FREQUENCY COUNTER

INTERFACE WITH COMPUTER

(เครื่องวัดสัญญาณความถี่ ควบคุมและแสดงผลโดยคอมพิวเตอร์)

คณะผู้จัดทำ

นาย ทวีเกียรติ ตรีธารทิพย์ รหัส 34103127

นาย ธีรยสต์ อุดมมณีธนกิจ รหัส 34103157

อาจารย์ที่ปรึกษา

ผศ. นิภา ลีลารุจิ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมโทรคมนาคม

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **FREQUENCY COUNTER INTERFACE WITH COMPUTER**

(เครื่องวัดสัญญาณความถี่ ควบคุมและแสดงผลโดยคอมพิวเตอร์)

คณะผู้จัดทำ

นาย ทวีเกียรติ ตรีธารทิพย์ รหัส 34103127

นาย ธีรยสถ์ อุดมมณีธนกิจ รหัส 34103157

(ผศ. นิภา ลีลารุจิ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า รายงานฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจจะศึกษาหลักการของเครื่องวัดความถี่ที่เชื่อมต่อกับคอมพิวเตอร์ ตลอดจนพัฒนาเครื่องวัดความถี่ให้สามารถวัดความถี่ได้ในช่วงที่กว้างมากยิ่งขึ้น เพื่อที่จะได้เครื่องวัดความถี่ที่มีประสิทธิภาพ และมีราคาไม่แพง ไว้ใช้ในการปฏิบัติงานทาง ด้านโทรคมนาคม โดยไม่จำเป็นต้องพึ่งเครื่องวัดจากต่างประเทศอีกต่อไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	1
บทนำ	3
บทที่ 1 เทคนิคเครื่องวัดความถี่	5
บทที่ 2 หลักการและการทำงานของ FREQUENCY COUNTER INTERFACE WITH COMPUTER	10
ผลการทดลอง	
ผลการทดลองการวัดความถี่	92
ผลการทดลองการตอบสนองความถี่ของวงจรขยายสัญญาณ	94
ตัวอย่างข้อมูลความถี่ที่ถูกพิมพ์ออกมา	96
สรุป	97
แนวทางการพัฒนาต่อไป	98
หนังสืออ้างอิง	99
กิตติกรรมประกาศ	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดความถี่ ควบคุมและแสดงผล โดยคอมพิวเตอร์

FREQUENCY COUNTER INTERFACE WITH COMPUTER

โดย นาย ทวีเกียรติ ตรีธารทิพย์
นาย ธีรยสถ์ อุดมมณีธนกิจ
อาจารย์ที่ปรึกษา อ. นิภา ลีลารุจิ

บทคัดย่อ

ในการปฏิบัติงานทางโทรคมนาคมที่เกี่ยวข้องกับสัญญาณโดยทั่วไป จำเป็นอย่างยิ่งที่จะต้องวัดความถี่ และ คาบของสัญญาณ เพื่อนำไปวิเคราะห์ หรือ คำนวณ ดังนั้น FREQUENCY COUNTER จึงเป็นเครื่องมือวัดที่มีความจำเป็น แต่ว่ามีราคาแพงเมื่อนำเข้าจากต่างประเทศ

ยิ่งในปัจจุบันมีการนำคอมพิวเตอร์เข้ามาช่วยงานต่างๆมากมายหลายสาขา ดังนั้นทางสาขาโทรคมนาคมก็นำคอมพิวเตอร์มาประยุกต์ในการใช้ได้ด้วย ซึ่งทางผู้จัดทำได้เล็งเห็นถึงแนวโน้มในอนาคตว่าจะมีบทบาทมากทางการสื่อสาร จึงได้ทดลองผลิต FREQUENCY COUNTER ที่สามารถแสดงผลที่วัดมาที่หน้าจอคอมพิวเตอร์ได้ โดยวิธีการนี้ทำให้เราสามารถเก็บข้อมูล , นำข้อมูลมาวาดกราฟได้ ดังนั้นลักษณะรายละเอียดวงจรจึงมีอุปกรณ์ในส่วนต่างๆหลายส่วน ซึ่งแต่ละส่วนทำหน้าที่ต่างๆกัน โดยจะกล่าวถึงในส่วนของเนื้อหาต่อไป

เนื้อหาในรายงานเล่มนี้ จะกล่าวถึงหลักการ ตลอดจนวงจร รวมไปถึงวิธีการสร้าง FREQUENCY COUNTER INTERFACE WITH COMPUTER

ABSTRACT

IN PRACTICE , TELECOMMUNICATION JOB THAT INVOLVED SIGNAL PROCESSING IS NECESSARY FOR MEASURING FREQUENCY AND PERIOD IN ORDER TO ANALYZE AND CALCULATE THEM . FREQUENCY COUNTER IS AN ESSENTIAL TOOL FOR THESE BUT IT IS VERY EXPENSIVE TO IMPORT IT FROM ABOARD .

AT PRESENT , COMPUTERS HAVE BEEN USED TO HELP IN MANY KINDS OF JOBS . THEREFORE , THEY HAVE BEEN APPLIED TO USE IN THIS PROJECT AS WELL . WE BELIEVE THAT IN THE FUTURE THEIR ROLES WILL HAVE AN IMPORTANT EFFECT IN COMMUNICATION .

THIS PROJECT IS TO PRODUCE FREQUENCY COUNTER THAT CAN SHOW THE RESULT ON THE SCREEN OF COMPUTER . THE METHOD USED IS STORING DATA AND CONVERTING IT TO GRAPH . FOR THE CIRCUIT, THERE ARE MANY PARTS WHICH EACH PART HAS ITS OWN FUNCTIONS.

THIS REPORT WILL DESCRIBE THE CONCEPT THAT USED, THE DETAIL OF THE CIRCUIT AND THE METHOD OF PRODUCING FREQUENCY COUNTER THAT INTERFACE WITH COMPUTER .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของวงจร

FREQUENCY COUNTER INTERFACE WITH COMPUTER

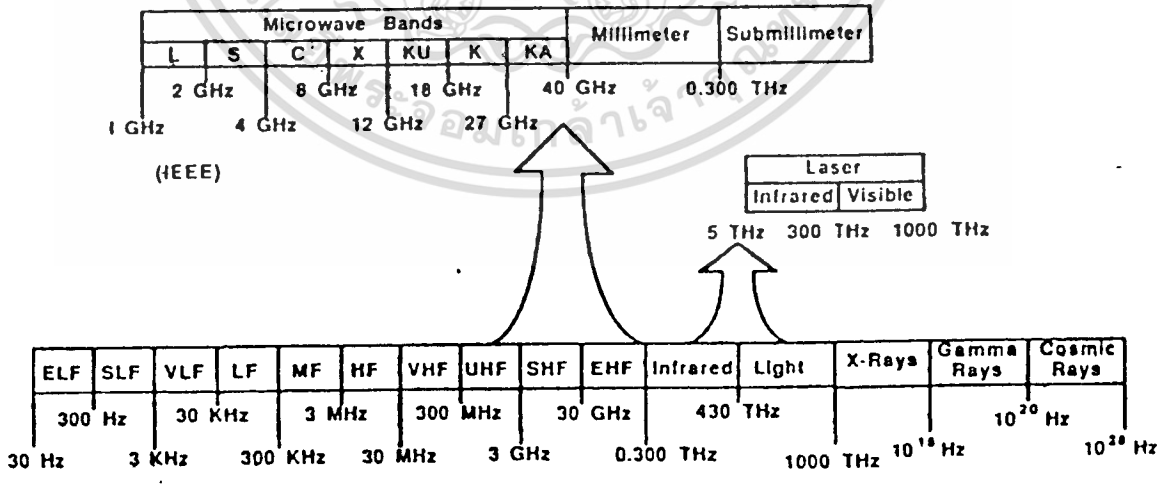
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

อัลวินทอฟฟ์เลอร์ (Alvin Tofler)ได้กล่าวไว้ในหนังสือคลื่นลูกที่สาม (The Third Wave) ว่าในอนาคตพนักงานบริษัทอาจจะไม่ต้องไปทำงานที่บริษัทหรือโรงงานก็ได้ แต่สามารถทำงานที่บ้านแล้วส่งข้อมูลนั้นผ่านระบบสื่อสารโทรคมนาคม ส่งผลให้การดำรงชีวิตของมนุษย์เปลี่ยนแปลงไปมาก ตราบเท่าที่เทคโนโลยีทางการสื่อสารโทรคมนาคมยังรุดหน้าต่อไปเรื่อยๆ ซึ่งคำกล่าวที่ว่านี้ดูเหมือนจะไม่ใช่สิ่งเพ้อฝันอีกต่อไป

แถบสเปคตรัมความถี่

แถบสเปคตรัมของความถี่เริ่มต้นตั้งแต่ 30 Hz ในย่านความถี่ต่ำกว่าเสียง ขึ้นไปจนถึง 1×10^{28} Hz ในย่านความถี่รังสีคอสมิค ซึ่งจะได้แสดงให้เห็นการจัดแถบสเปคตรัมความถี่ และ ชื่อประกอบการเรียกแถบความถี่ในแต่ละย่าน ดังรูป



รูปแสดงแถบสเปคตรัมของความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถบความถี่	ชื่อที่เรียกแถบความถี่
3-30 kHz	VLF (Very Low Frequencies)
30-300 kHz	LF (Low Frequencies)
300 kHz-3 MHz	MF (Medium Frequencies)
3-30 MHz	HF (High Frequencies)
30-300 MHz	VHF (Very High Frequencies)
300 MHz-3 GHz	UHF (Ultra High Frequencies)
3-30 GHz	SHF (Super High Frequencies)
30-300 GHz	EHF (Extremely High Frequencies)

การเรียกชื่อแถบความถี่ย่านไมโครเวฟของ I.E.E.E.

แถบความถี่	ย่านความถี่, GHz
HF	0.003-0.030
VHF	0.030-0.300
UHF	0.300-1.00
L	1.00-2.00
S	2.00-4.00
C	4.00-8.00
X	8.00-12.0
Ku	12.0-18.0
K	18.0-27.0
Ka	27.0-40.0
Millimeter	40.0-300.0
Submillimeter	มากกว่า 300

ตารางแสดงการเรียกชื่อแถบย่านความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
เทคนิคเครื่องวัดความถี่

อดีตของเครื่องวัดความถี่เริ่มแรกประกอบขึ้นจากบอร์ดที่เต็มไปด้วย IC ประกอบด้วย TTL และส่วน input จะเป็นวงจรประเภท analog ซึ่งบริษัทผู้ผลิตก็จะใช้อุปกรณ์ประเภทเดียวกัน ซึ่งความแตกต่างในสมัยนั้นจะมีเพียง ความสวยงามของหน้าปัทม์ , ปุ่มกด และความละเอียดของส่วนแสดงผล ซึ่งเป็น LED อาจจะมี 6,7 หรือ 8 หลัก กล่าวคือ การแข่งขันทางเทคโนโลยีในสมัยนั้นยังไม่ดีเท่าที่ควร จนมาในปี พ.ศ. 2513 จากที่เราต้องนำ IC(TTL) ซึ่งเป็น logic gate มาต่อกันเป็นวงจรนับ ซึ่งอาจจะต้องใช้ทฤษฎีของ QM(Quine Mcluskey) หรือ Karnaugh Maps ซึ่งค่อนข้างยุ่งยาก มาเป็นวงจรรวมซึ่งมีมาตรฐานของวงจรนับความถี่ หรือเป็นการย่อขนาดของวงจรให้เล็กลง และเป็นวงจรอยู่ใน IC ตัวเดียว หรือ เรียกว่า Small Scale Integrator และถัดจากนี้ก็เป็นยุค Microprocessor ทำให้เทคโนโลยีทางด้านเครื่องวัดความถี่ก้าวหน้าขึ้นมาก และผลจากการก้าวหน้าทำให้กำลังในการผลิตอุปกรณ์จำพวกนี้เพิ่มมากขึ้น เพราะขั้นตอนในการผลิตทำได้ง่ายกว่าสมัยก่อนมาก ได้มีการจำแนกเครื่องวัดความถี่ออกเป็น 3 ประเภทด้วยกัน คือ

1.เครื่องวัดความถี่อย่างง่าย (Basic frequency measurement)

2.เครื่องวัดความถี่และเวลา (Time and frequency measurement)

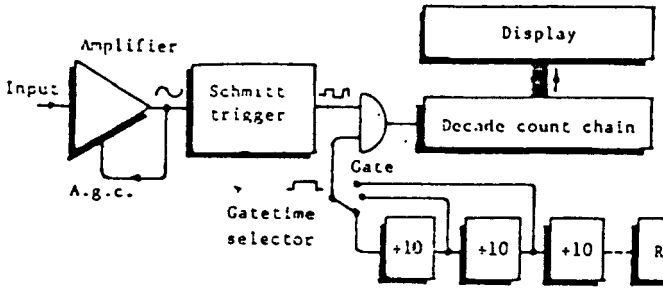
3.เครื่องวัดไมโครเวฟ (Microwave Counters)

คราวนี้มาดูการทำงานของเครื่องนับความถี่พื้นฐานโดยอาศัยรูปที่ 1.1 จากรูปจะเห็นในลักษณะของ Block Diagram โดยส่วนแรกจะเป็นส่วนที่รับ input ทำหน้าที่ขยายสัญญาณให้มีขนาดใหญ่ขึ้น ซึ่ง output ที่ออกจากภาคขยาย (Amplifier) จะต้องมีความถี่ตรงกับสัญญาณ input ที่ได้รับเข้ามา โดยจะเห็นได้ว่าเราจะมีการควบคุมด้วย A.G.C. (Automatic Gain Control) ทำให้ได้ประสิทธิภาพดีขึ้น

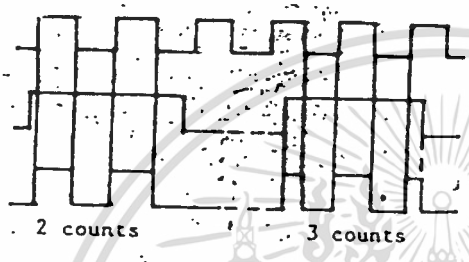
เมื่อผ่านภาคขยายแล้วสัญญาณจะมีขนาดพอที่จะหนีการรบกวนจากสัญญาณรบกวน แล้วสัญญาณจะถูกส่งเข้าภาคเปลี่ยนสัญญาณรูปสี่เหลี่ยมจตุรัส (Square Wave) ซึ่งภาคนี้ประกอบด้วยวงจรชมิททริกเกอร์โดยการเปลี่ยนสัญญาณนั้นความถี่ต้องเท่าเดิมหรือผิดพลาดไปเพียงเล็กน้อยเท่านั้น ซึ่งการผิดพลาดนั้นเกิดจาก noise ที่ผสมมากับสัญญาณ input

หลังจากได้สัญญาณสี่เหลี่ยมจตุรัสที่มีความถี่เท่ากับ input แล้ว ก็มาถึงการนับโดยผ่านเกทซึ่งเป็น AND GATE ดังรูป 1.1 โดยจะมีสัญญาณสี่เหลี่ยมที่เป็นตัวอ้างอิงอยู่ซึ่งสร้างจากตัวออสซิลเลเตอร์ และมีวงจรหาร 10 อยู่ ซึ่งเราจะสามารถเลือกได้ว่าจะหาร

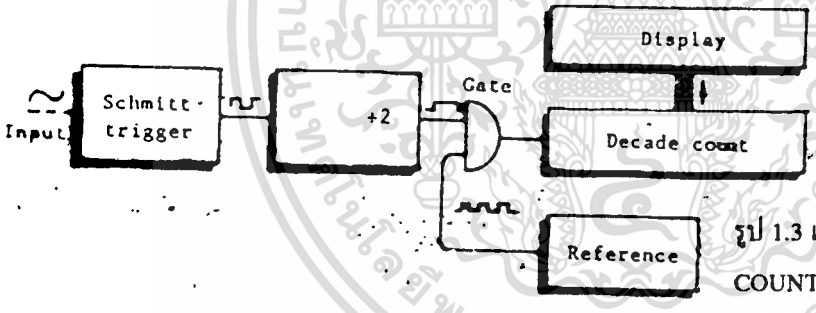
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



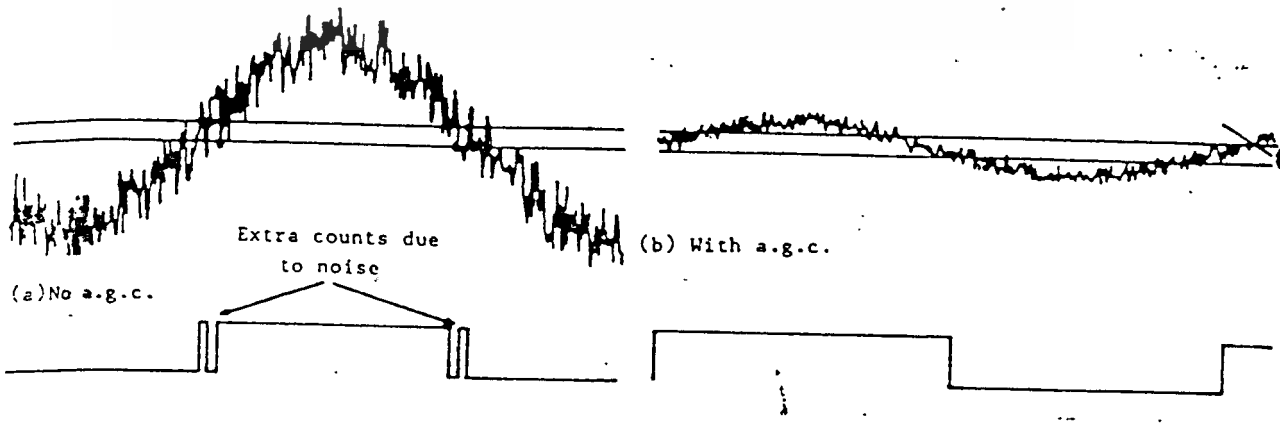
รูป 1.1 แสดง BLOCK DIAGRAM ของเครื่องนับความ
ขั้นพื้นฐาน ซึ่งสัญญาณอินพุท (INPUT SIGNAL) จะ
เปลี่ยนเป็นสัญญาณรูปสี่เหลี่ยม (SQUARE WAVE) ไ
ผ่านวงจรชมิททริกเกอร์ (SCHMITT TRIGGER) และ
ไปนับโดยเทียบผ่านทางเกท (GATE TIME) ความละเ
ของการนับมีโอกาสผิด ±1 รูปคลื่น



รูป 1.2 แสดงการคาดเคลื่อนของการนับ คือ สัญญาณอินพุ
กับสัญญาณที่ GATE TIME มีความสัมพันธ์ผิดพลาด คือ ช่
หนึ่งนับได้ 2 ลูกคลื่น แต่อีกช่วงหนึ่งได้ 1 ลูกคลื่น (นั่นคือ
นับผิดไป 1 ลูกคลื่น)



รูป 1.3 แสดงพื้นฐานเครื่องนับคาบเวลา (BASIC PERIOD
COUNTER) ซึ่งจะตรงข้ามกับรูป 1.1 คาบของสัญญาณอิน
จะครอบคลุมเกท (GATE TIME) ซึ่งมี CLOCK PULSE เ
แบบจำลองความละเอียด 1 สัญญาณ CLOCK PULSE



รูป 1.4 แสดงความแตกต่างของวงจรที่มี AGC กับ ไม่มี AGC
ซึ่งจะเห็นประสิทธิภาพของการทำงานที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

กี่ 10 ขึ้นอยู่กับตัว Gate time Selector และสังเกตได้ว่าความกว้างของพัลซ์จากหน่วยอ้างอิง จะครอบคลุมสัญญาณที่ได้มาจาก input ทำให้ input จะถูกจับนับเมื่อมีพัลซ์จากหน่วยอ้างอิงเท่านั้น ซึ่งทำให้เวลาที่พัลซ์ของหน่วยอ้างอิงเป็นศูนย์จะทำให้เกิดการแสดงผลหรือทำให้มันค้างเพื่ออ่านตัวเลขนั่นเอง ส่วนที่ทำหน้าที่นับคือ Decade Count Chain ซึ่งมันจะนับเป็นลูกโซ่ไป และส่งต่อไปแสดงผล (Display) ซึ่งอาจเป็น LED Display หรืออย่างอื่นก็ได้ จากวงจรนี้จะมีความผิดพลาดไม่เกิน ± 1 รูปคลื่น ดังในรูป 1.2 และประสิทธิภาพของการนับจะอยู่ที่การครอบคลุมของพัลซ์จาก Gate time Selector ซึ่งอาจทำให้วัดได้ละเอียดถึง 1Hz ทีเดียว

สำหรับเครื่องวัดความถี่ถ้าเราป้อนสัญญาณ input จากสัญญาณอ้างอิงซึ่งมีความถี่คงที่สิ่งที่เกิดขึ้น คือ ส่วนของการแสดงผลจะให้ค่าคงที่ค่าหนึ่งตลอด (เพราะความถี่คงที่) ส่วนเครื่องวัดเวลา (time measurement) ดังในรูป 1.3 ซึ่งในเครื่องวัดส่วนใหญ่สมควรจะวัดได้ทั้งเวลาและความถี่ แต่จะมีความยุ่งยากในการสร้าง และเป็นสิ่งที่เรารู้จักในชื่อ Universal Counter timers

1.1 ลักษณะทางอินพุท

ในภาคอินพุทของเครื่องนับความถี่จะรับสัญญาณแบบอนาลอกเข้ามา ถ่ายทอดออกมาทางเอาท์พุทได้เหมือนสัญญาณอินพุทโดยไม่มีความผิดเพี้ยน ดังนั้นภาคอินพุทจะต้องตอบสนองความถี่ได้กว้าง คือ อย่างน้อยต้องตอบสนองความถี่จาก 10 Hz ไปถึง 500 MHz แต่ถ้าออกแบบให้ตอบสนองได้มากกว่านี้จะดีมาก นอกจากนี้ยังต้องรับขนาดแรงดันของสัญญาณอินพุทได้ในช่วงน้อยกว่า 10 mV ไปจนถึง 250 V และสิ่งสุดท้ายที่ออกจากภาคนี้ คือขบวนพัลซ์ที่มีแอมพลิจูดคงที่ค่าหนึ่ง และความถี่ที่ไม่คลาดเคลื่อนจากสัญญาณที่รับเข้ามาทางอินพุท

ในปัจจุบันยังไม่มีวงจรใดที่สมบูรณ์แบบได้ดังที่กล่าว สิ่งที่ทำได้และทำอยู่ในขณะนี้ คือ เราแบ่งความถี่ออกเป็นช่วงๆ ซึ่งผลที่ออกมาคือ เราต้องสร้างวงจรตอบสนองความถี่ออกมาหลายชุด โดยใช้อุปกรณ์ที่แตกต่างกันไป เช่น IC จำพวก CMOS จะตอบสนองได้ดีในช่วง 10 HZ ถึง 10 MHz , IC จำพวก TTL และ ECL จะตอบสนองในช่วง 5 MHz ถึง 100 MHz และ สำหรับ IC จำพวก High speed ECL จะตอบสนองในช่วง 50 Hz ถึง 500 MHz เป็นต้น สิ่งเหล่านี้จะเป็นข้อมูลของนักออกแบบได้เป็นอย่างดี นอกจากสิ่งที่กล่าวนี้แล้วยังมีปัญหาการรบกวนของสัญญาณรบกวนที่ย่านความถี่สูง (High frequency signal) จะทำให้พัลซ์ที่ได้รับในภาคนี้ผิดพลาดไปจากความเป็นจริง ทำให้ผลการนับไม่ถูกต้อง เพราะฉะนั้นเราจะต้องมี Low pass Filters เพื่อช่วยขจัด noise ความถี่สูงๆ ได้

เนื่องจากการตอบสนองย่านไดนามิคค่อนข้างกว้าง เราต้องมีการควบคุมเกนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขยาย เพื่อให้สัญญาณออกมาสวยงาม คือ เราจะมีระบบ AGC หรือ Automatic gain control Systems ซึ่งมีอยู่ในวงจรขยายทั่วไป AGC นี้จะทำให้สัญญาณมีค่าสูงสุดต่ำสุดที่แน่นอน และมีขนาดใหญ่กว่าช่วง hysteresis band ของวงจรชมิททริกเกอร์(Schmitt trigger) และช่วง hysteresis band คือช่วงที่เป็นขอบเขตของพัลซ์ว่าระดับไหนเป็น high ระดับไหนเป็น low (ดูในรูปที่ 2)

จากที่กล่าวมาแล้วว่า AGC สามารถแก้ปัญหาจากสัญญาณรบกวนได้ดี และนอกจากนี้สัญญาณประเภทลอจิก AGC จะช่วยลด Slew rate ได้ (เป็นผลมาจาก noise) สิ่งที่ตามมาอีกเนื่องจากว่า เครื่องวัดความถี่จะรับสัญญาณประเภท ac อย่างแน่นอน ทำให้เกิดปัญหาเมื่อรูปสัญญาณไม่สมมาตร (asymmetric) ตัวอย่างเช่น พัลซ์ต่างๆซึ่งมันจะมีผลต่อจุดทริกและส่งผลถึงความผิดพลาดในการวัดได้ ดังนั้นเครื่องมือวัดที่ดีจะต้องมีคู่มือในการปรับระดับของจุดทริกรวมทั้งการควบคุมความไวของเครื่องด้วย ทั้งนี้เครื่องวัดความถี่และเครื่องวัดคาบเวลาก็มีลักษณะเดียวกันดังที่กล่าวมาแล้ว นั่นคือการควบคุมจุดทริกเป็นความแน่นอนและเที่ยงตรงของเครื่องวัดที่จะแสดงออกมาให้เห็นด้วยค่าความผิดพลาดของเครื่องหรือความละเอียดนั่นเอง

1.2 การนับตัวเลข

ระบบดิจิตอลในปัจจุบันเข้ามามีบทบาทอยู่มาก ในวงจรเครื่องนับก็จะมีในส่วนของวงจรหาร, นับสิบ, ส่วนแสดงผล (Display) ที่ใช้กำลังขับต่ำ (low power) ความเร็วในการแสดงผลอยู่ระหว่าง 5 ถึง 100 MHz คุณสมบัติเหล่านี้ ต่อมาได้ถูกรวบรวมไว้ใน chip ซึ่งจะมีชื่อเรียกว่า high speed counter chips ที่มีการทำงานได้อย่างเฉียบคมมาก ผู้ผลิตส่วนใหญ่รับรู้เทคโนโลยีระดับนี้มานานแล้ว แต่ว่ายังขาดกำลังในการผลิต ซึ่งในปัจจุบันก็ยังคงมีเทคโนโลยีในระดับนี้อยู่ เป็นผลให้เครื่องรับในปัจจุบันมีความละเอียดสูงและความผิดพลาดน้อย

ปัญหาอีกอย่างของเครื่องนับความถี่ ก็คือข้อจำกัดทางด้านความถี่ เพราะว่าการทำงานของอุปกรณ์อิเลคทรอนิคส์ในปัจจุบันจะมีความเร็วได้สูงสุดประมาณ 100 MHz ซึ่งเครื่องนับธรรมดาก็ทำได้ แต่ถ้าเราต้องการวัดความถี่ที่สูงกว่านี้ เราต้องมาสร้างอุปกรณ์ที่ทำงานที่ความถี่สูงกว่านี้ ก็ต้องมีการวิจัยกันมากมาย แต่ถ้าเราใช้เทคนิคที่เรียกว่า *เทคนิคการหาร* (Prescaling technique) คือเราจะลดทอนความถี่ที่เรารับเข้ามาก่อนที่จะเอาไปนับ โดยเราจะต้องมีการเทียบสเกลกลับด้วยการคูณกับแฟคเตอร์ เมื่อเราใช้เทคนิคนี้แล้วจะทำให้สามารถนับความถี่ได้ถึง 1 GHz และเราอาจออกแบบให้มากกว่านั้นก็ได้ นอกจากนี้ยังมีเทคโนโลยีใหม่ คือ เทคโนโลยี GaAs เป็นสิ่งประดิษฐ์ชิ้นใหม่ ซึ่งจะนับได้อย่างน้อย 4 GHz แต่สิ่งที่ตามมาก็คือความละเอียดจะลดลงและความผิดพลาดจะมากขึ้น ซึ่งถ้าเรายอมรับได้ก็ไม่มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากเทคนิคการหารแล้ว ยังมีวิธีเพิ่มความสามารถในการนับ โดยใช้เกทที่ทำงานที่ความเร็วสูงขึ้น เกทที่ว่านี้ก็คือ เกทหลัก (main GATE) ซึ่งจากที่กล่าวมาในตอนต้นความผิดพลาดจะมีค่าไม่เกิน ±1รูปคลื่น ซึ่งวิธีนี้จะต้องระวังเรื่องการเปลี่ยนแปลงของเวลาขาขึ้นขาลงต้องเปลี่ยนแปลงทันกับความถี่ที่เราจะนับ ซึ่งวิธีนี้ไม่นิยมกันเพราะหาเกทที่มีความเร็วสูงได้ยาก และ ราคาแพงด้วย

แต่ในอนาคตเราคงจะได้พบกับชิพตัวใหม่ที่มีชื่อว่า "Counter on the chip" ซึ่งจะเป็นลักษณะของยุค Microprocessor และความเร็วในการทำงานจะสูงมาก ความละเอียดจะสูง คิดว่าจะได้รับความนิยมอย่างแน่นอน

1.3 ความผิดพลาดของการนับ

สำหรับเครื่องวัดความถี่จากที่กล่าวมาในหัวข้อแรก ๆ มันควรจะผิดพลาดไปไม่เกิน ±1พัลซ์ของการนับนั้น ซึ่งจะมีผลมากที่ความถี่ต่ำๆ แต่ที่ความถี่สูงเมื่อเทียบสัดส่วนแล้วจะเห็นว่าอาจเป็น 1ในพัน หรือ 1ในแสน ซึ่งเราขอมรับได้

มาพิจารณาที่ความถี่ต่ำ เช่น ความถี่ 1Hz นั้นคือ ช่วงเวลาที่เปิดเกทนาน 1วินาที ส่วนของการนับจะต้องนับได้ 1พัลซ์ แต่ถ้ามันเกิดนับผิดไปเป็น 2พัลซ์ใน 1วินาที จะเห็นว่าผิดพลาดไปถึง 50% ซึ่งความละเอียดของเครื่องมือวัดเราจะดูกันที่ตรงนี้ แต่ในปัจจุบันนี้ไปจนถึงอนาคตการนำเอา Microprocessor เข้ามาใช้ควบคุมการทำงานของการนับความถี่ หรือการวัดคาบเวลาจะทำให้ความละเอียดในการวัดเป็นไปตามต้องการได้ แต่อย่างไรก็ตามในเมื่อความละเอียดเป็นอัตราส่วนระหว่างความถี่อินพุทเทียบกับความถี่ของสัญญาณนาฬิกา ดังนั้นที่ความถี่ต่ำก็จะถูกจำกัดอยู่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034863

