

ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท วิศวกรรมความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA
TEN ORDER FIR DIGITAL FILTER IMPLEMENTATION WITH FPGA DEVICE

ชื่อนักศึกษา	1. นายกษานต์ ศรีกุลนาถ	รหัสประจำตัว	40031402
	2. นางสาวธิดารัตน์ บุตรโคตร	รหัสประจำตัว	40031414
	3. นางสาวประทีป เข้มทอง	รหัสประจำตัว	40031421
	4. นายศักดิ์ชัย ศรีมากรณ์	รหัสประจำตัว	40031429

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ควบคุมปริญญาโท

1. อาจารย์ วรวิทย์ สมหา
2. อาจารย์ กิตติพงศ์ มะโน



คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์ วรวิทย์ สมหา	
2. อาจารย์ อัมพล ทองระอา	
3. อาจารย์ โภศล ตราชู	
4. อาจารย์ อมรชัย ชัยชนะ	
5. อาจารย์ สุรัชย์ พิมพ์สาลี	

วัน / เดือน / ปีที่สอบ วันที่ 22 เดือน พฤศจิกายน พ.ศ. 2541 เวลา 19.30 ถึง 19.45

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม



ภาควิชารับรองแล้ว

ศาสตราจารย์ ดร.วิมล เทพหัสดิน ณ อยุธยา

ผู้อำนวยการภาควิชาครุศาสตร์วิศวกรรม

.....เดือน..... พ.ศ.

เลขหม.....

เลขทะเบียน..... 32824

วัน, เดือน, ปี 10 สิงหาคม 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง วงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA_s
TEN ORDER FIR DIGITAL FILTER IMPLEMENTATION WITH FPGA_s DEVICE

ผู้จัดทำ

- | | |
|--------------------|-----------|
| 1. นายกษานต์ | ศรีกุลนาถ |
| 2. นางสาววิภารัตน์ | บุตรโคตร |
| 3. นางสาวประทีป | เข้มทอง |
| 4. นายศักดิ์ชัย | ศรีมากรณ์ |

อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลงนาม
(อาจารย์วรวิทย์ สมหา)

ลงนาม
(อาจารย์กิติพงศ์ มะโน)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม
(ผศ. ดร.ธีรภัศ เทพหัสติน ณ อุซุชยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง วงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA_s

TEN ORDER FIR DIGITAL FILTER IMPLEMENTATION WITH FPGA_s DEVICE

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ อุปกรณ์ FPGA_s
2. เพื่อทำการออกแบบวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ อุปกรณ์ FPGA_s
3. เพื่อทำการสร้างวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ อุปกรณ์ FPGA_s
4. เพื่อนำวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ อุปกรณ์ FPGA_s ไปใช้งานได้

ประโยชน์ที่คาดว่าจะได้รับ

1. นำหลักการของการประมวลผลเชิงเลขมาสร้างเป็นวงจรกรองความถี่แทนวงจรกรองที่ถูกสร้างขึ้นด้วยระบบดิจิทัล
2. ทำการโปรแกรมลงบนชิปทำให้ได้ชิปที่มีขนาดเล็ก
3. นำไปใช้งานในด้านการสื่อสารหรือเทคโนโลยีที่ต้องการความถูกต้องและมีขนาดเล็ก เช่น โทรศัพท์มือถือระบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้
อุปกรณ์ FPGA_s

นายกษานต์	ศรีกุลนาถ
นางสาวธิดารัตน์	บุตร โคตร
นางสาวประทีป	เข้มทอง
นายศักดิ์ชัย	ศรีมากรณ์

อาจารย์ที่ปรึกษา

อาจารย์วรวิทย์	สมหา
อาจารย์กิตติพงศ์	มะโน

ปีการศึกษา 2541

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการออกแบบของระบบการประมวลผลสัญญาณเชิงเลข โดยนำมาสร้างเป็นวงจรรองเชิงเลข แบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ FPGA_s วงจรรองเชิงเลขแบบผลตอบสนองจำกัด ซึ่งสามารถสุ่มสัญญาณความถี่ได้ไม่เกิน 8 กิโลเฮิร์ตซ์ มีความถี่ในการปฏิบัติงานไม่เกิน 4 กิโลเฮิร์ตซ์ นอกจากนี้ยังสามารถโปรแกรมให้เป็นวงจรรองผ่านความถี่, วงจรรองผ่านความถี่ต่ำ หรือวงจรรองผ่านความถี่สูง โดยการกำหนดข้อมูลจาก EPROM

TEN ORDER FIR DIGITAL FILTER IMPLEMENTATION WITH FPGA_s DEVICE

MR.KASAN	SRIKULNATH
MISS.TIDARAT	BOOTHKOT
MISS.PRATEEP	KHEMTHONG
MR.SAKCHAI	SRIMAKORM

ADVISOR

MR.WORAWIT	SOMHA
MR.KITIPONG	MANO

1998

ABSTRACT

This thesis is present implementation of ten order FIR digital signal processing with FPGA_s devices. The system consisted a ADC, DAC EPROM and signal processing unit (FPGA_s), The specification of filter, it use sampling frequency is 8 kHz and, operate frequency is 4 kHz. It can change cut off frequency by changing coefficients data in EPROM. Experiment measurements using this design methodologics as well as digital simulation in the view simulation program.

III

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ดี ด้วยความพยายาม ความอดทน ความตั้งใจ รวมทั้งความรักความสามัคคีปรองมอดของคณะผู้จัดทำ ซึ่งร่วมมือร่วมใจกันต่อสู้แก้ไขปัญหาที่เกิดขึ้นมากมายในการทำโครงการครั้งนี้ อีกทั้งยังมีผู้ให้ความช่วยเหลือในการทำงานครั้งนี้อีกมากมาย อาทิ เช่น เพื่อน ๆ ทุกคนที่คอยช่วยเหลือให้คำปรึกษา ครูอาจารย์ที่ช่วยอบรมสั่งสอน บิдамารดาผู้ให้กำเนิด และคนรอบข้างที่คอยให้กำลังใจ

ขอขอบพระคุณท่านอาจารย์ผู้ควบคุมปริญญาโท และอาจารย์ประจำภาควิชาครุศาสตร์ วิศวกรรมทุกท่านที่ให้คำแนะนำที่เป็นประโยชน์ต่อการทำโครงการครั้งนี้

คณะผู้จัดทำขอขอบคุณงามความดีทั้งหมดที่ได้กล่าวมานี้มอบให้แก่ผู้มีพระคุณทุกท่านด้วยความเคารพและเทิดทูน



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ชี้ความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	1
บทที่ 2 หลักการและทฤษฎี	3
2.1 บทนำ	3
2.2 ส่วนประกอบในระบบประมวลผลสัญญาณคิจิตอล	3
2.2.1 วงจรสุ่มสัญญาณ	3
2.2.2 วงจรควอน ไตซ์เซชั่น	4
2.2.3 วงจรแบ่งสัญญาณรวมกับการแบ่งชั้นสัญญาณ	4
2.2.4 วงจรประมวลผลสัญญาณ	4
2.2.5 วงจรสร้างสัญญาณคีน	5
2.3 ประโยชน์ของการประมวลผลเชิงเลข	5
2.4 ข้อดีของการประมวลผลสัญญาณคิจิตอล	5
2.5 การสุ่มสัญญาณและการสร้างสัญญาณคีน	6
2.6 ทฤษฎีการสุ่มสัญญาณ	7
2.7 การสร้างสัญญาณกลับคีน	9
2.8 การแบ่งชั้นสัญญาณควอน ไตซ์เซชั่น	11
2.9 ระบบแบบไม่ต่อเนื่อง	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.10 ความเป็นเชิงเส้นและไม่แปรตามเวลา	12
2.11 ผลตอบสนองจากสัญญาณอิมพัลส์	13
2.12 ตัวกรองแบบจำกัดผลตอบสนองความถี่และ ตัวกรองแบบไม่จำกัดผลตอบสนองความถี่	15
2.13 สมการผลต่าง	16
2.14 เสถียรภาพ	16
2.15 ตัวกรองแบบจำกัดผลตอบสนองความถี่	18
2.16 การออกแบบโดยวิธีวิน โควส์	18
2.17 การสร้างตัวกรองจำกัดผลตอบสนองความถี่	20
2.18 ทฤษฎีและหลักการของ FPGA _s	22
2.18.1 โครงสร้างภายใน	22
2.18.2 ส่วนประกอบของลอจิก	24
2.18.3 ส่วนอินพุตและเอาต์พุต	25
2.18.4 รายละเอียดการใช้งาน	26
2.18.5 การใช้งานแบบสเตปซีเรียล	28
2.18.6 การใช้งานในแบบมาสเตอร์ซีเรียล	30
2.18.7 ข้อควรระวังในการใช้ FPGA _s	31
2.18.8 การใช้ซอฟต์แวร์ของบริษัทเววลอจิกและบริษัทไซลิงค์	32
บทที่ 3 การออกแบบและการสร้าง	34
3.1 กล่าวนำ	34
3.1.1 ข้อกำหนดในการทำงานของบล็อกไดอะแกรมแต่ละส่วน	34
3.2 การออกแบบและการสร้างวงจรแต่ละส่วน	35
3.2.1 วงจรบวกขนาด 16 บิต	35
3.2.2 วงจรคูณเลข 8 บิตคูณ 8 บิต	37
3.2.3 วงจรเขียนข้อมูลขนาด 8 บิต	39
3.2.4 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	40
3.2.5 วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก	41

VI

สารบัญ (ต่อ)

เรื่อง	หน้า
3.2.6 วงจรนับ 0-9	43
3.2.7 วงจรเลขชี้	47
3.2.8 วงจรถอดรหัส	48
3.2.9 หน่วยประมวลผลของวงจรกรองเชิงเลข อันดับ 10 แบบจำกัดผลตอบสนองความถี่	50
3.3 การเชื่อมต่อวงจรถอดรหัสหน่วยประมวลผลวงจรถอด ความถี่อันดับ 10 และวงจรกำเนิดความถี่สุ่ม 8 กิโลเฮิร์ตซ์	52
3.4 สรุปขั้นตอนการออกแบบและการสร้าง	53
3.4.1 ขั้นตอนการออกแบบและจำลองการทำงาน โดยใช้ซอฟต์แวร์ของบริษัททวิลोजิก	53
3.4.2 ขั้นตอนการโปรแกรมลงบนอุปกรณ์ FPGA _s โดยใช้ซอฟต์แวร์ XDM ของบริษัทไซลิงค์	61
บทที่ 4 การทดลองและผลการทดลอง	68
4.1 การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล	68
4.1.1 ลำดับขั้นการทดลอง	68
4.1.2 สรุปผลการทดลอง	68
4.2 การทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	70
4.2.1 ลำดับขั้นการทดลอง	70
4.2.2 สรุปผลการทดลอง	70
4.3 การทดสอบแบบจำลองบนโปรแกรม View Trace	72
4.3.1 การกำหนดค่าสัญญาณของวงจรทั้งหมด	72
4.3.2 ผลการทดลอง	73
4.4 การทดลองการทำงานของบอร์ดวงจรกรอง	74
บทที่ 5 สรุปและวิจารณ์	77
5.1 สรุป	77
5.2 ปัญหาที่พบ	77

เอกสารนี้เป็นเอกสาร 5.2.1 ส่วนของฮาร์ดแวร์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านก 77 คำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VII

สารบัญ (ต่อ)

เรื่อง	หน้า
5.2.2 ส่วนของซอฟต์แวร์	77
5.2.3 ส่วนของการออกแบบ	77
5.3 การพัฒนา	78
5.3.1 ส่วนของฮาร์ดแวร์	78
5.3.2 การพัฒนาการออกแบบ	78
5.3.3 การพัฒนาซอฟต์แวร์	78
ภาคผนวก ก รายละเอียดอุปกรณ์	79
ภาคผนวก ข การต่อวงจรรวม ADC, DAC และวงจร FPGA _s	100
บรรณานุกรม	102
ประวัติผู้แต่ง	103



VIII

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การนับเขตของเขตพื้นที่ฐาน	22
ตารางที่ 2.2 รูปแบบต่างๆ ของการกำหนดองค์ประกอบในการทำงาน	27
ตารางที่ 3.1 สภาวะการทำงานของวงจรมี 0-9	45
ตารางที่ 3.2 สภาวะการทำงานของวงจรถอดรหัส	50
ตารางที่ 4.1 ผลการทดลองของวงจรถอดรหัส A/D	68
ตารางที่ 4.2 ผลการทดลองของวงจรถอดรหัส D/A	70



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 2.1 การสุมสัญญาณด้วยสวิตช์อุดมคติ	6
รูปที่ 2.2 สัญญาณที่เกิดขึ้นเมื่อใช้ F_s ต่ำกว่า $2F_{max}$	7
รูปที่ 2.3 ผลตอบสนองความถี่ของตัวกรองแบบช่วงความถี่ซับซ้อน	8
รูปที่ 2.4 การสุมด้วยความถี่สูงกว่า $2F_{max}$ มาก ๆ	9
รูปที่ 2.5 สัญญาณขาเข้าและขาออก	10
รูปที่ 2.6 ผลตอบสนองของสัญญาณอิมพัลส์	13
รูปที่ 2.7 การหาผลตอบสนองของระบบโดยการประสาน	14
รูปที่ 2.8 ผลตอบสนองของสัญญาณอิมพัลส์ของตัวกรองแบบจำกัด	16
รูปที่ 2.9 ระบบที่ไม่เป็นเหตุภาพ	18
รูปที่ 2.10 ผลตอบสนองความถี่ของตัวกรองอุดมคติ	19
รูปที่ 2.11 แผนภาพสร้างตัวกรองแบบจำกัดผลตอบสนองโดยการประสาน	21
รูปที่ 2.12 แผนภาพการสร้างตัวกรองแบบจำกัดผลตอบสนองอาศัยสมมาตร	21
รูปที่ 2.13 แผนภาพการสร้างตัวกรองแบบจำกัดผลตอบสนองโดยการประสานรวดเร็ว	21
รูปที่ 2.14 แผนผัง CUB ตระกูล 4000	25
รูปที่ 2.15 แผนผัง IBO _s ตระกูล 4000	26
รูปที่ 2.16 ผังอันดับในการคอนฟิกเมื่อเริ่มป้อนแหล่งจ่ายไฟฟ้า	28
รูปที่ 2.17 การต่อใช้งานแบบสเตลเฟซีเลียน	29
รูปที่ 2.18 แผนภูมิเวลาการป้อนข้อมูลโปรแกรมคอนฟิกแบบสเตลเฟซีเลียน	29
รูปที่ 2.19 การต่อใช้งานในลักษณะมาสเตอร์ซีเลียน	30
รูปที่ 2.20 การต่อใช้งานแบบมาสเตอร์ซีเลียน	31
รูปที่ 3.1 บล็อกไดอะแกรมวงจรกรองเชิงเลข	34
รูปที่ 3.2 บล็อกไดอะแกรมวงจรบวกเลขขนาด 16 บิต	35
รูปที่ 3.3 วงจรบวกเลขขนาด 16 บิต	36
รูปที่ 3.4 วงจรคูณเลข 8 บิตคูณ 8 บิต	38
รูปที่ 3.5 วงจรเลื่อนข้อมูลขนาด 8 บิต	39
รูปที่ 3.6 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	40
รูปที่ 3.7 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	42

สารบัญรูปภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.8 วงจรนับ 0-9	44
รูปที่ 3.9 วงจรเลขชี้	48
รูปที่ 3.10 วงจรถอดรหัส	49
รูปที่ 3.11 หน่วยประมวลผลวงจรรองเชิงเลขแบบจำกัดผลตอบสนองอันดับ 10	51
รูปที่ 3.12 การเชื่อมต่อหน่วยประมวลผลเข้ากับวงจรถอดรหัส	52
รูปที่ 3.13 หน้าต่างการใช้งาน โปรแกรมหลัก View Logic	53
รูปที่ 3.14 หน้าต่างวงจรถอดรหัส Cockpit	54
รูปที่ 3.15 การสร้าง Project File	54
รูปที่ 3.16 การติดตั้ง Library	55
รูปที่ 3.17 การสร้าง Schematic	55
รูปที่ 3.18 การสร้าง Symbol	56
รูปที่ 3.19 หน้าต่าง View Draw / File Open	56
รูปที่ 3.20 หน้าต่าง View Draw	57
รูปที่ 3.21 การใส่ Component	57
รูปที่ 3.22 Component ที่ถูกเลือก	58
รูปที่ 3.23 การใช้ View Sim	59
รูปที่ 3.24 การกำหนดการทำงานของขาสัญญาณ	60
รูปที่ 3.25 หน้าต่างการกำหนดขาสัญญาณกับ โปรแกรม View Trace	60
รูปที่ 3.26 การทำงานของ View Trace เมื่อสั่งคำสั่ง SIM	61
รูปที่ 3.27 การเริ่มต้นของโปรแกรม XACT Design Manager	62
รูปที่ 3.28 หน้าต่าง Design Manager	62
รูปที่ 3.29 หน้าต่าง New Project	63
รูปที่ 3.30 หน้าต่างอินพุต Desing	63
รูปที่ 3.31 หน้าต่าง Translate	64
รูปที่ 3.32 หน้าต่าง Part Selecter	64
รูปที่ 3.33 ผลการ Translate	65
รูปที่ 3.34 หน้าต่าง XC4000 Desing Implementation	65

สารบัญรูปภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.35 หน้าต่างกระบวนการสร้างไฟล์บัสสตรีม	66
รูปที่ 3.36 หน้าต่าง Communication Ation Setup	66
รูปที่ 4.1 การทดลองวงจร A / D	69
รูปที่ 4.2 ผลการทดลองวงจร A / D	69
รูปที่ 4.3 การทดลองวงจร D / A	71
รูปที่ 4.4 ผลการทดลองวงจรวงจร D / A	71
รูปที่ 4.5 การกำหนดขาสัญญาณ View Sim	72
รูปที่ 4.6 เอาต์พุตที่ได้เมื่อทำงานบน โปรแกรม View Trace	73
รูปที่ 4.7 การบอร์ควงจรรองความถี่เชิงเลขแบบจำกัดผลตอบสนอง	74
รูปที่ 4.8 รูปคลื่นแรงดันทางอินพุต	75
รูปที่ 4.9 ผลตอบสนองความถี่ทางด้านเอาต์พุตในช่วงความถี่ต่ำผ่าน	75
รูปที่ 4.10 กราฟช่วงการคัดออกของความถี่ต่ำ	77



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริิณญาานิพนธ์

โดยทั่วไปการสร้างวงจรความถี่ส่วนใหญ่จะถูกสร้างขึ้นในระบบของสัญญาณ แอนะล็อกซึ่งระบบดังกล่าวนี้จะทำให้การทำงานของวงจรไวต่อสภาพแวดล้อมทำให้ผลของสัญญาณขาดความน่าเชื่อถือ ดังนั้นในปัจจุบันจึงเกิดแนวคิดที่จะสร้างวงจรที่สามารถทำหน้าที่เป็นวงจรฟิลเตอร์ได้ โดยการนำระบบดิจิทัลมาแทนการทำงานในระบบแอนะล็อก ระบบดังกล่าวนี้ถูกเรียกว่า การประมวลผลสัญญาณเชิงเลข เพื่อที่จะลดความผิดพลาดต่าง ๆ ที่เกิดขึ้นจากวงจรในระบบแอนะล็อก จากเหตุผลดังกล่าวมาข้างต้นจึงเป็นผลทำให้นำหลักการการประมวลผลเชิงเลขดังกล่าวมาสร้างเป็นวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ FPGA ขึ้น

วัตถุประสงค์ของการทำปริิณญาานิพนธ์ เพื่อศึกษาการทำงานของวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้หลักการทำงานของวงจรประมวลผลเชิงเลข อีกทั้งยังเพื่อทำการออกแบบและจัดสร้างวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 รวมทั้งการเขียนโปรแกรมเพื่อควบคุมการทำงานของวงจรความถี่แบบจำกัดผลตอบสนองอันดับ 10 ได้ตามแนวทางที่กำหนดให้ ทั้งนี้เป็นแนวทางในการศึกษาและพัฒนาความรู้ในระบบประมวลผลเชิงเลขต่อไป

1.2 ขีดความสามารถของโครงการ

ขีดความสามารถของวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 นี้ สามารถสุ่มสัญญาณความถี่ได้ไม่เกิน 8 กิโลเฮิรตซ์ และมีความถี่ในการปฏิบัติงานไม่เกิน 4 กิโลเฮิรตซ์ อีกทั้งยังสามารถโปรแกรมให้เป็น วงจรกรองผ่านความถี่กลาง, วงจรกรองผ่านความถี่ต่ำ หรือวงจรกรองผ่านความถี่สูง ได้ซึ่งการกำหนดชนิดของวงจรความถี่กำหนดได้จากข้อมูลใน EPROM

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริิณญาานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่าง ๆ เพื่อความสะดวกและเข้าใจต่อการศึกษา ในแต่ละบทจะประกอบด้วยเนื้อหาสำคัญดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 หลักการและทฤษฎี ในบทนี้จะกล่าวถึงทฤษฎีกับดิจิทัลฟิลเตอร์ ส่วนประกอบในระบบประมวลผลสัญญาณดิจิทัล การสุ่มสัญญาณการสร้างสัญญาณในระบบแบบไม่ต่อเนื่อง เติลยรภาพ ตัวกรองแบบจำกัดผลตบสนองความถี่ และทฤษฎีพื้นฐานของ FPGA_s

บทที่ 3 การออกแบบการสร้าง ในบทนี้จะกล่าวถึงการสร้างและการออกแบบวงจรบวกเลขขนาด 16 บิต วงจรคูณเลข 8 บิตคูณ 8 บิต วงจรเลื่อนข้อมูลขนาด 8 บิต วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก วงจรนับ 0-9 วงจรถอดรหัส วงจรแก้พีช และวงจรรองความถี่เชิงเลขแบบจำกัดผลตบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA_s

บทที่ 4 การทดลองและผลการทดลอง ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก และวงจรรองความถี่เชิงเลขแบบจำกัดผลตบสนองอันดับ 10

บทที่ 5 สรุปปัญหาแนวทางการแก้ไขปัญหาที่เกิดขึ้นตลอดจนทั้งข้อบกพร่องที่พบ ในการทำงานของวงจรรองความถี่เชิงเลขแบบจำกัดผลตบสนองอันดับ 10 พร้อม ทั้งแนวคิดพัฒนาวงจรรองเชิงเลขแบบจำกัดผลตบสนองอันดับ 10 ให้มีประสิทธิภาพและการใช้งานที่สูงกว่าเดิม

ภาคผนวก แสดงรายละเอียดของอุปกรณ์ต่าง ๆ และการต่อวงจรรวม ADC, DAC และวงจร FPGA_s

ภาคผนวก ก รายละเอียดของอุปกรณ์ต่าง ๆ

ภาคผนวก ข การต่อวงจรรวม ADC, DAC และวงจร FPGA_s

บทที่ 2

หลักการและทฤษฎี

2.1 บทนำ

โดยทั่วไปดิจिटอลมักจะมี ความหมายทางด้านปริมาณเชิงเลข โดยตรง ส่วนฟิลเตอร์ในความหมายทางไฟฟ้าอิเล็กทรอนิกส์ หมายถึง การยอมรับให้สัญญาณไฟฟ้าที่ผ่านความถี่ความถี่หนึ่งผ่านได้

จริง ๆ แล้วดิจिटอลฟิลเตอร์มิได้จำกัดอยู่แต่เฉพาะสาขาวิชาอิเล็กทรอนิกส์ ในการกรองความถี่ของสัญญาณเท่านั้น แต่ดิจिटอลฟิลเตอร์มิใช้อยู่โดยทั่วไปทุกสาขาที่มีการวัดค่าตัวแปรในรูปของปริมาณ ตัวอย่างเช่น การวัดความดันเลือด การวัดความสั้นสะเทือนของผิวโลก การวัดระดับศักดาไฟฟ้าที่เกิดจากสัญญาณเสียงพูดในการสนทนาทางโทรศัพท์ การวัดความเข้มแสงของดวงดาว การวัดจำนวนประชากร การวัดคลื่นสมอง การวัดปริมาณการจำหน่ายสินค้าในตลาด จะเห็นว่าปริมาณต่าง ๆ เหล่านี้จะแปรผันไปตามเวลา ซึ่งเรามักจะแทนฟังก์ชันที่ขึ้นกับเวลาว่าเป็น $X(t)$ จริง ๆ แล้วจะเป็นปริมาณที่ต่อเนื่องกันไป แต่ในด้านปฏิบัติแทนที่จะเก็บข้อมูลเป็น $X(t)$ เราเพียงแต่เก็บข้อมูลเป็นช่วงเท่ากัน โดยขบวนการสุ่มสัญญาณ X_n ของฟังก์ชัน $X(t)$ และประมาณค่าระดับสัญญาณวิธีพิเศษขึ้น หรือพิเศษทิ้ง แทนเพื่อให้จำนวนหลักของตัวเลขลดลงในปริมาณที่จำกัด การแทนปริมาณในรูปหลักของตัวเลข (ในระบบดิจिटอลใช้เลขฐานสอง) เราเรียกว่าการควอนไทซ์ ที่ทำการควอนไทซ์เซชันได้นั้น เราก็จะนำไปเปลี่ยนเป็นสัญญาณดิจिटอลและนำไปประมวลผลซึ่งง่ายต่อความเข้าใจ

2.2 ส่วนประกอบในระบบประมวลผลสัญญาณดิจिटอล

2.2.1 วงจรสุ่มสัญญาณ

วงจรสุ่มสัญญาณจะมีสัญญาณขาเข้าของวงจรนี้เป็นสัญญาณแบบแอนะล็อก หรือสัญญาณต่อเนื่องซึ่งจะเขียนเป็นสัญลักษณ์ได้เป็น $x(t)$ โดยที่ t แทนเวลาที่มีความต่อเนื่อง ส่วนสัญญาณขาออกเป็นสัญญาณไม่ต่อเนื่อง ซึ่งเขียนสัญลักษณ์ได้เป็น $x(n)$ โดยที่ n เป็นตัวชี้ที่มีค่าไม่ต่อเนื่อง โดยทั่วไปจะใช้ n เป็นจำนวนเต็ม นั่นคือ $n = \dots, -2, -1, 0, 1, 2, \dots$ ถ้าให้ความถี่ของการสุ่มเป็น f_s และคาบของการสุ่มเป็น $T = 1/f_s$ โดยสุ่มที่ทุก ๆ เวลา $= nT$ จะได้ว่า $x(n) = x(t)|_{t=nT}$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 วงจรควอนไทซ์เซชัน

วงจรควอนไทซ์เซชันหรือวงจรแบ่งขั้นสัญญาณ สัญญาณ $x(n)$ ถึงแม้มีความไม่ต่อเนื่องในเชิงเวลาแต่ยังมีความต่อเนื่องในเชิงขนาดอยู่จริงนี้จะทำการแบ่งขั้นของสัญญาณและเข้ารหัสเป็นเลขฐานสองซึ่งจะให้สัญญาณขาออกเป็นสัญญาณดิจิทัล (แทนด้วยรหัสของค่า 0 กับ 1) ความละเอียดของการแบ่งขั้นขึ้นอยู่กับจำนวนบิตที่ใช้แทนแต่ละจุดของ $x(n)$

2.2.3 วงจรสุ่มสัญญาณรวมกับการแบ่งขั้นสัญญาณ

วงจรสุ่มสัญญาณรวมกับการแบ่งช่วงสัญญาณในทางปฏิบัติก็คือตัวแปลงสัญญาณแอนะล็อกเป็นดิจิทัลนั่นเอง

คำว่าสัญญาณดิจิทัลกับสัญญาณไม่ต่อเนื่อง โดยทั่วไปหมายถึงสัญญาณในลักษณะเดียวกัน แต่สัญญาณไม่ต่อเนื่อง หมายถึง สัญญาณที่มีความไม่ต่อเนื่องในเชิงเวลาเท่านั้น นั่นก็คือสัญญาณขาออกของข้อ 1 นั่นเอง ส่วนคำว่าดิจิทัลมักหมายถึงสัญญาณไม่ต่อเนื่องที่แต่ละจุดถูกแบ่งขั้น และเข้ารหัสอยู่ในรูป 0 กับ 1 แล้ว ในเนื้อหาของวิชานี้เราจะพูดถึงคำว่าสัญญาณไม่ต่อเนื่องเป็นส่วนใหญ่ นั่นก็คือเราจะละเลยผลของการแบ่งขั้นสัญญาณชั่วคราวและถือเอาว่าสัญญาณ $x(n)$ เป็นสัญญาณขาเข้าของวงจรประมวลผลสัญญาณเลย อย่างไรก็ตามจะย้อนกลับมาดูความคลาดเคลื่อนของระบบที่เราออกแบบ ซึ่งเป็นผลมาที่เกิดจากการแบ่งขั้นสัญญาณอีกครั้ง

2.2.4 วงจรประมวลผลสัญญาณ

วงจรประมวลผลสัญญาณเป็นวงจรที่ประมวลผลสัญญาณ $x(n)$ เพื่อกระทำผลบางอย่าง เช่น เป็นวงจรกรองความถี่และให้สัญญาณขาออกเป็น $y(n)$ ข้อนี้เป็นส่วนหลักที่เราจะเกี่ยวข้องด้วย ในวิชานี้ วงจรประมวลผลสัญญาณนี้แท้ที่จริงก็คือ การคำนวณนั่นเอง ถ้าเรามองที่สัญญาณขาออกขณะหนึ่ง ๆ เช่น $y(10)$ จะได้ว่า $y(10)$ สามารถเกิดจากการประมวลผลกัน (บวก, คูณ, ... ,) ระหว่างสัญญาณขาเข้าในปัจจุบัน คือ $x(10)$, สัญญาณขาเข้าในอดีต ได้แก่ $x(9), x(8), \dots$, สัญญาณขาเข้าในอนาคต ได้แก่ $x(11), x(12), \dots$, อาจรวมถึงสัญญาณขาออกในอดีตด้วย ได้แก่ $y(9), y(8), \dots$ การคำนวณดังกล่าว สามารถกระทำโดยวิธีการดังต่อไปนี้

1. การเขียนซอฟต์แวร์ เพื่อใช้กับ CPU ทั่ว ๆ ไป ซึ่งอาจเป็นการกระทำในคอมพิวเตอร์หรือใช้วงจรที่ตนเองและมีไมโครโพรเซสเซอร์ควบคุม

2. การใช้ CPU ที่ออกแบบเฉพาะสำหรับงานประมวลผลสัญญาณหรือเรียกกันทั่วไปว่าการประมวลผลสัญญาณเชิงเลข หรือ DSP chips ซึ่ง CPU ประเภทนี้ถูกออกแบบมาให้สามารถทำฟังก์ชันที่ใช้ในการประมวลผลสัญญาณได้อย่างรวดเร็ว บางชนิดยังสามารถทำการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลาย ๆ ส่วนได้พร้อมกันในตัวเดียวกันด้วย บริษัทที่เป็นผู้นำด้านการผลิต CPU ประเภทนี้ ได้แก่ TEXAS Instruments, โมโตโลลา, วงจรขับสัญญาณแอนะล็อก และ AT&T เป็นต้น

3. การใช้ฮาร์ดแวร์หรือไอซีซีพ ที่ออกแบบเฉพาะเพื่อทำงานประมวลผลสัญญาณเฉพาะทาง

4. การเลือกใช้ที่ขึ้นอยู่กับลักษณะงานและต้นทุน สำหรับงานประมวลผลสัญญาณแบบตามเวลาจริง อาจต้องใช้ DSP chip หรือฮาร์ดแวร์ พิเศษในการทำการประมวลผล

2.2.5 วงจรสร้างสัญญาณคีน

วงจรสร้างสัญญาณคีนใช้ในระบบที่มีสัญญาณขาออกสุดท้ายเป็นสัญญาณต่อเนื่อง โดยทำหน้าที่แปลงสัญญาณไม่ต่อเนื่อง $y(n)$ ให้กลับเป็นสัญญาณต่อเนื่อง $y(t)$ ซึ่งจะสามารถเป็นขาออกสุดท้ายของระบบ วงจรประเภทนี้ก็คือ วงจรที่แปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกนั่นเอง

2.3 ประโยชน์ของการประมวลผลเชิงเลข

มีงานหลายอย่างที่ได้นำเอาการประมวลผลสัญญาณดิจิทัลไปใช้งาน เราคงจะยกตัวอย่างได้แค่ส่วนหนึ่งของมัน ซึ่งได้แก่

2.3.1 การประมวลผลเสียง เช่น การเข้ารหัสเสียง, การรู้จำเสียง, ซาวด์เอฟเฟกต์, การผสมเสียง, การกรองสัญญาณรบกวน, Synthesizer เป็นต้น

2.3.2 ในระบบสื่อสาร ได้แก่ มอดูเลชัน / ดิมอดูเลชัน, แฟกซ์, คอมพิวเตอร์แบบโยกย้าย, ระบบโทรศัพท์เคลื่อนที่, วงจรกรองความถี่แอ็คทีฟ, การเชื่อมต่อระบบเครือข่าย, เรดาร์, โซนาร์ และ GPS เป็นต้น

2.3.3 ในวงการแพทย์ ได้แก่ Biomedical, Bio - Signal Analysis / Processing, เครื่องช่วยได้ยิน (Hearing Aid)

2.3.4 การประมวลผลแบบหลายมิติ สามารถประยุกต์ใช้ในการประมวลผลภาพ, วีดีโอ, ภาพสามมิติ ได้

2.3.5 ในการวิเคราะห์สถิติและทำนายค่าในอนาคต

2.4 ข้อดีของการใช้การประมวลผลสัญญาณดิจิทัล

ข้อดีของการใช้การประมวลผลสัญญาณดิจิทัล ที่เหนือกว่าการใช้วงจรในระบบแอนะ

ล็อก มีดังนี้

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการโปรแกรมได้ทำให้ง่ายต่อการออกแบบเปลี่ยนแปลงแก้ไขและทดสอบสำหรับแอนะล็อก ถ้าต้องการเปลี่ยนคุณสมบัติอะไรบางอย่างอาจ หมายถึง การที่ต้องออกแบบวงจรใหม่เลย

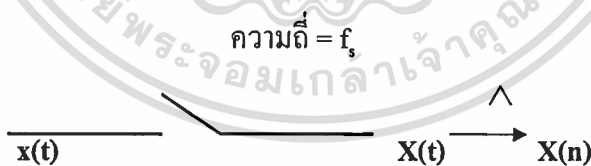
ความถูกต้องแม่นยำที่ดีกว่าความถูกต้องของการประมวลผลสัญญาณดิจิทัล ขึ้นอยู่กับจำนวนบิตที่ใช้แทนสัญญาณและพารามิเตอร์ต่าง ๆ ซึ่งมีความยืดหยุ่นและควบคุมได้ง่าย คือ ในงานที่เราต้องการความแม่นยำสูง เราก็จะใช้จำนวนบิตที่มากขึ้น อีกทั้งในช่วงของการออกแบบ การจำลองระบบที่ออกแบบในคอมพิวเตอร์ จะให้ผลที่ตรงกับความเป็นจริงเมื่อนำไปสร้างเป็นวงจรจริงมีเสถียรภาพที่ไม่ขึ้นกับเวลาและอุณหภูมิ

เกี่ยวข้องกับโดยตรงกับเทคโนโลยีคอมพิวเตอร์และ VLSI ซึ่งกำลังมีความเจริญก้าวหน้าอย่างรวดเร็ว ดังนั้น DSP chip จึงมีความก้าวหน้าในทำนองเดียวกัน ทั้งความเร็วที่สูงขึ้นและราคาก็ถูกลง เรามักพบว่า การประยุกต์ใช้การประมวลผลสัญญาณดิจิทัลในงานหลาย ๆ อย่าง มีต้นทุนที่ต่ำกว่าวงจรแบบแอนะล็อกเดิม

2.5 การสุ่มสัญญาณและการสร้างสัญญาณขึ้น

2.5.1 การสุ่มสัญญาณ

ในภาพรวมอย่างง่าย ๆ การสุ่มสัญญาณ คือ การนำเอาสัญญาณเข้าแบบต่อเนื่องมาผ่านสวิตช์อุดมคติที่มีการตัดต่อด้วยความถี่คงที่ คือ f_s หรือมีคาบเท่ากับ $T = 1/f_s$ ดังรูป 2.1



รูปที่ 2.1 การสุ่มด้วยสวิตช์อุดมคติ

ในแง่คณิตศาสตร์เราอาจมองว่าการสุ่มสัญญาณ คือ การนำเอาสัญญาณมาเข้ามาคูณเข้ากับสัญญาณอิมพัลส์ที่มีคาบเท่ากับ T สมมติว่าสัญญาณขาเข้าคือ $x(t)$ และสัญญาณอิมพัลส์ คือ $\delta(t)$ เรานิยามสัญญาณอิมพัลส์ จุดเดี่ยวนี้นว่า

$$\delta(t) = \begin{cases} 1 & , t = 0 \\ 0 & , t = \text{ค่าอื่นๆ} \end{cases} \quad (2.1)$$

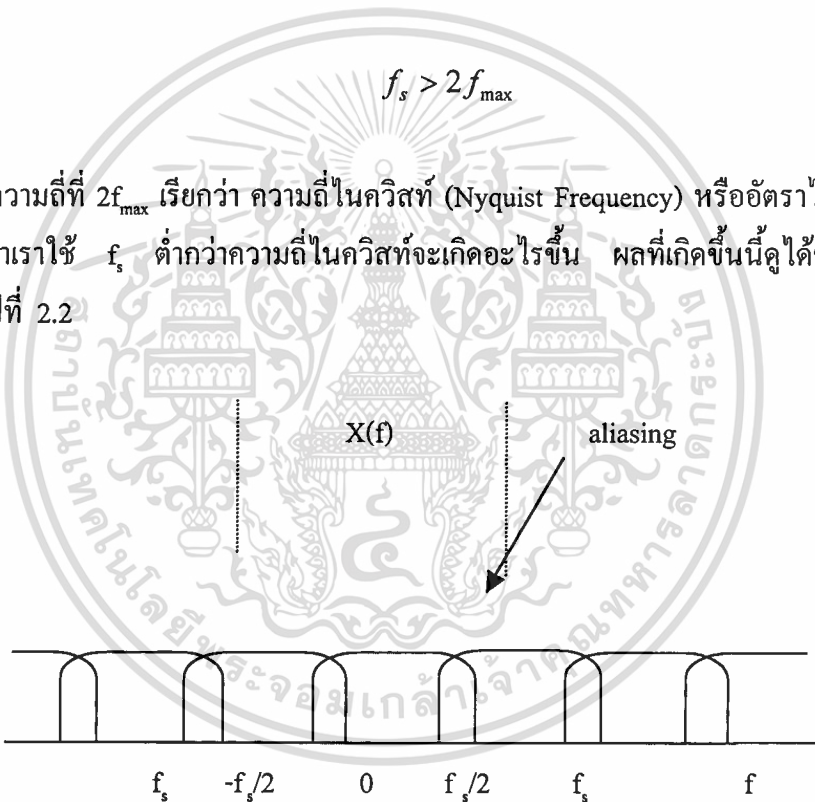
ถ้าให้ $s(t)$ เป็นสัญญาณอิมพัลส์ หลาย ๆ จุดที่มีคาบคงที่เท่ากับ T จะได้ว่า $s(t)$ เป็นผลรวมของสัญญาณอิมพัลส์ ที่จุด $\dots, -2T, 0, T, 2T, \dots$

