ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ การใช้ FPGAs สร้างตัวประมวลผลกลางขนาด 8 บิต

FPGAs Implementation of a Central Processing Unit 8 Bits

ชื่อนักศึกษา

1. นายจุติ	จุติประเสริฐ	รหัสประจำตัว	42035292
2. นางสาวปิยะวรรณ	คำน้ำนอง	รหัสประจำตัว	42035310
3. นายอรุณชัย	ศุภฤกษ์จันทราภา	รหัสประจำตัว	42035321

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์สุชิน อาจหาญ

อาจารย์ที่ปรึกษาร่วม อาจารย์กิติพงค์ มะโน

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์สุชิน	อาจหาญ	.....
2. อาจารย์กิติพงค์	มะโน	.....
3. อาจารย์อำพล	ทองระอา	.....
4. อาจารย์ไพบูลย์	พวงวงศ์ตระกูล	.....
5. อาจารย์สุระชัย	พิมพ์สาลี	.....

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 10.30 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธำรง)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 5 เดือน ธ.ค. พ.ศ. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

การใช้ FPGAs สร้างตัวประมวลผลกลางขนาด 8 บิต

FPGAs IMPLEMENTATION OF

A CENTRAL PROCESSING UNIT 8 BITS

นายจุติ จุติประเสริฐ
นางสาวปิยะวรรณ คำน้ำนอง
นายอรุณชัย ศุภฤกษ์จันทราภา

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน 40169
วัน, เดือน, ปี 17 ส.ค. 2544
b 11092646
i.....

เอกสารนี้ใช้ได้เฉพาะสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง การใช้ FPGAs สร้างตัวประมวลผลกลางขนาด 8 บิต

FPGAs Implementation of A Central Processing Unit 8 Bits

วัตถุประสงค์

1. ศึกษาหลักการทำงานของตัวประมวลผลขนาด 8 บิต
2. ศึกษาการใช้งานและการทำงานของโปรแกรม Xilinx Foundation Series
3. ศึกษาการใช้งานและการทำงานของ FPGAs
4. ออกแบบและสร้างตัวประมวลผลขนาด 8 บิตในรูป Schematic
5. สร้างเป็นชุดทดลองและสร้างใบงาน
6. ทดสอบการทำงานของชุดทดลองและใบงานที่สร้างขึ้น
7. ทดสอบการทำงานของตัวประมวลผลขนาด 8 บิตที่ได้ทำการออกแบบ
8. นำตัวประมวลผลขนาด 8 บิตที่ออกแบบได้มาใช้งานได้จริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้เกี่ยวกับหลักการทำงานของตัวประมวลผลขนาด 8 บิต
2. ได้รับความรู้เกี่ยวกับการใช้โปรแกรม Xilinx Foundation Series
3. ได้รับความรู้เกี่ยวกับการใช้งานและการทำงานของ FPGAs
4. ได้ตัวประมวลผลขนาด 8 บิตสามารถที่ใช้งานได้จริง
5. ได้ชุดทดลองและใบงานการทดลอง 10 ใบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การใช้ FPGAs สร้างตัวประมวลผลกลางขนาด 8 บิต	
นักศึกษา	นายจุติ	จุติประเสริฐ
	นางสาวปิยะวรรณ	คำน้ำนอง
	นายอรุณชัย	ศุภฤกษ์จันทราภา
อาจารย์ที่ปรึกษา	อาจารย์สุชิน	อาจหาญ
อาจารย์ที่ปรึกษาร่วม	อาจารย์กิติพงค์	มะโน
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์	
ปีการศึกษา	2543	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอการใช้ FPGAs สร้างตัวประมวลผลกลางขนาด 8 บิต ตัวประมวลผลกลางออกแบบเป็นสองส่วน คือส่วนของข้อมูลและส่วนควบคุม ซึ่งกำหนดให้บัสข้อมูลขนาด 8 บิต บัสตำแหน่ง 16 บิต ALU ขนาด 8 บิต และมีคำสั่ง 32 คำสั่ง การออกแบบตัวประมวลผลกลางทั้งหมดสร้างให้ทำงานได้ด้วยอุปกรณ์ FPGAs และนำตัวประมวลผลกลางไปออกแบบเป็นระบบคอมพิวเตอร์บอร์ดเดี่ยว จากผลการทดสอบการทำงานของตัวประมวลผลกลางและระบบคอมพิวเตอร์บอร์ดเดี่ยวปรากฏว่า สามารถทำงานได้ตรงตามวัตถุประสงค์ที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยดี จากความร่วมมือของสมาชิกในกลุ่มทุกท่าน นอกจากนี้ยังได้รับความกรุณาจากอาจารย์ที่ปรึกษาโครงงานและอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ได้ให้คำปรึกษาแนะนำและความช่วยเหลือในด้านต่างๆ ตลอดจนให้โอกาสในการทำโครงงานนี้อย่างเต็มที่ ทั้งด้านเวลา สถานที่ เครื่องมือ และอุปกรณ์ต่างๆ ขอขอบพระคุณบิดา มารดา ผู้ให้กำเนิดที่ให้โอกาสในการศึกษา และขอขอบคุณเพื่อนๆ ตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่ให้คำแนะนำต่างๆ และเป็นกำลังใจอย่างเต็มเปี่ยมในการทำโครงงานครั้งนี้ จนทำให้โครงงานนี้สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	2
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 กล่าวนำ	4
2.2 วิวัฒนาการของ FPGAs	4
2.3 เอฟพีจีเอ (FPGAs : Field Programmable Gate Array)	4
2.4 โครงสร้างภายในของอุปกรณ์ FPGAs	5
2.5 สถาปัตยกรรมภายในของ FPGAs ตระกูล XC4000	6
2.5.1 ซีแอลบี (CLB : Configurable Logic Block)	7
2.5.2 ไอโอบี (IOB : Input Output Block)	8
2.5.3 อินแตอร์คอนเน็ค (Interconnect)	9
2.6 คุณสมบัติโดยทั่วไปของ FPGAs ตระกูล XC4000	11
2.7 การโปรแกรม FPGAs ตระกูล XC4000	13
2.7.1 การใช้งานในลักษณะสเลฟซีเรียล	14
2.7.2 การใช้งานในลักษณะมาสเตอร์ซีเรียล	16
2.8 การใช้ความสามารถของแรมใน FPGAs ตระกูล XC4000	17
2.9 ทฤษฎีไมโครโปรเซสเซอร์	18
2.10 โครงสร้างภายในของไมโครโปรเซสเซอร์	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.11 หน้าที่เฉพาะรีจิสเตอร์สำหรับการอ้างตำแหน่งต่างๆ	21
2.12 สัญญาณติดต่อกับหน่วยความจำ	22
2.13 การต่อ 8255 ใช้งาน	24
2.13.1 รายละเอียดการเรียงขาของ 8255	26
2.13.2 โหมด 0 : Basic Register I/O	27
2.13.3 ตัวอย่างการใช้งาน 8255 ในโหมด 0	31
2.13.4 ตัวอย่างการใช้งาน 8255 ในโหมด 1	32
2.13.5 ตัวอย่างการใช้งาน 8255 ในโหมด 2	35
บทที่ 3 การทำงาน การสร้างและการทำงาน	37
3.1 ลักษณะการออกแบบ	37
3.2 โครงสร้างของระบบไมโครคอมพิวเตอร์	37
3.2.1 คุณสมบัติหน่วยประมวลผลกลาง	37
3.2.2 โครงสร้างของหน่วยประมวลผล	38
3.2.3 รายละเอียดของคำสั่ง	39
3.2.4 หน้าที่การทำงานของขาสัญญาณแต่ละขา	41
3.3 ส่วนรับข้อมูล	42
3.4 หน่วยความจำ	42
3.5 หน่วยแสดงผล	43
3.6 ส่วนประกอบร่วมของวงจร	44
3.6.1 วงจรรีเซ็ต	44
3.6.2 วงจรกำหนดสัญญาณนาฬิกา	44
3.7 วงจรประยุกต์การใช้งาน	45
3.8 ลำดับการสร้างและทดลอง	47
3.8.1 การออกแบบด้วยโปรแกรม Xilinx Foundation Series	47
3.8.2 การจำลองการทำงาน	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
3.9.3 การโปรแกรมลงบนไอซี FPGAs	48
บทที่ 4 การทดลองและผลการทดลอง	50
4.1 การทดลอง	50
4.2 การออกแบบส่วนต่างๆ	51
4.2.1 การออกแบบ Program Counter	51
4.2.2 ผลการ Simulate คำสั่ง AND และ OR	53
4.2.3 ผลการ Simulate คำสั่ง LOAD	54
4.2.4 ผลการ Simulate คำสั่ง Complement register A และ B	55
4.2.5 ผลการ Simulate คำสั่ง NOR และ XNOR	56
4.2.6 ผลการ Simulate คำสั่ง XOR และ NAND	57
4.3 การเชื่อมต่อบอร์ดทดลองกับสายดาวน์โหลด	57
4.4 ผลการทดสอบบอร์ดทดลอง	58
4.4.1 แอลอีดี	58
4.4.2 หน่วยแสดงผลเจ็ดส่วน	59
4.4.3 ทดสอบการทำงานของอุปกรณ์ FPGAs เบอร์ 4010E และเบอร์4005E	61
บทที่ 5 สรุป	62
5.1 สรุป	62
5.2 ปัญหา	63
5.3 แนวทางแก้ไขปัญหา	63
5.4 แนวทางการพัฒนาต่อ	64
ภาคผนวก ก รูป Schematic ของหน่วยประมวลผลกลางขนาด 8 บิต	65
ภาคผนวก ข วงจร แผ่นวงจรพิมพ์ และรายละเอียดของอุปกรณ์บนบอร์ดทดลอง	70
ภาคผนวก ค ใบงานการทดลอง	77
บรรณานุกรม	133
ประวัติผู้แต่ง	134

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 คุณสมบัติของ FPGAs ตระกูลต่างๆ	5
ตารางที่ 2.2 โหมดต่างๆของการโปรแกรม FPGAs ตระกูล XC4000	13
ตารางที่ 2.3 จำนวนของแรมภายในของ FPGAs ตระกูล XC4000	18
ตารางที่ 3.1 รายละเอียดคำสั่งต่างๆ	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ผังวงจร CLB ภายในของ FPGAs ตระกูล XC4000	8
รูปที่ 2.2 ผังวงจร IOB ภายในของ FPGAs ตระกูช XC4000	9
รูปที่ 2.3 Interconnect ระหว่าง IOB กับ CLB ของ FPGAs ตระกูล XC4000	10
รูปที่ 2.4 ผังวงจร IOBs ภายในของ FPGAs ตระกูล XC4000	11
รูปที่ 2.5 ขั้นตอนในการคอนฟิกเมื่อเริ่มป้อนแหล่งจ่ายไฟเข้าไอซีและการโปรแกรมใหม่	14
รูปที่ 2.6 การต่อใช้งานในลักษณะสเลฟซีเรียล	15
รูปที่ 2.7 แผนภูมิเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในลักษณะสเลฟซีเรียล	15
รูปที่ 2.8 การต่อใช้งานในลักษณะมาสเตอร์ซีเรียล	16
รูปที่ 2.9 การต่อใช้งานในลักษณะมาสเตอร์พาราเรล	17
รูปที่ 2.10 แผนผังระบบไมโครคอมพิวเตอร์	18
รูปที่ 2.11 สถาปัตยกรรมภายในของไมโครโปรเซสเซอร์ 8 บิต	19
รูปที่ 2.12 รีจิสเตอร์ตำแหน่งขนาด 16 บิต ที่ใช้สร้างข้อมูลบนบัสตำแหน่ง	21
รูปที่ 2.13 ลำดับสัญญาณการติดต่อเพื่ออ่านข้อมูลจาก RAM	23
รูปที่ 2.14 ลำดับสัญญาณการติดต่อเพื่อเขียนข้อมูลลง RAM	24
รูปที่ 2.15 แผนผังการทำงานและเวลาการวางตำแหน่งขาของ 8255	25
รูปที่ 2.16 รายละเอียดแต่ละบิตของรีจิสเตอร์ควบคุมของ 8255	27
รูปที่ 2.17 การเซ็ต 8255 ให้อยู่ในโหมด 0	28
รูปที่ 2.18 ตัวอย่างการกำหนดพอร์ต B เป็นพอร์ตอินพุต	28
รูปที่ 2.19 ลักษณะต่างๆในการใช้งานในโหมด 0	29
รูปที่ 2.20 แผนผังการต่อ 8255 เข้ากับคีย์บอร์ดขนาด 16 คีย์	31
รูปที่ 2.21 แผนผังลักษณะการทำงานของการติดต่อระหว่าง 8255 กับอุปกรณ์ภายนอกในลักษณะ Handshake	32
รูปที่ 2.22 แผนผังเวลาของการส่งผ่านข้อมูลจากพอร์ต A ไปยังอุปกรณ์ภายนอก	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่ 2.23 แผนผังเวลาของสัญญาณที่เกิดขึ้น ขณะส่งผ่านจากอุปกรณ์ภายนอกเข้าสู่ 8255	33
รูปที่ 2.24 แผนผังแสดงขบวนการการส่งข้อมูลออกไปยังพอร์ต A และใช้ 4 บิตบนของพอร์ต C เป็นสัญญาณ Handshake	34
รูปที่ 2.25 แผนผังการทำงานพอร์ต A ในโหมด 2	35
รูปที่ 2.26 รายละเอียดของแต่ละขาของพอร์ต C ในการใช้งานในโหมด 2	36
รูปที่ 3.1 โครงสร้างของระบบไมโครคอมพิวเตอร์	37
รูปที่ 3.2 โครงสร้างหน่วยประมวลผลกลาง	38
รูปที่ 3.3 ขาใช้งานของหน่วยประมวลผลกลาง	41
รูปที่ 3.4 วงจรส่วนรับข้อมูล	42
รูปที่ 3.5 การต่อ EPROM และ RAM	43
รูปที่ 3.6 วงจรภาคแสดงผลหน่วยแสดงผลเจ็ดส่วน	43
รูปที่ 3.7 วงจรรีเซ็ต	44
รูปที่ 3.8 วงจรกำเนิดสัญญาณนาฬิกา	44
รูปที่ 3.9 วงจรประยุกต์ใช้งาน	45
รูปที่ 3.10 Schematic ของรีจิสเตอร์ A	47
รูปที่ 3.11 การจำลองการทำงานของรีจิสเตอร์ A	48
รูปที่ 3.12 การเชื่อมต่อกันระหว่างคอมพิวเตอร์กับบอร์ดทดลองของ FPGAs	49
รูปที่ 4.1 บอร์ดทดลอง	50
รูปที่ 4.2 วงจรโครงสร้างภายใน Program Counter	51
รูปที่ 4.3 นำโครงสร้างภายในของ Program Counter มาทำเป็น macro	52
รูปที่ 4.4 ผลการ Simulate Program Counter	52
รูปที่ 4.5 ผลการ Simulate คำสั่ง AND และ OR	53
รูปที่ 4.6 ผลการ Simulate คำสั่ง LOAD	54
รูปที่ 4.7 ผลการ Simulate คำสั่ง Complement register A และ B	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่ 4.8 ผลการ Simulate คำสั่ง NOR และ XNOR	56
รูปที่ 4.9 ผลการ Simulate คำสั่ง XOR และ NAND	57
รูปที่ 4.10 การต่อสายดาวน์โหลดกับบอร์ดทดลอง	58
รูปที่ 4.11 ผลการดาวน์โหลดโปรแกรมเพื่อทดสอบแอลอีดี	59
รูปที่ 4.12 ผลการดาวน์โหลดโปรแกรมเพื่อทดสอบหน่วยแสดงผลเจ็ดส่วน	60
รูปที่ ก.1 ส่วนของตัวประมวลผลกลางขนาด 8 บิต	66
รูปที่ ก.2 ส่วนของการควบคุมตัวประมวลผลกลางขนาด 8 บิต	67
รูปที่ ก.3 ส่วน Data Part 1 ของตัวประมวลผลกลางขนาด 8 บิต	68
รูปที่ ก.4 ส่วน Data Part 2 ของตัวประมวลผลกลางขนาด 8 บิต	69
รูปที่ ข.1 วงจรในส่วนของหน่วยประมวลผลกลางและหน่วยความจำ	71
รูปที่ ข.2 วงจรในส่วนของอุปกรณ์อินพุตและเอาต์พุต	72
รูปที่ ข.3 แผ่นวงจรพิมพ์ด้านหน้า	73
รูปที่ ข.4 แผ่นวงจรพิมพ์ด้านหลัง	74
รูปที่ ข.5 ลักษณะการวางอุปกรณ์บนแผ่นวงจรพิมพ์	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

อุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ เป็นอุปกรณ์ที่ได้รับการพัฒนามาเมื่อประมาณ 20 ปีก่อน ขณะนั้นการพัฒนาอุปกรณ์อิเล็กทรอนิกส์เป็นไปอย่างล่าช้า เพราะเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์ ได้รับการพัฒนาไปอย่างรวดเร็วดังที่จะเห็นได้ว่ามีวงจรดิจิตอลอิเล็กทรอนิกส์หลายวงจร ที่แต่เดิมถูกสร้างขึ้นมาจากชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์จำนวนหลายชิ้น ถูกนำมาประกอบกันอยู่บนแผงวงจรไฟฟ้า (Printed Circuit Board หรือ PCB) ที่มีขนาดใหญ่ แต่ในปัจจุบันสามารถที่จะใช้เทคโนโลยีการออกแบบและผลิตวงจรรวมขนาดใหญ่มาก (Very Large Scale Integration : VLSI) รวมอุปกรณ์ต่างๆ เหล่านั้นให้อยู่บนชิ้นอุปกรณ์สารกึ่งตัวนำ ที่มีขนาดประมาณ 1 – 2 ตร.ซม.ได้ ซึ่งเป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้น ตลอดจนความน่าเชื่อถือ และความคงทนต่อสภาพแวดล้อมสูง การที่จะเปลี่ยนอุปกรณ์ใหม่เป็นสิ่งที่ต้องใช้งบประมาณมาก และจะประสบปัญหาเช่นเดิม คือ อุปกรณ์ใหม่ได้รับการพัฒนามานานแล้วเช่นกันเพราะในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกร และเวลาสำหรับดำเนินการมาก ฉะนั้นวิธีการที่จะช่วยพัฒนาวงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น และโครงการดังกล่าวมีชื่อว่า Very High Speed Integrated Circuits หรือ VHSIC ใช้สำหรับบรรยายพฤติกรรมของวงจรหรือ Hardware ของระบบ เป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร (Simulation Language) ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง (High Level Language) เรียกว่า Hardware Description Language หรือ HDL ดังนั้นภาษามาตรฐานนี้จึงมีชื่อว่า VHSIC-HDL หรือ VHDL ปัจจุบันในการออกแบบวงจรดิจิตอล มักใช้อุปกรณ์และซอฟแวร์ ช่วยในการออกแบบ เพื่อให้การออกแบบนั้นมีประสิทธิภาพสะดวก รวดเร็ว วงจรที่ได้มีขนาดเล็ก ซอฟแวร์ที่ใช้ในปัจจุบันได้แก่ VHDL FPGAs และโปรแกรมอื่นๆ

ดังนั้นในโครงงานนี้ จึงเสนอการออกแบบตัวประมวลผลกลางขนาด 8 บิต โดยใช้วิธีการออกแบบส่วนต่างๆของตัวประมวลผลขนาด 8 บิตให้อยู่ในรูปของ Schematic แล้วโปรแกรมลงบนอุปกรณ์ที่สามารถโปรแกรมได้ ซึ่งได้เลือกใช้อุปกรณ์ FPGAs ซึ่งเป็นวงจรรวมที่สามารถโปรแกรมเป็นวงจรเชิงเลขโดยโปรแกรมลงบนสเตติกแรมภายในด้วยข้อมูลที่อยู่ภายนอกและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบและการสร้างหน่วยประมวลผลกลางขนาด 8 บิต กล่าวถึงโครงสร้างและการออกแบบ, วิธีการออกแบบ, ลำดับขั้นตอนการออกแบบหน่วยประมวลผลกลางขนาด 8 บิต และการออกแบบวงจรอินเตอร์เฟส

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทดลอง Simulate ในขั้นตอนที่เป็นบล็อก เป็นอุปกรณ์ FPGAs และหน่วยประมวลผลกลางขนาด 8 บิต

บทที่ 5 สรุปอภิปรายและข้อเสนอแนะ กล่าวถึงข้อสรุป, ปัญหา, แนวทางการแก้ไขและแนวทางการพัฒนาเพื่อประยุกต์ใช้งาน

ภาคผนวก ก รูป Schematic ของหน่วยประมวลผลกลางขนาด 8 บิต

ภาคผนวก ข วงจร แผ่นวงจรพิมพ์ และรายละเอียดของอุปกรณ์บนบอร์ดทดลอง

ภาคผนวก ค ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

ปัจจุบันการออกแบบระบบเชิงเลขนั้นมีความยุ่งยากซับซ้อนมากยิ่งขึ้น ทำให้คอมพิวเตอร์เข้ามามีบทบาทที่สำคัญและนำมาเพื่อช่วยในการออกแบบวงจรเชิงเลขเราเรียกว่า CAD (Computer Aided Design)วิธีการออกแบบใหม่ๆ ได้ถูกพัฒนาขึ้นจากซอฟต์แวร์ใหม่ๆ เพื่ออำนวยความสะดวกให้กับนักออกแบบ ดังนั้นโปรแกรม Xilinx Foundation Series จึงเป็นเครื่องมืออย่างหนึ่งที่มีการพัฒนาขึ้นมา และช่วยในกระบวนการออกแบบวงจรเชิงเลขให้สะดวกมากยิ่งขึ้น เมื่อได้ทำการออกแบบโดยใช้โปรแกรม Xilinx Foundation Series ให้อยู่ในรูป Schematic แล้ว จึงนำมาโปรแกรมลงบนอุปกรณ์ FPGAs ก็สามารถที่จะนำมาใช้งานได้

2.2 วิวัฒนาการของ FPGAs

ในปี ค.ศ. 1985 บริษัท Xilinx แนะนำการสร้างขึ้นอันดับแรกของ Field Programmable Gate Array (FPGAs) ประกอบด้วยอุปกรณ์ 2 ตัว และสนับสนุนการใช้มือออกแบบแก้ไขสำหรับการวางและจัดเส้นทาง 20 ปีผ่านไปก็ยังไม่มีการปฏิวัติของอุปกรณ์เหล่านั้น แต่อย่างไรก็ตามระยะ 10 ปีหลังได้ทีการเพิ่มความเร็วของสัญญาณนาฬิกาจาก 10 MHz ไปจนถึง 100 MHz กฎการออกแบบมีความสมบูรณ์มากขึ้น, เพิ่มความหนาแน่นเกตจาก 100 ไปจนถึง 20,000 เกต และเป็นการง่ายที่จะใช้เครื่องมือด้านซอฟต์แวร์ที่มีการพัฒนาเพื่อจะสนับสนุนอุปกรณ์เหล่านั้น สิ่งเหล่านี้เป็นการเพิ่มโครงสร้าง, เครื่องมือและเทคโนโลยีขั้นก้าวหน้าอื่นๆ อีกจำนวนมาก มีการช่วยเหลือที่จะรองรับความเร็วของอุปกรณ์เหล่านั้น

2.3 เอฟพีจีเอ (FPGAs : Field Programmable Gate Array)

เป็นอุปกรณ์ที่ถูกพัฒนาต่อจากอุปกรณ์แอลซีเอของบริษัทไซลิงซ์ (XILINX Inc.) โดยมีประสิทธิภาพการทำงานและมีปริมาณความหนาแน่นของเกตสูง สามารถจะกำหนดฟังก์ชันการทำงานได้ตามความต้องการของผู้ใช้โดยผ่านการโปรแกรม FPGAs ได้รวบรวมข้อดีทั้งหมดของที่ทำ Custom VLSI มารวมไว้ทั้งหมดได้แก่ การออกแบบการผลิต, ลดเวลาที่จะส่งตัวผลิตภัณฑ์ออกตลาด ซึ่งเป็นประโยชน์ต่อการผลิตวงจรเป็นอย่างมาก นักออกแบบเพียงกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันการทำงานของวงจร ดังนั้นการออกแบบวงจรโดยใช้ FPGAs สามารถออกแบบและทดสอบภายในเวลาเพียง 2 – 3 วัน เท่านั้น ตรงกันข้ามกับการออกแบบโดยใช้เกตอาเรย์ ซึ่งใช้เวลาหลายสัปดาห์ การแก้ไขแบบก็เช่นเดียวกัน จากประโยชน์ของ FPGAs ดังกล่าวมาทำให้เกิดการประหยัดค่าใช้จ่ายเป็นอย่างมาก เพราะได้ความเสี่ยงในการที่จะต้องแก้ไขตัววงจร การเลื่อนเวลา การออกแบบผลิตภัณฑ์ ลดค่าเอ็นอาร์อี (NRE : Nonrecurring Engineering Cost)

2.4 โครงสร้างภายในของอุปกรณ์ FPGAs

FPGAs เป็นวงจรรวมที่สามารถโปรแกรมเป็นวงจรเชิงเลขโดยโปรแกรมลงบนสแตติกแรมภายในด้วยข้อมูลที่อยู่ภายนอก และสามารถโปรแกรมใหม่ได้ด้วยการรีเซ็ตด้วยสัญญาณไฟฟ้า FPGAs ยังประหยัดไฟและมีความจุวงจรสูง (เกตมาก)

วงจรรวมผลิตโดยบริษัทไซลิงค์ (Xillinx) ซึ่งค้นคว้าร่วมกับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นกลุ่มของเกตจำนวน 600 – 25,000 เกต ดังตารางที่ 2.1

ตารางที่ 2.1 คุณสมบัติของ FPGAs ตระกูลต่างๆ

FPGAs	Appr.gate Count	Max I/Os	Flip-Flop	RAM bits	Available CLBs
XC2064	1,000	58	122	0	64
XC2018	1,500	74	174	0	100
XC3020/3120	1,800	64	256	0	64
XC3030/3130	2,700	80	360	0	100
XC3042/3142	3,700	96	480	0	144
XC3064/3164	5,500	120	688	0	244
XC3090/3190	7,500	144	928	0	320
XC3195	9,000	176	1,320	0	484
XC4002A	2,000	64	256	2,048	64
XC4003/4003A	3,000	80	360	3,200	100
XC4003H	3,000	160	200	3,200	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1(ต่อ) คุณสมบัติของ FPGAs ตระกูลต่างๆ

FPGAs	Appr.gate Count	Max I/Os	Flip-Flop	RAM bits	Available CLBs
XC4005/4005A	5,000	122	616	6,072	196
XC4005H	5,000	192	392	6,272	196
XC4006	6,000	128	768	8,192	256
XC4008	8,000	144	936	10,368	324
XC4010	10,000	160	1,120	12,800	400
XC4013	13,000	192	1,536	18,432	576
XC4025	25,000	256	2,560	32,768	1,024

FPGAs มีโครงสร้างภายในใกล้เคียงกับสถาปัตยกรรมของเกตอาเรย์ (Gate Array Logic, GAL) มาก สามารถโปรแกรมและลบคอนฟิกกูเรชั่น (Configuration) ภายในสแตติกแรม (Static RAM) ได้โดยการใช้กระแสไฟฟ้า ซึ่งทำการโปรแกรมได้โดยดึงข้อมูลฐานสิบหกมาจากภายนอก เช่น Parallel EPROM หรือ Serial PROM ต่างกับ EPLD,PAL ที่มี EPROM อยู่ในตัว ภายใน FPGAs จะจัดเรียงเป็นลอจิกเซลล์ล้อมรอบภายนอกด้วยอินพุต, เอาต์พุตเซลล์ FPGAs ตัวแรกที่ผลิตโดยบริษัทไซลิงค์คือเบอร์ XC2064 (2000 Family) ประกอบด้วยเซลล์เรียงกันเป็นเมตริกซ์ (Matric) จำนวน 64 เซลล์ หลังจากนั้นผลิต FPGAs ตระกูล 3000 และ 4000 ซึ่งมีโครงสร้างซับซ้อนขึ้นสามารถเพิ่มจำนวนเกตได้มากขึ้นและดีขึ้นแต่ละเซลล์เรียกว่า CLB (Configurable Logic Block)

2.5 สถาปัตยกรรมภายในของ FPGAs ตระกูล XC4000

FPGAs (Field Programmable Gate Array) เป็นชิพที่มีสนามวงจรขนาดใหญ่อยู่ภายในที่เราสามารถนำมาใช้และออกแบบวงจรต่างๆได้ตามที่เราต้องการ โดยมีโปรแกรมสำเร็จรูปที่ใช้ในการออกแบบอยู่แล้ว มีความเร็วสูง มีสถาปัตยกรรมการออกแบบคล้าย CPLD (Complex Programmable Logic device) แต่มีส่วนประกอบที่ซับซ้อน และมีประสิทธิภาพมากกว่า ในการออกแบบวงจรสามารถทำได้ง่าย ทั้งในการเชื่อมต่อและการออกแบบวงจรเป็นอย่างมาก สถาปัตยกรรมภายในของ FPGAs แบ่งออกเป็น 3 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

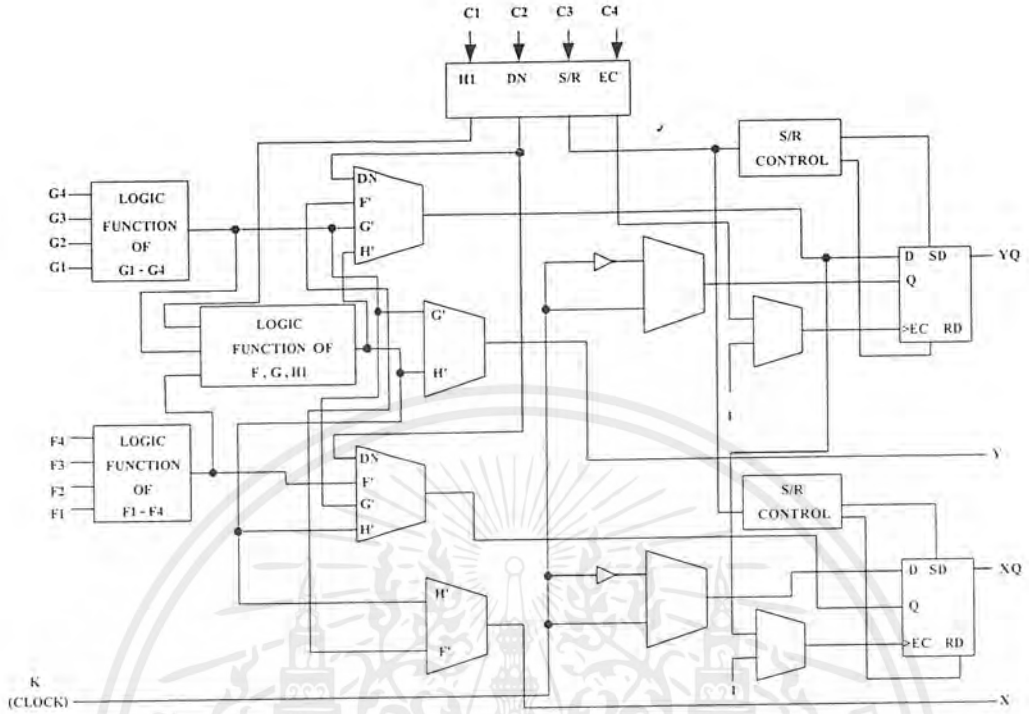
- CLB (Configuration Logic Block)
- IOB (Input Output Block)
- Interconnect

ซึ่งภายในมีสถาปัตยกรรมของ FPGAs ซึ่งมีลักษณะเป็นตารางของลอจิกบล็อก (Logic Block) และล้อมรอบไปด้วยบล็อกการเชื่อมต่อของ ไอโอ (I/O Interface Block) การเชื่อมต่อระหว่างซีแอลบี (CLB : Configuration Logic Block) และไอโอบี (IOB : Input Output Block) ซึ่งทำได้โดยผ่านช่องที่ว่างพาดผ่านระหว่างแถวและคอลัมน์ มีการทำงานเหมือนกับไมโครโปรเซสเซอร์ ตัวแอลซีเอจะทำงานได้จะต้องใช้ Program-driven Logic Device หน้าที่ของซีแอลบีและไอโอบีแต่ละตัว การเชื่อมต่อภายใน (Interconnection) ถูกกำหนดไว้ในโปรแกรมคอนฟิกคูเรชัน (Configuration Program) หรือเก็บไว้ในอีพรอม ภายในแอลซีเอ (LCA : Logic Cell Array)โปรแกรมจะถูกโหลดเข้าสู่แอลซีเอเมื่อมีการจ่ายไฟ (Power – Up) โดยทางคำสั่ง (Command) ซึ่งเป็นส่วนหนึ่งของการเริ่มต้นระบบ (System Initialization) ประสิทธิภาพของแอลซีเอกำหนดโดย ความเร็วของลอจิกส่วนประกอบหน่วยความจำและการโปรแกรมการเชื่อมต่อต่างๆ ความเร็วของอัตราของระบบสัญญาณนาฬิกา (System Clock Rate) ถูกกำหนดด้วย ทอกเกิลฟลิปฟลอป สำหรับการประยุกต์ใช้โดยทั่วไปจะอยู่ที่ประมาณ 1 / 3 ถึง 1 / 2 ค่าสูงสุดของทอกเกิลเกต

2.5.1 ซีแอลบี (CLB : Configurable Logic Block)

ภายใน LCA (Logic Cell Array) คือเมทริกซ์ของแอลซีบีแต่ละตัวประกอบด้วยหน่วยของคอมบิเนชั่นลอจิกที่สามารถโปรแกรมได้ (Programmable Combination Logic) และส่วนของรีจิสเตอร์เก็บข้อมูล (Storage Register) ส่วนของวงจรคอมบิเนชั่นลอจิกสามารถใช้สร้างวงจรทางด้านฟังก์ชันบูลีนของอินพุต ส่วนรีจิสเตอร์รับค่าจากส่วนคอมบิเนชั่นหรือโดยตรงจากเอาต์พุตของซีแอลบี สามารถขับวงจรคอมบิเนชั่นลอจิกโดยตรงผ่านเส้นทางเดินย้อยกลับ (Feedblack Path) CLB จะจัดเรียงกันเป็นแบบเมตริกซ์แบบอาเรย์ขนาด M x N การออกแบบนั้นสามารถทำได้โดยการจัดวาง CLB ให้ต่อถึงกัน เราสามารถจัด CLB ให้เชื่อมต่อถึงกันได้โดยการทำด้วยมือหรือโปรแกรมที่สนับสนุน FPGAs ทำให้โดยอัตโนมัติ โดยวิธีของมันเอง สำหรับไฟล์ที่ได้จากโปรแกรมเหล่านี้ เราเรียกว่า ไฟล์ที่กำหนดการวางอุปกรณ์ (Configuration File) ซึ่งจะบรรจุโครงร่างภายในของ CLB ตามความเหมาะสม อีกด้านหนึ่งไฟล์กำหนดการวางอุปกรณ์นั้นจะเป็นไฟล์กระแสข้อมูล (Bit Stream) ซึ่งสามารถใช้โปรแกรมหน่วยความจำภายในของ FPGAs สำหรับรูปที่ 2.1 แสดง CLB ของ FPGAs ตระกูล 4000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

