หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

ไมโครโพรเซสเซอร์และโปรแกรมควบคุมการทำงาน

ระบบปลุกเตือนนี้ใช้เครื่อง SMC 8080 (ซึ่งใช้ไมโครโพรเซสเซอร์เบอร์ 8080) ควบคุมการทำงานของระบบ โดยโปรแกรมต่าง ๆ จะถูกเก็บไว้ในหน่วยความจำกึ่งถาวร (EPROM) เบอร์ 2708 ใน SMC 8080 และสามารถถอดเปลี่ยนได้

6.1 เครื่อง SMC 8080

SMC 8080 เป็นไมโครคอมพิวเตอร์ซึ่งใช้ไมโครโพรเซสเซอร์เบอร์ 8080 เป็นหน่วยจัดการข้อมูลกลาง (CPU; Central Processing Unit) ใช้สัญญาณนาฬิกา (Clock) 2 ล้านครั้งต่อวินาที (M Hz.) มีหน่วยความจำชั่วคราว (RAM) 2 พันไบท์ และหน่วยความจำกึ่งถาวร (EPROM) 2 พันไบท์ ซึ่งใช้บรรจุโปรแกรมควบคุมป้อนข้อมูลด้วยคีบอร์ด (Keyboard) เป็นรหัสเลขฐาน 16 พร้อมฟังค์ชันคีย์ (Function Key) อีก 9 ตัว แสดงผลด้วยไดโอดเปล่งแสงเจ็ดส่วน 6 ตัว และไดโอดเปล่งแสงอีก 8 ตัว มีขั้วต่อสำหรับบัสแอดเดรส บัสข้อมูล และบัสควบคุม ซึ่งสามารถใช้ต่อกับวงจนภายนอกเพื่อใช้ควบคุมวงจรอีเล็กทรอนิคส์อื่น ๆ ได้ SMC 8080 ประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

1. หน่วยจัดการข้อมูลกลางซึ่งประกอบด้วยไอซีเบอร์

8080	1 ตัว
8224	1 ตัว
8228	1 ตัว
แร่บังคับความถี่ (Crystal) 18 ล้านครั้งต่อวินาที	1 ตัว

2. หน่วยความจำซึ่งประกอบด้วยไอซีเบอร์

2114 (RAM)	2 ตัว
(มีซอกเกตให้เพิ่มได้อีก 2 ตัว)	

2708 (EPROM) ซึ่งใช้เป็นรอม (ROM) 1 ตัว
(มีซอกเกตให้เพิ่มได้อีก 1 ตัว)

3. ดีสเพลยอินเตอร์เฟส (Display Interface) ประกอบด้วยไอซีเบอร์

555 1 ตัว
7445 1 ตัว
7489 2 ตัว
74 LS 193 1 ตัว
74 LS 240 1 ตัว

4. คีบอร์ดอินเตอร์เฟส (Keyboard Interface) ประกอบด้วยไอซีเบอร์

74 LS 175 1 ตัว
74 LS 240 1 ตัว

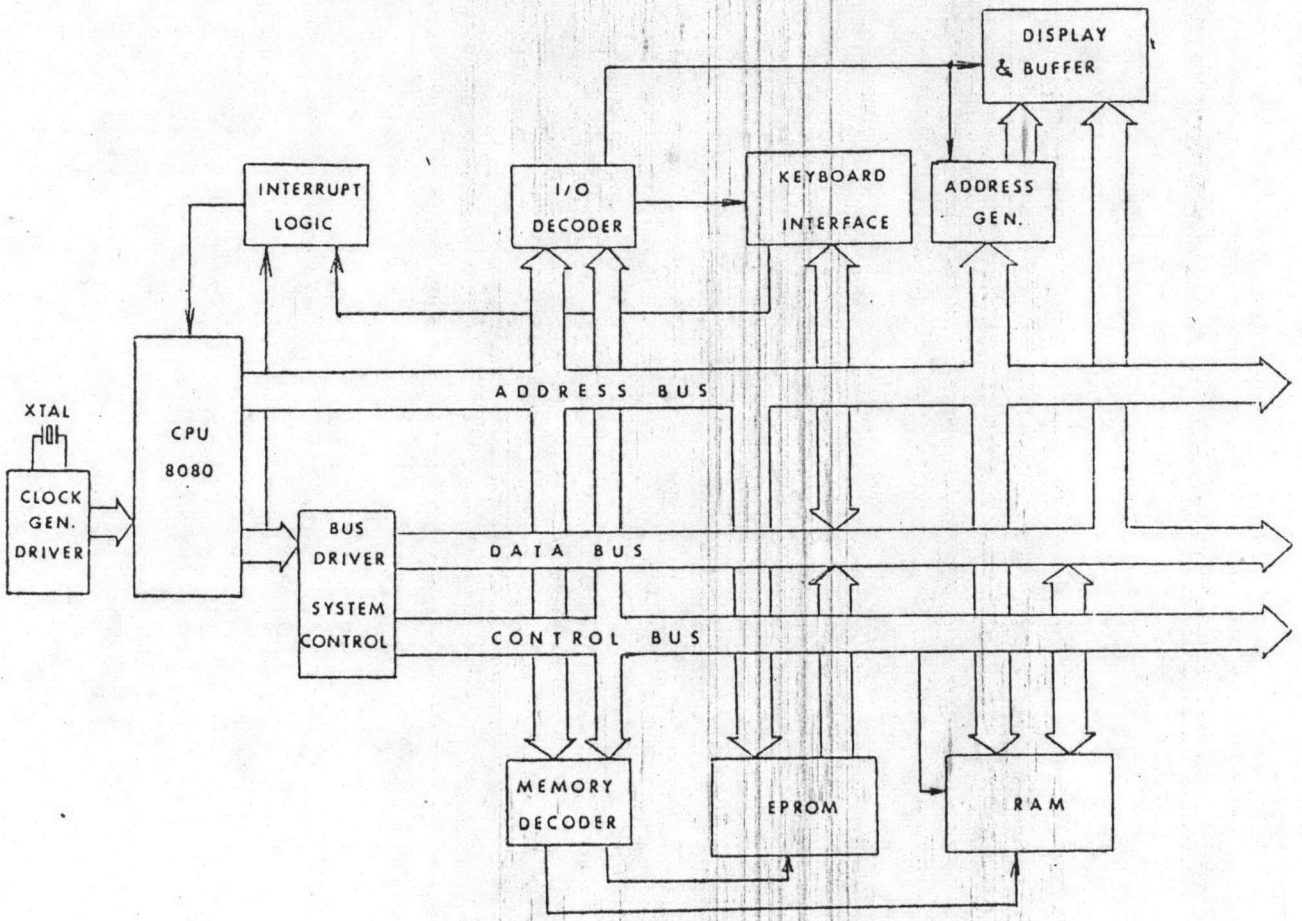
5. วงจรถอดรหัสและควบคุมลอจิก (Decoder and Control Logic) ประกอบด้วยไอซีเบอร์

74 LS 04 1 ตัว
74 LS 08 1 ตัว
74 LS 32 3 ตัว
74 LS 74 1 ตัว
74 LS 139 1 ตัว

6. คีย์บอร์ดสีเขียวซึ่งแสดงเป็นเลขฐาน 16 ทั้งหมด 16 ตัว (0 - F)

7. คีย์บอร์ดสีส้ม 9 ตัวคือ

H สำหรับ ตั้งแอดเดรสสูง
L " ตั้งแอดเดรสต่ำ
ST " เก็บข้อมูลไว้ที่แอดเดรสหรือรีจีสเตอร์ที่แสดง โดยไดโอดเปล่งแสงเจ็ดส่วน



รูปที่ 6.1.1 โครงสร้างของเครื่อง SMC 8080

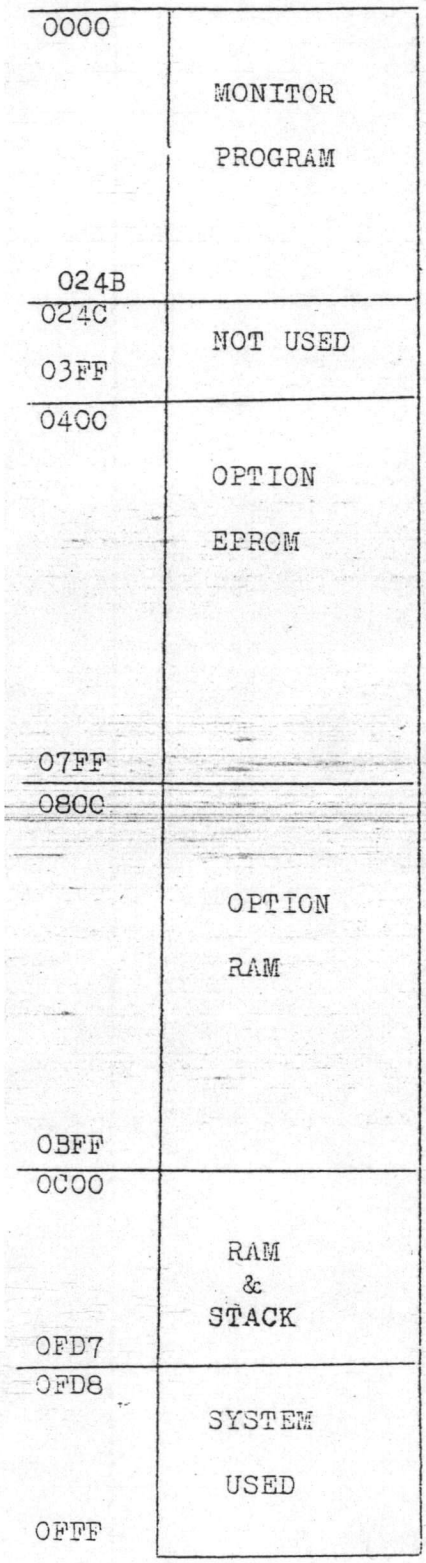
RG	สำหรับ	แสดงค่าของข้อมูลในรีจีสเตอร์
BS	"	ลดค่าแอดเดรสที่แสดงลงหนึ่ง
GO	"	เอ็กซีคิวท์โปรแกรมจากแอดเดรสที่แสดงจนจบโปรแกรม หรือจนพบเบรคพอยท์ (Break Point)
SS	"	เอ็กซีคิวท์โปรแกรมทีละคำสั่ง
BP	"	ตั้งแอดเดรสที่ต้องการให้โปรแกรมหยุด
RS	"	รีเซทให้เริ่มที่แอดเดรสเริ่มต้นของหน่วยความจำชั่วคราว พร้อมกับลบเบรคพอยท์ทั้งหมดออก

8. หน่วยแสดงประกอบด้วยไดโอดเปล่งแสงเจ็ดส่วน 6 ตัว โดยที่สี่ตัวแรกจะแสดงแอดเดรสหรือรีจีสเตอร์ ส่วนอีกสองตัวแสดงข้อมูลในหน่วยความจำที่แอดเดรสนั้นหรือข้อมูลในรีจีสเตอร์นั้น และไดโอดเปล่งแสง 8 ตัวเพื่อแสดงข้อมูลเช่นเดียว 2 ส่วนหลังของไดโอดเปล่งแสงเจ็ดส่วนแต่แสดงเป็นรหัสเลขฐาน 2

การทำงานของเครื่อง SMC 8080

หน่วยจัดการข้อมูลกลาง ประกอบด้วยไอซีกำเนิดสัญญาณนาฬิกา 8224 ซึ่งต่อกับแร่บังคับความถี่ 18 ล้านครั้งต่อวินาที โดย 8224 จะหาร 18 ล้านครั้งต่อวินาทีด้วย 9 เป็นสัญญาณนาฬิกา 2 เฟส (Phase) ที่เหลื่อมกัน $\emptyset_1$ และ $\emptyset_2$ การรีเซททำได้สองวิธีคือ เมื่อเปิดเครื่องจะรีเซทโดยอัตโนมัติ หรือโดยการกดปุ่มรีเซท (RS) การรีเซทจะส่งสัญญาณเข้าขา RESIN ของไอซี 8224 ขา RDYIN(READY) ของไอซี 8224 ต่อกับ +5V ผ่านความต้านทาน 10 กิโลโอห์ม ทำให้ 8080 พร้อมทำงานเสมอ การอินเทอร์รับ ทำได้ 2 วิธีคือ โดยใช้สัญญาณอินเทอร์รับจากภายนอก หรือใช้ซอฟแวร์ควบคุมเมื่อต้องการให้โปรแกรมทำงานทีละคำสั่ง

