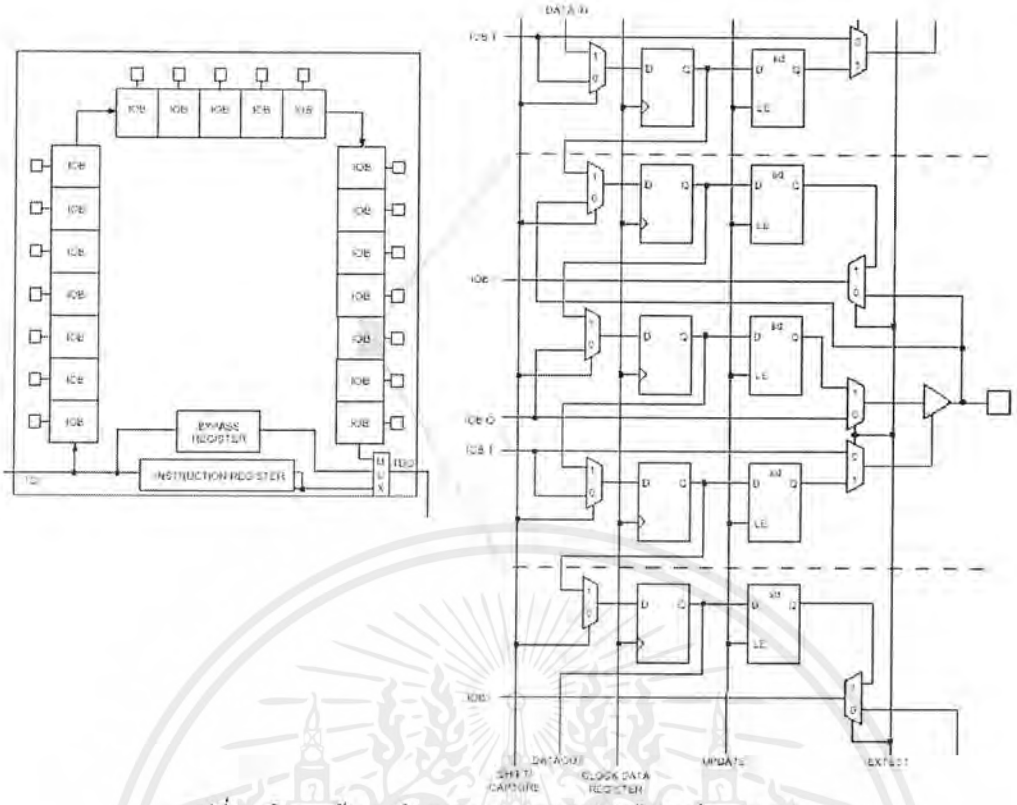
2. Multi-project chip (MPC) เป็นการเจือสารการออกแบบหลายวงจรลงบนไดชุดเดียวกันโดยที่ไดทุกชุดบนเวเฟอร์เดียวกันจะมีลักษณะเหมือนกัน การจัดวางวงจรที่ออกแบบลงบนไค เน้นการใช้พื้นที่ซิลิคอนให้เป็นประโยชน์สูงสุดเป็นสำคัญ ตัวอย่างผู้ผลิตที่ให้บริการแบบนี้ได้แก่ อาวา (AWA) ในออสเตรเลีย และ อีเอสทู (ES II) ในสหราชอาณาจักร

3. Multi-project reticle (MPR) เป็นการเจือสารในลักษณะผสมผสานระหว่างฟูลคัสตัมกับเซลล์มาตรฐานกล่าวคือไดจะแบ่งออกเป็นส่วนๆ อย่างสม่ำเสมอแต่ละส่วนบรรจุการออกแบบแต่ละวงจร โดยไดชุดหนึ่งจะบรรจุวงจรที่ออกแบบได้ประมาณ 8 วงจร และไดทุกชุดในแผ่นเวเฟอร์ จะมีลักษณะเหมือนกัน ในปัจจุบันมีเพียงอาวาที่ใช้บริการการเจือสารลักษณะนี้ โดยจำกัดเฉพาะประเภทเซลล์มาตรฐานเท่านั้น

3.3 เอฟพีจีเอ (FPGA: Field Programmable Gate Array)

เป็นอุปกรณ์ที่ถูกพัฒนาต่อจากอุปกรณ์แอลซีเอของบริษัทไซรินซ์ (XILINX Inc.) โดยมีประสิทธิภาพการทำงานและมีปริมาณความหนาแน่นของเกตสูง สามารถจะกำหนดฟังก์ชั่นการทำงานได้ความต้องการของผู้ใช้โดยผ่านการโปรแกรมเอฟพีจีเอได้รวบรวมข้อดีทั้งหมดของการทำคัสตัมวีแอลเอสไอ (Custom VLSI) มารวมไว้ทั้งหมดได้แก่ การออกแบบการผลิต, ลดเวลาที่จะส่งตัวผลิตภัณฑ์ออกตลาด ซึ่งเป็นประโยชน์ต่อการผลิตวงจรเป็นอย่างมาก นักออกแบบเพียงกำหนดฟังก์ชั่นการทำงานของวงจร ดังนั้นการออกแบบวงจรโดยใช้เอฟพีจีเอ สามารถออกแบบและทดสอบภายในเวลาเพียง 2-3 วัน เท่านั้น ตรงกันข้ามกับการออกแบบโดยใช้เกตอาเรย์ ซึ่งใช้เวลาหลายอาทิตย์ การเปลี่ยนแปลงแก้ไขแบบก็เช่นเดียวกัน จากประโยชน์ของเอฟพีจีเอ ดังกล่าวมา ทำให้เกิดการประหยัดค่าใช้จ่ายเป็นอย่างมาก เพราะได้ความเสี่ยงในการที่จะต้องแก้ไขตัววงจร การเลื่อนเวลาการออกผลิตภัณฑ์ ลดค่าเอ็นอาร์อี (NRE: Nonrecurring Engineering Cost) ลงไปด้วยโครงสร้างภายในของ FPGA จะเป็นดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โครงสร้างภายในของ FPGA ของบริษัทไซรินซ์ (XILINX Inc.)

3.3.1 สถาปัตยกรรมภายในของเอฟพีจีเอตระกูล XC4000

สถาปัตยกรรมภายในคล้ายๆ กับเกตอาเรย์โดยทั่วไปจะมีส่วนประกอบที่สำคัญดังนี้

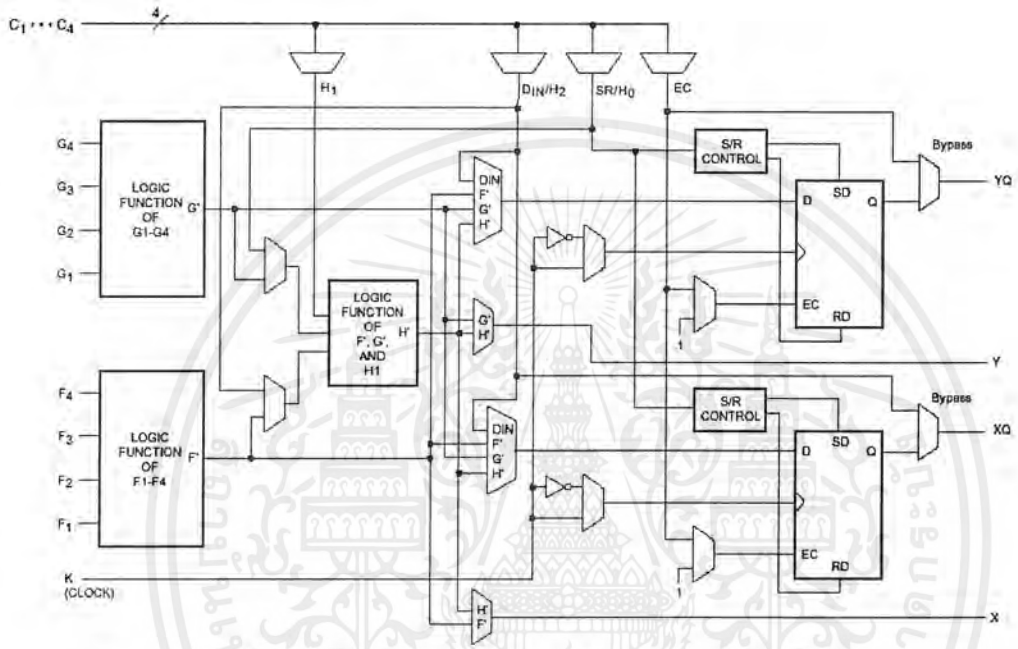
1. ซีแอลบี (CLB: Configuration Logic Block)
2. ไอโอบี (IOB: Input Output Block)
3. อินเตอร์คอนเน็ค (Interconnection)

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 ภายในแอลซีเอโปรแกรมจะถูกโหลดเข้าสู่แอลซีเอเมื่อมีการจ่ายไฟ (Power-up) โดยทางคำสั่ง (Command) ซึ่งเป็นส่วนหนึ่งของการเริ่มต้นระบบ (System initialization) ประสิทธิภาพของแอลซีเอกำหนดโดย ความเร็วของลอจิกส่วนประกอบหน่วยความจำและการโปรแกรมการเชื่อมต่อต่างๆ ความเร็วของอัตราของระบบสัญญาณนาฬิกา (System clock rate) ถูกกำหนดด้วยทอกเกิลฟลิปฟลอป สำหรับการประยุกต์ใช้โดยทั่วไปจะอยู่ที่ประมาณ 1/3 ถึง 1/2 ค่าสูงสุดของทอกเกิลเกต (Maximum toggle gate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.1 ซีแอลบี (CLB: Configuration Logic Block)

ภายในแอลซีเอคือเมทริกซ์ของซีแอลบีแต่ละตัวประกอบด้วยหน่วยของคอมเบนิชั่นลอจิกที่สามารถโปรแกรมได้ (Programmable combination logic) และส่วนของรีจิสเตอร์เก็บข้อมูล (Storage register) ส่วนของวงจรคอมเบนิชั่นลอจิกสามารถใช้สร้างวงจรทางด้านฟังก์ชั่นบูลีนของอินพุต ส่วนเรจิสเตอร์รับค่าจากส่วนคอมบิเนชั่นหรือโดยตรงจากเอาต์พุตของซีแอลบี สามารถขับวงจรคอมบิเนชั่นลอจิกโดยตรงผ่านเส้นทางเดินย้อนกลับ (Feedback path) ดังรูปที่ 3.9

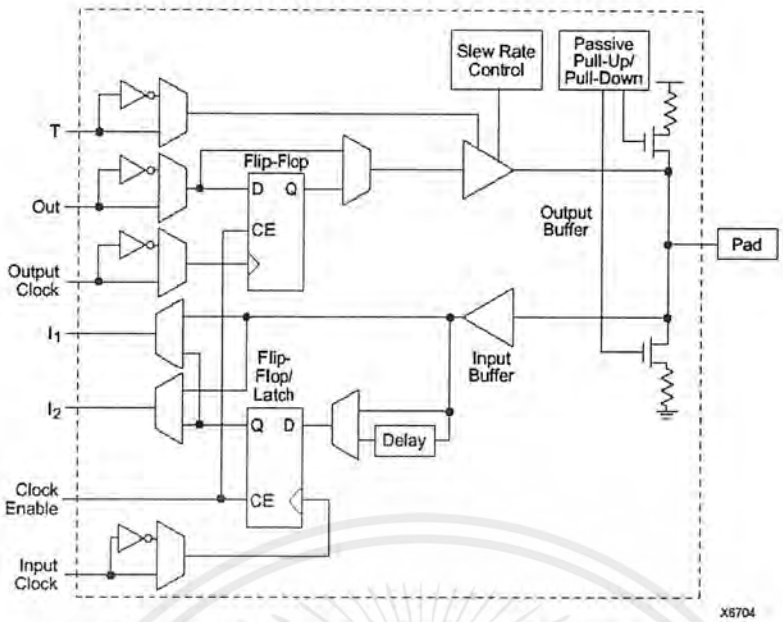


รูปที่ 3.9 แสดงผังวงจรภายในของซีแอลบีของเอฟพีจีเอตระกูล XC4000

3.3.1.2 ไอโอบี (IOB: Input Output Block)

เป็นส่วนติดต่อกับวงจรภายนอกของแอลซีเอสร้างมาจากส่วนของอุปกรณ์อินพุต/เอาต์พุตที่สามารถโปรแกรมได้ (Programmable Input/Output device) แต่ละตัวสามารถโปรแกรมได้อย่างอิสระโดยจะให้เป็นอินพุต/เอาต์พุตแบบ 3 สถานะหรือไอโอแบบสองทิศทางก็ได้ โดยอินพุตสามารถโปรแกรมให้รู้จักทั้งระดับสัญญาณทีทีแอลและซีมอสเทรดโฮล ของไอโอบี แต่ละตัวมีฟลิปฟลอปสามารถใช้เป็นบัฟเฟอร์สำหรับอินพุตและเอาต์พุต ดังรูปที่ 3.10

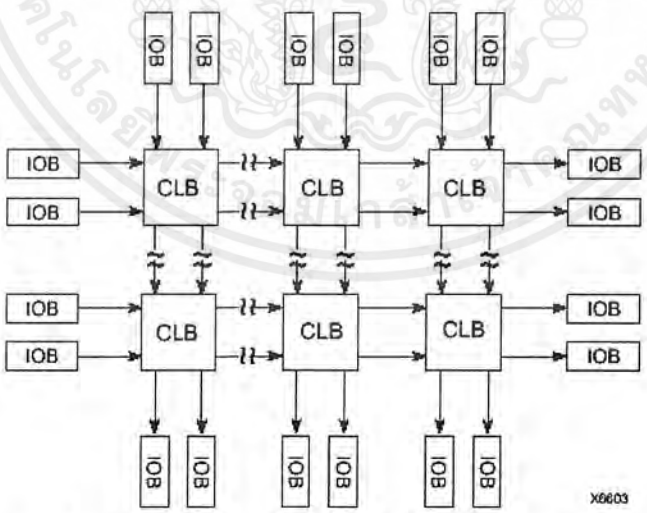
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงผังวงจรของไอโอบีภายในเอฟพีจีเอตระกูล XC4000

3.3.1.3 อินเตอร์คอนเน็ค (Interconnect)

ความยืดหยุ่นของการใช้แอลซีเอมาทำเป็นอุปกรณ์ขึ้นอยู่กับการโปรแกรม ทรัพยากรต่างๆ ที่อยู่ภายในเข้าด้วยกันการที่จะควบคุมการเชื่อมต่อระหว่างจุดสองจุดภายในชิปเหมือนกับเกตอาเรย์ทั่วๆ ไป การเชื่อมต่อภายในแอลซีเอประกอบด้วยเน็ตเวริ์ค 2 ทิศทางคือทาง แถวและคอลัมน์ซึ่งวางอยู่ระหว่าง CLB programmable switch จะทำการเชื่อมต่ออินพุตและเอาต์พุตของไอโอบีและซีแอลบีที่จุดต่อร่วมระหว่างแถวกับคอลัมน์สามารถสลับสัญญาณจาก เส้นทางไปยังส่วนต่างๆ ดังรูปที่ 3.11



รูปที่ 3.11 แสดงเส้นทางการเชื่อมต่อระหว่างไอโอบีกับซีแอลบีของเอฟพีจีเอตระกูล XC4000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 คุณสมบัติโดยทั่วไปของเอฟพีจีเอตระกูล XC4000

1. เป็นอุปกรณ์รุ่นที่สามของเอฟพีจีเอ
   - มีฟลิปฟลอปเป็นจำนวนมาก
   - ในการผลิตฟังก์ชั่นของการทำงานมีความยืดหยุ่นสูง
   - มีจำนวนเกตภายในจำนวน 2,000 – 10,000 เกต
   - เพิ่มความสามารถพิเศษของเรจิสเตอร์และอินพุต/เอาต์พุต
   - มีค่าแฟนเอาต์ (fan-out) สูง
   - มีบัสภายใน 3 สถานะ
   - ทำงานกับสัญญาณทีทีแอลและซีมอส
   - มีออสซิเลเตอร์แอมปลิฟลายเออร์ภายใน
   - มีแรมภายในความเร็วสูง (< 25 Ns)
   - ใช้กับงานที่ต้องการความเร็วสูง (ใช้งานได้ที่ความถี่ 70/100/125 MHz)
   - มี Wide edge decoder
   - เส้นทางการเชื่อมต่อ (Interconnect line) เป็นแบบลำดับชั้น
   - มีการกระจายกำลังงานของสัญญาณต่ำ
2. มีสถาปัตยกรรมภายในที่ยืดหยุ่น
   - มีลอจิกบล็อคและไอโอบล็อคที่สามารถโปรแกรมได้
   - มีอินเตอร์คอนเน็คและ Wide decoder ที่โปรแกรมได้
3. ทำกระบวนการซับไมครอนชนิดซีมอสได้
   - มีลอจิกและอินเตอร์คอนเน็คที่มีความเร็วสูง
   - ใช้กำลังงานต่ำ
4. คุณลักษณะทาง System-Oriented
   - รองรับมาตราฐาน IEEE 1149.1 ในการทำ boundary-scan logic
   - สามารถโปรแกรมค่า output slew rate ได้
   - สามารถโปรแกรมให้อินพุตมีลักษณะพูลอัพ (Pull-up) หรือ พูลดาวน์ (Pull-down) รีจิสเตอร์ได้
   - ให้กระแสเอาต์พุตได้ตั้งแต่ 12-24 มิลลิแอมป์ (ขึ้นอยู่แต่ละรุ่น)
5. ทำการโหลดเอาแฟ้มข้อมูลประเภทไบนารี
   - ไม่จำกัดจำนวนครั้งในการโปรแกรมซ้ำ
   - มีโหมดในการโปรแกรมให้เลือก 6 โหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. มีโปรแกรมช่วยพัฒนาได้แก่ XACT Development System ที่ทำงานบนคอมพิวเตอร์รุ่นต่างๆ เช่น '486/Pentiums, NEC PC, Apollo, Sun-4, HP700

- สามารถติดต่อกับโปรแกรมอื่นๆ ได้ เช่น Viewlogic, Mentor Graphic และ OrCAD เป็นต้น
- มีโปรแกรมการวางและเชื่อมโยงอุปกรณ์ภายในแบบอัตโนมัติ (automatic place and routing) ที่ครบสมบูรณ์
- มี Interactive Design Editor ที่ใช้สำหรับการทำ optimization
- มี 288 มาโคร 34 ฮาร์ดมาโคร และ แรม/รอมคอมพายเลอร์

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	32X32	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

ตารางที่ 3.1 แสดงรายละเอียดของอุปกรณ์ภายในเอฟพีจีเอตระกูล XC4000

3.4 การโปรแกรมเอฟพีจีเอตระกูล XC4000

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 โหมดการโปรแกรม

เอฟพีจีเอในตระกูล XC4000 มีโหมดการโปรแกรม 6 โหมด ซึ่งแต่ละโหมดจะถูกกำหนดโดยจากบิตโหมด ได้แก่ บิต M0, M1 และ M2 โดยมีโหมดการโปแกรมดังตารางที่ 3.2

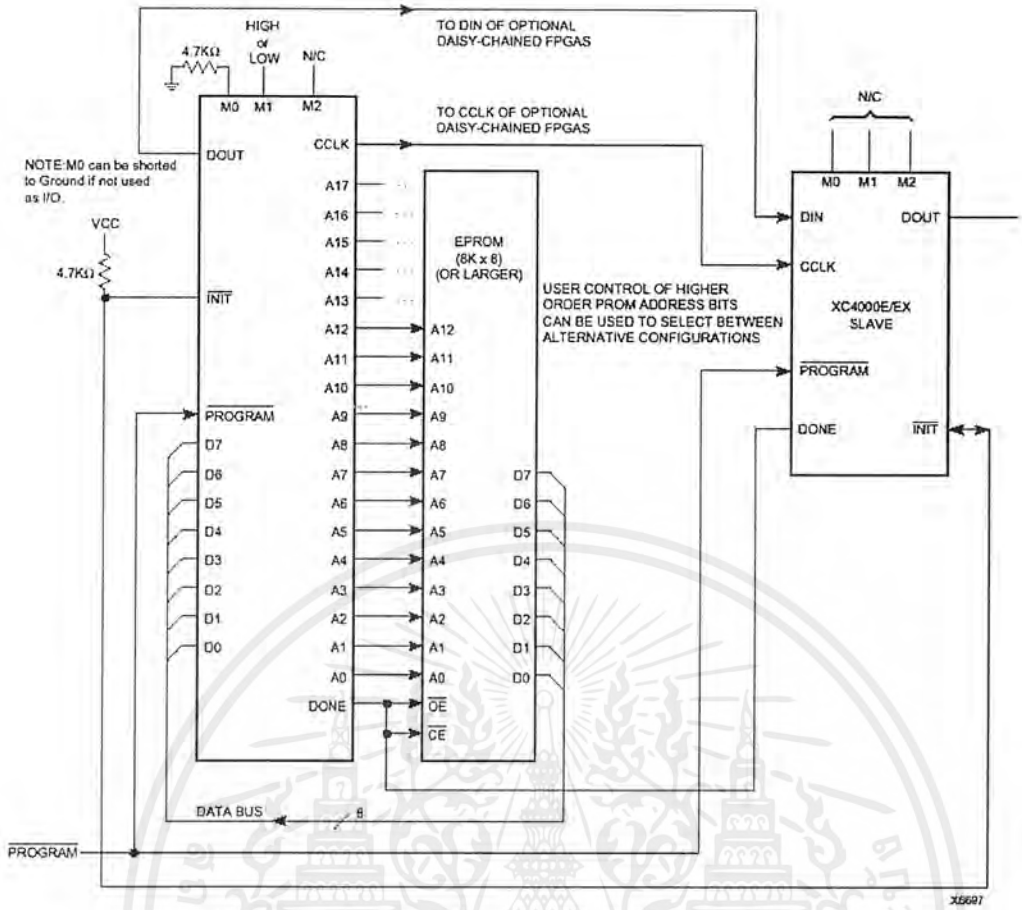
Mode	M2	M1	M0	Clock	Data
Master Serial	0	0	0	Output	Bit-serial
Slave Serial	1	1	1	Input	Bit-serial
Master parallel up	1	0	0	Output	Byte-Wide,00000
Master parallel down	1	1	0	Output	Byte-Wide,3FFFF
Peripheral Synch.	0	1	1	Input	Byte-Wide
Peripheral Asynch.	1	0	1	Output	Byte-Wide