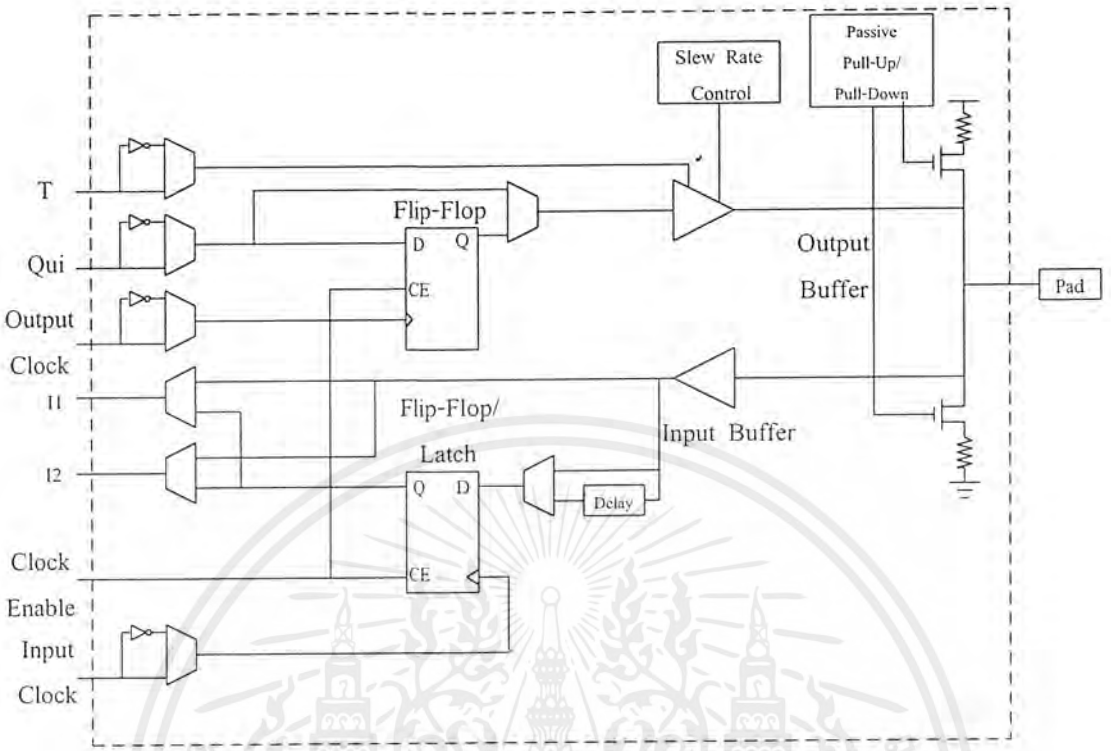


รูปที่ 2.1 ผังวงจร CLB ภายในของ FPGAs ตระกูล XC4000

2.5.2 ไอโอบี (IOB : Input Output Block)

เป็นส่วนติดต่อกับวงจรภายนอกของแอลซีเอสร้างมาจากส่วนของอุปกรณ์อินพุต / เอาต์พุตที่สามารถโปรแกรมได้ (Programmable Input/Output) แต่ละตัวสามารถโปรแกรมได้อย่างอิสระ โดยจะให้เป็นอินพุต/เอาต์พุตแบบ 3 สถานะ หรือ ไอโอแบบสองทิศทางก็ได้ โดยอินพุตสามารถโปรแกรมให้รู้จักทั้งระดับสัญญาณทีทีแอลและซีมอสเทรดโฮล ของไอโอบี แต่ละตัวมีฟลิปฟลอปสามารถใช้เป็นบัฟเฟอร์สำหรับอินพุตและเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

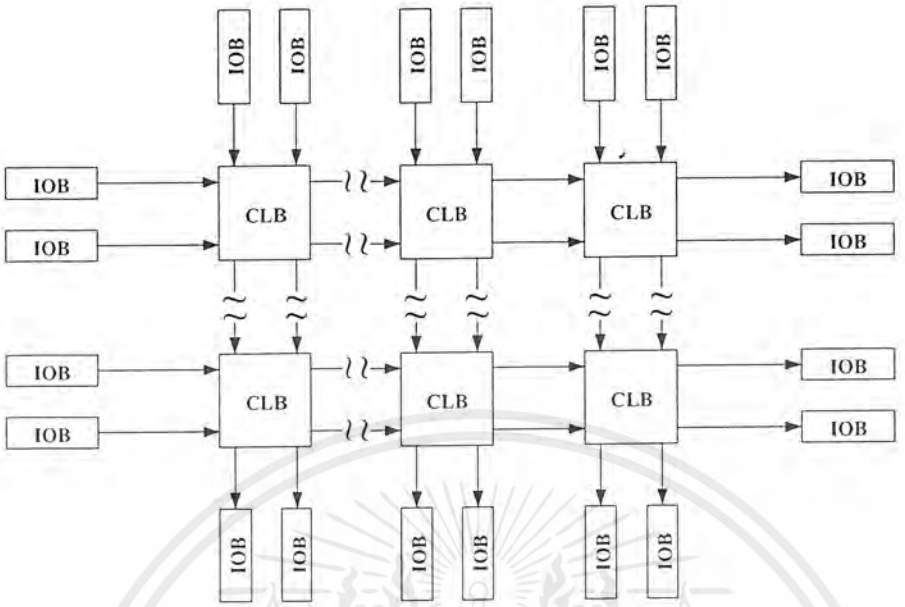


รูปที่ 2.2 ผังวงจร IOB ภายในของ FPGA ตระกูล XC4000

2.5.3 อินเตอร์คอนเน็ค

ความยืดหยุ่นของการใช้แอลซีเอมาทำเป็นอุปกรณ์ขึ้นอยู่กับการโปรแกรม ทรัพยากรต่างๆ ที่อยู่ภายในเข้าด้วยกัน การที่จะควบคุมการเชื่อมต่อระหว่างจุดสองจุดภายในชิพเหมือนกับเกตอาเรย์ทั่วๆ ไป การเชื่อมต่อภายในแอลซีเอประกอบด้วยเน็ตเวิร์ค 2 ทิศทาง คือทางแถวและคอลัมน์ ซึ่งวางอยู่ระหว่าง CLB Programmable Switch จะทำการเชื่อมต่ออินพุตและเอาต์พุตของไอโอบี และซีแอลบี ที่จุดต่อร่วมระหว่างแถวกับคอมลัมน์สามารถสลับสัญญาณจาก เส้นทางไปยังส่วนต่างๆ

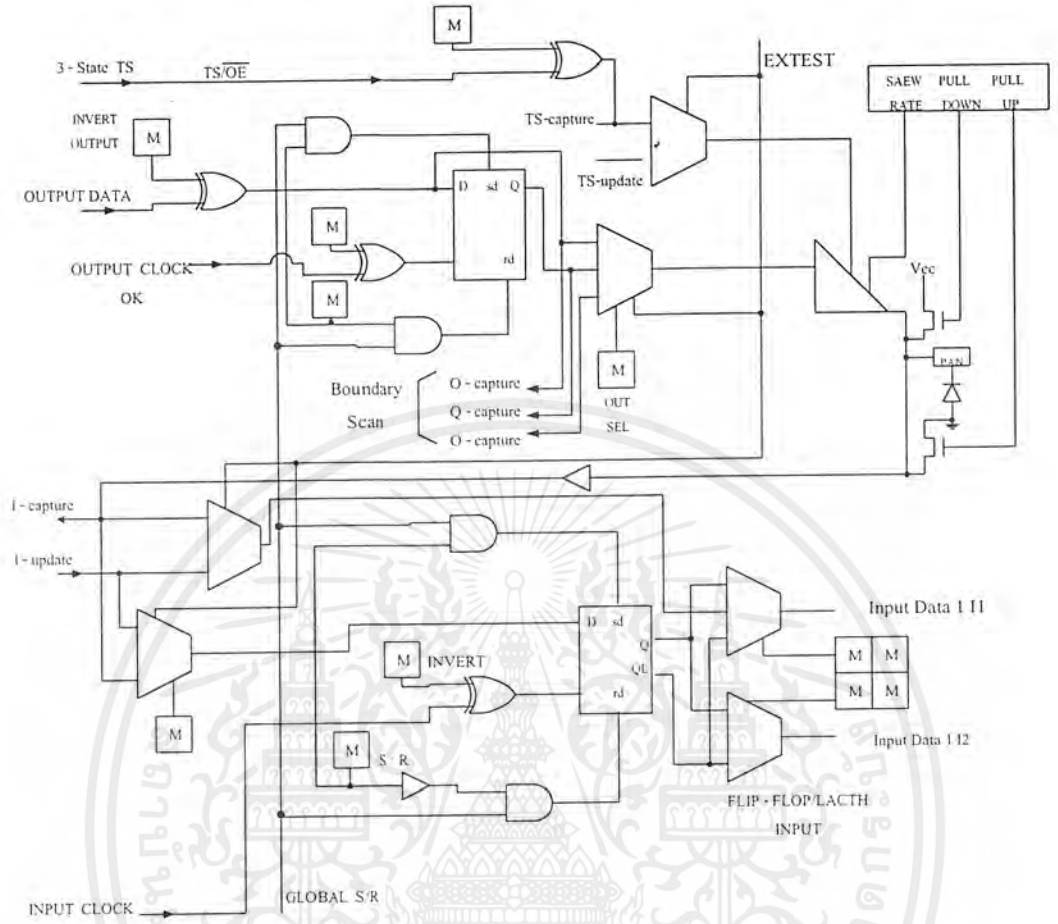
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 อินเตอร์คอนเน็คระหว่าง IOB กับ CLB ของ FPGAs ตระกูล XC4000

รอบนอกของ FPGAs จะประกอบด้วย IOBs ประมาณ 64 ถึง 144 ตัว ซึ่งขึ้นอยู่กับตระกูลของ FPGAs ซึ่ง IOBs จะเป็นตัวเชื่อมต่อระหว่างภายในกับภายนอกของวงจรลอจิกของ FPGAs ลักษณะของ IOBs จะมีลักษณะ 2 ทิศทาง สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้ สำหรับรูปที่ 2.4 แสดง IOBs ของ FPGAs ตระกูล 4000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ผังวงจร IOBs ภายในของ FPGAs ตระกูล XC4000

2.6 คุณสมบัติโดยทั่วไปของ FPGAs ตระกูล XC4000

1.) เป็นอุปกรณ์รุ่นที่ 3 ของ FPGAs

- มีฟลิปฟลอปเป็นจำนวนมาก
- ในการผลิตฟังก์ชั่นของการทำงานมีความยืดหยุ่นสูง
- มีจำนวนเกตภายในจำนวน 2,000 - 10,000 เกต
- เพิ่มความสามารถพิเศษของรีจิสเตอร์และอินพุต / เอาต์พุต
- มีค่าแฟนเอาต์ (Fan-Out) สูง
- มีบัสภายใน 3 สถานะ
- ทำงานกับสัญญาณทีทีแอลและซีมอส
- มีออสซิลเลเตอร์แอมปลิฟลายเออร์ภายใน
- มีแรมภายในความเร็วสูง (< 25Ns)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้กับงานที่ต้องการความเร็วสูง (ใช้งานได้ที่ความถี่ 70/100/125 MHz)
- มี Wide Edge Decoder
- เส้นทางการเชื่อมต่อ (Interconnect Line) เป็นแบบลำดับชั้น
- มีการกระจายกำลังงานของสัญญาณต่ำ

2.) มีสถาปัตยกรรมภายในที่ยืดหยุ่น
- มีลอจิกบล็อกและไอโอบล็อกที่สามารถโปรแกรมได้
- มีอินเตอร์คอนเน็คและ Wide Decoder ที่โปรแกรมได้

3.) ทำกระบวนการซับไมครอนชนิดซีมอสได้
- มีลอจิกและอินเตอร์คอนเน็คที่มีความเร็วสูง
- ใช้กำลังงานต่ำ

4.) คุณลักษณะทาง System – Oriented
- รองรับมาตราฐาน IEEE 1149.1 ในการทำ Boundary-Scan Logic
- สามารถโปรแกรมค่า output slew rate ได้
-สามารถโปรแกรมให้อินพุตมีลักษณะพูลอัพ(Pull – Up)หรือพูลดาวน์ (Pull – Down) รีจิสเตอร์ได้
- ให้กระแสเอาต์พุตได้ตั้งแต่ 12 – 24 มิลลิแอมป์ (ขึ้นอยู่กับแต่ละรุ่น)

5.) ทำการโหลดเอาแฟ้มข้อมูลประเภทไบนารี
- ไม่จำกัดจำนวนครั้งในการโปรแกรมซ้ำ
- มีโหมดในการโปรแกรมให้เลือก 6 โหมด

6.) มีโปรแกรมช่วยพัฒนาได้แก่ XCAT Development System (ปัจจุบัน Foundation Series) ที่ทำงานบนคอมพิวเตอร์รุ่นต่างๆ เช่น 486/Pentiums, NEC PC, Apollo, Sun-4 , HP700
- สามารถติดต่อกับโปรแกรมอื่นๆได้ เช่น Viewlogic , Mentor Graphic และ OrCAD เป็นต้น
- มีโปรแกรมการวางและเชื่อมโยงอุปกรณ์ภายในแบบอัตโนมัติ (Automatic Place and Routing) ที่ครบสมบูรณ์
- มี Interactive Design Editor ที่ใช้สำหรับการทำ Optimization
- มี 288 มาโคร 34 ฮาร์ดมาโคร และ แรม / รอม คอมพายเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การโปรแกรม FPGAs ตระกูล XC4000

คือ กระบวนการในการโหลดข้อมูลในโปรแกรมไปยังแอลซีเอ เพื่อกำหนดหน้าที่ของการทำงานในแต่ละบล็อกภายใน และการเชื่อมต่อ ซึ่ง FPGAs ตระกูล XC4000 จะต้องใช้ข้อมูลเกี่ยวกับการโปรแกรมประมาณ 350 บิตต่อซีแอลบี โดยแต่ละบิตจะบอกถึงสถานะของหน่วยความจำสเตติกที่ควบคุมบิตในการควบคุมตารางฟังก์ชั่น (Function Table Bit) และมัลติเพลกซ์อินพุตหรือการเชื่อมต่อกันระหว่างทรานซิสเตอร์

2.7.1 โหมดการโปรแกรม

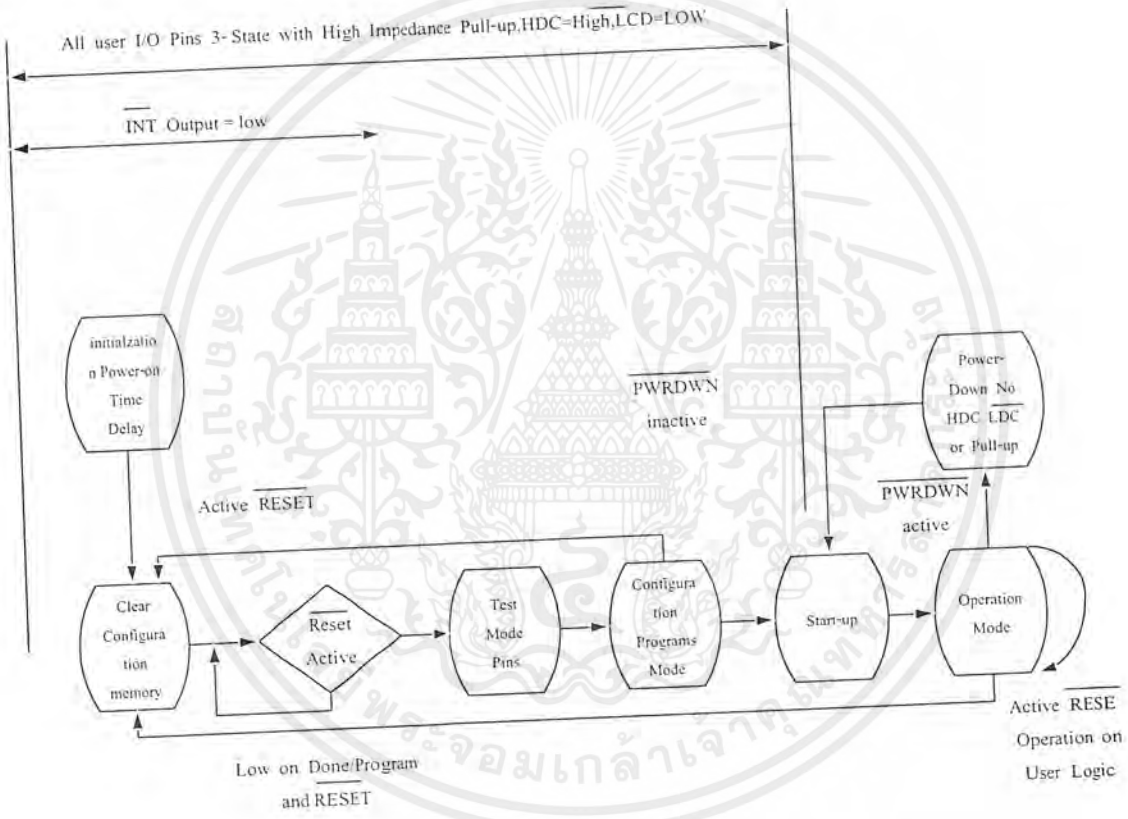
FPGAs สามารถทำงานได้หลายลักษณะโดยกำหนดได้ที่ขาสัญญาณ M0 M1 M2 ดังแสดงในตารางที่ 2.2 ในลักษณะมาสเตอร์พาราเรล (Master Parallel Mode) รันโปรแกรมคอนฟิกทีละไบต์ (Byte) จากหน่วยความจำภายนอกที่เป็นแบบขนาน โดยสามารถรันโปรแกรมคอนฟิก (Config) จากแอดเดรส (Address) ต่ำหรือสูงก่อนก็ได้ การต่อลักษณะเพอริเฟอรัล (Peripheral) จะรับโปรแกรมคอนฟิกทีละ 1 ไบต์ จากไมโครโปรเซสเซอร์ โดยสามารถโต้ตอบกันได้ว่าพร้อมหรือไม่ที่จะรับข้อมูลต่อไป การต่อลักษณะสเลฟซีเรียล (Slave Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิตจากไมโครโปรเซสเซอร์ตามสัญญาณอินพุต CCLK ส่วนการต่อลักษณะมาสเตอร์ซีเรียล (Master Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิตจากหน่วยความจำภายนอกที่เป็นแบบอนุกรม

ตารางที่ 2.2 โหมดต่างๆของการโปรแกรม FPGAs ตระกูล XC4000

Mode	M2	M1	M0	CCLK	DATA
Master Serial	0	0	0	Output	Bit – Serial
Slave Serial	1	1	1	Input	Bit – Serial
Master Parallel up	1	0	0	Output	Byte – Wide, 00000 up
Master Parallel down	1	1	0	Output	Byte – Wide, 3FFFF
Peripheral Synchr.	0	1	1	Input	Byte – wide
Peripheral Asynchr.	1	0	1	Output	Byte – wide
Reserved	0	1	0	-	-
Reserved	0	0	1	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน FPGAs ในการต่อลักษณะสเลฟซีเรียลและมาสเตอร์ซีเรียล เมื่อเริ่มจ่ายไฟเข้าตัว FPGAs จะทำการลบข้อมูลหน่วยความจำที่ใช้ในคอนฟิก (Configuration Memory) ตรวจสอบลักษณะการคอนฟิกว่าเป็นลักษณะใดในตารางที่ 2.2 ว่าเป็นแบบอนุกรมหรือขนาน หลังจากนั้นจะเริ่มทำการโปรแกรมคอนฟิกสัญญาณ Done/Program เป็น "0" ซึ่งอยู่ระหว่างโปรแกรมและเมื่อข้อมูลตรงกับส่วนหัวของข้อมูลคอนฟิกสัญญาณ Done/Program จะเป็น "1" ซึ่งหมายถึง โปรแกรมทำการคอนฟิกเสร็จสิ้น ดูรูปที่ 2.5 ประกอบ



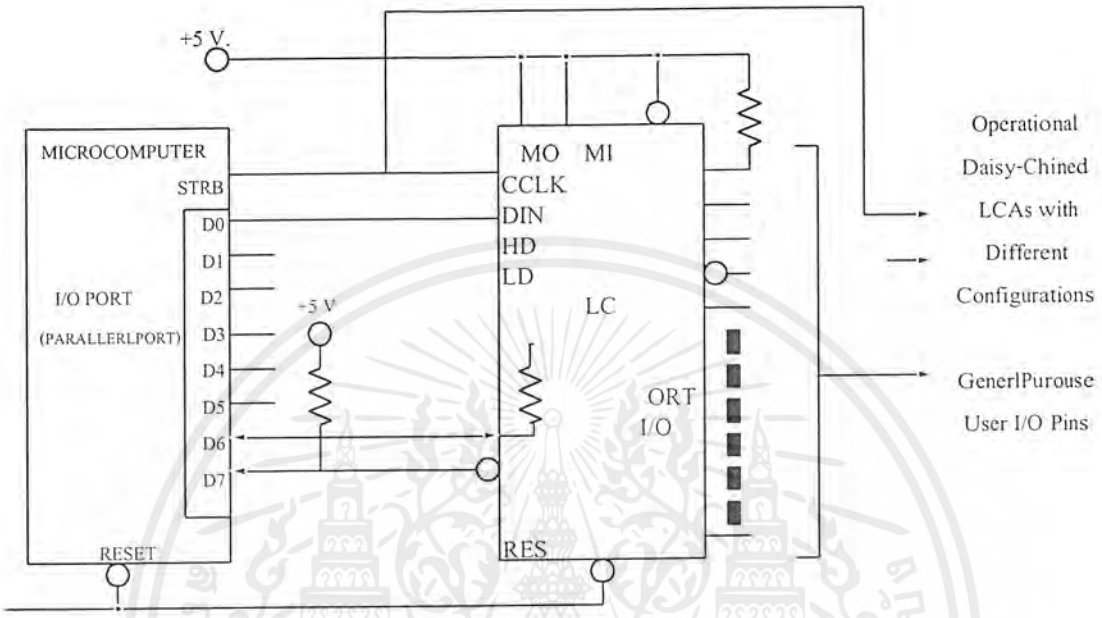
รูปที่ 2.5 ขั้นตอนในการคอนฟิกเมื่อเริ่มป้อนแหล่งจ่ายไฟเข้าไอซีและการโปรแกรมใหม่

2.7.1.1 การใช้งานในลักษณะสเลฟซีเรียล

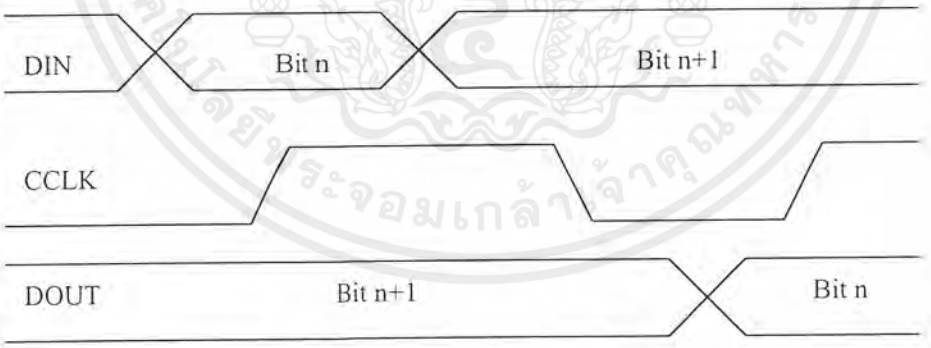
การต่อในลักษณะนี้เหมาะสมกับวงจรที่ออกแบบมาเพื่อทำงานร่วมกับไมโครคอมพิวเตอร์ ทั้งนี้เพราะ FPGAs ได้ใช้ความสามารถของไมโครคอมพิวเตอร์ในการเก็บและส่งข้อมูลคอนฟิกให้ เพียงแต่ต้องเขียนโปรแกรมเพื่อส่งโปรแกรมคอนฟิกให้เพิ่มลักษณะการต่อในลักษณะนี้เป็น ดังรูป 2.6 ซึ่งไมโครคอมพิวเตอร์จะสร้างสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGAs การป้อนโปรแกรมคอนฟิกให้ FPGAs ทำได้โดยการต่อสัญญาณ Strobe เข้ากับขา CCLK และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต D0 เข้ากับขา DIN สร้างสัญญาณคล็อกป้อนที่ขา CCLK และป้อนโปรแกรมคอนฟิกแบบอนุกรมเข้าที่ขา DIN ดังรูปที่ 2.6



รูปที่ 2.6 การต่อใช้งานในลักษณะสเลฟซีเรียล

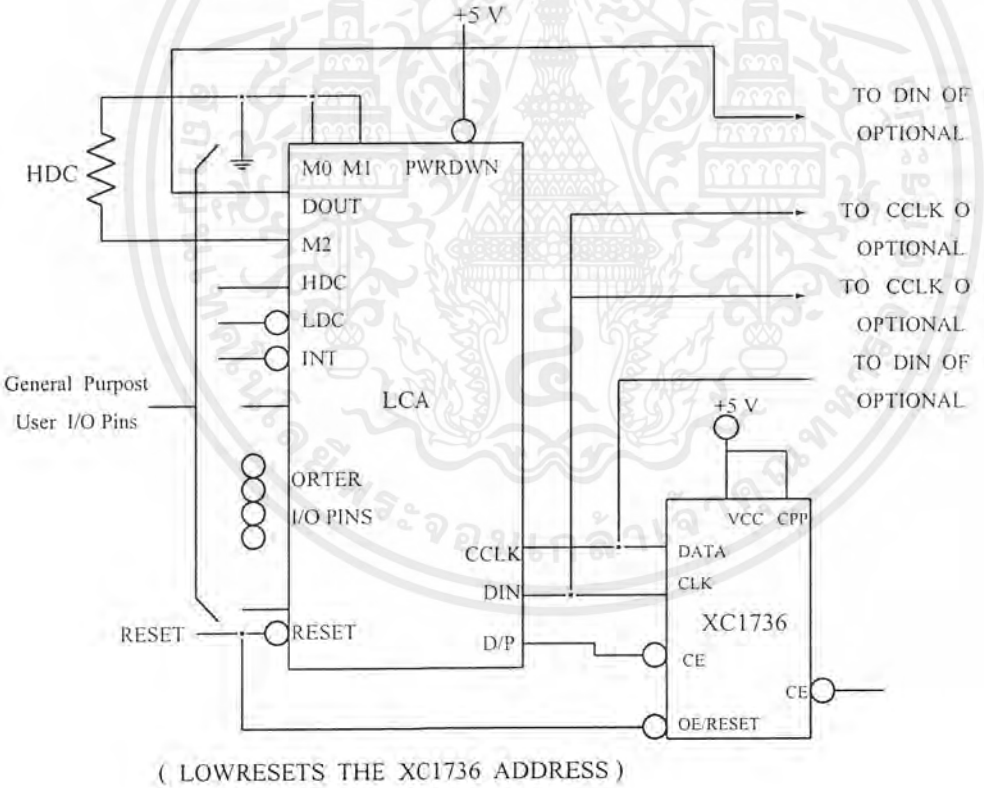


รูปที่ 2.7 แผนภูมิเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในลักษณะสเลฟซีเรียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1.2 การใช้งานลักษณะมาสเตอร์ซีเรียล

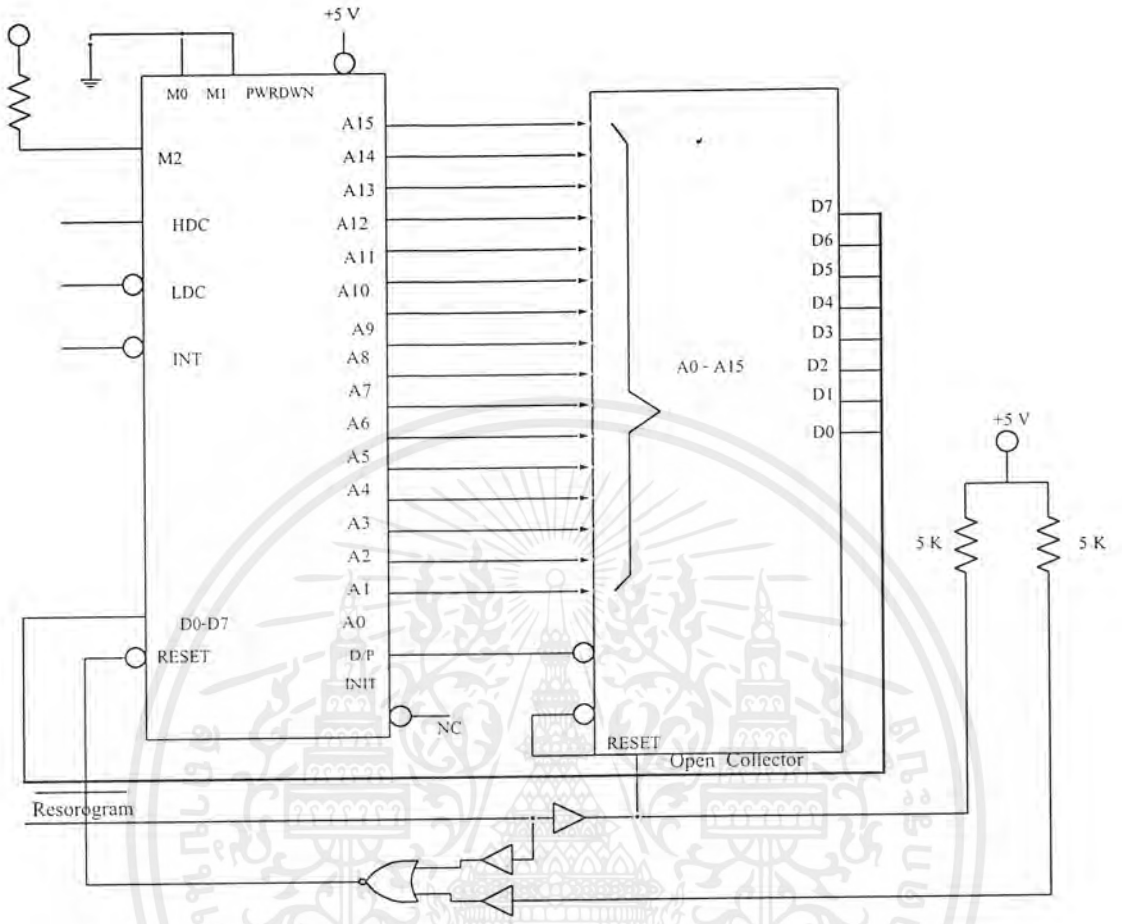
การต่อใช้งานในลักษณะนี้ส่วนที่เก็บโปรแกรมคอนฟิกจะต่างจากการต่อลักษณะสเลฟซีเรียล คือใช้ PROM เบอร์ XC17XXX เป็นตัวเก็บโปรแกรม ทำให้ไม่ต้องเสียเวลาเขียนโปรแกรมเพื่อทำการคอนฟิก ซึ่งวิธีการอัดโปรแกรมคอนฟิกลง PROM ทำตามขั้นตอนดังนี้ คือ เมคบิต (MakeBits) สร้างไฟล์ .BIT ของวงจรที่ออกแบบ และใช้โปรแกรม MakePROM สร้าง Hex ไฟล์แล้วทำการอัดโปรแกรมลง PROM ด้วยอุปกรณ์อัด PROM ที่มาพร้อมกับตัวโปรแกรมของไซลิงค์ PROM XC17XXX จะส่งสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGAs ดังแสดงในรูป 2.9 D0-D7 เป็นขารับข้อมูลที่ใช้ในการคอนฟิกแบบขนาน A0-A15 เป็นแอดเดรสที่ FPGAs สร้างให้กับ EPROM เพื่ออ่านข้อมูลจากหน่วยความจำ มาเก็บไว้ในสแตติกแรม (Static RAM) แอดเดรสทั้ง 16 เส้น ไม่จำเป็นต้องต่อให้ครบก็ได้ ขึ้นอยู่กับขนาดหน่วยความจำ EPROM ที่ใช้และสามารถกำหนดให้นับขึ้นหรือลงได้



รูปที่ 2.8 การต่อใช้งานในลักษณะมาสเตอร์ซีเรียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.9 การต่อใช้งานในลักษณะมาสเตอร์พาราเรล

2.8 การใช้ความสามารถของแรมใน FPGAs ตระกูล XC4000

แอลซีเอทำงานโดยใช้ตารางการค้นหา (Look-Up Table) ซึ่งจะทำการเก็บตารางที่ว่านี้ในสเตติกแรมซึ่งจะถูกเขียนในระหว่างการโปรแกรมโครงแบบลงบนแอลซีเอและจะถูกอ่านในการ Operation ดังนั้นแรมภายในจึงควรถูกรวมไว้ในการออกแบบของผู้ใช้ด้วย

หน้าที่ของแรมในเอฟพีจีเอตระกูล XC4000 มีหน้าที่คล้ายแรมโดยทั่วๆ ไป เช่น เอฟไอเอฟโอ (FIFO : First In First Out) แอลไอเอฟโอ (LIFO : Last In First Out) รีจิสเตอร์ไฟล์รวมทั้งแอปพลิเคชั่นบางอย่างอย่างเช่น รีจิสเตอร์เลื่อนข้อมูล (Shift Register) แรมของ FPGAs ตระกูล XC4000 มีความเร็วสูงเสมือนเอสแรม (SRAM) จึงไม่จำเป็นต้องคำนึงถึงเวลาหน่วงของการเชื่อมต่อ (Interconnection Delay)

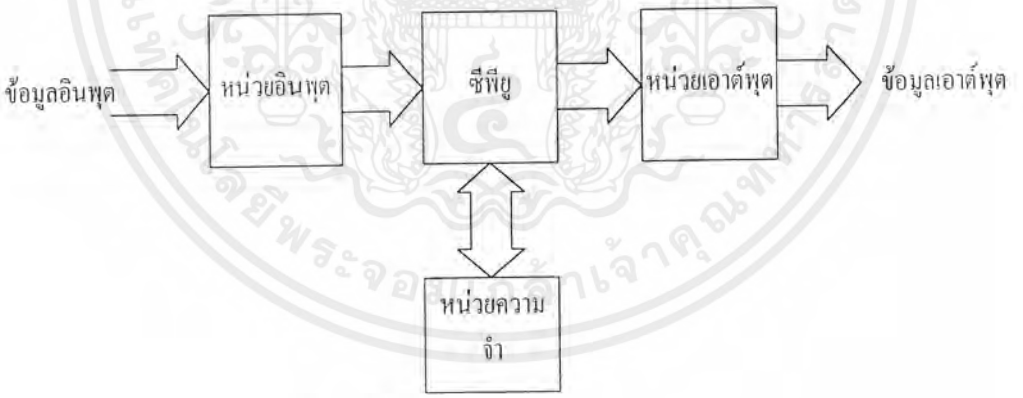
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
40169

ตารางที่ 2.3 จำนวนของแรมภายในของ FPGAs ตระกูล XC4000

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 Generators and One 3 input Function Generator (F + G + H)	100	196	400

2.9 ทฤษฎีไมโครโปรเซสเซอร์

ไมโครโปรเซสเซอร์ คือ อุปกรณ์ที่รวบรวมหน่วยต่างๆ ของคอมพิวเตอร์ อันได้แก่ หน่วยควบคุม (Control Unit) หน่วยความจำบางส่วน หน่วยคำนวณ (ALU) วงจรควบคุม I/O บางส่วน ทั้งหมดนี้รวมไว้ในแผ่นวงจรเดียวกัน ซึ่งสามารถทำหน้าที่ประมวลข้อมูลและควบคุมหน่วยอื่นๆให้ทำงานด้วยกันได้ เช่น ใช้เป็นตัวประมวลผลกลางในคอมพิวเตอร์ เรียกคอมพิวเตอร์นั้นว่า ไมโครคอมพิวเตอร์ (Microcomputer) ดังมีรูปร่างตามแผนผังนี้

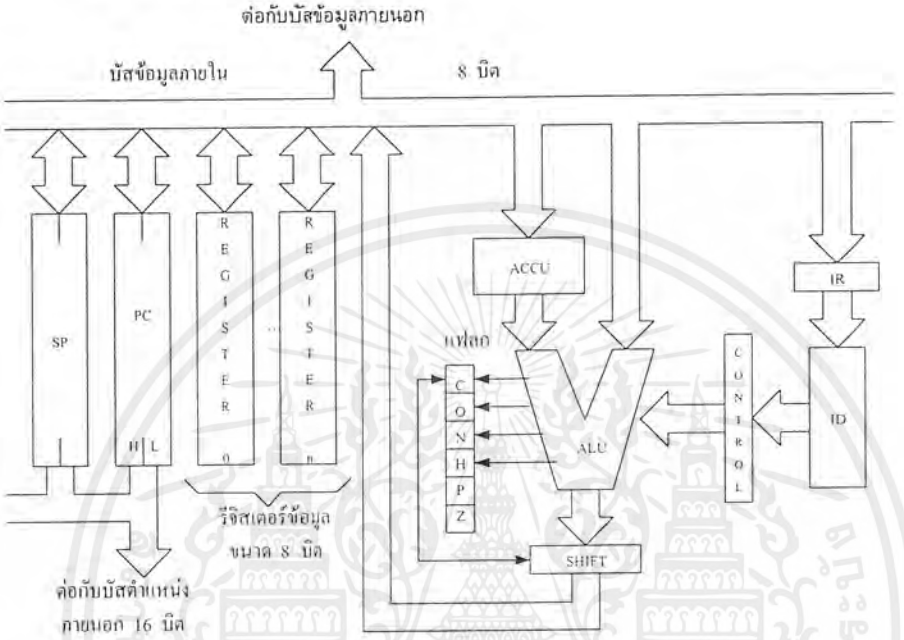


รูปที่ 2.10 ระบบไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 โครงสร้างภายในไมโครโปรเซสเซอร์