2.6 ทฤษฎีการสุ่มสัญญาณ

ทฤษฎีการสุ่มสัญญาณระบุว่าถ้าสัญญาณที่ต้องการสุ่มมีความถี่สูงสุดที่ f_{\max} ความถี่ในการสุ่มต้องมีค่ามากกว่าสองเท่าของความถี่สูงสุด หรือ

$$f_s > 2f_{\max} \quad (2.2)$$

ซึ่งความถี่ที่ $2f_{\max}$ เรียกว่า ความถี่ไนควิสต์ (Nyquist Frequency) หรืออัตราไนควิสต์ ลองพิจารณาว่าถ้าเราใช้ f_s ต่ำกว่าความถี่ไนควิสต์จะเกิดอะไรขึ้น ผลที่เกิดขึ้นนี้ดูได้ชัดเจนในภาคความถี่ในรูปที่ 2.2



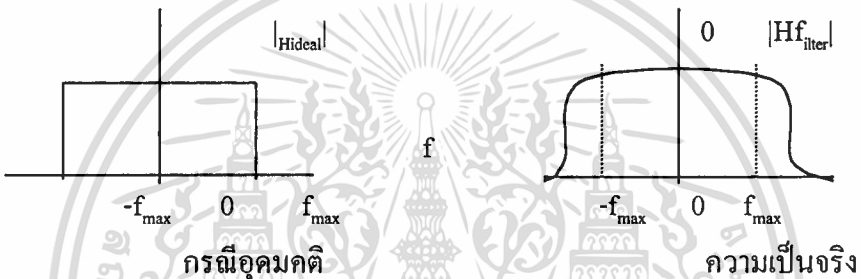
รูปที่ 2.2 สัญญาณที่เกิดขึ้นถ้าใช้ f_s ต่ำกว่า $2f_{\max}$

เมื่อ $f_s < 2f_{\max}$ จะเห็นได้ว่าสำเนาของ $X(f)$ ที่เกิดขึ้นมีช่วงของความถี่ที่ซ้อนทับกันและเรียกส่วนที่ซ้อนทับกันนี้ว่า ช่วงความถี่ซ้อนทับ (Aliasing) ซึ่งช่วงความถี่ซ้อนทับนี้เป็นผลร้ายอย่างยิ่งกับระบบประมวลผลสัญญาณเนื่องจากการประมวลผลสัญญาณเราจะสนใจความถี่ในช่วง $-f_s/2$ จนถึง $f_s/2$ (เรียกว่าช่วงไนควิสต์) ซึ่งช่วงความถี่ซ้อนทับจะซ้อนในส่วนปลายของความถี่ในช่วงนี้ นั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะทำให้สัญญาณขาเข้าที่สุ่มได้มีความผิดเพี้ยนไปก่อนที่จะเข้าไปในส่วนของวงจรประมวลผลเสียอีก

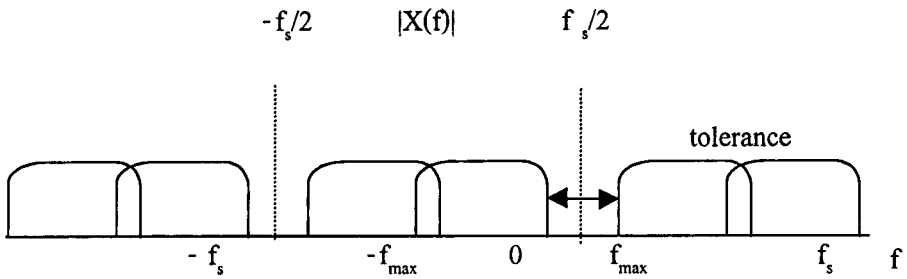
การใช้ตัวกรองเพื่อป้องกันช่วงความถี่ซ้อนทับโดยทั่วไปสัญญาณมักจะไม่มีควมถี่จำกัดอยู่ที่ f_{\max} เพราะอาจมีสัญญาณรบกวนความถี่สูงปนา มา จึงควรที่จะต้องใช้ตัวกรองผ่านความถี่ต่ำ (แบบแอนะล็อก) เพื่อจำกัดความถี่ของสัญญาณให้อยู่ในช่วงที่สนใจเท่านั้น (ต่ำกว่า f_{\max}) นั่นคือ เราต้องการตัวกรองผ่านความถี่ต่ำที่มีความถี่คutoff ที่ f_{\max} ดังในรูปซ้ายมือของรูปที่ 2.3 ซึ่งแสดงผลตอบสนองความถี่ของตัวกรองช่วงความถี่ซ้อนทับแบบอุดมคติ



รูปที่ 2.3 ผลตอบสนองความถี่ของตัวกรองแบบช่วงความถี่ซ้อนทับ

อย่างไรก็ตามในชีวิตจริงตัวกรองแบบอุดมคติที่ต้องการไม่สามารถทำได้และถ้าต้องการสร้างตัวกรองใกล้เคียงกับอุดมคติก็จะทำให้ต้นทุนสูง ตัวกรองที่สามารถทำได้ในทางปฏิบัติมีรูปร่างประมาณในรูปขวามือของรูปที่ 2.3 ซึ่งผลจากการที่ตัวกรองแบบช่วงความถี่ซ้อนทับไม่เป็นอุดมคติ จึงทำให้อาจมีช่วงความถี่ซ้อนทับบางส่วนเกิดขึ้นได้ถ้าเราใช้ความถี่ในการสุ่มเท่ากับ $2f_{\max}$

การสุ่มโดยใช้ความถี่เกินกว่า $2f_{\max}$ มาก ๆ หรือเรียกว่าการทำ โอเวอร์แซมปลิง การสุ่มด้วยความถี่สูงขึ้นเป็นวิธีที่ใช้ป้องกันช่วงความถี่ซ้อนทับเสริมจากวิธีแรกเช่นเดียวกันเราพิจารณาในเชิงความถี่จะเห็นว่า เมื่อ f_s มีค่าสูงขึ้น จะมีช่วงที่ยินยอมให้เกิดช่วงความถี่ซ้อนทับได้กว้างขึ้น และย่านความถี่ที่จะทับกันก็น้อยลงด้วยการสุ่มด้วยความถี่สูงจึงทำให้เราสามารถในตัวกรองแบบช่วงความถี่ซ้อนทับที่ไม่ต้องมีคุณสมบัติดีมากนักได้ในทางปฏิบัติจึงมักใช้ f_s ดังรูปที่ 2.4



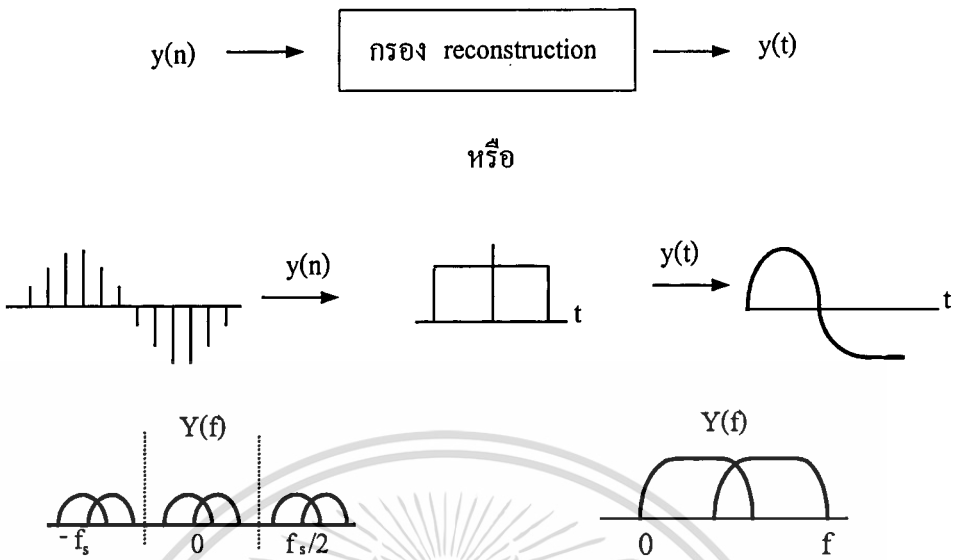
รูปที่ 2.4 การสุ่มด้วยความถี่สูงกว่า $2f_{\max}$ มาก ๆ

f_s สูงสุดที่เราจะสุ่มได้โดยทั่วไปก็ขึ้นอยู่กับขีดจำกัดของวงจร แปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลและวงจรประมวลผล ความถี่ที่สูงขึ้นก็หมายถึงต้องใช้วงจรที่แพงขึ้นในระบบที่ต้องการประมวลผลแบบตามเวลาจริง ช่วงเวลาที่ใช้ประมวลผลของแต่ละจุดของสัญญาณที่สุ่มมาต้องมีค่าน้อยกว่าคาบของการสุ่มไม่พยายามให้ช่วงความถี่ซ้อนทับเกิดขึ้นหรือเกิดน้อยที่สุดซึ่งกระทำได้สองวิธี คือ

$$T_{\text{proc}} < T \quad (2.3)$$

2.7 การสร้างสัญญาณกลับคืน

การสร้างสัญญาณกลับคืนทำได้โดยผ่านสัญญาณแบบไม่ต่อเนื่องเข้าไปยังตัวกรองแบบผ่านความถี่ต่ำที่มีความถี่คัตออฟที่ $f_s/2$ ตัวกรองนี้บางครั้งเรียกว่า ตัวกรองรีคอนสตรัคชัน ผลที่ได้ก็คือเราจะได้สัญญาณแอนะล็อกมีช่วงความถี่ในช่วงระหว่าง $-f_s/2$ และ $f_s/2$ ออกมา ซึ่งจะเป็นสัญญาณขาออกของระบบวงจรรีคอนสตรัคชัน ในทางปฏิบัติอาจใช้วงจรโพล เพื่อสร้างสัญญาณที่มีลักษณะเป็นขั้นบันไดจากสัญญาณดิจิทัลก่อน แล้วจึงผ่านไปสู่วงจรกรองความถี่ดังในรูปที่ 2.5



รูปที่ 2.5 สัญญาณขาเข้าและขาออกของตัวกรองรีคอนสตรัคชัน

เช่นเดียวกันกับตัวกรองแบบช่วงความถี่ซ้อนทับคือเราไม่สามารถทำตัวกรองอุดมคติสำหรับรีคอนสตรัคชันได้ ซึ่งผลข้างเคียงก็คือจะทำให้สัญญาณที่อยู่ในช่วงความถี่สูงกว่า $f_s/2$ หลุดลอดออกมาที่สัญญาณขาออกด้วย ซึ่งก็จะทำให้สัญญาณผิดเพี้ยนไป อย่างไรก็ตาม การที่เราใช้ความถี่ในการสุ่มสูง ๆ หรือโอเวอร์แซมปลิงเพื่อให้มีระยะห่างระหว่างสำเนาในเชิงความถี่ ของสัญญาณมากขึ้นจะช่วยแก้ปัญหานี้ได้เช่นเดียวกันกับการป้องกันช่วงความถี่ซ้อนทับที่ได้กล่าวมาแล้ว

ทฤษฎีการสุ่มสัญญาณบอกไว้ว่าถ้าเราสุ่มสัญญาณแอนะล็อกด้วยความถี่ $f_s/2f_{\max}$ เราจะสามารถสร้างสัญญาณแอนะล็อกขึ้นมาได้โดยสมบูรณ์ นั่นคือ สัญญาณไม่ต่อเนื่องที่เกิดจากการสุ่มเป็นตัวแทนที่สมบูรณ์ของสัญญาณต้นฉบับ ซึ่งจะบรรจุข้อมูลทั้งหมดของสัญญาณแอนะล็อกไว้เงื่อนไขนี้เป็นจริง 100% เมื่อตัวกรองแบบช่วงความถี่ซ้อนทับและตัวกรองสำหรับรีคอนสตรัคชันเป็นแบบอุดมคติ

2.8 การแบ่งขั้นสัญญาณควอนไทซ์เซชัน

การแบ่งขั้นสัญญาณเป็นกระบวนการที่แทนค่าสัญญาณที่ถูกสุ่มแล้วด้วยเลขฐานสองที่มี

เอกสาร จำนวนบิตจำกัดวงการใช้จำนวนบิตแทนสัญญาณยิ่งมากเท่าไรก็จะได้จำนวนขั้นที่ละเอียดมากขึ้นค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และแทนสัญญาณจริงได้ถูกต้องมากขึ้น ถ้าเราให้ B เป็นจำนวนบิต, R เป็นช่วงขนาดของสัญญาณที่ต้องการแทน, และ Q เป็นความกว้างของขั้น จะได้ว่า

สัญญาณก่อนที่จะแบ่งขั้นจะต้องมีค่าอยู่ในช่วง $-R/2$ ถึง $R/2$ ถ้าเราพิจารณาตัวอย่างสัญญาณที่จุดหนึ่ง ๆ คือ $x(n)$ ซึ่งหลังจากถูกแบ่งขั้นแล้วจะมีค่าเปลี่ยนไปเพื่อให้ตรงกับขั้นที่แทนค่าได้สมมติว่าได้ค่าใหม่เป็น $x_q(n)$ ถ้าเราใช้กฎเกณฑ์การปัดแบบราวคี่ดิง (Rounding) จะได้ว่า $x(n)$ ที่มีค่าอยู่ระหว่าง $x_q(n) - Q/2$ ถึง $x_q(n) + Q/2$ จะถูกปัดเป็นค่า $x_q(n)$ ซึ่งก็จะทำให้ผลต่างระหว่าง $x_q(n)$ และ $x(n)$ มีค่าไม่เกิน $Q/2$

$$-Q/2 < x_q(n) - x(n) < Q/2 \quad (2.4)$$

เรานิยาม $e(n)$ ว่าเป็นฟังก์ชันของการผิดพลาดที่เกิดขึ้นจากการแบ่งขั้นสัญญาณ หรืออาจคิดได้ว่า $e(n)$ คือสัญญาณรบกวนที่เกิดจากการแบ่งขั้นสัญญาณควอนไทซ์เซชันจะได้ว่า

$$e(n) = x_q(n) - x(n) \text{ และ } -Q/2 < e(n) < Q/2 \quad (2.5)$$

ฟังก์ชัน $e(n)$ นี้เป็นฟังก์ชันแรมคอมชนิดที่มีการกระจายตัวสม่ำเสมอในช่วงของค่าระหว่าง $-Q/2$ ถึง $Q/2$ หรือมีฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability Density Function) เป็น

$$P_e = \begin{cases} 1/Q & -Q/2 < e < Q/2 \\ 0 & \text{, } e = \text{ค่าอื่นๆ} \end{cases} \quad (2.6)$$

จากทฤษฎีของความน่าจะเป็นเราจะได้ว่าค่าเฉลี่ยของ e มีค่าเท่ากับ 0 และค่าเฉลี่ยของกำลังสองของ e มีค่าเท่ากับ $Q^2/12$ ซึ่งค่านี้มีความหมายเป็นค่ากำลังเฉลี่ย นั่นเองเราใช้สัญลักษณ์ P_e แทนค่ากำลังเฉลี่ยของสัญญาณรบกวนจากการแบ่งขั้น จะได้ว่า

$$P_e = \int_{-Q/2}^{Q/2} e^2 p(e) * de = Q^2 / 12 \quad (2.7)$$

ถ้าเราทราบว่ากำลังเฉลี่ยของสัญญาณมีค่าเท่ากับ P_s เราสามารถหาค่า SNR ที่เนื่องมาจากสัญญาณรบกวนของการแบ่งขั้นสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$SNR = P_s / P_e$$

หรือ $|SNR (db) = P_s (db) - P_e (db)$ (2.8)

2.9 ระบบแบบไม่ต่อเนื่อง

ในหัวข้อนี้จะพูดถึงระบบแบบไม่ต่อเนื่อง ซึ่งระบบการประมวลผลสัญญาณดิจิทัลที่จะศึกษาต่อไปก็เป็นระบบแบบไม่ต่อเนื่องนี้ จึงมีความจำเป็นที่จะต้องเข้าใจเนื้อหาส่วนนี้ให้ดี ถ้าใครได้ศึกษาระบบแบบต่อเนื่องมา ก็จะพบว่าระบบทั้งสองมีลักษณะแนวทางในการคิดและการวิเคราะห์ที่คล้ายคลึงกัน ไม่ว่าจะเป็นเรื่องของความเป็นเชิงเส้นและไม่แปรตามเวลา, การหาผลตอบ, การแปลงระหว่างสัญญาณ ในเชิงเวลากับความถี่, เสถียรภาพและอื่น ๆ

2.10 ความเป็นเชิงเส้นและไม่แปรตามเวลา

ระบบที่น่าจะสนใจเป็นพิเศษในเนื้อหาเบื้องต้นของการประมวลผลสัญญาณ คือ ระบบแบบเชิงเส้นและไม่แปรตามเวลา ก่อนที่จะพูดถึงความสำคัญของระบบแบบนี้ มาดูวิธีคิดก่อนว่าระบบแบบใดที่มีคุณสมบัติดังกล่าว

สมมติว่าระบบแบบไม่ต่อเนื่องที่สนใจระบบหนึ่ง เมื่อป้อนสัญญาณขาเข้า $x_2(n)$ ทำให้เกิดสัญญาณขาออก $y_2(n)$

ถ้าให้ $x(n)$ เป็นสัญญาณขาเข้าที่เกิดจากการคำนวณแบบเชิงเส้นระหว่าง $x_1(n)$ และ $x_2(n)$ นั่นคือ

$$x(n) = a_1 x_1(n) + a_2 x_2(n) \quad (2.9)$$

กล่าววาระบบนี้เป็นระบบแบบเชิงเส้นถ้า $x(n)$ ทำให้เกิดสัญญาณขาออก $y(n)$ โดยที่

$$y(n) = a_1 y_1(n) + a_2 y_2(n) \quad (2.10)$$

นั่นคือ จะต้องได้ $y(n)$ เป็นการคำนวณแบบเชิงเส้นระหว่าง $y_1(n)$ และ $y_2(n)$ โดยที่มีสัมประสิทธิ์ที่ใช้คูณ (a_1 และ a_2) ตัวเดียวกันกับสัญญาณขาเข้าระบบเชิงเส้นบอก

ถ้าคุณสัญญาณขาเข้าด้วยค่าสัมประสิทธิ์ค่าหนึ่งสัญญาณขาออกก็เปลี่ยนไปด้วยตัวคูณเดียว

เอกสารกัน เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าสัญญาณขาเข้าเป็นผลรวมของสัญญาณหลาย ๆ สัญญาณ สัญญาณขาออกก็จะเป็นผลรวมของสัญญาณขาออก ที่เกิดจากสัญญาณขาเข้าแต่ละตัวมารวมกันคุณสมบัติของระบบที่เป็นเชิงเส้นนี้บางทีถูกเรียกว่าคุณสมบัติซูเปอร์โพสิชันสำหรับคุณสมบัติความไม่แปรตามเวลาของระบบไม่ต่อเนื่องมีเงื่อนไขว่าถ้าสัญญาณขาเข้า $x(n)$ ทำให้เกิดสัญญาณขาออก $y(n)$ แล้ว สัญญาณขาเข้า $x(n-k)$ จะทำให้เกิดสัญญาณขาออก $y(n-k)$ โดยที่ k เป็นจำนวนเต็มใด ๆ

ระบบแบบไม่แปรตามเวลาบอกเราว่า ไม่ว่าเราจะใส่สัญญาณขาเข้าที่เวลาใด เราจะได้สัญญาณขาออกที่เหมือนเดิม นั่นก็คือระบบของเราไม่มีค่าพารามิเตอร์ใด ๆ ที่แปรตามเวลา หรือแปรตามสถานะแวดล้อมใด ๆ ที่แปรตามเวลา

2.11 ผลตอบสนองจากสัญญาณอิมพัลส์

ระบบเชิงเส้นไม่แปรตามเวลามีคุณลักษณะพิเศษ คือ สามารถระบุคุณลักษณะของระบบได้โดยสมบูรณ์ด้วยผลตอบสนองจากสัญญาณอิมพัลส์ซึ่งเขียนสัญลักษณ์แทนว่า $h(n)$ หมายถึงสัญญาณขาออกของระบบเมื่อสัญญาณขาเข้าเป็นสัญญาณอิมพัลส์ $\delta(n)$ ซึ่งมีค่าเป็น 1 ที่ $n = 0$ ดังในรูปที่ 2.6



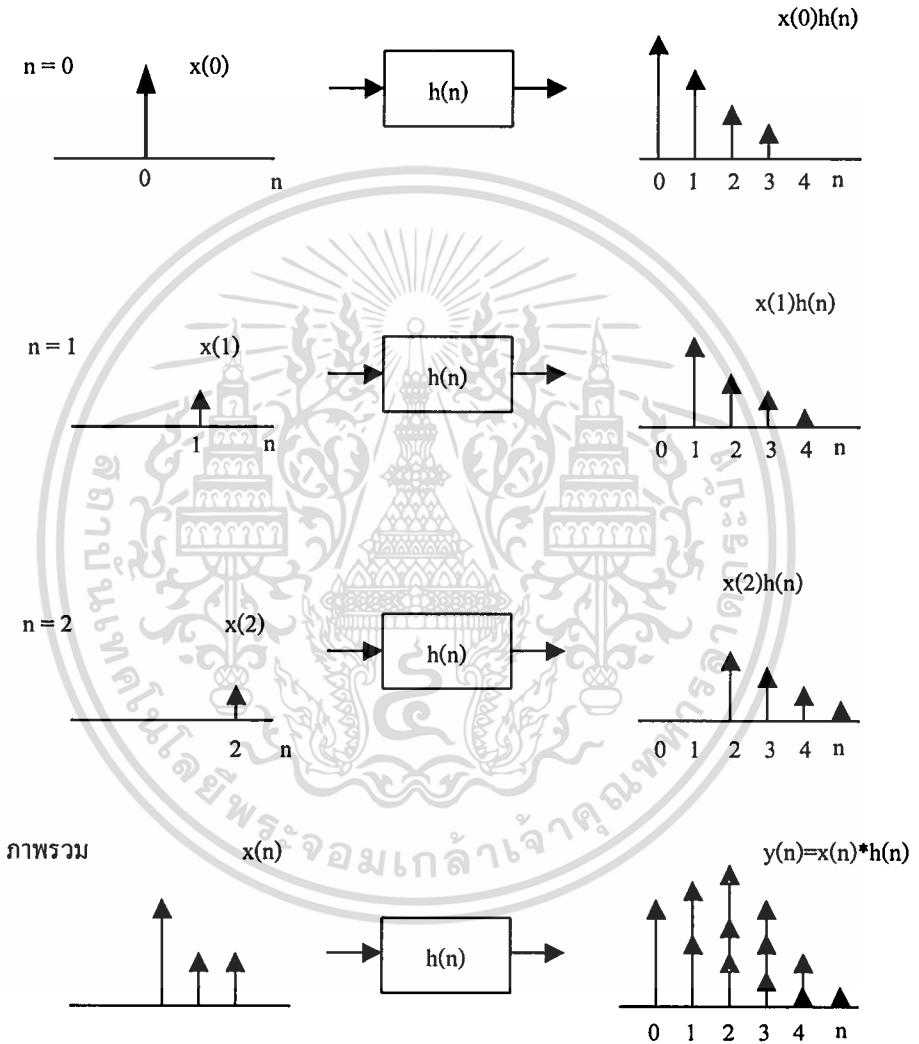
รูปที่ 2.6 ผลตอบสนองจากสัญญาณอิมพัลส์

สังเกตว่าเมื่อรู้ผลตอบสนองจากสัญญาณอิมพัลส์สามารถหาผลตอบของระบบเมื่อสัญญาณขาเข้าเป็นสัญญาณใด ๆ ได้อย่างไร สมมติว่าสัญญาณขาเข้าเป็นสัญญาณไม่ต่อเนื่องใด ๆ $x(n)$ เพื่อให้ง่ายต่อการวิเคราะห์จะสมมติว่า x เริ่มมีค่าเมื่อ $n = 0$ เราสามารถเขียน $x(n)$ ได้เป็นผลรวมของสัญญาณอิมพัลส์ที่มีขนาดเท่ากับ $x(n)$ ที่ค่า $n = 0, 1, 2, \dots$ นั่นคือ

$$x(n) = x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยคุณสมบัติความไม่แปรตามเวลาของระบบ เมื่อ $\delta(n)$ ทำให้เกิดผลตอบ $h(n)$ ก็จะได้ว่า $\delta(n-1)$ ทำให้เกิดผลตอบ $h(n-1)$ และ $\delta(n-2)$ ทำให้เกิดผลตอบ $h(n-2)$ เป็นเช่นนี้ไปเรื่อย ๆ ดังรูปที่ 2.7



รูปที่ 2.7 การหาผลตอบสนองของระบบโดยการประสาน

อาศัยคุณสมบัติความเป็นเชิงเส้นของระบบก็จะได้ว่า $x(0)\delta(n)$ ทำให้เกิดผลตอบ $x(0)h(n)$ และ $x(1)\delta(n-1)$ ทำให้เกิดผลตอบ $x(1)h(n-1)$ เป็นเช่นนี้ไปเรื่อย ๆ ดังแสดงในรูป ดังนั้นผลตอบ $y(n)$ ที่เกิดจาก $x(n)$ ก็จะมาจากผลรวมของผลรวมของผลตอบแต่ละตัว นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(n) = x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) = \dots$$

$$y(n) = \sum_{m=0}^{\infty} x(m)h(n-m) \quad (2.12)$$

ถ้าไม่จำกัดว่า $x(n)$ เริ่มมีค่าที่ $n = 0$ ก็จะได้เป็นสมการทั่วไปของผลตอบของระบบ เป็น

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \quad (2.13)$$

$y(n)$ ที่ได้นี้ก็คือ ผลคูณการประสาน (Convolution) แบบไม่ต่อเนื่องนั่นเองนิยามสัญลักษณ์ * แทนการกระทำนี้ และจะสามารถเขียนสมการของ $y(n)$ ได้ใหม่เป็น

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(n-(n-k)) \quad (2.15)$$

ถ้าแทน m ด้วย $n-k$ ในสมการของการประสานจะได้ว่า

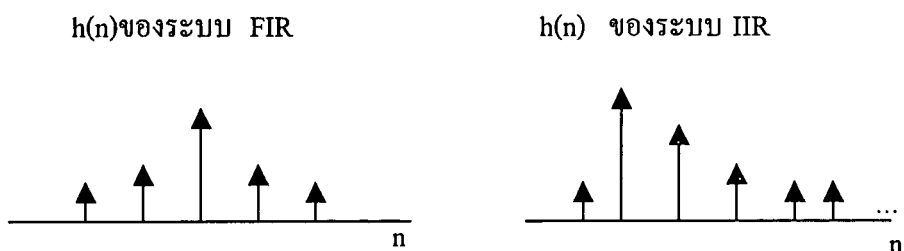
นี่ก็คืออีกรูปหนึ่งในการประสานซึ่งจะพบว่ารูปแบบหลังนี้ (ใช้ $n-k$ เป็น Index ของ x) สามารถนำมาใช้งานได้สะดวกกว่าในการสลับตัวแปรระหว่าง m กับ k เพราะ m และ k ในสมการข้างต้นเป็นเพียงดัชนี (Index) เพื่อให้เกิดการบวกกันใน \sum เท่านั้นแต่ดัชนีของเวลาจริง ๆ คือค่า n

2.12 ตัวกรองแบบจำกัดผลตอบสนองความถี่และตัวกรองแบบไม่จำกัดผลตอบสนองความถี่

ตัวกรองแบบผลตอบสนองความถี่และตัวกรองแบบไม่จำกัดผลตอบสนองความถี่จะอยู่ในรูปย่อ ดังนี้คือ FIR และ IIR ซึ่ง FIR มาจาก Finite Impulse Response เป็นตัวกรองชนิดที่มีความยาวของ $h(n)$ จำกัด ถ้า $h(n)$ ของระบบมีความยาวจำกัดเท่ากับ N จุด เรากล่าวว่า ระบบนี้เป็นตัวกรองแบบ FIR ที่มีคำสั่งเท่ากับ $N-1$ ถ้าสมมติว่า $h(n)$ มีค่าแรกที่ $n = 0$ จะได้ว่า $h(n) = 0$ ที่ $n < 0$ และ $n > N-1$

สำหรับ IIR มาจาก Infinite Impulse Response เป็นตัวกรองชนิดที่มีความยาวของ $h(n)$ ไม่จำกัด ซึ่ง $h(n)$ อาจมีความยาวเป็นอนันต์ในช่วงที่ n เป็นค่าบวก หรือลบได้ ดังในรูปที่ $h(n)$ มีค่าไม่เป็นศูนย์ (แต่อาจเข้าสู่ศูนย์) ที่ $n > 0$ ดังแสดงในรูปที่ 2.8

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ผลตอบสนองจากสัญญาณอิมพัลส์ของตัวกรองแบบจำกัดผลตอบสนองความถี่และตัวกรองแบบไม่จำกัดผลตอบสนองความถี่

2.13 สมการผลต่าง

ระบบแบบไม่ต่อเนื่อง นอกจากสามารถถูกกำหนดลักษณะได้ด้วย $h(n)$ ยังสามารถถูกกำหนดได้ด้วยสมการผลต่าง ซึ่งสมการผลต่างในที่นี้ก็คือ สมการที่บ่งบอกความสัมพันธ์ระหว่างสัญญาณขาเข้าของระบบ คือ $x(n)$ และสัญญาณขาออก หรือผลตอบ คือ $y(n)$

2.14 เสถียรภาพ

ระบบที่มีเสถียรภาพมีเงื่อนไขว่าสัญญาณขาเข้าที่มีขอบเขตของขนาดจำกัดทำให้เกิดสัญญาณขาออกที่มีขอบเขตจำกัด หรือ เขียนเป็นสัญลักษณ์ว่า

ถ้า $|x(n)| < A$ โดยที่ A เป็นจำนวนจริงบวกใด ๆ ที่น้อยกว่าอนันต์แล้ว

จะได้ $|y(n)| < B$ โดยที่ B เป็นจำนวนจริงบวกใด ๆ ที่น้อยกว่าอนันต์

เงื่อนไขนี้บางครั้งเรียกว่าเสถียรภาพแบบ เบาวนด์-อินพุต/เบาวนด์-เอาต์พุต (Bounded Input / Bounded Output) หรือ BIBO เงื่อนไขดังกล่าว สามารถแปลงเป็นเงื่อนไขทางคณิตศาสตร์ที่ใช้บังคับ $h(n)$ ได้ โดยมีเงื่อนไขว่าระบบที่เสถียรจะต้องมี $h(n)$ ตามเงื่อนไขนี้

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \quad (2.16)$$

เงื่อนไขทั้งสองสมมูลกันแต่จะสามารถอธิบายได้ง่ายกว่าโดยใช้เงื่อนไขที่สอง ซึ่งก็บอกว่า $h(n)$ จะต้องเข้าสู่ศูนย์เมื่อ n มีค่ามากขึ้น หรือน้อยลงในระบบแบบไม่เป็นเหตุกภาพ (Anticausal) ในทางปฏิบัติระบบส่วนใหญ่ที่เราใช้เป็นระบบที่เสถียรแต่ระบบที่ไม่เสถียรก็มีการนำมาใช้ประโยชน์ได้ เช่น ในวงจรสร้างสัญญาณเมื่อสัญญาณขาเข้าเป็นอิมพัลส์ ที่ $n = 0$ สัญญาณขาออกอาจเป็นสัญญาณที่ต้องการสร้าง เช่น ไซน์เวฟ (Sine wave) ที่มีความยาวไม่จำกัดก็ได้

สำหรับคุณสมบัติการก่อให้เกิดสัญญาณเหตุกภาพ (Causality) เป็นคุณสมบัติที่กำหนดได้กับทั้งสัญญาณและระบบลองมาดูสัญญาณก่อนว่า

สัญญาณ $x(n)$ เป็นสัญญาณแบบการก่อให้เกิดสัญญาณเหตุกภาพ

$$\text{เมื่อ } x(n) = 0 \quad \text{ที่ } n < 0 \quad (2.18)$$

นั่นก็ คือ สัญญาณนี้มีค่าเริ่มต้นที่ $n = 0$ และมีค่าเฉพาะช่วงที่ n เป็นบวกและเรากล่าวว่า

สัญญาณ $x(n)$ เป็นสัญญาณแบบการก่อให้เกิดสัญญาณเมื่อ $x(n) = 0$ ที่ $n \geq 0$

สัญญาณที่มีค่าทั้งส่วนที่ n เป็นบวกและลบ เราเรียกว่า สัญญาณแบบ Two-Sided

เมื่อพิจารณาในแง่ของระบบก็ได้ในทำนองเดียวกัน โดยเราจะพิจารณาจากผลตอบจากสัญญาณอิมพัลส์ของระบบ คือ $h(n)$ ดังนี้

ระบบเป็นแบบเหตุกภาพก็ต่อเมื่อ

$$h(n) = 0 \quad \text{ที่ } n < 0 \quad (2.19)$$

ระบบเป็นแบบไม่เป็นเหตุกภาพก็ต่อเมื่อ

$$h(n) = 0 \quad \text{ที่ } n \geq 0 \quad (2.20)$$

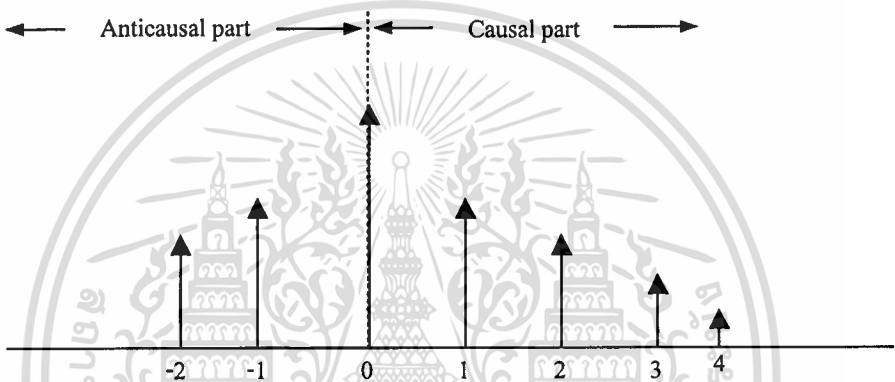
ระบบที่ $h(n)$ มีค่าทั้งฝั่ง n เป็นบวกและลบ ก็เรียกว่า ระบบแบบ Two-sided

จากในรูปที่ 2.9 ซึ่งเห็นได้ว่า $h(-1)$ และ $h(-2)$ มีค่าไม่เท่ากับศูนย์ เมื่อเราพิจารณาจากความจริงว่า $h(n)$ เป็นผลตอบเมื่อสัญญาณขาเข้าเป็นสัญญาณอิมพัลส์ที่เวลา $m = 0$ จะได้ว่าระบบให้ผลตอบบางส่วนก่อนที่สัญญาณขาเข้าจะเกิดขึ้นเสียอีก นั่นคือมีผลตอบที่ไม่เป็นศูนย์ที่เวลา $n = -1$ และ $n = -2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในมุมมองของสัญญาณที่เข้ามาก็กล่าวได้ว่า ระบบไม่เป็นเหตุภาพสามารถสร้างผลตอบสนองล่วงหน้าบางส่วนก่อนที่สัญญาณขาเข้าจริงจะเข้ามา หรือ ในมุมมองของระบบก็กล่าวได้ว่า ระบบที่ไม่เป็นเหตุภาพ มีการนำเอาสัญญาณขาเข้าในอนาคตมาคำนวณหาผลตอบในปัจจุบัน คำกล่าวนี้จะเห็นได้ชัดเจนเมื่อเราเขียนสมการผลต่างของระบบออกมา โดยจะได้สมการผลต่างของระบบนี้ดังแสดงประกอบในรูปที่ 2.9

$$y(n] = h(-2)x(n+2) + h(-1)x(n+1) + h(0)x(n) + h(1)x(n-1) + \dots \quad (2.21)$$



รูปที่ 2.9 ของระบบที่ไม่เป็นเหตุภาพ

2.15 ตัวกรองแบบจำกัดผลตอบสนองสองความถี่

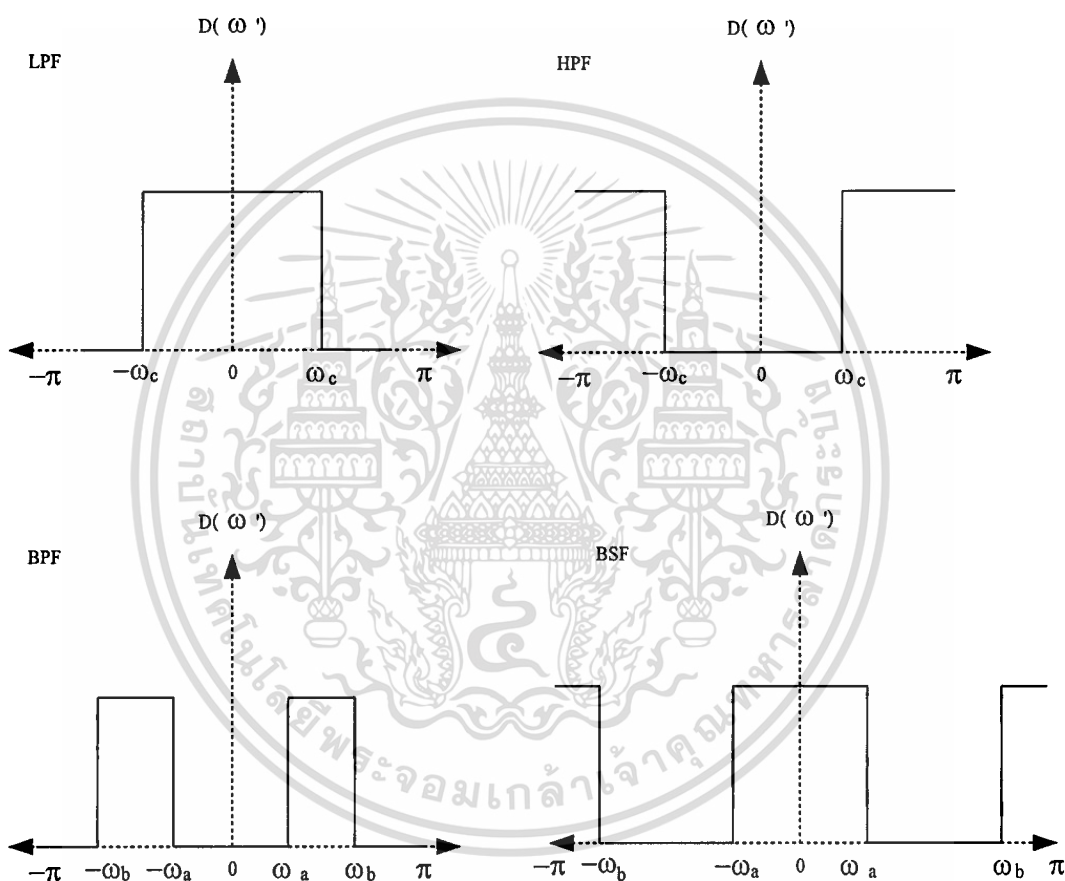
ในการออกแบบตัวกรองแบบจำกัดผลตอบสนองสองความถี่นั้นสิ่งที่ต้องการหา คือ ผลตอบจากสัญญาณอินพุต หรือ $h(n)$ ของระบบ สำหรับตัวกรองแบบจำกัดผลตอบสนองความถี่ที่มี $h(n)$ ยาว N จุด ตัวกรองนี้มีคำสั่งเท่ากับ $n-1$ เหตุผลก็คือ มีการใช้สัญญาณขาเข้าในอดีตย้อนหลังไป $N-1$ จุด หรือ กำลังลบต่ำสุดของ $H(z)$ คือ $-(N-1)$

