



ภาควิชาวิศวกรรมศาสตร์วิศวกรรม
 คณะวิศวกรรมศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใบรับรองปริญญาโท

ชื่อหัวข้อ ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51
 Digital System Design for Demonstration by FPGAs Device with MCS-51 Microcontroller

ชื่อนักศึกษา 1. นางสาวกรรณา แยมพราย รหัสประจำตัว 42035326
 2. นายสุทธิพงษ์ จันทะรัง รหัสประจำตัว 42035356
 3. นางสาวสุรีย์พร วงษ์คำหาญ รหัสประจำตัว 42035359

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา

อาจารย์ที่ปรึกษาร่วม อาจารย์กิติพงศ์ มะโน

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา	
2. อาจารย์กิติพงศ์ มะโน	
3. อาจารย์สุชิน อาจหาญ	
4. อาจารย์อำพล ทองระอา	
5. อาจารย์สุระชัย พิมพ์สาดี	

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 25 พฤศจิกายน พ.ศ. 2543 เวลา 11.30 น.

สถานที่สอบ ห้อง ค.310 คณะวิศวกรรมศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
 ลงนาม.....
 (ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาวิศวกรรมศาสตร์วิศวกรรม
 วันที่ 21 เดือน มกราคม พ.ศ. 2544



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับ
ไมโครคอนโทรลเลอร์ MCS-51

DIGITAL SYSTEM DESIGN FOR DEMONSTRATION
BY FPGAs DEVICE WITH MCS-51 MICROCONTROLLER



นางสาวกรรณา เข้มพราย

นายสุทธิพงษ์ จันทะรัง

นางสาวสุรีย์พร วงษ์คำหาญ

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

ปี พ.ศ.

ที่ ๒๕๔๓

๒๕๔๖

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๔๓

เลขหมู่.....

เลขทะเบียน..... 40170

วัน, เดือน, ปี 17 ส.ค. 2544

b. 1109๙๖๘๘
i.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์
MCS-51
Digital System Design for Demonstration by FPGAs Device with MCS-51
Microcontroller

วัตถุประสงค์

1. เพื่อทำการศึกษาลักษณะการทำงานของอุปกรณ์ FPGAs, ไมโครคอนโทรลเลอร์ MCS-51, วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล และวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
2. เพื่อทำการออกแบบ ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 และใบงานการทดลอง
3. เพื่อทำการสร้าง ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 และใบงานการทดลอง
4. เพื่อทำการทดสอบประสิทธิภาพการทำงานของ ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51
5. เพื่อที่จะสามารถนำเอาชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ไปใช้ในการสอนวิชาออกแบบดิจิทัล

ประโยชน์ที่คาดว่าจะได้รับ

1. รู้และเข้าใจหลักการทำงานของอุปกรณ์ FPGAs, ไมโครคอนโทรลเลอร์ MCS-51, วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล และวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
2. ได้ทำการออกแบบ ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51
3. ได้เครื่องต้นแบบของชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 และใบงานการทดลอง
4. ได้ผลการทดสอบประสิทธิภาพชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ได้นำชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์FPGAsร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ไปใช้ในการสอนวิชาออกแบบดิจิทัล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51	
นักศึกษา	นางสาวกรรณา	แย้มพราย
	นายสุทธิพงษ์	- จันทะรัง
	นางสาวสุรีย์พร	วงษ์คำหาญ
อาจารย์ที่ปรึกษา	ผศ.ดร.ธีระพล	เทพหัสติน ณ อยุธยา
อาจารย์ที่ปรึกษาร่วม	อาจารย์กิติพงศ์	มะโน
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์	
ปีการศึกษา	2543	

บทคัดย่อ

ปฏิญานิพนธ์นี้เป็นการนำเสนอชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ซึ่งสามารถออกแบบวงจรดิจิทัลโดยใช้ภาษา VHDL ในการจำลองการทำงานของวงจรร่วมกับชุดฝึก ซึ่งชุดฝึกจะประกอบด้วยวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก, วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51, วงจรสำหรับควบคุมการทำงานของ FPGAs, วงจรดีเบาส์วิตซ์, วงจรกำเนิดสัญญาณพัลส์, วงจรจ่ายแรงดัน และวงจรภาคแสดงผล มีใบงานประกอบการทดลอง จำนวน 10 ใบงาน ซึ่งผลการทดลองใช้ชุดฝึกดังกล่าวปรากฏว่าให้ผลตามจุดประสงค์

II

Thesis Title	Digital System Design for Demonstration by FPGAs Device with MCS-51 Microcontroller	
Students	Miss Karuna	Yampray
	Mr.Sutthipong	Juntarung
	Miss Sureeporn	Wongkhamhan
Advisor	Assist. Prof. Dr.Threrapon	Thephasadin na ayuthya
Co-Advisor	Mr.Kitipong	Mano
Education Level	Bachelor of Science in Industrial Education	
Program in	Electronics and Computer	
Academic Year	2000	

ABSTRACT

This thesis presents the Digital system design for demonstration by FPGAs device with MCS-51 microcontroller. This system is designed and controlled by VHDL language for demonstration. This demonstrator consists of analog to digital converter circuit, digital to analog converter circuit, the circuit for control MCS-51 microcontroller, the circuit for control FPGAs, debouce switch circuit, pulse generation circuit, display circuit and voltage distribution circuit.

Moreover the demonstration also has 10 work sheet for practicing. After trying out this demonstration the result is successful as goods.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงได้ด้วยดี เนื่องจากความช่วยเหลือจากอาจารย์ที่ปรึกษาปริญญานิพนธ์ คือ อาจารย์กิติพงศ์ มะโน และคณาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ท่านได้กรุณาให้คำปรึกษา ข้อเสนอแนะและเอื้อเฟื้อเครื่องมือที่ใช้ในการทำงานต่างๆ เพื่อนๆ ร่วมกลุ่มปริญญานิพนธ์ทุกคนที่ร่วมมือร่วมแรงกันจนงานสำเร็จลุล่วงไปด้วยดี บุคคลอันเป็นที่รักที่คอยเป็นกำลังใจให้เราไม่ทอดยอต่ออุปสรรคในการทำงาน และที่ควรระลึกถึงอย่างยิ่งก็คือคุณพ่อคุณแม่ของพวกเราที่ท่านได้เป็นกำลังใจให้เราเสมอมาตั้งแต่อดีตจนถึงปัจจุบัน



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปริิญาานิพนธ์	1
1.2 จุดความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	4
2.1 กล่าวนำ	4
2.2 หลักการใช้งานภาษาแอสเซมบลี	4
2.2.1 ความหมายของคำสั่ง (Instruction)	4
2.2.2 ข้อเสียของการเขียนภาษาแอสเซมเบลอร์	9
2.3 หลักการและโครงสร้างภายในไมโครคอนโทรลเลอร์ เบอร์ AT89C52	11
2.3.1 คุณสมบัติของไมโครคอนโทรลเลอร์ เบอร์ AT89C52	12
2.3.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52	13
2.4 การออกแบบจากระดับบนลงล่าง (Top – Down Design)	17
2.5 ภาษา VHDL	20
2.5.1 โครงสร้างของภาษา VHDL	23
2.5.2 ลักษณะของการเขียนบรรยายด้วยภาษา VHDL	24
2.6 ทฤษฎี และหลักการของ FPGAs	24
2.6.1 โครงสร้างภายในของอุปกรณ์ FPGAs เบอร์ XC4010E	24
2.6.2 รายละเอียดการใช้งานของอุปกรณ์ FPGAs	27
2.6.3 การใช้ความสามารถของแรมใน FPGAs ตระกูล XC4000	31

สารบัญ (ต่อ)

เรื่อง	หน้า
2.6.4 ข้อควรระวังในการใช้งานอุปกรณ์ FPGAs	32
บทที่ 3 การออกแบบ การสร้างและการทำงาน	33
3.1 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	33
3.2 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	34
3.3 วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51	34
3.4 วงจรสำหรับควบคุมการทำงานของ FPGAs	36
3.5 วงจรดีเบาสวิตช์	38
3.6 วงจรกำเนิดสัญญาณพัลส์	38
3.7 วงจรจ่ายแรงดัน	39
3.8 วงจรภาคแสดงผล	39
3.8.1 วงจรภาคแสดงผลตัวเลขเจ็ดส่วน	39
3.8.2 วงจรภาคแสดงผลแอลอีดี	40
3.9 รายละเอียดอุปกรณ์บนแผ่นวงจรพิมพ์	41
บทที่ 4 การทดลอง และผลการทดลอง	44
4.1 การทดลอง และผลการทดลองของวงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51	44
4.1.1 ลำดับขั้นตอนการทดลอง	44
4.1.2 ผลการทดลอง	45
4.2 การทดลอง และผลการทดลองของวงจรสำหรับควบคุมการทำงานของ FPGAs	45
4.2.1 ลำดับขั้นตอนการทดลอง	45
4.2.2 ผลการทดลอง	46
4.3 การทดลอง และผลการทดลองของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	46
4.3.1 ลำดับขั้นตอนการทดลอง	46
4.3.2 ผลการทดลอง	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.4 การทดลอง และผลการทดลองของวงจรแปลงสัญญาณดิจิทัลเป็น สัญญาณแอนะล็อก	47
4.4.1 ลำดับขั้นการทดลอง	47
4.4.2 ผลการทดลอง	47
4.5 การทดลอง และผลการทดลองของวงจรดีเบาส์วิตช์	48
4.5.1 ลำดับขั้นการทดลอง	48
4.5.2 ผลการทดลอง	48
4.6 การทดลอง และผลการทดลองของวงจรผลิตสัญญาณพัลส์	49
4.6.1 ลำดับขั้นการทดลอง	49
4.6.2 ผลการทดลอง	49
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	50
5.1 บทสรุป	50
5.2 ปัญหา และแนวทางแก้ไข	50
5.3 แนวทางการพัฒนา	51
ภาคผนวก ก เครื่องต้นแบบ	52
ภาคผนวก ข แผ่นวงจรพิมพ์ และลายทองแดง	54
ภาคผนวก ค รายละเอียดของอุปกรณ์	59
ภาคผนวก ง คู่มือการใช้งาน	87
ภาคผนวก จ วงจรรวมของชุดฝึก	91
บรรณานุกรม	93
ประวัติผู้แต่ง	94

สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 2.1 หน้าที่อื่นๆ ของพอร์ต P3 ในเบอร์ AT89C52	15
ตารางที่ 2.2 คุณสมบัติของ FPGAs ตระกูลต่างๆ	25
ตารางที่ 2.3 โหมดต่างๆ ของการคอนฟิกูเรชัน	28
ตารางที่ 2.4 จำนวนของแรมภายใน FPGAs ตระกูล X4000	32
ตารางที่ 3.1 ตำแหน่งการต่อขาตัวเลขเจ็ดส่วน กับขา FPGAs	40
ตารางที่ 3.2 ตำแหน่งการต่อขาแอลอีดี กับขา FPGAs	40
ตารางที่ 3.3 หน้าที่ของจัมป์เปอร์ และสวิตช์ต่างๆ บนชุดฝึก	41
ตารางที่ 4.1 ผลการทดลองของวงจรสำหรับควบคุมการทำงานของ ไมโครคอนโทรลเลอร์ MCS – 51	45
ตารางที่ 4.2 ผลการทดลองของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	47
ตารางที่ 4.3 ผลการทดลองของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	48
ตารางที่ 4.4 ผลการทดลองของวงจรดีไบซ์สวิตช์	48

สารบัญรูป

เรื่อง	หน้า
รูปที่ 2.1 ตำแหน่งและหน้าที่ขาต่างๆ ของไมโครคอนโทรลเลอร์ เบอร์ AT89C52	12
รูปที่ 2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52	16
รูปที่ 2.3 ขั้นตอนของการออกแบบจากระดับบนลงล่าง	18
รูปที่ 2.4 Dataflow Description ของ Multiply – Accumulate Function	20
รูปที่ 2.5 แผนผัง CLB ของตระกูล 4000	26
รูปที่ 2.6 แผนผัง IOBs ของตระกูล 4000	27
รูปที่ 2.7 ลำดับไดอะแกรมในการคอนฟิก	29
รูปที่ 2.8 การต่อใช้งานในลักษณะสเลฟซีเรียล และมาสเตอร์ซีเรียล	30
รูปที่ 2.9 แผนภูมิเวลาการป้อนข้อมูลโปรแกรม	30
รูปที่ 2.10 การต่อใช้งานในลักษณะมาสเตอร์พาราเรล	31
รูปที่ 3.1 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	33
รูปที่ 3.2 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	34
รูปที่ 3.3 วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51	35
รูปที่ 3.4 วงจรสำหรับควบคุมการทำงานของ FPGAs	37
รูปที่ 3.5 วงจรดีเบาส์วิตช์	38
รูปที่ 3.6 วงจรกำเนิดสัญญาณพัลส์	38
รูปที่ 3.7 วงจรจ่ายแรงดัน	39
รูปที่ 3.8 วงจรภาคแสดงผลตัวเลขเจ็ดส่วน	39
รูปที่ 3.9 วงจรภาคแสดงผลแอลอีดี	41
รูปที่ 3.10 การวางตำแหน่งอุปกรณ์บนแผ่นวงจรพิมพ์	43
รูปที่ 4.1 โปรแกรมไฟวิ่งจากบิตสูง (MSB) ไปหาบิตต่ำ (LSB)	44
รูปที่ 4.2 วงจรสำหรับควบคุมการทำงานของ FPGAs	46
รูปที่ 4.3 ผลการทดลองวัดรูปคลื่นจากวงจรผลิตสัญญาณพัลส์	49

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญญาประดิษฐ์

ในการเรียนวิชาดิจิทัล หรือ วิชาออกแบบวงจรดิจิทัล ปัญหาอย่างหนึ่งที่พบอยู่บ่อยครั้งก็คือ การทดลองวงจรหรือระบบที่ออกแบบ ปกติในการทดลองวงจรหรือระบบดิจิทัล ผู้ออกแบบจะต้องต่อวงจรขึ้นมาโดยใช้อุปกรณ์ไอซีชนิดต่างๆ ลงบนแผงทดลอง แล้วจึงทำการทดลอง ทำให้เกิดความยุ่งยาก เมื่อต้องการแก้ไขวงจรหรือระบบ ปัจจุบันได้มีการคิดค้นผลิต อุปกรณ์ FPGAs ขึ้นมาอุปกรณ์นี้สามารถกำหนดตัวเองเป็นวงจร หรือระบบดิจิทัลแบบต่างๆ ได้ง่ายเพียงแค่ทำการกำหนดการทำงานของอุปกรณ์ และยังสามารถแก้ไขวงจรหรือระบบได้ง่าย ทำให้ง่ายต่อการทดลองวงจรหรือระบบจริง แต่อุปกรณ์ FPGAs ยังมีข้อด้อยบางประการ เช่น การคำนวณที่ช้าๆ กังการ ทำงานที่เป็นลำดับบางประการ เพราะถ้าอุปกรณ์ FPGAs จะทำการต่อเกิดภายในให้เป็นวงจรดังกล่าว จะต้องใช้เกตจำนวนมาก ทำให้เกิดความยุ่งยากขึ้น

ดังนั้นโครงการนี้จึงได้นำ ไมโครคอนโทรลเลอร์ MCS-51 เข้ามาช่วยอุปกรณ์ FPGAs ในการคำนวณ อีกทั้งยังได้เพิ่มวงจรต่างๆ ในการทดลองลงบนชุดฝึก เพื่อให้สามารถทดลองวงจรได้อย่างหลากหลายตามความต้องการมากขึ้น

1.2 ขีดความสามารถของโครงการ

โครงการนี้มีขีดความสามารถดังต่อไปนี้

1. สามารถกำหนดการทำงานให้กับ FPGAs ได้สองทางคือ EEPROM และ SERIAL PROM
2. มีวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลและวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ที่สามารถนำมาต่อกับอุปกรณ์ FPGAs หรือนำไปต่อกับไมโครคอนโทรลเลอร์ MCS-51 ได้
3. มีวงจรในการสื่อสารข้อมูลแบบอนุกรม โดยวงจรจะต่ออยู่กับไมโครคอนโทรลเลอร์ MCS-51
4. มีอุปกรณ์แสดงผลที่ต่ออยู่กับอุปกรณ์ FPGAs เป็นตัวเลขเจ็ดส่วน (7-Segment) และแอลอีดี 8 ตัว
5. มีอุปกรณ์ที่ใช้สำหรับป้อนอินพุตให้กับอุปกรณ์ FPGAs เป็นคิปสวิตช์ (Dip Switch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 หลัก

6. มีสวิตช์ที่ใช้ป้อนสัญญาณขอขาขึ้นและ ขอบขาลงให้กับ FPGAs

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญานิพนธ์ฉบับนี้ จะแบ่งออกเป็นบทต่างๆ ทั้งหมด 5 บท เพื่อความสะดวกต่อการศึกษา และทำความเข้าใจ ซึ่งในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 1 บทนำ

ในบทนี้จะกล่าวถึงความสำคัญ และที่มาของปฏิญานิพนธ์ รวมไปถึงขีดความสามารถต่างๆ ของชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ตลอดจนลักษณะโดยทั่วไปของปฏิญานิพนธ์อย่างกว้างๆ เพื่อให้ผู้ที่ศึกษาได้ทราบลักษณะทั่วไป และส่วนประกอบของเนื้อหาสาระที่สำคัญภายในปฏิญานิพนธ์นี้

บทที่ 2 ทฤษฎี และหลักการ

ในบทนี้ ประกอบด้วยเนื้อหาทางทฤษฎี และหลักการต่างๆ ที่เกี่ยวข้อง เช่น หลักการใช้งานภาษาแอสเซมบลี (ASSEMBLY), หลักการและโครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52, การออกแบบจากระดับบนลงล่าง (Top – Down Design), ภาษา VHDL และทฤษฎีและหลักการของ FPGAs

บทที่ 3 การออกแบบ การสร้าง และการทำงาน

ในบทนี้จะกล่าวถึงหลักการออกแบบ และการสร้างวงจรต่างๆ ในชุดฝึก ซึ่งประกอบด้วย วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก, วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51, วงจรสำหรับควบคุมการทำงานของ FPGAs, วงจรดีเบาสวิตช์, วงจรกำเนิดสัญญาณพัลส์, วงจรจ่ายแรงดัน และวงจรภาคแสดงผล ลำดับขั้นของการออกแบบ รวมไปถึงลำดับการสร้างชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ในส่วนต่างๆ

บทที่ 4 การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองวงจรต่างๆ ที่ประกอบอยู่ในชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 คือ วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก, วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51, วงจรสำหรับควบคุมการทำงานของ FPGAs, วงจรดีเบาสวิตช์ และวงจรมกำเนิดสัญญาณพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

บทสรุป ปัญหา แนวทางการแก้ไข และพัฒนา จะเป็นการสรุปข้อบกพร่องต่างๆ ของชุดฝึกที่ได้จัดทำขึ้น การอภิปรายถึงสาเหตุของข้อบกพร่อง และแนวทางการแก้ไขในการปรับปรุงชุดฝึกนี้ ในภาคผนวกจะแสดงรายละเอียดต่างๆ ดังนี้

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข แผ่นวงจรพิมพ์ และลายทองแดง

ภาคผนวก ค รายละเอียดของอุปกรณ์

ภาคผนวก ง คู่มือการใช้งาน

ภาคผนวก จ วงจรรวมของชุดฝึก



บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาของปริยญาณิพนธ์ในบทนี้เป็นทฤษฎีและหลักการที่นำมาใช้ประกอบการสร้างโครงงาน โดยประกอบด้วยหลักการใช้งานภาษาแอสเซมบลี, หลักการและโครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52, การออกแบบจากระดับบนลงล่าง, ภาษา VHDL และทฤษฎีและหลักการของ FPGAs

2.2 หลักการใช้งานภาษาแอสเซมบลี

2.2.1 ความหมายของคำสั่ง (Instruction)

ชุดคำสั่งต่างๆ ของไมโครคอนโทรลเลอร์คือ ชุดการป้อนระบบตัวเลขฐานสองเข้าไปในซีพียู (CPU) เพื่อกำหนดการทำงานในระหว่างวัฏจักรคำสั่งหนึ่งๆ ชุดคำสั่งหนึ่งคือ ตารางฟังก์ชันการทำงานต่างๆ ของ ซีพียู เช่น ทางคณิตศาสตร์ ตรรกศาสตร์ และการเลื่อนบิต เป็นต้น

ตัวคำสั่งหนึ่งจะมีรูปแบบตัวเลขฐานสองบิตจำนวนหนึ่ง ที่ถูกป้อนเข้าสู่ตัว ซีพียู ไปปรากฏ ณ เวลาช่วงหนึ่งที่แน่นอน ซึ่งมันจะถูกแปรเป็นรูปแบบการทำงานตามคำสั่งนั้นๆ ตัวอย่างเช่น เมื่อไมโครคอนโทรลเลอร์ MCS-51 ได้รับรูปแบบตัวเลขฐานสอง 8 บิต คือ 00101000₂ ซึ่งถือว่า ซีพียู ได้แพทชเอาค่านี้อไปเพื่อถอดรหัสเป็นรูปแบบการทำงานดังนี้

“ บวกตัวเลขใน OR กับตัวเลขใน ACC แล้วเก็บผลลัพธ์ที่ ACC ”

ไมโครคอนโทรลเลอร์ก็เหมือนกับระบบไมโครโพรเซสเซอร์อื่นๆ ที่จะรับรู้รูปแบบตัวเลขฐานสองซึ่งอาจเป็นคำสั่งหรือข้อมูลต่างๆ ที่เก็บอยู่ในหน่วยความจำเท่านั้น โดยไม่ใช้อยู่ในรูปของตัวเลขฐานอื่นเช่น ตัวเลขฐานแปด, ตัวเลขฐานสิบ หรือ ตัวเลขฐานสิบหก

การที่ไมโครคอนโทรลเลอร์จะทำงานให้บรรลุงานๆ หนึ่งได้นั้น ก็ต้องอาศัยคำสั่งจำนวนหนึ่งในการสั่งงานให้ไมโครคอนโทรลเลอร์ทำงานบรรลุถึงเป้าหมายที่วางไว้ ที่เราเรียกคำสั่งที่มาต่อๆ กันนี้ว่าโปรแกรม (Program) โดยที่ตัวไมโครคอนโทรลเลอร์จะทำงานตามลำดับคำสั่งที่วางไว้ เช่นการจัดการในเรื่องบวกตัวเลขสองจำนวน โปรแกรมก็จะมีคำสั่งในการรับตัวเลขทั้งสองจำนวนนี้เพื่อมาบวกกัน แล้วจึงนำผลลัพธ์ที่ได้ไปเก็บยังตัวรับผลข้อมูล (Destination) ที่ในคำสั่งกำหนด

ปกติไมโครคอนโทรลเลอร์จะทำงานตามลำดับถ้าไม่มีคำสั่งอะไรที่มาสั่งให้มันข้ามขั้นหรือกระโดดไปทำคำสั่งอื่นที่ไม่เรียงลำดับ โปรแกรมตัวอย่างการบวกเป็นชุดของตัวเลขที่อยู่ในหน่วยความจำภายในที่ 30₁₆ และ 31₁₆ และใส่ผลลัพธ์ที่ได้ที่ตำแหน่งแอดเดรส 32₁₆

```
1110 0101 MOV A,30H
```

```
0011 0000 ADD A,31H
```

```
0011 0001
```

```
1111 0101 MOV 32H,A
```

```
0011 0010
```

โปรแกรมรหัสที่เขียนมาจะเป็นภาษาเครื่อง ถ้าโปรแกรมจัดวางเข้าไปในหน่วยความจำโปรแกรมของ MCS-51 มันก็จะสามารถทำงานบวกตัวเลขดังกล่าวได้โดยตรง

ปัญหาของการเขียนโปรแกรมแบบนี้ที่เป็นภาษาเครื่องซึ่งเป็นตัวเลขฐานสองจะมีความยุ่งยากมาก ปัญหาที่พบ คือ

- 1) จะมีความยุ่งยากที่จะทำความเข้าใจโปรแกรมหรือหาข้อผิดพลาดและแก้ไขได้ยาก เพราะตัวเลขจะเป็นตัวเลข 1 กับ 0 เมื่อมองนานๆ ก็จะสับสนและลำดับตัวเลขไม่ถูก
- 2) การป้อนโปรแกรมจะทำได้ช้าเพราะเราต้องป้อนทีละบิตหรือไบต์
- 3) โปรแกรมจะไม่มีคำอธิบายการทำงานของงานชนิดหนึ่งๆ ที่เราต้องการที่จะสั่งงานให้ไมโครคอนโทรลเลอร์ทำงาน
- 4) โปรแกรมที่ยาวจะทำให้เมื่อยล้าในการเขียนได้
- 5) ผู้เขียนโปรแกรมจะต้องใช้ความระมัดระวังอย่างมากในการเขียน และเมื่อผิดพลาดจะหาที่ผิดได้ลำบาก

ตัวอย่างต่อไปนี้เป็นารแสดงถึงข้อเสียที่เกิดขึ้นในการเขียนโปรแกรมภาษาเครื่อง

โปรแกรมตัวอย่างบวกเมื่อเป็นภาษาเครื่องที่ผิดพลาดจะเขียนดังนี้

```
1110 01012
```

```
0011 00002
```

```
0100 01012
```

```
0011 00012
```

```
1111 00102
```

ให้ลองหาที่ผิดดู โปรแกรมตามข้างบนเป็นโปรแกรมภาษาเครื่องแต่เมื่อดูจะไม่คุ้นเคย และจะต้องใช้เวลานานในการค้นหาที่ผิดและถ้าเกิดกับโปรแกรมใหญ่ก็จะเพิ่มความยุ่งยากและสับสนในตัวโปรแกรมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าลองหาวิธีที่ดีกว่านี้ด้วยการใช้ระบบตัวเลขแทน เช่นเปลี่ยนเป็นระบบตัวเลขฐานสิบ, ฐานแปดหรือฐานสิบหกเราจะพบระบบตัวเลขฐานสิบหกจะใช้แทนตัวเลขฐานสองได้เหมาะสมที่สุด เพราะการจัดกลุ่มตัวเลขฐานสองของหลักตัวเลขฐานสิบหกแต่ละหลักจะลงตัวพอดีฉะนั้นจากตัวอย่างข้างบนจะเขียนใหม่ได้เป็น

$$E\ 5_{16}$$

$$3\ 0_{16}$$

$$4\ 5_{16}$$

$$3\ 1_{16}$$

$$F\ 5_{16}$$

$$3\ 2_{16}$$

ซึ่งจะทำให้ดูตัวเลขสบายตาขึ้นเพราะจำนวนน้อยลง ฉะนั้นการเปรียบเทียบหาข้อผิดพลาดที่ป้อนตัวเลขเข้ามาใครก็จะตรวจสอบและแก้ไขได้ง่ายขึ้น

$$E\ 5_{16}$$

$$3\ 0_{16}$$

$$2\ 5_{16}$$

$$3\ 1_{16}$$

$$F\ 5_{16}$$

$$3\ 2_{16}$$

จะเกิดอะไรขึ้นกับโปรแกรมตัวเลขฐานสิบหกเพราะตัวไมโครจะเข้าใจรหัสคำสั่งตัวเลขฐานสองเท่านั้น ดังนั้นจึงต้องมีโปรแกรมแปลงระบบตัวเลขจากฐานสิบหกเป็นฐานสองเสียก่อนที่จะป้อนเข้าหน่วยความจำโปรแกรมได้ซึ่งจะเรียกว่า Hexadecimal Loader แต่ถ้าเขียนในรูปของตัวเลขฐานสองอยู่แล้วโปรแกรมตัวนี้ก็ไม่จำเป็นต้องใช้

แม้ว่าการใช้ระบบตัวเลขฐานสิบหกเข้าแทนตัวเลขโปรแกรมภาษาเครื่องแล้วก็ตามตัวเลขที่ฐานสิบหกทุกตัวก็จะดูคล้ายกันหมดคือ จับตัวเลขชุดละ 2 หลักเป็น 8 บิต ตามความสามารถของไมโครคอนโทรลเลอร์ แต่จะไม่ทราบเลยว่ารหัสตัวเลขตัวไหนเป็นรหัสคำสั่ง หรือเป็นข้อมูลจริงหรือเป็นแอดเดรสข้อมูลซึ่งรหัสแต่ละชุดของแต่ละตัวไมโครก็จะมี ความแตกต่างกัน ทำให้ต้องมีการอธิบายความหมายของรหัสแต่ละตัวอย่างมากมาย

ดังนั้นจึงมีการปรับปรุงการเขียนโปรแกรมให้ชัดเจนยิ่งขึ้น ด้วยการให้มีชื่อแทนแต่ละรหัสคำสั่ง ชื่อรหัสคำสั่งนี้จะถูกเรียกว่านิวมอนิก (Mnemonic) คำสั่งนิวมอนิกนี้จะต้องมีการอธิบายมาให้ชัดเจนด้วย ดังนั้นผู้ผลิตไมโครคอนโทรลเลอร์ต่าง ๆ ก็จะมีคำอธิบายคำสั่งนิวมอนิกสำหรับชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งหนึ่ง ๆ ของตัวไมโครคอนโทรลเลอร์ตระกูลหนึ่งซึ่งจะถือเป็นมาตรฐาน สำหรับผู้ที่ต้องการจะนำไมโครคอนโทรลเลอร์นั้นมาใช้ก็จะต้องใช้ชุดคำสั่งนิวมอนิกนั้นด้วย ซึ่งส่วนมากก็จะมีมาพร้อมกับรายละเอียดของอุปกรณ์หรือคู่มือการใช้งานของไมโครคอนโทรลเลอร์นั้น เช่น MCS - 51 ไมโครคอนโทรลเลอร์นี้เป็นของอินเทล ดังนั้นก็จะค้นดูได้จากหนังสือของบริษัทอินเทลที่ได้ผลิตไมโครคอนโทรลเลอร์ตัวนี้คือหนังสืออ้างอิงที่ 3 ในบทบรรณานุกรมเป็นต้น

ถ้าเราใช้คำสั่งและนิวมอนิกมาตรฐานของ MCS-51 แล้ว โปรแกรมการรวมตัวเลขของเรา ก็จะเขียนใหม่ได้เป็น

```
MOV A,30H
```

```
ADD A,31H
```

```
MOV 32H,A
```

โปรแกรมก็จะดูง่ายขึ้น คำย่อของนิวมอนิก ก็จะแสดงความหมายของการทำงานได้ชัดเจน เช่น MOV หรือ MOVE คือการย้ายข้อมูล ในส่วนของ A,30H ก็จะแสดงความหมายของแหล่งข้อมูล (Source) คือ ตำแหน่งความจำภายใน 30H ซึ่งจะอยู่ทางขวาของเครื่องหมาย Comma(,) และ A เป็นแอกคูมิวเลเตอร์ที่เป็นตัวรับผลข้อมูล ซึ่งอยู่ทางซ้ายของเครื่องหมายComma โปรแกรมที่ให้มานี้จะเป็นรูปแบบคำสั่งโปรแกรมภาษาแอสเซมบลีแบบง่าย ๆ

```
mnemonic <destination> , <source>
```

อย่างไรก็ตามก็เกิดมีคำถามว่าเราจะใส่โปรแกรมภาษาแอสเซมบลีลงไปในระบบไมโครได้อย่างไร นั่นคือเราจะแปลโปรแกรมให้อยู่ในรูปของตัวเลขฐานสองหรือภาษาเครื่องหรือจะเป็นตัวเลขฐานสิบหกก็ได้ เราสามารถที่จะแปลโปรแกรมภาษาแอสเซมบลีด้วยมือ โดยแปลคำสั่งจากการเปิดตารางดู ก็จะได้โปรแกรมหงี้

ชื่อคำสั่ง	รหัสฐานสิบหก
MOV A, 30H	E5 30
ADD A, 31H	25 31
MOV 32H,A	D5 22

การแปลรหัสเครื่องในลักษณะนี้จะเป็นวิธีที่ทำได้ยากและเสียเวลา ถ้าหากโปรแกรมนั้นเป็นโปรแกรมยาวเป็นร้อย ๆ คำสั่ง หรือแม้แต่โปรแกรมแบบอื่น ๆ ที่มีคำสั่งโปรแกรมกระโดดก็จะมีสูตรสำหรับการคำนวณจำนวนไบต์ที่จะต้องกระโดดไป ไม่ว่าจะเป็นการกระโดดถอยหลังหรือ

ไปข้างหน้า ดังนั้นจึงไม่เป็นวิธีที่ถูกต้องที่จะใช้คนแปล เพราะจะทำให้เกิดข้อผิดพลาดและจะทำได้
อย่างเชื่องช้า

ดังนั้นจึงได้ใช้วิธีการย้ายเครื่องมือที่สำคัญหรือคอมพิวเตอร์เป็นตัวช่วยแปล โปรแกรมภาษา
แอสเซมบลี คอมพิวเตอร์จะไม่มีข้อผิดพลาดเมื่อแปลเป็นรหัส มันจะรู้ว่าในแต่ละคำสั่งนั้นจะมีคำ
รหัสและอยู่ในรูปแบบการกำหนดเลขที่อยู่อะไร โปรแกรมที่ทำงานลักษณะงานชนิดนี้ให้เรา คือ
การแอสเซมเบลอร์ โดยที่ตัวแอสเซมเบลอร์จะแปลโปรแกรมของผู้ใช้หรือโปรแกรมต้นกำเนิด
(Source Program) ที่เขียนเป็นภาษาแอสเซมบลีกลายเป็นโปรแกรมภาษาเครื่องหรือโปรแกรม
ประสงค์ (Object) ที่ตัวไมโครสามารถทำงานได้ ดังนั้นแอสเซมเบลอร์ คือการแปลงอินพุตที่เป็น
โปรแกรมต้นกำเนิดและได้เอาต์พุตออกมาเป็นโปรแกรมประสงค์

การใช้แอสเซมเบลอร์ก็จะต้องเรียนรู้ถึงข้อกำหนดต่าง ๆ และหลักการใช้งานที่มีอยู่ของมัน
แอสเซมเบลอร์นั้นไม่ใช่เป็นเพียงแต่จะแปลชื่อคำสั่งรีจิสเตอร์นิวมอนิกให้เป็นรหัสตัวเลขฐานสอง
เท่านั้น ตัวแอสเซมเบลอร์จะต้องรวมความสามารถต่าง ๆ ต่อไปนี้เข้าไปด้วยคือ