โครงสร้างไมโครโปรเซสเซอร์ขนาด 8 บิต โดยทั่วไปในปัจจุบันจะมีสถาปัตยกรรมภายในที่ใช้คล้ายๆ กัน ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 สถาปัตยกรรมภายในของไมโครโปรเซสเซอร์ 8 บิต

ส่วนประกอบภายในที่สำคัญของไมโครโปรเซสเซอร์ประกอบด้วย วงจรคณิตศาสตร์และลอจิก (Arithemetic Logic Unit หรือ ALU) รีจิสเตอร์คำสั่ง (Instruction Register หรือ IR) วงจรถอดรหัสคำสั่ง (Instruction Decode หรือ ID) หน่วยควบคุม (Control Unit) และรีจิสเตอร์ตำแหน่ง (Address Register) ส่วนประกอบที่สำคัญภายในเหล่านี้จะติดต่อกันด้วยบัสภายใน (Internal Bus) และบัสภายในนี้ต่อกับบัสภายนอกโดยผ่านบัฟเฟอร์ เพื่อเป็นการติดต่อกับวงจรภายนอกอีกทีหนึ่ง ซึ่งหน้าที่ของหน่วยต่างๆ อธิบายได้ดังนี้

1) หน่วยควบคุม (Control Unit)

หน่วยควบคุมเป็นหน่วยที่ใช้สร้างสัญญาณเพื่อควบคุมการทำงานภายในของไมโครโปรเซสเซอร์ ให้ทำหน้าที่อย่างมีระเบียบและสัมพันธ์กันโดยที่หน่วยควบคุมนี้จะรับสัญญาณมาจากวงจรถอดรหัสคำสั่ง ซึ่งจะทำการถอดรหัสคำสั่งจากรีจิสเตอร์คำสั่ง ดังนั้นการทำงานภายในไมโครโปรเซสเซอร์จึงขึ้นอยู่กับคำสั่งที่ป้อนเข้ามาให้กับไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หน่วยคณิตศาสตร์และลอจิก (Aritric Logic Unit)

หน่วยคณิตศาสตร์และลอจิกหรือเรียกย่อๆ ว่า ALU หน่วยนี้มีหน้าที่กระทำทางคณิตศาสตร์และลอจิก หน่วย ALU นี้จะใช้รีจิสเตอร์พิเศษตัวหนึ่งเป็นอินพุต ซึ่งเรียกว่า แอคคิวมูเลเตอร์ จะเป็นได้ทั้งอินพุตและเอาต์พุตของ ALU นอกจากนี้ยังสามารถใช้ในการเลื่อน (Shift) และหมุน (Rotate) ข้อมูลได้อีกด้วย

3) รีจิสเตอร์ (Register)

จากสถาปัตยกรรมของไมโครโปรเซสเซอร์ดังรูปที่ 2.11 เราอาจแบ่งรีจิสเตอร์ที่มีอยู่ภายในไมโครโปรเซสเซอร์ออกได้เป็น 2 กลุ่ม คือ รีจิสเตอร์เพื่อใช้งานทั่วไป (General Purpose Register) และรีจิสเตอร์สำหรับอ้างตำแหน่ง (Address Register) ซึ่งรีจิสเตอร์ทั้ง 2 อย่าง เราสามารถอธิบายการทำงานได้ดังนี้

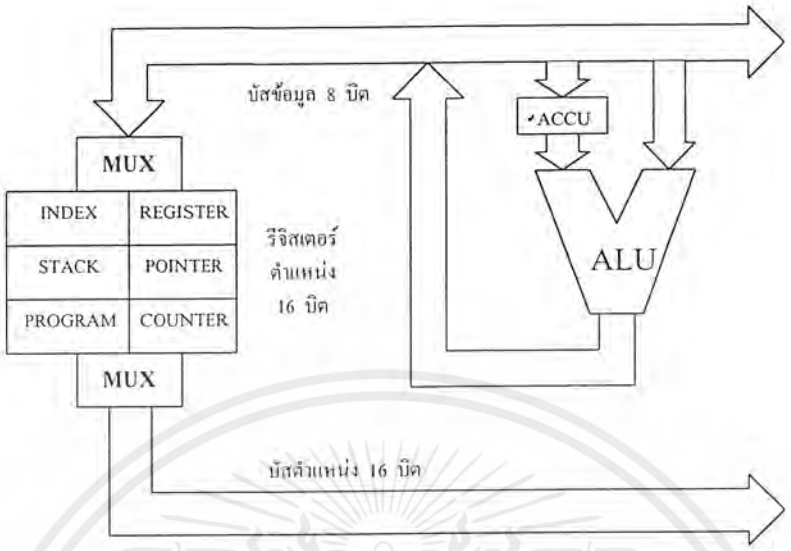
3.1) รีจิสเตอร์ใช้งานทั่วไป

ในไมโครโปรเซสเซอร์ขนาด 8 บิต รีจิสเตอร์นี้จะมีขนาด 8 บิต หน้าที่ของรีจิสเตอร์เหล่านี้ไม่ได้ถูกกำหนดเฉพาะเจาะจงลงไป แต่โดยทั่วไปจะใช้สำหรับเก็บข้อมูลเพื่อให้ ALU กระทำกับข้อมูลต่างๆเหล่านั้นด้วยความเร็วสูง เนื่องจากไม่ต้องติดต่อกับหน่วยความจำที่อยู่ภายนอก นอกจากนี้ไมโครโปรเซสเซอร์บางตัว ยังสามารถนำรีจิสเตอร์ 2 ตัวมาต่อร่วมกันได้ เรียกว่า คู่รีจิสเตอร์ (Register Pair) โดยที่คู่รีจิสเตอร์นี้จะกลายเป็นรีจิสเตอร์ขนาด 16 บิต หรือใช้เป็นตัวชี้ตำแหน่งข้อมูลในหน่วยความจำก็ได้ ซึ่งจะขึ้นอยู่กับคำสั่งที่ใช้

3.2) รีจิสเตอร์สำหรับการอ้างตำแหน่ง

หน้าที่ของรีจิสเตอร์นี้ ใช้สำหรับเก็บตำแหน่งของหน่วยความจำที่ต้องการอ้างถึง ซึ่งอาจเป็นการอ้างถึงโดยคำสั่ง หรือการอ้างถึงโดยระบบก็ตาม ขนาดของรีจิสเตอร์นี้อาจเป็น 8 บิต หรือ 16 บิต ก็ได้ ขึ้นอยู่กับชนิดของไมโครโปรเซสเซอร์นั้นๆ บางครั้งอาจเรียกรีจิสเตอร์เหล่านี้ว่า Data Counter หรือ Pointer ในไมโครโปรเซสเซอร์ทั่วๆไป ควรจะมีรีจิสเตอร์สำหรับการอ้างตำแหน่งอย่างน้อย 2 ตัวคือ โปรแกรมเคาน์เตอร์ (PC) และแสตคพอยเตอร์ (SP) ส่วนรีจิสเตอร์ตัวอื่นๆ เช่น อินเด็กซ์รีจิสเตอร์ (IX) อาจจะมีหรือไม่ก็ได้ รีจิสเตอร์นี้จะต่อไว้กับบัสตำแหน่งดังแสดงไว้ในรูปเอาต์พุตของรีจิสเตอร์เหล่านี้จะต่อไว้กับบัสตำแหน่งโดยมีตัวเลือกข้อมูล (Multiplexer) เพื่อทำหน้าที่เลือกว่าจะนำข้อมูลมาจากรีจิสเตอร์ใดเพื่อไปกำหนดตำแหน่งที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 รีจิสเตอร์ตำแหน่งขนาด 16 บิต ที่ใช้สร้างข้อมูลบนบัสตำแหน่ง

2.11 หน้าที่เฉพาะของรีจิสเตอร์สำหรับการอ้างตำแหน่งต่างๆ

2.11.1 โปรแกรมเคาน์เตอร์ (Program Counter หรือ PC)

โปรแกรมเคาน์เตอร์จะต้องมีอยู่ในทุกโปรเซสเซอร์ ข้อมูลในโปรแกรมเคาน์เตอร์ คือ ตำแหน่งของคำสั่งต่อไปที่โปรเซสเซอร์จะต้องอ่านเพื่อปฏิบัติ กลไกการปฏิบัติตามรหัสคำสั่ง และลำดับขั้นของการทำงานตามคำสั่งที่วางไว้จะกำหนดโดยข้อมูลที่อยู่ในโปรแกรมเคาน์เตอร์นี้เอง กล่าวโดยสรุปคือการปฏิบัติโปรแกรมจะเป็นไปแบบเรียงลำดับ และเพื่อที่จะดึงคำสั่งต่อไป โปรเซสเซอร์จำเป็นต้องดึงคำสั่งมาจากหน่วยความจำซึ่งกระบวนการนี้ ข้อมูลในโปรแกรมเคาน์เตอร์จะส่งมาบนบัสตำแหน่ง และส่งต่อไปยังหน่วยความจำ หน่วยความจำจะอ่านข้อมูลจากตำแหน่งมาถูกอ้างถึง และส่งข้อมูลที่อ่านได้ไปบนบัสข้อมูล โปรเซสเซอร์จะอ่านข้อมูลบนบัสข้อมูลนั้น ซึ่งข้อมูลหรือคำที่อ่านได้คือคำสั่ง (Operation Code)

2.11.2 สแตคพอยน์เตอร์ (Stack Pointer หรือ SP)

ไมโครโปรเซสเซอร์ที่มีความสามารถสูง โดยทั่วไปจะต้องมีสแตค ซึ่งสแตคอาจเป็นรีจิสเตอร์ที่อยู่ในไมโครโปรเซสเซอร์เอง หรือใช้หน่วยความจำภายนอกส่วนหนึ่ง เพื่อกำหนดให้เป็นสแตคในการเก็บรักษาตำแหน่งสูงสุดของสแตคที่อยู่ในหน่วยความจำ จะใช้รีจิสเตอร์ที่เรียกว่า สแตคพอยน์เตอร์ (SP) สแตคและสแตคพอยน์เตอร์เป็นสิ่งที่จำเป็นและขาดไม่ได้ในการทำโปรแกรมย่อย (Subroutine) และการทำโปรแกรมเกี่ยวกับการอินเตอร์รัพต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.3 อินเด็กซ์รีจิสเตอร์ (Index Register หรือ IX)

อินเด็กซ์รีจิสเตอร์ เป็นรีจิสเตอร์สำหรับการอ้างตำแหน่งอีกตัวหนึ่งที่มีในไมโครโปรเซสเซอร์บางชนิดเท่านั้น และใช้กับคำสั่งที่มีการเข้าถึงข้อมูลแบบอินเด็กซ์ (Index Address Mode) ซึ่งอินเด็กซ์รีจิสเตอร์นี้จะทำให้การเข้าถึงหน่วยความจำเป็นกลุ่มได้โดยสะดวก ข้อมูลในอินเด็กซ์รีจิสเตอร์อาจเป็นค่าระยะห่าง (Displacement) ซึ่งจะนำไปบวกกับค่าตำแหน่งฐาน (Base Address) เพื่อใช้ชี้ตำแหน่งของหน่วยความจำที่ต้องการอ้างอิง หรือข้อมูลในอินเด็กซ์รีจิสเตอร์อาจเป็นค่าตำแหน่งฐานเพื่อนำไปบวกกับค่าระยะห่างที่กำหนดมากับคำสั่งก็ได้ ทั้งนี้แล้วแต่ไมโครโปรเซสเซอร์ตัวนั้นๆ

2.12 สัญญาณติดต่อกับหน่วยความจำ

การเชื่อมต่อกับหน่วยความจำจะประกอบด้วยสัญญาณที่จำเป็น 4 กลุ่มคือ

1.) ขาสัญญาณไฟเลี้ยง

2.) ขาบัสแอดเดรส

3.) ขาบัสข้อมูล

4.) ขาบัสควบคุม

การนำข้อมูลที่จัดเก็บอยู่ภายในหน่วยความจำออกมาใช้งานนั้น จะเรียกว่าเป็น การอ่าน (Read) และการนำข้อมูลเข้าไปเก็บไว้ในหน่วยความจำเรียกว่า การเขียน (Write) กระบวนการนำข้อมูลออกมาจากหน่วยความจำและนำข้อมูลเข้าไปเก็บไว้ในหน่วยความจำนั้น จะกระทำอยู่ภายในช่วงเวลาหนึ่งมารวมเรียกว่า รอบเวลาการอ่าน (Read Cycle) และรอบเวลาการเขียน (Write Cycle) ซึ่งก็คือ ช่วงเวลาหนึ่งที่ใช้ในการอ่านและเขียนข้อมูลภายในหน่วยความจำ

ลำดับสัญญาณการติดต่อเพื่ออ่านข้อมูลจาก RAM

1.) ไมโครคอนโทรลเลอร์ส่งค่าแอดเดรสที่ต้องการออกมาทางบัสแอดเดรส

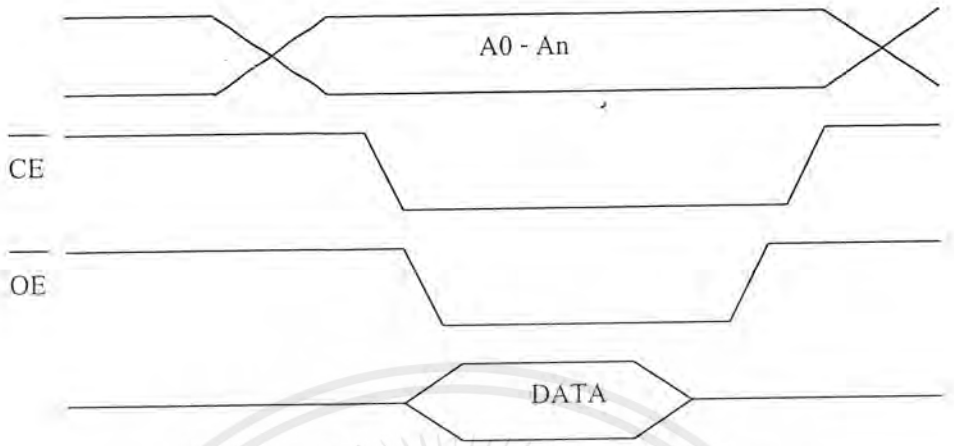
2.) สัญญาณ CE เป็นลอจิกต่ำเพื่อเลือกให้ RAM ทำงาน

3.) สัญญาณ OE หรือ RD เป็นลอจิกต่ำ เพื่อระบุว่าต้องการอ่านข้อมูลภายใน RAM

4.) ไมโครคอนโทรลเลอร์หยุดรอให้ช่วงระยะเวลาหนึ่ง เพื่อให้วงจรภายใน RAM ทำการถอดรหัสแอดเดรสและอ่านข้อมูล

5.) ข้อมูลถูกส่งออกที่บัสข้อมูล ไมโครคอนโทรลเลอร์สามารถอ่านข้อมูลนี้ไปประมวลผลต่อไป ช่วงต่อมาสัญญาณ OE กลับเป็นระดับลอจิกสูง เช่นเดิม เพื่อเป็นการทำให้สภาวะขาสัญญาณบัสข้อมูลมีอิมพีแดนซ์สูง และไม่มีผลต่อการทำงานอื่นๆ ที่เกี่ยวข้องกับบัสข้อมูลของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ลำดับสัญญาณการติดต่อเพื่ออ่านข้อมูลจาก RAM

ลำดับสัญญาณการติดต่อเพื่อเขียนข้อมูลลง RAM

1.) ไมโครคอนโทรลเลอร์ส่งค่าแอดเดรสที่ต้องการออกมาทางบัสแอดเดรส

2.) ส่งข้อมูลที่ต้องการเขียนไปบนบัสข้อมูล

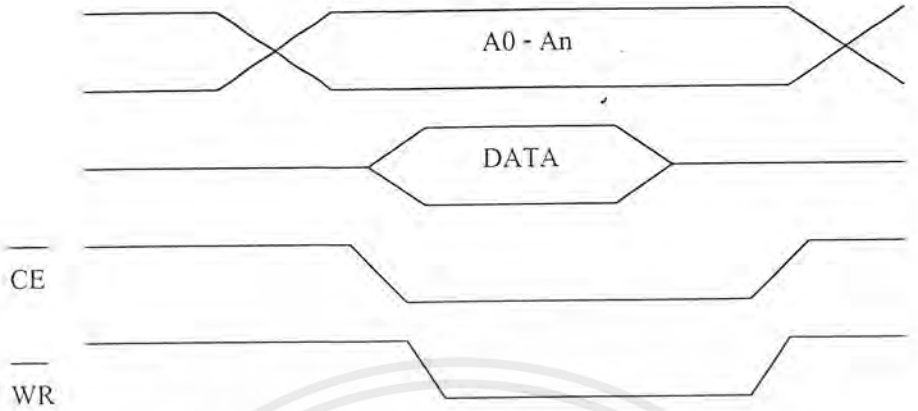
3.) สัญญาณ CE เป็นลอจิกต่ำ เพื่อเลือกให้ RAM ทำงาน

4.) ไมโครคอนโทรลเลอร์หยุดรอในช่วงระยะเวลาหนึ่ง เพื่อให้วงจรภายใน RAM ทำการถอดรหัสแอดเดรสและเตรียมการเขียนข้อมูล

5.) สัญญาณ WR เป็นลอจิกต่ำ เพื่อสั่งงานให้มีการนำข้อมูลในบัสข้อมูลเขียนลงในหน่วยความจำ

6.) ช่วงต่อมาสัญญาณ WR กลับเป็นลอจิกระดับสูง เช่นเดิม เพื่อเป็นการทำให้สภาวะขาสัญญาณบัสข้อมูลมีอิมพีแดนซ์สูง และไม่มีผลต่อการทำงานอื่นๆ ที่เกี่ยวข้องกับบัสข้อมูลของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

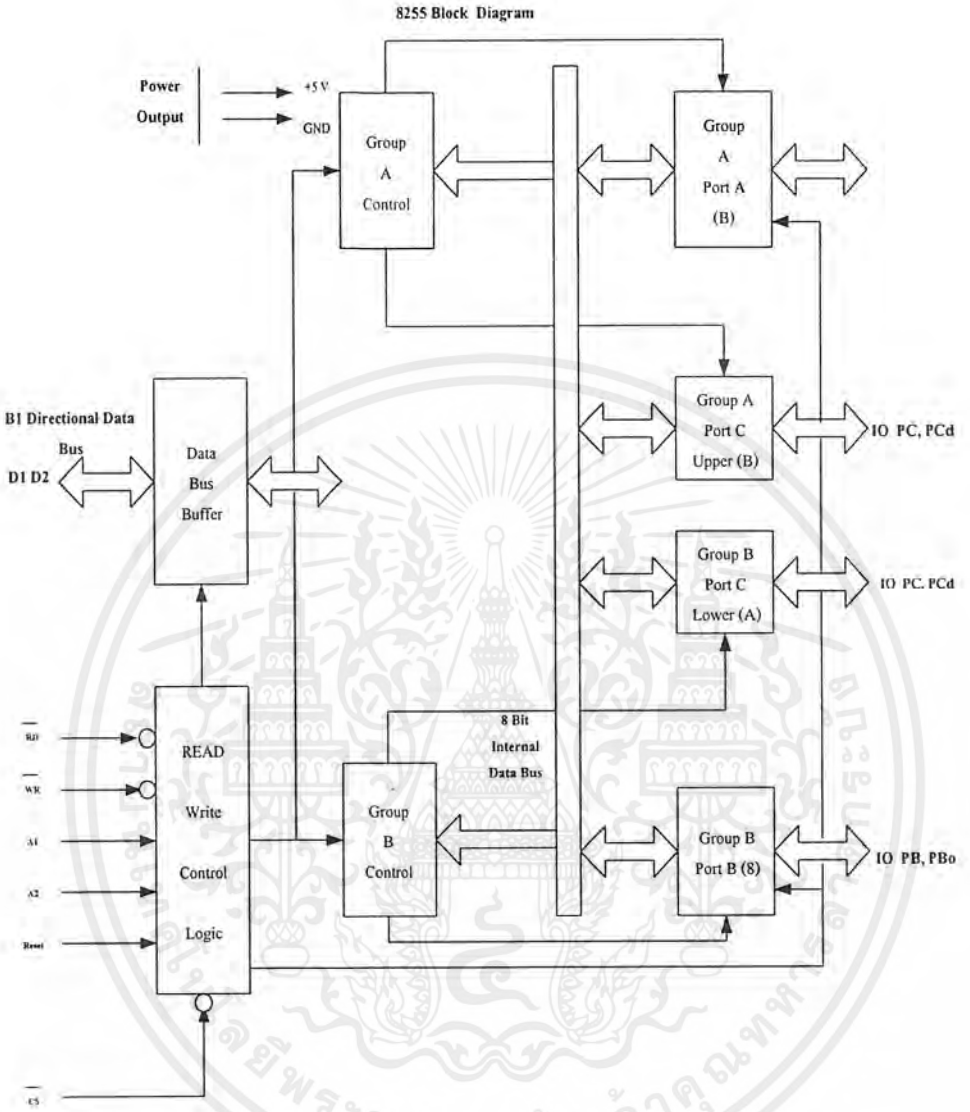


รูปที่ 2.14 ลำดับสัญญาณการติดต่อเพื่อเขียนข้อมูลลง RAM

2.13 การต่อ 8255 ใช้งาน

8255 เป็นอุปกรณ์ LSI (Large Scale Intergrated Circuit) บรรจุอยู่ใน Package 40 ขาแบบ DIP (Dual-In-Line Package) เริ่มผลิตโดยบริษัท Intel Coporation ผู้ผลิตไมโครโปรเซสเซอร์เบอร์ 8080 จุดประสงค์เพื่อใช้งานร่วมกับ 8080 โดยเฉพาะ แต่ในภายหลังได้มีการนำ 8255 ไปประยุกต์ใช้งานร่วมกับไมโครโปรเซสเซอร์อื่นๆ ดังรูปที่ 2.15 แสดงแผนผังการทำงานของ 8255 ซึ่งหน้าที่ของแต่ละบล็อกมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แผนผังการทำงานและเวลาการวางตำแหน่งขาของ 8255

บล็อกกลุ่มแรกที่เราจะพูดถึงนี้ ได้แก่ บล็อกจำนวน 4 บล็อก ที่อยู่ทางด้านขวาของรูปที่ 2.15 ซึ่งจะเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกอื่นๆ โดยมีสาย PA0-PA7, PB0-PB7 และ PC0-PC7 เป็นทางผ่านของข้อมูลระหว่างอุปกรณ์ภายนอกกับ 8255 สายสัญญาณเหล่านี้จะถูกแบ่งออกเป็น 3 I/O พอร์ต ได้แก่ A (PA), พอร์ต B (PB) และพอร์ต C (PC) พอร์ตเหล่านี้แต่ละพอร์ตสามารถเป็นได้ทั้งพอร์ตอินพุตและเอาต์พุต และแต่ละบล็อกจะมีสายสัญญาณเชื่อมเข้ากับบัสข้อมูลภายในของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกกลุ่มถัดมาได้แก่ Group A Control และ Group B Control ซึ่งจะเป็นตัวกำหนดลักษณะการทำงานทั้ง 3 I/O พอร์ต (8255 มีลักษณะการทำงานที่แตกต่างกันภายหลัง) จากรูปที่ 2.15 จะเห็นว่าพอร์ต C นี้จะประกอบด้วยพอร์ตขนาด 4 บิต 2 พอร์ต กลุ่มหนึ่งจะถูกควบคุมโดย Group A Control และอีกกลุ่มหนึ่งจะถูกควบคุมโดย Group B Control สำหรับเหตุผลนั้นจะกล่าวถึงในภายหลัง

บล็อกกลุ่มสุดท้ายที่จะกล่าวถึงได้แก่ Data Bus Buffer และ Read/Write Control Logic ซึ่งบล็อกเหล่านี้จะเป็นส่วนที่ติดต่อกับ CPU, Data Bus Buffer นี้จะเป็นบัฟเฟอร์ให้กับข้อมูลของ CPU ส่วน Read/Write Control Logic จะเป็นส่วนที่ควบคุมให้บัฟเฟอร์เข้าหรือออกจากรีจีสเตอร์ภายในตัวที่ถูกต้อง และในเวลาที่เหมาะสม

2.13.1 รายละเอียดการเรียงขาของ 8255

ในส่วนนี้เราจะพิจารณาหน้าที่ของแต่ละขาของ 8255 ซึ่งข้อมูลเหล่านี้จะมีประโยชน์ในการเชื่อมต่อเข้ากับระบบบัสของ CPU สำหรับการจัดขาแสดงไว้ในรูปที่ 1.16 รายละเอียดของขาแต่ละขามีดังนี้ คือ

D0-D7 : เป็นสายข้อมูลอินพุต/เอาต์พุตแบบสองทิศทาง (Bi-Directional Bus) จะเป็นทางผ่านของข้อมูลระหว่างพอร์ตต่างๆ ของ 8255 กับบัสข้อมูลของ Z-80

CS (Chip Select Input) : เมื่อขานี้มีสถานะลอจิกเป็น "0" CPU สามารถที่จะอ่านหรือเขียนข้อมูลกับ 8255 ได้

RD (Read Input) : เมื่อขานี้มีสถานะลอจิกเป็น "0" และสัญญาณ CS มีลอจิกเป็น "0" ข้อมูลจาก 8255 จะปรากฏสู่ระบบบัสข้อมูล CPU ก็จะสามารถอ่านข้อมูลออกไปได้ (ในการตั้งชื่อของขาสัญญาณนี้จะถือเอา CPU เป็นหลัก)

WR (Write Input) : เมื่อขานี้มีสถานะลอจิกเป็น "0"และขาสัญญาณ CS มีลอจิกเป็น "0" ข้อมูลจากระบบบัสข้อมูลจะถูกเขียนเข้าไปยัง 8255 ได้

A0-A1 (Address Input) : จะเป็นตัวกำหนดการเลือกใช้รีจีสเตอร์ภายในของ 8255 ซึ่งจะกล่าวรายละเอียดในภายหลัง

RESET : เมื่อขานี้มีสถานะเป็น "1" 8255 จะอยู่ในสภาวะรีเซ็ตทุกๆ พอร์ตของ 8255 จะถูกเซ็ตให้อยู่ในโหมดอินพุต

PA0-PA7, PB0-PB7 : ขาสัญญาณเหล่านี้จะถูกใช้เป็นพอร์ต I/O ขนาด 8 บิตใช้ต่อเข้ากับอุปกรณ์ภายนอกอื่นๆ

PC0-PC7 : ขาสัญญาณนี้ถูกใช้เป็นพอร์ต I/O ขนาด 8 บิต เช่นเดียวกับ PA0-PA7 และ PB0-PB7 แต่กลุ่มของขาสัญญาณเหล่านี้สามารถแบ่งเป็น 2 กลุ่ม โดยแต่ละกลุ่มมีขนาด 4 บิต ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มแรกจะใช้ควบคุม PB0-PB7 และกลุ่มที่ 2 ใช้ควบคุม PA0-PA7 (ซึ่งจะกล่าวถึงรายละเอียดในภายหลัง)

หน้าที่ของรีจีสเตอร์หมายเลข 0-2 จะถูกกำหนดลักษณะการทำงานจากรีจีสตอร์หมายเลข 3 (รีจีสเตอร์ควบคุม) รูปที่ 2 จะแสดงรายละเอียดของแต่ละบิตของรีจีสเอตร์ควบคุมนี้ต่อไปเราจะกล่าวถึงลักษณะการทำงานของ 8255 ทั้ง 3 โหมด และการโปรแกรมให้อยู่ในโหมดต่างๆ ได้ดังต่อไปนี้คือ

บิตที่	กลุ่ม	ความหมาย
D0	B	พอร์ต C ล่าง 1 = อินพุต 0 = เอาท์พุต
D1	B	พอร์ต B 1 = อินพุต 0 = เอาต์พุต
D2	B	เลือกโหมด 1 = โหมด 0 0 = โหมด 1
D3	A	พอร์ต C บน 1 = อินพุต 0 = เอาท์พุต
D4	A	พอร์ต A 1 = อินพุต 0 = เอาท์พุต
D5 D6	A	เลือกโหมด 0 = โหมด 0 01 = โหมด 1 1x = โหมด 2
D7		โหมดของเซ็ตแอคทีฟ 1 = แอคทีฟ

รูปที่ 2.16 รายละเอียดแต่ละบิตของรีจีสเตอร์ควบคุมของ 8255

2.13.2 โหมด 0 : Basic Register I/O

ในการเซ็ต 8255 ให้อยู่ในโหมด 0 นั้น เราจะต้องควบคุม (Control Word) ให้แก่รีจีสเตอร์ควบคุมก่อน คำสั่งควบคุมนี้จะกำหนดลักษณะการทำงานให้แก่แต่ละพอร์ตของ 8255 ตัวอย่างหนึ่งของคำสั่งควบคุมที่จะสั่งให้ 8255 ทำงานอยู่ในโหมด 0 นี้ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

รูปที่ 2.17 การเซ็ต 8255 ให้อยู่ในโหมด 0

จากรูปที่ 2.17 เราจะเห็นว่า

บิต D7 เป็นตัวกำหนดว่าเป็นตัวควบคุม (Control Word)

บิต D6 และ D5 กำหนดโหมดการทำงานของพอร์ต A D_6, D_5 มีค่าเป็น "0" แสดงว่าอยู่ในโหมด 0

บิต D4 = "0" เป็นตัวกำหนดให้พอร์ต A เป็นพอร์ตเอาต์พุต

บิต D3 = "0" เซ็ตพอร์ต C 4 บิตบนเป็นพอร์ตเอาต์พุต

บิต D2 = "0" เซ็ตโหมดของพอร์ต B เป็นพอร์ต B อยู่ในโหมด 0

บิต D1 = "0" เซ็ตพอร์ต B เป็นพอร์ตเอาต์พุต

บิต D0 = "0" เซ็ตพอร์ต C ให้ 4 บิตล่างเป็นพอร์ตเอาต์พุต

คำสั่งควบคุมนี้จะกำหนดให้พอร์ตทั้ง 3 ของ 8255 ทำงานอยู่ในโหมด 0 และเป็นพอร์ตเอาต์พุตซึ่งจะได้สายสัญญาณซึ่งสามารถติดต่อกับอุปกรณ์ภายนอกได้ถึง 24 สาย

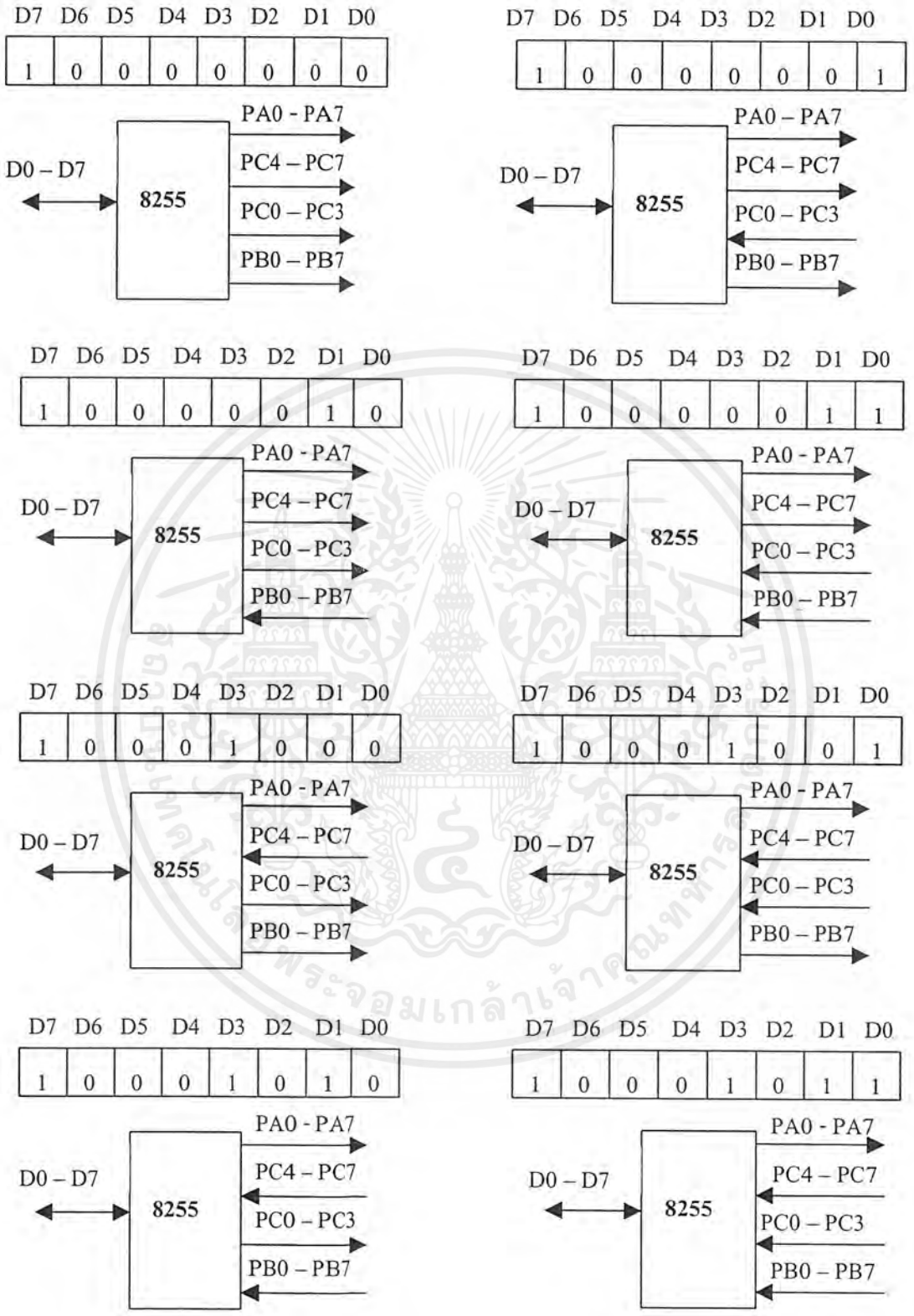
ในการทำงานในโหมด 0 ของ 8255 นี้อาจจะสั่งให้พอร์ตของ 8255 เป็นอินพุตหรือเอาต์พุตก็ได้อย่างเช่น ให้พอร์ต A และพอร์ต C เป็นพอร์ตเอาต์พุต และพอร์ต B เป็นพอร์ตอินพุต เราจะต้องส่งคำสั่งควบคุมให้แก่รีจีสเตอร์ควบคุมในลักษณะดังนี้คือ

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0

รูปที่ 2.18 ตัวอย่างการกำหนดพอร์ต B เป็นพอร์ตอินพุต

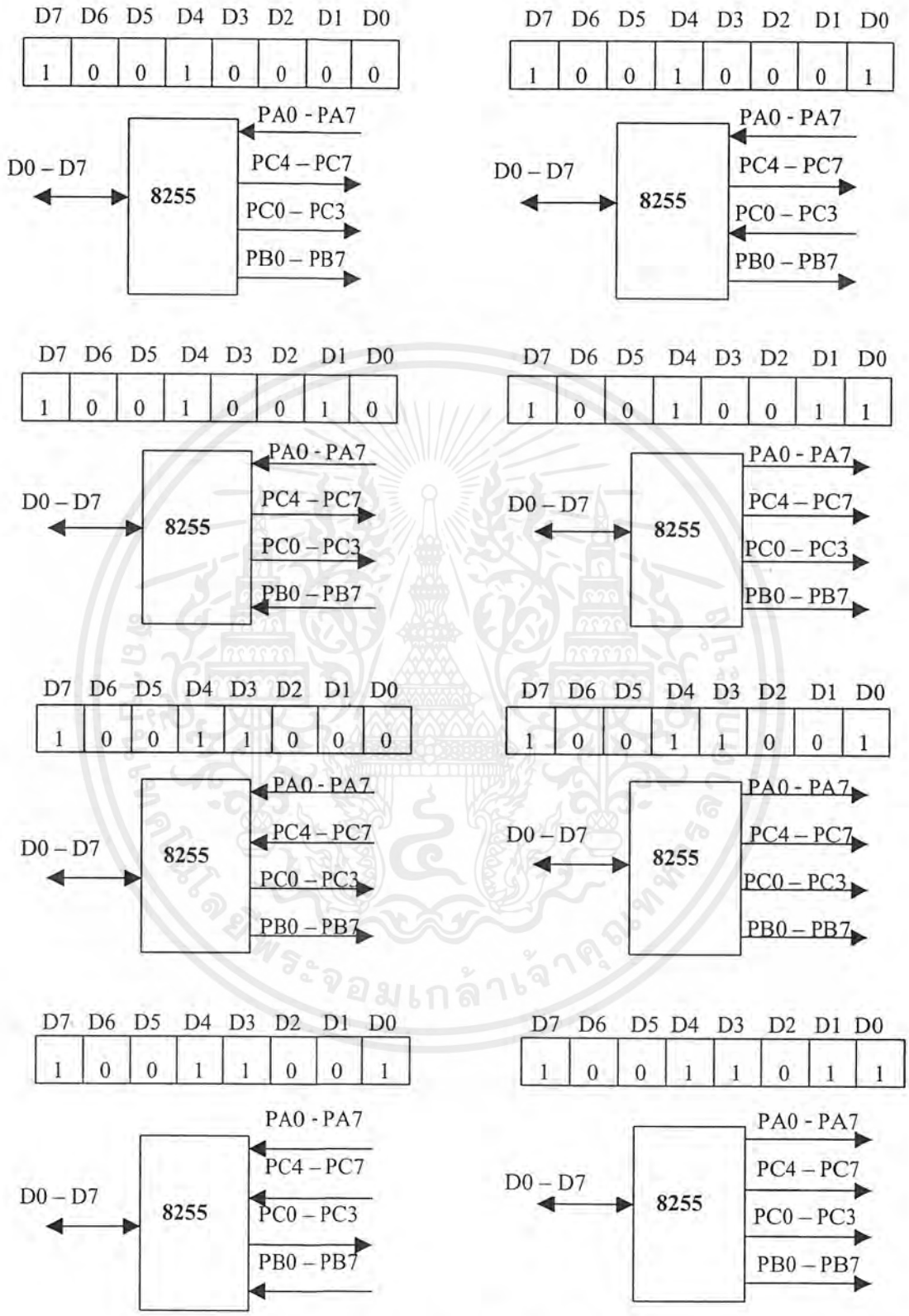
ลักษณะการทำงานของพอร์ตต่างๆ ที่สามารถกำหนดได้ในโหมด 0 แสดงไว้ในรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ลักษณะต่างๆ ในการใช้งานในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