2.16 การออกแบบโดยวิธีวินโดว์

วิธีวินโดว์สามารถใช้ออกแบบตัวกรองแบบผ่านความถี่ต่ำ, ผ่านความถี่สูง, ช่วงแถบความถี่ผ่าน, และหยุดแถบความถี่ ได้โดยมีความถี่คutoffเป็นค่าใด ๆ เราจะทำการออกแบบโดยคำนึงถึง

เอกสารที่ความถี่ดิจิทัล ω ที่ย่านความถี่ที่สนใจในช่วง $-\pi$ ถึง π หรือ π ในช่วง -1 ถึง 1 ใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มออกแบบโดยใช้ต้นแบบจากตัวกรองอุดมคติ ผลตอบสนองความถี่ของตัวกรองอุดมคติ ทั้งสี่แบบแสดงอยู่ในรูปที่ 2.10 โดยมีความถี่คutoff เท่ากับ ω_c สำหรับแบบผ่านต่ำ และผ่านสูง ส่วนแบบแถบความถี่ผ่านและหยุดแถบความถี่ที่มีความถี่คutoff ของแถบความถี่ที่ ω_a และ ω_b



รูปที่ 2.10 ผลตอบสนองความถี่ของตัวกรองอุดมคติ

ผลตอบจากสัญญาณอิมพัลส์ ของตัวกรองอุดมคติเหล่านี้โดยการแปลง IDTFT สมมติให้ $d(n)$ แทนผลตอบจากสัญญาณอิมพัลส์ และ $D(\omega')$ แทนผลตอบสนองความถี่ สำหรับตัวกรองแบบผ่านต่ำดังในรูปที่ 2.12 จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

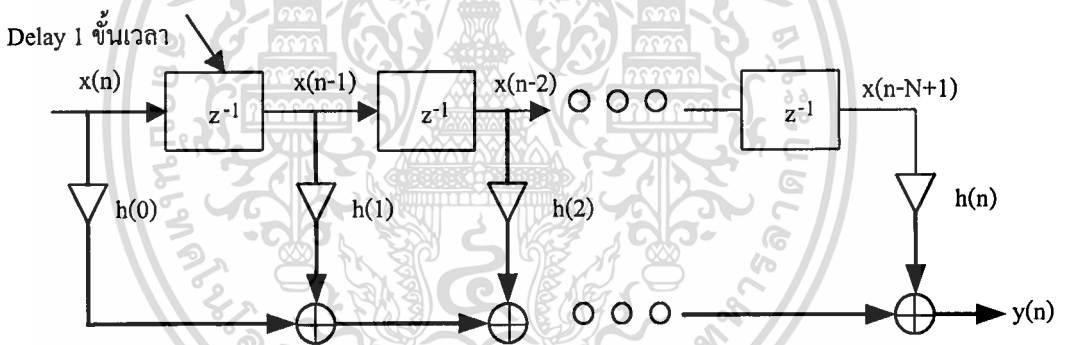
$$d(n) = IDTFT(D)$$
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ $n = 0$ จะได้ว่า $d(n)$ มีค่าเป็นเศษศูนย์ส่วนศูนย์ ซึ่งสามารถหาค่า $d(0)$ ได้โดยใช้ทฤษฎีบทของโลปีตัส จะได้ว่า

ในทำนองเดียวกันสามารถหา $d(n)$ สำหรับตัวกรองแบบอื่น ๆ ได้โดย DTFT ในตารางที่ 2.1 เป็นการสรุปค่า $d(n)$ สำหรับตัวกรองแบบต่าง ๆ เพื่อใช้เป็นตัวอ้างอิงในการออกแบบต่อไป

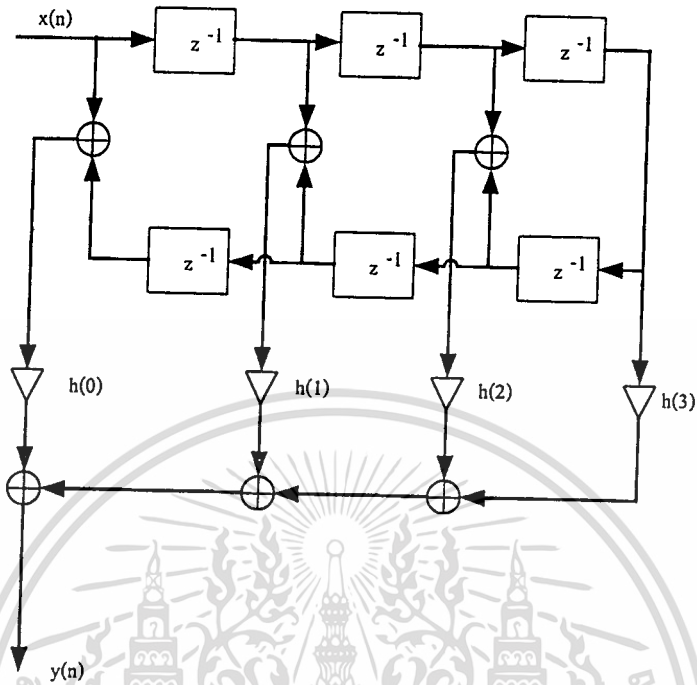
2.17 การสร้างตัวกรองจำกัดผลตอบสนองความถี่

การใช้งานในเวลาจริง (Realization) หมายถึง การนำเอาสิ่งที่ได้ออกแบบแล้วไปประยุกต์เป็นอุปกรณ์ที่ใช้งานได้จริง ๆ ขึ้นมาสำหรับตัวกรองแบบจำกัดผลตอบสนองความถี่เราจะใช้ผลตอบจากสัญญาณอิมพัลส์ หรือ $h(n)$ เพื่อสร้างตัวกรอง ซึ่งกระบวนการของตัวกรองในที่นี้ก็คือ การทำการประสานระหว่าง $h(n)$ และสัญญาณขาเข้า $x(n)$ นั่นเอง ซึ่งสามารถเขียนเป็นแผนภาพได้ดังรูปที่ 2.11



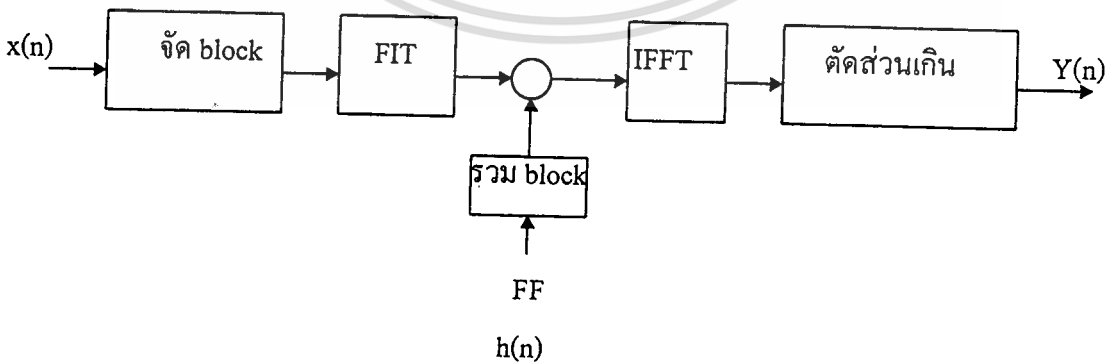
รูปที่ 2.11 แผนภาพการสร้างตัวกรองแบบจำกัดผลตอบสนองความถี่โดยใช้การประสานปกติ

จากแผนภาพนี้สามารถนำไปประยุกต์เขียนเป็นโปรแกรมคอมพิวเตอร์หรือทำเป็นฮาร์ดแวร์พิเศษเพื่อทำตัวกรองแบบจำกัดผลตอบสนองความถี่ได้โดยตรง ในกรณีที่ตัวกรองเป็นแบบเฟสเชิงเส้นซึ่ง หมายถึง $h(n)$ จะมีสมมาตร ณ จุดกึ่งกลางใช้คุณสมบัตินี้ลดโครงสร้างของตัวกรองความถี่แบบจำกัดผลตอบสนองความถี่ ให้เล็กลงได้ในรูปที่ 2.12



รูปที่ 2.12 แผนภาพการสร้างตัวกรองแบบจำกัดผลตอบสนองความถี่เมื่ออาศัยสมมาตร

และโดยอาศัยวิธีการประสานแบบรวดเร็วก็สามารถใช้แทนการการประสานแบบปกติเพื่อใช้สร้างตัวกรองแบบจำกัดผลตอบสนองความถี่ได้ดังในรูปที่ 2.13



รูปที่ 2.13 แผนภาพการสร้างตัวกรองแบบจำกัดโดยวิธีการประสานแบบรวดเร็ว

FPGA จัดเป็นวงจรรวมเฉพาะกิจชนิดหนึ่งที่สามารถโปรแกรมเป็นวงจรถติจิตอลโค ๆ ได้ เช่นเดียวกับ EPLD ต่างกันที่ EPLD โปรแกรมลงบน EPROM ภายใน และสามารถโปรแกรมใหม่

ได้หลังจากที่นำไปลบด้วยแสงอุลตราไวโอเลต แต่สำหรับ FPGA_s จะโปรแกรมลงบนสแตติกแรมภายในด้วยข้อมูลที่อยู่นอก และสามารถโปรแกรมใหม่ได้โดยการรีเซตด้วยสัญญาณไฟฟ้าจากนั้น FPGA_s ยังประหยัดพลังงานและมีความจุจริงที่มีขนาดใหญ่ (จำนวนเกตมาก ๆ) ได้อีกด้วย

2.18 ทฤษฎีและหลักการของ FPGA_s

2.18.1 โครงสร้างภายใน

วงจรรวมชนิดนี้ผลิตโดยบริษัทไซลิงค์ซึ่งเป็นบริษัทที่ร่วมกันทำการค้นคว้ากับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นอะเรย์ที่ประกอบด้วยเกตจำนวน 600-25,000 เกต ในตารางที่ 2.1 การที่ต้องบอกขนาดของวงจรรวมเป็นจำนวนเกตเพราะจะรู้ว่าขนาดของวงจรถูกได้ออกแบบไว้สามารถโปรแกรมลงบนวงจรรวม FPGA_s ได้หรือไม่

ตารางที่ 2.1 การนับเกตของเกตพื้นฐาน

FPGA _s	App.Gate Count	Max I/O	Flip-Flops	RSM bits	Available CLBs
XC2064	1,000	58	122	0	64
XC2018	1,500	74	174	0	100
XC3020/3120	1,800	64	256	0	64
XC3030/3130	2,700	80	360	0	100
XC3042/3142	3,700	96	480	0	144
XC3064/3164	5,500	120	688	0	224
XC30903190	7,500	144	928	0	320
XC3195	9,000	176	1,320	0	484
XC402A	2,000	64	256	2,048	64
XC4003/4003A	3,000	80	360	3,200	100
XC4003H	3,000	160	200	3,200	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ) การนับเกตของเกตพื้นฐาน

XC4004A	4,000	960	480	4,608	144
XC4005/4005A	5,000	122	616	6,272	196
XC4005H	5,000	192	392	6,272	196
XC4006	6,000	128	768	8,192	256
XC4008	8,000	144	936	10,368	324
XC4010	10,000	160	1,120	12,800	400
XC4010D	10,000	160	1,120	0	400
XC4013	13,000	192	1,536	18,432	576
XC4025	25,000	256	2,560	32,768	1,024

(NAIIR2 หมายถึงเกต NAND2 หรือเกต NOR2)

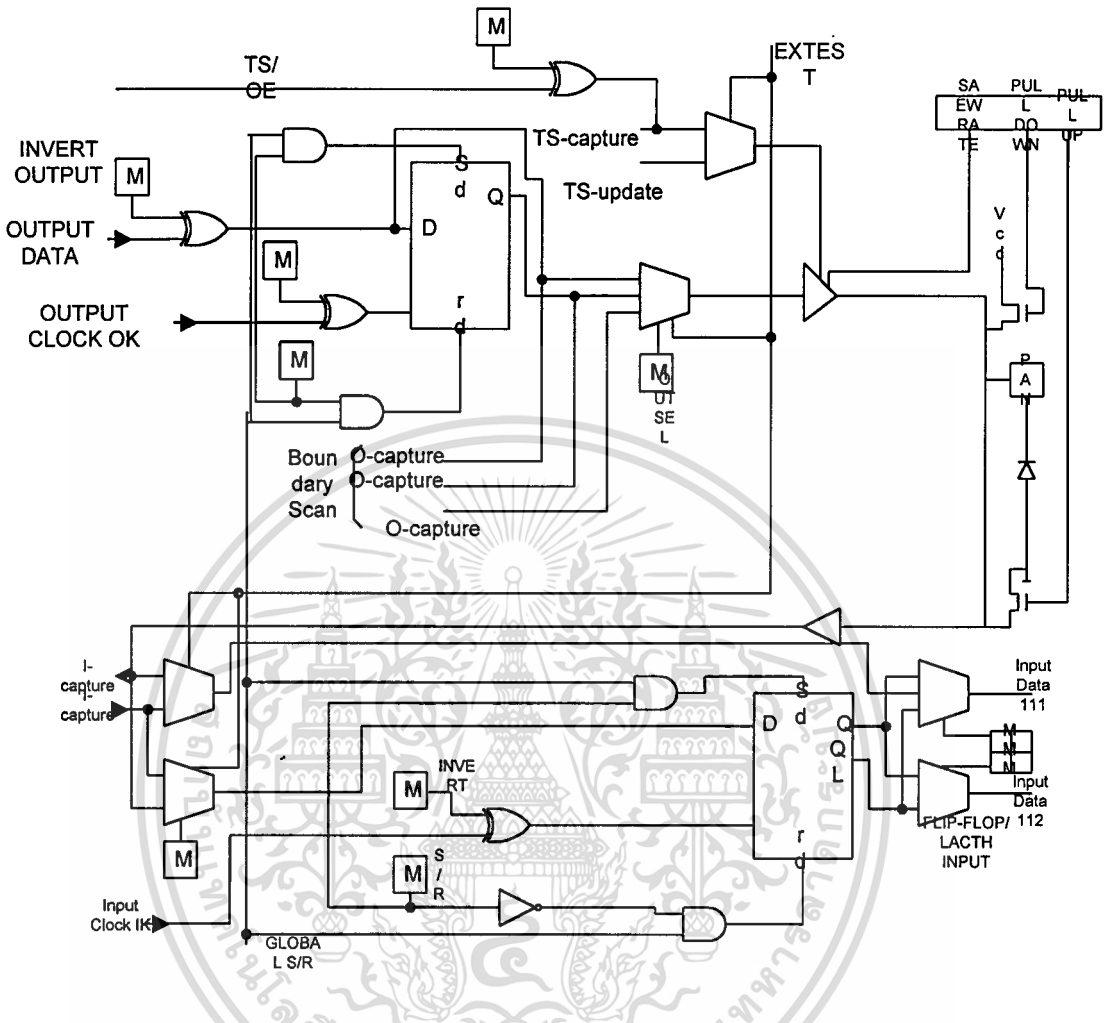
Gate	Equivalent gate count	Gate	Equivalent gate count
INV	1	RS แลทช์	3
NAIIR2	1	D แลทช์	4
NA/NR3	2	D แลทช์ with CLR	5
NAIIR4	2	D แลทช์ with PRE	5
NAIIR6	5	D แลทช์ with PRE/CLR	6
NAIIR8	6	DFF	6
NAIIR9	7	DFF with CLR	7
NAIIR12	8	DFF with PRE	7
NAIIR16	11	DFF with PRE/CLR	8
BUFF	2	JKFF	9
ANIIOR2	2	JKFF with CLR	12
ANIIOR3	2	JKFF with PRE/CLR	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA_s มีโครงสร้างภายในใกล้เคียงกับสถาปัตยกรรมของเกตอะเรย์มาก สามารถโปรแกรมและลอคองค์ประกอบสแตตริกแรม ภายในได้ด้วยใช้กระแสไฟฟ้า ซึ่งทำการลงโปรแกรมได้โดยดึงข้อมูลฐานสิบหกออกมาจากภายนอก เช่น Parallel EPROM หรือ Serial PROM ต่างกับ EPLD, PAL ที่มี EPROM อยู่ในตัวภายใน FPGA_s จะจัดเรียงเป็นลอจิกเกตล้อมรอบภายนอกด้วย อินพุต เอาต์พุต เซลล์ FPGA_s ตัวแรกที่ผลิตโดยไซลิงค์คือ XC2000 (2000 FAMILY) ประกอบด้วย เซลล์เรียงกันเป็นเมตริกเป็นจำนวน 64 เซลล์ หลังจากนั้น ผลิตตระกูล 3000 และ 4000 มีโครงสร้างซับซ้อนขึ้นสามารถบรรจุจำนวนเกตได้สูงและดีขึ้นแต่ละเซลล์เรียกว่า CLB (Configuration Logic Block)

2.18.2 ส่วนประกอบของลอจิกเกต

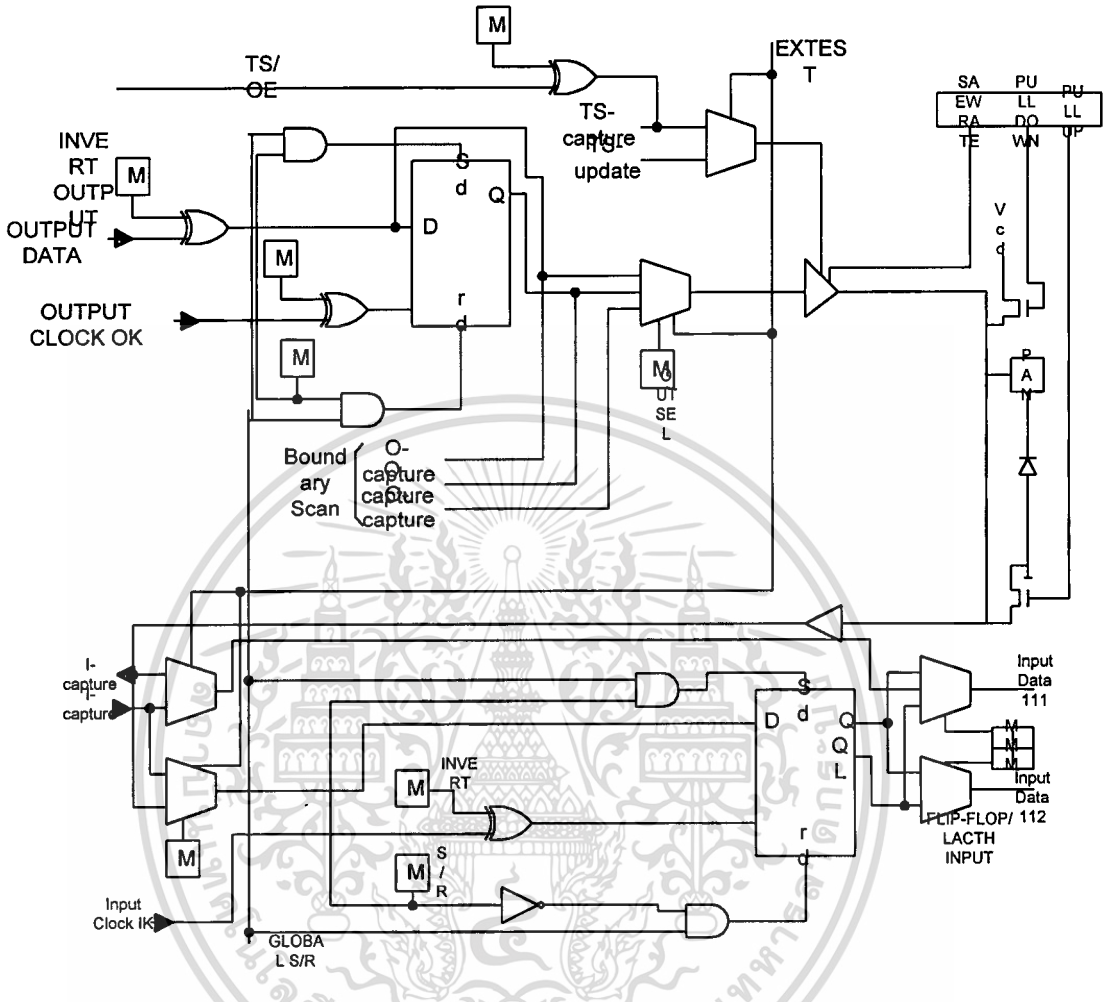
CLB จะจัดเรียงเป็นแบบเมตริกแบบอะเรย์ขนาด $M \times N$ การออกแบบนั้นสามารถทำได้โดยการจัดวาง CLB และต่อเชื่อมขาของ CLB ให้ต่อกันสามารถจัด CLB ให้เชื่อมต่อกันได้โดยการทำด้วยมือ หรือ โปรแกรมที่สนับสนุน FPGA_s ทำได้โดยอัตโนมัติโดยวิธีของมันเอง สำหรับไฟล์ที่ได้จากโปรแกรมเหล่านี้เรียกว่าไฟล์โครงร่าง ซึ่งจะบรรจุโครงร่างภายในของ CLB ตามความเหมาะสมในอีกด้านหนึ่งไฟล์โครงร่างนั้นจะเป็นไฟล์กระแสข้อมูล ซึ่งสามารถใช้โปรแกรมหน่วยความจำภายในของ FPGA_s (Internal FPGA_s Configuration Memory) ได้ สำหรับรูป CLB ของ FPGA_s ตระกูล 4000 ดังรูปที่ 2.14



รูปที่ 2.14 แผนผัง CLB ตระกูล 4000

2.18.3 ส่วนอินพุตและเอาต์พุต

รอบ ๆ นอกของ FPGA_s ประกอบด้วย I/Os ประมาณ 64 ถึง 144 ตัวขึ้นอยู่กับตระกูลของ FPGA_s ซึ่ง I/Os จะเป็นตัวเชื่อมต่อระหว่างภายในและภายนอกของวงจรถลอจิก FPGA_s I/Os จะมีลักษณะ 2 ทิศทาง สามารถโปรแกรมให้เป็นอินพุต หรือเอาต์พุตก็ได้สำหรับรูปที่ 2.13 I/Os ของ FPGA_s ตระกูล 4000



รูปที่ 2.15 แผงผัง IBOs ของตระกูล 4000

2.18.4 รายละเอียดการใช้งาน

FPGA_s สามารถทำงานได้หลายรูปแบบโดยกำหนดได้ที่ขาสัญญาณ M0, M1 และ M2 ดังในตารางที่ 2.2 ในแบบมาสเตอร์พาราแลลรับโปรแกรมคอนฟิกทีละ 1 ไบต์ จากหน่วยความจำภายนอกที่เป็นแบบขนาน โดยสามารถรับ โปรแกรมคอนฟิกจากตำแหน่งต่ำหรือสูงก่อนก็ได้ แบบเฟิร์มแวร์รับโปรแกรมคอนฟิกทีละ 1 ไบต์ จากไมโครโพรเซสเซอร์โดยสามารถตอบโต้กันได้ว่าพร้อมหรือไม่พร้อมที่จะรับข้อมูลต่อไปแบบสเต็ปรับ โปรแกรมคอนฟิกทีละ 1 บิต จากไมโครโพรเซสเซอร์ตามสัญญาณอินพุต CCLK ส่วนในแบบมาสเตอร์ซีเรียลรับโปรแกรมคอนฟิกทีละ 1

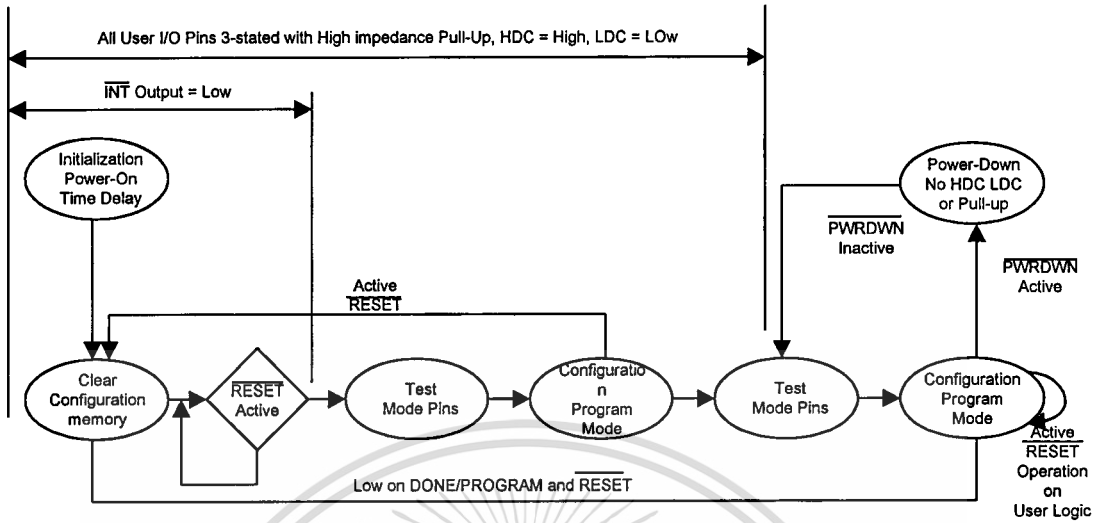
เอกสาร บิต จากหน่วยความจำภายนอกที่เป็นแบบอนุกรม หากท่านนั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 รูปแบบต่าง ๆ ของการกำหนดดองค์ประกอบในการทำงาน

M0	M1	M2	CCLK	MODE	DATA
0	0	0	เอาต์พุต	Master parallel	BIT SERIAL
0	0	1	เอาต์พุต	Master parallel	BYTE WIDE ADDR.=0000 up
0	1	0	-----	Reserved	-----
0	1	1	เอาต์พุต	Master parallel	BYTE WIDE ADDR.=FFFF up
1	0	0	-----	Reserved	-----
1	0	1	เอาต์พุต	Perpheral	BYTE WIDE
1	1	0	-----	Reserved	-----
1	1	1	อินพุต	Slave serial	BIT Serial

จากความต้องการสร้างให้ใช้กระแสไฟฟ้าต่ำในแบบการใช้งานทั้ง 5 แบบจึงมีเพียง 2 แบบเท่านั้นที่เหมาะสม คือ แบบมาสเตอร์ซีเรียลแบบ และแบบสเลฟซีเรียลส่วนในมาสเตอร์พาราเรลต้องใช้ EPROM 27Cxxx ซึ่งกินกระแสมากกว่า PROM XC17xxx เหมาะในการทดสอบต้นแบบก่อนเมื่อวงจรต้นแบบทำงานได้ถูกต้องแล้วจึงทำการอัดโปรแกรมลงบน PROM อีกทีหนึ่งเพราะว่าในแบบพาราเรล EPROM สามารถโปรแกรมได้ใหม่ต่างกับ PROM ที่โปรแกรมได้ครั้งเดียว

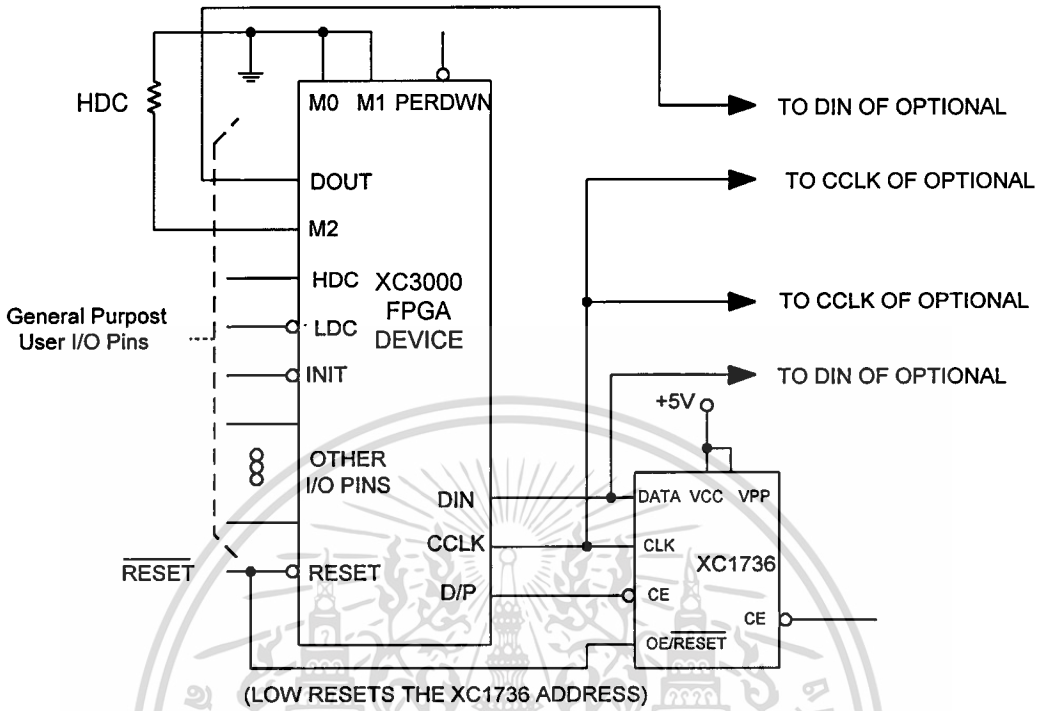
การใช้งาน FPGA_s ในแบบสเลฟซีเรียลและมาสเตอร์ซีเรียล เมื่อเริ่มจ่ายไฟเข้าตัว FPGA_s จะทำการเคลียร์หน่วยความจำใช้คอนฟิคตรวจสอบแบบการทำการคอนฟิคว่าเป็นแบบใด ในตารางที่ 2.2 (เป็นแบบอนุกรมหรือขนาน) หลังจากนั้นจะเริ่มโปรแกรมคอนฟิคสัญญาณ Done / Program เป็น 0 (อยู่ระหว่างโปรแกรม) และเมื่อข้อมูลในการคอนฟิคที่รับมาจากภายนอกเต็มหน่วยความจำที่ใช้ในการคอนฟิคและตามความยาวของข้อมูลตรงกับที่ส่วนหัวของข้อมูลส่วนคอนฟิคสัญญาณ Done / Program เป็น 1 ดังในรูปที่ 2.16



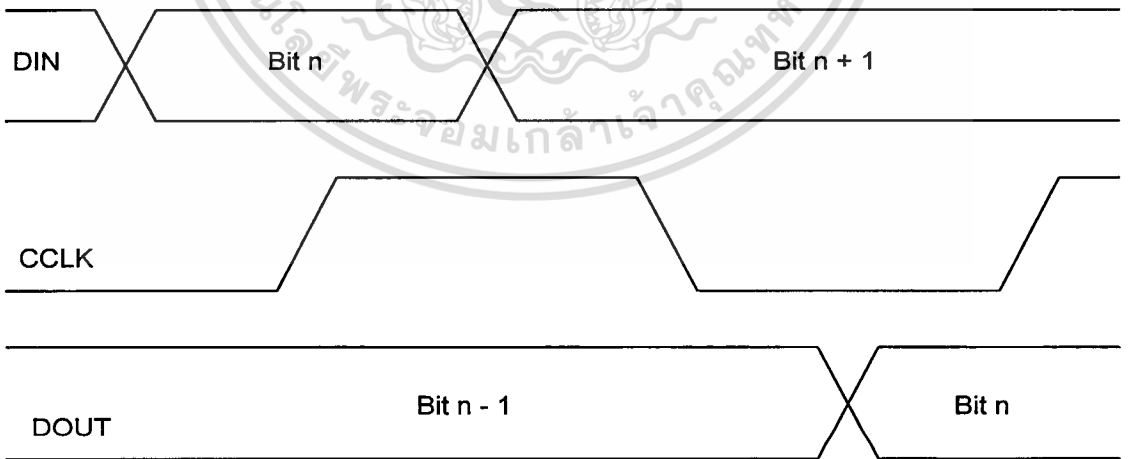
รูปที่ 2.16 ฟังก์ชันในการคอนฟิกเมื่อเริ่มป้อนแหล่งจ่ายไฟเข้าไอซีและการโปรแกรมใหม่

2.18.5 การใช้งานแบบสเลฟซีเรียล

การต่อการใช้งานในแบบนี้เหมาะสมกับวงจรที่ออกแบบมาเพื่อทำงานร่วมกับไมโครคอนโทรลเลอร์แล้วทั้งนี้เพราะ FPGA_s ได้ใช้ความสามารถของไมโครคอนโทรลเลอร์ในการเก็บและส่งข้อมูลคอนฟิกให้ เพียงแต่ต้องเขียน โปรแกรมเพื่อส่งโปรแกรมคอนฟิกให้เพิ่มลักษณะการต่อในแบบนี้เป็นดังรูปที่ 2.17 ซึ่งในแบบนี้ไมโครคอนโทรลเลอร์จะสร้างสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGA_s การป้อนโปรแกรมคอนฟิกให้อุปกรณ์ FPGA_s ทำได้โดยต่อสัญญาณ Strobe เข้ากับขา CCLK และพอร์ต Do เข้ากับขา Din สร้างสัญญาณนาฬิกาป้อนที่ขา CCLK และป้อนโปรแกรมคอนฟิกแบบอนุกรมเข้าที่ขา Din ดังแผนภูมิในรูปที่ 2.18

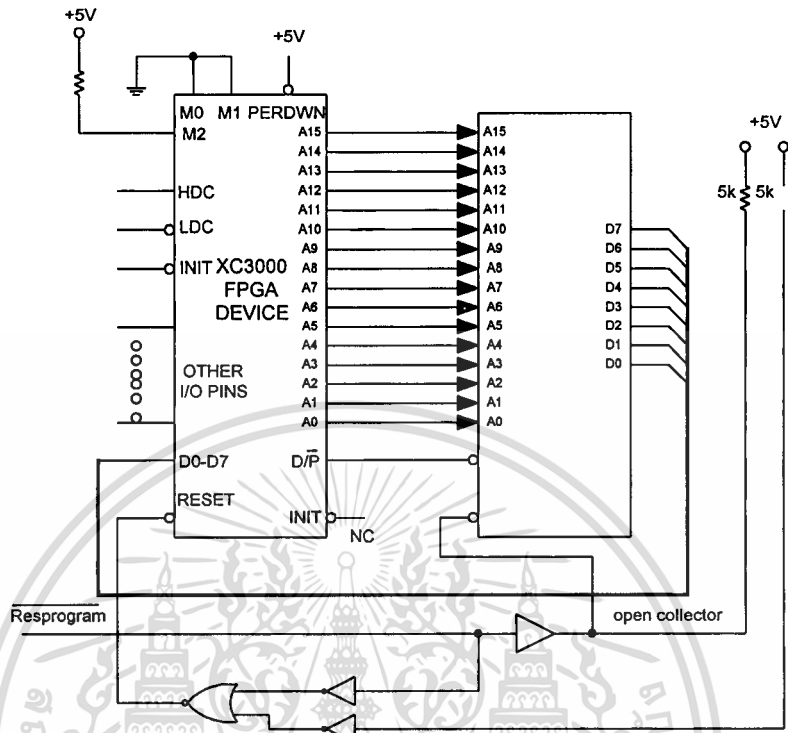


รูปที่ 2.17 การต่อใช้งานในลักษณะสเลฟซีเรียล



รูปที่ 2.18 แผนภูมิเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในแบบสเลฟซีเรียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การต่อใช้งานในแบบมาสเตอร์พาราเรล

2.18.7 ข้อควรระวังในการใช้ FPGA_s

สิ่งแรกที่สำคัญคือไอซีกลัวความร้อนเป็นที่สุด การบัดกรีโดยหัวแร้งกำลังสูง หรือบัดกรีโดยหัวแร้งที่ไอซีนาน ๆ จะทำให้ไอซีเสียหายได้ง่ายระยะเวลาในการบัดกรี 1 จุดไม่ควรเกิน 5-10 วินาทีควรใช้ Socket ไอซีในการประกอบวงจรลงบนแผ่นปริ้นท์

การป้อนแหล่งจ่ายไฟให้กับไอซีไม่ควรผิดพลาด ถ้าสลับขั้วบวกหรือลบกันจะทำให้ไอซีเสียหาย นอกจากนั้นแหล่งจ่ายไฟต้องอยู่ในช่วงที่โรงงานกำหนดมา สำหรับ FPGA_s แรงดันของไอซีอยู่ในช่วงทำงานได้ $V_{cc} = 4.75-5.25$ โวลต์ และแรงดันที่ทนได้อยู่ในช่วง $-0.5-7$ โวลต์ ดังนั้นก่อนป้อนแรงดันไฟโปรดตรวจสอบให้ดีให้แน่นอนก่อน

การต่อวงจรผิดและส่งผลให้วงจรเสียหายทันทีที่ได้แก่ การต่อเอาต์พุตของไอซีไปเข้าขั้วบวกหรือขั้วลบของแหล่งจ่ายไฟโดยตรง IC TTL ในขณะที่ CMOS ยังพอทนได้บ้างเพราะมีการจัดกระแสขาออกอยู่ในตัว ดังนั้นก่อนป้อนแรงดันเข้าตัวไอซีควรตรวจสอบบริเวณเอาต์พุตของไอซีด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกประการหนึ่งที่ไอซีประเภท CMOS กลัวมากที่สุดคือไฟฟ้าสถิต ขาที่เป็นอันตรายได้ง่ายคือขาอินพุตความต้านทานขาเข้าของขาอินพุตของไอซีประเภทนี้มักจะสูงมากจึงมีคุณสมบัติเหมือนกับเป็นฉนวนที่ขาอินพุตนี้อาจรับไฟฟ้าสถิตจากที่ใดมาเก็บไว้เมื่อสะสมไว้มาก ๆ ไม่มีทางระบายออกก็จะทำให้วงจรบริเวณอินพุตเสียหายได้ ในปัจจุบันเทคโนโลยีการผลิตไอซีเจริญรุดหน้าไปมากทางด้านอินพุตของไอซี MOS จะมีวงจรป้องกันไว้ภายในอย่างไรก็ตามเพื่อป้องกันเหตุการณ์ดังกล่าวควรเก็บรักษาไอซีให้ถูกต้องและควรเก็บไว้ที่ที่ปลอดภัยจากไฟฟ้าสถิต โดยเก็บไว้บนแผ่นเชื้อสังเคราะห์แบบนำกระแสได้หรือห่อหุ้มด้วยกระดาษอะลูมิเนียม

