พีซี ลอจิกอนาไลซ์เซอร์ 8 แชนแนล

PC LOGIC ANALYZER 8 CHANNEL

โดย

นาย ดำรงค์ อัคนิจ

นาย โสภณ แซ่เตีย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034748

หัวข้อปริญญานิพนธ์ พีซี ลอจิกอนาไลซ์เซอร์ 8 แชนแนล

PC LOGIC ANALYZER 8 CHANNEL

ชื่อนักศึกษา นาย ดำรงค์ อัคนิจ

นาย โสภณ แซ่เตีย

อาจารย์ที่ปรึกษา ดร. ไพศาล นาคพิพัฒน์

อ. ชวลิต เบญจางคประเสริฐ

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2537

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พีซี ลอจิกอนาไลซ์เซอร์ 8 แชนแนล

โดย	นาย ดำรงค์	อัคนิจ	รหัส 36012008
	นาย โสภณ	แซ่เตีย	รหัส 36012041

อาจารย์ที่ปรึกษา	ดร. ไพศาล	นาคพิพัฒน์
	อ. ชวลิต	เบญจางคประเสริฐ

ปีการศึกษา 2537

บทคัดย่อ

ในการทำงานทางด้านอิเล็กทรอนิคส์ มีความจำเป็นต้องมีความเกี่ยวข้องกับสัญญาณทั้งสัญญาณอนาล็อกและสัญญาณดิจิตอล ซึ่งสัญญาณทั้งสองนี้สามารถใช้เครื่องมือวัดทำการตรวจเช็คได้ และในกรณีสัญญาณดิจิตอลนั้นเครื่องมือที่ใช้วัดมีหลายชนิดและหนึ่งในนั้นก็มีลอจิกอนาไลซ์เซอร์ ซึ่งมีความสามารถพิเศษในการวัดสัญญาณดิจิตอลได้หลายช่องสัญญาณในเวลาเดียวกัน

ลอจิกอนาไลซ์เซอร์ยังสามารถแสดงรูปแบบของสัญญาณได้หลายแบบเช่น Timing Diagram , Binary and Hexadecimal เป็นต้น โดยที่สัญญาณที่วัดได้นั้นมีค่าความเที่ยงตรงสูงเนื่องจากมีสัญญาณ Time Base ที่สูงกว่าสัญญาณภายนอก ไม่ต่ำกว่า 2 เท่า

สำหรับเครื่องลอจิกอนาไลซ์เซอร์ที่ได้ทำการออกแบบไว้ สามารถทำงานได้ในความถี่ Time Base สูงสุด 20 MHz และยังสามารถทำการเก็บผลจากการวัดไว้ได้ เพื่อความสะดวกในการตรวจสอบย้อนหลัง อีกทั้งยังสามารถค้นหาข้อมูลที่ต้องการได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC LOGIC ANALYZER 8 CHANNEL

BY	MR. DAMRONG	AUKKANU	NO.36.12008
	MR. SOPHON	SAE-TIA	NO.36012041

ADVISER	DR. PAISAL	NAKPIPAT
	MR. CHAWALIT	BENJANGKAPRASERT

YEAR 1994

ABSTRACT

In the electronics working, both analog signal and digital signal are necessary. The signals can be measured by various instuments. One of them is Logic Analyzer which it can measure and display on multi channel.

Logic Analyzer can also analyse of signal or data, i.e. Timing diagram,Binary and Hexadecimal. The accuracy is so high due to the time base is twice of the analyse signal.

For this Logic Analyzer have been designed. The time base of as high 20 Mhz can be used. The result can be stroed in order to make the next calling and it can be used in searching mode too.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงได้ด้วยดี ก็เนื่องจากได้รับความช่วยเหลืออย่างดีของอาจารย์ที่ปรึกษา ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ ที่เป็นประโยชน์ในการทำวิทยานิพนธ์ ตลอดจนเพื่อน ๆที่ช่วยให้คำแนะนำ จนทำให้การทำวิทยานิพนธ์นี้เสร็จได้ด้วยดี

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา - มารดา ที่สนับสนุนทั้งด้านการเงินและให้กำลังใจแก่ผู้วิจัยมาจนสำเร็จการศึกษา

คณะผู้จัดทำวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย		ก
บทคัดย่อภาษาอังกฤษ		ช
กิตติกรรมประกาศ		ค
สารบัญภาพ		ง
บทที่	1.ปัญหาและเครื่องมือที่ใช้ในการแก้ไขปัญหาทางด้านดิจิตอล	1
	1.1 ซอร์ฟแวร์ที่ใช้ในการแก้ปัญหาทางดิจิตอล	1
	1.2 ฮาร์ดแวร์ที่ใช้ในการแก้ไขปัญหาทางดิจิตอล	1
บทที่	2.ทฤษฎีและการทำงานของลอจิกอนาไลซ์เซอร์	3
	2.1 ลักษณะการทำงานทั่วไปของลอจิกอนาไลซ์เซอร์	3
	2.2 หลักการแซมปลิ้งสัญญาณ	4
	2.3 การสุ่มข้อมูลเข้าไปเก็บในหน่วยความจำ	6
	2.5 บล็อกไดอะแกรมทั่วไปของลอจิกอนาไลซ์เซอร์	7
	2.6 บล็อกไดอะแกรมในการออกแบบลอจิกอนาไลซ์เซอร์	9
บทที่	3.วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่าง	12
	3.1 ระบบเวลา	13
	3.2 วงจรสร้างสัญญาณนาฬิกาจากคริสตอล	15
	3.3 วงจรหารความถี่	15
	3.4 วงจรนับ หาร	16
	3.5 วงจรเลือกสัญญาณการสุ่ม	19
	3.6 วงจรนับชนิดวงแหวน	21
	3.7 วงจรนับวงแหวนโหมด 4	22
	3.8 วงจรทริกเกอร์	24
	3.9 ระบบรีเซ็ทวงจร	25
	3.10 วงจรโมโนสเตเบิ้ล	26
	3.11 Fan out	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		หน้า
บทที่	4.หน่วยความจำที่ใช้ในการเก็บข้อมูลจากการแซมปลิ้ง	31
	4.1 ชนิดของแรม	31
	4.2 วงจรแรมเบื้องต้น	31
	4.3 โครงสร้างของสแตติกแรม	32
	4.4 ลำดับการเขียนข้อมูลในแรม	32
	4.5 ลำดับการอ่านข้อมูลจากแรม	33
	4.6 แรมเบอร์ 6116-3	34
	4.7 การทดสอบสแตติกแรมเบอร์ 6116-3 เบื้องต้น	36
	4.8 การออกแบบวงจรที่ใช้ในการเก็บข้อมูลที่ได้จากการแซมปลิ้ง	38
	4.9 อินพุทของวงจร	39
	4.10 แรงดันอินพุทและเอาท์พุทของเกท	40
	4.11 วงจรแลทช์	41
	4.12 เอาท์พุทของวงจร	42
	4.13 วงจรที่ใช้ในการเก็บข้อมูลที่ออกแบบ	43
	4.14 การอ่านข้อมูลจากแรม	43
บทที่	5.ส่วนของการติดต่อกับคอมพิวเตอร์	47
	-คุณสมบัติ 8255	53
บทที่	6.FLOW CHART และ ผลการทดลอง	56
	- Flow chart	56
	- การทดลองและผลการทดลอง	61
	7.บทสรุปและวิจารณ์	64
เอกสารอ้างอิง		65
ภาคผนวก		66
	- ก) วิธีการใช้งานลอจิกอนาไลซ์เซอร์	
	- ข) PROGRAM	
	- ค) DATA CHEET	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

บทที่	หน้าที่
บทที่ 1 ปัญหาและเครื่องมือที่ใช้ในการแก้ไขปัญหาทางด้านดิจิตอล	
- รูปที่ 1.1 ระดับการใช้งานของเครื่องมือที่ใช้ตรวจสอบสัญญาณดิจิตอล	2
บทที่ 2 ทฤษฎีและการทำงานของลอจิกอนาไลซ์เซอร์	
- รูปที่ 2.1 หลักการทำงานของลอจิกอนาไลซ์เซอร์	4
- รูปที่ 2.2 การสุ่มสัญญาณ	5
- รูปที่ 2.3 การสุ่มสัญญาณที่มีขนาดสัญญาณสุ่มน้อยกว่าสัญญาณอินพุท 2 เท่า	6
- รูปที่ 2.4 การสุ่มข้อมูลเข้าไปเก็บในหน่วยความจำ	6
- รูปที่ 2.5 บล็อกไดอะแกรมทั่วไปของลอจิกอนาไลซ์เซอร์	7
- รูปที่ 2.6 บล็อกไดอะแกรม ลอจิกอนาไลซ์เซอร์	9
- รูปที่ 2.7 สัญญาณการทำงานต่าง ๆ ของลอจิกอนาไลซ์เซอร์	10
บทที่ 3 วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มข้อมูล	
- รูปที่ 3.1 วงจรสร้างฐานเวลาและระบบควบคุม	12
- รูปที่ 3.2 วงจรสร้างฐานเวลามาตรฐานอ้างอิง	14
- รูปที่ 3.3 วงจรสร้างสัญญาณนาฬิกา	15
- รูปที่ 3.4 การใช้ฟลิบฟลอบหารความถี่	15
- รูปที่ 3.5 การใช้ฟลิบฟลอบ 4 ตัว	16
- รูปที่ 3.6 วงจรนับ-หาร โหมด 6	17
- รูปที่ 3.7 สัญญาณวงจรนับ	17
- รูปที่ 3.8 โครงสร้าง ไอซี 74LS90	18
- รูปที่ 3.9 ตารางความจริงของ ไอซี 74LS90	18
- รูปที่ 3.10 วงจรหาร 2 ,หาร 10	19
- รูปที่ 3.12 วงจรตัวเลือกข้อมูล 4 อินพุท	20
- รูปที่ 3.13 โครงสร้างไอซี 74LS153	20
- รูปที่ 3.16 วงจรและสัญญาณ วงจรวงแหวน	22
- รูปที่ 3.19 การออกแบบวงจรริงเคาท์เตอร์โดยใช้ D F/F	23
- รูปที่ 3.20 วงจรสร้างแอดเดรสและวงจรเช็คฟูล	24
- รูปที่ 3.21 วงจรทริกเกอร์	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่	หน้าที่

บทที่ 3 ต่อ

- รูปที่ 3.22 วงจรรีเซ็ทระบบ 25
- รูปที่ 3.23 การทำงานของไอซีเบอร์ 74LS123 26
- รูปที่ 3.24 การต่ออุปกรณ์ภายนอกของไอซี 74LS123 27
- รูปที่ 3.25 การทำงานของวงจรโมโนสเตเบิ้ล 27
- รูปที่ 3.27 วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่าง 29

บทที่ 4 หน่วยความจำที่ใช้ในการเก็บข้อมูลจากการแซมปลิ้ง

- รูปที่ 4.1 แรมขนาด 1 บิท 31
- รูปที่ 4.2 โครงสร้างพื้นฐานของแรม 32
- รูปที่ 4.3 แผนผังเวลาในการเขียนข้อมูล 33
- รูปที่ 4.4 แผนผังเวลาในการอ่านข้อมูล 34
- รูปที่ 4.5 แรมเบอร์ 6116 34
- รูปที่ 4.6 การต่อแรมหลายตัวรวมกัน 36
- รูปที่ 4.7 วงจรทดลองเขียนและอ่านข้อมูลจากแรม 6116 เบื้องต้น 37
- รูปที่ 4.8 บล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บข้อมูล 38
- รูปที่ 4.9 การทำงานอินพุทและเอาท์พุทของอินเวอร์เตอร์แบบชมิททริกเกอร์ 39
- รูปที่ 4.10 โครงสร้างของ ไอซี 74LS 244 40
- รูปที่ 4.11 ระดับสัญญาณทางด้านอินพุทและเอาท์พุทของทีทีแอล 40
- รูปที่ 4.12 โครงสร้างของวงจรแลทช์ข้อมูล 41
- รูปที่ 4.13และ4.14 โครงสร้างการทำงานของไอซี 74LS374 42
- รูปที่ 4.15 วงจรที่ใช้ในการเก็บข้อมูลที่ออกแบบเสร็จ 46
- รูปที่ 4.16 การทำงานของแรมทั้ง 4 ตัว 45
- รูปที่ 4.17 สัญญาณการอ่านข้อมูลจากแรม 6116-3 44

บทที่ 5 ส่วนของการติดต่อกับคอมพิวเตอร์

- รูปที่ 5.1 การต่อใช้งานของคอมพิวเตอร์กับอุปกรณ์ภายนอก 47
- รูปที่ 5.2 การนำข้อมูลเข้าและออก 47
- รูปที่ 5.3 การใช้คอมพิวเตอร์ในการควบคุมอุปกรณ์ภายในบ้าน 48
- รูปที่ 5.4 สล็อตบนเครื่องคอมพิวเตอร์ 49
- รูปที่ 5.5 ขาสัญญาณต่างใน I/O slot IBM PC 51
- รูปที่ 5.6 วงจรของการเชื่อมต่อที่ได้ออกแบบไว้ 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่	หน้าที่
บทที่ 5 ต่อ	
- รูปที่ 5.7 โครงสร้างของ 8255	53
- รูปที่ 5.8 โครงสร้างภายในของ 8255	54
บทที่ 6 FLOW CHART และ ผลการทดลอง	
- รูปที่ 6.1 เมนูในการทำงาน	61
- รูปที่ 6.2 การวัดสัญญาณแอดเดรสของระบบ	62
- รูปที่ 6.3 การแสดงในรูปเลขฐาน 2 และ ฐาน 16	62
- รูปที่ 6.4 การวัดสัญญาณริงเคาท์เตอร์ของระบบ	63
- รูปที่ 6.5 การแสดงในรูปเลขฐาน 2 และ ฐาน 16	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ปัญหาและเครื่องมือที่ใช้ในการแก้ปัญหาทางดิจิตอล

สำหรับปัญหาที่เกิดขึ้นทางดิจิตอล มีทั้งปัญหาที่เกิดจาก ซอร์ฟแวร์และฮาร์ดแวร์ ดังนั้น เครื่องมือที่ใช้ในการแก้ปัญหาจึงมีทั้ง ทางด้านซอร์ฟแวร์และฮาร์ดแวร์

1.1 ซอร์ฟแวร์ที่ใช้ในการแก้ไขปัญหาทางดิจิตอล

ซอร์แวร์สามารถแก้ไขปัญหาที่เกิดขึ้นจากซอร์ฟแวร์เอง โดยเฉพาะปัญหาทางโปรแกรม แอสเซมบลี้ ซึ่งมักเกิดจากการเขียนคำสั่งผิดพลาด ทำให้การทำงานผิดพลาด นอกจากนี้ ซอร์ฟแวร์ยังสามารถแก้ไขปัญหาที่เกิดจากฮาร์ดแวร์ได้ เช่น การป้อนโปรแกรมตรวจสอบการทำงานของ ฮาร์ดแวร์ว่าทำงานถูกต้องหรือไม่ ซึ่งโปรแกรมที่ใช้ในการตรวจสอบฮาร์ดแวร์นั้น ผู้ทำการตรวจสอบมักเป็นผู้เขียนขึ้นเอง สำหรับการแก้ไขปัญหาที่เกิดขึ้นจากซอร์ฟแวร์ มีโปรแกรมที่ช่วยในการตรวจสอบแก้ไข เช่น DOS DEBUG

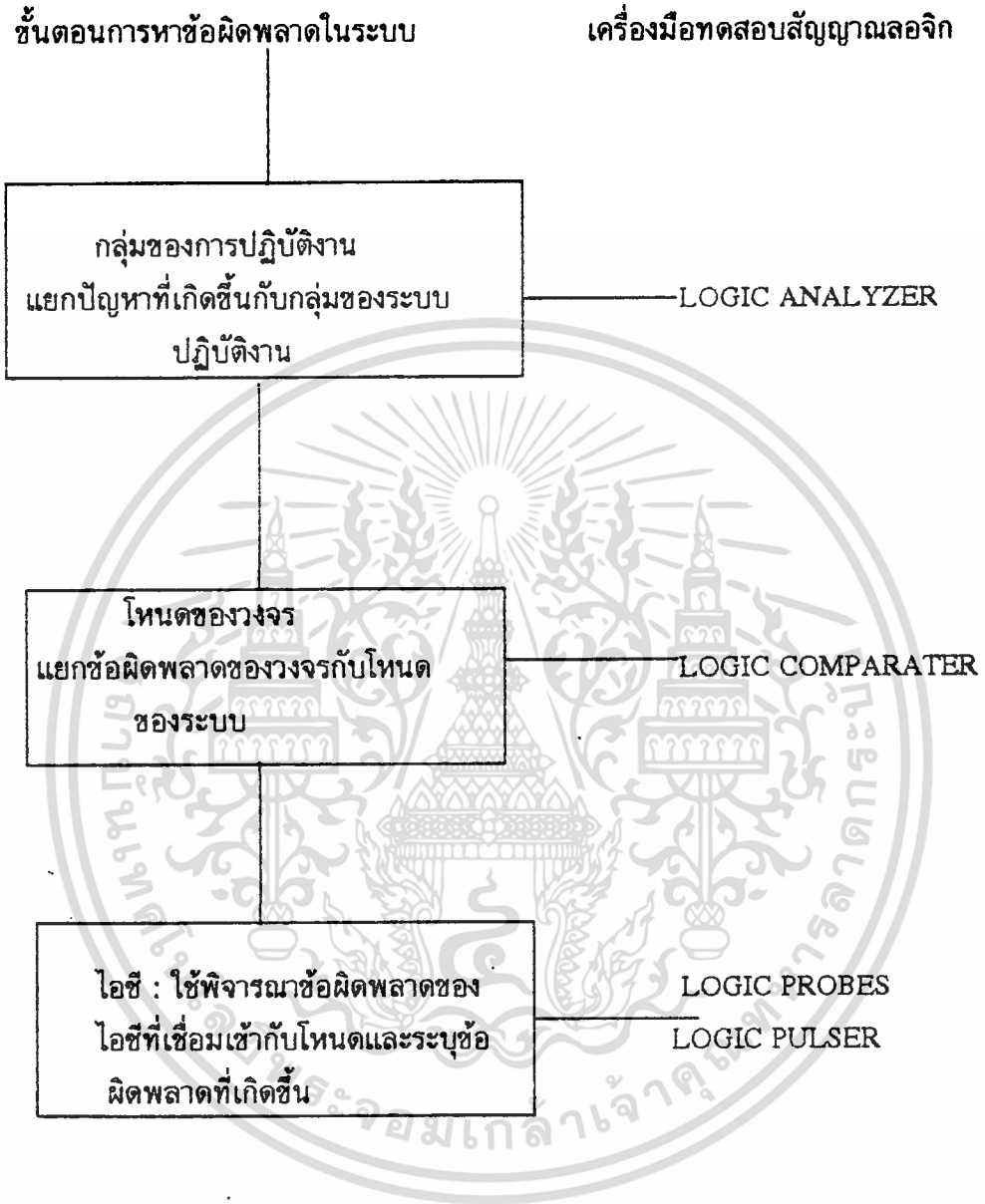
1.2 ฮาร์ดแวร์ที่ใช้ในการแก้ไขปัญหาทางดิจิตอล

สำหรับปัญหาที่เกิดขึ้นในทางฮาร์ดแวร์มีหลายสาเหตุ เช่น การลัดวงจร สายขาด อุปกรณ์เสื่อมคุณภาพหรือเสีย การเปลี่ยนแปลงของระดับลอจิกแล้วเกิดการส่งผ่านตัวเก็บประจุเล็ก ๆ ซึ่งเรียกว่า สไปร์ ความล่าช้าในการเปลี่ยนแปลงสถานะของระดับลอจิก ซึ่งเกิดจากคุณสมบัติของ IC ซึ่งเรียกว่า เรซ(RACE) การใช้อุปกรณ์หลายประเภทร่วมกัน ฯลฯ

ดังนั้นจึงต้องมีเครื่องมือที่ใช้ในการตรวจสอบจับสัญญาณความผิดพลาดที่เกิดขึ้นนี้ ซึ่งมีหลายประเภทด้วยกัน โดยสามารถแบ่งระดับของการใช้งานได้ดังรูปที่ 1.1

จากปัญหาต่าง ๆที่เกิดขึ้นนี้มีทั้งปัญหาที่ง่ายต่อการแก้ไขและยุ่งยากซับซ้อน ดังนั้นการเลือกและใช้งานเครื่องมือที่ใช้ในการตรวจสอบจึงเป็นสิ่งที่จำเป็นเพราะการใช้เครื่องมือที่มีประสิทธิ์ภาพต่ำ อาจทำให้ไม่สามารถพบสาเหตุของปัญหาที่แท้จริงที่เกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 ระดับการใช้งานของเครื่องมือที่ใช้ตรวจสอบสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีการทำงานของลอจิกอนาไลซ์เซอร์

ในปัจจุบันนี้เครื่องมือวัดมีมากมายหลายประเภทเช่น ออสซิโลสโครบ นั้นยังแบ่งออกได้เป็นชนิดต่าง ๆ แตกต่างกันไปทางด้าน สมรรถนะ การใช้งาน และราคาแต่ในที่นี้จะขอแบ่งออกเป็น 2 ประเภทคือ

-ประเภทการใช้งานอเนกประสงค์

-ประเภทการใช้งานเฉพาะอย่าง

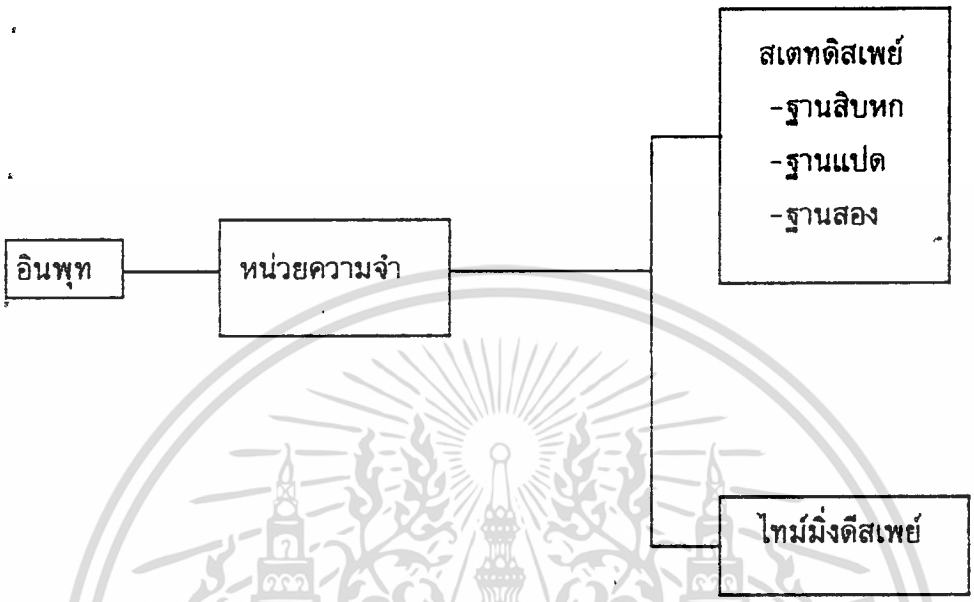
สำหรับการใช้งานอเนกประสงค์ของออสซิลโลสโครบขนาดความถี่ 10 MHz นั้นก็นับว่าเพียงพอแล้ว แต่สำหรับงานที่มีความยุ่งยากมากขึ้นก็อาจใช้แบบ 2 ช่องความถี่ ที่มีแถบความถี่ 15-20 MHz

ส่วนประเภทที่ใช้งานเฉพาะอย่าง ออสซิลโลสโครบที่ใช้ได้แก่ SAMPING OSCILLOSCOPE ,สตอเรจสโครบ (Storage Scope), มอนิเตอร์ริ่งสโครบ (Monitoring scope), ลอจิกอนาไลซ์เซอร์ (Logic Analyzer), ดิจิตอลเมมโมลี่สโครบ (Digital Memory Scope),โปรแกรมเมเบิ้ลสโครบ (Programmable Scope)และสเป็คตรัมอนาไลเซอร์ (Spectrum Analyzer) การจะใช้เครื่องมือประเภทไหนขึ้นอยู่กับสิ่งที่ต้องการตรวจสอบหรือวิเคราะห์ เช่นในการวิเคราะห์ เทอมของศักดาชั่วขณะ (Instantanous Voltage) กับเวลาหรือพวกสัญญาณอนาล็อกที่มีคาบเวลาแน่นอน เราสามารถใช้ ออสซิลโลสโครบธรรมดาวัดได้ ซึ่งวิธีนี้มีประโยชน์ในการวิเคราะห์เกี่ยวกับสัญญาณรบกวน (Noise Burst),ริงกิ้ง (Ringing),สไปค์ (Spike), การสวิงของศักดา (Voltage Swing) ฯลฯ อย่างไรก็ตามในระบบดิจิตอลที่ซับซ้อน (Complex Digital) ส่วนใหญ่จะวิเคราะห์ในเทอมของความสัมพันธ์ทางฟังก์ชั่นของแต่ละสัญญาณ (Function Relation) และสนใจตำแหน่งของข้อมูลดิจิตอลที่มีเข้ามาโดยการเปรียบเทียบกับเวลาของระบบ (System time) ดังนั้นในการใช้ออสซิลโลสโครบธรรมดา จึงไม่สามารถทำได้แต่สำหรับลอจิกอนาไลซ์เซอร์เป็นเครื่องมือที่ออกแบบขึ้นเพื่อวิเคราะห์วงจรดิจิตอล โดยมันสามารถเก็บสะสมและแสดงข้อมูลได้หลายช่องความถี่ พร้อม ๆ กันโดยมีการกำหนดทริกเกอร์ทำให้สามารถเลือกจุดสุ่มสัญญาณที่ต้องการได้ดังนั้นลอจิกอนาไลซ์เซอร์จึงเหมาะสมในงานนี้มาก

2.1ลักษณะการทำงานทั่วไปของลอจิกอนาไลซ์เซอร์

จากรูปที่ 2.1 แสดงให้เห็นถึงการทำงานของลอจิกอนาไลซ์เซอร์ โดยสัญญาณอินพุทที่แซมปลิ้ง หรือสุ่มเอามาจาก Clock ภายในหรือภายนอกแล้วนำเอาข้อมูลไปเก็บในหน่วยความจำ จากนั้นนำข้อมูลในหน่วยความจำไปแสดงที่เอาท์พุท เช่น จอมอนิเตอร์ โดยอาจจะแสดงในรูปของ State Display หรือ timing Display ซึ่งเครื่องนี้มีประโยชน์มากในการแก้ไข Debug ของไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

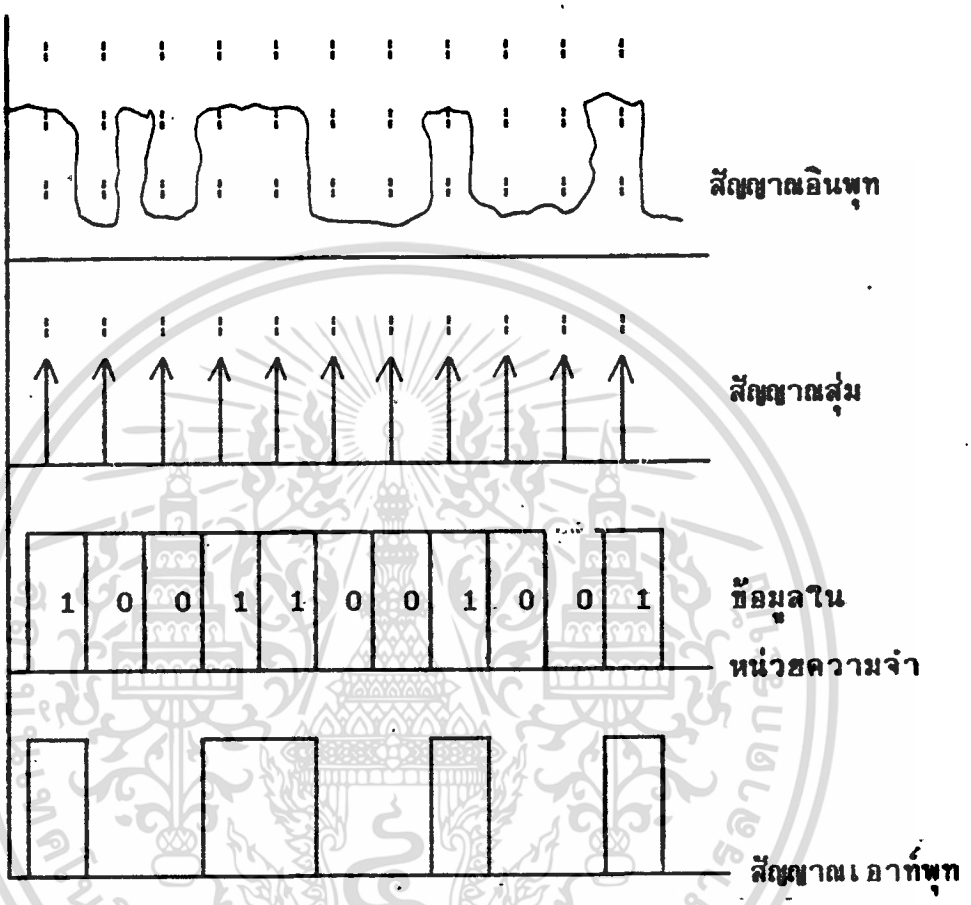


รูปที่ 2.1 หลักการทำงานของลอจิกอนาไลซ์เซอร์

2.2 หลักการแซมปลิ้ง (Sampling) สัญญาณ

การแซมปลิ้งสัญญาณ หมายถึง การสุ่มสัญญาณเป็นช่วง (มีสัญญาณเป็นช่วง ๆมาทำการสุ่ม) จากรูปที่ 2.2 จะเห็นว่าสัญญาณการสุ่มจะเป็นพัลซ์แคบ ๆ สัญญาณการสุ่มข้อมูลจะต้องเป็นสัญญาณคล็อก ที่มีความถี่มากกว่าสัญญารที่เข้ามาทำการสุ่มอย่างน้อยสองเท่าหรือมากกว่า เพื่อที่จะทำให้สัญญาณที่สุ่มออกมามีประสิทธิภาพและมีค่าความถูกต้องสูง โดยในที่นี้เราจะทำการสุ่มสัญญาณเข้าไปเก็บในหน่วยความจำ คือนำสัญญาณที่ได้เป็นช่วง ๆ ไปเก็บไว้ในหน่วยความจำ จากนั้นจะนำข้อมูลจากหน่วยความจำไปทำการประมวลผลบนเครื่องคอมพิวเตอร์ จากนั้นจะแสดงผลออกมาในรูปไทม์มิ่ง,หรือเลขไบนารี่ ลักษณะของสัญญาณจะแสดงดังในรูปที่2.2

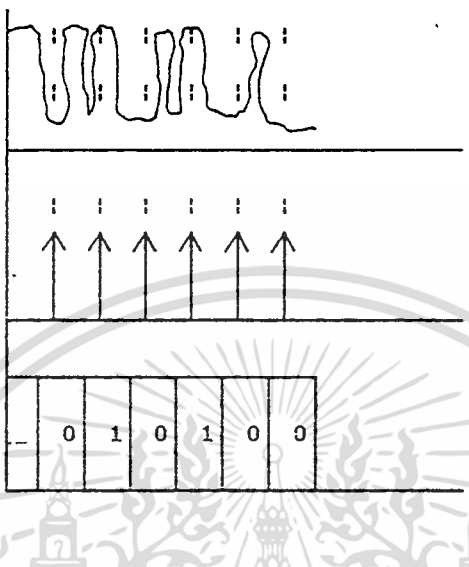
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การสุ่มสัญญาณ

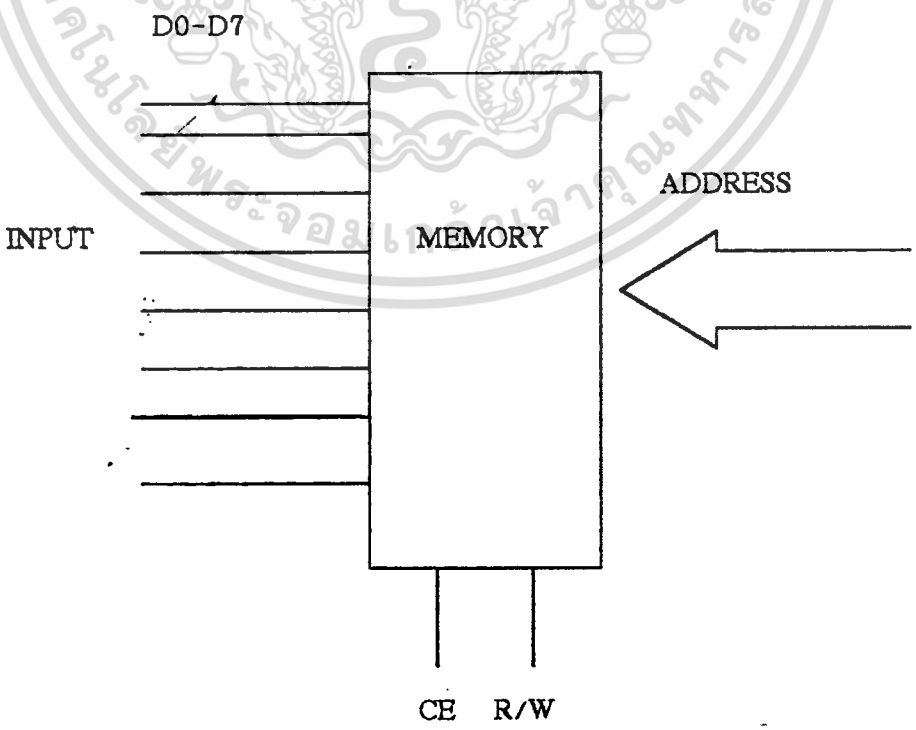
จากรูปที่ 2.2 ถ้าสัญญาณการสุ่มตรงกับสัญญาณอินพุทที่มีค่าแรงดัน 2-5 volt (TTL) จะมีสถานะเป็นลอจิก 1 ก็จะนำค่าลอจิก 1 ที่ได้ไปเก็บไว้ในหน่วยความจำและถ้าสัญญาณการสุ่มตรงกับสัญญาณอินพุทที่มีค่าแรงดัน 0-0.8 volt เป็นลอจิก 0 ก็จะนำค่า ลอจิก 0 ที่ได้ไปเก็บไว้ในหน่วยความจำและจะทำการสุ่มไปเรื่อย ๆ จนข้อมูลเต็มหน่วยความจำจากนั้นก็จะนำข้อมูลที่ได้ไปประมวณผลบนเครื่องคอมพิวเตอร์และจะแสดงผลออกมาในรูปของ timing หรือ binary ก็ได้ จากรูปที่ 2.2 จะเห็นว่าสัญญาณเอาท์พุทที่ได้จะมีค่าความถูกต้องใกล้เคียงกับสัญญาณอินพุทมาก คือสัญญาณในการสุ่มจะต้องมีค่ามากกว่าสัญญาณอินพุทที่เข้ามาอย่างน้อย 2 เท่าหรือมากกว่า ถ้าสัญญาณสุ่มมีค่าน้อยกว่า 2 เท่า คือไม่ถึง 2 เท่า ของความถี่อินพุท ข้อมูลที่ได้จะมีความผิดพลาดสูงจะเห็นได้จากรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การสุ่มที่มีขนาดของสัญญาณสุ่มน้อยกว่า 2 เท่าของสัญญาณอินพุท

2.3 การสุ่มข้อมูลเข้าไปเก็บไว้ในหน่วยความจำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4การสุ่มข้อมูลเข้าไปเก็บไว้ในหน่วยความจำ

ขั้นตอนในการสุ่มข้อมูล

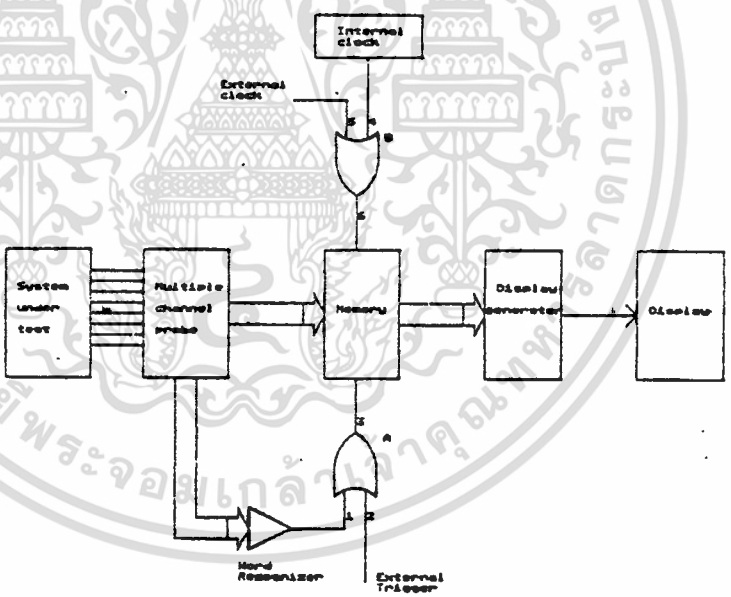
1.จะต้องกำหนด ADDRESS ของหน่วยความจำ

2.เลือกขา CE ให้ทำงานตามลอจิกที่ระบุ CE เป็นสัญญาณบังคับให้หน่วยความจำอันนี้รับรู้สัญญาณควบคุมอื่น ๆ ทั้งหมดซึ่งหมายความว่าหน่วยความจำอันนี้ได้ถูกเลือกให้ทำงานและขา CE นี้ จะต้องต่อกับสัญญาณสุ่มเพื่อเลือกชิบให้เก็บข้อมูลเป็นช่วง ๆ ไปจนกว่าข้อมูลจะเต็มหน่วยความจำแล้วทำการหยุดสุ่ม

3.เลือกขา R/W ให้ทำการเขียนข้อมูล

เมื่อกระทำการขั้นตอนทั้ง 3 ขั้นตอนแล้วข้อมูลจะถูกนำเข้าไปเก็บไว้ในหน่วยความจำตาม ADDRESS ที่เรากำหนดไว้ ถ้าต้องการอ่านข้อมูลก็จะมีลักษณะขั้นตอนเหมือนกับการเขียนข้อมูลเข้าไปเก็บไว้ในหน่วยความจำแต่เวลาอ่านจะต้องเลือกขา Read

2.5 บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์



รูปที่ 2.5 บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์

จากคุณสมบัติของเครื่องลอจิกอนาไลซ์เซอร์ ที่สามารถวัดสัญญาณทางด้านดิจิตอล ได้หลายช่องสัญญาณในเวลาเดียวกันและสัญญาณที่ออกมาจะเป็นในลักษณะ Sequential Digital Data และสามารถที่จะแสดงรูปแบบของสัญญาณได้หลายแบบเช่น timing, binary ,hex ดังนั้นจากบล็อกไดอะแกรมสามารถอธิบายการทำงานได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของเครื่องลอจิกอนาไลซ์เซอร์

1. System Under Test คือ อุปกรณ์ที่จะนำมาใช้ทำการวัดสัญญาณจากเครื่องลอจิกอนาไลซ์เซอร์

2. Multiple channel probe คือตัวตรวจจับสัญญาณจาก System Under Test อาจจะมี 8 Channel หรือ 16 Channel ตามแต่คุณสมบัติของเครื่องนั้น ๆ

3. Memory คือหน่วยความจำที่ใช้ในการเก็บข้อมูลที่ได้จากการวัดสัญญาณเข้ามาซึ่งในการเก็บข้อมูลของเครื่องลอจิกอนาไลซ์เซอร์นี้จะทำการเก็บแบบ FIFO(first in first out)

4. Display Generator คือ ส่วนที่กำเนิดสัญญาณที่จะแสดงทางหน้าจอ

5. Display คือตัวแสดงสัญญาณที่วัดได้

6. Inter-External clock คือสัญญาณนาฬิกาที่ทำหน้าที่ให้สัญญาณที่วัดได้เกิดการ Synce กันระหว่างสัญญาณที่นำมาวัด (เข้ามา) และสัญญาณของเครื่อง อันจะทำให้สัญญาณที่วัดได้มีความถูกต้องและเที่ยงตรงสูง

7.Trigger คือส่วนที่ทำการผลิตสัญญาณพัลซ์ขึ้นมาเพื่อที่จะทำให้เกิดสัญญาณการเริ่มสุ่มข้อมูลที่จะไปเก็บในหน่วยความจำ โดยทั่วไปการทริกนี้มีอยู่ 3 ประเภทดังนี้