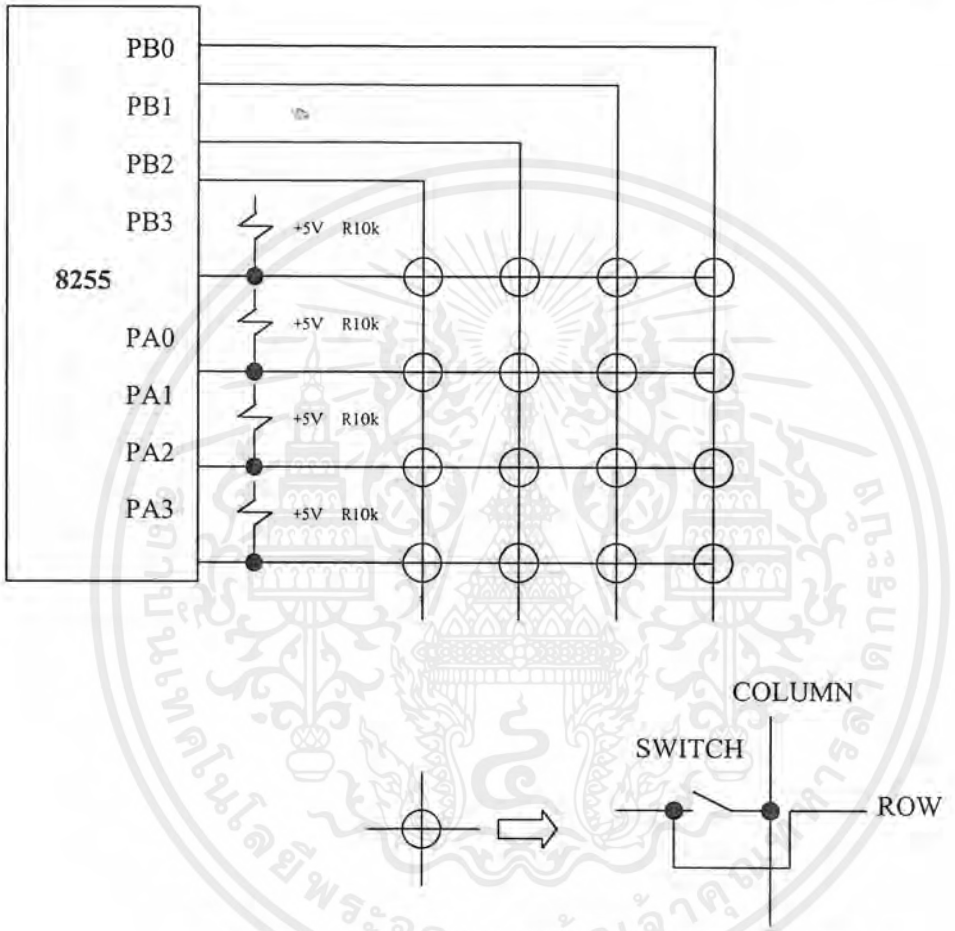


รูปที่ 2.19(ต่อ) ลักษณะต่างๆ ในการใช้งานในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.3 ตัวอย่างการใช้งาน 8255 ในโหมด 0

ตัวอย่างที่จะกล่าวนี้เป็นการเชื่อมต่อ (Interface) คีย์บอร์ดเข้ากับระบบของ Z - 80 โปรเซสเซอร์ คีย์บอร์ดที่เราจะพิจารณาถึงนี้ประกอบด้วยสวิตช์แบบกดติดปล่อยดับ SPST (Singel Pole, Singel Throw) จัดอยู่ในลักษณะของเมตริกซ์ (Matrix) ขนาด 4 x 4 ดังแสดงในรูปที่ 2.20



รูปที่ 2.20 แผนผังการต่อ 8255 เข้ากับคีย์บอร์ดขนาด 16 คีย์

โดยนำเอา PB0-PB3 ของพอร์ต B ซึ่งเป็นพอร์ตเอาต์พุต ไปต่อเข้ากับแนวคอลัมน์ (Column) ของคีย์บอร์ด และนำเอา PA0-PA3 ของพอร์ A ซึ่งเป็นพอร์ตอินพุตต่อเข้ากับแนวแถว (Row) ของคีย์บอร์ดและทุกๆ บิตของพอร์ต A ที่ต่อเข้ากับแนวโรวจะต้องต่อความต้านทานขนาด 10KΩ กับไฟบวกไว้ (Pull Up) ในการตรวจสอบการกดคีย์บอร์ดของเรานี้จะให้พอร์ต B สแกนลอจิก "0" ทีละบิตดังนี้

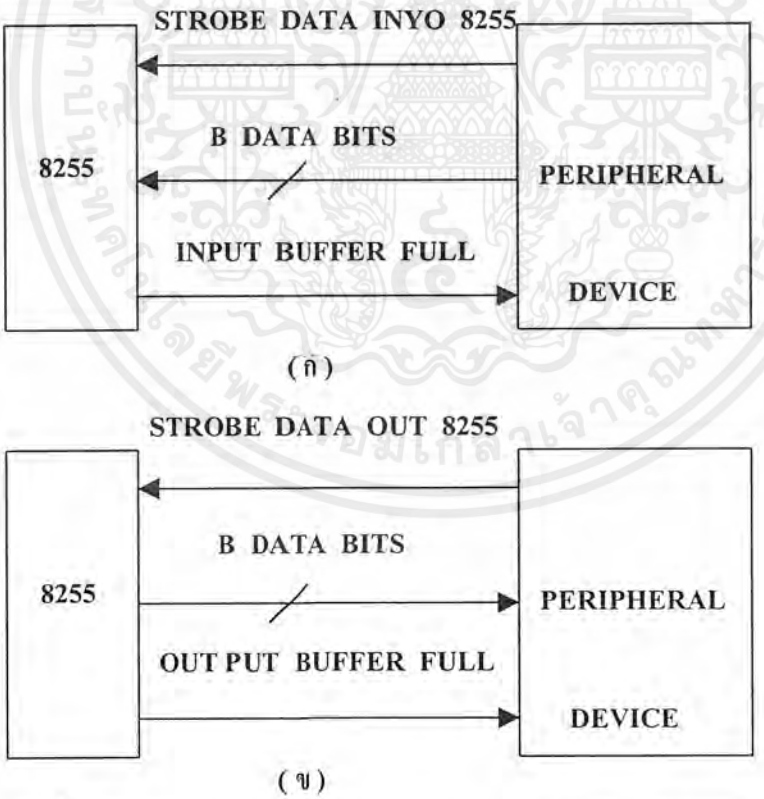
XXXX1110→XXXX1101→XXXX1011→XXXX0111→XXXX1110→.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะมีการเปลี่ยนแปลงค่าข้อมูลเอาต์พุตของพอร์ต B พอร์ต A ก็จะทำการอ่านข้อมูลเข้ามาถ้าไม่มีการกดคีย์บอร์ดแล้วค่าที่ได้จากการอ่านจากพอร์ต A จะมี 4 บิตล่าง (PA0-PA3) เป็นลอจิก "1" หมดทั้งนี้เพราะมีความต้านทานดึงขึ้นไฟบวกอยู่แต่ถ้ามีการกดคีย์บอร์ดคีย์ใดคีย์หนึ่งแล้วข้อมูลที่อ่านได้จะไม่มีค่าเป็น "1" หมด จะต้องมีบิตใดบิตหนึ่งมีลอจิกเป็น "0" มาถึงคอลัมน์ของสวิตซ์ที่ถูกกดจะทำให้แถว (Row) กับคอลัมน์ (Column) แตะกัน ในขณะที่พอร์ต B สแกนลอจิก "0" ด้วย เราจะทราบว่าคีย์ใดถูกกดโดยการตรวจสอบว่าบิตใดของพอร์ต B และของพอร์ต A มีลอจิกเป็น "0"

2.13.4 ตัวอย่างการใช้งาน 8255 ในโหมด 1

การทำงานของ 8255 ในโหมด 1 นี้เป็นการทำงานในลักษณะของการ Handshake, พอร์ต A และพอร์ต B จะเป็นพอร์ตข้อมูล ส่วนพอร์ต C นี้จะถูกใช้เป็นสัญญาณ Handshake ให้กับพอร์ต B หลังการรับส่งข้อมูลในวิธีการของ Handshake นี้ คือการให้อุปกรณ์ภายนอกส่งสัญญาณแสดงสภาวะความพร้อมให้กับ 8255 ดังแสดงในรูปที่ 2.21

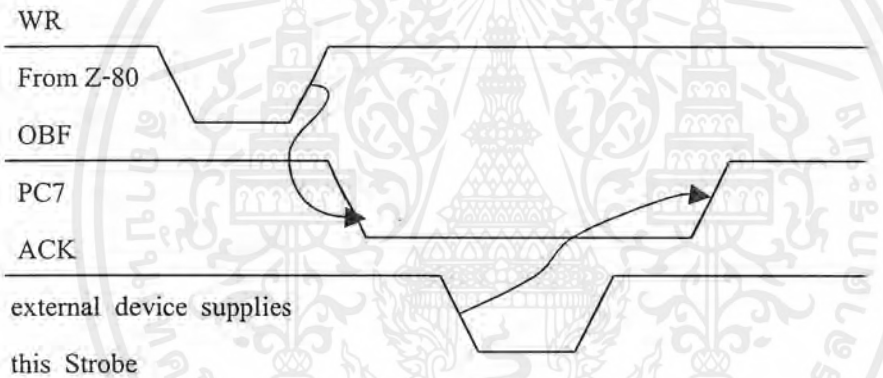


รูปที่ 2.21 แผนผังลักษณะการทำงานของการติดต่อระหว่าง 8255 กับ อุปกรณ์ภายนอกในลักษณะ Handshake

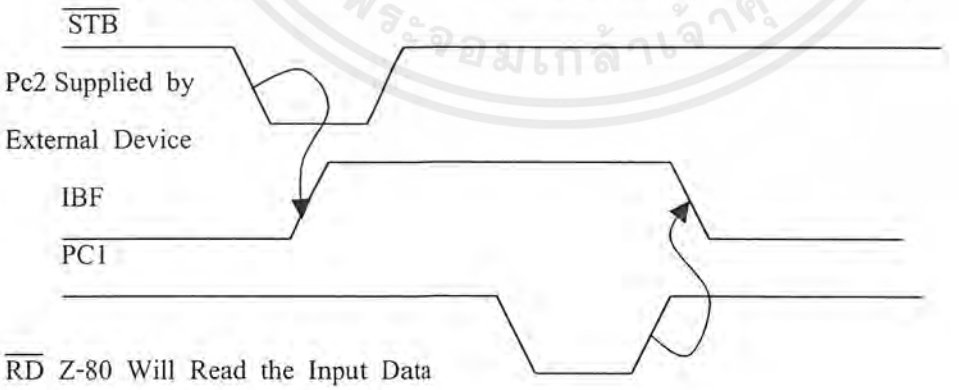
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.21 นี้ข้อมูลจะถูกส่งออกจากอุปกรณ์ภายนอกเข้าสู่ 8255 ก่อนที่อุปกรณ์ภายนอกจะเขียนข้อมูลให้แก่ 8255 จะต้องมีการตรวจสอบ Input Buffer Full Flag เสียก่อน ถ้า Flag นี้เป็นจริงแสดงว่าข้อมูลในบัฟเฟอร์ 8255 นี้ยังไม่ได้ถูกอ่านโดย Z-80 คือข้อมูลจากอุปกรณ์ภายนอกส่งข้อมูลให้กับ 8255 Z-80 ยังไม่ได้อ่านเข้าไป แต่ถ้า Flag นี้เป็นเท็จแสดงว่า Z-80 อ่านข้อมูลออกไปแล้ว อุปกรณ์ภายนอกก็จะเขียนข้อมูลใหม่ให้ 8255 ได้ และเมื่ออุปกรณ์ภายนอกเขียน ข้อมูลให้ 8255 แล้ว Input Buffer Full ก็จะกลับมาเป็นจริงอีกครั้งหนึ่ง

ในรูปที่ 2.21 (ข) 8255 จะทำหน้าที่เป็นตัวส่งข้อมูลให้กับอุปกรณ์ภายนอกก่อนที่ 8255 จะส่งข้อมูลให้กับอุปกรณ์ภายนอกจะต้องเซ็ต Output Buffer Full Flag เสียก่อนเพื่อบ่งบอกให้อุปกรณ์ภายนอกทราบว่าขณะนี้ 8255 มีข้อมูลพร้อมอยู่แล้ว Input Buffer Full Flag จะเปลี่ยนเป็นเท็จ เพื่อบ่งบอกให้อุปกรณ์ภายนอกรู้ว่าขณะนี้ไม่มีข้อมูลอยู่ใน 8255 Z-80 สามารถส่งข้อมูลใหม่ออกไปให้ 8255 ได้



รูปที่ 2.22 แผนผังเวลาของการส่งผ่านข้อมูลจากพอร์ต A ไปยังอุปกรณ์ภายนอก

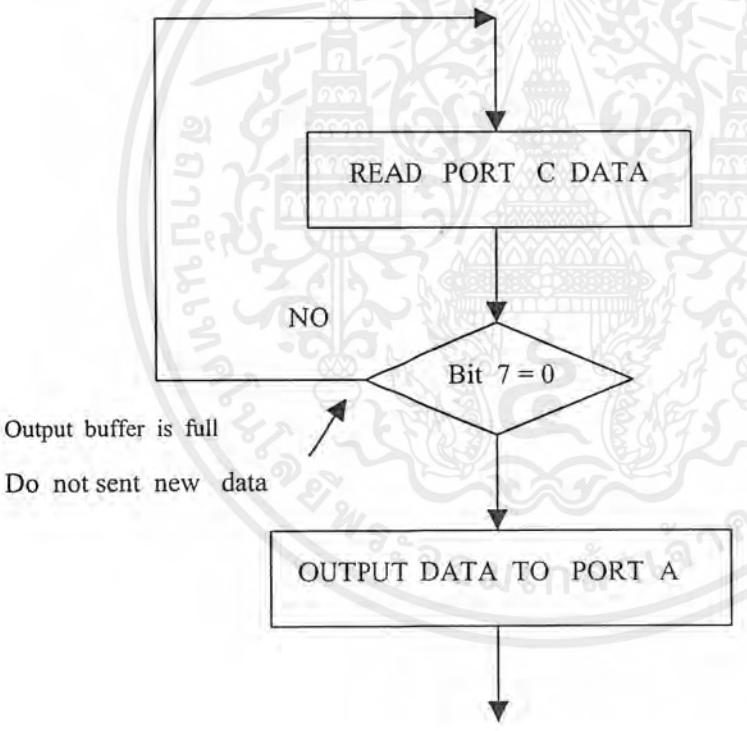


รูปที่ 2.23 แผนผังเวลาของสัญญาณที่เกิดขึ้น ขณะส่งผ่านจากอุปกรณ์ภายนอกเข้าสู่ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งข้อมูลออกไปยังอุปกรณ์ภายนอก 8255 จะต้องตรวจสอบ Output Buffer Full (PC7) ว่ามีค่าลอจิกเป็น "1" หรือไม่ ซึ่งทำได้โดยการอ่านข้อมูลเข้ามาทางพอร์ต C โดยใช้คำสั่ง Input ถ้าบิต D7 ของพอร์ต C มีลอจิกเป็น "1" ก็จะแสดงว่าอุปกรณ์ภายนอกได้รับเอาข้อมูลจาก 8255 ไปแล้ว และ Z-80 จะสามารถส่งข้อมูลใหม่ให้ 8255 ได้ แต่ถ้าบิต D7 ของพอร์ต C มีลอจิกเป็น "0" จะแสดงว่าข้อมูลที่อยู่ใน 8255 ยังไม่ถูกอุปกรณ์ภายนอกอ่านออกไป Z-80 จะไม่ส่งข้อมูลออกไป จนกว่าบิต D7 ของพอร์ต C เปลี่ยนจากลอจิก "0" เป็น "1"

ในขณะที่บิต D7 ของพอร์ต C มีลอจิกเป็น "0" อุปกรณ์ภายนอกจะรับเอาข้อมูลจาก 8255 โดยการส่งสัญญาณ Strobe ACK ให้กับบิต D6 ของพอร์ต C และเมื่อมีการรับเอาข้อมูลออกไปแล้ว บิต D7 ของพอร์ต C นี้จะเปลี่ยนระดับลอจิกจาก "0" เป็น "1" ดังรูปที่ 2.24 แสดงแผนผังของการทำงานดังกล่าว

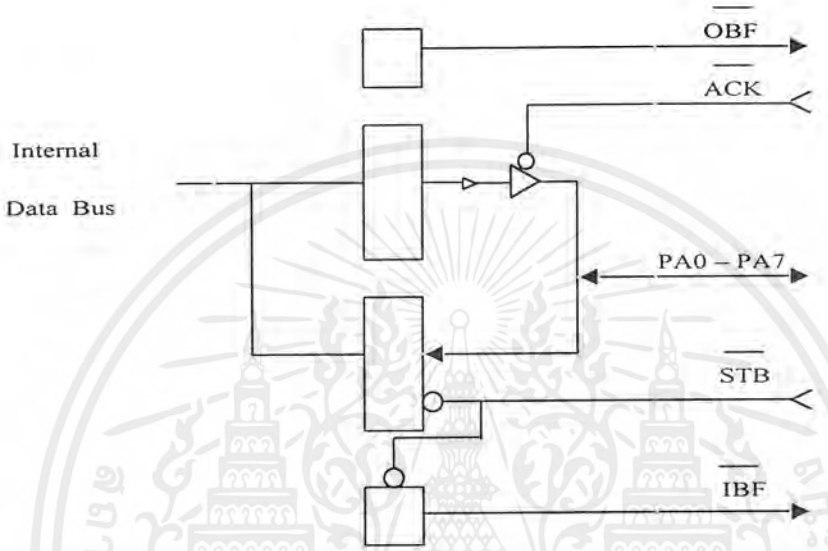


รูปที่ 2.24 แผนผังแสดงขบวนการการส่งข้อมูลออกไปยังพอร์ต A และ ใช้ 4 บิตบนของพอร์ต C เป็นสัญญาณ Handshake

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.5 ตัวอย่างการใช้งาน 8255 ในโหมด 2

การทำงานของ 8255 ในโหมด 2 นี้จะเป็นการใช้งานในลักษณะที่พอร์ต A เป็นพอร์ตข้อมูลแบบสองทิศทาง เมื่อ 8255 ถูกโปรแกรมให้พอร์ต A อยู่ในโหมด 2 นี้แล้วพอร์ต A จะมีลักษณะการทำงานตามบล็อกไดอะแกรม ตามรูปที่ 2.25



รูปที่ 2.25 แผนผังการทำงานของพอร์ต A ในโหมด 2

การทำงานในโหมดนี้ก็คือ การใช้พอร์ต A เป็นอินพุตและเอาต์พุตแลทช์ เอาต์พุตแลทช์ (Output Latch) หมายถึง การเก็บเอาข้อมูลไว้เพื่อรออุปกรณ์ภายนอกรับเอาข้อมูลออกไป ส่วนอินพุตแลทช์ (Input Latch) หมายถึง การเก็บเอาข้อมูลไว้เพื่อรออุปกรณ์ภายนอกส่งเข้ามาเพื่อรอให้ CPU อ่านเข้าไป ต่อไปเราจะพิจารณาการทำงานของบล็อกไดอะแกรมดังรูปที่ 2.25 และการศึกษาว่าจะมีการรับหรือส่งข้อมูลกับอุปกรณ์ภายนอกได้อย่างไร

ในการส่งข้อมูลให้อุปกรณ์ภายนอกเท่านั้น ขั้นตอนแรกที่เกิดขึ้นก็คือ CPU จะต้องส่งข้อมูลไปแลทช์ไว้ในพอร์ต A ซึ่งจะทำให้ Output Buffer Full (OBF) ถูกเซ็ตให้เป็นจริง สัญญาณนี้จะบอกให้อุปกรณ์ภายนอกรู้ว่าขณะนี้พอร์ต A มีข้อมูลอยู่และบอกให้ CPU รู้ว่าข้อมูลของพอร์ต A ที่ส่งไปให้นั้นยังไม่ถูกอุปกรณ์ภายนอกอ่านออกไป อุปกรณ์ภายนอกจะต้องส่งสัญญาณ $\overline{\text{ACK}}$ (Acknowledge) ให้กับ 8255 สัญญาณนี้จะเป็นการบ่งบอกให้ CPU รู้ว่าข้อมูลที่อยู่ภายในพอร์ต A ถูกอุปกรณ์ภายนอกอ่านออกไปแล้วและสามารถส่งข้อมูลใหม่ไปให้พอร์ต A ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับข้อมูลจากอุปกรณ์ภายนอกนั้น อุปกรณ์ภายนอกต้องตรวจสอบสถานะของ Input Buffer Full (IBF) เสียก่อน ถ้า IBF มีลอจิกเป็น "1" แสดงว่าขณะนี้พอร์ต A มีข้อมูลอยู่ CPU ยังไม่ได้อ่านข้อมูลเข้าไป แต่ถ้ามีลอจิกเป็น "0" แสดงว่าไม่มีข้อมูลอยู่บนพอร์ต A อุปกรณ์ภายนอกจะส่งข้อมูลและสัญญาณ Strobe (STB) ให้ 8255 สัญญาณนี้จะส่งให้มีการนำข้อมูลไปแลทช์ไว้ในพอร์ต A และเซ็ตให้ IBF มีสถานะเป็นจริง และ CPU จะตรวจสอบสถานะของ IBF ได้โดยการอ่านข้อมูลจากพอร์ต C เมื่อ IBF มีสถานะเป็นจริงแสดงว่ามีข้อมูลที่พร้อมที่จะให้ CPU อ่านออกไป เมื่อ CPU อ่านข้อมูลออกไปแล้ว IBF จะมีสถานะลอจิกเป็น "0" และอุปกรณ์ภายนอกสามารถส่งข้อมูลใหม่เข้ามาได้ ในการใช้งานในโหมด 2 นี้ พอร์ต C จะเป็นตัวแสดงสภาวะของสัญญาณดังกล่าวและรายละเอียดของแต่ละบิตจะแสดงดังรูปที่ 2.26

Port C Line	Definition
PC0	I/O
PC1	I/O
PC2	I/O
PC3	$\overline{\text{INTRA}}$
PC4	$\overline{\text{STBA}}$
PC5	$\overline{\text{IBFA}}$
PC6	$\overline{\text{ACKA}}$
PC7	$\overline{\text{OBFA}}$

รูปที่ 2.26 รายละเอียดของแต่ละขาของพอร์ต C ในการใช้งานในโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

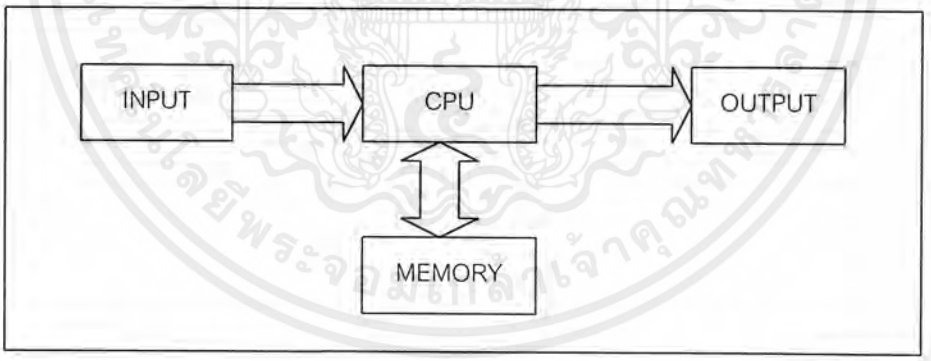
บทที่ 3
การออกแบบ การสร้างและการทำงาน

3.1 ลักษณะการออกแบบ

การออกแบบและการทำงานของหน่วยประมวลผลกลางขนาด 8 บิต โครงงานนี้ได้ทำการแยกองค์ประกอบของหน่วยประมวลผลออกเป็นส่วนๆ เมื่อออกแบบส่วนย่อยๆ แต่ละส่วนแล้วจึงทำการลดลองการทำงานของส่วนย่อยๆ นั้นจนเป็นที่น่าพอใจแล้วจึงนำส่วนย่อยที่เราได้ทำการทดลองเหล่านั้นมารวมกันให้เป็นส่วนใหญ่ ซึ่งการทำงานลักษณะนี้ทำให้ไม่มีโอกาสที่รูปแบบการทำงานจะผิดไปจากวัตถุประสงค์เพราะสามารถจำลองการทำงานของวงจรที่ออกแบบ จนแน่ใจว่าทำงานได้ตามวัตถุประสงค์ที่ต้องการ เมื่อเกิดข้อผิดพลาดก็สามารถแก้ไขได้ทันที หลังจากทำการตรวจสอบจนเป็นที่น่าพอใจแล้ว จึงนำโปรแกรมที่ได้ไปลงบนอุปกรณ์ FPGAs

3.2 โครงสร้างของระบบไมโครคอมพิวเตอร์

โครงสร้างของระบบไมโครคอมพิวเตอร์แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างของระบบไมโครคอมพิวเตอร์

3.2.1 คุณสมบัติหน่วยประมวลผลกลาง

หน่วยประมวลผลกลางขนาด 8 บิต มีคุณสมบัติดังนี้

1.1) มี Data Bus ขนาด 8 บิต

1.2) มี Address Bus ขนาด 16 บิต

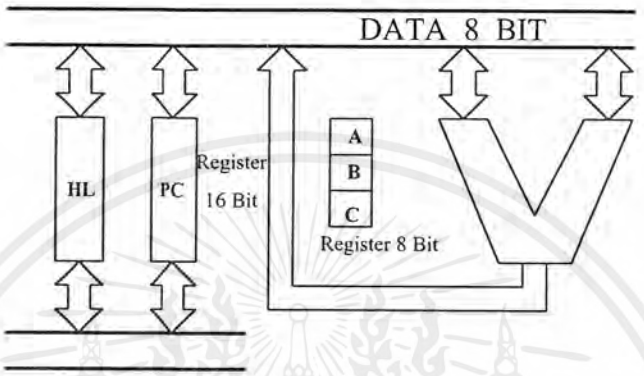
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3) สามารถกระทำคำสั่งต่างๆได้ 32 คำสั่ง

1.4) ใช้สัญญาณนาฬิกาขนาด 2 เมกะเฮริตซ์

3.2.2 โครงสร้างของหน่วยประมวลผลกลาง

โครงสร้างของหน่วยประมวลผลกลางที่ได้ออกแบบไว้ ดังแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างหน่วยประมวลผลกลาง

จากรูปที่ 3.2 จะเห็นได้ว่าโครงสร้างภายในของหน่วยประมวลผลกลาง ประกอบด้วย หน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) รีจิสเตอร์ขนาด 8 บิต และรีจิสเตอร์ขนาด 16 บิต ซึ่งแต่ละส่วนมีหน้าที่ดังนี้

1) หน่วยประมวลผลทางคณิตศาสตร์และลอจิก

มีหน้าที่ในการคำนวณทางคณิตศาสตร์ เช่น การบวกและลบเลขขนาด 4 บิต และการกระทำทางลอจิก เช่น การ AND หรือการ OR กัน

2) รีจิสเตอร์ขนาด 8 บิต

หน่วยประมวลผลกลางมีรีจิสเตอร์ใช้งานขนาด 8 บิต อยู่ 3 ตัว คือ

- รีจิสเตอร์ A เป็นแอคคิวมูเลเตอร์รีจิสเตอร์
- รีจิสเตอร์ B เป็นรีจิสเตอร์ขนาด 8 บิต
- รีจิสเตอร์ C เป็นรีจิสเตอร์ขนาด 8 บิต

3) รีจิสเตอร์ขนาด 16 บิต

หน่วยประมวลผลกลางมีรีจิสเตอร์ใช้งานขนาด 16 บิต อยู่ 2 ตัวคือ

- PC เป็นรีจิสเตอร์ที่เก็บตำแหน่งของคำสั่งต่อไปที่หน่วยประมวลผลกลางจะทำงานต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- HL เป็นรีจิสเตอร์ที่เก็บตำแหน่งขนาด 16 บิต

3.2.3 รายละเอียดของคำสั่ง

โครงงานนี้ได้จัดแบ่งกลุ่มของคำสั่งต่างๆไว้ดังนี้

- กลุ่มคำสั่งโอนย้ายข้อมูล

  LOAD

- กลุ่มคำสั่งคณิตศาสตร์และลอจิก

  ADD, ADC, SUB , AND, OR, INC, DEC, XOR, NAND, XNOR

- กลุ่มคำสั่งการกระโดด

  JP

- กลุ่มคำสั่งรับส่งข้อมูลกับอุปกรณ์ภายนอก

  IN, OUT

กลุ่มของคำสั่งที่ได้ทำการแบ่งไว้นั้น สามารถนำมาแจกแจงรายละเอียดของคำสั่งต่างๆตามกลุ่มที่ได้แบ่งไว้ ดังแสดงตามตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดคำสั่งต่างๆ

คำสั่ง	Op-Code	การทำงาน
NOP	00H	หยุดการทำงาน
LD A,data	01nH	นำข้อมูลไปเก็บไว้ใน A
LD B,data	02nH	นำข้อมูลไปเก็บไว้ใน B
LD HL,data	03nnH	นำข้อมูลไปเก็บไว้ใน HL
LD A,B	04H	นำข้อมูลที่ B ไปเก็บไว้ใน A
LD B,A	05H	นำข้อมูลที่ A ไปเก็บไว้ใน B
LD A,(HL)	06nnH	นำตำแหน่งของ HL ไปชี้ที่ A
LD (HL),A	07nnH	นำข้อมูลที่ A ไปเก็บไว้ใน HL
ADD	08H	เป็นคำสั่งบวก
ADC	09H	เป็นคำสั่งบวกที่คิดตัวทด
SUB	0AH	เป็นคำสั่งลบ
INC A	0BH	เพิ่มค่าที่ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

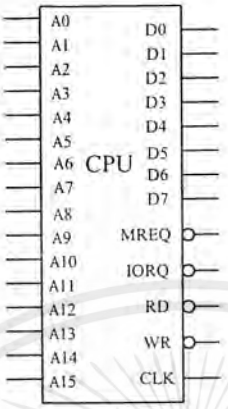
ตารางที่ 3.1(ต่อ) รายละเอียดคำสั่งต่างๆ

คำสั่ง	Op-Code	การทำงาน
INC B	0CH	เพิ่มค่าที่ B
INC HL	0DH	เพิ่มค่าที่ HL
DEC A	0EH	ลดค่าที่ A
DEC B	0FH	ลดค่าที่ B
DEC HL	10H	ลดค่าที่ HL
AND	11H	นำข้อมูลมา AND กับข้อมูลที่ A
OR	12H	นำข้อมูลมา OR กับข้อมูลที่ A
XOR	13H	นำข้อมูลมา XOR กับข้อมูลที่ A
NAND	14H	นำข้อมูลมา NAND กับข้อมูลที่ A
NOR	15H	นำข้อมูลมา NOR กับข้อมูลที่ A
XNOR	16H	นำข้อมูลมา XNOR กับข้อมูลที่ A
CPL A	17H	กลับค่า A
CPL B	18H	กลับค่า B
CMP A,B	19H	เปรียบเทียบค่า A กับ B output ออกที่ flag
JP data	1AnnH	กระโดดไปที่ตำแหน่ง data
DJNZ	1BH	กระโดดเมื่อ HL ไม่เท่ากับ "0"
SET C	1CH	เซต carry flag ให้เป็น “1”
CLR C	1DH	เคลียร์ carry flag
IN	1EH	นำข้อมูลเก็บที่ A ณ.ตำแหน่งของ HL
OUT	1FH	นำข้อมูลออกที่ A ณ.ตำแหน่งของ HL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 หน้าที่การทำงานของขาสัญญาณแต่ละขา

ขาสัญญาณต่างๆ ในหน่วยประมวลผลกลางดังแสดงได้ในรูปที่ 3.3



รูปที่ 3.3 ขาใช้งานของหน่วยประมวลผลกลาง

ขา CLK เป็นขาอินพุตทำงานที่ขอบขาขึ้น ทำหน้าที่รับสัญญาณฐานเวลาให้แก่วงจรของหน่วยประมวลผลกลางขนาด 8 บิต

ขา $A_{15} - A_0$ เป็นขาสัญญาณกำหนดตำแหน่ง (Address Bus) โดยที่ A_0 เป็นบิตทางด้านต่ำ (LSB) และ A_{15} เป็นบิตทางด้านสูง (MSB) จะทำงานที่สภาวะ "1" บัสนี้มีด้วยกัน 16 เส้น สามารถติดต่อกับหน่วยความจำได้ $2^{16} = 65,536$ ตำแหน่ง (64 Kbytes)

ขา $D_7 - D_0$ เป็นขาสัญญาณข้อมูล (Data Bus) D_0 เป็นบิตทางด้านต่ำ และ D_7 เป็นบิตทางด้านสูง ลักษณะเป็นบัสแบบสองทิศทาง ขนาด 8 บิต ทำงานสภาวะ "1" ใช้เป็นเส้นทางผ่านของข้อมูลระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำหรืออุปกรณ์อินพุต/เอาต์พุตต่างๆ

ขา MREQ (Memory Requet) ทำงานที่สภาวะ "0" เมื่อสายสัญญาณนี้ทำงาน แสดงว่าขณะนี้ไมโครโปรเซสเซอร์ต้องการติดต่อกับหน่วยความจำเพื่ออ่านหรือเขียนข้อมูลโดยที่ตำแหน่งของหน่วยความจำอยู่บนบัสแอดเดรสแล้ว

ขา IORQ (Input/Output Request) ทำงานที่สภาวะ "0" เมื่อสายสัญญาณนี้ทำงาน แสดงว่าขณะนี้อุปกรณ์อินพุต/เอาต์พุต ต้องการติดต่อกับไมโครโปรเซสเซอร์หรือหน่วยความจำเพื่ออ่านหรือเขียนข้อมูล

