สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประมวลผลสัญญาณดิจิตอลโดยวิธีการ Convolution

DIGITAL SIGNAL PROCESSING BY CONVOLUTION METHOD

T 086855

นายประพฤทธิ์ แจวตระกูล
นายสัมฤทธิ์ เกียวมิตรไมตรี
นายโสภณ ศรีสวัสดิ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 86855
วัน,เดือน,ปี 16 ส.ค. 2552

.b 10873880
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การประมวลผลสัญญาณดิจิตอลโดยวิธีการ **Convolution**

DIGITAL SIGNAL PROCESSING BY CONVOLUTION METHOD

ชื่อนักศึกษา นายประพฤทธิ์ แจวตระกูล

นายสัมฤทธิ์ เกียวมิตรไมตรี

นายโสภณ ศรีสวัสดิ์

อาจารย์ที่ปรึกษา รศ.ดร. กนก เจนจิระพงศ์เวช

ผศ. ชวลิต เบญจางคประเสริฐ

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2541

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การประมวลผลสัญญาณดิจิตอลโดยวิธีการ Convolution

โดย	นายประพฤทธิ์	แจวตระกูล
	นายสัมฤทธิ์	เกียวมิตรไมตรี
	นายโสภณ	ศรีสวัสดิ์
อาจารย์ที่ปรึกษา	รศ.ดร. กนก	เจนจิระพงศ์เวช
	ผศ. ชวลิต	เบญจางคประเสริฐ
ปีการศึกษา	2541	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอวิธีการออกแบบสร้างตัวกรองสัญญาณป้อนกลับเชิงเลข จากต้นแบบทางคณิตศาตร์ถ่ายโอนไปสู่วงจรอิเล็กทรอนิกส์หรือฮาร์ดแวร์ ใช้โครงสร้างของตัวประมวลผลเชิงเลขแบบเลขคณิตกระจาย หรือแบบ รอม-แอกคูมูเลเตอร์ ให้ทำงานได้จริงในทางปฏิบัติและให้ผิดเพี้ยนจากทฤษฎีต้นแบบน้อยที่สุด โดยหลักการแล้วตัวประมวลผลเชิงเลขชนิดนี้มีความเร็วในการทำงานสูง ทำให้ตัวกรองเชิงเลขที่สร้างมีประสิทธิภาพดีในเชิงของการประมวลผลและใช้ฮาร์ดแวร์ไม่มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Signal Processing By Convolution Method

By	Mr.Prapraut Jawtrakool
	Mr.Somrit Kyomitmaitee
	Mr.Sophon Srisawat
Advisor	Assoc.Prof.Dr. Kanok Janchirapongvej
	Asst.Prof. Chawalit Benjangkaprasert
Year	1998

ABSTRACT

This project proposes a design of digital filter base on distributed arithmetic.With the used of the arithmetic prototype then the hardware can be implemented. This circuit consist of Rom-accumulator and the other commercial chips,they give good efficiency in the form of processing and computation al speed without hardware complexity.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงงานนี้ คณะผู้จัดทำได้รับความช่วยเหลือ และคำแนะนำต่าง ๆ มากมายจากบุคคลหลาย ๆ ท่าน ณ โอกาสนี้คณะผู้จัดทำขอขอบพระคุณ

รศ.ดร. กนก เจนจิระพงศ์เวช และ ผศ.ชวลิต เบญจางคประเสริฐ อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำ,การดูแลเอาใจใส่อย่างดีตลอดมา

อาจารย์ทุกท่านในภาควิชา เทคนิคอุตสาหกรรม ที่ได้ให้คำแนะนำอันมีประโยชน์

เพื่อน ๆ และ น้อง ๆ เทคโนโลยีโทรคมนาคม ผู้ให้ความช่วยเหลือยามต้องการ

เพื่อนร่วมกลุ่มโครงงาน ที่ได้ทำให้โครงงาน และปริญญานิพนธ์นี้สำเร็จลุล่วงด้วยดี

คณะผู้จัดทำขอขอบพระคุณบิดา มารดา ครูบาอาจารย์และผู้อุปถัมภ์ทุกท่าน ที่ได้อบรมสั่งสอน ให้ความช่วยเหลือ และให้กำลังแรงใจกับพวกเราด้วยดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ (ภาษาไทย)	ก
บทคัดย่อ (ภาษาอังกฤษ)	ข
กิตติกรรมประกาศ	ค
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีบทการประมวลผลสัญญาณเชิงเลข	4
2.1 นิยามของสัญญาณ	4
2.2 สัญญาณพื้นฐานของการประมวลผลสัญญาณเชิงเลข	5
2.3 ระบบสัญญาณไม่ต่อเนื่อง (Discrete-time system)	7
2.4 องค์ประกอบของระบบการประมวลผลสัญญาณเชิงเลข	8
2.5 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิตอล	11
บทที่ 3 ระบบแบบไม่ต่อเนื่อง (Discrete-time System)	13
บทที่ 4 รูปแบบของเลขฐานสองในการคำนวณทางคอมพิวเตอร์	17
4.1 การบวก/ลบเลขฐานสองที่กำหนดโดยใช้ค่าและเครื่องหมาย	20
4.2 การบวก/ลบเลขฐานสองแบบ 1 ‘s COMPLEMENT	26
4.3 การบวก/ลบเลขฐานสองแบบ 2 ‘s COMPLEMENT	30
บทที่ 5 วงจรคูณเลขฐานสอง	49
บทที่ 6 D/A และ A/D	56
6.1 ทฤษฎีการแปลงสัญญาณ (Data Acquistion and Conversion)	56
6.2 วงจรเปลี่ยนสัญญาณดิจิตอลให้เป็นอนาล็อก (DAC)	61
6.3 วงจรเปลี่ยนสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล (ADC)	63
บทที่ 7 การทดลอง และผลการทดลอง	65
7.1 การทดลองภาค A/DConverter	65
7.2 การทดลองภาค Shift Register	77
7.3 การทดลองภาค Look Up Table	79
7.4 การทดลองภาค Accumulater	94
7.5 การทดลองภาค D/A Converter	96
7.6 การทดลองชุดทดลองบอร์ดประมวลผลสัญญาณเชิงเลข	109
สรุปผล และวิจารณ์	114
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

		หน้า
รูปที่ 2-1	ลำดับที่ได้จากการสุ่มตัวอย่างฟังก์ต่อเนื่อง	5
รูปที่ 2-2	อิมพัลส์เชิงเลข	5
รปที่ 2-3	อิมพัลส์เชิงเลขที่หน่วงไป n0 วินาที	6
รูปที่ 2-4	ลำดับหนึ่งหน่วย	6
รูปที่ 2-5	ลำดับชี้กำลังลดลง	7
รูปที่ 2-6	ผังของระบบสัญญาณไม่ต่อเนื่อง	8
รูปที่ 2-7	กราฟแสดงตัวบวก	9
รูปที่ 2-8	กราฟแสดงตัวคูณค่าคงที่	9
รูปที่ 2-9	กราฟแสดงตัวคูณของสัญญาณ	9
รูปที่ 2-10	กราฟแสดงตัวหน่วงเวลา	10
รูปที่ 2-11	กราฟแสดงการเกิดล่วงหน้าหนึ่งหน่วยเวลา	10
รูปที่ 3-1	แสดงผลตอบจากสัญญาณ impulse	14
รูปที่ 3-2	แสดงการหาผลตอบของระบบโดย Convolution	15
รูปที่ 4-1	ตัวอย่างวงจรบวก/ลบที่ใช้สำหรับเลขฐานสองที่กำหนดโดยใช้ค่าและเครื่องหมาย	22
รูปที่ 4-2	แสดงไดอะแกรมของวงจรบวก/ลบเลขฐาน 2 แบบ 1's Complement ขนาด 3 บิท	28
รูปที่ 4-3	วงจรบวก/ลบ เลขฐาน 2 แบบ 1' s Complement ขนาด 5 บิท	29
รูปที่ 4-4	วงจรบวก/ลบ เลขฐาน 2 แบบ 2' s Complement ขนาด 3 บิท	32
รูปที่ 4-5	วงจรบวก/ลบ เลขฐาน 2 แบบ 2' s Complement ขนาด 5 บิท	33
รูปที่ 4-6	วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 4 บิท	37
รูปที่ 4-7	วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 8 บิท ที่ใช้วิธี FLA เข้าช่วย	38
รูปที่ 4-8	วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 24 บิท ที่ใช้วิธี SLA เข้าช่วย	40
รูปที่ 4-9	วงจรบวกเลขฐาน 2 ขนาด 16 บิท ที่ใช้วงจรบวกตัวทดแบบอนุกวมร่วมกับวงจรบวกแบบ CARRY LOOK – AHEAD	42
รูปที่ 4-11	วงจรบวกเลขฐาน 2 ขนาด32บิทที่สร้างขึ้นจากวงจรบวกในรูปที่4-10จำนวน 2 วงจร	44
รูปที่ 4-12	วงจรแปลงเลขฐาน 2 แบบอนุกรมเป็นเลข BCD แบบขนานขนาด 2 หลัก	45
รูปที่ 4-13	วงจรแปลง BCD แบบขนานขนาด 3 หลัก เป็นเลขฐาน 2 แบบขนาน	46
รูปที่ 4-14	วงจรแปลง BCD แบบอนุกรมขนาด 3 หลักเป็นเลขฐาน 2 แบบอนุกรมวงจรแปลงระหว่างเลข BCD และรหัส Excess-3	47
รูปที่ 4-15(a)	วงจรแปลงเลข BCD ขนาด 1 หลักเป็นรหัส Excess-3	48
รูปที่ 4-15(b)	วงจรแปลงเลขรหัส Excess-3 เป็นเลข BCD ขนาด 1 หลัก	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

		หน้า
รูปที่ 5-1	ไดอะแกรมของวงจรคูณเลขฐานสองขนาด 8 บิท	51
รูปที่ 5-2	ไดอะแกรมเวลาสำหรับการทำงานของวงจรคูณในรูปที่ 5-1	52
รูปที่ 5-3	ค่าดีเลย์ที่เกิดขึ้นในขั้นตอนต่าง ๆ ของการคูณ	55
รูปที่ 6-1	แสดงระบบที่มีการประมวลผลข้อมูลทางดิจิตอล	56
รูปที่ 6-2	แสดงการสุ่มสัญญาณ	58
รูปที่ 6-3	แสดง Transfer Function ของการควอนไตซ์ 4 บิท	59
รูปที่ 6-4	แสดงความผิดพลาดควอนไตซ์	60
รูปที่ 6-5	แสดงวงจรพื้นฐานของการ Sampling and Hold	60
รูปที่ 6-6	แสดงวงจร Sampling and Hold ที่ใกล้เคียงกับวงจรที่ใช้งานจริง	60
รูปที่ 6-7	แสดงการตกของแรงดันที่ hold ไว้	61
รูปที่ 6-8	แสดงรูปคลื่นสัญญาณจากวงจร Sampling and Hold ในทางอุดมคติ	61
รูปที่ 6-9	แสดง Transfer Function ของ DAC ขนาด 3 บิท	62
รูปที่ 6-10	แสดง block diagram ของ DAC	62
รูปที่ 6-11	แสดงวงจร DAC ขนาด 4 บิท แบบ R-2R ladder	62
รูปที่ 6-12	แสดงการแบ่งแรงดันจากแรงดันอ้างอิง	63
รูปที่ 6-13	แสดง function ต่าง ๆ ใน ADC comparator	63
รูปที่ 6-14	timing diagram ของ ADC แบบ SAR	63
รูปที่ 7-1	วงจรภาค A/D Converter	65
รูปที่ 7-2	วงจรภาค Shift Register	77
รูปที่ 7-3	วงจรภาค Look Up Table (Rom Accumulator)	79
รูปที่ 7-4	วงจรภาค Accumulator	94
รูปที่ 7-5	วงจรภาค D/A Converter	96
รูปที่ 7-6	สัญญาณที่ 1	109
รูปที่ 7-7	ผลลัพธ์ของสัญญาณที่ 1	110
รูปที่ 7-8	สัญญาณที่ 2	110
รูปที่ 7-9	ผลลัพธ์ของสัญญาณที่ 2	111
รูปที่ 7-10	สัญญาณที่ 3	112
รูปที่ 7-11	ผลลัพธ์ของสัญญาณที่ 3	113

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 4-1	ตารางแสดงความจริงสำหรับ output ที่เป็นบิทเครื่องหมาย	23
ตารางที่ 4-2	ตารางแสดงความจริงสำหรับ output ของวงจร XOR ที่ 2	24
ตารางที่ 4-3	ตารางแสดงความจริงสำหรับ output ของวงจร INVERTER ที่ 2	25

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1

บทนำ

การประมวลผลสัญญาณทางดิจิตอล (DSP) เป็นการนำสัญาณดิจิตอลมากระทำการทางคณิตศาสตร์ เช่น บวก, ลบ, คูณ, ถอดรากที่สอง, หรือ การอินทิเกรต ทั้งนี้ เพื่อให้ได้ผลลัพธ์ตามชนิดของงานในระบบนั้น ๆ อาทิ งานควบคุม งานสื่อสาร เป็นต้น

แต่เดิมนั้น ระบบต่าง ๆ ใช้สัญญาณอนาลอก คือ สัญญาณจะเข้าและออกจากระบบต่าง ๆ อย่างต่อเนื่องกับเวลา ระบบดังกล่าวมักมีราคาถูก และ ออกแบบง่าย แต่ข้อเสียก็คือมักมีขีดจำกัดทางด้านประสิทธิภาพ และความแม่นยำในการประมวลผล ซึ่งข้อเสียนี้กลายเป็นอุปสรรคต่อระบบสมัยใหม่ อันเป็นระบบที่ต้องการความถูกต้องแม่นยำมาก ดังนั้นจึงได้มีการปรับปรุง ระบบการประมวลผลสัญญาณ ให้เป็นระบบดิจิตอล โดยใช้เทคโนโลยีวงจรรวม (Integrated Circuit) ซึ่งได้รับการพัฒนาให้สามารถทำงานได้อย่างมีประสิทธิภาพสูงและราคาไม่แพงนัก อีกทั้งยังสามารถใช้งานร่วมกับไมโครโปรเซสเซอร์ คอมพิวเตอร์ และเกทต่าง ๆ ได้

ข้อดีของการประมวลผลสัญญาณดิจิตอลมีดังนี้

1. สัญญาณดิจิตอล เป็นสัญญาณที่มีสถานะตายตัว คือ “0” หรือ “1” ทำให้การประมวลผลมีความแน่นอนสูง การใช้งานจึงทำได้อย่างมีประสิทธิภาพ ถูกต้องแม่นยำ มีความละเอียดมากกว่าหนึ่งในพันส่วน ในขณะที่ระบบอนาลอกทำได้ไม่ดีเท่า เนื่องจากอุปกรณ์แบบอนาลอกมีคุณสมบัติ แปรค่าตามสภาพแวดล้อม เช่น อุณหภูมิ จึงทำให้ความแม่นยำและความเชื่อถือได้ของระบบต่ำ

2. มีความยืดหยุ่นมากกว่าการประมวลผลแบบอนาลอก เพราะสามารถเปลี่ยนแปลงเพียงอัลกอริธึม เพื่อให้ฮาร์ดแวร์เดียวกันทำงานหลาย ๆ อย่างได้ จึงใช้งานได้กว้างกว่า ซึ่งจะมีผลทำให้ต้นทุนโดยรวมสำหรับการใช้งานหลาย ๆ อย่างนั้น มีราคาต่ำกว่าการที่จะต้องสร้างวงจรอนาลอกขึ้นมาหลาย ๆ วงจร เพื่อทำงานทุกงานอย่างนั้น

อย่างไรก็ตาม การประมวลผลสัญญาณเชิงดิจิตอล ก็มีจุดบกพร่องอยู่เช่นกัน คือ

1. เนื่องจากสัญญาณต่าง ๆ ทางธรรมชาติเป็นแบบอนาลอก ดังนั้นการใช้งานของระบบสัญญาณดิจิตอลร่วมกับอนาลอก จึงจำเป็นต้องมีอุปกรณ์ประเภท A/D และ D/A เข้ามาช่วย ซึ่งอุปกรณ์ดังกล่าว ที่มีประสิทธิภาพและความเร็วสูงนั้นมักมีราคาแพง

2. ระบบสัญญาณดิจิตอล มักมีข้อจำกัดทางด้านแถบความถี่ (Bandwidth) เนื่องจากความเร็วสูงสุดในการทำงานของวงจรดิจิตอลมีค่าจำกัด ซึ่งในปัจจุบันความเร็วดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังมีค่าต่ำมาก ทำให้การประมวลผลมีความเร็วต่ำ และใช้ได้กับสัญญาณที่มีแถบความถี่ไม่กว้างมากเท่าระบบสัญญาณอนาลอก แต่อย่างไรก็ตาม เทคโนโลยี IC ได้ถูกพัฒนาให้มีความเร็วสูงขึ้น จึงเป็นไปได้ว่าในอนาคตอันใกล้นี้ระบบสัญญาณดิจิตอลจะสามารถใช้กับแถบความถี่กว้าง ๆ ได้

ปัจจุบันการใช้การประมวลผลสัญญาณทางดิจิตอล เข้ามามีบทบาทกับหลาย ๆ อย่างเช่น งานควบคุมระบบ, งานการฟฟิก, ใช้ในเครื่องมือวัดต่าง ๆ เช่น เครื่องวิเคราะห์สเปคตรัม (Spectrum analyzer), เครื่องสร้างสัญญาณ (Function generator), งานประมวลผลทางเสียง เช่น การจดจำเสียงพูด (Speech recognition), การวิเคราะห์เสียง (Speech synthesis), ใช้ในงานทางด้านการสื่อสาร เช่น การกำจัดเสียงสะท้อน (Echo cancellation), ตู้สาขาโทรศัพท์ดิจิตอล (Digital PBXs), อีควอไลเซอร์ปรับค่าได้ (Adaptive equalizers) เป็นต้น ในระบบต่าง ๆ ที่กล่าวมานี้จำเป็นอย่างยิ่งที่จะต้องใช้การกรองเชิงเลข (Digital Filtering) ซึ่งเป็นการประมวลสัญญาณทางดิจิตอลชนิดหนึ่ง ที่ถูกนำมาใช้ในเกือบทุกระบบ การกรองเชิงเลขมีหลายประเภท และมักเกี่ยวเนื่องกับการกำจัดสัญญาณรบกวน (noise) และการลดความพร่าเพี้ยน (Distortion) ดังนั้นจึงเป็นการดีสำหรับการเริ่มต้นศึกษาเรื่องการประมวลผลสัญญาณทางดิจิตอล โดยจะเริ่มศึกษาเกี่ยวกับ "การกรองเชิงเลข" ก่อน ดังจะมีรายละเอียดเรื่องนี้ในบทต่อ ๆ ไป

ในงานทางด้านการสื่อสารโทรคมนาคมในปัจจุบัน การสื่อสารระบบดิจิตอลได้ถูกนำมาใช้อย่างแพร่หลาย เนื่องจากสาเหตุดังได้กล่าวข้างต้น ดังนั้น ในปริญญานิพนธ์ฉบับนี้จึงเน้นศึกษาการประมวลผลสัญญาณดิจิตอล เพื่อนำมาใช้ในงานทางการสื่อสารโทรคมนาคม

งานที่สามารถประยุกต์ใช้ได้

มีงานหลายอย่างที่ได้นำเอาการประมวลผลสัญญาณดิจิตอลไปใช้งาน เราคงจะยกตัวอย่างได้แค่ส่วนหนึ่งของมันซึ่งได้แก่

1. การประมวลผลเสียง เช่น การเข้า code เสียง, การรู้จำเสียง (Speech Recogntion), Sound effect การผสมเสียง , การกรองสัญญาณรบกวน , Synthesizer เป็นต้น
2. ในระบบสื่อสารได้แก่ Modulation /Demodulation , Fax , Mobile computing , mobile phone system, echo cancellation, adaptive filter, adaptive equalizer, adaptive antennaarray, network interfaces, radar, sonar, navigator system , GPS เป็นต้น
3. ในระบบควบคุม Digital Control System ได้แก่ process control , เครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในวงการแพทย์ ได้แก่ biomedical, bio-signal analysis/processing, เครื่องช่วยได้ยิน (hearing aid)
5. การประมวลผลแบบหลายมิติ สามารถประยุกต์ใช้ในการประมวลผล ภาพ , วีดีโอ, holography (ภาพสามมิติ) ได้
6. ในการวิเคราะห์สถิติ และทำนายค่าในอนาคต

ข้อดีของการใช้การประมวลผลสัญญาณดิจิตอล

ข้อดีของการใช้การประมวลผลสัญญาณดิจิตอล ที่เหนือกว่าการใช้วงจรในระบบอนาลอก มีดังนี้

1. ความสามารถในการโปรแกรมได้ ทำให้ง่ายต่อการออกแบบ, เปลี่ยนแปลงแก้ไข, และทดสอบ สำหรับวงจรอนาลอก ถ้าต้องการเปลี่ยนคุณสมบัติอะไรบางอย่าง อาจหมายถึงการต้องออกแบบวงจรใหม่หมดเลย
2. ความถูกต้องแม่นยำที่ดีกว่าความถูกต้องของการประมวลผลสัญญาณดิติตอล ขึ้นอยู่กับจำนวนบิตที่ใช้แทนสัญญาณ และพารามิเตอร์ต่าง ๆ ซึ่งมีความยืดหยุ่น และควบคุมได้ง่าย คือ ใน งานที่เราต้องการความแม่นยำสูง เราก็จะใช้จำนวนบิตที่มากขึ้น อีกทั้งในช่วงของการออกแบบการจำลองระบบที่ออกแบบในคอมพิวเตอร์ จะให้ผลที่ตรงกับความเป็นจริงเมื่อนำออกไปสร้างเป็นวงจรจริง
3. มีเสถียรภาพที่ไม่ขึ้นกับเวลา และอุณหภูมิ
4. เกี่ยวข้องโดยตรงกับเทคโนโลยีคอมพิวเตอร์ และ VLSI ซึ่งกำลังมีความเจริญก้าวหน้าอย่างรวดเร็ว ดังนั้น DSP chip จึงมีความก้าวหน้าในทำนองเดียวกัน ทั้งความเร็วที่สูงขึ้นและ ราคาก็ถูกลง เรามักพบว่า การประยุกต์ใช้การประมวลผลสัญญาณดิจิตอลในงานหลายๆอย่าง มีต้นทุนที่ต่ำกว่าวงจรแบบอนาลอกเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีบทการประมวลผลสัญญาณเชิงเลข

ก่อนที่เราจะทำการกล่าวถึงทฤษฎีการประมวลผลสัญญาณเชิงเลข ขอให้ทำความเข้าใจกับคำนิยามของศัพท์เทคนิคที่เกี่ยวข้องกับระบบการปรกะมวลผลสัญญาณเชิงเลข เพื่อให้การสื่อความหมายและความเข้าใจเป็นแบบอย่างเดียวกัน

2.1 นิยามของสัญญาณ

1. สัญญาณเชิงอุปมาน (Analog signal)ใช้กับการกล่าวถึงสัญญาณที่มีรูปคลื่น (waveform) ที่แปรไปอย่างต่อเนื่องกับพิสัยเวลา โดยที่แอมพลิจูด (amplitude) หรือค่าขนาดของสัญญาณก็มีการแปรไปอย่างต่อเนื่องด้วย เช่น สัญญาณไซน์ (sin)

2. สัญญาณเวลาต่อเนื่อง (Continuous-time signal) สัญญาณแบบนี้รูปคลื่นของสัญญาณแปรค่าไปอย่างต่อเนื่องกับพิสัยเวลา แต่แอมพลิจูดไม่ได้เจาะจงว่าต้องแปรไปอย่างต่อเนื่องซึ่งก็หมายความว่าอาจแปรอย่างไม่ต่อเนื่องก็ได้ เพราะฉะนั้นอาจกล่าวได้ว่า สัญญาณเชิงอุปมานเป็นสัญญาณเชิงเวลาต่อเนื่องได้

3. สัญญาณเชิงเวลาเต็มหน่วย (Discrete-time Signal) เป็นสัญญาณ x(t) ที่ค่าของฟังก์ชั่น กำหนดเฉพาะเซตของเวลาที่แน่นอนอันหนึ่งเท่านั้น สัญญาณแบบนี้อาจแบ่งตามลักษณะของแอมพลิจูดได้เป็น 2 แบบ คือ

3.1 สัญญาณเชิงข้อมูลเต็มหน่วย (Discrete data signal) สัญญาณแบบนี้ แอมพลิจูดจะต่อเนื่อง หรือกล่าวได้ว่าแอมพลิจูดมีค่าเท่ากันทุกประการกับสัญญาณเชิงอุปมานที่เป็นตัวต้นแบบในการสุ่มตัวอย่าง (sampling) ตัวอย่างเช่น สัญญาณออกของ วงจรสุ่มและคงค่าสัญญาณ จะจัดเป็นสัญญาณเชิงเต็มหน่วย สัญญาณแบบนี้บางที่เรียกว่าข้อมูล หรือ สัญญาณเชิงเต็มหน่วย

3.2 สัญญาณเชิงเลข (Digital signal) สัญญาณแบบนี้แอมพลิจูดของสัญญาณมีค่าเฉพาะเซตของค่าที่แน่นอนเซตหนึ่งเท่านั้น เช่นสัญญาณที่ออกจากวงจร A/D

4. ระบบเวลาจริง (Real-time system) โดยทั่วไปคำนี้ใช้กับระบบการประมวลผลสัญญาณที่การคำนวณการประมวลผลทำได้เสร็จสิ้นก่อนที่จะมีการสุ่มตัวอย่างสัญญาณลำดับใหม่อย่างไรก็ตามคาบเวลาในการสุ่มตัวอย่างสัญญาณควรมีค่าน้อย

5. ลำดับ (Sequence) หนังสือบางเล่มอาจเรียกว่าเป็น ลำดับข้อมูล หรือ ลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

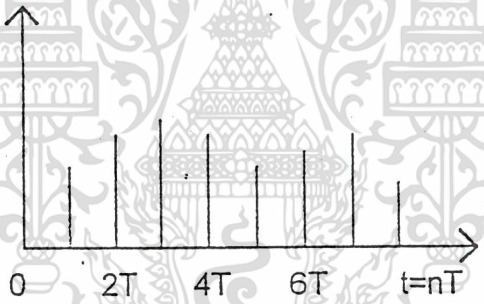
สัญญาณหรือลำดับ ก็ได้ ซึ่งคำว่า ลำดับ อาจใช้แทนสัญญาณเชิงเต็มหน่วย หรือ สัญญาณเชิงเลขก็ได้

2.2 สัญญาณพื้นฐานของการประมวลผลสัญญาณเชิงเลข

1. ลำดับสัญญาณ (sequence) ถ้าหากเรามีสัญญาณเชิงอุปมาน x(t) แล้วทำการสุ่มตัวอย่าง (sampling interval) T วินาทีเท่านั้นและเท่ากันโดยตลอดสิ่งที่ได้ต่อไปนี้เรียกว่าลำดับ ซึ่งเขียนแทนได้ดังสมการ

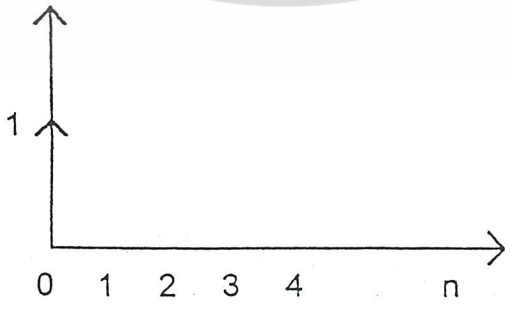
$$x(n) = x(nT) = x(t)\Big|_{t=nT} \quad ; -\infty < n < \infty$$

โดยที่ n = 0, 1, 2,... ซึ่งเป็นค่าคงตัวเต็มหน่วย และโดยทั่วไปใช้ T = 1 วินาที สมการนี้แสดงว่าลำดับ x(t) ถูกสุ่มตัวอย่างมาจากสัญญาณเชิงอุปมาน x(t) ทุกคาบเวลาใน nT วินาที



รูปที่ 2-1 ลำดับที่ได้จากการสุ่มตัวอย่างฟังก์ชั่นต่อเนื่อง

2. อิมพัลส์เชิงเลข (Digital impulse) ซึ่งนิยามได้เป็น

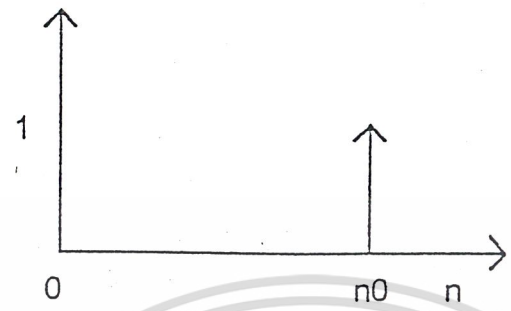


รูปที่ 2-2 อิมพัลส์เชิงเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อิมพัลส์เชิงเลขที่หน่วงเวลาไป n_0 วินาที นิยามได้เป็น

$$\delta(n-n_0) = 1 \quad ; \quad n = n_0$$



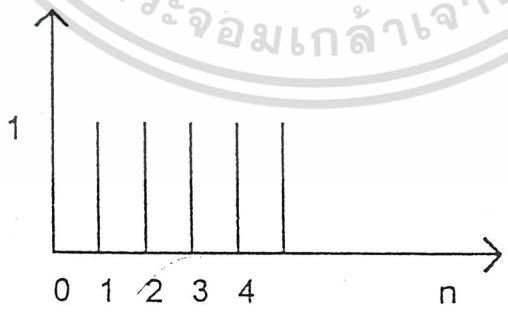
รูปที่ 2-3 อิมพัลส์เชิงเลขที่หน่วงไป n0 วินาที

4. ลำดับหนึ่งหน่วย (Unit step sequence) นิยามได้เป็น

$$u(n) = 1 \quad ; \quad n \geq 0$$
$$0 \quad ; \quad n \leq 0$$

ซึ่งลำดับหนึ่งหน่วยนี้อาจเขียนอยู่ในรูปแบบของผลบวกของอิมพัลส์เชิงเลขได้คือ

$$u(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m)$$

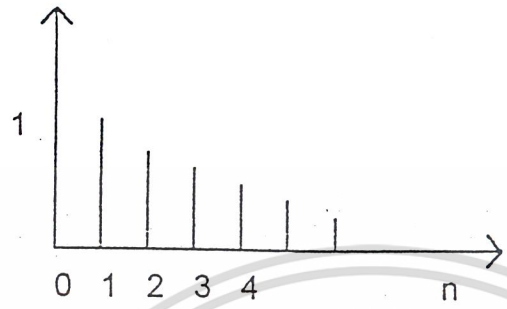


รูปที่ 2-4 ลำดับหนึ่งหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ลำดับชี้กำลังลดลง (Decay exponential) เมื่อ a < 1 ลำดับแบบนี้สามารถนิยามได้ คือ

$$g(n) = a^n \; ; \; n \geq 0$$
$$= 0 \; ; \; n \leq 0$$



รูปที่ 2-5 ลำดับชี้กำลังลดลง

ลักษณะรูปร่างของลำดับสัญญาณที่กล่าวถึงนี้ ได้แสดงรวมกันไว้ในรูป 2-2 ...2-5 จากนิยามเหล่านี้ ถ้าหากเรามีลำดับสัญญาณเชิงเลขอยู่ชุดหนึ่งมีลำดับเป็น x(0), x(1), x(2),..., x(n) เราอาจเขียนแทนลำดับสัญญาณเหล่านี้ใหม่ ให้อยู่ในพจน์ของอิมพัลส์เชิงเลขที่หน่วงออกไปได้คือ

$$x(n) = [x(0), x(1), x(2), ..., x(m)]$$
$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m)$$

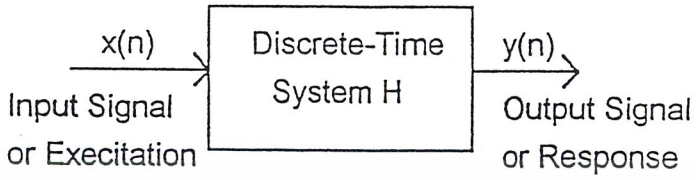
2.3 ระบบสัญญาณไม่ต่อเนื่อง (Discrete-time system)

ในการใช้งานของระบบประมวลผลสัญญาณเชิงเลข ระบบนี้จะมีอินพุทเป็นสัญญาณไม่ต่อเนื่อง และประมวลผลสัญญาณได้เอาท์พุทเป็นสัญญาณ ไม่ต่อเนื่องเช่นกัน โดยปรกติระบบจะประมวลผลสัญญาณ x(n) เพื่อให้ได้สัญญาณเอาท์พุท y(n) หรืออาจกล่าวได้อีกอย่างว่า คือการแปลงสัญญาณ x(n) ของระบบให้เป็นสัญญาณ y(n) สมการที่ใช้แสดงความสัมพันธ์ระหว่าง x(n) กับ y(n) คือ

$$x(n) = H[x(n)]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ H คือการเปลี่ยนแปลง (Transformation) ซึ่งเป็นการประมวลผลของระบบที่มีต่อสัญญาณ x(n) นั่นเอง ความสัมพันธ์ทางคณิตศาสตร์ของสมการ สามาถแสดงได้ดังรูป 2-6



รูปที่ 2-6 ผังของระบบสัญญาณไม่ต่อเนื่อง

มีหลายวิธีที่ใช้อธิบายถึงคุณลักษณะของระบบซึ่งเป็นการประมวลผล x (n) ให้ได้ y(n) ในบทนี้เราจะกล่าวถึงลักษณะกว้าง ๆ ของระบบ หากต้องการศึกษารายละเอียดสามารถหาอ่านได้จากตำราหรือเอกสารที่อ้างอิงท้ายเล่ม

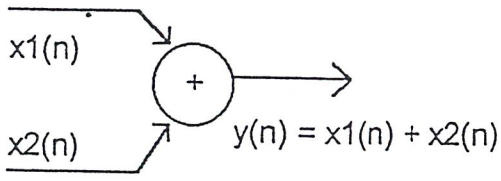
2.4 องค์ประกอบของระบบการประมวลผลสัญญาณเชิงเลข

หลังจากทำความเข้าใจพื้นฐานของสัญญาณที่ใช้ในการประมวลผลแล้ว เราสนใจกับระบบของสัญญาณไม่ต่อเนื่อง ในระบบการประมวลผลสัญญาณไม่ต่อเนื่องจะมีองค์ประกอบพื้นฐานอันได้แก่ ตัวบวก , ตัวคูณค่าคงที่ , ตัวคูณสัญญาณ , ตัวหน่วงหน่วยเวลา และ ตัวล่วงหน้าหนึ่งหน่วยเวลา ในที่นี้จะแสดงให้เห็นถึงบล็อคไดอะแกรมพื้นฐานที่จะถูกนำไปใช้ต่อเชื่อมกับระบบที่ซับซ้อนยิ่งขึ้น

2.4.1 ตัวบวก (ADDER)

จากรูป 2-7 ข้างล่าง แสดงให้เห็นถึงระบบ (ตัวบวก) ซึ่งจะทำงานโดยการรวมลำดับของสัญญาณสองลำดับเข้าด้วยกัน ซึ่งผลการรวมของสัญญาณคือ y(n) สังเกตว่าไม่จำเป็นต้องมีการเก็บจดจำลำดับสัญญาณใดสัญญาณหนึ่งเพื่อที่จะนำมาใช้ในการรวมสัญญาณ

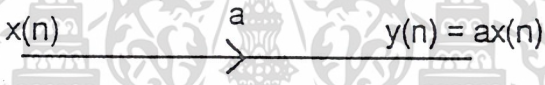
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-7 กราฟแสดงตัวบวก