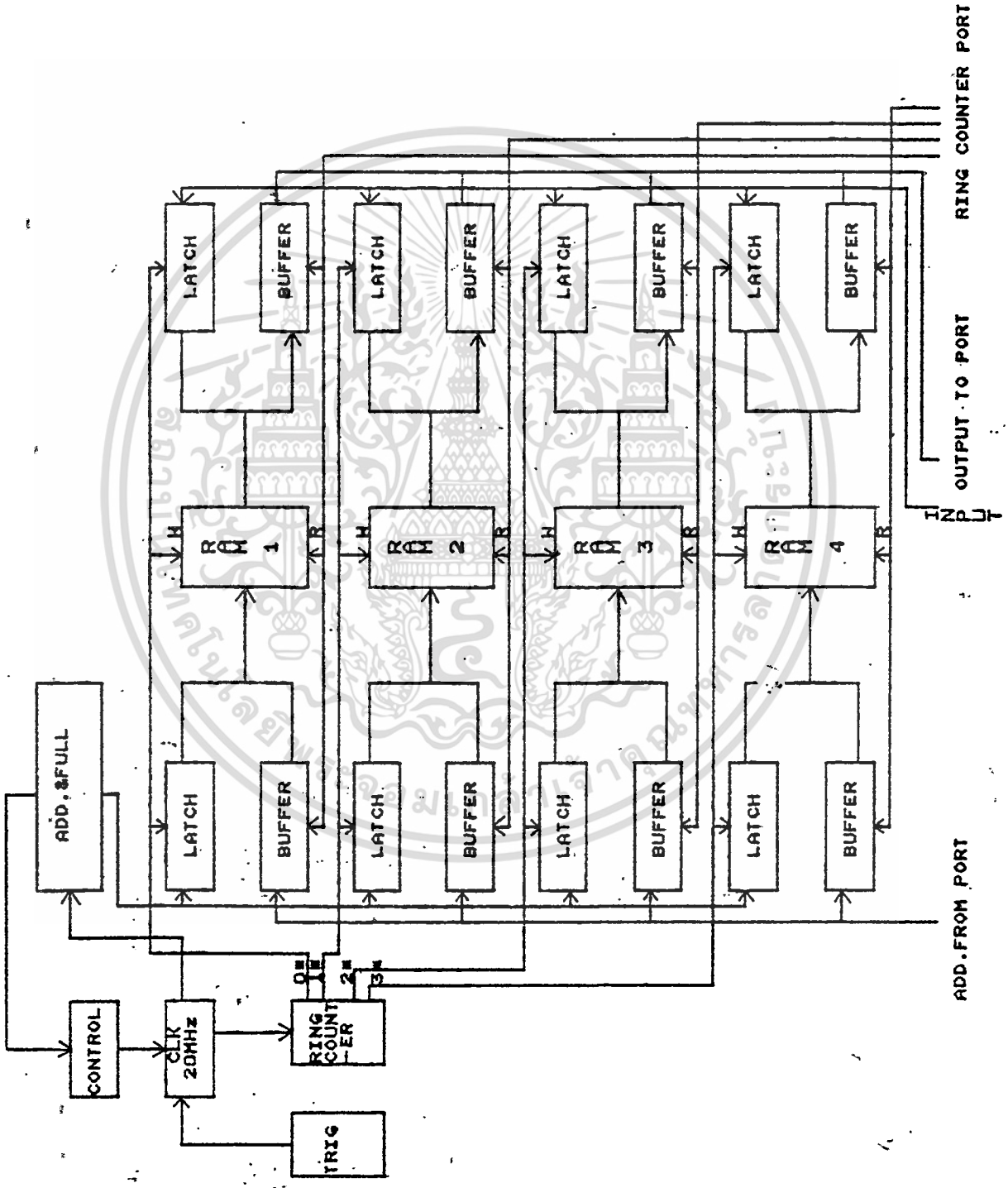
7.1การทริกภายนอก สัญญาณชนิดนี้จะส่งพัลซ์จากภายนอกเข้าไปเพื่อทำให้เกิดการเริ่มสุ่มข้อมูลจากอินพุท ดังนั้นการเก็บข้อมูลใน Memory ก็จะเริ่มขึ้น

7.2การทริกภายใน จะมีการทริกด้วยสัญญาณ ทริกเกอร์เวอร์ด

7.3การทริกแบบ MANUAL เป็นการทริกที่ทำให้เกิดสัญญาณพัลซ์ด้วยมือโดยผู้ใช้ทำการกดเองตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 บล็อกไดอะแกรมในการออกแบบเครื่อง PC LOGIC ANALYZER

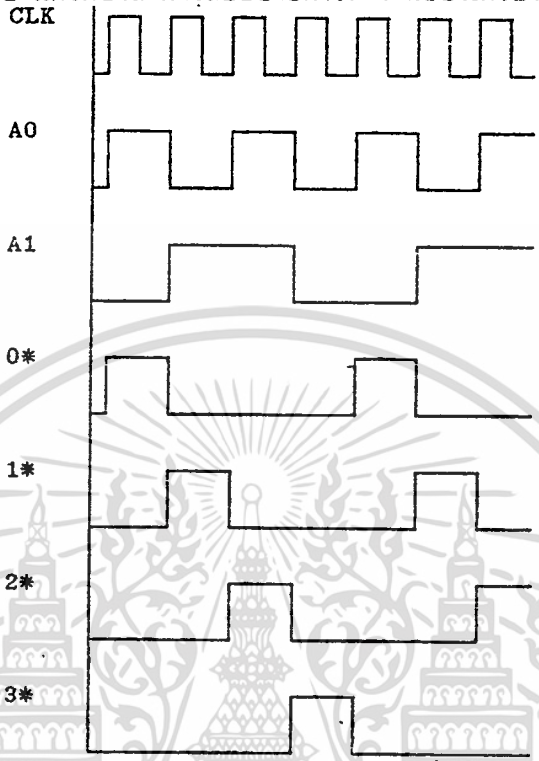


รูปที่ 2.6 บล็อกไดอะแกรมของ PC LOGIC ANALYZER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034748

จากบล็อกไดอะแกรมซึ่งสามารถอธิบายการทำงานของเครื่องลอจิกอนาไลซ์เซอร์ได้ดังนี้



รูปที่ 2.7 สัญญาณการทำงานต่าง ๆ ของเครื่องลอจิกอนาไลซ์เซอร์

A0,A1 แอดเดรสจากวงจรสร้างแอดเดรส

0^{*} ,1^{*} ,2^{*} ,3^{*} เป็นเอาท์พุทของวงจร Ring counter

<u>หลักในการสุ่มข้อมูลเข้าไปเก็บไว้ในหน่วยความจำ</u>

เมื่อให้สัญญาณการทริกแก่ระบบจะให้วงจรสร้างแอดเดรสและวงจร Ring counter ทำงานไปพร้อม ๆ กัน ซึ่งมีการทำงานดังนี้

1.เมื่อเริ่มต้นให้วงจรทำงานที่ตำแหน่ง 00 วงจร Ring counter 0* จะทำการส่งลอจิก 1 ไปแลทแอดเดรสที่ตำแหน่ง 00 และจะทำการแลทข้อมูลอินพุทขณะนั้นไว้ด้วย และสัญญาณ Ring counter 0* จะถูกหน่วงเวลาไว้สักครู่หนึ่งและจะทำการเขียนข้อมูลเข้าไปเก็บไว้ในหน่วยความจำที่แรม 1 ที่ตำแหน่ง 00

2.เมื่อเริ่มให้วงจรทำงานที่ตำแหน่ง 01 วงจร Ring counter 1* จะทำการส่ง ลอจิก 1 ไปแลทแอดเดรสที่ตำแหน่ง 01 และจะแลทข้อมูลอินพุทขณะนั้นไว้ และสัญญาณ Ring counter 1* จะถูกหน่วงไว้สักครู่หนึ่งและทำการเขียนข้อมูลเข้าไปเก็บไว้ใน แรมตัวที่ 2 ที่ตำแหน่ง 01

3.เมื่อเริ่มให้วงจรทำงานที่ตำแหน่ง 10 วงจร Ring counter 2* จะทำการส่ง ลอจิก 1 ไปแลทแอดเดรสที่ตำแหน่ง 10 และจะแลทข้อมูลอินพุทขณะนั้นไว้ และสัญญาณ Ring counter 2* จะถูกหน่วงไว้สักครู่หนึ่งและทำการเขียนข้อมูลเข้าไปเก็บไว้ใน แรมตัวที่ 3 ที่ตำแหน่ง 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.เมื่อเริ่มให้วงจรทำงานที่ตำแหน่ง 11 วงจร Ring counter 3* จะทำการส่ง ลอจิก 1 ไปแลทแอดเดรสที่ตำแหน่ง 11 และจะแลทข้อมูลอินพุทขณะนั้นไว้ และสัญญาณ Ring counter 3* จะถูกหน่วงไว้สักครู่หนึ่งและทำการเขียนข้อมูลเข้าไปเก็บไว้ใน แรมตัวที่ 4 ที่ตำแหน่ง 11

<u>การเขียนข้อมูลลงในหน่วยความจำ</u>

ในการนำข้อมูลเข้าไปเก็บไว้ในหน่วยความจำนั้นจะต้องมีการระบุแอดเดรสตั้งแต่ 0-255 ตำแหน่ง เมื่อมีการเก็บจนครบทุกตำแหน่งแล้วจะทำการหยุดเก็บเพื่อให้มีการแสดงผล แล้วจึงจะมีการเริ่มเก็บข้อมูลใหม่

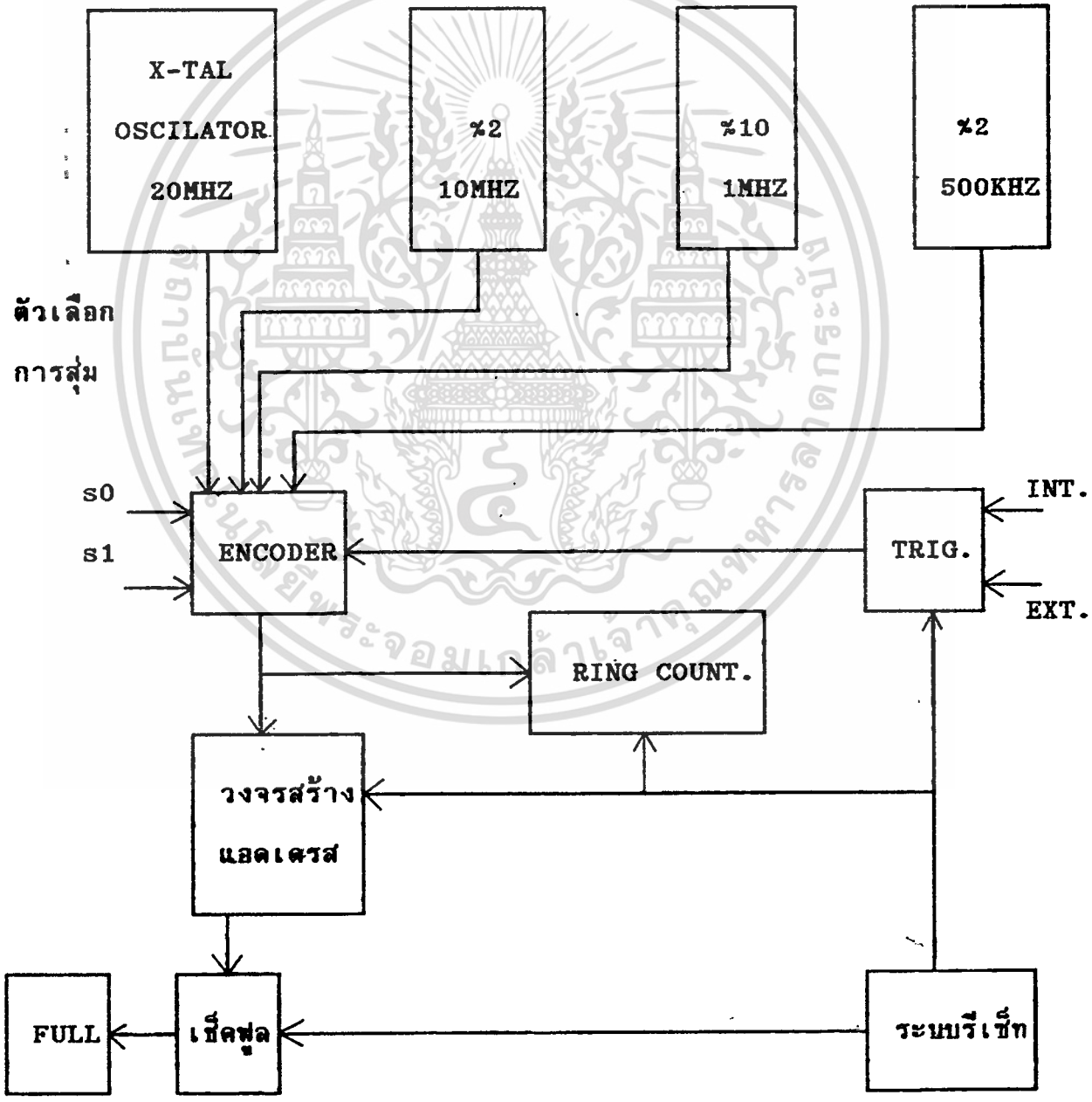
<u>การอ่านข้อมูล</u>

ในการอ่านข้อมูลเครื่อง พีซี จะทำการส่งตำแหน่งแอดเดรสและสัญญาณ Ring counter ออกไปทางพอร์ตเพื่อไปชี้ตำแหน่งที่หน่วยความจำแรมจากตำแหน่ง 0-255 แล้วทำการดึงเอาข้อมูลที่ได้จากแรมเข้ามาที่เครื่อง พีซี แล้วทำการประมวลผลออกมา และแสดงค่าที่ได้จากการประมวลผลบนหน้าจอ เสร็จแล้วก็จะเริ่มทำการสุ่มข้อมูลใหม่ จนข้อมูลเต็มหน่วยความจำแล้วทำการประมวลผลออกมาอีกจะเป็นอย่างนี้เรื่อย ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่าง

วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่างมีความสำคัญมาก สำหรับเครื่องลอจิกอนาไลซ์เซอร์ ซึ่งถ้าระบบมีการควบคุมอย่างถูกต้องแล้วค่าที่วัดได้จากการวัดก็จะมีค่าความถูกต้องสูง ซึ่งวงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่างจะแสดงได้เป็นบล็อก ไดอะแกรมดังรูปที่ 3.1



รูปที่ 3.1 วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถอธิบายการทำงานของวงจรแยกออกเป็นแต่ละบล็อกเพื่อให้สามารถเข้าใจการทำงานของระบบได้ง่ายขึ้น

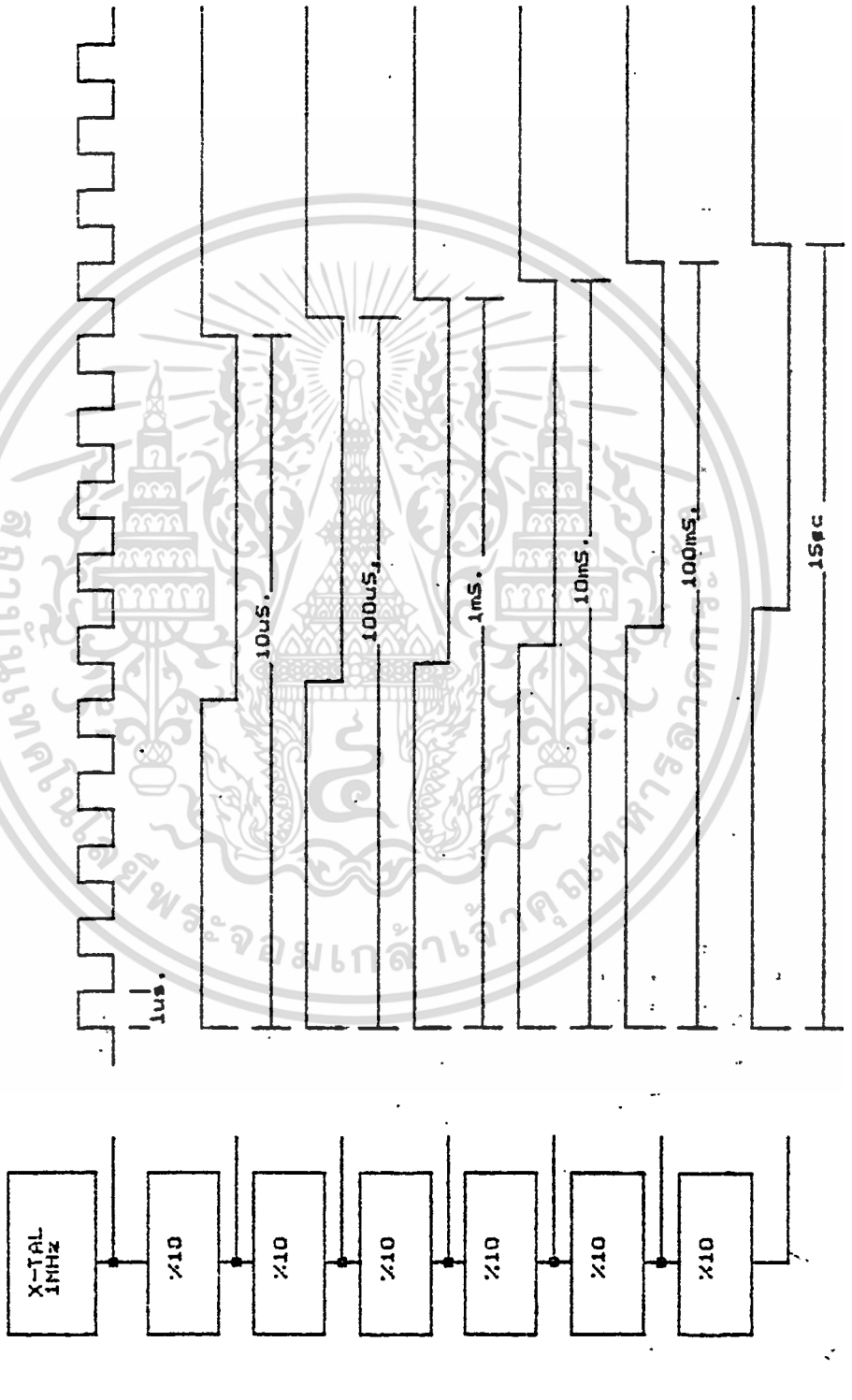
<u>3.1 ระบบเวลา</u>

การทำงานทางด้านดิจิตอลมักจะเกี่ยวข้องกับตัวพารามิเตอร์ที่เป็นเวลาเสมอดังรูปบล็อกไดอะแกรมรูปที่ 3.2 เป็นระบบเวลาที่เราสามารถสร้างเวลาอ้างอิงมาตรฐานได้ถ้าสมมุติว่าคริสตอลที่เราใช้เป็นอุปกรณ์ที่สร้างความถี่คงที่ ซึ่งเหมือนกับเป็นตัวสร้างสัญญาณนาฬิกาและถ้าต้องการให้ความถี่ของสัญญาณนาฬิกาคงที่ยิ่งขึ้นตัวคริสตอลมักจะบรรจุไว้ในอุณหภูมิที่คงที่

ความถี่ที่ออกจาก ฟลิบ ฟลอบ แต่ละตัวจะลดลงมาเรื่อย ๆ หรือในกรณีที่เราใช้ ไอซี เป็นวงจรหาร 10 ความถี่ที่ออกจากวงจรหาร 10 จะมีค่าลดลงไป 10 เท่า นั่นคือถ้าให้ คริสตอลสร้างความถี่ของสัญญาณนาฬิกามีค่า 1 MHz เมื่อใช้วงจรนับ - หาร10
1 ภาค ความถี่จะลดลงมาเหลือ 100 kHz และถ้าหารต่อไปเรื่อย ๆ ก็จะลดลงมาเป็น 10 kHz ,1 kHz,100Hz,10Hzและ 1Hz ถ้าพิจารณาทางด้านเวลาของสัญญาณที่ได้ความถี่ 1 MHz จะมีคาบเวลา 1 us ครั้นเราใช้วงจรหาร 10 หารความถี่เราจะได้คาบเวลา 10 us ,100 us,1ms,10ms,100ms and 1s. มาตรฐานเวลา 1 วินาที นับได้ว่าเป็นเวลาที่ได้จากคริสตอลที่มีค่าความเชื่อถือได้สูง

วงจรสร้างเวลาที่กล่าวมานี้จะมีค่าเท่าไรขึ้นอยู่กับค่าของ คริสตอลและจำนวนภาคของวงจรหาร สำหรับวงจรหารนั้นอันที่จริงแล้ว เราจะหารด้วยตัวเลขเท่าไรก็ได้ แต่การหารด้วยตัวเลขบางค่าจะทำให้การสร้างวงจรยุ่งยากมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

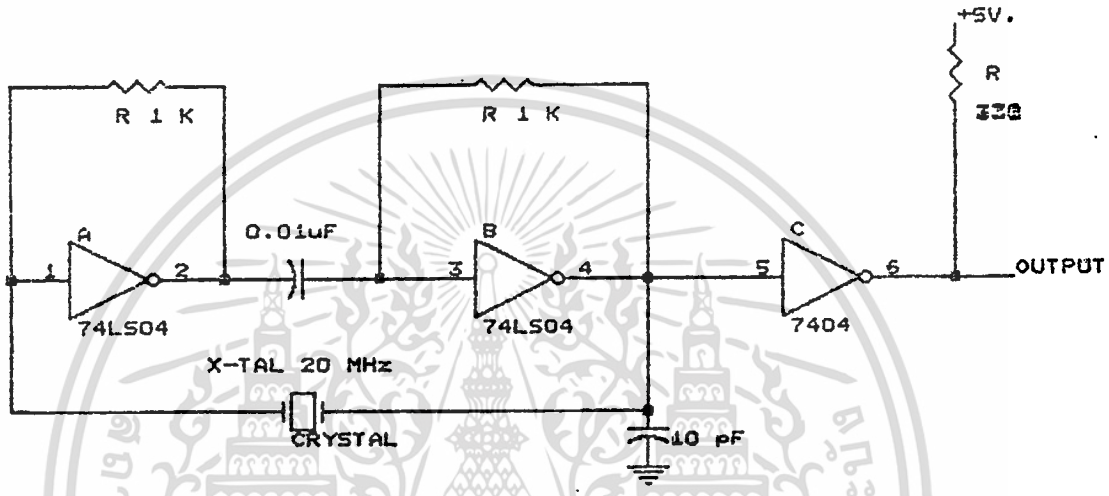


รูปที่ 3.2 วงจรสร้างเวลามาตรฐานอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรสร้างสัญญาณนาฬิกาจากคริสตอล

การออกแบบวงจรสร้างสัญญาณนาฬิกาที่ใช้ในวงจรดิจิตอลนั้นมีอยู่หลายแบบขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน ในกรณีที่นำไปใช้เป็นวงจรฐานเวลาจะต้องใช้วงจรที่มีค่าความเที่ยงตรงสูง ซึ่งปรกติแล้วจะประกอบด้วยวงจรคริสตอล ดังแสดงในรูปที่ 3.3

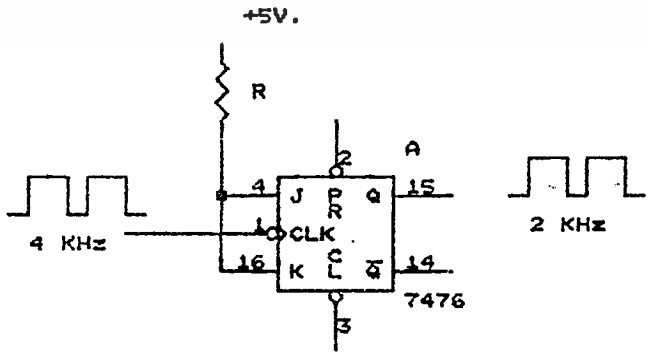


รูปที่ 3.3 วงจรสร้างสัญญาณนาฬิกาจากคริสตอล

จากรูปเป็นการสร้างสัญญาณนาฬิกาที่ใช้ ไอซี ทีทีแอล 2 ตัว มาต่อร่วมกับค่าความต้านทาน R1,R2 ที่ทำหน้าที่เป็นตัวไบอัสให้กับ ไอซี 1,2 จะเห็นได้ว่าค่าของความถี่ที่ได้นี้

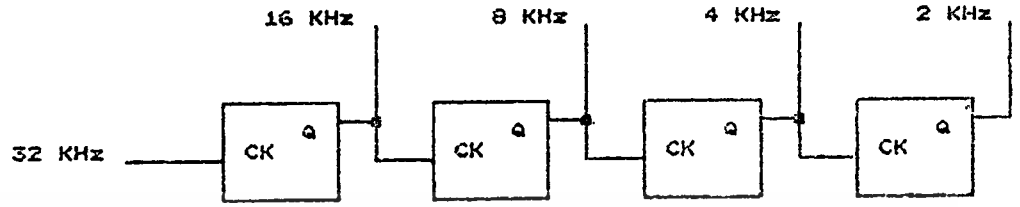
3.3 วงจรหารความถี่

ฟลิบฟลอบมีคุณสมบัติพิเศษคือ ตัวมันสามารถหารความถี่ของสัญญาณนาฬิกาที่อินพุทได้ แต่อย่างไรก็ตามการหารความถี่ของฟลิบฟลอบจะเกิดได้ก็ต่อเมื่อต่อขา J AND K ของฟลิบฟลอบมาอยู่ที่ลอจิก 1



รูปที่ 3.4การใช้ฟลิบฟลอบหารความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การใช้ฟลิบฟลอบ 4 ตัวสามารถหารความถี่ได้ 16

วงจรนับ-หารด้วย n

ในการต่อฟลิบฟลอบ n ตัวเป็นวงจรนับชนิด Ripple จะทำให้การนับเลขไบนารี่เป็นไปได้ถึง 2^n สถานะ เช่นถ้าใช้ฟลิบฟลอบ 2 ตัวก็จะนับได้ 4 สถานะ ถ้าใช้ 3 ตัว ก็จะนับได้ 8 สถานะ ปัญหาที่ตามมาก็คือจะทำอย่างไรเราจึงจะให้วงจรนับหารด้วยค่าตัวเลขใด ๆ ก็ได้ เช่น 3,5,6

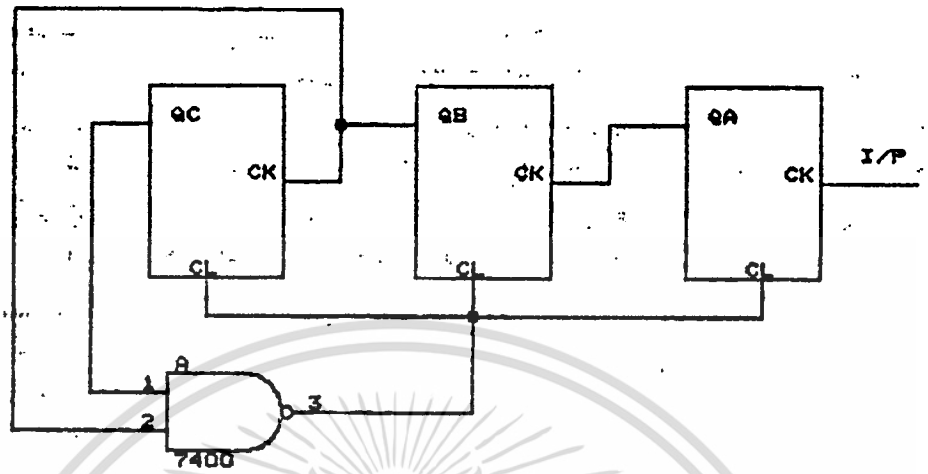
วิธีที่จะทำให้วงจรนับนับได้ถึงค่าที่เราต้องการแล้วเริ่มต้นกลับมานับใหม่ เช่น นับจาก 0-9 แล้วกลับมาเป็น 0 ใหม่นั้น ทำได้หลายวิธีด้วยกัน วิธีแรกเราทำให้วงจรนับหยุดนับเมื่อถึงค่าที่ต้องการ เช่น เมื่อมี ฟลิบฟลอบ 4 ตัว วงจรจะนับจาก 0000-1111 แล้วกลับมาเป็น 0000 ใหม่

<u>หลักการเบื้องต้นในการสร้างวงจรนับ (MOD -N)</u>

คือใช้ขา CLR ของฟลิบฟลอบให้เป็นประโยชน์ โดยทำให้ฟลิบฟลอบถึงสถานะที่เราต้องการแล้วกระโดดข้ามสถานะที่เหลือไปอยู่ที่สถานะ 0 ใหม่ นั่นคือเราต้องการ Detect ค่าของวงจรนับว่าถึงค่าที่เราต้องการแล้วหรือยัง เมื่อถึงแล้วจึงไปทำการเคลียร์ฟลิบฟลอบ

วิธีการที่เราจะออกแบบวงจรนับ MOD-N ทำได้ไม่ยากนัก เราเริ่มด้วยการหาจำนวนฟลิบฟลอบน้อยที่สุดที่ใช้ ซึ่งโดยปรกติถ้าเป็นจำนวน ฟลิบฟลอบ เราจะได้ 2^n มากกว่าหรือเท่ากับ N เช่นถ้าต้องการ MOD-10 เราต้องใช้ ฟลิบฟลอบ 4 ตัว เมื่อได้จำนวนฟลิบฟลอบแล้วก็จะนำฟลิบฟลอบเหล่านั้นมาต่อเป็นวงจร Ripple counter โดยใช้ NAND GATE 1 ตัว โดยให้เอาท์พุทของ NAND GATE ต่อไปยังขา CLR ทุกตัวของ ฟลิบฟลอบ ส่วนอินพุทของ NAND GATE นำมาจากเอาท์พุทที่แสดงลอจิก 1 เมื่อนับถึงสถานะ N หรือแสดงค่าเป็นเลขไบนารี่ N-1

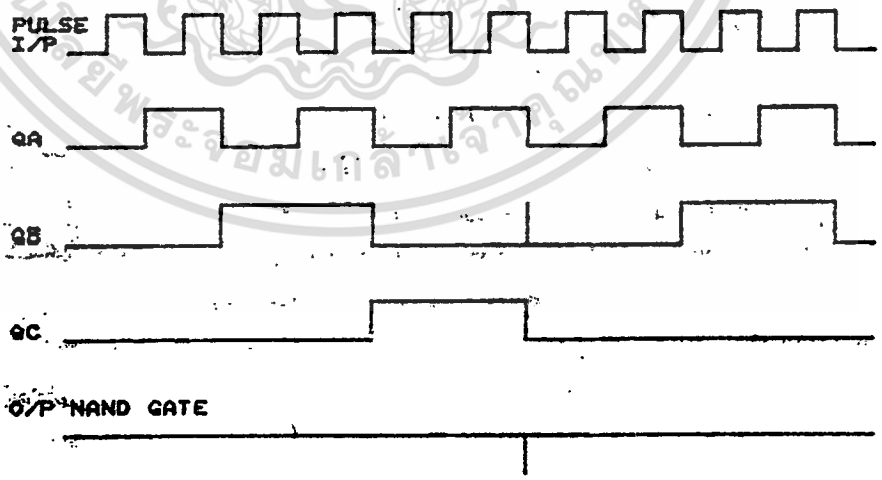
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรนับ-หาร MOD6

ดังตัวอย่างในรูปที่ 3.6 เป็นวงจรนับ-หาร 6 การนับจะนับจาก 000,001,010,011,100,101 และแทนที่จะนับต่อไปเป็น 110 วงจรจะได้รับการเคลียร์ให้เป็น 000 แล้วเริ่มนับใหม่ ลักษณะของรูปคลื่นจะแสดงในรูปที่ 3.7

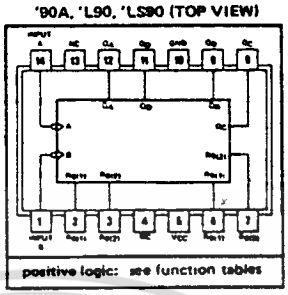
การเคลียร์จะเกิดขึ้นเมื่อขา CLR ของฟลิบฟลอบแสดงสภาวะลอจิกเป็น 0 ข้อเสียของวงจรนับแบบนี้คือมักจะให้เอาท์พุทบางส่วนเป็นพัลซ์แหลม ดังตัวอย่างเช่น พัลซ์จาก เอาท์พุท จาก QB ในขณะเคลียร์ จะมีพัลซ์ลักษณะเป็นพัลซ์แหลมเกิดขึ้น



รูปที่ 3.7 สัญญาณการทำงานของวงจรนับ MOD6

การออกแบบวงจรหารความถี่โดยใช้ ไอซี เบอร์ 74LS90 ซึ่งมีรายละเอียดเกี่ยวกับคุณสมบัติของตัว ไอซี ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โครงสร้างของ ไอซี เบอร์ 74LS90

จากโครงสร้างภายในของ ไอซี เบอร์ 74LS90 เป็นวงจรหาร 2 และหาร 5 คืออินพุท A ให้เอาท์พุท QA เป็นวงจรหาร 2 อินพุท B ให้เอาท์พุท QB,QC and QD เป็นวงจรหาร 5 เมื่อต้องการให้กลายเป็นวงจรนับ 10 จะต่อสัญญาณนาฬิกาเข้าที่ อินพุท A และ QA เข้าที่อินพุท B โดยขา R01 and R02 และ R91 and R92 ต่อเข้าลอจิก 0 ทั้งหมด

เงื่อนไขในการทำงานของขา 0 set คือ R01 and R02 และ 9 set คือ R91 and R92 เป็นดังนี้

-ถ้าให้ขา R01 หรือ R02 และ R91 หรือ R92 มีลอจิกเป็น 0 จะทำให้วงจรนับได้ตามปรกติ

-ถ้าให้ R01 and R02 มีลอจิกเป็น 1 ทั้งคู่และ R91 หรือ R92 ขาใดขาหนึ่งหรือทั้ง 2 ขามีลอจิกเป็น 0 เอาท์พุทของวงจรนับจะมีลอจิกเป็น 0

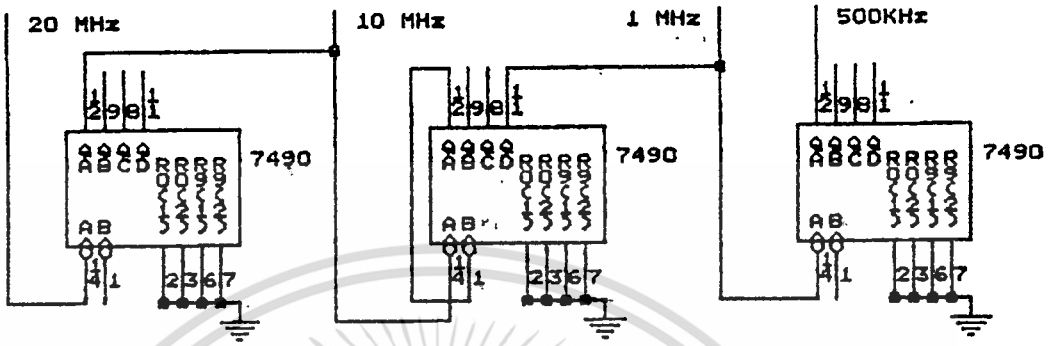
-ถ้าให้ขา R91 and R92 มีลอจิกเป็น 1 ทั้งคู่ส่วนขา R01 and R02 ได้รับลอจิกอะไรก็ได้จะทำให้เอาท์พุทมีค่าเป็น 9 (1001) ซึ่งพิจารณาได้จากตารางความจริง

RESET INPUT				OUTPUT			
R01	R02	R91	R92	QD	QC	QB	QA
1	1	0	X	0	0	0	0
1	1	X	0	0	0	0	0
X	X	1	1	1	0	0	1
X	0	X	0	COUNTER			
0	X	0	X	COUNTER			
0	X	X	0	COUNTER			
X	0	0	X	COUNTER			

รูปที่ 3.9 ตารางความจริงการ Reset และการนับของ ไอ ซี 74LS90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรที่ใช้งานจริง



รูปที่ 3.10 การต่อวงจรหาร 2 หาร10 และ หาร 2

3.5 วงจรเลือกสัญญาณการสุ่ม

วงจรตัวเลือกข้อมูลหรือมัลติเพลกซ์เซอร์ (DATA SELECT OR MULTIPLEXER) จะเป็นวงจร COMBINATION ที่ใช้สำหรับทำหน้าที่เป็นสวิทช์สำหรับเลือกให้เอาท์พุทเป็นค่าใดค่าหนึ่งของอินพุทที่มีหลายอินพุทโดยมีอินพุทเพิ่มแยกต่างหากเข้ามาทำหน้าที่ควบคุมการเลือกให้เป็นไปตามต้องการ

ในขั้นแรกนี้จะลองพิจารณาวงจรเลือกข้อมูลที่มี 4 อินพุทคือ C0,C1,C2,C3 มีตัวเลือกหรือตัวควบคุม (select input) เป็น A,B และมีอีนาเบิ้ลหรือสโตรป G และมีเอาท์พุทเป็น Y ลักษณะการทำงานจะเป็นไปตามตารางความจริงดังนี้

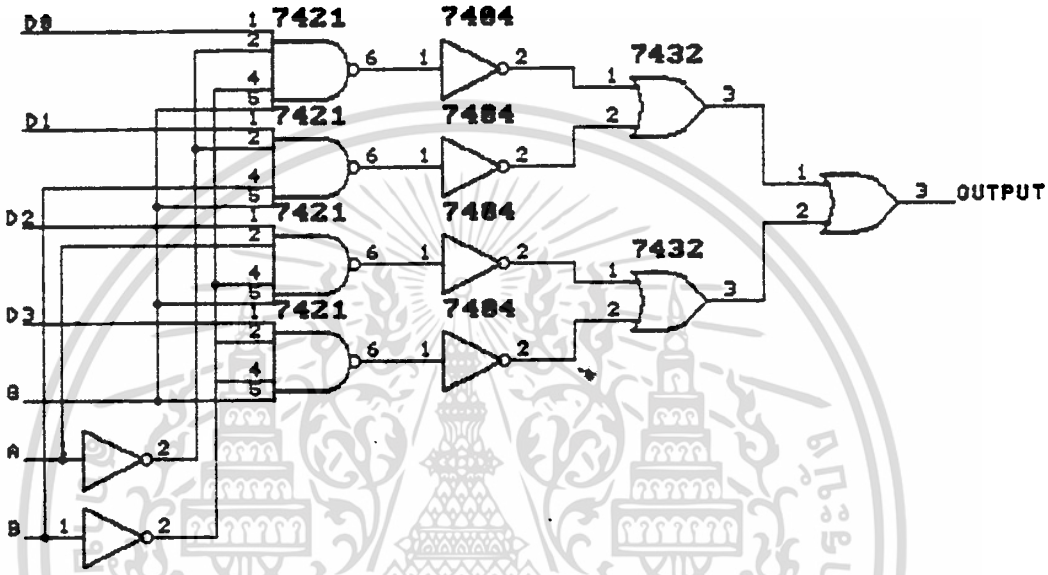
ตัวควบคุมการเลือก		ข้อมูลอินพุท				สโตรบ	เอาท์พุท
A	B	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 ตารางความจริงของวงจรตัวเลือกข้อมูลจาก 4 อินพุท

(Xจะหมายถึงว่า เป็น H หรือ L ก็ได้ วงจรนี้จะทำงานตาม G=0 ซึ่งเรียกวงจรประเภทนี้ว่า Active low) เอาท์พุท Y สามารถเขียนเป็นฟังก์นทางลอจิกและเขียนเป็นวงจรได้ดังนี้

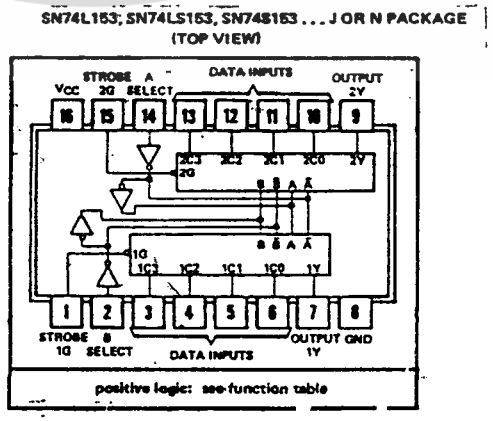
Y=/G/A/BC0+/GA/BC1+/G/ABC2+/GABC3



รูปที่ 3.12 วงจรตัวเลือกข้อมูล 4 อินพุท

<u>การออกแบบวงจรเลือกข้อมูลโดยใช้ ไอ ซี เบอร์ 74LS153</u>

โครงสร้างภายในเป็นวงจรมัลติเพลกซ์เซ้า 4 ออก 1 มี 2 ชุด คือ ข้อมูลอินพุท 1C3,1C2,1C1,1C0 และ 2C3,2C2,2C1,2C0 เอาท์พุทจะเป็น 1Y,2Y ซาสโตรบ 1G ,2G แยกกันควบคุมซา select A,B ใช้ร่วมกันทั้ง 2 ชุด ลักษณะข้อมูลทางเอาท์พุทจะเหมือนกับอินพุทซึ่งพิจารณาได้จากตารางฟังค์ชั่นการทำงาน