ตารางที่ 3.2 แสดงรูปแบบของโหมดต่างๆ ในการโปรแกรมเอฟพีจีเอตระกูล XC4000

3.4.1.1 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 and master high mode โดยจะรับข้อมูลแบบขนานมาจากบิต D0-D7 โดยสัมพันธ์กับแอดเดรสที่กำหนดโดยแอลซีเอดังรูปที่ 3.12 จะเป็นการแสดงตัวอย่างการเชื่อมต่อในโหมดหลักแบบขนาน (Master Parallel mode) โดยเริ่มต้นที่แอดเดรส 0000 ไบต์ข้อมูล (data byte) จะถูกอ่านเข้ามาแบบขนานทุกๆ สัญญาณอาร์คล็อก (RCLK: Read Clock) และส่งเข้าไปภายในแบบอนุกรมโดยอาศัยสัญญาณนาฬิกาโครงแบบ (configuration clock)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



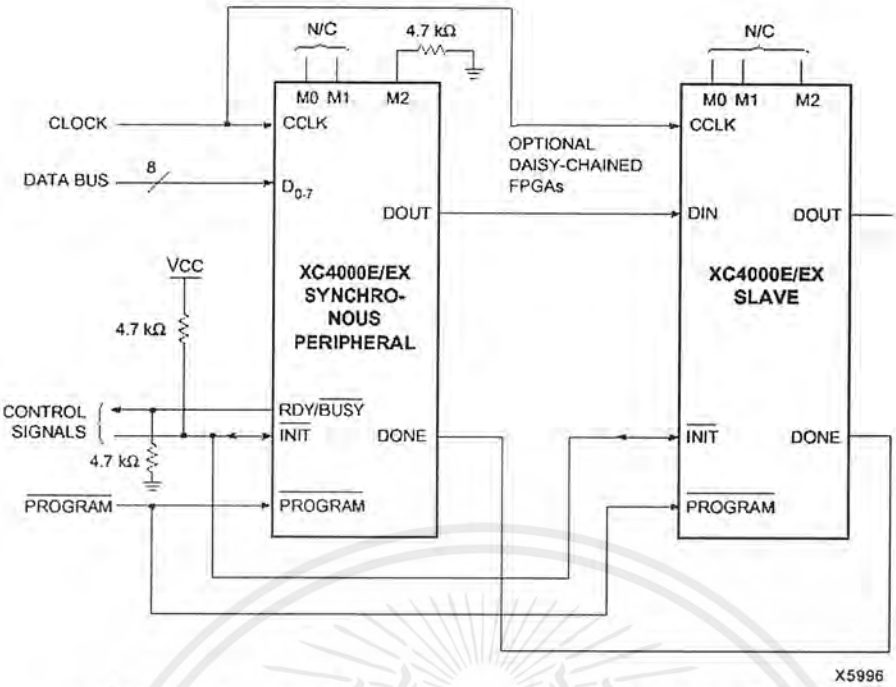
รูปที่ 3.12 แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดหลักแบบขนาน

3.4.1.2 เพอริเฟอรัลโหมด (Peripheral modes)

โหมดนี้จะรับข้อมูลเป็นไบต์จากบัส สถานะพร้อม (Ready) / กำลังทำงาน (Busy) จะมีอยู่เพื่อทำให้สัญญาณต่างๆ สื่อสารกันได้มีประสิทธิภาพมากขึ้น

ในอะซิงโครนัสโหมดนั้นออสซิเลเตอร์ภายในจะสร้างสัญญาณ CCLK เพื่อรับข้อมูลที่เป็นไบต์ให้อยู่ในรูปแบบที่เหมาะสม และในซิงโครนัสโหมดนั้นสัญญาณนาฬิกาภายนอกจะเป็นตัวควบคุมการรับข้อมูลที่เป็นไบต์ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

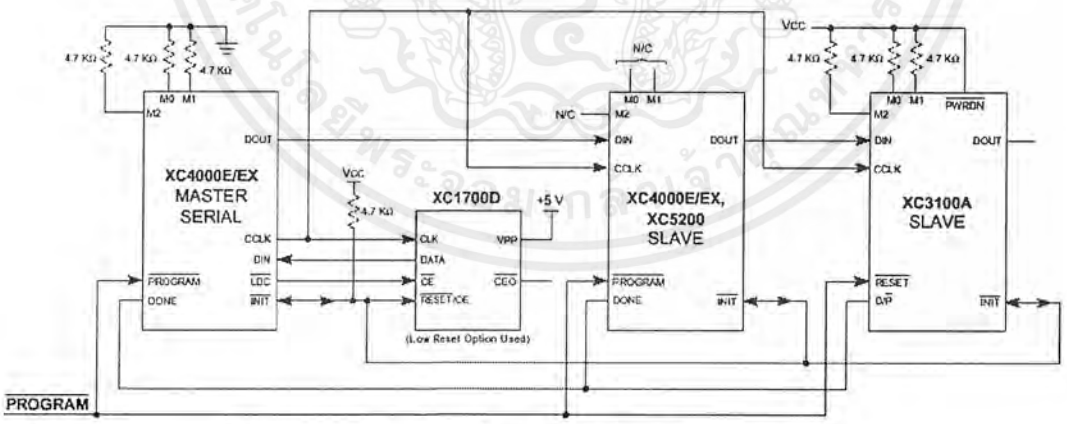


รูปที่ 3.13แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดเพอริเฟอรัลแบบอะซิงโครนัส

(Asynchronous Peripheral mode)

3.4.1.3 โหมดรองแบบอนุกรม (Slave Series modes)

ในโหมดนี้แอลซีเอจะรับข้อมูลโครงแบบที่เป็นอนุกรมในขาขึ้นของสัญญาณ CCLK และหลังจากรับข้อมูลมาแล้วจะส่งข้อมูลเพิ่มเติมออกไปด้วยและแอลซีเอก็จะถูกทำการซิงโครไนซ์ในขาลงถัดไปของสัญญาณ CCLK ดังรูปที่ 3.14



รูปที่ 3.14 แสดงผังวงจรการเชื่อมต่อเอฟพีจีเอในโหมดรองแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การใช้ความสามารถของแรมในเอฟพีจีเอตระกูล XC4000

แอลซีเอทำงานโดยใช้ตารางการค้นหา (Look-up table) ซึ่งจะทำการเก็บตารางที่ว่านี้ใน สแตติกแรม ซึ่งจะถูกเขียนในระหว่างการโปรแกรมโครงแบบลงบนแอลซีเอและจะถูกอ่านในการโอเปอเรชั่น ดังนั้นแรมภายในจึงควรถูกรวมไว้ในการออกแบบของผู้ใช้ด้วย

หน้าที่ของแรมในเอฟพีจีเอตระกูล XC4000 มีหน้าที่คล้ายแรมโดยทั่วๆ ไป เช่น เอฟไอเอฟโอ (FIFO: First In First Out) แอลไอเอฟโอ (LIFO: Last In First Out) รีจิสเตอร์ไฟล์รวมทั้งแอปพลิเคชั่นบางอย่าง อย่างเช่น รีจิสเตอร์เลื่อนข้อมูล (Shift register) แรมของเอฟพีจีเอตระกูล XC4000 มีความเร็วสูงเสมือนเอสแรม (SRAM) จึงไม่จำเป็นต้องคำนึงถึงเวลาหน่วงของการเชื่อมต่อ (Interconnection delay)

RAM Module	Equivalent Logic	XC4003	XC4005	XC4010
16 x 1	4 – input Function Generator (F or G)	200	392	800
32 x 1	Two – 4 – input Function Geranators and One 3 – input Function Geranator (F+G+H)	100	196	400

ตารางที่ 3.3 จำนวนของแรมภายในเอฟพีจีเอตระกูล XC4000

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
VCC	P11	I/O	P25
GND	P12	I/O	P26
I/O,PGCK1(A16)	P13	I/O	P27
I/O(A17)	P14	I/O	P28

ตารางที่ 3.4 รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

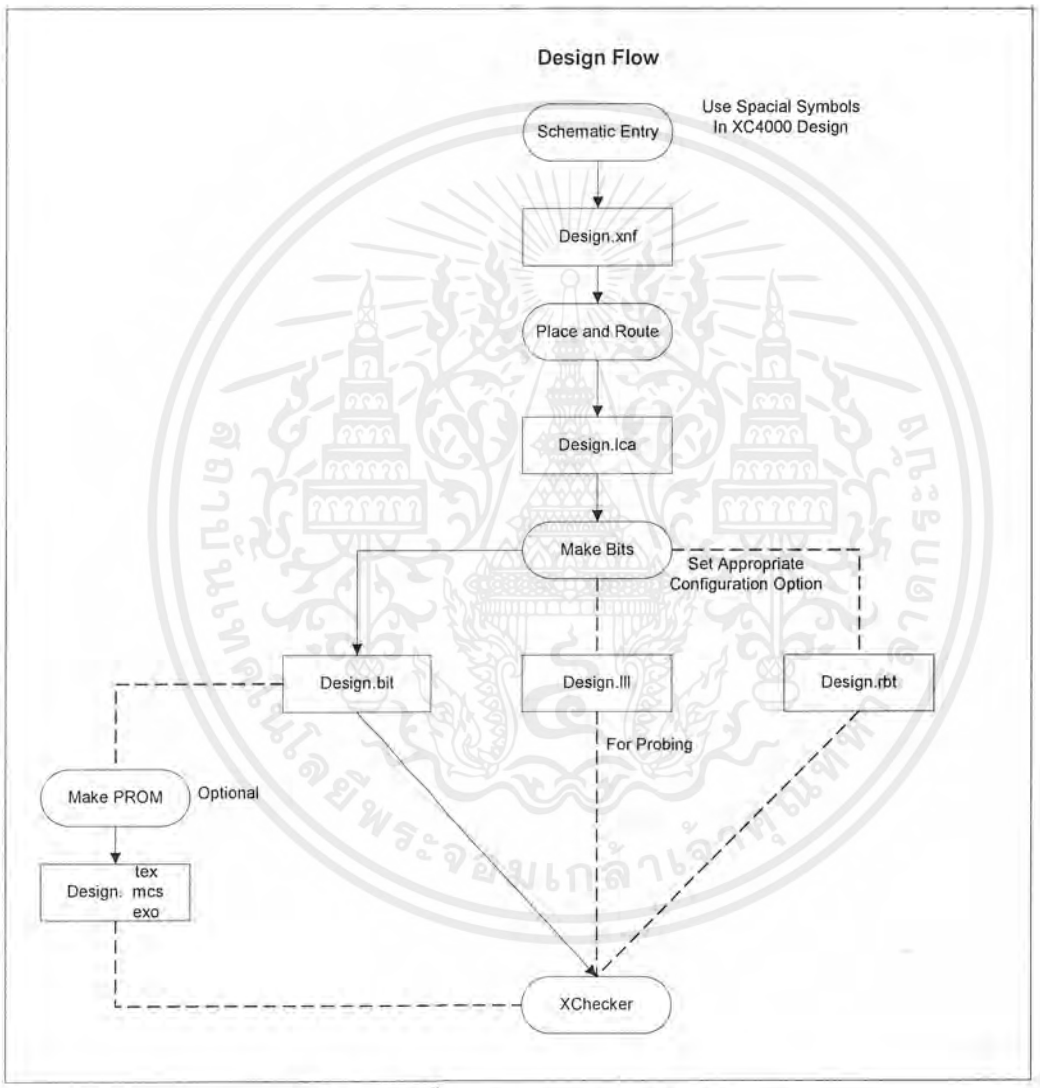
Pad Name	Pin Number	Pad Name	Pin Number
I/O,SCGK2	P29	I/O, PGCK3	P57
O(M1)	P30	I/O(D6)	P58
GND	P31	I/O(D5)	P59
I(M0)	P32	I/O($\overline{CS0}$)	P60
VCC	P33	I/O(D4)	P61
I(M2)	P34	I/O	P62
I/O,PGCK2	P35	VCC	P63
I/O($\overline{HDC}$)	P36	GND	P64
I/O(LDC)	P37	I/O(D3)	P65
I/O	P38	I/O(RS)	P66
I/O	P39	I/O(D2)	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
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 (CS1,A2)	P79
GND	P52	I/O(A3)	P80
DONE	P53	I/O(A4)	P81
VCC	P54	I/O(A5)	P82
$\overline{PROGRAM}$	P55	I/O(A6)	P83
I/O (D7)	P56	I/O(A7)	P84

ตารางที่ 3.4(ต่อ) รายละเอียดของขาอุปกรณ์เอฟพีจีเอเบอร์ XC4010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้ Software เพื่อการโปรแกรมลง FPGA

ในบทนี้เป็นการอธิบายส่วนของการใช้โปรแกรม Design Manager ในชุดโปรแกรมพัฒนา Xilinx Foundation series ของ XILINX ที่ใช้สำหรับแปลงเน็ตลิสต์ (Netlist) ของวงจรที่ได้จากการสังเคราะห์จากโปรแกรมพวก HDL Synthesis เช่น Leonardo Spectrum series,Synplify series,Synopsys FPGA Express series เป็นต้น เพื่อโปรแกรมลง FPGA ในการทำชิพต้นแบบ (Prototype) ต่อไป



รูปที่ 4.1 แสดงขั้นตอนการออกแบบวงจรโดยใช้ FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบจะเป็นดังรูปที่ 4.1 ซึ่งมีขั้นตอนดังนี้

1. Schematic Entity เป็นการออกแบบวงจรที่จะนำมาใช้
2. Design.xnf ออกแบบขาของวงจรว่าจะใช้กับขาใดของเอฟพีจีเอ
3. Place and Route
4. Design.lca
5. Make Bits
6. Design.bit
7. Xchecker

จากขั้นตอนที่ 3 ถึง 7 นั้นจะทำโดยโปรแกรมทั้งหมดโดยจะได้กล่าวต่อไป จากรูปที่ 4.1 ถ้าเราจะใช้ PROM เพื่อการดาวน์โหลดข้อมูลแทน Xchecker จะต้องใช้ ไฟล์ .bit มาใส่ลงไปใน PROM

4.1 การอิมพลิเมนท์ (Implementation)

ขั้นตอน Place & Route โดยใช้ซอฟต์แวร์ของ XILINX Foundation Series 2.1i เมื่อทำการสังเคราะห์วงจรจากโปรแกรม Leonardo Spectrum 1999g โดยผลของการสังเคราะห์จะถูกเขียนไว้ในรูปของเน็ตลิสต์แบบ XNF (Xilinx Netlist Format) เพื่อที่จะนำไปผ่านขั้นตอน Place & Route ด้วยโปรแกรม XILINX Foundation Series 2.1i โดยมีขั้นตอนดังต่อไปนี้

4.1.1 เรียก Design Manager ของ XILINX Foundation Series 2.1i โดยคลิ๊ก ไอคอน Design Manager

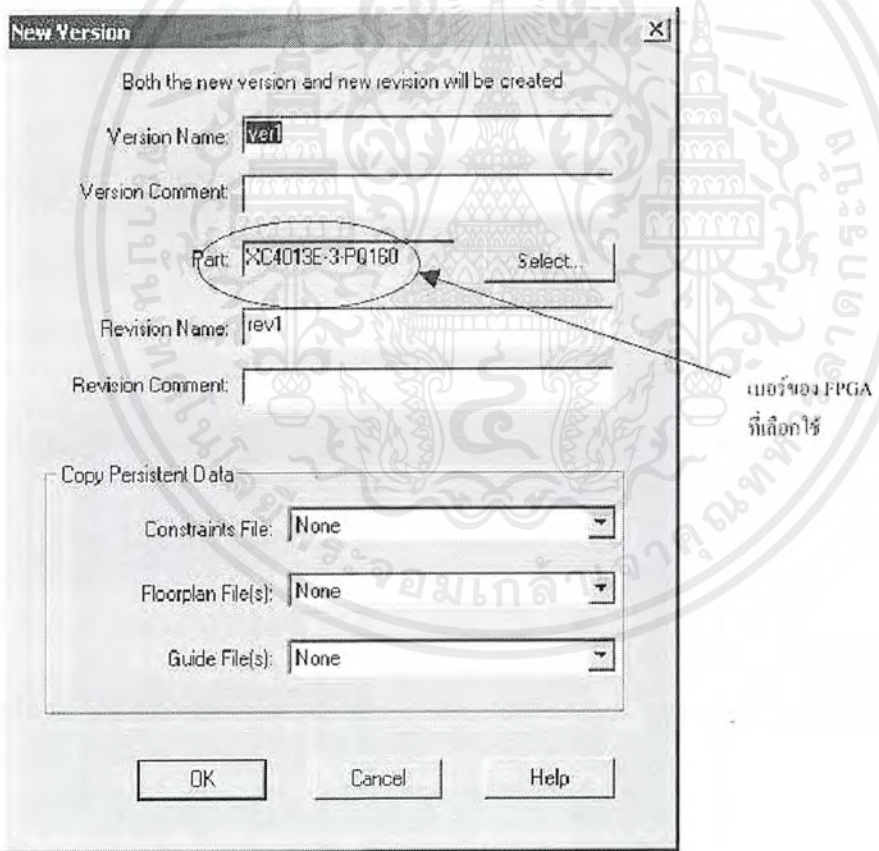
รูปที่ 4.2 หน้าต่างของ Design Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 สร้างโปรเจคใหม่สำหรับดีไซน์ File > New Project..

รูปที่ 4.3 หน้าต่างของ New Project เมื่อสร้างโปรเจคใหม่

ตัวอย่างนี้ไฟล์ XNF อยู่ในไดทรอรี่ C:\Project\Mydesign.xnf

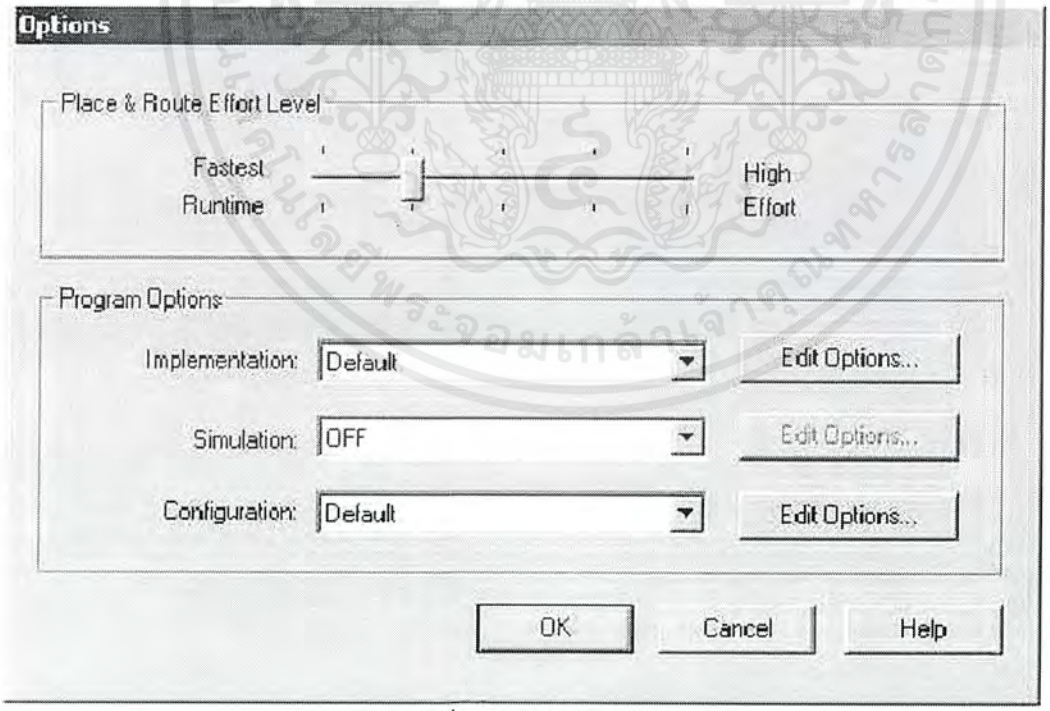


รูปที่ 4.4 หน้าต่างของการกำหนด Version และเบอร์ของ FPGA ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

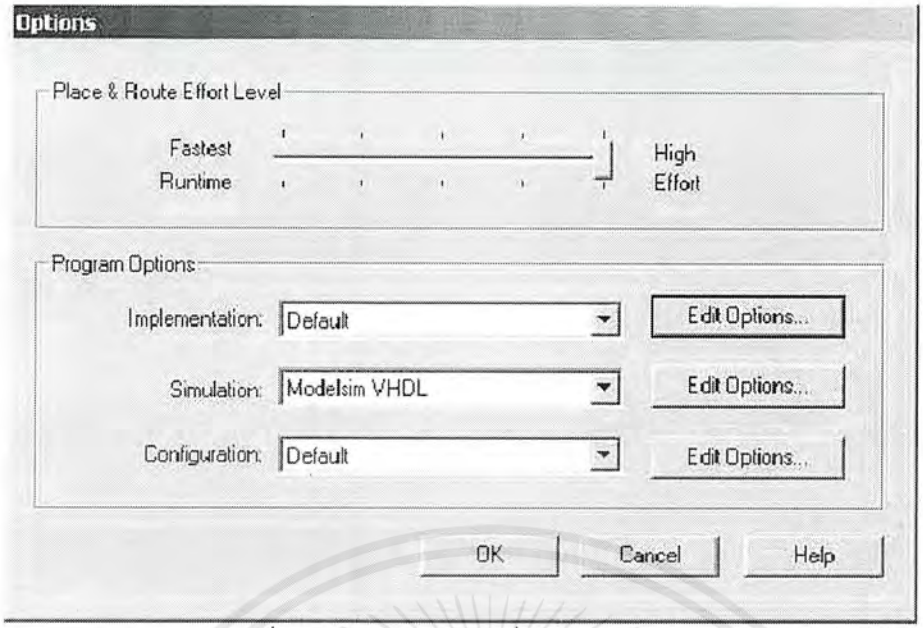
รูปที่ 4.5 หน้าต่าง Project ที่ทำการ New Version Complete