2.4.2 **ตัวคูณของค่าคงที่ (CONSTANT MULTIPLIER)**

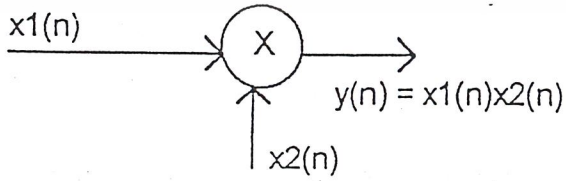
การทำงานของตัวคูณของค่าคงที่จะเป็นไปดังรูปที่ 2-8 ซึ่งแสดงให้เห็นง่าย ๆ ว่าเป็นการใช้ เฟคเตอร์ในการสเกลค่า Input x(n) การทำงานนี้จะไม่ต้องการใช้หน่วยความจำเลย (memoryless)



รูปที่ 2-8 กราฟแสดงตัวคูณค่าคงที่

2.4.3 **ตัวคูณสัญญาณ (SIGNAL MULTIPLETS)**

รูปที่ 2-9 แสดงให้เห็นถึงการคูณของสองลำดับสัญญาณ ซึ่งจะให้ลำดับสัญญาณผลลัพธ์เป็น y(n) เช่นเดียวดับสองขบวนการที่ผ่านมา จะพบว่าการทำงานของการคูณสัญญาณนี้ไม่ต้องใช้หน่วยความจำเช่นกัน



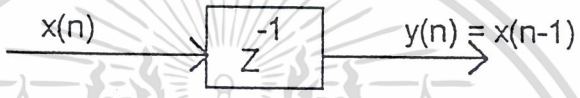
รูปที่ 2-9 กราฟแสดงการคูณของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ตัวหน่วงหนึ่งหน่วยเวลา (UINT DELAY ELEMENT)

ตัวหน่วงหนึ่งหน่วยเวลา เป็นระบบเฉพาะที่ใช้สำหรับหน่วงสัญญาณที่จะผ่านตัวมัน ให้สัญญาณภายหลังไปหนึ่งแซมเปิล รูปที่ 2-10 เป็นการแสดงถึงระบบดังกล่าวถ้าหาก input ของระบบเป็น x(n) ก็จะได้ output เป็น xd(n-1) โดยความเป็นจริงแล้วแซมเปิลที่ x(n-1) จะถูกเก็บไว้ในหน่วยความจำ ณ เวลาที่ (n-1) และสัญญาณดังกล่าวจะถูก เรียกออกจากหน่วยความจำที่เวลา (n)

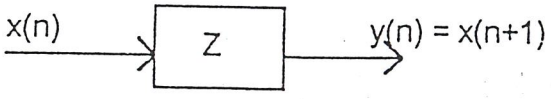
ตัวหน่วงนี้จำเป็นต้องใช้หน่วยความจำ โดยสัญญลักษณ์ Z^{-1} จะหมายถึงการหน่วงสัญญาณไปหนึ่งหน่วยเวลา



รูปที่ 2-10 กราฟแสดงตัวหน่วงเวลา

2.4.5 ตัวล่วงหน้าหนึ่งหน่วยเวลา (UNIT ADVANCE ELEMENT)

ตัวล่วงหน้าหนึ่งหน่วยเวลานี้จะทำงานตรงข้ามกับตัวหน่วงหนึ่งหน่วยเวลา ตัวล่วงหน้านี้จะเลื่อนสัญญาณ input x(n) ให้เกิดก่อนหนึ่งแซมเปิล กล่าวคือให้ปรากฏเป็น x(n-1) ณ เวลา (n) รูปที่ 2-11 แสดงให้เห็นถึงการทำงานดังกล่าวโดยตัว Z จะเป็นสัญลักษณ์ของการเกิดล่วงหน้าหนึ่งหน่วยเวลา ในทางกายภายคงไม่เกิดลักษณะการทำงานดังกล่าว ทั้งนี้เพราะเป็นการมองเห็นสัญญาณที่เกิดล่วงหน้า แต่ถ้าเราเก็บสัญญาณลงในหน่วยความจำของเครื่องคอมพิวเตอร์ เราสามารถเรียกสัญญาณแซมเปิลใด ๆ ณ เวลาใดก็ได้



รูปที่ 2-11 กราฟแสดงการเกิดล่วงหน้าหนึ่งหน่วยเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

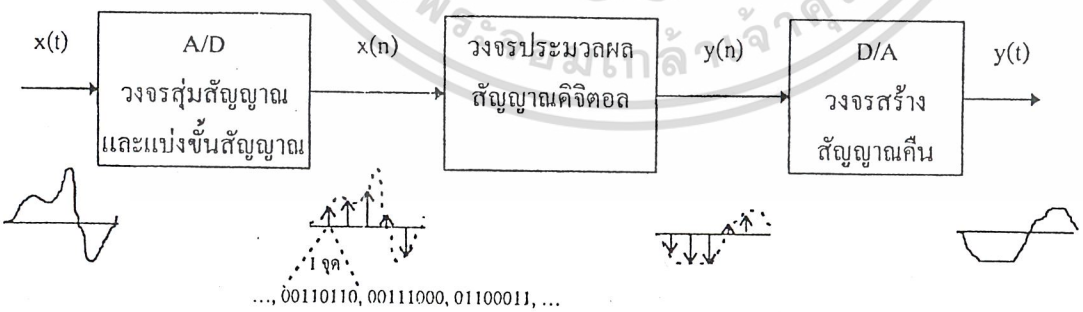
2.5 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิตอล

2.5.1 วงจรสุ่มสัญญาณ สัญญาณขาเข้าของวงจรนี้เป็นสัญญาณอนาลอกซ์หรือสัญญาณต่อเนื่องซึ่งจะเขียนเป็นสัญลักษณ์ได้เป็น x(t) โดยที่ t แทนเวลาที่มีความต่อเนื่อง ส่วนสัญญาณขาออกเป็นสัญญาณไม่ต่อเนื่องซึ่งเขียนสัญลักษณ์ได้เป็น x(n) โดยที่ n เป็นตัวชี้ที่มีค่าไม่ต่อเนื่อง โดยทั่วไปจะใช้ n เป็นจำนวนเต็มนั่นคือ n = ..., - 2 , -1 , 0 , 1 , 2 , ... ถ้าให้ความถี่ของการสุ่มเป็น f_s และค่าการสุ่มเป็น $T = 1/f_s$ โดยสุ่มที่ทุกๆ เวลา = nT จะได้ว่า $x(n) = x(t) \mid_{t=nT}$

2.5.2 วงจร Quantization หรือวงจรแบ่งขั้นสัญญาณ สัญญาณ x(n) ถึงแม้มีความไม่ต่อเนื่องในเชิงเวลา แต่ยังมีความต่อเนื่องในเชิงขนาดอยู่ วงจรนี้จะทำการแบ่งขั้นของสัญญาณและเข้า code เป็นเลขฐานสองซึ่งจะให้สัญญาณขาออกเป็นสัญญาณดิจิตอล(แทนด้วยรหัสของค่า 0กับ1)ความละเอียดในการแบ่งขั้นขึ้นอยู่กับจำนวน bit ที่ใช้แทนแต่ละจุดของ x(n)

วงจรสุ่มสัญญาณ รวมกับการแบ่งขั้นสัญญาณ ในทางปฏิบัติก็คือตัวแปลงสัญญาณอนาลอกซ์ เป็นดิจิตอล (A/D converter)

คำว่าสัญญาณดิจิตอลกับสัญญาณไม่ต่อเนื่องโดยทั่วไปหมายถึง สัญญาณในลักษณะเดียวกัน แต่สัญญาณไม่ต่อเนื่องหมายถึง สัญญาณที่มีความไม่ต่อเนื่องในเชิงเวลาเท่านั้น (Discretetime) นั่นคือสัญญาณขาออกของ ข้อ 1 นั่นเอง ส่วนคำว่าดิจิตอลมักหมายถึงสัญญาณไม่ต่อเนื่องที่แต่ละจุดถูกแบ่งขั้นและเข้ารหัสอยู่ในรูป 0 กับ 1 แล้ว จะพูดถึงสัญญาณไม่ต่อเนื่องเป็นส่วนใหญ่ นั้นก็คือ เราจะละเลยผลของการแบ่งขั้นสัญญาณชั่วคราวและถือเอาว่าสัญญาณ x(n) เป็นสัญญาณขาเข้าของวงจรประมวลผลสัญญาณเลย อย่างไรก็ตามเราจะย้อนกลับมาดูความคลาดเคลื่อนของระบบที่เราออกแบบ ซึ่งเป็นผลมาจากการแบ่งขั้นสัญญาณ



รูปที่ 2.1 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิตอล

2.5.3 วงจรประมวลผลสัญญาณ เป็นวงจรประมวลผลสัญญาณ x(n) เพื่อประมวลผลบางอย่างเช่น เป็นวงจรกรองความถี่และให้สัญญาณขาออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น y(n) ข้อนี้เป็นส่วนหลักที่เราจะเกี่ยวข้อง วงจรประมวลผลสัญญาณแท้ที่จริงก็คือการคำนวณนั่นเอง ถ้าเรามองที่สัญญาณขาออกขณะหนึ่ง ๆ เช่น y(10) จะได้ว่า y(10) เกิดจากการประมวลผลกัน (บวก,คูณ...) ระหว่างสัญญาณขาเข้าในปัจจุบันคือ x(10), สัญญาณขาเข้าในอดีตได้แก่ x(9), x(8),..., สัญญาณขาเข้าในอนาคตได้แก่ x(11),x(12),..., อาจรวมถึงสัญญาณขาออกในอดีตด้วย ได้แก่ y(9), y(8),... การคำนวณดังกล่าวสามารถกระทำโดย

- การเขียน software เพื่อใช้กับ CPU ทั่วๆ ไป ซึ่งอาจเป็นการกระทำในคอมพิวเตอร์หรือใช้วงจรที่ต่อเองและมี ไมโครโปรเซสเซอร์ควบคุม
- การใช้ CPU ที่ออกแบบเฉพาะสำหรับงานประมวลผลสัญญาณหรือเรียกกันทั่ว ๆ ไปว่า Digital Signal Processors หรือ DSP chips ซึ่ง CPU ประเภทนี้ถูกออกแบบมาให้สามารถทำฟังก์ชั่นที่ใช้ในการประมวลผลสัญญาณได้อย่างรวดเร็ว บางชนิดยังสามารถทำการประมวลผลหลาย ๆ ส่วนได้พร้อมกันในตัวเดียว (Multi-Processing) ด้วยบริษัทที่เป็นผู้นำด้านการผลิต CPU ประเภทนี้ได้แก่ Texas Instruments , Motorola , Analog Devices , และ AT&T เป็นต้น
- การใช้ Hardware หรือ IC chip ที่ออกแบบเฉพาะเพื่อทำงานประมวลผลสัญญาณเฉพาะทางการเลือกใช้ก็ขึ้นอยู่กับลักษณะงานและต้นทุนสำหรับงานประมวลผลสัญญาณแบบ Real time อาจต้องใช้ DSP chip หรือ Hardware พิเศษ ในการทำการประมวลผล

2.5.4 **วงจรสร้างสัญญาณคืน** (Reconstruction) ใช้ในระบบที่มีสัญญาณขาออกสุดท้ายเป็นสัญญาณต่อเนื่องโดยทำหน้าที่แปลงสัญญาณไม่ต่อเนื่อง y(n) ให้กลับเป็นสัญญาณต่อเนื่อง y(t) ซึ่งเป็นสัญญาณขาออกสุดท้ายของระบบ วงจรประเภทนี้ก็คือ วงจรที่แปลงสัญญาณดิจิตอลเป็นอนาลอกซ์นั้งเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบแบบไม่ต่อเนื่อง (Discrete-time Systems)

ในบทนี้จะกล่าวถึงระบบแบบไม่ต่อเนื่องซึ่งระบบการประมวลผลสัญญาณดิจิตอลที่จะศึกษาต่อไปก็จะเป็นระบบไม่ต่อเนื่องนี้ จึงมีความจำเป็นที่จะต้องเข้าใจเนื้อหาส่วนนี้ให้ดี ระบบทั้ง 2 แบบมีลักษณะการคิดและวิเคราะห์คล้ายคลึงกันไม่ว่าในเรื่องของความเป็นเชิงเส้นและไม่แปรตามเวลา, การหาผลตอบสนอง , การแปลงระหว่างสัญญาณในเชิงเวลากับความถี่ , เสถียรภาพ และอื่น ๆ

ความเป็นเชิงเส้น และไม่แปรตามเวลา (Linearity and time invariance)

ระบบที่เราจะสนใจเป็นพิเศษในเบื้องต้นของการประมวลผลสัญญาณ คือ ระบบแบบเชิงเส้นและไม่แปรตามเวลา ก่อนที่จะดูถึงความสำคัญของระบบแบบนี้ เรามาดูวิธีคิดก่อนว่าระบบแบบใดมีคุณสมบัติดังกล่าว

สมมุติว่าระบบแบบไม่ต่อเนื่องที่เราสนใจระบบหนึ่ง เมื่อป้อนสัญญาณขาเข้า $x_1(n)$ ทำให้เกิดสัญญาณขาออก $y_1(n)$ และเมื่อป้อนสัญญาณขาเข้า $x_2(n)$ ทำให้เกิดสัญญาณขาออก $y_2(n)$

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

$$x(n) = a_1x_1(n) + a_2x_2(n)$$

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

$$y(n) = a_1y_1(n) + a_2y_2(n)$$

นั่นคือ จะต้องได้ $y(n)$ เป็นการคำนวณแบบเชิงเส้นระหว่าง $y_1(n)$ และ $y_2(n)$ โดยที่มีสัมประสิทธิ์ ที่ใช้ (a_1 และ a_2) ตัวเดียวกันกับสัญญาณขาเข้า ระบบเชิงเส้นบอกเราว่า

- ถ้าเราคูณสัญญาณขาเข้าด้วย ค่าสัมประสิทธิ์ค่าหนึ่ง สัญญาณขาออกก็จะเปลี่ยนไปด้วยตัวคูณเดียวกัน
- ถ้าสัญญาณขาเข้าเป็นผลรวมของสัญญาณหลาย ๆ สัญญาณ สัญญาณขาออกก็จะเป็นผลรวมของสัญญาณขาออก ที่เกิดจากสัญญาณขาเข้า แต่ละตัวมารวมกัน

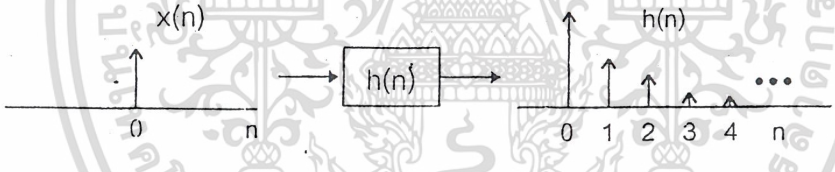
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของระบบที่เป็นเชิงเส้นนี้บางทีถูกเรียกว่า คุณสมบัติ superposition สำหรับคุณสมบัติความไม่แปรตามเวลาของระบบไม่ต่อเนื่องมีเงื่อนไขว่าถ้าสัญญาณขาเข้า x(n) ทำให้เกิดสัญญาณขาออก Y(n) แล้ว สัญญาณขาเข้า x(n-k) จะทำให้เกิดสัญญาณขาออก y(n-k) โดยที่ k เป็น จำนวนเต็มใด ๆ

ระบบแบบไม่แปรเวลาบอกเราว่า ไม่ว่าเราจะใส่สัญญาณขาเข้า ที่เวลาใด เราจะได้สัญญาณขาออกที่เหมือนเดิม นั่นก็คือระบบเราไม่มีค่าพารามิเตอร์ใดๆ ที่แปรตามเวลา หรือแปรตามสภาวะแวดล้อมใด ๆ ที่แปรตามเวลา

ผลตอบจากสัญญาณ impulse (impulse response)

ระบบที่เป็นเชิงเส้นและไม่แปรตามเวลา มีคุณลักษณะพิเศษคือ สามารถระบุคุณลักษณะของระบบได้โดยสมบูรณ์ด้วย ผลตอบจากสัญญาณ impulse ผลตอบจากสัญญาณ impulse ซึ่งเขียนสัญลักษณ์แทนว่า h(n) หมายถึง สัญญาณขาออกของระบบ เมื่อสัญญาณขาเข้าเป็นสัญญาณ impulse s(n) ซึ่งมีค่าเป็น 1 ที่ n = 0 ดังแสดงในรูป



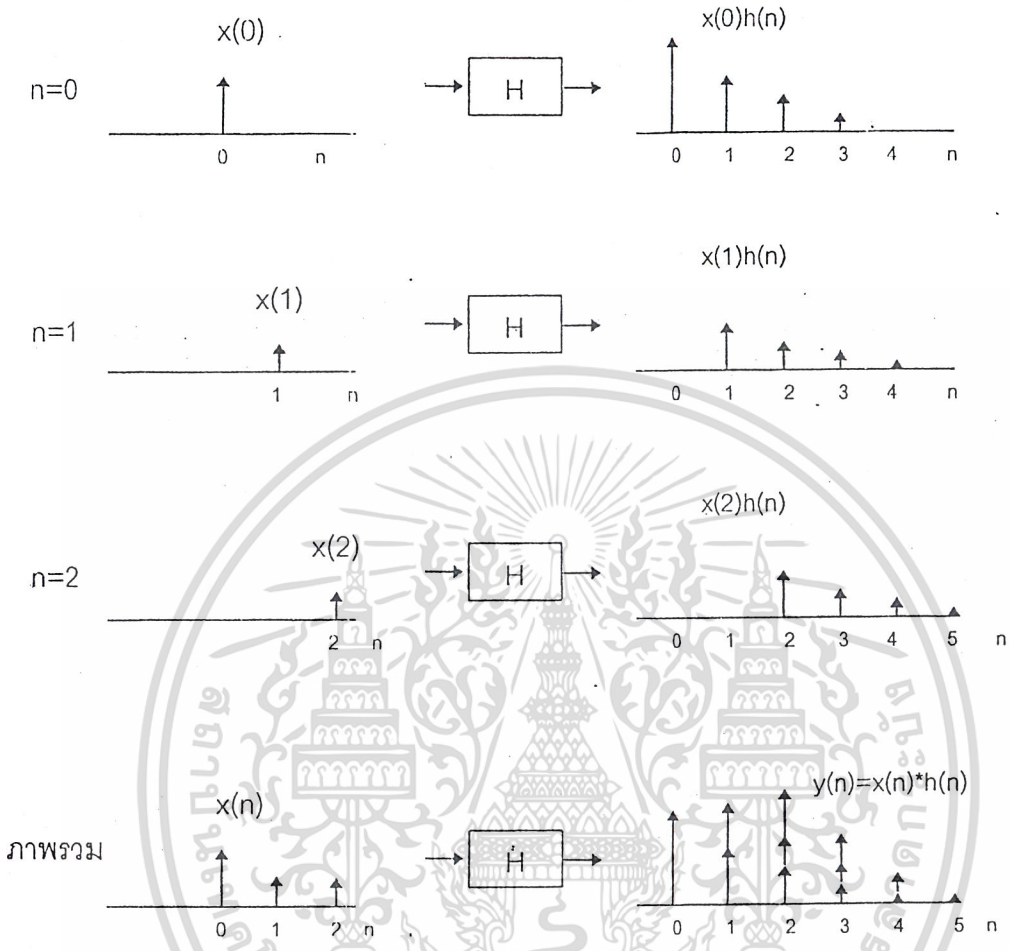
รูปที่ 3.1 แสดงผลตอบจากสัญญาณ impulse

ลองดูว่าเมื่อเรารู้ผลตอบสนองจากสัญญาณ impulse เราจะสามารถหาผลจากสัญญาณตอบสนอง เมื่อสัญญาณขาเข้าเป็นสัญญาณใด ๆ ได้อย่างไร สมมุติว่าสัญญาณขาเข้าเป็นสัญญาณไม่ต่อเนื่องใด ๆ x(n) เพื่อให้ง่ายต่อการวิเคราะห์ เราจะสมมุติว่า x เริ่ม มีค่าเมื่อ n = 0 เราสามารถเขียน x(n) ได้เป็นผลรวมของสัญญาณ impulse ที่มีขนาดเท่ากับ x(n) มีค่า n = 0, 1 , 2 ,... นั่นคือ

$$X(n) = x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \ldots$$

ด้วยคุณสมบัติความไม่แปรตามเวลาของระบบเมื่อ s(n) ทำให้เกิดผลตอบ h(n) ก็จะได้ว่า s(n-1) ทำให้เกิดผลตอบ h(n-1) และ s(n-2) ทำให้เกิดผลตอบ h(n-2) เป็นเช่นนี้ไปเรื่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการหาผลตอบของระบบโดย Convolution

อาศัยคุณสมบัติความเป็นเชิงเส้นของระบบเราก็จะได้ว่า x(0)s(n) ทำให้เกิดผลตอบ x (0)h(n) และ x(1)s(n-1) ทำให้เกิดผลตอบ x(1)h(n-1) เป็นเช่นนี้ไปเรื่อย ๆ ดังแสดงในรูป ดังนั้นผลตอบ y(n) ที่เกิดจาก x(n) ก็จะมาจากผลรวมของผลตอบแต่ละตัวนั่นคือ

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

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

ถ้าไม่จำกัดว่า x(n) เริ่มมีค่าที่ n = 0 ก็จะได้เป็นสมการทั่วไปของผลตอบของระบบเป็น

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

y(n) ที่ได้นี้ก็คือ ผลคูณ convolution แบบไม่ต่อเนื่องนั่นเอง เรานิยามสัญลักษณ์ * แทนการกระทำนี้และจะสามาถเขียนสมการของ y(n) ได้ใหม่เป็น

$$y(n) = x(n) * h(n)$$

ถ้าเราแทน m ด้วย n - k ในสมการของ convolution ได้ว่า

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

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

นี่ก็คืออีกรูปหนึ่งในการหา convolution ซึ่งเราจะพบว่ารูปแบบหลังนี้ ใช้ (n- k เป็น index ของ x)สามารถนำมาใช้งานได้สะดวกกว่า m และ k ในสมการขึ้นต้นเป็นเพียง index เพื่อให้เกิดการบวกกันใน $\sum$ เท่านั้น แต่ index ของเวลาจริง ๆ ก็คือค่า n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 4

รูปแบบของเลขฐานสองในการคำนวณทางคอมพิวเตอร์

การบวกเลขฐานสองมักจะต้องเกี่ยวข้องกับการบวกเลขฐานสองที่มีค่าทั้งบวกและลบ ดังนั้นในที่นี้จะได้กล่าวถึงการกำหนดเครื่องหมาย บวก/ลบ ของเลขฐานสองรวมถึงการบวกเลขฐานสองที่มีค่าเป็นบวกและลบ

ในการที่จะกำหนดค่าบวกหรือลบให้กับเลขฐานสองนั้นเราจำเป็นต้องเข้าใจถึงเลข 1's และ 2's Complement ของระบบเลขฐานสองก่อน สำหรับกรณีของเลข 2's Complement นั้นจะเรียกได้อีกอย่างว่า Radix Complement ของเลขฐานสิบ) ซึ่งค่า Radix Complement ของเลขฐานใดๆนั้น เราสามารถคำนวณได้จากสมการ :

$$A+A^* = r^N$$

หรือ $$A^* = r^N - A$$

โดยที่ A คือ เลขที่เราต้องการทำ complement

A* คือ Radix Complement ของ A

R คือ Radix ซึ่งจะมีค่าตามมาตรฐานของ A (r มีค่าเป็น 10 สำหรับเลขฐานสิบ และมีค่าเป็น 2 สำหรับเลขฐานสอง)

N คือ จำนวนหลักของ A

สำหรับเลข 1's Complement ของเลขฐานสองหรือเลข 9s Complement ของเลขฐานสิบนั้น จะเรียกได้อีกอย่างว่า Diminished Radix Complement ซึ่งจะมีค่าน้อยกว่า Radix Complement ของเลขฐานนั้นอยู่ 1 ดังนั้นการหาเลข 1's Complement หรือเลข 9's Complement จึงคำนวณได้จากสมการ

$$B + B^* = r^N - 1$$

หรือ $$B^* = r^N - B - 1$$

โดยที่ B คือ เลขที่เราต้องการทำ complement

B* คือ Diminished Radix Complement ของ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

86855

ทำให้เราทราบว่าการหาเลข 1's Complement จะเท่ากับการทำ complement ข้อมูลในทุกบิทของเลขฐานสองนั้น ส่วนเลข 2's Complement ของ 1101 จะมีค่าเท่ากับ 0010 บวกด้วย 1 หรือเท่ากับ 0011

สำหรับรูปแบบการกำหนดค่าบวกหรือลบให้กับเลขฐานสองที่จะกล่าวถึงในที่นี้นั้นจะใช้จุดของเลขฐานสองหรือ Binary Point เข้าช่วย โดยค่าตัวเลขจะถูกจัดให้อยู่ทางด้านขวาของจุด คือ เริ่มจากบิทMSB ซึ่งจะถูกจัดให้อยู่ด้านซ้ายสุด (ติดกับจุดฐานสอง) และบิทที่มีนัยสำคัญต่ำลงมา จะถูกจัดไล่ลงไปทางด้านขวาจนกระทั่งถึงบิท LSB จะถูกจัดให้เป็นบิททางด้านขวาสุด ดังนั้นค่าตัวเลขนี้จะมีค่าได้ไม่เกิน 1 คือจะมีค่าตั้งแต่ :

.0000....0

จนถึง .1111....1

สำหรับทางด้านซ้ายของจุดจะมีเพียงบิทเดียวเท่านั้น โดยบิทดังกล่าวนี้คือ **บิทเครื่องหมาย** หรือ SIGN BIT ซึ่งใช้สำหรับการกำหนดค่าบวกหรือลบของเลขฐานสอง โดยบิทเครื่องหมายจะมีข้อมูลเป็น "0" สำหรับเลขฐานสองที่มีค่าบวก และจะมีข้อมูลเป็น "1" สำหรับเลขฐานสองที่มีค่าเป็นลบ

การใช้จุดฐานสองในการกำหนดค่าบวกและลบของเลขฐานสองใดๆนั้น จะมีวิธีการในการกำหนดค่าที่เป็นลบของเลขฐานสองที่แตกต่างกัน 3 แบบ (ในการกำหนดค่าที่เป็นบวกนั้นทั้ง 3 แบบจะมีวิธีการกำหนดที่เหมือนกัน) คือ:

4.1 แบบที่แสดงค่าและเครื่องหมาย (SIGN AND MAGNITUDE):

สำหรับการกำหนดค่าโดยวิธีนี้ เลขที่มีค่าเป็นบวกและลบ จะถูกกำหนดโดยใช้ข้อมูลในบิทเครื่องหมายเพียงอย่างเดียวเท่านั้น ส่วนค่าของเลขที่อยู่ด้านขวาของจุดจะเหมือนกัน

4.2 แบบ 1'S Complement :

ในการกำหนดโดยวิธีนี้ เลขที่มีค่าเป็นลบจะเป็นเลข 1's Complement ของเลขบวกที่มีค่าเท่ากัน

4.3 แบบ 2's Complement :

ในการกำหนดโดยวิธีนี้ เลขที่มีค่าเป็นลบจะเป็นเลข 2s Complement ของเลขบวกที่มีค่าเท่ากัน

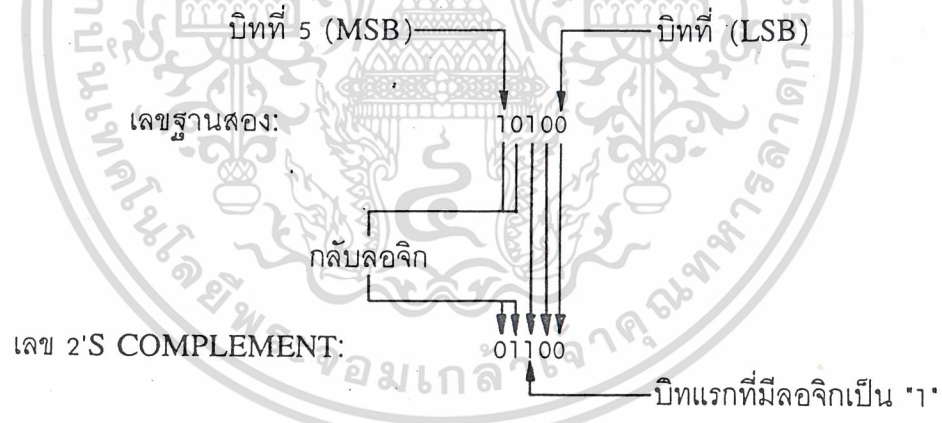
จากหลักการหาเลข 2's complement ที่ได้กล่าวถึงในขั้นต้นนั้น เราอาจสรุปวิธีการที่ง่ายต่อการหาได้ เป็น 2 วิธี ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โดยการหาเลข 1's complement แล้วบวกด้วย 1 ที่บิท LSB เช่น

		20	=	10100
	1's complement ของ	20	=	01011
ดังนั้น	2's complement ของ	20	=	01011
				+00001
หรือ			=	01100

2. พิจารณาจากข้อมูลในบิทต่างๆของเลขที่ต้องการหา 2's complement โดยเริ่มจากบิท LSB (บิทขวาสุด) เข้าไปหาบิท MSB (บิทซ้ายสุด) คือถ้าหากข้อมูลในบิท LSB เป็น "0" ให้ใส่ค่า "0" ไว้ในบิทของเลข 2's complement ที่ตรงกับบิทที่เราพิจารณาอยู่จากนั้นจึงพิจารณาบิทที่อยู่ถัดไปทางด้านซ้ายไปเรื่อยๆจนพบบิทที่มีข้อมูลเป็น"1" ให้ใส่ค่า "1" ไว้ในบิทของเลข 2's complement ที่ตรงกับบิทที่พิจารณาอยู่จากนั้นให้กลับค่าของข้อมูล (จาก "0" เป็น "1" และจาก "1" เป็น "0") ในทุกบิทที่อยู่ถัดไปทางซ้ายจนถึงบิท MSB เช่น การหาเลข 2's complement ของเลขฐานสอง 10100 :



ในบิทที่ 1 และ 2 (นับจากขวาสุด) มีข้อมูลเป็น "0" ข้อมูลในบิทที่ 1 และ 2 ของเลข 2's complement จึงมีค่าเป็น "0" สำหรับในบิทที่ 3 มีข้อมูลเป็น "1" ข้อมูลในบิทที่ 3 ของเลข 2's complement จึงมีค่าเป็น "1" ด้วยจาดนั้นข้อมูลในบิทที่อยู่ถัดมาคือ บิทที่ 4 และ 5 (ข้อมูลคือ "0" และ "1" ตามลำดับ) จะถูกทำคอมพลีเมนท์ก่อนที่จะถูกนำไปใช้เป็นข้อมูลในบิทที่ 4 และ 5 ของเลข 2's complement ทำให้ข้อมูลในบิทที่ 4 และ 5 ของเลข 2's complement มีค่าเป็น "1" และ "0" ตามลำดับ ดังนั้น เลข 2's complement ของเลข 10100 จึงเป็น 01100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การบวก/ลบเลขฐานสองที่กำหนดโดยใช้ค่าและเครื่องหมาย

ในการบวกเลขที่มีการกำหนดตามรูปแบบทั้ง 3 ดังที่ได้กล่าวไว้ข้างต้น จะมีวิธีการบวกตัวเลขที่มีค่าเป็นบวกทั้งตัวตั้งและตัวบวกเหมือนกันทั้ง 3 แบบ แต่สิ่งที่ต้องคำนึงถึงในการบวกก็คือผลบวกที่ได้จะต้องมีค่าไม่เกิน 1 ไม่เช่นนั้นผลบวกที่จะได้จะต้องมีค่าไม่เกิน 1 ไม่เช่นนั้นผลบวกที่ได้จะมีจำนวนบิทเกินกว่าจำนวนบิทของ register ที่ใช้สำหรับเก็บผลบวก (คือต้องมีค่าไม่เกิน .1111....1) สำหรับตัวอย่างเลขฐานสองที่ใช้ประกอบการอธิบายถึงวิธีการบวก/ลบเลขฐานสองทั้ง 3 แบบนี้จะใช้เลขฐานสองขนาด 5 บิท (2^{-5}) เหมือนกันทั้ง 3 แบบ

ในการบวกลบเลขฐานสองแบบที่มีการใช้ค่าและเครื่องหมายนั้นเราสามารถแบ่งออกได้เป็น 2 กรณีคือ

4.1.1 กรณีตัวตั้งและตัวบวกมีเครื่องหมายเดียวกัน :

คือ กรณีที่ทั้งตัวตั้งและตัวบวกมีเครื่องหมายเป็นบวกหรือลบเหมือนกันทั้งคู่ ในกรณีเช่นนี้เราสามารถทำการบวกเลขทั้งสองจำนวนได้ทันที โดยข้อมูลในบิทเครื่องหมายของผลลัพธ์จะเหมือนกับข้อมูลในบิทเครื่องหมายของตัวตั้งและตัวบวก (คือ การบวกในที่นี้จะบวกเฉพาะค่าของตัวเลขที่อยู่ทางด้านขวาของจุดเท่านั้น)

	(+20)	:	0.10100		(-20)	:	1.10100
+	(+ 9)	:	+0.01001		(- 9)	:	+1.01001
	+24	:	0.11101		-29	:	1.11101

4.1.2 กรณีที่ตัวตั้งและตัวบวกมีเครื่องหมายต่างกัน :

คือ กรณีของการบวกเลขที่มีค่าเป็นบวกกับเลขที่มีค่าเป็นลบ ซึ่งในกรณีนี้เราต้องทำ complement ค่าของตัวเลขที่เป็นบวก (ทำ complement หรือกลับลอจิกเฉพาะค่าของข้อมูลที่อยู่ทางด้านขวาของจุดนั้น) ก่อนที่จะทำการบวก โดยการบวกจะทำเฉพาะค่าของข้อมูลที่อยู่ทางด้านขวาของจุดเท่านั้น สำหรับการบวกเลขในกรณีที่ตัวตั้งและตัวบวกมีเครื่องหมายต่างกันนี้ จะแบ่งออกได้อีก 2 กรณี คือ :

2.1 กรณีที่ตัวตั้งมีค่ามากกว่าตัวบวก: คือ ค่าตัวเลข (ในที่นี้เมื่อกล่าวถึง " ค่าตัวเลข " จะหมายถึงเฉพาะค่าที่อยู่ทางด้านขวาของจุดเท่านั้น โดยค่าที่อยู่ทางด้านซ้ายของจุดจะเรียกเป็น"เครื่องหมาย') ของตัวตั้งมีค่ามากกว่าการบวก ซึ่งในกรณีนี้หลังจากการนำค่าตัวเลขของตัวบวกมาทำ complement และนำมาบอกกับค่าตัวเลขของตัวตั้งแล้วมักจะเกิดตัวทดที่เรียกว่า END-AROUND CARRY (EAC) ขึ้น ซึ่งตัวทดนี้จะต้องถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำกลับมาบวกกับผลบวกที่ได้อีกครั้งจึงจะได้ค่าตัวเลขที่เป็นผลลัพธ์ถูกต้อง ส่วนข้อมูลในบิทเครื่องหมายนั้น จะได้จากข้อมูลในบิทเครื่องหมายของตัวตั้งเช่น

(+20)	:	0.10100		(-20)	:	1.10100		ตัวตั้ง
+ (- 9)	:	+ 1.01001		+ (+ 9)	:	+ 0.01001		ตัวบวก
	:	10100			:	10100		ค่าของตัวตั้ง
	:	10110			:	10110		คอมพลีเมนท์ของตัวบวก
	:	01010			:	01010		ผลบวกเกิดตัวทด EAC
	:	1			:	1		นำ EAC บวกกับผลบวก
	:	01011			:	01011		
+11	:	0.01011		-11	:	1.01011		ผลลัพธ์ (บิทเครื่องหมาย-ได้จากตัวตั้ง)