- 1) ต้องอนุญาตให้ผู้ใช้กำหนดชื่อของตำแหน่งหน่วยความจำอุปกรณ์อินพุต และเอาต์พุต
และลำดับของคำสั่ง
- 2) สามารถทำงานด้านคณิตศาสตร์เป็นส่วนหนึ่งของการแอสเซมบลีได้
- 3) สามารถบอกให้ตัวโหลดเดอร์ (Loader) โปรแกรมสามารถวางโปรแกรมหรือข้อมูลลง
ในส่วนไหนของหน่วยความจำได้แล้วแต่คำสั่ง
- 4) อนุญาตให้ผู้ใช้สามารถกำหนดจองพื้นที่หน่วยความจำ เป็นที่อยู่ชั่วคราวของข้อมูล
หรือใส่คำสั่งสแกนที่ลงในบริเวณหน่วยความจำโปรแกรม
- 5) สามารถให้ข้อสนเทศคำอธิบายโปรแกรมต่าง ๆ ได้ ซึ่งเป็นมาตรฐานที่จะช่วยให้
อธิบายส่วนต่าง ๆ ของโปรแกรมได้เมื่อนำมาใช้ภายหลังจะทำความเข้าใจได้รวดเร็วขึ้น
- 6) อนุญาตให้เขียนเป็นโปรแกรมโมดูลที่สามารถเชื่อมต่อและย้ายตำแหน่งโปรแกรมขณะ
แอสเซมเบลอร์ได้
- 7) อนุญาตให้ผู้ใช้ควบคุมรูปแบบของรายการ โปรแกรมและควบคุมการใช้อุปกรณ์อินพุต
เอาต์พุต
- 8) การใช้สัญลักษณ์ในตัวแอสเซมเบลอร์ที่เป็นตัวสัญลักษณ์ ช่วยให้เราใช้เป็นตัวอ้างอิง
ข้ามไฟล์หรือโมดูลกันได้ (Cross Reference) ทั้งยังสามารถสร้างเป็นแมโคร และควบคุมการ
แอสเซมเบลอร์จะเกิดขึ้นได้ด้วยชื่อแม่ต่าง ๆ ตลอดถึงการรายงานข้อผิดพลาดต่าง ๆ ที่เกิดขึ้น

2.2.2 ข้อเสียของการเขียนภาษาแอสเซมบลีย์

ตัวแอสเซมบลีย์จะไม่สามารถแก้ปัญหาทั้งหมดของการโปรแกรม ในปัญหาหนึ่งก็คือจะต้องเขียนโปรแกรมหลายชุดคำสั่งจึงจะให้ไมโครทำงานลักษณะงานหนึ่ง เช่น การส่งข้อมูลซึ่งออกทางภาคแสดงผล การตรวจสัญญาณแอนะล็อกว่าจะเกิดระดับหรือการรับข้อมูลจากการกดคีย์ เหล่านี้เป็นต้น จะไม่สามารถเขียนหรือใช้คำสั่งระบุในหนึ่งคำสั่งของภาษาแอสเซมบลีย์ได้ แต่จะต้องเขียนด้วยกลุ่มคำสั่งเพื่อให้ได้งานหนึ่ง ๆ อย่างต่อเนื่องตามลำดับงานที่เกิดขึ้นในไมโครนั้น ๆ ได้

ดังนั้นการเขียนภาษาแอสเซมบลีย์จึงจำเป็นจะต้องศึกษาถึงรายละเอียดของไมโครตัวนั้นอย่างลึกซึ้งเสียก่อนจึงจะเขียนได้ โดยเฉพาะการศึกษาถึงโครงสร้างสถาปัตยกรรมและการจัดการหน่วยความจำของไมโครนั้น ๆ ตลอดจนถึงการเข้าถึงข้อมูลด้วยการใช้โหมดการกำหนดเลขที่อยู่ต่าง ๆ ตามความสามารถของชุดคำสั่งของไมโครนั้น ๆ

ข้อเสียอีกประการของการโปรแกรมภาษาแอสเซมบลีย์ก็คือ เมื่อใช้กับไมโครโปรเซสเซอร์ตระกูลหนึ่งแล้วจะไม่สามารถนำไปแปลรหัสหรือแอสเซมบลีย์ไมโครตระกูลอื่น ๆ ได้ นั่นก็หมายความว่าโปรแกรมภาษาแอสเซมบลีย์ที่เขียนขึ้นมาใช้ได้กับไมโครตระกูลหนึ่ง เท่านั้น

ซึ่งจะแตกต่างกับภาษาชั้นสูงที่เขียนขึ้นมาแล้ว ก็จะสามารถที่จะนำโปรแกรมที่เขียนมาใช้กับไมโครต่าง ๆ ได้ เพียงแต่ตัวแปลเป็นภาษารหัสเครื่องนั้น (หรือ Compiler) จะแตกต่างกันออกไปขึ้นกับการใช้ไมโครว่าเป็นตระกูลไหน เพื่อที่จะแปลรหัสได้ตรงกับไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ที่ใช้

ภาษาชั้นสูงที่ใช้งานสำหรับไมโครคอนโทรลเลอร์นั้นส่วนใหญ่จะใช้ภาษา C หรือ PL/M (Program Language Machine) ซึ่งจะมีความสามารถที่จะเขียนกำหนดถึงฮาร์ดแวร์โครงสร้างของไมโครได้มากที่สุดการใช้ภาษาชั้นสูงนอกจากจะมีข้อดีที่สามารถพอร์ตเข้าไปไมโครตระกูลไหนก็ได้แล้วยังมีความสะดวกในการเขียนคำสั่งตามอัลกอริทึมได้ง่ายและสั้นกว่าและมีSyntaxที่มาตรฐานข้อเสียในการใช้ภาษาชั้นสูงนั้น เมื่อคอมไพล์แล้วโปรแกรมประสงค์ที่ได้จะมีขนาดใหญ่กว่าหรือใช้พื้นที่ในหน่วยความจำเป็นสองเท่าตัว การใช้สัญลักษณ์ตัวแปรต่าง ๆ ในเนื้อที่หน่วยความจำของการกำหนดค่าก็จะใช้เนื้อที่มากกว่าเช่นกัน เนื่องจากการใช้โปรแกรมชั้นสูงทำให้โปรแกรมมีขนาดใหญ่ ก็จะทำให้การทำงานแบบ Real Time ของการควบคุมประยุกต์การใช้งานจะทำได้ค่อนข้างช้าหรือโปรแกรมไม่ได้ทันกับงานที่ต้องการควบคุม

จากข้อดีข้อเสียที่ระหว่างภาษาแอสเซมบลีย์กับภาษาชั้นสูง ดังนั้นเราจึงสามารถที่จะใช้ข้อดีข้อเสียในการเลือกใช้ภาษาใดภาษาหนึ่งในการพัฒนาสร้างโปรแกรมของเราได้

ในเรื่องการเขียนโปรแกรมสำหรับไมโครคอนโทรลเลอร์นั้น มีเหตุผลที่เราเลือกใช้ภาษาแอสเซมบลีย์เพราะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) ขนาดของโปรแกรมจะเขียนให้มีขนาดสั้นได้
- 2) เหมาะสำหรับการโปรแกรมภายในตัวชิป MCS-51 ที่การเข้าถึงข้อมูลแรมภายใน ด้วยตัวแปรต่าง ๆ ได้จำกัด ทำให้เราสามารถออกแบบใช้ชิปเดียวก็พอ กรณีที่ต้องการเนื้อที่น้อย
- 3) ถ้าต้องการใช้ควบคุมในเรื่องของ Real-Time การใช้ภาษาแอสเซมบลีจะเหมาะที่สุด เพราะสามารถที่จะเขียนโปรแกรมควบคุมข้อมูลเข้าถึงโครงสร้างสถาปัตยกรรมของไมโครได้โดยตรงและใช้ขนาดโปรแกรมได้สั้นกว่าและจำกัดเวลาการทำงานโปรแกรมได้เร็วขึ้น
- 4) การเข้าใจภาษาแอสเซมบลีจะช่วยลดข้อเสียบางประการของการใช้ภาษาสูงได้
- 5) การใช้งานจำนวนมากยังต้องการประสิทธิภาพของการโปรแกรมด้วยภาษาแอสเซมบลี
- 6) ช่วยทำให้เข้าถึงฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ได้อย่างลึกซึ้ง ที่สามารถจะนำความรู้นี้ไปประยุกต์ในการที่จะสร้างโครงงานใช้งานด้วยไมโครคอนโทรลเลอร์ได้อย่างรวดเร็วและมีประสิทธิภาพ

โปรแกรมแอสเซมเบลอร์สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ผู้เขียนได้พิจารณาความสามารถของซอฟต์แวร์หลาย ๆ ตัวแล้ว และเมื่อพิจารณาถึงคุณสมบัติของโปรแกรมด้วยการศึกษาถึงคุณสมบัติของ MCS-51 MACRO ASSEMBLER ของบริษัทอินเทลคอร์เปอร์เรชั่น ที่มีดังต่อไปนี้

- 1) ใช้ได้กับโปรแกรมตระกูล MCS-51 ในระบบพัฒนา
- 2) ให้สัญลักษณ์ในการเข้าถึงระบบฮาร์ดแวร์ MCS-51
- 3) สร้างไฟล์ประสงค์ ไฟล์รายการ และวินิจัยข้อผิดพลาด
- 4) ไฟล์ประสงค์สามารถเชื่อมต่อดำเนินรวมไฟล์อื่นได้
- 5) แบ่งส่วนซอฟต์แวร์สำหรับการใช้เลขที่อยู่ต่าง ๆ และความสามารถในการโยกย้ายข้อมูลต่าง ๆ
- 6) สัญลักษณ์การแอสเซมเบลอร์รองรับ ตารางสัญลักษณ์ การอ้างอิงย้ายไฟล์
- 7) ความสามารถการเขียนแมคโคร และการแอสเซมบลีแบบมีข้อแม้

MCS-51 MACRO ASSEMBLER แพลตฟอร์มต่าง ๆ ของโมดูล ภาษาแอสเซมบลี ให้เป็นโมดูลรหัสที่สามารถเชื่อมต่อกับตำแหน่งได้นิวมอนิกภาษาแอสเซมบลีเป็นตัวโปรแกรมที่อ่านได้ง่ายกว่าและมากกว่า คำสั่งรหัสตัวเลขฐานสองหรือฐานสิบหก ที่เป็นภาษาเครื่องโดยการให้ผู้เขียนสามารถโปรแกรมใช้ชื่อสัญลักษณ์ในหน่วยความจำที่โยกย้ายที่มากกว่าเป็นตำแหน่งสัมบูรณ์ การออกแบบซอฟต์แวร์และการหาแก้ไขข้อผิดพลาดจะถูกทำได้ง่ายรวดเร็วขึ้น มั่นใจขึ้น

การใช้ประโยชน์สูงสุดในการโปรแกรมคือ การใช้ความสามารถของการเขียนโปรแกรมแบบโมดูลที่มีความสามารถในการเชื่อมต่อกัน และย้ายตำแหน่งให้โดยอัตโนมัติ ทำให้เราสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาโปรแกรมด้วยเทคนิคการเขียนแบบโมดูลแทน การจัดการโปรแกรมแบบสัณฐานนี้ จะช่วยให้เราเข้าใจได้ง่ายขึ้น การซ่อมแซมโปรแกรมและความน่าเชื่อถือของโปรแกรมก็มีมากขึ้น

การแอสเซมบลีรองรับการกำหนดและเรียกใช้แมคโคร โดยวิธีนี้จะเพิ่มความสะดวกในการเขียนโปรแกรมซ้ำ ๆ ด้วยการเขียนเพียงครั้งเดียว ก็เรียกใช้ได้ตลอด ขณะเดียวกันก็มีคุณสมบัติในการใช้แอสเซมบลีแบบมีข้อแม้ ที่ต้องการให้ควบคุมการแอสเซมบลีในส่วนไหนของโปรแกรมได้

การอ้างอิงแบบไฟล์จะแจกแจงอยู่ในรายการสัญลักษณ์และแสดงให้ผู้ใช้เป็นบรรทัด ในการแสดงถึงสัญลักษณ์แต่ละที่ที่ถูกกำหนดและอ้างอิง

MCS-51 MACRO ASSEMBLER มีการใช้สัญลักษณ์ในการเข้าถึงที่จะใช้ประโยชน์ของการกำหนด เลขที่อยู่ได้ทุกรูปแบบตามสถาปัตยกรรมของ MCS-51 คุณสมบัติเหล่านี้รวมถึงการอ้างอิงสัญลักษณ์สำหรับตำแหน่งแบบบิต และไบต์ และสำหรับการใช้งาน 4 บิต สำหรับคณิตศาสตร์แบบตัวเลขแบบบีซีดี (BCD)

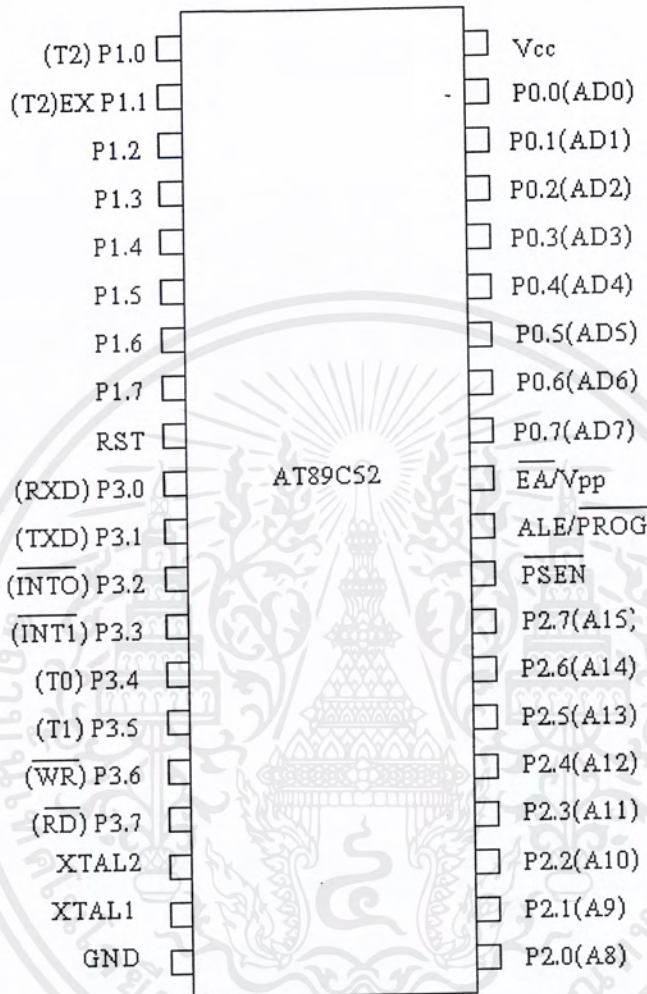
ตัวแอสเซมบลียังสามารถใช้สัญลักษณ์ในการเข้าถึงฮาร์ดแวร์ รีจิสเตอร์ พอร์ต I/O บิต ควบคุมต่าง ๆ และแรมแอดเดรส

MCS-51 MACRO ASSEMBLER สามารถใช้งานได้กับสมาชิกทุกตัวของตระกูล MCS-51

ถ้าโปรแกรม MCS-51 ปรากฏข้อผิดพลาด ตัวแอสเซมบลีจะวินิจฉัยข้อผิดพลาดมาให้รู้ถึงข้อผิดพลาดนี้จะรวมถึงรายการแอสเซมบลีหรือไฟล์อื่น

2.3 หลักการ และโครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต มีอุปกรณ์สนับสนุนประกอบภายในหลายอย่างได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้ง/ตัวนับเวลาอุปกรณ์รับส่งข้อมูลแบบอนุกรม การมีอุปกรณ์สนับสนุนประกอบภายใน ทำให้การใช้งานง่ายและมีประสิทธิภาพเพิ่มมากขึ้น โดยไม่ต้องเชื่อมต่ออุปกรณ์ภายนอกมากเหมือนกับไมโครโพรเซสเซอร์ทั่วไป นอกจากนี้หากต้องการใช้ไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่น เช่น ไอซีเบอร์ 8255 หรือหน่วยความจำภายนอก สามารถนำมาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์อีกด้วย และในรูปที่ 2.1 แสดงตำแหน่งและหน้าที่ขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C52



รูปที่ 2.1 ตำแหน่งและหน้าที่ขาต่างๆ ของไมโครคอนโทรลเลอร์ เบอร์ AT89C52

2.3.1 คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ AT89C52

เป็นชิปขนาด 40 ขามีคุณสมบัติพิเศษหลายอย่างดังนี้

- 1) มีหน่วยความจำโปรแกรมชนิดแฟลชเมโมรี่ (Flash Memory) หรือชนิดที่เขียนและลบได้เร็วขนาด 8 กิโลไบต์ทนต่อการเขียนและลบได้ 1,000 ครั้ง และคงค่าข้อมูลไว้ได้ 10 ปี
- 2) ทำงานที่สัญญาณนาฬิกา 0 – 24 เมกะเฮิร์ตซ์
- 3) ป้องกันการอ่านข้อมูลจากหน่วยความจำโปรแกรม (Program Memory Lock) ได้ 3

ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) มีหน่วยความจำแรม (RAM) ภายในตัวชิป 256 ไบต์
- 5) มีขาอินพุต/เอาต์พุต ที่สามารถ โปรแกรมได้จำนวน 32 ขา
- 6) มีตัวตั้ง/ตัวนับเวลาขนาด 16 บิต จำนวน 3 ตัว
- 7) สามารถรับการอินเทอร์รัพต์ (Interrupt) ได้ 8 แผล่ง
- 8) มีช่องรับส่งข้อมูลแบบอนุกรมที่สามารถโปรแกรมได้

2.3.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52

โครงสร้างภายในพื้นฐานจะมีลักษณะ ดังแสดงในรูปที่ 2.2 ซึ่งขาสัญญาณต่างๆ จะมีหน้าที่ และการใช้งานดังนี้

Vcc สำหรับต่อไฟเลี้ยง 5 โวลต์

GND สำหรับต่อกับกราวด์

XTAL1 เป็นอินพุตของภาคขยายสัญญาณแบบอินเวอร์สของวงจรถอดสัญญาณนาฬิกา

XTAL2 เป็นเอาต์พุตของภาคขยายสัญญาณแบบอินเวอร์สของวงจรถอดสัญญาณนาฬิกา

RST สัญญาณรีเซ็ตไมโครคอนโทรลเลอร์จะรีเซ็ตเมื่อสัญญาณนี้เป็นลอจิก “1”

ALE/ $\overline{\text{PROM}}$ เป็นสัญญาณเอาต์พุตส่งออกไปเป็นพัลส์เพื่อแลตซ์ค่าแอดเดรสไบต์ต่ำที่พอร์ต P0 ขณะที่ติดต่อกับหน่วยความจำภายนอก สัญญาณนี้จะถูกส่งออกไปด้วยอัตราคงที่ คือ 1/6 เท่าของความถี่สัญญาณนาฬิกา ซึ่งสามารถนำไปใช้เป็นสัญญาณนาฬิกาให้กับอุปกรณ์ภายนอกได้ สัญญาณพัลส์นี้จะถูกข้ามไป 1 พัลส์ เมื่อมีการติดต่อกับหน่วยความจำข้อมูลภายนอก และสัญญาณนี้จะใช้เป็นอินพุตควบคุมการโปรแกรมอีพ롬ภายในตัวไมโครคอนโทรลเลอร์ด้วย

$\overline{\text{PSEN}}$ เป็นเอาต์พุตสำหรับส่งสัญญาณสโตรป (พัลส์ต่ำ) เพื่ออ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก เมื่อ ซีพียู อ่านรหัสคำสั่งจากหน่วยความจำภายนอก จะส่งสัญญาณสโตรปออกมา 2 ครั้ง ใน 1 พัลส์สัญญาณนาฬิกา แต่สัญญาณสโตรปทั้ง 2 ครั้ง จะถูกข้ามไปหากเป็นช่วงที่ ซีพียู ติดต่อกับหน่วยความจำภายนอก

$\overline{\text{EA/Vpp}}$ เป็นสัญญาณอินพุตใช้สำหรับควบคุมให้ไมโครคอนโทรลเลอร์เลือกติดต่อกับโปรแกรมที่เก็บอยู่ในหน่วยความจำโปรแกรมหรือโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกตัวไมโครคอนโทรลเลอร์ หากใช้ลอจิก “1” ที่ขานี้จะเป็นการเลือกใช้หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ หากต้องการให้ซีพียูติดต่อกับหน่วยความจำโปรแกรมภายนอก ต้องต่อสัญญาณเข้ากับลอจิก “0” หรือต่อเข้ากับ GND ในกรณีของการโปรแกรมอีพ롬 ภายในของตัวไมโครคอนโทรลเลอร์ จะต่อขานี้เข้ากับแรงดันไฟ 12.75 โวลต์

P0.0-P0.7 เป็นพอร์ตอินพุตเอาต์พุต 2 ทางแบบ Open Drain ขนาด 8 บิตเมื่อใช้เป็นเอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 8 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก “1” ออกไปที่พอร์ตก่อนเพื่อทำให้มีความต้านทานสูง (High Impedance Input) นอกจากนี้พอร์ต P0 จะทำงานอีกหน้าที่หนึ่งคือ เป็นมัลติเพล็กซ์ของสัญญาณแอดเดรสด้านต่ำและสัญญาณข้อมูลในการติดต่อกับหน่วยความจำภายนอก ซึ่งการทำงานในลักษณะนี้จะมีตัวต้านทานพูลอัพ (Pull Up) คร่อมที่ไฟวอกกับขาของพอร์ต อยู่ภายในที่สามารถจ่ายกระแสให้กับอินพุตของ TTL ได้ 8 ตัว นอกจากนี้สองหน้าที่ดังกล่าวแล้ว พอร์ต P0 ยังใช้เป็นตัวรับข้อมูลในช่วงการโปรแกรมอีพรอม และเป็นตัวส่งข้อมูลออกมาในช่วงการตรวจสอบโปรแกรมภายในอีพรอม ซึ่งจะต้องใช้พูลอัพจากภายนอกในขณะที่ทำการตรวจสอบโปรแกรม

P1.0-P1.7 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเป็นเอาต์พุตสามารถต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้เป็นอินพุตต้องเริ่มต้นด้วยการส่งลอจิก “1” ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดพูลอัพภายใน เมื่อมีสัญญาณอินพุตเป็น “0” เข้ามาจะทำให้พอร์ตกจ่ายกระแสออก เนื่องจากการพูลอัพอยู่ภายใน นอกจากนี้พอร์ต P2 ยังสามารถทำหน้าที่รับแอดเดรสด้านต่ำ ในช่วงของการโปรแกรมอีพรอม และช่วงในการตรวจสอบโปรแกรมอีพรอม อีกด้วย

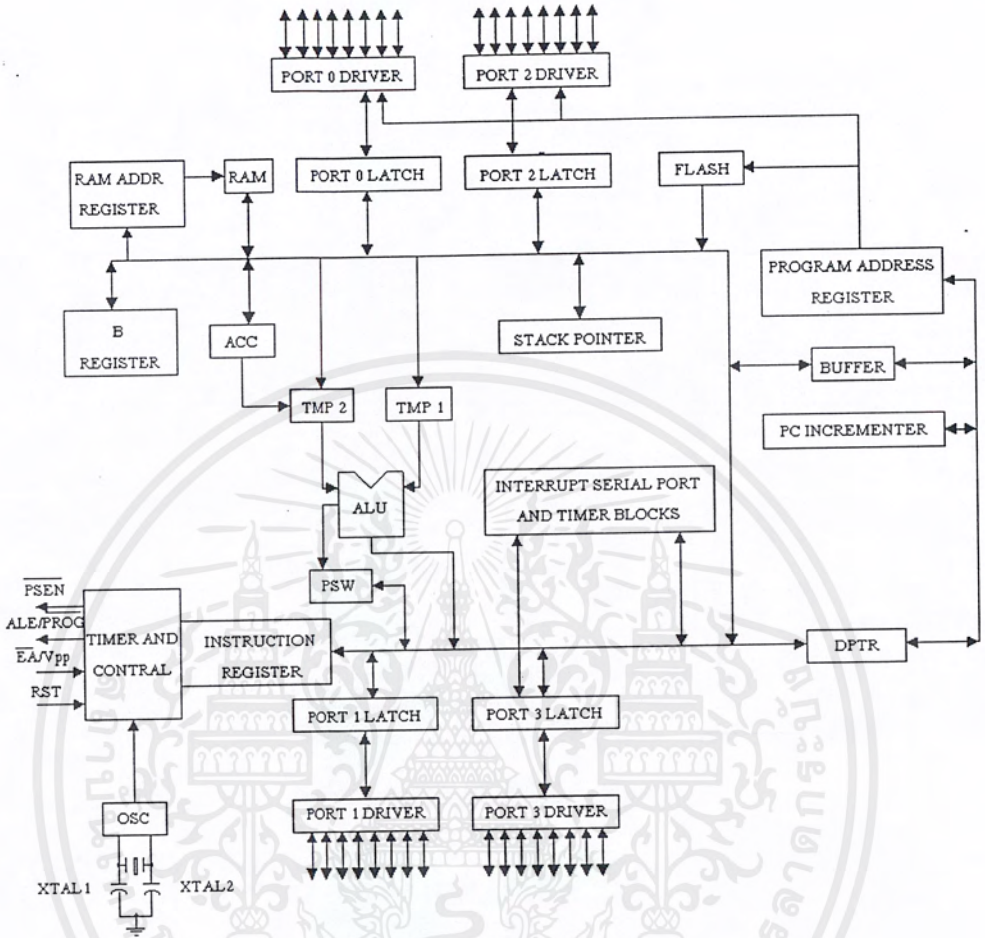
P2.0-P2.7 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเป็นเอาต์พุตจะต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุตต้องส่งค่าลอจิก “1” ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน หากสัญญาณอินพุตเข้ามาเป็น “0” จะทำให้พอร์ต P2 จ่ายกระแสออก เนื่องจากการพูลอัพภายในระหว่างการติดต่อกับโปรแกรมภายนอกหรือการติดต่อกับหน่วยความจำ ข้อมูลภายนอกที่การอ้างตำแหน่งแบบ 16 บิต พอร์ต P2 จะส่งแอดเดรสไบต์สูงออกไป ซึ่งการทำงานในลักษณะนี้จะมีการพูลอัพภายใน ในช่วงของการติดต่อกับหน่วยความจำข้อมูลภายนอกที่ใช้การอ้างตำแหน่งแบบ 8 บิต สัญญาณที่ขาพอร์ต P2 จะมีค่าเท่ากับ P2 ที่อยู่ไนรีจิสเตอร์เฉพาะ (SFR) นอกจากนี้พอร์ต P2 ยังทำหน้าที่รับตำแหน่งไบต์สูง ในช่วงของการโปรแกรมอีพรอม และการตรวจสอบโปรแกรมในอีพรอม อีกด้วย

P3.0-P3.7 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิตที่มีพูลอัพอยู่ภายใน ในกรณีเป็นเอาต์พุตจะต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุตต้องส่งค่าลอจิก “1” ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน หากสัญญาณอินพุตเข้ามาเป็น “0” จะทำให้พอร์ต P3 จ่ายกระแสออก เนื่องจากการพูลอัพภายใน นอกจากนี้พอร์ต P3 ยังทำหน้าที่เป็นสัญญาณอื่นๆ อีกดังแสดงในตารางที่ 2.1

เมื่อต้องการให้พอร์ต P3 ทำหน้าที่เป็นสัญญาณต่างๆ จะต้องเริ่มต้นด้วยการส่งค่าลอจิก “1” ออกไปที่แลตช์ที่พอร์ต P3 ก่อน

ตารางที่ 2.1 หน้าที่อื่นๆ ของพอร์ต P3- ในเบอร์ AT89C52

พอร์ต	สัญญาณ	หน้าที่
P3.0	RXD	อินพุตของพอร์ตอนุกรม
P3.1	TXD	เอาต์พุตของพอร์ตอนุกรม
P3.2	INT0	สัญญาณอินเทอร์รัพต์จากภายนอกตัวที่ 0
P3.3	INT1	สัญญาณอินเทอร์รัพต์จากภายนอกตัวที่ 0
P3.4	T0	อินพุตจากภายนอก ของตัวตั้งเวลา 0
P3.5	T1	อินพุตจากภายนอก ของตัวตั้งเวลา 1
P3.6	WR	สัญญาณการเขียนข้อมูลออกไป ภายนอก
P3.7	RD	สัญญาณการอ่านข้อมูลจากภายนอกเข้ามา



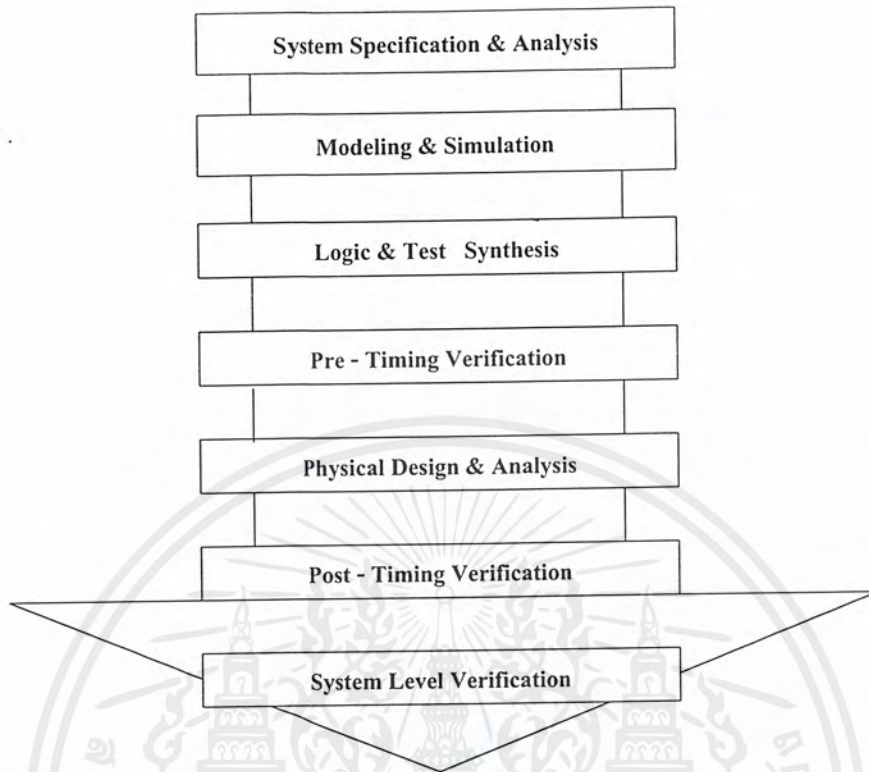
รูปที่ 2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การออกแบบจากระดับบนลงล่าง (Top – Down Design)

การนำภาษา VHDL มาใช้กำหนดและบรรยายพฤติกรรมฟังก์ชันการทำงานของฮาร์ดแวร์ในระบบดิจิทัลนั้น คือความอ่อนตัวของภาษาที่สามารถจำลองการทำงานจากหลักการของรูปแบบ (Simulate Conceptual Designs) แต่ในขณะเดียวกันก็สามารถจำลองการทำงานของฮาร์ดแวร์ ที่ให้รายละเอียดเกี่ยวกับเวลาอย่างถูกต้อง (Timing Based Simulation) และจากโครงสร้างของภาษายังสามารถจำลองการทำงานในรูปของลำดับชั้น (Hierarchy of Simulation Levels) ความสามารถดังกล่าวนี้ จึงช่วยให้การออกแบบสามารถที่จะเขียนรูปแบบบรรยายจากระดับบนสุดของวงจรที่อยู่ในรูปสังเขป (High Level of Abstraction) ลงสู่รายละเอียดในระดับล่างของวงจรได้ เช่น Gate Level เป็นต้น ในเวลานั้นวงการอุตสาหกรรมไมโครอิเล็กทรอนิกส์ ตลอดจนสถาบันวิจัยและศึกษา กำลังพัฒนาภาษาที่ใช้สำหรับการสังเคราะห์วงจรแบบอัตโนมัติ เพื่อลดเวลาในการพัฒนางจรลง ภาษา VHDL จึงถูกนำเข้าพิจารณาในโครงการนี้ด้วย โดยเพิ่มขีดความสามารถของภาษาขึ้น นอกเหนือจากเป็นภาษาที่ใช้สำหรับสังเคราะห์วงจร (Synthesis Language)

การเริ่มต้นด้วยวิธีการเขียนรูปแบบจากแนวความคิดอย่างสังเขป พร้อมทั้งการจำลองการทำงานของรูปแบบที่เขียนขึ้นเพื่อตรวจสอบความถูกต้อง ประกอบกับการกลั่นกรองเพื่อเพิ่มเติมรายละเอียดลงสู่ระบบดิจิทัลที่สมบูรณ์ในรูปของวงจรไฟฟ้าที่แต่ละขั้นนั้นเป็นขบวนการของการออกแบบจากระดับบนลงล่าง การที่เริ่มต้นด้วยการเขียนรูปแบบในระดับบน (Top-Level) ของแนวความคิดอย่างสังเขปนั้น วิศวกรออกแบบสามารถที่จะพัฒนาสภาพแวดล้อมต่าง ๆ เพื่อการตรวจสอบการทำงานของวงจร (Test Environment) ได้ตั้งแต่ในระยะแรก ๆ ของการออกแบบ เพื่อใช้สำหรับตรวจสอบความถูกต้องของแนวความคิดกับสิ่งที่ต้องการจริงหรือ Specification ของงาน ดังนั้นจึงเป็นไปได้ยากที่ในระดับล่างลงมาโดยที่ได้เพิ่มรายละเอียดของรูปแบบให้มากขึ้นตามลำดับจะเกิดข้อผิดพลาด หรือเบี่ยงเบนไปจากจุดประสงค์เดิม เพราะในแต่ละขั้นตอนย่อจะมีการสร้างสภาพแวดล้อมเพื่อการตรวจสอบขึ้นใหม่ โดยอ้างอิงจากระดับที่อยู่สูงกว่าขึ้นไปเสมอ จากรูปที่ 2.4 แสดงให้เห็นขั้นตอนของการออกแบบในลักษณะของการออกแบบจากระดับบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องมาจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้ในหลาย ๆ เทคโนโลยี เช่น Programmable Logic Devices อันได้แก่ PLA, FPGAs หรือ CPLD เป็นต้น นอกจากนี้ยังมี Semi-Custom IC (Gate Array, Standard Cell) และ Full Custom IC



รูปที่ 2.3 ขั้นตอนของการออกแบบจากระดับบนลงล่าง

2.4.1 ขบวนการออกแบบโดยใช้วิธีการออกแบบจากระดับบนลงล่าง

ขั้นตอนการออกแบบจากระดับบนลงล่าง มีรายละเอียดดังนี้

1) System Specification and Analysis ขั้นตอนของการสร้างข้อกำหนดของความต้องการ (Specification) และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2) Modeling and Simulation การเขียนรูปแบบของระบบที่ต้องการจะออกแบบ โดยภาษา VHDL หรือ HDL อื่น ๆ จากแนวความคิดอย่างสังเขปที่ได้ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงานเพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3) Logic and Test Synthesis หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นตอนที่เหมือนกัน คือ Modeling and Simulation จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจร หรือสังเคราะห์ (Synthesis) ในขั้นตอนนี้เอง เทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้นและระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์ และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเอกสารนี้เป็นเอกสารที่ส่งวนไวดำเนินการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Netlist ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้ นอกจากนั้นการผลิตบางเทคโนโลยี อาทิเช่น Gate Array หรือ Standard Cell และ Full Custom IC อาจจะมีความจำเป็นที่ต้องสร้างโครงสร้างของวงจรใหม่หลังจากที่สังเคราะห์ครั้งแรกแล้ว เพื่อความสะดวกต่อการตรวจสอบการทำงานหลังจากที่ผลิตเป็นวงจรต้นแบบแล้ว หรือที่เรียกว่า “Design-For Test” พร้อมทั้งข้อมูลในการตรวจสอบจะถูกกำหนดในขั้นตอนนี้

