ปีการศึกษา 2530

CENTRAL PROCESSING UNIT

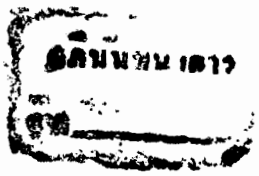
โดย

นายชัยพันธ์ ศรีรุ่งเรือง เลขประจำตัว 29.6302

นายอุดม ธีรธรรมานนท์ เลขประจำตัว 29.6327

อาจารย์ที่ปรึกษา

อาจารย์ชาญชัย พิสิทธิวิทยานนท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

024731 29 มค 2533

ปริญญานิพนธ์ปีการศึกษา 2530

เรื่อง CENTRAL PROCESSING UNIT

ผู้จัดทำ

1. นายชัยพันธ์ ศรีรุ่งเรือง เลขประจำตัว 29.6302
2. นายอุดม ธีรธรรมานนท์ เลขประจำตัว 29.6327

.....อาจารย์ที่ปรึกษา
(อาจารย์ชาญชัย พิสิทธิวิทยานนท์)

024731

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CENTRAL PROCESSING UNIT

โดย

นายชัยพันธ์ ศรีรุ่งเรือง
นายอุดม ชีรธรรมานนท์
อาจารย์ที่ปรึกษา
อาจารย์ชาญชัย พิสิทธิวิทยานนท์
ปีการศึกษา 2530

บทคัดย่อ

โครงงานนี้อธิบายถึงการนำสัญญาณต่าง ๆ ที่ SENSE มาจากรถยนต์ โดยโครงงาน AUTOMOBILE MEASURING SYSTEM มาประมวลผล โดยสัญญาณเหล่านี้จะถูก MULTIPLEX เข้ามาประมวลผลทีละสัญญาณ โดยผลลัพธ์ที่ได้จะถูกนำไปแสดงผลบนจอ CRT อีกทั้งยังนำไปเปรียบเทียบกับค่าที่ได้ตั้งไว้ เพื่อแสดงผลในรูปเสียงพูดภาษาไทยในกรณีที่เป็นการบอกกล่าว หรือคำเตือนสำหรับกรณีชัดข้องอย่างร้ายแรงของระบบรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CENTRAL PROCESSING UNIT

By

MR. CHAIYAN SRIRUNGRAUNG

MR. UDOM TEERATRAMMANON

ADVISOR

MR. CHANCHAI PISITVITTAYANON

ACADEMIC YEAR 1987

ABSTRACT

This Project describes about the getting of various signals which is sensed from Auto CAR system By the project "AUTOMOBILE MEASURING SYSTEM" and process it, these signals will multiplexed into this project signal by signal the result will be display on CRT and will be compare with set point in the case of statement or warning for the case of BAD failure in the Auto CAR system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	3
	2.1 หลักการเบื้องต้น	3
	2.2 system configurations	5
	2.3 สถาปัตยกรรมของ Microprocessor เบอร์ Z 80	6
	2.4 รีจิสเตอร์ต่าง ๆ ใน Z 80 CPU	8
	2.5 รายละเอียดของขา Z80 (Z80 pin outs)	10
	2.6 ไดอะแกรมเวลาของซีพียู	13
	2.7 ไซเคิลการเฟทช์ออฟโค้ด (OCF-opcode fetch)	15
	2.8 M_1 ไซเคิลที่มี wait states	16
	2.9 ไซเคิลการอ่านหน่วยความจำ (MR-memory read cycle) และ ไซเคิลการเขียนหน่วยความจำ (MW-memory write cycle)	17
	2.10 แมชีนไซเคิลของการเขียนเอาท์พุทและอ่านอินพุท (PW-port write and PR-port read)	18
	2.11 ไซเคิลการขอใช้บัสและการตอบสนองการใช้บัส (bus request and acknowledge cycle)	20
	2.12 ไซเคิลการรับสัญญาณอินเตอร์รัพท์และการตอบสนองต่อสัญญาณอินเตอร์รัพท์ (interrupt request and acknowledge cycle)	21
	2.13 การตอบสนองต่อการอินเตอร์รัพท์แบบนอนมาส-เคเบิ้ลอินเตอร์รัพท์ (normaskable interrupt response)	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การคำนวณและการสร้าง	23
3.1 memory map.	23
3.2 input/output port	24
3.3 8255 pio port # 1	25
3.4 ส่วนติดต่อกับ sensor	27
3.5 การติดต่อกับส่วน speech synthesizer	27
3.6 การติดต่อกับส่วน CRT display	28
3.7 การสร้างวงจร	29
3.8 การพัฒนาโปรแกรม	31
บทที่ 4 การทดลองและผลการทดลอง	32
4.1 การ interface โครงงาน AMS เข้ากับ โครงงาน CPU	32
4.2 การ interface โครงงาน GDS เข้ากับ โครงงาน CPU	34
4.3 การ interface โครงงาน speech synthesizer unit หรือ SSU เข้ากับ โครงงาน CPU	36
4.4 โปรโตคอลการส่ง control code จากโครงงาน CPU ไปยังโครงงาน SSU	38
บทที่ 5 บทวิจารณ์และสรุป	39
5.1 บทวิจารณ์	39
5.2 สรุป	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 1

บทนำ

ปริญญานิพนธ์ชุดนี้เป็นส่วนหนึ่งของโครงงาน Computerized Automotive Measuring System อันเป็นโครงงานที่ประกอบด้วยโครงงานย่อย ๆ อยู่ 4 โครงงาน คือ

1. โครงงาน Automobile Measuring System

มีหน้าที่ตรวจวัดค่าตัวแปรสภาวะต่าง ๆ ของเครื่องยนต์ และระบบไฟฟ้าที่สำคัญในรถยนต์ แล้วปรับระดับสัญญาณเหล่านั้นให้เป็นสัญญาณมาตรฐาน พร้อมทั้งทำการแปลงข้อมูลซึ่งเป็นแบบอนาลอกให้เป็นดิจิตอล เพื่อที่จะได้ส่งให้โครงงาน central processing unit ต่อไป

2. โครงงาน Central Processing Unit

มีหน้าที่เชื่อมโยงโครงงานที่เหลือทั้งหมดเข้ากับโครงงานนี้ โดยสัญญาณต่าง ๆ ที่ sense มาจากอุณหภูมิของเครื่องยนต์, ความเร็วรอบของเครื่องยนต์, ความเร็วรถแล่น, กระแสและแรงดันไฟฟ้าของแบตเตอรี่, ระบบไฟฟ้าในรถยนต์ ซึ่งสัญญาณเหล่านี้จะถูกแปลงให้เหมาะสมจากโครงงาน Automobile Measuring System ป้อนเข้าสู่โครงงานนี้ โดยสัญญาณเหล่านี้จะถูก Process และส่งไปสู่โครงงาน Graphic Displey System เพื่อแสดงผลบนจอ CRT และในส่วนของสัญญาณที่สำคัญอย่างเช่น สัญญาณเตือนอันเนื่องมาจากการขัดข้องของระบบรถยนต์ไม่ว่าจะเป็นปริมาณน้ำมันของรถยนต์หมด, หรืออุณหภูมิของเครื่องยนต์สูงมาก โครงงานนี้ก็จะส่ง control code พร้อมทั้งคำพูดที่จะให้พูดไปยังโครงงาน Speech Synthesizer

3. โครงงาน Graphic Display System

มีหน้าที่คอยรับcontrol code และข้อความต่าง ๆ ที่ส่งมาจากโครงงาน Central Processing Unit ในรูปรหัส ASCII แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ประมวลผลข้อมูลพร้อมทั้งนำออกไปแสดงผลบนจอ CRT

4. โครงงาน Speech Synthesizer Unit

มีหน้าที่คอยรับ control code และข้อมูลที่เป็นคำพูดที่ส่งมาจากโครงงาน Central Processing Unit แล้วทำการสังเคราะห์คำพูดนั้นและส่งออกลำโพง เพื่อเป็นสัญญาณเตือนหรือบอกกล่าวที่จำเป็น

ผู้จัดทำปริญญานิพนธ์ชุดนี้รับผิดชอบในส่วนของโครงงาน Central Processing Unit ซึ่งจะต้องออกแบบ hardware และ software ที่รองรับการเชื่อมต่อ (interface) ของโครงงานอื่น ๆ

โดยโครงงานนี้ใช้ Microprocessor เบอร์ Z80 เป็นตัวประมวลผล พร้อมทั้งรายละเอียดต่าง ๆ จะกล่าวถึงในบทถัดไป

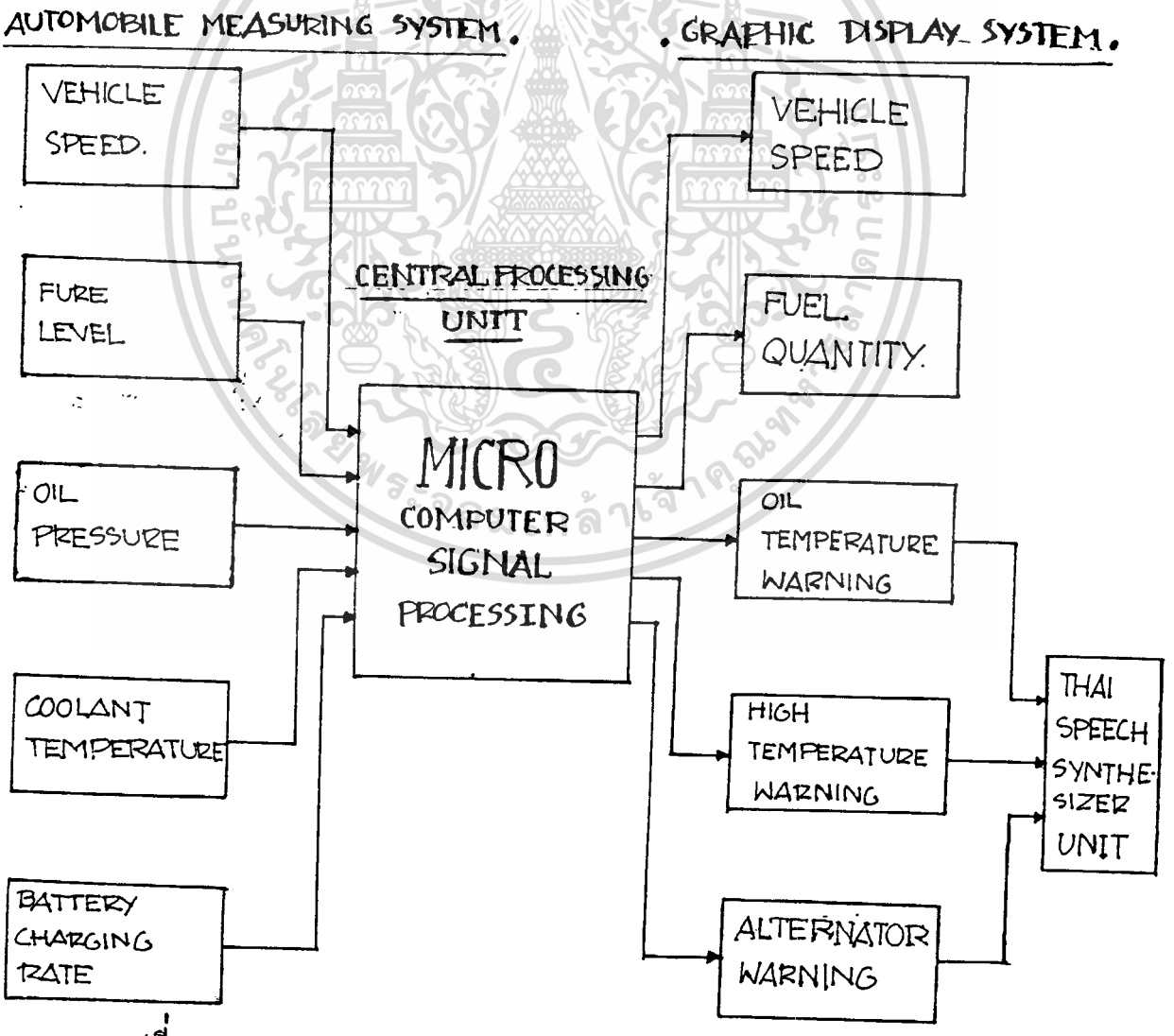
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการเบื้องต้น

เนื่องจากโครงงาน Central Processing Unit นี้จะต้องเชื่อมโยงกับหลายโครงงาน เช่น โครงงาน Automobile Measuring System, โครงงาน Graphic Display System , โครงงาน Speech Synthesizer, ดังนั้น ก่อนที่จะกล่าวถึงโครงงานนี้ จำเป็นต้องกล่าวถึงระบบใหญ่ทั้งหมดก่อน

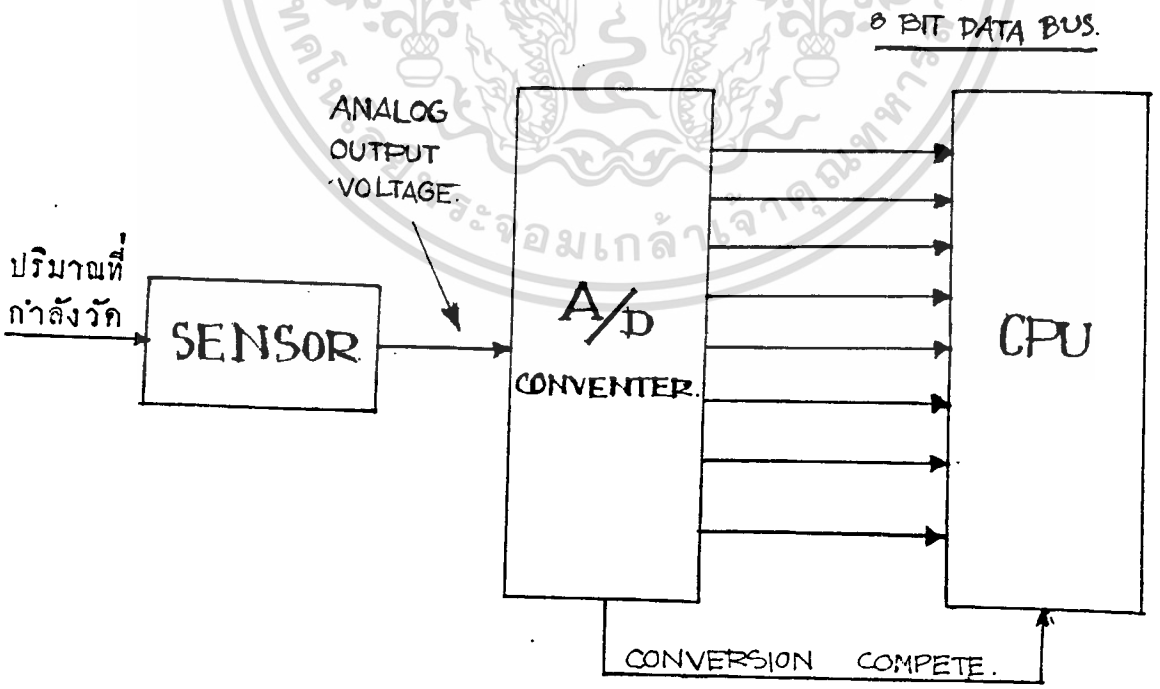


รูปที่ 1 Microcomputer based signal processing in auto car

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1 ในหน้าที่แล้ว ความเร็วรถ, ความเร็วรอบเครื่องยนต์, อุณหภูมิของเครื่องยนต์ระดับน้ำมัน, แรงดันและกระแสไฟฟ้าของ Battery จะถูกเปลี่ยนเป็นสัญญาณทางไฟฟ้าอย่างต่อเนื่อง (ANALOG SIGNAL) โดย sensor car signal measurement และก่อนที่จะส่งสัญญาณนี้ไปประมวลผลจะต้องนำไปผ่าน analog-to-digital converter (A/D) เสียก่อน ซึ่งในโครงงานนี้ใช้ ADC 0809

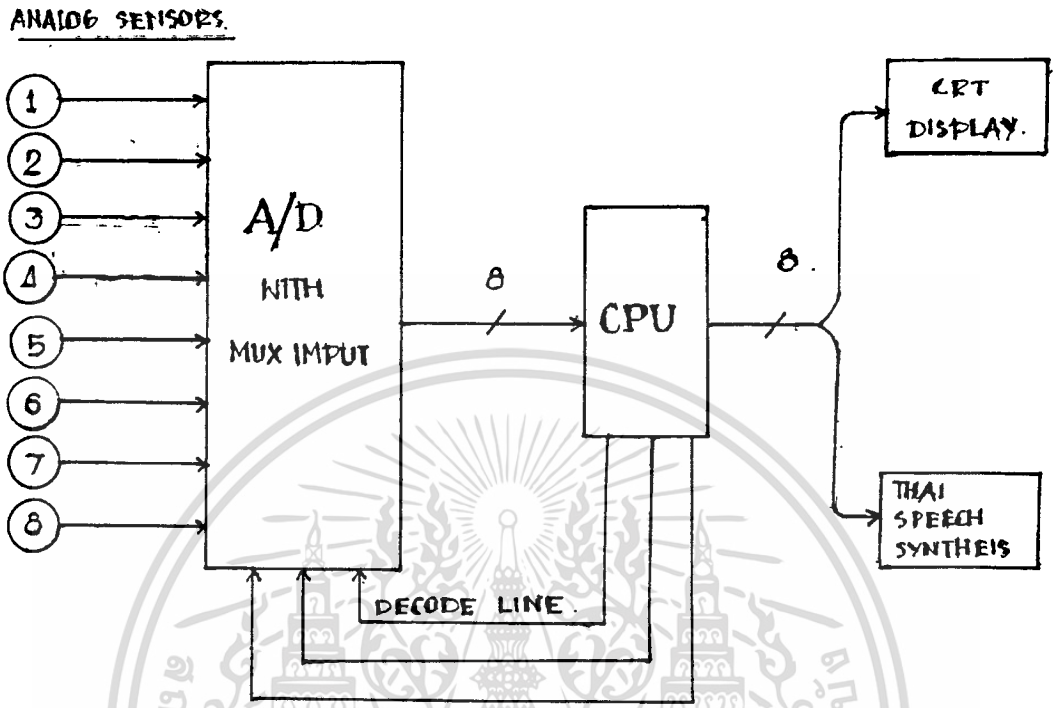
จากรูปที่ 2 สัญญาณที่กำลังวัดจะผ่าน sensor ซึ่งจะได้เป็น analog voltage output ป้อนเข้าสู่ A/D converter ได้เป็น 8 bit data เข้าสู่ CPU แต่ CPU จะคิดต่อในลักษณะเป็น input port ของ CPU แต่สัญญาณที่ตรวจวัดมีหลายช่อง จึงจำเป็นที่ต้องใช้เทคนิคการ MS multiplex เข้าช่วย ซึ่ง ADC 0809 สามารถที่จะ multiplex สัญญาณที่เข้า CPU ได้ $2^3 = 8$ ช่องสัญญาณ เมื่อ CPU รับสัญญาณจาก A/D converter CPU จะคำนวณค่าที่ได้แล้วส่งไปแสดงผลที่ graphic display และ Thai speech synthesizer unit



รูปที่ 2 analog-to-digital conversion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 System Configurations



รูปที่ 3 block diagram ของระบบ

จากรูป 3 เป็น block diagram ของระบบทั้งหมดอย่างง่ายโดย สัญญาณ analog จาก sensor output จะถูก decode โดย CPU ซึ่ง CPU จะมอบ A/D เป็น port ตามตำแหน่งต่าง ๆ (1-8) CPU จะนำเอาค่า analog แต่ละช่องสัญญาณ multiplex เข้าไปสู่ A/D converter ทีละช่องสัญญาณ จากนั้น CPU จะคำนวณผลของสัญญาณแต่ละช่องแล้วจะนำผลที่ได้ไปแสดงที่ CRT หรือถ้าเป็นการเตือน อันเนื่องมาจากการขัดข้องของระบบรถยนต์ก็จะถูกนำไปแสดงที่ Thai speech synthesizer อีกด้วย

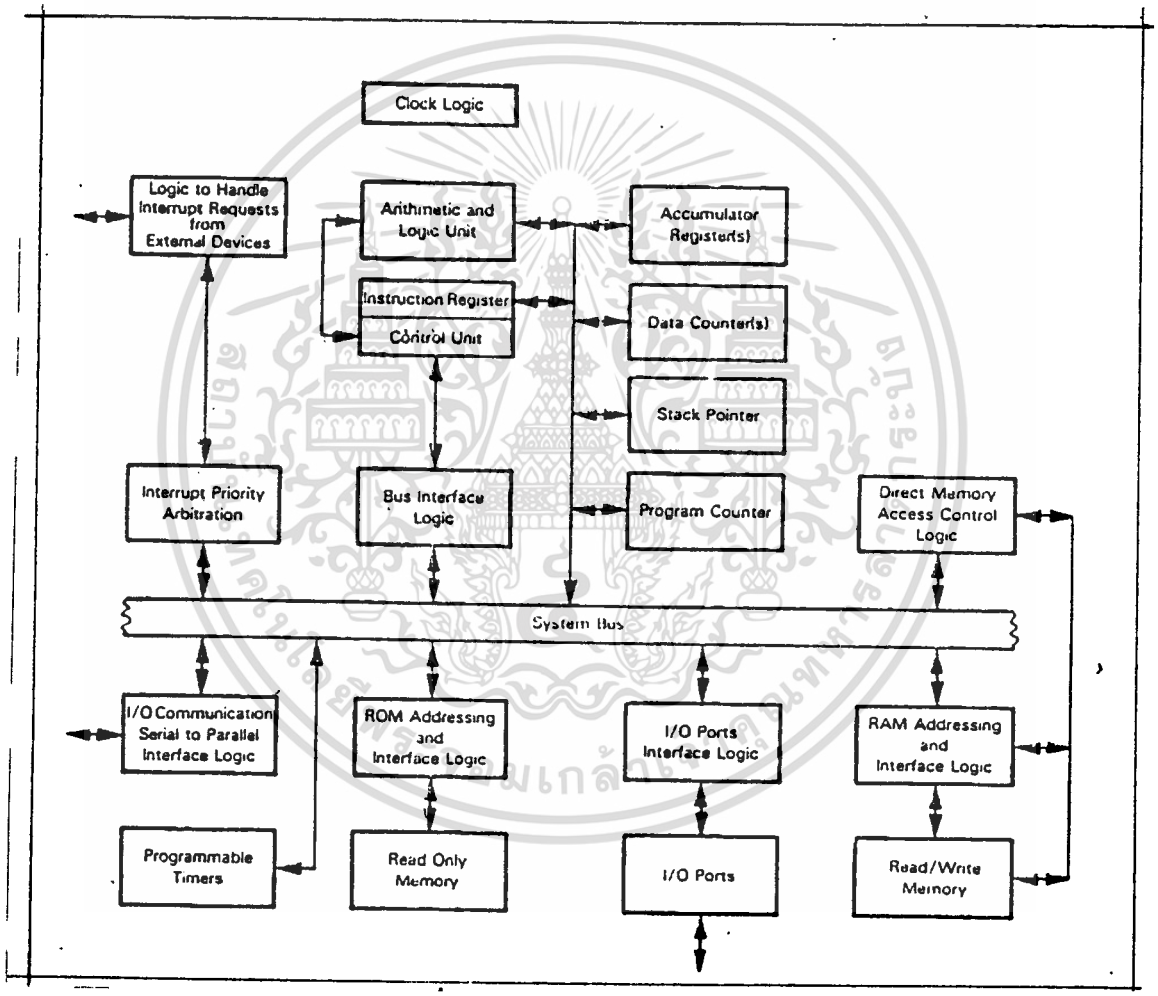
โดยโครงงาน Central processing unit ใช้ Z 80 เป็น CPU ดังนั้น หัวข้อถัดไปจะกล่าวถึงสถาปัตยกรรมและชุดคำสั่งของ Z 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างพอสังเขป เพราะ Z 80 เป็น microprocessor ที่แพร่หลายและรู้จักกันดีอยู่แล้ว อีกทั้งจะกล่าวถึงอุปกรณ์ที่ต่อร่วมกับ Z80 ด้วย ในภาคผนวก

2.3 สถาปัตยกรรมของ Microprocessor เบอร์ Z 80

Microprocessor เบอร์ Z 80 มีโครงสร้างภายในตาม block diagram ข้างล่างนี้ ดูรูปที่ 4



รูปที่ 4 แสดงบล็อกไดอะแกรมของ Z 80

ซึ่งแต่ละบล็อกมีลักษณะการทำงานดังต่อไปนี้คือ

1. Arithmetics logic unit (ALU) เป็นหน่วยที่ทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการคำนวณฟังก์ชันพื้นฐานทางคณิตศาสตร์ และการกระทำฟังก์ชันทางลอจิก เช่น AND และ OR, ALU จะสามารถทำหน้าที่ได้อย่างมีประสิทธิภาพมาก เพียงใดนั้นขึ้นอยู่กับการออกแบบวงจรภายในของ ALU

2. Control unit เป็นหน่วยที่ทำหน้าที่ในการส่งสัญญาณไปควบคุมอุปกรณ์ต่าง ๆ ที่ต่อเชื่อมกับ CPU ให้ทำงานร่วมกันได้อย่างถูกต้อง

3. Data bus เป็นบัสสองทิศทาง (BI-DIRECTIONAL) ที่ใช้ในการส่งผ่านข้อมูลระหว่าง CPU กับอุปกรณ์อื่น ๆ ภายในระบบ จำนวนเส้นของบัสข้อมูล (data bus) จะขึ้นอยู่กับชนิดของ CPU เช่นในกรณีของ Z 80 CPU จะส่งผ่านข้อมูลทีละ 8 บิท ดังนั้นจะมีจำนวนเส้นช่องบัสข้อมูล 8 เส้น

4. Control bus หรือ บัสควบคุม เป็นบัสทางเดียว (UNI-DIRECTIONAL BUS) ที่ใช้ในการส่งผ่านสัญญาณควบคุมให้กับอุปกรณ์ต่าง ๆ ในระบบ

5. Address bus เป็นบัสทางเดียว ใช้ส่งผ่านค่าแอดเดรสจาก CPU ออกไปยังหน่วยความจำเพื่อระบุตำแหน่งที่ต้องการรับหรือส่งข้อมูล หรือใช้ระบุตำแหน่งของพอร์ท I/O (INPUT!OUTPUT PORT) ที่ CPU ต้องการติดต่อด้วย

ต่อไปนี้เราจะกล่าวถึงรายละเอียดของ Z 80 CPU ซึ่งเป็นไมโครโปรเซสเซอร์ขนาด 8 บิทที่นิยมใช้อย่างแพร่หลายในปัจจุบัน

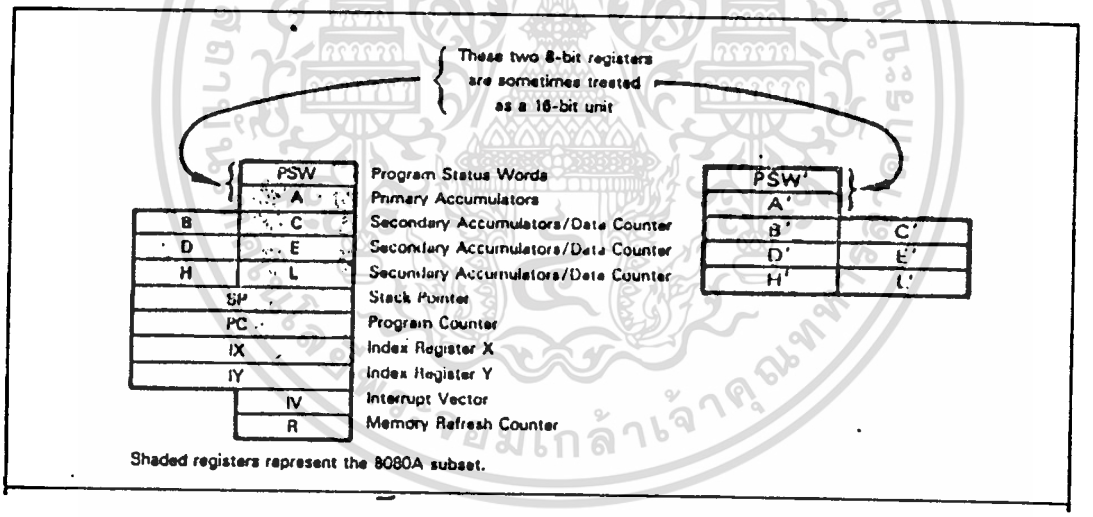
Z80 ไมโครโปรเซสเซอร์เป็น CPU ที่ผลิตจากบริษัท ZILOG INC. โดยกลุ่มวิศวกรชุดเดียวกับที่ผลิต 8080 CPU ของบริษัท INTEL COORPERATION., Z80 CPU ได้รับการพัฒนาให้มีข้อดีเหนือกว่า 8080 เช่น มีชุดคำสั่งมากถึง 158 คำสั่งโดยรวมชุดคำสั่งเดิมของ 8080 ไว้ 80 คำสั่ง นอกจากนี้ Z80 ยังมีรีจิสเตอร์มากกว่าใน 8080 ถึง 12 ตัว และ 8080 เพียงตัวเดียวก็ยังไม่สามารถที่จะนำไปใช้งานได้ต้องต่อกับอุปกรณ์สนับสนุนอีก 2 ตัวคือ clock generator chip, system controler chip รวมกันเรียกว่า THREE CHIP PROCESSOR แต่ใน Z80 CPU ได้รวมเอาลักษณะพื้นฐานเหล่านี้ไว้ในชิพเดียวกัน และเพิ่มประสิทธิภาพทาง HARDWARE, SOFTWARE และการ INTERFACE ให้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80 ไมโครโปรเซสเซอร์เพียงชิ้นเดียวไม่สามารถทำงานเป็นระบบคอมพิวเตอร์ได้ ต้องอาศัยอุปกรณ์อื่น ๆ อีก 2 ส่วน คือ หน่วยความจำ (memory) และหน่วยรับส่งข้อมูลเข้าออก(I/O device) ซึ่งในการทำงานตามคำสั่งจากโปรแกรมที่ป้อนเข้ามา Z80 CPU จะต้องทำการโอนย้ายคำสั่งหรือข้อมูลระหว่างหน่วยความจำกับรีจิสเตอร์ (register) ก่อนอื่นเราจะกล่าวถึงรีจิสเตอร์ภายในของ Z80 เสียก่อน

2.4 รีจิสเตอร์ต่าง ๆ ใน Z80 CPU

Z80 CPU จะประกอบไปด้วยรีจิสเตอร์ถึง 22 ตัวดังรูป 5 รีจิสเตอร์เหล่านี้จะแบ่งได้เป็น 2 กลุ่มคือ รีจิสเตอร์ที่ทำหน้าที่ทั่ว ๆ ไป และรีจิสเตอร์ที่ทำหน้าที่เฉพาะงาน



รูปที่ 5 แสดงรีจิสเตอร์ต่าง ๆ ภายใน

1. รีจิสเตอร์ที่ทำหน้าที่ทั่ว ๆ ไป แบ่งเป็นรีจิสเตอร์หลัก ได้แก่ A, B, C, D, E, H และ L มีความจุขนาด 8 บิท รีจิสเตอร์เหล่านี้ใช้เก็บข้อมูลชั่วคราว นอกจากนี้ยังสามารถรับข้อมูลจากหน่วยความจำหรืออาจจะทำการย้ายข้อมูลไปเก็บไว้ในหน่วยความจำก็ได้ และ รีจิสเตอร์สำรอง ได้แก่ A1,B1,C1,D1,E1,H1 และ L1 ซึ่งเป็นรีจิสเตอร์ที่ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์หลักในกรณีที่ต้องใช้รีจิสเตอร์หลักในการทำงานอย่างอื่นก่อน ดังนั้นรีจิสเตอร์กลุ่มนี้จึงไม่สามารถกระทำขบวนการทางคณิตศาสตร์และลอจิกได้

รีจิสเตอร์A เรียกว่า แอคคิวมูเลเตอร์ (Accumulator) ทำหน้าที่เก็บข้อมูลชั่วคราวที่ได้จากการทำขบวนการทางคณิตศาสตร์ เช่น บวกหรือลบข้อมูล 2 จำนวน ผลลัพธ์ที่ได้จะเก็บไว้ในรีจิสเตอร์ A นี้ นอกจากนี้ ในการปฏิบัติตามคำสั่งที่ใช้กับข้อมูลขนาด 16 บิท Z80 จะนำเอารีจิสเตอร์แฟลก "F" (FLAG REGISTER) มาใช้รวมกับรีจิสเตอร์ A เรียกว่า คู่รีจิสเตอร์ AF ซึ่งมีขนาด 16 บิท นอกจากนี้ยังมีคู่รีจิสเตอร์ 16 บิท อื่น ๆ อีก คือ BC, DE และ HL

2. รีจิสเตอร์ที่ใช้งานเฉพาะอย่าง ได้แก่ รีจิสเตอร์ I, R, IX, IY, SP และ PC ซึ่งทำหน้าที่ต่าง ๆ ดังนี้

รีจิสเตอร์ I (Interrupt page address register) เมื่อมีการอินเทอร์รัพท์เกิดขึ้นจำเป็นต้องบอกตำแหน่งของหน่วยความจำที่เก็บโปรแกรมตอบสนองการอินเทอร์รัพท์ รีจิสเตอร์ I จะทำหน้าที่เก็บค่า 8 บิท ของตำแหน่งข้อมูลในหน่วยความจำ ส่วนค่า 8 บิทล่างจะป้อนจากภายนอกให้แก่

ค่าทั้งสองจะประกอบกันเป็นค่าแอดเดรสที่ระบุตำแหน่งของโปรแกรมการตอบสนองการอินเทอร์รัพท์

