ปีการศึกษา 2530

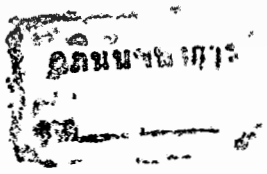
การส่งสัญญาณเสียงผ่านไมโครคอมพิวเตอร์

โดย

นายวิชัย นทีนันทสวัสดิ์ 27.1177

อาจารย์ที่ปรึกษา

อาจารย์สมยศ จุณณะปิยะ



024668 29 มค 2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2530

ภาควิชา โทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การส่งสัญญาณเสียงผ่านไมโครคอมพิวเตอร์

ผู้จัดทำ วิชัย นทีนันทสวัสดิ์

..... อาจารย์ที่ปรึกษา

(อ. สมยศ จุณณะปิยะ)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การส่งสัญญาณเสียงผ่านไมโครคอมพิวเตอร์

นายวิชัย นทีนันทสวัสดิ์
อ.สมยศ จุณณะปิยะ อาจารย์ที่ปรึกษา
ปีการศึกษา 2530

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้แสดงการส่งสัญญาณเสียง(อนาล็อก)ผ่านไมโครคอมพิวเตอร์ ซึ่งเป็นการประยุกต์ใช้งานไมโครคอมพิวเตอร์ โดยทำการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล เพื่อให้สามารถติดต่อกับไมโครคอมพิวเตอร์ได้ แล้วจึงทำการส่งข้อมูลผ่าน RS-232C ซึ่งใช้ 8251 USART เป็นตัวแปลงข้อมูลแบบขนานให้อยู่ในรูปของข้อมูลแบบอนุกรมและใช้ MC1488 กับ MC1489 เป็นตัวปรับระดับศักดาจาก TTL(5 โวลต์) ให้อยู่ในรูปของสัญญาณ $\pm$ 12 โวลต์ และแปลงสัญญาณ $\pm$ 12 โวลต์ กลับเป็นศักดา TTL ตามลำดับ เพื่อทำให้การส่งข้อมูลได้ชัดเจนและได้ระยะทางไกลขึ้น ส่วนเครื่องไมโครคอมพิวเตอร์ทางด้านรับ ก็จะเปลี่ยนข้อมูลแบบอนุกรมกลับเป็นข้อมูลแบบขนาน จากนั้นจึงทำการแปลงสัญญาณดิจิตอลกลับเป็นสัญญาณอนาล็อกตามเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VOICE TRANSMISSION ON MICROCOMPUTER

Mr.Wichai Nateenantasawasd

Mr.Somyot Junnapiya Advisor

1987

Abstract

This project shows the method of voice signal transmitting, analog signal ,on microcomputer . The application of using microcomputer in this project has done with analog to digital conversion. Then we send the data through RS-232 with the help of 8251 USART to transfer the parallel data to serial data.The use of MC1488 change the TTL level (5V) to ±12V and MC1489 change the ±12V back to TTL volt. In order to have better performance and longer distance. The receive microcomputer change the serial data back to parallel. Then transfer digital signal to analog signal.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
ชื่อปริญญานิพนธ์และผู้จัดทำ	ก
บทคัดย่อ	ข
ABSTRACT	ค
บทที่ 1 บทนำ	1
บทที่ 2 โครงสร้างของเครื่องแอปเบิ้ลทู	2
2.1 หน่วยความจำของเครื่องแอปเบิ้ลทู	2
2.2 การดีโค๊ดสล๊อต	3
2.3 กลุ่มสัญญาณควบคุม	7
บทที่ 3 บล็อกไดอะแกรมและการทำงานของระบบ	13
3.1 การแปลงสัญญาณอนาล็อกเป็นดิจิตอล	13
3.1.1 ADC ชนิด RAMP TYPE	14
3.1.2 ADC ชนิด SUCCESSIVE APPROXIMATION	15
3.2 การแปลงสัญญาณดิจิตอลเป็นอนาล็อก	16
3.3 การรับส่งข้อมูลแบบอนุกรม	17
3.4 8251 USART	21
บทที่ 4 การทดลอง และ ผลการทดลอง	32
บทที่ 5 สรุปผลและวิจารณ์	41
ภาคผนวก	42
กิตติกรรมประกาศ	51
หนังสืออ้างอิง	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ปัจจุบันไมโครคอมพิวเตอร์ได้มีการใช้งานกันอย่างแพร่หลาย อันเนื่องมาจากความสะดวกรวดเร็ว ความถูกต้องแม่นยำสูง และราคาก็ถูกลง การพัฒนาการเชื่อมต่อไมโครคอมพิวเตอร์เข้าด้วยกัน ตลอดจนอุปกรณ์เทอร์มินัลต่างๆ ได้ก้าวเข้าสู่โครงข่าย ซึ่งจะเป็นดิจิตอลทั้งหมด และ ให้บริการรวมมากมายเพื่อประโยชน์ในด้านความสะดวกและรวดเร็วแก่ลูกค้าหรือผู้ใช้บริการ โครงข่ายดิจิตอลที่ให้บริการดังกล่าวเราเรียกว่า ISDN (INTEGRATED SERVICE DIGITAL NETWORK) ดังนั้นพื้นฐานของการพัฒนาระบบดังกล่าวก็คือ ความพยายามในการปรับปรุงและเปลี่ยนระบบอนาล็อกให้เป็นระบบดิจิตอลทั้งหมด เพื่อให้โครงข่ายติดต่อสื่อสารกันได้ทั้งหมด ดังนั้นการเชื่อมโยงระหว่างไมโครคอมพิวเตอร์ 2 เครื่อง ซึ่งอยู่คนละสถานที่จึงเป็นจุดเริ่มต้นอย่างง่ายๆ ต่อการศึกษาระบบดังกล่าว

ปริญญานิพนธ์ฉบับนี้ได้ทำการศึกษาการเชื่อมต่อไมโครคอมพิวเตอร์ต้นทาง และปลายทาง โดยให้ติดต่อสื่อสารกันได้ และใช้มาตรฐานของ RS-232C ในการส่งผ่านข้อมูลแบบอนุกรม เพราะการรับส่งข้อมูลแบบอนุกรมจะมีข้อดีกว่าการรับส่งข้อมูลแบบขนานกล่าวคือ สามารถประหยัดค่าใช้จ่ายในการวางสายและความยุ่งยากซับซ้อนลงได้มาก ลำดับต่อไปก็จะทำการลองส่งข้อมูลซึ่งเป็นสัญญาณอนาล็อก จึงจำเป็นต้องหาวิธีให้ไมโครคอมพิวเตอร์สามารถเข้าใจและรับสัญญาณอนาล็อก ในที่นี้จึงใช้หลักการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลซึ่งได้เลือกใช้ไอซีที่ให้สัญญาณดิจิตอลออกมาเป็น 8 บิท เพื่อติดต่อกับเครื่องไมโครคอมพิวเตอร์ขนาด 8 บิท ระบบที่สร้างนี้จะสมบูรณ์ได้ก็ต่อเมื่อไมโครคอมพิวเตอร์ทางด้านรับต้องได้รับและแสดงสัญญาณเดิมเหมือนกับทางด้านส่งออกมา จึงต้องมีส่วนที่ใช้ในการแปลงสัญญาณดิจิตอลกลับมาเป็นสัญญาณอนาล็อกตามเดิม

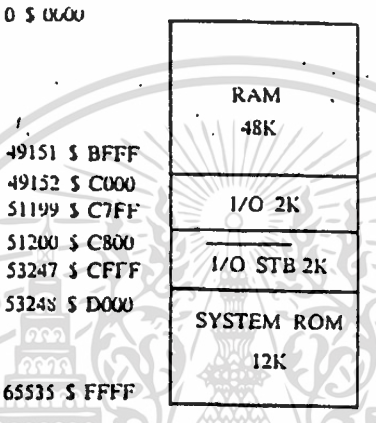
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

โครงสร้างของเครื่องแอปเปิ้ลทู

2.1 หน่วยความจำของเครื่องแอปเปิ้ลทู

เครื่องไมโครคอมพิวเตอร์ ที่ใช้ในการทำโปรเจคนี้คือ เครื่องแอปเปิ้ลทู ที่มี CPU เบอร์ 6502 และได้แบ่งหน่วยความจำทั้ง 64K ไบท์ ออกเป็นส่วน ๆ ดังรูป 2.1



รูปที่ 2.1 การแบ่งหน่วยความจำของเครื่องแอปเปิ้ลทู

2.1.1 ส่วนของ RAM มีขนาด 48K ไบท์ เริ่มจากแอดเดรส $0000-$BFFF เป็นส่วนที่เก็บตัวหนังสือบนจอภาพ (TEXT) รวมทั้งการใช้งานในโหมดกราฟฟิคและรวมทั้งที่เก็บ DOS ในกรณีที่ใช้ DISKDRIVE

2.1.2 ส่วนของพอร์ทมีขนาด 4K ไบท์ แบ่งเป็น

2.1.2.1 ON-BOARD I/O มีขนาด 128 ไบท์ อยู่ในช่วงแอดเดรส $C000 -$C07F ที่นำไปใช้งาน เช่น ลำโพง , คีย์บอร์ด , คาสเซ็ทไอโอ ส่วนแอดเดรสของพอร์ทต่างๆ พิจารณาตามตาราง 2.1

2.1.2.2 ส่วนที่เป็นสล๊อต มีทั้งหมด 8 สล๊อต เริ่มตั้งแต่ $C080-$C0FF เราสามารถนำเอาสล๊อตต่าง ๆ ไปใช้งานได้

2.1.3 ส่วนของ ROM มีขนาด 12K ไบท์ เริ่มจากแอดเดรส $D000-$FFFF ภายใน ROM บรรจุโปรแกรมมอนิเตอร์ 4K ไบท์ และอินเตอร์พรีเตอร์ภาษาเบสิค (APPLE SOFT) จำนวน 8K ไบท์

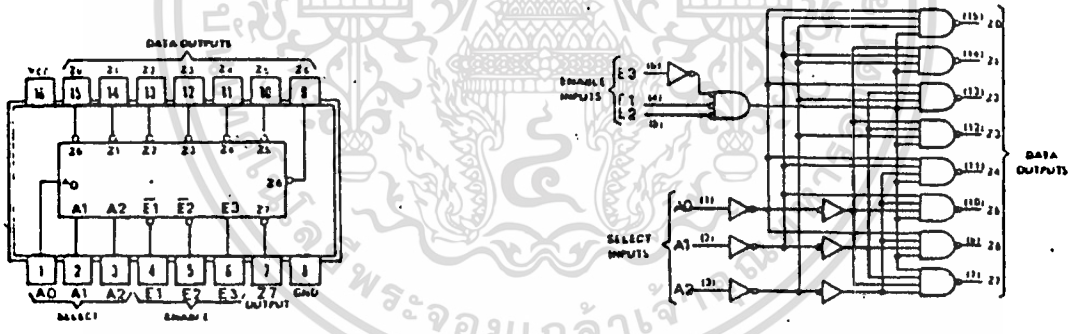
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่การทำงาน	แอดเดรส	
	เลขฐานสิบ	เลขฐานสิบหก
Keyboard data	49152	C000
Clear-keyboard strobe	49168	C010
Speaker	49200	C030
Cassette Output	49184	C020
Cassette Input	49256	C060
Flag Inputs	49249-49251	C061-C063
Analog Inputs	49252-49255	C064-C067
Analog Clear	49264	C070
Utility Strobe	49216	C040

ตาราง 2.1 หน้าที่การทำงานใน ON-BOARD I/O

2.2 การดีโค้ดสล๊อต

การดีโค้ดสล๊อตของแอปเปิ้ลใช้ไอซีเบอร์ 74LS138 (3 TO 8 LINE DECODER/ MULTIPLEXER) มีการทำงานดังรูป 2.2



INPUTS					OUTPUTS							
ENABLE		SELECT										
E3	G	A2	A1	A0	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

นา G คือ E1 + E2

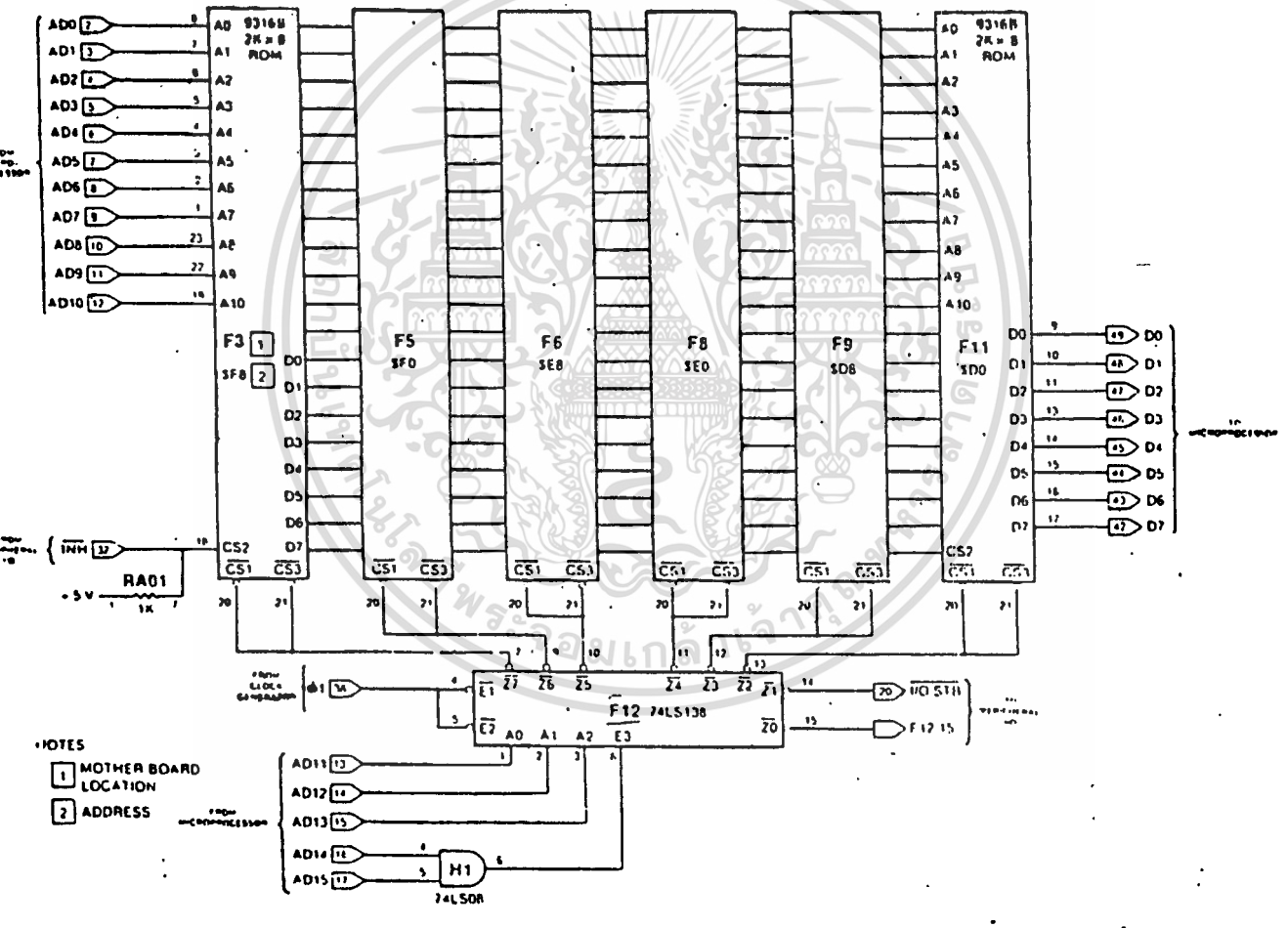
รูปที่ 2.2 โครงสร้างของไอซี 74LS138 และการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดีโค๊ด ช่วง ($C000-$CFFF) ต้องใช้ไอซีจำนวน 3 ตัว

2.2.1 ไอซี 74LS138 ตัวที่หนึ่ง ซึ่งอยู่ในตำแหน่ง F12 บนมาสเตอร์บอร์ด ทำหน้าที่ดีโค๊ดแอดเดรส $C000 เป็นต้นไป มีเงื่อนไขการดีโค๊ด ดังรูป 2.3

2.2.1.1 ขา AD14 และ AD15 ต่อกับอินพุทของ AND GATE แล้วจึงเข้าขา E3 การทำดังนี้ก็เพื่อกำหนดการดีโค๊ด เมื่อ AD14 และ AD15 มีลอจิกเป็น "1" นั่นเอง คือ แอดเดรสตั้งแต่ $C000 เป็นต้นไป ดูรูปที่ 2.3 ประกอบ



รูปที่ 2.3 แสดงการดีโค๊ดของไอซี F12

2.2.1.2 ขา AD11,AD12 และ AD13 ซึ่งต่ออยู่กับขา A0,A1,A2 จะเป็นบิทที่กำหนดการทำงานของ F12 ในช่วง แอดเดรสต่างๆ กันดังตาราง 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AD15,AD14,AD13,AD12,AD11	ช่วงแอดเดรส	จำนวนไบท์	การทำงานของ F_{12}	หน้าที่
1 1 0 0 0	$ C000-$ C7FF	2K	ขา Z_0	นำไปดีโค้ดต่อที่ไอซี H-12
1 1 0 0 1	$ C800-$ CFFF		Z_1	ขา $\overline{\text{I/O STB}}$ ของสล๊อต
1 1 0 1 0	$ D000-$D7FF		Z_2	CS ของ ROM F11
1 1 0 1 1	$ D800-$DFFF		Z_3	CS ของ ROM F9
1 1 1 0 0	$ E000-$ F7FF		Z_4	CS ของ ROM F8
1 1 1 0 1	$ E800-$EFFF		Z_5	CS ของ ROM F6
1 1 1 1 0	$ F000-$ F7FF		Z_6	CS ของ ROM F5
1 1 1 1 1	$ F800-$ FFFF		Z_7	CS ของ ROM F3

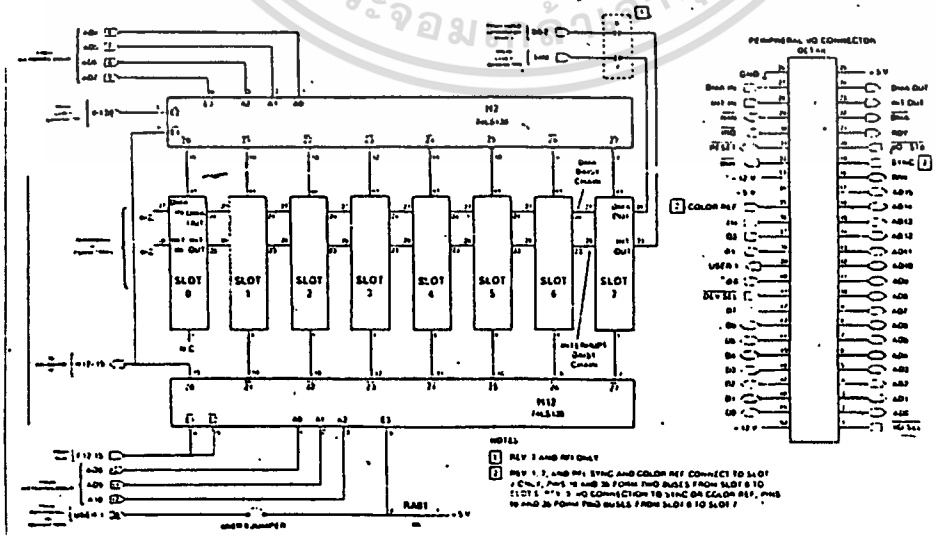
ตาราง 2.2 แสดงการดีโค๊ดของไอซี F12

2.2.2 ไอซี 74LS138 ตัวที่สอง ที่ตำแหน่ง H12 บนบอร์ด จะทำการดีโค๊ด ขา $\overline{\text{I/O SELECT}}$ ในแต่ละสล๊อต ซึ่งมีแอดเดรสอยู่ในช่วง $C000-$C7FF ตามเงื่อนไข ดังรูปที่ 2.4

2.2.2.1 ขา $\overline{Z0}$ ของ F12 ต่อกับขา $\overline{E1}$,$\overline{E2}$ ของ H12 จะเป็นตัวกำหนดการดีโค๊ดของช่วงแอดเดรส $C000-$C7FF

2.2.2.2 ขา E3 ของ H12 ต่อกับลอจิก "1" โดยผ่านความต้านทาน 1K โอห์มทำให้ H12 ดีโค๊ดได้ตลอดเวลาที่ขา $\overline{Z0}$ ของ F12 เป็นลอจิก "0"

2.2.2.3 ขา AD8, AD9, AD10 ที่ต่อกับขา A0, A1, A2 จะเป็นบิทที่กำหนดแอดเดรสต่างๆ ตามตาราง 2.3



รูปที่ 2.4 แสดงการดีโค๊ดขา $\overline{\text{I/O SEL}}$ และ $\overline{\text{DEV SEL}}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AD_{10}	AD_9	AD_8	ช่วงแอดเดรส	จำนวนไบท์	การทำงาน H_{12}	หน้าที่
0	0	0	$ C000-$ C0FF		Z_0	นำไปถอดรหัสต่อที่ไอซี H2 และ F13
0	0	1	$ C100-$ C1FF		Z_1	ขา I/O SELECT ของสล๊อต 1
0	1	0	$ C200-$ C2FF		Z_2	ขา I/O SELECT ของสล๊อต 2
0	1	1	$ C300-$ C3FF	256	Z_3	ขา I/O SELECT ของสล๊อต 3
1	0	0	$ C400-$ C4FF		Z_4	ขา I/O SELECT ของสล๊อต 4
1	0	1	$ C500-$ C5FF		Z_5	ขา I/O SELECT ของสล๊อต 5
1	1	0	$ C600-$ C6FF		Z_6	ขา I/O SELECT ของสล๊อต 6
1	1	1	$ C700-$ C7FF		Z_7	ขา I/O SELECT ของสล๊อต 7

