



บทที่ 4

การทำงานของโปรแกรมระบบในการติดต่อกับคีย์บอร์ด, จอภาพและเครื่องพิมพ์
การทำงานของคีย์บอร์ด

คีย์บอร์ดของเครื่อง IBM เอ็มพีซีเป็นคีย์บอร์ด 83 แป้น สามารถแบ่งออกได้เป็น 3 ชุด คือ แป้นอักษรหลัก 58 แป้น แป้นตัวเลข 15 แป้น และฟังก์ชันคีย์ 10 แป้น สามารถกำหนดรหัสคีย์บอร์ดได้ 256 รหัส ดังนี้

1. แป้นอักษรหลัก มี 58 แป้น แบ่งออกเป็นแป้นอักษร 48 แป้น แป้นควบคุม 10 แป้น

1.1 แป้นอักษร สามารถกำหนดรหัสคีย์บอร์ดได้ 95 ตัว และรหัสควบคุม 26 ตัว โดยแบ่งออกเป็น 3 ระดับคือ ระดับปกติ กำหนดรหัสได้ 48 ตัวได้แก่รหัสของคีย์บอร์ดเล็ก 26 ตัว ตัวเลข 10 ตัวและคีย์บอร์ดพิเศษอีก 12 ตัว ระดับที่สอง กำหนดรหัสคีย์บอร์ดได้ 47 ตัวคือ อักษรใหญ่ 26 ตัวคีย์บอร์ดพิเศษ 21 ตัว ระดับที่สาม กำหนดรหัสควบคุมได้ 26 ตัว

1.2 แป้นควบคุม มีทั้งหมด 10 แป้น ดังนี้

1.2.1 แป้นอีเอสซี (ESC Key) จะยกเลิกข้อความในบรรทัดปัจจุบันและเลื่อนไปบรรทัดใหม่ โดยจะแสดงเครื่องหมาย "\ " บนบรรทัดที่ยกเลิก

1.2.2 แป้นแบคสเปซ (Back Space) จะเลื่อนเคอร์เซอร์ย้อนกลับมา 1 คอลัมน์ พร้อมทั้งลบคีย์บอร์ดบนจอภาพ 1 คีย์บอร์ด

1.2.3 แป้นแท็บ (TAB Key) จะเลื่อนเคอร์เซอร์ไปครั้งละ 8 คอลัมน์

1.2.4 แป้นแครีเอจรีเทิร์น (Carriage-Return Key) จะส่งรหัสคีย์บอร์ด

ในคีย์บอร์ดไปให้โปรแกรมหลัก

1.2.5 แป้นซีทีอาร์แอล (CTRL Key) กำหนดให้ระดับคีย์บอร์ดเป็นระดับที่ 3 เมื่อคลิกกับแป้นอักษร A-Z แป้นใดแป้นหนึ่ง จะให้กำหนดรหัสคอนโทรลตัวใดตัวหนึ่งใน 26 ตัว

1.2.6 แป้นยกแคร่ มี 2 แป้น จะทำหน้าที่ยกระดับตัวอักษรจากระดับปกติเป็นระดับที่ 2 เมื่อกดคู่กับแป้นอักษรเป็นใดแป้นหนึ่งจะให้กำเนิดอักษรใหญ่ตัวใดตัวหนึ่งใน 26 ตัวหรือตัวอักษรพิเศษตัวใดตัวหนึ่งใน 21 ตัว

1.2.7 แป้นเอแอลที (ALT Key) จะให้กำเนิดรหัสตัวอักษรตัวใดตัวหนึ่งใน 256 ตัวโดยจะต้องกดพร้อมกับตัวเลขของแป้นตัวเลข

1.2.8 แป้นแคปล็อก (Caps Lock Key) จะล๊อคให้คีย์บอร์ดอยู่ในระดับที่ 2 โดยตลอดจนกว่าจะมีการกดแป้นอีกครั้ง เพื่อประโยชน์ในการใช้รหัสตัวอักษรในระดับที่สองอย่างต่อเนื่อง

1.2.9 แป้นพีอาร์ทีเอสซีอาร์ (Prtscr Key) ในระดับปกติจะกำเนิดรหัส "*" แต่เมื่อใช้คู่กับแป้นยกแคร่จะพิมพ์ทุกตัวอักษรที่ปรากฏบนจอภาพลงเครื่องพิมพ์