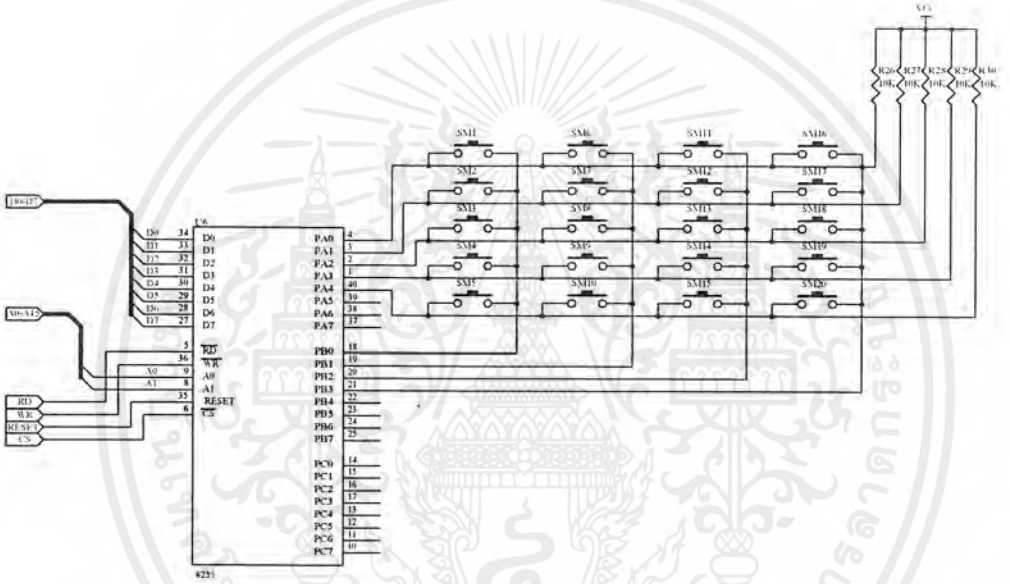
ขา RD (Memory Read) เป็นขาเอาต์พุต ทำงานที่สภาวะ "0" สัญญาณนี้ทำงานขณะที่ไมโครโปรเซสเซอร์ต้องการอ่านข้อมูลจากหน่วยความจำ หรือจากอุปกรณ์อินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา WR (Memory Write) เป็นขาเอาต์พุต ทำงานที่สภาวะ "0" เมื่อสัญญาณนี้แอคตีฟ แสดงว่าไมโครโปรเซสเซอร์ต้องการเขียนข้อมูลเข้าหน่วยความจำหรือเข้าอุปกรณ์อินพุต/เอาต์พุต

3.3 ส่วนรับข้อมูล

ส่วนรับข้อมูลจะทำหน้าที่รับข้อมูลมาจากเมตริกซ์สวิตช์ โดยตรวจสอบการกดคีย์ผ่านทาง 8255 ว่าคีย์ที่กดเป็นคีย์ใด และได้กำหนดคีย์ที่ใช้กดไว้ทั้งหมด 16 คีย์ คือ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, RUN, ENTER, RESET, FUNC ดังแสดงในรูปที่ 3.4



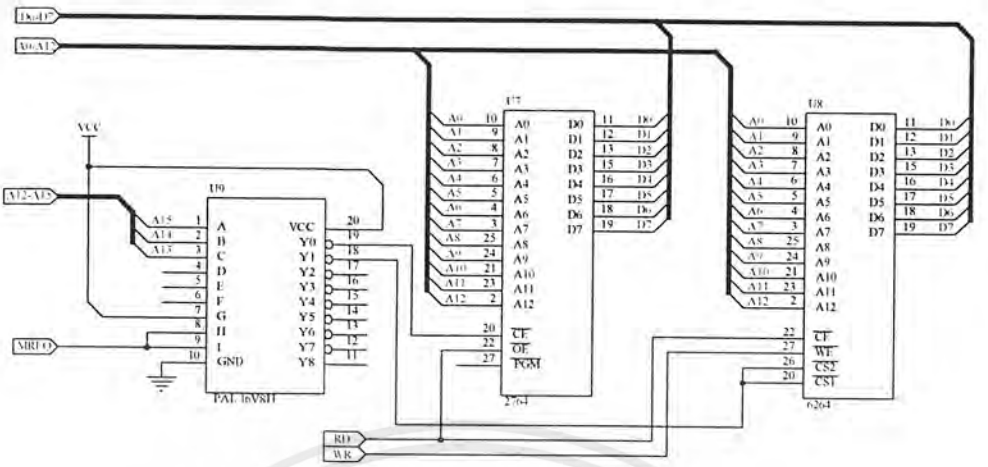
รูปที่ 3.4 วงจรส่วนรับข้อมูล

3.4 หน่วยความจำ (Memory Unit)

หน่วยความจำสามารถแยกออกเป็น หน่วยความจำภายในและหน่วยความจำภายนอก ซึ่งหน่วยความจำภายในออกแบบโดยใช้ EPROM เบอร์ 2764 มีความจุ 8 กิโลไบต์ ทำหน้าที่เก็บส่วนของคำสั่งภายในหน่วยประมวลผลกลาง และหน่วยความจำภายนอกใช้ RAM เบอร์ 6264 มีความจุ 8 กิโลไบต์

การต่อใช้งาน EPROM และ RAM นั้น จะต้องใช้สัญญาณที่ได้มาจากการดีโค๊ดแอดเดรสของ EPROM และ RAM โดยกำหนดให้แอดเดรสของ EPROM คือ 0000H–1FFFH แอดเดรสของ RAM คือ 2000H – 3FFFH เมื่อดีโค๊ดได้แล้วได้เลือกใช้ IC74LS138 เพื่อเลือกขาที่จะไปต่อใช้งานดังแสดงในรูปที่ 3.5

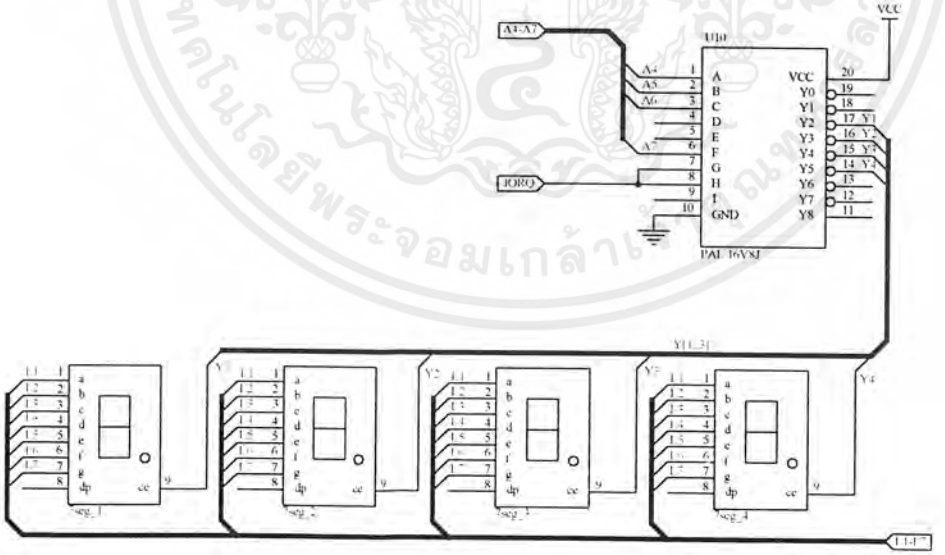
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การต่อ EPROM และ RAM

3.5 หน่วยแสดงผล (Output)

หน่วยแสดงผลจะใช้หน่วยแสดงผลเจ็ดส่วน (7 – Segment) จำนวน 4 หลัก เป็นตัวแสดงผลโดยการดีโค๊ดแอดเดรสผ่านโดยใช้ IC 47LS138 ซึ่งทำหน้าที่เป็นตัวเลือกตำแหน่งให้หน่วยแสดงผลเจ็ดส่วน แสดงผลดังแสดงไว้ในรูปที่ 3.6



รูปที่ 3.6 วงจรภาคแสดงผลหน่วยแสดงผลเจ็ดส่วน

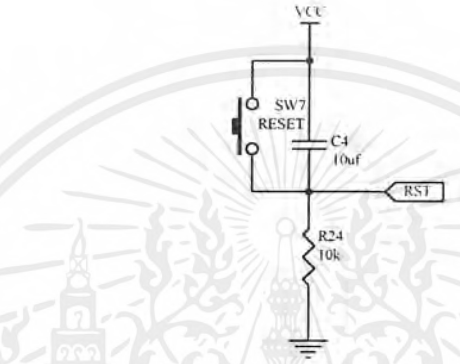
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ส่วนประกอบร่วมของวงจร

เพื่อให้วงจรสามารถทำงานได้สมบูรณ์ขึ้นจึงต้องมีส่วนประกอบต่างๆดังนี้

3.6.1 วงจรรีเซ็ต

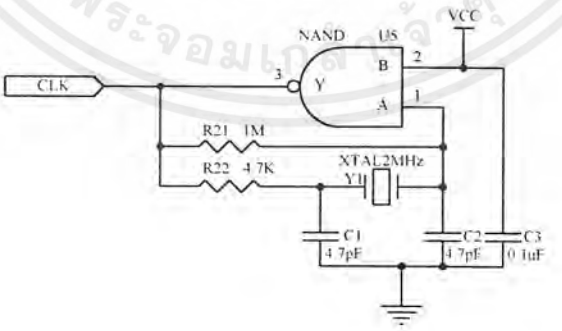
วงจรรีเซ็ตใช้ในการรีเซ็ตระบบไมโครโปรเซสเซอร์ โดยทำการรีเซ้ตที่ 8255 และตัว CPU เอง ให้พร้อมที่จะเริ่มทำงาน โดยเมื่อกดรีเซ็ตจะส่งสภาวะ "0" ไปทำการรีเซ็ตระบบ วงจรรีเซ็ตแสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 วงจรรีเซ็ต

3.6.2 วงจรกำเนิดสัญญาณนาฬิกา

วงจรกำเนิดสัญญาณนาฬิกาใช้กำเนิดสัญญาณนาฬิกาที่ความถี่ 2 MHz โดยใช้คริสตรัลความถี่ 2 MHz และอุปกรณ์อื่นเป็นตัวช่วยกำเนิดสัญญาณนาฬิกา ตามรูปที่ 3.8

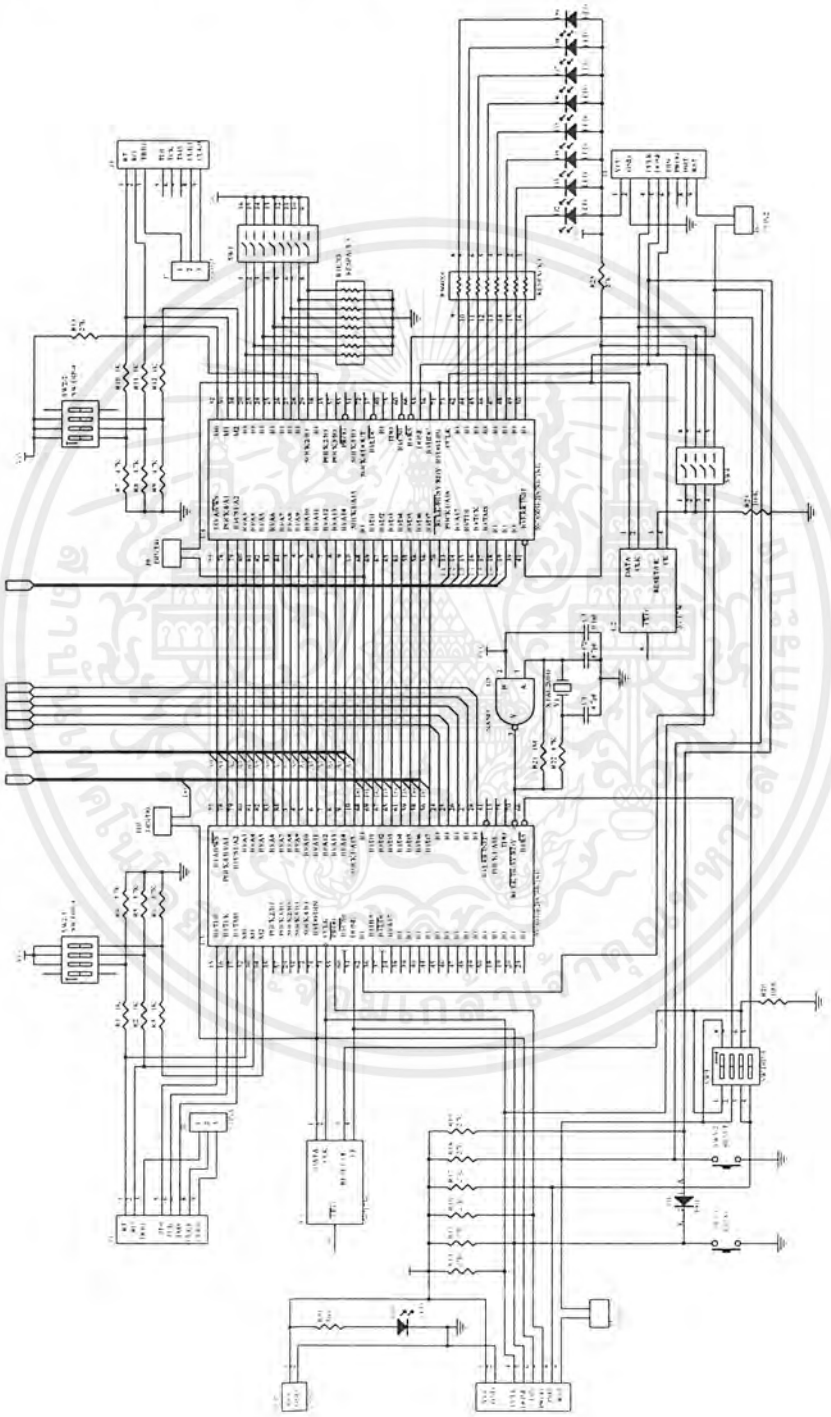


รูปที่ 3.8 วงจรกำเนิดสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

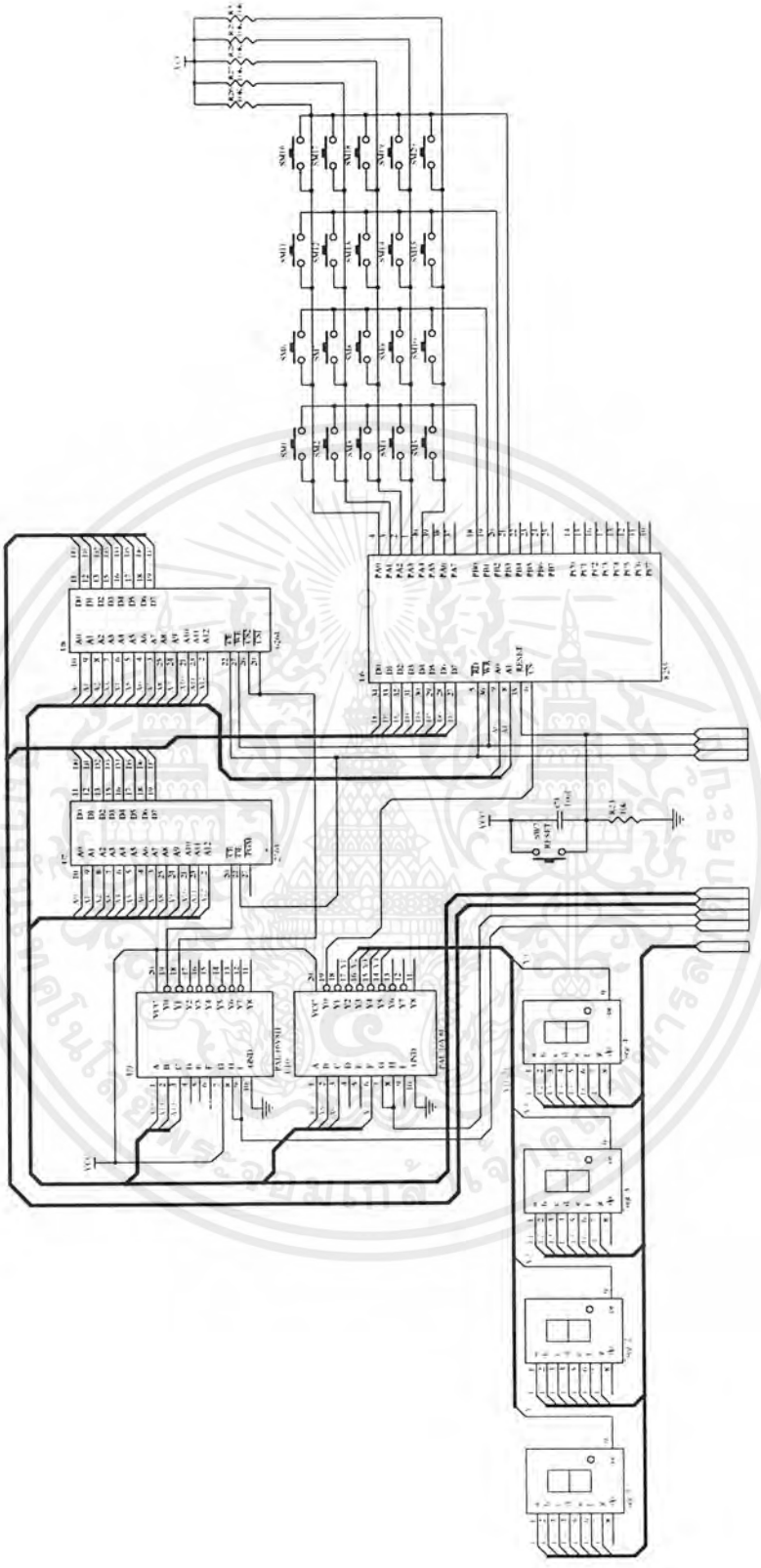
3.7 วงจรประยุกต์การใช้งาน

จากวงจรที่ได้ออกแบบมาแล้วทั้งหมดข้างต้น สามารถนำมารวม และสร้างเป็นวงจรใหม่ ดังรูปที่ 3.9 เพื่อนำมาประยุกต์ใช้งานได้หลายรูปแบบ



รูปที่ 3.9 วงจรประยุกต์การใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

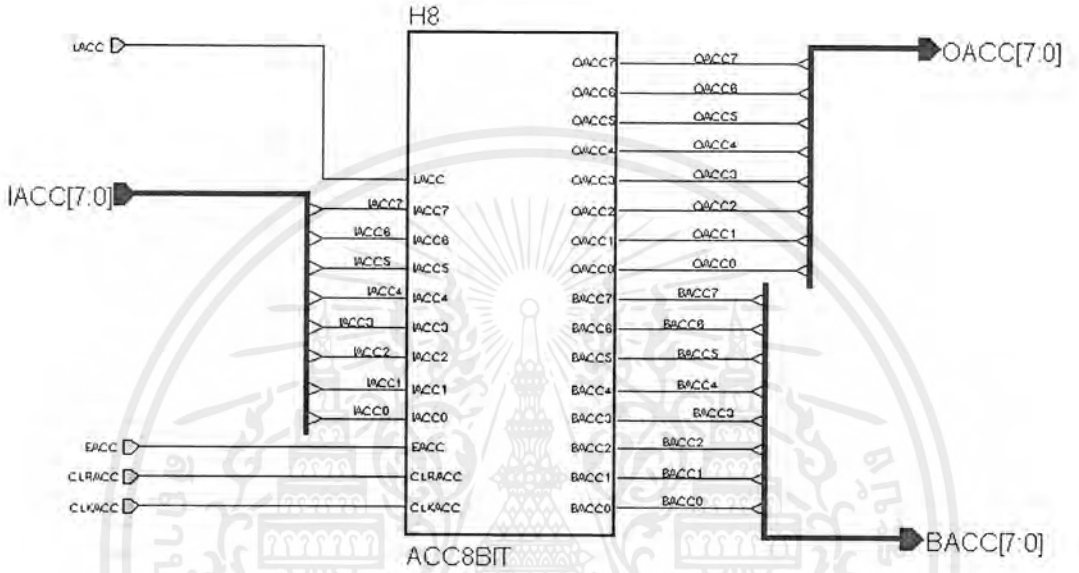


รูปที่ 3.9(ต่อ)วงจรประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 ลำดับการสร้างและทดลอง

3.8.1 การออกแบบด้วยโปรแกรม Xilinx Foundation Series

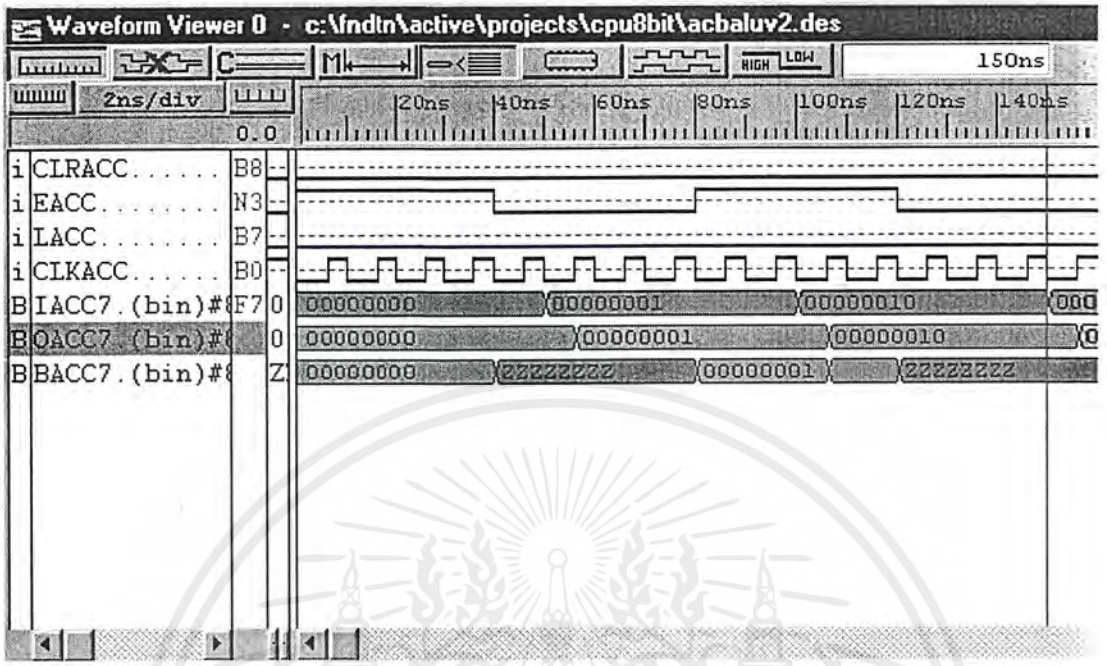


รูปที่ 3.11 Schematic ของรีจิสเตอร์ ACC

3.8.2 การจำลองการทำงาน

หลังจากทำการออกแบบหน่วยประมวลผลกลางให้อยู่ในรูป Schematic เรียบร้อยแล้ว และได้ทำการตรวจสอบแก้ไขจนไม่พบข้อผิดพลาดในขั้นต้นแล้ว ขั้นต่อมาก็คือการทดสอบการทำงานของ Schematic ที่เขียนขึ้นมาดูว่าการทำงานเป็นไปตามวัตถุประสงค์ที่ได้ออกแบบไว้หรือไม่ ในการจำลองการทำงานจะต้องสร้างสัญญาณทดสอบวงจรทุกสภาวะ เพื่อให้แน่ใจว่าผลลัพธ์ที่ได้นั้นตรงตามที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

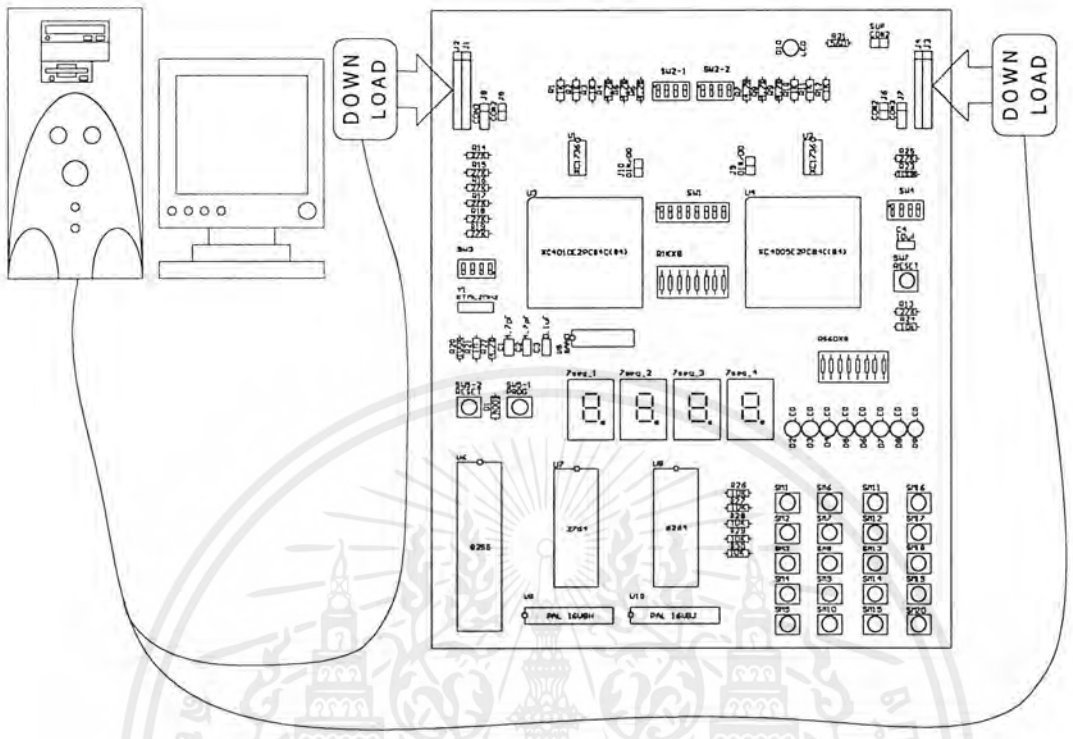


รูปที่ 3.12 การจำลองการทำงานของการ รีจิสเตอร์ ACC

3.8.3 การโปรแกรมลงบนไอซี FPGA

หลังจากนำแต่ละส่วนที่ได้ทำการออกแบบเชื่อมต่อกันแล้วจะได้ตัวหน่วยประมวลผลกลาง ขนาด 8 บิต แล้วขั้นตอนต่อมาเป็นการนำเอาวงจรที่ได้มาทำการดาวน์โหลดลงไอซี FPGA โดยใช้ซอฟต์แวร์ XACT Development Tools ของบริษัทไซลิงค์ในการดาวน์โหลดวงจรหน่วยประมวลผลกลางขนาด 8 บิต ที่ได้ลงบนเซลล์ต่างๆ ภายในไอซี FPGAs ผลสุดท้ายจะได้ออกมาเป็นวงจรหน่วยประมวลผลกลางขนาด 8 บิตบนไอซี FPGAs ซึ่งสามารถนำไปสร้างเป็นวงจรต้นแบบได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 การเชื่อมต่อกันระหว่างคอมพิวเตอร์กับบอร์ดทดลองของ FPGAs

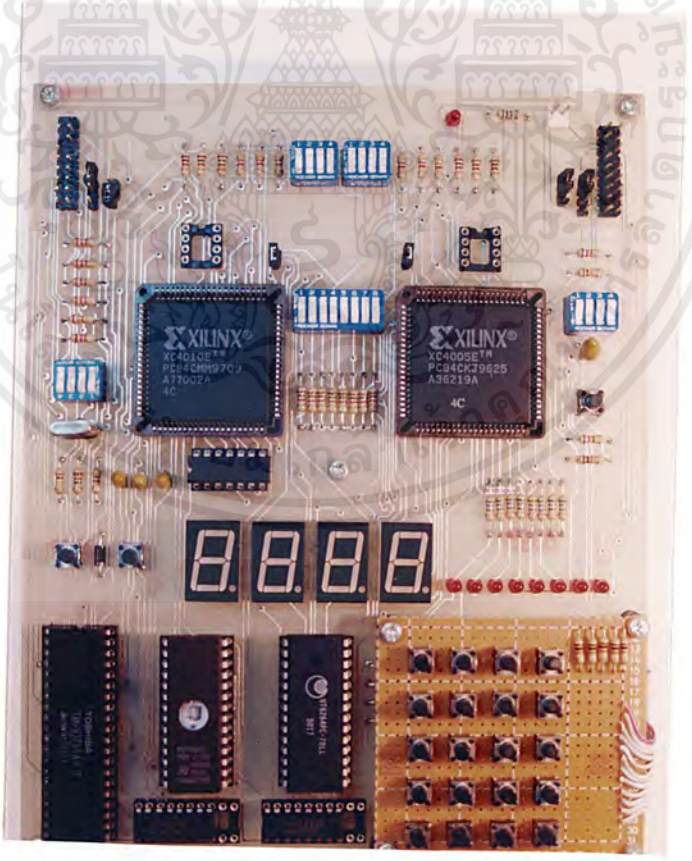
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

การออกแบบหน่วยประมวลผลกลางขนาด 8 บิต คือ ออกแบบในแต่ละส่วนของตัวประมวลผลกลางให้อยู่ในรูป Schematic โดยใช้โปรแกรม Xilinx Foundation Series แล้วนำไปจำลองการทำงานของแต่ละส่วนจนได้ผลตามที่ได้ออกแบบแล้ว นำไปรวมเป็นส่วนกลางจำลองการทำงานอีกครั้งหนึ่ง เมื่อได้ผลตามที่ได้ออกแบบแล้ว นำโปรแกรมไปดาวน์โหลดลงบนอุปกรณ์ FPGAs โดยทำการเลือก FPGAs เบอร์ 4010E เป็นหน่วยประมวลผลกลาง และเลือก FPGAs เบอร์ 4005E ทำหน้าที่เป็นอุปกรณ์ส่วนทดลองทางด้านโปรแกรม สำหรับบอร์ดที่ได้ทำการออกแบบนั้น มีลักษณะดังรูปที่ 4.1



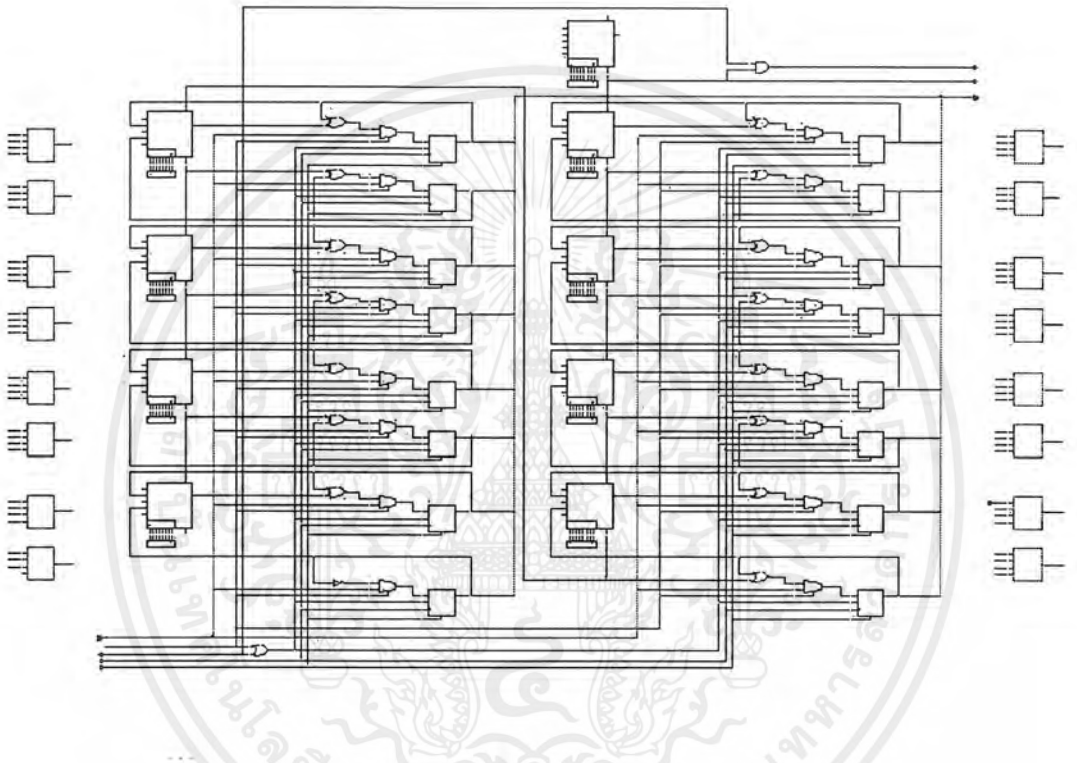
รูปที่ 4.1 บอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบส่วนย่อย

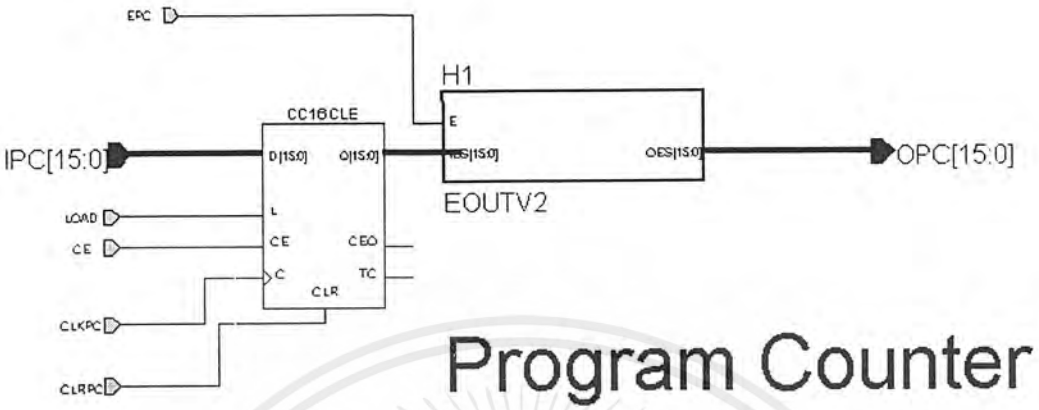
4.2.1 การออกแบบ Program Counter

จากรูปที่ 4.2 เป็นการแสดงวงจรภายในของ Program Counter ซึ่งประกอบไปด้วยเกตต่างๆ และฟลิปฟลอปต่างๆ

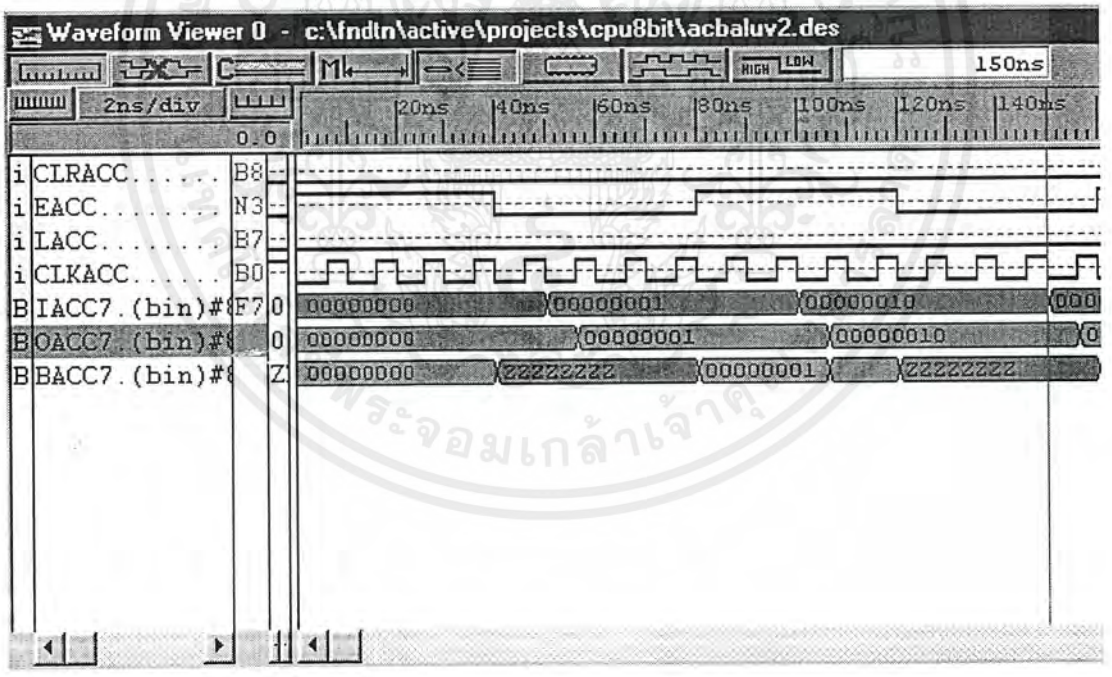


รูปที่ 4.2 วงจรโครงสร้างภายในของ Program Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



