ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

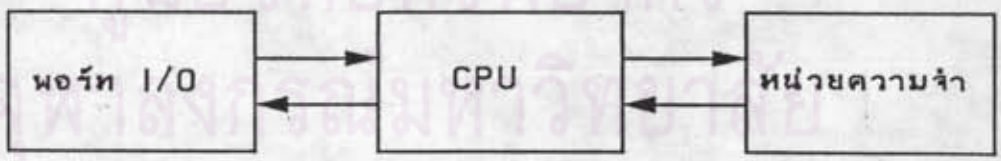
ส่วนที่เกี่ยวกับไมโครคอมพิวเตอร์และการอินเตอร์เฟส

การทดลองในรายงานนี้ ได้มีการอาศัยความสามารถด้านความแม่นยำ ในการประมวลผล, ความจำ, ความรวดเร็ว, การแบ่งเวลาการทำงาน(time sharing) ของคอมพิวเตอร์มาใช้ทั้งในการคำนวณและการรับข้อมูลภายนอก(data acquisition)

4.1 ส่วนประกอบของคอมพิวเตอร์

คอมพิวเตอร์ประกอบด้วยส่วนประกอบใหญ่ๆ 3 ส่วนดังรูป 4-1 คือ

1. หน่วยประมวลผลกลาง(central processing unit, CPU)
2. ส่วนติดต่อกับอินพุท/เอาท์พุท(input/output,I/O)
3. ส่วนความจำ(memory)



รูป 4-1 ส่วนประกอบหลักของคอมพิวเตอร์

ไมโครคอมพิวเตอร์คือคอมพิวเตอร์ขนาดเล็ก ที่นำเอาไมโครโปรเซสเซอร์(microprocessor)มาใช้เป็น CPU

4.2 ไมโครโปรเซสเซอร์

ไมโครโปรเซสเซอร์ เป็นวงจรเบ็ดเสร็จที่รวมเอาส่วนสำคัญของ CPU ไว้ด้วยกัน เช่น

ก. หน่วยพีชคณิตและตรรก(arithmatic and logic unit, ALU) ทำหน้าที่ปฏิบัติการทางด้านพีชคณิตและตรรก

ข. หน่วยควบคุม(control unit) ทำหน้าที่สับหลีกการติดต่อระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำต่างๆ หรือ I/O ต่างๆ ว่าจะผ่านหรือไม่ผ่าน ALU

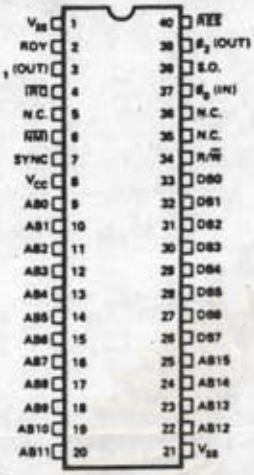
ค. รีจิสเตอร์(register) ซึ่งเป็นหน่วยความจำภายในไมโครโปรเซสเซอร์ เพื่อเก็บข้อมูลที่ไมโครโปรเซสเซอร์จะใช้ดำเนินการ หรือเพื่อผู้ใช้จะใช้ควบคุมการดำเนินการของไมโครโปรเซสเซอร์

ตัวอย่างของไมโครโปรเซสเซอร์ได้แก่ Z80, 8080, 6502 เป็นต้น แต่ในรายงานฉบับนี้จะกล่าวถึงไมโครโปรเซสเซอร์ที่ใช้กับไมโครคอมพิวเตอร์ Apple II ซึ่งเราได้ใช้ทดลองในรายงานฉบับนี้ และเป็นชนิดที่หาได้ง่ายในภาควิชาฟิสิกส์ ไมโครโปรเซสเซอร์ที่กล่าวถึงนี้คือเบอร์ 6502

4.3 ไมโครโปรเซสเซอร์เบอร์ 6502

เป็นไมโครโปรเซสเซอร์ขนาด 8 บิท เมื่อดูรูป 4-2 จะเห็นการจัดขาของ 6502 จะเห็นว่ามีขาส่งข้อมูล 8 ขา คือตั้งแต่ขา 26 ถึง 32 ซึ่งส่งข้อมูลโดยระดับศักดาไฟฟ้าซึ่งมี 2 สถานะคือ สถานะลอจิก 1 (logic 1) คือระดับศักดาไฟฟ้าประมาณ 5 โวลต์ และ สถานะลอจิก 0 (logic 0) คือระดับศักดาไฟฟ้าประมาณ 0 โวลต์ ส่วนข้อมูลจำเพาะของ 6502 แสดงในภาคผนวก ง7

SY6502 — 40 Pin Package

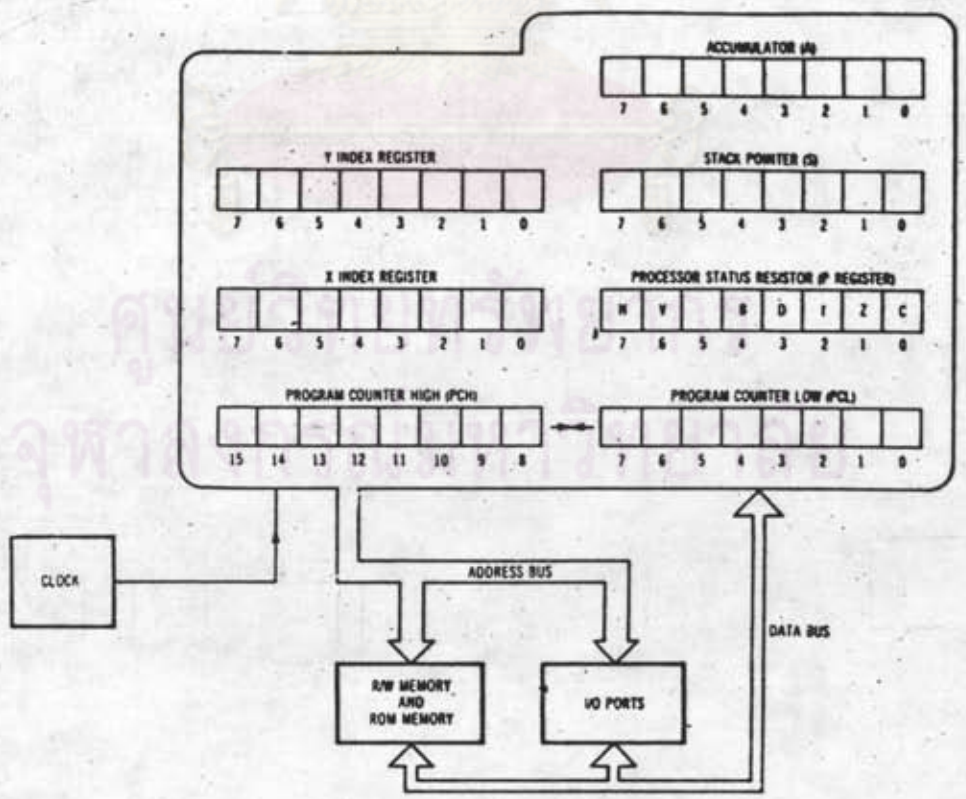


Features

- 65K Addressable Bytes of Memory
- $\overline{IRQ}$ Interrupt
- $\overline{NMI}$ Interrupt
- On-the-chip Clock
  - √ TTL Level Single Phase Input
  - √ Crystal Time Base Input
- SYNC Signal
  (can be used for single instruction execution)
- RDY Signal
  (can be used for single cycle execution)
- Two Phase Output Clock for Timing of Support Chips

รูป 4-2 การจัดขาของวงจรเบ็ดเสร็จ 6502

นั่นคือส่งข้อมูลเป็นเลขฐาน 2 และเป็นเลขฐานสอง 8 หลัก (D0-D7) หรือที่เรียกกันว่า 8 บิทนั่นเอง และมีขาส่งข้อมูลตำแหน่งอีก 16 ขา คือตั้งแต่ขา 9 ถึง 25 (A9-A25) หรือ 16 บิท ซึ่งจะกล่าวเพิ่มเติมในหัวข้อ 4.5 รูป 4-3 แสดงรีจิสเตอร์ภายใน 6502



รูป 4-3 รีจิสเตอร์ภายในไมโครโปรเซสเซอร์ 6502

รีจิสเตอร์ต่าง ๆ ของ6502 มีดังต่อไปนี้ (8)

1. แอคคูมูเลเตอร์(accumulator, A) ใช้สำหรับถ่ายโอนข้อมูลไปยังหน่วยความจำ และเป็นที่บรรจุ 1 ใน 2 ของจำนวนเลขที่ใช้ในทุกปฏิบัติการทางพีชคณิตและตรรก (arithmatic and logical operation)

2. รีจิสเตอร์ X และ Y ทั้งสองอันทำงานคล้ายกัน ใช้ถ่ายโอนข้อมูลเช่นเดียวกับแอคคูมูเลเตอร์ และถูกใช้เป็นดัชนี(index)ขนาด 8 บิท เนื่องจากจำนวนใน X และ Y สามารถเพิ่มค่าขึ้นทีละ 1 (increment) หรือลดค่าลงทีละ 1 (decrement) ดังนั้นจึงสามารถใช้เป็นตัวนับรอบของลูป(loop counter)ได้ด้วย

3. รีจิสเตอร์ P ประกอบด้วยแฟลก(flag)หรือรหัสสถานะ ซึ่งอาจเปลี่ยนแปลงตามการทำงานของแต่ละคำสั่งในโปรแกรม ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อ 4.6

4. รีจิสเตอร์ PC (program counter) เป็นรีจิสเตอร์ 16 บิท ที่คอยบอกตำแหน่งในหน่วยความจำ ซึ่งบรรจุไบท์อันถัดไปของโปรแกรมซึ่งกำลังทำงาน

5. รีจิสเตอร์ S (stack pointer)ใช้เก็บข้อมูลที่ต้องการต่อการทำงานของซับรูทีน(subroutine)ต่างๆ