รูปที่ 3.13 ซาการใช้งานของ ไอซี เบอร์ 74LS153

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมการเลือก		ข้อมูลอินพุท				สโตรบ	เอาท์พุท
A	B	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L

รูปที่ 3.14 ฟังค์ชั่นการทำงานของ ไอ ซี เบอร์ 74153

ดังนั้นเราจะสามารถเลือกสัญญาณการสุ่มได้ดังนี้และแสดงเป็นตารางได้

S1	S0	ความถี่ เอาท์พุท
0	0	20MHz
0	1	10 MHZ
1	0	1MHZ
1	1	500kHZ

รูปที่ 3.15 การเลือกสัญญาณในการสุ่มข้อมูล

จากตารางการเลือกสัญญาณในการสุ่มนี้จะสามารถโปรแกรมได้และจะโปรแกรมค่าไหนจะขึ้นอยู่กับความต้องการของผู้ใช้ โดยสัญญาณในการสุ่มนี้จะต้องมีค่ามากกว่าสัญญาณที่จะวัดอย่างน้อย 2 เท่า

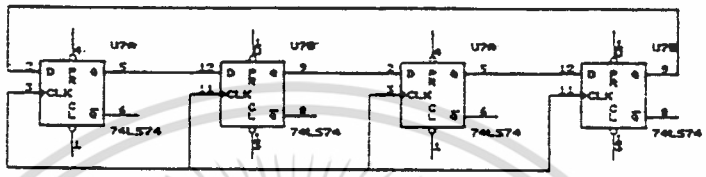
3.6 วงจรนับชนิดวงแหวน

วงจรนับชนิดวงแหวนเป็นวงจรแบบหนึ่งของ ชิบรีจีสเตอร์ ที่มีการป้อนกลับและเป็นแบบที่มีโครงสร้างง่ายมากลักษณะของวงจรแสดงให้เห็นดังรูปที่ 3.16 ซึ่งเป็นวงจรนับชนิดวงแหวน ขนาด 4 บิท ที่ประกอบด้วบฟลิบฟลอบ 4 ตัว

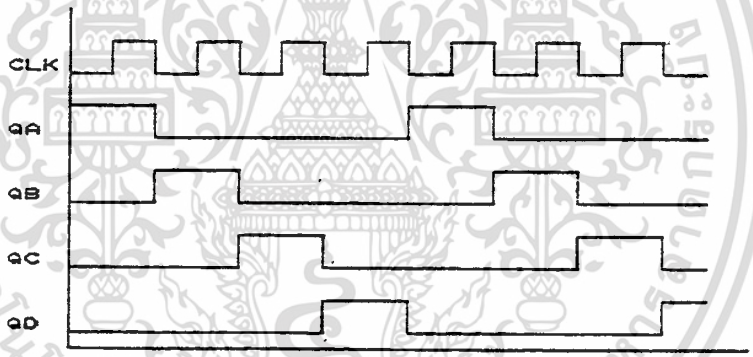
การทำงานของวงจร ก็จะเป็นการนำเอาสัญญาณไหลเวียนกันไปจากตัวแรกจนถึงตัวสุดท้ายแล้วกลับมาเริ่มต้นที่ตัวแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่ตามมาของวงจรมีอยู่ว่าในขณะเริ่มต้นฟลิบฟลอบตัวแรกควรมีสถานะเป็น 1 ส่วนฟลิบฟลอบตัวอื่น ๆ ควรมีสถานะเป็น 0 ซึ่งโดยปรกติเรามีการต่อวงจรเกทเพื่อป้อนให้แก่วงจร



รูปที่ 3.16 ก) วงจรนับชนิดวงแหวน



รูปที่ 3.16 ข) สัญญาณจุดต่าง ๆ

จากปัญหาที่เกิดขึ้นนี้จากการเริ่มสถานะแรกของวงจรแรกของริงเค้าเตอร์ เราจะต้องป้อนลอจิก 1 ให้แก่วงจรซึ่งเราจะสามารถออกแบบวงจรใหม่ได้ดังนี้

<u>วงจรนับแบบวงแหวน MOD 4 ที่ไม่ติดค้าง</u>

เรามีวิธีในการเลือกนิพจน์ของ D0 ได้หลายวิธี ขึ้นอยู่กับว่าจะให้ค่า X ต่าง ๆ มีค่าเป็น 0 หรือ 1 เช่น D0=Q3,D0=Q2Q3,D0=Q0Q1Q2 เราตรวจสอบได้ว่าในสองกรณีแรกจะติดค้างที่สถานะ 0000 ดังนั้นจะใช้ D0=Q0Q1Q2

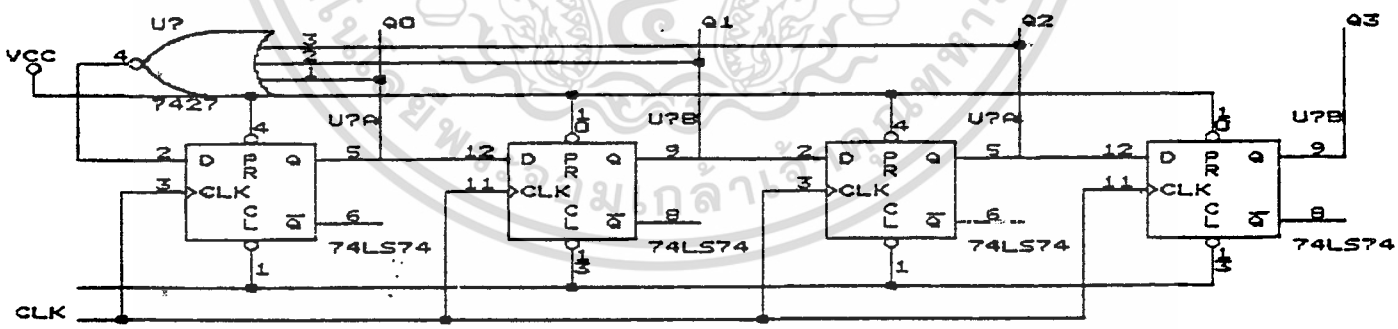
Q0	Q1	Q2	Q3	D0
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	1

รูปที่ 3.17 ตารางลำดับสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q_0Q_1 / Q_2Q_3	00	01	11	10	
00	d	0	d	0	$D_0=Q_0Q_1Q_2$
01	1	d	d	d	$D_0=Q_2Q_3$
11	d	d	d	d	$D_0=Q_3$
10	0	d	d	d	

รูปที่ 3.18 แผนภูมิคาร์นอร์ของ D0

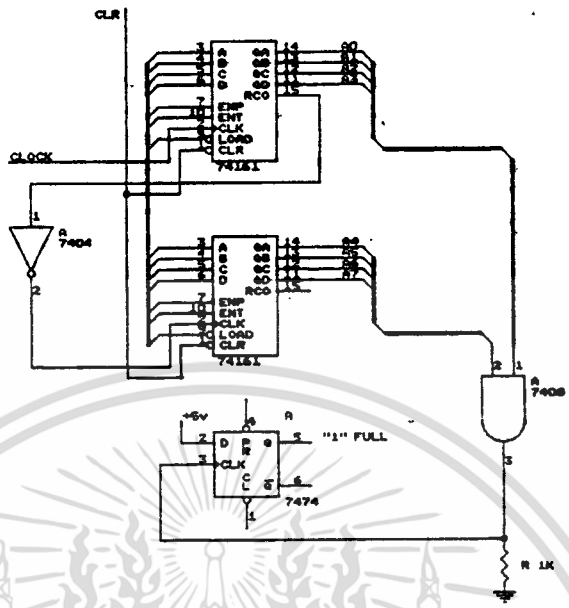


รูปที่ 3.19 การออกแบบวงจรริงเคาท์เตอร์โดยใช้ D ฟลิบฟลอบ 74LS74

3.7 วงจรสร้างสัญญาณแอดเดรสและตัวเช็คฟูล

วงจรนี้จะทำหน้าที่สร้างแอดเดรสให้แก่หน่วยความจำแรมเบอร์ 6116 จากรูปที่ 3.20 เมื่อมีสัญญาณเริ่มสุ่มข้อมูลเข้ามาวงจรก็จะทำหน้าที่สร้างแอดเดรสให้แก่แรม 6116 เพื่อใช้ระบุตำแหน่งของข้อมูลที่จะนำไปเก็บในหน่วยความจำเริ่มตั้งแต่แอดเดรสที่ 00H-FFH หรือเท่ากับ 256 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 วงจรสร้างสัญญาณแอดเดรสและวงจรเช็คฟูล

จากวงจรเมื่อวงจรนับครบ 256 ตำแหน่งจะทำให้เอาท์พุทของ แนนเกทเป็นลอจิก 0 เมื่อผ่าน NOT GATE จะทำให้ได้ ลอจิก 1 ซึ่งสัญญาณที่ได้จะส่งไปทริกไอซี 74LS74 ให้ส่งลอจิก 1 ไปที่ ฟูล เพื่อระบุว่าขณะนี้ข้อมูลในหน่วยความจำเต็มแล้ว (256 ตำแหน่ง) หลังจากเก็บข้อมูลในหน่วยความจำแล้วสัญญาณฟูลที่ได้จะฟ้องเครื่องพีซีให้ทำการอ่านข้อมูลจากแรม 6116 เพื่อไปทำการประมวลผลและแสดงผลที่ได้ออกมาทางจอภาพ

3.8 ทริกเกอร์

ทริกเกอร์คือส่วนที่ทำการผลิตสัญญาณพัลซ์ขึ้นมาเพื่อที่จะทำให้เกิดการส่งข้อมูลที่จะนำไปเก็บไว้ในหน่วยความจำโดยทั่วไปทริกเกอร์นี้มีอยู่ 3 ประเภทคือ

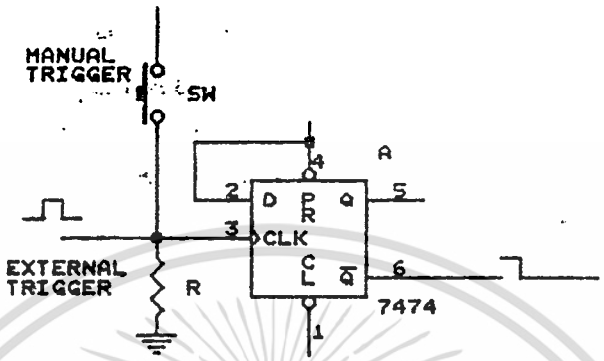
1.การทริกภายนอก สัญญาณทริกชนิดนี้จะส่งพัลซ์จากภายนอกเข้าไปเพื่อทำการให้เกิดการสุ่มข้อมูล จากอินพุทให้เข้าไปเก็บใน หน่วยความจำ

2.การทริกภายใน จะมีการทริกด้วยสัญญาณทริกเกอร์เวอร์ด(trigger word)

3.การทริกแบบ Manual เป็นการทริกที่ทำให้เกิดสัญญาณพัลซ์ด้วยมือ โดยผู้ใช้ทำการกดเองตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรทริกเกอร์



รูปที่ 3.21 วงจรทริกเกอร์

หลักการทำงานของวงจรทริกเกอร์

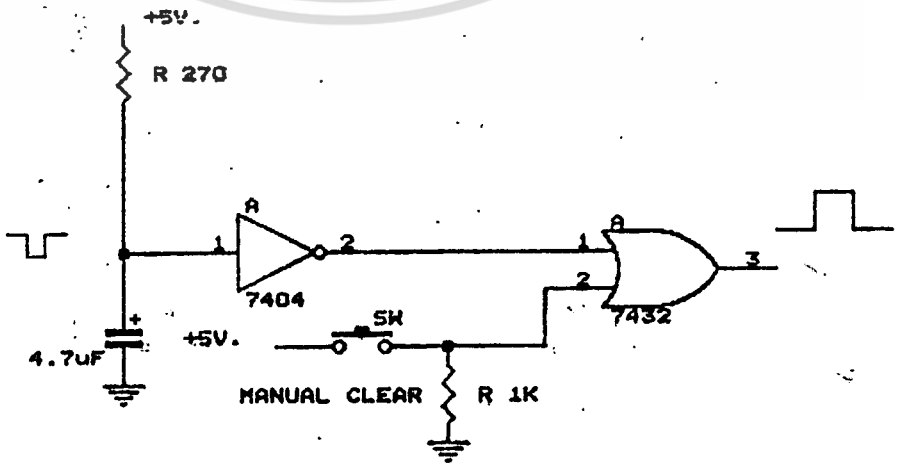
วงจรจะสามารถทริกได้ 2 แบบคือ external trigger and manual trigger

-external trigger เมื่อมีสัญญาณพัลซ์ขอบขาขึ้นจากภายนอกเข้ามาจะทำให้เอาท์พุทไอซีเบอร์ 74LS74 /Q มีสถานะเป็น 0 สัญญาณที่ได้นี้จะนำไปให้วงจรสุ่มข้อมูลเริ่มทำการสุ่มข้อมูลเข้าไปเก็บไว้ในหน่วยความจำ

-manual trigger เมื่อเรากดสวิทช์จะทำให้มีสัญญาณพัลซ์ขอบขาขึ้นไปทริกไอซีเบอร์ 74LS74 /Q จะมีสถานะเป็นลอจิก 0 สัญญาณที่ได้นี้จะนำไปให้วงจรสุ่มข้อมูลเริ่มทำการสุ่มข้อมูลเข้าไปเก็บภายในหน่วยความจำ

3.9 ระบบรีเซ็ทวงจร

วงจรรีเซ็ทจะทำการรีเซ็ทระบบให้ระบบสามารถเริ่มการทำงานได้ ซึ่งประกอบด้วย manual clear , auto clear และสามารถทำการเคลียร์ระบบด้วย Sofware ก็ได้ ซึ่งวงจรจะแสดงดังรูปที่ 3.22



รูปที่ 3.22 วงจรรีเซ็ทระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรรีเซ็ท

-auto clear เมื่อเริ่มจ่ายไฟเลี้ยงให้แก่วงจร C1จะเริ่มชาร์จประจุ ขณะเริ่มชาร์จจะมีแรงดันเป็น 0 โวลท์ชั่วขณะ เมื่อผ่านไปที่ not gate จะทำให้มีลอจิกเป็น 1 ซึ่งจะนำค่าที่ได้นี้ไปทำการรีเซ็ทระบบเริ่มแรกเมื่อเวลาผ่านไป C1 จะชาร์จประจุจนเต็มและจะมีแรงดันประมาณ 5 volt จึงทำให้ not gate มีลอจิกเป็น 0 ซึ่งจะหมายความว่าระบบได้ทำการรีเซ็ทตัวเองเรียบร้อยแล้ว

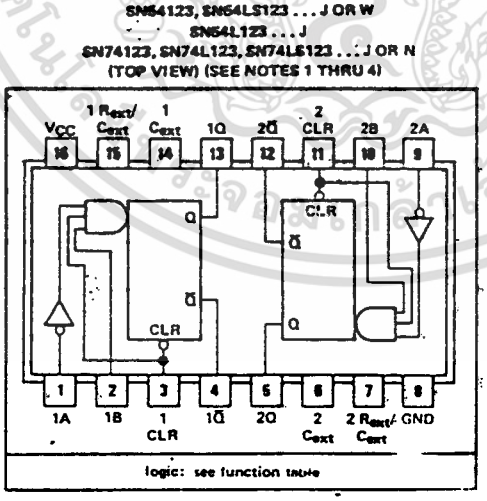
-manual clear เมื่อกดสวิทช์จะทำให้มีแรงดันประมาณ 5 โวลท์ตกคร่อม R2 ซึ่งจะทำให้เอาท์พุทของ OR gate มีลอจิกเป็น 1 ซึ่งจะทำให้เกิดการรีเซ็ทระบบเริ่มขึ้น เมื่อปล่อยสวิทช์ก็จะทำการหยุดรีเซ็ทระบบ

3.10 วงจรโมโนสเตเบิ้ล

เนื่องจากการอ่านข้อมูลจากแรมไม่ซิ้งกันกับเครื่องคอมพิวเตอร์จึงจำเป็นต้องสร้างวงจรโมโนสเตเบิ้ลขึ้นมาเพื่อที่จะทำให้ระบบซิ้ง จึงจะทำการอ่านข้อมูลจากแรมได้

การออกแบบจะใช้ ไอซี เบอร์ 74LS123 ซึ่งเป็นวงจรโมโนสเตเบิ้ลที่สามารถจุดฉนวนซ้ำได้คือหลังจากถูกจุดฉนวนซ้ำเอาท์พุทของพัลซ์จะเริ่มประวิงเวลาใหม่ โดยไม่สนใจว่าเอาท์พุทเดิมเวลาจะผ่านไปเท่าไร

จากตารางการทำงานของ IC เบอร์ 74LS123 การจุดฉนวนจะเกิดขึ้นเมื่อ อินพุท A เป็นลอจิก 0 B,CLR เป็นลอจิก 1



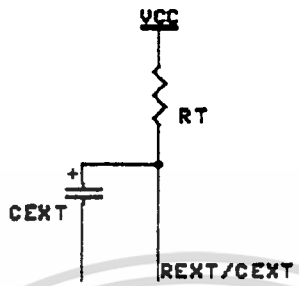
'123, 'L123, 'LS123
FUNCTION TABLE

INPUTS			OUTPUTS	
CLEAR	A	B	Q	Q̄
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⎎
H	↓	H	⎍	⎎
↑	L	H	⎍	⎎

รูปที่ 3.23 การทำงานของ IC เบอร์ 74LS123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรโมโนสเตเบิลโดยใช้ IC เบอร์ 74LS123



รูปที่ 3.24 การต่ออุปกรณ์ภายนอกของ ไอ ซี เบอร์ 74LS123

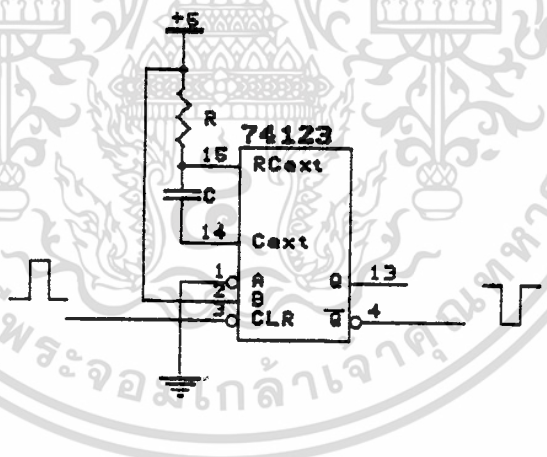
ความกว้างของพัลซ์สามารถพิจารณาได้จากสูตร

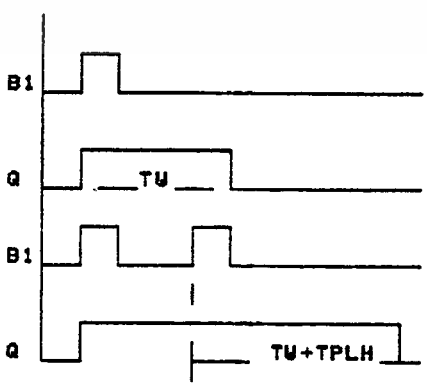
$$tw = 0.45 Rt * cext$$

เมื่อ Rt มีหน่วยเป็น $K\Omega$

cext " pf

tw " ns





รูปที่ 3.25 การทำงานของวงจรโมโนสเตเบิ้ล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 fanout

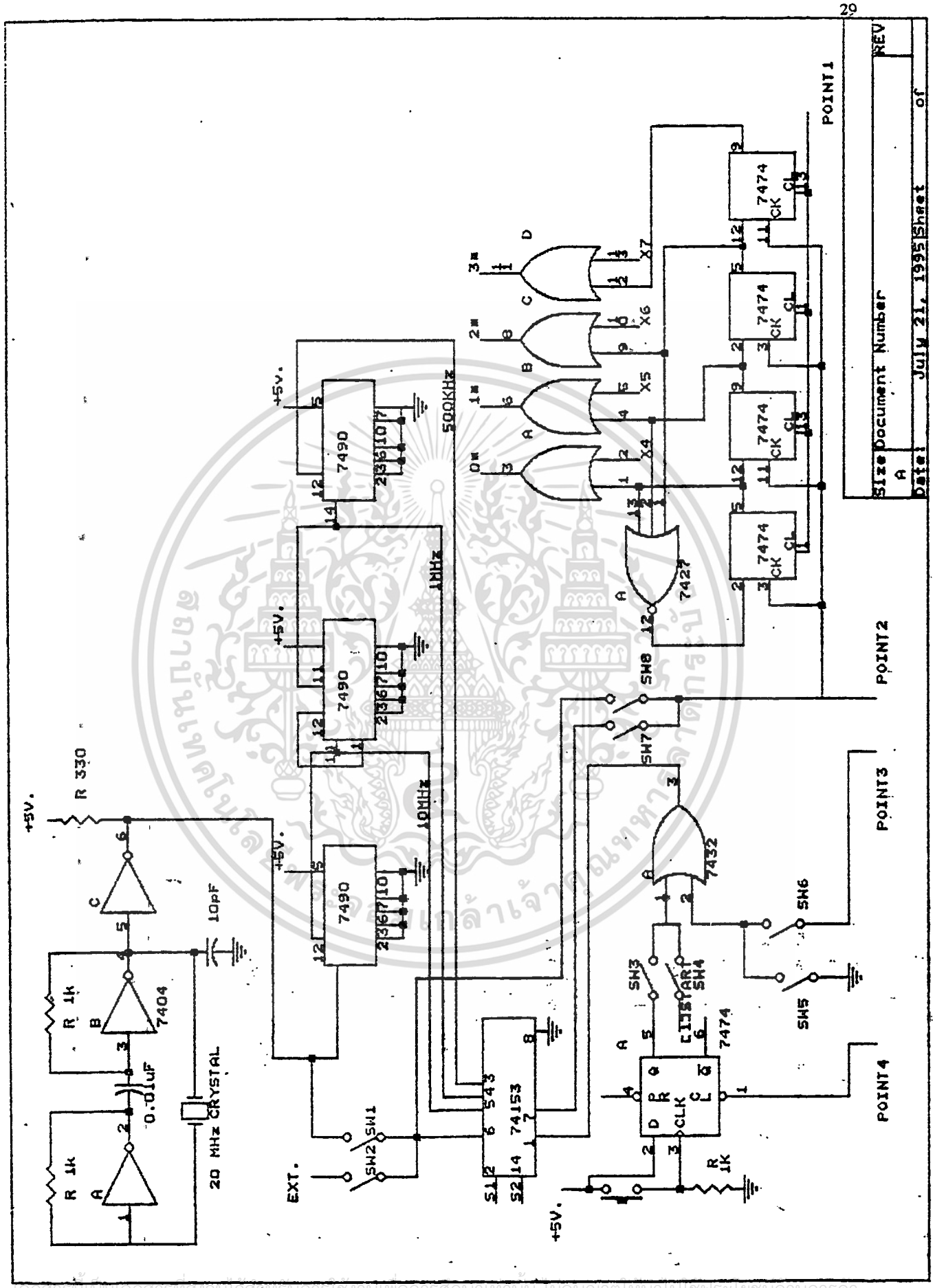
fanout คือความสามารถในการจ่ายกระแสทางเอาท์พุทให้กับเกทอื่น ๆหรือเอาท์พุท 1 ตัวสามารถต่อกับอินพุทได้กี่ตัวโดยไม่ทำให้แรงดันเอาท์พุทผิดพลาดไปจากค่าที่ใช้งานจากตาราง แสดงการเปรียบเทียบ fanout ระหว่าง ตระกูล ทีทีแอล กับ ซีมอส

เครื่องหมาย 1/2 หมายถึงถ้านำ 2 ตัวมาชนานกันก็จะนับได้ 1 ตัว สำหรับค่าแรงดัน ของ ซีมอส นั้นเป็น 5 โวลท์ เมื่อเปรียบเทียบกับตระกูลต่าง ๆ ก็จะได้

อินพุท เอาท์พุท	ทีทีแอล ธรรมดา	แ อ ล ที ที แอล	เฮชทีทีแอล	เอสทีทีแอล	แอล เอส ทีทีแอล	ซีมอส
ทีทีแอล	10	40	6	6	20	50
แ อ ล ที ที แอล	2	10	1	1	5	10
เฮช ทีที แอล	12	40	10	10	40	50
เอส ทีที แอล	12	40	10	10	40	50
แอล เอส ที ทีแอล	5	20	4	4	10	50
ซีมอส	1/2	1/2	1/2	1	1	50
เฮช ซี หรือ เฮชซีที	2	10	1	1	5	50

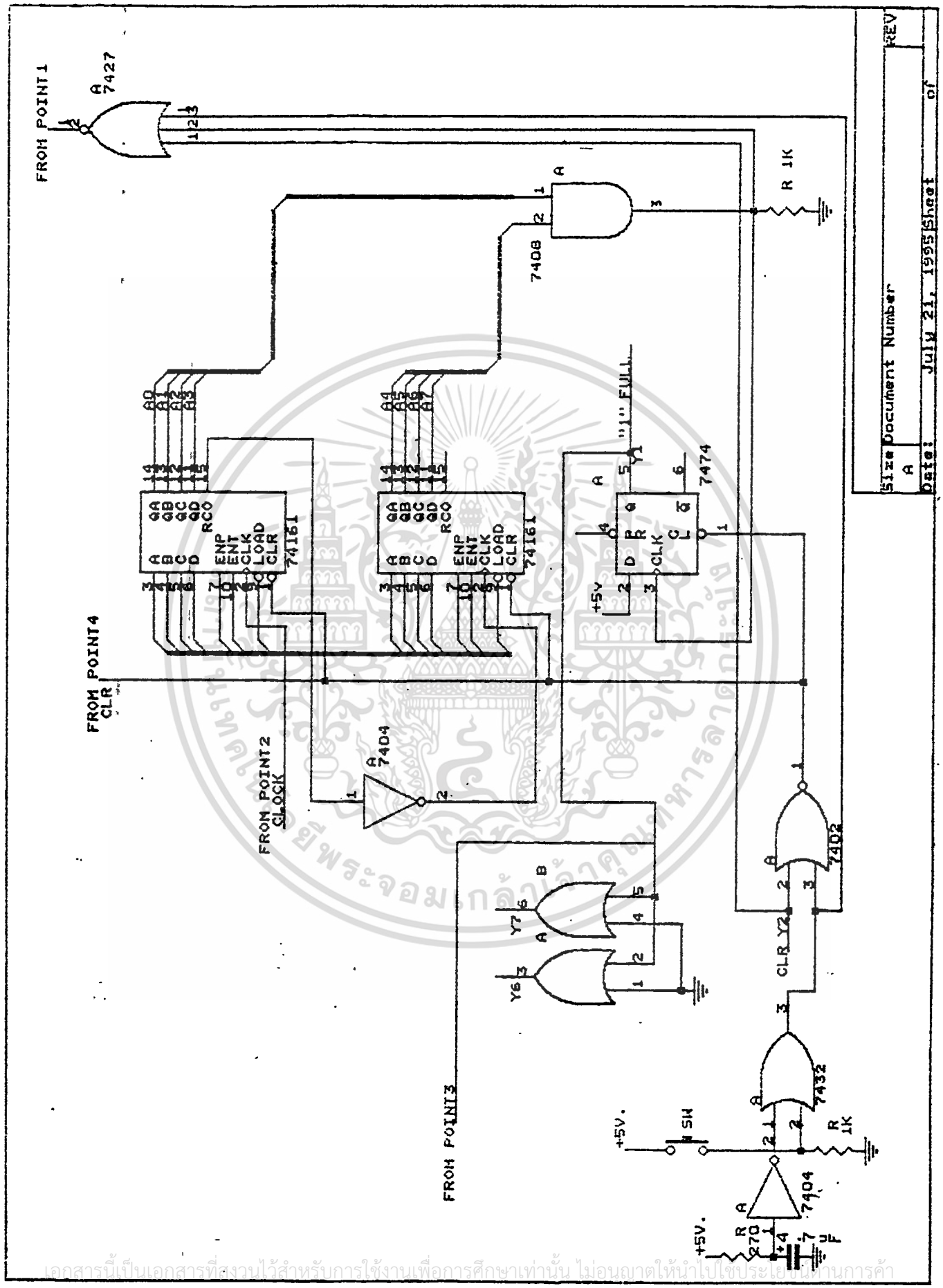
รูปที่ 3.26 ตารางการเปรียบเทียบ fanout ของไอซีลอจิกแต่ละตระกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 วงจรสร้างฐานเวลาและระบบควบคุมการสุ่มตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
หน่วยความจำที่ใช้ในการเก็บข้อมูลจากการแซมปลิ้ง

หน่วยความจำที่ใช้ในเครื่อง PC logicanalyser เครื่องนี้จะใช้ RAM ในการเก็บข้อมูล หน่วยความจำนี้จะสามารถเรียกข้อมูลที่ตำแหน่งใด ๆ ออกมาก็ได้โดยไม่ต้องเรียงลำดับตำแหน่ง การนำเอาข้อมูลเข้าไปเก็บในหน่วยความจำเป็นไปได้ง่ายเหมือนกับการเรียกข้อมูล แต่ข้อมูลที่ถูกเก็บจะยังคงสภาพได้เมื่อมีไฟเลี้ยงวงจรเท่านั้น RAM นี้บางครั้งเรียกว่า หน่วยความจำลบเลือนได้ (volatile memory) และถ้าต้องการเขียนข้อมูลเข้าไปใหม่ ข้อมูลนี้จะทับข้อมูลเดิมที่มีอยู่ใน RAM

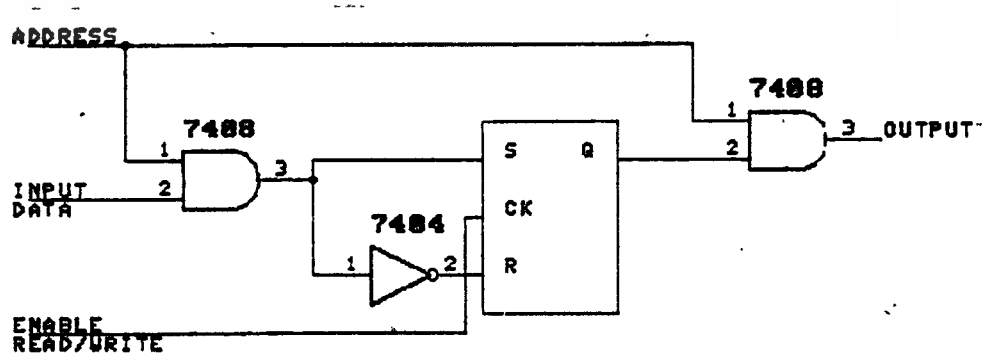
4.1ชนิดของ RAM

เราสามารถแบ่งประเภทออกได้เป็น 2 กลุ่มคือ สแตติกและไดนามิกแรม การทำงานของแรมแบบสแตติกไม่ต้องมีสัญญาณนาฬิกามาเลี้ยงวงจรโดยจะยังคงสามารถเก็บข้อมูลไว้ได้ตราบที่ยังมีไฟเลี้ยงวงจรอยู่ แรมแบบสถิตย์นี้ปกติจะใช้ทรานซิสเตอร์ชนิด ไบโพลาร์ทรานซิสเตอร์หรือทรานซิสเตอร์ชนิดมอสเป็นตัวเก็บข้อมูล

สำหรับแบบไดนามิกประกอบด้วยตัวเก็บประจุและมอสเฟ็ทที่เป็นตัวเก็บข้อมูล แต่เนื่องจากข้อมูลที่เก็บในตัวเก็บประจุจะค่อย ๆ ลดน้อยลงและหายไปในที่สุดจึงจำเป็นต้องมีวงจรเติมประจุข้อมูลเดิมเข้าไปเก็บใหม่ดังที่เดิมการเติมประจุเช่นนี้เรียกว่า ฟื้นฟูสถานะ (Refresh) ซึ่งจำเป็นต้องทำการฟื้นฟูทุก ๆ 2-3 ms. โดยเอาสัญญาณนาฬิกาที่แยกต่างหากต่อเข้ากระตุ้นวงจรเพื่อรักษาข้อมูลให้คงเดิม

4.2 วงจรแรมเบื้องต้น

แรมเป็นวงจรที่เก็บข้อมูลตัวเลขที่เป็นไบนารี่ได้ถ้าเก็บตัวเลขไบนารีได้ 1 บิทเราเรียกว่า 1 เซล แรมที่ใช้งานอยู่ทั่วไปจึงประกอบด้วยเซลหน่วยความจำเป็นจำนวนมาก ถ้าเราต้องการใช้หน่วยความจำแค่เพียงเซลเดียวเราอาจสร้างได้จากฟลิบฟลอบอย่างง่าย ๆ ตามรูปที่ 4.1 เป็นวงจรแรมขนาด 1 บิท

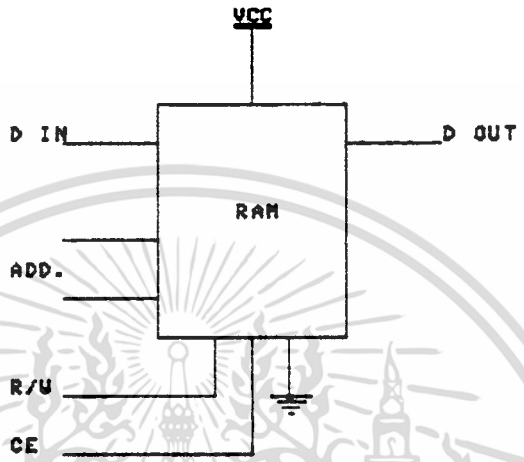


รูปที่ 4.1 แรมขนาด 1 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้นไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3โครงสร้างของ สแตติกแรม

เพื่อให้เข้าใจเกี่ยวกับการเชื่อมต่อแรมจึงจำเป็นจะต้องเข้าใจระบบพื้นฐานทั่วไปของแรมก่อน ซึ่งสามารถเขียนเป็นแผนผังได้ดังรูป



รูปที่ 4.2 โครงสร้างพื้นฐานของแรม

จากรูปที่ 4.2 เป็นโครงสร้างพื้นฐานของแรมซึ่งมีส่วนประกอบของสัญญาณที่เกี่ยวข้องกับการควบคุมอยู่หลายเส้น สัญญาณที่สำคัญประกอบด้วย

1.Din 0 เป็นสายสัญญาณที่จะนำข้อมูลเข้าไปเก็บไว้ในแรม เมื่อมีการเขียนข้อมูล

2.Dout เป็นสายสัญญาณที่จะนำข้อมูลออกจากแรมเมื่อข้อมูลได้รับการอ่านจากแรมเรียบร้อยแล้ว

3.Address เป็นสายสัญญาณที่จะทำหน้าที่กำหนด แอดเดรสของแรมเพื่อการเขียนหรือการอ่านข้อมูล

4.R/$\overline{W}$ เป็นสายสัญญาณที่ทำหน้าที่กำหนดการเขียนหรือการอ่านข้อมูลบนแรม

5./CE เป็นสายสัญญาณเพื่อเลือกชิฟในกรณีที่ต้องการต่อหลาย ชิฟ ในระบบเพื่อที่จะได้ทราบว่าชิฟใดได้รับการเลือก

6.VCC ,GND เป็นสายไฟเลี้ยงวงจรและกราวด์

การทำงานของแรมจะต้องใช้สายสัญญาณต่าง ๆ เหล่านี้รวมกันโดยสัญญาณควบคุมนี้จะสร้างมาจากวงจรเกทต่าง ๆ ก็ได้

4.4 ลำดับการเขียนข้อมูลในแรม

การเขียนข้อมูลในแรมจะมีลักษณะดังนี้

1.ขั้นแรกจะทำการกำหนดแอดเดรสให้กับหน่วยความจำตามตำแหน่งที่ต้องการ เขียนข้อมูล

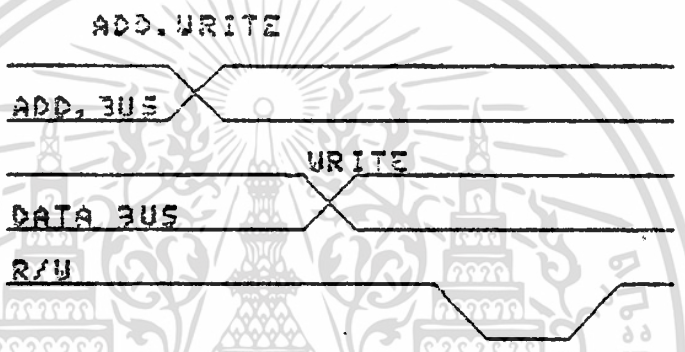
2.กำหนดสายสัญญาณข้อมูลเข้าโดยสายสัญญาณ Din

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ให้ระบบรอเวลาชั่วขณะหนึ่งเรียกช่วงเวลานี้ว่าช่วงเวลาการเขียน (Write Acess time) เพื่อให้วงจรภายในได้รับการถอดรหัสกำหนดตำแหน่งให้เรียบร้อยก่อน และสัญญาณ /ce ต้องมารอก่อนแล้ว

4.หลังจากรอเวลาให้กำหนดสัญญาณ R/$\overline{W}$ เพื่อการเขียนสัญญาณนี้จะเป็นพัลซ์เล็ก ๆ ที่พอเพียงกับการเขียนข้อมูลลงในแรม

แผนผังเวลาสำหรับการเขียนข้อมูลลงในแรมสามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 แผนผังเวลาการเขียนข้อมูลลงในหน่วยความจำ

<u>4.5 ลำดับการอ่านข้อมูลจากแรม</u>

สมมุติว่าต้องการอ่านข้อมูลจากแรมที่มีข้อมูลพร้อมอยู่แล้วมาใช้งานจะต้องมีลำดับการทำงานดังนี้

1.จะต้องกำหนดแอดเดรสที่ต้องการอ่านให้กับแรม วงจรภายในของแรมจะทำการถอดรหัสเพื่อกำหนดตำแหน่งที่แท้จริงสำหรับการอ่าน

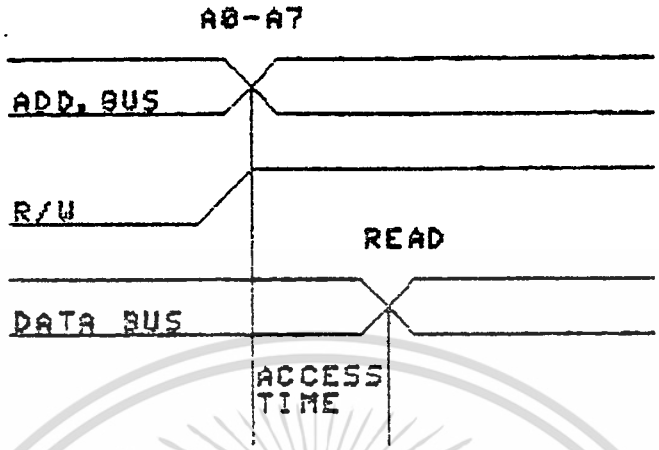
2.กำหนดสัญญาณที่ขา R/$\overline{W}$ ให้ถูกต้องตามลอจิกโดยการอ่านข้อมูลหน่วยความจำบางชิฟต้องกำหนดลอจิก 1 และบางชิฟจะต้องกำหนดลอจิก 0 การกำหนดให้เป็นลอจิก 0 หรือ 1 จะต้องพิจารณาโดยดูจากคู่มือของชิฟตัวนั้น

3.จะต้องให้ระบบรออยู่ชั่วขณะหนึ่งซึ่งเรียกว่าช่วงเวลาการอ่าน (Read Access time) ในการอ่านนี้จะใช้สัญญาณเลือกชิฟทำการเลือกโดยส่งสัญญาณ /ce มาก่อน

4.ในช่วงเวลาขณะที่รอนี้ข้อมูลที่ได้รับการอ่านจะมาปรากฏรออยู่ที่สายสัญญาณ Dout

เมื่อเขียนเป็นแผนผังเวลาเพื่อแทนการทำงานทั้ง 4 ขั้นตอนที่กล่าวถึงนี้จะได้ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

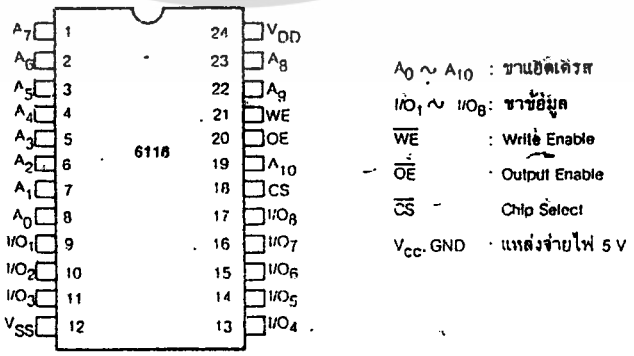


รูปที่ 4.4แผนผังเวลาการอ่านข้อมูลจากหน่วยความจำ

4.6แรมเบอร์ 6116-3

หน่วยความจำที่ใช้ในการเก็บข้อมูลจากการแซมปลิ้ง DATA เข้ามาเก็บไว้ในแรมในที่นี้จะใช้สแตติกแรมเบอร์ 6116-3 ซึ่งมีความเร็วในการทำงาน (ดูจากคู่มือ) Read cycle time ต่ำสุด 150 ns.