บทที่ 2

หลักการและการทำงาน

ของ FREQUENCY COUNTER INTERFACE

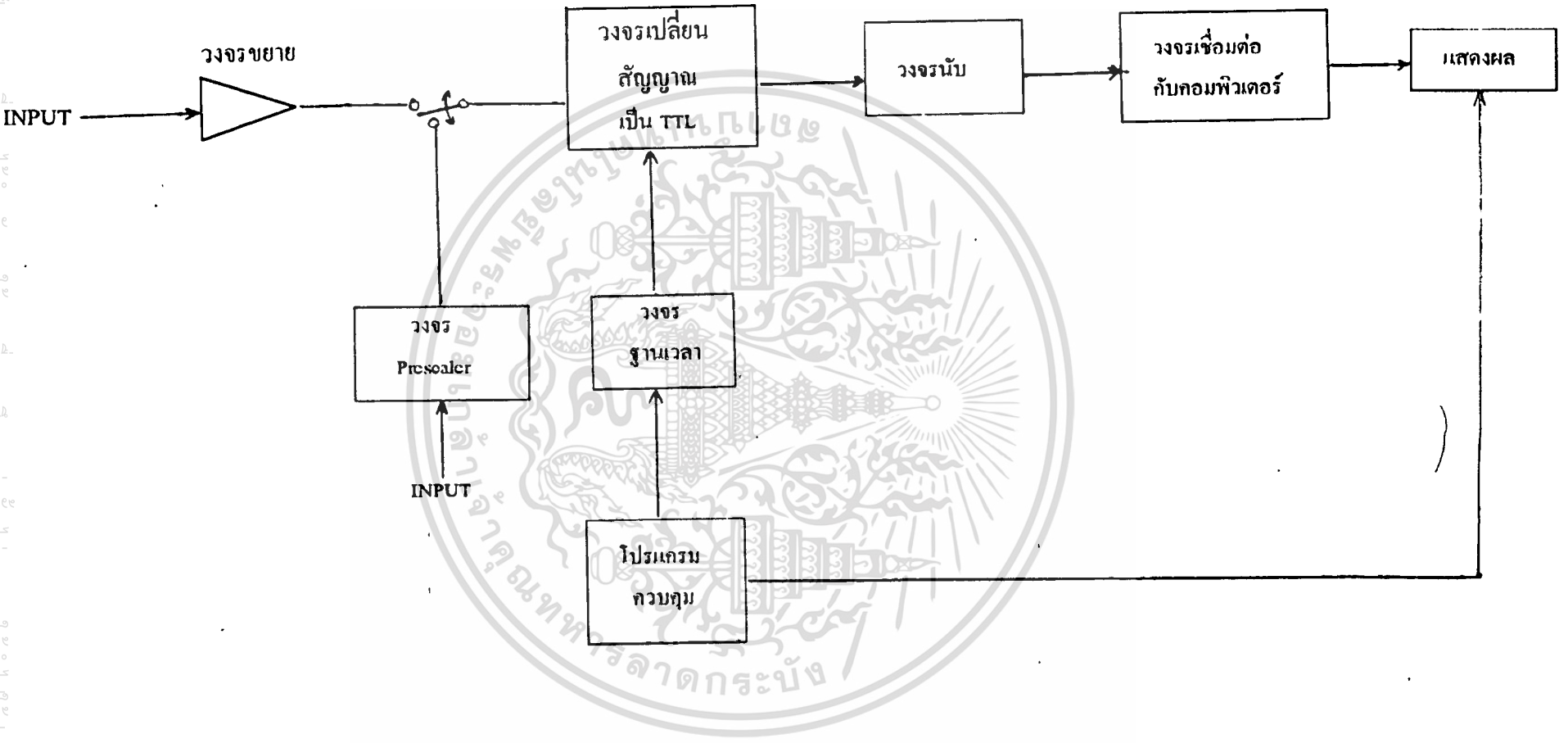
WITH COMPUTER

Frequency Counter ที่แสดงผลที่หน้าจอคอมพิวเตอร์นี้ สามารถวัดได้ทั้งความถี่ และคาบเวลาของความถี่ โดยวงจรนี้มีการใช้อุปกรณ์ทางอิเลคทรอนิกส์ต่าง ๆ มากมายในแต่ละส่วน ซึ่งประกอบด้วย ส่วนขยายสัญญาณทางด้าน INPUT และเปลี่ยนสัญญาณที่ได้รับได้ เป็นสัญญาณ TTL, ส่วนฐานเวลาซึ่งให้กำเนิด CLOCK ที่ความถี่ต่าง ๆ ,ส่วนวงจรนับ เป็นส่วนที่นำสัญญาณ TTL กับสัญญาณ CLOCK มาแอนด์กัน แล้วทำการนับ, ส่วน INTERFACE กับคอมพิวเตอร์ เป็นส่วนที่จะนำข้อมูล จากการวัดเข้าสู่คอมพิวเตอร์ และที่สำคัญ คือการแสดงผลออกที่หน้าจอคอมพิวเตอร์นี้ ทางผู้จัดทำจะใช้ Software ภาษาซี เป็นส่วนควบคุม สามารถเขียนกราฟของข้อมูลที่ได้รับทุก ๆ 1 วินาที และคุณสมบัติต่าง ๆ อีกมากมาย ซึ่งจะกล่าวถึงรายละเอียดทุกๆส่วนในภายหลัง

เนื่องจากการทำงานครั้งนี้ เป็นความพยายามของทางผู้จัดทำที่เคยมีประสบการณ์ทางด้านการทำเครื่องวัดความถี่ ซึ่งวัดได้ในช่วง 5.5 kHz ถึง 10 MHz เท่านั้น ด้วยเหตุผลนี้ จึงทำการออกแบบ และวางแผนการใช้อุปกรณ์ต่าง ๆ เพื่อหวังที่จะทำให้วัดค่าความถี่ให้ได้ในช่วงที่กว้างที่สุด โดยมีความคิดอยากจะให้วัดได้ตั้งแต่ช่วงที่เป็น Hz จนถึง GHz

รูปต่อไปนี้ เป็นแผนผังของวงจรที่ออกแบบไว้ในแต่ละส่วนทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

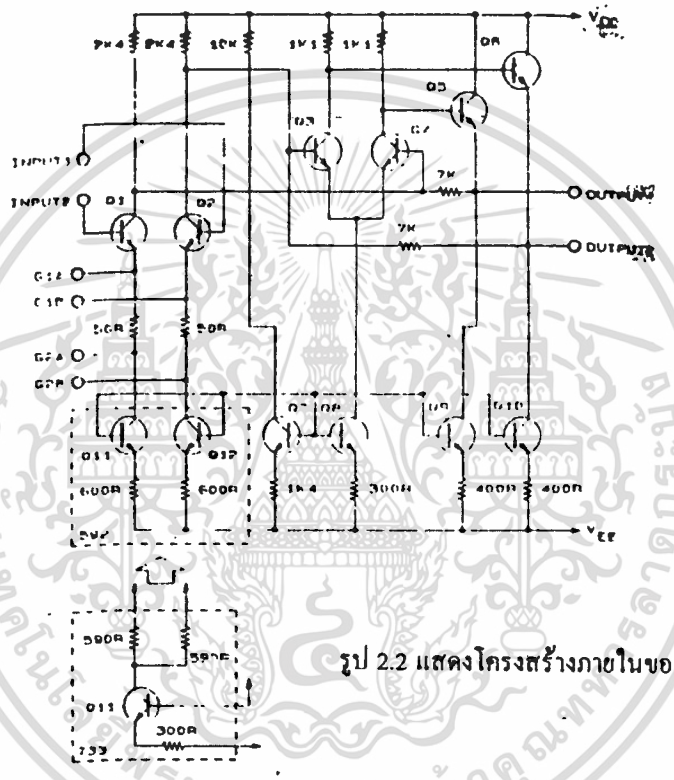


รูป 2.1 แสดง BLOCK DIAGRAM ของชิ้นงาน FREQUENCY COUNTER INTERFACE WITH COMPUTER

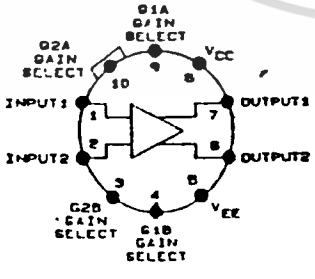
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้ทำซ้ำ ดัดแปลง หรือเผยแพร่ในรูปแบบใดๆ โดยไม่ได้รับอนุญาต

2.1 ส่วนขยายสัญญาณ INPUT และเปลี่ยนเป็นสัญญาณ TTL

ในเรื่องของเครื่องมือวัดนั้นจะเน้นหนักในเรื่องของวงจรขยายสัญญาณแบบย่านกว้าง (WIDE - BAND) เพราะฉะนั้นในเครื่องมือวัดหลาย ๆ ประเภทจึงต้องมีการนำเอาวงจรปรีแอมปลิฟายร์มาต่อใช้งานเข้าไป ช่วยในการขยายสัญญาณให้มีระดับความแรงของสัญญาณที่เหมาะสม ซึ่งจะพบได้ในเครื่องมือวัดหลาย ๆ ชนิด เช่น มิเตอร์ หรือออสซิลโลสโคปเป็นต้น



รูป 2.2 แสดงโครงสร้างภายในของ IC เบอร์ 592 และ 733

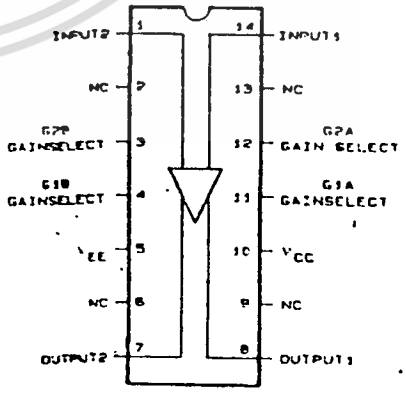


รูป 2.3 A

GAIN SELECT = สำหรับเลือกอัตราการขยายสัญญาณ

INPUT 1 = ขาสัญญาณอินพุท 1

INPUT 2 = ขาสัญญาณอินพุท 2



รูป 2.3 B

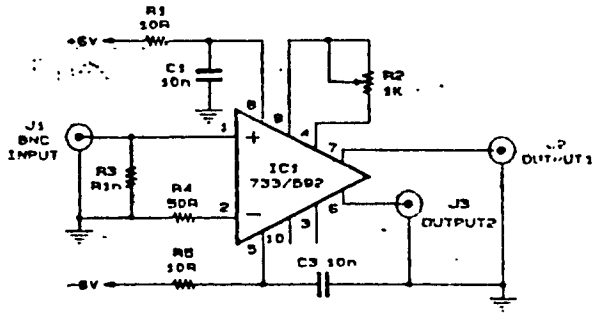
OUTPUT 1 = ขาสัญญาณเอาท์พุท 1

OUTPUT 2 = ขาสัญญาณเอาท์พุท 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

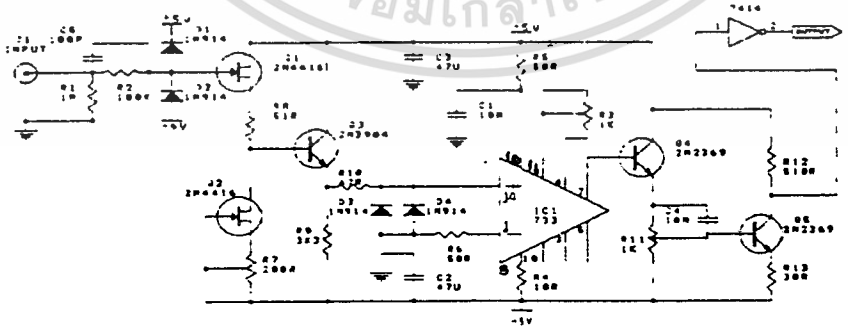
รูป 2.3 แสดงโครงสร้างภายในของ IC เบอร์ 592 และ 733 เมื่อมองจากด้านบนลงมา

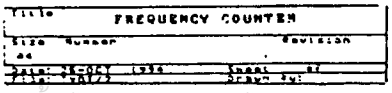
รูป 2.3 A) เป็นแบบตัวถังโลหะ รูป 2.3 B) เป็นแบบตัวถัง



รูป 2.4 ตัวอย่างการประยุกต์ใช้ IC เบอร์ 592 / 733

ยกตัวอย่าง วงจรใช้งานในรูปจะเป็นลักษณะของวงจรปรีแอมปลิฟายร์ แบบพื้นฐานที่ออกแบบมาสำหรับงานทางด้านเครื่องมือวัดแบบอเนกประสงค์ อุปกรณ์ที่ใช้หัวใจสำคัญอยู่ที่ไอซี ออปแอมป์ 592 หรือ 733 ผู้ใช้สามารถจะกำหนดค่าตัวต้านทาน R3(Rin) ให้มีค่าที่เหมาะสมอยู่ในราว ๆ หลักร้อยโอห์ม เพื่อให้ได้ค่าการวัดในย่านที่ต้องการออกมาได้ แต่อย่างไรก็ตาม วงจรในรูป นี้จะให้ค่าความต้านทานทางด้านอินพุทมีค่าต่ำ และค่าความต้านทานทางด้านเอ้าท์พุทมีค่าสูง ดังนั้นจึงได้มีการนำวงจร มาทำการปรับปรุง โดยประยุกต์วงจรปรีแอมป์ ดังกล่าวมาใช้ในวงจรเครื่องนับความถี่ (FREQUENCY COUNTER) และนำเฟ็ท Q1 มาทำเป็นวงจรบัฟเฟอร์ต่อคืนไว้ทางด้านอินพุทของไอซี ออปแอมป์ ทั้ง 2 ตัว ทำให้ค่าความต้านทานทางอินพุท หรืออินพุทอิมพีแดนซ์ของวงจรนี้มีค่าสูงขึ้นถึง 1 เมกกะโอห์ม และเพิ่มส่วนของวงจรป้องกันทางด้านอินพุทด้วยการต่อ ไดโอด D1, D2, D3, D4 แบบไบอัสตรง (FORWARD BIAS) คอยป้องกันไม่ให้สัญญาณขาเข้ามีความแรงเกินไป จนเกิดอาการโอเวอร์ไดร์ฟได้, สำหรับไดโอด D3 และ D4 นั้น ยังทำให้ค่าของสัญญาณเอ้าท์พุทมีค่าสูงขึ้นกว่าระดับอิ่มตัว วงจรเฟทบัฟเฟอร์ทางอินพุทยังช่วยให้แบนด์วิดธ์การขยายสัญญาณมีความกว้างถึง 100 MHz เป็นการแก้ไขที่แยบยลมาก



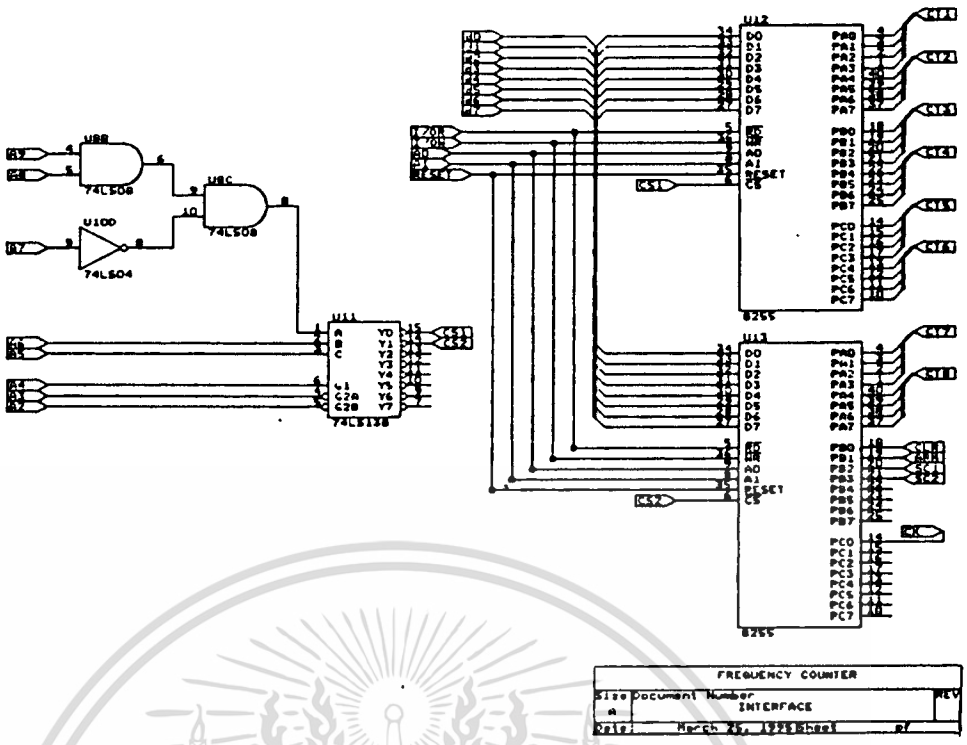


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.5 แสดงวงจรที่ใช้เปลี่ยนสัญญาณอินพุท เป็นสัญญาณ TTL

ทางด้านการเชื่อมต่อวงจรปรีแอมปลิฟายร์เข้ากับไอซีประเภททีทีแอล เช่น ไอซี ทีทีแอล ในวงจรของเครื่องนับความถี่ (FREQUENCY CONUTER) ในรูปนี้จะเห็นว่าวงจรทางด้านเอ้าท์พุทมีการต่อบัฟเฟอร์เอาไว้ด้วยการใช้ทรานซิสเตอร์ Q4, Q5 จากนั้นนำเข้าไปต่อผ่านไอซี 7414 ที่ทำหน้าที่เป็นวงจรอินเวอร์เตอร์ วงจรดังกล่าวนี้จะมีจุดการทำงานในการนับความถี่ได้ถึง 45 เมกกะเฮริทซ์ (MHz) โดยเกนอัตราขยายของไอซี 592 หรือไอซี 733 จะต้องมีค่าเท่ากับ 10 (ในกรณีของไอซี 733 ขาควบคุมเกนการขยายจะต้องเปิดวงจรเอาไว้) ส่วนในไอซี 592 จะใช้ตัวต้านทานจากภายนอกเป็นตัวคอยกำหนดค่าเกนการขยายซึ่งจะต่อตัวต้านทานคร่อมขั้ว GAIN SELECT INPUT ขา GIA และขา GIB (ตรงกับขาที่ 4 และขา 11 ของตัวถังแบบ DIP) แต่ถ้าจะให้สะดวกก็ใช้วอลลุ่มขนาด 1KΩ มาต่อเข้าไปแทนจะทำให้การปรับแต่งค่าทำได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 แสดงวงจรที่ใช้เชื่อมต่อกับคอมพิวเตอร์ (INTERFACE CIRCUIT)

2.2 ส่วนอินเตอร์เฟส

ส่วนนี้ใช้ในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องวัดความถี่ โดยทำเป็นการ์ดเสียบเข้ากับสล็อตของเครื่อง PC/XT และ PC/AT อาศัยบัสตำแหน่ง (address bus), บัสข้อมูล (data bus), สัญญาณควบคุมการอ่านเขียน, สัญญาณรีเซต และไฟเลี้ยงจากคอมพิวเตอร์ การควบคุมการทำงานของส่วนนี้ สามารถใช้โปรแกรมภาษาซี ควบคุมได้ โดยการใช้คำสั่ง inport และ outportb

การทำงานของส่วนนี้ใช้ไอซี เบอร์ 8255 A เป็นหลัก สายสัญญาณข้อมูลเข้า และข้อมูลออกทั้ง 24 เส้น ของ 8255 มีระดับสัญญาณ TTL จึงเป็นการง่ายในการใช้ 8255 เชื่อมต่อกับวงจรลอจิกอื่น ๆ และไอซีเบอร์ 74LS138 ถูกนำมาใช้เป็น DECODERS เพื่อเลือก address ของ 8255 A

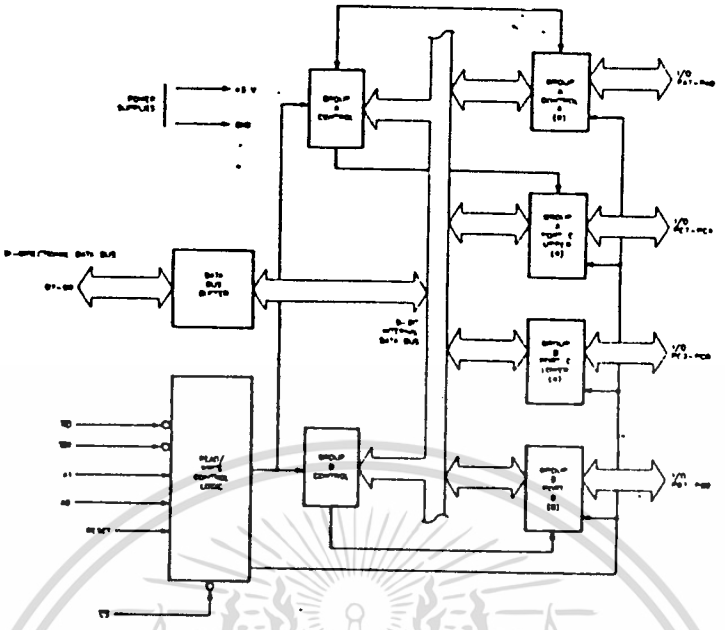
8255 นี้สามารถกำหนดการทำงานได้ 3 โหมด คือ โหมด 0, โหมด 1 และ โหมด 2

2.2.1 โครงสร้างภายในของ 8255

8255 A มีโครงสร้างภายในดังแสดงในรูป การเรียกชื่อพอร์ตของ 8255A จะเรียกพอร์ตว่า พอร์ต *A,B* และ C โดยพอร์ต C แยกเป็น 2 ส่วนคือ $PC_0 - PC_3$ เรียกว่า พอร์ต C ล่าง และ $PC_4 - PC_7$ เรียว่า พอร์ต C บน

จากรูป สามารถอธิบายหน้าที่การทำงานของแต่ละส่วนได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.7 แสดงโครงสร้างภายใน 8255

Data Bus Buffer

เป็นบัฟเฟอร์ 8 บิต สองทิศทาง 3 สถานะ (3-state) ใช้ในการอินเตอร์เฟสกับบัสข้อมูลของระบบ (system data bus) ข้อมูลจะถูกส่งหรือรับโดยผ่านทางบัฟเฟอร์ขึ้นอยู่กับการประมวลผลของซีพียูว่าให้ทำการส่งหรือรับข้อมูล

Read/Write and Control Logic

ทำหน้าที่จัดการเกี่ยวกับการส่งผ่านข้อมูลและ control หรือ status ทั้งภายในและภายนอก ส่วนนี้จะรับอินพุตจากบัสแอดเดรส และบัสควบคุมของซีพียูแล้วนำมาถอดรหัสในการควบคุมพอร์ต บล็อกนี้มีสายสัญญาณควบคุมจากภายนอก ดังต่อไปนี้

Chip Select ($\overline{CS}$) เมื่อขานี้ได้รับลอจิก "0" จะทำให้ 8255A ต่อเข้ากับระบบบัสของซีพียูอ่านหรือเขียนข้อมูลผ่านพอร์ตได้

Read ($\overline{RD}$) เป็นขาสัญญาณอินพุตที่ต้องส่งมาจากซีพียู เมื่อสัญญาณขานี้มีลอจิก "0" และ $\overline{CS}$ เป็น "0" ด้วย ซีพียูจะทำการอ่านข้อมูลจากบัสข้อมูลของ 8255A

Write ($\overline{WR}$) ขาสัญญาณการเขียนจะแอกทีฟเมื่อ $\overline{CS}$ เป็น "0" และ $\overline{R}$ เป็น "0" โดยสัญญาณนี้จะถูกส่งมาจากซีพียูทำให้ซีพียูสามารถเขียนข้อมูลบนบัสข้อมูลของ 8255 A ได้

Port Select 0 and Port Select 1 (A_0 and A_1) สัญญาณอินพุตที่สองขานี้จะต้องสัมพันธ์กับสัญญาณ $\overline{RD}$,$\overline{WR}$ และ $\overline{CS}$ เพื่อใช้ในการเลือกพอร์ตใช้งาน

Reset (RESET) แอกทีฟ "1" สำหรับขานี้ เมื่อมีลอจิก "1" เข้ามาจะทำให้เกิดการเคลียร์รีจิสเตอร์ควบคุม (control register) และทุกพอร์ตจะถูกเซตสู่โหมดอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

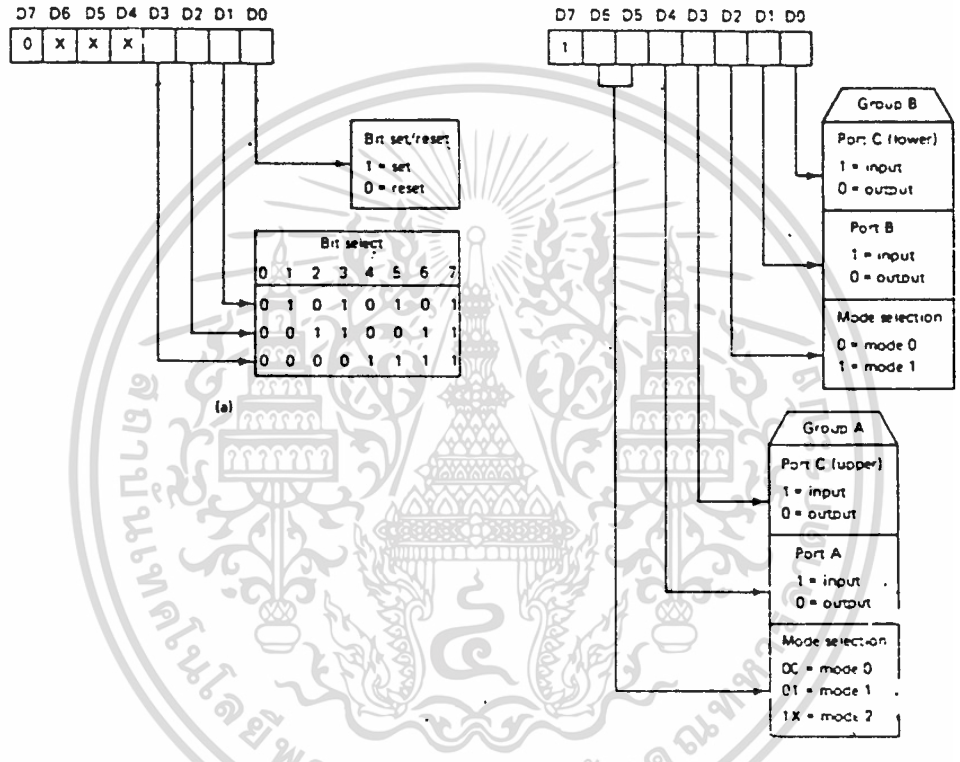
Group A and Group B controls แต่ละบล็อกจะรับคำสั่งจาก read/write control logic และรับ control word จากบัสข้อมูลภายใน เพื่อใช้ในการควบคุมพอร์ต

Control Group A-จะควบคุมพอร์ต A และพอร์ต C บน $(PC_4 - PC_7)$

Control Group B- จะควบคุมพอร์ต B และพอร์ต C ล่าง $(PC_0 - PC_3)$

Ports A, B and C8255A ประกอบด้วยพอร์ต 8 บิต 3 พอร์ต (พอร์ต A, B และ C) ซึ่งพอร์ตที่ 3 นี้ สามารถกำหนดรูปแบบการใช้งานได้ โดยอาศัยซอฟต์แวร์ช่วยจัดการ

การใช้งาน เราจะใช้เฉพาะโหมด 0 ซึ่งเราจะต้องเป็นผู้กำหนด CONTROL CODE PORT ตาม การทำงานที่เราต้องการดังต่อไปนี้



รูป 2.8 รายละเอียดของ CONTROL CODE PORT

แอดเดรสที่เราใช้ในวงจรนี้จะต้องเป็น address ที่ยังไม่ได้ใช้งานซึ่งการจัดสรรตำแหน่งแอดเดรสบนเครื่อง IBM PC แสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.9 การจัดสรรตำแหน่งแอดเดรสบนเครื่อง IBM PC

อินพุตเอาต์พุตแอดเดรส	หน้าที่การทำงาน
000H-00FH	ส่วนควบคุม DMA
020H-021H	ส่วนควบคุมการอินเตอร์รัพต์
040H-043H	ไทเมอร์/เคาน์เตอร์
060H-063H	รีจิสเตอร์ระบบ
080H-083H	รีจิสเตอร์ของ DMA
0A0H-0BFH	NMI รีจิสเตอร์อินเตอร์รัพต์
0C0H-0FFH	สงวนไว้
100H-1FFH	ส่วนควบคุมฟรอนต์พาเนล
200H-20FH	สำหรับเกมคอมพิวเตอร์ (พอร์ตเกม)
210H-217H	ส่วนขยายเพิ่มเติม
220H-24FH	สงวนไว้
278H-27FH	เครื่องพิมพ์ที่ 2
2F8H-2FFH	ซีเรียลอินเตอร์เฟสที่ 2
300H-31FH	สำหรับพอร์ตทดลองเพิ่มเติม
320H-32FH	ส่วนควบคุมฮาร์ดดิสก์
378H-37FH	พรินเตอร์อินเตอร์เฟส (พอร์ตขนาน)
380H-38FH	SDLC อินเตอร์เฟส
3A0H-3AFH	สงวนไว้
3B0H-3BFH	สำหรับโมโนโครมอะแดปเตอร์ และเครื่องพิมพ์
3C0H-3CFH	สงวนไว้
3D0H-3DFH	สำหรับการ์ดโคเลอร์กราฟิก
3E0H-3E7H	สงวนไว้
3F0H-3F7H	ส่วนควบคุมฟลอปปี้ดิสก์
3F8H-3FFH	ซีเรียลอินเตอร์เฟส

ซึ่งเราจะได้ address 300 H ถึง 307 H ในการทำงานของ 8255 ทั้งสองตัว นอกจากนี้สล๊อตยังต้องสัมพันธ์กับสัญญาณต่าง ๆ บนเครื่องพีซี ซึ่งแสดงได้ดังนี้

