ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดสาธิตสถาปัตยกรรมภายใน CPU

CPU Architecture Demonstrator

ชื่อนักศึกษา

1. นายธเนศ	สุขแสง	รหัสประจำตัว	43035412
2. นายอรรณพ	เรืองยศจันทนา	รหัสประจำตัว	43035444
3. นายบุญชนะ	นุชชม	รหัสประจำตัว	43035626

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา อาจารย์โกศล ตราชู

อาจารย์ที่ปรึกษาร่วม อาจารย์สุชิน อาจหาญ

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์โกศล	ตราชู	.....
2. อาจารย์สุชิน	อาจหาญ	.....
3. ผศ.วิสุทธิ์	อธิพรธรรม	.....
4. อาจารย์กิติพงค์	มะโน	.....
5. อาจารย์พีระวุฒิ	สุวรรณจันทร์	.....

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 22 พฤศจิกายน พ.ศ. 2544 เวลา 19.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 20 เดือน ธันวาคม พ.ศ. 2544

<BT4403062>

ชุดสาธิตสถาปัตยกรรมภายใน CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดสาธิตสถาปัตยกรรมภายใน CPU

CPU Architecture Demonstrator

นายธเนศ สุขแสง
นายอรรณพ เรืองยศจันทนา
นายบุญชนะ นุชชม

เลขหมู่.....
เลขทะเบียน 44211
วัน, เดือน, ปี 4 พ.ย. 2545

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดสาธิตสถาปัตยกรรมภายใน CPU

CPU Architecture Demonstrator

วัตถุประสงค์

1) เพื่อศึกษาสถาปัตยกรรมภายใน CPU และหลักการออกแบบ CPU
2) เพื่อออกแบบชุดสาธิตสถาปัตยกรรมภายใน CPU
3) เพื่อสร้างชุดสาธิตสถาปัตยกรรมภายใน CPU
4) เพื่อทำการทดลองชุดสาธิตสถาปัตยกรรมภายใน CPU
5) เพื่อนำชุดสาธิตสถาปัตยกรรมภายใน CPU ไปประกอบการเรียนการสอน

ประโยชน์ที่คาดว่าจะได้รับ

1) มีความรู้เกี่ยวกับสถาปัตยกรรมภายใน CPU และหลักการออกแบบ CPU
2) ได้วงจรต้นแบบของชุดสาธิตสถาปัตยกรรมภายใน CPU
3) ได้เครื่องต้นแบบของชุดสาธิตสถาปัตยกรรมภายใน CPU
4) ได้ปรับปรุงชุดสาธิตสถาปัตยกรรมภายใน CPU
5) ได้นำชุดสาธิตสถาปัตยกรรมภายในของ CPU ไปใช้ในการเรียนการสอนวิชาไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดสาธิตสถาปัตยกรรมภายใน CPU
นักศึกษา	นายธเนศ สุขแสง
	นายอรรณพ เรืองยศจันทนา
	นายบุญชนะ นุชชม
อาจารย์ที่ปรึกษา	อาจารย์สุชิน อาจหาญ
อาจารย์ที่ปรึกษาร่วม	อาจารย์โกศล ตราชู
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรรม
ปีการศึกษา	2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอการออกแบบและการสร้างชุดสาธิตสถาปัตยกรรมภายใน CPU เพื่อแสดงให้เห็นถึงหลักการทำงานของ CPU และขั้นตอนการทำงานของ CPU ในการประมวลผลคำสั่งต่างๆ ที่ CPU ได้รับคำสั่งให้กระทำ ชุดสาธิตสถาปัตยกรรมภายใน CPU ได้ถูกออกแบบ และสร้างขึ้นโดยใช้ ไอซีลอจิกเกตพื้นฐานประเภท TTL มีการแสดงผลข้อมูลต่างๆ ที่อยู่ภายในตัว CPU โดยใช้ตัวเลขแสดงผล 7 ส่วนเป็นอุปกรณ์แสดงผล เพื่อให้ผู้ที่ศึกษาการทำงานของชุดสาธิตสถาปัตยกรรมภายใน CPU แล้วมีความเข้าใจในการทำงานภายใน CPU และสามารถพัฒนาไปใช้งาน CPU เบอร์ต่างๆ ได้อย่างไม่ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	CPU Architecture Demonstrator
Students	Mr.Tanet Suksang
	Mr.Unnop Ruengyosjantana
	Mr.Bunchana Nuchchom
Advisor	Mr.Suchin Adhan
Co-Advisor	Mr.Koson Trachu
Education Level	Bachelor of Science in Industrial Education
Program in	Industrial Instrumentation Technology
Academic Year	2001

ABSTRACT

This thesis is presented for designing and building CPU Architecture Demonstrator to display operation of CPU in Instruction procesing. The CPU has desinged and bulied by using basic Integrat circuit (IC). It can display data inside CPU by using LED segments and operate as well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณอาจารย์สุชิน อาจหาญ อาจารย์ที่ปรึกษา และอาจารย์โกศล ตราชู อาจารย์ที่ปรึกษาร่วม รวมทั้งคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ให้คำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางการแก้ไขปัญหา รวมถึงขอบคุณ พิทยา ปิยพงษ์ และเพื่อนๆ หลายท่านที่ให้ความช่วยเหลือ ยืมวัสดุอุปกรณ์ ให้คำปรึกษา และให้กำลังใจในการทำโครงงานชิ้นนี้ให้ประสบผลสำเร็จด้วยดี ในการจัดทำปริญญานิพนธ์ ขอขอบคุณ ห้องสมุดคณะครุศาสตร์อุตสาหกรรม และสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ช่วยอำนวยความสะดวก และเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูล สุดท้ายที่ควรระลึกถึงอย่างยิ่ง บิดา มารดา คณาจารย์ที่ได้อบรมสั่งสอนมาในอดีตจนถึงปัจจุบัน รวมถึงญาติพี่น้อง ที่เป็นผู้ให้การสนับสนุนด้านการศึกษา และเป็นผู้ให้กำลังใจด้วยดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 หลักการทำงานของ CPU	3
2.2.1 การทำงานของ CPU	4
2.2.2 ส่วนประกอบของ CPU Z80	5
2.3 สถาปัตยกรรมภายใน CPU ที่ใช้ในชุดสาธิต	6
2.4 ส่วนประกอบของชุดสาธิตสถาปัตยกรรมภายใน CPU	6
2.4.1 รีจิสเตอร์อาเรย์	6
2.4.2 หน่วยคำนวณทางคณิตศาสตร์ทางลอจิก	12
2.4.3 อินสตรัคชั่นรีจิสเตอร์คอนโทรลลอจิกยูนิต	14
2.4.4 ดาต้าบัสบัฟเฟอร์	14
2.4.5 แอดเดรสบัสบัฟเฟอร์	15
2.5 วิธีการอ้างตำแหน่งที่อยู่ข้อมูลของ CPU	15
2.5.1 อิมพรายซ์แอดเดรสซิ่ง	15
2.5.2 อิมมีเดียสแอดเดรสซิ่ง	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.5.3 เอ็กซ์เทนเด็กซ์อิมมีเดียสแอดเดรสซิ่ง	17
2.5.4 รีจิสเตอร์แอดเดรสซิ่ง	17
2.5.5 รีจิสเตอร์อินไดร์เร็กซ์แอดเดรสซิ่ง	18
2.5.6 เอ็กซ์เทนเด็กซ์แอดเดรสซิ่งหรือ ไดร์เร็กซ์แอดเดรสซิ่ง	18
2.5.7 โมดิไฟล์เพจซีโร่แอดเดรสซิ่ง	18
2.5.8 รีเรย์ทีฟแอดเดรสซิ่ง	19
2.5.9 อินเด็กซ์แอดเดรสซิ่ง	20
2.5.10 บิตแอดเดรสซิ่ง	21
2.6 การทำงานภายใน CPU	21
2.6.1 กรณีการ เฟ็ทช์	21
2.6.2 กรณีการ เอ็กซ์ซีคิว	23
2.7 หลักการทำงานของภาคแสดงผล	27
2.7.1 หลักการ สแกนดิสเพลย์	29
บทที่ 3 การออกแบบการสร้างและการทำงาน	30
3.1 ขั้นตอนการออกแบบ	30
3.1.1 ส่วนวงจรภายใน CPU	30
3.1.2 ส่วนการป้อนโปรแกรม	30
3.1.3 ส่วนของการแสดงผลข้อมูล	30
3.2 วงจรภายใน CPU	30
3.2.1 รีจิสเตอร์ทั่วไปขนาด 8 บิต	30
3.2.2 ส่วนการประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิก	37
3.2.3 ส่วนการควบคุมการทำงานของ CPU (คอนโทรลลอจิก)	40
3.2.4 ส่วนการเชื่อมต่อสัญญาณต่าง ๆ ภายใน CPU	43
3.3 ส่วนการป้อนโปรแกรม	44
3.4 ส่วนการแสดงผลข้อมูล	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 การทดลองและผลการทดลอง	51
4.1 การทดลองแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต	51
4.1.1 หน้าที่ของจุดเชื่อมต่อบนแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต	51
4.1.2 การทดลองโหลดข้อมูล 8 บิตจากอินเตอร์นอลดาต้าบัส ลงในรีจิสเตอร์	52
4.1.3 การทดลองส่งข้อมูลที่เก็บในรีจิสเตอร์ออกสู่อินเตอร์นอลดาต้าบัส	53
4.2 การทดลองแผงวงจรรีจิสเตอร์ A, F	54
4.2.1 หน้าที่ของจุดเชื่อมต่อบนแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต	55
4.2.2 การทดลองโหลดข้อมูล 8 บิตจากอินเตอร์นอลดาต้าบัสลงในรีจิสเตอร์ A, F	55
4.2.3 การทดลองส่งข้อมูลที่เก็บในรีจิสเตอร์ออกสู่ อินเตอร์นอลดาต้าบัส	55
4.3 การทดลองแผงวงจรรีจิสเตอร์ IX, IY	56
4.4 การทดลองแผงวงจรสแตกพอยน์เตอร์ 16 บิต	58
4.5 การทดลองแผงวงจรโปรแกรมเคาท์เตอร์ 16 บิต	59
4.6 การทดลองแผงวงจรรีจิสเตอร์พักข้อมูล	61
4.7 การทดลองแผงวงจร ALU 8 บิต	62
4.8 การทดลองแผงวงจรตรวจสอบแฟลกศูนย์ และแฟลกพาริตี้	63
4.9 การทดลองแผงวงจรพอร์ต	64
4.10 การทดลองแผงวงจรเฟ็ทช์	65
4.11 การทดลองแผงวงจรมัลติเพล็กซ์ 12 ช่อง	66
4.12 การทดลองแผงวงจรขับตัวเลขแสดงผล 7 ส่วน	67
4.13 การทดลองแผงวงจรนับแบบวงแหวน 54 บิต	68
4.14 การทดลองแผงวงจรเข้ารหัสคีย์เป็นสัญญาณ BCD-8421	69
4.15 การทดลองแผงวงจรกำหนดค่าแอดเดรสของหน่วยความจำภายนอก	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	71
5.1 สรุป	71
5.2 ปัญหา และแนวทางแก้ไข	71
5.3 แนวทางการพัฒนา	73
ภาคผนวก ก เครื่องต้นแบบ	74
ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์	76
ภาคผนวก ค รายการอุปกรณ์	100
บรรณานุกรม	107
ประวัติผู้แต่ง	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 กลุ่มรีจิสเตอร์ขนาด 8 บิต	6
ตารางที่ 2.2 กลุ่มรีจิสเตอร์ขนาด 16 บิต	10
ตารางที่ 2.3 ตำแหน่งการจบโปรแกรม	19
ตารางที่ 2.4 การขับส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม	28
ตารางที่ 3.1 ค่าที่เขียนลงในอีพรอมเพื่อให้แสดงค่า 0-F	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ผังการทำงานของ CPU เบื้องต้น	3
รูปที่ 2.2 สถาปัตยกรรมภายใน CPU Z80	4
รูปที่ 2.3 สถาปัตยกรรมภายใน CPU	6
รูปที่ 2.4 การเลื่อนข้อมูลในรีจิสเตอร์ A, F เข้าในหน่วยความจำสแตก	12
รูปที่ 2.5 การเลื่อนข้อมูล รีจิสเตอร์ B, C เข้าออกในหน่วยความจำสแตก	12
รูปที่ 2.6 การเลื่อนบิตไปทางซ้ายมือ	13
รูปที่ 2.7 การหมุนข้อไปทางซ้ายมือ	13
รูปที่ 2.8 คอนโทรลลอจิก และอินสตรัคชั่นรีจิสเตอร์	14
รูปที่ 2.9 ดาต้าบัสบัฟเฟอร์	15
รูปที่ 2.10 แอดเดรสบัสบัฟเฟอร์	15
รูปที่ 2.11 จำนวนที่ใช้ในการเฟ็ทช์ข้อมูล	22
รูปที่ 2.12 ส่วนแสดงผล 7 ส่วน	27
รูปที่ 2.13 โครงสร้างส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม	27
รูปที่ 2.14 ผังการทำงานภาคสแกนดิสเพลย์ 54 หลัก	29
รูปที่ 3.1 รีจิสเตอร์เก็บค่าทั่วไป 8 บิต	31
รูปที่ 3.2 รีจิสเตอร์พักข้อมูล 8 บิต	32
รูปที่ 3.3 อินเด็กซ์รีจิสเตอร์	33
รูปที่ 3.4 โปรแกรมเคาท์เตอร์	35
รูปที่ 3.5 แสตกพอยน์เตอร์	36
รูปที่ 3.6 วงจรประมวลผลทางคณิตศาสตร์ 8 บิต	37
รูปที่ 3.7 วงจรบวกเลข 16 บิต	39
รูปที่ 3.8 วงจรเฟ็ทช์ และถอดรหัส	40
รูปที่ 3.9 อีพรอมต่อขนาน 4 ตัวเพื่อควบคุม 32 ช่องสัญญาณ	41
รูปที่ 3.10 วงจรแปลงสัญญาณกลุ่มข้อมูลเป็นกลุ่มสัญญาณตำแหน่ง	42
รูปที่ 3.11 วงจรเปลี่ยนบัสตำแหน่งภายในเป็นข้อมูลภายใน	43
รูปที่ 3.12 วงจรเข้ารหัสคีย์เป็น BCD-8421	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.13 ผังการทำงานของส่วนแสดงผล 7 ส่วน	45
รูปที่ 3.14 วงจรรับสัญญาณ BCD-8421	46
รูปที่ 3.15 วงจรแปลงสัญญาณ BCD-8421 ขับส่วนแสดงผล 7 ส่วน	46
รูปที่ 3.16 ภาคแสดงผลตัวเลขแสดงผล 7 ส่วน	48
รูปที่ 3.17 เส้นทางของสัญญาณในชุดสาธิตสถาปัตยกรรมภายใน CPU	48
รูปที่ 3.18 การติดต่อกับหน่วยความจำภายนอกของชุดสาธิตสถาปัตยกรรมภายใน CPU	49
รูปที่ 3.19 การติดต่อกับอุปกรณ์ภายนอกของชุดสาธิตสถาปัตยกรรมภายใน CPU	49
รูปที่ 3.20 การป้อนโปรแกรมของชุดสาธิตสถาปัตยกรรมภายใน CPU	50
รูปที่ 4.1 แผงวงจรรีจิสเตอร์ทั่วไป 8 บิต	51
รูปที่ 4.2 แผงวงจรรีจิสเตอร์ A, F	54
รูปที่ 4.3 แผงวงจรรีจิสเตอร์ IX, IY	56
รูปที่ 4.4 แผงวงจรสแตกพอยน์เตอร์	58
รูปที่ 4.5 แผงวงจร โปรแกรมเคาน์เตอร์ 16 บิต	60
รูปที่ 4.6 แผงวงจรรีจิสเตอร์พักข้อมูล	61
รูปที่ 4.7 แผงวงจรคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก 8บิต	62
รูปที่ 4.8 แผงวงจรตรวจสอบแฟลกศูนย์และแฟลกพาริตี้	63
รูปที่ 4.9 แผงวงจรพอร์ต	64
รูปที่ 4.10 แผงวงจรเฟ็ทช์	65
รูปที่ 4.11 แผงวงจรมัลติเพล็กซ์ 12 ช่อง	66
รูปที่ 4.12 แผงวงจรขับส่วนแสดงผล 7 ส่วน	67
รูปที่ 4.13 แผงวงจรนับแบบวงแหวน 54 บิต	68
รูปที่ 4.14 แผงวงจรเข้ารหัสคีย์เป็นสัญญาณ BCD-8421	69
รูปที่ 4.15 แผงวงจรกำหนดค่าแอดเดรสของหน่วยความจำภายนอก	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบัน คอมพิวเตอร์มีบทบาทเกี่ยวข้องกับชีวิตประจำวันมากมาย การมีความรู้ในเรื่องของคอมพิวเตอร์นับว่ามีความสำคัญอย่างยิ่ง โดยเฉพาะในด้านอุตสาหกรรม ซึ่งจะต้องอาศัยผู้เชี่ยวชาญในด้านการใช้งานเครื่องคอมพิวเตอร์กับงานอุตสาหกรรม เพื่อให้ได้การผลิตในจำนวนที่มากขึ้น และทำให้ประเทศไทยได้ก้าวหน้าทันกับเทคโนโลยีอุตสาหกรรมผู้เชี่ยวชาญเหล่านั้นจะต้องมีความเข้าใจในระบบของคอมพิวเตอร์ รู้โครงสร้าง, กระบวนการทำงาน, โปรแกรมควบคุม เป็นต้น วิธีการในการศึกษาเล่าเรียนที่จะให้มีความรู้มีทักษะดังกล่าว ทั้งหมดพบในหลักสูตร ปวช., ปวส. และระดับปริญญาตรีที่เป็นวิชาเกี่ยวกับไมโครโปรเซสเซอร์เพื่อเป็นแนวทางในการศึกษาระบบที่มีขนาดใหญ่ขึ้นไป

การเรียนเรื่องไมโครโปรเซสเซอร์ จะพบว่าผู้สอนจะต้องรู้ และเข้าใจการทำงานภายในของ CPU อย่างลึกซึ้ง จึงสามารถถ่ายทอดกระบวนการภายในของ CPU ให้ผู้เรียนเกิดจินตนาการตามได้ส่วนผู้เรียนจะต้องมีความสนใจ และมีจินตนาการตามผู้สอนมีความคิดสร้างสรรจึงสามารถรู้และเข้าใจกระบวนการทำงานภายใน CPU ได้อย่างถูกต้องจะทำให้ผู้สอนสามารถอธิบายโครงสร้างกระบวนการทำงานได้ง่ายขึ้น จะทำให้ผู้เรียนเกิดความสนใจมากกว่าที่เป็นอยู่ สามารถมองภาพกระบวนการทำงานที่เกิดขึ้นได้เข้าใจเป็นอย่างดี และสามารถนำความรู้ที่ได้ไปประยุกต์ใช้ในการเรียนรู้ในส่วนการทำงานของสถาปัตยกรรมภายใน CPU ที่มีความซับซ้อนได้

1.2 ขีดความสามารถของโครงงาน

โครงงานชุดสาธิตสถาปัตยกรรมภายใน CPU มีขีดความสามารถดังต่อไปนี้

1) เป็น CPU ที่มีขนาดดาต้าบัส 8 บิต (D7-D0) และมีขนาดแอดเดรสบัส 16 บิต (A15-A0)

2) มีรีจิสเตอร์ที่ใช้งานทั่วไป B, C, D, E, H, L ขนาด 8 บิตหรือใช้เป็นรีจิสเตอร์คู่ BC, DE, HL ขนาด 16 บิตได้

3) มีอินเด็กซ์รีจิสเตอร์ IX, IY เป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้อินเด็กซ์แอดเดรสซิ่งโหมด

4) มีสแตกพอยน์เตอร์ ขนาด 16 บิต ใช้เก็บตำแหน่งที่อยู่ของข้อมูลที่ถูกนำไปพักชั่วคราว

5) มีโปรแกรมเคาน์เตอร์ ขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) มีหน่วยคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก ขนาด 8 บิต เช่น การบวกเลข ลบเลข และการกระทำทางลอจิก แอน, ออร์, เอ็กคลูซีพออร์

7) ชุดคำสั่งที่ใช้ใน CPU สามารถเพิ่มได้โดยเขียนรูปแบบการทำคำสั่งลงในอีพรอมที่ใช้เก็บไมโครโค้ด

8) มีตัวเลขแสดงผล 7 ส่วน สำหรับใช้แสดงผลค่าต่างๆ ภายใน CPU จำนวน 54 หลัก

1.3 เนื้อหาโดยสังเขป

ปริญญานิพนธ์นี้ประกอบด้วยรายละเอียดต่าง ๆ มีหัวข้อดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึง ความเป็นมา และความสำคัญของปริญญานิพนธ์ ขีดความสามารถโครงงานชุดสาธิตสถาปัตยกรรมภายใน CPU

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึง โครงสร้างของ CPU หลักการทำงานพื้นฐานของ CPU และหลักการแสดงผลข้อมูลภายใน CPU

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึง การออกแบบวงจรของชุดสาธิตสถาปัตยกรรมภายใน CPU ในแต่ละภาค และการสร้างชุดสาธิตสถาปัตยกรรมภายใน CPU ในแต่ละภาค

บทที่ 4 การทดลอง และผลการทดลอง กล่าวถึง วิธีการทดลอง และผลที่ได้จากการทดลองของชุดสาธิตสถาปัตยกรรมภายใน CPU

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา กล่าวถึง สรุปการทำโครงงาน ปัญหาที่เกิดขึ้นระหว่างการทำโครงงาน รวมถึงแนวทางการแก้ไขปัญหาที่เกิดขึ้นพร้อมทั้งได้เสนอแนะแนวทางในการพัฒนาต่อไป

ในภาคผนวก แสดงรายละเอียดของการสร้างชุดสาธิตสถาปัตยกรรมภายใน CPU รวมถึงรายละเอียดของอุปกรณ์ต่างๆ ที่ใช้ในการทำโครงงานดังนี้

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์

ภาคผนวก ค รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี และหลักการ

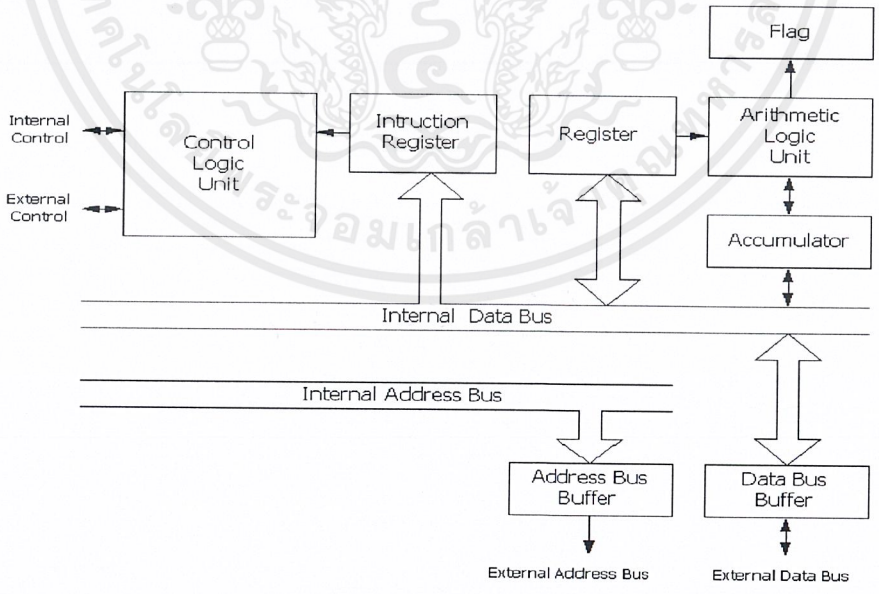
2.1 กล่าวนำ

เนื้อหาของปริญญานิพนธ์ในบทนี้เป็นทฤษฎี และหลักการที่นำมาใช้ประกอบการสร้างโครงงาน โดยประกอบไปด้วยเรื่องของหลักการทำงานของ CPU วิธีการอ้างตำแหน่งที่อยู่ข้อมูลของ CPU การทำคำสั่งของ CPU หลักการแสดงผลข้อมูลที่อยู่ในตัว CPU ซึ่งจะกล่าวรายละเอียดดังต่อไปนี้

2.2 หลักการทำงานของ CPU

หน่วยประมวลผลกลาง (Central Processing Unit : CPU) เป็นอุปกรณ์ที่ทำหน้าที่ปฏิบัติงานประมวลผลข้อมูลเลขฐานสอง ทำหน้าที่คิด ตัดสินเงื่อนไขต่างๆ ตามรูปแบบของคำสั่งที่ CPU ได้รับ โดยที่การปฏิบัติงานนั้นต้องไม่เกินความสามารถของ CPU

สามารถแบ่งออกเป็นส่วนสำคัญ ได้ดังผังการทำงานต่อไปนี้



รูปที่ 2.1 ผังการทำงานของ CPU เบื้องต้น

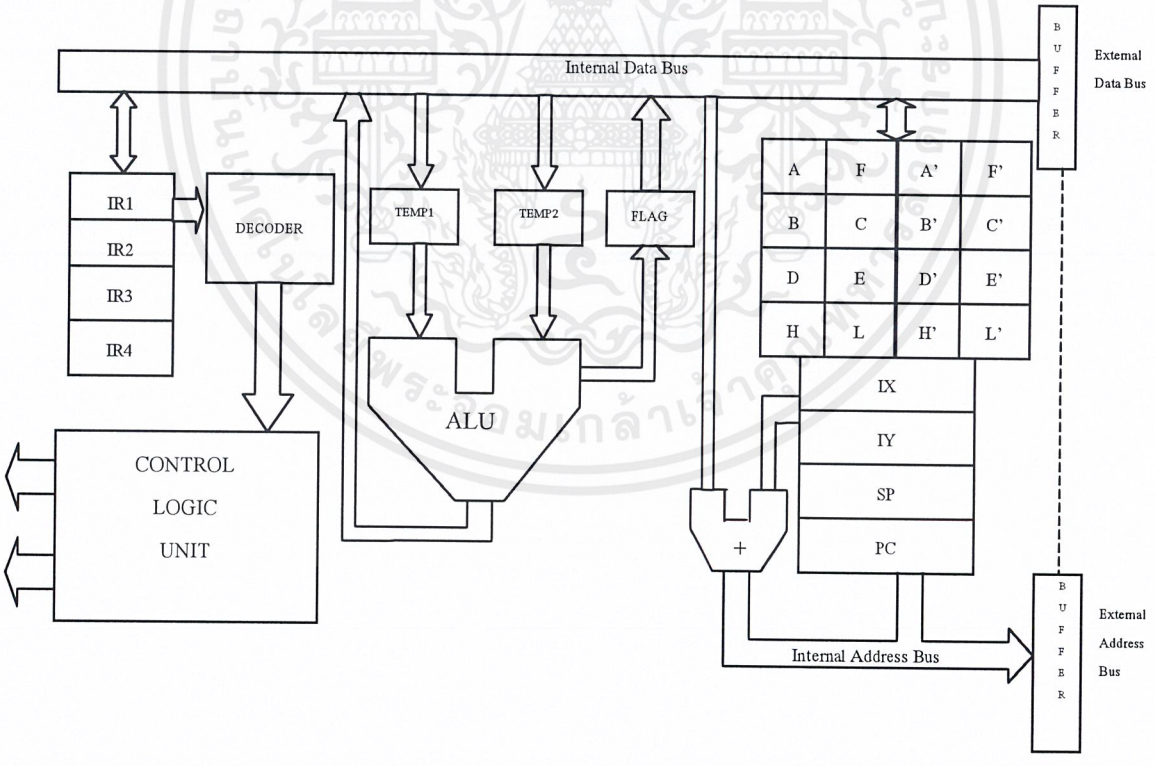
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การทำงานของ CPU มีอยู่ 2 สภาวะ ดังนี้

1) การเฟ็ทซ์ เป็นการนำรหัสคำสั่งซึ่งเป็นข้อมูลในรูปเลขฐานสองที่เก็บอยู่ในหน่วยความจำภายนอกเข้ามาเก็บที่อินสตรัคชั่นรีจิสเตอร์ซึ่งอยู่ภายในตัว CPU แล้วทำการแปลรหัสคำสั่ง (decode) ซึ่งเป็นข้อมูลเลขฐานสองที่เก็บอินสตรัคชั่นรีจิสเตอร์ว่าเป็นคำสั่งที่ต้องการให้ CPU ทำงานอะไร

2) การเอ็กซ์ซีคิว เป็นการกระทำคำสั่งตามที่ได้เฟ็ทซ์เข้ามาเก็บภายใน CPU กระบวนการนี้ CPU จะทำการสร้างสัญญาณขึ้นมาควบคุมการทำงานภายใน CPU ส่วนต่างๆ เพื่อให้ได้ผลลัพธ์ตามคำสั่งที่ CPU ได้รับ เมื่อทำคำสั่งนั้นเสร็จแล้ว CPU ก็มาทำการเฟ็ทซ์คำสั่งใหม่เข้ามาอีก และทำการเอ็กซ์ซีคิวเพื่อกระทำคำสั่งตามที่ได้รับเข้ามา ซึ่ง CPU ก็มีลักษณะการทำงานซ้ำๆ อยู่เช่นนี้

CPU เบอร์ Z80 เป็น CPU เบอร์หนึ่งที่ได้รับความนิยมเป็นอย่างสูง สำหรับผู้ที่เริ่มต้นในการเรียนรู้การทำงานของไมโครโปรเซสเซอร์ ซึ่ง CPU เบอร์นี้มีโครงสร้างภายใน และการใช้งานที่ง่ายต่อการศึกษาทำความเข้าใจเกี่ยวกับการทำงานของระบบไมโครโปรเซสเซอร์



รูปที่ 2.2 ผังการทำงานของ CPU Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ส่วนประกอบของ CPU Z80 มีขาดังนี้

1) รีจิสเตอร์แอคคิวมูเลเตอร์ ทำหน้าที่เก็บตัวตั้ง เพื่อใช้งานในคำสั่งที่เกี่ยวข้องกับ ALU (Arithemetic Logic Unit)แล้วใช้เก็บค่าผลลัพธ์ที่ออกจาก ALU ลงในแอคคิวมูเลเตอร์

2) รีจิสเตอร์แฟลก ทำหน้าที่เก็บค่าสถานะของคำตอบที่ได้จากการกระทำคำสั่งที่ผ่าน ALU

3) รีจิสเตอร์ B, C, D, E, H, L ทำหน้าที่เก็บข้อมูลขนาด 8 บิตที่อยู่ใน อินเตอร์นอลดาต้าบัส

4) รีจิสเตอร์ A' B' C' D' E' F' H' L' ใช้เป็นรีจิสเตอร์สำรอง เก็บข้อมูลขนาด 8 บิตที่อยู่ในรีจิสเตอร์หลัก B,C,D,E,H,L

5) รีจิสเตอร์ I (Interrupt Register) เป็นรีจิสเตอร์ที่ทำหน้าที่ เก็บข้อมูลเลขฐานสอง 8 บิตบนของการ Interrupt Mode 2

6) รีจิสเตอร์ R เป็นรีจิสเตอร์ทำหน้าที่เก็บค่าในการรีเฟรซหน่วยความจำ DRAM ที่ต่ออยู่ภายนอก

7) รีจิสเตอร์ IX, IY เป็นอินเด็กซ์รีจิสเตอร์ ทำหน้าที่ช่วยในการกำหนดแอดเดรสของหน่วยความจำให้คล่องตัวยิ่งขึ้น

8) โปรแกรมเคาน์เตอร์ (PC) ทำหน้าที่ชี้ค่าตำแหน่งที่อยู่ของข้อมูลในหน่วยความจำภายนอกที่จะทำการเฟ็ทซ์คำสั่งครั้งถัดไป

9) สแตกพอยน์เตอร์ (SP) ทำหน้าที่ชี้ตำแหน่งที่อยู่ของข้อมูลที่นำมาพักไว้ที่หน่วยความจำภายนอก

10) ดาต้าอินเอาท์บัฟเฟอร์ เป็นส่วนที่ควบคุมสัญญาณที่เข้าออกตัว CPU

11) แอดเดรสบัสบัฟเฟอร์ เป็นส่วนที่ทำหน้าที่ควบคุมสัญญาณแอดเดรสที่ส่งออกจาก CPU

12) แลตช์ ใน CPU Z80 ภายในมีวงจรแลตช์ 2 วงจร หรือเรียกว่า Temp1, Temp2 เพื่อพักตัวตั้ง และตัวกระทำก่อนเข้าอินพุตของ ALU

13) ALU ทำหน้าที่ทำคำสั่งทางคณิตศาสตร์ และกระทำทางลอจิก

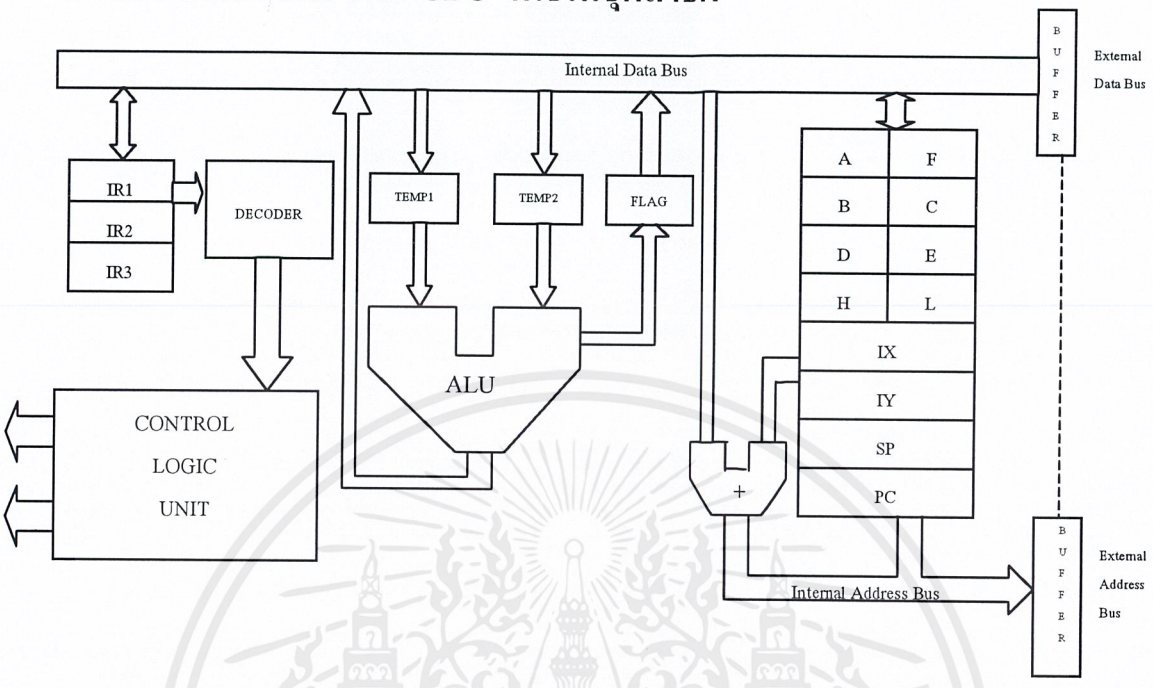
14) อินสตรัคชั่นรีจิสเตอร์ ทำหน้าที่เก็บรหัสคำสั่งจากการเฟ็ทซ์จากหน่วยความจำภายนอก

15) อินสตรัคชั่นดีโค้ด ทำหน้าที่แปลรหัสคำสั่งที่ได้รับการเฟ็ทซ์ ว่าคำสั่งนั้นเป็นคำสั่งที่ต้องการให้ CPU ทำหน้าที่อะไร

16) คอนโทรลยูนิต ทำหน้าที่กำเนิดสัญญาณความคุมส่วนต่างๆ ของ CPU ให้ได้ตามคำสั่งที่ CPU ได้รับเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถาปัตยกรรมภายใน CPU ที่ใช้ในชุดสาธิต



รูปที่ 2.3 สถาปัตยกรรมภายใน CPU ที่ใช้ในชุดสาธิต

ลักษณะสถาปัตยกรรมภายใน CPU นี้เป็นสถาปัตยกรรมที่ชื่อว่าฮาร์วาร์ด (Harvard) ซึ่งมีบัสภายใน 2 ชุด คือ บัสข้อมูล (ดาต้าบัส) และบัสตำแหน่ง (แอดเดรสบัส) แยกกัน

2.4 ส่วนประกอบของชุดสาธิตสถาปัตยกรรมภายใน CPU

แบ่งได้เป็น 5 ส่วนดังนี้คือ

2.4.1 รีจิสเตอร์อาเรย์ โดยแบ่งรีจิสเตอร์ได้เป็น 2 กลุ่ม คือ กลุ่มที่ 1

กลุ่มที่ 1 เป็นรีจิสเตอร์เดี่ยวขนาด 8 บิต โดยรีจิสเตอร์แต่ละตัวมีรายละเอียดดังนี้

ตารางที่ 2.1 กลุ่มรีจิสเตอร์ขนาด 8 บิต

A	F
B	C
D	E
H	L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) รีจิสเตอร์ A (แอคคิวมูเลเตอร์) เป็นรีจิสเตอร์หลักที่ใช้สำหรับพักข้อมูลเข้า และออกจาก ALU เมื่อ CPU ทำการประมวลผลคำสั่งทางคณิตศาสตร์หรือกระทำทางลอจิก CPU จะเก็บค่าตัวตั้งไว้ในแอคคิวมูเลเตอร์ และเมื่อ CPU ประมวลผลคำสั่งทางคณิตศาสตร์ และกระทำทางลอจิกเสร็จจะเก็บค่าคำตอบที่ได้ไว้ในแอคคิวมูเลเตอร์ และเป็นรีจิสเตอร์ A ยังใช้เป็นตัวหลักสำหรับการพักข้อมูลที่เข้าและออกจากตัว CPU ด้วย

2) รีจิสเตอร์ F (แฟลก) เป็นรีจิสเตอร์ที่ทำหน้าที่เก็บ และแสดงถึงสภาวะของผลลัพธ์ที่ CPU ได้ทำการประมวลผลคำสั่งทางคณิตศาสตร์ และกระทำทางลอจิก ซึ่งประกอบด้วย

2.1) แคร์รีแฟลก มีการทำงานสองลักษณะงาน คือ

แคร์รีแฟลกจะปรับเป็น 1 เมื่อผลลัพธ์ของเลขฐานสองสองจำนวนที่บวกกันแล้วเกิดมีค่าเกินตัวหลักที่แอคคิวมูเลเตอร์จะรับไว้ได้ เช่น

เลขฐานสอง	เลขฐานสิบหก
11111100	FC
+ 10000000	+ 80
101111100	17C

จะเห็นได้ว่าคำตอบมีการเกินจากบิตที่ 8 มา 1 ทำให้แคร์รีแฟลกมีค่าเป็น 1 ถ้าไม่มีค่าเกินจากบิตที่ 8 ตัวแคร์รีแฟลกจะมีค่าเป็น 0

ซึ่งบางคำสั่งเช่น ADD With Cary เมื่อต้องบวกตัวต่อไป คำสั่งนี้ก็จะเอาแคร์รีแฟลกรวมเข้าด้วยกัน

ใช้เป็นบิตตัวที่เก้า เมื่อต้องใช้ในการเลื่อนบิตหรือหมุนบิต และคำสั่งที่ต้องการจะนำมาบวกกับผลลัพธ์ด้วย เช่น กรณีการคูณหรือหารเลขฐานสอง

2.2) แฟลกเครื่องหมาย (N) หรือ ไซน์แฟลก (S)

S บิตจะมีผลโดยตรงกับตำแหน่ง MSB ของผลลัพธ์ที่ได้ เช่น ในแอคคิวมูเลเตอร์โดยค่าที่เป็น 1 จะแสดงถึงการเป็นเครื่องหมาย โดยทั่วไปไมโครโปรเซสเซอร์จะไม่สามารถตรวจสอบในบิตต่าง ๆ ของแอคคิวมูเลเตอร์ยกเว้นบิตที่ 7 หรือ MSB ซึ่งสามารถจะตรวจสอบในรูปของบวกหรือลบได้ และสามารถจะนำมาใช้ในการตรวจสอบข้อมูลเข้า และออกโดยวิธีการทางคณิตศาสตร์ และตรวจสอบผลที่ได้ก็จะรู้ถึงข้อมูลที่เข้า และออก

2.3) โอเวอร์โฟลแฟลก (v)

จะปรับเป็น 1 เช่น ตัวเลขที่ได้จากการคำนวณออกมามีค่ามากกว่าค่าที่รีจิสเตอร์นั้นสามารถเก็บได้โดยเฉพาะกรณีที่ใช้บิตสุดท้ายหรือ Most Significant Bit (MSB) เป็นการแสดงเครื่องหมายบวกลบหรือเรียกว่า ไซน์บิต คือ 1 เป็น ลบ และ 0 เป็นบวก เมื่อสองตัวเลขที่เป็นรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบทูคอมพลีเมนต์มาบวกกันตัวทดที่ได้จากการบวกหรือลบก็อาจจะไปทดเข้าบิตไซน์ในกรณีเช่น นี้ก็อาจจะไปเปลี่ยนแปลงค่าที่ควรจะเป็นแทนที่จะเป็นลบก็กลายเป็นบวกเป็นต้นจึงใช้โอเวอร์โฟลในกรณีตัวเลขของคณิตศาสตร์ทูคอมพลีเมนต์เกิดการเปลี่ยนแปลงพิจารณาตัวอย่างการใช้บิต

MSB เป็นตัวแสดงค่าบวกลบของสองตัวเลข เช่น

ตัวเลขฐานสิบหก	ตัวเลขฐานสอง
58	0101 1000
+61	+ 0110 0001
-39	1100 1001
	Sign bit บิตสูงสุดของตัวเลข