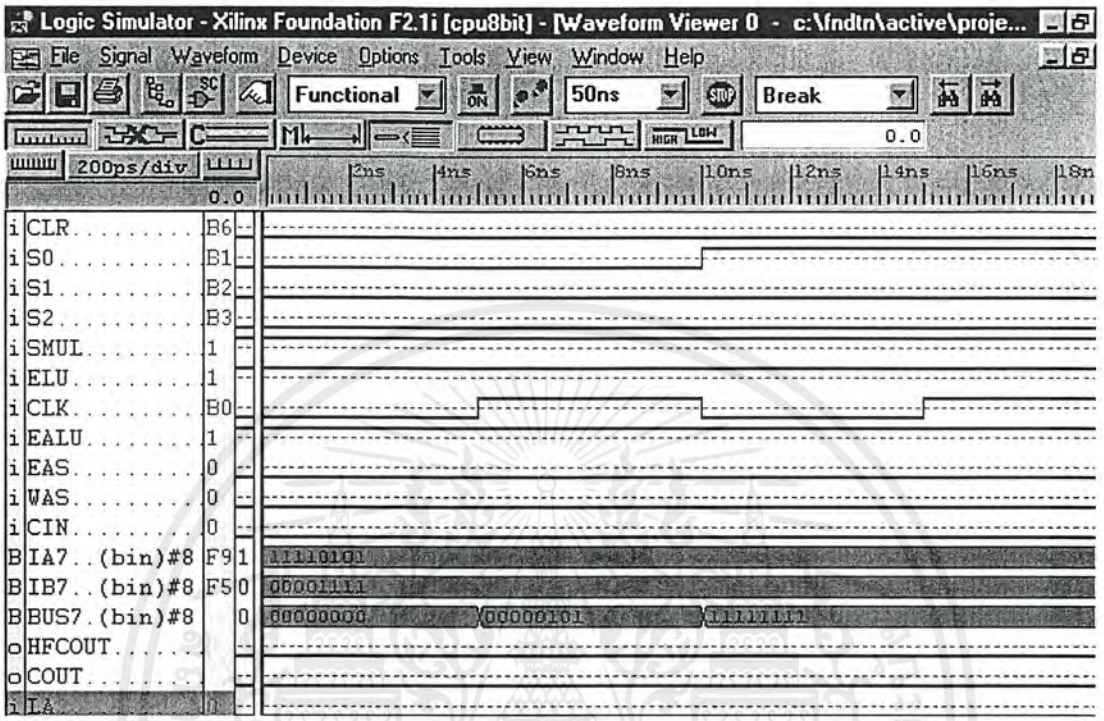
รูปที่ 4.3 นำโครงสร้างภายในของ Program Counter ทำเป็น macro



รูปที่ 4.4 ผลการ Simulate Program Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

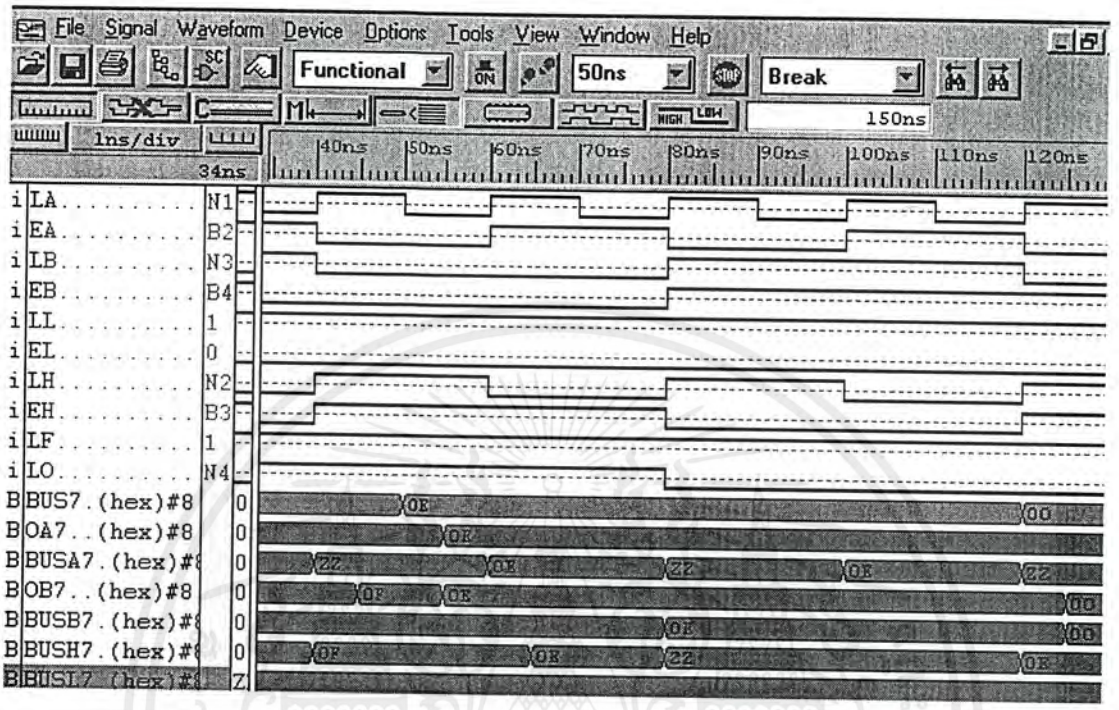
4.2.2 ผลการ Simulate คำสั่ง AND และ OR



รูปที่ 4.5 ผลการ Simulate คำสั่ง AND และ OR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

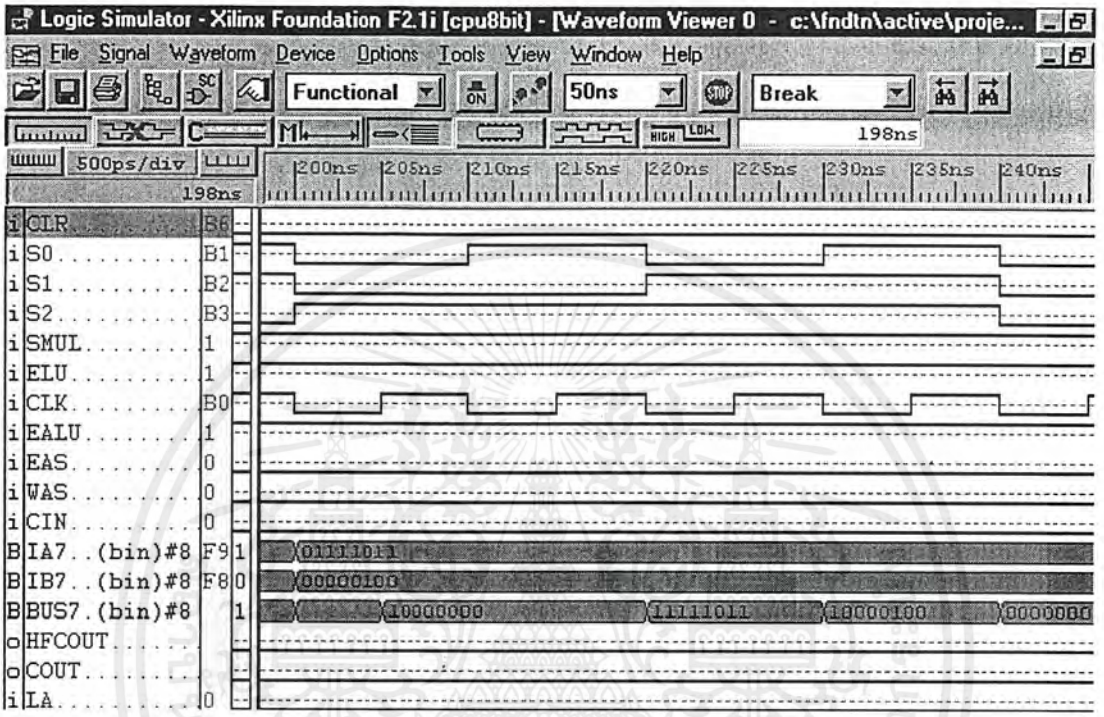
4.2.3 ผลการ Simulate คำสั่ง LOAD



รูปที่ 4.6 ผลการ Simulate คำสั่ง LOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

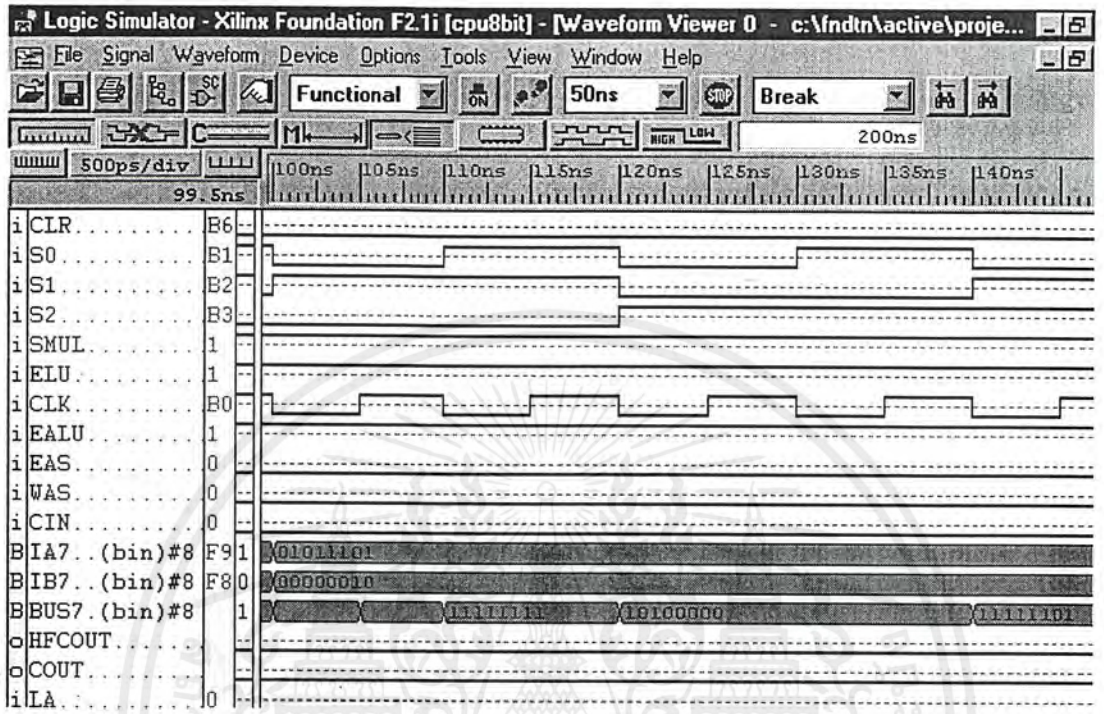
4.2.4 ผลการ Simulate คำสั่ง Complement register A และ B



รูปที่ 4.7 ผลการ Simulate คำสั่ง Complement register A และ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

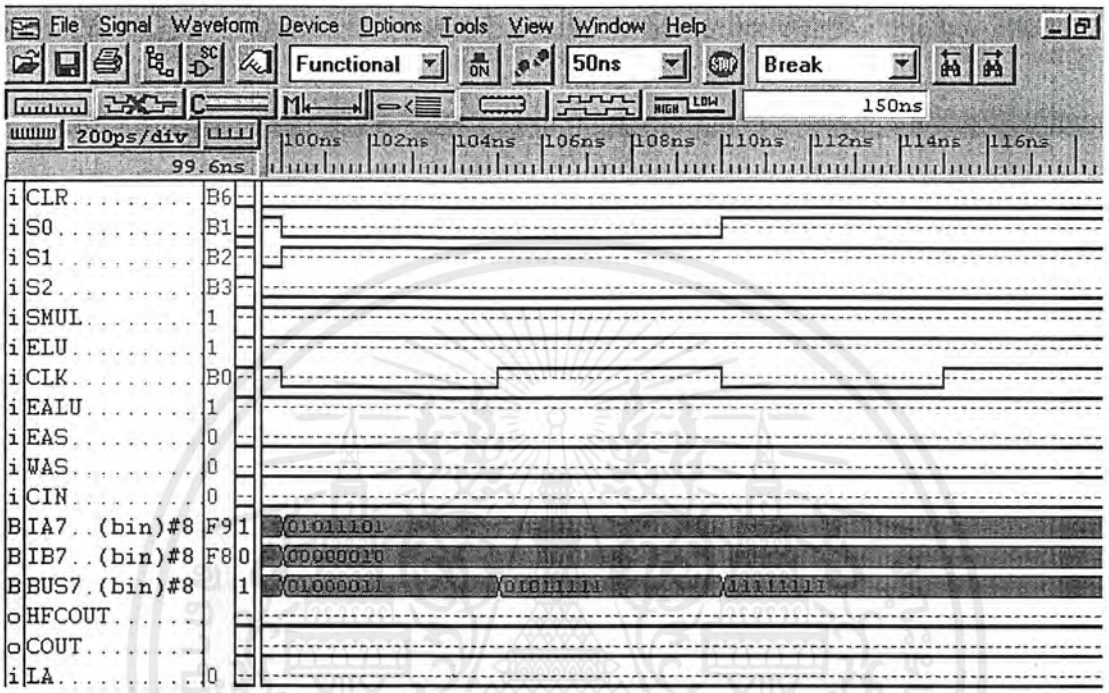
4.2.5 ผลการ Simulate คำสั่ง NOR และ XNOR



รูปที่ 4.8 ผลการ Simulate คำสั่ง NOR และ XNOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 ผลการ Simulate คำสั่ง XOR และ NAND

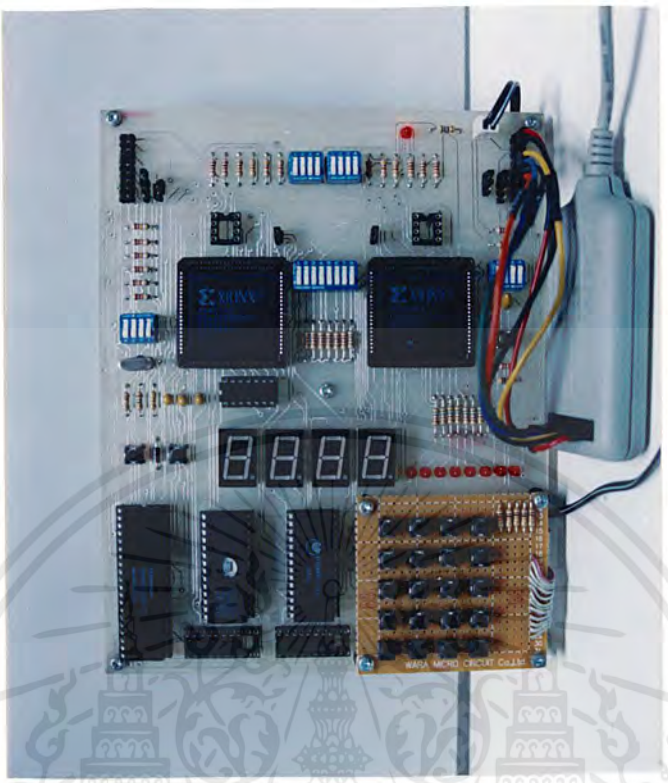


รูปที่ 4.9 ผลการ Simulate คำสั่ง XOR และ NAND

4.3 การเชื่อมต่อบอร์ดทดลองกับสายดาวน์โหลด

การใช้งานบอร์ดทดลองที่ได้ทำการออกแบบนั้นจะต้องทำการต่อสายดาวน์โหลดเพื่อควบคุมหน้าที่การทำงานของอุปกรณ์ FPGAs แต่ละตัวว่าจะทำงานในโหมดใด ซึ่งต้องมาจากการตั้งค่าดิพสวิตซ์ หลังจากการตั้งค่าดิพสวิตซ์แล้วและทำการเชื่อมต่อสายดาวน์โหลดเรียบร้อยแล้ว ก็ทำการทดลองที่บอร์ดทดลองได้ การต่อสายดาวน์โหลดทำได้ดังแสดงไว้ในรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 การต่อสายดาวน์โหลดกับบอร์ดทดลอง

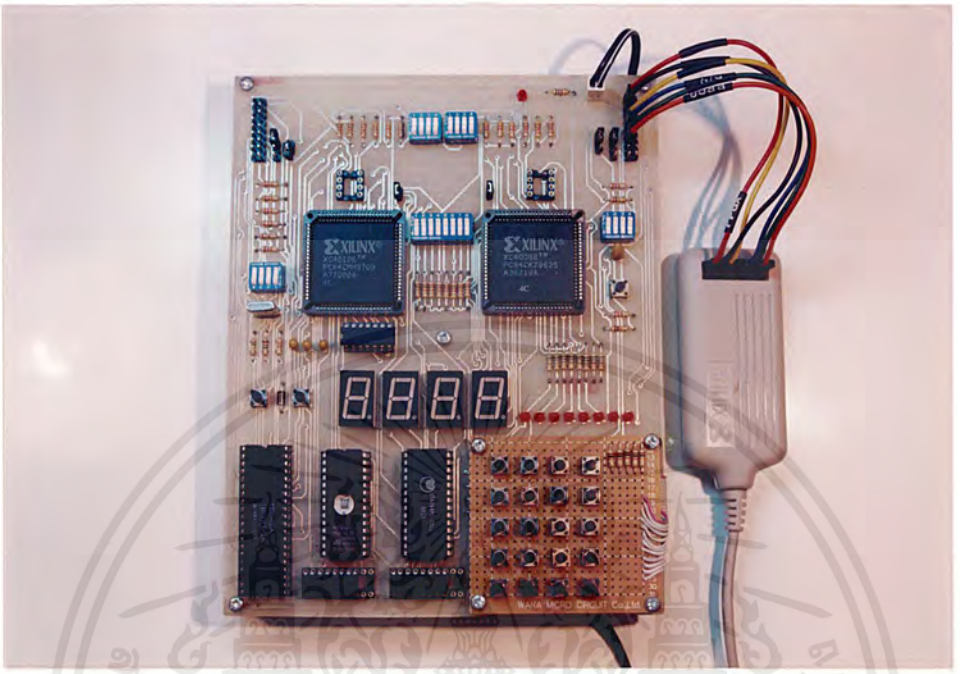
4.4 ผลการทดสอบบอร์ดทดลอง

หลังจากที่ได้ทำการประกอบวงจรตามแบบที่ได้วางไว้เสร็จแล้ว ดังรูปที่ 4.1 ก็ได้นำบอร์ดทดลองมาทำการทดลองโดยการทดสอบว่าลายวงจรเกิดการช๊อตกันหรือไม่ โดยการใช้มัลติมิเตอร์วัดค่าความต้านทาน ถ้าตรวจสอบแล้วลายวงจรไม่ช๊อตกันก็จะทำการทดสอบอุปกรณ์ทีละตัว

4.4.1 แอลอีดี

การทดสอบการทำงานของแอลอีดีนั้น ทำได้โดยการดาวน์โหลดตัวอย่างโปรแกรมที่มีการใช้ดิพสวิตช์ทดสอบ 8 ตัวเป็นอินพุตป้อนให้กับวงจร แสดงผลออกทางแอลอีดีผลที่ได้เมื่อทำการดาวน์โหลดดังแสดงไว้ในรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

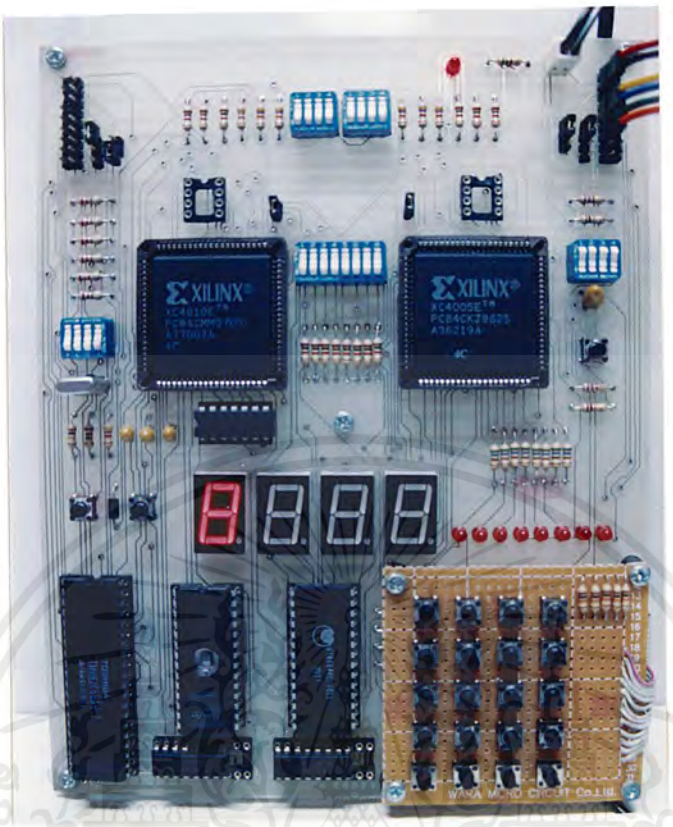


รูปที่ 4.11 ผลการดาวน์โหลดโปรแกรมเพื่อทดสอบแอลอีดี

4.4.2 หน่วยแสดงผลเจ็ดส่วน

การทดสอบการทำงานของหน่วยประมวลผลเจ็ดส่วนนั้น ทดสอบได้โดยทำการดาวน์โหลดตัวอย่างโปรแกรมที่มีการใช้ดิพสวิตซ์ทดสอบ 4 ตัวเป็นอินพุตป้อนให้แก่วงจร แสดงผลออกทางหน่วยแสดงผลเจ็ดส่วน ผลที่ได้เมื่อทำการดาวน์โหลดดังแสดงไว้ในรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

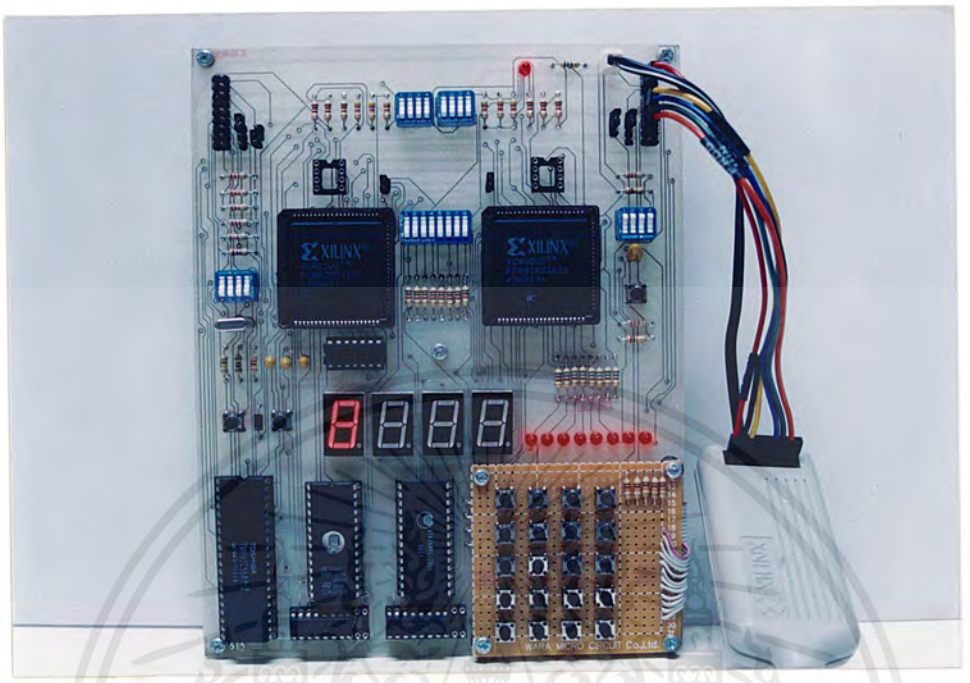


รูปที่ 4.12 ผลการดาวน์โหลดโปรแกรมเพื่อทดสอบหน่วยแสดงผลเจ็ดส่วน

4.4.3 ทดสอบการทำงานของอุปกรณ์ FPGAs เบอร์ 4010E และเบอร์ 4005E

ทำการดาวน์โหลดโปรแกรมลงอุปกร์ FPGAs เบอร์ 4005E ซึ่งได้ทำการต่อวงจรกับภาคแสดงผลและอุปกรณ์อินพุตคือ สวิตซ์ ดังนั้นทดสอบได้โดยในระหว่างทำการดาวน์โหลดลงนั้นให้ดูโปรแกรมตอนดาวน์โหลดว่าผ่านหรือไม่ ถ้ามีข้อผิดพลาดก็ให้ทำการแก้ไขที่โปรแกรมทดลองซ้ำ แต่ถ้าไม่มีข้อผิดพลาดใดๆ ก็ให้มาตรวจสอบที่ผลการทำงานของ FPGAs โดยที่เบอร์ 4005E สามารถตรวจสอบจากการโปรแกรมให้แสดงผลที่แอลอีดีหรือหน่วยแสดงผลเจ็ดส่วนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผลทดสอบการทำงานของอุปกรณ์ FPGAs เบอร์ 4010E และเบอร์ 4005E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 สรุป

หน่วยประมวลผลกลางขนาด 8 บิต โดยเลือกใช้โปรแกรม Xilinx Foundation Series ออกแบบให้อยู่ในรูปของ Schematic โดยการออกแบบจะมองภาพรวมของงานที่เราต้องการออกแบบว่าจะมีลักษณะใด แล้วทำการแบ่งเป็นส่วนๆ จากนั้นจึงเริ่มทำการสร้างในแต่ละส่วนของหน่วยประมวลผลกลางขนาด 8 บิต ทำให้ง่ายต่อความเข้าใจและการแก้ไข เมื่อทำการออกแบบให้อยู่ในรูป Schematic ทุกส่วนของหน่วยประมวลผลกลางขนาด 8 บิต เรียบร้อยแล้วขั้นต่อมาก็คือการนำโปรแกรมไปจำลองการทำงานเพื่อดูการทำงานของหน่วยประมวลผลกลางขนาด 8 บิตที่ได้ทำการออกแบบว่ามีการทำงานเป็นไปตามวัตถุประสงค์ที่ต้องการหรือไม่ ถ้าผลการจำลองการทำงานถูกต้องเป็นไปตามวัตถุประสงค์ที่ต้องการแล้ว จึงนำโปรแกรมไปแปลงเป็นไฟล์บิตสตรีมและทำการดาวน์โหลดลงบนบอร์ดตัวอย่างของ FPGAs ผ่านทางดาวน์โหลดเคเบิล

จากการทดสอบการทำงานของหน่วยประมวลผลกลางขนาด 8 บิต ปรากฏว่าหน่วยประมวลผลกลางมีความสามารถในการทำงานได้ในขอบเขตดังนี้คือ

1) ใช้สัญญาณนาฬิกาขนาด 2 MHz เป็นฐานเวลาในการทำงาน
2) สามารถอ้างถึงตำแหน่งหน่วยความจำได้ 65,536 ตำแหน่ง (64 Kbyte)
3) สามารถรับข้อมูลได้ 8 บิต
4) มีกลุ่มคำสั่งในการโอนย้ายข้อมูลได้แก่ LOAD
5) มีกลุ่มคำสั่งทางคณิตศาสตร์และลอจิก ได้แก่ ADD, ADC ,SUB , AND , OR ฯลฯ
6) มีกลุ่มคำสั่งการกระโดด ได้แก่ JP
7) มีกลุ่มคำสั่งรับส่งข้อมูลกับอุปกรณ์ภายนอก ได้แก่ IN , OUT
8) มีรีจิสเตอร์ใช้งานขนาด 8 บิต ได้แก่ A , B , C
9) มีรีจิสเตอร์ใช้งานขนาด 16 บิต ได้แก่ PC,HL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหา

ในการจัดทำโครงงานนี้ สามารถสรุปปัญหาที่เกิดขึ้นระหว่างทำโครงงานได้ดังนี้

1) ในการออกแบบให้อยู่ในรูป Schematic ในการเชื่อมต่อบัส เมื่อทำการ Simulate สัญญาณบัสจะไม่ทราบสถานะของตัวมันเอง

2) ในการออกแบบ Program Counter โดยใช้ JK Flip-Flop มีปัญหาคือไม่สามารถป้อนอินพุตให้กับ Program Counter ได้

3) การออกแบบในส่วนของสัญญาณควบคุม โดยการใช้ ภาษา VHDL เขียนเกิดความซับซ้อนของโปรแกรมในการกำหนดเงื่อนไขมาก

4) ในการออกแบบส่วนของบอร์ดทดลอง ออกแบบโดยใช้โปรแกรม Protel 99 SE มีปัญหาคือ ไม่มี Library ในส่วนของ Xilinx ตระกูล XC4000E

5) ในการออกแบบแผ่นวงจร PCB ประสบปัญหาเช่นเดียวกับการออกแบบบอร์ดทดลองคือไม่มี Library ของตัว Xilinx ตระกูล XC4000E เช่นเดียวกัน

6) สำหรับระบบบัสของบอร์ดทดลอง เกิดข้อผิดพลาดคือ ไม่ทราบสถานะของบัสว่าอยู่ในสถานะใด

7) สำหรับบอร์ดที่ทำการออกแบบเสร็จแล้วทำการตรวจสอบ แสดงว่าสวิตซ์ที่ใช้รับข้อมูล ยังขาดอีกหลายตัว

5.3 แนวทางแก้ไขปัญหา

1) ศึกษาการเชื่อมต่อบัสโดยศึกษาจาก Help และ คู่มือการใช้งานโปรแกรม Xilinx Foundation Series มีหลักการเชื่อมต่อบัสคือ บัสแต่ละเส้นจะต้องเป็นเส้นเดียวกันคือจะต้องไม่มีรอยต่อ จากนั้นต้องต่อ INBUF เข้าที่อินพุตของแต่ละภาค และ OUTBUF ที่เอาต์พุตของแต่ละภาคเช่นเดียวกัน เพื่อป้องกันสัญญาณจากบัสไปรบกวนอินพุตของภาคอื่น

2) ใช้ Symbol ที่มีอยู่ใน Library ของ Xilinx มาประยุกต์ใช้แทน

3) ออกแบบโดยใช้ Schematic ในการออกแบบแต่จะต้องใช้ขาสัญญาณควบคุมเป็นจำนวนมาก

4) ทำการเขียน Library ของ Xilinx ตระกูล XC4000E ขึ้นมาเอง

5) ทำการเขียน Library ของ Xilinx ตระกูล XC4000E ขึ้นมาเอง

6) ต้องกำหนดบัสให้เป็นเส้นเดียวกัน โดยไม่มีจุดต่อใดๆ

7) ทำการเพิ่มสวิตซ์โดยต้องแก้ไขบอร์ดในส่วนอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางการพัฒนาต่อ

1) หน่วยประมวลผลกลางที่ได้ทำการออกแบบสามารถเพิ่มจำนวนคำสั่งให้มากขึ้นได้ หรืออาจจะเพิ่มเป็น 16 บิตโดยใช้เทคโนโลยีการออกแบบ ใหม่ๆ ลงไปก็ได้ โดยใช้อุปกรณ์ FPGAs ที่มีจำนวนเกตภายในเพิ่มมากขึ้น

2) สามารถใช้การออกแบบให้อยู่ในรูป Schematic ออกแบบวงจรที่ใช้เฉพาะงานลงบนอุปกรณ์ FPGAs โดยเฉพาะวงจรที่สร้างขึ้นเองไม่ต้องลอกเลียน

3) สามารถใช้ Schematic และอุปกรณ์ FPGAs ออกแบบวงจรไปประยุกต์ใช้งานตามความสามารถและความต้องการของผู้ออกแบบแต่ละคนได้อย่างกว้างขวาง

4) ส่วนของบอร์ดทดลอง สามารถปรับปรุงให้มีลักษณะการใช้งานที่ดูมั่นคงแข็งแรงและสวยงามมากขึ้น และสามารถเพิ่มเติมอุปกรณ์ต่างๆ เข้าไปได้เพื่อใช้ประโยชน์ในการทดลอง

5) สามารถสร้างและประยุกต์ใช้งานบอร์ดให้สามารถต่อร่วมกับอุปกรณ์ภายนอก เพื่อใช้งานร่วมกัน โดยทำการสร้างพอร์ตอินพุตและเอาต์พุตเพิ่มขึ้น

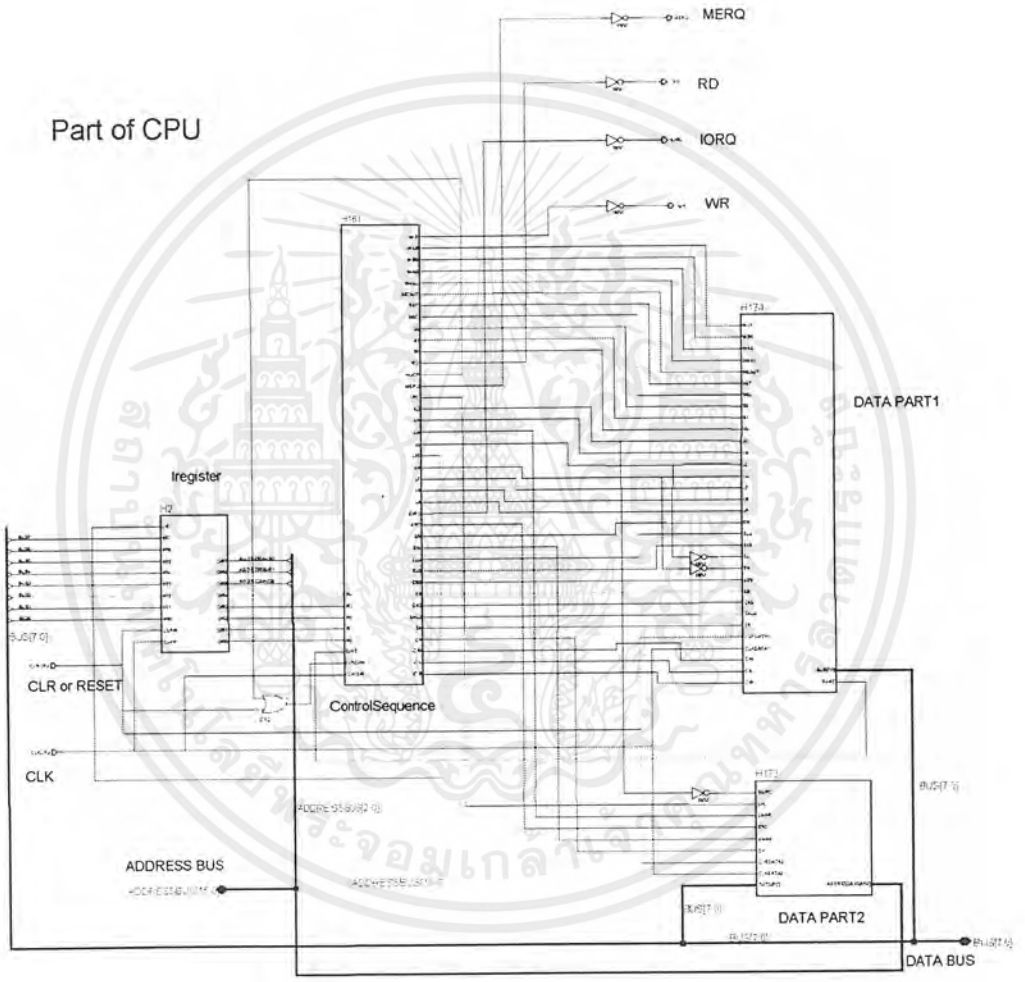
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
รูป Schematic ของหน่วยประมวลผลกลางขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป Schematic ของหน่วยประมวลผลกลางขนาด 8 บิต