ตาราง 2.3 แสดงการดีโค๊ดและค่าแอดเดรสของขา $\overline{\text{I/O SEL}}$ ในแต่ละสล๊อต

ดังนั้นการดีโค๊ดครั้งที่สองเป็นการดีโค๊ด $\overline{\text{I/O SELECT}}$ ในแต่ละสล๊อต ส่วนขา Z0 ของ H12 นอกจากจะนำไปดีโค๊ดต่อในไอซี 74LS138 ตัวที่ 3 แล้วยังนำไปเข้าขา E2 ของ 74LS138 ที่ตำแหน่ง F13 เพื่อทำการดีโค๊ด ON-BOARD I/O ในช่วงแอดเดรส $C00-$C07Fตามที่กล่าวมาแล้ว

2.2.3 ไอซีตัวที่ 3 ที่ตำแหน่ง H12 ตามรูป จะทำการดีโค๊ด DEVICE SELECT ของแต่ละสล๊อตในช่วงแอดเดรส $C08-$C0FF

2.2.3.1 ขา $\overline{E1}$ ของ H2 ต่อกับขา $\overline{Z0}$ ของ H12 เป็นการกำหนดในช่วง $C00-$C0FF

2.2.3.2 ขา $\overline{E2}$ ของ H2 ต่อกับ $\phi 1$ ด้วยเหตุผลเดียวกับการต่อ F12 คือกำหนดการทำงานของ H2 ในช่วงเวลาของ CPU

2.2.3.3 การต่อขา E3 เข้ากับขา AD7 เป็นการกำหนดว่า H2 จะทำการดีโค๊ดเมื่อ AD7 เป็น "1" เท่านั้น นั่นคือ H2 จะทำการดีโค๊ดตั้งแต่แอดเดรส $C08 เป็นต้นไป

2.2.3.4 ส่วนขา AD4, AD5, AD6 เป็นบิทที่กำหนดว่าช่วงการทำงานของ H2 ที่แอดเดรสต่างๆ ตามตาราง 2.4

นอกจากการดีโค๊ดดังกล่าว ยังสามารถทำ "FULL ADDRESS DECODER" ซึ่งเป็นการดีโค๊ด 16 บิท (AD0-AD15) ที่ตัวสล๊อตได้อีกด้วย

หลักในการทำ FULL ADDRESS DECODER มีหลายวิธี ตั้งแต่ GATE ต่างๆ ทำการดีโค๊ด ไอซีดีโค๊ด และ MAGNITUDE COMPARATOR แต่สิ่งที่ควรระวังอย่างหนึ่ง ก็คือในกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AD_6	AD_5	AD_4	ช่วงแอดเดรส	จำนวนไบท์	การทำงาน H2	หน้าที่
0	0	0	$ C080-$ C08F	16	Z_0	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 0
0	0	1	$ C090-$ C09F		Z_1	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 1
0	1	0	$ C0A0-$ C0AF		Z_2	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 2
0	1	1	$ C0B0-$ C0BF		Z_3	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 3
1	0	0	$ C0C0-$ C0CF		Z_4	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 4
1	0	1	$ C0D0-$ C0DF		Z_5	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 5
1	1	0	$ C0E0-$ C0EF		Z_6	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 6
1	1	1	$ C0F0-$ C0FF		Z_7	ขา $\overline{\text{DEVICE SELECT}}$ ของสล๊อต 7

ตาราง 2.4 แสดงการดีโค๊ดและค่าแอดเดรสของขา DEV SEL ในแต่ละสล๊อต

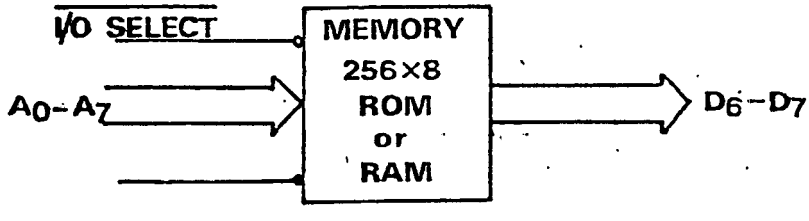
ที่แอดเดรสที่กำหนดดีโค๊ดครอบคลุมแอดเดรสต่างๆ ของสล๊อต ก็ควรจะทำการยกเลิกดีโค๊ดเสียก่อนโดยต่อขา USER1 (ขา39) กับลอจิก "Ø" ตามรูปที่ 2.4

ตามปกติมักจะไม่นิยมทำ FULL ADDRESS DECODER ในการอินเตอร์เฟสกับวงจรภายนอก เพราะว่าเนื่องจากการอินเตอร์เฟส โดยใช้ขา $\overline{\text{DEVICE SELECT}}$, $\overline{\text{I/O SELECT}}$, $\overline{\text{I/O STOBE}}$ ของสล๊อตมีความสะดวกมากอยู่แล้ว

2.3 กลุ่มสัญญาณควบคุม

2.3.1 $\overline{\text{I/O SELECT}}$ (ขา 1) มีทุกสล๊อตยกเว้น สล๊อต Ø จะแอดทีฟที่ลอจิก "Ø" เมื่อมีการเรียกแอดเดรสต่างๆ ตามตารางที่ 2.2 เนื่องจากขา $\overline{\text{I/O SELECT}}$ แต่ละสล๊อตมีช่วงแอดเดรส 256 ไบท์ จึงนำไปใช้งานได้ 2 วิธี คือ

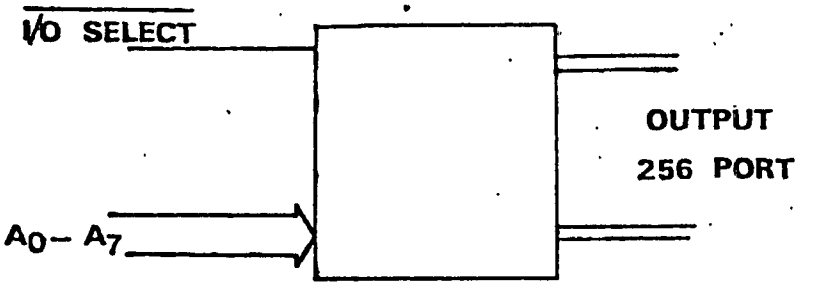
2.3.1.1 ใช้เป็นขา $\overline{\text{CS}}$ ของหน่วยความจำพวก RAM, ROM ที่มีความจุไม่มากนัก ไม่เกิน 256 ไบท์ จึงสามารถใช้ขา $\overline{\text{I/O SELECT}}$ ดังรูป 2.5



รูปที่ 2.5 การใช้ $\overline{\text{I/O SEL}}$ เป็นขา $\overline{\text{CS}}$

2.3.1.2 ใช้เป็นการ DECODE PORT ได้ 256 PORT ดังรูป 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

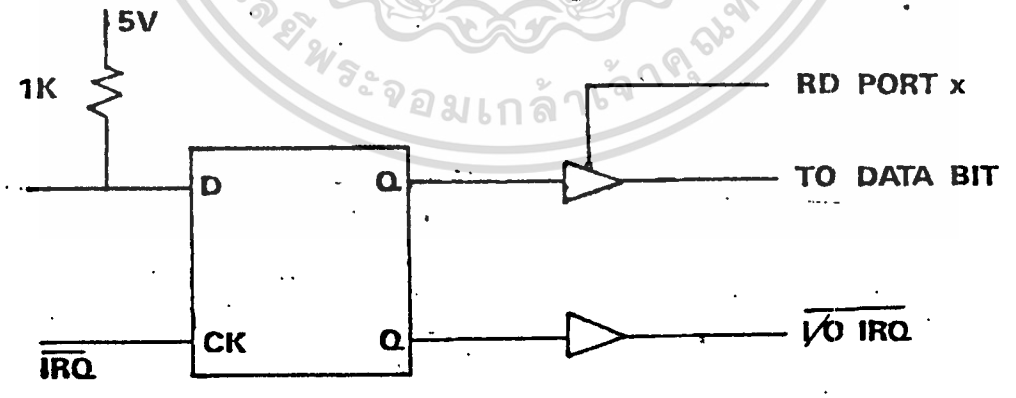


รูปที่ 2.6 แสดงการใช้ I/O SELECT เป็นขาควบคุมการดีโค๊ต

2.3.2 I/O STROBE (ขา 2Ø) แอคทีฟที่ลอจิก "Ø" เมื่อเรียกแอดเดรสในช่วง $C8ØØ-$CFFF (2K ไบท์) เมื่อสล๊อตใดสล๊อตหนึ่งใช้ไปแล้วสล๊อตอื่นจะใช้ไม่ได้

2.3.3 DEVICE SELECT (ขา 41) แอคทีฟที่ลอจิก "Ø" โดยมีช่วงห่างสล๊อตละ 16 แอดเดรส

2.3.4 IRQ (ขา 3Ø) และ NMI (ขา 29) เป็นขาอินเตอร์รัฟของ CPU เบอร์ 6502 ขา IRQ ย่อมาจาก INTERRUPT REQUEST เป็นชนิด MASKABLE ได้โดยใช้ซอฟแวร์ ส่วนขา NMI ย่อมาจาก NON-MASKABLE INTERRUPT เป็นชนิดที่ไม่สามารถห้ามการอินเตอร์รัฟได้ ปกติขา NMI จะต่อกับอุปกรณ์เพอร์ริเฟอร์รัลตัวที่มีความสำคัญมากที่สุดเพียงตัวเดียว และมี VECTOR ชี้ SUBROUTINE อยู่ที่แอดเดรส $Ø3FB



รูปที่ 2.7 แสดงวงจรอินเตอร์รัฟท์โดยขา IRQ

ส่วนขา IRQ สามารถใช้ร่วมกันได้กับอุปกรณ์เพอร์ริเฟอร์รัลได้หลายชนิด โดยอาศัยวงจรตามรูปข้างบน(รูป 2.7) และซอฟท์แวร์ตรวจสอบการอินเตอร์รัฟท์จากวงจร จะเห็นว่าเมื่อมีการขออินเตอร์รัฟท์มาที่ขา CK ของ D ฟลิปฟลอปของ Q จะส่งสัญญาณลอจิก "Ø"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปให้ CPU เพื่อขออินเตอร์รัฟท์โดยมีเวคเตอร์ในการอินเตอร์รัฟท์อยู่ที่แอดเดรส $Ø3FE และ $Ø3FF, CPU จะทำการหาอุปกรณ์ที่ขออินเตอร์รัฟท์โดยการตรวจสอบขา ซึ่งเป็นอินพุทพอร์ท ขนาด 1 บิท

2.3.5 INT IN (ขา 28) และ INT OUT (ขา 23) เป็นขาที่ใช้วัดความเป็นลำดับความสำคัญก่อนหลังของอุปกรณ์เพอร์ริเฟอร์รัลที่ขออินเตอร์รัฟท์ โดยวิธีการที่เรียกว่า "PULLING" ดังนั้นขณะที่อุปกรณ์เพอร์ริเฟอร์รัลที่มีความสำคัญสูง ส่งสัญญาณอินเตอร์รัฟท์ไปให้ก็จะเป็นการกันสัญญาณอินเตอร์รัฟท์จากอุปกรณ์อื่นๆ ที่มีความสำคัญต่ำกว่า โดยการให้ ลอจิก "Ø" กับขา AND GATE จนกว่าจะเสร็จสิ้นการตอบสนองอินเตอร์รัฟท์

2.3.6 $\overline{\text{DMA}}$ (ขา 22) เป็นสัญญาณแบบ 2 ทิศทาง (BIDIRECTION) แอคทีฟที่ลอจิก "Ø" เมื่อทำการเปิดเครื่อง หรือ กดปุ่ม RESET ที่คีย์บอร์ด ใช้สำหรับการ RESET อุปกรณ์เพอร์ริเฟอร์รัล หรือ 6502

2.3.7 $\overline{\text{RES}}$ (ขา 31) ใช้สำหรับการยกเลิกโปรแกรมมอนิเตอร์และอินเตอร์พรีเตอร์ภาษาเบสิค เมื่อต้องการใช้โปรแกรมของอุปกรณ์เพอร์ริเฟอร์รัลมาควบคุมระบบแทนโดยการต่อขา 31 ลงกราวด์

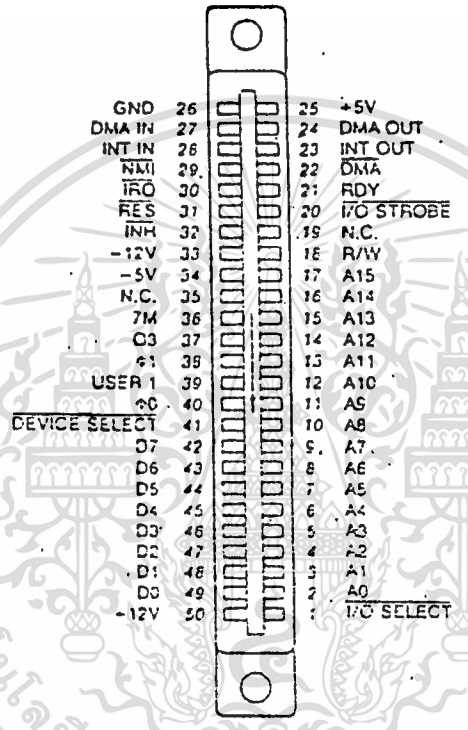
	ADDRESS RANGE	
Interface Slot	เลขฐาน 16	เลขฐาน 10
0	C080-C08F	49280-49295
1	C090-C09F	49296-49311
2	C0A0-C0AF	49312-49327
3	C0B0-C0BF	49328-49343
4	C0C0-C0CF	49344-49359
5	C0D0-C0DF	49360-49375
6	C0E0-C0EF	49376-49391
7	C0F0-C0FF	49392-49407

ตาราง 2.5 การจัดแอดเดรสของ DEV SEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

024668

จากตารางข้างบน ซึ่งเป็นตารางของสัญญาณ $\overline{\text{DEVICE SELECT}}$ ซึ่งก็คือ ขา 41 ของทุกสล๊อต ซึ่งได้มาจากไอซีดีโค๊ดเดอร์เบอร์ 74LS138 ตำแหน่ง H2 นั่นเอง ซึ่ง ขา 41 ของสล๊อต Ø จะแอคทีฟ LOW เมื่อ CPU ส่งแอดเดรสตั้งแต่ $C08Ø-$C08F หรือ (49280-49295 ฐาน 10) คือ เมื่อทำคำสั่ง POKE หรือ PEEK ที่ตำแหน่งดังกล่าวนั่นเอง



รูปที่ 2.8 แสดงรูป และ ไดอะแกรมของสล๊อต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่	ชื่อขา	การใช้งาน
1	$\overline{\text{I/O SELECT}}$	มีทุกสล๊อตยกเว้นสล๊อต ø แอดทีฟระหว่างที่ Φ1เป็น ø ที่ลอจิก "ø" เมื่อมีการเรียกแอดเดรสช่วง $ CNøø-$CNFF โดยที่ N เป็นเลขที่สล๊อต
2-17	Aø-A15	คือขาแอดเดรสของสล๊อต
18	R/$\overline{\text{W}}$	ควบคุมการอ่านหรือเขียนข้อมูล ที่สภาวะลอจิก"ø"เป็นการเขียนข้อมูลส่วนสภาวะ"1"เป็นการอ่านข้อมูล
19	SYNC	เป็นสัญญาณที่ใช้ซิ้งค์(SYNC)กับVIDEO TIMINGมีเฉพาะสล๊อต 7 เท่านั้น
2ø	$\overline{\text{I/O STROBE}}$	เป็นขาสัญญาณที่ต่อร่วมกันทุกสล๊อต แอดทีฟระหว่างที่ ø เป็น 0 ที่ลอจิก "ø" เมื่อมีการเรียกแอดเดรสช่วง $ C8ØØ-$ CFFF
21	RDY	เป็นขาที่แสดงสภาวะการทำงานของอุปกรณ์เพอร์ริเฟอรัลที่ส่งไปบอก CPU
22	$\overline{\text{DMA}}$	ควบคุมการทำ DIRECT-MEMORY ACCESS ของอุปกรณ์เพอร์ริเฟอรัล กับหน่วยความจำ
23	INT OUT	เป็นสัญญาณที่ใช้จัดลำดับความสำคัญของการทำอินเตอร์รัพ
24	DMA OUT	ใช้จัดลำดับความสำคัญของอุปกรณ์เพอร์ริเฟอรัลในการทำ DIRECT MEMORY ACCESS
25	+5 VOLTS	สามารถจ่ายกระแสถึง 5ØØ mA
26	GND	เป็นขากราวด์ของระบบ
27	DMA IN	ใช้ร่วมกับขา DMA OUT เพื่อจัดลำดับความสำคัญในการทำ DIRECT MEMORY ACCESS
28	INT IN	ใช้ร่วมกับขา INT OUT เพื่อจัดลำดับความสำคัญในการทำอินเตอร์รัพชนิด MASKABLE
29	$\overline{\text{NMI}}$	เป็นขาอินเตอร์รัพชนิด NON-MASKABLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

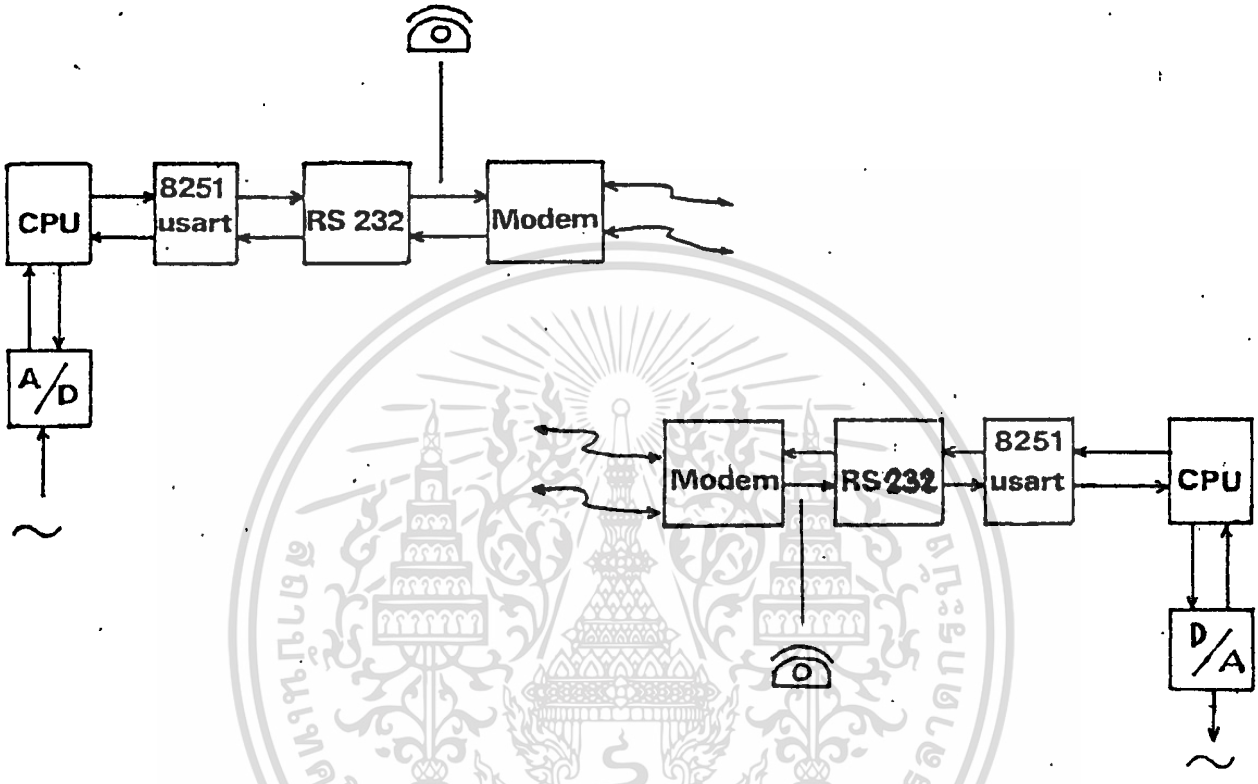
3Ø	$\overline{IRQ}$	เป็นขาอินเตอร์รัพชนิด MASKABLE และสามารถจัดลำดับความสำคัญในการขออินเตอร์รัพได้
31	$\overline{RES}$	เป็นขาชนิดสองทิศทางใช้สำหรับรีเซ็ท การทำงานของเครื่องแอปเปิ้ลและอุปกรณ์เพอร์ริเฟอรัล
32	$\overline{INH}$	ทำงานที่ลอจิก"Ø"เป็นการยกเลิก การทำงาน ROM ในช่วงแอดเดรส $D000-$FFFF
33	-12VOLTS	สามารถจ่ายกระแสได้ 200 mA
34	-5 VOLTS	สามารถจ่ายกระแสได้ 200 mA เช่นเดียวกับสัญญาณไฟ -12 VOLTS
35	COLOR REF	มีเฉพาะสล๊อต 7 ให้สัญญาณความถี่ 3.508 MHz
36	7M	เป็นขาสัญญาณความถี่มาตรฐาน 7.159 MHz
37	Q3	ให้สัญญาณความถี่มาตรฐาน 2.046 MHz
38	Φ1	เป็นขาความถี่ 1.023 MHz ของ CPU ใช้แทนขาสัญญาณนาฬิกา 6502 Φ1
39	USER 1	เป็นการยกเลิกการดีโค๊ดของไอซี H12 และ H2 ในช่วงแอดเดรส $ C000-$ C7FF เมื่อให้ลอจิก Ø
4Ø	Φ_0	ให้ความถี่เช่นเดียวกันกับขา Φ1 แต่มีเฟสตรงกันข้ามใช้แทนสัญญาณนาฬิกา 6502 Φ2
41	$\overline{DEV\ SEL}$	มีทุกสล๊อต แต่ละสล๊อตมีช่วงแอดเดรสที่ $CØXØ-$CØXF; X = N+8 เมื่อ N คือ เลขสล๊อต
42-49	D7-DØ	เป็นขาข้อมูลของสล๊อต
5Ø	+12 VOLT	สามารถจ่ายกระแสได้ถึง 250 mA