ชื่อสัญญาณ	ตำแหน่งขาบนสล๊อต		ตำแหน่งขาบนสล๊อต	ชื่อสัญญาณ
GND	B01		A01	I/O CHCK
RESET	B02		A02	D7
+5V	B03		A03	D6
IRQ2	B04		A04	D5
-5V	B05		A05	D4
DREQ2	B06		A06	D3
-12V	B07		A07	D2
reserved	B08		A08	D1
+12V	B09		A09	D0
GND	B10		A10	I/O CHRDY
MEMW	B11		A11	AEN
MEMR	B12		A12	A19
IOWC	B13		A13	A18
IORC	B14		A14	A17
DACK3	B15		A15	A16
DREQ3	B16		A16	A15
DACK1	B17		A17	A14
DREQ1	B18		A18	A13
DACK0	B19		A19	A12
CLK	B20		A20	A11
IRQ7	B21		A21	A10
IRQ6	B22		A22	A9
IRQ5	B23		A23	A8
IRQ4	B24		A24	A7
IRQ3	B25		A25	A6
DACK2	B26		A26	A5
TC	B27		A27	A4
ALE	B28		A28	A3
+5V	B29		A29	A2
OSC	B30		A30	A1
GND	B31		A31	A0

รูป 2.10 แสดงตำแหน่งขาบนสล๊อต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณต่าง ๆ บนบัสของระบบ

หน้าที่ของสัญญาณต่าง ๆ ในสล็อตแบบ 8 บิต ของเครื่องไอบีเอ็มพีซี และคอมแพตติเบล แสดงไว้ดังตารางที่ 2 ซึ่งมีสัญญาณที่สามารถนำไปใช้ประโยชน์และน่าสนใจเป็นพิเศษดังนี้

OSC (ขา B30) เป็นสัญญาณนาฬิกาของบัสอินพุตเอาต์พุต ซึ่งมีมาตรฐานที่ 4.77 เมกะเฮิรตซ์ ในรุ่น XT และ 14.138 เมกะเฮิรตซ์ ในรุ่น AT มีอัตราส่วน mark/space เป็น 1:1

CLK (ขา B20) เป็นสัญญาณนาฬิกาของระบบ ตามมาตรฐานของเครื่องไอบีเอ็มพีซี จะมีความถี่เท่ากับ 1/3 ของความถี่ออสซิลเลเตอร์ เช่น 1.59 เมกะเฮิรตซ์ (เมื่อใช้ความถี่ออสซิลเลเตอร์ 4.77 เมกะเฮิรตซ์) มีอัตราส่วน mark/space เป็น 1:2

ในปัจจุบันเครื่องพีซีได้ถูกพัฒนาให้มีความเร็วในการทำงานสูงขึ้นมากดังนั้นมาตรฐานของความถี่ทั้ง OSC และ CLK จึงเพิ่มขึ้นตามไปด้วย จุดนี้ก็อยู่ที่เครื่องของคุณว่าทำงานที่ความเร็วเท่าใด

RESET (ขา B02) ใช้เมื่อต้องการเริ่มต้นการทำงานของระบบใหม่ เมื่อเปิดสวิตช์เครื่อง หรือหลังจากเกิดอาการหยุดชะงัก หรือรีเซตการทำงานของฮาร์ดแวร์

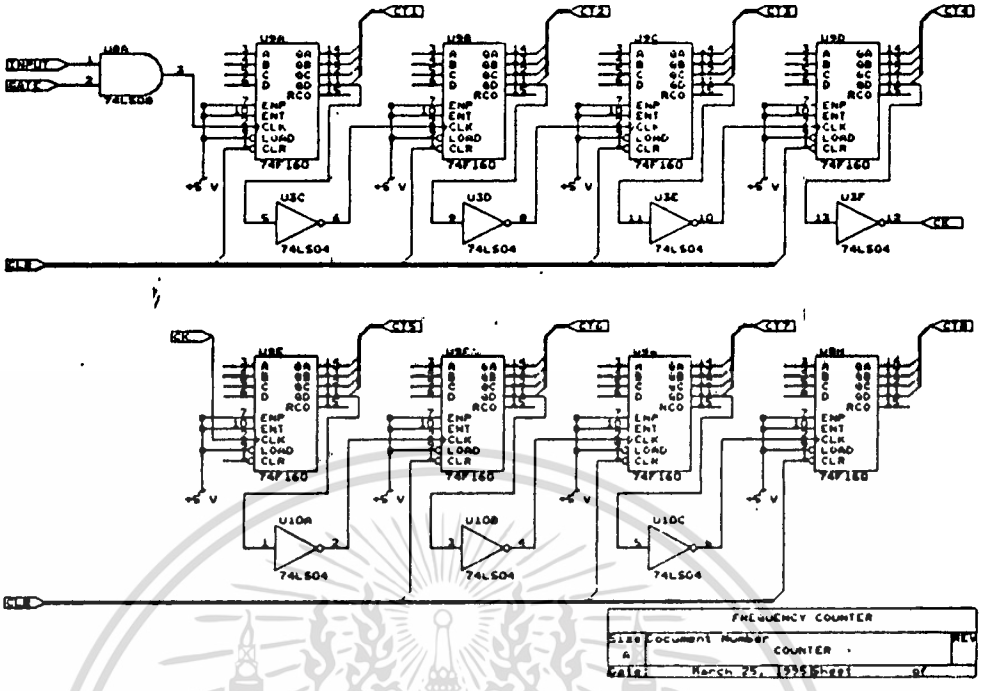
$\overline{IOWR}$ (ขา B13) เป็นสัญญาณจากบัสคอนโทรลเลอร์ เมื่อต้องการเขียนหรือส่งข้อมูลกับพอร์ตอินพุตเอาต์พุต

$\overline{IORD}$ (ขา B14) เป็นสัญญาณจากบัสคอนโทรลเลอร์เช่นกัน เมื่อต้องการอ่านข้อมูลจากพอร์ตอินพุตเอาต์พุต

$\overline{MEMRD}$ (ขา B12) เป็นสัญญาณแสดงการทำงานขณะที่โปรเซสเซอร์หรือ DMA คอนโทรลเลอร์กำลังอ่านข้อมูลจากหน่วยความจำ

$\overline{MEMWR}$ (ขา B11) เป็นสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำ ซึ่งกำหนดตำแหน่งแอดเดรสด้วยสายสัญญาณ A_0 ถึง A_{19}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.11 แสดงวงจรที่ใช้นับความถี่ (COUNTER CIRCUIT)

2.3 ส่วนของวงจรนับ

ในส่วนของวงจรนับ ใช้ไอซีเคาน์เตอร์เบอร์ 74F160 โดยเป็นไอซีเคาน์เตอร์แบบซิงโครนัส ซึ่งหมายความถึงว่า การทำงานของตัวฟลิปฟลอปที่อยู่ในไอซีเหล่านี้จะใช้สัญญาณนาฬิกา (CLOCK) ตัวเดียวกัน และทำงานพร้อมกันด้วย สัญญาณออกของมันจะเป็นแบบ BCD (Binary Code decimal) ไอซีเบอร์นี้จะนับสัญญาณดิจิตอลในช่วงของ 0 - 9 เท่านั้นซึ่งก็คือเลขฐานสิบ หนึ่งหลักนั่นเอง และมีสัญญาณออกเป็นข้อมูล 4 บิต

การทำงานของขาสัญญาณต่าง ๆ มีดังนี้

ขา ENABLE ของเคาน์เตอร์ เป็นขาที่ควบคุมให้ไอซีทำงานหรือไม่ถ้าเป็น "1" ไอซีจะทำงาน ถ้าเป็น "0" ไอซีจะหยุดทำงาน

ขา OUTPUT ของเคาน์เตอร์ สัญญาณข้อมูลออกขนาด 4 บิต แบบ BCD (Binary Code Decimal)

ขา INPUT ของ CLOCK เมื่อสัญญาณนาฬิกาเปลี่ยนจาก "0" ไปเป็น "1" ค่าที่อยู่ในเคาน์เตอร์จะถูกพวกเพิ่มอีกหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา CLEAR ถ้าเป็น "0" ค่าของเคาน์เตอร์จะถูกลบข้อมูลให้เป็นศูนย์หมด ไอซีจะลบข้อมูลทันทีที่ขานี้เป็น "0"

ขา INPUT ของพรีเซต สัญญาณต่าง ๆ ที่ป้อนให้กับช่องสัญญาณ A,B,C,D นี้จะถูกอ่านเข้าไปเมื่อขา LOAD เป็น "0" อยู่และสัญญาณคล็กเปลี่ยนจาก "0" ไปเป็น "1"

ขา OUTPUT ของแคร์รี่ (ตัวทด) เมื่อเคาร์เตอร์นับไปถึงค่าสูงสุด 1001 (เลข 9) แล้วจะเกิดตัวทดขึ้น ทำให้ขา RCO นี้เป็น "1" ออกมา กรณีอื่นจะอยู่ที่ "0" เสมอ

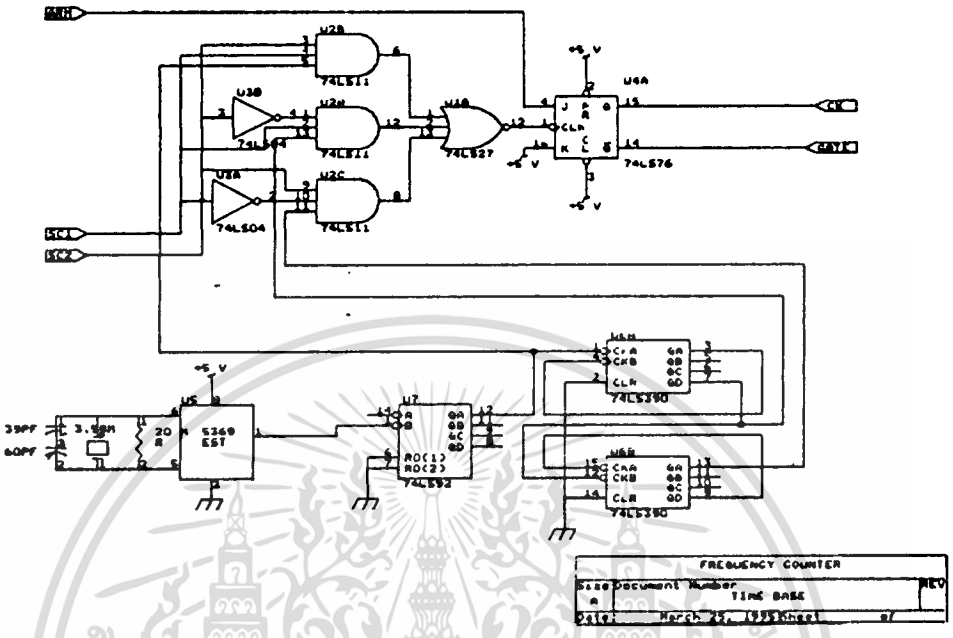
ขา LOAD สัญญาณเมื่อต้องการให้เคาน์เตอร์อ่านอินพุทที่ขาพรีเซตจะต้องป้อนสัญญาณเข้าขานี้เป็น "0" แต่ถ้าหากให้ขานี้เป็น "1" ก่อนที่สัญญาณนาฬิกาเปลี่ยนจาก "0" เป็น "1" จะไม่ทำให้ไอซีอ่านค่าที่ขาฟรีเซตเข้าไป

การนับของไอซี

สัญญาณที่ต้องการจะนับจะถูกนำเข้าไปแอนด์กับสัญญาณจากส่วนของฐานเวลาที่เราเลือกโดยใช้แอนด์เกทสัญญาณที่ได้จะนำไปป้อนเข้าที่ขา CLOCK ของ IC เบอร์ 74F160 โดยไอซีเบอร์นี้มีหลักการที่ว่าถ้าสัญญาณเปลี่ยนจาก "0" ไปเป็น "1" เคาน์เตอร์หรือไอซีนี้จะคอยตรวจจับ พัลส์ขาขึ้น และทำการนับโดยเพิ่มค่าในตัวมันขึ้นอีกหนึ่ง การนับจะควบคุมได้โดยขา Enable P และ Enable T ถ้าหากสองขานี้เป็น "1" ไอซีจะทำหน้าที่เป็นเคาน์เตอร์คือนับพัลส์ของสัญญาณที่เข้าขา CLOCK ไปเรื่อย ๆ จนกระทั่งขาใดขาหนึ่งเป็น "0" หรือไม่มีสัญญาณเข้าที่ขา CLOCK ไอซีจึงจะหยุด

เมื่อไอซีนับถึง 1000 (เลข 8) และ1001 (เลข 9) ขา Q_D จะเป็น 1 และเมื่อเริ่มนับ 0 ใหม่ Q_D จะเป็น 0 เมื่อเราต่อสัญญาณออกของ Q_D ผ่านน็อตเกท (NOT GATE) เมื่อนับ 1000 1001 และ 0000 จะทำให้สัญญาณออกจากน็อตเกท เปลี่ยนจาก "0" ไปเป็น "1" เมื่อเรานำไปป้อนเข้าที่ขา CLOCK ของไอซีเบอร์ 74F160 ตัวถัดไปก็จะทำให้ไอซีตัวถัดไปนั้นนับเป็นหลักถัดไปได้ สัญญาณออกของไอซีเคาน์เตอร์ Q_A, Q_B, Q_C, Q_D ทั้ง 9 ตัวจะถูกป้อนเข้าไปยัง PORT ต่าง ๆ ของ 8255 เพื่อเชื่อมต่อกับเครื่องคอมพิวเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.12 แสดงวงจรที่ใช้เป็นฐานเวลา (GATE TIME)

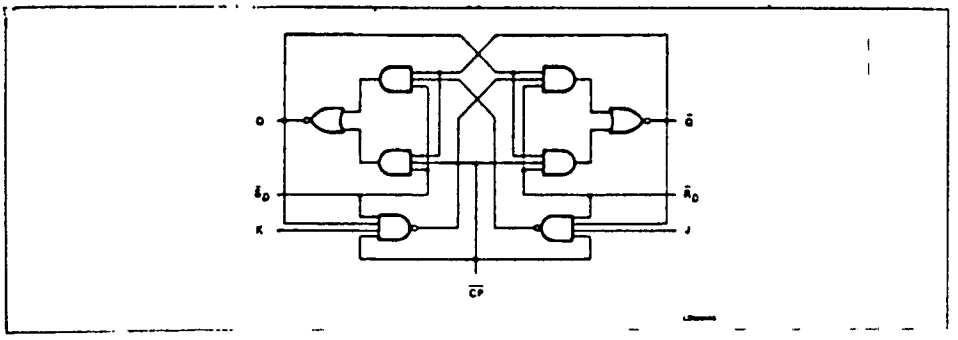
2.4 ส่วนฐานเวลา

ส่วนนี้เป็นส่วนที่สร้างสัญญาณนาฬิกาเพื่อนำไปควบคุมการเปิด-ปิด เกทให้แก่ไอซีเคาน์เตอร์สำหรับในส่วนนี้จะใช้ไอซีเบอร์ 5369AA ซึ่งสร้างฐานเวลาความถี่ 60 Hz สัญญานนาฬิกา 60 Hzนี้จะถูกหารลงจนเหลือ 10 Hz โดยใช้ ไอซีเบอร์ 74 LS 92 และหารลงเหลือ 1 Hz และ 0.1 Hz โดยใช้ 74 LS 390 สัญญาณนาฬิกา 10 , 1 , 0.1 Hz ที่ได้นี้ จะนำไปเป็นฐานเวลาให้แก่วงจรนับ ซึ่งสามารถเลือกฐานเวลาได้ โดยใช้ซอฟแวร์ควบคุม เมื่อสัญญาณฐานเวลาถูกเลือก สัญญาณนี้จะถูกป้อนเข้าที่ขา CLOCK ของ J-K ฟลิป-ฟลอป

ซึ่งมีตารางการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.13 แสดง LOGIC DIAGRAM ของ JK ฟลิปฟลอป



รูป 2.14 แสดงรูปแบบการทำงานของ JK ฟลิปฟลอป

OPERATING MODE	INPUTS					OUTPUTS	
	$\bar{S}_D$	$\bar{R}_D$	$\overline{CP}$[2]	J	K	Q	$\bar{Q}$
Asynchronous Set	L	H	X	X	X	H	L
Asynchronous Reset (Clear)	H	L	X	X	X	L	H
Undetermined[1]	L	L	X	X	X	H	H
Toggle	H	H	⎍	h	h	q	$\bar{q}$
Load "0" (Reset)	H	H	⎍	l	h	L	H
Load "1" (Set)	H	H	⎍	h	l	H	L
Hold "no change"	H	H	⎍	l	l	q	$\bar{q}$

H = HIGH voltage level steady state
h = HIGH voltage level one set-up time prior to the HIGH-to-LOW Clock transition
L = LOW voltage level steady state
l = LOW voltage level one set-up time prior to the HIGH-to-LOW Clock transition
q = Lower case letters indicate the state of the referenced output prior to the HIGH-to-LOW Clock transition
X = Don't care
⎍ = Positive Clock pulse

NOTES:
1 Both outputs will be HIGH while both $\bar{S}_D$ and $\bar{R}_D$ are LOW, but the output states are unpredictable if $\bar{S}_D$ and $\bar{R}_D$ go HIGH simultaneously
2 The 74LS76 is edge triggered. Data must be stable one set-up time prior to the negative edge of the Clock for predictable operation
3 The J and K inputs of the 7476 must be stable while the Clock is HIGH for conventional operation

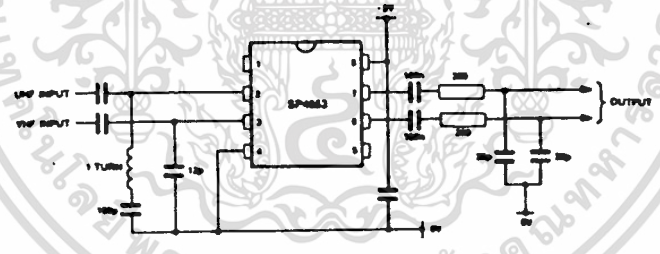
จากตารางการทำงาน เราต้องการฐานเวลาออกมาเพียงหนึ่งคาบเท่านั้น ทำได้โดยการป้อนลอจิก "1" เข้าที่ขา K , PRESET และ CLEAR โดยการควบคุมการเปิด-ปิดเกท ทำได้โดยใช้ขา J โดยเริ่มแรกส่งสัญญาณลอจิก "0" เข้าที่ขา J โดยการต่อขา J เข้ากับ PORT B ของ 8255 แล้วใช้ซอฟแวร์ควบคุม เมื่อขา J เป็น "0" มี CLOCK ขอบขาลงเข้ามาจะทำให้ ขา Q เป็น "0" หลังจากนั้นใช้ซอฟแวร์ควบคุมให้ขา J มีลอจิกเป็น "1" เมื่อมี CLOCK ขอบขาลงเข้ามา จะทำให้ขา Q เป็น "1" สัญญาณออกของขา Q จะถูกป้อนเข้าที่ PORT C ล่าง แล้วใช้ซอฟแวร์ทำการรับค่าจากขา Q มาดู เมื่อตรวจสอบได้แล้ว ว่าขา Q เป็น "1" ก็ทำการเปลี่ยนค่าของขา J ให้มีลอจิกเป็น "0" ดังเดิม ซึ่งการตรวจสอบและเปลี่ยนค่าของสัญญาณที่ขา J มีความรวดเร็วมาก เมื่อ CLOCK ขอบขาลง ถูกต่อไปเข้ามาก็จะทำให้ขา Q เป็น "0" ดังเดิม สัญญาณออกของ J-K ฟลิปฟลอป จึงออกมาเพียงหนึ่งค่าเวลา ไปเป็นฐานเวลาให้แก่วงจรนับ เมื่อวงจรนับ นับค่าสัญญาณที่เข้ามาเสร็จแล้ว ไอซีเบอร์ 8255 จะทำการรับค่า แล้วก็ส่งสัญญาณ CLEAR ให้ OUT PUT ของเคาเตอร์ทุกตัวเป็น 0000 แล้วจึงทำการส่งลอจิก "1" เข้าขา J เพื่อสร้างฐานเวลา 1 คาบ แล้วนำไปเปิดเกท เพื่อรับสัญญาณเข้ามา เป็นอย่างนี้เรื่อยไปตลอดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ส่วนวงจรหารความถี่

ในการวัดความถี่สูงเกินความถี่สูงสุดที่เครื่องวัดความถี่วัดได้ จำเป็นต้องใช้เทคนิคหารความถี่ก่อนที่จะเข้ามายังวงจรนับ โดยใช้ไอซีที่ออกแบบมาสำหรับหารความถี่สูง ๆ โดยเฉพาะมาใช้กับวงจรภาคอินพุทเพื่อเพิ่มประสิทธิภาพให้กับเครื่องวัดความถี่ โดยเลือกใช้ไอซีเบอร์ SP4653ซึ่งเป็นไอซีหารความถี่ด้วย 256 สามารถขยายอินพุทย่านกว้าง วัดความถี่สูงสุดได้ถึง 1 GHz ขยายอินพุทขนาดต่ำสุดตั้งแต่ขนาด 10 มิลลิโวลต์ขึ้นไป

หลักการทำงาน เมื่อมีสัญญาณความถี่สูงตั้งแต่ความถี่ 30-300 MHz (ย่าน VHF) สัญญาณความถี่ในช่วงนี้จะต้องป้อนเข้าที่ขา 3 ส่วนสัญญาณความถี่ตั้งแต่ 300 MHz ไปจนถึง 1GHz (อยู่ในย่าน UHF) ต้องป้อนเข้าที่ขา 2 สัญญาณอินพุทที่เข้ามานี้จะถูกหารด้วย 256 ได้เอาพุทออกขา 6 และขา 7 ตามลำดับ เอาท์พุทที่ออกมานี้ซึ่งมีขนาด 0.6 โวลต์พีคทูพีค (V_{p-p}) จะถูกป้อนเข้าวงจรอินพุท เพื่อขยายให้ได้ 5 โวลต์ป้อนไปยังวงจรนับต่อไป การแสดงผลทำได้โดยใช้ซอฟต์แวร์ รับค่าเข้ามาจากวงจรนับแล้วนำไปคูณ 256 เพื่อแสดงผลความถี่ที่แท้จริงออกมายังหน้าจอคอมพิวเตอร์ต่อไป



รูป 2.15 แสดงส่วนประกอบของวงจรหารความถี่ (PRESCALER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ส่วน SOFTWARE

ในการเขียนโปรแกรทที่ใช้ควบคุมการแสดงผลนี้ ทางกลุ่มผู้จัดทำได้ใช้ภาษาซีในการดำเนินงานทั้งหมด ซึ่งการทำงานของโปรแกรม ประกอบไปด้วยฉาก 3 ฉาก คือ

1. ฉากแนะนำเริ่มต้น เป็นลักษณะของตัวอักษรวิ่ง เพื่อแนะนำหน้าที่ของโปรแกรม และ กลุ่มผู้จัดทำ

2. ฉากก่อนเริ่มการทำงาน จะมีการรับข้อมูลจากผู้ใช้งาน เพื่อใช้ในการกำหนดเวลาการเปิดปิดเกท ซึ่งจะส่งผลถึงการแสดงผลต่อไป

3. ฉากประกอบการทำงาน มีหน้าที่การทำงานต่างๆ 6 หน้าที่ แบ่งเป็น

3.1 เริ่มต้นการทำงาน (START)

จะมีการรับค่าความถี่จากพอร์ทที่เชื่อมระหว่างคอมพิวเตอร์กับส่วนของวงจรนับ (COUTER CIRCUIT) ผ่านทางส่วนวงจรเชื่อมต่อ (INTERFACE CIRCUIT) แล้วนำค่าความถี่ที่ได้นี้มาทำการแสดงผลออกมาให้เห็น ทั้งยังนำไปวาดกราฟ เพื่อให้เห็นถึงความมีเสถียรภาพของเครื่องที่จ่ายสัญญาณความถี่นี้ออกมา

3.2 เก็บข้อมูล (SAVE)

เป็นการนำค่าความถี่ที่ได้จากการวัดในแต่ละครั้ง มาเก็บเป็นแฟ้มข้อมูล โดยที่ผู้ใช้งานสามารถตั้งชื่อแฟ้มข้อมูลนั้นได้เองตามอัธยาศัย

3.3 พิมพ์ข้อมูลที่เก็บไว้ในแฟ้มข้อมูลออกทางเครื่องพิมพ์ (PRINT)

เป็นการนำค่าความถี่ที่เก็บไว้ในแฟ้มข้อมูลที่ผู้ใช้ต้องการมาพิมพ์ออกที่เครื่องพิมพ์ เพื่อความสะดวกในการวิเคราะห์ และ เปรียบเทียบ ทั้งยังมีลายลักษณ์อักษรเป็นหลักฐาน

3.4 ลบหน้าจอการวาดกราฟ (RESET)

เป็นการลบหน้าจอที่ใช้วาดกราฟ เมื่อผู้ใช้งานต้องการเริ่มต้นการรับค่าความถี่มาวาดกราฟใหม่อีกครั้ง ทำให้แลดูสวยงาม

3.5 ออกจากการทำงาน (QUIT)

เป็นการออกจากฉากการทำงานเพื่อไปสู่ฉากก่อนการเริ่มต้นการทำงาน ซึ่งอาจใช้ในการกำหนดเวลาการเปิดปิดเกทใหม่ หรือ จะยกเลิกการทำงาน

3.6 นำข้อมูลที่อยู่ในแฟ้มข้อมูลกลับมาวาดกราฟ (LOAD)

จะมีการให้ผู้ใช้ป้อนชื่อแฟ้มข้อมูลที่ต้องการก่อน จากนั้นก็จะนำความถี่ที่เก็บไว้กลับมาวาดกราฟอีกครั้ง เพื่อใช้ในการพิจารณาความมีเสถียรภาพอีกครั้งหนึ่ง จึงเหมาะแก่การนำมาวิเคราะห์

ในรายละเอียดของโปรแกรม ยังมีส่วนปลีกย่อยอีกมาก ที่ใช้ในการตรวจสอบความผิดพลาดต่างๆ ตลอดจน สามารถเลือกกำหนดค่าต่างๆที่มีผลต่อการแสดงผลได้อีกหลายแบบ เป็นต้น

ลักษณะของโปรแกรม มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/*                          PROGRAM OF                                   */
/*      FREQUENCY COUNTER INTERFACE WITH COMPUTER                        */
/*                          PRODUCT BY                                   */
/*          1. TAWEEKIAT TRITRANTHIP 34103127                            */
/*          2. TEERAYOT UDOMMANEETHANAKIT 34103157                       */
/*                          ENGINEER OF KMITL                            */
/*************************************************************************/

#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<alloc.h>
#include<key.h>
#include<dos.h>
#include<string.h>
#include<bios.h>
#include<process.h>
#include<math.h>
#include <stdlib.h>