จะเห็นว่าเมื่อนำมาบวกกัน จะเกิดลักษณะตัวทดจากบิตสูงสุดของตัวเลขทดเข้าสู่ตัวบิตที่แสดงเครื่องหมายซึ่งเป็นการแสดงถึงค่าที่ได้มีขานดเกินค่าสูงสุดที่จะรับได้ ลักษณะเช่นนี้ผลลัพธ์ที่ได้จะแสดงค่าผิดพลาด ซึ่งจะแสดงที่รีจิสเตอร์แฟลก ตัวนี้ที่บิต โอเวอร์โฟล

ตัวเลขฐานสิบหก	ตัวเลขฐานสอง
(-2)	11111110
+(-2)	+11111110
+(-4)	111111100
	CS = 1 CP = 1

ตัวอย่างนี้เมื่อเอาตัวเลขแบบทูคอมพรีเมนต์ทั้งสองตัวเลขมาบวกกันจะเห็นตัวเลขตัวที่สองลักษณะคือ เกิดตัวทดจากบิตสูงสุดของตัวเลขทดเข้าสู่ค่าบิตที่แสดงเครื่องหมาย กับเกิดตัวทดเข้าสู่บิตตัวทดของรีจิสเตอร์แฟลก ด้วย จะเห็นว่าค่าผลลัพธ์ที่ได้จะเป็นค่าทูคอมพรีเมนต์แสดงเป็นค่าลบจากการเกิดลักษณะตัวทดเช่นนี้ เราจะใช้ตัวแทนด้วย CS = 1 ถ้าเกิดตัวทดออกจากบิตที่แสดงเครื่องหมายเข้าสู่บิตตัวทดของแฟลก และใช้ CP = 1 ถ้าเกิดตัวทดออกจากบิตค่าสูงสุดของตัวเลขเข้าสู่บิตที่แสดงเครื่องหมาย ซึ่งแสดงว่าสามารถจะเกิดได้สี่กรณีด้วยกัน ดังตัวอย่างต่อไปนี้

ตัวเลขฐานสิบหก	ตัวเลขฐานสอง	ตัวเลขฐานสิบหก	ตัวเลขฐานสอง
(-28)	11011000	(-39)	11000111
+59	+01011001	(-1A)	11100110
+31	00110001	(-53)	10101101
CS = 0 CP = 0		CS = 1 CP = 1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเลขฐานสิบหก	ตัวเลขฐานสอง	ตัวเลขฐานสิบหก	ตัวเลขฐานสอง
+02	00000010	(-0C)	11110100
+02	00000010	(-09)	00001001
(+04)	00000100	-03	11111101
CS = 0 CP = 0		CS = 0 CP = 0	
+45	01000101	(-6E)	10000010
+67	01100111	(-5C)	10100100
(-54)	10101100	+36	00110110
CS = 0 CP = 1		CS = 1 CP = 0	

จากตัวอย่างที่ให้มาทั้งหมดนี้จะเห็นว่า CS กับ CP มีค่าเหมือนกันตัวเลขผลลัพธ์จะถูกต้อง คือไม่เกิดโอเวอร์โฟลแต่ถ้าเกิด CS กับ CP ไม่เหมือนกันผลลัพธ์ที่ออกมาจะเกิดค่าผิดพลาดคือ เกิดโอเวอร์โฟลดังนั้นใน CPU จึงนำเอาบิต CS กับ CP มาเข้าเกต XOR เพื่อตรวจสอบค่า โอเวอร์โฟล แสดงในรีจิสเตอร์ของแฟลกเป็นโอเวอร์โฟล = CS (+) CP

2.4) ฮาร์ฟแฟลก (H)

ในการใช้ BCD ตัวเลขฐานสิบในรูปของไบนารี่จะแทนตัวเลขฐานสิบ ตัวเลขแทนหลักละ 4 บิต เมื่อนำมาบวกกันก็อาจจะเกิดการทดไปที่บิตที่ 5 ได้ เป็นการทดจากหลักนัยต่ำสี่บิตแรกไปสู่หลักนัยสำคัญสูง ถ้าเกิดกรณีเช่นนี้ H บิตก็จะปรับเป็น 1 เช่น

	เลขฐานสอง	เลขฐานสิบ
	0101 1000	58
	0000 1001	09
= 1	0110 0111	67

2.5) แฟลกศูนย์ (Z)

Z บิตจะปรับเป็น 1 เมื่อผลที่ได้หลังการทำงาน (เอ็กซ์ซีคิว) ตามคำสั่งทางคณิตศาสตร์ในแอคคิวมูเลเตอร์เป็น 0 หมดทุกบิต เช่น ใช้ในการเปรียบเทียบ และตรวจสอบคำสั่งข้อมูลทางอินพุตและเอาต์พุต เพื่อใช้หาว่าบิตในแอคคิวมูเลเตอร์ หรือรีจิสเตอร์ตัวใดตัวหนึ่งจะเปลี่ยนแปลงหรือไม่ โดยใช้วิธีทางตรรกของ XOR ด้วยค่าของรีจิสเตอร์กับค่าก่อนที่จะทำงาน (เอ็กซ์ซีคิว) ตามคำสั่งต่อมา ถ้าเป็น 0 แสดงว่าไม่มีการเปลี่ยนแปลงถ้ามีการเปลี่ยนแปลงผลหลังการทำตรรก XOR จะไม่เป็น 0 ในบิตที่ถูกตรวจสอบด้วย 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6) แฟลกพาริตี้ (P)

P บิตโดยธรรมดาทั่วไปจะไม่มีในไมโครโปรเซสเซอร์รุ่นเก่า ๆ แต่ในไมโครโปรเซสเซอร์รุ่นใหม่จะมี P บิตซึ่งใช้ในการตรวจสอบในด้านการส่งข้อมูลไปว่าถูกต้องหรือไม่กับข้อมูลก่อนที่จะส่งไป โดยหลักการจะนับจำนวนของตัวเลข 1 ในแอคคิวมูเลเตอร์ หรือรีจิสเตอร์นั้น 8 บิตถ้านับเลข 1 ในจำนวน 7 บิตแรกที่แทนตัวอักษร และถ้าต้องการให้เป็นจำนวนคู่หรือคี่ก็เพียงใส่เลข 0 หรือ 1 ไปที่บิตที่แปดให้ได้เป็นเลขคู่หรือคี่ พาริตี้บิตก็จะใช้ในการชี้ว่าข้อมูลที่ส่งไปเป็นพาริตี้คู่หรือคี่เปลี่ยนแปลงจากเดิมหรือไม่ เช่น ถ้าส่งเป็นพาริตี้คู่แล้วตรวจได้เป็นคี่ก็จะแสดงว่าข้อมูลที่ส่งไปส่งผิดหรือเกิดความผิดพลาดเป็นพาริตี้เออเลอร์ส่วนมากจะใช้พาริตี้ตรวจสอบในกรณีที่ส่งไปข้างนอกระบบไมโครโปรเซสเซอร์

3) รีจิสเตอร์ทั่วไป B, C, D, E, H, L เป็นรีจิสเตอร์เดี่ยวขนาด 8 บิต ใช้สำหรับเก็บข้อมูลเลขฐานสอง จำนวน 8 บิต โดยสามารถนำมาใช้เป็นรีจิสเตอร์คู่ขนาด 16 บิตได้ด้วย โดยจับคู่ระหว่างรีจิสเตอร์ B กับ C และจับคู่ระหว่าง D กับ E และจับคู่ระหว่าง H กับ L รีจิสเตอร์B, C, D, E, H, L นี้เป็นรีจิสเตอร์ที่เหมาะสำหรับใช้งานเก็บข้อมูล 8 บิตทั่วไป

ตารางที่ 2.2 กลุ่มรีจิสเตอร์ขนาด 16 บิต

IX
IY
SP
PC

4) อินเด็กซ์รีจิสเตอร์ (IX, IY) เป็นรีจิสเตอร์ขนาด 16 บิตใช้ในอินเด็กซ์แอดเดรสซิ่งโหมดโดยเอฟเฟ็กท์ทีฟแอดเดรสซิ่งซึ่งเป็นแอดเดรส ที่ชี้ไปยังเมมโมรี่นั้น จะเกิดจากเอาแอดเดรสซิ่ง ใน IX หรือ IY รวมกับดิสเพรสเมนต์แอดเดรส เช่น

LD (IX+d), n เป็นการนำค่าเลขฐานสองขนาด 8 บิตใดๆ ไปเก็บยังตำแหน่งที่อยู่ของข้อมูลที่ ค่า IX + d ชี้อยู่ เพื่อประโยชน์ในการชี้ข้อมูลที่เก็บอยู่ได้หลายตารางทำให้การทำกระบวนการข้อมูลเป็นไปอย่างรวดเร็ว และประหยัดเวลาทำงานตามโปรแกรมคำสั่ง

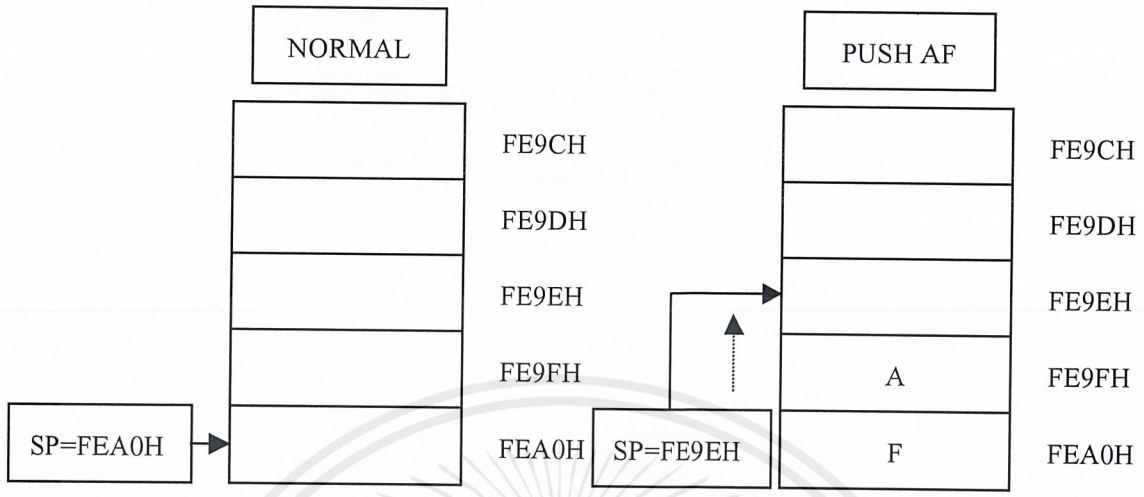
5) โปรแกรมเคาน์เตอร์ (PC) จะทำหน้าที่เก็บตำแหน่งที่ชี้ลำดับการทำงานของคำสั่ง (Instruction) หรือโปรแกรมตัวถัดไปที่เก็บอยู่ ณ ตำแหน่งนั้น ในขณะที่ให้โปรแกรมทำงานตามลำดับที่ต้องการได้ ซึ่งจะมี 16 บิต สำหรับ CPU ที่มีข้อมูลขนาด 8 บิต (Bit Word) สัญญาณวัฏจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

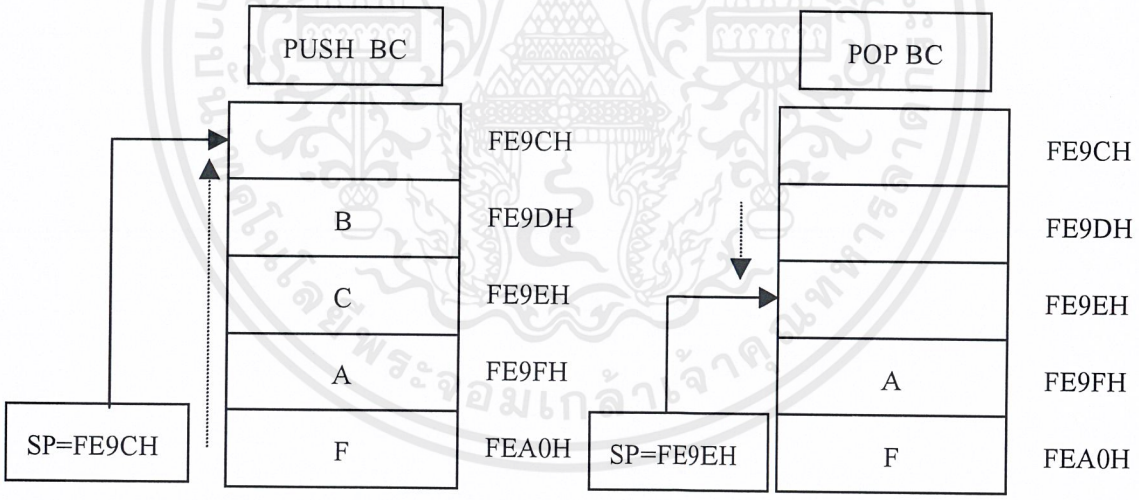
คำสั่ง (Instruction Cycle) จะเริ่มด้วยการที่ CPU จะส่งข้อมูล ตำแหน่งที่เก็บคำสั่งตัวแรกด้วย PC และจะส่งผ่านออกสู่บัสแอดเดรสเพื่อที่ CPU จะได้อ่านหรือ แฟทช์ (Fetch) การแฟทช์หมายถึง การที่ CPU อ่านข้อมูลจากตำแหน่งที่ PC ได้ชี้กำหนดไว้ หลังจากข้อมูลเข้า CPU PC ก็จะเพิ่มค่าตำแหน่งอีกหนึ่งหน่วยโดยอัตโนมัติ เพื่อที่จะดึงเอาข้อมูลของคำสั่งที่จะให้ CPU ทำอะไรต่อไปได้ตามโปรแกรมนั้นๆ บางครั้งคำสั่งต้องใช้ข้อมูลคำสั่งมากกว่าหนึ่งตำแหน่งที่อยู่ในหน่วยความจำ ณ ตำแหน่งนั้นๆ PC ของ CPU ก็จะเพิ่มหน่วยขึ้นทีละหน่วยจนกว่าจะครบข้อมูลคำสั่งที่ CPU รีจิสเตอร์นั้น ๆ จะเข้าใจ และทำงานตามคำสั่งหลายตำแหน่งหรือไบต์ (Byte) นั้นโดยปกติ CPU จะทำงานตามลำดับที่ละขั้นตอนตามคำสั่งตลอดเวลาที่ไม่มีการสั่งเปลี่ยนตำแหน่งหรือค่าข้อมูลที่อยู่ใน PC ด้วยคำสั่งของ JUMP

6) สแตกพอยน์เตอร์ (SP) ไมโครโปรเซสเซอร์ส่วนใหญ่จะต้องใช้สแตกพอยน์เป็นตัวเก็บข้อมูลของรีจิสเตอร์ตัวนับโปรแกรม (PC) เมื่อการทำงานตามคำสั่ง CALL เป็นการเปลี่ยนแปลงค่าของ PC ซึ่งก่อนจะเปลี่ยนแปลงค่าของ PC ไปยังตำแหน่งของโปรแกรมย่อย (SUBROUTINE) จะต้องเก็บตำแหน่งของคำสั่งถัดมาจากคำสั่ง CALL ไปที่บริเวณสแตกก่อน ทั้งนี้เมื่อไมโครโพรเซสเซอร์สิ้นสุดการทำงานโปรแกรมย่อยแล้ว หลังการทำคำ สั่ง RET ก็จะคืนกลับมายังตำแหน่งเดิมได้ด้วยการเปลี่ยนแปลงค่า PC จากการ POP เอาค่า PC เดิมจากบริเวณสแตกมาไว้ที่ PC เช่น เดียวกับการอินเตอร์รัพต์ โปรแกรมด้วยฮาร์ดแวร์ เพื่อที่จะนำเอาข้อมูลเหล่านี้มาใช้ใหม่หลังจากอินเตอร์รัพต์เสร็จเรียบร้อยแล้ว และต้องการจะทำงานต่อจากลำดับหลังก่อนที่จะมีการอินเตอร์รัพต์โดยสแตกจะเป็นแบบ Last-in, First-Out (LIFO'S) Memory และตัวชี้สแตกหมายถึงข้อมูลต่างๆที่เก็บอยู่ในหน่วยความจำที่กำหนดตำแหน่งด้วยตัวชี้สแตกถือว่าเป็นบริเวณของสแตกและจะเป็นตัวชี้ตำแหน่งที่อยู่ของข้อมูลของรีจิสเตอร์ต่าง ๆ ที่เก็บอยู่ และสามารถจะทำงานในโปรแกรมด้วยคำสั่ง PUSH กรณีใส่ข้อมูลเข้าไปในบริเวณของสแตก และPOP เมื่อจะอ่านข้อมูลจากหน่วยความจำบริเวณของสแตกโดยหลัง PUSH แล้ว SP ก็จะเพิ่มขึ้นหนึ่ง (Increment) ตำแหน่งโดยอัตโนมัติ ทำนองเดียวกัน POP เมื่อเกิดแล้วก็จะลดลงหนึ่ง (Decrement)ตำแหน่งเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเลื่อนข้อมูลในรีจิสเตอร์ A, F เข้าในหน่วยความจำสแตก



รูปที่ 2.5 การเลื่อนข้อมูลรีจิสเตอร์ B, C เข้าออกในหน่วยความจำสแตก

2.4.2 หน่วยคำนวณทางคณิตศาสตร์ และกระทำลอจิก (Arithmetic and Logic Unit)

ทำหน้าที่คำนวณทางคณิตศาสตร์ และกระทำทางลอจิก เช่น

1) การบวก (Addition)

2) การลบ (Subtraction)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การกระทำทางลอจิกแอน (Logical AND)

4) การกระทำทางลอจิก (Logical OR)

5) การกระทำทางลอจิกเอ็กคลูซีพออร์ (Logical Exclusive-OR)

6) การคอมพลีเมนต์ (Logical NOT)

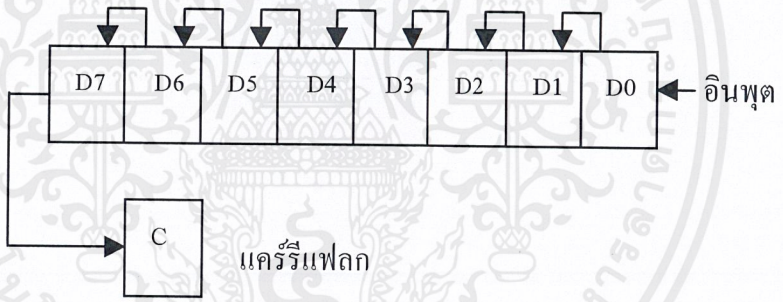
7) การเพิ่มค่าขึ้น 1 ค่า (Increment)

8) การลดค่าลง 1 ค่า (Decrement)

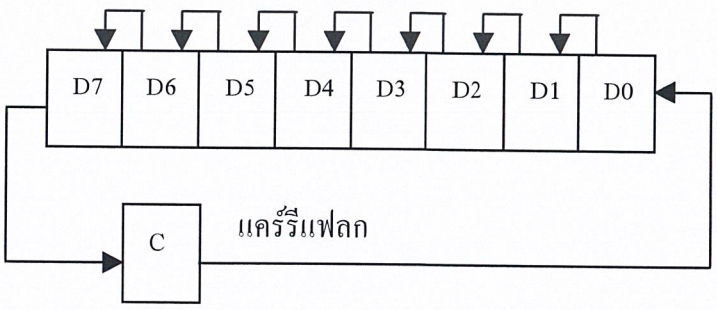
9) การเลื่อนบิตซ้ายขวา (Left, Right Shift)

10) การหมุนบิตซ้ายขวา (Left, Right Rotate)

ALU มีอินพุต 2 ทางคือ อินพุตด้านหนึ่งจะมีรีจิสเตอร์ขนาด1ไบต์ เรียกว่า แอคคิวมูเลเตอร์ ซึ่งเก็บข้อมูลที่จะนำมาคำนวณที่หน่วย ALU และเมื่อได้ค่าผลลัพธ์แล้วผลลัพธ์นั้นก็ถูกนำไปเก็บในแอคคิวมูเลเตอร์ เช่นกัน ALU ปกติสามารถที่จะทำการ เลื่อนข้อมูล (Shift Data), หมุนข้อมูล (Rotate Data) ได้ ซึ่งการเลื่อนข้อมูล หรือ การหมุนข้อมูลนี้จะกระทำผ่านแฟลกตัวทด



รูปที่ 2.6 การเลื่อนบิตไปทาง ซ้ายมือ

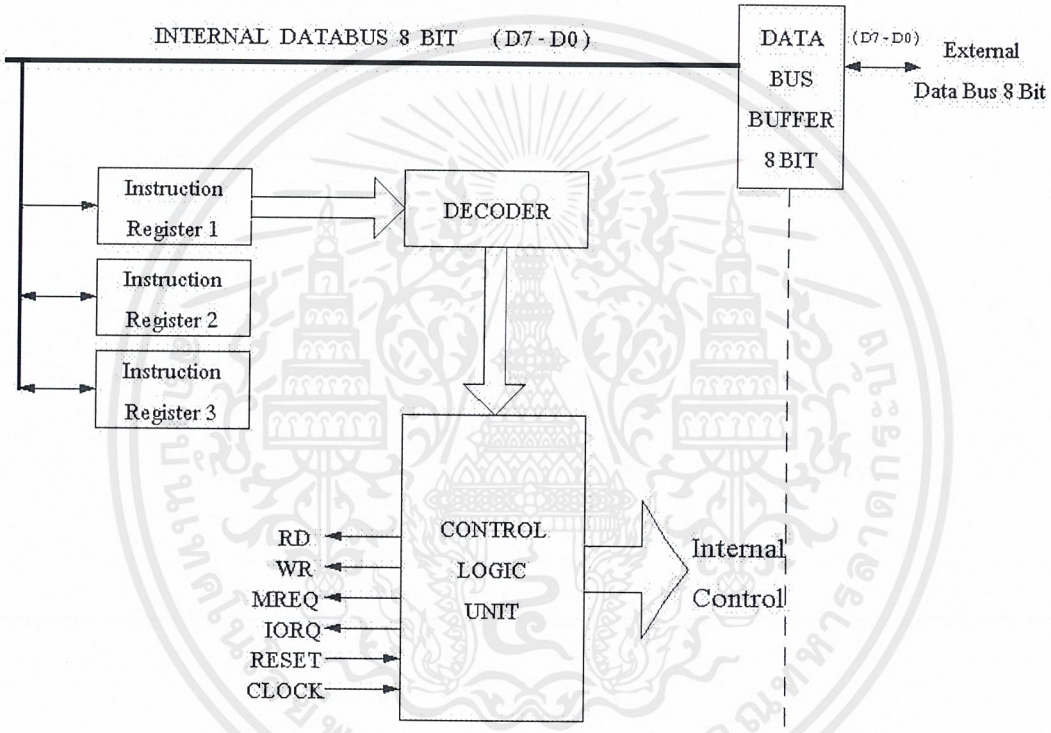


รูปที่ 2.7 การหมุนข้อมูลไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 อินสตรัคชั่นรีจิสเตอร์ และ คอนโทรลลอจิกยูนิต

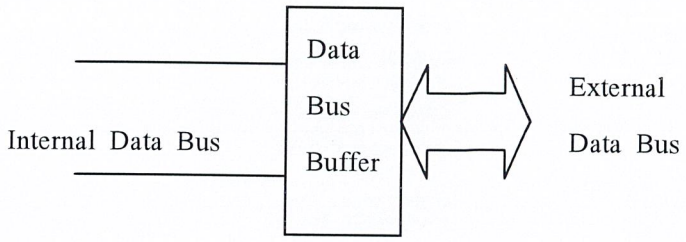
อินสตรัคชั่นรีจิสเตอร์ (IR) เป็นรีจิสเตอร์ขนาด 8 บิต3ตัวโดยเก็บคำสั่งตั้งแต่ความยาว 1- 3 ไบต์ทำงานโดยเมื่อ เฟ็ทซ์คำสั่งเข้ามาใน CPU คำสั่งไบต์แรกจะถูกเก็บยัง IR 1 และคำสั่งไบต์แรกยังถูกส่งไปยังอินสตรัคชั่นรีจิสเตอร์เพื่อถอดรหัสคำสั่งว่าเป็นคำสั่งให้ CPU จะต้องทำหน้าที่อะไร หน่วยคอนโทรลลอจิก ทำหน้าที่สร้างสัญญาณควบคุมการทำงานภายใน และภายนอก CPU ตามรหัสคำสั่งที่ได้รับการถอดรหัสจากอินสตรัคชั่นดีโค้ด



รูปที่ 2.8 คอนโทรลลอจิก และอินสตรัคชั่นรีจิสเตอร์

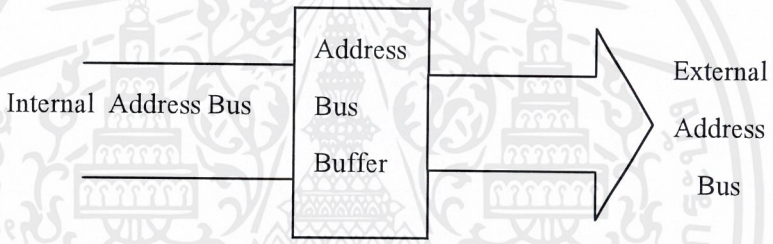
2.4.4 ดาต้าบัสบัฟเฟอร์ เป็น ไตรสเตท ขนาด 8 บิต ชนิด 3 สภาวะที่ใช้แยกดาต้าบัสภายใน CPU กับดาต้าบัสข้อมูลภายนอก CPU ซึ่งข้อมูลจะสามารถเข้าออกได้สองทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ดาต้าบัสบัฟเฟอร์

2.4.5 แอดเดรสบัสบัฟเฟอร์ เป็นไตรสเตทขนาด 16 บิต ทำหน้าที่เก็บตำแหน่งที่ใช้ชี้ตำแหน่งที่อยู่ของข้อมูลที่อยู่ในหน่วยความจำภายนอกทั้ง หน่วยความจำโปรแกรม และหน่วยความจำข้อมูล ซึ่งแอดเดรสบัสบัฟเฟอร์นี้ข้อมูลสามารถออกได้ทางเดียว



รูปที่ 2.10 แอดเดรสบัสบัฟเฟอร์

2.5 วิธีการอ้างตำแหน่งที่อยู่ข้อมูลของ CPU

วิธีการอ้างตำแหน่งที่อยู่ข้อมูล ในส่วนหน่วยความจำหรือรีจิสเตอร์ มี 10 วิธี ดังนี้

2.5.1 อิมพรายซ์แอดเดรสซิ่ง

การแอดเดรสชนิดนี้ รหัสคำสั่ง (Operation Code) จะเป็นแบบรหัสแน่นอน (Fix) ไม่มีส่วนที่แปรค่าได้ (Vary Field) เช่นคำสั่ง CPL, LD SP, IY LD I,A LD A, I และ RET เป็นต้น

เช่นคำสั่ง CPL : เป็นคำสั่ง คอมพลีเมนต์ ค่าในแอคคิวมูเลเตอร์

0010 1111	= 2FH (Opcode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง LD SP, IY : เป็นคำสั่งที่นำค่าใน IY ไปเก็บใน SP

1111 1101	= FDH (Opcode)

1111 1001	= F9H (Opcode)

2.5.2 อิมมีเดียสแอดเดรสซิ่ง

การแอดเดรส ชนิดนี้ข้อมูลจะเป็นส่วนประกอบอยู่ ในคำสั่ง โดยไบต์ที่ 2 หรือ 3 จะเป็นโอเปอร์แรนด์ มีประโยชน์ใช้ในการคำนวณ หรือปฏิบัติทางลอจิกด้วยข้อมูลที่คงที่

ถ้าข้อมูลมี 8 บิต จะถูกเก็บใน ไบต์ที่ 2 ของคำสั่ง

ถ้าข้อมูลมี 16 บิต 8 บทล่างอยู่ในไบต์ที่ 2 ของคำสั่ง และ 8 บิตบนจะอยู่ในไบต์ที่ 3 ไบต์สูงอยู่ที่หน่วยความจำตำแหน่งสูงกว่าไบต์ต่ำ

1) Add A, n : เป็นคำสั่งนำค่า N บวกกับค่าในรีจิสเตอร์ A (แอกคิวมูเลเตอร์)

1100 0110	Opcode = C6H
N	Operand

N : เป็นโอเปอร์แรนด์มีค่าตั้งแต่ 0-255 เรียกว่า อิมมีเดียสแวร์รู

ตัวอย่าง Add A, 30H

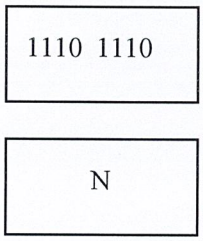
ก่อน A =	70	0111 0000 (70H)
		0011 0000 + (30H)
หลัง A =	A0	1010 0000 (A0H)

หลังปฏิบัติคำสั่งนี้แอกคิวมูเลเตอร์จะมี A0H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2) XOR N เป็นคำสั่งนำค่า N เอ็กคลูซีพออร์ กับค่าใน A



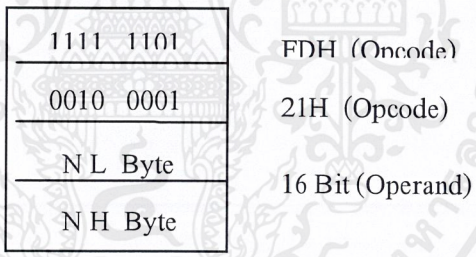
ตัวอย่าง XOR F0H

ก่อน A = F0H

หลัง A = 00

2.5.3 เอ็กซ์เทนเดทอิมมีเดียสแอดเดรสซิ่ง

คล้ายกับคำสั่งแบบ อิมมีเดียสแอดเดรสซิ่งแต่เป็นพวกที่ต้องการโอเปอร์แรนด์ขนาด16 บิต ตัวอย่าง LD IY, nn เป็นการนำค่า nn 16 บิต ไปเก็บไว้ใน IY เป็นคำสั่งขนาด 4 ไบต์



2.5.4 รีจิสเตอร์แอดเดรสซิ่ง

การแอดเดรสวิธีนี้รหัสคำสั่งที่ระบุถึงรีจิสเตอร์เดี่ยวขนาด 8 บิตภายใน CPU เช่น And R, Add R เป็นต้น

ตัวอย่าง And R เป็นคำสั่งที่นำค่าในรีจิสเตอร์ที่ระบุโดย R ไป And กับค่าใน A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

44211

2.5.5 รีจิสเตอร์อินไดร์เร็กซ์แอดเดรสซิ่ง

การแอดเดรสแบบนี้จะใช้ รีจิสเตอร์คู่ขนาด 16 บิต BC, DE, HL ในการชี้ข้อมูลในหน่วยความจำ เช่น INC (HL) เป็นคำสั่งเพิ่มค่าภายในหน่วยความจำตำแหน่งที่ชี้โดยรีจิสเตอร์คู่ HL

0 0 1 1 0 1 0 0	= 34H (Opcode)

2.5.6 เอ็กซ์เทนเดทแอดเดรสซิ่ง หรือไดร์เลคแอดเดรสซิ่ง

การแอดเดรสแบบนี้นั้น คำสั่งจะมีแอดเดรสอยู่ด้วย ดังนั้น คำสั่งพวกนี้จะมีหลายไบต์ ทำให้การแอดเดรสหน่วยความจำได้มากขึ้น

ตัวอย่าง LD A, (NN) เป็นคำสั่งนำค่าในหน่วยความจำตำแหน่งที่ชี้โดย NN ไปเก็บไว้ในรีจิสเตอร์ A

3A	Opcode
N L Byte	16 Bit Address
N H Byte	16 Bit Address

2.5.7 โมดิไฟท์เพจซีโร่แอดเดรสซิ่ง

การแอดเดรส แบบนี้ใช้สำหรับคำสั่ง RST P (Restart Page Zero) คำสั่งนี้จะกระโดดไปยัง Page 0 ซึ่งแบ่งเป็นมี 8 ไบต์ (256 ไบต์เรียกว่า 1 Page)

การแอดเดรสแบบนี้ จะสังเกตว่าประหยัดจำนวนไบต์ คือใช้ไบต์เดียวซึ่งปกติการอ้างหน่วยความจำต้องใช้หลายไบต์ เช่น คำสั่ง RST p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตำแหน่งการจบโปรแกรม

11	T	111	Opcode

P	T
00H	000
08H	001
10H	010
18H	011
20H	100
28H	101
30H	110
38H	111

2.5.8 รีเรย์ทีฟแอดเดรสซิ่ง

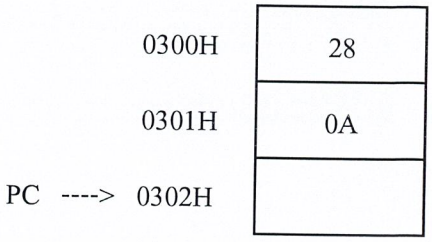
เป็นการแอดเดรสชนิดนี้จะประหยัดเนื้อที่ของหน่วยความจำ คือ สามารถแอดเดรสโดยใช้เพียง 2 ไบต์ (รวมรหัสคำสั่ง) แทนที่จะใช้ 3 หรือ 4 ไบต์ เหมือนคำสั่งอื่นๆ ส่วนใหญ่จะเป็นคำสั่งกระโดดไปตำแหน่งต่างๆ โดยจะสัมพันธ์ กับตำแหน่งของโปรแกรมเคาน์เตอร์

Opcode	
D	Operand

D = Displacement value มีค่าตั้งแต่ -128 ถึง + 127 คือใช้ 7 บิตส่วนบิตที่ 8 เป็นบิตเครื่องหมายบวกหรือลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง JR Z , 0AH

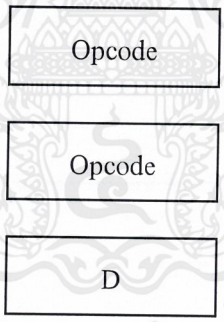


หลังจากปฏิบัติคำสั่ง JR Z, 0AH จะกระโดดไปตำแหน่ง 0302H + 0AH = 030CH ถ้า Z = 0 แต่ ถ้า Z ไม่เท่ากับ 0 จะทำตำแหน่ง 0302H ต่อไป

2.5.9 อินเด็กซ์แอดเดรสซิ่ง

เป็นการอ้างแอดเดรสโดยให้ค่าในรีจิสเตอร์ IX, IY 16 บิต รวมกับค่า ดีสเพรทเมนต์ (D) เป็น เอฟแฟกซ์ทีฟแอดเดรสเพื่อใช้ในการอ้างตำแหน่งติดต่อกับหน่วยความจำภายนอก

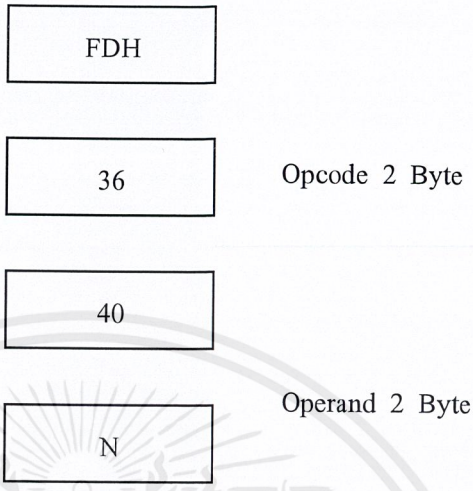
ตัวอย่าง



เอฟเฟ็กซ์ทีฟแอดเดรส = (IX+D) หรือ (IY+D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง LD (IY+ D), N เป็นคำสั่งนำค่า N (8 บิต) ไปเก็บที่หน่วยความจำที่ชี้โดย (IY+ d)



ถ้า IY = 1003H และ N=30H เมื่อปฏิบัติ LD (IY+40) , 30H

เอฟเฟ็กซ์ทีฟแอดเดรส = 1003H + 40H = 1043H

นำ N ไปเก็บที่ตำแหน่ง 1043H ซึ่งในที่นี้ N = 30H

2.5.10 บิตแอดเดรสซิ่ง

เป็นกลุ่มคำสั่งซึ่งมีประโยชน์มากใช้ในการ เซท และรีเซท หรือทดสอบ บิตต่างๆ ใน 8 บิต รีจิสเตอร์หรือ ในหน่วยความจำ เช่น คำสั่ง BIT B, R Set B, R และ Res 9, r เป็นต้น

เช่น SET 0, C เป็นคำสั่งที่ SET รีจิสเตอร์ C บิตที่ 0 เป็น 1

	ก่อน	หลัง
C	00011110	00011111

2.6 การทำงานภายใน CPU

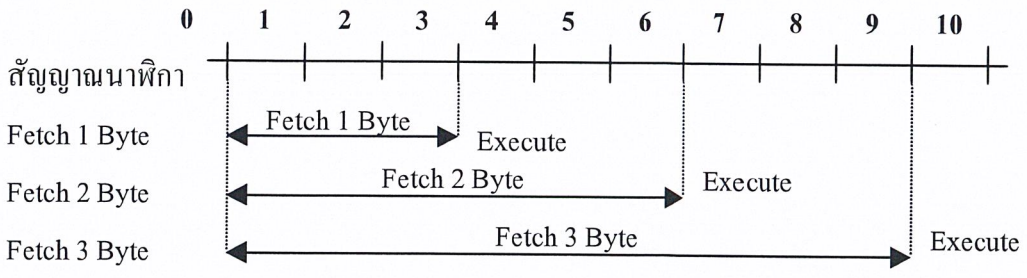
CPU นั้นมีการทำงานอยู่ 2 สภาวะคือการ เฟ็ทซ์ และเอ็กซ์ซีคิว

2.6.1 กรณีการเฟ็ทซ์ (นำข้อมูลคำสั่งจากหน่วยความจำภายนอกเข้า CPU)

ในครั้งแรกเมื่อ CPU ได้รับสัญญาณนาฬิกาลูกแรกเข้ามา CPU ก็จะเริ่มทำการเฟ็ทซ์โดย คอนโทรลลอจิกยูนิตจะสร้างสัญญาณสั่งให้ PC ส่งแอดเดรสที่ PC เก็บอยู่ออกนำข้อมูลส่งไปที่ อินเตอร์นอลแอดเดรสบัส และสร้างสัญญาณที่ขา MERQ, RD ให้เป็น 0 เพื่อเรียกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำให้อ่านข้อมูลออกมา และสั่งดาต้าบัสบัฟเฟอร์ให้รับข้อมูลจากเอ็กซ์เตอร์นอลแอดเดรสบัส เข้าสู่ อินเตอร์นอลดาต้าบัส ซึ่งจากสัญญาณนาฬิกาลูกแรกที่ CPU ได้รับข้อมูลที่เก็บอยู่ที่ตำแหน่ง PC ชี้อยู่จะถูกส่งเข้า อินเตอร์นอลดาต้าบัส ซึ่งเป็นข้อมูลเลขฐานสอง เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 2 เข้ามาข้อมูลที่อินเตอร์นอลดาต้าบัสจะถูกสั่งให้เก็บลง อินเตอร์นอลดาต้าบัสตัวที่ 1 และอินสตรัคชั่นดีโค้ดทำการแปลสัญญาณที่อินสตรัคชั่นรีจิสเตอร์ 1 เก็บอยู่ว่าเป็นคำสั่งอะไร และความยาวของคำสั่งนี้มีกี่ไบต์ เมื่อสัญญาณนาฬิกาลูกที่ 3 ถูกป้อนเข้าสู่ CPU คอนโทรลลอจิกยูนิต จะสร้างสัญญาณให้ โปรแกรมเคาท์เตอร์ เพิ่มค่าที่เก็บอยู่ขึ้น 1 ตำแหน่ง ขั้นตอนนี้ในกรณีที่ อินสตรัคชั่นดีโค้ด บอกว่าคำสั่งนี้ มีความยาวแค่ 1 ไบต์ เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 4 เข้ามา ก็จะไปทำการปฏิบัติตามคำสั่งที่ได้รับ แต่ถ้า อินสตรัคชั่นดีโค้ดบอกว่าคำสั่งนี้มีความยาว 2 ไบต์ เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 4 เข้ามาทำการเฟ็ทซ์ซึ่งกระทำเหมือนสัญญาณนาฬิกาลูกแรก ข้อมูลที่ตำแหน่งที่ PC ชี้อยู่ถูกส่งเข้า อินเตอร์นอลดาต้าบัสเมื่อสัญญาณนาฬิกาลูกที่ 5 เข้ามา คอนโทรลลอจิกยูนิตทำการเก็บค่าข้อมูลใน อินเตอร์นอลดาต้าบัสลงในอินสตรัคชั่นรีจิสเตอร์ 2 เมื่อสัญญาณนาฬิกาลูกที่ 6 ถูกป้อนเข้า CPU คอนโทรลลอจิกยูนิตจะทำการสร้างสัญญาณให้โปรแกรมเคาท์เตอร์เพิ่มค่าที่เก็บอยู่ขึ้น 1 ตำแหน่งขั้นตอนนี้ในกรณีที่ อินสตรัคชั่นดีโค้ดบอกว่าคำสั่งนี้ มีความยาวแค่ 2 ไบต์ เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 7 เข้ามา ก็จะไปทำการปฏิบัติตามคำสั่งที่ได้รับแต่ถ้าอินสตรัคชั่นดีโค้ดบอกว่า คำสั่งนี้มีความยาว 3 ไบต์ เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 7 เข้ามาก็ทำการเฟ็ทซ์ซึ่งกระทำเหมือนสัญญาณนาฬิกาลูกแรก ข้อมูลที่ตำแหน่งที่ PC ชี้อยู่ถูกส่งเข้าอินเตอร์นอลดาต้าบัสเมื่อสัญญาณนาฬิกาลูกที่ 8 เข้ามา คอนโทรลลอจิกยูนิตทำการเก็บค่าข้อมูลในอินเตอร์นอลดาต้าบัสลงในอินสตรัคชั่นรีจิสเตอร์ 3 เมื่อสัญญาณนาฬิกาลูกที่ 9 ถูกป้อนเข้า CPU คอนโทรลลอจิกยูนิตจะสร้างสัญญาณให้โปรแกรมเคาน์เตอร์เพิ่มค่าที่เก็บอยู่ขึ้น 1 ตำแหน่งขั้นตอนนี้ในกรณีที่ อินสตรัคชั่นดีโค้ด บอกว่าคำสั่งนี้ มีความยาว 3 ไบต์ เมื่อ CPU ได้รับสัญญาณนาฬิกาลูกที่ 10 เข้ามาจะทำตามคำสั่งที่ได้รับ