รีจิสเตอร์ R (memory refresh register) เป็นรีจิสเตอร์ขนาด 7 บิทที่ถูกใช้ในการรีเฟรช (refresh) Dynamic ram และค่ารีจิสเตอร์ R จะเพิ่มขึ้นเองโดยอัตโนมัติ ในทุก ๆ ครั้งที่มีการเฟทช์คำสั่งจากหน่วยความจำ รายละเอียดเกี่ยวกับ dynamic ram จะกล่าวถึงในบทที่ 5

รีจิสเตอร์ IX และ IY (Index register) เป็นรีจิสเตอร์ที่มีขนาด 16 บิท มีประโยชน์ใช้บ่งบอกตำแหน่งในหน่วยความจำแบบ Index addressing mode โดยจะกำหนดให้ค่าใน Index register เป็นค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
024731

อ้างอิง แล้วใช้คำสั่งบงบอกว่าตำแหน่งของข้อมูลที่ต้องการอยู่ห่างจากค่าอ้างอิงนี้เท่าใด โดยจะบอกค่าระยะห่างในรูปของ Two complement

รีจิสเตอร์ SP (Stack pointer) มีขนาด 16 บิท ในหน่วยความจำชนิด RAMจะมีส่วนหนึ่งที่ถูกกำหนดให้เป็นที่เก็บข้อมูลชั่วคราว ส่วนนี้เรียกว่าสแตก (stack) ซึ่งมีลักษณะการเก็บข้อมูลแบบ LIFO (Last in first out) เราสามารถที่จะเก็บข้อมูลลงบนสแตกโดยใช้คำสั่ง push และเมื่อต้องการดึงข้อมูลออกจากสแตกต้องใช้คำสั่ง POP

รีจิสเตอร์ PC (program counter) เป็นรีจิสเตอร์ขนาด 16 บิทที่ใช้ในการเก็บตำแหน่งของหน่วยความจำที่CPU จะเฟทช์ (fetch) คำสั่งหลังจากที่เฟทช์คำสั่งเรียบร้อยแล้ว ค่าในรีจิสเตอร์ PC จะเพิ่มขึ้น และจะชี้ไปยังตำแหน่งของคำสั่งถัดไป เราจะสามารถเปลี่ยนแปลงค่าใน PC ได้โดยใช้คำสั่ง CALL หรือ JUMP

รีจิสเตอร์ F (Flag register) ประกอบด้วย

sign flag (S) : แฟลกเครื่องหมาย

zero flag (Z) : แฟลกศูนย์

half carry flag (H) : แฟลกทดครึ่ง

parity/overflow flag (P/V) : แฟลกพาริตี้หรือโอเวอร์โฟล์ว

subtract flag (N) : แฟลกลบ

carry flag (C) : แฟลกตัวทด

ผู้ผลิต Z80 ได้นำเอาแฟลกเหล่านี้มาประกอบร่วมกับ บิทว่าง (X :ไม่มีความหมาย) อีก 2 บิทเพื่อทำเป็นรีจิสเตอร์ขนาด 8 บิท สำหรับรายละเอียดของแฟลกเหล่านี้จะไม่ขอกล่าวถึง

2.5 รายละเอียดของขา Z80 (Z80 PIN OUTS)

A0-A15 (Address bus) : เป็นขาสัญญาณเอาท์พุทแบบ TRI-State ใช้บงบอกตำแหน่งหน่วยความจำได้ถึง $2^{16} = 65536$ ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A0-A7 จะแสดงตำแหน่งของพอร์ทที่ Z80 ต้องการติดต่อด้วย นอกจากนี้ขา A0-A6 จะให้ค่ารีเฟรชแอดเดรสออกมาขณะที่ Z80 ให้สัญญาณรีเฟรช

D0-D7 (Data bus) : เป็นขาสัญญาณอินพุท/เอาท์พุท TRI-STATE แบบสองทิศทางซึ่งเป็นทางผ่านของข้อมูลระหว่าง Z80 กับหน่วยความจำและอุปกรณ์ I/O

$\overline{M1}$ (machine cycle one) : เป็นขาเอาท์พุทแอคทีฟที่ลอจิก "0" ขา $\overline{M1}$ นี้จะแอคทีฟขณะที่ Z80 ทำการเฟทช์ออฟโคดของคำสั่ง ในกรณีที่คำสั่งที่จะเฟทช์เข้ามานั้นมีขนาด 2 ไบท์ $\overline{M1}$ จะแอคทีฟในทุก ๆ ไซเคิลการเฟทช์แต่ละไบท์

$\overline{MREQ}$ (memory request) : เป็นสายเอาท์พุทแบบ TRI-STATE แอคทีฟที่ลอจิก "0" เพื่อเป็นการบ่งบอกว่า Z80 กำลังกระทำการติดต่อกับหน่วยความจำ

$\overline{IORQ}$ (Input/output request) : เป็นสายเอาท์พุทแบบ TRI-STATE จะแอคทีฟที่ลอจิก "0" เพื่อเป็นการบ่งบอกว่า Z80 กำลังทำการติดต่อกับอุปกรณ์ I/O และเมื่อ $\overline{IORQ}$ และ $\overline{M1}$ แอคทีฟทั้งคู่จะเป็นการบ่งบอกการตอบรับการอินเทอร์รัพท์ (Interrupt Acknowledge)

$\overline{RD}$ (memory resd) : เป็นขาเอาท์พุท TRI-STATE จะแอคทีฟที่ลอจิก "0" เมื่อ Z80 ต้องการอ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์ I/O และ Z80 จะรับข้อมูลจากบัสข้อมูลเข้าไปเมื่อสัญญาณนี้เปลี่ยนระดับลอจิกจาก "0" เป็น "1"

$\overline{WR}$ (memory write) : เป็นขาเอาท์พุทแบบ TRI-STATE จะแอคทีฟที่ลอจิก "0" เมื่อ Z80 ต้องการส่งข้อมูลออกไปให้หน่วยความจำหรืออุปกรณ์ I/O

$\overline{RFSH}$ (Refresh): เป็นขาเอาท์พุท จะแอคทีฟ เมื่อ 7 บิทล่าง (A0-A6) ของบัสแอดเดรสให้ค่ารีเฟรชออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{HALT}}$ (halt state) : เป็นขาเอาท์พุท จะแอคทีพที่ลอจิก "0" เมื่อ Z80 อยู่ในสภาวะของการ $\overline{\text{HALT}}$ คือ CPU จะทำคำสั่ง NOP (NO Operation) เพื่อให้เกิดการรีเฟรชได้และ Z80 จะหลุดพ้นจากสภาวะการ HALT เมื่อได้รับการรีเซ็ทหรือถูกอินเทอร์รัพท์

$\overline{\text{WAIT}}$: เป็นขาอินพุทแอคทีพที่ลอจิก "0" และจะมีการตรวจสอบสัญญาณนี้ที่ขอบขาลงของคล็อกลูกที่ 2 ของทุก ๆ machine cycle เมื่อมีการตรวจพบว่าขาอินพุทนี้แอคทีพจะมีการแทรก wait state ให้กับแต่ละ machine cycle เพื่อเป็นการรอให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ Z80 และ Z80 จะแทรก wait state จนกว่าจะมีการตรวจสอบพบว่าขา wait จะมีลอจิกเป็น "1"

$\overline{\text{INT}}$ (Interrupt request) : เป็นขาอินพุทแอคทีพที่ลอจิก "0" Z80 จะตรวจสอบระดับสัญญาณที่ขานี้ทุก ๆ การสิ้นสุดของ Instruction cycle (last state)

$\overline{\text{NMI}}$ (non maskable interrupt) : เป็นขาอินพุทแอคทีพที่ลอจิก "0" สัญญาณ non maskable interrupt เป็นสัญญาณที่มีระดับความสำคัญในการขออินเทอร์รัพท์สูงกว่าสัญญาณ interrupt request Z80 จะตอบรับการอินเทอร์รัพท์ชนิดนี้เสมอโดยที่เราไม่สามารถ disable ได้ด้วย software

$\overline{\text{RESET}}$: เป็นขาอินพุทแอคทีพที่ลอจิก "0" สัญญาณนี้จะทำการ initialize CPU โดยทำการรีเซ็ท interrupt flip-flop และเซ็ทค่าในโปรแกรมเคาน์เตอร์ (program counter) ให้เป็น 0000H และในสภาวะการรีเซ็ทนี้ บัสแอดเดรสและบัสข้อมูลจะอยู่ในสภาวะ high impedance และสัญญาณควบคุมต่าง ๆ จะอยู่ในสภาวะ Inactive

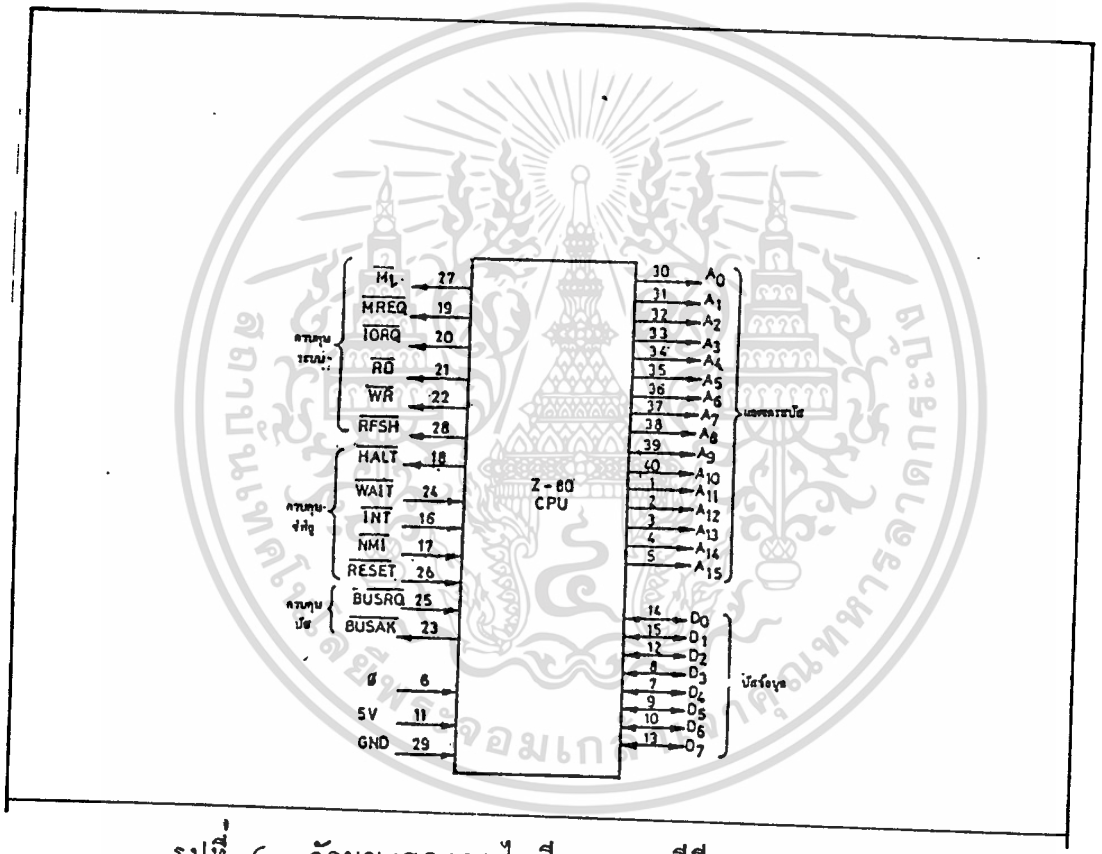
$\overline{\text{BUSRQ}}$ (bus request) : เป็นขาอินพุทแอคทีพที่ลอจิก "0" สัญญาณ bus request เป็นสัญญาณที่มีลำดับความสำคัญสูงกว่าสัญญาณ non maskable interrupt และมีการตรวจสอบสัญญาณนี้ทุก ๆ การสิ้นสุดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

machine cycle อุปกรณ์ภายนอกจะให้สัญญาณนี้แก่ Z80 เมื่อต้องการใช้บัสข้อมูลและบัสแอดเดรส โดยเปรียบเสมือนว่าเป็นการถอด Z80 ออกจากระบบบัส

$\overline{\text{BUSAK}}$ (bus acknowledge) : เป็นขาเอาท์พุทแอคทีพที่ลอจิก "0" ขานี้จะแอคทีพเมื่อ Z80 ตอบสนองการต่อสัญญาณ bus request และจะทำให้บัสข้อมูล บัสควบคุมและบัสแอดเดรสมีสภาวะเป็น high impedance ซึ่งทำให้อุปกรณ์ภายนอกใช้บัสเหล่านี้ได้โดยไม่มีผลต่อ CPU



รูปที่ 6 ลักษณะของขาไอซี Z80 ซีพียู

2.6 ไดอะแกรมเวลาของซีพียู

Z80 ซีพียูจะทำงานในลักษณะพื้นฐาน ที่สำคัญประกอบด้วย

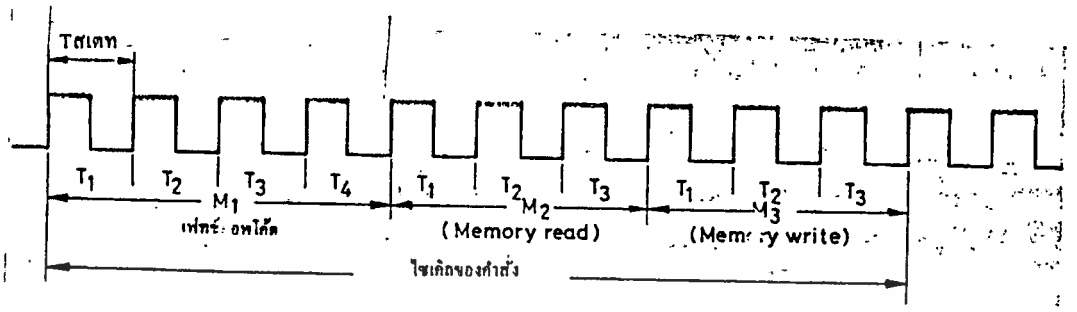
การเขียน–อ่านหน่วยความจำ

การเขียน–อ่านอุปกรณ์อินพุท–เอาท์พุท

การตอบสนองต่อการขัดจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการทำงานจะสัมพันธ์กับสัญญาณนาฬิกา ขอให้พิจารณาจากรูปที่ 7



รูปที่ 7

ไซเคิลการทำงานที่แสดงในรูปที่ 7 จะเห็นได้ว่าในหนึ่งไซเคิลของคำสั่งจะประกอบด้วยหลายแมชีนไซเคิล (M ไซเคิล) กล่าวคือ M_1, M_2, M_3... M_5 นั่นเอง ในกรณีของ Z80 จำนวนแมชีนไซเคิลของแต่ละคำสั่งไม่เท่ากันมากบ้างน้อยบ้าง ที่น้อยที่สุดใช้ 1 แมชีนไซเคิล ส่วนที่มากที่สุดใช้ 5 แมชีน-ไซเคิล คำสั่งที่มีสภาวะการทำงานแต่เพียงภายในซีพียู โดยเมื่อเฟทช์คำสั่งจากหน่วยความจำมาถึงความแล้ว ก็สามารถทำได้เลย เช่น LD r_1, r_2 เป็นต้น จะใช้เพียง 1 แมชีนไซเคิล

เนื่องจากในแต่ละคำสั่งจะประกอบด้วยแมชีนไซเคิลหลายแบบด้วยกัน แต่แมชีนไซเคิลที่ใช้ทั้งหมดสามารถสรุปออกมาเป็นแบบอย่างได้เป็น

1. ไซเคิลการเฟทช์ (M_1)
2. ไซเคิลการเขียนหรืออ่านหน่วยความจำ
3. ไซเคิลการรับส่งข้อมูลอินพุทหรือเอาท์พุท
4. ไซเคิลการต้องการใช้บัส หรือการตอบสนองการใช้บัส
5. ไซเคิลการอินเตอร์รัพท์ และการตอบสนองต่อการอินเตอร์รัพท์
6. ไซเคิลการรับอินเตอร์รัพท์แบบนอนมาสค์เคเบิ้ล
7. ไซเคิลการออกจากคำสั่ง HALT

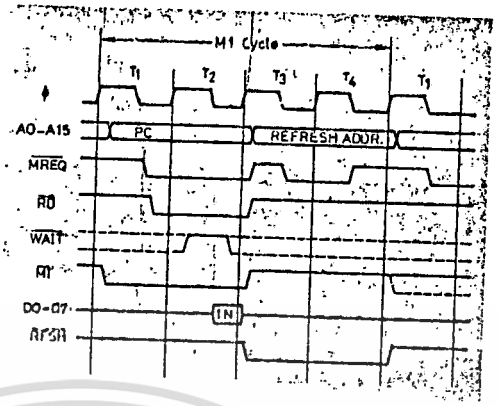
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในแต่ละไซเคิลจะตอบสนองได้ตามลักษณะของคำสั่งที่กระทำนั่นเอง คราวนี้ลองมาพิจารณารายละเอียดของแต่ละไซเคิลดูรายละเอียดต่อไป

2.7 ไซเคิลการเฟทช์ออพโค้ด (OCF-opcode fetch)

ในสภาวะ M_1 นี้เป็นภาวะเริ่มแรกของทุกคำสั่ง ในสภาวะ M_1 นี้ขอให้สังเกตจากระบบไดอะแกรมเวลาจะเห็นสภาวะการทำงานของซีพียูได้ดีกล่าวคือ เมื่อเริ่มเข้าสู่สภาวะเฟทช์ ข้อมูลจากโปรแกรมเคาน์เตอร์จะถายลงไปยังแอดเดรสบัสและหลังจากครึ่งไซเคิลของสัญญาณนาฬิกา ซีพียูก็จะส่งสัญญาณ $\overline{MREQ}$ ซึ่งเป็นสัญญาณบอกว่าซีพียูต้องการติดต่อกับหน่วยความจำ และขณะเดียวกันก็ส่งสัญญาณ $\overline{RD}$ บอกหน่วยความจำว่าต้องการ $\overline{RD}$ หรืออ่านข้อมูลในหน่วยความจำตามแอดเดรสที่ต้องการ ครั้นเมื่อถึงขาลงของ T_2 ซีพียูจะตรวจสอบสัญญาณที่ทางขา $\overline{WAIT}$ ว่าเป็น "0" หรือไม่ถ้าไม่ก็จะทำงานต่อแบบปกติ แต่ถ้าขา $\overline{WAIT}$ เป็น "0" ซีพียูจะเพิ่มสภาวะการรอ (WAIT state) เกิดขึ้น การอ่านข้อมูลจากหน่วยความจำในที่นี้ ถ้านำเอาสัญญาณ $\overline{MREQ}$ และ $\overline{RD}$ เป็นตัวอ่าน ข้อมูลที่บัสข้อมูลก็จะมีข้อมูลที่อ่านได้ในขณะที่ $\overline{MREQ}$ OR $\overline{RD}$ เป็น "0" ทั้งคู่ แต่ซีพียูจะรับการเปิดบัสนี้เข้าใน IR เพียงชั่วขณะสั้นเท่านั้น (ดูรูป) เมื่อข้อมูลจากบัสเข้า IR เรียบร้อย แล้วก็จบสภาวะการเฟทช์ สัญญาณที่ขา $\overline{M_1}$ ก็จะเป็น "1" บอกว่าเฟทช์เสร็จแล้ว แต่อย่างไรก็ตามสภาวะ $\overline{M_1}$ ไซเคิลยังไม่หมดและซีพียูจะส่งสัญญาณทำการรีเฟรชหน่วยความจำชนิดไดนามิกส์นั่นคือ ในแอดเดรสบัสจะมีข้อมูลของแอดเดรสที่ต้องการรีเฟรชและขา $\overline{RFSH}$ ก็จะบอกสภาวะว่าต้องการรีเฟรชนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

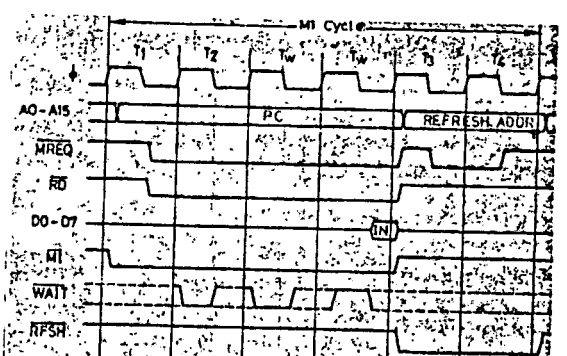


รูปที่ 8

ไซเคิลการเฟทช์ (M_1 Cycle)

2.8 M_1 ไซเคิลที่มี WAIT STATES

รูปที่ 9 เป็นสภาวะของ M_1 ไซเคิลที่มีการ $\overline{WAIT}$ กล่าวคือ ทุก ๆ ขาลงของ T_2 จะมีการตรวจสอบที่ขา $\overline{WAIT}$ ว่าเป็น "0" หรือไม่ สมมุติว่าเป็น "0" ซีพียูจะสร้างสเตทการ $\overline{WAIT}$ อีก 1 ไซเคิล โดยที่ค่าสัญญาณอื่น ๆ ไว้คังเดิม และทุก ๆ ขาลงของ T_w ก็จะตรวจสอบขา $\overline{WAIT}$ อีกเช่นกัน ถ้าขา $\overline{WAIT}$ เป็น "0" ก็จะสราง T_w ต่อไปอีกและเป็นเช่นนี้เรื่อยไปไม่สิ้นสุด ในขณะที่ซีพียูจะสร้างสัญญาณ T_w สัญญาณขาต่าง ๆ จะยังคงค่าเดิมไว้หมด สัญญาณต่าง ๆ เหล่านี้ได้แก A_0- A_{15} , $\overline{MREQ}$, $\overline{RD}$, $\overline{M}_1$ และ $\overline{RFSH}$ ครั้นเมื่อขาลงของ T_w ซีพียูตรวจสอบสัญญาณ $\overline{WAIT}$ พบลอจิก "1" มันจะเริ่มการอ่านข้อมูลจากบัสข้อมูลและต่อท้ายด้วย T_3 และ T_4 ต่อไป



รูปที่ 9

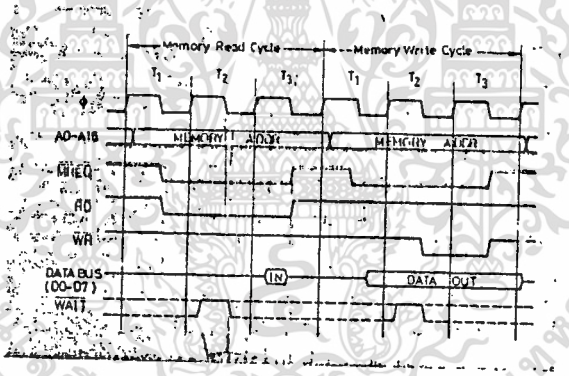
ไซเคิลการเฟทช์ที่มี WAIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ของการใช้ T_w ก็ด้วยเหตุผลในการยืดความกว้างของสัญญาณต่าง ๆ เพื่อซิงโครนัส หรือให้ทำงานร่วมกับอุปกรณ์ภายนอกที่ความเร็วช้าได้ เช่น ถ้าต้องการอ่านหน่วยความจำที่มีความเร็วต่ำ เราจำเป็นต้องยืดสัญญาณ $\overline{RD}$ และ $\overline{MERQ}$ ออกเพื่อให้ส่งไปควบคุมหน่วยความจำได้ เราก็สร้างสัญญาณ $\overline{WAIT}$ ซีพียูนั่นเอง

2.9 <u>ไซเคิลการอ่านหน่วยความจำ (MR-memory read cycle) และไซเคิลการเขียนหน่วยความจำ (MW-memory write cycle)</u>

รูปที่ 10 แสดงไดอะแกรมของการอ่านและเขียนหน่วยความจำ ในการอ่านหรือการเขียนหน่วยความจำนี้ ซีพียูจะใช้เวลาทั้งสิ้น 3 ไซเคิล สัญญาณนาฬิกาคือ T_1 ,T_2 ,T_3



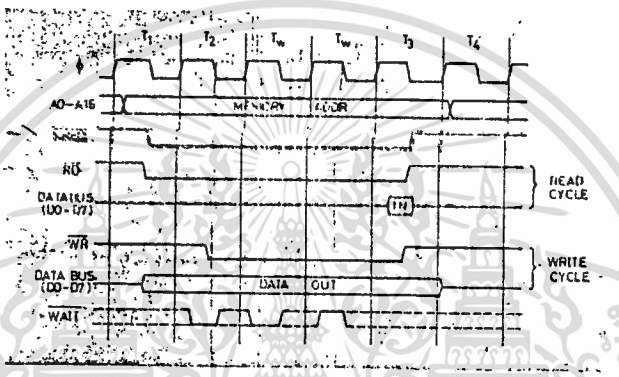
<u>รูปที่ 10</u> ไดอะแกรมเวลา การอ่านและการเขียนหน่วยความจำ

ในการอ่านหน่วยความจำนั้นซีพียูจะส่งแอดเดรสที่ต้องการอ่านเข้าสู่แอดเดรสบัส A_0- A_{15} ก่อนในตอนแรกของแมชีนไซเคิลนี้ หลังจากนั้นต่อมาก็ส่งสัญญาณ $\overline{MREQ}$ และ $\overline{RD}$ ออกมาและทุกครั้งของขาลงในช่วงจังหวะ T_2 ซีพียูจะตรวจสอบขา $\overline{WAIT}$ ว่าเป็นลอจิก "0" หรือไม่ ถ้าไม่ซีพียูก็จะทำงานตามปกติ แต่ถ้าเป็น "0" ก็จะสร้าง T_w เพิ่มขึ้น ช่วงจังหวะที่แอดเดรสบัสและสัญญาณ $\overline{MERQ}$ +$\overline{RD}$ ไปยังหน่วยความจำจะมีข้อมูลออกจากหน่วยความจำลงมาในบัสข้อมูลแล้ว แต่ซีพียูยังไม่อ่านจังหวะการอ่านของซีพียูจะเกิดขึ้นในจังหวะของขาลงของ T_3 ช่วงการอ่านของซีพียูจะใช้เวลาสั้น ๆ เพื่อป้องกันการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลผิดพลาดคือ ผิดตำแหน่งแอดเดรสหรืออ่านได้หลายแอดเดรสนั่นเอง

ในทำนองเดียวกันในจังหวะของ MW ก็เช่นกันประกอบด้วย T_1 T_2 และ T_3 (กรณีไม่มี T_w) ในที่นี้ซีพียูจะสร้างสัญญาณ $\overline{MREQ}$ กับ $\overline{WR}$ ให้แอคดีฟ อย่างไรก็ตามซีพียูจะส่งข้อมูลออกไปรอไว้ที่บัสข้อมูลก่อนและยาวนานกว่า พัลซ์การเขียน ($\overline{WR}$) การเขียนข้อมูล ($\overline{WR}$) จะใช้เวลาเพียงสั้น ๆ ในจังหวะที่ข้อมูลมารออยู่ที่บัสเรียบร้อยแล้ว



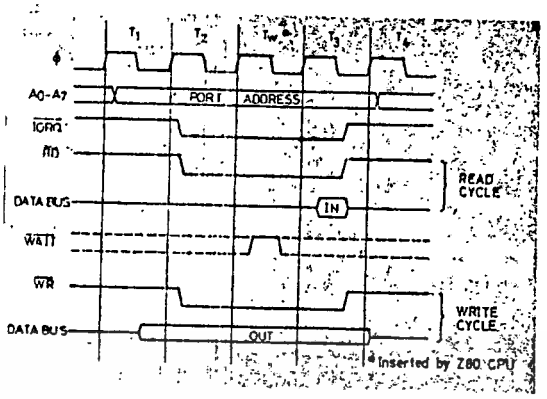
รูปที่ 11 ไดอะแกรมเวลาการอ่านและเขียนข้อมูลที่มี T_w

ในกรณีที่ต้องการหน่วยเวลาการทำงานในแมชีนไซเคิลเหล่านี้สามารถทำได้ด้วยการสร้างขา $\overline{WAIT}$ ให้มีค่าเป็น "0" ในขณะขาลงของ T_2 ซึ่งซีพียูจะสร้าง T_w ขึ้นตราบเท่าที่ขา $\overline{WAIT}$ ยังคงเป็น "0" อยู่

2.10 แมชีนไซเคิลของการเขียนเอาท์พุทและอ่านอินพุท (PW-port write and PR-port read)

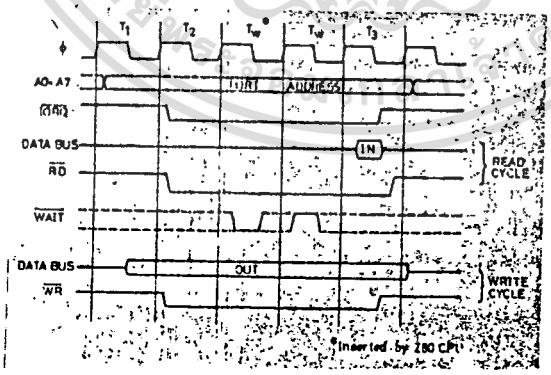
ในการเขียนเอาท์พุทและอ่านอินพุทก็คล้ายคลึงกับเขียนและอ่านหน่วยความจำนั่นเอง โดยการเปลี่ยนสัญญาณ $\overline{MREQ}$ เป็น $\overline{IORQ}$ เท่านั้น ไดอะแกรมการเขียนเอาท์พุทและอ่านอินพุทแสดงให้เห็นดังรูปที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 การเขียนเอาท์พุทและอ่านอินพุท

ข้อแตกต่างที่เห็นได้ชัดเมื่อเทียบกับการเขียนและอ่านหน่วยความจำอีกข้อหนึ่ง ส่วนของแอดเดรสนั้นจะแสดงเบอร์พอร์ทในส่วนของ A_0-A_7 เท่านั้น และในการทำงานแบบปกติซีพียูจะสร้าง Tw แทรกเข้ามา 1 ลูก โดยอัตโนมัติ แม้จะไม่มีการ $\overline{WAIT}$ ทั้งนี้เพราะช่วงจังหวะ $\overline{IORQ}$ นั้นซีพียูสร้างในขณะ T_2 และซีพียูจะไม่สามารถตรวจสอบจังหวะ $\overline{WAIT}$ ในขณะขาลงของ T_2 ได้ การตรวจสอบว่า $\overline{WAIT}$ จะเกิดขึ้นในจังหวะของ Tw แทนนั่นเอง และถ้าหากมีการ $\overline{WAIT}$ ซีพียูก็จะสร้างสัญญาณ Tw ต่อมาอีกเช่นเดียวกับที่กล่าวแล้ว

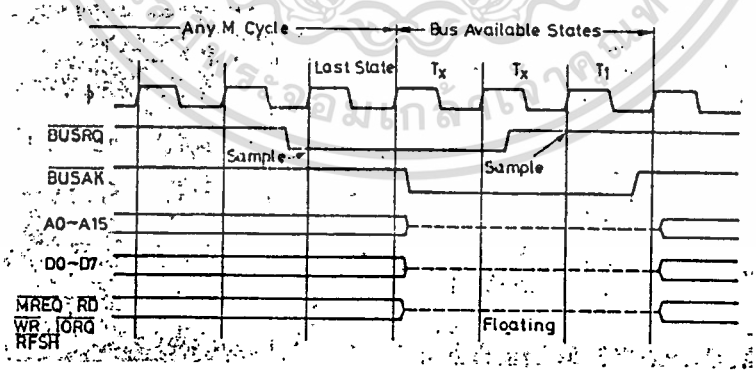


รูปที่ 13 การเขียนเอาท์พุทและอ่านอินพุทขณะมีการ WAIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 ไซเคิลการขอใช้บัสและการตอบสนองการให้ใช้บัส (BUS REQUEST and ACKNOWLEDGE CYCLE)

รูปที่ 14 เป็นไดอะแกรมเวลาของการขอใช้บัสและการตอบสนองต่อการใช้บัส การใช้บัสนั้นจะใช้สัญญาณที่ส่งเข้ามาขอทางขา $\overline{\text{BUSRQ}}$ ซีพียูจะทำการตรวจสอบขา $\overline{\text{BUSRQ}}$ ในขณะสัญญาณนาฬิกากำลังจะเปลี่ยนจาก "0" ไปเป็น "1" ในขณะสัญญาณสเตทสุดท้ายของทุก ๆ M ไซเคิล ถ้า $\overline{\text{BUSRQ}}$ แอคตีฟ ซีพียูจะทำการเซทแอดเดรสบัส และบัสข้อมูลให้อยู่ในสถานะไตรสเตท ในสัญญาณนาฬิกาลูกต่อมา และซีพียูก็จะส่งสัญญาณตอบรับการขอใช้บัส ($\overline{\text{BUSAK}}$) เมื่อซีพียูควบคุมไตรสเตทแล้ว อุปกรณ์ภายนอกก็จะใช้บัสได้ ซึ่งอาจจะเป็นกรรมวิธีการ DMA (direct memory access) หรือการติดต่อระหว่างอุปกรณ์ I/O กับหน่วยความจำโดยตรงได้ กรณีเช่นนี้ทำให้การถ่ายข้อมูลเป็นไปได้เร็วกว่าใช้ซีพียูควบคุมโดยตรง การถ่ายเทข้อมูลในกรณีนี้มักใช้สำหรับการถ่ายข้อมูลเป็นจำนวนมาก ขณะมีการขอใช้บัสนี้ ซีพียูจะตรวจสอบขา $\overline{\text{BUSRQ}}$ ทุก ๆ สเตทเช่นกัน และถ้า $\overline{\text{BUSRQ}}$ เป็น "1" เมื่อใดในสัญญาณนาฬิกาสเตทต่อไปก็จะส่ง $\overline{\text{BUSAK}}$ = "1" เพื่อบอกการเลิกใช้บัส และการควบคุมไตรสเตทก็หมดสิ้นภาระกิจ