4) Pre-Timing Verification หลังจากการสังเคราะห์วงจรให้อยู่ในรูป Gate-Level หรือ Netlist แล้วข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชัน (Functional Simulation) แล้ว ยังมีข้อมูลที่เกี่ยวข้องกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี Propagation Delay เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ nanosecond (10^{-9} second) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่าง ๆ จำนวน 10,000 เกตขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรรวมทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูง

5) Physical Design and Analysis คือขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้าที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวมเฉพาะงาน (ASIC)

6) Post-Timing Verification หลังจากที่ได้วงจรจริงมาเรียบร้อยแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิทัล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วย Input และ Output Pad ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

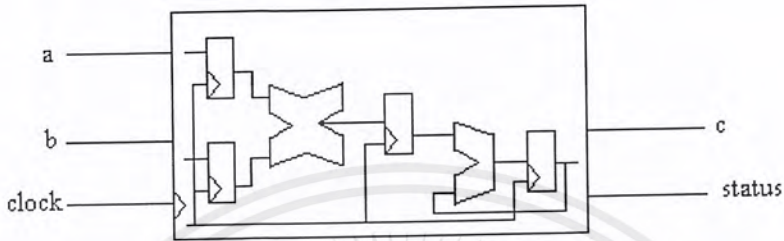
7) System Level Verification หลังจากทีนำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิทัลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้งเป็นการควบคุมคุณภาพของผลิตภัณฑ์

ในตัวอย่างของ Multiply Accumulate Algorithm แสดงให้เห็นขบวนการออกแบบในลักษณะของการออกแบบจากระดับบนลงล่าง ได้โดยการใช้ Multiply Accumulate Function จุดประสงค์แรกก็คือ การเขียนรูปแบบของฟังก์ชันในระดับบนสุดด้วยภาษา VHDL และจำลองการทำงานของแบบที่ได้เพื่อตรวจสอบความถูกต้องซึ่งฟังก์ชันนี้อาจจะเป็น Discrete Fourier Transform (DFT) ก็ได้ หลังจากที่ได้ฟังก์ชันที่ต้องการแล้วและมีชื่อว่า “multiply_accum” สามารถนำไปใช้ได้ในรูปแบบของ Function Call คือ

$$C \leq \text{multiply_accum}(a,b); \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าฟังก์ชันเป็นที่พอใจแล้วในขั้นต่อไปจะเป็นการเริ่มของการแปลงแนวความคิดไปสู่การสร้างวงจรจริง ขบวนการดังกล่าวเกิดขึ้น โดยการเพิ่มรายละเอียดให้กับแนวความคิดเดิม รูปที่ 2.4 แสดงให้เห็นวิธีการบรรยายแบบ Dataflow (Dataflow Description) ของ Multiply Accumulate Function



รูปที่ 2.4 Dataflow Description ของ Mutiply-Accumulate Function

จาก Dataflow Format ที่ได้นี้ ขบวนการต่อไปคือการสร้างวงจรให้อยู่ในรูปของอุปกรณ์พื้นฐานหรือ Gate-Level Implementation ซึ่งวิธีนี้อาจใช้วิธีการสังเคราะห์อัตโนมัติ ผลลัพธ์ที่ได้จากการสังเคราะห์จะเป็นการแสดงภาพวงจรด้วยเกตของฟังก์ชันต่างๆ หรือในรูปของ Netlist และจะเป็นการกำหนดเทคโนโลยีจำเพาะที่จะนำไปผลิตในภายหลัง

คุณสมบัติหลักของภาษา VHDL คือสามารถที่จะบรรยายฮาร์ดแวร์ ได้ในทุกๆ ระดับของภาพรวมทั้งระบบ ฉะนั้น ผู้ออกแบบจึงสามารถใช้เครื่องมือ (ภาษา) เพียงอันเดียวในการบรรยายทั้งระบบ ซึ่งก็เช่นเดียวกันกับเครื่องมือจำลองการทำงาน

2.5 ภาษา VHDL

ภาษา VHDL เริ่มประมาณปี ค.ศ.1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ Department Of Defense (DOD) เห็นว่าอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร เป็นอุปกรณ์ที่ได้รับการพัฒนามาเมื่อประมาณ 20 ปีก่อน ในขณะที่การพัฒนาอุปกรณ์อิเล็กทรอนิกส์เป็นไปอย่างล่าช้า เพราะเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์ ได้รับการพัฒนาไปอย่างรวดเร็ว ดังที่เห็นได้ว่ามีวงจรดิจิทัลอิเล็กทรอนิกส์หลายวงจร ที่แต่เดิมถูกสร้างขึ้นมาจากชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์จำนวนหลายชิ้น ประกอบกันอยู่บนแผงวงจรไฟฟ้า ที่มีขนาดใหญ่ แต่ในปัจจุบัน สามารถที่จะใช้เทคโนโลยีการออกแบบ และผลิตวงจรรวมขนาดใหญ่มาก รวม

อุปกรณ์ต่าง ๆ เหล่านั้นให้อยู่บนชิ้นอุปกรณ์สารกึ่งตัวนำ ที่มีขนาดประมาณ 1-2 ตร.ซม. ได้ ซึ่งเป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้น ตลอดจนความน่าเชื่อถือและความคงทนต่อสภาพแวดล้อมสูง ขณะเดียวกันนั้นในวงการทหารได้มีการนำระบบคอมพิวเตอร์และอิเล็กทรอนิกส์มาใช้ในระบบอาวุธอย่างแพร่หลาย ดังนั้นอุปกรณ์ที่มีใช้อยู่จึงไม่เหมาะสมกับเทคโนโลยีด้านอาวุธของประเทศคู่แข่ง การที่จะเปลี่ยนอุปกรณ์ใหม่เป็นสิ่งที่ต้องใช้งบประมาณมาก และก็จะประสบกับปัญหาเช่นเดิมคือ อุปกรณ์ใหม่ได้รับการพัฒนามานานแล้วเช่นกัน เพราะในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกร และเวลาสำหรับดำเนินการมาก ฉะนั้นทาง DOD จึงตั้งโครงการขึ้นมาเพื่อศึกษา วิธีการที่จะช่วยพัฒนาวงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิทัล ให้สามารถนำไปผลิตได้เร็วขึ้น และโครงการดังกล่าวมีชื่อว่า Very High Speed Integrated Circuits หรือ VHSIC ในระยะแรกนั้นโครงการเป็นความลับของประเทศและอยู่ในความดูแลควบคุมของ United States International Traffic and Arms Regulations (ITAR) ในปี ค.ศ. 1983 ตามคำแนะนำของคณะทำงาน (Woods Hole workshop) ทาง DOD ได้กำหนดความต้องการมาตรฐานของภาษาที่ใช้สำหรับบรรยายพฤติกรรมของ วงจร หรือฮาร์ดแวร์ของระบบสำหรับ โครงการ VHSIC ซึ่งมีสาระสำคัญพอสรุปได้ดังนี้

- 1) ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิทัล และมีคุณสมบัติที่สามารถจะเข้าใจได้ทั้งคนและเครื่องโดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
- 2) สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้ (Project Documentation)
- 3) ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร (Simulation Language) ฉะนั้นภาษาดังกล่าวนี้จึงเป็นภาษาโปรแกรมระดับสูง (High Level Language) เช่นเดียวกับภาษา PASCAL, FORTRAN และ ADA ซึ่งในทางวิศวกรรมการออกแบบฮาร์ดแวร์ เรียกว่า Hardware Description Language หรือ HDL ดังนั้นภาษามาตรฐานนี้จึงมีชื่อว่า VHSIC-HDL หรือ VHDL นั้นเอง

โครงการ DOD ได้มอบหมายให้บริษัท IBM, Texas Instruments และ Intermetrics เป็นผู้ศึกษาและพัฒนา การดำเนินงานได้กระทำไปอย่างต่อเนื่อง และได้ผลเป็นที่น่าพอใจ จนกระทั่งปี ค.ศ. 1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอดเทคโนโลยีทางการทหารออกจากโครงการนี้ ดังนั้น VHDL จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป จนกระทั่งทาง IEEE จึงได้รับภาษานี้เข้ามาศึกษา ประมาณปี ค.ศ. 1987 ได้ยอมรับกำหนดมาตรฐานของภาษา โดยให้ชื่อว่า IEEE 1076-1987 และมีชื่อเรียกว่า VHDL มาตรฐานนี้ได้รับการปรับปรุงจนปัจจุบัน ได้ชื่อว่า IEEE 1076-1993 หรือ VHDL 1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminology and Conventions

การเขียนรูปแบบของระบบดิจิทัลด้วยภาษา VHDL นั้นจะมีศัพท์เทคนิคเฉพาะ และนี่จะอธิบายศัพท์บางคำที่จะต้องพบ

1) ลักษณะของรูปแบบ (Model Styles) เป็นลักษณะของการเขียนรูปแบบด้วยภาษา VHDL สามารถแบ่งได้เป็น

1.1) Behavioral Model : หรือเรียกอีกอย่างได้ว่า Algorithmic description เป็นรูปแบบที่บรรยายพฤติกรรมของระบบดิจิทัล ในส่วนที่บรรยายมีโครงสร้างคล้ายกับภาษาชั้นสูง ทั่วไป เช่น PASCAL หรือ ภาษา C เป็นต้น ในการจำลองการทำงานคำสั่งแต่ละคำสั่ง (Statement) จะถูกประเมินผลเป็นไปตามลำดับ (Sequential) จากบนลงล่าง ยกเว้นในกรณีของคำสั่ง LOOP หรือการเรียกใช้โปรแกรมย่อย รูปแบบนี้จะไม่ให้รายละเอียดที่เกี่ยวกับผลิตหรือโครงสร้างของฮาร์ดแวร์ แต่ในทางตรงข้ามจะให้รายละเอียดเกี่ยวกับความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต

1.2) Dataflow Model : เรียกอีกอย่างหนึ่งได้ว่า “Register Transfer Level” (RTL) เป็นรูปแบบที่ถูกเขียนขึ้นเพื่อจุดประสงค์ที่จะใช้เครื่องมือสำหรับสังเคราะห์วงจรอัตโนมัติรูปแบบลักษณะนี้ส่วนใหญ่จะเป็น Procedural Constructs และ Functional Operators

1.3) Structural Model : เป็นรูปแบบที่แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ต่างๆ ที่ประกอบกันขึ้นเป็นวงจรหรือระบบดิจิทัลและสามารถเรียกอีกอย่างได้ว่า “Netlist Representation” เป็นการเขียนที่แสดงให้เห็นโครงสร้างของฮาร์ดแวร์

1.4) Mixed-Level Model : จากความอ่อนตัวของภาษา VHDL จึงสามารถที่จะเขียนรูปแบบโดยใช้ลักษณะต่างๆ ที่กล่าวมาแล้วข้างต้น บรรยายวงจรหรือระบบดิจิทัลเดียวกันได้ฉะนั้นรูปแบบเช่นนี้จึงมีการเขียนแบบผสม

2) Concurrency : ในภาษา VHDL นั้นชุดคำสั่งแต่ละชุดจะทำงานในเวลาเดียวกันและอิสระต่อกันซึ่งเป็นคุณสมบัติที่เป็นความจริงทางฟิสิกส์ของวงจรรีเลย์ทรอนิกส์ ชุดคำสั่งนี้เรียกว่า “Concurrent Statement” และจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงค่าของสัญญาณ

3) Sequential : นอกจากความสามารถที่จะทำงานแบบ Concurrent แล้ว บางครั้งการเขียนรูปแบบในลักษณะที่บรรยายพฤติกรรมของวงจรมีความจำเป็นที่จะต้องให้ชุดคำสั่งทำงานเป็นลำดับขั้นเรียงกันจากบนลงล่าง อย่างเช่นการเขียนแบบ Behavioral Model เป็นต้น ชุดคำสั่งที่เป็น Sequential นี้จะใช้ในโปรแกรมย่อยและ Process Statement

4) Driver : สัญญาณต่างๆ ใน VHDL นั้นจะถูกควบคุมด้วยตัวขับหรือ “Driver” สัญญาณเหล่านี้จะรับค่าใหม่ (ระดับของสัญญาณ) ได้ด้วยตัวขับนั่นเอง

5) Transaction : การเกิด Transaction กับ Signal นั้นเกิดขึ้นเมื่อมีการกำหนดค่าๆ หนึ่งให้กับสัญญาณนั้น ค่าใหม่ที่สัญญาณได้รับอาจจะไม่มีหรือไม่มีผลทำให้เกิดการเปลี่ยนแปลงของระดับสัญญาณ (event) เช่นการเปลี่ยนจากค่า Logic '0' เป็นค่า Logic '1' เป็นต้น

6) Event : คือการเปลี่ยนระดับค่าของ SIGNAL จากระดับหนึ่งไปสู่ระดับอื่น อย่างเช่นในระบบดิจิทัลการเปลี่ยนจาก Logic '0' เป็น Logic '1' หรือในทางตรงกันข้ามถือว่า SIGNAL นั้นเกิด "Event" ฉะนั้นจะเห็นได้ว่า การที่จะเกิด Event ใดนั้นจะต้องเกิด Transaction ด้วย แต่ในทางตรงข้ามการเกิด Transaction ไม่จำเป็นต้องเกิด Event ทุกครั้ง

7) Sensitivity List : คือรายชื่อของ SIGNAL ต่างๆ ที่มีผลให้เกิดการทำงานของ Concurrent Statement เมื่อเกิด Event ขึ้นกับ SIGNAL ตัวใดตัวหนึ่งหรือหลายตัวพร้อมกันในรายชื่อนั้น

8) Objects : ในภาษา VHDL นั้นคำว่า Object ใช้เขียนเพื่อบ่งบอกถึงองค์ประกอบส่วนหนึ่งของรูปแบบ ซึ่งเปรียบได้เหมือนกับภาษาซีที่มีไว้สำหรับบรรทัดต่างๆ สามารถแบ่งออกได้เป็นสามชั้นด้วยกันคือ

8.1) CONSTANT : ได้แก่ Object ประเภทหนึ่งที่มีค่าเริ่มต้นให้แล้วจะคงค่า นั้นไว้ตลอด ไม่สามารถตัดแปลงหรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศของรูปแบบ

8.2) SIGNAL : หมายถึง Object ประเภทหนึ่งที่สามารถกำหนดค่าที่สัมพันธ์กับเวลาให้ ใต้นั้นหมายความว่า SIGNAL สามารถรับค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง SIGNAL จะรับค่าๆ หนึ่งได้จากตัวขับสัญญาณหรือ Driver ซึ่งตัวขับนี้อาจจะเก็บค่าในอนาคตสำหรับ SIGNAL ไว้ด้วย SIGNAL สามารถประกาศใช้ได้ในส่วนที่เป็นเนื้อหาของ Concurrent Body เท่านั้น ดังนั้น SIGNAL จึงสามารถถูกนำไปใช้ได้ตลอดโครงสร้างของรูปแบบหรือที่เรียกว่า Global Object

8.3) VARIABLE : หรือตัวแปรได้แก่ Object ที่สามารถกำหนดค่าใดๆ ได้และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดการจำลองการทำงาน แต่จะเก็บค่าเพียงค่าเดียวเท่านั้น เนื่องจาก VARIABLE สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น VARIABLE จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้

2.5.1 โครงสร้างของภาษา VHDL

VHDL นั้นประกอบด้วยส่วนต่างๆ ที่สำคัญและเป็นพื้นฐานของการเขียนรูปแบบระบบดิจิทัลที่สำคัญ 4 หน่วยคือ

- 1) Entity Design Unit
- 2) Architecture Design Unit
- 3) Package Design Unit
- 4) Configuration Design Unit

2.5.2 ลักษณะของการเขียนบรรยายด้วยภาษา VHDL

การบรรยายเชิงพฤติกรรม

เป็นการบรรยายลักษณะการเปลี่ยนแปลงของข้อมูลในรูปแบบของอัลกอริทึม สำหรับการคำนวณผลลัพธ์ที่เกิดขึ้นสืบเนื่องมาจากการเปลี่ยนแปลงสถานะของข้อมูลที่เข้ามาโดยไม่คำนึงถึงว่าลักษณะโครงสร้างหรือความสัมพันธ์ของอุปกรณ์ที่อยู่ภายใน

การบรรยายเชิงข้อมูล

เป็นการบรรยายถึงการเคลื่อนไหวของข้อมูลผ่านรีจิสเตอร์และบัฟเฟอร์ของระบบ เป็นระดับขั้นการบรรยายที่อยู่ตรงกลางระหว่างการบรรยายเชิงพฤติกรรมและการบรรยายเชิงโครงสร้าง เครื่องมือที่ใช้ในการควบคุมการเคลื่อนไหวของข้อมูลได้แก่ Conditional Selected และ Guarded

การบรรยายเชิงโครงสร้าง

การบรรยายการทำงานของระบบในเชิงโครงสร้างจะต้องแสดงรายการของอุปกรณ์ทั้งหมดที่ใช้ในระบบและต้องกำหนดการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ด้วย เพราะว่าการบรรยายในระดับนี้เป็นการบรรยายที่ใกล้เคียงลักษณะของฮาร์ดแวร์จริงที่สุด

2.6 ทฤษฎีและหลักการของ FPGAs

2.6.1 โครงสร้างภายในของอุปกรณ์ FPGAs (เบอร์ XC4010E)

FPGAs จัดเป็นวงจรรวมเฉพาะกิจชนิดหนึ่งที่สามารถโปรแกรมเป็นวงจรเชิงเลขใดๆ ก็ได้ เช่นเดียวกับ EPLD ต่างกันที่ EPLD โปรแกรมลงภายในอีพროม และสามารถโปรแกรมใหม่ได้หลังจากนำไปลบด้วยแสง UV แต่ FPGAs โปรแกรมลงบนสแตติกเรมภายในด้วยข้อมูลที่อยู่ภายนอก และสามารถโปรแกรมใหม่ได้โดยการรีเซตด้วยสัญญาณไฟฟ้า นอกจากนั้น FPGAs ยังประหยัดไฟและมีความจุสูง (จำนวนเกตมาก) ได้อีกด้วย

วงจรรวมชนิดที่ใช้ในโครงการนี้ผลิตโดยบริษัทไซลิงค์ (Xilinx) ซึ่งเป็นบริษัทที่ทำการค้นคว้าร่วมกับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นกลุ่มของเกตจำนวน 600-25,000 เกต ดังแสดงในตารางที่ 2.2 การที่ต้องบอกขนาดของวงจรรวมเป็นจำนวนเกตเพราะจะได้รู้ว่าขนาดของวงจรที่ได้ออกแบบไว้สามารถโปรแกรมลงบนวงจรรวม FPGAs ได้หรือไม่

ตารางที่ 2.2 คุณสมบัติของ FPGAs ตระกูลต่าง ๆ

FPGAs	Appr.Gate Count	Max I/OS	Flip-Flops	RAM bits	Available CLBs
XC4002A	2000	64		2048	64
XC4003/4003A	3000	80		3200	100
XC4003H	3000	160		3200	100
XC4004A	4000	960		4608	144
XC4005/4005A	5000	122		6072	196
XC4005H	5000	192		6272	196
XC4006	6000	128		8192	256
XC4008	8000	144		10368	324
XC4010E*	10000	160		12800	400
XC4013	13000	192		18432	576
XC4025	25000	256		32768	1024

หมายเหตุ * หมายถึงเบอร์ที่ใช้ในโครงการชิ้นนี้

FPGAs มีโครงสร้างภายในใกล้เคียงกับสถาปัตยกรรมของเกตอะเรย์ (GAL, Gate Array Logic) มาก สามารถโปรแกรมและลบคอนฟิกูเรชัน (Configuration) ภายในสแตติกแรม (Static RAM) ได้โดยใช้กระแสไฟฟ้า ซึ่งทำการโปรแกรมได้โดยดึงข้อมูลฐานสิบหกมาจากภายนอก เช่น Parallel EPROM หรือ Serial PROM ต่างกับ EPLD, PAL ที่มี EPROM อยู่ในตัวภายใน FPGAs จะจัดเรียงเป็นลอจิกเซลล์ล้อมรอบภายนอกด้วยอินพุตเอาต์พุตเซลล์ FPGAs ตัวแรกที่ผลิตโดยบริษัทไซลิ่งค็อกโบร์ XC2064 (2000 Family) ประกอบด้วยเซลล์เรียงกันเป็นเมตริกซ์ (Matrix) เป็นจำนวน 64 เซลล์ หลังจากนั้นผลิต FPGAs ตระกูล 3000 และ 4000 มีโครงสร้างซับซ้อนขึ้นสามารถเพิ่มจำนวนเกตได้มากขึ้นและดีขึ้น แต่ละเซลล์เรียกว่า CLB (Configurable Logic Block)

ส่วนที่เป็นองค์ประกอบของลอจิก (Configurable Logic Block)

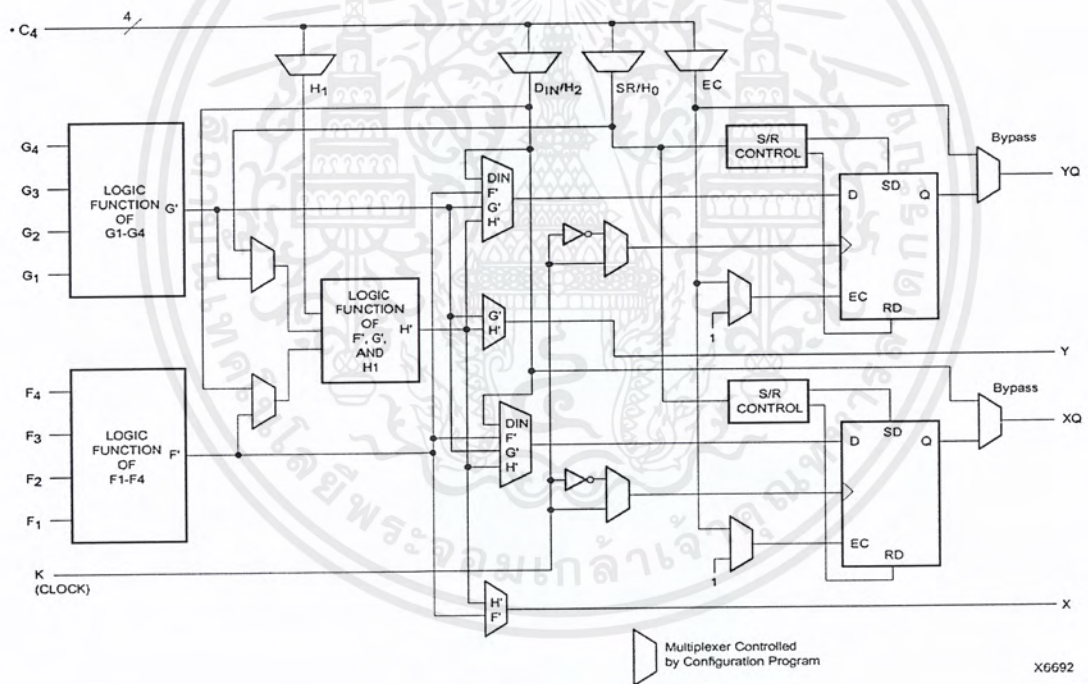
CLB จะจัดเรียงกันเป็นแบบเมตริกซ์แบบอะเรย์ขนาด $M \times N$ การออกแบบนั้นสามารถทำได้โดยการจัดการวาง CLB และต่อเชื่อมขาของ CLB ให้ต่อกัน เราสามารถจัด CLB ให้เชื่อมต่อกันได้โดยการทำได้ด้วยมือหรือให้โปรแกรมที่สนับสนุน FPGAs ทำให้โดยอัตโนมัติ โดยวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของมันเองสำหรับไฟล์ที่ได้จากโปรแกรมเหล่านี้ เราเรียกว่าไฟล์ที่กำหนดการวางอุปกรณ์ (Configuration File) ซึ่งจะบรรจุโครงร่างภายในของ CLB ตามความเหมาะสม อีกด้านหนึ่งไฟล์ที่กำหนดการวางอุปกรณ์นั้นจะเป็นไฟล์กระแสข้อมูล (Bit Stream) ซึ่งสามารถใช้โปรแกรมหน่วยความจำภายในของ FPGAs ได้ สำหรับรูปที่ 2.6 แสดง CLB ของ FPGAs ตระกูล 4000

ส่วนอินพุตและเอาต์พุตของอุปกรณ์ FPGAs

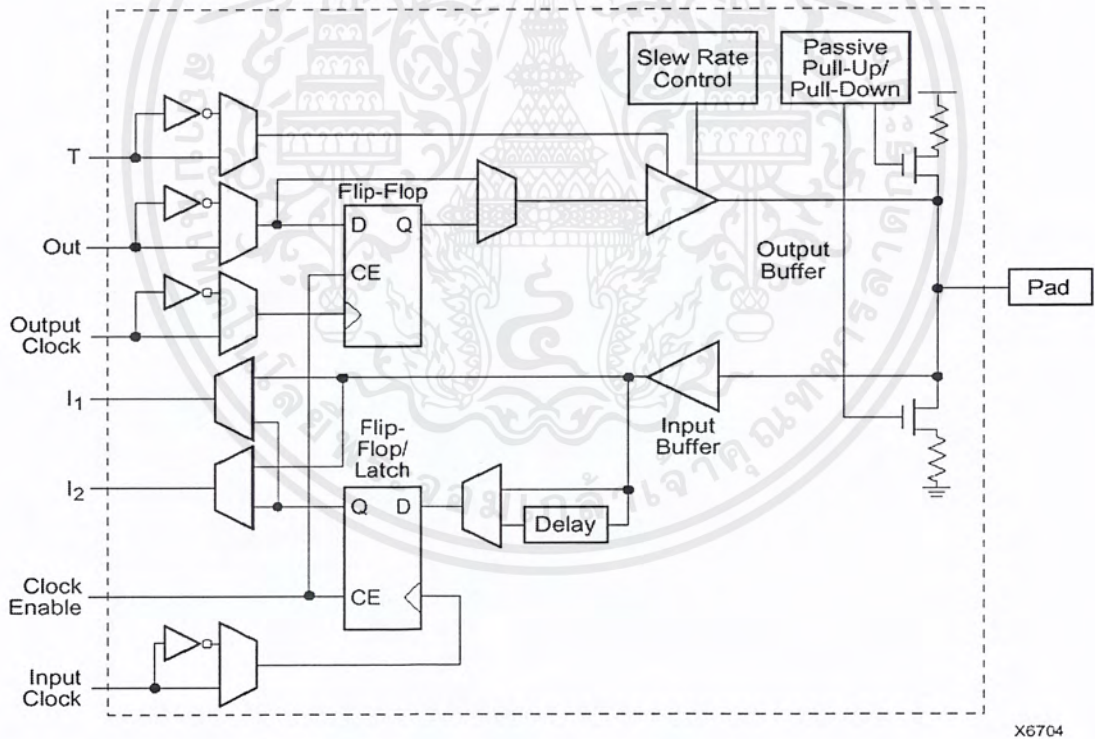
รอบนอกของ FPGAs จะประกอบด้วย IOBs ประมาณ 64 ถึง 144 ตัว ซึ่งขึ้นอยู่กับตระกูลของ FPGAs จะเป็นตัวเชื่อมต่อระหว่างภายในกับภายนอกของวงจรลอจิกของ FPGAs ลักษณะของ IOBs จะมีลักษณะ 2 ทิศทาง สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้ สำหรับรูปที่ 2.6 แสดง IOBs ของ FPGAs ตระกูล 4000



รูปที่ 2.5 แผนผัง CLB ของตระกูล 4000

2.6.2 รายละเอียดการใช้งานของอุปกรณ์ FPGAs

FPGAs สามารถทำงานได้หลายลักษณะ โดยกำหนดได้ที่ขาสัญญาณ M0 M1 M2 ดังแสดงในตารางที่ 2.3 ในลักษณะมาสเตอร์พาราเรล (Master Parallel Mode) รับโปรแกรมคอนฟิกทีละ 1 ไบต์ (Byte) จากหน่วยความจำภายนอกที่เป็นแบบขนาน โดยสามารถรับโปรแกรมคอนฟิก (Cofig) จากแอดเดรส (Address) ต่ำหรือสูงก่อนก็ได้ การต่อลักษณะเพริเฟอรัล (Peripheral) จะรับโปรแกรมคอนฟิกทีละ 1 ไบต์ จากไมโครโปรเซสเซอร์ โดยสามารถโต้ตอบกันได้ว่าพร้อมหรือไม่ที่จะรับข้อมูลต่อไป การต่อลักษณะสเลฟซีเรียล (Slave Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิต จากไมโครโปรเซสเซอร์ตามสัญญาณอินพุต CCLK ส่วนการต่อลักษณะมาสเตอร์ซีเรียล (Master Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิต จากหน่วยความจำภายนอกที่เป็นแบบอนุกรม



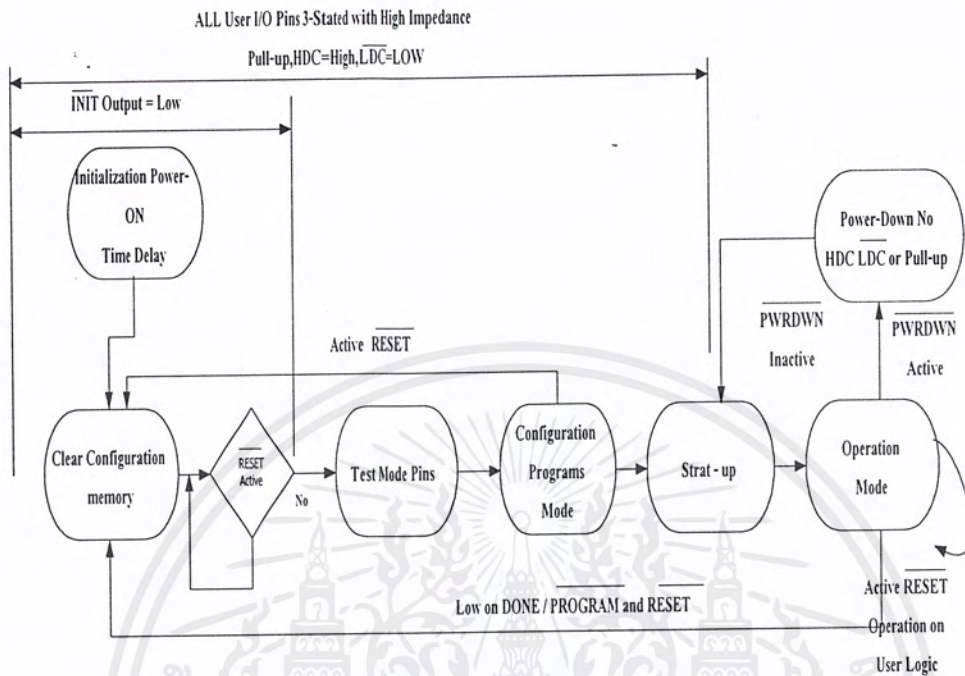
รูปที่ 2.6 แผนผัง IOBs ของตระกูล 4000

ตารางที่ 2.3 โหมดต่าง ๆ ของการคอนฟิกูเรชัน

Mode	M2	M1	M0	CCLK	Date
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 up
Master Parallel down	0	1	1	Input	Byte-Wide,3FFFF down
Peripheral Synchr.	1	0	1	Output	Byte-Wide
Peripheral Asynchr.	0	1	0	---	Byte-Wide
Reserved	0	0	1	---	---

จากความต้องการสร้างให้ใช้กระแสไฟฟ้าต่ำ จากลักษณะการต่อใช้งานทั้ง 5 แบบ จึงมีเพียง 2 แบบเท่านั้นที่เหมาะสม คือ แบบมาสเตอร์ซีเรียลและแบบสเลฟซีเรียล ส่วนแบบมาสเตอร์พาราเรล ต้องใช้ EPROM เบอร์ 27CXXX ซึ่งกินกระแสมากกว่า PROM เบอร์ XC17XXX เหมาะในการทดสอบต้นแบบก่อน เมื่อวงจรต้นแบบทำงานได้ถูกต้องแล้วจึงทำการอัปเดตโปรแกรมลง PROM อีกทีหนึ่งเพราะว่าในแบบพาราเรลนั้น EPROM สามารถโปรแกรมได้ใหม่ต่างกับ PROM ที่โปรแกรมเพียงได้ครั้งเดียว

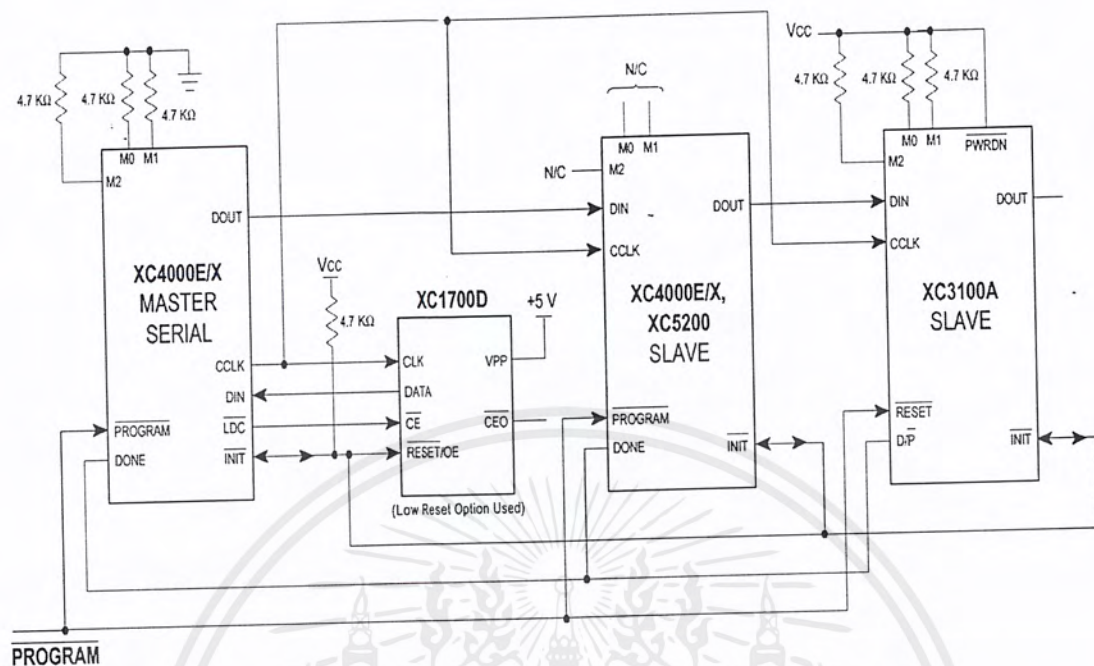
การใช้งาน FPGAs ในการต่อลักษณะสเลฟซีเรียลและมาสเตอร์ซีเรียล เมื่อเริ่มจ่ายไฟเข้าตัว FPGAs จะทำการลบข้อมูลหน่วยความจำที่ใช้ในคอนฟิก (Configuration Memory) ตรวจสอบลักษณะการคอนฟิกว่าเป็นลักษณะใดในตารางที่ 2.3 ว่าเป็นแบบอนุกรมหรือขนาน หลังจากนั้นจะเริ่มทำการโปรแกรมคอนฟิกสัญญาณ DONE/PROGRAM เป็น “0” ซึ่งอยู่ในระหว่างโปรแกรม และเมื่อข้อมูลในคอนฟิกที่รับมาจากภายนอกเต็มหน่วยความจำที่ใช้ในการคอนฟิก และความยาวของข้อมูลตรงกับที่ส่วนหัวของข้อมูลคอนฟิก สัญญาณ DONE/PROGRAM จะเป็น “1” ซึ่งหมายถึงโปรแกรมทำการคอนฟิกเสร็จสิ้น ดังแสดงในรูปที่ 2.7