ซึ่งรีจิสเตอร์สองอันหลังนี้เราไม่ได้ใช้งานโดยตรง

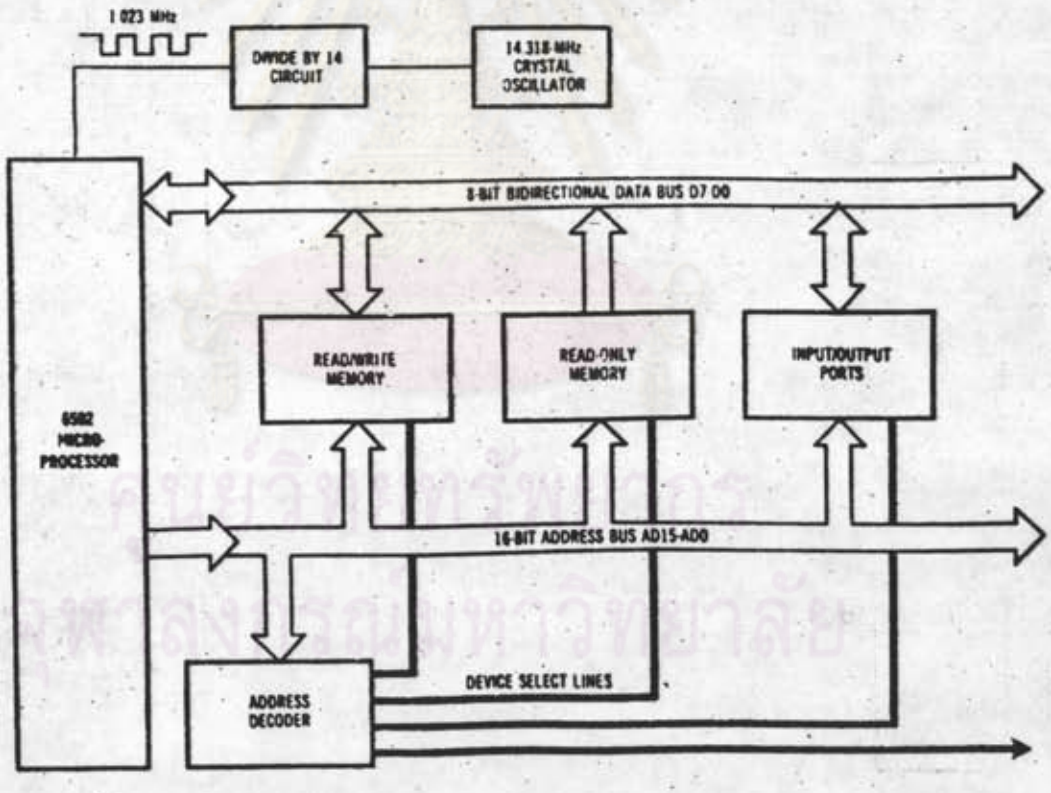
4.4 ส่วนประกอบของไมโครคอมพิวเตอร์

ไมโครคอมพิวเตอร์โดยทั่วไปจะประกอบด้วยส่วนต่าง ๆ ดังต่อไปนี้

1. ไมโครโปรเซสเซอร์ ซึ่งเปรียบเสมือนสมองของไมโครคอมพิวเตอร์

2. วงจรสร้างสัญญาณนาฬิกา ซึ่งควบคุมการทำงานของไมโครโปรเซสเซอร์ ตลอดการดำเนินการ
3. หน่วยการอ่าน/เขียน (read/write memory) ซึ่งได้แก่ แรม (RAM หรือ ramdom access memory)
4. หน่วยความจำรอม (ROM หรือ read-only memory) ซึ่งบรรจุโปรแกรมถาวร เพื่อให้ระบบไมโครคอมพิวเตอร์ทำงานได้
5. อุปกรณ์อินพุท/เอาท์พุท (input/output , I/O device) เช่นคีย์บอร์ด, จอมอนิเตอร์, ดิสค์ไดรว์ เป็นต้น
6. วงจรถอดรหัสแอดเดรส (address decoding circuits)

ซึ่งส่วนต่าง ๆ ของไมโครคอมพิวเตอร์ Apple II ได้แสดงไว้ในรูป 4-4



รูป 4-4 ระบบบัสของไมโครคอมพิวเตอร์ Apple II

4.5 ระบบบัสในไมโครคอมพิวเตอร์

ในไมโครคอมพิวเตอร์ การส่งผ่านข้อมูลส่วนมากจะเกิดขึ้นระหว่างตัวไมโครโปรเซสเซอร์กับหน่วยต่าง ๆ ซึ่งทำการส่งผ่านโดยผ่านกลุ่มของสายส่งที่เรียกว่าบัส (bus) บัสของไมโครคอมพิวเตอร์ทั้งเครื่องเป็นแบบใช้ร่วมกันหมด ระบบบัสของไมโครคอมพิวเตอร์ Apple II แสดงไว้ในรูป 4-4 ซึ่งจะมีบัสต่าง ๆ ที่สำคัญคือ

1. บัสข้อมูล (data bus) เป็นบัสเพื่อส่งข้อมูล มีขนาด 8 บิท คือตั้งแต่ DB0 ถึง DB7
2. บัสแอดเดรส (address bus) เป็นบัสเพื่อส่งหมายเลขตำแหน่งไปเรียกหน่วยต่างๆ มีขนาด 16 บิท คือตั้งแต่ AD0 ถึง AD15
3. บัสควบคุม (control bus) เป็นบัสเพื่อส่งสัญญาณควบคุมต่างๆ ซึ่งมีจำนวนมากมาย

4.6 สัญญาณนาฬิกา

ระบบไมโครคอมพิวเตอร์จะทำงานโดยอ้างอิงสัญญาณนาฬิกาเป็นหลัก ดังนั้นแต่ละขบวนการที่ไมโครคอมพิวเตอร์จะกระทำจะต้องใช้เวลา และต้องทำตามลำดับก่อนหลัง แต่ละขบวนการที่จะดำเนินการใน Apple II จะประกอบด้วยรอบสัญญาณนาฬิกา (clock cycle) สัญญาณนาฬิกาของ Apple II เกิดจากวงจรสร้างสัญญาณนาฬิกาซึ่งใช้วงจรออสซิลเลเตอร์ผลึก(crystal oscillator) ดังแผนผังในรูป 4-4

4.7 พอร์ทอินพุท/พอร์ทเอาท์พุท (I/O ports)(9)

พอร์ท I/O เป็นตำแหน่งของหน่วยความจำซึ่งสามารถโอนย้ายข้อมูลทั้งจากไมโครโปรเซสเซอร์ไปยังอุปกรณ์ภายนอก หรือจากอุปกรณ์ภายนอกไปยัง

ไมโครโปรเซสเซอร์ โดยส่งไปในรูปสถานะลอจิก 1 หรือ 0

เมื่อมีสัญญาณเรียกถูกส่งมาทางบัสแอดเดรส พอร์ท I/O แต่ละตัวจะได้รับสัญญาณพร้อม ๆ กัน แต่ตัวที่มีแอดเดรสตรงตามสัญญาณเรียกที่ส่งมาจึงจะทำงานตามคำสั่ง อุปกรณ์ I/O แต่ละตัวจึงต้องมีการกำหนดหมายเลขประจำตำแหน่งของมันโดยวิธีถอดรหัสแอดเดรส (address decoding)

4.7.1 อินพุทพอร์ท มีเพื่อรับข้อมูลจากภายนอกไปยังคอมพิวเตอร์ ซึ่งถูกควบคุมโดยศักย์ไฟฟ้าที่ขาของวงจรเบ็ดเสร็จซึ่งเรียกว่า ตัวปรับอินเตอร์เฟส (interface adapter) หรือเรียกง่าย ๆ ว่า อินพุทพอร์ท

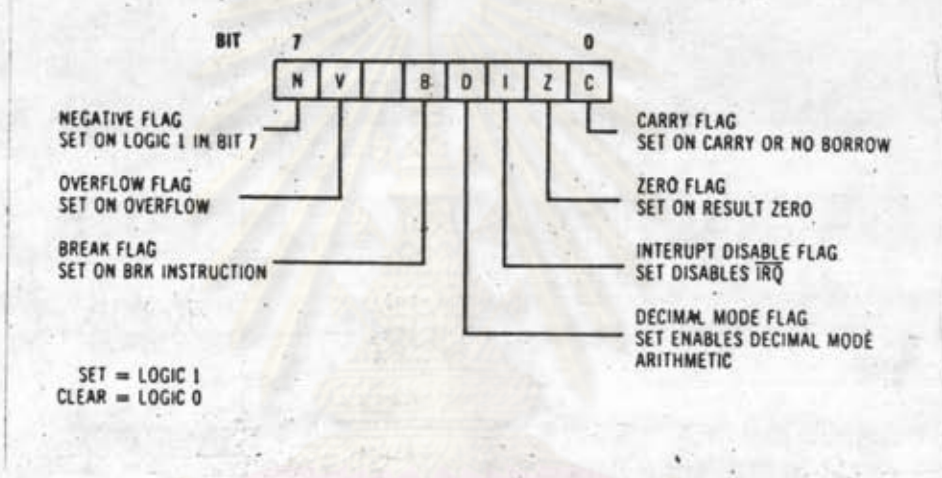
เมื่ออินพุทพอร์ทถูกเรียก จะเกิดพัลส์ควบคุมการเขียน (read pulse หรือ $\overline{RD}$ เครื่องหมายบาร์ข้างบนแสดงว่า จะทำงานเมื่อรับสัญญาณลบ (negative pulse)) อันจะทำให้สัญญาณเลือกอุปกรณ์ (device-select) เป็น $\overline{RD49120}$ เมื่อ 49120 เป็นหมายเลขแอดเดรส ซึ่งจะถูกเกท (gate) โดยการอีนาเบิล (enable) บัฟเฟอร์ 3 สถานะ (3-state buffer) ทำให้อินพุทพอร์ทสามารถตัดขาดตัวเองจากบัสได้เมื่อไม่ถูกเรียก เพื่อจะไม่รบกวนการใช้บัส

4.7.2 เอาท์พุทพอร์ท มีเพื่อส่งข้อมูลจากคอมพิวเตอร์ไปภายนอกโดยเป็นวงจรเบ็ดเสร็จที่ติดต่อกับบัสข้อมูลและบัสแอดเดรส ไมโครโปรเซสเซอร์จะเขียนข้อมูลลงไปในเอาท์พุทพอร์ท และจะควบคุมศักย์ไฟฟ้าบนขาของเอาท์พุทพอร์ท

เมื่อเอาท์พุทพอร์ทถูกเรียก จะเกิดพัลส์ควบคุมการเขียน (write pulse, $\overline{WR}$) เกิดขึ้นประมาณ 500 นาโนวินาที อันจะทำให้สัญญาณเลือกอุปกรณ์ เป็น $\overline{WR43255}$ (เมื่อ 43255 เป็นหมายเลขแอดเดรส) เพื่อเกทข้อมูลจากบัสข้อมูลไปยังเอาท์พุท แต่เนื่องจากช่วงของพัลส์ WR น้อยเกินกว่าที่จะให้อุปกรณ์ที่รับข้อมูลทำหน้าที่อะไรได้มากนัก ดังนั้นเอาท์พุทพอร์ทจะต้องมีวงจรที่จะช่วยกักข้อมูล (latch) ไว้จนกว่าจะมีการ update หรือจนกว่าจะดับไฟ

4.8 แฟลก (flag)

สัญญาณหนึ่งที่สำคัญมากสำหรับระบบไมโครคอมพิวเตอร์คือสัญญาณแฟลก ซึ่งเป็นสัญญาณคอยบอกสถานะของอุปกรณ์ต่างๆ จึงอาจเรียกว่าแฟลกสถานะ (status flag) ซึ่งจะต้องมีซอฟท์แวร์คอยตรวจสอบแฟลก รีจิสเตอร์สำคัญที่เกี่ยวข้องกับแฟลกของ Apple II คือรีจิสเตอร์ P ใน 6502 (รูป 4-2) บิททั้ง 8 ของรีจิสเตอร์ P ได้แสดงไว้ในรูป 4-5