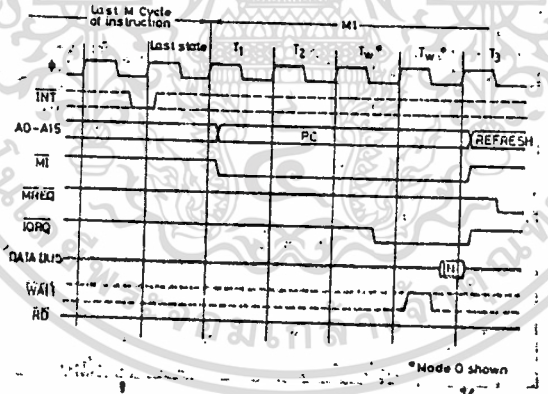
รูปที่ 14 ไดอะแกรมการขอใช้บัสและการตอบสนองต่อการใช้บัส

ในขณะการใช้บัสในกรณีนี้ซีพียูไม่สามารถตอบสนองต่อการอินเตอร์รัพท์ได้ ไม่ว่าจะเป็นการอินเตอร์รัพท์แบบ $\overline{\text{NMI}}$ หรือ $\overline{\text{INT}}$ ก็ตาม

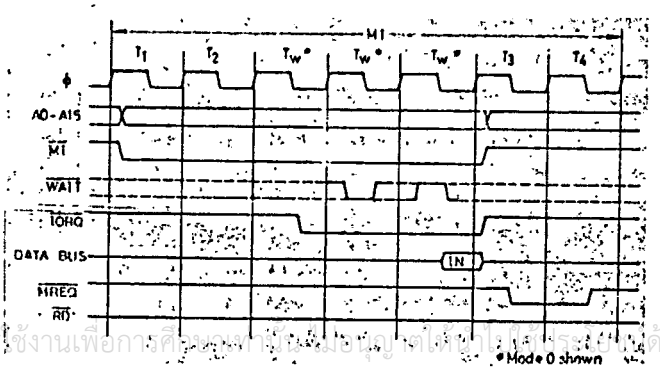
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ไซเคิลการรับสัญญาณอินเตอร์รัพท์และการตอบสนองต่อสัญญาณอินเตอร์รัพท์
(Interrupt request and acknowledge cycle)

รูปที่ 15 เป็นการแสดงไดอะแกรมเวลาของการอินเตอร์รัพท์ซีพียู ซีพียูจะทำการตรวจสอบสัญญาณ $\overline{INT}$ ทุก ๆ พัลซ์สัญญาณนาฬิกาลูกสุดท้ายของทุกคำสั่งที่กระทำ แต่อย่างไรก็ตามการตรวจสอบนี้จะไม่เกิดผลหากซอฟท์แวร์ได้สั่ง DI (Disable interrupte) ซึ่งจะทำให้อินเตอร์รัพท์ลิบฟลอบภายในไม่ได้รับการเซท นอกจากนี้ถ้าสัญญาณ $\overline{BUSRQ}$ แอคตีฟ การอินเตอร์รัพท์ก็ไม่ได้รับการตอบสนองเช่นกัน เมื่อการตอบสนองต่อ $\overline{INT}$ เกิดขึ้น ซีพียูจะสร้างสถานะพิเศษใน $\overline{M}_1$ ต่อมา ในสถานะพิเศษของ $\overline{M}_1$ นี้ $\overline{IORQ}$ จะแอคตีฟแทน $\overline{MREQ}$ เพื่อเป็นการตอบสนองต่อการอินเตอร์รัพท์และตอนท้ายของ $\overline{IORQ}$ ซีพียูจะทำการอ่านข้อมูล 8 บิทเป็นเวคเตอร์สำหรับการอินเตอร์รัพท์จากบัสข้อมูล สังเกตว่าในกรณีนี้ซีพียูจะสร้างสัญญาณ Tw สองสเตทขึ้นอย่างอัตโนมัติ การสร้าง Tw ขึ้นมาก็เพื่อชลอเวลาให้อุปกรณ์ภายนอกตรวจสอบตัวอุปกรณ์ที่สร้างอินเตอร์รัพท์และจัดลำดับความสำคัญ (priority) ของสัญญาณที่อินเตอร์รัพท์



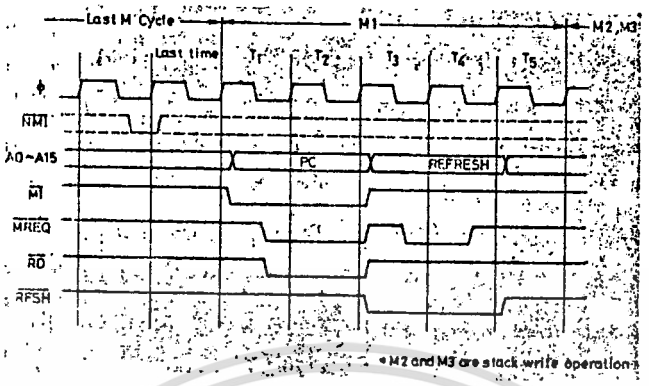
เรายังสามารถเพิ่มสเตทการ $\overline{WAIT}$ ขึ้นอีกได้เพื่อให้ซีพียูรออุปกรณ์ที่จะส่งเวคเตอร์ของการอินเตอร์รัพท์ทางบัสข้อมูล



รูปที่ 15
ไดอะแกรมการรับสัญญาณอินเตอร์รัพท์และการตอบสนองต่ออินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 การตอบสนองต่อการอินเตอร์รัพท์แบบอนมาสเคเบิ้ลอินเตอร์รัพท์ (Nonmaskable interrupt response)



รูปที่ 16 ไดอะแกรมการตอบสนองต่อการอินเตอร์รัพท์แบบอนมาสเคเบิ้ล

รูปที่ 16 เป็นไดอะแกรมเวลาของการตอบสนองต่อการอินเตอร์-รัพท์แบบอนมาสเคเบิ้ล เมื่อถึงพัลซ์ของสัญญาณนาฬิกาลูกสุดท้ายก่อนการกระทำการจบคำสั่งแต่ละคำสั่ง ซีพียูจะทำการตรวจสอบขา $\overline{NMI}$ และถ้า $\overline{NMI}$ มีค่าแอคตีฟ (= 0) ซีพียูก็จะแลทช์ไว้แล้วตอบสนองต่อการอินเตอร์รัพท์ทันที การอินเตอร์รัพท์ด้วยวิธีนี้ถือว่าซีพียูให้ความสำคัญสูงสุด และขบวนการทางซอฟท์แวร์ก็ไม่สามารถที่จะป้องกันไม่ให้ซีพียูตอบสนองได้ การตอบสนองจะเกิดขึ้นทันที ดังนั้นการอินเตอร์รัพท์ด้วยวิธีนี้จึงนำมาใช้ในกรณีที่สำคัญเป็นพิเศษ เช่น ใช้อินเตอร์รัพท์เมื่อมีแรงดันไฟตก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

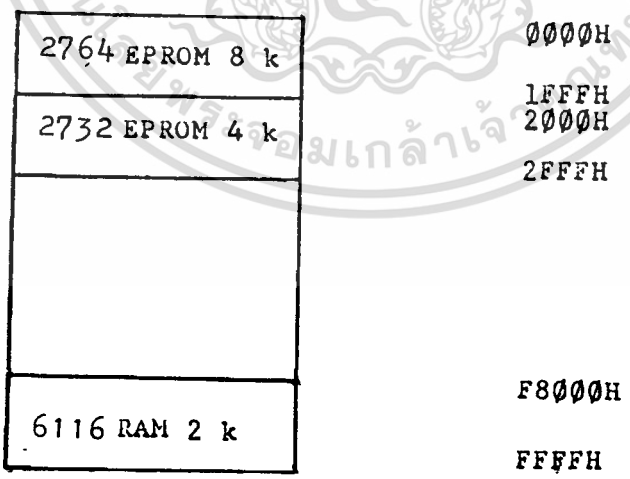
3.1 Memory Map.

การออกแบบเพื่อที่จะสร้าง ระบบ Microcomputer ขึ้นมาสักระบบหนึ่งไม่ว่าจะใช้ Processor เบอร์ใดก็ตาม สิ่งที่ต้องคำนึงถึงประการแรกคือ ตาราง Memory Map. ซึ่งประกอบด้วย

- ส่วนของโปรแกรม
- ส่วนของ stack

Memory Map. สำหรับใช้ควบคุมของระบบงานนี้เป็นของเบอร์ S-80 เพราะใช้กับระบบที่ไม่ใหญ่จนเกินไป สามารถอ้างแอดเดรสได้ง่ายและก็เนื่องมีหน่วยความจำที่อ้างได้ถึง 64K (A_0-A_{15})

ตาราง Memory Map.



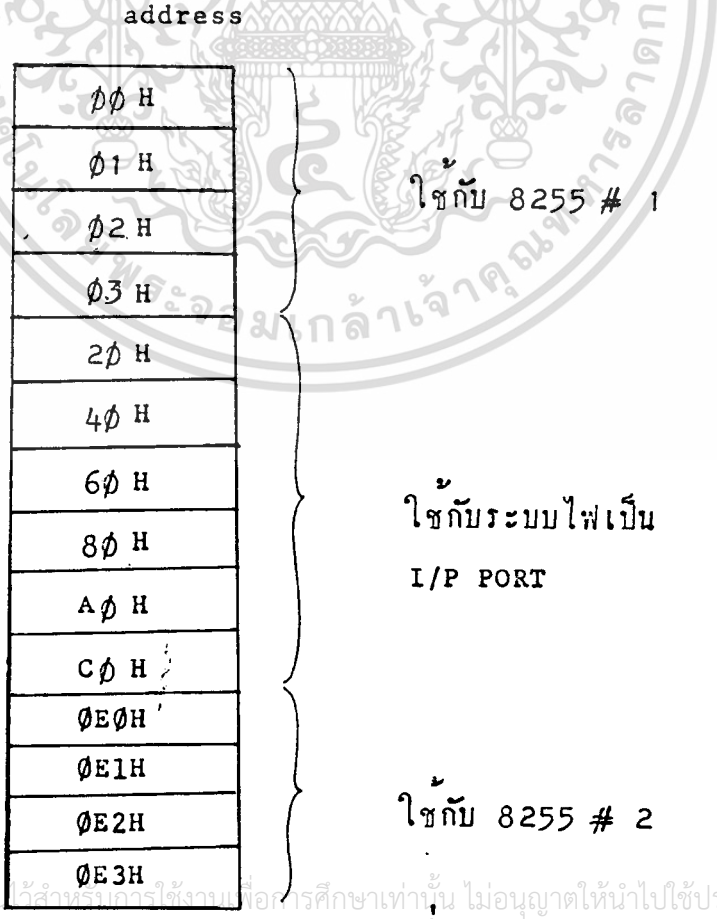
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A_{15}	A_{14}	A_{13}	A_{12}	$A_{11} - A_0$	Address	
Ø	Ø	Ø	Ø	Ø - Ø	ØØØØH	2764 (8k)
Ø	Ø	Ø	1	1 - 1	1FFFH	
Ø	Ø	1	Ø	Ø - Ø	2ØØØH	2732 (4k)
Ø	Ø	1	Ø	1 - 1	2FFFH	

รูปที่ 1

3.2 INPUT/OUTPUT PORT

เมื่อเราออกแบบ Memory Map. เรียบร้อยแล้ว ขั้นตอนต่อไปเราจะต้องกำหนดตำแหน่งของ port สำหรับ Z-80 มีตำแหน่ง port ทั้งหมด 256 port



รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

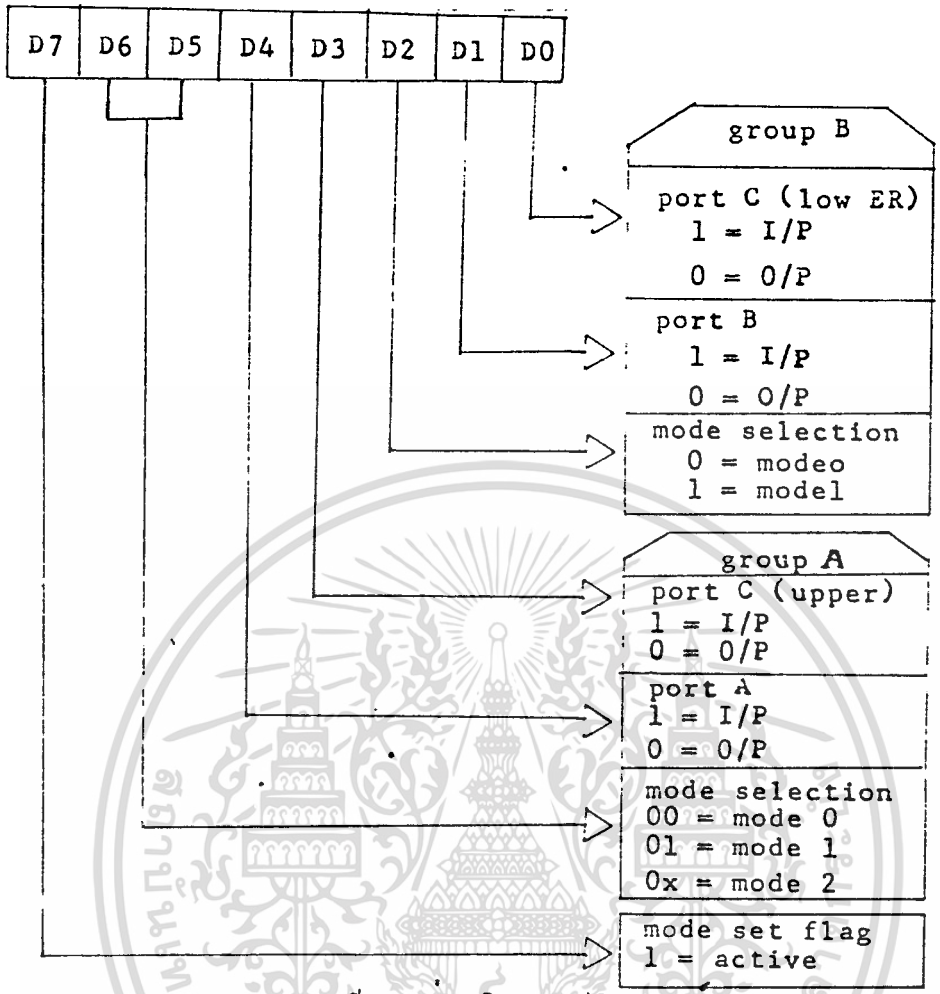
3.3 8255 PIO PORT # 1

A7	A6	A5	A4	A3	A3	A1	A0
Ø	Ø	Ø	x	x	x	Ø	Ø
1	1	1	x	x	x	1	1

สวน decode

DEVICE PIN				REGISTER NAME
RD	WR	A1	A0	
1	Ø	Ø	Ø	write port A data
Ø	1	Ø	Ø	read port A data
1	Ø	Ø	1	write port B data
Ø	1	Ø	1	read port B data
1	Ø	1	Ø	write port C data
Ø	1	1	Ø	read port C data
1	Ø	1	1	write control data
Ø	1	1	1	illegal read register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป แสดงรายละเอียดแต่ละบิทของรีจิสเตอร์ควบคุมของ 8255

รูปที่ 3

เรากำหนดให้ port A เป็น I/P port, port B เป็น O/P port

port C upper เป็น O/P, port C lower เป็น I/P

เราสามารถ set control data ได้ดังนี้

D7	D6	D5	D4	D3	D2	D1	D0	
1	Ø	Ø	1	Ø	Ø	Ø	1	⟹ 91 H

ตำแหน่ง control port จะเป็นดังนี้

A7	A6	A5	A4	A3	A2	A1	A0	
Ø	Ø	Ø	Ø	Ø	Ø	1	1	⟹ Ø3 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

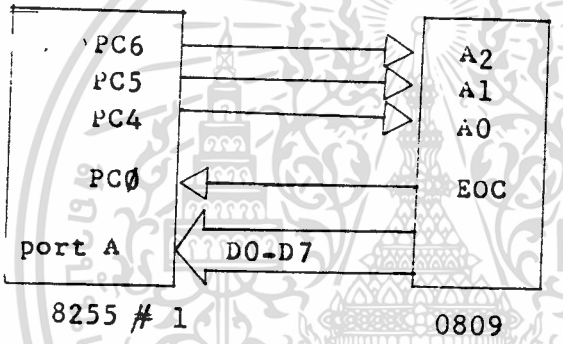
3.4 ส่วนติดต่อกับ sensor

เราใช้ port A เป็น I/P port สำหรับรับ DATA จาก 0809 ตำแหน่ง decode คือ ØØH

port C upper เป็น O/P port ซึ่งเป็นตัวส่งสัญญาณไป select channal ของ 0809 A/D

port C lower เป็น I/P port รับสัญญาณจาก EOC เป็นตัว check ได้ว่า0809 ได้ convert เสร็จแล้ว ตำแหน่ง decode คือ Ø2H

เราสามารถต่อ กับ 0809 อย่างง่าย ๆ ได้ดังรูป



รูปที่ 4

ขา A0, A1, A2 ต้องการข้อมูลที่ถูกlatch. O/P ของ 8255 ก็มีสภาวะ latch เหมือนกัน จึงต่อเข้าด้วยกันได้

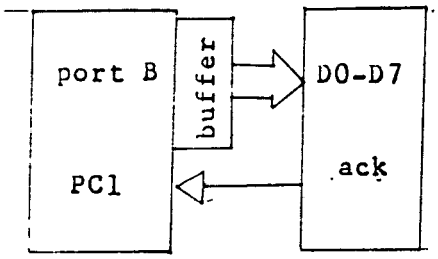
3.5 การติดต่อกับส่วน speech syncthesizer

เราใช้ port B เป็น O/P port

ตำแหน่ง decode คือ Ø1 H

ใช้วิธีการติดต่อแบบ hand shake ทาง sofeware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8255 # 1 8255 of speech

รูปที่ 5

ใช้ D7 เป็น strobe port C lower (PC1) เป็นขา ack nowlege

8255 PIO port # 2

เราจะใช้ 8255 ตัวที่ 2 สำหรับติดต่อกับส่วน display

3.6 การติดต่อกับส่วน CRT display

ใช้วิธีการติดต่อแบบ hand shake ทาง software เช่นเดียวกับการติดต่อกับส่วน speech เพื่อลดความยุ่งยากทาง hard ware ในการติดต่อโดย set ให้ 8255 เป็น mode 1 Hand check

การ hand shake ทาง soft ware กระทำได้ง่าย ๆ ดังนี้

set 8255 ให้ทำงานใน mode Ø เป็น I/P, O/P port ตามปกติ ให้ port B เป็น O/P, port Cupper เป็น I/P, port C lower เป็น O/P ∴ Control Data คือ 88 H

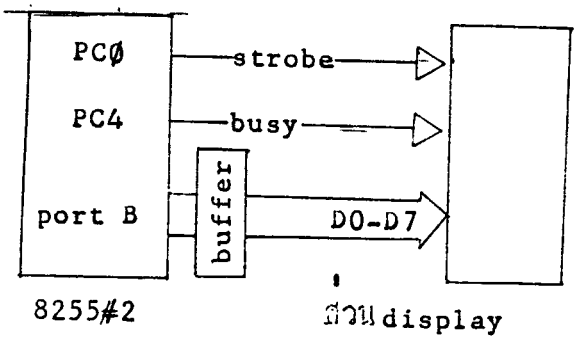
Control port คือ ØE3H

port B ตำแหน่ง decode คือ ØE1H

port C ตำแหน่ง decode คือ ØE2H

การติดต่อจะเป็นลักษณะดังรูปข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6

3.7 การสร้างวงจร

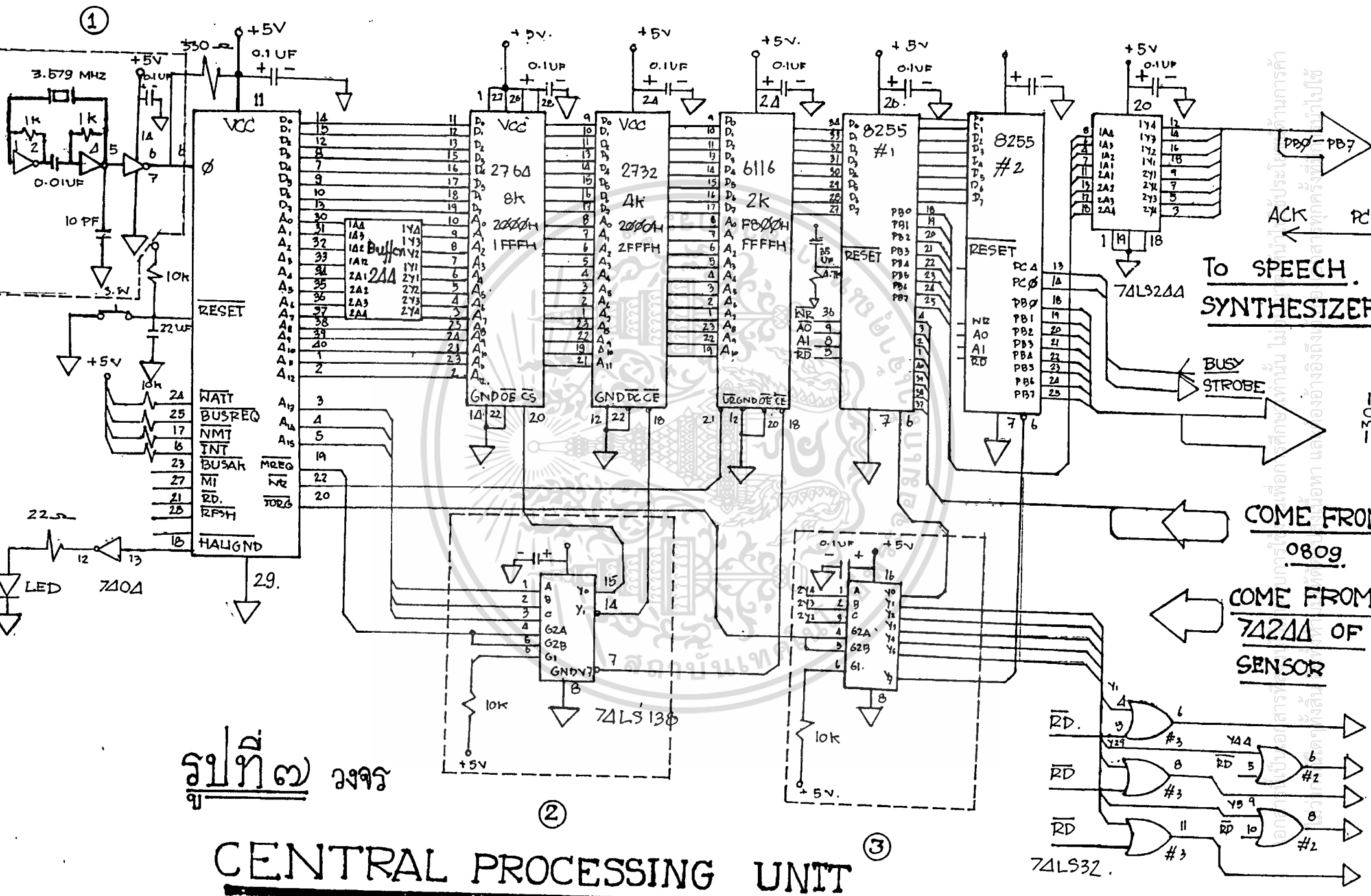
วงจร clock pluse สำหรับป้อนให้กับ Z-80 ตามรูปที่ 7 ส่วนที่ 1 (เส้นประ) ใช้ clystal ความถี่ 3.579 MHz เป็นตัวกำเนิดความถี่ โดยต่อร่วมกับวงจร OSC ผ่านการจัดรูปสัญญาณ โดย Inverter และ R 330 ป้อนเข้าขา 6 ความถี่ยังคงเท่าเดิม คือ 3.579 MHz ซึ่ง CPU เบอร์ Z-80 สามารถทำงานได้

ต่อวงจร reset มาที่ไม่ได้ใช้งาน ขาที่เป็น I/P จะต้องถูก pull up ส่วนขาที่เป็น O/P ปล่อยลอยไว้ได้ ต่อ port,memory เข้ากับ CPU ขา A0-A7 จะต่อผ่านวงจร Buffer address เพราะ A0-A2 จะถูก drive ไปยังส่วนภายนอกผ่านสายแพ และยังขับส่วนต่าง ๆ มากมาย ภายในวงจรนี้อีกด้วย

จากระบบที่ไม่ซับซ้อนเราสามารถสร้างวงจร Decode memory ตามที่ออกแบบไว้ได้อย่างง่าย ๆ โดยใช้ IC เบอร์ 74LS138 ดังรูป (เส้นประส่วนที่ 2)

สำหรับส่วน Decode port(เส้นประส่วนที่ 3) ก็เช่นเดียวกัน แต่มีเบอร์ port จำนวนหนึ่ง เมื่อถูก decode แล้วจะถูกไปต่อร่วมกับขา RD โดยใช้ OR gate IC เบอร์ 74LS32 เพื่อส่งไป Enable 74LS244 เพื่ออ่านข้อมูลที่เป็นระบบไฟเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


①
+5V
330
0.1 UF
3.579 MHz
1K
1K
0.01UF
10 PF
10K
3.3V
22UF
RESET
WAIT
BUSREQ
NMI
INT
BUSAK
M1
RD.
RFSH
HALIGND
MREQ
IORQ
22Ω
LED
74O4
29.
VCC
Ø
2764
8K
2000H
1FFFH
2732
4K
2000H
2FFFH
6116
2K
F800H
FFFFH
8255 #1
8255 #2
RESET
WR
AO
A1
RD
74LS244
PB0 – PB7
ACK
To SPEECH SYNTHESIZER
BUSY
STROBE
COME FROM 0809
COME FROM 74244 OF SENSOR
74LS138
10K
RD.
74LS32
②
③
รูปที่ ๓ วงจร
CENTRAL PROCESSING UNIT

3.8 การพัฒนาโปรแกรม

- การพัฒนาโปรแกรมบน single bord จะเป็นไปได้ช้ามาก เนื่องจากโปรแกรมจะอาศัยการเปิด table เป็นหลัก ซึ่งจะต้องใช้ data เป็นจำนวนมาก โปรแกรมมีขนาดใหญ่ การแก้ไข ดัดแปลง ทำได้ลำบากไม่สะดวก เสียเวลามาก ดังนั้นเราจึงหาวิธีที่จะพัฒนาโปรแกรมดังนี้

- ใช้ Editor ของ side kick เขียนโปรแกรม
- ใช้ Z-80 cross Assembler เป็นตัว translate
- ใช้ symdeb move โปรแกรมจากตำแหน่ง ที่มี CS เป็น segment offset 100 ถึง 1FFFH ไปยังตำแหน่งที่การ์ด down load เสียบต่ออยู่ คือ AC00 : 0
- RAM ที่อยู่บนการ์ด down load จะมีคำสั่งที่เป็น machine code บรรจุอยู่
- Reset CPU บนบอร์ดของเราให้ Fetch คำสั่งมาเอ็กซ์คิวไปจนจบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

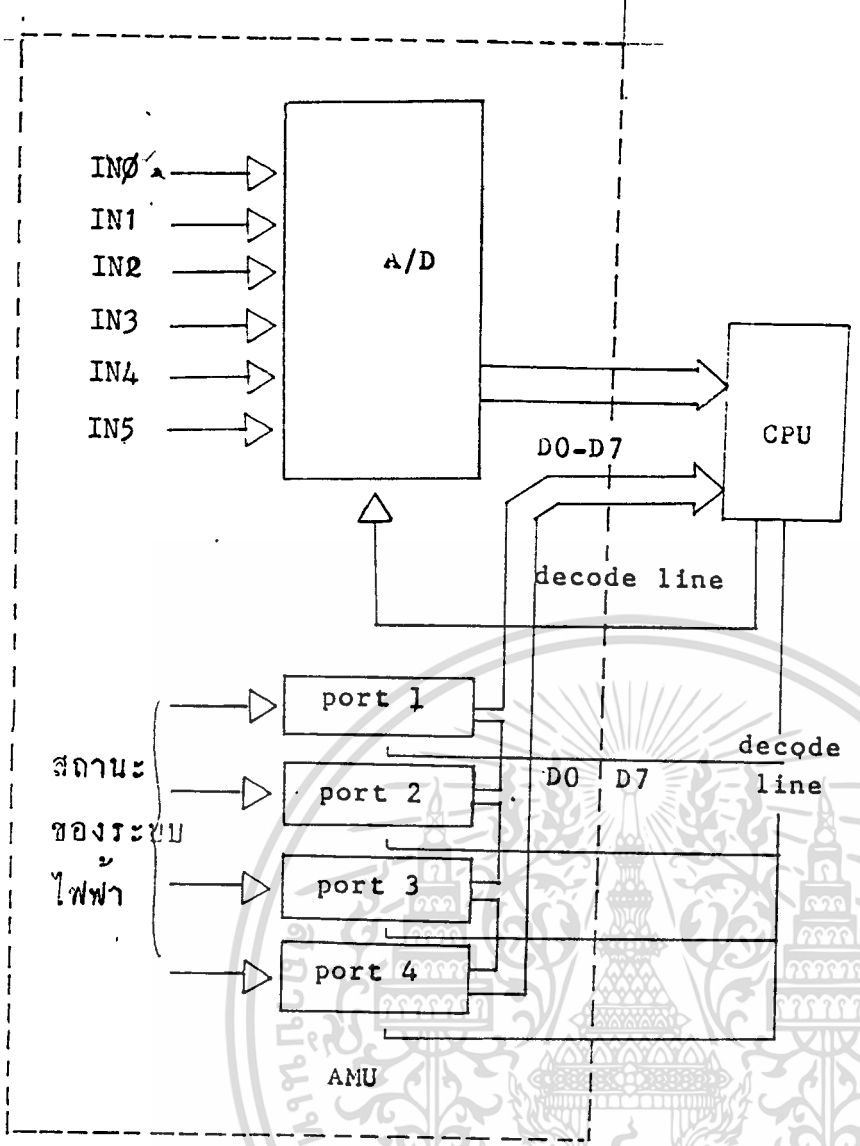
เนื่องจากโครงงาน Automotive computerized measuring system นี้ ประกอบด้วยโครงงานย่อย 4 ส่วน ดังนี้

1. AUTOMOBILE MEASURING SYSTEM หรือ AMS
2. CENTRAL PROCESSING UNIT หรือ CPU
3. GRAPHIC DISPLAY SYSTEM หรือ GDS
4. SPEECH SYNTHESIZER UNIT หรือ SSU

ซึ่งโครงงาน AMS, GDS, SSU จะต้องนำมา Interface กับโครงงาน CPU

4.1 การ Interface โครงงาน AMS เข้ากับโครงงาน CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 การนำโครงงาน AMU มาต่อ Interface กับโครงงาน CPU

จากรูป 1 โครงงาน CPU จะอ่านค่าสัญญาณต่าง ๆ ไม่ว่าจะเป็นความเร็วรถแล่น, ความเร็วรอบของเครื่องยนต์, ระดับน้ำมัน, อุณหภูมิของเครื่องยนต์, แรงดันและ กระแสไฟฟ้าของ battery ที่ผ่าน A/D มาแล้ว และสัญญาณที่ตรวจจับมาจากระบบไฟต่าง ๆ ไม่ว่าไฟเลี้ยวซ้าย, ไฟเลี้ยวขวา, ไฟเบรค, แตร, ฯลฯ เข้ามาเพื่อรอประมวลผลทีละสัญญาณ โดยแต่ละสัญญาณจะกำหนดไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IN0 = อุณหภูมิของเครื่องยนต์

IN1 = แรงดันไฟฟ้าของแบตเตอรี่

IN2 = ระดับน้ำมัน

IN3 = ความเร็วรอบ

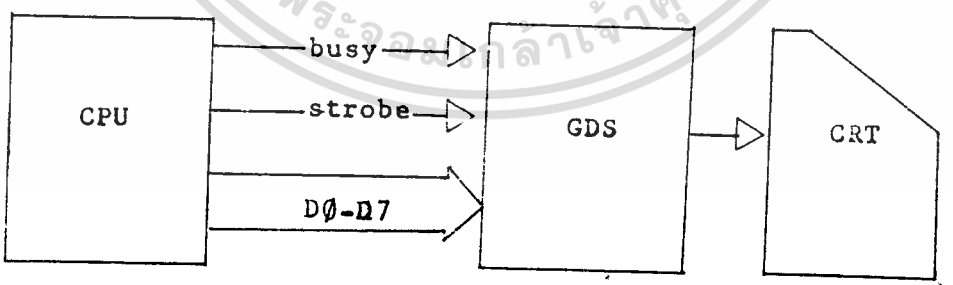
IN4 = ความเร็วรถแล่น

IN5 = กระแสลบ จากแบตเตอรี่

IN6 = กระแสบวก

ส่วนค่าที่อ่านมาได้จากบิทต่าง ๆ (D0-D7) ของ port 1-port 4 จะเป็นค่าสภาวะของระบบไฟฟ้าของรถยนต์ โดยเมื่อสวิทช์ on จะ test bit ได้เป็นลอจิก "0" และเมื่อสวิทช์ off จะ test bit ได้เป็นลอจิก "1" จากนั้นจะไป test bit ตรงหลอดไฟว่ามีลอจิกเป็น "0" หรือ "1" ถ้าสวิทช์ on แล้ว test bit ตรงหลอดไฟ ถ้าเป็น "0" ก็หมายถึง หลอดยังดีอยู่ แต่ถ้า test bit ได้เป็นลอจิก "1" ก็หมายถึง หลอดขาดมีบางบิทที่ใช้ test bit สำหรับตรวจสภาวะว่าเป็นกระแสไฟฟ้าบวกหรือลบ โดยถ้า test bit ได้เป็นลอจิก "0" ก็จะเป็นกระแสไฟฟ้าบวก แต่ถ้าเป็นลอจิก "1" ก็จะเป็นกระแสไฟฟ้าลบ