ไอซีประเภท CMOS มีความต้านทานขาเข้าสูงมาก กระแสจึงไม่ไหลจากแหล่งจ่ายไฟเข้าไปที่ขาอินพุตของ CMOS นั้นถึงแม้จะต่อความต้านทานคั่นกลางอยู่ แรงดัน 5 โวลต์ โดยที่ไม่มีแรงดันตกคร่อมบนตัวต้านทานเลยถ้ามีกระแสแสดงว่าไอซีตัวนั้นเสีย เอาต์พุตของไอซี CMOS อยู่ในสถานะ 1 จะสามารถขับโหลดได้หลายมิลลิแอมป์และถ้าเอาต์พุตอยู่ในสถานะ 0 ก็จะรับกระแสไหลเข้าได้หลายมิลลิแอมป์

2.18.8 การใช้ซอฟต์แวร์ของบริษัทวิวลอลจิกและบริษัทไซลิงค์

- ขั้นตอนการออกแบบและจำลองการทำงานโดยใช้ซอฟต์แวร์ของบริษัทวิวลอลจิก

เริ่มต้นกำหนดไลบรารีที่ต้องการจะใช้เป็น FPGA ตระกูล XC4000 และใช้โปรแกรมวาดรอยเขียนโปรแกรมวงจรจากนั้นแปลงไฟล์ที่เป็นผังวงจรให้เป็น Wirlist เพื่อนำไปจำลองการทำงานโดยโปรแกรมวิวมิม และใช้วีวเทรตเพื่อดูรูปร่างของสัญญาณที่ขาอินพุตหรือเอาต์พุต โดยรับสัญญาณอินพุตที่ผู้ออกแบบวงจรที่สร้างไว้ เพื่อทดสอบวงจรให้ได้ตรงกับความต้องการ ซึ่งมีหลายวิธีอาจจะป้อนที่ละคำสั่งหรือเขียนเป็นคำสั่งต่อ ๆ กันเก็บไว้ในรูปของไฟล์ข้อความนามสกุล .CMD

- ขั้นตอนในการทดสอบวงจรโดยใช้ซอฟต์แวร์ของบริษัทไซลิงค์

สร้างแม่พิมพ์จากเน็ตลิส (Netlist) โดยใช้โปรแกรม XNFMAP และใช้โปรแกรม MAP2LCA สร้างไฟล์ .LCA ซึ่งสามารถแก้ไขได้โดยผู้ออกแบบหรือใช้โปรแกรม ARR(Auto - automatic Place and Route) จัดการให้โดยอัตโนมัติ ในขั้นตอนทั้งหมดที่กล่าวมาข้างต้นสามารถใช้โปรแกรม XMAKE จากนั้นแปลงจาก ไฟล์ .LCA เป็น.BIT โดยโปรแกรม Makebit จากนั้นแปลงเป็น Intel Xex File เพื่อใช้เป็นคอนฟิกให้กับ FPGA ในกรณีที่ทดสอบวงจรให้กับมาสเตอร์พาราเรลหรือแปลงเป็นบิตสตรีม เพื่อใช้โปรแกรมลงบน PROM ตระกูล XC17xxx ซึ่งเป็นตัวโหลดอุปกรณ์คอนฟิกให้กับ FPGA ในกรณีที่ทดสอบวงจรในแบบมาสเตอร์ซีเรียล เพื่อให้จำลองการ

ทำงานของวงจรให้สมบูรณ์ยิ่งขึ้น หลังจากที่ได้ไฟล์ที่มีโครงสร้างเป็น .LCA แล้วโดยใช้ซอฟต์แวร์แปลงกลับเป็นผังวงจรเพื่อจำลองการทำงานใหม่อีกครั้ง ให้ผลจำลองตรงกับครั้งแรกที่จำลองการทำงาน ซึ่งข้อดีของผังวงจรใหม่ที่สร้างขึ้นนี้เปรียบเสมือนการขุบวงจรให้มันเอง



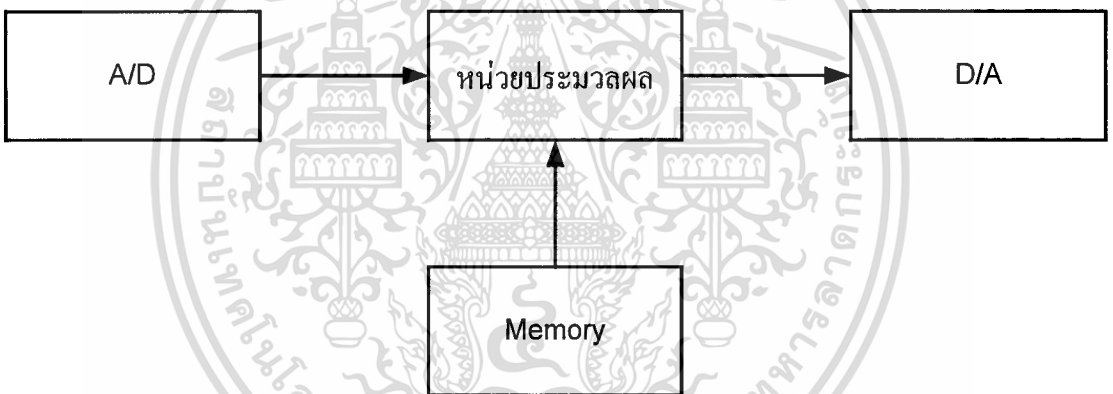
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

3.1 กล่าวนำ

วงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA_s จะประกอบไปด้วยวงจรย่อย ๆ ดังนี้คือวงจรบวกเลขขนาด 16 บิต วงจรคูณเลข 8 บิตคูณ 8 บิต วงจรเลื่อนข้อมูลขนาด 8 บิต วงจรนับเลข 0-9 วงจรถอดรหัส วงจรเลขชี้ วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล วงจรสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกและวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ FPGA_s



รูปที่ 3.1 บล็อกไดอะแกรมของวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้ อุปกรณ์ FPGA_s

3.1.1 ข้อกำหนดในการทำงานของบล็อกไดอะแกรมแต่ละส่วน

-วงจรบวกเลขขนาด 16 บิต ถูกออกแบบไว้ให้อยู่ภายในวงจรกรองความถี่เชิงเลขแบบ จำกัดผลตอบสนองอันดับ 10 ซึ่งจะมีทั้งหมด 9 ชุด แต่ละชุดสามารถประมวลผลการบวกเลขได้ 16 บิตจากนั้นส่งข้อมูลที่ทำกรบวกให้วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก

-วงจรคูณเลข 8 บิตคูณ 8 บิต ถูกออกแบบไว้ให้อยู่ภายในวงจรกรองความถี่เชิงเลขแบบ จำกัดผลตอบสนองอันดับ 10 ซึ่งจะมีจำนวนทั้งหมด 10 ชุด แต่ละชุดสามารถคูณเลขได้ 8 บิตคูณ 8 บิต ส่งข้อมูลที่ทำกรคูณให้วงจรบวกเลขขนาด 16 บิต

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-วงจรเลื่อนข้อมูลขนาด 8 บิต ถูกออกแบบไว้ให้อยู่ภายในวงจรรอกความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 ซึ่งจะมีทั้งหมด 10 ชุด แต่ละชุดจะทำหน้าที่ในการเลื่อนข้อมูลครั้งละ 8 บิต ส่งข้อมูลที่รับเข้ามาจากวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลไปยังวงจรคูณแต่ละชุด

-วงจรมับเลข 0-9 ถูกออกแบบไว้ให้อยู่ภายในวงจรรอกความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 ทำหน้าที่ในการนับรอบการทำงาน

-วงจรถอดรหัส ถูกออกแบบไว้ให้อยู่ภายในวงจรรอกความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 ทำหน้าที่ในการถอดรหัส

-วงจรเลขชี้ ถูกออกแบบไว้ให้อยู่ภายในวงจรรอกความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 ทำหน้าที่ในการค้ำข้อมูล

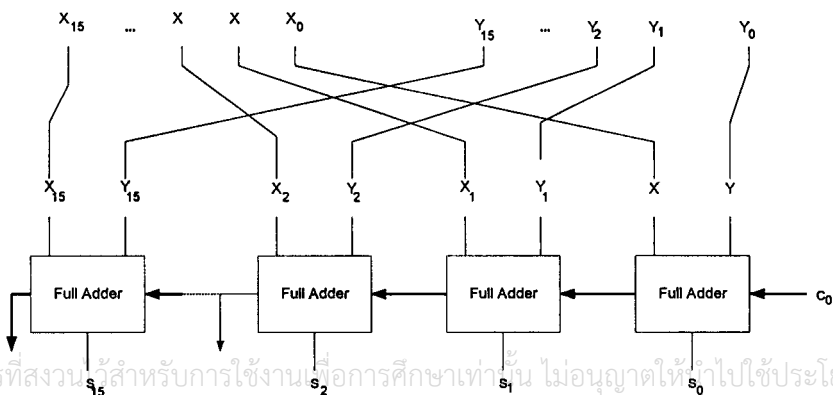
-วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ทำหน้าที่ในการรับข้อมูลจากภายนอกที่อยู่ในรูปของสัญญาณแอนะล็อกเข้ามา แล้วทำการแปลงสัญญาณแอนะล็อกที่ได้เป็นสัญญาณดิจิทัลส่งไปยังวงจรเลื่อนข้อมูลแต่ละชุด

-วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ทำหน้าที่ในการรับข้อมูลที่ส่งออกมาจากวงจรบวกเลขชุดสุดท้ายที่อยู่ในรูปของสัญญาณดิจิทัล เข้ามาแล้วทำการแปลงสัญญาณดิจิทัลที่ได้เป็นสัญญาณแอนะล็อกส่งไปแสดงผลภายนอก

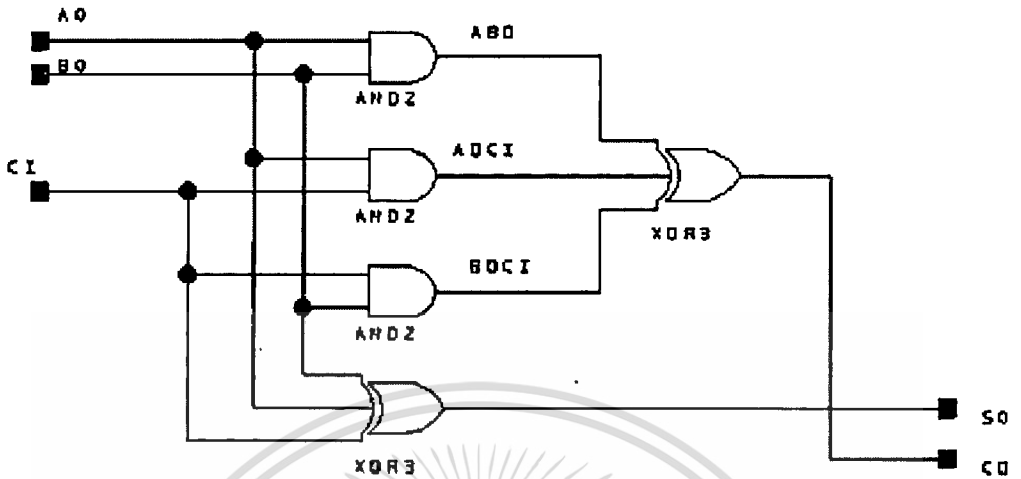
3.2 การออกแบบและการสร้างวงจรแต่ละส่วน

3.2.1 วงจรบวกขนาด 16 บิต

วงจรบวกเลขขนาด 16 บิต ถูกออกแบบให้อยู่ภายในวงจรรอกความถี่เชิงเลขแบบตอบสนองอันดับ 10 ทำหน้าที่ในการบวกเลข มีลักษณะวงจรดังรูปที่ 3.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกครั้งรูปที่ 3.2 บล็อกไดอะแกรมของวงจรบวกเลขขนาด 16 บิต



รูปที่ 3.3 วงจรบวกเลขขนาด 1 บิต

1. หลักการทำงานของวงจรวกเลขขนาด 16 บิต

จากรูปที่ 3.3 วงจรวกเลขขนาด 16 บิตจะมีขาสัญญาณทั้งหมด 5 ขา ซึ่งสามารถอธิบายการทำงานของแต่ละขาดังนี้คือ

- ขาสัญญาณอินพุต (มี 2 ขา) ทำหน้าที่ในการรับข้อมูลจากภายนอกเข้ามาเพื่อส่งให้วงจรวกเลข ทำการประมวลผลส่งไปยังเอาต์พุต

- ขาสัญญาณ ADD/SUB ทำหน้าที่ในการกำหนดสถานะการทำงานของวงจรวกเลข เมื่อขาสัญญาณ ADD/SUB ถูกต่อกับสัญญาณ VCC จะเป็นการทำให้วงจรวกเลขประมวลผลในการบวกเลข แต่เมื่อขาสัญญาณ ADD/ SUB ถูกต่อลง GND จะทำให้วงจรวกเลขประมวลผลในการลบเลข

- ขาสัญญาณ Carryin ทำหน้าที่ในการรับค่าตัวทศที่เกิดจากการบวกกันของวงจรวกเลขตัวก่อน จากนั้นจะนำค่าตัวทศที่รับมาทำการบวกรวมเข้ากับค่าอินพุตที่ได้รับ

- ขาสัญญาณ Carryout ทำหน้าที่ในการส่งค่าตัวทศที่เกิดจากการบวกของวงจรวกเลขตัวนั้น ๆ แล้วส่งค่าตัวทศที่มีไปยังขาสัญญาณแครี่อินของวงจรวกเลขตัวถัดไป

- ขาสัญญาณเอาต์พุต ทำหน้าที่ในการส่งค่าที่ได้จากการบวกเลขของวงจรวกเลข แต่ละชุด (เฉพาะค่าที่ไม่ใช่ตัวทศ) ส่งไปเป็นสัญญาณอินพุตของวงจรวกเลขตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลักการออกแบบและการสร้างวงจรบวกเลขขนาด 16 บิต

จากรูปที่ 3.2 การออกแบบวงจรบวกเลขขนาด 16 บิตจะใช้หลักการเดียวกันกับการออกแบบวงจรบวกเลขขนาด 2 บิต แล้วนำมาเชื่อมโยงกัน

$$x_{15} \dots x_4 x_3 x_2 x_1 x_0$$

$$y_{15} \dots y_4 y_3 y_2 y_1 y_0$$

$$s_{15} \dots s_4 s_3 s_2 s_1 s_0$$

$$c_{15} \dots c_4 c_3 c_2 c_1 c_0$$

Sum

Carry – out

ในการบวกจะบวกในบิตนัยสำคัญต่ำสุดก่อน ค่าที่ได้จะสามารถแสดงเป็นสมการบูลีนได้

$$S_0 = C_0 \oplus (X_0 \oplus Y_0)$$

$$C_1 = X_0 Y_0 + C_0 (X_0 \oplus Y_0)$$

ผลการบวกและฟังก์ชันของตัวทศจะสามารถแทนสมการได้ดังนี้

$$S_i = C_i \oplus (X_i \oplus Y_i)$$

$$C_{i+1} = X_i Y_i + C_i (X_i \oplus Y_i)$$

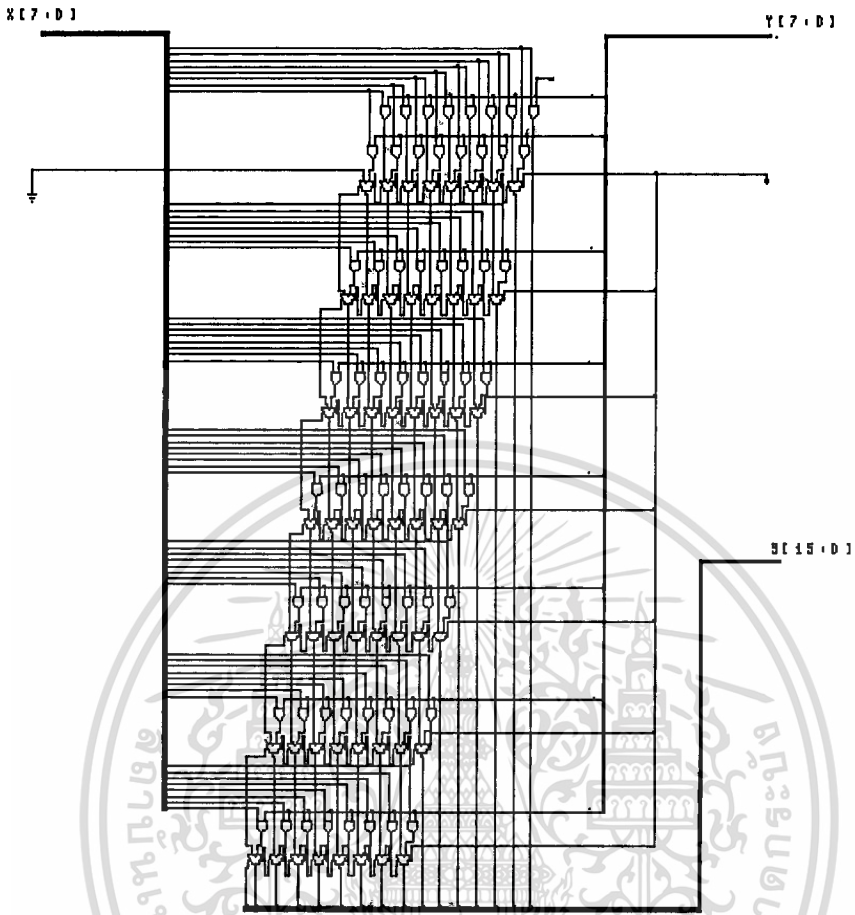
เมื่อ $i = 1, 2, 3, \dots$

3.2.2 วงจรคูณเลข 8 บิตคูณ 8 บิต

วงจรมคูณเลข 8 บิตคูณ 8 บิตถูกออกแบบให้อยู่ภายในวงจรกรองความถี่เชิงเลขอันดับ 10 ทำหน้าที่ในการคูณ มีลักษณะวงจรมคูณรูปที่ 3.4

1. หลักการทำงานของวงจรมคูณเลข 8 บิตคูณ 8 บิต

วงจรมคูณเลข 8 บิตคูณ 8 บิตทำหน้าที่ในการคูณเลขขนาด 8 บิต ที่รับเข้ามาจากวงจรมคูณข้อมูลขนาด 8 บิต กับข้อมูลขนาด 8 บิต ที่ถูกโปรแกรมอยู่ในวงจรมคูณผลลัพธ์ที่ได้จะมีขนาด 16 บิต ส่งผลที่ได้ไปยังวงจรมบวกเลขขนาด 16 บิต การคูณจะมีลักษณะเหมือนกับการคูณเลขทั่วไปดังนี้คือ



รูปที่ 3.4 วงจรคูณเลข 8 บิตคูณ 8 บิต

$$X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0$$

$$Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0$$

$$ZT_0 = Z_{07} Z_{06} Z_{05} Z_{04} Z_{03} Z_{02} Z_{01} Z_{00}$$

$$ZT_1 = Z_{17} Z_{16} Z_{15} Z_{14} Z_{13} Z_{12} Z_{11} Z_{10}$$

$$ZT_2 = Z_{27} Z_{26} Z_{25} Z_{24} Z_{23} Z_{22} Z_{21} Z_{20}$$

$$ZT_3 = Z_{37} Z_{36} Z_{35} Z_{34} Z_{33} Z_{32} Z_{31} Z_{30}$$

$$ZT_4 = Z_{47} Z_{46} Z_{45} Z_{44} Z_{43} Z_{42} Z_{41} Z_{40}$$

$$ZT_5 = Z_{57} Z_{56} Z_{55} Z_{54} Z_{53} Z_{52} Z_{51} Z_{50}$$

$$ZT_6 = Z_{67} Z_{66} Z_{65} Z_{64} Z_{63} Z_{62} Z_{61} Z_{60}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $ZT_7 = Z_{77} Z_{76} Z_{75} Z_{74} Z_{73} Z_{72} Z_{71} Z_{70}$ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

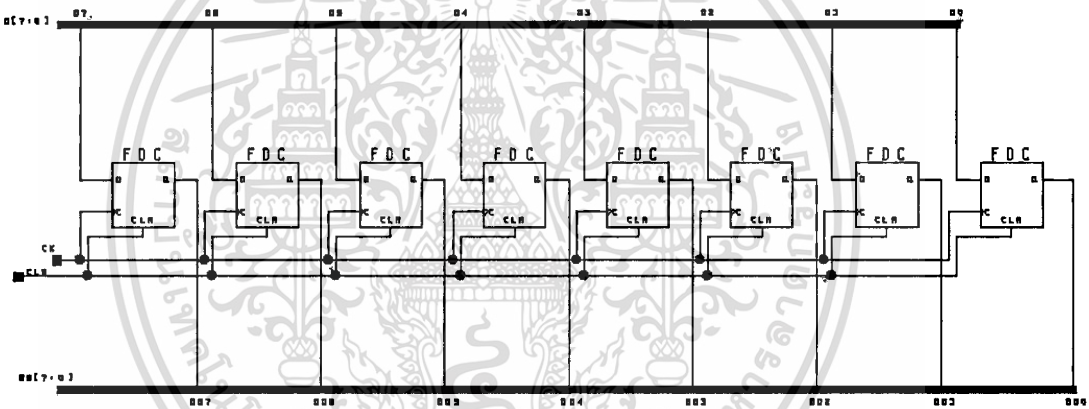
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้าม $ZT = ZT_0 + ZT_1 + ZT_2 + ZT_3 + ZT_4 + ZT_5 + ZT_6 + ZT_7$ ออกสารทุกครั้งที่มีการนำไปใช้

2. หลักการออกแบบและการสร้างวงคูณเลข 8 บิตคูณ 8 บิต

จากหลักการทำงานดังกล่าวข้างต้นจะพบว่าระหว่างทำการคูณของแต่ละหลักจะต้องทำการบวกผลคูณที่ได้ก่อนจึงจะทำการคูณหลักต่อไปได้เพราะฉะนั้นต้องเพิ่มวงจรบวกเลขเข้ามาเพื่อทำหน้าที่ในการบวกผลคูณที่ได้แต่ละหลักก่อนจึงจะผ่านไปทำการคูณหลักอื่นได้และจะทำอย่างนี้ไปเรื่อย ๆ จนกว่าจะครบทั้ง 8 บิต

3.2.3 วงจรเลื่อนข้อมูลขนาด 8 บิต

วงจรถ่ายโอนข้อมูลขนาด 8 บิต ถูกออกแบบให้อยู่ภายในวงจรรอกความถี่เชิงเลขอันดับ 10 ทำหน้าที่ในการเลื่อนข้อมูลขนาด 8 บิต มีลักษณะวงจรถ่ายโอนรูปที่ 3.5



รูปที่ 3.5 วงจรถ่ายโอนข้อมูลขนาด 8 บิต

1. หลักการทำงานของเครื่องส่งข้อมูลแบบขนาน

ชิพตรีจิสเตอร์ที่กล่าวมาแล้วเป็นรีจิสเตอร์ที่มีการส่งข้อมูลทั้งในการเลื่อนข้อมูลเข้าและออกเป็นแบบอนุกรมครั้งละ 1 บิต ซึ่งเป็นที่นิยมในอุปกรณ์รับงาน/แสดงผลระบบดิจิทัลทั่วไป แต่วิธีการเช่นนี้จะสูญเสียเวลาในการส่ง 1 บิต ต่อเวลาของสัญญาณเลื่อน 1 ตัว ฉะนั้นถ้าต้องการเลื่อนข้อมูลขนาด 4 บิตก็ต้องใช้เวลาเป็น 4 เท่าของสัญญาณเลื่อน 1 ตัว หรือเป็น 4 เท่า ของการเลื่อนข้อมูล 1 บิต ฉะนั้นยิ่งข้อมูลมีจำนวนมากก็ยิ่งเสียเวลามาก แต่ยังมีวิธีส่งหรือเลื่อนข้อมูลที่สามารถทำได้รวดเร็วกว่าโดยการส่งข้อมูลทุก ๆ ตัวพร้อมกันในเวลาเดียวกัน เรียกว่า การส่งแบบขนานซึ่งใช้กันมากในการส่งข้อมูลในระบบดิจิทัล

2. หลักการทำงานของวงจรเลื่อนข้อมูลขนาด 8 บิต

วงจรเลื่อนข้อมูลขนาด 8 บิต เป็นวงจรที่ทำหน้าที่ในการเลื่อนข้อมูลขนาด 8 บิต จากรูปที่ 3.5 สามารถอธิบายการทำงานได้ดังนี้ ขาสัญญาณ CK (Clock) เป็นตัวกำหนดสถานะการทำงานของเลื่อนข้อมูลขนาด 8 บิต ซึ่งจะทำงานในขอบขาลง ขาสัญญาณ D_0 ถึง D_7 ทำหน้าที่เป็นขาสัญญาณอินพุตเมื่อมีอินพุตเข้ามาที่ D_0 ถึง D_7 ของเลื่อนข้อมูลขนาด 8 บิต จะทำการเลื่อนข้อมูลชุดแรกที่เข้ามาไปเก็บไว้ในวงจรเลื่อนข้อมูลชุดต่อไปและจะทำงานในลักษณะนี้ไปเรื่อย ๆ จนกว่าข้อมูลที่เข้ามาจะหมด

3.2.4 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ทำหน้าที่รับข้อมูลภายนอกเข้าแล้วการแปลงสัญญาณให้อยู่ในรูปสัญญาณดิจิทัล มีลักษณะวงจรดังรูปที่ 3.6



รูปที่ 3.6 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

1. คุณสมบัติของไอซี ADC 908

- ความละเอียดและเที่ยงตรงขนาด 8 บิต
- การแปลงรหัสใช้งานในย่านอุณหภูมิ (0-90C) (ADC 908 GP)
- เวลาในการ Convert 5 ไมโครวินาที
- ยึดหุ่นในการ Interface

- กระแสสงบ 15 มิลลิแอมป์

ADC-908 ผลิตโดยเทคโนโลยี CMOS เมื่อป้อน Clock 1.3 กิโลเฮิร์ตซ์จะใช้เวลาในการ Convert 5 ไมโครวินาที ด้วยความเที่ยงตรงตลอดย่านอุณหภูมิใช้งานเอาต์พุตเป็น Tri- State Logic สามารถต่อตรงกับบัสระบบหรือ พอร์ตขา \overline{CS} และ \overline{RD} ใช้ควบคุมการทำงานของ ADC Interface เหมือนแรมหรือรอมหรือ Slow Memory Mode ในโหมดนี้จะใช้เอาต์พุตของ Busy ในการ Interface เพื่อให้สร้างเวลารอในช่วงเวลาคอนเวอร์เตอร์

2. หลักการทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

วงจรรับสัญญาณแอนะล็อกขนาด 8 ช่องแบบมัลติเพล็กซ์ ทำหน้าที่ในการรับสัญญาณ - แอนะล็อกอินพุตว่าจะรับเอาสัญญาณแอนะล็อกจากช่องไหนเข้าสู่วงจร ส่วนของวงจรสวิทช์ 8 ช่องแบบมัลติเพล็กซ์นี้ถูกควบคุมด้วยการเลือกสัญญาณอินพุตด้วยวงจรแอดเดรสแลตซ์และวงจรถอดรหัส

วงจรแอดเดรสแลตซ์และวงจรถอดรหัส ทำหน้าที่ในการรับข้อมูลแอดเดรสจากภายนอกมาทำการถอดรหัสและสัญญาณที่ถอดรหัสได้นี้จะนำไปควบคุมการเลือกรับสัญญาณอินพุตของวงจรสวิทช์ 8 ช่องแบบมัลติเพล็กซ์

วงจร 8 บิต A/D ประกอบด้วยวงจร 256R รีจิสเตอร์แอดเดรสและสวิทช์ทรี ทำหน้าที่เป็นวงจรสร้างแรงดันอ้างอิง เพื่อนำไปเปรียบเทียบกับสัญญาณแอนะล็อกอินพุต

วงจรคอมพารเตออร์ ทำหน้าที่เปรียบเทียบสัญญาณแอนะล็อกอินพุตกับสัญญาณที่ได้จากวงจรสวิทช์ทรี

วงจร Successive Approximate Resisier (SAR) จะทำหน้าที่ในการแปลงสัญญาณที่ได้จากคอมพารเตออร์ให้เป็นสัญญาณดิจิทัลโดยวิธี Successive Approximate

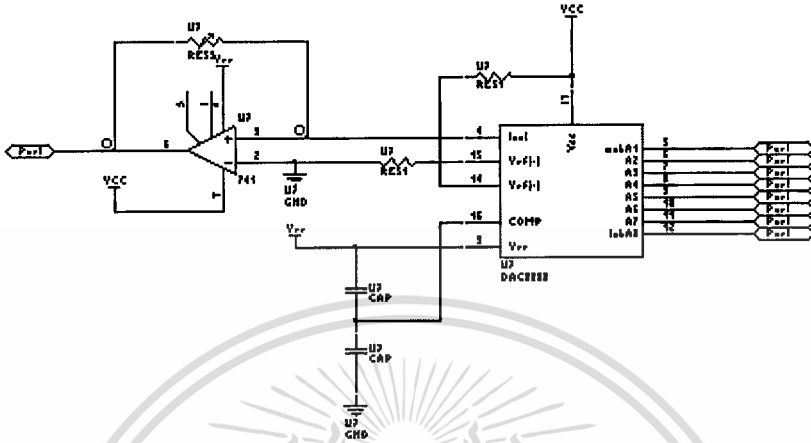
วงจรควบคุมเวลา ทำหน้าที่ในการรับสัญญาณควบคุมจากอุปกรณ์ภายนอกและส่งสัญญาณไปควบคุมภายใน เช่นรับสัญญาณเริ่มต้นที่ขา Start หรือส่งสัญญาณการแปลงสิ้นสุดสัญญาณ A/D (End of Conversion) ไปยังอุปกรณ์ภายนอก

วงจรทรี-สเตทเอาต์พุตแลตซ์ /บัฟเฟอร์ ทำหน้าที่ค้ำข้อมูลดิจิทัลขนาด 8 บิต เพื่อให้วงจรภายนอกรับข้อมูลนี้ออกไป

3.2.5 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ทำหน้าที่ในการรับข้อมูลที่ถูกระมวลผลจากวงจรบวกชุดสุดท้ายแล้วทำการแปลงสัญญาณที่ได้ให้อยู่ในรูปสัญญาณแอนะล็อกส่ง

เอกสารนี้ออกไปแสดงผลภายนอกมีลักษณะวงจรดังรูปที่ 3.7 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

1. หลักการทำงานแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

ในการควบคุมในงานอุตสาหกรรมมักจะต้องเกี่ยวข้องกับสัญญาณที่เป็นแอนะล็อกซึ่งสัญญาณแอนะล็อก ก็คือสัญญาณที่มีความแตกต่างหลาย ๆ ระดับ เช่นระดับแรงดันหรือกระแส ในการควบคุมอุปกรณ์ที่ต้องการสัญญาณแอนะล็อกโดยใช้ไมโครโปรเซสเซอร์ หรือไมโครคอมพิวเตอร์สัญญาณจากไมโครโปรเซสเซอร์หรือไมโครคอมพิวเตอร์จะเป็นสัญญาณแอนะล็อก ดังนั้นต้องมีการเปลี่ยนสัญญาณดิจิตอลให้เป็นแอนะล็อกเพื่อนำไปควบคุมอุปกรณ์เหล่านั้นต่อไป

2. รายละเอียดทางไฟฟ้าของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกมีดังต่อไปนี้

- ความละเอียด

ความละเอียดจะกำหนดโดยสัญญาณบิตดิจิตอล เช่นสัญญาณดิจิตอล 8 บิต จะมีระดับแรงดันเอาต์พุต 256 ระดับที่ไม่ซ้ำกันเลย ($2^8 = 256$) ดังนั้นความละเอียดจะมีค่าเท่ากับจำนวนที่ต่ำสุดหารด้วยจำนวนสูงสุดก็คือ $1/256$ นั่นเอง นิยมบอกในรูปของเปอร์เซ็นต์ เช่น สัญญาณดิจิตอล 12 บิต จะมีเปอร์เซ็นต์ความละเอียดเท่ากับ $1/212 = 1/4096 \times 100 = 0.024 \%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความแม่นยำ

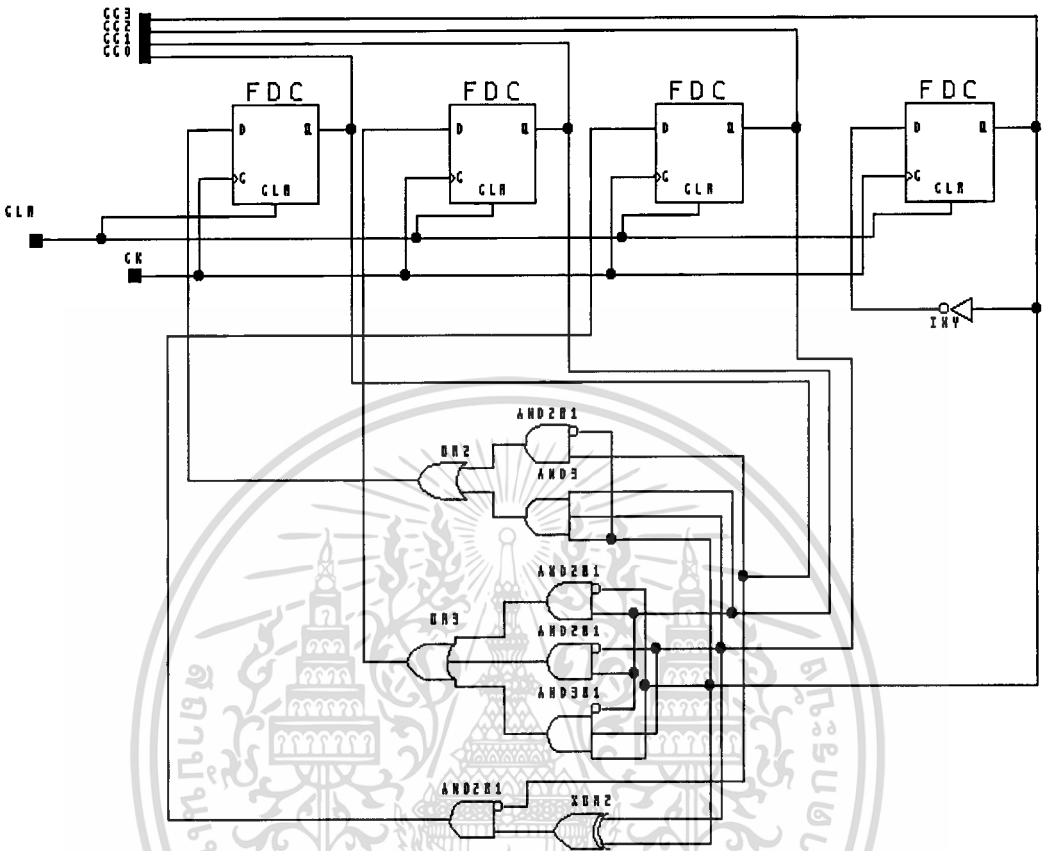
เป็นตัวแสดงว่าค่าที่ได้นั้นผิดพลาดไปจากค่าที่ถูกต้องเพียงใด ปกติจะระบุเป็นเปอร์เซ็นต์ของค่าสูงสุด หรือค่าแรงดันสูงสุดทางด้านเอาต์พุต ตัวอย่างเช่น วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก มีค่าเต็มสเกล 10 โวลต์ และมีความผิดพลาด $\pm 0.2\%$ ดังนั้นคิดในรูปแรงดันจะได้ $(0.2/10) \times 10$ โวลต์ = 20 มิลลิโวลต์โดยทั่วไป แปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก ควรมีความผิดพลาดไม่เกิน ± 0.5 ของหลัก LSB ตัวอย่างเช่น แปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก 8 บิต หลัก LSB เท่ากับ $(1/256) \times 100 = 0.39$ ดังนั้นควรมีค่าความผิดพลาดเท่ากับ $\pm 0.19\%$ หรือเท่ากับ ± 0.5 ของหลัก LSB นั่นเอง ค่าความผิดพลาดที่กล่าวมานี้เรียกว่า ค่าความแม่นยำสัมพัทธ์ (Relative Accuracy) นอกจากนี้ค่าความผิดพลาดยังมีอีกค่าหนึ่งได้แก่ ค่าผลต่างไม่เป็นเชิงเส้น (Differential Linearity) ซึ่งเป็นค่าความผิดพลาดที่คิดอยู่ในรูปของขนาดของอันดับ หมายถึงขนาดของอันดับที่ได้นั้นผิดพลาดจากค่าที่เป็นจริงไปเท่าไร

- ความเร็วในการทำงาน

ความเร็วของแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก เรียกว่า Setting Time หมายถึงความสูงสุดที่ต้องการในการเปลี่ยนเอาต์พุตจาก 0 ถึงค่าเต็มสเกล เมื่ออินพุตเปลี่ยนจากค่าต่ำสุดมาสูงสุด ปกติจะมีค่าระหว่าง 1-20 ไมโครวินาที ถ้าเป็น D/A แบบกระแสเอาต์พุต จะมี Setting Time สั้นกว่า D/A แบบแรงดันเอาต์พุต

3.2.6 วงจรวงจรรัน 0-9

วงจรรันเลข 0-9 ถูกออกแบบให้อยู่ภายในวงจรองความถี่เชิงเลขอันดับ 10 ทำหน้าที่ในการนับรอบการทำงาน มีลักษณะวงจรดังรูปที่ 3.8



รูปที่ 3.8 วงจรนับ 0-9

1. หลักการทำงานของวงจรรนับเลข 0-9

วงจรรนับเลข 0-9 ทำหน้าที่ในการนับรอบการทำงานในแต่ละครั้ง โดยจะเริ่มทำการนับจาก 0 ไปจนถึง 9 แล้วจะเริ่มทำการนับ 0 อีกครั้งหนึ่งจากรูปที่ 3.8 สามารถอธิบายการทำงานได้ดังนี้คือ D ฟลิปฟลอปทั้ง 4 ตัวจะได้รับสถานะการทำงานทางด้านอินพุตต่าง ๆ จากลอจิกเกตและส่งผลลัพธ์ที่ได้ไปยังขา Q ของ D ฟลิปฟลอปแต่ละตัว