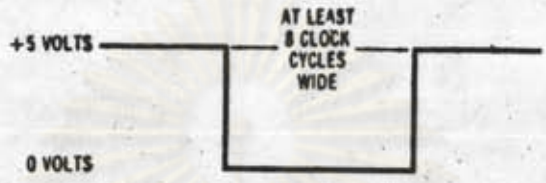
รูป 4-5 บิทต่าง ๆ ของรีจิสเตอร์ P ของ 6502

แฟลกอาจนำไปใช้ประโยชน์เพื่อบอกไมโครโปรเซสเซอร์ว่าอุปกรณ์ต่าง ๆ พร้อมที่จะติดต่อกับไมโครโปรเซสเซอร์หรือไม่ เราจะทราบบทบาทของแฟลกมากขึ้นในหัวข้อต่อๆไป

4.9 การอินเทอร์รัพท์ (interrupt)

ไมโครคอมพิวเตอร์ Apple II มีความสามารถในการรับสัญญาณจากภายนอก โดยรับในลักษณะของการขัดจังหวะหรืออินเทอร์รัพท์(interrupt)

ซึ่งจะขัดจังหวะปฏิบัติการปกติของไมโครโปรเซสเซอร์ และเริ่มปฏิบัติภารกิจอีกอันหนึ่งแทน 6502 มีความสามารถในการถูกขัดจังหวะได้ 2 แบบ แต่ที่เราใช้คือ การอินเทอร์รัพท์แบบ IRQ (interrupt request) ซึ่งจะเกิดขึ้นเมื่อวงจรภายนอกส่งศักย์ไฟฟ้าสถานะลอจิก 0 ให้กับขา $\overline{\text{IRQ}}$ ของ 6502 เป็นช่วงเวลาอย่างน้อย 8 รอบสัญญาณนาฬิกา ตามรูปที่ 4-6



รูป 4-6 สัญญาณซึ่งให้กับขา $\overline{\text{IRQ}}$ ของ 6502 เพื่อขอการอินเทอร์รัพท์

ถ้าหากแฟลก I ของรีจิสเตอร์ P ถูกเซ็ท ไมโครโปรเซสเซอร์ก็จะไม่รับการอินเทอร์รัพท์ แต่หากแฟลก I ของรีจิสเตอร์ P ถูกเคลียร์ ไมโครโปรเซสเซอร์ก็จะรับการอินเทอร์รัพท์ และการเซ็ทแฟลก I จะป้องกันไม่ให้มีการอินเทอร์รัพท์ผ่านเข้าทางขา $\overline{\text{IRQ}}$ ของ 6502 เนื่องจากรีจิสเตอร์ P ถูกเคลียร์ระหว่างคำสั่ง RTI (return from interrupt subroutine, ออกจากซับรูทีนอินเทอร์รัพท์) แฟลก I จะถูกเคลียร์หลังจากที่คำสั่ง RTI ถูกกระทำไปแล้ว แฟลก I สามารถถูกเซ็ทได้ด้วยคำสั่ง SEI(set interrupt flag) และแฟลก I สามารถถูกเคลียร์ได้ด้วยคำสั่ง CLI (clear interrupt flag)

ก่อนจะใช้ซับรูทีนที่เกี่ยวข้องกับการอินเทอร์รัพท์แบบ IRQ (ซึ่งเป็นซับรูทีนภาษาเครื่องของ 6502) ผู้เขียนโปรแกรมจะต้องใส่แอดเดรสไบท์ต่ำและแอดเดรสไบท์สูงของแอดเดรสเริ่มต้นของซับรูทีนลงในตำแหน่ง $03FE และ $03FF ตามลำดับเสียก่อน เพราะการอินเทอร์รัพท์แบบ IRQ นี้ ไมโครคอมพิวเตอร์ Apple II จะค้นหาแอดเดรสเริ่มต้นของซับรูทีนจากตำแหน่ง $03FE และ $03FF เสมอ

ในกรณีของการอินเทอร์รัพท์แบบ IRQ ไมโครคอมพิวเตอร์ Apple IIc

จะมีโปรแกรมมอนิเตอร์ เพื่อบันทึกค่าในแอคคูมูเลเตอร์ ไว้ที่ตำแหน่ง $45 เสมอ ดังนั้นทุกโปรแกรมที่ใช้การอินเทอร์รัพท์แบบ IRQ นี้ ผู้เขียนโปรแกรมจำต้องเรียกค่าของแอคคูมูเลเตอร์จากตำแหน่งนี้เสมอ เมื่อสิ้นสุดซับรูทีนของการอินเทอร์รัพท์

4.10 <u>การอินเทอร์เฟส (interfacing)</u>

การอินเทอร์เฟสคือการเชื่อมต่อคอมพิวเตอร์เข้ากับอุปกรณ์ภายนอก ซึ่งสามารถทำได้โดย

1. ต่ออุปกรณ์ภายนอกที่ต้องการเข้ากับวงจรอินเทอร์เฟสที่เหมาะสม

2. วงจรอินเทอร์เฟสที่เหมาะสมจะต้องสามารถรับสัญญาณควบคุม, สัญญานนาฬิกา และระบบบัส แบบเดียวกับที่ไมโครคอมพิวเตอร์ชนิดนั้น ๆ ใช้อยู่ได้อย่างถูกต้อง

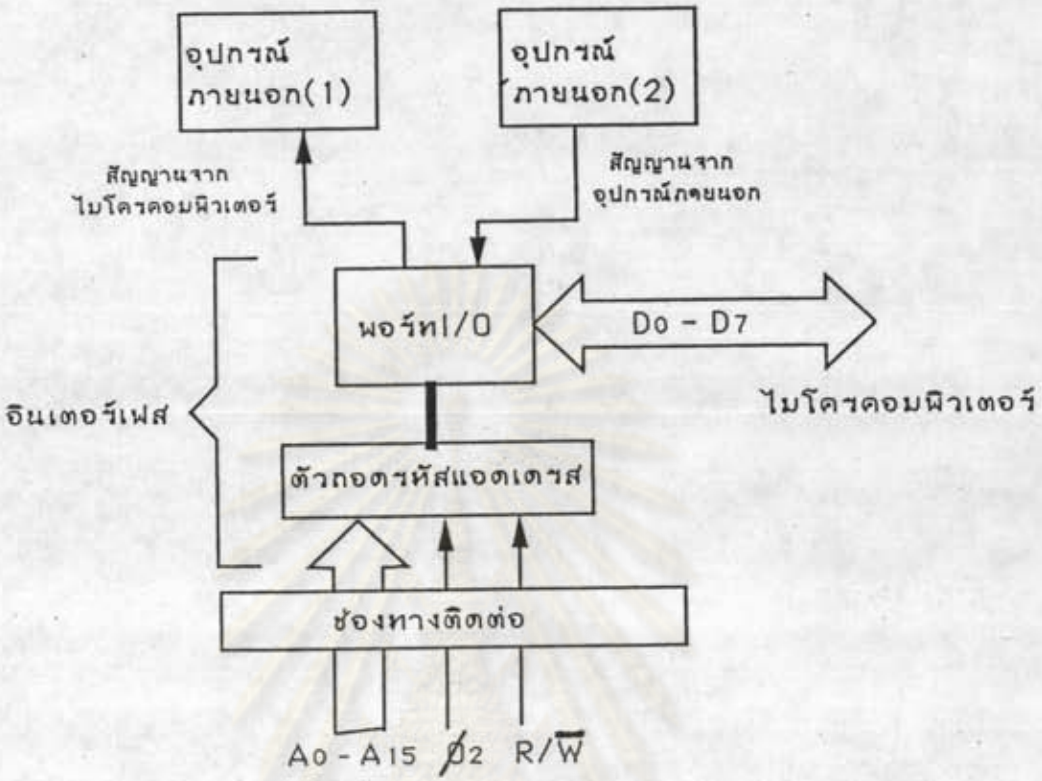
3. ต่อวงจรอินเทอร์เฟสดังกล่าวเข้ากับช่องทางติดต่อที่สะดวก

4. ไมโครคอมพิวเตอร์จะส่งผ่าน และรับข้อมูลกับอุปกรณ์ภายนอกผ่านวงจรอินเทอร์เฟสโดยอาศัยระบบบัส

5. ไมโครคอมพิวเตอร์สามารถเลือกหรือเรียกอุปกรณ์ภายนอกโดยสัญญาณเลือกแอดเดรสของพอร์ทที่อุปกรณ์ภายนอกได้ต่อไว้ และสัญญาณเลือกอุปกรณ์ ดังที่ได้อธิบายในหัวข้อ 4.7