โปรแกรมควบคุมการทำงานหรือโปรแกรมมอนิเตอร์ของเครื่อง SMC 8080 นั้นเก็บไว้ในหน่วยความจำถาวร (EPROM) 2708 ซึ่งมีแอดเดรส $(0000)_{16}$ ถึง $(03\,FF)_{16}$ EPROM 2708 ที่จะเพิ่มได้ในแผ่นวงจรพิมพ์ถูกกำหนดให้มีแอดเดรส $(0400)_{16}$ ถึง $(07\,FF)_{16}$ ถ้าจะเพิ่มหน่วยความจำชั่วคราวด้วยไอซี 2114 อีก 2 ตัว จะมีหน่วยความจำเพิ่มขึ้นเป็น 2 กิโลไบท์ และ



รูปที่ 6.1.2 แผนผังหน่วยความจำของ SMC 8080

แอดเดรสเริ่มต้นเมื่อรีเซทจะเปลี่ยนจาก $(0000)_{16}$ เป็น $(0800)_{16}$ โดยอัตโนมัติ (ดูรูป 3)

การรับข้อมูลจากคีย์บอร์ดและการแปลรหัสนั้นควบคุมโดยซอฟแวร์ในโปรแกรมมอนิเตอร์ เมื่อไม่มีการกดคีย์บอร์ดไอซี 74 LS175 จะทำการสแกน (Scan) คีย์บอร์ดที่ละแถวทางแนวนอน ไอซี 74 LS240 จะเป็นตัวรับสัญญาณจากคีย์บอร์ดทางแนวตั้งซึ่งจะเป็น "1" เมื่อมีการกดคีย์ใดคีย์หนึ่ง เส้นสัญญาณทางแนวตั้งตรงกับคีย์ที่ถูกกดจะเป็น "0" ในเมื่อเรารู้ว่าขณะนั้นกำลังสแกนแถวทางแนวนอนแถวใดก็จะสามารถกำหนดตำแหน่งของคีย์นั้นได้ นอกจากนี้ยังมีซอฟแวร์ที่จะช่วยแก้การเบาซ์ (Bounce) ของการกดและปล่อยคีย์ด้วย

การแสดงข้อมูลกระทำได้โดยเอาข้อมูลที่จะแสดงมาเก็บไว้ในไอซี 7489 2 ตัว ซึ่งเป็นหน่วยความจำชั่วคราวขนาดเล็ก หลังจากนั้นจะนำค่าที่เก็บไว้ออกไปแสดงที่ไดโอดเปล่งแสงทีละตัวสลับกัน โดยที่ไอซี 74 LS193 และ 7445 เป็นตัวกำหนดตำแหน่งของไดโอดเปล่งแสงที่จะแสดงข้อมูล ไอซี 555 เป็นตัวกำหนดความเร็วของการสแกนให้เร็วพอที่จะทำให้ตามองเห็นว่าไดโอดเปล่งแสงทั้งหลายสว่างหรือดับตลอดเวลา

6.2 โปรแกรมควบคุมการทำงาน

โปรแกรมควบคุมการทำงานประกอบด้วยโปรแกรมต่าง ๆ ดังต่อไปนี้

โปรแกรมเริ่มต้น

โปรแกรมหลัก

ซับรูทีน "CALLING"

ซับรูทีน "DIAL"

ซับรูทีน "DELAY"

ซับรูทีน "SHIFT"

ซับรูทีน "MINUTES"

ซับรูทีน "INTRPT"

โปรแกรมเหล่านี้ถูกเก็บไว้ในหน่วยความจำของเครื่อง SMC 8080 ในส่วนที่เป็นรอม (ROM, Read Only Memory) โดยเริ่มตั้งแต่แอดเดรส (Address) ที่ 04 00 (ฐาน 16) ถึง 08FF โดยคำสั่งต่าง ๆ ได้รับการโปรแกรมไว้ในอีพรอม (EPROM, Erasable Programmable Read Only Memory) เบอร์ 2708 โดยใช้เครื่อง IMSAI 8080 อีพรอมนี้สามารถถอดออกและใส่เข้าในเครื่อง SMC 8080 ได้พอดี จึงสะดวกกับการแก้ไขการควบคุมการทำงานของระบบเพราะเพียงเปลี่ยนแต่โปรแกรมเท่านั้น

ในหน่วยความจำแบบแรม (RAM,Ramdon Accoss Memory) ส่วนซึ่งใช้งานระเริ่มตั้งแต่ 0E00 ถึง 0ED8 โดยที่

1) 0E00 - 0ED1 สำหรับเก็บข้อมูลในการสั่งปลุกเตือนทั้งหมด 30 ชุด หนึ่งชุดประกอบด้วย 7 ไบท์ (byte) ดังนั้นหน่วยความจำทั้งหมดคือ 210 ข้อมูลของแต่ละชุดจะถูกเก็บเรียงต่อกันไปเรื่อย ๆ โดย 3 ไบท์แรกจะเก็บหมายเลขโทรศัพท์ สี่ไบท์หลังจะเก็บเลขของเวลา

2) 0ED2 เก็บค่าของ M1 โดยที่ค่า M1 จะเปลี่ยนไปตามลำดับของการสั่งปลุกเตือน คือ เมื่อเริ่มต่อโทรศัพท์เข้ามา M1 จะเป็น 0

เมื่อหมุนหมายเลขโทรศัพท์ตัวแรก M1 = 1

เมื่อหมุนหมายเลขโทรศัพท์ตัวที่สอง M1 = 2

เมื่อหมุนหมายเลขโทรศัพท์ตัวที่สาม M1 = 3

เมื่อหมุนเวลาหลักสิบของชั่วโมง M1 = 4

เมื่อหมุนเวลาหลักหน่วยของชั่วโมง M1 = 5

เมื่อหมุนเวลาหลักสิบของนาที M1 = 6

เมื่อหมุนเวลาหลักหน่วยของนาที M1 = 0

(เพราะการสั่งปลุกเตือนสมบูรณ์แล้ว)

3) 0E D3 และ 0E D4 เก็บค่าแอดเดรสของหน่วยความจำสำหรับใช้เก็บข้อมูลตัวต่อไปโดย 0E D3 จะเก็บ M2L ซึ่งคือแอดเดรสต่ำ (Low Address) และ 0E D4 จะเก็บ M2H ซึ่งคือแอดเดรสสูง (High Address)

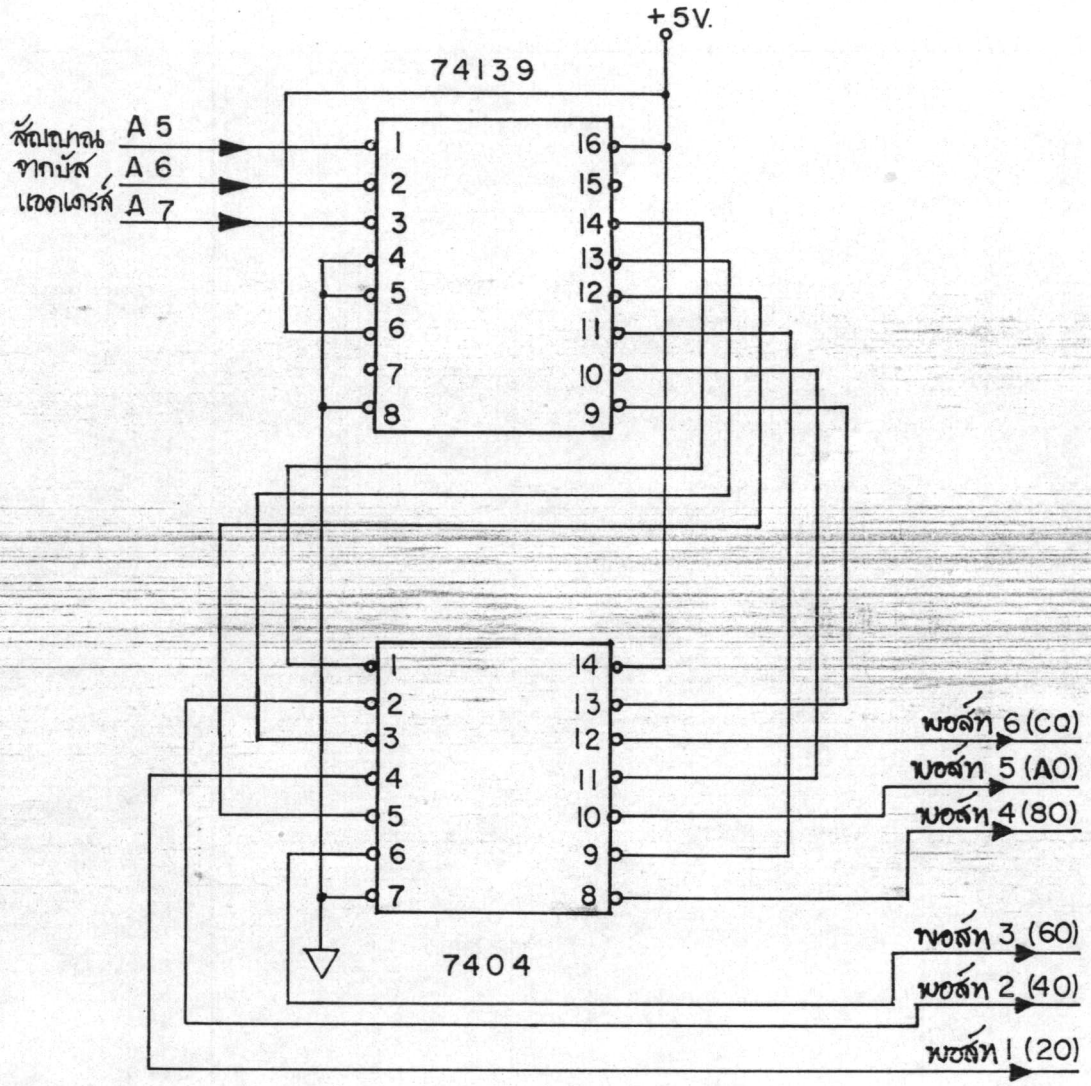
4) OE D5 และ OE D6 ใช้เก็บค่าของ REALTM ซึ่งคือเวลาของเวลาจริง ซึ่งได้แปลงเป็นนาทีแล้วโดยใช้ต่างกันสองไบท์ OE D6 เป็นไบท์แรก และ OE D5 เป็นไบท์หลัง

5) OE D7 และ OE D8 ใช้เก็บค่า STORETM ซึ่งคือค่าของเวลาสั่งปลุกเตือนที่ถูกแปลงเป็นนาทีแล้ว โดยใช้ต่อกันสองไบท์ OE D8 เป็นไบท์แรกและ OE D7 เป็นไบท์หลัง

เมื่อไมโครโพรเซสเซอร์ต้องการติดต่อกับวงจรต่าง ๆ ในระบบ ก็จะส่งสัญญาณ $\overline{\text{I/O R}}$ หรือ $\overline{\text{I/O W}}$ เพื่ออ่านค่าจากบัสข้อมูลหรือให้ค่ากับบัสข้อมูล พร้อมทั้งแอดเดรส (ตั้งแต่ A0 ถึง A7) เพื่อใช้เป็นสัญญาณเลือกพอร์ท สำหรับวงจรในระบบนี้ใช้พอร์ททั้งหมด 6 พอร์ท ซึ่งสัญญาณเลือกพอร์ทเหล่านี้จะตรงกับแอดเดรสต่าง ๆ ดังนี้ คือ

แอดเดรส	สัญญาณเลือกพอร์ท	สัญญาณขาออกสำหรับ
20	1	ส่งสัญญาณเสียงความถี่เดียวนานครึ่งวินาที
40	2	ส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที
60	3	ส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีแล้ววางโทรศัพ
80	4	ส่งวงจรส่งสัญญาณต่อโทรศัพท์ ยกโทรศัพท์
A0	5	สั่งวงจรส่งสัญญาณต่อโทรศัพท์ ส่งพัลส์ต่อโทรศัพท์

แอดเดรส	สัญญาณเลือกพอร์ท	สัญญาณขาเข้าสำหรับ
20	1	เช็ควาวงจรส่งสัญญาณต่อโทรศัพท์ทำการปลุกเตือนอยู่หรือไม่
40	2	ข้อมูลได้จากการหมุนเลขบนโทรศัพท์
60	3	ข้อมูลของหลักสิบของชั่วโมงของเวลา
80	4	ข้อมูลของหลักหน่วยของชั่วโมงของเวลา
A0	5	ข้อมูลของหลักสิบของนาทีของเวลา
C0	6	ข้อมูลของหลักหน่วยของนาทีของเวลา