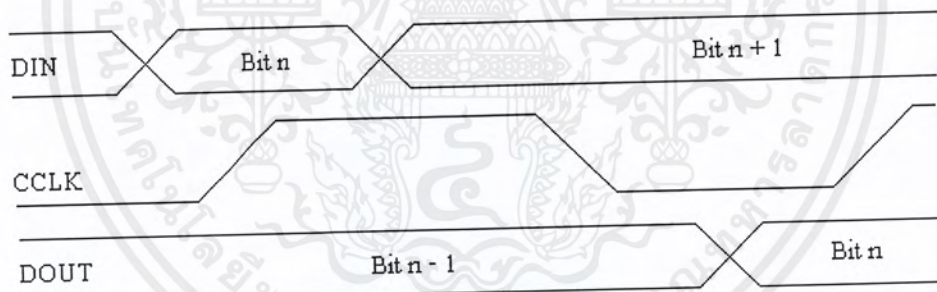
รูปที่ 2.7 ลำดับไดอะแกรมในการคอนฟิก

การใช้งานในลักษณะสเตฟซีเรียล

การต่อใช้งานในลักษณะนี้ เหมาะสมกับวงจรที่ออกแบบมาเพื่อทำงานร่วมกับ ไมโครคอมพิวเตอร์อยู่แล้ว ทั้งนี้เพราะ FPGAs ได้ใช้ความสามารถของไมโครคอมพิวเตอร์ในการเก็บและส่งข้อมูลคอนฟิกให้ เพียงแต่ต้องเขียนโปรแกรมเพื่อส่งโปรแกรมคอนฟิกให้เพิ่ม ลักษณะการต่อในลักษณะนี้เป็นดังรูปที่ 2.8 ซึ่งไมโครคอมพิวเตอร์จะสร้างสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGAs การป้อนโปรแกรมคอนฟิกให้ FPGAs ทำได้โดยต่อสัญญาณ Strobe เข้ากับขา CCLK และพอร์ต DO เข้ากับขา DIN สร้างสัญญาณคล็อกป้อนที่ขา CCLK และป้อนโปรแกรมคอนฟิกแบบอนุกรมเข้าที่ขา DIN ดังแผนภูมิในรูปที่ 2.9



รูปที่ 2.8 การต่อใช้งานในลักษณะสเลฟซีเรียล และมาสเตอร์ซีเรียล



รูปที่ 2.9 แผนภูมิเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในลักษณะสเลฟซีเรียล

การใช้งานในลักษณะมาสเตอร์ซีเรียล

การต่อใช้งานในลักษณะนี้ ส่วนที่เก็บโปรแกรมคอนฟิกจะต่างจากการต่อลักษณะแรก คือ ใช้ PROM เบอร์ XC17XXX เป็นตัวเก็บโปรแกรม ทำให้ไม่ต้องเสียเวลาเขียนโปรแกรมเพื่อทำการคอนฟิก ซึ่งวิธีการอัดโปรแกรมคอนฟิกลง PROM ทำตามขั้นตอนดังนี้คือ เมคบิต (MakeBits) สร้างไฟล์ .BIT จากวงจรที่ออกแบบ และใช้โปรแกรม MakePROM สร้าง Hex ไฟล์แล้วทำการอัดโปรแกรมลง PROM ด้วยอุปกรณ์อัด PROM ที่มาพร้อมกับตัวโปรแกรมของไซลิงค์ PROM XC17XXX จะส่งสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGAs ดังแสดงในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 จำนวนของแรมภายใน FPGAs ตระกูล X4000

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

2.6.4 ข้อควรระวังในการใช้อุปกรณ์ FPGAs

สิ่งที่สำคัญ คืออุปกรณ์ FPGAs ไวต่อความร้อนมาก การบัดกรีโดยหัวแร้งกำลังสูงหรือบัดกรีโดยจี้หัวแร้งที่ขาไอซีเป็นเวลานานจะทำให้ไอซีเสียหายได้ ระยะเวลาในการบัดกรีหนึ่งจุดไม่ควรเกิน 5-10 วินาที ควรใช้ซ็อกเกต (Socket) ไอซีในการประกอบวงจรลงแผ่นวงจรพิมพ์

การป้องกันไอซีจากแรงดันไฟฟ้า ไม่ควรต่อสลับขั้วบวกกับขั้วลบจะทำให้ไอซีเสียหายได้นอกจากนั้น แรงดันของแหล่งจ่ายไฟต้องอยู่ในช่วงที่โรงงานกำหนดมา สำหรับ FPGAs ค่าแรงดันที่ใช้งานอยู่ในช่วง $V_{cc} = 4.75-5.25$ V และแรงดันที่ทนได้อยู่ในช่วง $-0.5 - 7$ V ดังนั้นก่อนป้อนแรงดันควรตรวจเช็คให้แน่ใจก่อน

บทที่ 3

การออกแบบ การสร้างและการทำงาน

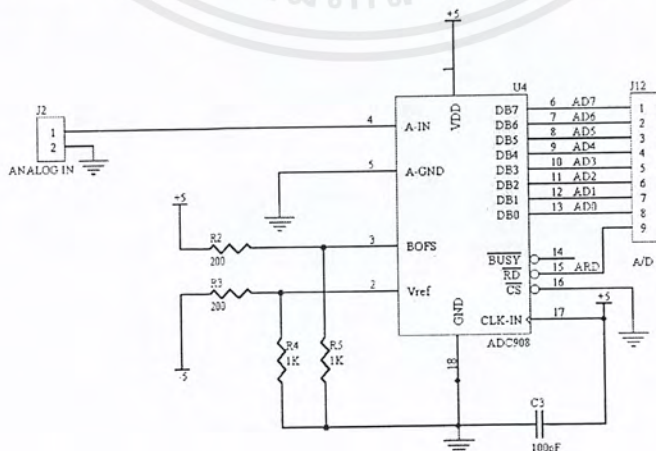
ในบทนี้จุดประสงค์จะกล่าวถึงการสร้างและการออกแบบชุดฝึกออกแบบระบบดิจิทัล โดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ โดยจะประกอบด้วยวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก, วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS - 51, วงจรสำหรับควบคุมการทำงานของ FPGAs, วงจรดีเบาส์วิตช์, วงจรกำเนิดสัญญาณพัลส์, วงจรจ่ายแรงดัน และวงจรภาคแสดงผล

ในการออกแบบวงจรต่างๆ ของชุดฝึกนี้ ได้ใช้โปรแกรม Protel 99SE ในการออกแบบ และสร้างแผ่นวงจรพิมพ์ทั้งหมด

3.1 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

หลักการการทำงานของวงจร

วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ดังแสดงในรูปที่ 3.1 จะใช้ไอซีเบอร์ ADC908 ซึ่งจะ เป็น ไอซี สำหรับแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบ Successive Approximation และให้ผลการแปลงเป็นสัญญาณดิจิทัล 8 บิต โดยการต่อวงจรใช้งานเป็นดังรูปที่ 3.1 จะเห็นได้ว่า วงจรนั้นต่อ C3 ไว้สำหรับสร้างสัญญาณนาฬิกาภายในตัวไอซี ซึ่งทำให้มีช่วงเวลาในการแปลงสัญญาณ (Conversion time) ได้ต่ำสุด 6 ไมโครวินาที ส่วนขา RD ใช้สำหรับการต่อสัญญาณนาฬิกาจากภายนอก ซึ่งสามารถใช้ความถี่ได้สูงถึง 1.35 MHz



รูปที่ 3.1 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

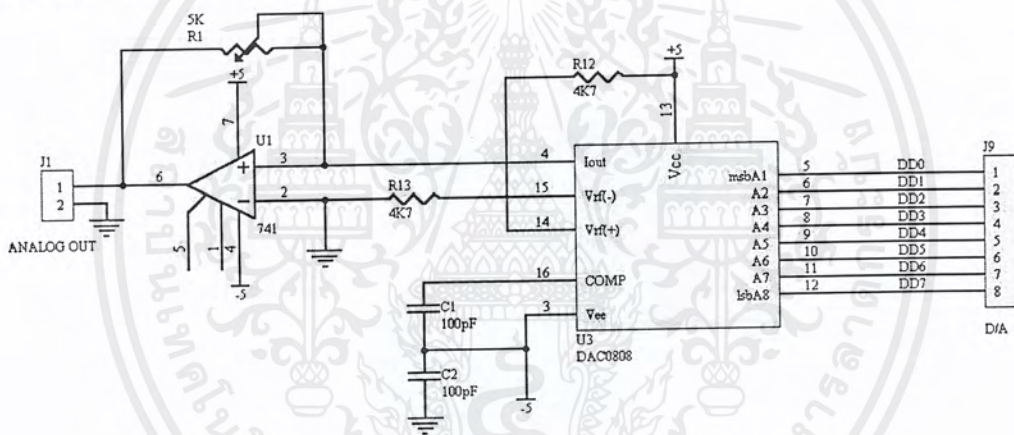
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

หลักการการทำงานของวงจร

การทำงานของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ดังแสดงในรูปที่ 3.2 ได้ใช้ไอซีเบอร์ DAC0808 เป็นตัวแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ซึ่งมีแหล่งกำเนิดกระแสอ้างอิงภายในตัวมันเอง ขา 5 ถึงขา 12 เป็นข้อมูลอินพุตแบบดิจิทัลขนาด 8 บิต จะถูกแปลงเป็นสัญญาณแอนะล็อกในรูปของกระแสไฟฟ้าออกมาทางขา 4 โดยมี R8 และ R9 ทำหน้าที่กำหนดแรงดันอ้างอิงภายในให้แก่ IC3

กระแสที่ออกมาจาก IC3 จะเข้ามายังวงจรแปลงกระแสไฟฟ้าเป็นแรงดัน โดย IC1 ซึ่งเป็นไอซีออปแอมป์เบอร์ LM741 แรงดันเอาต์พุตจะออกจากขา 6 ของ IC1 โดยมี R1 เป็นตัวควบคุมอัตราการขยาย



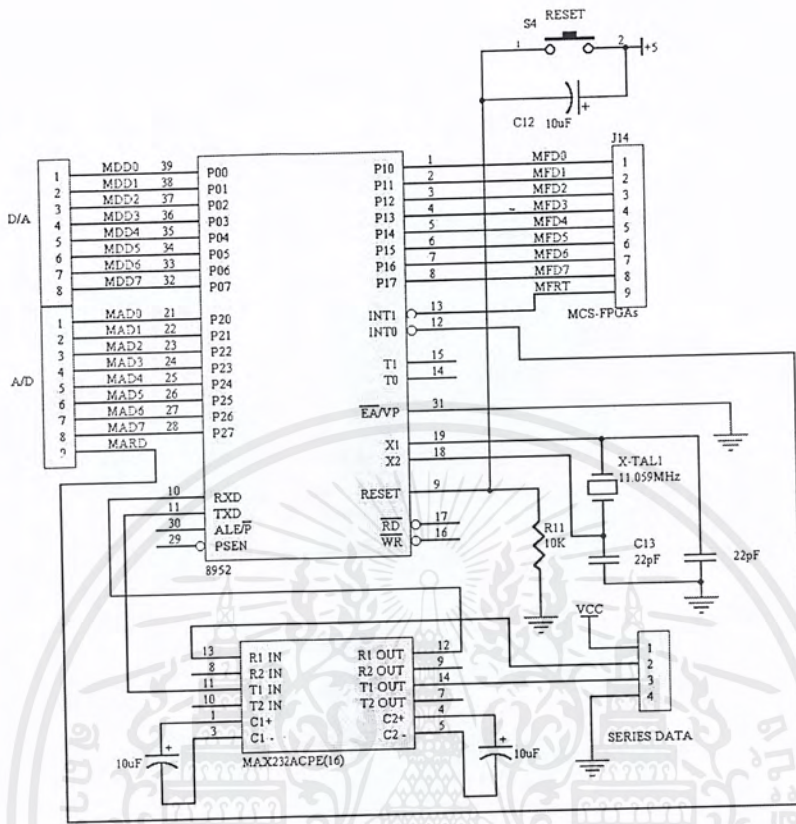
รูปที่ 3.2 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

3.3 วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51

หลักการการทำงานของวงจร

จากวงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS – 51 ดังแสดงในรูปที่ 3.3 ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 เป็นตัวประมวลผลกลาง เพราะวงจรภายในของตัวไมโครคอนโทรลเลอร์เบอร์นี้จะมีหน่วยความจำโปรแกรมที่สามารถเก็บข้อมูลโปรแกรมสำหรับไว้ใช้งานและสามารถเขียนและลบข้อมูลได้สะดวกรวดเร็วเหมาะแก่การใช้ในการทดลองเป็นอย่างยิ่ง จากรูปนอกจาก ไมโครโปรเซสเซอร์เบอร์ AT8952 แล้ว ยังประกอบด้วย คริสตอลขนาดความถี่ 11.059 MHz ต่อกับตัวเก็บประจุ ค่า 22 pF อีกสองตัว เพื่อเป็นวงจรสำหรับป้อนสัญญาณความถี่ขนาดประมาณ 12 MHz ให้กับตัวไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS - 51

ที่พอร์ต P0 ของไมโครคอนโทรลเลอร์จะต่อกับจัมป์เปอร์ที่สามารถต่อกับวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกได้ และที่พอร์ต 1 ของไมโครคอนโทรลเลอร์จะต่อกับจัมป์เปอร์ที่สามารถต่อกับอุปกรณ์โปรแกรมได้ FPGAs โดยจะใช้ขาของข้อมูลทั้งหมด 8 ขาและอีก 1 ขาจะเป็นขาแสดงสถานะว่าขณะนี้ได้ทำการส่งหรือรับข้อมูลอยู่ในขณะนั้น และที่พอร์ต 2 ของไมโครคอนโทรลเลอร์จะต่อกับจัมป์เปอร์ที่สามารถต่อกับ วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล โดยจะใช้ขาของข้อมูลทั้งหมด 8 ขาและอีก 1ขาจะเป็นขาสำหรับสั่งให้วงจรส่งข้อมูลที่ทำการแปลงเสร็จแล้วมาให้กับไมโครโพรเซสเซอร์เพื่อใช้ในการประมวลผลต่อไป

นอกจากนี้ยังมีวงจรสื่อสารข้อมูลแบบอนุกรม โดยใช้ไอซีเบอร์ MAX232 ซึ่งเป็นไอซีที่ใช้ในการแปลงระดับแรงดันของสัญญาณให้มีระดับที่เหมาะสมแก่การที่จะ ส่ง-รับ ข้อมูลจากภายนอกกับตัวไมโครคอนโทรลเลอร์

3.4 วงจรสำหรับควบคุมการทำงานของ FPGAs

หลักการการทำงานของวงจร

วงจรสำหรับควบคุมการทำงานของของ FPGAs ประกอบด้วยวงจรต่างๆ ดังแสดงในรูปแบบที่ 3.4 ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้

- 1) วงจรการโปรแกรม อุปกรณ์ FPGAs
- 2) วงจรอินพุตและเอาต์พุต
- 3) วงจรแสดงผล

การทำงานของแต่ละวงจรมีดังนี้

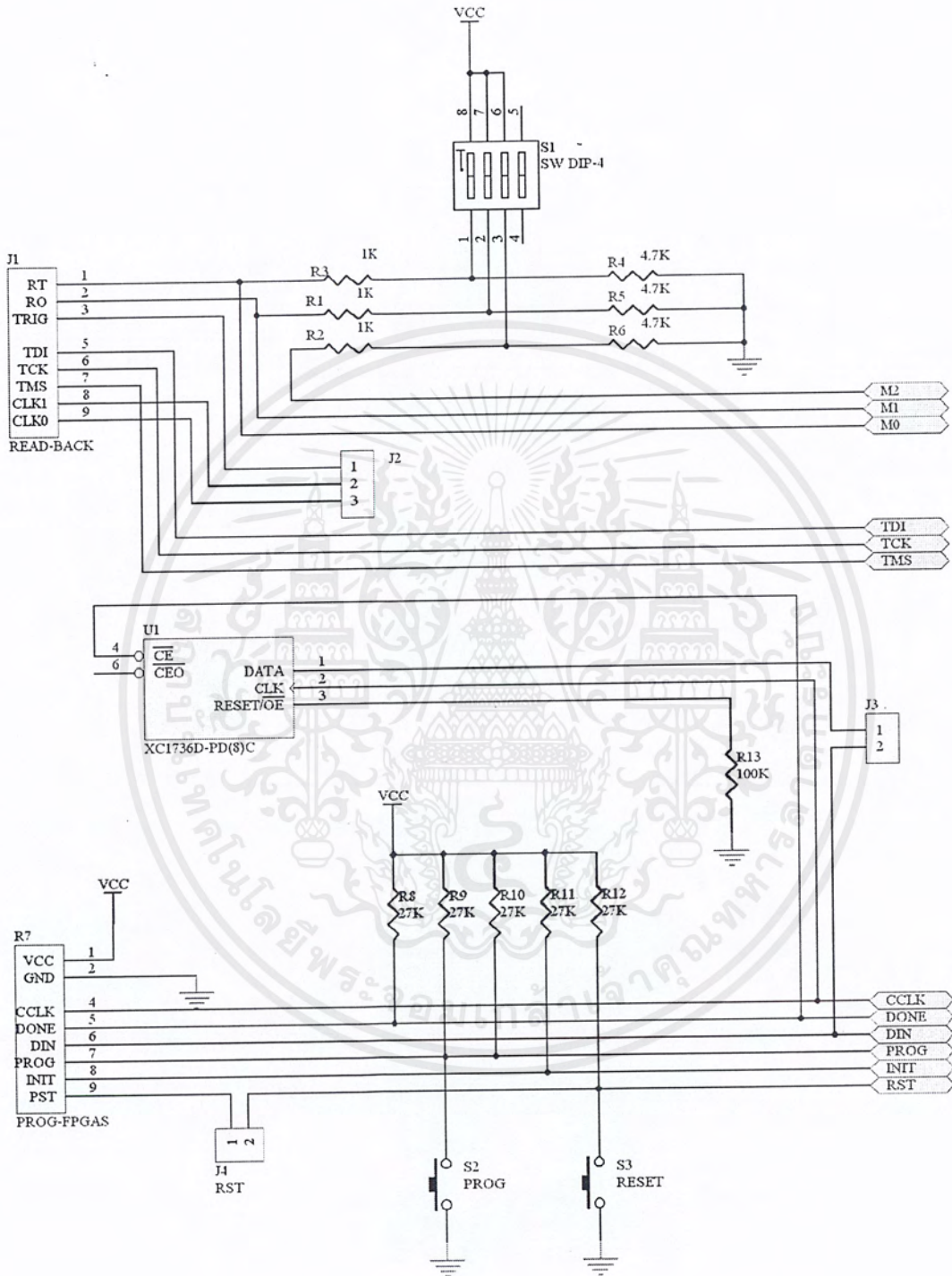
1) วงจรการโปรแกรม อุปกรณ์ FPGAs

ประกอบด้วย S1 เพื่อไว้สำหรับเลือกโหมดในการโปรแกรมและมี J3 และ J4 ใช้สำหรับต่อกับสาย Serial PROM มี IC2 ซึ่งเป็น EEPROM ไว้สำหรับเก็บโปรแกรม และมี S2, S3 ไว้สำหรับสั่งให้ FPGAs อยู่ในสถานะ PROG และ RESET

2) วงจรในส่วนอินพุตและเอาต์พุต

ประกอบด้วยอุปกรณ์ดังนี้

- 1) J19 ต่อไว้สำหรับติดต่อกับ MCS-51 โดย J9 มีทั้งหมด 9 ขาประกอบด้วย FMD0-FMD7 และ FMRT
 - 2) J8 ต่อกับวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ซึ่งมีทั้งหมด 8 ขา ดังนี้ FDD0-FDD1
 - 3) J11 ต่อกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ซึ่งมีทั้งหมด 9 ขา ดังนี้ FAD0-FAD7 และ FARD
 - 4) J7 ต่อกับ FPGAs เพื่อไว้สำหรับต่อใช้งานกับอุปกรณ์หรือวงจรอื่นๆ ได้ซึ่งมีทั้งหมด 9 ขา
 - 5) ส่วนของสวิตช์ที่ใช้ในการป้อนอินพุตลอจิก
 - 6) ส่วนของ Data Book ซึ่งเป็น EPROM เบอร์ 2764
- 3) วงจรแสดงผล
- ประกอบด้วยส่วนแสดงผลตัวเลขเจ็ดส่วน 1 หลัก 2 ตัว และแอลอีดี 8 ตัว



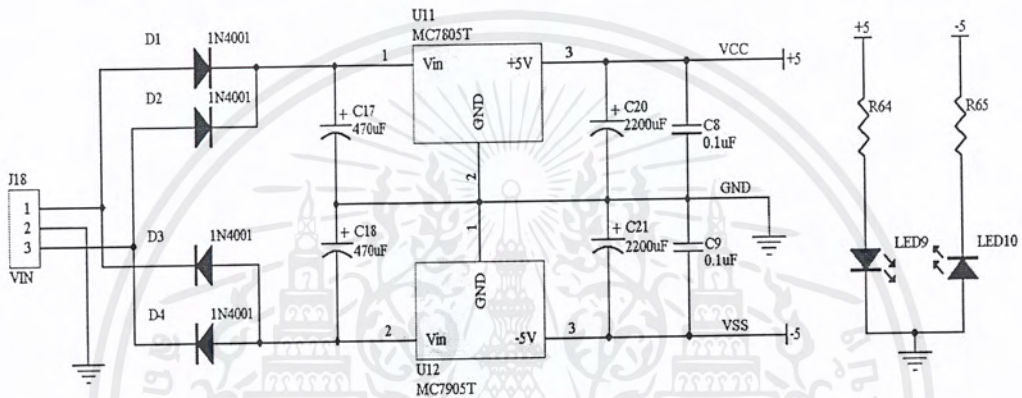
รูปที่ 3.4 วงจรสำหรับควบคุมการทำงานของ FPGAs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 วงจรจ่ายแรงดัน

หลักการการทำงานของวงจร

เนื่องจากวงจรทั้งหมดที่ออกแบบมา ต้องการใช้แรงดันเลี้ยงวงจร 5 โวลต์ ดังนั้น เราจึงต้องใช้วงจรแหล่งจ่ายไฟที่ให้แรงดันเป็น 5 โวลต์ ซึ่งวงจรแหล่งจ่ายไฟดังแสดงในรูปที่ 3.7 ทำงานโดยใช้วงจรไอซีเรกกูเลเตอร์ เบอร์ 7805 และ 7905 เป็นตัวให้แรงดันที่เอาต์พุตของวงจร



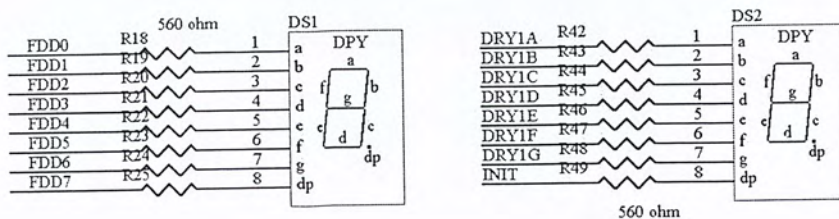
รูปที่ 3.7 วงจรจ่ายแรงดัน

3.8 วงจรภาคแสดงผล

3.8.1 วงจรภาคแสดงผลตัวเลขเจ็ดส่วน

หลักการการทำงานของวงจร

ในวงจรภาคแสดงผลตัวเลขเจ็ดส่วน ดังแสดงในรูปที่ 3.8 ได้ใช้ตัวเลขเจ็ดส่วน ชนิดแคโอดร่วม 1 หลักจำนวน 2 ตัว โดยนำขาแอนโอดของตัวเลขเจ็ดส่วน ไปต่อกับขา FPGAs ดังแสดงในตารางที่ 3.1 และมี ตัวต้านทานค่า 560 โอห์ม เป็นตัวจำกัดกระแส



รูปที่ 3.8 วงจรภาคแสดงผลตัวเลขเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ตำแหน่งการต่อขาตัวเลขเจ็ดส่วน กับขา FPGAs

Segment	DS1	DS2
a	24	37
b	25	38
c	26	39
d	27	40
e	28	44
f	29	45
g	35	46
dot	36	41

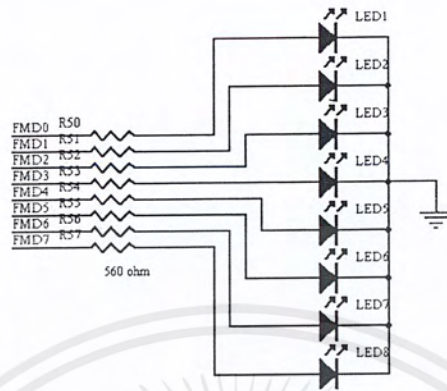
3.8.2 วงจรภาคแสดงผลแอลอีดี

หลักการทำงานของวงจร

วงจรภาคแสดงผลแอลอีดี ดังแสดงในรูปที่ 3.8 ได้ใช้แอลอีดีจำนวน 8 ตัวมาต่อแบบแคโทดร่วม แล้วนำขาแอกไปต่อกับขาของ FPGAs ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 ตำแหน่งการต่อขาแอลอีดี กับขา FPGAs

แอลอีดี	ขา FPGAs
1	81
2	82
3	83
4	84
5	3
6	4
7	5
8	6



รูปที่ 3.9 วงจรภาคแสดงผลแอลอีดี

3.9 รายละเอียดอุปกรณ์บนแผ่นวงจรพิมพ์

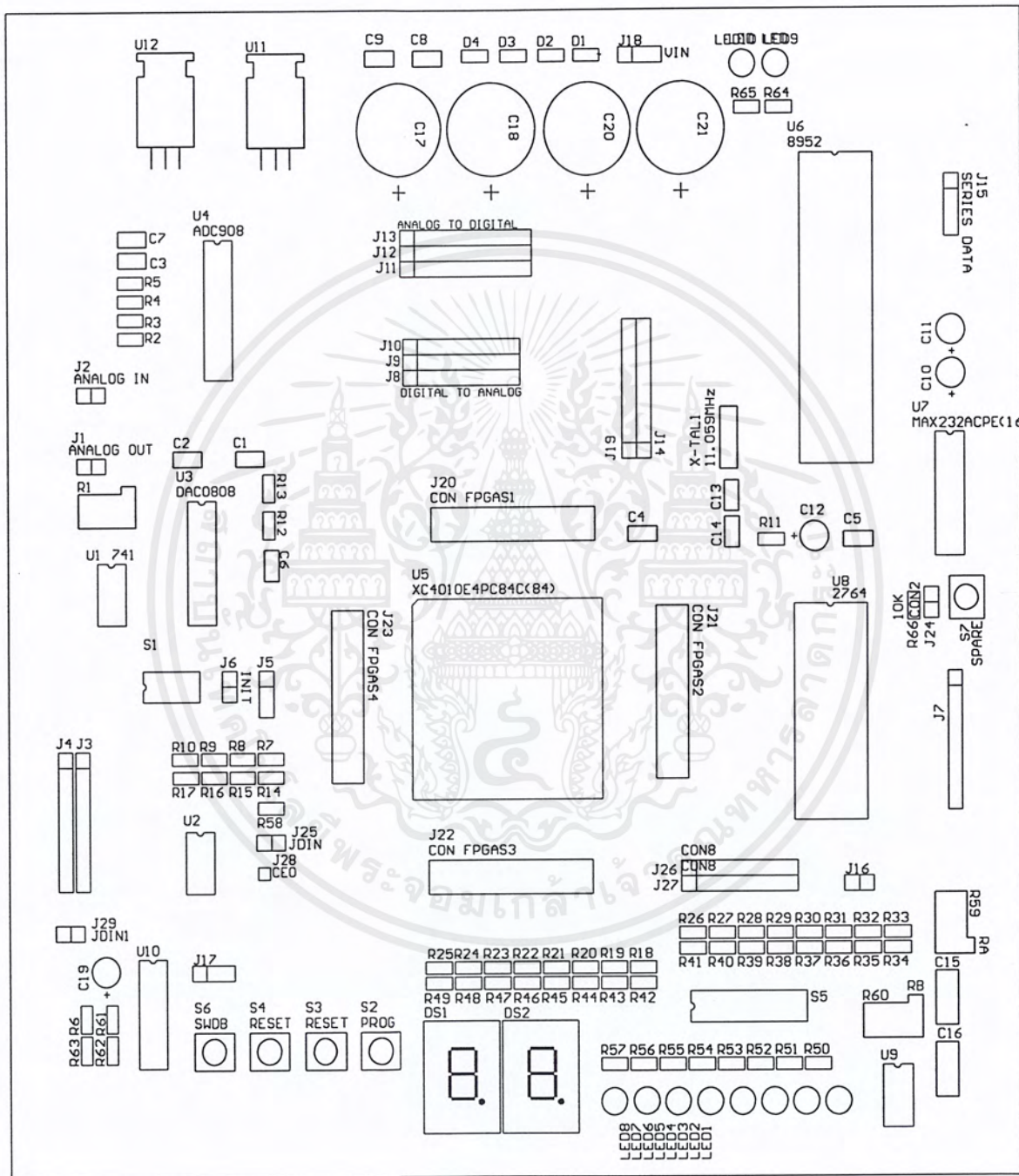
อุปกรณ์บนบอร์ด FPGAs สามารถแบ่งออกเป็นส่วนต่างๆ ดังแสดงในรูปที่ 3.10 ในส่วนของ สวิตช์ และจัมป์เปอร์ได้อธิบายหน้าที่ต่างๆ ไว้ดังนี้

ตารางที่ 3.3 หน้าที่ของจัมป์เปอร์ และสวิตช์ต่างๆ บนชุดฝึก

ตำแหน่ง	หน้าที่ของจัมป์เปอร์ และสวิตช์
J1	จัมป์เปอร์ของแอนะล็อกเอาต์พุตจากวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
J2	จัมป์เปอร์แอนะล็อกอินพุตจากวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
J3	จัมป์เปอร์สำหรับต่อกับขั้วต่อ Read Back ของสาย Xchecker
J4	จัมป์เปอร์สำหรับต่อกับขั้วต่อ Program ของสาย Xchecker
J5	จัมป์เปอร์ใช้ในการเลือกสัญญาณนาฬิกาในระบบที่ส่งเข้าควบคุมการทำงานของ Xchecker ขณะที่ทำการ Read Back
J6	จัมป์เปอร์เพื่อเลือกว่าต้องการต่อ ขา INIT ของ IC2 กับ FPGAs
J7	จัมป์เปอร์เพื่อไว้สำหรับต่อ FPGAs กับอุปกรณ์ภายนอกซึ่งมีทั้งหมด 9 ขา
J8	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกซึ่งถ้าต้องการต่อกับเพียงใช้จัมป์เปอร์ต่อ J8 กับ J9

ตำแหน่ง	หน้าที่ของจัมพ์เปอร์ และสวิทช์
J9	จัมพ์เปอร์ที่ต่อกับขาข้อมูลของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
J10	จัมพ์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกซึ่งถ้าต้องการต่อก้เพียงใช้จัมพ์เปอร์ต่อ J10 กับ J9
J11	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลซึ่งถ้าต้องการต่อก้เพียงใช้จัมพ์เปอร์ต่อ J11 กับ J12
J12	จัมพ์เปอร์ที่ต่อกับขาข้อมูลของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
J13	จัมพ์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลซึ่งถ้าต้องการต่อก้เพียงใช้จัมพ์เปอร์ต่อ J13 กับ J12
J14	จัมพ์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับ FPGAs ซึ่งถ้าต้องการต่อก้เพียงใช้จัมพ์เปอร์ต่อ J14 กับ J19
J15	คอนเน็คเตอร์ที่ใช้การติดต่อข้อมูลแบบอนุกรม มาตรฐาน RS232
J16	จัมพ์เปอร์ที่ใช้การเลือกว่าจะต่อวงจรผลิตสัญญาณรูปคลื่นสี่เหลี่ยมให้ต่อกับ FPGAs
J17	จัมพ์เปอร์ที่ใช้การเลือกว่าจะต่อวงจรดีเมาซ์สวิทช์ให้ต่อกับ FPGAs
J18	คอนเน็คเตอร์ที่ไว้ต่อกับแหล่งจ่ายแรงดันไฟ
J19	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับ MCS-51 ซึ่งถ้าต้องการต่อก้เพียงใช้จัมพ์เปอร์ต่อ J14 กับ J19
J20	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรภายนอก
J21	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรภายนอก
J22	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรภายนอก
J23	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรภายนอก
J24	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งาน Spare สวิทช์
J25	จัมพ์เปอร์เพื่อเลือกความต้องการต่อ ขา 1 (DATA) ของ IC2 กับ FPGAs
J26	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งานคิปสวิทช์ 8 หลักซึ่งใช้งานร่วมกับ J27
J27	จัมพ์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งานคิปสวิทช์ 8 หลักซึ่งใช้งานร่วมกับ J26
J28	จัมพ์เปอร์ที่ต่อกับ ขา 6 (CEO) ของ IC2
J29	จัมพ์เปอร์ที่ใช้ว่าต้องการต่อกับ ขา 6 (DIN) ของ J4 กับ FPGAs
S1	สวิทช์ที่ใช้เลือกโหมดการ โปรแกรมของ FPGAs
S2	สวิทช์ที่ใช้ทำการสั่งให้ FPGAs ทำการอยู่ในสภาวะรอรับการ โปรแกรม
S3	สวิทช์ที่ใช้ทำการสั่งให้ FPGAs ทำการรีเซต
S4	สวิทช์ที่ใช้ทำการสั่งให้ MCS-51 ทำการรีเซต
S5	สวิทช์ที่ใช้ป้อนลอจิกให้กับ FPGAs เพื่อทำการทดลอง
S6	สวิทช์ที่ใช้ป้อนสัญญาณขอบขาขึ้นหรือขอบขาลงให้กับ FPGAs
S7	สวิทช์ที่ไว้สำหรับเป็นสวิทช์ใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การวางตำแหน่งอุปกรณ์บนแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