ตาราง 2.6 แสดงขาสัญญาณและหน้าที่ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

บล็อกไดอะแกรม และ การทำงานของระบบ



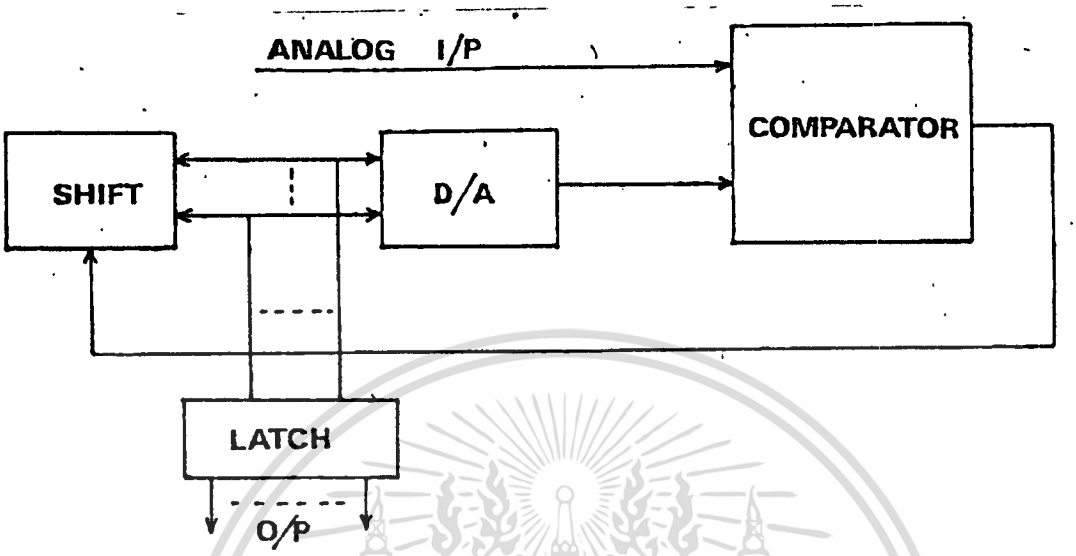
รูปที่ 3.1 แสดงการทำงานของระบบ

เมื่อระบบเริ่มการทำงาน CPU จะทำการสตาร์ท A/D ให้เตรียมพร้อมที่จะ CONVERT สัญญาณอินพุทซึ่งเป็นสัญญาณอนาล็อกที่เราป้อนให้ หลังจากที่ได้แปลงสัญญาณอนาล็อกเป็นดิจิตอลแล้ว CPU จะอ่านค่าที่แปลงเสร็จเก็บบันทึกไว้ จากนั้น CPU จะเตรียมการส่งข้อมูลที่แปลงเสร็จไปยังปลายทางอีกด้านหนึ่ง โดยเริ่มจากการที่ผู้ทำการติดต่อจะโทรศัพท์ไปยังสถานที่ที่ต้องการจะทราบข้อมูล เมื่อตกลงกันเรียบร้อยแล้วจึงทำการสับสวิทช์ที่โมเด็มให้ทำงาน (ในกรณีที่ส่งผ่านโมเด็ม) หลังจากนั้นเราก็จะสามารถส่งข้อมูลติดต่อสื่อสารกันได้

3.1 การแปลงสัญญาณอนาล็อกเป็นดิจิตอล

การแปลงสัญญาณอนาล็อกให้กลายเป็นค่าทางดิจิตอล เราสามารถทำได้ด้วยเทคโนโลยีทางด้าน ADC วิธีที่นิยมกันมากมี 2 วิธีด้วยกัน คือ แบบ RAMP และแบบ SUCCESSIVE APPROXIMATION ซึ่งทั้ง 2 วิธีมีหลักการทำงานที่คล้ายๆ กัน ดังรูป 3.2

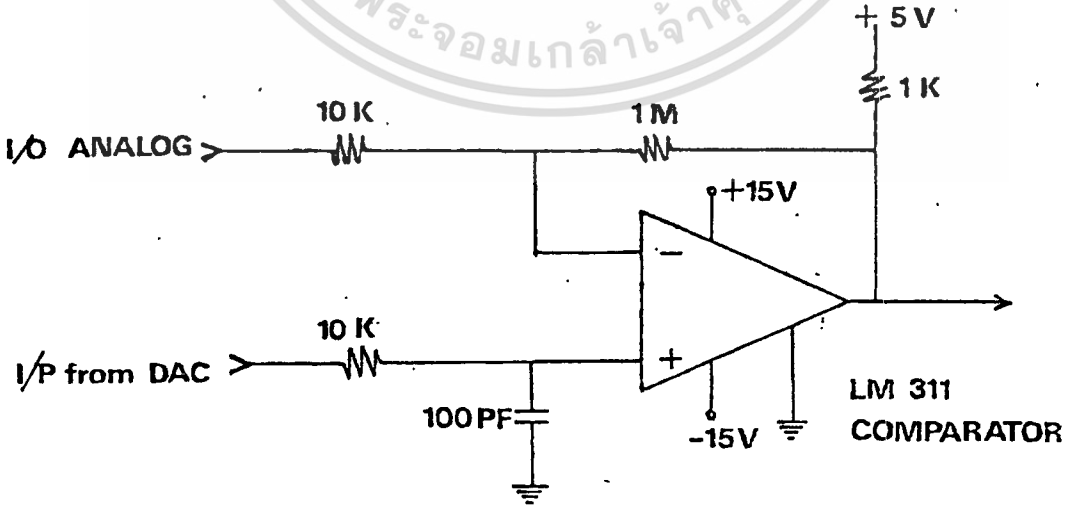
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการแปลง A/D

3.1.1 ADC ชนิด RAMP TYPE

การทำงานของ ADC ชนิดนี้ ใช้การเปรียบเทียบค่าระหว่างอนาล็อกอินพุทกับค่า RAMP FUNCTION จาก DAC ซึ่งวงจรหนึ่งที่ใช้ในการนี้สามารถดูได้จากรูปที่ 3.3



รูปที่ 3.3 การทำงานแบบ RAMP

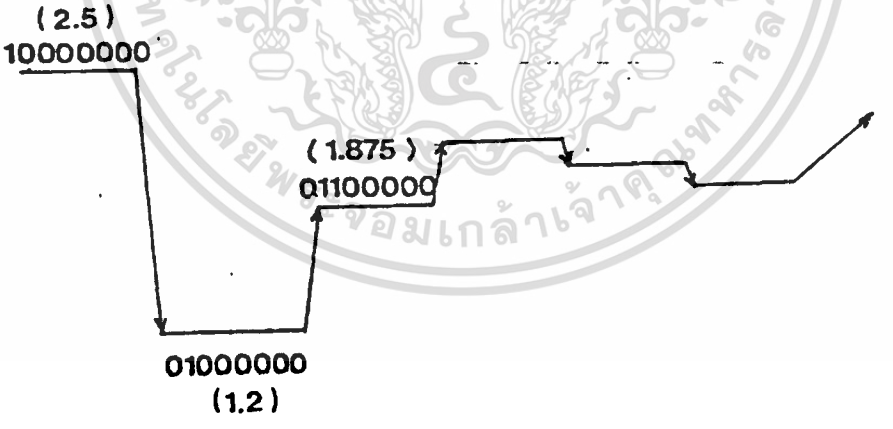
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรจะได้ว่าค่าเอาท์พุท จะมีค่าเป็น ø เมื่อค่า DAC มากกว่าอนาล็อกอินพุท และจะเป็น 1 เมื่อค่า DAC น้อยกว่าอนาล็อกอินพุท

การทำงานเริ่มจากค่า DAC=ø โวลต์ และจะเพิ่มขึ้นจนกระทั่งค่า DC เอาท์พุทโวลต์เตจ ของ DAC มากกว่า หรือ เท่ากับค่าอนาล็อกอินพุทโวลต์เตจ

3.1.2 ADC ชนิด SUCCESSIVE APPROXIMATION

การทำงานแบบ RAMP นั้น เป็นวิธีที่ง่ายแต่ว่าช้ามาก และเวลาที่ใช้ในการเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล จะขึ้นอยู่กับค่าของอนาล็อกอินพุท แต่วิธี SUCCESSIVE APPROXIMATION นี้ โดยทั่วไปจะสามารถทำงานได้เร็วกว่าแบบ RAMP ประมาณ 15 เท่า ในการทำงานอันเดียวกัน อย่างไรก็ดีหลักการทำงานของแบบ SUCCESSIVE APPROXIMATION ก็เหมือนๆ กันกับแบบ RAMP จะแตกต่างกันก็ตรงที่ว่าแบบ SUCCESSIVE APPROXIMATION จะเริ่มโดยเซ็ทค่า DAC = 2.5 โวลต์ ซึ่งเปรียบเทียบเป็นค่าทางดิจิตอล คือ 1øøøøøøø และจะทำการเปรียบเทียบค่าไปทีละบิท จากบิทที่มีค่าสูงสุดเป็นตัวเริ่มต้นดังรูป 3.4

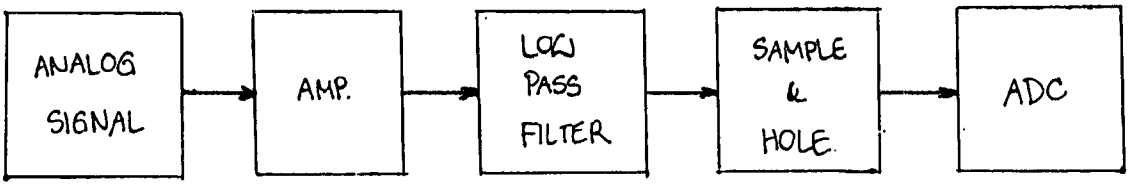


รูปที่ 3.4 การทำงานแบบ SUCCESSIVE APPROXIMATION

* คุณสมบัติของ ADC ø8øX

ไอซีเบอร์ ADC ø8øX เป็นซีมอสขนาด 8 บิท ชนิด SUCCESSIVE APPROXIMATION A/D CONVERTERS ซึ่งทำการแปลงสัญญาณอนาล็อกเป็นดิจิตอล โดยอาศัยขั้นศักดา ซึ่งสร้างจากวงจรเทียบเท่า 256R ซึ่งอยู่ภายในตัวไอซี

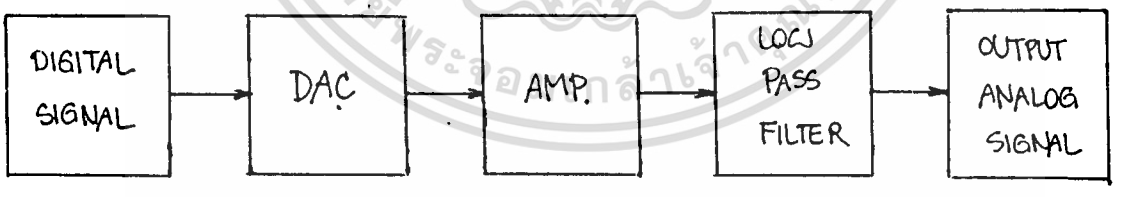
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 บล็อกไดอะแกรมการแปลง A/D

จากรูป 3.5 ก่อนที่จะทำการแปลงสัญญาณอนาล็อก เป็นดิจิตอล ต้องสร้างวงจรโลว์พาสฟิลเตอร์ เพื่อให้คัดออฟ (CUT OFF) ที่ความถี่ 3.4 KHz คือช่วงแบนด์วิท (BAND WIDTH) ของสัญญาณเสียง จากนั้นจะทำการแซมปลิ้ง (SAMPLING) เอาท์พุทจากโลว์พาสฟิลเตอร์ด้วยลอจิกจากสัญญาณขา EOC ของ ADC เอาท์พุทจากการแซมปลิ้งนี่เองที่จะนำไปแปลงเป็นดิจิตอล

3.2 การแปลงสัญญาณดิจิตอลเป็นอนาล็อก



รูปที่ 3.6 บล็อกไดอะแกรมการแปลง D/A

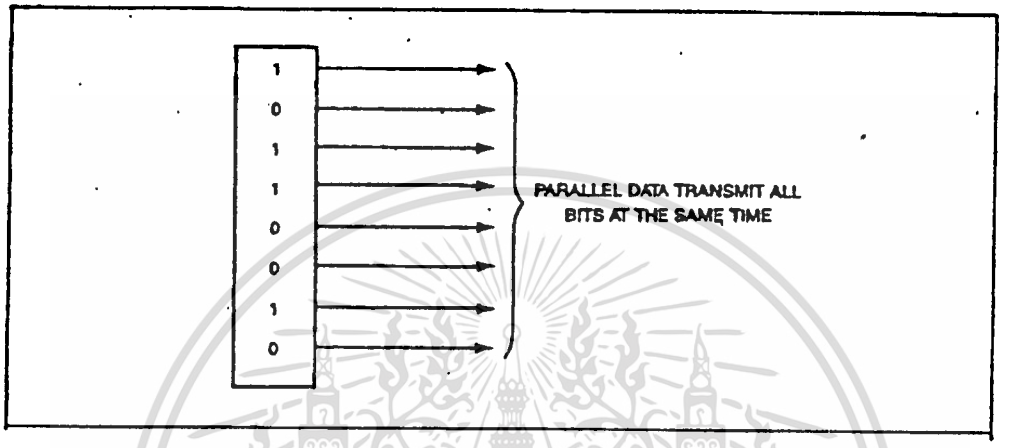
จากรูปข้อมูลขนาด 8 บิทจะถูกแปลงเป็นสัญญาณอนาล็อกโดยไอซีชนิด DAC เบอร์ MC1408 และก่อนที่สัญญาณจะผ่านโลว์พาสฟิลเตอร์ ก็จะถูกขยายโดยไอซีออปแอมป์ก่อน

สำหรับการทำงานของ DAC จะคล้ายกับ ADC กล่าวคือ ประกอบด้วยโครงสร้างของวงจรเทียบเท่า R/2R LADDER และเริ่มเปรียบเทียบจากค่า 10000000 ดูได้จากภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

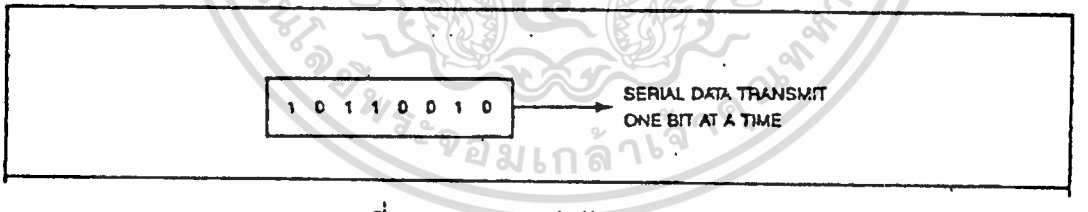
3.3 การรับส่งข้อมูลแบบอนุกรม (SERIAL COMMUNICATION)

โดยปกติข้อมูลทุก ๆ บิทจะถูกรับหรือส่งออกไปในเวลาเดียวกัน เช่น การอ่านหรือเขียนข้อมูลลงในหน่วยความจำ ซึ่งเราเรียกการรับส่งข้อมูลในลักษณะนี้ว่า "การรับส่งข้อมูลแบบขนาน (PARALLEL COMMUNICATION) "



รูปที่ 3.7 การรับส่งข้อมูลแบบขนาน

ส่วนการรับส่งข้อมูลในอีกรูปแบบหนึ่ง ซึ่งเป็นการรับส่งข้อมูลทีละบิทแทนที่จะทำการส่งข้อมูลพร้อมกันทุกบิทในเวลาเดียวกัน การรับส่งข้อมูลแบบนี้มีชื่อว่า "การรับส่งข้อมูลแบบอนุกรม (SERIAL COMMUNICATION) "



รูปที่ 3.8 การรับส่งข้อมูลแบบอนุกรม

สำหรับการรับส่งข้อมูลแบบขนานนั้น ถึงแม้ว่าจะมีความเร็วสูงกว่าแบบอนุกรมอยู่มากก็ตาม แต่ก็ต้องใช้จำนวนสายในการส่งผ่านข้อมูลเป็นจำนวนมากกว่าแบบอนุกรม ทำให้สิ้นเปลืองค่าใช้จ่ายในการวางสายโดยไม่จำเป็น และยังมีการลดทอนของสัญญาณมากกว่าแบบอนุกรมอีกด้วย ทำให้เกิดความผิดพลาดในการส่งผ่านข้อมูลได้ง่าย ดังนั้นในการส่งผ่านข้อมูลในระยะทางไกล ๆ เรามักจะเลือกใช้การรับส่งข้อมูลแบบอนุกรม เพื่อลดจำนวนของสายส่งซึ่งจะช่วยในการลดค่าใช้จ่ายในการวางสายลงได้อย่างมาก ถึงแม้ว่าการรับส่งข้อมูลแบบนี้จะมีความยุ่งยากและช้ากว่าการรับส่งข้อมูลแบบขนานอยู่บ้างก็ตาม

3.3.1 บอดเรท (BAUD RATE)

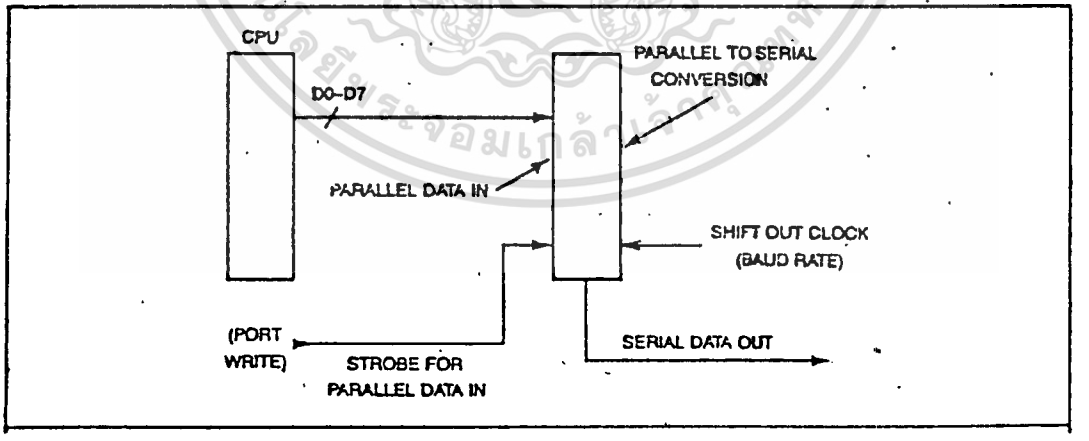
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่สำคัญมากสิ่งหนึ่งในการรับส่งข้อมูลแบบอนุกรมนี้ก็คือ ความถี่ที่ใช้ในการส่งข้อมูลซึ่งจะต้องสัมพันธ์กันระหว่างอุปกรณ์ที่ทำการรับและส่งข้อมูล และความถี่ที่ใช้นี้มีชื่อเรียกว่า " บอดเรท " ซึ่งมีความหมายถึง " อัตราการรับส่งข้อมูลเป็นจำนวนบิทใน 1 วินาที " ถ้าหากว่าเครื่องส่งใช้บอดเรทที่ไม่สัมพันธ์กับเครื่องรับแล้ว ก็จะทำให้การรับส่งข้อมูลเกิดผิดพลาดขึ้นได้ โดยทั่ว ๆ ไปค่าของบอดเรทนั้นจะใช้ค่าต่าง ๆ ดังต่อไปนี้คือ 110, 150, 300, 1200, 2400, 4800 และ 9600

3.3.2 การเปลี่ยนข้อมูลจากแบบขนานเป็นข้อมูลแบบอนุกรม

โดยทั่วไปแล้ว การรับส่งข้อมูลภายในระบบมักจะเป็นการรับส่งข้อมูลแบบขนาน เนื่องจากมีความเร็วในการส่งผ่านข้อมูลที่สูงกว่าแบบอนุกรมมาก และยังมีความยุ่งยากน้อยกว่าอีกด้วย ดังนั้นในการรับส่งข้อมูลในระยะทางไกลๆ ที่จำเป็นจะต้องใช้การส่งผ่านข้อมูลแบบอนุกรม จึงจำเป็นที่จะต้องทำการเปลี่ยนรูปแบบของข้อมูลจากแบบขนานไปเป็นแบบอนุกรม ก่อนที่จะทำการส่งข้อมูลไปตามสายส่งสำหรับหลักการง่าย ๆ ที่ใช้ในการเปลี่ยนรูปแบบข้อมูลนั้นมีขั้นตอนดังต่อไปนี้