2. หลักการออกแบบวงจรร นับเลข 0-9

ขั้นตอนที่ 1 กำหนดสถานะการทำงานลงในตารางความจริง

ตารางที่ 3.1 สภาวะการทำงานของวงจรมับ 0-9

จำนวนนับ	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

ขั้นตอนที่ 2 กำหนดฟลิปฟล็อปที่จะใช้งาน ในที่นี้ถูกกำหนดให้เป็น D ฟลิปฟล็อปจำนวน 4 ตัว

ขั้นตอนที่ 3 นำค่าสภาวะการทำงานของ D ฟลิปฟล็อปจากตารางความจริงมาใส่ลงในตาราง K-MAP

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	0	X	b
11	0	A	X	X
10	0	0	X	X

Flipflop A

AB \ CD	00	01	11	10
00	0	0	X	0
01	a	a	X	0
11	b	b	X	X
10	1	X	X	X

Flipflop B

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	A	B	X	X
10	0	1	X	X

Flipflop C

AB \ CD	00	01	11	10
00	a	a	X	A
01	b	b	X	B
11	b	b	X	X
10	a	a	X	X

Flipflop D

ขั้นตอนที่ 4 การลดรูปสมการ

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	0	0	0
11	0	1	X	X
10	0	0	X	X

Flipflop A

AB \ CD	00	01	11	10
00	0	0	X	0
01	1	1	X	0
11	0	0	X	X
10	1	1	X	X

Flipflop B

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	1	0	X	X
10	0	1	X	X

Flipflop C

AB \ CD	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	0	0	X	X
10	1	1	X	X

Flipflop D

ขั้นตอนที่ 5 นำสมการที่ขุดได้ไปวาดผังวงจรที่ 3.8

$$A = A\bar{C}\bar{D} + BCD$$

$$B = \bar{A}\bar{C}D + C\bar{D}$$

$$C = B\bar{C} + BCD + BC\bar{D}$$

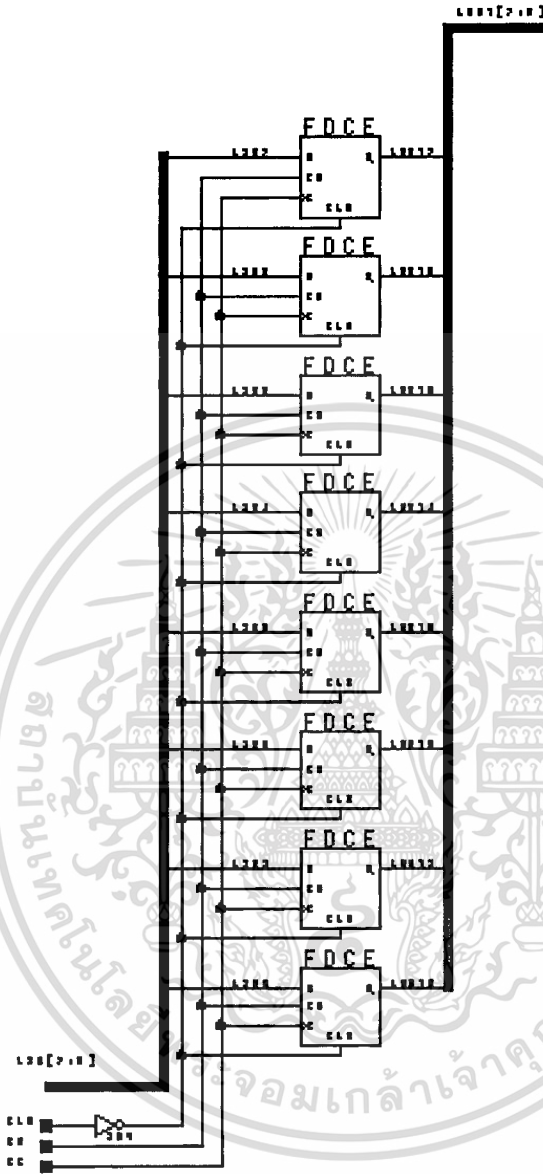
$$D = \bar{D}$$

3.2.7 วงจรเลขทศ

วงจรเลขทศ ถูกออกแบบให้อยู่ภายในวงจรกรองความถี่เชิงเลขอันดับ 10 ทำหน้าที่ในการ ค้างข้อมูล มีลักษณะวงจรดังรูปที่ 3.9

1. หลักการทำงานของวงจรเลขทศ

ลักษณะวงจรเลขทศ จะประกอบด้วยสัญญาณออก 2 สายซึ่งสายด้านหนึ่งจะเป็นค่าปกติ และอีกด้านหนึ่งจะเป็นค่าส่วนเติมซึ่งกันและกันอยู่ทางด้านสายสัญญาณเข้าของวงจร จะมีจำนวน สายต่างกันซึ่งจะขึ้นอยู่กับชนิดของฟลิปฟล็อป ค่าที่ป้อนทางสายสัญญาณเข้าจะทำให้สถานะของ ฟลิปฟล็อปเปลี่ยนไปเป็นสถานะใหม่เราเรียกสถานะของฟลิปฟล็อปก่อนให้สัญญาณเข้าว่าเป็น สถานะปัจจุบันและเรียกสถานะของฟลิปฟล็อปหลังจากให้สัญญาณเข้าว่าเป็นสถานะถัดไป การ ทำงานของวงจรจะมี 2 ลักษณะ คือ สถานะ 0 และสถานะ 1 สามารถทำได้สองแบบ คือ แบบแรกใช้ สัญญาณพัลส์ป้อนทางสัญญาณเข้าโดยตรงจะทำให้สัญญาณออกของฟลิปฟล็อปเปลี่ยนสถานะ ฟลิปฟล็อปประเภทนี้ เรียกว่า วงจรเลขทศ

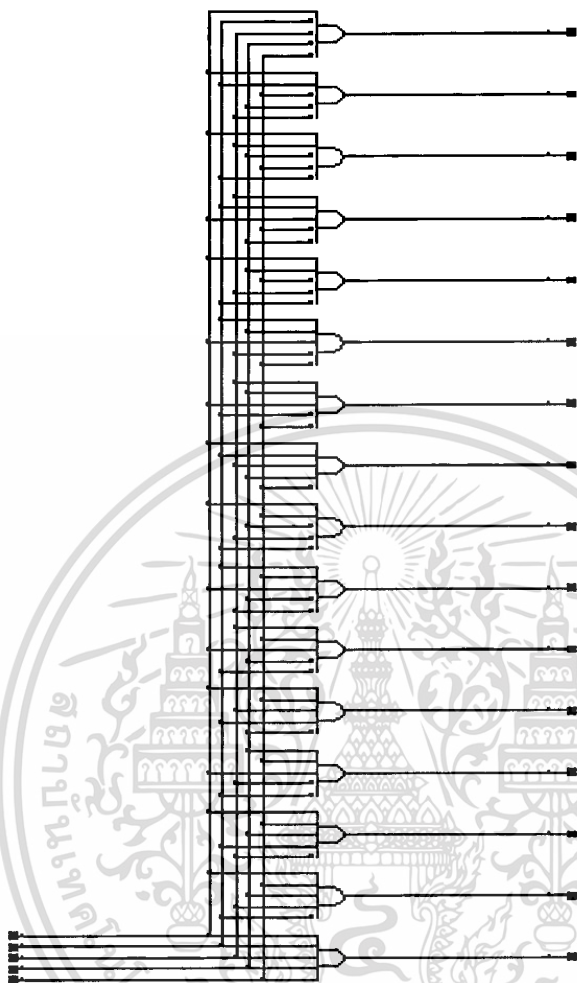


รูปที่ 3.9 วงจรเลขทศ

3.2.8 วงจรถอดรหัส

วงจรถอดรหัส ถูกออกแบบให้อยู่ภายในวงจรรอกความถี่เชิงเลขอันดับ 10 มีลักษณะวงจร ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจรถอดรหัส

1. หลักการทำงานของวงจรถอดรหัส

วงจรถอดรหัสคืออุปกรณ์แปลงรหัส ซึ่งจะทำงานเหมือนกับวงจรถอดรหัสที่คือวงจรถอดรหัสต่าง ๆ นั่นเอง ดังนั้นวงจรถอดรหัสเป็นวงจรถอดรหัสที่ทำหน้าที่เปลี่ยนรหัสเลขไบนารีที่อินพุตจำนวน N บิต ให้เป็นสายเอาต์พุต จำนวน M สาย โดยสายเอาต์พุตแต่ละสายจะได้รับการกระทำมาจากการจัดหมู่ของอินพุตที่เหมาะสมเพียงกลุ่มเดียว ยกตัวอย่างของวงจรถอดรหัสได้แก่ การเปลี่ยนรหัส BCD เป็นเลขฐาน 10

จากรูปที่ 3.10 จะเห็นว่าวงจรถอดรหัสจะทำงานเป็นแบบ Active high (นั่นคือสภาวะปกติจะเป็น Low) และมีจำนวนอินพุตเท่ากับ 4 บิต ($N = 4$) ซึ่งจะแทนรหัส BCD ทางอินพุตและจะสร้างเอาต์พุตจำนวน 10 เอาต์พุต ($M = 10$) โดยเอาต์พุตแต่ละเส้นจะแทนเลขฐาน 10 แต่ละตัวใช้

เนื่องจากอินพุตมี 4 บิต ดังนั้นจำนวนแถวของตารางความจริง จะเท่ากับ 2^4 เท่ากับ 16 แถว โดยแต่ละแถวจะสร้าง 1 เอาต์พุต เป็นแบบ Active high (สถานะปกติจะเป็น Low) ซึ่งจะแทนตัวเลขฐาน 10 อยู่ 1 ตัว ส่วนเอาต์พุตที่เหลือจะเป็น Low ดังนั้นจำนวนเอาต์พุตที่ใช้งานจะมีเพียง 10 เอาต์พุต ซึ่งจะแทนเลขฐาน 10 จาก 0-9 ส่วนจำนวนแถวที่เหลืออีก 6 แถว ของตารางความจริงจะไม่สร้างเอาต์พุตเป็น Active high จึงกำหนดให้มีเอาต์พุตเป็น LOW ทั้งหมดในรูปแบบที่ 3.10 ซึ่ง เอาต์พุตที่ไม่ได้ใช้งานจะไม่เขียนไว้

ตารางที่ 3.2 สถานะการทำงานของวงจรถอดรหัส

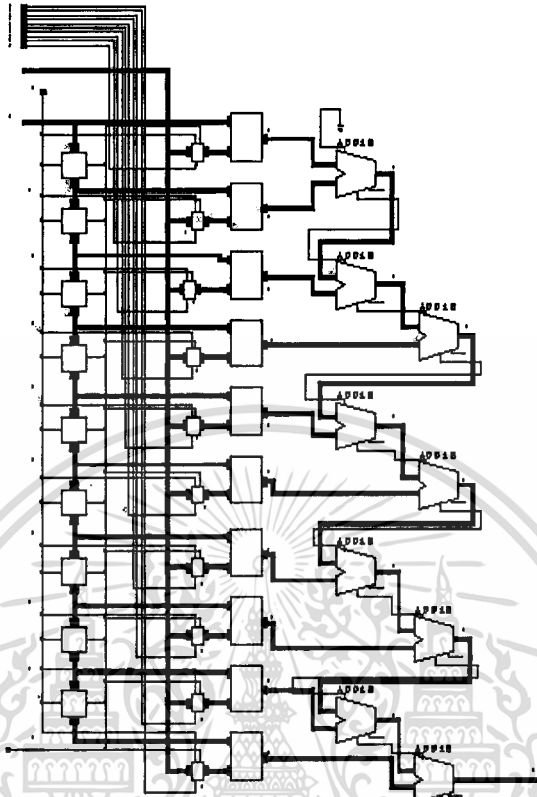
A3	A2	A1	A0	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	O ₉
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

3.2.9 หน่วยประมวลผลของวงจรถอดความถี่เชิงเลขแบบจำกัดผลตอบสนอง

อันดับ 10 (FIR)

หน่วยประมวลผลของวงจรถอดความถี่เชิงเลขแบบจำกัดผลตอบสนองความถี่อันดับ 10 ทำหน้าที่ในการประมวลผลเชิงเลข 10 อันดับ ซึ่งเป็นการรวมวงจรถอดความถี่ที่กล่าวมาข้างต้นมาบรรจุลงใน FPGA_s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



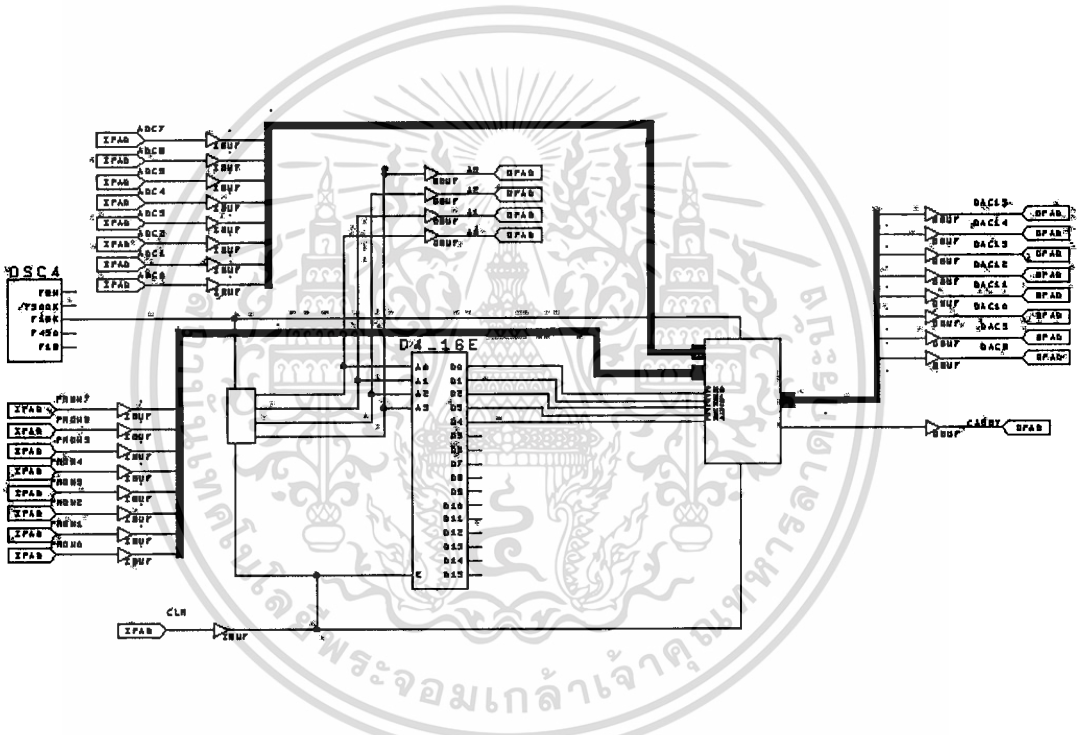
รูปที่ 3.11 หน่วยประมวลผลวงจรรองความถี่เชิงเลข แบบจำกัดผลตอบสนองอันดับ 10 (FIR)

1. หลักการทำงานและการออกแบบหน่วยประมวลผลวงจรรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10

จากรูปวงจรที่ 3.11 หน่วยประมวลผลวงจรรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล จะเป็นตัวรับข้อมูลจากภายนอกซึ่งมีลักษณะสัญญาณแอนะล็อกเมื่อผ่านเข้ามาในวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลจะทำการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลส่งผ่านเข้าไปในวงจรชิพพีซีทีขนาด 8 บิต วงจรเลื่อนข้อมูลขนาด 8 บิต จะทำการเลื่อนข้อมูลที่ได้รับจากวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล เข้าไปยังวงจรคูณขนาด 8 บิต (การเลื่อนข้อมูลจะมีลักษณะการเลื่อนข้อมูลลงมาเรื่อยๆจนกว่าจะหมด) ภายในวงจรคูณขนาด 8 บิตจะมีค่าคงที่ที่ถูกผู้สร้าง โปรแกรมไว้ยู่ค่าหนึ่งซึ่งมีขนาด 8 บิต เมื่อวงจรคูณได้รับข้อมูลจากวงจรเลื่อนข้อมูลขนาด 8 บิต แล้วจะทำการคูณระหว่างค่าที่ได้รับจากวงจรเลื่อนข้อมูลขนาด 8 บิต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาด 8 บิตกับค่าคงที่ที่ถูกโปรแกรมไว้ภายในวงจรคูณ ผลลัพธ์ที่ได้จะมีขนาด 16บิตถูกส่งผ่านเข้าไปในวงจรวงจรวกเลข ขนาด 16 บิต เพื่อทำการบวกค่าผลลัพธ์ที่ได้จากวงจรมultiplicationแต่ละตัวจากนั้นส่งข้อมูลที่ทำกรบวกเข้าไปยังวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกเพื่อทำการแปลงข้อมูลที่อยู่ในรูปของสัญญาณดิจิทัลให้เป็นสัญญาณแอนะล็อกอีกครั้งหนึ่งส่งไปแสดงผลภายนอก

3.3 การเชื่อมต่อวงจรถอดรหัสหน่วยความจำ หน่วยประมวลผลวงจรรองความถี่เชิงเลขและวงจรถ่ายความถี่ 8 กิโลเฮิร์ตซ์



รูปที่ 3.12 การเชื่อมต่อหน่วยประมวลผลเข้ากับวงจรถอดรหัสและตัวกำเนิดความถี่ 8 กิโลเฮิร์ตซ์

เมื่อเสร็จสิ้นวงจรรวมในการเชื่อมต่อหน่วยประมวลผลเข้ากับตัวถอดรหัสหน่วยความจำและตัวกำเนิดสัญญาณ 8 กิโลเฮิร์ตซ์ แล้วจำเป็นต้องใส่บัฟเฟอร์ทั้งทางด้านที่เป็นอินพุตและเอาต์พุตและต่อเข้ากับอุปกรณ์ IOPAD ถ้ากรณีเป็นอินพุตให้ใส่อุปกรณ์อินพุตบัฟเฟอร์และ IPAD เช่นเดียวกันกรณีเป็นสัญญาณเอาต์พุตให้ใส่อุปกรณ์เอาต์พุตบัฟเฟอร์และ OPAD ขาสัญญาณทั้งหมดมีดังนี้

- ออสซิลเลเตอร์ 8 กิโลเฮิร์ตซ์ (อินพุต) เป็นขากำเนิดสัญญาณสุ่มขนาด 8 กิโลเฮิร์ตซ์ เข้ากับขา RD ของ วงจร A/D

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ สัญญาณข้อมูล A/D (อินพุต) มีขนาด 8 บิต

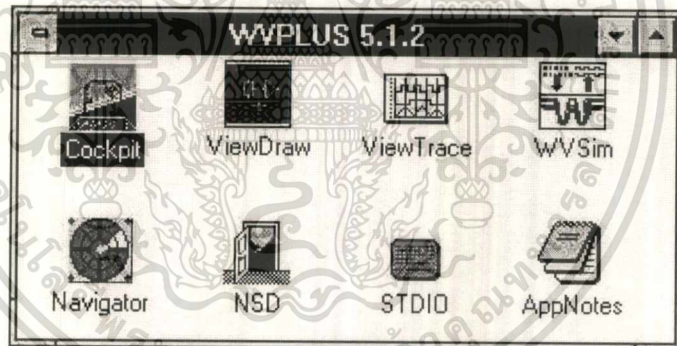
- สัญญาณข้อมูลจากหน่วยความจำ EPROM (อินพุต) มีขนาด 8 บิต
- สัญญาณถดถอรหัสตำแหน่ง EPROM (เอาต์พุต) A0,A1,A2
- สัญญาณรีเซต (อินพุต)
- สัญญาณข้อมูล D/A (เอาต์พุต) ขนาด 16 บิต นำเฉพาะ 8 บิตบนมาใช้งาน

3.4 สรุปขั้นตอนการออกแบบและการสร้าง

ในการสร้างวงจรแต่ละส่วนดังที่กล่าวมาแล้วในหัวข้อที่ 3.2 ยกเว้นวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลและวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกวงจรทั้งหมดจะถูกนำไปสร้างในโปรแกรมวิวลอจิกของบริษัทไซลิงค์ซึ่งมีขั้นตอนดังนี้

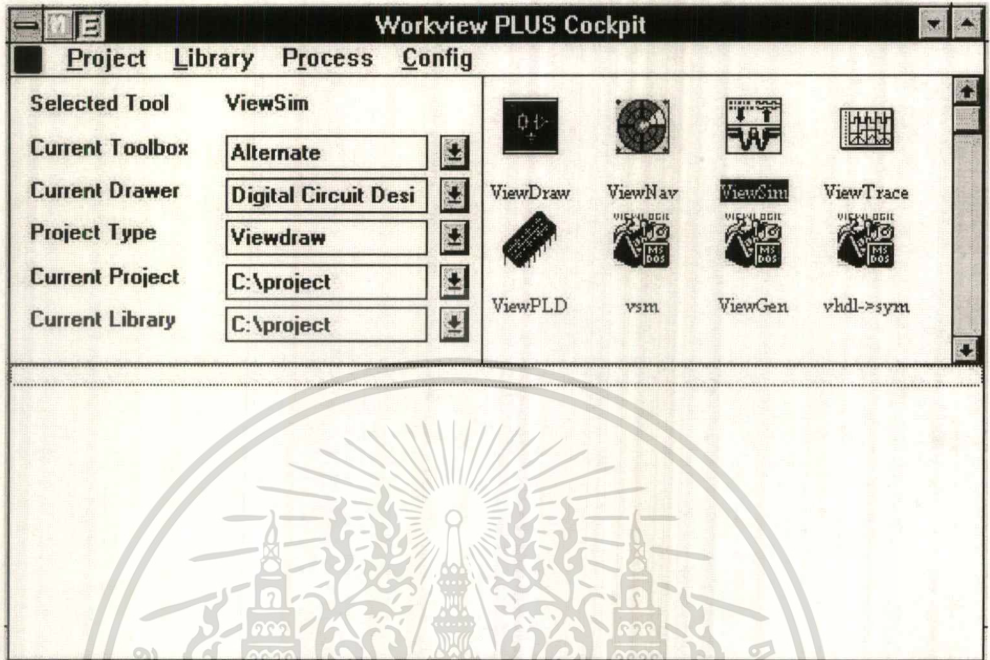
3.4.1 ขั้นตอนการออกแบบและจำลองการทำงานโดยใช้ซอฟต์แวร์ของบริษัทวิวลอจิก

ซอฟต์แวร์ของบริษัทวิวลอจิกเป็นโปรแกรม Workview Plus ที่ทำงานบนวินโดวส์ 3.13 โดยโปรแกรมหลักดังรูปที่ 3.13



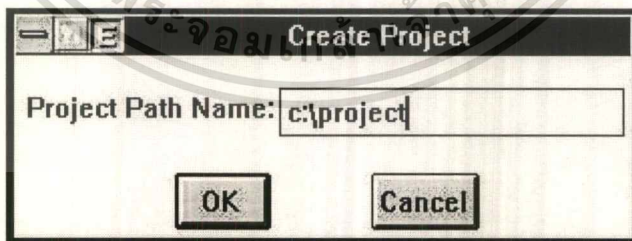
รูปที่ 3.13 หน้าต่างการใช้งานโปรแกรมหลัก View Logic

การใช้โปรแกรมหลัก Cockpit



รูปที่ 3.14 หน้าต่างของวงจรหลัก Cockpit

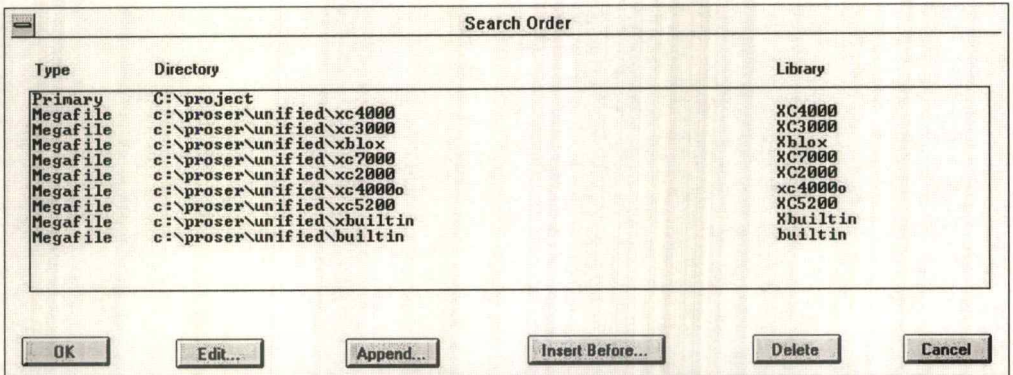
- การสร้างโปรเจกต์ไฟล์ ให้ไปที่เมนู Project Create ทำการคลิกเมาท์ เพื่อใส่ project Path Name ทำการพิมพ์ Project file



รูปที่ 3.15 การสร้าง Project file

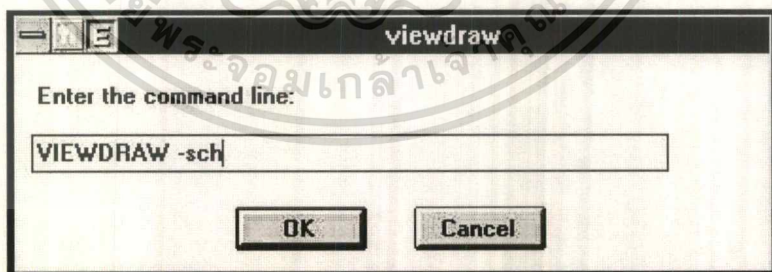
- การติดตั้ง Library เข้าไปที่เมนู Project Search Order ทำการคลิกเมาท์ เข้าไปที่ Append ให้คลิกเมาท์ที่ Type Mega จากนั้นใส่ Directroly ดังรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 การติดตั้ง Library

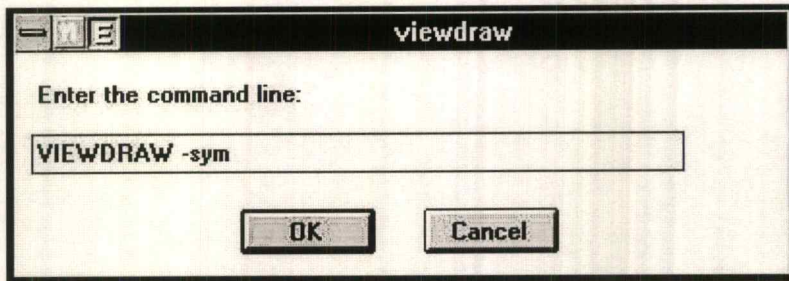
- การลบ Project File เข้าไปที่เมนู Project Delete Project ทำการคลิกเมาท์ จากนั้นดับเบิลคลิกที่ View Draw ในหน้าต่างของ Workview Plus Cockpit ทำการตั้งชื่อไฟล์ที่จะทำการสร้าง Schematic ดังรูปที่ 3.17



รูปที่ 3.17 การสร้าง Schematic

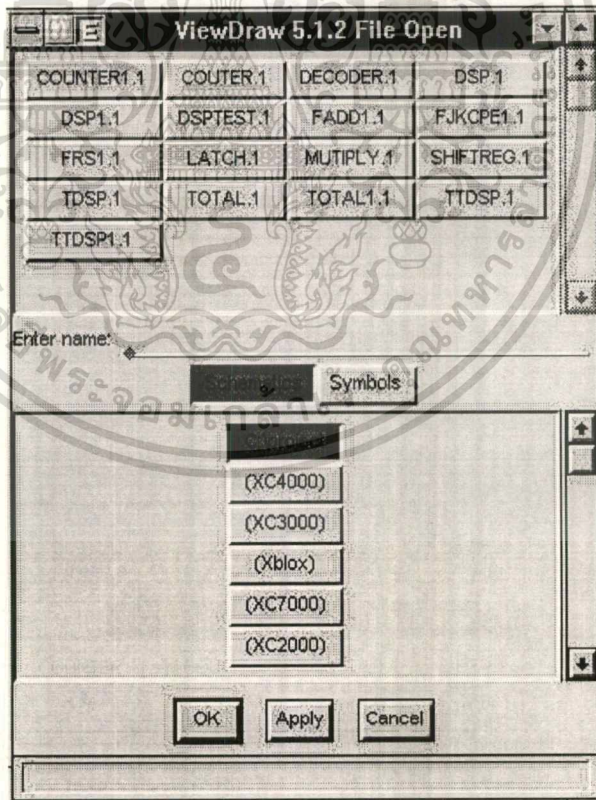
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การสร้าง Symbol ทำการพิมพ์ข้อความดังรูปที่ 3.18



รูปที่ 3.18 การสร้าง Symbol

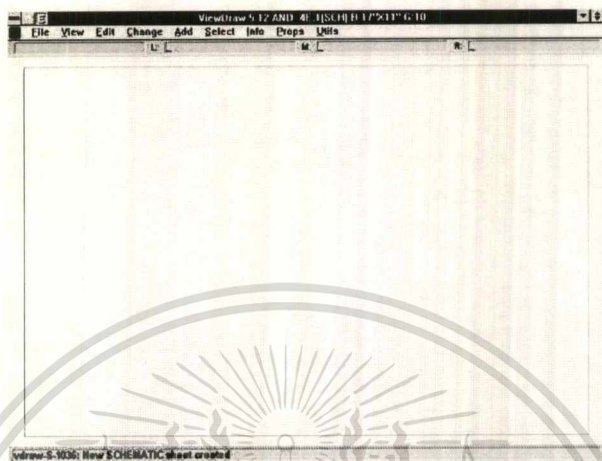
- ถ้าต้องการเปิดไฟล์ Schematic โดยไม่ทราบว่า มีไฟล์อะไรบ้างไม่ต้องทำการใส่ชื่อไฟล์จะ
แสดงหน้าจอ ดังรูปที่ 3.19



รูปที่ 3.19 หน้าต่าง View Draw / File Open

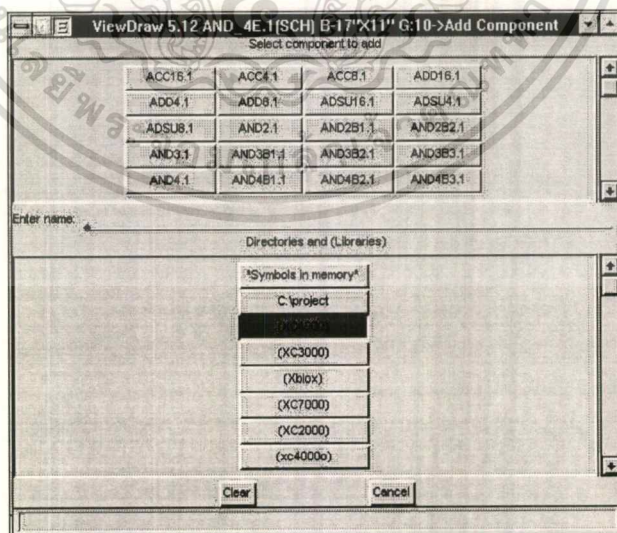
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อต้องการสร้างโปรแกรมใหม่ให้พิมพ์ชื่อไฟล์ที่ Enter Name คลิก OK จะแสดงหน้าต่าง View Draw



รูปที่ 3.20 หน้าต่าง View Draw

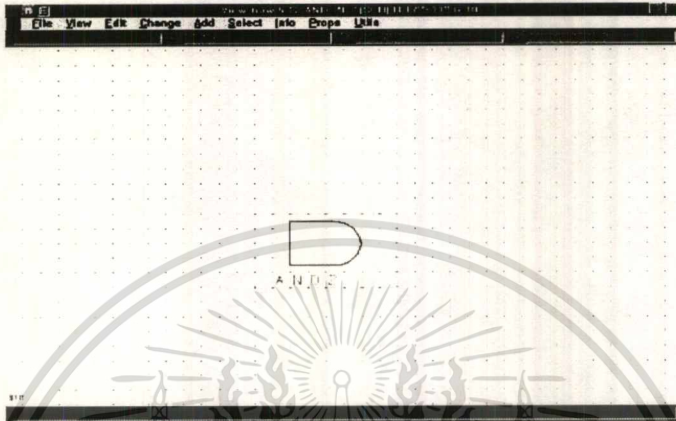
- การใส่ Component ให้ไปที่เมนู Add Component หรือใช้ Short Cut Key I จะปรากฏหน้าต่างดังรูปที่ 3.21



รูปที่ 3.21 การใส่ Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ให้เลือกตระกูลของไซลิ่งคี่ในที่นี้ใช้ตระกูล XC4000 ให้เลือก Component ที่ Select Component To Add ให้เลือก Component ชื่อ AND2.1 จากนั้นให้ไปที่ Sheet ของ View Draw แล้วกดปุ่ม F3 จะเป็นการวาง Component ที่เลือก



รูปที่ 3.22 Component ที่ถูกเลือก

- การคัดลอกส่วนประกอบ ให้ทำการคลิกที่อุปกรณ์ที่ต้องการคัดลอก แล้วกดที่ปุ่ม C เลื่อนเมาท์ไปยังตำแหน่งที่จะวางอุปกรณ์ เมื่อต้องการวางให้กดปุ่ม F3

- การเคลื่อนย้ายอุปกรณ์ ให้ทำการคลิกอุปกรณ์ที่ต้องการเคลื่อนย้าย จากนั้นให้กดปุ่ม m เลื่อนเมาท์ไปยังตำแหน่งที่ต้องการ เมื่อต้องการวางอุปกรณ์ให้กดปุ่ม F3

การลากเส้นสัญญาณมีอยู่ 2 ลักษณะคือ

1. การลากเส้น Wire หรือ Net ให้กดปุ่ม n ลากเมาท์ไปยังตำแหน่งที่ต้องการเชื่อมต่อเส้น Wire เข้ากับอุปกรณ์ตัวอื่น จากนั้นกดปุ่ม F3 เพื่อทำการวางเส้น Wire เมื่อสิ้นสุดตำแหน่งการวางเส้น Wire ให้ทำการคลิกเมาท์ทางขวา

2. การลากเส้นบัส ให้กดปุ่ม b ลากเมาท์ไปตำแหน่งที่ต้องการเชื่อมต่อเส้นบัสเข้ากับอุปกรณ์ตัวอื่น จากนั้นกดปุ่ม F3 เพื่อทำการวางอุปกรณ์ เมื่อสิ้นสุดตำแหน่งที่ต้องการให้ทำการคลิกเมาท์ทางขวา

- การตั้งชื่อสายสัญญาณต่าง ๆ ให้ไปที่ Net หรือ Bus แล้วกดปุ่ม 1 ทำการตั้งชื่อสายสัญญาณคลิก OK

- การสร้าง Symbol ให้ไปที่เมนู File Level Push Symbol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

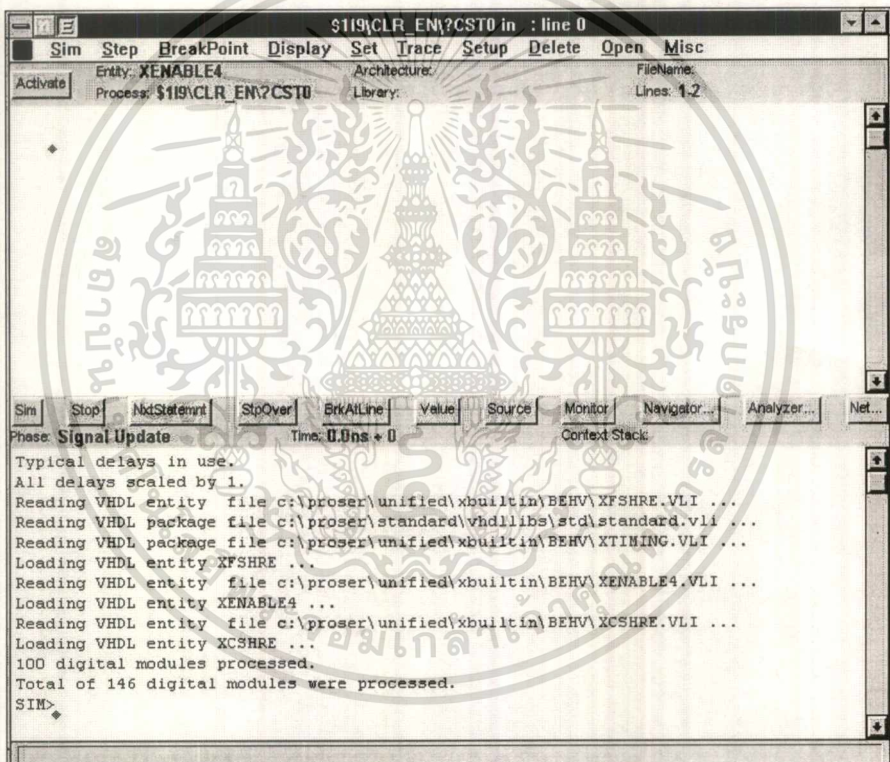
- การสร้าง Box ให้คลุมตัวอักษร B ลากเมาท์ไปที่จุดสิ้นสุดจากนั้นทำการกดปุ่ม F3 การสร้างขาสัญญาณ ให้คลุมตัวอักษร p ลากเมาท์ไปยังตำแหน่งสุดท้ายแล้วกดปุ่ม F3

- การกำหนดชื่อขาสัญญาณของ Symbol ต้องตั้งชื่อให้ตรงกับสัญญาณใน Schematic โดยการคลิกที่ขา (Pin) ที่ต้องการตั้งชื่อแล้วกดปุ่ม L ใส่ชื่อให้ตรงกับ Schematic เมื่อต้องการกลับไป Schematic ให้ไปที่ File Level Pop

หมายเหตุ เมื่อต้องการจะสร้างวงจรใหม่ต้องทำการ Save File เดิมก่อนโดยการกดปุ่ม W

- การสร้าง Netlist ให้ไปที่หน้าต่าง Cockpit เข้าไปที่ไอคอน VSM ให้ใส่ชื่อไฟล์คลิก OK

- การใช้งาน Viewsim ให้ไปที่หน้าต่าง Cockpit เข้าไปที่ไอคอน Viewsim



รูปที่ 3.23 การใช้ View Sim

- การกำหนด Clock ให้พิมพ์สัญญาณต่างดังรูปที่ 3.24

```

Sim Stop NbdStatemnt SpOver BriAllLine Value Source Monitor Navigator... Analyzer... Net...
Phase: Signal Update Time: 400.0ns + 1 Context Stack
Total of 146 digital modules were processed.
SIM>clock ck 0 1
SIM>h clr
SIM>h ce
SIM>vector in lin[7:0]
SIM>vector out lout[7:0]
SIM>a in 00001111
SIM>wave LATCH.wfm ck clr in out
SIM>sim
Simulation stopped at 100.0ns.
SIM>l clr
SIM>sim
Simulation stopped at 200.0ns.
SIM>cycle
  