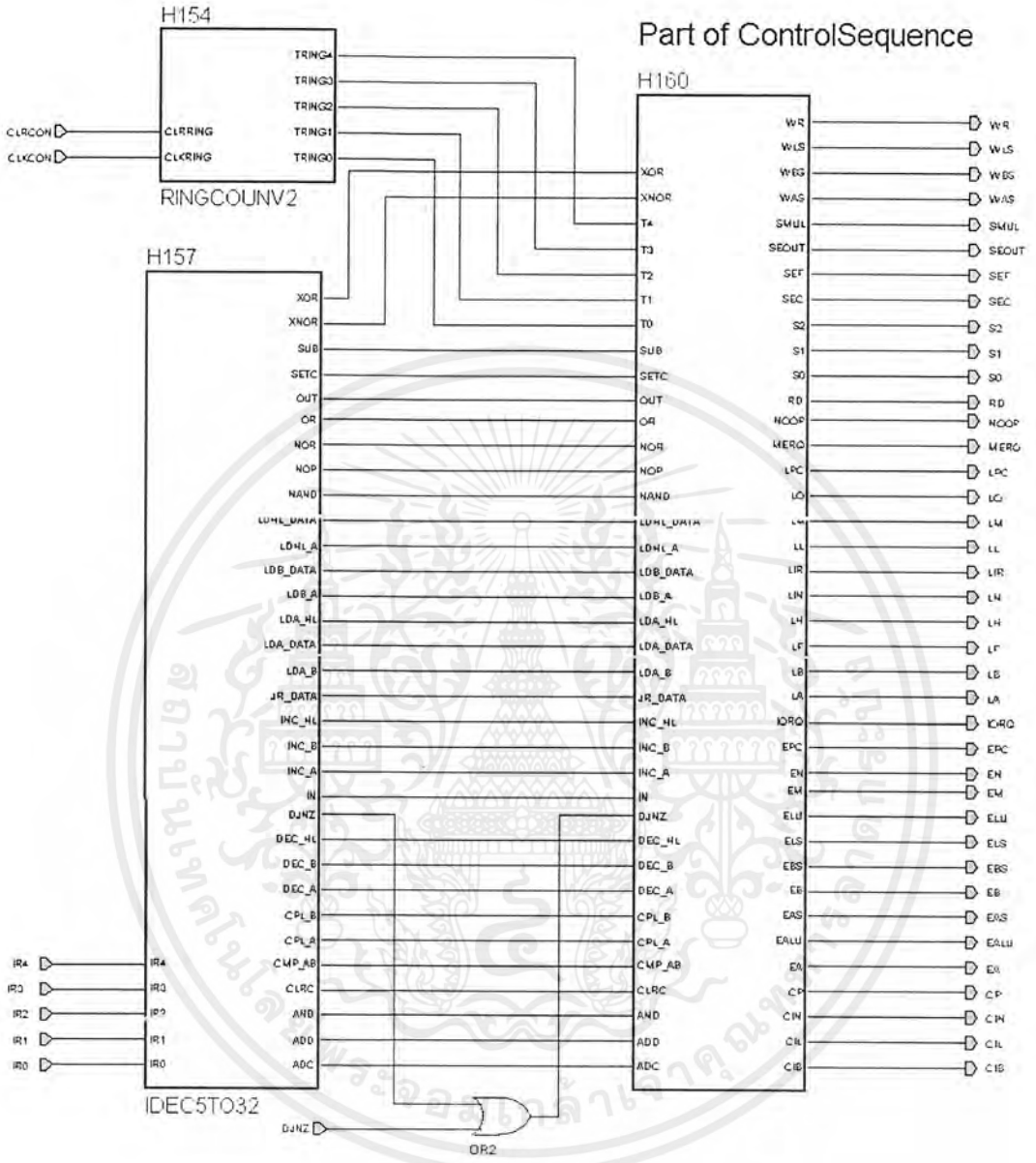
ทำการออกแบบตัวประมวลผลกลางขนาด 8 บิตให้อยู่ในรูป Schematic โดยออกแบบแต่ละส่วนของตัวประมวลผลแล้วทำการ Simulate เมื่อได้ผลตามที่ต้องการแล้วจึงนำมาเชื่อมต่อกัน ซึ่งจะได้ตัวประมวลผลกลางขนาด 8 บิต ดังรูปที่ ก.1



รูปที่ ก.1 ส่วนของตัวประมวลผลกลางขนาด 8 บิต

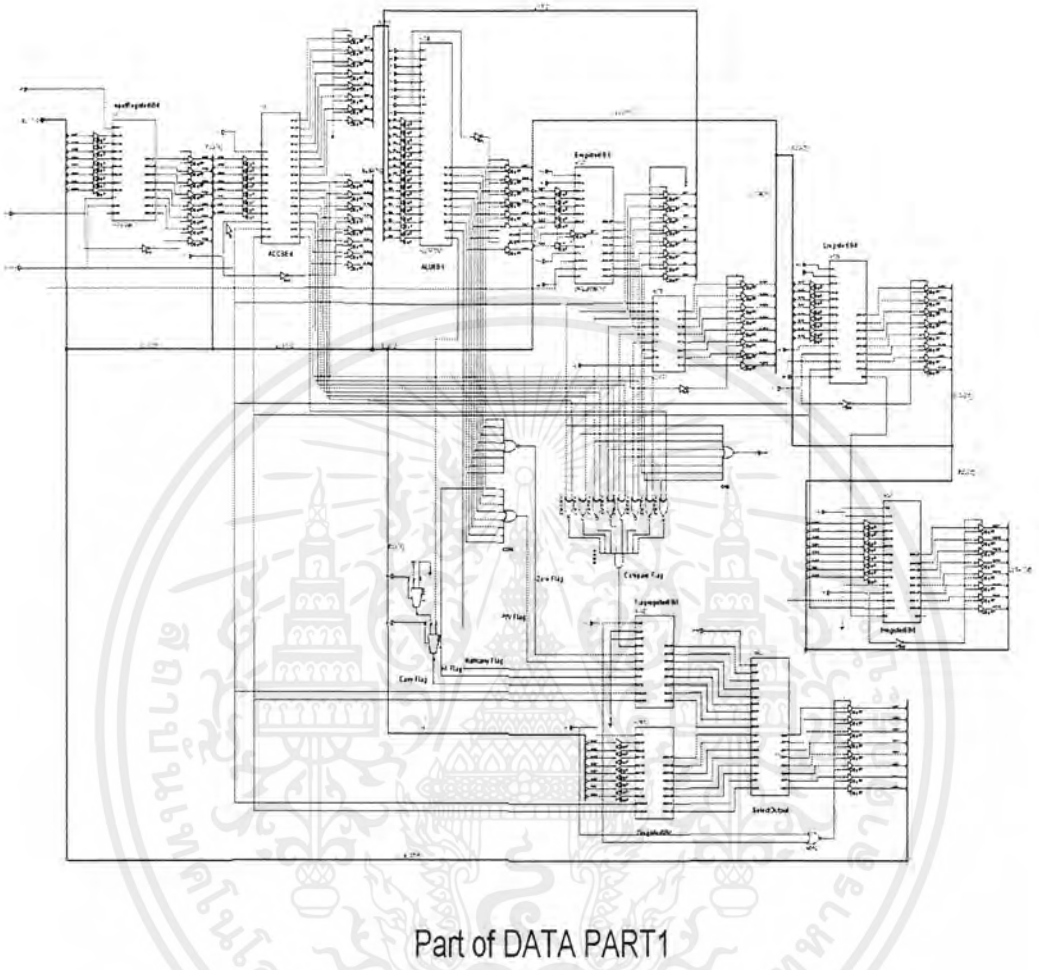
ซึ่งแต่ละส่วนของตัวประมวลผลขนาด 8 บิตนี้จะประกอบด้วยส่วนสำคัญได้ 2 ส่วนคือ ส่วนควบคุม(Control) และส่วนของข้อมูล(Data) ซึ่งแสดงดังรูปที่ ก.2 และรูปที่ ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 ส่วนของการควบคุมตัวประมวลผลขนาด 8 บิต

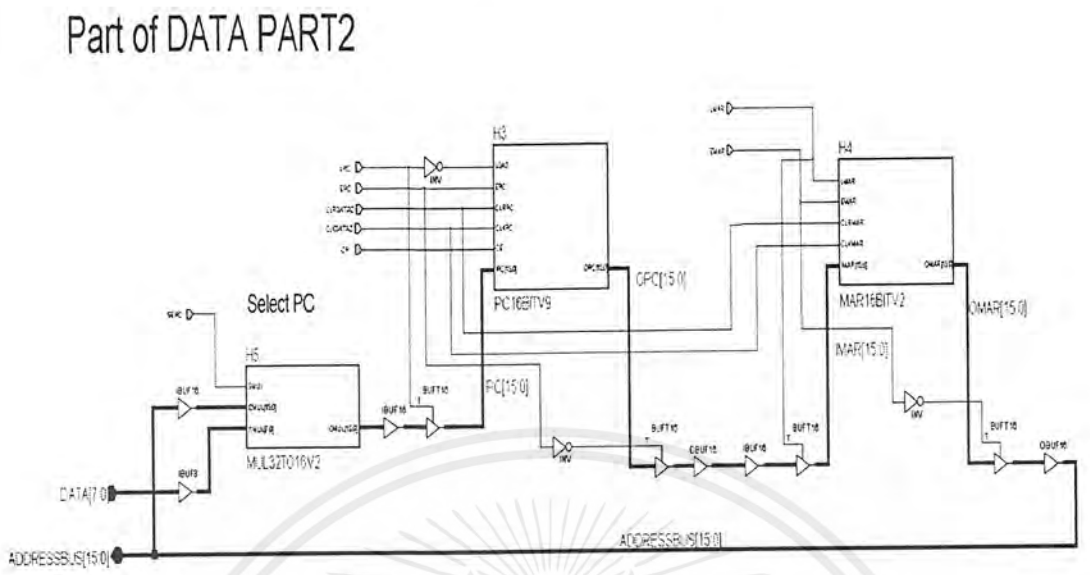
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 ส่วน Data Part 1 ของตัวประมวลผลขนาด 8 บิต

ส่วนของ Data Part เป็นส่วนที่ประกอบด้วยส่วนของรีจิสเตอร์ที่ใช้งานใน ตัวประมวลผลกลางขนาด 8 บิต ได้แก่ รีจิสเตอร์ A, B, H, L รวมถึง Flag Register, Input and Output Register นอกจากนี้ยังประกอบไปด้วย หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 ส่วน Data Part2 ของตัวประมวลผลขนาด 8 บิต

ส่วนของ Data Part2 ของตัวประมวลผลกลางขนาด 8 บิต จะประกอบไปด้วยส่วนสำคัญ 2 ส่วนคือ โปรแกรมเคาน์เตอร์(Program Counter) และ MAR (Memory Address Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

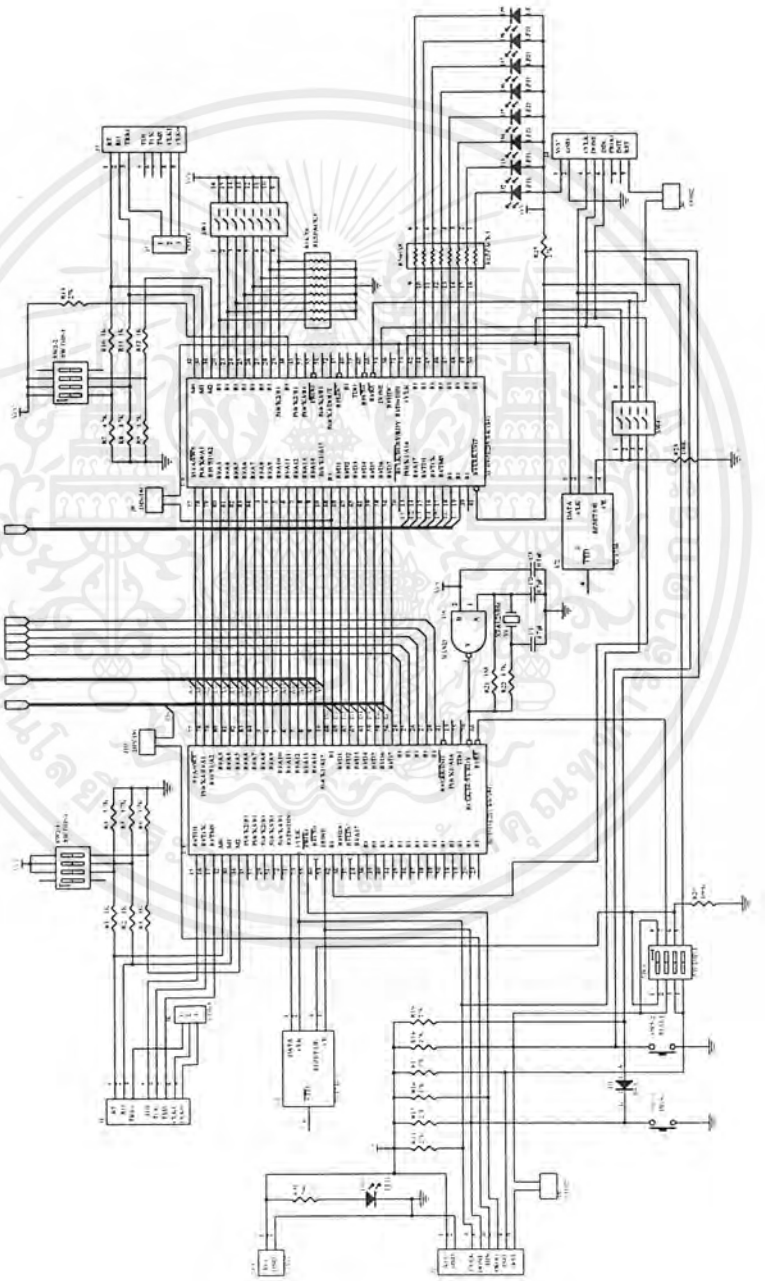
ภาคผนวก ข
วงจร แผ่นวงจรพิมพ์และรายการของอุปกรณ์บนบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบชุดทดลอง

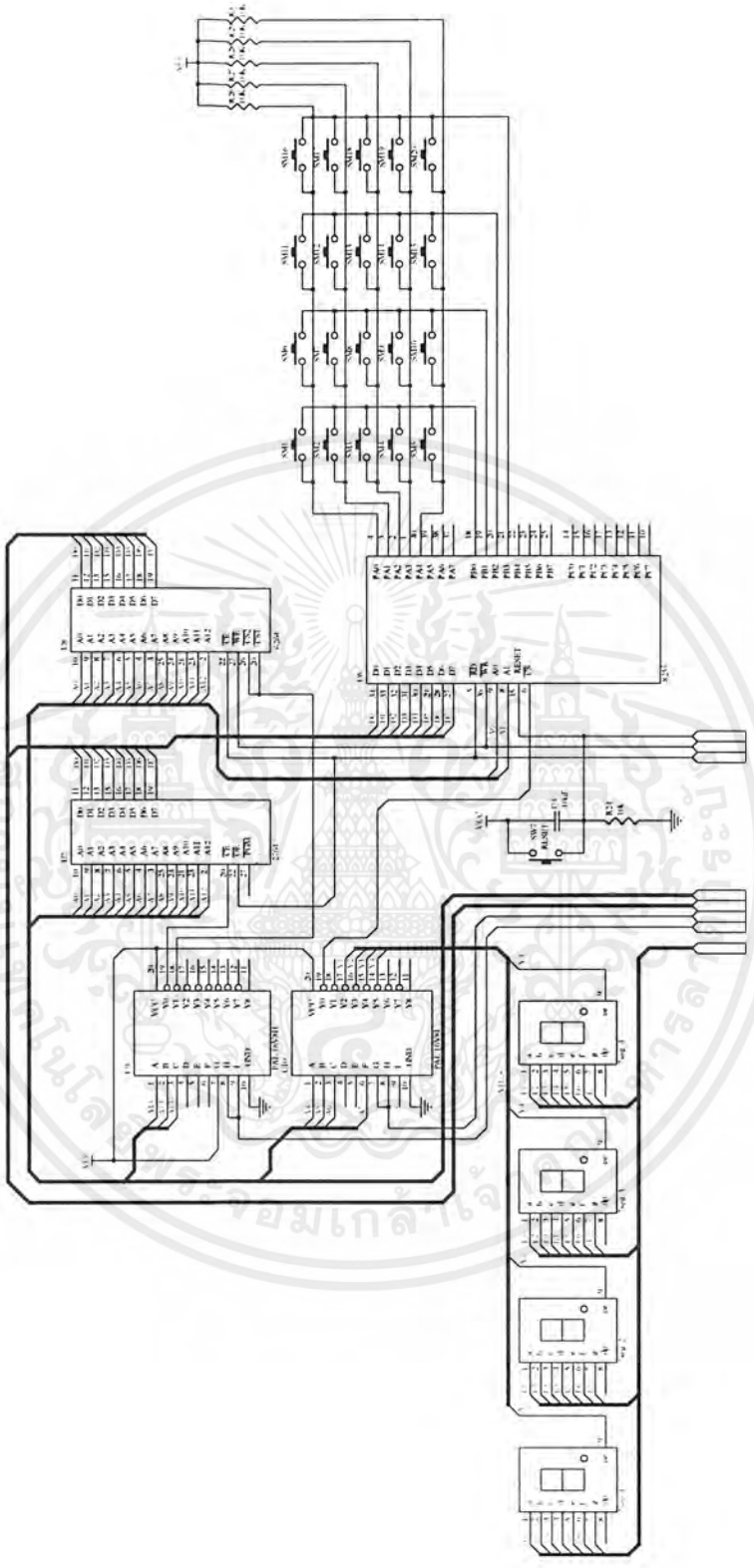
การออกแบบวงจรชุดทดลองได้ทำการออกแบบโดยใช้โปรแกรม Protel 99 SE ภายในวงจรชุดทดลอง จะแบ่งออกเป็น 2 ส่วนคือ

1. ส่วนของหน่วยประมวลผลกลาง และหน่วยความจำ ซึ่งแสดงดังรูปที่ ข.1
2. ส่วนของอุปกรณ์อินพุตและเอาต์พุต ซึ่งแสดงดังรูปที่ ข.2



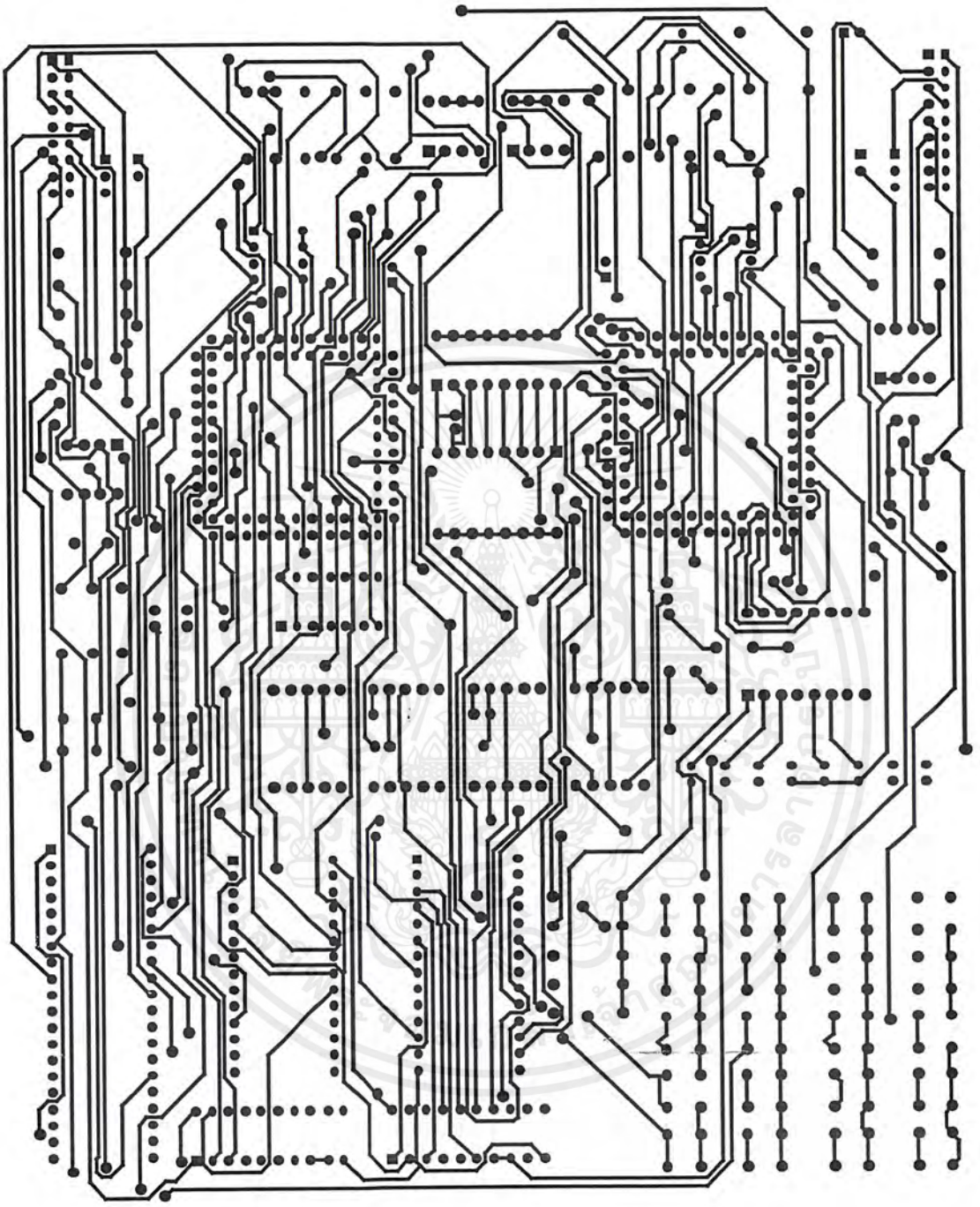
รูปที่ ข.1 วงจรในส่วนของหน่วยประมวลผลกลาง และหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



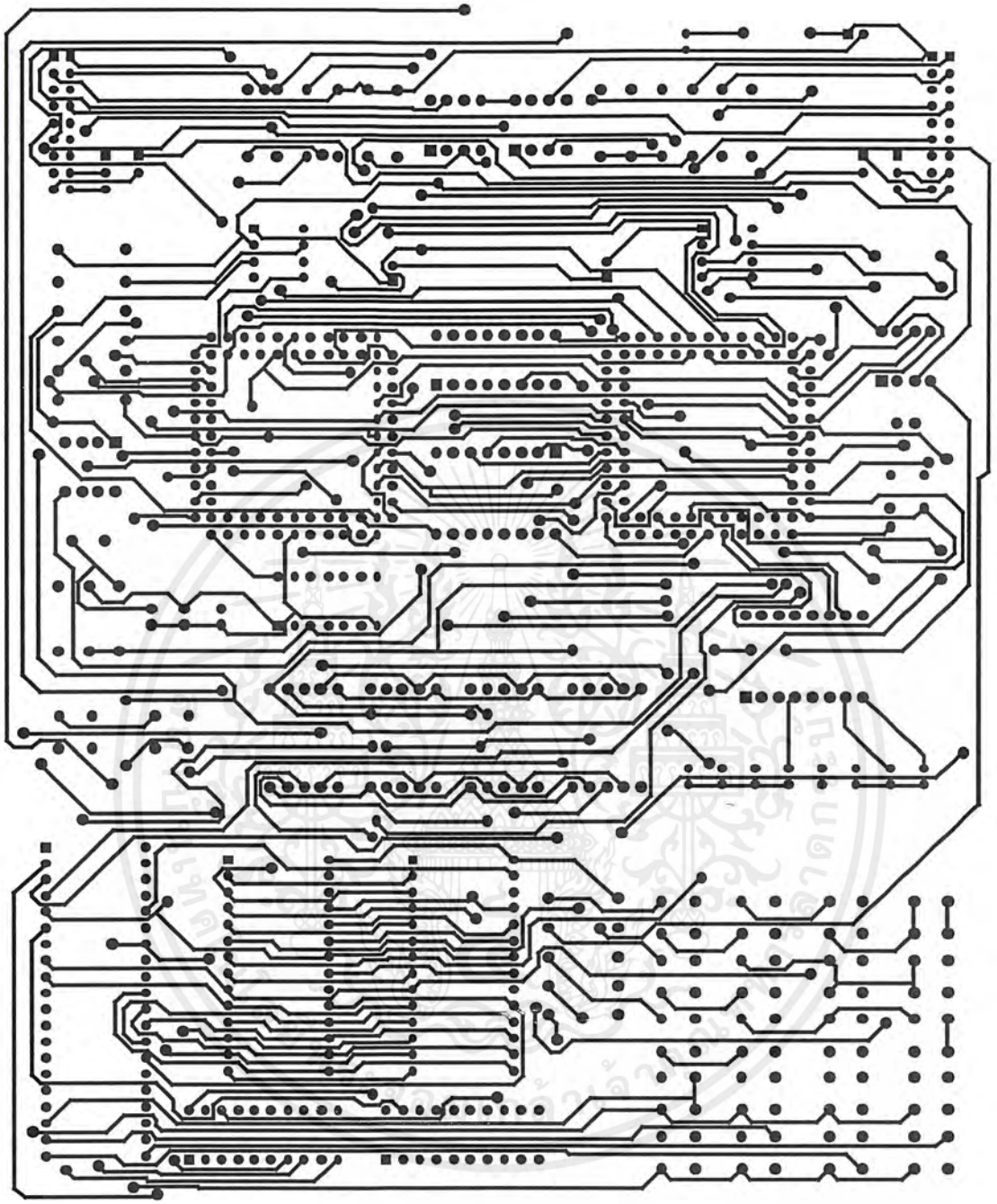
รูปที่ ข.2 วงจรในส่วนของอุปกรณ์อินพุตและเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



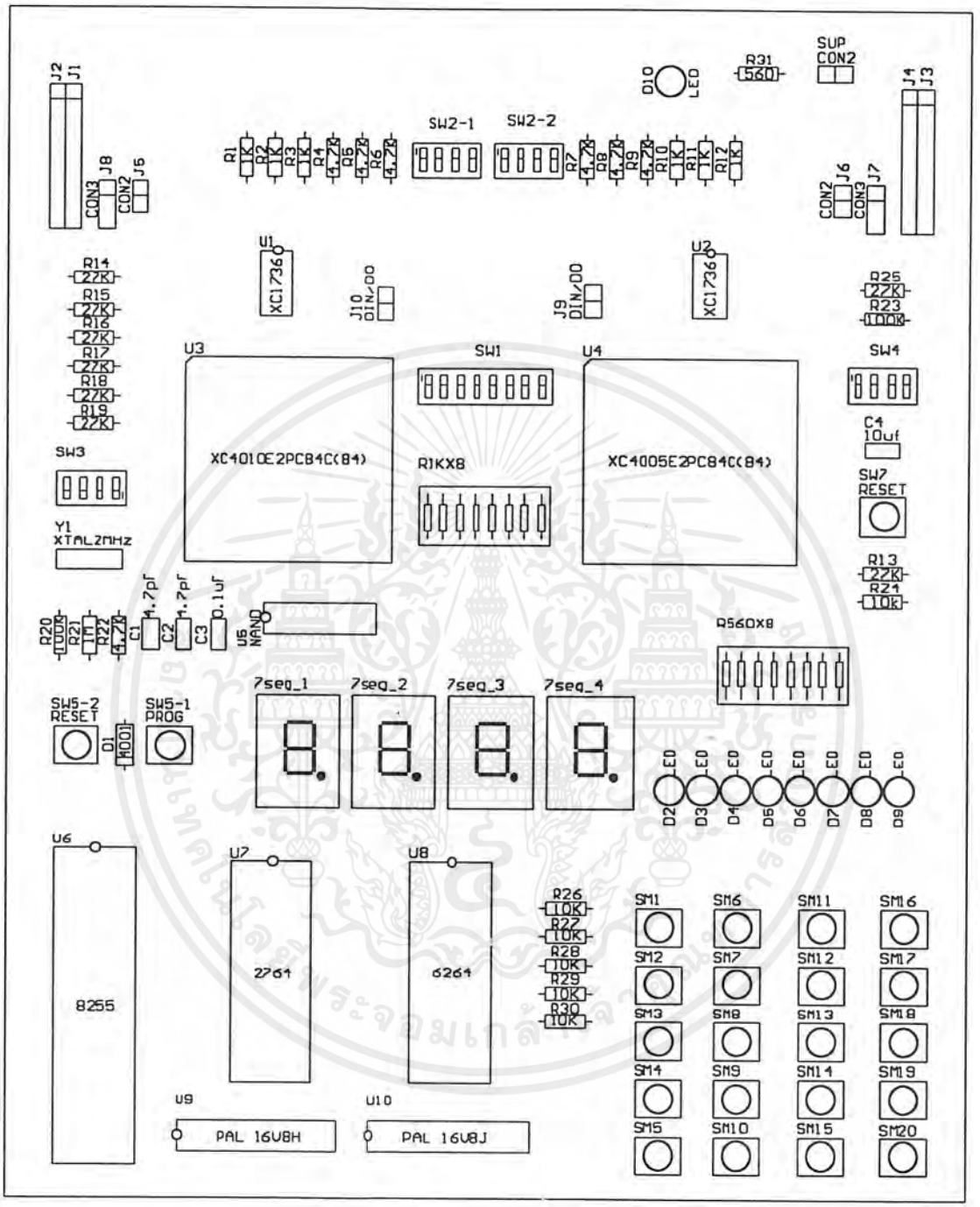
รูปที่ ข.3 แผ่นวงจรพิมพ์ด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 แผ่นวงจรพิมพ์ด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.5 ลักษณะการวางอุปกรณ์บนแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์บนบอร์ด

1. XC4010E	1	ตัว
2. XC4005E	1	ตัว
3. XC1736	2	ตัว
4. PAL16V8H	2	ตัว
5. IC เบอร์ 8255	1	ตัว
6. IC เบอร์ 2764	1	ตัว
7. IC เบอร์ 6264	1	ตัว
8. IC เบอร์ 4011	1	ตัว
9. 7 – Segment	4	ตัว
10. XTAL 2 MHz	1	ตัว
11. LED สีแดง	9	ตัว
12. ตัวต้านทาน R 560Ω	9	ตัว
ตัวต้านทาน R 1kΩ	14	ตัว
ตัวต้านทาน R 4.7kΩ	7	ตัว
ตัวต้านทาน R 10kΩ	6	ตัว
ตัวต้านทาน R 27kΩ	8	ตัว
ตัวต้านทาน R 100kΩ	2	ตัว
ตัวต้านทาน R 1MΩ	1	ตัว
13. ตัวเก็บประจุ C 4.7pF	2	ตัว
ตัวเก็บประจุ C 0.1μF	1	ตัว
ตัวเก็บประจุ C 10μF	1	ตัว
14. Diode 1N4001	1	ตัว
15. DIPSW. 8	1	ตัว
DIPSW . 4	4	ตัว
16. PBSW.(SW.กด)	23	ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1
คำสั่งโอนย้ายข้อมูล 1

จุดประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งโอนย้ายข้อมูลขนาด 8 บิตได้
2. สามารถเขียนโปรแกรมการโอนย้ายข้อมูลขนาด 8 บิตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

กลุ่มการโอนย้ายข้อมูลขนาด 8 บิต (8 bit load group)

กลุ่มคำสั่งนี้เป็นการโอนย้ายข้อมูลขนาด 8 บิต เข้าไปยังรีจิสเตอร์ภายในตัวซีพียู โดยที่ข้อมูลอาจออกมาจากรีจิสเตอร์ภายในตัวซีพียูเอง มาจากคำสั่งทันทีทันใด หรือมาจากหน่วยความจำภายนอก ในทุกกรณีรีจิสเตอร์ต้นทาง (source) จะไม่เปลี่ยนแปลงหลังจากที่ส่งถ่ายข้อมูลไปแล้ว ลักษณะคำสั่งต่างๆ อาจแยกออกเป็นกลุ่มย่อยๆได้ดังนี้

กลุ่มคำสั่ง LD r,s คือคำสั่งที่ทำการโอนย้ายข้อมูลเข้ามายังรีจิสเตอร์ที่กำหนดด้วยข้อมูลที่มากจากรีจิสเตอร์ต่างๆที่อยู่ภายในซีพียู เช่น LD r,r' ข้อมูลที่ได้จากคำสั่งทันที เช่น LD r,n หรือข้อมูลจากหน่วยความจำ เช่น LD r,(HL) เป็นต้น หลังจากที่กระทำคำสั่งเหล่านี้แล้วแฟลกต่างๆ จะไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมตามที่กำหนดให้ บันทึกผลที่ได้

1

Address	**Op-code**	**Mnemonic**
0000	01 01	LD A,01H
0002	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

2

Address	**Op-code**	**Mnemonic**
0000	02 02	LD B,02H
0002	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

3

Address	**Op-code**	**Mnemonic**
0000	02 01	LD B,01H
0002	04	LD A,B
0003	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

4.

Address	**Op-code**	**Mnemonic**
0000	01 05	LD A,05H
0002	05	LD B,A
0003	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. คำสั่งการโอนย้ายข้อมูลขนาด 8 บิตหมายถึงอะไร?

2. คำสั่ง LD r,r' แตกต่างกับคำสั่ง LD r,n หรือไม่ ถ้าแตกต่าง ต่างกันอย่างไร?

3. จงเขียนโปรแกรมโอนย้ายข้อมูล 0FH ไปเก็บไว้ในรีจิสเตอร์ A

Addres	Op-code	Mnemonic
_____	_____	_____
_____	_____	_____
_____	_____	_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

4. จงเขียนโปรแกรมโอนย้ายข้อมูล 0AH ไปเก็บไว้ในรีจิสเตอร์ B จากนั้นนำข้อมูลในรีจิสเตอร์ B ไปเก็บไว้ในรีจิสเตอร์ A

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2
คำสั่งโอนย้ายข้อมูล 2

จุดประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งโอนย้ายข้อมูลขนาด 16 บิตได้
2. สามารถเขียนโปรแกรมการโอนย้ายข้อมูลขนาด 16 บิตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

กลุ่มการโอนย้ายข้อมูลขนาด 16 บิต (16 bit load group)

กลุ่มคำสั่งนี้เป็นการโอนย้ายข้อมูลขนาด 16 บิต เข้าไปยังรีจิสเตอร์ภายในตัวซีพียู โดยที่ข้อมูลมาจากมาจากคำสั่งทันทีทันใด ได้แก่คำสั่ง LD HL,n n โดยส่วนมากใช้เก็บค่าแอดเดรสต่างๆ หรือใช้เก็บแอดเดรสของหน่วยความจำภายนอก ในทุกกรณีรีจิสเตอร์ต้นทาง (source) จะไม่เปลี่ยนแปลงหลังจากที่ส่งถ่ายข้อมูลไปแล้ว

กลุ่มคำสั่ง LD r,s การทำงานจะตรงข้ามกับกลุ่มแรก คือ ข้อมูลในรีจิสเตอร์ของซีพียูจะถูกส่งผ่านมายังหน่วยความจำ โดยวิธีการอ้างถึงหน่วยความจำแบบตรงหรือใช้คู่รีจิสเตอร์ HL คือคำสั่ง LD (HL),r คือใช้ข้อมูลในหลังจากที่กระทำคำสั่งเหล่านี้แล้วแฟลกต่างๆ จะไม่มีการเปลี่ยนแปลงคู่รีจิสเตอร์ HL เพื่อชี้ตำแหน่งหน่วยหน่วยความจำปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมตามที่กำหนดให้ บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	03 00 01	LD HL,0100H
0003	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

2.

Address	Op-code	Mnemonic
0000	03 00 01	LD HL,0100H
0003	06	LD A,(HL)
0004	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

ก่อนกระทำคำสั่งค่าตำแหน่ง 0100H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

หลังกระทำคำสั่งค่าตำแหน่ง 0100H = _____

3.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	03 00 01	LD HL,0100H
0005	07	LD (HL),A
0006	00	NOP

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

ก่อนกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

ก่อนกระทำคำสั่งค่าตำแหน่ง 0100H = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ H = _____

หลังกระทำคำสั่งค่าในรีจิสเตอร์ L = _____

หลังกระทำคำสั่งค่าตำแหน่ง 0100H = _____

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. คำสั่งการโอนย้ายข้อมูลขนาด 16 บิตหมายถึงอะไร?

2. คำสั่ง LD HL,nn แตกต่างกับคำสั่ง LD (HL),A หรือไม่ ถ้าแตกต่าง ต่างกันอย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จงเขียนโปรแกรมโอนย้ายข้อมูล 02H ไปเก็บไว้ในรีจิสเตอร์ B จากนั้นนำข้อมูลในรีจิสเตอร์ B ไปเก็บไว้ที่ตำแหน่ง 010FH

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____
_____
_____
_____

4. จงเขียนโปรแกรมโอนย้ายข้อมูล 0AH ไปเก็บไว้ในรีจิสเตอร์ A,B และที่ตำแหน่ง 0200H

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____
_____
_____
_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
คำสั่งทางคณิตศาสตร์ 1

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งทางคณิตศาสตร์ขนาด 8 บิตได้
2. สามารถเขียนโปรแกรมการคำนวณทางคณิตศาสตร์ขนาด 8 บิตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

กลุ่มคำสั่งการทำงานทางคณิตศาสตร์ขนาด 8 บิต

คำสั่งกลุ่มนี้ใช้สำหรับการ บวกลบข้อมูลขนาด 8 บิต 2 จำนวน ซึ่งข้อมูลจำนวนที่ 1 ต้องมาจากแอคคิวมูเลเตอร์ A ส่วนข้อมูลอีกจำนวนหนึ่งจะมาจากรีจิสเตอร์ B ภายในตัวซีพียู ข้อมูลทั้ง 2 จำนวนนี้จะมากระทำกันตามฟังก์ชันที่กำหนดโดยคำสั่ง ผลลัพธ์ที่ได้จากการกระทำคำสั่งจะเก็บไว้ที่แอคคิวมูเลเตอร์ A พร้อมทั้งสถานะต่างๆของการทำงานจะแสดงออกมาที่แฟลก

คำสั่งการบวก มีสองชนิด คือ ADD A,B และ ADC A,B คำสั่งแรกเป็นการบวกข้อมูลในแอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B ส่วนคำสั่งที่ 2 เป็นการบวกข้อมูลในแอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B และบวกแฟลกตัวทดเข้ามาที่ LSB ผลลัพธ์จากการบวกจะมาเก็บที่แอคคิวมูเลเตอร์ A

คำสั่งการลบ คือ SUB A,B เป็นการลบข้อมูลในแอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B ผลลัพธ์จากการลบจะมาเก็บที่แอคคิวมูเลเตอร์ A เช่นกัน

คำสั่งการบวกและลบข้อมูลมีอีก 2 คำสั่งคือ INC r และ DEC r คำสั่ง INC (Increment) คือคำสั่งการเพิ่มค่าขึ้นทีละ 1 ให้กับรีจิสเตอร์ ส่วนคำสั่ง DEC (Decrement) คือคำสั่งการลดค่าลงทีละ 1 ให้กับรีจิสเตอร์ ซึ่งมีลักษณะตรงข้ามกับคำสั่ง INC นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	02 02	LD B,02H
0004	08	ADD A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

2.