2. แป้นตัวเลข มี 15 แป้นแบ่งออกได้เป็น 2 ภาค คือภาคแป้นตัวเลขและภาคแป้นควบคุมเคอร์เซอร์ การเปลี่ยนภาคสามารถกระทำได้โดยการกดแป้นแฉล้มล็อก

2.1 ภาคแป้นตัวเลข ได้แก่ตัวเลข 0-9 จุดทศนิยม เครื่องหมาย "+", เครื่องหมาย "-"

2.2 ภาคควบคุมเคอร์เซอร์ จะใช้ได้เฉพาะในบางโปรแกรมเท่านั้นมี 10 แป้นดังนี้

2.2.1 แป้นโฮม (Home Key) จะเลื่อนเคอร์เซอร์ไปที่บรรทัดแรก คอลัมน์แรกของจอภาพ

2.2.2 แป้น ↑ จะเลื่อนเคอร์เซอร์ขึ้นข้างบน 1 บรรทัด โดยยังคงอยู่คอลัมน์เดิม

2.2.3 แป้นพีจียูพี (PgUp Key) จะเลื่อนเคอร์เซอร์ขึ้นข้างบน 1 หน้า

2.2.4 แป้น ← จะเลื่อนเคอร์เซอร์ไปทางซ้าย 1 คอลัมน์ โดยยังคงอยู่

บรรทัดเดิม

2.2.5 แป้น --> จะเลื่อนเคอร์เซอร์ไปทางขวา 1 คอลัมน์ โดยยังอยู่บรรทัดเดิม

2.2.6 แป้นอีเอ็นดี (End Key) จะเลื่อนเคอร์เซอร์ไปที่บรรทัดสุดท้าย คอลัมน์สุดท้ายของจอภาพ

2.2.7 แป้น ↓ จะเลื่อนเคอร์เซอร์ลงล่าง 1 บรรทัด โดยยังอยู่คอลัมน์เดิม

2.2.8 แป้นพีจีเอ็น (PgDn Key) จะเลื่อนเคอร์เซอร์ลงมา 1 หน้า

2.2.9 แป้นไอเอ็นเอส (Ins Key) จะแทรกหีสตัวอักษรลง ณ.ตำแหน่งที่เคอร์เซอร์อยู่

2.2.10 แป้นดีแอล (Del Key) จะตัดหีสตัวอักษร ณ.ตำแหน่งที่เคอร์เซอร์อยู่ทิ้ง

3. แป้นฟังก์ชันคีย์ มีทั้งหมด 10 แป้นคือ F1-F10 สร้างไว้ให้ผู้เขียนโปรแกรมใช้ในการควบคุมการทำงานบางอย่างของโปรแกรม สำหรับในภาคของดอส จะใช้ฟังก์ชันคีย์ดังนี้

3.1 แป้น F1 จะดึงหีสซึ่งเก็บไว้ในคีย์บอร์ดบัฟเฟอร์มา 1 ตัว

3.2 แป้น F2 จะดึงหีสตัวอักษรจากคีย์บอร์ดบัฟเฟอร์มาจนถึงตัวที่กำหนด

3.3 แป้น F3 จะดึงหีสตัวอักษรจากคีย์บอร์ดบัฟเฟอร์ทั้งหมดมาแสดงบนจอภาพ

3.4 แป้น F4 จะข้ามไม่ดึงตัวอักษรที่กำหนดจากคีย์บอร์ดบัฟเฟอร์มาแสดงบนจอภาพ

โดยทำตรงข้ามกับแป้น F2

กลุ่มของแป้นตัวอักษรทำหน้าที่ควบคุมการทำงานของระบบ มี 5 กลุ่ม ดังนี้

1. แป้นซีทีอาร์แอลกับแป้นเบรค (Break Key) เมื่อกดพร้อมกันจะยกเลิกการทำงาน
ของระบบในขณะนั้น

2. แป้นซีทีอาร์แอลกับแป้นเอ็มล๊อค เมื่อกดพร้อมกันจะหยุดการทำงานของระบบใน

ขณะนั้นลงชั่วขณะ จนกว่าจะมีการกดแป้นใด ๆ บนแป้นตัวอักษร

3. แป้นขี้อาร์แอลกับแป้นพีอาร์ทีเอสซี เมื่อกดพร้อมกันจะทำให้ทุก ๆ รหัสตัวอักษรที่กดและปรากฏบนจอภาพหลังจากที่กดแป้นทั้งสองแล้ว พิมพ์บนเครื่องพิมพ์ด้วยจนกว่าจะมีการกดแป้นทั้งสองอีกครั้ง