```

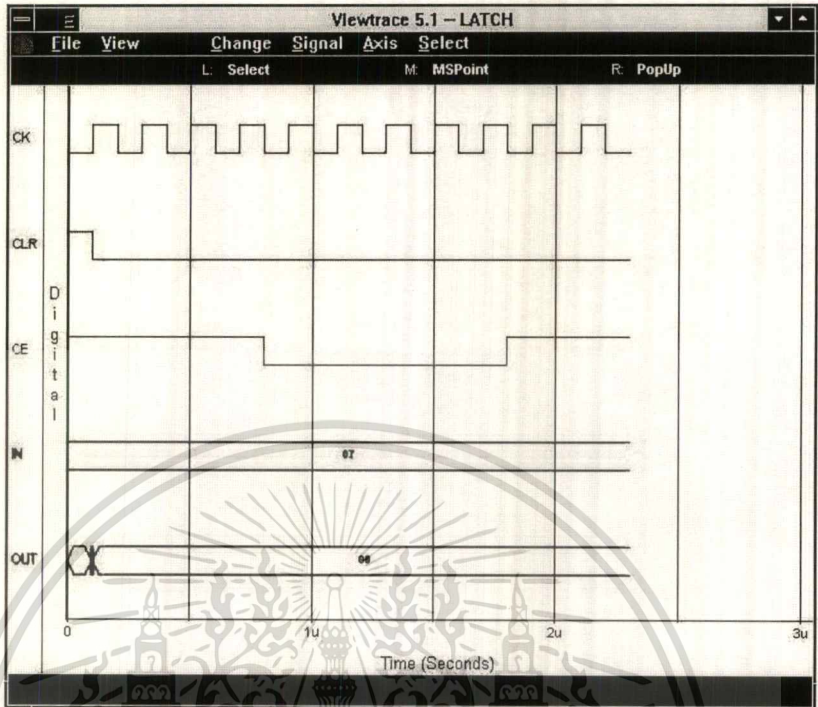
รูปที่ 3.24 การกำหนดการทำงานของขาสัญญาณ

-การใช้งาน โปรแกรม Viewtrace

-การกำหนด Waveform เพื่อแสดงการทำงานของวงจรจะกำหนดผ่านโปรแกรม Viewsim ที่เมนู Setting, wave แสดงหน้าต่างดังนี้



รูปที่ 3.25 หน้าต่างการกำหนดขาสัญญาณให้กับ โปรแกรม Viewtrace

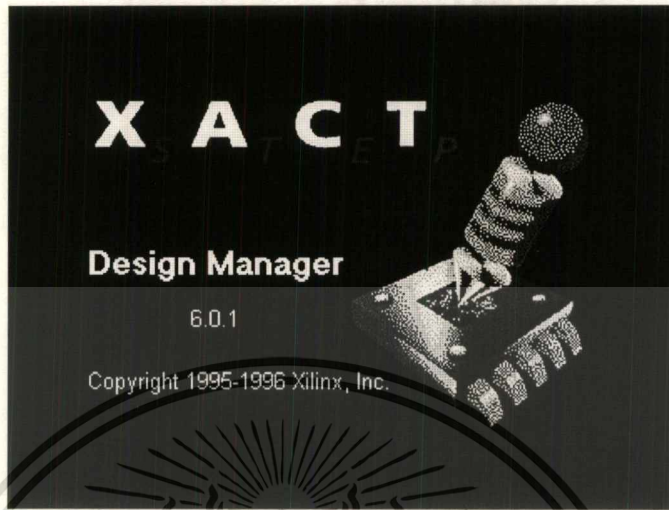


รูปที่ 3.26 หน้าต่างการทำงานของ Viewtrace เมื่อตั้งค่าสั่ง Sim หรือ Cycle แล้ว

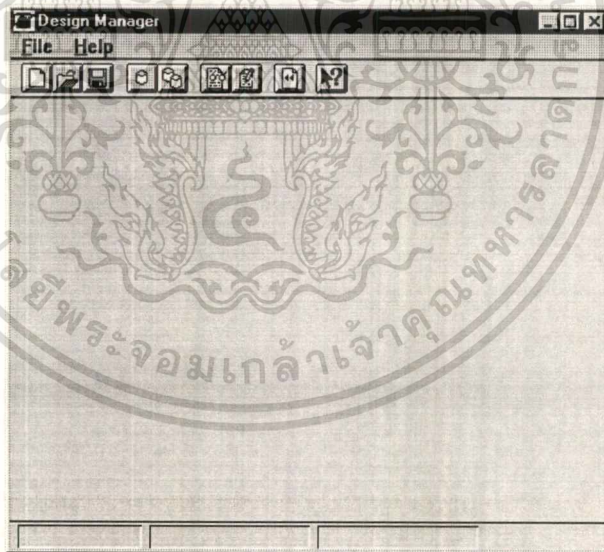
3.4.2 ขั้นตอนการโปรแกรมลงบนอุปกรณ์ FPGA โดยใช้ซอฟต์แวร์ XDM ของบริษัทไซลิ่งค์

เมื่อเริ่มเข้าสู่โปรแกรม XACT Design Manager (XDM) ที่หน้าจอจะแสดงรูปภาพ ดังรูปที่ 1 หลังจากนั้นจะเข้าสู่หน้าต่าง Design Manager ดังรูปที่ 3.27 ให้คลิกเมาส์ที่เมนู File และเลือกที่ New Project โดยจะแสดงหน้าต่างดังรูปที่ 3.28

คลิกเมาส์ที่ Browse บริเวณช่องอินพุต Design โดยให้ไปที่ไดเรกทอรีที่มีไฟล์ Fulladd.1 ซึ่ง เป็นวงจรระดับเกตของโปรแกรม Fulladd.vhd

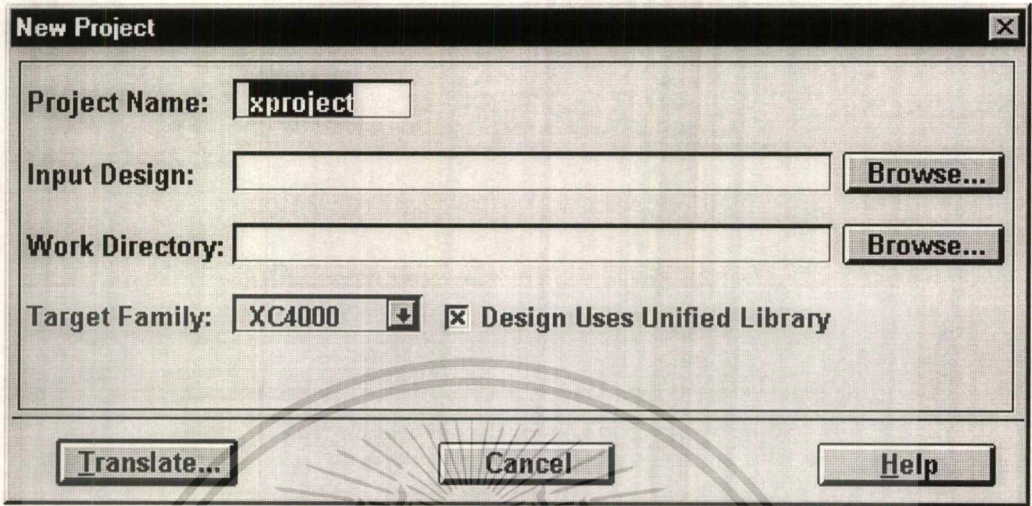


รูปที่ 3.27 หน้าจอเริ่มต้นของโปรแกรม XACT Design Manager

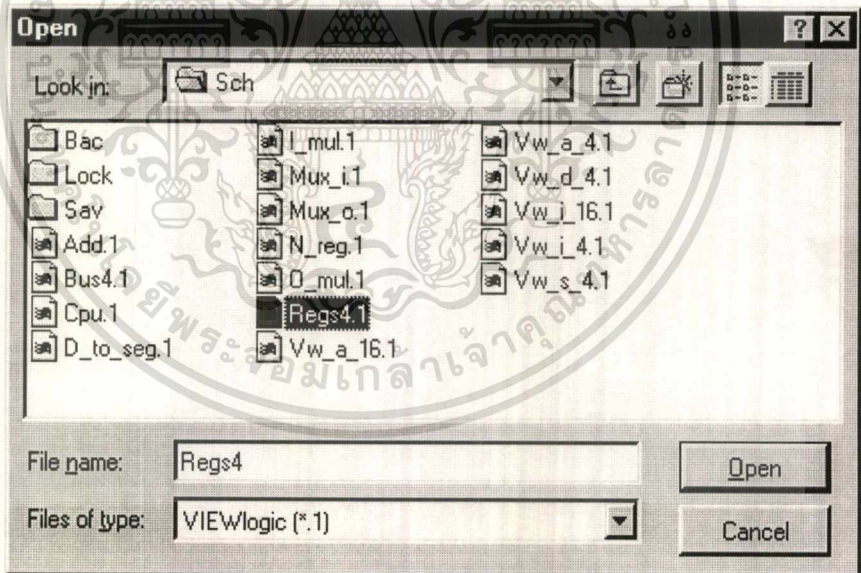


รูปที่ 3.28 หน้าต่าง Design Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



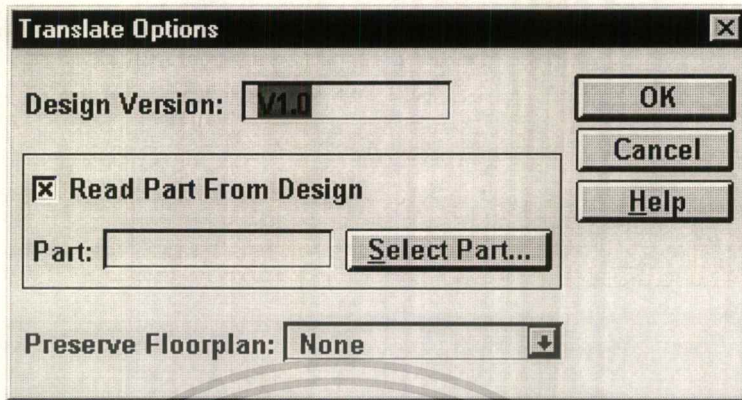
รูปที่ 3.29 หน้าต่าง New Project



รูปที่ 3.30 หน้าต่างอินพุต Design

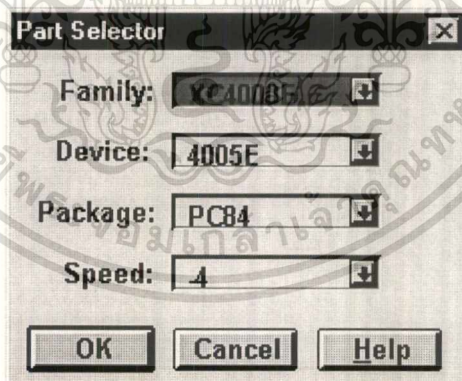
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิกเมาส์ที่ Translate จะแสดงหน้าต่าง Translate ดังรูปที่ 3.31



รูปที่ 3.31 หน้าต่าง Translate

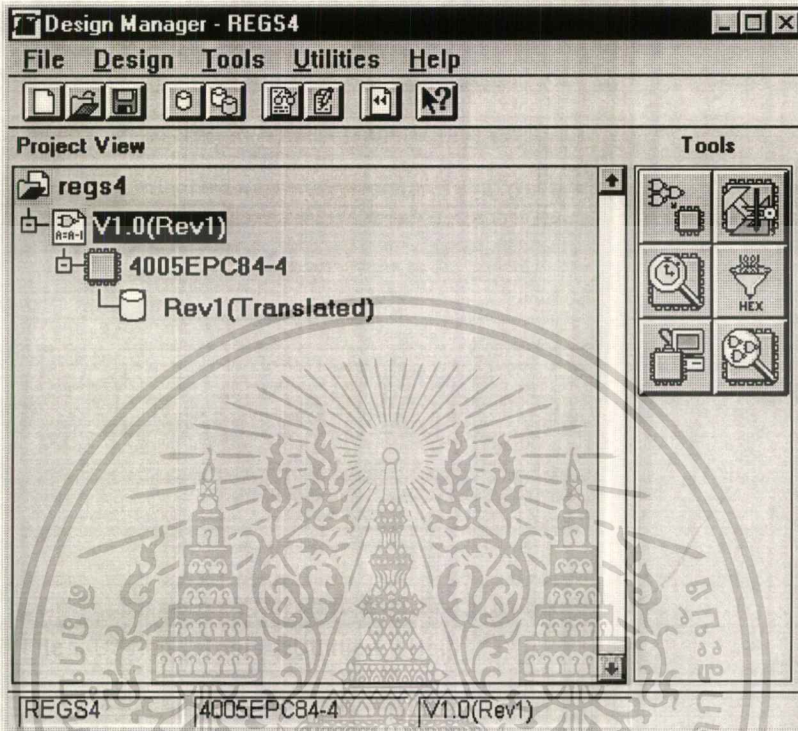
- คลิกเมาส์ที่ช่อง Real Part Design และที่ Select part ซึ่งจะแสดงหน้าต่างของ Part Selector และเปลี่ยน Speed จาก 4 เป็น 6 ซึ่งเป็นเบอร์ของไอซี FPGA เบอร์ XC4005APC84C-6 ที่ใช้ในบอร์ดตัวอย่างของ FPGA ดังรูปที่ 3.32



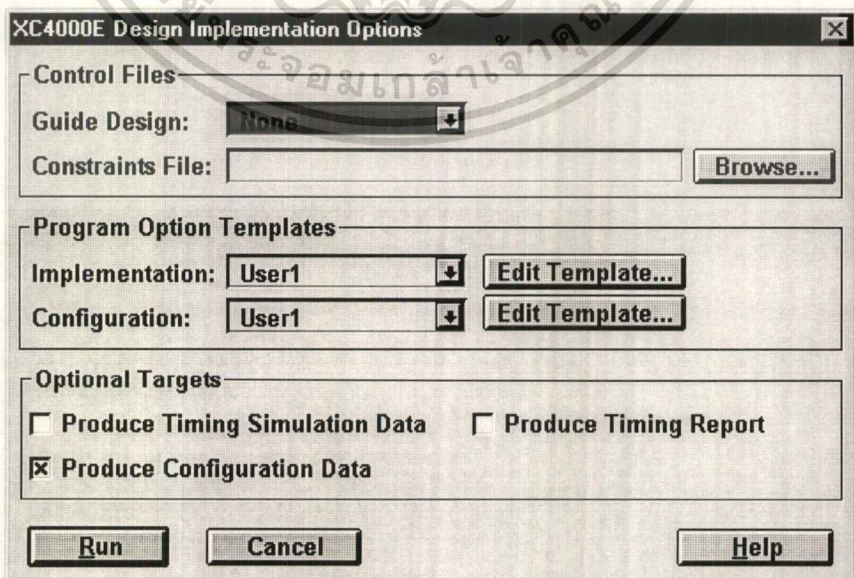
รูปที่ 3.32 หน้าต่าง Part Selector

- เมื่อทำการกำหนดค่าต่าง ๆ แล้วโปรแกรมจะทำการ Translate ให้เมื่อเสร็จการ Translate แล้ว จะแสดงหน้าต่างดังรูปที่ 3.32 หลังจากนั้นให้คลิกที่เมนู Design เสร็จแล้วเลือกคำสั่ง Implement จะแสดงหน้าต่าง XC4000 Design Implementation Option ดังรูปที่ 3.34 ที่ช่อง Option Targets จะให้เลือกการรายงานผลการ Run โดยให้เลือกรายงานผลทุกอย่าง เสร็จแล้วคลิกเมาส์ที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Run ซึ่งเป็นกระบวนการ Design จากวงจรระดับเกตของไฟล์ Fulladd.1 ให้เป็นไฟล์บิตสตรีม (Bit Stream) เพื่อใช้ดาวน์โหลดลงบนอุปกรณ์ FPGA_s ดังรูปที่ 3.35



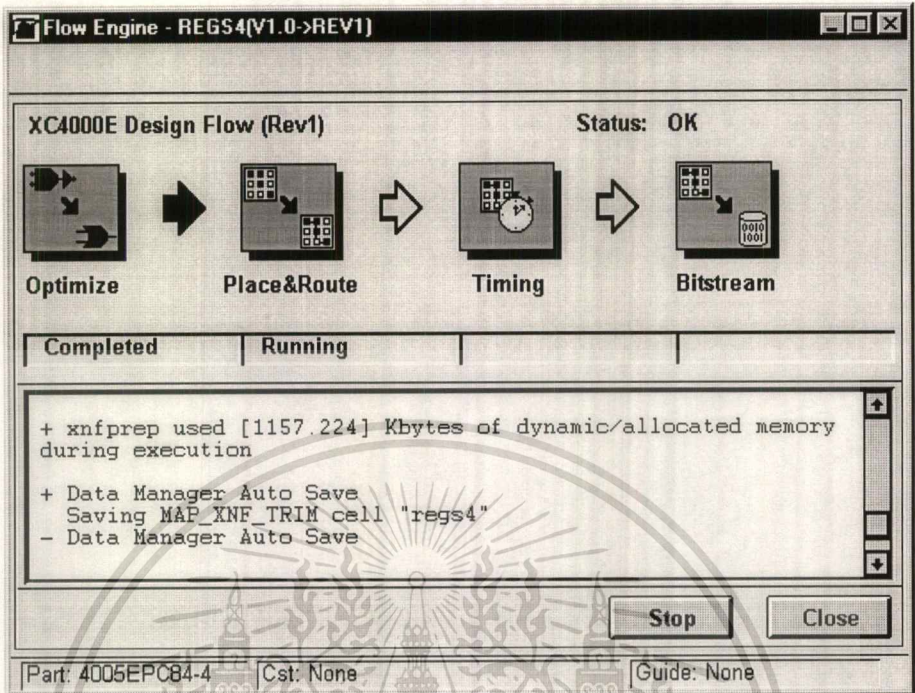
รูปที่ 3.33 ผลการ Translate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

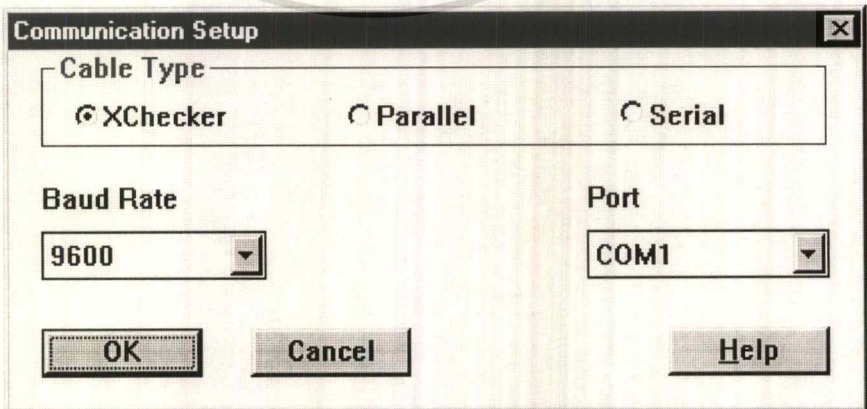
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกหนึ่งสิ่งที่มีค่าคือ การเรียนรู้และทำความเข้าใจกับเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.34 หน้าต่าง XC4000 Design Implementation Option



รูปที่ 3.35 หน้าต่างกระบวนการสร้างไฟล์บิตสตรีม

- เมื่อเสร็จกระบวนการแล้วจะได้ไฟล์ที่มีนามสกุล .bit เพื่อความถี่โหลดลงบนบอร์ดตัวอย่างของ FPGA_s โดยในการดาวน์โหลดไฟล์บิตสตรีมลงบนบอร์ดตัวอย่างของ FPGA_s นั้นให้คลิกเมาส์ที่เมนู Tools และเลือกที่ Hardware Debugger ซึ่งจะแสดงหน้าต่าง Communication ดังรูปที่ 3.36



รูปที่ 3.36 หน้าต่าง Communication Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับความรู้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อทำการกำหนดค่า Baud Rate และเลือก Port ที่จะส่งข้อมูลออกไปแล้วให้คลิกเมาส์ที่ OK เมื่อเสร็จแล้วให้ไปที่หน้าต่าง Hardware Debugger คลิกที่เมนู Download และเลือกที่ Download Design ซึ่งโปรแกรม Debugger จะทำการ Download Design ไฟล์ Fulladd.bit ลงบอร์ดตัวอย่างของ FPGA_s ผ่านทางสายคาวีโหลดเคเบิล เมื่อคาวีโหลดเสร็จเรียบร้อยแล้วจึงนำลงบอร์ดตัวอย่างของ FPGA_s ไปทดสอบการทำงานของวงจร Fulladder



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้กล่าวถึงการทดลองและผลการทดลอง การทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกและวงจรองความถี่เชิงเลขแบบจำกัดผลตอบอันดับ 10 โดยใช้อุปกรณ์ FPGA_s

4.1 การทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

4.1.1 ลำดับขั้นตอนทดลอง

1. ต่อวงจรตามรูปที่ 4.1
2. ป้อนความถี่ 8 กิโลเฮิร์ตซ์ให้กับวงจร
3. ต่อแหล่งจ่ายไฟที่สามารถปรับค่าได้ ตั้งแต่ -10 โวลต์ จนถึง +10 โวลต์
4. ทำการปรับแหล่งจ่ายไฟ ตามค่าที่กำหนดลงในตารางที่ 4.1
5. วัดค่าเอาต์พุตที่ได้จากผลการทดลอง

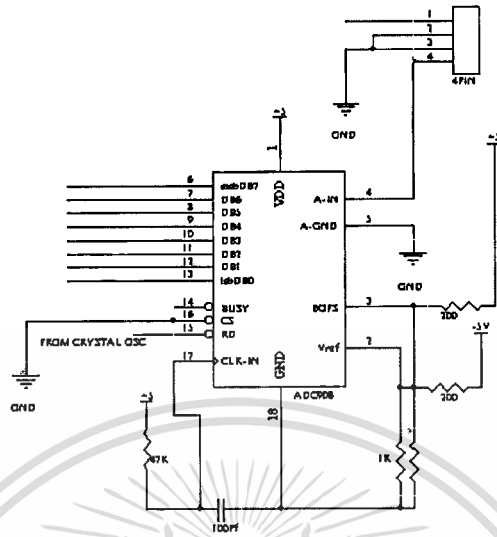
ตารางที่ 4.1 ผลการทดลองวงจร A/D

อินพุต (โวลต์)	เอาต์พุต (ตัวแสดงผล 7 ส่วน)
0	87
1	A3
2	BD
3	CF
4	FF
5	FF

4.1.2 สรุปผลการทดลอง

จากผลการทดลองจะพบว่า เมื่อป้อนสัญญาณอินพุตที่เป็นสัญญาณแอนะล็อก เข้าไปในวงจร A/D ภายในวงจร A/D จะทำการแปลงสัญญาณแอนะล็อกที่ได้ออกมาเป็นสัญญาณดิจิทัล ดัง

ค่าที่ได้ในตารางที่ 4.1 และรูปที่ 4.2 งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

รูปที่ 4.2 ผลการทดลองวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

4.2.1 ลำดับขั้นการทดลอง

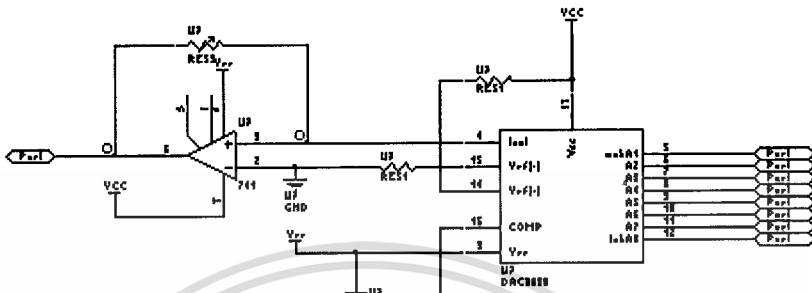
1. ต่อดังวงจรตามรูปที่ 4.3
2. ป้อนแหล่งจ่ายไฟ ± 5 โวลต์ ให้กับวงจร D/A
3. ปรับค่าอินพุตที่ป้อนให้วงจร D/A ตามตารางที่ 4.2
4. วัดค่าเอาต์พุต จากขา 6 ของ ออปแอมป์ เบอร์ 741

ตารางที่ 4.2 ผลการทดลองวงจร D/A

อินพุต	เอาต์พุต (มิลลิเมตร)
00	0
01	0.2
02	0.4
10	0.8
20	1.65
40	3.2
FF	4.2

4.2.2 สรุปผลการทดลอง

จากผลการทดลองจะพบว่าเมื่อป้อนสัญญาณอินพุตที่เป็นสัญญาณดิจิทัลเข้าไปในวงจร D/A ภายในวงจร D/A จะทำการแปลงสัญญาณดิจิทัลที่เข้ามาออกไปเป็นสัญญาณแอนะล็อกตามค่าที่ได้ในตารางที่ 4.2 และรูปที่ 4.4



รูปที่ 4.3 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

รูปที่ 4.4 ผลการทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสำหรับโครงการวิจัยที่สนับสนุนโดยศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานคณะกรรมการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบแบบจำลองจริงบนโปรแกรม Viewtrace

4.3.1 การกำหนดขาสัญญาณของวงจรทั้งหมดดังนี้

ลำดับขั้นการทดลอง

1. กำหนดบิตข้อมูลของ A/D เป็น Vector ขนาด 8 บิต
2. กำหนดบิตข้อมูลของ EPROM เป็น Vector ขนาด 8 บิต
3. กำหนดบิตข้อมูลของ D/A เป็น Vector ขนาด 16 บิต
4. กำหนดสัญญาณนาฬิกา
5. กำหนดสถานะของสัญญาณเคลียร์
6. กำหนดข้อมูล A/D
7. กำหนดข้อมูลของ EPROM

```

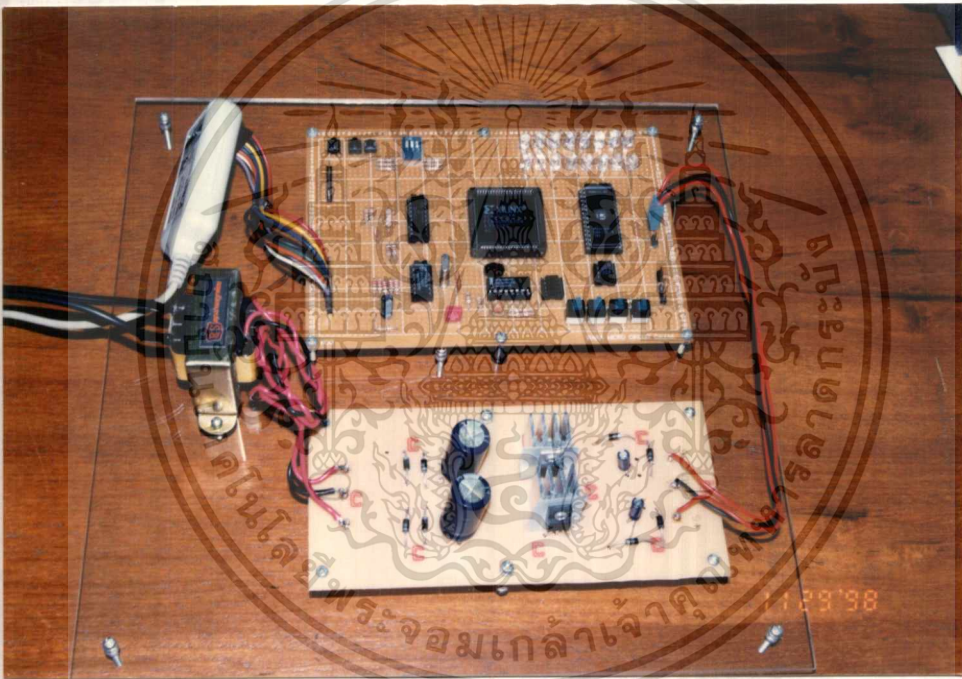
Sim Stop Next Statement Stop Over Break At Line Value Source Monitor Navigator... Analyzer... Net...
Phase: Signal Update Time: 300.0ns + 1 Context Stack
User attributes were detected. They are listed in vsimtemp.log.
SIM>vector adc adc[7:0]
SIM>vector dat dat[7:0]
SIM>vector dac dac[7:0]
SIM>l clr
SIM>sim
Simulation stopped at 100.0ns.
SIM>h clr
SIM>a adc 00100101
SIM>a dat 10010000
SIM>s
Simulation stopped at 200.0ns.
SIM>wave FIR5OR.wfm clr adc dat dac
SIM>a dat 10101100
  
```

รูปที่ 4.5 การกำหนดขาสัญญาณบน View Sim

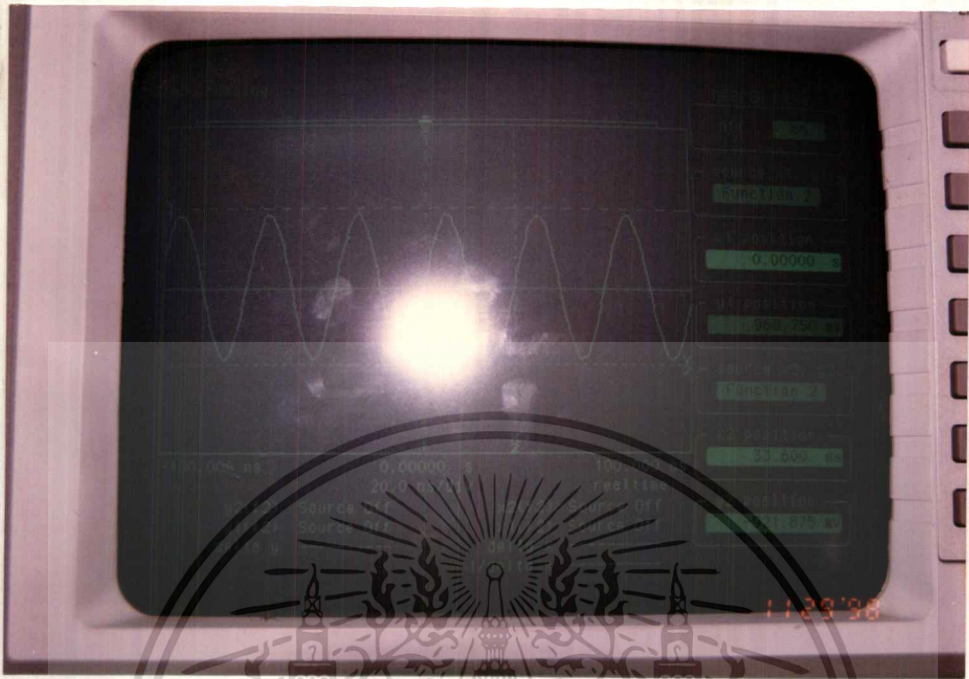
4.4 การทดสอบการทำงานของบอร์ดวงจรความถี่เชิงเลข

ลำดับขั้นการทดสอบ

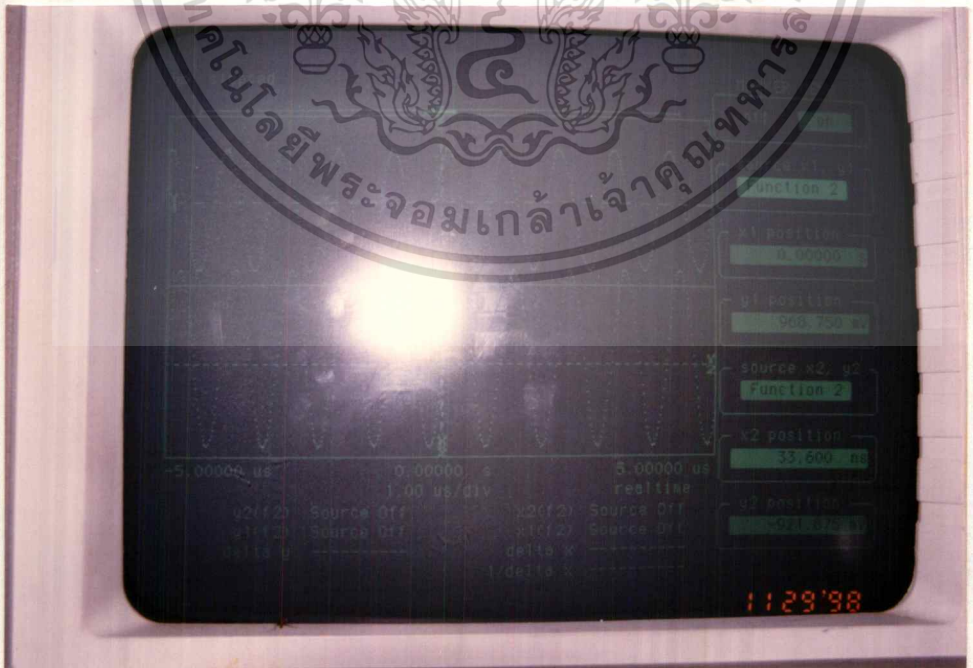
1. ป้อนสัญญาณทางอินพุต รูปสัญญาณที่มีความถี่ 10 กิโลเฮิร์ตซ์ 2.5 Vp-p ที่ A/D
2. เก็บข้อมูลลงใน EPROM ที่ต้องการจะกำหนดความถี่ผ่าน
3. ใช้ออสซิลโลสโคปวัดสัญญาณสุ่ม 8 กิโลเฮิร์ตซ์ จาก FPGA_s ที่ขาสัญญาณ S1
4. วัดแรงดันทางด้านเอาต์พุตช่วงความถี่ที่ได้
5. เปลี่ยนย่านความถี่โดยการเปลี่ยนแปลงข้อมูลของ EPROM



รูปที่ 4.7 บอร์ดวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้
อุปกรณ์ FPGA_s

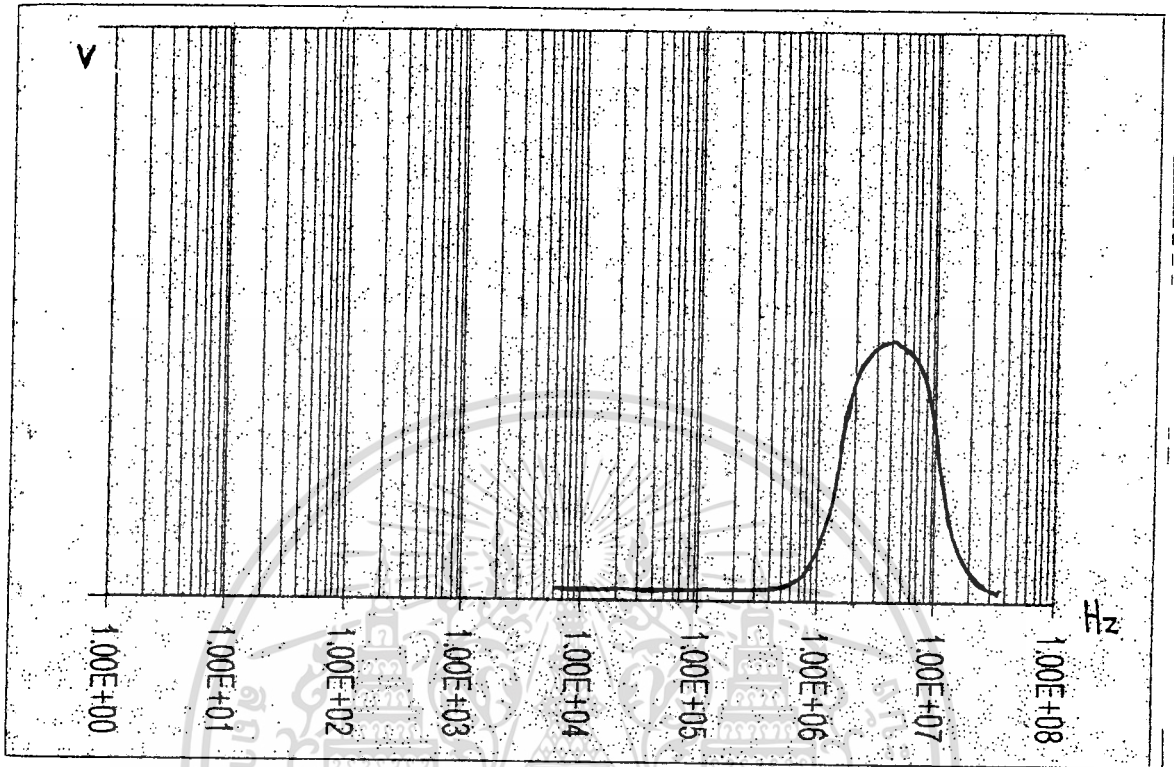


รูปที่ 4.8 รูปคลื่นแรงดันทางด้านอินพุต



รูปที่ 4.9 ผลตอบสนองความถี่ทางด้านเอาต์พุตเมื่อกำหนดช่วงความถี่ต่ำ 0-1 กิโลเฮิร์ตซ์

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟช่วงการคัตออฟความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

5.1 สรุป

วงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 โดยใช้อุปกรณ์ FPGA_s โดยการนำเอาหลักการของการประมวลผลเชิงเลขมาใช้ในการออกแบบ ซึ่งลักษณะของการทำงานของวงจรกรองเชิงเลขนี้จะมีลักษณะเหมือนกับวงจรกรองในระบบแอนะล็อกเพียงแต่ใช้การปรับค่าความถี่ด้วยหลักการของการประมวลผลเชิงเลขแทนอุปกรณ์ในระบบแอนะล็อกเท่านั้น

ผลการทดสอบการทำงานของวงจรกรองความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 ทำได้โดยการดาวน์โหลดโปรแกรมลงบนอุปกรณ์ FPGA_s ได้ผลการทดลองใกล้เคียงกับการออกแบบแต่มีข้อผิดพลาดคือความถี่ที่ได้สูงกว่าความถี่คัตออฟ ความชันของระบบจะน้อยกว่าที่ได้ทำการออกแบบไว้ ซึ่งความถี่คัตออฟนั้นสามารถเปลี่ยนแปลงค่าได้โดยการเปลี่ยนค่าสัมประสิทธิ์ของตัวกรองสัญญาณเชิงเลขที่เก็บไว้ใน EPROM ในการทดลองนี้ใช้ความถี่คัตออฟที่ 0-1 กิโลเฮิร์ตซ์

5.2 ปัญหาที่พบ

5.2.1 ในส่วนของฮาร์ดแวร์

ปัญหา ในการออกแบบไม่สามารถใช้เกตในไอซี FPGA_s ได้ตัวเดียวจำเป็นต้องใช้ชิพ FPGA_s มากกว่า 1 ตัวขึ้นไปจึงจะสามารถออกแบบได้ครบ 10 อันดับ

แนวทางแก้ไข ลดอันดับของวงจรกรองความถี่เชิงเลขลงให้เหลือ 5 อันดับจึงสามารถบรรจุลงในชิพ FPGA_s ได้

5.2.2 ในส่วนของซอฟต์แวร์

ปัญหา เครื่องมือมีจำนวนจำกัด การใช้จะใช้เครื่องมือในเวอร์ชันเก่าไม่มีไลบรารีบางตัวใน FPGA_s คือ XC4000E แต่ชิพที่ใช้เป็นชิพเบอร์ XC4010E

แนวทางแก้ไข เปลี่ยนเครื่องมือเป็นเวอร์ชันที่มีไลบรารีตระกูล XC4000E ทำการเปลี่ยนคอมโพเนนท์ที่ออกแบบใน View Draw เป็นไลบรารีของ XC4000E

5.2.3 ในส่วนการออกแบบ

เอกสารนี้เป็นเอกสารในการออกแบบวงจรและวงจรบวกเลข เกตที่ได้ในการออกแบบมีจำนวนมากเกินไป ทำให้ไม่เพียงพอกับจำนวนเกตใน FPGA_s และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางแก้ไข ในการออกแบบวงจรคูณและวงจรวกเลขมีอยู่ด้วยกัน 2 วิธี แต่จำเป็นต้องเลือกวิธีที่ประหยัดเกตให้มากขึ้นจึงทำให้การออกแบบไม่สามารถใช้วิธีแบบอัลกอริทึมได้ เนื่องจากการใช้วิธีอัลกอริทึมจะมีขนาดใหญ่กว่าวิธีการออกแบบโดยตรง

5.3 การพัฒนา

5.3.1 ในส่วนของฮาร์ดแวร์

- ในการออกแบบวงจรความถี่เชิงเลขแบบจำกัดผลตอบสนองอันดับ 10 จำเป็นต้องใช้การต่ออุปกรณ์ FPGA มากกว่า 1 ตัว

- ในทางด้านความถี่สูงควรมีการปรับเปลี่ยนได้ อาจจะใช้ซอฟต์แวร์แทน

5.3.2 การพัฒนาการออกแบบ

- ถ้าเป็นอุปกรณ์ที่ไม่ใช้อุปกรณ์ FPGA ควรจะนำการออกแบบที่มีอัลกอริทึมมาใช้งานเนื่องจากมีความเร็วในการประมวลผลมากกว่า

- การออกแบบวงจร FIR ที่มีอันดับมาก ๆ ควรมีทั้งฮาร์ดแวร์และซอฟต์แวร์ทำงานร่วมกันให้มากขึ้น เพื่อลดจำนวนเกตของ FPGA ลง

- การออกแบบควรคิดการประมวลผลแบบจุดทศนิยมและคิดเครื่องหมาย

5.3.3 การพัฒนาซอฟต์แวร์

- การนำค่าสัมประสิทธิ์ที่คำนวณ อาจจะใช้คอมพิวเตอร์ช่วยการคำนวณและสามารถติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตขนานหรืออนุกรมตามมาตรฐาน RS232 เพื่อความสะดวกแทนการเก็บค่าลงใน EPROM ซึ่งการเปลี่ยนแปลงค่าในโครงการนี้จะต้องลบข้อมูลที่เป็นสัมประสิทธิ์เดิมออกก่อนแล้วจึงบรรจุสัมประสิทธิ์ที่ความถี่คัดออกใหม่แทน



ภาคผนวก ก
รายละเอียดอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

DAC-08

HIGH SPEED
3-BIT MULTIPLYING D-TO-A
CONVERTER

SILICON MONOLITHIC
INTEGRATED CIRCUIT

HIGH SPEED 3-BIT MULTIPLYING D-TO-A CONVERTER

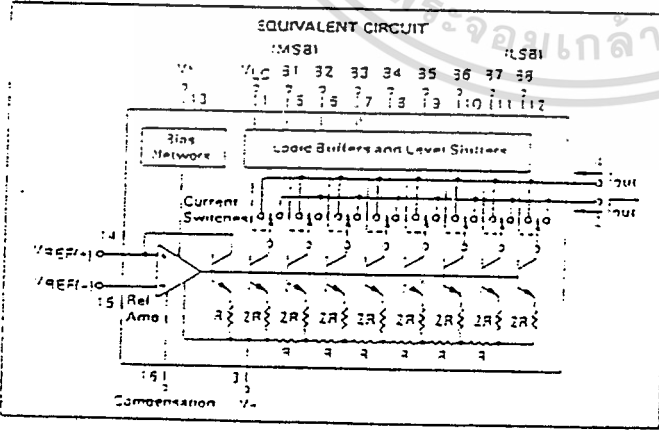
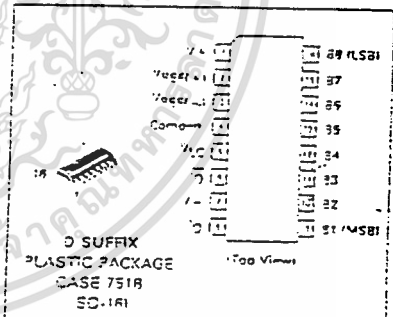
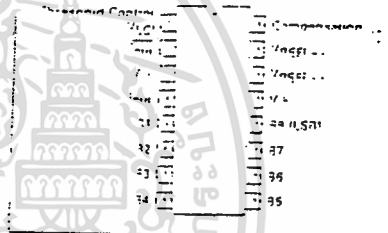
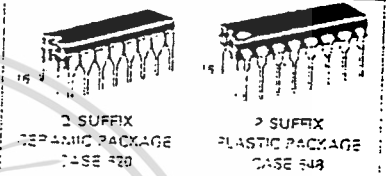
The DAC-08 series is a monolithic 3-bit high speed multiplying digital-to-analog converter, capable of settling to within 1/2 LSB (0.19%) in 35 ns. Monotonic multiplying performance is retained over a wide 40-to-1 reference current range. Full scale and reference currents are matched to within 1 LSB, therefore eliminating the need for full scale trim in most applications.