4.2 การ Interface โครงงาน GDS เข้ากับโครงงาน CPU

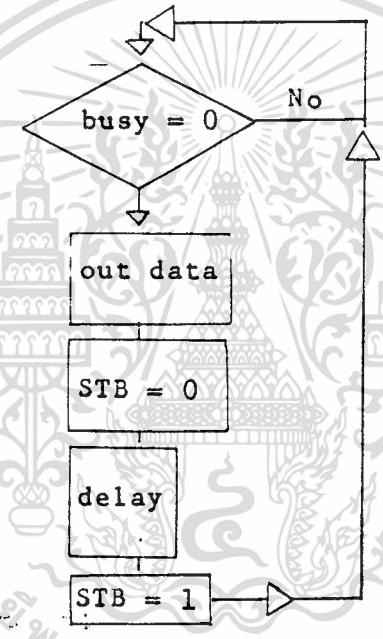


รูป 2 แสดงการนำโครงงาน GDS มาต่อ Interface เข้ากับโครงงาน CPU

จากรูป 2 เมื่อโครงงาน CPU ต้องการที่จะส่งข้อมูลออกไป display ที่ CRT โครงงาน CPU ก็จะตรวจสอบสัญญาณ busy จากโครงงาน GDS ที่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเสียก่อน ถ้าสัญญาณ busy เป็น "1" ก็จะแสดงว่า โครงงาน GDS ยังไม่ได้รับข้อมูลเก่าเข้าไป แต่ถ้าสัญญาณ busy เป็น "0" แสดงว่าโครงงาน GDS รับข้อมูลเก่าเข้าไปแล้ว และก็พร้อมที่จะรับข้อมูลใหม่เข้ามา จากนั้น โครงงาน CPU ก็จะส่งข้อมูลที่จะ display ออกไป เมื่อส่งข้อมูลออกไปแล้ว หลังจากนั้นก็จะส่งสัญญาณ strobe ออกไป โดยจะส่งลอจิก "0" ออกไปทางขา strobe แล้ว delay ไว้ช่วงหนึ่ง จากนั้นก็จะส่งลอจิก "1" ออกไป แล้วก็พร้อมที่ส่งข้อมูลออกไปใหม่ ดังรูป 3 แสดงโฟล์วชาร์ตของการทำงานดังกล่าว



รูป 3 โฟล์วชาร์ตแสดงขบวนการส่งข้อมูลออกไปสู่ GDS แบบ Handshake

ในการแสดงผลนั้น โครงงาน GDS ได้จัด control code ขึ้นมาเพื่อช่วยให้การจัดรูปแบบการแสดงผลนั้นง่ายเข้า โดย control code ที่ได้จัดขึ้นมามีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00H เป็น code ที่ใช้ clear screen

01H เป็น code ที่ใช้ เลือก screen memory page 1

02H เป็น code ที่ใช้ เลือก screen memory page 2

03H เป็น code ที่ใช้ เลือก screen memory page 3

04H เป็น code ที่ใช้ เลือก screen memory page 4

05H เป็น code ที่ใช้ แสดงผลแบบอินเวอร์ส

06H เป็น code ที่ใช้ แสดงผลแบบธรรมดา

07H เป็น code ที่ใช้ แสดงผล แบบ 80 คอลัมน์

08H เป็น code ที่ใช้ แสดงผล แบบ 40 คอลัมน์

09H เป็น code ที่ใช้ในการเลื่อน cursor แบบ gotoxy

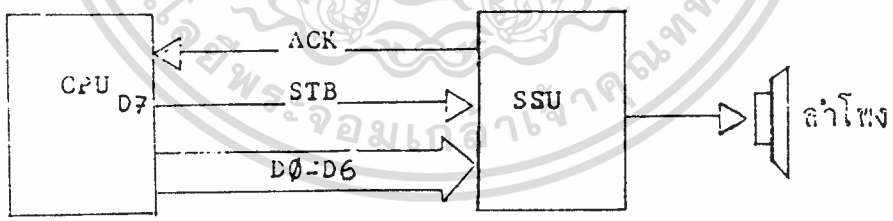
0AH เป็น code ที่ใช้ในการ Line feed

0DH เป็น code ที่ใช้ในการ Carrier return

0EH เป็น code ที่ใช้ในการ on cursor

0FH เป็น code ที่ใช้ในการ off cursor

4.3 การ Interface โครงงาน speech synthesizer unit หรือ SSU เข้ากับโครงงาน CPU

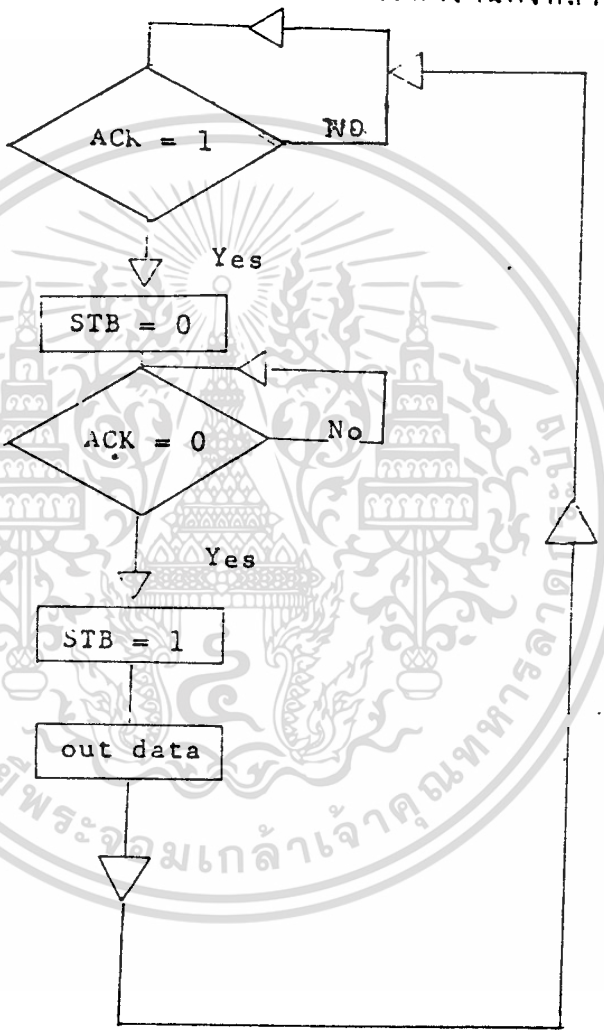


รูป 4 แสดงการ Interface โครงงาน SSU เข้ากับโครงงาน CPU

เมื่อโครงงาน CPU ต้องการจะสั่งให้ SSU ส่งเสียงพูดคำใดก็จะทำได้ โดยส่ง control code ออกมาสู่ SSU โดยการส่งข้อมูลจากโครงงาน CPU ไปสู่โครงงาน SSU จะเป็นแบบ Hand shake

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโครงงาน CPU ต้องการจะส่งข้อมูลออกไป, โครงงาน CPU ก็จะตรวจสอบขา ACK ว่าเป็น "1" หรือไม่ ถ้าเป็น "0" ก็จะวนอ่านดูต่อไป ถ้าเป็น "1" แล้ว ก็จะทำให้ขา STB มีลอจิกเป็น "0" จากนั้นก็จะ check ขา ACK ว่า "1" หรือไม่ ถ้าเป็นก็จะทำให้ STB มีลอจิกเป็น "1" พร้อมกับส่งข้อมูลออกไป ดังรูป 5 จะเป็นโฟล์วชาร์ตของการทำงานดังกล่าว



รูป 5 โฟล์วชาร์ตแสดงการส่งข้อมูลจากโครงงาน CPU ไปสู่โครงงาน SSU แบบ Hand shake

control code สำหรับโครงงาน CPU ส่งไปโครงงาน SSU เพื่อที่สังเคราะห์คำพูด มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ØØH ใช้เป็น code สำหรับคำพูด "ไฟเลี้ยวขวาเปิด"

Ø1H ใช้เป็น code สำหรับคำพูด "ไฟเลี้ยวขวาปิด"

Ø2H ใช้เป็น code สำหรับคำพูด "ไฟเลี้ยวซ้ายเปิด"

Ø3H ใช้เป็น code สำหรับคำพูด "ไฟเลี้ยวซ้ายปิด"

Ø4H ใช้เป็น code สำหรับคำพูด "ไฟสูงเปิด"

Ø5H ใช้เป็น code สำหรับคำพูด "ไฟสูงปิด"

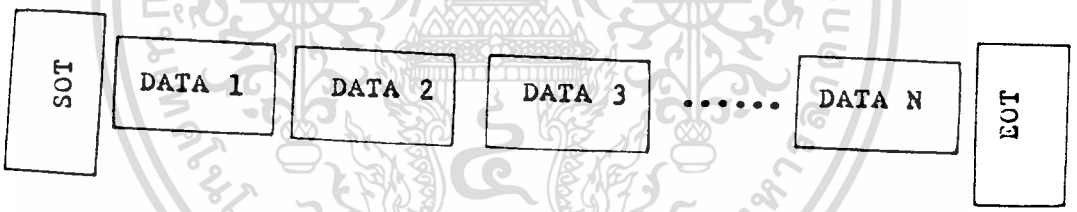
Ø6H ใช้เป็น code สำหรับคำพูด "ไฟหลังเปิด"

Ø7H ใช้เป็น code สำหรับคำพูด "ไฟหลังปิด"

Ø8H ใช้เป็น code สำหรับคำพูด "เตือนอุณหภูมิเกิน"

Ø9H ใช้เป็น code สำหรับคำพูด "น้ำมันหมด"

4.4 โปรโตคอล การส่ง control code จากโครงงาน CPU ไปยังโครงงาน SSU



รูป 6 โปรโตคอลการส่ง control code

โปรโตคอล คือ ข้อตกลงสำหรับการส่ง โดยเมื่อโครงงาน CPU ตรวจสอบสัญญาณ hand shake แล้ว และพร้อมที่จะส่งข้อมูล โครงงาน CPU ก็จะส่ง SOT (start of text) ไปยังโครงงาน SSU เพื่อให้โครงงาน CPU รับรู้และพร้อมที่จะรับข้อมูล จากนั้นโครงงาน CPU ก็จะส่ง DATA ออกไปจนหมดและปิดท้ายด้วย EOT (End of text) เพื่อให้โครงงาน SSU รู้ว่าข้อมูลที่จะส่งหมดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์

กล่าวได้ว่าปริญญานิพนธ์เรื่อง Central Processing Unit เรื่องนี้ได้รับผลเป็นที่น่าพอใจ ถึงแม้ว่ารูปแบบการแสดงผลบนจอ CRT อาจจะดูไม่สวยงามเหมือนรถยนต์คอมพิวเตอร์ที่มีใช้งานกันจริง ๆ อนึ่ง การทำปริญญานิพนธ์เรื่องนี้มีปัญหามากตรงที่เป็นโปรเจคร่วมของหลายโครงงาน โดยเฉพาะผู้ทำปริญญานิพนธ์เรื่องนี้รับผิดชอบในส่วนของ monitor ให้กับโครงงานอื่น ๆ เมื่อโครงงานอื่นเปลี่ยนแปลงขอบเขตของโครงงานไป หรือแม้กระทั่งเปลี่ยนแปลงข้อตกลงต่าง ๆ ก็จะทำให้ผู้จัดทำปริญญานิพนธ์เรื่องนี้ ต้องปรับปรุงให้โครงงานนี้สามารถรองรับการเชื่อมต่อ (Interface) จากโครงงานอื่น กล่าวคือ โครงงานนี้ก็คือ หัวใจของระบบนี้นั่นเอง

5.2 สรุป

ถึงแม้ว่าโครงงานแต่ละส่วนนั้นแทบจะมี CPU สำหรับประมวลผลควบคุมวงจรต่าง ๆ ของโครงงานเองแทบทั้งนั้น ทำให้ดูเหมือนว่าสิ้นเปลืองโดยใช้เหตุแต่ในส่วนดีของการทำโครงงานรูปแบบนี้ก็มีอยู่มาก ไม่ว่าจะเป็นความร่วมมือการทำงานในลักษณะแบบทีม

ผู้จัดทำคิดว่าโครงงานนี้น่าจะได้รับการพัฒนาต่อ ถึงแม้ว่าเมื่อเปรียบเทียบกับรถยนต์คอมพิวเตอร์ที่ใช้กันอยู่ในปัจจุบัน จะมีสมรรถนะต่ำกว่านั้นมาก แต่ในแง่ของการศึกษาและพัฒนาทางด้านเทคโนโลยี ก็คงจะเป็นจุดเริ่มต้นที่ดีมิใช่เหรอ ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80-CPU
Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability

*Note For Z80-CPU as AL and DL characteristics remain the same for the military grade parts except I_{CC}

I_{CC} = 200 mA

Z80-CPU D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} -.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250μA
I_{CC}	Power Supply Current			150	mA	
I_{LI}	Input Leakage Current			±10	μA	V_{IN}=0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT}=0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

Capacitance

T_A = 25°C, f = 1 MHz,
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80-CPU
Ordering Information

C – Ceramic
P – Plastic
S – Standard 5V ±5% 0° to 70°C
E – Extended 5V ±5% -40° to 85°C
M – Military 5V ±10% -55° to 125°C

Z80A-CPU D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

Symbol	Parameter	Min	Typ	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} -.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250μA
I_{CC}	Power Supply Current		90	200	mA	
I_{LI}	Input Leakage Current			10	μA	V_{IN}=0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT}=0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

Capacitance

T_A = 25°C, f = 1 MHz,
unmeasured pins returned to ground

Symbol	Parameter	Max	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80A-CPU
Ordering Information

C Ceramic
P Plastic
S Standard 5V ±5% 0° to 70°C

We reprint data sheets on pages 7-D2 through 7-D13 by permission of Zilog, Incorporated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Z80-CPU
A.C. Characteristics

T_A = 0°C to 70°C, Vcc = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition	
Φ	t_c	Clock Period	4	11.1	μsec		[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	180	[E]	nsec		
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	180	2000	nsec		
	t_r, t_f	Clock Rise and Fall Time		10	nsec		
A_{0-15}	$t_{D(A)}$	Address Output Delay		145	nsec	C_L = 50pF	
	$t_{f(A)}$	Delay to Float		110	nsec		
	t_{aM}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[1]		nsec		[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$
	t_{aI}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec		[2] $t_{aci} = t_c - 80$
	t_{ca}	Address Stable from $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ or $\overline{MREQ}$	[3]		nsec		[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$
	t_{cf}	Address Stable from $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec		[4] $t_{caf} = t_{w(\Phi L)} + t_r - 60$
D_{0-7}	$t_{D(D)}$	Data Output Delay		230	nsec	C_L = 50pF	[5] $t_{dcm} = t_c - 210$
	$t_{f(D)}$	Delay to Float During Write Cycle		90	nsec		[6] $t_{dci} = t_{w(\Phi L)} + t_r - 210$
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		nsec		[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 80$
	$t_{S\Phi(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec		
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec		
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec		
	t_{cdf}	Data Stable From $\overline{WR}$	[7]		nsec		
	t_H	Any Hold Time for Setup Time	0		nsec		
$\overline{MREQ}$	$t_{DL(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	C_L = 50pF	
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High		100	nsec		
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High		100	nsec		
	$t_{w(\overline{MR})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec		[8] $t_{w(MRL)} = t_c - 40$
	$t_{w(\overline{MR})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec		[9] $t_{w(MRH)} = t_{w(\Phi H)} + t_f - 40$
$\overline{IORQ}$	$t_{D\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low		90	nsec	C_L = 50pF	
	$t_{DL\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low		110	nsec		
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High		100	nsec		
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		110	nsec		
$\overline{RD}$	$t_{DL\Phi(RD)}$	[illegible]		[illegible]	[illegible]	[illegible]	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low		[illegible]	[illegible]	[illegible]	
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High		100	nsec	C_L = 50pF	
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		110	nsec		
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low		80	nsec	C_L = 50pF	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low		90	nsec		
	$t_{DH\overline{\Phi}(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ High		100	nsec		
	$t_{w(\overline{WR})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec		[10] $t_{w(WRL)} = t_c - 40$
$\overline{M1}$	$t_{DL(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ Low		130	nsec	C_L = 50pF	
	$t_{DH(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ High		130	nsec		
$\overline{RFSH}$	$t_{DL(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ Low		180	nsec	C_L = 50pF	
	$t_{DH(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ High		150	nsec		
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec		
$\overline{HALT}$	$t_{D(HT)}$	$\overline{HALT}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF	
$\overline{INT}$	$t_{s(IT)}$	$\overline{INT}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{NMI}$	$t_{w(\overline{NML})}$	Pulse Width, $\overline{NMI}$ Low	80		nsec		
$\overline{BUSRQ}$	$t_{s(BQ)}$	$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{BUSAK}$	$t_{DL(BA)}$	$\overline{BUSAK}$ Delay From Rising Edge of Clock, $\overline{BUSAK}$ Low		120	nsec	C_L = 50pF	
	$t_{DH(BA)}$	$\overline{BUSAK}$ Delay From Falling Edge of Clock, $\overline{BUSAK}$ High		110	nsec		
$\overline{RESET}$	$t_{s(RS)}$	$\overline{RESET}$ Setup Time to Rising Edge of Clock	90		nsec		
	$t_{f(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		100	nsec		
	t_{ml}	$\overline{M1}$ Stable Prior to $\overline{IORQ}$ (Interrupt Ack.)	[11]		nsec		[11] $t_{ml} = 3t_c + t_{w(\Phi H)} + t_f - 80$

NOTES

A. Data should be enabled onto the CPU data bus when $\overline{RD}$ is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and $\overline{IORQ}$ are both active.

B. All control signals are internally synchronized so they may be totally asynchronous with respect to the clock.

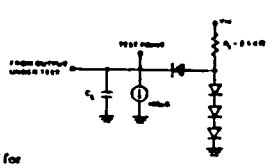
C. The $\overline{RESET}$ signal must be active for a minimum of 3 clock cycles.

D. Output Delay vs. Loaded Capacitance

TA = 70°C Vcc = +5V ±5%

Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines

E. Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum.



Load circuit for Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80-CPU
A.C. Characteristics

T_A = 0°C to 70°C, Vcc = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
Φ	t_c	Clock Period	4	[12]	nsec	
	$t_w(\Phi H)$	Clock Pulse Width, Clock High	180	[E]	nsec	
	$t_w(\Phi L)$	Clock Pulse Width, Clock Low	180	2000	nsec	
	t_r, t_f	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		145	nsec	C_L = 50pF
	$t_{F(AD)}$	Delay to Float		110	nsec	
	t_{acm}	Address Stable Prior to MREQ (Memory Cycle)	[1]		nsec	
	t_{aci}	Address Stable Prior to IORQ, RD or WR (I/O Cycle)	[2]		nsec	
	t_{ca}	Address Stable from RD, WR, IORQ or MREQ	[3]		nsec	
	t_{caf}	Address Stable From RD or WR During Float	[4]		nsec	
D_{0-7}	$t_{D(D)}$	Data Output Delay		230	nsec	C_L = 50pF
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		nsec	
	$t_{S\bar{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec	
	t_{dcm}	Data Stable Prior to WR (Memory Cycle)	[5]		nsec	
	t_{dci}	Data Stable Prior to WR (I/O Cycle)	[6]		nsec	
	t_{cdf}	Data Stable From WR	[7]		nsec	
	t_H	Any Hold Time for Setup Time	0		nsec	
MREQ	$t_{DL\Phi(MR)}$	MREQ Delay From Falling Edge of Clock, MREQ Low		100	nsec	C_L = 50pF
	$t_{DH\Phi(MR)}$	MREQ Delay From Rising Edge of Clock, MREQ High		100	nsec	
	$t_{DH\bar{\Phi}(MR)}$	MREQ Delay From Falling Edge of Clock, MREQ High		100	nsec	
	$t_w(MRL)$	Pulse Width, MREQ Low	[8]		nsec	
	$t_w(MRH)$	Pulse Width, MREQ High	[9]		nsec	
IORQ	$t_{DL\Phi(IR)}$	IORQ Delay From Rising Edge of Clock, IORQ Low		90	nsec	C_L = 50pF
	$t_{DL\bar{\Phi}(IR)}$	IORQ Delay From Falling Edge of Clock, IORQ Low		110	nsec	
	$t_{DH\Phi(IR)}$	IORQ Delay From Rising Edge of Clock, IORQ High		100	nsec	
	$t_{DH\bar{\Phi}(IR)}$	IORQ Delay From Falling Edge of Clock, IORQ High		110	nsec	
RD	$t_{DL\Phi(RD)}$	RD Delay From Rising Edge of Clock, RD Low		100	nsec	C_L = 50pF
	$t_{DL\bar{\Phi}(RD)}$	RD Delay From Falling Edge of Clock, RD Low		130	nsec	
	$t_{DH\Phi(RD)}$	RD Delay From Rising Edge of Clock, RD High		100	nsec	
	$t_{DH\bar{\Phi}(RD)}$	RD Delay From Falling Edge of Clock, RD High		110	nsec	
WR	$t_{DL\Phi(WR)}$	WR Delay From Rising Edge of Clock, WR Low		80	nsec	C_L = 50pF
	$t_{DL\bar{\Phi}(WR)}$	WR Delay From Falling Edge of Clock, WR Low		90	nsec	
	$t_{DH\bar{\Phi}(WR)}$	WR Delay From Falling Edge of Clock, WR High		100	nsec	
	$t_w(WRL)$	Pulse Width, WR Low	[10]		nsec	
M1	$t_{DL(M1)}$	M1 Delay From Rising Edge of Clock, M1 Low		130	nsec	C_L = 50pF
	$t_{DH(M1)}$	M1 Delay From Rising Edge of Clock, M1 High		130	nsec	
RFSH	$t_{DL(RF)}$	RFSH Delay From Rising Edge of Clock, RFSH Low		180	nsec	C_L = 50pF
	$t_{DH(RF)}$	RFSH Delay From Rising Edge of Clock, RFSH High		150	nsec	
WAIT	$t_{s(WT)}$	WAIT Setup Time to Falling Edge of Clock	70		nsec	
HALT	$t_{D(HT)}$	HALT Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF
INT	$t_{s(IT)}$	INT Setup Time to Rising Edge of Clock	80		nsec	
NMI	$t_{w(NML)}$	Pulse Width, NMI Low	80		nsec	
BUSRQ	$t_{s(BQ)}$	BUSRQ Setup Time to Rising Edge of Clock	80		nsec	
BUSAK	$t_{DL(BA)}$	BUSAK Delay From Rising Edge of Clock, BUSAK Low		120	nsec	C_L = 50pF
	$t_{DH(BA)}$	BUSAK Delay From Falling Edge of Clock, BUSAK High		110	nsec	
RESET	$t_{s(RS)}$	RESET Setup Time to Rising Edge of Clock	90		nsec	
	$t_{F(C)}$	Delay to Float (MREQ, IORQ, RD and WR)		100	nsec	
	t_{ms}	M1 Stable Prior to IORQ (Interrupt Ack.)	[11]		nsec	

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$

[2] $t_{aci} = t_c - 80$

[3] $t_{ca} = t_{w(\Phi L)} + t_f - 40$

[4] $t_{caf} = t_{w(\Phi L)} + t_f - 60$

[5] $t_{dcm} = t_c - 210$

[6] $t_{dci} = t_{w(\Phi L)} + t_f - 210$

[7] $t_{cdf} = t_{w(\Phi L)} + t_f - 60$

[8] $t_w(MRL) = t_c - 40$

[9] $t_{w(MRH)} = t_{w(\Phi H)} + t_f - 40$

[10] $t_{w(WRL)} = t_c - 40$

[11] $t_{ms} = 2t_c + t_{w(\Phi H)} + t_f - 80$

NOTES

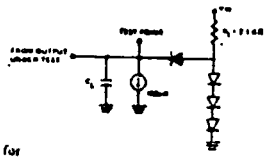
A Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when M1 and IORQ are both active.

B All control signals are internally synchronized so they may be totally asynchronous with respect to the clock.

C The RESET signal must be active for a minimum of 3 clock cycles.

D Output Delays are Guaranteed for: TA = 0°C to 70°C, Vcc = +5V ± 5%, Load Capacitance = 50pF.
Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines.

E Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 nsec maximum.



Load circuit for Output

Z80A-CPU
A.C. Characteristics

T_A = 0°C to 70°C, Vcc = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition	
Φ	t_c	Clock Period	25	[12]	μsec		[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	110	[7]	nsec		
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	110	2000	nsec		
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec		
A_{0-15}	$t_{D(AD)}$	Address Output Delay		110	nsec	C_L = 50pF	
	$t_{F(AD)}$	Delay to Float		90	nsec		
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[1]		nsec		[1] $t_{acm} = t_{w(\Phi H)} + t_f - 65$
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec		[2] $t_{aci} = t_c - 70$
	t_{ca}	Address Stable from $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ or $\overline{MREQ}$	[3]		nsec		[3] $t_{ca} = t_{w(\Phi L)} + t_f - 50$
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec		[4] $t_{caf} = t_{w(\Phi L)} + t_f - 45$
D_{0-7}	$t_{D(D)}$	Data Output Delay		150	nsec	C_L = 50pF	
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec		
	$t_{s\Phi(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	35		nsec		
	$t_{s\overline{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec		
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec		[5] $t_{dcm} = t_c - 170$
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec		[6] $t_{dci} = t_{w(\Phi L)} + t_f - 170$
	t_{cdf}	Data Stable From $\overline{WR}$	[7]		nsec		[7] $t_{cdf} = t_{w(\Phi L)} + t_f - 70$
	t_H	Any Hold Time for Setup Time	0		nsec		
$\overline{MREQ}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low		85	nsec	C_L = 50pF	
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High		85	nsec		
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High		85	nsec		
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec		[8] $t_{w(\overline{MRL})} = t_c - 10$
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec		[9] $t_{w(\overline{MRH})} = t_{w(\Phi H)} + t_f - 20$
$\overline{IORQ}$	$t_{D\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low		75	nsec	C_L = 50pF	
	$t_{DL\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low		85	nsec		
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High		85	nsec		
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		85	nsec		
$\overline{RD}$	$t_{DL\Phi(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low		85	nsec	C_L = 50pF	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low		95	nsec		
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High		85	nsec		
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		85	nsec		
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low		65	nsec	C_L = 50pF	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low		80	nsec		
	$t_{DH\overline{\Phi}(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ High		80	nsec		
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec		[10] $t_{w(\overline{WRL})} = t_c - 30$
$\overline{M1}$	$t_{DL(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ Low		100	nsec	C_L = 50pF	
	$t_{DH(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ High		100	nsec		
$\overline{RFSH}$	$t_{DL(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ Low		120	nsec	C_L = 50pF	
	$t_{DH(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ High		120	nsec		
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec		
$\overline{HALT}$	$t_{D(HT)}$	$\overline{HALT}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF	
$\overline{INT}$	$t_{s(IT)}$	$\overline{INT}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{NMI}$	$t_{w(\overline{NML})}$	Pulse Width, $\overline{NMI}$ Low	80		nsec		
$\overline{BUSRQ}$	$t_{s(BQ)}$	$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	50		nsec		
$\overline{BUSAK}$	$t_{DL(BA)}$	$\overline{BUSAK}$ Delay From Rising Edge of Clock, $\overline{BUSAK}$ Low		100	nsec	C_L = 50pF	
	$t_{DH(BA)}$	$\overline{BUSAK}$ Delay From Falling Edge of Clock, $\overline{BUSAK}$ High		110	nsec		
$\overline{RESET}$	$t_{s(RS)}$	$\overline{RESET}$ Setup Time to Rising Edge of Clock	60		nsec		
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		80	nsec		
	t_{mr}	M1 Stable Prior to IORQ (Interrupt Ack.)	[11]		nsec		[11] $t_{mr} = 2t_c + t_{w(\Phi H)} + t_f - 65$

NOTES

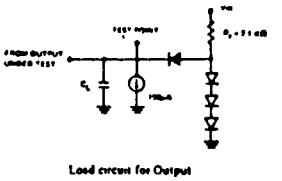
A Data should be enabled onto the CPU data bus when $\overline{RD}$ is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and $\overline{IORQ}$ are both active.

B All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.

C The $\overline{RESET}$ signal must be active for a minimum of 3 clock cycles.

D Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V ±5%
Add 10nsec delay for each 50pf increase in load up to maximum of 200pf for data bus and 100pf for address & control lines.

E Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum.



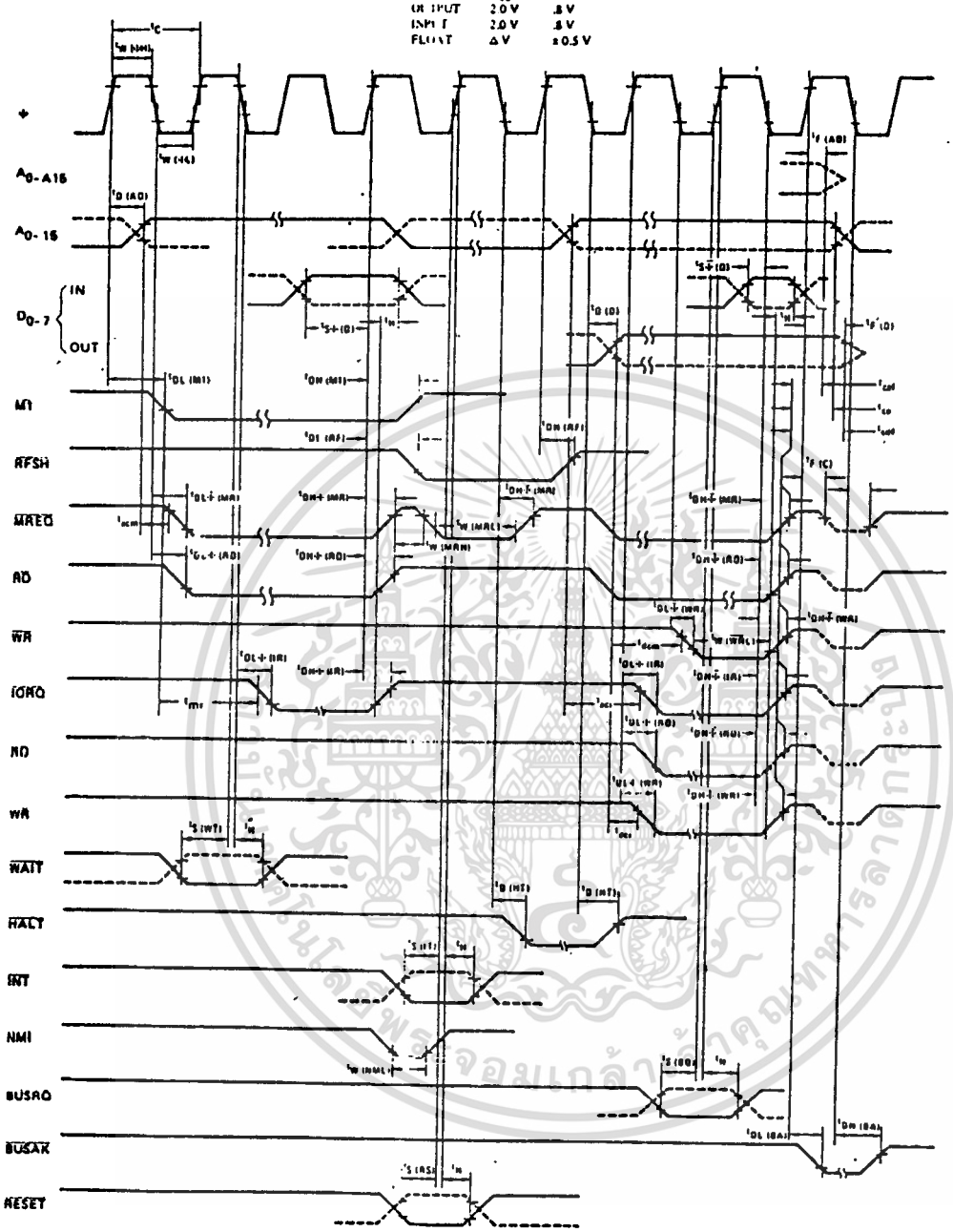
Load circuit for Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	V_{CC} -.6V	.45V
OUTPUT	2.0V	.8V
INPUT	2.0V	.8V
FLOAT	ΔV	±0.5V



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*Address Bus A0-A7 [C]
A8-A15 [B]

Table 7.2 A Summary of the Z80 Instruction Set

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
I/O	IN	A,port	2							[A]←[port] Input to Accumulator from directly addressed I/O port. Address Bus: A0-A7: port A8-A15: [A]
	IN	reg,(C)	2		X	X	P	X	0	[reg]←[[C]] Input to register from I/O port addressed by the contents of C.* If second byte is 70_{16} only the flags will be affected.
	INIR		2		1	?	?	?	1	Repeat until [B]=0: [[HL]]←[[C]] [B]←[B] - 1 [HL]←[HL] + 1 Transfer a block of data from I/O port addressed by contents of C to memory location addressed by contents of HL, going from low addresses to high. Contents of B serve as a count of bytes remaining to be transferred.*
	INCR		2		1	?	?	?	1	Repeat until [B]=0 [[HL]]←[[C]] [B]←[B] - 1 [HL]←[HL] - 1 Transfer a block of data from I/O port addressed by contents of C to memory location addressed by contents of HL, going from high addresses to low. Contents of B serve as a count of bytes remaining to be transferred.*
	IND		2		X	?	?	?	1	[[HL]]←[[C]] [B]←[B] - 1 [HL]←[HL] + 1 Transfer a byte of data from I/O port addressed by contents of C to memory location addressed by contents of HL. Decrement byte count and increment destination address.*
	OUT	port,A	2							[[HL]]←[[C]] [B]←[B] - 1 [HL]←[HL] - 1 Transfer a byte of data from I/O port addressed by contents of C to memory location addressed by contents of HL. Decrement both byte count and destination address.* [port]←[A] Output from Accumulator to directly addressed I/O port. Address Bus: A0-A7: port A8-A15: [A]
	OUT	(C),reg	2							[[C]]←[reg] Output from register to I/O port addressed by the contents of C.*
	OTIR		2		1	?	?	?	1	Repeat until [B]=0: [[C]]←[[HL]] [B]←[B] - 1 [HL]←[HL] + 1 Transfer a block of data from memory location addressed by contents of HL to I/O port addressed by contents of C, going from low memory to high. Contents of B serve as a count of bytes remaining to be transferred.