4.1.3 คลิ๊กเลือก Set Option



รูปที่ 4.6 หน้าต่าง Set Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



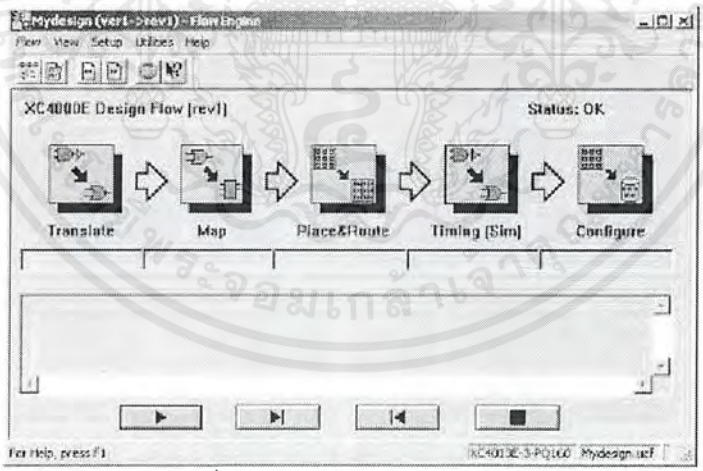
รูปที่ 4. 7 หน้าต่าง Set Option ที่ถูกกำหนดค่าต่างๆ

โดยทำการเลือก Place & Route Effort Level > High Effort

Program Options > Simulation > Modelsim VHDL

เพื่อนำไปจำลองการทำงานในการตรวจสอบฐานเวลาจากการทำ Place & Route ต่อไป

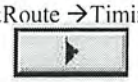
4.1.4 คลิ๊กเลือก Flow Engine



รูปที่ 4.8 หน้าต่าง Flow Engine

Flow Engine จะประกอบด้วยขั้นตอนย่อยดังต่อไปนี้

Translate →Map → Place&Route →Timing(Sim) → Configure และจะดำเนินการไปตามลำดับ

โดยคลิ๊กเริ่มทำงานที่ปุ่มรัน  โปรแกรมก็จะเริ่มทำงานตามขั้นตอนข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4.1 Translate & Map

Translate คือขั้นตอนที่โปรแกรมอ่านเน็ตนิสต์ไฟล์ แบบ XNF ของดีไซน์ตามที่ได้กำหนดไว้แล้ว แปลงให้อยู่ในรูปแบบข้อมูลของ XILINX เอง (ไฟล์.NGO) โดยใช้คำสั่ง NGDBuild ซึ่งจะแปลงให้เป็นเน็ตลิสต์แบบ .NGO (Native Generic Database) แล้วผ่านต่อไปยังขั้นตอน Map ที่เกี่ยวข้องกับการจัดการฮาร์ดแวร์ภายในของอุปกรณ์เป้าหมาย (Resource Allocation) รวมถึงการจัดแบ่งลอจิกในลอจิกบล็อค อาทิ เช่น CLB (Configuration Logic Block) และ IOB (Input Output Block) และผลการทำงานที่ได้จะถูกเก็บบันทึกไว้ในไฟล์ที่ชื่อว่า map.ncd เพื่อนำไปใช้ในขั้นตอน Place & Route ต่อไปหลังจากผ่านขั้นตอน Translate และ Map แล้วอาจจะมีการทำ Timing Analysis เพื่อทราบรายละเอียดเกี่ยวกับความล่าช้า (Delay time) ของสัญญาณอย่างคร่าวๆ ซึ่งเป็นค่าประมาณเท่านั้นความล่าช้าของสัญญาณอาจแบ่งออกได้เป็นสองจำพวกคือ ความล่าช้าที่เกิดจากการแบ่งลอจิกบล็อค (Block Delay) และความล่าช้าที่เกิดจากการเชื่อมโยงเส้นทางสัญญาณระหว่างบล็อคเหล่านั้น (Routing Delay) ในกรณีที่ได้มีการกำหนดข้อบังคับในเรื่องระยะเวลาให้แก่เส้นทางสัญญาณภายในวงจร ค่าความล่าช้าของสัญญาณที่คำนวณได้หลังจากได้ทำขั้นตอน Map (Post-Map delay) ไม่ควรเกิน 50 เปอร์เซนต์ของความล่าช้ารวมอาจจะเกินค่าที่ต้องการก็ได้และถ้าค่าความล่าช้าจริงมากกว่าค่าที่กำหนดไว้ก็มีทางเลือกสองทางคือ เปลี่ยนแปลงวงจรใหม่เพื่อให้ได้เส้นทางสัญญาณวิกฤต(Critical path) ที่สั้นลง หรืออีกทางหนึ่งก็คือ ยอมรับค่าความล่าช้าของสัญญาณที่ได้จริง

4.1.4.2 Place & Route

ขั้นตอนการทำ Place & Route เริ่มจากการอ่านข้อมูลของวงจรไฟล์ map.ncd และไฟล์ที่ลงท้ายด้วย .PCF (Physical Constraint File) ซึ่งภายในมีข้อบังคับต่างๆ ที่โปรแกรม PAR จะต้องคำนึงถึงเมื่อทำขั้นตอน Place & Route งานขั้นแรก คือ การวางลอจิกบล็อคลงใน FPGA ก่อนซึ่งเป็นหน้าที่ของ Placer และถ้ามีการกำหนดข้อบังคับในเรื่องเวลาแล้วโปรแกรม Placer ก็จะพยายามวางบล็อคเหล่านั้นให้อยู่ใกล้ๆ กันเพื่อลดค่าความล่าช้าของสัญญาณระหว่างบล็อคต่างๆ และผลการกระทำในแต่ละครั้งจะมีแต้มคะแนน (Placer Score) และครั้งที่มีแต้มคะแนนน้อยที่สุดจะถือว่าดีที่สุด และจบท้ายด้วยการเชื่อมโยงบล็อคต่างๆ ที่ถูกวางลงในตำแหน่งที่เหมาะสมแล้วโดย Router ซึ่งขั้นตอนนี้จะมีหลายครั้งเพื่อพยายามเชื่อมโยงเส้นทางภายในของวงจรให้ได้ตามข้อกำหนด

4.1.4.3 Timing Analysis

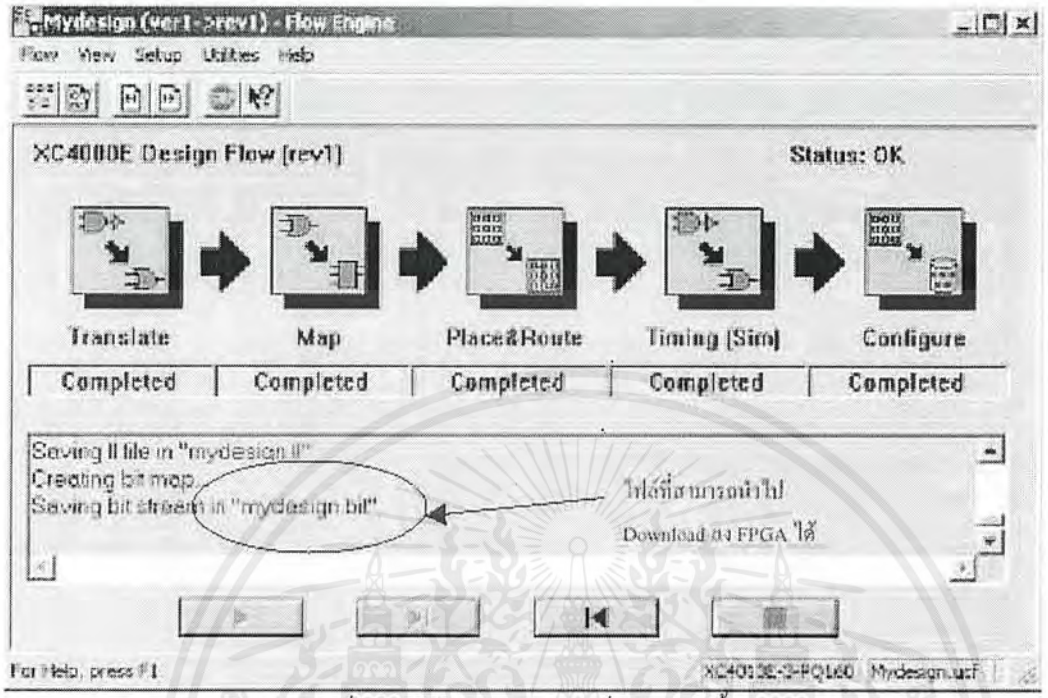
ในขั้นตอนนี้โปรแกรมจะทำการวิเคราะห์วงจรเพื่อหาค่าความล่าช้าของเส้นทางสัญญาณต่างๆ หลังจากที่ผ่านขั้นตอน Place & Route แล้วซึ่งค่าความล่าช้าที่คำนวณได้นี้ค่อนข้างแม่นยำและถูกต้อง ใกล้เคียงกับค่าความล่าช้าจริงเมื่อวงจรทำงานภายใน FPGA

4.1.4.4 Configure

เมื่อผ่านขั้นตอน Place & Route แล้วถือว่าการออกแบบสร้างดีไซน์สำหรับ FPGA ได้เสร็จสิ้นลงแล้ว และเหลือเพียงขั้นตอน Configure ซึ่งใช้ในการแปลงข้อมูลของวงจรที่ได้ให้อยู่ในรูปแบบของบิต สตรีม (Bit Stream) เพื่อนำไป Download ลงใน FPGA ต่อไปโดยข้อมูลนี้จะถูกบันทึกลงในไฟล์ที่มีนามสกุล .BIT ใน

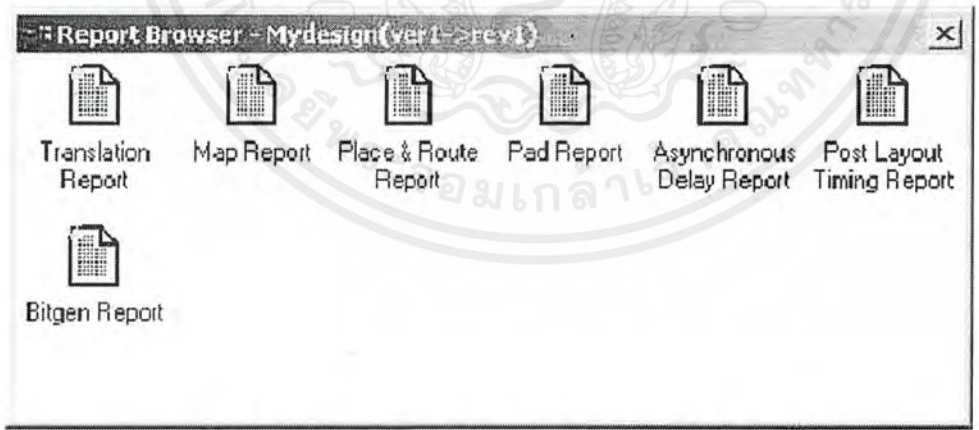
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการทำขั้นตอนแต่ละขั้นเริ่มตั้งแต่ Translate ไปจนถึง Configure ผู้ใช้สามารถหยุดการทำงานของ Flow Engine อย่างชั่วคราวได้ และสามารถสั่งให้ทำต่อไปได้ถ้าต้องการ โดยกดปุ่ม stop หรือ run ตามลำดับ



รูปที่ 4.9 หน้าต่าง Flow Engine ที่ทำครบทุกขั้นตอน

4.1.5. ผลการทำงานของขั้นตอนต่างๆ ใน Flow Engine สามารถตรวจสอบได้จาก คลิ๊กปุ่ม Browse Report โดยรายงานผลจะถูกแบ่งออกตามขั้นตอนดังรูปที่ 4.10

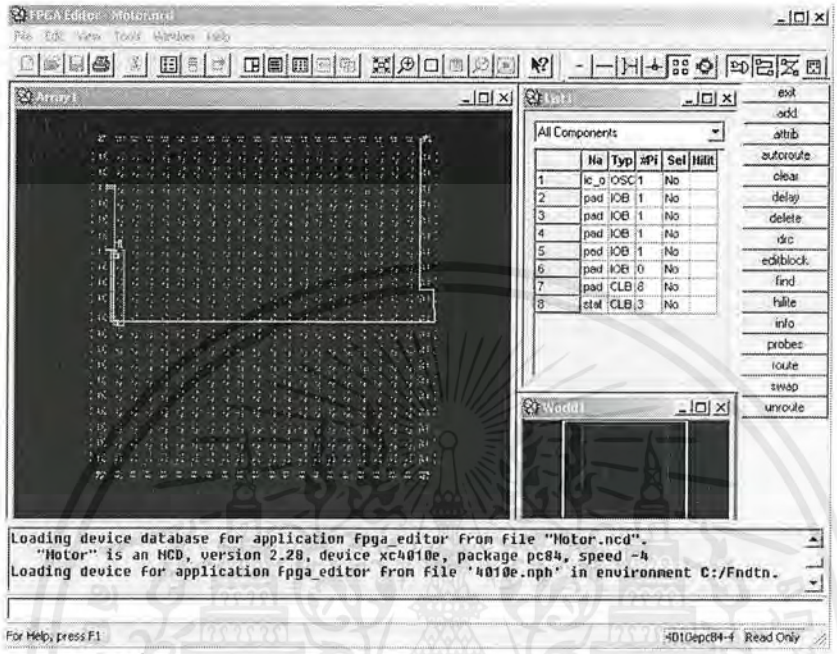


รูปที่ 4.10 หน้าต่างสำหรับเลือกแสดงผลการทำงานในขั้นตอนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้ FPGA Editor เพื่อตรวจสอบวงจรภายใน FPGA

หลังจากที่ได้ขั้นตอนต่างๆ ภายใน Flow Engine แล้ว เราสามารถเรียกใช้งานโปรแกรมย่อยอีกตัวหนึ่งก็คือ FPGA Editor โดยคลิ๊กปุ่ม เพื่อใช้ดู Layout ของวงจรภายใน FPGA โดยจะแสดงให้เห็นตำแหน่งของลอจิกบล็อคต่างๆ พร้อมเส้นทางของสัญญาณที่เชื่อมโยงภายในวงจร มีลักษณะดังรูปที่ 4.11



รูปที่ 4.11 หน้าต่าง FPGA Editor และ Layout ของวงจรภายใน FPGA

4.3 การจำลองฐานเวลาของวงจร (Simulation) หลังจากทำขั้นตอน Place & Route

หลังจากที่ผ่านขั้นตอน Place & Route แล้ว จะต้องมีการตรวจสอบความถูกต้องของวงจรที่ได้อีกครั้ง เพื่อที่จะแน่ใจว่าวงจรที่เราจะนำไปใช้งานกับอุปกรณ์กับ FPGA นั้นทำงานได้ถูกต้องทั้งในเรื่องของฟังก์ชั่นและเวลา เราเรียกขั้นตอนนี้ว่า Post-Layout Functional & Timing simulation เนื่องจากเราได้กำหนดตัวเลือกให้โปรแกรมสร้างเน็ตลิสต์ไฟล์แบบ VHDL สำหรับการจำลองฐานเวลาของวงจรไว้แล้ว ดังนั้นเราก็แค่นำสำเนาไฟล์ดังกล่าวไปใส่ไว้ในไดเรคทรอรี่ที่เราต้องการ ซึ่งมีอยู่สองไฟล์ด้วยกันคือ time_sim.vhd (VHDL) และ time_sim.sdf (SDF : Standard Delay Format)และสามารถนำไปการจำลองฐานเวลาของวงจรโดยใช้ Modelsim ได้แต่ก่อนที่จะทำการจำลองฐานเวลาของวงจรกับเน็ตลิสต์ที่สร้างจาก XILINX ได้ จะต้องมีการคอมไพล์สามไฟล์ตามลำดับคือ

<XILINX>\vhdl\src\simprims\simprim_Vcomponent.vhd

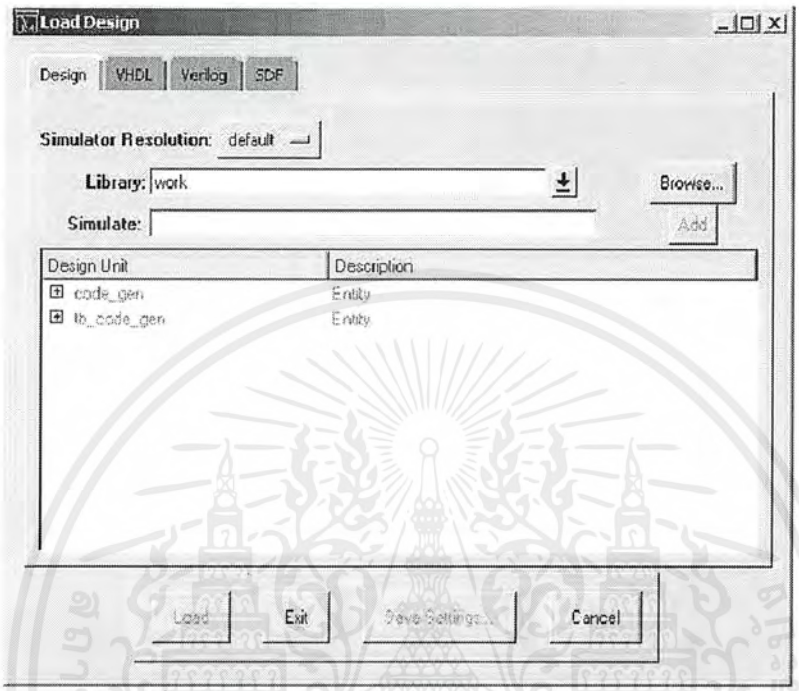
<XILINX>\vhdl\src\simprims\simprim_Vpackage.vhd

<XILINX>\vhdl\src\simprims\simprim_VITAL.vhd

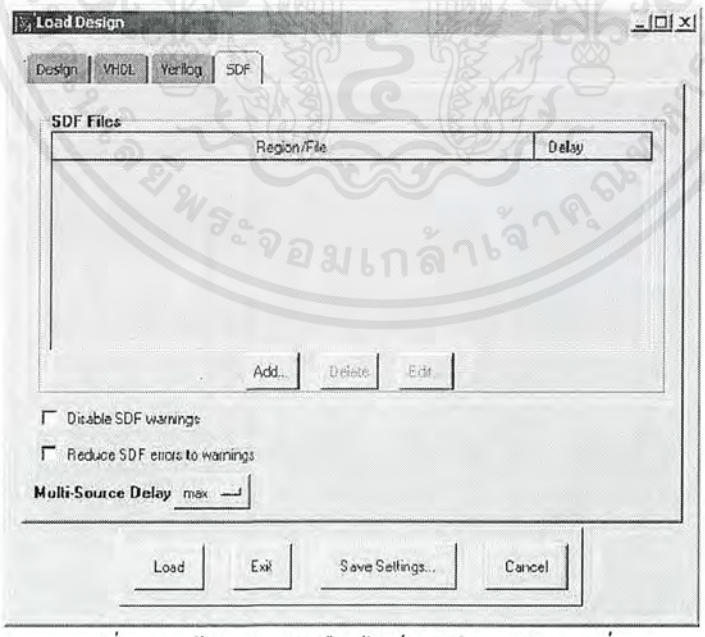
และจะต้องคอมไฟล์เก็บไว้ในไลบรารี่ที่มีชื่อ simprim

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้สร้างไลบรารี่ simprim และคอมไพล์ไฟล์ทั้งสามเก็บไว้ในไลบรารี่ดังกล่าวภายใน Project Directory และจากนั้นทำการคอมไพล์ไฟล์ time_sim.vhd เก็บลงในไลบรารี่ work และทำการ recompile Testbench ของดีไซน์ในครั้งที่ทำ Behavioral Simulation ให้ทำตามขั้นตอนเหมือนที่เคยทำ เพียงแต่ในคราวนี้จะต้องมีการกำหนดไฟล์ SDF ที่ชื่อ time_sim.sdf เพิ่มเข้าไป

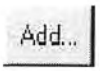


รูปที่ 4.12 หน้าต่าง Load Design ภายในโปรแกรม Modelsim

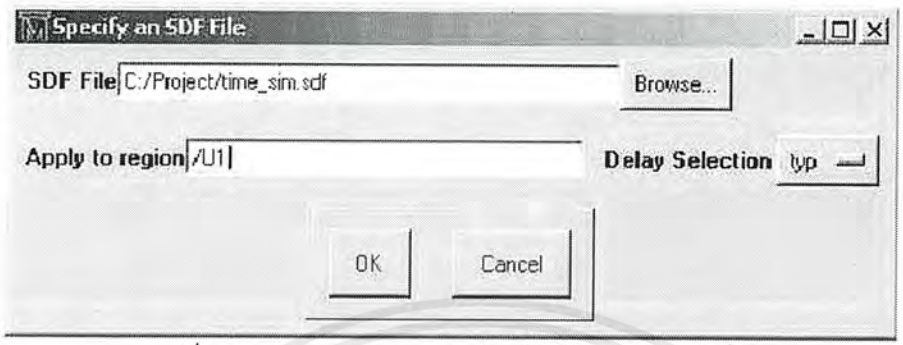


รูปที่ 4.13 หน้าต่างสำหรับเลือกไฟล์ SDF ในการซิมมูลเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

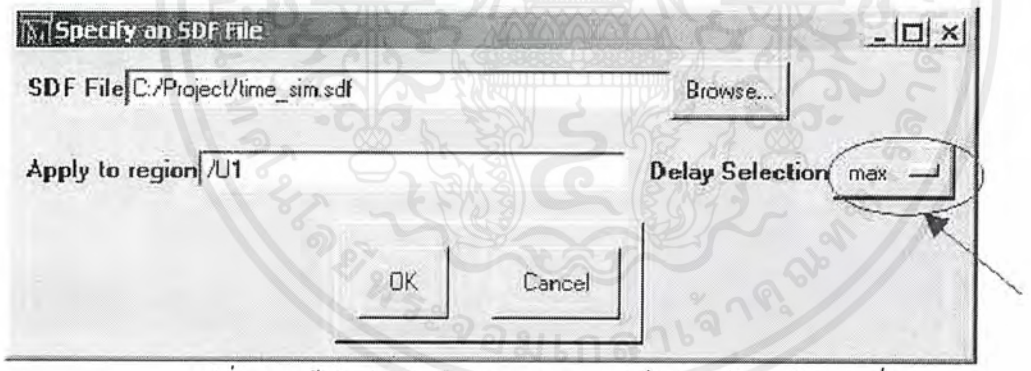
โดยเริ่มจากคลิ๊กปุ่ม Add 

เพื่อเลือกไฟล์ SDF เมื่อคลิ๊กปุ่ม Add จะปรากฏหน้าต่าง ดังรูปที่ 4.14 ในตัวอย่างเลือกไฟล์ C:\Project\time_sim.sdf และ U1 เป็นชื่อ instance เมื่อเรียกใช้ Mydesign โมดูลที่ออกแบบใน Test bench (TB_Mydesign.vhd)