Dual complementary current outputs with high voltage compliance provide added versatility and allow differential mode of operation to effectively double the peak-to-peak output swing. In many applications, output current-to-voltage conversion can be accomplished without requiring an external op amp. Noise-immune inputs permit direct interface with TTL and DTL levels when the logic threshold control, V_{TC} (Pin 1) is grounded. All other logic family thresholds are attainable by adjusting the voltage level of Pin 1. Performance characteristics are virtually unchanged over the entire ± 4.5 V to ± 18 V power supply range. Power consumption is typically 33 mW with ± 5.0 V supplies.

The DAC-08 is available in several versions, with nonlinearity as low as $\pm 0.1\%$ ($\pm 1/4$ LSB) over temperature. All versions are guaranteed monotonic over 3 bits. For an extra margin of performance, Motorola utilizes thin-film resistors permitting very accurate resistive values which are extremely stable over temperature.

High performance characteristics along with low cost, make the DAC-08 an excellent selection for applications such as CRT displays, waveform generation, high speed modems, and high speed analog to digital converters.

- Fast Settling Time — 35 ns
- Full Scale Current Matched to ± 1 LSB
- Nonlinearity Over Temperature to $\pm 0.1\%$ Max
- Differential Current Outputs
- High Voltage Compliance Outputs — 10 V to ± 18 V
- Wide Range Multiplying Capability
- Inputs Compatible With TTL, DTL, CMOS, PMOS, ECL, HTL
- Low Full Scale Current Drain
- Wide Power Supply Range ± 4.5 V to ± 18 V
- Low Power Consumption
- Thin-Film Resistors
- Low Cost



ORDERING INFORMATION

Device	Nonlinearity	Temperature Range	Package
DAC-08HQ	$\pm 0.1\%$		Ceramic
DAC-08EQ	$\pm 0.19\%$		Ceramic
DAC-08CD	$\pm 0.25\%$		SO-16
DAC-08ED	$\pm 0.19\%$	70°C to 110°C	SO-16
DAC-08MP	$\pm 0.1\%$		Plastic
DAC-08EP	$\pm 0.19\%$		Plastic
DAC-08CP	$\pm 0.25\%$		Plastic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

DAC-08

MAXIMUM-RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
V _S Supply to V ₊ Supply	V _S	16	V
Logic Inputs	—	V ₊ to V ₊ Plus 3σ	V
Logic Threshold Control	V _{TC}	V ₊ to V ₊	V
Analog Current Outputs	I _{OUT}	See Figure 7	mA
Reference Input (V14, v15)	V _{REF}	V ₊ to V ₊	V
Reference Input Differential Voltage (v14 to v15)	V _{DIFF}	±18	V
Reference Input Current (I14)	I _{REF}	±3.0	mA
Operating Temperature Range	T _A	0 to 70	°C
Storage Temperature	T _{STG}	-15 to +150	°C
Power Dissipation Derate above 100°C	P _D	500	mW
	P _{SD}	10	mW/°C

ELECTRICAL CHARACTERISTICS (V_S = ±15 V, I_{REF} = 2.0 mA, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	DAC-08H			DAC-08E			DAC-08C			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Resolution	—	1	1	1	1	1	1	1	1	1	bits
Monotonicity	—	1	1	1	1	1	1	1	1	1	bits
Nonlinearity, T _A = 0°C to 70°C	INL	—	—	±0.1	—	—	±0.15	—	—	±0.15	%FS
Settling Time to ±1/2 LSB All bits Switched On or Off T _A = 25°C (Figure 24 (Note 1))	t _S	—	15	—	15	—	—	—	15	—	ns
Propagation Delay, T _A = 25°C (Note 1) Each bit All bits Switched	t _{PLH} t _{PLL}	—	15	—	15	—	—	15	—	—	ns
Full Scale Tempco	FS _{Temp}	—	±10	—	±10	—	—	±10	—	—	ppm/°C
Output Voltage Compliance Full Scale Current Change ±1/2 LSB, R _{OUT} = 20 MΩ Typ.	V _{OC}	±10	—	±10	±10	±10	±10	±10	±10	±10	V
Full Range Current V _{REF} = 10,000 V A14, A15 = 5,000 mV T _A = 25°C	I _{RA}	1,364	1,352	1,300	1,34	1,39	1,34	1,34	1,39	1,34	mA
Full Range Symmetry (IC _{Q4} - IC _{Q2})	FS _S	—	±0.5	±1.0	—	±1.0	±1.0	—	±1.0	±16.0	μA
Zero Scale Current	IS	—	±1	±3	—	±2	±3	—	±2	±3	μA
Output Current Range V ₊ = +3.0 V V ₊ = -3.0 V to -10 V	I _{OH} I _{OL}	0 0	—	2.1 4.2	3 9	—	2.1 4.2	3 9	—	2.1 4.2	mA
Logic Input Levels (V _{IC} = 3 V) Logic "0" Logic "1"	V _{IL} V _{IH}	— 1.0	—	2.3	—	—	1.8	—	—	1.8	V
Logic Input Levels (V _{IC} = 3 V) Logic "0" V _{IL} = +10 V to +3.9 V Logic "1" V _{IH} = -2.0 V to -10 V	V _{IL} V _{IH}	— —	-3.0 -10	-10	—	-2.0	-10	—	-2.0	-10	V
Logic Input Swing, V ₊ = ±15 V	V _{IS}	±10	—	±10	±10	±10	±10	±10	±10	±10	V
Logic Threshold Range, V _{TC} = ±15 V	V _{THR}	±10	—	±13.5	±10	—	±13.5	±10	—	±13.5	V
Reference Bias Current	I _{IS}	—	±1.0	±2.0	—	±1.0	±1.0	—	±1.0	±2.0	μA
Reference Input Slow Rate Figure 19 (Note 1)	IS _{OUT}	—	3.0	—	3.0	—	—	3.0	—	—	mA/μs
Power Supply Sensitivity I _{REF} = 1.0 mA V ₊ = +3 V to 15 V V ₊ = -3 V to -15 V	PSS _{Leg} PSS _{Seg}	—	±0.0003 ±0.002	±0.01	—	±0.0003 ±0.002	±0.01	—	±0.0003 ±0.002	±0.01	%
Power Supply Current V _S = ±10 V, I _{REF} = 1.0 mA	I _S	—	1.3	1.3	—	1.3	1.3	—	1.3	1.3	mA
V _S = ±5.0 V, V ₊ = 15 V, I _{REF} = 2.0 mA	I _S	—	1.1	1.2	—	1.4	1.5	—	1.4	1.3	mA
V _S = ±15 V, I _{REF} = 2.0 mA	I _S	—	1.5	1.3	—	1.5	1.3	—	1.5	1.3	mA
Power Dissipation V _S = ±10 V, I _{REF} = 1.0 mA V _S = ±5.0 V, V ₊ = 15 V, I _{REF} = 2.0 mA V _S = ±15 V, I _{REF} = 2.0 mA	P _D	—	13 128 125	13 125 74	—	13 128 125	13 125 74	—	13 128 125	13 125 74	mW

Note 1. Parameter is not 100% tested; guaranteed by design.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Design Migration from XC4000 to XC4000E

XAPP 062 October 15, 1996 (Version 1.0)

Application Note by Lois Carter and Marc Baker

Summary

The XC4000E is an enhanced architecture based on the XC4000 family, but offers many new features, particularly Select-RAM™ memory. When converting XC4000, XC4000A, XC4000D, and XC4000H designs, the XC4000E is an excellent choice. The conversion process may be as simple as downloading the same bitstream into the XC4000E device (XC4000 and XC4000D bitstreams only), or it may involve changes to the schematic or HDL code. This Application Note describes techniques that should be employed to convert from any of the XC4000, XC4000A, XC4000D, or XC4000H families to the XC4000E family.

Xilinx Family

XC4000E

Demonstrates

Techniques for migrating XC4000 designs to the XC4000E architecture

Introduction

Some variations of the XC4000 family are no longer recommended for new designs. These include the XC4000 standard family, XC4000A, XC4000D, and XC4000H (referred to in this application note as the older XC4000 families). These have been superseded by the newer XC4000 Series families, which include the XC4000E, XC4000L, XC4000EX, and XC4000XL families. Designs in the older versions of the XC4000 can be converted to the new XC4000 Series. In some cases, particularly for XC4000D designs or other designs not including RAM, the XC5200 family should be considered for its lower cost. For a discussion of converting XC4000 family designs to the XC5200 family, see the Xilinx application note XAPP060, "Design Migration from XC4000 to XC5200."

Converting an older XC4000 design to an XC4000E design may or may not require schematic changes. Except for certain I/O capabilities, the XC4000E architecture offers a superset of the architectural features in the older XC4000 families, in addition to many new enhancements. (See Table 1.) Therefore, if none of the special I/O features in the XC4000A or XC4000H are used, the conversion becomes merely a matter of selecting the XC4000E speed grade that will give comparable performance, and recompiling the design targeting the selected device. If retargeting an XC4000 or XC4000D design, in fact, the XC4000E can actually accept the original bitstream. XC4000A and XC4000H designs must be recompiled.

Table 1: Comparison of Features in XC4000 Families

	XC4000	XC4000A	XC4000D	XC4000H	XC4000E
Output Drive/Pin (mA)	12	24	12	4/24	12
Output Slew Rate Options	Fast, Slow	Fast, Slow, Medium Fast, Medium Slow	Fast, Slow	SoftEdge/Resistive load	Fast, Slow
RAM	Yes	Yes	No	Yes	Yes
I/O Registers	Yes	Yes	Yes	No	Yes
Input Thresholds	TTL only	TTL only	TTL only	TTL or CMOS, per I/O	TTL or CMOS, Global
Output High Levels	TTL only	TTL only	TTL only	TTL or CMOS, per I/O	TTL or CMOS, Global
Wide Edge Decoders (per edge)	4	2	4	4	4
Speed Grades	-3, -5, -4	-3, -5, -4	-3, -5, -4	-3, -5	-4, -3, -2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Enhancements

The XC4000E offers a number of architectural improvements to the XC4000. Although the simplest conversion path uses only the resources provided in the originally targeted older XC4000 device, a design can be altered to take advantage of these features if desired.

- **PCI Compliance:** Fully compliant for -3 speed grades and faster.
- **Select-RAM Memory:** In addition to the XC4000 modes, RAM can now be configured for synchronous, edge-triggered Write operation or for dual-port RAM with simultaneous Read/Write. Also, XC4000E RAM can now be configured with user-defined initial values at power-up.
- **Increased System Speed:** Improvements in both device processing and system architecture result in path performance benefits that may have significant impact on the critical paths in a design.
- **More Flexible H Function Generator:** Additional inouts gives this resource the capability of being either partially or fully independent of the other two function generators, increasing the logic capacity and routability of the device.
- **Input/Output Block (IOB) Clock Enable:** I/O flip-flops have clock enables, making them more useful in many applications.
- **Increased Global Access to F and G Function Generators:** More function generator inouts are available to global routing resources, increasing flexibility and performance.
- **CMOS Threshold Compatibility:** Both inouts and/or outputs can be globally programmed for CMOS threshold compatibility.
- **Increased Carry Logic Speed:** The speed of the carry logic chain has increased dramatically, doubled in some cases.
- **Soft Startup:** When the configuration process is complete, outputs programmed for fast slew-rate operation are initially kept slow, avoiding ground-bounce problems when all the outputs are switched on.
- **Configuration Pin Pull-up Resistors:** During configuration, the three mode pins, M0, M1, and M2, have weak pull-up resistors. The effect on converted designs is discussed in "Configuration" on page 2.

Design Guidelines & Considerations

Because the XC4000E is a superset of the older XC4000 architecture, migrating a design from one of the older families to the XC4000E is relatively straightforward. The following issues may need to be addressed.

Configuration

The XC4000E can be configured using the same bitstream as an XC4000 or XC4000D device, or the design can be altered and recompiled to take advantage of some of the new features available in the XC4000E. XC4000A and XC4000H designs must be recompiled, targeting the new device.

One of the new features in the XC4000E is the addition of weak (50 k Ω) pull-up resistors to the configuration mode pins. These internal resistors make it possible to configure in Serial Slave mode (the most common configuration mode) without the addition of any external resistors. Older XC4000 families did not include these pull-up resistors, therefore a high-value pull-down resistor was acceptable to establish a Low value on any of these inouts. The XC4000E requires a resistor value of less than 7 k Ω to reliably generate a logic Low on the inout.

When converting from an older XC4000 device to an XC4000E, check the design for pull-down resistor values and change them, if necessary, to 4.7 k Ω resistors. Resistor values above the recommended level may prevent the XC4000E from properly configuring.

Footprint

Older XC4000 and XC4000E devices are footprint compatible for every common package. All control pins, configuration pins, and power pins are in the same locations. In general, no board re-layout is necessary when replacing an XC4000 device with an XC4000E.

An exception may occur when converting from an XC4000A or XC4000H to an XC4000E. In these transitions, the design is moved from a device with a maximum 24 mA sink current to one with a maximum 12 mA sink current, and it may be necessary to split the load across two outputs.

Design Performance

The XC4000E is available in -4, -3 and -2 speed grades. Consequently, the fastest XC4000E is significantly faster than any of the older XC4000 families. However, when converting an existing design, it is recommended that an XC4000E speed grade two grades faster than the original design be used for the initial compilation run. The resulting design should then be tested using XDelay, the Timing Analyzer, or timing simulation, to verify functionality at the required speed.

The reason for the recommended speed grade change is that the XC4000E is optimized for a three layer metal process and a higher overall system level performance, not for each individual specification. Although the majority of specifications are faster in the XC4000E than in any of the older XC4000 families, a few parameters are slower. The recommended approach should result in a successful conversion for the vast majority of designs.

After this first pass, it may be possible to use a slower speed grade. The best way to ensure that design performance will be met in the XC4000E is to use XACT-Performance™ to define the timing requirements of the design.

To place this suggestion in perspective, 30% of designs show equal or better performance in an XC4000E with the same speed grade as the original design. This conservative approach is recommended to maximize the chance that a given design will meet the required timing. See "Reporting Performance at Various Speed Grades" on page 5 for a quick method of evaluating the performance of a single routed design in several different speed grades.

Development System

The XC4000E is supported by the same XACTstep™ development system and uses the basic software tools that support the older XC4000 families. A new XC4000E library has been added to the set of Unified Libraries to support the architectural features of the XC4000E.

A software version that supports the XC4000E must be used. Users who have upgraded to at least the 6.0.1 (Windows) or 5.2.1 (DOS and UNIX) versions of XACTstep have the XC4000E library features and the necessary place and route tools. Otherwise, contact the local Xilinx sales representative for the availability of the most recent upgrade.

Input/Output Pads

For XC4000A and XC4000H designs, the difference in I/O capabilities may affect the conversion to an XC4000E.

XC4000A to XC4000E

XC4000A I/O have a 24 mA output drive, while the XC4000E has only 12 mA. Where a 24 mA drive is required, use two pins tied together to double the drive. Add the FAST parameter where speed is important.

XC4000A I/O allow two additional slew rate options, Medium Fast (MEDFAST) and Medium Slow (MEDSLCW). These parameters, if used in an XC4000A design, will need to be changed to FAST in the XC4000E version to achieve higher-than-default speed.

XC4000H to XC4000E

XC4000H devices have a higher I/O count than the corresponding XC4000E devices. If a high I/O count is required, a larger XC4000E device must be used to provide the same number of pins, as shown in Table 2.

Table 2: XC4000H Device Replacement Guide

XC4000H	Max. I/O	XC4000E	Max. I/O
XC4003H	160	XC4010E	160
XC4005H	192	XC4013E	192

XC4000H I/O have a default SoftEdge slew-rate control that limits drive to 4 mA. This is intended for capacitively loaded outputs, and can be selected with the CAP parameter. The alternative is the resistive (RES) mode, which increases drive to 24 mA. XC4000H designs will need the CAP or RES parameters removed when converted to the XC4000E. Add the FAST parameter where speed is important, and use two pins tied together to double the drive to 24 mA if necessary.

XC4000H I/O allow per-pin designation of CMOS or TTL thresholds and output levels, selected for each pin using a CMOS or TTL parameter (TTL is default). In the XC4000E, all input thresholds have the same value, and all output levels are the same, although the two are selected independently. If the XC4000H design has mixed I/O, the recommended approach is to designate the XC4000E inputs as TTL, since an input set for TTL can resolve both TTL and CMOS levels. The output level must depend on the destination devices. XC4000H designs will need the CMOS or TTL parameters removed when converted to the XC4000E.

Migration Methodology

If none of the new features in the XC4000E are used, an existing design can be compiled to the XC4000E using the XC4000 library, which supports all of the XC4000, XC4000A, XC4000D, and XC4000H families. If certain of the new features are to be used, new library components will be required. In this case, it is necessary to change the design to use the new XC4000E libraries. The features requiring the XC4000E library are:

- Synchronous (edge-triggered) or dual-port memory
- IOB clock enables

The library elements supporting these features are described in the *Libraries Supplement Guide*.

Because of the Xilinx Unified Library approach, a design can be migrated between families with a minimum of effort. The migration methodology itself is simple, and following the guidelines and considerations prescribed in this document will greatly improve the success of the migration.

This section describes how to perform the actual migration of an XC4000 design into the XC4000E library for third-party CAE interfaces.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Design Migration from XC4000 to XC4000E

VIEWlogic

To migrate an XC4000 VIEWlogic schematic to the XC4000E library, perform the following steps:

1. Add the XC4000E library to the VIEWlogic library search path. Edit the `viewdraw.ini` file in the project directory and add the XC4000E library path so that it appears in `viewdraw.ini` before the path to the XC4000 library.
2. To convert the XC4000 alias to XC4000E, run `altran`, the VIEWlogic library alias maintenance program:


```
altran -i primary xc4000=xc4000e
```

 where `xc4000` is the alias assigned to the XC4000 library, and `xc4000e` is the alias assigned to the XC4000E library.
3. Reprocess the design by running XMake or the Flow Engine.

Mentor

To migrate an XC4000 Mentor schematic to the XC4000E library, perform the following steps:

1. Invoke `PLD_DA` (it is not necessary to open the schematic).
2. On DA's desktop background (that is, outside of any schematic or symbol windows), call up the session pop-up menu with the mouse button on the right and select `Convert Design`.

Of the fields in the resulting dialog box these are the most relevant:

3. Select a group of designs from a list file? Whether you answer "yes" or "no" to this question affects the following field.
4. Enter Design name (List file = no). The name of the design to retarget. Convert Design does not traverse the hierarchy of a schematic.
5. Enter list file name (List file = yes). A file which lists designs, one per line, to retarget. This is useful if your design has many lower-level schematics.

TIP: A list file can easily be created by typing:

```
ls *.dgc_component.attn | sed
s/\.dgc_component.attn//g > listfile
```

The `ls` command lists all MGC components within a single directory. The `sed` command strips away the `.dgc_component.attn` trailer. The result is redirected to `listfile`.

5. Schematic name. The name of the schematic model (the default is "schematic").
7. Check & Save switch. Because all schematic sheets in Convert Design are literally re-drawn in Design Architect, you must apply Check & Save to each sheet. This switch controls whether to do this automatically. By default, this switch is set for manual checking because it allows you to spot Xilinx components that did not convert properly. Use the manual setting until you are comfortable with how Convert Design works and you are certain that all Xilinx components will convert properly.
3. From Technology. The device family from which you are converting (e.g., XC4000). This and the next field are case insensitive.
3. To Technology. The device family to which you are converting.
10. After filling out the fields in the dialog box and selecting "OK," you will see Convert Design doing its job directly in Design Architect.
11. Reprocess the design by running XMake or the Flow Engine.

Foundation

To migrate an XC4000 Foundation schematic to the XC4000E library, perform the following steps:

1. Select the Project Type option from the `Menu` file. Change Family, Part, and Speed settings, as desired, and click the `Change` button.
2. Open and save each schematic sheet macro. To do so, run the Schematic Editor and select the `Open` option from the `File` menu. The Open Sheet window allows you to quickly open all project top-level sheets and project schematic macros. Inspect the Project Manager messages for any warnings and errors.
3. Re-synthesize and update all FSM and HDL macros. Use the hierarchy browser in the Project Manager to search the project for the macros.
4. Note: if your project contains components that are not available in the new system library, you have to modify the project so as to preserve its functionality. In the case of a top level HDL project, you will need to re-synthesize the entire project.
5. Reprocess the design by running XMake or the Flow Engine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Additional Software Tips

Following are some additional tips that may help in converting a design.

Using the Old Design as a Guide

An XC4000 or XC4000D design can be used as a guide for an XC4000E design. Rename the old design, for example to "guide.lca", and add the following parameter when running PPR:

```
guide=guide.lca
```

Alternatively, in the Windows environment simply choose a previous XC4000 revision in the Guide File pull-down of the Implementation window.

Converting a Routed LCA File

The XDelay "convert" option can be used to translate a routed XC4000 or XC4000D LCA file to an XC4000E. The syntax is as follows:

```
xdelay -convert <design name>.lca <part & package>  
<new name>.lca
```

For example, to convert an XC4003 to an XC4003E, type:

```
xdelay -convert old.lca 4003EPC34 new.lca
```

The conversion utility will not allow you to choose a speed grade or write delay information into the new LCA file, so to perform these steps, run XDelay again with the following syntax:

```
xdelay -w -u -<new speed> <design name>
```

For example:

```
xdelay -w -u -3 new.lca
```

This command writes the new speed grade and delay information into the file without otherwise changing the design.

Reporting Performance at Various Speed Grades

XDelay can be used to report performance at various speed grades without changing the LCA file. To show the delays of the most critical paths, create a short XDelay report using the following command:

```
xdelay -u -<speed> -o critical.rpt <design name>
```

For example:

```
xdelay -u -2 -o critical.rpt new.lca
```

This command produces a text file called critical.rpt that contains the minimum worst-case pad-to-setup, clock-to-setup, and clock-to-pad values allowable for each clock in the design. Effectively, this report provides all the information necessary to evaluate the performance of the new speed grade.

Alternatively, in the Windows environment use the Performance Summary in the Timing Analyzer. Both XDelay and the Timing Analyzer can also be used to examine specific path delays in more detail if desired.

Additional Information

If there are problems with the conversion process, please contact the Xilinx Technical Support hotline for assistance.

Email: hotline@xilinx.com (24 hours)

Voice: 1-800-255-7778 (6:30AM - 5PM PST)

FAX: 1-408-379-1442 (24 hours)

Package-Specific Pinout Tables

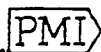
PC34 Package Pinouts

Pin	XC4003E	XC4005E XC4005L	XC4006E	XC4008E	XC4010E XC4010L
P1	GND	GND	GND	GND	GND
P2	VCC	VCC	VCC	VCC	VCC
P3	/O (A8)	/O (A8)	/O (A8)	/O (A8)	/O (A8)
P4	/O (A9)	/O (A9)	/O (A9)	/O (A9)	/O (A9)
P5	/O (A10)	/O (A10)	/O (A10)	/O (A10)	/O (A10)
P6	/O (A11)	/O (A11)	/O (A11)	/O (A11)	/O (A11)
P7	/O (A12)	/O (A12)	/O (A12)	/O (A12)	/O (A12)
P8	/O (A13)	/O (A13)	/O (A13)	/O (A13)	/O (A13)
P9	/O (A14)	/O (A14)	/O (A14)	/O (A14)	/O (A14)
P10	/O, SGCK1 A15	/O, SGCK1 A15	/O, SGCK1 A15	/O, SGCK1 A15	/O, SGCK1 A15
P11	VCC	VCC	VCC	VCC	VCC
P12	GND	GND	GND	GND	GND
P13	/O, PGCK1 A16	/O, PGCK1 A16	/O, PGCK1 A16	/O, PGCK1 A16	/O, PGCK1 A16
P14	/O (A17)	/O (A17)	/O (A17)	/O (A17)	/O (A17)
P15	/O, TDI	/O, TDI	/O, TDI	/O, TDI	/O, TDI
P16	/O, TCK	/O, TCK	/O, TCK	/O, TCK	/O, TCK
P17	/O, TMS	/O, TMS	/O, TMS	/O, TMS	/O, TMS
P18	/O	/O	/O	/O	/O
P19	/O	/O	/O	/O	/O
P20	/O	/O	/O	/O	/O
P21	-GND	GND	GND	GND	GND
P22	VCC	VCC	VCC	VCC	VCC
P23	/O	/O	/O	/O	/O
P24	/O	/O	/O	/O	/O
P25	/O	/O	/O	/O	/O
P26	/O	/O	/O	/O	/O
P27	/O	/O	/O	/O	/O
P28	/O	/O	/O	/O	/O
P29	/O, SCGK2	/O, SCGK2	/O, SCGK2	/O, SCGK2	/O, SCGK2
P30	/O (M1)	/O (M1)	/O (M1)	/O (M1)	/O (M1)
P31	GND	GND	GND	GND	GND
P32	/O (M0)	/O (M0)	/O (M0)	/O (M0)	/O (M0)
P33	VCC	VCC	VCC	VCC	VCC
P34	/O (M2)	/O (M2)	/O (M2)	/O (M2)	/O (M2)
P35	/O, PGCK2	/O, PGCK2	/O, PGCK2	/O, PGCK2	/O, PGCK2
P36	/O (HDC1)	/O (HDC1)	/O (HDC1)	/O (HDC1)	/O (HDC1)
P37	/O (LDC1)	/O (LDC1)	/O (LDC1)	/O (LDC1)	/O (LDC1)
P38	/O	/O	/O	/O	/O
P39	/O	/O	/O	/O	/O
P40	/O	/O	/O	/O	/O
P41	/O (INIT)	/O (INIT)	/O (INIT)	/O (INIT)	/O (INIT)
P42	VCC	VCC	VCC	VCC	VCC
P43	GND	GND	GND	GND	GND
P44	/O	/O	/O	/O	/O

Pin	XC4003E	XC4005E XC4005L	XC4006E	XC4008E	XC4010E XC4010L
P45	/O	/O	/O	/O	/O
P46	/O	/O	/O	/O	/O
P47	/O	/O	/O	/O	/O
P48	/O	/O	/O	/O	/O
P49	/O	/O	/O	/O	/O
P50	/O	/O	/O	/O	/O
P51	/O, SGCK3	/O, SGCK3	/O, SGCK3	/O, SGCK3	/O, SGCK3
P52	GND	GND	GND	GND	GND
P53	DONE	DONE	DONE	DONE	DONE
P54	VCC	VCC	VCC	VCC	VCC
P55	/O, GRAM	/O, GRAM	/O, GRAM	/O, GRAM	/O, GRAM
P56	/O (D7)	/O (D7)	/O (D7)	/O (D7)	/O (D7)
P57	/O, PGCK3	/O, PGCK3	/O, PGCK3	/O, PGCK3	/O, PGCK3
P58	/O (D6)	/O (D6)	/O (D6)	/O (D6)	/O (D6)
P59	/O (D5)	/O (D5)	/O (D5)	/O (D5)	/O (D5)
P60	/O (CS0)	/O (CS0)	/O (CS0)	/O (CS0)	/O (CS0)
P61	/O (D4)	/O (D4)	/O (D4)	/O (D4)	/O (D4)
P62	/O	/O	/O	/O	/O
P63	VCC	VCC	VCC	VCC	VCC
P64	GND	GND	GND	GND	GND
P65	/O (D3)	/O (D3)	/O (D3)	/O (D3)	/O (D3)
P66	/O (RS)	/O (RS)	/O (RS)	/O (RS)	/O (RS)
P67	/O (D2)	/O (D2)	/O (D2)	/O (D2)	/O (D2)
P68	/O	/O	/O	/O	/O
P69	/O (D1)	/O (D1)	/O (D1)	/O (D1)	/O (D1)
P70	/O (RCLK, RDY/ BUSY)	/O (RCLK, RDY/ BUSY)	/O (RCLK, RDY/ BUSY)	/O (RCLK, RDY/ BUSY)	/O (RCLK, RDY/ BUSY)
P71	/O (D0, DIN)	/O (D0, DIN)	/O (D0, DIN)	/O (D0, DIN)	/O (D0, DIN)
P72	/O, SGCK4 (DOUT)	/O, SGCK4 (DOUT)	/O, SGCK4 (DOUT)	/O, SGCK4 (DOUT)	/O, SGCK4 (DOUT)
P73	CCLK	CCLK	CCLK	CCLK	CCLK
P74	VCC	VCC	VCC	VCC	VCC
P75	/O, TDO	/O, TDO	/O, TDO	/O, TDO	/O, TDO
P76	GND	GND	GND	GND	GND
P77	/O (A0, WS)	/O (A0, WS)	/O (A0, WS)	/O (A0, WS)	/O (A0, WS)
P78	/O, PGCK4 A1	/O, PGCK4 A1	/O, PGCK4 A1	/O, PGCK4 A1	/O, PGCK4 A1
P79	/O (CS1, A2)	/O (CS1, A2)	/O (CS1, A2)	/O (CS1, A2)	/O (CS1, A2)
P80	/O (A3)	/O (A3)	/O (A3)	/O (A3)	/O (A3)
P81	/O (A4)	/O (A4)	/O (A4)	/O (A4)	/O (A4)
P82	/O (A5)	/O (A5)	/O (A5)	/O (A5)	/O (A5)
P83	/O (A6)	/O (A6)	/O (A6)	/O (A6)	/O (A6)
P84	/O (A7)	/O (A7)	/O (A7)	/O (A7)	/O (A7)

2/29/96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADC-908

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

Precision Monolithics Inc.

FEATURES

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

ORDERING INFORMATION¹

PACKAGE: 18-PIN DIP AND SO				
		MILITARY* TEMPERATURE -55°C TO +125°C	EXTENDED INDUSTRIAL TEMPERATURE -40°C TO +85°C	COMMERCIAL TEMPERATURE 0°C TO +70°C
±1/2 (LSB)	±3/4 (LSB)	ACC908AX	ACC908EX	ACC908GP
±3/4	±7/8	ACC908BX	ACC908FX	—
±3/4	±7/8	—	ACC908FP	—
±3/4	±7/8	—	ACC908FS	—

¹ For devices processed in total compliance to MIL-STD-883, add 883 after part number. Consult factory for 883 data sheet.

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

GENERAL DESCRIPTION

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

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

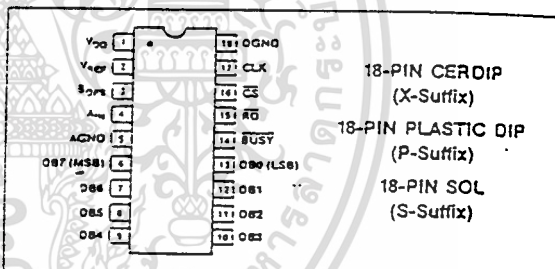
operations. This input structure permits the ADC-908 to be used as a memory-mapped input device. Depending on the control timing waveforms, the ADC-908 is interfaced like static RAM, ROM, or slow memory.

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

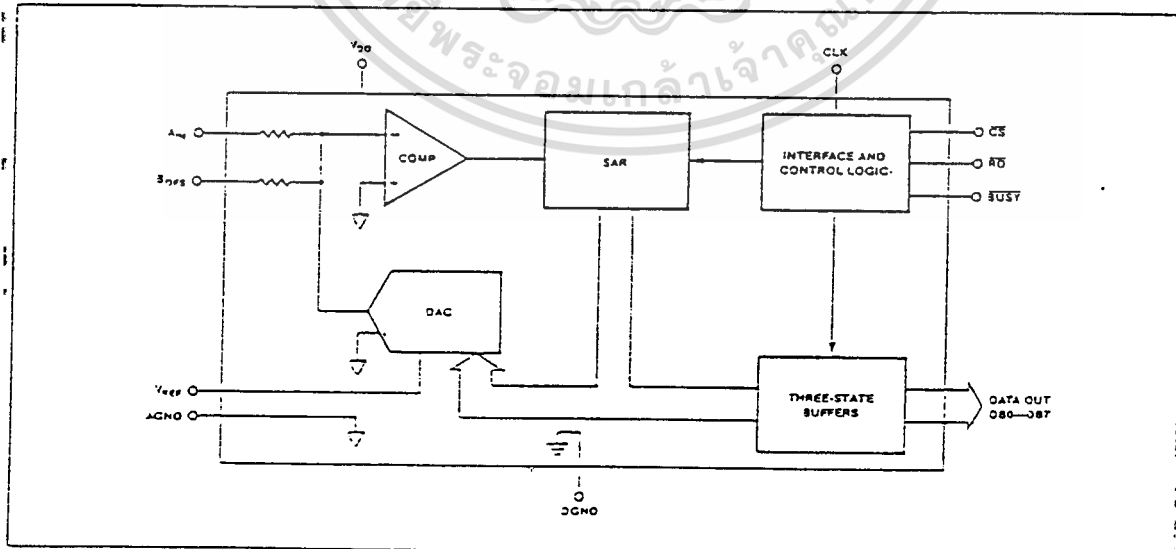
With its on-board comparator, interface logic, optional internal clock, and ±5V operation, the ADC-908 is the ideal low-cost solution for microprocessor-based 8-bit A/D systems.