ในบทที่ 4 นี้ จะเป็นการทดลอง และผลการทดลองของชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ในส่วนของวงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51, วงจรควบคุมการทำงานของ FPGAs, วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกวงจรดีเบอซ์สวิทช์ และวงจรผลิตสัญญาณพัลส์

4.1 การทดลอง และผลการทดลองของวงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51

4.1.1 ลำดับขั้นการทดลอง

ป้อนโปรแกรมดังแสดงในรูปที่ 4.1 ซึ่งเป็นโปรแกรมไฟวิ่งที่วิ่งจากบิตสูง(MSB)ไปหาบิตต่ำ (LSB) ให้กับไมโครคอนโทรลเลอร์ MCS-51 ด้วยเครื่องโปรแกรมไมโครคอนโทรลเลอร์ จากนั้นก็นำไมโครคอนโทรลเลอร์มาใส่ที่ชุดฝึกแล้วทำการป้อนแรงดันไฟฟ้าให้กับชุดฝึกผลการทดลองดังแสดงในตารางที่ 4.1

```
ORG 0000H
MOV A, #00000001B
START:  RR A
        MOV P0, A
        ACALL DELAY
        JMP START
DELAY:  MOV R1, #0FFH
DELAY1: MOV R2, #0FFH
        DJNZ R2, $
        DJNZ R1, DELAY1
        RET
```

รูปที่ 4.1 โปรแกรมไฟวิ่ง ที่วิ่งจาก บิตสูง (MSB) ไปหาบิตต่ำ(LSB)

4.1.2 ผลการทดลอง

ผลการทดลองดังแสดงในตารางที่ 4.1

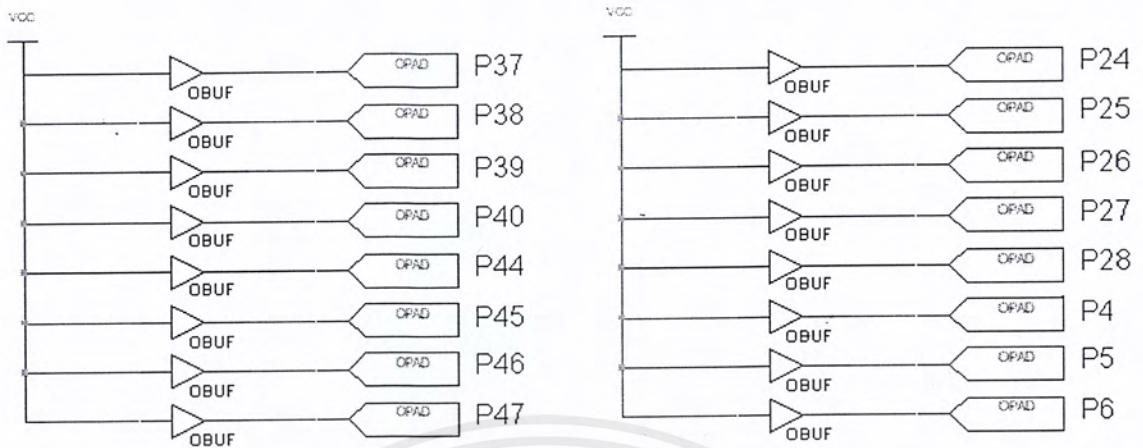
ตารางที่ 4.1 ผลการทดลองของวงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ตำแหน่งสถานะ	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1
ลำดับการติดดับ ของแอลอีดี	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ
	ดับ	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ
	ดับ	ดับ	ติด	ดับ	ดับ	ดับ	ดับ	ดับ
	ดับ	ดับ	ดับ	ติด	ดับ	ดับ	ดับ	ดับ
	ดับ	ดับ	ดับ	ดับ	ติด	ดับ	ดับ	ดับ
	ดับ	ดับ	ดับ	ดับ	ดับ	ติด	ดับ	ดับ
	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ติด	ดับ
	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ติด

4.2 การทดลองและผลการทดลองของวงจรสำหรับควบคุมการทำงานของ FPGAs

4.2.1 ลำดับขั้นการทดลอง

ในการทดลองวงจรในส่วนของ FPGAs นั้นได้ทดลองการโปรแกรมตัว FPGAs ผ่านทางสาย Xchecker โดยใช้วงจรที่เขียนด้วยโปรแกรม Foundation Version 2.1i เพื่อที่จะสั่งงานให้ FPGAs ส่งสัญญาณลอจิก “1” ออกไปที่ DS1 และ DS2 โดยวงจรที่ใช้กำหนดการทำงาน ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 วงจรสำหรับควบคุมการทำงานของ FPGAs

4.2.2 ผลการทดลอง

เมื่อป้อนวงจรผ่านทางสาย Xchecker แล้ว ที่ตัวเลขเจ็ดส่วน คือ DS1 และ DS2 ติดสว่างทุกส่วนเพราะ FPGAs ส่งสัญญาณลอจิกระดับสูงออกมาที่ขาที่กำหนดไว้จาก PAD ภายในวงจรซึ่งเกี่ยวข้องกับขาอินพุต และขาเอาต์พุตของ FPGAs

4.3 การทดลองและผลการทดลองของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

4.3.1 ลำดับขั้นการทดลอง

นำแหล่งจ่ายแรงดันไฟฟ้าเข้าที่ขา 4 ของไอซี ADC 908 ซึ่งเป็นขา แอนะล็อกอินพุต แล้วทำการปรับแรงดันไฟฟ้า เป็นค่าต่างๆ สังเกตระดับลอจิกที่ขาสัญญาณดิจิทัลเอาต์พุตและบันทึกผลการทดลอง ดังแสดงในตารางที่ 4.2

4.3.2 ผลการทดลอง

ผลการทดลองดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

ระดับแรงดันไฟฟ้า (โวลต์)	7	6	5	4	3	2	1	0
-5	0	0	0	0	0	0	0	0
-4	1	1	1	0	1	0	0	0
-3	1	1	1	1	0	1	0	0
-2	1	1	1	0	0	0	1	0
-1	0	0	0	0	0	1	1	0
0	0	0	0	1	1	1	1	0
1	1	1	1	1	1	0	0	1
2	0	0	0	1	1	1	0	1
3	0	0	0	0	1	0	1	0
4	1	0	0	1	0	1	1	1
5	1	1	1	1	1	1	1	1

4.4 การทดลอง และผลการทดลองของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

4.4.1 ลำดับขั้นการทดลอง

ทำการป้อนสัญญาณลอจิกเข้าที่ขาสัญญาณดิจิทัลอินพุต ของไอซี DAC 0808 แล้วทำการวัดแรงดันไฟฟ้าที่ขาสัญญาณแอนะล็อกเอาต์พุต เป็นค่าต่างๆ สังเกตระดับลอจิกที่ขาสัญญาณดิจิทัลเอาต์พุตและบันทึกผลการทดลอง ที่ได้ดังตารางที่ 4.3

4.4.2 ผลการทดลอง

ผลการทดลองดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

สัญญาณดิจิทัลอินพุต	ระดับแรงดันไฟฟ้า (โวลต์)
0000 0000	0
0000 0001	1
0000 0010	1.8
0000 0100	2
0000 1000	2.5
0001 0000	2.5
0010 0000	3
0100 0000	3.5
1000 0000	4
1111 1111	5

4.5 การทดลอง และผลการทดลองของวงจรดีเบาช์สวิทช์

วงจรดีเบาช์สวิทช์สามารถผลิตลอจิกได้ 2 ระดับ คือ “0” และ “1” ในเวลาเดียวกันแต่ถ้าหากมีการกดสวิทช์จะทำให้สถานะทางเอาต์พุตของวงจรดีเบาช์สวิทช์เปลี่ยนเป็นสถานะตรงกันข้าม

4.5.1 ลำดับขั้นการทดลอง

นำโวลต์มิเตอร์วัดระดับแรงดันเอาต์พุตของวงจรดีเบาช์สวิทช์ทั้งสองจุด สังเกตและบันทึกผลการทดลอง

4.5.2 ผลการทดลอง

ผลการทดลองดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 ผลการทดลองของวงจรดีเบาช์สวิทช์

ตำแหน่งเอาต์พุต	“0”	“1”
ไม่กดสวิทช์	0 โวลต์	5 โวลต์
กดสวิทช์	5 โวลต์	0 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลอง และผลการทดลองของวงจรผลิตสัญญาณพัลส์

4.6.1 ลำดับขั้นการทดลอง

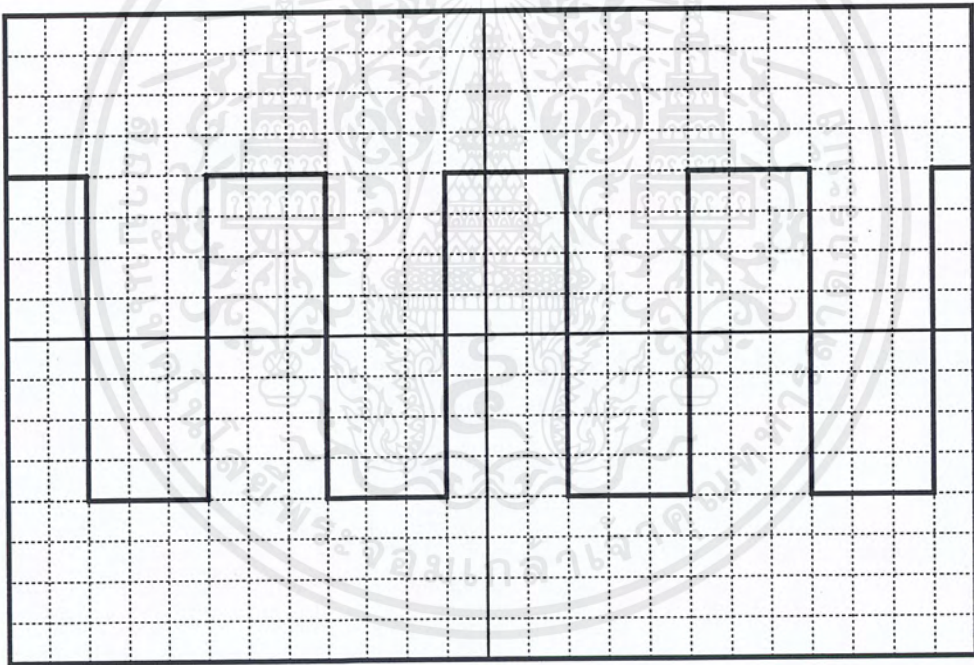
นำออสซิลโลสโคปมาวัดสัญญาณที่เอาต์พุตของวงจรผลิตสัญญาณพัลส์สังเกต และบันทึกผลการทดลอง

4.6.2 ผลการทดลอง

จากการทดลองหลังจากที่นำออสซิลโลสโคปมาวัดสัญญาณที่เอาต์พุต ของวงจรพัลส์ผลการทดลองที่ได้ ดังแสดงในรูปที่ 4.3

$$\text{TIMES/DIV} = 0.1 \text{ uSec}$$

$$\text{Volt/DIV} = 0.5 \text{ V}$$



รูปที่ 4.3 ผลการทดลองวัดรูปคลื่นจากวงจรผลิตสัญญาณพัลส์

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

ชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ซึ่งได้จัดทำขึ้นจะประกอบด้วยวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล, วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก, วงจรสำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS - 51, วงจรสำหรับควบคุมการทำงานของ FPGAs, วงจรดีไบซ์สวิตช์, วงจรกำเนิดสัญญาณพัลส์, วงจรจ่ายแรงดัน และวงจรภาคแสดงผล

5.2 ปัญหา และแนวทางแก้ไข

ในการจัดทำโครงการเกี่ยวกับชุดฝึกออกแบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ในครั้งนี้สามารถที่จะสรุปปัญหาที่เกิดขึ้น ได้ดังนี้

1. การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล จะเกิดความผิดพลาดเพิ่มขึ้นเนื่องจากกราวด์ของสัญญาณแอนะล็อกในชุดฝึกมีขนาดเล็ก
2. แนวทางแก้ไข ได้ทำการย้ายจุดต่อกราวด์ของสัญญาณแอนะล็อกไปในจุดที่มีขนาดใหญ่ขึ้น
3. การส่งข้อมูลระหว่าง FPGAs กับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลไม่สามารถทำได้โดยตรงเนื่องจากไม่มีตัวบัฟเฟอร์ระหว่าง FPGAs กับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
4. แนวทางการแก้ไข ทำได้โดยการส่งข้อมูลระหว่าง FPGAs กับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลโดยส่งข้อมูลผ่านไปยังไมโครคอนโทรลเลอร์ MCS-51 ก่อนที่จะเพื่อใช้หน่วยความจำภายในของไมโครคอนโทรลเลอร์ MCS - 51 เป็นบัฟเฟอร์ในการส่งข้อมูล
5. มีสัญญาณรบกวนที่เข้ามาทางขา Vcc ของไอซีต่างๆ ที่ใช้ในชุดฝึก
6. แนวทางการแก้ไข ได้นำตัวเก็บประจุมาต่อที่ขา Vcc ของ ไอซีทุกตัวในชุดฝึกเพื่อป้องกันสัญญาณรบกวน
7. วงจรที่ใช้ในการกำหนดโหมดในการทำงานของ FPGAs เมื่อป้อนสัญญาณแล้วเกิดการผิดพลาด ชุดฝึกไม่สามารถทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. แนวทางการแก้ไข ได้แยกวงจรที่ใช้ในการกำหนดโหมดในการทำงานของ FPGAs ออกมาเป็นแผ่นวงจรอีกแผ่นเพื่อที่จะให้สามารถใช้งานได้

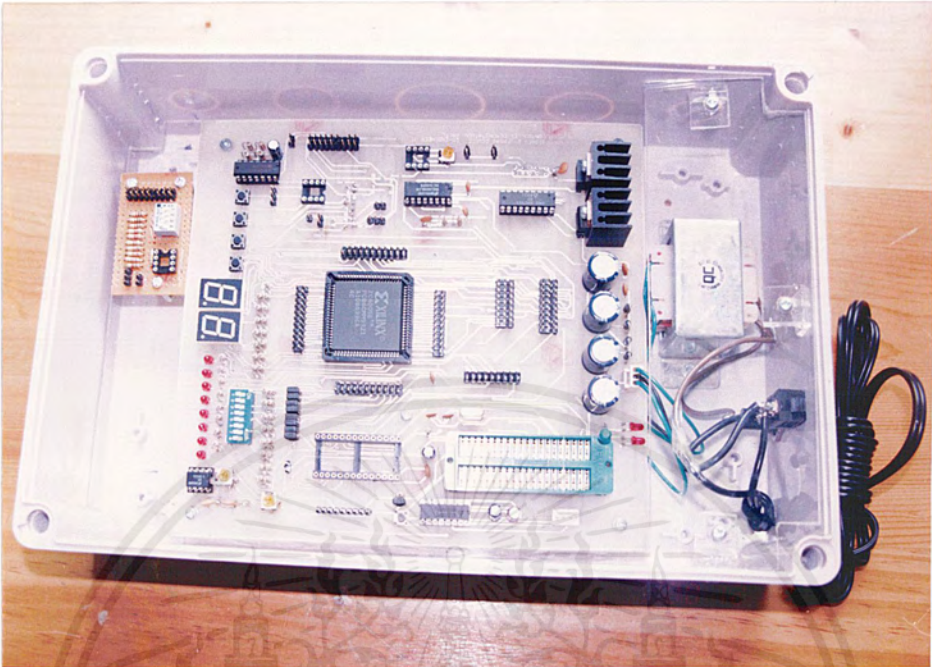
5.3 แนวทางการพัฒนา

ควรเพิ่มวงจร FIFO (First In First Out) หรือหน่วยความจำเพื่อเป็นบัฟเฟอร์ระหว่างการส่งผ่านข้อมูลระหว่าง FPGAs กับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล จะทำให้การทำงานของชุดฝึកมีประสิทธิภาพมากยิ่งขึ้น





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

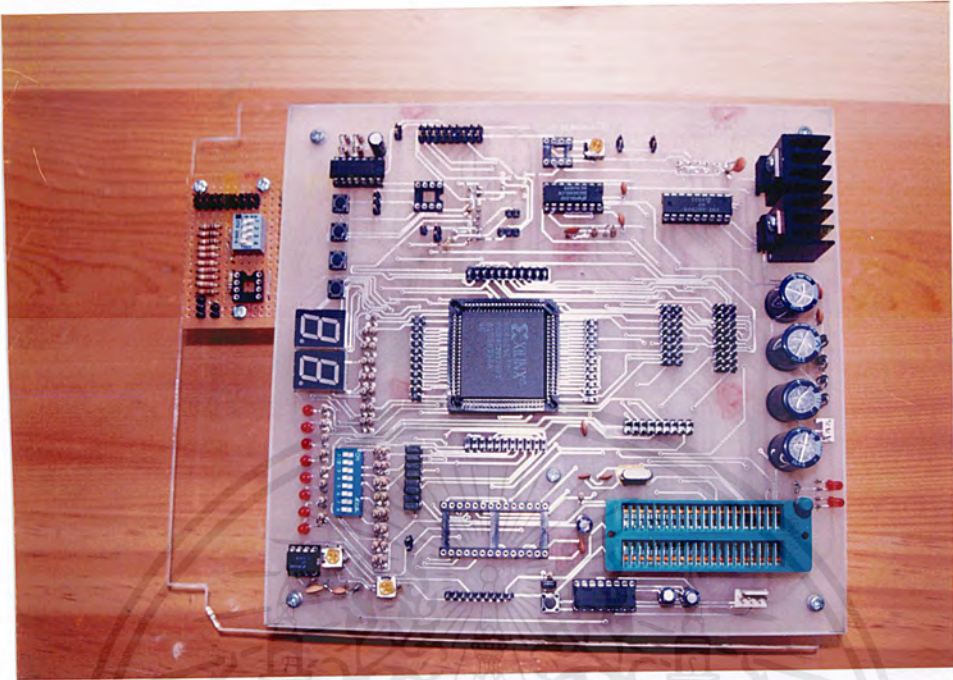


รูปที่ ก.1 เครื่องต้นแบบ

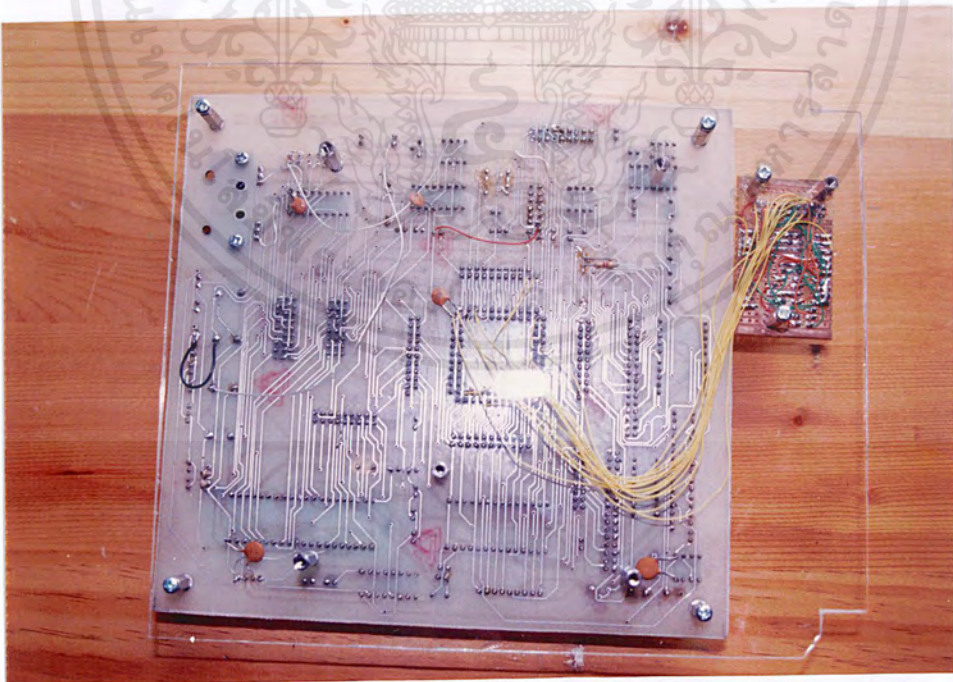
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

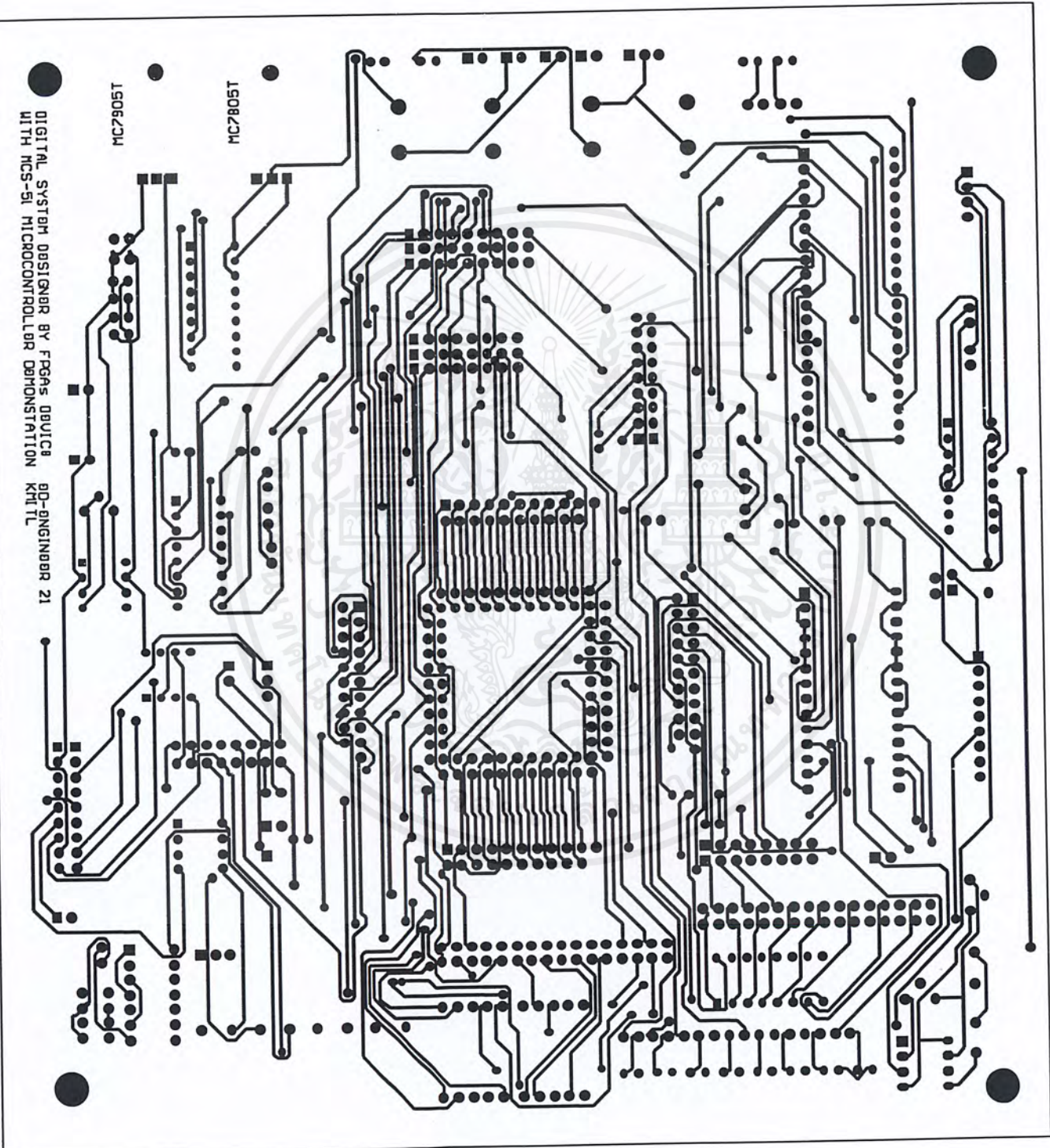


รูปที่ ข.1 แผงวงจรพิมพ์ของชุดฝึกด้านหน้า

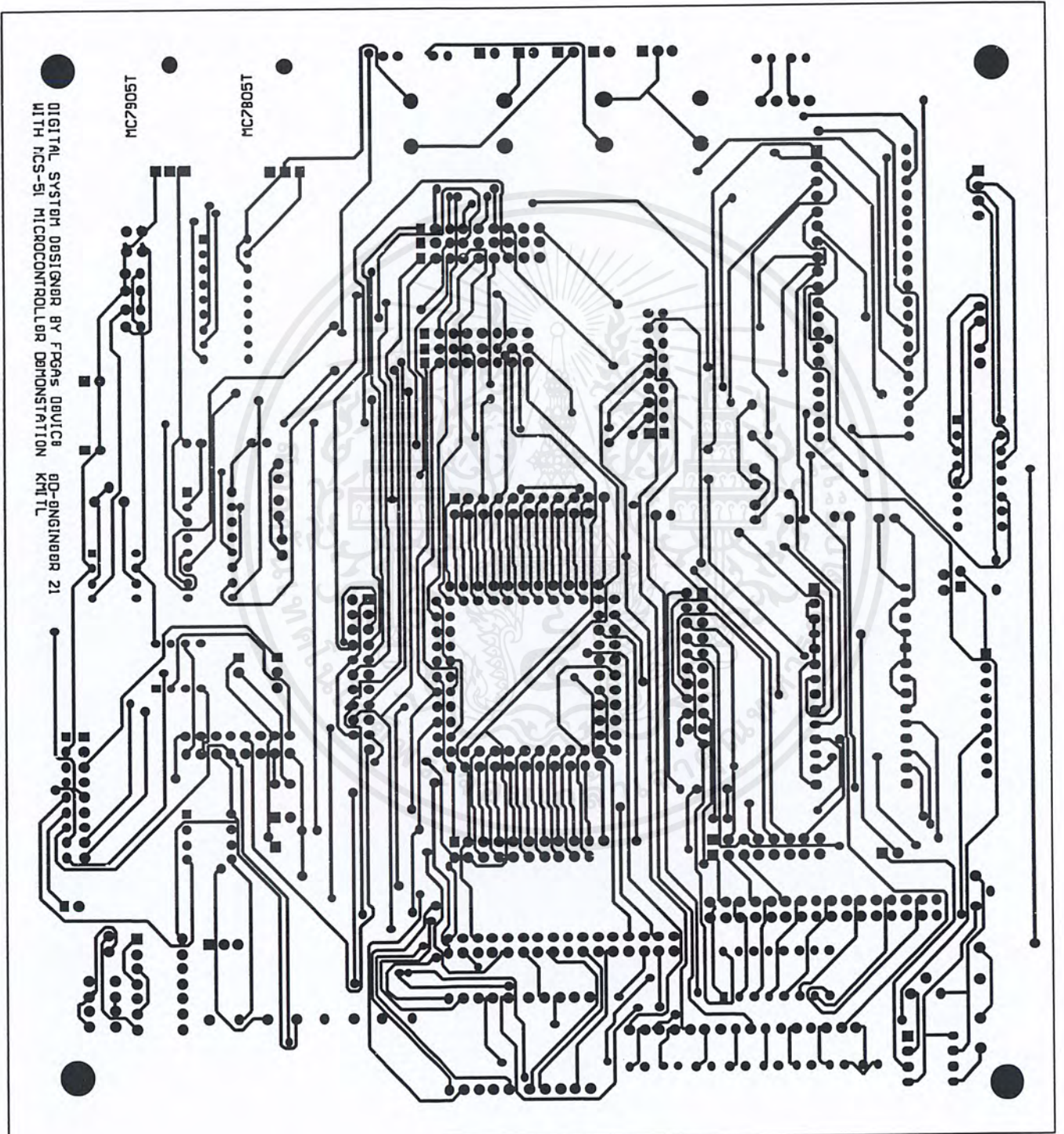


รูปที่ ข.2 แผงวงจรพิมพ์ของชุดฝึกด้านหลัง

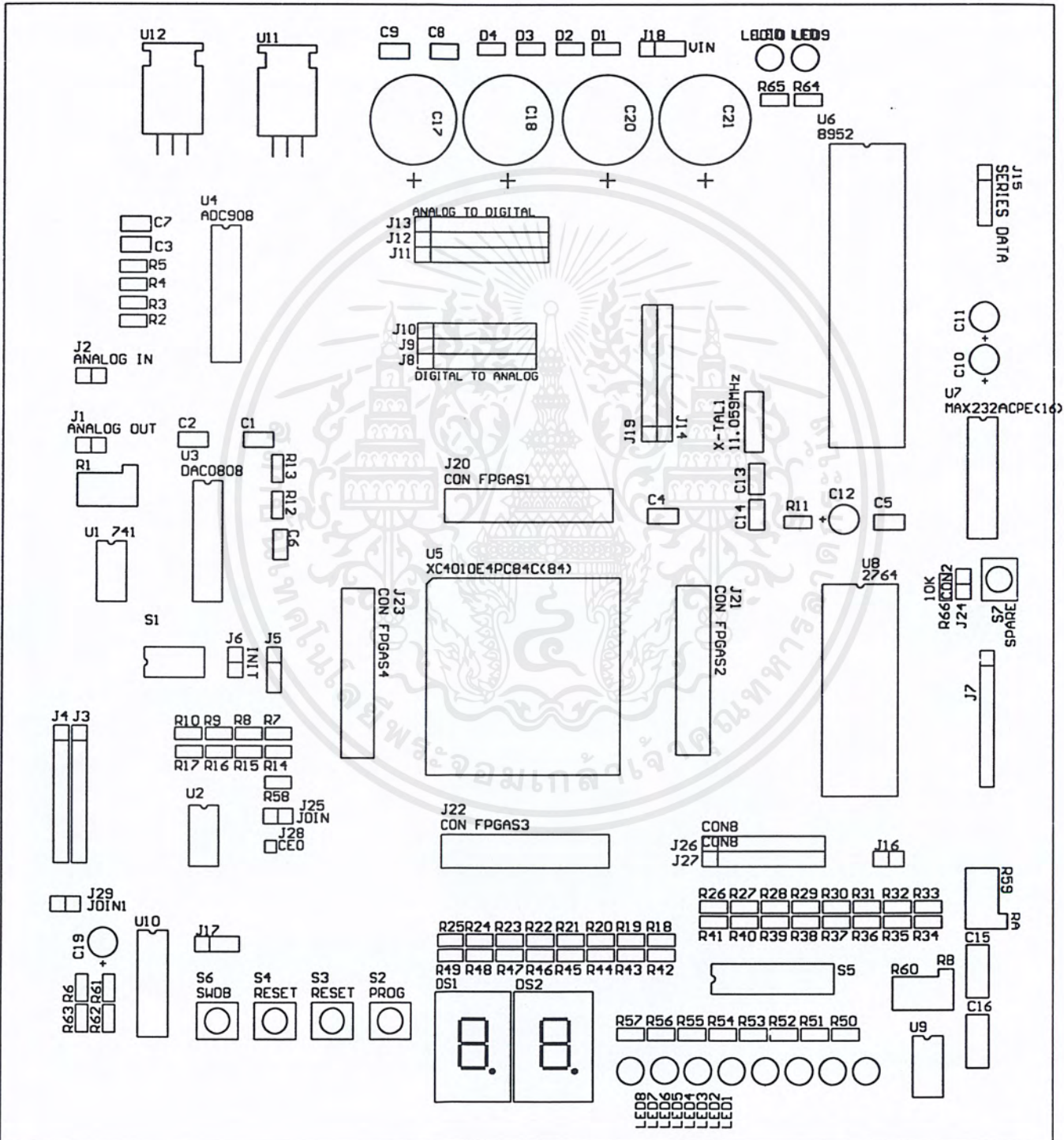
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ ข.3 ถ้ายกของแดงของแผ่นวงจรพิมพ์ด้านบนหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรในคณะวิศวกรรมศาสตร์เท่านั้นไปใช้ประโยชน์ด้านการค้า
รูปที่ ข.4 ลายทองแดงของแผ่นวงจรพิมพ์ด้านล่าง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **รูปที่ ข.5 การวางอุปกรณ์บนแผ่นวงจรมพิมพ์** ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค

รายละเอียดของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XC4000E Electrical Specifications

Definition of Terms

In the following tables, some specifications may be designated as Advance or Preliminary. These terms are defined as follows:

Advance: initial estimates based on simulation and/or extrapolation from other speed grades, devices, or device families. Values are subject to change. Use as estimates, not for production.

Preliminary: Based on preliminary characterization. Further changes are not expected.

Unmarked: Specifications not identified as either Advance or Preliminary are to be considered Final.

Except for pin-to-pin input and output parameters, the a.c. parameter delay specifications included in this document are derived from measuring internal test patterns. All specifications are representative of worst-case supply voltage and junction temperature conditions.

All specifications subject to change without notice.

XC4000E DC Characteristics

Absolute Maximum Ratings

Symbol	Description	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to +7.0	V
V_{IN}	Input voltage relative to GND (Note 1)	-0.5 to $V_{CC} + 0.5$	V
V_{TS}	Voltage applied to 3-state output (Note 1)	-0.5 to $V_{CC} + 0.5$	V
T_{STG}	Storage temperature (ambient)	-65 to +150	°C
\bar{T}_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+260	°C
T_J	Junction Temperature	Ceramic packages	+150
		Plastic packages	+125

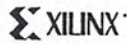
Note 1: Maximum DC excursion above V_{CC} or below Ground must be limited to either 0.5 V or 10 mA, whichever is easier to achieve. During transitions, the device pins may undershoot to -2.0 V or overshoot to $V_{CC} + 2.0$ V, provided this over or undershoot lasts less than 10 ns and with the forcing current being limited to 200 mA.

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Recommended Operating Conditions

Symbol	Description		Min	Max	Units
V_{CC}	Supply voltage relative to GND, $T_J = -0^\circ\text{C}$ to $+85^\circ\text{C}$	Commercial	4.75	5.25	V
	Supply voltage relative to GND, $T_J = -40^\circ\text{C}$ to $+100^\circ\text{C}$	Industrial	4.5	5.5	V
	Supply voltage relative to GND, $T_C = -55^\circ\text{C}$ to $+125^\circ\text{C}$	Military	4.5	5.5	V
V_{IH}	High-Level Input Voltage	TTL inputs	2.0	V_{CC}	V
		CMOS inputs	70%	100%	V_{CC}
V_{IL}	Low-Level Input Voltage	TTL inputs	0	0.8	V
		CMOS inputs	0	20%	V_{CC}
\bar{T}_{IN}	Input signal transition time			250	ns

Notes: At junction temperatures above those listed above, all delay parameters increase by 0.35% per °C. Input and output measurement thresholds for TTL are 1.5 V and for CMOS are 2.5 V.



DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V _{OH}	High-level output voltage @ $b_H = -4.0\text{mA}$, V_{CC} min	TTL outputs	2.4	V
	High-level output voltage @ $b_H = -1.0\text{mA}$, V_{CC} min	CMOS outputs	$V_{CC}-0.5$	V
V _{OL}	Low-level output voltage @ $b_L = 12.0\text{mA}$, V_{CC} min (Note 1)	TTL outputs	0.4	V
		CMOS outputs	0.4	V
I _{CCO}	Quiescent FPGA supply current (Note 2)	Commercial	3.0	mA
		Industrial	6.0	mA
		Military	6.0	mA
I _L	Input or output leakage current	-10	+10	μA
C _{IN}	Input capacitance (sample tested)	PQFP and MQFP packages	10	pF
		Other packages	16	pF
I _{RIN}	Pad pull-up (when selected) @ $V_N = 0\text{V}$ (sample tested)	-0.02	-0.25	mA
I _{RLL}	Horizontal Longline pull-up (when selected) @ logic Low	0.2	2.5	mA

Notes: With 50% of the outputs simultaneously sinking 12mA, up to a maximum of 64 pins.