1. ทำการเก็บข้อมูลแบบขนาน (ในที่นี้มีจำนวน 8 บิท) ไว้ใน ชิฟท์รีจสเตอร์
2. เลื่อนข้อมูลทั้ง 8 บิทออกไปให้กับอุปกรณ์รับข้อมูลทีละบิท โดยที่จะทำการส่งข้อมูลแต่ละบิทออกไปด้วยอัตราเดียวกับบอดเรท ที่กำหนดไว้รูปที่ 3.9



รูปที่ 3.9 การเปลี่ยนข้อมูลแบบขนานเป็นอนุกรม

บล็อกไดอะแกรมของการทำงานทั้ง 2 ขั้นตอน คือ ข้อมูลแบบขนานนั้นจะถูกส่งจาก CPU ให้เก็บในชิฟท์รีจิสเตอร์ จากนั้นจึงทำการเลื่อนข้อมูลออกทีละบิทด้วยอัตราของบอดเรทที่กำหนด (โดยเลื่อนบิท D0 ก่อนและ D7 เป็นบิทสุดท้าย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 หลักการเบื้องต้นของการรับส่งข้อมูลแบบอนุกรม

สำหรับหลักการเบื้องต้นของการรับส่งข้อมูลแบบอนุกรม สามารถที่จะสรุปเป็นข้อ ๆ ได้ดังนี้

1. ข้อมูลแบบขนานจากระบบจะถูกเปลี่ยนให้เป็นข้อมูลแบบอนุกรมเพื่อเตรียมที่จะส่งออกไปให้กับส่วนรับข้อมูล

2. ข้อมูลจะถูกส่งออกไปด้วยอัตราคงที่ค่าหนึ่ง ซึ่งเรียกว่า " บอดเรท " คือ ถ้าทำการส่งข้อมูลด้วยอัตรา 12ØØ บอด ก็แสดงว่าเป็นการส่งข้อมูลด้วยอัตรา 12ØØ บิท ต่อ 1 วินาที

3. ข้อมูลอนุกรมจะถูกส่งออกไปทีละบิท โดยทำการส่งบิท DØ เป็นบิทแรกและบิท D7 เป็นบิทสุดท้าย

4. ในขณะที่ยังไม่มีการส่งข้อมูลเข้าไปในสายส่ง สายส่งจะถูกทำให้อยู่ในสภาวะลอจิก "1" และเราเรียกสภาวะนี้ว่า " มาร์คกิ้ง (MARKING) "

5. อุปกรณ์ส่งข้อมูลจะเพิ่มข้อมูลอีก 1 บิท เข้าไปหน้าบิท DØ ของข้อมูลที่จะส่งให้กับเครื่องรับ บิทที่เพิ่มเข้าไปนี้เรียกว่า " สตาร์ทบิท (START BIT) "

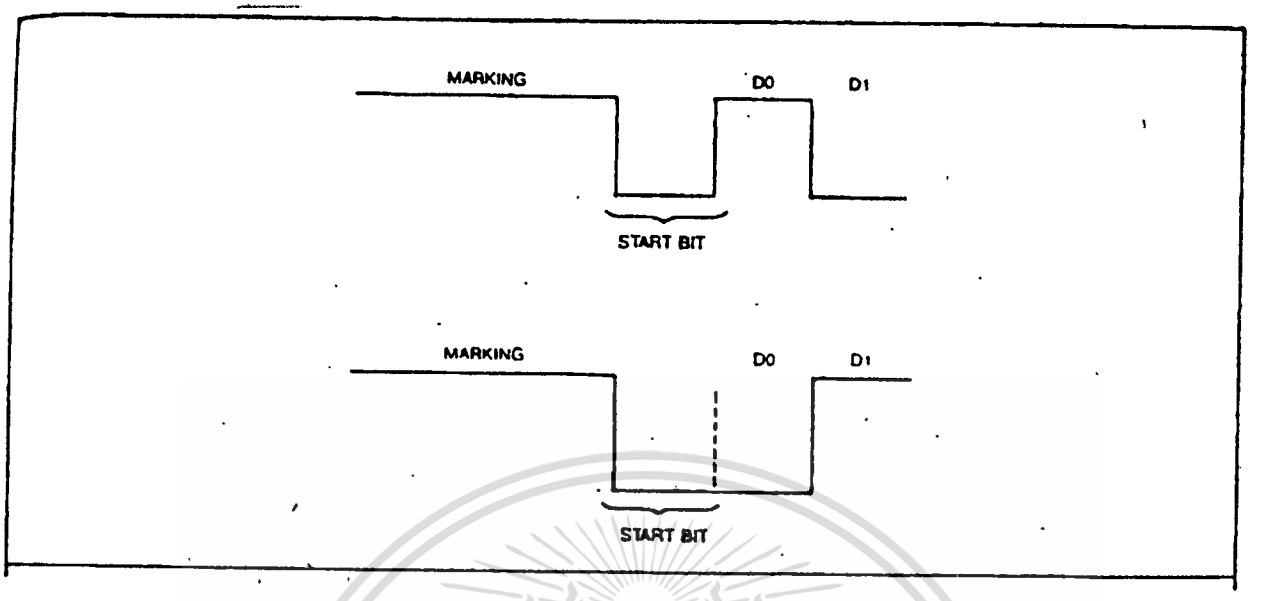
6. อุปกรณ์ส่งข้อมูลจะทำการเพิ่มพาริตี้บิท (PARITY BIT) เข้าไปหลังบิท D7 ของข้อมูล เพื่อใช้ในการตรวจสอบความผิดพลาดของข้อมูลที่เครื่องรับ

7. สำหรับบิทสุดท้ายที่ถูกเพิ่มเข้าไปหลังพาริตี้บิท เรียกว่า " สต๊อปบิท (STOP BIT " ซึ่งอาจจะมีจำนวน 1,1.5 หรือ 2 บิทก็ได้ และลอจิกของบิทนี้จะเป็นลอจิกเดียวกับลอจิกของ มาร์คกิ้ง

3.3.4 สตาร์ทบิท

ในการส่งผ่านข้อมูลแบบอนุกรมนี้ เราจำเป็นที่จะต้องทำให้อุปกรณ์ที่จะรับข้อมูลทราบว่า ข้อมูลที่ส่งมานั้นเริ่มต้นที่จุดใด ดังนั้นเราจึงจำเป็นที่จะต้องเพิ่มข้อมูล 1 บิทลงไปกรอนหน้าข้อมูลจริง (ACTUAL DATA) ที่จะทำการส่ง (การส่งอนุกรมจะส่งบิท DØ เป็นบิทแรก และบิท D7 เป็นบิทสุดท้าย) คือ ทำการเพิ่มบิทนี้ลงไปหน้าบิท DØ นั่นเอง และเรียกบิทนี้ว่า " สตาร์ทบิท " รูปที่ 3.1Ø

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1Ø แสดงถึงสตาร์ทบิท

เมื่ออุปกรณ์ที่จะส่งข้อมูลยังไม่ได้ทำการส่งข้อมูลใด ๆ ออกมานั้น สายส่งจะอยู่ในสภาวะที่เรียกว่า " มาร์คกิ้ง " ซึ่งเป็นสภาวะที่ไม่มีการรับส่งข้อมูลใด ๆ เกิดขึ้น ในกรณีนี้เราให้มาร์คกิ้งของสายส่งเป็นลอจิก "1" . สตาร์ทบิทที่จะเพิ่มเข้าไปนี้จะมีลอจิกที่ตรงข้ามกับลอจิกของมาร์คกิ้ง ดังนั้นในกรณีนี้ สตาร์ทบิทจะมีลอจิกเป็น "Ø" สำหรับสตาร์ทบิทนี้จะมีความกว้างเท่ากับ 1 บิทข้อมูล

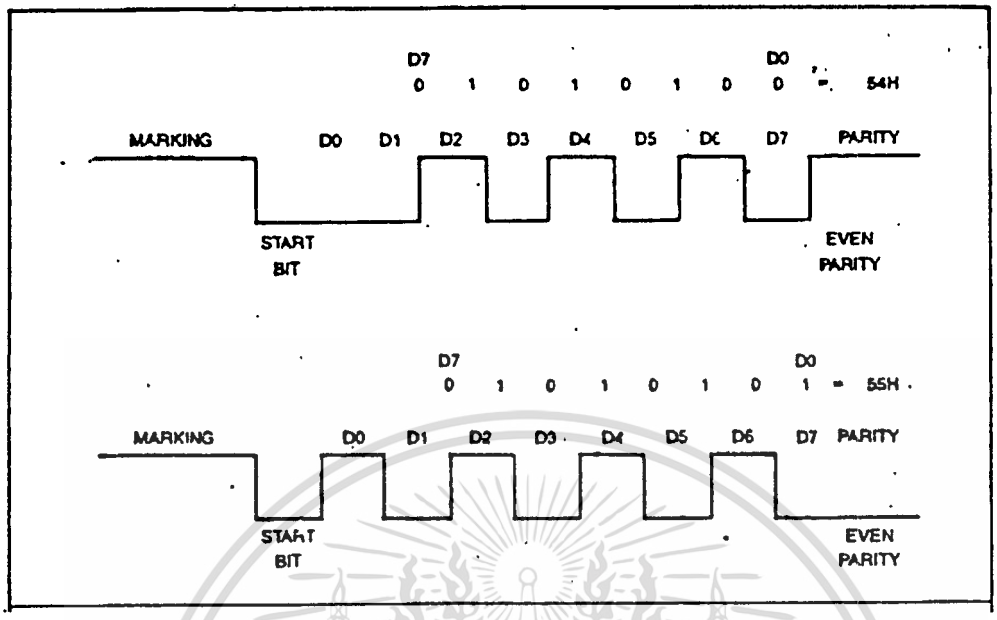
3.3.5 **พาริตี้บิท**

สำหรับบิทนี้จะทำหน้าที่ในการบอกให้ส่วนรับข้อมูลทราบว่า ข้อมูลที่ได้รับเข้ามานั้นมีความถูกต้องเหมือนกับข้อมูลที่ถูกส่งออกมาหรือไม่

พาริตี้บิทนี้จะถูกส่งออกมาโดยอุปกรณ์ส่งข้อมูล ซึ่งบิทนี้จะเป็น "1" หรือ "Ø" นั้นขึ้นอยู่กับข้อมูลที่ส่งออกมา (DØ-D7) ว่ามีจำนวนบิทที่เป็น "1" เป็นจำนวนคู่หรือคี่ และยังขึ้นกับอุปกรณ์รับส่งข้อมูลด้วยว่าถูกออกแบบ (โปรแกรม) ไว้ให้รับส่งพาริตี้บิทในลักษณะของพาริตี้คู่ หรือพาริตี้คี่ อีกด้วย รูป 3.11

สิ่งสำคัญอีกสิ่งหนึ่งก็คือ ถ้าอุปกรณ์ส่งข้อมูลทำการส่งในลักษณะพาริตี้คู่ หรือ คี่ ก็ตาม ส่วนรับข้อมูลก็จะต้องทำการรับในลักษณะพาริตี้เดียวกับอุปกรณ์ส่งข้อมูลด้วย เช่นในกรณีที่อุปกรณ์ส่งข้อมูลทำการส่งข้อมูลในลักษณะของพาริตี้คู่ อุปกรณ์รับข้อมูลก็จะต้องทำการรับข้อมูลในลักษณะของพาริตี้คู่ด้วย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงถึงพาริตี้บิท

3.3.6 สต๊อปบิท

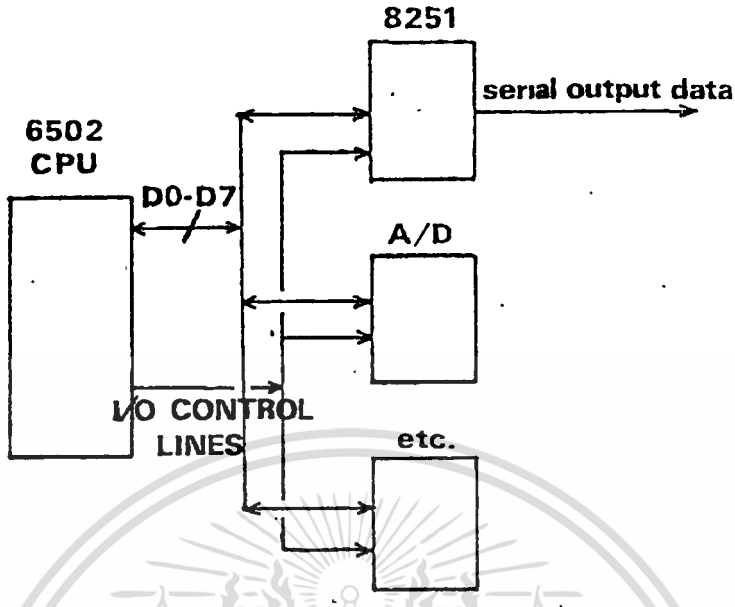
สำหรับบิทสุดท้ายที่เพิ่มเข้าไปนี้ จะใช้ในการตรวจสอบจุดสิ้นสุดของข้อมูล บิทนี้จะถูกเพิ่มเข้าไปหลังพาริตี้บิท ถ้าอุปกรณ์รับตรวจข้อมูลไม่พบบิทนี้ก็แสดงว่าข้อมูลที่ได้รับเข้ามานั้นมีความผิดพลาดเกิดขึ้น สำหรับ สต๊อปบิท นี้อาจมีจำนวนบิทเป็น 1,1.5 หรือ 2 บิทก็ได้

3.4 8251 USART

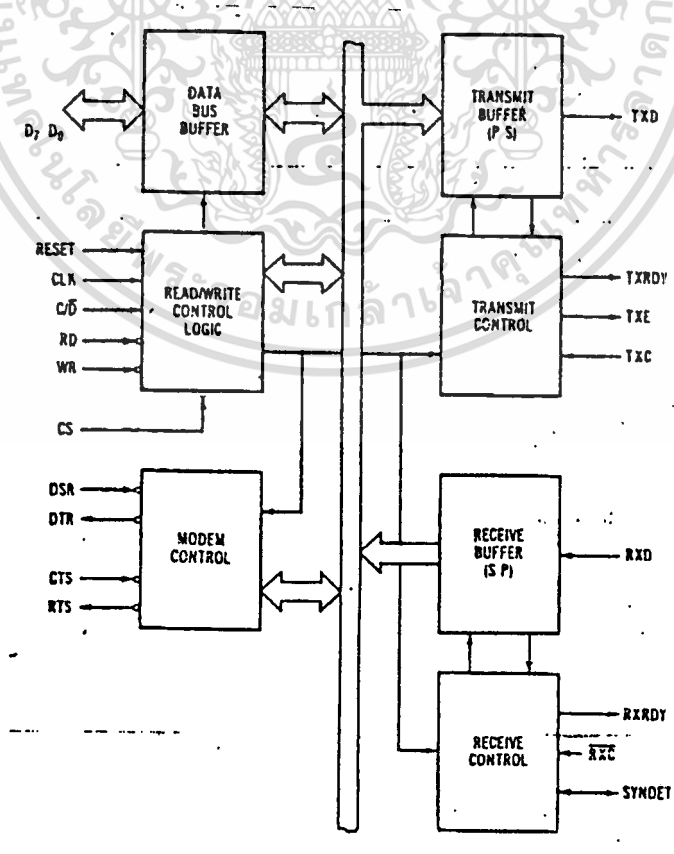
ดังที่ได้กล่าวไว้แล้วว่า การส่งข้อมูลแบบอนุกรมไปตามสายส่งได้นั้นเราจำเป็นที่จะต้องทำการเปลี่ยนรูปแบบของข้อมูลเสียก่อน แต่โดยวิธีที่กล่าวมาแล้วนั้นเป็นวิธีที่ยังมีประสิทธิภาพไม่ดีพอ เนื่องจากว่า CPU จะต้องทำการรับส่งข้อมูลเองในช่วงเวลาที่เหมาะสม และในส่วนรับข้อมูล CPU ก็จะต้องทำการตรวจสอบความผิดพลาดของข้อมูลที่รับเข้ามาเองทุกอย่าง ทำให้เกิดความยุ่งยากในการออกแบบ และในส่วนรับข้อมูลก็ยังจะต้องมีอุปกรณ์ที่ใช้ในการเปลี่ยนข้อมูลกลับมาเป็นแบบขนานอีก ทำให้เกิดความสิ้นเปลือง ดังนั้นจึงจำเป็นที่จะต้องใช้อุปกรณ์ที่มีความสามารถที่จะเป็นได้ทั้งอุปกรณ์รับและส่งข้อมูลในตัวเดียวกัน ดังนั้นจึงจะกล่าวถึงลักษณะและวิธีการใช้งานของไอซี 8251 USART (UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER/TRANSMITTER) ซึ่งเป็นพอร์ทที่ใช้ในการรับส่งข้อมูลแบบอนุกรมที่มีประสิทธิภาพมากตัวหนึ่ง

จากรูป 3.12 จะเห็นว่าเราสามารถนำ 8251 ไปเชื่อมต่อกับ 6502 ได้ในลักษณะเดียวกับ SHIP SUPPORT เบอร์อื่น ๆ ของ 6502 และในรูป 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การเชื่อมต่อ 8251 กับ CPU

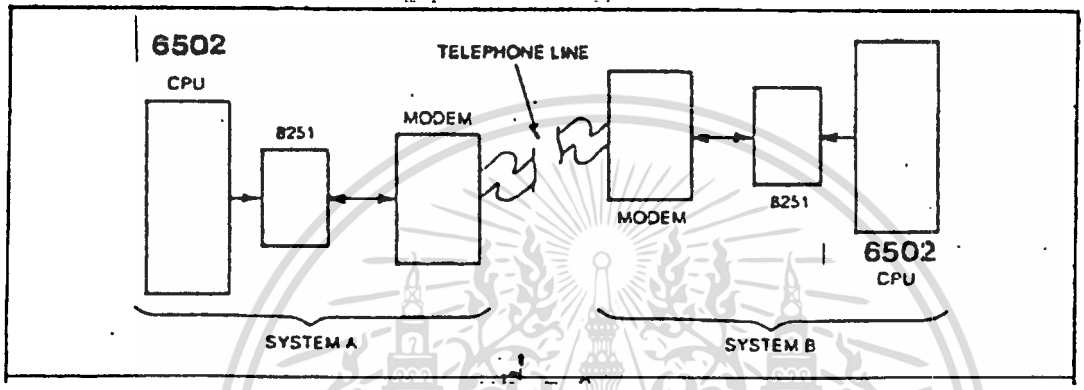


รูปที่ 3.13 โครงสร้างของ 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแสดงถึง บล็อกไดอะแกรมของ 8251 สำหรับส่วนแรกก็คือ DATA BUS BUFFER ส่วนนี้ 8251 จะใช้ในการเชื่อมต่อระหว่าง 8251 กับบัสข้อมูลของ 6502 ส่วนต่อไปก็คือ READ/WRITE CONTROL LOGIC ซึ่งจะทำหน้าที่ในการควบคุมการรับส่งภายในของ 8251 ให้เป็นไปอย่างถูกต้อง

สำหรับ MODEM CONTROL นั้น จะใช้ในการติดต่อระหว่าง 8251 กับ MODEM รูปที่ 3.14 จะแสดงถึงบล็อคไดอะแกรม ของการใช้งานของ 8251 ร่วมกับ MODEM



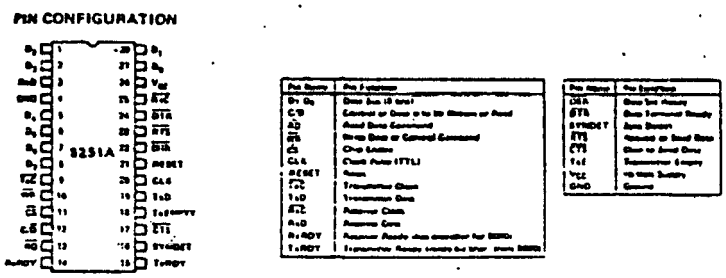
รูปที่ 3.14 การใช้งาน 8251 ร่วมกับ MODEM

ส่วนต่อไปก็คือ TRANSMIT BUFFER (P-S) และ TRANSMIT CONTROL (P-S;SERIAL TO PARALLEL CONVERSION) ซึ่งใช้ในการส่งและควบคุมการส่งข้อมูลไปตามสายส่ง

สำหรับส่วนสุดท้ายก็คือ RECIEVE BUFFER (S-P;SERIAL TO PARALLEL CONVERSION ซึ่งทำหน้าที่ในการรับและควบคุมการรับข้อมูลของ 8251

3.4.1 การจัดเรียงขาและหน้าที่

8251 เป็นไอซีขนาด 28 ขา ซึ่งได้แสดงไว้ในรูปที่ 3.15 และสามารถที่จะแบ่งขาของ 8251 ออกเป็นกลุ่มๆ ได้ดังนี้คือ



รูปที่ 3.15 แสดงการจัดเรียงขาและหน้าที่ของ 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กลุ่มที่ใช้ในการติดต่อกับ CPU