#define NOTFOUND 5
#define ENTER 0
#define NOT_ENTER 1
#define TRUE  1
#define FALSE 0
#define CLIP_ON 1   /* activates clipping in viewport */


void opengraph(void),bottom(int x,int y,int s,char *ch),filename(void);
void show(int x1,int y1,int x2,int y2),createscreen1(void);
int huge detect256(void);
void createscreen2(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void spacetoselect(int mo,int om,int enter,int page);
void plotgraph(int x,int y,int z);
void title(void);
void cha();
void run(char *str,int lon,int x,int y,int color,int sou);
int page1(int page),page2(int page);
void kmit_song(void);
int readkey(void);
void select();
void start(void);
void choice(FILE *fp);
void chose();
void filename(void);
void cursor(int row,int col);
int getvpage(void);
void setgraph(int p);
void reset();

int k1;
float k2;
int control,w,e;

void main(void)
{
  char *ch;
  unsigned key = 0x0000;
  int y,maxx,maxy,x,space1 = 1,space2 = 1,mo[8],i,page=1;
  char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};


  title();
  opengraph();   // begin graphics
  do
     {
          switch(page)
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case 1:
              createscreen1();
              kmit_song();
              spacetoselect(1,1,NOT_ENTER,page);
              page = page1(page);
              if(page == 0)
                    key = Alt_Q;
            break;
            case 2:
              page = page2(page);
            break;
          }
    }
  while(key != Alt_Q);
  closegraph();
}

int huge detect256(void)
{
   return 4;
   /*
      0 =  320 * 200
      1 =  640 * 350
      2 =  640 * 480
      3 =  800 * 600
      4 = 1024 * 768
   */
}

void title(void)
{
  char name1[10] = "  FREQUEN";
  char name2[10] = "  COUNTER";
  int i,j,k;
  textmode(1);
  for (k=0 ; k<=6 ; k++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  clrscr();
  textcolor (k+1);
  for (i=0 ; i<=8 ; i++)
   {
       for (j=0 ; j<=i ; j++)
         {
           gotoxy(17-i+j,7);
           putch(name1[j]);
           gotoxy(20+i-j,7);
           putch(name2[8-j]);
         }
       sound(i*50+250);
       delay(50);
       nosound();
   }
  for (i=1 ; i<=7 ; i++)
   {
       gotoxy(18,i);
       cprintf("CY");
       sound(i*50+250);
       delay(50);
       nosound();
       gotoxy(18,i);
       cprintf("  ");
   }
  gotoxy(18,i-1);
  cprintf("CY");
}
for (i=0 ; i<=18 ; i++)
  {
    gotoxy(10,7);
    textcolor(LIGHTGRAY);
    cprintf(" FREQUENCY  COUNTER ");
    delay(100);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    run("BY  ",5,5,12,13,900);
    delay(700);
    run("    1. Taweekiat Tritranthip",29,5,12,9,700);
    delay(600);
    run("    2. Teerayot Udommaneethanakit",34,5,14,9,700);
    delay(600);
    run(" ... ENGINEER OF KMITL... ",28,7,18,12,700);
  gotoxy(5,24);
  delay(1500);
  textcolor(GREEN|BLINK);
  cprintf(" Press any key when you ready !");
  sound(300);
  delay(400);
  nosound();
  getch();
}

void run(char *str,int lon,int x,int y,int color,int sou)
{
 int i,j;
 textcolor(color);
 for(i=1;i<=lon;i++)
 {
  if (str[i-1]!=' ')
  {
   for(j=40;j>=x+i;j=j-1)
   {
    gotoxy(j,y);putch(str[i-1]);
    sound(sou);delay(10);nosound();
    gotoxy(j,y);putch(' ');
   }
  gotoxy(j,y);putch(str[i-1]);
  }
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void opengraph(void)
{
  int gd,gm,y,i;
  installuserdriver("svga256",detect256);
  gd = DETECT;
  initgraph(&gd,&gm,"c:\\borlandc\\bgi");
}

void createscreen1(void)
{
    char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
    char *ch;
    int x,y,maxx,maxy,xco,yco;
    maxx = getmaxx();
    maxy = getmaxy();

    setfillpattern(pattern,1);
    bar(0, 0,maxx, maxy);  // fill background with blue color

    setfillpattern(pattern,80);
    bar(22,22,maxx-22,50);
    setfillpattern(pattern,24);
    bar(22,51,maxx-20,54);
    bar(maxx-20,22,maxx-19,54);
    setfillpattern(pattern,82);
    bar(19,20,maxx-22,22);
    bar(19,20,22,54);          // draw title bar

    cha();
    setcolor(24);
    line(50,24,maxx-50,24);
    line(50,24,50,48);
    setcolor(15);
    line(50,48,maxx-50,48);
    line(maxx-50,48,maxx-50,24);
    setcolor(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    moveto((maxx/2)-35,33);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    outtextxy((maxx/2)-100,25,"FREQUENCY COUNTER");

    strcpy(ch,"LOAD");
    bottom(870,200,1,ch);  // bottom LOAD
    strcpy(ch,"QUIT");
    bottom(870,300,1,ch);  //  bottom QUIT
}

void cha()
{
    int xco,yco;
    xco=190;yco=160;
    moveto(xco,yco);
    settextstyle(GOTHIC_FONT,HORIZ_DIR,8);
    setcolor(YELLOW);
    highvideo();
    outtext("FREQUENCY");
    normvideo();
    xco=240;yco=270;
    moveto(xco,yco);
    highvideo();
    outtext("COUNTER");
    normvideo();
    xco=190;yco=380;
    moveto(xco,yco);
    highvideo();
    outtext("INTERFACE");
    normvideo();
    xco=100;yco=490;
    moveto(xco,yco);
    highvideo();
    outtext("WITH COMPUTER");
    normvideo();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void createscreen2(void)
{
   char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
   char *ch,str[10];
   int x,y,i,maxx,maxy;
   char num[10];

   maxx = getmaxx();
   maxy = getmaxy();

   clrscr();
   setfillpattern(pattern,1);
   bar(0, 0,maxx, maxy);      // fill background with blue color

   setfillpattern(pattern,80);
   bar(22,22,maxx-22,50);
   setfillpattern(pattern,24);
   bar(22,51,maxx-20,54);
   bar(maxx-20,22,maxx-19,54);
   setfillpattern(pattern,82);
   bar(19,20,maxx-22,22);
   bar(19,20,22,54);          // draw title bar

   setcolor(24);
   line(50,24,maxx-50,24);
   line(50,24,50,48);
   setcolor(15);
   line(50,48,maxx-50,48);
   line(maxx-50,48,maxx-50,24);
   setcolor(4);
   moveto((maxx/2)-35,33);
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
   outtextxy((maxx/2)-60,25,"DATA SHOW");

   strcpy(ch,"START");
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bottom(870,120,1,ch);  // bottom START
strcpy(ch,"SAVE");
bottom(870,210,1,ch);  // bottom SAVE
strcpy(ch,"PRINT");
bottom(870,300,1,ch);  // bottom PRINT
strcpy(ch,"RESET");
bottom(870,390,1,ch);   // bottom RESET
strcpy(ch,"QUIT");
bottom(870,480,1,ch);  //  bottom QUIT
strcpy(ch,"LOAD");
bottom(870,570,1,ch);  // bottom LOAD
show(25,130,770,560);
setcolor(RED);
moveto(363,100);
outtext("GRAPH");
line(87,150,87,540);
line(50,520,740,520);
for(i=1;i<=10;i++)
{
   moveto(85+62*i,508);
   outtext("|");
}
moveto(82,325);
outtext("-");
for(i=1;i<=10;i++)
{
   sprintf(str,"%d",i);
   outtextxy(80+62*i,526,str);
}
moveto(95,140);
outtext("FREQUENCY");
moveto(720,495);
outtext("TIME");
setcolor(14);
ellipse(155,668,0,360,100,75);
setcolor(11);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    sprintf(num,"%3.1f",k2);
    outtextxy(138,652,num);
    setcolor(10);
    moveto(111,612);outtext("GATETIME");
    moveto(120,692);outtext("SECOND");
    moveto(350,675);
    setcolor(14);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
    outtext("    ** SELECT A BOTTON AND PRESS ENTER ** ");
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
}

void show(int x1,int y1,int x2,int y2)
{
    char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

    setfillpattern(pattern,80);
    bar(x1-2,y1-35,x2+2,y1-2);  // fill background for barselect

    setfillpattern(pattern,2);
    bar(x1-2,y1-2,x2+2,y1);
    bar(x1-2,y1-2,x1,y2+2);
    bar(x1-2,y2,x2+2,y2+2);
    bar(x1+1,y1-2,x2+2,y2+2);

    setcolor(24);
    line(x1+30,y1-30,x2-30,y1-30);
    line(x1+30,y1-30,x1+30,y1-5);
    setcolor(15);
    line(x1+30,y1-5,x2-30,y1-5);
    line(x2-30,y1-5,x2-30,y1-30);

    setfillpattern(pattern,11);
    bar(x1,y1,x2,y2);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void bottom(int x,int y,int select,char *ch)
{
   char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

   setfillpattern(pattern,80);
   bar(x,y,x+70,y+30);
   if(select == 1)
    {
        setfillpattern(pattern,24);
        bar(x,y+31,x+72,y+32);
        bar(x+71,y+31,x+72,y+32);
        setfillpattern(pattern,15);
        bar(x-2,y-2,x-1,y+32);
        bar(x-2,y-2,x+72,y-1);
    }
   else    // select == 0
    {
        setfillpattern(pattern,15);
        bar(x,y+31,x+72,y+32);
        bar(x+71,y+31,x+72,y+32);
        setfillpattern(pattern,24);
        bar(x-2,y-2,x-1,y+32);
        bar(x-2,y-2,x+72,y-1);
    }
   setcolor(4);
   outtextxy(x+12,y+5,ch);
}

void spacetoselect(int method,int select,int enter,int page)
{
   char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
   int x[9],y[9];
   switch(page)
   {
     case 1:
        x[1] = 870; y[1] = 200; // load
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        x[2] = 870; y[2] = 300; // quit
    break;
    case 2:
        x[1] = 870; y[1] = 120; // start
        x[2] = 870; y[2] = 210; // save
        x[3] = 870; y[3] = 300; // print
        x[4] = 870; y[4] = 390; // reset
        x[5] = 870; y[5] = 480; // quit
        x[6] = 870; y[6] = 570; // load
    break;
}

    if(select == 0)  /* not select */
      {
            setfillpattern(pattern,80);
            bar(x[method],y[method],x[method]+70,y[method]+30);
            setcolor(4);
      }
    else  // select
      {
            setfillpattern(pattern,2);
            bar(x[method],y[method],x[method]+70,y[method]+30);
            setcolor(0);
      }
    switch(page)
     {
       case 1:
            if(method == 1)
              outtextxy(x[method]+12,y[method]+5,"LOAD");
            else if(method == 2)
              outtextxy(x[method]+12,y[method]+5,"QUIT");
       break;
       case 2:
            if(method == 1)
              outtextxy(x[method]+12,y[method]+5,"START");
            else if(method == 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                outtextxy(x[method]+12,y[method]+5,"SAVE");
              else if(method == 3)
                outtextxy(x[method]+12,y[method]+5,"PRINT");
              else if(method == 4)
                outtextxy(x[method]+12,y[method]+5,"RESET");
              else if(method == 5)
                outtextxy(x[method]+12,y[method]+5,"QUIT");
              else if(method == 6)
                outtextxy(x[method]+12,y[method]+5,"LOAD");
            break;
          }
        if(enter == NOT_ENTER)
        {
            setfillpattern(pattern,24);
            bar(x[method],y[method]+31,x[method]+72,y[method]+32);
            bar(x[method]+71,y[method]+31,x[method]+72,y[method]);
            setfillpattern(pattern,15);
            bar(x[method]-2,y[method]-2,x[method]-1,y[method]+32);
            bar(x[method]-2,y[method]-2,x[method]+72,y[method]-1);
        }
        else if(enter == ENTER)
        {
            setfillpattern(pattern,15);
            bar(x[method],y[method]+31,x[method]+72,y[method]+32);
            bar(x[method]+71,y[method]+31,x[method]+72,y[method]);
            setfillpattern(pattern,24);
            bar(x[method]-2,y[method]-2,x[method]-1,y[method]+32);
            bar(x[method]-2,y[method]-2,x[method]+72,y[method]-1);
        }
    }

    int page1(int page)
    {
        char ch,pl;
        char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
        unsigned key = 0x0000,key2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int space1 = 1,space2 = 1;
int midx, midy, i,j;

 midx = getmaxx() / 2;
 midy = getmaxy() / 2;

do
 {
      if(key == Alt_L)
       {
         spacetoselect(space1,0,NOT_ENTER,page);
         space1 = 1;
         spacetoselect(1,1,ENTER,page);
         delay(400);
         show(145,690,860,740);
         moveto(460,660);
         setcolor(RED);
         outtext("SET INPUT");
         setpl :
         setcolor(4);
         moveto(170,705);
         sound(800); delay(300); nosound();
         delay(250);
         outtext(" DO YOU WANT TO USE THE PRESCALER ? [ y/n ] : ");
         pl = getch();
         moveto(170,705);
         switch(pl)
          {
               case 'y':
                sound(800); delay(300); nosound();
                outtext(" DO YOU WANT TO USE THE PRESCALER ? [ y/n ] :  y ");
                delay(500);
                setcolor(80);
                moveto(460,660);
                outtext("SET INPUT");
                setcolor(11);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        moveto(170,705);
        outtext(" DO YOU WANT TO USE THE PRESCALER ?  [ y/n ] :   y ");
        e = 1;
       break;
       case 'n':
        sound(800); delay(300); nosound();
        outtext(" DO YOU WANT TO USE THE PRESCALER ?  [ y/n ] :   n ");
        delay(500);
        setcolor(80);
        moveto(460,660);
        outtext("SET INPUT");
        setcolor(11);
        moveto(170,705);
        outtext(" DO YOU WANT TO USE THE PRESCALER ?  [ y/n ] :   n ");
        e = 2;
       break;
       default :
        setcolor(11);
        moveto(170,705);
        outtext(" DO YOU WANT TO USE THE PRESCALER ?  [ y/n ] :  ");
        setcolor(RED);
        delay(400);
        sound(800); delay(300); nosound();
        moveto(170,705);
        outtext(" YOU MUST PRESS BUTTON  y OR n .... TRY AGAIN ! ");
        delay(1700);
        setcolor(11);
        moveto(170,705);
        outtext(" YOU MUST PRESS BUTTON  y OR n .... TRY AGAIN ! ");
        goto setpl;
 }
delay(300);
for (i=0; i<250; i=i+15)
  {
        setfillpattern(pattern,80);
        for(j=0; j<150; j=j+15)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            bar(midx-i, midy-j, midx+i,midy+j);
        }
        setcolor(14);
        setlinestyle(SOLID_LINE,0,THICK_WIDTH);
        rectangle(midx-240, midy-135, midx+240,midy+135);
        setlinestyle(SOLID_LINE,0,NORM_WIDTH);
        setcolor(0);
        outtextxy(midx-25,midy-110,"DATA");
        setcolor(6);
        outtextxy(midx-220,midy-50," a :    FAST , GATE TIME  0.1  SECOND");
        outtextxy(midx-220,midy," b :  NORMAL , GATE TIME  1.0  SECOND");
        outtextxy(midx-220,midy+50," c :    SLOW , GATE TIME  10.0  SECOND");
        sound(800); delay(300); nosound();
        delay(200);
        moveto(460,660);
        setcolor(RED);
        outtext("SET DATA");
        moveto(170,705);
        outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :");
        ch = getch();
        moveto(170,705);
        switch(ch)
        {
            case 97: outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  a");
break;
            case 98: outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  b");
break;
            case 99: outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  c");
break;
        }
        delay(300);
        if (ch == 'a')
        {
          setcolor(11);
          moveto(170,705);
          outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  a");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  setcolor(RED);
  moveto(170,705);
  sound(800); delay(300); nosound();
  outtext (" YOU SELECT CHOICE  ..a..");
  delay(1800);
 }
else  if (ch == 'b')
 {
  setcolor(11);
  moveto(170,705);
  outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  b");
  setcolor(RED);
  moveto(170,705);
  sound(800); delay(300); nosound();
  outtext (" YOU SELECT CHOICE  ..b..");
  delay(1800);
 }
else  if (ch == 'c')
 {
       setcolor(11);
       moveto(170,705);
       outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  c");
       setcolor(RED);
       moveto(170,705);
       sound(800); delay(300); nosound();
       outtext(" YOU SELECT CHOICE  ..c..");
       delay(1800);
 }
 else
 {
       setcolor(11);
       moveto(170,705);
       outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  ");
       setcolor(RED);
       moveto(170,705);
       sound(800); delay(300); nosound();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        outtext (" YOU MUST PRESS BOTTON  a,b OR c ... TRY AGAIN !!!");
        delay (1800);
        select();
    }
    sound(800); delay(300); nosound();
    delay(300);
    spacetoselect(1,1,NOT_ENTER,page);
    delay(300);
    switch(ch) /*select gate*/
     {
        case'a':
        control=0xFF;
        k1=1;
        k2=0.1;
        break;
        case'b':
        control=0xF7;
        k1=2;
        k2=1.0;
        break;
        case'c':
        control=0xFB;
        k1=3;
        k2=10.0;
        break;
     }
    cleardevice();
    createscreen2();
    page = 2;
    spacetoselect(1,1,NOT_ENTER,page);  // OPEN bottom
    return(2);
  }
  else if(key == Down_Key)
    {
        space1++;
        if(space1 == 3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
              space1 = 1;
              space2 = 2;
            }
          else
           {
              space2 = space1 - 1;
            }
          spacetoselect(space2,0,NOT_ENTER,page);
          spacetoselect(space1,1,NOT_ENTER,page);
      }
    else if(key == Up_Key)
      {
       space1--;
       if(space1 == 0)
            {
             space1 = 2;
             space2 = 1;
            }
       else
            {
              space2 = space1 + 1;
            }
      spacetoselect(space2,0,NOT_ENTER,page);
      spacetoselect(space1,1,NOT_ENTER,page);
      }
   key=bioskey(0);
   if(key == Enter_Key)
    {
      if(space1 == 1)
        key = Alt_L;
      else if(space1 == 2)
        key = Alt_Q;
    }
 }
while(key != Alt_Q);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 spacetoselect(space1,0,NOT_ENTER,page);
 spacetoselect(2,1,ENTER,page);
 delay(500);
 spacetoselect(2,1,NOT_ENTER,page);
 delay(500);
 return(0);
}

int page2(int page)
{
      unsigned key = 0x0000;
      char str[10],ch,num[10],vg[10],stl[10];
      int space1 = 1,space2 = 1,i;
      int maxx,maxy,x,y,v,l,o,u,c;
      char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
      char sname[50],lname[50],pname[50];
      int midx, midy, t,j;
      long g,n,d,r,z,h,m,n1,m1;
      FILE *fp,*sfp,*pfp,*lfp;

          midx = (getmaxx() / 2)-93;
          midy = (getmaxy() / 2)-45;
          maxx = getmaxx();
          maxy = getmaxy();

      do
          {
                setcolor(BLUE);
                settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
                moveto(350,657);
                outtext(" NOW !!   FREQUENCY IS :                    ");
                moveto(440,715);
                outtext (" ... GOOD LUCK AND GOOD BYE ... ");
                moveto(350,675);
                setcolor(14);
                outtext("    ** SELECT A BOTTON AND PRESS ENTER ** ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
if(key == Alt_T)
 {
   spacetoselect(space1,0,NOT_ENTER,page);
   space1 = 1;
   spacetoselect(1,1,ENTER,page);
   delay(300);
   moveto(350,675);
   setcolor(BLUE);
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
   outtext("    ** SELECT A BOTTON AND PRESS ENTER ** ");
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
   setgraph(1);
   sound(800); delay(300); nosound();
   setviewport(midx-195, midy-150, midx+195,midy+150,CLIP_ON);
   clearviewport();
   setcolor(11);
   rectangle(0,0,390,300);
   setfillstyle(SOLID_FILL,11);
   floodfill(359,255,LIGHTCYAN);
   setviewport(0,0,maxx,maxy,CLIP_ON);
   delay(200);
   setcolor(14);
   moveto(350,657);
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
   outtext(" NOW !!  FREQUENCY IS :      ");
   sound(800); delay(300); nosound();
   delay(200);
   moveto(350,715);
   settextstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
   sound(800); delay(300); nosound();
   outtext(" Press any key when you want to stop the work ");
   start();
   spacetoselect(1,1,NOT_ENTER,page);
 }
 else if(key == Alt_S)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
spacetoselect(space1,0,NOT_ENTER,page);
space1 = 2;
spacetoselect(2,1,ENTER,page);
delay(300);
moveto(350,675);
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtext("    ** SELECT A BOTTON AND PRESS ENTER ** ");
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
filename();
sound(950); delay(300); nosound();
cursor(43.5,67);
scanf("%s",sname);
x = 385 ; y = 657;
setfillstyle(1,1);
bar(x-2,y-2,x+502,y+52);
if((fp=fopen("name.dat","rb"))==NULL)
  {
   int x = 385 ,y = 657;
   setfillstyle(1,80);
   bar(x,y,x+500,y+50);
   setfillstyle(1,24);
   bar(x,y+51,x+502,y+52);
   bar(x+501,y+51,x+502,y);
   setfillstyle(1,82);
   bar(x-2,y-2,x-1,y+51);
   bar(x-2,y-2,x+502,y-1);
   moveto(485,668);
   setcolor(4);
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
   delay(400);
   outtext(" ** ERROR IN READING DATA ** ");
   sound(950); delay(300); nosound();
   delay(1500);
   x = 385 ; y = 657;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setfillstyle(1,1);
        bar(x-2,y-2,x+502,y+52);
        delay(400);
        spacetoselect(2,1,NOT_ENTER,page);
      }
    else if((sfp=fopen(sname,"wb"))==NULL)
      {
        int x = 385 ,y = 657;
        setfillstyle(1,80);
        bar(x,y,x+500,y+50);
        setfillstyle(1,24);
        bar(x,y+51,x+502,y+52);
        bar(x+501,y+51,x+502,y);
        setfillstyle(1,82);
        bar(x-2,y-2,x-1,y+51);
        bar(x-2,y-2,x+502,y-1);
        moveto(490,668);
        setcolor(4);
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
        delay(400);
        outtext(" ** ERROR IN WRITING FILE ** ");
        sound(950); delay(300); nosound();
        delay(1500);
        x = 385 ; y = 657;
        setfillstyle(1,1);
        bar(x-2,y-2,x+502,y+52);
        fclose(fp);
        delay(400);
        spacetoselect(2,1,NOT_ENTER,page);
      }
    else
      {
        do
          {
            if(e==2)
              {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fread(&n,sizeof(long),1,fp);
                n1 = n;
                fwrite(&n1,sizeof(long),1,sfp);
            }
            if(e==1)
            {
                fread(&m,sizeof(long),1,fp);
                m1 = m;
                fwrite(&m1,sizeof(long),1,sfp);
            }
        }while(!feof(fp));
        fclose(fp);
        fclose(sfp);
        sound(950); delay(300); nosound();
        delay(1000);
        spacetoselect(2,1,NOT_ENTER,page);
    }
}
else if(key == Alt_R)
{
    spacetoselect(space1,0,NOT_ENTER,page);
    space1 = 4;
    spacetoselect(4,1,ENTER,page);
    moveto(350,675);
    setcolor(BLUE);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
    outtext("    ** SELECT A BOTTON AND PRESS ENTER ** ");
    delay(300);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    setviewport(93,160,715,510,CLIP_ON);
    clearviewport();
    setcolor(11);
    rectangle(0,0,718,510);
    setfillstyle(SOLID_FILL,11);
    floodfill(359,255,LIGHTCYAN);
    setviewport(90,530,760,550,CLIP_ON);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,670,20);
floodfill(317.5,10,LIGHTCYAN);
setviewport(26,322,83,335,CLIP_ON);
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,57,13);
floodfill(28.5,6.5,LIGHTCYAN);
sound(950); delay(300); nosound();
setviewport(0,0,maxx,maxy,CLIP_ON);
for(i=1;i<=10;i++)
 {
        setcolor(4);
        sprintf(str,"%d",i);
        outtextxy(80+62*i,526,str);
 }
 delay(400);
 spacetoselect(4,1,NOT_ENTER,page);
}
else if(key == Alt_P)
{
 spacetoselect(space1,0,NOT_ENTER,page);
 space1 = 3;
 spacetoselect(3,1,ENTER,page);
 delay(300);
 moveto(350,675);
 setcolor(BLUE);
 settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
 outtext("   ** SELECT A BOTTON AND PRESS ENTER ** ");
 settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
 filename();
 sound(950); delay(300); nosound();
 cursor(43.5,67);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
scanf("%s",pname);
x = 385 ; y = 657;
setfillstyle(1,1);
bar(x-2,y-2,x+502,y+52);
if((pfp=fopen(pname,"rb"))==NULL)
 {
    int x = 385 ,y = 657;
    setfillstyle(1,80);
    bar(x,y,x+500,y+50);
    setfillstyle(1,24);
    bar(x,y+51,x+502,y+52);
    bar(x+501,y+51,x+502,y);
    setfillstyle(1,82);
    bar(x-2,y-2,x-1,y+51);
    bar(x-2,y-2,x+502,y-1);
    moveto(527,668);
    setcolor(4);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    delay(400);
    outtext(" ** FILE NOT FOUND ** ");
    sound(950); delay(300); nosound();
    delay(1300);
    x = 385 ; y = 657;
    setfillstyle(1,1);
    bar(x-2,y-2,x+502,y+52);
    delay(400);
 }
else
 {
   i=1;
   do
       {
          if(e==2)
          {
             fread(&n,sizeof(long),1,pfp);
             fprintf(stdprn,"[%d] = %ld\n",i,n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            if(e==1)
            {
              fread(&m,sizeof(long),1,pfp);
              fprintf(stdprn,"[%d] = %ld\n",i,m);
            }
            i++;
          }while(!feof(pfp));
        delay(400);
        int x = 385 ,y = 657;
        setfillstyle(1,80);
        bar(x,y,x+500,y+50);
        setfillstyle(1,24);
        bar(x,y+51,x+502,y+52);
        bar(x+501,y+51,x+502,y);
        setfillstyle(1,82);
        bar(x-2,y-2,x-1,y+51);
        bar(x-2,y-2,x+502,y-1);
        moveto(545,668);
        setcolor(4);
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
        delay(400);
        outtext(" ** END OF FILE ** ");
        sound(950); delay(300); nosound();
        delay(700);
        x = 385 ; y = 657;
        setfillstyle(1,1);
        bar(x-2,y-2,x+502,y+52);
        fclose(pfp);
        delay(600);
      }
      spacetoselect(3,1,NOT_ENTER,page);
    }
    else if(key == Alt_L)
    {
      spacetoselect(space1,0,NOT_ENTER,page);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
space1 = 6;
spacetoselect(6,1,ENTER,page);
delay(300);
moveto(350,675);
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtext("   ** SELECT A BOTTON AND PRESS ENTER ** ");
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
filename();
sound(950); delay(300); nosound();
cursor(43.5,67);
scanf("%s",lname);
x = 385 ; y = 657;
setfillstyle(1,1);
bar(x-2,y-2,x+502,y+52);
if((lfp=fopen(lname,"rb"))==NULL)
 {
   int x = 385 ,y = 657;
   setfillstyle(1,80);
   bar(x,y,x+500,y+50);
   setfillstyle(1,24);
   bar(x,y+51,x+502,y+52);
   bar(x+501,y+51,x+502,y);
   setfillstyle(1,82);
   bar(x-2,y-2,x-1,y+51);
   bar(x-2,y-2,x+502,y-1);
   moveto(490,668);
   setcolor(4);
   settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
   delay(400);
   outtext(" ** ERROR IN READING FILE ** ");
   sound(950); delay(300); nosound();
   delay(1500);
   x = 385 ; y = 657;
   setfillstyle(1,1);
   bar(x-2,y-2,x+502,y+52);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(400);
    }
  else
    {
        setgraph(2);
        c=0;
        reset();
        do
            {
              for(i=0+c;i<10+c;i++)
                {
                 if(e==2)
                 fread(&n,sizeof(long),1,lfp);
                 if(e==1)
                 fread(&m,sizeof(long),1,lfp);
                 if((i%10)==0)
                  {
                        setcolor(RED);
                        setusercharsize(1, 3, 1, 3);
                        if(e==2)      sprintf(vg,"%ld",n);
                        if(e==1)      sprintf(vg,"%ld",m);
                        outtextxy(29,322,vg);
                        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
                        outtextxy(147,328-5,".");
                  }
                 else
                  {
                        if(e==2)    d = n - g;
                        if(e==1)    d = m - g;
                        if(w==1)    v = 328 - (175*d)/1000 ;
                        if(w==2)    v = 328 - (175*d)/5000 ;
                        if(w==3)    v = 328 - (175*d)/10000 ;
                        if(w==4)    v = 328 - (175*d)/50000 ;
                  }
                l = i%10;
                if(l>0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        outtextxy(147+(62*l),v-5,".");
    }
delay(100);
if(l==1)
        line(147,341,209,v + 13);
if(l>1)
 {
        if(w==1)
          line(147+(62*(l-1)),341-(175*z)/1000,
                        147+(62*l),v +13);
        if(w==2)
          line(147+(62*(l-1)),341-(175*z)/5000,
                        147+(62*l),v +13);
        if(w==3)
          line(147+(62*(l-1)),341-(175*z)/10000,
                        147+(62*l),v +13);
        if(w==4)
          line(147+(62*(l-1)),341-(175*z)/50000,
                        147+(62*l),v +13);
 }
 z = d;
 if(l==0)
  {
        if(e==2)    g = n;
        if(e==1)    g = m;
  }
}

int x = 385 ,y = 657;
setfillstyle(1,80);
bar(x,y,x+500,y+50);
setfillstyle(1,24);
bar(x,y+51,x+502,y+52);
bar(x+501,y+51,x+502,y);
setfillstyle(1,82);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bar(x-2,y-2,x-1,y+51);
bar(x-2,y-2,x+502,y-1);
moveto(470,668);
setcolor(4);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
delay(400);
outtext(" ** PRESS ANY KEY TO CONTINUE ** ");
sound(950); delay(300); nosound();
getch();
x = 385 ; y = 657;
setfillstyle(1,1);
bar(x-2,y-2,x+502,y+52);
delay(400);
setviewport(93,160,715,510,CLIP_ON);
clearviewport();
setcolor(11);
rectangle(0,0,718,510);
setfillstyle(SOLID_FILL,11);
floodfill(359,255,LIGHTCYAN);
setviewport(90,530,725,550,CLIP_ON);
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,635,20);
floodfill(317.5,10,LIGHTCYAN);
setviewport(26,322,83,335,CLIP_ON);
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,57,13);
floodfill(28.5,6.5,LIGHTCYAN);
setviewport(0,0,maxx,maxy,CLIP_ON);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
setcolor(RED);
c=i;
o=i+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for(i=0+o;i<10+o;i++)
                {
                    u=i%10;
                    sprintf(stl,"%d",i);
                    if (u!=0)
                      outtextxy(78+(62*u),526,stl);
                    else
                      outtextxy(698,526,stl);
                }
              }while(!feof(lfp));
            delay(500);
            int x = 385 ,y = 657;
            setfillstyle(1,80);
            bar(x,y,x+500,y+50);
            setfillstyle(1,24);
            bar(x,y+51,x+502,y+52);
            bar(x+501,y+51,x+502,y);
            setfillstyle(1,82);
            bar(x-2,y-2,x-1,y+51);
            bar(x-2,y-2,x+502,y-1);
            moveto(545,668);
            setcolor(4);
            settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
            delay(400);
            outtext(" ** END OF FILE ** ");
            sound(950); delay(300); nosound();
            delay(1000);
            x = 385 ; y = 657;
            setfillstyle(1,1);
            bar(x-2,y-2,x+502,y+52);
            fclose(lfp);
            delay(500);
          }
         spacetoselect(6,1,NOT_ENTER,page);
        }
        else if(key == Down_Key)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
         space1++;
         if(space1 == 7)
          {
                space1 = 1;
                space2 = 6;
          }
         else
          {
                space2 = space1 - 1;
          }
         spacetoselect(space2,0,NOT_ENTER,page);
         spacetoselect(space1,1,NOT_ENTER,page);
        }
else if(key == Up_Key)
        {
         space1--;
         if(space1 == 0)
          {
                space1 = 6;
                space2 = 1;
          }
         else
          {
                space2 = space1 + 1;
          }
         spacetoselect(space2,0,NOT_ENTER,page);
         spacetoselect(space1,1,NOT_ENTER,page);
        }
key=bioskey(0);
if(key == Enter_Key)
        {
         if(space1 == 1)
         key = Alt_T;
         else if(space1 == 2)
         key = Alt_S;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    else if(space1 == 3)
                    key = Alt_P;
                    else if(space1 == 4)
                    key = Alt_R;
                    else if(space1 == 5)
                    key = Alt_Q;
                    else if(space1 == 6)
                    key = Alt_L;
                   }
          } while(key != Alt_Q);
     spacetoselect(space1,0,NOT_ENTER,page);
     space1 = 5;
     spacetoselect(5,1,ENTER,page);
     delay(430);
     spacetoselect(5,1,NOT_ENTER,page);
     delay(300);
     return(1);
}

void kmit_song(void)
{
#define  C  262
#define  D  294
#define  E  330
#define  F  350
#define  G  392
#define  A  440
#define  B  467
#define  CH 524
#define  DH 588
#define  EH 660
#define  FH 700
#define  GH 785
#define  AH 880
#define  BH 934
#define  REST 25000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  speed 1.556
#define  QQE 937.5
#define  Q 375
#define  EN 187.5
#define  ES 287.25
#define  S 93.75
#define  EP 187.5
#define  QP 375
#define  BLANK 5
#define MAXNOTE 109
        unsigned int key=0x0000;
        int i=0;
        char dummy;
        int melody[] = {A,DH,CH,A,G,F,D,G,F,D,C,A,DH,CH,G,A,DH,CH,FH,DH,A,CH,F,G,A,G,
                                C,A,G,F,D,C,F,G,A,G,A,CH,G,A,G,A,DH,CH,A,REST,A,G,D,F,
                                B,CH,B,CH,REST,CH,REST,DH,CH,B,CH,A,G,F,
REST,G,REST,G,REST,G,C,G,CH,A,G,E,D,E,G,E,D,C,REST,