4. แป้นยกแคร่กับแป้นพีอาร์ทีเอสซี จะนำรหัสตัวอักษรที่ปรากฏบนจอภาพในขณะนั้นทุกตัวพิมพ์ลงบนเครื่องพิมพ์

5. แป้นขี้อาร์แอลกับแป้นเอแอลทีกับแป้นคีโอแอล เมื่อกดพร้อมกันทั้ง 3 ตัว จะเซตระบบขึ้นมาใหม่

สแกนโค้ด และ รหัสตัวอักษร

ในการกดแป้นตัวอักษรแต่ละครั้งนอกจากได้รหัสตัวอักษรแล้ว จะได้สแกนโค้ดที่สมมูลกับแป้นอักษรนั้นด้วย ค่าของสแกนโค้ดจะมีทั้งหมด 83 ค่า เท่ากับจำนวนแป้นตัวอักษร ซึ่งสามารถแยกเป็นกลุ่มได้ดังนี้

1. กลุ่มของแป้นฟังก์ชัน

แป้นตัวอักษร	สแกนโค้ด	รหัสอักษร
F1	59	0
F2	60	0
F3	61	0
F4	62	0
F5	63	0
F6	64	0
F7	65	0
F8	66	0
F9	67	0
F10	68	0



ตารางที่ 4.1 สแกนโค้ดและรหัสตัวอักษร

2. กลุ่มของแป้นอักษรหลัก

แป้นตัวอักษร	สแกนโค้ด	รหัสอักษร ระดับที่ 1	รหัสอักษร ระดับที่ 2
1	2	49	33
2	3	50	64
3	4	51	35
4	5	52	36
5	6	53	37
6	7	54	94
7	8	55	38
8	9	56	42
9	10	57	40
0	11	48	41
-	12	45	95
=	13	61	43
Q	16	113	81
W	17	119	87
E	18	101	69
R	19	114	82
T	20	116	84
Y	21	121	89
U	22	117	85
I	23	105	73
O	24	111	79
P	25	112	80
{	26	91	123
}	27	93	125
A	30	97	65
S	31	115	83
D	32	100	68
F	33	102	70
G	34	103	71

ตารางที่ 4.1 (ต่อ)

แป้นตัวอักษร	สแกนโค้ด	รหัสอักษร	
		ระดับที่ 1	ระดับที่ 2
H	35	104	72
J	36	106	74
K	37	107	75
L	38	108	76
;	39	59	58
'	40	39	34
‘	41	96	126
\	43	92	124
Z	44	122	90
X	45	120	88
C	46	99	67
V	47	118	86
B	48	98	66
N	49	110	78
M	50	109	77
,	51	44	60
.	52	46	62
/	53	47	63
Prtsc	55	42	
SPACE	57	32	

3. กลุ่มของแป้นตัวเลข

แป้นตัวอักษร	สแกนโค้ด	รหัสตัวอักษร	
		ภาคตัวเลข	ภาคเคอร์เซอร์
7	71	55	0
8	72	56	0
9	73	57	0
-	74	45	45
4	75	52	0
5	76	53	0
6	77	54	0
+	78	43	43
1	79	49	0
2	80	50	0

แป้นตัวอักษร	สแกนโค้ด	รหัสตัวอักษร	รหัสตัวอักษร
		ภาคตัวเลข	ภาคเคอร์เซอร์
3	81	51	0
4	82	52	
.	83	46	

4. กลุ่มของแป้นควบคุม

แป้นตัวอักษร	สแกนโค้ด	รหัสอักษร
Esc	1	27
Back Space	14	8
Num Lock	69	0
Scroll Lock	70	0
Tap	15	0
Enter	28	10
Ctrl	29	27
Left Shift	42	
Right Shift	54	
Alt	56	
Caps Lock	58	

ตารางที่ 4.1 (ต่อ)