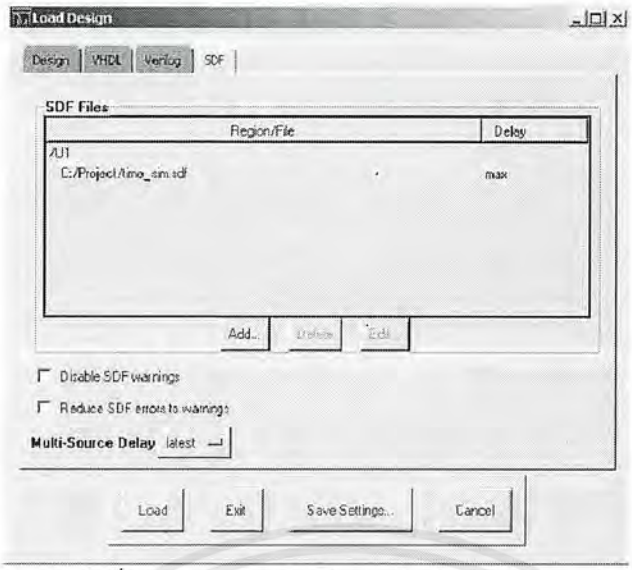
รูปที่ 4.14 หน้าต่างตัวอย่างการเลือกไฟล์ SDF ในการซิมมูลเลชั่น

เลือก Delay Selection เป็น MAX ในการจำลองฐานเวลาของวงจร



รูปที่ 4.15 หน้าต่างสำหรับเลือก Delay Selection เป็น MAX ในการซิมมูลเลชั่น

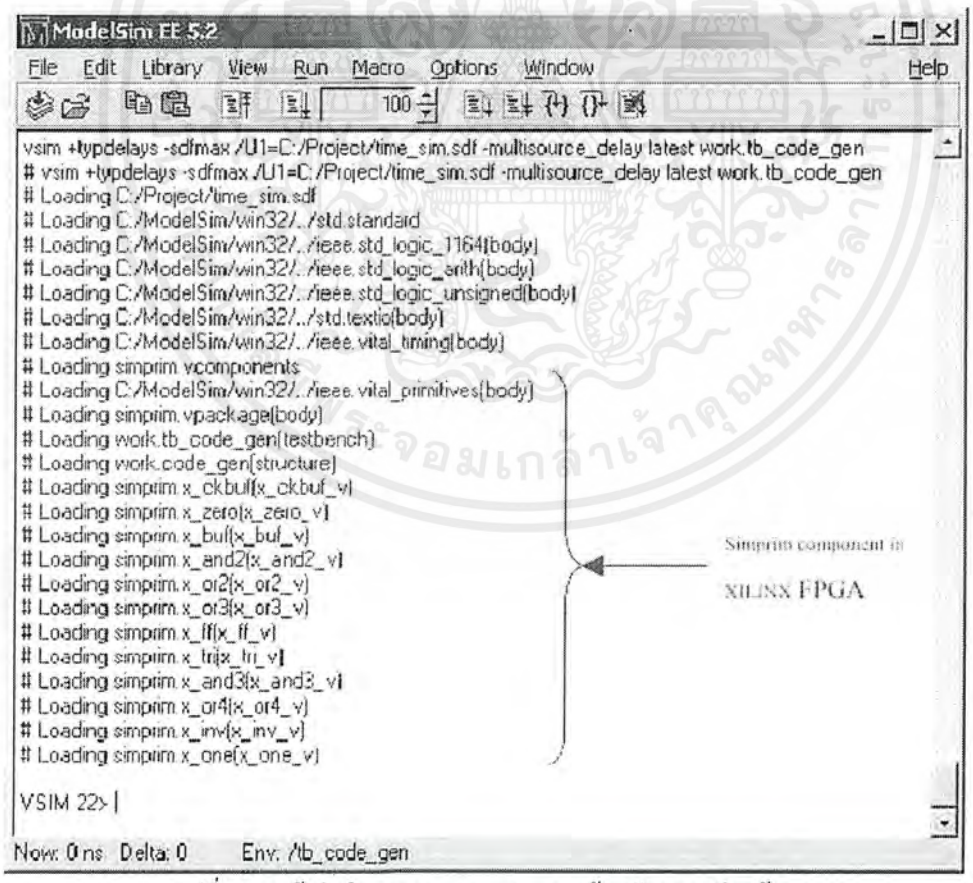
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าต่างที่แสดงรายละเอียดสำหรับการเลือกไฟล์ SDF ในการซิมมูลเลชั่น

เมื่อครบทุกขั้นตอนให้คลิ๊กปุ่ม Load

เพื่อการ Load Test bench ขึ้นมาซิมมูลเลชั่นในระดับ Timing ต่อไป



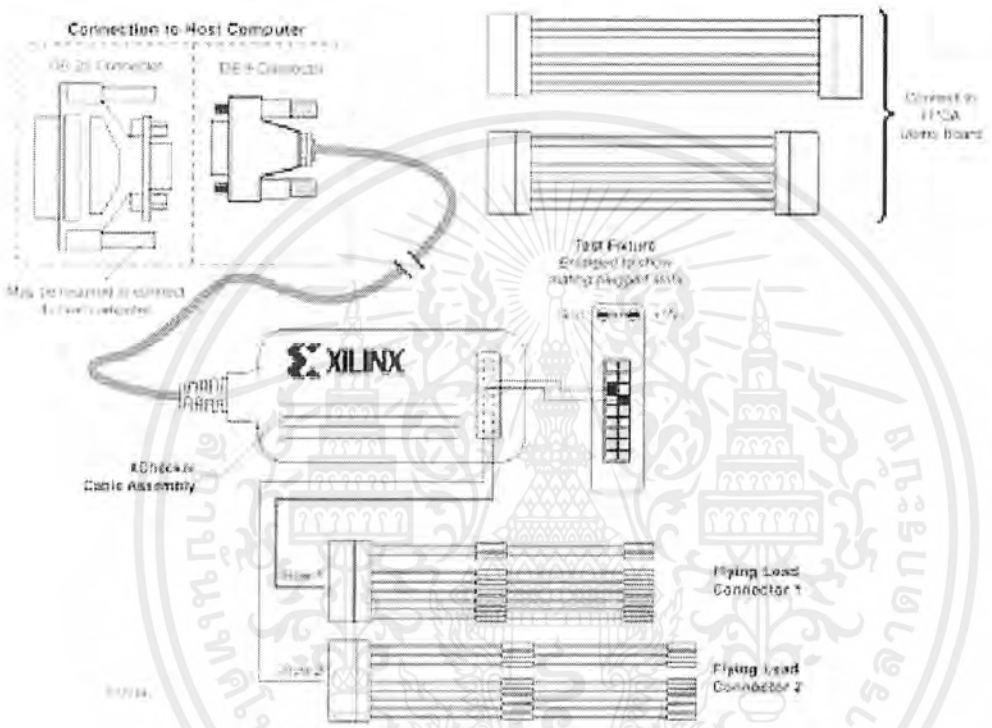
รูปที่ 4.17 หน้าต่างในการ Load Test bench พร้อมกับ SDF เรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นการเปรียบเทียบดูสัญญาณ (waveform) ว่าตรงกับที่จำลองการทำงานในระดับฟังก์ชั่น (Functional) หรือไม่ เพื่อเป็นการยืนยันว่าการทำงานนั้นยังถูกต้องจากการที่นำไปสังเคราะห์

4.4 การทดสอบวงจรกับฮาร์ดแวร์บนบอร์ดทดลองที่ใช้เอฟพีจีเอ

เมื่อผ่านขั้นตอนต่างๆ จาก Flow Engine แล้วผู้ใช้สามารถนำวงจรที่ได้สร้างขึ้นสำหรับ FPGA ไปทดสอบกับฮาร์ดแวร์จริงได้โดยการ Download ไฟล์ โดยผ่านโปรแกรม Hardware Debugger โดยคลิ๊กปุ่ม ในโปรแกรม Design Manager เพื่อ Download ข้อมูลลงใน FPGA บนบอร์ดทดสอบ

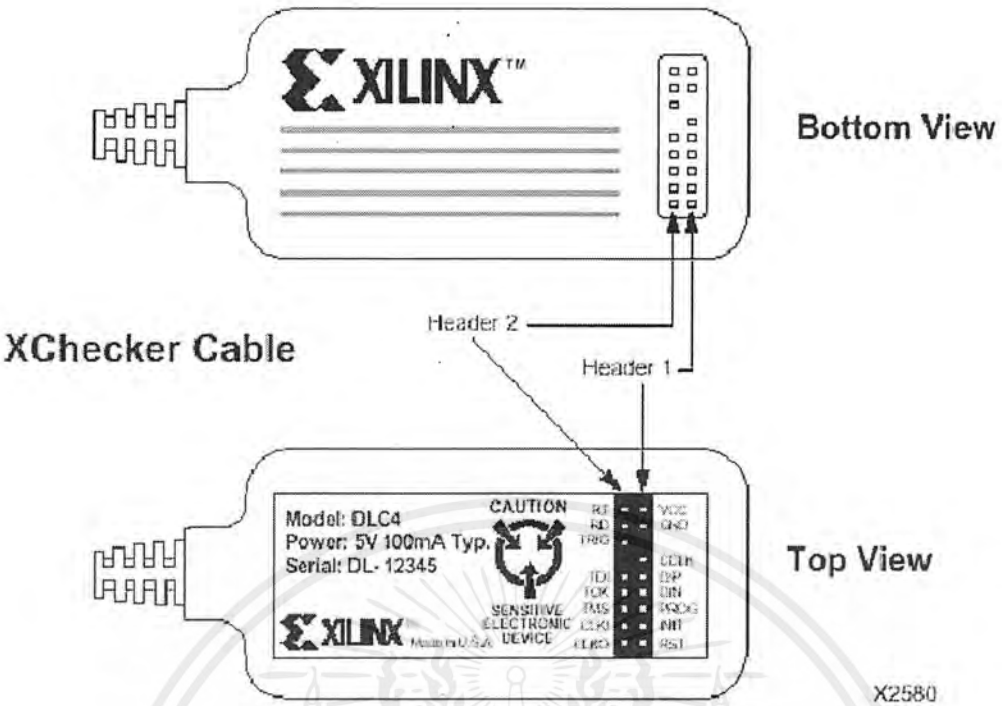


รูปที่ 4.18 สาย X-Checker พร้อมแถบสายต่อไปยังบอร์ดทดสอบ

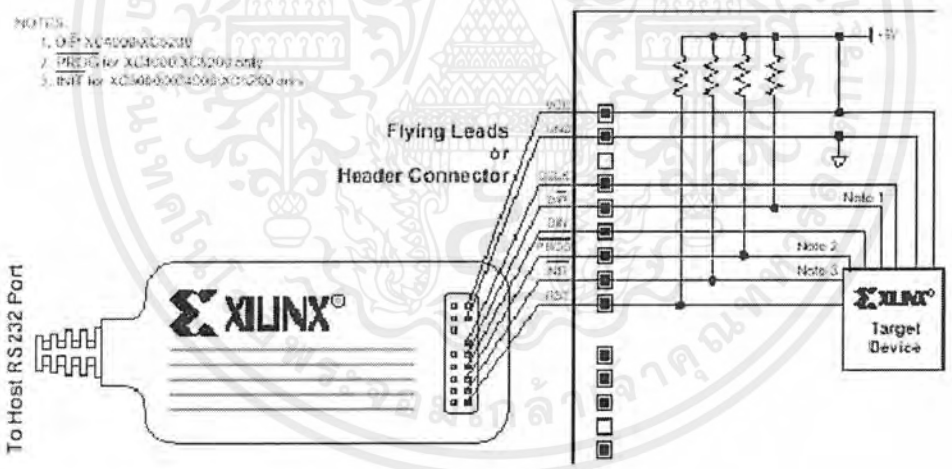
ส่วนประกอบของสาย X-Checker จะประกอบด้วยขาต่างๆเพื่อใช้ต่อกับแถบสายต่อไปยังบอร์ดทดสอบมีดังนี้

1. VCC = ไฟเลี้ยงเอฟพีจีเอ 5 หรือ 3.3 โวลล์
2. GND = กราวน์ของบอร์ดเอฟพีจีเอ
3. CCLK = ต่อกับขา CCLK ของเอฟพีจีเอ
4. D/P = ต่อกับขา DONE ของเอฟพีจีเอ
5. DIN= ต่อกับขา DIN ของเอฟพีจีเอ
6. PROG= ต่อกับขา PROGRAM ของเอฟพีจีเอ
7. INIT= ต่อกับขา INIT ของเอฟพีจีเอ
8. RST= ไม่ได้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

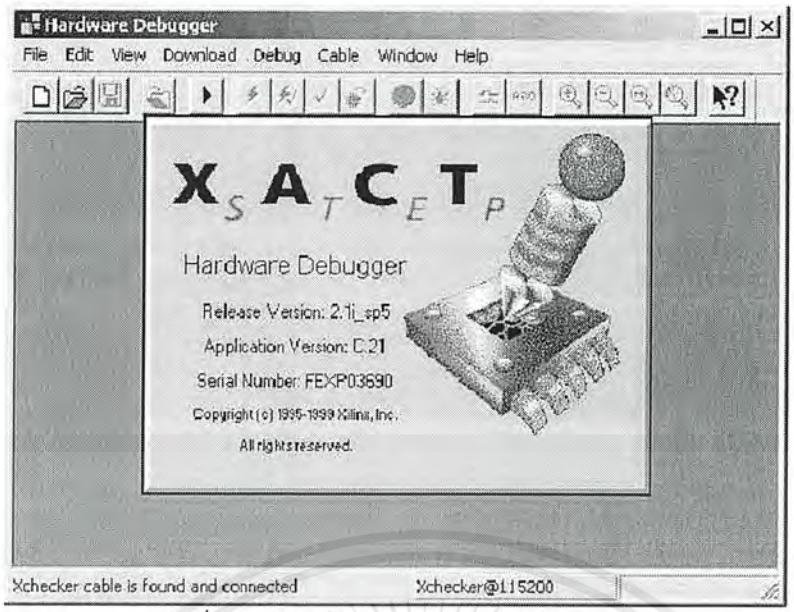


รูปที่ 4.19 ส่วนหัวของสาย X-Checker

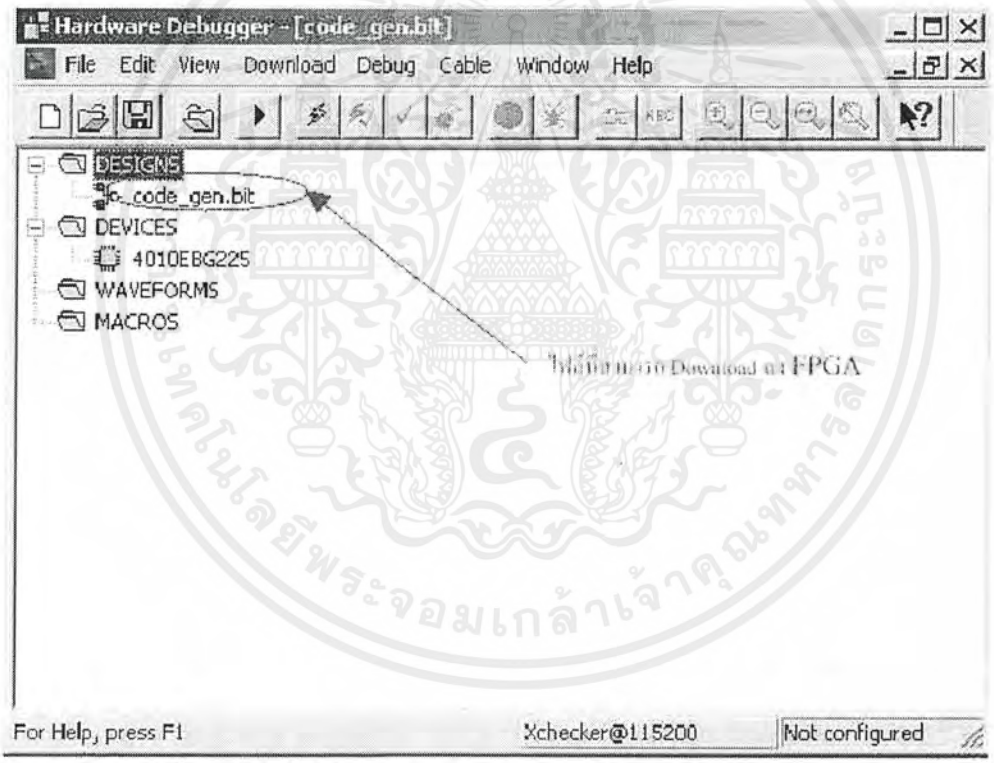


รูปที่ 4.20 การเชื่อมระหว่าง X-Checker กับบอร์ดทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



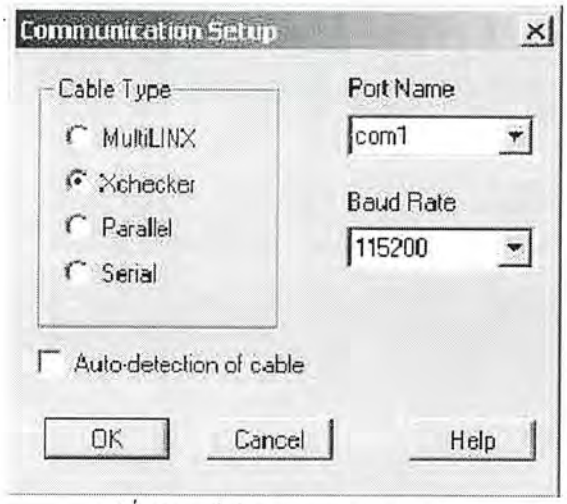
รูปที่ 4.21 หน้าต่างโปรแกรม Hardware Debugger



รูปที่ 4.22 หน้าต่างโปรแกรม Hardware Debugger ในการเปิด Design project

เลือก Cable > Communications... เพื่อการ Setup การเชื่อมต่อของ X-Checker cable กับเครื่องคอมพิวเตอร์ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 หน้าต่าง Communication Setup

เมื่อเชื่อมทุกอย่างเรียบร้อย จากนั้นทดลอง Download วงจรลงสู่ FPGA โดยคลิ๊กปุ่ม

รูปที่ 4.24 หน้าต่างแสดงการ Download วงจรลงสู่ FPGA

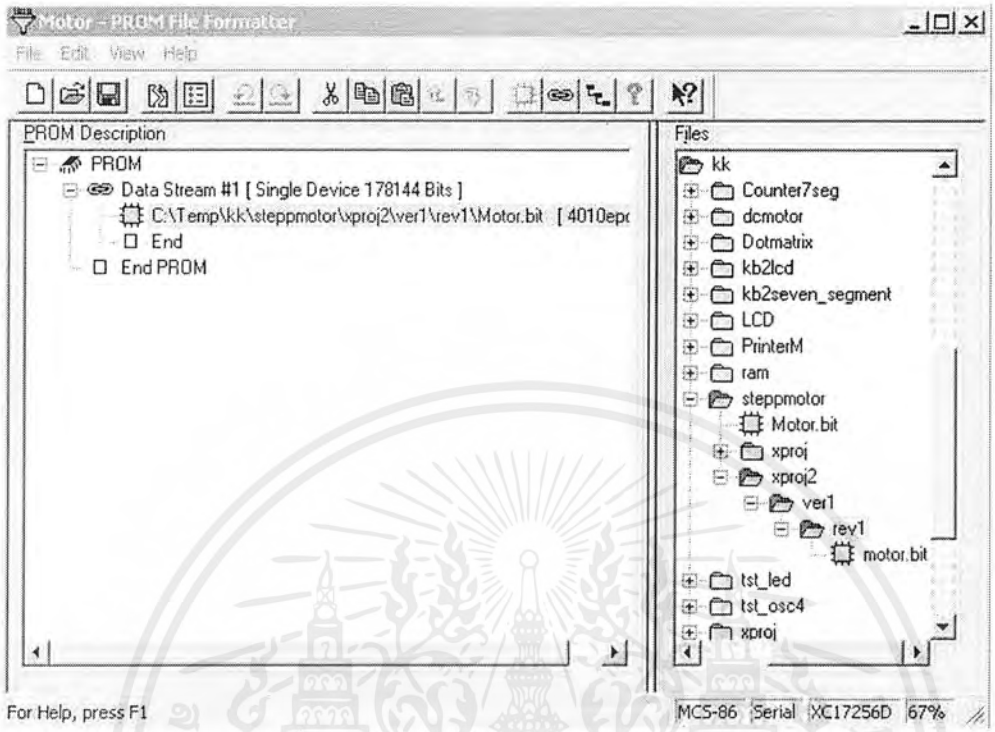
รูปที่ 4.25 หน้าต่างแสดงการ Download วงจรเสร็จสมบูรณ์

4.5 การคอนฟิกก์พรอม(Configuration PROM)

เป็นขั้นตอนการแปลง BIT ไฟล์ที่ใช้ในการ Download ผ่าน X-Checker เพื่อใช้เขียนลงสู่ Serial PROM ใช้การ Configuration แบบ Master Series mode แทนการใช้ X-Checker cable ซึ่งเป็นแบบ Slave series mode ได้โดย

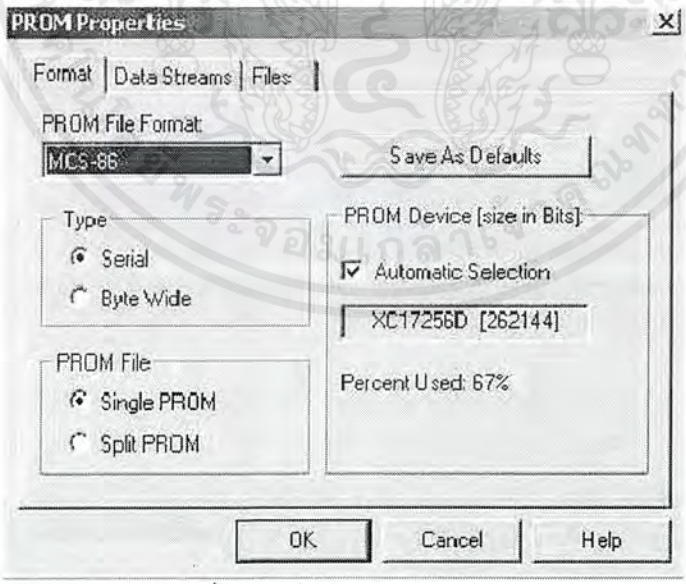
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 คลิ๊กปุ่ม PROM File Fomatter ใน Design Manager