        A,DH,CH,A,G,F,D,G,F,D,C,A,DH,CH,G,A,DH,CH,FH,DH,A,CH,A,G,D,F};
        int del[] = {EN,EN,QQE,EN,EN,EN,EN,EN,EN,EN,Q,EN,EN,ES,S,EN,EN,EN,
EN,EN,EN,EN,EN,EN,EN,Q,
                                EN,EN,QQE,EN,EN,EN,ES,S,ES,S,Q,EN,EN,ES,S,
EN,EN,EN,S,BLANK,S,EN,EN,QQE,

        ES,S,EN,Q,20,Q,EP,ES,S,EN,EN,EN,EN,EN,EP,S,BLANK,S,BLANK,EN,EN,EN,EN,EN,EN,ES,S,
EN,EN,EN,EN,Q,QP,

        EN,EN,QQE,EN,EN,EN,EN,EN,EN,EN,Q,EN,EN,ES,S,EN,EN,EN,EN,EN,EN,
EN,EN,EN,EN,QQE};
        do
        {
                sound(melody[i]);
                delay(del[i]*speed);
                if(i<MAXNOTE) i++;
                if(i==MAXNOTE)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    nosound();
                    sleep(2);
                    i=0;
                }
                key=readkey();
        }while(key == 0xffff);
        nosound();
}

int readkey(void)
{
        unsigned int scrsave_c1=0,scrsave_c2=0;
        _AH=0x11;
        geninterrupt(0x16);
        if(!(_FLAGS&64))
        {
                scrsave_c1=scrsave_c2=0;
                _AH=0x10;
                geninterrupt(0x16);
                return(_AX);
        }
        scrsave_c1++;
        if(scrsave_c1==5500) {scrsave_c1=0;scrsave_c2++;}
        return 0xffff;
}

void select()
{
  char ch;
     setcolor(11);
     moveto(170,705);
     sound(800); delay(300); nosound();
     outtext (" YOU MUST PRESS BOTTON  a,b OR c ... TRY AGAIN !!!");
     setcolor(RED);
     moveto(170,705);
     outtext(" SELECT  GATETIME  CHOICE  a = 0.1s , b = 1s , c = 10s  :  ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ch = getch();
moveto(170,705);
switch(ch)
    {
      case 97: outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : a"); break;
      case 98: outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : b"); break;
      case 99: outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : c"); break;
    }
delay(300);
if (ch == 'a')
     {
       setcolor(11);
       moveto(170,705);
       outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : a");
       setcolor(RED);
       moveto(170,705);
       sound(800); delay(300); nosound();
       outtext (" YOU SELECT CHOICE ..a..");
       delay(1800);
     }
else if (ch == 'b')
    {
       setcolor(11);
       moveto(170,705);
       outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : b");
       setcolor(RED);
       moveto(170,705);
       sound(800); delay(300); nosound();
       outtext (" YOU SELECT CHOICE ..b..");
       delay(1800);
    }
else if (ch == 'c')
    {
       setcolor(11);
       moveto(170,705);
       outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s : c");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setcolor(RED);
        moveto(170,705);
        sound(800); delay(300); nosound();
        outtext (" YOU SELECT CHOICE ..c..");
        delay(1800);
    }
else
    {
        setcolor(11);
        moveto(170,705);
        outtext(" SELECT GATETIME CHOICE a = 0.1s , b = 1s , c = 10s :  ");
        setcolor(RED);
        moveto(170,705);
        sound(800); delay(300); nosound();
        outtext (" YOU MUST PRESS BOTTON a,b OR c ... TRY AGAIN !!!");
        delay (1800);
        select();
    }
   switch(ch) /*select gate*/
        {
            case'a':
            control=0xFF;
            k1=1;
            k2=0.1;
            break;
            case'b':
            control=0xF7;
            k1=2;
            k2=1.0;
            break;
            case'c':
            control=0xFB;
            k1=3;
            k2=10.0;
            break;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void start(void)
{
    int p,q,r,j,lz;  /*programe error selectgate divide time */
    int i=7;
    int data[9];
    char ch,num[10],mum[10],psl[20],npl[10];
    char str[10],sav[10],*sno;
    int x,y,t,maxx,maxy,v,a,b,u,h,o,l ;
    long  n,g,d,z,m;
    FILE *fp;

    maxx = getmaxx();
    maxy = getmaxy();
    a=0;
    b=0;
    d=0;

    if((fp=fopen("name.dat","wb"))==NULL)
        {
            int x = 385 ,y = 657;
            setfillstyle(1,80);
            bar(x,y,x+500,y+50);
            setfillstyle(1,24);
            bar(x,y+51,x+502,y+52);
            bar(x+501,y+51,x+502,y);
            setfillstyle(1,82);
            bar(x-2,y-2,x-1,y+51);
            bar(x-2,y-2,x+502,y-1);
            moveto(490,668);
            setcolor(4);
            settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
            delay(400);
            outtext(" ** ERROR IN OPEN FILE ** ");
            sound(950); delay(300); nosound();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(1500);
        x = 385 ; y = 657;
        setfillstyle(1,1);
        bar(x-2,y-2,x+502,y+52);
        delay(400);
    }
else
{
    reset();
    outportb(0x307,0x99);
    outportb(0x303,0x9B);
    do
     {
       for(t=0+a;t<10+a;t++)
        {
          start:
          control=control & 0xFE;      //remove clear
          outportb(0x305,control);
          p=0x00;
          q=0x01;
          r=0x00;
          while(p!=0x01)
          {
                p=inportb(0x306);
                p=p&0x01;
          }
          control=control & 0xFD;   //open gate
          outportb(0x305,control);
          while(q!=0x00)
          {
                q=inportb(0x306);
                q=q&0x01;
          }
          control=control | 0x02;   //close gate
          outportb(0x305,control);
          while(r!=0x01)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    r=inportb(0x306);
    r=r&0x01;
}
j=inportb(0x300);
data[0]=j&0x0F;
data[1]=j>>4;
j=inportb(0x301);
data[3]=j&0x0F;
data[2]=j>>4;
j=inportb(0x302);
data[4]=j&0x0F;
data[5]=j>>4;
j=inportb(0x304);
data[6]=j&0x0F;
data[7]=j>>4;
control=control | 0x01;   //clear counter
outportb(0x305,control);
lz=FALSE;
settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
setfillstyle(1,1);
bar(660,660,950,686);
setcolor(11);
if(c==1)  setcolor(1);
if((t%10)!=0)
{
if(k1==2)
{
    for(i=7;i>0;i--)
    if((data[i]==0)&(!lz))
    {
      moveto(683,660);
      outtext(" ");
    }
     else
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sprintf(num,"%d",data[i]);
        outtextxy(788-15*i,660,num);
        if(i==6)
        outtextxy(709,660,",");
        if(i==3)
        outtextxy(754,660,",");
        lz=TRUE;
      }
      sprintf(mum,"%d  Hertz",data[0]);
      outtextxy(788,660,mum);
      sprintf(sno,"%d%d%d%d%d%d%d%d%d",data[7],data[6],data[5],
                  data[4],data[3],data[2],data[1],data[0]);
}
if(k1==1)
{
      for(i=6;i>=0;i--)
      if((data[i]==0)&(!lz))
      {
       moveto(683,660);
       outtext(" ");
      }
      else
      {
        sprintf(num,"%d",data[i]);
        outtextxy(773-15*i,660,num);
        if(i==5)
        outtextxy(709,660,",");
        if(i==3)
        outtextxy(754,660,",");
        lz=TRUE;
      }
      data[8]=0;
      sprintf(mum,"%d  Hertz",data[8]);
      outtextxy(788,660,mum);
      sprintf(sno,"%d%d%d%d%d%d%d%d%d",data[6],data[5],data[4],
                  data[3],data[2],data[1],data[0],data[8]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
if(k1==3)
{
    for(i=7;i>=2;i--)
    if((data[i]==0)&(!lz))
    {
      moveto(683,660);
      outtext(" ");
    }
    else
    {
      sprintf(num,"%d",data[i]);
      outtextxy(803-15*i,660,num);
      if(i==7)
      outtextxy(709,660,",");
      if(i==4)
      outtextxy(754,660,",");
      lz=TRUE;
    }
    sprintf(mum,"%d  Hertz",data[1]);
    outtextxy(788,660,mum);
    sprintf(sno,"%d%d%d%d%d%d%d",data[7],data[6],data[5],
                        data[4],data[3],data[2],data[1]);
}
if(e==1)
{
    setcolor(11);
    n = atol(sno);
    m = n*256;
    sprintf(psl,"%ld",m);
    outtextxy(683,660,psl);
    outtextxy(850,660,"Hertz");
}
if((t%10)==1 && w==1)
{
  if(e==2)    n = atol(sno);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
else if((t%10)==1 && w!=1)
{
  setcolor(RED);
  setusercharsize(1, 3, 1, 3);
  if(e==2)
      {
        n = atol(sno);
        sprintf(npl,"%ld",n);
        outtextxy(29,322,npl);
      }
  if(e==1)     outtextxy(29,322,psl);
  outtextxy(147,328,".");
 }
else
 {
  if(e==2)
  {
          n = atol(sno);
          d = n - g;
  }
  if(e==1)          d = m - g;
  if(w==2)       v = 328 - (175*d)/1000 ;
  if(w==3)       v = 328 - (175*d)/5000 ;
  if(w==4)       v = 328 - (175*d)/10000 ;
 }
setcolor(4);
l = t%10;
if(l>1 && w!=1)
 {
  outtextxy(147+(62*(l-1)),v-7,".");
 }
if(l==2 && w!=1)  line(147,338,209,v + 10);
if(l>2 && w!=1)
 {
  if(w==2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    line(147+(62*(l-2)),338-(175*z)/1000,147+(62*(l-1)),v + 10);
    if(w==3)
    line(147+(62*(l-2)),338-(175*z)/5000,147+(62*(l-1)),v + 10);
    if(w==4)
    line(147+(62*(l-2)),338-(175*z)/10000,147+(62*(l-1)),v + 10);
   }
  z = d;
  if(l==1)
   {
    if(c==2)   g = n;
    if(c==1)   g = m;
   }
  if(c==2)
    fwrite(&n,sizeof(long),1,fp);
  if(c==1)
    fwrite(&m,sizeof(long),1,fp);
 }
 }
delay(1000);
setviewport(93,160,715,510,CLIP_ON);
clearviewport();
setcolor(11);
rectangle(0,0,718,510);
setfillstyle(SOLID_FILL,11);
floodfill(359,255,LIGHTCYAN);
setviewport(90,530,760,550,CLIP_ON);
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,670,20);
floodfill(317.5,10,LIGHTCYAN);
setviewport(26,322,83,335,CLIP_ON);
clearviewport();
setcolor(11);
setfillstyle(SOLID_FILL,11);
rectangle(0,0,57,13);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        floodfill(28.5,6.5,LIGHTCYAN);
        setviewport(0,0,maxx,maxy,CLIP_ON);
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
        setcolor(RED);
        a=t;
        o=t+1;
        for(i=0+o;i<10+o;i++)
         {
           u=i%10;
           sprintf(str,"%d",i);
           if (u!=0) outtextxy(78+(62*u),526,str);
           else   outtextxy(698,526,str);
         }
      }while(!kbhit());
  }
getch();
setcolor(BLUE);
setfillstyle(1,1);
bar(660,660,950,686);
moveto(350,715);
settextstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
outtext(" Press any key when you want to stop the work ");
moveto(350,715);
setcolor(YELLOW);
sound(800); delay(300); nosound();
outtext(" Do you want to stop the work ? [ y/n ] : ");
ch = getch();
moveto(350,715);
switch(ch)
    {
        case 'y':
          setcolor(YELLOW);
          sound(800); delay(300); nosound();
          outtext(" Do you want to stop the work ? [ y/n ] :  y ");
          delay(1000);
          setcolor(BLUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        moveto(350,715);
        outtext(" Do you want to stop the work ? [ y/n ] :  y ");
        setcolor(YELLOW);
        moveto(440,715);
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
        sound(800); delay(300); nosound();
        outtext (" ... GOOD LUCK AND GOOD BYE ... ");
        delay(1000);
        fclose(fp);
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
      break;
      case 'n':
        setcolor(YELLOW);
        sound(800); delay(300); nosound();
        outtext(" Do you want to stop the work ? [ y/n ] :  n ");
        delay(1000);
        setcolor(BLUE);
        moveto(350,715);
        outtext(" Do you want to stop the work ? [ y/n ] :  n ");
        setcolor(YELLOW);
        moveto(350,715);
        sound(800); delay(300); nosound();
        outtext (" Press any key to continue the work !! ");
        getch();
        setcolor(BLUE);
        moveto(350,715);
        outtext (" Press any key to continue the work !! ");
        moveto(350,715);
        setcolor(YELLOW);
        settextstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
        sound(800); delay(300); nosound();
        outtext(" Press any key when you want to stop the work ");
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
        reset();
        goto start;
      break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            default:
              setcolor(BLUE);
              moveto(350,715);
              outtext(" Do you want to stop the work ? [ y/n ] :   ");
              delay(400);
              setcolor(YELLOW);
              moveto(350,715);
              sound(800); delay(300); nosound();
              outtext (" You must press button  y or n ... TRY AGAIN !!!");
              delay (2000);
              choice(fp);
            break;
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    }
}

void choice(FILE *fp)
{
   char ch;

    setcolor(BLUE);
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
    moveto(350,715);
    outtext (" You must press button  y or n ... TRY AGAIN !!!");
    moveto(350,715);
    setcolor(YELLOW);
    sound(800); delay(300); nosound();
    outtext(" Do you want to stop the work ? [ y/n ] :   ");
    ch = getch();
    moveto(350,715);
    switch(ch)
      {
          case 'y':
            setcolor(YELLOW);
            sound(800); delay(300); nosound();
            outtext(" Do you want to stop the work ? [ y/n ] :  y ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delay(1000);
    setcolor(BLUE);
    moveto(350,715);
    outtext(" Do you want to stop the work ? [ y/n ] :  y ");
    setcolor(YELLOW);
    moveto(440,715);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
    sound(800); delay(300); nosound();
    outtext (" ... GOOD LUCK AND GOOD BYE ... ");
    delay(1000);
    fclose(fp);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
  break;
  case 'n':
    setcolor(YELLOW);
    sound(800); delay(300); nosound();
    outtext(" Do you want to stop the work ? [ y/n ] :  n ");
    delay(1000);
    setcolor(BLUE);
    moveto(350,715);
    outtext(" Do you want to stop the work ? [ y/n ] :  n ");
    setcolor(YELLOW);
    moveto(350,715);
    sound(800); delay(300); nosound();
    outtext (" Press any key to continue the work !! ");
    getch();
    setcolor(BLUE);
    moveto(350,715);
    outtext (" Press any key to continue the work !! ");
    moveto(350,715);
    setcolor(YELLOW);
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,3);
    sound(800); delay(300); nosound();
    outtext(" Press any key when you want to stop the work ");
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    start();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
            default:
              setcolor(BLUE);
              moveto(350,715);
              outtext(" Do you want to stop the work ? [ y/n ] :  ");
              delay(400);
              setcolor(YELLOW);
              moveto(350,715);
              sound(800); delay(300); nosound();
              outtext (" You must press button  y or n ... TRY AGAIN !!!");
              delay (2000);
              choice(fp);
            break;
      settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
      }
}

void chose()
{
  char ch;
      setcolor(1);
      moveto(296,675);
      sound(800); delay(300); nosound();
      outtext (" YOU MUST PRESS BOTTON  a,b,c OR d ... TRY AGAIN !!!");
      setcolor(14);
      moveto(360,675);
      outtext(" SET GRAPH  a , b , c  OR  d  : ");
      ch = getch();
      moveto(360,675);
      switch(ch)
         {
           case 97: outtext(" SET GRAPH  a , b , c  OR  d  :  a"); break;
           case 98: outtext(" SET GRAPH  a , b , c  OR  d  :  b"); break;
           case 99: outtext(" SET GRAPH  a , b , c  OR  d  :  c"); break;
           case 100: outtext(" SET GRAPH  a , b , c  OR  d  :  d"); break;
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(300);
if (ch == 'a')
    {
      setcolor(1);
      moveto(360,675);
      outtext(" SET GRAPH  a , b , c  OR  d  :  a");
      setcolor(14);
      moveto(360,675);
      sound(800); delay(300); nosound();
      outtext (" YOU SELECT CHOICE  ..a..");
      delay(1800);
      setcolor(1);
      moveto(360,675);
      outtext (" YOU SELECT CHOICE  ..a..");
    }
else  if (ch == 'b')
    {
      setcolor(1);
      moveto(360,675);
      outtext(" SET GRAPH  a , b , c  OR  d  :  b");
      setcolor(14);
      moveto(360,675);
      sound(800); delay(300); nosound();
      outtext (" YOU SELECT CHOICE  ..b..");
      delay(1800);
      setcolor(1);
      moveto(360,675);
      outtext (" YOU SELECT CHOICE  ..b..");
    }
else  if (ch == 'c')
    {
      setcolor(1);
      moveto(360,675);
      outtext(" SET GRAPH  a , b , c  OR  d  :  c");
      setcolor(14);
      moveto(360,675);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sound(800); delay(300); nosound();
        outtext (" YOU SELECT CHOICE  ..c..");
        delay(1800);
        setcolor(1);
        moveto(360,675);
        outtext (" YOU SELECT CHOICE  ..c..");
      }
  else  if (ch == 'd')
      {
        setcolor(1);
        moveto(360,675);
        outtext(" SET GRAPH  a , b , c  OR  d  :  d");
        setcolor(14);
        moveto(360,675);
        sound(800); delay(300); nosound();
        outtext (" YOU SELECT CHOICE  ..d..");
        delay(1800);
        setcolor(1);
        moveto(360,675);
        outtext (" YOU SELECT CHOICE  ..d..");
      }
  else
      {
        setcolor(1);
        moveto(360,675);
        outtext(" SET GRAPH  a , b , c  OR  d  :  ");
        setcolor(14);
        moveto(296,675);
        sound(800); delay(300); nosound();
        outtext (" YOU MUST PRESS BOTTON  a,b,c OR d ... TRY AGAIN !!!");
        delay (1800);
        chose();
      }
      switch(ch) /*select gate*/
       {
        case'a':
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            w=1;
          break;
          case'b':
            w=2;
          break;
          case'c':
            w=3;
          break;
          case'd':
            w=4;
          break;
        }
}

void filename(void)
{
    int x = 385 ,y = 657;
    setfillstyle(1,80);
    bar(x,y,x+500,y+50);
    setfillstyle(1,24);
    bar(x,y+51,x+502,y+52);
    bar(x+501,y+51,x+502,y);
    setfillstyle(1,82);
    bar(x-2,y-2,x-1,y+51);
    bar(x-2,y-2,x+502,y-1);
    setcolor(15);
    line(x+119,y+35,x+471,y+35);
    line(x+471,y+11,x+471,y+35);
    setcolor(24);
    line(x+119,y+11,x+119,y+35);
    line(x+119,y+11,x+471,y+11);
    setfillstyle(1,0);
    bar(x+120,y+12,x+470,y+35);
    setcolor(4);
    outtextxy(x+10,y+13,"FILENAME");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void cursor(int row,int col)
{
   union REGS ireg;
   ireg.h.ah = 0x02;
   ireg.h.bh = getvpage();
   ireg.h.dh = row - 1;
   ireg.h.dl = col - 1;
   int86(0x10,&ireg,&ireg);
}

int getvpage(void)
{
   union REGS ireg;
   ireg.h.ah = 0x0f;
   int86(0x10,&ireg,&ireg);
   return(ireg.h.bh);
}

void reset()
{
    char str[10];
    int i;
    int maxx,maxy;

      maxx = getmaxx();
      maxy = getmaxy();

      setviewport(93,160,715,510,CLIP_ON);
      clearviewport();
      setcolor(11);
      rectangle(0,0,718,510);
      setfillstyle(SOLID_FILL,11);
      floodfill(359,255,LIGHTCYAN);
      setviewport(90,530,760,550,CLIP_ON);
      clearviewport();
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    setcolor(11);
    setfillstyle(SOLID_FILL,11);
    rectangle(0,0,670,20);
    floodfill(317.5,10,LIGHTCYAN);
    setviewport(26,322,83,335,CLIP_ON);
    clearviewport();
    setcolor(11);
    setfillstyle(SOLID_FILL,11);
    rectangle(0,0,57,13);
    floodfill(28.5,6.5,LIGHTCYAN);
    sound(950); delay(300); nosound();
    setviewport(0,0,maxx,maxy,CLIP_ON);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    for(i=1;i<=10;i++)
    {
        setcolor(4);
        sprintf(str,"%d",i);
        outtextxy(80+62*i,526,str);
    }
}

void setgraph(int p)
{
    char ch;
    int midx, midy, t,j;
    char pattern[8] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

    midx = (getmaxx() / 2)-93;
    midy = (getmaxy() / 2)-45;

      for (t=0; t<200; t=t+15)
       {
         setfillpattern(pattern,80);
         for(j=0; j<165; j=j+15)
         bar(midx-t, midy-j, midx+t,midy+j);
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(0);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
rectangle(midx-195, midy-150, midx+195,midy+150);
setcolor(0);
outtextxy(midx-50,midy-110,"SET GRAPH");
setcolor(6);
switch(p)
{
  case 1:
        outtextxy(midx-120,midy-50," a :  NOT DRAW GRAPH");
        outtextxy(midx-120,midy," b :  RANGE  1 KHz");
        outtextxy(midx-120,midy+50," c :  RANGE  5 KHz");
        outtextxy(midx-120,midy+100," d :  RANGE  10 KHz");
  break;
  case 2:
     outtextxy(midx-120,midy-50," a :  RANGE  1 KHz");
     outtextxy(midx-120,midy," b :  RANGE  5 KHz");
     outtextxy(midx-120,midy+50," c :  RANGE  10 KHz");
     outtextxy(midx-120,midy+100," d :  RANGE  50 KHz");
  break;
}
setcolor(14);
moveto(360,675);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
sound(800); delay(300); nosound();
outtext(" SET GRAPH  a , b , c  OR  d  : ");
ch = getch();
moveto(360,675);
switch(ch)
{
  case 97: outtext(" SET GRAPH  a , b , c  OR  d  :  a"); break;
  case 98: outtext(" SET GRAPH  a , b , c  OR  d  :  b"); break;
  case 99: outtext(" SET GRAPH  a , b , c  OR  d  :  c"); break;
  case 100: outtext(" SET GRAPH  a , b , c  OR  d  :  d"); break;
}
delay(300);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (ch == 'a')
{
   setcolor(1);
   moveto(360,675);
   outtext(" SET GRAPH  a , b , c  OR  d  :  a");
   setcolor(14);
   moveto(360,675);
   sound(800); delay(300); nosound();
   outtext (" YOU SELECT CHOICE  ..a..");
   delay(1800);
   setcolor(1);
   moveto(360,675);
   outtext (" YOU SELECT CHOICE  ..a..");
}
else  if (ch == 'b')
{
   setcolor(1);
   moveto(360,675);
   outtext(" SET GRAPH  a , b , c  OR  d  :  b");
   setcolor(14);
   moveto(360,675);
   sound(800); delay(300); nosound();
   outtext (" YOU SELECT CHOICE  ..b..");
   delay(1800);
   setcolor(1);
   moveto(360,675);
   outtext (" YOU SELECT CHOICE  ..b..");
}
else  if (ch == 'c')
{
   setcolor(1);
   moveto(360,675);
   outtext(" SET GRAPH  a , b , c  OR  d  :  c");
   setcolor(14);
   moveto(360,675);
   sound(800); delay(300); nosound();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    outtext (" YOU SELECT CHOICE  ..c..");
    delay(1800);
    setcolor(1);
    moveto(360,675);
    outtext (" YOU SELECT CHOICE  ..c..");
  }
  else  if (ch == 'd')
  {
    setcolor(1);
    moveto(360,675);
    outtext(" SET GRAPH  a , b , c  OR  d  :  d");
    setcolor(14);
    moveto(360,675);
    sound(800); delay(300); nosound();
    outtext (" YOU SELECT CHOICE  ..d..");
    delay(1800);
    setcolor(1);
    moveto(360,675);
    outtext (" YOU SELECT CHOICE  ..d..");
  }
  else
  {
    setcolor(1);
    moveto(360,675);
    outtext(" SET GRAPH  a , b , c  OR  d  :  ");
    setcolor(14);
    moveto(296,675);
    sound(800); delay(300); nosound();
    outtext (" YOU MUST PRESS BOTTON  a,b,c OR d ... TRY AGAIN !!!");
    delay (1800);
    chose();
  }
  switch(ch) /*select gate*/
  {
   case'a':
      w=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
        case'b':
           w=2;
        break;
        case'c':
           w=3;
        break;
        case'd':
           w=4;
        break;
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**********************/
/*          key.h          */
/**********************/

#define Esc_Key   0x11B
#define Enter_Key 0x1c0d
#define Home_Key 0x4700
#define End_Key 0x4f00
#define PageUp_Key 0x4900
#define PageDown_Key 0x5100
#define Left_Key 0x4b00
#define Right_Key 0x4d00
#define Up_Key 0x4800
#define Down_Key 0x5000
#define F1_Key 0x3b00
#define F2_Key 0x3c00
#define F3_Key 0x3d00
#define F4_Key 0x3e00
#define F5_Key 0x3f00
#define F6_Key 0x4000
#define F7_Key 0x4100
#define F8_Key 0x4200
#define F9_Key 0x4300
#define F10_Key 0x4400
#define Insert_Key 0x5200
#define Delete_Key 0x5300
#define BackSpace_Key 0x0e08

#define Alt_Q  0x1000
#define Alt_W  0x1100
#define Alt_E  0x1200
#define Alt_R  0x1300
#define Alt_T  0x1400
#define Alt_Y  0x1500
```

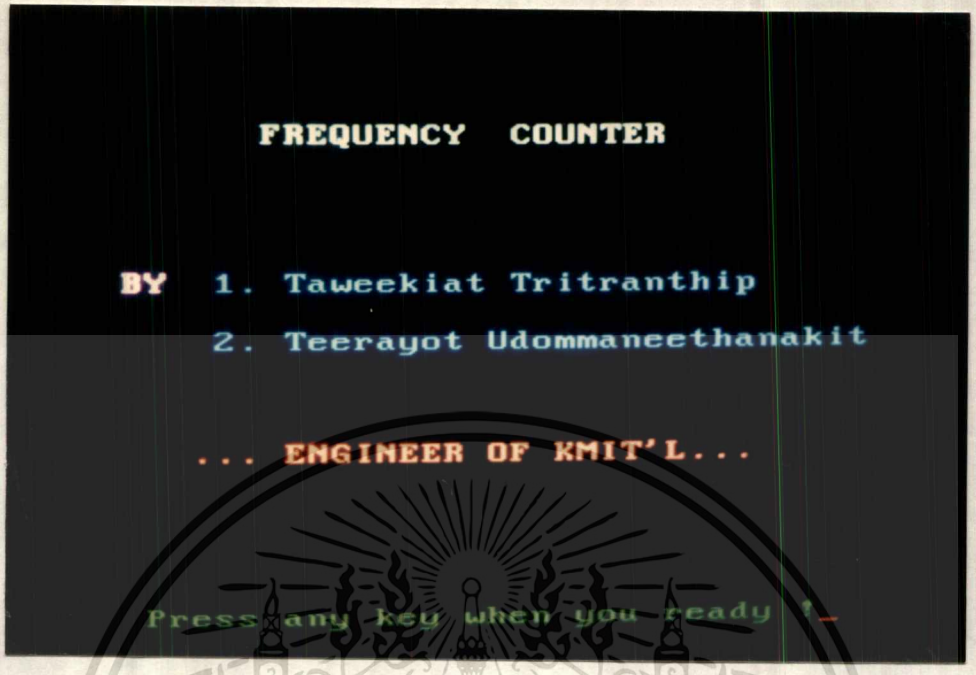
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Alt_U  0x1600
#define Alt_I  0x1700
#define Alt_O  0x1800
#define Alt_P  0x1900

#define Alt_A  0x1e00
#define Alt_S  0x1f00
#define Alt_D  0x2000
#define Alt_F  0x2100
#define Alt_G  0x2200
#define Alt_H  0x2300
#define Alt_J  0x2400
#define Alt_K  0x2500
#define Alt_L  0x2600

#define Alt_Z  0x2c00
#define Alt_X  0x2d00
#define Alt_C  0x2e00
#define Alt_V  0x2f00
#define Alt_B  0x3000
#define Alt_N  0x3100
#define Alt_M  0x3200

#define Shift_F1 0x5400
#define Shift_F2 0x5500
#define Shift_F3 0x5600
#define Shift_F4 0x5700
#define Shift_F5 0x5800
#define Shift_F6 0x5900
#define Shift_F7 0x5a00
#define Shift_F8 0x5b00
#define Shift_F9 0x5c00
#define Shift_F10 0x5d00

#define Ctrl_F1 0x5e00
#define Ctrl_F2 0x5f00
```

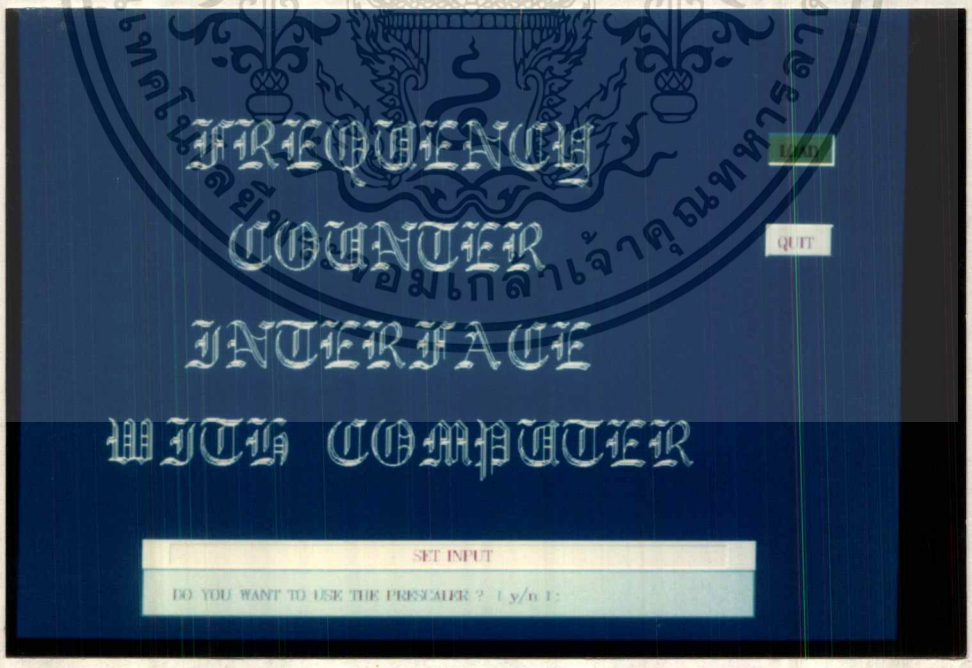
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Ctrl_F3 0x6000
#define Ctrl_F4 0x6100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