1.1 DØ-D7: ใช้ในการาติดต่อกับ DATA BUS ของ CPU โดยตรงซึ่งจะทำหน้าที่ในการรับส่งข้อมูลและคำสั่งต่าง ๆ ระหว่าง 8251 กับ CPU

1.2 RESET: 8251 จะถูกรีเซ็ทเมื่อขานี้ได้รับลอจิก "1" ซึ่งเราอาจจะต่อมาจากขา RESET ของ 6502 โดยผ่าน INVERTER ก่อนก็ได้

1.3 CLK (CLOCK): ใช้ในการควบคุมช่วงเวลาการทำงานภายในของ 8251 สำหรับการใช้งานนั้นจะต่อเข้าโดยตรงกับระบบ อย่างไรก็ตามสัญญาณที่ขา CLK นี้ไม่เกี่ยวข้องกับอัตราการรับส่งข้อมูลหรือ บอดเรท แต่อย่างใด

1.4 $\overline{RD}$: เมื่อขานี้ได้รับลอจิก "Ø" 8251 จะทำการส่งข้อมูลแบบขนานจาก DATA BUS เพื่อส่งให้กับ CPU

1.5 $\overline{WR}$: เมื่อขานี้ได้รับลอจิก "Ø" 8251 จะทำการรับข้อมูลแบบขนานจาก DATA BUS ของระบบ

1.6 C/$\overline{D}$ (CONTROL/DATA): ขา C/$\overline{D}$ นี้จะใช้ในการทำให้ 8251 ทราบว่า CPU ต้องการที่จะติดต่อกับ CONTROL REGISTER หรือ DATA REGISTER โดยที่ ถ้าขานี้ได้รับลอจิก "1" ก็แสดงว่า CPU ต้องการที่จะติดต่อกับ CONTROL REGISTER แต่ถ้าได้รับลอจิก "Ø" ก็แสดงว่า CPU ต้องการที่จะติดต่อกับ DATA REGISTER

1.7 $\overline{CS}$ (CHIP SELECT): ในกรณีที่ขานี้ได้รับลอจิก "Ø" ก็จะเป็นการ ENABLE 8251 โดยทั่วไปแล้วสัญญาณที่ขานี้จะ ได้มาจากการถอดรหัสพอร์ทแอดเดรส ดังที่ใช้กับ CHIP SUPPORT อื่นๆ

2. กลุ่มที่ใช้ในการติดต่อกับ MODEM

2.1 $\overline{DSR}$ (DATA SET READY): ขานี้เป็นขาที่ใช้ในการรับสัญญาณจากอุปกรณ์ภายนอก ซึ่ง CPU สามารถที่จะตรวจสอบสัญญาณที่ขานี้ได้ โดยการอ่านค่าในรีจีสเตอร์สถานะ และระดับของสัญญาณที่ขานี้จะใช้ในการแสดงว่าอุปกรณ์ภายนอกพร้อมที่จะทำการติดต่อด้วยหรือยัง

2.2 $\overline{DTR}$ (DATA TERMINAL READY): ขานี้เป็นขาเอาท์พุทที่ใช้ในการบอกให้อุปกรณ์ภายนอกทราบว่า CPU พร้อมที่จะทำการติดต่อด้วย

2.3 $\overline{CTS}$ (CLEAR TO SEND): ขานี้เป็นขาอินพุทที่ใช้ในการทำให้ 8251 เริ่มทำการส่งข้อมูลได้ สิ่งที่ต้องระวังในการใช้งานขานี้ก็คือ เมื่อไม่ได้ใช้งานขานี้จะต้องถูกต่อเข้ากับลอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จิก "ø" ถ้าไม่เช่นนั้น 8251 จะทำการส่งข้อมูลไม่ได้

2.4 $\overline{\text{RTS}}$ (READY TO SEND): ขานี้เป็นเอาท์พุทที่ CPU จะเป็นผู้ควบคุมสัญญาณที่ขานี้เอง (ขา $\overline{\text{DTR}}$ ก็จะถูกควบคุมโดย CPU เช่นกัน)

3. กลุ่มที่ใช้ในการส่งข้อมูล

3.1 TxD (TRANSMIT DATA OUTPUT): เป็นขาที่ใช้ส่งข้อมูลไปตามสายส่ง

3.2 TxC (TRANSMIT BAUD RATE CLOCK): ขานี้เป็นขาที่ใช้ในการส่งสัญญาณคล็อกที่ใช้ในการส่งข้อมูล ซึ่งก็คือความถี่ที่ใช้ในการกำหนด บอดเรท นั่นเอง โดยปกติแล้วจะต้องช้ากว่าสัญญาณคล็อกของระบบไม่น้อยกว่า 3ø เท่า

3.3 TxRDY : ขานี้จะใช้ในการทำให้ CPU ทราบว่า 8251 พร้อมที่จะรับข้อมูลจาก CPU เพื่อที่จะทำการส่งต่อไปแล้วหรือยัง และขานี้อาจจะนำไปใช้ในการขออินเตอร์รัพท์ก็ได้

3.4 TxEMPTY : ขานี้จะใช้ในการแสดงว่าข้อมูลที่ CPU ส่งให้กับ 8251 นั้นได้ถูกส่งออกไปให้อุปกรณ์อื่นๆ หมดแล้ว โดยที่ 8251 จะทำให้ขานี้เป็น "1" และเมื่อ CPU ทำการส่งข้อมูลชุดต่อไปให้กับ 8251 ขา TxEMPTY ก็จะเป็น "ø" จนกว่า 8251 จะทำการส่งข้อมูลนี้ออกไปหมด 8251 ก็จะทำให้ขานี้กลับเป็น "ø" อีกครั้ง

4. กลุ่มที่ใช้ในการรับข้อมูล

4.1 RxD : ใช้ในการรับข้อมูลแบบอนุกรมจากสายส่ง

4.2 RxC : เป็นขาที่ใช้ในการรับสัญญาณคล็อกที่ใช้ในการรับข้อมูลโดยปกติแล้วจะทำการต่อเข้ากับ TxC โดยตรง

4.3 RxRDY : จะใช้ในการแสดงว่า 8251 พร้อมที่จะส่งข้อมูลให้กับ CPU และขานี้อาจจะใช้ในการขออินเตอร์รัพท์ได้เช่นเดียวกับขา TxRDY

4.4 SYNDET : ขานี้ใช้ในการรับส่งข้อมูลแบบ SYNCHRONOUS เท่านั้น (8251 สามารถที่จะทำการรับส่งข้อมูลได้ทั้งแบบ SYNCHRONOUS และแบบ ASYNCHRONOUS ซึ่งจะได้กล่าวถึงต่อไป) โดยที่เราสามารถที่จะโปรแกรมให้ขานี้เป็นอินพุทหรือเอาท์พุทก็ได้ โดยที่เมื่อขา SYNDET นี้ถูกโปรแกรมเป็นเอาท์พุทนั้นขา SYNDET จะให้ลอจิก "1" เมื่อ 8251 สามารถที่จะตรวจจับ SYNC CHARACTER ได้และจะให้ลอจิก "ø" เมื่อ CPU ทำการอ่านรีจีสเตอร์สถานะสำหรับขา SYNDET นี้จะให้ลอจิก "1" ในอีกกรณีหนึ่งก็คือ เมื่อ8251 ได้รับข้อมูลจากสายส่งเป็น "ø" หมด ตั้งแต่ สตาร์ทบิท จนถึง สต๊อปบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ขา SYNDET ถูกโปรแกรมให้เป็นอินพุทนั้น ถ้าขานี้ได้รับสัญญาณขอบขาขึ้น (สัญญาณเปลี่ยนจากลอจิก "Ø" เป็น "1") 8251 ก็จะถือว่าข้อมูลขา RxD เป็นข้อมูลทันที และเราสามารถที่จะทำให้ลอจิกที่ขานี้กลับเป็น "Ø" ได้ในสัญญาณ RxC ลูกต่อไป

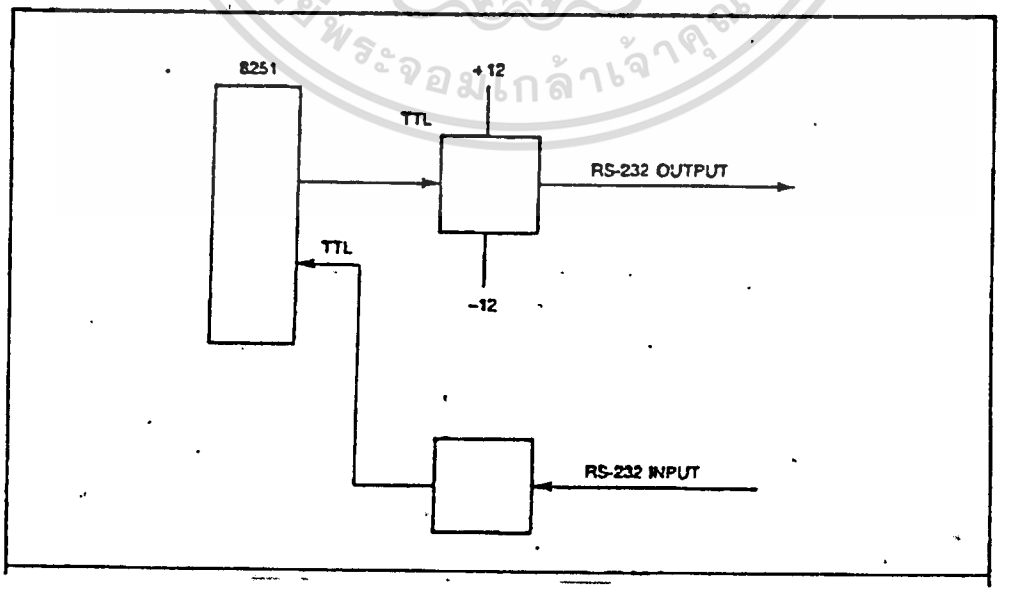
5. กลุ่มไฟเลี้ยงของ 8251

8251 ใช้ไฟเลี้ยงเพียงชุดเดียว คือ +5V กับ GND เท่านั้น ดังนั้นขาไฟเลี้ยงของ 8251 จึงมีเพียง 2 ขา คือ Vcc กับ GND

3.4.2 การเชื่อมต่อกับสายส่งข้อมูล

การรับส่งข้อมูลแบบอนุกรมนั้น นิยมนำไปใช้ในการรับส่งข้อมูลในระยะทางไกล เพราะสามารถที่จะลดค่าใช้จ่ายในการวางสายลงได้มาก แต่อย่างไรก็ตามระดับสัญญาณที่ใช้ในวงจร (+5V) นั้นไม่สามารถที่จะส่งไปได้ไกลนัก ดังนั้น ก่อนการส่งข้อมูลไปในสายส่งจะต้องทำการเปลี่ยนระดับแรงดันของสัญญาณใหม่เพื่อให้สามารถที่จะส่งสัญญาณได้ไกลขึ้น แต่การที่จะเปลี่ยนระดับของสัญญาณไปเป็นเท่าใดนั้น ขึ้นอยู่กับมาตรฐานที่ใช้ในการส่ง

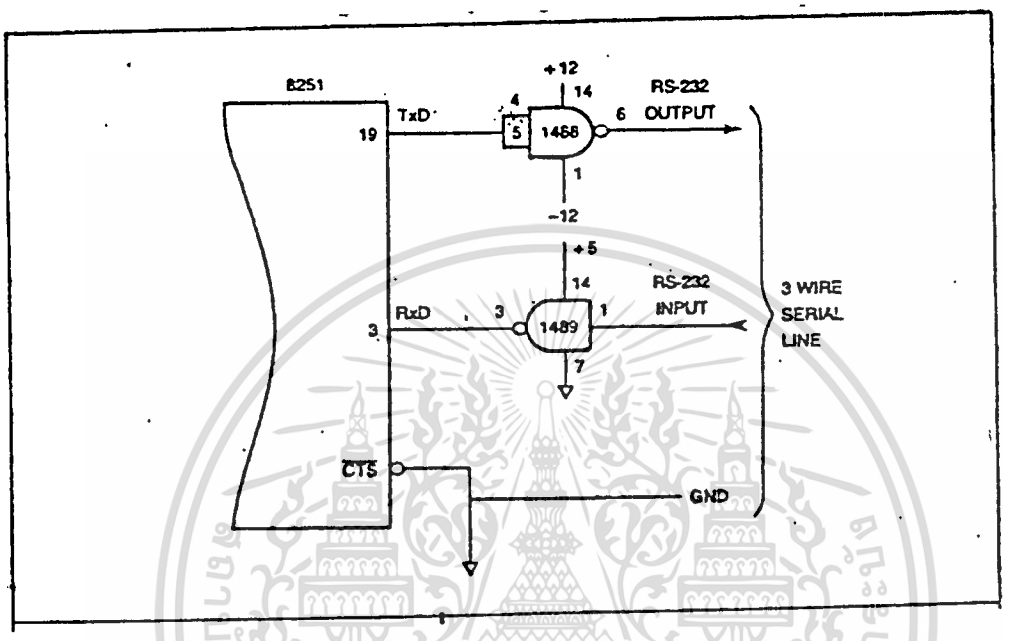
สำหรับมาตรฐาน RS-232 นี้จะใช้ระดับแรงดันในสายส่งประมาณ +/-12V รูปที่ 3.16 จะแสดงบล็อกไดอะแกรมของการเปลี่ยนระดับแรงดันของสัญญาณจากระดับสัญญาณที่ใช้กับอุปกรณ์พวก TTL ไปเป็นระดับสัญญาณที่ใช้กับมาตรฐาน RS-232 ในการส่งข้อมูล และการเปลี่ยนจากระดับของ RS-232 ไปเป็นระดับสัญญาณ TTL ในการรับข้อมูลจากสายส่ง



รูปที่ 3.16 การเปลี่ยนระดับศักดาจาก TTL เป็น RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้ในการทำหน้าที่ทั้งสองอย่างนี้ก็คือ ไอซี เบอร์ MC1488 ซึ่งทำหน้าที่ในการเปลี่ยนระดับของสัญญาณจาก TTL ไปเป็น RS-232 และ MC1489 ซึ่งทำหน้าที่ในการเปลี่ยนระดับของสัญญาณจาก RS-232 ไปเป็น TTL



รูปที่ 3.17 แสดงการต่อ MC1488 และ MC1489 ร่วมกับ 8251

ในรูปที่ 3.17 จะแสดงวิธีในการนำเอา MC1488 และ MC1489 มาใช้งานร่วมกับ 8251 และจะเห็นว่าขา $\overline{CTS}$ จะถูกต่อกับลอจิก "Ø" ด้วย และจะเห็นว่าเราใช้สายส่งเพียง 3 เส้น คือ TxD , RxD และ GND เท่านั้น

3.4.3 การโปรแกรม 8251

การรับส่งข้อมูลบน 8251 นั้นสามารถที่จะทำได้ 2 แบบ คือ แบบ SYNCHRONOUS และแบบ ASYNCHRONOUS ซึ่งมีลักษณะความแตกต่างในการรับส่งข้อมูลแบบนี้ คือ

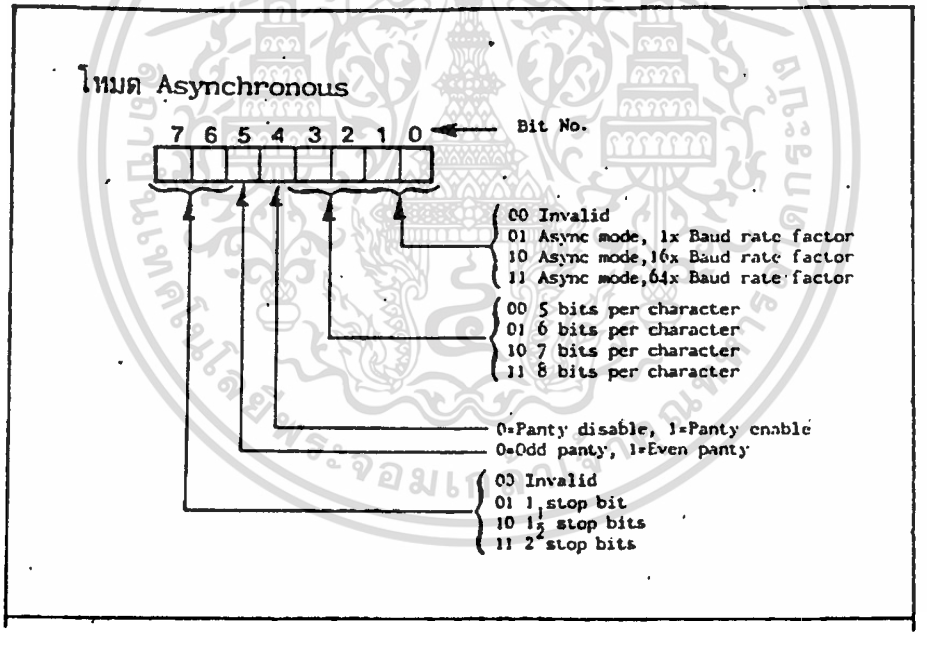
การรับส่งข้อมูลแบบ ASYNCHRONOUS นั้น จะทำการส่ง สตาร์ทบิท และ สต๊อปบิท ออกไปพร้อมกับข้อมูลด้วย ส่วนแบบ SYNCHRONOUS นั้น จะไม่มีการส่ง สตาร์ทบิท และ สต๊อปบิท แต่จะทำการส่ง SYNC CHARACTER แทนซึ่ง 8251 จะทำการส่ง SYNC CHARACTER ออกไปเรื่อยๆ ในทันทีที่ CPU สั่งให้ 8251 ทำการส่งข้อมูลได้จนกว่า CPU จะทำการส่งข้อมูลให้กับ 8251 เพื่อทำการส่งไปในสายส่ง

สำหรับการทำงานใน ASYNCHRONOUS MODE นั้น เมื่อ8251 รับข้อมูลเข้ามาแล้วก็จะทำการตรวจสอบ สตาร์ท , สต๊อป และ พาริตี้บิท ถ้าเกิดความผิดพลาดขึ้นก็จะไปแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ในรีจีสเตอร์สถานะ ส่วนการรับข้อมูลใน SYNCHRONOUS MODE นั้น 8251 จะพยายาม SYNC กับ SYNC CHARACTER ที่เครื่องส่ง ส่งออกมาโดยการเลื่อนข้อมูลเข้ามาทีละบิทเพื่อนำมาเทียบกับ SYNC CHARACTER ที่ได้โปรแกรมไว้ว่าตรงกันหรือไม่ และเมื่อ 8251 ตรวจจับ SYNC CHARACTER ได้แล้ว (SYNC CHARACTER ที่ส่งมาตรงกับที่ได้โปรแกรมไว้) ก็จะทำให้ขา SYNDET เป็น "1" เพื่อให้ CPU รับทราบ (ซึ่งอาจจะใช้วิธีการอินเตอร์รัพท์ก็ได้) การที่จะให้ 8251 ทำการตรวจจับ SYNC CHARACTER นั้นจะต้องทำการโปรแกรมให้ 8251 ทำงานใน HUNT MODE และ 8251 จะออกจาก HUNT MODE เอง เมื่อทำการ SYNC ได้

รูปที่ 3.18 และ รูปที่ 3.19 จะแสดงการจัดเรียงบิทใน CONTROL WORD (MODE WORD) ทั้งใน ASYNCHRONOUS และ SYNCHRONOUS MODE ซึ่ง MODE WORD นี้จะถูกส่งให้กับรีจีสเตอร์ควบคุม



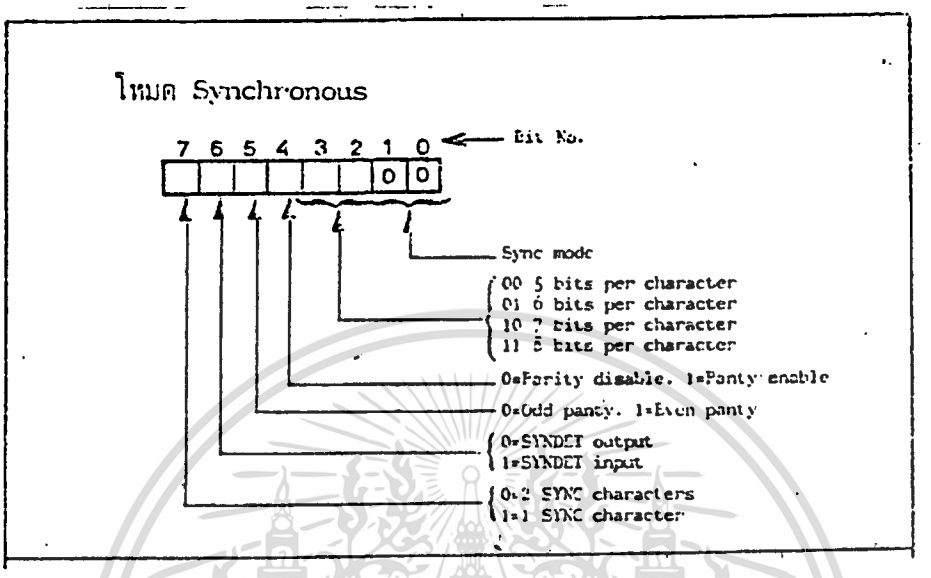
รูปที่ 3.18 MODE WORD ใน ASYNCHRONOUS

สำหรับโหมด ASYNCHRONOUS นั้น หลังจากที่ได้ส่ง MODE WORD ให้กับ 8251 แล้ว ข้อมูลไบท์ต่อไปที่ถูกส่งให้กับรีจีสเตอร์ควบคุมจะถือว่าเป็น COMMAND WORD ทั้งสิ้น จนกว่า 8251 จะได้รับการรีเซ็ทอีก จึงจะถือว่าข้อมูลที่ส่งมาให้กับรีจีสเตอร์ควบคุมนั้นเป็น MODE WORD รูปที่ 3.20