แผนผังของการอินเทอร์เฟสแสดงไว้ในรูปที่ 4-7

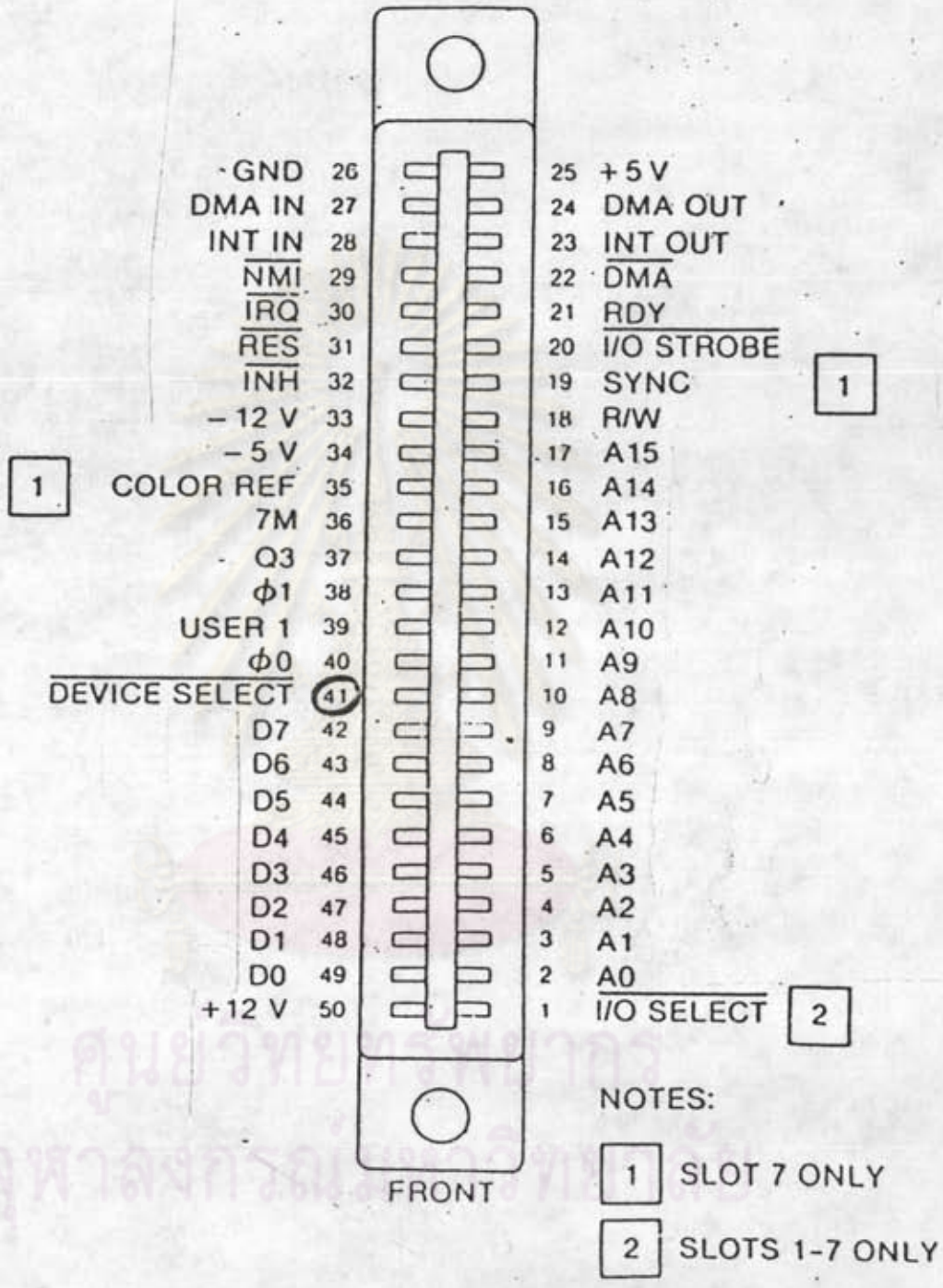
I10304587



รูป 4-7 แผนผังการอินเทอร์เฟสกับไมโครคอมพิวเตอร์ Apple II

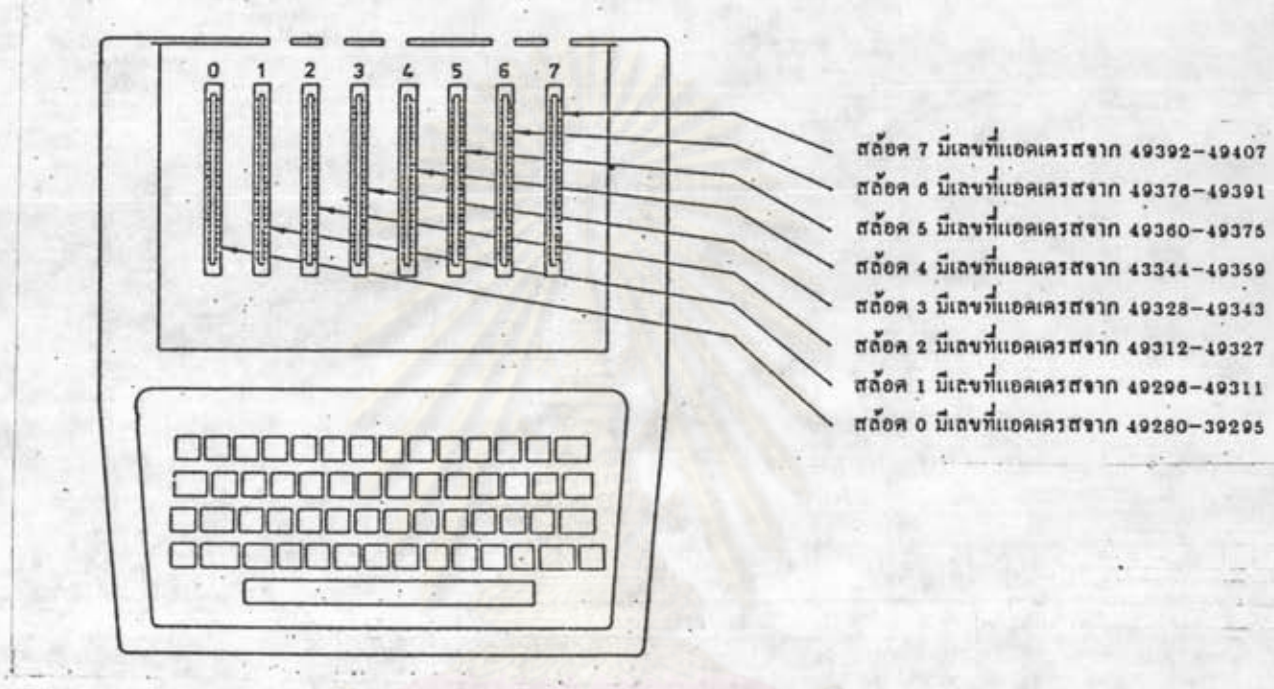
4.11 การอินเทอร์เฟสใน Apple II

ไมโครคอมพิวเตอร์ Apple II มีช่องทางติดต่อกับอุปกรณ์ภายนอกหลายแห่ง เช่น สล็อตอินเตอร์เฟส (peripheral interface slots) ซึ่งมีทั้งหมด 8 สล็อต รูป 4-8 แสดงช่องทางต่อ (connector) ต่าง ๆ ของสล็อตหนึ่งตัว



รูป 4-8 ช่องทางติดต่อกับอุปกรณ์ภายนอกของ Apple II (สล็อตอินเตอร์เฟส)

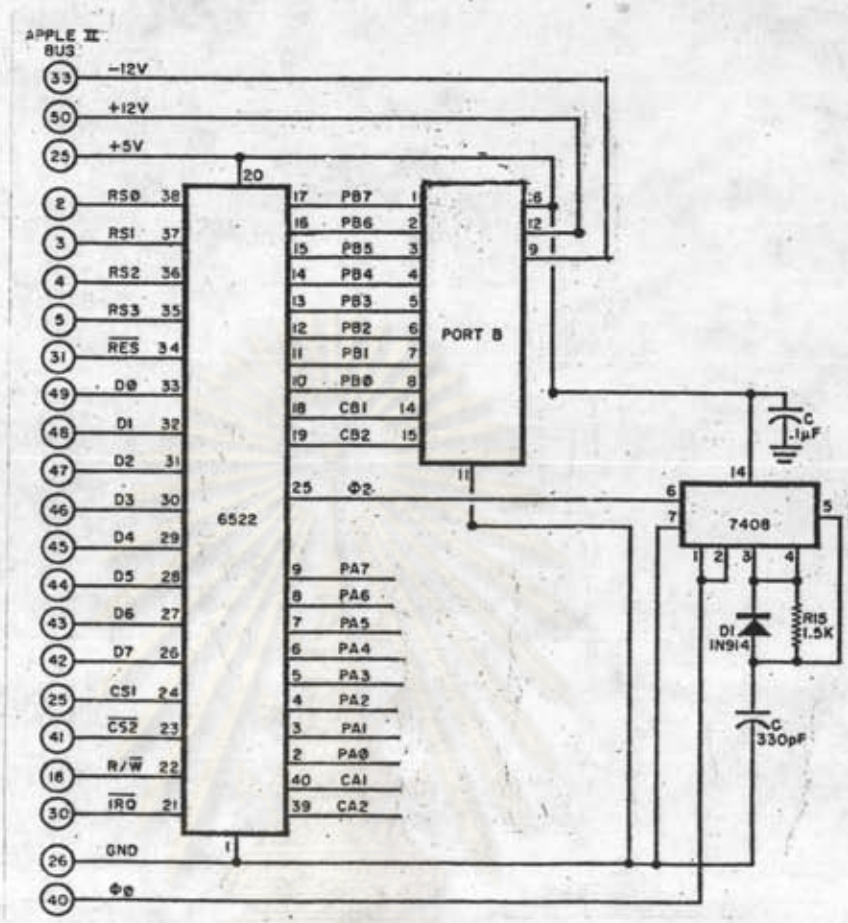
และรูป 4-9 แสดงหมายเลขแอดเดรสของสล็อตตั้งแต่หมายเลข 0 ถึงหมายเลข 7 เนื่องจากสล็อตแต่ละตัวมีการเข้ารหัสแอดเดรสไว้เป็นช่วง ดังนั้นสามารถมีแอดเดรสบนสล็อตแต่ละตัวได้ถึง 16 แอดเดรสย่อย ซึ่งทำได้โดยอาศัยรอยต่อ A0 ถึง A3 ของสล็อต



รูป 4-9 หมายเลขแอดเดรสของสล็อตทั้ง 8 ของ Apple II

4.12 วงจรอินเทอร์เฟสที่นำมาใช้

วงจรอินเทอร์เฟสที่นำมาใช้ในการทดลองครั้งนี้ แสดงในรูป 4-10 และแสดงลายวงจรในภาคผนวก ข และรายการอุปกรณ์ที่ใช้ในภาคผนวก ค วงจรอินเตอร์เฟสอันนี้มีอุปกรณ์อิเลกทรอนิกส์ที่สำคัญคือ 6522 VIA (versatile interface adapter) ข้อมูลจำเพาะของ 6522 แสดงในภาคผนวก ง8



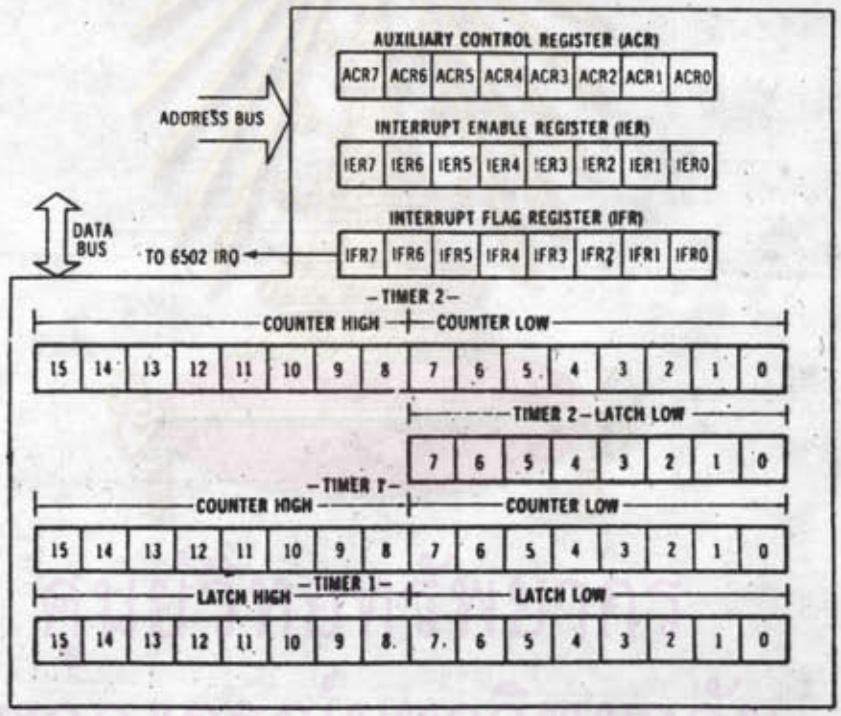
รูป 4-10 วงจรอินเทอร์เฟส

4.13 6522 VIA(10)