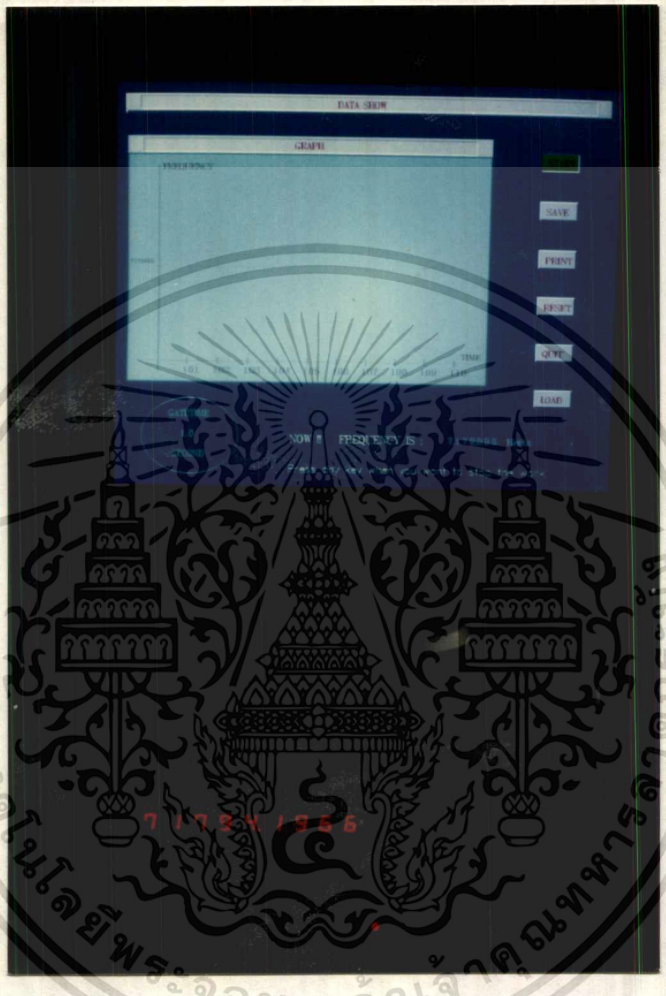


รูปที่ 2.16 แสดงฉากแนะนำเริ่มต้น



รูปที่ 2.17 แสดงฉากก่อนการเริ่มต้นการทำงาน
ซึ่งต้องมีการกำหนดข้อมูลบางอย่าง

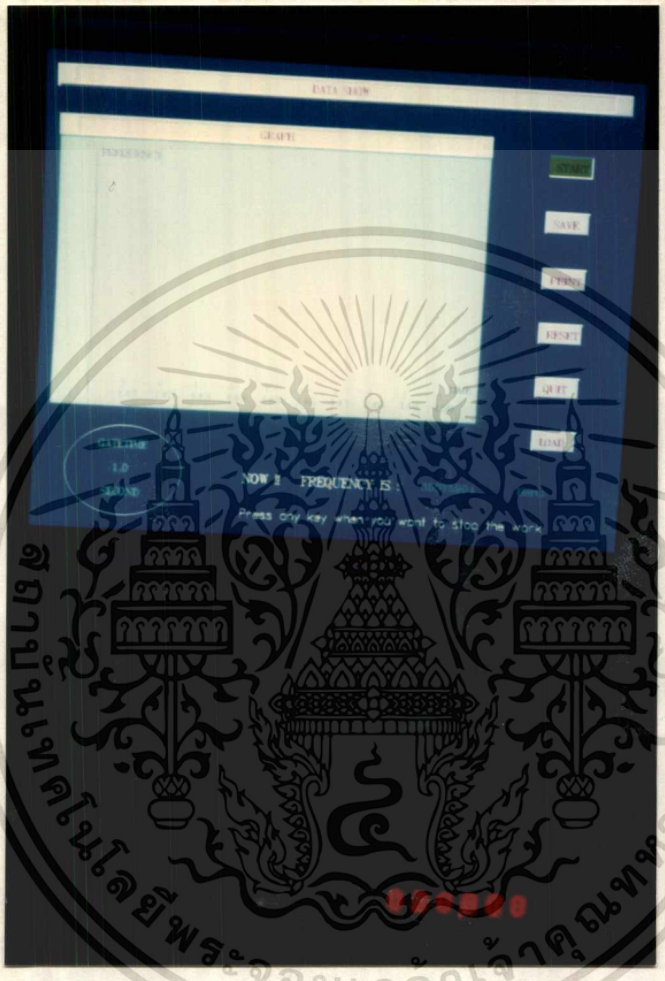
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



7 1 7 2 4 / 2 5 5 6

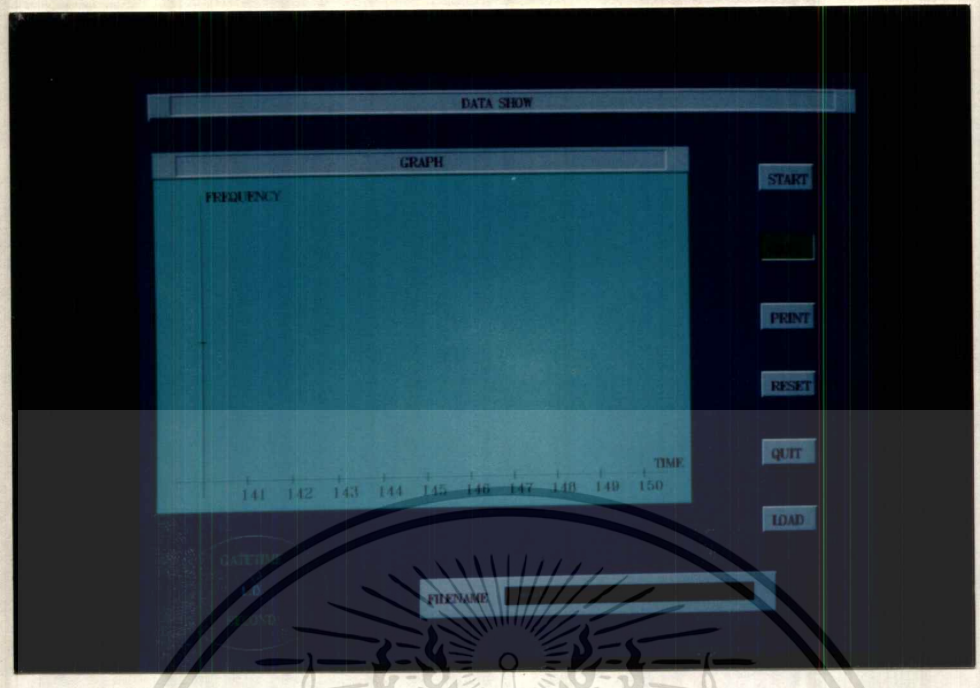
รูปที่ 2.18 แสดงฉากประกอบการทำงาน
ในกรณีที่ไม่ใช้ตัวหารความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

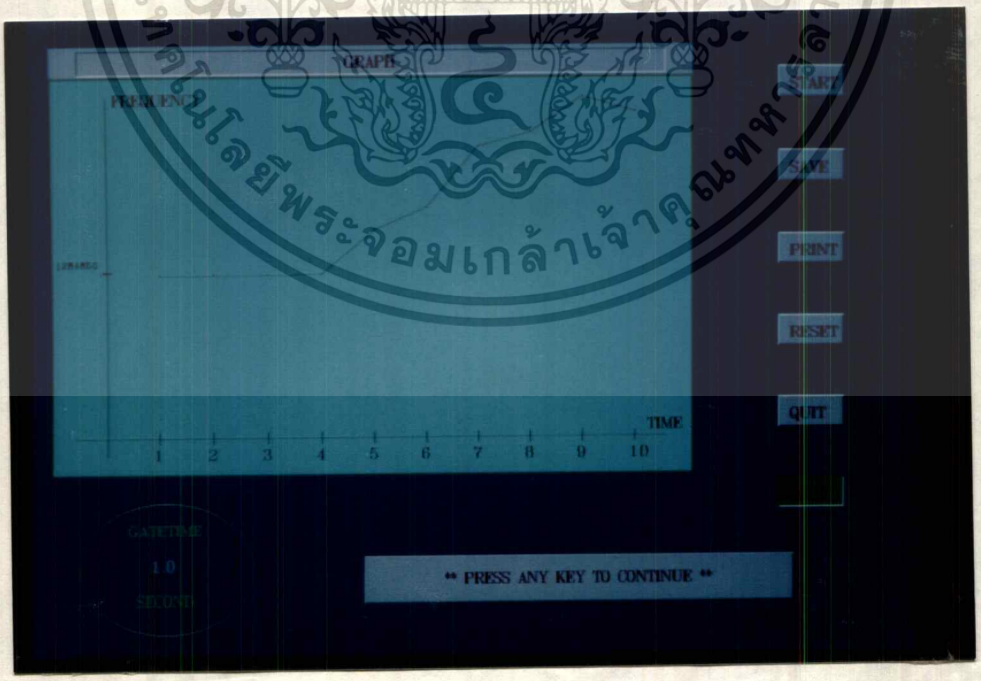


รูปที่ 2.19 แสดงฉากประกอบการทำงาน
ในกรณีที่ใช้ตัวหารความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงฉากการทำงานที่ต้องการเก็บข้อมูล
ที่วัดได้ลงแฟ้มข้อมูล



รูปที่ 2.21 แสดงฉากการทำงานที่แสดงค่าความถี่
ในแฟ้มข้อมูลที่เก็บไว้มาวาดกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

สัญญาณเข้าที่ไม่ผ่านตัวหารความถี่

FREQUENCY SYNTHESIZER MG 440 C	MICROWAVE COUNTER TR 5211 A	FREQUENCY COUNTER (PROJECT)	PERCENT ERROR (%)
100 Hz	100 Hz	130 Hz	30 %
200 Hz	200 Hz	220 Hz	10 %
400 Hz	400 Hz	420 Hz	5 %
1 KHz	1 KHz	1.01 KHz	1 %
5 KHz	5 KHz	5.01 KHz	0.2 %
10 KHz	10 KHz	10.01 KHz	0.1 %
50 KHz	50 KHz	50 KHz	0 %
100 KHz	100 KHz	100 KHz	0 %
500 KHz	500 KHz	499.98 KHz	0.004 %
1 MHz	1 MHz	0.99996 MHz	0.0004 %
5 MHz	5 MHz	4.9997 MHz	0.00006 %
10 MHz	10 MHz	9.9998 MHz	0.02 %
15 MHz	15 MHz	14.9997 MHz	0.002 %
20 MHz	20 MHz	19.9991 MHz	0.0045 %
25 MHz	25 MHz	24.9988 MHz	0.0048 %
30 MHz	30 MHz	29.9994 MHz	0.002 %
35 MHz	35 MHz	34.998 MHz	0.005 %
36 MHz	36 MHz	35.991 MHz	0.025 %
37 MHz	37 MHz	36.952 MHz	0.129 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

สัญญาณเข้าที่ผ่านตัวหารความถี่

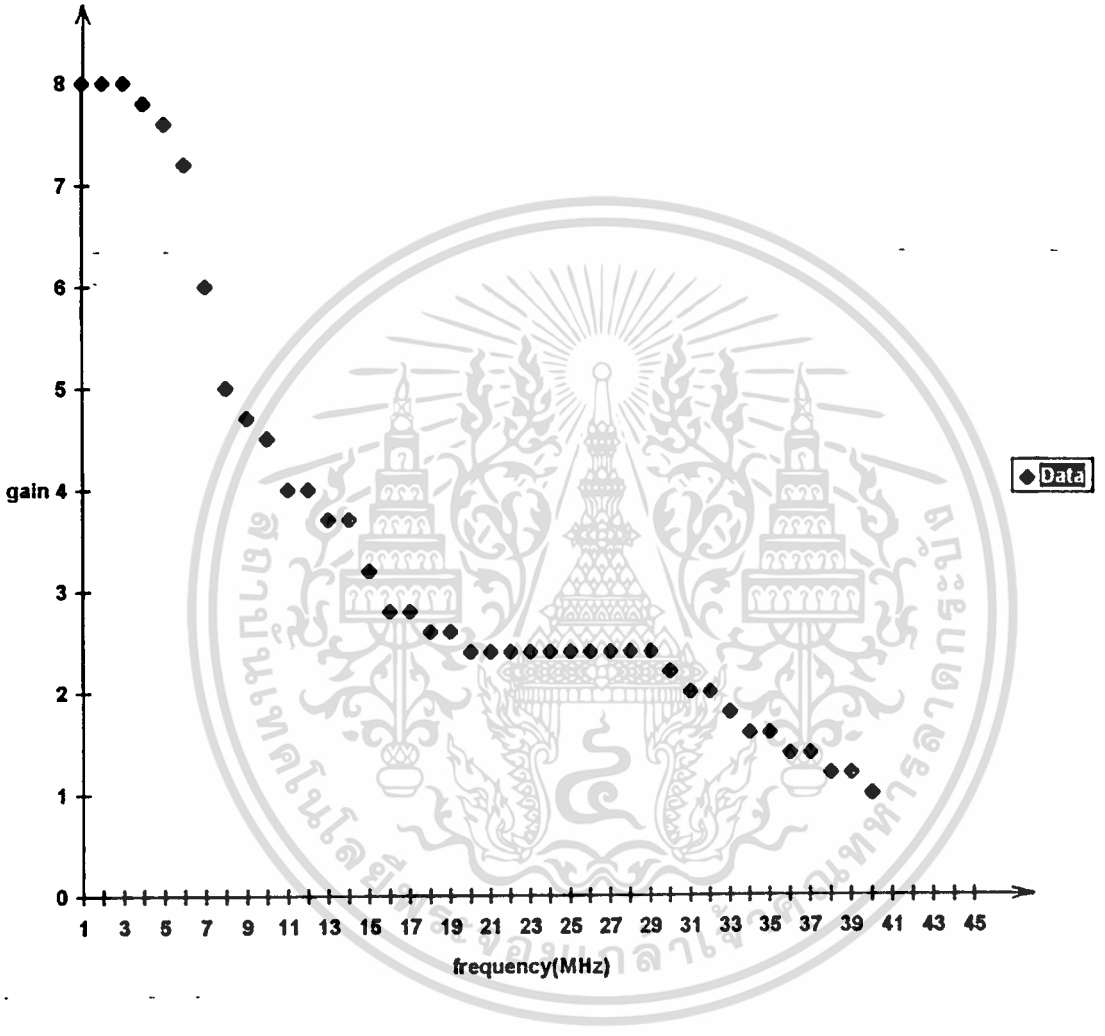
SIGNAL GENERATOR 8640 M	MICROWAVE COUNTER TR 5211 A	FREQUENCY COUNTER (PROJECT)	PERCENT ERROR (%)
50 MHz	50.0003 MHz	50.05 MHz	0.1 %
100 MHz	100.0007 MHz	99.996 MHz	0.004%
150 MHz	150.0006 MHz	149.996 MHz	0.002 %
200 MHz	199.9991 MHz	199.992 MHz	0.004 %
250 MHz	249.9995 MHz	249.984 MHz	0.006 %
300 MHz	299.9992 MHz	299.988 MHz	0.004 %
350 MHz	350.0004 MHz	349.804 MHz	0.056 %
400 MHz	400.0002 MHz	399.716 MHz	0.071 %
450 MHz	449.9995 MHz	449.581 MHz	0.093 %
500 MHz	500.0003 MHz	499.497 MHz	0.1 %
550 MHz	549.9991 MHz	550.936 MHz	0.17 %
576 MHz	576.0002 MHz	574.994 MHz	0.175 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQUENCY (MHz)	GAIN
1	8
2	8
3	8
4	7.8
5	7.6
6	7.2
7	6
8	5
9	4.7
10	4.5
11	4
12	4
13	3.7
14	3.7
15	3.2
16	2.8
17	2.8
18	2.6
19	2.6
20	2.4
21	2.4
22	2.4
23	2.4
24	2.4
25	2.4
26	2.4
27	2.4
28	2.4
29	2.4
30	2.2
31	2
32	2
33	1.8
34	1.6
35	1.6
36	1.4
37	1.4
38	1.2
39	1.2
40	1

ตารางแสดงข้อมูลระหว่างความถี่กับอัตราการขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟแสดงการตอบสนองความถี่ ของวงจรขยายสัญญาณ (IC เบอร์ 733)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQUENCY DATA

Date : 3/4/1995
Time : 2:4

[1] = 1284850
[2] = 1284860
[3] = 1284850
[4] = 1284920
[5] = 1287680
[6] = 1289030
[7] = 1292770
[8] = 1293610
[9] = 1295800
[10] = 1294850
[11] = 1294760
[12] = 1294530
[13] = 1294250
[14] = 1293820
[15] = 1293550
[16] = 1292880
[17] = 1292760
[18] = 1292040
[19] = 1292040
[20] = 1292040

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

จากรายละเอียดที่กล่าวข้างต้น ประกอบกับการดำเนินงาน 1 ปีที่ผ่านมาทำให้ผู้จัดทำได้เรียนรู้ถึงปัญหาต่าง ๆ ที่มักจะเกิดขึ้นในการทำงาน ต้องพยายามหาวิธีแก้ไข และจากการทำงานชิ้นนี้ทำให้ได้รับความรู้ในหลาย ๆ ด้าน ทั้งทางด้านการออกแบบวงจร , ออกแบบลายวงจรลงแผ่นปริ้นซ์ตลอดจนความรู้ทางด้านการเชื่อมต่อกับเครื่องคอมพิวเตอร์ รวมถึงทางด้านการเขียนโปรแกรมภาษาซีที่ใช้ในการควบคุมและแสดงผลด้วย

ถึงแม้ว่าค่าความถี่ที่ได้จากการวัดจะเกิดความผิดพลาดขึ้นบ้าง จนบางทีอาจไม่เป็นที่น่าประทับใจ แต่ก็สามารถวัดได้ถูกต้องในบางย่านความถี่ โดยแทบจะไม่มีความผิดพลาดเกิดขึ้นเลย จากผลการทดลองจะเห็นว่า ในกรณีที่ไม่ผ่านตัวหารความถี่ ช่องความถี่ที่วัดได้ผิดพลาดน้อยลง ตั้งแต่ 1 kHZ ขึ้นไป ส่วนในกรณีที่ผ่านตัวหารความถี่พบว่าตั้งแต่ 50 Mhz จนถึง 450 Mhz ผลการวัดเป็นที่น่าพึงพอใจ แต่ยังไม่ถึงกับดีมาก เพราะยังผิดพลาดอยู่บ้างแต่หลังจาก 450 Mhz ขึ้นไปจะเกิดความผิดพลาดมากขึ้นเรื่อย ๆ

สาเหตุในการเกิดความผิดพลาดนั้นอาจมาจากลักษณะของวงจรและตัวอุปกรณ์เหมาะสมกับการใช้งานในย่านความถี่ที่จำกัดเท่านั้น อีกทั้งยังอาจเกิดจากเวลาในการดำเนินงานของโปรแกรมไม่ตรงกับเวลาในการเปิดปิดเกท จึงมีผลทำให้ค่าความถี่ที่วัดผิดเพี้ยนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาต่อไป

จากการทำงานในระยะ 1 ปีที่ผ่านมานี้ ปัญหาต่างๆได้เกิดขึ้นมากมาย ทั้งทางด้านส่วนของชิ้นงาน และ ส่วนของโปรแกรมควบคุม โดยทางกลุ่มผู้จัดทำพยายามคิดและแก้ไขในสิ่งที่เกิดขึ้น ซึ่งอาจจะแก้ได้ดีบ้าง หรือไม่ดีบ้างปะปนกันไป ดังนั้นจึงส่งผลให้งานชิ้นนี้น่าจะมีจุดที่ควรเพิ่มเติม หรือปรับเปลี่ยนบ้างตามสถานการณ์และระยะเวลาที่ล่วงเลยไป

จะขอแจกแจงเป็นเป็นประเด็นๆ ที่น่าสนใจดังนี้

1. *ส่วนของชิ้นงาน (HARDWARE)*

ในขณะนี้ผลการทดลองที่วัดออกมามีความแม่นยำในบางช่วง และ เกิดความผิดพลาดมากขึ้นเมื่อวัดความถี่ที่สูงมากขึ้น สาเหตุอาจมาจาก ลักษณะของวงจรที่ออกแบบเองสามารถส่งผลดีในบางช่วงความถี่ที่วัดเท่านั้น หรือ อาจเป็นที่ตัวอุปกรณ์ทางไฟฟ้า ที่เหมาะสมกับการทำงานในย่านความถี่ที่ทนได้ ดังนั้นเพื่อความละเอียดและผิดพลาดน้อยลง น่าจะลองปรับเปลี่ยนอุปกรณ์บางตัว ถ้าต้องการวัดความถี่ได้สูงขึ้น ก็น่าจะมีการเปลี่ยนตัวหารความถี่ (PRESCALER) และ ยิ่งถ้าต้องการเพิ่มหลักของเลขที่ใช้แสดงผล ก็ต้องมีการเพิ่มพอร์ตที่ใช้ติดต่อกับคอมพิวเตอร์ เป็นต้น

2. *ส่วนของโปรแกรม (SOFTWARE)*

ทางโปรแกรมที่เขียนไว้ มีหน้าที่การทำงานอยู่ 6 แบบ โดยอาจจะมีการเพิ่มหน้าที่การทำงานให้มากขึ้น หรือ อาจมีการเพิ่มเติมส่วนที่คอยตรวจสอบความผิดพลาดในการแสดงผล ซึ่งบางทีทางผู้จัดทำอาจหลงลืมไปก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. COMPONENT DATA CATALOG 1987 , INTERSIL INC , P 14.80 - P 14.90

2. ELECTRONIC WORLD ฉบับที่ 144 , หน้า 7 - 14

3. อุทัย จึงภักดี, บทความเรื่อง "เทคนิคเครื่องวัดความถี่" ใน ELECTRONIC WORLD ฉบับที่ 136 , หน้า 101 - 108

4. ทฤษฎีและการใช้งานย่านความถี่ไมโครเวฟ หน้า 1 - 5 , พิชัย ภักดีพานิช สำนักพิมพ์ ฟิสิกส์เซ็นเตอร์

5. มุมนักทดลองดิจิตอล , ยืน ภู่สุวรรณ , หน้า 174 - 179

6. MICROPROCESSOR APPLICATION BY MR. John Unfenbeck

7. เทคนิคการเชื่อมต่อ IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานและผลงานที่เกิดขึ้นมานี้สำเร็จลุล่วงได้เนื่องจากคำแนะนำและแนวคิดของท่านผู้มีอุปการะคุณ ดังนี้

1. อาจารย์ นิภา ลีลารุจิ

2. พี่ สนทยา

และทางคณะผู้จัดทำขอขอบคุณท่านผู้มีอุปการะคุณทุกท่านที่อยู่เบื้องหลังความสำเร็จทุกๆ อย่าง ถึงแม้จะไม่ได้เอ่ยนามในที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2N4416 (SILICON)

Silicon N-channel junction field-effect transistor designed for VHF/UHF amplifier applications.

CASE 20 (1)
(TO-72)

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

Rating	Symbol	Value	Unit
Drain-Source Voltage	V_{DS}	30	Vdc
Drain-Gate Voltage	V_{DG}	30	Vdc
Gate-Source Voltage	V_{GS}	30	Vdc
Gate Current	I_G	10	mAdc
Total Device Dissipation @ T_A @ 25°C Derate above 25°C	P_D	300 1.7	mW mW/°C
Storage Temperature Range	T_{stg}	-65 to +200	°C

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS				
Gate-Source Breakdown Voltage (I_G = -1.0 μAdc, V_{DS} = 0)	$V_{(BR)GSS}$	30	-	Vdc
Gate-Source Cutoff Voltage (I_D = 1.0 nAdc, V_{DS} = 15 Vdc)	$V_{GS(off)}$	-	6.0	Vdc
Gate-Source Voltage (I_D = 0.5 mAdc, V_{DS} = 15 Vdc)	V_{GS}	1.0	5.5	Vdc
Gate-Source Forward Voltage (I_G = 1.0 mAdc, V_{DS} = 0)	$V_{GS(f)}$	-	1.0	Vdc
Gate Reverse Current (V_{GS} = -20 Vdc, V_{DS} = 0) (V_{GS} = -20 Vdc, V_{DS} = 0, T_A = +150°C)	I_{GSS}	- -	100 200	pAdc
ON CHARACTERISTICS				
Zero-Gate Voltage Drain Current* (V_{DS} = 15 Vdc, V_{GS} = 0)	I_{DSS}*	5.0	15	mAdc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

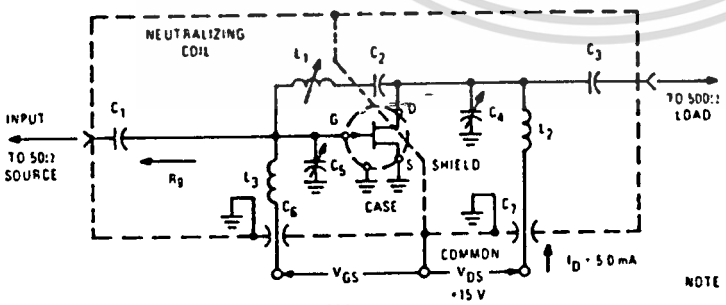
2N4416 (continued)

SMALL-SIGNAL CHARACTERISTICS

Characteristic	Symbol	Min	Max	Unit
Forward Transfer Admittance[1] (V_{DS} = 15 Vdc, V_{GS} = 0, f = 1.0 kHz)	$\|y_{fs}\|$	4500	7500	μmhos
Real Part of Forward Transconductance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 400 MHz)	$RE(y_{fs})$	4000	-	μmhos
Real Part of Input Conductance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 100 MHz) (V_{DS} = 15 Vdc, V_{GS} = 0, f = 400 MHz)	$RE(y_{is})$	- -	100 1000	μmhos
Output Admittance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 1.0 kHz)	$\|y_{os}\|$	-	50	μmhos
Real Part of Output Conductance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 100 MHz) (V_{DS} = 15 Vdc, V_{GS} = 0, f = 400 MHz)	$RE(y_{os})$	- -	75 100	μmhos
Imaginary Part of Input Susceptance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 100 MHz) (V_{DS} = 15 Vdc, V_{GS} = 0, f = 400 MHz)	$IM(y_{is})$	- -	2500 10,000	μmhos
Imaginary Part of Output Susceptance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 100 MHz) (V_{DS} = 15 Vdc, V_{GS} = 0, f = 400 MHz)	$IM(y_{os})$	- -	1000 4000	μmhos
Input Capacitance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 1.0 MHz)	C_{iss}	-	4.0	pF
Common-Source Output Capacitance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 1.0 MHz)	C_{osp}	-	2.0	pF
Reverse Transfer Capacitance (V_{DS} = 15 Vdc, V_{GS} = 0, f = 1.0 MHz)	C_{rss}	-	0.8	pF
Common-Source Spot Noise Figure (Figure 1) (V_{DS} = 15 Vdc, I_D = 5.0 mAdc, R_g = 1000 ohms, f = 100 MHz) (V_{DS} = 15 Vdc, I_D = 5.0 mAdc, R_g = 1000 ohms, f = 400 MHz)	NF	- -	2.0 4.0	dB
Small-Signal Power Gain (Figure 1) (V_{DS} = 15 Vdc, I_D = 5.0 mAdc, f = 100 MHz) (V_{DS} = 15 Vdc, I_D = 5.0 mAdc, f = 400 MHz)	G_{ps}	18 10	- -	dB

(1) Pulse Test: Pulse Width = 300 μs, Duty Cycle = 1.0%.

FIGURE 1 – 100 MHz & 400 MHz NEUTRALIZED AMPLIFIER



Reference Designation	VALUE 100 MHz	VALUE 400 MHz
C_1	7.0 pF	1.8 pF
C_2	1000 pF	17 pF
C_3	3.0 pF	1.0 pF
C_4	1.12 pF	0.8-8.0 pF
C_5	1.12 pF	0.8-8.0 pF
C_6	0.0015 μF	0.001 μF
C_7	0.0015 μF	0.001 μF
L_1	3.0 μH*	0.2 μH**
L_2	0.15 μH*	0.03 μH**
L_3	0.14 μH*	0.022 μH**

NOTE The noise source is a hot cold body (AIL type 70 or equivalent) with a test receiver (AIL type 136 or equivalent)

* L_1 17 turns (approx – depends upon circuit layout) AWG #28 enameled copper wire, close wound on 9/32" ceramic coil form. Tuning provided by a powdered iron slug.

L_2 4 1/2 turns, AWG #18 enameled copper wire, 5/16" long, 3/8" I.D. (AIR CORE)

L_3 3 1/2 turns, AWG #18 enameled copper wire, 1/4" long, 3/8" I.D. (AIR CORE)

** L_1 6 turns (approx – depends upon circuit layout) AWG #24 enameled copper wire, close wound on 7/32" ceramic coil form. Tuning provided by an aluminum slug.