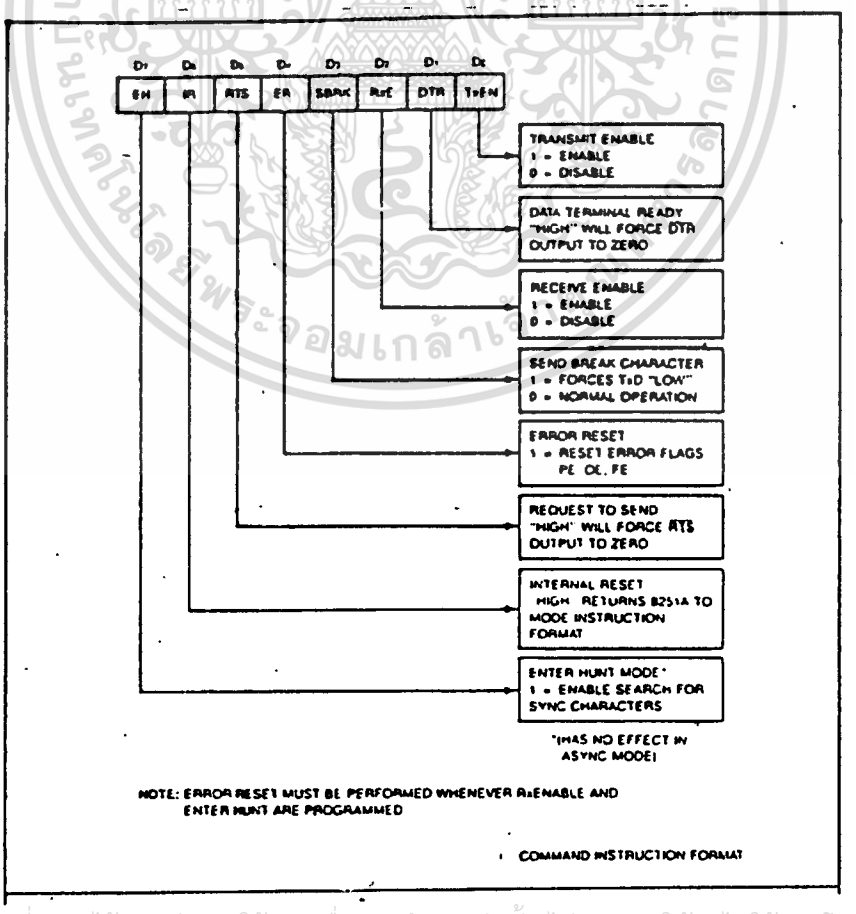
ส่วนใน SYNCHRONOUS MODE นี้ หลังจากส่ง MODE WORD ให้กับรีจีสเตอร์ควบคุมแล้ว 8251 จะถือว่าไบท์ต่อไปที่ CPU ส่งให้นั้นเป็น SYNC CHARACTER ในกรณีนี้เราจะต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการส่ง SYNC CHARACTER ให้กับ 8251 อีก 2 ไบท์ และข้อมูลไบท์ต่อไป 8251 จะถือว่าเป็น COMMAND WORD



รูปที่ 3.19 MODE WORD ใน SYNCHRONOUS

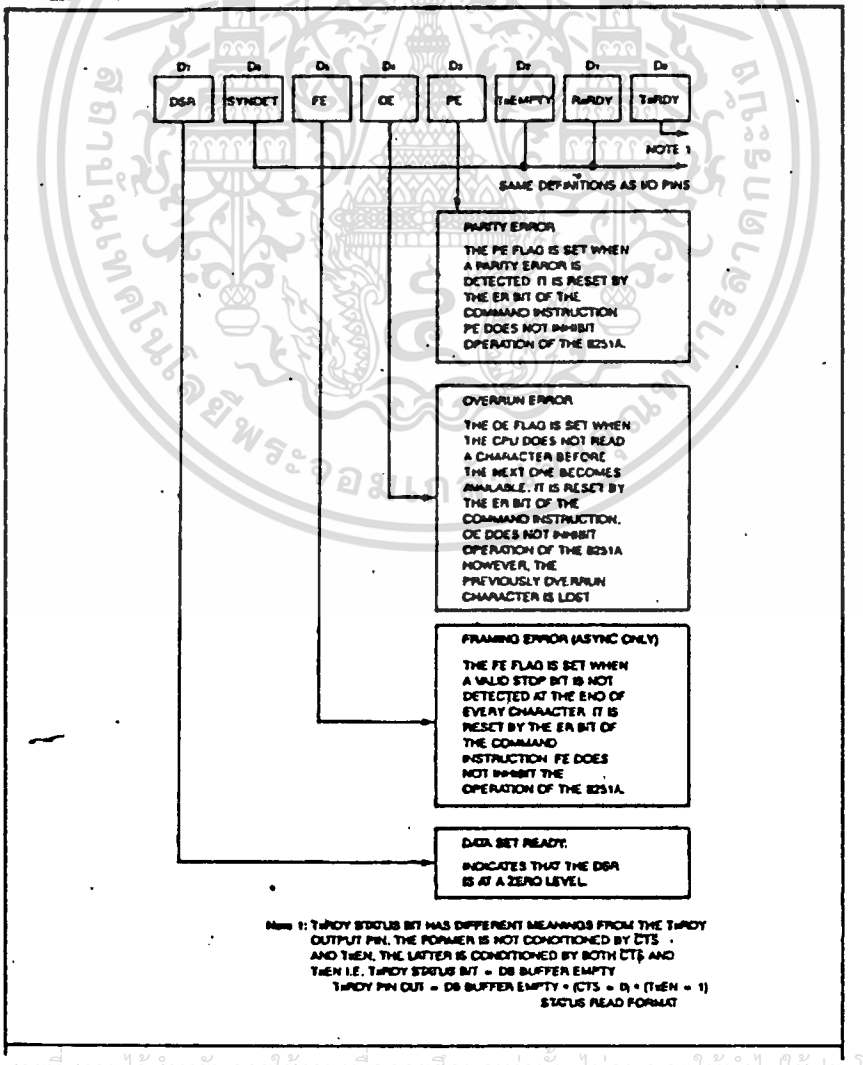


รูปที่ 3.20 แสดงรูปแบบของ COMMAND WORD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้กำหนดให้ 8251 ทำการรับและส่งข้อมูลได้ (บิท Ø และ 2 เป็น "1") ทำการรีเซ็ท ERROR FLAG ในรีจีสเตอร์สถานะคือ PE , FE และ OE สำหรับการทำ INTERNAL RESET (IR) นั้นสามารถที่จะทำได้โดยการทำให้บิท 6 (IR) ของ COMMAND WORD เป็น "1" แต่ในที่นี้จะยังไม่ทำ INTERNAL RESET ดังนั้น COMMAND WORD ที่จะส่งให้กับ 8251 ก็คือ ØØØ1Ø1Ø1B หรือ 15H สำหรับการส่ง COMMAND WORD นี้ให้กับรีจีสเตอร์ควบคุม ของ 8251 นั้นสามารถที่จะทำได้โดยใช้วิธีเดียวกับการส่ง MODE WORD คือส่งออกไปที่พอร์ทควบคุม

สำหรับรีจีสเตอร์ตัวต่อไปที่จะกล่าวถึงก็คือรีจีสเตอร์สถานะ ซึ่งจะทำหน้าที่ในการแสดงสภาวะของ 8251 ในขณะที่ทำงานอยู่ว่ามีความผิดพลาดเกิดขึ้นหรือไม่และยังมีหน้าที่ในการแสดงสถานะที่ขาต่างๆ ของ 8251 ที่เกี่ยวข้องกับการรับส่งข้อมูลด้วยว่าเป็นอย่างไร รูปที่ 3.21



รูปที่ 3.21 แสดงรีจิสเตอร์สถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรีจีสเตอร์สถานะนี้ CPU สามารถที่จะทำการตรวจสอบได้โดยการอ่านข้อมูลจากพอร์ทควบคุม ส่วนบิทต่างๆ ที่ใช้ในการแสดงสถานะของขาของ8251 คือ DSR , SYNDET , TxEMPTY , RxRDY และ TxRDY นั้นจะช่วยให้ CPU ทราบถึงความพร้อมของอุปกรณ์รับส่งข้อมูล สำหรับอีก 3 บิทที่เหลือนั้นจะแสดงความผิดพลาดที่เกิดจากการรับส่งข้อมูลของ 8251 ซึ่งเราสามาถที่จะทำการรีเซ็ทบิททั้งสามนี้ได้โดยการทำให้บิท 4 ใน COMMAND WORD เป็น "1"

สำหรับบิททั้ง 3 ที่แสดงความผิดพลาดของการรับส่งข้อมูลมีดังนี้

1. PE (RARITY ERROR) จะเป็น "1" ในกรณีที่ 8251 พบว่ามีความผิดพลาดในการตรวจสอบพาริตี้

2. OE (OVERRUN ERROR) จะเป็น "1" เมื่อข้อมูลใหม่ถูกส่งเข้ามาทับข้อมูลเก่าโดยที่ข้อมูลเก่ายังไม่ถูก CPU

3. FE (FRAMING ERROR) จะเป็น "1" ในกรณีที่ 8251 ตรวจหาสต๊อปบิทไม่พบ

3.4.5 การเชื่อมต่อระหว่าง 8251 กับ 6502

เนื่องจาก 8251 USART เป็น BUS-COMPATIBLE มันจึงไม่ยากนักที่จะเชื่อมต่อการทำงานกับเครื่องแอปเปิ้ล เราควรจะจำ DATA BUS INPUT , RD และ WR CONTROL INPUT และ CS (CHIP SELECT INPUT) เพราะว่า USART ประกอบด้วย MODE 2 MODE ซึ่งจะมีหนทางที่จะแบ่งแยกระหว่าง C/D ที่ขา 12 ซึ่งจะมีการทำงานและหน้าที่ คือ เมื่อลอจิก "1" เป็นการเลือก CONTROL MODE หรือ COMMAND MODE ขณะที่ลอจิก "Ø" เลือก DATA MODE ADDRESS BIT ถูกเชื่อมกับขานี้ 1 เส้น (AØ) เพื่อที่จะทำให้ 8251 ทำงานในแต่ละ MODE (สำหรับ COMMAND MODE และ DATA MODE)

เนื่องด้วย USART จะต้องติดต่อกับ SERIAL DEVICE ที่เป็นแบบ ASYNCHRONOUS อื่นๆ ดังนั้นจึงต้องมีมาตรฐานสำหรับอัตราที่ใช้เพื่อให้แน่ใจว่า DATA RATE ของอุปกรณ์ที่ใช้ส่งและ รับใกล้เคียงกัน เพราะว่าการติดต่อทุกชนิดจะไม่เป็นผลถ้าปราศจาก SOFTWARE ที่จะ DRIVE ตัว 8251 ให้ทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และ ผลการทดลอง

4.1 การทดลอง

ในบทนี้ได้ทำการทดลองสร้างวงจรแต่ละส่วนตามบล็อกไดอะแกรมที่กล่าวในบทที่ 3 เพื่อประกอบเป็นระบบที่สามารถทำงานได้ โดยจะแบ่งการทดลองเป็น 3 ส่วนใหญ่ ๆ คือ

- A/D
- D/A
- การรับส่งข้อมูลแบบอนุกรม

จากนั้นจึงออกแบบเขียนซอฟแวร์เพื่อควบคุมการทำงานของฮาร์ดแวร์

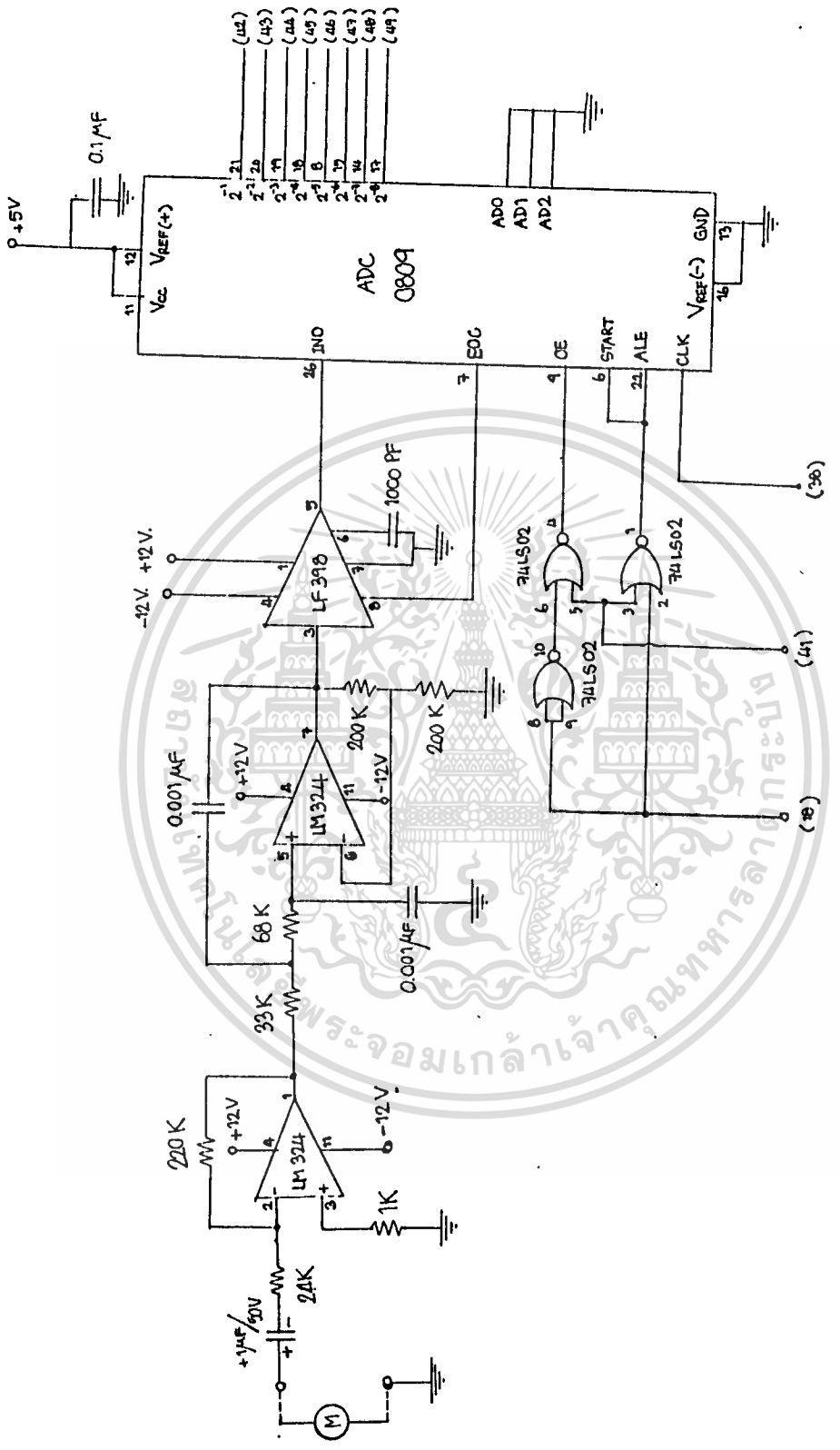
รูปที่ 4.1 วงจร A/D ที่ใช้ทดลอง

รูปที่ 4.2 วงจร D/A ที่ใช้ทดลอง

รูปที่ 4.3 วงจรที่ใช้ทดลองการรับส่งข้อมูลแบบอนุกรม

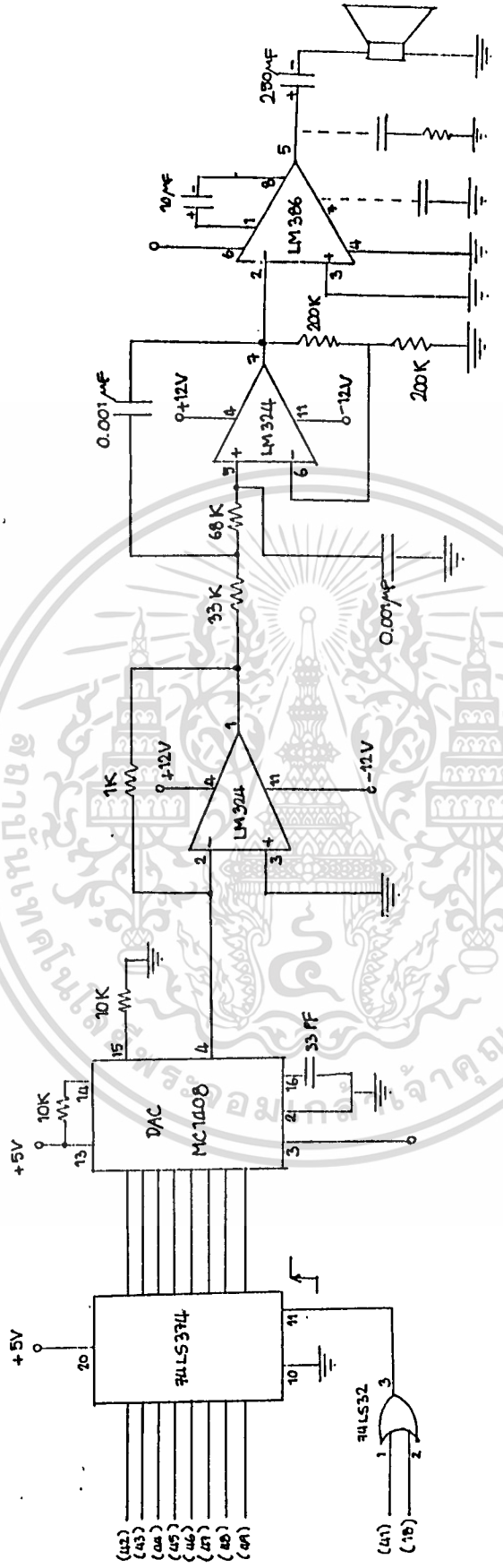
อนึ่งโฟล์วชาร์ตและโปรแกรมที่ใช้ทั้งหมดได้แทรกไว้ในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



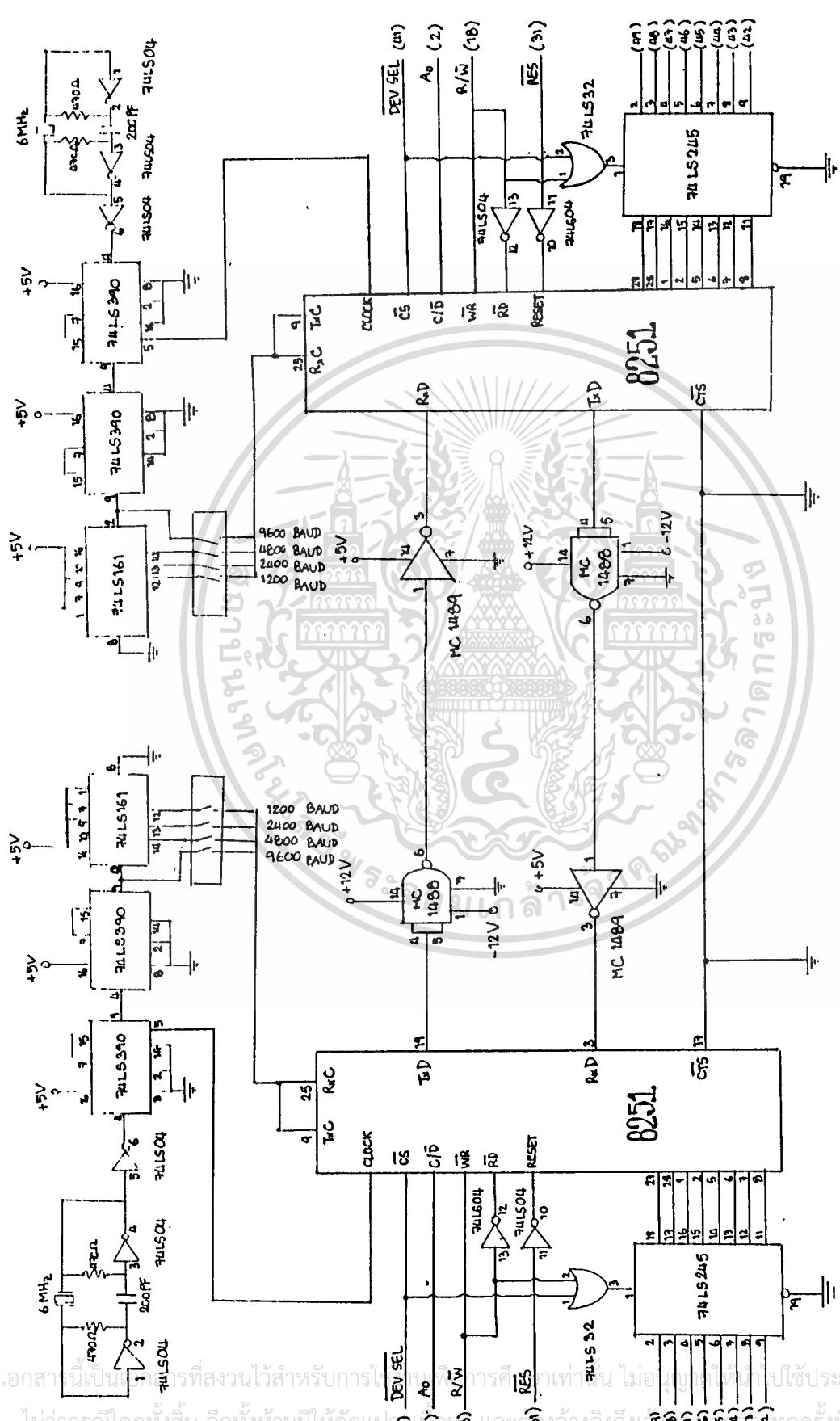
รูปที่ 4.1 วงจร A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 วงจร D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ TEST A/D

```
10  POKE 49280,0
20 Y =  PEEK (49280)
30  PRINT Y
40  PRINT  SPC( Y);"*"
50  GOTO 10
```