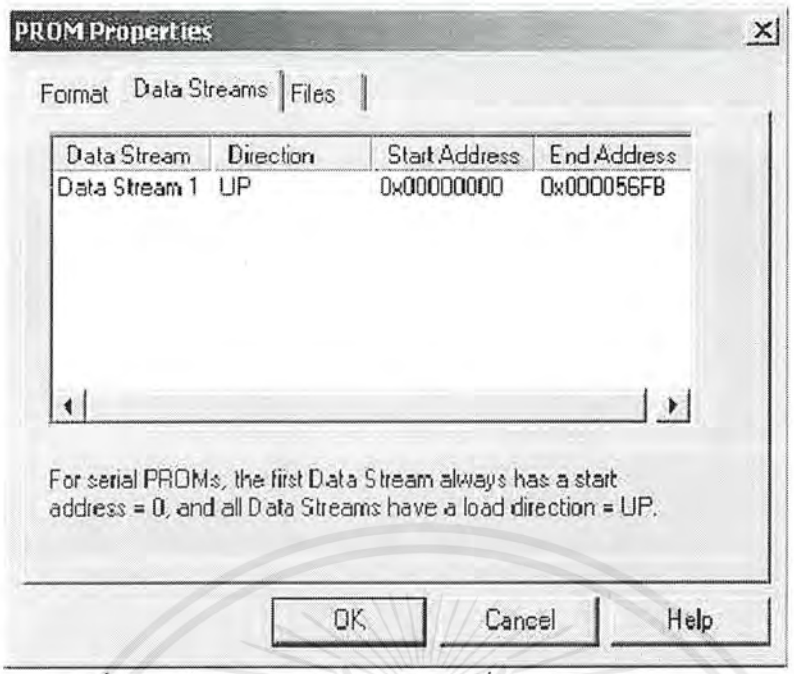
รูปที่ 4.26 หน้าต่างโปรแกรม PROM File Formatter

4.5.2 การเลือก format ของ PROM



รูปที่ 4.27 หน้าต่าง PROM Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 หน้าต่าง PROM Properties ในการเริ่มเขียนแบบ UP Direction

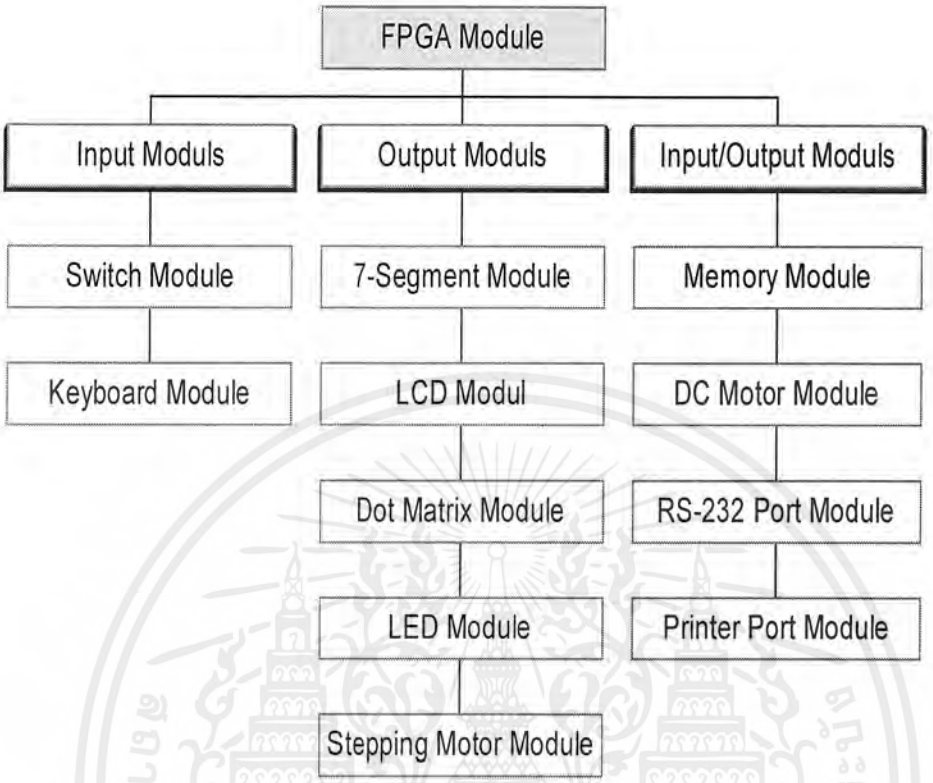
4.6 สรุป

ขั้นตอนการออกแบบวงจรตั้งแต่เริ่มต้นมีดังนี้

1. ต้องมีการออกแบบวงจรก่อนอาจจะเป็น Schematic หรือ วีเอชดีแอล
2. จากนั้นนำมา Synthesis เพื่อให้ได้ .xnf
3. นำไฟล์ .xnf มาแก้ไขขาอุปกรณ์ให้ถูกต้องตามที่เราได้ออกแบบไว้
4. นำไฟล์ .xnf ที่แก้ไขแล้วมา Place & Route
5. เสร็จเรียบร้อยแล้วจึงดาวน์โหลดผ่านสาย Xchecker ลงที่บอร์ดทดสอบ
6. ถ้าเราต้องการ simulate ดูว่าวงจรที่เราออกแบบมาทำงานถูกต้องหรือไม่เราต้องนำไฟล์ .vhd มา simulate โดยใช้ Modelsim

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบฮาร์ดแวร์



รูปที่ 5.1 แสดงถึงโมดูลต่างๆ

จากรูปที่ 5.1 ในการออกแบบบอร์ดทดลองเอฟพีจีเอนี้ภายในบอร์ดทดลองจะประกอบเอฟพีจีเอ โมดูลเป็นโมดูลหลักและจะประกอบด้วยโมดูลต่างๆแบ่งตามลักษณะการติดต่อกับเอฟพีจีเอ โมดูลได้ดังนี้

- Input Module
    1. Switch โมดูล
    2. Keyboard โมดูล
- Output Module
    1. 7-Segment โมดูล
    2. LCD โมดูล
    3. Dot Matrix โมดูล
    4. LED โมดูล
    5. Stepping Motor โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Input/Output Module
    1. เมโมรี(Memory) โมดูล
    2. DC-Motor โมดูล
    3. RS-232 port โมดูล
    4. Printer port โมดูล

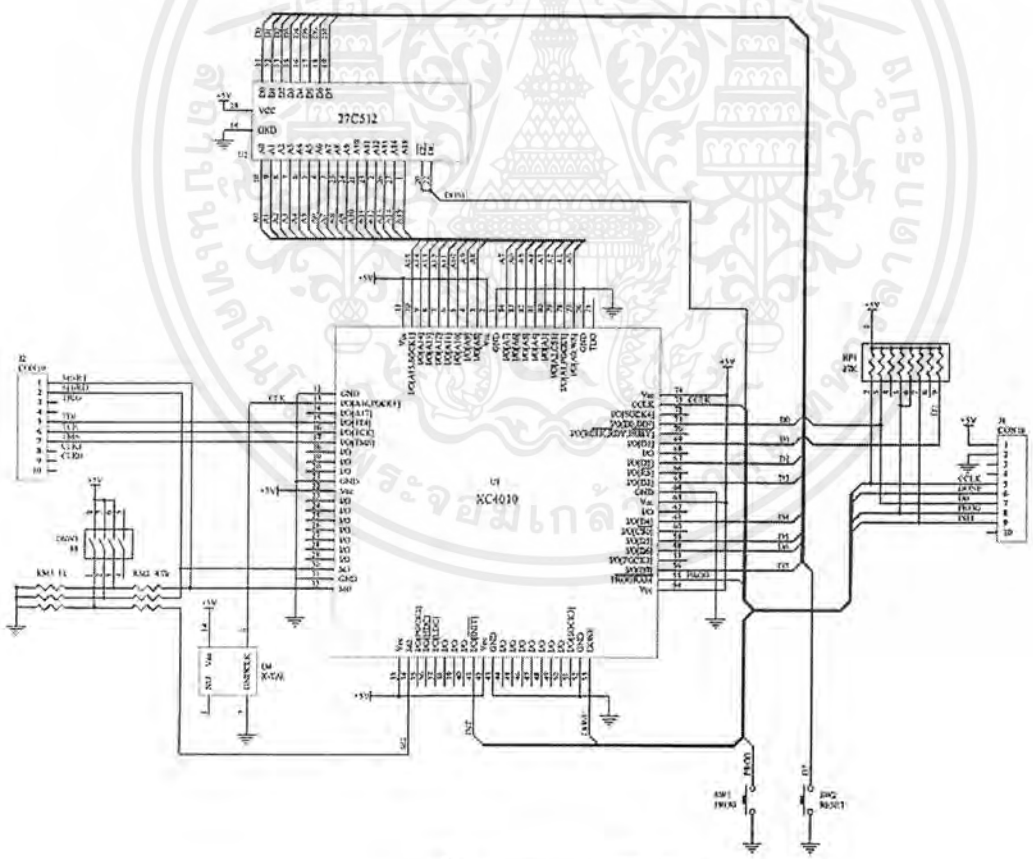
บอร์ดทดลองเอฟพีจีเอได้ออกแบบให้มีหลายโมดูลเพื่อที่จะได้ให้ผู้ใช้งานสามารถทดลองได้หลากหลายรูปแบบโดยที่แต่ละโมดูลจะเป็นอิสระต่อกัน คือสามารถใช้โมดูลเดียวได้หรือสามารถนำแต่ละโมดูลมาใช้ด้วยกันก็ได้

5.1 โมดูลต่างๆของบอร์ดทดลอง

5.1.1 เอฟพีจีเอโมดูล

เอฟพีจีเอโมดูลจะมีส่วนประกอบที่สำคัญคือ

- Chip FPGA เบอร์ 4010e
- EPROM เบอร์ 27C512 ขนาด 64K x 8 bit



รูปที่ 5.2 วงจรของเอฟพีจีเอโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.2 จะมีเอฟพีจีเอ เบอร์ XC4010 เป็นตัวควบคุม โดยจะมี jumper J1 ที่ใช้ต่อกับสาย Xchecker เพื่อดาวน์โหลดโปรแกรมจากเครื่องคอมพิวเตอร์ลงเอฟพีจีเอและมี Dip Switch DSW1 เป็นตัวเซตโหมดของเอฟพีจีเอในการดาวน์โหลด ถ้าเราไม่ต้องการที่จะดาวน์โหลดจากสาย Xchecker เราสามารถจะใช้ EPROM เบอร์ 27C512 ซึ่งเก็บโปรแกรมอยู่แล้วมาดาวน์โหลดลงเอฟพีจีเอโดยการกด Switch SW1 ซึ่งเป็น Switch ที่ใช้สำหรับการโปรแกรมด้วย EPROM และนอกจากนี้ในเอฟพีจีเอโมดูลยังมีสัญญาณนาฬิกาจากภายนอกขนาด 10 MHz โดยได้รับมาจาก XTAL U4

ขาต่างๆที่ใช้งานจะมีดังตารางที่ 5.1

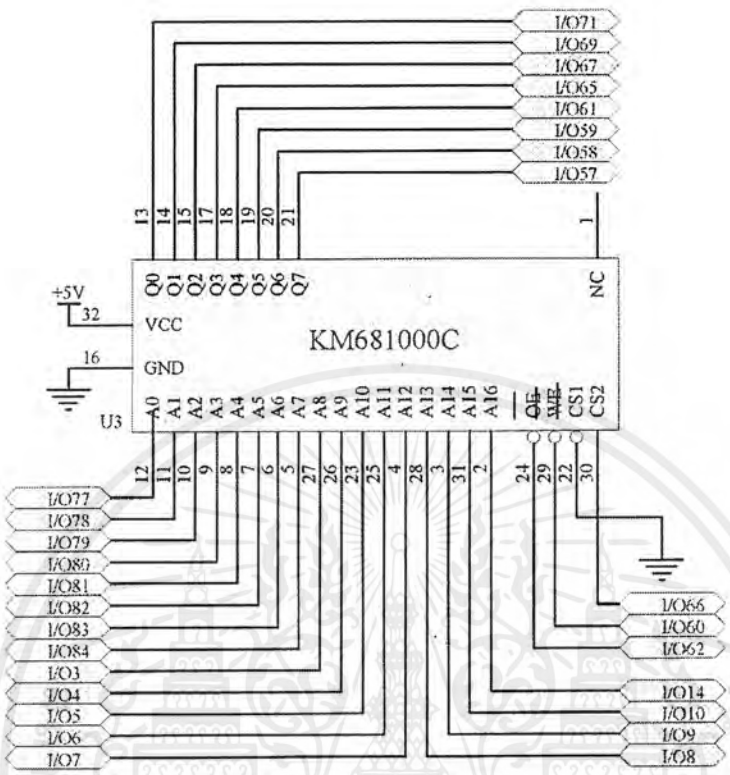
FPGA	FPGA	FPGA	FPGA
GND	VCC	GND	GND
VCC	23	44	65
3	24	45	66
4	25	46	67
5	26	47	68
6	27	48	69
7	28	49	70
8	29	50	71
9	M1	51	72
10	GND	GND	CCLK
VCC	M0	DONE	VCC
GND	VCC	VCC	75
XTAL	M2	$\overline{\text{PROG}}$	76
14	35	56	77
15	36	57	78
16	37	58	79
17	38	59	80
18	39	60	81
19	40	61	82
20	41	62	83
GND	VCC	VCC	84

ตารางที่ 5.1 แสดงถึงขาของเอฟพีจีเอที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 เมโมรี(Memory) โมดูล

มีส่วนประกอบที่สำคัญคือ SRAM เบอร์ KM68100C ซึ่งมีขนาด 128 K x 8



รูปที่ 5.3 วงจรของเมโมรีโมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.2 นี้

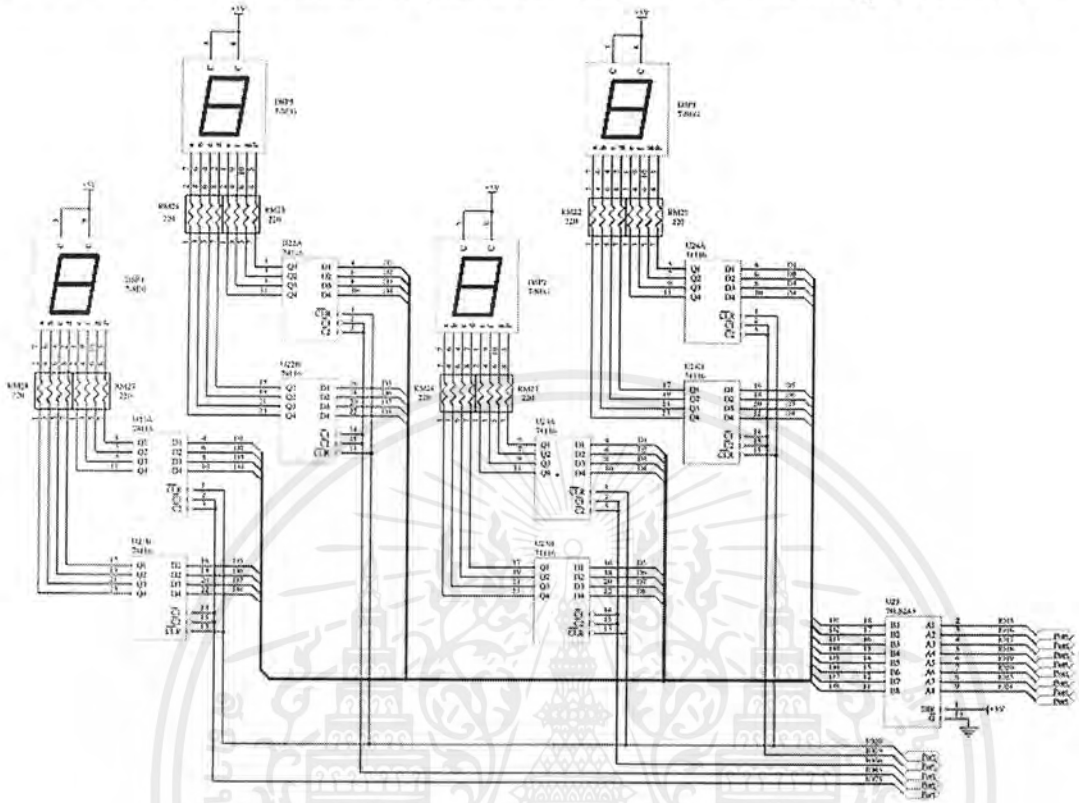
FPGA	SRAM	FPGA	SRAM
3	A8	62	$\overline{OE}$
4	A9	65	D3
5	A10	66	CS2
6	A11	67	D2
7	A12	69	D1
8	A13	71	D0
9	A14	77	A0
10	A15	78	A1
14	A16	79	A2
57	D7	80	A3
58	D6	81	A4
59	D5	82	A5
60	$\overline{WE}$	83	A6
61	D4	84	A7

ตารางที่ 5.2 แสดงขาของ SRAM ที่ต่อเข้ากับขาต่างๆ ของเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 7-Segmentโมดูล

ประกอบไปด้วย 7-segment จำนวน 4 ตัวและแต่ละตัวทำงานเป็นอิสระต่อกันสามารถใช้งานพร้อมกันได้และมี IC เบอร์ 74116 คอย latch ข้อมูลที่ส่งมาจาก 74LS245 และ 74LS245 รับข้อมูลมาจากเอฟพีจีเอ



รูปที่5.4วงจรของ 7-Segment โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.3 นี้

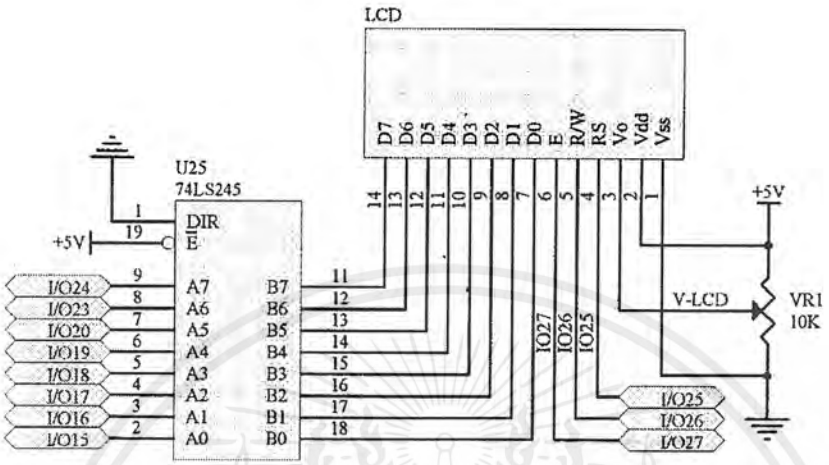
FPGA	7-Segment
15	dp
16	g
17	f
18	e
19	d
20	c
23	b
24	a
29	$\overline{EI}$
30(M1)	$\overline{E2}$
45	$\overline{E3}$
75(TDO)	$\overline{E4}$

ตารางที่ 5.3 แสดงขาของ 7-Segment ที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 LCD โมดูล

มีส่วนประกอบที่สำคัญคือ LCD ขนาด 4 x 20 ตัวอักษรซึ่งมี R ปรับค่าขนาด 10 k เป็นตัวปรับความสว่างของ LCD และมี 74LS245 เป็นตัวรับข้อมูลมาจากเอฟพีจีเออีกต่อหนึ่งเพื่อเป็นการป้องกันวงจรเวลาเกิดกระแสเกินข้างใดข้างหนึ่งจะไม่ได้รับความเสียหาย



รูปที่ 5.5 วงจรของ LCD โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.4 นี้

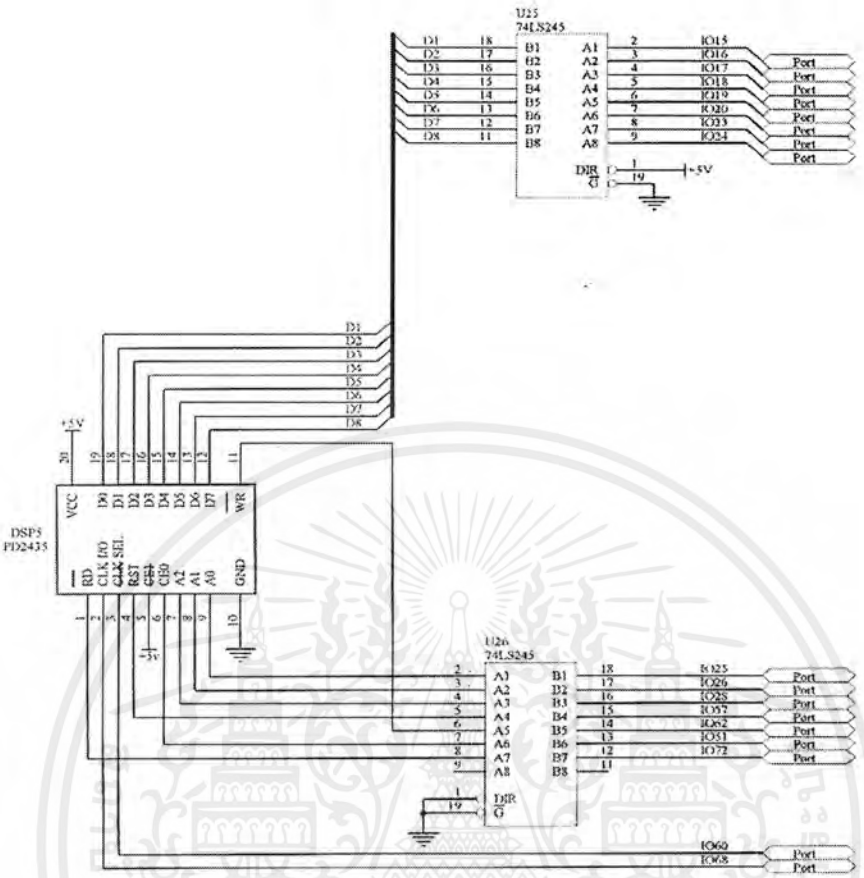
FPGA	LCD
15	D0
16	D1
17	D2
18	D3
19	D4
20	D5
23	D6
24	D7
25	RS
26	R/W
27	E