รูปที่ 6.2 วงจรถอดรหัสสัญญาณเลือกพอร์ท

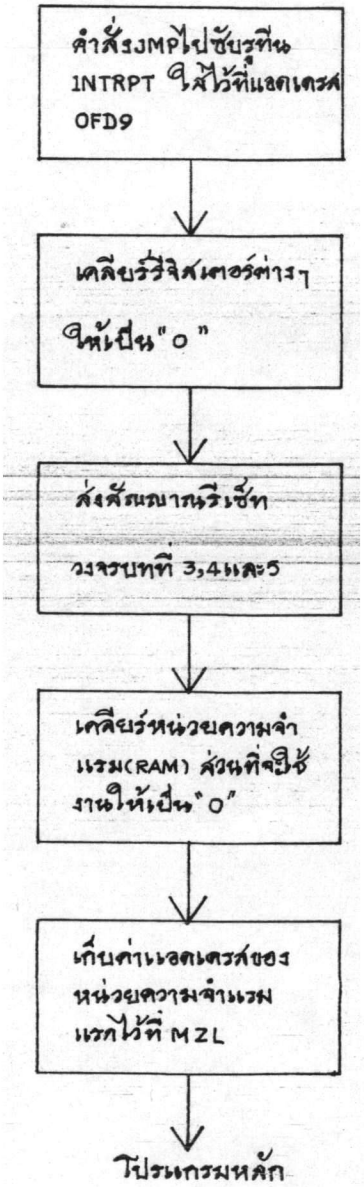
สัญญาณเลือกพอร์ทได้จากการใช้ไอซีสำหรับดอดรหัส (Decorder) เบอร์ 74139 โดยใช้แอดเดรสที่ A5, A6 และ A7 ต่อเข้ากับแอดเดรสบัสของเครื่อง SMC 8080 แล้วขับสัญญาณเลือกพอร์ทที่ได้โดยใช้ไอซีเบอร์ 7404 ดังรูปที่ 6.2

หมายเลขโทรศัพท์ที่สถาบันคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งได้ใช้เป็นสถานที่ทดลองระบบปลุกเตือนนี้ มีทั้งหมด 50 หมายเลข ดังต่อไปนี้คือ

4	11	81	91	171	181
5	12	82	92	172	182
6	13	83	93	173	183
7	14	84	94	174	184
	15	85	95	175	185
	16	86	96	176	186
		87	97	177	187
		88	98	178	188
		89	99	179	189
		80	90	170	180

6.2.1 โปรแกรมเริ่มต้น

เริ่มต้นการทำงานของระบบปลุกเตือนโดยเปิดสวิชเครื่องและเปิดสวิชเครื่อง SMC 8080 ตั้งแอดเดรสไว้ที่ 0400 แล้วกดปุ่ม GO โปรแกรมจะถูกเอ็กซิคิวท์ โดยจะเริ่มเอาคำสั่ง JMP ไปที่แอดเดรสของซับรูทีน "INTRPT" เก็บไว้ที่แอดเดรส OF D9 เพราะเมื่อเครื่อง SMC 8080 ได้รับสัญญาณอินเทอร์รับก็จะมาเริ่มเอ็กซิคิวท์ที่แอดเดรส OF D9 จากนั้นจะทำการเคลียร์รีจีสเตอร์ต่าง ๆ ให้เป็นศูนย์ แล้วส่งสัญญาณให้กับวงจรบทที่ 3,4 และ 5 เพื่อทำการรีเซ็ทวงจร จากนั้นจะทำการเคลียร์ค่าในหน่วยความจำแบบแรม ตั้งแต่แอดเดรสที่ OE 00 ถึงแอดเดรสที่ OE D8 ให้เป็นศูนย์หมด แล้วเอาค่าแอดเดรสแรกของแรม (ที่ใช้งาน) คือ OE 00 ใส่ไว้ในรีจีสเตอร์ HL แล้วจึงเข้าสู่โปรแกรมหลัก



6.2.1 ผังของโปรแกรมเริ่มต้น

โปรแกรมเริ่มต้น

```
BEGIN   DI
        MVI     A,0C3H
        STA     0FD9H
        LXI     H,INTRPT
        SHLD    0FDAH
        MVI     A,0
        MOV     B,A
        MOV     C,A
        MOV     D,A
        MOV     E,A
        OUT     60H
        OUT     80H
        OUT     0A0H
        LXI     H,0E00H
        PUSH    H
AAA     MVI     A,0
        MOV     M,A
        INX     H
        MVI     A,0D8H
        CMP     L
        JNZ     AAA
        MVI     A,0EH
        CMP     H
        JNZ     AAA
        POP     H
        SHLD    M2L
        EI
```

6.2.2 โปรแกรมหลัก

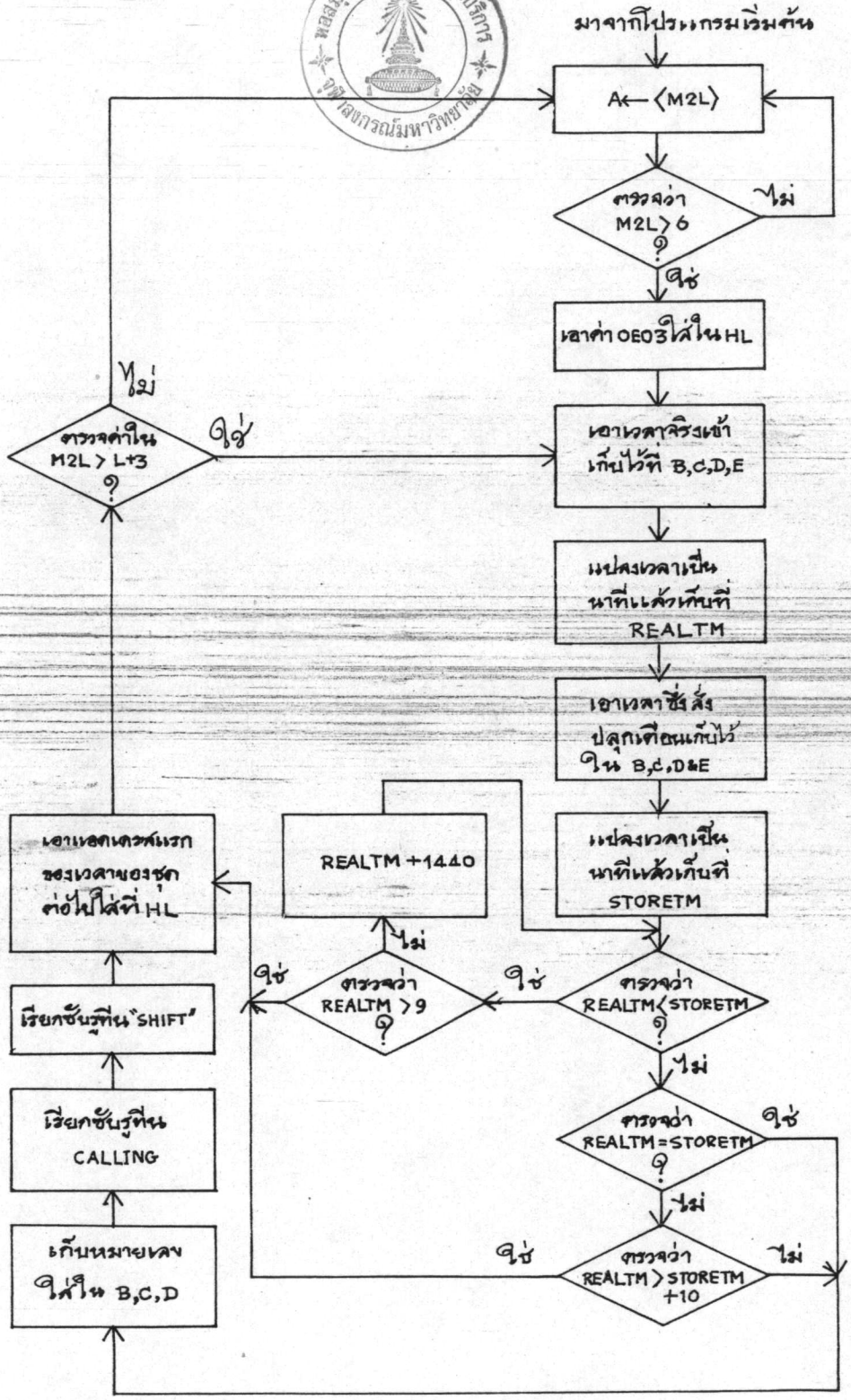
เมื่อโปรแกรมเริ่มต้นทำการ เคลียร์ค่าในหน่วยความจำส่วนที่จะใช้งานและรีเซ็ทวงจรต่าง ๆ แล้ว จึงเข้าสู่โปรแกรมหลัก โดยจะเริ่มเอาค่าใน M2L มาตรวจดูว่ามากกว่า 6 หรือไม่ ถ้าน้อยกว่าแสดงว่ายังไม่มีการสั่งปลุกเตือนถ้ามากกว่าแสดงว่าได้มีการสั่งปลุกเตือนแล้วอย่างน้อย 1 ชุด ก็จะทำการตรวจเช็คเวลา โดยจะเอาเวลาจริง เข้ามาแล้วแปลงให้เป็นนาทีเอาเก็บไว้ที่ REALTM แล้วจึงเอาเวลาของการสั่งปลุกเตือนชุดนั้นเข้ามา เพื่อแปลงให้เป็นนาทีแล้วเก็บไว้ที่ STORETM แล้วจึงทำการตรวจดูเวลาใน **REALTM** กับ STORETM ถ้า REALTM น้อยกว่า STORETM แสดงว่ายังไม่ถึงเวลา ก็จะไปทำการตรวจเช็คคำสั่งปลุกเตือนชุดต่อไป แต่ก่อนที่จะไปทำการตรวจเช็คข้อมูลชุดต่อไปจะต้องทำการตรวจเสียก่อนว่า REALTM มีค่าน้อยกว่า 9 หรือไม่ เพราะอาจจะเกิดกรณีที่ผู้สั่งปลุกเตือนสั่งปลุกเตือนเวลาตั้งแต่ 23 : 51 ถึง 23 :59 แต่เนื่องจากมีคำสั่งปลุกเตือนเวลาใกล้เคียงกันหลายชุดจนกระทั่งทำให้การตรวจเช็คเวลาต้องล่าช้าไป เมื่อโปรแกรมมาเช็คเวลาของคำสั่งชุดนี้ เวลาจริงอาจเลย 00 : 00 ซึ่งก็คือ 24 น. แต่เวลาสั่งปลุกเตือนเป็น 23 : 51 ถึง 23 : 59 ซึ่งจะทำให้การสั่งปลุกเตือนชุดนี้ถูกข้ามไปโดยที่ไมโครโพรเซสเซอร์เข้าใจว่ายังไม่ถึงเวลาเพราะเวลาจริงน้อยกว่าเวลาสั่งปลุกเตือน ดังนั้นถ้าเวลาจริงเป็นเวลาระหว่าง 00 : 00 ถึง 00 : 09 น. โปรแกรมจะทำการบวกค่า 1,440 (คือ 24 ชั่วโมง มี 1,440 นาที) เข้ากับค่าใน REALTM แล้วกลับไปตรวจเช็คเวลาอีกครั้ง

เมื่อตรวจพบว่าเวลาสั่งปลุกเตือนไม่มากกว่าเวลาจริง ก็จะทำการตรวจเช็คว่าเท่ากันหรือไม่ถ้าเท่ากันก็จะทำการสั่งปลุกเตือน ถ้าไม่เท่ากันก็จะมาตรวจดูว่าเวลาจริงมากกว่าเวลาสั่งปลุกเตือนถึง 10 นาทีหรือไม่ ถ้ามากกว่า 10 นาทีก็ให้ไปทำการเช็คเวลาของชุดต่อไป ถ้า REALTM น้อยกว่าหรือเท่ากับ STORETM+10 แล้วก็จะสั่งให้ปลุกเตือน

การสั่งปลุกเตือนทำโดยเอาหมายเลขของโทรศัพท์ซึ่งเก็บไว้ในหน่วยความจำมาเก็บไว้ในรีจีสเตอร์ B,C และ D โดยที่ B จะเก็บเลขตัวแรก C เก็บเลขตัวต่อมา และ B เก็บเลขตัวที่สาม ในกรณีที่หมายเลขโทรศัพท์มีไม่ครบ 3 เลข เลขที่เหลือจะเป็นศูนย์ แล้วจึงเรียกซับรูทีน