แรมเบอร์ 6116 นี้เป็นแรมที่มีขนาดความจุของหน่วยความจำถึง 2KB โดยแรมชิฟนี้จะมีสายแอ็ดเดรสทั้งหมด 11 เส้นและสายสัญญาณ I/O จำนวน 8 เส้นสำหรับการต่อเข้ากับ บัสข้อมูล การควบคุมจะมีสัญญาณ $\overline{W}$ และสัญญาณ /ce และสัญญาณ /oe หรือการอีนาเบิ้ลสัญญาณที่สาย I/O อีกหนึ่งสัญญาณรายละเอียดข้อมูลทางเทคนิคของแรมเบอร์ 6116 จะแสดงได้ดังรูปที่ 4.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แรมเบอร์ 6116 2KB CMOS

แรมเบอร์ 6116 นี้เป็น ไอซีขนาด 24 ขาจะมีขนาดใหญ่กว่า ไอซี 16หรือ 14 ขาทั่วไป ขาแหล่งจ่ายไฟคือ VCC and GND ใช้กับแหล่งจ่ายไฟ 5 โวลท์

ขา A0-A10 รวมทั้งหมด 11 ขาเป็นขาแอดเดรสหมายถึงกำหนดตำแหน่งที่อยู่ของกลุ่มหน่วยความจำ ไอซีเบอร์นี้มีกลุ่มหน่วยความจำทั้งหมด 2048 กลุ่ม กลุ่มหนึ่งมีขนาด 8 บิท แต่ละกลุ่มจะอยู่เรียงลำดับกันอยู่ภายในการอ่านหรือเขียนข้อมูลกับกลุ่มหน่วยความจำใดจำเป็นต้องบอกเบอร์ของกลุ่มก่อน การบอกเบอร์ก็คือ การบอกแอดเดรสจำนวนตำแหน่งที่อยู่ของกลุ่มข้อมูลหน่วยความจำอย่างเช่นเรากำหนดแอดเดรสโดยให้สัญญาณที่ขา A0-A10 เป็น 0000 0000 1001 หมายถึงกำหนดเบอร์กลุ่มเป็นกลุ่มเบอร์ 9

ขา I/O 01-I/O 08 รวมทั้งหมด 8 ขาเป็นขาที่เราใช้ป้อนข้อมูลที่เราจะเขียนและใช้อ่านข้อมูลจากกลุ่มหน่วยความจำภายในการเขียนหรือการอ่านข้อมูลจะทำได้พร้อมกันเป็นกลุ่ม กลุ่มละ 8 บิท จึงต้องมีขาข้อมูล 8 ขา

เมื่อเราต้องการเขียนข้อมูลลงไปในกลุ่มหน่วยความจำเริ่มแรกเราต้องให้แอดเดรสของกลุ่มหน่วยความจำนั้นที่ ขาแอดเดรส A0-A10 จากนั้นเราจะป้อนข้อมูลที่ ขา I/0 1-I/O 8 พร้อมกัน 8 ขาทันที ต่อไปเราต้องป้อน 0 เข้าที่ ขา $\overline{W}$ ข้อมูลที่ ขา I/O1-I/O 8 ที่เราป้อนไว้จะถูกส่งเข้าไปเก็บในหน่วยความจำกลุ่มที่เรากำหนดด้วยแอดเดรส ข้อมูลใหม่นี้จะเขียนทับข้อมูลเดิมที่อาจเก็บไว้ก่อนการเขียนนี้

ถ้าเราต้องการอ่านข้อมูลจากหน่วยความจำเราก็ป้อนแอดเดรสของกลุ่มหน่วยความจำเข้าที่ขา A0-A10 ให้ขา $\overline{W}$ เป็นลอจิก 1 และให้ขา /oe เป็นลอจิก 0 ข้อมูลภายในกลุ่มหน่วยความจำที่ตรงกับแอดเดรสนั้นจะถูกอ่านแล้วส่งออกมาที่ขา I/O1-I/O 8 เช่นกัน

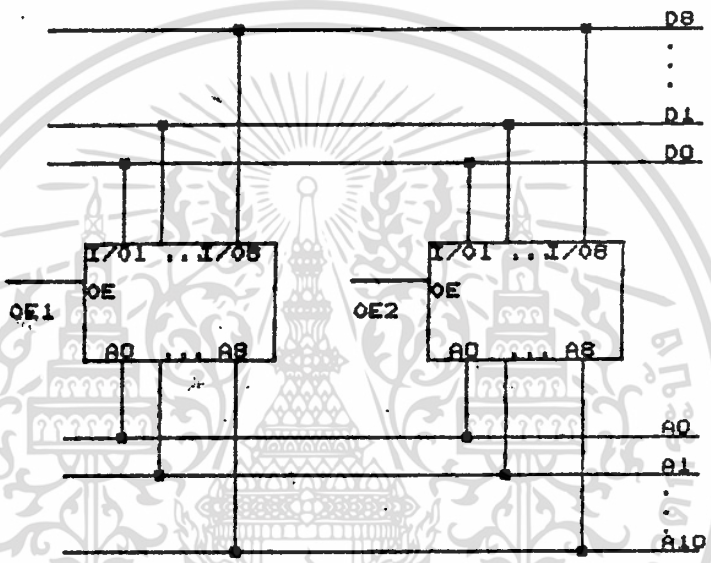
เราจะเห็นว่าขา I/O 1-I/O8 นี้เป็นได้ทั้งอินพุทและเอาท์พุทเมื่อเราเขียนข้อมูลขา I/O1-I/O8 จะเป็นอินพุทให้ข้อมูลของเราเข้าไปได้เมื่อเราอ่านข้อมูลขา I/O1-I/O8 จะเป็นเอาท์พุทส่งข้อมูลออกมา ที่ขาไอซีเป็นได้ทั้งอินพุทและเอาท์พุทนี้ทำให้สามารถประหยัดการเดินสายภายในวงจรได้

ขาข้อมูลนี้ยังมีคุณสมบัติพิเศษอีกอย่างหนึ่งคือ สามารถอยู่ในสภาวะลอยตัวได้ เป็นสภาวะที่มีอิมพีแดนซ์สูงเปรียบเหมือนไม่ได้ต่อกับวงจรภายนอก สภาวะนี้เราเรียกว่าสภาวะที่ 3 หรือ ไตรสเตท ถ้าเราป้อน 1 ให้ขา /oe จะทำให้ขา I/O1-I/O8 อยู่ในสภาวะไตร์สเตทนี้

สภาวะไตรสเตทนี้มีสภาวะที่เราสามารถนำมาใช้งานได้คือ เราสามารถต่อขาข้อมูลของไอซีแรมหลาย ๆ ตัวมารวมกันได้การต่อขารวมกันในวงจรดิจิตอลจะทำไม่ได้เลยถ้าขาของไอซีเหล่านั้นเป็นขาเอาท์พุททั้งคู่ เพราะแต่ละเส้นจะส่งผ่านสัญญาณเอาท์พุทออกมาชนกันถ้าฝ่ายหนึ่งส่งลอจิก 1 และฝ่ายหนึ่งส่งลอจิก 0 จะมีกระแสไหลจากฝ่ายหนึ่งไปยังอีกฝ่ายหนึ่งอย่างมาก จะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีเสียหายทันที แต่ถ้าเอาท์พุทสามารถอยู่ในสภาวะไตรสเตท ซึ่งเปรียบเสมือนการไม่ต่อกับวงจรเราสามารถต่อเอาท์พุทเหล่านั้นรวมกันได้โดยเราจะต้องควบคุมให้ขาของไอซีเพียงตัวเดียวเท่านั้นที่เป็นเอาท์พุทในขณะที่ขาของไอซีตัวอื่น เป็นไตรสเตท สำหรับไอซีแรมตัวนี้สามารถนำเอาข้อมูลมาต่อรวมกันหลาย ๆ ตัวได้ เมื่อเราต้องการอ่านข้อมูลจากไอซีตัวใดเราก็จะส่งสัญญาณ 0 ไปที่ ขา /oe ของไอซีตัวนั้นในขณะที่ขา /oe ของไอซีตัวอื่น เป็นหนึ่งตลอดเวลา



รูปที่ 4.6 การต่อขาข้อมูลของไอซีแรมหลาย ๆ ตัวร่วมกัน

สำหรับขา /cs (chip select) เป็นขาเลือกให้ไอซีเริ่มทำงานเมื่อป้อนลอจิก 0 ให้ขานี้ไอซีจะเริ่มอ่านแอดเดรสเข้าไปภายใน ปกติเราใช้ขา /cs นี้เป็นขาสำหรับเลือกไอซีแรม ในกรณีที่มีแรมในระบบหลายตัวบางครั้งเราก็ใช้ขานี้เป็นขาแอดเดรสขาหนึ่งของไอซีแรมขา /cs นี้แรมบางเบอร์อาจตั้งชื่อขานี้ว่า /ce (chip enable)

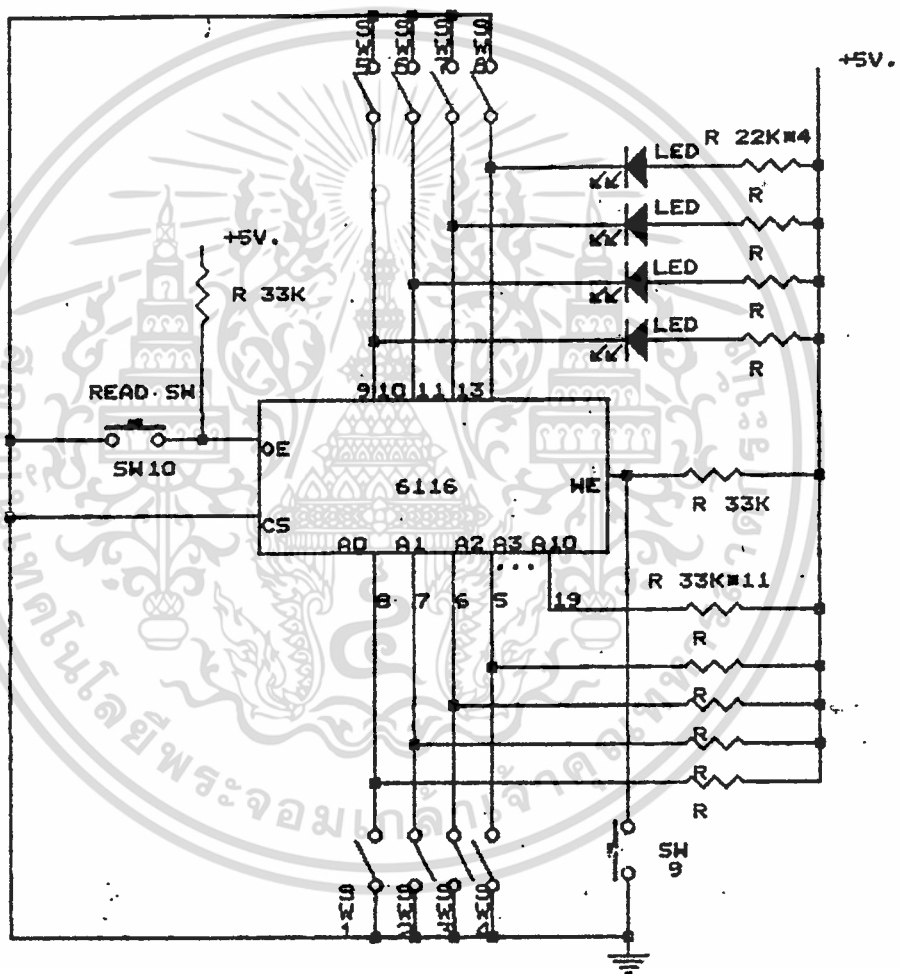
4.7การทดลองสแตติกแรมเบอร์ 6116-3 เบื้องต้น

เราจะทดลองเบอร์ 6116-3 มาทำการทดลองอ่านข้อมูลและเขียนข้อมูลซึ่งสามารถต่อวงจรทดลองได้ดังรูปที่ 4.7

เราใช้ดิพสวิทช์ขนาด 4 P แทน SW1-SW4 และ SW5-SW8 ดิพสวิทช์นี้จะใช้ในการป้อนแอดเดรสและข้อมูลเข้าไปใน ไอซีแรมเนื่องจากขาแอดเดรสมีถึง 11 ขา แต่เราจะทดลองป้อน แอดเดรสเพียง 4 ขา เท่านั้น สำหรับข้อมูลก็เช่นกัน เราจะเขียนข้อมูลและอ่านข้อมูลจาก I/O1-I/O4 เพียง 4 ขาเท่านั้น ซึ่งจะง่ายต่อการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิทช์ SW9 และ SW10 ใช้เป็นสวิทช์ปุ่มกดต่อที่ขา $\overline{W}$ และ /oe ปกติถ้าไม่กดปุ่มสวิช์จะเปิดขาทั้งสองนี้จะได้รับสัญญาณลอจิก 1 อยู่ตลอดเวลา



รูปที่ 4.7 วงจรทดลองการเขียนและการอ่านข้อมูลจากไอซีแรม 6116 เบื้องต้น

เมื่อต่อวงจรเสร็จแล้วและตรวจสอบว่าเดินสายวงจรไม่ผิดให้เปิด SW1ถึง SW8 ทั้งหมดก่อนเปิดแหล่งจ่ายไฟ

เราจะเริ่มทำการทดลองด้วยการป้อนข้อมูลเข้าไปในแรมก่อน เราป้อนแอดเดรสได้ 4 ขา ดังนั้น เราจึงสามารถป้อนข้อมูลเข้าไปในกลุ่มหน่วยความจำได้ 16 กลุ่ม คือกลุ่มที่มีแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

111111 0000-11111111111 ขา A4-A10 จะเป็นลอจิก 1 อยู่เสมอ เราลองตั้งดิพสวิทช์ SW1-SW4 เมื่อป้อนแอดเดรส สมมุติตั้งไว้ที่ 0000 คือให้ปิดสวิทช์ทั้งหมด

จากนั้นเราจะตั้งข้อมูลที่ดิพสวิทช์ SW5-SW8 สมมุติว่าเราตั้งข้อมูลไว้เป็น 0101 0 ในที่นี้หมายถึงปิดสวิทช์ 1 ในทีนี้หมายถึงเปิดสวิทช์จากนั้นกด SW9 1 ครั้ง ข้อมูล 0101 จะถูกเขียน เข้าไปเก็บในกลุ่มหน่วยความจำที่มีแอดเดรส 1111110000 ทันที

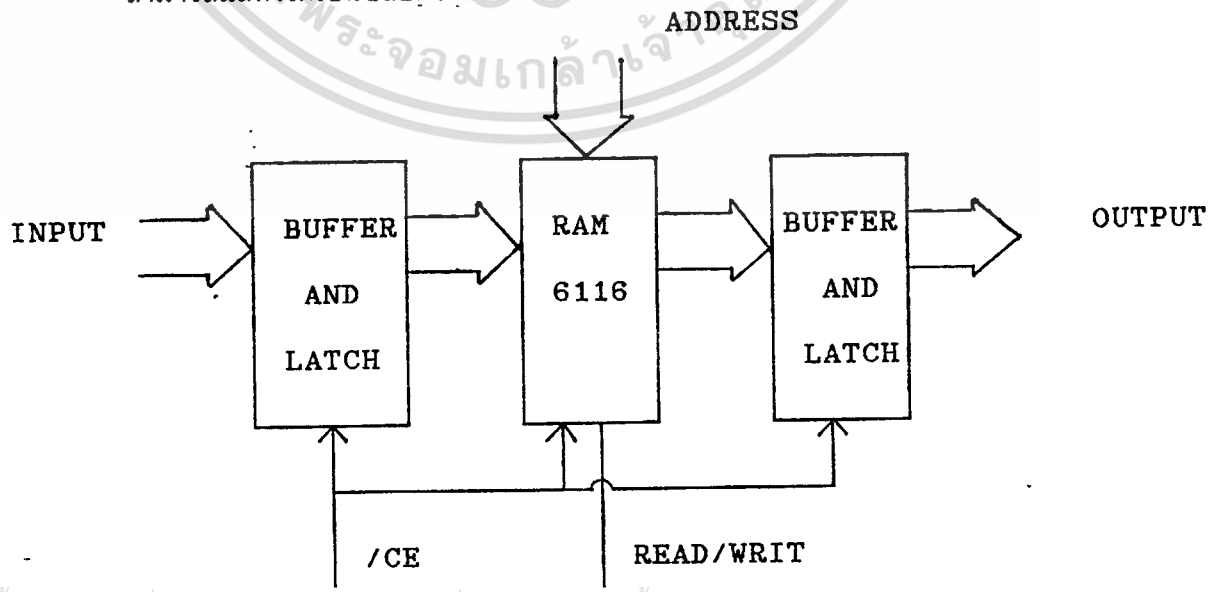
เราทำการกำหนดแอดเดรสอื่น ๆ และป้อนข้อมูลเข้าไปด้วยวิธีการนี้หลาย แอดเดรสจากนั้นก็มาถึงขั้นการอ่านข้อมูลก่อนการอ่านให้เปิด SW5-SW8 ทั้งหมด ตัวนี้สำคัญมาก ถ้ายังไม่ปิดสวิทช์ค้างอยู่จะทำให้ ไอซีเสียหายได้

ตั้งสวิทช์ SW1-SW8 ไปถึงแอดเดรสที่เราต้องการอยากอ่านจากนั้นกด SW10 จะเห็นแอล อี ดี ที่ขา I/O1-I/O4 ติดสว่างตามข้อมูลที่เราป้อนเข้าไป คราวนี้ขา I/O1-I/O4 จะเป็นขาเอาท์พุทการติดของ แอล อี ดี จะมีความหมาดังนี้ ถ้าติดสว่างแสดงว่า บิทนั้นเป็นลอจิก 0 ถ้าดับแสดงว่าบิทนั้นเป็นลอจิก 1 ลองเปลี่ยนแอดเดรสอื่นแล้วลองอ่านข้อมูลเปรียบเทียบกับข้อมูลที่เราเคยป้อนเข้าไป ดูว่าตรงกันหรือไม่

ไอซีแรมนี้มักใช้ในการเก็บข้อมูลชั่วคราว ในระบบไมโครคอมพิวเตอร์ การเก็บข้อมูลชั่วคราวหมายถึง เก็บไว้แล้วจะป้อนข้อมูลใหม่ก็ได้ ข้อมูลในแรมจะไม่คงอยู่ตลอดไปถ้าเราดับแหล่งจ่ายไฟ ข้อมูลจะหายไปหมดทันที เราลองดับแหล่งจ่ายไฟของวงจร รอนานประมาณ 1 นาที จึงเปิดใหม่ คราวนี้ลองอ่านข้อมูลใน ไอซี ดูจะเห็นว่าข้อมูลที่เก็บไว้หายไปแล้วอาจจะเปลี่ยนเป็นข้อมูลใหม่ซึ่งจะเป็นอะไรก็คาดเดาไม่ได้

4.8 การออกแบบวงจรที่ใช้ในการเก็บข้อมูลที่ได้จากการแซมปลิ้ง

สามารถแสดงได้เป็นบล็อกไดอะแกรมดังต่อไปนี้



รูปที่ 4.8 บล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

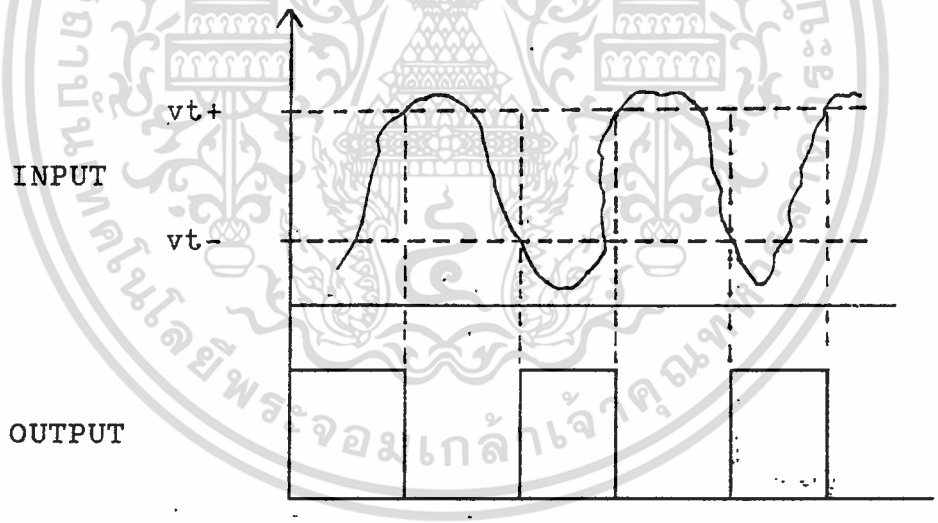
จากบล็อกไดอะแกรมประกอบด้วยส่วนรับส่งข้อมูลอินพุท ซึ่งจะเป็นบัฟเฟอร์และแลทช์แรมเบอร์ 6116 และเอาท์พุทที่จะต่อไปยังเครื่อง พีซี จะประกอบไปด้วยบัฟเฟอร์และ แลทช์

4.9 อินพุทของวงจร

อินพุทของวงจรซึ่งเป็นบัฟเฟอร์แบบชมิททริกเกอร์ ซึ่งทำหน้าที่เป็นตัวกันชนและใช้ต่อวงจรเพื่อชับโหลดให้ได้มากขึ้นโดยไม่ให้ฟังค์ชั่นทำงานผิดพลาดไป

เกทแบบชมิททริกเกอร์สำหรับวงจรดิจิตอลเป็นไอซีลอจิกเกทที่มีคุณสมบัติของอินพุทเป็นแบบใช้สำหรับการเปลี่ยนแปลงของรูปคลื่นที่ไม่เป็นพัลซ์ หรือสัญญาณพัลซ์ที่มีสิ่งรบกวน ให้เป็นสัญญาณพัลซ์ที่นำมาใช้งานได้ดีรูปคลื่นที่ไม่เป็นพัลซ์ได้แก่รูปคลื่นไซด์และรูปคลื่นสามเหลี่ยม และสัญญาณฟันเลื่อย

การเขียนแผนภาพลอจิกของเกทที่มีวงจรชมิททริกเกอร์อยู่ด้วย คงมีลักษณะเหมือนเกททั่วไป เพียงแต่การเขียนเครื่องหมายพิเศษเพิ่มเข้าไปในเกทเท่านั้น ไอซีที่มีคุณสมบัติเป็นแบบชมิททริกเกอร์มีอยู่หลายเบอร์ แต่ในที่นี้จะใช้เบอร์ 74LS244 ซึ่งเป็นเกทแบบ 3 สถานะด้วย

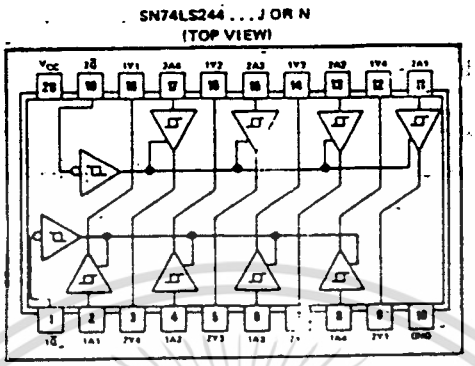


รูปที่ 4.9 การทำงานอินพุทและเอาท์พุทของอินเวอร์เตอร์แบบชมิททริกเกอร์

คุณสมบัติที่สำคัญของชมิททริกเกอร์คือ Vt+(position going theshold voltage) และ Vt-(negative going theshold voltage) Vt+ เป็นค่าแรงดันที่ทำให้เกททำงานเมื่อแรงดันอินพุทเปลี่ยนจากค่าต่ำจนถึงค่านี้ขณะนี้อินพุทถือว่าได้รับลอจิก 1 เอาท์พุทจะมีการเปลี่ยนแปลงตามคุณสมบัติของเกทนั้น ๆ หลังจากนั้นแรงดันอินพุทก็จะเพิ่มขึ้นเรื่อย ๆ จนถึงค่าสูงสุดและค่อย ๆ ลดลงขณะนี้เอาท์พุทเกทจะไม่มีการเปลี่ยนแปลงยังคงถือว่าได้รับลอจิก 1 จนกว่าแรงดันอินพุทจะตกลงมาถึง Vt- เกทจะถือว่าได้รับลอจิก 0 เอาท์พุทจะเกิดการเปลี่ยนแปลงตามคุณสมบัติของเกท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74LS244

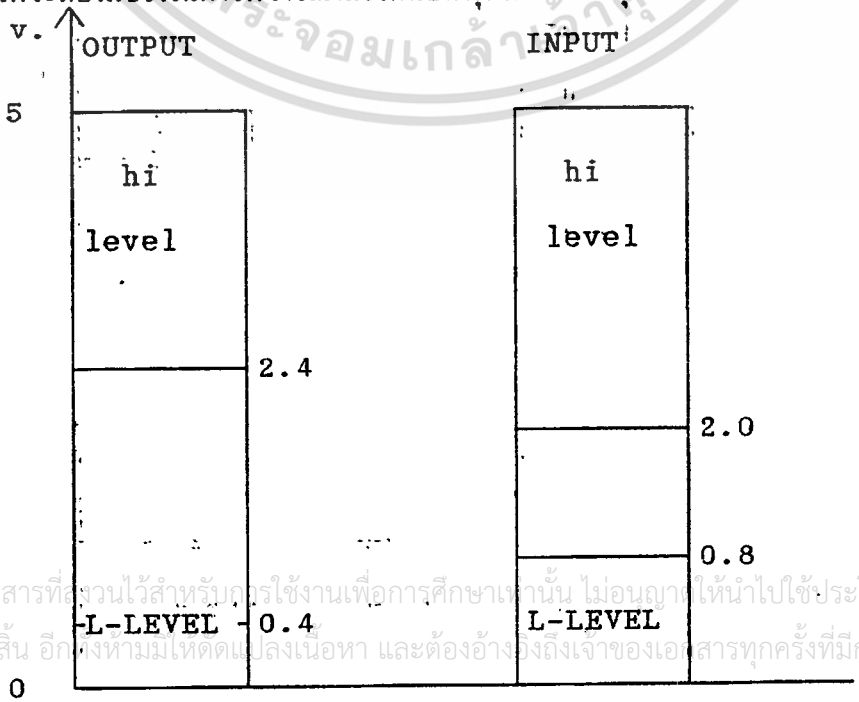


รูปที่ 4.10 74LS244

74LS244 จะเป็นชมิททริกเกอร์ 3 สถานะ ซึ่งโครงสร้างของไอซีแสดงดังรูปที่ 4.10 ซึ่งมีอินพุท 8 อินพุทและเอาท์พุท 8 เอาท์พุท และขาควบคุมการทำงานคือ ขา 1 /1G และ /2G ถ้าขาทั้งสองนี้ได้รับลอจิก 1 จะทำให้เกิด อิมพีแดนซ์สูงคือสถานะที่เอาท์พุทไม่ต่อกับวงจรภายในถ้าให้ขา /1G และ /2G ได้รับลอจิก 0 จะทำให้วงจรทำงานปกติ ถ้าอินพุทเข้ามาเป็นลอจิก 0 เอาท์พุทที่ได้ก็จะเป็นลอจิก 0 และถ้าอินพุทเข้ามาเป็นลอจิก 1 เอาท์พุทที่ได้ก็จะเป็นลอจิก 1

4.10 แรงดันอินพุทและเอาท์พุทของเกท

ลอจิก 0 หรือ ลอจิกา1 ที่ใช้ทำการอธิบายการทำงานของวงจรลอจิกนั้นมีขอบเขตและเงื่อนไขในการทำงานที่แตกต่างกัน อันเนื่องมาจากคุณสมบัติทางไฟฟ้า ซึ่งเกิดจากมาตรฐานที่กำหนดและขบวนการผลิตอุปกรณ์ ดังนั้นการที่จะทำให้วงจรทำงานได้ถูกต้อง ตามหลักการของวงจรลอจิกจะต้องมีช่วงในการพิจารณาแรงดันอินพุทและเอาท์พุทดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

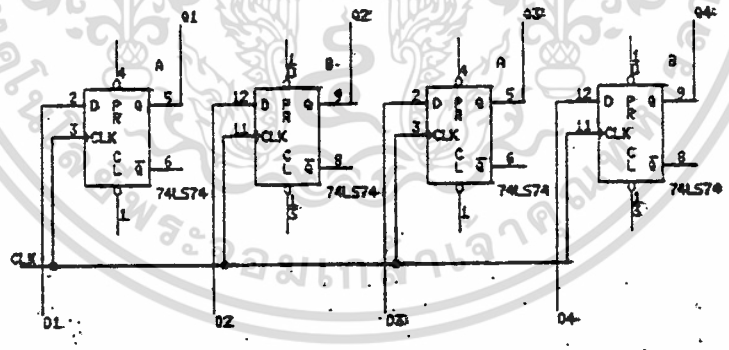
รูปที่ 4.11 ระดับสัญญาณทางอินพุทและเอาท์พุทของไอซีตระกูลทีทีแอล

อินพุทมีแรงดันตั้งแต่ 0-0.8 โวลท์ พิจารณาว่าเป็นระดับแรงดันต่ำหรือลอจิก 0
อินพุทมีแรงดันตั้งแต่ 2-5 โวลท์ พิจารณาว่าเป็นระดับแรงดันสูงหรือลอจิก 1
เอาท์พุทมีแรงดันตั้งแต่ 0-0.4 โวลท์ พิจารณาว่าเป็นระดับแรงดันต่ำหรือลอจิก 0
เอาท์พุทมีแรงดันตั้งแต่ 2.4-5 โวลท์ พิจารณาว่าเป็นระดับแรงดันสูงหรือลิอจิก 1

อินพุทที่มีค่าระหว่าง 0.8-2 โวลท์ ไม่ควรนำมาเป็นอินพุทของวงจรเกท เพราะจะทำให้ได้สัญญาณเอาท์พุทที่ไม่แน่นอน ซึ่งในการทดลองอาจใช้ได้ แต่ระดับแรงดันที่จะแบ่งว่าอินพุทเป็นลอจิก 0 หรือ 1 นั้นมีค่าไม่เท่ากัน อีกทั้งการกำหนดช่วงของลอจิก 0 หรือ 1 แตกต่างกันนั้นจะช่วยป้องกันสัญญาณรบกวนที่แฝงมากับสัญญาณอินพุทได้ด้วย

4.11วงจรแลทช์

วงจรแลทช์เป็นวงจรที่ให้ข้อมูลผ่านเข้ามาเมื่อมีการให้สัญญาณแลทช์จะทำให้ข้อมูลไม่สามารถผ่านเข้ามาได้อีก โดยที่วงจรจะทำการเก็บข้อมูลชุดสุดท้ายที่ผ่านมาไว้ จนกว่าจะเปิดวงจรและได้รับสัญญาณใหม่อีก



รูปที่ 4.12 โครงสร้างของวงจรแลทช์ข้อมูล

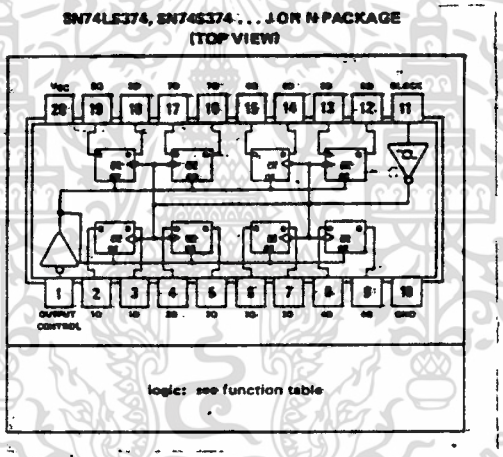
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 ใช้ ดีฟลิบฟลอบ 4 ตัวมาต่อเป็นวงจรแลทช์ 4 บิท เมื่อมีสัญญาณพัลช์ป้อนเข้ามาที่ขา CLK ของ ดีฟลิบฟลอบ จะทำให้ข้อมูลที่ขา ดี ถูกส่งผ่านไปที่เอาท์พุท Q เช่น สมมุติว่าเราจะทำการป้อนข้อมูล 1010 เข้าไปที่ขา D1-D4 เมื่อมีสัญญาณพัลช์เข้ามาที่ขา CLK จะทำให้สัญญาณที่ได้ทางเอาท์พุทเป็น 1010 เหมือนอินพุทดังนั้นจะเห็นได้ว่าเมื่อไม่มี CLK ให้กับ ดีฟลิบฟลอบจะทำให้ข้อมูลเป็น 1010 ไม่ว่าอินพุทที่เข้ามาจะมีค่าเป็นเท่าไรก็ตาม

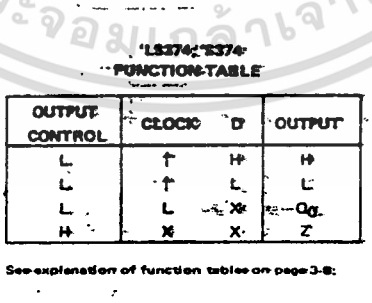
การออกแบบวงจรการใช้งานจะใช้ไอซีเบอร์ 74LS374 ซึ่งเป็นไอซีแลทช์ข้อมูล

74LS374

เป็นไอซีแลทช์ข้อมูลซึ่งทำหน้าที่ในการเก็บข้อมูลชั่วคราวอินพุทรับข้อมูลแบบขนานและเอาท์พุทส่งข้อมูลแบบขนาน โครงสร้างของไอซีแสดงดังรูปที่ 4.13



รูปที่ 4.13 ไอซีเบอร์ 74LS374



'LS374, 'S374
FUNCTION TABLE

OUTPUT CONTROL	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

See explanation of function tables on page 3-8.