ตารางที่ 5.4 แสดงขาของ LEDที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.5 Dot Matrix โมดูล

เป็น Dot matrix ขนาด 5 x 7 จำนวน 4 ชุด แต่รวมอยู่ในตัวถังเดียวกันมีชื่อว่า PD2435



รูปที่ 5.6 วงจรของ Dot Matrix โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.5 นี้

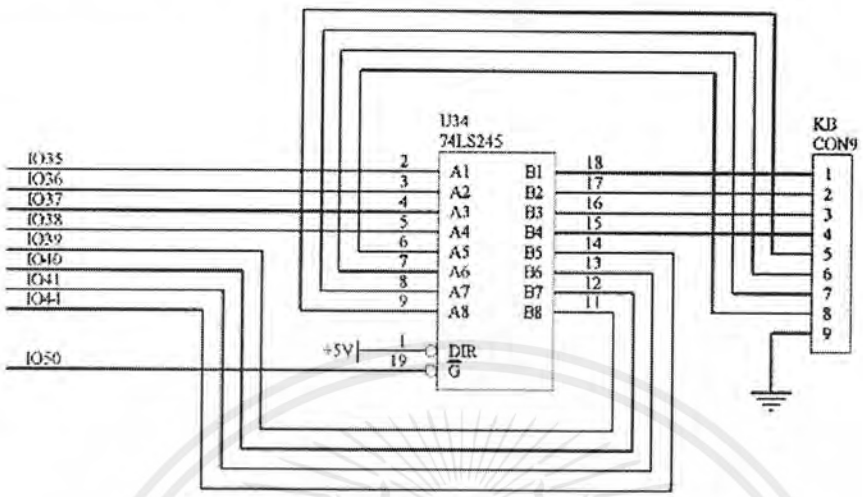
FPGA	Dot Matrix
15	D0
16	D1
17	D2
18	D3
19	D4
20	D5
23	D6
24	D7
25	A0
26	A1
28	A2
51	CE0
57	RST
60	$\overline{\text{CLK SEL}}$
62	$\overline{\text{WR}}$
68	CLK I/O
72	$\overline{\text{RD}}$

ตารางที่ 5.5แสดงขาของ Dot Matrix ที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.6 Keyboard โมดูล

เป็น Keyboard ขนาด 4 x 4 คีย์จำนวน 1 ชุดและมี IC 74LS245 เป็นตัวรับและส่งข้อมูลจาก Keyboard เพื่อส่งต่อให้กับเอฟพีจีเอ



รูปที่ 5.7 วงจรของ Keyboard โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.6 นี้

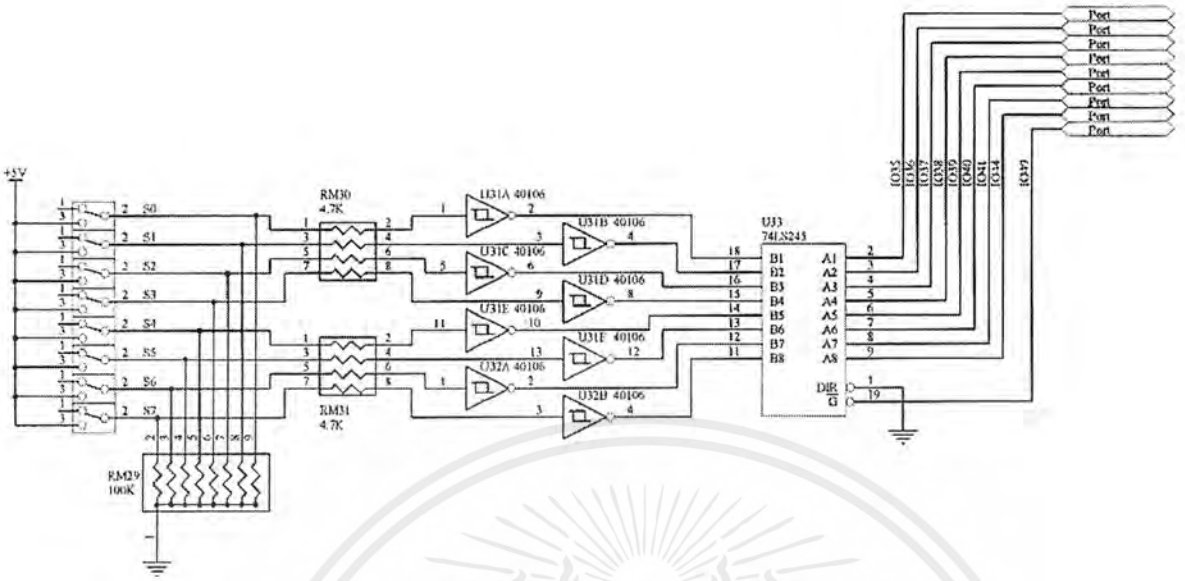
FPGA	Keyboard
35	1
36	2
37	3
38	4
39	5
40	6
41	7
44	8
50	$\overline{E}$

ตารางที่ 5.6 แสดงขาของ Keyboard ที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.7 Switch โมดูล

เป็นสวิทซ์โยก 8 ตัว



รูปที่ 5.8 วงจรของ Switch โมดูล

จากวงจรจะใช้ Switch ทั้งหมด 8 ตัวเพื่อให้เป็นอินพุทขนาด 8 bit และจะมี IC เบอร์ 40106 เป็น Schmitt Trigger และใช้ 74LS245 เพื่อป้องกันเอฟพีจีเอเสียหายถ้าเกิดกระแสเข้ามาจาก Switch เกิน

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.7 นี้

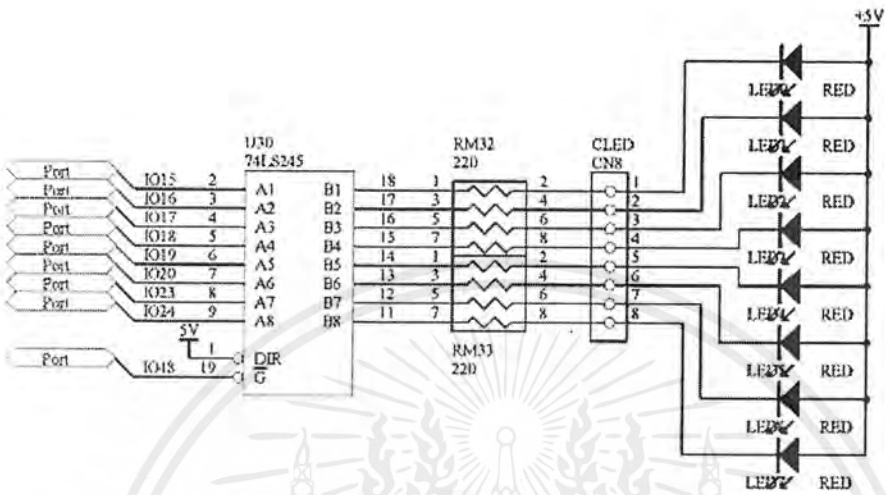
FPGA	Switch
35	S0
36	S1
37	S2
38	S3
39	S4
40	S5
41	S6
44	S7
49	$\overline{E}$

ตารางที่ 5.7 แสดงขาของ Switchที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.8 LED โมดูล

เป็น LED จำนวน 8 หลอด และแต่ละหลอดยังมีจุดที่สามารถต่อไปใช้งานภายนอกได้อีกด้วย โดยที่ได้รับข้อมูลมาจาก IC 74LS245 และยังทำหน้าที่เป็นตัว Enable ให้กับโมดูลนี้อีกด้วย และ 74LS245 ได้รับข้อมูลและคำสั่ง Enable มาจากเอฟพีจีเอ



รูปที่ 5.9 วงจรของ LED โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.8 นี้

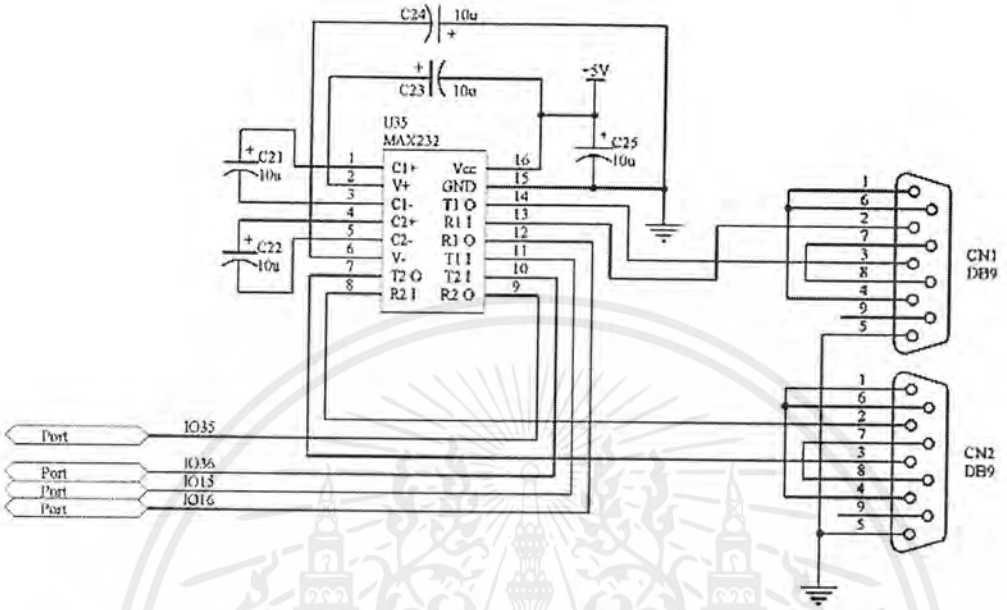
FPGA	LED
15	LED0
16	LED1
17	LED2
18	LED3
19	LED4
20	LED5
23	LED6
24	LED7
48	$\overline{E}$

ตารางที่ 5.8 แสดงขาของ LED ที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.9 RS-232 Port โมดูล

จะประกอบด้วย RS-232 จำนวน 2 ตัวเป็นตัวผู้และตัวเมียอย่างละตัวต่อกับ MAX232 เพื่อที่จะได้ต่อกับเอฟพีจีเอได้ เพราะว่าเอาท์พุทที่ออกมาจาก RS-232 เป็น 12 โวลต์แต่เอฟพีจีเอใช้ 5 โวลต์



รูปที่ 5.10 วงจรของ RS-232 Port โมดูล

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.9 นี้

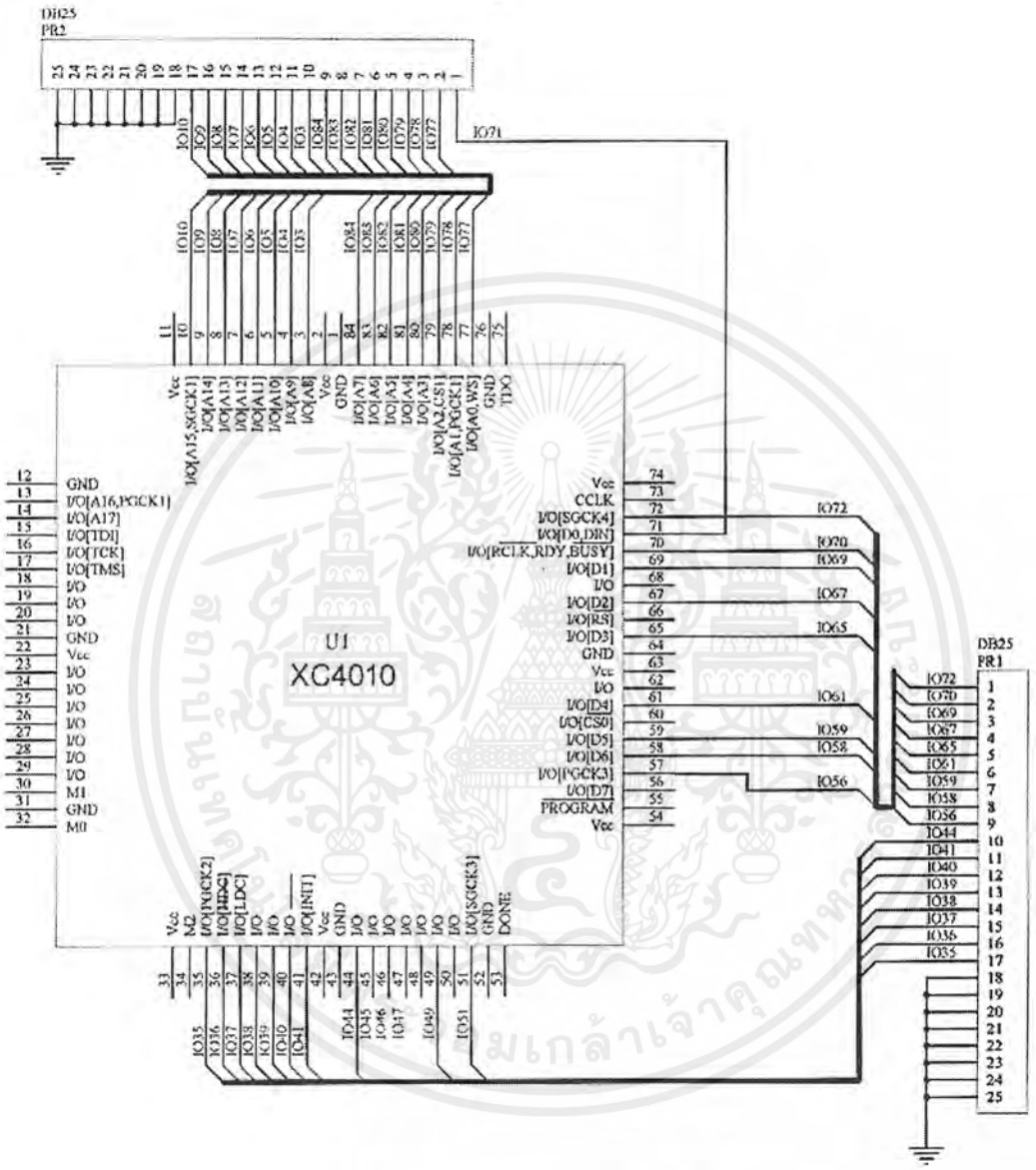
FPGA	RS-232(Male)	RS-232(Female)
15	TX	
16	RX	
35		RX
36		TX

ตารางที่ 5.9 แสดงขาของ RS-232 ทั้ง 2ตัวที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.10 Printer Port โมดูล

จะประกอบด้วย port printer จำนวน 2 ตัวเป็นตัวผู้และตัวเมียอย่างละตัว



รูปที่ 5.11 วงจรของ Printer Port โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.10 นี้

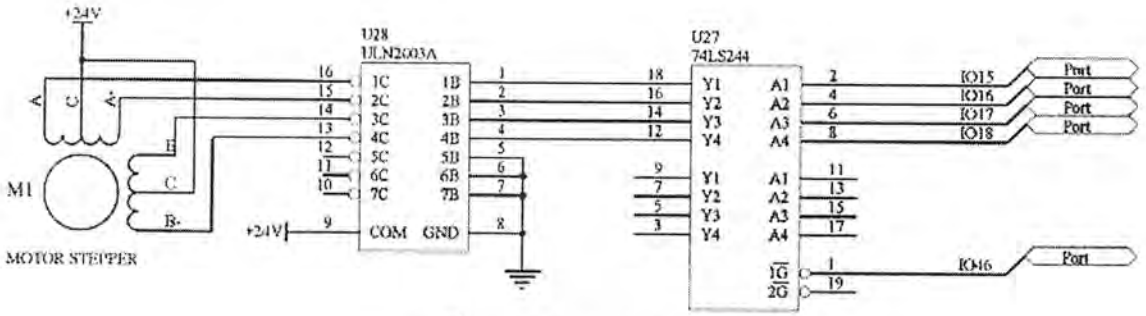
FPGA	Printer Port (Male)	Printer Port (Female)
3	10	
4	11	
5	12	
6	13	
7	14	
8	15	
9	16	
10	17	
35		17
36		16
37		15
38		14
39		13
40		12
41		11
44		10
57		9
58		8
59		7
61		6
65		5
67		4
69		3
70		2
72		1
76	1	
77	2	
78	3	
79	4	
80	5	
81	6	
82	7	
83	8	
84	9	

ตารางที่ 5.10 แสดงขาของ Printer Port ทั้ง 2 ตัวที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.11 Stepping Motor โมดูล

เป็น Stepping Motor 4 phase ใช้ไฟเลี้ยง 24 โวลท์



รูปที่ 5.12 วงจรของ Stepping Motor โมดูล

จากรูปที่ 5.12 เป็นวงจรของ Stepping Motor ซึ่งต้องใช้ IC ULN2003A มาเป็นตัวขับมอเตอร์และใช้ IC เบอร์ 74LS244 เป็นตัวป้องกันเอฟพีจีเอถ้าเกิดมอเตอร์หรือตัวขับทำให้เกิดกระแสเกิน

ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.11 นี้

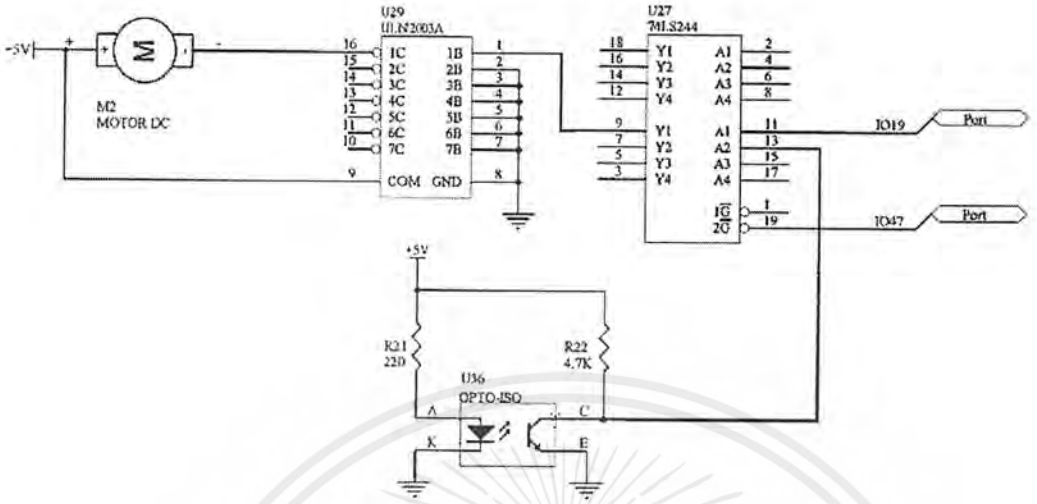
FPGA	Stepping Motor
15	A
16	$\overline{A}$
17	B
18	$\overline{B}$
46	$\overline{E}$

ตารางที่ 5.11 แสดงขาของ Stepping Motor ที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.12 DC-Motor โมดูล

มีส่วนประกอบที่สำคัญคือ DC Motor และชุด Sensor



รูปที่ 5.13 วงจรของ DC Motor โมดูล

จากวงจร DC Motor เราก็ต้องใช้ตัวขับ ULN2003A เช่นเดียวกับ Stepping Motor และในวงจรนี้ยังมีชุด Sensor ที่มีทั้งตัวรับและตัวส่งอยู่ในตัวเดียวกันเพื่อที่ให้ผู้ออกแบบโปรแกรมสามารถนับรอบความเร็วของ DC Motor ได้

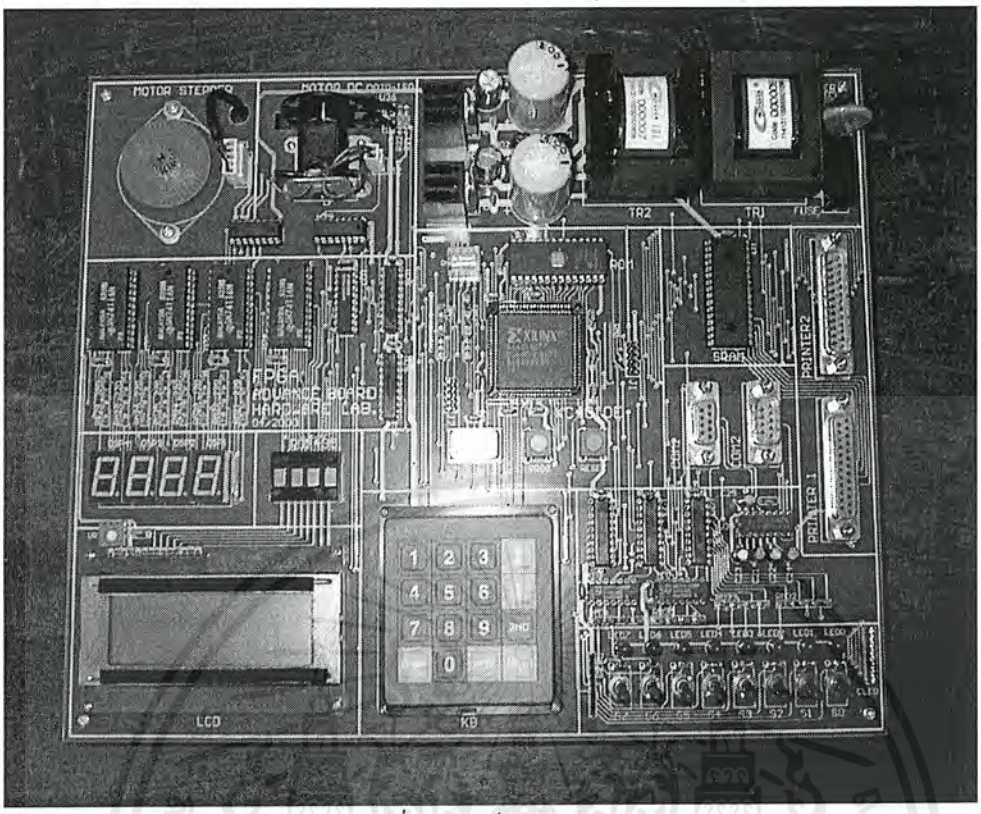
ขาที่ใช้งานกับ FPGA มีดังตารางที่ 5.12 นี้