"CALLING" โปรแกรมของซับรูทีนนี้ก็จะทำการส่งสัญญาณต่อโทรศัพท์เพื่อปลุกเตือน แล้วจึงเรียกซับรูทีน "SHIFT" เพื่อเลื่อนเวลาของการสั่งปลุกเตือนชุดต่อ ๆ ไปมาแทนที่ แล้วจึงตรวจสอบว่าได้ตรวจเช็คหมดหรือยัง ถ้าหมดแล้วก็จะกลับไปเริ่มต้นการตรวจเช็คตั้งแต่ชุดแรกใหม่ ถ้ายังไม่หมดก็จะทำการตรวจเช็คชุดต่อไปจนกระทั่งหมด ทำเช่นนี้ตลอดไป

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.2.2 ผังของโปรแกรมหลัก

โปรแกรมหลัก

```
MAIN      LDA     0ED3H
          CPI     07H
          JC      MAIN
          LXI     H,0E03H
RDTIME    IN      60H
          MOV     B,A
          IN      80H
          MOV     C,A
          IN      0A0H
          MOV     D,A
          IN      0C0H
          MOV     E,A
          PUSH    H
          CALL    MINUTES
          SHLD    REALTM
          POP     H
          MOV     B,M
          INX     H
          MOV     C,M
          INX     H
          MOV     D,M
          INX     H
          MOV     E,M
          INX     H
          PUSH    H
          CALL    MINUTES
          SHLD    STORETM
          XCHG
          LHLD    REALTM
RECOMP    MOV     A,H
          CMP     D
          JC      TEST
          JNZ     TEST
          MOV     A,L
          CMP     E
          JC      TEST
          JZ      TELPHON
          PUSH    H
          LXI     H,000AH
          DAD     D
          XCHG
          POP     H
          MOV     A,D
          CMP     H
          JC      NEXT
          JNZ     TELPHON
```

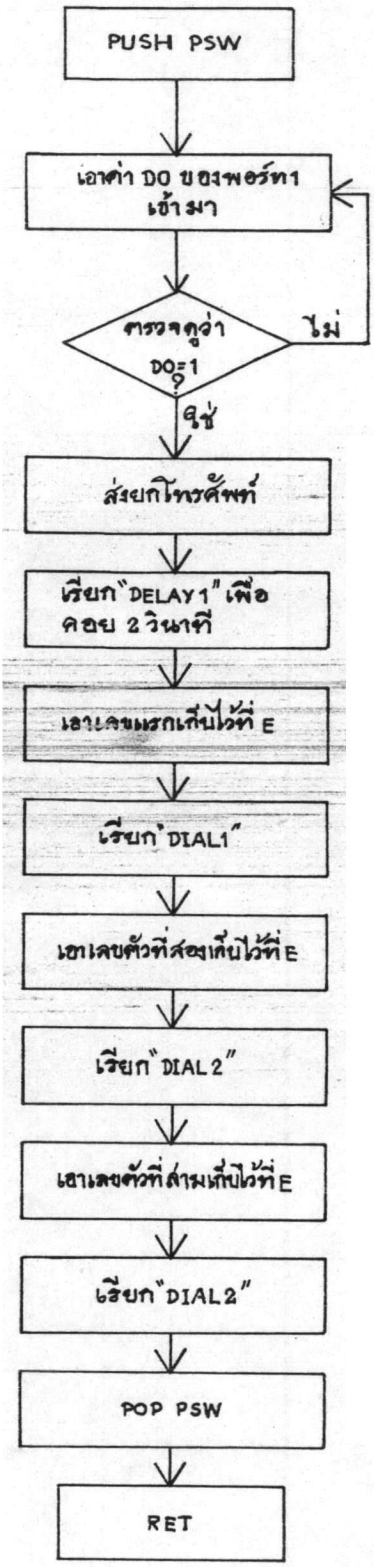
```
                MOV     A,E
                CMP     L
                JC      NEXT
TELPHON         POP     H
                DCX     H
                DCX     H
                DCX     H
                DCX     H
                DCX     H
                MOV     D,M
                DCX     H
                MOV     C,M
                DCX     H
                MOV     B,M
                CALL    CALLING
                CALL    SHIFT
                JMP     INXH
TEST            MOV     A,H
                CPI     O
                JNZ     NEXT
                MOV     A,L
                CPI     OAH
                JNC     NEXT
                LXI     D,05AOH
                DAD     D
                XCHG
                LHLD    STORETM
                XCHG
                JMP     RECOMP
NEXT            POP     H
INXH            INX     H
                INX     H
                INX     H
                PUSH    H
                INX     H
                INX     H
                INX     H
                INX     H
                XCHG
                POP     H
                LDA     M2L
                CMP     E
                JC      MAIN
                JMP     RDTIME

หมายเหตุ        REALTM  =   OE05
                STORETM =   OE07
```

6.2.3 ซับรูทีน "CALLING"

ก่อนที่ซับรูทีนนี้จะถูกเรียก หมายเลขโทรศัพท์ซึ่งต้องการสั่งปลุกเตือนจะถูกเก็บไว้ในรีจีสเตอร์ B,C และ D แล้ว โดย B จะเก็บเลขตัวแรก C จะเก็บเลขตัวที่สอง ส่วน D จะเก็บเลขตัวที่สาม ในกรณีที่หมายเลขโทรศัพท์มีไม่ครบสามตัวตัวที่เหลือจะเป็นศูนย์

เริ่มต้นโปรแกรมด้วยการ PUSH PSW แล้วจะเอาค่า D0 ของพอร์ท 1 เข้ามาเพื่อเช็คดูว่าโทรศัพท์กำลังสั่งปลุกเตือนอยู่หรือไม่ ถ้าเป็น "0" แสดงว่าเครื่องกำลังใช้สั่งปลุกเตือนอยู่ ก็จะทำการตรวจเช็คอยู่ตลอดเวลา เมื่อพบว่าเป็น "1" จึงจะส่งสัญญาณ I/O W พร้อมกับสัญญาณเลือกพอร์ท 4 เพื่อให้วงจรบทที่ 4 ทำการยกโทรศัพท์แล้วเรียกซับรูทีน "DELAY 1" เพื่อทำการดอย 2 วินาที แล้วจึงเอาค่าในรีจีสเตอร์ B เก็บไว้ในรีจีสเตอร์ E แล้วเรียกซับรูทีน "DIAL 1" เพื่อทำการส่งจำนวนพัลส์ไปตามคู่สายโทรศัพท์ตามเลขในรีจีสเตอร์ E แล้วจึงเอาค่าใน C เก็บไว้ใน E แล้วจึงเรียก "DIAL 2" เพื่อส่งจำนวนพัลส์ของเลขตัวที่สอง แล้วเอาค่าใน D เก็บไว้ที่ E แล้วเรียก "DIAL 2" อีกครั้งเพื่อส่งจำนวนพัลส์ของเลขตัวสุดท้าย เสร็จแล้ว POP PSW แล้วจึงกลับเข้าสู่โปรแกรมหลัก หมายเลขโทรศัพท์ซึ่งถึงเวลาตามที่สั่งปลุกเตือนไว้ก็จะได้รับการต่อเพื่อใช้เสียงกริ่งเป็นสัญญาณปลุกเตือน



รูปที่ 6.2.3 ผังของซับรูทีน "CALLING"

ซับรูทีน "CALLING"

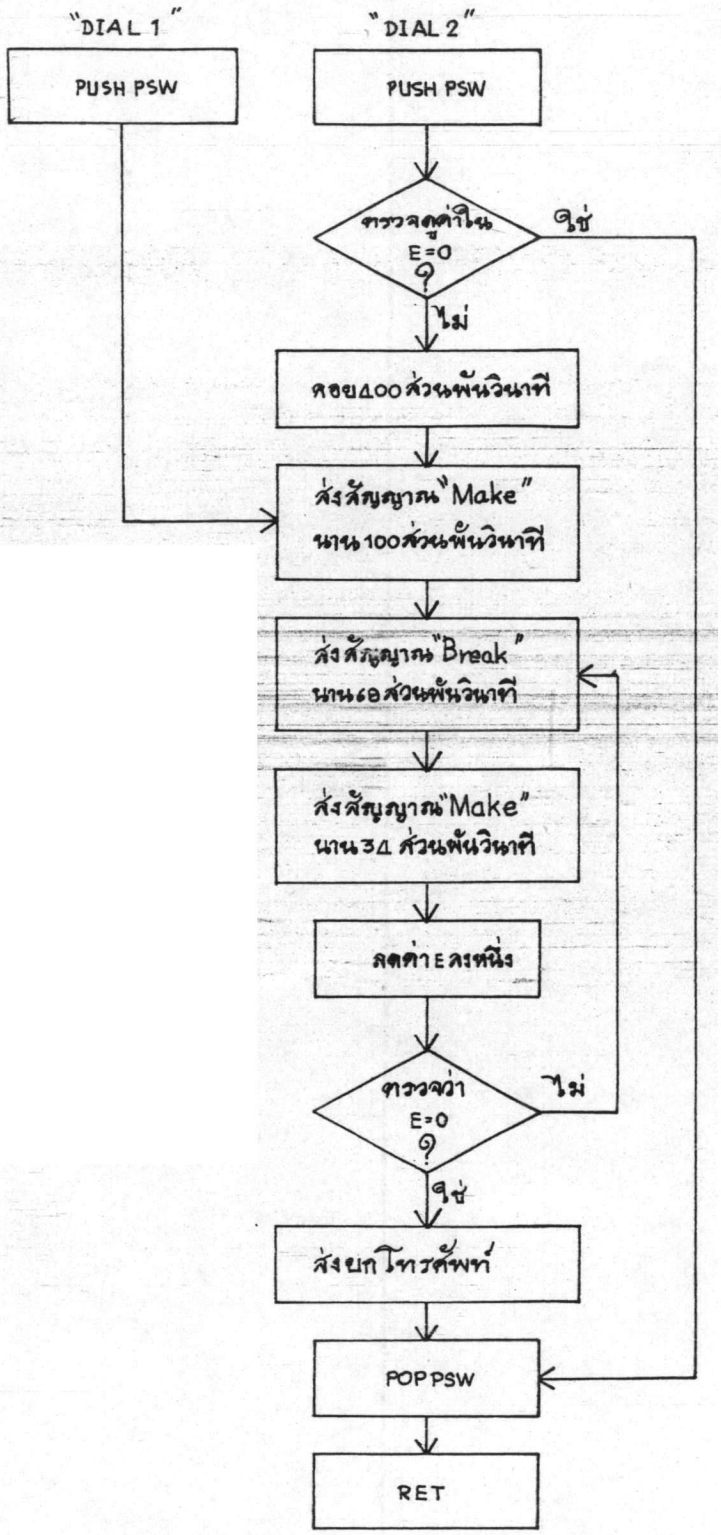
```
CALLING   PUSH        PSW
 BBB      IN          20H
          RAR
          JNC         BBB
          OUT         80H
          CALL        DELAY1
          DI
          MOV         E,B
          CALL        DIAL1
          EI
          DI
          MOV         E,C
          CALL        DIAL2
          EI
          DI
          MOV         E,D
          CALL        DIAL2
          EI
          POP         PSW
          RET
```

6.2.4 ซับรูทีน "DIAL"

ซับรูทีนนี้แยกได้เป็นซับรูทีนสองซับรูทีนซึ่งใช้คำสั่งเกือบเหมือนกันหมดแตกต่างกันเล็กน้อยเท่านั้นเอง คือซับรูทีน "DIAL 1" กับซับรูทีน "DIAL 2"