L_2 1 turn, AWG #16 enameled copper wire, 3/8" I.D. (AIR CORE)

L_3 1/2 turn, AWG #16 enameled copper wire, 1/4" I.D. (AIR CORE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

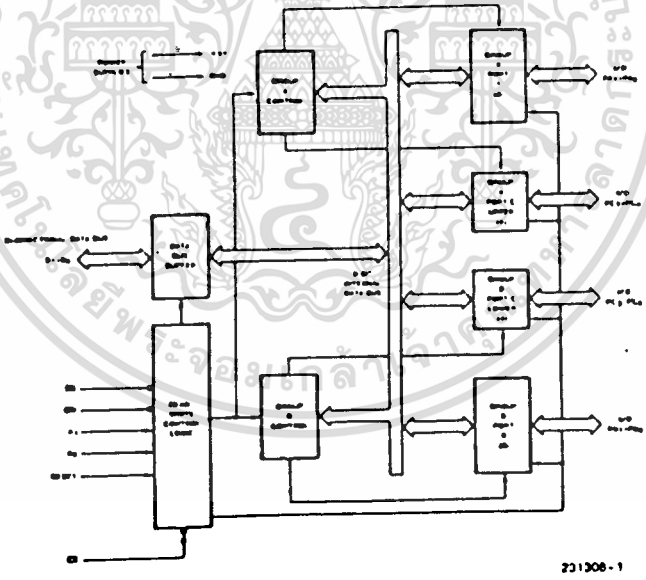
intel®

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range
- 40 Pin DIP Package or 44 Lead PLCC

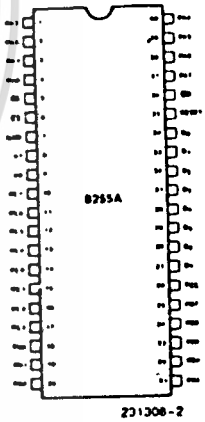
(See Intel Packaging Order Number 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



231308-1

Figure 1. 8255A Block Diagram



231308-2

Figure 2. Pin Configuration

September 1987
Order Number: 231308-002

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

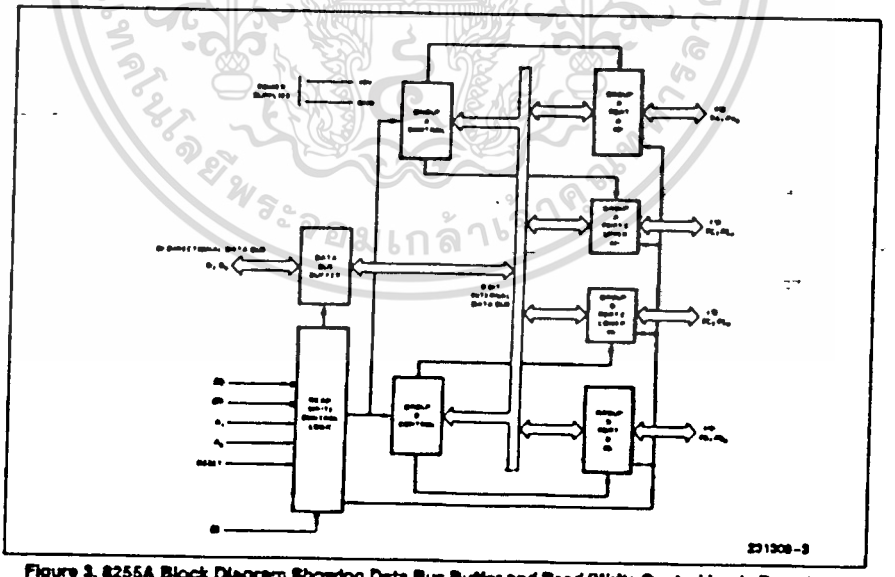
Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).



231308-3

Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (READ)
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
					Output Operation (WRITE)
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
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7–C4)

Control Group B—Port B and Port C lower (C3–C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

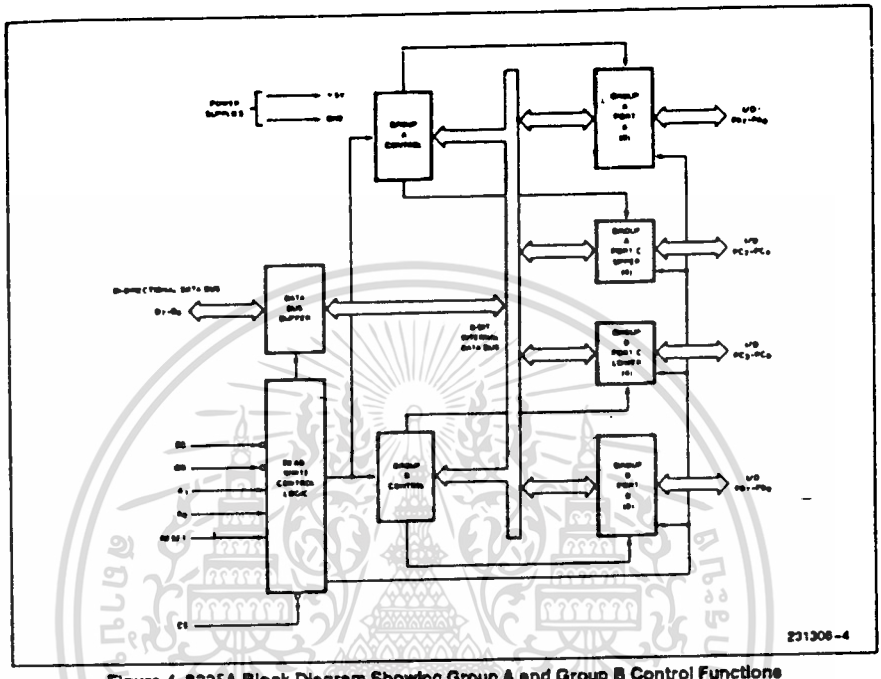
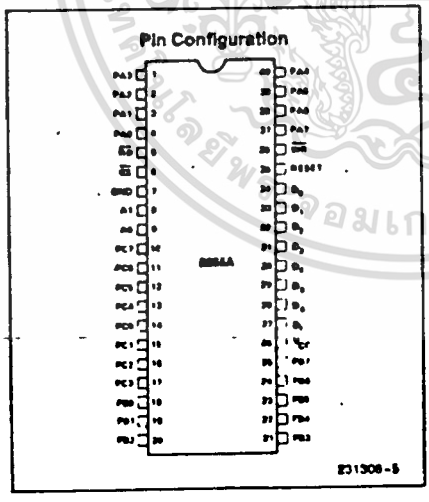


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions



Pin Names

D_7–D_0	Data Bus (Bi-Directional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7–PA0	Port A (BIT)
PB7–PB0	Port B (BIT)
PC7–PC0	Port C (BIT)
V_{CC}	+ 5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

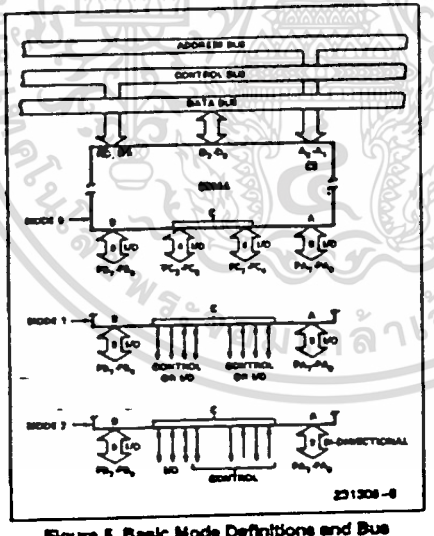


Figure 5. Basic Mode Definitions and Bus Interface

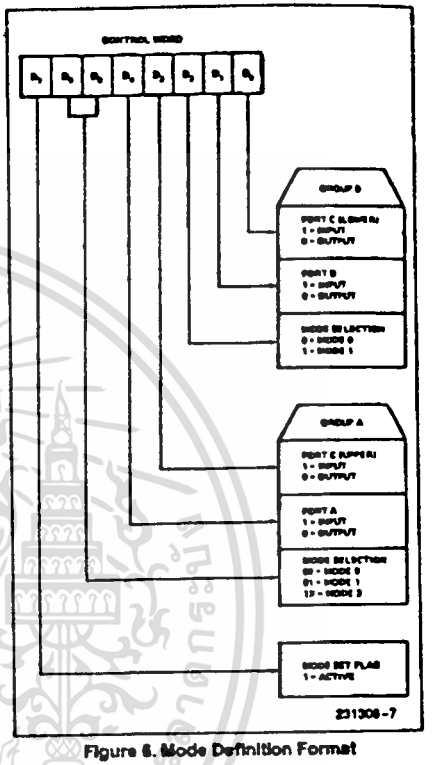


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

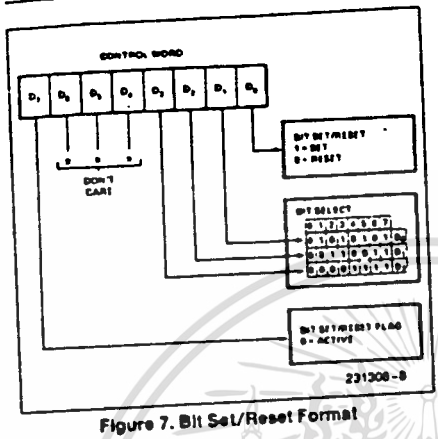


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:
All Mask flip-flops are automatically reset during mode selection and device Reset.

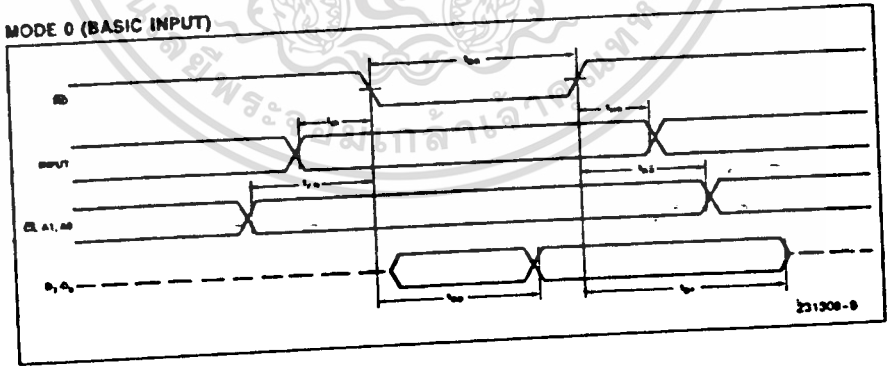
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

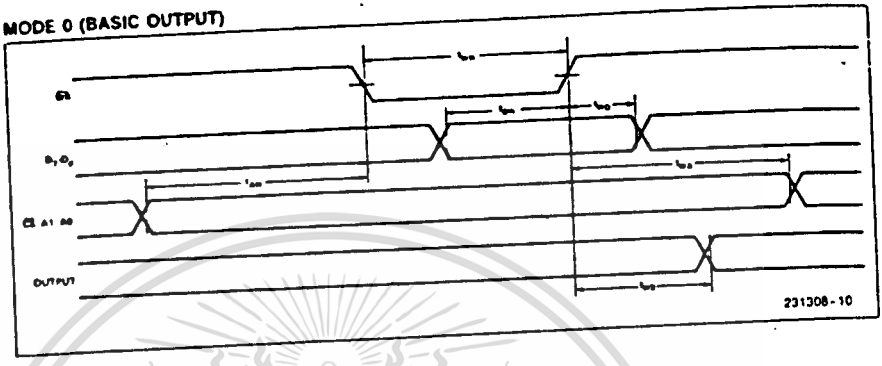
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



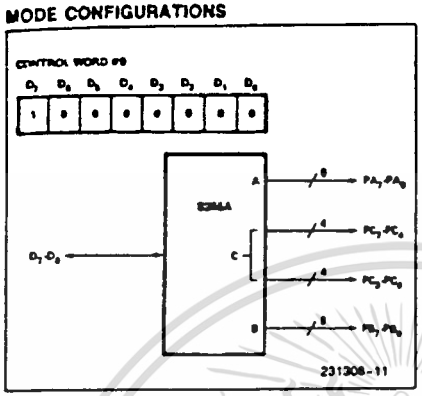
MODE 0 (BASIC OUTPUT)

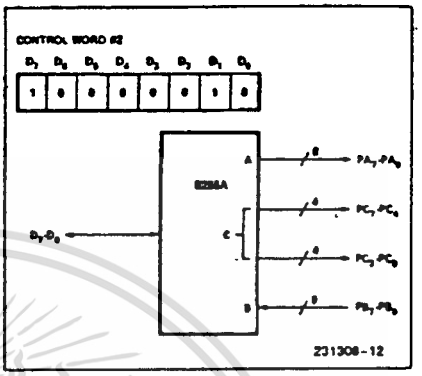


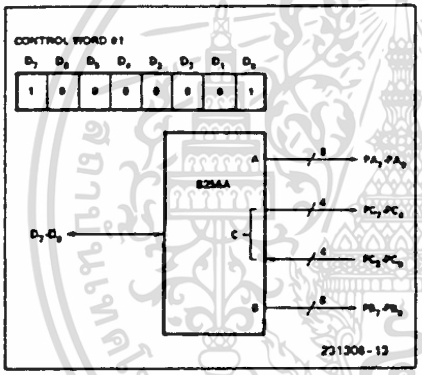
MODE 0 PORT DEFINITION

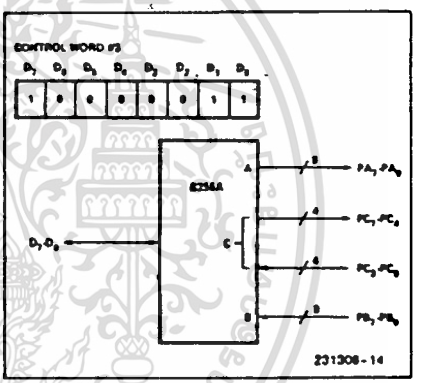
A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

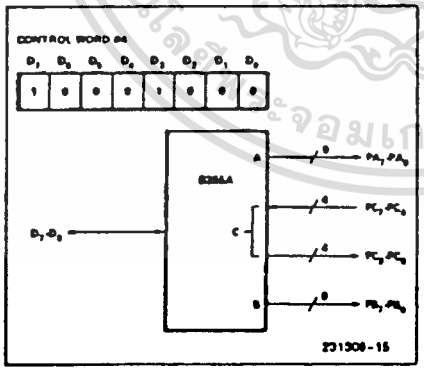
MODE CONFIGURATIONS

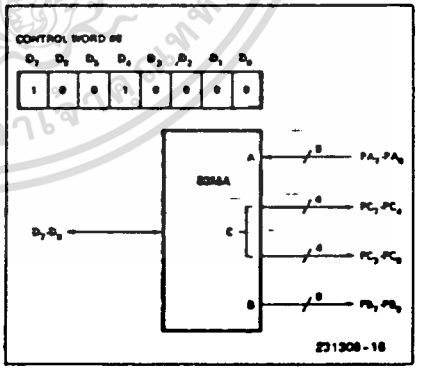


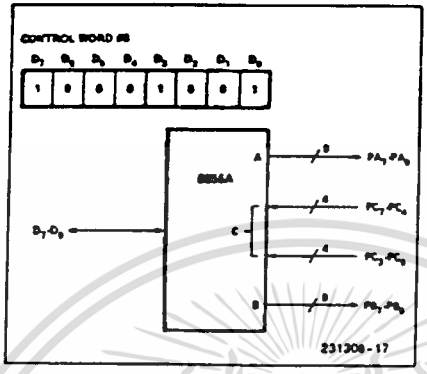


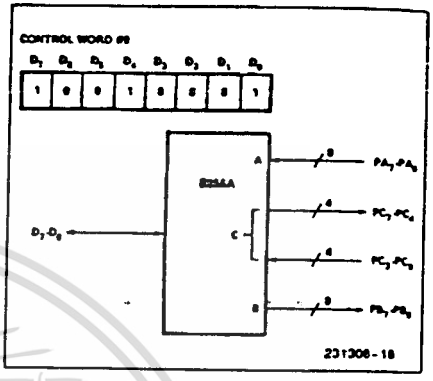


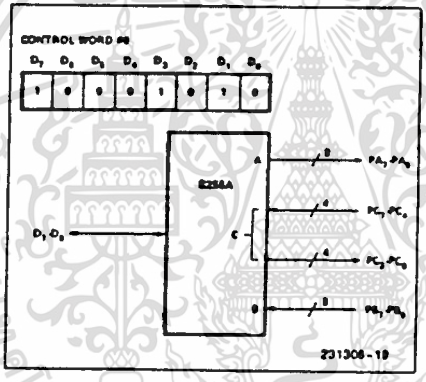


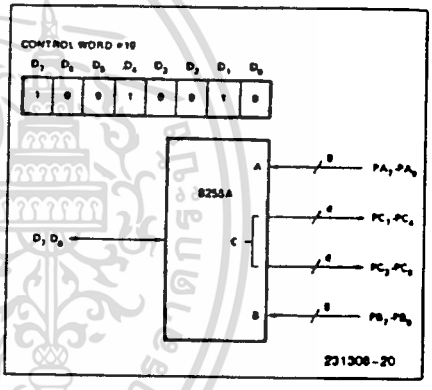


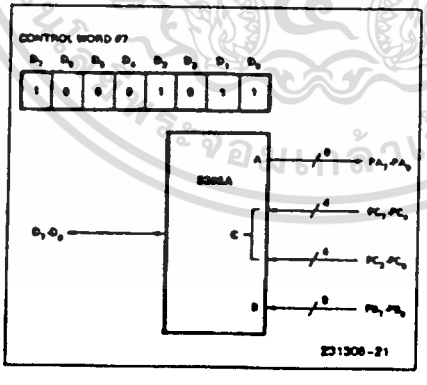


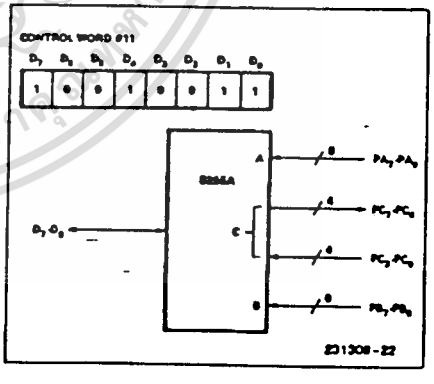

CONTROL WORD #8
231308-17

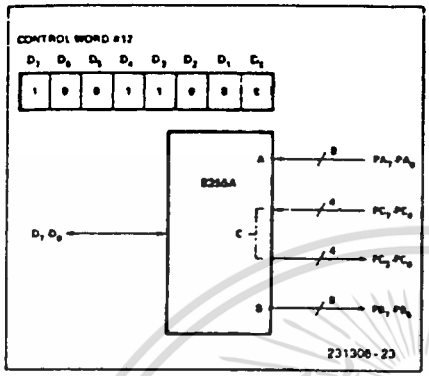

CONTROL WORD #9
231308-18

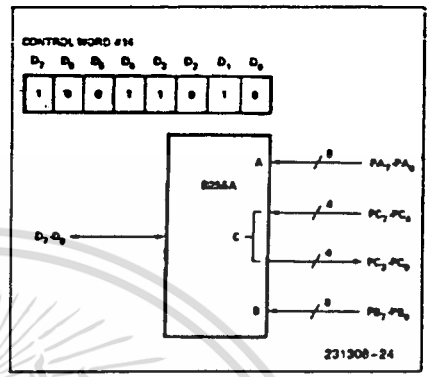

CONTROL WORD #6
231308-19

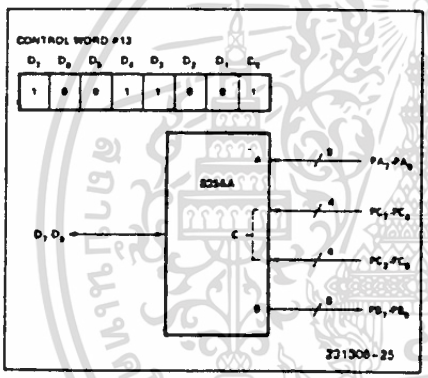

CONTROL WORD #10
231308-20

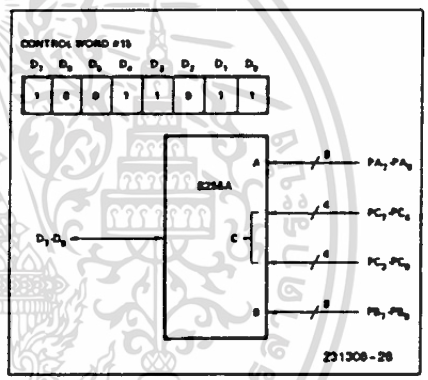

CONTROL WORD #7
231308-21


CONTROL WORD #11
231308-22









Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

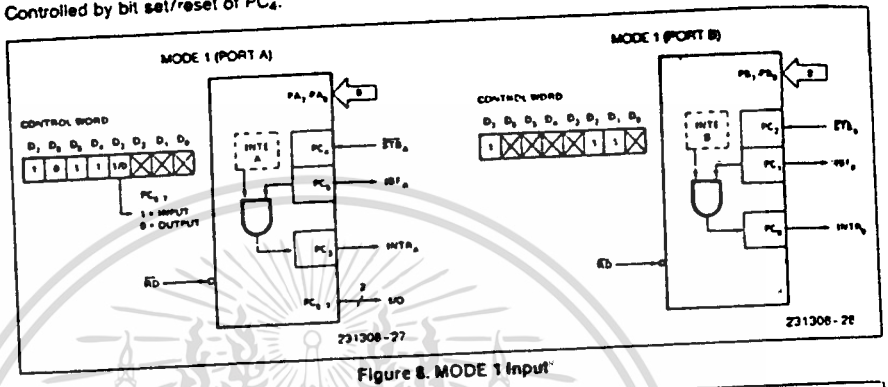


Figure 8. MODE 1 Input

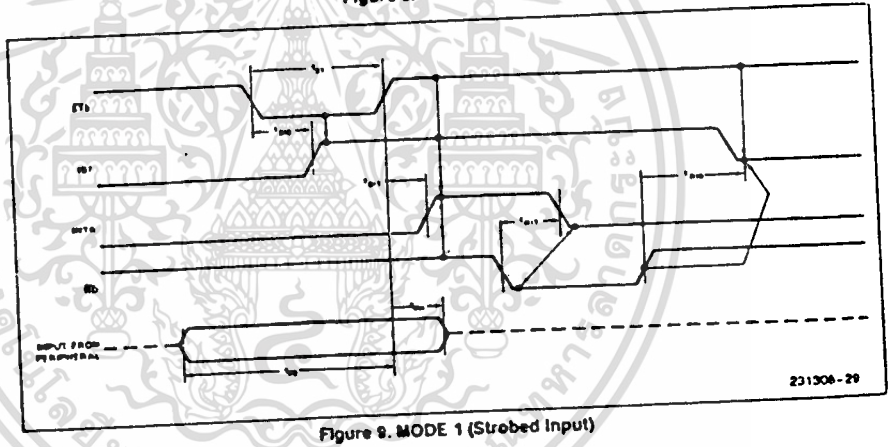


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

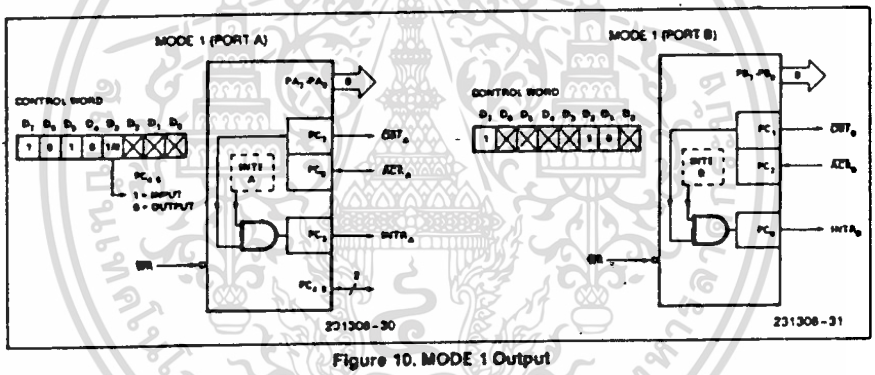


Figure 10. MODE 1 Output

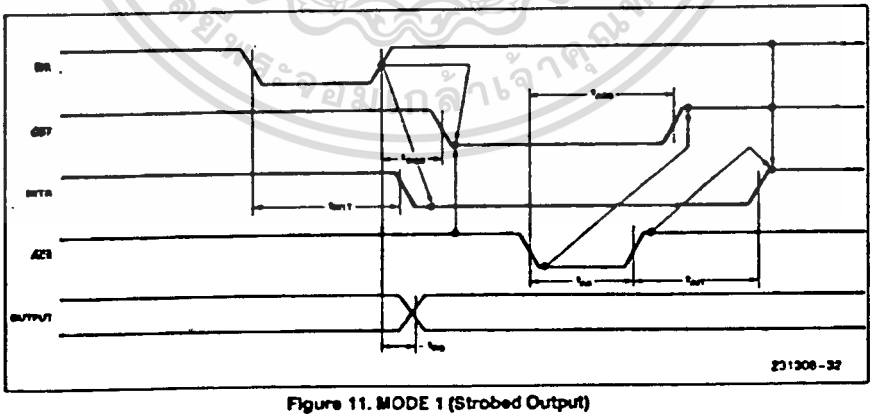


Figure 11. MODE 1 (Strobed Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

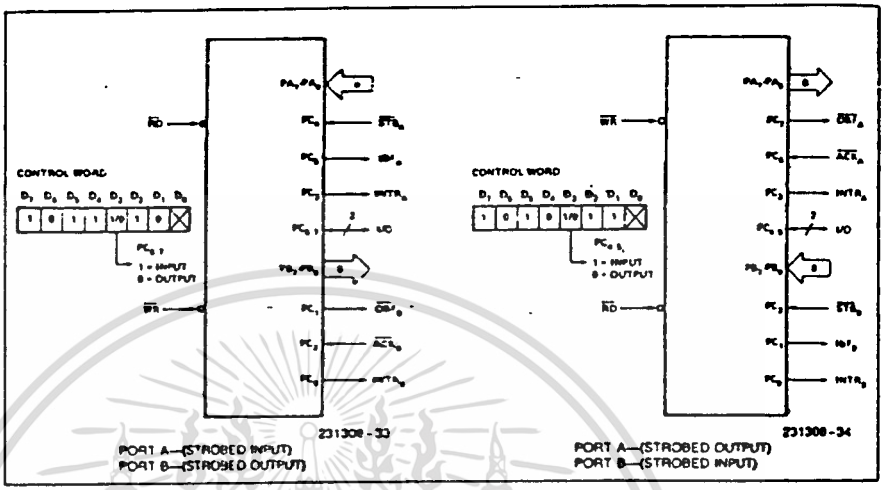


Figure 12. Combinations of MODE 1

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in MODE 1 to support a wide variety of strobed I/O applications.

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

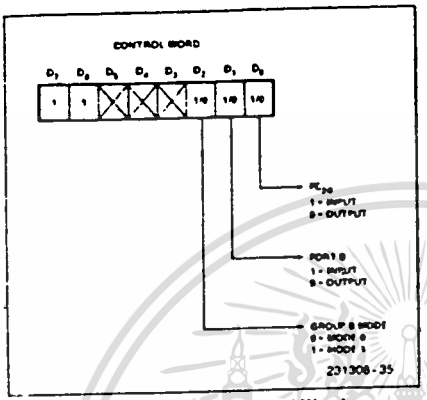


Figure 13. MODE Control Word

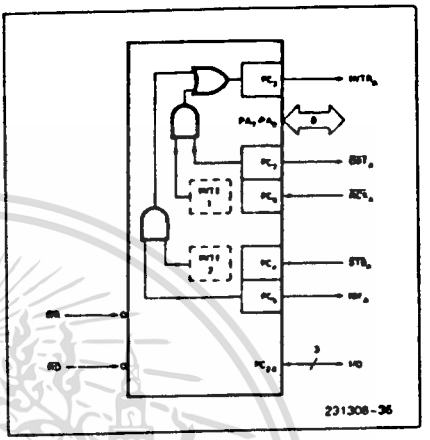
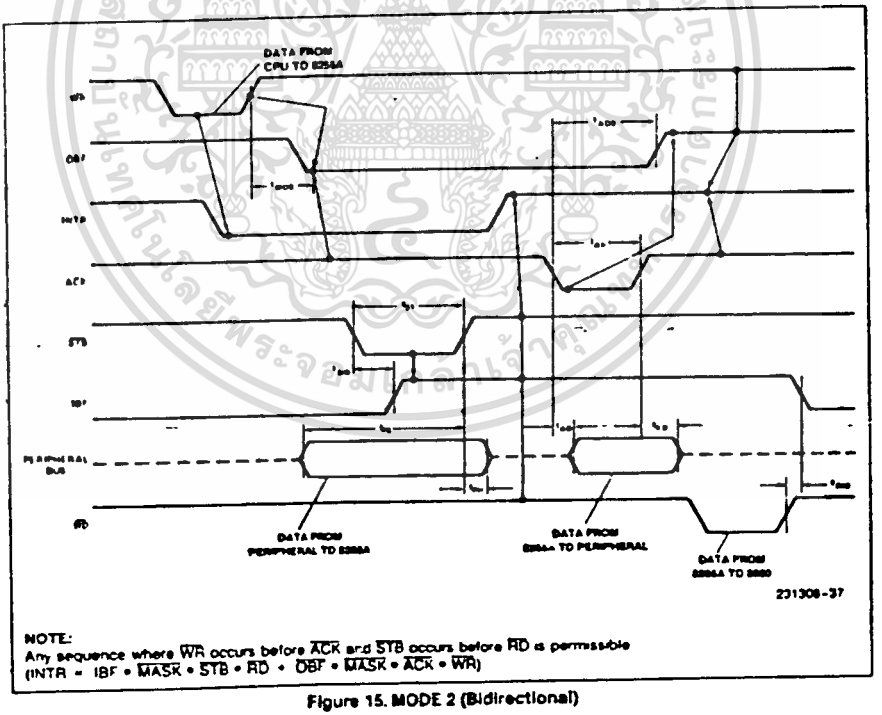


Figure 14. MODE 2



NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

Figure 15. MODE 2 (Bidirectional)

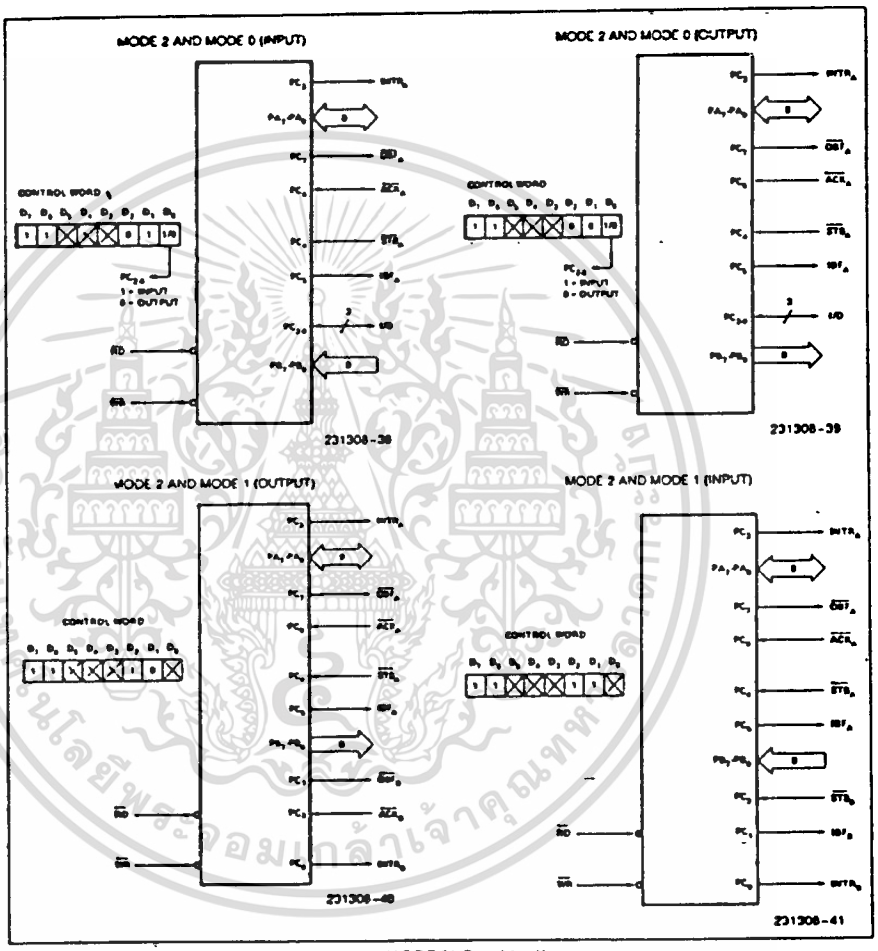


Figure 16. MODE 1 Combinations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs—

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs—

Bits in C upper (PC_7–PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3–PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1 mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

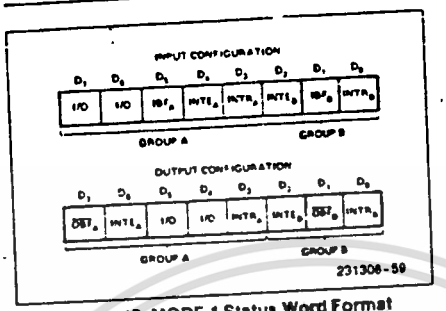


Figure 17. MODE 1 Status Word Format

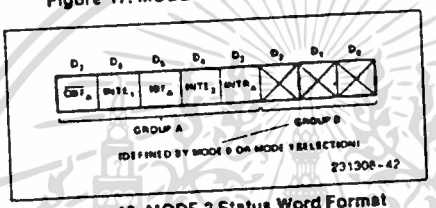


Figure 18. MODE 2 Status Word Format

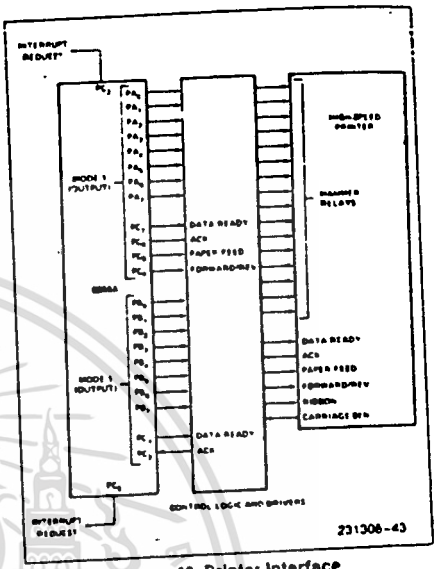


Figure 19. Printer Interface

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 represent a few examples of typical applications of the 8255A.