*Address Bus: A0-A7: [C]
A8-A15: [B]

Table 7-2 A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
I/O (Continued)	OTDR		2		1	?	?	?	1	Repeat until [B]=0: [[C]]←[[HL]] [B]←[B] - 1 [HL]←[HL] - 1 Transfer a block of data from memory location addressed by contents of HL to I/O port addressed by contents of C, going from high memory to low. Contents of B serve as a count of bytes remaining to be transferred.*
	OUTI		2		?	?	?	?	1	[[C]]←[[HL]] [B]←[B] - 1 [HL]←[HL] + 1 Transfer a byte of data from memory location addressed by contents of HL to I/O port addressed by contents of C. Decrement byte count and increment source address.*
	OUTD		2		?	?	?	?	1	[[C]]←[[HL]] [B]←[B] - 1 [HL]←[HL] - 1 Transfer a byte of data from memory location addressed by contents of HL to I/O port addressed by contents of C. Decrement both byte count and source address.*
PRIMARY MEMORY REFERENCE	LD	A,(addr)	3							[A]←[addr] Load Accumulator from directly addressed memory location.
	LD	HL,(addr)	3							[H]←[addr + 1], [L]←[addr] Load HL from directly addressed memory.
	LD	rp,(addr) xy,(addr)	4							[rp(HI)]←[addr + 1], [rp(LO)]←[addr] or [xy(HI)]←[addr + 1], [xy(LO)]←[addr] Load register pair or index register from directly addressed memory.
	LD	(addr),A	3							[addr]←[A] Store Accumulator contents in directly addressed memory location.
	LD	(addr),HL	3							[addr + 1]←[H], [addr]←[L] Store contents of HL to directly addressed memory location.
	LD	(addr),rp (addr),xy	4							[addr + 1]←[rp(HI)], [addr]←[rp(LO)] or [addr + 1]←[xy(HI)], [addr]←[xy(LO)] Store contents of register pair or index register to directly addressed memory.
	LD	A,(BC) A,(DE)	1							[A]←[[BC]] or [A]←[[DE]] Load Accumulator from memory location addressed by the contents of the specified register pair.
	LD	reg,(HL)	1							[reg]←[[HL]] Load register from memory location addressed by contents of HL.
	LD	(BC),A (DE),A	1							[[BC]]←[A] or [[DE]]←[A] Store Accumulator to memory location addressed by the contents of the specified register pair.
	LD	(HL),reg	1							[[HL]]←[reg] Store register contents to memory location addressed by the contents of HL.
	LD	reg,(xy + disp)	3							[reg]←[[xy] + disp] Load register from memory location using base relative addressing.
	LD	(xy + disp),reg	3							[[xy] + disp]←[reg] Store register to memory location addressed relative to contents of index register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7-2. A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
BLOCK TRANSFER AND SEARCH	LDIR		2				0	0	0	Repeat until [BC]=0: [[DE]]←[[HL]] [DE]←[DE] + 1 [HL]←[HL] + 1 [BC]←[BC] - 1 Transfer a block of data from the memory location addressed by the contents of HL to the memory location addressed by the contents of DE, going from low addresses to high. Contents of BC serve as a count of bytes to be transferred.
	LDDR		2				0	0	0	Repeat until [BC]=0: [[DE]]←[[HL]] [DE]←[DE] - 1 [HL]←[HL] - 1 [BC]←[BC] - 1 Transfer a block of data from the memory location addressed by the contents of HL to the memory location addressed by the contents of DE, going from high addresses to low. Contents of BC serve as a count of bytes to be transferred.
	LDI		2				X	0	0	[[DE]] ← [[HL]] [DE]←[DE] + 1 [HL]←[HL] + 1 [BC]←[BC] - 1 Transfer one byte of data from the memory location addressed by the contents of HL to the memory location addressed by the contents of DE. Increment source and destination addresses and decrement byte count.
	LDD		2				X	0	0	[[DE]]←[[HL]] [DE]←[DE] - 1 [HL]←[HL] - 1 [BC]←[BC] - 1 Transfer one byte of data from the memory location addressed by the contents of HL to the memory location addressed by the contents of DE. Decrement source and destination addresses and byte count.
	CPIR		2		X	X	X	X	1	Repeat until [A]=[[HL]] or [BC]=0: [A] - [[HL]] (only flags are affected) [HL]←[HL] + 1 [BC]←[BC] - 1 Compare contents of Accumulator with those of memory block addressed by contents of HL, going from low addresses to high. Stop when a match is found or when the byte count becomes zero.
	CPDR		2		X	X	X	X	1	Repeat until [A]=[[HL]] or [BC]=0: [A] - [[HL]] (only flags are affected) [HL]←[HL] - 1 [BC]←[BC] - 1 Compare contents of Accumulator with those of memory block addressed by contents of HL, going from high addresses to low. Stop when a match is found or when the byte count becomes zero.

Table 7-2. A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
BLOCK TRANSFER AND SEARCH (Continued)	CPI		2		X	X	X	X	1	[A] - [[HL]] (only flags are affected) [HL]—[HL] + 1 [BC]—[BC] - 1 Compare contents of Accumulator with those of memory location addressed by contents of HL. Increment address and decrement byte count.
	CPD		2		X	X	X	X	1	[A] - [[HL]] (only flags are affected) [HL]—[HL] - 1 [BC]—[BC] - 1 Compare contents of Accumulator with those of memory location addressed by contents of HL. Decrement address and byte count.
SECONDARY MEMORY REFERENCE	ADD	(HL)	1	X	X	X	V	X	0	[A]—[A] + [[HL]] or [A]—[A] + [[xy] + disp]
		(xy + disp)	3							Add to Accumulator using implied addressing or base relative addressing.
	ADC	(HL)	1	X	X	X	V	X	0	[A]—[A] + [[HL]] + C or [A]—[A] + [[xy] + disp] + C
		(xy + disp)	3							Add with Carry using implied addressing or base relative addressing.
	SUB	(HL)	1	X	X	X	V	X	1	[A]—[A] - [[HL]] or [A]—[A] - [[xy] + disp]
		(xy + disp)	3							Subtract from Accumulator using implied addressing or base relative addressing.
	SBC	(HL)	1	X	X	X	V	X	1	[A]—[A] - [[HL] - C or [A]—[A] - [[xy] + disp] - C
		(xy + disp)	3							Subtract with Carry using implied addressing or base relative addressing.
	AND	(HL)	1	0	X	X	P	1	0	[A]—[A] ∧ [[HL]] or [A]—[A] ∧ [[xy] + disp]
		(xy + disp)	3							AND with Accumulator using implied addressing or base relative addressing
	OR	(HL)	1	0	X	X	P	1	0	[A]—[A] ∨ [[HL]] or [A]—[A] ∨ [[xy] + disp]
		(xy + disp)	3							OR with Accumulator using implied addressing or base relative addressing.
	XOR	(HL)	1	0	X	X	P	1	0	[A]—[A] ⊻ [[HL]] or [A]—[A] ⊻ [[xy] + disp]
		(xy + disp)	3							Exclusive-OR with Accumulator using implied addressing or base relative addressing.
	CP	(HL)	1	X	X	X	V	X	1	[A] - [[HL]] or [A] - [[xy] + disp]
		(xy + disp)	3							Compare with Accumulator using implied addressing or base relative addressing. Only the flags are effected.
	INC	(HL)	1		X	X	V	X	0	[[HL]]—[[HL]] + 1 or [[xy] + disp]—[[xy] + disp] + 1
		(xy + disp)	3							Increment using implied addressing or base relative addressing.
	DEC	(HL)	1		X	X	V	X	1	[[HL]]—[[HL]] - 1 or [[xy] + disp]—[[xy] + disp] - 1
		(xy + disp)	3							Decrement using implied addressing or base relative addressing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7-2 A Summary of the Z80 Instruction Set (Continued)

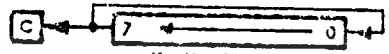
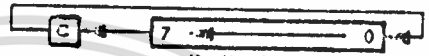
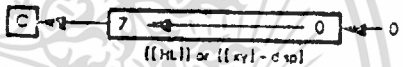
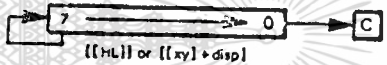
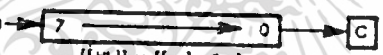
TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS C	Z	S	P/O	A_C	N	OPERATION PERFORMED
MEMORY SHIFT AND ROTATE	RLC	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Rotate contents of memory location (implied or base relative addressing) left with branch Carry.
	RL	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Rotate contents of memory location left through Carry.
	RRC	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Rotate contents of memory location right with branch Carry.
	RR	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Rotate contents of memory location right through Carry.
	SLA	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Shift contents of memory location left and clear LSB (Arithmetic Shift).
	SRA	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Shift contents of memory location right and preserve MSB (Arithmetic Shift).
	SRL	(HL) (xy + disp)	2 4	X	X	X	P	0	0	 [[HL]] or [[xy] + disp] Shift contents of memory location right and clear MSB (Logical Shift).
IMMEDIATE	LD	reg,data	2							[reg]—data Load immediate into register.
	LD	rp,data16 xy,data16	3 4							[rp]—data16 or [xy]—data16 Load 16 bits of immediate data into register pair or index register.
	LD	(HL),data (xy + disp),data	2 4							[[HL]]—data or [[xy] + disp]—data Load immediate into memory location using implied or base relative addressing.

Table 7-2 A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERANDS	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
JUMP	JP	label	3							[PC]—label Jump to instruction at address represented by label
	JR	disp	2							[PC]—[PC] + 2 + disp Jump relative to present contents of Program Counter
	JP	(HL) (xy)	1 2							[PC]—[HL] or [PC]—[xy] Jump to address contained in HL or Index register
SUBROUTINE CALL AND RETURN	CALL	label	3							[[SP]-1]—[PCHI] [[SP]-2]—[PCLO] [SP]—[SP] 2 [PC]—label Jump to subroutine starting at address represented by label
	CALL	cond,label	3							Jump to subroutine if condition is satisfied; otherwise continue in sequence.
	RET		1							[PCLO]]—[[SP]] [PCHI]—[[SP] + 1] [SP]—[SP] + 2 Return from subroutine.
	RET	cond	1							Return from subroutine if condition is satisfied; otherwise continue in sequence
IMMEDIATE OPERATE	ADD	data	2	X	X	X	O	X	0	[A]—[A] + data Add immediate to Accumulator.
	ADC	data	2	X	X	X	O	X	0	[A]—[A] + data + C Add immediate with Carry.
	SUB	data	2	X	X	X	O	X	1	[A]—[A] - data Subtract immediate from Accumulator
	SBC	data	2	X	X	X	O	X	1	[A]—[A] - data - C Subtract immediate with Carry.
	AND	data	2	0	X	X	P	1	0	[A]—[A] ∧ data AND immediate with Accumulator.
	OR	data	2	0	X	X	P	1	0	[A]—[A] ∨ data OR immediate with Accumulator.
	XOR	data	2	0	X	X	P	1	0	[A]—[A] ∨ data Exclusive-OR immediate with Accumulator.
	CP	data	2	X	X	X	V	X	1	[A] - data Compare immediate data with Accumulator contents only the flags are affected

ลิขสิทธิ์เป็นของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
JUMP ON CONDITION	JP	cond,label	3							If cond. then [PC]—label Jump to instruction at address represented by label if the condition is true
	JR	C,disp	2							If C=1 then [PC]—[PC] + 2 + disp Jump relative to contents of Program Counter if Carry flag is set
	JR	NC,disp	2							If C=0 then [PC]—[PC] + 2 + disp Jump relative to contents of Program Counter if Carry flag is reset
	JR	Z,disp	2							If Z=1 then [PC]—[PC] + 2 + disp Jump relative to contents of Program Counter if Zero flag is set
	JR	NZ,disp	2							If Z=0 then [PC]—[PC] + 2 + disp Jump relative to contents of Program Counter if Zero flag is reset
	DJNZ	disp	2							[B]—[B] - 1 If [B] ≠ 0, then [PC]—[PC] + 2 + disp Decrement contents of B and Jump relative to contents of Program Counter if result is not 0.
REGISTER-REGISTER MOVE	LD	dst,src	1							[dst]—[src] Move contents of source register to destination register. Register designations src and dst may each be A, B, C, D, E, H or L.
	LD	A,IV	2		X	X	I	0	0	[A]—[IV] Move contents of Interrupt Vector register to Accumulator
	LD	A,R	2		X	X	I	0	0	[A]—[R] Move contents of Refresh register to Accumulator
	LD	IV,A	2							[IV]—[A] Load Interrupt Vector register from Accumulator
	LD	R,A	2							[R]—[A] Load Refresh register from Accumulator
	LD	SP,HL	1							[SP]—[HL] Move contents of HL to Stack Pointer
	LD	SP,xy	2							[SP]—[xy] Move contents of Index register to Stack Pointer.
	EX	DE,HL	1							[DE]——[HL] Exchange contents of DE and HL.
	EX	AF,AF'	1							[AF]——[AF'] Exchange program status and alternate program status
	EXX		1							$\begin{pmatrix}[BC]\\[DE]\\[HL]\end{pmatrix}$ —— $\begin{pmatrix}[BC']\\[DE']\\[HL']\end{pmatrix}$ Exchange register pairs and alternate register pairs

Table 7-2 A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
REGISTER-REGISTER OPERATE	ADD	reg	1	↕	X	X	O	↕	0	[A]←[A] + [reg] Add contents of register to Accumulator.
	ADC	reg	1	↕	X	X	O	↕	0	[A]←[A] + [reg] + C Add contents of register and Carry to Accumulator.
	SUB	reg	1	↕	X	X	O	↕	1	[A]←[A] - [reg] Subtract contents of register from Accumulator.
	SBC	reg	1	↕	X	X	O	↕	1	[A]←[A] - [reg] - C Subtract contents of register and Carry from Accumulator.
	AND	reg	1	0	X	X	P	1	0	[A]←[A] ∧ [reg] AND contents of register with contents of Accumulator
	OR	reg	1	0	X	X	P	↕	0	[A]←[A] ∨ [reg] OR contents of register with contents of Accumulator
	XOR	reg	1	0	X	X	P	↕	0	[A]←[A] ⊻ [reg] Exclusive-OR contents of register with contents of Accumulator.
	CP	reg	1	↕	X	X	O	↕	1	[A] - [reg] Compare contents of register with contents of Accumulator. Only the flags are affected.
	ADD	HL,rp	1	↕				↕	0	[HL]←[HL] + [rp] 16-bit add register pair contents to contents of HL.
	ADC	HL,rp	2	↕	X	X	O	↕	0	[HL]←[HL] + [rp] + C 16-bit add with Carry register pair contents to contents of HL.
	SBC	HL,rp	2	↕	X	X	O	↕	1	[HL]←[HL] - [rp] - C 16-bit subtract with Carry register pair contents from contents of HL.
	ADD	IX,pp	2	↕				↕	0	[IX]←[IX] + [pp] 16-bit add register pair contents to contents of Index register IX (pp=BC, DE, IX, SP)
	ADD	IY,rr	2	↕				↕	0	[IY]←[IY] + [rr] 16-bit add register pair contents to contents of Index register IY (rr=BC, DE, IY, SP).
REGISTER OPERATE	DAA		1	↕	X	X	P	↕		Decimal adjust Accumulator, assuming that Accumulator contents are the sum or difference of BCD operands.
	CPL		1					1	1	[A]←[$\overline{A}$] Complement Accumulator (ones complement).
	NEG		2	↕	X	X	O	↕	1	[A]←[$\overline{A}$] + 1 Negate Accumulator (twos complement).
	INC	reg	1		X	X	O	↕	0	[reg]←[reg] + 1 Increment register contents.
	INC	rp xy	1							[rp]←[rp] + 1 or [xy]←[xy] + 1 Increment contents of register pair or Index register.
	DEC	reg	2		X	X	O	↕	1	[reg]←[reg] - 1 Decrement register contents.
	DEC	rp xy	1 2							[rp]←[rp] - 1 or [xy]←[xy] - 1 Decrement contents of register pair or Index register.

Table 7-2 A Summary of the Z80 Instruction Set (Continued)

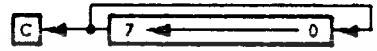
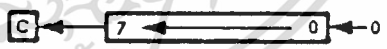
TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	A_C	N	
REGISTER SHIFT AND ROTATE	RLCA		1	X				0	0	 [A] Rotate Accumulator left with branch Carry
	RLA		1	X				0	0	[A] Rotate Accumulator left through Carry.
	RRCA		1	X				0	0	[A] Rotate Accumulator right with branch Carry.
	RRA		1	X				0	0	[A] Rotate Accumulator right through Carry.
	RLC	reg	2	X	X	X	P	0	0	[reg] Rotate contents of register left with branch Carry
	RL	reg	2	X	X	X	P	0	0	[reg] Rotate contents of register left through Carry.
	RRC	reg	2	X	X	X	P	0	0	[reg] Rotate contents of register right with branch Carry.
	RR	reg	2	X	X	X	P		0	[reg] Rotate contents of register right through Carry.
	SLA	reg	2	X	X	X	P		0	 [reg] Shift contents of register left and clear LSB (Arithmetic Shift)
	SRA	reg	2	X	X	X	P		0	[reg] Shift contents of register right and preserve MSB (Arithmetic Shift)

Table 7-2. A Summary of the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS C	Z	S	P/O	A_C	N	OPERATION PERFORMED
REGISTER SHIFT AND ROTATE (Continued)	SRL		2	X	X	X	P	0	0	[reg] Shift contents of register right and clear MSB (Logical Shift).
	RLD		2		X	X	P	0	0	Rotate one BCD digit left between the Accumulator and memory location (implied addressing) Contents of the upper half of the Accumulator are not affected.
	RRD	reg	2		X	X	P	0	0	Rotate one BCD digit right between the Accumulator and memory location (implied addressing) Contents of the upper half of the Accumulator are not affected.
BIT MANIPULATION	BIT	b,reg	2		X	?	?	1	0	Z—$\overline{[reg]b}$ Zero flag contains complement of the selected register bit.
	BIT	b,(HL) b,(xy + disp)	2 4		X	?	?	1	0	Z—$\overline{[(HL)]b}$ or Z—$\overline{[(xy] + disp)b}$ Zero flag contains complement of selected bit of the memory location (implied addressing or base relative addressing).
	SET	b,reg	2							reg(b)—1 Set indicated register bit.
	SET	b,(HL) b,(xy + disp)	2 4							[(HL)](b)—1 or [(xy] + disp)(b)—1 Set indicated bit of memory location (implied addressing or base relative addressing).
	RES	b,reg	2							reg(b)—0 Reset indicated register bit.
	RES	b,(HL) b,(xy + disp)	2 4							[(HL)](b)—0 or [(xy] + disp)(b)—0 Reset indicated bit in memory location (implied addressing or base relative addressing).
STACK	PUSH	qq xy	1 2							[(SP)-1]—[qq(HI)] [(SP)-2]—[qq(LO)] [SP]—[SP]-2 Put contents of register pair or index register on top of Stack and decrement Stack Pointer.
	POP	qq xy	1 2							[qq(LO)]—[(SP)] [qq(HI)]—[(SP) + 1] [SP]—[SP] + 2 Put contents of top of Stack in register pair or index register and increment Stack Pointer.
	EX	(SP),HL (SP),xy	1 2							[H]—[(SP) + 1] [L]—[(SP)] Exchange contents of HL or index register and top of Stack.

Table 7-2 A Summar[illegible] the Z80 Instruction Set (Continued)

TYPE	MNEMONIC	OPERAND(S)	BYTES	STATUS						OPERATION PERFORMED
				C	Z	S	P/O	AC	N	
INTERRUPT	DI		1							Disable interrupts.
	EI		1							Enable interrupts.
	RST	n	1							[[SP]-1]—[PC(HI)] [[SP]-2]—[PC(LO)] [SP]—[SP]-2 [PC]—$(8 \cdot n)_{16}$ Restart at designated location.
	RETI		2							Return from interrupt.
	RETN		2							Return from nonmaskable interrupt.
	IM	0 1 2	2							Set interrupt mode 0, 1, or 2.
STATUS	SCF		1	1				0	0	C — 1 Set Carry flag.
	CCF		1	X				X	0	$C — \overline{C}$ Complement Carry flag.
	NOP		1							No operation — volatile memories are refreshed
	HALT		1							CPU halts, executes NOPs to refresh volatile memories

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

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.

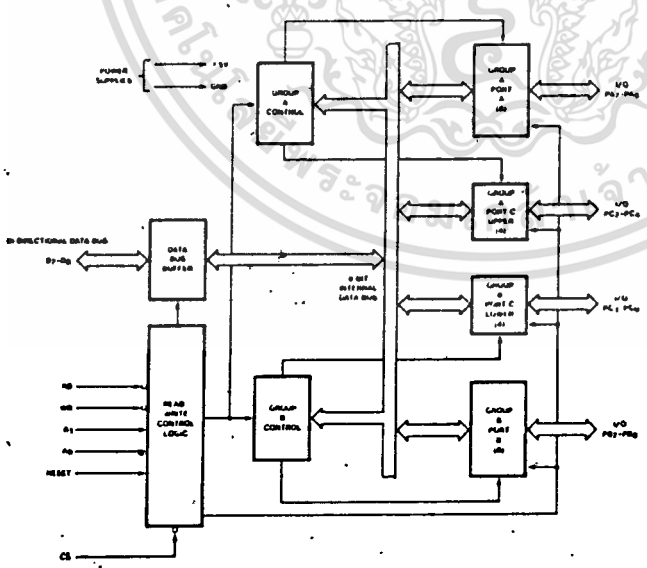


Figure 1. 8255A Block Diagram

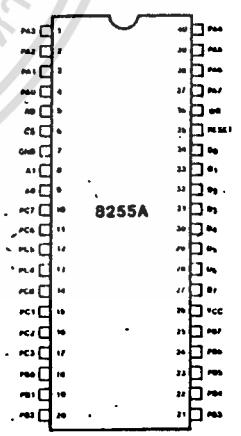


Figure 2. Pin Configuration

©INTEL CORPORATION, 1982.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(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.

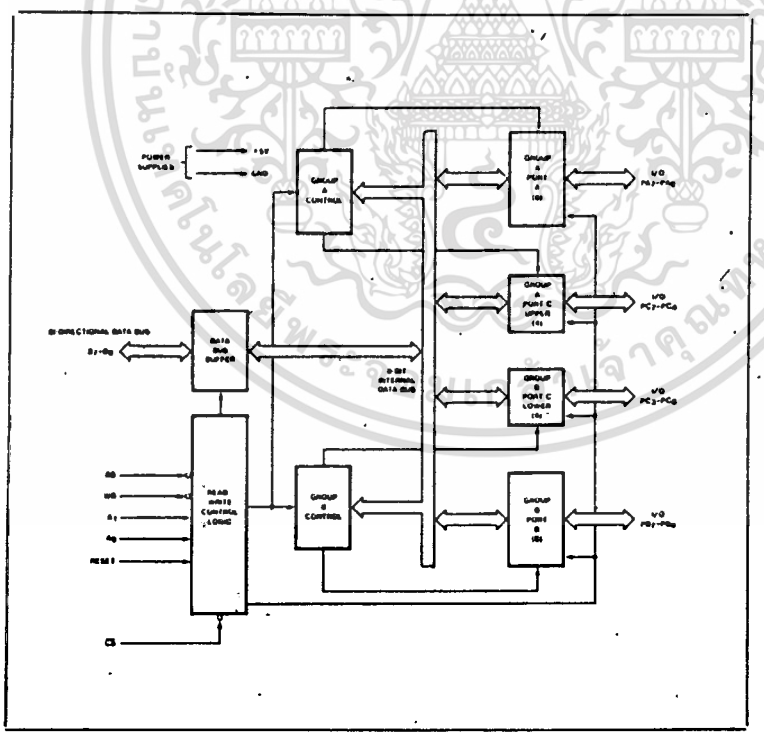
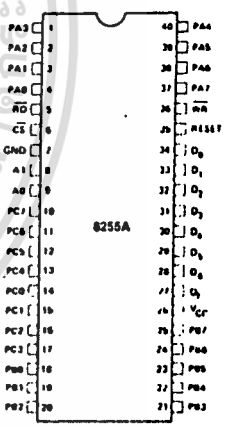


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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.

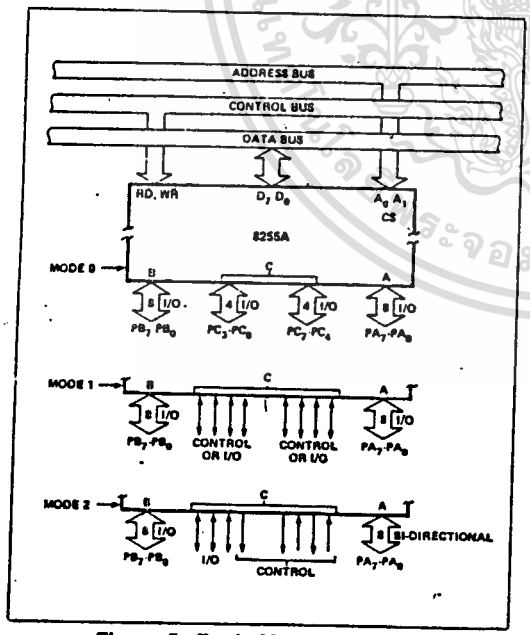


Figure 5. Basic Mode Definitions and Bus Interface

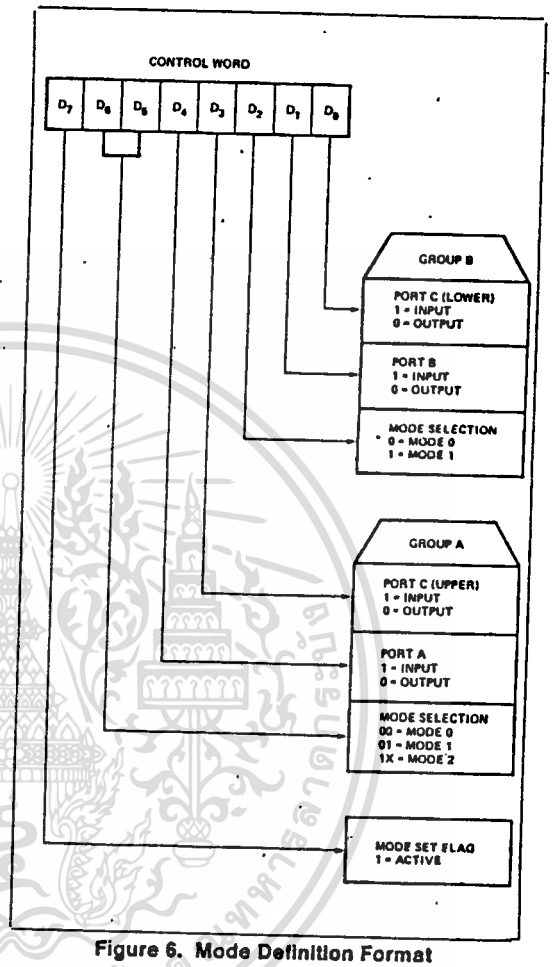


Figure 6. Mode Definition Format

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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

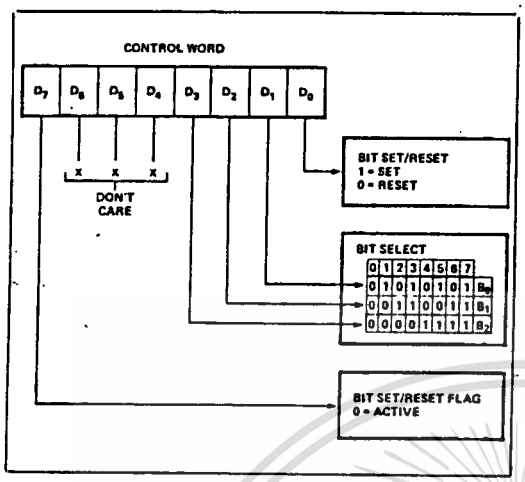


Figure 7. Bit Set/Reset Format

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

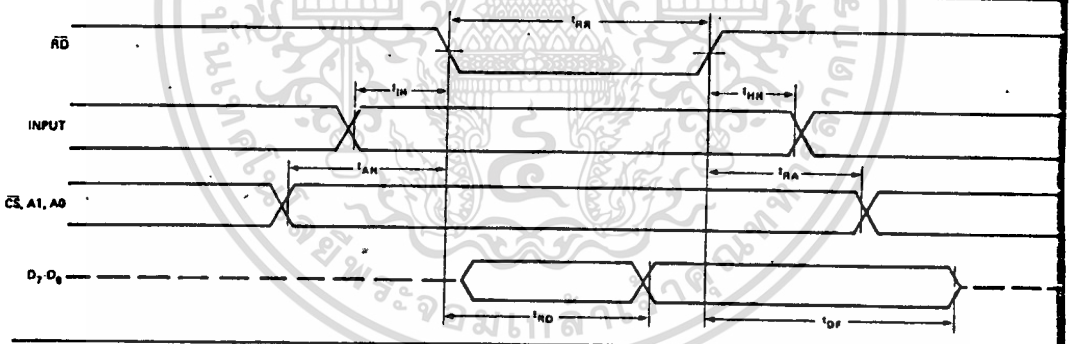
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

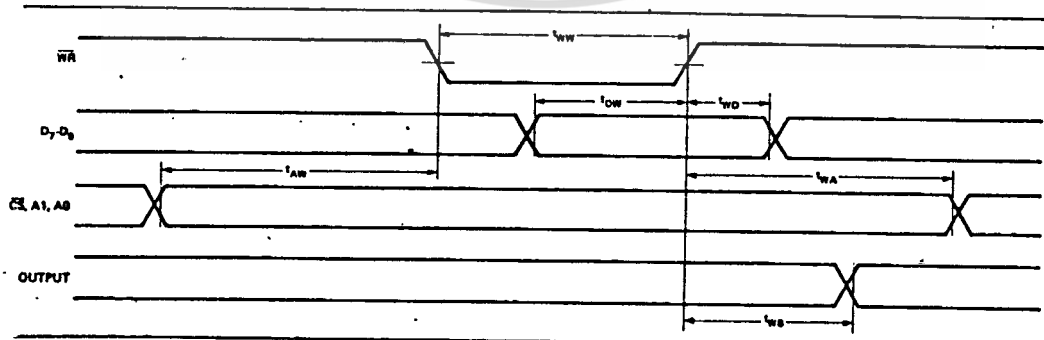
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

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)

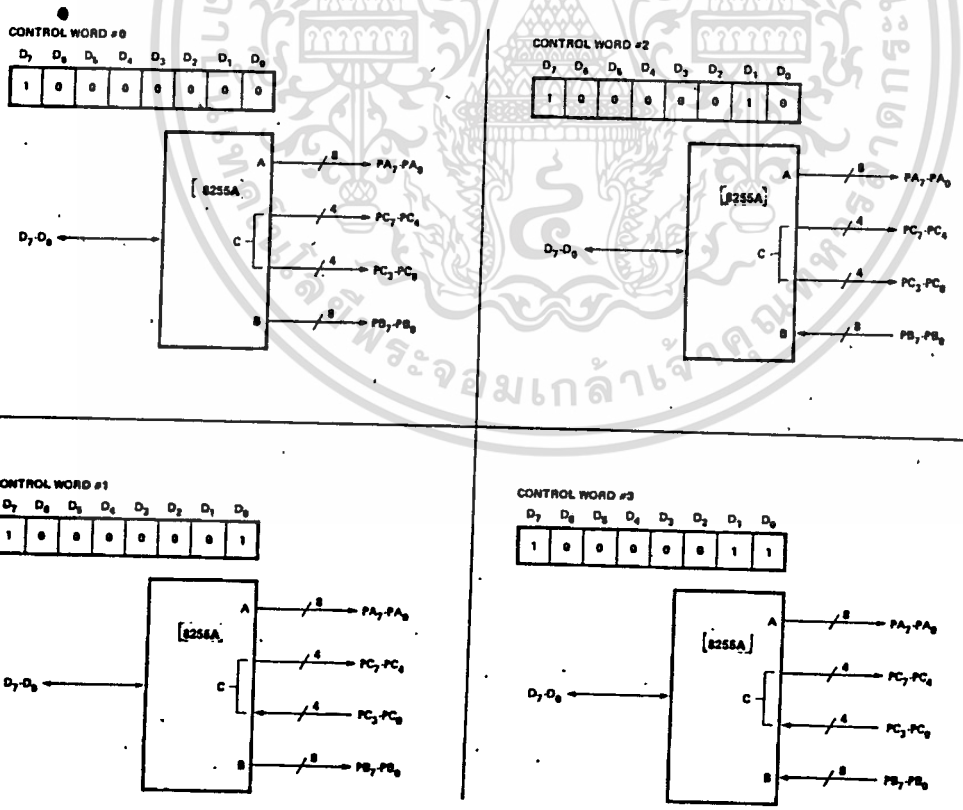
AFN-00744

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

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

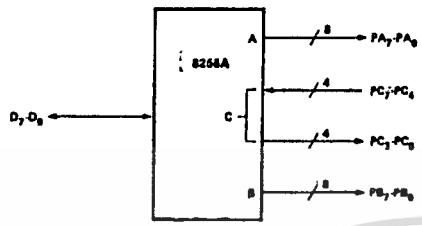
MODE 0 Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