6522 เป็นวงจรเบ็ดเสร็จสเกลใหญ่ (large scale, LSI) ในตระกูลเดียวกันกับไมโครโปรเซสเซอร์ 6502 ซึ่งสามารถทำการอินพุท/เอาท์พุท และอินเตอร์เฟสได้อย่างสะดวกกับระบบไมโครคอมพิวเตอร์ที่ใช้ 6502 เป็นพื้นฐาน เนื่องจากใช้บัสและสัญญาณควบคุมแบบเดียวกันกับ ที่ใช้กับ 6502 ได้แก่ R/W, และสัญญาณ DEVSEL เป็นต้น และรับสัญญาณนาฬิกาของ 6502 (สัญญาณนาฬิกา ϕ_2) ได้ ข้อมูลจำเพาะของ 6522 แสดงไว้ในภาคผนวก ง-6 แอดเดรสต่างๆที่เกี่ยวข้องกับแผ่นวงจรอินเทอร์เฟส จะถูกอ้างอิง

ที่สล็อตหมายเลข 7 เนื่องจากเราจะใช้สล็อตนี้เพียงสล็อตเดียวเท่านั้น

6522 มีตัวจับเวลา (interval timer) หนึ่งตัว เรียกว่า T1 และมีตัวนับ/จับเวลา(counter/timer)อีกหนึ่งตัวเรียกว่า T2 ทั้ง T1 และ T2 ต่างก็เป็นรีจิสเตอร์ขนาด 16 บิท บิทต่าง ๆ ของ T1 และ T2 ได้ถูกแสดงไว้ในรูป 4-11 จำนวนใดๆที่ใส่ไว้ในรีจิสเตอร์นี้จะถูกลดค่าลงทีละหนึ่งทั้งโดยสัญญาณนาฬิกาของระบบ หรือโดยสัญญาณจากภายนอก จนกระทั่งถูกลดค่าผ่านศูนย์ และตอนนี้เองแฟลกในรีจิสเตอร์อีกตัวหนึ่งซึ่งเรียกว่า IFR (interrupt flag register) จะถูกเซ็ท หรือทำให้เกิดสัญญาณขอการอินเทอร์รัพท์ (interrupt request, IRQ)



รูป 4-11 รีจิสเตอร์ที่เกี่ยวกับการนับและจับเวลาภายใน 6522

รูป 4-11 ยังได้แสดงรีจิสเตอร์อื่นๆ ซึ่งเกี่ยวข้องกับการนับและการจับเวลา และยังแสดงรายละเอียดเพิ่มเติมในรูป 4-12 และ 4-13 ส่วนหน้าที่ของ T1 และ T2 ได้แสดงไว้ในตาราง 4-1 ส่วนแอดเดรสของรีจิสเตอร์ซึ่งเกี่ยวข้องกับการนับและการจับเวลา ได้แสดงไว้ในตาราง 4-2

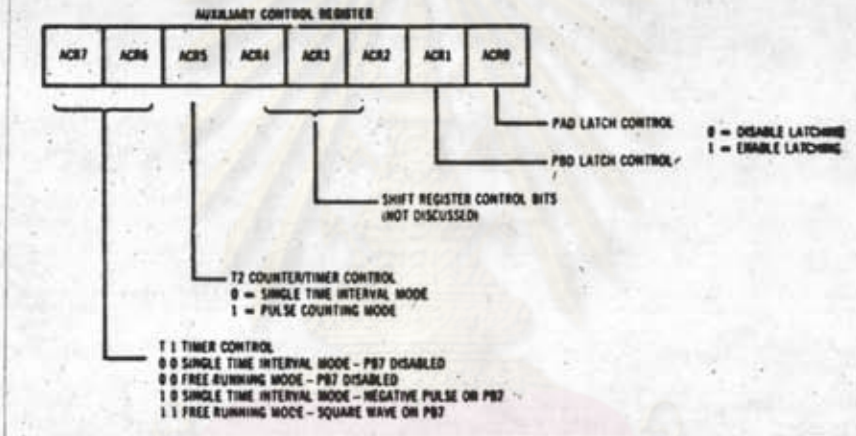
แอดเดรส	สัญลักษณ์	ปฏิบัติการ	หน้าที่
$C0F4	T1LL	WRITE T1CL	เขียนจำนวนเลข 8 บิทลงในไบท์ตำแหน่งต่ำของแลทช์ของ T1
$C0F4	T1CL	READ T1CL	อ่านข้อมูลที่บรรจุในไบท์ตำแหน่งต่ำของเคาน์เตอร์และเคลียร์อินเทอร์รัพท์แฟลก, IFR6
$C0F5	T1LH	WRITE T1LH	เขียนจำนวนเลข 8 บิทลงในไบท์ตำแหน่งสูงของแลทช์ของ T1 , ถ่ายโอนข้อมูลซึ่งบรรจุในแลทช์ทั้งสองของ T1 ไปยังเคาน์เตอร์ทั้งสองของ T1, เคลียร์ IFR 6 และเริ่มกระบวนการนับลง
$C0F5	T1CH	READ T1CH	อ่านข้อมูลที่บรรจุในไบท์ตำแหน่งสูงของเคาน์เตอร์ของ T1
$C0F8	T2LL	WRITE T2LL	เขียนจำนวนเลข 8 บิทลงในไบท์ตำแหน่งต่ำของ T2
$C0F8	T2CL	READ T2CL	อ่านข้อมูลที่บรรจุในไบท์ตำแหน่งต่ำของเคาน์เตอร์ของ T2 และเคลียร์ IFR5
$C0F9	T2CH	WRITE T2CH	เขียนจำนวนเลข 8 บิทลงในไบท์ตำแหน่งสูงของเคาน์เตอร์ของ T2, ถ่ายโอนข้อมูลที่บรรจุ ในไบท์ตำแหน่งต่ำของแลทช์ของ T2 ลงในไบท์ตำแหน่งต่ำของเคาน์เตอร์ของ T2, เคลียร์ IFR 5 และเริ่มกระบวนการนับลง
$C0F9	T2CH	READ T2CH	อ่านข้อมูลที่บรรจุในไบท์ตำแหน่งสูงของเคาน์เตอร์ของ T2

หมายเหตุ : ถ้าเป็นสัญญาณเลือกแบบ I/O select จะต้องบวกแอดเดรสใน ตาราง 4-1 ด้วย $610

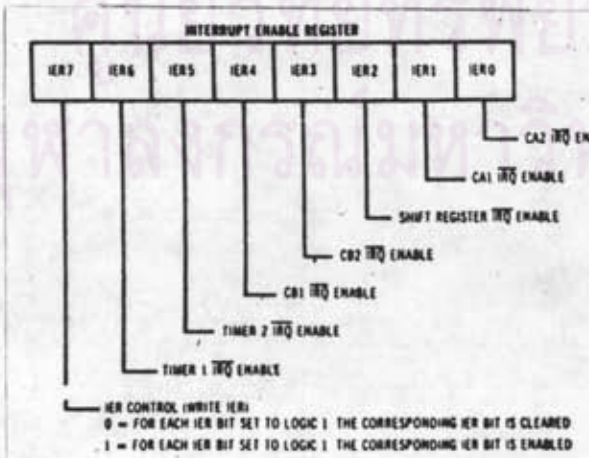
ตาราง 4-1 การกำหนดตำแหน่งและหน้าที่ของรีจิสเตอร์ T1 และ T2 ของ6522 เมื่อใช้บอร์ด 6522 บนสล็อตที่ 7 และใช้สัญญาณเลือกแบบ device select

แอดเดรส	สัญลักษณ์	ชื่อ/ความหมาย
$C0FB	ACR	Auxilliary control register
$C0FD	IFR	Interrupt flag register
$C0FF	IER	Interrupt enable register

ตาราง 4-2 ตำแหน่งและสัญลักษณ์ของรีจิสเตอร์ที่เกี่ยวข้องกับการนับ/จับเวลาเมื่อใช้บอร์ด 6522 บนสล็อตที่ 7 และใช้สัญญาณเลือกแบบ device select



รูป 4-12 ACR รีจิสเตอร์



รูป 4-13 IER รีจิสเตอร์

4.14 การใช้ T1 ของ 6522(8)

เราจะใช้ T1 ได้โดยการโหลด (load) เลข 8 บิท 2 จำนวนลงใน T1 โดยโหลดไบท์ตำแหน่งต่ำ (low order byte) ลงในแลทช์ตำแหน่งต่ำ (latch low) ของ T1 ซึ่งเรียกย่อว่า T1LL และโหลดไบท์ตำแหน่งสูง (high order byte) ลงในแลทช์ตำแหน่งสูง (latch high) ซึ่งเรียกย่อ ๆ ว่า T1LH

ก่อนจะใช้ T1 จะต้องตั้งค่าในรีจิสเตอร์ ACR (auxilliary control register, ดูรูป 4-12) เพื่อกำหนดโหมด (mode) ของ T1 เช่น single time interval mode (หรือ one-shot mode) และ free-running mode เป็นต้น แต่ในรายงานฉบับนี้ จะกล่าวถึงโหมดที่เรานำมาใช้คือ free-running mode

ใน free-running mode นี้ T1 จะทำงานอย่างต่อเนื่องโดยที่แลทช์ของมัน จะไม่ต้องถูกโหลดค่าใหม่ทุกครั้ง ในโหมดนี้ T1 สามารถผลิตสัญญาณอินเทอร์รัพท์ได้อย่างต่อเนื่องโดยมีความกว้างเท่าๆกัน ซึ่งจะเกี่ยวข้องกับรีจิสเตอร์ IER (interrupt enable register, รูปที่ 4-13) เมื่อขา $\overline{\text{IRQ}}$ ของ 6522 ต่อกับขา $\overline{\text{IRQ}}$ บนไมโครโปรเซสเซอร์ 6502 โดยผ่านสล็อต 6522 ก็จะสามารถผลิตสัญญาณขอการอินเทอร์รัพท์ (IRQ) เมื่อแฟลกใดๆใน IFR ถูกเซ็ท และถ้าหาก IER ได้ถูกอินิเชียลไลซ์ไว้อย่างเหมาะสมแล้ว เมื่อนั้นก็จะเกิดการอินเทอร์รัพท์ชนิด IRQ เกิดขึ้น ทั้งนี้เนื่องจากทุกบิทของ IER จะสมนัยกับบิทของรีจิสเตอร์ IFR บิทต่อบิท นอกจากบิทที่ 8 (most significant bit, MSB)เท่านั้น ถ้าหากบิทใดใน IER ถูกอีนาเบิ้ลแฟลกที่สมนัยกันก็จะถูกเซ็ทและขา $\overline{\text{IRQ}}$ จะมีลอจิก 0 และขอการอินเทอร์รัพท์