รูปที่ 4.14 function table ของ 74LS374

จากตารางการทำงานของ 74LS374 ถ้าเอาท์พุทคอนโทรลเป็น L ถ้า clock ขอบขาขึ้นเข้ามาเอาท์พุทที่ได้จะเปลี่ยนแปลงตาม ดี และถ้าไม่มี clock ขอบขาขึ้นเข้ามาเอาท์พุทที่ได้จะคงเดิม Qo ถ้าเอาท์พุทคอนโทรลเป็น H หรือลอจิก 1 ขา clock หรือขา ดี เป็นอะไรก็ได้เอาท์พุทจะเป็นอิมพีแดนซ์สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12 เอาท์พุทของวงจร

เอาท์พุทของวงจรจะมีการทำงานคล้าย ๆ กับอินพุทเอาท์พุทของวงจรนี้จะนำไปต่อกับ 8255 เพื่อที่จะรับข้อมูลที่ได้ไปประมวลผลบนเครื่องคอมพิวเตอร์ซึ่งเอาท์พุทจะเป็น บัฟเฟอร์ และแลทช์เหมือนกับอินพุท

4.13 วงจรที่ใช้ในการเก็บข้อมูลที่ออกแบบเสร็จแล้ว

วงจรที่ใช้ในการเก็บข้อมูลแสดงดังรูปที่ 4.15 ซึ่งมีสัญญาณควบคุมการทำงาน อยู่ 2 เส้น คือ $R/\overline{W}$ และ /ce

/ce จะเป็นสัญญาณเลือกว่าจะให้แรมทำงานหรือไม่ ถ้าต้องการให้แรมทำงานจะต้องให้ /ce มีลอจิกเป็น 0 และถ้าไม่ต้องการให้ทำงานจะต้อง ให้ /ce มีลอจิกเป็น 1

$R/\overline{W}$ เป็นสายสัญญาณที่ทำหน้าที่กำหนดการเขียนหรือการอ่านข้อมูลบนแรมถ้าต้องการเก็บข้อมูลจากการสุ่มตัวอย่างจะต้องทำให้ขา $R/\overline{W}$ มีลอจิกเป็น 0 และถ้าต้องการอ่านข้อมูลที่ได้จากแรมไปทำการประมวลผลบนเครื่องคอมพิวเตอร์จะต้องทำให้ขา $R/\overline{W}$ มีลอจิกเป็น 1

ในการเก็บข้อมูลนี้จะทำการเก็บเพียง 256 ตำแหน่งเท่านั้นซึ่งจะทำให้ค่าที่ได้มีความละเอียดพอสมควร

แรมเบอร์ 6116-3 ซึ่งมี read cycle time ต่ำสุด 150 ns. จะใช้กับความถี่สูงสุดประมาณ 6 MHz. ซึ่งจะสามารถใช้วัดความถี่ที่เข้าอินพุทได้สูงสุด 3 MHZ ถ้าต้องการให้สุ่มสัญญาณได้ถึง 20 MHz เราจะต้องใช้แรมเบอร์ 6116 มาต่ออนุกรมกัน 4 ตัว คือจะใช้วงจร ริงเคาร์เตอร์คือจะทำให้แรมทั้ง 4 ตัวผลัดกันทำงานซึ่งแสดงได้ดังรูปที่ 4.16

4.14 การอ่านข้อมูลจากแรม

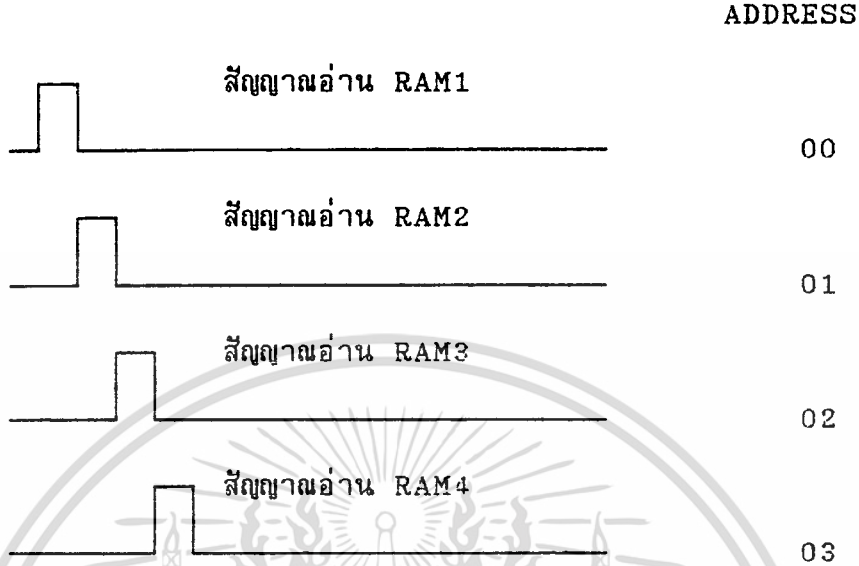
การอินเตอร์เฟสระหว่างส่วนที่เก็บข้อมูลกับเครื่องคอมพิวเตอร์จะใช้การ์ดอินเตอร์เฟสโดยใช้ ไอซีเบอร์ 8255 เป็นตัวรับส่งข้อมูล

ในการอ่านข้อมูลมีขั้นตอนดังนี้

1.จะต้องส่งแอดเดรสมาที่แรม

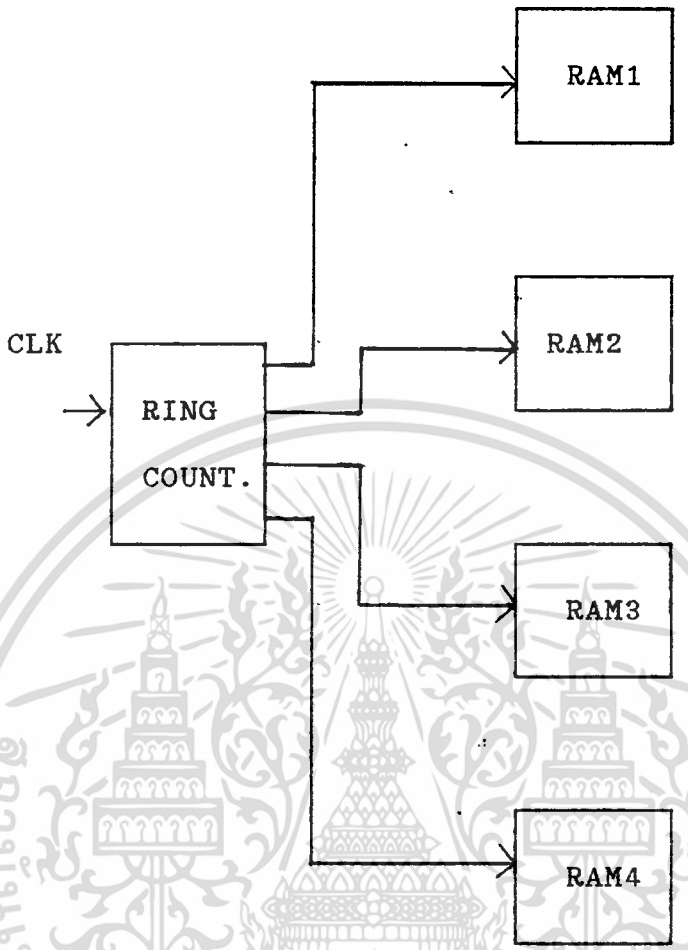
2.ส่งสัญญาณเลือกแรมว่าจะทำการอ่านข้อมูลจากแรมตัวไหนคือ แรมทั้งหมดมี่ 4 ตัว จะต้องทำการเลือกอ่านจากแรมทีละตัวจนครบ 256 ตำแหน่ง ซึ่งสัญญาณการอ่านแรมจะมีสัญญาณดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

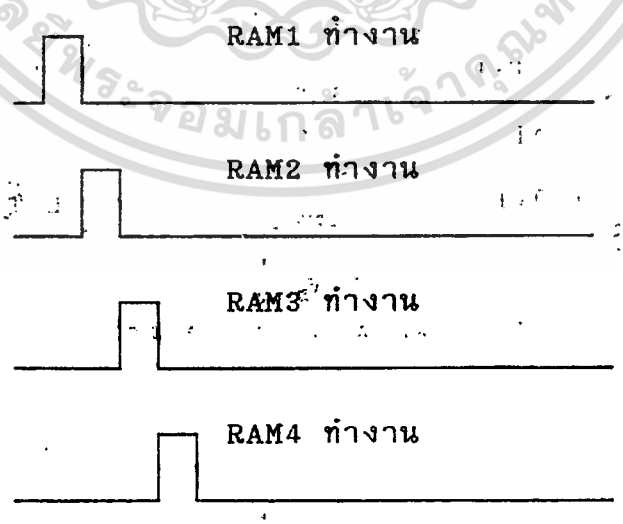


รูปที่ 4.17 สัญญาณการอ่านข้อมูลจากแรมเบอร์ 6116-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

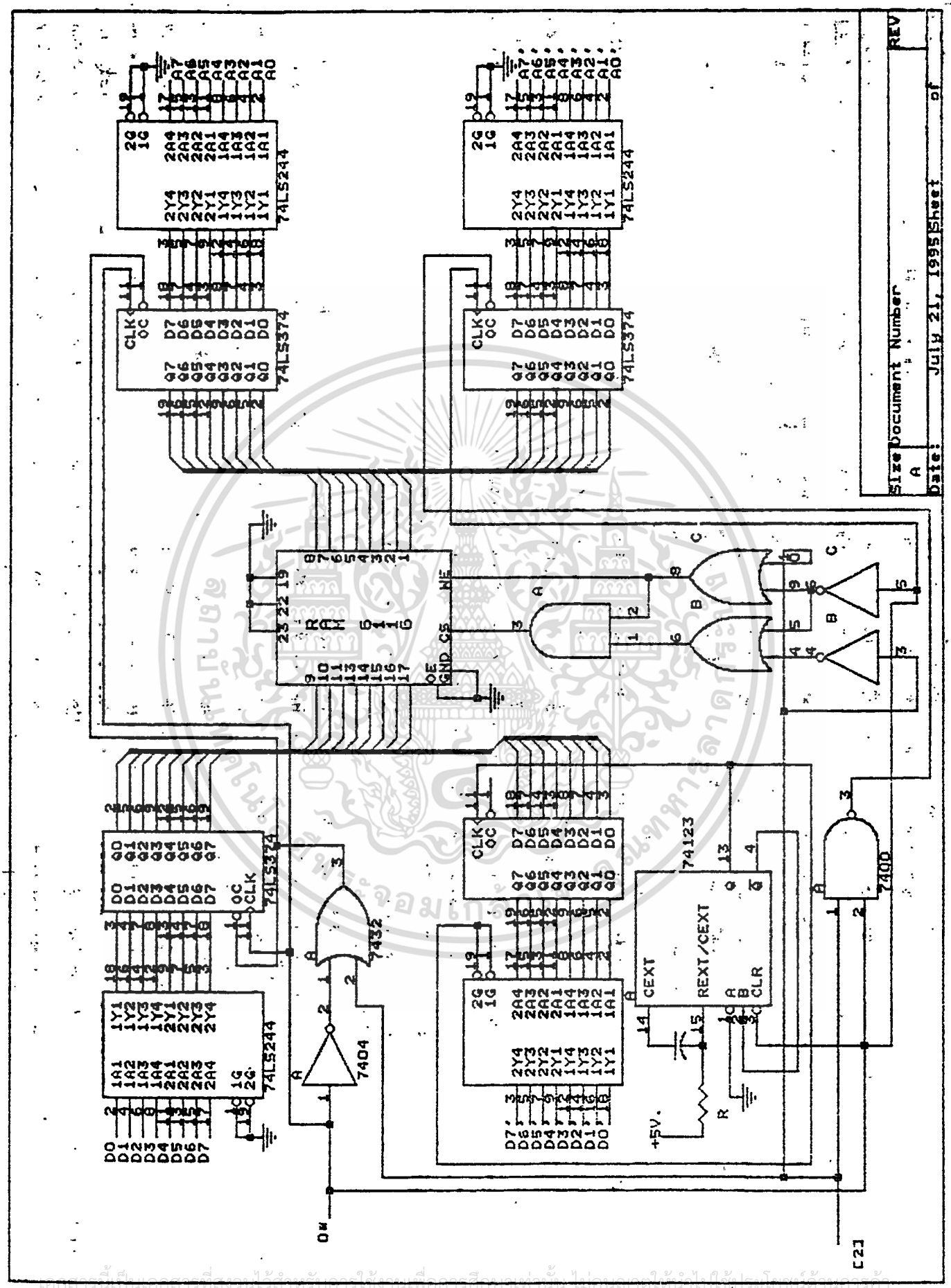


รูปที่ 4.16 ก. บล็อกไดอะแกรมการทำงานของแรมทั้ง 4 ตัว



รูปที่ 4.16ข.สัญญาณการทำงานของแรมทั้ง 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

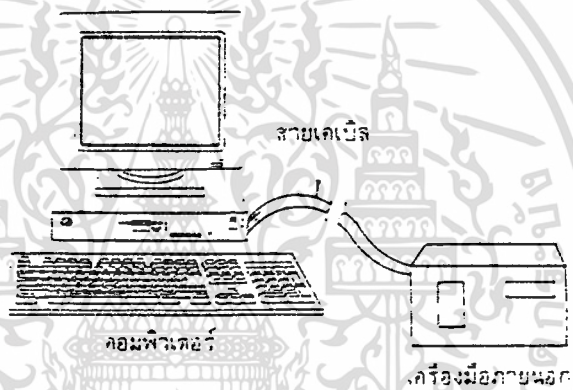


รูปที่ 4.15 วงจรที่ใช้ในการเก็บข้อมูลที่ออกแบบเสร็จ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

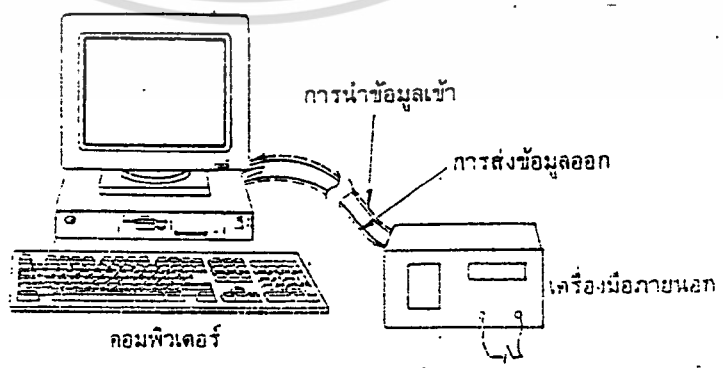
บทที่ 5
ส่วนของการติดต่อกับคอมพิวเตอร์

ไมโครคอมพิวเตอร์ส่วนบุคคล IBM PC มีใช้กันอยู่โดยทั่วไปไม่ว่าจะเป็นในบ้าน สำนักงาน หรือโรงงานอุตสาหกรรม นอกจากใช้ในงานด้านซอฟแวร์แล้วยังสามารถใช้งานเชื่อมต่อกับเครื่องมือภายนอกต่าง ๆ เพื่อทำการวัดปริมาณทางกายภาพ และส่งผลกลับ ในการควบคุมปริมาณทางกายภาพที่แวดล้อมตัวเรา อาจจะเป็นในบ้าน ห้องทดลอง โรงงานอุตสาหกรรม ก็ได้ ปริมาณทางกายภาพเหล่านี้ได้แก่ อุณหภูมิ ความชื้น ระดับ เป็นต้น ในการเชื่อมต่อเครื่องมือภายนอกเข้ากับไมโครคอมพิวเตอร์นั้นปกติจะใช้สายเคเบิลในการสื่อสารโดยแสดงดังรูปที่ 5.1



รูปที่ 5.1 การใช้สายเคเบิลเชื่อมต่อเครื่องมือภายนอกเข้ากับคอมพิวเตอร์

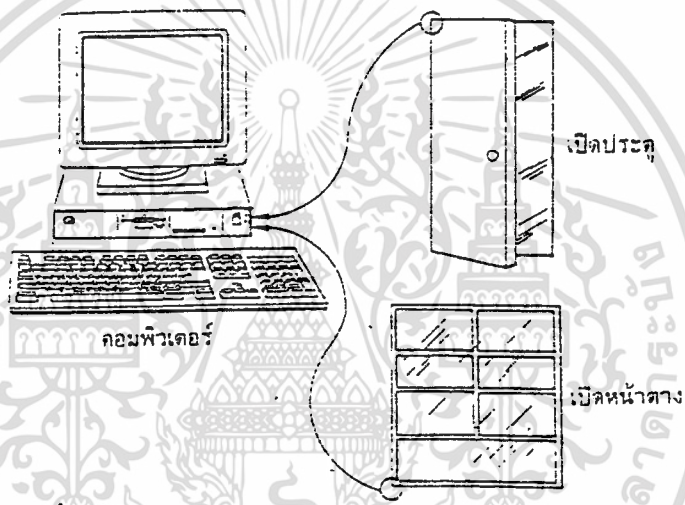
คอมพิวเตอร์มีเส้นทางที่ใช้ในการเชื่อมต่อกับเครื่องมือและอุปกรณ์ภายนอกโดยมีการเรียกข้อมูลเข้ามา และเมื่อต้องการควบคุมก็ทำการส่งข้อมูลออกไป ดังนั้นหลักในการทำการเชื่อมต่อสามารถกล่าวได้คือ



รูปที่ 5.2 การนำข้อมูลเข้าและส่งข้อมูลออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก เช่น เมื่อมีขโมยขึ้นบ้านให้คอมพิวเตอร์ส่งเสียงเตือน เมื่อมีการเปิดประตูหรือหน้าต่างจะให้คอมพิวเตอร์ส่งเสียงเตือนขึ้นที่ลำโพง เมื่อไม่มีการเปิดประตูหรือหน้าต่าง ก็จะไม่มีการส่งข้อมูลไปที่ลำโพง แต่ถ้าเมื่อมีการเปิดประตูหรือหน้าต่าง คอมพิวเตอร์จะทำการส่งข้อมูลเป็นเสียงเตือนไปที่ลำโพงทันทีดังรูปที่ 5.3



รูปที่ 5.3 การนำเข้าข้อมูลในการเปิดประตูหรือหน้าต่างเข้าสู่คอมพิวเตอร์ และการส่งเสียงเตือนแก่ลำโพงจากคอมพิวเตอร์

เราพิจารณาต่อไปจะเห็นว่าคอมพิวเตอร์จะทราบได้อย่างไรว่ามีการเปิดประตูหรือหน้าต่าง ๆ ซึ่งแน่นอนคอมพิวเตอร์ไม่สามารถตรวจจับการเปิดประตูหรือหน้าต่างได้โดยตรง ซึ่งจำเป็นที่จะต้องกระทำผ่านทางตัวแปลงสัญญาณ ซึ่งตัวแปลงสัญญาณนี้ก็คือ Interface card นั่นเอง โดยปกติการเชื่อมต่อระหว่างอุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์นั้นประกอบด้วยอุปกรณ์ส่วนใหญ่ 2 ส่วนคือ

1.ส่วนของ hardware (Interface card) โดยส่วนนี้จะทำหน้าที่ในการแปลงสัญญาณอนาล็อกจากอุปกรณ์ภายนอกมาเป็นสัญญาณดิจิตอลเพื่อส่งให้กับคอมพิวเตอร์ประมวลผลต่อไป และทำหน้าที่ในการแปลงสัญญาณดิจิตอลที่ผ่านการประมวลผลแล้วนั้นเป็นสัญญาณอนาล็อกเพื่อส่ง

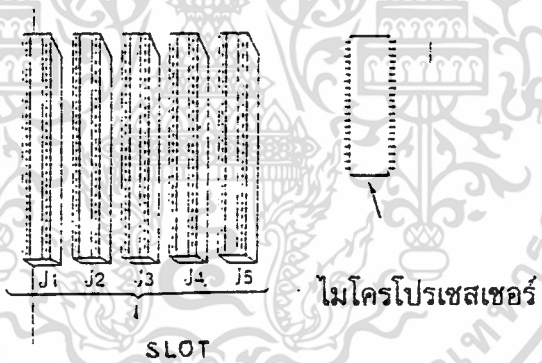
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปควบคุมอุปกรณ์ output ต่อไป นอกจากนี้ยังมีหน้าที่อีกหลายอย่างซึ่งแล้วแต่ผู้ออกแบบว่าจะกำหนดให้ card นี้ทำหน้าที่อะไรบ้าง

2.ส่วนของ software ส่วนนี้ทำหน้าที่ในการควบคุมการทำงานของ Interface card และนำสัญญาณจากภายนอกที่ผ่านการแปลงเรียบร้อยแล้วมาแสดงผลทางจอภาพ โดยในส่วนนี้จะประกอบไปด้วยโปรแกรมต่าง ๆ ที่ทำหน้าที่ในการสั่งการให้คอมพิวเตอร์ทำงานตามที่ผู้เขียนโปรแกรมได้กำหนดไว้ เช่นให้รับค่าอินพุทเข้ามาประมวลผล ,การแสดงผลหน้าจอในโหมดกราฟฟิค ,การกำหนดพารามิเตอร์ต่าง ๆ เกี่ยวกับการควบคุมเป็นต้น จากที่ได้กล่าวมาเป็นรายละเอียดของการ Interface โดยทั่วไป ต่อไปนี้จะได้กล่าวถึง Interface card ของโครงงานนี้ต่อไปคือ

1.ในส่วนของ Interface card

ในการนำข้อมูลเข้า (input data) และการส่งข้อมูลออก (output data) สำหรับ IBM PC นั้นต้องกระทำผ่านทาง Interface card ซึ่งเสียบลงบน I/O slot ของคอมพิวเตอร์ ดังนั้นก่อนอื่นเราควรทราบรายละเอียดเกี่ยวกับ I/O slot ก่อนดังรูป ที่ 5.4 ซึ่งแสดงระบบ I/O slot บน IBM PC



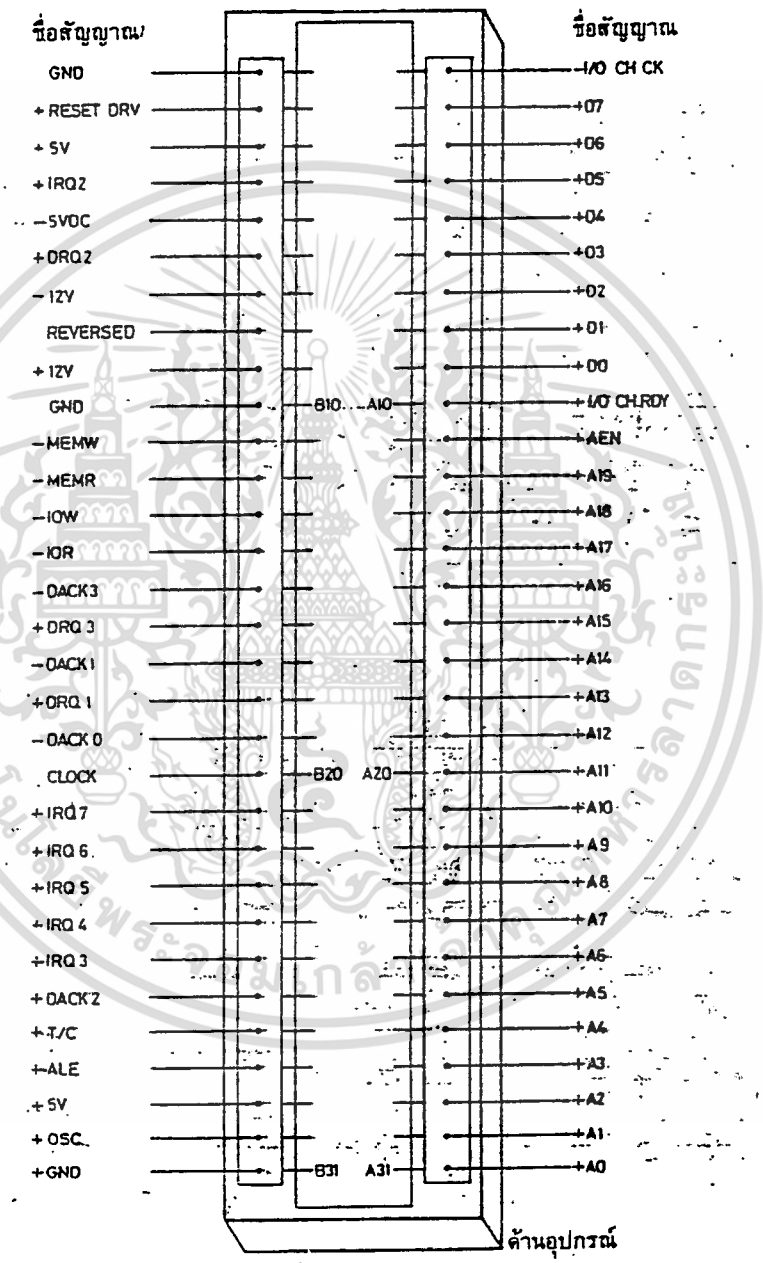
รูปที่ 5.4 slot บนคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 การจัดสรรแอดเดรสที่ใช้ติดต่อกับอุปกรณ์ภายนอกบน IBM PC

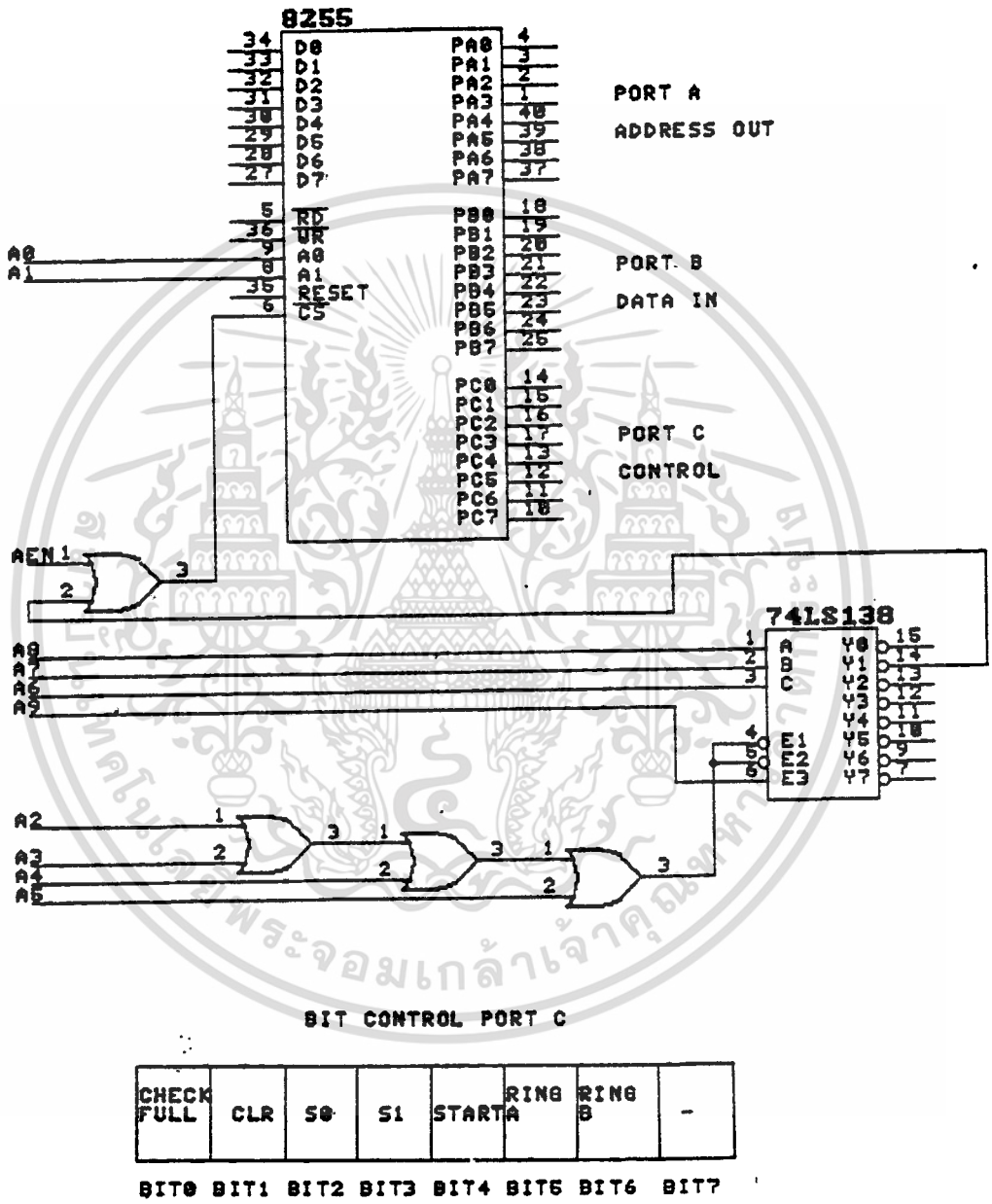
พื้นที่แอดเดรสของอุปกรณ์ I/O											
เลขฐานสิบหก	9	8	7	6	5	4	3	2	1	0	อุปกรณ์ I/O
00-0F	0	0	0	0	0	0	A3	A2	A1	A0	DMA CHIP 8237-2
20-21	0	0	0	0	1	0	Z	Z	Z	A0	INTERRUPT 8259A
40-43	0	0	0	1	0	0	Z	Z	A1	A0	TIMER 8253-5
60-63	0	0	0	1	1	0	Z	Z	A1	A0	PPI 8255A-5
80-83	0	0	1	0	0	0	Z	Z	A1	A0	DMA PAGE REGS
AX	0	0	1	0	1	0					NMI MASK REG
CX	0	0	1	1	0	0					RESERVED
EX	0	0	1	1	1	0					RESERVED
3F8-3FF	1	1	1	1	1	1	1	A2	A1	A0	TP RS-232-C CD
3F0-3F7	1	1	1	1	1	1	0	A2	A1	A0	5¼" DRV ADAPTOR
2F8-2FF	1	0	1	1	1	1	1	Z	A1	A0	RESERVED
378-37F	1	1	0	1	1	1	1	Z	A1	A0	PARALLEL PRTR PRT
3D0-3DF	1	1	1	1	0	1	A3	A2	A1	A0	COLOR/GRAPHICS ADAPTER
278-27F	1	0	0	1	1	1	1	Z	A1	A0	RESERVED
200-20F	1	0	0	0	0	0	A3	A2	A1	A0	GAME I/O ADAPTER
3B0-3BF	1	1	1	0	1	1	A3	A2	A1	A0	IBM MONOCHROME.DISPLAY
											PARALLEL PRINTER ADAPTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 สัญญาณต่างใน I/O slot บน IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 วงจร ของ Interface card ที่ได้ออกแบบไว้ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารภายในคอมพิวเตอร์มีสายข้อมูล (data bus) เพื่อส่งและรับข้อมูลถึงกัน การส่งและรับข้อมูล ถ้าไม่มีตำแหน่งในการส่งและรับก็ไม่รู้ว่าส่งไปที่ไหนและให้รับที่ใด ดังนั้นจึงจำเป็นต้องมีสายแอดเดรส (address bus) ในการกำหนดตำแหน่งภายในคอมพิวเตอร์

ในกรณีที่มีการสื่อสารกับอุปกรณ์ภายนอก ก็ต้องมีการถอดรหัสแอดเดรสให้หมายเลขแก่อุปกรณ์ภายนอกนั้น การถอดรหัสแอดเดรสต้องไม่มีการซ้ำซ้อนกับแอดเดรสที่ใช้สื่อสารกับอุปกรณ์ภายนอกตัวอื่น แอดเดรสที่ใช้ในการสื่อสารกับอุปกรณ์ภายนอกสำหรับ IBM PC แสดงรายระเอียดดังในตารางที่ 5.1

จากตารางจะเห็นว่า แอดเดรสในช่วง 300-30F นั้นไม่มีการใช้งาน ดังนั้นเราจึงสามารถถอดรหัสแอดเดรสใช้งาน ในการเชื่อมต่อในช่วงดังกล่าวได้ เพื่อใช้ในการนำข้อมูลเข้าออก

การถอดรหัสแอดเดรสเพื่อการเลือกพอร์ต (port select) ได้ใช้สายแอดเดรส A0-A11 ซึ่งจะกล่าวรายละเอียดต่อไป แต่การที่จะส่งหรือรับข้อมูลจาก data bus ได้ต้องมีสายสัญญาณ AEN (address enable) ของ IBM PC ต่อร่วมด้วยเพราะไม่เช่นนั้น การส่งหรือรับข้อมูลจากสายข้อมูลจะเกิดการผิดพลาดได้ ซึ่งวงจรโดยสมบูรณ์สของ Inter face card ได้แสดงไว้ดังในรูปที่ 5.6 ก่อนอื่นเราควรทราบรายละเอียดเกี่ยวกับ IC 8255 กันก่อนพอสังเขปคือ 8255Aเป็น IC parrallel port ซึ่งสามารถต่อเป็น port ให้กับไมโครโปรเซสเซอร์ได้ 3 port โดยมีโครงสร้างลักษณะพื้นฐานดังรูปที่ 5.7

<u>คุณสมบัติของ 8255A</u>

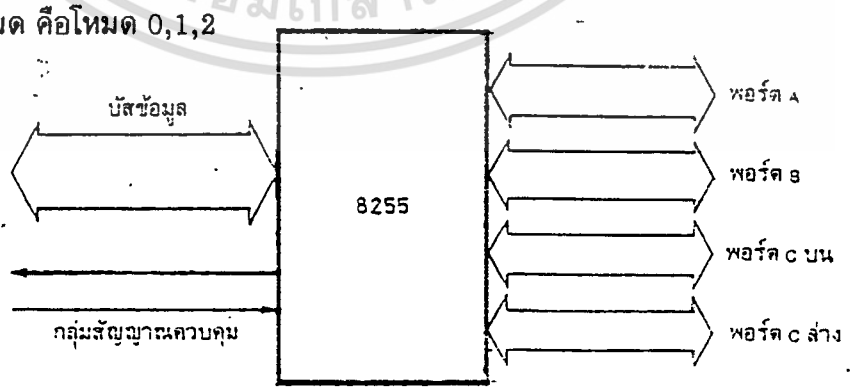
1.มีอินพุตและเอาท์พุท 24 ขาที่สามารถโปรแกรมได้

2.ระดับสัญญาณเดียวกับ TTL

3.มีคุณสมบัติที่เหมือนกับไมโครโปรเซสเซอร์ตระกูลอินเทล

4.สามารถเซทและรีเซทบิทได้โดยตรง

8255A สามารถโปรแกรมให้เป็นอินพุทหรือเอาท์พุทพอร์ตก็ได้ สามารถกำหนดโหมดการทำงานได้ 3 โหมด คือโหมด 0,1,2

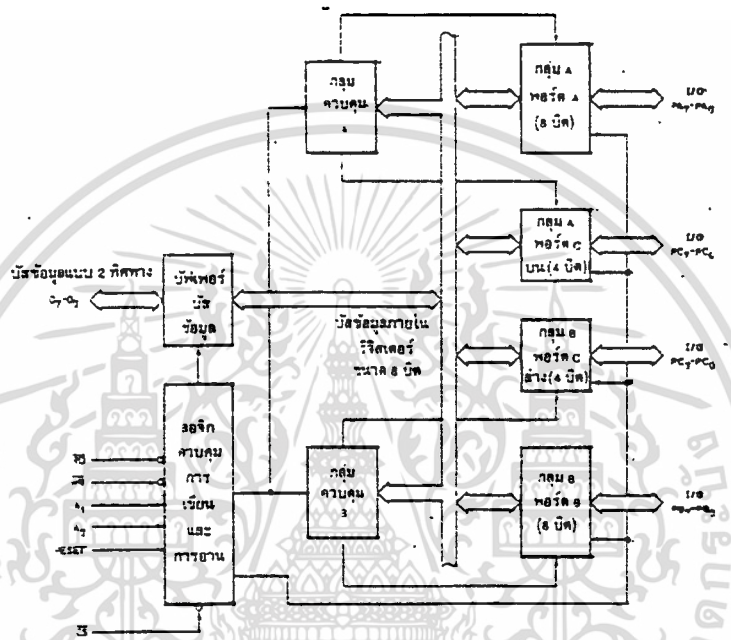


รูปที่ 5.7 โครงสร้างของ IC 8255A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในของ 8255A

8255A มีโครงสร้างภายในดังรูปที่ 5.8 คือ



การเรียกชื่อ port ของ 8255A จะเรียกชื่อพอร์ตว่า พอร์ต A,B และ C โดย port C แยกเป็น 2 ส่วนคือ PC0-PC3 เรียกว่า port C ล่าง และ PC4-PC7 เรียกว่าport C บน

จากรูปที่ 5.8 สามารถอธิบายหน้าที่การทำงานของแต่ละส่วนได้ดังนี้

- Data bus buffer

ทำหน้าที่เป็นบัฟเฟอร์ 8 bit สองทิศทาง 3 สถานะ (tri state) ใช้ในการอินเตอร์เฟสข้อมูลของระบบ (system data bus) ข้อมูลจะถูกส่งหรือรับโดยผ่านทางบัฟเฟอร์ขึ้นอยู่กับการประมวลผลของ CPU ว่าจะให้ทำการส่งหรือรับข้อมูล

- Read/Write and Control logic

ทำหน้าที่จัดการเกี่ยวกับการส่งผ่านข้อมูลและ Control หรือ status ทั้งภายในและภายนอก ส่วนนี้จะรับอินพุตจากบัสแอดเดรสและบัสควบคุมของ CPU แล้วนำมาถอดรหัสในการควบคุมพอร์ต โดยบล็อกนี้มีสายสัญญาณควบคุมจากภายนอกดังต่อไปนี้คือ

-Chip select เมื่อขานี้ได้รับลอจิก "0" จะทำให้ 8255A ต่อเข้ากับระบบบัสของ CPU เพื่อให้ CPU อ่านหรือเขียนข้อมูลผ่านพอร์ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Read เป็นขาสัญญาณอินพุตที่ต้องส่งมาจาก CPU เมื่อสัญญาณนี้มีลอจิก "0"และ Chip select เป็น "0" ด้วย CPU จะทำการอ่านข้อมูลจากบัสข้อมูลของ 8255A

-Write ขาสัญญาณการเขียนจะแอคตีฟเมื่อ Chip select เป็น "0" และ Write เป็น "0" โดยสัญญาณนี้จะถูกส่งมาจาก CPU ทำให้ CPU สามารถเขียนข้อมูลบนบัสข้อมูลของ 8255A ได้

- port select 0 and port select 1 (A0 and A1) สัญญาณอินพุททั้งสองนี้จะต้องสัมพันธ์กับสัญญาณ Read,Write and Chip select เพื่อใช้ในการเลือกพอร์ตใช้งานรายละเอียดดังในตารางที่ 2.2

-Reset แอคตีฟที่ลอจิก "1" สำหรับขานี้ เมื่อมีลอจิก "1" เข้ามาจะทำให้เกิดการเคลียร์รีจิสเตอร์ควบคุม (control registor) และทุกพอร์ตจะถูกเซตสู่โหมดอินพุต

-Group A and Group B controls แต่ละบล็อกจะรับคำสั่งจาก read/write control logic และรับ control word จากบัสข้อมูลภายในเพื่อใช้ในการควบคุมพอร์ต

-Control Group A จะควบคุมพอร์ต A และ port C บน (PC4-PC7)

-Control Group B จะควบคุม port B และ port C ล่าง (PC0-PC3)

port A,B andC ของ 8255A ประกอบด้วย port 8 bit 3 port (port AB and C) ซึ่ง port ทั้ง 3 นี้สามารถกำหนดรูปแบบการใช้งานได้ โดยอาศัย software ช่วยจัดการ

ตารางที่ 5.2 รายละเอียดการเลือกใช้ port

ตารางที่ 2 รายละเอียดการเลือกใช้พอร์ต

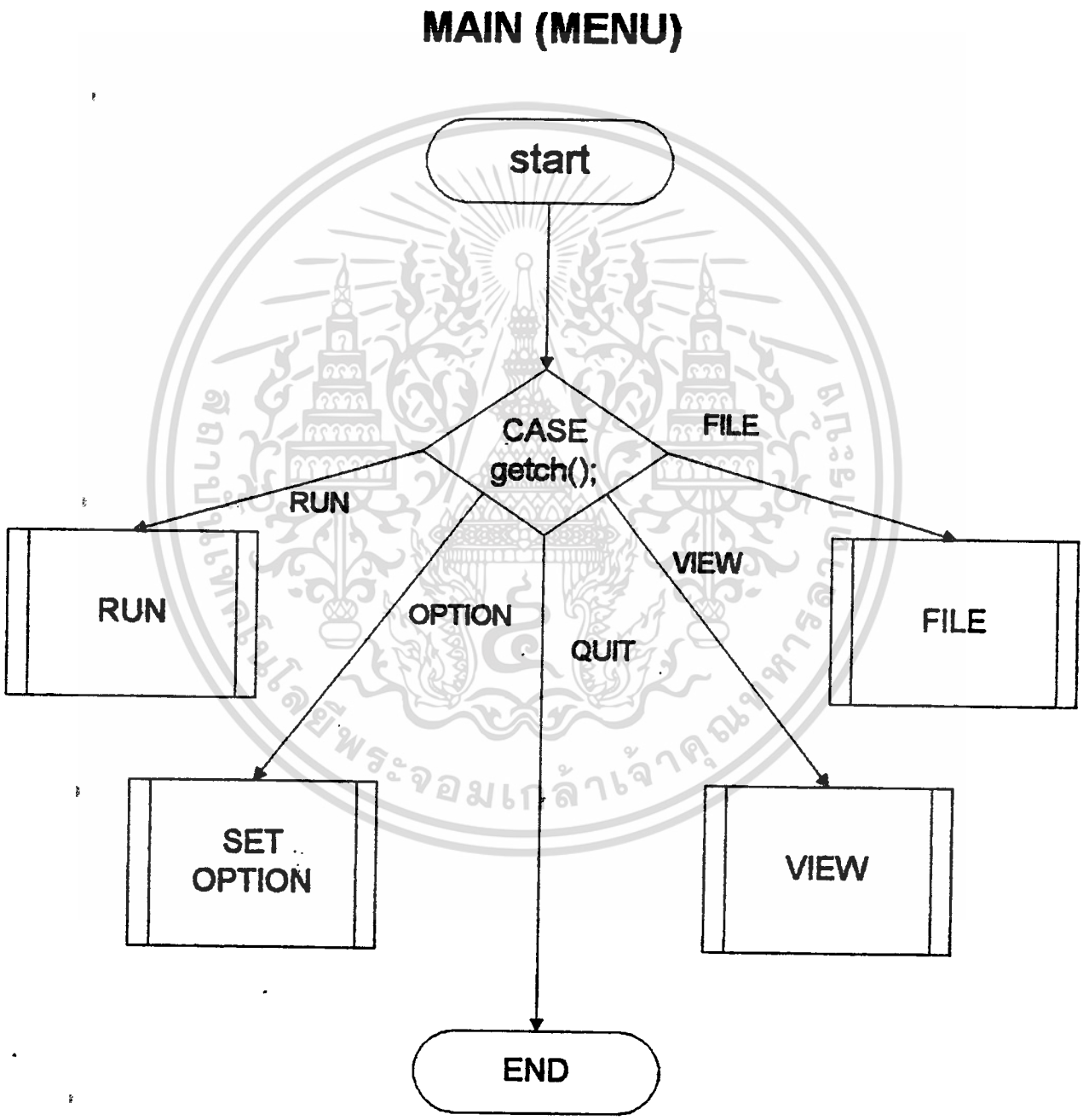
A_1	A_0	RD	WR	CS	กระบวนการทางอินพุต (อ่าน)
0	0	0	1	0	PORT A - DATA BUS
0	1	0	1	0	PORT B - DATA BUS
1	0	0	1	0	PORT C - DATA BUS
					กระบวนการทางเอาต์พุต (เขียน)
0	0	1	0	0	DATA BUS - PORT A
0	1	1	0	0	DATA BUS - PORT B
1	0	1	0	0	DATA BUS - PORT C
1	1	1	0	0	DATA BUS - CONTROL
					ฟังก์ชันยกเลิกการทำงาน
X	X	X	X	1	DATA BUS - 3-STATE
1	1	0	1	0	ELEGA CONDITION
X	X	1	1	0	DATA BUS - 3-STATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