รูปที่ 2.11 จำนวนที่ใช้ในการเฟ็ทซ์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 กรณีการเอ็กซ์ซีคิว (การกระทำคำสั่งที่ CPU ได้รับ)

คำสั่งที่ CPU ได้รับนี้มีการทำงานอยู่หลายจำพวกซึ่งแบ่งได้ดังนี้

1) กลุ่มคำสั่งโหลดข้อมูล 8 บิต

ใช้เคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์ ด้วยกันหรือระหว่างรีจิสเตอร์กับเมมโมรี่ขนาด 8บิต

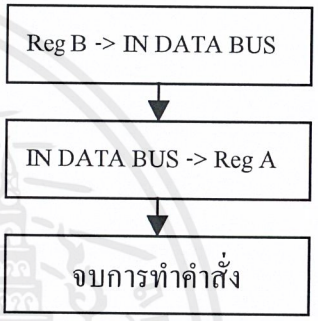
ตัวอย่าง การทำงานของคำสั่งในกลุ่มนี้

คำสั่ง LD A, B (B คือ รีจิสเตอร์เก็บข้อมูลต้นทาง , A คือ รีจิสเตอร์เก็บข้อมูลปลายทาง) คำสั่งนี้คือการนำข้อมูลเลขฐานสองขนาด 8 บิตที่เก็บอยู่ในรีจิสเตอร์ B นำไปเก็บในรีจิสเตอร์ A

ก่อนทำคำสั่ง รีจิสเตอร์ A = 35H, รีจิสเตอร์ B = 12H

1) นำข้อมูลใน รีจิสเตอร์ B ออกสู่ อินเตอร์นอลดาต้าบัส

2) สั่งเก็บข้อมูลที่ อินเตอร์นอลดาต้าบัสลงในรีจิสเตอร์ A



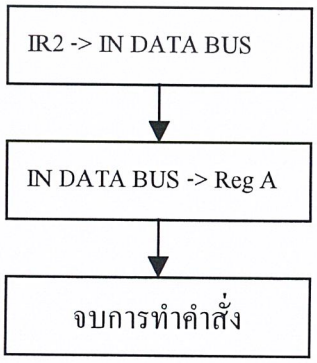
หลังทำคำสั่งค่าใน รีจิสเตอร์ A = 12H, รีจิสเตอร์ B = 12H

คำสั่ง LD A, n (n คือ ค่าในหน่วยความจำภายนอกขนาด 8 บิต, A คือ รีจิสเตอร์เก็บข้อมูลปลายทาง) คำสั่งนี้คือการนำข้อมูลเลขฐานสองขนาด 8 บิตที่เก็บอยู่ในหน่วยความจำภายนอกนำไปเก็บในรีจิสเตอร์ Aก่อนทำคำสั่งค่าใน รีจิสเตอร์ A = 35H, n = 9DH

ขณะที่เฟ็ทซ์ค่า n จะถูกนำมาเก็บใน อินสตรัคชั่นรีจิสเตอร์ 2 (ค่า n มาจากหน่วยความจำภายนอกในตำแหน่งถัดจากตำแหน่งที่อยู่ของ ออปโค้ดไบต์แรก)

1) นำข้อมูลใน อินสตรัคชั่นรีจิสเตอร์ 2 ออกสู่ อินเตอร์นอลดาต้าบัส

2) สั่งเก็บข้อมูลที่ อินเตอร์นอลดาต้าบัส ลงใน รีจิสเตอร์ A

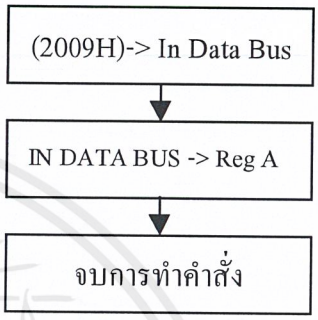


หลังทำคำสั่งค่าใน รีจิสเตอร์ A = 9DH, n = 9DH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง LD A, (HL) (HL คือ ค่าแอดเดรสของที่อยู่ของข้อมูลในหน่วยความจำภายนอก A คือ รีจิสเตอร์เก็บข้อมูลปลายทาง) คำสั่งนี้คือการนำข้อมูลจากหน่วยความจำภายนอกในตำแหน่งที่ HL ชี้อยู่ นำไปเก็บในรีจิสเตอร์ A ก่อนทำคำสั่งค่าใน รีจิสเตอร์ A = 35H, (HL) = 2009H เก็บข้อมูล 37H

1) นำข้อมูล HL ออกสู่ แอดเดรสบัส
เรียกใช้หน่วยความจำภายนอกให้อ่านข้อมูล
ตำแหน่ง 2009H เข้าสู่ อินเตอร์ดาต้าบัส
2) สั่งเก็บข้อมูลที่ อินเตอร์ดาต้าบัสลงในรีจิสเตอร์ A



หลังทำคำสั่งค่าใน รีจิสเตอร์ A = 37H, (HL) = 2009H เก็บข้อมูล 37H

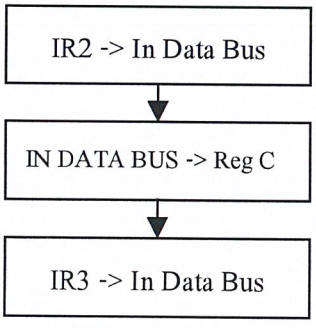
2) กลุ่มคำสั่งโหลดข้อมูล 16 บิต

ใช้เคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์ด้วยกันหรือระหว่างรีจิสเตอร์ กับเมมโมรี่ ขนาด 16 บิต

คำสั่ง LD BC, nn เป็นการนำค่าข้อมูลเลขฐานสองขนาด 16 บิต ไปเก็บยังรีจิสเตอร์คู่ BC ก่อนทำคำสั่ง B = 65H, C = 84H, n ไบต์ต่ำ = 54H, n ไบต์สูง = 23H

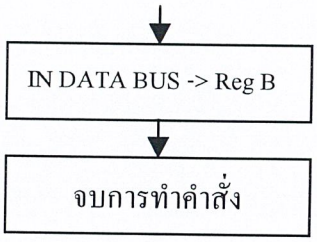
ขณะที่เฟิทช์ค่า n ไบต์ต่ำ จะถูกนำมาเก็บใน อินสตรัคชั่นรีจิสเตอร์ 2 (ค่า n ไบต์ต่ำมาจากหน่วยความจำภายนอกในตำแหน่งถัดจากตำแหน่งที่อยู่ของออปโค้ดไบต์แรก) ค่า n ไบต์สูง จะถูกนำมาเก็บใน อินสตรัคชั่นรีจิสเตอร์ 3 (ค่า n ไบต์สูงมาจากหน่วยความจำภายนอกในตำแหน่งถัดจากตำแหน่งที่อยู่ของ n ไบต์ต่ำ)

1) นำค่า n ไบต์ต่ำเก็บอยู่ใน IR2 สู่ อินเตอร์นอลดาต้าบัส
2) เก็บค่าในอินเตอร์นอลดาต้าบัสลงใน รีจิสเตอร์ C
3) นำค่า n ไบต์สูงเก็บอยู่ใน IR3 สู่อินเตอร์นอลดาต้าบัส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เก็บค่า ใน อินเตอร์นอลดาต้าบัสลงใน รีจิสเตอร์ B



3) กลุ่มคำสั่งคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก 8 บิต

ใช้ปฏิบัติกับข้อมูลที่อยู่ใน แอคคิวมูเลเตอร์, รีจิสเตอร์ หรือใน เมมโมรี่ โดยผลลัพธ์ที่ได้การคำนวณนี้เป็นขนาด 8 บิต

คำสั่ง ADD A, B เป็นการนำค่าในรีจิสเตอร์ A บวกกับค่าในรีจิสเตอร์ B แล้วนำผลลัพธ์เก็บยัง รีจิสเตอร์ A ก่อนทำคำสั่งค่าใน รีจิสเตอร์ A = 15H, รีจิสเตอร์ B = 23H

1) นำค่าใน รีจิสเตอร์ A สู่ อินเตอร์นอลดาต้าบัส

2) เก็บค่าใน อินเตอร์นอลดาต้าบัส ลงใน Temp1

3) นำค่าใน รีจิสเตอร์ B สู่ อินเตอร์นอลดาต้าบัส

4) เก็บค่าใน อินเตอร์นอลดาต้าบัส ลงใน Temp2

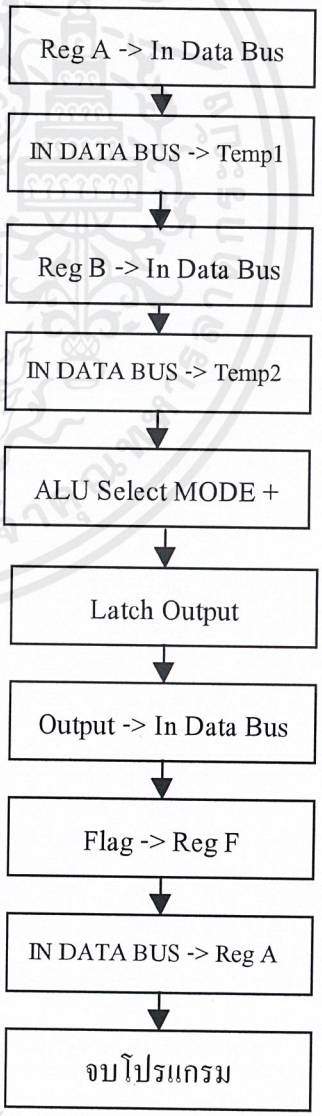
5) เลือก ALU โหมด บวกเลข

6) แลดซ์ค่าผลลัพธ์ไว้

7) นำค่าผลลัพธ์สู่ อินเตอร์นอลดาต้าบัส

8) เก็บ ค่าแฟลกลงในรีจิสเตอร์ F

9) เก็บค่าผลลัพธ์ลงรีจิสเตอร์ A



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) กลุ่มคำสั่งในการหมุนข้อมูล (Rotate) และเลื่อนข้อมูล (Shift)

เป็นคำสั่งที่ใช้หมุนค่าของข้อมูล และเลื่อนบิตของข้อมูล เช่นคำสั่ง SLA s เลื่อนข้อมูลในรีจิสเตอร์ A ไปทางซ้ายมือ โดยนำ 0 ไปแทนที่ D0 ก่อนทำคำสั่ง ค่าใน รีจิสเตอร์ A = 00011011

1) นำค่ารีจิสเตอร์ A สู่อินเตอร์นอลดาต้าบัส

2) นำค่าในอินเตอร์นอลดาต้าบัสเก็บยัง ALU

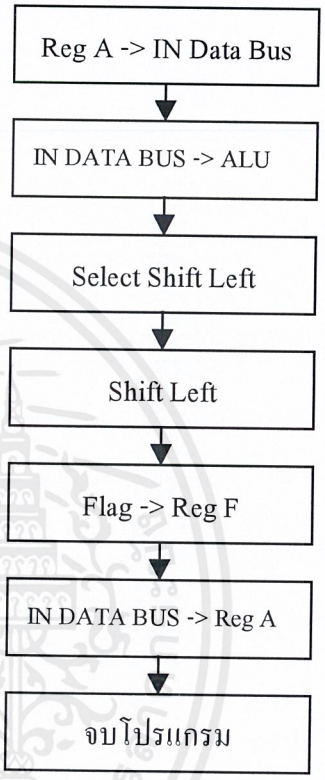
3) เลือก ALU ส่วนวงจรเลื่อนบิตทางซ้าย

4) ข้อมูลทำการเลื่อนบิตทางซ้าย

5) เก็บค่า แฟลก

6) นำเอาต์พุตสู่อินเตอร์นอลดาต้าบัส

7).นำค่าใน อินเตอร์นอลดาต้าบัสเก็บยังรีจิสเตอร์ A



5) กลุ่มคำสั่ง เซท, รีเซท และเทสบิต

ใช้ เซท, รีเซท และเทสบิตโดยสามารถเปลี่ยนแปลง และทดสอบข้อมูลแต่ละบิต บิต0-บิต7 ในแอคคิวมูเลเตอร์รีจิสเตอร์เมมโมรี่โดยผลของการเปลี่ยนแปลง และการทดสอบข้อมูลแต่ละบิตจะถูกแสดงใน แฟลก

6) กลุ่มคำสั่งการกระโดด

ได้แก่คำสั่งการกระโดดไปทำงานยังแอดเดรสปลายทาง เช่น JP nn, JR e

7) กลุ่มคำสั่งการเรียกโปรแกรมย่อย และกลับเข้าสู่โปรแกรมหลัก

ได้แก่คำสั่ง Jump โดยการตรวจสอบสภาวะของแฟลกก่อน แบบมีเงื่อนไข และ Jump โดยไม่มีการตรวจสอบสภาวะของแฟลกก่อนรวมทั้งคำสั่งที่ใช้เรียกโปรแกรมย่อย และคำสั่งพวกกลับจากโปรแกรมย่อย เช่น CALL nn, RET, RST p

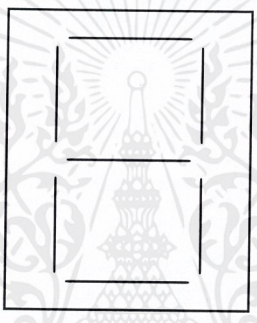
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) กลุ่มคำสั่งรับส่งข้อมูลเข้าออก จากอุปกรณ์ภายนอก

คำสั่งพวกนี้ใช้เคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์กับหน่วยความจำ ไปยังอุปกรณ์อินพุต / เอาต์พุต ภายนอกมีคำสั่งพิเศษสำหรับเคลื่อนย้าย Block ของข้อมูล สูงสุด 256 ไบต์ ระหว่างอุปกรณ์ อินพุต / เอาต์พุต กับ เมมโมรี่ เช่น IN (BC), n, OUT n, (BC)

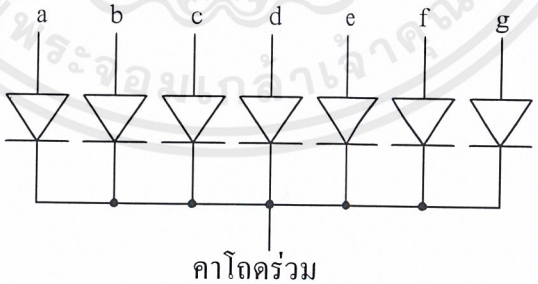
2.7 หลักการทำงานของภาคแสดงผล

การแสดงผลข้อมูล เช่น ค่าต่างๆ ในรีจิสเตอร์ ของชุดสาธิตสถาปัตยกรรมภายใน CPU ซึ่งเก็บข้อมูลในรหัส BCD-8421 จึงต้องมีการแปลงข้อมูล รหัส BCD-8421 นี้ เป็น เอาต์พุต a, b, c, d, e, f และ g เพื่อใช้ขับ ส่วนแสดงผล 7 ส่วน มีหลักการทำงานดังนี้



รูปที่ 2.12 ส่วนแสดงผล 7 ส่วน

โครงสร้างของส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม



รูปที่ 2.13 โครงสร้างส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

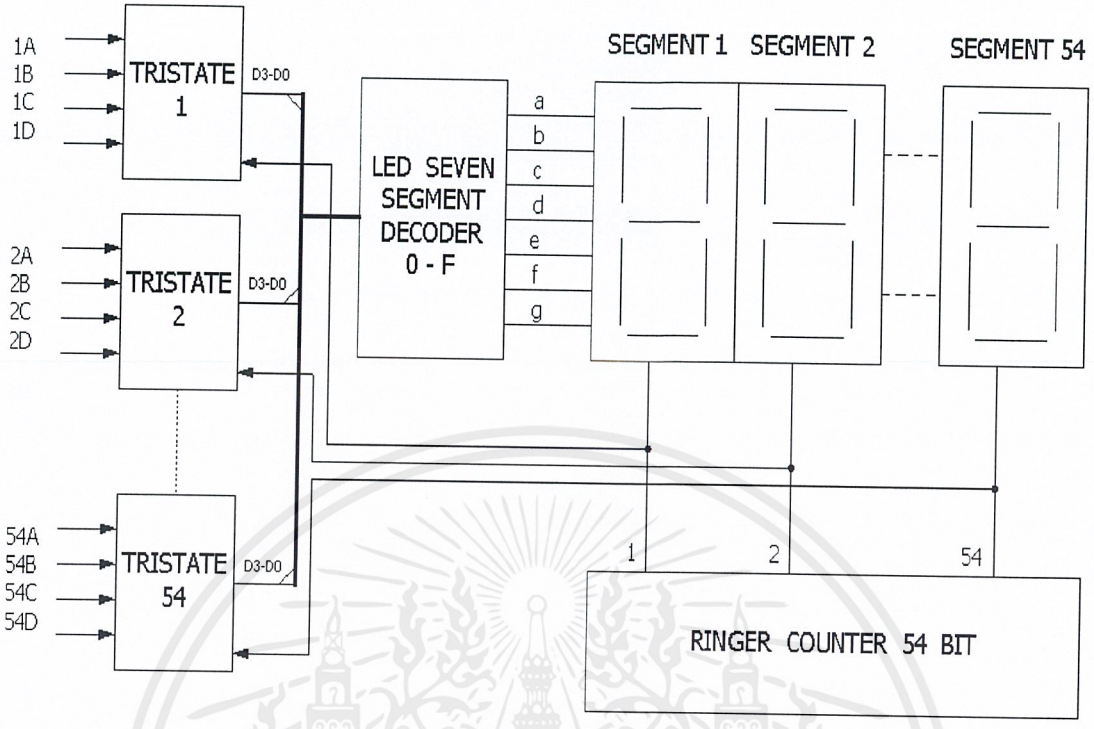
ตารางที่ 2.4 การขับส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม

หมายเลข	INPUT BCD-8421				OUTPUT LED Seven Segment Common Cathode								รหัส
	D	C	B	A	H	G	f	e	d	c	b	a	
0	0	0	0	0	0	0	1	1	1	1	1	1	3F
1	0	0	0	1	0	0	0	0	0	1	1	0	06
2	0	0	1	0	0	1	0	1	1	0	1	1	5B
3	0	0	1	1	0	1	0	0	1	1	1	1	4F
4	0	1	0	0	0	1	1	0	0	1	1	0	66
5	0	1	0	1	0	1	1	0	1	1	0	1	6D
6	0	1	1	0	0	1	1	1	1	1	0	1	7D
7	0	1	1	1	0	0	0	0	0	1	1	1	07
8	1	0	0	0	0	1	1	1	1	1	1	1	7F
9	1	0	0	1	0	1	1	0	1	1	1	1	6F
A	1	0	1	0	0	1	1	1	0	1	1	1	77
B	1	0	1	1	0	1	1	1	1	1	0	0	7C
C	1	1	0	0	0	0	1	1	1	0	0	1	39
D	1	1	0	1	0	1	0	1	1	1	1	0	5E
E	1	1	1	0	0	1	1	1	1	0	0	1	79
F	1	1	1	1	0	1	1	1	0	0	0	1	71

การทำให้ตัวส่วนแสดงผล 7 ส่วนชนิดคาโถดร่วม ติดนั้น ขาคาโถดร่วม ต้องต่อลงกราวด์ และจ่ายไบอัสไฟบวกให้กับหลักที่ต้องการติด เช่นต้องการให้เลข 1 ติด ให้ลอจิก 1 กับขา B, C เป็นต้น

การแสดงผลของส่วนเลขแสดง 7 ส่วน สามารถป้อนค่า BCD-8421 ดังตารางจะให้ค่าเอาต์พุตเพื่อใช้ขับ ส่วนแสดงผล 7 ส่วน ได้ตั้งแต่หมายเลข 0 – F โดยการนำค่ารหัสในตารางโปรแกรมลง อีพรอมแล้วต่อเข้ากับวงจร สแกนดิสเพลย์ 54 หลัก ดังผังการทำงาน รูป 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.14 ผังการทำงานภาค สแกนดิสเพลย์ 54 หลัก

2.7.1 หลักการ สแกนดิสเพลย์ มีดังนี้

วงจรนับแบบวงแหวนนับ 54 เอาต์พุต ทำหน้าที่นับวนรอบ เมื่อเอาต์พุต 1 เป็น 1 ทำให้ส่วน แสดงผล 7 ส่วน ดวงที่ 1 ถูกเลือก และอินพุต BCD-8421 ชุดที่1 ถูกป้อนเข้าสู่วงจรดีโค้ดเดอร์เพื่อแปลง รหัส BCD –8421 ให้อยู่ ในรูป เอาต์พุต a, b, c, d, e, f และ g เพื่อขับส่วนแสดงผล 7 ส่วน ดวงที่ 1 วงจรนับจะนับวนจาก 1 ถึง 54 โดยความถี่ในการนับ ประมาณ 1 kHz จะทำให้เวลามองตัวเลขแสดงผล 7 ส่วน แล้วรู้สึกว่า ทุกหลักติดตลอดเวลา การทำงานของวงจรจะวนอยู่อย่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 ขั้นตอนการออกแบบ

การออกแบบชุดสาธิตสถาปัตยกรรมภายใน CPU แบ่งได้ 3 ส่วน ดังนี้

3.1.1 ส่วนวงจรภายใน CPU

ทำหน้าที่รับคำสั่งจากหน่วยความจำภายนอกนำเข้าสู่ตัว CPU เราเรียกว่าสภาวะการเฟ็ทช์ ต่อไป CPU ก็ทำการแปลคำสั่งนั้นว่าเป็นคำสั่งอะไรแล้วกระทำตามคำสั่งนั้นเราเรียกว่าสภาวะการเอ็กซ์ซีคิว เมื่อเสร็จแล้ว CPU ก็ทำการรับคำสั่งใหม่เข้ามาประมวลผลอีก

3.1.2 ส่วนการป้อนโปรแกรม

ทำหน้ารับการกดคีย์ 0 – F จากภายนอกแล้วแปลงค่าที่ได้จากการกดคีย์ ให้อยู่ในรูปรหัส BCD-8421แล้วนำรหัสนี้ป้อนเข้าสู่หน่วยความจำภายนอกรหัสดังกล่าวก็คือภาษาเครื่องหรือ Machine Code นั่นเอง เพื่อให้ CPU ทำการประมวลผลรหัสดังกล่าวต่อไป

3.1.3 ส่วนของการแสดงผลข้อมูล

ทำหน้าที่ในการแสดงผลข้อมูล เช่น ค่าต่างๆ ในรีจิสเตอร์ของชุดสาธิตสถาปัตยกรรมภายใน CPU ซึ่งเก็บข้อมูลในรหัส BCD-8421 จึงมีการแปลงข้อมูล รหัส BCD-8421 นี้ เป็น เอาต์พุต A, B, C, D, E, F, G เพื่อใช้ขับ ส่วนแสดงผล 7 ส่วน จำนวน 54 หลัก ใช้วิธีการสแกนดิสเพลย์

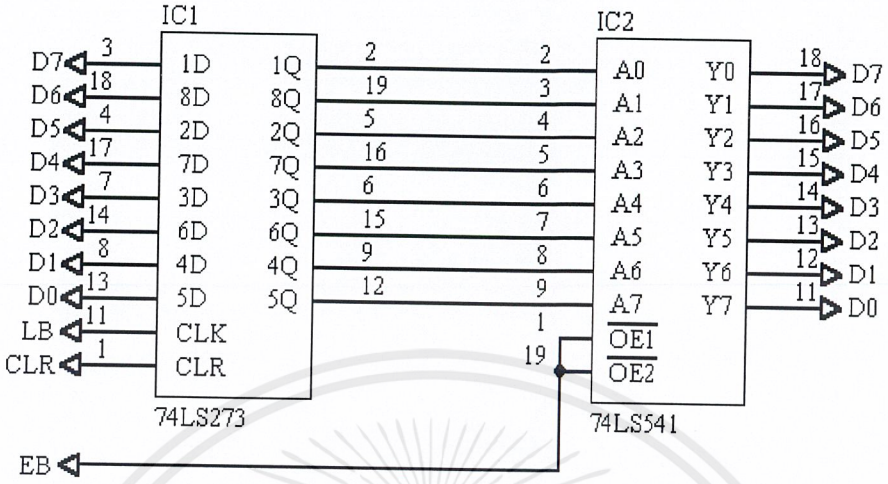
3.2 วงจรภายใน CPU สามารถแบ่งออกเป็น 4 ส่วน มีดังนี้

3.2.1 รีจิสเตอร์ทั่วไปขนาด 8 บิต

1) รีจิสเตอร์ทั่วไปทำหน้าที่เก็บข้อมูลเลขฐานสองจำนวน 8 บิต

โดยรีจิสเตอร์ A เป็นรีจิสเตอร์ตัวหลักที่ใช้ในการเก็บค่าเลขฐานสอง 8 บิตที่ใช้เป็นตัวตั้งและใช้เก็บผลลัพธ์ที่ได้จากการทำคำสั่งทางคณิตศาสตร์ และลอจิกรีจิสเตอร์ F ทำหน้าที่เก็บค่าแฟลกที่ได้จากทำคำสั่งทางคณิตศาสตร์ และกระทำทางลอจิก รีจิสเตอร์ B, C, D, E, H, L ทำหน้าที่เก็บข้อมูลทั่วไปขนาด 8 บิต และสามารถใช้เป็นรีจิสเตอร์คู่ขนาด 16 บิตได้โดยนำรีจิสเตอร์ 2 ตัวมาต่อร่วมกัน (BC, DE, HL) รีจิสเตอร์ IR 1 ทำหน้าที่เก็บคำสั่งที่ได้จากการ เฟ็ทช์ ไบต์แรก รีจิสเตอร์ IR 2 ทำหน้าที่เก็บคำสั่งที่ได้จากการเฟ็ทช์ไบต์ที่ 2 รีจิสเตอร์ IR 3 ทำหน้าที่เก็บคำสั่งที่ได้จากการเฟ็ทช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 รีจิสเตอร์เก็บค่าทั่วไป 8 บิต

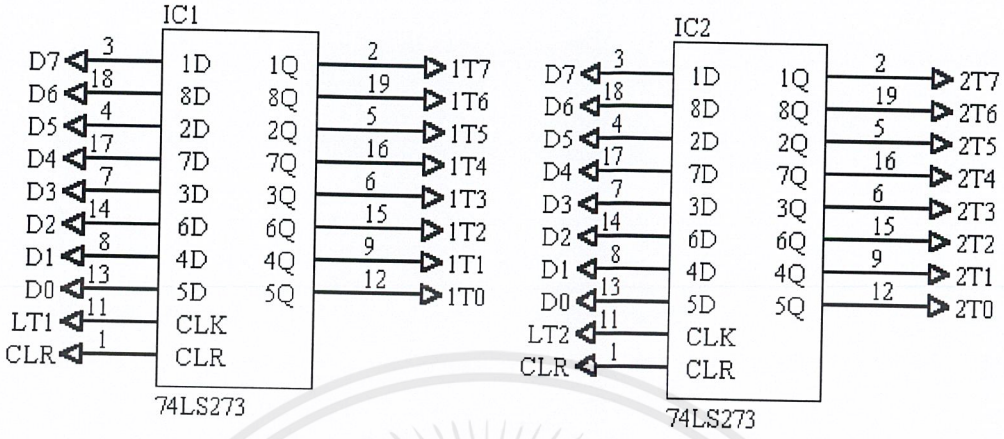
การทำงานของรีจิสเตอร์ทั่วไป 8 บิต มีดังนี้

ใช้ ไอซีเบอร์ 74LS273 เป็นไอซีที่มี D ฟลิปฟลอปภายใน 8 ตัว โดย D ฟลิปฟลอปแต่ละตัวทำหน้าที่เก็บข้อมูลเลขฐานสองตัวละ 1 บิตโดยมีขาควบคุมคือขา 11 ทำหน้าโหลดข้อมูลจากอินเตอร์นอลดาต้าบัส เข้าสู่ D ฟลิปฟลอปภายในตัวไอซี เมื่อต้องการโหลดข้อมูลเข้าสู่รีจิสเตอร์ก็ให้พัลซ์ขอบขาขึ้นที่ขา 11 ของไอซี โดยขา 1D-8D เป็นขารับข้อมูลเข้า และขา 1Q-8Q เป็นขาข้อมูลออก เมื่อต้องการเคลียร์ค่าในรีจิสเตอร์โดยให้ขา CLR เป็น 0 ข้อมูลใน ไอซี 1 จะเป็น 0 ทั้ง 8 เอาต์พุต ปกติขานี้ถูกต่อกับสัญญาณลอจิก 1 เอาต์พุตของ ไอซี 1 ถูกต่อกับ อินพุต ไอซี 2 เบอร์ 74LS541 ทำหน้าที่ ควบคุมสัญญาณที่ ไอซี 74LS273 เก็บข้อมูลอยู่เพื่อเปิดหรือปิดข้อมูลนี้สู่อินเตอร์นอลดาต้าบัส โดยเอาต์พุตของ ไอซี 74LS541 ถูกต่อกับอินเตอร์นอลดาต้าบัสมีขาควบคุมการเปิดปิดข้อมูลสู่ อินเตอร์นอลดาต้าบัส คือขา 1, 19 ของไอซี 2 เมื่อเป็น 0 ข้อมูลจะถูกส่งเข้าสู่อินเตอร์นอลดาต้าบัส และขา 1, 19 เมื่อได้รับลอจิก 1 เอาต์พุตของ ไอซี 2 จะถูกตัดออกจากอินเตอร์นอลดาต้าบัส

2) รีจิสเตอร์พักข้อมูล 8 บิต

รีจิสเตอร์ Temp1 และ Temp2 ทำหน้าที่เก็บข้อมูลเลขฐานสองจำนวน 8 บิต โดยรีจิสเตอร์ Temp1 และ Temp2 เป็นรีจิสเตอร์ที่ทำหน้าที่ในการเก็บค่าเลขฐานสอง 8 บิต Temp1 เก็บค่าตัวตั้ง และTemp2 เก็บค่าตัวกระทำ พักไว้ก่อนส่งเข้า การทำคำสั่งทางคณิตศาสตร์ และลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 รีจิสเตอร์พักข้อมูล 8 บิต

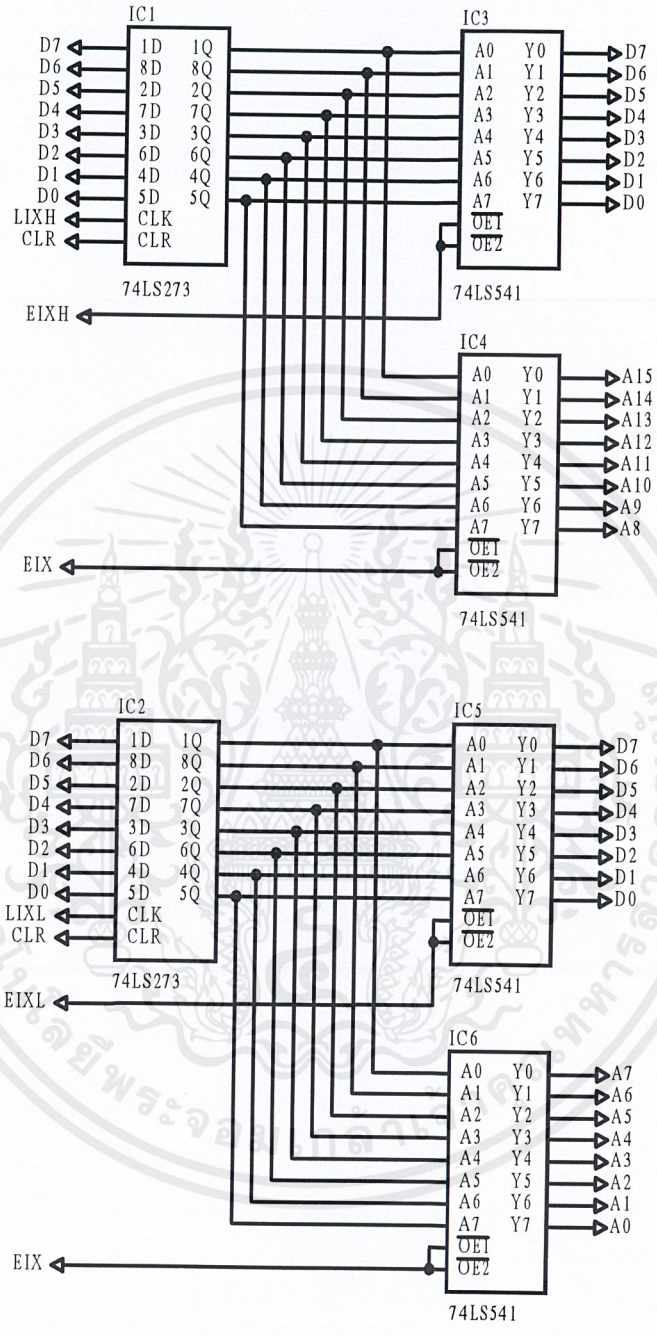
การทำงานของรีจิสเตอร์ Temp1, Temp2 มีดังนี้

การทำงานเหมือนกับรีจิสเตอร์ทั่วไป 8 บิต เพียงแต่ Temp1 เก็บค่าตัวตั้ง และTemp2 เก็บค่าตัวกระทำ พักไว้ก่อนส่งเข้า การทำคำสั่งทางคณิตศาสตร์ และลอจิก โดย ไอซี 1 และไอซี 3 ที่ใช้คือเบอร์ 74LS273 เป็นไอซีที่มี D ฟลิปฟลอป ภายใน 8 ตัว โดย D ฟลิปฟลอปแต่ละตัวทำหน้าที่เก็บข้อมูลเลขฐานสองตัวละ 1 บิต โดยมีขาควบคุมคือขา 11 ทำหน้าโหลดข้อมูลจากอินเตอร์นอลดาต้าบัส 8 บิต เข้าสู่ D ฟลิปฟลอป ภายในตัวไอซี เมื่อต้องการโหลดข้อมูลเข้าสู่รีจิสเตอร์ก็ให้พัลซ์ขอบขาขึ้นที่ขา 11 ของไอซี โดยขา 1D-8D เป็นขารับข้อมูลเข้า และขา 1Q-8Q เป็นขาข้อมูลออก เมื่อต้องการเคลียร์ค่าในรีจิสเตอร์โดยให้ขา CLR เป็น 0 ข้อมูลใน ไอซี 1 และไอซี 3 จะเป็น 0 ทั้ง 8 เอาต์พุต ปกติขานี้ถูกต่อกับสัญญาณลอจิก 1 เอาต์พุตของ ไอซี 1 ถูกต่อกับ อินพุตของวงจรประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิกด้านตัวตั้ง และเอาต์พุตของไอซี 3 ถูกต่อกับ อินพุตของวงจรประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิกด้านตัวกระทำ

3) อินเด็กซ์รีจิสเตอร์ขนาด 16 บิต

อินเด็กซ์รีจิสเตอร์เป็นรีจิสเตอร์คู่ขนาด 16 บิต ทำหน้าที่เก็บข้อมูลเลขฐาน 2 จำนวน 16 บิต เพื่อใช้ในการอ้างตำแหน่งที่อยู่ของข้อมูลจากหน่วยความจำภายนอก ซึ่ง CPU นี้มีอินเด็กซ์รีจิสเตอร์จำนวน 2 ตัว คือ IX และ IY อินเด็กซ์รีจิสเตอร์นี้ช่วยให้การเขียนโปรแกรมที่มีการติดต่อกับหน่วยความจำภายนอกทำได้สะดวกมากยิ่งขึ้น ถ้าการติดต่อหน่วยความจำภายนอกไม่ใช้อินเด็กซ์รีจิสเตอร์แล้วการเขียนโปรแกรมโดยใช้รีจิสเตอร์ทั่วไปมาอ้างตำแหน่งหน่วยความจำภายนอกก็จะต้องเขียนโปรแกรมถึงหลายบรรทัด จึงสามารถทำให้เขียนโปรแกรมได้สั้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 อินเด็กซ์รีจิสเตอร์

การทำงานคล้ายกับรีจิสเตอร์ทั่วไป 8 บิตเพียงแต่อินเด็กซ์รีจิสเตอร์ใช้รีจิสเตอร์ขนาด 8บิต 2 ตัวต่อร่วมกันเป็นรีจิสเตอร์ขนาด 16 บิต โดย ไอซี1 และไอซี3 ที่ใช้คือเบอร์ 74LS273 เป็นไอซีที่มี D ฟลิปฟลอป ภายใน 8 ตัวโดย D ฟลิปฟลอปแต่ละตัวทำเก็บข้อมูลเลขฐานสองตัวละ 1 บิตโดยมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาควบคุมคือขา 11 ทำหน้าโหลดข้อมูลจาก อินเตอร์นอลดาต้า 8 บิต เข้าสู่ D ฟลิฟลอป ภายในตัวไอซี เมื่อต้องการโหลดข้อมูลเข้าสู่รีจิสเตอร์ก็ให้พัลซ์ขอบขาขึ้นที่ขา 11 ของไอซี โดยขา1D-8D เป็นขารับข้อมูลเข้า และขา 1Q-8Q เป็นขาข้อมูลออก เมื่อต้องการเคลียร์ค่าในรีจิสเตอร์โดยให้ขา CLR เป็น 0 ข้อมูลใน ไอซี 1 และไอซี 3 จะเป็น 0 ทั้ง 8 เอาต์พุต ปกติขานี้ถูกต่อกับสัญญาณลอจิก 1 เอาต์พุตของ ไอซี1 ถูกต่อกับอินพุต ไอซี 2 เบอร์ 74LS541 ทำหน้าที่ ควบคุมสัญญาณที่ไอซี 74LS273 เก็บข้อมูลอยู่เพื่อเปิดหรือปิดข้อมูลนี้สู่อินเตอร์นอลดาต้า ด้าน A7 - A0โดยเอาต์พุตของ ไอซี 74LS541 ถูกต่อกับ อินเตอร์นอลแอดเดรสบัสด้าน A7 - A0 มีขาควบคุมการเปิดปิดข้อมูลสู่อินเตอร์นอลแอดเดรสบัสด้าน A7 - A0 คือขา 1, 19 ของ ไอซี 2 เมื่อเป็น 0 ข้อมูลจะถูกส่งเข้าสู่ อินเตอร์นอลแอดเดรสบัส ด้าน A7 - A0 และขา 1, 19 เมื่อได้รับลอจิก 1 เอาต์พุตของ ไอซี 2 จะถูกตัดออกจาก อินเตอร์นอลแอดเดรสบัสด้าน A7 - A0 และเอาต์พุตของ ไอซี 3 ถูกต่อกับอินพุต ไอซี 4 เบอร์ 74LS541 ทำหน้าที่ ควบคุมสัญญาณที่ ไอซี 74LS273 เก็บข้อมูลอยู่เพื่อเปิดหรือปิดข้อมูลนี้สู่อินเตอร์นอลแอดเดรสบัส ด้าน A15 – A8โดยเอาต์พุตของ ไอซี 74LS541 ถูกต่อกับอินเตอร์นอลแอดเดรสบัส ด้าน A15 – A8 มีขาควบคุมการเปิดปิดข้อมูลสู่อินเตอร์นอลแอดเดรสบัสด้าน A15-A8 คือขา 1, 19 ของ ไอซี 4 เมื่อเป็น 0 ข้อมูลจะถูกส่งเข้าสู่อินเตอร์นอลแอดเดรสบัสด้าน A15 – A8 และขา 1, 19 เมื่อได้รับลอจิก 1 เอาต์พุตของ ไอซี 4 จะถูกตัดออกจากอินเตอร์นอลแอดเดรสบัส ด้าน A15 – A8

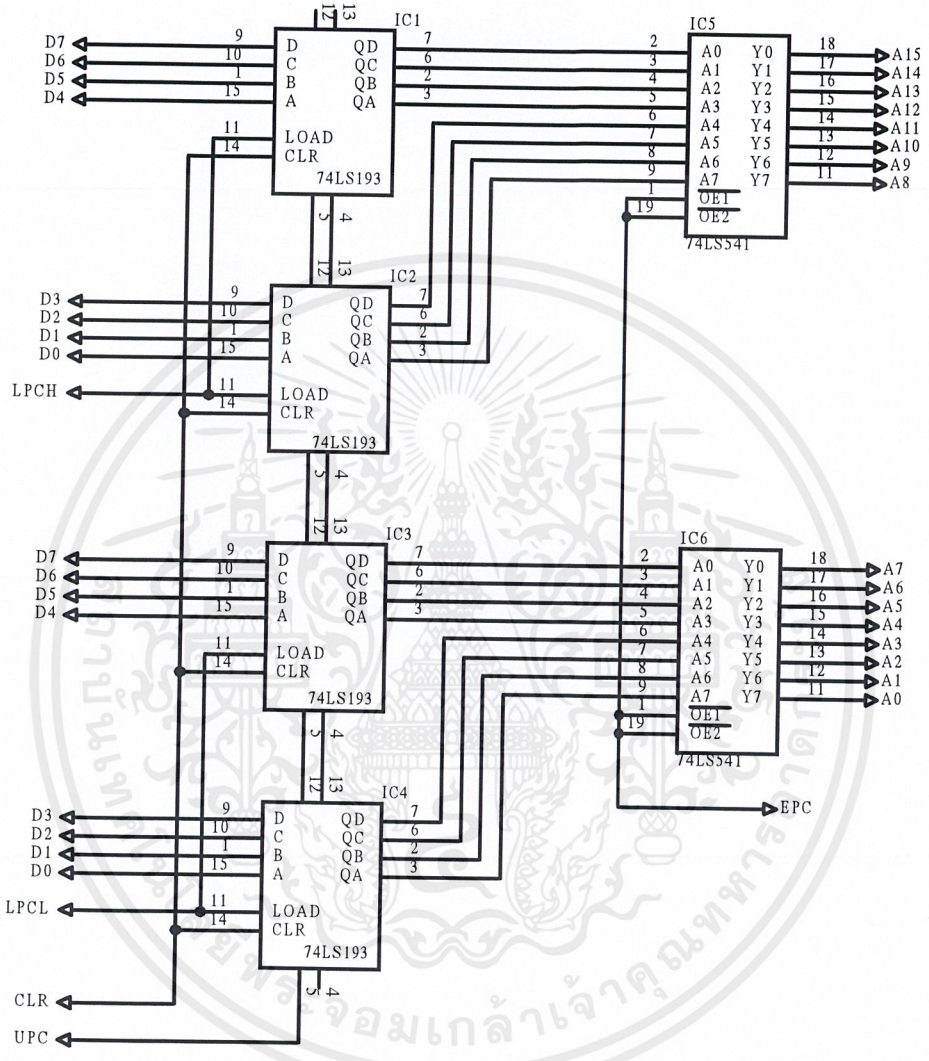
4) โปรแกรมเคาน์เตอร์ (PC)