เมื่อ T1 อยู่ในโหมดนี้ การอินเทอร์รัพท์อย่างต่อเนื่องอาจเกิดขึ้นได้ และภารกิจใด ๆ ซึ่งไมโครคอมพิวเตอร์จะต้องกระทำในแบบปรกติสามารถทำได้ในโปรแกรมส่วนของการอินเทอร์รัพท์ (interrupt routine) แต่ก่อนอื่นต้องโหลดตำแหน่งที่ $30FE และ $30FE ด้วยแอดเดรสเริ่มต้นของอินเทอร์รัพท์รูทีน

(ซึ่งเป็นภาษาเครื่อง) เสียก่อน

ในโหมดนี้ จำนวนที่บรรจุอยู่ในแลทช์จะถูกย้ายไปยังเคาน์เตอร์ (counter) ของมันคือ T1CL และ T1CH โดยอัตโนมัติ เมื่อสิ้นสุดช่วงของการนับ

4.15 <u>การใช้ T2 นับสัญญาณจากภายนอก</u>(8)

เริ่มต้นด้วยการตั้ง ACR ให้อยู่ในโหมดนับของ T2 (pulse counting mode) ในโหมดนี้ T2 จะนับสัญญาณจากภายนอกที่เข้ามาเป็นขบวน (pulse train) ซึ่งจะส่งเข้ามาทางขา PB6 ของ 6522 สิ่งสำคัญที่ควรทราบเพิ่มเติมคือ

1. 6522 นับสัญญาณลอจิค 0 หรือสัญญาณลบ(negative pulse)

2. ความถี่สูงสุดของสัญญาณที่สามารถนับได้คือ ประมาณครึ่งหนึ่งของความถี่สัญญาณนาฬิกา คือประมาณ 500 KHz

3. แฟลกซึ่งจะส่งสัญญาณบอกการสิ้นสุดของการนับ จะถูกเซ็ทภายหลังการนับลง(count down) ได้ผ่านศูนย์ไปแล้ว ซึ่งหมายถึงว่าถ้า N เป็นจำนวน 16 บิทใด ๆ ซึ่งถูกโหลดให้กับ T2 ก็จะมีสัญญาณจำนวน N+1 สัญญาณเกิดขึ้น ก่อนที่แฟลก IFR5 จะถูกเซ็ท

4.16 <u>โปรแกรมที่ใช้</u>

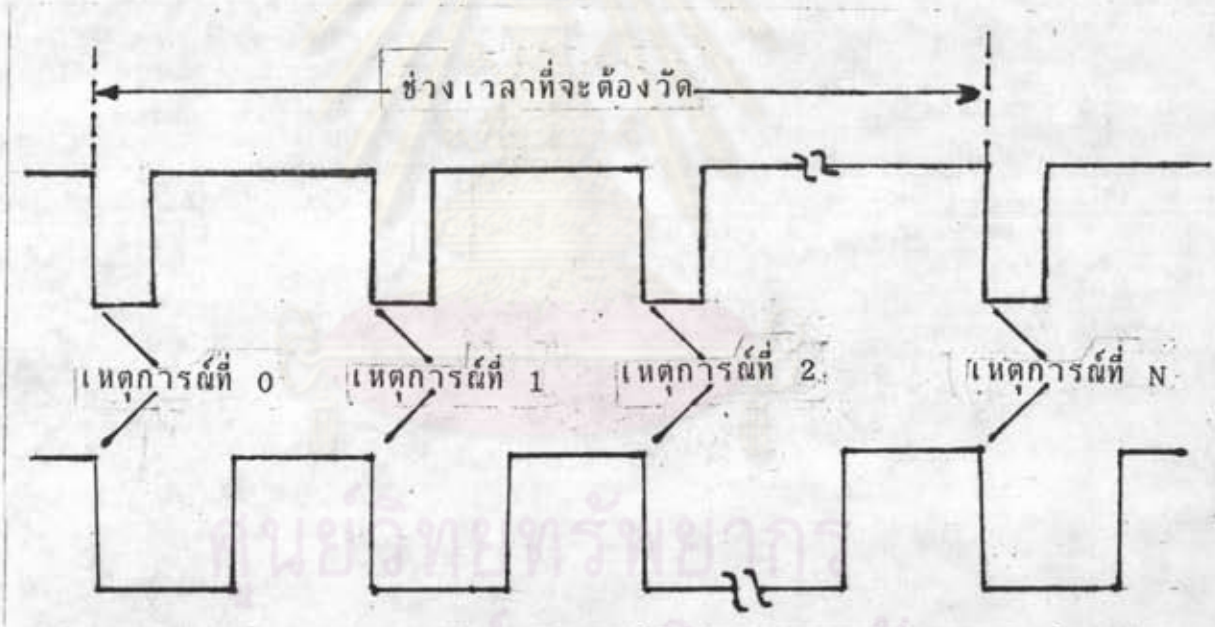
ในการวิจัยครั้งนี้ เราได้ใช้ภาษาโปรแกรม Applesoft BASIC เป็นหลักเพื่อความสะดวก ยกเว้นส่วนที่เกี่ยวข้องกับการตรวจจับสัญญาณจากภายนอก(จากมิกเซอร์ของออสซิลเลเตอร์สองตัว) ซึ่งส่วนนี้เราใช้ภาษาเครื่องของไมโครโปรเซสเซอร์ 6502 และได้แสดงชุดคำสั่งของภาษาเครื่องนี้ไว้ในภาคผนวก ฉ และเขียนโปรแกรมในภาษาเบสิค ให้เรียกซับรูทีนในภาษาเครื่อง

นี้ได้ โปรแกรมทั้งหมดจะอธิบายในหัวข้อต่อไป รายละเอียดเกี่ยวกับภาษาโปรแกรมจะละไว้ในฐานที่เข้าใจ แต่ได้เขียนสรุปไว้ในภาคผนวก ช

4.17 ซับรูทีน TIMER

เป็นซับรูทีนซึ่งเราได้เขียนด้วยภาษาเครื่อง เพื่อใช้นับจำนวนพัลส์ ซึ่งได้ใช้ทั้ง T2 ในโหมดการนับ และ T1 ใน free-running mode ควบคู่กัน โปรแกรมนี้ใช้จับเวลาซึ่งต้องใช้ในการเกิด N เหตุการณ์ (events หรือรอบ (cycle)) ความแม่นยำของการวัดจะประมาณ ±50 ไมโครวินาที

เนื่องจาก T2 นับสัญญาณลบ เหตุการณ์ที่จะถูกจับเวลาจะต้องให้สัญญาณลบได้เช่น สัญญาณตามรูป 4-14



รูป 4-14 แผนผังเวลาสำหรับโปรแกรม TIMER

สัญญาณเหล่านี้จะถูกส่งเข้าทางขา PB6 ของ 6522 และความกว้างของสัญญาณต้องกว้างอย่างน้อย 500 nsec และจำนวนของเหตุการณ์ , N จะต้องไม่เกิน 65,536 ($FFFF) และสัญญาณไม่จำเป็นต้องมีความกว้างสม่ำเสมอ ตามรูป 4-14 สัญญาณหมายเลข 0 (zeroth pulse) เป็นจุดเริ่มต้นของขบวนการนับ และสัญญาณหมายเลข 1 (first pulse) จะเป็นจุดสิ้นสุดของขบวนการจับ

เวลา ในกรณีนี้ N=1 ในโปรแกรมที่เราใช้นั้น N เป็นพารามิเตอร์ที่ถูกใส่เข้าไปในส่วนของโปรแกรมใช้งาน (เขียนด้วย Applesoft BASIC) ซึ่งเราจะกล่าวในภายหลัง

เนื่องจากโปรแกรม TIMER นี้จะนับรอบของสัญญาณนาฬิกา ดังนั้นค่าของเวลาจึงหาได้จากการนำรอบของสัญญาณนาฬิกามาหารด้วยความถี่ของสัญญาณนาฬิกาของระบบ ซึ่งมีค่าประมาณ 1.0227 MHZ

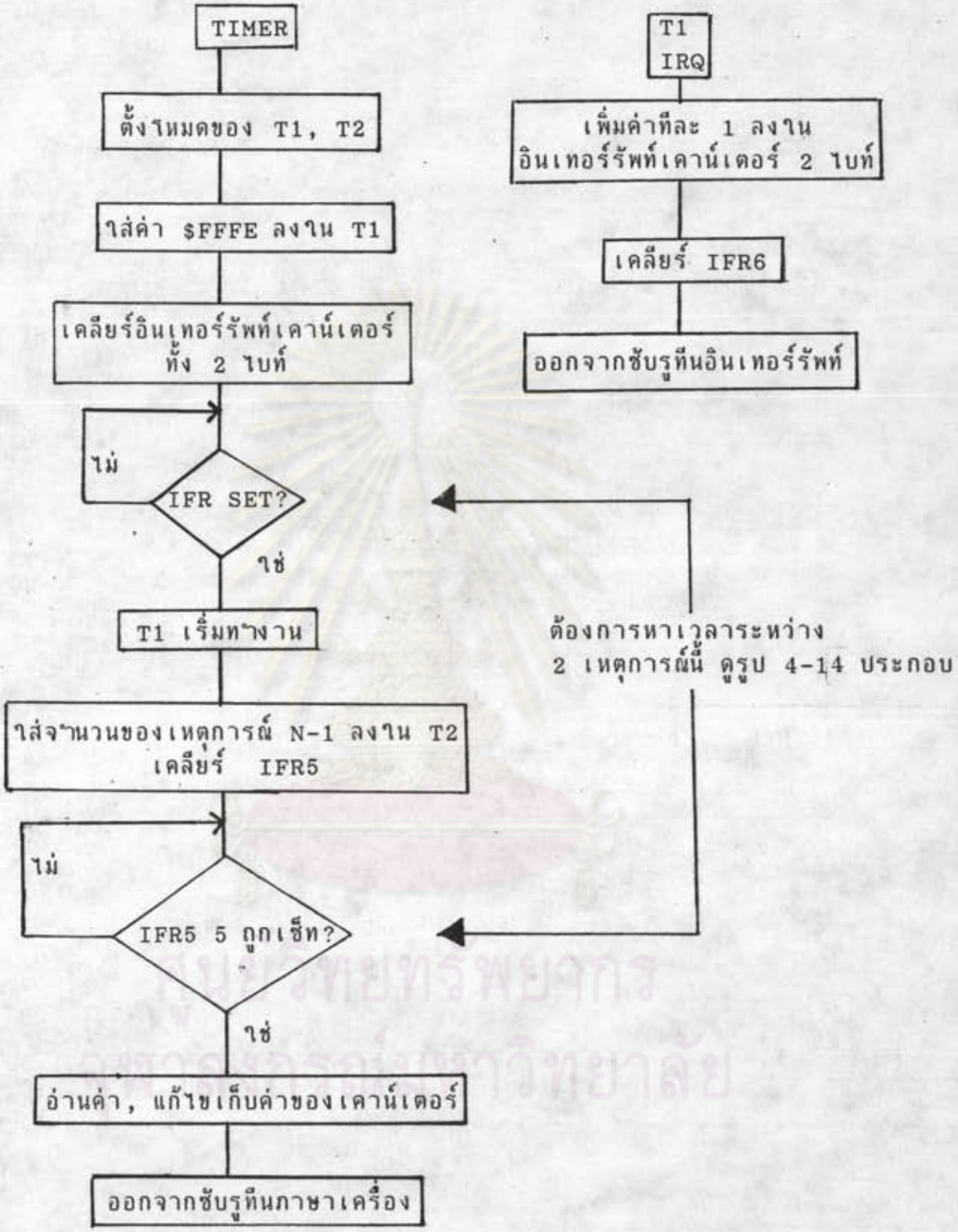
ซับรูทีน TIMER ประกอบด้วย 2 ส่วน คือซับรูทีนหลัก และซับรูทีนอินเทอร์รัพท์ โฟลว์ชาร์ตของโปรแกรม TIMER แสดงในรูปที่ 4-15 และตามด้วยลิสต์(list) ของโปรแกรม และคำอธิบายโปรแกรมตามลำดับ