With no output current loads, no active input or Longline pull-up resistors, all package pins at V_{CC} or GND, and the FPGA configured with a Development system Tie option.

*Characterized Only.

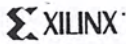
XC4000E Switching Characteristics

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values where one global clock input drives one vertical clock line in each accessible column, and where all accessible IOB and CLB flip-flops are clocked by the global clock net.

When fewer vertical clock lines are connected, the clock distribution is faster; when multiple clock lines per column are driven from the same global clock, the delay is longer. For more specific, more precise, and worst-case guaranteed data, reflecting the actual routing structure, use the values provided by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature).

Global Buffer Switching Characteristic Guidelines

Description	Symbol	Device	Speed Grade				Units
			-4	-3	-2	-1	
			Max	Max	Max	Max	
From pad through Primary buffer, to any clock K	T _{PG}	XC4003E	7.0	4.7	4.0	3.5	ns
		XC4005E	7.0	4.7	4.3	3.8	ns
		XC4006E	7.5	5.3	5.2	4.6	ns
		XC4008E	8.0	6.1	5.2	4.6	ns
		XC4010E	11.0	6.3	5.4	4.8	ns
		XC4013E	11.5	6.8	5.8	5.2	ns
		XC4020E	12.0	7.0	6.4	6.0	ns
		XC4025E	12.5	7.2	6.9	-	ns
From pad through Secondary buffer, to any clock K	T _{SG}	XC4003E	7.5	5.2	4.4	4.0	ns
		XC4005E	7.5	5.2	4.7	4.3	ns
		XC4006E	8.0	5.8	5.6	5.1	ns
		XC4008E	8.5	6.6	5.6	5.1	ns
		XC4010E	11.5	6.8	5.8	5.3	ns
		XC4013E	12.0	7.3	6.2	5.7	ns
		XC4020E	12.5	7.5	6.7	6.5	ns
		XC4025E	13.0	7.7	7.2	-	ns



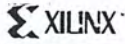
Horizontal Longline Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted. The following guidelines reflect worst-case values over the recommended operating conditions.

Description	Speed Grade		-4	-3	-2	-1	Units
	Symbol	Device	Max	Max	Max	Max	
TBUF driving a Horizontal Longline (LL):							
I going High or Low to LL going High or Low, while T is Low. Buffer is constantly active. (Note1)	T _{IO1}	XC4003E	5.0	4.2	3.4	2.9	ns
		XC4005E	5.0	5.0	4.0	3.4	ns
		XC4006E	6.0	5.9	4.7	4.0	ns
		XC4008E	7.0	6.3	5.0	4.3	ns
		XC4010E	8.0	6.4	5.1	4.4	ns
		XC4013E	9.0	7.2	5.7	4.9	ns
		XC4020E	10.0	8.2	7.3	5.6	ns
XC4025E	11.0	9.1	7.3	-	ns		
I going Low to LL going from resistive pull-up High to active Low. TBUF configured as open-drain. (Note1)	T _{IO2}	XC4003E	5.0	4.2	3.6	3.1	ns
		XC4005E	6.0	5.3	4.5	3.8	ns
		XC4006E	7.8	6.4	5.4	4.6	ns
		XC4008E	8.1	6.8	5.8	4.9	ns
		XC4010E	10.5	6.9	5.9	5.0	ns
		XC4013E	11.0	7.7	6.5	5.5	ns
		XC4020E	12.0	8.7	8.7	7.4	ns
XC4025E	12.0	9.6	9.6	-	ns		
T going Low to LL going from resistive pull-up or floating High to active Low. TBUF configured as open-drain or active buffer with I = Low. (Note1)	T _{ON}	XC4003E	5.5	4.6	3.9	3.5	ns
		XC4005E	7.0	6.0	5.7	4.7	ns
		XC4006E	7.5	6.7	5.7	4.9	ns
		XC4008E	8.0	7.1	6.0	5.2	ns
		XC4010E	8.5	7.3	6.2	5.4	ns
		XC4013E	8.7	7.5	7.0	6.2	ns
		XC4020E	11.0	8.4	7.1	6.3	ns
XC4025E	11.0	8.4	7.1	-	ns		
T going High to TBUF going inactive, not driving LL	T _{OFF}	All devices	1.8	1.5	1.3	1.1	ns
T going High to LL going from Low to High, pulled up by a single resistor. (Note 1)	T _{PUS}	XC4003E	20.0	14.0	14.0	12.0	ns
		XC4005E	23.0	16.0	16.0	14.0	ns
		XC4006E	25.0	18.0	18.0	16.0	ns
		XC4008E	27.0	20.0	20.0	16.0	ns
		XC4010E	29.0	22.0	22.0	18.0	ns
		XC4013E	32.0	26.0	26.0	21.0	ns
		XC4020E	35.0	32.5	32.5	26.0	ns
XC4025E	42.0	39.1	39.1	-	ns		
T going High to LL going from Low to High, pulled up by two resistors. (Note1)	T _{PUF}	XC4003E	9.0	7.0	6.0	5.4	ns
		XC4005E	10.0	8.0	6.8	5.8	ns
		XC4006E	11.5	9.0	7.7	6.5	ns
		XC4008E	12.5	10.0	8.5	7.5	ns
		XC4010E	13.5	11.0	9.4	8.0	ns
		XC4013E	15.0	13.0	11.7	9.4	ns
		XC4020E	16.0	14.8	14.8	10.5	ns
XC4025E	18.0	16.5	16.5	-	ns		

Note 1: These values include a minimum load. Use the static timing analyzer to determine the delay for each destination.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



XC4000E CLB Characteristics Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-39510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted

CLB Switching Characteristics Guidelines

Speed Grade		-4		-3		-2		-1		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Combinatorial Delays										
F/G inputs to X/Y outputs	T _{ILO}		2.7		2.0		1.6		1.3	ns
F/G inputs via H to X/Y outputs	T _{IHO}		4.7		4.3		2.7		2.2	ns
C inputs via SR through H to X/Y outputs	T _{HH0O}		4.1		3.3		2.4		1.9	ns
C inputs via H to X/Y outputs	T _{HH1O}		3.7		3.6		2.2		1.6	ns
C inputs via DIN through H to X/Y outputs	T _{HH2O}		4.5		3.6		2.6		1.9	ns
CLB Fast Carry Logic										
Operand inputs (F1, F2, G1, G4) to COUT	T _{OPCY}		3.2		2.6		2.1		1.7	ns
Add/Subtract input (F3) to COUT	T _{ASCY}		5.5		4.4		3.7		2.5	ns
Initialization inputs (F1, F3) to COUT	T _{INCY}		1.7		1.7		1.4		1.2	ns
CIN through function generators to X/Y outputs	T _{SUM}		3.8		3.3		2.6		1.8	ns
CIN to COUT, bypass function generators	T _{BYP}		1.0		0.7		0.6		0.5	ns
Sequential Delays										
Clock K to outputs Q	T _{CKO}		3.7		2.8		2.8		1.9	ns
Setup Time before Clock K										
F/G inputs	T _{ICK}	4.0		3.0		2.4		1.8		ns
F/G inputs via H	T _{IHCK}	6.1		4.6		3.9		2.8		ns
C inputs via H0 through H	T _{HH0CK}	4.5		3.6		3.5		2.4		ns
C inputs via H1 through H	T _{HH1CK}	5.0		4.1		3.3		2.1		ns
C inputs via H2 through H	T _{HH2CK}	4.8		3.8		3.7		2.5		ns
C inputs via DIN	T _{DICK}	3.0		2.4		2.0		1.0		ns
C inputs via EC	T _{EICK}	4.0		3.0		2.6		2.0		ns
C inputs via S/R, going Low (inactive)	T _{RCK}	4.2		4.0		4.0		1.5		ns
C _{IN} input via F/G	T _{CCK}	2.5		2.1						ns
C _{IN} input via F/G and H	T _{CHCK}	4.2		3.5						ns

XC4000E CLB Characteristics Guidelines (Continued)

Speed Grade		-4		-3		-2		-1		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Hold Time after Clock K										
F/G inputs	T _{CKH}	0		0		0		0		ns
F/G inputs via H	T _{CKIH}	0		0		0		0		ns
C inputs via H0 through H	T _{CKHH0}	0		0		0		0		ns
C inputs via H1 through H	T _{CKHH1}	0		0		0		0		ns
C inputs via H2 through H	T _{CKHH2}	0		0		0		0		ns
C inputs via DIN	T _{CKDI}	0		0		0		0		ns
C inputs via EC	T _{CKEC}	0		0		0		0		ns
C inputs via SR, going Low (inactive)	T _{CKR}	0		0		0		0		ns
Clock										
Clock High time	T _{CH}	4.5		4.0		4.0		3.0		ns
Clock Low time	T _{CL}	4.5		4.0		4.0		3.0		ns
Set/Reset Direct										
Width (High)	T _{RPW}	5.5		4.0		4.0		3.0		ns
Delay from C inputs via S/R, going High to Q	T _{RIO}		6.5		4.0		4.0		3.0	ns
Master Set/Reset (Note 1)										
Width (High or Low)	T _{MRW}	13.0		11.5		11.5		10.0		ns
Delay from Global Set/Reset net to Q	T _{MRQ}		23.0		18.7		17.4		15.0	ns
Global Set/Reset inactive to first active clock K edge	T _{MRK}									ns
Toggle Frequency (Note 2)	f _{TOG}		111		125		125		166	MHz

Note 1: Timing is based on the XC4005E. For other devices see the static timing analyzer.

Note 2: Export Control Max. flip-flop toggle rate.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wide Decoder Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted. The following guidelines reflect worst-case values over the recommended operating conditions.

Description	Symbol	Device	Speed Grade				Units
			-4	-3	-2	-1	
			Max	Max	Max	Max	
Full length, both pull-ups, inputs from IOB I-pins	T _{WAF}	XC4003E	9.2	5.0	5.0	4.3	ns
		XC4005E	9.5	6.0	6.0	5.1	ns
		XC4006E	12.0	7.0	7.0	6.0	ns
		XC4008E	12.5	8.0	8.0	6.5	ns
		XC4010E	15.0	9.0	9.0	7.5	ns
		XC4013E	16.0	11.0	11.0	8.6	ns
		XC4020E	17.0	13.9	13.9	10.1	ns
		XC4025E	18.0	16.9	16.9	–	ns
Full length, both pull-ups, inputs from internal logic	T _{WAFI}	XC4003E	12.0	7.0	7.0	5.5	ns
		XC4005E	12.5	8.0	8.0	6.4	ns
		XC4006E	14.0	9.0	9.0	7.0	ns
		XC4008E	16.0	10.0	10.0	7.5	ns
		XC4010E	18.0	11.0	11.0	8.5	ns
		XC4013E	19.0	13.0	13.0	10.0	ns
		XC4020E	20.0	15.5	15.5	11.8	ns
		XC4025E	21.0	18.9	18.9	–	ns
Half length, one pull-up, inputs from IOB I-pins	T _{WAO}	XC4003E	10.5	6.0	6.0	5.1	ns
		XC4005E	10.5	7.0	7.0	6.0	ns
		XC4006E	13.5	8.0	8.0	6.5	ns
		XC4008E	14.0	9.0	9.0	7.0	ns
		XC4010E	16.0	10.0	10.0	7.5	ns
		XC4013E	17.0	12.0	12.0	10.0	ns
		XC4020E	18.0	15.0	15.0	11.8	ns
		XC4025E	19.0	17.6	17.6	–	ns
Half length, one pull-up, inputs from internal logic	T _{WAOI}	XC4003E	12.0	8.0	8.0	6.0	ns
		XC4005E	12.5	9.0	9.0	7.0	ns
		XC4006E	14.0	10.0	10.0	7.6	ns
		XC4008E	16.0	11.0	11.0	8.4	ns
		XC4010E	18.0	12.0	12.0	9.2	ns
		XC4013E	19.0	14.0	14.0	10.8	ns
		XC4020E	20.0	16.8	16.8	12.6	ns
		XC4025E	21.0	19.6	19.6	–	ns

Note 1: These delays are specified from the decoder input to the decoder output.

Note 2: Fewer than the specified number of pullup resistors can be used, if desired. Using fewer pullups reduces power consumption but increases delays. Use the static timing analyzer to determine delays if fewer pullups are used.



XC4000E and XC4000X Series Field Programmable Gate Arrays

CLB Edge-Triggered (Synchronous) RAM Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted.

Single Port RAM	Speed Grade		-4		-3		-2		-1		Units
	Size	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation											
Address write cycle time (clock K period)	16x2	T _{WCS}	15.0		14.4		11.6		8.0		ns
	32x1	T _{WCTS}	15.0		14.4		11.6		8.0		ns
Clock K pulse width (active edge)	16x2	T _{WPS}	7.5	1 ms	7.2	1 ms	5.8	1 ms	4.0		ns
	32x1	T _{WPTS}	7.5	1 ms	7.2	1 ms	5.8	1 ms	4.0		ns
Address setup time before clock K	16x2	T _{ASS}	2.8		2.4		2.0		1.5		ns
	32x1	T _{ASTS}	2.8		2.4		2.0		1.5		ns
Address hold time after clock K	16x2	T _{AHS}	0		0		0		0		ns
	32x1	T _{AHTS}	0		0		0		0		ns
DIN setup time before clock K	16x2	T _{DSS}	3.5		3.2		2.7		1.5		ns
	32x1	T _{DSTS}	2.5		1.9		1.7		1.5		ns
DIN hold time after clock K	16x2	T _{DHS}	0		0		0		0		ns
	32x1	T _{DHTS}	0		0		0		0		ns
WE setup time before clock K	16x2	T _{WSS}	2.2		2.0		1.6		1.5		ns
	32x1	T _{WSTS}	2.2		2.0		1.6		1.5		ns
WE hold time after clock K	16x2	T _{WHS}	0		0		0		0		ns
	32x1	T _{WHTS}	0		0		0		0		ns
Data valid after clock K	16x2	T _{WOS}		10.3		8.8		7.9		6.5	ns
	32x1	T _{WOTS}		11.6		10.3		9.3		7.0	ns

Note 1: Timing for the 16x1 RAM option is identical to 16x2 RAM timing.

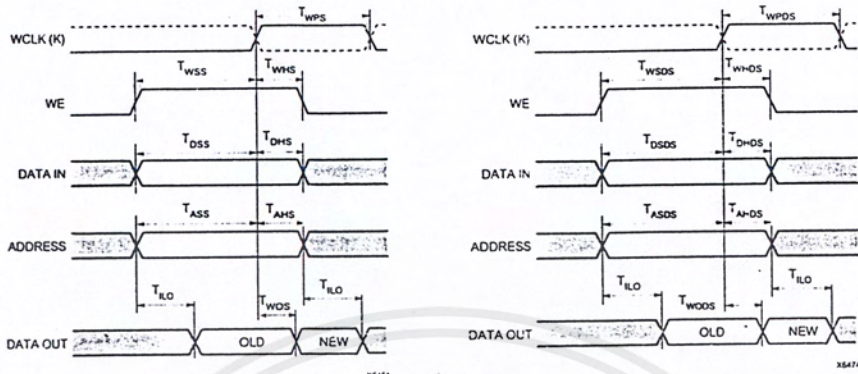
Note 2: Applicable Read timing specifications are identical to Level-Sensitive Read timing.

Dual-Port RAM	Speed Grade		-4		-3		-2		-1		Units
	Size	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation											
Address write cycle time (clock K period)	16x1	T _{WCDS}	15.0		14.4		11.6		8.0		ns
Clock K pulse width (active edge)	16x1	T _{WPDS}	7.5	1 ms	7.2	1 ms	5.8	1 ms	4.0		ns
Address setup time before clock K	16x1	T _{ASDS}	2.8		2.5		2.1		1.5		ns
Address hold time after clock K	16x1	T _{AHDS}	0		0		0		0		ns
DIN setup time before clock K	16x1	T _{DSDS}	2.2		2.5		1.6		1.5		ns
DIN hold time after clock K	16x1	T _{DHDS}	0		0		0		0		ns
WE setup time before clock K	16x1	T _{WSDS}	2.2		1.8		1.6		1.5		ns
WE hold time after clock K	16x1	T _{WSDS}	0.3		0		0		0		ns
Data valid after clock K	16x1	T _{WODS}		10.0		7.8		7.0		6.5	ns

Note: Applicable Read timing specifications are identical to Level-Sensitive Read timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLB RAM Synchronous (Edge-Triggered) Write Timing Waveforms

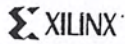


Single Port

Dual Port



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



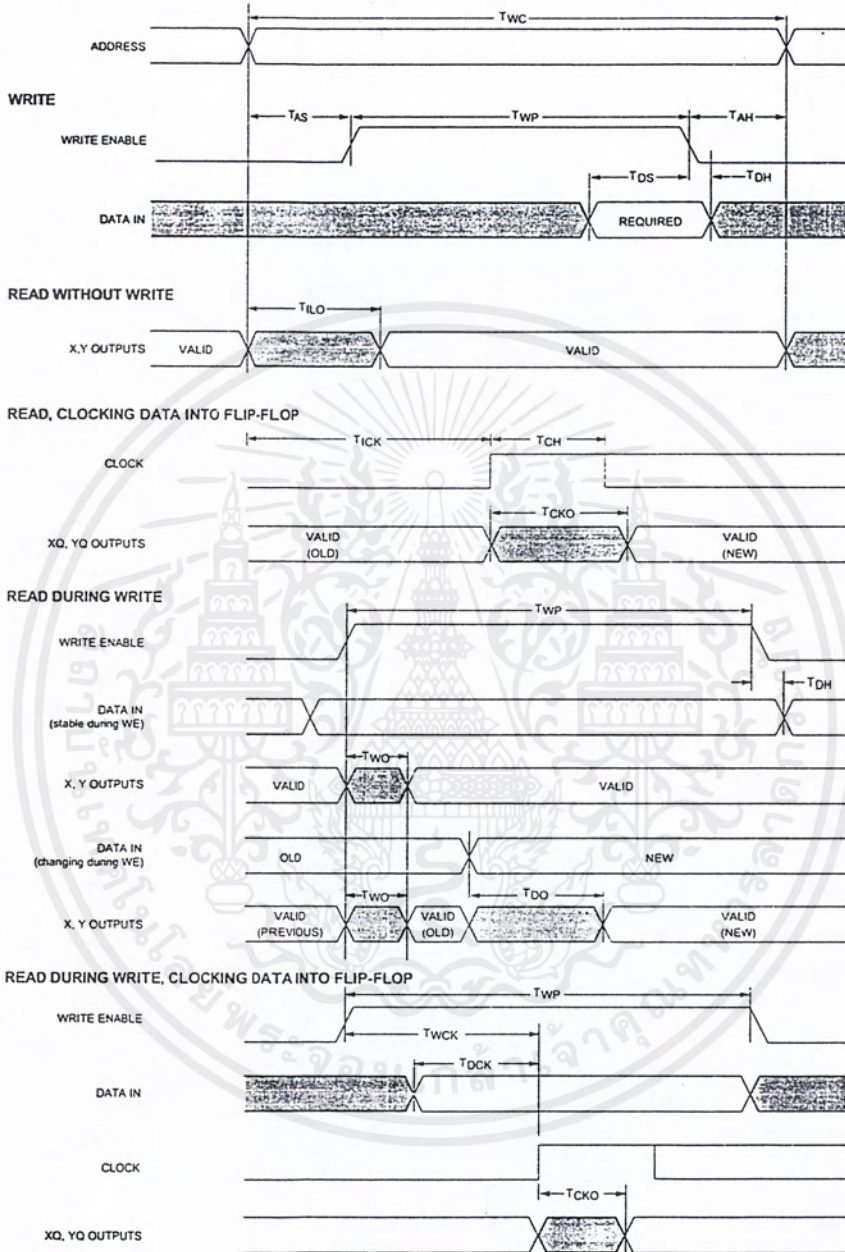
CLB Level-Sensitive RAM Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted.

Description	Speed Grade		-4		-3		-2		-1		Units
	Size	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation											
Address write cycle time	16x2	T_{WC}	8.0		8.0		8.0		8.0		ns
	32x1	T_{WCT}	8.0		8.0		8.0		8.0		ns
Write Enable pulse width (High)	16x2	T_{WP}	4.0		4.0		4.0		4.0		ns
	32x1	T_{WPT}	4.0		4.0		4.0		4.0		ns
Address setup time before WE	16x2	T_{AS}	2.0		2.0		2.0		2.0		ns
	32x1	T_{AST}	2.0		2.0		2.0		2.0		ns
Address hold time after end of WE	16x2	T_{AH}	2.5		2.0		2.0		2.0		ns
	32x1	T_{AHT}	2.0		2.0		2.0		2.0		ns
DIN setup time before end of WE	16x2	T_{DS}	4.0		2.2		0.8		0.8		ns
	32x1	T_{DST}	5.0		2.2		0.8		0.8		ns
DIN hold time after end of WE	16x2	T_{DH}	2.0		2.0		2.0		2.0		ns
	32x1	T_{DHT}	2.0		2.0		2.0		2.0		ns
Read Operation											
Address read cycle time	16x2	T_{RC}	4.5		3.1		2.6		2.6		ns
	32x1	T_{RCT}	6.5		5.5		3.8		3.8		ns
Data valid after address change (no Write Enable)	16x2	T_{ILO}		2.7		1.8		1.6		1.6	ns
	32x1	T_{IHO}		4.7		3.2		2.7		2.7	ns
Read Operation, Clocking Data into Flip-Flop											
Address setup time before clock K	16x2	T_{ICK}	4.0		3.0		2.4		2.4		ns
	32x1	T_{IHCK}	6.1		4.6		3.9		3.9		ns
Read During Write											
Data valid after WE goes active (DIN stable before WE)	16x2	T_{WO}		10.0		6.0		4.9		4.9	ns
	32x1	T_{WOT}		12.0		7.3		5.6		5.6	ns
Data valid after DIN (DIN changes during WE)	16x2	T_{DO}		9.0		6.6		5.8		5.8	ns
	32x1	T_{DOT}		11.0		7.6		6.2		6.2	ns
Read During Write, Clocking Data into Flip-Flop											
WE setup time before clock K	16x2	T_{WCK}	8.0		6.0		5.1		5.1		ns
	32x1	T_{WCKT}	9.6		6.8		5.8		5.8		ns
Data setup time before clock K	16x2	T_{DCK}	7.0		5.2		4.4		4.4		ns
	32x1	T_{DCKT}	8.0		6.2		5.3		5.3		ns

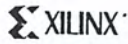
Note 1: Timing for the 16x1 RAM option is identical to 16x2 RAM timing.

CLB Level-Sensitive RAM Timing Waveforms



X2640

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



XC4000E Guaranteed Input and Output Parameters (Pin-to-Pin, TTL I/O)

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Pin-to-pin timing parameters are derived from measuring external and internal test patterns and are guaranteed over worst-case operating conditions (supply voltage and junction temperature). Listed below are representative values for typical pin locations and normal clock loading. For more specific, more precise, and worst-case guaranteed data, reflecting the actual routing structure, use the values provided by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. Values apply to all XC4000E devices unless otherwise noted.

		Speed Grade					
		-4	-3	-2	-1		
Description	Symbol	Device					Units
Global Clock to Output (fast) using OFF 	T_{ICKOF} (Max)	XC4003E	12.5	10.2	8.7	5.8	ns
		XC4005E	14.0	10.7	9.1	6.2	ns
		XC4006E	14.5	10.7	9.1	6.4	ns
		XC4008E	15.0	10.8	9.2	6.6	ns
		XC4010E	16.0	10.9	9.3	6.8	ns
		XC4013E	16.5	11.0	9.4	7.2	ns
		XC4020E	17.0	11.0	10.2	7.4	ns
		XC4025E	17.0	12.6	10.8	-	ns
Global Clock to Output (slew-limited) using OFF 	T_{ICKO} (Max)	XC4003E	16.5	14.0	11.5	7.8	ns
		XC4005E	18.0	14.7	12.0	8.2	ns
		XC4006E	18.5	14.7	12.0	8.4	ns
		XC4008E	19.0	14.8	12.1	8.6	ns
		XC4010E	20.0	14.9	12.2	8.8	ns
		XC4013E	20.5	15.0	12.8	9.2	ns
		XC4020E	21.0	15.1	12.8	9.4	ns
		XC4025E	21.0	15.3	13.0	-	ns
Input Setup Time, using IFF (no delay) 	T_{PSUF} (Min)	XC4003E	2.5	2.3	2.3	1.5	ns
		XC4005E	2.0	1.2	1.2	0.8	ns
		XC4006E	1.9	1.0	1.0	0.6	ns
		XC4008E	1.4	0.6	0.6	0.2	ns
		XC4010E	1.0	0.2	0.2	0	ns
		XC4013E	0.5	0	0	0	ns
		XC4020E	0	0	0	0	ns
		XC4025E	0	0	0	-	ns
Input Hold Time, using IFF (no delay) 	T_{PHF} (Min)	XC4003E	4.0	4.0	4.0	1.5	ns
		XC4005E	4.6	4.5	4.5	2.0	ns
		XC4006E	5.0	4.7	4.7	2.0	ns
		XC4008E	6.0	5.1	5.1	2.5	ns
		XC4010E	6.0	5.5	5.5	2.5	ns
		XC4013E	7.0	6.5	5.5	3.0	ns
		XC4020E	7.5	6.7	5.7	3.5	ns
		XC4025E	8.0	7.0	5.8	-	ns
Input Setup Time, using IFF (with delay) 	T_{PSU} (Min)	XC4003E	8.5	7.0	6.0	5.0	ns
		XC4005E	8.5	7.0	6.0	5.0	ns
		XC4006E	8.5	7.0	6.0	5.0	ns
		XC4008E	8.5	7.0	6.0	5.0	ns
		XC4010E	8.5	7.0	6.0	5.0	ns
		XC4013E	8.5	7.0	6.0	5.0	ns
		XC4020E	9.5	7.0	6.8	5.0	ns
		XC4025E	9.5	7.6	6.8	-	ns
Input Hold Time, using IFF (with delay) 	T_{PH} (Min)	XC4003E	0	0	0	0	ns
		XC4005E	0	0	0	0	ns
		XC4006E	0	0	0	0	ns
		XC4008E	0	0	0	0	ns
		XC4010E	0	0	0	0	ns
		XC4013E	0	0	0	0	ns
		XC4020E	0	0	0	0	ns
		XC4025E	0	0	0	-	ns

OFF = Output Flip-Flop, IFF = Input Flip-Flop or Latch

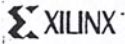
XC4000E IOB Input Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Pin-to-pin timing parameters are derived from measuring external and internal test patterns and are guaranteed over worst-case operating conditions (supply voltage and junction temperature). Listed below are representative values for typical pin locations and normal clock loading. For more specific, more precise, and worst-case guaranteed data, reflecting the actual routing structure, use the values provided by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. Values apply to all XC4000E devices unless otherwise noted.

Description	Speed Grade		-4		-3		-2		-1		Units
	Symbol	Device	Min	Max	Min	Max	Min	Max	Min	Max	
Propagation Delays (TTL Inputs)											
Pad to I1, I2	T_{PID}	All devices		3.0		2.5		2.0		1.4	ns
Pad to I1, I2 via transparent latch, no delay	T_{PLI}	All devices		4.8		3.6		3.6		2.8	ns
with delay	T_{PDLI}	XC4003E		10.4		9.3		6.9		6.4	ns
		XC4005E		10.8		9.6		7.4		6.5	ns
		XC4006E		10.8		10.2		8.1		6.9	ns
		XC4008E		10.8		10.6		8.2		7.0	ns
		XC4010E		11.0		10.8		8.3		7.3	ns
		XC4013E		11.4		11.2		9.8		8.4	ns
		XC4020E		13.8		12.4		11.5		9.0	ns
		XC4025E		13.8		13.7		12.4		—	ns
Propagation Delays (CMOS Inputs)											
Pad to I1, I2	T_{PIDC}	All devices		5.5		4.1		3.7		1.9	ns
Pad to I1, I2 via transparent latch, no delay	T_{PLIC}	All devices		8.8		6.8		6.2		3.3	ns
with delay	T_{PDLIC}	XC4003E		16.5		12.4		11.0		6.9	ns
		XC4005E		16.5		13.2		11.9		7.0	ns
		XC4006E		16.8		13.4		12.1		7.4	ns
		XC4008E		17.3		13.8		12.4		7.4	ns
		XC4010E		17.5		14.0		12.6		7.8	ns
		XC4013E		18.0		14.4		13.0		9.0	ns
		XC4020E		20.8		15.6		14.0		9.5	ns
		XC4025E		20.8		15.6		14.0		—	ns
Propagation Delays											
Clock (IK) to I1, I2 (flip-flop)	T_{IKRI}	All devices		5.6		2.8		2.6		2.7	ns
Clock (IK) to I1, I2 (latch enable, active Low)	T_{IKLI}	All devices		6.2		4.0		3.9		3.2	ns
Hold Times (Note 1)											
Pad to Clock (IK), no delay	T_{IKPI}	All devices	0	0	0	0	0	0	0	0	ns
with delay	T_{IKPID}	All devices	0	0	0	0	0	0	0	0	ns
Clock Enable (EC) to Clock (IK), no delay	T_{IKEC}	All devices	1.5	1.5	0.9	0	0	0	0	0	ns
with delay	T_{IKECD}	All devices	0	0	0	0	0	0	0	0	ns

Note 1: Input pad setup and hold times are specified with respect to the internal clock (IK). For setup and hold times with respect to the clock input pin, see the pin-to-pin parameters in the Guaranteed Input and Output Parameters table.

Note 2: Voltage levels of unused pads, bonded or unbonded, must be valid logic levels. Each can be configured with the internal pull-up (default) or pull-down resistor, or configured as a driven output, or can be driven from an external source.



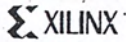
XC4000E IOB Input Switching Characteristic Guidelines (Continued)

Description	Speed Grade		-4		-3		-2		-1		Units
	Symbol	Device	Min	Max	Min	Max	Min	Max	Min	Max	
Setup Times (TTL Inputs)											
Pad to Clock (IK), no delay with delay	T_{PICK}	All devices	4.0		2.6		2.0		1.5		ns
	T_{PICKD}	XC4003E	10.9		8.2		6.0		4.8		ns
		XC4005E	10.9		8.7		6.1		5.1		ns
		XC4006E	10.9		9.2		6.2		5.8		ns
		XC4008E	11.1		9.6		6.3		5.8		ns
		XC4010E	11.3		9.8		6.4		6.0		ns
		XC4013E	11.8		10.2		7.9		7.6		ns
		XC4020E	14.0		11.4		9.4		8.2		ns
		XC4025E	14.0		11.4		10.0		-		ns
Setup Time (CMOS Inputs)											
Pad to Clock (IK), no delay with delay	T_{PICKC}	All devices	6.0		3.3		2.4		2.4		ns
	T_{PICKDC}	XC4003E	12.0		8.8		6.9		5.3		ns
		XC4005E	12.0		9.7		8.0		5.6		ns
		XC4006E	12.3		9.9		8.1		6.3		ns
		XC4008E	12.8		10.3		8.2		6.3		ns
		XC4010E	13.0		10.5		8.3		6.5		ns
		XC4013E	13.5		10.9		10.0		7.9		ns
		XC4020E	16.0		12.1		12.1		8.1		ns
		XC4025E	16.0		12.1		12.1		-		ns
(TTL or CMOS)											
Clock Enable (EC) to Clock (IK), no delay with delay	T_{EICK}	All devices	3.5		2.5		2.1		1.5		ns
	T_{EICKD}	XC4003E	10.4		8.1		4.3		4.3		ns
		XC4005E	10.4		8.5		5.6		5.0		ns
		XC4006E	10.4		9.1		6.7		6.0		ns
		XC4008E	10.4		9.5		6.9		6.0		ns
		XC4010E	10.7		9.7		7.1		6.5		ns
		XC4013E	11.1		10.1		9.0		8.0		ns
		XC4020E	14.0		11.3		10.6		9.0		ns
		XC4025E	14.0		11.3		11.0		-		ns
Global Set/Reset (Note 3)											
Delay from GSR net through Q to I1, I2	T_{RRI}			12.0		7.8		6.8		6.8	ns
	T_{MRW}		13.0		11.5		11.5		10.0		ns
GSR width											
GSR inactive to first active Clock (IK) edge	T_{MRI}										ns

Note 1: Input pad setup and hold times are specified with respect to the internal clock (IK). For setup and hold times with respect to the clock input pin, see the pin-to-pin parameters in the Guaranteed Input and Output Parameters table.

Note 2: Voltage levels of unused pads, bonded or unbonded, must be valid logic levels. Each can be configured with the internal pull-up (default) or pull-down resistor, or configured as a driven output, or can be driven from an external source.

Note 3: Timing is based on the XC4005E. For other devices see the XACT timing calculator.



IOB Output Switching Characteristics Guidelines (Continued)

Speed Grade		-4		-3		-2		-1		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Setup and Hold										
Output (O) to clock (OK) setup time	T_{OOK}	5.0		4.6		3.8		2.3		ns
Output (O) to clock (OK) hold time	T_{OKO}	0		0		0		0		ns
Clock Enable (EC) to clock (OK) setup	T_{ECOK}	4.8		3.5		2.7		2.0		ns
Clock Enable (EC) to clock (OK) hold	T_{OKEC}	1.2		1.2		0.5		0		ns
Clock										
Clock High	T_{CH}	4.5		4.0		4.0			3.0	ns
Clock Low	T_{CL}	4.5		4.0		4.0			3.0	ns
Global Set/Reset (Note 3)										
Delay from GSR net to Pad	T_{RPO}		15.0		11.8		8.7		7.0	ns
GSR width	T_{MRW}	13.0		11.5		11.5				ns
GSR inactive to first active clock (OK) edge	T_{MRO}									ns

Note 1: Output timing is measured at pin threshold, with 50pF external capacitive loads (incl. test fixture). Slew-rate limited output rise/fall times are approximately two times longer than fast output rise/fall times. For the effect of capacitive loads on ground bounce, see the "Additional XC4000 Data" section of the Programmable Logic Data Book.

Note 2: Voltage levels of unused pads, bonded or unbonded, must be valid logic levels. Each can be configured with the internal pull-up (default) or pull-down resistor, or configured as a driven output, or can be driven from an external source.

Note 3: Timing is based on the XC4005E. For other devices see the XACT timing calculator.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XC4000E IOB Output Switching Characteristic Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. These path delays, provided as a guideline, have been extracted from the static timing analyzer report. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all XC4000E devices unless otherwise noted.