เมื่อเริ่มต่อหมายเลขตัวแรกซับรูทีน "DIAL 1" จะถูกเรียก คำสั่ง $\overline{\text{I/O W}}$ พร้อมกับสัญญาณเลือกพอร์ท 5 โดย D0 มีค่าเป็น "1" และ D1 มีค่าเป็น "1" จะถูกส่งให้กับวงจรบทที่ 4 เพื่อทำการต่อสายโทรศัพท์เข้าด้วยกันซึ่งก็คือการส่งสัญญาณ "Make" นั่นเอง แล้วเรียกซับรูทีน "DELAY 2" สามครั้งติดต่อกันเพื่อให้ได้สัญญาณ "Make" นาน 100 ส่วนพันวินาที แล้วจึงส่งสัญญาณ $\overline{\text{I/O W}}$ พร้อมกับสัญญาณเลือกพอร์ท 5 โดยค่า D0 เป็น "0" และ D1 เป็น "1" เพื่อเป็นการแยกสายโทรศัพท์ออกจากกันซึ่งก็คือการส่งสัญญาณ "Break" นั่นเอง แล้วจึงเรียกซับรูทีน "DELAY 2" สองครั้งติดต่อกันเพื่อให้ได้สัญญาณ "Break" นาน 68 ส่วนพันวินาที แล้วจึงส่งสัญญาณ $\overline{\text{I/O W}}$ พร้อมกับสัญญาณเลือกพอร์ท 5 โดยที่ D0 เป็น "1" และ D1 เป็น "1" เพื่อส่งสัญญาณ "Make" แล้วเรียกซับรูทีน "DELAY 2" เพื่อให้สัญญาณ "Make" นาน 34 ส่วนพันวินาที เป็นการส่งพัลส์ไปตามคู่สายโทรศัพท์หนึ่งลูก แล้วจึงทำการลดค่า E ลงหนึ่งตรวจดูว่าค่า E เป็นศูนย์หรือยัง ถ้ายังกลับไปทำการส่งพัลส์ลูกต่อไปทำเช่นนี้ไปจนกระทั่ง E เป็นศูนย์ แสดงว่าทำการส่งพัลส์ครบจำนวนตามเลขซึ่งอยู่ใน E แล้ว จึงส่งสัญญาณ $\overline{\text{I/O W}}$ พร้อมกับสัญญาณเลือกพอร์ท 5 โดยที่ทั้ง D0 และ D1 มีค่าเป็น "0" ให้กับวงจรบทที่ 4 จะทำให้รีเรย์ 12 ต่อคู่สายโทรศัพท์ไปยังตัวต้านทาน 100 โอมห์ ซึ่งเปรียบเสมือนการยกหูโทรศัพท์ไว้ธรรมดา ดังนั้น ขบวนพัลส์สำหรับเลขตัวแรกก็จะถูกส่งไปตามคู่สายโทรศัพท์

สัญญาณขบวนพัลส์ของเลขตัวที่สองและสามจะถูกส่งออกโดยซับรูที "DIAL 2" เมื่อซับรูทีนนี้ถูกเรียก รีจีสเตอร์ E จะถูกเช็คดูว่ามีค่าเป็นศูนย์หรือไม่ หากเป็นศูนย์ก็จะกลับเข้าสู่ซับรูทีน "CALLING" เลย หากไม่ก็จะทำการคอยนาน 400 ส่วนพันวินาที โดยการเรียกซับรูทีน "DELAY 3" แล้วจึงทำเช่นเดียวกับซับรูทีน "DIAL 1"



รูปที่ 6.2.4 ผังของซับรูทีน "DIAL"

ซับรูทีน "DIAL"

```
DIAL1   PUSH    PSW
        JMP     CCC
DIAL2   PUSH    PSW
        MVI     A,0
        CMP     E
        JZ      DDD
        CALL    DELAY3
CCC     MVI     A,3
        OUT     0AOH
        CALL    DELAY2
        CALL    DELAY2
        CALL    DELAY2
EEE     MVI     A,2
        OUT     0AOH
        CALL    DELAY2
        CALL    DELAY2
        MVI     A,3
        OUT     0AOH
        CALL    DELAY2
        DCR     E
        JNZ     EEE
        MVI     A,0
        OUT     0AOH
DDD     POP     PSW
        RET
```

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

6.2.5 ซับรูทีน "DELAY"

ประกอบด้วยซับรูทีน "DELAY 1", "DALAY 2" และ "DELAY 3" ซับรูทีน "DELAY 1" และซับรูทีน "DELAY 2" ทำงานเหมือนกันเพียงแต่ให้ค่าในรีจีสเตอร์ B ต่างกันเท่านั้นเอง ส่วน "DELAY 3" เป็นซับรูทีนซึ่งเรียกซับรูทีน "DELAY 2" หลาย ๆ ครั้งตามจำนวนในรีจีสเตอร์

1) ซับรูทีน "DELAY 1" จะทำให้โปรแกรมคอยอยู่สองวินาที โดยคำนวณได้ตามเวลาของคำสั่งต่าง ๆ ที่ใช้ในซับรูทีนดังนี้

จำนวนคล็อก (Clock) ทั้งหมด

$$= (\text{PUSH B})+(\text{MVI B})+(\text{JMP})+(\text{PUSH PSW})+(\text{PUSH D})+b1\{\text{MVI D}+d1[\text{MVI E}+e1(\text{DCR E}+\text{JNZ})+\text{DCR D}+\text{JNZ}]+\text{DCR B}+\text{JNZ}\}+(\text{POP D})+(\text{POP PSW})+(\text{POP B})+(\text{RET})$$

$$= 11+7+10+11+11+b1\{7+d1[7+e1(5+10)+5+10]+5+10\}+10+10+10+10$$

$$= 90+b1[22+d1(22+15e1)]$$

$$\therefore \text{เวลาคอย} = \frac{90+b1[22+d1(22+15e1)]}{\text{ความถี่ของคล็อก}}$$

โดยที่ความถี่ = 2 ล้านครั้งต่อวินาที

เมื่อแทนค่าความถี่ของคล็อกแล้วใช้วิธีไทรแอลแอนด์เออเรอร์ (Trial and Error) แล้วจะได้ค่า

$$b1 = 60$$

$$d1 = 18$$

$$e1 = 246$$

ซึ่งทำให้ได้เวลาคอย $= \frac{90+60[22+18(22+15\times246)]}{2\times10^6}$

$= 2.005185$ วินาที

2) ซับรูทีน "DELAY 2" จะทำให้โปรแกรมคอยอยู่เป็นเวลา 34 ส่วนพันวินาที โดยคำนวณได้ตามเวลาของคำสั่งต่าง ๆ ที่ใช้ในซับรูทีนดังนี้

จำนวนคล็อกทั้งหมด

$$= (\text{PUSH B})+(\text{MVI B})+(\text{PUSH PSW})+(\text{PUSH D})+b2\left\{\text{MVI D}+d2\left[\text{MVI E}+e2(\text{DCR E}+\text{JNZ})+\text{DCR D}+\text{JNZ}\right]+\text{DCR B}+\text{JNZ}\right\}+(\text{POP D})+(\text{POP PSW})+(\text{POP B})+(\text{RET})$$

$$= 11+7+11+11+b2\left\{7+d2\left[7+e2(5+10)+5+10\right]+5+10\right\}+10+10+10+10$$

$$= 80+b2\left[22+d2(22+15e2)\right]$$

$$\therefore \text{เวลาคอย} = \frac{80+b2\left[22+d2(22+15e2)\right]}{\text{ความถี่ของคล็อก}}$$

แก้สมการโดยวิธีไทรแอลแอนด์เออเรอร์ จะได้ค่า

$$b2 = 1$$

$$d2 = 18$$

$$e2 = 246$$

ซึ่งทำให้ได้เวลาคอย $= \dfrac{80+1\left[22+18(22+15\times246)\right]}{2\times10^{6}}$

$= 33.459$ ส่วนพันวินาที

3) ซับรูทีน "DELAY 3" จะทำให้โปรแกรมคอยอยู่เป็นเวลา 400 ส่วนพันวินาที โดยคำนวณได้ดังนี้

เวลาที่คอย = เวลาคอยของ "DELAY 2" x ค่าในรีจีสเตอร์ A

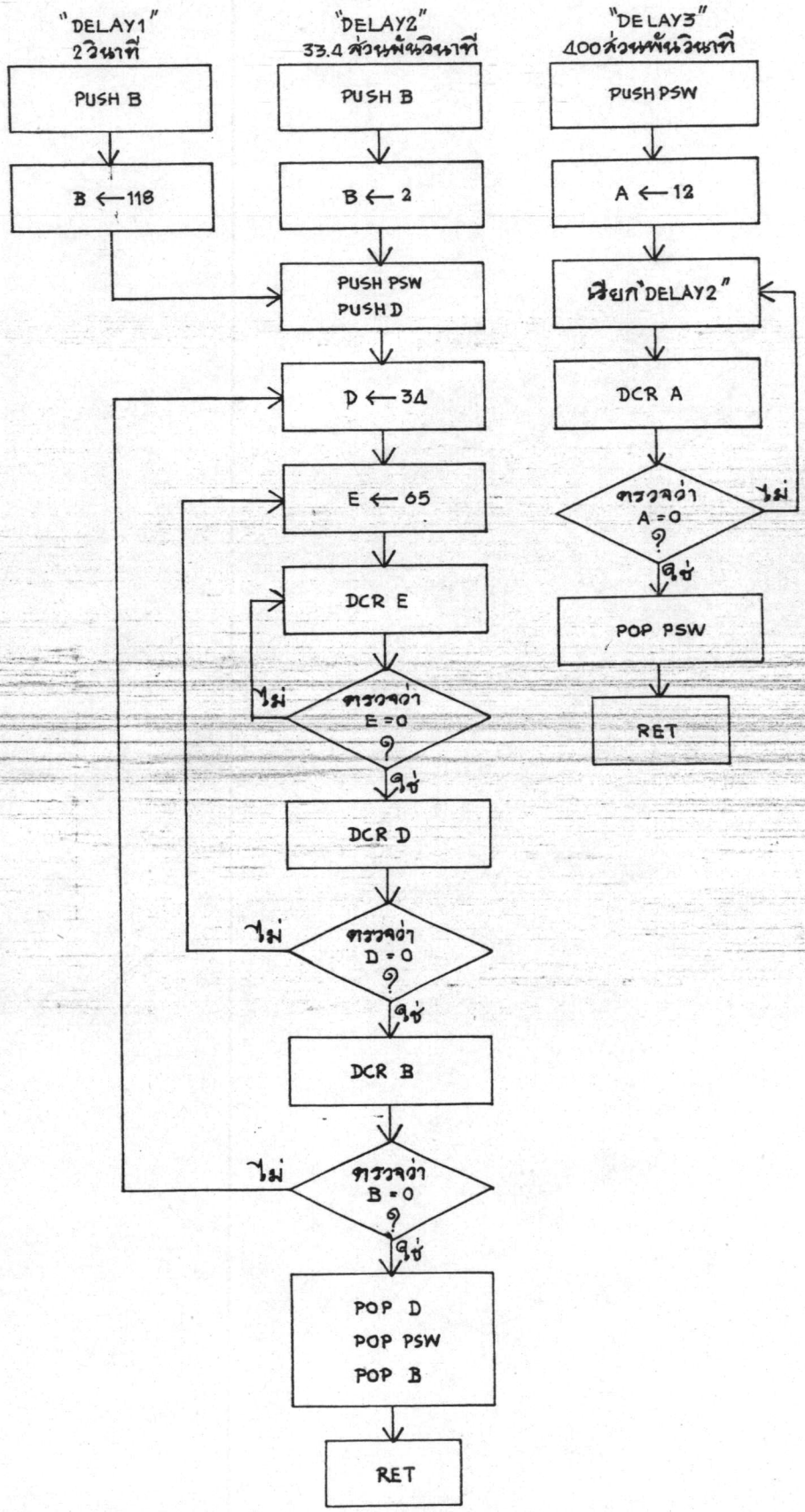
$\therefore$ ค่าในรีจีสเตอร์A = 400/33.459

= 11.95

ดังนั้นใช้ค่าในรีจีสเตอร์ A = 12

จะทำให้ได้เวลาคอย = 401.5 ส่วนพันวินาที

ซับรูทีนเหล่านี้จะทำให้ไมโครโพรเซสเซอร์ทำงานวนอยู่ในลูบ (Loop) เพื่อเป็นการคอยตามเวลาดังได้คำนวณไว้ข้างต้น



รูปที่ 6.2.5 ผังของซับรูทีน "DELAY"