2.2 กรณีที่ตัวตั้งมีค่าน้อยกว่าหรือเท่ากับตัวบวก : คือ ค่าตัวเลขของตัวตั้งมีค่าน้อยกว่าตัวบวก ซึ่งในกรณีนี้หลังจากการนำค่าตัวเลขของตัวบวกมาทำ complement และบวกเข้ากับค่าตัวเลขของตัวตั้งแล้วมักจะไม่เกิดตัวทด EAC ขึ้นสำหรับผลบวกที่ได้จะเป็นค่า complement ซึ่งเราจำเป็นต้องนำผลบวกที่ได้นี้มาทำ complement ก่อนจึงได้ผลลัพธ์ที่ถูกต้อง ส่วนข้อมูลในบิทเครื่องหมายของผลลัพธ์นี้จะได้มาจากบิทเครื่องหมายของตัวบวก เช่น:

(+ 9)	:	0.01001		(- 9)	:	1.01001		ตัวตั้ง
+ (-20)	:	+ 1.10100		+ (+20)	:	+ 0.10100		ตัวบวก
	:	01001			:	01001		ค่าของตัวตั้ง
	:	01011			:	01011		คอมพลีเมนท์ของตัวบวก
	:	10100			:	10100		ผลบวกไม่เกิดตัวทด EAC
	:	01011			:	01011		ทำคอมพลีเมนท์ผลบวก
-11	:	1.01011		+11	:	0.01011		ผลลัพธ์ (บิทเครื่องหมาย-ได้จากตัวบวก)

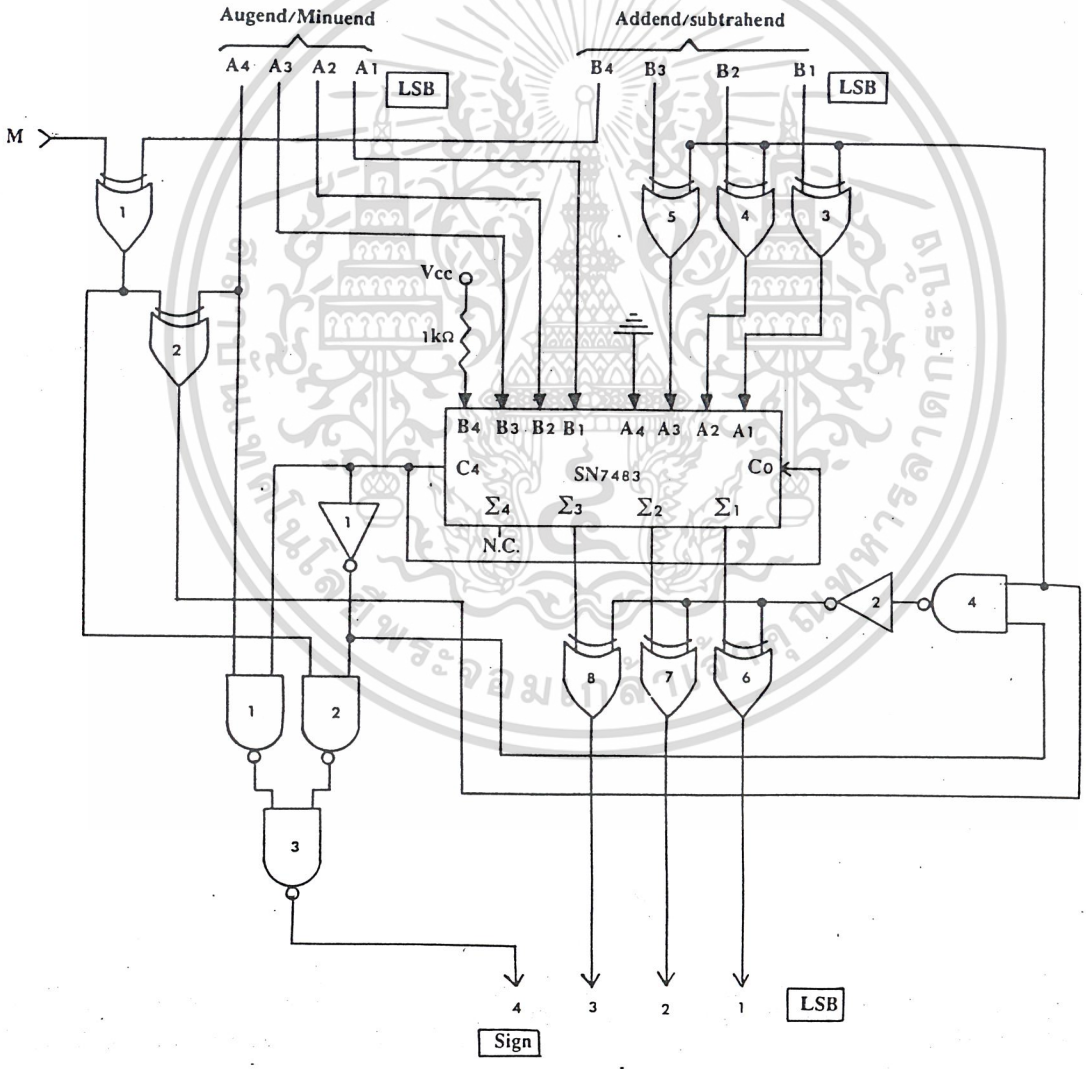
สำหรับการลบเลขที่มีการกำหนดโดยใช้ค่าและเครื่องหมายนี้ สามารถทำได้โดยการเปลี่ยนเครื่องหมายของตัวลบก่อน คือ เปลี่ยนข้อมูลในบิทเครื่องหมายของตัวลบจาก"0" เป็น "1" หรือจาก "1" เป็น "0" จากนั้นจึงทำการบวกตัวลบที่ถูกเปลี่ยนเครื่องหมายแล้วนี้ เข้ากับตัวตั้งตามวิธีการที่กล่าวถึงข้างต้น [A-(+B) = A + (-B) และ A – (-B) = A + (+B)] เช่น :

(+20) : 0.10100 - (- 9) : - 0.01001	จะมีค่าเท่ากับ	(+20) : 0.10100 + (+ 9) : + 0.01001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวอย่างวงจรบวก/ลบที่ใช้สำหรับเลขฐานสองที่กำหนดโดยใช้ค่าและเครื่องหมายนี้จะแสดงในรูปที่ 4.1

วงจรในรูปที่ 4.1 เป็นตัวอย่างวงจรที่ทำหน้าที่ในการบวกลบ เลขฐานสองที่มีขนาด 3 บิท (บิทที่ 4 คือ A_4 และ B_4 เป็นบิทเครื่องหมาย) โดยมีรีจิสเตอร์A (บิท A_1-A_3) เป็นตัวตั้งและรีจิสเตอร์ B (B_1-B_3) เป็นตัวบวก/ลบ ซึ่งการบวกหรือลบของวงจรจะถูก ควบคุมด้วย ลอจิกที่input M โดยวงจรจะทำหน้าที่เป็นวงจรบวก เมื่อ input M ได้รับลอจิกเป็น "0" และทำหน้าที่เป็นวงจรลบ เมื่อ input M ได้รับลอจิกเป็น "1" (เราใช้วงจรบวกทำงานทั้งการบวกและลบทั้งนี้เนื่องจากการลบก็คือการบวกตัวตั้งด้วยค่าลบของตัวบวก เช่น +3 ลบด้วย +2 จะมีค่าเท่ากับ +3 บวกด้วย –(+2) เป็นต้น) สำหรับวงจรบวกลบที่แสดงในรูปที่ 1 สามารถแบ่งออกได้ 3 ส่วนคือ :



รูปที่ 4.1 ตัวอย่างวงจรบวก / ลบ ที่ใช้สำหรับเลขฐานสองที่กำหนดโดยใช้ค่าและเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **ส่วนที่เป็นบิทเครื่องหมาย :**

ประกอบด้วยวงจร NAND ที่ 1-3,วงจร XOR ที่ 1 และวงจร INVERTER ที่ 1 โดยวงจรในส่วนนี้

อินพุทเป็นบิท A_4,B_4 และ M รวมถึงตัวทด EAC คือ C_4 ด้วยซึ่ง output ของวงจรจะเป็นบิทเครื่องหมาย (output 4 หรือSIGN) ตามตารางที่ 4.1 :

ตารางที่ 4.1 ตารางความจริงสำหรับ output ที่เป็นบิทเครื่องหมาย (เอาท์พุท "4" หรือ "Sign")

อินพุท		เอาท์พุท 4 (SIGN)	
A_4	B_4	M = 0 (บวก)	M = 1 (ลบ)
0	0	0	$\overline{C}_4$
0	1	$\overline{C}_4$	0
1	0	C_4	1
1	1	1	C_4

จากตารางความจริงข้างต้น ข้อมูลในบิทเครื่องหมายของผลลัพธ์ (บิท 4 หรือSIGN) จะแบ่งออกได้เป็น 4 กรณี

ข้อมูลในบิทเครื่องหมายเป็น "0": กรณีนี้ผลลัพธ์ที่ได้จะเป็นบวก ซึ่งเกิดขึ้นในการบวก (input M ได้รับลอจิกเป็น "0" คอลัมน์ที่ 3 จากซ้ายในตารางที่ 1) ตัวตั้งที่มีค่าเป็นบวก ($A_4 = 0$) กับตัวบวกที่มีค่าเป็นบวก
(B_4=0) หรือการลบตัวตั้ง ที่มีค่าเป็นบวกกับตัวลบที่มีค่าเป็นลบ(M=1,A_4=0,B_4=1)

ข้อมูลในบิทเครื่องหมายเป็น "1": กรณีนี้ผลลัพธ์ที่ได้จะมีค่าเป็นลบ ซึ่งเกิดขึ้นในกรณีของการบวกตัวตั้งที่เป็นลบกับตัวบวกที่เป็นลบ (M=0,A_4=1,B_4=1) หรือในการลบตัวตั้งที่เป็นลบด้วยตัวลบที่เป็นบวก (M=1,A_4=1,B_4=0)

ข้อมูลในบิทเครื่องหมายเป็นค่าคอมพลีเมนท์ของตัวทด EAC ($\overline{C}_4$): กรณีนี้จะเกิดขึ้นเมื่อทำการบวกตัวตั้งที่มีค่าเป็นบวกด้วยตัวบวกที่มีค่าเป็บลบ (M=0,A_4=0,B_4=1) หรือในการลบตัวตั้งที่เป็นบวกด้วยตัวลบที่เป็นบวก (M=1,A_4=0,B_4=0) ทั้งนี้เมื่อพิจารณาจากกรณีที่ 2.1 และ

2.2 ของการบวกลบเลขฐานสองที่มีเครื่องหมายต่างกันแล้ว จะเห็นได้ว่าเมื่อทำการบวกตัวตั้งที่มีค่าเป็นบวกกับตัวบวกที่มีค่าเป็นลบแล้วมีตัวทด EAC เกิดขึ้น ข้อมูลในบิทเครื่องหมายของผลลัพธ์จะมีค่าเป็น "0" แต่ถ้าไม่มีตัวทด EAC เกิดขึ้น ข้อมูลในบิทเครื่องหมายของผลลัพธ์จะมีค่าเป็น"1" ในกรณีนี้จึงทำให้ข้อมูลในบิทเครื่องหมายของผลลัพธ์เป็นค่าคอมพลีเมนท์ของตัวทดEACคือ $\overline{C}_4$

ข้อมูลในบิทเครื่องหมายมีค่าตามตัวทด EAC (C_4) : กรณีนี้จะเกิดขึ้นในการบวกตัวตั้งที่เป็นลบกับตัวบวกที่เป็นบวก ($M=0, A_4=1, B_4=1$) ซึ่งเมื่อพิจารณาจากกรณีที่ 2.1 และ 2.2 จะเห็นได้ว่าข้อมูลในบิทเครื่องหมายของผลลัพธ์จะมีค่าตามตัวทดEAC ที่เกิดขึ้นจากการบวก

2. **ส่วนของวงจรที่ทำคอมพลีเมนท์** :

ในส่วนนี้จะประกอบด้วยวงจรคอมพลีเมนท์ทางด้านอินพุท ซึ่งแยกออกพิจารณาได้ดังนี้ :

วงจรคอมพลีเมนท์ทางด้านอินพุท : ในส่วนนี้จะประกอบด้วยวงจร XOR ที่ 3,4,5 และควบคุมการทำคอมพลีเมนท์โดยวงจร XOR ที่ 1 และ 2 สำหรับหน้าที่ของวงจรในส่วนนี้คือ ทำคอมพลีเมนท์ข้อมูลของตัวบวก/ลบ (ข้อมูลของรีจิสเตอร์B บิท B_1-B_4) เมื่อเป็นการบวกที่ตัวตั้งและตัวบวกมีเครื่องหมายต่างกันคือ(-A) + (-B) หรือการลบที่ตัวตั้งและตัวลบที่มีเครื่องหมายเหมือนกันคือ (+A) – (+B) และ (-A) – (-B) ซึ่งวงจรวนส่วนนี้จะให้เอาท์พุทเป็นค่าคอมพลีเมนท์ของข้อมูลของตัวบวก/ลบ เมื่อเอาท์พุทของวงจร XOR ที่ 2 เป็นลอจิก "1" และจะให้เอาท์พุทตรงตามข้อมูลของตัวบวก/ลบ เมื่อเอาท์พุทของวงจร XOR ที่ 2 เป็นลอจิก "0" สำหรับตารางแสดงสถานะทางเอาท์พุทของวงจร XOR ที่ 2 เมื่อ ได้รับลอจิกที่อินพุท M, A4, และ B ต่าง ๆ กันนั้น จะเป็นดังตารางที่ 4.2 :

ตารางที่ 4.2 ตารางความจริงสำหรับเอาท์พุทของวงจร XOR ที่ 2

อินพุท		เอาท์พุท วงจร XOR ที่ 2	
A4	B4	M = 0 (บวก)	M = 1 (ลบ)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุท = 1 : วงจรจะทำคอมพลีเมนท์ที่อินพุท B1-B4

เอาท์พุท = 0 : วงจรจะไม่ทำคอมพลีเมนท์ทีอินพุท B1-B4

วงจรคอมพลีเนท์ทางด้านเอาท์พุท วงจรในส่วนนี้จะทำคอมพลีเนท์เอาท์พุทจากวงจรบวก เบอร์ SN7483 เมื่อเอาท์พุทของวงจร XOR ที่ 2 เป็นลอจิก "1" (คือในการบวกที่ตัวตั้งและตัวบวกมีเครื่องหมายต่างกัน หรือในการลบที่ตัวตั้งและตัวลบมีเครื่องหมายเหมือนกัน) และไม่มีตัวทด EAC เกิดขึ้นจากการบวก (หรือลบ) ทั้งนี้เป็นกรณีที่ 2.2 ของการบวดเลขที่มีเครื่องหมายต่างกันที่ได้กล่าวถึงในตอนต้น สำหรับส่วนของวงจรคอมพลีเมนท์นี้ จะประกอบด้วยวงจร XOR ที่ 6, 7, 8 และควบคุมการทำคอมพลีเมนท์ด้วยวงจร XOR ที่ 1, 2, วงจรเกท HAND ที่ 4 และวงจร INVERTER ที่ 2 ซึ่งวงจรจะทำการคอมพลีเมนท์เมื่อเอาท์พุทของวงจร INVERTER เป็นลอจิก "1" สำหรับเอาท์พุทของวงจร INVERTER ที่ 2 เทียบกับเอาท์พุทของวงจร XOR ที่ 2 และสถานะของ C4 จะแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ตารางความจริงสำหรับเอาท์พุทของวงจร INVERTER ที่ 2

วงจร XOR ที่ 2	C_4	เอาท์พุทของวงจร INVERTER ที่ 2
0	0	0
0	1	0
1	0	1
1	1	0

เอาท์พุท = 1: วงจรจะทำคอมพลีเมนท์เอาพุท $\sum_1 - \sum_3$

เอาท์พุท = 0: วงจรจะไม่ทำคอมพลีเมนท์ที่เอาท์พุท $\sum_1 - \sum_3$

นอกจากส่วนต่าง ๆ ของวงจรที่ได้กล่าวถึงแล้ว ยังมีอีกจุดหนึ่งที่น่าสนใจคือ ขาอินพุท A4 และ B4 ขวงวงจรเบอร์ SN7438 จะถูกต่อเข้าลอจิก "1" และ "0" ตามลำดับ ทั้งนี้เมื่อเกิดตัวทด EAC ขึ้นจากการบวกของบิท A1 –A3 กับ B1 – B3 ตัวทด EAC นี้จะถูกบวกกับบิท A4 และ B4 ภายในวงจรเบอร์ SN7438 ทำให้เกิดตัวทดขึ้นที่ C4 ได้ ถึงแม้ว่าเราจะทำการบวกเพียง 3 บิท ก็ตาม (เมื่อตัวทด EAC เป็น 1 ถูกนำไปบวกกับข้อมูลในบิท A4 และ B4 ภายมในวงจรเบอร์ SN7483 จะได้เป็น $1+1+0 = 10_2$ ซึ่งจะทำให้ข้อมูลเอาท์พุทที่ บิท $\sum_4$ เป็น "0" และมีตัวทดเกิดขึ้นที่เอาท์พุท C4 ด้วย และเมื่อตัวทด EAC เป็น 0 ถูกนำไปบวกกัยข้อมูลในบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A4 และ B4 ภายในวงจรเบอร์ SN7483 จะได้เป็น 0+1+0 = 01_3 ซึ่งทำให้ข้อมูลเอาท์พุทที่บิท Σ_4 เป็น "1" และไม่มีตัวทดเกิดขึ้นที่เอาท์พุท C4)

4.2 การบวกลบเลขฐานสองแบบ 1 's COMPLEMENT

สำหรับการบวกเลขแบบ 1 's complement นี้จะมีวิธีการที่เหมือนกันไม่ว่าจะเป็นการบวก ที่ทั้งตัวตั้งและตัวบวกมีเครื่องหมายเหมือนกันหรือ ไม่ก็ตาม โดยที่เลขที่มีค่าเป็นลบจะต้องอยู่ในรูปของ 1' complement เสมอ ไม่ว่าจะเป็นตัวตั้งหรือตัวบวกก็ตาม ส่วนเลขที่มีค่าเป็นบวกจะยังคงอยู่ในรูปเดิมไม่ว่าจะเป็นตัวตั้งหรือตัวบวก สำหรับการบวกจะทำการบวกข้อมูลในบิทเครื่องหมายร่วมด้วย แกละเมื่อเกิดตัวทด EAC ขึ้นจากการบวกของเครื่องหมายจะต้องนำตัวทด EAC นั้นกลับมาบวกกับข้อมูลที่บิท LSB ของผลบวกอีกครั้ง จึงจะได้ผลลัพธ์จากการบวกที่ถูกต้อง ส่วนในกรณีที่ไม่มีตัวทด EAC เกิดขึ้นนั้น ผลบวกที่ได้จะเป็นผลลัพธ์ที่ถูกต้อง (ในกรณีของเลขฐาน สองที่กำหนดค่าและเครื่องหมาย จะต้องทำคอมพลีเมนท์ก่อน) สามารถนำมาใช้เป็นคำตอบได้ทันที สำหรับในการลบนั้นเราจะต้องทำคอมพลีเมนท์เลขที่เป็น ตัวลบก่อน แล้วจึงนำมาบวกกับตัวตั้ง (เนื่องจากมักจะนิยมกล่าวถึงในการลบ ในรูปแบบของการบวกมากกว่า และวงจรลบโดยทั่วไป จะถูกสร้างในรูปแบบของวงจรบวก) เช่น (+A) + (-B) หรือ เท่ากับ (A) + ($\overline{B}$) โดยในที่นี้ -B จะมีค่าเท่ากับ $\overline{B}$ เป็นต้น สำหรับตัวอย่างของการบวกเลข 1's complement ในลักษณะต่าง ๆ จะแสดงได้ดังนี้, ในกรณีของการลบก็เพียงแต่ทำ complement เลขที่เป็นตัวลบแล้วนำมาบวกกับตัวตั้งตามปกติ

ในกรณีของเลข 1's complement : +20 คือ 0.10100 | -20 คือ 1.01011
+9 คือ 0.01001 | -9 คือ 1.10110

1. เลขบวก 2 จำนวน

+20	0.10100	ตัวตั้ง
+ 9	0.01001	ตัวบวก
+29	0.11101	ผลลัพธ์

2. เลขลบ 2 จำนวน

-20	1.01011	ตัวตั้ง
- 9	1.10110	ตัวบวก
	1.00001	ผลบวก
	1	เกิด EAC
-29	1.00010	ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลขบวกกับเลขลบที่ให้ผลเป็นบวก

+20	0.10100	ตัวตั้ง
− 9	1.10110	ตัวบวก
	0.01010	ผลบวก
	1	เกิด EAC
+11	0.01011	ผลลัพธ์

− 9	1.10110	ตัวตั้ง
+20	0.10100	ตัวบวก
	0.01010	ผลบวก
	1	เกิด EAC
+11	0.01011	ผลลัพธ์

4. เลขบวกกับเลขลบที่ให้ผลเป็นลบ

−20	1.01011	ตัวตั้ง
+ 9	0.01001	ตัวบวก
−11	1.10100	ผลลัพธ์

+ 9	0.01001	ตัวตั้ง
−20	1.01011	ตัวบวก
−11	1.10100	ผลลัพธ์

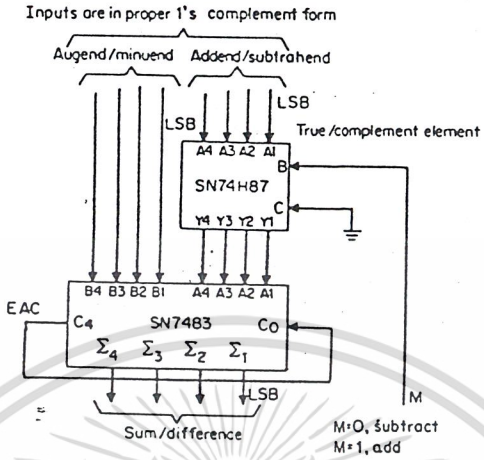
สำหรับตัวอย่างของการลบเลข 1's complement เช่น +20 - +9 จะทำได้โดย

1. ทำ complement ค่าของตัวลบ คือ +9 ได้เป็น −9 หรือ = 1.10110_2
2. ทำการบวกตามปกติ:

+20	0.10100	ตัวตั้ง
− 9	1.10110	ตัวบวก (ค่าคอมพลีเมนท์ของตัวลบ)
	0.01010	ผลบวก
	1	เกิด EAC
+11	0.01011	ผลลัพธ์

สำหรับตัวอย่างวงจรบวก / ลบ เลขแบบ 1 's complement ขนาด 3 บิท (เมื่อรวมบิทเครื่องหมายจะเป็น 4 บิท) จะแสดงในรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



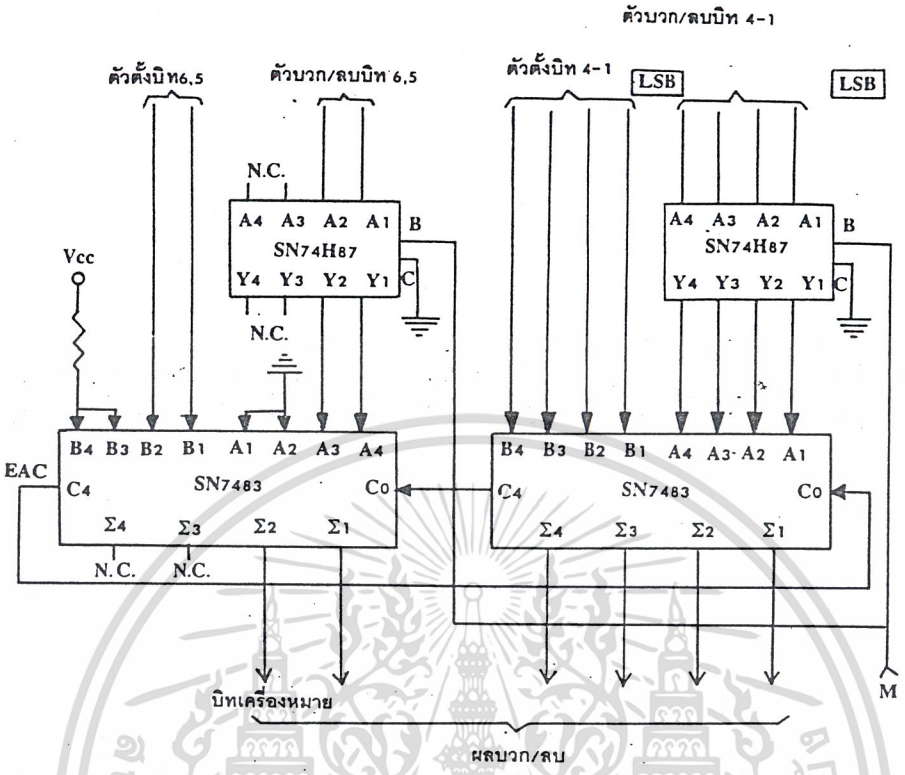
รูปที่ 4.2 แสดงไดอะแกรมของวงจรบวก / ลบ เลขฐาน 2 แบบ 1's complement ขนาด 3 บิท

จากวงจรในรูปที่ 4.2 ข้อมูลอินพุทที่ในส่วนที่เป็นตัวตั้ง คือ บิท B1 – B4 (บิท B4 เป็นบิทของเครื่องหมาย) และตัวบวก / ลบ คือ บิท A1 – A4 (บิท A4 เป็นบิทที่เป็นบิทเครื่องหมาย) จะต้องเป็นเลขฐานสองที่เป็นรูปแบบ 1 's complement คือ ข้อมูลที่มีค่าเป็นลบจะต้องเป็นค่า complement ของข้อมูลที่เป็นบวก เช่น อินพุทที่มีค่าเป็นบวก 3 จะเป็น 0.011

ในขณะที่อินพุท –3 จะมีค่าเป็น 1.100 เป็นต้น โดยในส่วนของวงจรจะประกอบด้วยวงจรบวกขนาด 4 บิท เบอร์ SN7483 และวงจร TRUE/COMPLEMENT ขนาด 4 บิท เบอร์ SN74H87 ซึ่งในส่วนของวงจรเบอร์ SN74H87 นั้นจะให้เอาท์พุทที่เป็นค่า complement ของข้อแลที่เป็นตัวลบในระหว่างการลบ (อินพุท M = 0) และจะให้เอาท์พุทตามค่าของข้อมูลที่เป็นลบ (ไม่ทำ complement) ในระหว่างการบวก (M = 1)

สำหรับในส่วนของในวงจรบวกเบอร์ SN7483 นั้นจะรับข้อมูลอินพุทของตัวตั้ง (B1-B4 โดยตรง) และรับข้อมูลของอินพุทตัวบวก / ลบ (A1 –A4) ผ่านวงจร TRUE /COMPLEMENT (เบอร์ SN74H87) ตัวทด EAC จะเกิดขึ้นทางเอาท์พุท C4 ซึ่งถูกนำกลับมาป้อนให้กับอินพุทของตัวทด (C0) เพื่อรวมกับผลบวกของรีจิสเตอร์A (บิท A1 - A4) กับ B(บิท B1-B4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 วงจรบวก/ลบ เลขฐานสองแบบ 1's complement ขนาด 5 บิท

ในกรณีที่ต้องการเพิ่มจำนวนบิทของข้อมูลอินพุททั้งตัวตั้งและตัวบวกขึ้นนั้น จะทำได้โดยเพิ่มจำนวนของวงจรบวก (SN7483) และวงจร TRUE/COMPLEMENT (เบอร์ SN74H87) ขึ้นตามจำนวนบิทที่ต้องการซึ่งการเพิ่มวงจร SN7483 และ SN74H87 ขึ้นอย่างละ 1 วงจรจะเพิ่มจำนวนบิทของข้อมูลอินพุทขึ้นได้ 4 บิทโดยจะต้องจัดให้บิทเครื่องหมายเป็นอินพุทของบิทที่มีนัยสำคัญสูงสุด (MSB) และตัวทด EAC จะนำออกจากเอาท์พุท C4 ของวงจรบวกที่มี นับสำคัญสูงสุด (คือ วงจรบวกที่ทำการบวกบิทเครื่องหมาย) กลับมาป้อนให้กับอินพุท C0 ของวงจรบวกที่มีนับสำคัญต่ำสุด (วงจรบวกที่ทำการบวกบิท LSB) ดังตัวอย่างวงจรบวก/ลบเลขฐานสองขนาด 5 บิท (เมื่อรวมบิทเครื่องหมายจะเป็น 6 บิท) ในรูปที่ 4.3 (อินพุทของตัวตั้งและตัวบวก/ลบจะต้องอยู่ในรูปแบบ 1's COMPLEMENT)

จากไดอะแกรมในรูปที่ 4.3 วงจรบวกทางด้านซ้ายคือวงจรที่มีนัยสำคัญสูงสุด ซึ่งเอาพุท C4 คือตัวทด EAC ที่ถูกนำกลับมาป้อนให้กับอินพุท C0 ของวงจรบวกที่มีนัยสำคัญต่ำสุด (วงจรบวกทางด้านขวา) ในขณะที่อินพุท C0 ของวงจรบวกทางด้านซ้าย จะได้จากเอาท์พุท C4 ของวงจรบวกทางด้านขวาสิ่งที่น่าสนใจในวงจรนี้คือ อินพุทของตัวตั้ง (บิท B4 และ B3) และตัวบวก/ลบ (บิท A4 และ A3) ของวงจรบวกที่มีนัยสำคัญสูงสุด ที่ไม่ถูกใช้งานนั้น จะถูกต่อเข้ากับลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และลอจิก "0" ตามลำดับทั้งนี้เพื่อให้ตัวทด EAC ที่เกิดขึ้นจากการบวกบิทเครื่องหมายนั้น ถูกส่งออกมายังเอาท์พุท C4 ได้

4.3 การบวก/ลบเลขฐานสองแบบ 2'S COMPLEMENT

การบวกเลขฐานสองที่กำหนด แบบ 2 's complement จะมีวิธีการเหมือนกันไม่ว่าเครื่องหมายของตัวตั้งและตัวบวกจะเหมือนกันหรือไม่ก็ตาม โดยเลขที่มีค่าเป็นลบจะอยู่ในรูปของ 2s complement และเลขที่มีค่าเป็นบวกจะยังคงอยู่ในรูปเดิมไม่ว่าจะเป็นตัวตั้งหรือตัวบวกก็ตาม สำหรับในการบวกนั้นจะทำการบวกข้อมูลในบิทเครื่องหมายร่วมด้วย โดยตัวทดที่เกิดขึ้นจากการบวกของบิทเครื่องหมายจะถูกตัดทิ้ง (ในกรณีของเลข 1's complement ตัวทดดังกล่าวคือตัวทด EAC ที่ถูกนำกลับมาบวกกับบิท LSB ของผลบวก) สำหรับการลบเลขแบบ 2's complement จะทำได้โดยการหาค่า 1's complement ของตัวลบก่อนจากนั้นจึงนำมาบวกกับตัวตั้ง แล้วบวกด้วย 1 ที่บิท LSB ซึ่งวิธีการเช่นนี้ก็คือ การหาค่า 2s complement ของตัวลบแล้วนำมาบวกกับตัวตั้งนั่นเอง (เลข 2's complement = 1's complement +1) หลังจากการบวก/ลบ เลขแบบ 2's complement แล้วผลบวก/ลบจะเป็นผลลัพธ์ที่ถูกต้องสามารถนำไปใช้งานได้ทันที สำหรับตัวอย่างการบวก/ลบเลขแบบ 2's complement ในลักษณะต่างๆจะแสดงได้ดังนี้:

ในกรณีของเลข 2's complement : +20 คือ 0.10100 | -20 คือ 0.01100

+9 . คือ 0.01001 | -9 คือ 1.10111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เลขบวก 2 จำนวน

+20	0.10100	ตัวตั้ง
+ 9	0.01001	ตัวบวก
+29	0.11101	ผลลัพธ์

2. เลขลบ 2 จำนวน

-20	1.01100	ตัวตั้ง
- 9	1.10111	ตัวบวก
	1 1.00011	ผลบวก
		ตัดทิ้ง
-29	1.00011	ผลลัพธ์

3. เลขบวกกับเลขลบที่ให้ผลเป็นบวก

+20	0.10100	ตัวตั้ง
- 9	1.10111	ตัวบวก
	1 0.01011	ผลบวก
		ตัดทิ้ง
+11	0.01011	ผลลัพธ์

- 9	1.10111	ตัวตั้ง
+20	0.10100	ตัวบวก
	1 0.01011	ผลบวก
		ตัดทิ้ง
+11	0.01011	ผลลัพธ์

4. เลขบวกกับเลขลบที่ให้ผลเป็นลบ

-20	1.01100	ตัวตั้ง
+ 9	0.01001	ตัวบวก
-11	1.10101	ผลลัพธ์

+ 9	0.01001	ตัวตั้ง
-20	1.01100	ตัวบวก
-11	1.10101	ผลลัพธ์

5. การลบเลข

ตัวตั้ง	+20	0.10100
ตัวลบ	(- 9)	1.10111

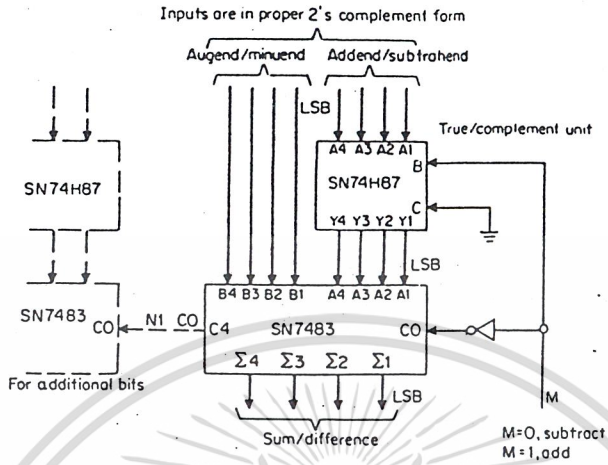
ตัวตั้ง	+20	0.10100
ตัวลบ	(+ 9)	0.01001

	ตัวตั้ง	0.10100
1'S COMPLEMENT		
	ของตัวลบ	0.01000
	ผลบวก	0.11100
	บวกด้วย 1	1
ผลลัพธ์ :	+29	0.11101

	ตัวตั้ง	0.10100
1'S COMPLEMENT		
	ของตัวลบ	1.10110
	ผลบวก	1 0.01010
		ตัดทิ้ง
	บวกด้วย 1	1
ผลลัพธ์ :	+11	0.01011

สำหรับวงจรบวก/ลบเลขฐานสองแบบ 2's complement ขนาด 3 บิท (รวม บิท เครื่องหมายเป็น 4 บิท) จะแสดงในรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