4.18 โปรแกรมใช้งาน

โปรแกรมใช้งานจะเป็นโปรแกรมที่ติดต่อกับผู้ใช้งาน ได้ถูกเขียนด้วยภาษา Applesoft BASIC โดยมีหน้าที่ดังต่อไปนี้

1. ติดต่อกับผู้ใช้งาน
2. เรียกซับรูทีน TIMER
3. แปลงค่าจากซับรูทีน TIMER มาเป็นความถี่
4. บันทึกความถี่เริ่มต้นก่อนการฉาบโลหะแต่ละครั้ง
5. หาการเปลี่ยนแปลงความถี่เมื่อมีการฉาบโลหะและคำนวณออกมาเป็นความหนา

โฟลว์ชาร์ตของโปรแกรมใช้งานแสดงในรูปที่ 4-16 และตามด้วยลิสต์ของโปรแกรม



รูป 4-15 โฟลว์ชาร์ตของโปรแกรม TIMER

ซับรูทีน TIMER

```
 1           OBJ $1900
 2           ORG $1900
 3 TIME      EPZ $001D
 4 NUMB      EPZ $0019
 5 T1CL      EQU $C0F4
 6 T1CH      EQU $C0F5
 7 T1LL      EQU $C0F6
 8 T2CL      EQU $C0F8
 9 T2CH      EQU $C0F9
10 ACR       EQU $C0FB
11 IFR       EQU $C0FD
12 IER       EQU $C0FE
13 ********************
14 ;INTERRUPT ROUTINE
15           INC TIME
16           BNE BR1
17           INC TIME+1
18 BR1       LDA T1CL
19           LDA $45
20           RTI
21 ********************
22 ;BASIC ROUTINE
23           CLD
24           LDX #$FF
25           LDA #$60
26           STA ACR
27           LDA #$FE
28           STA T1LL
29           LDA #$C0
30           STA IER
31           LDA #00
32           STA TIME
33           STA TIME+1
34           STA T2CL
35           STA T2CH
36           LDA #$20
37 WAIT      BIT IFR
38           BEQ WAIT
39           STX T1CH
40           CLI
41           LDA NUMB
42           STA T2CL
43           LDA NUMB+1
44           STA T2CH
45           LDA #$20
46 LOAF      BIT IFR
47           BEQ LOAF
48           LDY T1CL
49           LDX T1CH
50           SEI
51           CPY #04
52           BCS ARND
53           INX
54           BNE ARND
55           SEC
56           LDA TIME
57           SBC #01
58           STA TIME
59           LDA TIME+1
60           SBC #00
61           STA TIME+1
62 ARND      STY TIME-2
63           STX TIME-1
64           LDA #$FE
65           SBC TIME-2
66           STA TIME-2
67           LDA #$FF
68           SBC TIME-1
69           STA TIME-1
70           RTS
71           END
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บรรทัดที่	คำอธิบายโปรแกรม TIMER
15	เพิ่มค่าทีละ 1 ให้กับอินเทอร์รัพท์เคาน์เตอร์ไบท์แรก
16	สำหรับการอินเทอร์รัพท์ของ T1 แต่ละครั้ง
17	เพิ่มค่าทีละ 1 ให้กับอินเทอร์รัพท์เคาน์เตอร์ไบท์ถัดไป
18	เคลียร์อินเทอร์รัพท์แฟลกของ T1
19	restore แอคคูมูเลเตอร์
20	ออกจากซัพรูทีนอินเทอร์รัพท์
23	เคลียร์แฟลกของโหมดเลขฐานสิบ (decimal mode flag) เพื่อให้ปฏิบัติการพีชคณิตตลอดซับรูทีนที่เป็นเลขฐานสอง (binary mode)
24	เอาค่า $FF เก็บไว้ที่ X เพื่อใส่ใน T1CH ในบรรทัดที่ 39
25	เอาค่า $60 (=01100000) ใส่ใน A
26	ใส่ข้อมูลของ A ลงใน ACR เพื่อตั้งให้ T1 อยู่ใน free-running mode และ T2 อยู่ในโหมดนับพัลส์
27	เอาค่า $FE ใส่ใน A
28	ใส่ข้อมูลของ A ลงใน T1LL
29	เอาค่า $C0 (=11000000) ใส่ใน A
30	ใส่ข้อมูลของ A ลงใน IER เพื่อตั้งให้ T1 สามารถรับสัญญาณอินเทอร์รัพท์
31	เอาค่า $00 ใส่ใน A
32	ใส่ข้อมูลของ A ลงในแอดเดรส 001D ซึ่งเป็นไบท์สูงของอินเทอร์รัพท์เคาน์เตอร์ในอินเทอร์รัพท์รูทีนเพื่อเคลียร์
33	ใส่ข้อมูลของ A ลงในแอดเดรส 001E ซึ่งเป็นไบท์ต่ำของอินเทอร์รัพท์เคาน์เตอร์เพื่อเคลียร์
34	ใส่ข้อมูลของ A ลงใน T2CL เพื่อใส่ค่า 0 ในไบท์ต่ำของ T2 (เพื่อให้จับเหตุการณ์หมายเลขศูนย์)
35	ใส่ข้อมูลของ A ลงใน T2CH เพื่อใส่ค่า 0 ในไบท์สูงของ T2 (ดูรูป 4-14)
36	เอาค่า $20 (=00100000) ใส่ใน A เพื่อทดสอบอินเทอร์รัพท์แฟลกของ T2 คือ IFR5 (บิทที่ 5 ของ IFR)

37 ทดสอบ IFR5 ว่าเป็น 1 แล้วยัง โดย AND กับค่าใน A
ถ้ายังไม่เป็น 1 จะทำให้แฟลก Z เป็น 0

38 หากแฟลก Z ยังเป็น 0 ให้กลับไปทำบรรทัด 37 จนกว่าเป็น 1 จึง
ไปทำบรรทัดต่อไป

39 ใส่ข้อมูลของ X ลงใน T1CH เพื่อทำให้ T1 เริ่มทำงาน

40 เคลียร์อินเทอร์รัพท์แฟลก เพื่ออนุญาตให้มีการอินเทอร์รัพท์แบบ IRQ

41 ใส่ค่าที่เก็บใส่แอดเดรส $19 ซึ่งเป็นไบท์ต่ำของจำนวนเหตุการณ์
(NLO) ลงใน A

42 ใส่ข้อมูลของ A ลงใน T2CL

43 ใส่ค่าที่เก็บในแอดเดรส $1A ซึ่งเป็นไบท์สูงของจำนวนเหตุการณ์
(NHI) ลงใน A

44 ใส่ข้อมูลของ A ลงใน T2CH

45 เอาค่า $20 (=00100000) ใส่ใน A อีกครั้งหนึ่ง เพื่อทดสอบ IFR5

46 ทดสอบ IFR5 ว่าเป็น 1 แล้วยัง โดย AND กับค่าใน A
ถ้ายังไม่เป็น 1 จะทำให้แฟลก Z เป็น 0

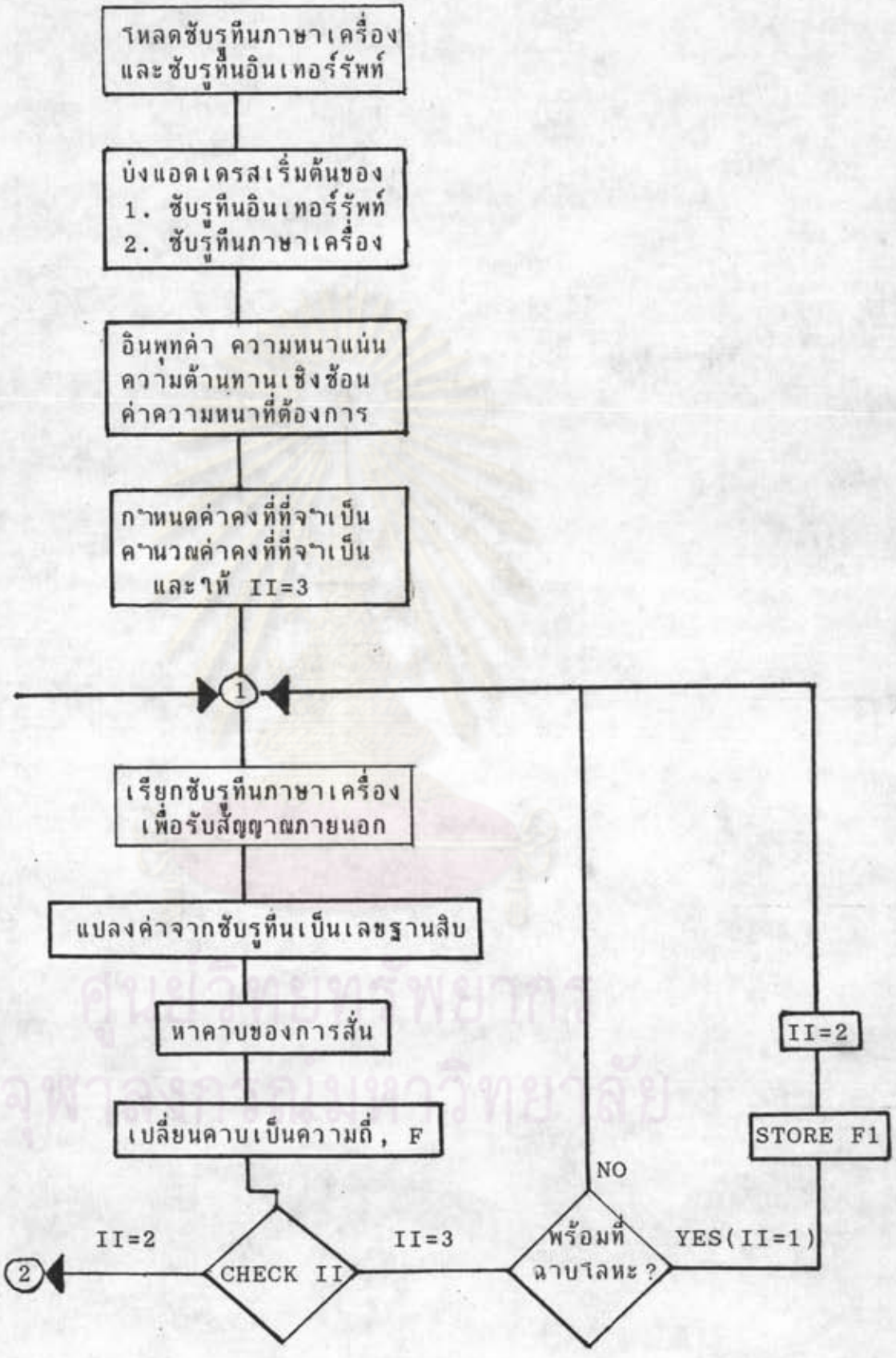
47 หากแฟลก Z ยังเป็น 0 กลับไปทำบรรทัด 46 จนกว่าจะเป็น 1
จึงไปบรรทัดต่อไป

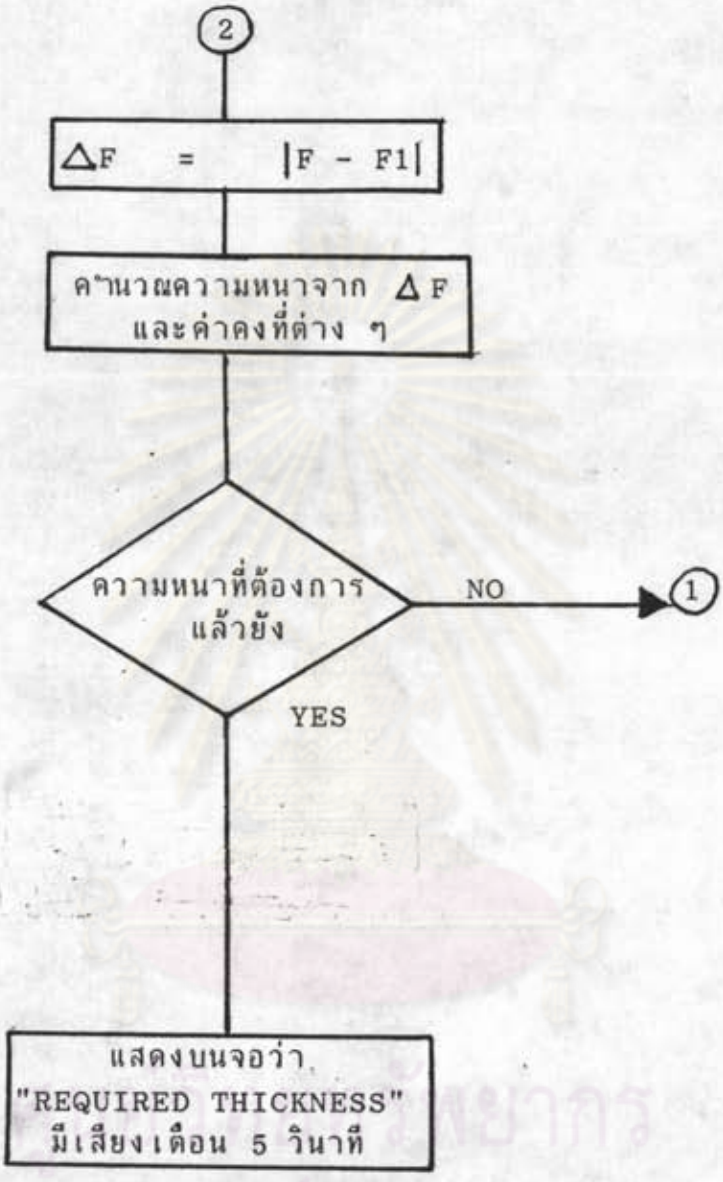
48 เอาค่าของ T1CL ใส่ใน Y (อ่านค่าไบท์ต่ำของ T1)

49 เอาค่าของ T1CH ใส่ใน X (อ่านค่าไบท์สูงของ T1)

50 เซ็ทอินเทอร์รัพท์แฟลก เพื่อป้องกันไม่ให้มีการอินเทอร์รัพท์อันต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


โหลดซับรูทีนภาษาเครื่อง
และซับรูทีนอินเทอร์รัพท์
บ่งแอดเดรสเริ่มต้นของ
1. ซับรูทีนอินเทอร์รัพท์
2. ซับรูทีนภาษาเครื่อง
อินพุทค่า ความหนาแน่น
ความต้านทานเชิงซ้อน
ค่าความหนาที่ต้องการ
กำหนดค่าคงที่ที่จำเป็น
คำนวณค่าคงที่ที่จำเป็น
และให้ II=3
1
เรียกซับรูทีนภาษาเครื่อง
เพื่อรับสัญญาณภายนอก
แปลงค่าจากซับรูทีนเป็นเลขฐานสิบ
หาคาบของการสั่น
เปลี่ยนคาบเป็นความถี่, F
II=2
STORE F1
NO
พร้อมที่
ฉาบโลหะ ?
YES(II=1)
II=2
CHECK II
II=3
2



รูป 4-16 โฟลว์ชาร์ตของโปรแกรมแสดงความหนา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมใช้งาน