โปรแกรมเคาน์เตอร์ (PC) ทำหน้าที่เก็บค่าตำแหน่งที่อยู่ของข้อมูลขนาด 16 บิต (Address A15-A0) เพื่อใช้สำหรับชี้ตำแหน่งที่อยู่ของข้อมูลที่ CPU จะทำการเฟ็ทซ์ คำสั่งเป็นไบต์ถัดไป และเมื่อเฟ็ทซ์ คำสั่งไปแล้วโปรแกรมเคาน์เตอร์จะทำการเพิ่มค่าตำแหน่งที่อยู่ของข้อมูล A15-A0 ขึ้น 1 ตำแหน่ง เพื่อชี้ในตำแหน่งที่อยู่ของข้อมูลที่ต้องทำการเฟ็ทซ์ในตำแหน่งถัดไปอีก
การทำงานของโปรแกรมเคาน์เตอร์ มีดังนี้

โปรแกรมเคาน์เตอร์ 16 บิต จะมีลักษณะการทำงานเป็นวงจรนับขนาด 16 บิต ใช้ไอซีเบอร์ 74LS193 เป็นไอซีนับขึ้นลงแบบ ซิงโครนัส 4 บิต ในวงจรนี้เราใช้นับขึ้นอย่างเดียว ต่ออนุกรมกัน 4 ตัวก็จะได้เป็นวงจรนับขึ้นขนาด 16 บิต ขา 5 เป็นขานับขึ้น ซึ่งทำงานที่พัลซ์ขอบขาลง มีขา12 เป็นขาตัวทดออก ใช้ต่อกับขา Up (ขา 5 ของไอซีตัวถัดไป) มีขา 13 เป็นขาตัวยืมใช้ต่อกับขา Down (ขา 4 ของไอซีตัวถัดไป) เพื่อให้สามารถนับได้จำนวนบิตมากยิ่งขึ้น ขา 14 เป็นขา เคลียร์ค่าที่นับให้เป็น 0 โดยขานี้ทำงานที่พัลซ์ขอบขาขึ้น และขา 11 เป็นขา LOAD ถ้าขานี้ได้รับลอจิก 1จะเป็นวงจรนับ และถ้าได้รับลอจิก 0 ไอซีจะทำการโหลดข้อมูลเข้าวงจรในไอซี โดยขา A, B, C, D ของ ไอซี 1, ไอซี 2 ต่อกับอินเตอร์นอลดาต้าบัส 8 บิต เพื่อใช้ค่านี้เป็นค่าแอดเดรสบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบต์ต่ำขา A, B, C, D ของ ไอซี 3, ไอซี 4 ต่อกับอินเตอร์นอลดาต้าบัส 8 บิต เพื่อใช้ค่านี้เป็นค่าแอดเดรสบัส ไบต์สูง

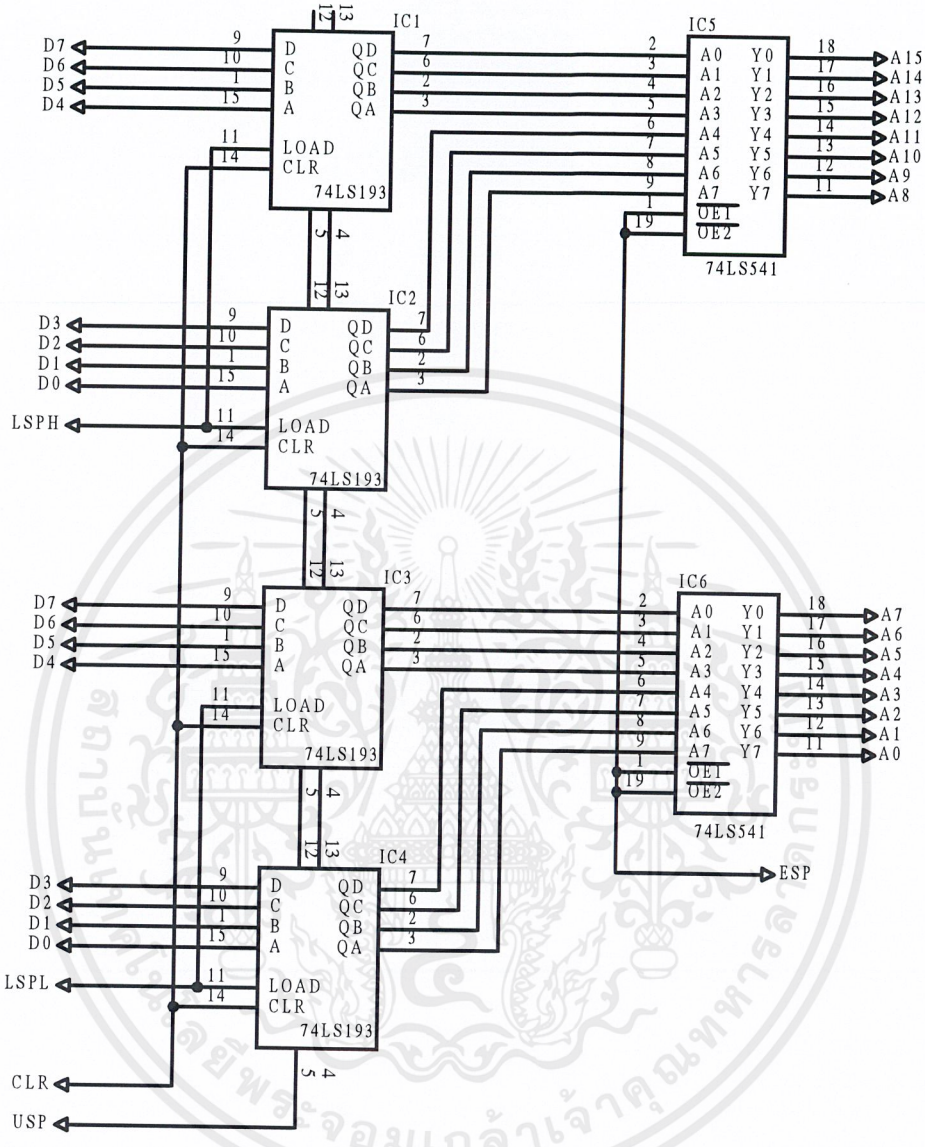


รูปที่ 3.4 โปรแกรมเคาน์เตอร์ 16 บิต

5) แสตกพอยน์เตอร์ (SP)

แสตกพอยน์เตอร์ (SP) ทำหน้าที่เก็บค่าตำแหน่งที่อยู่ของข้อมูลขนาด 16 บิต (A15-A0) เพื่อใช้สำหรับชี้ตำแหน่งที่อยู่ของข้อมูลที่ CPU นำไปพักชั่วคราวที่หน่วยความจำภายนอก และทำการเพิ่มค่าแอดเดรสในแสตกพอยน์เตอร์ ขึ้น 1 ตำแหน่ง เรียกว่าการ PUSH และ CPU นำค่าที่นำไปพักชั่วคราวออกมาสู่ตำแหน่งที่อยู่เดิม และทำการลดค่าตำแหน่งในแสตกพอยน์เตอร์ลง 1 ตำแหน่ง เรียกว่าการ POP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสตกพอยน์เตอร์

การทำงานของแสตกพอยน์เตอร์ มีดังนี้

แสตกพอยน์เตอร์ 16 บิตจะมีลักษณะการทำงานเป็นวงจรนับขนาด 16 บิต ใช้ ไอซีเบอร์ 74LS193 เป็นไอซีนับขึ้นลงแบบ ซิงโครนัส 4 บิต ในวงจรนี้เราใช้ทั้งนับขึ้น และนับลง ต่ออนุกรมกัน 4 ตัวก็จะได้เป็นวงจรนับขึ้นลงขนาด 16 บิต ขา 5 เป็นขานับขึ้น ซึ่งทำงานที่พัลซ์ขอบขาลง และขา 4 เป็นขานับลง ซึ่งทำงานที่พัลซ์ขอบขาลง มีขา12 เป็นขา Carry Out ใช้ต่อกับขา Up (ขา 5 ของไอซีตัวถัดไป) มีขา 13 เป็นขาตัวยืม ใช้ต่อกับขา Down (ขา 4 ของไอซีตัวถัดไป) ขา 14

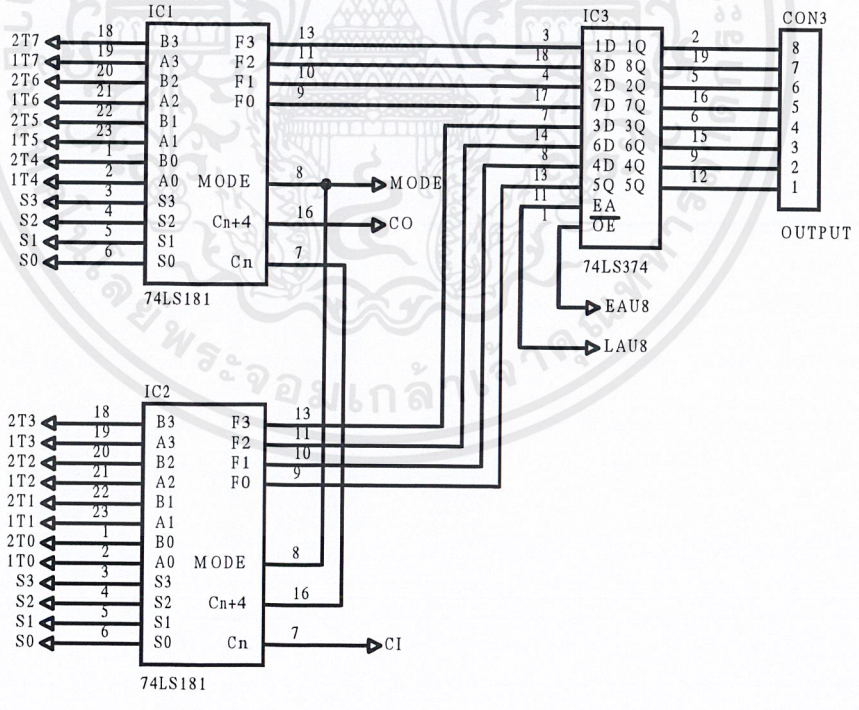
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นขาเคลียร์ค่าที่นับให้เป็น 0 โดยขานี้ทำงานที่พัลซ์ขอบขาขึ้น และขา 11 เป็นขา LOAD ถ้าขานี้ได้รับลอจิก 1 จะเป็นวงจรนับ และถ้าได้รับลอจิก 0 ไอซีจะทำการโหลดข้อมูลเข้าวงจรในไอซี โดยขา A, B, C, D ของไอซี 1, ไอซี 2 ต่อกับอินเตอร์ดาต้าบัส 8 บิตเพื่อใช้ค่านี้เป็นค่าแอดเดรสบัสไบต์ต่ำขา A, B, C, D ของไอซี 3, ไอซี 4 ต่อกับอินเตอร์นอลดาต้าบัส 8 บิตเพื่อให้มีค่าไบต์ที่สูง

3.2.2 ส่วนการประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิก ประกอบด้วย 4 ส่วน คือ

1) วงจรประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิก ขนาด 8 บิต

วงจรประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิก ขนาด 8 บิต ทำหน้าคำนวณทางคณิตศาสตร์ เช่น บวกเลข, บวกเลขแบบคิดตัวทด, ลบเลข, ลบเลขแบบคิดตัวทด และกระทำทางลอจิก เช่น AND, OR, XOR เป็นต้น ใช้ ไอซี 74LS181 เป็น ไอซีที่ทำหน้า ALU ขนาด 4 บิต จำนวน 2 ตัว ต่ออนุกรมกันเพื่อให้สามารถทำงานได้ 8 บิต โดยมีขา S0, S1, S2, S3 เป็นตัวเลือกฟังก์ชั่นการทำงาน และมีขาโหมดเลือกระหว่างการประมวลผลทางคณิตศาสตร์ และกระทำทางลอจิก ขาCn ทำหน้าที่รับสัญญาณตัวทดเข้า แล้วขาสัญญาณ Cn+4 ทำหน้าที่เอาต์พุตของตัวทดออก ขาสัญญาณ A7-A0 ทำหน้าที่รับตัวตั้งมีขา F7-F0 เป็น เอาต์พุตของวงจร



รูปที่ 3.6 วงจรประมวลผลทางคณิตศาสตร์ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

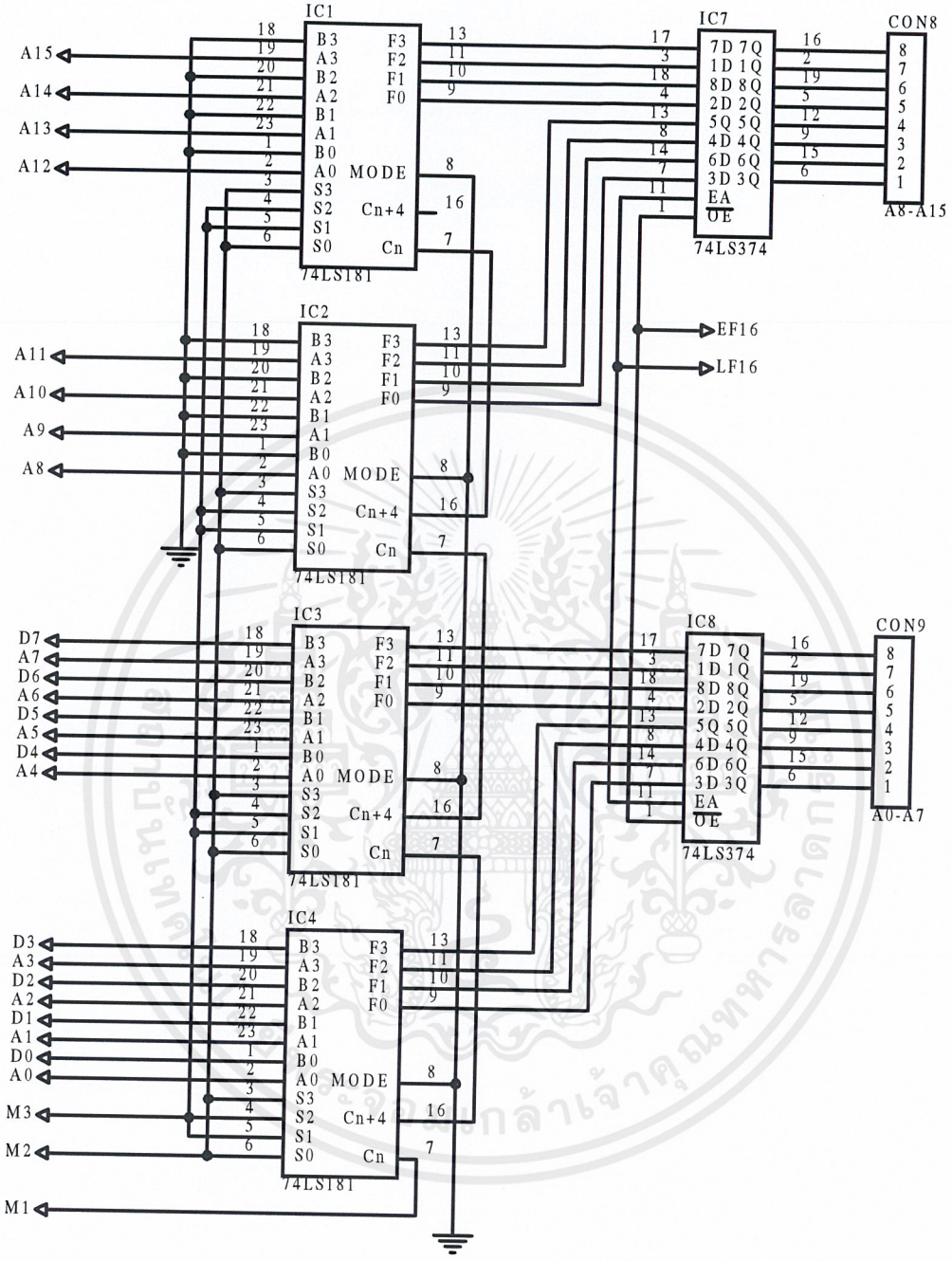
ตารางที่ 3.1 การใช้งานไอซี 74LS181

Selection				Active Hi Data		
				M=H Logic	M=L ; Arithmetic Operation	
S3	S2	S1	S0	Function	Cn=H(no carry)	Cn=L(with carry)
0	0	0	0	F= $\overline{A}$	F=A	F=A Plus 1
0	0	0	1	F= $\overline{A+B}$	F=A+B	F=(A+B) Plus 1
0	0	1	0	F= $\overline{A}$ B	F=A+ $\overline{B}$	F=(A+ $\overline{B}$) Plus 1
0	0	1	1	F=0	F=Minus 1(2'Compl)	F=Zero
0	1	0	0	F= $\overline{AB}$	F= A Plus A $\overline{B}$	F= A Plus A $\overline{B}$ Plus 1
0	1	0	1	F= $\overline{B}$	F=(A+B) Plus A $\overline{B}$	F=(A+B)Plus A $\overline{B}$ Plus1
0	1	1	0	F= A $\oplus$ B	F=A Minus B Minus 1	F=A Minus B
0	1	1	1	F=A $\overline{B}$	F= A $\overline{B}$ Minus 1	F= A $\overline{B}$
1	0	0	0	F= $\overline{A}$ +B	F=A Plus AB	F=A Plus AB Plus 1
1	0	0	1	F= $\overline{A \oplus B}$	F=A Plus B	F=A Plus B Plus 1
1	0	1	0	F=B	F=(A+ $\overline{B}$) Plus AB	F=(A+ $\overline{B}$)Plus AB Plus1
1	0	1	1	F=AB	F=AB Minus 1	F=AB
1	1	0	0	F=1	F=A Plus A*	F=A Plus A Plus 1
1	1	0	1	F= A+ $\overline{B}$	F=(A+B) Plus A	F=(A+B) Plus A Plus 1
1	1	1	0	F=A+B	F= (A+ $\overline{B}$) Plus A	F= AB Minus 1
1	1	1	1	F=A	F=A Minus 1	F=A

2) วงจรบวกเลขขนาด 16 บิต

วงจรบวกเลขขนาด 16 บิต ใช้ ไอซี 74LS181 เป็นไอซีที่ทำหน้า ALU ขนาด 4 บิต ซึ่งใช้งานไอซีดังกล่าวในโหมดการบวกเลข ใช้จำนวน 4 ตัว ต่ออนุกรมกันเพื่อให้สามารถทำงานได้ 16 บิต โดยมีขา S0, S1, S2, S3 ทำหน้าที่เลือกฟังก์ชั่นการทำงาน และมีขาโหมดเลือกให้เป็นการประมวลผลทางคณิตศาสตร์ ขา Cn ทำหน้าที่รับสัญญาณตัวทดเข้า แล้วขาสัญญาณ Cn+4 ทำหน้าที่เป็นเอาต์พุตของตัวทดออก ขาสัญญาณ A15-A0 ทำหน้าที่รับตัวตั้ง ขาสัญญาณ B7-B0 ทำหน้าที่รับตัวกระทำ มีขา F15-F0 เป็นขา เอาต์พุตของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรบวกเลข 16 บิต

3.2.3 ส่วนการควบคุมการทำงานของ CPU (Control Logic)

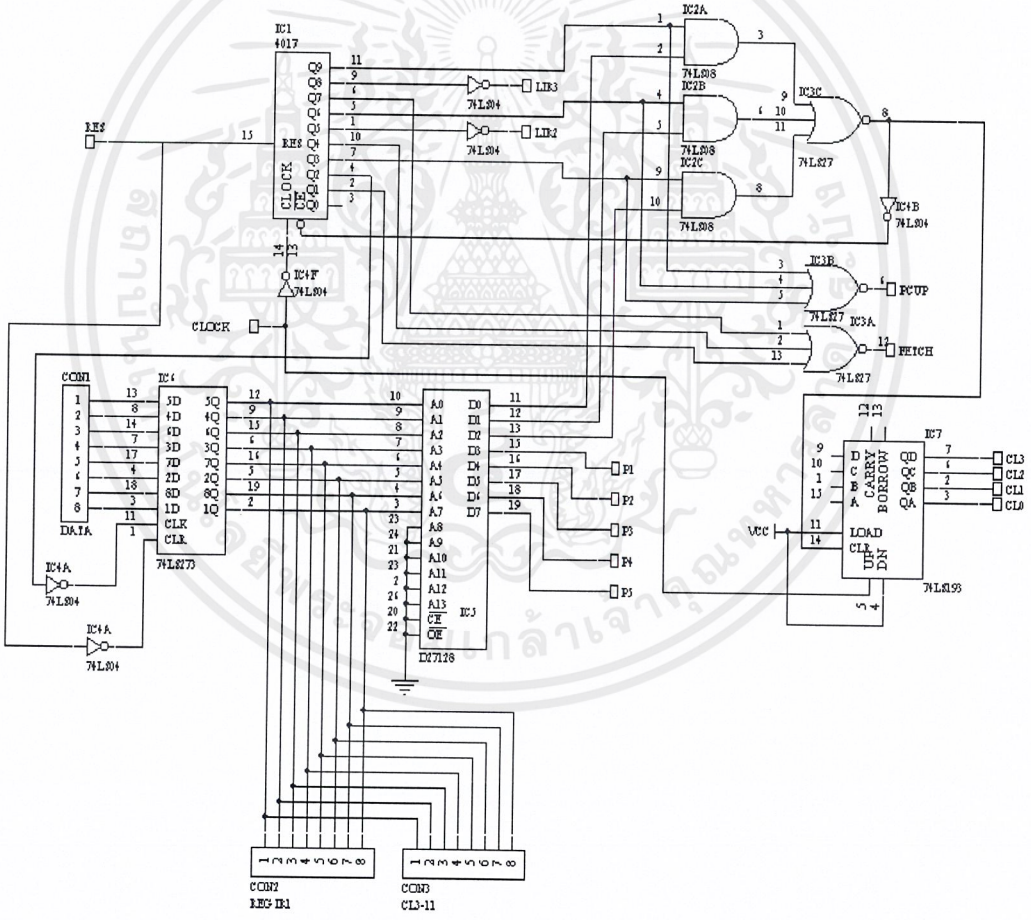
ส่วนนี้เป็นควบคุมส่วนต่างๆ ในการทำงานภายใน CPU ทำหน้าที่สร้างสัญญาณควบคุมส่วนต่างๆ ภายใน CPU ขึ้นมาตามคำสั่งที่ CPU ได้รับเข้ามา และสร้างสัญญาณควบคุมภายนอกขึ้นมา เช่น สัญญาณอ่านข้อมูลจากหน่วยความจำ(RD) สัญญาณเขียนข้อมูลลงหน่วยความจำ (WR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณร้องขอใช้หน่วยความจำภายนอก (MERQ) สัญญาณร้องขอการใช้อุปกรณ์อินพุตเอาต์พุต (IORQ) โดยแบ่งการทำงาน 2 ส่วนคือ

1)วงจรนำรหัสคำสั่งเก็บยัง CPU และแปลรหัสคำสั่ง

เมื่อสัญญาณนาฬิกาลูกแรกจ่ายเข้ามาที่ขาอินพุต ของ นอตเกต ซึ่งเอาต์พุตของนอตเกตต่อเข้ากับขา 14 ของ ไอซี 4017 ส่วนส่วนสัญญาณนาฬิกาที่เข้าขา Up ของไอซี 74LS193 นั้นไอซียังไม่ทำการนับเนื่องจากขา CLR ของไอซียังเป็น 1 อยู่ไอซีจึงยังไม่สามารถนับได้ ขั้นตอนแรก CPU จะทำการเฟ็ทซ์คำสั่งโดย ไอซี 4017 ทำหน้าที่วงจรนับแบบวงแหวน โดยเอาต์พุตของไอซี 4017 เมื่อนับถึง Q1 , Q4, Q7 3 เอาต์พุตนี้จะถูกต่ออยู่กับนอร์เกต 3 อินพุต เมื่อเอาต์พุตใด



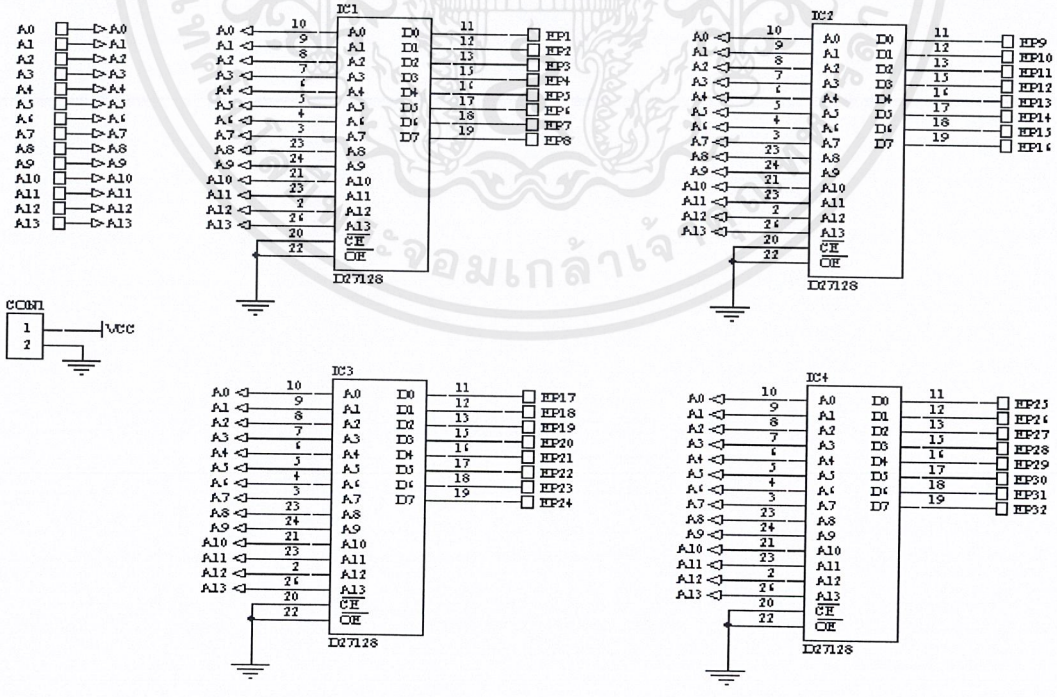
รูปที่ 3.8 วงจรเฟ็ทซ์ และถอดรหัส

เป็น 1 ที่เอาต์พุตของนอร์เกต จะเป็น 0 ทำให้ไปทริก J-K ฟลิปฟลอปซึ่งทำหน้าที่ ทอกเกิล เปลี่ยนสถานะ เป็น 0 ทำให้ อินพุต ของแอนเกต RD, MERQ, EPC เป็น 0 ทำให้เกิดการร้องขอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้หน่วยความจำภายนอกเพื่ออ่านข้อมูลที่ตำแหน่ง A15-A0 ในโปรแกรมเคาน์เตอร์ออกสู่แอดเดรสบัสชี้อยู่เข้ามาในอินเตอร์ดาต้าบัส ใน CPU และวงจรนับแบบวงแหวน นับถึงเอาต์พุตถัดไป คือ Q2 ซึ่งต่อกับนอตเกต เมื่อ Q2 เป็น 1 ผ่านนอตเกต ทำให้เป็น 0 ไปเข้าขา Clk ของไอซี 74LS273 เพื่อเขียนข้อมูลจากอินเตอร์นอลดาต้าบัสลงในรีจิสเตอร์ IR1 เอาต์พุตของรีจิสเตอร์ IR1 ถูกต่อกับ ไอซี อีพรอม 27512 เพื่อทำการถอดรหัส ว่าคำสั่งนั้นเป็นคำสั่งขนาดความยาวกี่ไบต์ เป็นคำสั่ง 1 ไบต์ หรือ 2 ไบต์ หรือ 3 ไบต์ ซึ่งจะมี 3 เอาต์พุตคือ 1 ไบต์ สัญญาณขา D2 จะเป็น 1 ถ้าคำสั่งเป็น 2 ไบต์ สัญญาณที่ขา D1 จะเป็น 1 ถ้าคำสั่งเป็น 3 ไบต์ สัญญาณที่ขา D0 จะเป็น 1 คือ คำสั่งจะกี่ไบต์เอาต์พุต นั้นจะเป็น 1 ส่วนอีก 2 ขา จะเป็น 0 นั่นเอง และขั้นตอนถัดไปเมื่อขา Q3 เป็น 1 จะต่อกับ แอนท์เกต มีเอาต์พุตออก 1 เข้า นอร์เกต 3 อินพุต ซึ่งนอร์เกต นี้เมื่ออินพุตใดเข้า 1 เพียงอินพุตเดียวเอาต์พุตออก 0 ทันที และถ้าอินพุตในเป็น 0 หมดเอาต์พุตจะออก 1 เมื่อเอาต์พุตของนอร์เกต เป็น 0 แล้วจะเพิ่มค่าใน PC ขึ้น 1 ค่า และเอาต์พุตของ นอร์เกต อีกตัวเป็น 0 ทำให้ ไอซี 74LS193 ขา CLR ที่เป็น 1 อยู่ (สั่งหยุดอยู่) กลับมาเป็น 0 ทำให้ ไอซี 74LS193 นี้ทำการนับขึ้น 0-F และ เอาต์พุต ของนอร์เกต ต่อกับนอตเกต ซึ่งต่อกับขา CE จะมีค่าเป็น 1 ทำให้ ไอซี 4017 หยุดนับซึ่งจังหวะนี้ก็ทำการเริ่มจังหวะการเอ็กซ์ซีคิว แล้วนั่นเอง

2) ส่วนกระทำคำสั่ง



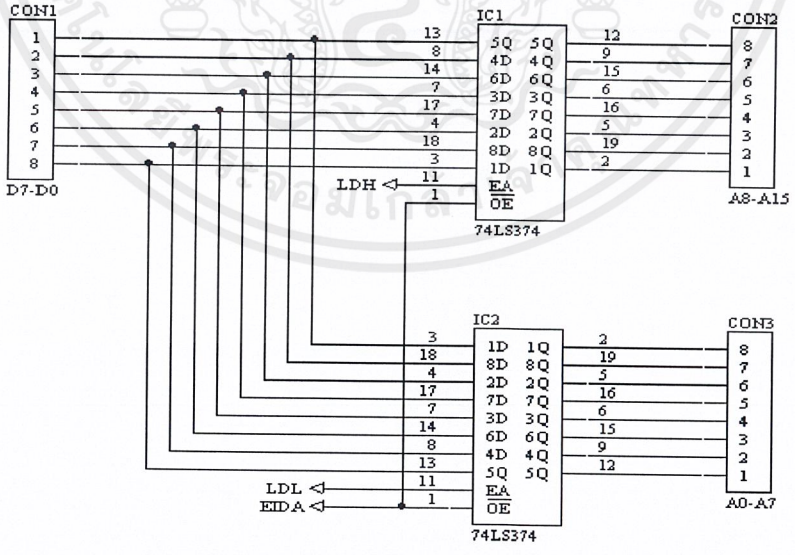
รูปที่ 3.9 อีพรอม ต่อขนาน 4 ตัวเพื่อควบคุม 32 ช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเอาต์พุตของ IR1 ซึ่งเป็นตัวกำหนดชุดคำสั่งว่าจะทำงานที่ชุดคำสั่งใด ถูกต่อกับ ขา แอดเดรสของอีพรอม 4 ตัวขนานกัน ทำหน้าที่เก็บรหัสการทำงานภายใน CPU โดย อีพรอม ทั้ง 4 ตัวนี้ขา Data D7 - D0 จะแยกกันทำให้มีขาควบคุมการทำงานภายใน CPU ถึง 32 ขาด้วยกันขา EP1-EP6 ทำการต่อเข้าวงจรดีโค้ดเดอร์ ไอซี 74LS138 เพื่อช่วยในการควบคุมการทำงานภายใน CPU ให้ได้หลายหน้าที่มากขึ้น ในส่วนนี้ให้ทำหน้าที่ควบคุมการโหลดข้อมูลลงรีจิสเตอร์ต่างๆ ขา EP7-EP12 ทำการต่อเข้าวงจรดีโค้ดเดอร์ ไอซี 74LS138 เพื่อช่วยในการควบคุมการทำงานภายใน CPU ให้ได้หลายหน้าที่มากขึ้น ในส่วนนี้ให้ทำหน้าที่ควบคุมการเปิดปิดสัญญาณตามจุดต่างๆ และ ขา แอดเดรส A0-A3 ของอีพรอม ที่ขนานกันทั้ง 4 ตัว ถูกต่อกับ 74LS193 และขา A11-A4 ของ อีพรอมที่ขนานกันทั้ง 4 ตัว ต่อกับเอาต์พุตของ IR1 การเอ็กซีคิ้ว นี้ทำโดยโค๊ดที่เก็บในอีพรอมทั้ง 4 ตัวขา A11-A4 เป็นตัวกำหนดแอดเดรส ของชุดคำสั่งและ A0-A3 ทำหน้าที่ทำขั้นตอนของคำสั่งนั้น ทีละขั้นตอนโดยสัญญาณนาฬิกา เข้ามา 1 ลูกขั้นตอนจะถูกเปลี่ยนไป 1 ขั้นตอนก็ขึ้นอยู่กับว่าคำสั่งนั้นเป็นคำสั่งอะไรและคำสั่งนั้นมีกี่ขั้นนตอนในการทำงาน โดยมีขาของอีพรอม 1 ขา เป็นตัวบอกว่าขณะนี้ได้ทำการเอ็กซ์ซีคิ้ว เสร็จแล้วให้ทำการเฟ็ทซ์ต่อการทำงานจะสลับไปมาระหว่างเฟ็ทซ์และ เอ็กซ์ซีคิ้ว อยู่อย่างนี้

3.2.4 ส่วนการเชื่อมต่อสัญญาณต่างๆ ภายใน CPU

1) วงจรแปลงสัญญาณกลุ่มข้อมูลเป็นกลุ่มสัญญาณตำแหน่ง



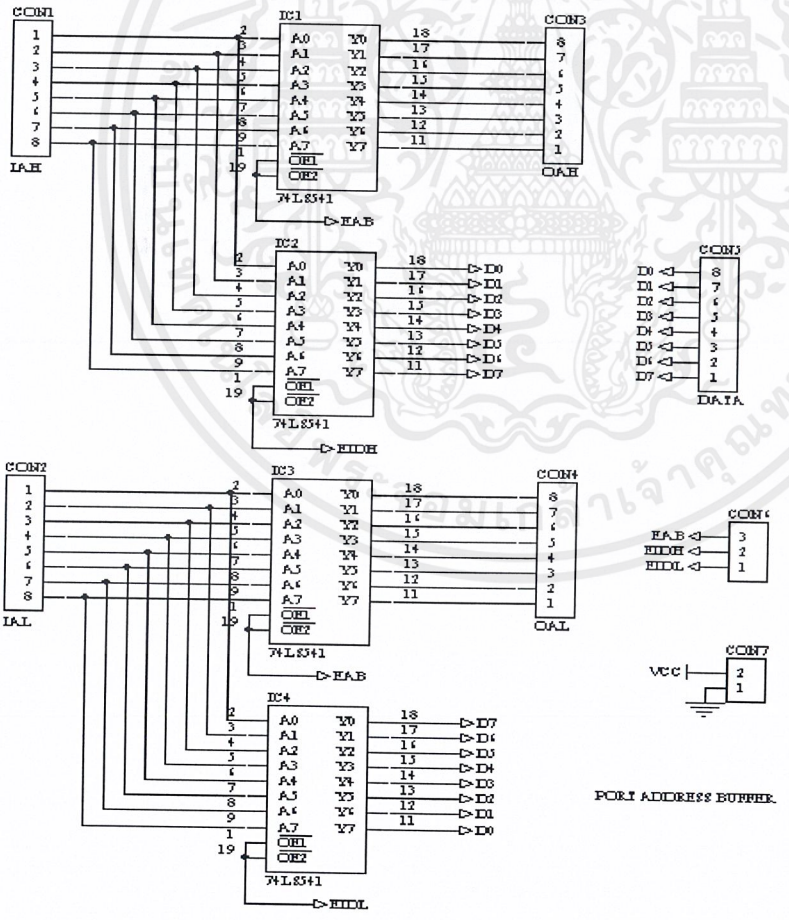
รูปที่ 3.10 วงจรแปลงสัญญาณกลุ่มข้อมูลเป็นกลุ่มสัญญาณตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการนำข้อมูลเลขฐานสอง 8 บิตที่เก็บในรีจิสเตอร์ทั่วไปอ้างสู่ตำแหน่งแอดเดรสบัสของ หน่วยความจำภายนอกโดยการนำค่าในรีจิสเตอร์ตัวแรกเก็บยังวงจรส่วนไบต์ต่ำ และนำค่าในรีจิสเตอร์ตัวที่สองเก็บยังวงจรส่วนไบต์สูง และนำค่าที่เก็บใช้อ้างตำแหน่งที่อยู่ของหน่วยความจำภายนอก

วงจรนี้ทำงานโดยใช้ ไอซี เบอร์ 74LS374 ภายในเป็นวงจร D ฟลิปฟลอปขนาด 8 บิต ที่เอาต์พุตมี ไตรสเตรท เป็นตัวกั้นเอาต์พุตได้ โดยอินพุตของ ไอซี 74LS374 ทั้งสองตัวต่อกับอินเตอร์นอลดาต้าบัสเมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัส ลงในไอซี 1 74LS374 ในส่วนไบต์ต่ำ เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัส ลงใน ไอซี 2 74LS374 ในส่วนไบต์สูง ขา EIDA ทำหน้าที่เปิดข้อมูลออกสู่อินเตอร์นอลแอดเดรสบัส วงจรนี้ใช้เมื่อต้องการอ้างข้อมูลจากรีจิสเตอร์คู่ออกอินเตอร์นอลแอดเดรสบัส

2) วงจรเปลี่ยนข้อมูลบัสตำแหน่งภายในเป็นข้อมูลภายใน



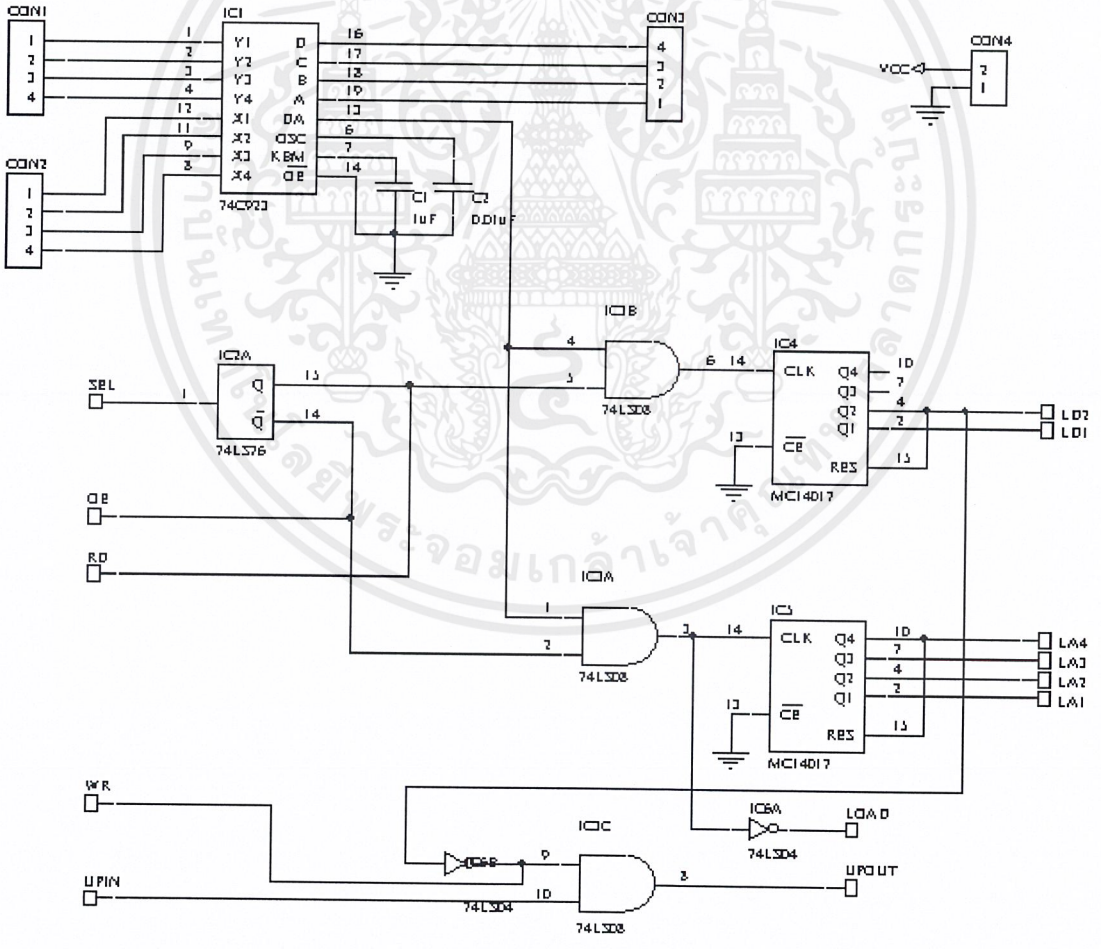
รูปที่ 3.11 วงจรเปลี่ยนข้อมูลบัสตำแหน่งภายในเป็นข้อมูลภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการนำข้อมูลเลขฐานสอง 8 บิตของ แอดเดรสบัส โดยการนำค่าของ แอดเดรสบัสตัวไบต์ต่ำส่งขึ้นอินเตอร์นอลดาต้าบัสแล้วเก็บยังรีจิสเตอร์ที่ต้องการ นำค่าของ แอดเดรสบัสตัวไบต์สูงส่งขึ้นอินเตอร์นอลดาต้าบัสแล้วเก็บยังรีจิสเตอร์ที่ต้องการ และวงจรนี้ยังควบคุมการเปิดปิดระหว่างอินเตอร์นอลดาต้าบัส และเอ็กเตอร์นอลแอดเดรสบัส อีกด้วย