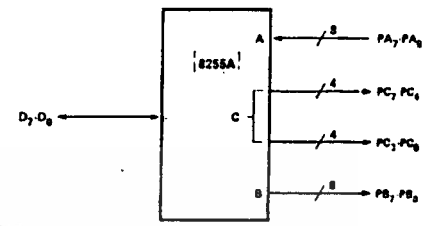
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



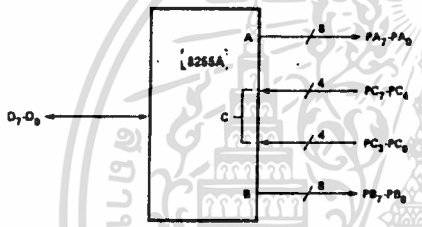
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



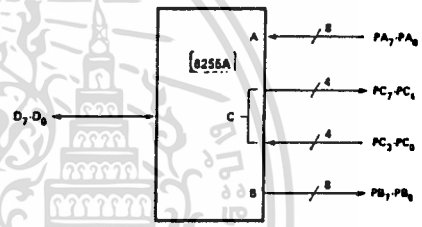
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



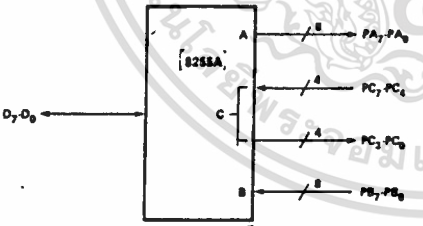
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



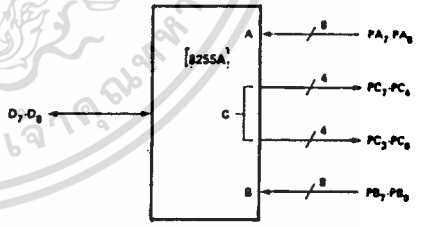
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



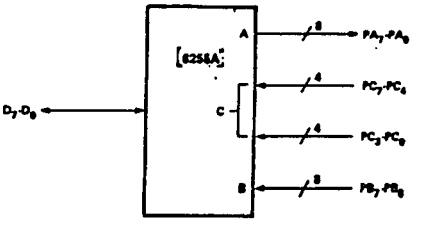
CONTROL WORD #10

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



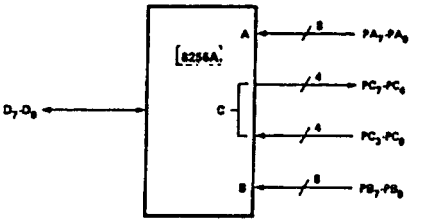
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



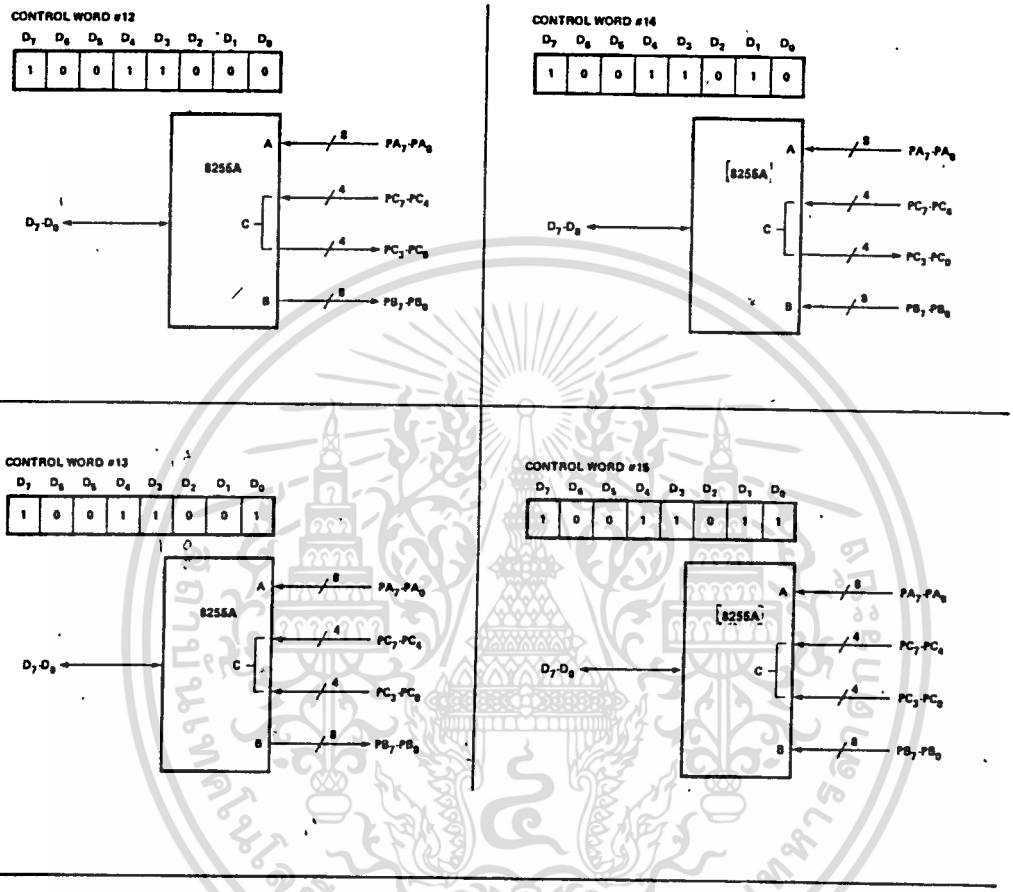
CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



AFN-00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Control Signal Definition

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.

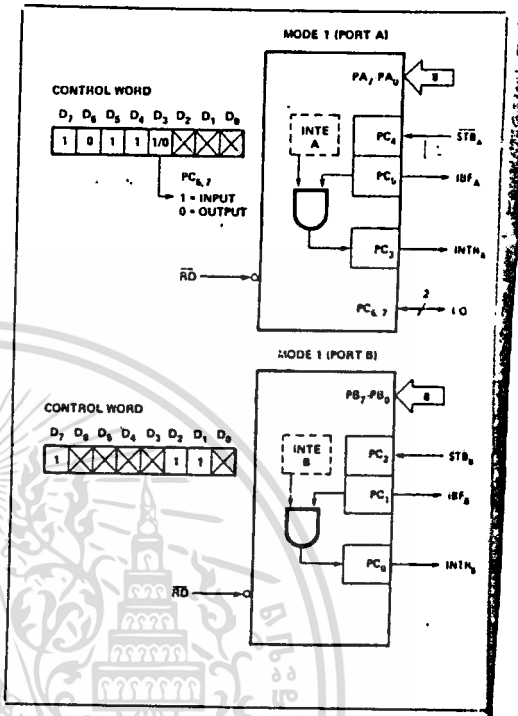


Figure 8. MODE 1 Input

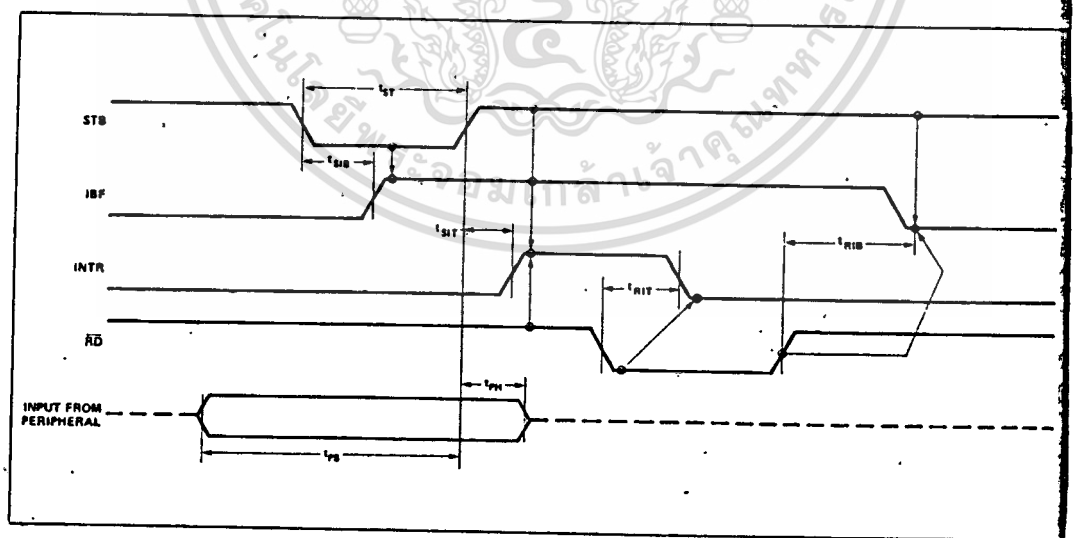


Figure 9. MODE 1 (Strobed Input)

AFN-00744

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Output Control Signal Definition

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

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

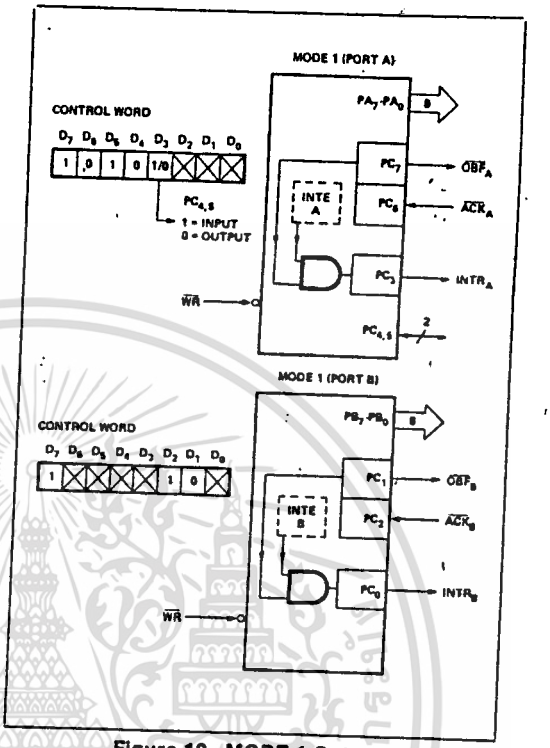


Figure 10. MODE 1 Output

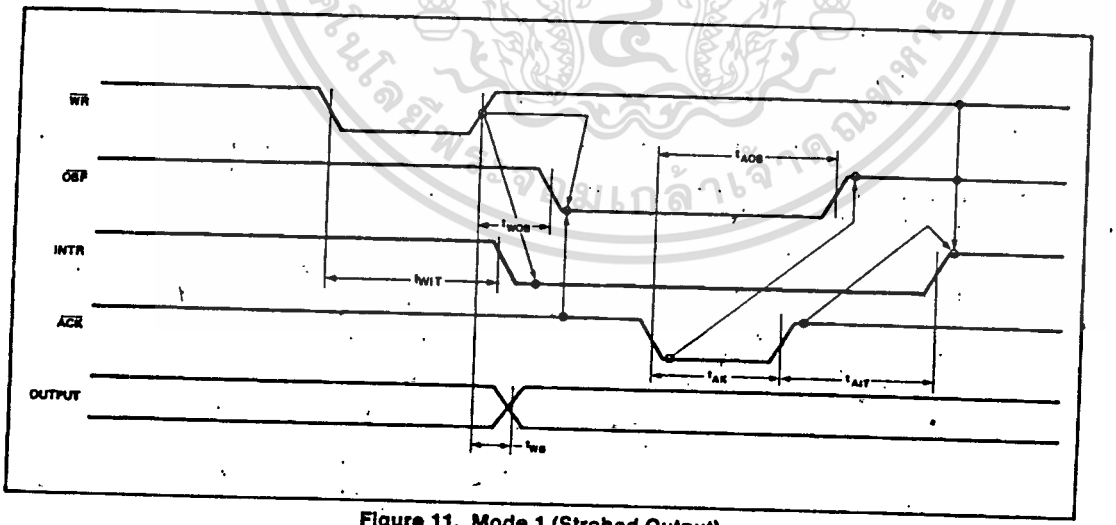


Figure 11. Mode 1 (Strobed Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

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.

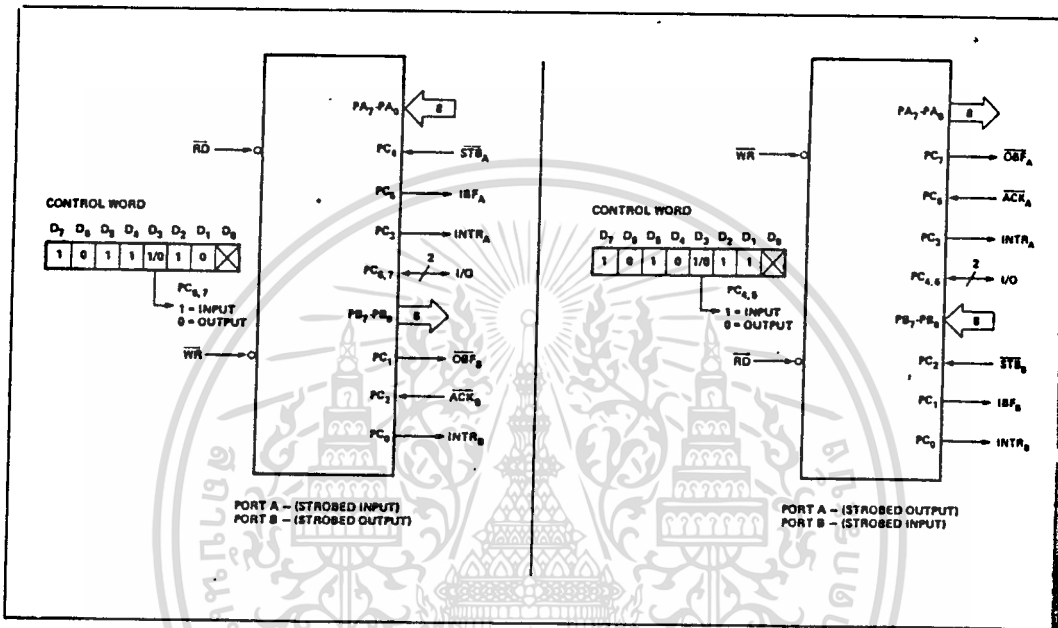


Figure 12. Combinations of MODE 1

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 Ful). The 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)

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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

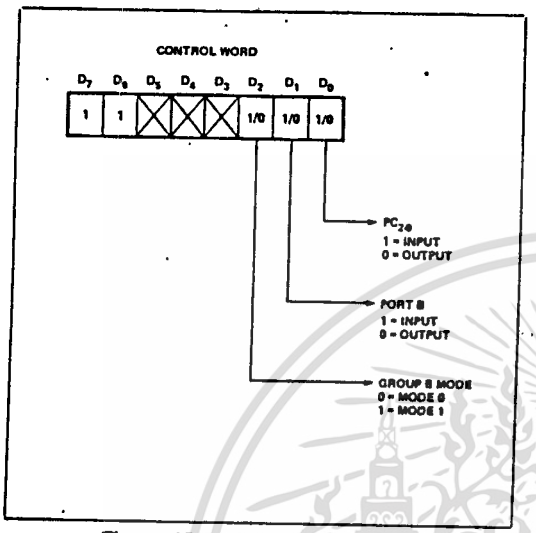


Figure 13. MODE Control Word

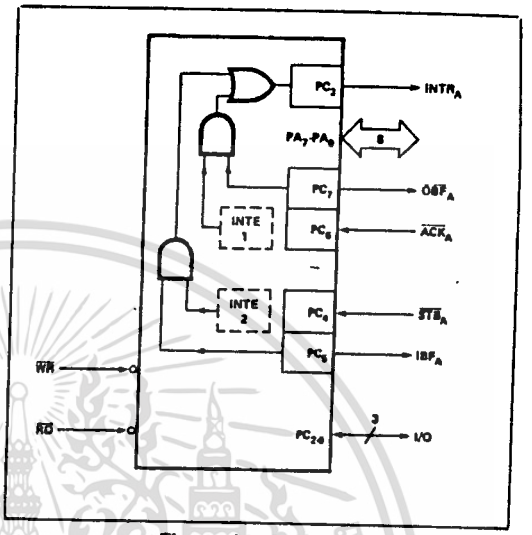


Figure 14. MODE 2

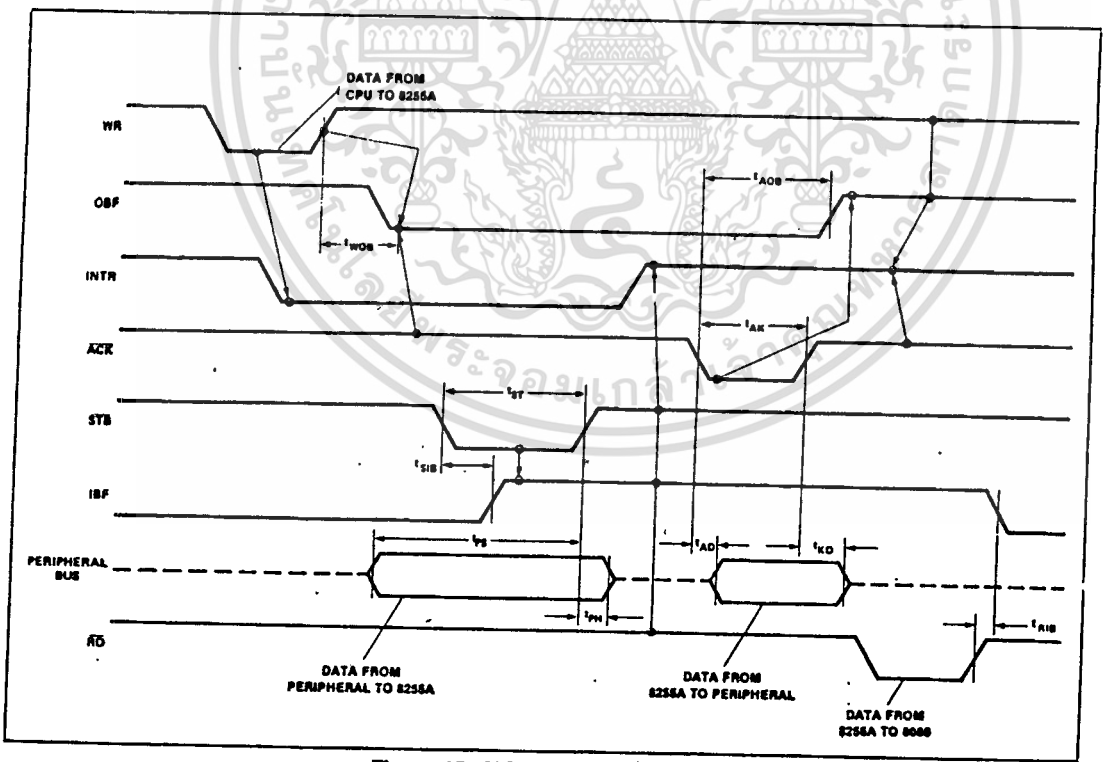


Figure 15. MODE 2 (Bidirectional)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

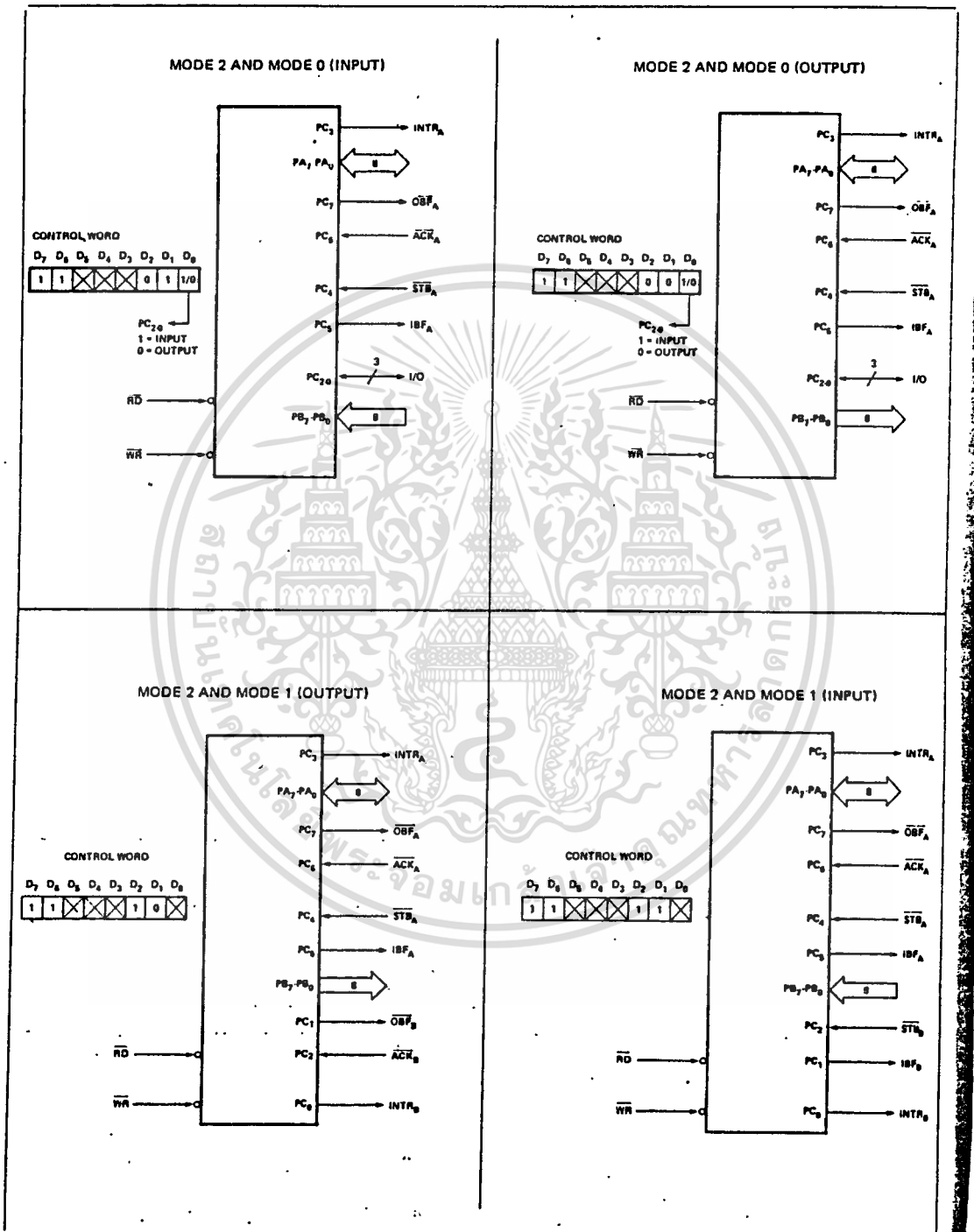


Figure 16. MODE ¼ Combinations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

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	—
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	$\overline{OBF}_A$	$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

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 1mA 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.

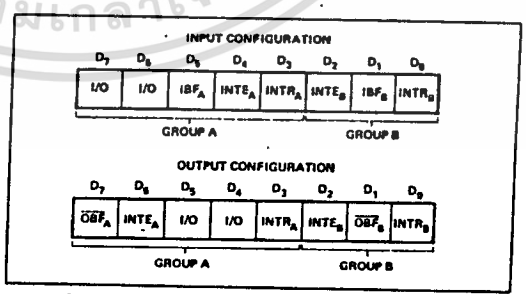


Figure 17. MODE 1 Status Word Format

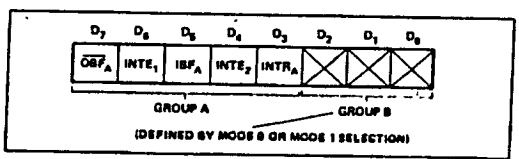


Figure 18. MODE 2 Status Word Format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 present a few examples of typical applications of the 8255A.

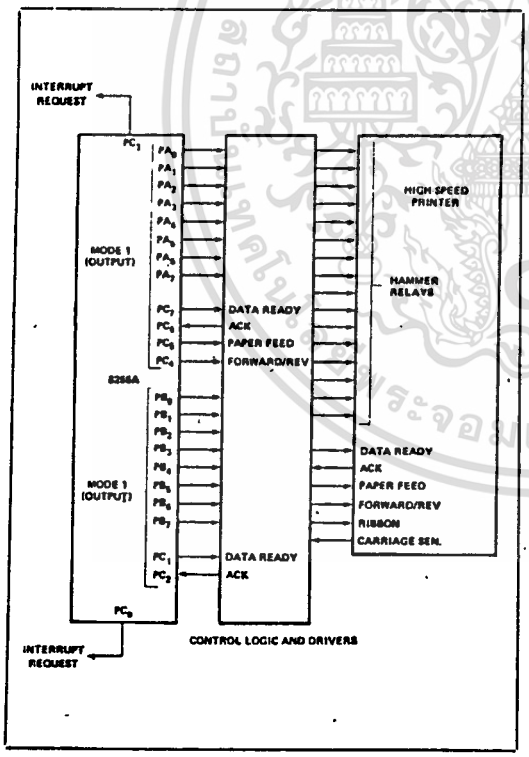


Figure 19. Printer Interface

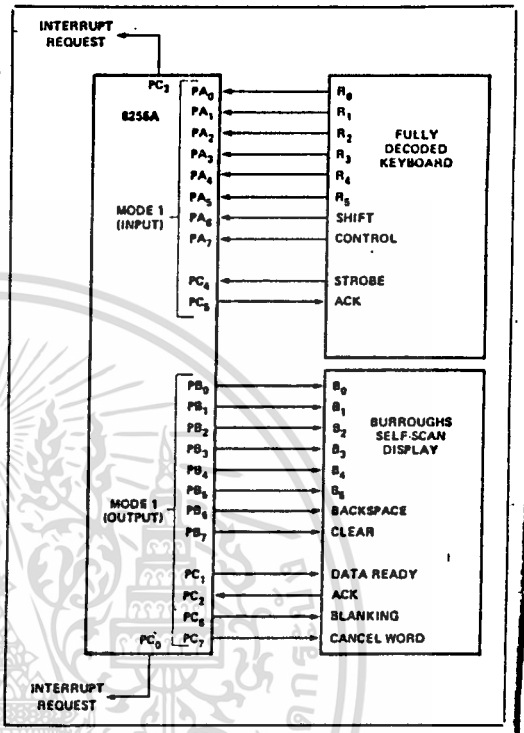


Figure 20. Keyboard and Display Interface

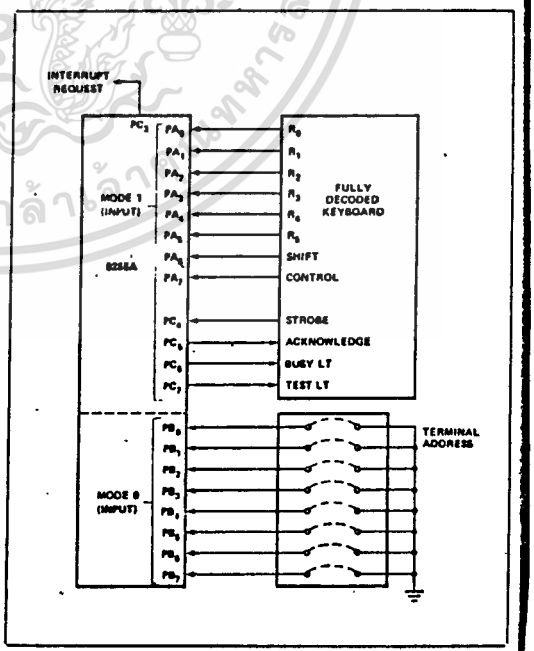


Figure 21. Keyboard and Terminal Address Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

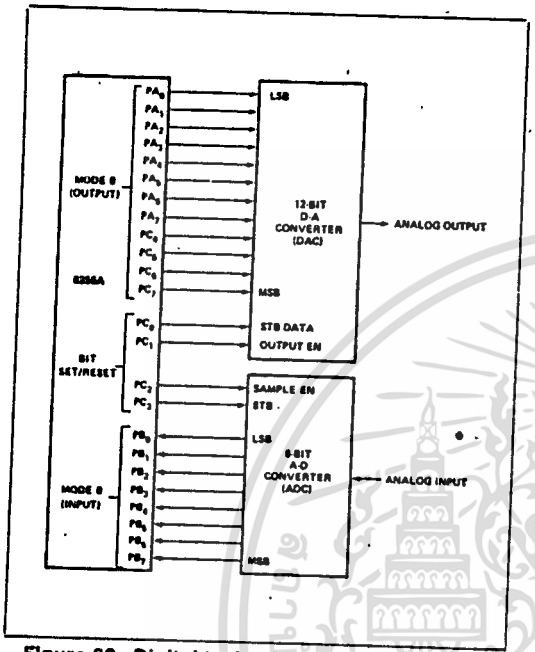


Figure 22. Digital to Analog, Analog to Digital

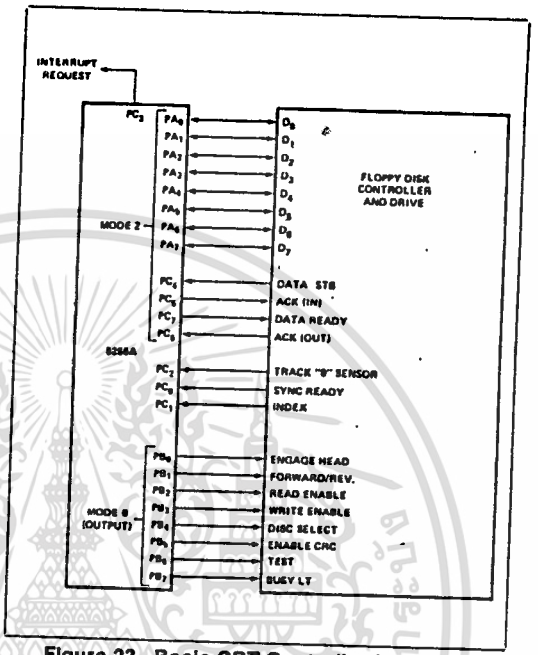


Figure 23. Basic CRT Controller Interface

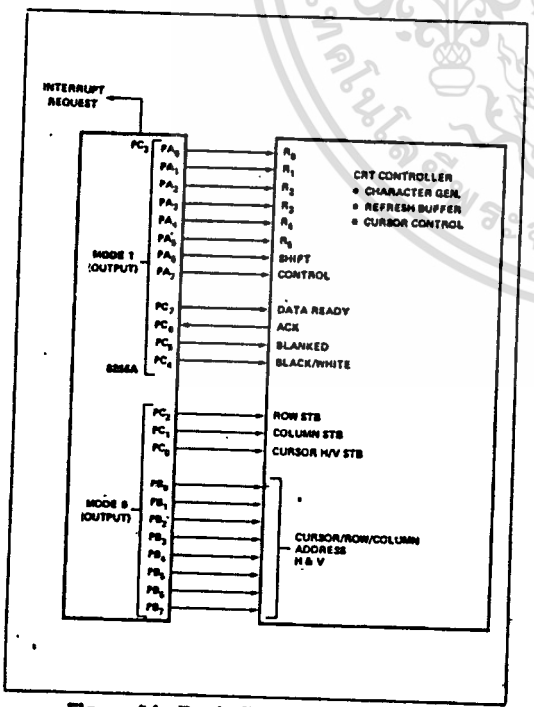


Figure 24. Basic Floppy Disc Interface

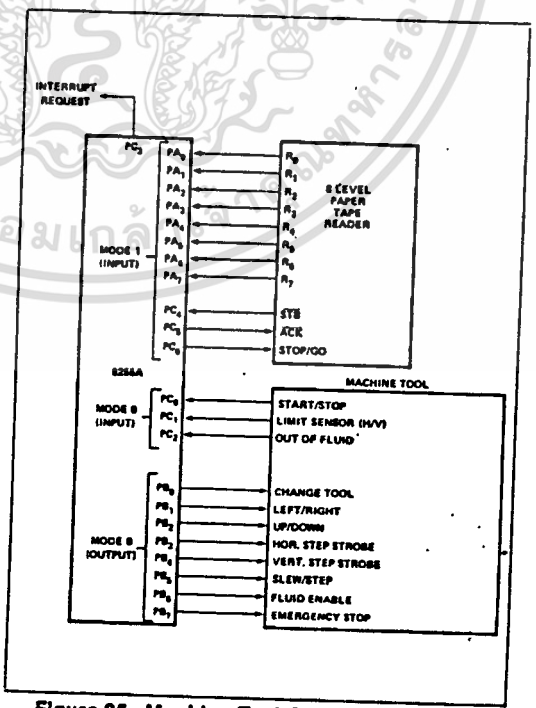


Figure 25. Machine Tool Controller Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin
With Respect to Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1 Watt

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V)*

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5mA
V_{OL} (PER)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7mA
V_{OH} (DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = -400μA
V_{OH} (PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = -200μA
I_{DAR} [1]	Darlington Drive Current	-1.0	-4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to .45V

NOTE:
1. Available on any 8 pins from Port B and C.