FPGA	Dc Motor
19	IN
44	OUT
47	$\overline{E}$

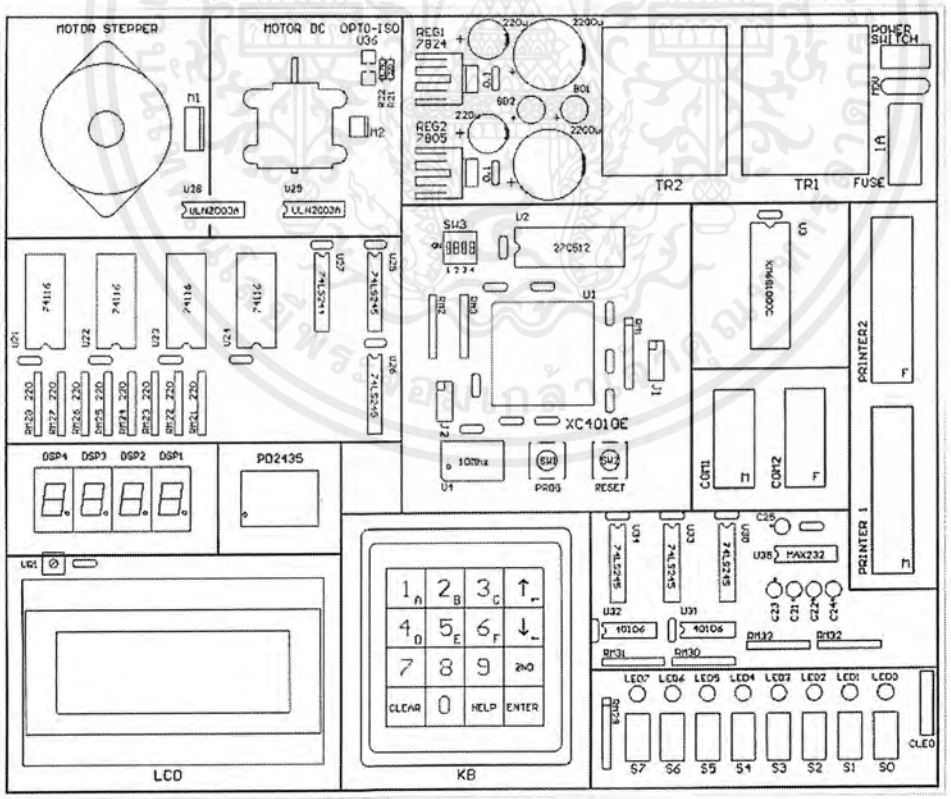
ตารางที่ 5.12 แสดงขาของ DC Motor โมดูลที่ต่อใช้งานกับเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 วงจรรวมของบอร์ดทดลอง

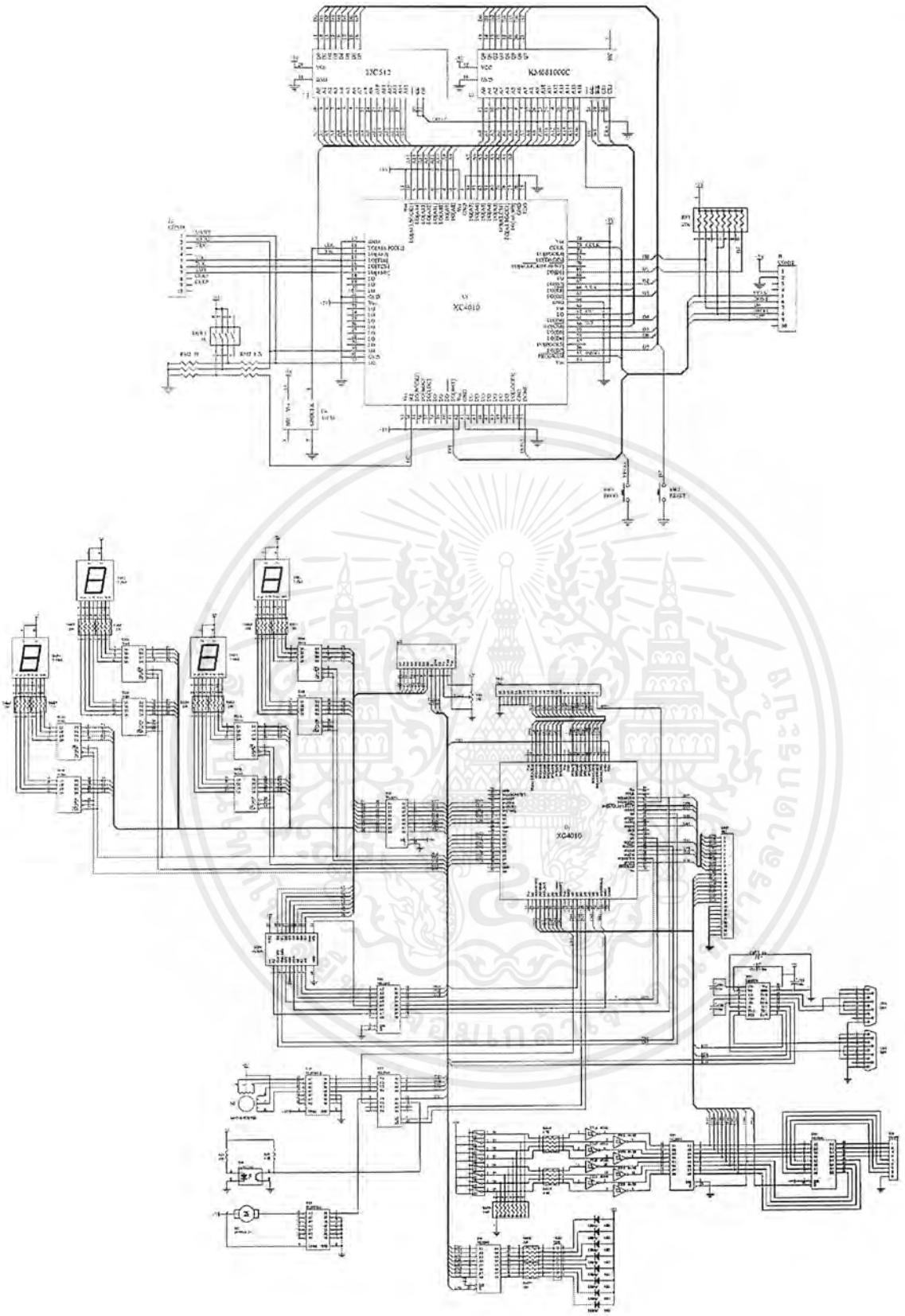


รูปที่ 5.14 บอร์ดทดลองจริง



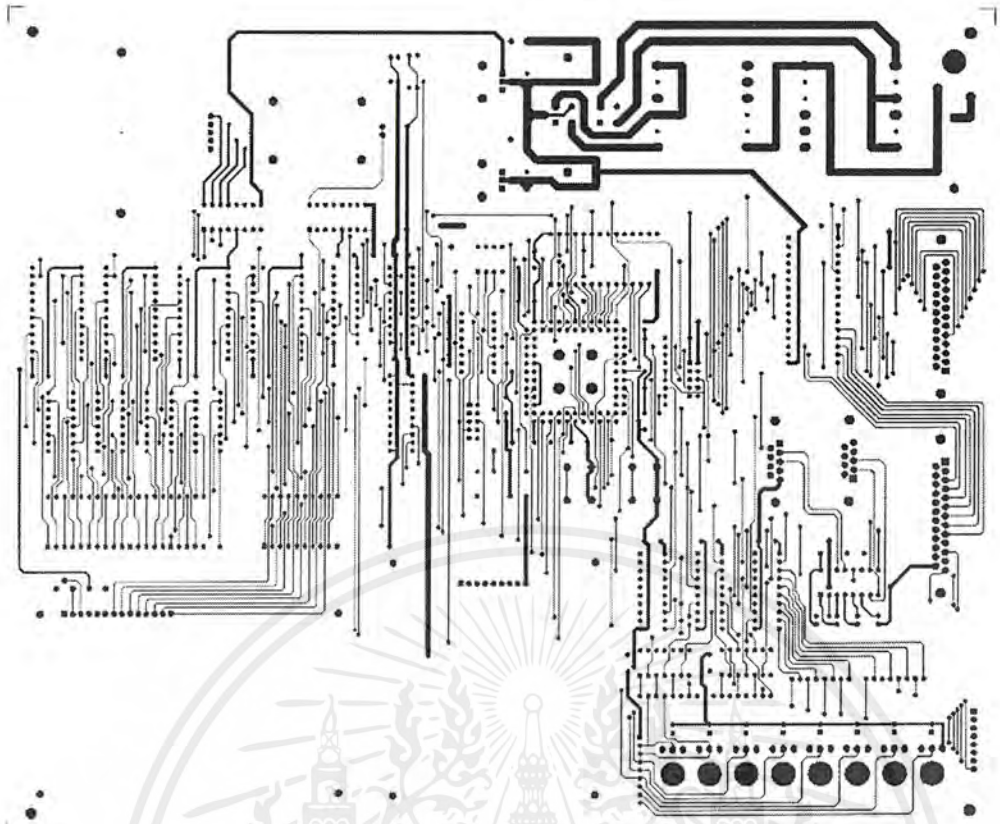
รูปที่ 5.15 การจัดวางโมดูลต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

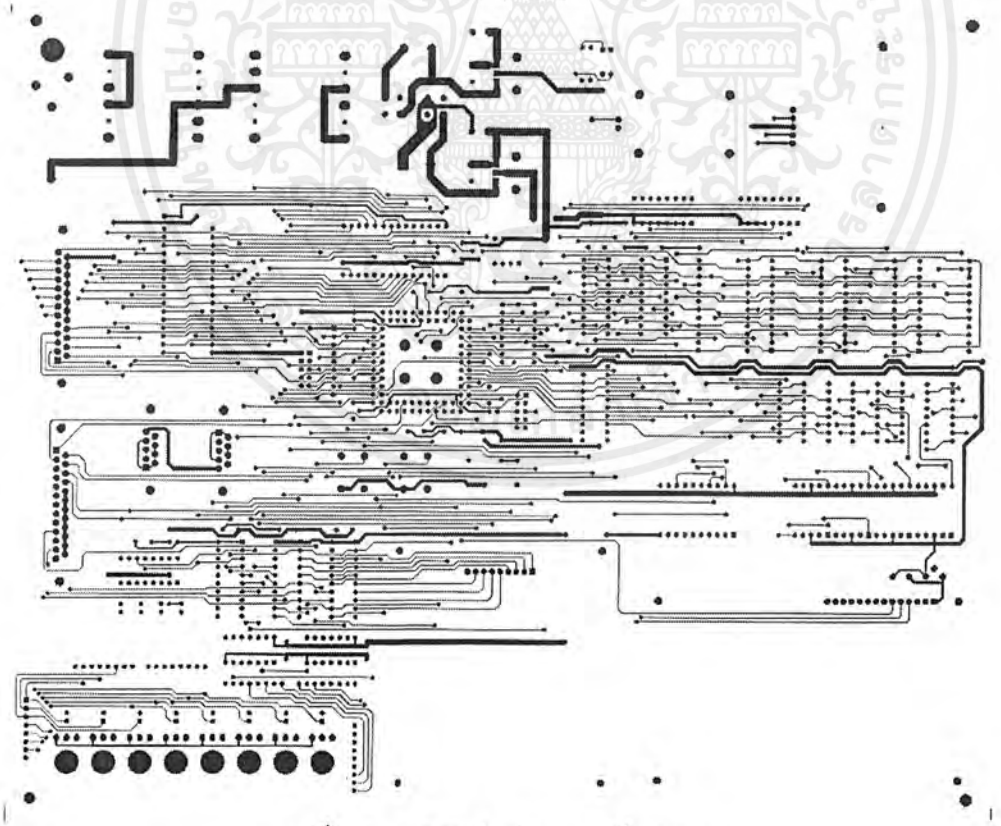


รูปที่ 5.16 วงจร schematic รวมของบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

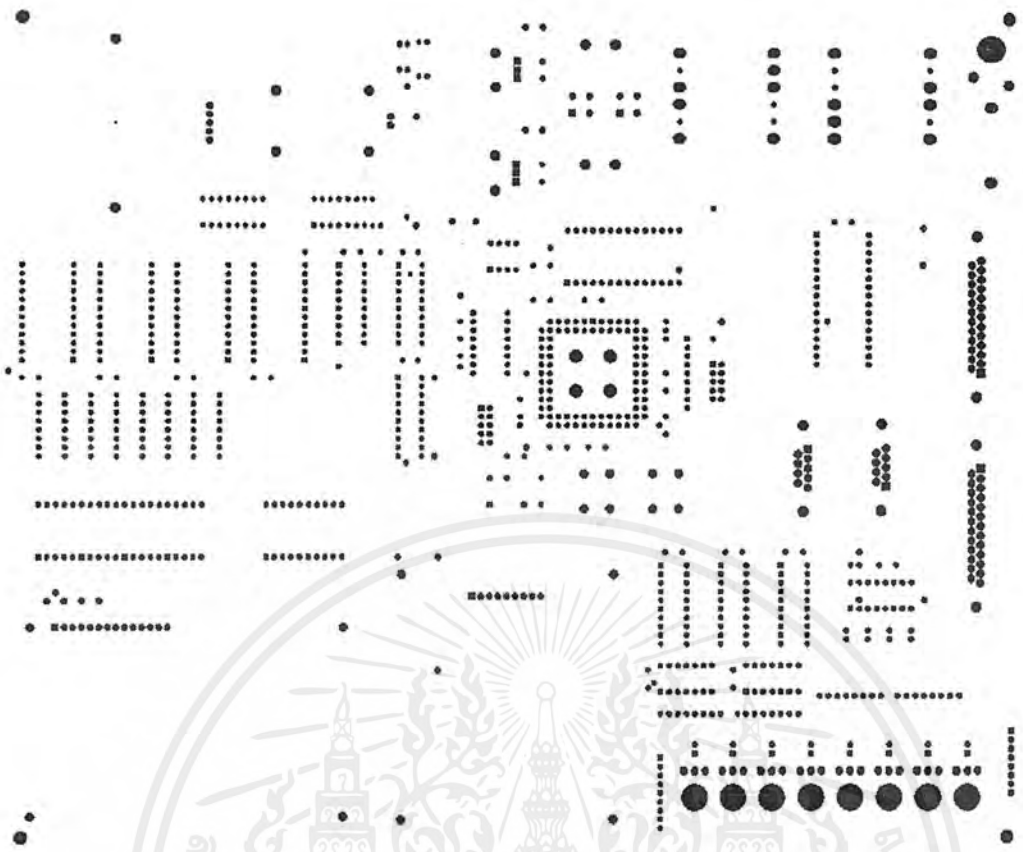


รูปที่ 5.17 ลาย PCBด้านบนของบอร์ดทดลอง



รูปที่ 5.18 ลาย PCB ด้านล่างของบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 รูของขาอุปกรณ์ทั้งหมดที่ใช้ในบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

6.1 การทดลองที่ 1

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

3. โลจิกโพรบ

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้ส่งค่าพัลซ์ออกที่ I/O 15
- ทดลองให้ออกที่ I/O อื่นๆ

ผลการทดลอง

- มีสัญญาณพัลซ์ที่ขา I/O ที่กำหนด

สรุปผลการทดลอง

- เอฟพีจีเอโมดูลสามารถทำงานได้อย่างถูกต้อง

6.2 การทดลองที่ 2

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- เมโมรี(Memory) โมดูล
- สวิทซ์โมดูล
- 7-Segment โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้เก็บค่าของข้อมูลไว้ในเมโมรีแล้วใช้สวิทซ์ในการเลือกให้ข้อมูลมาแสดงผลที่ 7-Segment

ผลการทดลอง

- 7-Segment มีการแสดงผลได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองที่ 3

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- 7-Segment โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลเป็นเคาท์เตอร์โดยให้นับจาก 0-F ให้มาแสดงผลที่ 7-Segment

ผลการทดลอง

- 7-Segment มีการแสดงผลได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.4 การทดลองที่ 4

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- LCD โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้แสดงคำว่า "HELLO" ให้วิ่งวนไปเรื่อยๆใน LCD

ผลการทดลอง

- LCD มีการแสดงผลได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.5 การทดลองที่ 5

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- Dot Matrix โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้แสดงคำว่า "STOP"

ผลการทดลอง

- Dot Matrix มีการแสดงผลได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.6 การทดลองที่ 6

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- สวิทช์โมดูล
- LED โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้ใช้สวิทช์ในการควบคุมการติดดับของหลอด LED

ผลการทดลอง

- สวิทช์และหลอด LED มีการแสดงผลได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.7 การทดลองที่ 7

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- Keyboard โมดูล
- LCD โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลเมื่อมีการกดปุ่มที่คีบอร์ดแล้วมีการแสดงผลที่ LCD

ผลการทดลอง

- เมื่อมีการกดปุ่มคีบอร์ดแล้วมีการแสดงผลที่ LCD ได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.8 การทดลองที่ 8

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- 7-Segment โมดูล
- Printer Port โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้รับค่ามาจาก Printer Port แล้วมาแสดงผลที่ 7-Segment
- เขียนโปรแกรมด้วยภาษาแอสแซมบลีให้ส่งค่าของรหัสแอสกีของคีบอร์ดที่กดมาที่ Printer Port

ผลการทดลอง

- เมื่อมีการกดปุ่มคีบอร์ดแล้วมีการแสดงผลที่ 7-Segment ได้ถูกต้องตามที่ได้กำหนดไว้

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.9 การทดลองที่ 9

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอโมดูล
- Stepping Motor โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้ Stepping Motor มีการหมุนทีละเฟส

ผลการทดลอง

- Stepping Motor มีการทำงานได้ถูกต้องตามที่เรากำหนด

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

6.10 การทดลองที่ 10

โมดูลที่ใช้ในการทดลอง

- เอฟพีจีเอ โมดูล
- DC Motor โมดูล

อุปกรณ์ 1. บอร์ดทดลองเอฟพีจีเอ

2. สาย Parallel สำหรับดาวน์โหลดข้อมูล

ขั้นตอนการทดลอง

- เขียนโปรแกรมด้วยภาษาวีเอชดีแอลให้ DC Motor มีการหมุนโดยใช้ความกว้างของพัลซ์เป็นตัวควบคุมความเร็ว

ผลการทดลอง

- DC Motor มีการทำงานได้ถูกต้องตามที่เรากำหนด

สรุปผลการทดลอง

- ทุกโมดูลที่ใช้ในการทดลองทำงานตามที่กำหนดได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลและบทวิจารณ์

จากการออกแบบและการทดลองทั้งทางด้านฮาร์ดแวร์ที่ได้กล่าวผ่านมาแล้วนั้นทำให้ผู้ใช้งานสามารถนำบอร์ดทดลองเอฟพีจีเอในโครงงานนี้ไปใช้ประโยชน์ได้จริงๆ โดยไม่จำเป็นต้องอาศัยเครื่องมืออื่นๆ มากนัก เพราะว่าในบอร์ดทดลองได้มีอุปกรณ์ที่จำเป็นต่อการทดสอบได้ด้วยตนเองพอสมควรในระดับหนึ่งแล้ว

อย่างไรก็ตามเนื่องจากเวลาในการทำโครงงานนี้มีจำกัด ในขณะที่เนื้อหาและสิ่งที่ต้องทำในโครงงานมีมาก มีทั้งการทำบอร์ดทดลอง ,การเขียนโปรแกรมเพื่อทดสอบการทำงานของบอร์ดทดลอง และการหาวิธีทดสอบบอร์ดทดลองเอฟพีจีเอ ซึ่งมักเกิดปัญหาที่คาดไม่ถึงอยู่เสมอ และต้องใช้เวลานานกว่าจะพบและทำการแก้ไขข้อผิดพลาดได้ทั้งที่บนบอร์ดทดลองและโปรแกรมที่ใช้ทดสอบจึงเป็นไปได้ที่อาจจะมีข้อผิดพลาดบางประการที่ทางผู้จัดทำยังทดสอบไปไม่ถึง ซึ่งทางผู้จัดทำต้องขออภัยล่วงหน้าและจะได้หาโอกาสดำเนินการแก้ไขต่อไป

7.1 วิธีการดำเนินงาน

ในโครงงานนี้ได้ทำการศึกษาถึงการทำเอฟพีจีเอมาใช้งาน ,ศึกษาการออกแบบวงจรต่างๆ เพื่อใช้งานกับเอฟพีจีเอ ,ศึกษาภาษาวีเอชดีแอลเพื่อใช้ในการทดสอบบอร์ดทดลองที่ใช้เอฟพีจีเอ และได้ศึกษาการใช้ซอฟแวร์ต่างๆ ที่ต้องนำมาใช้งานในการออกแบบบอร์ดทดลองและการเขียนโปรแกรมด้วยภาษาวีเอชดีแอล

จากนั้นได้ทำการสร้างบอร์ดทดลองเอฟพีจีเอขึ้นมาและเขียนโปรแกรมทดสอบการทำงานของโมดูลต่างๆ

7.2 การออกแบบและทดสอบ

ในโครงงานนี้ตัวบอร์ดทดลองจะมีการแบ่งเป็นโมดูลต่างๆ เพื่อให้สะดวกแก่การใช้งานและให้ผู้ที่ใช้งานเข้าใจได้ง่ายโดยจะมีโมดูลต่างๆดังนี้

1. เอฟพีจีเอโมดูล
2. เมโมรี(memory) โมดูล
3. 7-segment โมดูล
4. LCD โมดูล
5. Dot Matrix โมดูล
6. Keyboard โมดูล
7. Switch โมดูล
8. LED โมดูล
9. RS-232 port โมดูล
10. Printer port โมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Steping Motor โมดูล

12. DC-Motor โมดูล

การทดสอบได้เขียนซอฟแวร์ทดสอบด้วยภาษาวีเอชดีแอล โดยการทดสอบจะทดสอบทีละโมดูล หรือโมดูลใดที่ไม่สามารถทดสอบเพียงโมดูลเดียวได้ก็จะใช้ทดสอบร่วมกับโมดูลอื่นๆ