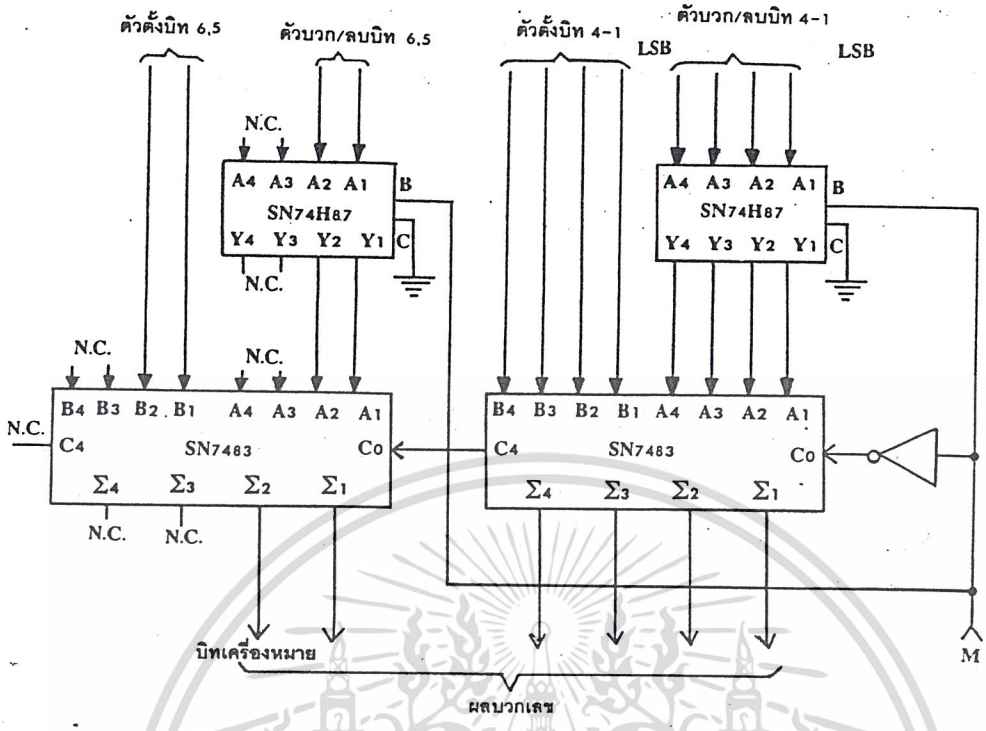
รูปที่ 4.4 วงจรบวก/ลบเลขฐานสองแบบ 2's complement ขนาด 3 บิท

จากวงจรในรูปที่ 4.4 วงจร TRUE/COMPLEMENT (วงจรเบอร์ SN74H87)จะทำคอมพลีเมนท์ข้อมูลที่เป็นตัวบวก/ลบในระหว่างการลบ (อินพุท M=0 : จะเท่ากับการทำ 1's complement ข้อมูลที่เป็นตัวลบ) โดยอินพุทตัวทด C0 จะเป็นค่าคอมพลีเมนท์ของอินพุท M (คือ $\overline{M}$) ดังนั้นในช่วงที่อินพุท M=0 คือในช่วงที่วงจรทำหน้าที่เป็นวงจรลบเลขนั้น อินพุท C0 จะได้รับข้อมูลเป็น "1" ซึ่งจะทำให้ได้ผลลัพธ์ที่เอาท์พุท $\sum_1 - \sum_4$ ของวงจรเบอร์ SN7483 (ในช่วงของการลบ) เป็นผลการบวกข้อมูลของตัวตั้งกับค่า 1's complement ของข้อมูลที่เป็นตัวลบที่ถูกบวกด้วย 1 (คือ เท่ากับการบวกตัวตั้งกับค่า 2's complement ของตัวลบ)

สำหรับในช่วงของการบวกตัวเลขนั้น (อินพุท M=1) อินพุท C0 จะได้รับข้อมูลเป็น "0" โดยที่วงจรเบอร์ SN74H87 จะให้เอาท์พุทตามค่าของข้อมูลที่เป็นตัวบวก/ลบ ที่ได้รับทางอินพุทซึ่งในกรณีนี้เอาท์พุท $\sum_1 - \sum_4$ จะมีค่าเท่ากับผลบวกของข้อมูลที่เป็นตัวตั้งและตัวบวก/ลบ

สำหรับการเพิ่มจำนวนบิทของข้อมูลที่เป็นตัวตั้งและตัวบวก/ลบนั้น จะทำได้ในลักษณะเดียวกับวงจรบวก/ลบเลขแบบ 1's complement เพียงแต่ตัวทดที่ออกจากเอาท์พุท C4 ของวงจรบวก/ลบ นั้นเราสามารถปล่อยลอยไว้ได้ เนื่องจากไม่จำเป็นต้องใช้ตัวทดที่ได้จากบิทเครื่องหมาย (จากเอาท์พุท C4) นั่นเองสำหรับข้อมูลที่เป็นอินพุททั้งตัวตั้งและตัวบวก/ลบที่มีค่าเป็นลบจะต้องอยู่ในรูปของ 2's complement ขนาด 5 บิทและบิทเครื่องหมาย 1 บิท (รวม 6 บิท) ซึ่งใช้วงจรเบอร์ SN7483 และ SN74H87 อย่างละ 2 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 วงจรบวก/ลบเลขฐานสองแบบ 2's complement ขนาด 5 บิท

จากที่ได้กล่าวมาถึงการบวก/ลบเลขฐานสองทั้ง 3 แบบจะเห็นได้ว่าการบวก/ลบเลข 2's complement สามารถทำได้ง่ายและสะดวกที่สุด ในขณะที่การบวก/ลบเลข 1's complement จะมีความซับซ้อนมากขึ้น และแบบที่ใช้ในการกำหนดค่าและเครื่องหมายจะมีความซับซ้อนที่สุด อย่างไรก็ตามในการคูณและหารนั้น เลข 2's complement จะเกิดปัญหาในการคำนวณมากที่สุด อย่างไรก็ตามในการคูณและหารนั้น เลข 2's complement จะเกิดปัญหาในการคำนวณมากที่สุด ในขณะที่การใช้เลขฐานสองที่กำหนดค่าและเครื่องหมายนั้น จะทำได้ง่ายและสะดวกที่สุด ส่วนข้อเสียของเลข 1's complement นั้นจะอยู่ที่ตัวทด EAC (End Around Carry) ซึ่งอาจจะทำให้ค่า delay ของเอาท์พุทมีค่ามากขึ้น เนื่องจาก delay ของตัวทด

การเพิ่มความเร็วในวงจรบวก

วงจรบวกที่ได้กล่าวมาถึงในตอนแรกนั้น เป็นวงจรบวกที่มีการบวกตัวทดแบบอนุกรมซึ่งมีข้อเสียที่ตัวทดที่เกิดขึ้นขากการบวกในแต่ละบิทนั้น จะถูกส่งจากบิท LSB จนถึง MSB ในแบบอนุกรม ทำให้เกิดค่า delay เนื่องจากการส่งผ่านตัวทดภายในวงจร โดยค่า delay ที่เกิดจากการส่งผ่านตัวทดนี้ จะเพิ่มขึ้นตามจำนวนบิทที่เพิ่มขึ้นของเลขที่นำมาบวกกัน เวลาที่วงจรใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณจึงมากขึ้นตามไปด้วย ซึ่งในบางกรณีเวลาที่วงจรใช้ในการคำนวณอาจมากเกินกว่าที่จะยอมรับได้ ดังนั้นจึงจำเป็นต้องใช้วิธีเพิ่มความเร็วในการคำนวณของวงจร ซึ่งโดยทั่วไปก็คือการลดค่า delay ที่เกิดจากการส่งผ่านตัวทดในวงจรนั่นเอง สำหรับการเพิ่มความเร็วในการคำนวณของวงจรนี้ จะมีหลายวิธีเช่น การบวกแบบ CARRY LOOK-AHEAD, แบบอะซิงโครนัส (หรือ Carry Completion) และการบวกแบบ CONDITIONAL SUM เป็นต้น ซึ่งในที่นี้จะได้กล่าวถึงเฉพาะวิธีการบวกแบบ CARRY LOOK-AHEAD เท่านั้น ส่วนวิธีการบวกแบบอื่นสามารถศึกษาได้ จากหนังสือที่เกี่ยวกับการคำนวณภายในระบบคอมพิวเตอร์ทั่วไป

การบวกแบบ CARRY LOOK-AHEAD :

การบวกในแต่ละบิทของวงจรที่ได้กล่าวถึงก่อนหน้านี้นั้น จะต้องรออินพุทตัวทดที่ได้จากผลบวกของบิทที่มีนัยสำคัญต่ำกว่า (บิทที่อยู่ทางขวา) เพื่อนำมาใช้ในการบวกของบิทนั้นด้วย ดังนั้นผลการบวกในบิท LSB จึงเกิดขึ้นเป็นบิทแรก และผลการบวกในบิท MSB จะเกิดขึ้นเป็นบิทสุดท้าย จึงทำให้เวลาที่วงจรใช้ในการคำนวณเพิ่มขึ้นตามจำนวนบิทที่เพิ่มขึ้นของเลขท นำมาบวกกัน

วิธีการบวกแบบ CARRY LOOK-AHEAD นี้จะแก้ไขข้อเสียที่เกิดขึ้นจากการบวกตัวทดแบบอนุกรมโดยการสร้างวงจรคำนวณตัวทดสำหรับการบวกในแต่ละบิทโดยตรง (แทนที่จะรอตัวทดจากผลการบวกในบิทที่มีนัยสำคัญที่ต่ำกว่า) ซึ่งวิธีการนี้จะช่วยให้การบวกในบิท LSB และ MSB เกิดขึ้นในแวลาที่ใกล้เคียงกันและกันเป็นผลให้ค่าดีเลย์ที่เกิดจากการส่งผ่านตัวทดลอง

สำหรับในการศึกษาวิธีการบวกแบบ CARRY LOOK-AHEAD นี้จะกำหนดสัญลักษณ์พิเศษขึ้น 3 ชนิด คือ :

$$^{1}G = AB$$

$$^{0}G = \overline{A}\,\overline{B}$$

$$P = A\overline{B} + \overline{A}B = A \oplus B$$

โดยที่ : A และ B คือ บิทของเลข 2 จำนวนที่ถูกนำมาบวกกันภายในวงจรบวก

^{1}G คือ ตัวทดที่มีค่าเป็น “1” หรือ 1-CARRY (คือกรณีที่เกิดตัวทดขึ้นนั้นเองซึ่งจะเกิดขึ้น (มีลอจิกเป็น “1”) เมื่อข้อมูลในบิท A และ B ที่ถูกนำมาบวกกันนั้นเป็น “1” ทั้งคู่ โดยไม่ขึ้นกับตัวทดทางอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

^0G คือ ตัวทดที่มีค่าเป็น "0" หรือ 0 – CARRY (คือกรณีที่ไม่มีตัวตนทดเกิดขึ้นนั้นเอง) ซึ่งจะเกิดขึ้น (มีลอจิก เป็น "1") เมื่อข้อมูลในบิท A และ B ที่นำมาบวกกันนั้นเป็น "0" ทั้งคู่ โดยไม่ขึ้นกับตัวทดทางอินพุท

P คือ กรณีที่ข้อมูลของตัวทดทางเอาท์พุทมีข้อมูลเหมือนกับข้อมูลของตัวทดทางอินพุทซึ่งจะ เกิดขึ้นเมื่อข้อมูลในบิท A และ B แตกต่างกันคือ เมื่อ A = 0 และ B = 1 หรือ A = และ B = 0

จากสัญลักษณ์ที่ได้แสดงไว้ข้างต้นนั้นเราสามารถกำหนดสมการของตัวทดที่มีค่าเป็น 1 (1 – CARRY) ของการบวกในบิทที่ n ได้เป็น :

$$^1C_n = {}^1G_n + P_n \; {}^1C_{n-1}$$

โดยที่ : ^1C_n คือ เอาท์พุทของกรณีที่ตัวทดมีค่าเป็น 1 (มีค่าเป็น 1 เมื่อมีตัวทดเกิดขึ้น)

$^1C_{n-1}$ คือ ตัวทดที่มีค่าเป็น 1 ที่ได้รับจากการบวกในบิทที่มีนัยสำคัญต่ำกว่า

สำหรับสมการของตัวทดที่มีค่าเป็น 0 (0 – CARRY) ของการบวกในบิทที่ n ได้เป็น:

$$^0C_n = 0G_n + P_n \; {}^0C_{n-1}$$

โดยที่: ^0C_n คือ เอาท์พุทของกรณีที่ตัวทดมีค่าเป็น 0 (มีค่าเป็น 0 เมื่อไม่มีตัวทดเกิดขึ้น)

$^0C_{n-1}$ คือ ตัวทดที่มีค่าเป็น 0 ที่ได้รับจากการบวกในบิทที่มีนัยสำคัญต่ำกว่า

สำหรับกรณีสมการของตัวทดที่มีค่าเป็น 1 หรือ ^1C_n ในบิทที่ KI ของการบวกนั้นเราสมารถกำหนดขึ้นใหม่ได้เป็น :

$$C_k = G_k + P_k \; C_{k-1}$$

ซึ่งสมการข้างต้นนั้นเป็นสมการเดียวกับสมการของ ^1C_n เพียงแต่ได้ตัดเลข 1 ที่นำหน้าอยู่ออกและกำหนดให้สมการในรูปแบบดังกล่าวเป็นสมการของตัวทดที่มีค่าเป็น 1 เมื่อแทนค่า G_k และ p_k ลงในสมการแล้ว จะได้รูปของสมการ C_k เป็น :

$$C_k = A_k B_k + (\overline{A_k} B_k + A_k \overline{B_k}) C_{k-1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และโดยการใช้ตาราง KARNAUGH MAPS สมการข้างต้นสามารถลดรูปลงได้เป็น :

C_{n-1} \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_k = A_kB_k + B_kC_{k-1} + C_{k-1} \text{ หรือ}$$
$$C_k = A_kB_k + C_{k-1}(A_k + B_k)$$

จากสมการ $C_k = G_k + P_k C_{k-1}$ ทำให้เราสามารถแทนเทอม P_k ได้ด้วยฟังก์ชั่น OR ของบิท A_k และ B_k แทนที่จะเป็นฟังก์ชั่น Exclusive-OR ดังที่ได้แสดงไว้ข้างต้น ซึ่งช่วยให้เราสามารถลดจำนวนวงจรเกทที่จำเป็นต้องใช้ในการสร้างวงจรในส่วนนี้ลงได้ สำหรับเอาท์พุทตัวทดในบิทที่ K-1 นั้นจะเป็นไปตามสมการ:

$$C_{k-1} = G_{k-1} + P_{k-1} C_{k-2}$$

ซึ่งถ้าเราทำการแทนสมการของ C_{k-1} ที่ได้นี้ลงในสมการของ C_k จะทำให้สมการดเป็น

$$C_k = G_k + P_k(G_{k-1} + P_{k-1} + C_{k-2})$$
$$= G_k + p_k G_{k-1} + P_k P_{k-1} C_{k-2}$$

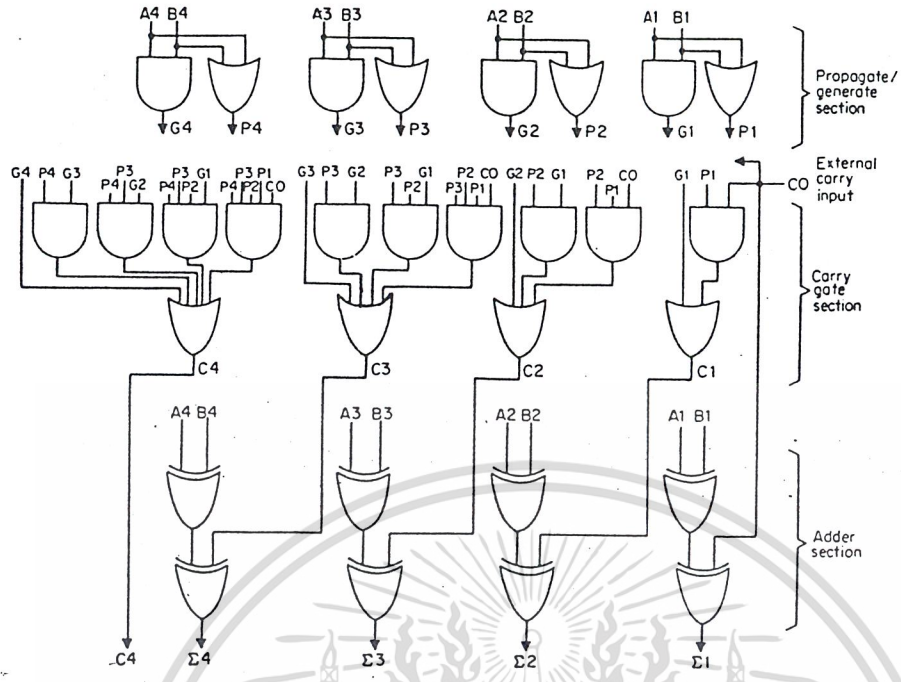
และโดยการแทนสมการของตัวทดในบิทที่มีนัยสำคัญต่ำลงเรื่อย ๆ จะทำให้เราได้สมการของ C_k เป็น:

$$C_k = G_k + P_{k-1} G_{k-1} + P_k P_{k-1} G_{k-2} + \ldots + P_k P_{k-1} P_{k-2} \ldots P_1 O_0$$

โดยที่ : C_0 คือตัวทดที่ได้รับจากภายนอก

สำหรับวงจรบวกแบบ CARRY LOOK-AHEAD ซึ่งใช้วงจรสร้างตัวทดสำหรับการบวกในบิทต่าง ๆ ตามสมการของ C_k ที่ได้นั้น จะเป็นดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



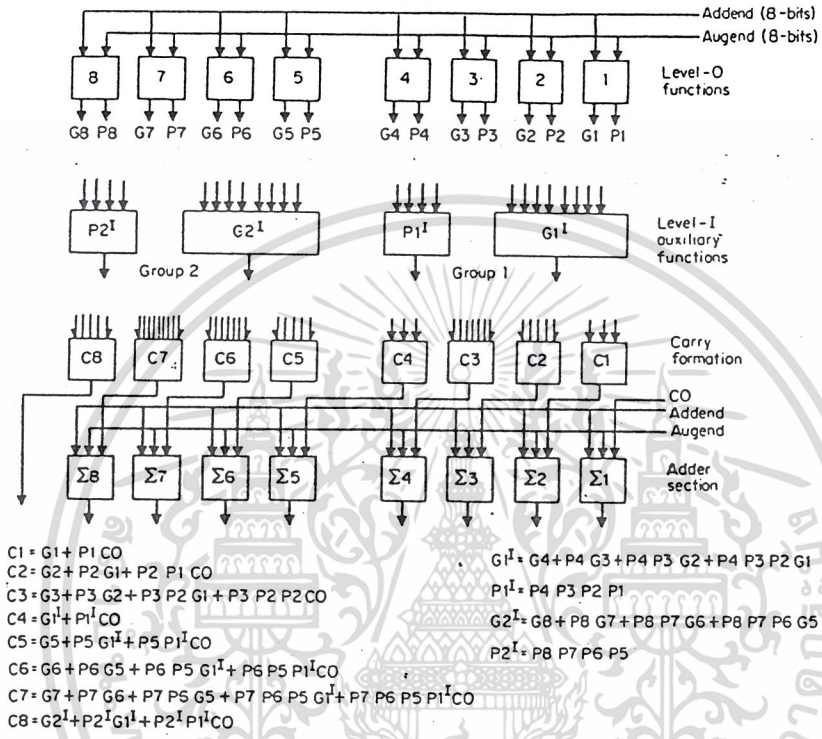
รูปที่ 4.6 วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 4 บิท

จากรูปที่ 4.6 เมื่อเทียบกับวงจรบวกแบบขนานที่มีการบวกตัวทดแบบอนุกรม แล้วจะเห็นได้ว่า วงจรบวกขนาด 4 บิท นั้น ตัวอินพุท C_0 จะต้องผ่านวงจรเกท ถึง 8 ระดับ ก่อนที่จะถูกส่งผ่านออกมาทางเอาท์พุทตัวทด C_4 ในขณะที่วงจรบวกแบบ CARRY LOOK – AHEAD ในรูปที่ 4.6 นั้นตัวทด C_0 จะถูกส่งผ่านวงจรเกทเพียง 3 ระดับเท่านั้น ซึ่งความแตกต่างของจำนวนระดับของวงจรบวกทั้งสองแบบนั้น จะเพิ่มมากขึ้นตามจำนวนบิทที่เพิ่มขึ้นของเลขที่นำมาบวกกัน โดยเมื่อจำนวนบิทของเลขเพิ่มขึ้น 1 บิท จะทำให้จำนวนระดับของวงจรเกท เพิ่มขึ้น 2 ระดับในขณะที่จำนวนระดับของวงจรเกทในรูปที่ 4.6 นั้นจะคงอยู่ที่ 3 ระดับไม่ว่าจำนวนบิทของเลขที่นำมาบวกกันนั้น จะเพิ่มขึ้นหรือไม่ก็ตาม อย่างไรก็ตามวงจรแบบ CARRY LOOK - AHEAD นี้ ยังคงมีข้อเสียตรงที่จำนวนวงจรเกทที่ใช้ในส่วนของวงจรสร้างตัวทดนั้น จะเพิ่มขึ้นเป็นทวีคูณเมื่อมีการเพิ่มจำนวนบิทของเลขที่นำมาบวกกันขึ้นซึ่งนอกจากจะเป็นการสิ้นเปลืองค่าใช้จ่ายในการสร้างวงจรแล้วยังทำให้เกิดความยุ่งยากทางด้านค่า FAN – IN และ FAN – OUT ของวงจรเกทที่ใช้ด้วย

จากข้อเสียของวงจรบวกแบบ CARRY LOOK-AHEAD ที่ได้กล่าวมาแล้ว ทำให้เกิดปัญหาในการใช้งานวงจรบวกแบบนี้กับเลขที่มีจำนวนบิทสูงขึ้น ดังนั้นจึงจำเป็นต้องแก้ไขวิธีการสร้างวงจรบวกแบบ CARRY LOOK – AHEAD นี้ใหม่ โดยใช้วิธีการเรียกว่า "FLA' (FIRST LEVEL AUSILIARY: ในบางกรณีอาจเรียกเป็น 2 –LEVEL LOOK – AHEAD ส่วนวิธีการเดิมที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้กล่าวถึงนั้นก็คือ 0 – LEVEL LOOK – AHEAD) ซึ่งวิธีการแบบ FLA นี้จะอาศัยหลักการลดจำนวนวงจรเกทที่จำเป็นต้องใช้ลงโดยยอมลดความเร็วในการคำนวณลง สำหรับตัวอย่างของหลักการสร้างวงจรบวกเลขฐานสองขนาด 8 บิท CARRY LOOK – AHEAD ที่ใช้วิธี FLA เข้าช่วยนั้น จะแสดงไว้ในรูปที่ 4.7



รูปที่ 4.7 วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 8 บิท ที่ใช้วิธี FLA เข้าช่วย

หลักการของ FLA นั้นจะใช้การแบ่งจำนวนบิทของเลขที่นำมาบวกกันออกเป็นกลุ่มย่อย เพื่อลดความซับซ้อนของวงจรส่วนที่คำนวณตัวทดลง (โดยเฉพาะในบิทที่มีนัยสำคัญสูง ซึ่งจะมีความซับซ้อนมาก) สำหรับหลักการของวงจรในรูปที่ 4.7 นั้น จะแบ่งจำนวนบิทของเลขที่นำมาบวกกันออกเป็น 2 กลุ่มๆ ละ 74 บิท โดยวงจรที่สร้างขึ้นนั้นจะถูกแบ่งออกเป็นระดับต่าง ๆ คือ :

วงจรในระดับที่ 0 :

สำหรับวงจรในระดับที่ 0 ของรูปที่ 4.7 นี้เป็นวงจรที่ใช้ในการสร้า' G_k และ P_k ในรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรในระดับที่ 1 :

วงจรในระดับที่ 1 นี้จะแบ่งออกเป็น 2 กลุ่ม โดยแต่ละกลุ่มจะมีขนาด 4 บิท ซึ่งจะเป็นวงจรที่ใช้หลักการ FLA ในการสร้าง $G_n^{'}$ และ $P_n^{'}$ เพื่อนำไปใช้ในการสร้างตัวทดของวงจรในระดับต่อไป ซึ่งทำให้สมการของตัวทดที่สร้างขึ้นจาก $G_n^{'}$ และ $P_n^{'}$ นี้เป็น:

$$C_k = G_n^{'} + P_n^{'} G_{n-1}^{'} + P_n^{'} P_{n-1}^{'} G_{n-2}^{'} + \ldots + P_n^{'} P_{n-1}^{'} P_{n-2}^{'} \ldots P_1^{'} C_0$$

โดยที่ k คือ ตำแหน่งของบิทที่ต้องการหาตัวทด

n คือ หมายเลขกลุ่มของบิท k

ดังนั้นในกรณีของสมการสำหรับการสร้างตัวทด C_8 ซึ่งอยู่ในกลุ่มที่ 2 โดยใช้หลักการ FLA จะเป็นดังนี้ :

$$C_8 = G_2^{'} + P_2^{'} G_1^{'} + P_2^{'} P_1^{'} C_0$$

โดยที่ K=8 และ n=2

สำหรับการใช้หลักการ FLA นี้จะสามารถเห็นได้ชัดเจนที่สุด ในกรณีของบิทที่มีนัยสำคัญสูงสุดของแต่ละกลุ่ม ซึ่งในที่นี้ก็คือบิทที่ 4 และ 8 นั่นเอง (ขอให้พิจารณาจากสมการของตัวทดในบิทที่ 1 ถึง 8 ในรูปที่ 4.7)

จากสมการของการสร้างตัวทด C_8 แบบ CARRY LOOK-AHEAD :

$$C_8 = G_8 + P_8G_7 + P_8P_7G_6 + P_8P_7P_6G_5 + P_8P_7P_6P_5G_4 + P_8P_7P_6P_5P_4G_3 + P_8P_7P_6P_5P_4P_3G_2 + P_8P_7P_6P_5P_4P_3P_2G_1 + P_8P_7P_6P_5P_4P_3P_2P_1G_0$$

เราสามารถจัดรูปสมการใหม่ได้เป็น :

$$C_8 = (G_8 + P_8G_7 + P_8P_7G_6 + P_8P_7P_6G_5) + (P_8P_7P_6P_5) (G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1) + (P_8P_7P_6P_5) (P_4P_3P_2P_1) (C_0)$$

เมื่อพิจารณาร่วมกับสมการ C_8 ของ FLA แล้วจะได้สมการของ $G_n^{'}$ และ $P_n^{'}$ เป็น :

$$G_n^{'} = G_k + P_kG_{k-1} + \ldots + P_kP_{k-1} \ldots P_{k-g+2}G_{k-g+1}$$

$$P_n^{'} = P_kP_{k-1} \ldots P_{k-g+2}P_{k-g+1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ n คือ หมายเลขกลุ่มของบิท

k คือ ตำแหน่งของบิทที่มีนัยสำคัญสูงสุดของกลุ่ม

g คือ จำนวนบิทในกลุ่ม และ $n = k/g$

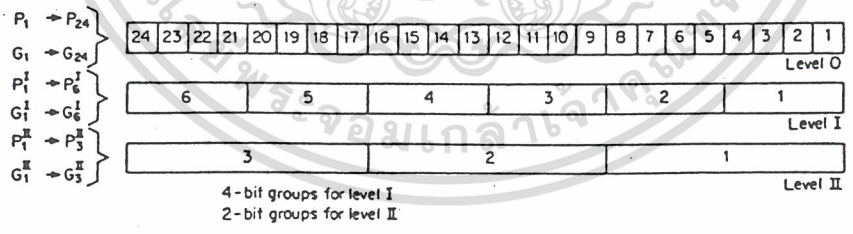
เช่น กรณีของ FLA ในกลุ่มที่ 1 (n = 1) ซึ่งมีจำนวนบิทในกลุ่มเท่ากับ 4 บิท จะมีบิทที่มีนัยสำคัญสูงสุดในกลุ่มคือ บิทที่ 4 (k = n*g = 1*4=4) จะได้สมการ FLA ของ G_1^I เป็นดังนี้ :

$$G_1^I = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1$$

วงจรสร้างตัวทดและวงจรบวก :

ในส่วนของวงจรบวกจะมีลักษณะที่เหมือนกับในวงจรบวกแบบ CARRY LOOK - AHEAD แต่ในส่วนของวงจรสร้างตัวทดนั้น จะมีข้อแตกต่างกันตามสมการของตัวทดที่ใช้ G_n^I และ P_n^I ร่วมกับ G_k และ P_k ดังตัวอย่างสมการของตัวทดแสดงในรูปที่ 4.7

จากการใช้วิธี FLA นั้น ถึงแม้ว่าจะช่วยลดความซับซ้อนของวงจรบวกแบบ CARRY LOOK – AHEAD ลงได้ก็ตาม แต่ในกรณีที่จำนวนบิทของเลขที่นำมาบวกกันเพิ่มมากขึ้น ก็จะทำให้เกิดความซับซ้อนขึ้นอีก แต่เรายังคงสามารถแก้ไขปัญหาในลักษณะนี้ได้ โดยการเพิ่มจำนวนระดับของวงจรขึ้นมากอีก 1 ระดับ ,ซึ่งวิธีการดังกล่าวนี้จะเรียกว่า SLA (SECOND – LEVEL AUXILIARY) สำหรับ ตัวอย่างของการใช้วิธี SLA นี้จะแสดงในรูปที่ 4.8 ซึ่งเป็นตัวอย่างของการบวกเลขฐานสองขนาด 24 บิท



$$C_{14} = G_{14} + P_{14}G_{13} + P_{14}P_{13}G_3^I + P_{14}P_{13}P_3^IG_1^{II} + P_{14}P_{13}P_3^IP_1^{II}C_0$$

$$C_{22} = G_{22} + P_{22}G_{21} + P_{22}P_{21}G_5^I + P_{22}P_{21}P_5^IG_2^{II} + P_{22}P_{21}P_5^IP_2^{II}G_1^I + P_{22}P_{21}P_5^IP_2^{II}P_1^IC_0$$

$$G_2^{II} = G_4^I + P_4^IG_3^I$$

รูปที่ 4.8 วงจรบวกแบบ CARRY LOOK – AHEAD ขนาด 24 บิท ที่ใช้ SLA เข้าช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 ในส่วนของวงจรในระดับที่ 0 และระดับที่ 1 ซึ่งใช้ในการสร้าง G_k , P_k และ G_n^I , P_n^I ตามลำดับนั้น จะเหมือนกับวงจรในระดับที่ 0 และระดับที่ 1 ของวงจรที่ใช้หลักการของ FLA สำหรับวงจรในระดับที่ 2 (SLA) นั้นจะถูกสร้างขึ้นโดยใช้หลักการและสมการเดียวกับวงจรในระดับที่ 1 (FLA) เพียงแต่ตัวยกหรือ SUPERSCRIPT ของ G และ P ในสมการนั้นจะถูกเพิ่มขึ้น 1 ดังนี้

สมการของวงจรในระดับที่ 1 (FLA)

$$G_N^I = G_K + P_k G_{k-1} + \ldots + P_k P_{k-1} \ldots P_{k-g+2} G_{k-g+1}$$

$$P_n^I = P_k P_{k-1} \ldots P_{k-g+2} + P_{k-g+1}$$

สมการของวงจรในระดับที่ 2 (SLA)

$$G_n^{II} = G_k^I + P_k^I G_{k-1}^I + \ldots + P_k^I P_{k-1}^I \ldots P_{k-g+2}^I G_{k-g+1}^I$$

$$P_n^{II} = P_k^I P_{k-1}^I \ldots P_{k-g+2}^I + P_{k-g+1}^I$$

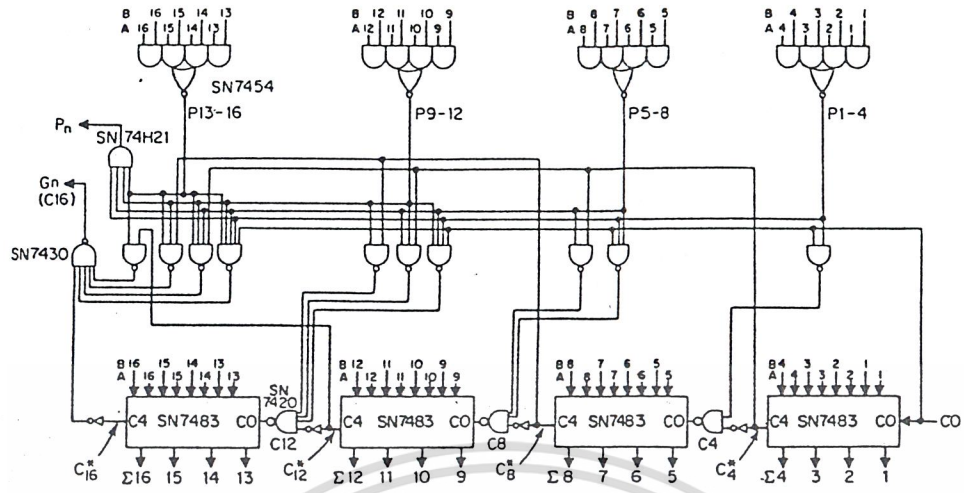
สำหรับในส่วนของวงจรสร้างตัวทดยังคงอาศัยหลักการเดิม เพียงแต่ใช้ทั้ง G_k , P_k , G_n^I , P_n^I และ G_n^{II}, P_n^{II} ร่วมกันเพื่อลดความซับซ้อนของวงจรลงเท่านั้น ดังตัวอย่างสมการของตัวทดบิทที่ 14 และ 22 (C_{14} และ C_{22}) ที่แสดงไว้ในรูปที่ 4.8 อย่างไรก็ตามเนื่องจากการที่จำนวนระดับของวงจรเพิ่มขึ้น จึงทำให้วงจรบวกที่ใช้หลักการ FLA และ SLA นั้น มีความเร็วลดลงต่ำกว่าวงจรบวกแบบ CARRY LOOK – AHEAD ทั่วไป ทั้งนี้ขึ้นอยู่กับกลุ่มของบิทที่ทำการแบ่งไว้นั้น จะมีจำนวนบิทมากหรือน้อยเพียงใดในกลุ่ม ส่วนกรณีของวงจรบวกที่ใช้หลักการ SLA นั้นความเร็วในการคำนวณจะเพิ่มขึ้นเป็นค่ายกกำลังสองตามการเพิ่มของจำนวนบิทในกลุ่ม

นอกจากการใช้หลักการ FLA และ SLA ในการเพิ่มวงจรในระดับที่ 0 และ 1 แล้ว เรายังสามารถใช้หลักการดังกล่าวในการเพิ่มระดับของวงจรขึ้น เพื่อลดความยุ่งยากในการสร้างวงจรลงได้อีกด้วย โดยส่วน

ของวงจรสร้างตัวทดและวงจรบวกยังคงใช้หลักการเดิม

การออกแบบวงจรบวกเลขฐานสองนี้ นอกจากการเลือกใช้เฉพาะวงจรบวกที่มีการบวกตัวทดแบบอนุกรม (RIPPLE CARRY) หรือวงจรบวกแบบ CARRY LOOK – AHEAD (รวมถึงวงจรที่ใช้หลักการ FLA และ SLA ด้วย) แล้วเรายังสามารถออกแบบวงจรบวกที่ใช้ทั้งหลักการของวงจรที่มีการบวกตัวทดแบบอนุกรมและวงจรแบบ CARRY LOOK – AHEAD ร่วมกันได้ ดังตัวอย่างของวงจรบวกเลขฐานสองขนาด 16 บิท ที่แสดงในรูป 4.9

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้ทำสำเนาหรือดัดแปลงส่วนใดส่วนหนึ่งของเอกสารนี้ เพื่อการค้า



รูปที่ 4.9 วงจรบวกเลขฐานสองขนาด 16 บิท ที่ใช้วงจรบวกตัวทดแบบอนุกรมร่วมกับวงจรบวกแบบ CARRY LOOK – AHEAD