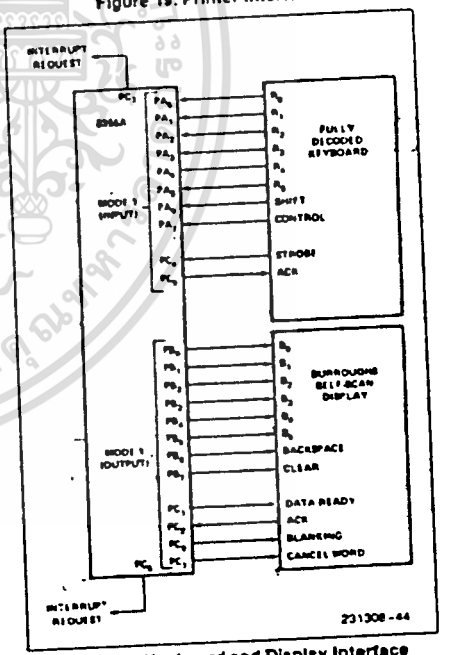


Figure 20. Keyboard and Display Interface

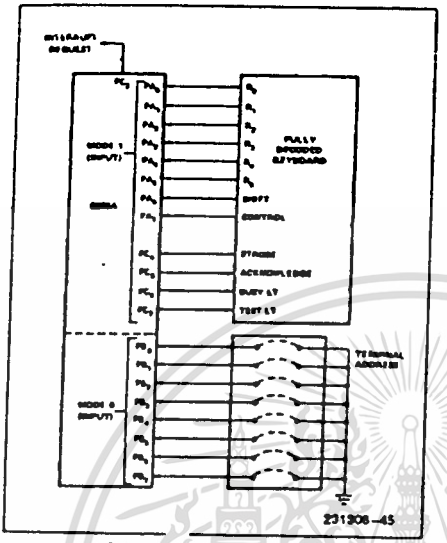


Figure 21. Keyboard and Terminal Address Interface

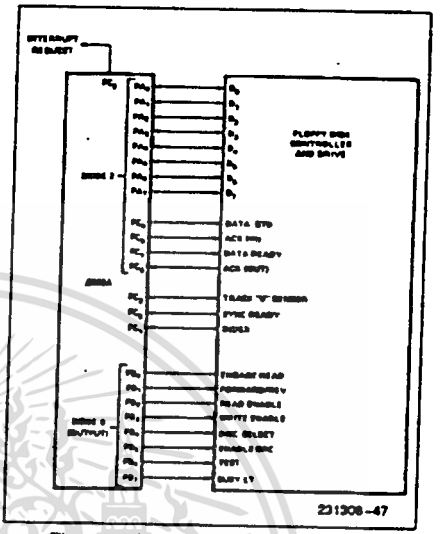


Figure 23. Basic Floppy Disk Interface

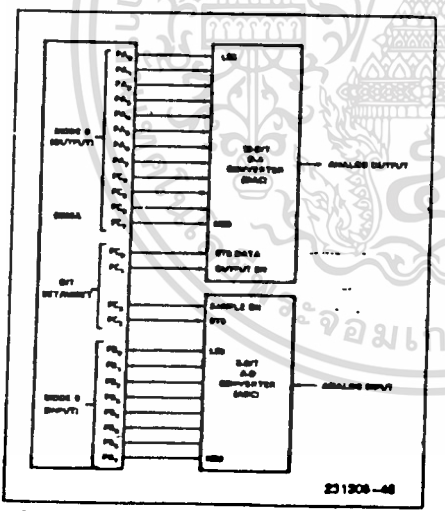


Figure 22. Digital to Analog, Analog to Digital

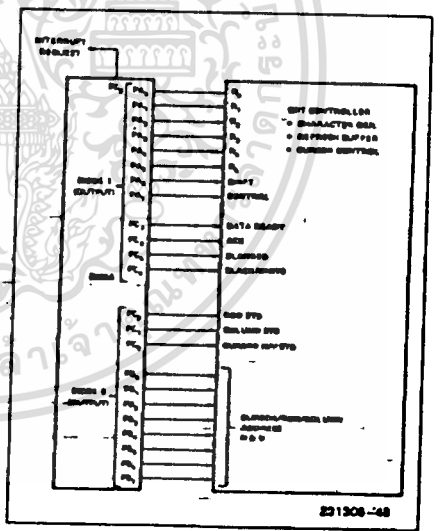


Figure 24. Basic CRT Controller Interface

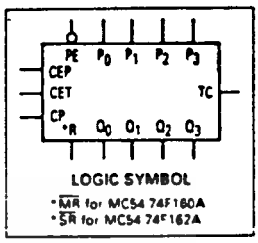
MOTOROLA

MC54F/74F160A
MC54F/74F162A

SYNCHRONOUS PRESETTABLE BCD DECADE COUNTER

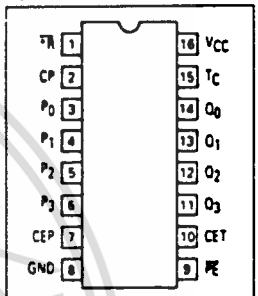
The MC54 74F160A and MC54 74F162A are high-speed synchronous decade counters operating in the BCD (8421) sequence. They are synchronously presettable for application in programmable dividers and have two types of Count Enable inputs plus a Terminal Count output for versatility in forming synchronous multistage counters. The MC54 74F160A has an asynchronous Master Reset input that overrides all other inputs and forces the outputs LOW. The MC54 74F162A has a Synchronous Reset input that overrides counting and parallel loading and allows the outputs to be simultaneously reset on the rising edge of the clock. The MC54 74F160A and MC54 74F162A are high-speed versions of the MC54 74F160 and MC54 74F162.

- Synchronous Counting and Loading
- High-Speed Synchronous Expansion
- Typical Count Rate of 120 MHz



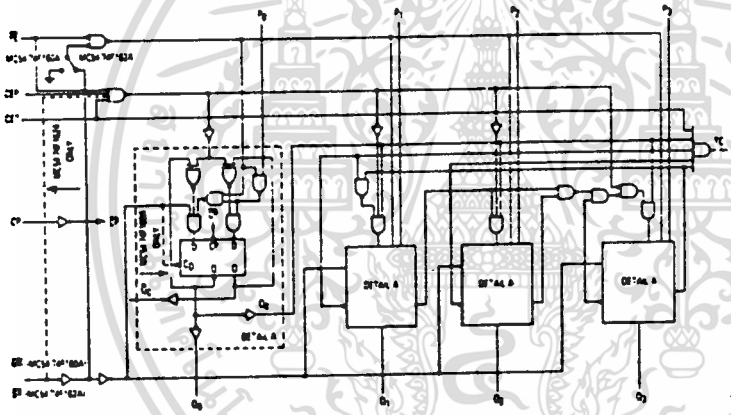
LOGIC SYMBOL

*$\overline{MR}$ for MC54 74F160A
*$\overline{SR}$ for MC54 74F162A



CONNECTION DIAGRAM

J Suffix — Case 620-09 (Ceramic)
N Suffix — Case 648-08 (Plastic)
D Suffix — Case 751B-03 (SOIC)



Please note that this diagram is provided only for the understanding of logic operations and should not be used to estimate propagation delays

Logic Diagram

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54, 74	4.5	5	5.5	V
T_A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I_{OH}	Output Current — High	54, 74			-1	mA
I_{OL}	Output Current — Low	54, 74			20	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

The MC54 74F160A and MC54 74F162A count modulo-10 in the BCD (8421) sequence. From state 9 (HLLH) they increment to state 0 (LLLL). The clock inputs of all flip-flops are driven in parallel through a clock buffer. Thus all changes of the Q outputs (except due to Master Reset of the MC54 74F160A) occur as a result of, and synchronous with, the LOW-to-HIGH transition of the CP input signal. The circuits have four fundamental modes of operation, in order of precedence: asynchronous reset (MC54 74F160A), synchronous reset (MC54 74F162A), parallel load, count-up and hold. Five control inputs — Master Reset ($\overline{MR}$, MC54 74F160A), Synchronous Reset ($\overline{SR}$ MC54 74F162A), Parallel Enable ($\overline{PE}$), Count Enable Parallel (CEP) and Count Enable Trickle (CET) — determine the mode of operation, as shown in the Mode Select Table. A LOW signal on $\overline{MR}$ overrides all other inputs and asynchronously forces all outputs LOW. A LOW signal on $\overline{SR}$ overrides counting and parallel loading and allows all outputs to go LOW on the next rising edge of CP. A LOW signal on $\overline{PE}$ overrides counting and allows information on the Parallel Data (P_n) inputs to be loaded into the flip-flops on the next rising edge of CP. With $\overline{PE}$ and $\overline{MR}$ (MC54 74F160A) or $\overline{SR}$ (MC54 74F162A) HIGH, CEP and CET permit counting when both are HIGH. Conversely, a LOW signal on either CEP or CET inhibits counting.

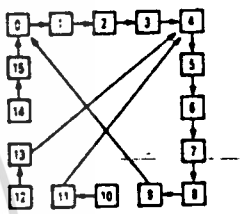
The MC54 74F160A and MC54 74F162A use D-type edge triggered flip-flops and changing the $\overline{SR}$, $\overline{PE}$, CEP and CET inputs when the CP is in either state does not cause errors, provided that the recommended setup and hold times, with respect to the rising edge of CP, are observed.

TRUTH TABLE

*$\overline{SR}$	$\overline{PE}$	CET	CEP	Action on the Rising Clock Edge (∫)
L	X	X	X	Reset (Clear)
H	L	X	X	Load ($P_n \rightarrow O_n$)
H	H	H	H	Count (Increment)
H	H	L	X	No Change (Hold)
H	H	X	L	No Change (Hold)

*For MC54 74F162A only H = HIGH Voltage Level L = LOW Voltage Level X = Don't Care

State Diagram



DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Units	Test Conditions	
V_{IH}	Input HIGH Voltage		2			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage				0.8	V	Guaranteed Input LOW Voltage for All Inputs	
V_{IK}	Input Clamp Diode Voltage				−1.2	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54, 74	2.5	3.4		V	I_{OH} = −1 mA	V_{CC} = 4.5 V
		74	2.7	3.4		V	I_{OH} = −1 mA	V_{CC} = 4.75 V
V_{OL}	Output LOW Voltage			0.35	0.5	V	I_{OL} = 20 mA	V_{CC} = MIN
I_{IH}	Input HIGH Current $\overline{MR}$, Data, CEP, Clock $\overline{PE}$, CET, $\overline{SR}$				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
	$\overline{MR}$, Data, CEP, Clock $\overline{PE}$, CET				0.1 0.1	mA	V_{CC} = MAX, V_{IN} = 7 V	
I_{IL}	Input LOW Current $\overline{MR}$, Data, CEP, Clock $\overline{PE}$, CET, $\overline{SR}$				−0.6 −1.2	mA	V_{CC} = MAX, V_{IN} = 0.5 V	
I_{OS}	Short Circuit Current		−60		−150	mA	V_{CC} = MAX, V_{OUT} = 0 V	
I_{CC}	Power Supply Current Total, Output HIGH Total, Output LOW			37	55	mA	V_{CC} = MAX	

NOTES: 1. For conditions such as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
2. Not more than one output should be shorted at a time, nor for more than 1 second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Terminal Count (TC) output is HIGH when CET is HIGH and the counter is in state 9. To implement synchronous multistage counters, the TC outputs can be used with the CEP and CET inputs in two different ways. Please refer to the MC54 74F568 data sheet. The TC output is subject to decoding spikes due to internal race conditions and is therefore not recommended for use as a clock or asynchronous reset for flip-flops, counters or registers. In the MC54 74F160A and MC54 74F162A decade counters, the TC output is fully decoded and can only be HIGH in state 9. If a decade counter is preset to an illegal state, or assumes an illegal state when power is applied, it will return to the normal sequence within two counts, as shown in the State Diagram.

Logic Equations: Count Enable = $CEP \cdot CET \cdot \overline{PE}$
$TC = Q_0 \cdot \overline{Q}_1 \cdot \overline{Q}_2 \cdot Q_3 \cdot CET$

AC CHARACTERISTICS See Page 5 for waveforms and load configurations

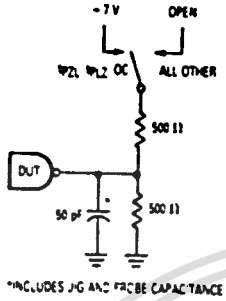
Symbol	Parameter	54 74F $T_A = +25°C$ $V_{CC} = +5\ V$ $C_L = 50\ pF$		54F $T_A = -55°C$ to $+125°C$ $V_{CC} = 5\ V \pm 10\%$ $C_L = 50\ pF$		74F $T_A = 0°C$ to $70°C$ $V_{CC} = 5\ V \pm 10\%$ $C_L = 50\ pF$		Units
		Min	Max	Min	Max	Min	Max	
f_{max}	Maximum Count Frequency	100		75		90		MHz
t_{PLH} t_{PHL}	Propagation Delay, Count CP to Q_n ($\overline{PE}$ Input HIGH)	3.5 3.5	7.5 10	3.5 3.5	9 11.5	3.5 3.5	8.5 11	ns
t_{PLH} t_{PHL}	Propagation Delay CP to Q_n ($\overline{PE}$ Input LOW)	3.5 4	8.5 8.5	4 4	10 10	3.5 4	9.5 9.5	ns
t_{PLH} t_{PHL}	Propagation Delay CP to TC	5 4.5	14 14	5 5	16.5 15	5 4.5	15 15	ns
t_{PLH} t_{PHL}	Propagation Delay CET to TC	2.5 2.5	7.5 7.5	2.5 2.5	9 9	2.5 2.5	8.5 8.5	ns
t_{PHL}	Propagation Delay $\overline{MR}$ to Q_n (MC54 74F160A)	5.5	12	5.5	14	5.5	13	ns
t_{PHL}	Propagation Delay $\overline{MR}$ to TC (MC54 74F160A)	4.5	10.5	4.5	12.5	4.5	11.5	ns

AC OPERATING REQUIREMENTS: See Page 5 for waveforms

Symbol	Parameter	54 74F $T_A = +25°C$ $V_{CC} = +5\ V$ $C_L = 50\ pF$		54F $T_A = -55°C$ to $+125°C$ $V_{CC} = 5\ V \pm 10\%$ $C_L = 50\ pF$		74F $T_A = 0°C$ to $70°C$ $V_{CC} = 5\ V \pm 10\%$ $C_L = 50\ pF$		Units
		Min	Max	Min	Max	Min	Max	
$t_s(H)$ $t_s(L)$	Setup Time, HIGH or LOW P_n to CP	5 5		5.5 5.5		5 5		ns
$t_h(H)$ $t_h(L)$	Hold Time, HIGH or LOW P_n to CP	2 2		2.5 2.5		2 2		ns
$t_s(H)$ $t_s(L)$	Setup Time, HIGH or LOW $\overline{PE}$ or $\overline{SR}$ to CP	11 8.5		13.5 10.5		11.5 9.5		ns
$t_h(H)$ $t_h(L)$	Hold Time, HIGH or LOW $\overline{PE}$ or $\overline{SR}$ to CP	2 0		2 0		2 0		ns
$t_s(H)$ $t_s(L)$	Setup Time, HIGH or LOW CEP or CET to CP	11 5		13 6		11.5 5		ns
$t_h(H)$ $t_h(L)$	Hold Time, HIGH or LOW CEP or CET to CP	0 0		0 0		0 0		ns
$t_w(H)$ $t_w(L)$	Clock Pulse Width (Load) HIGH or LOW	5 5		5 5		5 5		ns
$t_w(H)$ $t_w(L)$	Clock Pulse Width (Count) HIGH or LOW	4 6		5 8		4 7		ns
$t_w(L)$	$\overline{MR}$ Pulse Width, LOW (MC54 74F160A)	5		5		5		ns
t_{rec}	Recovery Time $\overline{MR}$ to CP (MC54 74F160A)	6		6		6		

4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*INCLUDES JIG AND PROBE CAPACITANCE

Figure 1. Test Load

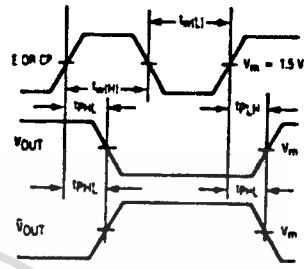


Figure 2. Propagation Delays from Rising-edge Clock or Enable

Figure 3. Waveform for Non-Inverting Functions

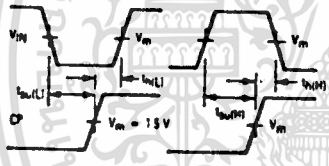


Figure 4. Setup and Hold Times, Rising-edge Clock

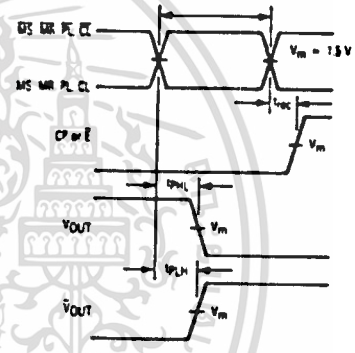


Figure 5. Asychronous Set, Reset, Parallel Load or Clear, Active Rising-edge Clock or Active-LOW Enable

4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

PLESSEY
Semiconductors

SP4653

1GHz ÷ 256 PRESCALER WITH LOW CURRENT AND LOW RADIATION

The SP4653 ÷ 256 prescaler is one of Plessey Semiconductors' latest range of high speed dividers for consumer frequency synthesis and measurement systems. It has a lower supply current giving reduced dissipation and operating temperatures in an 8-pin plastic DIL package. Spurious radiation has been reduced from all stages.

The SP4653 incorporates an on-chip preamplifier with differential inputs, and has balanced ECL outputs.

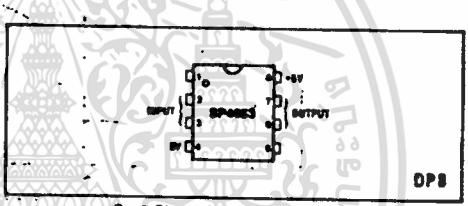


Fig.1 Pin connections - top view

FEATURES

- Low Supply Current
- Low Radiation
- Input Wideband Amplifier
- High Input Sensitivity
- High Input Impedance
- Balanced ECL Outputs

ABSOLUTE MAXIMUM RATINGS

Supply voltage	Vcc +7V
Input voltage	2.5V p-p
Storage temperature	-55°C to +125°C
Operating temperature range	0°C to +80°C

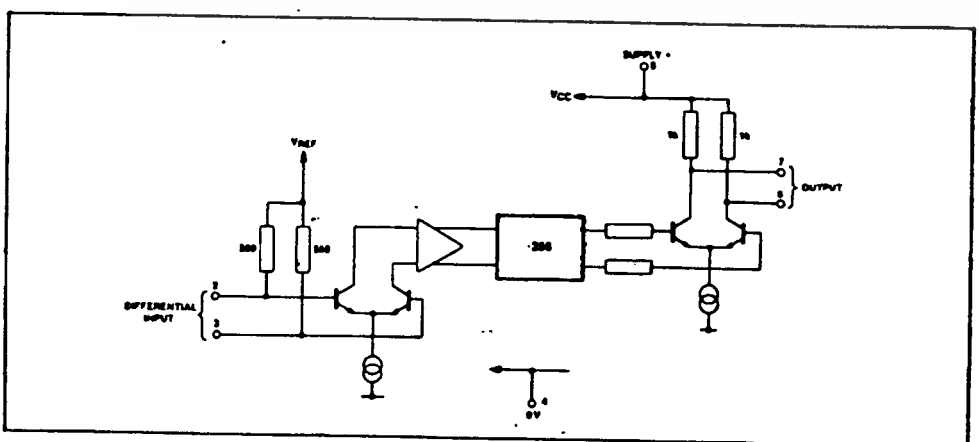


Fig.2 SP4653 block diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

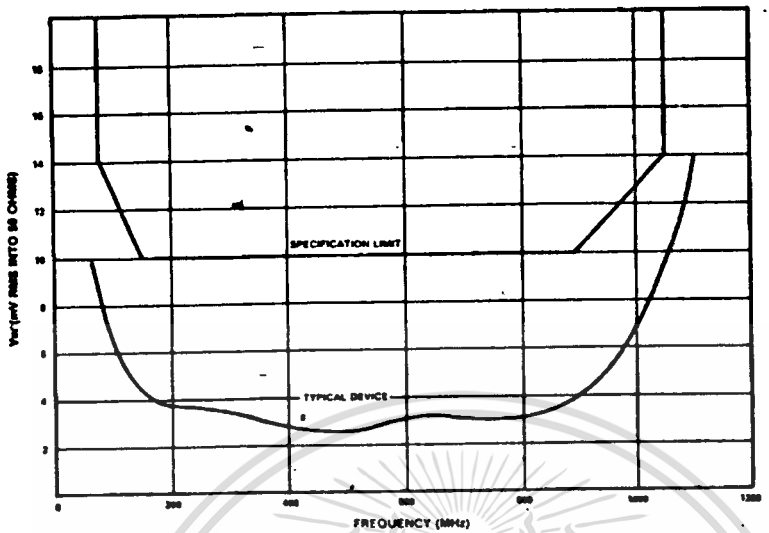


Fig.5 Typical input sensitivity

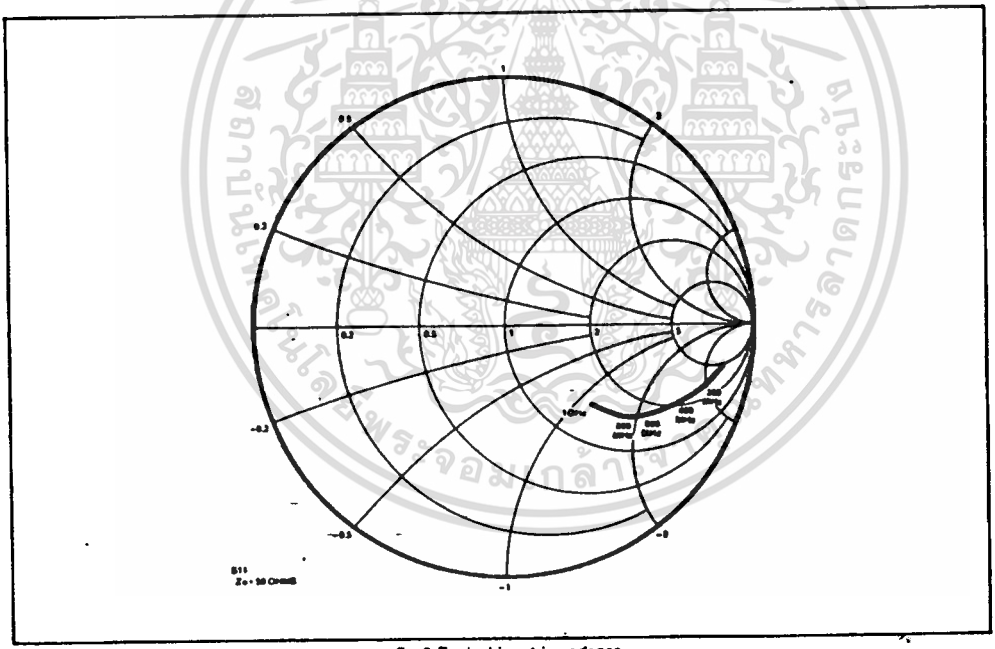


Fig.6 Typical input impedance

ELECTRICAL CHARACTERISTICS

Test conditions (unless otherwise stated):
T_{amb} = 0°C to +70°C, Vcc = 4.5V to 5.5V (Test circuit see Fig.3)

Characteristic	Pin	Value			Units	Conditions
		Min.	Typ.	Max.		
Supply current	8		25	35	mA	Vcc = 5V
Input sensitivity	2,3			17.5		RMS sinewave
70MHz			8	14	mV	
150MHz			4	10	mV	
300MHz			3	10	mV	
500MHz			3	10	mV	
700MHz			3	10	mV	
900MHz			4	10	mV	
1050MHz			6	14	mV	
Input overload	2,3	200			mV	70MHz to 1050MHz operating frequency
Input impedance	2,3		50		ohms	See Fig.6
			2		pF	
Output voltage no load	6	0.8			V p-p	fin = 1GHz Vcc = 5V
	7	0.8			V p-p	
Output voltage load as Fig.3	6	0.6			V p-p	fin = 1GHz Vcc = 5V
	7	0.6			V p-p	
Output impedance	6		1		kohms	
	7		1		kohms	
Output imbalance	6,7		0.1		V	

NOTE
The difference between the maximum input sensitivity and minimum overload voltages is the guaranteed dynamic range. Input signal levels should be maintained within these limits at all frequencies

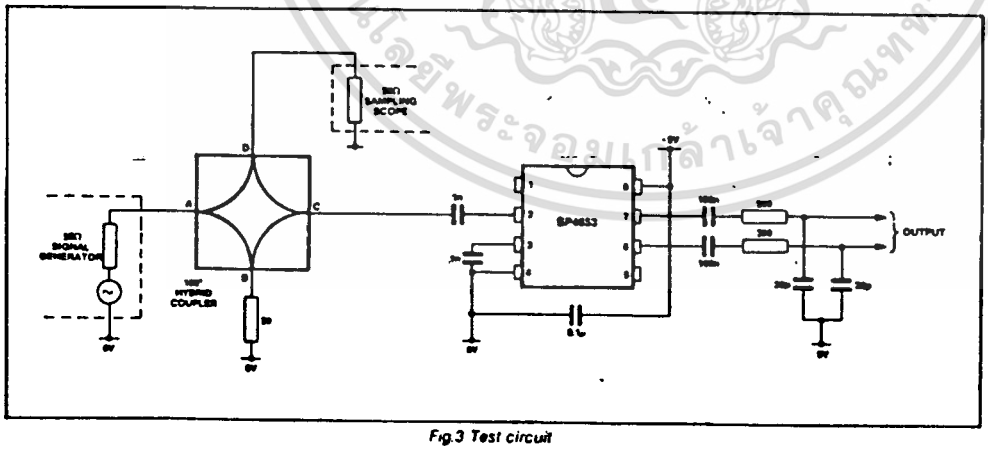


Fig.3 Test circuit

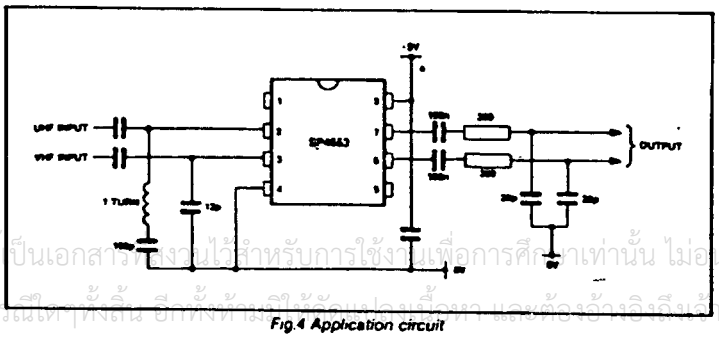


Fig.4 Application circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงความเป็นของเอกสารทุกครั้งที่มีการนำไปใช้