Speed Grade		-4		-3		-2		-1		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Propagation Delays (TTL Output Levels)										
Clock (OK) to Pad, fast	T_{OKPOF}		7.5		6.5		4.5		3.0	ns
	slew-rate limited	T_{OKPOS}	11.5		9.5		7.0		5.0	ns
Output (O) to Pad, fast	T_{OPF}		8.0		5.5		4.8		3.2	ns
	slew-rate limited	T_{OPS}	12.0		8.5		7.3		5.2	ns
3-state to Pad hi-Z (slew-rate independent)	T_{TSHZ}		5.0		4.2		3.8		3.0	ns
3-state to Pad active and valid, fast	T_{TSONF}		9.7		8.1		7.3		6.8	ns
	slew-rate limited	T_{TSONS}	13.7		11.1		9.8		8.8	ns
Propagation Delays (CMOS Output Levels)										
Clock (OK) to Pad, fast	T_{OKPOFC}		9.5		7.3		7.0		4.0	ns
	slew-rate limited	T_{OKPOSC}	13.5		11.6		10.4		7.0	ns
Output (O) to Pad, fast	T_{OPFC}		10.0		9.7		8.7		4.0	ns
	slew-rate limited	T_{OPSC}	14.0		13.4		12.1		6.0	ns
3-state to Pad hi-Z (slew-rate independent)	T_{TSHZC}		5.2		4.3		3.9		3.9	ns
3-state to Pad active and valid, fast	T_{TSONFC}		9.1		7.6		6.8		6.8	ns
	slew-rate limited	T_{TSONSC}	13.1		11.4		10.2		8.8	ns

Note 1: Output timing is measured at pin threshold, with 50pF external capacitive loads (incl. test fixture). Slew-rate limited output rise/fall times are approximately two times longer than fast output rise/fall times. For the effect of capacitive loads on ground bounce, see the "Additional XC4000 Data" section of the Programmable Logic Data Book.

Note 2: Voltage levels of unused pads, bonded or unbonded, must be valid logic levels. Each can be configured with the internal pull-up (default) or pull-down resistor, or configured as a driven output, or can be driven from an external source.

XC4000E Boundary Scan (JTAG) Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are not measured directly. They are derived from benchmark timing patterns that are taken at device introduction, prior to any process improvements. For more detailed, more precise, and more up-to-date information, use the values provided by the XACT timing calculator and used in the simulator. These values can be printed in tabular format by running LCA2XNF -S.

The following guidelines reflect worst-case values over the recommended operating conditions. They are expressed in units of nanoseconds and apply to all XC4000E devices unless otherwise noted.

Speed Grade		-4		-3		-2		-1		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Setup Times										
Input (TDI) to clock (TCK)	T_{TDITCK}	30.0		30.0		30.0		20.0		ns
Input (TMS) to clock (TCK)	T_{TMSTCK}	15.0		15.0		15.0		10.0		ns
Hold Times										
Input (TDI) to clock (TCK)	T_{TCKTDI}	0		0		0		0		ns
Input (TMS) to clock (TCK)	T_{TCKTMS}	0		0		0		0		ns
Propagation Delay										
Clock (TCK) to Pad (TDO)	T_{TCKPO}		30.0		30.0		30.0		20.0	ns
Clock										
Clock (TCK) High	T_{TCKH}	5.0		5.0		5.0		4.0		ns
Clock (TCK) Low	T_{TCKL}	5.0		5.0		5.0		4.0		ns
Frequency	F_{MAX}		15.0		15.0		15.0		25.0	MHz

Note 1: Input setup and hold times and clock-to-pad times are specified with respect to external signal pins.

Note 2: Output timing is measured at pin threshold, with 50pF external capacitive loads (incl. test fixture). Slew-rate limited output rise/fall times are approximately two times longer than fast output rise/fall times. For the effect of capacitive loads on ground bounce, see the "Additional XC4000 Data" section of the Programmable Logic Data Book.

Note 3: Voltage levels of unused pads, bonded or unbonded, must be valid logic levels. Each can be configured with the internal pull-up (default) or pull-down resistor, or configured as a driven output, or can be driven from an external source.

Revision Control

Version	Nature of Changes
3/30/98 (1.5)	As submitted for the 1999 data book
1/29/99 (1.5)	Updated Switching Characteristics Tables
5/14/99 (1.6)	Replaced Electrical Specification and pinout pages for E, EX, and XL families with separate updates and added URL link on placeholder page for electrical specifications/pinouts for WebLINX users
8/27/99 (1.7)	Included missing IOB Propagation Delay page (6-113)
2/11/00 (1.8)	Altered IOB heads (Acrobat PDF file problem), corrected Dual-port Write Mins for -4 speed grade.



ADC-908

CMOS MICROPROCESSOR-COMPATIBLE
FAST 8-BIT A/D CONVERTER

Precision Monolithics Inc.

FEATURES

- 8-Bit Resolution and Accuracy
- No Missing Codes over Full Temperature Range
- 6µs Conversion Time
- Flexible µP Interface
- 2.5mA Maximum Standby Current
- Replaces AD7574 with Improved Speed
- Available in Die Form

ORDERING INFORMATION †

PACKAGE: 18-PIN DIP AND SO				
INL (LSB)	DNL (LSB)	MILITARY* TEMPERATURE -55°C TO +125°C	EXTENDED INDUSTRIAL TEMPERATURE -40°C TO +85°C	COMMERCIAL TEMPERATURE 0°C TO +70°C
±1/2	±3/4	ADC908AX	ADC908EX	ADC908GP
±3/4	±7/8	ADC909BX	ADC908FX	—
±3/4	±7/8	—	ADC908FP	—
±3/4	±7/8	—	ADC908FS	—

* For devices processed in total compliance to MIL-STD-883, add /883 after part number. Consult factory for 883 data sheet.

† Burn-in is available on commercial and industrial temperature range parts in CerDIP, plastic DIP, and TO-can packages. For ordering information, see PMI's Data Book, Section 2.

GENERAL DESCRIPTION

The ADC-908 is a monolithic CMOS successive-approximation analog-to-digital converter. When used with a 1.35MHz clock, a conversion time of 6µs is achieved, with full accuracy over the operating temperature range.

The ADC-908 outputs use 3-state logic, allowing direct connection to the data bus or system input port. Active-LOW chip select (CS) and read/write (RD) inputs are used to control all

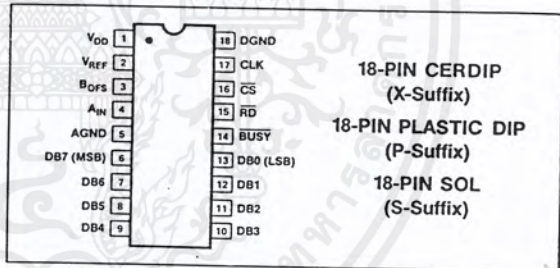
operations. This input structure permits the ADC-908 to be used as a memory-mapped input device. Depending on the control timing waveforms, the ADC-908 is interfaced like static RAM, ROM, or slow memory.

The low power consumption of the ADC-908 is derived from a single +5V supply. A negative reference voltage must also be supplied. Optimum accuracy is achieved when the reference is at -10.00V with a low output resistance. For a low-cost precision -10V/-10.24V reference, ask your PMI sales representative about the REF-08.

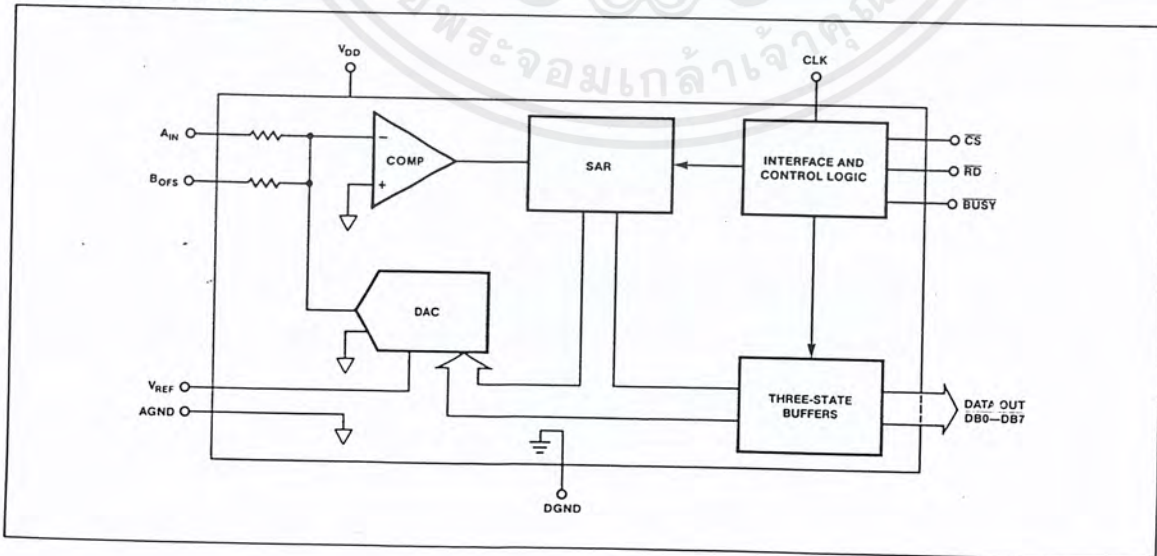
With its on-board comparator, interface logic, optional internal clock, and +5V operation, the ADC-908 is the ideal low-cost solution for microprocessor-based 8-bit A/D systems.

PMI's ADC-908 is pin-and-function compatible with the PM-7574, but offers faster conversion time and faster microprocessor bus interface timing. Conversion time has been reduced by 60% and most key timing specifications, including data access time, START command propagation delay (tWBPD), and reset time, have been improved.

PIN CONNECTIONS



FUNCTIONAL DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS ($T_A = +25^\circ\text{C}$, unless otherwise noted)

V_{DD} to AGND	0V, +7.0V
V_{DD} to DGND	0V, +7.0V
AGND to DGND	-0.3V, V_{DD}
CS, RD to DGND	-0.3V, $V_{DD} + -0.3V$
DB ₀ -DB ₇ to DGND	-0.3V, V_{DD}
CLK, <u>BUSY</u> to DGND	-0.3, V_{DD}
B _{OFS} , A _{IN}	$\pm 20V$
V_{REF}	0V, -20V

Operating Temperature Range

ADC-908AX, BX	-55°C to +125°C
ADC-908EX, FX, FP, FS	-40°C to +85°C
ADC-908GP	0°C to +70°C

Storage Temperature -65°C to +150°C
 Lead Temperature (Soldering, 10 sec) +300°C

PACKAGE TYPE	θ_{JA} (Note 2)	θ_{JC}	UNITS
18-Pin Hermetic DIP (X)	79	11	°C/W
18-Pin Plastic DIP (P)	70	30	°C/W
18-Pin SOL (S)	88	25	°C/W

NOTES:

- Digital pins are zener protected. However, proper ESD handling precautions are recommended.
- θ_{JA} is specified for worst case mounting conditions, i.e., θ_{JA} is specified for device in socket for TO, CerDIP, P-DIP, and LCC packages; θ_{JA} is specified for device soldered to printed circuit board for SO and PLCC packages.

ELECTRICAL CHARACTERISTICS at $V_{DD} = +5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ for ADC-908E/F, $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ for ADC-908G, $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ for ADC-908A/B, unless otherwise noted.

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
ACCURACY						
Resolution	N		8	-	-	Bits
Integral Nonlinearity	INL	A/E/G Grades	-1/2	-	+1/2	LSB
		B/F Grades	-3/4	-	+3/4	
Differential Nonlinearity	DNL	A/E/G Grades	-3/4	-	+3/4	LSB
		B/F Grades	-7/8	-	+7/8	
Gain Error	G_{FSE}	A/E/G Grades $T_A = +25^\circ\text{C}$	-3	-	+3	LSB
		$T_A = \text{Full Temp Range}$	-4.5	-	+4.5	
		B/F Grades $T_A = +25^\circ\text{C}$	-5	-	+5	
		$T_A = \text{Full Temp Range}$	-6.5	-	+6.5	
Offset Error	V_{ZSE}	A/E/G Grades $T_A = +25^\circ\text{C}$	-30	-	+30	mV
		$T_A = \text{Full Temp Range}$	-50	-	+50	
		B/F Grades $T_A = +25^\circ\text{C}$	-60	-	+60	
		$T_A = \text{Full Temp Range}$	-80	-	+80	
ANALOG INPUTS						
Resistance Mismatch B _{OFS} to A _{IN}	ΔR_{AB}		-1	-	+1	%
Input Resistance at V_{REF} (Note 1)	R_{REF}		5	-	15	k Ω
Input Resistance at B _{OFS} , A _{IN}	R_{BOFS} R_{AIN}		10	-	30	k Ω
Reference Voltage Range	V_{REF}	Specified Conversion Accuracy	-	-10	-	V
Reference Voltage Range	V_{REF}	Degraded Conversion Accuracy	-5	-	-15	V
Reference Current (Note 6)	I_{REF}	Conversion Complete Prior to Reset	-	-	2.4	mA
Nominal Analog Input Range						
Unipolar Mode	V_{INU}		-	0 to $ V_{REF} $	-	V
Bipolar Mode	V_{INB}		-	$- V_{REF} $ to $ V_{REF} $	-	V

ELECTRICAL CHARACTERISTICS at $V_{DD} = +5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ for ADC-908E/F, $0^{\circ}C \leq T_A \leq +70^{\circ}C$ for ADC-908G, $-55^{\circ}C \leq T_A \leq +125^{\circ}C$ for ADC-908A/B, unless otherwise noted.
Continued

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
LOGIC INPUTS						
Input HIGH Voltage RD, CS Inputs	V_{IH}		2.4	—	—	V
Input LOW Voltage RD, CS Inputs	V_{IL}		—	—	0.8	V
Input Current RD, CS Inputs	I_{IN}	$T_A = +25^{\circ}C$ $T_A = \text{Full Temp Range}$	—	—	1 10	μA
Input Capacitance RD, CS Inputs (Note 6)	C_{IN}		—	—	5	pF
Input HIGH Voltage, Clock Input	V_{IH}		2.4	—	—	V
Input LOW Voltage, Clock Input	V_{IL}		—	—	0.8	V
Input HIGH Current, Clock Input	I_{IH}		—	—	2	mA
Input LOW Current, Clock Input	I_{IL}	$T_A = +25^{\circ}C$ $T_A = \text{Full Temp Range}$	—	—	1 10	μA
LOGIC OUTPUTS						
Output HIGH Voltage BUSY, DB0-7	V_{OH}	$I_{SOURCE} = 40\mu A$	4.0	—	—	V
Output LOW Voltage BUSY, DB0-7	V_{OL}	$I_{SINK} = 1.6mA$	—	—	0.4	V
Floating Leakage Current, DB0-7	I_{LKG}	$T_A = +25^{\circ}C$ $T_A = \text{Full Temp Range}$	—	—	1 10	μA
Floating State Output Capacitance	C_{OZ}	(Note 6)	—	—	7	pF
POWER REQUIREMENTS						
Standby Current	I_{DD}	$V_{DD} = +4.75V$ to $+5.25V$	—	—	2.5	mA
DIGITAL INTERFACE TIMING						
CS Minimum Pulse Width (Note 6)	t_{CS}	$T_A = +25^{\circ}C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	60 50 90	— — —	— — —	ns
RD to CS Setup Time (Note 6)	t_{WCS}		0	—	—	ns
CS to BUSY Propagation Delay (Note 6)	t_{CBPD}	BUSY Load = 20pF $T_A = +25^{\circ}C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$ BUSY Load = 100pF $T_A = +25^{\circ}C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	— — — — — — —	— — — — — — —	120 100 150 150 120 200	ns
BUSY to RD Setup Time (Notes 2, 6)	t_{BSR}		0	—	—	ns
BUSY to CS Setup Time (Note 6)	t_{BSCS}		0	—	—	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

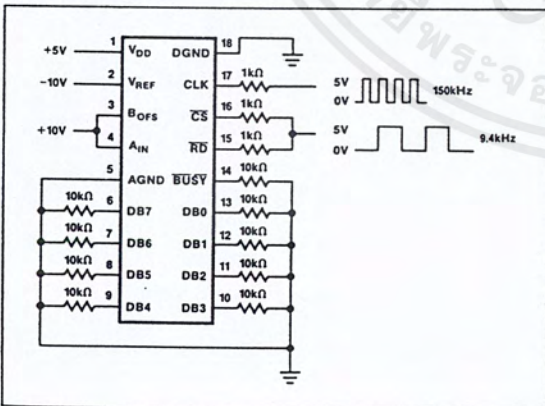
ELECTRICAL CHARACTERISTICS at $V_{DD} = +5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ C \leq T_A \leq +85^\circ C$ for ADC-908E/F, $0^\circ C \leq T_A \leq +70^\circ C$ for ADC-908G, $-55^\circ C \leq T_A \leq +125^\circ C$ for ADC-908A/B, unless otherwise noted.
Continued

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
Data Access Time (Note 6)	t_{RAD}	$C_L = 20pF$				
		$T_A = +25^\circ C$	—	—	140	
		$T_A = T_{MIN}$	—	—	100	
		$T_A = T_{MAX}$	—	—	200	
		$C_L = 100pF$				ns
		$T_A = +25^\circ C$	—	—	170	
Data Hold Time (Notes 3, 6)	t_{RHD}	$T_A = +25^\circ C$ (Note 3)	30	—	100	
		$T_A = T_{MIN}$	20	—	70	ns
		$T_A = T_{MAX}$	40	—	140	
\overline{CS} to \overline{RD} Hold Time (Note 6)	t_{RHCS}	$T_A = +25^\circ C$	—	—	200	
		$T_A = T_{MIN}$	—	—	120	ns
		$T_A = T_{MAX}$	—	—	250	
Reset Time Requirement (Note 6)	t_{RESET}	$T_A = +25^\circ C$	450	—	—	
		$T_A = \text{Full Temp. Range}$	500	—	—	ns
Conversion Time (Note 4)	$t_{CONVERT}$	Static RAM Mode				
		External Clock				
		$f = 1.35MHz$	—	—	6	μs
		ROM Mode				
(Notes 4, 5, 6)		Internal Clock	—	—	7	
\overline{RD} HIGH to \overline{BUSY} Propagation Delay, ROM Mode (Notes 4, 5, 6)	t_{WBPD}	$C_L = 20pF$				
		$T_A = +25^\circ C$	—	—	600	
		$T_A = T_{MIN}$	—	—	400	ns
		$T_A = T_{MAX}$	—	—	800	

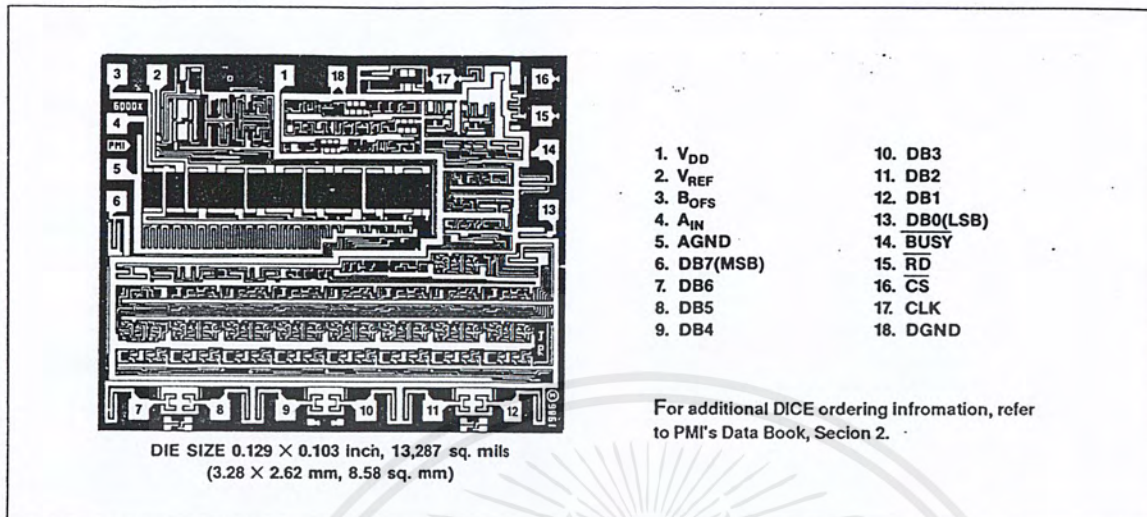
NOTES:

- For optimum gain accuracy over the full temperature range, the source resistance at pin 2 should be kept low.
- In ROM mode, \overline{RD} can go LOW prior to $\overline{BUSY} = \text{HIGH}$, but must not return HIGH until $\overline{BUSY} = \text{HIGH}$.
- Output loading 10pF. A 3k Ω pullup resistor to +5V is used for V_{OL} to High-Z, for V_{OH} to High-Z, a 3k Ω pulldown to GND is used. Measured to 0.5V output change.
- When using the ADC-908 internal oscillator, actual conversion time depends on clock resistor and capacitor as well as temperature.
- ROM interface mode conversion times are typically 1 μs longer than conversion times for other modes, but the ROM interface mode includes an automatic reset in the conversion time.
- Guaranteed but not tested.

BURN-IN CIRCUIT



DICE CHARACTERISTICS



WAFER TEST LIMITS at $V_{DD} = +5V$, $V_{REF} = -10.000V$, AGND = DGND = 0V, $T_A = +25^\circ C$, unless otherwise noted.

PARAMETER	SYMBOL	CONDITIONS	ADC-908 LIMIT	UNITS
STATIC ACCURACY				
Resolution	N		8	Bits MIN
Integral Nonlinearity	INL		$\pm 3/4$	LSB MAX
Differential Nonlinearity	DNL		$\pm 7/8$	LSB MAX
Gain Error	G_{FSE}		± 5	LSB MAX
Offset Error	V_{ZSE}		± 60	mV MAX
ANALOG INPUTS				
Resistance Mismatch B_{OFS} to A_{IN}	ΔR_{AB}		± 1	% MAX
Input Resistance at V_{REF}	R_{REF}		5/15	k Ω MIN/MAX
Input Resistance at B_{OFS} , A_{IN}	$R_{B_{OFS}}$, R_{IN}		10/30	k Ω MIN/MAX
DIGITAL INPUTS				
Input HIGH Voltage at \overline{RD} , \overline{CS} Inputs	V_{IH}		2.4	V MIN
Input LOW Voltage at \overline{RD} , \overline{CS} Inputs	V_{IL}		0.8	V MAX
Input Current \overline{RD} , \overline{CS} Inputs	I_{IN}		± 1	μA MAX
Input HIGH Voltage Clock Input	V_{IH}		2.4	V MIN
Input LOW Voltage Clock Input	V_{IL}		0.8	V MAX
Input HIGH Current Clock Input	I_{IH}		2	mA MAX
Input LOW Current Clock Input	I_{IL}		1	μA MAX

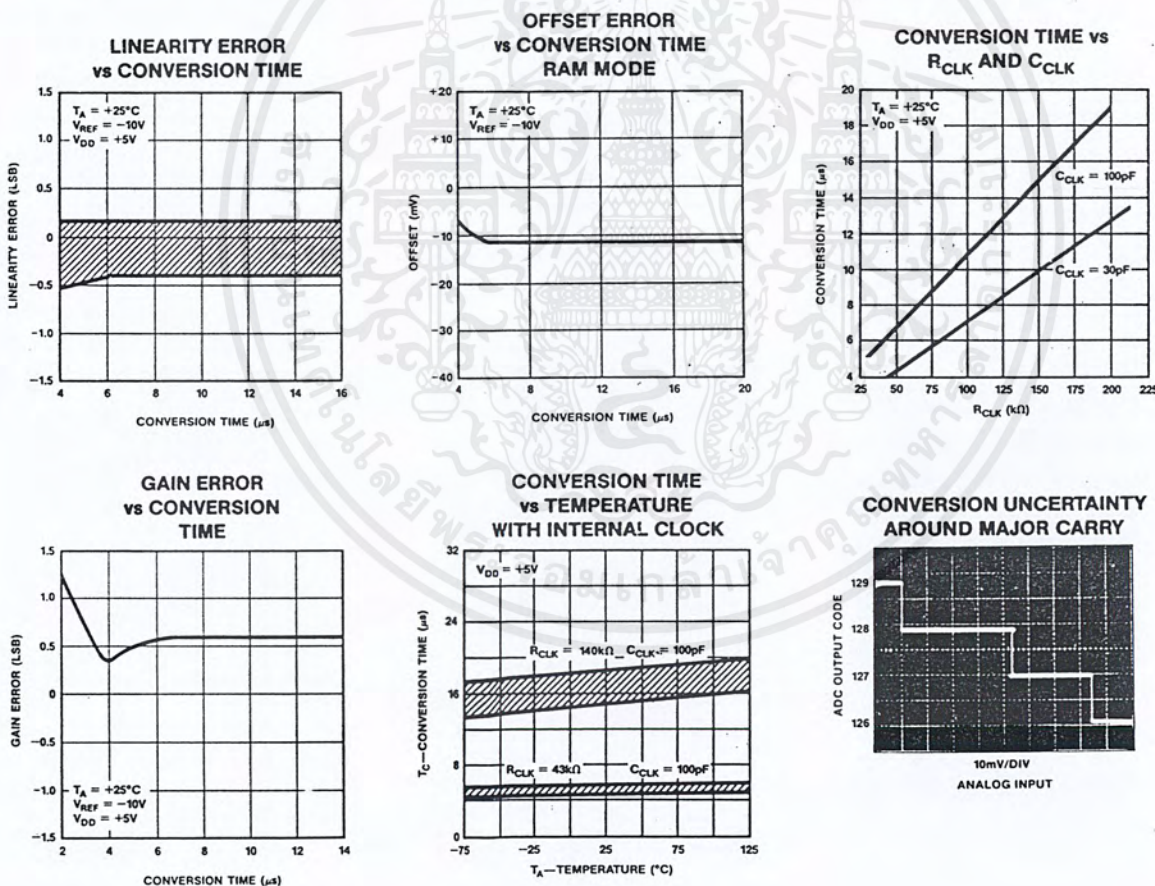
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAFER TEST LIMITS at $V_{DD} = +5V$, $V_{REF} = -10.000V$, $AGND = DGND = 0V$, $T_A = +25^\circ C$, unless otherwise noted. (Continued)

PARAMETER	SYMBOL	CONDITIONS	ADC-908	
			LIMIT	UNITS
DIGITAL OUTPUTS				
Output HIGH Voltage BUSY, DB0-7	V_{OH}	$I_{SOURCE} = 40\mu A$	4	V MIN
Output LOW Voltage BUSY, DB0-7	V_{OL}	$I_{SINK} = 1.6mA$	0.4	V MAX
Floating Leakage Current	I_{LKG}		1	μA
POWER REQUIREMENTS				
Standby Current	I_{DD}	$V_{DD} = +4.75V$ to $5.25V$	2.5	mA MAX
TIMING				
Conversion Time	$t_{CONVERT}$	Static RAM Mode, External Clock, $f = 1.35MHz$	6	μs MAX

NOTE:
Electrical tests are performed at wafer probe to the limits shown. Due to variations in assembly methods and normal yield loss, yield after packaging is not guaranteed for standard product dice. Consult factory to negotiate specifications based on dice lot qualification through sample lot assembly and testing.

TYPICAL PERFORMANCE CHARACTERISTICS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL CIRCUIT INFORMATION

The ADC-908 is an 8-bit analog-to-digital converter which uses a successive approximation technique to convert an unknown analog input into a digital code output. The control logic inputs allow easy interface to most microprocessors while three-state outputs allow direct connection to the data bus. Most applications require only passive RC clock components, a $-10V$ reference, and a $+5V$ power supply. The RC-timed internal clock may be used, or an external clock may be applied to the ADC to maximize performance.

When a Start Conversion command is applied to the \overline{CS} or \overline{RD} inputs (see Operating Descriptions for details), \overline{BUSY} goes LOW indicating a conversion in progress. \overline{BUSY} may be used as an interrupt to halt the controlling microprocessor during conversion or may be polled to prevent premature data reads.

Starting with the most significant bit (MSB), each successive bit in the DAC is turned on (see Figure 1). The comparator then decides if the DAC output is less than or greater than the signal being converted, and that bit is latched on or off, respectively, before proceeding to the next lower bit and repeating the cycle. When all eight bits have been tested, \overline{BUSY} goes HIGH, signaling a completed conversion.

Under control of the \overline{RD} input, the three-state data outputs (D0-D7) change from high-impedance to presenting the new conversion results to the data bus. Following the data read, \overline{RD} returns HIGH resetting the SAR to 1000 0000 and preparing the ADC for its next conversion.

PIN FUNCTIONS

NOTE: For greater detail on digital input functions, consult Truth Tables and Timing Diagrams.

- Pin 1. V_{DD} Power Supply input, $+5V$.
 Pin 2. V_{REF} Voltage Reference input, nominal $-10V$.
 Pin 3. B_{OFS} Bipolar Offset input. $+10V$ input for bipolar mode operation, tie to V_{IN} for unipolar mode operation.
 Pin 4. A_{IN} Analog Input. $0V$ to $+10V$ in unipolar mode, $-10V$ to $+10V$ in bipolar mode.

- Pin 14. \overline{BUSY} Conversion status output. \overline{BUSY} indicates conversion in progress by going LOW at start of conversion and returning HIGH at end of conversion. May be used to interrupt controlling microprocessor or to gate control inputs.
 Pin 15. \overline{RD} READ input. Used to read data (on falling edge) and to reset converter (on rising edge).
 Pin 16. \overline{CS} Chip Select input. Asserted to allow ADC operation. Starts conversion when converter is in reset condition. Note: Holding \overline{CS} HIGH will not prevent a rising edge on \overline{RD} from resetting the converter.
 Pin 17. CLK External clock input/internal clock RC timing input.

APPLICATIONS INFORMATION

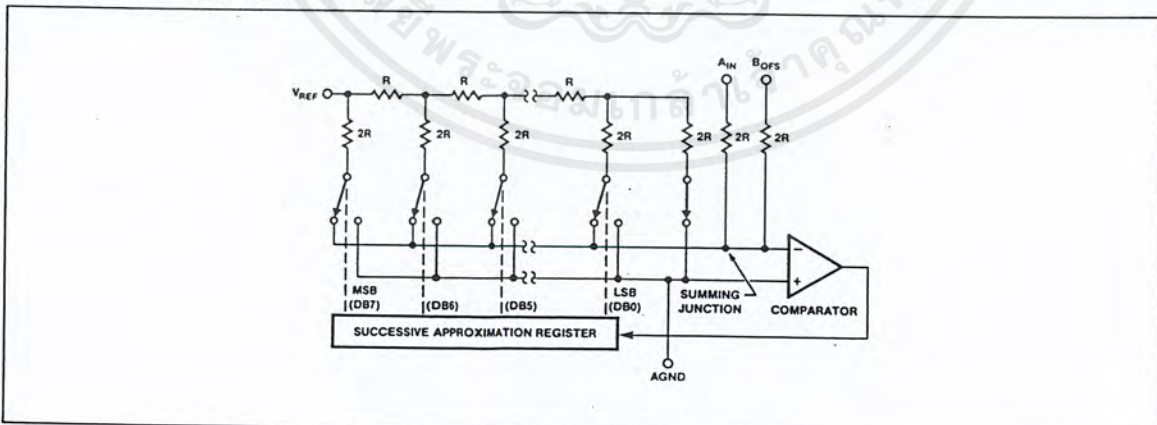
The ADC-908 may be interfaced as if it were a static RAM, a ROM, or a slow-memory device. Each of these interface modes has its own timing and software requirements as described below. These requirements must be rigidly met, as improper timing may cause the ADC-908 to change modes.

HOW TO CHOOSE AN OPERATING MODE

The static-RAM interface mode offers advantages in a tightly controlled hardware and software environment, where the relationship between WRITE and READ instruction pairs is certain. As long as minimum timing is satisfied, converted data may be read at any convenient time after conversion. The use of separate commands to start a conversion, and then read the results, is conceptually easy. However, if the software is subject to uncontrolled modifications, then the paired relationship between WRITE and READ instructions may be lost. Resulting software bugs may result in converted data of unknown age, or altogether invalid data being read.

By contrast, the ROM mode may be more resistant to software bugs. As long as minimum timing is satisfied, each READ instruction obtains new, valid data. However, since the data

FIGURE 1: D/A Converter Used in ADC-908



output at any previous READ instruction is obtained from a conversion performed just after the previous READ instruction, data may be out-of-date. To be sure of obtaining up-to-date data, READ instructions may be coded in pairs (with some NOPs between them); use only the data from the second READ in each pair. The first READ starts the conversion, acting as a substitute for the static-RAM mode WRITE command; the second READ gets the results. The advantage of the ROM mode is the use of a single command, rather than the alternating READ-WRITE required by static-RAM mode.

The slow-memory mode is the simplest mode of all. It is the method of choice where compact coding is essential, or where software bugs are a hazard. In this mode, a single READ instruction will initiate a data conversion, interrupt the microprocessor until completion (WAIT states are introduced), then read the results. If the system throughput tolerates WAIT states, and the hardware is correct, then the slow-memory mode is virtually immune to subsequent software modifications.

OPERATING DESCRIPTION: STATIC-RAM MODE

In this mode, input \overline{CS} is derived from the ADC-908 address decoder, and input \overline{RD} is derived from an active-LOW memory READ signal. (See Figure 2.)

To start a conversion, execute a memory WRITE to the ADC-908. The completed conversion data is obtained by executing a memory READ to the ADC-908. During conversion, output $BUSY$ will be LOW. Do not attempt to read data until $BUSY$ returns HIGH. The required minimum time between WRITE and READ is usually obtained by including one or more NOP or other program instructions. The use of branch or conditional commands between the WRITE and READ instructions is not recommended due to the possibility of software bugs.

It is important that the WRITE and READ commands be alternately executed. A WRITE instruction has no effect unless the results of the previous WRITE have already been read. Once data has been read, the ADC-908 is internally reset. In other words, two or more READ operations cannot be used in

succession, since only the first READ will produce valid data. A new conversion must be started using WRITE, and the conversion must be completed, before a new READ will produce valid data.