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V)*

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ[1]		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS (Continued)

WRITE

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

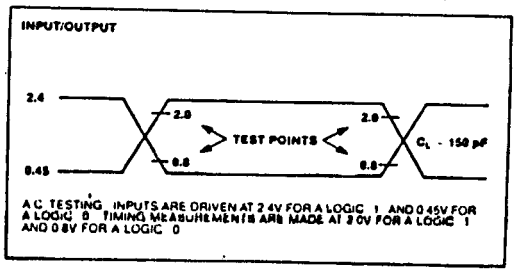
OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{WB}	WR = 1 to Output[1]		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output[1]		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0[1]		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1[1]		350		350	ns
t_{SIB}	STB = 0 to IBF = 1[1]		300		300	ns
t_{RIB}	RD = 1 to IBF = 0[1]		300		300	ns
t_{RIT}	RD = 0 to INTR = 0[1]		400		400	ns
t_{SIT}	STB = 1 to INTR = 1[1]		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1[1]		350		350	ns
t_{WIT}	WR = 0 to INTR = 0[1,3]		450		450	ns

NOTES:
1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR↑ may occur as early as $\overline{WR}$↓.

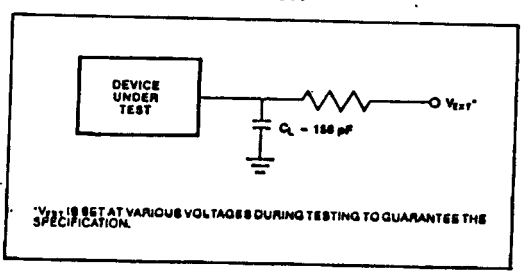
* For Extended Temperature EXPRESS, use M8255A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT

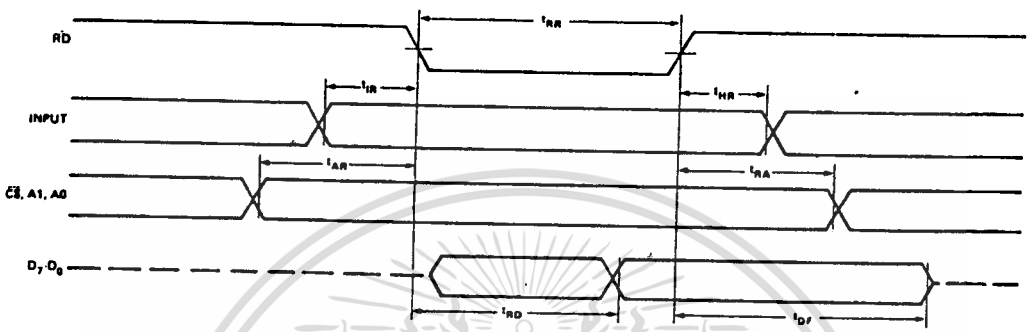


*V_{TST} IS SET AT VARIOUS VOLTAGES DURING TESTING TO GUARANTEE THE SPECIFICATION.

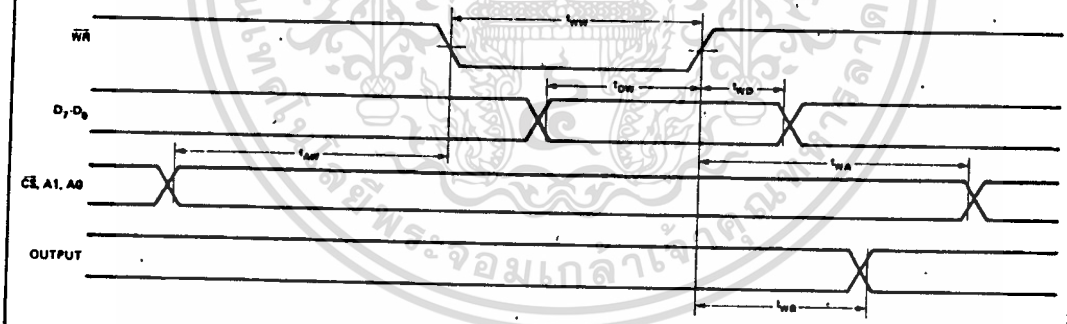
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

MODE 0 (BASIC INPUT)



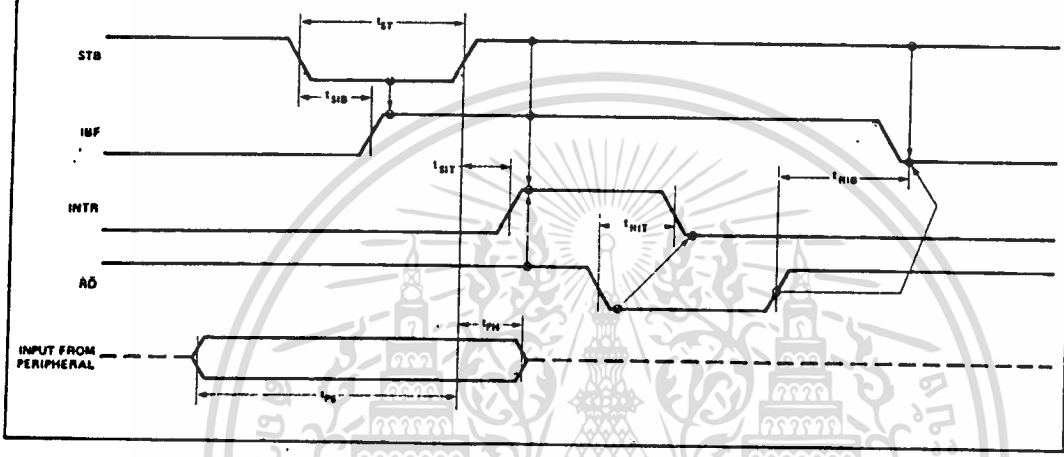
MODE 0 (BASIC OUTPUT)



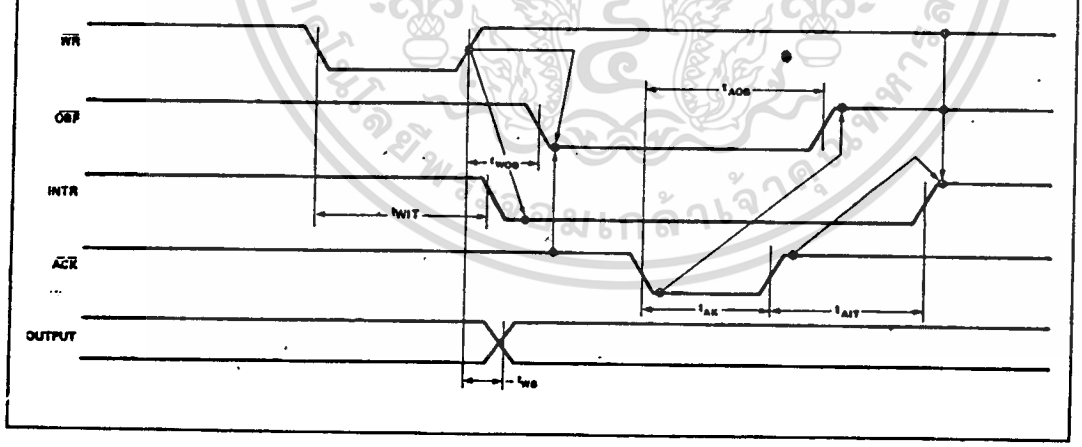
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



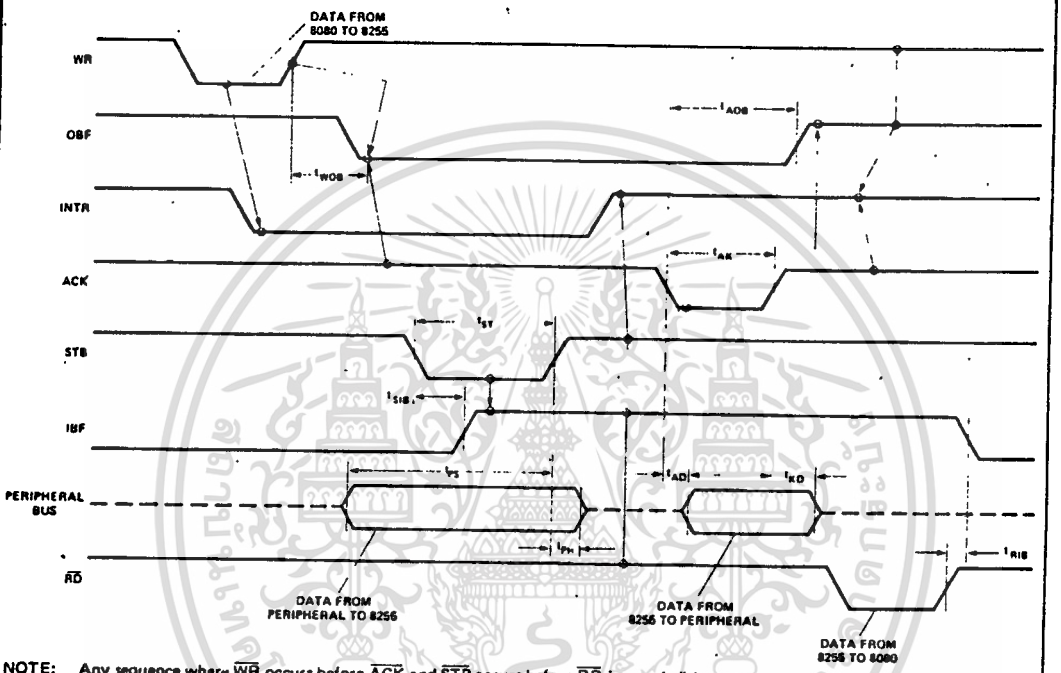
MODE 1 (STROBED OUTPUT)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

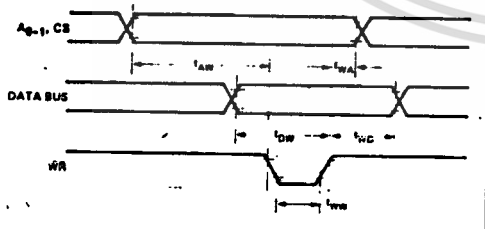
WAVEFORMS (Continued)

MODE 2 (BIDIRECTIONAL)

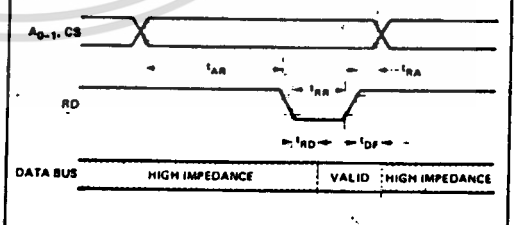


NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

WRITE TIMING



READ TIMING



AFN-00744C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่อง CENTRAL PROCESSING UNIT เรื่องนี้ คงจะไม่สำเร็จไปได้ด้วยดี ถ้าหากปราศจากคำแนะนำอันมีคุณค่าของอาจารย์ชาญชัย และคำแนะนำจากเพื่อน ๆ อีกหลายคน ซึ่งไม่สามารถนำชื่อมากล่าว ณ ที่นี่ได้หมด แต่ผู้จัดทำก็สำนึกถึงความหวังดี และกำลังใจจากอาจารย์ และเพื่อนๆ เสมอ ขอขอบคุณ ณ.โอกาสนี้ด้วย

นายชัยพันธ์ ศรีรุ่งเรือง
นายอุดม ธีรธรรมานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ยืน ภู่วรวรรณ และ วัฒนา เชียงกูล, "ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์ (Z-80 MICROPROCESSOR)", บริษัท ซีเอ็ดยูเคชั่น จำกัด, พ.ศ. 2527.

2. ชูชัย ชนสารตั้งเจริญ, ดนัย แสงสุริยศิลป์, ทินกร ดุ๊ก, ธงชัย อุดมกิจโกศล และ ธานินทร์ ถาวรศาสนวงศ์, "การใช้ Z 80", ซิสกส์เซ็นเตอร์ การพิมพ์, พ.ศ. 2529.

3. INTEL, "MICROPROCESSOR AND PERIPHERAL HANDBOOK", INTEL CORPORATION, 1983.

4. LANCE A LEVENTHAL, " Z80 ASSEMBLY LANGUAGE PROGRAMMING", 1980.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************************************
;*                                                                       *
;*       MAIN PROJECT       COMPUTERLIZED AUTOMOTIVE CAR                 *
;*       SUB  PROJECT       CENTRAL PROCESSING UNIT                      *
;*       PROGRAM            MONITOR                                      *
;*       WRITE  BY                                                       *
;*              MR.CHAIYAPAN  SRIRUNGROUNG                               *
;*              MR.UDOM     TEEPATRAMMANON                               *
;*       ADVISOR                                                         *
;*              MR.CHANCHAI PISITVITTAYANON                              *
;*       DEPARTMENT  OF  INDUSTRIAL INSTRUMENTATION TECHNOLOGY           *
;*       FACULTY   OF  ENGINEERING                                       *
;*       KING MONGKUT'S INSTITUTE OF TECHNOLOGY                          *
;*       LADKRABANG  BANGKOK 10520 THAILAND                              *
;*                                                                       *
;*************************************************************************


;######################## SYSTEM EQUATION ##########################
;equ of set port
    DC_P1:  EQU 91H     ;set port#1 cl i/p,ch o/p, b o/p, a i/p
     C_P1:  EQU 03H     ;controlport#1
    IO_C1:  EQU 02H
     O_B1:  EQU 01H     ;port b o/p used to speech synthesizer
     I_A1:  EQU 00H     ;port a i/p used to sensor part
    DC_P2:  EQU 88H     ;set port#2 cl o/p,ch i/p, a & b o/p
     C_P2:  EQU 0E3H    ;controlport#2
    IO_C2:  EQU 0E2H
     O_B2:  EQU 0E1H    ;port b o/p used to display part
     O_A2:  EQU 0E0H    ;port a o/p drive led
;equ of select channal a/d
     CHA0:  EQU 00H
     CHA1:  EQU 10H
     CHA2:  EQU 20H
     CHA3:  EQU 30H
     CHA4:  EQU 40H
     CHA5:  EQU 50H
     CHA6:  EQU 60H
     CHA7:  EQU 70H
;equ of speech
      SOT:  EQU 1AH     ;start of text
      EOT:  EQU 1BH     ;end of text
       _A:  EQU 0AH     ;normal speech
       _B:  EQU 0BH     ;speed  speech
       _C:  EQU 0CH     ;fuel   speech
;equ of convert ascii
   ASCON:  EQU 30H     ;add to convert ascii
      MH:  EQU 0F0H
      ML:  EQU 0FH
;equ of display
    HIGH:  EQU 0FFH    ;used send strobe
     LOW:  EQU 00H     ;used send strobe
    CON1:  EQU 00H

;###################### END SYSTEM EQUATION ########################
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;------------------------- MAIN PART -------------------------
          ORG   0000H
    MAIN:
          LD    B,0
       S: DJNZ  S
          LD    SP,0FFFFH
          CALL  SET_P
          CALL  BACK_G      ;send background

          CALL  LEFT
          CALL  RIGHT
          CALL  BEAM_L
          CALL  BEAM_R
          CALL  BK_L
          CALL  BK_R
          CALL  HB
          CALL  DIAL

   SLOOP:
          LD    B,2
      S1: PUSH  BC
          CALL  R_T_D
          POP   BC
          DEC   B
          JR    NZ,S1

          CALL  T_SPEECH
          CP    0CAH
          JR    NC,Z0
          JR    Z,Z0
          JR    C,Z1
       Z0: CALL  D_TEM
       Z1: NOP

          CALL  DELAY
          CALL  DELAY

          LD    B,2
      S2: PUSH  BC
          CALL  R_F_D
          POP   BC
          DEC   B
          JR    NZ,S2

          CALL  F_SPEECH
          CP    0
          JR    Z,U0
          JR    NZ,U1
      U0: CALL  R_0
      U1: NOP


          CALL  DELAY
          CALL  DELAY

          LD    B,2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    S3:  PUSH  BC
         CALL  R_A_D
         POP   BC
         DEC   B
         JR    NZ,S3

         CALL  DELAY
         CALL  DELAY

         LD    B,2
    S4:  PUSH  BC
         CALL  R_V_D
         POP   BC
         DEC   B
         JR    NZ,S4

         CALL  DELAY
         CALL  DELAY

         LD    D,6
LOOPN1:  PUSH  DE

         LD    B,2
    N1:  PUSH  BC
         CALL  R_S_D
         POP   BC
         DEC   B
         JR    NZ,N1

         CALL  DELAY

         LD    B,2
    N2:  PUSH  BC
         CALL  R_P_D
         POP   BC
         DEC   B
         JR    NZ,N2

         CALL  DELAY

         POP   DE
         DEC   D
         JR    NZ,LOOPN1

         CALL  DELAY
         CALL  DELAY

         LD    B,2
    S5:  PUSH  BC
         CALL  R_T_D
         POP   BC
         DEC   B
         JR    NZ,S5

         CALL  DELAY
         CALL  DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LD    D,6
AGAIN:    PUSH  DE

          LD    B,2
N3:       PUSH  BC
          CALL  R_G_D
          POP   BC
          DEC   B
          JR    NZ,N3

          CALL  DELAY

          LD    B,2
N4:       PUSH  BC
          CALL  R_P_D
          POP   BC
          DEC   B
          JR    NZ,N4

          CALL  DELAY

          POP   DE
          DEC   D
          JR    NZ,AGAIN


          CALL  LIGHT_L
          CALL  LIGHT_R

          CALL  TE_R
          CALL  TE_L
          CALL  TE_B
          CALL  TE_BK

          JR    SLOOP

;*************** SUBROUTINE SENSOR SYSTEM LIGHT **************
TE_R:
          CALL   INCS3
          OUT   (0E0H),A
          JP    LOOP
          BIT   6,A
          JR    Z,IN_P
          JR    NZ,_SOUND01

IN_P:     CALL   INCS3
          LD    B,20H
          AND   B
          CP    00
          JR    Z,_SOLND00
          JR    NZ,L_BR
L_ER:     CALL  SH_RIGHT
          JP    CONTI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_SOUND00:  CALL  OP_R
           JP    CONTI

_SOUND01:  CALL  OP_R

  CONTI:   RET


   TE_L:
           CALL   INCS3
           OUT    (0E0H),A
           JP     LOOP
           BIT    6,A
           JR     Z,IN_P
           JR     NZ,_SOUND03

   IN_P:   CALL   INCS3
           LD     B,10H
           AND    B
           CP     00
           JR     Z,_SOUND02
           JR     NZ,L_BR
   L_BR:   CALL   SH_LEFT
           JP     CONTI

_SOUND02:  CALL OP_L
           JP   CONTI

_SOUND03:  CALL OP_L

  CONTI:   RET

   TE_B:
           CALL   INCS3
           OUT    (0E0H),A
           JP     LOOP
           BIT    0,A
           JR     Z,IN_P
           JR     NZ,_SOUND05

   IN_P:   CALL   INCS2
           LD     B,0C0H
           AND    B
           CP     00
           JR     Z,_SOUND04
           JR     NZ,L_BR
   L_BR:   CALL   SH_BEAML
           CALL   SH_BEAMR
           JP     CONTI

_SOUND04:  CALL  OP_B
           JP    CONTI

_SOUND05:  CALL  OP_B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CONTI:  RET

    TE_BK:
            CALL   _INCS4
            OUT   (0E0H),A
            JP    LOOP
            BIT   6,A
            JR    Z,IN_P
            JR    NZ,_SOUND07

     IN_P:  CALL   _INCS4
            LD    B,06H
            AND   B
            CP    00
            JR    Z,_SOUND06
            JR    NZ,L_BR
     L_BR:  CALL  SH_BRL
            CALL  SH_BKR
            JP    CONTI

 _SOUND06:  CALL  OP_R
            JP    CONTI

 _SOUND07:  CALL  CP_R

    CONTI:  RET

;****************** END SUB SENSOR SYSTEM LIGHT ******************

;************ SUBROUTINE READN PORT OF SYSTEM LIGHT *************

   _INCS2:
            LD    B,3
    VL2:    IN    A,(60H)
            DEC   B
            JR    NZ,VL2
            RET

   _INCS3:
            LD    B,3
    VL3:    IN    A,(60H)
            DEC   B
            JR    NZ,VL3
            RET

   _INCS4:
            LD    B,3
    VL4:    IN    A,(80H)
            DEC   B
            JR    NZ,VL4
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  _INCS5:
            LD    B,3
    VL5:    IN    A,(0A0H)
            DEC   B
            JR    NZ,VL5
            RET
;**************** END SUB IN PORT OF SYSTEM LIGHT ***************

;******************** SUBROUTION CALL SPEECH *******************
      R_0:
            LD    A,09H
            CALL  NORMAL
            RET
    0_TEM:
            LD    A,08H
            CALL  NORMAL
            RET
    _OP_R:
            LD    A,00H
            CALL  NORMAL
            RET

    _CP_R:
            LD    A,01H
            CALL  NORMAL
            RET

    _OP_L:
            LD    A,02H
            CALL  NORMAL
            RET

    _CP_L:
            LD    A,03H
            CALL  NORMAL
            RET

    _OP_B:
            LD    A,04H
            CALL  NORMAL
            RET

    _CP_B:
            LD    A,05H
            CALL  NORMAL
            RET
    _OP_R:
            LD    A,06H
            CALL  NORMAL
            RET
    _CP_R:
            LD    A,07H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL  NORMAL
            RET

;*****************************************************************

;****************** SUBROUTINE LIGHT SYSTEM DISPLAY **************
    LEFT:
            PUSH  HL
            LD    L,13
            LD    H,0FEH
            CALL  GOTO_D13
            POP   HL
            RET

   RIGHT:
            PUSH  HL
            LD    L,28
            LD    H,0FEH
            CALL  GOTO_D13
            POP   HL
            RET

  BEAM_L:
            PUSH  HL
            LD    L,10
            LD    H,2AH
            CALL  GOTO_D13
            POP   HL
            RET

  BEAM_R:
            PUSH  HL
            LD    L,31
            LD    H,2AH
            CALL  GOTO_D13
            POP   HL
            RET

LIGHT_L:
            PUSH  HL
            LD    L,10
            LD    H,2AH
            CALL  GOTO_D15
            POP   HL
            RET

LIGHT_R:
            PUSH  HL
            LD    L,31
            LD    H,2AH
            CALL  GOTO_D15
            POP   HL
            RET

   BK_L:
```

```
            PUSH  HL
            LD    L,13
            LD    H,0FEH
            CALL  GOTO_D18
            POP   HL
            RET

      BK_R:
            PUSH  HL
            LD    L,28
            LD    H,0FEH
            CALL  GOTO_D18
            POP   HL
            RET

      DIAL:
            PUSH  HL
            LD    L,7
            LD    H,0FEH
            CALL  GOTO_D18
            POP   HL
            RET

       HS:
            PUSH  HL
            LD    L,37
            LD    H,0FEH
            CALL  GOTO_D18
            POP   HL
            RET

    SH_LEFT:
            PUSH  HL
            LD    L,13
            LD    H,0D6H
            CALL  GOTO_D13
            POP   HL
            RET

  SH_RIGHT:
            PUSH  HL
            LD    L,28
            LD    H,0D6H
            CALL  GOTO_D13
            POP   HL
            RET

SH_BEAM_L:
            PUSH  HL
            LD    L,10
            LD    H,0D6H
            CALL  GOTO_D13
            POP   HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
            RET

SH_BEAM_R:
            PUSH  HL
            LD    L,31
            LD    H,0DBH
            CALL  GOTO_D13
            POP   HL
            RET

SH_LIGHT_L:
            PUSH  HL
            LD    L,10
            LD    H,0DBH
            CALL  GOTO_D15
            POP   HL
            RET

SH_LIGHT_R:
            PUSH  HL
            LD    L,31
            LD    H,0DBH
            CALL  GOTO_D15
            POP   HL
            RET

  SH_BK_L:
            PUSH  HL
            LD    L,13
            LD    H,0DBH
            CALL  GOTO_D18
            POP   HL
            RET

  SH_BK_R:
            PUSH  HL
            LD    L,28
            LD    H,0DBH
            CALL  GOTO_D18
            POP   HL
            RET

SH_DIAL:
            PUSH  HL
            LD    L,7
            LD    H,0D9H
            CALL  GOTO_D18
            POP   HL
            RET

   SH_HB:
           PUSH  HL
           LD    L,37
```

```
            LD    H,0DBH
            CALL  GOTO_D13
            POP   HL
            RET

;************************ END SUB LIGHT ************************

;******** SUBROUTINE READ DATA FROM A/D CHA0 TO DISPLAY *******
   R_T_D:
            LD    B,CHA0
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    IY,TABLE1
            CALL  TEMP1
            RET
   T_SPEECH:
            LD    B,CHA0
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            RET

;************************ END SUB R_T_D ************************

;******** SUBROUTINE READ DATA FROM A/D CHA2 TO DISPLAY *******
   R_F_D:
            LD    B,CHA2
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    IY,TABLE2
            CALL  FUEL2

            LD    B,CHA2
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    HL,TABLE2_1
            CALL  FUEL2_1
            RET

 F_SPEECH:
            LD    B,CHA2
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            RET

;************************ END SUB R_F_D ************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
;******** SUBROUTINE READ DATA FROM A/D CHA4 TO DISPLAY *******
   R_S_D:
            LD    B,CHA4
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    IY,TABLE3
            CALL  SPEED3

            LD    B,CHA4
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    HL,TABLE3_1
            CALL  SPEED3_1
            RET
;*************************** END SUB R_S_D ***************************


;******* SUBROUTINE READ DATA FROM A/D CHA5,6 TO DISPLAY ******
   _R_A_D:
            LD    B,CHA5
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    IY,TABLE4
            CALL  _AMP4

            LD    B,CHA5
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    HL,TABLE4_1
            CALL  _AMP4_1
            RET

   R_A_D:
            LD    B,CHA6
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  D_DELAY
            LD    IY,TABLE5
            CALL  AMP5

            LD    B,CHA6
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
            CALL  0_DELAY
            LD    HL,TABLE5_1
            CALL  AMP5_1
            RET
;************************ END SUB R_A_D ***************************

;******** SUBROUTINE READ DATA FROM A/D CHA1 TO DISPLAY *******
   R_V_D:
            LD    B,CHA1
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  0_DELAY
            LD    IY,TABLE6
            CALL  VOLT6

            LD    B,CHA1
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  0_DELAY
            LD    HL,TABLE5_1
            CALL  VOLT6_1
            RET
;************************ END SUB R_V_D ***************************

;******** SUBROUTINE READ DATA FROM A/D CHA3 TO DISPLAY *******
    R_P_D:
            LD    B,CHA3
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  0_DELAY
            LD    IY,TABLE7
            CALL  RPM7

            LD    B,CHA3
            CALL  READ
            CALL  READ
            OUT   (0E0H),A
            CALL  0_DELAY
            LD    HL,TABLE7_1
            CALL  RPM7_1
            RET
;************************ END SUB R_P_D ***************************

;********************** SUBROUTINE SET PORT ***********************
    SET_P:
            LD    A,DC_P1
            OUT   (C_PT),A
            LD    A,DC_P2
            OUT   (C_P2),A
            RET
;************************ END SUB SET PORT ************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******************** SUBROUTINE BACKGROUND **********************
   BACK_G:
           LD    BC,740      ; counter
           LD    HL,DATA     ; index
    NEXT1:
           LD    A,(HL)      ; get character
           CALL  SEND
           INC   HL
           DEC   BC
           LD    A,B
           OR    C
           JR    NZ,NEXT1
           RET
;*************** SUBROUTINE READ DATA FROM A/D *****************
;ch o/p 4,5,6
;ch i/p 0 (eoc)
;pass parameter by reg b
;get value by reg a
     READ:
           PUSH  BC
           LD    A,B
           OUT   (IO_C1),A
    WTLOW: IN    A,(IO_C1)
           BIT   0,A
           JR    NZ,WTLOW
     WTHI: IN    A,(IO_C1)
           BIT   0,A
           JR    Z,WTLOW
           IN    A,(I_A1)
           POP   BC
           RET
;************************** END SUB A/D ***************************

;************************ SUBROUTINE TEMP ************************
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table1)
    TEMP1:
           PUSH  HL
           CALL  OPEN2
           CALL  ASCII2
           LD    (0F800H),HL
           LD    L,04H
           CALL  GOTO_D6
           LD    A,(0F800H)
           LD    H,A
           LD    L,05H
           CALL  GOTO_D6
           POP   HL
           RET
;************************ END SUB TEMP ***************************

;********************** SUBROUTINE FUEL **************************
;pass parameter by reg a (data from a/d)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;pass parameter by reg iy (address of table2)
    FUEL2:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0F800H),HL
            LD    L,12
            CALL  GOTO_D6
            LD    A,(0F800H)
            LD    H,A
            LD    L,13
            CALL  GOTO_D6
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table2_1)
  FUEL2_1:
            PUSH  HL
            CALL  OPEN1
            CALL  ASCII1
            LD    H,C
            LD    L,11
            CALL  GOTO_D6
            POP   HL
            RET
;************************ END SUB FUEL ***************************

;********************** SUBROUTINE SPEED ************************
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table3)
   SPEED3:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0F800H),HL
            LD    L,20
            CALL  GOTO_D6
            LD    A,(0F800H)
            LD    H,A
            LD    L,21
            CALL  GOTO_D6
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table3_1)
 SPEED3_1:
            PUSH  HL
            CALL  OPEN1
            CALL  ASCII1
            LD    H,C
            LD    L,19
            CALL  GOTO_D6
            POP   HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
;************************ END SUB SPEED ************************

;************************ SUBROUTINE AMP ************************
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table4)
    _AMP4:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0F800H),HL
            LD    L,30
            CALL  GOTO_D8
            LD    A,(0F800H)
            LD    H,A
            LD    L,31
            CALL  GOTO_D8
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table4)
  _AMP4_1:
            PUSH  HL
            CALL  OPEN1
            CALL  ASCII1
            LD    H,B
            LD    L,27
            CALL  GOTO_D3
            LD    H,C
            LD    L,28
            CALL  GOTO_D8
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table5)
     AMP5:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0F800H),HL
            LD    L,30
            CALL  GOTO_D6
            LD    A,(0F800H)
            LD    H,A
            LD    L,31
            CALL  GOTO_D6
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table5_1)
  AMP5_1:
            PUSH  HL
            CALL  OPEN1
```

```
            CALL  ASCII1
            LD    H,B
            LD    L,27
            CALL  GOTO_D6
            LD    H,C
            LD    L,28
            CALL  GOTO_D6
            POP   HL
            RET
;************************** END SUB AMP ************************

;********************** SUBROUTINE VOLTAGE ***********************
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table6)
    VOLT6:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0FB00H),HL
            LD    L,37
            CALL  GOTO_D6
            LD    A,(0FB00H)
            LD    H,A
            LD    L,38
            CALL  GOTO_D6
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table6_1)
  VOLT6_1:
            PUSH  HL
            CALL  OPEN1
            CALL  ASCII1
            LD    H,B
            LD    L,34
            CALL  GOTO_D6
            LD    H,C
            LD    L,35
            CALL  GOTO_D6
            POP   HL
            RET
;************************** END SUB VOLT ************************

;************************ SUBROUTINE RPM ************************
;pass parameter by reg a (data from a/d)
;pass parameter by reg iy (address of table7)
    RPM7:
            PUSH  HL
            CALL  OPEN2
            CALL  ASCII2
            LD    (0FB00H),HL
            LD    L,21
            CALL  GOTO_D14
            LD    A,(0FB00H)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
            LD    H,A
            LD    L,22
            CALL  GOTO_D14
            POP   HL
            RET
;pass parameter by reg a (data from a/d)
;pass parameter by reg hl (address of table7_1)
   RPM7_1:
            PUSH  HL
            CALL  OPEN1
            CALL  ASCII1
            LD    H,B
            LD    L,19
            CALL  GOTO_D14
            LD    H,C
            LD    L,20
            CALL  GOTO_D14
            POP   HL
            RET
;************************ END SUB RPM ***************************

;********************** SUBROUTINE OPENTABLE *********************
;pass parameter by reg a (data)
;pass parameter by reg hl (address)
;get value by reg de
    OPEN1:
            PUSH  HL
            LD    C,A         ; safe data in reg c
            LD    B,00
            ADD   HL,BC       ; add address with data
            LD    D,(HL)      ; data of integer
            POP   HL
            RET
   OPEN2:
            PUSH  IY
            LD    C,A         ; safe data in reg c
            LD    B,00
            ADD   IY,BC       ; add address with data
            LD    E,(IY+0)    ; data of mantissa
            POP   IY
            RET
;********************** END SUB OPENTABLE ************************

;******************* SUBROUTINE CONVERT ASCII ********************
;pass parameter by reg de
;get value by reg bc & reg hl
  ASCII1:
            PUSH  DE
            LD    C,D         ; load integer(2 digit)
            LD    A,ML        ; mark digit low
            AND   C
            LD    C,A
            LD    A,ASCON
            ADD   A,C         ; add to convert ascii
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD    C,A         ; safe ascii in reg c
            LD    B,D         ; load integer(2 digit)
            LD    A,MH        ; mark digit high
            AND   B
            RLA
            RLA
            RLA
            RLA
            RLA
            LD    B,A
            LD    A,ASCON
            ADD   A,B         ; add to convert ascii
            LD    B,A         ; safe ascii in reg b
            POP   DE
            RET
;ascii of integer in reg bc
   ASCII2:
            PUSH  DE
            LD    L,E
            LD    A,ML
            AND   L
            LD    L,A
            LD    A,ASCON
            ADD   A,L
            LD    L,A
            LD    H,E
            LD    A,MH
            AND   H
            RLA
            RLA
            RLA
            RLA
            RLA
            LD    H,A
            LD    A,ASCON
            ADD   A,H
            LD    H,A
            POP   DE
            RET
;ascii of mantissa in reg hl
;*************** END SUBROUTINE CONVERT ASCII ***************