Address	Op-code	Mnemonic
0000	01 02	LD A,02H
0002	02 01	LD B,01H
0004	0A	SUB A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

3.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	02 02	LD B,02H
0004	1A	SET C
0005	09	ADC A,B
0006	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	0B	INC A
003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

5.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	0E	DEC A
003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

6.

Address	Op-code	Mnemonic
0000	02 02	LD B,02H
0002	0C	INC B
003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

7.

Address	Op-code	Mnemonic
0000	02 02	LD B,02H
0002	0F	DEC B
003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. คำสั่งการทำงานทางคณิตศาสตร์หมายถึงอะไร?

2. คำสั่ง INC A ต่างกับคำสั่ง DEC A อย่างไรจงอธิบาย?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จงเขียนโปรแกรมลดค่ารีจิสเตอร์ A ลงที่ละ 1 จนเหลือ 0 โดยจากเดิมรีจิสเตอร์ A มีข้อมูลอยู่ 03H

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
คำสั่งทางคณิตศาสตร์ 2

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งทางคณิตศาสตร์ที่มีการอ้างตำแหน่งแบบแฝงได้
2. สามารถเขียนโปรแกรมการคำนวณทางคณิตศาสตร์ที่มีการอ้างตำแหน่งแบบแฝงได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

คำสั่งทางคณิตศาสตร์ที่มีการอ้างตำแหน่งแบบแฝง (Implied Addressing)

คำสั่งทางคณิตศาสตร์ที่มีการอ้างตำแหน่งแบบแฝง (Implied Addressing) คือคำสั่งที่ไม่มีตัวกระทำ หรือมีตัวกระทำเพียง 1 ตัว คำสั่งที่มีตัวกระทำ 1 ตัวได้แก่ คำสั่ง CPL A และ CPL B

คำสั่ง CPL A เป็นคำสั่งที่ทำให้ข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ A เปลี่ยนเป็น 1's Complement ของข้อมูลเดิม คือถ้าบิตใดของข้อมูลเดิมเป็น "0" จะถูกเปลี่ยนเป็น "1" และบิตใดที่เป็น "1" จะถูกเปลี่ยนเป็น "0" ส่วนคำสั่ง CPL B เป็นคำสั่งที่ทำให้ข้อมูลในรีจิสเตอร์ B เปลี่ยนเป็น 1's Complement ของข้อมูลเดิมเช่นเดียวกัน

คำสั่ง CMP A,B เป็นคำสั่งที่ใช้ในการเปรียบเทียบข้อมูลระหว่างรีจิสเตอร์แอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B ว่าทีค่าเท่ากันหรือไม่ โดยคำสั่งนี้จะแสดงผลออกทางแฟลก CMP หากข้อมูลระหว่างรีจิสเตอร์แอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B มีค่าเท่ากัน แฟลก CMP จะเป็น "1" หากข้อมูลระหว่างรีจิสเตอร์แอคคิวมูเลเตอร์ A กับรีจิสเตอร์ B มีค่าไม่เท่ากัน แฟลก CMP จะเป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 0F	LD A,0FH
0002	17	CPL A
0003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

2.

Address	Op-code	Mnemonic
0000	02 F0	LD B,F0H
0002	18	CPL B
0003	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

3.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	02 02	LD B,02H
0004	19	CMP A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง CMP Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง CMP Flag = _____

4.

Address	Op-code	Mnemonic
0000	01 0F	LD A,0FH
0002	02 0F	LD B,0FH
0004	19	CMP A,B
0005	00	NOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง CMP Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง CMP Flag = _____

สรุปผลการทดลอง

_____

_____

_____

_____

_____

_____

_____

_____

คำถามท้ายการทดลอง

1. การเปรียบเทียบข้อมูลโดยใช้คำสั่ง CMP จะพิจารณาผลของการเปรียบเทียบได้ที่ใด?

_____

_____

_____

_____

_____

2. จงเขียนโปรแกรมที่ทำให้ตำแหน่ง 2000H เก็บค่า 1's Complement ของข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ A เมื่อข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ A มีค่า 0AH

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_____ _____ _____
_____ _____ _____
_____ _____ _____

อธิบายการทำงานของโปรแกรม

_____
_____
_____
_____
_____

3. จงเขียนโปรแกรมที่ทำให้ตำแหน่ง 3FFFH เก็บค่า 2's Complement ของข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ A เมื่อข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ A มีค่า 0FH

Address	**Op-code**	**Mnemonic**
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____
_____
_____
_____
_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5
คำสั่งทางคณิตศาสตร์ 3

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งทางคณิตศาสตร์ขนาด 16 บิตได้
2. สามารถเขียนโปรแกรมการคำนวณทางคณิตศาสตร์ขนาด 16 บิตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

กลุ่มคำสั่งการทำงานทางคณิตศาสตร์ขนาด 16 บิต

คำสั่งกลุ่มนี้ใช้สำหรับการเพิ่มค่าหรือลดค่าทีละ 1 ให้กับรีจิสเตอร์ ขนาด 16 บิต ซึ่งก็คือ รีจิสเตอร์คู่ HL ได้แก่ คำสั่ง INC HL และ DEC HL คำสั่ง INC (Increment) คือคำสั่งการเพิ่มค่าขึ้นทีละ 1 ให้กับรีจิสเตอร์คู่ HL ส่วนคำสั่ง DEC (Decrement) คือคำสั่งการลดค่าลงทีละ 1 ให้กับรีจิสเตอร์คู่ HL ซึ่งมีลักษณะตรงข้ามกับคำสั่ง INC นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	03 00 02	LD HL,0200H
0003	0D	INC HL
0004	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ H = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ L = _____

หลังกระทำคำสั่งรีจิสเตอร์ H = _____

หลังกระทำคำสั่งรีจิสเตอร์ L = _____

2.

Address	Op-code	Mnemonic
0000	03 FF 01	LD HL,01FFH
0003	10	DEC HL
0004	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ H = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ L = _____

หลังกระทำคำสั่งรีจิสเตอร์ H = _____

หลังกระทำคำสั่งรีจิสเตอร์ L = _____

3.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 03	LD B,03H
0004	03 00 02	LD HL,0200H
0007	09	ADD A,B
0008	0B	INC A
0009	07	LD (HL),A
000A	00	NOP

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งตำแหน่ง 0200H = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

4.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 03	LD B,03H
0004	03 00 02	LD HL,0200H
0007	09	ADD A,B
0008	0D	INC HL
0009	07	LD (HL),A
000A	00	NOP

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งตำแหน่ง 0201H = _____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. เหตุใดจึงต้องใช้รีจิสเตอร์คู่ในการคำสั่งการทำงานทางคณิตศาสตร์ขนาด 16 บิต?

2. คำสั่ง INC HL ต่างกับคำสั่ง DEC HL อย่างไรจงอธิบาย?

3. จงเขียนโปรแกรมนำข้อมูลในรีจิสเตอร์ B ไปเก็บไว้ที่ตำแหน่ง 020FH โดยจากเดิมรีจิสเตอร์ B มีข้อมูลอยู่ 03H และรีจิสเตอร์คู่ HL มีข้อมูลอยู่ 020CH

Address	Op-code	Mnemonic
_____	_____	_____
_____	_____	_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 6
คำสั่งทางลอจิก

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งทางลอจิกขนาด 8 บิตได้
2. สามารถเขียนโปรแกรมการคำนวณทางลอจิกขนาด 8 บิตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

กลุ่มคำสั่งการทำงานทางลอจิกขนาด 8 บิต

คำสั่งกลุ่มนี้ใช้สำหรับการ AND OR XOR NOR NAND และ XNOR ข้อมูลขนาด 8 บิต 2 จำนวน ซึ่งข้อมูลจำนวนที่ 1 ต้องมาจากแอคคิวมูเลเตอร์ A ส่วนข้อมูลอีกจำนวนหนึ่งจะมาจากรีจิสเตอร์ B ภายในตัวซีพียู ข้อมูลทั้ง 2 จำนวนนี้จะมากระทำกันตามฟังก์ชันที่กำหนดโดยคำสั่ง ผลลัพธ์ที่ได้จากการกระทำคำสั่งจะเก็บไว้ที่แอคคิวมูเลเตอร์ A พร้อมทั้งสถานะต่างๆ ของการทำงานจะแสดงออกมาที่แฟลก

โดยคำสั่งกลุ่มนี้ตัวกระทำหลักจะมาจากแอคคิวมูเลเตอร์ A ส่วนตัวกระทำอีกหนึ่งตัวจะมาจากรีจิสเตอร์ B ข้อมูลทั้ง 2 จำนวนนี้จะมากระทำกันตามฟังก์ชันลอจิกแบบบิตต่อบิต ผลลัพธ์ที่ได้จากการกระทำคำสั่งจะเก็บไว้ที่แอคคิวมูเลเตอร์ A การทำงานคำสั่งลอจิกนี้แฟลก C และ N จะเป็น 0 เสมอ แฟลก P/V จะเป็น แฟลก Parity ส่วนแฟลก S จะเป็น 1 เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	11	AND A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

2.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	12	OR A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

3.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	13	XOR A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	14	NAND A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

5.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	15	NOR A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

6.

Address	Op-code	Mnemonic
0000	01 05	LD A,05H
0002	02 0A	LD B,0AH
0004	16	XNOR A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของคำสั่งทางลอจิกขนาด 8 บิต?

2. จงอธิบายการทำงานของคำสั่ง NAND A,B ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จงอธิบายการทำงานของคำสั่ง XNOR A,B ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 7
คำสั่งการเซตและรีเซต

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งการเซตและรีเซตได้
2. สามารถเขียนโปรแกรมการเซตและรีเซตได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

คำสั่งการเซตและรีเซต

คำสั่งการเซตและรีเซตมีการอ้างตำแหน่งแบบแฝง (Implied Addressing) คือคำสั่งที่ไม่มีตัวกระทำ หรือมีตัวกระทำเพียง 1 ตัว คำสั่งที่มีตัวกระทำ 1 ตัวได้แก่ คำสั่ง SET C และ CLR C โดยทั้ง 2 คำสั่งนี้จะเป็นคำสั่งในระดับบิต หรือทำให้เกิดการเปลี่ยนแปลงที่บิตใดบิตหนึ่งเพียงบิตเดียวเท่านั้น

คำสั่ง SET C และ CLR C เป็นคำสั่งที่ทำงานเกี่ยวกับแฟลกตัวทด โดยที่คำสั่ง SET C (Set Carry Flag) จะทำให้ข้อมูลในแฟลกตัวทดถูกเซตเป็น "1" ส่วนคำสั่ง CLR C (Clear Carry Flag) จะทำให้ข้อมูลในแฟลกตัวทดถูกเซตเป็น "0" คำสั่งทั้ง 2 คำสั่งนี้ อาจใช้เพื่อทำการกำหนดค่าเริ่มต้นให้กับแฟลกตัวทด ก่อนที่จะกระทำคำสั่งทางคณิตศาสตร์หรือคำสั่งอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	1A	SET C
0001	00	NOP

ก่อนกระทำคำสั่งแฟลก Carry = _____

หลังกระทำคำสั่งแฟลก Carry = _____

2.

Address	Op-code	Mnemonic
0000	1A	SET C
0001	1B	CLR C
0002	00	NOP

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่ง Carry Flag = _____

3.

Address	Op-code	Mnemonic
0000	01 FF	LD A,FFH
0002	02 01	LD B,01H
0004	08	ADD A,B
0005	1B	CLR C
0006	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง Carry Flag = _____

4.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	02 02	LD B,02H
0004	1A	SET C
0005	09	ADC A,B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0006	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง Carry Flag = _____

5.

Address	**Op-code**	**Mnemonic**
0000	01 01	LD A,01H
0002	02 02	LD B,02H
0004	1A	SET C
0005	1B	CLR C
0006	09	ADC A,B
0007	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง Carry Flag = _____

6.

Address	**Op-code**	**Mnemonic**
0000	01 01	LD A,02H
0002	02 02	LD B,08H
0004	1A	SET C
0005	0A	SUB A,B
0006	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง Carry Flag = _____

7.

Address	Op-code	Mnemonic
0000	01 01	LD A,02H
0002	02 02	LD B,08H
0004	0A	SUB A,B
0005	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

ก่อนกระทำคำสั่ง Carry Flag = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่ง Carry Flag = _____

สรุปผลการทดลอง

_____

_____

_____

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. แฟลกตัวทดมีประโยชน์อะไรบ้าง?

2. การ Set และ Clear แฟลกตัวทดมีวิธีการใดบ้าง?

3. จงเขียนโปรแกรมที่ทำให้แฟลกตัวทดถูกเซตเป็น "1" โดยไม่ใช้คำสั่ง SET C

Address	**Op-code**	**Mnemonic**

อธิบายการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 8
คำสั่งการกระโดด

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งการกระโดดได้
2. สามารถเขียนโปรแกรมการกระโดดได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

คำสั่งการกระโดด

คำสั่งการกระโดดจะแบ่งเป็นคำสั่งที่มีการกระโดดอย่างไม่มีเงื่อนไข(Uncondition Jump) และ คำสั่งที่มีการกระโดดแบบมีเงื่อนไข(Condition Jump)

คำสั่งกระโดด คือ คำสั่งที่ทำการเปลี่ยนแปลงค่าในโปรแกรมเคาน์เตอร์ คำสั่งที่มีการกระโดดแบบไม่มีเงื่อนไขสามารถกระโดดไปยังตำแหน่งใดๆ ก็ได้ ได้แก่คำสั่ง JP nn โดย nn คือ เลขฐาน 16 ที่ใช้ระบุตำแหน่งของหน่วยความจำที่ต้องการกระโดดไป เช่น JP 17FFH หมายถึง ต้องการไปยังหน่วยความจำตำแหน่ง 17FFH

ส่วนคำสั่งการกระโดดแบบสัมพันธ์อย่างมีเงื่อนไข ก็ใช้วิธีการแบบเดียวกันนอกจากว่าการกระโดดจะเกิดขึ้นเมื่อเงื่อนไขตรงตามคำสั่งนั้นๆ คำสั่ง DJNZ e (Dcrement and Jump if Not Zero) การทำงานของคำสั่งนี้ ค่าในรีจิสเตอร์ B จะถูกลดลงไป "1" ถ้าค่าในรีจิสเตอร์ B ยังไม่เป็น "0" ก็จะเกิดการกระโดดไปยังตำแหน่งที่กำหนดจนค่าในรีจิสเตอร์ B เป็น "0" ก็จะกระทำคำสั่งต่อไป

ค่า e หาได้จาก

$$\text{ตำแหน่งที่ใช้งาน} = PC + 2 + e$$

หรือ $$e = \text{ตำแหน่งที่ใช้งาน} - PC - 2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	1C 07 00	JP 0007H
0005	02 02	LD B,02H
0007	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

2.

Address	Op-code	Mnemonic
0000	1C 07 00	JP 0007H
0003	01 01	LD A,01H
0005	02 02	LD B,02H
0007	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

3.

Address	Op-code	Mnemonic
0000	01 01	LD A,09H
0002	02 02	LD B,02H
0004	0E	DEC A
0005	1D FD	DJNZ FD
0007	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.

Address	Op-code	Mnemonic
0000	01 01	LD A,01H
0002	02 0F	LD B,0FH
0004	09	ADC A,B
0005	1D FD	DJNZ FD
0007	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เขียนโปรแกรมดังตัวอย่างที่ 5 และ 6 เปรียบเทียบผลที่ได้

5.

Address	Op-code	Mnemonic
0000	02 FF	LD B,FFH
0002	1D FC	DJNZ FC
0007	00	NOP

6.

Address	Op-code	Mnemonic
0000	02 FF	LD B,FFH
0002	1D FC	DJNZ FC
0004	02 FF	LD B,FFH
0006	1D F6	DJNZ F
0008	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ B = _____

หลังกระทำคำสั่งรีจิสเตอร์ B = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. คำสั่งกระโดดคืออะไร?

2. รีจิสเตอร์ B สัมพันธ์กับคำสั่ง DJNZ อย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คำสั่ง DJNZ มีประโยชน์อย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 9
คำสั่งการอินพุทและเอาท์พุท

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถอธิบายการทำงานของคำสั่งการอินพุทและเอาท์พุทได้
2. สามารถเขียนโปรแกรมการอินพุทและเอาท์พุทได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

คำสั่งการอินพุทและเอาท์พุท

คำสั่งกลุ่มนี้เป็นคำสั่งเพื่อใช้ในการส่งถ่ายข้อมูลระหว่างรีจิสเตอร์แอคคิวมูเลเตอร์ A กับอุปกรณ์อินพุทและเอาท์พุทภายนอก คำสั่งในกลุ่มนี้ ได้แก่ คำสั่ง IN A และ OUT A

คำสั่ง IN A เป็นคำสั่งเพื่อนำข้อมูลจากอุปกรณ์ภายนอกเข้ามาทางพอร์ตอินพุท ซึ่งใช้รีจิสเตอร์คู่ HL เป็นตัวชี้ตำแหน่ง มาเก็บไว้ในรีจิสเตอร์แอคคิวมูเลเตอร์ A ในที่นี้ใช้เมทริกซ์สวิตช์เป็นอุปกรณ์อินพุท ต่ออยู่กับ ไอซีเบอร์ 8255 ทำงานในโหมด 1 ให้พอร์ต A เป็นพอร์ตอินพุท พอร์ต B เป็นพอร์ตเอาท์พุท สามารถทำได้โดยกำหนดข้อมูลค่า 90H ให้ Control PORT

คำสั่ง OUT A เป็นคำสั่งเพื่อนำข้อมูลจากรีจิสเตอร์แอคคิวมูเลเตอร์ A ออกอุปกรณ์เอาท์พุทภายนอก ซึ่งใช้รีจิสเตอร์คู่ HL เป็นตัวชี้ตำแหน่ง อุปกรณ์เอาท์พุทภายนอกในที่นี้ได้แก่ ไดโอดเปล่งแสง(Light Emitting Diode : LED) ซึ่งใช้แสดงผลระดับบิต และ ไดโอดเปล่งแสงแบบ 7 ส่วน (7-Segment) ซึ่งประกอบไปด้วยไดโอดเปล่งแสง 7 ดวง ใช้แสดงผลเป็นตัวอักษรหรือตัวเลข

ตำแหน่งของอุปกรณ์อินพุทมีดังนี้

IC # 8255	PORT A	อยู่ที่ตำแหน่ง	FC00H
	PORT B	อยู่ที่ตำแหน่ง	FC01H
	PORT C	อยู่ที่ตำแหน่ง	FC02H
	Control PORT	อยู่ที่ตำแหน่ง	FC03H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของอุปกรณ์เอาท์พุทมีดังนี้

7-Segment 1	อยู่ที่ตำแหน่ง	FCA0H
7-Segment 2	อยู่ที่ตำแหน่ง	FCB0H
7-Segment 3	อยู่ที่ตำแหน่ง	FCC0H
7-Segment 4	อยู่ที่ตำแหน่ง	FCD0H
LED	อยู่ที่ตำแหน่ง	FCE0H

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 06	LD A,06H
0002	03 A0 FC	LD HL,FCA0H
0005	1F	OUT A
0006	1C 05 00	JP 0005H
0009	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

_____

2.

Address	Op-code	Mnemonic
0000	01 5B	LD A,5BH
0002	03 B0 FC	LD HL,FCB0H
0005	1F	OUT A
0006	1C 05 00	JP 0005H
0009	00	NOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

_____

3.

Address	Op-code	Mnemonic
0000	01 4F	LD A,4FH
0002	03 C0 FC	LD HL,FCC0H
0005	1F	OUT A
0006	1C 05 00	JP 0005H
0009	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

_____

4.

Address	Op-code	Mnemonic
0000	01 66	LD A,66H
0002	03 D0 FC	LD HL,FCD0H
0005	1F	OUT A
0006	1C 05 00	JP 0005H
0009	00	NOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

_____

5.

Address	**Op-code**	**Mnemonic**
0000	01 FF	LD A,FFH
0002	03 E0 FC	LD HL,FCE0H
0005	1F	OUT A
0006	1C 05 00	JP 0005H
0009	00	NOP

ก่อนกระทำคำสั่งรีจิสเตอร์ A = _____

ก่อนกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังกระทำคำสั่งรีจิสเตอร์ A = _____

หลังกระทำคำสั่งรีจิสเตอร์ HL = _____

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

_____

6.

Address	**Op-code**	**Mnemonic**
0000	01 90	LD A,90H
0002	03 03 FC	LD HL,FC03H
0005	07	LD (HL),A
0006	01 0E	LD A,0EH
0008	03 01 FC	LD HL,FC01H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000B	07	LD (HL),A
000C	02 1E	LD B,1EH
000E	03 00 FC	LD HL,FC00H
0011	1E	IN A
0012	11	AND A,B
0013	05	LD B,A
0014	1D FB	DJNZ FB
0016	01 6E	LD A,6EH
0018	03 D0 FC	LD HL,FCD0H
001B	1F	OUT A
001C	1C 1B 00	JP 001BH
001F	00	NOP

หลังจากกระทำตามโปรแกรมทดลองกดสวิตช์ผลที่ได้ คือ

_____

_____

_____

_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. คำสั่ง OUT A และคำสั่ง IN A ต่างกันอย่างไร?

2. อุปกรณ์เอาท์พุทที่ตำแหน่ง FCD0H และ FCE0H ต่างกันอย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คำสั่ง DJNZ ที่ใช้ในโปรแกรมที่ 6 มีประโยชน์อย่างไร?

_____

_____

_____

_____

_____

4. จงเขียนโปรแกรมให้ 7-Segment ทั้ง 4 ตัวแสดงคำว่า IDED เมื่อกดเมทริกซ์สวิตช์ ปุ่ม4

Address	Op-code	Mnemonic
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 10
การประยุกต์ใช้งานคำสั่งต่างๆ

วัตถุประสงค์เชิงพฤติกรรม

1. สามารถเขียนโปรแกรมด้วยคำสั่งต่างๆได้
2. สามารถเขียนโปรแกรมควบคุมอุปกรณ์อินพุทและเอาท์พุทด้วยคำสั่งต่างๆได้

เครื่องมือและอุปกรณ์

1. ชุดฝึกไมโครคอมพิวเตอร์เบื้องต้น
2. คู่มือคำสั่ง

ทฤษฎีเบื้องต้น

ในการเขียนโปรแกรมขึ้นมาแต่ละโปรแกรมขึ้นมาใช้งานเพื่อให้ได้ผลตามที่เราต้องการนั้น จำเป็นที่จะต้องใช้คำสั่งที่แตกต่างกัน ทั้งรูปแบบและปริมาณของคำสั่งโดยจะมีมากหรือน้อยก็จะขึ้นอยู่กับความซับซ้อนของโปรแกรมที่เขียนขึ้น ซึ่งจะถูกกำหนดโดยความต้องการหรือผลลัพธ์ของโปรแกรม

โปรแกรมอย่างง่ายหรือโปรแกรมที่มีความซับซ้อนไม่มากนัก ตัวโปรแกรมก็จะมีขนาดสั้นๆ หรือมีจำนวนคำสั่งและรูปแบบของคำสั่งไม่มากนัก สามารถเข้าใจการทำงานของโปรแกรมได้ง่าย อาจใช้รีจิสเตอร์เพียงแค่ 8 บิต ในการกระทำคำสั่งภายในโปรแกรม

ส่วนโปรแกรมที่มีความซับซ้อนมาก ตัวโปรแกรมก็จะมีขนาดยาวมากขึ้น ขึ้นอยู่กับความซับซ้อนของโปรแกรมนั้นๆ โปรแกรมที่ซับซ้อนมากจำนวนคำสั่งและรูปแบบของคำสั่งก็จะยิ่งมากขึ้น การทำความเข้าใจถึงการทำงานของโปรแกรมก็ทำได้ยากขึ้น ต้องแบ่งโปรแกรมออกเป็นส่วนๆ เพื่อทำความเข้าใจในแต่ละส่วนจากนั้นจึงนำแต่ละส่วนมารวมกัน

คำสั่งแบบเงื่อนไขจะมีประโยชน์อย่างมากในการลดขนาดของโปรแกรมที่มีความซับซ้อน เช่น ต้องการกระทำคำสั่งหนึ่งหรือชุดคำสั่งหนึ่งซ้ำกันหลายๆ ครั้ง ทำให้ต้องเขียนคำสั่งหรือชุดคำสั่งนั้นซ้ำๆ กัน โปรแกรมก็มีขนาดยาวขึ้น หากใช้คำสั่ง DJNZ จะช่วยลดจำนวนของคำสั่งที่ต้องการทำซ้ำๆ กันจำนวนหลายครั้ง ให้เขียนเพียงครั้งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

เขียนโปรแกรมดังตัวอย่าง บันทึกผลที่ได้

1.

Address	Op-code	Mnemonic
0000	01 00	LD A,00H
0002	03 E0 FC	LD HL,FCE0H
0005	1F	OUT A
0006	02 FF	LD B,FFH
0008	1D FE	DJNZ FE
000A	0B	INC A
000B	1C 05 00	JP 0005H
000E	00	NOP

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

2.

Address	Op-code	Mnemonic
0000	01 00	LD A,00H
0002	03 A0 FC	LD HL,FCA0H
0005	1F	OUT A
0006	02 FF	LD B,FFH
0008	1D FE	DJNZ FE
000A	0B	INC A
000B	1C 05 00	JP 0005H
000E	00	NOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

3.

Address	Op-code	Mnemonic
0000	01 FF	LD A,FFH
0002	03 A0 FC	LD HL,FCA0H
0005	1F	OUT A
0006	02 FF	LD B,FFH
0008	1D FE	DJNZ FE
000A	0E	DEC A
000B	1C 05 00	JP 0005H
000E	00	NOP

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

4.

Address	Op-code	Mnemonic
0000	01 FF	LD A,FFH
0002	03 A0 FC	LD HL,FCA0H
0005	1F	OUT A
0006	02 FF	LD B,FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0008	1D FE	DJNZ FE
000A	02 03	LD B,03H
000C	0E	DEC A
000D	1D FD	DJNZ FD
000F	1C 05 00	JP 0005H
0012	00	NOP

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

_____

_____

_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

5.

Address	**Op-code**	**Mnemonic**
0000	01 90	LD A,90H
0002	03 03 FC	LD HL,FC03H
0005	07	LD (HL),A
0006	01 0E	LD A,0EH
0008	03 01 FC	LD HL,FC01H
000B	07	LD (HL),A
000C	02 1E	LD B,1EH
000E	03 00 FC	LD HL,FC00H
0011	1E	IN A
0012	11	AND A,B
0013	05	LD B,A
0014	1D FB	DJNZ FB
0016	01 01	LD A,01H
0018	03 E0 FC	LD HL,FCE0H
001B	1F	OUT A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001C	02 FF	LD B,FFH
001E	1D FE	DJNZ FE
0020	01 02	LD A,02H
0022	1F	OUT A
0023	02 FF	LD B,FFH
0025	1D FE	DJNZ FE
0027	01 02	LD A,05H
0029	1F	OUT A
002A	02 FF	LD B,FFH
002C	1D FE	DJNZ FE
002E	01 02	LD A,0AH
0030	1F	OUT A
0031	02 FF	LD B,FFH
0033	1D FE	DJNZ FE
0035	01 02	LD A,15H
0037	1F	OUT A
0038	02 FF	LD B,FFH
003A	1D FE	DJNZ FE
003C	01 02	LD A,2AH
003E	1F	OUT A
003F	02 FF	LD B,FFH
0041	1D FE	DJNZ FE
0043	01 02	LD A,55H
0045	1F	OUT A
0046	02 FF	LD B,FFH
0048	1D FE	DJNZ FE
004A	01 02	LD A,AAH
004C	1F	OUT A
004D	02 FF	LD B,FFH
004F	1D FE	DJNZ FE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0051	01 02	LD A,54H
0053	1F	OUT A
0054	02 FF	LD B,FFH
0056	1D FE	DJNZ FE
0058	01 02	LD A,A8H
005A	1F	OUT A
005B	02 FF	LD B,FFH
005D	1D FE	DJNZ FE
005F	01 02	LD A,50H
0061	1F	OUT A
0062	02 FF	LD B,FFH
0064	1D FE	DJNZ FE
0066	01 02	LD A,A0H
0068	1F	OUT A
0069	02 FF	LD B,FFH
006B	1D FE	DJNZ FE
006D	01 02	LD A,40H
006F	1F	OUT A
0071	02 FF	LD B,FFH
0072	1D FE	DJNZ FE
0074	01 02	LD A,80H
0076	1F	OUT A
0077	02 FF	LD B,FFH
0079	1D FE	DJNZ FE
007B	1C 2E 00	JP 002EH
007E	00	NOP

หลังจากกระทำตามโปรแกรมผลที่ได้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของโปรแกรม

สรุปผลการทดลอง

คำถามท้ายการทดลอง

1. โปรแกรมคืออะไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คำสั่ง DJNZ ที่ใช้ในโปรแกรมมีประโยชน์อย่างไร?

_____

_____

_____

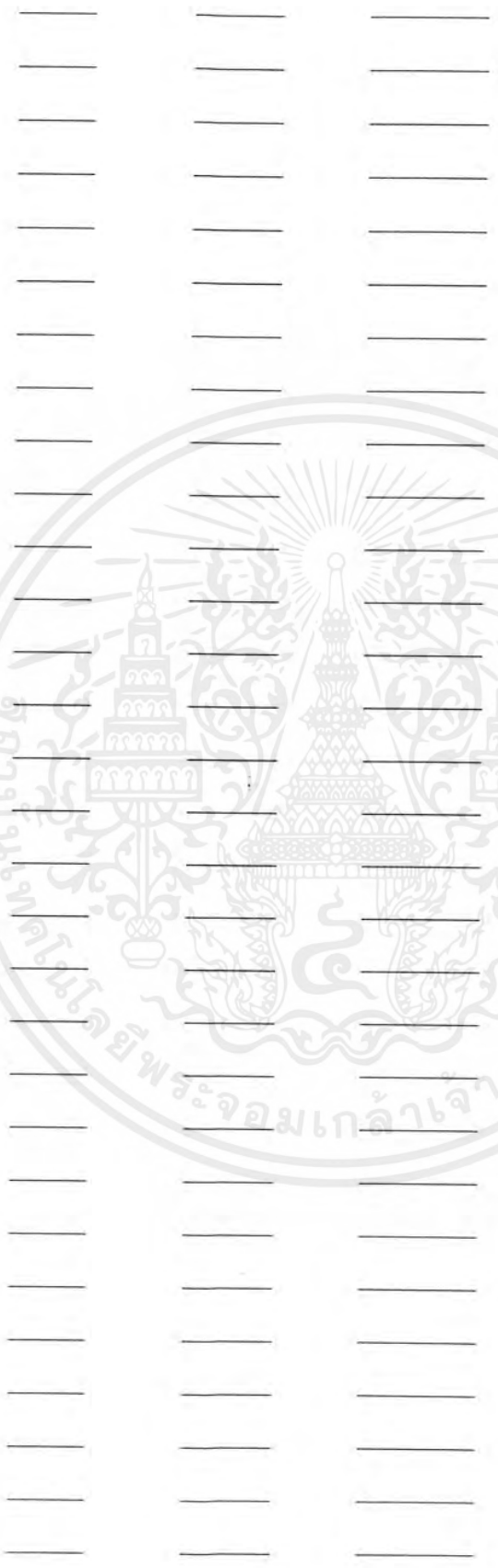
_____

_____

3. จงเขียนโปรแกรมให้ 7-Segment แสดงการนับถอยหลังจาก 9 - 0 เมื่อกดเมทริกซ์สวิตช์ปุ่มที่ 1

Address	Op-code	Mnemonic
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

อธิบายการทำงานของโปรแกรม

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ยืน ภู่วรวรรณ และวัฒนา เชียงกูล. ไมโครโปรเซสเซอร์ไมโครคอมพิวเตอร์ : ซีเอ็ดยูเคชั่น จำกัด วิบูลย์ ชื่นแขก. ไมโครโปรเซสเซอร์ : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2533

Richard Larry Ukriley. Field Programmable Gate Array (FPGAs) : Practice-Hall linc. 1993

Xilinx Inc. Foundation Series Software : Xilinx Inc. 1994

Xilinx Inc. The Programmable Logic Data Book : Xilinx Inc. 1994

Xilinx Inc. Viewlogic Tutorials : Xilinx Inc. 1994

Xilinx Inc. XACT Hardware&Peripherals Guide : Xilinx Inc. 1994

Xilinx Inc. XCAT Reference Guide. Volume II : Xilinx Inc. 1994

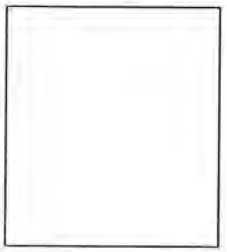
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายจุติ จุติประเสริฐ
วันเดือนปีเกิด	23 พฤศจิกายน 2522
สถานที่เกิด	จังหวัดนครศรีธรรมราช
ภูมิลำเนาเดิม	1363 ถ.ราชดำเนิน ต.คลัง อ.เมือง จ.นครศรีธรรมราช 80000 (075-356714)
ที่อยู่ปัจจุบัน	25/11 ซ.เกษมสันต์ 1 ถ.พระราม 1 เขตพญาไท กรุงเทพฯ 10330
โทรศัพท์	02-2151867
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนศรีธรรมราชศึกษา
มัธยมศึกษา	โรงเรียนศรีธรรมราชศึกษา
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคนครศรีธรรมราช
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	วิทยาลัยเทคนิคนครศรีธรรมราช
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รางวัล	-
ทุนการศึกษา	-
คติพจน์	ตนเป็นที่พึ่งแห่งตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นางสาวปิยะวรรณ คำน้ำนอง
วันเดือนปีเกิด	8 ตุลาคม 2520
สถานที่เกิด	จังหวัดลำปาง
ภูมิลำเนาเดิม	308/2 หมู่ 7 ต.พระบาท อ.เมือง จ.ลำปาง 52000
ที่อยู่ปัจจุบัน	205-205/1 ซ.ฉลองชัย1 ถ.ฉลองกรุง แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	02-7373859 ต่อ 22
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนพินิจวิทยา
มัธยมศึกษา	โรงเรียนลำปางกัลยาณี
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคลำปาง
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคพายัพ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รางวัล	-
ทุนการศึกษา	ยกเว้นหน่วยกิต
คติพจน์	ตนเป็นที่พึ่งแห่งตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายอรุณชัย ศุภฤกษ์จันทราภา
วันเดือนปีเกิด	16 กันยายน 2521
สถานที่เกิด	จังหวัดสุราษฎร์ธานี
ภูมิลำเนาเดิม	456/53 ถ.ตลาดใหม่ ซ.พาราไดส์ ต.ตลาด อ.เมือง จ.สุราษฎร์ธานี 84000 (077-286015)
ที่อยู่ปัจจุบัน	261/117 ยุคลรัตน์คอนโด หมู่ 1 ซ.ดับเพลิง ถ.หลวงแพ่ง แขวงทับยาว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	02-7380005
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเทพมิตรศึกษา
มัธยมศึกษา	โรงเรียนสุราษฎร์พิทยา
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคสุราษฎร์ธานี
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตพระนครเหนือ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รางวัล	-
ทุนการศึกษา	-
คติพจน์	ตนเป็นที่พึ่งแห่งตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้