วงจรนี้ทำงานโดยใช้ไอซีเบอร์ 74LS541 จำนวน 4 ตัวโดย ภายในเป็นวงจร ไตรสเตท ขนาด 8 บิต การใช้งานนั้น ไอซี 3 และไอซี 4 ทำหน้าควบคุมการเปิดปิดระหว่างอินเตอร์นอลแอดเดรสบัส และเอ็กเตอร์นอลแอดเดรสบัส และไอซี 1 ทำหน้าที่เปิดสัญญาณไบต์ต่ำสู่อินเตอร์นอลดาต้าบัส ไอซี 2 ทำหน้าที่เปิดสัญญาณไบต์สูงสู่อินเตอร์นอลดาต้าบัส

3.3 ส่วนการป้อนโปรแกรม



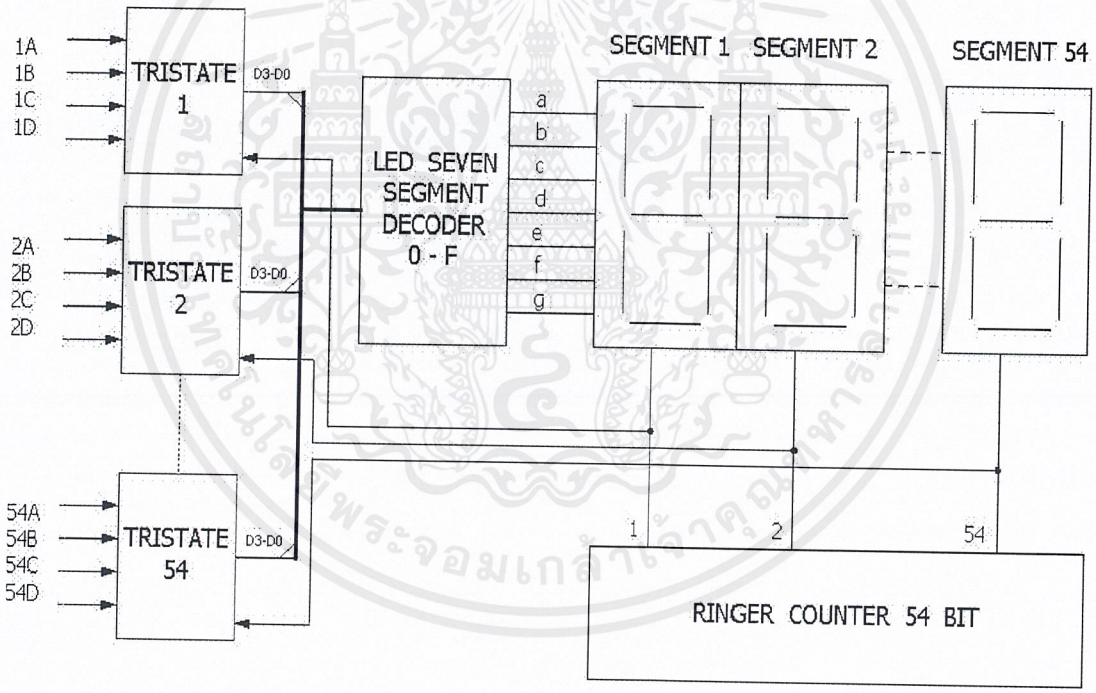
รูปที่ 3.12 วงจรเข้ารหัสคีย์เป็น BCD-8421

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำหน้ารับค่าได้จากการกดคีย์เพื่อแปลงค่าคีย์ที่ได้รับการกดเป็นข้อมูลรหัส BCD-8421 การกด มีได้16 คีย์มีค่าตั้งแต่ 0-F เมื่อได้รับการกดคีย์ 1 ครั้งจะมีลอจิก 1 ออกที่ขา DA ของไอซี 74C923 ซึ่งเป็น 1 นานตามคีย์กด การป้อนโปรแกรมครั้งแรกต้องทำการกำหนดแอดเดรส ที่ต้องการเริ่มโปรแกรมก่อน เมื่อป้อนแอดเดรส เสร็จจึงทำการเลือกโหมดเป็นดาต้า เมื่อทำการกดคีย์แล้วข้อมูล BCD-8421 จะปรากฏที่ดาต้า เมื่อกดครบ 2 หลักแอดเดรส จะเพิ่มขึ้น 1 ตำแหน่ง และทำการเขียนข้อมูลลงหน่วยความจำการทำงานก็เป็นไปตามลักษณะนี้จนเสร็จสิ้นการทดลองป้อน

3.4 ส่วนการแสดงผลข้อมูล

ทำหน้าที่แสดงผลเลข 0-F จำนวน 54 หลัก เพื่อแสดงค่าในรีจิสเตอร์ต่างๆ โดยใช้การสแกนดิสเพลย์

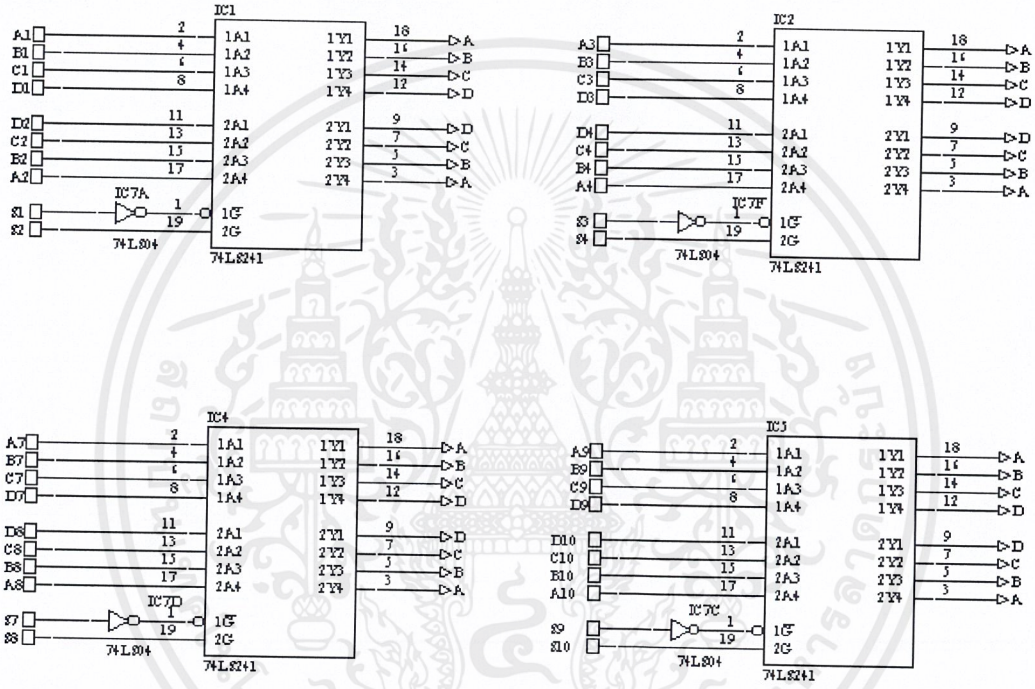


รูปที่ 3.13 ผังการทำงานของส่วนแสดงผล

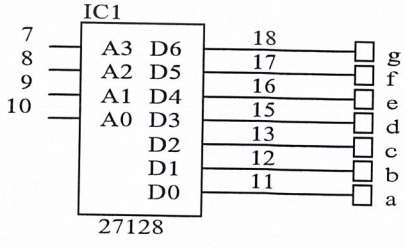
วงจรนี้ทำงานโดยใช้อีพรอม เก็บรหัสการแสดงผล 0-F ไว้ในตัวอีพรอม เช่น ถ้าแอดเดรสมีค่าเป็น 0100 ที่เอาต์พุต A, B, C, D, E, F และ G จะออกค่าตามที่กำหนดไว้ เพื่อขับ ส่วนแสดงผล 7 ส่วน เลข 4 ถ้าแอดเดรสมีค่าเป็น 1100 ที่เอาต์พุต A, B, C, D, E, F และ G จะออกค่าตามที่กำหนดไว้ เพื่อขับ ส่วนแสดงผล 7 ส่วน เลข C การเลือกข้อมูลที่ออกออก ส่วนแสดงผล 7 ส่วน หลักใดก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการให้หลักนั้นเป็น 1 ส่วนแสดงผล 7 ส่วน เป็นชนิดคอมมอนคาโถดทางด้านอินพุต BCD-8421 ถูกเลือกพร้อมกับ ส่วนแสดงผล 7 ส่วน หลักนั้น ซึ่งถูกเลือกโดยวงจรนับแบบวงแหวน ทางด้านอินพุตที่รับสัญญาณ BCD-8421 เป็นไตรสเตท ขนาด 4 บิต ซึ่งการแสดงผลแบบสแกนดิสเพลย์ นี้ต้องทำการแสกนที่ความถี่สูงประมาณ 1 kHz ขึ้นไป จึงทำให้ตามนุษย์มองเห็นค่าใน ส่วนแสดงผล 7 ส่วน ติดตลอด การกำเนิดสัญญาณนาฬิกาควบคุมการแสกนใช้ไอซี 555 เป็นตัวกำเนิดความถี่ออสซิลเลเตอร์ขนาด ประมาณ 1 kHz



รูปที่ 3.14 วงจรรับสัญญาณ BCD 8421 จำนวน 8 ชุด



รูปที่ 3.15 วงจรแปลงสัญญาณ BCD 8421ขับส่วนแสดงผล 7 ส่วน

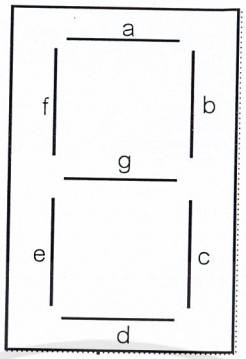
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมค่าลงในอีพรอมเพื่อใช้เป็นไอซีดีโค้ดเดอร์ขับตัวเลขแสดงผล 7 ส่วน

ตารางที่ 3.2 ค่าที่เขียนลงในอีพรอมเพื่อให้แสดงค่า 0-F

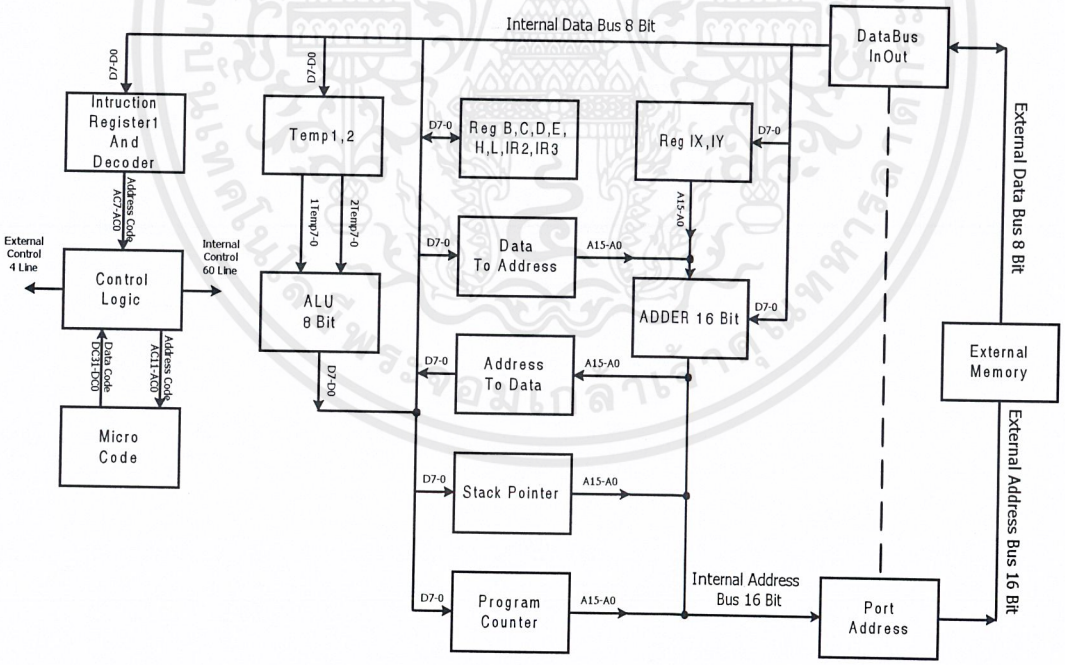
ตัวเลข	แอดเดรส	h	G	F	E	d	C	B	A	ค่าที่ป้อน
		D7	D6	D5	D4	D3	D2	D1	D0	
		8	4	2	1	8	4	2	1	
0	0000	0	0	1	1	1	1	1	1	3F
1	0001	0	0	0	0	0	1	1	0	06
2	0002	0	1	0	1	1	0	1	1	5B
3	0003	0	1	0	0	1	1	1	1	4F
4	0004	0	1	1	0	0	1	1	0	66
5	0005	0	1	1	0	1	1	0	1	6D
6	0006	0	1	1	1	1	1	0	1	7D
7	0007	0	0	0	0	0	1	1	1	07
8	0008	0	1	1	1	1	1	1	1	7F
9	0009	0	1	1	0	1	1	1	1	6F
A	000A	0	1	1	1	0	1	1	1	77
B	000B	0	1	1	1	1	1	0	0	7C
C	000C	0	0	1	1	1	0	0	1	39
D	000D	0	1	0	1	1	1	1	0	5E
E	000E	0	1	1	1	1	0	0	1	79
F	000F	0	1	1	1	0	0	0	1	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



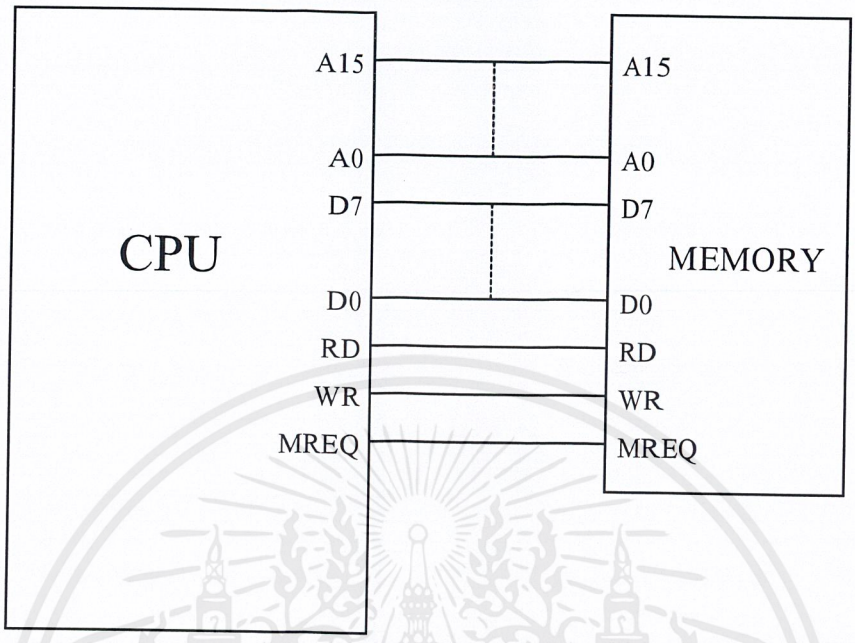
รูปที่ 3.16 ภาคแสดงผลตัวเลขแสดงผล 7 ส่วน

ต้องการตัวเลขแสดงผล 7 ส่วนจุดใดติด ให้แทนค่าด้วยลอจิก 1 เนื่องจากเป็นตัวเลขแสดงผลชนิดคาโถดร่วม

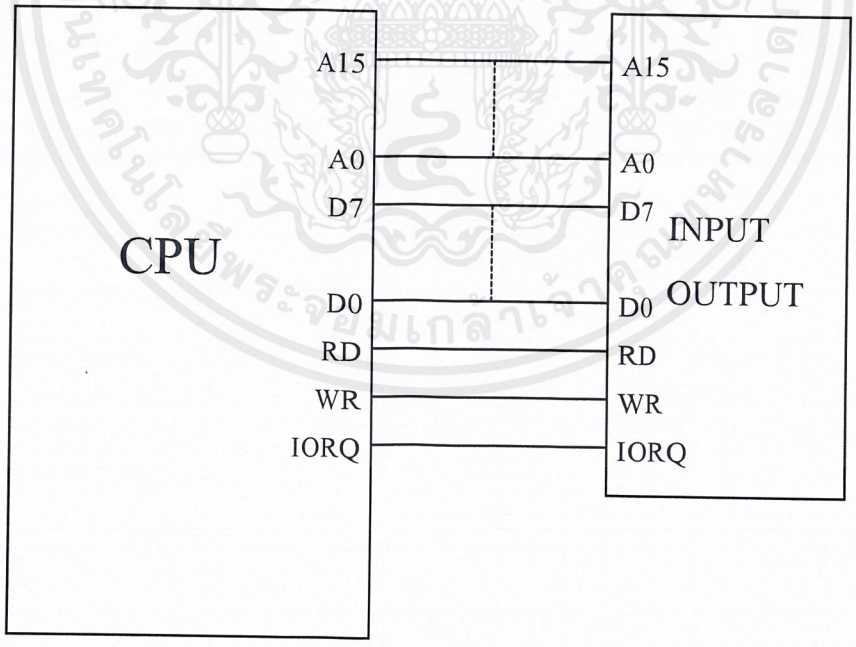


รูปที่ 3.17 เส้นทางของสัญญาณในชุดสาธิตสถาปัตยกรรมภายใน CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

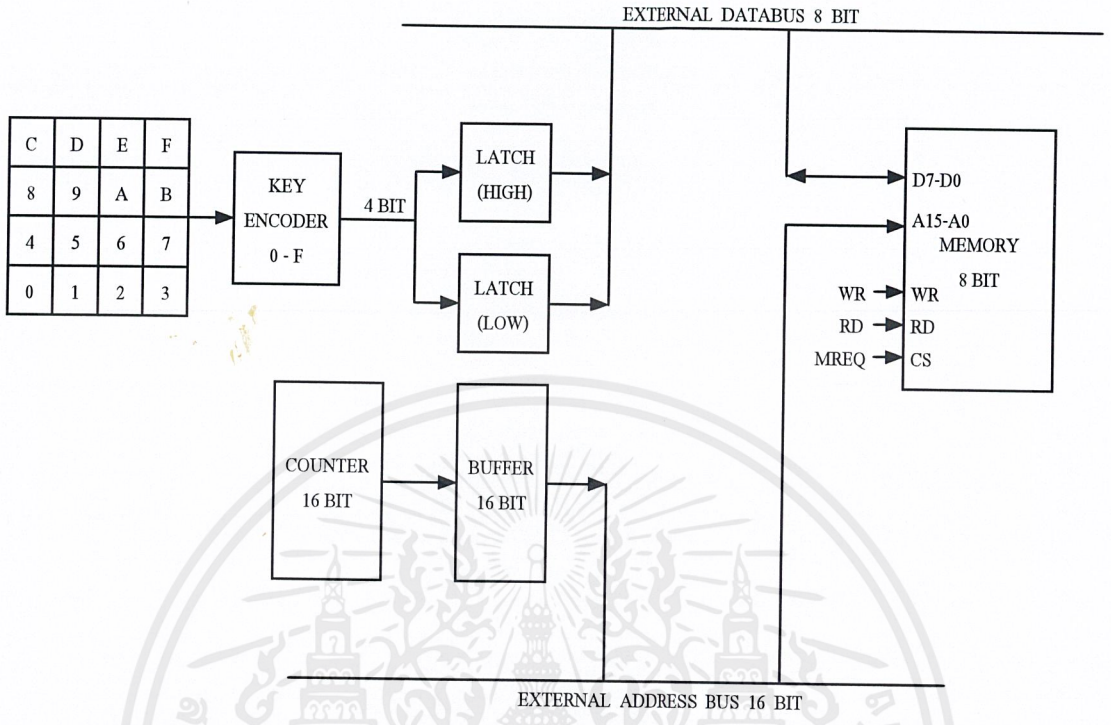


รูปที่ 3.18 การติดต่อกับหน่วยความจำภายนอกของชุดสาธิตสถาปัตยกรรมภายใน CPU



รูปที่ 3.19 การติดต่อกับอุปกรณ์ภายนอกของชุดสาธิตสถาปัตยกรรมภายใน CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 การป้อนโปรแกรมของชุดสาธิตสถาปัตยกรรมภายใน CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

ชุดสาธิตสถาปัตยกรรมภายใน CPU มีการแบ่งแผงวงจรเป็นส่วนต่างๆ ตามหน้าที่ซึ่งการทดลองจะแบ่งออกเป็นแต่ละแผ่นก่อนแล้วจึงประกอบรวมกันเป็นชุดสาธิตสถาปัตยกรรมภายใน CPU

4.1 การทดลองแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต

รูปที่ 4.1 แผงวงจรรีจิสเตอร์ทั่วไป 8 บิต

รีจิสเตอร์ทั่วไป 8 บิต จะมีรีจิสเตอร์ขนาด 8 บิตมีจำนวน 8 ตัว ประกอบด้วยรีจิสเตอร์ B, C, D, E, H, L, IR2, IR3 ซึ่งทั้ง 8 ตัวมีการทำงานเหมือนกัน คือ สามารถโหลดข้อมูลจากอินเตอร์นอลดาต้าบัสมาเก็บในตัวรีจิสเตอร์ และสามารถส่งค่าที่เก็บในตัวรีจิสเตอร์ไปยังอินเตอร์นอลดาต้าบัส เมื่อได้รับการสั่งจาก คอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 หน้าที่ของจุดเชื่อมต่อบนแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต

1) โดยกำหนดให้ LB เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ B

2) โดยกำหนดให้ LC เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ C

3) โดยกำหนดให้ LD เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ D

4) โดยกำหนดให้ LE เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ E

5) โดยกำหนดให้ LH เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ H

6) โดยกำหนดให้ LL เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ L

7) โดยกำหนดให้ LIR3 เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ IR 3

8) โดยกำหนดให้ LIR4 เป็น 0 ->1 เมื่อต้องการเก็บค่าจากอินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ IR 4

9) โดยกำหนดให้ EB เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ B ออกสู่อินเตอร์นอลดาต้าบัส

10) โดยกำหนดให้ EC เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ C ออกสู่อินเตอร์นอลดาต้าบัส

11) โดยกำหนดให้ ED เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ D ออกสู่อินเตอร์นอลดาต้าบัส

12) โดยกำหนดให้ EE เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ E ออกสู่อินเตอร์นอลดาต้าบัส

13) โดยกำหนดให้ EH เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ H ออกสู่อินเตอร์นอลดาต้าบัส

14) โดยกำหนดให้ EL เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ L ออกสู่อินเตอร์นอลดาต้าบัส

15) โดยกำหนดให้ EIR 2 เป็น 1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ IR 2 ออกสู่อินเตอร์นอลดาต้าบัส

16) โดยกำหนดให้ EIR 3 เป็น1 -> 0 เมื่อต้องการให้ข้อมูลในรีจีสเตอร์ IR 3 ออกสู่อินเตอร์นอลดาต้าบัส

17) โดยกำหนดให้ CLR เป็น 1-> 0 เมื่อต้องการรีเซ็ตค่าที่เก็บในรีจิสเตอร์ทุกตัวให้เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การทดลองโหลดข้อมูล 8 บิตจากบัสข้อมูลภายในลงในรีจิสเตอร์

ให้ทำการทดลองดังนี้

1) ป้อนไฟ + 5 โวลต์ เข้าวงจร

2) ป้อนข้อมูล 11001010 เข้าที่บัสข้อมูลภายใน

3) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LB ของ รีจิสเตอร์ B รีจิสเตอร์ B จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงตัวในรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ B ขณะนี้มีค่า 11001010

4) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LC ของรีจิสเตอร์ C รีจิสเตอร์ C จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงตัวในรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ C ขณะนี้มีค่า 11001010

5) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LD ของรีจิสเตอร์ D รีจิสเตอร์ D จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงตัวในรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ D ขณะนี้มีค่า 11001010

6) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LE ของ รีจิสเตอร์ E รีจิสเตอร์ E ซึ่งจะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ E ขณะนี้มีค่า 11001010

7) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LH ของ รีจิสเตอร์ H รีจิสเตอร์ H จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ H ขณะนี้มีค่า 11001010

8) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LL ของรีจิสเตอร์ Lรีจิสเตอร์ L จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ L ขณะนี้มีค่า 11001010

9) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LIR 2 ของ รีจิสเตอร์ IR2 รีจิสเตอร์ IR2 จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ IR 2 ขณะนี้มีค่า 11001010

10) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LIR 3 ของ รีจิสเตอร์ IR 3 รีจิสเตอร์ IR3 จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ IR 3 ขณะนี้มีค่า 11001010

4.1.3 การทดลองส่งข้อมูลที่เก็บในรีจิสเตอร์ออกสู่ อินเตอร์นอลดาต้าบัส

ให้ทำการทดลองดังนี้

1) ปลดสัญญาณ 11001010 ที่ป้อนให้กับอินเตอร์นอลดาต้าบัสออก

2) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EB รีจิสเตอร์ B จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010 ถ้าขา EB เป็น 1 ข้อมูลไม่ถูกส่งออก

3) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EC รีจิสเตอร์ C จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010 ถ้าขา EC เป็น 1 ข้อมูลไม่ถูกส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา ED รีจิสเตอร์ D จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010

5) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EE รีจิสเตอร์ E จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010

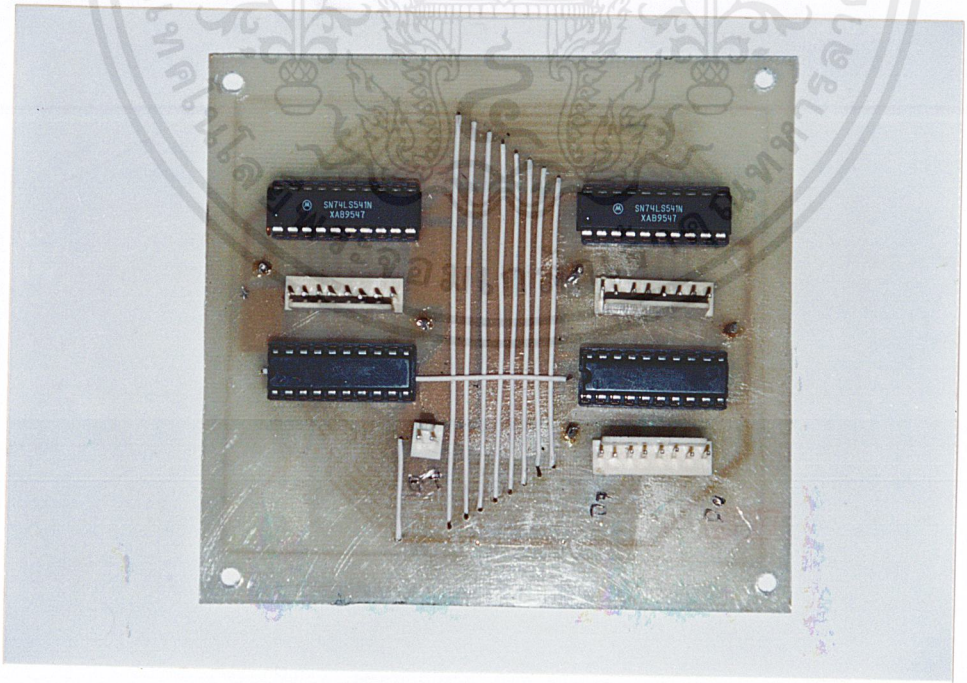
6) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EH รีจิสเตอร์ H จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010

7) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EL รีจิสเตอร์ L จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010 ถ้าขา EL เป็น 1 ข้อมูลไม่ถูกส่งออก

8) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EIR 2 รีจิสเตอร์ IR 2 จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัสจะเป็น 11001010 ถ้าขา EIR 2 เป็น 1 ข้อมูลจะไม่ถูกส่ง

9) ให้ ขา CLR เป็น 0 ค่าในรีจิสเตอร์ทุกตัวจะมีค่าเป็น 0

4.2 การทดลองแผงวงจรรีจิสเตอร์ A, F



รูปที่ 4.2 แผงวงจรรีจิสเตอร์ A-F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 หน้าที่ของจุดเชื่อมต่อบนแผงวงจรรีจิสเตอร์ทั่วไป 8 บิต

1) โดยกำหนดให้ LA เป็น 0 ->1 เมื่อต้องการเก็บค่าจาก อินเตอร์นอลดาต้าบัสลงรีจีสเตอร์ A

2) โดยกำหนดให้ LF เป็น 0 ->1 เมื่อต้องการเก็บค่าจาก อินเตอร์นอลดาต้าบัสลงรีจิสเตอร์ F

3) โดยกำหนดให้ EA เป็น 1->0 เมื่อต้องการที่จะให้ข้อมูลในรีจิสเตอร์ Aออกสู่อินเตอร์นอลดาต้าบัส

4) โดยกำหนดให้ EF เป็น 1-> 0 เมื่อต้องการที่จะให้ข้อมูลในรีจิสเตอร์ F ออกสู่อินเตอร์นอลดาต้าบัส

5) โดยกำหนดให้ CLR เป็น 1 -> 0 เมื่อต้องการรีเซตค่าที่เก็บในรีจิสเตอร์ทุกตัวให้เป็น 0

4.2.2 การทดลองโหลดข้อมูล 8 บิตจากอินเตอร์นอลดาต้าบัสลงในรีจิสเตอร์ A, F

ให้ทำการทดลองดังนี้

1) ป้อนแรงดัน + 5 โวลต์ เข้าวงจร

2) ป้อนข้อมูล 10011010 เข้าที่อินเตอร์นอลดาต้าบัส

3) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LA ของ รีจิสเตอร์ A รีจิสเตอร์ A จะเก็บค่าจากอินเตอร์นอลดาต้าบัสลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ A ขณะนี้มีค่า 10011010

4) ป้อนสัญญาณจาก 0 เป็น 1 ที่ขา LF ของ รีจิสเตอร์ F รีจิสเตอร์ F จะเก็บค่าที่ได้จากอินเตอร์นอลดาต้าบัส ลงในตัวรีจิสเตอร์นั้น ซึ่งในรีจิสเตอร์ F ขณะนี้มีค่า 10011010

4.2.3 การทดลองส่งข้อมูลที่เก็บในรีจิสเตอร์ออกสู่อินเตอร์นอลดาต้าบัส

ให้ทำการทดลองดังนี้

1) ปลดสัญญาณ 10011010 ที่ป้อนให้กับ อินเตอร์นอลดาต้าบัสออก

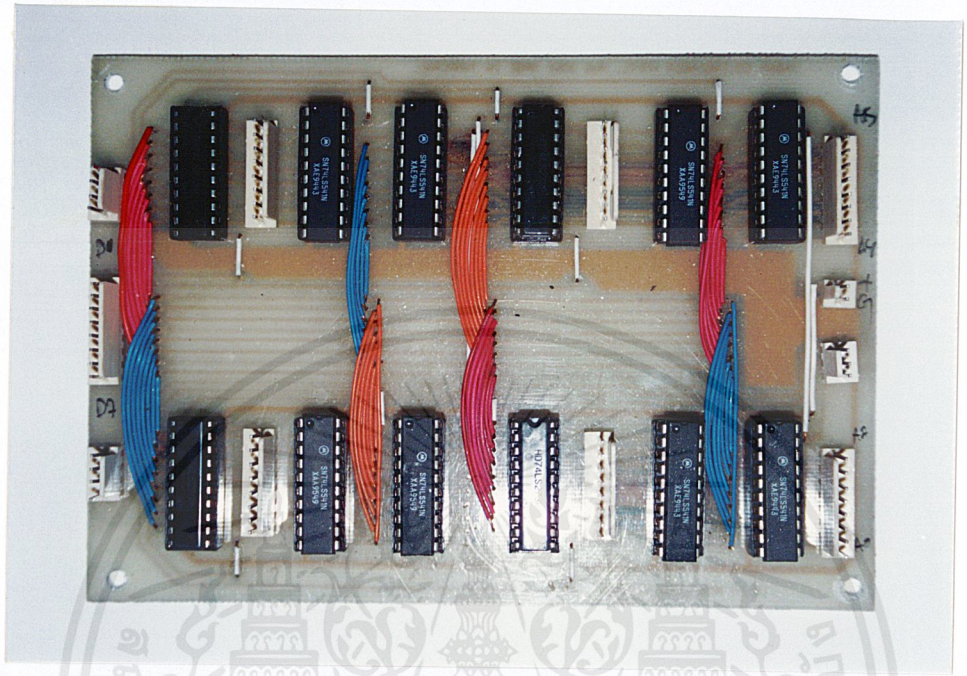
2) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EA รีจิสเตอร์ A จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าใน อินเตอร์นอลดาต้าบัสจะเป็น 10011010 ถ้าขา EA เป็น 1 ข้อมูลไม่ถูกส่งออก

3) ป้อนสัญญาณจาก 1 เป็น 0 ที่ขา EF รีจิสเตอร์ F จะส่งค่าที่เก็บไว้ในตัวส่งออกสู่อินเตอร์นอลดาต้าบัสซึ่งค่าในอินเตอร์นอลดาต้าบัส เป็น 10011010 ถ้าขา EF เป็น 1 ข้อมูลไม่ถูกส่งออก

4) ให้ ขา CLR เป็น 0 ค่าในรีจิสเตอร์ทุกตัวจะมีค่าเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองแผงวงจรรีจิสเตอร์ IX, IY



รูปที่ 4.3 แผงวงจรรีจิสเตอร์ IX, IY

แผงวงจรรีจิสเตอร์ IX, IY ซึ่งประกอบด้วยรีจิสเตอร์ 16 บิต 2 ตัวคือรีจิสเตอร์ IX และรีจิสเตอร์ IY ทำหน้าที่เก็บข้อมูลขนาด 16 บิต ไว้ใช้อ้างแอดเดรสของหน่วยความจำภายนอก

4.3.1) การทดลอง

1) โดยกำหนดให้ LIXH เป็น 0->1 เมื่อต้องการเก็บค่า D7-D0 ไว้ในรีจิสเตอร์ IX A15-A8

2) โดยกำหนดให้ LIXL เป็น 0->1 เมื่อต้องการเก็บค่า D7-D0 ไว้ในรีจิสเตอร์ IX A7-A0

3) โดยกำหนดให้ LIYH เป็น 0->1 เมื่อต้องการเก็บค่า D7-D0 ไว้ในรีจิสเตอร์ IY A15-A8

4) โดยกำหนดให้ LIYL เป็น 0->1 เมื่อต้องการเก็บค่า D7-D0 ไว้ในรีจิสเตอร์ IY A7-A0

5) โดยกำหนดให้ EIXH เป็น 1->0 เมื่อต้องการนำค่าที่อยู่ในรีจิสเตอร์ IX A15-A8 ออกสู่อินเตอร์นอลดาต้าบัส D7-D0

6) โดยกำหนดให้ EIXL เป็น 1->0 เมื่อต้องการนำค่าที่อยู่ในรีจิสเตอร์ IX A7-A0 ออกสู่อินเตอร์นอลดาต้าบัส D0- D7

7) โดยกำหนดให้ EIYH เป็น 1->0 เมื่อต้องการนำค่าที่อยู่ในรีจิสเตอร์ IX A15-A8 ออกสู่อินเตอร์นอลดาต้าบัส D7-D0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) โดยกำหนดให้ EIYL เป็น 1->0 เมื่อต้องการนำค่าที่อยู่ในรีจิสเตอร์ IX A7-A0 ออกสู่อินเตอร์นอลดาต้าบัส D7-D0

9) โดยกำหนดให้ EIX เป็น 1->0 เมื่อต้องการส่งข้อมูลที่อยู่ในรีจิสเตอร์ IX ออกสู่แอดเดรสบัส A15-A0

10) โดยกำหนดให้ EIY เป็น 1->0 เมื่อต้องการส่งข้อมูลที่อยู่ในรีจิสเตอร์ IY ออกสู่แอดเดรสบัส A15-A0

11) โดยกำหนดให้ CLR เป็น 1->0 เมื่อต้องการ รีเซตค่าในรีจิสเตอร์ IX, IY ให้เป็น 0

4.3.2) ผลการทดลอง

1) เมื่อ LIXH เป็น 1 ค่าใน อินเตอร์นอลดาต้าบัส D7-D0 ไว้ในรีจิสเตอร์ IXH A15-A8

2) เมื่อ LIXH เป็น 1 ค่าใน อินเตอร์นอลดาต้าบัส D7-D0 ไว้ในรีจิสเตอร์ IXL A15-A8

3) เมื่อ LIYH เป็น 1 ค่าใน อินเตอร์นอลดาต้าบัส D7-D0 ไว้ในรีจิสเตอร์ IYH A15-A8

4) เมื่อ LIYL เป็น 1 ค่าใน อินเตอร์นอลดาต้าบัส D7-D0 ไว้ในรีจิสเตอร์ IYL A7-A0

5) เมื่อ EIXH เป็น 0 เมื่อต้องการนำค่าในรีจิสเตอร์ IX A15-A8 สู่อินเตอร์นอลดาต้าบัส D7-D0

6) เมื่อ EIXL เป็น 0 เมื่อต้องการนำค่าในรีจิสเตอร์ IX A7-A0 สู่อินเตอร์นอลดาต้าบัส D7-D0

7) เมื่อ EIYH เป็น 0 เมื่อต้องการนำค่าในรีจิสเตอร์ IX A15-A8 สู่อินเตอร์นอลดาต้าบัส D7-D0

8) เมื่อ EIYL เป็น 0 เมื่อต้องการนำค่าในรีจิสเตอร์ IX A7-A0 สู่อินเตอร์นอลดาต้าบัส D7-D0

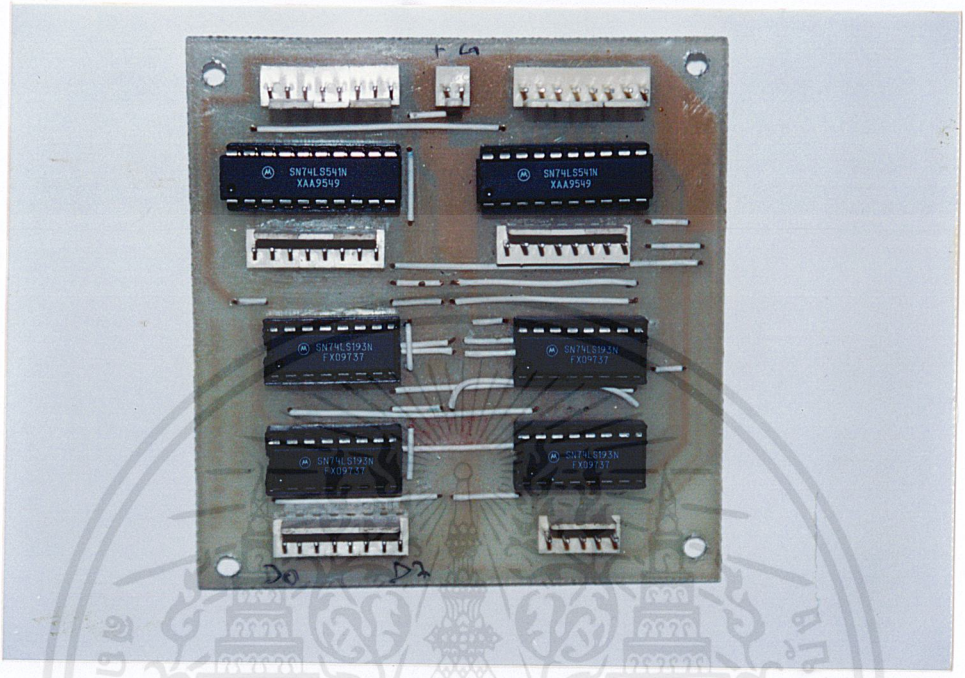
9) เมื่อ EIX เป็น 0 เมื่อต้องการส่งข้อมูลในรีจิสเตอร์ IX ออกสู่ แอดเดรสบัส A15-A0

10) เมื่อ EIY เป็น 0 เมื่อต้องการส่งข้อมูลในรีจิสเตอร์ IY ออกสู่ แอดเดรสบัส A15-A0

11) เมื่อ CLR เป็น 0 ค่าที่เก็บจะเป็น 0 ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองแผงวงจรสแตกพอยน์เตอร์ 16 บิต



รูปที่ 4.4 แผงวงจรสแตกพอยน์เตอร์ 16 บิต

สแตกพอยน์เตอร์ 16 บิตทำหน้าที่ชี้ตำแหน่งหน่วยความจำภายนอกที่ CPU นำข้อมูลไปพักไว้ แผ่นวงจรนี้ มีเอาต์พุต คือ แอดเดรสบัส A15 - A0 สามารถเลื่อนค่าขึ้นและลงได้และมีอินพุต คือ อินเตอร์นอลดาต้าบัส D7 - D0 มีขาการควบคุมจากคอลโทรลลอจิก คือ

สภาวะปกติให้ป้อนขา CLR เป็น 0 และขา ESP, DSP, USP, LSPH, LSPL เป็น 1

1) โดยกำหนดให้ขา ESP เป็น 1 -> 0 เพื่อนำค่า A15-A0 ออกสู่ อินเตอร์แอดเดรสบัส A15 – A0

2) โดยกำหนดให้ขา CLR เป็น 1 -> 0 เมื่อต้องการรีเซตค่าที่เก็บในสแตกพอยน์เตอร์ทั้งหมด