TABLE 1: Truth Table, Static RAM Mode

INPUTS		OUTPUTS		ADC-908 OPERATION
CS	RD	BUSY	DB7-DB0	
L	H	H	HIGH-Z	Start Convert (Write Cycle)
L	\downarrow	H	HIGH-Z to DATA	Read Data (Read Cycle)
L	\uparrow	H	DATA to HIGH-Z	Reset Converter
H	X (Note 1)	X	HIGH-Z	No Effect (Not Selected)
L	H	L	HIGH-Z	No Effect (Converter Busy)
L	\downarrow	L	HIGH-Z	No Effect (Converter Busy)
L	\uparrow (Note 1)	L	HIGH-Z	Conversion Error Not Allowed

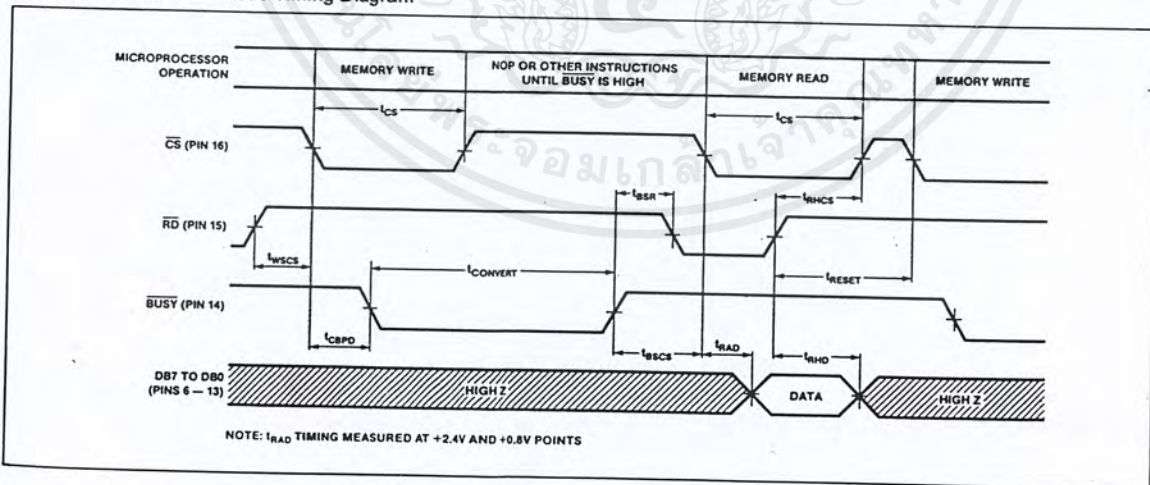
NOTE 1: If \overline{RD} goes LOW to HIGH, the ADC is internally reset, regardless of the states of \overline{CS} or $BUSY$.

OPERATING DESCRIPTION: ROM MODE

In ROM mode, input \overline{CS} is tied LOW, and input \overline{RD} is derived from the ADC-908 address decoder. To satisfy timing, it is recommended that the decoder be enabled by a system MEMRD (8080), VMA (6800), or similar strobe. (See Figure 3.)

In ROM mode, data is read by executing a READ instruction to the ADC-908 address. At the conclusion of the READ instruction, the ADC-908 automatically resets itself and then proceeds to perform a new data conversion. Output $BUSY$ is LOW during conversion. A new READ instruction to the ADC-908 must not be executed until $BUSY$ returns HIGH.

FIGURE 2: Static RAM Mode Timing Diagram



ANALOG INPUT VOLTAGE

The ADC-908 unipolar operation is obtained when the analog input voltage is between 0V and $|V_{REF}|$. With the A_{IN} and B_{OFS} pins tied together, input 0V will correspond to code 0000 0000, and input full-scale will correspond to code 1111 1111.

Bipolar operation is obtained by using the B_{OFS} input to offset the A_{IN} input voltage. For example, with $V_{REF} = -10V$, an offset voltage of +10V may be applied to B_{OFS} . The analog signal range will then be $-10V$ to $+10V$ at A_{IN} . Code 0000 0000 will correspond to $-10V$, and positive full scale will be code 1111 1111. Calibration may be performed using trimmers in series with A_{IN} and B_{OFS} . (See Figure 8).

Another method of obtaining bipolar operation is to use an op-amp with gain = $-1/2$, to sum the analog signal with the reference voltage. With a $-10V$ reference and $-10V$ to $+10V$ analog signal, the op amp output will then be 0V to $+10V$. This signal is then treated as an ordinary unipolar input to the ADC-908. With this arrangement, input $+10V$ corresponds to code 0000 0000, and negative full-scale corresponds to code 1111 1111.

UNIPOLAR BINARY OPERATION

Figure 7 shows the analog circuit connections for unipolar operation. The REF-08 supplies the necessary $-10V$ reference input.

Calibration for offset should be made before gain calibration is attempted.

Offset calibration must be performed in the signal conditioning circuitry which drives the A_{IN} input.

To adjust offset:

- 1) Apply $-39.1mV$ (1 LSB) to the input of the buffer amplifier driving A_{IN} .
- 2) While performing continuous conversions, adjust the buffer amplifier's offset adjustment potentiometer until DB7 to DB1 are LOW and DB0 (LSB) flickers.

Following offset calibration, full scale gain can be calibrated:

- 1) Apply $-9.961V$ to the input of the buffer amplifier.

- 2) While performing continuous conversions, adjust the reference trim pot until DB7 to DB1 are HIGH, and DB0 (LSB) flickers.

BIPOLAR OPERATION

Offset Binary—Figure 8 shows a circuit for offset binary bipolar operation. Offset correction should be made at the buffer amplifier driving A_{IN} . Gain error correction should be accomplished by adjusting V_{REF} .

To calibrate this circuit:

- 1) Adjust R1 until $V_{REF} = -10.00V$.
- 2) Adjust R2 and R3 to their mid-points.
- 3) Apply $+10.000V$ to the input buffer amplifier.
- 4) While performing continuous conversions, adjust R2 until DB7 to DB1 are LOW and DB0 (LSB) flickers.
- 5) Ground the input of the input buffer circuit.
- 6) While performing continuous conversions, adjust R3 until the ADC's output code flickers between 0111 1111 and 1000 0000.
- 7) Apply $-10.000V$ to the signal input.
- 8) While performing continuous conversions, adjust R1 until DB7 to DB1 are LOW and the DB0 (LSB) flickers.
- 9) Apply $+9.922V$ to the signal input.
- 10) If the ADC output code is not 1111 1110 ± 1 bit, repeat the calibration procedure, omitting step 1.

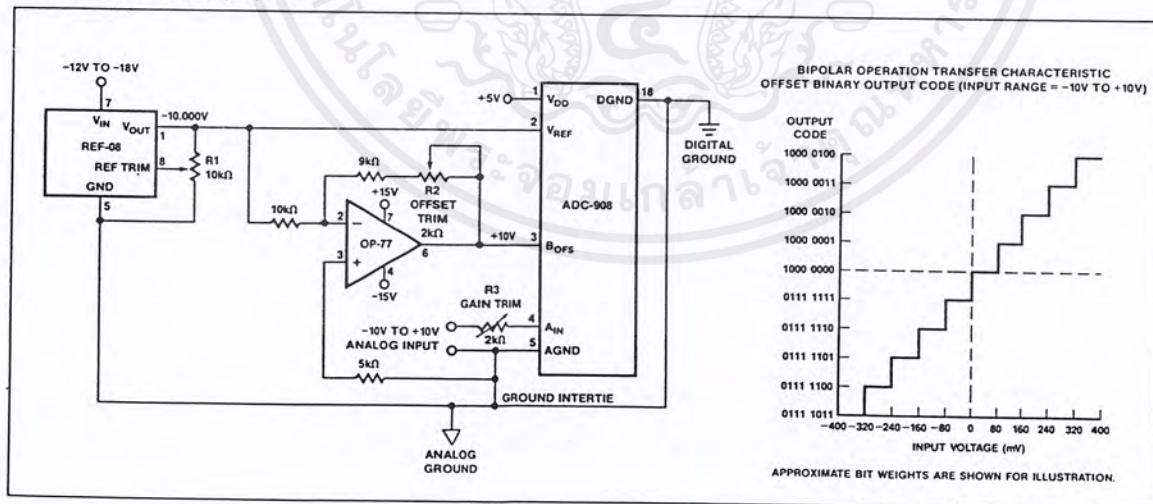
Complementary Offset Binary—Figure 9 shows a complementary offset binary circuit. In this bipolar mode, the $+10V$ to $-10V$ analog input is conditioned to a 0 to $+10V$ signal range for normal unipolar conversion.

In calibrating this circuit, adjust offset before gain.

Offset Adjustment:

- 1) Adjust R1 until $V_{REF} = -10.000V$.
- 2) Adjust R3 to its mid-point.
- 3) Adjust R2 until its tap is at 0V.
- 4) Ground the analog input.
- 5) While performing continuous conversions, adjust R2 until the ADC output flickers between 0111 1111 and 1000 0000.

FIGURE 8: Offset Binary Operation



Gain Adjustment:

- 1) Apply +9.922V across the analog input.
- 2) While performing continuous conversions, adjust R3 until DB7 to DB1 are HIGH and DB0 (LSB) flickers.

DIGITAL CONSIDERATIONS

Control Timing—Fresh data from a recent conversion must be read before beginning a new conversion. Following the data READ, as \overline{RD} goes HIGH, it resets the SAR and clears the data from the previous conversion.

The timing restrictions detailed in the interface timing diagrams must be observed to prevent the ADC-908 from changing interface modes. For example, if \overline{CS} is held LOW too long while in RAM mode, the converter will change to ROM mode and initiate a new conversion.

Logic Deglitching—Unrelated activity on the address bus may cause unexpected glitch inputs to the ADC. The glitches may cause unwanted READs, resets, or conversions. In ROM or RAM modes, these may be avoided by gating the address decode logic with \overline{RD} or \overline{WR} (8080) or \overline{VMA} (6800). In slow-memory mode, ALE (8085) or SYNC (8080) may be used to latch the address.

Initialization—Following power-up, the SAR is in an unknown state. Executing a memory READ (disregard the data) will reset the ADC.

ANALOG CONSIDERATIONS

Analog Input Impedances—Low impedance sources must be used to drive the V_{REF} , A_{IN} , and B_{OFS} inputs. Excessive source

impedances may cause errors due to the loading effects of the inputs' finite impedances.

Ground Management—AGND and DGND pins should be connected at or near the ADC to minimize noise effects. If the two grounds cannot be connected near the ADC, the grounds should be clamped with back-to-back Schottky diodes between the AGND and DGND pins.

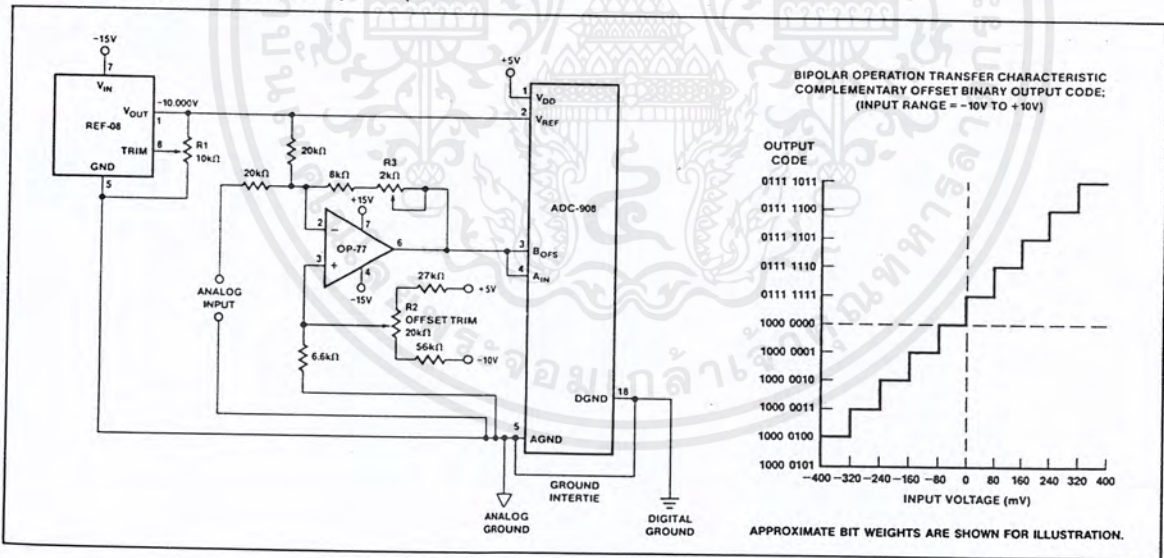
Offset Correction—Conversion offset errors may be corrected by counter-offsetting the buffer amplifier driving A_{IN} . This offset correction may be accomplished by applying a correction current to the buffer's summing junction or by tapping a voltage divider sitting between V_{DD} and V_{REF} , and applying this tap voltage to the noninverting input of the buffer.

Ratiometric Operation—The R-2R type DAC in the ADC-908 permits ratiometric operation of the ADC. Performance degradation may, however, occur as V_{REF} varies from $-10.000V$. This decrease in performance is due to comparator limitations including offset-voltage, gain, and input noise.

The ADC-908 uses the reference as a power supply for the comparator to increase speed and accuracy. Reference voltages of a magnitude less than $-9V$ must be avoided for accurate comparator operation. For best accuracy, the use of a $0.1\mu F$ bypass capacitor from V_{REF} (Pin 2 to AGND) is recommended.

Power Supply Bypassing—For best accuracy, V_{DD} (Pin 1) should be bypassed to AGND with a $0.1\mu F$ capacitor.

FIGURE 9: Complementary Offset Bipolar Operation



ANALOG INPUT VOLTAGE

The ADC-908 unipolar operation is obtained when the analog input voltage is between 0V and $|V_{REF}|$. With the A_{IN} and B_{OFS} pins tied together, input 0V will correspond to code 0000 0000, and input full-scale will correspond to code 1111 1111.

Bipolar operation is obtained by using the B_{OFS} input to offset the A_{IN} input voltage. For example, with $V_{REF} = -10V$, an offset voltage of +10V may be applied to B_{OFS} . The analog signal range will then be $-10V$ to $+10V$ at A_{IN} . Code 0000 0000 will correspond to $-10V$, and positive full scale will be code 1111 1111. Calibration may be performed using trimmers in series with A_{IN} and B_{OFS} . (See Figure 8).

Another method of obtaining bipolar operation is to use an op-amp with gain = $-1/2$, to sum the analog signal with the reference voltage. With a $-10V$ reference and $-10V$ to $+10V$ analog signal, the op amp output will then be 0V to $+10V$. This signal is then treated as an ordinary unipolar input to the ADC-908. With this arrangement, input $+10V$ corresponds to code 0000 0000, and negative full-scale corresponds to code 1111 1111.

UNIPOLAR BINARY OPERATION

Figure 7 shows the analog circuit connections for unipolar operation. The REF-08 supplies the necessary $-10V$ reference input.

Calibration for offset should be made before gain calibration is attempted.

Offset calibration must be performed in the signal conditioning circuitry which drives the A_{IN} input.

To adjust offset:

- 1) Apply $-39.1mV$ (1 LSB) to the input of the buffer amplifier driving A_{IN} .
- 2) While performing continuous conversions, adjust the buffer amplifier's offset adjustment potentiometer until DB7 to DB1 are LOW and DB0 (LSB) flickers.

Following offset calibration, full scale gain can be calibrated:

- 1) Apply $-9.961V$ to the input of the buffer amplifier.

- 2) While performing continuous conversions, adjust the reference trim pot until DB7 to DB1 are HIGH, and DB0 (LSB) flickers.

BIPOLAR OPERATION

Offset Binary—Figure 8 shows a circuit for offset binary bipolar operation. Offset correction should be made at the buffer amplifier driving A_{IN} . Gain error correction should be accomplished by adjusting V_{REF} .

To calibrate this circuit:

- 1) Adjust R1 until $V_{REF} = -10.00V$.
- 2) Adjust R2 and R3 to their mid-points.
- 3) Apply $+10.000V$ to the input buffer amplifier.
- 4) While performing continuous conversions, adjust R2 until DB7 to DB1 are LOW and DB0 (LSB) flickers.
- 5) Ground the input of the input buffer circuit.
- 6) While performing continuous conversions, adjust R3 until the ADC's output code flickers between 0111 1111 and 1000 0000.
- 7) Apply $-10.000V$ to the signal input.
- 8) While performing continuous conversions, adjust R1 until DB7 to DB1 are LOW and the DB0 (LSB) flickers.
- 9) Apply $+9.922V$ to the signal input.
- 10) If the ADC output code is not 1111 1110 ± 1 bit, repeat the calibration procedure, omitting step 1.

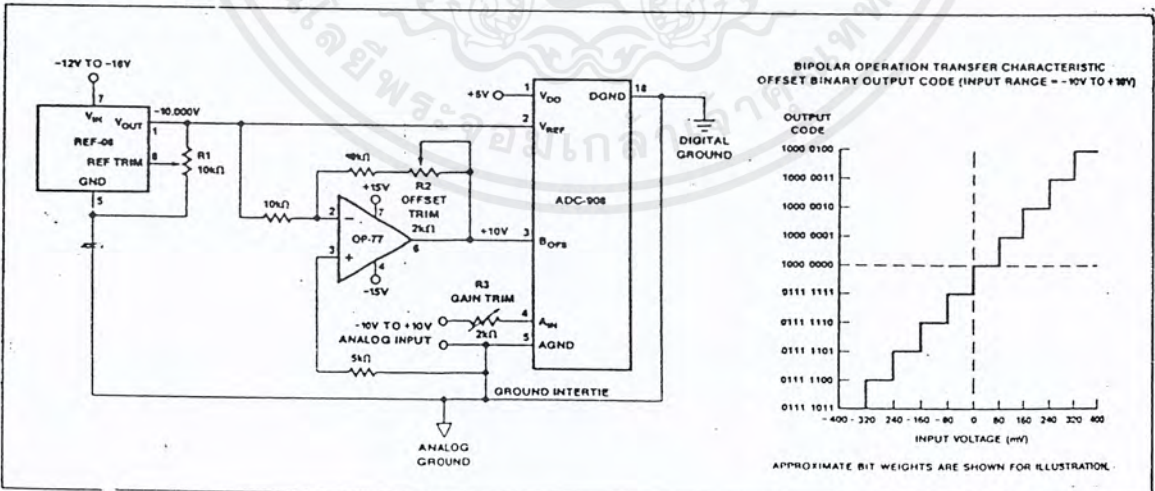
Complementary Offset Binary—Figure 9 shows a complementary offset binary circuit. In this bipolar circuit, the $+10V$ to $-10V$ analog input is conditioned to a 0 to $+10V$ signal range for normal unipolar conversion.

In calibrating this circuit, adjust offset before gain.

Offset Adjustment:

- 1) Adjust R1 until $V_{REF} = -10.000V$.
- 2) Adjust R3 to its mid-point.
- 3) Adjust R2 until its tap is at 0V.
- 4) Ground the analog input.
- 5) While performing continuous conversions, adjust R2 until the ADC output flickers between 0111 1111 and 1000 0000.

FIGURE 8: Offset Binary Operation



Gain Adjustment:

- 1) Apply +9.822V across the analog input.
- 2) While performing continuous conversions, adjust R3 until DB7 to DB1 are HIGH and DB0 (LSB) flickers.

DIGITAL CONSIDERATIONS

Control Timing—Fresh data from a recent conversion must be read before beginning a new conversion. Following the data READ, as RD goes HIGH, it resets the SAR and clears the data from the previous conversion.

The timing restrictions detailed in the interface timing diagrams must be observed to prevent the ADC-908 from changing interface modes. For example, if CS is held LOW too long while in RAM mode, the converter will change to ROM mode and initiate a new conversion.

Logic Deglitching—Unrelated activity on the address bus may cause unexpected glitch inputs to the ADC. The glitches may cause unwanted READs, resets, or conversions. In ROM or RAM modes, these may be avoided by gating the address decode logic with RD or WR (8080) or VMA (6800). In slow-memory mode, ALE (8085) or SYNC (8080) may be used to latch the address.

Initialization—Following power-up, the SAR is in an unknown state. Executing a memory READ (disregard the data) will reset the ADC.

ANALOG CONSIDERATIONS

Analog Input Impedances—Low impedance sources must be used to drive the VREF, AIN, and BOFS inputs. Excessive source

impedances may cause errors due to the loading effects of the inputs' finite impedances.

Ground Management—AGND and DGND pins should be connected at or near the ADC to minimize noise effects. If the two grounds cannot be connected near the ADC, the grounds should be clamped with back-to-back Schottky diodes between the AGND and DGND pins.

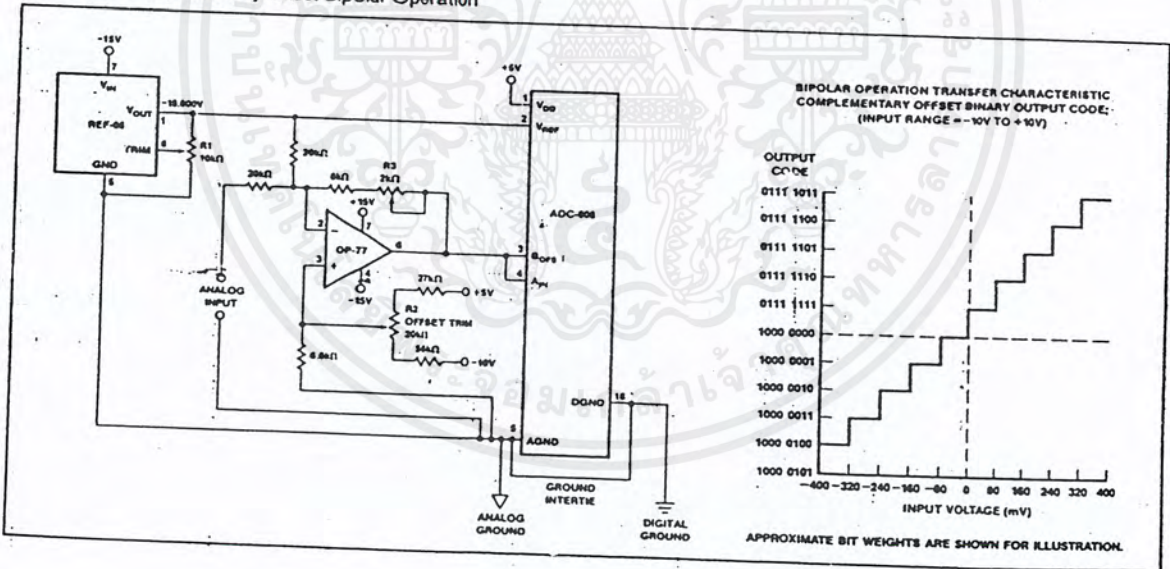
Offset Correction—Conversion offset errors may be corrected by counter-offsetting the buffer amplifier driving AIN. This offset correction may be accomplished by applying a correction current to the buffer's summing junction or by tapping a voltage divider sitting between VDD and VREF, and applying this tap voltage to the noninverting input of the buffer.

Ratiometric Operation—The R-2R type DAC in the ADC-908 permits ratiometric operation of the ADC. Performance degradation may, however, occur as VREF varies from -10.000V. This decrease in performance is due to comparator limitations including offset-voltage, gain, and input noise.

The ADC-908 uses the reference as a power supply for the comparator to increase speed and accuracy. Reference voltages of a magnitude less than -9V must be avoided for accurate comparator operation. For best accuracy, the use of a 0.1µF bypass capacitor from VREF (Pin 2 to AGND) is recommended.

Power Supply Bypassing—For best accuracy, VDD (Pin. 1) should be bypassed to AGND with a 0.1µF capacitor.

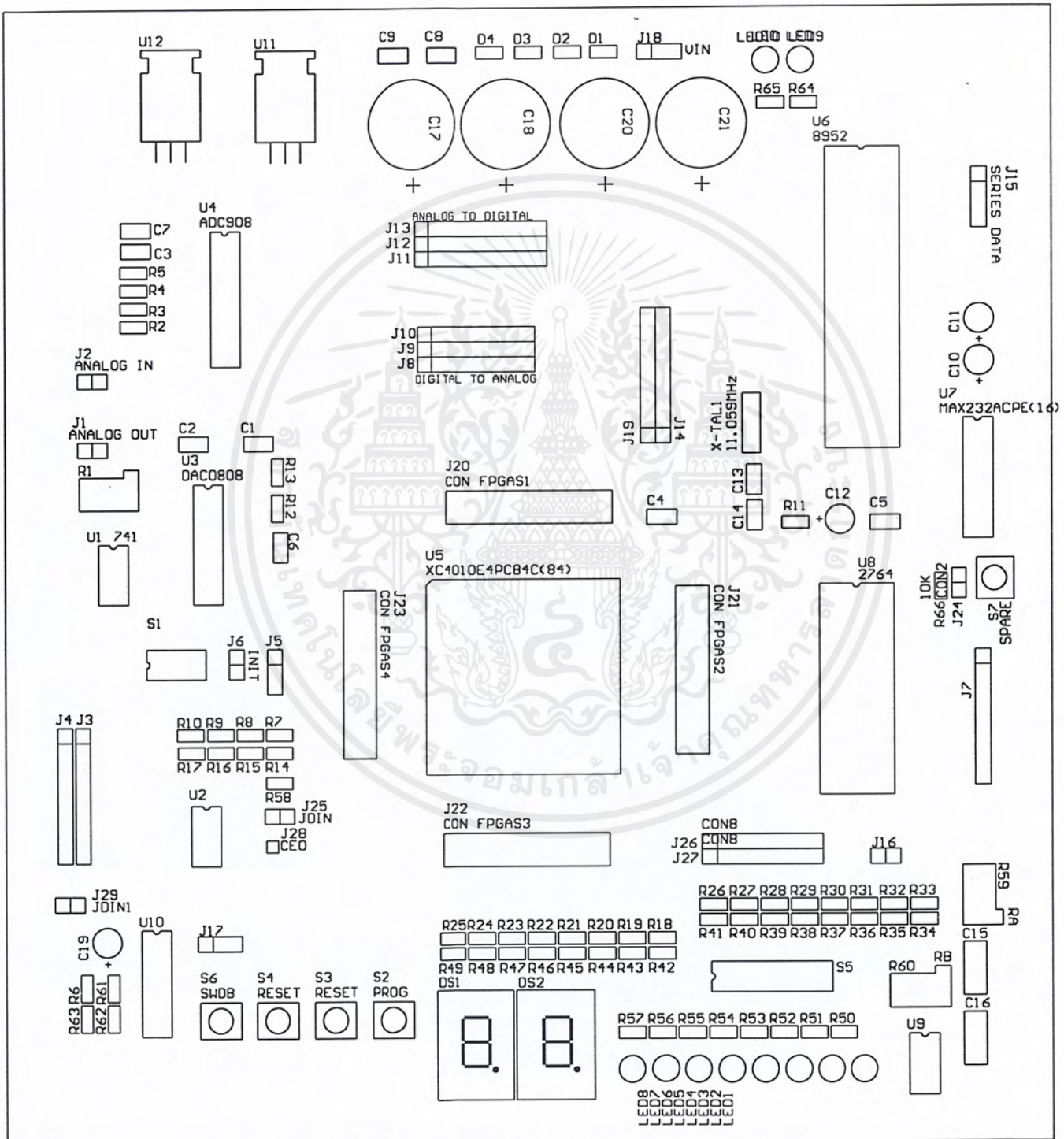
FIGURE 9: Complementary Offset Bipolar Operation





ภาคผนวก ง
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ ง.1 ด้านหน้าของชุดฝึก
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ของสวิตช์ และจัมป์เปอร์ต่างๆ บนชุดฝักออกเบรบบระบบดิจิทัลโดยใช้อุปกรณ์ FPGAs ร่วมกับไมโครคอนโทรลเลอร์ MCS-51 ได้แสดงไว้ในตาราง

ตารางที่ ง.1 หน้าที่ของจัมป์เปอร์ และสวิตช์ต่างๆ บนชุดฝัก

ตำแหน่ง	หน้าที่ของจัมป์เปอร์ และสวิตช์
J1	จัมป์เปอร์ของแอนะล็อกเอาต์พุตจากวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
J2	จัมป์เปอร์ของแอนะล็อกอินพุตจากวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
J3	จัมป์เปอร์สำหรับต่อกับขั้วต่อ Read Back ของสาย Xchecker
J4	จัมป์เปอร์สำหรับต่อกับขั้วต่อ Program ของสาย Xchecker
J5	จัมป์เปอร์ใช้ในการเลือกสัญญาณนาฬิกาของระบบที่ส่งเข้าควบคุมการทำงานของ Xchecker ขณะที่ทำการ Read Back
J6	จัมป์เปอร์เพื่อเลือกว่าต้องการต่อ ขา INIT ของ IC2 กับ FPGAs
J7	จัมป์เปอร์เพื่อไว้สำหรับต่อ FPGAs กับอุปกรณ์ภายนอกซึ่งมีทั้งหมด 9 ขา
J8	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกซึ่งถ้าต้องการต่อกั้เพียงใช้จัมป์เปอร์ต่อ J8 กับ J9
J9	จัมป์เปอร์ที่ต่อกับขาข้อมูลของวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก
J10	จัมป์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับวงจรสัญญาณแปลงดิจิทัลเป็นสัญญาณแอนะล็อกซึ่งถ้าต้องการต่อกั้เพียงใช้จัมป์เปอร์ต่อ J10 กับ J9
J11	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลซึ่งถ้าต้องการต่อกั้เพียงใช้จัมป์เปอร์ต่อ J11 กับ J12
J12	จัมป์เปอร์ที่ต่อกับขาข้อมูลของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
J13	จัมป์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลซึ่งถ้าต้องการต่อกั้เพียงใช้จัมป์เปอร์ต่อ J13 กับ J12
J14	จัมป์เปอร์ที่ต่อกับ MCS-51 ซึ่งมีไว้สำหรับต่อกับ FPGAs ซึ่งถ้าต้องการต่อกั้เพียงใช้จัมป์เปอร์ต่อ J14 กับ J19
J15	คอนเน็คเตอร์ที่ใช้การติดต่อข้อมูลแบบอนุกรม มาตรฐาน RS232
J16	จัมป์เปอร์ที่ใช้การเลือกว่าจะต่อวงจรผลิตสัญญาณรูปคลื่นสี่เหลี่ยมให้ต่อกับ FPGAs
J17	จัมป์เปอร์ที่ใช้การเลือกว่าจะต่อวงจรตีเบาะสวิตช์ให้ต่อกับ FPGAs
J18	คอนเน็คเตอร์ที่ไว้ต่อกับแหล่งจ่ายแรงดันไฟ

ตำแหน่ง	หน้าที่ของจัมป์เปอร์ และสวิตช์ต่างๆ
J19	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับ MCS-51 ซึ่งถ้าต้องการต่อก็เพียงใช้จัมป์เปอร์ต่อ J14 กับ J19
J20	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรมานอก
J21	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรมานอก
J22	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรมานอก
J23	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับต่อกับอุปกรณ์หรือวงจรมานอก
J24	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งาน Spare สวิตช์
J25	จัมป์เปอร์เพื่อเลือกความต้องการต่อ ขา 1 (DATA) ของ IC2 กับ FPGAs
J26	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งานคิปสวิตช์ 8 หลักซึ่งใช้งานร่วมกับ J27
J27	จัมป์เปอร์ที่ต่อกับ FPGAs ซึ่งมีไว้สำหรับเลือกใช้งานคิปสวิตช์ 8 หลักซึ่งใช้งานร่วมกับ J26
J28	จัมป์เปอร์ที่ต่อกับ ขา 6 (CEO) ของ IC2
J29	จัมป์เปอร์ที่ใช้ว่าต้องการต่อกับ ขา 6 (DIN) ของ J4 กับ FPGAs
S1	สวิตช์ที่ใช้เลือกโหมดการ โปรแกรมของ FPGAs
S2	สวิตช์ที่ใช้ทำการสั่งให้ FPGAs ทำการอยู่ในสภาวะรอรับการ โปรแกรม
S3	สวิตช์ที่ใช้ทำการสั่งให้ FPGAs ทำการรีเซต
S4	สวิตช์ที่ใช้ทำการสั่งให้ MCS-51 ทำการรีเซต
S5	สวิตช์ที่ใช้ป้อนลอจิกให้กับ FPGAs เพื่อทำการทดลอง
S6	สวิตช์ที่ใช้ป้อนสัญญาณขอบขาขึ้นหรือขอบขาลงให้กับ FPGAs
S7	สวิตช์ที่ไว้สำหรับเป็นสวิตช์ใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- กำธร พาณิชปฐมพงษ์ และชูชัย ธนสารตั้งเจริญ. ภาษาแอสเซมบลี 80286/80386. กรุงเทพฯ : ฟิสิกส์เซ็นเตอร์. ม.ป.ป.
- ชวลิต สุวรรณศิริ และคณะ. “ชุดทดลองวงจรดิจิทัลควบคุมและแสดงผลโดยคอมพิวเตอร์.” ปรินญาณิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541
- นภัทร วัจนเทพินทร์. วงจรดิจิทัล (ภาคปฏิบัติ)พิมพ์ครั้งที่3. กรุงเทพฯ : สถาบันเทคโนโลยีราชมงคลวิทยาเขตนนทบุรี. 2540
- บริษัทซีเอ็ดยูเคชั่น จำกัด มหาชน. “รู้จักไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C52 และ AT89C1051/2051.” HOBBY ELECTRONIC. (95) : หน้า 48 – 53. 2543
- อุดม จินประดับ. ไมโครคอนโทรลเลอร์ MCS-51. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2541
- Sandige, R.S. **Modern Digital Design**. Singapore: McGraw-Hill. 1990
- Wolf, W.H. **Modern VLSI design system on silicon**. W.H. 2 nd ed. New Jersey : A Simon & Schuster Company. 1998

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวสุรีย์พร วงษ์คำหาญ
วันเดือนปีเกิด	31 มีนาคม 2522
สถานที่เกิด	จังหวัดหนองคาย
ภูมิลำเนา	29 หมู่ 4 ต. หินโงม อ. เมือง จ. หนองคาย 43000
ที่อยู่ปัจจุบัน	29 หมู่ 4 ต. หินโงม อ. เมือง จ. หนองคาย 43000
โทรศัพท์	-
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนหิน โงมวิทยาคม
มัธยมศึกษาตอนต้น	โรงเรียนหิน โงมพิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคหนองคาย
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคหนองคาย
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	ทุนกู้ยืมรัฐบาล
คติพจน์	เก็บประสบการณ์ในวันนี้ เพื่อเป็นบทเรียนในวันหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานิพนธ์	นายสุทธิพงษ์ จันทะรัง
วันเดือนปีเกิด	2 ธันวาคม 2521
สถานที่เกิด	จังหวัดนครพนม
ภูมิลำเนา	252 / 4 ม. 13 บ. หนงม้นปลาต. ธาดูเชิงชุม อ. เมือง จ. สกลนคร 47000
ที่อยู่ปัจจุบัน	159 / 204 หมู่บ้านพาราไคร์เทอเรซ ถ.ร่มเกล้า แขวงคลองสามประเวศ เขตลาดกระบัง 10520
โทรศัพท์	02 - 7379202
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนสุนทรวิจิตร (บำรุงวิทยา)
มัธยมศึกษาตอนต้น	โรงเรียนสกลราชวิทยานุกูล
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคสกลนคร
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตพระนครเหนือ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	ทุนกู้ยืมรัฐบาล
คติพจน์	ไม่มีใครให้อะไรกับเรา มีแต่ตัวเราที่จักแสวงหาเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวกรรณา แยมพราย
วันเดือนปีเกิด	10 มีนาคม 2522
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนา	855 ถ. สืบศิริ ซ. สืบศิริ19 อ. เมือง จ. นครราชสีมา 30000
ที่อยู่ปัจจุบัน	855 ถ. สืบศิริ ซ. สืบศิริ19 อ. เมือง จ. นครราชสีมา 30000
โทรศัพท์	(044) 273912
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเมืองนครราชสีมา
มัธยมศึกษาตอนต้น	โรงเรียนสุนารีวิทยา
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขต ภาคตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
กติพนธ์	ฝันให้ไกล แล้วไปให้ถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้