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

PIN CONNECTIONS



FUNCTIONAL DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

V_{DD} to AGND	0V, +7.0V
V_{DD} to DGND	0V, +7.0V
AGND to DGND	-0.3V, V_{DD}
CS, RD to DGND	-0.3V, $V_{DD} + 0.3V$
DB ₇ , DB ₆ to DGND	-0.3V, V_{DD}
CLK, BUSY to DGND	-0.3, V_{DD}
B _{0FS} , A _{1IN}	$\pm 20V$
V _{REF}	0V, -20V
Operating Temperature Range	
ADC-908AX, BX	-55°C to +125°C
ADC-908EX, FX, FP, FS	-40°C to +85°C
ADC-908GP	0°C to +70°C

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

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

NOTES:

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

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

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
ACCURACY						
Resolution	N		8	-	-	Bits
Integral Nonlinearity	INL	A/E/G Grades	-1/2	-	-1/2	LSB
		B/F Grades	-3/4	-	-3/4	
Differential Nonlinearity	DNL	A/E/G Grades	-3/4	-	-3/4	LSB
		B/F Grades	-7/8	-	-7/8	
Gain Error	$Q_{FS\%}$	A/E/G Grades	-3	-	-3	LSB
		$T_A = -25^\circ\text{C}$	-4.5	-	-4.5	
		B/F Grades	-5	-	-5	
		$T_A = \text{Full Temp Range}$	-6.5	-	-6.5	
Offset Error	$V_{FS\%}$	A/E/G Grades	-30	-	-30	mV
		$T_A = -25^\circ\text{C}$	-50	-	-50	
		B/F Grades	-60	-	-60	
		$T_A = \text{Full Temp Range}$	-80	-	-80	
ANALOG INPUTS						
Resistance Mismatch B _{0FS} to A _{1IN}	ΔR_{AB}		-1	-	-1	%
Inout Resistance at V _{REF} (Note 1)	R_{REF}		5	-	15	k Ω
Inout Resistance at B _{0FS} , A _{1IN}	R_{B0FS} R_{A1IN}		10	-	30	k Ω
Reference Voltage Range	V _{REF}	Specified Conversion Accuracy	-	-10	-	V
Reference Voltage Range	V _{REF}	Degraded Conversion Accuracy	-5	-	-15	V
Reference Current (Note 6)	I _{REF}	Conversion Complete Prior to Reset	-	-	2.4	mA
Nominal Analog Inout Range						
Unipolar Mode	V _{INU}		-	0 to +V _{REF}	-	V
Bipolar Mode	V _{INB}		-	-V _{REF} to +V _{REF}	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADC-908 MICROPROCESSOR-COMPATIBLE FAST 8-BIT D/A CONVERTER

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

Continued

PARAMETER	SYMBOL	CONDITIONS	ADC-908			UNITS
			MIN	TYP	MAX	
LOGIC INPUTS						
Inout HIGH Voltage RD, CS Inouts	V_{IH}		2.4	—	—	V
Inout LOW Voltage RD, CS Inouts	V_{IL}		—	—	0.8	V
Inout Current RD, CS Inouts	I_{IN}	$T_A = -25^\circ C$ $T_A = \text{Full Tempo Range}$	—	—	1	μA
Inout Capacitance RD, CS Inouts (Note 6)	C_{IN}		—	—	5	pF
Inout HIGH Voltage, Clock Input	V_{IH}		2.4	—	—	V
Inout LOW Voltage, Clock Input	V_{IL}		—	—	0.8	V
Inout HIGH Current, Clock Input	I_{IH}		—	—	2	mA
Inout LOW Current, C xck Input	I_{IL}	$T_A = -25^\circ C$ $T_A = \text{Full Tempo Range}$	—	—	1	μA
LOGIC OUTPUTS						
Output HIGH Voltage BUSY, DB0-7	V_{OH}	$I_{SOURCE} = 40\mu A$	4.0	—	—	V
Output LOW Voltage BUSY, DB0-7	V_{OL}	$I_{SINK} = 1.6mA$	—	—	0.4	V
Floating Leakage Current, DB0-7	I_{LKG}	$T_A = -25^\circ C$ $T_A = \text{Full Tempo Range}$	—	—	1	μA
Floating State Output Capacitance	C_{OZ}	(Note 6)	—	—	7	pF
POWER REQUIREMENTS						
Standby Current	I_{DD}	$V_{DD} = -4.75V \text{ to } +5.25V$	—	—	2.5	mA
DIGITAL INTERFACE TIMING						
CS Minimum Pulse Width (Note 5)	t_{CS}	$T_A = -25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	50	—	—	ns
RD to CS			—	—	—	
Setup Time (Note 6)	t_{WCS}		0	—	—	ns
CS to BUSY Propagation Delay (Note 6)	t_{C2PO}	BUSY Load = 20pF $T_A = -25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	—	—	120	ns
BUSY to RD Setup Time (Notes 2, 5)	t_{SR}	BUSY Load = 100pF $T_A = +25^\circ C$ $T_A = T_{MIN}$ $T_A = T_{MAX}$	—	—	150	ns
BUSY to CS Setup Time (Note 5)	t_{SCS}		—	—	120	ns
			—	—	200	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC-908 CMOS MICROPROCESSOR-COMPATIBLE FAST 8-BIT A/D CONVERTER

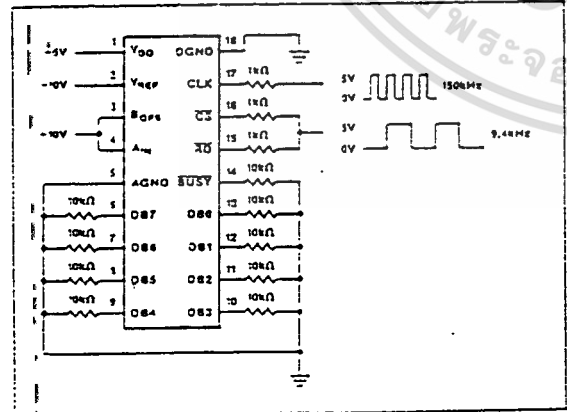
ELECTRICAL CHARACTERISTICS at $V_{DD} = +5V$, $V_{REF} = -10V$, Unipolar Configuration, $R_{CLK} = 43k\Omega$, $C_{CLK} = 100pF$; $-40^\circ C \leq T_A \leq +85^\circ C$ for ADC-908E/F, $0^\circ C \leq T_A \leq +70^\circ C$ for ADC-908G, $-55^\circ C \leq T_A \leq +125^\circ C$ for ADC-908A/B, unless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

Continued

PARAMETER	SYMBOL	CONDITIONS	MIN	ADC-908 TYP	MAX	UNITS	
Data Access Time (Note 6)	t_{RAO}	$C_L = 20pF$					
		$T_A = -25^\circ C$	—	—	140		
		$T_A = T_{MIN}$	—	—	100		
		$T_A = T_{MAX}$	—	—	200		
		$C_L = 100pF$					ns
		$T_A = -25^\circ C$	—	—	170		
Data Hold Time (Notes 3, 6)	t_{GHO}	$T_A = -25^\circ C$ (Note 3)	30	—	100		
		$T_A = T_{MIN}$	20	—	70	ns	
		$T_A = T_{MAX}$	40	—	140		
\overline{RD} to \overline{RD} Hold Time (Note 6)	t_{RHCS}	$T_A = -25^\circ C$	—	—	200		
		$T_A = T_{MIN}$	—	—	120	ns	
		$T_A = T_{MAX}$	—	—	250		
Reset Time Requirement (Note 6)	t_{RESET}	$T_A = -25^\circ C$	450	—	—	ns	
		$T_A = \text{Full Temp. Range}$	500	—	—		
Conversion Time (Note 4)	$t_{CONVERT}$	Static RAM Mode					
		External Clock			5	μs	
		$f = 1.25MHz$					
		ROM Mode			7		
\overline{RD} HIGH to BUSY Propagation Delay, ROM Mode (Notes 4, 5, 6)	t_{WBPO}	$C_L = 20pF$			500	ns	
		$T_A = -25^\circ C$			400		
		$T_A = T_{MAX}$			300		

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

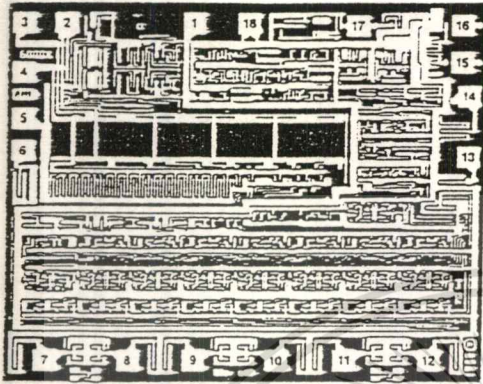
BURN-IN CIRCUIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



DICE CHARACTERISTICS



- 1. V_{DD}
- 2. V_{REF}
- 3. B_{OFS}
- 4. A_{IN}
- 5. AGND
- 6. DB7(MSB)
- 7. DB6
- 8. DB5
- 9. DB4
- 10. DB3
- 11. DB2
- 12. DB1
- 13. DB0(LSB)
- 14. \overline{BUSY}
- 15. \overline{RD}
- 16. \overline{CS}
- 17. CLK
- 18. DGND

DIE SIZE 0.129 X 0.103 inch, 13.287 sq. mils
(3.28 X 2.62 mm, 8.58 sq. mm)

For additional DICE ordering information, refer to PMI's Data Book, Section 2.

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

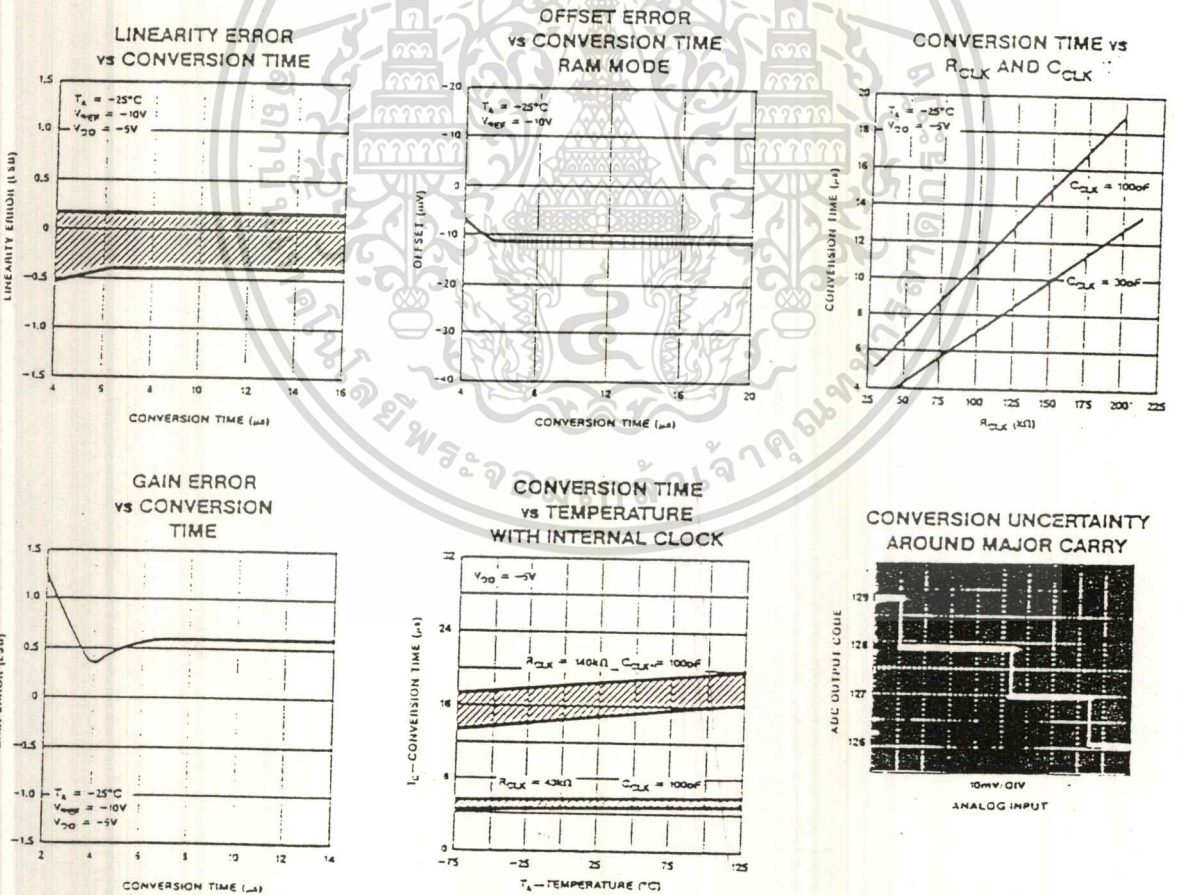


WAFER TEST LIMITS at $V_{DD} = +5V$, $V_{REF} = -10.000V$, $AGND = DGND = 0V$, $T_A = +25^\circ C$, unless otherwise noted. (Continued)

PARAMETER	SYMBOL	CONDITIONS	ADC-908 LIMIT	UNITS
DIGITAL OUTPUTS				
Output HIGH Voltage BUSY, D80-7	V_{OH}	$I_{SOURCE} = 40\mu A$	4	V MIN
Output LOW Voltage BUSY, D80-7	V_{OL}	$I_{SINK} = 1.5mA$	0.4	V MAX
Floating Leakage Current	I_{LKG}		1	μA
POWER REQUIREMENTS				
Standby Current	I_{DD}	$V_{DD} = -4.75V$ to $5.25V$	2.5	mA MAX
TIMING				
Conversion Time	$t_{CONVERT}$	Static RAM Mode, External Clock, $f_c = 1.35MHz$	5	μs MAX

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

TYPICAL PERFORMANCE CHARACTERISTICS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL CIRCUIT INFORMATION

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

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

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

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

PIN FUNCTIONS

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

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

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

APPLICATIONS INFORMATION

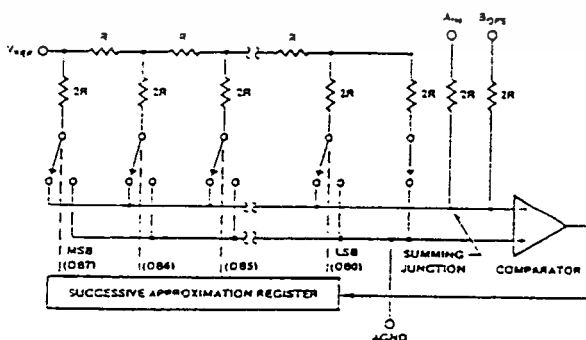
The ADC-908 may be interfaced as if it were a static RAM, a ROM, or a slow-memory device. Each of these interface modes has its own timing and software requirements as described below. These requirements must be rigidly met, as improper timing may cause the ADC-908 to change modes.

HOW TO CHOOSE AN OPERATING MODE

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

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

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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



This requirement may be met by inserting NOP or other program instructions between consecutive READ operations. Conditional or branch instructions may be used, but keep in mind that data may become out-of-date if excessive time elapses between consecutive READ instructions.

TABLE 2: Truth Table, ROM Mode

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

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

OPERATING DESCRIPTION: SLOW-MEMORY MODE

The slow-memory mode is intended for systems in which the ADC-908 BUSY output is used as an interrupt to force the

microprocessor into WAIT states during data conversion.

In slow-memory mode, inputs CS and RD are tied together. The common RD and CS signal is derived from the ADC-908 address decoder. To satisfy the timing requirements, it is advisable to latch the address using ALE (8085) or SYNC (8080). For 8080 or 8085-based systems, connect the microprocessor READY input to the ADC-908 BUSY output. (See Figure 4.)

TABLE 3: Truth Table, Slow-Memory Mode

INPUTS		OUTPUTS		ADC-908 OPERATION
CS & RD	BUSY	DB7-DB0		
H	H	HIGH-Z		No Effect (Not Selected)
\downarrow	\downarrow	HIGH-Z		Start Conversion
L	L	HIGH-Z		Conversion in Progress. μ P in WAIT State
L	\uparrow	HIGH-Z to DATA		Conversion Complete. Read Data
\uparrow	H	DATA to HIGH-Z		Reset and Deselect Converter

FIGURE 3: ROM Mode Timing Diagram (CS Held LOW)

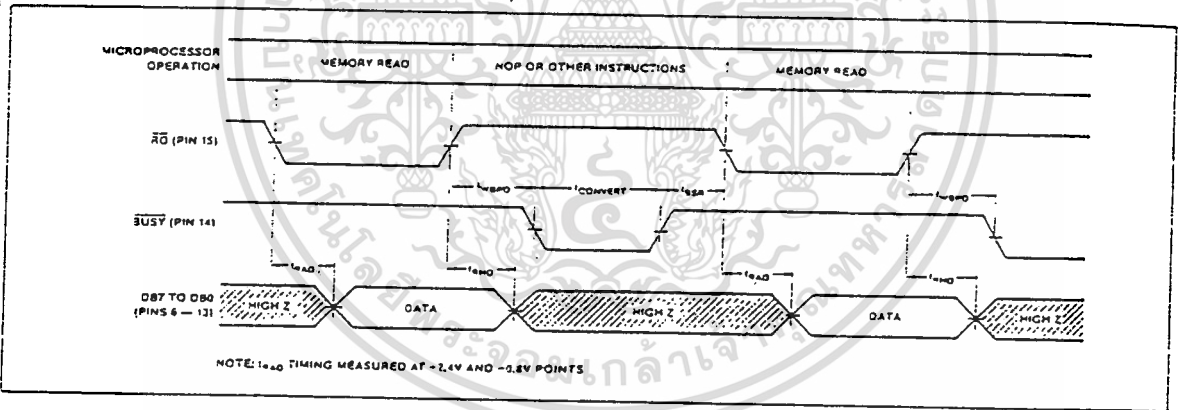
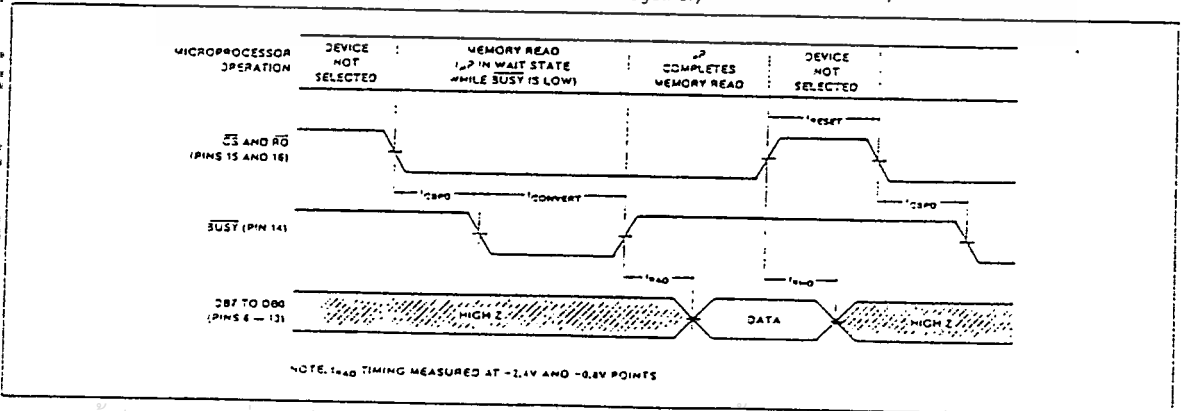


FIGURE 4: Slow-Memory Mode Timing Diagram (CS and RD Tied Together)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Do not execute a WRITE instruction at the ADC-908 address when in slow-memory mode, since bus conflicts will arise. In some architectures, an accidental WRITE instruction may be locked out in hardware, by proper strobing of the ADC-908 address decoder.

INITIALIZATION

In all operating modes, the ADC-908 is initialized by executing a READ instruction to the ADC-908 address. The data obtained should be ignored.

CLOCK OSCILLATOR

The ADC-908 may be used with its internal asynchronous clock oscillator. An external resistor and capacitor are required. Typical values are $R = 43k\Omega$ and $C = 100pF$, for conversion times in the $6\mu s$ range. For applications in which the fastest conversion times are required, an external clock is recommended. The external clock must be gated by the use of a 74125-type three-state buffer, with an output pullup resistor. Optimum conversion accuracy is obtained when CS goes LOW on a positive clock edge. The maximum external clock frequency is 1.35MHz (See Figure 5 and 6.)

REFERENCE VOLTAGE

A negative reference voltage must be applied to the ADC-908 V_{REF} input. Optimum full-scale accuracy is obtained using $-10.00V$, although V_{REF} may be $-5.00V$, $-10.24V$, or other voltages within its specified range.

Over the full temperature range, optimum gain accuracy is obtained when the input to the V_{REF} pin is from a low-impedance source. A resistor or trimmer may be used in series with the V_{REF} pin, but this trim technique is not as accurate as a low-impedance source. (See Figure 7.)

For a cost-effective $+10.00V$ or $-10.24V$ reference with excellent accuracy and low temperature coefficient, ask for PMI's REF-08. Consult your sales representative for availability.

FIGURE 5: Using the Internal Clock Oscillator

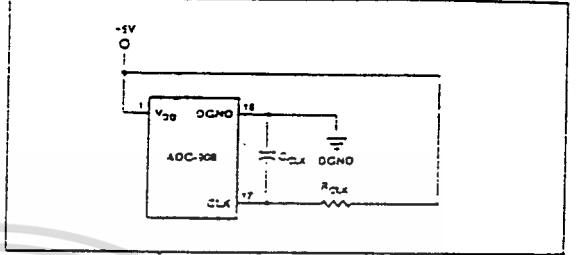


FIGURE 5: Using an External Clock

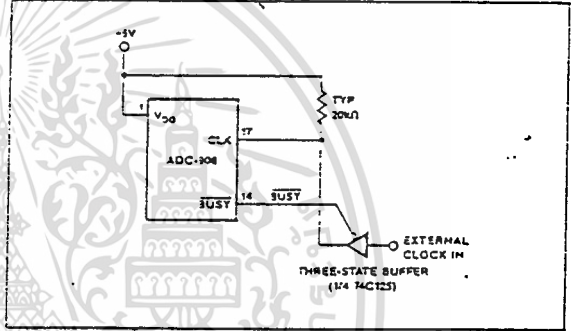
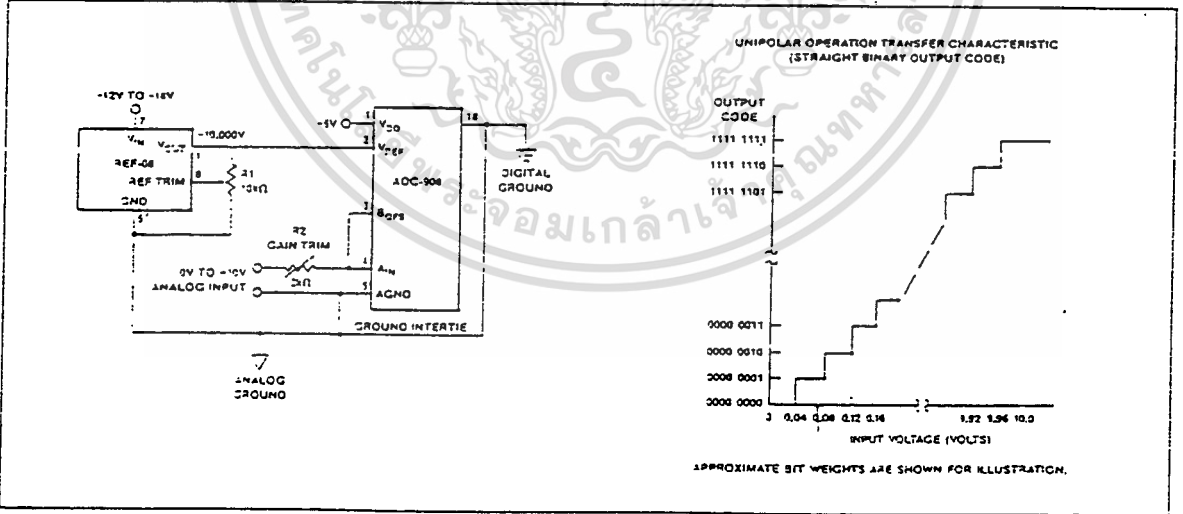


FIGURE 7: Unipolar Operation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PMI

ANALOG INPUT VOLTAGE

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

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

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

UNIPOLAR BINARY OPERATION

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

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

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

To adjust offset:

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

Following offset calibration, full scale gain can be calibrated:

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

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

BIPOLAR OPERATION

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

To calibrate this circuit:

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

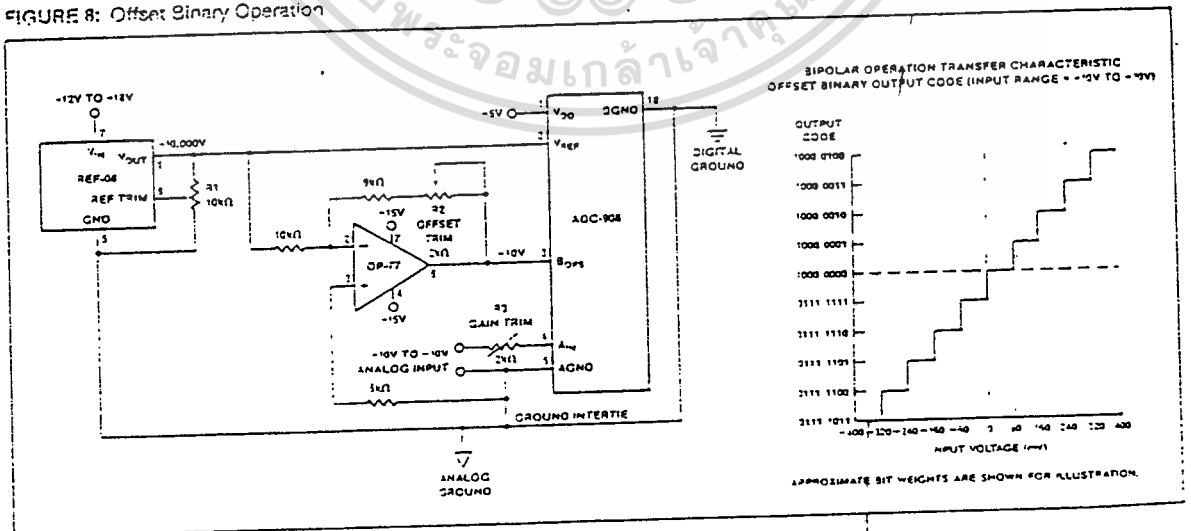
Complementary Offset Binary—Figure 9 shows a complementary offset binary circuit. In this bipolar mode, the $-10V$ to $+10V$ analog input is conditioned to a 0 to $+10V$ signal range for normal unipolar conversion.

In calibrating this circuit, adjust offset before gain.

Offset Adjustment:

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

FIGURE 8: Offset Binary Operation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Gain Adjustment

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

DIGITAL CONSIDERATIONS

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

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

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

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

ANALOG CONSIDERATIONS

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

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

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

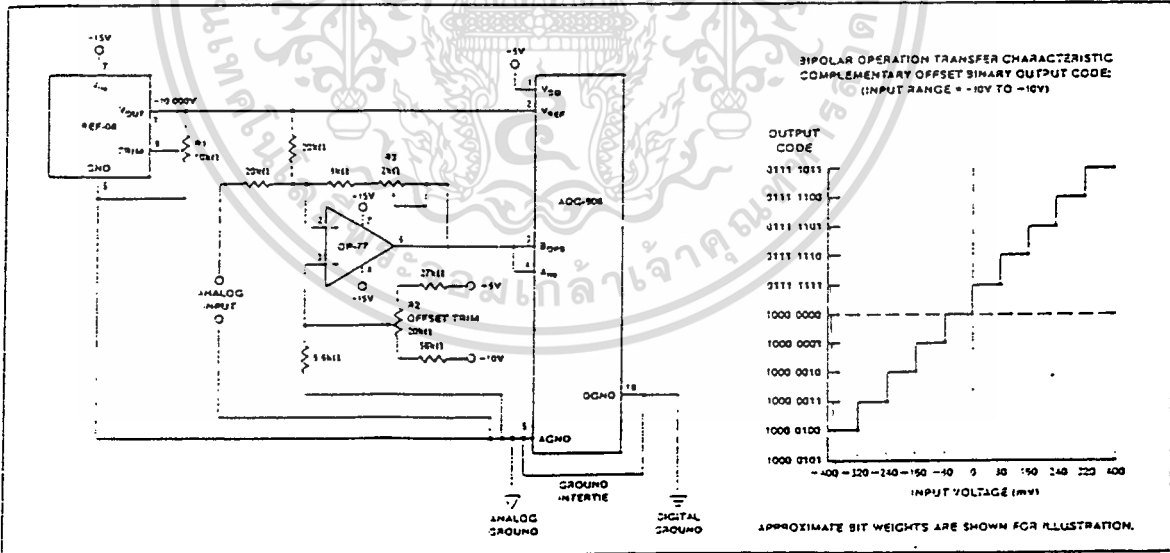
Offset Correction—Conversion offset errors may be corrected by counter-offsetting the buffer amplifier driving A_{IN} . This offset correction may be accomplished by applying a correction current to the buffer's summing junction or by tapping a voltage divider sitting between V_{DD} and V_{REF} , and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

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

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

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

FIGURE 9: Complementary Offset Bipolar Operation



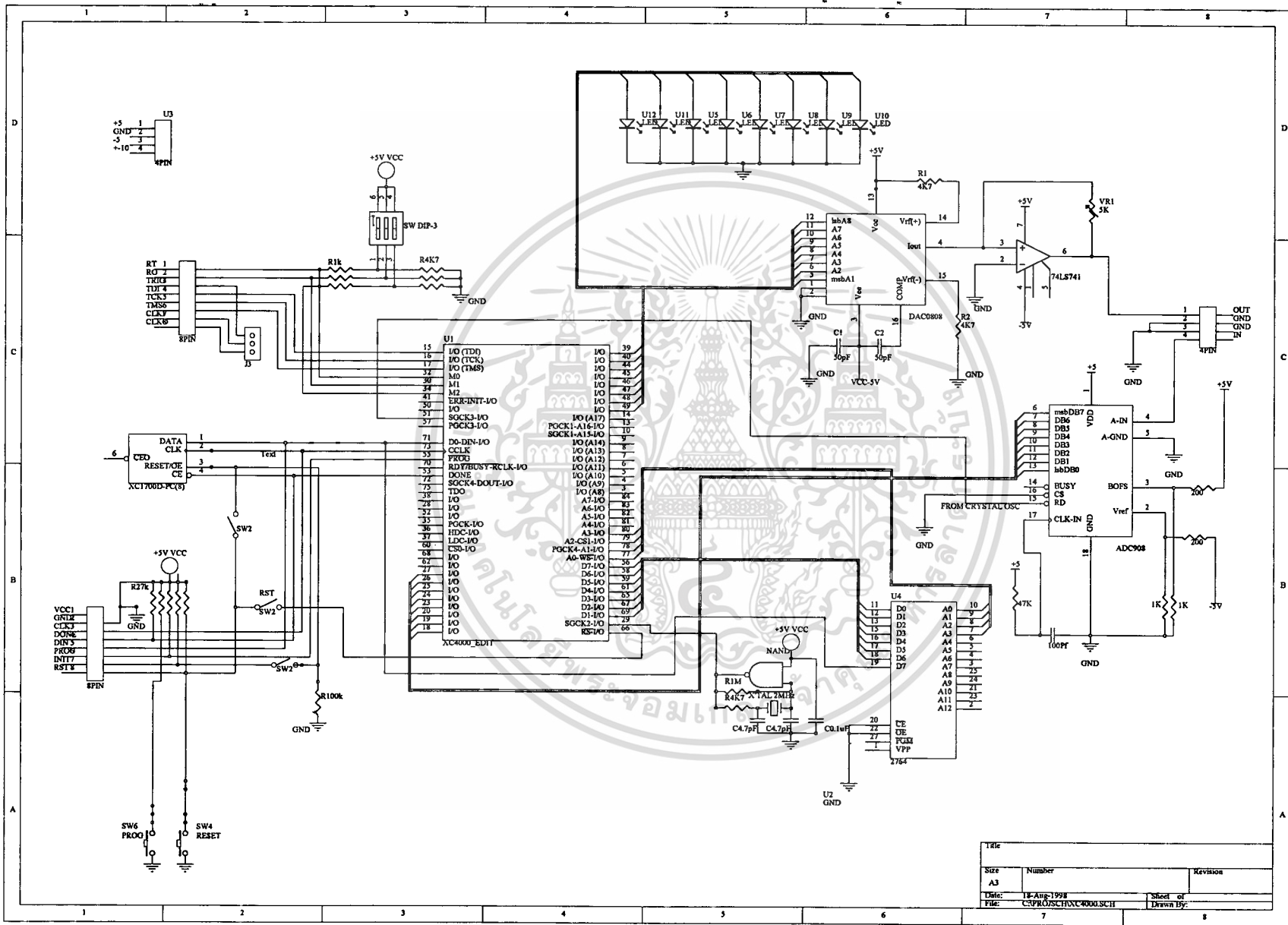
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

การต่อวงจร ADC, DAC, และวงจร FPGA_s

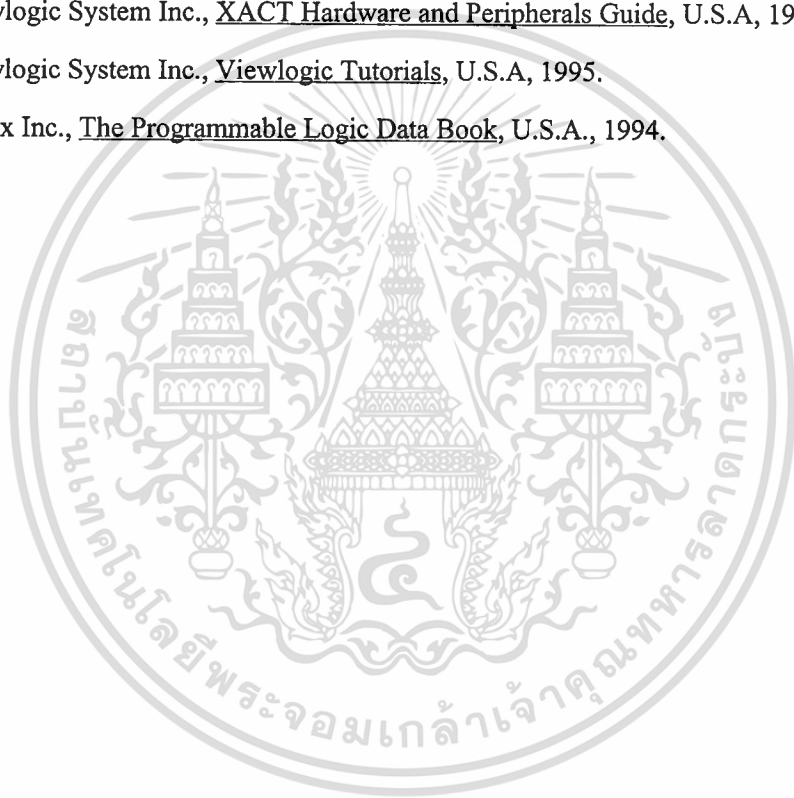
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		
Size	Number	Revision
A3		
Date:	18-Aug-1998	Sheet of
File:	C:\PROJ\CHVC4000.SCH	Drawn By:

บรรณานุกรม

- [1] ไพรัช ชัยพงษ์, การประมวลผลสัญญาณดิจิทัล ตอน การออกแบบวงจรองดิจิทัล, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2535.
- [2] วัลลภ สุระกำพลธร, การประมวลผลสัญญาณเชิงเลข, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2533.
- [3] Thomas Richard Mccalla, Digital Logic and Computer Design, University of Science and Arts of Ohlahoma, U.S.A., 1990.
- [4] Viewlogic System Inc., XACT Hardware and Peripherals Guide, U.S.A, 1994.
- [5] Viewlogic System Inc., Viewlogic Tutorials, U.S.A, 1995.
- [6] Xilinx Inc., The Programmable Logic Data Book, U.S.A., 1994.



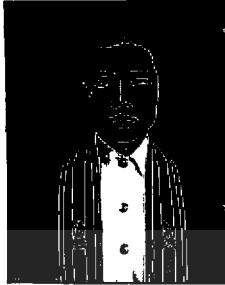
ประวัติผู้แต่ง



ชื่อ	นายกسانติ ศรีกุลนาถ
วันเดือนปี	4 กุมภาพันธ์ 2520
สถานที่เกิด	กรุงเทพฯ
ภูมิลำเนาเดิม	287/33 ถ. สาธุประดิษฐ์ เขตยานนาวา กรุงเทพฯ
โทรศัพท์	02-426-1521
ที่อยู่ปัจจุบัน	53 หมู่บ้านประชาสุขซิติ ถ. ประชาอุทิศ เขตราชบุรีระ กรุงเทพมหานคร
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนมูลนิธิวัดศรีอุบลรัตนาราม
มัธยมศึกษา	โรงเรียนปทุมพิทยาคม
ปวช.	วิทยาลัยเทคนิคอุบลราชธานี
ปวส.	วิทยาลัยเทคนิคอุบลราชธานี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงาน	-
คติพจน์	ถ้าคิดจะทำ ต้องทำให้ถึงที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

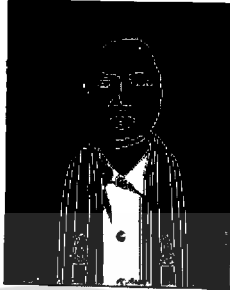
ประวัติผู้แต่ง



ชื่อ	นางสาวธิดารัตน์ บุตรโคตร
วันเดือนปี	21 มกราคม 2520
สถานที่เกิด	อุบลราชธานี
ภูมิลำเนาเดิม	137 บ. คอนสำราญ ต. กุลชมพู อ. พิบูลมังสาหาร จ. อุบลราชธานี 34000
โทรศัพท์	045-441480
ที่อยู่ปัจจุบัน	98/238 อาคารสุหทัยเพลช แขวงคลอง 3 ประเวศ เขตลาดกระบัง กรุงเทพฯ 10520
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนพิบูลมังสาหารวิภาคย์วิทยากรณ์
มัธยมศึกษา	โรงเรียนนารีบุกุล
ปวช.	วิทยาลัยเทคนิคอุบลราชธานี
ปวส.	วิทยาลัยเทคนิคอุบลราชธานี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงาน	-
คติพจน์	อดีตคือภาพพจน์ อนาคตคือความฝัน ปัจจุบันคือความจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ	นางสาวประทีป เข็มทอง
วันเดือนปี	31 กรกฎาคม 2519
สถานที่เกิด	นครราชสีมา
ภูมิลำเนาเดิม	2174 ถ.มิตรภาพ ต.ในเมือง อ.เมือง จ.นครราชสีมา
	30000
โทรศัพท์	044-282582
ที่อยู่ปัจจุบัน	-
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดบ้านหลักร้อย
มัธยมศึกษา	โรงเรียนสมบูรณวิद्याนุกูล
ปวช.	วิทยาลัยเทคนิคนครราชสีมา
ปวส.	วิทยาลัยเทคนิคนครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงาน	-
คติพจน์	อย่ามองคนเพียงแค่ผลของงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ	นายศักดิ์ชัย ศรีมากรณ์
วันเดือนปี	16 มิถุนายน 2519
สถานที่เกิด	ขอนแก่น
ภูมิลำเนาเดิม	95 หมู่ 7 ต. หนองโก อ. กระนวน จ. ขอนแก่น 40170
ที่อยู่ปัจจุบัน	95 หมู่ 7 ต. หนองโก อ. กระนวน จ. ขอนแก่น 40170
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านหนองแวงคู
มัธยมศึกษา	โรงเรียนศรีกระนวนวิทยาคม
ปวช.	วิทยาลัยเทคนิคขอนแก่น
ปวส.	สถาบันเทคโนโลยีราชมงคลวิทยาเขตขอนแก่น
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงาน	-
คติพจน์	อย่าคิด อย่าหวัง หรือฝันในสิ่งที่ไม่เป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้