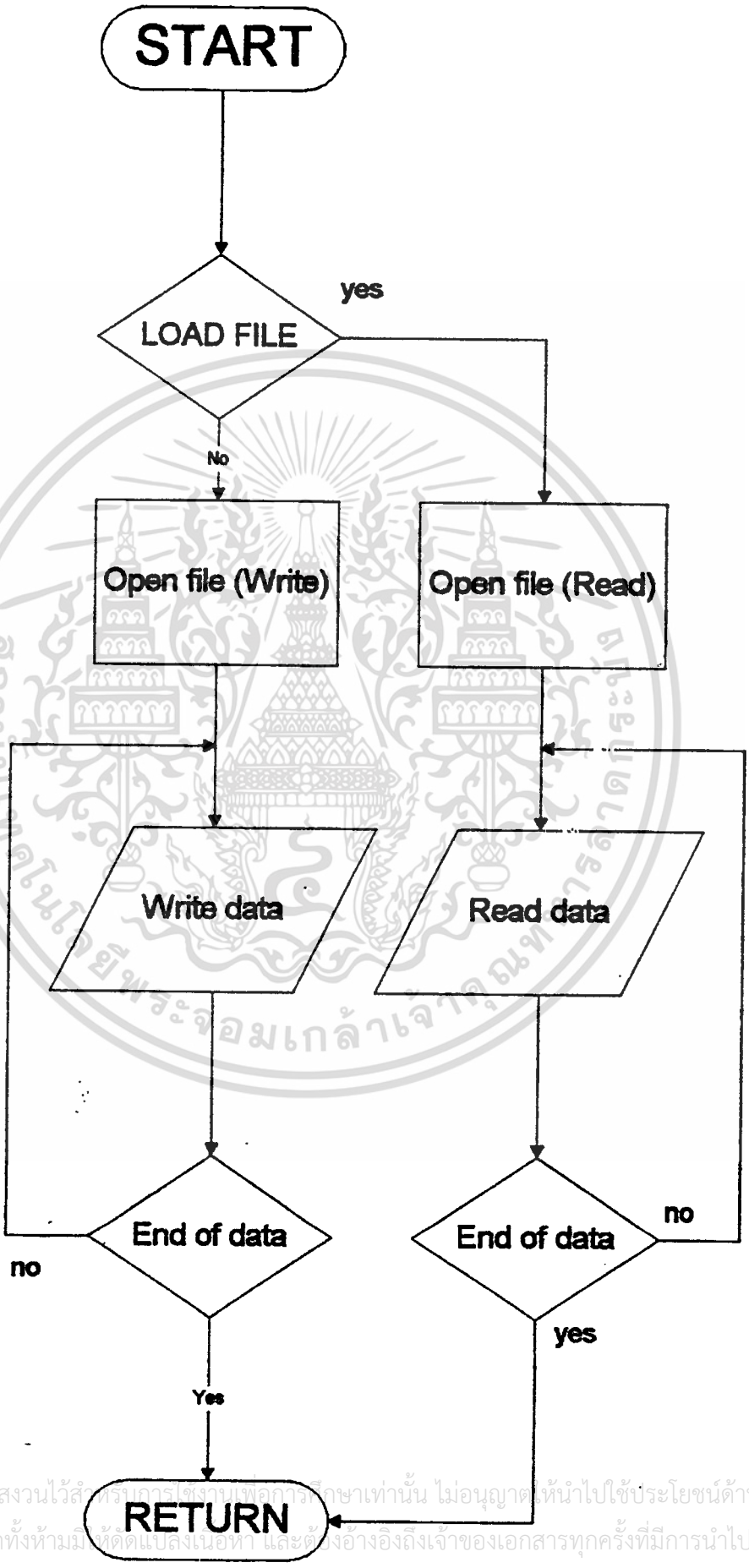
FLOW CHART และผลการทดลอง

FLOW CHART



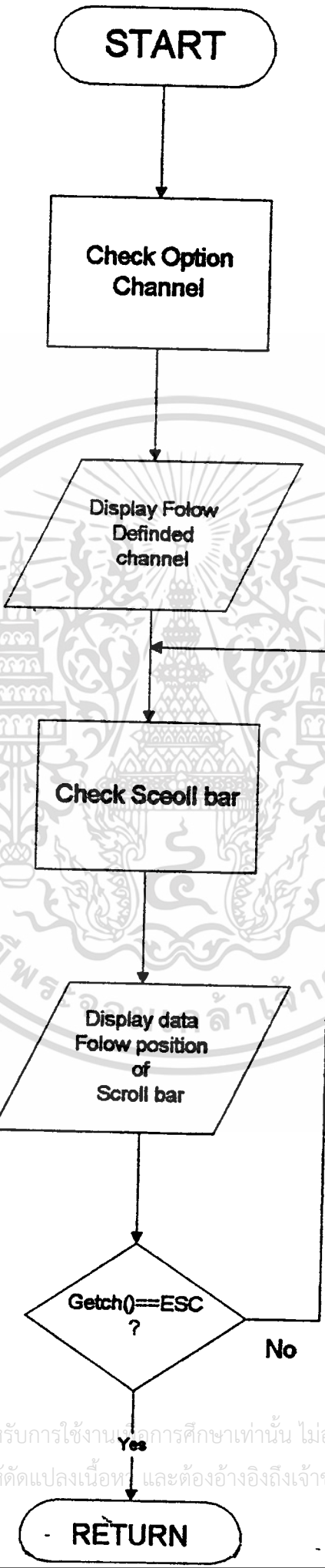
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE



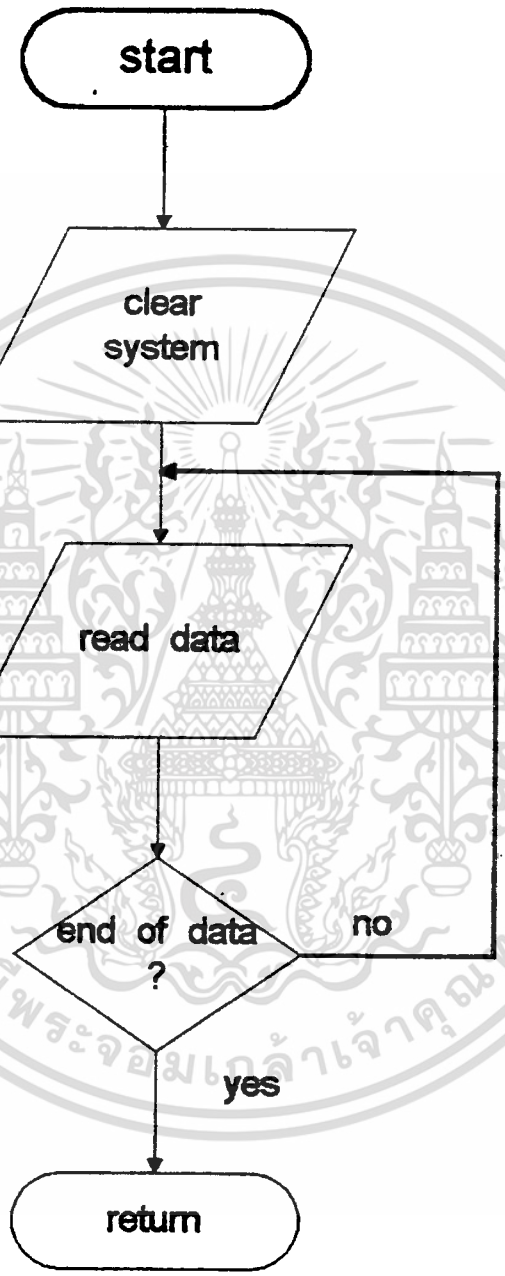
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VIEW



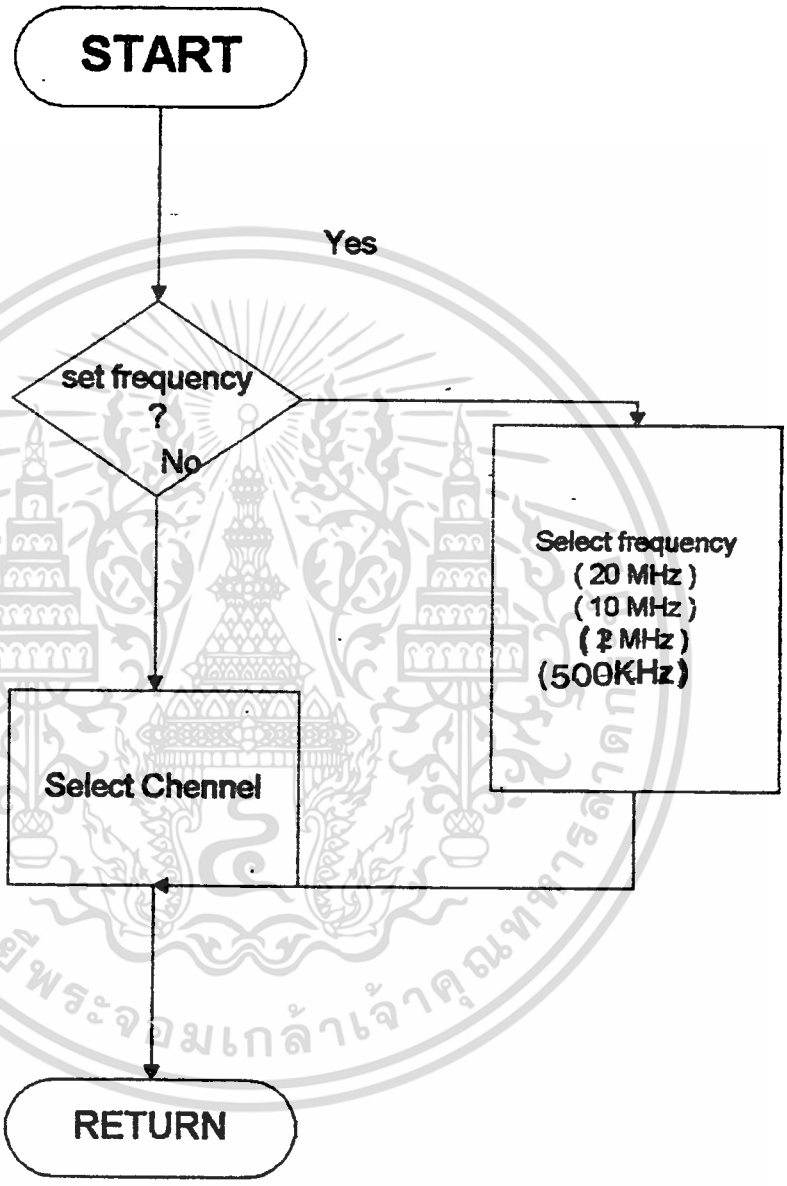
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RUN



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

set option



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองและผลการทดลอง

ในเครื่อง Logic Analyzer ต้นแบบที่ได้ทำการออกแบบมานี้ ได้ออกแบบส่วนควบคุมให้สะดวกต่อการใช้งาน โดยเครื่องสามารถทำการ Test ระบบก่อนนำไปใช้งานได้ ซึ่งก็ได้ทำส่วนของการทดสอบขึ้นคือ การทดสอบของริงเคาท์เตอร์ และการทดสอบของการรันแอดเดรส เป็นต้น การทดสอบก็ทำได้โดยการนำเอาสายคริบมาจับที่ขาอุปกรณ์ที่ต้องการ แล้วใช้โปรแกรมเป็นตัวควบคุมและแสดงที่หน้าจอคอมพิวเตอร์

การใช้งานทางด้านแหล่งจ่ายไฟให้ใช้แหล่งจ่ายภายนอก โดยใช้ 5 โวลต์ 1.5 - 2 แอมป์

รูปที่ 6.1 เมนูในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การทดลอง</u>

1.ต่อแหล่งจ่ายไฟให้กับเครื่องลอจิกอนาไลซ์เซอร์ โดยใช้ขนาด 5 โวลต์

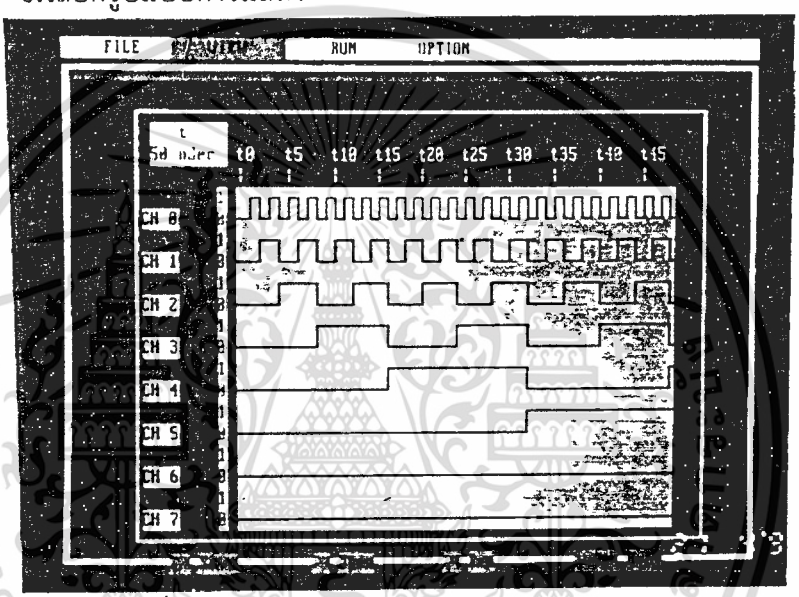
2.เรียกโปรแกรม เมนู เพื่อทำงาน

3.ต่อสายสัญญาณที่ต้องการวัดเข้าที่ขาอินพุทที่เตรียมไว้

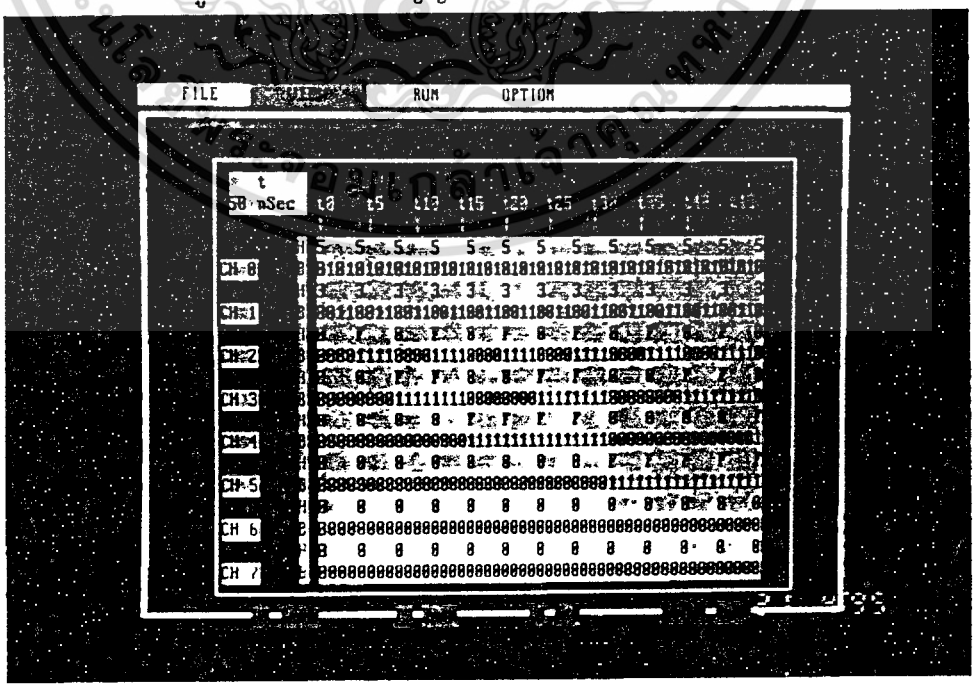
4.เลือก Option ในโปรแกรม

5.Run โปรแกรมในเมนู

6.เลือกรูปแบบการแสดง

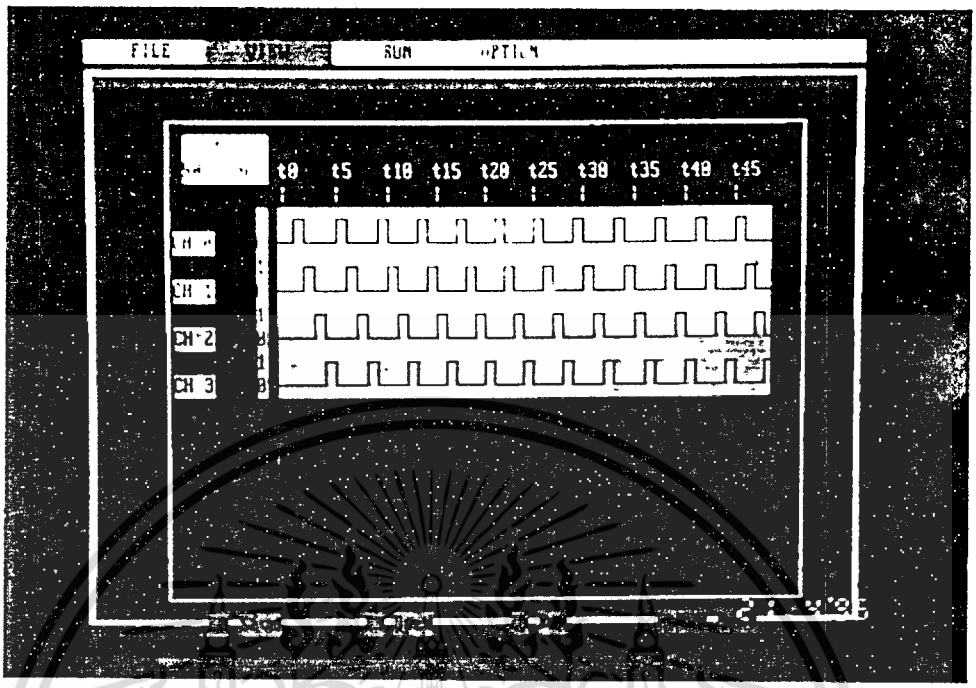


รูปที่ 6.2 การวัดสัญญาณแอดเดรสของระบบ

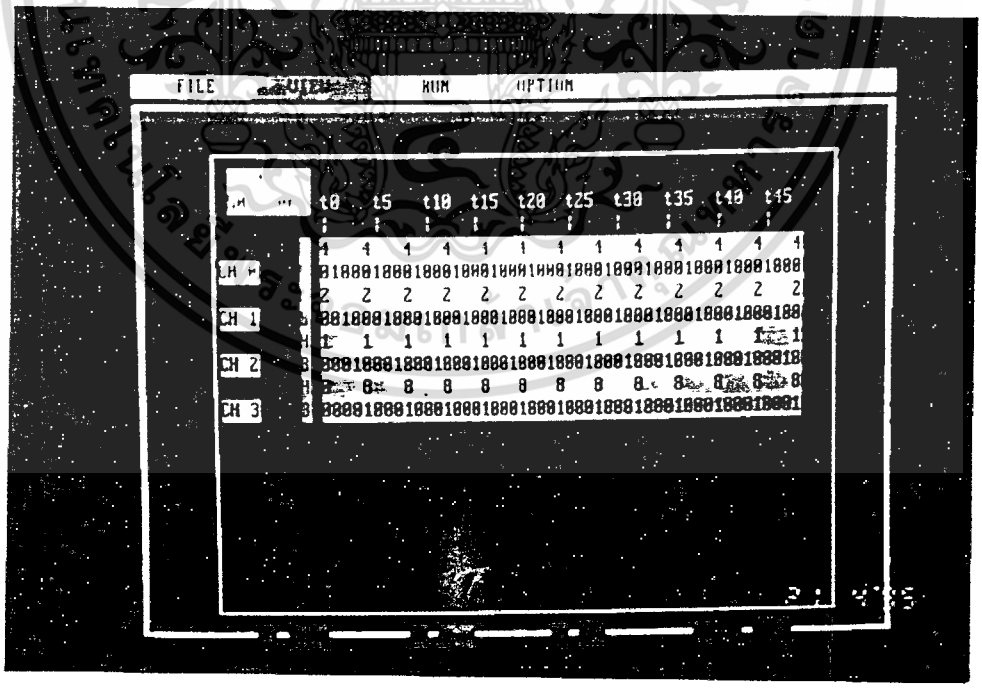


รูปที่ 6.3 การแสดงในรูปเลขฐาน 2 และ ฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 การวัดสัญญาณริงเคาท์เตอร์ ของระบบ



รูปที่ 6.5 การแสดงในรูปเลขฐาน 2 และ ฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทสรุปและวิจารณ์

จากรูป Timing diagram ที่ได้จากการทดลองบทที่ 6 ของเครื่องลอจิกอนาไลซ์เซอร์ 8 แชนแนล จะเห็นได้ว่าความถี่ที่สามารถทำการวัดได้คือ 10 MHz และการแสดงผลนั้นสามารถแสดงได้ทั้งหมด 3 รูปแบบได้แก่ Timing diagram , Binary and Hexadecimal ในการออกแบบนั้นสามารถทำการเก็บข้อมูลได้ 256 ตำแหน่ง ต่อการรัน 1 ครั้ง ซึ่งก็นับว่ายังน้อยและในบางตำแหน่งก็อาจมีความผิดพลาดในการเก็บข้อมูล ซึ่งอาจเป็นไปได้ในเรื่องของอุปกรณ์ที่ทำงานไม่ทันในความถี่สูง แต่ที่ความถี่ต่ำนั้นค่าความผิดพลาดก็ลดลงมากเมื่อเทียบกับที่ความถี่สูง

แนวทางในการพัฒนาเครื่องลอจิกอนาไลซ์เซอร์ สามารถทำการพัฒนาทั้งทางฮาร์ดแวร์และซอร์ฟแวร์ คือทางด้านฮาร์ดแวร์อาจเพิ่มจำนวนช่องการวัดให้มากกว่า 8 แชนแนล และสามารถเก็บข้อมูลได้มากขึ้น และการวัดที่ความถี่ที่สูงขึ้น ส่วนทางด้านซอร์ฟแวร์อาจทำการเพิ่มฟังก์ชั่นให้มากและสะดวกยิ่งขึ้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ยืน ภู่วรวรรณ ทฤษฎีและการใช้งานอิเล็กทรอนิกส์ เล่ม 3 กรุงเทพฯ: นำอักษรการพิมพ์,2532

สุชาย ธนวเสถียรและชัยยงค์ วงศ์ชัยสุวัฒน์ หลักการออกแบบวงจรลอจิก ช พิมพ์ครั้งที่ 6 กรุงเทพฯ:บ.เอเซียเพชรจำกัด,2530

กฤษดา วิศวธีรานนท์ เรียน/เล่น/ใช้ ไอซีดิจิตอล กรุงเทพฯ : เอช-เอน การพิมพ์, 2532

ยืน ภู่วรวรรณ ทฤษฎีและการประยุกต์ใช้งานไมโครโปรเซสเซอร์ กรุงเทพฯ: เอช-เอน การพิมพ์ ,2533

ยืน ภู่วรวรรณ การใช้งานภาษาซี

อนุรักษ์ ชนะกุล วงจรดิจิตอลและการใช้งาน พิมพ์ครั้งที่ 5 อุดรธานี: งานเอกสารการพิมพ์ ว.ท. อุดรธานี,2536

รศ.มัณฑนา ปราการสมุทร การเขียนชุดคำสั่งภาษาซี บ.ดวงกมล จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก) วิธีการใช้งานลอจิกอนาไลซ์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

PC LOGIC ANALYZER 8 CHANNEL

ขั้นตอนการเช็กระบบทางด้านฮาร์ดแวร์

1. ต่อแหล่งจ่ายไฟ(5 โวลท์)เข้าทางด้านหลังเครื่องลอจิกอนาไลซ์เซอร์
2. เสียบการ์ดที่ออกแบบไว้เข้ากับ Slot ของเครื่องคอมพิวเตอร์
3. ต่อสายแพเข้ากับ Connector และต่อสายวัดสัญญาณกับ Connector กับเครื่องลอจิกอนาไลซ์เซอร์
4. เรียกโปรแกรมการทำงาน(MENU) เพื่อเริ่มขั้นตอนการทำงาน

หน้าที่การทำงานของ MENU ต่างๆ

เมื่อทำการเรียกเมนูเพื่อทำงาน จะปรากฏฟังก์ชั่นในการใช้งาน 4 เมนูย่อยดังนี้

- FILE
- VIEW
- RUN
- OPTION

FILE

ประกอบด้วยเมนูย่อยดังนี้

- Open ทำหน้าที่ในการเปิดแฟ้มข้อมูลเก่าที่เก็บไว้ในแผ่น DISK
- Save ทำหน้าที่ในการจัดเก็บแฟ้มข้อมูลลงในแผ่น DISK
- Quit ทำหน้าที่ออกจากระบบการทำงาน

VIEW

ประกอบด้วยเมนูย่อยดังนี้

- Timing Diagram ทำหน้าที่แสดงผลข้อมูลเป็นแบบ Timing บนหน้าจอ Monitor
- Binary Format ทำหน้าที่แสดงผลข้อมูลเป็นแบบ BinaryและHex. บนหน้าจอ Monitor
- Trig_W ทำหน้าที่ในการแสดงข้อมูลในแอดเดรสที่ต้องการ(0-256)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ช) PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <c:\box.h>
#include <c:\w_pulse.h>
#include <c:\cur_on.h>
#include <c:\cur_off.h>
#include <c:\view.h>
#include <c:\start.h>

void main()
{
int i=0,j=0,k=0,l=0,m=0,position,data[256],ch=1,shift=0,flag_key=1,fla
int x=2;
int y=1;
int fr=0,fr1=0;
int c0=1,c1=1,c2=1,c3=1,c4=1,c5=1,c6=1,c7=1;
int test;
int flag_case9=0,f_i,error_buf[1024];
int mode=0;          /* view mode , 0=timing 1=banary  */
FILE *fp;
char buf1[1024],buf2[1024],name[25];
textbackground(0);
clrscr();
cur_off();

textcolor(15);
textbackground(9);
box(2,2,79,25,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
textbackground(0);
box(10,4,73,24,0);

textcolor(4);
textbackground(10);
gotoxy(14,25);
cprintf("F2");
gotoxy(29,25);
cprintf("F3");
gotoxy(44,25);
cprintf("F9");
gotoxy(59,25);
cprintf("ALT-X");

textcolor(0);
textbackground(10);
gotoxy(17,25);
cprintf("Save");
gotoxy(32,25);
cprintf("Open");
gotoxy(47,25);
cprintf("Run");
gotoxy(65,25);
cprintf("Quit");

textcolor(0);
textbackground(10);
gotoxy(x,y);
cprintf("     FILE   ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        i+=1;
       break;
      default: break;
     }

  if(i==0){
    if(ch==13)
    flag_menu=1;

    textcolor(0);
    textbackground(10);
    gotoxy(x,y);
    cprintf("     FILE   ");

    textcolor(0);
    textbackground(7);
    cprintf("    VIEW    ");
    cprintf("     RUN    ");
    cprintf("    OPTION   ");
     }else
      if(i==1){
  if(ch==13)
  flag_menu=2;

  textcolor(0);
  textbackground(7);
  gotoxy(x,y);
  cprintf("     FILE   ");

  textcolor(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
textcolor(0);
textbackground(7);
cprintf("    VIEW    ");
cprintf("     RUN    ");
cprintf("   OPTION                                  ");

while(ch!=45||flag_key!=0)
 {
   if(ch!=13){
     ch=getch();
     if(ch==0){
       ch=getch();
       flag_key=0;
       }else
       flag_key=1;
     }
     if(ch==67)
       flag_menu=3;

     switch(flag_menu){
      case 0:switch(ch)
    {
     case 75: if(i==0)
        i=3;
      else
        i-=1;
      break;
     case 77: if(i==3)
        i=0;
      else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
textbackground(10);
cprintf("    VIEW    ");
textcolor(0);
textbackground(7);
cprintf("     RUN    ");
cprintf("   OPTION   ");
       }else
       if(i==2){
if(ch==13)
flag_menu=3;

textcolor(0);
textbackground(7);
gotoxy(x,y);
cprintf("    FILE    ");
cprintf("    VIEW    ");

textcolor(0);
textbackground(10);
cprintf("     RUN    ");

textcolor(0);
textbackground(7);
cprintf("   OPTION   ");
}else
if(i==3){
 if(ch==13)
  flag_menu=4;
textcolor(0);
textbackground(7);
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gotoxy(x,y);
cprintf("     FILE   ");

cprintf("    VIEW    ");
cprintf("     RUN    ");
textcolor(0);
textbackground(10);
cprintf("   OPTION   ");
}
        break;

        case 1: gettext(3,2,20,6,buf1);

        textcolor(0);
        textbackground(7);
        box(3,2,20,6,0);

        textcolor(0);
        textbackground(10);
        gotoxy(4,3);
        cprintf(" Open          F3 ");

        textcolor(0);
        textbackground(7);
        gotoxy(4,4);
        cprintf(" Save          F2 ");

        gotoxy(4,5);
        cprintf(" Quit      Alt-X ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        flag_menu=10;
        ch=1;
        break;

        case 2: gettext(16,2,33,5,buf1);
        textcolor(0);
        textbackground(7);
        box(16,2,33,5,0);
        ch=1;
        while ((ch!=27)&&(ch!=13)) {
    textcolor(0);
    if (mode) {textbackground(7);} else textbackground(10);
    gotoxy(17,3);
    cprintf(" Timing Diagram ");
    if (mode) {textbackground(10);} else textbackground(7);
    gotoxy(17,4);
    cprintf(" Binary  Format ");
    ch=getch();
    if (ch==0) {ch=getch();}
    if ((ch==72)||(ch==80)) {mode = 1 - mode;} /* toggle mode */
        }
        puttext(16,2,33,5,buf1);
        if (ch==27) {
    ch=1;
    flag_menu=15;
    break;
        }
        position=0;
        if(c0==1)
    position=position|1;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        flag_menu=10;
        ch=1;
        break;

        case 2: gettext(16,2,33,5,buf1);
        textcolor(0);
        textbackground(7);
        box(16,2,33,5,0);
        ch=1;
        while ((ch!=27)&&(ch!=13)) {
    textcolor(0);
    if (mode) {textbackground(7);} else textbackground(10);
    gotoxy(17,3);
    cprintf(" Timing Diagram ");
    if (mode) {textbackground(10);} else textbackground(7);
    gotoxy(17,4);
    cprintf(" Binary  Format ");
    ch=getch();
    if (ch==0) {ch=getch();}
    if ((ch==72)||(ch==80)) {mode = 1 - mode;} /* toggle mode */
        }
        puttext(16,2,33,5,buf1);
        if (ch==27) {
    ch=1;
    flag_menu=15;
    break;
        }
        position=0;
        if(c0==1)
  position=position|1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(c1==1)
  position=position|2;
        if(c2==1)
  position=position|4;
        if(c3==1)
  position=position|8;
        if(c4==1)
  position=position|16;
        if(c5==1)
  position=position|32;
        if(c6==1)
  position=position|64;
        if(c7==1)
  position=position|128;


        test=view(data,position,fr1,mode);


        if(test==0){
    flag_menu=3;
    break;
    }else
     if(test==5){
      ch=45;
      flag_key=0;
      break;
      }


        ch=1;
        flag_menu=0;
  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 3: gotoxy(35,15);
        printf("  Running... ");
        fr1=fr;
        start(data,fr);
        printf("  OK.!");
        ch=1;
        flag_menu=0;
  break;
        case 4: gettext(37,2,52,5,buf1);

        textcolor(0);
        textbackground(7);
        box(37,2,52,5,0);

        textcolor(0);
        textbackground(10);
        gotoxy(38,3);
        cprintf(" Frq_Sampling ");

        textcolor(0);
        textbackground(7);

        gotoxy(38,4);
        cprintf("    channel    ");

        flag_menu=5;
        ch=1;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 5:  flag_case9=0;
switch(ch)
   {
    case 72:if(j==0)
       j=1;
     else
       j=0;
     break;
    case 80:if(j==1)
       j=0;
    else
       j=1;
    break;
    case 27 :flag_menu=0;
     puttext(37,2,52,5,buf1);
     j=0;
     break;
    default: break;
   }
    if(ch==27)
      break;

    if(j==0){
if(ch==13){
 flag_menu=6;
 k=fr;
 break;
 }

    textcolor(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      textbackground(10);
      gotoxy(38,3);
      cprintf(" Frq_Sampling ");

      textcolor(0);
      textbackground(7);

      gotoxy(38,4);
      cprintf("    channel   ");

     }else
     {  if(ch==13){
   flag_menu=7;
   break;
   }
      textcolor(0);
      textbackground(7);
      gotoxy(38,3);
      cprintf(" Frq_Sampling ");

      textcolor(0);
      textbackground(10);

      gotoxy(38,4);
      cprintf("    channel   ");
      }
      break;

         case 6: gettext(39,4,54,9,buf2);
         textcolor(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    textbackground(7);
    box(39,4,54,9,0);

    flag_menu=8;
    ch=13;
    break;
    case 7: gettext(39,5,54,14,buf2);
    textcolor(0);
    textbackground(7);
    box(39,5,54,14,0);

    flag_menu=9;
    ch=13;
    break;

    case 8:  switch(ch)
{
 case 72:if(k==0)
    k=3;
  else
    k-=1;
  break;
 case 80:if(k==3)
    k=0;
  else
   k+=1;
  break;

 case 27 :flag_menu=5;
  puttext(39,4,54,9,buf2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
      default: break;
     }
     if(ch==27)
        break;

        textcolor(0);
        textbackground(7);

      gotoxy(40,5);
      cprintf("     20 MHz   ");
      gotoxy(40,6);
      cprintf("     10 MHz   ");
      gotoxy(40,7);
      cprintf("      1 MHz   ");
      gotoxy(40,8);
      cprintf("    500 KHz   ");

      if(k==0){
       textcolor(0);
       textbackground(10);
       gotoxy(40,5);

        if(ch==13){
          cprintf(" /   20 MHz   ");
          fr=0;
          ch=1;
          }
          else
  cprintf("     20 MHz   ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
      }else
      if(k==1){
       textcolor(0);
       textbackground(10);
       gotoxy(40,6);
       if(ch==13){
cprintf("  /  10 MHz   ");
fr=1;
ch=1;
       }else
        cprintf("     10 MHz   ");
        break;
      }else
      if(k==2){
       textcolor(0);
       textbackground(10);
       gotoxy(40,7);
       if(ch==13){
        cprintf("  /  1 MHz   ");
        fr=2;
        ch=1;
        }else
cprintf("      1 MHz   ");
break;
      }else
      if(k==3){
       textcolor(0);
       textbackground(10);
       gotoxy(40,8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(ch==13){
         cprintf("  / 500 KHz   ");
         fr=3;
         ch=1;
         }else
 cprintf("    500 KHz   ");
         break;
      }
      break;

         case 9:  switch(ch)
      {
       case 72:if(l==0)
          l=7;
        else
          l-=1;
        break;
       case 80:if(l==7)
          l=0;
        else
          l+=1;
        break;
       case 13: if(flag_case9==0)
          flag_case9=1;
        else
         switch(l){
          case 0:if(c0==0)
        c0=1;
       else
        c0=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
   case 1:if(c1==0)
 c1=1;
else
 c1=0;
break;
   case 2:if(c2==0)
 c2=1;
else
 c2=0;
break;
   case 3:if(c3==0)
 c3=1;
else
 c3=0;
break;
   case 4:if(c4==0)
 c4=1;
else
 c4=0;
break;
   case 5:if(c5==0)
 c5=1;
else
 c5=0;
break;
   case 6:if(c6==0)
 c6=1;
else
 c6=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
   case 7:if(c7==0)
 c7=1;
else
 c7=0;
break;
   }
   ch=1;
   break;
case 27 :flag_menu=5;
 puttext(39,5,54,14,buf2);
 break;
default: break;
}

if(ch==27)
 break;

 textcolor(0);
 if(l==0)
  textbackground(10);
  else
   textbackground(7);

 gotoxy(40,6);
 if(c0==0)
  cprintf("     CH 0     ");
  else
  cprintf("  /  CH 0     ");
 if(l==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,7);
if(c1==0)
 cprintf("     CH 1     ");
 else
 cprintf("  /  CH 1     ");
if(l==2)
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,8);
if(c2==0)
 cprintf("     CH 2     ");
 else
 cprintf("  /  CH 2     ");
if(l==3)
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,9);
if(c3==0)
 cprintf("     CH 3     ");
 else
 cprintf("  /  CH 3     ");
if(l==4)
 textbackground(10);
 else
  textbackground(7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gotoxy(40,10);
if(c4==0)
 cprintf("     CH 4     ");
 else
 cprintf("  /  CH 4     ");
if(l==5)
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,11);
if(c5==0)
 cprintf("     CH 5     ");
 else
 cprintf("  /  CH 5     ");
if(l==6)
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,12);
if(c6==0)
 cprintf("     CH 6     ");
 else
 cprintf("  /  CH 6     ");
if(l==7)
 textbackground(10);
 else
  textbackground(7);
gotoxy(40,13);
if(c7==0)
 cprintf("     CH 7     ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        cprintf("  /  CH 7     ");

      break;
         case 10:
    switch(ch)
      {
       case 72:if(m==0)
          m=2;
        else
          m-=1;
        break;
       case 80:if(m==2)
          m=0;
       else
          m+=1;
       break;
       case 27 :puttext(3,2,20,6,buf1);
        flag_menu=0;
        break;
       default: break;
      }
       if(ch==27)
         break;

       if(m==0){
   if(ch==13){
    flag_menu=11;
    break;
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        textcolor(0);
        textbackground(10);
        gotoxy(4,3);
        cprintf(" Open        F3 ");

        textcolor(0);
        textbackground(7);

        gotoxy(4,4);
        cprintf(" Save        F2 ");

        gotoxy(4,5);
        cprintf(" Quit     Alt-X ");

    }else
    if(m==1)
    { if(ch==13){
flag_menu=12;
break;
}

        textcolor(0);
        textbackground(7);
        gotoxy(4,3);
        cprintf(" Open        F3 ");

        textcolor(0);
        textbackground(10);

        gotoxy(4,4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cprintf(" Save        F2 ");

    textcolor(0);
    textbackground(7);

    gotoxy(4,5);
    cprintf(" Quit     Alt-X ");

}else
if(m==2){
  if(ch==13){
    ch=45;
    flag_key=0;
    break;
    }
    textcolor(0);
    textbackground(7);
    gotoxy(4,3);
    cprintf(" Open        F3 ");

    gotoxy(4,4);
    cprintf(" Save        F2 ");

    textcolor(0);
    textbackground(10);
    gotoxy(4,5);
    cprintf(" Quit     Alt-X ");

  }
 break;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 11:gettext(5,4,30,6,buf2);
        textcolor(0);
        textbackground(7);
        box(5,4,30,6,0);
        textcolor(15);
        textbackground(0);
        gotoxy(6,5);
        cprintf("                          ");

        gotoxy(6,5);
        cur_on();
        scanf("%s",name);
        cur_off();

        if((fp=fopen(name,"rb"))==NULL){
  gettext(25,10,53,12,error_buf);
  textcolor(15);
  textbackground(4);
  gotoxy(28,10);
  cprintf("can't open file !..");
  gotoxy(25,12);
  cprintf("press any key to continue");
  getch();
  puttext(25,10,53,12,error_buf);
  puttext(5,4,30,6,buf2);
  puttext(3,2,20,6,buf1);
  flag_menu=1;
  break;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fread(data,sizeof(int),256,fp);
    fread(&fr1,sizeof(int),1,fp);
    fclose(fp);

    puttext(5,4,30,6,buf2);
    puttext(3,2,20,6,buf1);
    flag_menu=0;
    ch=1;

  break;
  case 12:gettext(5,5,30,7,buf2);
  textcolor(0);
  textbackground(7);
  box(5,5,30,7,0);
  textcolor(15);
  textbackground(0);
  gotoxy(6,6);
  cprintf("                    ");

  gotoxy(6,6);
  cur_on();
  scanf("%s",name);
  cur_off();
  if((fp=fopen(name,"wb"))==NULL){
  gettext(25,15,53,17,error_buf);
  textcolor(15);
  textbackground(4);
  gotoxy(28,15);
  cprintf("can't open file !..");
  gotoxy(26,17);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cprintf("press any key to continues");
getch();
puttext(25,15,53,17,error_buf);
puttext(5,5,30,7,buf2);
puttext(3,2,20,6,buf1);
flag_menu=1;
break;
}
  fwrite(data,sizeof(int),256,fp);
  fwrite(&fr1,sizeof(int),1,fp);
  fclose(fp);

  puttext(5,5,30,7,buf2);
  puttext(3,2,20,6,buf1);
  flag_menu=0;
  ch=1;

break;

      default : break;
   }

  }
 cur_on();
 textcolor(7);
 textbackground(0);
 clrscr();
 gotoxy(25,13);
 cprintf("  Goodbye.... ! ");
 delay(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <conio.h>
void box(x1,y1,x2,y2,d)
int x1,y1,x2,y2;
unsigned char d;
{
 int lt,rt,ld,rd,h,v;
 int i;
  if(d==0)
   lt=218,rt=191,ld=192,rd=217,h=196,v=179;
   else
     lt=201,rt=187,ld=200,rd=188,h=205,v=186;
 gotoxy(x1,y1);
 putch(lt);

 for(i=x1+1;i!=x2;i+=1)
   putch(h);
   putch(rt);
 for(i=y1+1;i!=y2;i+=1){
   gotoxy(x2,i);
   putch(v);
   gotoxy(x1,i);
   putch(v);
   }
   gotoxy(x1,y2);
   putch(ld);

  for(i=x1+1;i!=x2;i+=1)
   putch(h);
   putch(rd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void pulse(x1,y1,p,pos,shift,type,start_nibble)
int x1,y1,pos,p[128],shift,type,start_nibble;
/* type = 0      - timing diagram
   1 - binary format     */
{
 int lt,rt,ld,rd,h;
 int i,j,flag=0,temp;

 lt=218,rt=191,ld=192,rd=217,h=196;

 if(shift)
   switch(pos){
 case 0:flag=p[0]&1;
break;
 case 1:flag=p[0]&2;
break;
 case 2:flag=p[0]&4;
break;
 case 3:flag=p[0]&8;
break;
 case 4:flag=p[0]&16;
break;
 case 5:flag=p[0]&32;
break;
 case 6:flag=p[0]&64;
break;
 case 7:flag=p[0]&128;
break;
 }else
flag=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (type==0){
      gotoxy(x1,y1);
      if(flag!=0)
        gotoxy(x1,y1-1);
      }
      else gotoxy(x1,y1);
    for(i=0+shift;i!=49+shift;i+=1) {
       j=i-shift;
       switch(pos){
    case 0:temp=p[i]&1;
   break;
    case 1:temp=p[i]&2;
   break;
    case 2:temp=p[i]&4;
   break;
    case 3:temp=p[i]&8;
   break;
    case 4:temp=p[i]&16;
   break;
    case 5:temp=p[i]&32;
   break;
    case 6:temp=p[i]&64;
   break;
    case 7:temp=p[i]&128;
   break;
      }  /* case */

       switch(type){
   case 0:
   if(temp==0) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(flag==0){
   putch(h);
   gotoxy(x1+j,y1-1);
   putch(' ');
   gotoxy(x1+j+1,y1);
   }
 else
  { putch(rt);
    gotoxy(x1+j,y1);
    putch(ld);
    flag=0;
  }
} /* if */
else   {
if(flag==0)
   { putch(rd);
     gotoxy(x1+j,y1-1);

     putch(lt);
     flag=1;
   }else
   {
   putch(h);
   gotoxy(x1+j,y1);
   putch(' ');
   gotoxy(x1+j+1,y1-1);
   }
}
break;
case 1:if (temp==0) putch('0'); else putch('1');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(x1+j,y1-1);
        if (i%4==start_nibble) {putch(hex(p,i,pos));} else putch(' ');
        gotoxy(x1+j+1,y1);
        break;
      } /* case */
  }    /* for */