;********************** SUBROUTINE DELAY ************************
   DELAY:
            EX    AF,AF'
            PUSH  BC
            LD    BC,FFFFH
   DELAY1:  DEC   BC
            LD    A,B
            OR    C
            JR    NZ,DELAY1
            POP   BC
            EX    AF,AF'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
            RET

  D_DELAY:  EX    AF,AF'
            PUSH  BC
            LD    BC,0FFFH
    ODELAY: DEC   BC
            LD    A,B
            OR    C
            JR    NZ,ODELAY
            POP   BC
            EX    AF,AF'
            RET
;************************ END SUB DELAY ************************

;****************** SUBROUTINE DISPLAY LINE 06 ******************
;pass parameter by reg h (data)
;pass parameter by reg l (column)
  GOTO_D6:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,06H
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
;************************ END SUB DISP 06 ************************

;****************** SUBROUTINE DISPLAY LINE 08 ******************
;pass parameter by reg h (data)
;pass parameter by reg l (column)
  GOTO_D8:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,08H
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
;************************ END SUB DISP 08 ************************

;****************** SUBROUTINE DISPLAY LINE 14 ******************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;pass parameter by reg h (data)
;pass parameter by reg l (column)
  GOTO_D14:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,14
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
;********************** END SUB DISP 14 ************************

;****************** SUBROUTINE DISPLAY LINE 13 ******************
;pass parameter by reg h (data)
;pass parameter by reg l (column)
 GOTO_D13:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,13
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
;********************** END SUB DIS 13 ************************

;****************** SUBROUTINE DISPLAY LINE 15 ******************
;pass parameter by reg h (data)
;pass parameter by reg l (column)
  GOTO_D15:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,15
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
;************************ END SUB DISP 15 ************************

;******************** SUBROUTINE DISPLAY LINE 18 ********************
;pass parameter by reg h (data)
;pass parameter by reg l (column)
  GOTO_D18:
            EX    AF,AF'
            PUSH  HL
            LD    A,09H
            CALL  SEND
            LD    A,L
            CALL  SEND
            LD    A,18
            CALL  SEND
            LD    A,H
            CALL  SEND
            POP   HL
            EX    AF,AF'
            RET
;************************ END SUB DISP 18 ************************

;********************* SUBROUTINE OFF CURSOR *********************
  OFF_CUR:  EX    AF,AF'
            LD    A,0FH
            CALL  SEND
            EX    AF,AF'
            RET
;************************ END SUB OFF CUR ************************

;********************* SUBROUTINE CLEAR SCREEN *******************
      CLS:  EX    AF,AF'
            LD    A,00H
            CALL  SEND
            EX    AF,AF'
            RET
;************************** END SUB CLS **************************

;************************ SUBROUTINE SENT ************************
;ack in bit 1
;stobe in bit.7
;calling  sequence
;ld   a,data
;call   sent
     SENT:
            LD    B,A         ;stor data
    SENT1:  IN    A,(IO_C1)   ;check  ack=1 ?
            BIT   1,A
            JR    Z,SENT1
            LD    A,LOW
            OUT   (C_81),A    ;sent stobe=0
     ACK1:  IN    A,(IO_C1)   ;check ack=0 ?
            BIT   1,A
            JR    NZ,ACK1
```

```
            LD    A,B         ;restor data
            SET   7,A         ;sent stobe=1
            OUT   (0_B1),A
     ACK2:  IN    A,(10_C1)  ;check ack=1 ?
            BIT   1,A
            JR    Z,ACK2
            RET
;************************ END SUB SENT ************************

;********************* SUBROUTINE NORMAL ***********************
;polling sequence
;ld   a,data
;call normal
   NORMAL:
            EX    AF,AF'      ;stor data
            LD    A,SOT       ;sent start of text
            CALL  SENT
            LD    A, A        ;sent type speech
            CALL  SENT
            EX    AF,AF'      ;restor data
            CALL  SENT
            LD    A,EOT       ;sent end of text
            CALL  SENT
            RET
;************************ END SUB NORMAL ************************

;********************* SUBROUTINE SPEED_1 ***********************
;polling sequence
;ld   a,data
;call  speed_1
  SPEED_1:
            EX    AF,AF'      ;stor data
            LD    A,SOT
            CALL  SENT
            LD    A, B        ;sent start of text
            CALL  SENT
            EX    AF,AF'      ;sent type speech
            CALL  SENT
            LD    A,EOT       ;sent end of text
            CALL  SENT
            RET
;********************** END SUB SPEED_1 ************************

;********************** SUBROUTINE SPEED_2 **********************
;polling  sequence
;ld   e,first data
;ld   d,next data
;call  speed_2
  SPEED_2:
            LD    A,SOT
            CALL  SENT
            LD    A, B
            CALL  SENT
            LD    A,E
            CALL  SENT
            LD    A,D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL  SENT
            LD    A,EOT
            CALL  SENT
            RET
;************************ END SUB SPEED_2 ************************

;************************ SUBROUTINE SEND ************************
;passparameter by reg a
     SEND:
            EX    AF,AF'
            LD    A,HIGH
            OUT   (IO_C2),A  ; STB = 1 DEACTIVATE

    WAITT:  IN    A,(IO_C2)  ; CH = I/P
            BIT   4,A        ; BUSY(1) OR READY(0)
            JR    NZ,WAITT
            EX    AF,AF'     ; RESTORE DATA TO SEND
            OUT   (IO_B2),A  ; SENT TO RX
            LD    A,LOW      ; STB = 0  ACTIVE
            OUT   (IO_C2),A  ; SEND STB
            NOP
            LD    A,HIGH     ; DEACTIVATE STB
            OUT   (IO_C2),A
            RET
;************************ END  SEND ****************************

;======================== DATA PART ============================
     DATA:  DB  09H,00H,02H,0C9H
            DB  0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH
            DB  0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH
            DB  0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH
            DB  0CDH,0CBH,0CDH,0CDH,0CDH,0CDH,(CDH,0CDH,0CBH
            ;## line num 0 (43) ##
            DB  0BAH,09H,08H,03H,0BAH,09H,10H,03H,0BAH
            DB  09H,19H,03H,0BAH,09H,20H,03H,0BAH,09H
            DB  27H,03H,0BAH
            ;## line num 1 (21) ##
            DB  0BAH,09H,09H,04H,0BAH,09H,10H,04H,0BAH
            DB  09H,19H,04H,0BAH,09H,20H,04H,0BAH,09H
            DB  27H,04H,0BAH
            ;## line num 2 ##
            DB  0BAH,09H,0BH,05H,0BAH,09H,10H,05H,0BAH
            DB  09H,19H,0EH,0BAH,09H,20H,05H,0BAH,09H
            DB  27H,05H,0BAH
            ;## line num 3 ##
            DB  0BAH,09H,08H,06H,0BAH,09H,10H,06H,0BAH
            DB  09H,19H,06H,0BAH,09H,20H,06H,0BAH,09H
            DB  27H,06H,0BAH
            ;## line num 4 ##
            DB  0BAH,09H,03H,07H,0BAH,09H,10H,07H,0BAH
            DB  09H,19H,07H,0BAH,09H,20H,07H,0BAH,09H
            DB  27H,07H,0BAH
            ;## line num 5 ##
            DB  0BAH,09H,08H,08H,0BAH,09H,10H,08H,0BAH
            DB  09H,19H,08H,0BAH,09H,20H,08H,0BAH,09H
            DB  27H,08H,0BAH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;## line num 6 ##
DB  0BAH,09H,09H,09H,0BAH,09H,10H,09H,0BAH
DB  09H,15H,09H,0BAH,09H,20H,09H,0BAH,09H
DB  27H,09H,0BAH
;## line num 7 ##
DB  0C8H
DB  0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CAH,0CDH,0CDH
DB  0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CEH,0CDH,0CDH,0CDH,0CDH
DB  0CDH,0CDH,0CDH,0CDH,0CEH,0CDH,0CDH,0CDH,0CDH,0CDH
DB  0CDH,0CAH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0BCH
;## line num 8 (40) ##
DB  09H,10H,09H,0BAH,09H,19H,09H,0BAH
;## line num 9 ##
DB  09H,10H,0AH,0BAH,09H,19H,0AH,0BAH
;## line num 10 ##
DB  09H,10H,0BH,0BAH,09H,19H,0BH,0BAH
;## line num 11 ##
DB  09H,10H,0CH,0BAH,09H,19H,0CH,0BAH
;## line num 12 ##
DB  09H,10H,0DH,0BAH,09H,19H,0DH,0BAH
;## line num 13 ##
DB  09H,10H,0EH,0BAH,09H,19H,0EH,0BAH
;## line num 14 ##
DB  09H,10H,0FH,0BAH,09H,19H,0FH,0BAH
;## line num 15 ##
DB  09H,10H,10H,0BAH,09H,19H,10H,0BAH
;## line num 16 ##
DB  09H,10H,11H,0BAH,09H,19H,11H,0BAH
;## line num 17 ##
DB  09H,10H,12H,0C8H,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH,0CDH
DB  0CDH,0BCH
;## line num 18 ##
;## total write line part = 315 ##
DB  09H,02H,04H,'TEMP',36H,09H,0AH,04H,'FUEL:',09H,12H,04H
DB  'SPEED',3AH,09H,1BH,04H,'AMP',3AH,09H,22H,04H,'VOLT:'
;## char on line 4 (38) ##
DB  09H,03H,06H,'000',09H,11,06H,30H,30H,30H
DB  09H,19,6,'000',09H,26,6,'+00.00',09H,34,6,'00.00'
;## value on line 6 (35) ##
DB  09H,03H,08H,0F9H,'C',09H,12,08H,'%',09H,13H,06H,'km/h'
DB  09H,26,8,'-00.00',09H,36,8,'V'
;## char on line 8 (29) ##
DB  09,19,12,'RPM:'
DB  09,19,14,'0000'
DB  09,19,16,'rpm'
;## rpm part (19) ##
DB  09H,09H,21,'COMPUTER BASED AUTO CAR',0AH,00H
DB  09,19,23,'KMITL',0FH
;# char on line 21 (37) ##

DB  9,3,12,0DAH,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0BFH
DB  0DAH,0C4H,0BFH,9,27,12,0DAH,0C4H,0BFH,0DAH,0C4H
DB  0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0BFH
;## line 12 (30) ##
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
        DB  9,3,16,0C0H,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0D9H
        DB  0C0H,0C4H,0D5H,9,27,16,0C0H,0C4H,0D9H,0C0H,0C4H
        DB  0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0D9H
        ;## line 16 (30) ##
        DB  9,3,13,0B3H,' BEAM:',9,11,13,0B3H,0B3H,9,14,13,0B3H,9,27
        DB  13,0B3H,9,29,13,0B3H,0B3H,9,32,13,':BEAM',9,38,13,0B3H
        ;## line 13 (40) ##
        DB  9,3,15,0B3H,'LIGHT:',9,11,15,0B3H,0B3H,'L',0B3H,9,27
        DB  15,0B3H,'R',0B3H,0B3H,9,32,15,':LIGHT',0B3H
        ;## line 15 (34) ##
        DB  9,3,14,0B3H,9,11,14,0B3H,0B3H,9,14,14,0B3H,9,27,14
        DB  0B3H,9,29,14,0B3H,0B3H,9,33,14,0B3H
        ;## line 14 (26) ##
        DB  9,1,17,0DAH,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0BFH,0DAH,0C4H
        DB  0C4H,0C4H,0C4H,0BFH,9,27,17,0DAH,0C4H,0C4H,0C4H,0C4H
        DB  0BFH,0DAH,0C4H,0C4H,0C4H,0C4H,0BFH
        ;## line 17 (32) ##
        DB  9,1,16,0B3H,'DIAL:',9,8,16,0B3H,0B3H,'BK:',9,14,16,0B3H
        DB  9,27,16,0B3H,9,29,16,':BK',0B3H,0B3H,'HB:',9,38,16,0B3H
        ;## line 16 (40) ##
        DB  9,1,19,0C0H,0C4H,0C4H,0C4H,0C4H,0C4H,0C4H,0D9H,0C0H,0C4H
        DB  0C4H,0C4H,0C4H,0D5H,9,27,19,0C0H,0C4H,0C4H,0C4H,0C4H
        DB  0D9H,0C0H,0C4H,0C4H,0C4H,0C4H,0D9H
        ;## line 19 (32) ##

TABLE1: DB  00H,00H,00H,01H,01H,02H,02H,03H,03H,04H,04H,04H,05H,05H,06H,06H
        DB  07H,07H,07H,08H,08H,08H,09H,09H,09H,10H,10H,10H,11H,11H,12H,12H
        DB  13H,13H,13H,14H,14H,14H,15H,15H,15H,16H,16H,16H,17H,17H,18H,18H
        DB  19H,19H,19H,20H,20H,21H,21H,22H,22H,23H,23H,23H,24H,24H,25H,25H
        DB  26H,26H,26H,27H,27H,27H,28H,28H,28H,29H,29H,29H,30H,30H,31H,31H
        DB  32H,32H,32H,33H,33H,33H,34H,34H,34H,35H,35H,35H,36H,36H,37H,37H
        DB  38H,38H,38H,39H,39H,40H,40H,41H,41H,42H,42H,42H,43H,43H,44H,44H
        DB  45H,45H,45H,46H,46H,46H,47H,47H,47H,48H,48H,48H,49H,49H,50H,50H
        DB  51H,51H,51H,52H,52H,52H,53H,53H,53H,54H,54H,54H,55H,55H,56H,56H
        DB  57H,57H,57H,58H,58H,59H,59H,60H,60H,61H,61H,61H,62H,62H,63H,63H
        DB  64H,64H,64H,65H,65H,65H,66H,66H,66H,67H,67H,67H,68H,68H,69H,69H
        DB  70H,70H,70H,71H,71H,71H,72H,72H,72H,73H,73H,73H,74H,74H,75H,75H
        DB  76H,76H,76H,77H,77H,77H,78H,78H,78H,79H,79H,79H,80H,80H,81H,81H
        DB  82H,82H,82H,83H,83H,84H,84H,85H,85H,86H,86H,86H,87H,87H,88H,88H
        DB  89H,89H,89H,90H,90H,90H,91H,91H,91H,92H,92H,92H,93H,93H,94H,94H
        DB  95H,95H,95H,96H,96H,96H,97H,97H,97H,98H,98H,98H,99H,99H,99H,99H

TABLE2: DB  00H,01H,02H,03H,04H,06H,07H,08H,09H,11H,11H,11H,12H,12H,12H,13H
        DB  13H,13H,14H,14H,14H,15H,15H,15H,15H,15H,16H,17H,17H,18H,18H,18H
        DB  19H,19H,20H,20H,20H,21H,21H,21H,22H,22H,22H,23H,23H,23H,24H,24H
        DB  25H,25H,25H,26H,26H,26H,27H,27H,27H,28H,28H,29H,29H,29H,30H,30H
        DB  30H,31H,31H,31H,31H,31H,32H,32H,32H,32H,33H,33H,34H,34H,35H,35H
        DB  36H,36H,37H,37H,37H,38H,38H,39H,39H,39H,40H,40H,40H,41H,41H,41H
        DB  42H,42H,42H,43H,43H,43H,44H,44H,44H,45H,45H,46H,46H,46H,47H,47H
        DB  48H,48H,48H,49H,49H,50H,50H,50H,51H,51H,51H,52H,52H,52H,52H,53H
        DB  53H,53H,54H,54H,55H,55H,55H,55H,56H,56H,57H,57H,57H,58H,58H,59H
        DB  59H,59H,60H,60H,60H,61H,61H,61H,62H,62H,62H,63H,63H,64H,64H,64H
        DB  65H,65H,65H,66H,66H,67H,67H,67H,68H,68H,68H,69H,69H,70H,70H,70H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
        DB  71H,71H,71H,72H,72H,72H,73H,73H,74H,74H,74H,75H,75H,75H,76H,76H
        DB  76H,77H,77H,77H,78H,78H,78H,79H,79H,79H,80H,80H,81H,81H,81H,82H
        DB  82H,82H,83H,83H,84H,84H,84H,85H,85H,85H,86H,86H,86H,87H,87H,88H
        DB  88H,88H,89H,89H,89H,90H,90H,90H,91H,91H,92H,92H,92H,93H,93H,93H
        DB  94H,94H,95H,95H,95H,96H,96H,96H,97H,97H,98H,98H,98H,99H,99H,00H

TABLE2_1:  DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1

  TABLE3:  DB  00H,00H,01H,02H,02H,03H,04H,04H,05H,06H,06H,07H,08H,09H,09H,10H
           DB  11H,12H,12H,13H,14H,14H,15H,16H,16H,17H,18H,18H,19H,20H,20H,21H
           DB  22H,23H,23H,24H,25H,26H,26H,27H,28H,29H,29H,30H,31H,31H,32H,33H
           DB  33H,34H,35H,35H,36H,37H,38H,38H,39H,40H,40H,41H,42H,43H,43H,44H
           DB  45H,45H,46H,47H,48H,48H,49H,50H,51H,51H,52H,53H,54H,54H,55H,56H
           DB  56H,57H,58H,58H,59H,60H,60H,61H,61H,62H,63H,63H,64H,65H,66H,67H
           DB  67H,68H,69H,69H,70H,71H,72H,72H,73H,74H,74H,75H,76H,77H,78H,78H
           DB  79H,80H,80H,81H,81H,82H,83H,83H,84H,85H,85H,86H,87H,88H,89H,90H
           DB  91H,91H,92H,93H,93H,94H,95H,95H,96H,97H,97H,98H,98H,99H,99H,00H
           DB  00H,01H,02H,03H,03H,04H,05H,06H,07H,08H,08H,09H,10H,10H,11H,12H
           DB  13H,13H,14H,15H,15H,16H,17H,17H,18H,19H,19H,20H,21H,22H,23H,23H
           DB  24H,25H,25H,26H,27H,27H,28H,29H,29H,30H,31H,31H,32H,33H,33H,34H
           DB  34H,35H,36H,37H,38H,38H,39H,40H,41H,42H,42H,43H,44H,44H,45H,45H
           DB  46H,47H,47H,48H,49H,50H,50H,51H,52H,52H,53H,54H,55H,55H,56H,57H
           DB  58H,58H,59H,60H,60H,61H,62H,62H,63H,64H,64H,65H,66H,67H,67H,68H
           DB  69H,70H,70H,71H,72H,72H,73H,74H,74H,75H,76H,76H,77H,76H,79H,80H

TABLE3_1:  DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
           DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
           DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
        DB  01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H,01H

TABLE4: DB  00H,02H,04H,06H,08H,10H,25H,40H,55H,70H,90H,07H,26H,44H,63H,81H
        DB  00H,19H,38H,57H,76H,94H,13H,32H,51H,70H,90H,07H,26H,44H,63H,81H
        DB  00H,18H,36H,53H,71H,90H,67H,24H,42H,60H,80H,90H,20H,40H,60H,80H
        DB  00H,18H,36H,53H,71H,90H,07H,24H,42H,60H,80H,97H,16H,34H,53H,71H
        DB  90H,08H,26H,43H,61H,79H,97H,14H,32H,50H,70H,90H,10H,30H,50H,70H
        DB  90H,08H,76H,43H,61H,79H,97H,14H,32H,50H,70H,87H,06H,24H,43H,61H
        DB  80H,00H,19H,37H,56H,74H,93H,12H,31H,50H,69H,07H,06H,26H,43H,61H
        DB  80H,0CH,19H,37H,56H,74H,93H,12H,31H,50H,67H,84H,01H,19H,35H,93H
        DB  70H,90H,10H,30H,50H,70H,90H,10H,30H,50H,67H,84H,01H,19H,36H,53H
        DB  70H,80H,90H,00H,10H,20H,30H,40H,50H,60H,90H,20H,50H,80H,10H,40H
        DB  70H,89H,08H,27H,46H,64H,83H,02H,21H,40H,57H,74H,95H,09H,26H,43H
        DB  60H,80H,00H,20H,40H,60H,80H,00H,20H,40H,57H,74H,91H,09H,26H,43H
        DB  60H,79H,98H,17H,36H,54H,73H,92H,11H,30H,49H,67H,86H,84H,23H,41H
        DB  60H,79H,98H,17H,31H,54H,73H,92H,11H,30H,45H,67H,86H,04H,23H,41H
        DB  60H,78H,96H,13H,31H,49H,67H,84H,02H,20H,37H,54H,71H,89H,06H,23H
        DB  40H,60H,80H,00H,20H,40H,60H,80H,00H,20H,30H,80H,10H,40H,70H,00H

TABLE4_1: DB  00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,01H,01H,01H,01H,01H
        DB  02H,02H,02H,02H,02H,02H,03H,03H,03H,03H,03H,04H,04H,04H,04H,04H
        DB  05H,05H,05H,05H,05H,05H,06H,06H,06H,06H,06H,07H,07H,07H,07H,07H
        DB  08H,08H,08H,08H,08H,09H,09H,09H,09H,09H,09H,09H,10H,10H,10H,10H
        DB  10H,11H,11H,11H,11H,11H,11H,12H,12H,12H,12H,12H,13H,13H,13H,13H
        DB  13H,14H,14H,14H,14H,14H,14H,15H,15H,15H,15H,15H,16H,16H,16H,16H
        DB  16H,17H,17H,17H,17H,17H,17H,18H,18H,18H,18H,18H,19H,19H,19H,19H
        DB  19H,20H,20H,20H,20H,20H,20H,21H,21H,21H,21H,21H,22H,22H,22H,22H
        DB  22H,22H,23H,23H,23H,23H,23H,24H,24H,24H,24H,24H,25H,25H,25H,25H
        DB  25H,25H,25H,26H,26H,26H,26H,26H,26H,26H,26H,27H,27H,27H,28H,28H
        DB  28H,28H,29H,29H,29H,29H,29H,30H,30H,30H,30H,30H,30H,31H,31H,31H
        DB  31H,31H,32H,32H,32H,32H,32H,33H,33H,33H,33H,33H,33H,34H,34H,34H
        DB  34H,34H,34H,35H,35H,35H,35H,35H,36H,36H,36H,36H,36H,37H,37H,37H
        DB  37H,37H,37H,38H,38H,38H,38H,39H,39H,39H,39H,39H,39H,40H,40H,40H
        DB  40H,40H,40H,41H,41H,41H,41H,41H,42H,42H,42H,42H,42H,42H,43H,43H
        DB  43H,43H,43H,44H,44H,44H,44H,44H,45H,45H,45H,45H,46H,46H,46H,47H

TABLE5: DB  00H,30H,60H,90H,20H,50H,80H,10H,40H,70H,90H,10H,30H,50H,70H,90H
        DB  10H,13H,17H,20H,23H,27H,30H,34H,37H,40H,79H,17H,56H,94H,33H,71H
        DB  10H,34H,59H,83H,08H,32H,57H,81H,06H,30H,43H,56H,69H,81H,94H,07H
        DB  20H,26H,33H,40H,47H,53H,60H,67H,73H,80H,13H,45H,79H,11H,44H,77H
        DB  10H,37H,63H,90H,17H,43H,70H,97H,23H,50H,60H,70H,80H,90H,10H,31H
        DB  20H,46H,71H,97H,28H,53H,79H,99H,24H,50H,55H,67H,78H,84H,93H,01H
        DB  10H,28H,46H,63H,81H,99H,17H,34H,52H,70H,93H,10H,39H,61H,84H,07H
        DB  30H,54H,7EH,03H,28H,52H,76H,01H,26H,50H,65H,75H,39H,01H,14H,27H
        DB  60H,57H,73H,90H,07H,23H,40H,57H,73H,90H,11H,33H,54H,76H,97H,19H
        DB  40H,66H,91H,17H,42H,18H,93H,19H,44H,70H,79F,87H,95H,04H,13H,21H
        DB  30H,49H,68H,87H,06H,24H,43H,62H,81H,00H,20H,40H,60H,80H,00H,20H
        DB  40H,54H,69H,83H,97H,12H,27H,81H,26H,70H,94H,19H,42H,67H,91H,16H
        DB  40H,67H,93H,20H,47H,73H,00H,27H,53H,80H,91H,03H,14H,26H,37H,48H
        DB  60H,84H,09H,33H,58H,82H,06H,31H,56H,80H,91H,03H,14H,86H,37H,49H
        DB  60H,79H,96H,17H,36H,54H,73H,92H,11H,30H,49H,67H,86H,04H,23H,41H
        DB  60H,80H,00H,20H,40H,60H,80H,00H,20H,40H,57H,73H,90H,17H,20H,40H

TABLE5_1: DB  00H,00H,00H,00H,01H,01H,01H,02H,02H,02H,02H,03H,03H,03H,03H,03H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB  04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,04H,05H,05H,05H,06H,06H
        DB  07H,07H,07H,07H,08H,08H,08H,08H,09H,09H,09H,09H,09H,09H,09H,10H
        DB  10H,10H,10H,10H,10H,10H,10H,10H,10H,10H,11H,11H,11H,11H,12H,12H
        DB  13H,13H,13H,13H,14H,14H,14H,14H,15H,15H,15H,15H,15H,15H,16H,16H
        DB  16H,16H,16H,16H,17H,17H,17H,17H,18H,18H,18H,18H,18H,18H,18H,19H
        DB  19H,19H,19H,19H,19H,19H,20H,20H,20H,20H,20H,21H,21H,21H,21H,22H
        DB  22H,22H,22H,23H,23H,23H,23H,24H,24H,24H,24H,24H,24H,25H,23H,25H
        DB  25H,25H,25H,25H,26H,26H,26H,26H,26H,26H,27H,27H,27H,27H,27H,28H
        DB  28H,28H,28H,29H,29H,29H,29H,30H,30H,30H,30H,30H,30H,31H,31H,31H
        DB  31H,31H,31H,31H,32H,32H,32H,32H,32H,33H,33H,33H,33H,33H,34H,34H
        DB  34H,34H,34H,34H,34H,35H,35H,35H,35H,35H,35H,36H,36H,36H,36H,37H
        DB  37H,37H,37H,38H,38H,38H,39H,39H,39H,39H,39H,40H,40H,40H,40H,40H
        DB  40H,40H,41H,41H,41H,41H,42H,42H,42H,42H,42H,43H,43H,43H,43H,43H
        DB  43H,43H,43H,44H,44H,44H,44H,44H,45H,45H,46H,45H,45H,46H,46H,46H
        DB  46H,46H,47H,47H,47H,47H,47H,48H,48H,48H,48H,48H,48H,49H,49H,49H

TABLE6: DB  00H,80H,14H,19H,24H,30H,35H,40H,46H,51H,56H,62H,67H,73H,78H,84H
        DB  89H,95H,01H,07H,12H,18H,24H,30H,35H,41H,47H,52H,58H,63H,68H,75H
        DB  80H,86H,91H,97H,03H,08H,14H,20H,25H,31H,36H,42H,47H,52H,57H,63H
        DB  68H,74H,80H,86H,92H,97H,03H,09H,15H,21H,26H,31H,36H,40H,45H,50H
        DB  55H,61H,67H,73H,79H,86H,92H,98H,04H,10H,15H,21H,27H,32H,36H,42H
        DB  49H,54H,60H,65H,70H,76H,82H,87H,93H,98H,03H,08H,13H,19H,24H,26H
        DB  34H,40H,46H,52H,58H,64H,70H,76H,82H,89H,94H,99H,05H,10H,15H,21H
        DB  27H,32H,38H,43H,48H,54H,59H,64H,70H,75H,81H,87H,93H,99H,05H,11H
        DB  17H,22H,28H,33H,39H,84H,50H,55H,60H,66H,72H,78H,84H,89H,95H,01H
        DB  07H,12H,18H,23H,28H,34H,40H,45H,51H,56H,62H,67H,73H,79H,85H,90H
        DB  96H,01H,07H,12H,17H,23H,28H,33H,38H,44H,49H,55H,60H,66H,71H,77H
        DB  82H,88H,94H,95H,05H,11H,17H,22H,28H,34H,39H,43H,48H,53H,58H,62H
        DB  67H,73H,78H,84H,90H,95H,01H,07H,12H,18H,24H,33H,36H,42H,48H,54H
        DB  60H,66H,71H,77H,83H,88H,94H,00H,05H,11H,16H,22H,27H,32H,38H,43H
        DB  49H,54H,59H,65H,69H,75H,80H,86H,91H,95H,02H,07H,13H,19H,25H,30H
        DB  36H,43H,50H,57H,64H,72H,79H,86H,93H,86H,00H,07H,15H,22H,29H,24H

TABLE6_1: DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DB  0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1
        DB  1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2
        DB  2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3
        DB  3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4
        DB  4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5
        DB  5,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6
        DB  6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7
        DB  7,7,7,7,7,7,7,7,7,7,7,7,7,7,8,8
        DB  8,8,8,8,8,9,8,8,8,8,8,8,8,8,8,8
        DB  8,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9
        DB  09H,09H,09H,09H,10H,10H,10H,10H,10H,10H,10H,10H,10H,10H,10H,10H,10H,10H
        DB  10H,10H,10H,10H,10H,10H,11H,11H,11H,11H,11H,11H,11H,11H,11H,11H,11H,11H
        DB  11H,11H,11H,11H,11H,11H,11H,12H,12H,12H,12H,12H,12H,12H,12H,12H,12H,12H
        DB  12H,12H,12H,12H,12H,12H,12H,12H,12H,12H,13H,13H,13H,13H,13H,13H
        DB  13H,13H,13H,13H,13H,13H,13H,13H,13H,13H,14H,14H,14H,14H,14H,14H

TABLE7: DB  00H,30H,60H,90H,20H,50H,80H,10H,40H,70H,87H,04H,21H,38H,55H,72H
        DB  90H,13H,36H,60H,83H,86H,30H,53H,76H,00H,21H,42H,64H,85H,07H,28H
        DB  50H,73H,96H,20H,43H,66H,90H,13H,36H,60H,85H,11H,37H,62H,88H,14H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB  40H,60H,80H,00H,20H,40H,60H,80H,00H,20H,45H,71H,97H,22H,48H,74H
        DB  00H,23H,46H,70H,93H,16H,40H,63H,86H,10H,27H,44H,61H,78H,95H,12H
        DB  30H,86H,83H,10H,36H,63H,90H,16H,43H,70H,95H,21H,47H,72H,98H,24H
        DB  50H,76H,03H,30H,56H,83H,40H,36H,63H,90H,07H,24H,41H,58H,75H,92H
        DB  10H,33H,56H,80H,03H,26H,50H,73H,96H,20H,41H,62H,84H,05H,27H,48H
        DB  70H,96H,23H,50H,76H,03H,30H,56H,83H,10H,35H,61H,87H,12H,38H,64H
        DB  90H,13H,36H,60H,83H,06H,30H,53H,76H,00H,21H,42H,64H,85H,07H,25H
        DB  50H,73H,96H,20H,43H,66H,90H,13H,36H,60H,85H,11H,37H,62H,88H,14H
        DB  40H,60H,80H,00H,20H,40H,60H,80H,00H,20H,45H,71H,97H,22H,48H,74H
        DB  00H,26H,53H,80H,06H,33H,60H,86H,13H,40H,61H,82H,04H,85H,45H,68H
        DB  90H,13H,36H,60H,83H,06H,30H,53H,76H,00H,17H,34H,51H,68H,85H,02H
        DB  20H,46H,73H,00H,26H,53H,80H,06H,33H,60H,85H,11H,37H,62H,88H,14H
        DB  40H,63H,86H,10H,33H,56H,80H,03H,26H,50H,75H,00H,25H,50H,75H,00H

TABLE7_1:   DB  00H,00H,00H,00H,01H,01H,01H,02H,02H,02H,02H,03H,03H,03H,03H,03H
        DB  03H,04H,04H,04H,04H,05H,05H,05H,06H,06H,06H,06H,06H,06H,07H,07H
        DB  07H,07H,07H,08H,08H,08H,08H,09H,09H,09H,09H,10H,10H,10H,10H,11H
        DB  11H,11H,11H,12H,12H,12H,12H,12H,13H,13H,13H,13H,13H,14H,14H,14H
        DB  15H,15H,15H,15H,15H,16H,16H,16H,16H,17H,17H,17H,17H,17H,17H,18H
        DB  18H,18H,18H,19H,19H,19H,19H,20H,20H,20H,20H,21H,21H,21H,21H,22H
        DB  22H,22H,23H,23H,23H,23H,24H,24H,24H,24H,25H,25H,25H,25H,25H,25H
        DB  26H,26H,26H,26H,27H,27H,27H,27H,27H,28H,28H,28H,28H,29H,29H,29H
        DB  29H,29H,30H,30H,30H,31H,31H,31H,31H,32H,32H,32H,32H,33H,33H,33H
        DB  33H,34H,34H,34H,34H,35H,35H,35H,35H,36H,36H,36H,36H,36H,37H,37H
        DB  37H,37H,37H,38H,38H,38H,38H,39H,39H,39H,39H,40H,40H,40H,40H,41H
        DB  41H,41H,41H,42H,42H,42H,42H,42H,43H,43H,43H,43H,43H,44H,44H,44H
        DB  45H,45H,45H,45H,46H,46H,46H,46H,47H,47H,47H,47H,48H,48H,48H,48H
        DB  48H,49H,49H,49H,49H,50H,50H,50H,50H,51H,51H,51H,51H,51H,51H,52H
        DB  52H,52H,52H,53H,53H,53H,53H,54H,54H,54H,54H,55H,55H,55H,55H,56H
        DB  56H,56H,56H,57H,57H,57H,57H,58H,58H,58H,58H,59H,59H,59H,59H,60H

;========================== END DATA ==========================
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้