```
]LIST

5  TEXT
10  REM DEVSEL
15  HOME : INVERSE : PRINT "THIN
     FILM THICKNESS MONITOR": NORMAL
     : PRINT : PRINT
20 II = 3: GOSUB 400
29  PRINT
30  PRINT  CHR$ (4)"BLOAD TIMER
34  REM  SET UP JUMP VECTORS
35  POKE 10,76: POKE 11,12: POKE
     12,25
36  POKE 1022,00: POKE 1023,25
37  DIM FE(100): DIM ACC(100)
38 FR = 6144000
39  GOSUB 1000
40 I = 0:J = 1:II = 3
60  GOSUB 700
64  REM CALL TIMER SUBROUTINE

65 Z =  USR (0)
69  REM  CONVERT NUMBER OF CLOCK
     CYCLES FROM HEX TO DEC .
70 A =  PEEK (27)
75 B = 256 *  PEEK (28) + A
80 C = 65536 *  PEEK (29) + B
85 D = 16777216 *  PEEK (30) + C
100 T = D / 1022714
105 F =  INT (M / T)
130  HTAB 1: VTAB 23: PRINT "F MI
     XER = ";F;" Hz."
135 I = I + 1
136  IF I = 1 THEN FE(1) = F:ACC(
     1) = 1
137  IF I > 1 THEN  GOSUB 200
138  IF I = 20 THEN  GOSUB 300: GOSUB
     400: GOSUB 550
150  GOTO 65
200  REM COLLECT NO.OF DIFFERENCE

205 CNT = J
210  FOR Z = 1 TO J
225  IF F = FE(Z) THEN ACC(Z) = A
     CC(Z) + 1:CNT = CNT - 1:Z =
     J
230  NEXT Z
240  IF CNT = J THEN J = J + 1:FE
     (J) = F:ACC(J) = 1

250  RETURN
```

```
300  REM
305 OFT = ACC(1):F2 = FE(1)
310  FOR Z = 2 TO J
320  IF ACC(Z) > OFT THEN OFT = A
     CC(Z):F2 = FE(Z)
330  NEXT Z
340  HTAB 1: VTAB 24: PRINT "PROB
     ABLE F = ";F2
345 I = 0:J = 1
350  IF II = 1 THEN  GOSUB 600
360  IF II = 2 THEN  GOSUB 2010
390  RETURN
400  IF II = 3 THEN  PRINT : PRINT
     "TO QUIT TYPE Q,": PRINT "OT
     HERWISE PRESS RETURN ";: GET
     S$: HOME : IF S$ = "QQ" THEN
      END
405  RETURN
550  IF II = 3 THEN  PRINT : PRINT
     "ARE YOU READY TO COAT?";: GET
     S$: IF S$ = "Y" THEN II = 1:
     S$ = "N"
590  RETURN
600  REM STORE FIRST FREQ.
610 F1 = F2
615 FQ = FR + F1
620 II = 2

650  RETURN
700  REM GIVE NO. OF EVENTS
705  REM INPUT THE NUMBER OF EVEN
     T
710  REM THIS NUMBER MUST BE LESS
      THAN 65537
715 M = 30000
720 N = M - 1
725  REM POKE N INTO 2 LOCATIONS
730 NHI =  INT (N / 256)
735  POKE 26,NHI
740 NLO = (N / 256 - NHI) * 256
745  POKE 25,NLO
799  RETURN
1000  REM ENTER INFORMATIONS
1001  HOME
1005  INVERSE : PRINT "ENTER INFO
     RMATIONS": NORMAL : PRINT
1009 PI = 3.14285714
1010 RQ = 2.651
1011  REM QUARTZ DENSITY IN g/cc.

1012 ZQ = 8.83
1013  REM ACOUSTIC IMPEDANCE OF Q

1015 NQ = 1.668E13
1016  REM FREQ CONSTANT FOR ATcut
```

```
1020  PRINT : INPUT "FILM DENSITY
     (IN g/cc.) = ";RF
1030  PRINT : INPUT "ACOUSTIC IMP
     EDANCE = ";ZF
1035 ZR = ZQ / ZF
1040  PRINT : INPUT "REQUIRED THI
     CKNESS(IN Ang.) = ";DD
1050 CC = NQ * RQ / RF
1080  HOME
1090  RETURN
2000  REM CALCULATE THICKNESS AND
      COMPARE WITH WHAT REQUIRED
2010 FD =  ABS (F1 - F2)
2015 FF = FR + F2
2020 DF = CC * FD / FF ^ 2
2021  REM
2024  HTAB 23: VTAB 23: PRINT "DE
     LTA F = ";FD;" Hz."
2025  HTAB 23: VTAB 24: PRINT "TH
     ICKNESS = ";DF;" Ang."
2030  IF DF = DD THEN  GOSUB 3000

2050  RETURN
3000  REM WARNING
3080  HTAB 1: VTAB 24: FLASH : PRINT
     "REQUIRED THICKNESS": NORMAL

3085  GOSUB 4000
3086 II = 3: GOSUB 400
3090  RETURN
4000  REM  SOUNDING
4003  FOR H = 1 TO 400
4005 C =  PEEK (49200)
4010  NEXT
4015  RETURN
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย