สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์

นางสาวกรรณิการ์ กำจายกิตติกุล
นายไชยกิจ กิจส่งแสง

ปพ.
ก173ค
2536

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

b1955034

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

MICROCONTROLLER BASED NUCLEAR RADIATION COUNTER

MISS KANNIKAR KAMJAIKITTIKUL

MR. CHAIYAKIT KITSONGSANG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF APPLIED PHYSICS

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1993

หัวข้อโครงงานพิเศษ	เครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์
โดย	นางสาวกรรณิการ์ กำจายกิตติกุล
	นายไชยกิจ กิจส่งแสง
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.ดร. บุญส่ง ศิวโมกษธรรม
	อ. วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับรายงานโครงงานพิเศษฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

(ผศ. ปรีชา เทียนสมประสงค์) หัวหน้าภาควิชาฟิสิกส์

คณะกรรมการสอบโครงงานพิเศษ

(ผศ.ดร. บุญส่ง ศิวโมกษธรรม) ประธานกรรมการ

(อ. วิชิต ศิริโชติ) กรรมการ

(ดร. ริตินัย แก้วแดง) กรรมการ

(อ. วิชาญ กกกนทา) กรรมการ

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	เครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์
โดย	นางสาวกรรณิการ์ กำจายกิตติกุล
	นายไชยกิจ กิจส่งแสง
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.ดร. บุญส่ง ศิวโมกษธรรม
	อ. วิชิต ศิริโชติ
ปีการศึกษา	2536

บทคัดย่อ

โครงงานพิเศษนี้เป็นการสร้างเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งจะประกอบด้วย หัววัดรังสี (แบบไกเกอร์-มุลเลอร์), แหล่งจ่ายแรงดันไฟฟ้าขนาดสูง, พรีแอมพลิไฟเออร์, วงจรขยายและปรับแต่งสัญญาณ, วงจรนับ, วงจรตั้งเวลา และส่วนแสดงผล โดยใช้ไมโครคอนโทรลเลอร์มาควบคุมวงจรนับและภาคแสดงผล อีกทั้งเป็นวงจรตั้งเวลาในตัวด้วย ส่วนแหล่งจ่ายแรงดันไฟฟ้าขนาดสูงจะใช้แบบสวิตช์ชิ่ง นอกจากนี้ ยังออกแบบให้สามารถนับความถี่ของสัญญาณอื่นๆ เช่น sine wave, square wave และ triangle wave ได้อีกด้วย โดยการเพิ่มส่วนของวงจรปรับแต่งสัญญาณเข้าไปอีกหนึ่งชุด

Special Project Title	Microcontroller Based Nuclear Radiation Counter
Name	Miss Kannikar Kamjaikittikul
	Mr. Chaiyakit Kitsongsang
Special Project Advisor	Dr. Bunsong Siwamoksatham
	Mr. Wichit Sirichote
Department	Applied Physics
Acadamic Year	1993

Abstract

This special project is the development of a microcontroller based nuclear radiation counter. The counter consists of Geiger-Müller counter, high voltage, preamplifier, amplifier, counter , timer and display unit. The counter uses the 8751 single chip microcontroller to control counter and display unit and to be a timer. The high voltage is a switching type power supply. In addition, the counter can be used for counting frequency of other signal (for example, sine wave, square wave and triangle wave) by using a waveshaping circuit.

กิตติกรรมประกาศ

โครงงานพิเศษนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือจากบุคคลต่างๆ ดังดังต่อไปนี้

บิดา-มารดา รวมทั้งผู้อุปการะที่ได้ให้การศึกษา, เลี้ยงดู, ช่วยเหลือ และให้กำลังใจในการทำงาน

ผศ.ดร. บุญส่ง ศิวโมกษธรรม และ อ. วิชิต ศิริโชติ ที่คอยให้คำแนะนำและคำปรึกษาในการทำโครงงานพิเศษ

บุคคลต่างๆที่อยู่ในสำนักงานพลังงานปรมาณูเพื่อสันติ

- กองฟิสิกส์

  ดร. สมพร จองคำ, อ. ชาญชัย, พี่วราวุฒิและอีกหลายๆ คน ที่เอื้อเฟื้อสถานที่ทดลอง, เครื่องมือในการทดลอง และให้คำแนะนำทางด้านระบบการวัดรังสี

- กองอุปกรณ์อิเลคทรอนิคส์

  คุณสุรชัย พงษ์เจริญสุข ที่ให้คำแนะนำและอุปกรณ์ทางอิเลคทรอนิคส์

เพื่อนๆ หลายๆคนที่มีน้ำใจ

- รุ่งเรือง, เสริมชัย และ อภินันท์ เอื้อเฟื้อคอมพิวเตอร์
- ศุภชัย, อรรถพร, ประวิทย์ และสมนึก เอื้อเฟื้ออุปกรณ์
- ชาญวิทย์, ภูเบศร์ และชินวุฒน์ ให้คำแนะนำด้านวงจรอิเลคทรอนิคส์

คณะกรรมการที่ช่วยตรวจทานและแก้ไขรายงานให้เป็นฉบับที่สมบูรณ์

ขอขอบพระคุณกลุ่มบุคคลเหล่านี้ด้วยใจจริง

กรรณิการ์ กำจายกิตติกุล

ไชยกิจ กิจส่งแสง

สารบัญตาราง

		หน้า
ตารางที่ 2.1	วัตถุที่ใช้ทำหัววัดแบบชินทิลเลชั่น	19
ตารางที่ 5.1	แสดงผลการทดลองที่ 5.1 ตอนที่ 1	68
ตารางที่ 5.2	แสดงผลการทดลองที่ 5.1 ตอนที่ 2	69
ตารางที่ 5.3	แสดงผลการทดลองที่ 5.2	71
ตารางที่ 5.4	แสดงผลการทดลองที่ได้จากเครื่องนับอ้างอิง	73
ตารางที่ 5.5	แสดงผลการทดลองที่ได้จากเครื่องนับที่สร้างขึ้น	73
ตารางที่ 5.6	เปรียบเทียบผลการทดลองของเครื่องนับอ้างอิงกับเครื่องนับที่สร้างขึ้น	74

สารบัญรูป

		หน้า
รูปที่ 2.1	บล็อคไดอะแกรมแสดงส่วนประกอบของระบบทั่วไปของการนับวัดรังสี	3
รูปที่ 2.2	ระบบของหัววัดแบบซินทิลเลชั่น	4
รูปที่ 2.3	รูปร่างสัญญาณ การแปรค่าของแอมพลิจูดสัญญาณตามเวลา	6
รูปที่ 2.4	แสดงสเปคตรัมของสัญญาณ	7
รูปที่ 2.5	ตัวอย่างสเปคตรัมจากหัววัดแบบซินทิลเลชั่นต้นกำเนิดรังสีคือ ^{99m}Tc	8
รูปที่ 2.6	วงจรพื้นฐานสำหรับหัววัดรังสีแบบใช้กาซ	10
รูปที่ 2.7	รูปร่างของสัญญาณขึ้นกับค่าไทม์คอนสแตนท์ของวงจรสำหรับหัววัดรังสี	11
รูปที่ 2.8	กราฟของความสูงสัญญาณ กับค่าแรงดันไฟฟ้าคร่อมระหว่างหัววัดรังสีแบบกาซ	11
รูปที่ 2.9	กราฟแสดงค่า gas multiplication กับค่าแรงดันไฟฟ้าสำหรับความกดดัน 10 และ 40 ซม.ปรอทในถังอาร์กอน, เส้นผ่าศูนย์กลางของอาโนด = 0.01 นิ้ว, เส้นผ่าศูนย์กลางของคาโทด = 0.87 นิ้ว	13
รูปที่ 2.10	ขบวนการแตกตัวของกาซในหัววัดไกเกอร์-มุลเลอร์	14
รูปที่ 2.11	กราฟแสดงลักษณะเฉพาะการทำงานหัววัดรังสีแบบไกเกอร์-มุลเลอร์	15
รูปที่ 2.12	กราฟแสดงความสัมพันธ์ของเดดไทม์, ริโคเวอรีไทม์และริซอลวิงไทม์	17
รูปที่ 2.13	ภาพแสดงรอยต่อของหัววัดรังสีแบบสารกึ่งตัวนำ	18
รูปที่ 2.14	ประสิทธิภาพของการวัดรังสีแกมมาพลังงานต่างๆ สำหรับผลึกของหัววัดรังสีแบบซินทิลเลชั่นที่ดี	20
รูปที่ 2.15	แผนภาพแสดงถึงเหตุการณ์ที่เกิดขึ้น เมื่อตอนผ่านเข้าไปในหัววัดรังสีซินทิลเลชั่น	20
รูปที่ 3.1	ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51	23
รูปที่ 3.2	ไดอะแกรมโครงสร้างของ 8051	24
รูปที่ 3.3	แผนภูมิหน่วยความจำของ 8051	26
รูปที่ 3.4	สถาปัตยกรรมภายในของ 8051	28
รูปที่ 3.5	ไดอะแกรมขาของ 8051 แบบ DIP	28
รูปที่ 3.6	ค่าของรีจีสเตอร์เมื่อเกิดการรีเซ็ท 8051	31
รูปที่ 3.7	วงจรออสซิลเลเตอร์ภายใน 8051	33

รูปที่ 3.8	8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก	33
รูปที่ 3.9	Special Function Register (SFR)	34
รูปที่ 3.10	แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท	35
รูปที่ 3.11	Serial Port Control Register (SCON)	38
รูปที่ 3.12	TMOD Timer/Counter Mode Register	39
รูปที่ 3.13	TCON Timer Control Register	40
รูปที่ 3.14	แหล่งกำเนิดสัญญาณขัดจังหวะ	41
รูปที่ 3.15	Interrupt Enable Register	42
รูปที่ 3.16	IP :Interrupt Priority Register	43
รูปที่ 3.17	ระบบการขัดจังหวะของ 8052 และ 83154	44
รูปที่ 3.18	PCON : Power Control Register	45
รูปที่ 3.19	ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ	47
รูปที่ 3.20	ไดอะแกรมเวลาของการรีเซ็ท	49
รูปที่ 3.21	ไดอะแกรมของการอ่านโปรแกรมจากหน่วยความจำภายนอก	51
รูปที่ 3.22	ไดอะแกรมเวลาของการอ่านข้อมูลจากหน่วยความจำ สำหรับข้อมูลภายนอก 8051	52
รูปที่ 3.23	ไดอะแกรมเวลาการเขียนข้อมูลไปยังหน่วยความจำ สำหรับข้อมูลภายนอก 8051	53
รูปที่ 4.1	บล็อกไดอะแกรมแสดงโครงสร้างของเครื่องนับรังสี ควบคุมด้วยไมโครคอนโทรลเลอร์	55
รูปที่ 4.2	แผนภาพแสดงส่วนของหัววัดรังสี, พรีแอมพลิไฟเออร์ และส่วนขยายสัญญาณ	56
รูปที่ 4.3	แผนภาพของวงจรนับที่ต่อเข้ากับไมโครคอนโทรลเลอร์	58
รูปที่ 4.4	แผนภาพแสดงการต่ออุปกรณ์ต่างๆ เข้ากับไมโครคอนโทรลเลอร์	59
รูปที่ 4.5	แผนภาพแสดงการต่อ Keyboard เข้ากับไมโครคอนโทรลเลอร์	60
รูปที่ 4.6	แผนภาพแสดงการต่อ Dot Matrix LCD เข้ากับไมโครคอนโทรลเลอร์	61
รูปที่ 4.7	การควบคุมขาสัญญาณเพื่อให้ Display LCD ทำงานในลักษณะต่างๆ	62
รูปที่ 4.8	บล็อกไดอะแกรมแสดงการทำงานของแหล่งจ่ายไฟแบบสวิตชิ่ง	63
รูปที่ 4.9	หลักการของคอนเวอร์เตอร์แบบฟลายแบ็ค	64

รูปที่ 4.10 วงจรของแหล่งจ่ายแรงดันไฟฟ้าขนาดสูง ที่ใช้ในเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์ 65
รูปที่ 4.11 แผนภาพแสดงวงจรปรับแต่งสัญญาณ 66
รูปที่ 5.1 แสดงการต่อวงจรของการทดลองที่ 5.2 70
รูปที่ 5.2 แสดงการจัดอุปกรณ์ทดลองในการทดลองที่ 5.3 72

สารบัญ

	หน้า
บทคัดย่อโครงงานพิเศษภาษาไทย	ก
บทคัดย่อโครงงานพิเศษภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญตาราง	ง
สารบัญรูป	จ
บทที่ 1 บทนำ	1
บทที่ 2 ระบบการนับวัดปริมาณรังสี	2
2.1 รังสี	2
2.2 การนับวัดปริมาณรังสี	3
บทที่ 3 ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51	23
3.1 โครงสร้างของ 8051	24
3.2 สถาปัตยกรรมของ 8051	26
3.3 รีจิสเตอร์ฟังก์ชันพิเศษ	34
3.4 การขัดจังหวะ	47
3.5 การรีเซ็ท	49
3.6 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ	51
บทที่ 4 เครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์	54
บทที่ 5 การทดลองวงจรและเครื่องนับรังสี	68
การทดลองที่ 5.1 ทดสอบวงจรนับที่ควบคุมด้วยไมโครคอนโทรลเลอร์	68
การทดลองที่ 5.2 ทดสอบแหล่งจ่ายไฟกระแสตรงแรงดันสูง	70
การทดลองที่ 5.3 ทดสอบเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์	72

บทที่ 6 สรุปผลการศึกษาวิจัยและข้อเสนอแนะ 76

ภาคผนวก ก. ชุดคำสั่งของ MCS51

ภาคผนวก ข. แนวทางการเขียนโปรแกรม

ภาคผนวก ค. โปรแกรมควบคุมไมโครคอนโทรลเลอร์

เอกสารอ้างอิง

ประวัติผู้เขียน

บทที่ 1

บทนำ

ในปัจจุบันนี้เทคโนโลยีทางด้านนิวเคลียร์ ได้เข้ามามีบทบาทมากในชีวิตประจำวันของมนุษย์ ตัวอย่าง เช่น โรงงานไฟฟ้านิวเคลียร์ ซึ่งแสดงให้เห็นว่าพลังงานนิวเคลียร์ได้ถูกนำมาใช้งานอย่างแพร่หลาย ดังนั้นการศึกษาถึงพื้นฐานของเทคโนโลยีทางด้านนี้จึงเป็นสิ่งที่น่าสนใจอย่างยิ่ง โดยพื้นฐานเหล่านี้ ได้แก่ ระบบการนับรังสี ซึ่งนักวิทยาศาสตร์ได้นำมาใช้ทดลองเพื่อค้นคว้าหาข้อมูล, ทฤษฎี และยังมีการนำไปประยุกต์ใช้งานด้านอื่นๆ อีกด้วย เช่น ใช้ในทางอุตสาหกรรมและในทางการแพทย์

ดังนั้นในโครงงานนี้จึงได้สร้างระบบการนับอย่างง่ายขึ้นมาเพื่อใช้ในการนับวัดปริมาณรังสีทั่วไป ซึ่งระบบการนับที่ถูกสร้างขึ้นมานี้เป็นเครื่องนับรังสีที่ถูกควบคุมด้วยไมโครคอนโทรลเลอร์ตระกูล 51 และมีส่วนแสดงผลเป็น LCD แบบ Dot Matrix ทำให้การใช้งานเครื่องนับรังสีมีความสะดวกขึ้น โดยยังมีฟังก์ชันการทำงานของเครื่องนับแบบเดิมอยู่ครบ

1.1 วัตถุประสงค์ของโครงงาน

1.1.1) เพื่อศึกษาและสร้างเครื่องนับรังสีโดยควบคุมด้วยไมโครคอนโทรลเลอร์

1.1.2) สามารถนำเอาความรู้ทางด้านนิวเคลียร์และอิเล็กทรอนิกส์มาใช้งานได้

1.2 วิธีดำเนินการโดยย่อ

1.2.1) ศึกษาหาข้อมูลของระบบการนับรังสี และไมโครคอนโทรลเลอร์

1.2.2) ออกแบบวงจรนับรังสี โดยแบ่งเป็นส่วนย่อย ได้แก่ counter, pre-amplifier, amplifier, waveshaping และ high voltage

1.2.3) สร้างและทดสอบการทำงานของวงจรส่วนย่อยต่าง ๆ

1.2.4) ทดสอบระบบโดยรวมเมื่อนำมาประกอบกันเป็นเครื่องนับรังสี

1.2.5) สรุปผลการศึกษาวิจัยและหาหัวข้อเสนอแนะ

1.3 ประโยชน์ที่ได้รับจากการทำโครงงาน

1.3.1) สามารถประยุกต์ใช้งานไมโครคอนโทรลเลอร์กับงานทางด้านนิวเคลียร์เข้าด้วยกันได้

1.3.2) ได้รับความรู้ทางด้านการตรวจวัดรังสี , อิเล็กทรอนิกส์ รวมทั้งไมโครคอนโทรลเลอร์

บทที่ 2

ระบบการนับวัดปริมาณรังสี

2.1 รังสี

จากการที่นิวเคลียสของอะตอมประกอบด้วยอนุภาคหลักอยู่ 2 ชนิด คือโปรตอนและนิวตรอนนั้น เราพบว่ามีนิวเคลียสของธาตุบางอย่างที่เราเรียกว่า "สารกัมมันตรังสี" เป็นนิวเคลียสที่ไม่เสถียรและพยายามที่จะปรับตัวเองให้อยู่ในสภาพที่เสถียร โดยการปล่อยอนุภาคพลังงานสูงหรือคลื่นแม่เหล็กไฟฟ้าที่เราเรียกว่า "รังสี" ออกมา ซึ่งรังสีที่แผ่ออกมาจากนิวเคลียสของสารกัมมันตรังสีที่สำคัญมี 3 ชนิด คือ

1. รังสีอัลฟา (alpha rays) รังสีชนิดนี้เป็นนิวเคลียสของธาตุฮีเลียมที่แผ่ออกมาจากนิวเคลียสของสารกัมมันตรังสีบางชนิด เช่น เรเดียม รังสีชนิดนี้มีอำนาจทะลุทะลวงต่ำมาก แต่อำนาจการไอออไนซ์สูงมาก เคลื่อนที่เร็วประมาณ 0.1 เท่าของความเร็วแสง

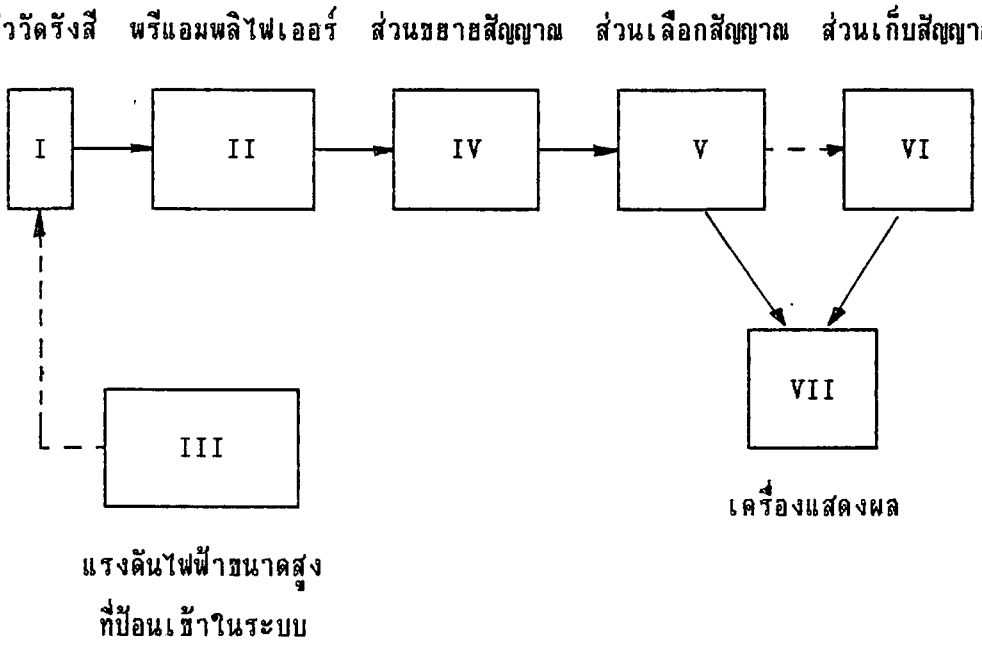
2. รังสีเบตา (beta rays) รังสีเบตาเป็นอิเลคตรอนพลังงานสูงที่แผ่ออกมาจากนิวเคลียส บางตัวอาจเคลื่อนที่เร็วถึง 0.99 เท่าของความเร็วแสง มีอำนาจในการทะลุทะลวงปานกลาง อำนาจในการไอออไนซ์ต่ำกว่ารังสีอัลฟา แต่พิสัย (range) มากกว่ารังสีอัลฟา

3. รังสีแกมมา (gamma rays) เป็นคลื่นแม่เหล็กไฟฟ้าหรือโฟตอนที่แผ่ออกมาจากนิวเคลียส มีความเร็วเท่าแสงและมีอำนาจในการทะลุทะลวงสูงมาก แต่อำนาจในการไอออไนซ์ต่ำมาก

นอกจากรังสีทั้ง 3 ชนิดนี้ ซึ่งเป็นรังสีที่แผ่ออกมาจากนิวเคลียสของสารกัมมันตรังสีเป็นส่วนใหญ่แล้ว ยังมีรังสีชนิดอื่นๆ ที่อาจแผ่ออกมาจากนิวเคลียสของสารกัมมันตรังสีบางชนิดได้อีก เช่น รังสีโปรตอน นิวตรอนและโปรสิตรอน เป็นต้น

2.2 การนับวัดปริมาณรังสี

มนุษย์ไม่สามารถที่จะวัดรังสีโดยใช้ประสาทสัมผัส, การมองเห็น ฯลฯ ดังนั้น ในการวัดหรือตรวจสอบรังสีนั้น มนุษย์จึงจำเป็นต้องคิดค้นประดิษฐ์เครื่องมือขึ้นมาวัดรังสี ซึ่งเครื่องมือวัดรังสีโดยทั่วไป จะมีระบบการนับวัดปริมาณรังสีที่ประกอบด้วยส่วนต่างๆ ดังรูปที่ 2.1 หัววัดรังสีไม่ว่าชนิดใดก็ตาม จะทำงานได้ก็ต่อเมื่อได้รับการป้อนแรงดันไฟฟ้าขนาดสูง หลังจากได้มีขบวนการก่อให้เกิดสัญญาณในหัววัดรังสีขึ้นแล้ว สัญญาณดังกล่าวจะถูกส่งต่อไปยังส่วนขยายเพื่อให้ปริมาณไฟฟ้าถูกนับวัดได้โดยง่าย จากนั้นจะถูกส่งต่อไปยังส่วนเลือกสัญญาณ ซึ่งจะทำหน้าที่กำจัดสัญญาณที่มิใช่เกิดจากสารกัมมันตรังสี ต่อไปคือส่วนเก็บสัญญาณ และองค์ประกอบท้ายสุดคือส่วนแสดงผล

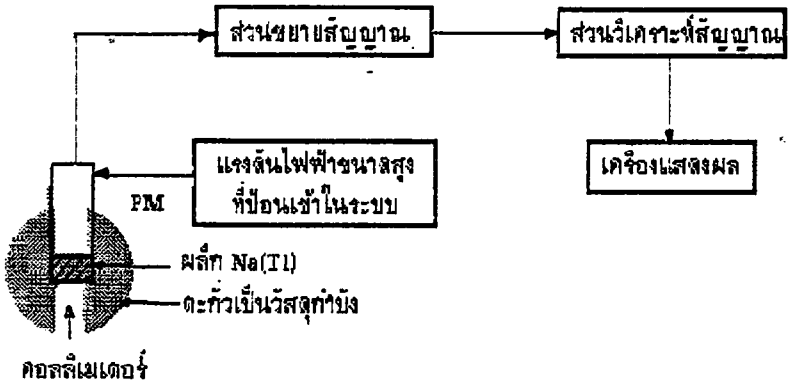


รูปที่ 2.1 บล็อคไดอะแกรมแสดงส่วนประกอบของระบบทั่วไปของการนับวัดรังสี

อย่างไรก็ตาม หลักการคิดและการสุ่มโครงสร้างของระบบการวัดดังกล่าวเป็นเพียงพื้นฐานทั่วๆ ไป ในหัววัดบางชนิด เช่น หัววัดไกเกอร์-มุลเลอร์ มักจะมีเฉพาะส่วนที่ III, IV และ VII โดยขาดส่วนที่ II และ VI เป็นต้น แต่สำหรับหัววัดแบบสารกึ่งตัวนำและหัววัดซินทิลเลชั่นนั้นจำเป็นต้องมีส่วนประกอบดังกล่าวแยกโดยเฉพาะ รูปที่ 2.2 แสดงระบบของหัววัดซินทิลเลชั่น ที่นิยมใช้ในสาขาเวชศาสตร์นิวเคลียร์เพื่อความถูกต้องของค่าที่นับวัด สัญญาณของหัววัดจะอยู่ในรูปของสัญญาณไฟฟ้า มีลักษณะสมบัติอยู่ในเทอมของ

แอมพลิจูด (amplitude) ช่วงเวลาการเกิดสัญญาณ และค่า rise time ซึ่งค่าต่างๆ เหล่านี้แตกต่างกันตามชนิดของหัววัด และในบางกรณีอาจจะรวมทั้งชนิดของรังสีที่ทำการวัดด้วย กล่าวคือ ชนิดของรังสี เช่น แกมมาหรือเบตา และพลังงานที่สอดคล้อง

ในที่นี้จะขอกล่าวถึงรายละเอียดต่างๆ ขององค์ประกอบระบบการนับวัดปริมาณรังสีพอสมควร เพื่อใช้เป็นพื้นฐานในการใช้เครื่องมือต่อไป



รูปที่ 2.2 ระบบของหัววัดแบบซินทิลเลชั่น โดยที่ส่วนทั้งสี่ที่อยู่ในกรอบมักจะรวมเป็นอุปกรณ์ชิ้นเดียว

1. พรีแอมพลิไฟเออร์ (preamplifier)

สัญญาณจากหัววัดรังสีที่ถูกส่งต่อไปยังส่วนขยายสัญญาณตามสายเคเบิ้ลอาจถูกรบกวน(distort) หรือถูกลดขนาดลงได้ นอกจากนี้หัววัดรังสีมักเป็นเครื่องมือประเภทความจุประจุต่ำแต่อิมพีแดนซ์สูง ดังนั้นจึงได้มีการเสนอเครื่องที่อยู่ก่อนส่วนขยายสัญญาณเพื่อทำหน้าที่เป็นตัวปรับค่าอิมพีแดนซ์ให้พอเหมาะที่จะเป็นสัญญาณ "อินพุท" ของเครื่องขยายสัญญาณและเรียกชื่อตามหน้าที่ในระบบนับวัดรังสีว่า "พรีแอมพลิไฟเออร์" (preamplifier) หรือ "ส่วนก่อนขยายสัญญาณ" กรณีที่หัววัดสามารถทำให้เกิดกระแสไฟฟ้าได้มากกว่า 10 พิโคคูลอมบ์ (picocoulomb) ต่อหนึ่งสัญญาณ (เมื่อต่อกับสายเคเบิ้ลสัญญาณซึ่งยาวน้อยกว่า 1 เมตร) มักจะตัดส่วนก่อนขยายสัญญาณออก

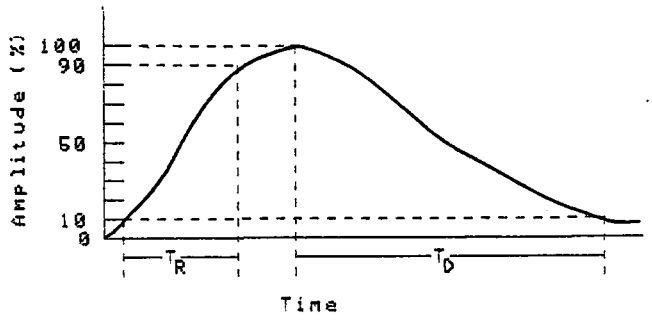
พรีแอมพลิไฟเออร์ชนิดง่ายที่สุดเรียก "คาโทด ฟอลโลเวอร์" (cathod follower) ทำหน้าที่ปรับอิมพีแดนซ์ระหว่างหัววัดกับส่วนขยายสัญญาณได้ดี เนื่องจากวงจรชนิดนี้มีค่าอัตราขยาย (gain) น้อยกว่าหนึ่ง ("เกน" คือค่าอัตราส่วนแอมพลิจูดของสัญญาณขาออกกับสัญญาณขาเข้า) พรีแอมพลิไฟเออร์ได้ถูกวิวัฒนาการให้มีความไวต่อประจุ (หรือกระแส) มาก หัววัดแบบสารกึ่งตัวนำต้องใช้พรีแอมพลิไฟเออร์ที่มีความไวต่อประจุเนื่องจากค่าความจุเป็นฟังก์ชันแปรค่ากับแรงดันไฟฟ้าที่ป้อนให้ (รวมทั้งเงื่อนไขอื่นๆ ที่อยู่ในระบบด้วย) อย่างไรก็ตาม เมื่อพิจารณาคุณสมบัติมักจะกล่าวถึง "ความไวต่อประจุ" มากกว่า "อัตราขยาย" ของพรีแอมพลิไฟเออร์ ซึ่งหมายถึงแรงดันไฟฟ้าเอาท์พุทอันเป็นผลจากการเกิดประจุขึ้นในหัววัดรังสีซึ่งมีค่าอิมพีแดนซ์สูง สำหรับหัววัดสารกึ่งตัวนำบางชนิดอาจมีค่าดังกล่าวเป็นมิลลิโวลต์ต่อพิโคคูลอมบ์ (picocoulomb) หรือมิลลิโวลต์ต่อ MeV

2. ส่วนขยายสัญญาณ (amplifier)

ส่วนขยายสัญญาณหรือส่วนที่เรียกว่า "แอมพลิไฟเออร์" (amplifier) ทำหน้าที่ขยายขนาดของสัญญาณที่ได้จากหัววัดรังสี (หรือได้จากส่วนก่อนขยายสัญญาณ) โดยที่อุปกรณ์ส่วนนี้จะต้องมีอัตราขยายที่มากพอที่จะทำให้ส่วนเลือกสัญญาณทำงานได้สะดวก ขณะเดียวกันต้องรักษาจุดทำงานของหัววัดให้อยู่ในช่วงที่กำหนดไว้ด้วย และอัตราขยายต้องมีค่าคงที่แม้ว่าจะมีค่ารบกวนซึ่งเกิดจากการเปลี่ยนแปลงแรงดันไฟฟ้า จากอุณหภูมิ หรือ จากแฟคเตอร์อื่นซึ่งอาจเกิดแบบชั่วขณะหรือเกิดขึ้นเป็นเวลานานก็ตาม นอกจากนี้จะต้องประกอบด้วยค่าไรส์ไทม์ (rise time) ที่ถูกต้อง เพื่อมิให้เกิดการสูญเสียข้อมูลในการนับวัด กล่าวคือ ไรส์ไทม์ต้องมีค่าพอเหมาะ ไม่ช้าหรือเร็วเกินไป (ถ้าเร็วเกินไปจะทำให้ได้ค่านับวัดอันเกิดจากรังสีรบกวนอื่นๆ นอกเหนือจากปริมาณที่ต้องการนับวัดแท้จริง)

การแบ่งระดับของส่วนขยายสัญญาณ มักจะพิจารณาจากความไวต่อแรงดันไฟฟ้า หรือ ความไวต่อประจุเช่นเดียวกับการพิจารณาส่วนก่อนขยายสัญญาณ รูปที่ 2.3 แสดงค่าขนาดความสูงของสัญญาณที่เปลี่ยนแปลงตามเวลา โดยขนาดความสูงของสัญญาณอาจเป็นค่าแรงดันไฟฟ้าเอาท์พุท หรือกระแสเอาท์พุทที่เกิดขึ้นในช่วงเวลาจำกัดค่าใดค่าหนึ่ง

- "ไรส์ไทม์" หรือ "เวลาที่เพิ่มขึ้น" (rise time) หรือ T_R คือช่วงเวลาที่ค่าสูงของสัญญาณเริ่มเพิ่มขนาดจาก 10% จนถึง 90% ของค่าสูงสุด
- "เวลาการสลายตัว" (decay time) หรือ T_D เป็นช่วงเวลาที่สัญญาณมีขนาดลดลงจากค่าสูงสุดจนถึงค่าเพียง 10% ของค่าสูงสุด
- "ดีเลย์ไทม์" (delay time) เป็นช่วงเวลาที่สัญญาณจะเพิ่มขนาดจนถึง 50% ของค่าสูงสุด



รูปที่ 2.3 รูปร่างสัญญาณ การแปรค่าของแอมพลิจูดสัญญาณตามเวลา

- "ช่วงเวลาเกิดสัญญาณ" (pulse duration) ในทางปฏิบัติค่านี้วัดจากสัญญาณเริ่มต้นจนถึงเวลาที่สัญญาณมีค่าลดลงถึง 10% ของค่าสูงสุด

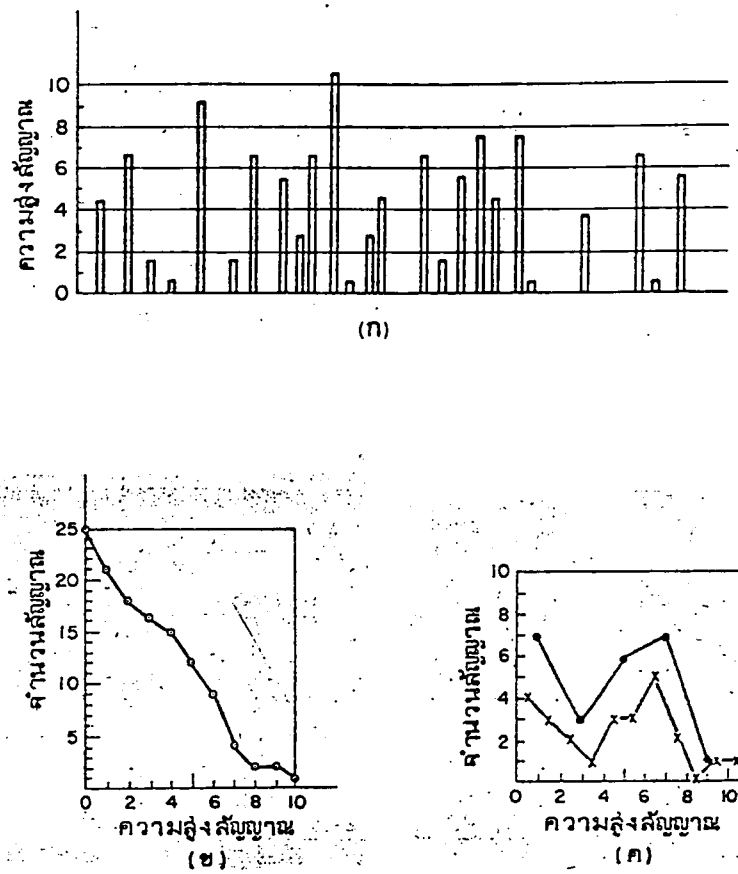
สัญญาณเอาท์พุทของหัววัดรังสีนั้นอาจถูกกำหนดลักษณะสมบัติโดยช่วงเวลาการเกิดประจุ หรือช่วงเวลาการเก็บประจุ ไรส์ไทม์ของส่วนขยายสัญญาณควรมีค่าสั้นกว่าช่วงเวลาจับเก็บประจุของหัววัดรังสี หน้าที่สำคัญของส่วนขยายสัญญาณคือ

- ขยายสัญญาณ
- จัดรูปร่างสัญญาณ

3. ส่วนวิเคราะห์สัญญาณ (pulse height analyzer)

สัญญาณที่ส่งจากหัววัดรังสีไปยังส่วนขยาย จะถูกส่งต่อไปยังส่วนวิเคราะห์ความสูงของสัญญาณ เพื่อทำการเลือกเฉพาะสัญญาณที่เกิดจากพลังงานของรังสีที่ต้องการจะวัด ส่วนที่เกิดจากรังสีอื่นถือว่าเป็นรังสีรบกวนหรือที่เรียกว่า "แบ็คกราวนด์" (background) ซึ่งอุปกรณ์ส่วนนี้จะกำจัดออกไป ตัวอย่างดังแสดงในรูปที่ 2.4 เป็นสัญญาณที่มีความสูงต่างๆ

จากค่าในรูปที่ 2.4 (ก) จะเห็นได้ว่ามีสัญญาณสูงกว่า 0 อยู่จำนวน 25 สูงกว่า 1 จำนวน 21 ฯลฯ ซึ่งทำการพล็อตค่าลงบนแกน xy จะได้ลักษณะกราฟดังแสดงในรูปที่ 2.4 (ข) วิธีการนี้เรียกว่าการพล็อตแบบรวมหรือแบบ "อินทิกรัล" (integral) ถ้านำค่าจากรูปที่2.4 (ข) ลบกันแบบจุดต่อจุดจะได้กราฟเป็นสเปคตรัมดังแสดงในรูปที่ 2.4 (ค) เรียกวิธีดังกล่าวว่าเป็นแบบ "ดิฟเฟอเรนเชียล" (differential) กล่าวคือเป็น



รูปที่ 2.4 แสดงสเปคตรัมของสัญญาณ

(ก) ส่วนขยายสัญญาณขาออก แสดงสัญญาณที่ถูกวิเคราะห์

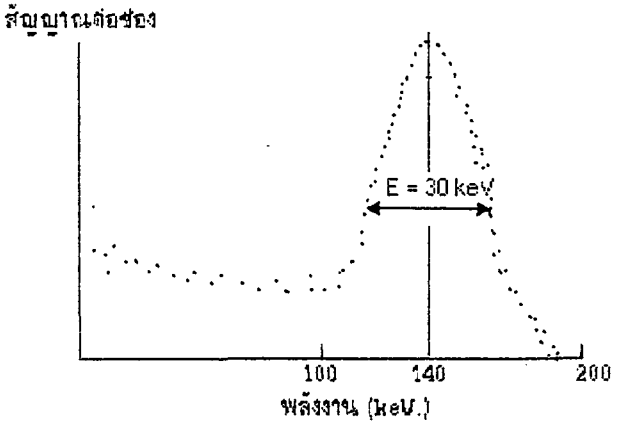
(ข) พล็อตแบบรวมค่าแอมพลิจูดในสเปคตรัมของสัญญาณ

(ค) พล็อตแบบ "ดิฟเฟอเรนเชียล" โดยใช้ค่าแอมพลิจูดของสัญญาณเดียวกันกับข้อ (ข) x แทนสัญญาณสูง 1 หน่วยและ . แทนสัญญาณสูง 2 หน่วย

การแยกบันทึกค่าพลังงานของรังสีที่ปรากฏอยู่ในแต่ละช่วงคลื่น โดยทั่วไปในทางปฏิบัติมักนิยมนับวัดปริมาณรังสีแบบดิฟเฟอเรนเชียล แอมพลิจูดของสัญญาณแสดงถึงพลังงานของรังสีที่มีอยู่ในช่วงคลื่นชนวนหนึ่งซึ่งสังเกตได้โดยง่ายจากสเปคตรัม ระบบการวัดแบบนี้ต้องอาศัยตัวกำจัดสัญญาณ 2 ตัว เรียกว่าตัวกำจัดสัญญาณต่ำกว่า (E_1) (lower discriminator)

และตัวกำจัดสัญญาณสูงกว่า (E_2) (upper discriminator) มีอินพุทต่อขนานกันและเอาท์พุทของตัวกำจัดสัญญาณแต่ละตัวจะต่อกับวงจร "แอนติโคอินซิเดนซ์" (anticoincidence) สมมติมีสัญญาณที่สูงเกินระดับพลังงาน ดังนั้น สัญญาณดังกล่าวจะผ่านตัวกำจัดสัญญาณทั้งคู่ สัญญาณเอาท์พุทจะเดินทางไปถึงตำแหน่งทำการบันทึกค่าในเวลาเดียวกัน ซึ่งจะถูกกำจัดโดยวงจรแอนติโคอินซิเดนซ์ความแตกต่างของระดับทั้งสองเรียกว่า "window" ใช้สัญญลักษณ์ ΔE

มัลติแชนแนลอะนาลัยเซอร์ (multichannel analyser) สามารถใช้หาสเปคตรัมของรังสีได้รวดเร็วกว่าแบบชิงเกิลแชนแนลอะนาลัยเซอร์ (single channel analyser) โดยแบ่งสัญญาณออกเป็น 256 - 512 กรุ๊ป ถ้าวัดพลังงานของรังสีแกมมา 140 keV จาก ^{99m}Tc ด้วยหัววัด NaI(Tl) จะได้รูปการกระจายของมัลติแชนแนลอะนาลัยเซอร์ดังแสดงในรูปที่ 2.5 พลังงานสูงสุดที่พีคมีค่า 140 keV สัญญาณจากพลังงานต่ำมักเนื่องจากอิเลคตรอนซึ่งได้จากการชนกันแบบคอมป์ตัน (Compton) ซึ่งโฟตอนที่กระเจิง



รูปที่ 2.5 ตัวอย่างสเปคตรัมจากหัววัดแบบซินทิลเลชั่น ต้นกำเนิดรังสีคือ ^{99m}Tc ซึ่งสลายตัวให้รังสีแกมมา 140 keV ความกว้างที่ครึ่งความสูงของพีคพลังงานคือ 30keV

(scattered) สามารถหลุดพ้นจากผลึกได้ อย่างไรก็ตามเมื่อรังสีแกมมา 140 keV ถูกดูดกลืนในผลึก พลังงานดังกล่าวจะทำให้เกิดอิเลคตรอนที่โฟโตคาโทดปริมาณไม่เท่ากันสม่ำเสมอ เป็นสาเหตุให้พีคพลังงานกว้างออก เหตุผลดังกล่าวจึงเป็นขีดความสามารถในการ

แยกพลังงานต่างกันจากต้นกำเนิดรังสีของหัววัด NaI(Tl) ซึ่งเรียกว่า "เรสโซลูชั่น" (resolution) หาค่าโดยใช้ความสัมพันธ์

$$\text{resolution} = \frac{\Delta E}{E} \times 100\%$$

โดย ΔE คือความกว้าง ณ ตำแหน่งสูงครึ่งหนึ่งของพีค (full width at half maximum, fwhm)

E คือพลังงานที่พีค

4. แรงดันไฟฟ้าขนาดสูง

แอมพลิจูดของสัญญาณเอาท์พุทของแต่ละหัววัดขึ้นกับค่าแรงดันไฟฟ้าขนาดสูงที่ไบแอสในระบบ ดังนั้น ค่าดังกล่าวจึงนับเป็นส่วนสำคัญของระบบอิเลคทรอนิคส์ที่สมบูรณ์เพื่อใช้ในการขยายและวิเคราะห์สัญญาณ สำหรับหัววัดแบบซินทิลเลชั่นจะมีผลให้แอมพลิจูดของสัญญาณเอาท์พุทเปลี่ยนแปลงไป 10% (หรือมากกว่า) ถ้าไบแอสด้วยค่าเปลี่ยนแปลงไป 1% ส่วนหัวแบบไกเกอร์-มุลเลอร์ และ แบบสารกึ่งตัวนำจะเกิดผลเปลี่ยนแปลงเพียงเล็กน้อยเท่านั้น

แฟคเตอร์ที่มีส่วนเกี่ยวข้องต่อเสถียรภาพของแรงดันไฟฟ้าขนาดสูงนั้นมีอยู่มากมาย เช่น เวลา, อุณหภูมิ, แรงดันไฟฟ้าเข้า และโหลดขาออก หัววัดแบบซินทิลเลชั่นมักถูกสร้างโดยปรับแรงดันขนาดสูงให้มีค่าสม่ำเสมอ ดังนั้นแฟคเตอร์สำคัญที่มีผลต่อเสถียรภาพของแรงดันขาออกคือเวลาและอุณหภูมิ โดยทั่วไปผู้สร้างมักกำหนดค่าเสถียรภาพของแรงดันขาออกจะไม่มีการเปลี่ยนแปลงตลอดช่วงเวลา 24 ชั่วโมง เมื่ออุณหภูมิมีการเปลี่ยนแปลงประมาณ 10 องศาเซลเซียส

5. เครื่องแสดงผล (indicator)

เป็นเครื่องมือแสดงปริมาณรังสีที่นับวัดได้ อาจแสดงเป็นตัวเลข, กราฟ, ตัวเลขพิมพ์บนกระดาษ หรือภาพ ชื่อเรียกมีแตกต่างกัน เช่น สเกลเลอร์ (scaler), เรตมิเตอร์ (ratemeter), พรินต์เตอร์ (printer) และเครื่องบันทึก (recorder) ซึ่งมีทั้งแบบกราฟ, เทปแม่เหล็ก, ทีวี, ออสซิลโลสโคป (oscilloscope) และคอมพิวเตอร์ (computer)

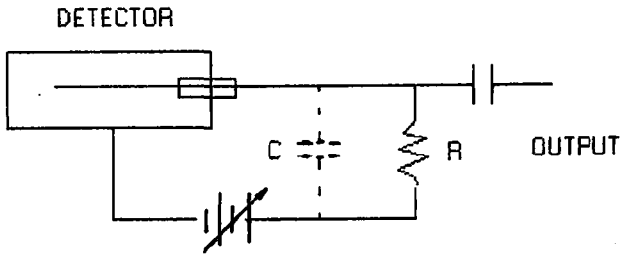
6. หัววัดรังสี

หัววัดรังสีเป็นส่วนสำคัญของระบบนับวัดปริมาณรังสี ดังนั้นการศึกษาเพื่อทำความเข้าใจกับกลไกของหัววัดรังสีจึงนับเป็นประโยชน์อย่างยิ่ง โดยในที่นี้จะขอกล่าวถึง

รายละเอียดและกลไกของหัววัดรังสีชนิดต่างๆ ซึ่งแยกเป็นหัววัดแบบใช้กาซ (gas-filled detectors), หัววัดแบบสารกึ่งตัวนำ (semiconductor detectors) และหัววัดแบบซินทิลเลชั่น (scintillation detectors) โดยที่หัววัดแบบใช้กาซยังได้แยกออกเป็นหัววัดไอออไนเซชั่นแชมเบอร์ (ionization chamber), หัววัดแบบสัดส่วน (proportional counter) และหัววัดแบบไกเกอร์-มุลเลอร์ (Geiger-Müller counter)

6.1 หัววัดรังสีแบบใช้กาซ (gas-filled detectors)

ในรูปที่ 2.6 จะแสดงถึงวงจรพื้นฐานอย่างง่ายๆ ซึ่งประกอบด้วยแหล่งกำเนิดแรงดันไฟฟ้าขนาดสูงที่เปลี่ยนค่าได้ (V), ความต้านทานที่มีค่าสูง (R), หัววัดรังสี



รูปที่ 2.6 วงจรพื้นฐานสำหรับหัววัดรังสีแบบใช้กาซ

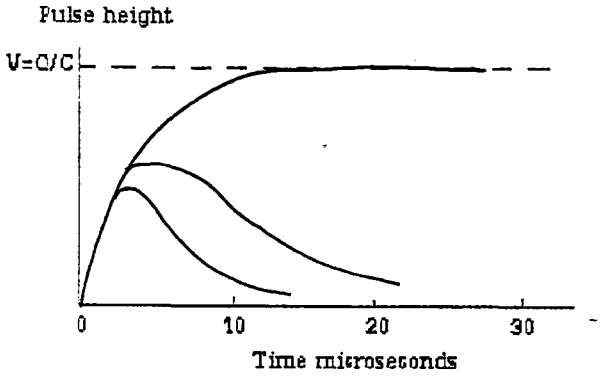
บรรจุด้วยกาซ (D) ภายในมีลวดนำไฟฟ้า 2 เส้นซึ่งแยกออกจากกันด้วยฉนวนต้านทานไฟฟ้าอย่างดี เส้นที่หนึ่งจะอยู่ตรงกลางหลอด อีกเส้นจะเชื่อมติดกับแผ่นโลหะทรงกระบอกติดผนังหลอดรังสี C จะเป็นตัวเก็บประจุ (capacitor) เมื่อรังสีวิ่งผ่านหัววัดรังสี จะทำให้กาซที่อยู่ภายในเกิดการแตกตัวเป็นประจุไฟฟ้าเรียกว่าไอออน (ions) เมื่อประจุบวกวิ่งไปยังขั้วคาโทด, ประจุลบวิ่งไปยังขั้วอาโนด จะทำให้เกิดกระแสไฟฟ้าไหลในวงจร

ค่าไทม์คอนสแตนท์ (time constant , RC) ของวงจรมีค่ามากกว่าเวลาที่หัววัดรังสีจับไอออนที่เกิดขึ้นมาทั้งหมด, จะได้ขนาดของสัญญาณ (voltage pulse)

$$V = \frac{Q}{C}$$

เมื่อ Q = จำนวนประจุทั้งหมดที่ถูกจับ

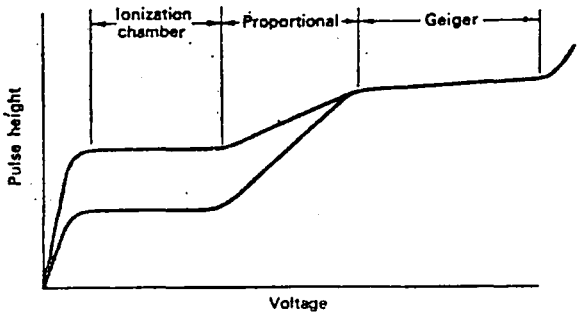
C = ค่าความจุ (capacitance) ของวงจร



รูปที่ 2.7 รูปร่างของสัญญาณขึ้นกับค่าไทม์คอนสแตนท์ของวงจรสำหรับหัววัดรังสีกราฟบน-สำหรับกรณีที่ RC = ∞, กราฟกลางสำหรับกรณีที่ RC มีค่าน้อยกว่าเวลาที่ใช้จับไอออนและกราฟล่างสุดสำหรับกรณีที่ RC มีค่าน้อยกว่าเวลาที่ใช้จับไอออนมากๆ

รูปร่างของสัญญาณดังรูปที่2.7 (รูปกราฟบนสุด) ปรากฏตรงเอาท์พุทของวงจรวัดรังสี ถ้าค่าไทม์คอนสแตนท์มีค่าน้อยกว่าเวลาที่ใช้ในการจับไอออนทั้งหมด ความสูงของสัญญาณจะมีค่าน้อย แต่สัญญาณมีค่าแคบมาก ซึ่งแสดงดังในกราฟรูปที่ 2.7 ลักษณะสัญญาณเช่นนี้เรียกว่า "clipping" ผลทำให้สัญญาณแยกจากกันและนับแยกกันได้

6.1.1 หัววัดไอออไนเซชั่นแชมเบอร์ (ionization chamber)



รูปที่ 2.8 กราฟของความสูงสัญญาณกับค่าแรงดันไฟฟ้าคร่อมระหว่างหัววัดรังสีแบบกาซ, แสดงถึงบริเวณไอออไนเซชั่นแชมเบอร์, บริเวณสัดส่วน และบริเวณไกเกอร์

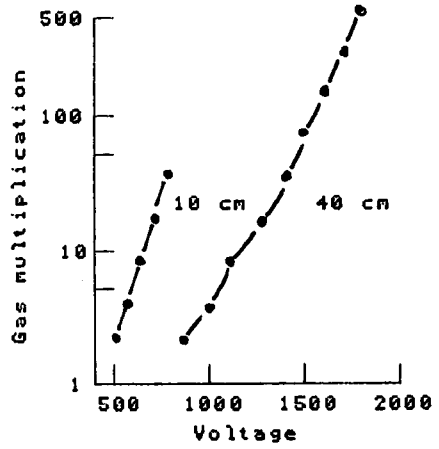
ถ้าเราให้ลำรังสีผ่านหัววัดรังสีด้วยปริมาณคงที่, ให้แรงดันไฟฟ้า, V, ซึ่งสามารถเปลี่ยนแปลงได้ โดยเริ่มตั้งแต่ 0 โวลต์และค่อยๆ เพิ่มมากขึ้น เราจะได้บริเวณที่สำคัญในการวัดรังสีหลายบริเวณดังรูปที่ 2.8 บริเวณแรกมีชื่อเรียกว่าบริเวณไอออไนเซชั่นแชมเบอร์ (ionization chamber region) ที่บริเวณนี้ไอออนบวกที่เกิดขึ้นมาทั้งหมดจะถูกจับโดยคาโทดที่อยู่รอบๆ ผนังหลอด และไอออนลบหรืออิเลคตรอนจะถูกจับโดยอาโนดที่อยู่ตรงกึ่งกลางของหลอด ค่าแรงดันไฟฟ้าช่วงนี้ไม่สามารถทำให้ไอออนเหล่านี้เคลื่อนที่ด้วยความเร่งได้ ดังนั้นไม่ว่าเราจะเพิ่มค่าแรงดันไฟฟ้าในบริเวณนี้เท่าใดก็ตาม ขนาดของสัญญาณที่เกิดขึ้นมาก็มีค่าคงที่และไม่เกิดไอออนทุติยภูมิ (secondary ionization) เนื่องจากไอออนไปชนกับผนังหลอด จำนวนของอิเลคตรอนที่ถูกจับโดยอาโนดจะมีค่าเท่ากับจำนวนของอนุภาคที่เกิดการแตกตัวครั้งแรก (primary ionizing particle) ขนาดของสัญญาณไม่ขึ้นกับค่าแรงดันไฟฟ้า แต่จะขึ้นอยู่กับจำนวนของไอออนที่เกิดขึ้นมา (ในช่วงระยะที่ลำรังสีผ่านเข้าไปในหัววัดรังสี) เพียงอย่างเดียว

จากความจริงที่ว่าขนาดของสัญญาณในช่วงบริเวณไอออไนเซชั่นแชมเบอร์ขึ้นอยู่กับจำนวนของไอออนที่เกิดขึ้นในหัววัดรังสี ทำให้เราสามารถแยกรังสีต่างๆ ที่มีค่าไอออไนเซชั่นจำเพาะ (specific ionization) แตกต่างกัน เช่น รังสีอัลฟา, รังสีเบตา หรือรังสีแกมมา

6.1.2 หัววัดแบบสัดส่วน (proportional counter)

หัววัดรังสีชนิดนี้ ใช้แรงดันไฟฟ้าระหว่างขั้วคาโทดและอาโนดต่อจากค่าแรงดันไฟฟ้าในช่วงบริเวณไอออไนซ์เซชั่นแชมเบอร์ (ประมาณ 300 - 750 โวลต์) เพื่อต้องการให้สัญญาณมีค่าสูงขึ้นเข้าสู่วงจรสเกเลอร์ และพัฒนาขนาดของสัญญาณซึ่งขึ้นอยู่กับการไอออไนซ์ (ionize) (จุดประสงค์เพื่อแยกรังสีที่มีความแตกต่างกัน) ในขณะที่แรงดันไฟฟ้าเพิ่มมากกว่าในช่วงบริเวณไอออไนเซชั่นแชมเบอร์ จะทำให้ไอออนวิ่งด้วยความเร่งชนขั้วคาโทดทำให้เกิดอิเลคตรอนทุติยภูมิ (secondary electrons) ขึ้นมา แรงดันไฟฟ้าค่านี้จะเป็นค่าเริ่มต้นในบริเวณที่เรียกว่า บริเวณสัดส่วน ค่าแรงดันไฟฟ้าคร่อมระหว่างตัวต้านทาน, R , ในช่วงบริเวณสัดส่วนนี้จะมีค่ามากกว่าในช่วงของบริเวณไอออไนเซชั่นแชมเบอร์ ทั้งนี้เนื่องจากการเพิ่มขึ้นของอิเลคตรอนนั่นเอง ค่า "gas multiplication factor" จะมีค่ามากกว่า 1 การทวีคูณของไอออนในก๊าซจะมีชื่อเรียกว่าอะวาลานช์ (avalanche) การเพิ่มค่าแรงดันไฟฟ้าเป็นสาเหตุการเพิ่มค่าอะวาลานช์โดยการแผ่ออกจากอาโนด โดยที่ขนาดของสัญญาณแรงดันไฟฟ้าที่เกิดขึ้นมา เราพิจารณาได้จากจำนวนของอิเลคตรอนที่ถูกจับโดยอาโนด ขนาดของสัญญาณที่ออกจากหัววัดรังสีจะเป็นปฏิภาคโดยตรงกับค่าความสูงของ

แรงดันไฟฟ้าระหว่างขั้วหลอดทั้งสองของหัววัดรังสี นอกจากนั้นค่าความสูงของแรงดันไฟฟ้าระหว่างขั้วของหลอดวัดรังสีจะทำให้ค่า gas multiplication ขึ้นอยู่กับเส้นผ่าศูนย์กลางของอิเลคโทรดที่จับอิเลคตรอน (เนื่องจากความเข้มของสนามไฟฟ้าที่อยู่ใกล้อาโนดจะมีค่า



รูปที่ 2.9 กราฟแสดงค่า gas multiplication กับค่าแรงดันไฟฟ้าสำหรับความกดดัน 10 และ 40 ซม. ปรอทในถังอาร์กอน, เส้นผ่าศูนย์กลางของอาโนด = 0.01 นิ้ว, เส้นผ่าศูนย์กลางของคาโทด = 0.87 นิ้ว

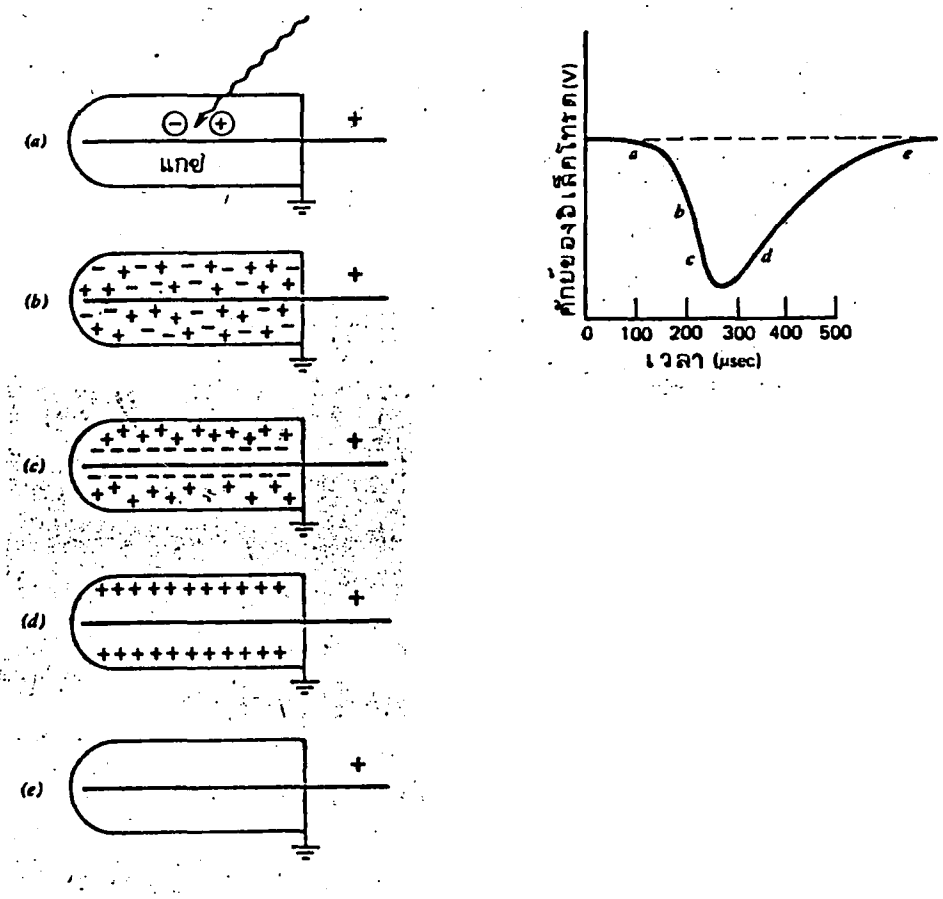
ลดลง) และความกดดันของกาซภายในหลอดรังสี (ความกดดันของกาซลดลงทำให้ค่า gas multiplication มีค่าเพิ่มขึ้น) ดังรูปที่ 2.9

เนื่องจาก ขนาดของสัญญาณที่ออกมาจากหัววัดรังสีจะเป็นปฏิภาคโดยตรงกับค่าแรงดันไฟฟ้าระหว่างขั้วคาโทดและอาโนด เพราะฉะนั้นหัววัดรังสีแบบนี้จะต้องมีแหล่งจ่ายไฟฟ้าแรงสูงที่คงที่มากๆ

6.1.3 หัววัดแบบไกเกอร์-มุลเลอร์(Geiger-Müller counter)

ถ้าเราเพิ่มค่าแรงดันไฟฟ้าขึ้นไปอีกต่อจากย่านสัดส่วน (ประมาณ 800 - 1500 โวลต์) จะเป็นย่านที่เรียกว่าย่านไกเกอร์ (geiger region) ที่จุดนี้ค่าสัญญาณแรงดันไฟฟ้ามีขนาดเท่ากันหมด ดังนั้นจึงไม่สามารถแยกชนิดของรังสีได้ หัววัดชนิดนี้จะใช้ลวดขนาดเล็กเป็นขั้วไฟฟ้า กระแสที่มิได้เกิดจากการแตกตัวในหัววัดจะมีค่าต่ำมากเนื่องจากมีความดันอากาศต่ำมาก (ประมาณ 1/75 บรรยากาศ)

เมื่อมีอนุภาคหรือโฟตอนวิ่งเข้าไปในหัววัด จะทำให้เกิดการแตกตัวโดยมี



รูปที่ 2.10 (a) หัววัดไกเกอร์-มุลเลอร์ประกอบด้วยกาซซึ่งถูกทำให้แตกตัวโดยรังสีแกมมา

(b) อิเลคโทรดศูนย์กลางซึ่งมีแรงดันบวกสูง (เช่น 1000 โวลต์) ถูกชนโดยอิเลคตรอน ซึ่งจะได้รับพลังงานเพิ่มในการทำให้กาซแตกตัวจนกระทั่งทั้งปริมาตรประกอบไปด้วยคู่ไอออน

(c) เป็นช่วงที่อิเลคตรอนถูกจับโดยเร็ว แรงดันของอิเลคโทรดศูนย์กลางลดลงดังแสดงในกราฟ

(d) ไอออนบวกซึ่งหนักและเคลื่อนที่ช้าๆ ไปยังผนังด้านนอก

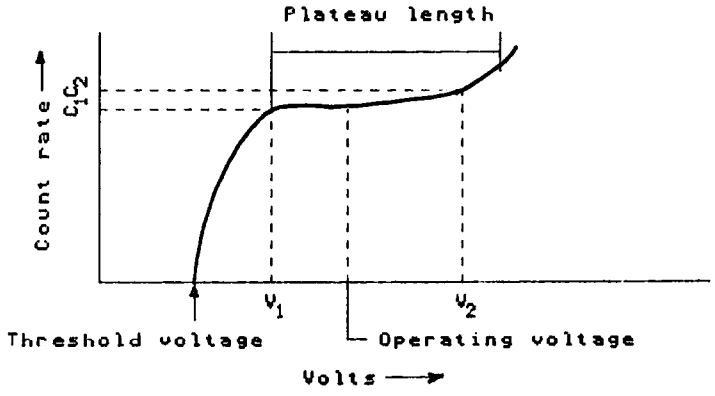
(e) หลังจากขบวนการในรูป (d) ผ่านไปประมาณ 400 μsec หัววัดพร้อมที่จะเริ่มต้นขบวนการใหม่จาก (a) - (e) นั่นเอง

ขั้นตอนของขบวนการเช่นเดียวกับหัววัดแบบสัดส่วน แต่การทวีคูณของอิเลคตรอนมีค่ามากกว่าจนทำให้เกิดการแตกตัวของกาซทั้งหมดในแชมเบอร์ ดังนั้น สัญญาณแรงดันไฟฟ้าขาออกจึง

มีค่ามากและมักจะมีความสูงเท่ากับการเกิดการแตกตัว (ณ ค่าแรงดันไฟฟ้าที่ป้อนให้ค่าหนึ่งตามที่กำหนดไว้) สัญญาณดังกล่าวจะมีค่าคงที่และเป็นอิสระต่อพลังงานของอนุภาคหรือโฟตอนที่วิ่งไปชน ทำให้การวิเคราะห์และบันทึกค่าทางอิเลคทรอนิกส์เป็นไปโดยไม่ยุ่งยาก ซึ่งแสดงแผนภาพดังรูปที่2.10

จากการพล็อตกราฟระหว่างค่าแรงดันไฟฟ้ากับค่าอัตราการนับรังสี จะได้กราฟดังรูปที่ 2.11 ช่วงที่ราบมีชื่อเรียกว่า "พลาโต" (plateau) ประสิทธิภาพของเครื่องมือชนิดนี้ขึ้นอยู่กับความยาวของพลาโต , ความลาดชันของพลาโต (the slope of the plateau) และค่าริซอลวิงไทม์ (resolving time , จะขอกล่าวในหัวข้อต่อไป) ซึ่งค่าความลาดชันของพลาโต ปกติกำหนดในรูปของเปอร์เซ็นต์ของอัตราการนับที่เพิ่มขึ้นต่อ 100 โวลต์ จะได้ว่า

$$\text{ค่าความลาดชัน} = \frac{(C_2-C_1)/C_1}{0.01(V_2-V_1)}$$



รูปที่ 2.11 กราฟแสดงลักษณะเฉพาะการทำงานหัววัดรังสีแบบไกเกอร์-มุลเลอร์

สำหรับหัววัดไกเกอร์-มุลเลอร์นั้นจะมีค่าความลาดชันประมาณ 3% ต่อ 100 โวลต์ ปกติค่าแรงดันไฟฟ้าที่ใช้ (operating voltage) มีค่าประมาณ 1/4 ถึง 1/2 ของช่วงความยาวพลาโต

6.1.3.1 การเควนชิง (quenching)

เมื่อไอออนบวกถูกจับด้วยขั้วคาโทดที่อยู่รอบๆ ผนังหลอด ไอออนบวกมีพลังงานมากมันจะไปชนผนังหลอด พลังงานส่วนใหญ่จะถ่ายเทให้กับผนังหลอดในรูปของความ

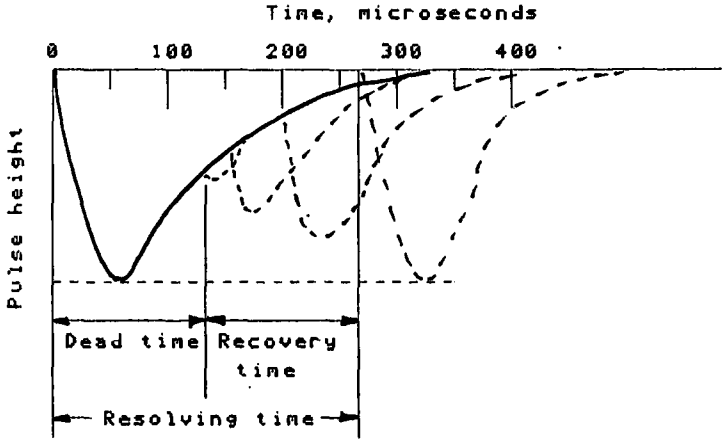
ร้อน บางส่วนของพลังงานจลน์นั้นจะไปกระตุ้น (excite) อะตอมในผนังหลอด อะตอมเหล่านี้เมื่อถูกกระตุ้นและกลับเข้าสู่สภาวะปกติ (ground state) จะคายพลังงานออกมาในรูปของรังสีอุลตราไวโอเลต (U.V. photon) ในขณะเดียวกัน ขั้วอาโนดก็จะสร้างสนามไฟฟ้าขึ้นมาใหม่ (หลังจากจับอิเลคตรอนแล้ว) การเกิดปฏิกิริยาของรังสีอุลตราไวโอเลตกับกาซที่อยู่ในหัววัดรังสี บางทีเป็นการเริ่มต้นเกิดอะวาลานช์ และจากสาเหตุนี้ทำให้เกิดการนับที่ไม่ถูกต้อง เพื่อป้องกันการเกิดการนับที่ไม่ถูกต้องเรียกว่า "quenching" ซึ่งวิธีการ quenching อาจทำได้โดยใช้วงจรอิเลคทรอนิคส์ หรือวิธีการทางเคมีโดยใส่กาซ (เรียกว่า self-quenching gas) เข้าไปในหลอดวัดรังสี กาซนี้จะดูดกลืนรังสีอุลตราไวโอเลตโดยไม่ทำให้ตัวมันเองเกิดการไอออไนซ์ วิธีการหนึ่งที่ใช้คือการเติมธาตุออแกนิค (organic) เช่น อัลกอฮอล์ หรืออีเธอร์ ลงในหลอดรังสีเพียงเล็กน้อย พลังงานของรังสีอุลตราไวโอเลตจะทำให้โมเลกุลของธาตุออแกนิคแยกออกจากกัน ดังนั้นพลังงานของรังสีอุลตราไวโอเลตก็จะถูกทำลายหมดไป

6.1.3.2 ริซอลวิงไทม์ (resolving time)

ถ้ามีอนุภาครังสี 2 ตัว เข้าสู่หัววัดในเวลาไล่เลี่ยกัน อะวาลานช์ของไอออนจากอนุภาคตัวแรกจะทำให้เครื่องนับรังสีเกิดอัมพาต เป็นผลทำให้เครื่องวัดรังสีไม่สามารถตอบสนองต่ออนุภาคตัวที่ 2 ได้ เพราะว่าความเข้มสนามไฟฟ้าจะมีค่ามากที่สุดที่บริเวณผิวอาโนด อะวาลานช์ของการเกิดไอออไนเซชั่น เริ่มต้นอยู่ใกล้ชิดกับอาโนดและจะแผ่ไปตามความยาวของอาโนด ไอออนลบจะวิ่งตรงไปยังขั้วอาโนดในขณะที่ไอออนบวกจะวิ่งไปที่ขั้วคาโทด ไอออนลบหรืออิเลคตรอนจะเกิดการเคลื่อนที่ด้วยความเร็วมากและจะถูกคาโทดจับไว้ในขณะที่ไอออนบวกมีขนาดใหญ่เคลื่อนที่ช้า เมื่อเทียบกับอิเลคตรอนแล้ว คาบเวลาที่ขั้วคาโทดจับจะมีค่ามากกว่าคาบเวลาที่อิเลคตรอนถูกอาโนดจับ ไอออนบวกที่เคลื่อนที่ช้าเหล่านี้จะรวมตัวกันเป็นปลอกรอบๆ อาโนด จากสาเหตุนี้เองความเข้มของสนามไฟฟ้ารอบๆ อาโนดจะมีค่าลดลงอย่างมากมาย และมันจะทำให้อนุภาคอื่นไม่มีโอกาสที่จะทำให้เริ่มต้นเกิดอะวาลานช์ ขณะที่ปลอกไอออนบวกเคลื่อนที่ไปยังขั้วคาโทด ความเข้มสนามไฟฟ้ามีค่ามากขึ้น, จนถึงจุดนี้อะวาลานช์ที่เกิดจากอนุภาคอื่นจึงจะเริ่มต้นเกิด เวลาที่ต้องการใช้เพื่อให้เกิดความเข้มสนามไฟฟ้าใหม่เรียกว่า "เดดไทม์" (dead time) ภายหลังจากที่หมดเดดไทม์แล้ว อะวาลานช์อื่นสามารถเริ่มเกิดได้ แต่สัญญาณที่เกิดขึ้นมาของอะวาลานช์ยังมีขนาดเล็ก ความเข้มของสนามไฟฟ้ามีค่าไม่มากเพียงพอที่จะทำให้เกิดสัญญาณการนับรังสีได้ และขณะที่ไอออนบวกยังคงเคลื่อนที่อยู่นั้นอนุภาคอื่นๆ ทำให้กาซแตกตัวเป็นสัญญาณมีขนาดใหญ่ขึ้น เมื่อสัญญาณเอาท์พุท (output pulse,ตัวแรก) ผ่านวงจรดิสคริมิเนเตอร์และเครื่อง

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

นับเรียบร้อยแล้ว เราจะพูดได้ว่าเครื่องนับรังสีจะอยู่ในระยะพักฟื้นจากการเกิดไอออไนเซชั่นจากอนุภาคตัวแรก และช่วงระยะเวลาระหว่างเดดไทม์กับเวลาที่เครื่องวัดรังสีใช้ในการพักฟื้นอย่างเต็มที่เรียกว่า "ริโคเวอรีไทม์" (recovery time) โดยผลบวกของค่าเดดไทม์กับริโคเวอรีไทม์จะถูกเรียกว่า "ริซอลวิงไทม์" ซึ่งเราอาจจะให้คำนิยามริซอลวิงไทม์ได้ว่า "เป็นเวลาที่น้อยที่สุดที่เครื่องวัดรังสีจะนับรังสีตัวที่สองที่เข้ามาในเครื่องวัดรังสีต่อจากรังสีตัวแรก" รูปที่2.12 แสดงความสัมพันธ์ระหว่างเดดไทม์, ริโคเวอรีไทม์และริซอลวิงไทม์ ค่าริซอลวิงไทม์ของเครื่องวัดรังสีแบบไกเกอร์-มุลเลอร์จะอยู่ในช่วง 100 μsec หรือมากกว่า



รูปที่ 2.12 กราฟแสดงความสัมพันธ์ระหว่างเดดไทม์, ริโคเวอรีไทม์และริซอลวิงไทม์

6.2 หัววัดรังสีแบบสารกึ่งตัวนำ (semiconductor detectors)

สาเหตุที่มีการใช้หัววัดรังสีแบบของแข็งแทนกาซ (เช่น ไอออไนเซชั่นแชมเบอร์ดังอธิบายไว้ข้างต้น) น่าจะเป็นไปได้เนื่องจากเหตุผลสองประการดังนี้คือ

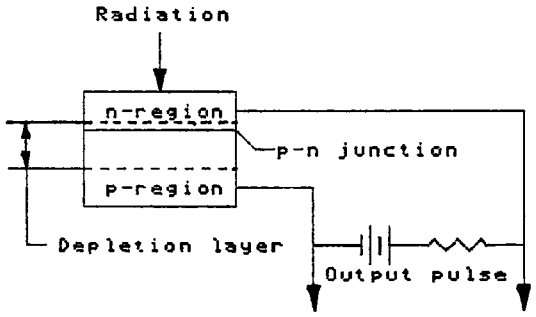
1. ของแข็งมีคุณสมบัติในการสกัดกั้นและสามารถดูดเก็บพลังงานของอนุภาครังสีหรือโฟตอนได้

2. พลังงานเฉลี่ยที่ใช้ในการทำให้เกิดการแตกตัว (เป็นไอออนบวกและลบ)

หนึ่งหน่วย ในกาซมีค่าประมาณ 30 eV แต่ในของแข็งต้องการพลังงานเพื่อทำให้เกิดอิเลคตรอน-โฮลหนึ่งคู่ (กล่าวคือ เกิดอิเลคตรอนอิสระซึ่งเป็นผลให้เกิดช่องว่างเรียกว่า"โฮล") ประมาณ 3 eV จึงเห็นได้ว่า ถ้าให้พลังงานเท่ากัน ปริมาณรังสีที่เกิดจากการแตกตัวใน

ของแข็งจะมีมากกว่าในกาซถึง 10 เท่า ซึ่งทำให้ง่ายต่อการบันทึกค่า ทั้งยังทำให้ค่าทางสถิติถูกต้องขึ้นอีกด้วย

สารกึ่งตัวนำที่กล่าวถึงนี้ ได้แก่สารที่มีคุณสมบัติในการนำไฟฟ้าอยู่ระหว่างสารตัวนำไฟฟ้า (good conductor) กับฉนวนไฟฟ้า (insulator) ถึงแม้ว่าจะมีสารเป็นจำนวนมากที่เราสามารถจัดเป็นพวกสารกึ่งตัวนำ แต่สารกึ่งตัวนำที่ใช้กันมากและรู้จักกันดีได้แก่ ซิลิคอน (silicon) และเยอรมันเนียม (germanium) เครื่องมือวัดรังสีชนิดนี้มีส่วนประกอบดังในรูปที่ 2.13 บริเวณเอ็น (n-region) หรือมีชื่อเรียกอีกอย่างหนึ่งว่า "สารกึ่งตัวนำชนิดเอ็น" (n-type semiconductor) หมายถึงสารกึ่งตัวนำที่มีอิเลคตรอนเป็นจำนวนมากเกินพอ และบริเวณพี (p-region) หรือเรียกว่า "สารกึ่งตัวนำชนิดพี" (p-type semiconductor) ซึ่งหมายถึงสารกึ่งตัวนำที่มีจำนวนโฮล (hole) มากเกินพอ



รูปที่ 2.13 ภาพแสดงรอยต่อของหัววัดรังสีแบบสารกึ่งตัวนำ เครื่องวัดรังสีนี้ใช้ได้ดีสำหรับวัดอิเลคตรอนหรืออนุภาคที่มีประจุอื่นๆ สำหรับการวัดรังสีแกมมาจะมีอิเลคตรอนเรดิเอเตอร์ (eletron radiator) เข้ามาแทรกระหว่างต้นกำเนิดรังสีและหัววัดรังสี , โฟโตและคอมป์ตันอิเลคตรอน (photo and compton electrons) ที่เกิดขึ้นมาจะถูกวัดโดยหัววัดรังสี

ถ้าเราติดบริเวณพี กับบริเวณเอ็น จะมีชื่อเรียกใหม่ว่า "รอยต่อพี-เอ็น" (p-n junction) และถ้าใส่แรงดันไฟฟ้าให้กับรอยต่อพี-เอ็น แบบฟอร์เวิดไบแอส (for ward bias) กล่าวคือให้บริเวณพีเป็นขั้วบวก และบริเวณเอ็นเป็นขั้วลบ ค่าอิมพีแดนซ์ระหว่างรอยต่อจะมีค่าต่ำมากๆ จะทำให้กระแสไฟฟ้าสามารถไหลผ่านรอยต่อ ถ้าเราใส่แรงดันไฟฟ้ากลับขั้วกันโดยให้บริเวณพีเป็นขั้วลบ และบริเวณเอ็นเป็นขั้วบวก ลักษณะแบบนี้

เรียกว่า "รีเวอส์ไบแอส" (reverse bias) กระแสไฟฟ้าจะไม่สามารถไหลผ่านรอยต่อได้ (ยกเว้นมีกระแสไฟฟ้าน้อยมากๆ ไหลผ่านได้ เนื่องจากความร้อนจะทำให้เกิดโฮลและอิเลคตรอน) บริเวณที่อยู่รอบรอยต่อจะเกิดเป็นที่ว่างขึ้นมาทำให้เกิดแรงดันไฟฟ้าที่แตกต่างกันของบริเวณพีและบริเวณเอ็นบริเวณนี้จะมีชื่อเรียกว่า "depletion layer" และจะเป็นปริมาตรที่ไวที่สุดของเครื่องมือวัดรังสีสารกึ่งนำ เมื่อรังสีวิ่งผ่านหัววัดรังสี (หรือรังสีวิ่งผ่าน depletion layer) จะคายพลังงานให้แก่หัววัดรังสีทำให้อะตอมของหัววัดรังสีเกิดการแตกตัว (ionization) ได้อิเลคตรอนและโฮล อิเลคตรอนจะสามารถวิ่งไปที่ขั้วบวกได้ทำให้เกิดสัญญาณไฟฟ้าขึ้นมา สัญญาณนี้จะถูกส่งต่อไปยังเครื่องนับแสดงผลออกมา

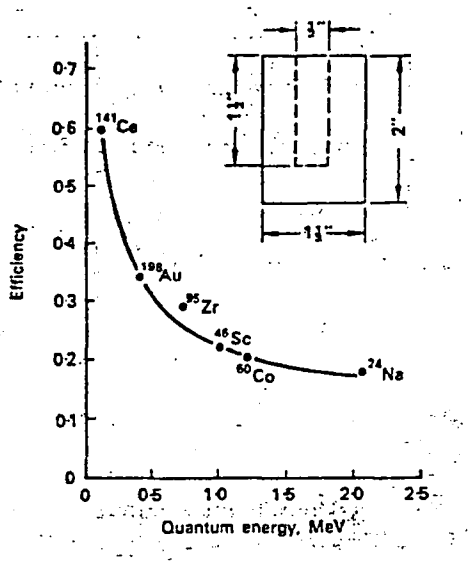
6.3 หัววัดรังสีแบบซินทิลเลชั่น (scintillation counter)

หัววัดชนิดนี้ใช้หลักการเปลี่ยนพลังงานจลน์ของอนุภาครังสี กลายเป็นแสงวาบ (flash light) โดยรัทเธอร์ฟอร์ดได้ทำการทดลองศึกษาการสะท้อนของอนุภาคอัลฟา ใช้ผลึก zinc sulfide เป็นหัววัดรังสี และใช้ตาดูแสงวาบที่ปรากฏขึ้นมา เมื่ออนุภาคชนกับผลึก zinc sulfide ปัจจุบันนี้นักวิทยาศาสตร์ใช้อุปกรณ์ทางอิเลคทรอนิคส์ได้แก่ หลอดโฟโตมัลติพลายเออร์ (photomultiplier tube) รับแสงวาบที่เกิดขึ้นมาแทนการมองด้วยนัยตา หลอดนี้จะไปขยายสัญญาณเอาท์พุทที่ออกมา จำแนกออกเป็นตามขนาดของมันและนับ รังสีต่างชนิดกันอาจจะใช้วัดโดยหัววัดรังสีแบบซินทิลเลชั่น โดยเลือกใช้ผลึกตามชนิดของงานที่ต้องการดังแสดงในตารางที่ 2.1

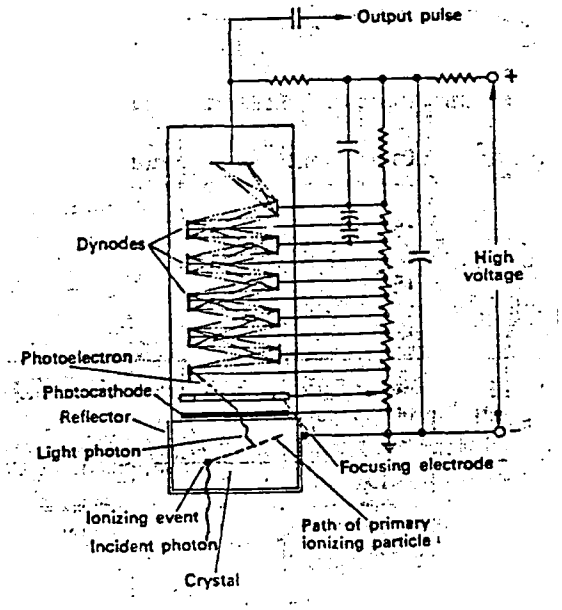
ตารางที่ 2.1 วัตถุที่ใช้ทำหัววัดแบบซินทิลเลชั่น

Phosphor	Density (g/cm^3)	Wavelength of maximum emission, Å	Relative pulse height	Decay time (usec)
NaI (Tl)	3.67	4100	210	0.25
CsI (Tl)	4.51	Blue	55	1.1
KI (Tl)	3.13	4100	50	1.0
Anthracene	1.25	4400	100	0.032
Trans-Stilbene	1.16	4100	60	0.0064
plastic		3550-4500	28-48	0.003-0.005
liquid		3550-4500	27-49	0.002-0.008
p-Terphenyl	1.23	4000	40	0.005

From R. Swank, Characteristics of scintillators, Annual Review of Nuclear Science, vol. V, 1954.



รูปที่ 2.14 ประสิทธิภาพของการวัดรังสีแกมมาพลังงานต่างๆ สำหรับผลึกของหัววัดรังสีแบบซินทิลเลชั่นที่ดี



รูปที่ 2.15 แผนภาพแสดงถึงเหตุการณ์ที่เกิดขึ้นเมื่อตอนผ่านเข้าไปในหัววัดรังสีซินทิลเลชั่น

ปกติหัววัดรังสีชนิดนี้ ใช้ผลึกโซเดียมไอโอไดด์ (sodium iodide) กระตุ้นโดยสารทอลเลียม (thallium) , (Na I (Tl)) , หัววัดรังสีนี้จะอยู่คู่กับหลอดโฟโตมัลติพลายเออร์ สารทอลเลียมนี้จะอยู่ในรูปของสารไม่บริสุทธิ์ (impurity) ในโครงสร้างของผลึกโซเดียมไอโอไดด์ประมาณ 0.2% , มันจะเปลี่ยนพลังงานที่ถูกดูดกลืนในผลึกไปเป็นแสง ผลึกที่มีความหนาแน่นสูงและมีค่าตัวเลขอะตอมสูง จะทำให้เครื่องวัดรังสีมีประสิทธิภาพสูงดังรูปที่ 2.14

เมื่อรังสีแกมมาผ่านเข้าไปในผลึก มันจะไปทำปฏิกิริยาทางนิวเคลียร์กับอะตอมของผลึกโดยขบวนการดูดกลืนโฟโตอิเลคทริค (photoelectric absorption) , ขบวนการสะท้อนคอมป์ตัน (Compton scattering) และขบวนการแพร์โพดักชั่น (pair production) พลังงานของอนุภาคปฐมภูมิที่เกิดจากขบวนการเหล่านี้จะถูกทำให้หายไปโดยการกระตุ้น (exciting) และไอออไนซิง (ionizing) อะตอมในผลึก อะตอมที่อยู่ในสภาวะกระตุ้น (excited) เมื่อกับคืนสู่สภาวะปกติ (ground state)จะปล่อยแสงออกมา แสงเหล่าจะไปชนกับโฟโตคาโทด (photocathode) ของหลอดโฟโตมัลติพลายเออร์ที่มีความไวมาก จะทำให้อิเลคตรอนหลุดออกมาดังนั้นอิเลคตรอนก็จะหลุดออกจากขั้วคาโทด อิเลคตรอนเหล่านี้จะวิ่งด้วยความเร่งไปยังแผ่นอิเลคโทรด (electrode) แผ่นที่ 2 (ซึ่งมีขั้วเป็นบวกเมื่อเทียบกับแผ่นโฟโตคาโทด) แผ่นนี้มีชื่อเรียกว่า "ไดโนด" (dynode) ด้วยแรงดันไฟฟ้าประมาณ 100 โวลต์ แต่ละอิเลคตรอนที่ชนแผ่นไดโนดจะทำให้เกิดอิเลคตรอนหลายตัวหลุดออกจากแผ่นไดโนดนี้ วิธีการนี้เรียกว่า "การทวีคูณ" (multiplying) เมื่ออิเลคตรอนหลุดออกจากแผ่นไดโนดแผ่นที่ 2 มันจะวิ่งด้วยความเร่งไปชนไดโนดแผ่นที่ 3 (มีแรงดันไฟฟ้าสูงกว่าแผ่นที่ 2 และมีแรงดันเป็นบวกเมื่อเทียบกับแผ่นที่ 2) ขบวนการเหล่านี้จะทำให้เกิดซ้ำๆ กันประมาณ 10 ครั้ง ผลสุดท้ายอิเลคตรอนทั้งหมดจะถูกจับโดยขั้วอาโนดของหลอดโฟโตมัลติพลายเออร์ จะทำให้เกิดสัญญาณกระแส (current pulse) ขึ้นมา ขนาดของสัญญาณกระแสจะเป็นปฏิภาคโดยตรงกับค่าพลังงานปฐมภูมิของอนุภาคที่เกิดจากการไอออไนซ์ รูปที่ 2.15 แสดงเหตุการณ์ที่เกิดขึ้นมาเมื่อลำโฟตอนผ่านเข้าไปในหัววัดรังสีชินทิลเลชั่น

6.3.1 ชินทิลเลชั่นสเปคโตรมิเตอร์ (scintillation spectrometer)

ชินทิลเลชั่นสเปคโตรมิเตอร์เป็นเครื่องมือที่แยกขนาดของสัญญาณเอาท์พุทจากหัววัดชินทิลเลชั่น ขนาดของสัญญาณเอาท์พุทที่ออกจากหัววัดชินทิลเลชั่นจะเป็นปฏิภาคโดยตรงกับค่าพลังงานของรังสีแกมมา เครื่องชินทิลเลชั่นสเปคโตรมิเตอร์สามารถจะแสดงให้เราเห็นรูปร่างของเส้นสเปคตร้าจากสารกัมมันตรังสีที่ให้รังสีแกมมา ดังนั้นเครื่องมือ

ชนิดนี้จึงมีประโยชน์ในการวิเคราะห์สารกัมมันตรังสีที่ไม่รู้จัก (unknown isotopes)

บทที่ 3

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

(Single Chip Microcontroller system -51 family Architectural)

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือ ไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดี่ยว เหมาะสำหรับงานควบคุมอุปกรณ์อื่นๆ แบบอัตโนมัติเพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MCS51 อันได้แก่ เบอร์ 8051 และ 8052 ซึ่งมีโครงสร้างและชุดคำสั่งแตกต่างกันเพียงเล็กน้อยดังตารางในรูปที่ 3.1

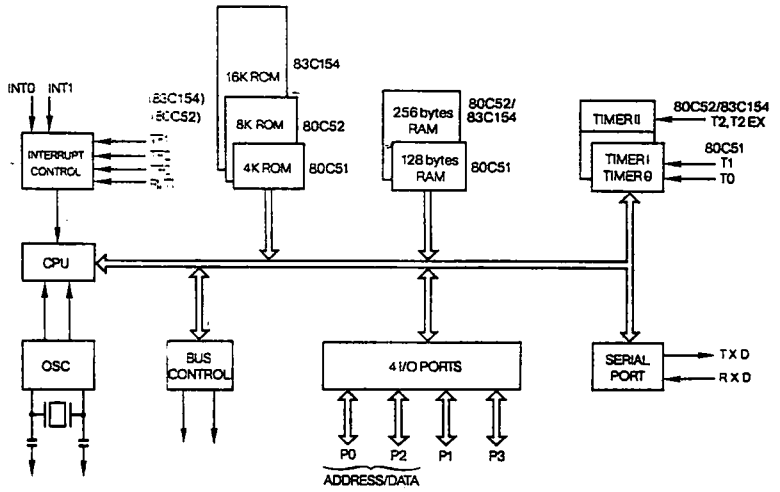
Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	87528H	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
80C52	80C32	—	8K	256	4	3		✓					8/6	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C152JA	80C152JA	—	3K	256	5	2		✓		✓	2		19/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

รูปที่ 3.1 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51

สำหรับในบทนี้จะขอกล่าวถึง โครงสร้าง, สถาปัตยกรรม, รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register , SFR), การขัดจังหวะ, การรีเซ็ทและไดอะแกรมเวลาของการติดต่อกับหน่วยความจำของ MCS51 โดยใช้ 8051 เป็นตัวอย่างในการอธิบาย เพราะไมโครคอนโทรลเลอร์ในตระกูลนี้จะแตกต่างกันเพียงเล็กน้อยเท่านั้น ส่วนชุดคำสั่งของ MCS51 จะขอกล่าวไว้ในภาคผนวก

3.1 โครงสร้างของ 8051

ภายใน 8051 จะประกอบขึ้นด้วย GATE ต่างๆ เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้ จะถูกนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง (Intruction decoder), วงจรสร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังไดอะแกรมในรูปที่ 3.2



รูปที่ 3.2 ไดอะแกรมโครงสร้างของ 8051

ไดอะแกรมในรูปที่ 3.2 เป็นโครงสร้างใหญ่ๆ ของ 8051 เนื่องจากลักษณะของ 8051 เป็นคอมพิวเตอร์จิงประกอบด้วย 3 ส่วนหลักๆ คือ

ส่วนที่ 1 คือ CPU (Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีวงจรที่ทำหน้าที่ สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่า วงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากส่วน CPU นี้จะทำการสร้างสัญญาณ โดยการถอดรหัสจากคำสั่ง (Indtruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์ เพื่อให้ทุกๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ใน CPU นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic

Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหารข้อมูล แล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

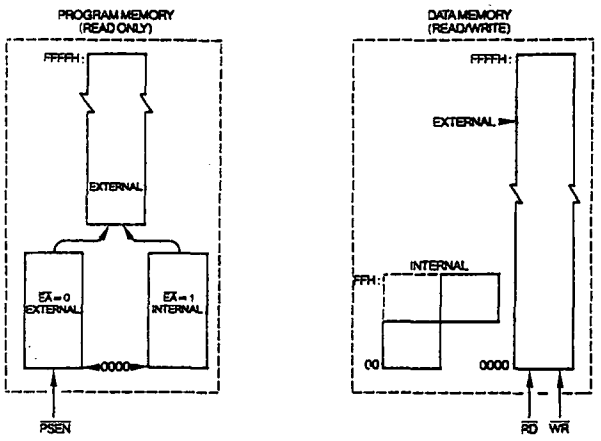
ส่วนที่ 2 คือหน่วยความจำ (Memory) มีไว้สำหรับจดจำข้อมูล โดยมีหน่วยความจำแบ่งไว้เป็น 2 แบบตามลักษณะของการใช้งานคือ

1. Program Memory เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และ ผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non volatite ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย) การเขียนข้อมูลลงไปบน ROM จะต้องใช้เครื่องมือพิเศษ ในระหว่างการทำงานของ 8051 ผู้ใช้จะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้ จำนวนตำแหน่งของหน่วยความจำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่ง (Address) จะเขียนเป็นเลขฐาน 16 ได้ ตั้งแต่ 0000H ถึง FFFFH หน่วยความจำตำแหน่ง 0000H ถึง 0FFFH จำนวน 4 กิโลไบท์ นั้นผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051 (ไมโครคอนโทรลเลอร์เบอร์อื่นๆ เช่น 8052 จะมีขนาดของ ROM ส่วนนี้ได้ถึง 8 กิโลไบท์ ตำแหน่ง 0000H ถึง 1FFFH) ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้ป้อนสัญญาณสภาวะลอจิก High (1) เข้าที่ขา $\overline{EA}$ ของ 8051 แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก Low (0) เข้าที่ขา $\overline{EA}$ ของ 8051 ส่วนหน่วยความจำที่ตำแหน่ง 1FFFH ถึง FFFFH จะต้องต่ออยู่ภายนอก 8051เสมอ ดังแสดงในแผนภูมิหน่วยความจำ (Memory Map) ในรูปที่ 3.3

Internal Memory หมายถึงหน่วยความจำนั้นอยู่ภายใน 8051 ส่วน External Memory หมายถึงหน่วยความจำนั้นอยู่ภายนอก 8051

ไมโครคอนโทรลเลอร์เบอร์ 8031, 8051 และ 8751 นั้น โดยโครงสร้างและรหัสคำสั่งจะเหมือนกันทุกประการ แตกต่างกันที่

- 8031 จะไม่มี ROM ขนาด 4 กิโลไบท์ อยู่ภายใน ผู้ใช้ต้องเลือกการใช้งาน Program Memory อยู่ภายนอกวงจรรวมทั้งหมด 64 กิโลไบท์
- 8051 จะมี ROM ขนาด 4 กิโลไบท์ อยู่ภายใน ถ้าต้องการเก็บคำสั่งควบคุมการทำงานไว้ในหน่วยความจำส่วนนี้ ต้องส่งโปรแกรมคำสั่งไปให้โรงงานผู้ผลิตทำการเขียน



รูปที่ 3.3 แผนภูมิหน่วยความจำของ 8051

ใส่ใน ROM ให้ตั้งแต่ในขั้นตอนของการผลิตวงจรรวม ผู้ใช้ไม่สามารถแก้ไขโปรแกรมได้เอง ถ้าจะนำมาใช้งานโดยเก็บโปรแกรมไว้ในหน่วยความจำช่วง 4 กิโลไบท์แรกอยู่ภายนอก ก็สามารถทำได้โดยการต่อ ROM ไว้ภายนอก แล้วต่อขา $\overline{EA}$ ของ 8051 ไว้กับสัญญาณที่มีสภาวะลอจิกเป็น 0

- 8751 จะมี หน่วยความจำขนาด 4 กิโลไบท์ เป็นแบบ EPROM (Erasable Program Read Only Memory) อยู่ภายในวงจรรวม เอาไว้ใช้เก็บโปรแกรมคำสั่งที่จะให้ 8751 ทำงานผู้ใช้สามารถเขียนคำสั่งลงไปใน EPROM ได้เอง โดยใช้เครื่องมือที่เรียกว่าเครื่องโปรแกรม EPROM (EPROM Programmer) และผู้ใช้สามารถแก้ไขโปรแกรมที่อยู่ใน EPROM ได้โดยการล้างข้อมูลในทุกตำแหน่งของ EPROM ออก ด้วยการฉายแสงอุลตราไวโอเลตผ่านกระจกใสบนวงจรภายใน ตามเวลาที่กำหนดในคู่มือเฉพาะ (Data sheet) ของ 8751 จากนั้นก็ใช้เครื่องโปรแกรม EPROM เขียนโปรแกรมลงไปใหม่ 8751 นี้จะสะดวกมากสำหรับการพัฒนาโปรแกรม

2. Data Memory เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพักเก็บข้อมูลแล้วเรียกมาใช้ใหม่ระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ในโปรแกรม Memory หน่วยความจำแบบนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหายแต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้วข้อมูลใน RAM ก็จะสูญหายไป การสูญหายของข้อมูลไม่ได้หมายความว่าไม่มีอะไรอยู่เลยแต่เป็นการที่มีข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ ซึ่ง

เรียกการเกิดลักษณะแบบนี้ว่า ข้อมูลสูญหายไป หน่วยความจำแบบ Data Memory ของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบท์ ที่ตำแหน่ง 00H ถึง 7FH (เบอร์ 8052 จะมี 256 ไบท์อยู่ที่ตำแหน่ง 00H ถึง FFH) และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของวงจรรวม 8051 มีได้สูงสุด 65536ไบท์ (64 กิโลไบท์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH ดังแสดงในรูปที่ 3.3 หน่วยความจำแบบ Data Memory ภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ทุกตำแหน่งจะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register (SFR) เพราะจะใช้หน่วยความจำเหล่านี้สำหรับงานพิเศษเท่านั้น แต่ละตำแหน่งของหน่วยความจำแบบ SFR นี้อาจเป็น RAM หรือวงจรนับ (Counter) วงจรตั้งเวลา (Timer) ก็ได้ ถ้าเป็น 8052 ซึ่งมี Data Memory ขนาด 256 ไบท์ จะใช้บางตำแหน่งของหน่วยความจำช่วงตำแหน่ง 80H ถึง FFH เป็น SFR ส่วนตำแหน่งอื่นที่เห็นเป็น RAM เหมือนหน่วยความจำช่วง 00H ถึง 7FH นั่นเอง

ส่วนที่ 3 อุปกรณ์อินพุทและเอาท์พุท (Input/output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกับภายนอกได้ ดังในไดอะแกรมรูปที่ 3.2 อุปกรณ์อินพุทและเอาท์พุท ได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial port การทำงานของแต่ละส่วนมีผลดังนี้

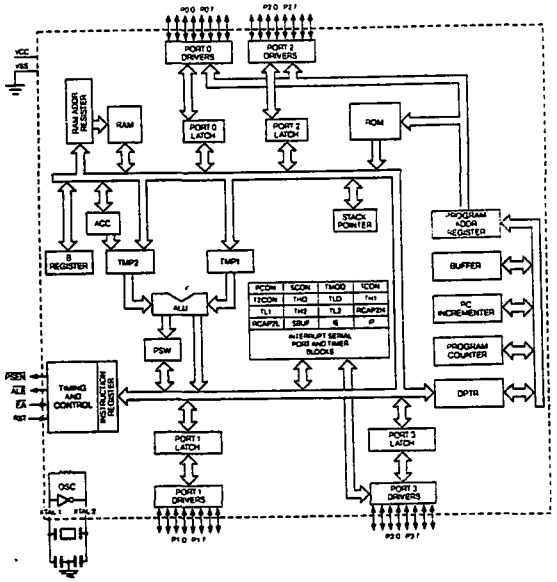
1. 4 I/O Port คำว่าพอร์ท หมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่ใช้สำหรับรับส่งข้อมูล ซึ่งเป็นสัญญาณแบบดิจิตอลเข้าหรือออกจากตัว MCS-51 พอร์ทมีทั้งหมด 4 พอร์ท โดยแต่ละพอร์ทจะรับ-ส่งข้อมูลได้ 8 บิท มีพอร์ท P0, P1, P2 และ P3 บางพอร์ทจะทำงานมากกว่า 1 อย่างก็ได้ เช่น พอร์ท P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการติดต่อ และพอร์ท P0 จะใช้รับส่งข้อมูลเมื่อติดต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดที่เวลาเดียวกัน แต่จะใช้วิธีทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับสัญญาณนาฬิกา

2. Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

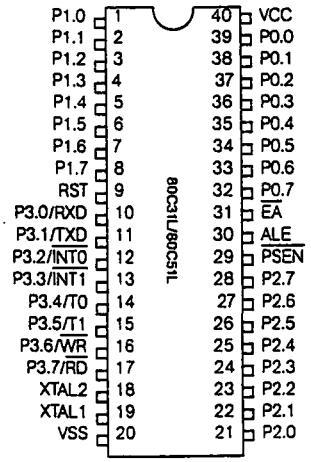
3. Serial Port หรือ พอร์ทอนุกรม CPU จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิท แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปทีละบิทออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิททางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิท เพื่อให้ CPU อ่านไปใช้งานต่อไป

3.2 สถาปัตยกรรมของ 8051

ในรูปที่ 3.4 เป็นสถาปัตยกรรมภายในของ 8051 ซึ่งจะอธิบายส่วนย่อยๆ ของ 8051 เพียงชิพเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังแสดงในรูปที่ 3.5



รูปที่ 3.4 สถาปัตยกรรมภายในของ 8051



รูปที่ 3.5 ไดอะแกรมขาของ 8051 แบบ DIP

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีขาอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขา นั้นจะใช้งานต่างๆ ดังนี้

Vcc

ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ระดับโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

Vss

ขา 20 เป็นขาที่ต้องต่อกับกราวน์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวน์ของอุปกรณ์ต่อเข้าด้วยกัน

Port 0

เป็นพอร์ทขนานขนาด 8 บิทอยู่ที่ขา 39 ถึง 32 เริ่มจากบิท 0 ถึงบิท 7 ตามลำดับดังในรูปที่ 3.5 แต่ละขาจะเขียนว่า P0.0, P0.1,..., P0.7 นั้น P0.7 หมายถึงบิท 7 ของพอร์ท 0 ซึ่งเป็นบิทที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือ บิท 0 ของพอร์ท 0 เป็นบิทที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกที่พอร์ท 0 จะถูก Latch ไว้ที่ขาของพอร์ท ซึ่งสามารถสรุปการใช้งานพอร์ท 0 ได้ดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้คือ 64 กิโลไบท์ จึงมีค่าตำแหน่งหน่วยความจำ 16 บิทของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิทล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิทบนจะส่งออกไปทางพอร์ท 2

2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory

3. ใช้รับ-ส่งข้อมูลทางพอร์ทโดยตรงกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory

Port 1

เป็นพอร์ทขนานขนาด 8 บิท ในรูปที่ 3.5 คือขา P1.0 ถึง P1.7 (ขา 1-8) P1.0 หมายถึงบิท 0 ของพอร์ท 1 ซึ่งเป็นบิท Least Significant Bit และบิท P1.7 หมายถึงบิทที่ 7 ของพอร์ท 1 ซึ่งเป็นบิท Most Significant Bit พอร์ท 1 นี้จะใช้เป็นตัวรับส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางขา

Port 2

พอร์ทขนานขนาด 8 บิท คือขา P2.0 ถึง P2.7 (บิท 0 ถึงบิท 7 ของพอร์ท 2) ในรูปที่ 3.5 พอร์ท 2 จะใช้งานได้ดังนี้

1.ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อค่าตำแหน่งนี้เป็น 8 บิทบนของค่าตำแหน่ง

2.ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

Port 3

คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับในรูปที่ 3.5 พอร์ทที่ 3 นี้แต่ละบิทสามารถจะใช้ในการทำงานเป็นพอร์ทรับ-ส่งข้อมูลหรือใช้ทำงานในฟังก์ชันอื่นๆ ได้โดยใช้คำสั่งควบคุมการทำงานโดยแต่ละบิทของพอร์ท 3 จะมีฟังก์ชันอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/$\overline{\text{INT0}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/$\overline{\text{INT1}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer/Counter0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณT0 นี้หรือสัญญาณนาฬิกาก็ได้

P3.5/T1 (Timer/Counter1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

P3.6/$\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลไปยังภายนอก 8051

P3.7/$\overline{\text{RD}}$ (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

RST

ขารีเซ็ทขานี้จะใช้ทำการรีเซ็ทการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับขากราวน์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลต์ เพื่อให้เกิดการรีเซ็ทเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซ็ทจะทำให้ค่าในรีจิสเตอร์ต่างๆ เปลี่ยนไปเป็นค่าหนึ่งดังในตารางรูปที่ 3.6

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
THO	00H
TLO	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

รูปที่ 3.6 ค่าของรีจีสเตอร์เมื่อเกิดการรีเซ็ท 8051

ในตารางรูปที่ 3.6 ช่องทางขวาแสดงค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซ็ทในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซ็ทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิทตลอดเวลาที่สัญญาณของขา RST เป็น High อยู่

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ท 8051 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC(Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้ต้องเขียนโปรแกรมมาเก็บไว้ที่ ตำแหน่ง 0000H

ALE

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่1/6เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1)จะมีการส่งข้อมูลที่เป็น 8 บิทล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท0จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอกสัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

$\overline{\text{PSEN}}$

Program Store Enable เป็นขาที่ 29 ในรูปที่ 3.5 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 8051 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา $\overline{\text{PSEN}}$ นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

$\overline{\text{EA}}$

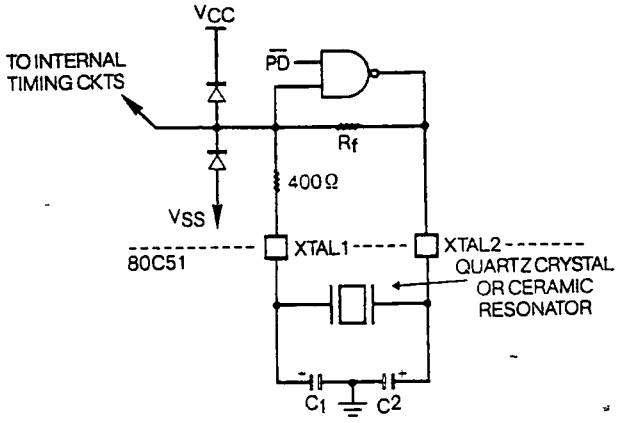
External Acess ขา 31 ของรูปที่ 3.5 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 3.4 เพื่อควบคุมการสร้างสัญญาณ $\overline{\text{PSEN}}$ ถ้าป้อนสัญญาณลอจิก 0เข้าไปที่ ขา $\overline{\text{EA}}$ นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ $\overline{\text{PSEN}}$ ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา $\overline{\text{EA}}$ เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆจาก ROM ภายใน 8051

XTAL 1

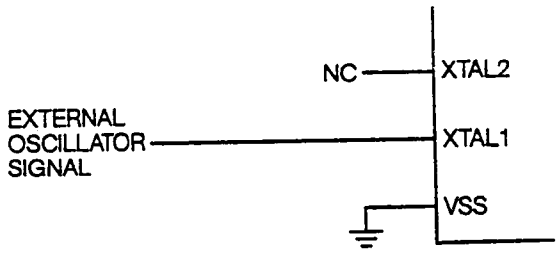
ขาที่ 19 ของรูปที่ 3.5 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 3.7 จะเห็นวงจรภายในของออสซิลเลเตอร์ NAND Gate จะทำหน้าที่เป็นวงจรขยายกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิท PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกาควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crytalหรือเซรามิคเรโซเนเตอร์ดังรูปที่ 3.7 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF

XTAL 2

ขาที่ 18 ของรูปที่ 3.5 ขานี้เป็นจุดเอาท์พุทของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุท คือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 3.8



รูปที่ 3.7 วงจรออสซิลเลเตอร์ภายใน 8051



รูปที่ 3.8 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

3.3 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register, SFR)

ใน 8051จะใช้วิธีการกำหนดชื่อให้กับตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน (Internal Data Memory) ที่เรียกว่าการ Symbolize เช่น การให้ชื่อหน่วยความจำแต่ละตำแหน่งในแต่ละ Bank ซึ่งอยู่ในช่วงหน่วยความจำตำแหน่ง 00H ถึง 1FH แล้วในคำสั่งจะอ้างอิงหน่วยความจำแต่ละตำแหน่งโดยการใช้ชื่อ R0, R1, R2, R3, R4, R5, R6 และ R7 หน่วยความจำตำแหน่งเหล่านี้จะเรียกอีกอย่างหนึ่งว่า เป็นรีจิสเตอร์ซึ่งมีหน้าที่ในการเก็บหรือพักข้อมูล หรือใช้สำหรับการกระทำบางอย่าง รีจิสเตอร์กลุ่มหนึ่งใน 8051 ที่เรียกว่า Special Function Register (SFR) เป็นรีจิสเตอร์ที่ใช้สำหรับงานเฉพาะ คือข้อมูลที่ถูกนำไปเก็บไว้ในรีจิสเตอร์เหล่านี้ จะมีความหมายเฉพาะตัวของรีจิสเตอร์ที่แต่ละตำแหน่งของ SFR อาจจะไม่ใช่เป็นหน่วยความจำ (RAM) แต่อาจเป็นตัวนับ (Count Register), Shift Register หรือ Latch ซึ่งการอ้างอิงข้อมูลในแต่ละตำแหน่งนั้น 8051 จะถือเสมือนว่าเป็นหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งนี้ว่า Memory Map I/O รีจิสเตอร์กลุ่มนี้มีดังในรูปที่ 3.9

Table 1

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 1 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H
*IOCON (1)	IO Control	F8H

+ 80C52 and 83C154 only * bit addressable
(1) 83C154 only

รูปที่ 3.9 Special Function Register (SFR)

ในรูปที่ 3.9 ช่อง Symbol ทางซ้ายจะเป็นสัญลักษณ์ของรีจิสเตอร์ ในช่องถัดมาคือชื่อของรีจิสเตอร์ตามสัญลักษณ์ที่อยู่ทางซ้าย ในช่องขวาสุดจะเป็นตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่แทนด้วยชื่อหรือสัญลักษณ์ทางซ้ายนั่นเอง การอ่าน หรือเขียน

ข้อมูลกับรีจิสเตอร์เหล่านี้ สามารถทำได้โดยการใช้คำสั่งในกลุ่มการเคลื่อนย้ายข้อมูล และรีจิสเตอร์บางตัวในกลุ่มนี้ยังสามารถใช้คำสั่งในกลุ่ม Boolean Instruction เพื่อการทำงานกับแต่ละบิทในรีจิสเตอร์เหล่านี้ได้ จากตารางในรูปที่ 3.9 รีจิสเตอร์ที่มีเครื่องหมาย * อยู่ข้างหน้า จะสามารถใช้คำสั่ง Boolean Instruction จัดการกับแต่ละบิทได้ รีจิสเตอร์ที่มีเครื่องหมาย + นำหน้าหมายความว่ารีจิสเตอร์นั้นจะมีเฉพาะใน 80C52 และ 83C154 เท่านั้นไม่มีใน 8051

Direct Byte Address	Bit Address (MSB)							(LSB)	Special Function Register Symbol
	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF	
0F8H	FF	FE	FD	FC	FB	FA	F9	F8	IOCON
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CY	AC	F0	RS1	RS0	OV	F1	P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0CDH	Not Bit Addressable								TH2
0CCH	Not Bit Addressable								TL2
0CBH	Not Bit Addressable								RCAP2H
0CAH	Not Bit Addressable								RCAP2L
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	
0C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
	PCT		PT2	PS	PT1	PX1	PT0	PX0	
0B8H	BF	-	BD	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA		ET2	ES	ET1	EX1	ET0	EX0	
0A8H	AF	-	AD	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	Not Bit Addressable								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
8DH	Not Bit Addressable								TH1
8CH	Not Bit Addressable								TH0
8BH	Not Bit Addressable								TL1
8AH	Not Bit Addressable								TL0
89H	Not Bit Addressable								TMOD
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
87H	Not Bit Addressable								PCON
83H	Not Bit Addressable								DPH
82H	Not Bit Addressable								DPL
81H	Not Bit Addressable								SP
80H	87	86	85	84	83	82	81	80	P0

รูปที่ 3.10 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท

ในรูปที่ 3.10 ในช่องสี่เหลี่ยมเล็กๆ จะเป็นตำแหน่งของบิทนั้นในแต่ละรีจิสเตอร์ เช่น ในช่องซ้ายสุดของรีจิสเตอร์ TCON มีค่า 8FH ซึ่งเป็นตำแหน่งบิท 7 ของหน่วยความจำตำแหน่ง 88H

รีจิสเตอร์ในกลุ่ม Special Function Registerมีดังนี้

1. Accumulator ตำแหน่งหน่วยความจำภายในเท่ากับ 0E0H

รีจิสเตอร์นี้มีขนาด 8 บิท เป็นรีจิสเตอรที่ใช้มาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างอิงถึงรีจิสเตอร์นี้โดยใช้สัญลักษณ์ A เช่น MOV A,#15H คำสั่งที่อ่านหรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น เช่น MOVX @R0,A หรือ MOVX A, @R0เป็นต้น และข้อมูลที่อยู่ในรีจิสเตอร์นี้ก็สามารถที่จะให้โปรแกรมตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้ เช่น JZ rel

2. B Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H

เป็นรีจิสเตอร์ขนาด 8 บิทที่ใช้ คำสั่งในการคูณ (MUL AB) และ คำสั่งการหาร (DIV AB) เท่านั้น โดยรีจิสเตอร์ B นี้จะเก็บตัวคูณและผลลัพธ์บิท 8 ถึง 15 ในคำสั่งการคูณ ส่วนในคำสั่งการหารนั้นรีจิสเตอร์ B จะเก็บตัวหารและผลการหาร การเขียนข้อมูลไปยังรีจิสเตอร์นี้จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H, 25H จะเป็นการกำหนดค่า 25H ให้กับรีจิสเตอร์ B

3. Program Status Word ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H

เป็นรีจิสเตอร์ขนาด 8 บิท แต่ละบิทของ PSW จะสามารถกำหนดให้เป็น 1 หรือ 0 ได้ด้วยคำสั่ง SETB หรือ CLR ตามลำดับ ค่าตำแหน่งบิท 0 ถึงบิท 7 ของรีจิสเตอร์ PSW เท่ากับ D0H ถึง D7H ตามลำดับ

4. Stack Pointer ตำแหน่งหน่วยความจำภายในเท่ากับ 081H

เป็นรีจิสเตอร์ขนาด 8 บิท รีจิสเตอร์นี้จะใช้ตำแหน่งหน่วยความจำภายใน 8051 ที่ใช้เก็บตำแหน่ง (Address) เดิมของโปรแกรมก่อนทำงานคำสั่ง CALL หรือตำแหน่งที่ใช้เก็บข้อมูลด้วยคำสั่ง PUSH และตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำการรีเซ็ท 8051 โดยการป้อนสัญญาณสภาวะลอจิก 1 เข้าไปที่ขา RST ของ 8051 จะทำให้ข้อมูลในรีจิสเตอร์นี้มีค่าเป็น 07H หมายความว่า รีจิสเตอร์ SP ชี้หน่วยความจำภายใน 8051 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการใช้คำสั่งเคลื่อนย้ายข้อมูล หรือการทำงานของคำสั่ง PUSH, POP, CALL

5. Data Pointer Register ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H

รีจิสเตอร์ DPTR มีขนาด 16 บิท หน้าที่ของรีจิสเตอร์นี้ก็คือ ใช้สำหรับชี้ตำแหน่งในหน่วยความจำ DTPRนี้สามารถใช้อ้างอิงตำแหน่งหน่วยความจำได้สูงสุด 60x1024 ตำแหน่ง เช่นคำสั่ง MOVX A, @DPTR หรือใช้ชี้ตำแหน่งโปรแกรมที่ต้องการกระโดดข้ามไปทำงาน เช่นคำสั่ง JMP @A + DPTR รีจิสเตอร์ DPTR นี้ประกอบด้วยรีจิสเตอร์ขนาด 8 บิท 2 ตัว

คือ DPH ซึ่งอยู่ที่ตำแหน่ง 83H และ DPL ซึ่งอยู่ที่ตำแหน่ง 82H ในหน่วยความจำภายในสำหรับข้อมูลภายใน 8051 ดังนั้นการแก้ไขข้อมูลในรีจิสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิท เช่นคำสั่ง MOV DPTR ,#data16 หรือจัดการทีละ 8 บิทโดยการแก้ไขข้อมูลใน DPH หรือ DPL ด้วยคำสั่ง MOV 83H,#data 8 หรือ MOV 82H ,#data8

6. Port 0 ถึง 3 ตำแหน่งหน่วยความจำภายในเท่ากับ 80H , 90H, 0A0H, 0B0H

Special Function Register ชื่อ P0, P1, P2 และ P3เป็นรีจิสเตอร์ขนาด 8 บิท ของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 80H, 90H, 0A0H และ 0B0H ตามลำดับ การเขียนข้อมูลลงไปยังหน่วยความจำแต่ละตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้นๆ ของ 8051 ข้อมูลที่เขียนออกไปจะถูก Latch ค้างไว้ และปรากฏที่แต่ละบิทของพอร์ท ในการอ่านข้อมูลในรีจิสเตอร์แต่ละตัวนั้นจะเป็นการอ่านสภาวะลอจิกของสัญญาณที่ปรากฏอยู่ที่แต่ละขาของพอร์ทนั้น การอ่านข้อมูลแต่ละพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้นๆ เสียก่อน ทุกบิทของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยใช้คำสั่ง SETB bit และ CLR bit

7.Serial Data Buffer ตำแหน่งหน่วยความจำภายในเท่ากับ 99H

รีจิสเตอร์นี้มีขนาด 8 บิทและมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 เท่ากับ 99H โครงสร้างภายในแล้วรีจิสเตอร์นี้มี 2 ตัวที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก 8051 และอีกตัวหนึ่งสำหรับรับข้อมูลแบบอนุกรมที่เข้ามาดังนั้น Serial Port ของ 8051 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกันเนื่องจากมีรีจิสเตอร์และรับแยกออกจากกัน ข้อมูลที่ต้องการจะส่งออกก็ให้เขียนไปยังรีจิสเตอร์ SBUF แล้วสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจิสเตอร์จะเริ่มส่งออกโดยเริ่มจากบิท 0 ถึงบิท 7 ตามลำดับ ถ้ามีข้อมูลเข้ามาทางขา RXD ก็จะเก็บไปไว้ในรีจิสเตอร์นี้โดยถือว่าข้อมูลบิทแรกที่เข้ามาคือบิท 0

8. SCON (Serial Port Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H รีจิสเตอร์ SCON มีขนาด 8 บิท ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิทของข้อมูลในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.11 ซึ่งมีบิท RI เป็นชื่อของบิท 0 และ SM0 เป็นบิท 7 ของรีจิสเตอร์ SCON

9. Timer Register TH0, TL0, TH1, TL1 ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH, 8BH

ใน 8051 จะมีวงจร Timer อยู่ 2 ชุดคือ Timer 0 และ Timer 1 (8052 จะ

SCON : SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier.(NOTE 1).
SM1	SCON.6	Serial Port mode specifier. (NOTE 1).
SM2	SCON.5	Enables the multiprocessor communication feature in mode 2 &3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 9).
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 &3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1 :

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

SERIAL PORT SET-UP : **Table 9**

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

รูปที่ 3.11 Serial Port Control Register (SCON)

มี Timer 2 อีก 1 ชุด) ใน Timer แต่ละชุดจะมี Register อยู่ 2 ตัวเพื่อเก็บค่าการนับของ Timerได้สูงสุดถึง 16 บิท ใน Timer 0 รีจิสเตอร์นี้คือ TH0, TL0 และ ใน Timer 1 คือ Register TH1, TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค่าของการนับ 8 บิทล่าง และ THx จะเก็บค่าของการนับ 8 บิทบน ผู้ใช้จะสามารถเลือกให้วงจร Timer ทำงานในโมด Timer หรือโมด Counterได้โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Timer/Counter Mode Control Register) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิล (Cycle) ของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ TLx และ THx จะเก็บได้คือค่า FFFFH แล้วยังนับต่อไป ค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ Set บิทบางบิทในรีจิสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflowนี้ในการให้วงจร Timer ทำงานเป็น

Counter ก็คือการใช้รีจิสเตอร์ THx และ TLx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 8051 สัญญาณที่เข้ามาทางขา T0 หรือ T1 อาจจะมาจากอุปกรณ์ตรวจจับ (Sensor) ก็ได้ แต่สภาวะของสัญญาณนี้จะต้องมีระดับโวลเตจของสภาวะลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลเตจไม่เกิน 0.6 โวลต์และลอจิก 1 ต้องมีโวลเตจมากกว่า 2.4 โวลต์

10. TMOD Timer/Counter Mode Register ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TMOD เป็นรีจิสเตอร์ขนาด 8 บิท ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิทในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.12

GATE	C/T	M1	M0	GATE	C/T	M1	M0
TIMER 1				TIMER 0			

GATE When TRx (in TCON) is set and GATE=1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE=0, TIMER/COUNTERx will run only while TRx=1 (software control).

C/T Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1 :

M1	M0		Operating Mode
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

รูปที่ 3.12 TMOD Timer/Counter Mode Register

ในรูปที่ 3.12 M0 เป็นชื่อของบิท 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจิสเตอร์นี้แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิท คือบิท 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และ บิท 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิทที่มีชื่อเดียวกันจะเหมือนกัน

11. TCON Timer Control Register ตำแหน่งหน่วยความจำภายในเท่ากับ 088H

รีจิสเตอร์ขนาด 8 บิทนี้ ใช้ควบคุมการทำงานและบอกสภาวะ ของ Timer 0 และ Timer 1 แต่ละบิทของรีจิสเตอร์นี้จะทำงานต่างกันดังรูปที่ 3.13

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ONOFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify failling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify failling edge/low level triggered External Interrupt.

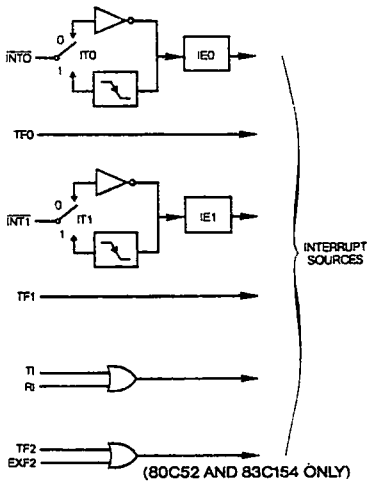
รูปที่ 3.13 TCON Timer Control Register

รูปที่ 3.13 T0 เป็นชื่อของบิท 0 และ TF เป็นชื่อของบิท 7 โดย 4 บิทแรกจะเกี่ยวข้องกับการขัดจังหวะ (Interrupt) ส่วน 4 บิทหลังนั้นจะเกี่ยวข้องกับโมดการทำงานของ Timer

ในขณะที่ Timer ทำงานในโมดของ Timer นั้น รีจิสเตอร์ที่ทำหน้าที่เป็นตัวนับจะมีค่าเพิ่มขึ้น 1 ทุกๆ 1ไซเคิลของเครื่อง ซึ่งเท่ากับ 12 คาบของสัญญาณจากออสซิลเลเตอร์ ในกรณีที่ Timer ทำงานเป็น Counter เพื่อจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 รีจิสเตอร์จะเพิ่มค่าไป 1 เมื่อมีการเปลี่ยนสถานะของสัญญาณที่ขาดังกล่าวจาก 1 เป็น 0 โดยวงจรภายใน 8051 จะตรวจสอบสถานะของสัญญาณที่ขาดังกล่าวในช่วงเวลาเฟส 2 ของ State 5 (S5P2) ในทุกๆ 1ไซเคิลของเครื่อง เช่น ในเวลา S5P2 ครั้งหนึ่งพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 1 และในเวลา S5P2 ของไซเคิลของเครื่องถัดมาพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 0 ก็จะทำให้ค่าในรีจิสเตอร์ตัวนับเพิ่มค่าไป 1แต่ละไซเคิลของเครื่องจะเท่ากับ 12 ไซเคิล ของสัญญาณจากออสซิลเลเตอร์ดังนั้นสัญญาณที่จะนับได้จะต้องเป็น 1 อย่างน้อยให้ถูกจับได้ใน 1ไซเคิลของเครื่องและเป็น 0 อย่างน้อยก็ต้องให้ถูกตรวจจับได้ใน 1 ไซเคิลของเครื่องเช่นกัน ทำให้ความถี่สูงสุดของสัญญาณที่ขา T0 หรือ T1 ที่นับได้ต้องไม่เกิน 1/24 เท่าของสัญญาณจากออสซิลเลเตอร์

12. IE Interrupt Enable Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0A8H

การขัดจังหวะการทำงาน (Interrupt) เป็นการที่มีสัญญาณหนึ่งหรือคำสั่งหนึ่งที่ (ไม่ใช่คำสั่ง CALL หรือ JMP) จะทำให้การทำงานปกติของโปรแกรมถูกขัดจังหวะแล้วข้ามไปทำงานยังตำแหน่งหนึ่งตำแหน่งใดที่กำหนดไว้ เมื่อทำงานในโปรแกรมขัดจังหวะเสร็จสิ้นก็จะกับมาทำงานในโปรแกรมที่ตำแหน่งก่อนจะไปทำงานยังโปรแกรมขัดจังหวะ โปรแกรมที่ถูกกระโดดไปทำงานเรียกว่า โปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service -Routine) ใน 8051 จะสามารถขัดจังหวะด้วยสัญญาณจาก 6 แหล่งดังรูปที่ 3.14 ถ้าเป็น 8052 หรือ 83154 จะสามารถขัดจังหวะได้ด้วยสัญญาณจาก 8 แหล่งคือ สัญญาณในชุดล่างของรูปที่ 3.14



รูปที่ 3.14 แหล่งกำเนิดสัญญาณขัดจังหวะ

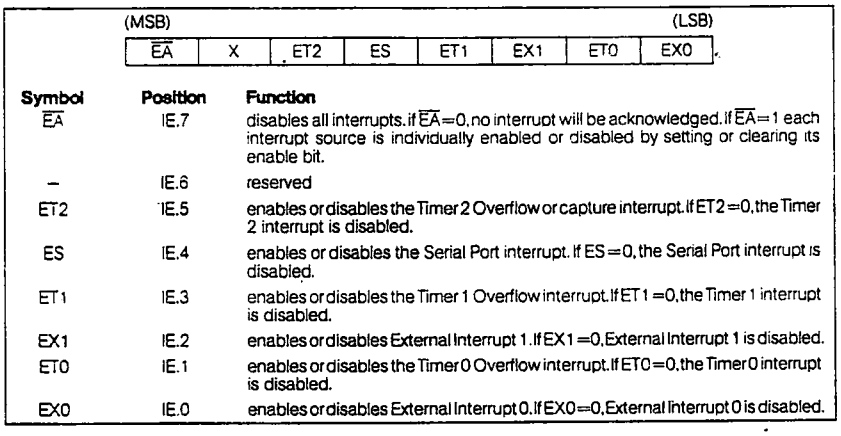
สัญญาณขัดจังหวะที่ 5 ในรูปที่ 3.14 จะสามารถทำให้เกิดการขัดจังหวะได้ 2 วิธี คือ มีข้อมูลเข้ามาทางพอร์ทอนุกรมเก็บอยู่ที่รีจิสเตอร์ SBUF และกรณีที่ข้อมูลใน SBUF ส่งออกไปทางพอร์ทอนุกรมหมดแล้ว ไม่ว่าเกิดกรณีใดๆ ก็ทำให้เกิดการขัดจังหวะขึ้น

สัญญาณภายนอกที่เข้ามายัง 8051 ทางขา $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$ จะสามารถทำให้เกิดการขัดจังหวะการทำงาน 8051 ได้ (สัญญาณที่ 1 และ 3 ในรูปที่ 3.14) โดยสภาวะของสัญญาณนั้นเปลี่ยนจาก 1 เป็น 0 หรือเมื่อสัญญาณนั้นเป็น 0 แล้วแต่การกำหนดในบิท IT0 และ IT1 ของรีจิสเตอร์ TCON จะทำให้บิท IE0 กับ IE1 เป็นตัวสร้างสัญญาณขัดจังหวะต่อไป

จาก Timer 0 และ Timer 1 เมื่อค่าการนับในแต่ละโมดถึงค่าสูงสุดในโมดนั้น

แล้วเมื่อทำการนับต่อไปค่าการนับต่อไปจะเป็น 0 (หรืออาจจะเป็นค่าที่ Reload จาก THx ในโมด 2)และทำให้บิท TF0, TF1 เป็น 1 ซึ่งสัญญาณจาก 2 บิทนี้จะสามารถทำให้เกิดการขัดจังหวะได้เช่นกันดัง เช่นสัญญาณขัดจังหวะที่ 2 และ 4 ใน รูปที่ 3.14

แหล่งกำเนิดสัญญาณทั้ง 6 ที่สามารถทำให้เกิดการขัดจังหวะได้ 5 แบบนี้ ผู้ใช้สามารถกำหนดให้สัญญาณใดบ้างเกิดการขัดจังหวะเรียกว่า Enable หรือไม่ให้เกิดการขัดจังหวะเรียกว่า Disable โดยการกำหนดในรีจิสเตอร์ IE (Interrupt Enable Register) ซึ่งมี 8 บิท แต่ละบิทสามารถ Enable ให้ขัดจังหวะได้จากแต่ละสัญญาณดัง รูปที่ 3.15



	(MSB)							(LSB)
	$\overline{EA}$	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
$\overline{EA}$	IE.7	disables all interrupts. if $\overline{EA}$=0, no interrupt will be acknowledged. If $\overline{EA}$=1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	reserved
ET2	IE.5	enables or disables the Timer 2 Overflow or capture interrupt. If ET2=0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt. If ES=0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1=0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External Interrupt 1. If EX1=0, External Interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0=0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt 0. If EX0=0, External Interrupt 0 is disabled.

รูปที่ 3.15 Interrupt Enable Register

ถ้าต้องการ Enable บิทใดก็ให้โปรแกรมกำหนดค่าในบิทนั้นเป็น 1 ถ้าค่าในบิทนั้นเป็น 0 หมายถึง Disable การ Disable จะทำให้ไม่มีการขัดจังหวะการทำงานของโปรแกรมเนื่องจากสัญญาณของการขัดจังหวะนั้นๆ EX0 เป็นชื่อบิท 0 และ $\overline{EA}$ เป็นชื่อของบิท 7 โดย X (บิท 6) จะไม่มีการใช้งาน

การกำหนดให้บิทใด Enable หรือ Disable นั้นจะเป็นไปโดยอิสระไม่ขึ้นแก่กันจึงสามารถกำหนดให้บิทใดหรือมากกว่า 1 บิท Enable ก็ได้ ดังนั้น 8051 จึงมีรีจิสเตอร์อีกตัวที่ใช้เลือกว่า ถ้ามีสัญญาณของการขัดจังหวะโปรแกรมเข้ามาพร้อมกันมากกว่า 1 แล้วจะทำโปรแกรมตอบสนองการขัดจังหวะอันใดก่อนรีจิสเตอร์นั้นคือ IP Interrupt Priority Register

13. IP Interrupt Priority Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H

ในการตอบสนองต่อสัญญาณขัดจังหวะของ 8051 นั้น ถ้าสัญญาณขัดจังหวะทั้งหมดเข้ามาพร้อมกัน 8051 จะต้องเลือกทำงานโปรแกรมตอบสนองการขัดจังหวะ โดยการตรวจสอบสัญญาณเรียงตามลำดับ ซึ่งเรียกวิธีการ Polling สัญญาณขัดจังหวะหนึ่งจะถูกตรวจสอบแล้วสัญญาณอื่นๆ จะถูกตรวจสอบต่อมา ถ้าสัญญาณนั้นขอขัดจังหวะ 8051 จะสร้างคำสั่ง CALL เป็นพิเศษขึ้นมาเพื่อไปทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณนั้น เมื่อเสร็จสิ้นแล้วก็จะกลับมาทำงานในโปรแกรมเดิมก่อนการขัดจังหวะ ทำให้เสมือนว่าสัญญาณแต่ละสัญญาณมีลำดับความสำคัญไม่เท่ากัน สัญญาณขัดจังหวะจะมีลำดับความสำคัญดังนี้โดยเรียงจากลำดับความสำคัญสูงสุดถึงต่ำสุด

1. IE0
2. TF0
3. IE1
4. TF1
5. RI+TI

แต่ในการใช้งานบางครั้ง จำเป็นที่จะต้องให้สัญญาณใดสัญญาณหนึ่งมีลำดับความสำคัญสูงสุด (Highest Priority) เพื่อจะทำงานโปรแกรมตอบสนองการขัดจังหวะได้ก่อนการขัดจังหวะของสัญญาณอื่น จะสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ใหม่โดย

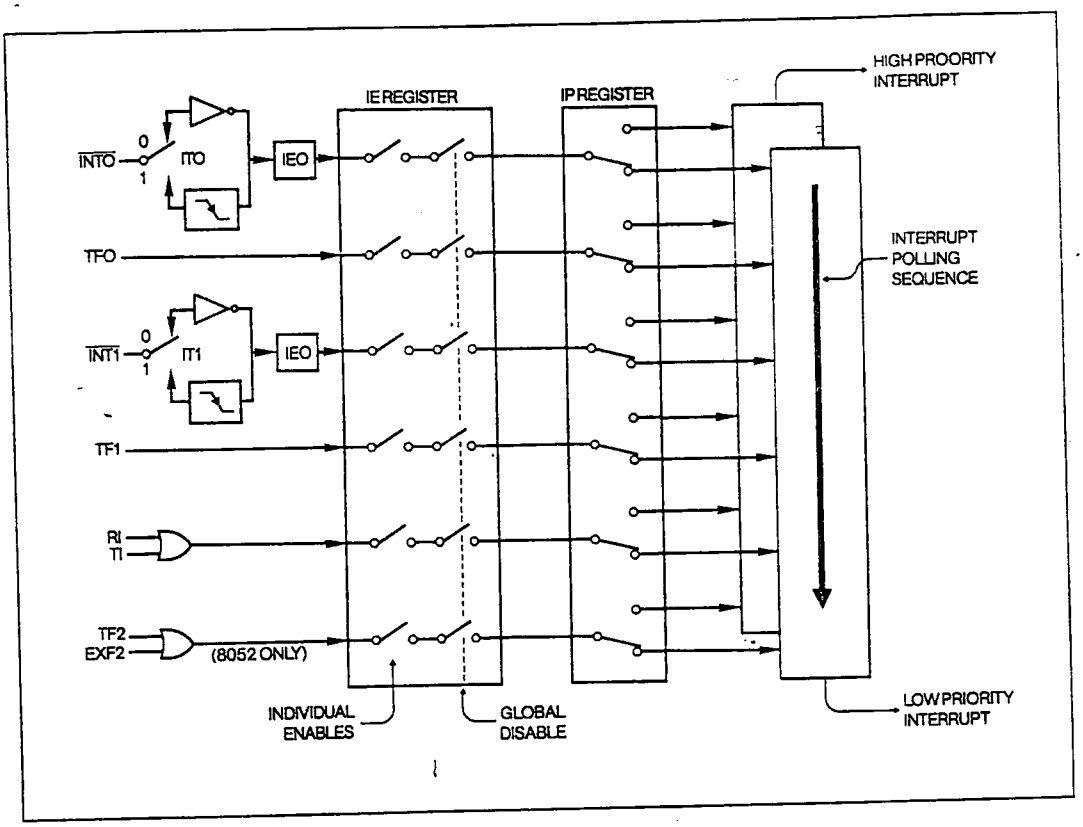
(MSB) (LSB)

X	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	PCT=1, only one level
—	IP.6	reserved
PE2	IP.5	defines the Timer 2 interrupt priority level. PT2=1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1=1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0=1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

รูปที่ 3.16 IP :Interrupt Priority Register

การกำหนดข้อมูลในบิทของรีจิสเตอร์ IP (Interrupt Priority Register) ตามตำแหน่งของแต่ละบิทในรูปที่ 3.16



รูปที่ 3.17 ระบบการขัดจังหวะของ 8052 และ 83154

รูปที่3.17 เป็นแผนภาพแสดงระบบขัดจังหวะของ 8052 ซึ่งแตกต่างจากของ8051 ตรงที่ 8052 จะมีสัญญาณขัดจังหวะมาจาก TF2 , EFX2 คือชุดล่างในภาพ

ในรูปจะเห็นว่าแต่ละสัญญาณจะมีสวิตช์ควบคุมอยู่ 3 ตัว 2 ตัวแรกอยู่ในกรอบสี่เหลี่ยม IE Register และอีกหนึ่งสวิตช์อยู่ในกรอบ IP Register สวิตช์ตัวแรกอยู่ทางซ้ายสุดจะควบคุมด้วยข้อมูลแต่ละบิท บิท 0 ถึงบิท 5 ของรีจิสเตอร์ IE ถ้าข้อมูลเป็น 1 จะทำให้สวิตช์นั้นปิดวงจร (Closed Circuit) การควบคุมสวิตช์ทางซ้ายสุดของแต่ละสัญญาณจะไม่ขึ้นแก่กันสวิตช์ที่ 2 ถัดมาของทุกสัญญาณจะควบคุมร่วมกันด้วยบิท EA ในรีจิสเตอร์ IE ถ้าบิทนี้เป็น 0 สวิตช์ที่ 2 ของสัญญาณจะเปิดวงจร (Opened Circuit) ทำให้ไม่มีสัญญาณขอขัดจังหวะผ่านไปได้ สวิตช์ที่ 3 ทางขวาสุดจะใช้สำหรับเลือกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูง (High Priority Interrupt) หรือลำดับความสำคัญต่ำ (Low Pri-

ority Interrupt)

ถ้าต้องการให้สัญญาณใดๆ มีลำดับความสำคัญสูงก็ให้กำหนดบิทนั้นในรีจิสเตอร์ IP เป็น 1 สวิตช์ที่ 3 จะเลื่อนไปอยู่ตำแหน่งบน ถ้าไม่ต้องการก็กำหนดให้บิทนั้นเป็น 0 บิทใดเป็น 1 เรียกว่า สัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญสูง และบิทใดเป็น 0 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญต่ำ ถ้าในกลุ่มลำดับความสำคัญสูงมีเพียง 1 สัญญาณ ก็จะเรียกว่าสัญญาณนั้นมีลำดับความสำคัญสูงสุดในกลุ่มลำดับความสำคัญเดียวกัน ก็จะมีการจัดลำดับความสำคัญเฉพาะกลุ่มโดยวิธี Polling เหมือนเดิม เช่น กรณีที่มีการกำหนดในบิทของรีจิสเตอร์ IP ให้มีลำดับความสำคัญสูงหรือต่ำเหมือนกัน แล้วเกิดมีความต้องการขอขัดจังหวะจากสัญญาณนั้นๆมาพร้อมกัน 8051 ก็จะทำงานในโปรแกรมตอบสนองการขัดจังหวะเรียงตามลำดับความสำคัญ 5 ลำดับที่กล่าวมาแล้วข้างต้นในขณะที่ 8051 กำลังทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะที่มีลำดับความสำคัญต่ำอยู่ ถ้ามีสัญญาณขัดจังหวะที่มีลำดับความสำคัญสูงกว่าเกิดขึ้น การทำงานของโปรแกรมก็จะกระโดดไปทำงานในตำแหน่งโปรแกรมตอบสนองการขัดจังหวะของสัญญาณที่มีลำดับความสำคัญสูง เสร็จแล้วจึงกลับมาทำงานที่โปรแกรมตอบสนองการขัดจังหวะลำดับความสำคัญต่ำต่อไป

14. PCON (Power Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 87H

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
HPD	PCON.6 (83C154 only)	Hard Power Down bit. Setting this bit allows CPU to enter in Power Down state on an external event (1 to 0 transition) on bit T1 (p.3-5) the CPU quit the Hard Power Down mode when bit T1 (p.3-5) go high or when reset is activated.
RPD	PCON.5 (83C154 only)	Recover from idle or Power Down bit. When 0 RPD has no effect. When 1, RPD permits to exit from idle or Power Down with any non enabled interrupt source (except timex 2). In this case the program start at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
–	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Ide mode bit. Setting this bit activates idle mode operation.

รูปที่ 3.18 PCON : Power Control Register

8051 เป็นไมโครคอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยี ทั้งแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS มีข้อดีตรงที่ใช้กำลังไฟต่ำกว่าแบบ HMOS ดังนั้นต่อไปในอนาคตจึง

จะมีเฉพาะรุ่น CHMOS เท่านั้น นอกจากนี้แล้ว 8051 ยังมีข้อดีตรงที่สามารถลดการใช้กำลังไฟลงได้โดยการทำงานใน Idle Mode และ Power Down Mode ใน Idle Mode นั้นสัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน Interrupt , Serial Port และ Timer ในส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยงแต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร การใช้กำลังไฟจึงลดลงมากส่วนใน Power Down Mode นั้น ออสซิลเลเตอร์จะหยุดทำงาน ทำให้ไม่มีสัญญาณนาฬิกาไปเลี้ยงส่วนใดๆ ในวงจรเลย แต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูญหายไป การสั่งงานให้ 8051 ทำงานในโมดของ Idle หรือ Power Down จะสามารถทำได้โดยใช้กำหนดค่าในรีจิสเตอร์ PCON (Power Control Register)

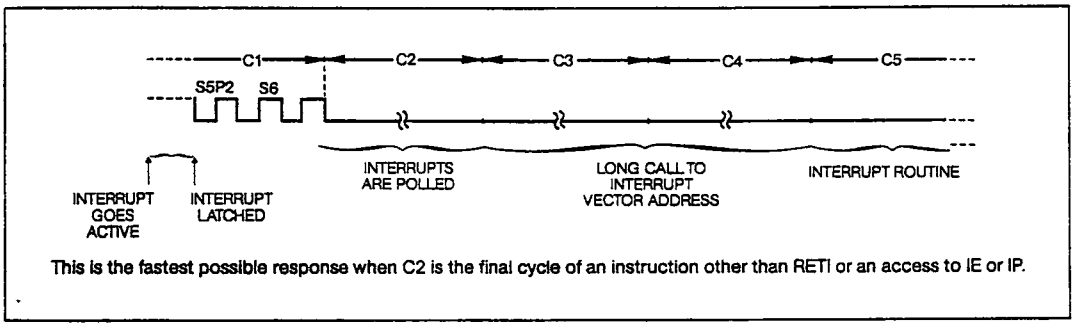
3.4 การขัดจังหวะ (Interrupt)

การขัดจังหวะคือสภาวะที่คอมพิวเตอร์กำลังทำงานอยู่แล้วถูกขัดจังหวะด้วยสัญญาณหรือคำสั่งพิเศษที่ทำให้คอมพิวเตอร์ต้องละจากงานที่กำลังทำอยู่ ไปทำงานในโปรแกรมตอบสนอง การขัดจังหวะนั้นเมื่อเสร็จแล้วก็จะกลับมาทำงานเดิมต่อไปได้ ใน 8051จะสามารถขัดจังหวะการทำงานได้ 6 แหล่งคือ

1. $\overline{\text{INT0}}$, $\overline{\text{INT1}}$ เป็น 2 ขาของ 8051 ที่จะรับสัญญาณจากภายนอกการขัดจังหวะจะเกิดขึ้นถ้าสัญญาณที่ขาดังกล่าวมีสภาวะลอจิกเป็น 0 หรือเปลี่ยน จาก 1เป็น 0 โดยเลือกด้วยการกำหนดในบิท IT0 หรือ IT1 ในรีจิสเตอร์ TCON

2. TF0, TF1 เป็นบิทหนึ่งที่จะบอกการทำงานของ Timer 0, Timer 1 เมื่อเกิด Overflow ขึ้นใน Timer จะทำให้บิทนี้เป็น 1 และเกิดการขัดจังหวะการทำงานของ 8051 ได้

3. TI, RI เป็น 2 บิท ในรีจิสเตอร์ SCON ถ้าบิทนี้ถูกเซ็ทให้เป็น 1 โดยฮาร์ดแวร์อันเนื่องจากเสร็จสิ้นการส่งหรือรับข้อมูลจะสามารถทำให้เกิดการขัดจังหวะได้



รูปที่ 3.19 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ

8051 จะทำการอ่านสัญญาณจากทั้ง 6 แหล่งที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่องเข้าเก็บ และในช่วงของไซเคิลของเครื่องถัดไปก็จะตรวจสอบสถานะของสัญญาณทั้ง 6 ที่เก็บเข้ามา ถ้าสัญญาณนั้นมีการขัดจังหวะที่ถูกต้อง 8051 ก็จะละทิ้งการทำงานเดิมไว้ชั่วคราว แล้วสร้างคำสั่ง LCALL ขึ้นมาภายใน 8051 เพื่อไปทำงานในโปรแกรมตอบสนองการขัดจังหวะแต่ละสัญญาณนั้น เมื่อทำงานในโปรแกรมตอบสนองการขัดจังหวะเสร็จสิ้นก็จะสามารถกลับมาทำงานเดิมได้ โดยใช้คำสั่ง RETI เป็นคำสั่งสุดท้ายในโปรแกรมตอบสนองการขัดจังหวะ สัญญาณขัดจังหวะจากแต่ละแหล่งจะมีตำแหน่งหน่วยความจำที่จะเก็บโปรแกรม

ตอบสนองการขัดจังหวะไว้ต่างกันดังนี้

สัญญาณที่ขอขัดจังหวะ	ตำแหน่งเริ่มต้นโปรแกรมตอบสนองการขัดจังหวะ
1 INT0	0003H
2 TF0	000BH
3 INT1	0013H
4 TF1	001BH
5 TI,RI	0023H

เมื่อ 8051 ทำการตรวจสอบสัญญาณการขอขัดจังหวะ ที่เก็บเข้ามาเมื่อเวลา S5P2 แล้วพบว่ามีการขอขัดจังหวะนั้น แม้ว่าจะมีการ Enable ในรีจิสเตอร์ IE ถูกต้อง แต่จะต้องมีเงื่อนไขดังนี้ด้วย

1. ไม่ได้กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะซึ่งมีลำดับความสำคัญสูงกว่าหรือเท่ากัน เช่น กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณ $\overline{\text{INT0}}$ อยู่แล้วมีการขอขัดจังหวะจากสัญญาณ $\overline{\text{INT1}}$ อีก จะไม่เกิดการทิ้งงานเดิม คือไม่มีการไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะของสัญญาณ $\overline{\text{INT1}}$

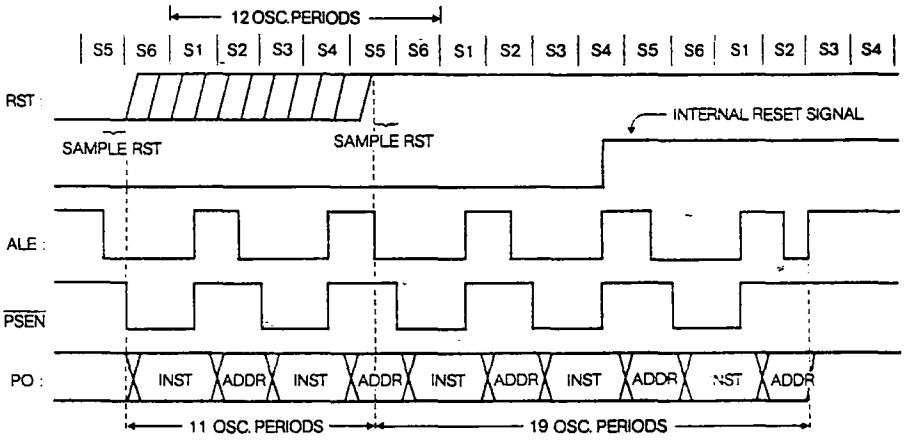
2. เนื่องจากการสุ่มสัญญาณเข้าไปเพื่อตรวจสอบนั้นจะทำที่เวลา S5P2 ของในไซเคิลสุดท้ายของคำสั่ง และคำสั่งที่อยู่ถัดมาจะต้องใช้เวลาการทำงาน 2 ไซเคิลของเครื่อง ดังนั้นการตรวจสอบจะกระทำในไซเคิลแรก แม้ว่าจะมีการขอการขัดจังหวะเข้ามาก็จะไม่ทำโปรแกรมตอบสนองการขัดจังหวะ จะต้องอ่านสัญญาณที่เวลา S5P2 อีกครั้งแล้วไปตรวจสอบที่ไซเคิลที่ 2 ของคำสั่งถ้ามีการขอการขัดจังหวะถูกต้องจึงจะข้ามไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ

3. คำสั่งที่กำลังทำงานอยู่ขณะที่ตรวจสอบสัญญาณขอขัดจังหวะ จะต้องไม่ใช่คำสั่ง RET หรือคำสั่งใดๆ ก็ตามที่พยายามเขียนข้อมูลไปยังรีจิสเตอร์ IE หรือ IP

สัญญาณขอขัดจังหวะที่ถูกอ่านเข้าไปที่เวลา S5P2 นี้ไม่ว่าได้รับการตอบสนองหรือไม่ก็จะถูกทิ้งไป แล้วอ่านเข้าไปใหม่ทุกเวลา S5P2

3.5 การรีเซ็ท

เมื่อป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปทางขา RST จะไม่ได้เกิดการรีเซ็ทขึ้นทันทีทันใด แต่ลำดับการเกิดการรีเซ็ทจะแสดงได้ดังไดอะแกรมตามเวลาในรูปที่ 3.20



รูปที่ 3.20 ไดอะแกรมเวลาของการรีเซ็ท

ในรูปที่ 3.20 เป็น Timing Diagram ของการรีเซ็ทสภาวะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 เฟส 2 State 5 ของทุกๆ ไซเคิลเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซ็ทจะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์ภายใน 8051 เพื่อให้แน่ใจว่าสัญญาณรีเซ็ทจะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ทำการรีเซ็ท 8051 ออสซิลเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ทขึ้นภายในที่เวลา S4P2 ของไซเคิลเครื่องถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมาจะยังคงปรากฏที่พอร์ท จนกว่าจะเกิดการรีเซ็ทขึ้น ซึ่งต้องใช้เวลา 19 ไซเคิลของสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณรีเซ็ท ในระหว่างเวลา 19 ไซเคิลนี้จะยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

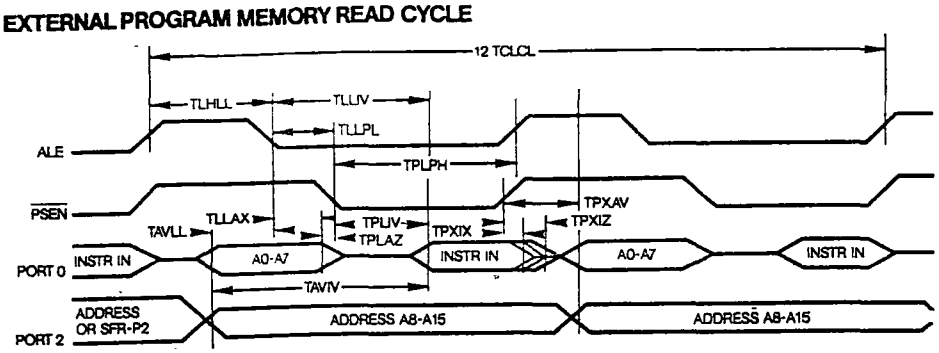
สภาวะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุกไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็จะยังไม่เกิดการตรวจสอบสัญญาณรีเซ็ทดังในรูปที่ 3.20 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State

ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ต่อมา ซึ่งเป็นเวลา S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของเครื่อง 8051 ก็จะต้องทำงานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ทได้ โดย 8051 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรกๆ ในคำสั่งอาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลของเครื่องสุดท้ายมีสภาวะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ทขึ้น

ที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ของไซเคิลของเครื่องถัดไป จึงจะทำให้สัญญาณรีเซ็ทภายในนั้นเปลี่ยนสภาวะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลของเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งที่เข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ท โดยการเขียนข้อมูล 0 ลงไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึง พอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูป 3.6 ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูล ในระหว่างการเขียนข้อมูลลงไปยัง SFR นั้นจะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่ง จนกว่าจะถึง S3P1 ของไซเคิลของเครื่องที่ 2 (นับตั้งแต่ไซเคิลของเครื่องที่ตรวจพบลอจิกที่ขา RST) ก็จะทำให้สภาวะลอจิกขา ALE และ $\overline{\text{PSEN}}$ ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลานับตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ $\overline{\text{PSEN}}$ ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์ เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีกหนึ่งถึง 2 ไซเคิลของเครื่อง สัญญาณ ALE และ $\overline{\text{PSEN}}$ จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

3.6 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

1. การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051



รูปที่ 3.21 ไดอะแกรมของการอ่านโปรแกรมจากหน่วยความจำภายนอก

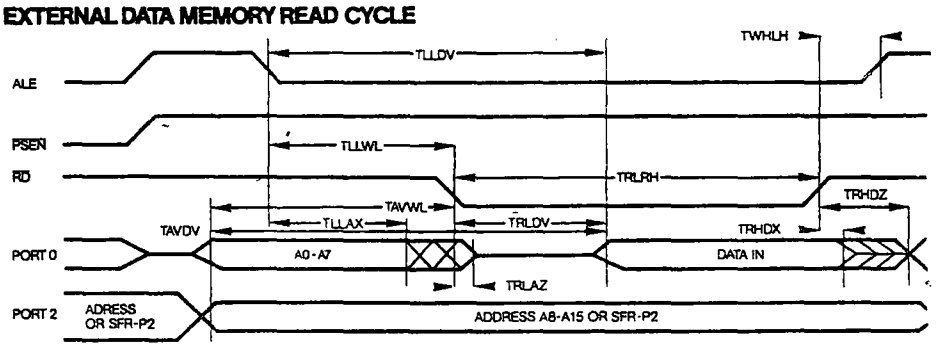
จากรูปที่ 3.21 จะเป็นลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่ง โดยการอ่านคำสั่ง (Fetch) จาก Program Area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา $\overline{\text{PSEN}}$ จะเป็น 1 จากนั้นพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่าง และพอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาแล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์จะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยจำเข้าสู่ High Impedance และสัญญาณ $\overline{\text{PSEN}}$ จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register แล้วสัญญาณ $\overline{\text{PSEN}}$ จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลาตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ไบท์แรกนั่นเอง

2. การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 8051

การอ่าน-เขียนข้อมูลกับ Data Memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับ Data Memory ภายนอกอันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าว

ถูกอ่านเข้ามายัง Instruction Register แล้ว Timing and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

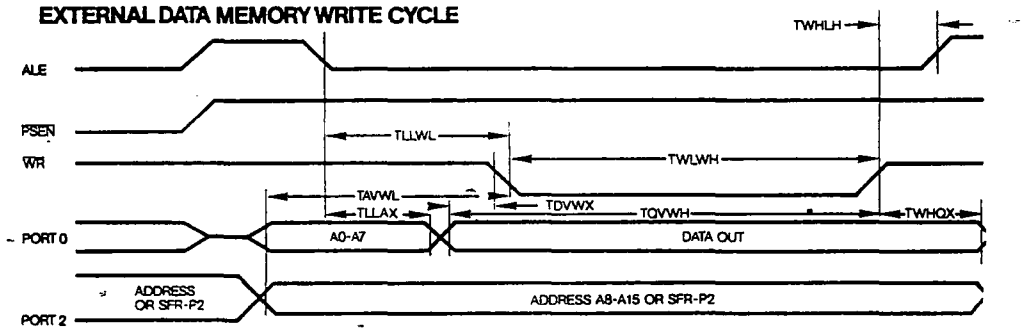
- การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 3.22



รูปที่ 3.22 ไดอะแกรมเวลาของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 8051

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิทล่างออกทางพอร์ท 0 และ 8 บิทบนออกทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้วสัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนพอร์ท 0 จะเปลี่ยนแปลงไปก็จะยังมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำ ในระหว่างการติดต่อกับ Data Memory นี้สัญญาณ $\overline{\text{PSEN}}$ จะเป็น 1 ตลอด เพราะสัญญาณ $\overline{\text{PSEN}}$ จะActive (เป็น0) ก็ต่อเมื่อเป็นการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เท่านั้น 8051 จะส่งสัญญาณลอจิก 0 ออกมาทางขา $\overline{\text{RD}}$ (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไป เมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 จะทำให้พอร์ท 0 เข้าสู่สถานะ High Impedance พร้อมที่จะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ข้อมูลบนพอร์ท 0 ซึ่งส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ $\overline{\text{RD}}$ จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไป ในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้พอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาตลอดเวลา

- การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 มีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 3.23



รูปที่ 3.23 ไดอะแกรมเวลาการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

เมื่อ 8051 ส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างไปทางพอร์ท 0 และ 8 บิทบนลงไปทางพอร์ท 2 แล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว 8051 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ท 0 แล้วจะให้สัญญาณ $\overline{WR}$ เปลี่ยนสภาวะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ $\overline{WR}$ จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน

บทที่ 4

เครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์

ในบทนี้จะกล่าวถึงรายละเอียดและหลักการเบื้องต้นของส่วนประกอบต่างๆ ของเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์ โดยเราจะแบ่งโครงสร้างของเครื่องนับออกเป็น 2 ส่วนตามลักษณะการใช้งาน ทั้งนี้เนื่องจากเครื่องนับที่สร้างขึ้นนี้สามารถจะใช้งานได้ 2 ลักษณะดังนี้

1. ใช้ในการนับวัดปริมาณรังสีได้
2. ใช้ในการนับความถี่ของสัญญาณชนิดต่างๆ

จากรูปที่ 4.1 จะแสดงถึงโครงสร้างของเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งถูกแบ่งออกเป็น 2 ส่วน

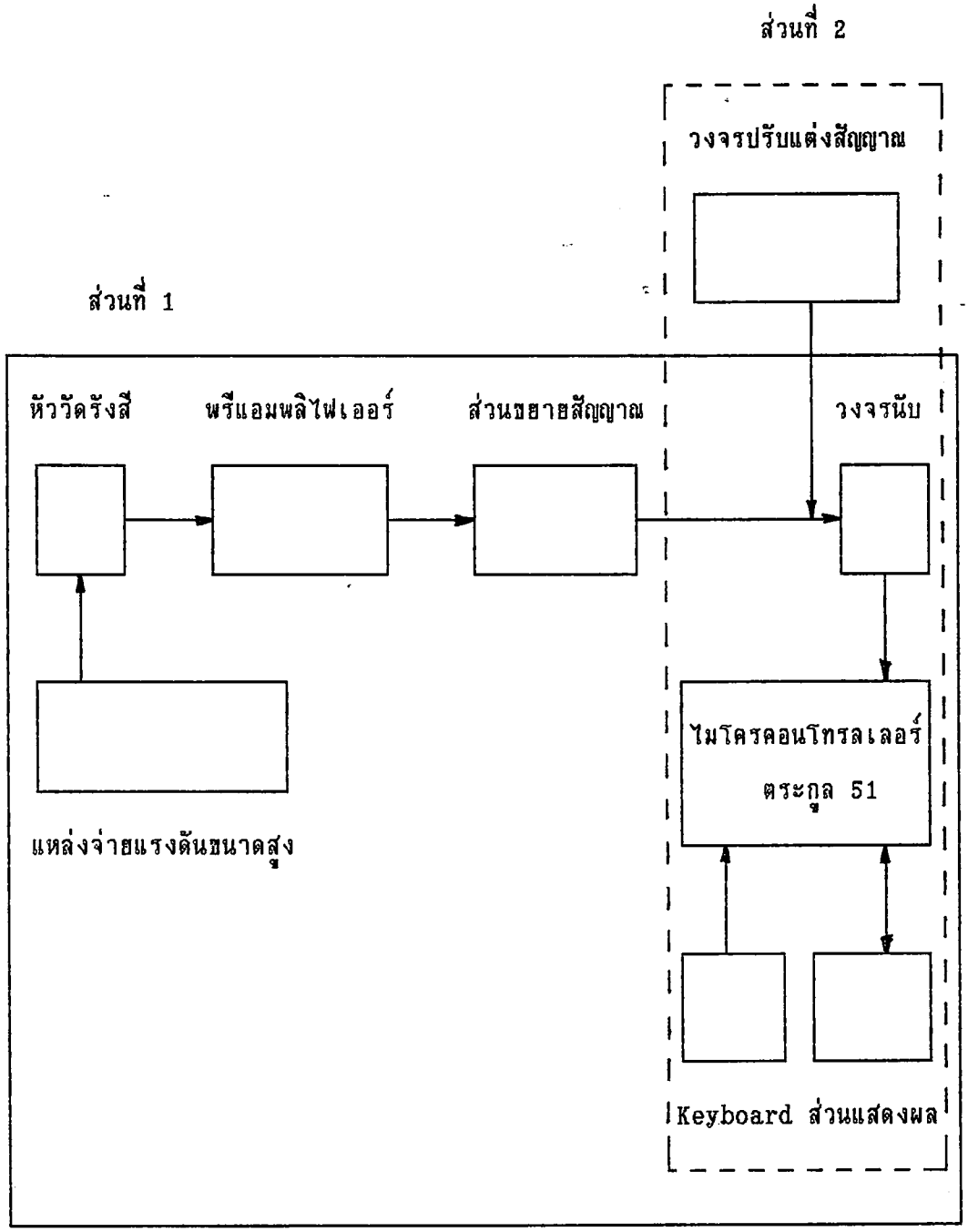
ส่วนที่ 1 คือส่วนที่ล้อมรอบด้วยเส้นทึบ จะเป็นส่วนที่ใช้ในการนับวัดปริมาณรังสีที่ถูกตรวจจับโดยหัววัดรังสี สำหรับส่วนนี้จะประกอบด้วย หัววัดรังสี, แหล่งจ่ายแรงดันไฟฟ้าขนาดสูง, พรีแอมพลิไฟเออร์, วงจรนับ, ไมโครคอนโทรลเลอร์ตระกูล 51, ส่วนแสดงผล และ Keyboard

ส่วนที่ 2 คือส่วนที่ล้อมรอบด้วยเส้นประ ส่วนนี้ใช้ในการนับความถี่ของสัญญาณที่ต่อเข้าจากภายนอก ซึ่งสามารถนับได้ทั้งสัญญาณรูปไซน์, รูปสามเหลี่ยมและรูปสี่เหลี่ยม ในส่วนนี้จะประกอบด้วย วงจรปรับแต่งสัญญาณ, วงจรนับ, ไมโครคอนโทรลเลอร์ตระกูล 51, ส่วนแสดงผล และ Keyboard

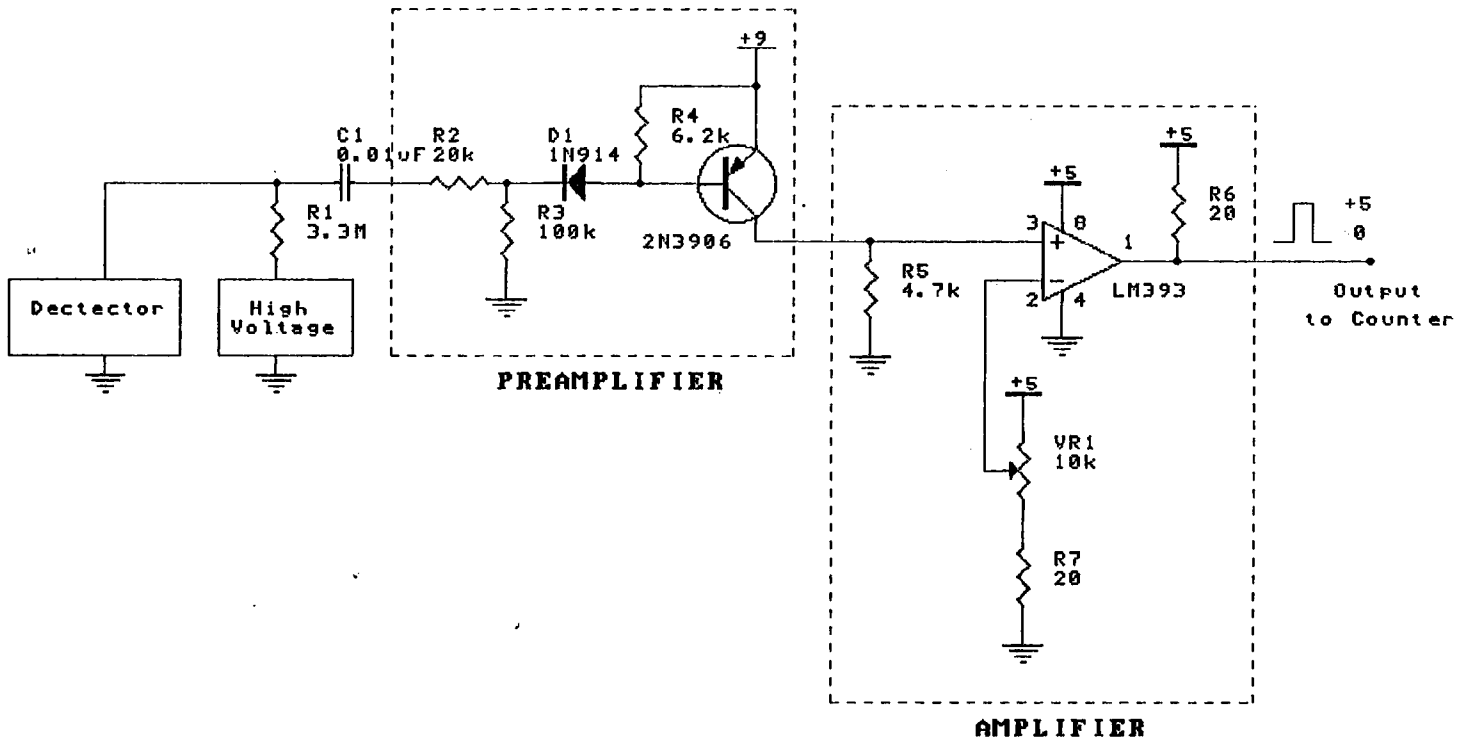
สังเกตได้ว่าโครงสร้างในส่วนที่ 1 และส่วนที่ 2 จะมีส่วนประกอบที่เหมือนกันอยู่ 4 อย่างคือ วงจรนับ, ไมโครคอนโทรลเลอร์ตระกูล 51, ส่วนแสดงผล และ Keyboard โดยส่วนประกอบทั้ง 4 ส่วนนี้จะทำงานเหมือนกันในโครงสร้างทั้ง 2 ส่วน ดังนั้นจึงจะขอกล่าวถึงรายละเอียดของส่วนประกอบของโครงสร้างทั้งสองส่วนแบบรวมๆ ดังต่อไปนี้

1. หัววัดรังสี

หัววัดที่ใช้กับเครื่องวัดนี้จะเป็นหัววัดรังสีแบบไกเกอร์-มุลเลอร์ ซึ่งทำหน้าที่ตรวจจับรังสี และมีหลักการทำงานดังได้กล่าวมาแล้วในบทที่ 2 หัววัดรังสีนี้จะถูกป้อนด้วยแรงดันไฟฟ้าขนาดสูงจากแหล่งจ่ายแรงดันขนาดสูง ดังรูปที่ 4.2 และเนื่องจากขั้วอาโนดของหัววัดรังสีจะมีค่าความต่างศักย์ค่าบวกสูงมากเมื่อเทียบกับกราวน์ ดังนั้นจึงต้องต่อตัวเก็บ



รูปที่ 4.1 บล็อกไดอะแกรมแสดงโครงสร้างของเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์



รูปที่ 4.2 แผนภาพแสดงส่วนของหัววัดรังสี, พรีแอมพลิไฟเออร์และส่วนขยายสัญญาณ

ประจุ C_1 เพื่อกันค่าความต่างศักย์ไฟตรงก่อนที่สัญญาณจะเข้าสู่วงจรนับ ส่วนตัวต้านทาน R_2 จะถูกต่ออนุกรมกับแหล่งจ่ายแรงดันไฟฟ้าขนาดสูงเพื่อแยกแหล่งจ่ายแรงดันกับขั้วอาโนด และสัญญาณที่ออกจาก C_1 จะมีลักษณะเป็นพัลส์ลบ

2. พรีแอมพลิไฟเออร์ (Preamplifier)

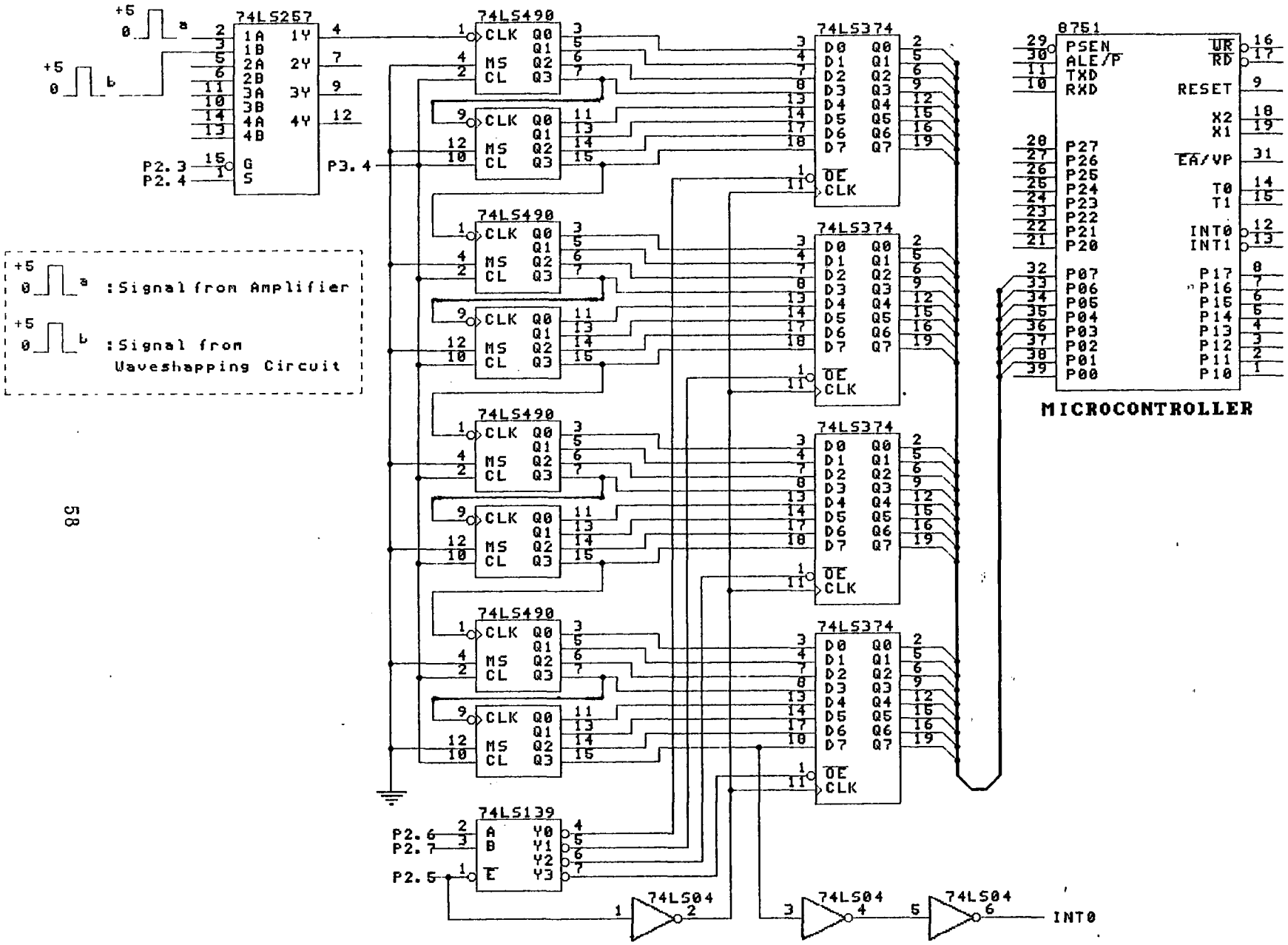
วงจรในส่วนนี้ ทำหน้าที่เป็นตัวปรับค่าอิมพีแดนซ์ระหว่างหัววัดรังสีกับส่วนขยาย และยังทำการแปลงพัลส์ลบที่ออกจากส่วนของหัววัดรังสีให้กลายเป็นพัลส์บวกอีกด้วย สำหรับวงจรของพรีแอมพลิไฟเออร์ที่ใช้จะถูกแสดงไว้ในรูปที่ 4.2 โดยวงจรนี้จะถูกต่ออยู่ระหว่างส่วนของหัววัดรังสีกับส่วนขยาย

3. ส่วนขยายสัญญาณ (Amplifier)

สำหรับส่วนขยายสัญญาณนั้นจะเป็นส่วนที่รับสัญญาณพัลส์บวกที่ได้จากเอาท์พุทของพรีแอมพลิไฟเออร์มาเปรียบเทียบกับค่าแรงดันอ้างอิงโดยใช้วงจร Comparator แล้วทำการจัดรูปร่างสัญญาณให้อยู่ในรูปของสัญญาณแบบดิจิตอลดังในรูปที่ 4.2 ทั้งนี้จะสามารถปรับค่าแรงดันอ้างอิงเพื่อให้วงจรขยายสามารถรับสัญญาณได้ในช่วงที่กว้างขึ้น ก็โดยการปรับค่าความต้านทานปรับค่าได้ (VR1) และสัญญาณเอาท์พุทที่ได้จากส่วนขยายนี้จะถูกส่งเข้าสู่วงจรนับต่อไป

4. วงจรนับ (Counter)

วงจรนับเป็นส่วนที่ถูกออกแบบให้ใช้ได้กับอินพุท 2 ช่องดังรูปที่ 4.3 โดยมี IC เบอร์ 74LS257 (2 -Line-to-1-Line Data Seletor) เป็นตัว Multiplex เลือกสัญญาณอินพุทระหว่างขา 1A กับ 1B สัญญาณใดสัญญาณหนึ่ง โดยสัญญาณอินพุทที่เข้ามาทางขา 1A จะเป็นสัญญาณที่ได้มาจากส่วนขยาย ส่วนสัญญาณที่เข้ามาทางขา 1B เป็นสัญญาณที่ได้จากวงจรปรับแต่งสัญญาณ สัญญาณที่ถูกเลือกมานี้จะถูกส่งเข้าขา CK ของ IC เบอร์ 74LS490 ซึ่งมีวงจรหารสิบ 2 ตัว เมื่อนำขาสัญญาณ Q_3 ไปต่อกับขา CK ของวงจรหารสิบตัวต่อไปก็จะสามารถนับสัญญาณได้หลายหลักขึ้น สำหรับวงจรที่สร้างขึ้นนี้จะออกแบบให้นับได้ 8 หลัก จึงต้องใช้ IC เบอร์ 74LS490 จำนวน 4 ตัว ส่วนขา CL ของ 74LS490 จะต่อถึงกันทั้งหมดเพื่อใช้ในการรีเซ็ทข้อมูลในวงจรนับ ขา Q_0, Q_1, Q_2 และ Q_3 ของวงจรหารสิบ 2 ตัว (IC 74LS490 หนึ่งตัว) จะถูกต่อเข้ากับขา Data in (D_0 - D_7) ของ IC เบอร์ 74LS374 (Latch) 1ตัว ทำเช่นเดียวกันทั้ง 4 ตัว และเราสามารถคงค่า Count ได้โดยการควบคุมขา CK ของ Latch ที่ต่อถึงกันทั้ง 4 ตัว ส่วนขา G ของLatchจะต่ออยู่กับ IC เบอร์ 74LS139 (Decoder) เพื่อเลือกให้สัญญาณออกของ Latch เป็นตัวใดก็ได้ใน 4 ตัว โดยการควบคุมขา A และ B ของ Decoder ส่วนขา E นอกจากจะใช้เป็นตัว

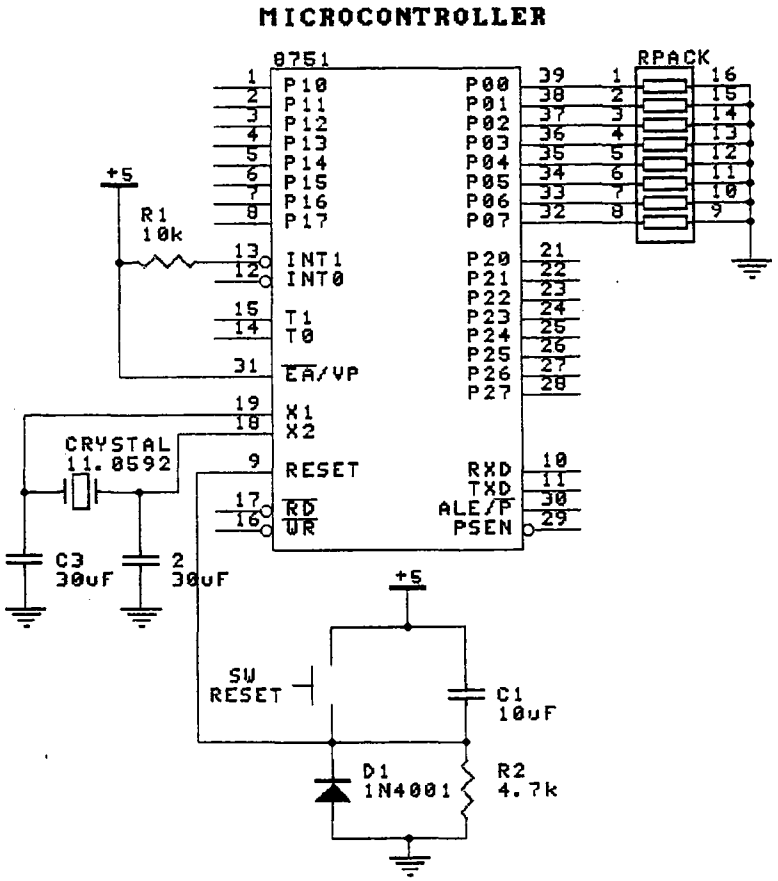


รูปที่ 4.3 แผนภาพของวงจรนับที่ต่อเข้ากับไมโครคอนโทรลเลอร์

เลือกให้มีข้อมูลออกผ่าน Data Bus หรือให้อยู่ในสภาวะ High Impedance แล้วยังถูกใช้เป็นตัว Latch ข้อมูลของ 74LS374 อีกด้วยโดยการผ่าน Not Gate แล้วต่อเข้ากับขา G ของLatch นั่นเอง ขา Q_3 ของวงจรหารสิบตัวสุดท้ายจะถูกนำไปใช้ในการ Interrupt ไมโครคอนโทรลเลอร์เพื่อบอกให้ทราบว่านับจนเต็มสเกลแล้ว โดยต่อเข้ากับ Not Gate 2 ตัว (ซึ่งเปรียบเสมือนเป็น Buffer) ที่ต่ออยู่กับ INT0

5. ไมโครคอนโทรลเลอร์ตระกูล 51

ในที่นี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ 8751 โดยต่อขาของไมโครคอนโทรลเลอร์ดังรูปที่ 4.4 นอกจากนี้ยังต้องต่อขาของไมโครคอนโทรลเลอร์เข้ากับส่วนของวงจรนับเพื่อควบคุมการนับดังรูปที่ 4.3 , ต่อเข้ากับส่วนของ Keyboard เพื่อรับอินพุทเข้ามาดัง

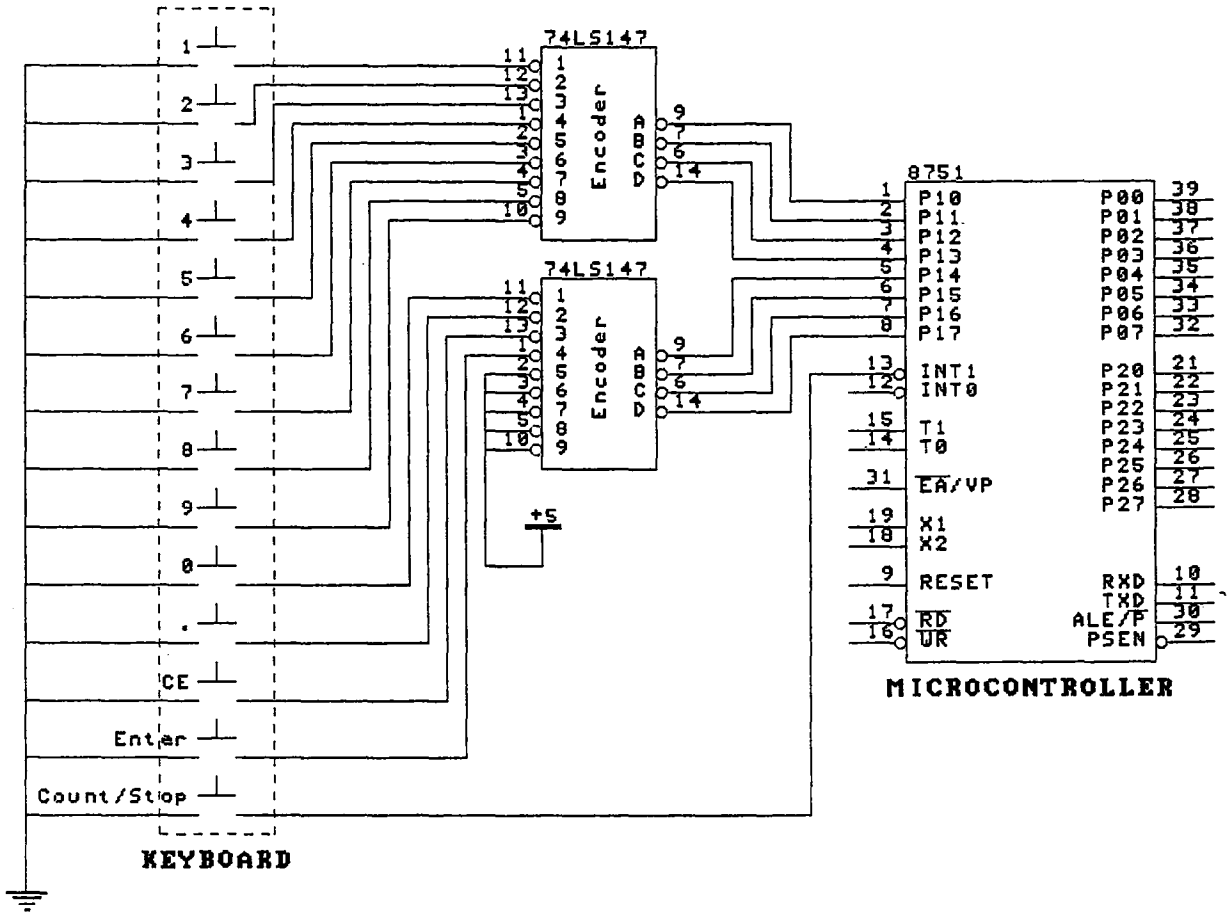


รูปที่ 4.4 แผนภาพแสดงการต่ออุปกรณ์ต่างๆ เข้ากับไมโครคอนโทรลเลอร์

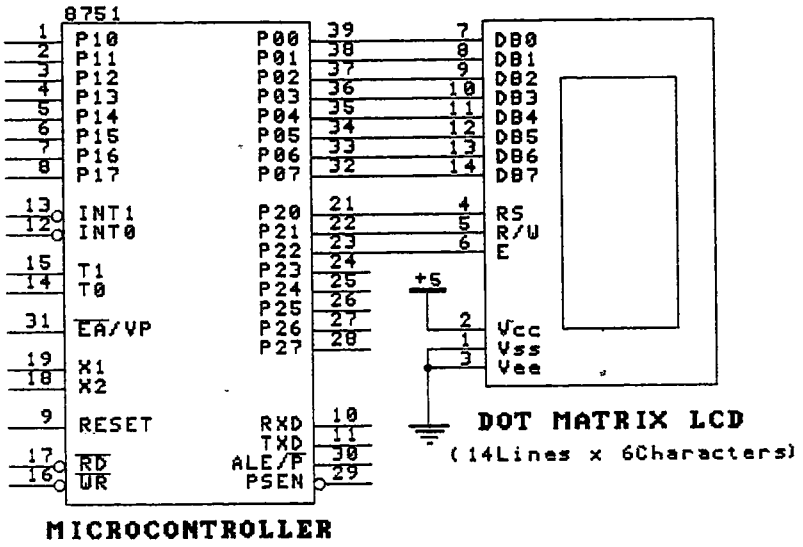
รูปที่ 4.5 และต่อเข้ากับส่วนแสดงผล (Dot Matrix LCD) ซึ่งเป็นส่วนของเอาท์พุทดังรูปที่ 4.6 โดยเราต้องเขียนโปรแกรมเพื่อควบคุมวงจรนับ, รับสัญญาณอินพุทจาก Keyboard, ควบคุมส่วนแสดงผล, รับการ Interrupt รวมทั้งตั้งเวลาด้วย ซึ่งสำหรับแนวทางการเขียนโปรแกรมและตัวโปรแกรมจะแสดงไว้ในภาคผนวก

6. Keyboard

เป็นส่วนอินพุทที่ถูกออกแบบให้มี คีย์ 0-9 , "." , Clear Entry (CE), Enter และ Count/Stop ดังรูปที่ 4.5 โดยที่เราต้องต่อ Keyboard ผ่าน 74LS147 ซึ่งเป็นตัวเข้ารหัสให้เป็น BCD ส่งเข้าไปยัง Port 1 ทั้ง 8 บิท ยกเว้นคีย์ Count/Stop ที่ออกแบบให้เป็นสัญญาณ Interrupt เพื่อให้มีความรวดเร็วในการติดต่อ



รูปที่ 4.5 แผนภาพแสดงการต่อ Keyboard เข้ากับไมโครคอนโทรลเลอร์



รูปที่ 4.6 แผนภาพแสดงการต่อ Dot Matrix LCD เข้ากับไมโครคอนโทรลเลอร์

7. ส่วนแสดงผล

เป็นส่วนแสดงผลแบบ Dot Matrix LCD Module ขนาด4บรรทัด16ตัวอักษร ซึ่งจะถูกต่อเข้าไมโครคอนโทรลเลอร์ดังรูปที่ 4.6 และในตัวมันจะประกอบด้วยส่วนประกอบใหญ่ๆ แบ่งได้เป็น

1. Dot Matrix LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดและการเปิดตัวเองกับแสง ก็คือส่วนที่เป็นตัวกระจกบรรจุผลึก

2. Driver เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่ง

3. Controller เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมา และจัดการควบคุม LCD Module ให้ทำงานแสดงผลต่างๆ เช่น การลบจอ, การสร้างตัวอักษร เป็นต้น

ขาที่ต่อใช้งานของส่วน Display LCD มีดังต่อไปนี้

1. RS (Register Selection) จะเป็นขาเลือก Register ภายในซึ่งมีอยู่ 2 ตัวคือ Insruction Register (IR) และ Data Register (DR) โดยถ้าเป็น 1 จะเป็นการเลือก Data และถ้าเป็น 0 จะเป็นการเลือก Instruction

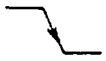
2. R/W (Read/Write) เป็นตัวเลือกว่าจะเขียนหรืออ่านข้อมูลจากตัว IC

โดยอ่านข้อมูล = 1, เขียนข้อมูล =0

3. E (Enable Signal) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล

4. DB0-DB7 เป็นขารับส่งข้อมูลจากตัวไมโครคอนโทรลเลอร์

The relation between the operation and the combination of RS, R/W

RS	RW	E	OPERATION
0	0		Write instruction code
0	1		Read busy flag and address counter
1	0		Write data
1	1		Read data

When performing data and instruction code by 4 bit, transfer RS, R/W every time.

รูปที่ 4.7 การควบคุมขาสัญญาณเพื่อให้ Display LCD ทำงานในลักษณะต่างๆ

5. Vcc เป็นไฟเลี้ยงตัววงจร

6. Vss เป็นขากราวน์

7. Vee เป็นขารับ Voltage ในการขับ LCD ให้สว่างหรือมืด

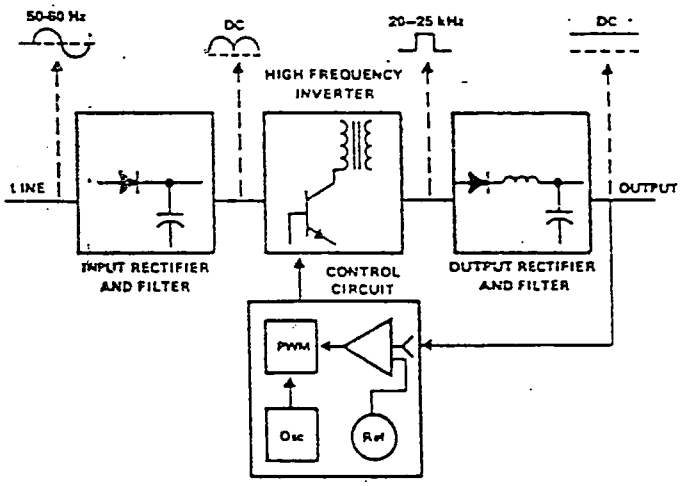
8. **แหล่งจ่ายแรงดันไฟฟ้าขนาดสูง (High Voltage)**

เป็นแหล่งจ่ายแรงดันไฟฟ้าแบบสวิตชิ่งที่มีส่วนประกอบหลัก 4 ส่วนด้วยกันคือ

- ภาคเรคติไฟล์และฟิลเตอร์ด้านอินพุท
- ภาคอินเวอร์เตอร์ความถี่สูง
- ภาคเรคติไฟล์และฟิลเตอร์ด้านเอาท์พุท
- วงจรภาคควบคุม

ไฟสลับแรงดันสูง จะต่อเข้ากับภาคเรคติไฟล์และฟิลเตอร์ทางด้านอินพุทเป็นไฟกระแสตรงแรงดันสูง จากนั้นวงจรอินเวอร์เตอร์จะทำการสวิตช์เปลี่ยนแรงดันไฟตรงเป็นพัลส์สี่เหลี่ยมความถี่สูง อยู่ในช่วง 20 kHz - 500 kHz ซึ่งในวงจรสวิตชิ่งนี้อาจจะสร้าง

ได้จากอุปกรณ์หลายชนิด เช่น เพาเวอร์ทรานซิสเตอร์, เพาเวอร์มอสเฟต, เอสซีอาร์และอื่นๆ พัลส์สี่เหลี่ยมแรงดันสูงจากวงจรสวิตชิ่งจะส่งเข้าหม้อแปลงความถี่สูงเพื่อให้ได้แรงดันเอาท์พุทตามต้องการและแยกกราวน์ไปในตัว แรงดันที่ได้นี้จะส่งเข้าภาคเรคติไฟล์และฟิลเตอร์ด้านเอาท์พุทอีกครั้งเป็นไฟตรงตามระดับแรงดันที่ต้องการ ส่วนหนึ่งจะถูกส่งกลับเข้ามาวงจรควบคุมเพื่อคอยตรวจสอบและควบคุมแรงดันเอาท์พุทให้คงที่ ไม่ว่าแรงดันไฟสลับด้านอินพุท หรือโหลดที่เอาท์พุทจะเปลี่ยนแปลงไปบ้างก็ตาม



รูปที่ 4.8 บล็อกไดอะแกรมแสดงการทำงานของแหล่งจ่ายไฟแบบสวิตชิ่ง

ในส่วนของวงจรควบคุมประกอบไปด้วย วงจรกำเนิดความถี่เพื่อป้อนแก่วงจรพัลส์มอดูเลเตอร์ (PWM) วงจรขยายความผิดพลาดและแรงดันอ้างอิงที่เที่ยงตรง เมื่อแรงดันเอาท์พุทเปลี่ยนแปลง วงจรขยายความผิดพลาดจะทำการเปรียบเทียบแรงดันเอาท์พุทกับแรงดันอ้างอิง แล้วส่งค่าความผิดพลาดนั้นไปทำการปรับปรุงความกว้างของพัลส์ของ (PWM) ขบวนพัลส์ที่ได้จะส่งมาควบคุมวงจรสวิตชิ่งในสวิตช์ปิด-เปิดปล่อยกำลังไฟฟ้าไปสู่โหลดให้มีค่าเปลี่ยนแปลงตามไปด้วยเพื่อรักษาระดับแรงดันเอาท์พุทให้คงที่ นอกจากนี้สวิตชิ่งเพาเวอร์ซัพพลายยังต้องมีส่วนประกอบอื่นๆ อีก เช่น วงจรกรอง RFI, วงจรป้องกันไฟกระชาก, วงจรจำกัดกระแส, วงจรป้อนแรงดันเอาท์พุทเกิน ฯลฯ

ในส่วนต่อไปที่จะกล่าวถึงนี้จะเป็นส่วนของ "ดีซีทูดีซีคอนเวอร์เตอร์" (DC-DC

Converter ซึ่งเป็นส่วนที่สำคัญจากไดอะแกรมการทำงานของสวิตชิงเพาเวอร์ซัพพลายดีซีทูดีซีคอนเวอร์เตอร์

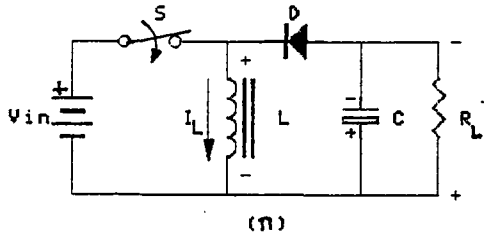
ในรูปที่ 4.8 จะเห็นได้ว่าในส่วนของดีซีทูดีซีคอนเวอร์เตอร์ประกอบด้วย

- อินเวอร์เตอร์ความถี่สูง
- ภาคเรคติไฟล์และฟิลเตอร์ด้านเอาท์พุท
- วงจรควบคุม

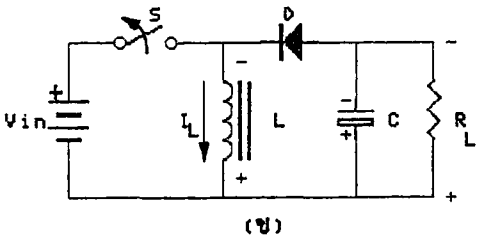
วงจรควบคุมทำหน้าที่ป้อนพัลส์สี่เหลี่ยมความถี่สูง (20 kHz - 500 kHz) ซึ่งสามารถควบคุมความกว้างของพัลส์ให้กับภาคอินเวอร์เตอร์ ในส่วนสำคัญของดีซีทูดีซีคอนเวอร์เตอร์นั้นคือ อินเวอร์เตอร์ความถี่สูงกับวงจรเรคติไฟล์และฟิลเตอร์ด้านเอาท์พุท รวมเรียกสั้นๆ ว่าคอนเวอร์เตอร์

แม้ว่าคอนเวอร์เตอร์มีอยู่หลายรูปแบบแต่ก็ล้วนมีพื้นฐานมาจากวงจร3แบบคือ

- ฟลายแบ็คหรือบั๊คบูสต์ (Flyback or Buck-boost)
- ฟอร์เวิร์ดหรือบั๊ค (Forward or Buck)
- พุช-พูลหรือบั๊คดีไรฟ์ (Push-Pull or Buck-Derived)

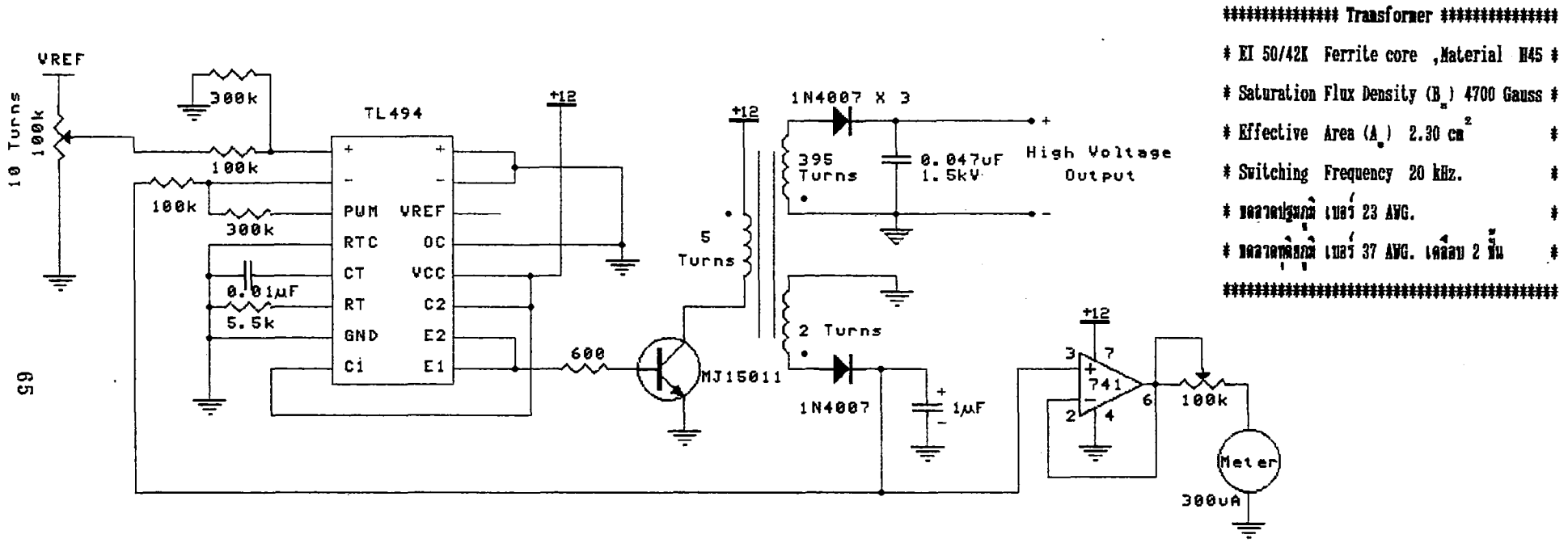


(ก)



(ข)

รูปที่ 4.9 หลักการของคอนเวอร์เตอร์แบบฟลายแบ็ค



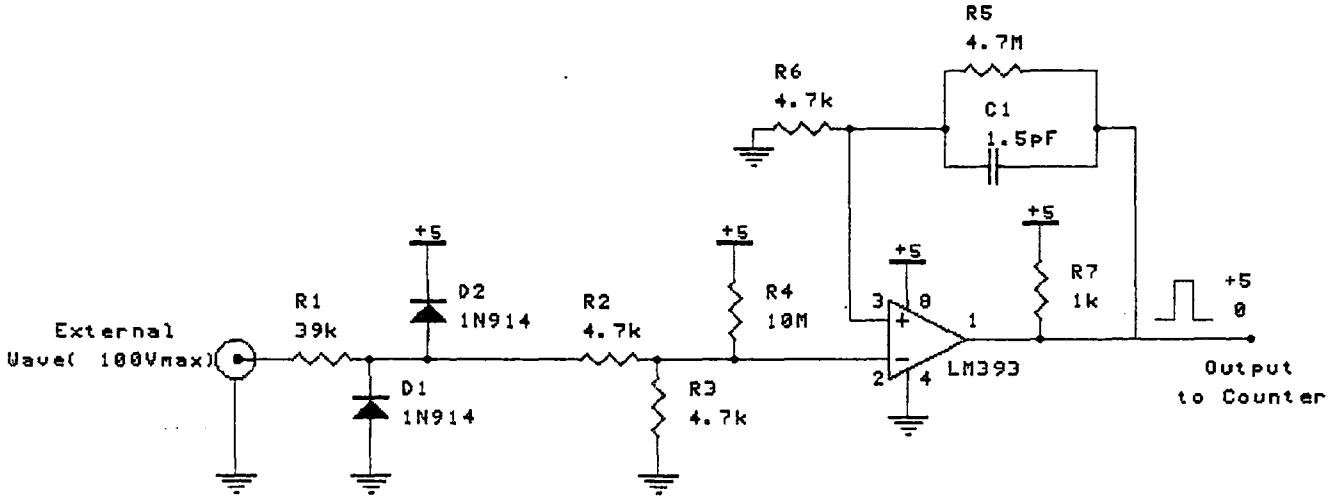
รูปที่ 4.10 วงจรของแหล่งจ่ายแรงดันไฟฟ้าขนาดสูงที่ใช้ในเครื่องนับรังสีควบคุมด้วยไมโครคอนโทรลเลอร์

รูปที่ 4.9 แสดงลักษณะของคอนเวอร์เตอร์แบบฟลายแบ็ค เมื่อสวิตช์ S ปิด รูปที่ 4.9 (ก) กระแสจะไหลผ่านตัวเหนี่ยวนำ L สะสมพลังงานในรูปของสนามแม่เหล็ก ไดโอด D จะถูกไบแอสกลับจึงยังไม่มีแรงดันตกคร่อมโหลด R_L ต่อมาเมื่อสวิตช์เปิด รูปที่ 4.3 (ข) ตัวเหนี่ยวนำก็จะคายพลังงานออกมา (เนื่องจากสนามแม่เหล็กยุบตัว) แต่ขั้วของแรงดันจะกลับกันกับที่ป้อนเข้ามา (ซึ่งเป็นไปตามธรรมชาติของตัวเหนี่ยวนำ) ทำให้ไดโอดได้รับไบแอสตรงนำกระแส จากการคายประจุกลับพลังงานของตัวเหนี่ยวนำไปสู่โหลดจะมีขั้วแรงดันตรงข้ามกับแรงดันอินพุท ถ้าสวิตช์ปิด-เปิดเป็นจังหวะอย่างต่อเนื่องกระแสอินพุทและเอาท์พุทก็จะมีลักษณะเป็นพัลส์อย่างต่อเนื่องด้วย จึงพอสรุปได้ว่าในคอนเวอร์เตอร์แบบฟลายแบ็คนั้น พลังงานจะถูกเก็บในตัวเหนี่ยวนำในระหว่างที่สวิตช์ และพลังงานจะถูกปล่อยไปสู่โหลดในระหว่างสวิตช์เปิดหรือฟลายแบ็คนั่นเอง

ในรูปที่ 4.10 จะเป็นวงจรของแหล่งจ่ายแรงดันไฟฟ้าขนาดสูงแบบสวิตชิ่งที่ใช้ในเครื่องนับรังสี แหล่งจ่ายแรงดันนี้จะถูกต่อเข้ากับหัววัดรังสีดังรูปที่ 4.2

9. วงจรปรับแต่งสัญญาณ (Waveshaping Circuit)

วงจรนี้ ทำหน้าที่รับสัญญาณจากภายนอกเข้ามาแล้วจัดรูปร่างของสัญญาณให้เป็นสัญญาณแบบดิจิตอลเพื่อส่งต่อไปให้วงจรนับทำการนับความถี่ต่อไป วงจรนี้สามารถใช้ได้กับ



รูปที่ 4.11 แผนภาพแสดงวงจรปรับแต่งสัญญาณ

อินพุทที่มีแอมพลิจูดถึง 100โวลต์ ดังรูปที่ 4.11 ความต้านทาน R_1 ที่ต่อกับไดโอด D_1 และ D_2 เป็นการกำจัดช่วงการแกว่งของสัญญาณอินพุทให้มีค่า -0.6 ถึง 5หรือ6 โวลต์ โดยใช้การประมาณค่าความต้านทานแบบ Divider R_2R_3 จะใช้ในการจำกัดค่าการแกว่งของสัญญาณในช่วงลบได้ถึง 0.3 โวลต์ซึ่งเป็นขีดจำกัดของ Comparator 393 R_5 และ R_6 จะทำให้เกิด Hyteresis โดย R_4 จะเป็นตัวตั้งจุด Trigger ที่สมมาตรกับกราวน์ ค่าอินพุทอิมพีแดนซ์นั้นเกือบจะคงที่เนื่องจากตัวต้านทาน R_1 มีค่ามากเมื่อเทียบกับความต้านทานทางขาอินพุท สาเหตุที่ใช้ LM393 เนื่องจากอินพุทสามารถใช้ได้ช่วงกว้างและการทำงานก็ใช้แหล่งจ่ายไฟเพียงตัวเดียว (+5โวลต์)

บทที่ 5

การทดลองวงจรและเครื่องนับรังสี

การทดลองที่ 5.1 **ทดสอบวงจรนับที่ควบคุมด้วยไมโครคอนโทรลเลอร์**

ตอนที่ 1 **ทดสอบการตอบสนองของวงจรนับที่ความถี่ต่างๆ**

วิธีทดลอง

1. ป้อนสัญญาณดิจิตอลความถี่คงที่จากเครื่องกำเนิดความถี่ให้กับวงจรนับ โดยเปลี่ยนความถี่จาก 10 - 2 MHz.
2. ตั้งเวลาของวงจรนับเป็น 1 วินาที
3. ทดลองนับค่า count แล้วบันทึกผลลงในตารางที่ 5.1

ผลการทดลอง

ตารางที่ 5.1 แสดงผลการทดลองที่ 5.1 ตอนที่ 1

ความถี่จากเครื่องกำเนิดความถี่ (Hz.)	ค่า count ที่นับได้จากวงจรนับ เมื่อตั้งเวลาเป็น 1 วินาที				
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	%ผิดพลาด
10	10	10	10	10	0.00
50	50	50	50	50	0.00
100	100	100	100	100	0.00
500	496	496	496	496	-0.80
1000	996	996	996	996	-0.40
5000	4996	4995	4994	4995	-0.10
10000	9993	9994	9995	9994	-0.06
50000	50000	49995	49982	49992	-0.02
100000	100146	100098	100086	100110	0.11
500000	500485	499707	499413	499868	-0.03
1000000	1001392	1001142	1000847	1001127	0.11
2000000	2002094	2002055	2002004	2002051	0.10

สรุปผลการทดลอง วงจรนับที่ควบคุมด้วยไมโครคอนโทรลเลอร์นี้ สามารถทำงานได้ดีตลอดช่วงความถี่ 0-2 MHz. โดยมีค่าเปอร์เซ็นต์ความผิดพลาดที่น้อยมาก

ตอนที่ 2 **ทดสอบการตอบสนองของวงจรนับที่เวลาค่าต่างๆ**

วิธีทดลอง

1. ป้อนสัญญาณดิจิตอลความถี่คงที่ จากเครื่องกำเนิดความถี่ให้วงจรนับ ในการทดลองนี้ใช้ความถี่ 100 kHz.
2. เปลี่ยนค่าเวลาที่ใช้นับของวงจรนับ จาก 1 ถึง 10 วินาที
3. ทดลองนับค่า count แล้วบันทึกผลลงในตารางที่ 5.2

ผลการทดลอง

ตารางที่ 5.2 แสดงผลการทดลองที่ 5.1 ตอนที่ 2

เวลาที่ใช้นับของวงจรนับ (sec.)	ค่า count เมื่อให้ความถี่จากเครื่องกำเนิดความถี่เป็น 100 kHz.					
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	count/sec.	% ผิดพลาด
1	100145	100156	100157	100153	100153.0	0.15
2	200288	200714	200321	200441	100220.5	0.22
3	300493	300515	300537	300515	100171.7	0.17
4	400704	400710	400721	400712	100178.0	0.18
5	500916	500924	500943	500928	100185.6	0.19
6	601218	601271	601282	601257	100209.5	0.21
7	701537	701542	701565	701548	100221.1	0.22
8	801715	801732	801743	801730	100216.3	0.22
9	902180	901971	901746	901965	100218.3	0.22
10	1001290	1001174	1001121	1001195	100119.5	0.12

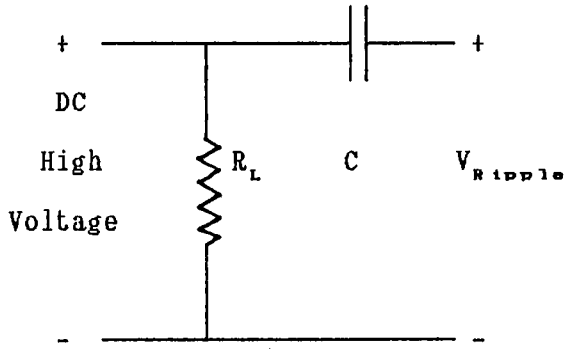
สรุปผลการทดลอง วงจรนับที่ควบคุมด้วยไมโครคอนโทรลเลอร์ สามารถนับสัญญาณแบบดิจิตอลที่มีความถี่คงที่ได้โดยไม่ขึ้นกับเวลา ซึ่งแสดงให้เห็นว่าวงจรนับนี้มีเสถียรภาพที่ดี

วิจารณ์ผลการทดลองที่ 5.1

ข้อผิดพลาดของการทดลองเนื่องมาจาก เครื่องกำเนิดความถี่ให้สัญญาณความถี่ที่ไม่คงที่ ซึ่งสังเกตได้จากส่วนแสดงผลของเครื่องนั่นเอง

การทดลองที่ 5.2 ทดสอบแหล่งจ่ายไฟกระแสตรงแรงดันสูง

วิธีทดลอง 1. ต่อวงจรดังรูปที่ 5.1 โดย C มีค่า 0.047 μF, 1.5 kV.



รูปที่ 5.1 แสดงการต่อวงจรของการทดลองที่ 5.2

2. ปลดโหลด R_L ออกจากวงจร
3. ทดลองเพิ่มค่า High Voltage จากต่ำสุดจนถึงค่าสูงสุด พร้อมทั้งบันทึกค่าแรงดันกระเพื่อมของ V_{Ripple}
4. ใส่โหลด R_L ค่าต่างๆ แล้วทำซ้ำข้อ 3. และบันทึกค่ากระแสที่ไหลผ่านโหลดด้วย

ผลการทดลอง

ตารางที่ 5.3 แสดงผลการทดลองที่ 5.2

DC High Voltage (Volts)	ไม่มีโหลด	R_L =10 เมกกะโอห์ม		R_L = 2 เมกกะโอห์ม	
	V_{Ripple} (mV.)	V_{Ripple} (mV.)	กระแส (μA.)	V_{Ripple} (mV.)	กระแส (μA.)
300	100	150	30	300	147
400	110	180	40	400	198
500	150	220	50	450	247
600	170	250	60	500	297
700	200	300	70	600	348
800	250	320	80	700	399
850	320	330	86	800	425
900	250	350	91	-	-
950	240	400	96	-	-
1000	250	-	-	-	-
1050	400	-	-	-	-

สรุปผลการทดลอง 1. วงจรแหล่งจ่ายไฟกระแสตรงแรงดันสูงมีเสถียรภาพที่ดี เนื่องจากมีการกระเพื่อมของเอาท์พุทน้อย โดยแรงดันที่กระเพื่อมนั้นขึ้นกับค่าความต้านทานของโหลด (R_L) ที่มาต่อร่วมกับวงจร ถ้าโหลดมีความต้านทานมากการกระเพื่อมจะมีค่าน้อย ถ้าความต้านทานน้อยการกระเพื่อมจะมีค่ามาก

2. กระแสที่ไหลผ่านโหลดจะเป็นไปตามความสัมพันธ์ V = IR

การทดลองที่ 5.3 ทดสอบเครื่องนับรังสีที่ควบคุมด้วยไมโครคอนโทรลเลอร์

วิธีทดลอง

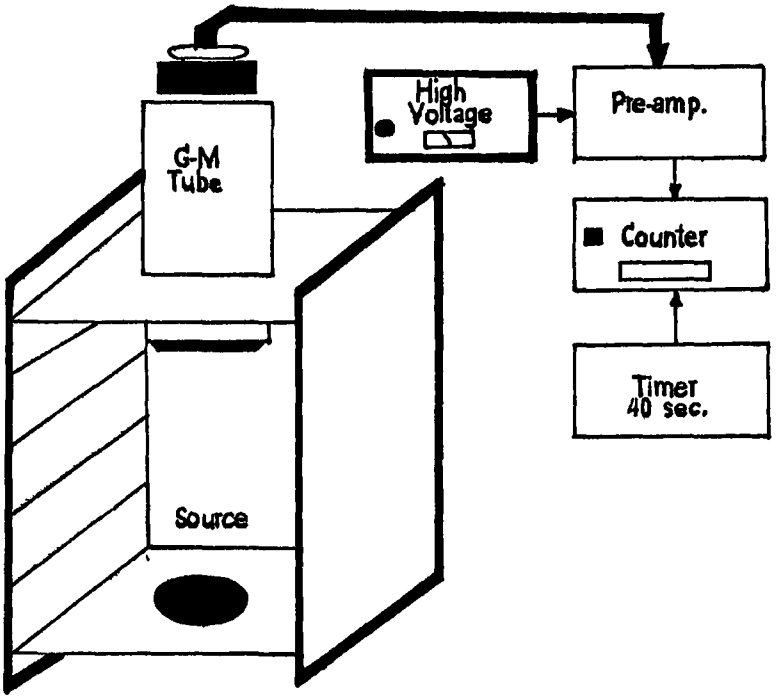
1. จัดอุปกรณ์การทดลองดังรูปที่ 5.2 โดยจัดให้แหล่งกำเนิดรังสี (Ba-133) อยู่ในชั้นที่ห่างจากหัววัดรังสี 2 เซนติเมตร
2. ตั้งเวลาของเครื่องนับรังสีอ้างอิงเป็น 40 วินาที
3. ทดลองนับรังสี แล้วบันทึกผลลงในตารางที่ 5.4
4. เลื่อนตำแหน่งของแหล่งกำเนิดรังสีไปยังชั้นที่อยู่ห่างจากหัววัด 4, 6, 8 และ 10 เซนติเมตร
5. บันทึกค่านับวัดของแต่ละตำแหน่ง
6. คำนวณค่านับวัดเฉลี่ย ($\bar{X}$) และ ส่วนเบี่ยงเบนมาตรฐาน (S.D.) ของแต่ละตำแหน่ง โดย

$$S.D. = \sqrt{\frac{\Sigma(X_i-\bar{X})^2}{N}}$$

เมื่อ N = จำนวนครั้งของการทดลอง

7. เปลี่ยนเครื่องนับรังสีอ้างอิงเป็นเครื่องนับรังสีที่สร้างขึ้น แล้วทำซ้ำตั้งแต่ต้น พร้อมทั้งบันทึกผลลงในตารางที่ 5.5
8. คำนวณค่าเปอร์เซนต์ผิดพลาดของเครื่องนับรังสีอ้างอิงและเครื่องนับรังสีที่สร้างขึ้นในแต่ละตำแหน่ง แล้วคำนวณค่าเปอร์เซนต์ผิดพลาดเฉลี่ยของเครื่องนับแต่ละชนิด โดย

$$\% \text{ ผิดพลาด} = \frac{S.D. \times 100}{\bar{X}}$$



รูปที่ 5.2 แสดงการจัดอุปกรณ์ทดลองในการทดลองที่ 5.3

ผลการทดลอง

ตารางที่ 5.4 แสดงผลการทดลองที่ได้จากเครื่องนับอ้างอิง

ระยะห่างระหว่างหัววัดกับแหล่งกำเนิดรังสี (ซม.)	ค่า count ที่นับได้จากเครื่องนับอ้างอิงในเวลา 40 วินาที						
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	$\bar{X}$	S.D.
2	2263	2267	2366	2343	2328	2313.4	41.4
4	933	893	878	939	935	915.6	25.1
6	471	460	468	444	428	454.2	16.1
8	269	273	285	270	256	270.6	9.3
10	176	197	178	172	171	178.8	9.5

ตารางที่ 5.5 แสดงผลการทดลองที่ได้จากเครื่องนับที่สร้างขึ้น

ระยะห่างระหว่างหัววัดกับแหล่งกำเนิดรังสี (ซม.)	ค่า count ที่นับได้จากเครื่องนับที่สร้างขึ้นในเวลา 40 วินาที						
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	$\bar{X}$	S.D.
2	2241	2255	2188	2243	2254	2236.2	24.7
4	958	943	907	993	990	958.2	31.9
6	430	472	475	446	508	466.2	26.7
8	262	280	279	286	267	274.8	8.9
10	194	218	207	178	191	197.6	13.7

วิเคราะห์ผลการทดลอง

ตารางที่ 5.6 เปรียบเทียบผลการทดลองของเครื่องนับอ้างอิงกับเครื่องนับที่สร้างขึ้น

ระยะห่างระหว่างหัววัดกับแหล่งกำเนิดรังสี (ซม.)	% ผิดพลาด	
	เครื่องนับอ้างอิง	เครื่องนับที่สร้างขึ้น
2	1.8	1.1
4	2.7	3.3
6	3.5	5.7
8	3.4	3.2
10	5.3	6.9
เฉลี่ย	3.34	4.04

สรุปผลการทดลอง

1. เครื่องนับรังสีที่สร้างขึ้นจะมีเปอร์เซนต์ความผิดพลาดในการนับมากกว่าเครื่องนับรังสีอ้างอิงเพียง 0.7% เท่านั้น
2. เครื่องนับที่สร้างขึ้นจะมีประสิทธิภาพด้อยกว่าเครื่องนับอ้างอิงเพียงเล็กน้อยเท่านั้น

บทที่ 6

สรุปผลการศึกษาวิจัยและข้อเสนอแนะ

สรุปผลการศึกษาวิจัย

โครงงานพิเศษนี้สามารถสร้างเครื่องนับรังสีได้ โดยมีการควบคุมส่วนของวงจรนับ, วงจรตั้งเวลา, ส่วนแสดงผล และอินพุทข้อมูลเข้า ด้วยตัวไมโครคอนโทรลเลอร์แบบชิพเดี่ยว เบอร์ 8751 สำหรับในส่วนของวงจรจ่ายไฟกระแสตรงแรงดันสูงนั้น ใช้หลักการของสวิตชิ่งเพาเวอร์ซัพพลาย ที่ควบคุมด้วย ไอซี Pulse width mod เบอร์ TL494 ส่วนวงจรพรีแอมป์พลิฟลายเออร์ และแอมป์พลิฟลายเออร์ ใช้วงจรที่สามารถแปลงค่าสัญญาณที่ได้จากหัววัดที่เป็นพัลส์ลบ เพื่อนำมาเข้าวงจรนับได้ เนื่องจากในโครงงานพิเศษนี้ใช้หัววัดแบบไกเกอร์-มูลเลอร์ จึงยังไม่ต้องมีส่วนของวงจรวิเคราะห์ความสูงของพัลส์ ซึ่งจัดว่าเป็นส่วนสำคัญในระบบการวัดรังสีอย่างหนึ่ง

นอกจากนี้ยังมีส่วนของวงจรที่เพิ่มเติมเข้าไป เพื่อนับความถี่ของรูปคลื่นภายนอก ซึ่งสามารถตรวจสอบคลื่นได้หลายรูปแบบ และหลายค่าอัมปลิจูด ทำให้ได้ใช้วงจรนับที่มีความสามารถในการนับความถี่สูงอย่างเต็มประสิทธิภาพ อีกทั้งยังนำมาประยุกต์ใช้ในทางอิเล็กทรอนิกส์ได้อีกด้วย

ข้อเสนอแนะแนวทางค้นคว้าวิจัยในขั้นต่อไป

1. เนื่องจากในโครงงานพิเศษนี้ยังพัฒนาให้ควบคุมระบบการนับรังสีทั้งระบบได้ไม่ทั้งหมด เช่น ในส่วนของวงจรจ่ายไฟกระแสตรงแรงดันสูง และวงจรขยาย จึงน่าที่จะพัฒนาให้ควบคุมได้ด้วยไมโครคอนโทรลเลอร์
2. นอกจากนี้การออกแบบระบบการวัดรังสี ที่ทำให้ผู้ที่ทำการทดลองกับสารรังสีปลอดภัยจากรังสี ด้วยการควบคุมจากระยะไกลก็สามารถกระทำได้ เนื่องจากไมโครคอนโทรลเลอร์เบอร์นี้ มีคุณสมบัติทางด้านการสื่อสารแบบ Serial อยู่ด้วย

ภาคผนวก ก.

Table 1. MCS®-51 Instruction Set Description

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Incriment indirect RAM	1	1
INC	DPTR	Incriment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

LOGICAL OPERATIONS

Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1. (Cont.)

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte form stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,1 bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,1 bit	OR complement of direct bit to Carry	2	2
MOV	C/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Byte	Cyc
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)

Mnemonic		Description	Byte	Cyc
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct. rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	– Working register R0-R7
direct	– 128 internal RAM locations, any I/O port, control or status register
@Ri	– Indirect internal RAM location addressed by register R0 or R1
#data	– 8-bit constant included in instruction
#data 16	– 16-bit constant included as bytes 2 & 3 of instruction
bit	– 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr 16	– Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11	– Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel	– SJMP and all conditional jumps include an 8-bit offset byte. Range is +127–128 bytes relative to first byte of the following instruction.

All mnemonics copyrighted © Intel Corporation 1979

Table 2. Instruction Opcodes In Hexadecimal Order

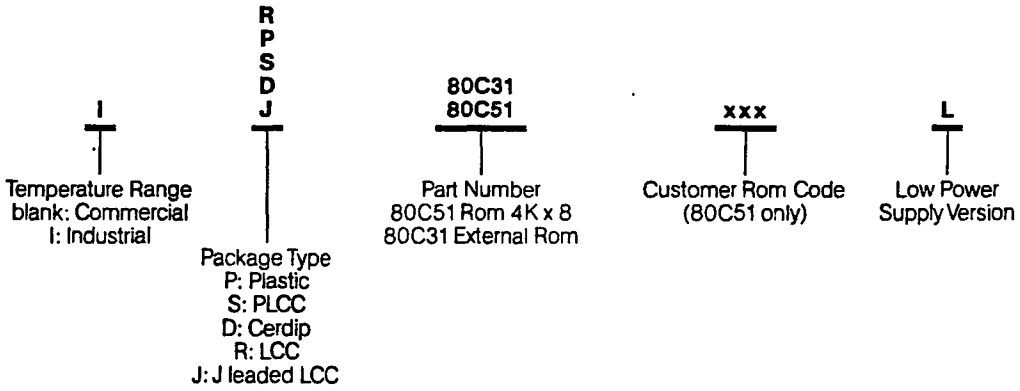
Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C, bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

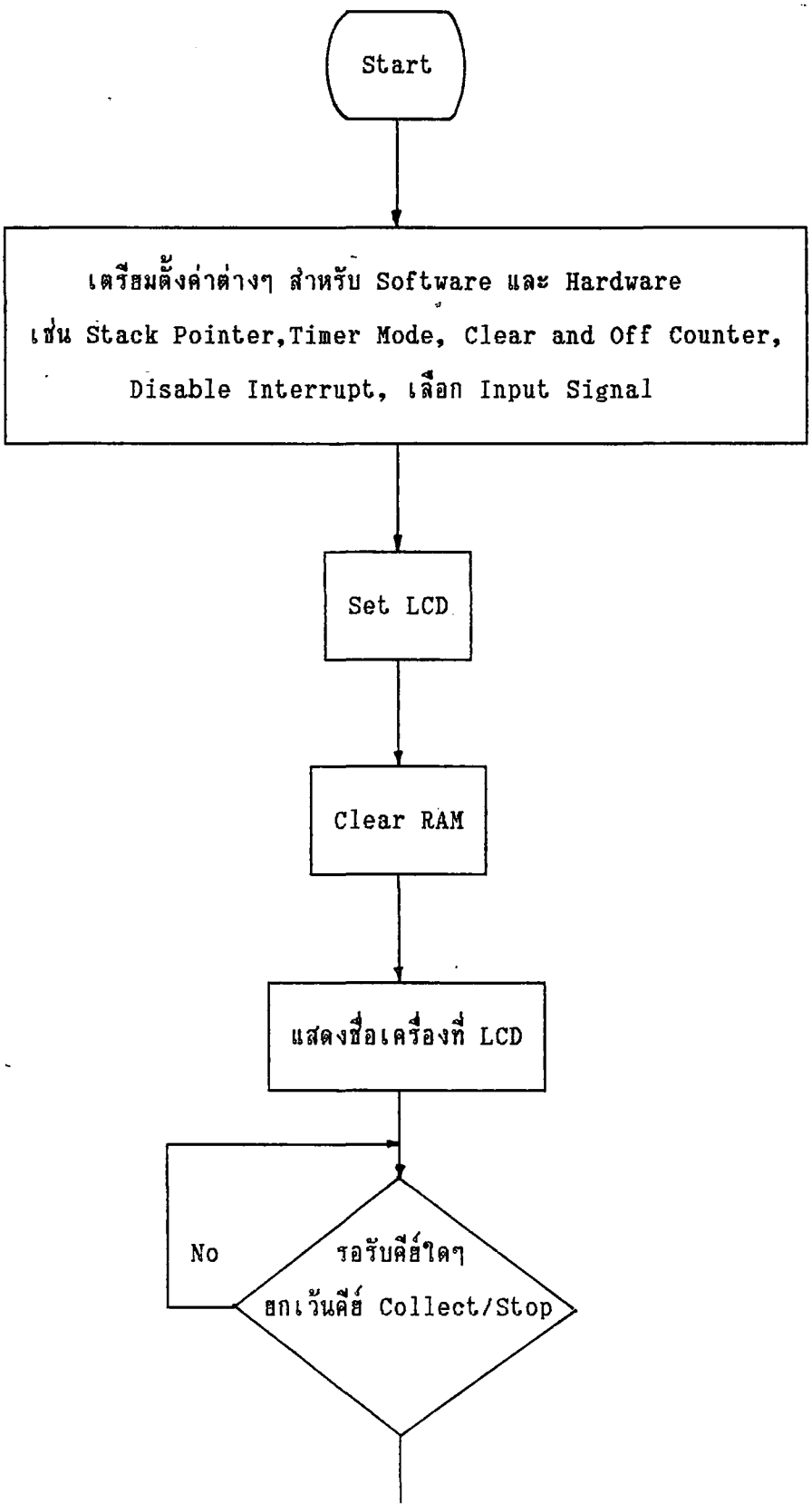
Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

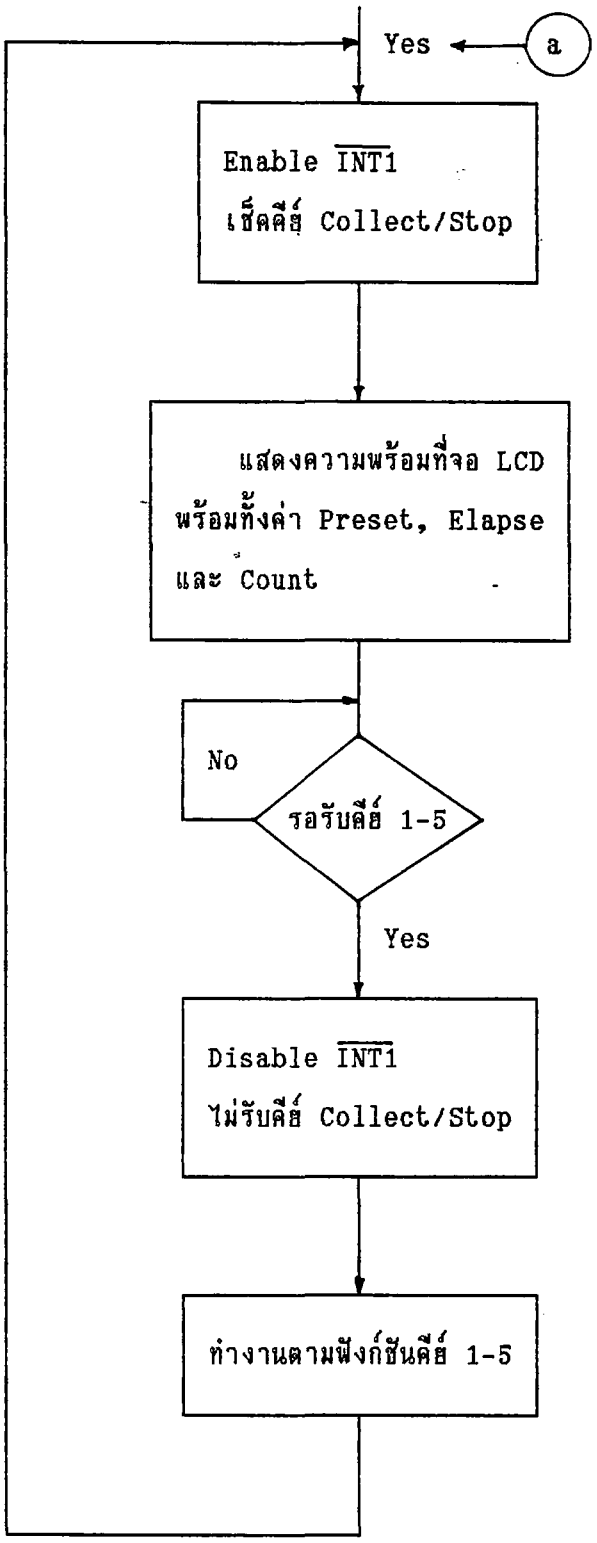


ภาคผนวก ข.

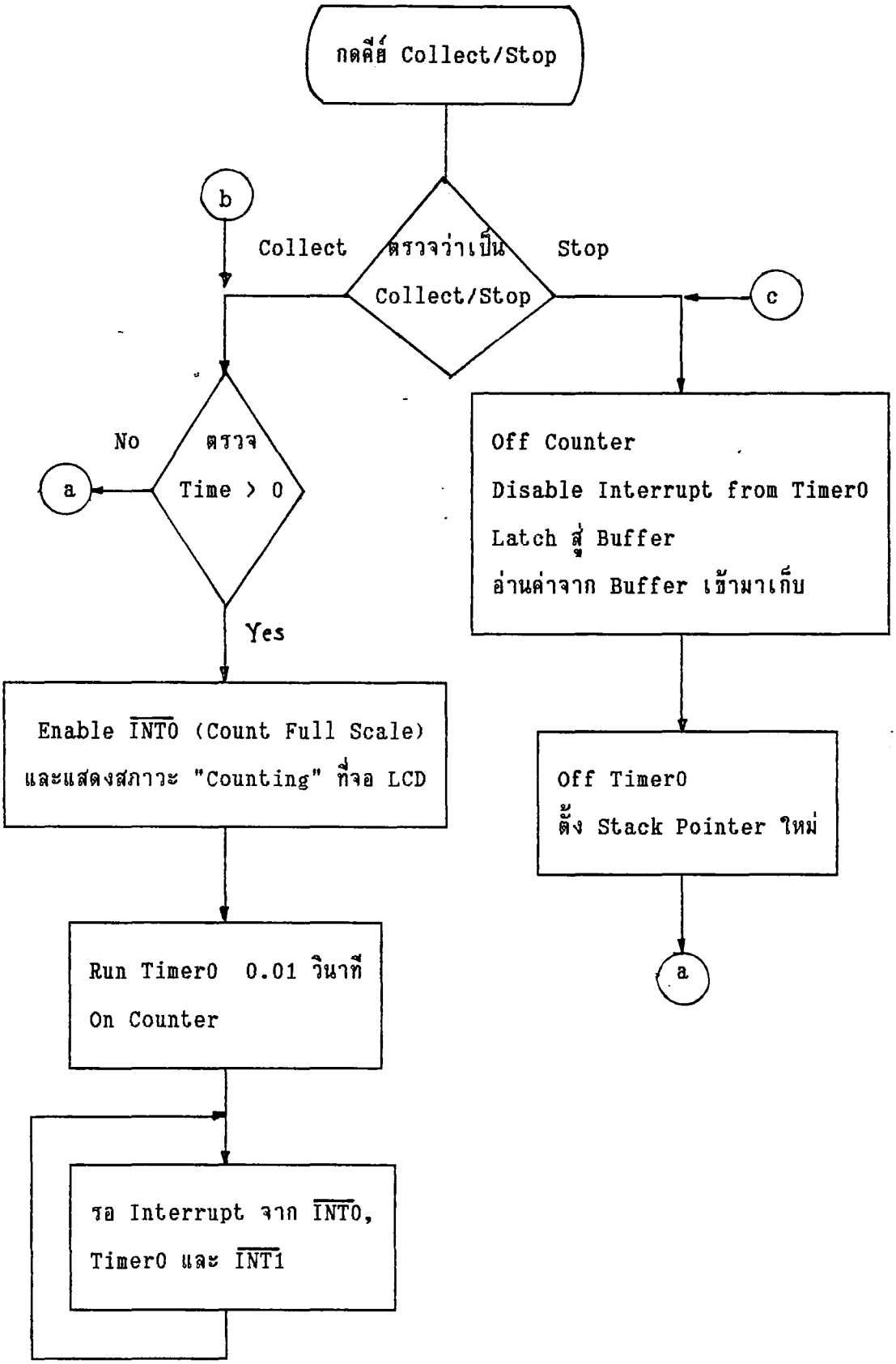
แนวทางการเขียนโปรแกรม

Main Program

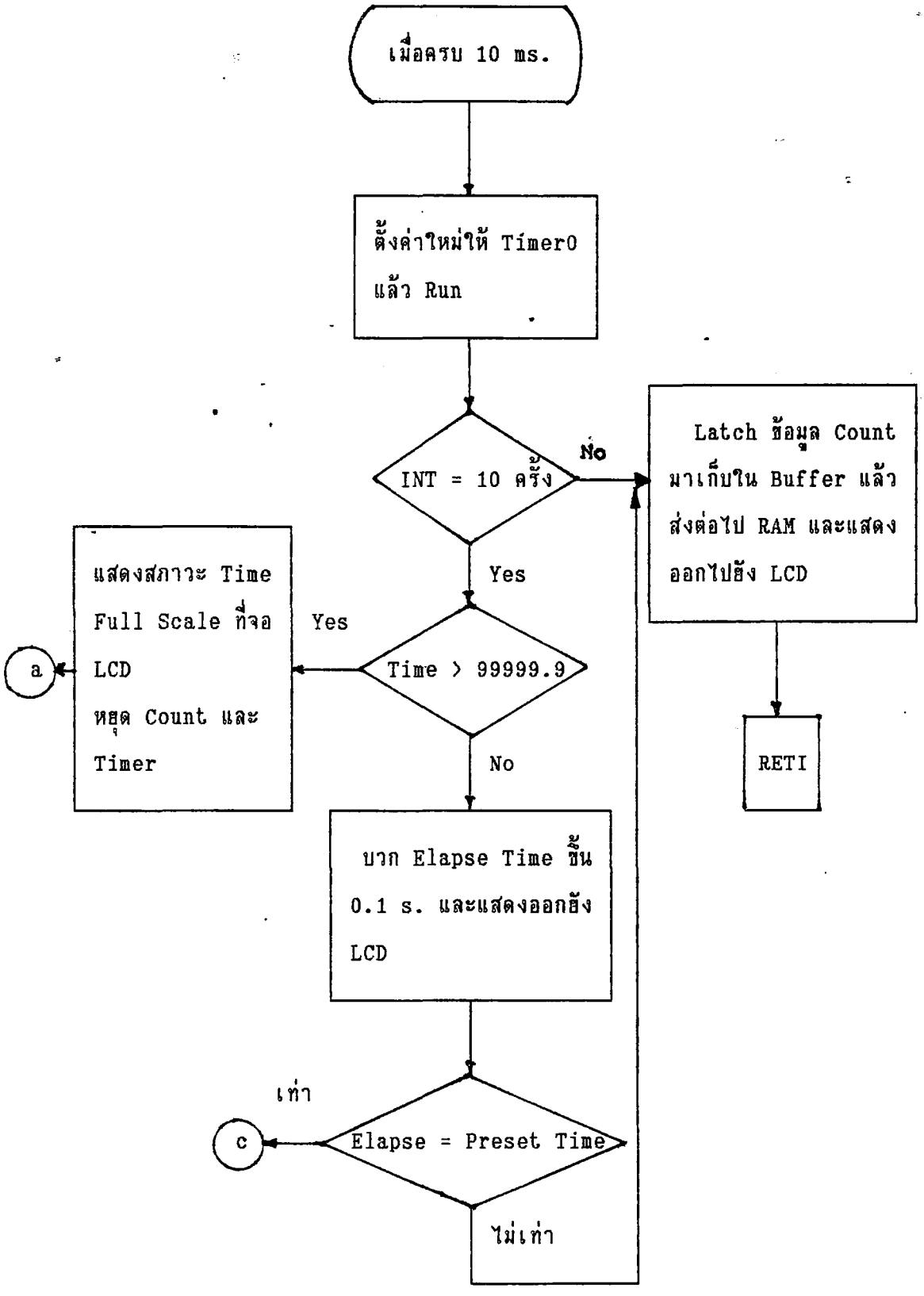



Yes
a
Enable $\overline{INT1}$
เช็คคีย์ Collect/Stop
แสดงความพร้อมที่จอ LCD
พร้อมทั้งค่า Preset, Elapse
และ Count
No
รอรับคีย์ 1-5
Yes
Disable $\overline{INT1}$
ไม่รับคีย์ Collect/Stop
ทำงานตามฟังก์ชันคีย์ 1-5

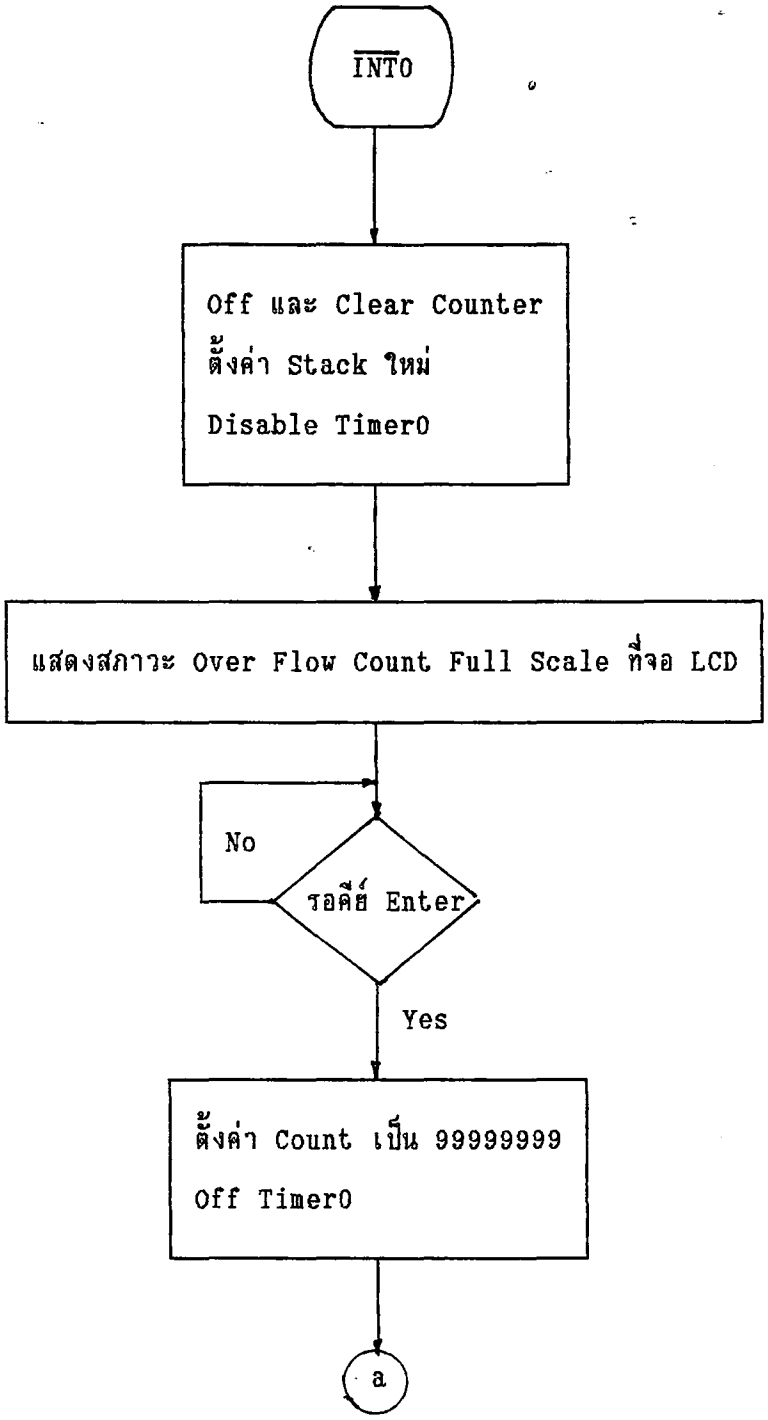
การรับ Interrupt $\overline{INT1}$ จากคีย์ Collect/Stop



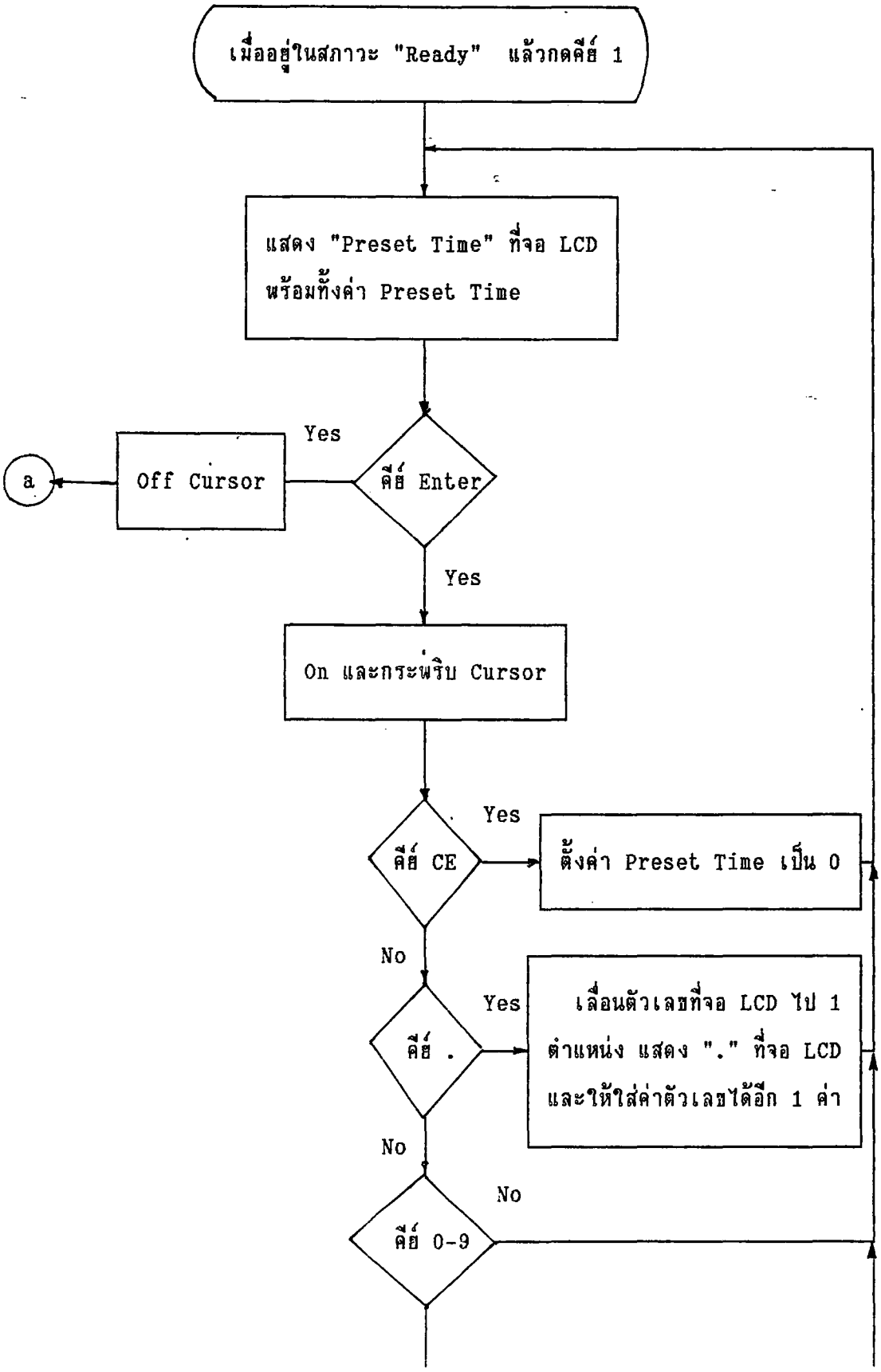
การรับ Interrupt จาก Timer0

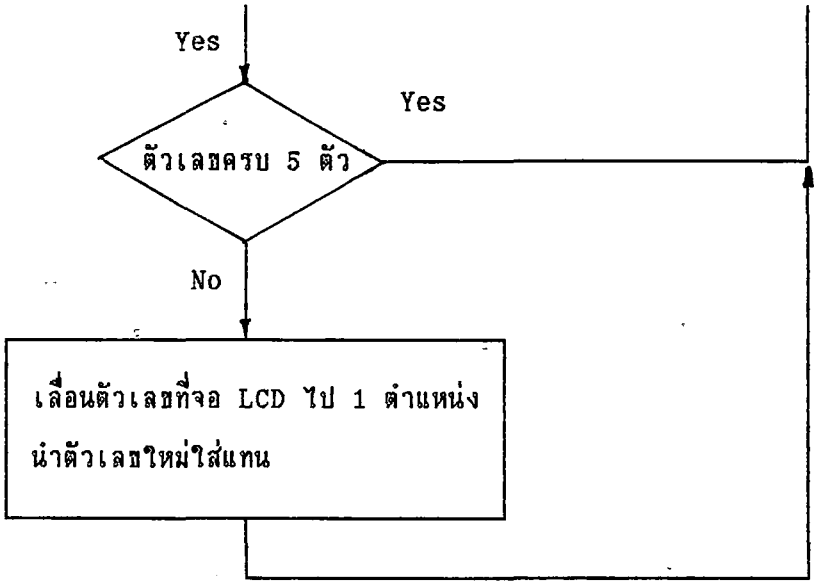


การรับ Interrupt จาก $\overline{\text{INT0}}$ เพื่อแสดงสภาวะ Over Flow Count Full Scale

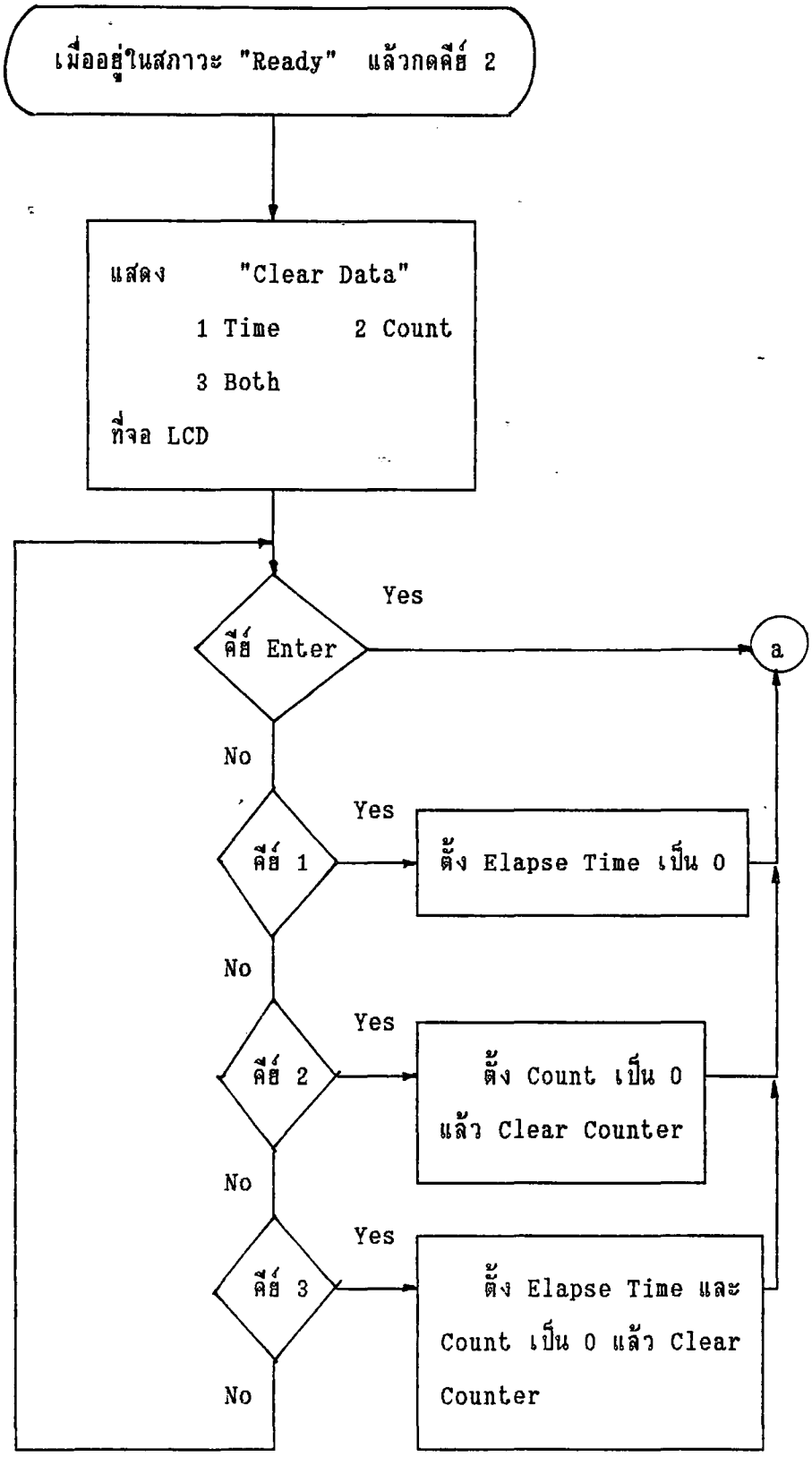


ฟังก์ชัน 1 Preset Time

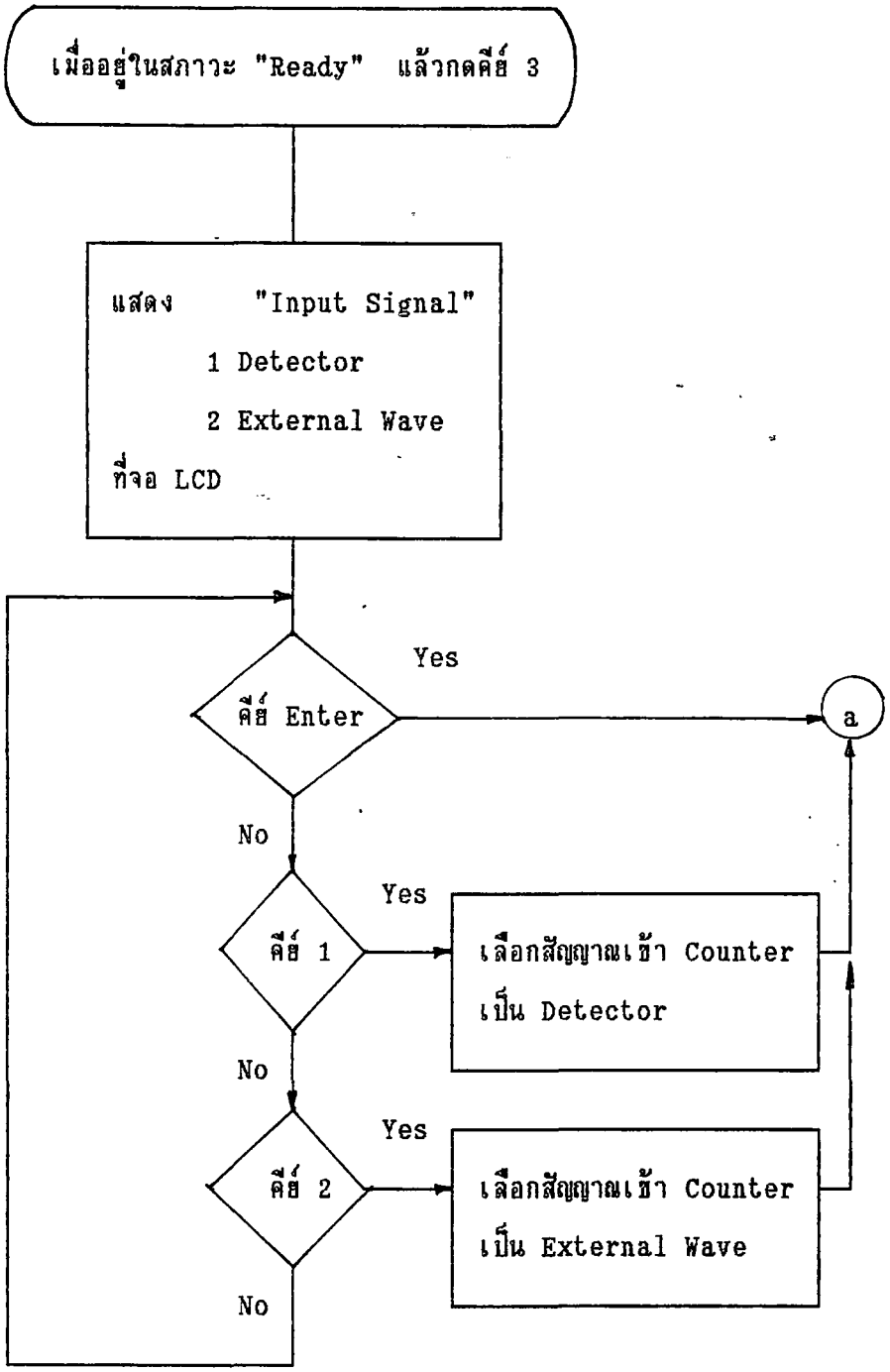



Yes
Yes
ตัวเลขครบ 5 ตัว
No
เลื่อนตัวเลขที่จอ LCD ไป 1 ตำแหน่ง
นำตัวเลขใหม่ใส่แทน

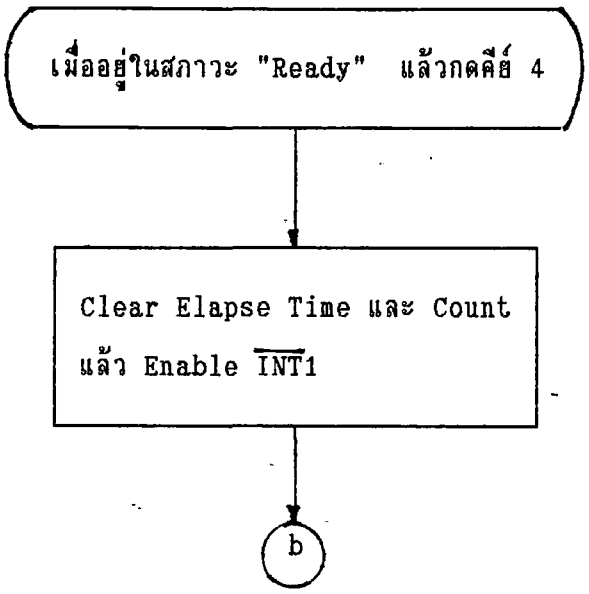
ฟังก์ชัน 2 Clear Data



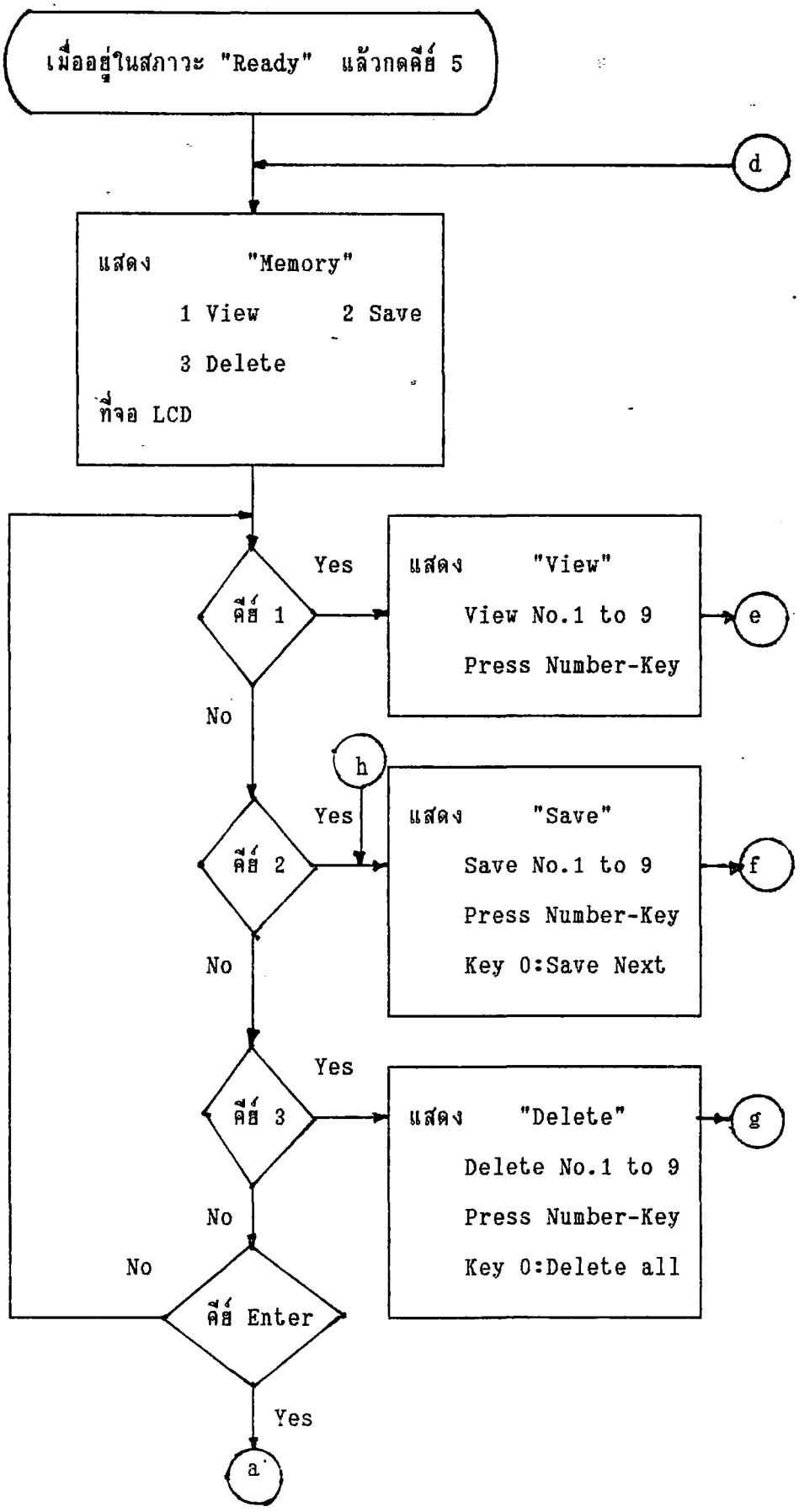
ฟังก์ชัน 3 Input Signal

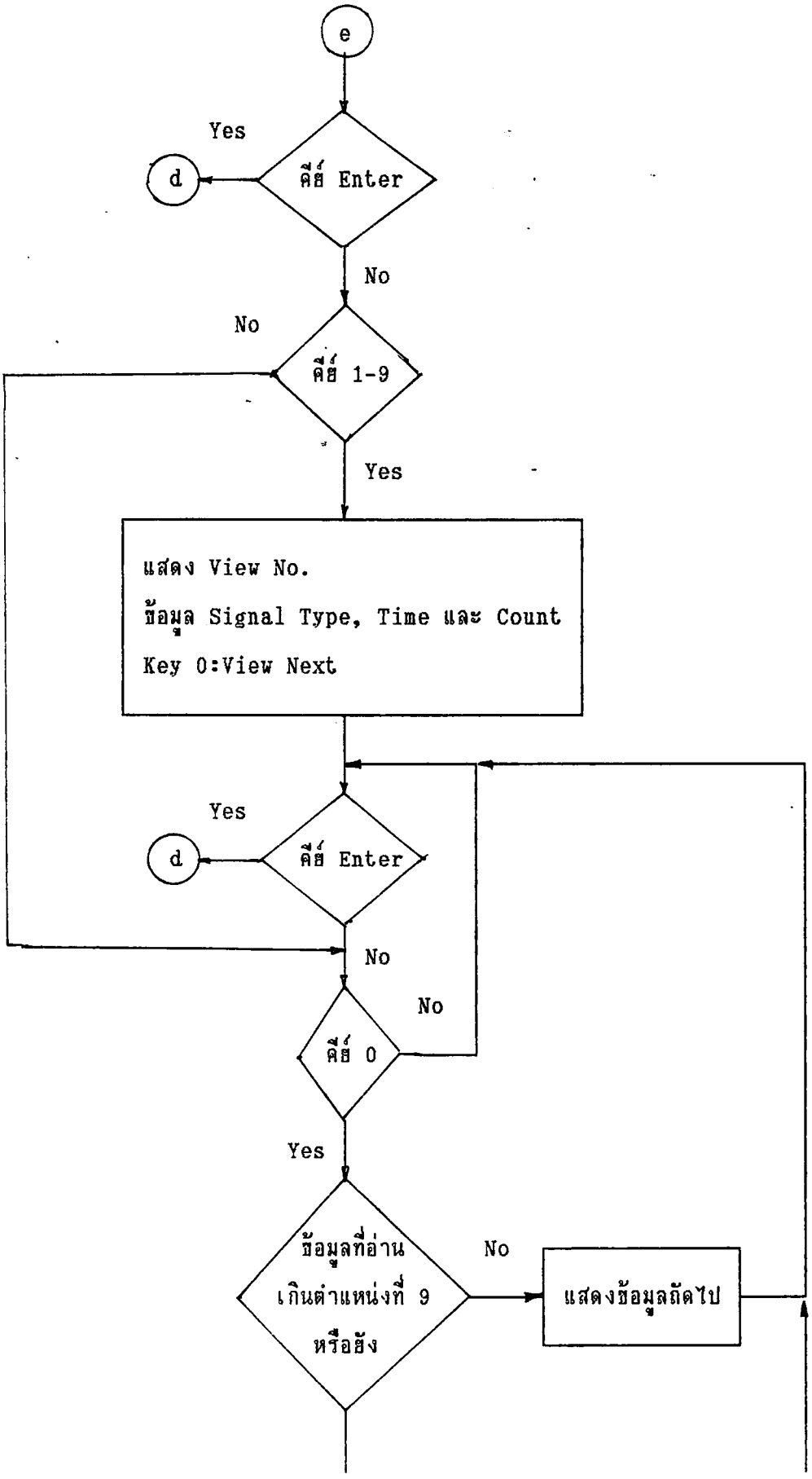


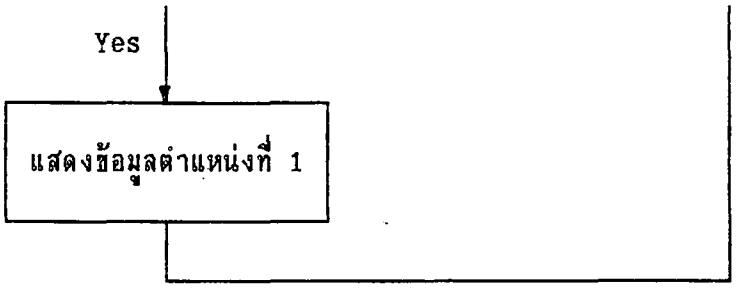
ฟังก์ชัน 4 Auto Run

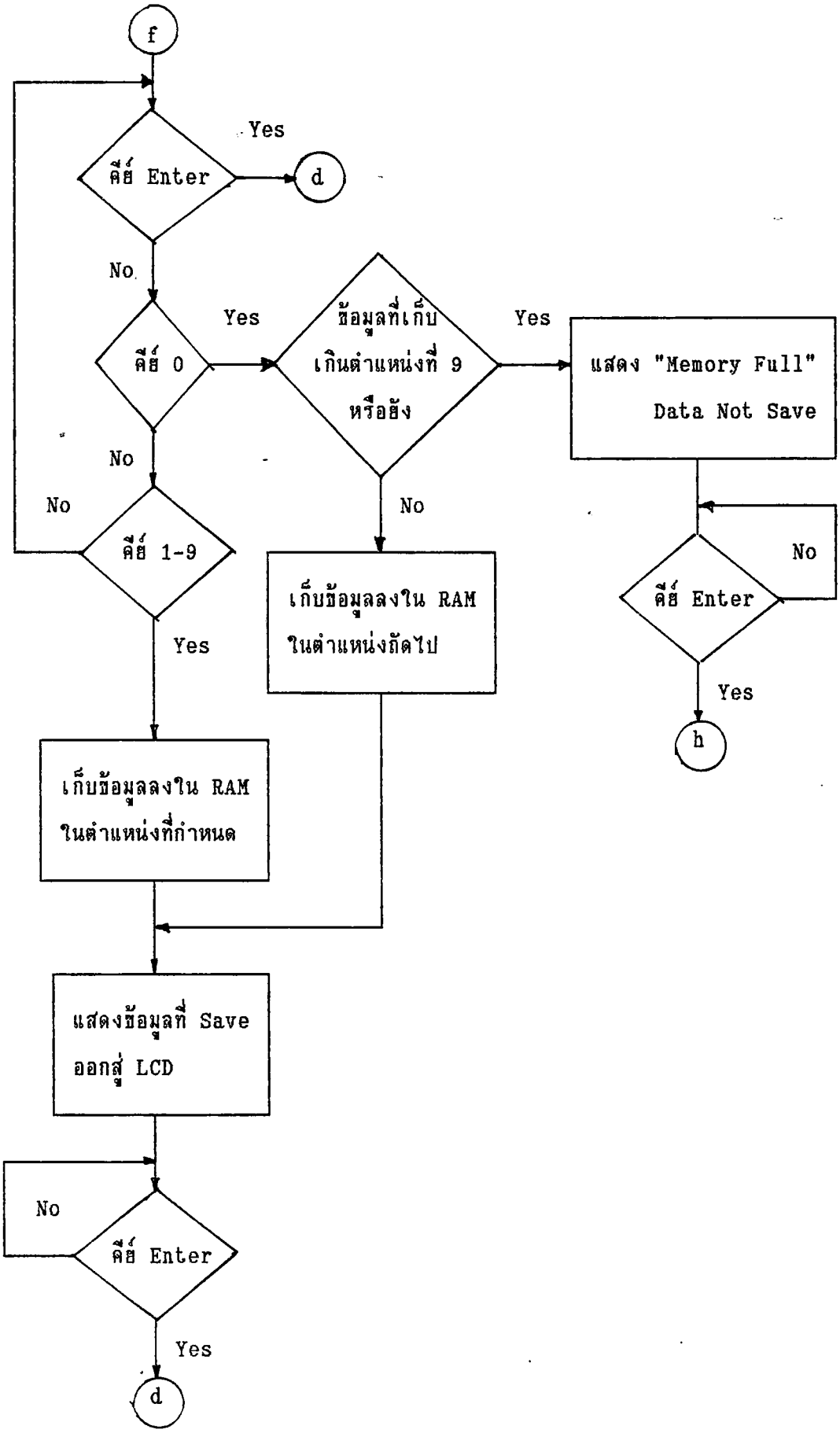


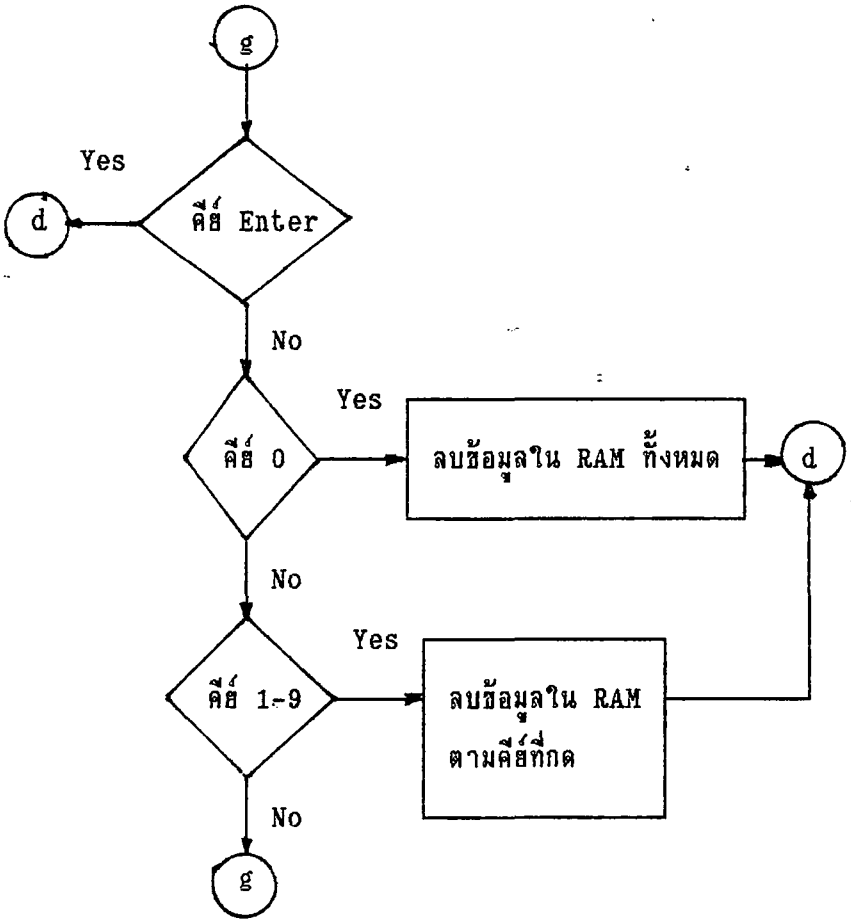
ฟังก์ชัน 5 Memory




e
Yes
d
คีย์ Enter
No
No
คีย์ 1-9
Yes
แสดง View No.
ข้อมูล Signal Type, Time และ Count
Key 0:View Next
Yes
d
คีย์ Enter
No
No
คีย์ 0
Yes
ข้อมูลที่อ่าน
เกินตำแหน่งที่ 9
หรือยัง
No
แสดงข้อมูลถัดไป


Yes
แสดงข้อมูลตำแหน่งที่ 1


f
คีย์ Enter
Yes
d
No
คีย์ 0
Yes
ข้อมูลที่เก็บ
เกินตำแหน่งที่ 9
หรือยัง
Yes
แสดง "Memory Full"
Data Not Save
No
No
คีย์ 1-9
No
เก็บข้อมูลลงใน RAM
ในตำแหน่งถัดไป
No
คีย์ Enter
Yes
h
Yes
เก็บข้อมูลลงใน RAM
ในตำแหน่งที่กำหนด
แสดงข้อมูลที่ Save
ออกสู่ LCD
No
คีย์ Enter
Yes
d


g
Yes
d
คีย์ Enter
No
คีย์ 0
Yes
ลบข้อมูลใน RAM ทั้งหมด
d
No
คีย์ 1-9
Yes
ลบข้อมูลใน RAM
ตามคีย์ที่กด
No
g

ภาคผนวก ค.

```
;*************************************************************************
;*       FILENAME COUNT.ASM                                              *
;*       DATE     22/11/96                                               *
;*       HARDWARE PORT0X :DATA BUS FOR LCD AND LATCH                     *
;*                PORT1X :KEYBOARD                                       *
;*                PORT2X :CONTROL     (LCD)      (2 TO 1)     (2 TO 4)   *
;*                       :PORT     2.0 2.1 2.2   2.3 2.4    2.5 2.6 2.7 *
;*                       :SIGNAL   RS  R/W EN     G/  S     EN/  A   B  *
;*                PORT3X :PORT     3.2(INT0/)   3.3(INT1/)       3.4     *
;*                       :SIGNAL   OVER FLOW   COLLECT/STOP   CLR COUNTER *
;*************************************************************************

;DEFINE BIT ADDRESS FOR CHECKING
        STARTB   EQU  00H
        DECIMAL  EQU  01H
        LCD0     EQU  02H
        SIGNALT  EQU  03H          ;UPDATE SIGANL TYPE
        NO1      EQU  04H          ;BIT ADRESS FOR SVAE SIGNAL TYPE DATA NO.1-9
        NO2      EQU  05H
        NO3      EQU  06H
        NO4      EQU  07H
        NO5      EQU  08H
        NO6      EQU  09H
        NO7      EQU  0AH
        NO8      EQU  0BH
        NO9      EQU  0CH

;DEFINE BYTE ADDRESS FOR SVAE DATA
;CO: COUNT DATA   RT: REAL TIME    PST: PRESET TIME   ADRL: ADDRESS LINE
        CO0      EQU  10H
        CO1      EQU  11H
        CO2      EQU  12H
        CO3      EQU  13H
        RT0      EQU  14H
        RT1      EQU  15H
        RT2      EQU  16H
```

```
        RT3     EQU  17H
        PST1    EQU  18H
        PST2    EQU  19H
        PST3    EQU  1AH
        ADRL    EQU  1BH
        ADRL2   EQU  1CH
        ADRL3   EQU  1DH
        ADRL4   EQU  1EH
        SKEY    EQU  1FH        ;SAVE KEY
        DSP     EQU  63H        ;DATA SAVE POINTER
        STACK   EQU  64H        ;STACK POINTER AT INTERNAL RAM THIS ADDRESS
        BYTEAD  EQU  23H        ;BYTE ADDRESS POINTER
        DATAS   EQU  1DH        ;DATA SVAE IN RAM BEGIN AT THIS ADDRESS + 07

;*************************** START PROGRAM ************************************

        ORG     0000H
        AJMP    BEGIN
        ORG     0003H
        AJMP    OVERFLOW        ;INT0
        ORG     000BH
        AJMP    TIME0           ;T0
        ORG     0013H           ;INT1
        ACALL   WAITIE1         ;CHECK KEY COLLECT/STOP
        JB      STARTB,STOPRETI ;IF BIT SET JUMP STOPRETI
        SETB    STARTB          ;BIT CHECK FOR SOFTWARE
        MOV     SP,#STACK       ;DEFINE STACK AGAIN
        MOV     DPTR,#COLLECT
        PUSH    DPL             ;SAVE DPTR IN STACK
        PUSH    DPH
        RETI
STOPRETI:SETB   P2.3            ;OFF COUNTER
        CLR     ET0
        SETB    P2.5
        CLR     P2.5            ;LATCH
        CLR     EX0             ;DISABLE INT0
        ACALL   DATA_IN
        AJMP    PRERETI
```

```
BEGIN:    MOV     SP,#STACK       ;STACK POINTER AT INTERNAL RAM
          ACALL   INITAIL         ;DEFINE INITAIL VALUE FOR SOFTWARE AND HARDWARE
          ACALL   LCDSET          ;FUNCTION TO SET LCD
          ACALL   LCDSADRL        ;SAVE ADRESS LINE OF DDRAM IN RAM
          ACALL   CLRRALL         ;CLEAR RAM ALL (DATA: CO,RT AND PT)
          ACALL   DELALL          ;CLEAR RAM TO SAVE DATA
          ACALL   LCDPSH          ;WRITE PAGE SHOW TO LCD
SHOW:     ACALL   KEYDOWN         ;WAIT FOR ANY KEY
          JZ      SHOW            ;IF NOT KEY PRESSED PROGRAM IN THE LOOP
          ACALL   LCDPRD          ;DISPLAY LCD TO READY MODE
          ACALL   KEYUP           ;WAIT FOR KEY UP
PUPSCAN:  CLR     STARTB          ;START COLLECT FROM EXTERNAL INTERRUPT 1
          SETB    EX1             ;ENABLE COLLECT
UPSCAN:   ACALL   LCDPRD          ;DISPLAY LCD TO READY MODE
SCAN:     ACALL   SCANK           ;WAIT FOR ANY KEY
          CLR     EX1             ;DISABLE COLLECT
NEXT1:    CJNE    A,#01H,NEXT2    ;CHECK KEY PRESET
          AJMP    PRESET
NEXT2:    CJNE    A,#02H,NEXT3    ;CHECK KEY CLEAR DATA
          AJMP    CLRDATA
NEXT3:    CJNE    A,#03H,NEXT4    ;CHECK KEY INPUT SIGNAL
          AJMP    SIGNALS
NEXT4:    CJNE    A,#04H,NEXT5    ;CHECK KEY AUTO RUN
          AJMP    AUTORUN
NEXT5:    CJNE    A,#05H,OTHERS   ;CHECK KEY INPUT SIGNAL
          AJMP    MEMORY
OTHERS:   SETB    EX1             ;OTHER KEY WILL ENABLE COLLECT
          SJMP    SCAN            ;AND GO TO SCAN LOOP
COLLECT:  ACALL   CHKPST0         ;CHECK PRESET TIME AS ZERO
          JZ      PUPSCAN         ;IF PRESET TIME AS ZERO JUMP TO PUPSCAN
          ACALL   LCDPCO          ;DISPLAY "COUNTING" TO LCD
          CLR     IE0             ;CLR INT0 FLAG
          SETB    EX0             ;PROMT FOR INT0
COUNT_ON: ACALL   RUNTIME         ;START TIMER 0 GENERATE TIME 0.01 SECOND
          CLR     P2.3            ;ON COUNTER
COUNT_IN: SJMP    COUNT_IN        ;WAIT FOR TIMER INTERRUPT
SIGNALS:  ACALL   LCDPIS          ;DISPLAY PAGE SELECT INPUT SIGNAL TO LCD
          ACALL   KEYUP           ;WAIT FOR KEY UP
```

```
SIGNALSL:ACALL   SCANK            ;SCAN KEY
         CJNE    A,#40H,SIGNALSD ;CHECK KEY ENTER
         AJMP    GOBACK          ;EXIT
SIGNALSD:CJNE    A,#01H,SIGNALSN ;CHECK KEY 1.DETECTTOR
         CLR     P2.4            ;CHOOSE INPUT SIGNAL FROM DETECTOR (HARDWARE)
         CLR     SIGNALT         ;BIT CHECK FOR SOFTWARE
         AJMP    GOBACK          ;FINISH SELECTED
SIGNALSN:CJNE    A,#02H,SIGNALSL ;CHECK KEY 2.EXTERNAL WAVE
         SETB    P2.4            ;CHOOSE INPUT SIGNAL FROM EXTERNAL WAVE
         SETB    SIGNALT         ;BIT CHECK FOE SOFTWARE
         AJMP    GOBACK          ;FINISH SELECTED
OVERFLOW:SETB    P2.3            ;OFF COUNTER
         SETB    P3.4            ;HARDWARE CLEAR COUNTER
         CLR     P3.4            ;COUNTER RADY TO COUNT
         SETB    P2.5            ;OFF BUS
         CLR     ET0             ;DISABLE TIMER 0
         CLR     EX0             ;DISABLE INT0
         MOV     SP,#STACK       ;SET STACK POINTER AGAIN
         ACALL   LCDPO           ;SHOW PAGE OVERFLOW COUNT FULL SCALE
WAITO:   ACALL   SCANK
         CJNE    A,#40H,WAITO    ;WAIT FOR ENTER KEY
         ACALL   KEYUP
SETCO:   MOV     R0,#CO3         ;SET COUNT AS 99999999
         MOV     R2,#04H
SETCOL:  MOV     @R0,#99H
         DEC     R0
         DJNZ    R2,SETCOL
PRERETI: CLR     TR0             ;TIMER0 NOT RUN
         MOV     SP,#STACK       ;SET STACK POINTER AGAIN
         SETB    P2.5            ;DATA BUS CONTACT TO LCD
         MOV     DPTR,#PUPSCAN   ;DEFINE NEW VALUE FOR PC
         PUSH    DPL
         PUSH    DPH
         RETI
TIME0:   ACALL   RUNTIME         ;START TIMER 0 TO GENERATE 0.01 SEC
         MOV     A,RT0           ;COPY CONTENT IN RT0 TO REGISTER A
         ADD     A,#10H          ;REAL TIME UPDATE 0.01 SEC
         MOV     RT0,A           ;SAVE CONTENT FROM A TO RT0
```

```
        ACALL   CHKRT99          ;CHECK REAL TIME TO 99999.9
        CJNE    A,#0A0H,TIMEN10 ;CHECK TIME AS 0.1 SEC
        MOV     RT0,#00H         ;CLEAR RT0 TO ZERO
UPTIME: CLR     CY               ;CLEAR CARY FLAG FOR LOOP
        MOV     R1,#03H          ;NUMBER OF LOOP
        MOV     A,#01H           ;SET A TO ONE
        MOV     R0,#RT1
UPLOOP: ADDC    A,@R0            ;SUBROUTINE FOR INCREMENT TIME
        DA      A
        MOV     @R0,A
        CLR     A
        INC     R0
        DJNZ    R1,UPLOOP
        SETB    P2.5
        ACALL   LCDRDTR          ;SEND DATA REAL TIME TO LCD
        MOV     R0,#PST3         ;SUBROUTINE TO CHECK PST EQUAL RT
        MOV     A,@R0
        CJNE    A,RT3,TIMEN10
        DEC     R0
        MOV     A,@R0
        CJNE    A,RT2,TIMEN10
        DEC     R0
        MOV     A,@R0
        CJNE    A,RT1,TIMEN10
OVERTIME:CLR    P2.5             ;OFF BUS
        AJMP    STOPRETI         ;REAL TIME EQU PRESET TIME JUMP STOP
RUNTIME: CLR    TR0              ;CLR TIMER RUN 0
        MOV     TH0,#0DCH        ;PRESET TIMER0 DC09H TO MAKE 10 mS.
        MOV     TL0,#09H
        SETB    ET0              ;ENABLE TIMMER INTERRUPT 0
        SETB    TR0              ;RUN TIMER 0 ,TOTAL 9 CYCLE
        RET
TIMEN10: SETB   P2.5             ;LATCH LOW TO HIGH FROM NOT GATE
        CLR     P2.5
        ACALL   DATA_IN          ;MOVING DATA FROM LCATCH TO RAM
        SETB    P2.5             ;BUS CONNECTED TO LCD
        ACALL   LCDRDTC          ;SEND DATA COUNT TO LCD
        RETI
```

```
CLRDATA: ACALL   LCDPC          ;DIPLAY PAGE CLEAR DATA TO LCD
         ACALL   KEYUP          ;WAIT FOR KEY UP
SCANC:   ACALL   SCANK          ;SCAN KEY
KEY1C:   CJNE    A,#01H,KEY2C   ;CHECK KEY CLEAR REAL TIME
         ACALL   CLRRT
         SJMP    GOBACK
KEY2C:   CJNE    A,#02H,KEY3C   ;CHECK KEY CLEAR COUNT
         ACALL   CLRCO
         SJMP    PGOBACK
KEY3C:   CJNE    A,#03H,KEY4C   ;CHECK KEY REAL TIME AND COUNT
         ACALL   CLRRTCO
PGOBACK: SETB    P3.4           ;HARDWARE CLEAR COUNTER
         CLR     P3.4
         SJMP    GOBACK
KEY4C:   CJNE    A,#40H,SCANC   ;CHECK KEY ENTER
GOBACK:  ACALL   KEYUP          ;WAIT FOR KEY UP
GOBACKD: CLR     IE1            ;CLEAR INT1 FLAG
         SETB    EX1            ;ENABLE COLLECT
         AJMP    UPSCAN         ;JUMP TO LABLE UPSCAN
AUTORUN: ACALL   CLRRTCO        ;CLEAR RAEL TIME AND COUNT
         SETB    STARTB         ;SET BIT TO START COLLECT
         SETB    P3.4           ;HARDWARE CLEAR COUNTER
         CLR     P3.4
         ACALL   KEYUP          ;WAIT FOR KEY UP
         CLR     IE1            ;CLEAR INT1 FLAG
         SETB    EX1            ;ENABLE COLLECT
         AJMP    COLLECT        ;JUMP TO LABLE COLLECT
PRESET:  ACALL   LCDPPS         ;DISPLAY PAGE PRESET TIME TO LCD
PUPSCANP:MOV     R4,#00H        ;CLEAR R4 FOR LOOP COUNTING
         CLR     DECIMAL        ;CLEAR BIT DECIMAL FOR SOFTWARE CHECK
UPSCANP: ACALL   KEYUP          ;WAIT FOR KEY UP
SCANP:   ACALL   SCANK          ;SCAN KEY
         CJNE    R4,#00H,ENTER  ;IF R4 EQUAL ZERO GIVE ON CURSOR
         PUSH    ACC            ;SAVE KEY
         MOV     A,#00001111B   ;ON AND BLINK CURSOR
         ACALL   LCDWI
         POP     ACC            ;KEY OCCUR AGAIN
ENTER:   CJNE    A,#40H,KEYCLR  ;CHECK ENTER KEY
```

```
        MOV     A,#00001100B    ;CURSOR OFF
        ACALL   LCDWI
        AJMP    GOBACK
KEYCLR: CJNE    A,#30H,CLROLD   ;CHECK CLEAR ENTRY KEY
        ACALL   CLRPST          ;CLEAR PRESET TIME
        CLR     DECIMAL         ;CLEAR BIT DECIMAL
        ACALL   PSTWLCD         ;WRITE PRESET TIME (0) TO LCD
        SJMP    PUPSCANP
CLROLD: CJNE    R4,#00H,CHKKEYD ;IF R4 EQUL ZERO
        ACALL   CLRPST          ;CLEAR PRESET TIME
CHKKEYD: JB     DECIMAL,CHECK5  ;CHECK KEY . REPEAT ?
KEY.:   CJNE    A,#20H,CHECK5   ;CHECK KEY .
        AJMP    PSTKEYD
CHECK5: CJNE    R4,#05H,KEY0    ;CHECK NUMBER OF CHARACTER
        SJMP    SCANP           ;IF R4 EQUAL 5 WAIT FOR KEY.,ENTER,CLR ENTRY
KEY0:   CJNE    A,#10H,KEY1TO9  ;CHECK KEY 0
        MOV     A,#00H          ;GIVE A AS ZERO
        CJNE    R4,#00H,NUMKEY  ;CHECK KEY 0 BEFORE KEY 1 TO 9
        SJMP    DNUMKEY
KEY1TO9: CLR    CY              ;CHECK KEY 1-9
        SUBB    A,#0AH
        JNC     SCANP
        ADD     A,#0AH
        AJMP    NUMKEY
NUMKEY: INC     R4              ;INCREMENT ONE CHARACTER
DNUMKEY: JB     DECIMAL,PSTSD   ;IF BIT DECIMAL SET JUMP TO PSTSD
        ACALL   PSTSHFD         ;SHIFT PRESET TIME ONE DIGIT
PSTS:   SWAP    A
        MOV     @R1,A           ;SAVE NUMBER IN PST1 UPPER NIBBLE
PSTSC:  ACALL   PSTWLCD         ;WRITE PRESSET TIME TO LCD
        AJMP    UPSCANP         ;COME BACK TO LOOP
PSTSD:  MOV     R1,#PST1
        XCHD    A,@R1           ;SAVE NUMBER IN PST1 LOWER NIBBLE
        AJMP    PSTSC

;************************ LCD SUBROUTINE ***************************

;LCD WRITE INSTRUCTION
```

```
LCDWI:   ACALL    LCDRB
         CLR      P2.0            ;RS = 0
         CLR      P2.1            ;R/W = 0
LCDEN:   MOV      P0,A
         CLR      P2.2            ;ENABLE LOW TO HIGH
         SETB     P2.2            ;EN = HIGH
         CLR      P2.2            ;EN = LOW
         RET

;LCD WRITE DATA
LCDWD:   ACALL    LCDRB
         SETB     P2.0            ;RS = 1
         CLR      P2.1            ;R/W = 0
         SJMP     LCDEN

;LCD READ BUSY
LCDRB:   CLR      P2.0            ;RS = 0
         SETB     P2.1            ;R/W = 1
         MOV      P0,#0FFH        ;CONFIG P0 AS INPUT
LCDWAIT: CLR      P2.2            ;EN = LOW
         SETB     P2.2            ;EN = HIGH
         JB       P0.7,LCDWAIT    ;BF = 1 BUSY
         CLR      P2.2
         RET

;LCD WRITE DATA LINE ,USE DPTR POINT DATA ROM
LCDWDL:  MOV      R3,#10H         ;LOOP FOR DATA 16 CHARACTER
LCDWCHA: CLR      A               ;PRESET REGISTER A
         MOVC     A,@A+DPTR       ;GET DATA 1 BYTE TO REGISTER A
         ACALL    LCDWD
         INC      DPTR
         DJNZ     R3,LCDWCHA
         RET

;LCD WRITE DATA FROM RAM ,USE @R1 TO POINT DATA
LCDWDR:  MOV      R3,#04H         ;LOOP FOR DATA 8 CHARACTER 4 BYTE
         CLR      LCD0
LCDWDRC: MOV      A,@R1           ;GET DATA 1 BYTE TO REGISTER A
```

```
          ANL     A,#0F0H
          SWAP    A
          ACALL   LCDC0
          ADD     A,#30H
          ACALL   LCDWD
          MOV     A,@R1
          ANL     A,#0FH
          CJNE    R3,#01H,LCDPASS ;CHECK 0 FINAL COLUMN
          SETB    LCD0
LCDPASS:  ACALL   LCDC0
          ADD     A,#30H
          ACALL   LCDWD
          DEC     R1
          DJNZ    R3,LCDWDRC
          RET
LCDWDRT:  MOV     R3,#03H         ;LOOP FOR DATA 6 CHARACTER 3 BYTE
          CLR     LCD0
LCDWDRTL:MOV      A,@R1           ;GET DATA 1 BYTE TO REGISTER A
          ANL     A,#0F0H
          SWAP    A
          CJNE    R3,#01H,LCDWDRTN
          SETB    LCD0
LCDWDRTN:ACALL    LCDC0
          ADD     A,#30H
          ACALL   LCDWD
          CJNE    R3,#01H,LCDLNB
          JNB     DECIMAL,LCDWDRTE
          MOV     A,#2EH
          ACALL   LCDWD
LCDLNB:   MOV     A,@R1
          ANL     A,#0FH
          ACALL   LCDC0
          ADD     A,#30H
          ACALL   LCDWD
LCDWDRTE:DEC      R1
          DJNZ    R3,LCDWDRTL
          RET
```

```
;LCD CUT ZERO
LCDC0:   JB       LCD0,LCDN0       ;CHECK NUMBER
         JNZ      LCDN0            ;CHECK A
         MOV      A,#0E0H
         RET
LCDN0:   SETB     LCD0             ;NUMBER OCCUR
         RET

;LCD WRITE PAGE ,USE @R0 POINT DDRAM ADDRESS ,R2 LOOP FOR 4 LINE
LCDWP:   MOV      R2,#04H
LCDWL1:  MOV      R0,#ADRL         ;IF WRITE ONE AND TOP LINE CALL LCDWL1
LCDWPL:  MOV      A,@R0            ;SET ADDRESS LINE
         ACALL    LCDWI
         ACALL    LCDWDL           ;WRITE DATA LINE
         INC      R0
         DJNZ     R2,LCDWPL
         RET

;LCD READ DATA PRESET TIME,REAL TIME,COUNT
LCDRDTP: MOV      R0,#ADRL2
         MOV      R1,#PST3
LCDRDTPN:SETB     CY
         ACALL    LCDRDT
         SJMP     LCDWDRT
LCDRDTR: MOV      R0,#ADRL3
         MOV      R1,#RT3
         SJMP     LCDRDTPN
LCDRDTC: MOV      R0,#ADRL4
         MOV      R1,#CO3
         CLR      CY
         ACALL    LCDRDT
         SJMP     LCDWDR

;LCD READ DATA   RAM ,USE @R0 POINT DDRAM ADDRESS
LCDRDT:  MOV      A,@R0            ;SET ADDRESS LINE
         ADDC     A,#08H           ;COLUMN 8 FROM BACK
         ACALL    LCDWI
         SETB     DECIMAL
```

```
          RET

;LCD READ DATA ALL
;LCD PAGE SHOW,READY,PRESET,CLEAR,OVERFLOW,COUNT,TEST
LCDPSH:   MOV     DPTR,#LCDP0     ;DPTR POINT DATA TO DISPLAY
          AJMP    LCDWP
LCDPRD:   MOV     DPTR,#LCDP1     ;DPTR POINT DATA TO DISPLAY READY PAGE
          ACALL   LCDWP
          ACALL   STYPE
LCDRDTA:  ACALL   LCDRDTP
          ACALL   LCDRDTR
          AJMP    LCDRDTC
LCDPPS:   MOV     DPTR,#LCDP2     ;DPTR POINT DATA TO DISPLAY PRESET PAGE
          ACALL   LCDWP
          MOV     A,#0D9H
          MOV     R1,#PST3
          SETB    DECIMAL
          ACALL   LCDWI
          AJMP    LCDWDRT
LCDPC:    MOV     DPTR,#LCDP3     ;DPTR POINT DATA TO DISPLAY CLEAR PAGE
          AJMP    LCDWP
LCDPO:    MOV     DPTR,#LCDP4     ;DPTR POINT DATA TO DISPLAY OVER FLOW PAGE
          AJMP    LCDWP
LCDPCO:   MOV     DPTR,#LCDLCO
LCDPCOL:  MOV     R2,#01H
          ACALL   LCDWL1
          AJMP    STYPE
LCDPIS:   MOV     DPTR,#LCDPSI
          AJMP    LCDWP
LCDPTF:   MOV     DPTR,#LCDPFT    ;DPTR POINT DATA TO DISPLAY TIME FULL
          AJMP    LCDWP
LCDSADRL:MOV      R0,#ADRL
          MOV     @R0,#80H
          INC     R0
          MOV     @R0,#0C0H
          INC     R0
          MOV     @R0,#90H
          INC     R0
```

```
        MOV     @R0,#0D0H
        RET
LCDSET: MOV     A,#00111000B    ;FUNCTION SET 8 BIT,1/16 DUTY,5*7 DOTS
        ACALL   LCDWI
        MOV     A,#00001100B    ;DISPLAY ON,CURSOR OFF AND NO BLINK
        ACALL   LCDWI
        MOV     A,#00000001B    ;CLEAR DISPLAY
        ACALL   LCDWI
        MOV     A,#00000110B    ;ENTRY MODE SET,INCREMENT CURSOR AND DDRAM
        ACALL   LCDWI
        RET

;************************** END OF LCD SUBROUTINE **********************

;RAM CLEAR 4 BYTES
RAMCLR: MOV     R2,#04H
RAMCLRL:MOV     @R0,#00H
        DEC     R0
        DJNZ    R2,RAMCLRL
        RET
CLRPST: MOV     R0,#PST3
        MOV     R2,#03H
        SJMP    RAMCLRL
CLRRT:  MOV     R0,#RT3
        SJMP    RAMCLR
CLRCO:  MOV     R0,#CO3
        SJMP    RAMCLR
CLRRTCO:MOV     R0,#RT3
        MOV     R2,#08H
        SJMP    RAMCLRL
CLRRALL:MOV     R0,#PST3
        MOV     R2,#0CH
        SJMP    RAMCLRL

;CHECK PRESET TIME EQUAL ZERO
CHKPST0:MOV     R0,#PST1
        MOV     R2,#03H
CHKPSTOL:CJNE   @R0,#00H,OUTLCHK
```

```
          INC       R0
          DJNZ      R2,CHKPST0L
          DEC       R0
OUTLCHK:  MOV       A,@R0
          RET

;CHECK PRESET TIME EQUAL NINENINE
CHKRT99:  MOV       R0,#RT3
          MOV       R2,#03H
CHKRT99L:CJNE       @R0,#99H,OUTLCHK9
          DEC       R0
          DJNZ      R2,CHKRT99L
          CJNE      @R0,#10H,OUTLCHK9
          AJMP      TIMEFULL
OUTLCHK9:RET

;SOFTWARE DELAY 20 mS.
DELAY:    MOV       07H,#24H        ;2 CYCLE AND CALL 2 CYCLE
DELAYL1:  PUSH      07H             ;2 CYCLE
          MOV       07H,#0FFH       ;2 CYCLE
DELAYL0:  DJNZ      07H,DELAYL0     ;2 CYCLE
          POP       07H             ;2 CYCLE
          DJNZ      07H,DELAYL1     ;2 CYCLE
          RET                       ;2 CYCLE
;WAIT FOR INTERRUPT ONE  (COLLECT/STOP)
WAITIE1:  CLR       EX1
          CLR       ET0
          CLR       TR0
IE1UP:    SETB      P3.3
          JNB       P3.3,IE1UP
          ACALL     DELAY
          SETB      P3.3
          JNB       P3.3,IE1UP
          SETB      EX1
          RET

;WAIT FOR KEY UP
KEYUP:    ACALL     KEYDOWN
```

```
        JNZ     KEYUP
        MOV     07,#59H
        ACALL   DELAYL1         ;50mS.
        ACALL   KEYDOWN
        JNZ     KEYUP
        RET

;WAIT FOR KEY DOWN
KEYDOWN: MOV    P1,#0FFH
        MOV     A,P1
        CPL     A
        RET

;MOVE DATA FROM LATCH TO RAM (C00-C03)
DATA_IN: MOV    A,#0FFH
        CLR     P2.6            ;P2.6 = A
        CLR     P2.7            ;P2.7 = B
        MOV     P0,A            ;CONFIG PORT 0 AS INPUT
        MOV     C00,P0          ;B A ! DIGIT ! SAVE IN
        SETB    P2.6            ;0 0 !  0-1  !   C00
        MOV     P0,A
        MOV     C01,P0          ;0 1 !  2-3  !   C01
        CLR     P2.6            ;1 0 !  4-5  !   C02
        SETB    P2.7            ;1 1 !  6-7  !   C03
        MOV     P0,A
        MOV     C02,P0
        SETB    P2.6
        MOV     P0,A
        MOV     C03,P0
        RET

;SHIFT PRESET TIME ONE DIGIT
PSTSHFD: PUSH   ACC
        MOV     R0,#PST3
        MOV     R1,#PST2
        MOV     A,@R0
        SWAP    A
        MOV     @R0,A
```

```
        MOV     A,@R1
        SWAP    A
        XCHD    A,@R0
        MOV     @R1,A
        DEC     R0
        DEC     R1
        MOV     A,@R1
        SWAP    A
        XCHD    A,@R0
        MOV     @R1,A
        POP     ACC
        RET

;WRITE PRESET TIME TO LCD FOR TIME UNIT SEC
PSTWLCD: MOV    A,#0D9H
        MOV     R1,#PST3
        ACALL   LCDWI
        JB      DECIMAL,PSTWLCDN
        MOV     A,#20H
        ACALL   LCDWD
        ACALL   LCDWD
PSTWLCDN:ACALL  LCDWDRT
        MOV     A,#00010000B    ;CURSOR SHIFT LEFT
        AJMP    LCDWI

;SET VALUE WHEN PRESS KEY .
PSTKEYD: SETB   DECIMAL
        MOV     R4,#04H
        ACALL   PSTWLCD
        MOV     A,#20H
        ACALL   LCDWD
        MOV     A,#00010000B
        ACALL   LCDWI
        AJMP    UPSCANP

;DISPLAY TYPE INPUT SIGNAL
STYPE:  MOV     A,ADRL
        ACALL   LCDWI
```

```
        JB      SIGNALT,TEW
        MOV     A,#'D'
        AJMP    LCDWD
TEW:    MOV     A,#'E'
        AJMP    LCDWD

;TIME RUN TO 99999.9
TIMEFULL:SETB   P2.3            ;OFF COUNTER
        SETB    P3.4            ;HARDWARE CLEAR
        CLR     P3.4            ;READY TO COUNT
        SETB    P2.5            ;OFF BUS
        CLR     ET0
        CLR     TR0
        ACALL   LCDPTF
WAITTF: ACALL   SCANK
        CJNE    A,#40H,WAITTF
        ACALL   KEYUP
        MOV     RT0,#00H
        AJMP    PRERETI

;SET INITAIL VALUE FOR HARDWARE AND SOFTWARE
INITAIL: CLR    IT1             ;INTERRUPT SIGNAL TRIGGERED FALLING EDGE
        SETB    IT0
        MOV     TMOD,#00100001B ;TIMER1 MODE 2,TIMER0 MODE 1
        MOV     IE,#10000000B   ;ENABLE IE0,TF0 DISABLE TI OR RI,TF1,IE1
        SETB    ET0
        MOV     IP,#00000101B   ;TIMER0 LOW PIORITY
        SETB    P2.5            ;OFF DATA BUS FROM COUNTER
        SETB    P2.3            ;OFF COUNTER
        SETB    P3.4            ;CLEAR COUNT
        CLR     P3.4            ;READY TO COUNT
        CLR     EX1             ;DISABLE COLLECT
        CLR     P2.4            ;SELECT DETECTOR INPUT SIGNAL (HARDWARE)
        CLR     SIGNALT         ;SELECT DETECTOR INPUT SIGNAL (SOFTWARE)
        MOV     DSP,#00H        ;DATA SAVE POINTER NUMBER
        RET

;SCAN KEY
```

```
SCANK:   ACALL    KEYDOWN          ;CHECK KEY PRESSED
         JZ       SCANK            ;IF NOT KEY PRESSED THEN PROGRAM IN LOOP
         MOV      SKEY,A           ;SAVE OLD KEY IN RAM
         ACALL    DELAY            ;SOFTWARE DELAY 20 mS.
         ACALL    KEYDOWN          ;CHECK KEY PRESSED AGAIN
         CJNE     A,SKEY,SCANK     ;IF KEY NOT EQUAL AS NOISE
         RET

;MEMORY
MEMORY:  MOV      DPTR,#LCDPM      ;DPTR POINT DATA TO DISPLAY MEMORY PAGE
         ACALL    LCDWP
         ACALL    KEYUP            ;WAIT FOR KEY UP
SCANM:   ACALL    SCANK            ;WAIT FOR ANY KEY
KEY1M:   CJNE     A,#01H,KEY2M     ;CHECK KEY VIEW
         AJMP     MVIEW
KEY2M:   CJNE     A,#02H,KEY3M     ;CHECK KEY SAVE
         AJMP     MSAVE
KEY3M:   CJNE     A,#03H,KEY4M     ;CHECK KEY DELETE
         AJMP     MDEL
KEY4M:   CJNE     A,#40H,SCANM     ;CHECK KEY EXIT
         AJMP     GOBACK

;MEMORY VIEW
MVIEW:   MOV      DPTR,#LCDPMV
         ACALL    LCDWP
         ACALL    KEYUP
SCANMV:  ACALL    SCANK
         CJNE     A,#40H,MV1TO9
         AJMP     MEMORY
MV1TO9:  CLR      CY
         SUBB     A,#0AH
         JNC      SCANMV
         ADD      A,#0AH
         MOV      R6,A
         MOV      DPTR,#LCDPMVN
         ACALL    LCDWP
USCANMVN:ACALL    GETADT
         ACALL    LCDWDS
```

```
          ACALL     KEYUP
SCANMVN:  ACALL     SCANK
          CJNE      A,#40H,SCANMVN0
          AJMP      MEMORY
SCANMVN0:CJNE       A,#10H,SCANMVN
          CJNE      R6,#09H,MVNONINE
          MOV       R6,#00H
MVNONINE:INC        R6
          AJMP      USCANMVN

;MEMORY SAVE
MSAVE:    MOV       DPTR,#LCDPMS
          ACALL     LCDWP
USCANMS:  ACALL     KEYUP
SCANMS:   ACALL     SCANK
          CJNE      A,#40H,SCANMS0
          AJMP      MEMORY
SCANMS0:  CJNE      A,#10H,MS1TO9
          MOV       A,DSP
          CJNE      A,#09H,MSNONINE
          MOV       DPTR,#LCDPMF
          ACALL     LCDWP
          ACALL     KEYUP
MSNINE:   ACALL     SCANK
          CJNE      A,#40H,MSNINE
          AJMP      MSAVE
MSNONINE:INC        DSP
          MOV       R6,DSP
          AJMP      USCANMSN
MS1TO9:   CLR       CY
          SUBB      A,#0AH
          JNC       SCANMS
          ADD       A,#0AH
          MOV       DSP,A
          MOV       R6,A
USCANMSN:MOV        DPTR,#LCDPMSN
          ACALL     LCDWP
          ACALL     SINRAM
```

```
          ACALL     LCDWDS
          ACALL     KEYUP
SCANMSN:  ACALL     SCANK
          CJNE      A,#40H,SCANMSN
          AJMP      MEMORY

;MEMORY DELETE
MDEL:     MOV       DPTR,#LCDPMD
          ACALL     LCDWP
          ACALL     KEYUP
SCANMD:   ACALL     SCANK
          CJNE      A,#40H,SCANMD0
          AJMP      MEMORY
SCANMD0:  CJNE      A,#10H,MD1TO9
          ACALL     DELALL
          AJMP      MEMORY
MD1TO9:   CLR       CY
          SUBB      A,#0AH
          JNC       SCANMD
          ADD       A,#0AH
          MOV       R6,A
          ACALL     GETADT
          ACALL     DEL7B1B
          AJMP      MEMORY

;SAVE DATA IN RAM
SINRAM:   ACALL     GETADT
          ACALL     SSTINR
          AJMP      SDTINR

;SAVE SIGNAL TYPE IN RAM
SSTINR:   JB        SIGNALT,SETEW
          AJMP      CSTDS1
SETEW:    AJMP      SSTDS1

;GET ADDRESS DATA
GETADT:   MOV       A,R6
          ADD       A,#02H
```

```
          MOVC     A,@A+PC
          SJMP     OVERC
          DB       1DH,24H,2BH,32H,39H,40H,47H,4EH,55H,5CH
OVERC:    MOV      BYTEAD,A
          RET

;SAVE DATA IN RAM
SDTINR:   MOV      R1,BYTEAD
          MOV      R0,#C00
          MOV      R2,#07H
SDTINRL:  CJNE     R0,#RT0,SDTINRN
          INC      R0
SDTINRN:  MOV      A,@R0
          MOV      @R1,A
          INC      R0
          INC      R1
          DJNZ     R2,SDTINRL
          RET

;WRITE DATA SVAE TO LCD
LCDWDS:   ACALL    LCDWDST
          AJMP     LCDWDSN

;WRITE DATA SIGNAL TYPE TO LCD
LCDWDST:  MOV      A,#88H
          ACALL    LCDWI
          MOV      A,R6
          ADD      A,#30H
          ACALL    LCDWD
          MOV      A,#8FH
          ACALL    LCDWI
          ACALL    WSTDS1
          AJMP     LCDWD

;WRITE DATA SVAE NEXT (TIME,COUNT)
LCDWDSN:  MOV      R0,#ADRL2
          MOV      A,BYTEAD
          ADD      A,#06H
```

```
        MOV     R1,A
        SETB    CY
        ACALL   LCDRDT
        ACALL   LCDWDRT
        MOV     R0,#ADRL3
        MOV     A,BYTEAD
        ADD     A,#03H
        MOV     R1,A
        CLR     CY
        ACALL   LCDRDT
        AJMP    LCDWDR

;DELETE 7 BYTE 1 BIT
DEL7B1B: MOV    A,BYTEAD
        MOV     R1,A
        MOV     R2,#07H         ;LOOP COUNTER
        CLR     A               ;SET A AS ZERO
DEL7B1BL:MOV    @R1,A
        INC     R1
        DJNZ    R2,DEL7B1BL
        ACALL   CSTDS1
        DEC     R6
        MOV     DSP,R6
        RET

;DELETE MEMORY ALL
DELALL:  CLR    NO1
        CLR     NO2
        CLR     NO3
        CLR     NO4
        CLR     NO5
        CLR     NO6
        CLR     NO7
        CLR     NO8
        CLR     NO9
DELBYTE: MOV    R1,#24H         ;ADRESS DATA FRIST BYTE
        MOV     R2,#3FH         ;63 BYTE TO DELETE
        CLR     A               ;CLEAR A AS ZERO
```

```
DELBYTEL:MOV     @R1,A
         INC     R1
         DJNZ    R2,DELBYTEL
         MOV     DSP,#00H
         RET

;CLEAR BIT SIGNAL TYPE DATA SAVE NO.1-9
CSTDS1:  MOV     A,R6
         CJNE    A,#01H,CSTDS2
         CLR     NO1
         RET
CSTDS2:  CJNE    A,#02H,CSTDS3
         CLR     NO2
         RET
CSTDS3:  CJNE    A,#03H,CSTDS4
         CLR     NO3
         RET
CSTDS4:  CJNE    A,#04H,CSTDS5
         CLR     04H
         RET
CSTDS5:  CJNE    A,#05H,CSTDS6
         CLR     NO5
         RET
CSTDS6:  CJNE    A,#06H,CSTDS7
         CLR     NO6
         RET
CSTDS7:  CJNE    A,#07H,CSTDS8
         CLR     NO7
         RET
CSTDS8:  CJNE    A,#08H,CSTDS9
         CLR     NO8
         RET
CSTDS9:  CJNE    A,#09H,CSTDSA
         CLR     NO9
CSTDSA:  RET

;SET BIT SIGNAL TYPE DATA SAVE NO.1-9
SSTDS1:  MOV     A,R6
```

```
          CJNE    A,#01H,SSTDS2
          SETB    NO1
          RET
SSTDS2:   CJNE    A,#02H,SSTDS3
          SETB    NO2
          RET
SSTDS3:   CJNE    A,#03H,SSTDS4
          SETB    NO3
          RET
SSTDS4:   CJNE    A,#04H,SSTDS5
          SETB    NO4
          RET
SSTDS5:   CJNE    A,#05H,SSTDS6
          SETB    NO5
          RET
SSTDS6:   CJNE    A,#06H,SSTDS7
          SETB    NO6
          RET
SSTDS7:   CJNE    A,#07H,SSTDS8
          SETB    NO7
          RET
SSTDS8:   CJNE    A,#08H,SSTDS9
          SETB    NO8
          RET
SSTDS9:   CJNE    A,#09H,SSTDSA
          SETB    NO9
SSTDSA:   RET

;WRITE SIGNAL TYPE DATA SAVE
WSTDS1:   MOV     A,R6
          CJNE    A,#01H,WSTDS2
          JB      NO1,WSTDSEW
WSTDSD:   MOV     A,#'D'
          RET
WSTDSEW:  MOV     A,#'E'
WSTDSEX:  RET
WSTDS2:   CJNE    A,#02H,WSTDS3
          JB      NO2,WSTDSEW
```

```
          SJMP     WSTDSD
WSTDS3:   CJNE     A,#03H,WSTDS4
          JB       NO3,WSTDSEW
          SJMP     WSTDSD
WSTDS4:   CJNE     A,#04H,WSTDS5
          JB       NO4,WSTDSEW
          SJMP     WSTDSD
WSTDS5:   CJNE     A,#05H,WSTDS6
          JB       NO5,WSTDSEW
          SJMP     WSTDSD
WSTDS6:   CJNE     A,#06H,WSTDS7
          JB       NO6,WSTDSEW
          SJMP     WSTDSD
WSTDS7:   CJNE     A,#07H,WSTDS8
          JB       NO7,WSTDSEW
          SJMP     WSTDSD
WSTDS8:   CJNE     A,#08H,WSTDS9
          JB       NO8,WSTDSEW
          SJMP     WSTDSD
WSTDS9:   CJNE     A,#09H,WSTDSEX
          JB       NO9,WSTDSEW
          SJMP     WSTDSD

LCDP0:    DB "**  NUCLEAR   **"
          DB "** RADIATION  **"
          DB "**  COUNTER   **"
          DB " < by KMIT'L  > "

LCDP1:    DB "   * READY *    "
          DB " PRESET         "
          DB " ELAPSE         "
          DB " COUNTS         "

LCDP2:    DB "* PRESET TIME * "
          DB "Press number-key"
          DB "and ENTER finish"
          DB "PRESET          "
```

```
LCDP3:   DB " * CLEAR DATA * "
         DB "1.TIME  2.COUNT "
         DB "3.BOTH          "
         DB "PRESS NUMBER 1-3"

LCDP4:   DB " * OVER FLOW *  "
         DB "COUNT FULL SCALE"
         DB "Press ENTER key "
         DB "  to continue   "

LCDPFT:  DB " * OVER FLOW *  "
         DB "TIMES FULL SCALE"
         DB "Press ENTER key "
         DB "  to continue   "

LCDPSI:  DB "* INPUT SIGNAL *"
         DB " 1.DETECTOR     "
         DB " 2.EXTERNAL WAVE"
         DB "PRESS NUMBER 1-2"

LCDLCO:  DB "  * COUNTING *  "

LCDPM:   DB "   * MEMORY *   "
         DB "1.VIEW   2.SAVE "
         DB "3.DELETE        "
         DB "PRESS NUMBER 1-3"

LCDPMV:  DB "   * VIEW *     "
         DB "VIEW No.1 To 9  "
         DB "Press number-key"
         DB " ENTER to EXIT  "

LCDPMVN: DB "VIEW No.   St.  "
         DB " TIMES          "
         DB " COUNTS         "
         DB "Key 0: View next"

LCDPMS:  DB "   * SAVE *     "
```

```
          DB "SAVE No.1 to 9  "
          DB "Press number-key"
          DB "Key 0: Save next"

LCDPMSN: DB "SAVE No.   St.  "
          DB " TIMES          "
          DB " COUNTS         "
          DB " ENTER to EXIT  "

LCDPMD:  DB "   * DELETE *   "
          DB "DELETE No.1 TO 9"
          DB "Press number-key"
          DB "Key 0:Delete all"

LCDPMF:  DB "* MEMORY  FULL *"
          DB " DATA NOT SAVE !"
          DB "Press ENTER key "
          DB "  to continue   "

          END

;******************************** END OF PROGRAM ********************************
```

เอกสารอ้างอิง

1. Ayala, Kenneth J. "The 8051 Microcontroller" West Publishing Company:United State of America, 1991
2. William J. Price "Nuclear Radiation Detection" Second Edition, page 115-141, McGraw-Hill:United State of America, 1964
3. J, Sharpe "Nuclear Radiation detection" page 158-176, Great Britain, 1955
4. Knoll, GlennF. "Radiation Detection and Measurement" Second Edition, John Wiley & Sons Inc.,:Singapore, 1989
5. ไพเสริฐ ธรรมานุธรรม "ฟิสิกส์รังสี" หน้า 103-119, โรงพิมพ์คุณพินอักษรกิจ: กรุงเทพฯ, พ.ศ. 2527
6. สุเจตน์ จันทรังสี "ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051" วิทยาลัยมหานคร:กรุงเทพฯ, พ.ศ. 2535
7. ละอองทิพย์ ชนะชัย "ฟิสิกส์ประยุกต์ทางชีวการแพทย์" กิ่งจันทร์การพิมพ์:กรุงเทพฯ, พ.ศ. 2526
8. ชัชวาลย์ ไทยอุบุญ และสุรเกียรติ อุตตมชัย "แหล่งจ่ายไฟแรงดันสูงสำหรับฮีเลียม-นีออนเลเซอร์" วิทยานิพนธ์ระดับปริญญาบัณฑิต ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2535
9. อารีรัตน์ คอนดวงแก้ว "การทดลองเรื่อง หัววัดไกเกอร์-มุลเลอร์" การฝึกอบรมหลักสูตรอุปกรณ์นิวเคลียร์อิเลคทรอนิกส์และการบำรุงรักษา, หน้า 1-9, สำนักงานปรมาณูเพื่อสันติ กระทรวงวิทยาศาสตร์ เทคโนโลยีและการพลังงาน, พ.ศ. 2533

ประวัติผู้แต่ง

นางสาวกรรณิการ์ กำจายกิตติกุล เกิดเมื่อวันที่ 14 สิงหาคม พ.ศ. 2515 ที่จังหวัดชลบุรี สำเร็จการศึกษาระดับประถมจากโรงเรียนวัดท้องคุ้ง จังหวัดชลบุรี เมื่อปี พ.ศ. 2526 ระดับมัธยมตอนต้น และมัธยมศึกษาตอนปลายจากโรงเรียน ชลบุรี "สุขบท" จังหวัดชลบุรี เมื่อปีการศึกษา พ.ศ. 2532 และได้เข้าศึกษาต่อในระดับอุดมศึกษาในสาขาวิชาฟิสิกส์ประยุกต์ (Solid state electronics) คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี พ.ศ. 2533 เลขประจำตัว 33504001

ประวัติการฝึกงาน

การปิโตรเลียมแห่งประเทศไทย ถนนวิภาวดีรังสิต กรุงเทพฯ

นายไชยกิจ กิจส่งแสง เกิดเมื่อวันที่ 18 ตุลาคม พ.ศ. 2514 ที่จังหวัดสมุทรสงคราม สำเร็จการศึกษาระดับประถมจากโรงเรียนวัดกระจับพินิจ เขตธนบุรี กรุงเทพฯ ในปี พ.ศ. 2526 ระดับมัธยมตอนต้น และมัธยมศึกษาตอนปลายจากโรงเรียนมัธยมวัดมกุฎกษัตริย์ เขตดุสิต กรุงเทพฯ ในปี พ.ศ. 2532 และเข้าศึกษาต่อในระดับอุดมศึกษาในสาขาวิชาฟิสิกส์ประยุกต์ (Solid state electronics) คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี พ.ศ. 2533 เลขประจำตัว 33504009

ประวัติการฝึกงาน

สำนักงานพลังงานปรมาณูเพื่อสันติ บางเขน กรุงเทพฯ