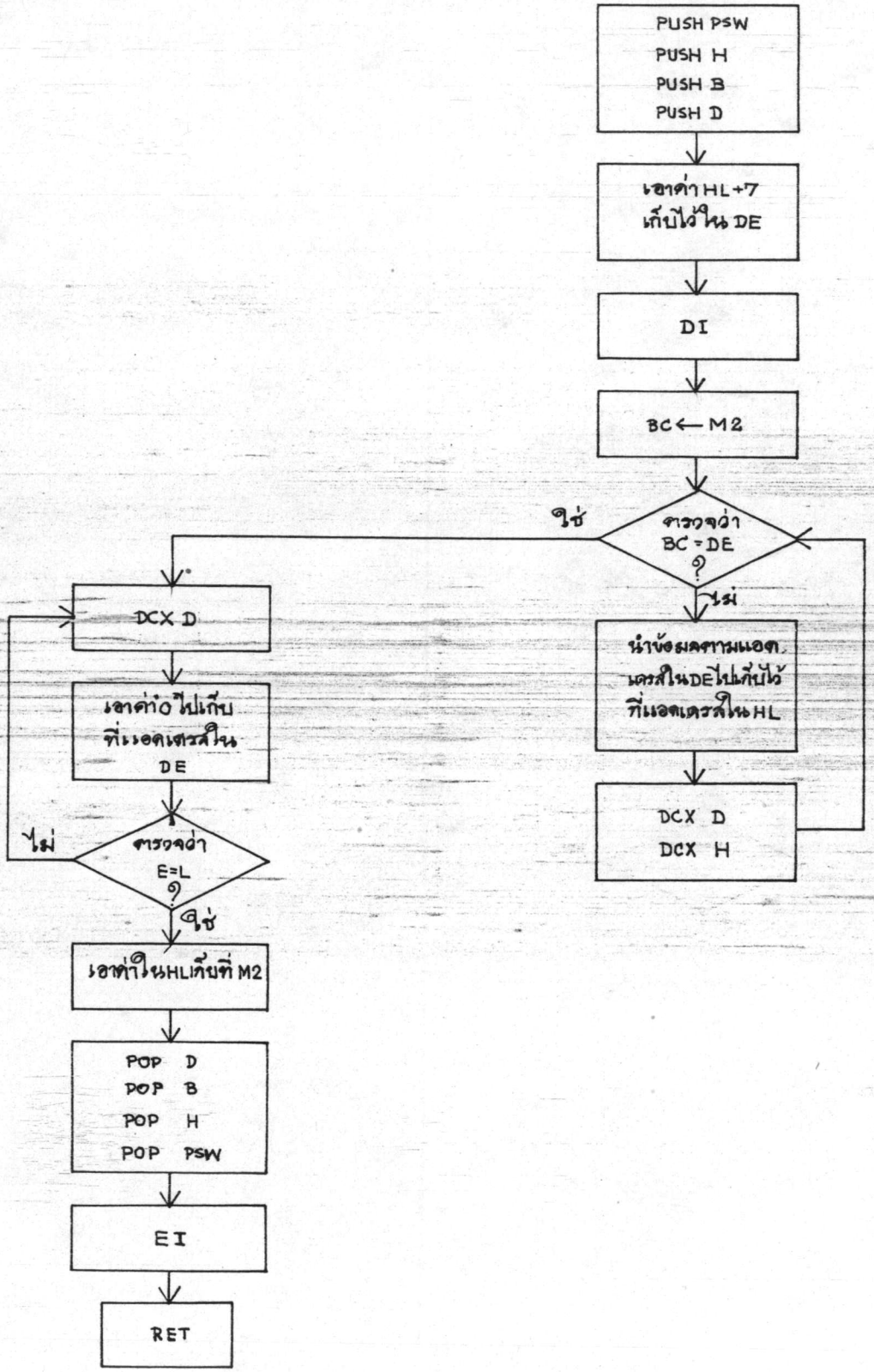
ซับรูทีน "DELAY"

```
DELAY1    PUSH    B
          MVI     B,03CH
          JMP     FFF
DELAY2    PUSH    B
          MVI     B,1
FFF       PUSH    PSW
          PUSH    D
GGG       MVI     D,12H
HHH       MVI     E,0F6H
III       DCR     E
          JNZ     III
          DCR     D
          JNZ     HHH
          DCR     B
          JNZ     GGG
          POP     D
          POP     PSW
          POP     B
          RET
DELAY3    PUSH    PSW
          MVI     A,0CH
JJJ       CALL    DELAY2
          DCR     A
          JNZ     JJJ
          POP     PSW
          RET
```

6.2.6 ซับรูทีน "SHIFT"

ซับรูทีนนี้ทำหน้าที่ย้ายข้อมูลในหน่วยความจำโดยจะเลื่อนข้อมูลในหน่วยความจำตั้งแต่แอดเดรสที่ HL+7 ขึ้นไปมาแทนที่ ดังนั้นข้อมูลตั้งแต่แอดเดรสที่ HL+7 จะถูกเขยิบไปเก็บไว้ที่แอดเดรสต่ำลง 7 ทุก ๆ ไบท์ไป ดังนั้นข้อมูลเก่าตั้งแต่แอดเดรสที่ HL จนถึงแอดเดรสที่ HL+6 จะถูกแทนที่ด้วยข้อมูลซึ่งอยู่ที่แอดเดรสสูงกว่าแทน

ก่อนที่ซับรูทีนนี้จะถูกเรียก แอดเดรสซึ่งเก็บเลขตัวแรกของหมายเลขโทรศัพท์ซึ่งได้ทำการปลุกเตือนไปแล้ว จะถูกเก็บไว้ใน HL เมื่อซับรูทีนนี้ถูกเรียกโปรแกรมจะสั่งให้ PUSH PSW PUSH H, PUSH B, PUSH D และ PUSH H จากนั้นจะเอาค่า HL+7 เก็บไว้ใน DE แล้วสั่ง DI เพื่อกันความสับสนในการเก็บข้อมูลของการสั่งปลุกเตือน แล้วเอาค่าใน M2 เก็บไว้ที่ BC ทำการเทียบค่าใน BC กับ DE ว่าเท่ากันหรือไม่ ถ้าไม่เท่ากันก็ทำการย้ายข้อมูลซึ่งแอดเดรสอยู่ที่ DE ไปเก็บไว้ที่แอดเดรสซึ่งอยู่ใน HL แล้วเพิ่มค่า DE และ HL ขึ้น 1 แล้วทำการตรวจเช็คว่าค่าใน BC เท่ากับ DE หรือยังถ้ายังก็จะทำการย้ายข้อมูลไปเรื่อย ๆ จนกระทั่ง BC เท่ากับ DE แล้วจึงจะทำการเคลียร์ข้อมูล 7 ไบท์สุดท้ายให้เป็นศูนย์ โดยการเอาศูนย์ไปเก็บไว้ในแอดเดรสซึ่งอยู่ใน DE แล้วลดค่า DE ลง 1 แล้วเทียบดูว่า DE เท่ากับ HL หรือยัง ถ้ายังก็เอาศูนย์ไปใส่ไว้ในแอดเดรสซึ่งอยู่ใน DE ทำไปเรื่อย ๆ จนกระทั่งได้ค่า DE เท่ากับ HL แสดงว่าได้ทำการเคลียร์ข้อมูลซึ่งไม่ต้องการแล้ว จึงเอาค่าแอดเดรสซึ่งพร้อมจะรับข้อมูลของการสั่งปลุกเตือนเก็บไว้ที่ M2 แล้วจึง POP D, POP B, POP H และ POP PSW แล้วสั่ง EI แล้วจึงกลับเข้าซับรูทีน "CALLING" ต่อไป



รูปที่ 6.2.6 ผังของซับรูทีน "SHIFT"

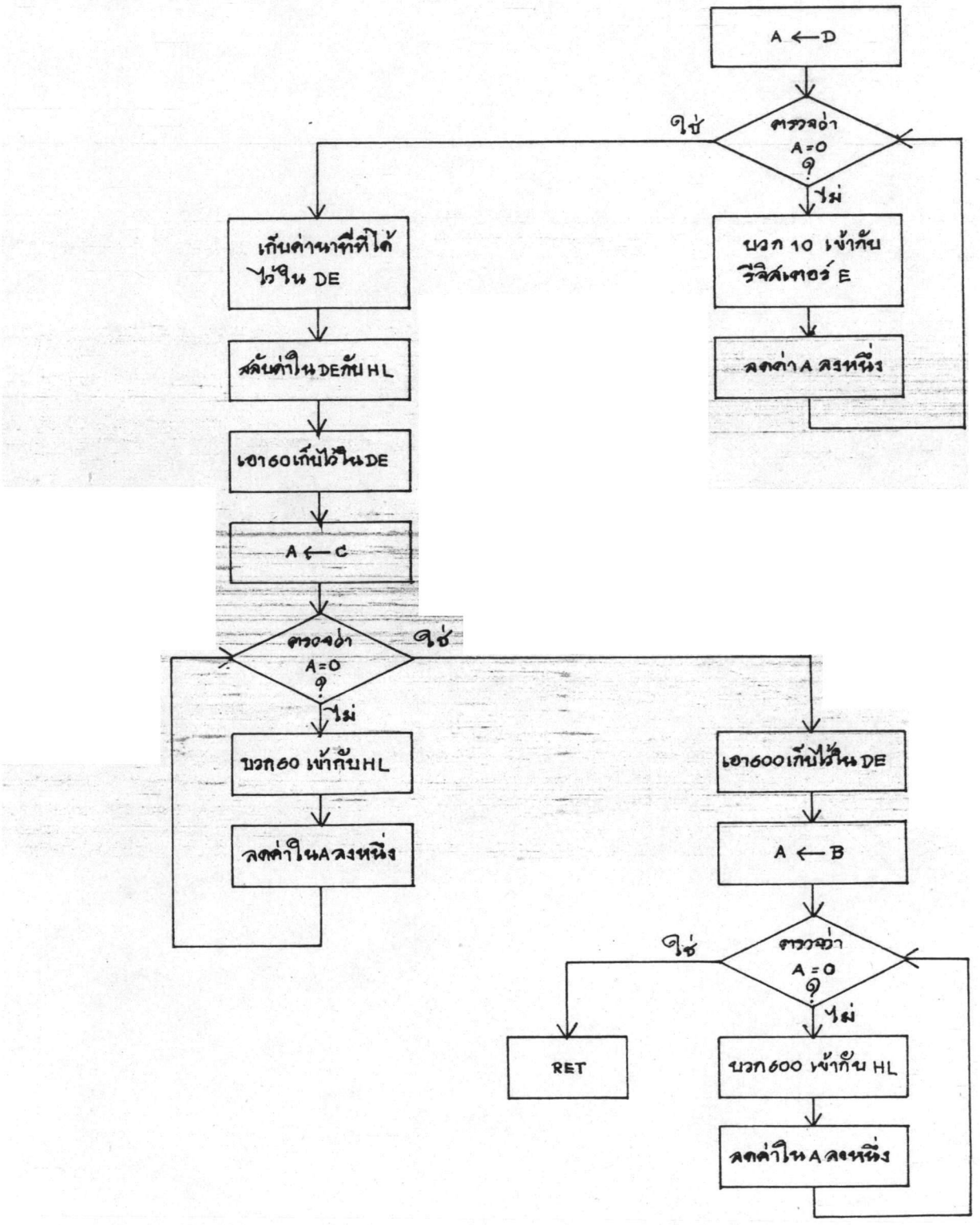
ซับรูทีน "SHIFT"

```
SHIFT   PUSH    PSW
        PUSH    H
        PUSH    B
        PUSH    D
        PUSH    H
        LXI     D,7
        DAD     D
        XCHG
        DI
        LHLD    M2L
        PUSH    H
        POP     B
        POP     H
A4      MOV     A,B
        CMP     D
        JNZ     A5
        MOV     A,C
        CMP     E
        JZ      A6
A5      LDAX    D
        MOV     M,A
        INX     D
        INX     H
        JMP     A4
A6      MVI     A,0
        DCX     D
        STAX    D
        MOV     A,E
        CMP     L
        JNZ     A6
        SHLD    M2L
        POP     D
        POP     B
        POP     H
        POP     PSW
        EI
        RET
```

6.2.7 ซับรูทีน "MINUTES"

ก่อนที่ซับรูทีนนี้จะถูกเรียก ตัวเลขหลักสิบของชั่วโมงจะถูกนำมาไว้ที่ B ตัวเลขหลักหน่วยของชั่วโมงจะถูกนำมาไว้ที่ C ตัวเลขหลักสิบของนาทีจะถูกนำมาไว้ที่ D และตัวเลขหลักหน่วยของนาทีจะถูกนำมาไว้ที่ E