3) โดยกำหนดให้ขาDSP เป็น 1 -> 0 เพื่อลดค่าแอดเดรสที่เก็บในสแตกพอยน์เตอร์ลง 1 ค่า

4) โดยกำหนดให้ขาUSP เป็น 1 -> 0 เพื่อเพิ่มค่าแอดเดรสที่เก็บในสแตกพอยน์เตอร์ขึ้น 1 ค่า

5) โดยกำหนดให้ขา LSPH เป็น 1 -> 0 เพื่อเก็บค่าจากอินเตอร์ดาต้าบัสลงใน A15-A8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) โดยกำหนดให้ขา LSPL เป็น 1 -> 0 เพื่อเก็บค่าจาก อินเตอร์ดาต้าบัสลงใน A7-A0

4.4.1) การทดลอง

1) ป้อนสัญญาณ 10010011 เข้าที่ดาต้าบัส ของ สแตกพอยน์เตอร์

2) ป้อนสัญญาณให้ขา CLR เป็น 0 และขา ESP, DSP, USP, LSPH, LSPL เป็น 1

3) เมื่อให้ขา LSPH เป็น 1 -> 0 ค่า 10010011 จากอินเตอร์นอลดาต้าบัส ถูกเก็บใน A15-A8

4) เมื่อให้ขา LSPL เป็น 1 -> 0 ค่า 10010011 จากอินเตอร์นอลดาต้าบัสถูกเก็บใน A7-A0

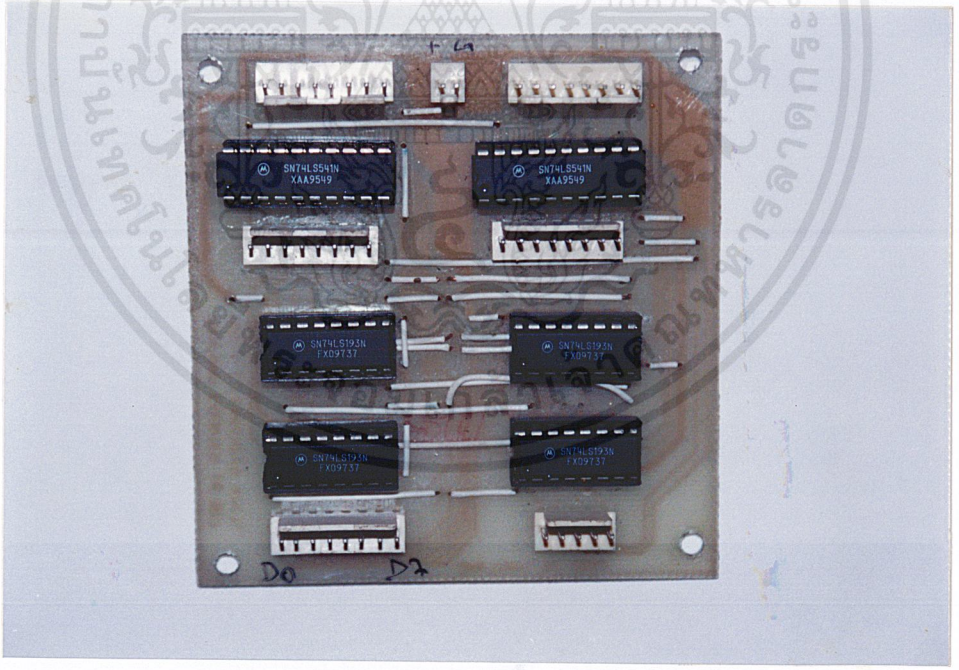
5) เมื่อให้ขา ESPเป็น 0 ค่าที่เก็บในสแตกพอยน์เตอร์จะส่งออกสู่อินเตอร์นอลแอดเดรสบัส

6) เมื่อให้ขา CLR เป็น 0 -> 1 ค่าที่เก็บใน สแตกพอยน์เตอร์ทั้งหมดมีค่าเป็น 0

7) เมื่อให้ขา DSP เป็น 1 -> 0 ค่า แอดเดรส ที่เก็บใน สแตกพอยน์เตอร์ลดลง 1 ค่า

8) เมื่อให้ขา USP เป็น 1 -> 0 ค่า แอดเดรสที่เก็บใน สแตกพอยน์เตอร์เพิ่มขึ้น 1 ค่า

4.5 การทดลองแผงวงจรโปรแกรมเคาน์เตอร์ 16 บิต



รูปที่ 4.5 แผงวงจรโปรแกรมเคาน์เตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเคาน์เตอร์ 16 บิตทำหน้าที่ชี้ตำแหน่งหน่วยความจำภายนอกที่ CPU ทำการเฟ็ทซ์ คำสั่งเข้ามาไว้ใน CPU แผ่นวงจรนี้ มีเอาต์พุต คือ แอดเดรสบัส A15 - A0 สามารถทำการเลื่อนค่าขึ้นได้อย่างเดียว และมีอินพุต คือ อินเตอร์นอลดาต้าบัส D7 - D0 มีขาการควบคุมจากคอนโทรลลอจิก คือ

สภาวะปกติให้ป้อนขา CLR เป็น 0 และขา EPC, UPC, LPCH, LPCL เป็น 1

1) โดยกำหนดให้ขาEPC เป็น 1 -> 0 เพื่อนำค่า A15-A0 ออกสู่ อินเตอร์แอดเดรสบัส A15 – A0

2) โดยกำหนดให้ขา CLR เป็น 1 -> 0 เมื่อต้องการ รีเซตค่าที่เก็บในโปรแกรมเคาน์เตอร์ทั้งหมด

3) โดยกำหนดให้ขา UPC เป็น 1 -> 0 เพื่อเพิ่มค่าแอดเดรสที่เก็บในโปรแกรมเคาน์เตอร์ขึ้น 1 ค่า

4) โดยกำหนดให้ขา LPCH เป็น 1 -> 0 เพื่อเก็บค่าจากอินเตอร์นอลดาต้าบัสลงใน A15-A8

5) โดยกำหนดให้ขา LPCL เป็น 1 -> 0 เพื่อเก็บค่าจาก อินเตอร์นอลดาต้าบัสลงใน A7-A0

4.5.1) การทดลอง

1) ป้อนสัญญาณ 10010011 เข้าที่ดาต้าของสแตกพอยน์เตอร์

2) ป้อนสัญญาณให้ขา CLR เป็น 0 และขา EPC, UPC, LPCH, LPCL เป็น 1

3) เมื่อให้ขา LPCH เป็น 1 -> 0 ค่า 10010011จากอินเตอร์นอลดาต้าบัสถูกเก็บใน A15-A8

4) เมื่อให้ขา LPCL เป็น 1 -> 0 ค่า 10010011จากอินเตอร์นอลดาต้าบัสถูกเก็บใน A7-A0

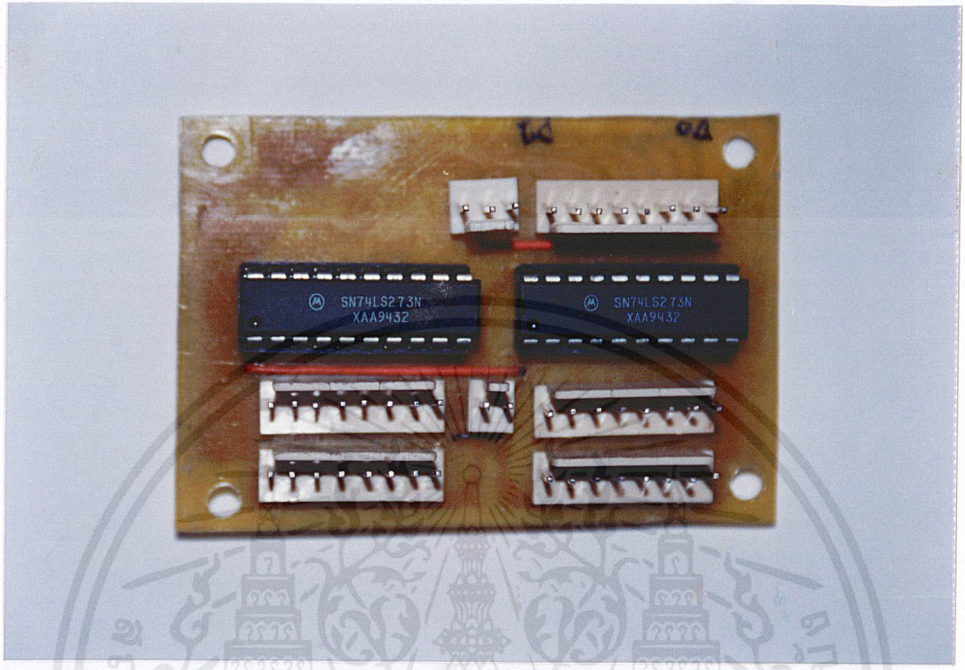
5) เมื่อให้ขา EPC เป็น 0 ค่าที่เก็บในโปรแกรมเคาน์เตอร์จะส่งออกสู่ อินเตอร์นอลดาต้าบัส

6) เมื่อให้ขา CLR เป็น 0 -> 1 ค่าที่เก็บในโปรแกรมเคาน์เตอร์ทั้งหมดมีค่าเป็น 0

7) เมื่อให้ขา UPC เป็น 1 -> 0 ค่า แอดเดรสที่เก็บในโปรแกรมเคาน์เตอร์เพิ่มขึ้น 1 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลองแผงวงจรรีจิสเตอร์พักข้อมูล



รูปที่ 4.6 แผงวงจรรีจิสเตอร์พักข้อมูล

4.6.1) การทดลอง วงจรรีจิสเตอร์พักข้อมูล (Temp Register)

1) โดยกำหนดให้ CLR เป็น 0 เมื่อต้องการรีจิสเตอร์

2) โดยกำหนดให้ LT1 เป็น 0->1 เมื่อต้องการเก็บข้อมูลจากดาต้าบัสเข้า TEMP 1

3) โดยกำหนดให้ LT2 เป็น 0->1 เมื่อต้องการเก็บข้อมูลจากดาต้าบัส เข้า TEMP 2

4.6.2) ผลการทดลอง

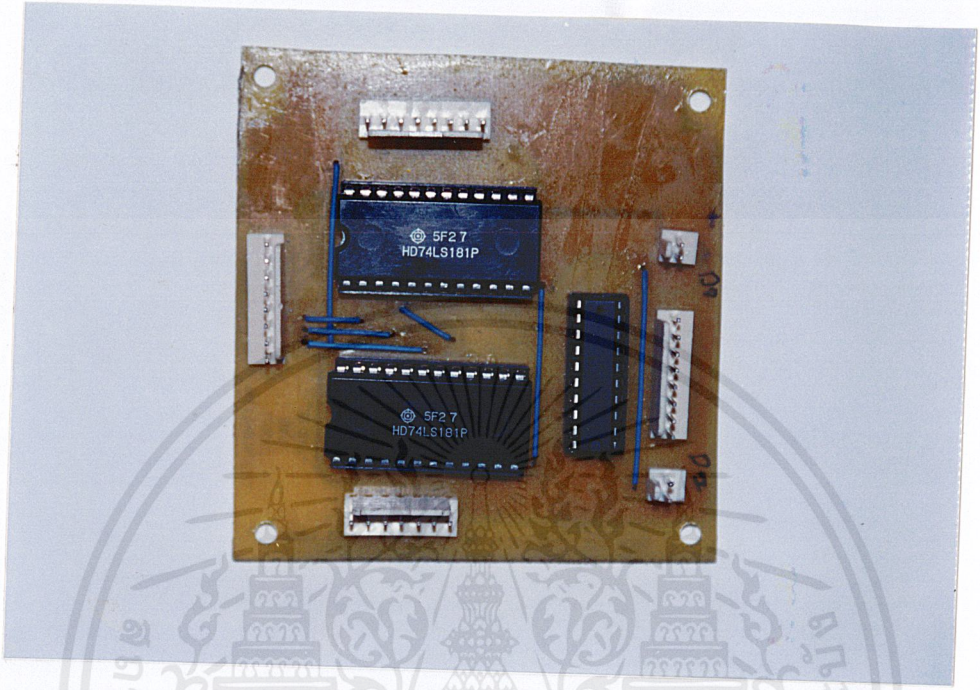
1) เมื่อให้ CLR เป็น 0 เอาต์พุตทั้งหมดมีค่าเป็น 0

2) เมื่อให้ LT1 เป็น 0->1 ข้อมูลจากดาต้าบัสถูกเก็บเข้า TEMP 1

3) เมื่อให้ LT2 เป็น 0->1 ข้อมูลจากดาต้าบัสถูกเก็บเข้า TEMP 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การทดลองแผงวงจรคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก 8 บิต



รูปที่ 4.7 แผงวงจรคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก 8 บิต

แผ่นวงจร ALU 8 บิต ทำหน้าที่ประมวลผลคำสั่งทางคณิตศาสตร์ เช่น การบวกการลบ และกระทำทางลอจิก เช่น AND, OR, XOR เป็นต้น

การต่อแผ่นวงจรมีดังนี้

1) โดยกำหนดให้ 1T1 - 1T8 ต่อกับเอาต์พุตของแผงวงจรรีจิสเตอร์พักข้อมูล

2) โดยกำหนดให้ 1T1 - 2T8 ต่อกับเอาต์พุตของแผงวงจรรีจิสเตอร์พักข้อมูล

3) โดยกำหนดให้ LALU8 เป็น 0->1 เมื่อต้องการแลตช์ค่าผลลัพธ์ไว้

4) โดยกำหนดให้ EALU8 เป็น 1->0 เมื่อต้องการส่งค่าผลลัพธ์สู่อินเตอร์นอลดาต้าบัส

5) โดยกำหนดให้ MODE, CI, CO, S3, S2, S1, S0 ดูตารางการใช้งานของไอซี 74181

4.7.1) การทดลอง

1) ให้ทำการป้อนค่า 11100011 เข้าที่ Temp1

2) ให้ทำการป้อนค่า 00000111 เข้าที่ Temp2

3) ใช้งานไอซี 74181 เป็นโหมดบวกเลข โดยป้อนลอจิก 1 ที่ขา S3, S0, CI ป้อนลอจิก 0 ที่ขา S2, S1, MODE

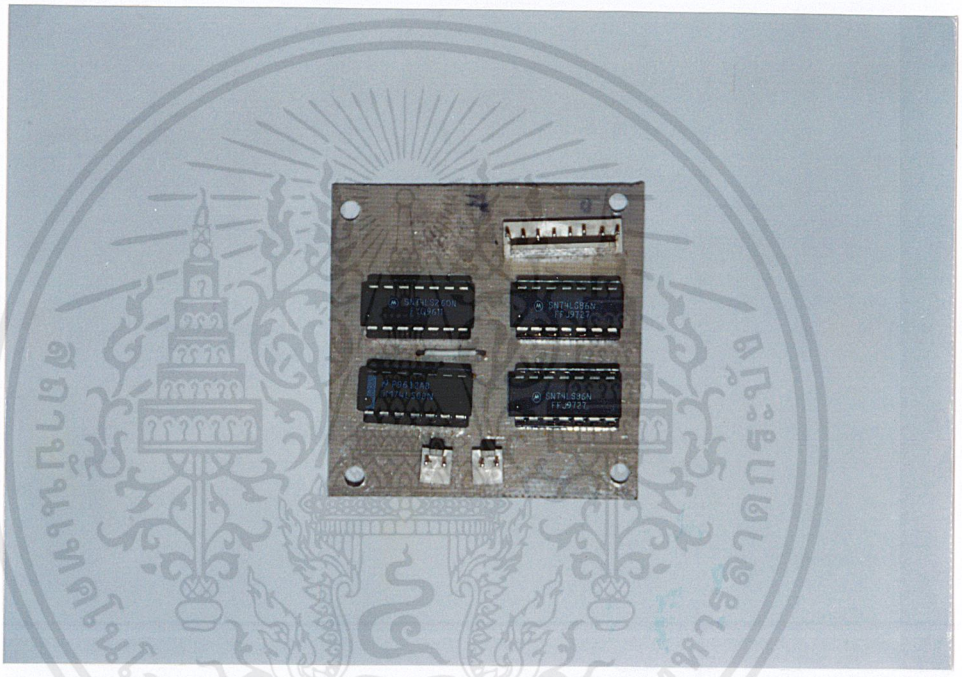
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ป้อนลอจิก 0->1 ให้ขา LALU8 เพื่อแลตช์ค่าผลลัพธ์ไว้

5) ป้อนลอจิก 1->0 ให้ขา EALU8 เพื่อส่งค่าผลลัพธ์สู่ อินเตอร์นอลดาต้าบัส

6) ดูสัญญาณที่อินเตอร์นอลดาต้าบัสจะได้ค่า 11101010 แสดงว่าบวกเลขถูกต้อง

4.8 การทดลองแผงวงจรตรวจสอบแฟลกศูนย์ และแฟลกพาริตี้



รูปที่ 4.8 แผงวงจรตรวจสอบแฟลกศูนย์ และแฟลกพาริตี้

แผงวงจรตรวจสอบ แฟลกศูนย์ และแฟลกพาริตี้บิตทำหน้าที่ตรวจคำตอบที่ส่งออกจาก CPU ว่าคำตอบมีสถานะเป็น 0 หรือไม่ และคำตอบนั้นเป็นบิตคู่หรือบิตคี่

4.8.1) การทดลอง

1)ป้อนสัญญาณ D7-D0 ที่อินพุต = 00001011 สังเกตเอาต์พุตที่ได้

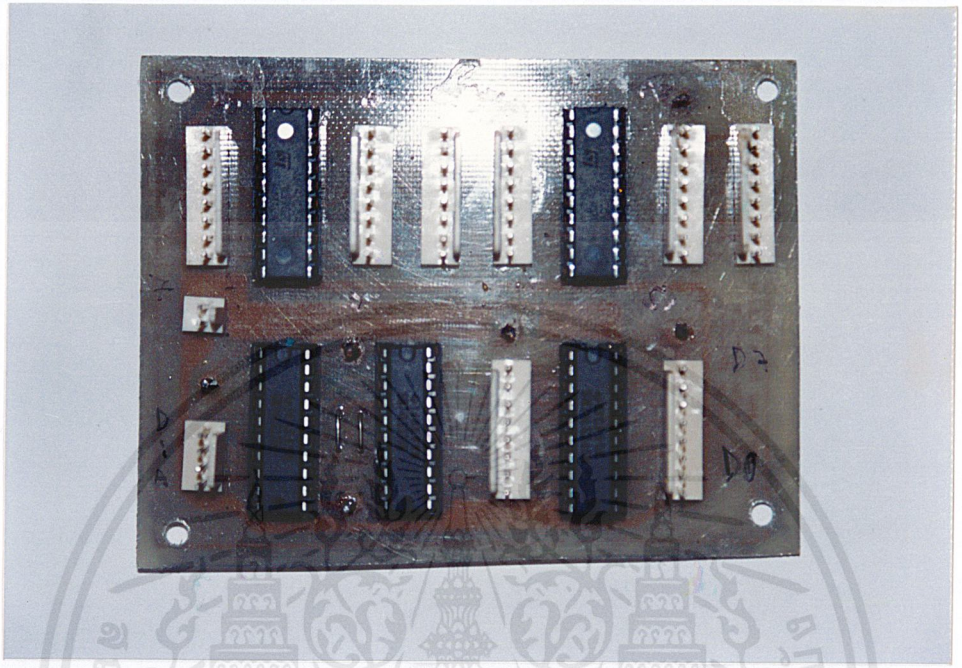
4.8.2) ผลการทดลอง

1) เอาต์พุต 0 เป็น 0 เมื่อคำตอบมีค่าตั้งแต่ 1 ขึ้นไป

2) เอาต์พุตพาริตี้ เป็น 1 เมื่อคำตอบเป็นบิตคี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การทดลองแผงวงจรพอร์ต



รูปที่ 4.9 แผงวงจรพอร์ต

4.9.1) การทดลอง

1) ป้อนสัญญาณ 1011 เข้าอินพุต ABCD

2) ป้อนสัญญาณ 0-> 1 ให้จุด L1 เพื่อต้องการเก็บข้อมูล 1011 เข้าใน D7-D4

3) ป้อนสัญญาณ 1010 เข้าอินพุต ABCD

4) ป้อนสัญญาณ 0-> 1 ให้จุด L2 เพื่อต้องการเก็บข้อมูล 1010 เข้าใน D3-D0

5) ป้อนสัญญาณ 0 ให้จุด OE1 เพื่อให้ D7-D0 ที่เกิดจากการกดคีย์ ออกดาต้าบัสภายนอก

6) ป้อนสัญญาณ 11110000 ให้แอดเดรส A15 – A8

7) ป้อนสัญญาณ 00001111 ให้แอดเดรส A7 – A0

8) ป้อนสัญญาณ 0 ให้จุด OE3 เพื่อให้แอดเดรสอินพุตออกสู่บัสข้อมูลภายนอก

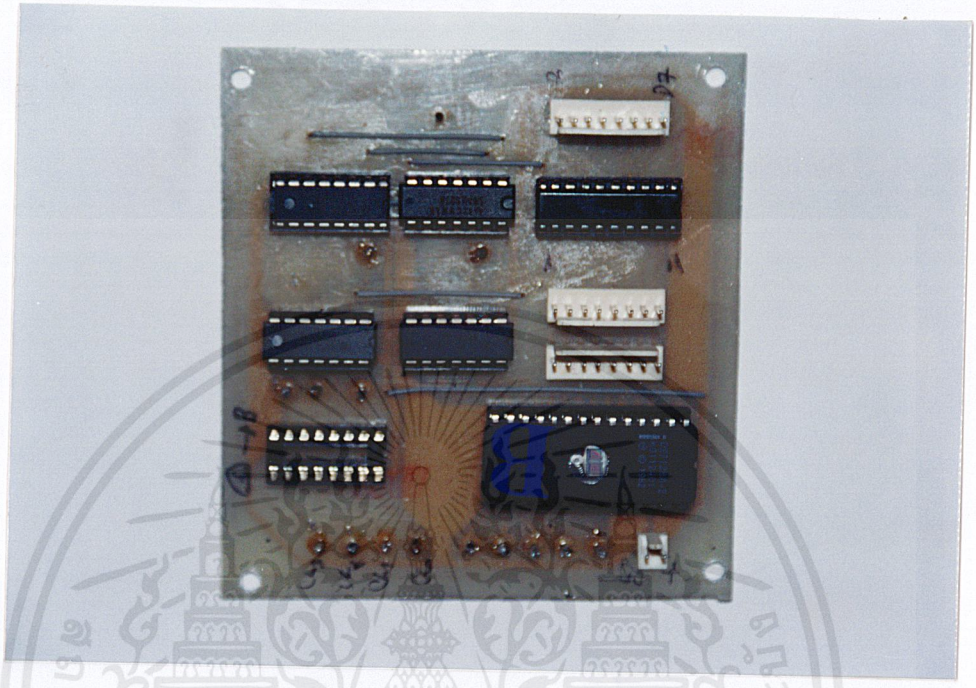
4.9.2) ผลการทดลอง

1) สังเกตเอาต์พุต D7 - D0 จะได้ 10111010

2) สังเกตเอาต์พุต A15 - A0 จะได้ 1111000000001111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 การทดลองแผงวงจรเฟ็ทช์



รูปที่ **4.10** แผงวงจรเฟ็ทช์

4.10.1) การทดลอง

1) ป้อนสัญญาณที่จุดรีจิสเตอร์เป็น 0

2) ป้อนสัญญาณนาฬิกาเข้าที่จุดสัญญาณนาฬิกาทีละลูก

3) ป้อนสัญญาณ 10110000 เข้าที่ดาต้าบัส D7 – D0

4.10.2) ผลการทดลอง

1) เมื่อป้อนสัญญาณนาฬิกาลูกแรก เอาต์พุตที่จุดเฟ็ทช์ เป็น 0

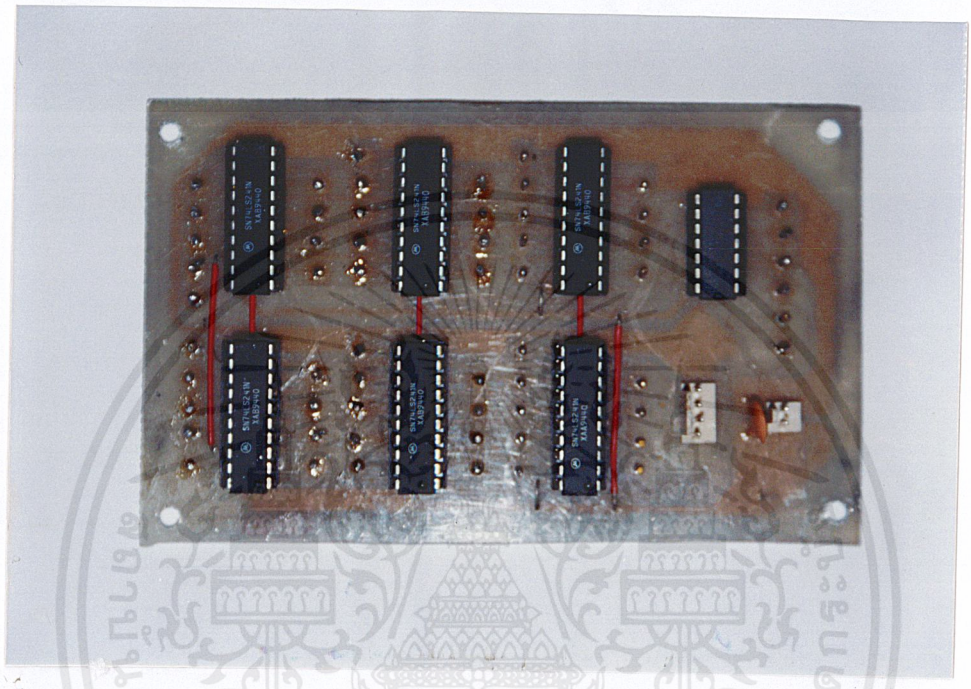
2) เมื่อป้อนสัญญาณนาฬิกาลูกที่ 2 สัญญาณจุด LIR 1 เป็น 1 ข้อมูลจากดาต้าบัส 10110000 ถูกเก็บลงรีจิสเตอร์ IR 1

3) เมื่อป้อนสัญญาณนาฬิกาลูกที่ 3 สัญญาณที่จุด PCUP เป็น 0 เพื่อเพิ่มค่าใน PC ขึ้น 1 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.11 การทดลองแผงวงจรมัลติเพล็กซ์ 12 ช่อง

ทำหน้าที่เลือกสัญญาณอินพุต 12 ช่องออกสู่เอาต์พุตทีละ 1 ช่อง



รูปที่ 4.11 แผงวงจรมัลติเพล็กซ์ 12 ช่อง

4.11.1) การทดลอง

1) ป้อนข้อมูล 1100 เข้าที่ A1, B1, C1, D1 เป็นข้อมูล 8421 ชุดที่1

2) ป้อนสัญญาณลอจิก 1 เข้าจุด S1 เพื่อเลือกข้อมูลชุดที่ 1 ออกสู่ A, B, C, D OUT

4.11.2) ผลการทดลอง

1) ดูที่เอาต์พุต A, B, C, D มีค่าเป็น 1100

4.12 การทดลองแผงวงจรขับส่วนแสดงผลตัวเลข 7 ส่วน

4.12.1) การทดลอง

1) ป้อนลอจิก 1 เข้าที่จุด IS1

2) ทดลองป้อนลอจิก 1 เข้าที่จุด IS ต่างๆ

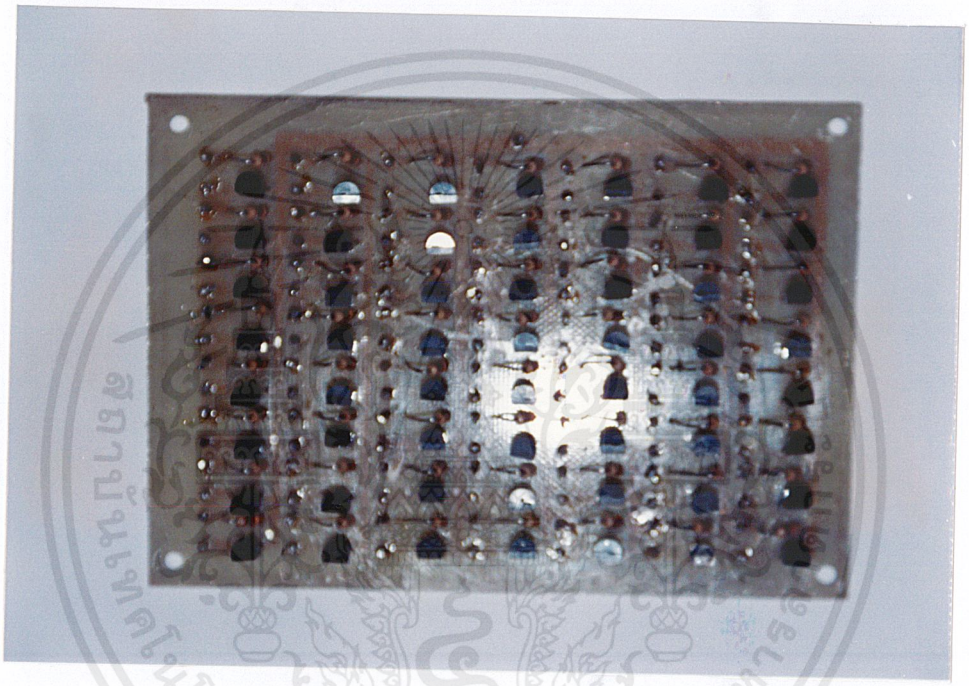
3) สังเกตผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12.2) ผลการทดลอง

1) สังเกตเอาต์พุต เป็นลอจิก 1

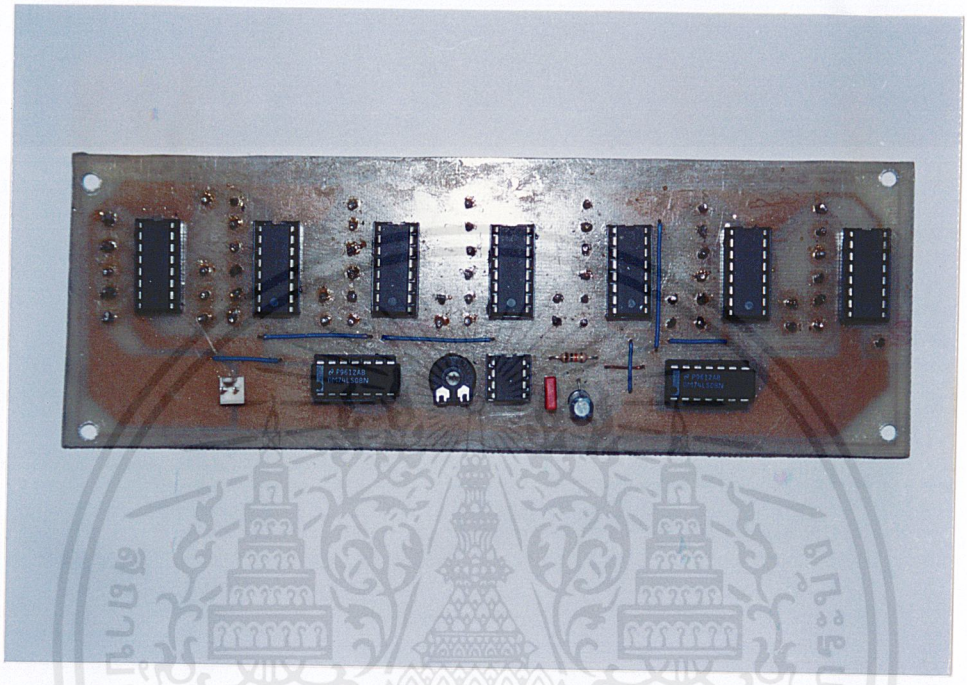
2) สังเกตเอาต์พุตต่างๆ ตามที่ป้อนอินพุตเข้าไป



รูปที่ 4.12 แผงวงจรขับส่วนแสดงผลตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.13 การทดลองแผงวงจรนับแบบวงแหวน 54 บิต



รูปที่ 4.13 แผงวงจรนับแบบวงแหวน 54 บิต

4.13.1) การทดลอง ต่อสัญญาณเข้าดังนี้

1) ป้อนแรงดัน + 5 โวลต์ เข้าวงจร

2) ปรับ VR 100 kΩ ไปทางขวา

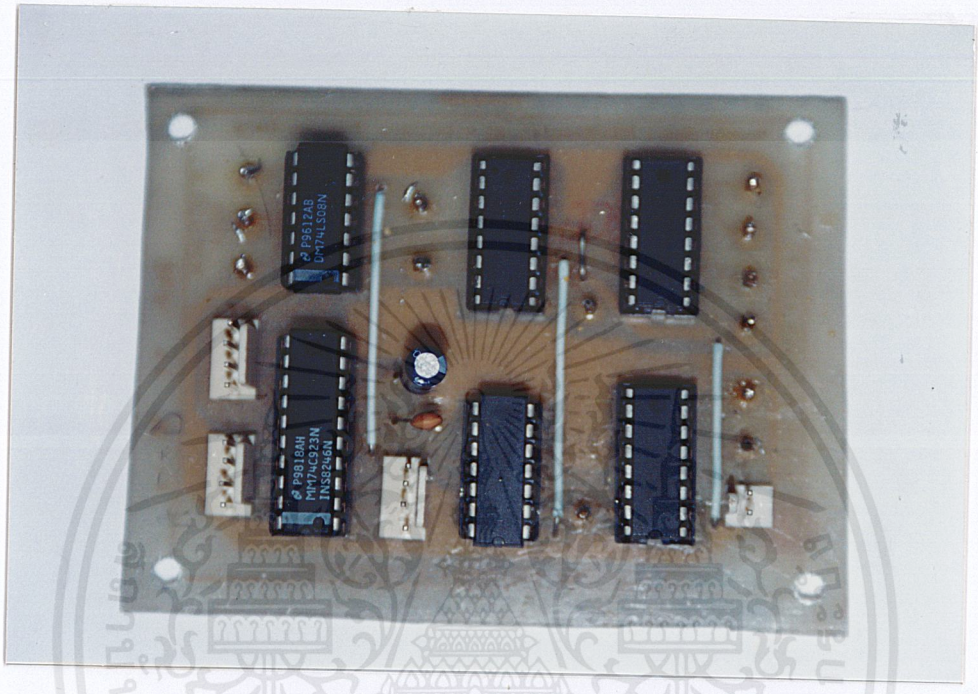
3) สังเกตผลการที่เอาต์พุต 1- 54

4.13.2) ผลการทดลอง

1) เอาต์พุต 1-54 จะเป็น 1 ครั้งละ 1 เอาต์พุต สลับกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.14 การทดลองแผงวงจรเข้ารหัสคีย์เป็นสัญญาณ BCD-8421



รูปที่ 4.14 แผงวงจรเข้ารหัสคีย์เป็นสัญญาณ BCD-8421

4.14.1) การทดลอง

1) ต่อปุ่มกด (Y1, Y2, Y3, Y4) ต่อกับคีย์เป็นแนวนอน

2) ต่อปุ่มกด (X1, X2, X3, X4) ต่อกับคีย์เป็นแนวตั้ง

3) กดปุ่ม 2

4) ทดลองกดปุ่ม ต่างๆ ครั้งละ 1 ปุ่ม

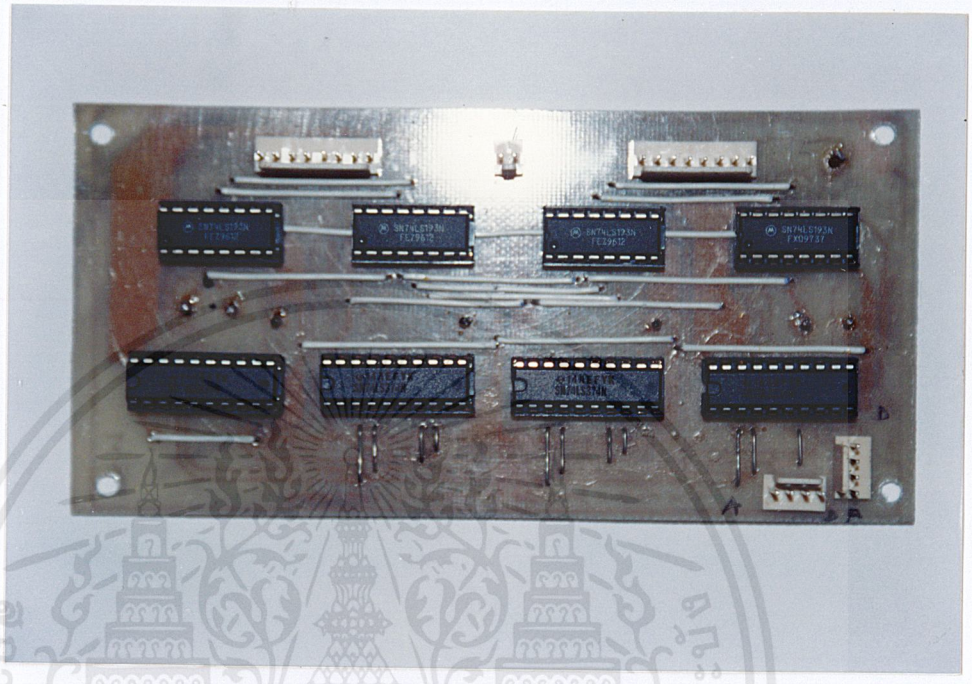
4.14.2) ผลการทดลอง

1) เอาต์พุต D, C, B, A จะออกเป็น BCD 8421 มีค่าเท่ากับ 0010

2) เอาต์พุต D, C, B, A เปลี่ยนแปลงตามปุ่มที่กด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.15 การทดลองแผงวงจรกำหนดค่าแอดเดรสของหน่วยความจำภายนอก



รูปที่ 4.15 การทดลองแผงวงจรกำหนดค่าแอดเดรสของหน่วยความจำภายนอก

4.15.1) การทดลอง

1) ป้อนสัญญาณ 1001 (A, B, C, D IN) รับสัญญาณ A, B, C, D 4 บิตจาก KEYEN

2) ป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA1

3) ป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA2

4) ป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA3

5) ป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA4

4.15.2) ผลการทดลอง

1) เมื่อป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA 1 สัญญาณ 1001 ถูกเก็บยัง A15-A12

2) เมื่อป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA 2 สัญญาณ 1001 ถูกเก็บยัง A11-A8

3) เมื่อป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA 3 สัญญาณ 1001 ถูกเก็บยัง A7-A4

4) เมื่อป้อนสัญญาณ 0 -> 1 เข้าที่จุด LA 4 สัญญาณ 1001 ถูกเก็บยัง A3-A0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

5.1 สรุป

โครงงานชุดสาธิตสถาปัตยกรรมภายใน CPU เป็นการนำเสนอการทำงานภายใน CPU แสดงถึงกระบวนการทำคำสั่งภายใน CPU ซึ่งโครงสร้างของชุดสาธิตสถาปัตยกรรมภายใน CPU มีการทำคำสั่งคล้ายกับ CPU เบอร์ Z80 ที่มีขนาดบัสข้อมูล (ดาต้าบัส) 8 บิต (D7-D0) และมีขนาดบัสตำแหน่ง (แอดเดรสบัส) 16 บิต (A15-A0) โดยมีรีจิสเตอร์ที่ใช้งานทั่วไป B, C, D, E, H, L ขนาด 8 บิต หรือใช้เป็นรีจิสเตอร์คู่ BC, DE, HL ขนาด 16 บิต ได้ มี อินเด็กซ์รีจิสเตอร์ IX, IY เป็นรีจิสเตอร์ขนาด 16 บิต ที่ใช้ใน อินเด็กซ์แอดเดรสซิ่งโหมดมีสแตกพอร์ตเตอร์ขนาด 16 บิตใช้เก็บตำแหน่งที่อยู่ของข้อมูลที่ถูกนำไปพักชั่วคราวมีโปรแกรมเคาท์เตอร์ขนาด 16 บิต ใช้เก็บตำแหน่งที่อยู่ของข้อมูลที่ CPU ทำการเฟ็ทช์ครั้งถัดไป มี ALU ขนาด 8 บิตทำหน้าที่ประมวลผลทางคณิตศาสตร์บวกลบ และกระทำทางลอจิกAnd, Or, Xor และสามารถเลื่อนบิตข้อมูล และหมุนบิตข้อมูลได้ ชุดคำสั่งที่ใช้ใน CPU สามารถเพิ่มได้โดยเขียนรูปแบบการทำคำสั่งลงในอีพรอมที่ใช้เก็บไมโครโค้ดมี ส่วนแสดงผล 7 ส่วน ใช้แสดงผลค่าต่างๆ ภายใน CPU จำนวน 54 หลัก เพื่อแสดงค่าข้อมูลที่อยู่ภายในตัว CPU สามารถปรับสัญญาณนาฬิกาที่จ่ายให้ CPU เพื่อดูการทำงานภายในช้าหรือเร็วได้ เพื่อดูการเปลี่ยนแปลงค่าต่างๆ ขณะ CPU ทำคำสั่ง

เมื่อทำการต่อแผ่นวงจรพิมพ์ของแต่ละวงจรเข้าด้วยกันพบว่า ยังไม่สามารถทำงานได้ทั้งหมด กล่าวคือสามารถทำการทดสอบโดยการใช้สวิตซ์ป้อนลอจิกทดสอบ แทนการทำงานของอีพรอม เมื่อทำการป้อนลอจิกด้วยสวิตซ์ ชุดสาธิตสถาปัตยกรรมภายใน CPU สามารถโอนย้ายข้อมูลระหว่างรีจิสเตอร์แต่ละตัวได้ และสามารถทำการคำนวณทางคณิตศาสตร์ และกระทำทางลอจิกได้ แต่ยังไม่สามารถเฟ็ทช์ข้อมูลจาก หน่วยความจำภายนอกมาเก็บภายใน CPU ได้ และวงจรรับการกดสวิตซ์ป้อนโปรแกรมยังทำงานได้ไม่สมบูรณ์

5.2 ปัญหา และแนวทางแก้ไข

1) ปัญหาการสร้างแผ่นวงจรพิมพ์ที่ใช้ในชุดสาธิตสถาปัตยกรรมภายใน CPU เนื่องจากสาธิตสถาปัตยกรรมภายใน CPU ประกอบด้วยไอซีลอจิกเกต TTL และไอซีอื่นๆ จำนวนประมาณ 160 ตัว จึงได้แบ่งแผ่นวงจรพิมพ์ออกเป็นภาคๆ ใช้แผ่นวงจรพิมพ์ชนิดหน้าเดียว ทำให้ง่ายต่อการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบ แต่พบปัญหาว่าเมื่อแผ่นวงจรพิมพ์มีจำนวนมาก ทำให้การต่อสายเชื่อมวงจรมีจำนวนมาก ส่งผลให้เกิดความยุ่งของสายที่ใช้เชื่อมระหว่างแผ่นวงจรพิมพ์แต่ละแผ่นอีกทั้งยังส่งผลต่อการทำงานของวงจรอีกด้วย

แนวทางการแก้ไข ควรออกแบบแผ่นวงจรพิมพ์ให้เป็นแบบสองหน้า และนำอุปกรณ์ที่อยู่ใกล้กันมารวมกันให้ได้มากที่สุด แล้วออกแบบแผ่นวงจรพิมพ์ให้เป็นแบบสองหน้า ก็จะสามารถแก้ปัญหาดังกล่าวได้

2) ปัญหาทางด้านการแสดงผล เนื่องจากชุดสาธิตสถาปัตยกรรมภายใน CPU มีการใช้ส่วนแสดงผล 7 ส่วน เป็นจำนวน 54 หลัก เพื่อการแสดงผลค่าข้อมูลต่างๆ ภายใน CPU เนื่องจากในครั้งแรกไม่สามารถหาไอซีที่สามารถแปลงรหัส BCD 8421 เป็นตัวเลข 0 – F ทำให้ระบบการแสดงผลไปใช้การสแกนแทน ซึ่งการแสดงผลแบบแลตซ์ จะให้ความสว่างได้ดีกว่าแบบสแกน

แนวทางการแก้ไข ควรเปลี่ยนการแสดงผลเป็นแบบแลตซ์โดยใช้ไอซีเบอร์ MC14495 หรือถ้าใช้แบบ สแกนก็เพิ่มแรงดันในการขับ ส่วนแสดงผล 7 ส่วน เพิ่มจาก 5 โวลต์ เป็นประมาณ 12 โวลต์ หรือ เปลี่ยน ส่วนแสดงผล 7 ส่วน เป็นชนิดคอมม่อนอาโนด และเพิ่มแรงดันไฟบวกร่วมขึ้นไปให้มากกว่า 5 โวลต์ ก็ทำให้ ส่วนแสดงผล 7 ส่วน สว่าง และเห็นได้อย่างชัดเจน

3) ปัญหาการเชื่อมต่อระหว่างแผ่นวงจรพิมพ์ โครงงานนี้ใช้คอนเน็คเตอร์สีขาวเมื่อใช้งานจริงเป็นจำนวนมาก เกิดปัญหาการเชื่อมต่อไม่แน่นทำให้สัญญาณในแต่ละไบต์อาจไม่ครบ

แนวทางการแก้ไข เมื่อพบบิตใดแสดงผลไม่ถูกต้องให้ทำการขยับคอนเน็คเตอร์เล็กน้อย ถ้าแสดงผลถูกต้องควรล็อกสายไว้ หรือถ้ายังไม่ดีขึ้นให้ถอดสายนั้นออก และใส่ไส้ในใหม่ หรือเปลี่ยนคอนเน็คเตอร์เป็นชนิดอื่นแทน ก็สามารถแก้ไขปัญหาดังกล่าวได้

4) ปัญหาแหล่งจ่ายพลังงาน โครงงานนี้การทดลองครั้งแรกได้ใช้แหล่งจ่ายพลังงานแบบลิเนียร์เร็กกูเลเตอร์พบว่าเกิดความร้อนมากเนื่องจากเกิดแรงดันตกคร่อมในตัวไอซีเร็กกูเลเตอร์สูง ทำให้แรงดันตกคร่อมที่ตัวไอซีเรกูเลเตอร์เปลี่ยนรูปแรงดันตกคร่อมเป็นพลังงานความร้อนขึ้นซึ่งมีความร้อนค่อนข้างสูง

แนวทางการแก้ไข เปลี่ยนระบบจ่ายพลังงานไฟฟ้าจากลิเนียร์เร็กกูเลเตอร์เป็นแบบสวิตชิ่งที่ใช้กับเครื่องคอมพิวเตอร์ก็สามารถแก้ไขปัญหาดังกล่าวได้เนื่องจากสวิตชิ่งสามารถจ่ายกระแสได้ถึง 23 แอมป์ ซึ่งมีพลังงานจ่ายให้กับโครงงานอย่างเพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนา

1) เพิ่มคำสั่งในชุดสาธิตสถาปัตยกรรมภายใน CPU โดยการสร้างวงจรด้วยเกตพื้นฐาน และเขียนรหัสคำสั่งใหม่ในอีพรอมที่เก็บรหัสคำสั่ง หรือไมโครโค้ดก็จะได้คำสั่งเพิ่มขึ้น

2) เพื่อความสวยงาม และเป็นระเบียบของชุดสาธิตสถาปัตยกรรมภายใน CPU ควรมีการจัดทางเดินของสายเป็นหมวดหมู่ของทางเดินสัญญาณ

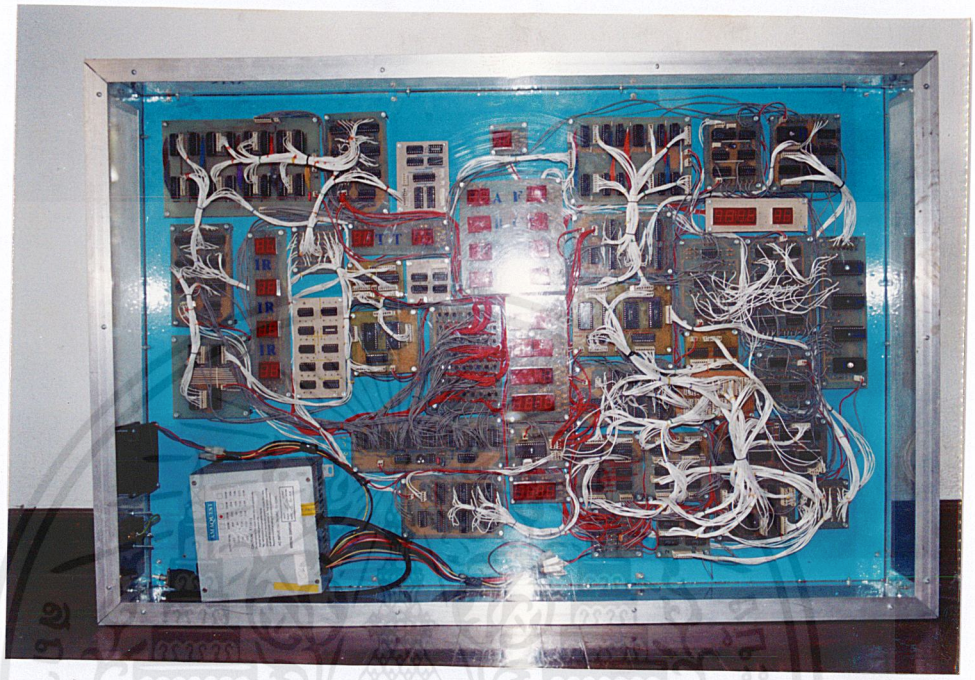
3) ในส่วนการทำงานแต่ละภาคควรมีสัญญาณไฟ โดยแสดงว่าภาคไหนทำงานอยู่ในขณะทำคำสั่งนั้น

4) เพื่อให้เกิดประโยชน์สูงสุดของผู้เรียนควรมีหนังสือไมโครโปรเซสเซอร์ และคู่มือการใช้งานที่สมบูรณ์แบบประกอบการใช้ด้วย

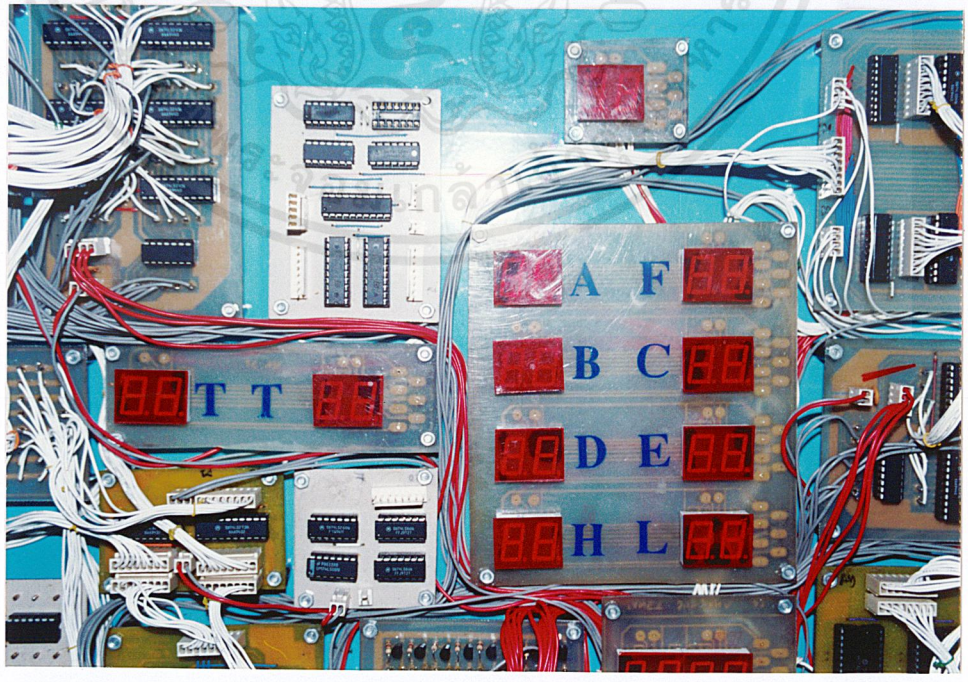
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 เครื่องต้นแบบ

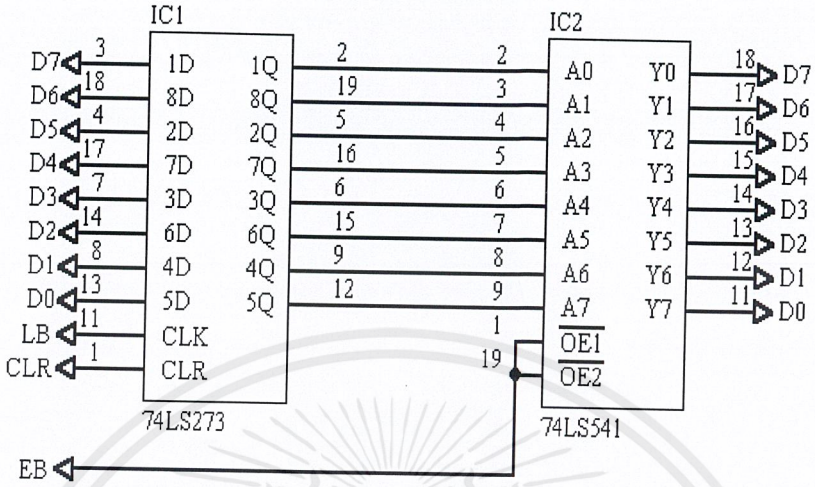


รูปที่ ก.2 แผงวงจรรีจิสเตอร์ทั่วไป

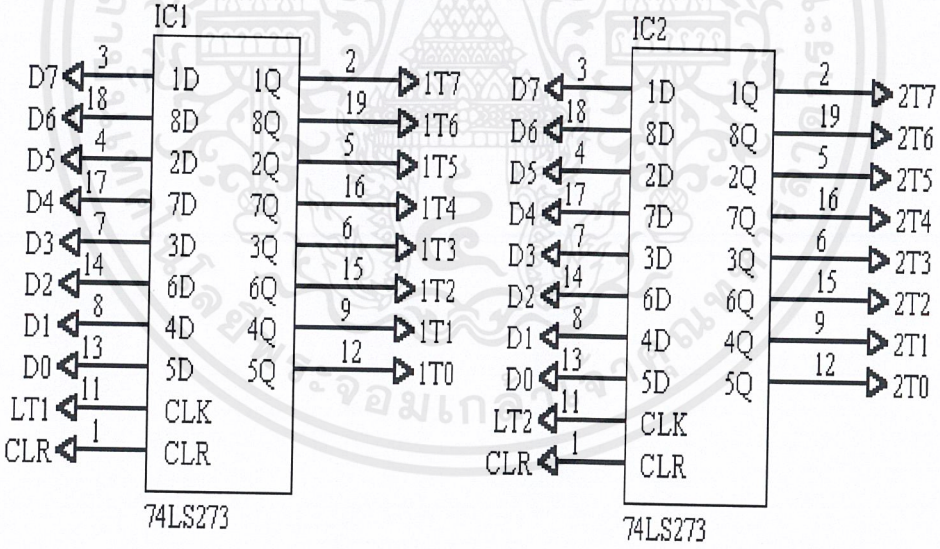
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
วงจร และแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

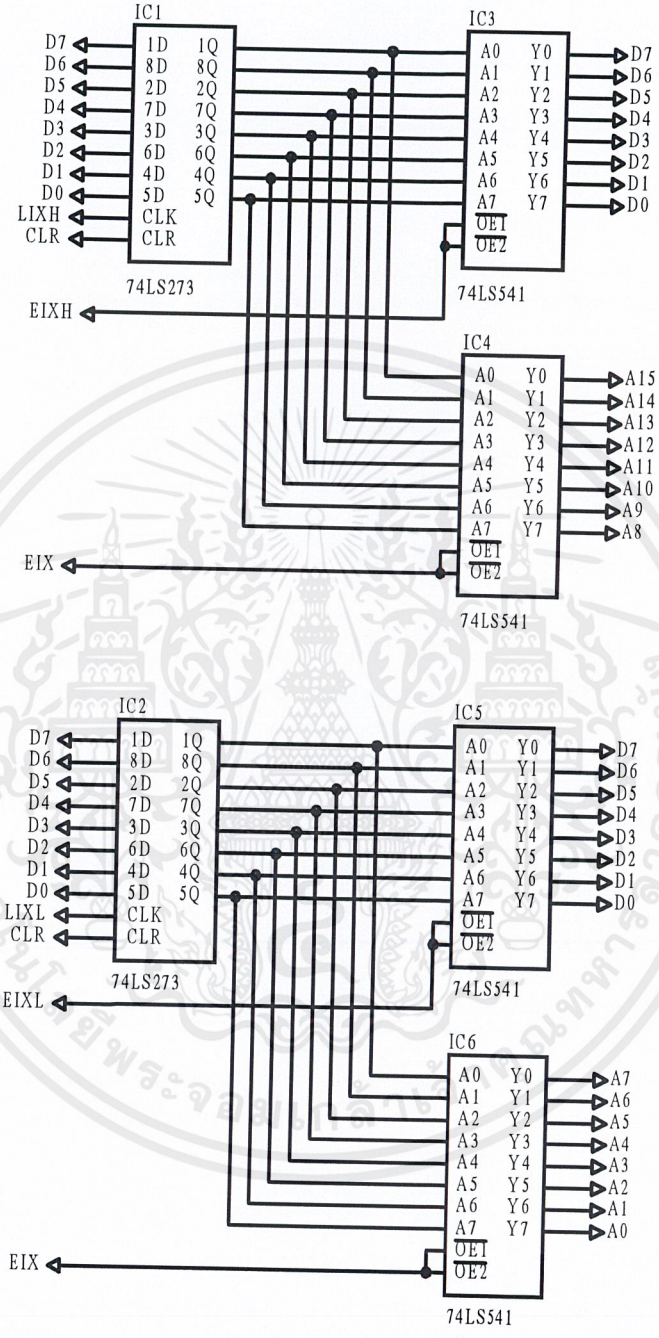


รูปที่ ข.1 วงจรรีจิสเตอร์เก็บค่าทั่วไป 8 บิต



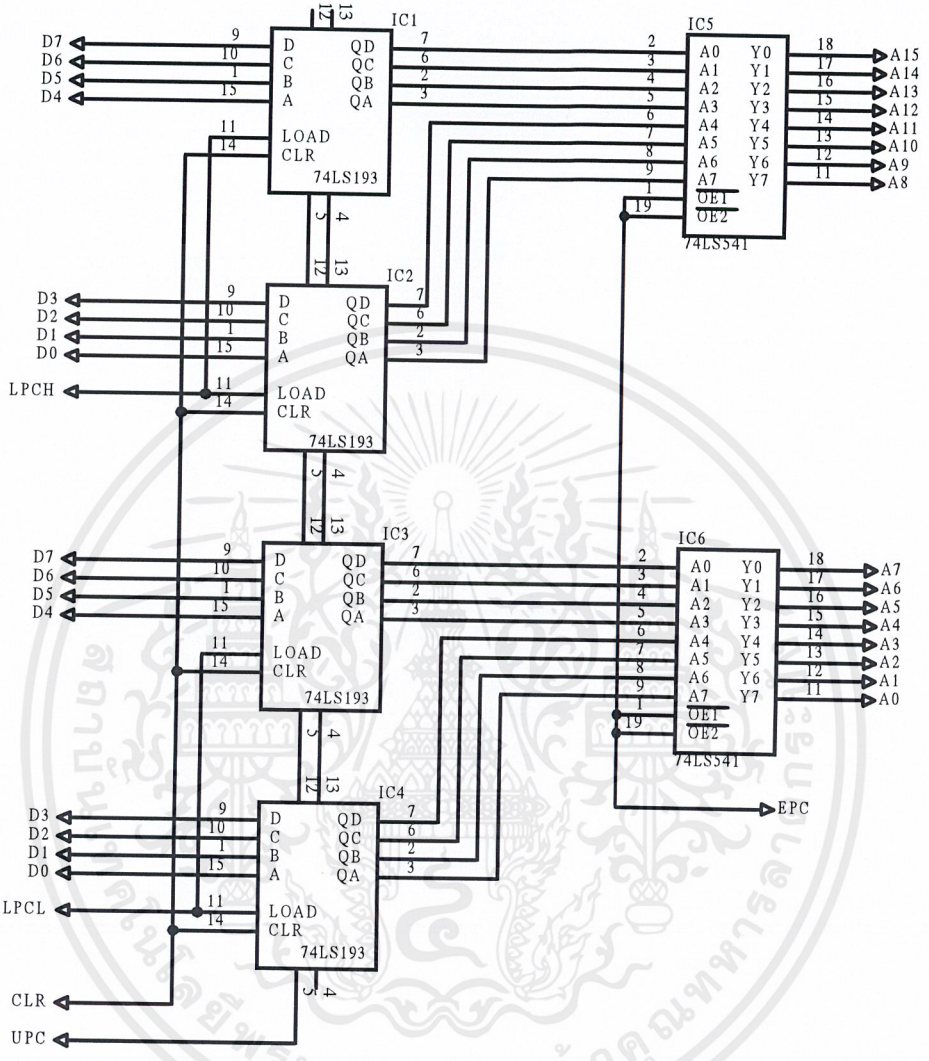
รูปที่ ข.2 วงจรรีจิสเตอร์พักข้อมูล (Temp1 , Temp2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



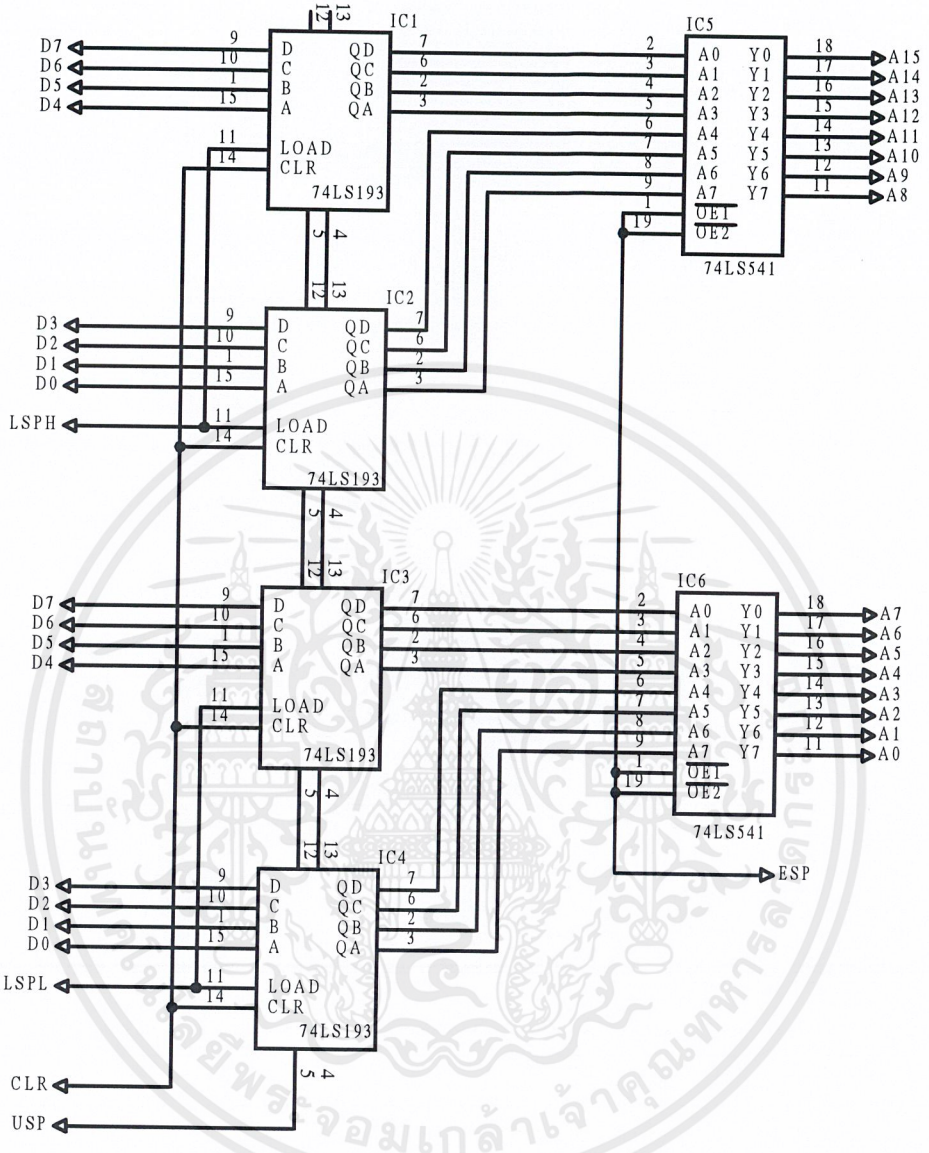
รูปที่ ข.3 รีจิสเตอร์ IX, IY 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



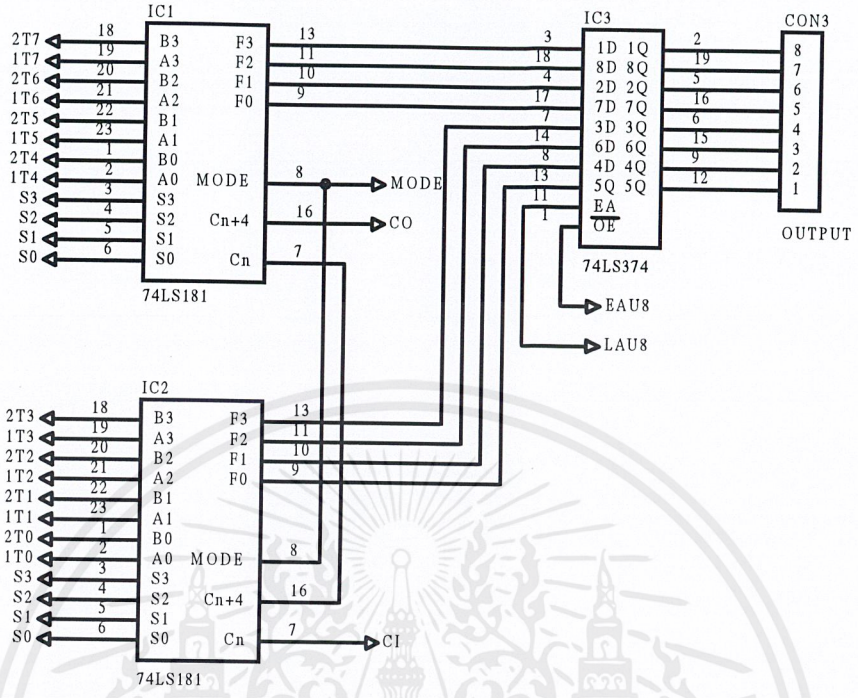
รูปที่ ข.4 โปรแกรมเคาน์เตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



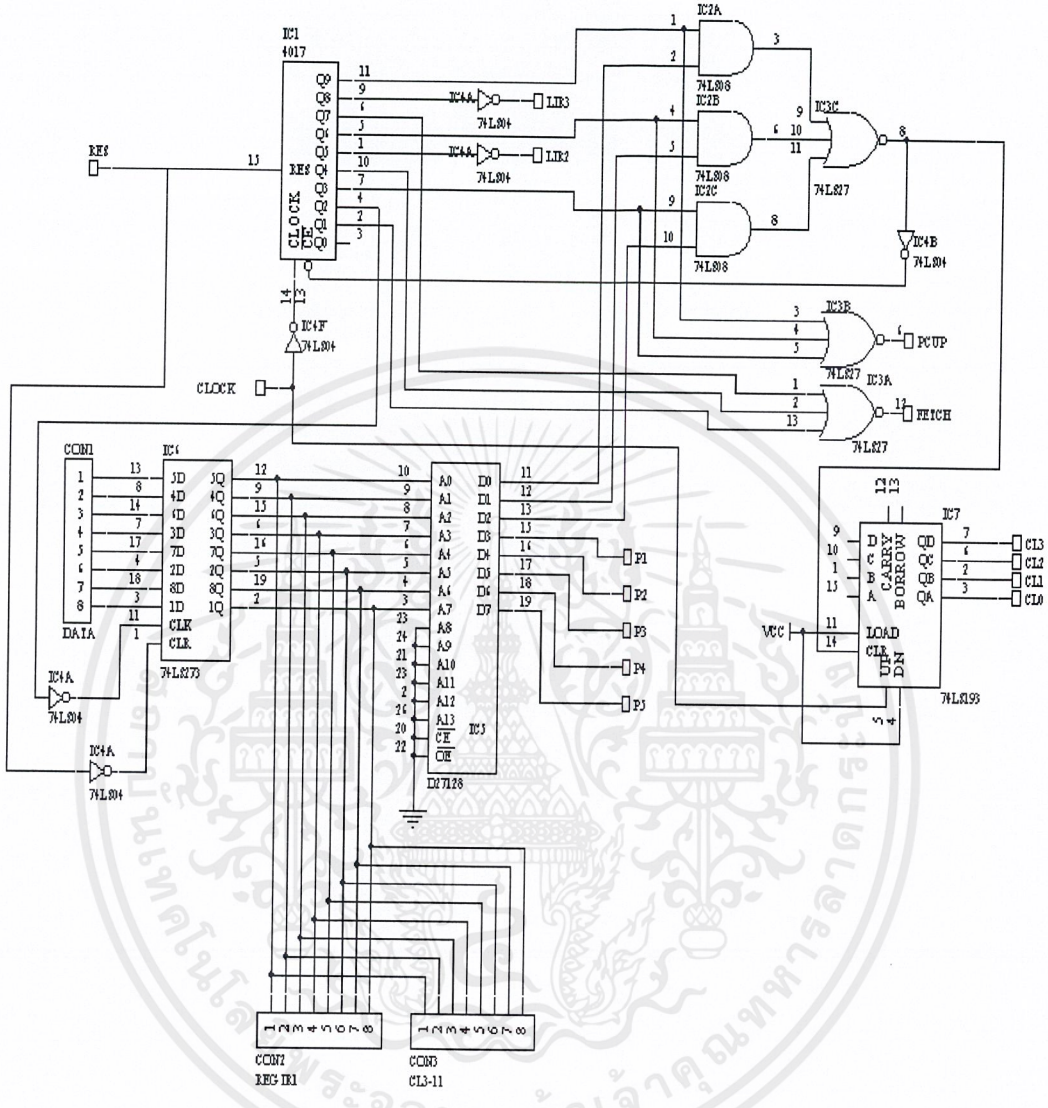
รูปที่ ข.5 สแตกพอยน์เตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

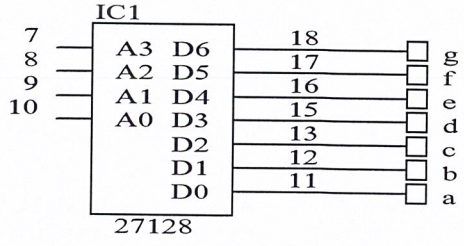


รูปที่ ข.6 วงจรคำนวณทางคณิตศาสตร์ และกระทำทางลอจิก 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

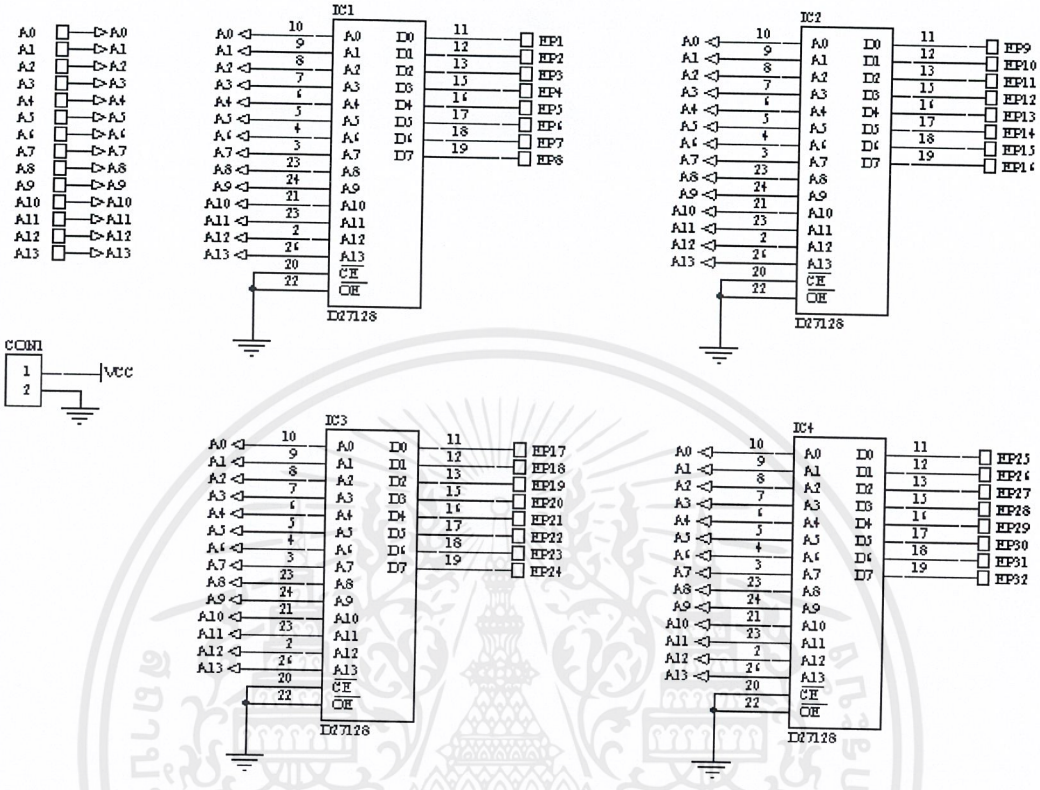


รูปที่ ข.7 วงจรเฟิทซ์ และแปลรหัสคำสั่ง

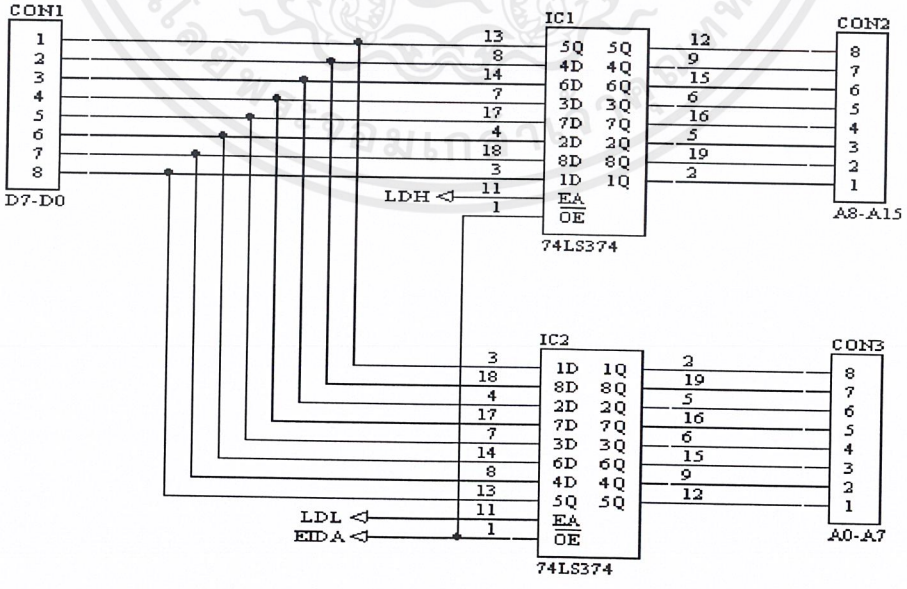


รูปที่ ข.8 วงจรแปลงสัญญาณ BCD 8421ขับส่วนแสดงผล 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

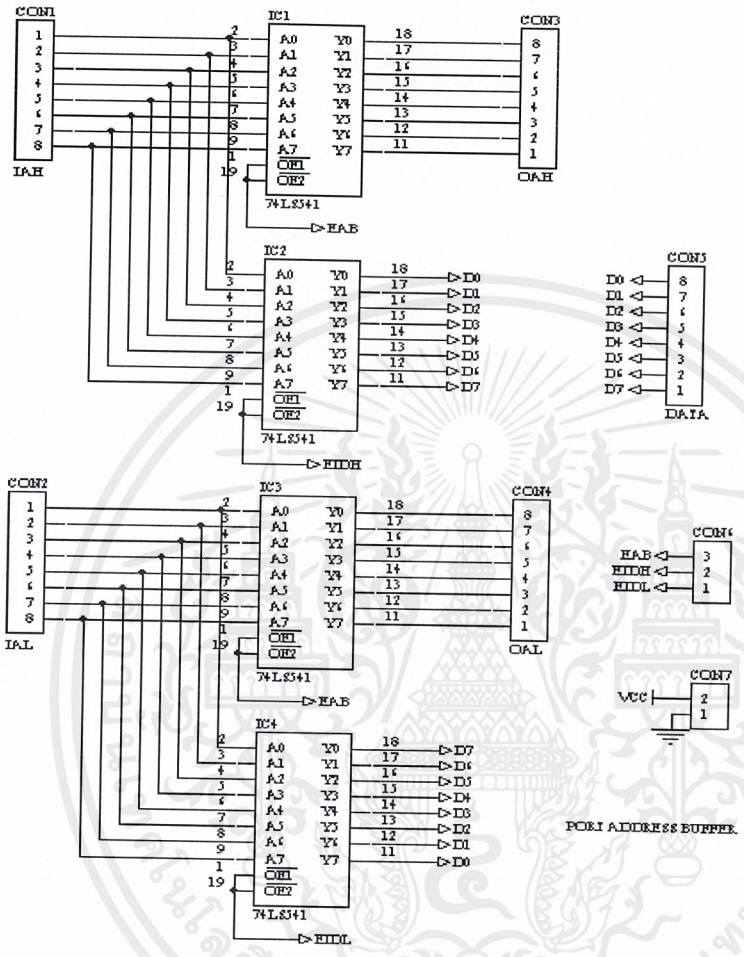


รูปที่ ข.9 วงจรเก็บรหัสไมโครโค้ด



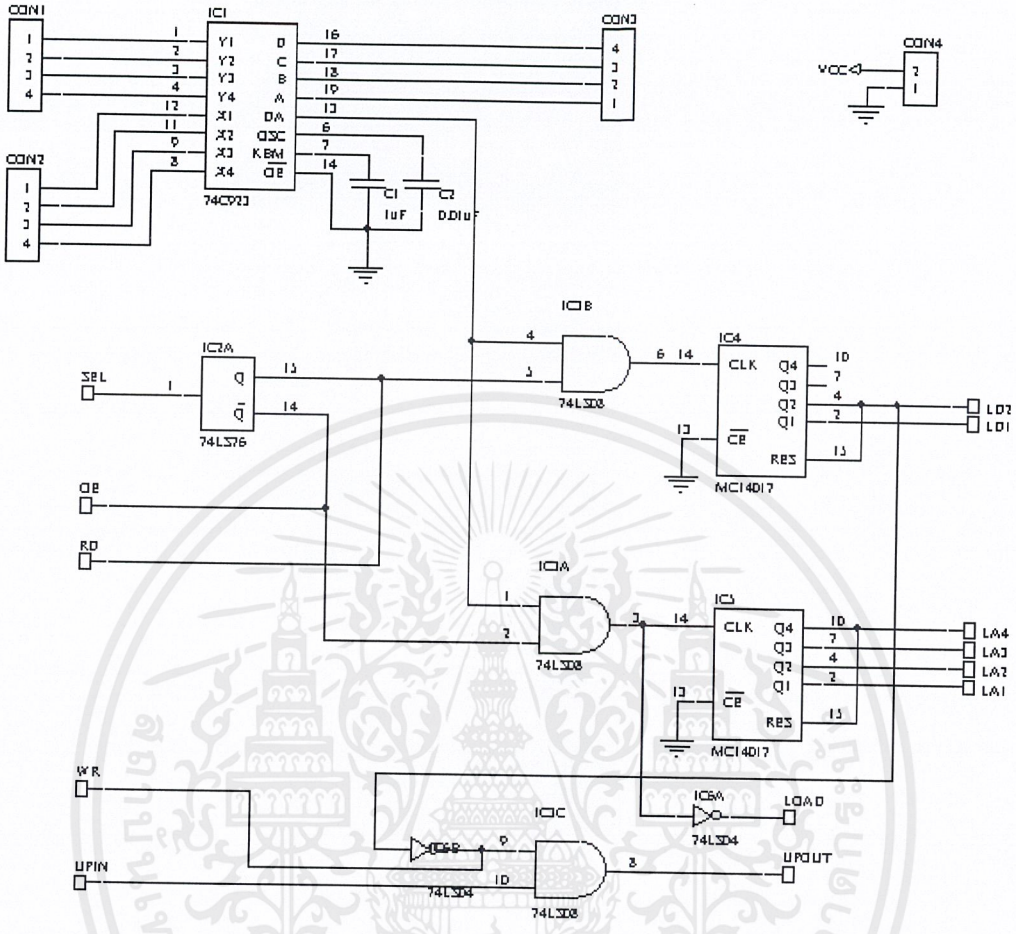
รูปที่ ข.10 วงจรแปลงกลุ่มสัญญาณข้อมูลเป็นกลุ่มสัญญาณตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



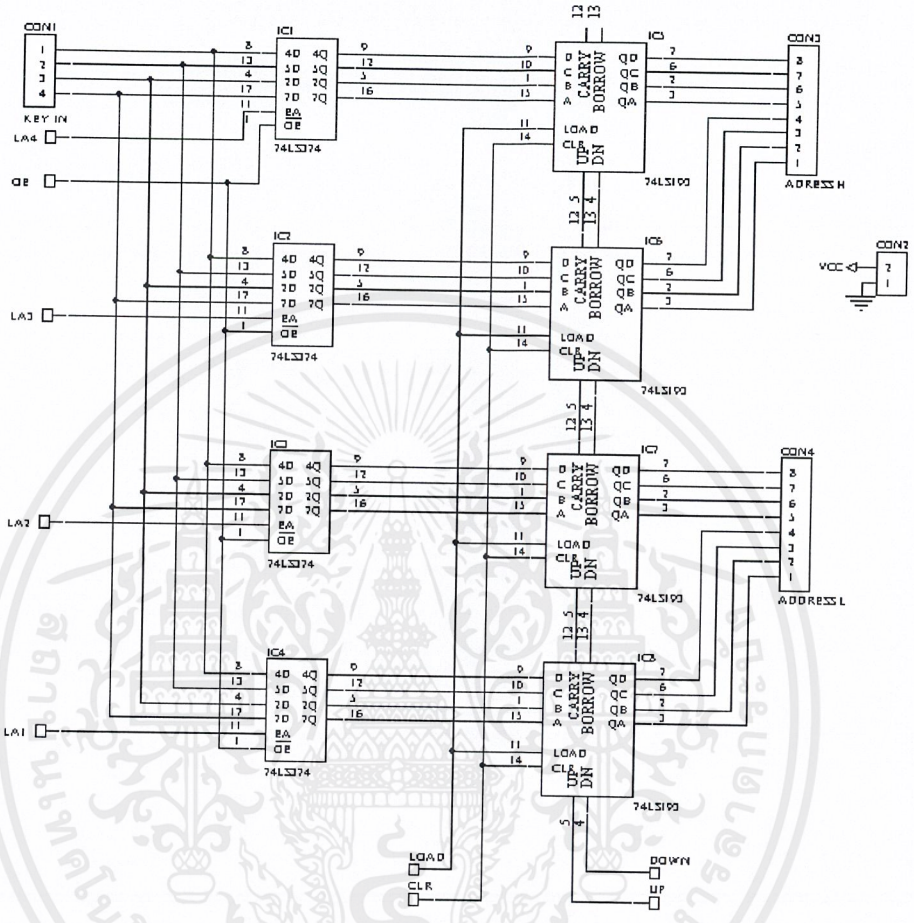
รูปที่ ข.11 วงจรแปลงสัญญาณกลุ่มตำแหน่งเป็นกลุ่มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



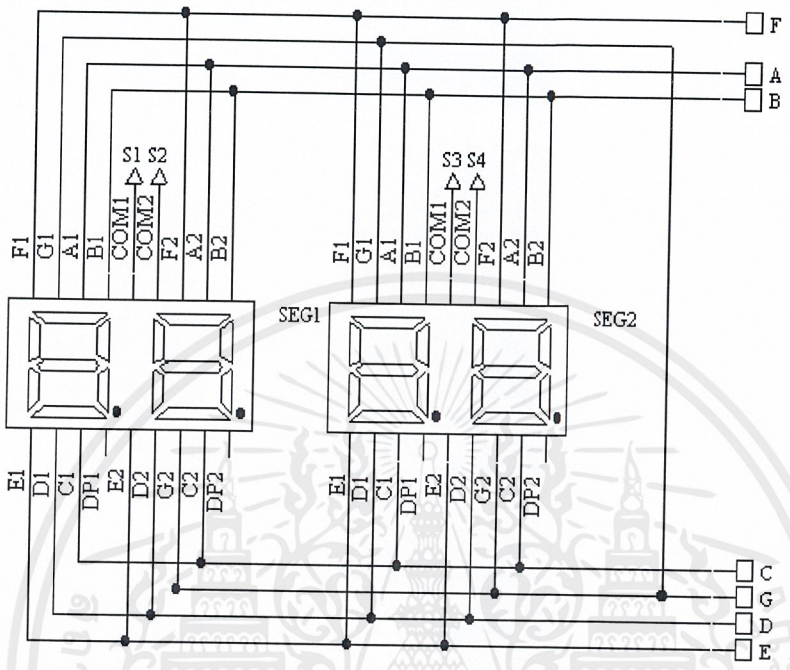
รูปที่ ข.12 วงจรเข้ารหัสคีย์เมตริกซ์เป็นรหัส BCD-8421

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



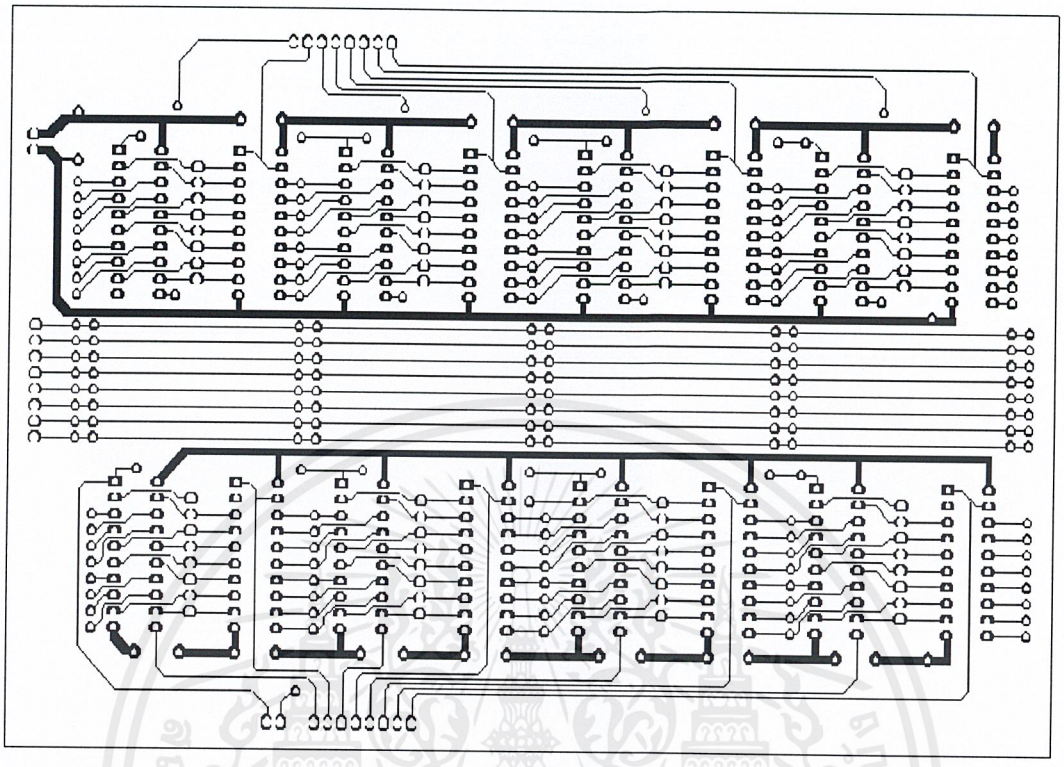
รูปที่ ข.13 วงจรกำหนดตำแหน่งที่อยู่ข้อมูลเมื่อป้อนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

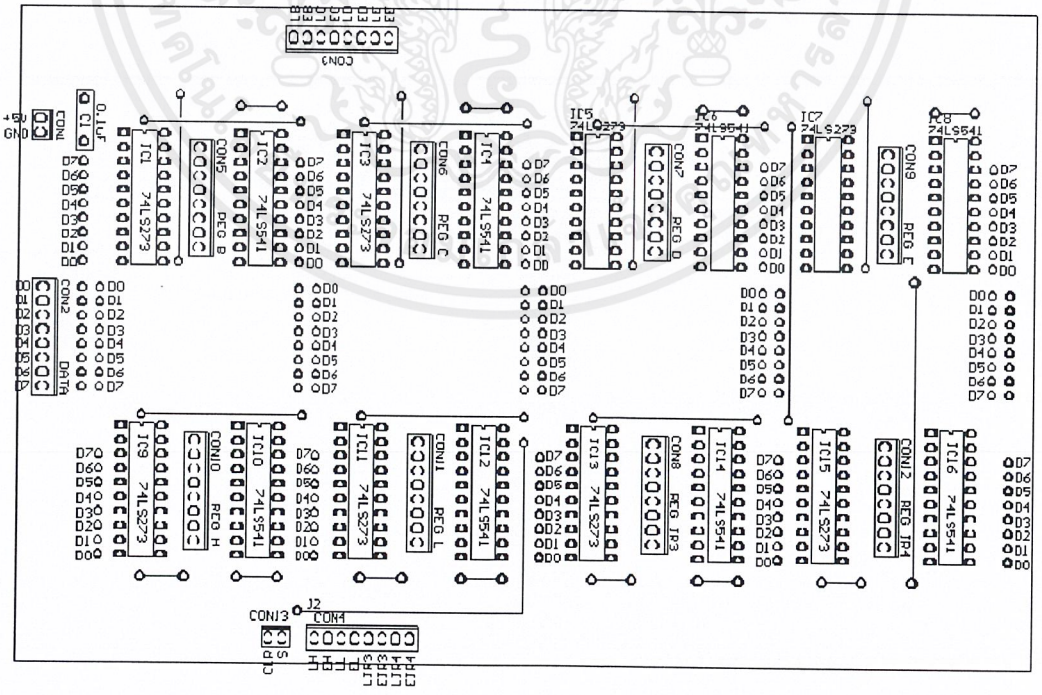


รูปที่ ข.14 วงจรตัวเลขแสดงผล 7 ส่วน 4 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

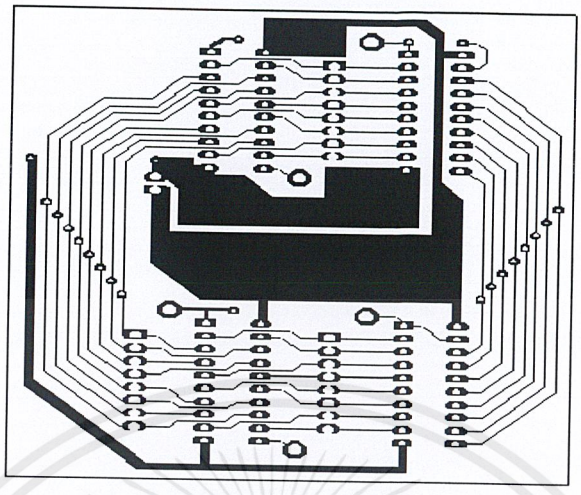


รูปที่ ข.15 ลายวงจรพิมพ์รีจิสเตอร์ทั่วไป 8 บิต

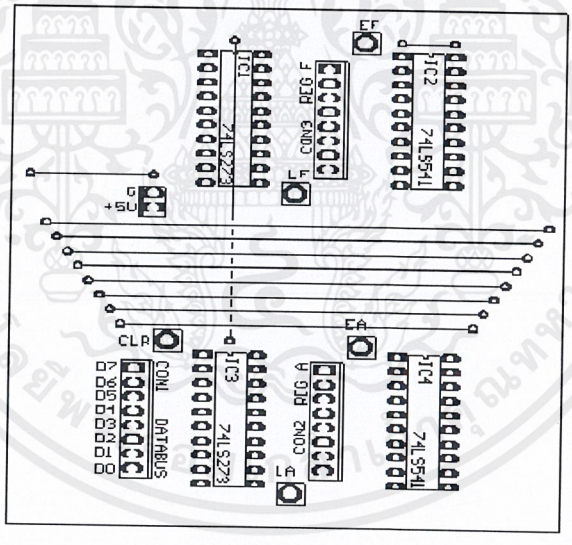


รูปที่ ข.16 การลงอุปกรณ์ภาครีจิสเตอร์ทั่วไป 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

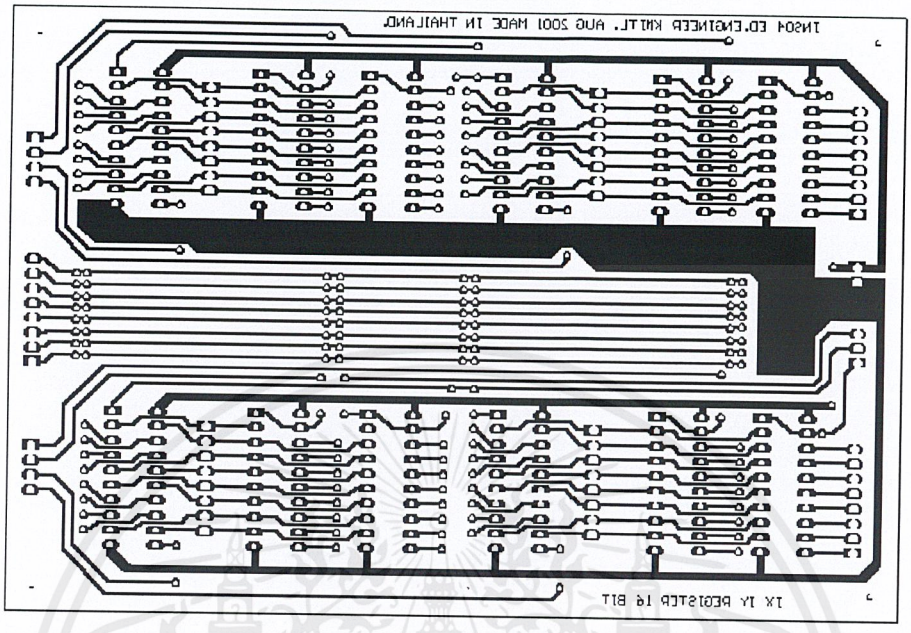


รูปที่ ข.17 ลายวงจรพิมพ์ของภาครีจิสเตอร์ A, F

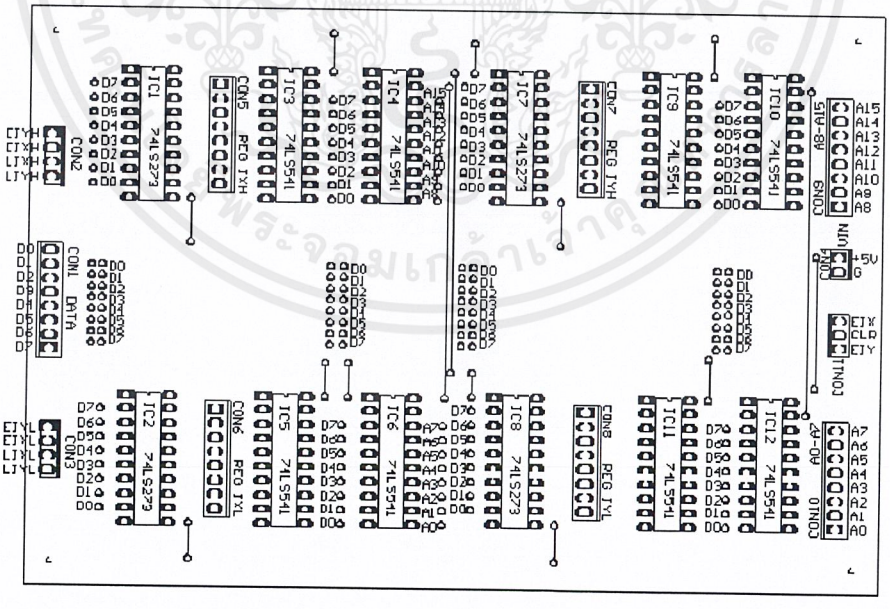


รูปที่ ข.18 การลงอุปกรณ์ภาครีจิสเตอร์ A, F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

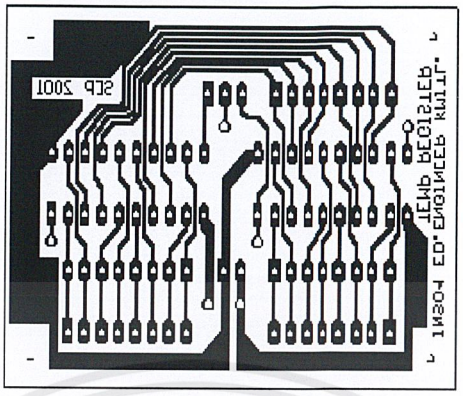


รูปที่ ข.19 ลายวงจรพิมพ์ของภาครีจิสเตอร์ IX, IY

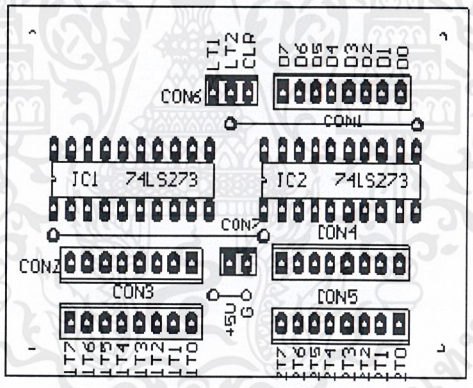


รูปที่ ข.20 การลงอุปกรณ์ของภาครีจิสเตอร์ IX, IY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

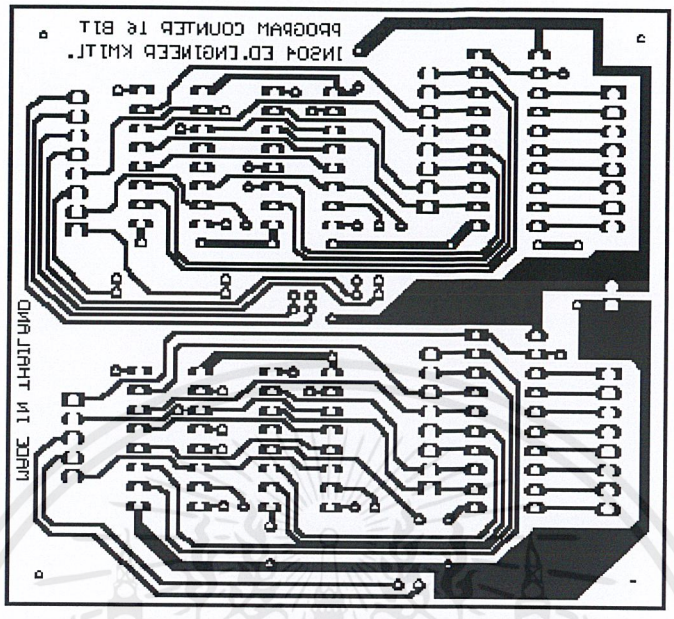


รูปที่ ข.21 ลายวงจรพิมพ์ของภาค รีจิสเตอร์พักข้อมูล 8 บิต

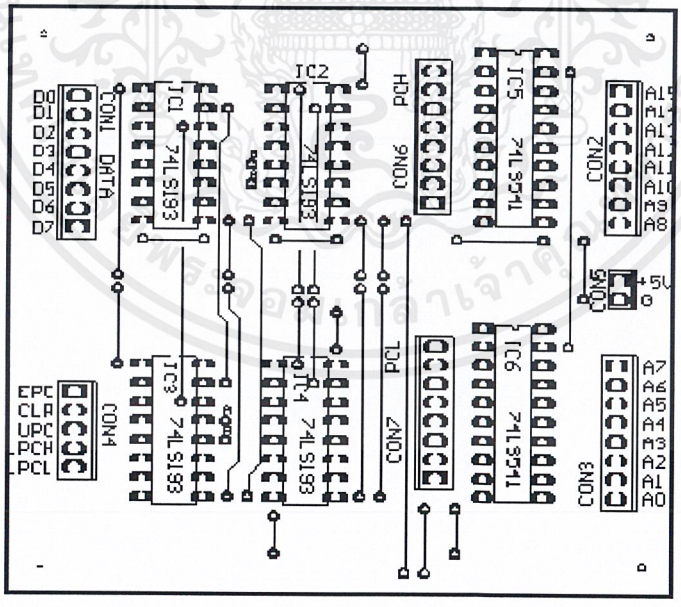


รูปที่ ข.22 การลงอุปกรณ์ของภาค รีจิสเตอร์พักข้อมูล 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

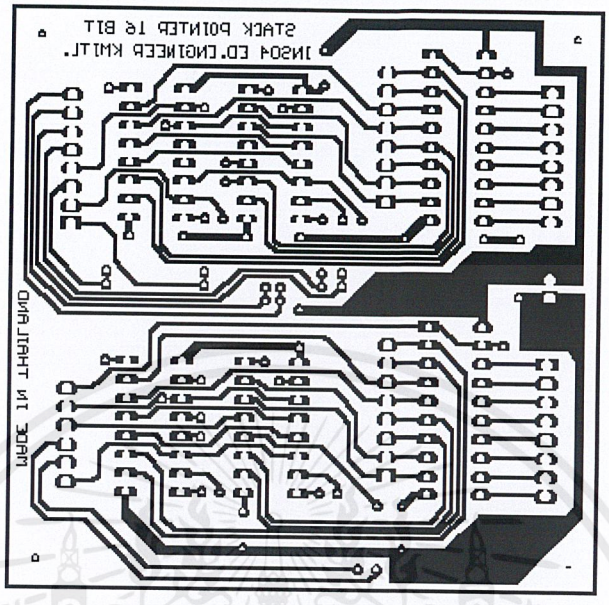


รูปที่ ข.23 ลายวงจรพิมพ์ของภาคโปรแกรมเคาน์เตอร์ 16 บิต

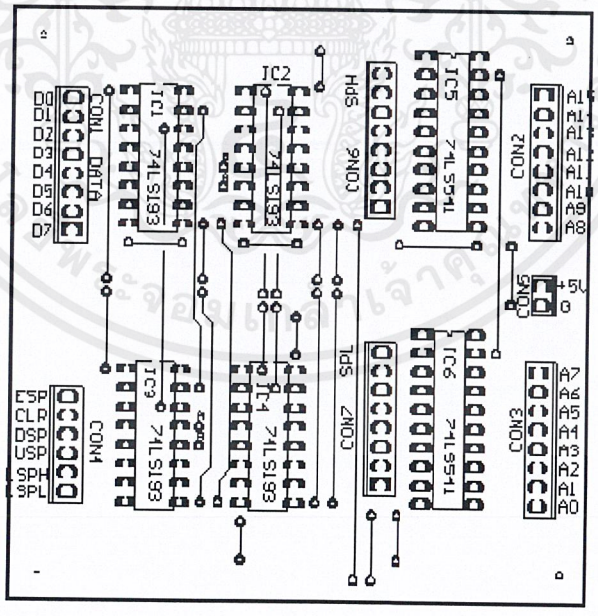


รูปที่ ข.24 การลงอุปกรณ์ของภาคโปรแกรมเคาน์เตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

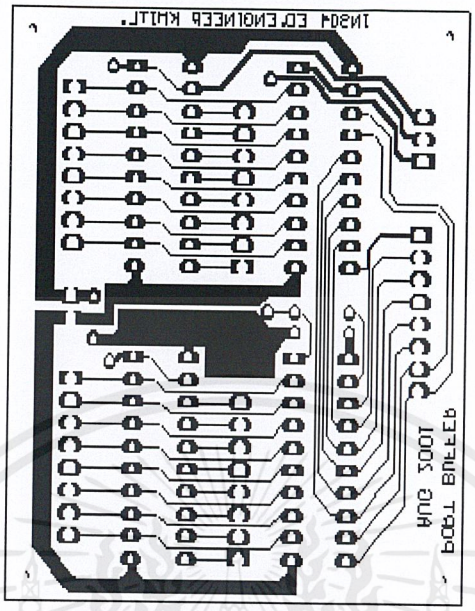


รูปที่ ข.25 ลายวงจรพิมพ์ของภาคสแตกพอยน์เตอร์ 16 บิต

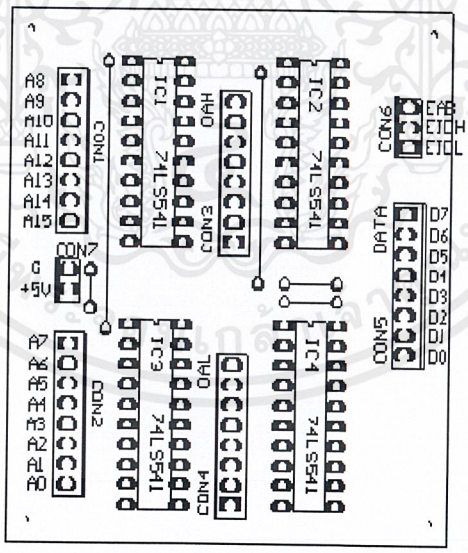


รูปที่ ข.26 การลงอุปกรณ์ของภาคสแตกพอยน์เตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

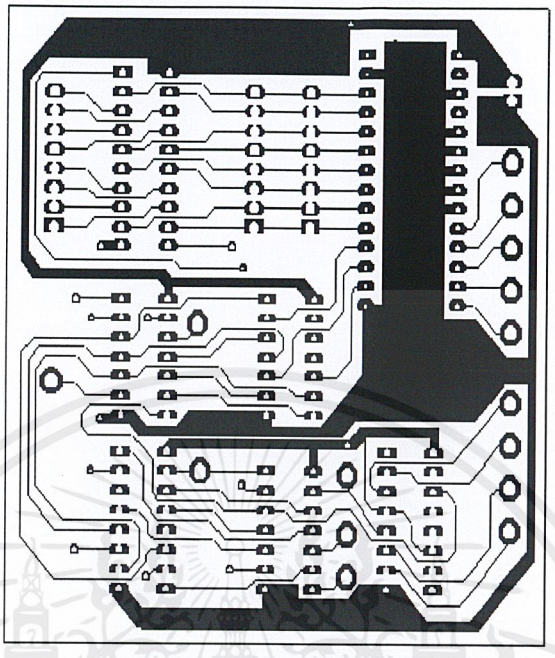


รูปที่ ข.27 ลายวงจรพิมพ์ของภาคพอร์ต

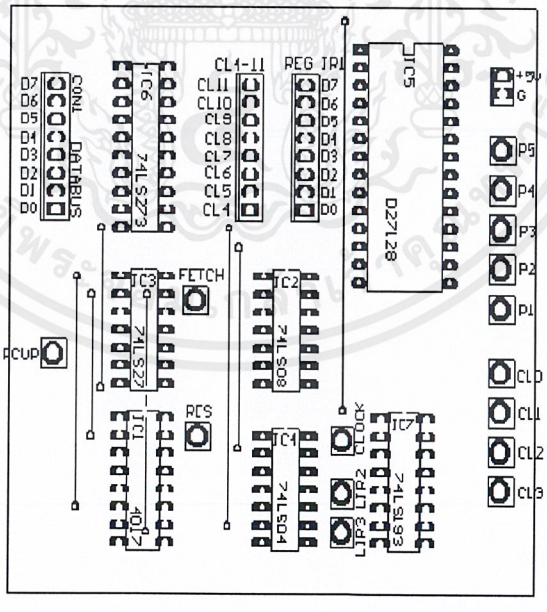


รูปที่ ข.28 การลงอุปกรณ์ของภาคพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

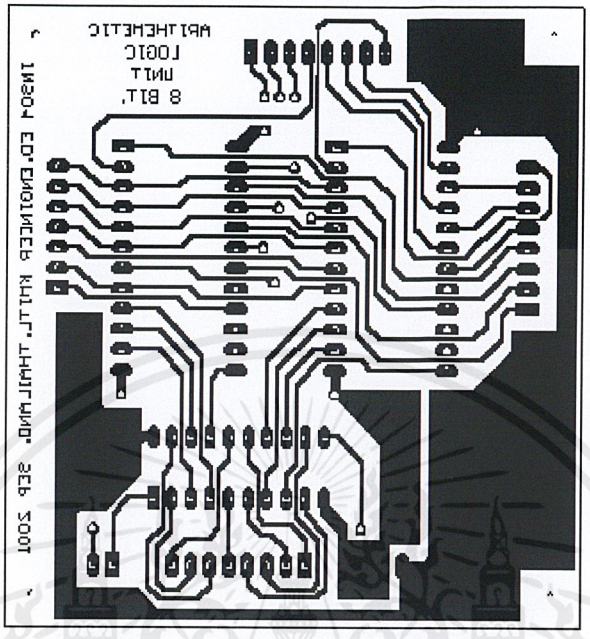


รูปที่ ข.29 ลายวงจรพิมพ์ของภาคเฟ็ทซ์ และถอดรหัสคำสั่ง

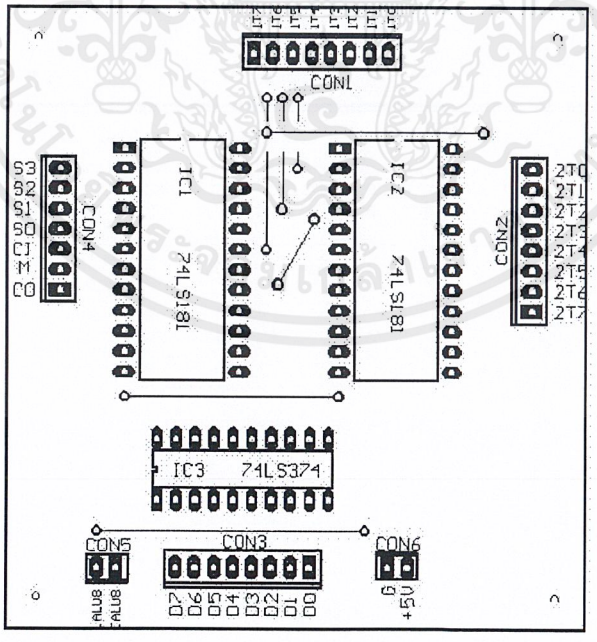


รูปที่ ข.30 การลงอุปกรณ์ของภาคเฟ็ทซ์ และถอดรหัสคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

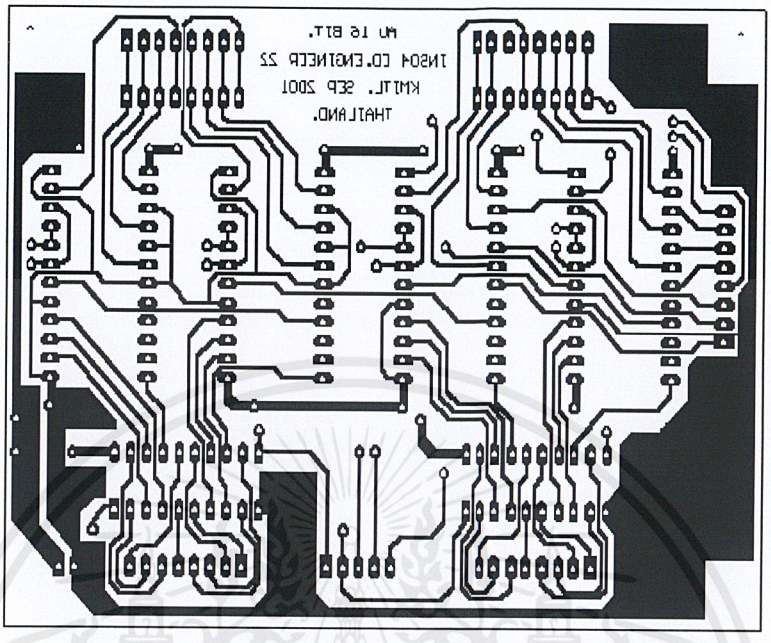


รูปที่ ข.31 ลายวงจรพิมพ์ของภาค ALU 8 บิต

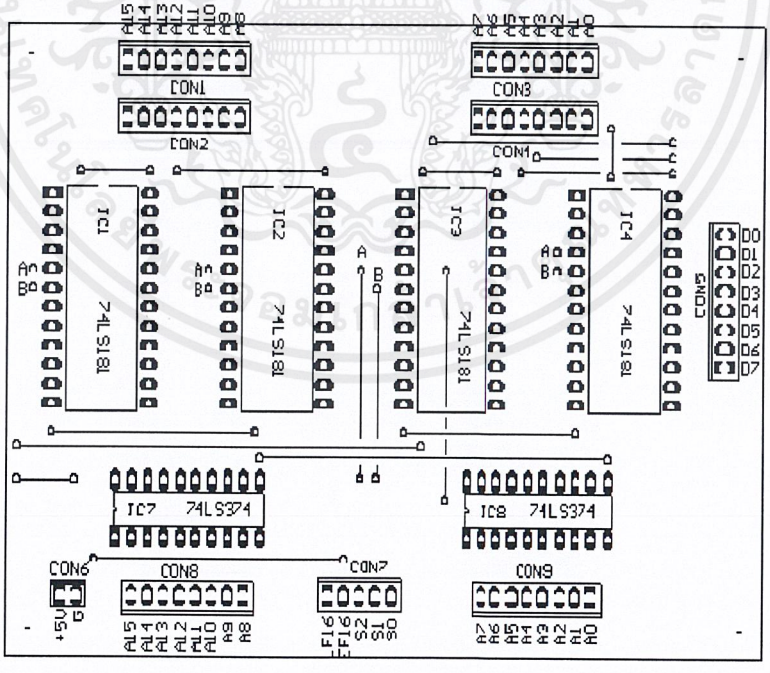


รูปที่ ข.32 การลงอุปกรณ์ของภาค ALU 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

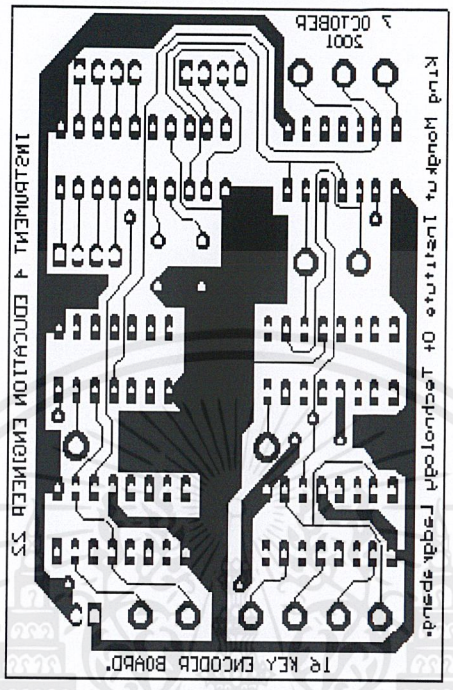


รูปที่ ข.33 ลายวงจรพิมพ์ของภาคบวกเลข 16 บิต

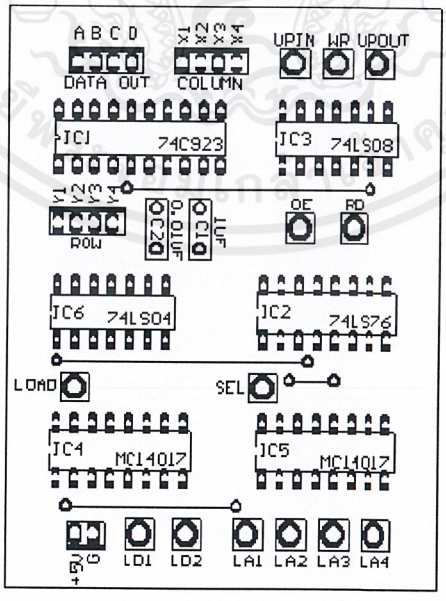


รูปที่ ข.34 การลงอุปกรณ์ของภาคบวกเลข 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

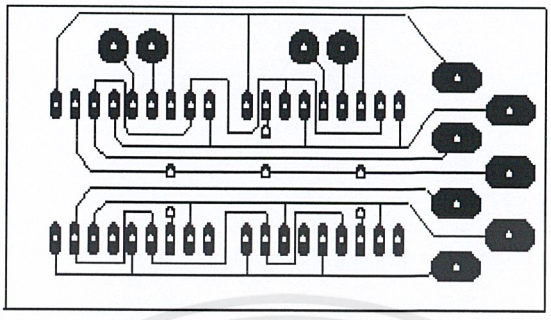


รูปที่ ข.35 ลายวงจรพิมพ์ของภาคถอดรหัสคีย์เมตริกซ์

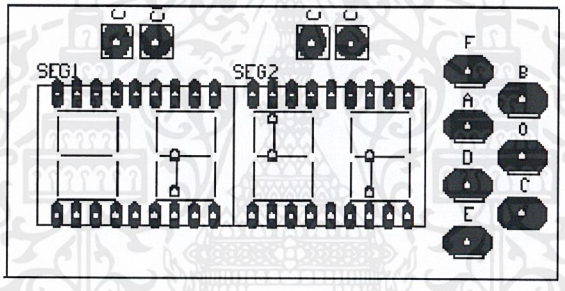


รูปที่ ข.36 การลงอุปกรณ์ของภาคถอดรหัสคีย์เมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.37 ลายวงจรพิมพ์ของภาคส่วนแสดงผล 7 ส่วน



รูปที่ ข.38 การลงอุปกรณ์ของภาคส่วนแสดงผล 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์

1.ภาครีจิสเตอร์ทั่วไป 8 บิต

รายการอุปกรณ์	จำนวน
1. 74LS273	8 ตัว
2. 74LS541	8 ตัว
3. C 0.1μF	1 ตัว
4. คอนเน็คเตอร์ 8 ขา	11 ตัว
5. คอนเน็คเตอร์ 2 ขา	ตัว

2.ภาครีจิสเตอร์พักข้อมูล

รายการอุปกรณ์	จำนวน
1. 74LS273	2 ตัว
2. คอนเน็คเตอร์ 8 ขา	5 ตัว
3. คอนเน็คเตอร์ 3 ขา	1 ตัว
4. คอนเน็คเตอร์ 2 ขา	1 ตัว

3.ภาคอินเด็กซ์รีจิสเตอร์

รายการอุปกรณ์	จำนวน
1. 74LS273	4 ตัว
2. 74LS541	8 ตัว
3. คอนเน็คเตอร์ 8 ขา	7 ตัว
4. คอนเน็คเตอร์ 4 ขา	2 ตัว
5. คอนเน็คเตอร์ 3 ขา	1 ตัว
6. คอนเน็คเตอร์ 2 ขา	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ (ต่อ)

4.ภาคโปรแกรมเคาท์เตอร์ 16 บิต

รายการอุปกรณ์	จำนวน
1. 74LS193	4 ตัว
2. 74LS541	2 ตัว
3. คอนเน็คเตอร์ 8 ขา	5 ตัว
4. คอนเน็คเตอร์ 5 ขา	1 ตัว

5.ภาคกระทำทางคณิตศาสตร์ 8 บิต

รายการอุปกรณ์	จำนวน
1. 74LS181	2 ตัว
2. 74LS374	1 ตัว
3. คอนเน็คเตอร์ 8 ขา	4 ตัว
4. คอนเน็คเตอร์ 8 ขา	2 ตัว

6.ภาคแปลงสัญญาณแอดเดรส

รายการอุปกรณ์	จำนวน
1. 74LS541	4 ตัว
2. คอนเน็คเตอร์ 8 ขา	5 ตัว
3. คอนเน็คเตอร์ 2 ขา	1 ตัว
4. คอนเน็คเตอร์ 3 ขา	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ (ต่อ)

7.ภาคเฟทซ์และดีโค๊ดเดอร์

รายการอุปกรณ์	จำนวน
1. 27128	1 ตัว
2. 74LS273	1 ตัว
3. 4017	1 ตัว
4. 74LS27	1 ตัว
5. 74LS08	1 ตัว
6. 74LS193	1 ตัว
7. คอนเน็คเตอร์ 8 ขา	3 ตัว

8.ภาคบวกเลข 16 บิต

รายการอุปกรณ์	จำนวน
1. 74LS181	4 ตัว
2. 74LS374	2 ตัว
3.คอนเน็คเตอร์ 8 ขา	6 ตัว
4.คอนเน็คเตอร์ 5 ขา	1 ตัว
5.คอนเน็คเตอร์ 2 ขา	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ (ต่อ)

9.ภาคแปลงคีย์เป็นสัญญาณ BCD-8421

รายการอุปกรณ์	จำนวน
1. 74C923	1 ตัว
2. 74LS08	1 ตัว
3. 74LS04	1 ตัว
4. MC14017	2 ตัว
5. 74LS76	1 ตัว
6. C 0.01 μF 50 V	1 ตัว
7. C 1 μF 50 V	1 ตัว
8. คอนเน็คเตอร์ 4 ขา	3 ตัว
9. คอนเน็คเตอร์ 2 ขา	1 ตัว

10.ภาคโปรแกรมแอดเดรส

รายการอุปกรณ์	จำนวน
1. 74LS193	4 ตัว
2. 74LS374	4 ตัว
3. คอนเน็คเตอร์ 8 ขา	2 ตัว
4. คอนเน็คเตอร์ 4 ขา	2 ตัว
5. คอนเน็คเตอร์ 2 ขา	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ (ต่อ)

11.ภาคนับแบบวงแหวน 54 บิต

รายการอุปกรณ์	จำนวน
1. 4017	7 ตัว
2. 74LS08	2 ตัว
3. NE 555	1 ตัว
4. R 10 kΩ	1 ตัว
5. VR 100 kΩ	1 ตัว
6. C 0.1 μF 50 V	1 ตัว
7. C 0.01 μF 50 V	1 ตัว

12.ภาคเลือกสัญญาณ BCD-8421

รายการอุปกรณ์	จำนวน
1. 74LS241	6 ตัว
2. 74LS04	1 ตัว
3. C 0.1 μF 50 V	1 ตัว
4. คอนเน็คเตอร์ 8 ขา	1 ตัว
5. คอนเน็คเตอร์ 4 ขา	1 ตัว
6. คอนเน็คเตอร์ 2 ขา	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์ (ต่อ)

13.ภาคแปลงสัญญาณ BCD 8421 ขับตัวเลข แสดงผล 7 ส่วน

รายการอุปกรณ์	จำนวน
1. 27128	1 ตัว
2. คอนเน็คเตอร์ 4 ขา	5 ตัว
3. คอนเน็คเตอร์ 2 ขา	1 ตัว

14.ภาคแสดงผล 7 ส่วน 54 หลัก

รายการอุปกรณ์	จำนวน
1.ส่วนแสดงผล 7 ส่วนคาโถดร่วม C – 562	54 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ไพฑูรย์วัฒนกิจ.“ไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์.” กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538

บุญวัฒน์ อัตชู.“ทฤษฎีและการใช้งานไมโครโปรเซสเซอร์.” กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายธเนศ สุขแสง
วันเดือนปีเกิด	22 สิงหาคม 2522
สถานที่เกิด	จังหวัดยโสธร
ภูมิลำเนาเดิม	33 หมู่ 7 ตำบลกระจาย อำเภอป่าติ้ว จังหวัดยโสธร 35150
ที่อยู่ปัจจุบัน	33 หมู่ 7 ตำบลกระจาย อำเภอป่าติ้ว จังหวัดยโสธร 35150
โทรศัพท์	0-1549-9581, 0-1390-1367
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนชุมชนบ้านหนองคู
มัธยมศึกษาตอนต้น	โรงเรียนยโสธรพิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	โรงเรียนโปลีเทคนิคภาคตะวันออกเฉียงเหนือ
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	วิทยาลัยเทคนิคนครราชสีมา
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
คติพจน์	น้ำเน่ายังมีเงาพระจันทร์ ดวงตะวันยังมีจุดดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายอรรณพ เรืองยศจันทนา
วันเดือนปีเกิด	18 มกราคม 2524
สถานที่เกิด	จังหวัดฉะเชิงเทรา
ภูมิลำเนาเดิม	11 ถนนศุภกิจ ตำบลหน้าเมือง
	อำเภอเมือง จังหวัดฉะเชิงเทรา 24000
ที่อยู่ปัจจุบัน	11 ถนนศุภกิจ ตำบลหน้าเมือง
	อำเภอเมือง จังหวัดฉะเชิงเทรา 24000
โทรศัพท์	0-3851-2425
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนทวีวิทยาคุณ (ป.1-ป.5)
	โรงเรียนปัญจพิทยาคาร (ป.6)
มัธยมศึกษาตอนต้น	โรงเรียนเซ็นหลุยส์
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคฉะเชิงเทรา
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคฉะเชิงเทรา
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
คติพจน์	คิดดี พูดดี ทำดี ได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายบุญชนะ นุชชม
วันเดือนปีเกิด	17 กันยายน 2521
สถานที่เกิด	จังหวัดพิษณุโลก
ภูมิลำเนาเดิม	98 หมู่ 2 ตำบลท้อแท้ อำเภอวัดโบสถ์ จังหวัดพิษณุโลก 65160
ที่อยู่ปัจจุบัน	98 หมู่ 2 ตำบลท้อแท้ อำเภอวัดโบสถ์ จังหวัดพิษณุโลก 65160
โทรศัพท์	0-9859-1745
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนจ่าการบุญ
มัธยมศึกษาตอนต้น (ม.3)	โรงเรียนพิษณุโลกพิทยาคม
มัธยมศึกษาตอนปลาย (ม.6)	โรงเรียนพิษณุโลกพิทยาคม
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	โรงเรียนบริหารธุรกิจ และเทคโนโลยีพิษณุโลก
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
คติพจน์	จงอย่าลืมตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้