โปรแกรม TEST การรับส่งข้อมูลแบบอนุกรม

ด้านส่ง

```
0300-   A9 DD      LDA   #$DD
0302-   8D B1 C0   STA   $C0B1
0305-   A9 15      LDA   #$15
0307-   8D B1 C0   STA   $C0B1
030A-   20 50 03   JSR   $0350
030D-   AD B1 C0   LDA   $C0B1
0310-   29 01      AND   #$01
0312-   C9 01      CMP   #$01
0314-   D0 F7      BNE   $030D
0316-   B1 FC      LDA   ($FC),Y
0318-   8D B0 C0   STA   $C0B0
031B-   C8         INY
031C-   D0 EF      BNE   $030D
031E-   00         BRK
```

```
0350-   A0 00      LDY   #$00
0352-   A9 FF      LDA   #$FF
0354-   85 FD      STA   $FD
0356-   A2 3A      LDX   #$3A
0358-   86 FC      STX   $FC
035A-   60         RTS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านรับ

```
0300-   A9 DD      LDA   #$DD
0302-   8D B1 C0   STA   $C0B1
0305-   A9 15      LDA   #$15
0307-   8D B1 C0   STA   $C0B1
030A-   20 50 03   JSR   $0350
030D-   AD B1 C0   LDA   $C0B1
0310-   29 02      AND   #$02
0312-   C9 02      CMP   #$02
0314-   D0 F7      BNE   $030D
0316-   AD B0 C0   LDA   $C0B0
0319-   91 FC      STA   ($FC),Y
031B-   C8         INY
031C-   D0 EF      BNE   $030D
031E-   00         BRK

0350-   A0 00      LDY   #$00
0352-   A9 10      LDA   #$10
0354-   85 FD      STA   $FD
0356-   A2 00      LDX   #$00
0358-   86 FC      STX   $FC
035A-   60         RTS
```

MAIN PROGRAM

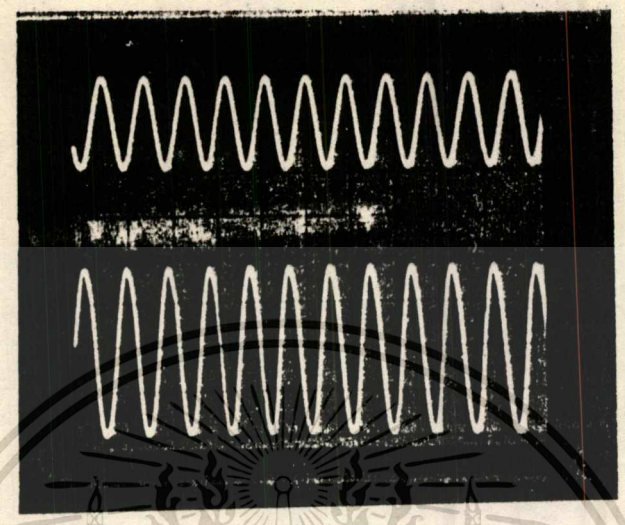
```
1000-   8D 80 C0   STA   $C080
1003-   20 00 03   JSR   $0300
1006-   AD 80 C0   LDA   $C080
1009-   8D B0 C0   STA   $C0B0
100C-   4C 00 10   JMP   $1000
```

DELAY PROGRAM

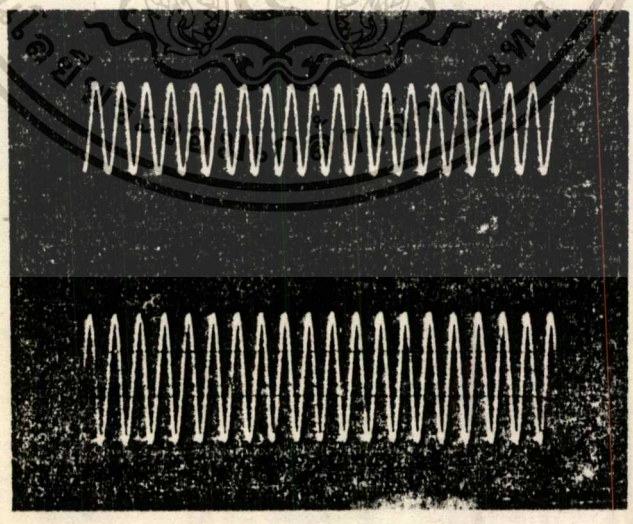
```
0300-   A9 05      LDA   #$05
0302-   A2 00      LDX   #$00
0304-   85 00      STA   $00
0306-   C6 00      DEC   $00
0308-   D0 FC      BNE   $0306
030A-   60         RTS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

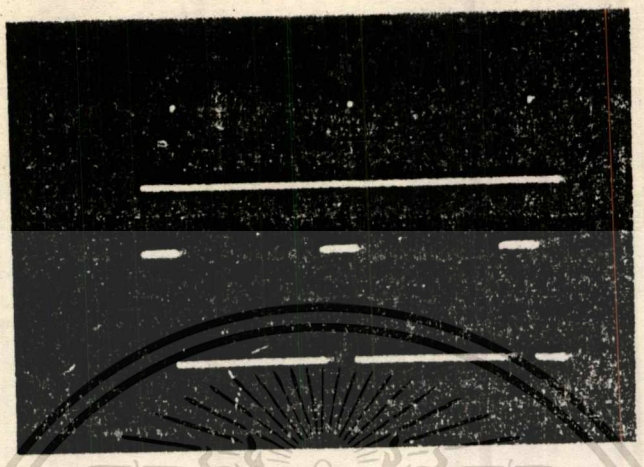


รูปที่ 4.4 แสดงสัญญาณ o/p ของฟิลเตอร์เทียบกับ i/p
ที่ความถี่ 1 KHz ของ A/D



รูปที่ 4.5 แสดงสัญญาณ o/p ของฟิลเตอร์เทียบกับ i/p
ที่ความถี่ 3.4 KHz ของ A/D

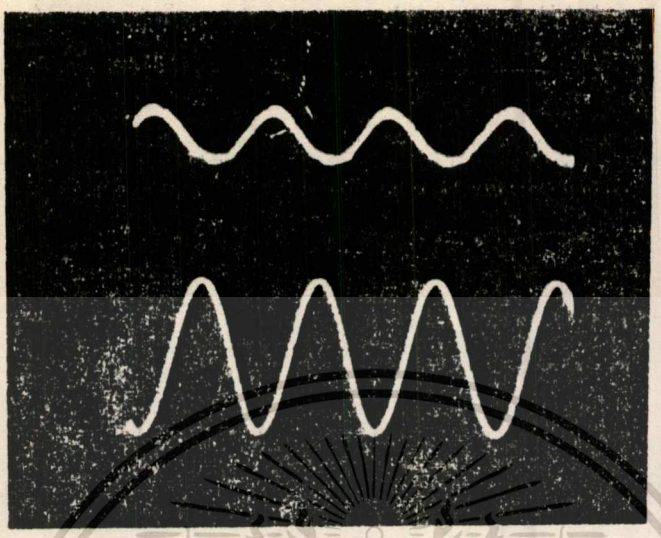
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



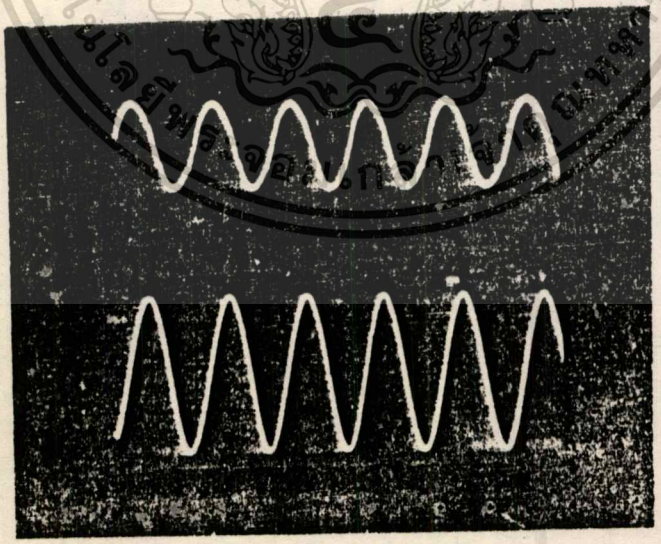
รูปที่ 4.6 แสดงสัญญาณขา EOC และ A/D CONVERSION DELAY

รูปที่ 4.7 แสดงสัญญาณเสียงที่ขา i/p และ o/p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงสัญญาณ o/p ของฟิลเตอร์เทียบกับ i/p
ที่ความถี่ 1 KHz ของ D/A



รูปที่ 4.9 แสดงสัญญาณ o/p ของฟิลเตอร์เทียบกับ i/p
ที่ความถี่ 3.4 KHz ของ D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผล และ วิจารณ์

จากการทดลองปัญหาที่พบมากในส่วนของ A/D คือเรื่องของ CONVERSION TIME กล่าวคือ A/D กับ คอมพิวเตอร์จะทำงานไม่ทันกัน ดังนั้นจึงต้องแก้ปัญหาโดยเขียนโปรแกรมหน่วงเวลา (DELAY) เพื่อให้คอมพิวเตอร์ทำงานได้พร้อม ๆ กับ A/D นอกจากนี้ยังมีการรบกวนที่เกิดกับเอาท์พุทของโลว์พาสฟิลเตอร์ซึ่งอาจเป็นสาเหตุมาจากตัวลำโพงเอง

อย่างไรก็ตาม ระบบที่ทดลองนี้ยังใช้งานได้ดีพอสมควร และอาจจะพัฒนาให้มีประสิทธิภาพที่ดีขึ้นกว่านี้

สำหรับการรับส่งข้อมูลแบบอนุกรมนั้น คล็อกที่ใช้กำหนดบอดเรท นับว่ามีความสำคัญมาก เพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพ คล็อกทางด้านส่งและรับควรจะมีค่าเท่ากันหรือใกล้เคียงกันมาก

การทำงานของส่วนต่าง ๆ ของระบบ จะไม่มีความหมายถ้าขาดซอฟแวร์คอยควบคุมให้เป็นตามที่ต้องการอย่างถูกต้อง จะเห็นว่าเมื่อต้องการควบคุมการทำงานทางฮาร์ดแวร์นั้น ภาษาที่สะดวกต่อการใช้งานก็คือ ภาษาเครื่อง ซึ่งนับว่ามีความสำคัญมากทีเดียว

สำหรับ RS-232 ที่เราสร้างขึ้น นับว่าเป็นมาตรฐานสากลที่ใช้กันมาก ดังจะเห็นว่าในปัจจุบันเครื่องไมโครคอมพิวเตอร์ต่างมี RS-232 CONNECTOR อยู่ในตัวเครื่อง

การที่จะสร้างระบบการติดต่อสื่อสารขึ้นมาจึงเป็นสิ่งจำเป็นเพราะว่า อนาคตข้างหน้าความสะดวกและรวดเร็วจะกลายเป็นความจำเป็นในชีวิตประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

A to D, D to A

ADC0808, ADC0809 8-Bit μP Compatible A/D Converters With 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

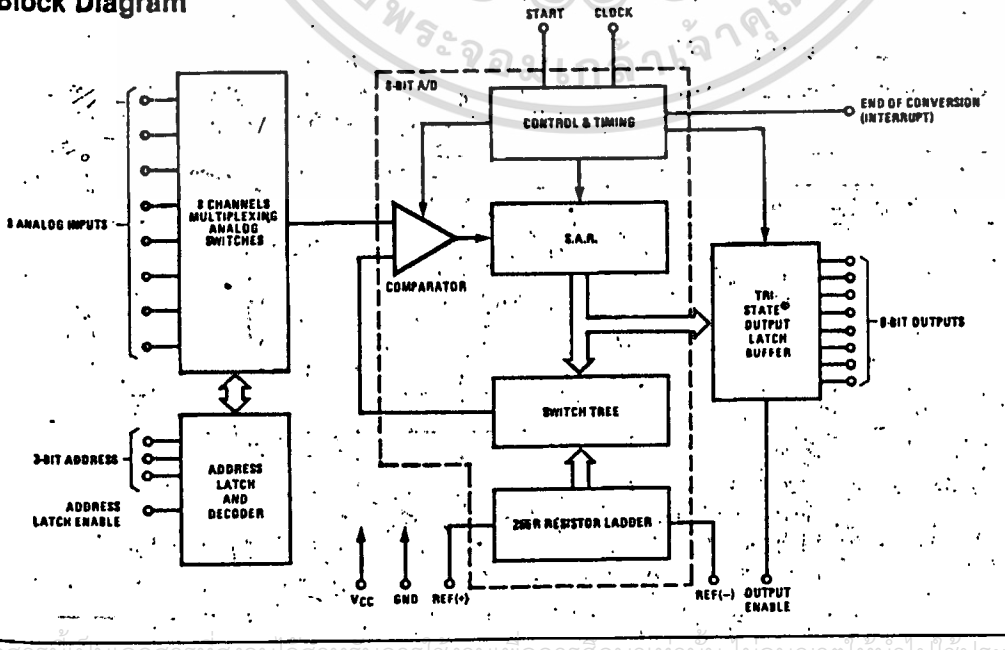
The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet.

Features

- Resolution — 8-bits
- Total unadjusted error — ±1/2 LSB and ±1 LSB
- No missing codes
- Conversion time — 100 μs
- Single supply — 5 V_{DC}
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T^2L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range −40°C to +85°C or −55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE® output

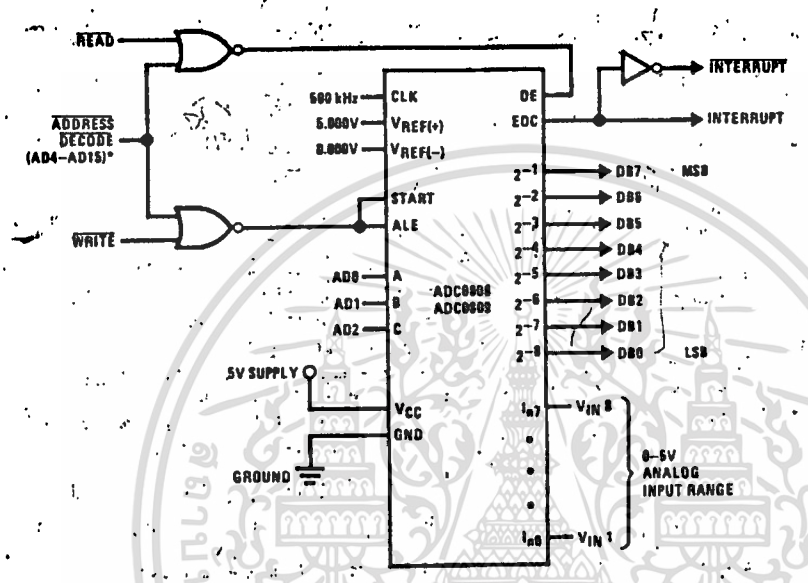
Block Diagram



8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Application



* Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

MICROPROCESSOR INTERFACE TABLE

PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	MEMR	MEMW	INTR (Thru RST Circuit)
8085	RD	WR	INTR (Thru RST Circuit)
Z-80	RD	WR	INT (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA·Φ2·R/W	VMA·Φ2·R/W	IRQA or IRQB (Thru PIA)

Ordering Information

TEMPERATURE RANGE		−40°C to +85°C		−55°C to +125°C
Error	± 1/2 Bit Unadjusted	ADC0808CCN	ADC0808CCJ	ADC0808CJ
	± 1 Bit Unadjusted	ADC0809CCN		
Package Outline		N28A Molded DIP	J28A Hermetic DIP	J28A Hermetic DIP

8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

A to D, D to A

LM1508/LM1408 8-Bit D/A Converter

General Description

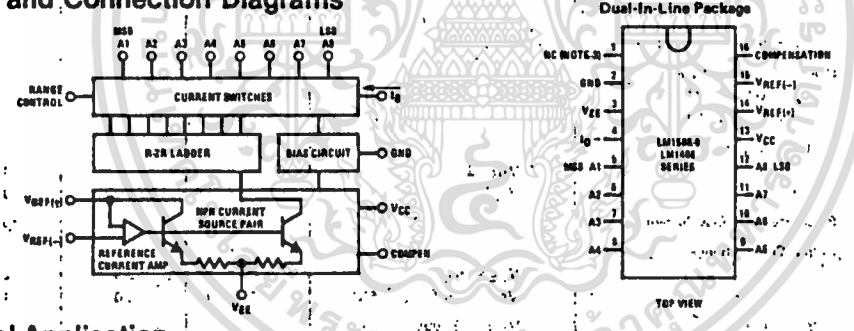
The LM1508/LM1408 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 I_{REF}/256. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 μA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the LM1508/LM1408 are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The LM1508/LM1408 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet. For more information, see DAC0808 data sheet.

Features

- Relative accuracy: ±0.19% error maximum LM1508-8 and LM1408-8
- Full scale current match: ±1 LSB typ
- 7 and 6-bit accuracy available
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/μs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

Block and Connection Diagrams



Typical Application

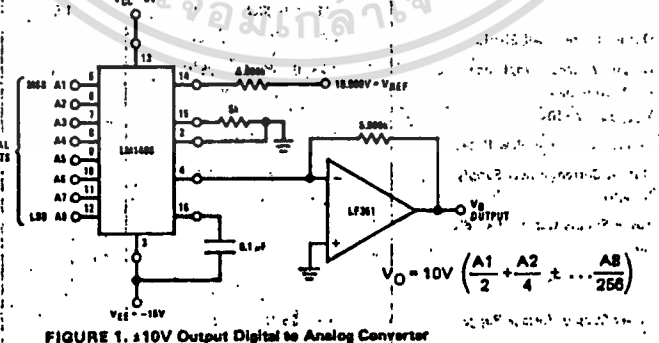


FIGURE 1. ±10V Output Digital to Analog Converter

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS*		
		HERMETIC PACKAGE (D16C)	HERMETIC PACKAGE (J16A)	PLASTIC PACKAGE (N16A)
8-Bit	−55°C ≤ T_A ≤ +125°C	LM1508D-8	LM1508J-8	
8-Bit	0°C ≤ T_A ≤ +75°C		LM1408J-8	LM1408N-8
7-Bit	0°C ≤ T_A ≤ +75°C		LM1408J-7	LM1408N-7
6-Bit	0°C ≤ T_A ≤ +75°C		LM1408J-6	LM1408N-6

*Note. Devices may be ordered by using either order number.

8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

LF198/LF298/LF398, LF198A/LF398A Monolithic Sample and Hold Circuits

General Description

The LF198/LF298/LF398 are monolithic sample and hold circuits which utilize BI-FET technology to obtain ultra-high dc accuracy with fast acquisition of signal and low droop rate. Operating as a unity gain follower, dc gain accuracy is 0.002% typical and acquisition time is as low as 6μs to 0.01%. A bipolar input stage is used to achieve low offset voltage and wide bandwidth. Input offset adjust is accomplished with a single pin and does not degrade input offset drift. The wide bandwidth allows the LF198 to be included inside the feedback loop of 1 MHz op amps without having stability problems. Input impedance of $10^{10}\Omega$ allows high source impedances to be used without degrading accuracy.

P-channel junction FET's are combined with bipolar devices in the output amplifier to give droop rates as low as 5 mV/min with a 1μF hold capacitor. The JFET's have much lower noise than MOS devices used in previous designs and do not exhibit high temperature instabilities. The overall design guarantees no feedthrough from input to output in the hold mode even for input signals equal to the supply voltages.

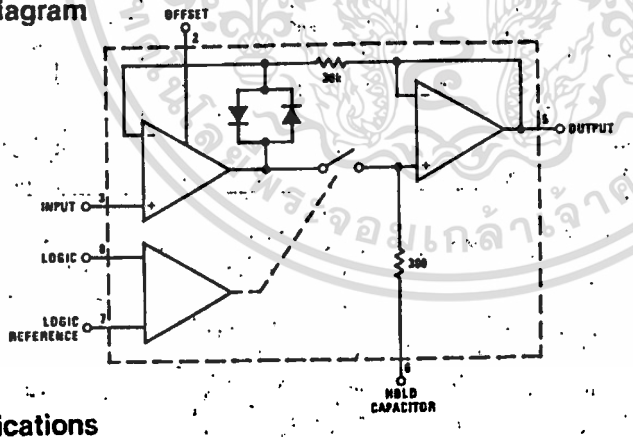
Features

- Operates from ±5V to ±18V supplies
- Less than 10μs acquisition time
- TTL, PMOS, CMOS compatible logic input
- 0.5 mV typical hold step at $C_h = 0.01\mu F$
- Low input offset
- 0.002% gain accuracy
- Low output noise in hold mode
- Input characteristics do not change during hold mode
- High supply rejection ratio in sample or hold
- Wide bandwidth

Logic inputs on the LF198 are fully differential with low input current, allowing direct connection to TTL, PMOS, and CMOS. Differential threshold is 1.4V. The LF198 will operate from ±5V to ±18V supplies. It is available in an 8-lead TO-5 package.