เมื่อซับรูทีนถูกเรียกก็จะทำการแปลงค่าหลักสิบของนาทีในรีจีสเตอร์ D ให้เป็นนาทีแล้วบวกเข้ากับค่าในรีจีสเตอร์ E แล้วเก็บค่านี้ไว้ในรีจีสเตอร์ DE (ค่าในรีจีสเตอร์ D จะเป็นศูนย์) แล้วสลับค่าในรีจีสเตอร์ของ HL กับ DE แล้วทำการแปลงหลักหน่วยของชั่วโมงดังตัวเลขในรีจีสเตอร์ C ให้เป็นนาที แล้วบวกกับค่าเดิมใน HL เก็บไว้ใน HL โดยเช็คค่าใน C ว่าเป็นศูนย์หรือไม่ ถ้าไม่เป็นก็ทำการบวก 60 เข้ากับรีจีสเตอร์ HL แล้วลดค่าใน C ลงหนึ่งและเช็คดูว่าเป็นศูนย์หรือยัง ถ้ายังก็เอา 60 ไปบวกเข้ากับ HL อีก แล้วลดค่าใน C ลงหนึ่ง แล้วกลับไปเช็คค่าในรีจีสเตอร์ C อีก ทำจนกระทั่งค่าใน C เป็นศูนย์แล้วจึงไปทำการแปลงเวลาของเลขหลักสิบของชั่วโมงในรีจีสเตอร์ B โดยเทียบค่า B ดูว่าเท่ากับศูนย์หรือไม่ ถ้าไม่ให้ทำการบวก 600 เท่ากับค่าใน HL แล้วลดค่า B ลงหนึ่ง แล้วกลับไปเทียบค่า B ใหม่ ทำเช่นนี้เรื่อยไปจนกระทั่ง B เป็นศูนย์ จึงกลับเข้าสู่โปรแกรมหลักต่อไป ดังนั้นเวลาจะถูกแปลงเป็นนาทีแล้วเก็บไว้ในรีจีสเตอร์ HL



รูปที่ 6.2.7 ผังของซับรูทีน "MINUTES"

ซับรูทีน "MINUTES"

```
MINUTES   MOV     A,D
COMPD     CPI     0
          JZ      ADHOUR1
          PUSH    PSW
          MVI     A,0AH
          ADD     E
          MOV     E,A
          POP     PSW
          DCR     A
          JMP     COMPD
ADHOUR1   MVI     D,0
          XCHG
          LXI     D,003CH
          MOV     A,C
COMPC     CPI     0
          JZ      ADHOR10
          DAD     D
          DCR     A
          JMP     COMPC
ADHOR10   LXI     D,0258H
          MOV     A,B
COMPB     CPI     0
          RZ
          DAD     D
          DCR     A
          JMP     COMPB
```

6.2.8 ซับรูทีน "INTRPT"

ซับรูทีนนี้เป็นโปรแกรมซึ่งใช้ควบคุมการรับข้อมูลของตัวเลขซึ่งเกิดจากการหมุนเลขบนเครื่องโทรศัพท์ เมื่อเลขบนเครื่องโทรศัพท์ถูกหมุน สัญญาณขบวนพัลส์จะถูกส่งจากเครื่องโทรศัพท์ซึ่งสั่งปลุกเตือนให้กับวงจรรับสัญญาณจากเครื่องโทรศัพท์ (บทที่ 5) วงจรรี้จะเปลี่ยนสัญญาณขบวนพัลส์ที่ได้รับให้เป็นสัญญาณขนานในรูปของรหัสเลขฐานสอง แล้วจะส่งสัญญาณ $\overline{INT}$ ให้กับ SMC 8080 เครื่องไมโครโพรเซสเซอร์ก็จะส่งสัญญาณ $\overline{INTA}$ มาให้วงจรรับสัญญาณจากเครื่องโทรศัพท์ เพื่ออ่านสัญญาณคำสั่ง เริ่มต้นใหม่ จากบัสข้อมูล ซึ่งจะอ่านค่าได้คือ 11,001,111 (CF เลขฐานสิบหก) คำสั่งนี้จะทำให้ไมโครโพรเซสเซอร์หยุดการเอ็กซิคิวท์โปรแกรมซึ่งกำลังทำอยู่ แล้วไปเริ่มเอ็กซิคิวท์โปรแกรมที่แอดเดรส 00 08 ที่แอดเดรสนี้ในหน่วยความจำของเครื่อง SMC 8080 จะมีคำสั่ง JMP OF D9 อยู่ ดังนั้นในโปรแกรมเริ่มต้นจึงได้ให้ใส่คำสั่ง JMP ไปที่ซับรูทีน "INTRPT" ไว้ที่แอดเดรส OF D9 ดังนั้นเมื่อไมโครโพรเซสเซอร์ได้รับสัญญาณ $\overline{INT}$ จึงเรียกซับรูทีน "INTRPT" ได้

ในซับรูทีนนี้มีการเซ็ทค่าดรรชนีของลำดับขั้นในการสั่งปลุกเตือน และดรรชนีนี้ถูกเก็บไว้ในหน่วยความจำที่แอดเดรส OE D2 หรือเรียกว่า M1 เมื่อซับรูทีนนี้ถูกเรียกค่าดรรชนีนี้จะถูกนำไปไว้ในรีจีสเตอร์ C เพื่อนำไปใช้งานในซับรูทีนต่อไป ลำดับขั้นของการสั่งปลุกเตือนบอกไว้โดยจำนวนเลขของดรรชนีดังได้อธิบายไว้ในตอนต้นของบท

เมื่อซับรูทีนนี้ถูกเรียกก็จะทำการ PUSH PSW, PUSH B, PUSH D นำค่าดรรชนีใน M2 ไปเก็บไว้รีจีสเตอร์ C และนำค่าแอดเดรสสำหรับเก็บข้อมูลจาก M2 ไปเก็บไว้ในรีจีสเตอร์ DE แล้วจึงทำการเช็คดูว่าการสั่งปลุกเตือนทั้งหมดมีมากกว่า 30 ชุดหรือไม่ (สามารถรับได้เต็มที่ 30 ชุด) โดยการเช็คดูว่าค่าแอดเดรสซึ่งบรรจุอยู่ใน DE มากกว่าค่า OE D1 หรือไม่ เพราะถ้ามากกว่า (จะเป็น OE D2 แสดงว่าข้อมูลของการสั่งปลุกเตือนถูกเก็บไว้เต็ม 30 ชุดแล้ว ถ้าไม่มากกว่าแสดงว่ายังรับข้อมูลได้ ก็จะทำการรับข้อมูลจากวงจรรับสัญญาณจากเครื่องโทรศัพท์มาเก็บไว้ที่รีจีสเตอร์ A แล้วเช็คดูว่า A น้อยกว่า 2 หรือไม่ ถ้าไม่น้อยกว่า 2 แสดงว่าข้อมูลที่รับมาเกิดจากการหมุนเลขบนเครื่องโทรศัพท์ ถ้าน้อยกว่า 2 แสดงว่าข้อมูลที่รับมาเป็นข้อมูลซึ่งเกิดจากการวางหูโทรศัพท์ ถ้าเป็นข้อมูลซึ่งเกิดจากการหมุนเลขโทรศัพท์ก็จะลดค่าใน A ลงหนึ่ง เพื่อให้จำนวนใน A

ตรงกับตัวเลขจริงแล้วเก็บไว้ใน D เพื่อความสะดวกในการเอ็กซิคิวท์โปรแกรม ถ้าเป็นข้อมูลซึ่งเกิดจากการวางโทรศัพท์ก็จะทำการเคลียร์ข้อมูลเก่าของการสั่งปลุกเตือนชุดนี้ออกจากหน่วยความจำ โดยจะเช็คดูว่าค่า C เป็นศูนย์หรือไม่ ถ้าเป็นแสดงว่าเป็นการหมุนเลขเลขแรกก็ไม่จำเป็นต้องเคลียร์ข้อมูลเก่า โปรแกรมจะทำการส่งค่าสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีแล้วตามด้วยการวางโทรศัพท์ ถ้า C ไม่เป็นศูนย์ก็จะทำการเคลียร์ข้อมูลเก่าโดยลดค่าใน DE ลงหนึ่งแล้วเอาค่าศูนย์ไปเก็บไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วลดค่า C ลงหนึ่งแล้วกลับไปเช็คค่า C ใหม่ว่าเท่ากับศูนย์หรือไม่ ทำเช่นนี้จนกระทั่ง C เป็นศูนย์

ถ้าเป็นข้อมูลซึ่งเกิดจากการหมุนเลขโทรศัพท์ ก็จะตรวจเช็คค่า C ว่าเป็นศูนย์หรือไม่ ถ้าไม่แสดงว่าเป็นเลขตัวแรก แล้วจึงทำการเช็คดูว่าข้อมูลใน B มีค่าเท่ากับ 1, หรือ 8 หรือ 9 หรือไม่ (ถ้าหมายเลขโทรศัพท์ขึ้นต้นด้วย 1 หรือ 8 หรือ 9 แสดงว่าจำนวนเลขของหมายเลขโทรศัพท์มีมากกว่าหนึ่งขึ้นไป) ถ้าใช่ก็จะทำการเก็บค่าใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้น 1 เพิ่มค่าดรรชนีขึ้น 1 แล้วสั่งส่งสัญญาณเสียงความถี่เดียวนานครึ่งวินาที (OUT 20) แล้วกลับเข้าเข้าสู่โปรแกรมเดิม

ถ้าข้อมูลใน B ไม่เท่ากับ 1 หรือ 8 หรือ 9 ก็เช็คว่าเท่ากับ 4 หรือ 5 หรือ 6 หรือ 7 หรือไม่ ถ้าไม่แสดงว่าเป็นเลขซึ่งไม่มีในระบบ โปรแกรมจะสั่งให้ส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที แล้ววางโทรศัพท์ (OUT 60) แล้วกลับเข้าโปรแกรมเดิม ถ้าใช่แสดงว่าเป็นหมายเลขซึ่งเป็นเลขตัวเดียว ก็จะเก็บค่าใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่าใน DE ขึ้น 3 และเพิ่มค่าใน C ขึ้น 3 แล้วสั่งส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที (OUT 40) แล้วกลับเข้าสู่โปรแกรมเดิม

ถ้าค่าใน C ไม่เท่ากับศูนย์ก็เทียบว่าเท่ากับหนึ่งหรือไม่ ถ้าไม่ก็ไปเทียบต่อว่าเท่ากับ 2 หรือไม่ ถ้าค่าใน C เท่ากับหนึ่ง แสดงว่าได้รับเลขของหมายเลขโทรศัพท์มาหนึ่งตัวแล้ว ดังนั้นจึงต้องเช็คดูก่อนว่าเลขตัวก่อนคือเลขอะไร ถ้าเลขตัวก่อนเท่ากับ 8 หรือ 9 แสดงว่าหมายเลขซึ่งสั่งปลุกเตือนเป็นหมายเลขซึ่งเป็นเลขสองตัว ดังนั้นจะทำการเก็บเลขใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่าใน DE ขึ้นสอง และเพิ่มค่าใน C ขึ้นสอง แล้วสั่งส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีแล้วจึงกลับเข้าโปรแกรมเดิม ถ้าค่าใน B ไม่เท่ากับ 8 หรือ 9 ก็เช็ค

อีกว่ามากกว่าหกหรือไม่ ถ้าไม่แสดงว่าหมายเลขเป็นเลขสองตัวก็ทำเช่นเดียวกับกรณีเลขนั้นขึ้นต้นด้วย 8 หรือ 9 ถ้ามากกว่าหกแสดงว่าเป็นเลขสามตัวก็ทำการเช็คอีกว่ามากกว่า 8 หรือไม่ ถ้ามากกว่าแสดงว่าหมายเลขโทรศัพท์ที่สั่งปลุกเตือนไม่มีในระบบโทรศัพท์นี้ โปรแกรมจะสั่งให้ส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีตามด้วยการวางโทรศัพท์ (OUT 60) แล้วกลับเข้าสู่โปรแกรมเดิม ถ้าค่า B ไม่มากกว่า 8 แสดงว่าเป็นหมายเลขชนิดเลขสามตัวก็จะทำการเก็บค่า B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่งและเพิ่มค่า C ขึ้นหนึ่ง แล้วส่งสัญญาณเสียงความถี่เดียวนานครึ่งวินาที (OUT 20)

ถ้าค่า C ไม่เท่ากับหนึ่งเช็คดูว่าเท่ากับสองหรือไม่ถ้าไม่ให้ไปเช็คค่า C อีกว่าเท่ากับสามหรือไม่ ถ้าค่า C เท่ากับสองก็จะทำการเก็บค่าใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่ง และเพิ่มค่า C ขึ้นหนึ่ง แล้วส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที (OUT 40)