จากรูปที่ 4.9 จะเห็นได้ว่าเราใช้วงจรบวกขนาด 4 บิท เบอร์ SN7483 ซึ่งเป็นวงจรบวกเลขฐานสองที่มีการบวกตัวทดแบบอนุกรม จำนวน 4 วงจร ร่วมกับวงจรสร้างตัวทดแบบ CARRY LOOK – AHEAD โดยที่วงจรสร้างเฉพาะตัวทดจากบิทที่ 4 ของวงจรบวกเบอร์ SN7483 แต่ละวงจรเท่านั้น (จากขาเอาท์พุท C_4 ของแต่ละวงจร) และเนื่องจากวงจรสร้างตัวทดแบบ CARRY LOOK – AHEAD ในรูปที่ 4.9 จะมีลักษณะเป็นวงจรที่ใช้หลักการ FLA ที่มีการแบ่งจำนวนบิทเป็น 4 กลุ่ม กลุ่มละ 4 บิท จึงจะมีสมการของวงจรสร้างตัวทดเป็น :

$$C_4 = G_1' + P_1' + C_0$$

$$C_8 = G_2' + P_2'G_1' + P_2'P_1'C_0$$

$$C_{12} = G_3' + P_3'G_2' + P_3'P_2'G_1' + P_3'P_2'P_1'C_0$$

$$C_{16} = G_4' + P_4'G_3' + P_4'P_3'G_2' + P_4'P_3'P_2'P_1'G_1' + P_4'P_3'P_2'P_1'C_0$$

ในที่นี้ $G_n' = G_k + P_k G_{k-1} + P_kP_{k-1}G_{k-2} + P_kP_{k-1}P_{k-2}G_{k-3}$

(ในกรณีนี้ $g = 4$ ทำให้ $k - g + 1 = k - 3$)

และจากสมการ

$$C_k = G_k + P_kG_{k-1} + P_kP_{k-1}G_{k-2} + \ldots + P_kP_{k-1}P_{k-2} \ldots P_1C_0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากในกรณีของเอาท์พุท C_4 จากวงจรบวกเบอร์ SN74863 (C_{4n}^{*}) แต่ละวงจรเป็นตัวทดที่ได้จากการบวกเฉพาะเลขฐานสองขนาด 4 บิท สองจำนวนโดยไม่รวมกับตัวทดและผลบวกจากวงจรอื่น ดังนั้นเทอมของ G และ P ในสมการของตัวทด C_0 จึงถูกตัดออก ทำให้ได้สามารของตัวทดจากวงจรเบอร์ SN7483 แต่ละวงจร (C*) เป็น

$$C_k = G_k + P_k G_{k-1} + P_k P_{k-1} G_{k-2} + P_k P_{k-1} P_{k-2} G_{k-3} = C_{4n^*}$$
$$= G_n^{'}$$

โดยที่ n = หมายเลขของกลุ่มของบิท โดยบิทที่ 4 คือกลุ่มที่ 1 และบิทที่ 16 คือกลุ่มที่ 4 ($n = k/4$)

สำหรับสมการของ $P_n^{'}$ จะเป็น

$$P_n^{'} = P_k P_{k-1} P_{k-2} P_{k-3} \text{ หรือ } P_{k-(k-3)}$$

สมการของตัวทดทั้ง 4 บิท จึงเขียนใหม่ได้ดังนี้:

$$C_4 = C_4^* + P_1 P_2 P_3 P_4 C_0 \text{ หรือ } C_4^* + P_{1-4} C_0$$
$$C_8 = C_8^* + P_{5-8} C_4^* + P_{1-8} C_0$$
$$C_{12} = C_{12}^* + P_{9-12} C_8^* + P_{5-12} C_4^* + P_{1-12} C_0$$
$$C_{16} = C_{16}^* + P_{13-16} C_{12}^* + P_{9-16} C_8^* + P_{5-16} C_4^* + P_{1-16} C_0$$

การทำงานของวงจรในรูปที่ 4.9 จะแบ่งออกเป็น 2 ส่วนคือ

1. **วงจรบวกเบอร์ SN7483** : ในส่วนของวงจรนี้จะทำการบวกเลขฐานสองแบบขนานและทำการบวกตัวทดแบบอนุกรม ซึ่งในกรณีที่เกิดตัวทดขึ้นในการบวกบิทแรก (บิท LSB) ของวงจร จะทำให้ตัวทดถูกส่งผ่านวงจรอีก 3 ระดับก่อนที่จะออกมาทางเอาท์พุท C_4 (C_{4n^*}) ของวงจร ซึ่งเป็นกรณีที่ตัวทดต้องใช้เวลามากที่สุดก่อนที่จะถูกส่งออกมาทางเอาท์พุท (WORST CASE) สำหรับผลบวกที่ถูกต้องนั้นจะได้รับหลังจากที่วงจรบวกได้รับอินพุทตัวทด C_0 จากวงจรสร้างตัวทด

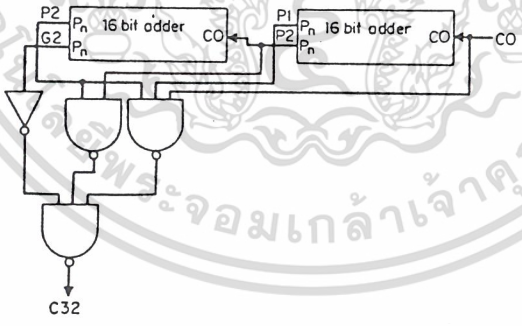
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรสร้างตัวทด : วงจรในส่วนนี้จะเป็นวงจรสร้างตัวทดแบบ CARRY LOOK – AHEAD ซึ่งจะทำการสร้างตัวทดเฉพาะในบิทที่ 4 , 8 , 12 และ 16 โดยใช้สมการของตัวทดทั้งสี่บิท ที่แสดงไว้ข้างต้น

วงจรบวกในรูปที่ 4.9 นั้นจะเตรียมเอาท์พุทไว้อีก 2 เอาท์พุทคือ G_n และ P_n โดยที่

$$G_n = C_{16} \text{ และ } P_n = P_{1-16}$$

สำหรับเอาท์พุท G_n และ P_n นี้จะถูกใช้ในกรณีที่ต้องการเพิ่มจำนวนบิทของเลขที่นำมาบวกกัน ดังตัวอย่างของการเพิ่มจำนวนบิทของเลขฐานสองที่นำมาบวกกันเป็น 32 บิท ในรูปที่ 4.10 ซึ่งใช้ในวงจรบวกเลขขนาด 16 บิท ในรูปที่ 4.9 สองวงจรมาประกอบกัน โดยใช้เอาท์พุท G_n และ P_n ของวงจรทั้งสองร่วมกันในการสร้างเอาท์พุทตัวทดบิทที่ 32 นอกจากนี้เราสามารถใช้หลักการของวงจรในรูปที่ 4.10 ในการเพิ่มจำนวนบิทของเลขที่นำมาบวกกันได้ โดยการเพิ่มจำนวนอุปกรณ์ที่ใช้ในวงจรขึ้นตามจำนวนบิทที่ต้องการ เช่น การเพิ่มจำนวนบิทของวงจรขึ้นเป็น 28 เราสามารถจะทำได้โดยการเพิ่มจำนวนของวงจรเบอร์ SN74863 ขึ้นอีก 3 วงจร (7*4 = 28) และจำเป็นต้องใช้วงจรเกท HAND ขนาด 8 อินพุท ในวงจรสร้างตัวทด นอกจากนี้อาจจำเป็นต้องเพิ่มวงจรขับสำหรับวงจรส่วนที่สร้าง P_n ด้วย ทั้งนี้เนื่องจากจำนวนโหลดที่เพิ่มขึ้นนั่นเอง



รูปที่ 4.11 วงจรบวกเลขฐานสองขนาด 32 บิท ที่สร้างขึ้นจากวงจรบวกในรูปที่ 4.10 จำนวน 2 วงจร

นอกจากการสร้างวงจรโดยใช้หลักการร่วมกันระหว่างการบวกตัวทดแบบอนุกรมกับแบบ CARRY LOOK - AHEAD ดังที่ได้แสดงไว้ในตัวอย่างข้างต้นแล้ว เรายังสามารถสร้างวงจรที่ใช้หลักการในลักษณะนี้ แต่มีการผสมกันระหว่างวงจรบวกทั้งสองแบบในรูปที่แตกต่างกันได้ ทั้งนี้การที่จะเลือกใช้วงจรในรูปแบบใดนั้น ขึ้นอยู่กับคุณสมบัติการทำงานของวงจรที่เราต้องการ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

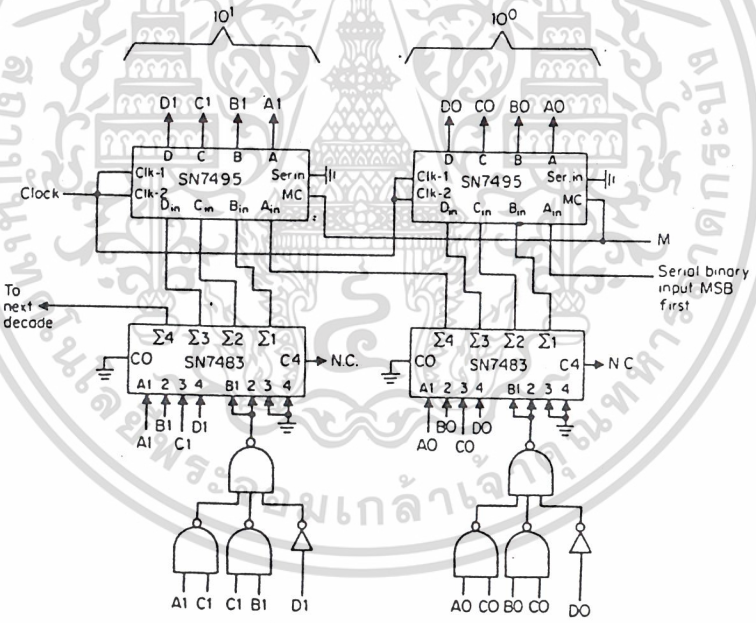
จำนวนบิทของเลขฐานสองที่ต้องการนำมาบวกกัน ,ความเร็วของวงจร , ราคา , ชนิดของอุปกรณ์ที่ใช้เพื่อให้เกิดความสะดวกและประหยัดที่สุด เป็นต้น

การประยุกต์ใช้วงจรบวกในการแปลงรหัสเลขฐานสอง

วงจรบวกเลขฐานสองขนาด 4 บิท เบอร์ SN7483 ที่ได้กล่าวถึงในตอนต้นนั้น เราสามารถนำมาประยุกต์ใช้ในการสร้างวงจรแปลงรหัสเลขฐานสองแบบต่าง ๆ ได้ ซึ่งในหัวข้อนี้จะได้กล่างถึงการนำเอาวงจรบวกดังกล่าวมาประยุกต์ใช้ในงานลักษณะนี้

วงจรแปลงเลขฐานสองแบบอนุกรมเป็นเลข BCD แบบขนาน

สำหรับหลักการแปลงเลขฐานสองเป็นเลข BCD นั้นได้กล่าวถึงแล้วในบทที่ 6 ในที่นี้จึงเพียงแสดงตัวอย่างวงจรที่สร้างขึ้นโดยใช้วงจรเบอร์ SN7483 เข้าช่วยเท่านั้น ดังตัวอย่างวงจรในรูปที่ 4.12



รูปที่ 4.12 วงจรแปลงเลขฐานสองแบบอนุกรมเป็นเลข BCD แบบขนานขนาด 2 หลัก

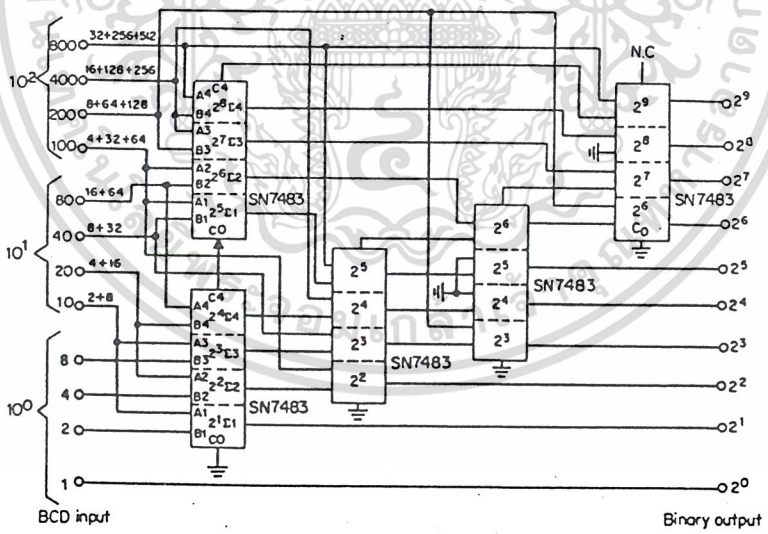
วงจรในรูปที่ 4.12 เป็นวงจรแปลงเลขฐานสองเป็นเลข BCD ขนาด 2 หลัก (8 บิท) ซึ่งประกอบด้วยวงจรเลื่อนบิทข้อมูลแบบอนุกรมที่ให้เอาท์พุทเป็นแบบขนาน เบอร์ SN7495 สองวงจร , วงจรบวกแบบขนานเบอร์ SN7483 สองวงจร และวงจรเกทที่ควบคุมการแปลงรหัส สำหรับการทำงานของวงจรจะเริ่มจากการลบข้อมูลเดิมของวงจรเลื่อนบิทข้อมูลทั้งสองวงตรก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการป้อนอินพุทที่เป็นระดับลอจิก "0" ให้กับดินพุท M ของวงจรนานเท่ากับช่วงเวลาของสัญญาณคล็อก 4 ลูก จากนั้นวงจรจะเริ่มการทำการเลื่อนบิทและแปลงรหัสเมื่อสัญญาณที่ขาอินพุท M เปลี่ยนเป็นระดับลอจิก "1" โดยวงจรบวกและวงจรเกทที่ต่ออยู่กับอินพุท B1 และ B2 ของวงจรบวกทั้งสองวงจรนั้น จะทำหน้าที่ในการบวกข้อมูลที่ได้รับจากวงจรเลื่อนบิท (A0 – D0 และ A1 – D1) ด้วย 0 เมื่อข้อมูลนั้นมีค่าตั้งแต่ 0 ถึง 4 และ จะบวกข้อมูลที่ได้รับจากวงจรเลื่อนบิทด้วย 3 (ก่อนที่จะมีการเลื่อนบิทในครั้งต่อไป) เมื่อข้อมูลนั้นมีค่าตั้งแต่ 5 ถึง 9 ซึ่งจะเป็นผลให้หลังจากการเลื่อนบิทแล้วข้อมูลในวงจรเลื่อนบิท (คือ เอาท์พุท A0 – D0 และ A1 – D1 ของวงจรนั่นเอง) นั้นมีค่าไม่เกิน 9 การแปลงเลขฐานสองเป็นเลข BCD ในวงจรนี้จะใช้ช่วงเวลาของสัญญาณคล็อกเท่ากับ n ลูก สำหรับการแปลงเลขฐานสองขนาด n บิท

วงจรแปลงเลข BCD แบบขนานเป็นเลขฐานสองแบบขนาน

สำหรับตัวอย่างวงจรแปลงเลข BCD เป็นเลขฐานสองนี้จะแสดงในรูปที่ 4.13 ซึ่งเป็นวงจรที่ถูกจัดให้มีการทำงานที่ตรงที่สุดเพื่อให้สามาถทำการแปลงรหัสได้ที่ความเร็วสูง โดยความเร็วในการทำงานจะขึ้นอยู่กับค่าดีเลย์ภายในวงจรบวก ซึ่งมีค่าประมาณ 250 ns สำหรับตัวอย่างวงจรที่แสดงไว้นี้เราสามารถนำหลักการไปดัดแปลงเพื่อใช้กับเลข BCD ที่มีจำนวนหลักสูงขึ้นได้

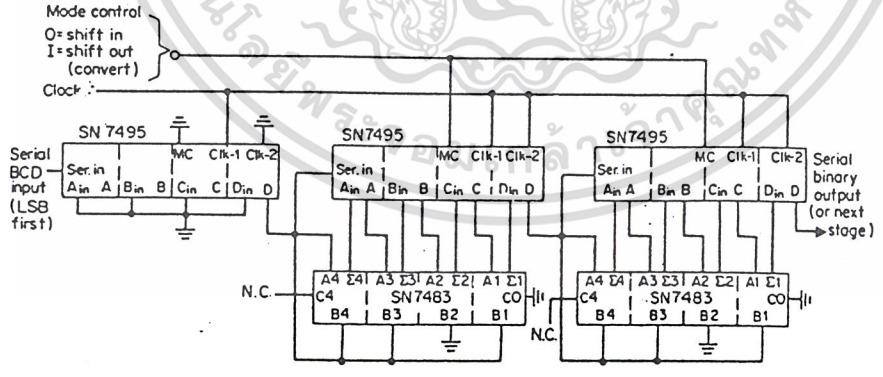


รูปที่ 4.13 วงจรแปลงเลข BCD แบบขนานขนาด 3 หลัก เป็นเลขฐานสองแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรแปลงเลข BCD แบบอนุกรมเป็นเลขฐานสองแบบอนุกรม

วงจรในรูปที่ 4.14 จะแสดงวงจรแปลงเลข BCD ขนาด 3 หลักแบบอนุกรมเป็นเลขฐานสองแบบอนุกรม โดยการทำงานของวงจรจะเริ่มต้นจากการเลื่อนบิทของเลข BCD ขนาด 3 หลัก (หลักที่มีนัยสำคัญสูงสุดจะอยู่ทางด้านซ้ายสุด และหลักที่มีนัยสำคัญต่ำสุดจะอยู่ทางด้านขวาสุด) เข้าไปเก็บไว้ในรีจิสเตอร์ที่ประกอบด้วยวงจรเลื่อนบิทเบอร์ SN7495 จำนวน 3 วงจร ซึ่งในช่วงเวลานี้อินพุท M ของวงจรจะต้องได้รับลอจิกเป็น "0" และหลังจากที่เลข BCD ทั้ง 3 หลักถูกเลื่อนเข้าไปเก็บไว้ในรีจิสเตอร์เรียบร้อยแล้ว เลข BCD จะถูกแปลงเป็นเลขฐานสองและถูกเลื่อนออกทางเอาท์พุทแบบอนุกรม เมื่ออินพุท M ได้รับลอจิก เป็น " 1 " สำหรับในส่วนของวงจรแปลงเลข BCD เป็นเลขฐานสองจะประกอบด้วยวงจรบวกเบอร์ SN7483 ซึ่งถูกจัดวงจรให้ทำการลบข้อมูลในรีจิสเตอร์ด้วย 3 (ในทางปฏิบัติก็คือการบวกด้วยค่า2's complement ของ 3 คือ 1101_2) เมื่อบิทที่มีข้อมูลเป็น "1" ถูกเลื่อนผ่านจากหลักที่มีนัยสำคัญสูงกว่าเข้าไปในหลักที่มีนัยสำคัญต่ำกว่า (ในที่นี้จะทำการลบก่อนที่มีการเลื่อนบิท คือ จะทำการลบเมื่อข้อมูลในบิท LSB หรือเอาท์พุท D ของหลักที่มีนัยสำคัญสูงกว่ามีข้อมูลเป็น "1" แล้วจึงทำการเลื่อนบิท ทั้งนี้ขอให้ดูวิธีการแปลงเลข BCD เป็นเลขฐานสอง) สำหรับวงจรเลื่อนบิทเบอร์ SN 7495 นั้นเมื่อได้รับลอจิก "1" ทางอินพุท M วงจรเลื่อนบิทจะรับข้อมูลแบบขนาน ซึ่งถูกแปลงเป็นเลขฐานสองแล้วทางอินพุท Ain-Din จากเอาท์พุทของวงจรบวก และวงจรเลื่อนบิทนี้จะถูกจัดให้ทำการเลื่อนบิทออกทางเอาท์พุทแบบอนุกรมด้วย (เอาท์พุท D ของวงจรเลื่อนบิททางขวาสุด) สำหรับการเพิ่มจำนวนหลักของเลข BCD นั้นสามารถทำได้โดยใช้เอาท์พุท D นี้เป็นอินพุทให้กับวงจรเลื่อนบิทในหลักที่เพิ่มขึ้น



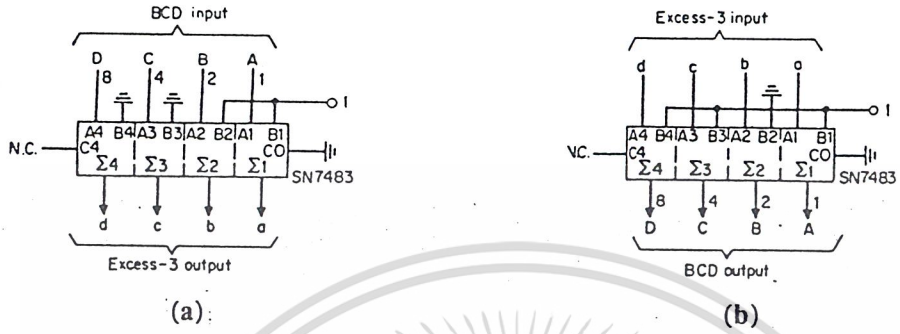
รูปที่ 4.14 วงจรแปลงเลข BCD แบบอนุกรมขนาด 3 หลักเป็นเลขฐานสองแบบอนุกรม

วงจรแปลงระหว่างเลข BCD และรหัส Excess-3 :

การแปลงเลข BCD เป็นรหัส Excess-3 สามารถทำได้โดยการบวกเลข BCD ด้วย 3 และในทางกลับกันการแปลงรหัส Excess-3 เป็น BCD ก็จะทำได้โดยการลบรหัส Excess-3 นั้นด้วย 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในทางปฏิบัติก็คือการบวกด้วย 13_{10} หรือ 1101_2 นั่นเอง สำหรับตัวอย่างของวงจรแปลงเลข BCD เป็นรหัส Excess-3 และแปลงรหัส Excess-3 เป็นเลข BCD ขนาด 1 หลักเท่านั้นจะแสดงไว้ดังรูปที่ 4.15(a) และ 4.15(b) ตามลำดับ



รูปที่ 4.15 (a) วงจรแปลงเลข BCD ขนาด 1 หลักเป็นรหัส Excess-3
(b) วงจรแปลงเลขรหัส Excess-3 เป็นเลข BCD ขนาด 1 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วงจรคูณเลขฐานสอง

ในกรณีของวงจรคูณเลขฐานสองที่สร้างขึ้นโดยใช้วิธีการทางดิจิตอลนั้น สามารถทำได้ในหลายลักษณะตั้งแต่การใช้วิธีการบวกตัวตั้งตามจำนวนบิทของตัวคูณ ซึ่งเป็นวิธีการที่ทำได้ง่ายและสะดวก แต่วงจรที่ได้จะมีความเร็วในการทำงานต่ำมาก จนถึงวิธีการสร้างวงจรที่สามารถให้เอาท์พุทเป็นผลคูณได้โดยตรงเมื่อได้รับอินพุทเป็นตัวตั้งและตัวคูณ ซึ่งวิธีการนี้จะทำให้ได้วงจรคูณที่มีความเร็วสูง แต่จะทำให้วงจรมีความซับซ้อนมากขึ้นเป็นทวีคูณ เมื่อจำนวนบิทของเลขที่นำมาคูณกันเพิ่มขึ้น

สำหรับตัวอย่างวงจรคูณที่จะกล่าวถึงในที่นี้ เป็นวงจรที่สร้างขึ้นจากหลักการคูณเลขที่ใช้อยู่โดยทั่วไป ซึ่งวิธีการดังกล่าวนี้จะมีความเร็วในการทำงานสูงกว่าวิธีการบวกตัวตั้งตามจำนวนบิทของตัวคูณในขณะที่ความซับซ้อนของวงจรจะต่ำกว่าวงจรที่สามารถให้เอาท์พุทเป็นผลคูณได้โดยตรง สำหรับวิธีการทั่วไปในการคูณเลขฐานสองขนาด 4 บิทที่มีตัวตั้งเป็น Y (สมมติเป็น 1010_2) และตัวคูณเป็น X (สมมติเป็น 1011_2) จะเป็นดังนี้ :

```
Y =   1010
X =   1011
      1010
     1010
    0000
   1010
ผลคูณเท่ากับ  1101110
```

จากวิธีการคูณดังที่ได้สรุปไว้ข้างต้น เราสามารถแยกออกเป็นขั้นตอนต่างๆ สำหรับการสร้างวงจรคูณได้ดังนี้ :

1.ในกรณีที่บิทแรกของตัวคูณมีข้อมูลเป็น “1” ให้โหลดค่าตัวตั้ง Y ลงมาเก็บไว้ในแอ็คคิวมูเลเตอร์ (ACCUMULATOR : เป็นรีจิสเตอร์สำหรับเก็บผลลัพธ์) ส่วนในกรณีที่บิทแรกของตัวคูณมีข้อมูลเป็น “0” ให้ข้ามขั้นตอนนี้ไป (ก่อนที่จะเริ่มต้นการคูณ ข้อมูลภายในแอ็คคิวมูเลเตอร์ต้องถูกเคลียร์ให้เป็น 0 หมดทุกบิท)

ลิขสิทธิ์เป็นของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ทำการคูณในบิทที่ 2 ของตัวคูณ โดยเลื่อนตัวตั้ง Y ไปทางซ้าย 1 บิท แล้วบวกเข้ากับข้อมูลในแอ็คคิวมูเลเตอร์เมื่อข้อมูลในบิทที่ 2 ของตัวคูณ X มีข้อมูลเป็น "1" แต่ในกรณีที่ข้อมูลในบิทที่ 2 ของตัวคูณ X มีข้อมูลเป็น "0" ให้ทำเฉพาะการเลื่อนบิทของตัวตั้ง Y โดยไม่ต้องทำการบวก

3.ทำการคูณในบิทที่มีนัยสำคัญสูงขึ้นของตัวคูณต่อไป (บิทที่ 3,4,5,...) จนกระทั่งครบทุกบิทโดยใช้หลักการที่ได้กล่าวถึงในขั้นตอนที่ 2 เพียงแต่เปลี่ยนจากบิทที่ 2 เป็นบิทที่ 3,4,5,...ตามลำดับ ซึ่งหลังจากทำการคูณในบิทสุดท้ายของตัวคูณแล้ว ข้อมูลที่เก็บไว้ในแอ็คคิวมูเลเตอร์จะเป็นผลลัพธ์ที่ได้จากการคูณ

จากวิธีการคูณที่ได้แสดงไว้นั้น จะเห็นได้ว่าการคูณในแต่ละบิทของตัวคูณจะประกอบด้วยขั้นตอนหลักเพียง 2 ขั้นตอนคือ การบวกตัวตั้งเข้ากับข้อมูลในแอ็คคิวมูเลเตอร์ (เฉพาะในกรณีที่ข้อมูลของตัวคูณในบิทนั้นมีข้อมูลเป็น "1" โดยในบิทแรกก็คือการบวกตัวตั้งเข้ากับข้อมูลที่มีค่าเป็น 0 นั่นเอง) และการเลื่อนบิทของตัวตั้ง Yไปทางซ้าย 1 บิทแต่ในทางปฏิบัติการเลื่อนบิทของตัวตั้งไปทางซ้ายจะทำให้จำเป็นต้องใช้จำนวนของรีจีสเตอร์ภายในวงจรเพิ่มมากขึ้น ดังนั้นในขั้นตอนการเลื่อนบิทจึงใช้การเลื่อนบิทของข้อมูลใน แอ็คคิวมูเลเตอร์ไปทางขวาแทนซึ่งจะให้ผลเท่ากับการเลื่อนบิทของตัวตั้ง Y ไปทางซ้ายก่อนที่จะทำการบวกข้อมูลที่เก็บไว้ในแอ็คคิวมูเลเตอร์ ด้วยข้อมูลของตัวตั้ง Y สำหรับตัวอย่างไดอะแกรมของวงจรคูณเลขขนาด 8 บิท 2 จำนวนที่ใช้หลักการดังที่กล่าวไว้ข้างต้น จะแสดงไว้ในรูปที่ 12.1

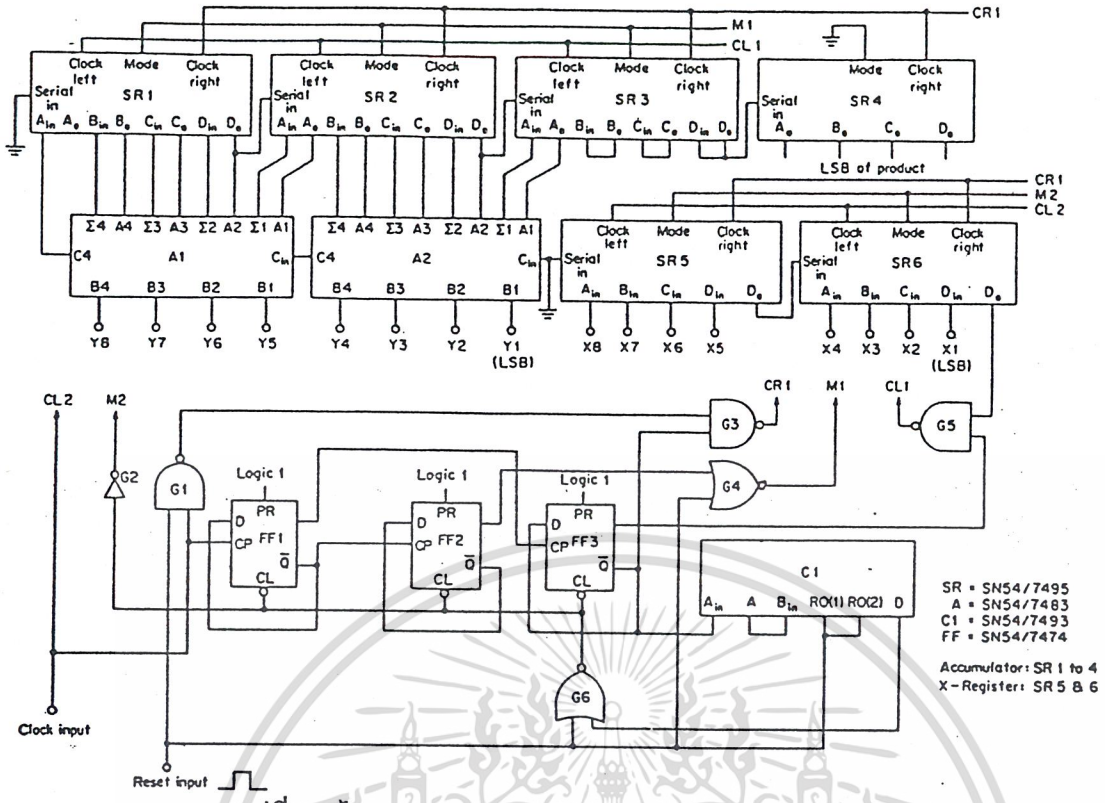
สำหรับวงจรในรูปที่ 12.1 จะประกอบด้วยวงจรหลักในส่วนต่างๆ คือ :

1.วงจรในส่วนของแอ็คคิวมูเลเตอร์ ประกอบด้วยวงจรเลื่อนบิทเบอร์ SN54/7495 ซึ่งสามารถนำมาใช้ในการเป็นรีจิสเตอร์สำหรับพักข้อมูลที่มีอินพุทและเอาท์พุทเป็นแบบขนาน ที่สามารถเลื่อนบิทของข้อมูลภายในได้ ด้วยการควบคุมลอจิกที่ป้อนให้กับขาอินพุท Mode Control โดยในส่วนของแอ็คคิวมูเลเตอร์นี้จะใช้วงจรเบอร์SN54/7495 จำนวน 4 วงจร คือ SR1 ถึง SR4

2.วงจรบวกข้อมูลในแอ็คคิวมูเลเตอร์ด้วยข้อมูลของตัวตั้ง (Y1 ถึง Y8) ประกอบด้วยวงจรบวกเลขฐานสองขนาด 4 บิทเบอร์ SN54/7483 จำนวน 2 วงจร คือ A1 และ A2

3.ริจิสเตอร์สำหรับเก็บข้อมูลที่เป็นตัวคูณ (X1 ถึง X8) ประกอบด้วยวงจรเลื่อนบิทเบอร์ SN54/7495จำนวน 2 วงจร คือ SR5 และ SR6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 ไดอะแกรมของวงจรคูณเลขฐานสองขนาด 8 บิท

4.วงจรสร้างสัญญาณคล็อกสำหรับใช้ภายในวงจร ซึ่งประกอบด้วยวงจรฟลิปฟลอปแบบ D เบอร์ SN54/7474 จำนวน 3 วงจร คือ FF1,FF2 และ FF3 ซึ่งเอาท์พุทของวงจรฟลิปฟลอป FF1 จะมีความถี่เป็น $\frac{1}{2}$ ของสัญญาณคล็อกที่ป้อนให้กับวงจร และจากการที่วงจร FF2 และ FF3 ได้รับอินพุทคล็อกจากเอาท์พุท $\overline{Q}$ และ Q ของวงจร FF1 ตามลำดับ จึงทำให้เอาท์พุทของวงจร FF2 และ FF3 มีความถี่เป็น $\frac{1}{4}$ ของสัญญาณคล็อกที่ป้อนให้กับวงจร (1/2ของเอาท์พุทของวงจร FF1) แต่มีเฟสที่ต่างกันอยู่ 90 องศา

การทำงานของวงจรในรูปที่ 12.1 จะใช้หลักการที่กล่าวถึงในตอนต้น ซึ่งสามารถแบ่งออกเป็นขั้นตอนต่างๆได้ดังนี้ :

1. หรับการเริ่มต้นการทำงานของวงจรจะเกิดขึ้น เมื่อวงจรได้รับสัญญาณพัลส์ของลอจิก "1" ที่ขาดินพุท RESET ซึ่งในช่วงที่สัญญาณที่ขาอินพุท RESET เป็นลอจิก "1" นั้น เอาท์พุท Q ของวงจรฟลิปฟลอป FF1 , FF2 และ FF3 จะถูกเคลียร์เป็นลอจิก "0" (ขาอินพุท CL ของวงจรฟลิปฟลอปทั้งสามได้รับลอจิกเป็น "0") ทำให้เอาท์พุทของวงจรเกท G4 หรือสัญญาณ M1 ที่ต่ออยู่กับขาอินพุท Mode Control ของวงจรเบอร์SN54/7495 ที่ประกอบเป็นแอ็คคิวมูเลเตอร์มีลอจิกเป็น "0" ด้วย ซึ่งเป็นผลให้วงจรเบอร์ SN54/7495 เข้าสู่โหมดการเลื่อนบิททางขวา (Right Shift) และจากการที่สัญญาณอินพุท RESET มีลอจิกเป็น "1" นี้จะทำให้สัญญาณคล็อกที่ป้อนให้กับวงจรถูกส่งผ่านวงจรเกท G1 และ G3 ออกมาเป็นสัญญาณคล็อก CR1 ป้อนให้

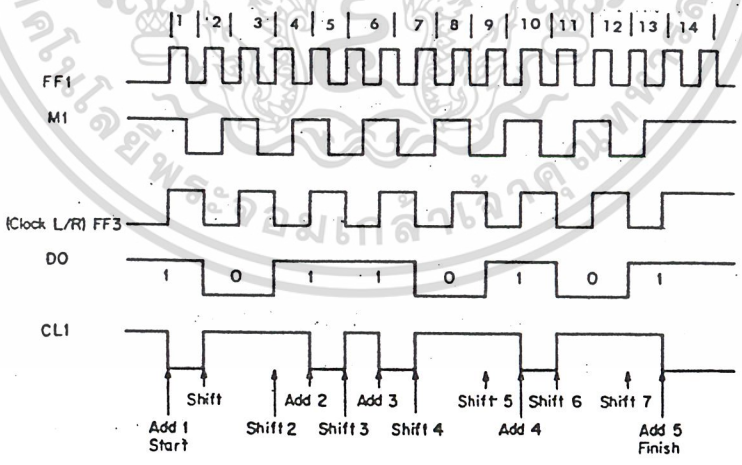
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับขาอินพุท Clock Right ทำให้วงจรแอคคิวมูเลเตอร์เริ่มทำการเลื่อนบิทของข้อมูลที่ป้อนเข้าทางขาอินพุท Serial In ของวงจร SR1 คือ "0" (ต่ออยู่กับกราวนด์) ไปทางขวาเรื่อย ๆ จนกระทั่งครบทั้ง 16 บิท ของวงจรแอ็คคิวมูเลเตอร์ ซึ่งจะทำให้ข้อมูลภายในแอ็คคิวมูเลเตอร์ถูกเคลียร์เป็น 0 หมดทุกบิท ดังนั้นสัญญาณ RESET จึงควรมีความกว้างของพัลส์ที่เป็นลอจิก "1" นานเท่ากับความกว้างของสัญญาณคล็อกที่ป้อนให้กับวงจรจำนวน 16 ลูกเป็นอย่างน้อย

ในช่วงที่สัญญาณ RESET มีลอจิกเป็น "1" นี้จะทำให้สัญญาณ M2 ที่ป้อนให้กับขาอินพุท Mode Control ของวงจร SR5 และ SR6 ซึ่งเป็นรีจิสเตอร์สำหรับเก็บตัวคูณ คือ X1 ถึง X8 จะถูกโหลดเข้าสู่วงจร เมื่อวงจรได้รับสัญญาณคล็อก (ขอบขาลง) ที่ป้อนให้กับวงจรผ่านทางขาสัญญาณ CL2

วงจรจะเริ่มต้นการคูณหลังจากที่สัญญาณอินพุท RESET เปลี่ยนกลับเป็นลอจิก "0" ซึ่งในช่วงนี้วงจร FF1 , FF2 และ FF3 จะเริ่มต้นสร้างสัญญาณคล็อก สำหรับสัญญาณ M1 และ CR1 จะได้จากสัญญาณคาล็อกที่เอาท์พุทของวงจร FF2 (เอาท์พุท Q) และ FF3 (เอาท์พุท) ตามลำดับ และสัญญาณ M2 จะเปลี่ยนเป็นลอจิก "0" ทำให้รีจิสเตอร์ที่เก็บตัวคูณ X (SR5 และ SR6) เข้าสู่โหมดการเลื่อนบิทางขวา (Right Shift)

2. สำหรับการศึกษาถึงการทำงานของวงจรในที่นี้จะกำหนดให้ตัวคูณ X มีข้อมูลเป็น 10101101_2 (บิทขวาสุดคือ LSB) ซึ่งจะทำให้ได้อะแกรมเวลาของวงจรเป็นดังรูปที่ 5.2



รูปที่ 5.2 ไดอะแกรมเวลาสำหรับการทำงานของวงจรคูณในรูปที่ 5.1

ในช่วงขอบขาขึ้นของสัญญาณคล็อกลูกแรก (ในที่นี้สัญญาณคลอ็กจะหมายถึงเฉพาะสัญญาณคลอ็กที่ได้จากเอาท์พุทของวงจร FF1 เท่านั้น) เอาท์พุทของวงจร FF3 (สัญญาณ CR1) จะเปลี่ยนเป็นลอจิก "1" ทำให้ค่าคอมพลีเมนท์ของเอาท์พุท D_0 จากวงจร SR6 ถูกส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมายังเอาท์พุทของวงจรเกท G5 (สัญญาณ CL1) ซึ่งในกรณีนี้เอาท์พุท D_0 จะมีลอจิกตามข้อมูลในบิท LSB ของตัวคูณ X คือ 1 ทำให้สัญญาณ CL1 ที่อยู่กับอินพุท Clock Left ของวงจร SR1 ถึง SR3 เปลี่ยนจากลอจิก "1" เป็น "0" (ขอบขาลง) และเนื่องจากในช่วงเวลานี้สัญญาณ M1 ที่ต่อยู่กับอินพุท Mode Control จะมีลิจิกเป็น "1" อยู่ การเกิดขอบขาลงขึ้นที่ขาอินพุท Clock Left จึงทำให้วงจรแอ็คคิวมูเลเตอร์ (วงจร SR1 ถึง SR3) ทำการโหลดข้อมูลของตัวตั้ง Y ที่ได้รับจามวงจรบวก A1 และ A2 มาเก็บไว้ สำหรับวงจรบวก A1 และ A2 นี้ถูกจัดวงจรให้ทำหน้าที่บวกตัวตั้ง Y เข้ากับข้อมูลที่เก็บไว้ในแอ็คคิวมูเลเตอร์ แล้วส่งผลบวกที่ได้กลับไปให้กับแอ็คคิวมูเลเตอร์อีกครั้ง ซึ่งใน กรณีของบิท LSB ของตัวคูณ X นี้ ข้อมูลในแอ็คคิวมูเลเตอร์เป็น 0 อยู่ เมื่อทำการบวกกับตัวตั้ง Y จึงได้ค่าเป็น ค่า Y เช่นเดียวกับการโหลดค่า Y ลงไปในแอ็คคิวมูเลเตอร์โดยตรง ซึ่งวิธีการทำงานในลักษณะนี้จะเท่ากับการคูณค่าตัวตั้ง Y ด้วยบิท LSB (บิทแรก :X1) ของตัวคูณ X อย่างไรก็ตามในกรณีที่ข้อมูลในบิท X1 ที่ถูกส่งออกมายังเอาท์พุท D_0 ของวงจร SR6 มีค่าเป็น 0 เอาท์พุทของวงจรเกท G5 ยังคงเป็นลอจิก "1" โดยไม่เกิดการเปลี่ยนแปลงใด ๆ ขึ้น ดังนั้นข้อมูลในแอ็คคิวมูเลเตอร์จะไม่ถูกบวกด้วยค่าตัวตั้ง Y ในการคูณด้วยบิทของตัวคูณที่มีค่าเป็น 0

3. หลังจากขั้นตอนการคูรตัวตั้ง Y ด้วยบิท X1 แล้ว วงจรจะต้องทำการเลื่อนบิทข้อมูลในแอ็คคิวมูเลเตอร์ไปทางขวา 1 บิท ซึ่งในช่วงขอบขาลงของสัญญาณคล็อกลูกแรกสัญญาณ M1 (เอาท์พุทของวงจร FF2) จะเปลี่ยนจากลอจิก "1" เป็น "0" ทำให้วงจรในส่วนแอ็คคิวมูเลเตอร์เข้าสู่โหมดการเลื่อนบิทข้อมูลไปทางขวา และข้อมูลในแอ็คคิวมูเลเตอร์จะถูกเลื่อนไป ทางขวา 1 บิท ในช่วงขอบขาลงของสัญญาณคล็อก CR1 (เอาท์พุทของวงจรเกท G3 ซึ่งได้จากเอาท์พุทของวงจร FF3) อย่างไรก็ตามจากที่ได้กล่าวถึงลักษณะการทำงานของวงจรเบอร์ SN54/7495 ไว้แล้วว่าการเปลี่ยนแปลงลอจิกของสัญญาณ M1 นั้น อาจจะทำให้เกิดความผิดพลาดในการทำงานของวงจรเลื่อนบิทได้ ถ้าสัญญาณอินพุท Clock Right และ Clock Left อยู่ในสภาวะลอจิกที่ไม่เหมาะสม (รายละเอียดได้กล่าวถึงแล้ว ในหัวข้อ " วงจรเลื่อนบิทในอนุกรม SN54/74 " ในก่อนหน้านี้) ดังนั้นสัญญาณคล็อกที่ใช้งานภายในวงจรถึงถูกจัดไว้ด้วยวงจร FF1 , FF2 และ FF3 ให้สัญญาณ CR1 (Clock Right : เอาท์พุทวงจร FF3) มีลอจิกเป็น "0" ในช่วงขอบขาขึ้นของสัญญาณ M1 และจัดให้สัญญาณ CL1 (Clock Left : เอาท์พุทวงจรเกท G5) มีลอจิกเป็น "0" ในช่วงขอบขาขึ้นของสัญญาณ M1 สำหรับสัญญาณ M1 นี้จะถูกจัดโดยวงจร FF2 ให้มีสถานะของสัญญาณ CR1 และ CL1 ด้วยเพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้เนื่องจากสถานะของสัญญาณคล็อกที่วงจรเลื่อนบิทได้รับ เมื่อมีการเปลี่ยนแปลงสถานะของสัญญาณ M1 (อินพุท Mode Control)

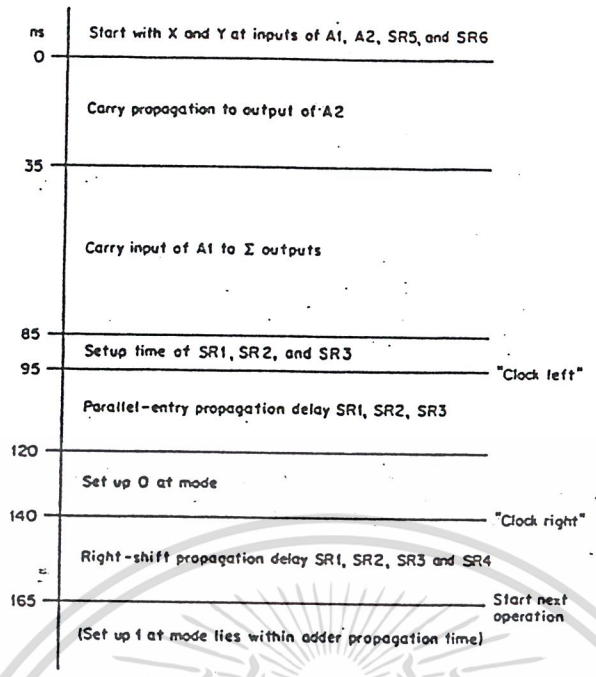
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สำหรับในส่วนของวงจร SR5 และ SR6 ที่ใช้สำหรับเก็บข้อมูลที่เป็นตัวคูณ X นั้น จะทำการเลื่อนบิทของตัวคูณ X ไปทางขวา 1 บิทในช่วงขอบขาลงของสัญญาณ CR1 เช่นเดียวกับในแอ็คคิวมูเลเตอร์ซึ่งหลังจากการเลื่อนบิทนี้แล้วเอาท์พุท D_0 ของวงจร SR6 จะเป็นข้อมูลในบิท X_2 จากนั้นในช่วงขอบขาลงของสัญญาณคล็อกลูกที่ 2 สัญญาณ M1 จะกลับเป็นลอจิก "1" อีกครั้ง ซึ่งเป็นการสิ้นสุดขั้นตอนการคูณในบิทแรก จากนั้นวงจรจะวนกลับไปทำงานตามขั้นตอนที่ 2 ถึง 4 เพื่อทำการคูณตัวตั้ง Y ด้วยบิทของตัวคูณที่มีนัยสำคัญสูงขึ้นไป (เปลี่ยนบิทของตัวคูณ) ในขั้นตอนที่ 2 ถึง 4 จาก X1 เป็น X2 , X3 ...จนถึง X8) โดยการทำงานของวงจรจะสิ้นสุดลง หลังจากที่การคูณในบิท X8 (บิท MSB) สิ้นสุดลง ซึ่งหลังจากการคูณสิ้นสุดลงแล้ว ผลลัพธ์ที่ได้จากการคูณจะถูกเก็บไว้ในแอ็คคิวมูเลเตอร์ (วงจร SR1 ถึง S) โดยที่เอาท์พุทของวงจร SR4 จะเป็นบิท LSB ของผลคูณที่ได้

สำหรับการหยุดการทำงานของวงจรคูณนั้น เราใช้วงจรนับเลขฐานสองขนาด 4 บิท เบอร์ SN54/7493 (วงจร C1) เข้าช่วยโดยที่วงจรจะรับสัญญาณคล็อกจากเอาท์พุท ของวงจร FF3 (ค่าคอมพลีเมนท์ของสัญญาณ CR1)

ซึ่งจะทำให้ ค่าการนับของวงจร C1 เพิ่มขึ้น 1 ในช่วงขอบขาขึ้นของสัญญาณ CR1 หรือทุกครั้งที่เกิดการคูณในแต่ละบิท (การทำงานในขั้นตอนที่ 2) และเมื่อวงจรทำการคูณจนครบทั้ง 8 บิทของตัวคูณแล้ว วงจร C1 จะได้รับสัญญาณคล็อก 8 ลูกด้วยเช่นกัน ซึ่งในช่วงเวลานี้เอาท์พุทบิท D ของวงจร C1 จะมีลอจิกเป็น "1" ทำให้เอาท์พุทของวงจรเกท G6 มีลอจิกเป็น "0" เป็นผลให้วงจรสร้างสัญญาณคล็อก FF1 ถึง FF3 ถูกเคลียร์ และการทำงานของวงจรจะสิ้นสุดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 ค่าดีเลย์ที่เกิดขึ้นในขั้นตอนต่าง ๆ ของการคูณ

ในรูปที่ 5.3 จะแสดงค่าดีเลย์ต่าง ๆ ที่เกิดขึ้นในการคูณของวงจรตามขั้นตอนที่ 2 ถึง 4 (การบวกและเลื่อนบิท สำหรับการคูณในแต่ละบิท) ซึ่งจะเป็นส่วนสำคัญในการกำหนดความเร็วสูงสุด (รวมถึงค่าความถี่สูงสุดของสัญญาณคล็อก) ของวงจร สำหรับค่าดีเลย์ที่เกิดขึ้นจากขั้นตอนที่ 2 ถึง 4 จะมีค่ารวมทั้งหมด 165 ns ซึ่งค่าดังกล่าวนี้จะช่วยให้เราสามารถคำรวณเวลาที่วงจรต้องใช้ในการคูณเลขที่มีขนาดต่าง ๆ กันได้เช่น ในการคูณเลขขนาด 32 ท บิท วงจรจะต้องใช้เวลาในการคำนวณอย่างต่ำประมาณ 5.3 ns (165 ns * 32) เป็นต้น

สำหรับการที่จะให้วงจรทำงานที่ความเร็วสูงสุดได้ดังที่แสดงไว้นั้น อาจจำเป็นต้องใช้วิธีการค่าง ๆ เข้าช่วยเพื่อลดผลจากสิ่งต่างๆ ที่มีผลต่อความเร็วในการทำงานของวงจรด้วย เช่น การกราวด์ หรือการทำดีคัปปลิ้ง (Decoupling) เป็นต้น นอกจากนี้ในส่วนของวงจรฟลิปฟลอปแบบ D ที่ใช้สร้างเป็นวงจรฟลิปฟลอปแบบ T เพื่อสร้างสัญญาณคล็อกนั้น เราสามารถเลือกใช้วงจร TTL ในอนุกรม SN54/74H (High – Speed) เพื่อให้สามารถรับสัญญาณที่มีความถี่สูงขึ้นได้

ในการเพิ่มขนาดของวงจรในรูปที่ 5.1 เพื่อให้สามารถทำการคูณเลขที่มีจำนวนบิทมากขึ้นนั้น จะทำได้โดยการเพิ่มจำนวนของวงจรเบอร์ SN54/7495 และ SN54/7483 ขึ้นอย่างละ 1 วงจร สำหรับทุก 4 บิทที่เพิ่มขึ้นของตัวตั้ง Y (เพิ่มวงจรเบอร์ SN54/7495 เข้าไปในส่วนของแอ็คคิวมูเลเตอร์ : SR และ เพิ่มวงจรเบอร์ SN54/7483 เข้าไปในส่วนของวงจรบวก : A) และเพิ่มวงจรเบอร์ SN54/7495 ขึ้น 2 วงจร สำหรับทุก 4 บิท ที่เพิ่มขึ้นของตัวคูณ X (เพิ่มวงจรเบอร์ SN54/7495 1 วงจรเข้าไปในส่วนของแอ็คคิวมูเลเตอร์ : SR1-4 และเพิ่มอีก 1 วงจร เข้าไปในส่วนของรีจิสเตอร์ ที่ใช้เก็บตัวคูณ X; SR5-6)

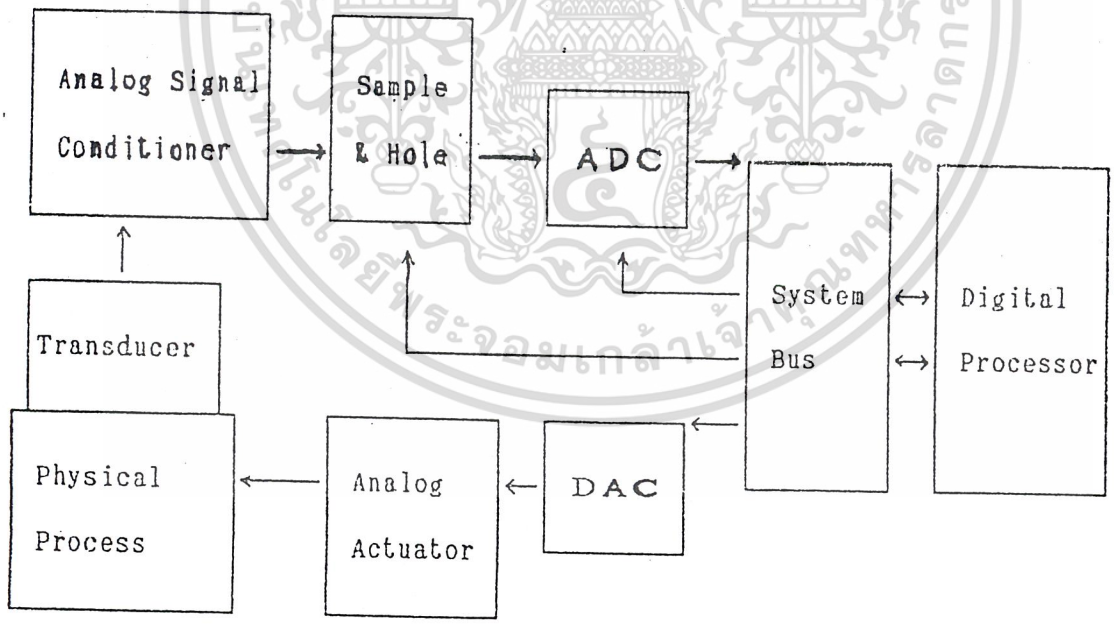
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

D/A และ A/D

6.1 ทฤษฎีการแปลงสัญญาณ (Data Acquistion and Conversion)

รูปแบบของสัญญาณทางไฟฟ้าที่เราคุ้นเคยกันแต่ดั้งเดิมนั้น โดยมากมักจะอยู่ในรูปของสัญญาณอนาลอก แต่ก่อนนั้นการที่จะนำเอาสัญญาณทางไฟฟ้ามาประมวลผล เพื่อให้เกิดรูปแบบตามที่ต้องการนั้นจะกระทำได้โดยใช้อุปกรณ์ทางอนาลอกนั่นเอง แต่ปัจจุบันเทคโนโลยีทางดิจิตอลก้าวหน้าไปมากทำให้การประมวลผลสัญญาณทางดิจิตอลสามารถทำได้อย่างรวดเร็ว มีประสิทธิภาพและเชื่อถือได้ ดังนั้นการเปลี่ยนรูปของสัญญาณ(Conversation) จึงได้มีความจำเป็นขึ้น จากสัญญาณอนาลอกที่มีอยู่แล้ว ได้ถูกเปลี่ยนให้เป็นสัญญาณดิจิตอล โดยอุปกรณ์ Analog to Digital Convertor และจะถูกประมวลผลโดยตัวประมวลผลสัญญาณดิจิตอล เช่น คอมพิวเตอร์ เป็นต้น จากนั้นผลลัพธ์ที่ได้ อาจถูกนำมาแสดงผลโดยตรงเลย หรือถูกเปลี่ยนกลับให้อยู่ในรูปของสัญญาณอนาลอกที่ใช้งานได้ การที่จะประมวลผลสัญญาณดิจิตอลกลับเป็นสัญญาณอนาลอกนั้นสามารถทำได้โดยใช้ Digital to Analog Converter



รูปที่ 6.1 แสดงระบบที่มีการประมวลผลข้อมูลทางดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้นสามารถอธิบายได้คือ การเปลี่ยนแปลงทางกายภาพในลักษณะใดๆก็ตาม เช่น อุณหภูมิ ความดัน ความเร็ว จะถูกเปลี่ยนให้มาเป็นสัญญาณไฟฟ้าแบบอนาลอกโดย Transducer ที่มีรูปแบบเหมาะสมกับ ลักษณะทางกายภาพของเครื่องวัดนั้นๆ จากนั้นสัญญาณทางไฟฟ้าก็จะถูกปรับให้อยู่ในรูปแบบ และขนาดที่เหมาะกับวงจร A/D โดยวงจรต่างๆเช้น วงจรขยายหรือวงจรกรองสัญญาณ เป็นต้น

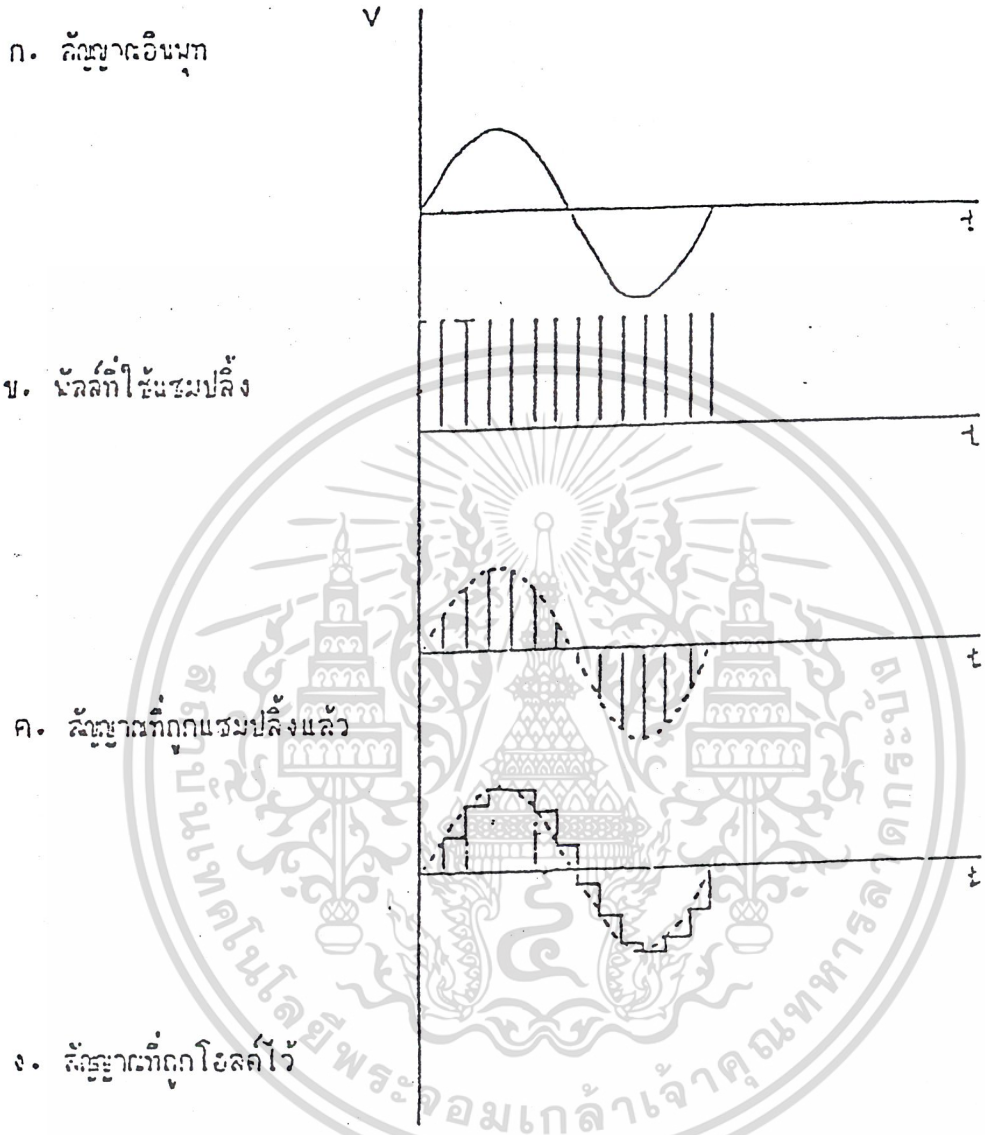
วงจร Sample and Hold จะสุ่มขนาดของสัญญาณอนาลอกออกมา แล้วทำการ Hold สัญญาณนั้นไว้ชั่วขณะ เพื่อให้ไม่จำเป็นต้องใช้ ADC ที่มีความเร็วมากนัก เมื่อสัญญาณถูกแปลงเป็นดิจิตอลโดย ADC แล้วข้อมูลดิจิตอลจะถูกส่งต่อไปยังบัสของระบบ จากนั้นตัวโปรเซสเซอร์จะทำการประมวลผลข้อมูล แล้วเปลี่ยนข้อมูลผลลัพธ์กลับมา เพื่อควบคุมกิจกรรมทางกายภาพของระบบโดยผ่านตัวกระทำของกล (Analog Actuator)

6.1.1 ทฤษฎีของการสุ่มข้อมูล (Sampling)

ในการสุ่มข้อมูลนั้นสัญญาณอนาลอกจะถูกสุ่มเป็นระยะคงที่ กลุ่มของสัญญาณที่สุ่มจะแทนด้วยความสูง ซึ่งเกิดจากการตัดต่อสัญญาณอนาลอกด้วยระยะเวลาอันสั้น ผลของการสุ่มด้วยความเร็ว จะเหมือนกับการคูณขบวนสัญญาณพัลซ์กับสัญญาณอนาลอก ซึ่งจะได้สัญญาณที่ Modulate ระหว่างขบวนพัลซ์กับสัญญาณอนาลอก ดังรูป และถ้าหากเรามีการ Hold สัญญาณที่สุ่มได้เอาไว้เราก็จะได้สัญญาณดังรูป ง

อัตราการสุ่มสัญญาณ หรือความถี่ของการสุ่มสัญญาณควรจะมีค่าเท่าใดที่ข้อมูลที่สุ่มได้นี้จะเป็นตัวแทนที่ดีของสัญญาณต่อเนื่องนั้น คำตอบคือขึ้นอยู่กับความถี่ของสัญญาณอนาลอก และจากทฤษฎีของการสุ่ม (Nyquist Theorem) กล่าวไว้ว่า " ถ้าสัญญาณดังกล่าวต่อเนื่องที่มี Harmonic Frequency) ไม่เกิน f แล้ว สัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาเช่นเดิม โดยไม่สูญเสียรายละเอียด หรือผิดเพี้ยนไป ถ้าอัตราการสุ่มมากกว่า 2f "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

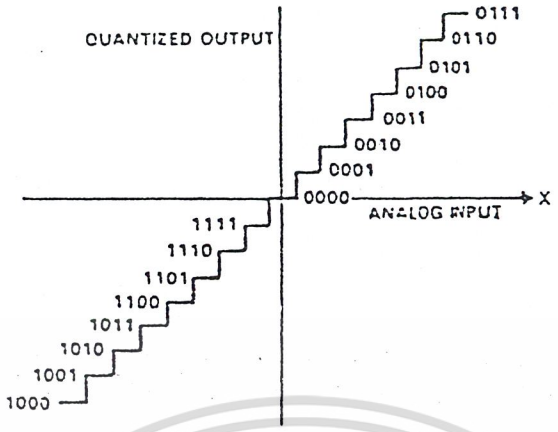


รูปที่ 6.2 แสดงการสุ่มสัญญาณ

6.1.2 ทฤษฎีการควอนไตซ์ (Quantizing Therorem)

การควอนไตซ์ เป็นขบวนการที่เปลี่ยนสัญญาณอนาลอก ให้เป็นข้อมูลทางดิจิตอลที่เป็นสัดส่วนกับสัญญาณอนาลอก เช่นอยู่ในรูปของรหัสไบนารี่ เป็นต้น หากเรานำoutput ที่ได้จากการควอนไตซ์ และขนาดของสัญญาณอนาลอกมาเขียนเป็นกราฟ ก็จะได้กราฟ Quantize Transfer Function) ดังแสดงในรูปที่ 6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดง Transfer Function ของการควอนไตซ์ 4 bit

จุดสำคัญของการควอนไตซ์นั้นได้แก่ Resolution ของตัวควอนไตซ์ ซึ่งสามารถกำหนดได้จากจำนวน bit ของ output รหัสดิจิตอล หรือจากกราฟรูปที่ 6.3 ก็คือส่วนกลับของความกว้างของขึ้นบันไดทางแกนนอนนั่นเอง

ค่าความกว้างของขั้นบันไดนี้ สามารถคำนวณได้จาก

$$Q = FSR / 2^n$$

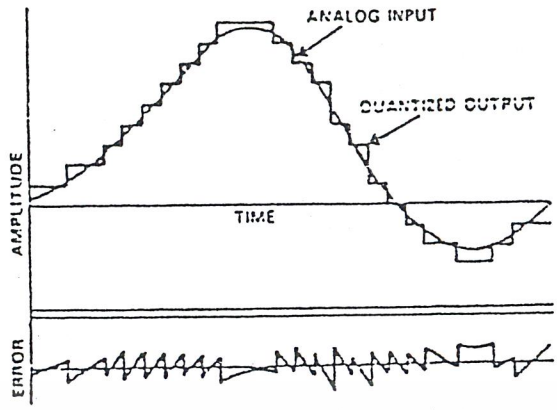
โดยที่ Q : ค่าความกว้างของขั้นบันไดทางแกนนอน

FSR : ช่วงเต็มสเกลของสัญญาณอนาลอก (Full Scale Range)

n : จำนวน bit ของรหัสดิจิตอล

จากสมการข้างต้นจะเห็นว่าเมื่อจำนวน bit มากขึ้น ขนาดของ Q ก็จะลดลง และถ้าเรานำสัญญาณอนาลอกใดๆ มาทำการควอนไตซ์ จะเห็นได้ว่า เมื่อนำเอาผลที่ได้จากการควอนไตซ์ มาเปรียบเทียบกับสัญญาณอนาลอกแล้ว ก็จะพบว่ามีข้อผิดพลาดเกิดขึ้น ซึ่งเราจะเรียกความผิดพลาดนี้ว่า ความผิดพลาดควอนไตซ์ (Quantizing Error) ดังแสดงในรูปที่ 6.4

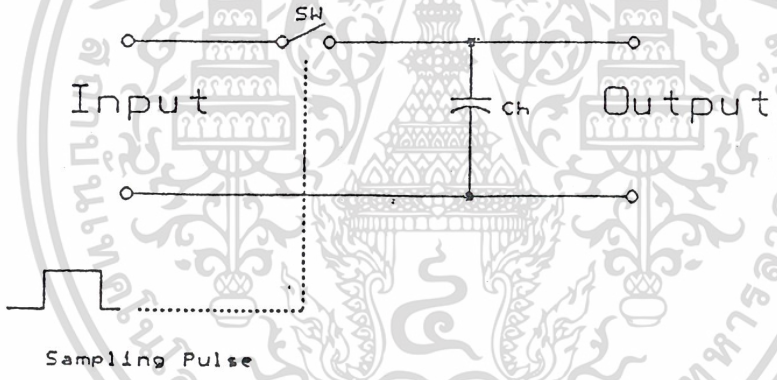
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 แสดงความผิดพลาดควอนไตซ์

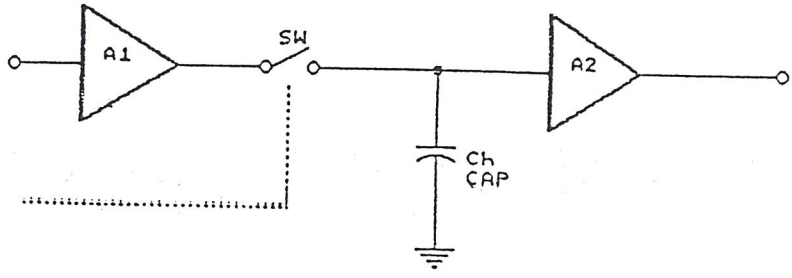
6.1.3 วงจร Sampling and Hold

วงจร Sampling and Hold โดยพื้นฐานแล้วเป็นวงจร หรืออุปกรณ์เก็บแรงดัน (voltage memory) ซึ่งใช้อุปกรณ์ที่สำคัญ คือ ตัวเก็บประจุ ดังแสดงในรูปที่ 6.5



รูปที่ 6.5 แสดงวงจรพื้นฐานของการ Sample and Hold

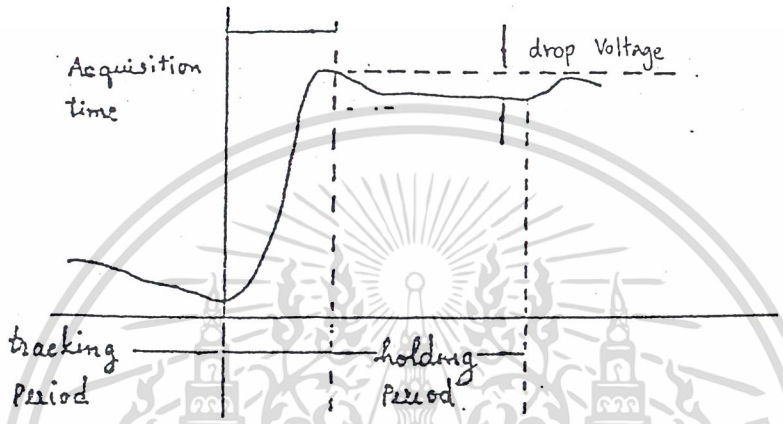
อิเล็คทรอนิคส์สวิทช์จะต่อสัญญาณแรงดันเข้าตัวเก็บประจุ ซึ่งสวิทช์จะถูกควบคุมจาก sampling pulse ช่วงเวลาการตัดสวิทช์ และเวลาในการประจุจนแรงดันถึงค่าที่สุ่มมานั้นจะเรียกว่า Aperture Time ของ Sample and Hold



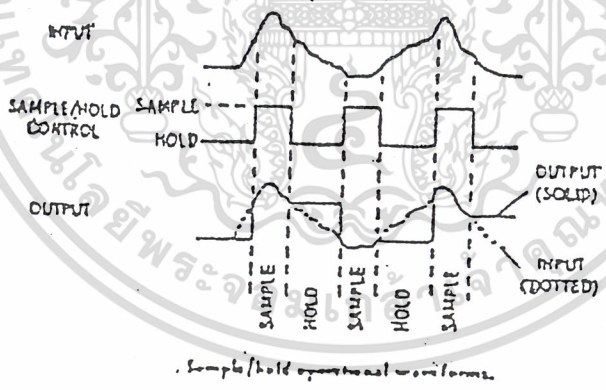
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.6 แสดงวงจร Sample and Hold ที่ใกล้เคียงกับวงจรที่ใช้งานจริง

จากรูปโดยการเพิ่ม Buffer Amplifier เข้าทางส่วน input และ output ของวงจรนี้ Amplifier ทางด้าน input จะช่วยทำให้วงจรมี input impedance สูง ทำให้เก็บประจุได้เร็วขึ้น ส่วน Amplifier ทางด้าน output จะช่วยทำให้มี out impedance ต่ำ สามารถขับ ADC ได้ง่าย ส่วนสำคัญที่ต้องพิจารณาคือ จะต้องใช้ Amplifier ที่กินกระแสต่ำ เพื่อให้ดึงกระแสจากตัวเก็บประจุในช่วง hold สัญญาณน้อยที่สุด มิฉะนั้นจะเกิด การตก (drop) แก่แรงดันที่ hold ดังแสดงในรูปที่ 6.7



รูปที่ 6.7 แสดงการตกของแรงดันที่ hold ไว้

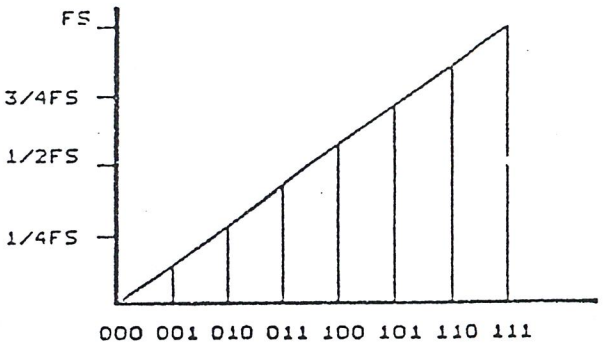


รูปที่ 6.8 แสดงรูปคลื่นสัญญาณจากวงจร sample and hold ในทางอุดมคติ

6.2 วงจรเปลี่ยนสัญญาณ Digital ให้เป็น Analog (DAC)

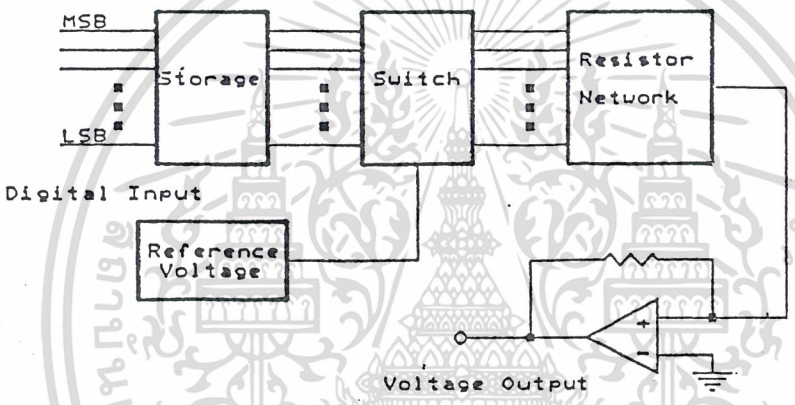
DAC เป็นอุปกรณ์สำคัญอย่างหนึ่งที่ทำให้การนำเอาคอมพิวเตอร์ไปใช้ร่วมกับอุปกรณ์อื่นๆ รูปที่ 6.9 แสดง Transfer function ของDAC ขนาด 3 bit จะเห็นได้ว่า input ที่เป็นรหัส digital 1 word จะเปลี่ยนเป็นแรงดัน analog 1 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



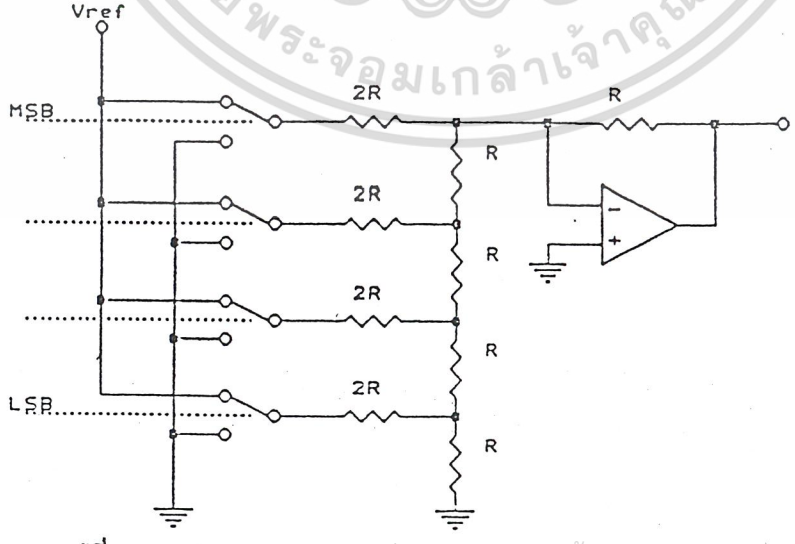
รูปที่ 6.9 แสดงTransfer function ของ DAC ขนาด 3 bit

ลักษณะการจัดวงจร DAC จะเป็นลักษณะดังรูปที่ 6.10



รูปที่ 6.10 แสดง block diagram ของ DAC

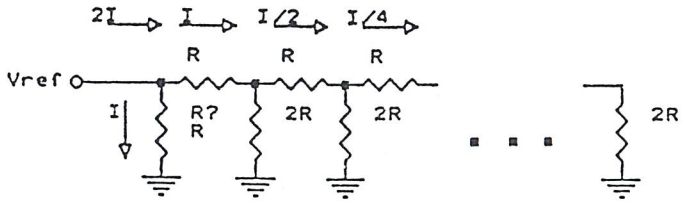
วงจร DAC ที่ใช้กันมีหลายแบบแต่ในที่นี้จะขอกล่าวถึง วงจร DAC ที่เป็นแบบตัวต้านทานขั้นบันได (R-2R ladder) ซึ่งเป็นแบบที่นิยมใช้กันมาก ลักษณะของวงจรดังรูป 6.11



รูปที่ 6.11 แสดงวงจร DAC ขนาด 4 bit แบบ R-2R ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

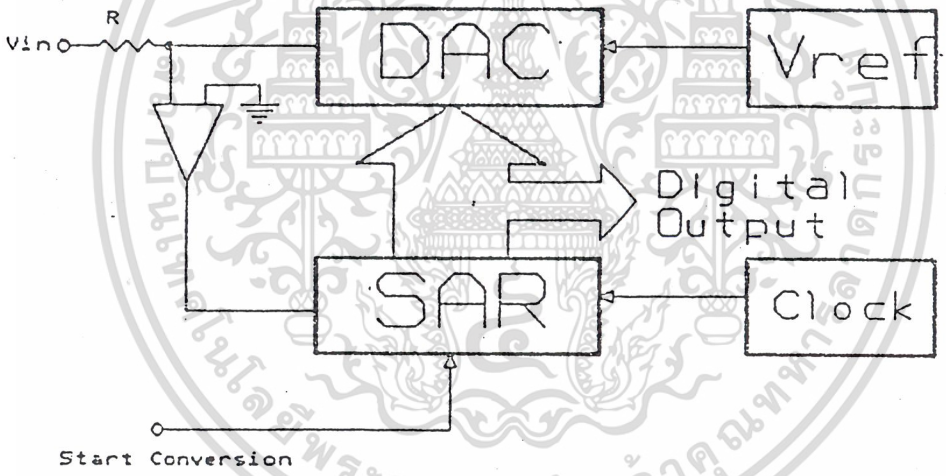
ในวงจรนี้ switch จะตัดต่อให้แรงดันอ้างอิงต่อเข้ากับวงจรขั้นบันได จะเห็นได้ว่า เมื่อมองเข้าไปจะเห็นคู่ของ register ระหว่างจุดต่อ R-2R ที่ติดกัน กระแสจะถูกบั่นทอนไปในอัตรา 2/1 ซึ่งสอดคล้องกับรหัส binary ดังรูปที่ 6.12



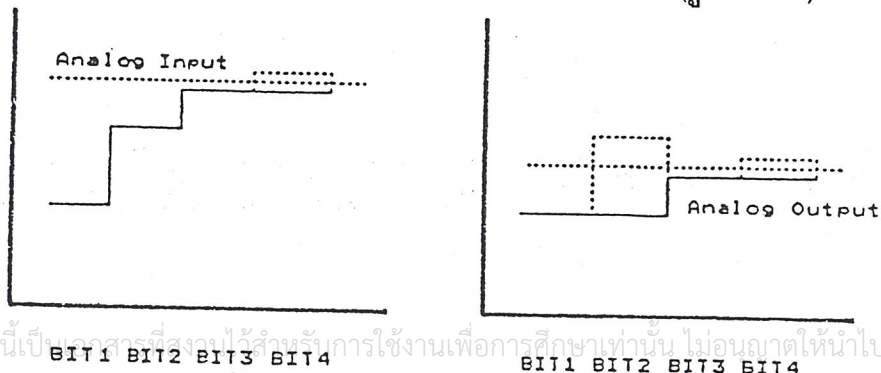
รูปที่ 6.12 แสดงการแบ่งแรงดันจากแรงดันอ้างอิง

6.3 วงจรเปลี่ยนสัญญาณ Analog ให้เป็นสัญญาณ Digital (ADC)

วงจร ADC ส่วนใหญ่จะอยู่ในรูปของ IC และแบบที่นิยมใช้คือ แบบ Successive Approximation ซึ่งนิยมใช้ในงานที่ต้องการความเร็วสูงและปานกลาง วงจรแบบนี้คล้ายกับ counter ที่ทำงานในลักษณะป้อนกลับ



รูปที่ 6.13 แสดง function ต่างๆใน ADC comparator จะคอยเปรียบเทียบ output จาก ADC กับสัญญาณ input ที่เป็น analog ซึ่ง output ที่ได้จะควบคุม register ของตัวประมาณแบบ Successive การทำงานของ SAR จะเป็นดังนี้ (รูปที่ 6.14)



รูป 6.14 timing diagram ของ ADC แบบ SAR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสัญญาณนาฬิกาเข้ามา 1 ลูก จะทำให้ bit สูงสุดเป็น "1" ส่วน bit อื่นๆ ยังคงเป็น "0" DAC จะเปลี่ยน output ของ SAR ให้เป็นสัญญาณ analog เปรียบเทียบกับสัญญาณanalog input ถ้าผลการเปรียบเทียบที่ตัวเปรียบเทียบน้อยกว่าinput ก็คง bit นั้นเป็น "1" แต่ถ้ามากกว่าก็ให้ bit นั้นเป็น "0" จากนั้นจะทำการทดสอบ bit ถัดไป โดยทำให้ bit นั้นเป็น "1" หากผลรวมของสอง bit มีค่ามากกว่าก็ทำให้ bit นั้นเป็น "0" แต่ถ้าน้อยกว่าก็คงค่า "1" เอาไว้แล้วทำการทดสอบ bit ถัดไปด้วยวิธีดังกล่าวจนครบทุกbit หรือจนกว่าoutput จะต่างจากค่าแรงดัน input ไม่เกิน "1" LSB

ข้อจำกัดประการหนึ่งสำหรับการทำงาน คือ analog input จะต้องคงที่ ในช่วงเวลาที่ทำการเปลี่ยนสัญญาณ โดยเปลี่ยนแปลงได้ไม่เกิน ½ LSB ในช่วงสุดท้ายของการเปลี่ยนสัญญาณ digital output จะออกมาขนานกันทุก bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

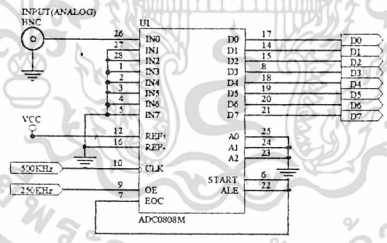
บทที่ 7
การทดลอง และ ผลการทดลอง

การทดลองที่ 7.1 การทดลองภาค A/D Converter

วัตถุประสงค์ เพื่อทำการทดสอบ การทำงานของภาค A/D Converter ว่าสามารถแปลงจากสัญญาณ Analog เป็น Digital ได้หรือไม่

ขั้นตอนการทดลอง

1. ป้อนสัญญาณ Analog มีค่า 0.5V เข้า Input ของ ไอซี เบอร์ ADC0808 ดังรูป
2. ป้อนความถี่ให้แก่ไอซี เพื่อการ Sampling จะใช้ความถี่ fs < 640 KHz และในการทดลองเราจะใช้ความถี่ Sampling = 500 KHz
3. ใช้แรงดันอ้างอิงที่ REF+ = +5V, REF- = 0V
4. วัด O/P ที่ได้ทางทางขา Data (D0 – D7) ซึ่งเป็น Binary 8 bit



รูปที่ 7.1 วงจรภาค A/D Converter

การทำงานของวงจรภาคอินพุทแบบอนาลอก

ภาคอินพุทแบบอนาลอกนี้ใช้ไอซี ADC (ANALOG TO DIGITAL INVERTER)ทำหน้าที่เปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณทางดิจิตอลซึ่งสามาถเลือกใช้ได้ 2 ขนาด คือ ADC0809 (ADC ขนาด 8 บิท) ซึ่งไอซี ADC นี้สามารถรับสัญญาณอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบอนาลอกได้โดยตรงสูงถึง +5V. หรือหากต้องการให้รับค่าอินพุทได้สูงกว่า +5V. แล้วก็สามารถทำได้ โดยการเพิ่มวงจรขยายย่านวัดเข้าไปทางภาคหน้าของวงจรได้

หลักการทำงานของวงจร

ใช้ ADC จะทำหน้าที่เปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล โดยใช้หลักการสุ่มรับสัญญาณอนาลอกทางอินพุท (SAMPLING) แล้วนำขนาดของวสัญญาณอนาลอกนั้น มาเปรียบเทียบกับค่าแรงดันอ้างอิง ของวงจรแล้วจึงเปลี่ยนเป็นค่าข้อมูลแบบดิจิตอลซึ่งความเร็วในการสุ่มขนาดสัญญาณ (SAMPLING RATE) สามารถกำหนดได้จากความถี่ของสัญญาณนาฬิกา ของวงจร

ความสัมพันธ์ของอินพุทและเอาท์พุทของไอซี ADC

ดังได้กล่าวมาแล้วว่าไอซี ADC จะใช้การเปรียบเทียบขนาดของสัญญาณอินพุท กับแรงดันอ้างอิงของวงจรแล้วจึงเปลี่ยนเป็นข้อมูลแบบดิจิตอล ซึ่งมีความสัมพันธ์ดังนี้คือ

- ถ้าค่าของขนาดสัญญาณอนาลอกอินพุทเท่ากับขนาดของแรงดันอ้างอิงจะได้ข้อมูลมีค่าเป็นครึ่งหนึ่งของข้อมูลสูงสุด
- ถ้าค่าของขนาดสัญญาณอินพุทของวงจรถูกกำหนดโดยค่าตัวแปรต่างๆดังนี้

1. **แรงดันอ้างงอิงของวงจร** (Voltage Reference หรือ Vref) แรงดันอ้างอิงของวงจรนี้เป็นจุดอ้างอิงมาตรฐานของวงจรเป็นตัวกำหนดความเที่ยงตรงในการเปลี่ยนขนาดของสัญญาณอนาลอกให้เป็นสัญญาณทางดิจิตอล ซึ่งวงจรต้องการแรงดันอ้างอิงที่ความเที่ยงตรงสูง เพราะหากแรงดันอ้างอิงของวงจรไม่มีความเที่ยงตรงแน่นอน มีการเปลี่ยนแปลงแล้ว การทำงานของวงจรก็จะเปลี่ยนแปลงตามด้วย นั้นก็คือจะส่งผลให้ค่าของสัญญาณข้อมูลดิจิตอลที่ได้ก็จะเปลี่ย่นแปลงขึ้น ๆ ลง ๆ ตามไปด้วย ถึงแม่ว่าสัญญาณอินพุทจะมีขนาดคงที่ก็ตาม และแรงดันอ้างอิงของวงจรนี้ยังเป็นตัวกำหนดค่าสูงสุดของสัญญาณอินพุทบวกของวงจร โดยค่าสัญญาณอินพุทบวกของวงจรจะถูกกำหนดให้มีค่าสูงสุด เท่ากับแรงดันบวกอ้างอิง(Vref+) โดยมีจุดเริ่มต้นที่สัญญาณอินพุทเท่ากับแรงดันอ้างอิงลบ(Vref-) ของวงจรเสมอ ค่าแรงดันอ้างอิงที่ทำให้ไอซี ADC รับสัญญาณอินพุทได้โดยตรงสูงสุด +5.0 V. แต่ผู้ใช้สามารถเลือกใช้แรงดันอ้างอิงให้กับวงจร.
2. **สัญญาณอินพุทบวก** (Voltage Input หรือ VIN) แรงดันอินพุทบวกนี้ ก็คือค่าแรงดันอินพุทของวงจรนั่นเองซึ่งต้องกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้มีขนาดสูงสุดไม่เกินกว่าค่าที่วงจร ADC รับได้ ซึ่งต้องเป็นค่าแรงดันที่มีค่าอยู่ในช่วงบวกด้วย เนื่องจากไอซี ADC ไม่สามารถรับสัญญาณอินพุทที่มีค่าเป็นลบได้ สัญญาณอินพุทนี้จะมีค่าสูงสุด เท่ากับแรงดันอ้างอิงบวกโดยมีจุดเริ่มต้นที่สัญญาณอินพุทของวงจรและค่าสูงสุดของสัญญาณอินพุทบวกนี้ต้องมีค่าต่ำกว่า +5.0V ด้วยเสมอ

ตัวอย่าง

เราใช้ไอซี ADC ขนาด 8 บิท (ADC0809)

ให้แรงดันอ้างอิงบวกของวงจร (Vref+) มีค่าเป็น 5.00 V.

ให้แรงดันอ้างอิงลบของวงจร (Vref-) มีค่าเป็น 0.00 V. จะได้ว่า

_ แรงดันอินพุทบวกของวงจร (VIN) ต้องมีค่าอยู่ระหว่าง 0.00 V. - 5.00 V.

_ ถ้าขนาดของสัญญาณอนาลอกอินพุทมีค่าเป็น 0.00 V. จะได้ค่าของข้อมูลเอาท์พุทเป็น 00H

_ ถ้าขนาดของสัญญาณอนาลอกอินพุทมีค่าเป็น 2.50 V. จะได้ค่าของข้อมูลเอาท์พุทเป็น 7FH

_ถ้าขนาดของสัญญาณอนาลอกอินพุทมีค่าเป็น 5.00 V. จะได้ค่าของข้อมูลเทาท์พุทเป็น FFH

ตัวอย่าง

เราใช้ไอซี ADC ขนาด 8 บิท (ADC0809)

ให้แรงดันอ้างอิงบวกของวงจร (Vref+) มีค่าเป็น 3.50 V.

ให้แรงดันอ้างอิงลบของวงจร (Vref-) มีค่าเป็น 0.50 V. จะได้ว่า

_ แรงดันอินพุทบวกของวงจร (VIN) ต้องมีค่าระหว่าง 0.50 V. – 3.50 V.

_ ถ้าขนาดของสัญญาณอนาลอกอินพุทเป็น 0.5V จะได้ค่าของข้อมูลเอาท์พุทเป็น 00H

_ ถ้าขนาดของสัญญาณอนาลอกอินพุทเป็น 2.0V จะได้ค่าของข้อมูลเอาท์พุทเป็น 7FH

_ ถ้าขนาดของสัญญาณอนาลอกอินพุทเป็น 3.5V จะได้ค่าของข้อมูลเอาท์พุทเป็น FFH

ผลการทดลอง ภาค A/D Converter

แรงดัน (V)	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0
0.02	0	0	0	0	0	0	0	1
0.04	0	0	0	0	0	0	1	0
0.06	0	0	0	0	0	0	1	1
0.08	0	0	0	0	0	1	0	0
0.1	0	0	0	0	0	1	0	1
0.12	0	0	0	0	0	1	1	0
0.14	0	0	0	0	0	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.16	0	0	0	0	1	0	0	0
0.18	0	0	0	0	1	0	0	1
0.2	0	0	0	0	1	0	1	0
0.22	0	0	0	0	1	0	1	1
0.24	0	0	0	0	1	1	0	0
0.26	0	0	0	0	1	1	0	1
0.28	0	0	0	0	1	1	1	0
0.3	0	0	0	0	1	1	1	1
0.32	0	0	0	1	0	0	0	0
0.34	0	0	0	1	0	0	0	1
0.36	0	0	0	1	0	0	1	0
0.38	0	0	0	1	0	0	1	1
0.4	0	0	0	1	0	1	0	0
0.41	0	0	0	1	0	1	0	1
0.43	0	0	0	1	0	1	1	0
0.45	0	0	0	1	0	1	1	0
0.47	0	0	0	1	1	0	0	0
0.49	0	0	0	1	1	0	0	1
0.51	0	0	0	1	1	0	1	0
0.53	0	0	0	1	1	0	1	1
0.55	0	0	0	1	1	1	0	0
0.57	0	0	0	1	1	1	0	1
0.59	0	0	0	1	1	1	1	0
0.61	0	0	0	1	1	1	1	1
0.63	0	0	1	0	0	0	0	0
0.65	0	0	1	0	0	0	0	1
0.66	0	0	1	0	0	0	1	0
0.68	0	0	1	0	0	0	1	1
0.7	0	0	1	0	0	1	0	0
0.72	0	0	1	0	0	1	0	1
0.74	0	0	1	0	0	1	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.76	0	0	1	0	0	1	1	1
0.78	0	0	1	0	1	0	0	0
0.8	0	0	1	0	1	0	0	1
0.82	0	0	1	0	1	0	1	0
0.84	0	0	1	0	1	0	1	1
0.86	0	0	1	0	1	1	0	0
0.88	0	0	1	0	1	1	0	1
0.9	0	0	1	0	1	1	1	1
0.94	0	0	1	1	0	0	0	0
0.96	0	0	1	1	0	0	0	1
0.98	0	0	1	1	0	0	1	0
1	0	0	1	1	0	0	1	1
1.02	0	0	1	1	0	1	0	0
1.04	0	0	1	1	0	1	0	1
1.06	0	0	1	1	0	1	1	0
1.08	0	0	1	1	0	1	1	1
1.09	0	0	1	1	1	0	0	0
1.11	0	0	1	1	1	0	0	1
1.13	0	0	1	1	1	0	1	0
1.15	0	0	1	1	1	0	1	1
1.17	0	0	1	1	1	1	0	0
1.19	0	0	1	1	1	1	0	1
1.21	0	0	1	1	1	1	1	0
1.23	0	0	1	1	1	1	1	1
1.25	0	1	0	0	0	0	0	0
1.27	0	1	0	0	0	0	0	1
1.29	0	1	0	0	0	0	1	0
1.31	0	1	0	0	0	0	1	1
1.33	0	1	0	0	0	1	0	0
1.35	0	1	0	0	0	1	0	1
1.37	0	1	0	0	0	1	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.39	0	1	0	0	0	1	1	1
1.41	0	1	0	0	1	0	0	0
1.43	0	1	0	0	1	0	0	1
1.45	0	1	0	0	1	0	1	0
1.46	0	1	0	0	1	0	1	1
1.48	0	1	0	0	1	1	0	0
1.50	0	1	0	0	1	1	0	1
1.52	0	1	0	0	1	1	1	0
1.54	0	1	0	0	1	1	1	1
1.56	0	1	0	1	0	0	0	0
1.58	0	1	0	1	0	0	0	1
1.6	0	1	0	1	0	0	1	0
1.62	0	1	0	1	0	0	1	1
1.64	0	1	0	1	0	1	0	0
1.66	0	1	0	1	0	1	0	1
1.68	0	1	0	1	0	1	1	0
1.7	0	1	0	1	0	1	1	1
1.72	0	1	0	1	1	0	0	0
1.74	0	1	0	1	1	0	0	1
1.76	0	1	0	1	1	0	1	0
1.78	0	1	0	1	1	0	1	1
1.8	0	1	0	1	1	1	0	0
1.82	0	1	0	1	1	1	0	1
1.84	0	1	0	1	1	1	1	0
1.86	0	1	0	1	1	1	1	1
1.88	0	1	1	0	0	0	0	0
1.9	0	1	1	0	0	0	0	1
1.92	0	1	1	0	0	0	1	0
1.93	0	1	1	0	0	0	1	1
1.95	0	1	1	0	0	1	0	0
1.97	0	1	1	0	0	1	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.99	0	1	1	0	0	1	1	0
2.01	0	1	1	0	0	1	1	1
2.03	0	1	1	0	1	0	0	0
2.05	0	1	1	0	1	0	0	1
2.07	0	1	1	0	1	0	1	0
2.09	0	1	1	0	1	0	1	1
2.11	0	1	1	0	1	1	0	0
2.13	0	1	1	0	1	1	0	1
2.15	0	1	1	0	1	1	1	0
2.17	0	1	1	0	1	1	1	1
2.19	0	1	1	1	0	0	0	0
2.21	0	1	1	1	0	0	0	1
2.23	0	1	1	1	0	0	1	0
2.25	0	1	1	1	0	0	1	1
2.27	0	1	1	1	0	1	0	0
2.29	0	1	1	1	0	1	0	1
2.31	0	1	1	1	0	1	1	0
2.33	0	1	1	1	0	1	1	1
2.35	0	1	1	1	1	0	0	0
2.36	0	1	1	1	1	0	0	1
2.38	0	1	1	1	1	0	1	0
2.4	0	1	1	1	1	0	1	1
2.42	0	1	1	1	1	1	0	0
2.44	0	1	1	1	1	1	0	1
2.46	0	1	1	1	1	1	1	0
2.48	0	1	1	1	1	1	1	1
2.5	1	0	0	0	0	0	0	0
2.52	1	0	0	0	0	0	0	1
2.54	1	0	0	0	0	0	1	0
2.56	1	0	0	0	0	0	1	1
2.58	1	0	0	0	0	1	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6	1	0	0	0	0	1	0	1
2.62	1	0	0	0	0	1	1	0
2.64	1	0	0	0	0	1	1	1
2.66	1	0	0	0	1	0	0	0
2.68	1	0	0	0	1	0	0	1
2.7	1	0	0	0	1	0	1	0
2.72	1	0	0	0	1	0	1	1
2.74	1	0	0	0	1	1	0	0
2.76	1	0	0	0	1	1	0	1
2.77	1	0	0	0	1	1	1	0
2.79	1	0	0	0	1	1	1	1
2.81	1	0	0	1	0	0	0	0
2.83	1	0	0	1	0	0	0	1
2.85	1	0	0	1	0	0	1	0
2.87	1	0	0	1	0	0	1	1
2.89	1	0	0	1	0	1	0	0
2.91	1	0	0	1	0	1	0	1
2.93	1	0	0	1	0	1	1	0
2.95	1	0	0	1	0	1	1	1
2.97	1	0	0	1	1	0	0	0
2.99	1	0	0	1	1	0	0	1
3.01	1	0	0	1	1	0	1	0
3.03	1	0	0	1	1	0	1	1
3.05	1	0	0	1	1	1	0	0
3.07	1	0	0	1	1	1	0	1
3.09	1	0	0	1	1	1	1	0
3.11	1	0	0	1	1	1	1	1
3.13	1	0	1	0	0	0	0	0
3.15	1	0	1	0	0	0	0	1
3.16	1	0	1	0	0	0	1	0
3.18	1	0	1	0	0	0	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2	1	0	1	0	0	1	0	0
3.22	1	0	1	0	0	1	0	1
3.24	1	0	1	0	0	1	1	0
3.26	1	0	1	0	0	1	1	1
3.28	1	0	1	0	1	0	0	0
3.3	1	0	1	0	1	0	0	1
3.32	1	0	1	0	1	0	1	0
3.34	1	0	1	0	1	0	1	1
3.36	1	0	1	0	1	1	0	0
3.38	1	0	1	0	1	1	0	1
3.4	1	0	1	0	1	1	1	0
3.42	1	0	1	0	1	1	1	1
3.44	1	0	1	1	0	0	0	0
3.48	1	0	1	1	0	0	0	1
3.5	1	0	1	1	0	0	1	0
3.52	1	0	1	1	0	0	1	1
3.54	1	0	1	1	0	1	0	0
3.56	1	0	1	1	0	1	0	1
3.58	1	0	1	1	0	1	1	0
3.6	1	0	1	1	0	1	1	1
3.61	1	0	1	1	1	0	0	0
3.63	1	0	1	1	1	0	0	1
3.65	1	0	1	1	1	0	1	0
3.67	1	0	1	1	1	0	1	1
3.69	1	0	1	1	1	1	0	0
3.71	1	0	1	1	1	1	0	1
3.73	1	0	1	1	1	1	1	0
3.75	1	0	1	1	1	1	1	1
3.77	1	1	0	0	0	0	0	0
3.79	1	1	0	0	0	0	0	1
3.81	1	1	0	0	0	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.83	1	1	0	0	0	0	1	1
3.85	1	1	0	0	0	1	0	0
3.87	1	1	0	0	0	1	0	1
3.89	1	1	0	0	0	1	1	0
3.91	1	1	0	0	0	1	1	1
3.93	1	1	0	0	1	0	0	0
3.95	1	1	0	0	1	0	0	1
3.97	1	1	0	0	1	0	1	0
3.99	1	1	0	0	1	0	1	1
4.0	1	1	0	0	1	1	0	0
4.02	1	1	0	0	1	1	0	1
4.04	1	1	0	0	1	1	1	0
4.06	1	1	0	0	1	1	1	1
4.08	1	1	0	1	0	0	0	0
4.1	1	1	0	1	0	0	0	1
4.12	1	1	0	1	0	0	1	0
4.14	1	1	0	1	0	0	1	1
4.16	1	1	0	1	0	1	0	0
4.18	1	1	0	1	0	1	0	1
4.2	1	1	0	1	0	1	1	0
4.22	1	1	0	1	0	1	1	1
4.24	1	1	0	1	1	0	0	0
4.26	1	1	0	1	1	0	0	1
4.28	1	1	0	1	1	0	1	0
4.3	1	1	0	1	1	0	1	1
4.32	1	1	0	1	1	1	0	0
4.34	1	1	0	1	1	1	0	1
4.36	1	1	0	1	1	1	1	0
4.38	1	1	0	1	1	1	1	1
4.4	1	1	1	0	0	0	0	0
4.42	1	1	1	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.43	1	1	1	0	0	0	1	0
4.45	1	1	1	0	0	0	1	1
4.47	1	1	1	0	0	1	0	0
4.49	1	1	1	0	0	1	0	1
4.51	1	1	1	0	0	1	1	0
4.53	1	1	1	0	0	1	1	1
4.55	1	1	1	0	1	0	0	0
4.57	1	1	1	0	1	0	0	1
4.59	1	1	1	0	1	0	1	0
4.61	1	1	1	0	1	0	1	1
4.63	1	1	1	0	1	1	0	0
4.65	1	1	1	0	1	1	0	1
4.67	1	1	1	0	1	1	1	0
4.69	1	1	1	0	1	1	1	1
4.71	1	1	1	1	0	0	0	0
4.73	1	1	1	1	0	0	0	1
4.75	1	1	1	1	0	0	1	0
4.77	1	1	1	1	0	0	1	1
4.79	1	1	1	1	0	1	0	0
4.8	1	1	1	1	0	1	0	1
4.82	1	1	1	1	0	1	1	0
4.84	1	1	1	1	0	1	1	1
4.86	1	1	1	1	1	0	0	0
4.88	1	1	1	1	1	0	0	1
4.9	1	1	1	1	1	0	1	0
4.92	1	1	1	1	1	0	1	1
4.94	1	1	1	1	1	1	0	0
4.96	1	1	1	1	1	1	0	1
4.98	1	1	1	1	1	1	1	0
5.0	1	1	1	1	1	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลอง หน้าที่ของภาค A/D คือ การแปลงสัญญาณจากสัญญาณ Analog เป็นสัญญาณ Digital 8 bit โดยที่มีสัญญาณ Analog มีค่า 0 – 5 V ซึ่งจากการทดลอง ค่า 0V มีค่า 00000000 ค่า 2 – 5V มีค่า 1000 0000 ซึ่งเป็นค่ากลาง และค่า 5V จะเป็น 11 11 11 11 การปรับสเกลของ Analog จะทำให้ได้ Binary 8 bit มีค่าปรับขึ้นไปเรื่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

สัญญาณปัจจุบัน	Shift ครั้งที่ 1	Shift ครั้งที่ 2
00000000	xxxxxxxx	xxxxxxxx
00000001	00000000	xxxxxxxx
00000010	00000001	00000000
00000111	00000010	00000001
00000100	00000011	00000010
00000101	00000100	00000011
00000110	00000101	00000100
00000111	00000110	00000101
00001000	00000111	00000110

สรุปผลการทดลอง

Shift Register จะทำหน้าที่ Delay สัญญาณที่เป็น Digital ที่มาจาก O/P ของภาค A/D Converter โดยจะใช้สัญญาณนาฬิกาที่ความถี่เดียวกับภาค A/D Converter จากรูปของวงจรภาค Shift Register จะมีการ Delay ดังนี้คือ

1. สัญญาณ O/P ของไอซี 74LS374 ตัวที่ 2 จะเป็นสัญญาณอดีตของสัญญาณ O/P ของไอซี 74LS374 ตัวที่ 1
2. สัญญาณ O/P ของไอซี 74LS374 ตัวที่ 1 จะเป็นสัญญาณอดีตของสัญญาณ I/P ของไอซี 74LS374 ตัวที่ 1
3. สัญญาณ I/P ของไอซี 74LS374 ตัวที่ 1 จะเป็นสัญญาณปัจจุบัน

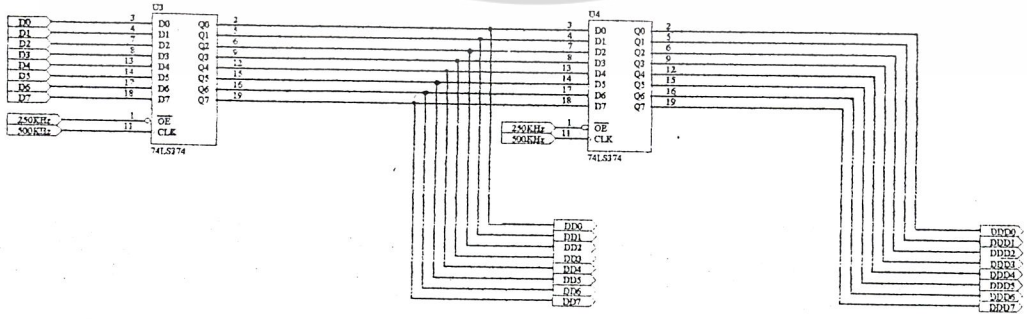
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7.2 การทดลองภาค SHIFT REGISTER (PIPO)

วัตถุประสงค์ เพื่อทดสอบ การทำงานของภาค Shift Register ว่าสามารถ Delay ข้อมูลที่มาจากภาค A/D Converter ได้จริงหรือไม่

ขั้นตอนการทดลอง

1. ต่อสัญญาณที่มาจาก O/P ของภาค A/D Converter (D0 –D7) จะส่งมาที่ไอซีเบอร์ 74LS374 ตัวที่ 1
2. ต่อสัญญาณที่มาจาก O/P ของไอซีเบอร์ 74LS374 ตัวที่ 1 จะเป็น I/P ของไอซี 74LS374 ตัวที่ 2 ซึ่งการต่อของไอซีทั้ง 2 ตัว จะเป็นแบบ Case Cade กัน
3. ต่อสัญญาณ O/P ของไอซี 74LS374 ของตัวที่ 1 จะเป็นการ Shift ครั้งที่ 1
4. ต่อสัญญาณ O/P ของไอซี 74LS374 ของตัวที่ 2 จะเป็นการ Shift ครั้งที่ 2
5. ต่อสัญญาณ I/P และ O/P ของไอซี 74LS374 ตัวที่ 1 และสัญญาณ O/P ของไอซี 74LS374 ตัวที่ 2 ซึ่งรวมทั้งหมด 3 สัญญาณ โดยมีสัญญาณละ 8 bit จะถูกส่งไปยัง EPROM เพื่อไปเป็น Address แต่ละชุด
6. วัดผลของการ Shift แต่ละตัว



รูปที่ 7.2 วงจรภาค Shift Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

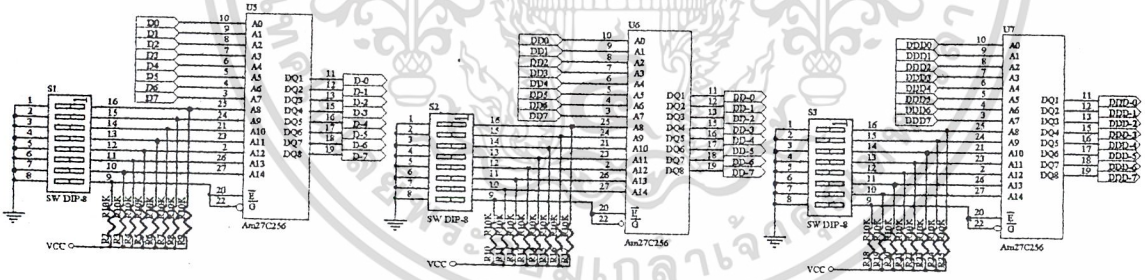
การทดลองที่7.3 การทดลองภาค LOOK UP TABLE

วัตถุประสงค์

1. เพื่อทำการทดลอง การทำงานของภาค LOOK UP TABLE ว่าสามารถเก็บข้อมูลได้จริงหรือไม่
2. เพื่อทำการทดลอง การทำงานของภาค LOOK UP TABLE ว่าสามารถเรียกผลลัพธ์ได้จริงหรือไม่

ขั้นตอนการทดลอง

1. ต่อสัญญาณ O/P ของภาค Shift Register 3 สัญญาณ เข้ามาเข้ามาเป็น I/P ของ AM27C256 ของทั้ง 3 ตัว
2. ทำการ Program ข้อมูลของสัมประสิทธิ์ลงใน Eprom ทั้ง 3 ตัว ซึ่งการคำรวณหา Program จะใช้ MatLap ช่วย
3. ใช้ DIP SW เป็นตัวเลือกสัมประสิทธิ์ที่จะคูณกับข้อมูลที่เป็น I/P ของภาคนี้
4. วัดผลที่ได้ทาง O/P



รูปที่ 7.3 วงจรภาค Look – Up Table (Rom Accumulator)

ผลการทดลอง

เขียนการใช้ MatLab คำนวณหา Program ใน Eprom และ แสดง Program

ชุดคำสั่งของ Mathlab กรณีค่าสัมประสิทธิ์ h(n) = 1.0;

```
S=1.0          ;      A=D*0.01953125;
D=0:1:255      ;      B=S*A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ได้จากการคำนวณที่ h(n) = 1.0;

B =

Columns 1 through 7

0 0.0195 0.0391 0.0586 0.0781 0.0977 0.1172

Columns 8 through 14

0.1367 0.1562 0.1758 0.1953 0.2148 0.2344 0.2539

Columns 15 through 21

0.2734 0.2930 0.3125 0.3320 0.3516 0.3711 0.3906

Columns 22 through 28

0.4102 0.4297 0.4492 0.4688 0.4883 0.5078 0.5273

Columns 29 through 35

0.5469 0.5664 0.5859 0.6055 0.6250 0.6445 0.6641

Columns 36 through 42

0.6836 0.7031 0.7227 0.7422 0.7617 0.7812 0.8008

Columns 43 through 49

0.8203 0.8398 0.8594 0.8789 0.8984 0.9180 0.9375

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Columns 50 through 56

0.9570 0.9766 0.9961 1.0156 1.0352 1.0547 1.0742

Columns 57 through 63

1.0938 1.1133 1.1328 1.1523 1.1719 1.1914 1.2109

Columns 64 through 70

1.2305 1.2500 1.2695 1.2891 1.3086 1.3281 1.3477

Columns 71 through 77

1.3672 1.3867 1.4062 1.4258 1.4453 1.4648 1.4844

Columns 78 through 84

1.5039 1.5234 1.5430 1.5625 1.5820 1.6016 1.6211

Columns 85 through 91

1.6406 1.6602 1.6797 1.6992 1.7188 1.7383 1.7578

Columns 92 through 98

1.7773 1.7969 1.8164 1.8359 1.8555 1.8750 1.8945

Columns 99 through 105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.9141 1.9336 1.9531 1.9727 1.9922 2.0117 2.0312

Columns 106 through 112

2.0508 2.0703 2.0898 2.1094 2.1289 2.1484 2.1680

Columns 113 through 119

2.1875 2.2070 2.2266 2.2461 2.2656 2.2852 2.3047

Columns 120 through 126

2.3242 2.3438 2.3633 2.3828 2.4023 2.4219 2.4414

Columns 127 through 133

2.4609 2.4805 2.5000 2.5195 2.5391 2.5586 2.5781

Columns 134 through 140

2.5977 2.6172 2.6367 2.6562 2.6758 2.6953 2.7148

Columns 141 through 147

2.7344 2.7539 2.7734 2.7930 2.8125 2.8320 2.8516

Columns 148 through 154

2.8711 2.8906 2.9102 2.9297 2.9492 2.9688 2.9883

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Columns 155 through 161

3.0078 3.0273 3.0469 3.0664 3.0859 3.1055 3.1250

Columns 162 through 168

3.1445 3.1641 3.1836 3.2031 3.2227 3.2422 3.2617

Columns 169 through 175

3.2812 3.3008 3.3203 3.3398 3.3594 3.3789 3.3984

Columns 176 through 182

3.4180 3.4375 3.4570 3.4766 3.4961 3.5156 3.5352

Columns 183 through 189

3.5547 3.5742 3.5938 3.6133 3.6328 3.6523 3.6719

Columns 190 through 196

3.6914 3.7109 3.7305 3.7500 3.7695 3.7891 3.8086

Columns 197 through 203

3.8281 3.8477 3.8672 3.8867 3.9062 3.9258 3.9453

Columns 204 through 210

3.9648 3.9844 4.0039 4.0234 4.0430 4.0625 4.0820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Columns 211 through 217

4.1016 4.1211 4.1406 4.1602 4.1797 4.1992 4.2188

Columns 218 through 224

4.2383 4.2578 4.2773 4.2969 4.3164 4.3359 4.3555

Columns 225 through 231

4.3750 4.3945 4.4141 4.4336 4.4531 4.4727 4.4922

Columns 232 through 238

4.5117 4.5312 4.5508 4.5703 4.5898 4.6094 4.6289

Columns 239 through 245

4.6484 4.6680 4.6875 4.7070 4.7266 4.7461 4.7656

Columns 246 through 252

4.7852 4.8047 4.8242 4.8438 4.8633 4.8828 4.9023

Columns 253 through 256

4.9219 4.9414 4.9609 4.9805

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่าผลลัพธ์ที่ได้จากโปรแกรม Matlab มาแปลงเป็นเลขฐาน $(\ )_{16}$ โดยเทียบจากตารางดังนี้

แรงดัน (V)	เลขฐาน $(\ )_{16}$ ที่โปรแกรมใน Eprom
0	00H
0.02	01H
0.04	02H
0.06	03H
0.08	04H
0.1	05H
0.12	06H
0.14	07H
0.16	08H
0.18	09H
0.2	0AH
0.22	0BH
0.24	0CH
0.26	0DH
0.28	0EH
0.3	0FH
0.32	10H
0.34	11H
0.36	12H
0.38	13H
0.4	14H
0.41	15H
0.43	16H
0.45	17H
0.47	18H
0.49	19H
0.51	1AH
0.53	1BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.55	1CH
0.57	1DH
0.59	1EH
0.61	1FH
0.63	20H
0.65	21H
0.66	22H
0.68	23H
0.7	24H
0.72	25H
0.74	26H
0.76	27H
0.78	28H
0.8	29H
0.82	2AH
0.84	2BH
0.86	2CH
0.88	2DH
0.9	2EH
0.94	2FH
0.96	30H
0.98	31H
1	32H
1.02	33H
1.04	34H
1.06	35H
1.08	36H
1.09	37H
1.11	38H
1.13	39H
1.15	3AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.17	3BH
1.19	3CH
1.21	3DH
1.23	3EH
1.25	3FH
1.27	40H
1.29	41H
1.31	42H
1.33	43H
1.35	44H
1.37	45H
1.39	46H
1.41	47H
1.43	48H
1.45	49H
1.46	4AH
1.48	4BH
1.50	4CH
1.52	4DH
1.54	4EH
1.56	4FH
1.58	50H
1.6	51H
1.62	52H
1.64	53H
1.66	54H
1.68	55H
1.7	56H
1.72	57H
1.74	58H
1.76	59H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.78	5AH
1.8	5BH
1.82	5CH
1.84	5DH
1.86	5EH
1.88	5FH
1.9	60H
1.92	61H
1.93	62H
1.95	63H
1.97	64H
1.99	65H
2.01	66H
2.03	67H
2.05	68H
2.07	69H
2.09	6AH
2.11	6BH
2.13	6CH
2.15	6DH
2.17	6EH
2.19	6FH
2.21	70H
2.23	71H
2.25	72H
2.27	73H
2.29	74H
2.31	75H
2.33	76H
2.35	77H
2.36	78H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.38	79H
2.4	7AH
2.42	7BH
2.44	7CH
2.46	7DH
2.48	7EH
2.5	7FH
2.52	80H
2.54	81H
2.56	82H
2.58	83H
2.6	84H
2.62	85H
2.64	86H
2.66	87H
2.68	88H
2.7	89H
2.72	8AH
2.74	8BH
2.76	8CH
2.77	8DH
2.79	8EH
2.81	8FH
2.83	90H
2.85	91H
2.87	92H
2.89	93H
2.91	94H
2.93	95H
2.95	96H
2.97	97H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.99	98H
3.01	99H
3.03	9AH
3.05	9BH
3.07	9CH
3.09	9DH
3.11	9EH
3.13	9FH
3.15	A0H
3.16	A1H
3.18	A2H
3.2	A3H
3.22	A4H
3.24	A5H
3.26	A6H
3.28	A7H
3.3	A8H
3.32	A9H
3.34	AAH
3.36	ABH
3.38	ACH
3.4	ADH
3.42	AEH
3.44	AFH
3.48	B0H
3.5	B1H
3.52	B2H
3.54	B3H
3.56	B4H
3.58	B5H
3.6	B6H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.61	B7H
3.63	B8H
3.65	B9H
3.67	BAH
3.69	BBH
3.71	BCH
3.73	BDH
3.75	BEH
3.77	BFH
3.79	C0H
3.81	C1H
3.83	C2H
3.85	C3H
3.87	C4H
3.89	C5H
3.91	C6H
3.93	C7H
3.95	C8H
3.97	C9H
3.99	CAH
4.0	CBH
4.02	CCH
4.04	CDH
4.06	CEH
4.08	CFH
4.1	D0H
4.12	D1H
4.14	D2H
4.16	D3H
4.18	D4H
4.2	D5H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.22	D6H
4.24	D7H
4.26	D8H
4.28	D9H
4.3	DAH
4.32	DBH
4.34	DCH
4.36	DDH
4.38	DEH
4.4	DFH
4.42	E0H
4.43	E1H
4.45	E2H
4.47	E3H
4.49	E4H
4.51	E5H
4.53	E6H
4.55	E7H
4.57	E8H
4.59	E9H
4.61	EAH
4.63	EBH
4.65	ECH
4.67	EDH
4.69	EEH
4.71	EFH
4.73	F0H
4.75	F1H
4.77	F2H
4.79	F3H
4.8	F4H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.82	F5H
4.84	F6H
4.86	F7H
4.88	F8H
4.9	F9H
4.92	FAH
4.94	FBH
4.96	FCH
4.98	FDH
5.0	FEH

เมื่อได้ค่าจากโปรแกรม Matlab ก็นำค่าระดับในแต่ละ step โดยแต่ละ step มีค่า 0.2 V นำมาเทียบกับค่าที่ได้จากตารางด้านบนเพื่อดูว่าได้ค่าเลขฐานสิบหกค่าใดแล้วนำค่านั้นไปโปรแกรมใน Eprom โดยการ copy โปรแกรมลง Eprom นี้เราใช้โปรแกรม SunShine Eprom Programmer V7.0 เป็นการโปรแกรมลง Eprom ใช้งานในวงจรนี้

<u>การทำงานของวงจร</u>

ไอซี AM27C256 จะทำหน้าที่เก็บข้อมูลของสัมประสิทธิ์ที่จะนำมาคูณกับ I/P ที่ Delay ซึ่ง O/P จะออกมาได้โดยการมี Address ซึ่งเป็น I/P ของภาคนี้

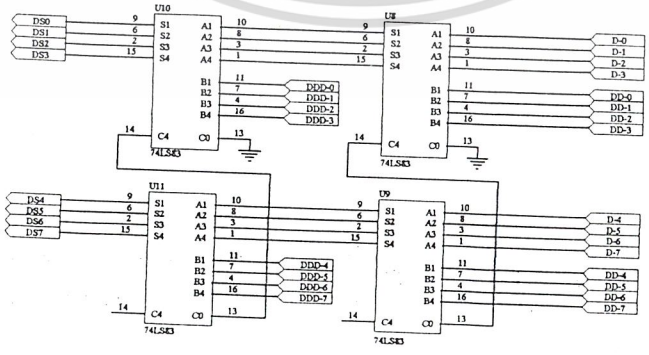
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7.4 การทดลองภาค Accumulator

วัตถุประสงค์ เพื่อทำการทดลองการทำงานของภาค Accumulator ว่าสามารถบวกข้อมูลได้จริงหรือไม่

ขั้นตอนการทดลอง

1. ต่อสัญญาณ O/P ของภาค LOOK UP TABLE สัญญาณที่ที่ Delay ไปครั้งที่ 1 ครึ่งหนึ่งบวกกับสัญญาณปัจจุบันครึ่งหนึ่งด้วยไอซีเบอร์ 74LS83 ตัวที่ 1
2. ต่อสัญญาณ O/P ของภาค LOOK UP TABLE สัญญาณที่ที่ Delay ไปครั้งที่ 1 ครึ่งที่เหลือบวกกับสัญญาณปัจจุบันครึ่งที่เหลือ ด้วยไอซีเบอร์ 74LS83 ตัวที่ 2
3. ต่อ O/P ของไอซีเบอร์ 74LS83 ตัวที่ 1 จะนำไปบวกกับสัญญาณที่ Delay ไปครั้งที่ 2 ครึ่งหนึ่งด้วยไอซีเบอร์ 74LS83 ตัวที่ 3
4. ต่อ O/P ของไอซีเบอร์ 74LS83 ตัวที่ 2 จะบวกกับสัญญาณที่ Delay ไปครั้งที่ 2 ครึ่งที่เหลือด้วยไอซีเบอร์ 74LS83 ตัวที่ 4
5. O/P ของไอซี ตัวที่ 3 กับ ตัวที่ 4 จะเป็นข้อมูลที่บวกกันแล้ว
6. วัดผลการบวก



รูปที่ 7.4 วงจรภาค Accumulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

	0	0	0	1	1	1	0	1	=	29
+	1	1	1	0	0	0	1	0	=	-29
	0	1	0	0	0	1	1	1	=	71
	0	1	0	1	1	0	1	0	=	90

ผลการบวก

	1	1	1	0	0	0	1	0	=	-29	ของเลขฐาน 10
+	0	1	0	0	0	1	1	1	=	71	ของเลขฐาน 10
+	0	1	0	1	1	0	1	0	=	90	ของเลขฐาน 10
	1	0	0	0	0	1	0	0	=	132	ของเลขฐาน 10

การบวกเลขติดลบ จะใช้ One Complement ช่วย

สรุปผลการทดลอง

O/P ที่ได้จากภาค Look up table ทั้ง 3 สัญญาณ จะเข้ามาบวกกันโดยจะใช้ไอซีเบอร์ 74LS83 4 ตัว ที่ทำหน้าที่เป็น ADDER ซึ่งแต่ละตัวจะเชื่อมต่อกันด้วย ขา C0 และ C4 ทำให้ได้ O/P ที่ออกมาของภาค Accumulator เป็นสัญญาณ 8 bit

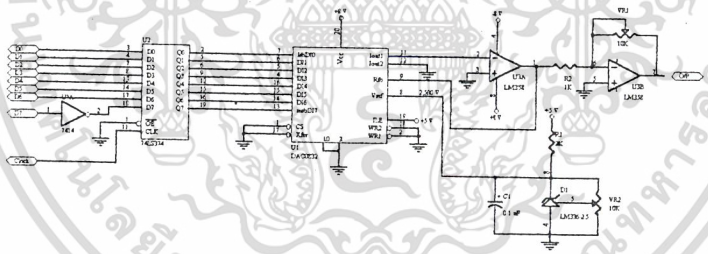
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7.5 การทดลองภาค D/A Converter

วัตถุประสงค์ เพื่อทำการทดลองการทำงานของภาค D/A Converter ว่าสามารถแปลงสัญญาณจากสัญญาณ Digital เป็น Analog ได้หรือไม่

ขั้นตอนการทดลอง

1. ป้อนข้อมูลให้แก่ Data Input ของ DAC0832 เพื่อการทดลองหาผลลัพธ์ที่ได้โดยป้อนค่าตั้งแต่ 00H ถึง FFH โดย DAC0832 สามารถรับข้อมูลได้ 8 บิท
2. ใช้แรงดันอ้างอิงที่ VREF+ = +2.500V
3. วัด O/P ที่ได้ทางทางขา Analog Output จากวงจรได้ค่าดังตาราง



รูปที่ 7.5 วงจรภาค D/A Converter

การทำงานของวงจรภาคเอาท์พุทแบบอนาลอก

ภาคเอาท์พุทแบบอนาลอก ใช้ไอซี DAC (DIGITAL TO ANALOG CONVERTER) เป็นส่วนประกอบที่สำคัญ ของวงจรโดยไอซี DAC จะทำหน้าที่ตรงข้ามกับ ADC กล่าวคือ มันจะทำหน้าที่เปลี่ยนสัญญาณข้อมูลแบบดิจิตอบ ให้เป็นสัญญาณเอาท์พุทแบบ อนาลอก ไอซี DAC ที่ใช้คือ DAC0832 ซึ่งเป็นไอซี DAC ขนาด 8 บิท ซึ่งไอซี DAC นี้จะให้สัญญาณเอาท์พุทออกมาเป็นแบบอนาลอกแต่ขนาดของสัญญาณที่ได้ยังมีขนาดต่ำไม่สามารถนำไปใช้ควบ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุมการทำงานของวงจรอื่นได้โดยตรง ต้องมีวงจรขยายสัญญาณเอาท์พุทให้มีขนาดสูงขึ้นก่อนนำไปใช้งาน

หลักการทำงานของวงจร

ไอซี DAC จะทำหน้าที่นำข้อมูลอินพุทแบบดิจิตอลมาเปรียบเทียบกับแรงดันอ้างอิงของวงจร แล้วเปลี่ยนเป็นสัญญาณอนาลอกทางเอาท์พุท ซึ่งขนาดของสัญญาณเอาท์พุทที่ได้จะมีความสัมพันธ์กับขนาดของข้อมูลอินพุทแบบเป็นสัดส่วนเชิงเส้น (LINEAR) โดยขนาดของสัญญาณเอาท์พุทนั้นสามารถแบ่งขนาดออกได้เป็นช่วง ๆ (STEP) ตามขนาดของไอซี DAC ที่ใช้ค่า STEP นี้จะเป็นตัวแสดงถึงความละเอียดของสัญญาณเอาท์พุทของวงจร DAC ในแต่ละช่วงของการเปลี่ยนแปลงของจำนวนข้อมูลอินพุท ว่าเมื่อค่าข้อมูลเพิ่มขึ้น 1 ระดับ ขนาดของสัญญาณเอาท์พุทเพิ่มขึ้นเท่าไร ถ้าไอซี DAC มีจำนวนบิทมาก ค่าความละเอียดก็จะมาก ถ้าไอซี DAC มีจำนวนบิทน้อยค่าความละเอียดก็จะมีค่าน้อยตามไปด้วย

ความสัมพันธ์ของอินพุทและเอาท์พุทของไอซี DAC

ดังได้กล่าวมาแล้วว่า ไอซี DAC จะใช้การเปรียบเทียบจำนวนของข้อมูลอินพุทแบบดิจิตอล แบ่งแรงดันอ้างอิงของวงจรแล้วจึงเปลี่ยนขนาดของข้อมูลอินพุทนั้นกลับเป็นสัญญาณแบบอนาลอกส่งออกทางเอาท์พุท ซึ่งขนาดของข้อมูลอินพุท กับขนาดของสัญญาณเอาท์พุทมีความสัมพันธ์กันดังนี้ คือ

- ถ้าค่าของข้อมูลอินพุทมีค่าสูงสุด (เป็นโลจิกสูง “1” ทั้งหมดทุกบิท) จะได้ขนาดของสัญญาณอนาลอกเอาท์พุทสมีค่าเป็น 2 เท่าของแรงดันอ้างอิงของวงจร
- ถ้าค่าของข้อมูลอินพุท มีค่าเป็นครึ่งหนึ่งของข้อมูลสูงสุด จะได้ขนาดสัญญาณอนาลอกเอาท์พุท มีค่าเท่ากับค่าของแรงดันอ้างอิงของวงจร

ซึ่งขนาดของสัญญาณเอาท์พุทของวงจรถูกกำหนดโดยค่าตัวแปรต่าง ๆ ดังนี้คือ

1. **แหล่งจ่ายไฟเลี้ยงวงจร** DAC
   ค่าของแรงดันของแหล่งจ่ายไฟที่ป้อนให้วงจร DAC นี้จะเป็นตัวกำหนดขนาดสูงสุดของสัญญาณอนาลอกเอาท์พุท คือ แหล่งจ่ายไฟตรงจากภายนอกต้องเป็นแรงดันไฟตรงที่ผ่านการ RECTIFIER และ FILTER แล้วเท่านั้น
2. **แรงดันอ้างอิงของวงจร** (Voltage Reference **หรือ** Vref/2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันอ้างอิงของวงจรนี้เป็นจุดอ้างอิงมาตรฐานของวงจรเป็นตัวกำหนดความเที่ยงตรงในการเปลี่ยนขนาดของสัญญาณข้อมูลแบบดิจิตอล ให้เป็นสัญญาณอนาลอก ซึ่งวงจรต้องการแรงดันอ้างอิงที่มีความเที่ยงตรงสูงเพราะหากแรงดันอ้างอิงของวงจรไม่มีความเที่ยงตรงแน่นอน มีการเปลี่ยนแปลงแล้ว การทำงานของวงจรก็จะเปลี่ยนแปลงตามด้วย นั่นก็คือจะส่งผลให้ค่าของสัญญาณอนาลอกเอาท์พุทที่ได้เปลี่ยนแปลง ขึ้น ๆ ลง ๆ ตามไปด้วย ถึงแม่ว่าค่าของข้อมูลอินพุทที่ป้อนให้ไอซี DAC จะมีขนาดคงที่ก็ตาม และแรงดันอ้างอิงของวงจรนี้ยังเป็นตัวกำหนดสัดส่วนของข้อมูลอินพุทและสัญญาณเอาท์พุทอีกด้วย ซึ่งใช้ไอซี LM336 เป็นตัวสร้างสัญญาณแรงดันอ้างอิง ขนาด 2.50V

แรงดันอ้างอิงจากภายนอก โดยแรงดันอ้างอิงนี้เป็นไฟตรงมีค่าสูงสุดได้ไม่เกิน +10V และต้องควบคุมให้มีค่าน้อยกว่า แรงดันของแหล่งจ่ายไฟที่ป้อนให้กับวงจร DAC เสมอ

3. **ค่าเกณฑ์การขยายวงจร** OP-AMP

เนื่องจากสัญญาณอนาลอกเอาท์พุทได้จากไอซี DAC นั้นมีขนาดต่ำ ไม่สามารถนำไปใช้งานได้โดยตรงจึงต้องใช้วงจรขยายขนาดของสัญญาณให้สูงขึ้น ดังนั้นเกณฑ์การขยายของวงจรจึงเป็นตัวกำหนดอัตราส่วนของขนาดสัญญาณสูงสุดที่ต้องการด้วย

4. **ค่าของข้อมูลอินพุท** (DATA)

ค่าของข้อมูลอินพุทนี้เป็นตัวกำหนดขนาดของสัญญาณเอาท์พุทโดยตรง ซึ่งค่าของตัวแปรอื่น ๆ นั้นเรากำหนดเพียงครั้งเดียว แล้วกำหนดคงที่ไว้ที่ตำแหน่งนั้นเลย. แต่ค่าของข้อมูลอินพุทนี้สามารถเปลี่ยนแปลง ขึ้น ๆ ลง ๆ ได้ตลอดเวลาเพื่อควบคุมขนาดของสัญญาณเอาท์พุทในช่วงเวลาต่าง ๆ ตามความต้องการซึ่งถ้าค่าของข้อมูลมีค่ามากก็จะได้ขนาดของสัญญาณเอาท์พุทมาก แต่ถ้าค่าของข้อมูลมีค่าน้อยก็จะทำให้ได้ขนาดของสัญญาณเอาท์พุทน้อยตามไปด้วยเช่นกัน

การกำหนดค่าของข้อมูลเพื่อกำหนดขนาดของสัญญาณเอาท์พุท

การที่เราจะสามารถกำหนดค่าของข้อมูลอินพุทของไอซี DAC เพื่อให้ได้ขนาดของสัญญาณเอาท์พุท ตามความต้องการของเรานั้น ก่อนอื่นเราต้องทราบความสามารถของไอซี DAC เสียก่อนว่าสามารถให้ค่าสัญญาณเอาท์พุทต่อ STEP เป็นเท่าไรเสียก่อน แล้วจึงหาว่าหากเราต้องการได้ค่าสัญญาณเอาท์พุทขนาดเท่านี้แล้วต้องใช้กี่ STEP นี้สามารถกำหนดได้จากจำนวน บิทข้อมูลของไอซี DAC ที่เราใช้ในวงจร

ถ้าเราใช้ไอซี DAC ขนาด 8 บิท จะได้ค่าของข้อมูลอยู่ระหว่าง 00H-FFH หรือ 0 – 255 นั่นก็คือ ไอซี DAC สามารถให้สัญญาณเอาท์พุทเป็นช่วง ๆ ทั้งหมด 256 ช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(0 – 255) นั่นเอง

ถ้าเราใช้ไอซี DAC ขนาด 12 บิท จะได้ค่าของข้อมูลอยู่ระหว่าง 000H – FFFH หรือ 0 – 1023 นั่นก็คือไอซี DAC สามารถให้สัญญาณเอาท์พุทเป็นช่วง ๆ ทั้งหมด 1024 ช่วง (0 – 1023) นั่นเอง

ซึ่งเมื่อเราทราบว่าไอซี DAC มีค่า STEP ทั้งหมดเท่าใดแล้ว ก็สามารถทราบได้ว่าใน 1 STEP จะได้ขนาดของสัญญาณเอาท์พุทต่อ 1 STEP เป็นเท่าใดโดยสามารถหาได้จากสูตร

STEP = ANALOG MAX / DAC STEP

เมื่อ STEP คือ ค่าของขนาดสัญญาณอนาลอกเอาท์พุทต่อ 1 ช่วงข้อมูล

ANALOG MAXIMUM คือ ค่าของขนาดสัญญาณอนาลอกเอาท์พุทสูงสุดที่ใช้ในวงจร

DAC STEP คือ ค่าของจำนวน STEP ของไอซี DAC ที่ใช้ในวงจรโดย

* ถ้าใช้ไอซี DAC0832 (DAC ขนาด 8 บิท) DAC STEP มีค่าเป็น 256
* ถ้าใช้ไอซี DAC1232 (DAC ขนาด 12 บิท) DAC STEP มีค่าเป็น 1024

หาค่าของข้อมูลได้จากสูตร DATA = Vout / STEP

เมื่อ DATA คือ ค่าของข้อมูลที่ต้องการหา

Vout คือ ค่าของขนาดสัญญาณอนาลอกเอาท์พุทที่ต้องการ

STEP คือ ค่าของขนาดสัญญาณอนาลอกเอาท์พุทต่อ 1 ช่วงข้อมูล

<u>ตัวอย่าง</u>

ใช้ไอซี DAC ขนาด 8 บิท (DAC0832)

ใช้แรงดันอ้างอิงของวงจร (Vref / 2) เป็น 2.50V

ให้สัญญาณเอาท์พุทมีขนาดสูงสุด +5V

จะได้ค่าของสัญญาณเอาท์พุทต่อ 1 STEP เป็น

STEP = 5/256

= 0.02V

จะเห็นได้ว่าเราได้ค่าของสัญญาณเอาท์พุทต่อ STEP มีค่าประมาณ 0.02 V หรืออาจกล่าวได้ว่า หากเราให้ ข้อมูลอินพุทกับวงจรด้วยค่า 01H (00000001B) เราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเอาท์พุทมีขนาดประมาณ 0.02V ดังนั้นหากเราต้องการได้ขนาดของสัญญาณเอาท์พุทมีค่าเป็น 2.50V เราต้องให้ค่าข้อมูลกับวงจรเป็น

DATA = 2.50/0.02

= 128 (7FH หรือ 01111111B)

ผลการทดลองภาค D/A Converter

D7	D6	D5	D4	D3	D2	D1	D0	แรงดัน (V)
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0.02
0	0	0	0	0	0	1	0	0.04
0	0	0	0	0	0	1	1	0.06
0	0	0	0	0	1	0	0	0.08
0	0	0	0	0	1	0	1	0.1
0	0	0	0	0	1	1	0	0.12
0	0	0	0	0	1	1	1	0.14
0	0	0	0	1	0	0	0	0.16
0	0	0	0	1	0	0	1	0.18
0	0	0	0	1	0	1	0	0.2
0	0	0	0	1	0	1	1	0.22
0	0	0	0	1	1	0	0	0.24
0	0	0	0	1	1	0	1	0.26
0	0	0	0	1	1	1	0	0.28
0	0	0	0	1	1	1	1	0.3
0	0	0	1	0	0	0	0	0.32
0	0	0	1	0	0	0	1	0.34
0	0	0	1	0	0	1	0	0.36
0	0	0	1	0	0	1	1	0.38
0	0	0	1	0	1	0	0	0.4
0	0	0	1	0	1	0	1	0.41
0	0	0	1	0	1	1	0	0.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	0	0	1	0	1	1	0	0.45
0	0	0	1	1	0	0	0	0.47
0	0	0	1	1	0	0	1	0.49
0	0	0	1	1	0	1	0	0.51
0	0	0	1	1	0	1	1	0.53
0	0	0	1	1	1	0	0	0.55
0	0	0	1	1	1	0	1	0.57
0	0	0	1	1	1	1	0	0.59
0	0	0	1	1	1	1	1	0.61
0	0	1	0	0	0	0	0	0.63
0	0	1	0	0	0	0	1	0.65
0	0	1	0	0	0	1	0	0.66
0	0	1	0	0	0	1	1	0.68
0	0	1	0	0	1	0	0	0.7
0	0	1	0	0	1	0	1	0.72
0	0	1	0	0	1	1	0	0.74
0	0	1	0	0	1	1	1	0.76
0	0	1	0	1	0	0	0	0.78
0	0	1	0	1	0	0	1	0.8
0	0	1	0	1	0	1	0	0.82
0	0	1	0	1	0	1	1	0.84
0	0	1	0	1	1	0	0	0.86
0	0	1	0	1	1	0	1	0.88
0	0	1	0	1	1	1	1	0.9
0	0	1	1	0	0	0	0	0.94
0	0	1	1	0	0	0	1	0.96
0	0	1	1	0	0	1	0	0.98
0	0	1	1	0	0	1	1	1
0	0	1	1	0	1	0	0	1.02
0	0	1	1	0	1	0	1	1.04
0	0	1	1	0	1	1	0	1.06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	0	1	1	0	1	1	1	1.08
0	0	1	1	1	0	0	0	1.09
0	0	1	1	1	0	0	1	1.11
0	0	1	1	1	0	1	0	1.13
0	0	1	1	1	0	1	1	1.15
0	0	1	1	1	1	0	0	1.17
0	0	1	1	1	1	0	1	1.19
0	0	1	1	1	1	1	0	1.21
0	0	1	1	1	1	1	1	1.23
0	1	0	0	0	0	0	0	1.25
0	1	0	0	0	0	0	1	1.27
0	1	0	0	0	0	1	0	1.29
0	1	0	0	0	0	1	1	1.31
0	1	0	0	0	1	0	0	1.33
0	1	0	0	0	1	0	1	1.35
0	1	0	0	0	1	1	0	1.37
0	1	0	0	0	1	1	1	1.39
0	1	0	0	1	0	0	0	1.41
0	1	0	0	1	0	0	1	1.43
0	1	0	0	1	0	1	0	1.45
0	1	0	0	1	0	1	1	1.46
0	1	0	0	1	1	0	0	1.48
0	1	0	0	1	1	0	1	1.50
0	1	0	0	1	1	1	0	1.52
0	1	0	0	1	1	1	1	1.54
0	1	0	1	0	0	0	0	1.56
0	1	0	1	0	0	0	1	1.58
0	1	0	1	0	0	1	0	1.6
0	1	0	1	0	0	1	1	1.62
0	1	0	1	0	1	0	0	1.64
0	1	0	1	0	1	0	1	1.66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	1	0	1	0	1	1	0	1.68
0	1	0	1	0	1	1	1	1.7
0	1	0	1	1	0	0	0	1.72
0	1	0	1	1	0	0	1	1.74
0	1	0	1	1	0	1	0	1.76
0	1	0	1	1	0	1	1	1.78
0	1	0	1	1	1	0	0	1.8
0	1	0	1	1	1	0	1	1.82
0	1	0	1	1	1	1	0	1.84
0	1	0	1	1	1	1	1	1.86
0	1	1	0	0	0	0	0	1.88
0	1	1	0	0	0	0	1	1.9
0	1	1	0	0	0	1	0	1.92
0	1	1	0	0	0	1	1	1.93
0	1	1	0	0	1	0	0	1.95
0	1	1	0	0	1	0	1	1.97
0	1	1	0	0	1	1	0	1.99
0	1	1	0	0	1	1	1	2.01
0	1	1	0	1	0	0	0	2.03
0	1	1	0	1	0	0	1	2.05
0	1	1	0	1	0	1	0	2.07
0	1	1	0	1	0	1	1	2.09
0	1	1	0	1	1	0	0	2.11
0	1	1	0	1	1	0	1	2.13
0	1	1	0	1	1	1	0	2.15
0	1	1	0	1	1	1	1	2.17
0	1	1	1	0	0	0	0	2.19
0	1	1	1	0	0	0	1	2.21
0	1	1	1	0	0	1	0	2.23
0	1	1	1	0	0	1	1	2.25
0	1	1	1	0	1	0	0	2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	1	1	1	0	1	0	1	2.29
0	1	1	1	0	1	1	0	2.31
0	1	1	1	0	1	1	1	2.33
0	1	1	1	1	0	0	0	2.35
0	1	1	1	1	0	0	1	2.36
0	1	1	1	1	0	1	0	2.38
0	1	1	1	1	0	1	1	2.4
0	1	1	1	1	1	0	0	2.42
0	1	1	1	1	1	0	1	2.44
0	1	1	1	1	1	1	0	2.46
0	1	1	1	1	1	1	1	2.48
1	0	0	0	0	0	0	0	2.5
1	0	0	0	0	0	0	1	2.52
1	0	0	0	0	0	1	0	2.54
1	0	0	0	0	0	1	1	2.56
1	0	0	0	0	1	0	0	2.58
1	0	0	0	0	1	0	1	2.6
1	0	0	0	0	1	1	0	2.62
1	0	0	0	0	1	1	1	2.64
1	0	0	0	1	0	0	0	2.66
1	0	0	0	1	0	0	1	2.68
1	0	0	0	1	0	1	0	2.7
1	0	0	0	1	0	1	1	2.72
1	0	0	0	1	1	0	0	2.74
1	0	0	0	1	1	0	1	2.76
1	0	0	0	1	1	1	0	2.77
1	0	0	0	1	1	1	1	2.79
1	0	0	1	0	0	0	0	2.81
1	0	0	1	0	0	0	1	2.83
1	0	0	1	0	0	1	0	2.85
1	0	0	1	0	0	1	1	2.87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้