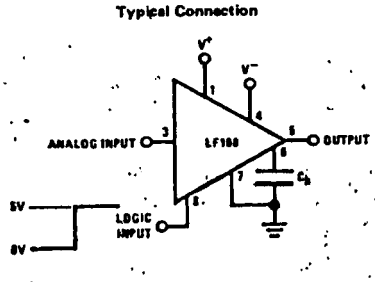
An "A" version is available with tightened electrical specifications.

Functional Diagram

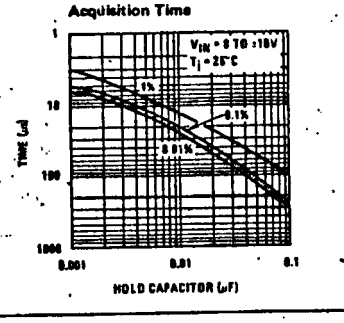


Typical Applications

Typical Connection



Acquisition Time



7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

Audio, Radio and TV Circuits

LM386 Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value up to 200.

The inputs are ground referenced while the output is automatically biased to one half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

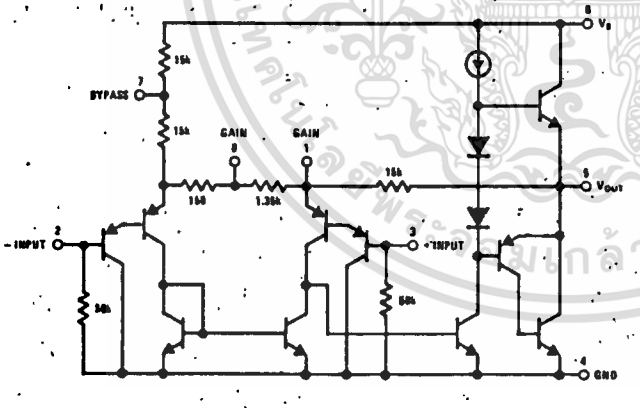
Features

- Battery operation
- Minimum external parts
- Wide supply voltage range 4V–12V or 5V–18V
- Low quiescent current drain 4 mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion
- Eight pin dual-in-line package

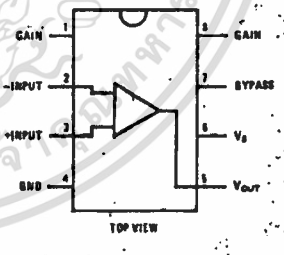
Applications

- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams



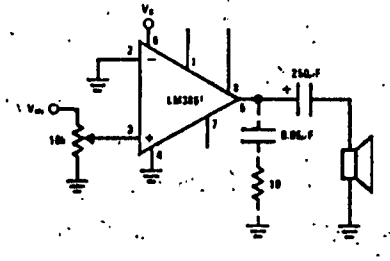
Dual-In-Line Package



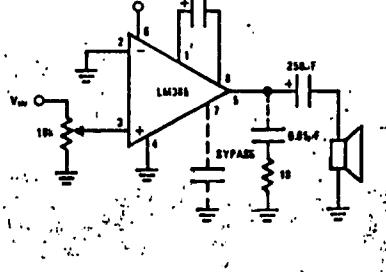
Order Number LM386N-1, LM386N-3 or LM386N-4
See NS Package N08B

Typical Applications

Amplifier with Gain = 20
Minimum Parts



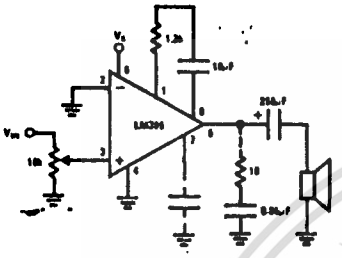
Amplifier with Gain = 200



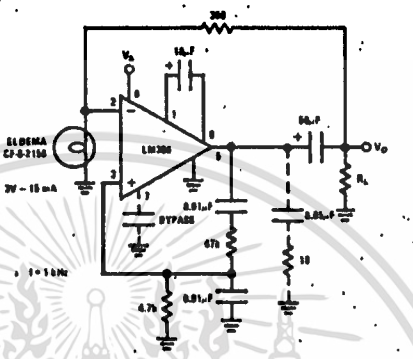
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

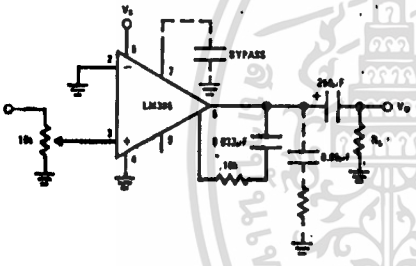
Amplifier with Gain = 50



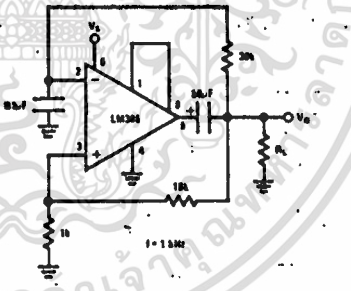
Low Distortion Power Wienbridge Oscillator



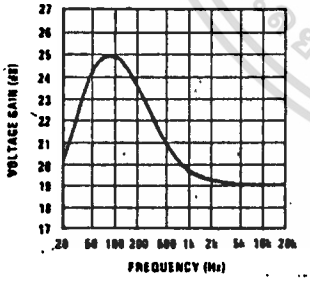
Amplifier with Bass Boost



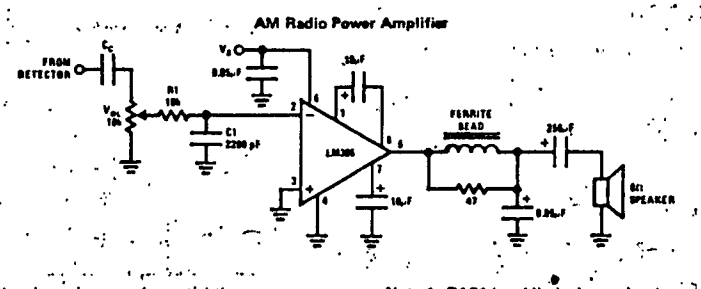
Square Wave Oscillator



Frequency Response with Bass Boost



AM Radio Power Amplifier



Note 1: Twist supply lead and supply ground very tightly.
Note 2: Twist speaker lead and ground very tightly.
Note 3: Ferrite bead is Ferroxcube K5-001-001/3B with 3 turns of wire.
Note 4: R1C1 band limits input signals.
Note 5: All components must be spaced very close to IC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

Operational Amplifiers/Buffers

LM124/LM224/LM324, LM124A/LM224A/LM324A, LM2902 Low Power Quad Operational Amplifiers

General Description

The LM124 series consists of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124 series can be directly operated off of the standard +5 V_{DC} power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional ±15 V_{DC} power supplies.

Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.
- The unity gain cross frequency is temperature compensated.
- The input bias current is also temperature compensated.

Advantages

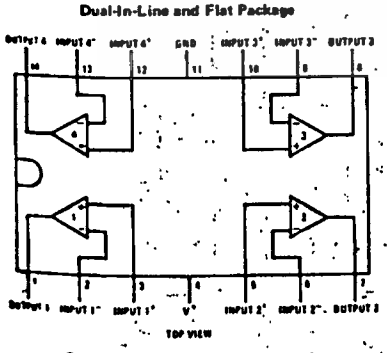
- Eliminates need for dual supplies
- Four internally compensated op amps in a single package
- Allows directly sensing near GND and V_{OUT} also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

- Internally frequency compensated for unity gain
- Large dc voltage gain 100 dB
- Wide bandwidth (unity gain) 1 MHz (temperature compensated)
- Wide power supply range:
  - Single supply 3 V_{DC} to 30 V_{DC}
  - or dual supplies ±1.5 V_{DC} to ±15 V_{DC}
- Very low supply current drain (800μA) — essentially independent of supply voltage (1 mW/op amp at +5 V_{DC})
- Low input biasing current 45 nA_{DC} (temperature compensated)
- Low input offset voltage 2 mV_{DC} and offset current 5 nA_{DC}
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing 0 V_{DC} to V^+ − 1.5 V_{DC}

Connection Diagram

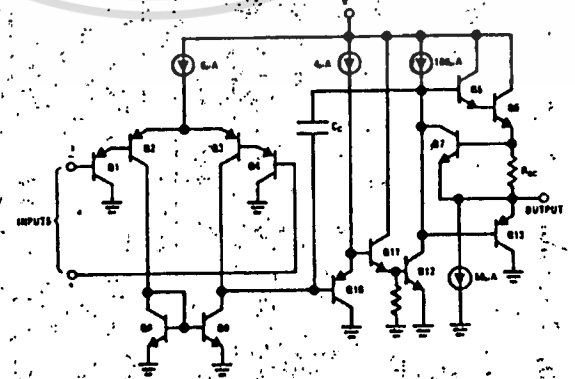
Dual-In-Line and Flat Package



Order Number LM124J, LM124AJ, LM224J, LM224AJ, LM324J, LM324AJ or LM2902J
See NS Package J14A

Order Number LM324N, LM324AN or LM2902N
See NS Package N14A

Schematic Diagram (Each Amplifier)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications (Continued) (V^+ = 5.0 V_{DC})

Voltage Controlled Oscillator Circuit

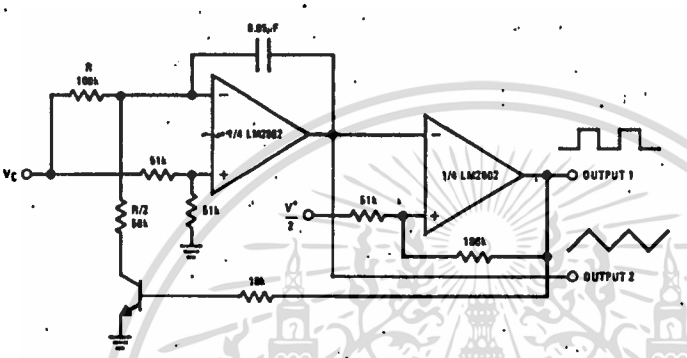
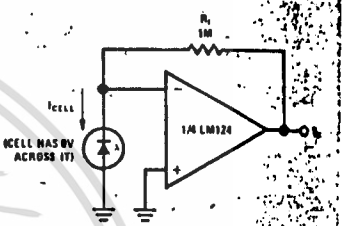
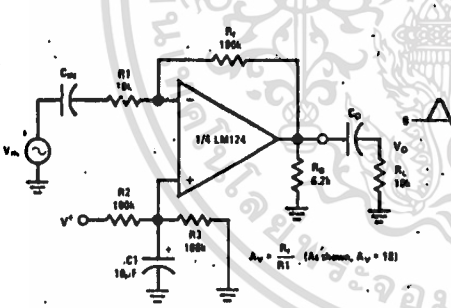


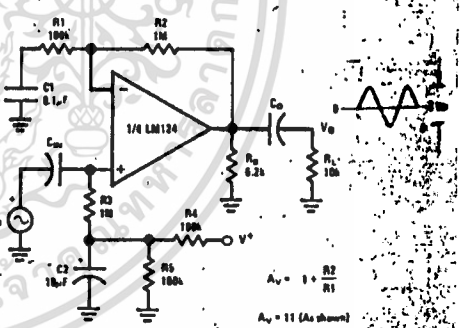
Photo Voltaic-Cell Amplifier



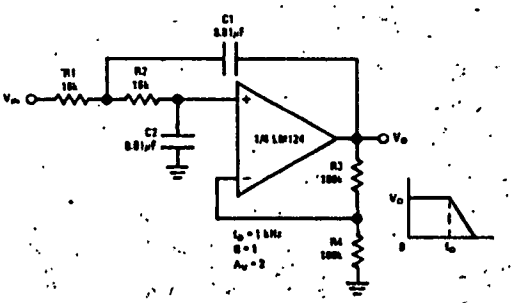
AC Coupled Inverting Amplifier



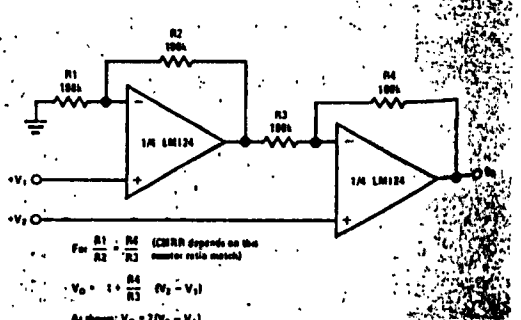
AC Coupled Non-Inverting Amplifier



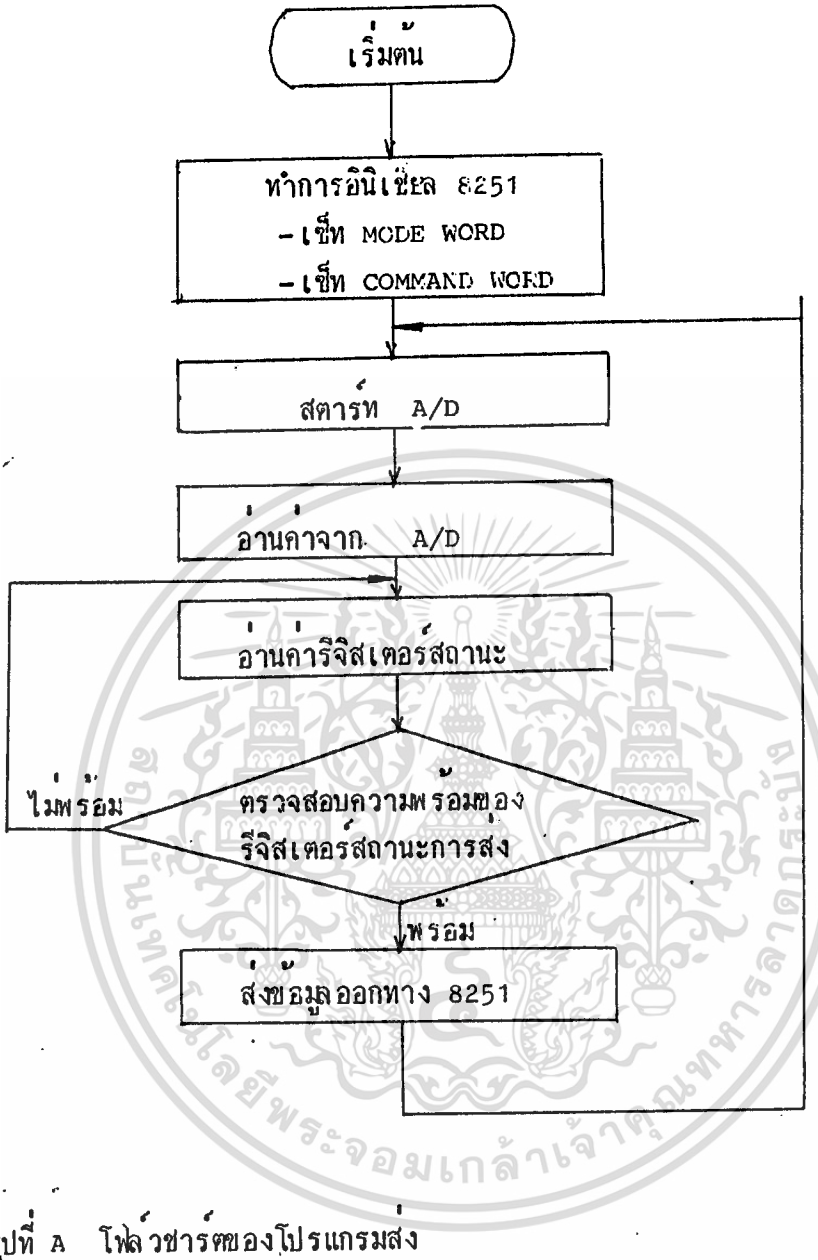
DC Coupled Low-Pass RC Active Filter



High Input Z, DC Differential Amplifier

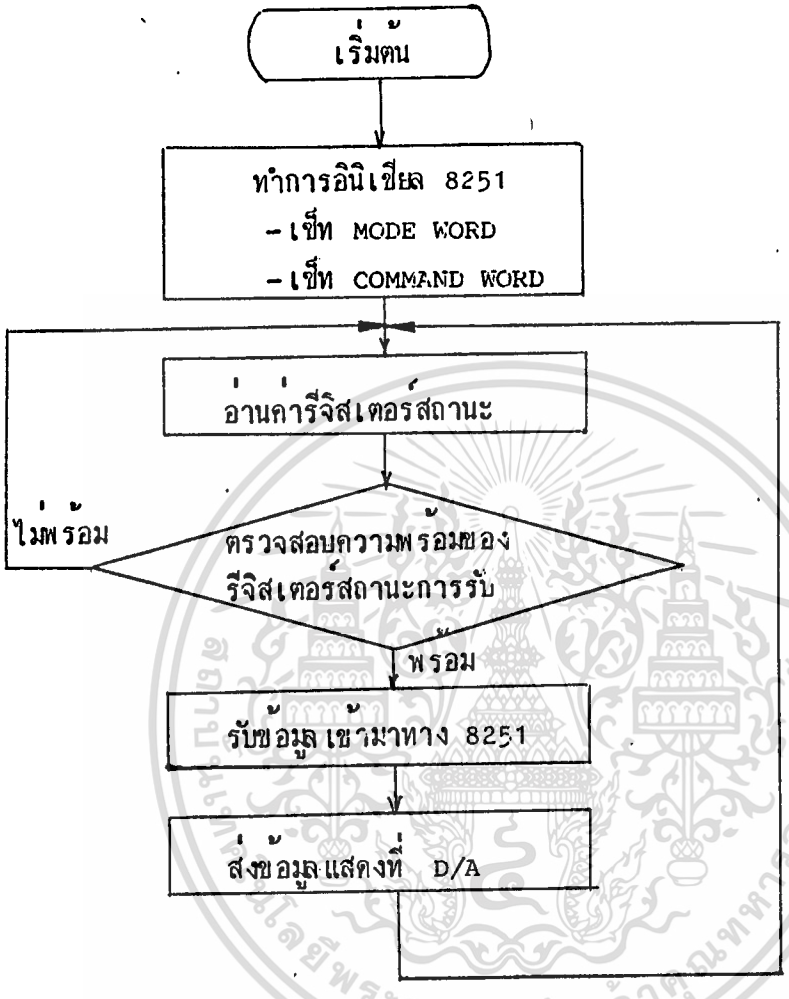


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A โฟล์วชาร์ตของโปรแกรมส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ B โฟล์วชาร์ตของโปรแกรมรับ

ตัวอย่างโปรแกรมส่ง

```
0300-   A9 DD      LDA   #$DD
0302-   8D B1 C0   STA   $C0B1
0305-   A9 15      LDA   #$15
0307-   8D B1 C0   STA   $C0B1
030A-   8D 90 C0   STA   $C090
030D-   AD 90 C0   LDA   $C090
0310-   48         PHA
0311-   AD B1 C0   LDA   $C0B1
0314-   29 01      AND   #$01
0316-   C9 01      CMP   #$01
0318-   D0 F7      BNE   $0311
031A-   68         PLA
031B-   8D B0 C0   STA   $C0B0
031E-   4C 0A 03   JMP   $030A
```

ตัวอย่างโปรแกรมรับ

```
0300-   A9 DD      LDA   #$DD
0302-   8D B1 C0   STA   $C0B1
0305-   A9 15      LDA   #$15
0307-   8D B1 C0   STA   $C0B1
030A-   AD B1 C0   LDA   $C0B1
030D-   29 02      AND   #$02
030F-   C9 02      CMP   #$02
0311-   D0 F7      BNE   $030A
0313-   AD B0 C0   LDA   $C0B0
0316-   8D 90 C0   STA   $C090
0319-   4C 0A 03   JMP   $030A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณ อาจารย์สมยศ จุณณะปิยะ อาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง ที่ให้คำปรึกษา อำนวยความสะดวกต่างๆ ตลอดจนอุปกรณ์และสถานที่ในการทดลองจนเป็นผลสำเร็จด้วยดี

ผู้จัดทำ

นายวิชัย นทีนันทสวัสดิ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หนังสืออ้างอิง

1. นคร อนันตชัย,"ความรู้เบื้องต้นของการส่งข้อมูล",วารสารไมโครคอมพิวเตอร์,ฉบับที่ 17, หน้า 90-97,2528

2. Jame W Coffron," Z80 APPLICATION ", Sybex Inc, pp209-237, 1983

3. J.A.Titus, D.G.Larsen and C.A.Titus," APPLE INTERFACEING ",Howard W.Sams & Co.,Inc.,Indianapolis,pp164-179, 1981

4. J.N.Fernandez,D.N.Tabler and R.Ashley," 6502 ASSEMBLY LANGUAGE PROGRAMMING ",John Wiley & Sons,Inc., New York, pp272-273

5. " APPLE REFERENCE MANUAL "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้