ถ้าค่า C ไม่เท่ากับสองก็ทำการเช็คว่าค่า C เท่ากับสามหรือไม่ แต่ก่อนเช็คว่าค่า C เท่ากับสามหรือไม่นั้นต้องเช็คก่อนว่าข้อมูลที่รับมาเท่ากับสิบหรือไม่ ถ้าเท่ากับสิบต้องทำให้เป็นศูนย์เสียก่อน (เพราะเมื่อหมุนตั้งเวลาเลข "0" จะได้รับสัญญาณพัลส์ 10 ลูก) ถ้าเช็คได้ว่าค่าใน C ไม่เท่ากับสามก็ไปเช็คว่าเท่ากับสี่หรือไม่ ถ้าเช็คได้ว่าค่า C เท่ากับสามก็เช็คดูว่าค่าใน B มากกว่าสองหรือไม่ ถ้ามากกว่าสองแสดงว่าเป็นเวลาซึ่งไม่มีจริง (หลักสิบของชั่วโมงไม่เกินสอง) ก็จะทำการเคลียร์ข้อมูลก่อนที่เก็บไว้ พร้อมกับนำแอดเดรสสำหรับการเก็บข้อมูลแรกของชุดต่อไปไปเก็บไว้ที่ DE กับเคลียร์ค่าใน C ให้เป็นศูนย์ แล้วส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีตามด้วยการวางโทรศัพท์ (OUT 60) ถ้าค่าใน B ไม่มากกว่าสองก็จะทำการเก็บค่าใน B ไว้ที่แอดเดรสซึ่งบรรจุไว้ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่ง และเพิ่มค่าใน C ขึ้นหนึ่ง แล้วสั่งส่งสัญญาณเสียงความถี่เดียวนานครึ่งวินาที (OUT 20) แล้วกลับเข้าสู่โปรแกรมเดิม

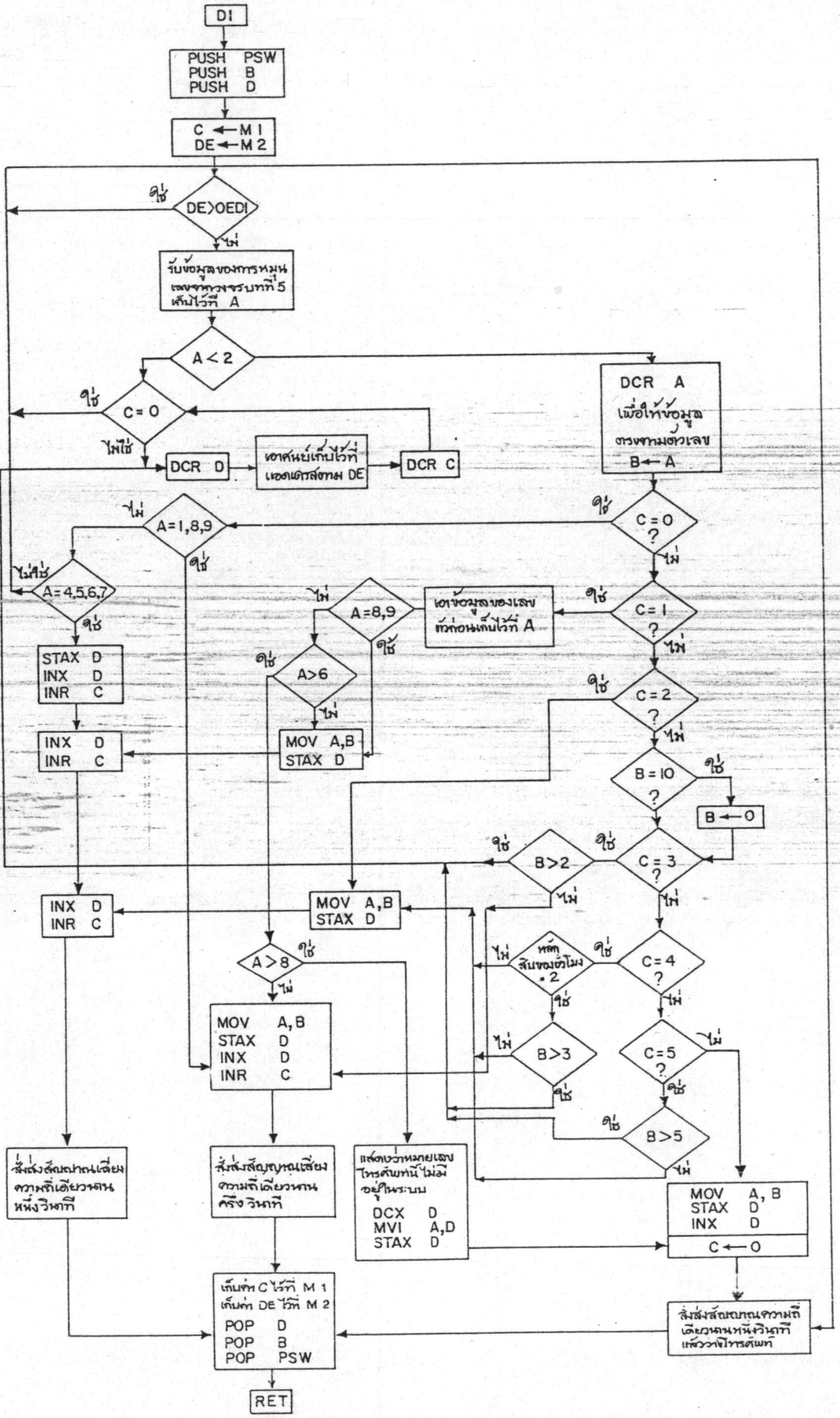
ถ้าค่า C ไม่เท่ากับสามก็จะเช็คอีกว่า C เท่ากับสี่หรือไม่ ถ้าไม่เท่ากับสี่ให้ไปเช็คดูว่า C เท่ากับห้าหรือไม่ ถ้าค่าใน C เท่ากับสี่ก็จะทำการเช็คเลขหลักสิบของชั่วโมงก่อนว่าเท่ากับสองหรือไม่ ถ้าไม่เท่ากับสองก็จะทำการเก็บของมูลใน B ไว้ที่แอดเดรสซึ่งบรรจุไว้ใน DE แล้วเพิ่ม DE ขึ้นหนึ่ง และเพิ่ม C ขึ้นหนึ่ง แล้วส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที (OUT 40)

แล้วกลับเข้าโปรแกรมเดิม ถ้าหลักสิบของชั่วโมงเท่ากับสองก็ทำการเช็คว่าค่าใน B มากกว่าสามหรือไม่ ถ้ามากกว่าสามแสดงว่าเป็นเวลาซึ่งไม่มีจริง (หลักหน่วยของชั่วโมงมีค่าไม่เกินสาม) ถ้าไม่มากกว่าสามจึงทำการเก็บค่าใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่ง และเพิ่มค่า C ขึ้นหนึ่ง แล้วส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาที (OUT 40) แล้วกลับเข้าสู่โปรแกรมเดิม

ถ้าค่า C ไม่เท่ากับสี่ก็ทำการเช็คอีกครั้งว่า ค่า C เท่ากับห้าหรือไม่ ถ้าเท่ากับห้าก็จะเช็คว่าค่าใน B มากกว่าห้าหรือไม่ ถ้าค่าใน B มากกว่าห้า (เป็นเวลาซึ่งไม่มีจริง) ก็จะทำการเคลียร์ข้อมูลก่อนที่เก็บไว้ในหน่วยความจำพร้อมกับนำแอดเดรสสำหรับการเก็บข้อมูลแรกของการสั่งปลุกเตือนครั้งต่อไปเก็บไว้ที่ DE กับเคลียร์ค่าใน C ให้เป็นศูนย์ แล้วส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีตามด้วยการวางโทรศัพท์ (OUT 60) แล้วกลับเข้าสู่โปรแกรมเดิม ถ้าค่าใน B ไม่มากกว่าห้าก็จะทำการเก็บข้อมูลใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่งและเพิ่มค่า C ขึ้นหนึ่ง แล้วส่งสัญญาณเสียงความถี่เดียวนานครึ่งวินาที (OUT 20) แล้วกลับเข้าสู่โปรแกรมเดิม

ถ้าค่าใน C ไม่เท่ากับห้า ก็ทำการเก็บข้อมูลใน B ไว้ที่แอดเดรสซึ่งบรรจุอยู่ใน DE แล้วเพิ่มค่า DE ขึ้นหนึ่ง แล้วเคลียร์ค่า C เป็นศูนย์ แล้วสั่งส่งสัญญาณเสียงความถี่เดียวนานหนึ่งวินาทีตามด้วยการวางโทรศัพท์ (OUT 60) แล้วกลับเข้าสู่โปรแกรมเดิม แสดงว่าการสั่งปลุกเตือนครั้งนี้ได้เสร็จสมบูรณ์แล้ว

ทุกครั้งก่อนที่จะกลับเข้าสู่โปรแกรมเดิมนั้น จะทำการเก็บค่าใน C ไว้ที่ M1 และเก็บค่า DE ไว้ที่ M2 ก่อน แล้วจึง POP D, POP B และ POP PSW แล้วจึงกลับเข้าสู่โปรแกรมเดิม เพื่อว่าเมื่อไมโครโพรเซสเซอร์ได้รับสัญญาณอินเทอร์รัปอีก จะได้รู้ว่าจะต้องเก็บข้อมูลไว้ที่แอดเดรสใด (M2) และลำดับขั้นของการเก็บข้อมูลก็เช็คได้จากค่าดรรชนีใน M1



รูปที่ 6.2.8 ผังของซับรูทีน "INTRPT"

ชั้นรูทีน "INTRPT"

```
INTRPT  DI
        PUSH    PSW
        PUSH    B
        PUSH    D
        LDA     M1
        MOV     C,A
        PUSH    H
        LHLD    M2L
        XCHG
        POP     H
        MVI     A,OEH
        CMP     D
        JC      A1
        MVI     A,OD1H
        CMP     E
        JC      A1
        IN      4OH
        CPI     2
        JNC     LLL
KKK     MVI     A,O
        CMP     C
        JZ      A1
KKKK    DCK     D
        STAX    D
        DCR     C
        JMP     KKK
LLL     DCR     A
        MOV     B,A
        MVI     A,O
        CMP     C
        JNZ     MMM
        MOV     A,B
        CPI     1
        JZ      NNN
        CPI     8
        JZ      NNN
        CPI     9
        JNZ     OOO
NNN     MOV     A,B
        STAX    D
        INX     D
        INR     C
        OUT     2OH
        JMP     A3
OOO     CPI     4
        JC      A1
        CPI     8
```

```
        JNC     A1
        STAX    D
        INX     D
        INR     C
RRR     INX     D
        INR     C
TTT     INX     D
        INR     C
        OUT     40H
        JMP     A3
MMM     MVI     A,1
        CMP     C
        JNZ     PPP
        DCX     D
        LDAX    D
        INX     D
        CPI     8
        JC      QQQ
        CPI     0AH
        JNC     QQQ
A2      MOV     A,B
        STAX    D
        JMP     RRR
QQQ     MOV     A,B
        CPI     7
        JNC     SSS
        JMP     A2
SSS     CPI     9
        JNC     UUU
        JMP     NNN
UUU     DCX     D
        MVI     A,0
        STAX    D
        JMP     VVV
PPP     MVI     A,2
        CMP     C
        JNZ     WWW
ZZZ     MOV     A,B
        STAX    D
        JMP     TTT
WWW     MVI     A,0AH
        CMP     B
        JNZ     XYZ
        MVI     B,0
        MVI     A,03
        CMP     C
        JNZ     Z1
        MVI     A,02
        CMP     B
        JC      KKKK
        JMP     NNN
Z1      MVI     A,04
        CMP     C
```

```
        JNZ     Z2
        DCX     D
        LDAX    D
        INX     D
        CPI     02
        JNZ     ZZZ
        MVI     A,03
        CMP     B
        JC      KKKK
        JMP     ZZZ
Z2      MVI     A,05
        CMP     C
        JNZ     Z4
        MVI     A,05
        CMP     B
        JC      KKKK
        JMP     NNN
Z4      MOV     A,B
        STAX    D
        INX     D
VVV     MVI     C,0
A1      OUT     60H
A3      MOV     A,C
        STA     M1
        PUSH    H
        XCHG
        SHLD    M2L
        POP     H
        POP     D
        POP     B
        POP     PSW
        EI
        RET
```