7.3 ปัญหาและอุปสรรคในการดำเนินงาน

เนื่องจากการออกแบบฮาร์แวร์เป็นวงจรรวมขนาดใหญ่ในการทำ PCB ต้องใช้เวลาในการหาเส้นทางนาน และการเขียนซอฟแวร์ทดสอบ โปรแกรมที่ใช้เขียนมีขนาดใหญ่และใช้หน่วยความจำมาก ดังนั้นเครื่องคอมพิวเตอร์ที่ใช้ในการออกแบบและเขียนโปรแกรมต้องมีความเร็วสูงและหน่วยความจำมาพอสมควรจึงจะให้การทำงานที่เร็วขึ้นกว่านี้และเครื่องไม่แฮงค์ (Hank)

ภาษาวีเอชดีแอลเนื่องจากในประเทศไทยยังมีการใช้งานไม่แพร่หลายมากนักจึงหาข้อมูลและผู้ให้คำปรึกษาได้ยาก

ตัวเอฟพีจีเอมีราคาสูงมากทำไม่สามารถหามาทดสอบได้หลายเบอร์

7.4 คำวิจารณ์

บอร์ดทดสอบที่ใช้เอฟพีจีเอนี้เหมาะแก่ผู้ที่ทำการศึกษาเบื้องต้นเท่านั้นเพราะว่าโมดูลที่ใช้งานมีอยู่จำกัด และตัวบอร์ดเองก็มีขนาดใหญ่เกินไปและนอกจากนั้นบอร์ดทดลองยังสามารถใช้กับเอฟพีจีเอได้เพียงเบอร์เดียวเท่านั้นคือ เบอร์ 4010e

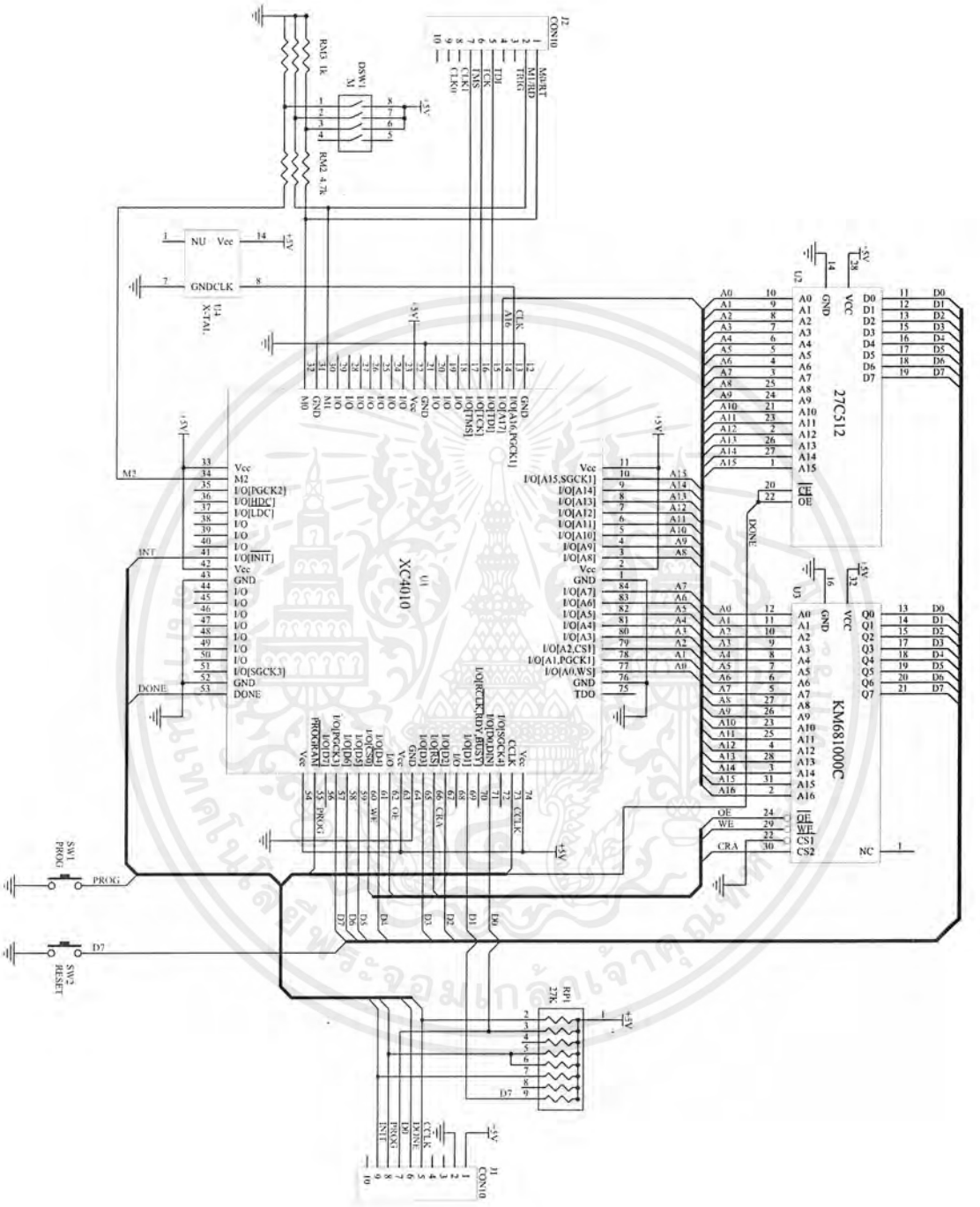
7.5 สิ่งที่ต้องพัฒนาต่อไป

- ตัวบอร์ดต้องสามารถใช้กับเอฟพีจีเอเบอร์ต่างๆ ได้โดยต้องมีการทำขาหรือ Socket เป็นแบบ Universal คือจะใส่เอฟพีจีเอเบอร์อะไรและมีกี่ขาก็ได้
- บอร์ดต้องมีขนาดเล็กลงและใช้งานได้หลากหลายอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

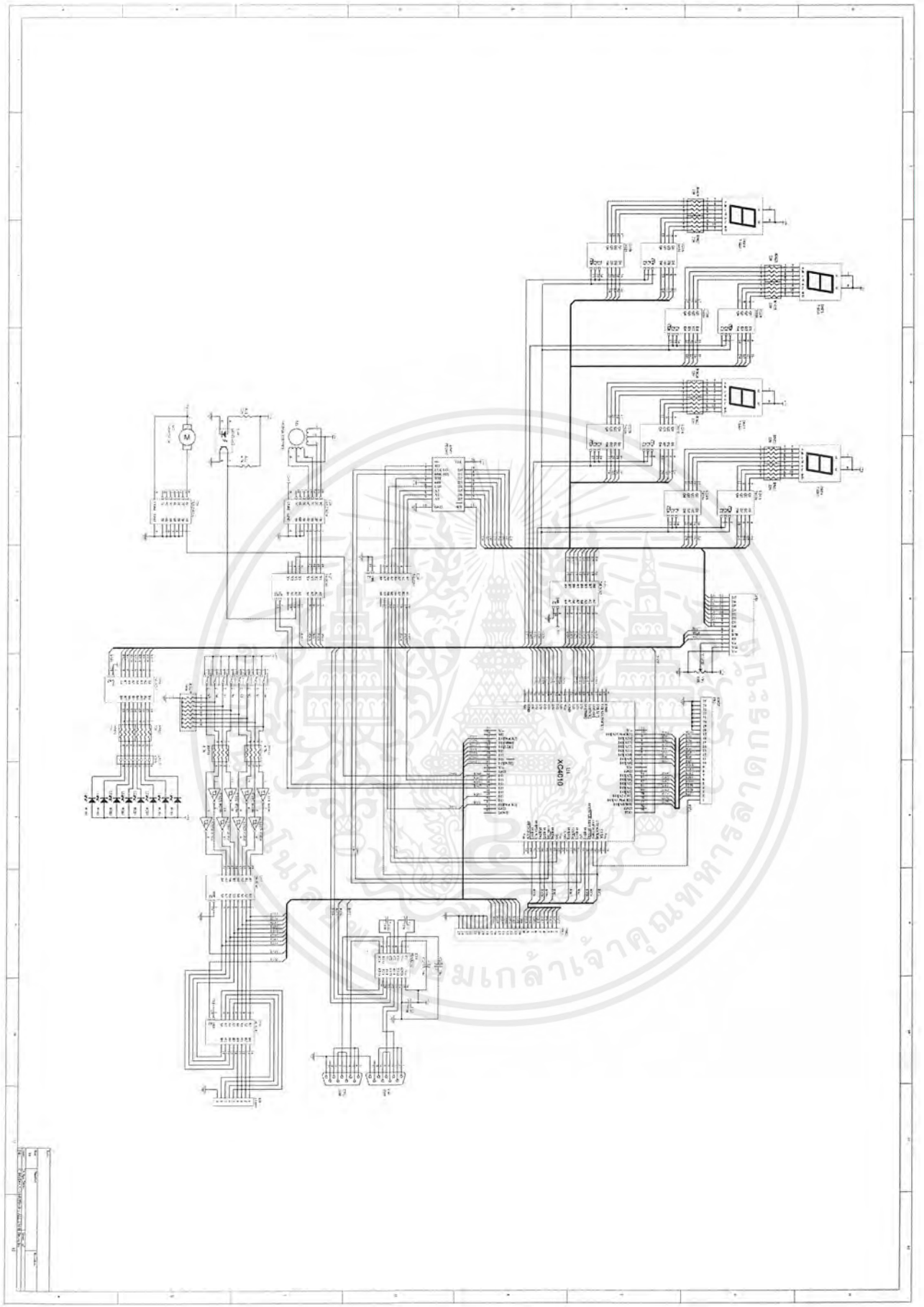
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

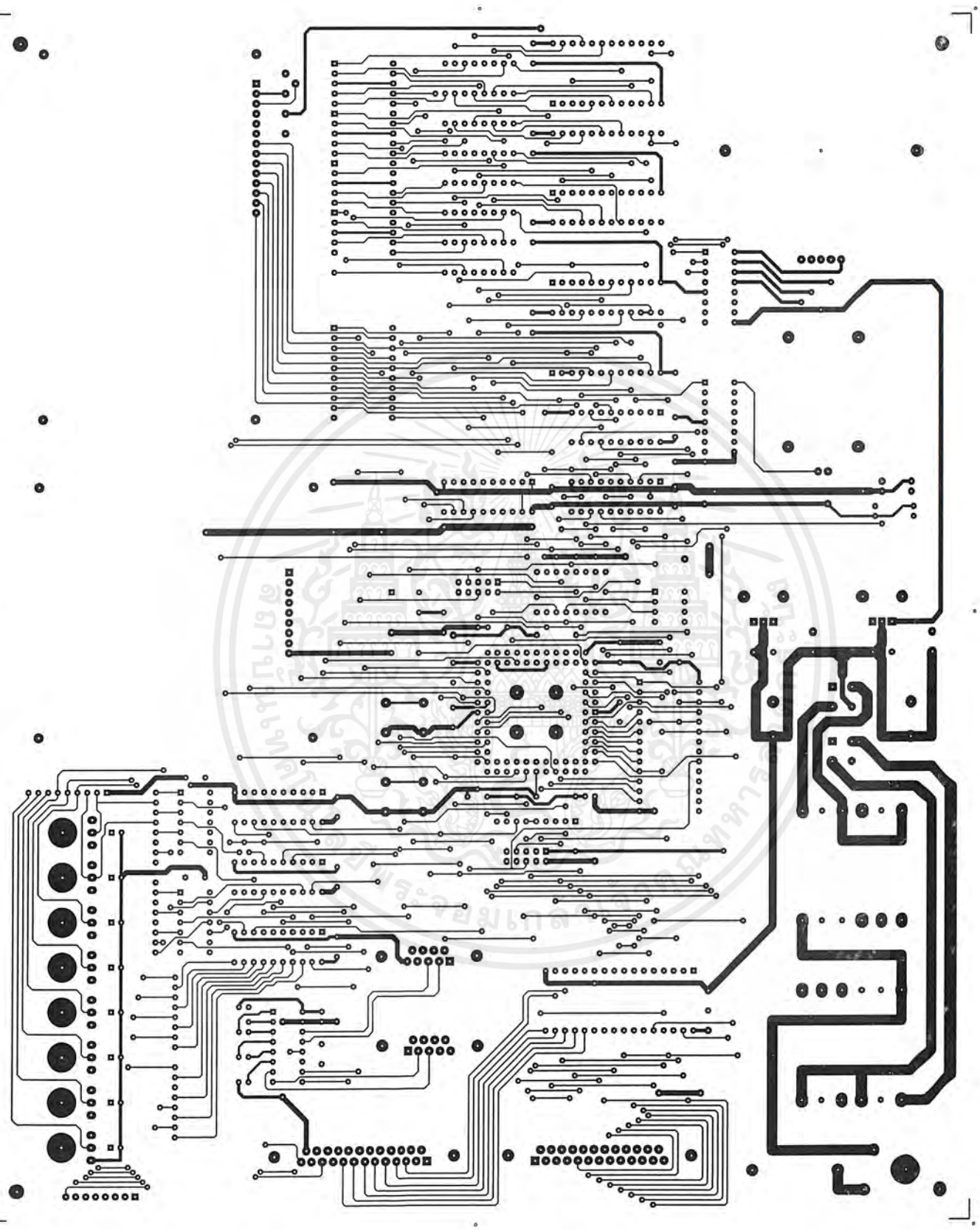


Title		
Size: A3	Number:	Revision:
Date: 17-May-2000	Sheet of	
File: D:\MYDOCU~1\MYPROJ~1\ALL-1.SCH	Drawn By:	

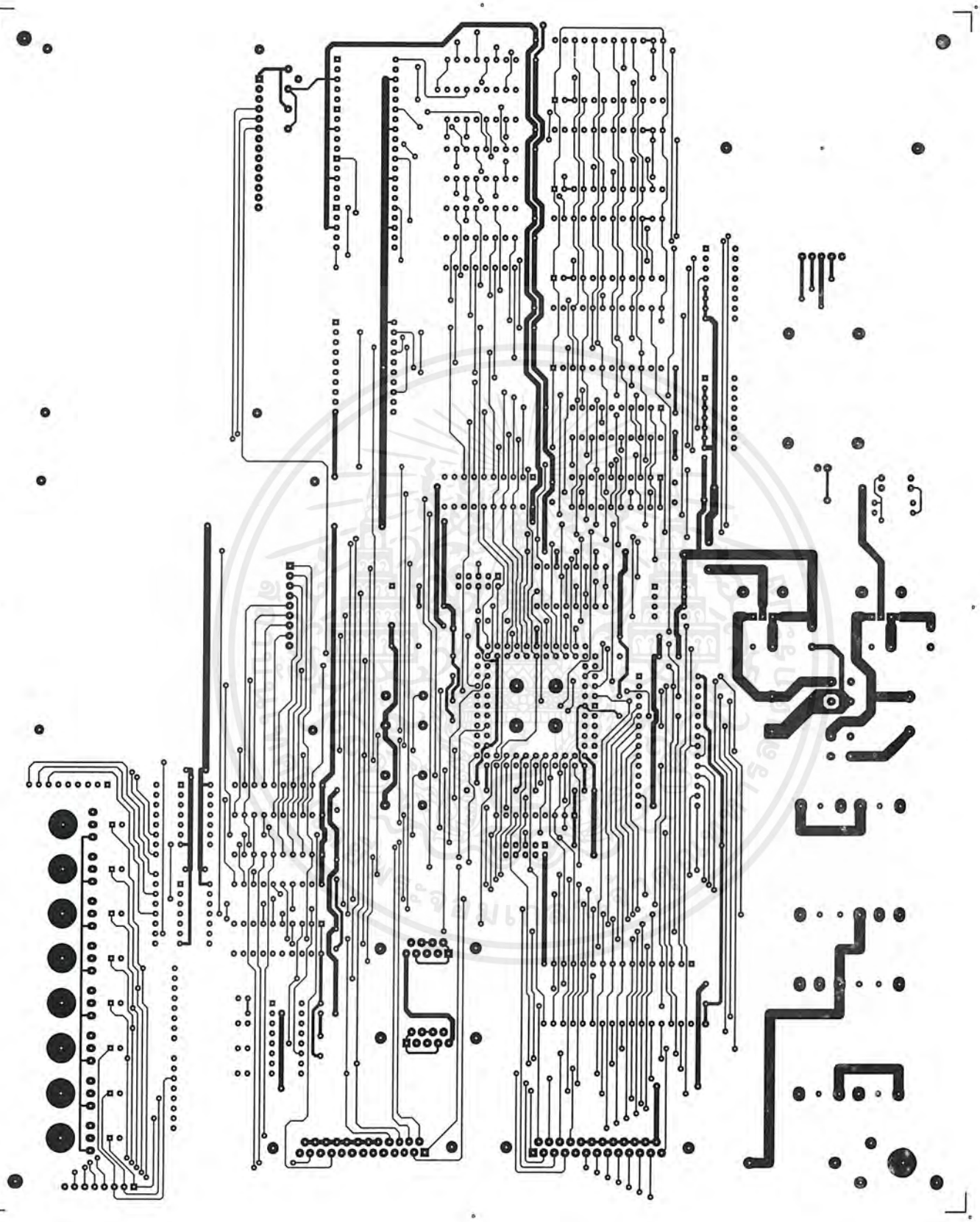
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA	XC4010E	ROM	SRAM	7	LCD	Dot	LED	Switch	KB	RS-232	RS-232	Printer	Printer	DC	Step
Pin No	Detail			Segment		Matrix				1 (F)	2 (M)	1 (F)	2 (M)	Motor	Motor
1	GND														
2	VCC														
3	I/O, A8	A8	A8									ACK			
4	I/O, A9	A9	A9									BUSY			
5	I/O, A10	A10	A10									PE			
6	I/O, A11	A11	A11									SLCT			
7	I/O, A12	A12	A12									AUTO F			
8	I/O, A13	A13	A13									ERR			
9	I/O, A14	A14	A14									INIT			
10	I/O, A15	A15	A15									SLCT IN			
11	VCC														
12	GND														
13	I/O, PGCK1,A16														
14	I/O, A17		A16												
15	I/O, TDI			P	D0	D0	LED0				TX				A
16	I/O, TCK			g	D1	D1	LED1				RX				A-
17	I/O, TMS			f	D2	D2	LED2								B
18	I/O			e	D3	D3	LED3								B-
19	I/O			d	D4	D4	LED4							IN	
20	I/O			c	D5	D5	LED5								
21	GND														
22	VCC														
23	I/O			b	D6	D6	LED6								
24	I/O			a	D7	D7	LED7								
25	I/O				RS	A0									
26	I/O				R/W	A1									
27	I/O				E										
28	I/O			CLEAR		A2									
29	I/O, SGCK2			L0											
30	O, M1			L1											
31	GND														
32	I, M0														
33	VCC														
34	I, M2														
35	I/O, PGCK2							S0	1	RX			SLCT IN		
36	I/O, HDC							S1	2	TX			INIT		
37	I/O, LDC							S2	3				ERR		
38	I/O							S3	4				AUTO F		
39	I/O							S4	5				SLCT		
40	I/O							S5	6				PE		
41	I/O, INIT							S6	7				BUSY		
42	VCC														
43	GND														
44	I/O							S7	8				ACK	OUT	
45	I/O			L2											
46	I/O														E
47	I/O													E	
48	I/O						E								
49	I/O							E							
50	I/O								E						
51	I/O, SGCK3					CE0									
52	GND														
53	DONE	CE/OE													
54	VCC														
55	PROGRAM														
56	I/O, D7	D7											D7		
57	I/O, PGCK3		D7			RST									
58	I/O, D6	D6	D6										D6		
59	I/O, D5	D5	D5										D5		
60	I/O, CS0	WE	WE			SEL									
61	I/O, D4	D4	D4										D4		
62	I/O	OE	OE			WR									
63	VCC														
64	GND														
65	I/O, D3	D3	D3										D3		
66	I/O, RS	CS2	CS2												
67	I/O, D2	D2	D2										D2		
68	I/O					CLK I/O									
69	I/O, D1	D1	D1										D1		
70	I/O, RCLK, RDY/BUSY												D0		
71	I/O, D0, DIN	D0	D0									STB			
72	I/O, SGCK4, DOUT					RD							STB		
73	CCLK														

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

74	VCC														
75	TDO			L3											
76	GND														
77	I/O, A0, WS	A0	A0									D0			
78	I/O, PGCK4	A1	A1									D1			
79	I/O, CS1, A2	A2	A2									D2			
80	I/O, A3	A3	A3									D3			
81	I/O, A4	A4	A4									D4			
82	I/O, A5	A5	A5									D5			
83	I/O, A6	A6	A6									D6			
84	I/O, A7	A7	A7									D7			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Sajjan., G. Shiva. Computer Design & Architecture 2nd ed. New York. HarperCollins Inc. 1991.

[2] Juan, H.P. et. Al. "Top-Down Modeling of RISC Processors in VHDL." IEEE. 1993. pp. 454-459

[3] Bhasker, J. VHDL Primer. Prentice-Hall. 1992.

[4] Z. Navabi. VHDL Analysis and Modeling of Digital Systems. McGraw-Hill Inc. 1993

[5] Ott, D. E. and Wilderotter, T. J. A Designer's Guide to VHDL Synthesis. Klower Academic. 1994.

6] Pak, K. Chan and Samiha Mourad. Digital Design using field programmable gate arrays. PTR Prentice Hall . 1994

[7] Xilinx Inc. The Programmable Logic Data Book. Xilinx Inc. 1994.

[8] Xilinx Inc. Xilinx XACT Viewlogic Interface user guide. Xilinx Inc. 1994.

[9] K.C Chang., Digital design and Modeling with VHDL and Synthesis. IEEE Computer Socicty Pross. California. 1997.

[10] Doulos. "For high quality VHDL and Verilog training for HDL-based FPGA and ASIC design leadership for true expertise in High Level Design." [Online]. Available : http://www.doulos.co.uk August. 1999.

[11] Hardware Debugger reference/User guide, Xilinx Inc. 1 995

[12] Hardware & Peripherals User guide, Xilinx Inc. 1 995

[13] Watch Design – Implementation Tools Tutorial (2.i) , Xilinx Inc. 1 999

[14] การออกแบบวงจรดิจิตอลเพื่อใช้งานกับอุปกรณ์ FPGA ฝ่ายออกแบบวงจรรวมศูนย์วิจัยและพัฒนาเทคโนโลยีไมโครอิเล็กทรอนิกส์ ศูนย์เทคโนโลยีอิเล็กทรอกนิกส์และคอมพิวเตอร์แห่งชาติ พ.ศ. 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้