} /* pulse */

int hex(p,i,pos)
int p[128],i,pos;
{
int ch=0;
    if ((p[i  ]&pow(2,pos))!=0) { ch+=8;}
    if ((p[i+1]&pow(2,pos))!=0) { ch+=4;}
    if ((p[i+2]&pow(2,pos))!=0) { ch+=2;}
    if ((p[i+3]&pow(2,pos))!=0) { ch++;}
    if (ch<10) {ch+=48;} else ch+=55;
    return ch;
}

int pow(base,power)
int base,power;
{
  int i,j=1;
  if (power==0) { return 1;}
  else {
    for (i=1;i<=power;i++) {
      j=j*2;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return j;
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void pulse(x1,y1,p,pos,shift,type,start_nibble)
int x1,y1,pos,p[128],shift,type,start_nibble;
/* type = 0      - timing diagram
  1 - binary format     */
{
 int lt,rt,ld,rd,h;
 int i,j,flag=0,temp;

 lt=218,rt=191,ld=192,rd=217,h=196;

 if(shift)
   switch(pos){
 case 0:flag=p[0]&1;
break;
 case 1:flag=p[0]&2;
break;
 case 2:flag=p[0]&4;
break;
 case 3:flag=p[0]&8;
break;
 case 4:flag=p[0]&16;
break;
 case 5:flag=p[0]&32;
break;
 case 6:flag=p[0]&64;
break;
 case 7:flag=p[0]&128;
break;
 }else
flag=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (type==0){
      gotoxy(x1,y1);
      if(flag!=0)
        gotoxy(x1,y1-1);
      }
      else gotoxy(x1,y1);
    for(i=0+shift;i!=49+shift;i+=1) {
       j=i-shift;
       switch(pos){
    case 0:temp=p[i]&1;
   break;
    case 1:temp=p[i]&2;
   break;
    case 2:temp=p[i]&4;
   break;
    case 3:temp=p[i]&8;
   break;
    case 4:temp=p[i]&16;
   break;
    case 5:temp=p[i]&32;
   break;
    case 6:temp=p[i]&64;
   break;
    case 7:temp=p[i]&128;
   break;
      }   /* case */

       switch(type){
   case 0:
   if(temp==0) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(flag==0){
   putch(h);
   gotoxy(x1+j,y1-1);
   putch(' ');
   gotoxy(x1+j+1,y1);
   }
 else
  { putch(rt);
    gotoxy(x1+j,y1);
    putch(ld);
    flag=0;
  }
} /* if */
else   {
if(flag==0)
   { putch(rd);
     gotoxy(x1+j,y1-1);

     putch(lt);
     flag=1;
   }else
   {
   putch(h);
   gotoxy(x1+j,y1);
   putch(' ');
   gotoxy(x1+j+1,y1-1);
   }
}
break;
case 1:if (temp==0) putch('0'); else putch('1');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(x1+j,y1-1);
        if (i%4==start_nibble) {putch(hex(p,i,pos));} else putch(' ');
        gotoxy(x1+j+1,y1);
        break;
      } /* case */
    }    /* for */

} /* pulse */

int hex(p,i,pos)
int p[128],i,pos;
{
int ch=0;
    if ((p[i  ]&pow(2,pos))!=0) { ch+=8;}
    if ((p[i+1]&pow(2,pos))!=0) { ch+=4;}
    if ((p[i+2]&pow(2,pos))!=0) { ch+=2;}
    if ((p[i+3]&pow(2,pos))!=0) { ch++;}
    if (ch<10) {ch+=48;} else ch+=55;
    return ch;
}

int pow(base,power)
int base,power;
{
  int i,j=1;
  if (power==0) { return 1;}
  else {
    for (i=1;i<=power;i++) {
      j=j*2;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return j;
    }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int view(data,position,t,mode)
int data[256],position,t,mode;
{
int i,j,ch,shift=0,stnib;
char time[13];

switch(t){
case 0:strcpy(time,"50 nSec ");
       break;
case 1:strcpy(time,"100 nSec");
       break;
case 2:strcpy(time," 1 uSec ");
       break;
case 3:strcpy(time," 2 uSec ");
       break;
default:break;
  }

textcolor(4);
textbackground(7);

gotoxy(12,5);
cprintf("   t    ");

gotoxy(12,6);
cprintf("%s",time);
textcolor(3);
textbackground(0);

gotoxy(21,6);
```

เอกสารนี้มีไว้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("                                                    ");
gotoxy(21,7);
for(i=0;i!=49;i+=1)
 if((i % 5)==0){
   gotoxy(21+i,7);
   cprintf("¦");
   gotoxy(21+i,6);
   cprintf("t%d",i);
   }
   else
   {gotoxy(21+i,7);
    cprintf(" ");
    }
textcolor(9);
textbackground(7);

write_pulse(&data[0],position,0,mode,0);

  i=0;
while(ch!=27)
 { ch=getch();
   if(ch==0)
    ch=getch();
   switch(ch)
   {
    case 75: if(i!=0)
        i-=1;
else
shift=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

       case 77: if(i==0&shift==0)
         shift=1;
         else
          if(i!=195)
    i+=1;

         break;
       case 67:return(0);
       break;
       case 45:return(5);
       break;
       default: break;
       }

    textcolor(3);
    textbackground(0);

    gotoxy(21,7);
    for(j=i+shift;j!=i+shift+49;j+=1)
     if((j % 5)==0){
       cprintf("¦");
       }
       else
        cprintf(" ");

    gotoxy(21,6);
    printf("                                                    ");
    gotoxy(21,6);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(j=i+shift;j!=i+shift+49;j+=1)
 if(((j) % 5)==0){
   gotoxy(21+j-i-shift,6);
   cprintf("t%d",j);
   }
textcolor(9);
textbackground(7);

switch(i%4){
  case 0:stnib=0;break;
  case 1:stnib=3;break;
  case 2:stnib=2;break;
  case 3:stnib=1;break;
}
write_pulse(&data[i],position,shift,mode,stnib);
}
return(1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <c:\puls.h>
void write_pulse(data,position,shift,type,st_nib)
int position,data[256],shift,type,st_nib;
/*  type = 0  - timing diagram
    1  - binary          */
{
 int i,x=21,y=9,temp=1,temp2,count_line=0;

 for(i=0;i!=8;i+=1){
    if(position&temp){
count_line+=1;

textcolor(4);
textbackground(7);
gotoxy(x-10,y);
cprintf("CH %d",i);

textcolor(5);
textbackground(7);
gotoxy(x-2,y);
if (type==0) {putch('0');} else putch('B');
gotoxy(x-2,y-1);
if (type==0) {putch('1');} else putch('H');

textcolor(9);
textbackground(7);
pulse(x,y,data,i,shift,type,st_nib);
y+=2;
}
temp=temp*2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

textcolor(15);
textbackground(0);
temp2=count_line;
count_line=(8-count_line)*2;
for(i=0;i!=count_line;i+=1)
{
gotoxy(11,temp2*2+8+i);
cprintf("%s","
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void cur_on()
{
  union REGS reg;
  reg.h.ch = 0x00;
  reg.h.cl = 13;
  reg.h.ah = 1;
  int86(0x10,&reg,&reg);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void cur_off()
{
  union REGS reg;
  reg.h.ch = 0x20;
  reg.h.cl = 0;
  reg.h.ah = 1;
  int86(0x10,&reg,&reg);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค) DATA CHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2-12: The 6116: Another Static RAM Device

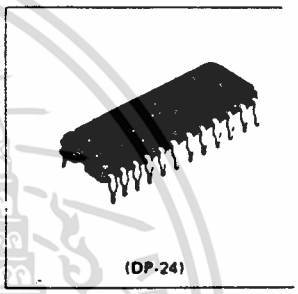
In this section we will present another popular static RAM that can be easily used in a Z80-controlled system: the 6116—a 2K × 8, common I/O static RAM. Figure 2.19 shows a partial data sheet for this device. The 6116 has operating characteristics that are similar to the 2114 static RAM.

HM6116P-2, HM6116P-3, HM6116P-4

2048-word × 8-bit High Speed Static CMOS RAM

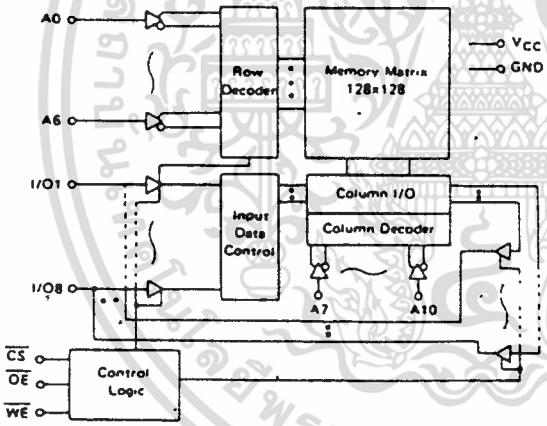
■ FEATURES

- Single 5V Supply and High Density 24 pin Package
- High Speed Fast Access Time 120ns/150ns/200ns (max.)
- Low Power Standby and Low Power Operation: Standby 100μW (typ.)
  Operation: 180mW (typ.)
- Completely Static RAM. No clock or Timing Strobe Required
- Directly TTL Compatible: All Input and Output
- Pin Out Compatible with Standard 16K EPROM/MASK ROM
- Equal Access and Cycle Time

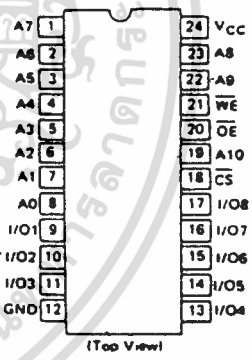


(DP-24)

■ FUNCTIONAL BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Rating	Unit
Voltage on Any Pin Relative to GND	V_{IN}	−0.5 to +7.0	V
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Temperature Under Bias	T_{bias}	−10 to +85	°C
Power Dissipation	P_T	1.0	W

■ TRUTH TABLE

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	Mode	V_{CC} Current	I/O Pin	Ref. Cycle
H	X	X	Not Selected	I_{SB}, I_{SB1}	High Z	
L	L	H	Read	I_{CC}	Dout	Read Cycle (1) ~ (3)
L	H	L	Write	I_{CC}	Din	Write Cycle (1)
L	L	L	Write	I_{CC}	Din	Write Cycle (2)

Figure 2.19: A data sheet for the 6116—a 2K by 8-bit, common I/O static RAM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ RECOMMENDED DC OPERATING CONDITIONS (T_a = 0 to +70°C)

Item	Symbol	min	typ.	max	Unit
Supply Voltage	V_{CC}	4.5	5.0	5.5	V
	GND	0	0	0	V
Input Voltage	V_{IH}	2.2	3.5	6.0	V
	V_{IL}	−1.0*	–	0.8	V

* Pulse Width, 50 ns, DC V_{IL} min = −0.3V

■ DC AND OPERATING CHARACTERISTICS (V_{CC} = 5V ±10%, GND = 0V, T_a = 0 to +70°C)

Item	Symbol	Test Conditions	HM6116P-2			HM6116P-3/-4			Unit
			min	typ*	max	min	typ*	max	
Input Leakage Current	$\|I_{LI}\|$	V_{CC} = 5.5V, V_{in} = GND to V_{CC}	–	–	10	–	–	10	µA
Output Leakage Current	$\|I_{LO}\|$	$\overline{CS}$ = V_{IH} or $\overline{OE}$ = V_{IH}, $V_{I/O}$ = GND to V_{CC}	–	–	10	–	–	10	µA
Operating Power Supply Current	I_{CC}	$\overline{CS}$ = V_{IL}, $I_{I/O}$ = 0mA	–	40	80	–	35	70	mA
	I_{CC1}**	V_{IH} = 3.5V, V_{IL} = 0.6V, $I_{I/O}$ = 0mA	–	35	–	–	30	–	mA
Average Operating Current	I_{CC2}	Min. cycle, duty = 100%	–	40	80	–	35	70	mA
Standby Power Supply Current	I_{SB}	$\overline{CS}$ = V_{IH}	–	5	15	–	5	15	mA
	I_{SB1}	$\overline{CS}$ ≥ V_{CC} −0.2V, V_{in} ≥ V_{CC} −0.2V or V_{in} ≤ 0.2V	–	0.02	2	–	0.02	2	mA
Output Voltage	V_{OL}	I_{OL} = 4mA	–	–	0.4	–	–	–	V
		I_{OL} = 2.1mA	–	–	–	–	–	0.4	
	V_{OH}	I_{OH} = −1.0mA	2.4	–	–	2.4	–	–	V

* V_{CC} = 5V, T_a = 25°C
** Reference Only

■ AC CHARACTERISTICS (V_{CC} = 5V ±10%, T_a = 0 to +70°C)

● AC TEST CONDITIONS

Input Pulse Levels: 0.8 to 2.4V
Input Rise and Fall Times: 10 ns
Input and Output Timing Reference Levels: 1.5V
Output Load: 1TTL Gate and C_L = 100pF (including scope and Jig)

● READ CYCLE

Item	Symbol	HM6116P-2		HM6116P-3		HM6116P-4		Unit
		min	max	min	max	min	max	
Read Cycle Time	t_{RC}	120	–	150	–	200	–	ns
Address Access Time	t_{AA}	–	120	–	150	–	200	ns
Chip Select Access Time	t_{ACS}	–	120	–	150	–	200	ns
Chip Selection to Output in Low Z	t_{CLZ}	10	–	15	–	15	–	ns
Output Enable to Output Valid	t_{OE}	–	80	–	100	–	120	ns
Output Enable to Output in Low Z	t_{OLZ}	10	–	15	–	15	–	ns
Chip deselection to Output in High Z	t_{CHZ}	0	40	0	50	0	60	ns
Chip Disable to Output in High Z	t_{OHZ}	0	40	0	50	0	60	ns
Output Hold from Address Change	t_{OH}	10	–	15	–	15	–	ns

● WRITE CYCLE

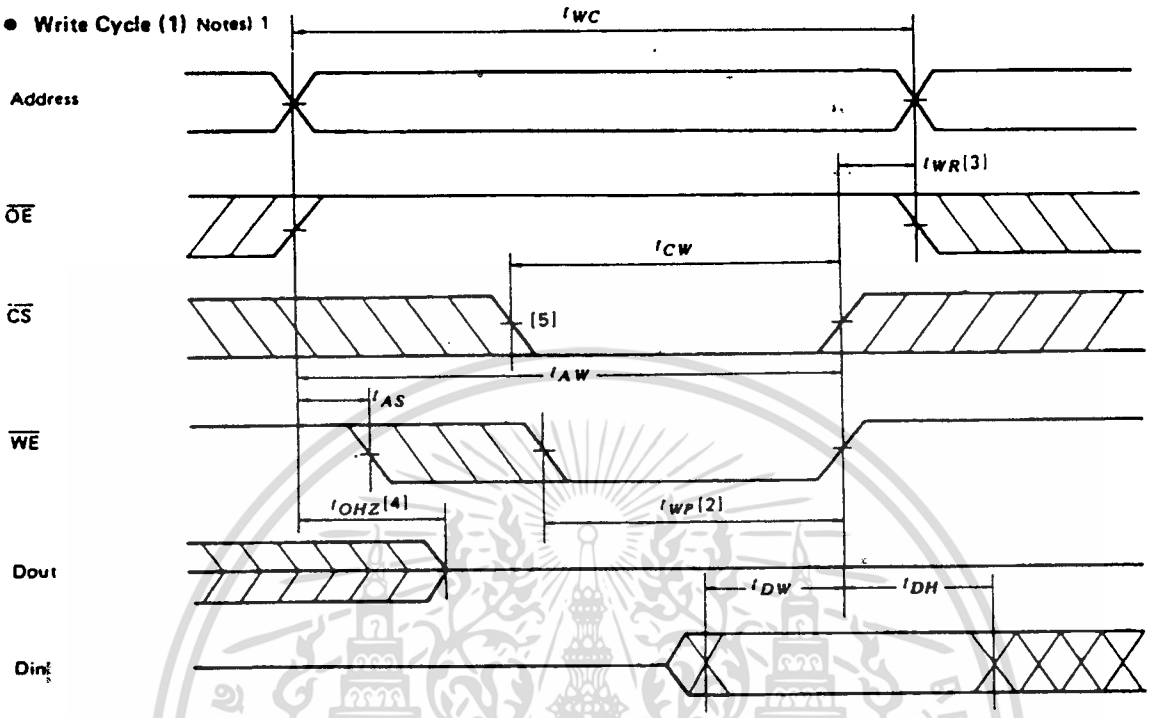
Item	Symbol	HM6116P-2		HM6116P-3		HM6116P-4		Unit
		min	typ	min	max	min	max	
Write Cycle Time	t_{WC}	120	–	150	–	200	–	ns
Chip Selection to End of Write	t_{CW}	70	–	90	–	120	–	ns
Address Valid to End of Write	t_{AW}	105	–	120	–	140	–	ns
Address Set-Up Time	t_{AS}	20	–	20	–	20	–	ns
Write Pulse Width	t_{WP}	70	–	90	–	120	–	ns
Write Recovery Time	t_{WR}	5	–	10	–	10	–	ns
Output Disable to Output in High Z	t_{OHZ}	0	40	0	50	0	60	ns
Write to Output in High Z	t_{WHZ}	0	50	0	60	0	60	ns
Data to Write Time Overlap	t_{DW}	35	–	40	–	50	–	ns
Data Hold from Write Time	t_{DH}	5	–	10	–	10	–	ns
Output Active from End of Write	t_{OW}	5	–	10	–	10	–	ns

Figure 2.19: Data sheet continued.

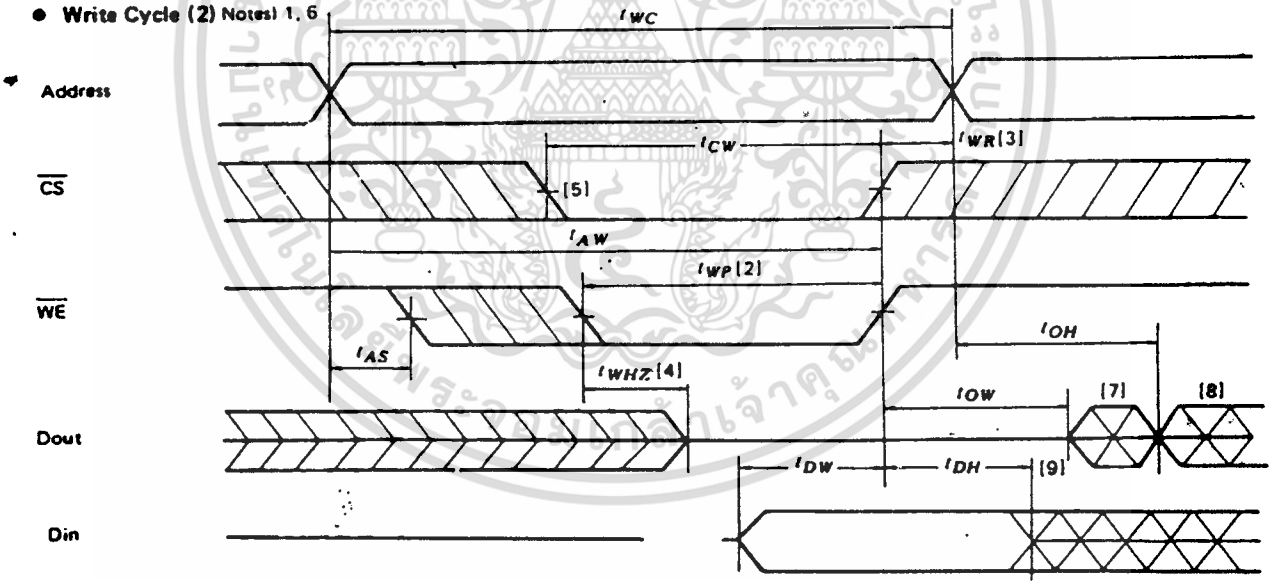
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ TIMING WAVEFORM

● **Write Cycle (1)** Notes) 1



● **Write Cycle (2)** Notes) 1, 6



NOTES:
1. $\overline{WE}$ must be high during all address transitions.
2. A write occurs during the overlap (t_{WP}) of a low $\overline{CS}$ and a low $\overline{WE}$.
3. t_{WR} is measured from the earlier of $\overline{CS}$ or $\overline{WE}$ going high to the end of write cycle.
4. During this period, I/O pins are in the output state so that the input signals of opposite phase to the outputs must not be applied.
5. If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transitions or after the $\overline{WE}$ transition, output remain in a high impedance state.
6. $\overline{OE}$ is continuously low. ($\overline{OE}$ = V_{IL})
7. D_{out} is the same phase of write data of this write cycle.
8. D_{out} is the read data of next address.
9. If $\overline{CS}$ is Low during this period, I/O pins are in the output state. Then the data input signals of opposite phase to the outputs must not be applied to them.

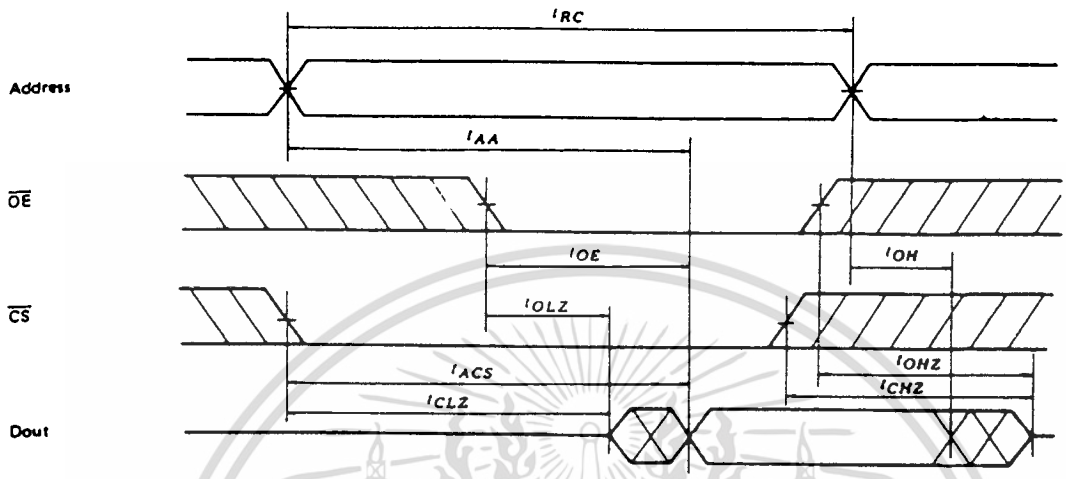
Figure 2.19: Data sheet continued.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

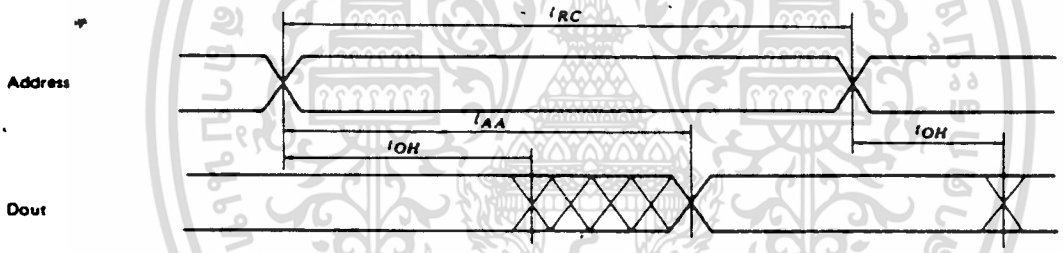
■ CAPACITANCE (f = 1MHz, T_a = 25°C)

Item	Symbol	Test Conditions	typ.	max.	Unit
Input Capacitance	C_{in}	V_{in} = 0V	3	5	pF
Input/Output Capacitance	$C_{I/O}$	$V_{I/O}$ = 0V	5	7	pF

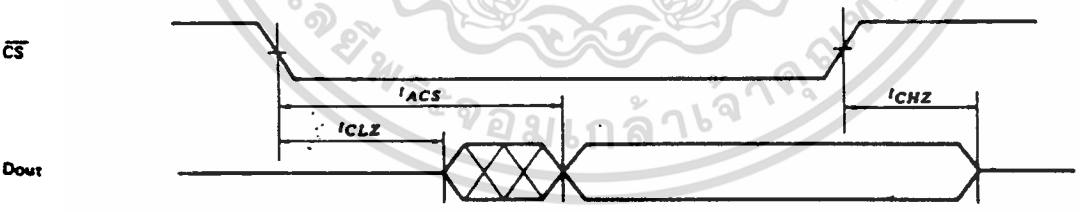
● **Read Cycle (1)** Notes) 1, 5



● **Read Cycle (2)** Notes) 1, 2, 4, 5



● **Read Cycle (3)** Notes) 1, 3, 4, 5



NOTES: 1. $\overline{WE}$ is High for Read Cycle.
2. Device is continuously selected, $\overline{CS}$ = V_{IL}.
3. Address Valid prior to or coincident with $\overline{CS}$ transition Low.
4. $\overline{OE}$ = V_{IL}.
5. When $\overline{CS}$ is Low, the address input must not be in the high impedance state.

Figure 2.19: Data sheet continued.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

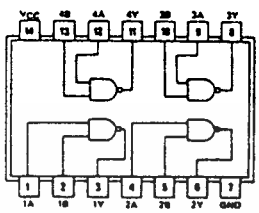
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

03

positive logic:
$Y = \overline{AB}$

See page 6-4



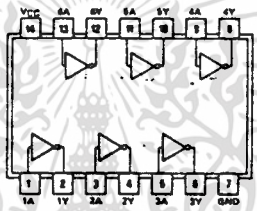
SN5403 (J) SN7403 (J, N)
SN54L03 (J) SN74L03 (J, N)
SN54LS03 (J, W) SN74LS03 (J, N)
SN54S03 (J, W) SN74S03 (J, N)

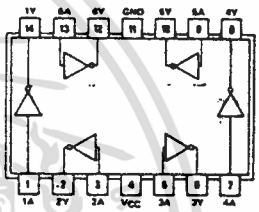
HEX INVERTERS

04

positive logic:
$Y = \overline{A}$

See page 6-2





SN5404 (J) SN7404 (J, N)
SN54H04 (J) SN74H04 (J, N)
SN54L04 (J) SN74L04 (J, N)
SN54LS04 (J, W) SN74LS04 (J, N)
SN54S04 (J, W) SN74S04 (J, N)

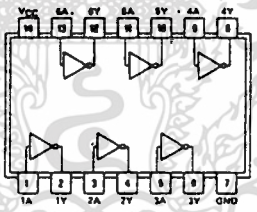
SN5404 (W)
SN54H04 (W)
SN54L04 (T)

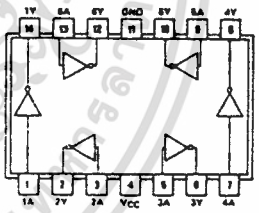
HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS

05

positive logic:
$Y = \overline{A}$

See page 6-4





SN5405 (J) SN7405 (J, N)
SN54H05 (J) SN74H05 (J, N)
SN54LS05 (J, W) SN74LS05 (J, N)
SN54S05 (J, W) SN74S05 (J, N)

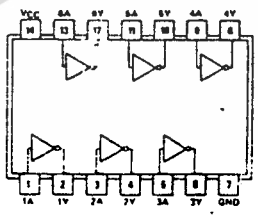
SN5405 (W)
SN54H05 (W)

HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS

06

positive logic:
$Y = \overline{A}$

See page 6-24



SN5406 (J, W) SN7406 (J, N)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

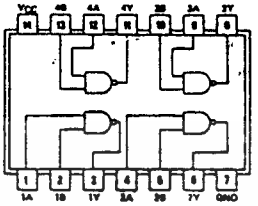
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT HIGH-VOLTAGE INTERFACE POSITIVE-NAND GATES

26

positive logic:
$Y = \overline{AB}$

See pages 6-24 and 6-26



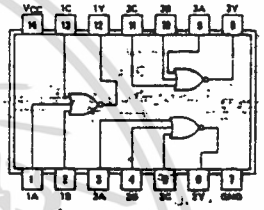
SN5426 (J) SN7426 (J, N)
SN54LS26 (J, W) SN74LS26 (J, N)

TRIPLE 3-INPUT POSITIVE-NOR GATES

27

positive logic:
$Y = \overline{A+B+C}$

See page 6-8



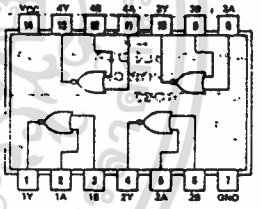
SN5427 (J, W) SN7427 (J, N)
SN54LS27 (J, W) SN74LS27 (J, N)

QUADRUPLE 2-INPUT POSITIVE-NOR BUFFERS

28

positive logic:
$Y = \overline{A+B}$

See page 6-20



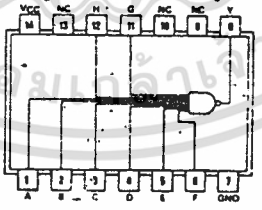
SN5428 (J, W) SN7428 (J, N)
SN54LS28 (J, W) SN74LS28 (J, N)

8-INPUT POSITIVE-NAND GATES

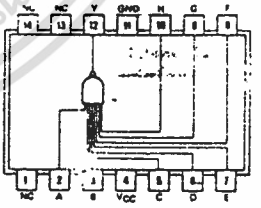
30

positive logic:
$Y = \overline{ABCDEFGH}$

See page 6-2



SN5430 (J) SN7430 (J, N)
SN54H30 (J) SN74H30 (J, N)
SN54L30 (J) SN74L30 (J, N)
SN54LS30 (J, W) SN74LS30 (J, N)
SN54S30 (J, W) SN74S30 (J, N)



SN5430 (W)
SN54H30 (W)
SN54L30 (T)

NC—No internal connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

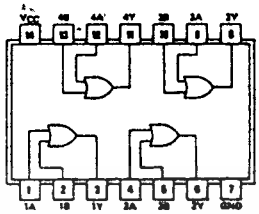
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT POSITIVE-OR GATES

32

positive logic:
$Y = A+B$

See page 6-28



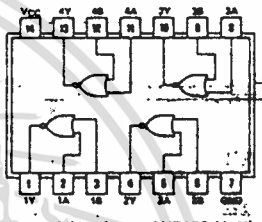
SN5432 (J, W) SN7432 (J, N)
SN54LS32 (J, W) SN74LS32 (J, N)
SN54S32 (J, W) SN74S32 (J, N)

QUADRUPLE 2-INPUT POSITIVE-NOR BUFFERS WITH OPEN-COLLECTOR OUTPUTS

33

positive logic:
$Y = \overline{A+B}$

See pages 6-24 and 6-26



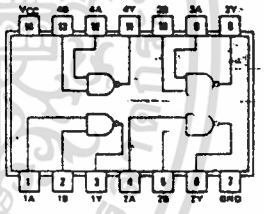
SN5433 (J, W) SN7433 (J, N)
SN54LS33 (J, W) SN74LS33 (J, N)

QUADRUPLE 2-INPUT POSITIVE-NAND BUFFERS

37

positive logic:
$Y = \overline{AB}$

See page 6-20



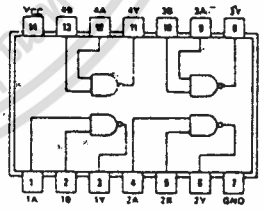
SN5437 (J, W) SN7437 (J, N)
SN54LS37 (J, W) SN74LS37 (J, N)
SN54S37 (J, W) SN74S37 (J, N)

QUADRUPLE 2-INPUT POSITIVE-NAND BUFFERS WITH OPEN-COLLECTOR OUTPUTS

38

positive logic:
$Y = \overline{AB}$

See pages 6-24 and 6-26



SN5438 (J, W) SN7438 (J, N)
SN54LS38 (J, W) SN74LS38 (J, N)
SN54S38 (J, W) SN74S38 (J, N)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN ASSIGNMENTS (TOP VIEWS)

AND-GATED J-K MASTER-SLAVE FLIP-FLOPS WITH PRESET AND CLEAR

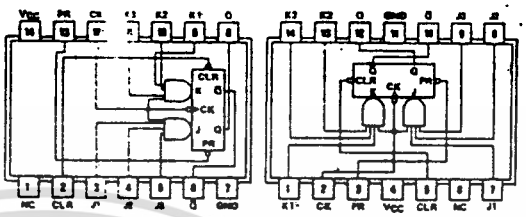
72

FUNCTION TABLE

INPUTS					OUTPUTS	
PRESET	CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	⊓_	L	L	Q_0	$\bar{Q}_0$
H	H	⊓_	H	L	H	L
H	H	⊓_	L	H	L	H
H	H	⊓_	H	H	TOGGLE	

positive logic: J = J1·J2·J3; K1·K2·K3

See pages 6-46, 6-50, and 6-54



SN5472 (J) SN7472 (J, N) SN5472 (W)
SN54H72 (J) SN74H72 (J, N) SN54H72 (W)
SN54L72 (J) SN74L72 (J, N) SN54L72 (T)

NC—No internal connection

DUAL J-K FLIP-FLOPS WITH CLEAR

73

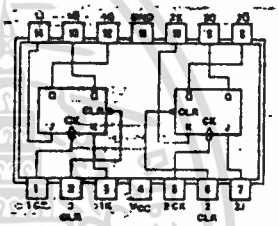
'73, 'H73, 'L73
FUNCTION TABLE

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	⊓_	L	L	Q_0	$\bar{Q}_0$
H	⊓_	H	L	H	L
H	⊓_	L	H	L	H
H	⊓_	H	H	TOGGLE	

'LS73A
FUNCTION TABLE

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	⊓_	L	L	Q_0	$\bar{Q}_0$
H	⊓_	H	L	H	L
H	⊓_	L	H	L	H
H	⊓_	H	H	TOGGLE	
H	H	X	X	Q_0	$\bar{Q}_0$

See pages 6-46, 6-50, 6-54, and 6-56



SN5473 (J, W) SN7473 (J, N)
SN54H73 (J, W) SN74H73 (J, N)
SN54L73 (J, T) SN74L73 (J, N)
SN54LS73A (J, W) SN74LS73A (J, N)

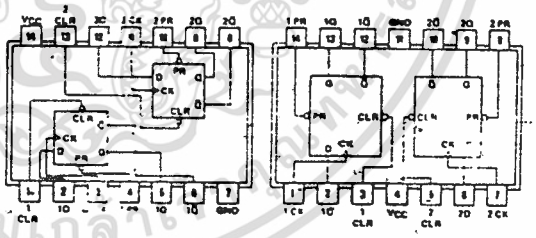
DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

74

FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	⊓_	H	H	L
H	H	⊓_	L	L	H
H	H	L	X	Q_0	$\bar{Q}_0$

See pages 6-46, 6-50, 6-54, and 6-56



SN5474 (J) SN7474 (J, N) SN5474 (W)
SN54H74 (J) SN74H74 (J, N) SN54H74 (W)
SN54L74 (J) SN74L74 (J, N) SN54L74 (T)
SN54LS74A (J, W) SN74LS74A (J, N)
SN54S74 (J, W) SN74S74 (J, N)

See explanation of function tables on page 3-8.

* This configuration is nonstable; that is, it will not persist when preset and clear inputs return to their inactive (high) level.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL MSI

TYPES SN5490A, SN5492A, SN5493A, SN54L90, SN54L93, SN54LS90, SN54LS92, SN54LS93, SN7490A, SN7492A, SN7493A, SN74L90, SN74L93, SN74LS90, SN74LS92, SN74LS93 DECADE, DIVIDE-BY-TWELVE, AND BINARY COUNTERS

BULLETIN NO. DL-S 7411807, MARCH 1974 – REVISED OCTOBER 1976

'90A, 'L90, 'LS90 . . . DECADE COUNTERS

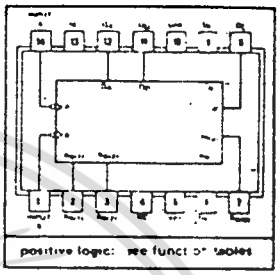
'92A, 'LS92 . . . DIVIDE-BY-TWELVE COUNTERS

'93A, 'L93, 'LS93 . . . 4-BIT BINARY COUNTERS

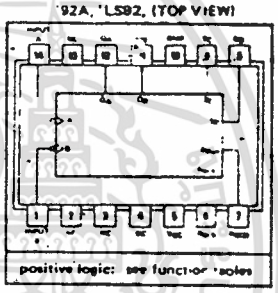
TYPES	TYPICAL POWER DISSIPATION
'90A	145 mW
'L90	20 mW
'LS90	45 mW
'92A, '93A	130 mW
'LS92, 'LS93	46 mW
'L93	16 mW

SN54', SN54LS' . . . J OR W PACKAGE
SN54L' . . . J OR T PACKAGE
SN64', SN74L', SN74LS' . . . J OR N PACKAGE

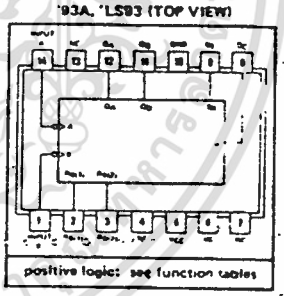
'90A, 'L90, 'LS90 (TOP VIEW)



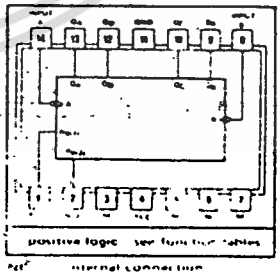
'92A, 'LS92, (TOP VIEW)



'93A, 'LS93 (TOP VIEW)



'L93 (TOP VIEW)



description

Each of these monolithic counters contains four master-slave flip-flops and additional gating to provide a divide-by-two counter and a three-stage binary counter for which the count cycle length is divide-by-five for the '90A, 'L90, and 'LS90, divide-by-six for the '92A and 'LS92, and divide-by-eight for the '93A, 'L93, and 'LS93.

All of these counters have a gated zero reset and the '90A, 'L90, and 'LS90 also have gated set-to-nine inputs for use in BCD nine's complement applications.

To use their maximum count length (decade, divide-by-twelve, or four-bit binary) of these counters, the B input is connected to the Q_A output. The input count pulses are applied to input A and the outputs are as described in the appropriate function table. A symmetrical divide-by-ten count can be obtained from the '90A, 'L90, or 'LS90 counters by connecting the Q_D output to the A input and applying the input count to the B input which gives a divide-by-ten square wave at output Q_A.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'90A, 'L90, 'LS90
BCD COUNT SEQUENCE
(See Note A)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

'90A, 'L90, 'LS90
BI-QUINARY (5-2)
(See Note B)

COUNT	OUTPUT			
	Q_A	Q_D	Q_C	Q_B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

'92A, 'LS92
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

'93A, 'L93, 'LS93
COUNT SEQUENCE
(See Note C)

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

'90A, 'L90, 'LS90
RESET/COUNT FUNCTION TABLE

RESET INPUTS				OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	H
X	X	H	H	L	L	H	L
X	L	X	L	COUNT			
L	X	L	X	COUNT			
X	X	X	L	COUNT			
X	L	L	X	COUNT			

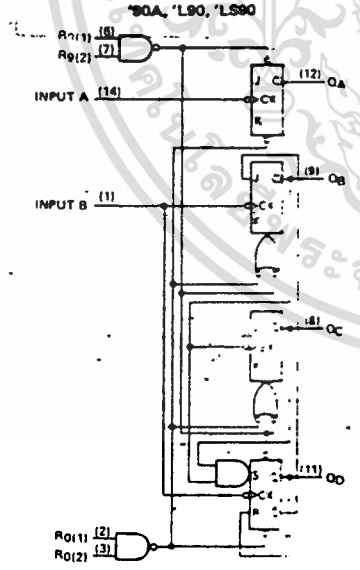
'92A, 'LS92, '93A, 'L93, 'LS93
RESET/COUNT FUNCTION TABLE

RESET INPUTS		OUTPUT			
$R_{0(1)}$	$R_{0(2)}$	Q_D	Q_C	Q_B	Q_A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

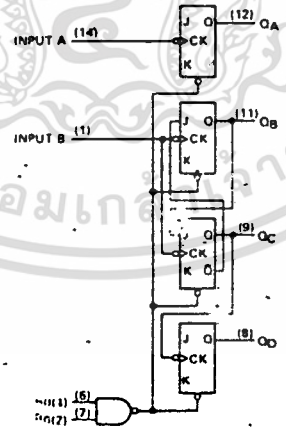
NOTES: A. Output Q_A is connected to input B for BCD count.
B. Output Q_D is connected to input A for bi-quinary count.
C. Output Q_A is connected to input B.
D. H = high level, L = low level, X = irrelevant

functional block diagrams

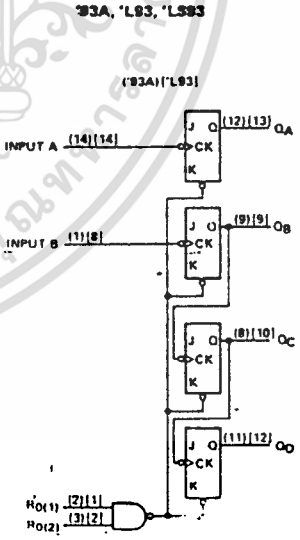
'90A, 'L90, 'LS90



'92A, 'LS92



'93A, 'L93, 'LS93



The J and K inputs shown without connection are for reference only and are functionally at a high level.

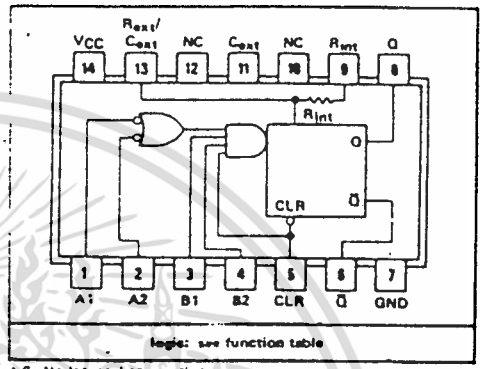
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 501 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES SN54122, SN54123, SN54L122, SN54L123, SN54LS122, SN54LS123, SN74122, SN74123, SN74L122, SN74L123, SN74LS122, SN74LS123
RETRIGGERABLE MONOSTABLE MULTIVIBRATORS

- D-C Triggered from Active-High or Active-Low Gated Logic Inputs
- Retriggerable for Very Long Output Pulses, Up to 100% Duty Cycle
- Overriding Clear Terminates Output Pulse
- Compensated for V_{CC} and Temperature Variations
- '122, 'L122, 'LS122 Have Internal Timing Resistors

SN54122, SN54LS122 . . . J OR W
SN54L122 . . . J OR T
SN74122, SN74L122, SN74LS122 . . . J OR N
(TOP VIEW) (SEE NOTES 1 THRU 4)



logic: see function table

NC—No internal connection.

'122, 'L122, 'LS122
FUNCTION TABLE

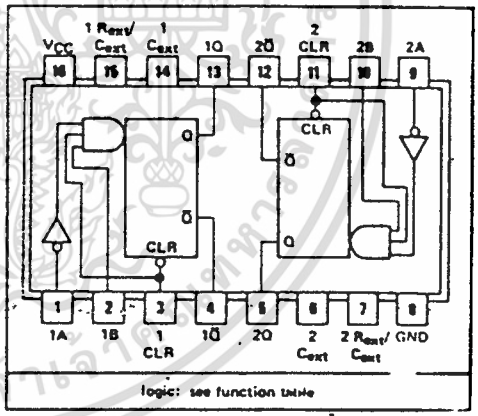
INPUTS					OUTPUTS	
CLEAR	A1	A2	B1	B2	Q	$\overline{Q}$
L	X	X	X	X	L	H
X	H	H	X	X	L	H
X	X	X	L	X	L	H
X	X	X	X	L	L	H
H	L	X	↑	H	⎍	⎎
H	L	X	H	↑	⎍	⎎
H	X	L	↑	H	⎍	⎎
H	X	L	H	↑	⎍	⎎
H	H	↓	H	H	⎍	⎎
H	↓	↓	H	H	⎍	⎎
H	↓	H	H	H	⎍	⎎
↑	L	X	H	H	⎍	⎎
↑	X	L	H	H	⎍	⎎

'123, 'L123, 'LS123
FUNCTION TABLE

INPUTS			OUTPUTS	
CLEAR	A	B	Q	$\overline{Q}$
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⎎
H	↓	H	⎍	⎎
↑	L	H	⎍	⎎

See explanation of function tables on page 3-8.

SN54123, SN54LS123 . . . J OR W
SN54L123 . . . J
SN74123, SN74L123, SN74LS123 . . . J OR N
(TOP VIEW) (SEE NOTES 1 THRU 4)



logic: see function table

description

These d-c triggered multivibrators feature output pulse width control by three methods. The basic pulse time is programmed by selection of external resistance and capacitance values (see typical application data). The '122, 'L122, and 'LS122 have internal timing resistors that allow the circuits to be used with only an external capacitor, if so desired. Once triggered, the basic pulse width may be extended by retriggering the gated low-level-active (A) or high-level-active (B) inputs, or be reduced by use of the overriding clear. Figure 1 illustrates pulse control by retriggering and early clear.

The 'LS122 and 'LS123 are provided enough Schmitt hysteresis to ensure jitter-free triggering from the B input with transition rates as slow as 0.1 millivolt per nanosecond.

NOTES: 1. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).
2. To use the internal timing resistor of '122, 'L122 or 'LS122, connect R_{int} to V_{CC}.
3. For improved pulse width accuracy and repeatability, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
4. To obtain variable pulse widths, connect an external variable resistance between R_{int} or R_{ext}/C_{ext} and V_{CC}.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA FOR '122, '123, 'L122, 'L123

For pulse widths when $C_{ext} < 1000$ pF, See Figures 4 and 5.

The output pulse is primarily a function of the external capacitor and resistor. For $C_{ext} > 1000$ pF, the output pulse width (t_w) is defined as:

$$t_w = K \cdot R_T \cdot C_{ext} \left(1 + \frac{0.7}{R_T}\right)$$

where

K is 0.32 for '122, 0.28 for '123,
0.37 for 'L122, 0.33 for 'L123

R_T is in kΩ (internal or external timing resistance.

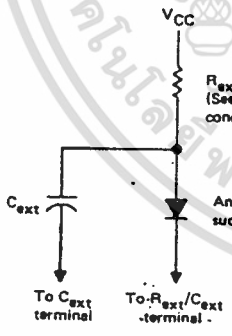
C_{ext} is in pF

t_w is in nanoseconds

To prevent reverse voltage across C_{ext}, it is recommended that the method shown in Figure 2 be employed when using electrolytic capacitors and in applications utilizing the clear function. In all applications using the diode, the pulse width is:

$$t_w = K_D \cdot R_T \cdot C_{ext} \left(1 + \frac{0.7}{R_T}\right)$$

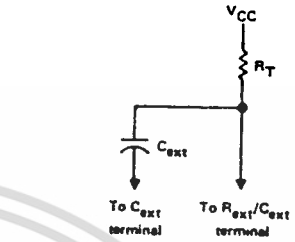
K_D is 0.28 for '122, 0.25 for '123,
0.33 for 'L122, 0.29 for 'L123



TIMING COMPONENT CONNECTIONS WHEN $C_{ext} > 1000$ pF AND CLEAR IS USED

FIGURE 2

Applications requiring more precise pulse widths (up to 28 seconds) and not requiring the clear feature can best be satisfied with the '121 or 'L121.



TIMING COMPONENT CONNECTIONS

FIGURE 3

'122, '123
TYPICAL OUTPUT PULSE WIDTH
vs
EXTERNAL TIMING CAPACITANCE

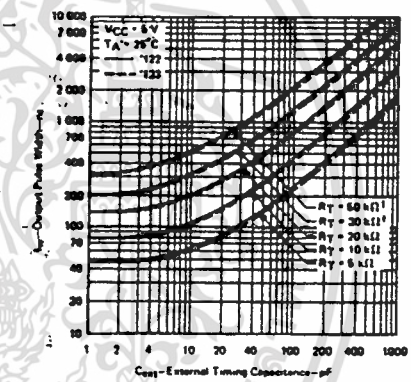


FIGURE 4

'L122
TYPICAL OUTPUT PULSE WIDTH
vs
EXTERNAL TIMING CAPACITANCE

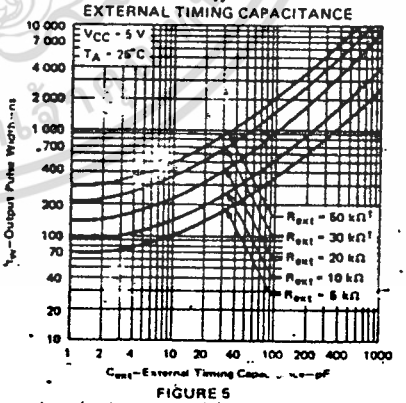


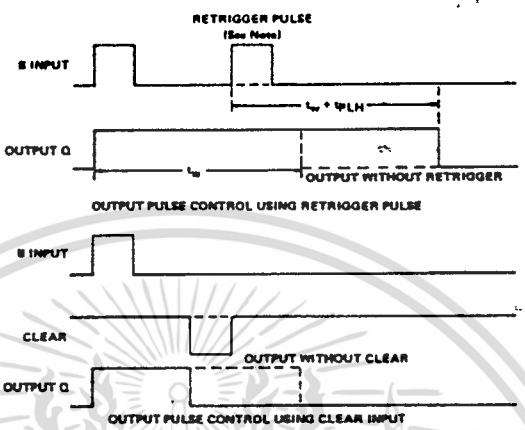
FIGURE 5

† These values of resistance exceed the maximum recommended for use over the full temperature range of the SN54' and SN54L' circuits

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

description (continued)

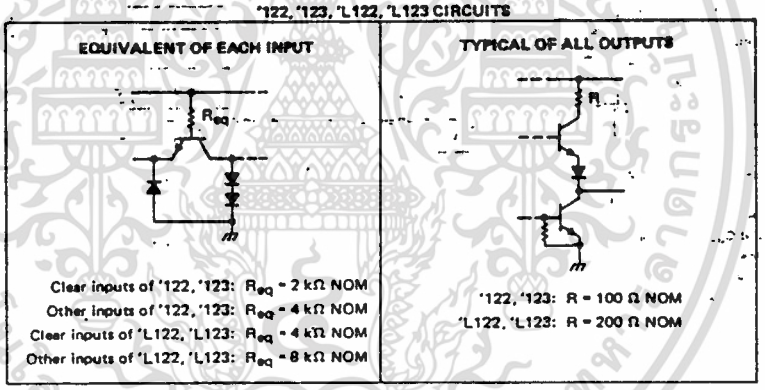


NOTE: Retrigger pulse must not start before 0.22 C_{ext} (in picofarads) nanoseconds after previous trigger pulse.

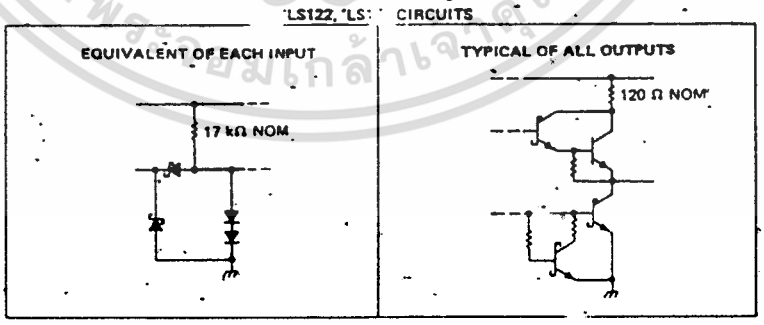
FIGURE 1–TYPICAL INPUT/OUTPUT PULSES

schematics of inputs and outputs

'122, '123, 'L122, 'L123 CIRCUITS



'LS122, 'LS123 CIRCUITS



TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA FOR 'LS122, 'LS123

The basic output pulse width is essentially determined by the values of external capacitance and timing resistance. For pulse widths when $C_{ext} \leq 1000$ pF, see Figure 7.

When $C_{ext} > 1000$ pF, the output pulse width is defined as:

$$t_w = 0.45 \cdot R_T \cdot C_{ext}$$

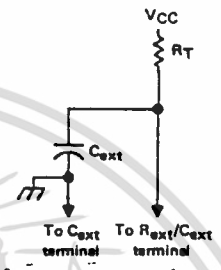
where

R_T is in kΩ (internal or external timing resistance.)

C_{ext} is in pF

t_w is in nanoseconds

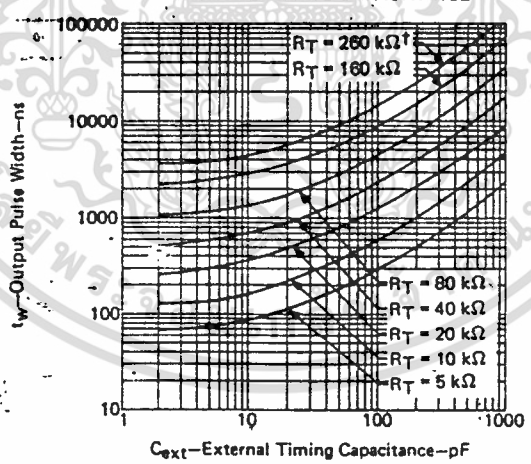
For best results, system ground should be applied to the C_{ext} terminal. The switching diode is not needed for electrolytic capacitance applications.



TIMING COMPONENT CONNECTIONS

FIGURE 6

'LS122, 'LS123
TYPICAL OUTPUT PULSE WIDTH
vs
EXTERNAL TIMING CAPACITANCE



† This value of resistance exceeds the maximum recommended for use over the full temperature range of the SN54LS circuits.

FIGURE 7

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 9012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL
MSI

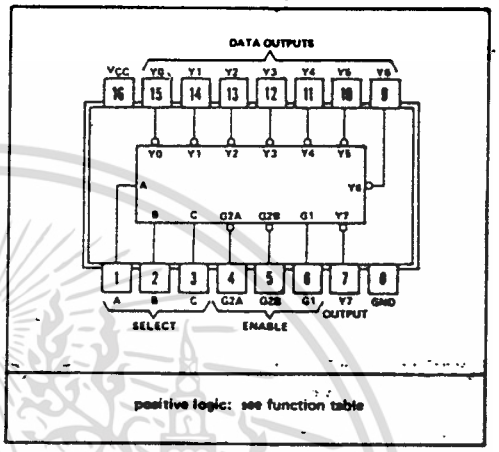
TYPES SN54LS138, SN54LS139, SN54S138, SN54S139, SN74LS138, SN74LS139, SN74S138, SN74S139 DECODERS/DEMULTIPLEXERS

BULLETIN NO. DL-S 7611804, DECEMBER 1972—REVISED OCTOBER 1976

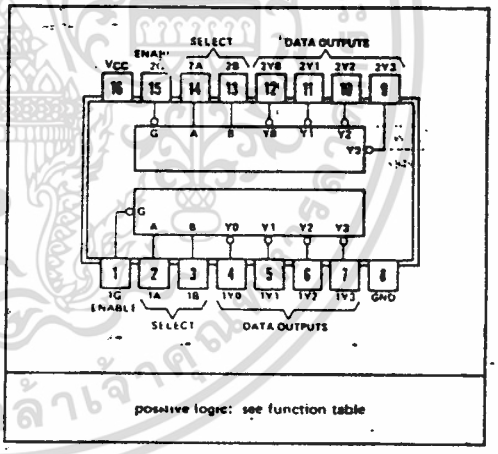
- Designed Specifically for High-Speed:
  Memory Decoders
  Data Transmission Systems
- 'S138 and 'LS138 3-to-8-Line Decoders Incorporate 3 Enable Inputs to Simplify Cascading and/or Data Reception
- 'S139 and 'LS139 Contain Two Fully Independent 2-to-4-Line Decoders/ Demultiplexers
- Schottky Clamped for High Performance

TYPE	TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
'LS138	22 ns	32 mW
'S138	8 ns	245 mW
'LS139	22 ns	34 mW
'S139	7.5 ns	300 mW

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . J OR N PACKAGE
(TOP VIEW)



SN54LS139, SN54S139 . . . J OR W PACKAGE
SN74LS139, SN74S139 . . . J OR N PACKAGE
(TOP VIEW)



description

These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast-enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'LS138 and 'S138 decode one-of-eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The 'LS139 and 'S139 comprise two individual two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

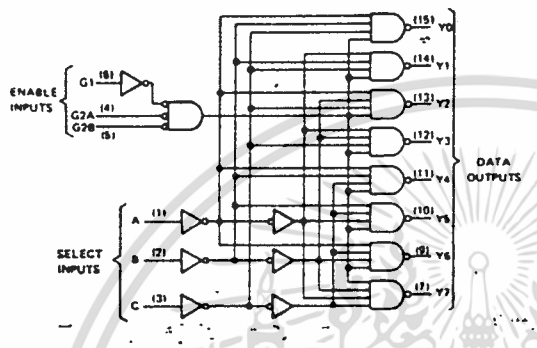
All of these decoders/demultiplexers feature fully buffered inputs each of which represents only one normalized Series 54LS/74LS load ('LS138, 'LS139) or one normalized Series 54S/74S load ('S138, 'S139) to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design. Series 54LS and 54S devices are characterized for operation over the full military temperature range of −55°C to 125°C; Series 74LS and 74S devices are characterized for 0°C to 70°C industrial systems.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

functional block diagrams and logic

'LS138, 'S138



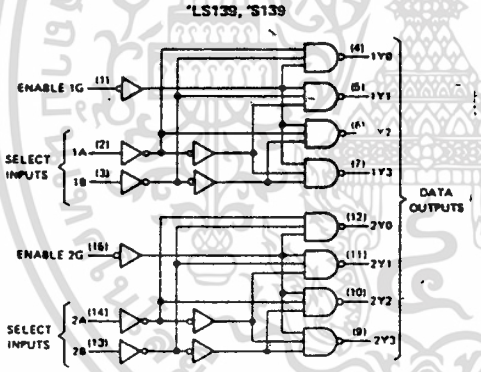
'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
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

*G2 = G2A + G2B

H = high level, L = low level, X = irrelevant

'LS139, 'S139

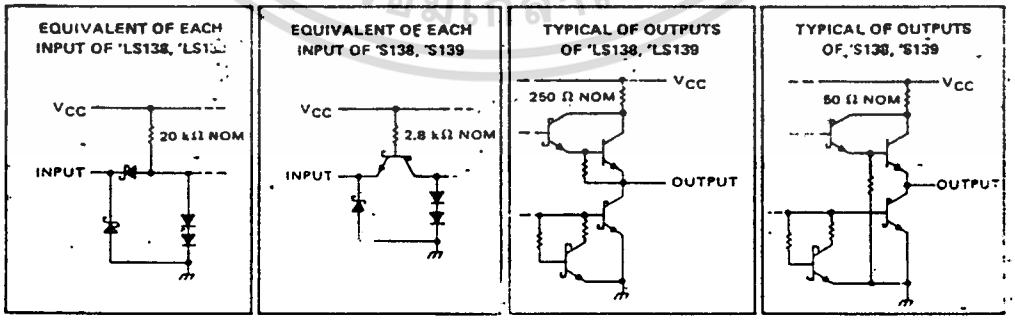


'LS139, 'S139
(EACH DECODER/DEMULTIPLEXER)
FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = high level, L = low level, X = irrelevant

schematics of inputs and outputs



TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

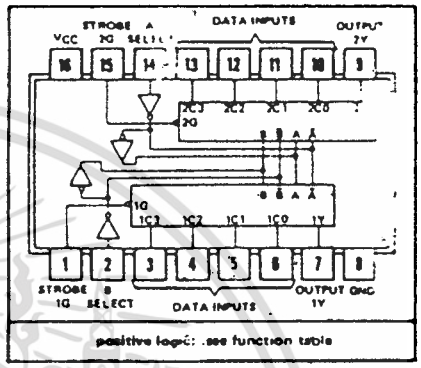
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL
MSI

TYPES SN54153, SN54L153, SN54LS153, SN54S153, SN74153, SN74L153, SN74LS153, SN74S153
DUAL 4-LINE-TO-1-LINE DATA SELECTORS/MULTIPLEXERS

BULLETIN NO. DL-S 7611862, DECEMBER 1972 – REVISED OCTOBER 1976

SN54153, SN54LS153, SN54S153 . . . J OR W PACKAGE
SN54L153 . . . J PACKAGE
SN74153, SN74L153, SN74LS153, SN74S153 . . . J OR N PACKAGE
(TOP VIEW)



- Permits Multiplexing from N lines to 1 line
- Performs Parallel-to-Serial Conversion
- Strobe (Enable) Line Provided for Cascading (N lines to n lines)
- High-Fan-Out, Low-Impedance, Totem-Pole Outputs
- Fully Compatible with most TTL and DTL Circuits

TYPE	TYPICAL AVERAGE PROPAGATION DELAY TIMES FROM DATA	FROM STROBE	FROM SELECT	TYPICAL POWER DISSIPATION
'153	14 ns	17 ns	22 ns	180 mW
'L153	27 ns	34 ns	44 ns	90 mW
'LS153	14 ns	19 ns	22 ns	31 mW
'S153	6 ns	8.5 ns	12 ns	225 mW

description

Each of these monolithic, data selectors/multiplexers contains inverters and drivers to supply fully complementary, on-chip, binary decoding data selection to the AND-OR-invert gates. Separate strobe inputs are provided for each of the two four-line sections.

FUNCTION TABLE

SELECT INPUTS		DATA INPUTS				STROBE	OUTPUT
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

Select inputs A and B are common to both sections.
H = high level, L = low level, X = irrelevant

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage: '153, 'L153, 'S153 . . . . . 5.5 V
'LS153 . . . . . 7 V
Operating free-air temperature range: SN54', SN54L', SN54LS', SN54S' Circuits . . . . . −55°C to 125°C
SN74', SN74L', SN74LS', SN74S' Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 12 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL MSI

TYPES SN54160 THRU SN54163, SN54LS160A THRU SN54LS163A, SN54S162, SN54S163, SN74160 THRU SN74163, SN74LS160A THRU SN74LS163A, SN74S162, SN74S163 SYNCHRONOUS 4-BIT COUNTERS

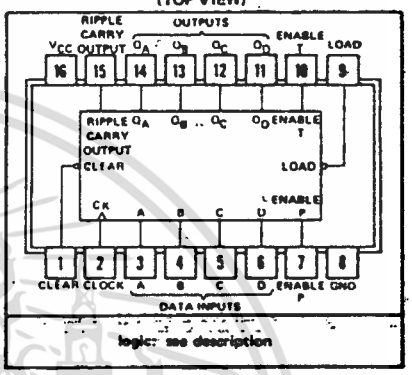
BULLETIN NO. DL-S 7711385, OCTOBER 1976—REVISED AUGUST 1977

'160, '161, 'LS160A, 'LS161A . . . SYNCHRONOUS COUNTERS WITH DIRECT CLEAR
'162, '163, 'LS162A, 'LS163A, 'S162, 'S163 . . . FULLY SYNCHRONOUS COUNTERS

- Internal Look-Ahead for Fast Counting
- Carry Output for n-Bit Cascading
- Synchronous Counting
- Synchronously Programmable
- Load Control Line
- Diode-Clamped Inputs

TYPE	TYPICAL PROPAGATION TIME, CLOCK TO Q OUTPUT	TYPICAL MAXIMUM CLOCK FREQUENCY	TYPICAL POWER DISSIPATION
'160 thru '163	14 ns	32 MHz	305 mW
'LS160A thru 'LS163A	14 ns	32 MHz	93 mW
'S162 and 'S163	9 ns	70 MHz	475 mW

SERIES 54', 54LS', 54S' . . . J OR W PACKAGE
SERIES 74', 74LS', 74S' . . . J OR N PACKAGE
(TOP VIEW)



description

These synchronous, presettable counters feature an internal carry look-ahead for application in high-speed counting designs. The '160, '162, 'LS160A, 'LS162A, and 'S162 are decade counters and the '161, '163, 'LS161A, 'LS163A, and 'S163 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes that are normally associated with asynchronous (ripple clock) counters. A buffered clock input triggers the four flip-flops on the rising (positive-going) edge of the clock input waveform.

These counters are fully programmable; that is, the outputs may be preset to either level. As presetting is synchronous, setting up a low level at the load input disables the counter and causes the outputs to agree with the setup data after the next clock pulse regardless of the levels of the enable inputs. Low-to-high transitions at the load input of the '160 thru '163 should be avoided when the clock is low if the enable inputs are high at or before the transition. This restriction is not applicable to the 'LS160A thru 'LS163A or 'S162 or 'S163 . The clear function for the '160, '161, 'LS160A, and 'LS161A is asynchronous and a low level at the clear input sets all four of the flip-flop outputs low regardless of the levels of clock, load, or enable inputs. The clear function for the '162, '163, 'LS162A, 'LS163A, 'S162, and 'S163 is synchronous and a low level at the clear input sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily as decoding the maximum count desired can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to 0000 (LLLL). Low-to-high transitions at the clear input of the '162 and '163 should be avoided when the clock is low if the enable and load inputs are high at or before the transition.

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. Instrumental in accomplishing this function are two count-enable inputs and a ripple-carry output. Both count-enable inputs (P and T) must be high to count, and input T is fed forward to enable the ripple carry output. The ripple carry output thus enabled will produce a high-level output pulse with a duration approximately equal to the high-level portion of the Q_A output. This high-level overflow ripple carry pulse can be used to enable successive cascaded stages. High-to-low-level transitions at the enable P or T inputs of the '160 thru '163 should occur only when the clock input is high. Transitions at the enable P or T inputs of the 'LS160A thru 'LS163A or 'S162 and 'S163 are allowed regardless of the level of the clock input.

'LS160A thru 'LS163A, 'S162 and 'S163 feature a fully independent clock circuit. Changes at control inputs (enable P or T, or clear) that will modify the operating mode have no effect until clocking occurs. The function of the counter (whether enabled, disabled, loading, or counting) will be dictated solely by the conditions meeting the stable setup and hold times.

The 'LS160A thru 'LS163A are completely new designs. Compared to the original 'LS160 thru 'LS163, they feature 0-nanosecond minimum hold time and reduced input currents I_{IH} and I_{IL}.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

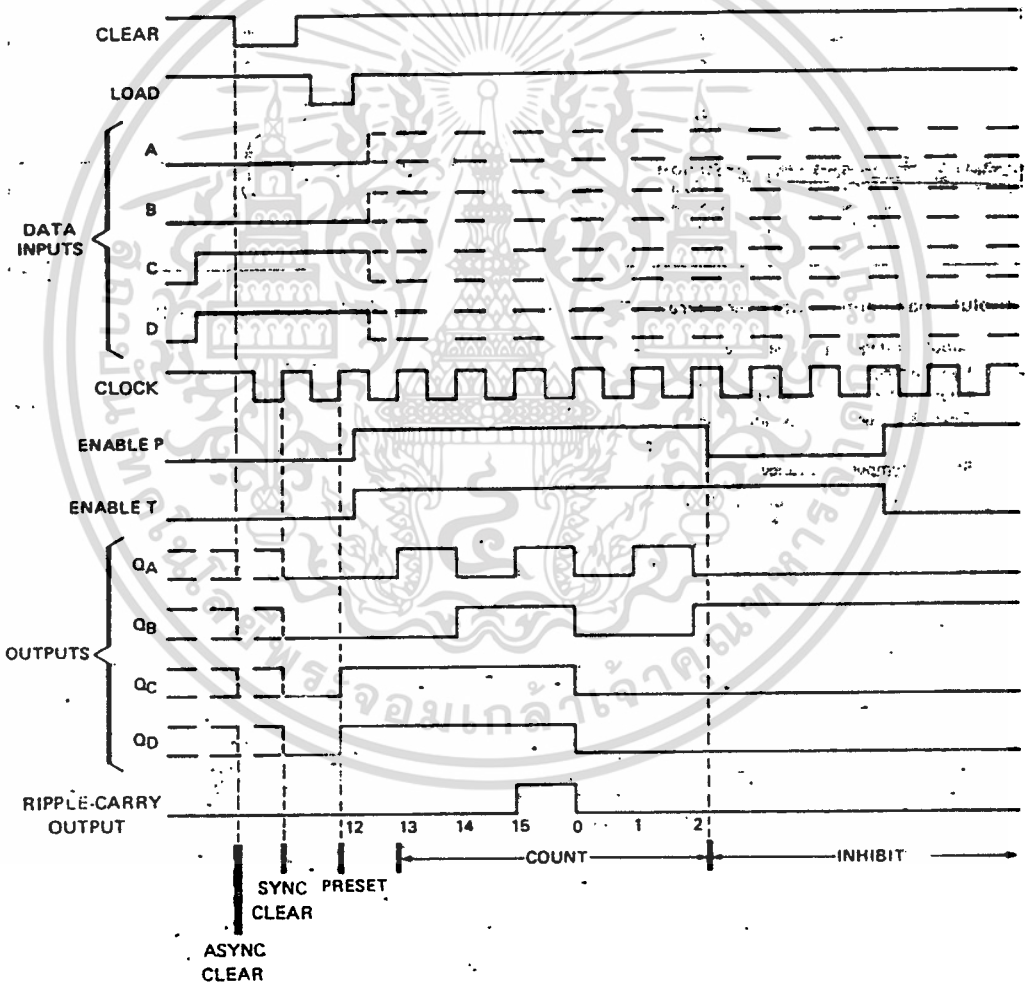
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'161, 'LS161A, '163, 'LS163A, 'S163 BINARY COUNTERS

typical clear, preset, count, and inhibit sequences

Illustrated below is the following sequence:

1. Clear outputs to zero ('161 and 'LS161A are asynchronous; '163, 'LS163A, and 'S163 are synchronous)
2. Preset to binary twelve
3. Count to thirteen, fourteen fifteen, zero, one, and two
4. Inhibit



TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

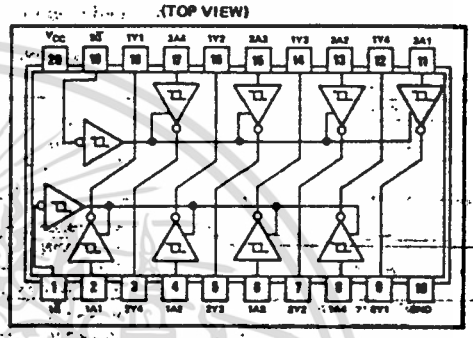
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

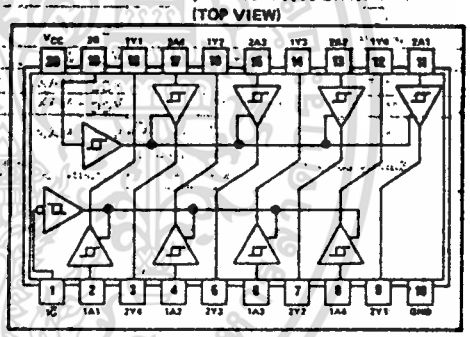
	Typical I_{OL} (Sink Current)	Typical I_{OH} (Source Current)	Typical Propagation Delay Times		Typical Enable/ Disable Times	Typical Power Dissipation (Enabled)	
			Inverting	Noninverting		Inverting	Noninverting
SN54LS'	12 mA	−12 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN74LS'	24 mA	−15 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN54S'	48 mA	−12 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW
SN74S'	64 mA	−15 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

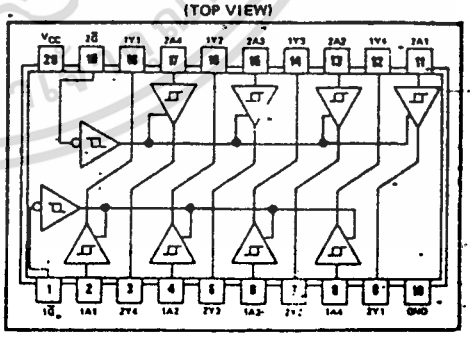
SN64LS240, SN54S240 . . . J
SN74LS240, SN74S240 . . . J OR N
(TOP VIEW)



SN54LS241, SN54S241 . . . J
SN74LS241, SN74S241 . . . J OR N
(TOP VIEW)



SN54LS244 . . . J
SN74LS244 . . . J OR N
(TOP VIEW)

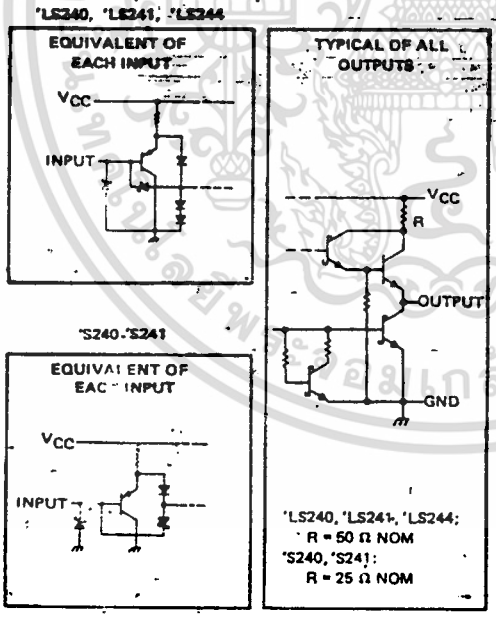


description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical $\overline{G}$ (active-low output control) inputs, and complementary G and $\overline{G}$ inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

schematics of inputs and outputs

'LS240, 'LS241, 'LS244

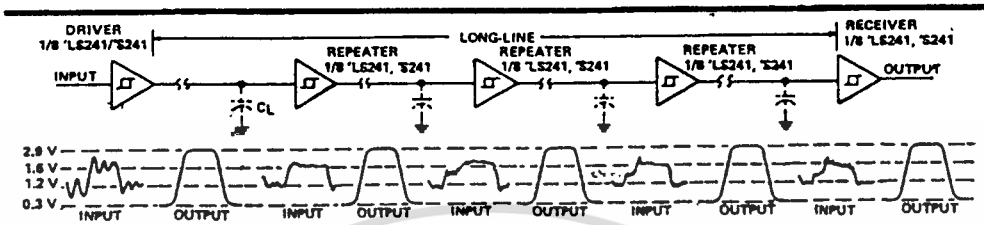


'S240, 'S241

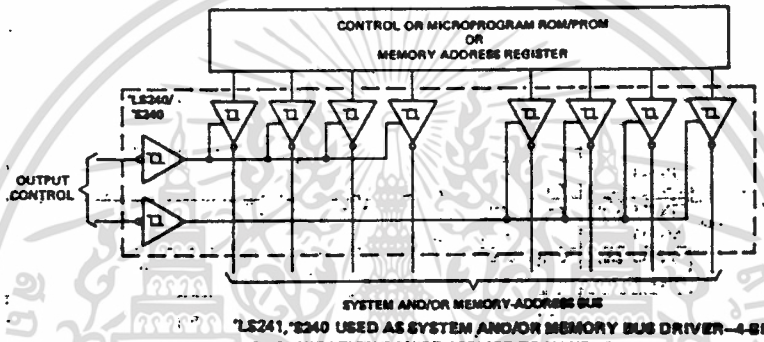
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

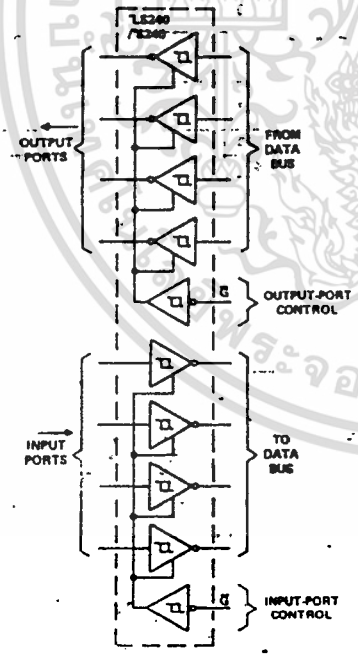
TYPES SN54LS240,SN54LS241, SN54LS244,SN54S240,SN54S241,SN74LS240, SN74LS241,SN74LS244,SN74S240,SN74S241
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS



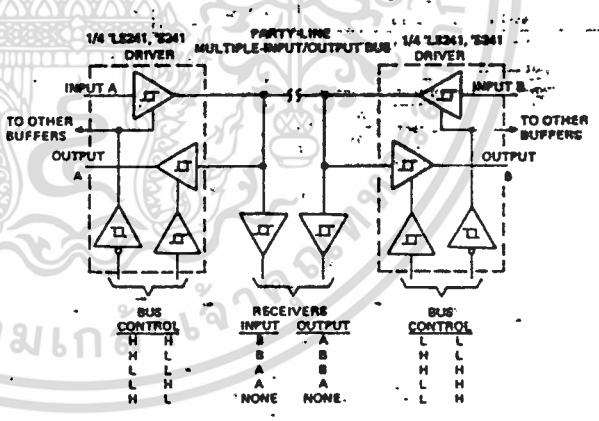
'LS241, 'S241 USED AS REPEATER/LEVEL RESTORER



'LS241, 'S240 USED AS SYSTEM AND/OR MEMORY BUS DRIVER—4-BIT ORGANIZATION CAN BE APPLIED TO HANDLE BINARY OR BCD



INDEPENDENT 4-BIT BUS DRIVERS/RECEIVERS IN A SINGLE PACKAGE



PARTY-LINE BUS SYSTEM WITH MULTIPLE INPUTS, OUTPUTS, AND RECEIVERS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

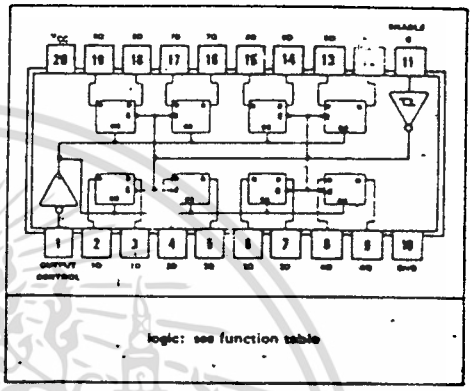
TTL
MSI

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

BULLETIN NO. DL-S 7712360, OCTOBER 1976—REVISED A... IST 1977

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)
- SN54LS363 and SN74LS364 Are Similar But Have Higher V_{OH} For MOS Interface

SN54LS373, SN54S373 . . . J PACKAGE
SN74LS373, SN74S373 . . . J OR N PACKAGE
(TOP VIEW)

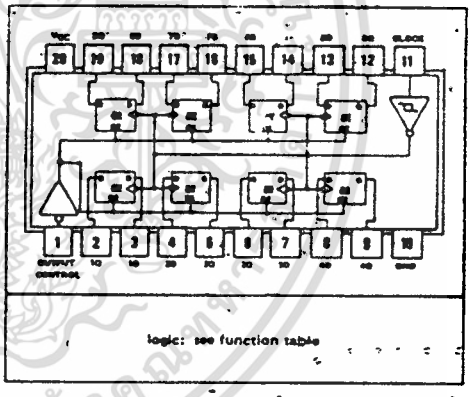


logic: see function table

'LS373, 'S373
FUNCTION TABLE

OUTPUT CONTROL	ENABLE G	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS374, SN54S374 . . . J PACKAGE
SN74LS374, SN74S374 . . . J OR N PACKAGE
(TOP VIEW)



logic: see function table

'LS374, 'S374
FUNCTION TABLE

OUTPUT CONTROL	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

See explanation of function tables on page 3-8.

description

These 8-bit registers feature totem-pole three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was setup.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้