โปรแกรมระบบที่ควบคุมการทำงานของคีย์บอร์ด

มีโปรแกรมระบบอยู่ 2 ส่วนที่เกี่ยวข้อง คือ

1. โปรแกรมคีย์บอร์ดอินเตอร์รัฟ ใช้อินเตอร์รัฟ 9H

เมื่อมีการกดแป้นอักษรบนคีย์บอร์ด จะเกิดอินเตอร์รัฟไปยังซีพียู 8259 โปรแกรมคีย์บอร์ดอินเตอร์รัฟจะเริ่มทำงาน โดยการอ่านค่าสแกนโค้ดจากพอร์ท PA ของซีพียู 8255 ดังส่วนของโปรแกรม

IN AL, 60H ; อ่านค่าจากพอร์ท PA ของ 8255
 PUSH AX ; AX - ค่าสแกนโค้ด


```

IN      AL,61H    ; อ่านค่าจากพอร์ท PB
OR      AL,80H    ; เขตสัญญาณแอกโทนเลตจ
OUT     61H,AL    ; ส่งกลับไปยังพอร์ท PB
AND     AL,7FH    ; รีเซตค่าสัญญาณแอกโทนเลตจ
OUT     61H,AL    ;
POP     AX        ; AX - ค่าสแกนโค้ด

```

หลังจากนั้นจะทำการแปลงค่าสแกนโค้ดที่ได้ เป็นรหัสตัวอักษรที่สมนัยกัน หลังจากได้รหัสตัวอักษรแล้ว ค่าสแกนโค้ดและรหัสตัวอักษรจะถูกเก็บไว้ในคีย์บอร์ดบัฟเฟอร์ขนาด 16 เวิร์ต เพื่อให้โปรแกรมคีย์บอร์ด ไอโอ นำไปใช้งานต่อไป

```

MOV     BX,BUFFER_TAIL ; ตำแหน่งของบัฟเฟอร์ที่จะเก็บ
MOV     SI,BX          ; ตัวอักษร
ADD     BX,2           ; เขต ไปยังเวิร์คถัดไป
CMP     BX,OFFSET BUFFER_END ; เป็นตำแหน่งสุดท้ายของบัฟเฟอร์
JNE     K1            ; ไม่ใช่
MOV     BX,OFFSET KB_BUFFER ; เขต ไปยังตำแหน่งเริ่มต้น
K1:    CMP     BX,BUFFER_HEAD ; เป็นตำแหน่งเดียวกับตำแหน่งตัว
        ; อักษรที่จะส่ง
JE      K2            ; ใช่ เสียงเตือน
MOV     [SI],AX       ; เก็บค่าในบัฟเฟอร์
MOV     BUFFER_TAIL,BX ; เก็บตำแหน่งปัจจุบันไว้
JMP     K3
K2:    CALL   ERROR_BEEP
K3:    :
        :

```

2. โปรแกรมคีย์บอร์ด ไอโอ ใช้อินเตอร์วัพ 16H

มีทั้งหมด 3 ฟังก์ชันดังนี้

- เมื่อ AH = 0 จะส่งค่าสแกนโค้ดและรหัสตัวอักษรที่เก็บไว้ในคีย์บอร์ดบัฟเฟอร์ ไปให้โปรแกรมหลัก โดยค่าสแกนโค้ดจะเก็บไว้ใน AH รหัสตัวอักษรจะเก็บไว้ใน AL
- เมื่อ AH = 1 เขต ZF แพล็ก โดย ZF แพล็กมีความหมายดังนี้

ZF = 1 ไม่มีรหัสตัวอักษรในคีย์บอร์ดบัฟเฟอร์

ZF = 0 มีรหัสตัวอักษรในคีย์บอร์ดบัฟเฟอร์พร้อมที่จะนำมาใช้ได้

3. เมื่อ AH = 2 บอกสถานะของการยกแคร่ โดยส่งผลมายัง AL

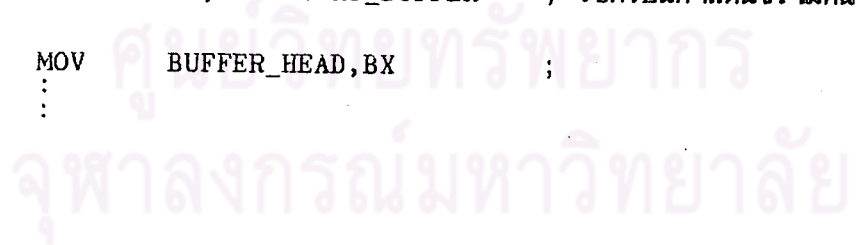
โปรแกรมคีย์บอร์ดไอโอ ฟังก์ชัน AH = 0

จะส่งค่าสแกนโค้ดและรหัสตัวอักษรในคีย์บอร์ดบัฟเฟอร์กลับไปให้โปรแกรมหลัก การที่

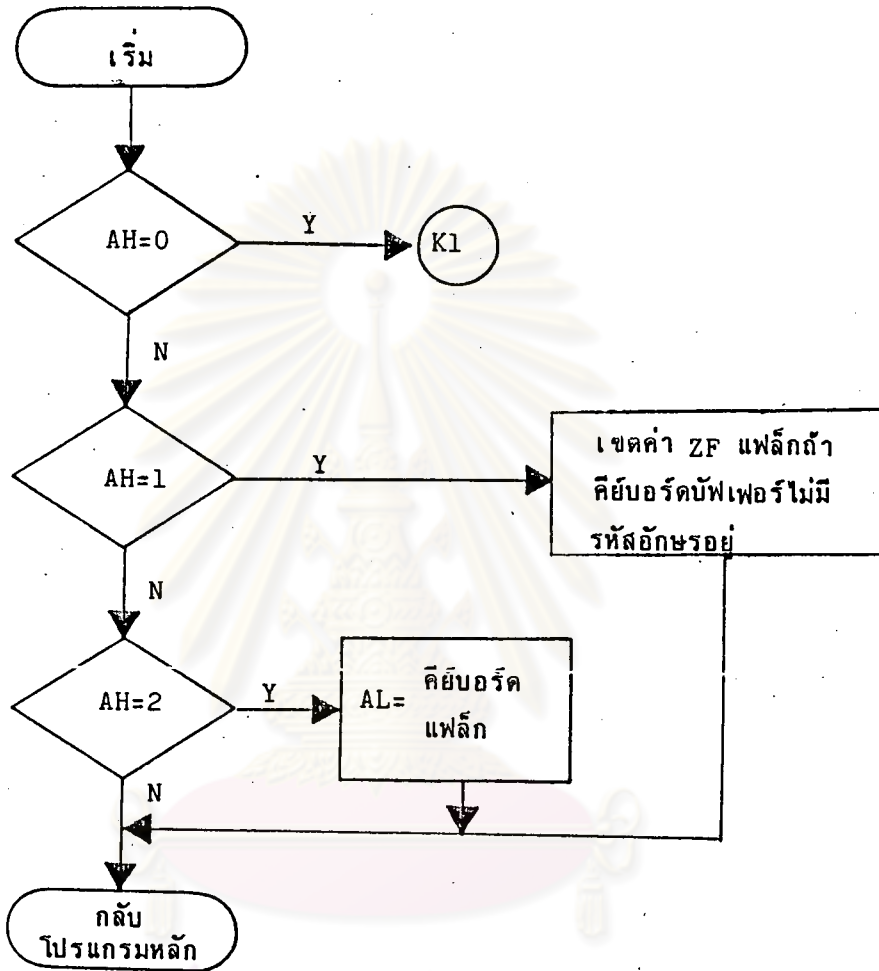
ไม่มีค่าในคีย์บอร์ดบัฟเฟอร์จะรอจนกระทั่งมีการกดแป้นอักษรบนคีย์บอร์ด สามารถแสดงได้ดังนี้

```

K1:      MOV     BX,BUFFER_HEAD      ; ตำแหน่งของตัวอักษรที่จะส่ง
          CMP     BX,BUFFER_TAIL    ; บัฟเฟอร์ว่าง
          JZ      K1                ; ใช่
          MOV     AX,[BX]           ; เก็บรหัสอักษรและสแกนโค้ด
          ADD     BX,2              ; เปลี่ยนตำแหน่งที่จะส่งตัวอักษร
                                   ; ไปอีก 1 เวิร์ด
          CMP     BX,OFFSET KB_BUFFER_END; เป็นตำแหน่งสุดท้ายของบัฟเฟอร์
          JNE     K2                ;
          MOV     BX,OFFSET KB_BUFFER ; เซตเป็นตำแหน่งเริ่มต้น
K2:      MOV     BUFFER_HEAD,BX
          ...
  
```

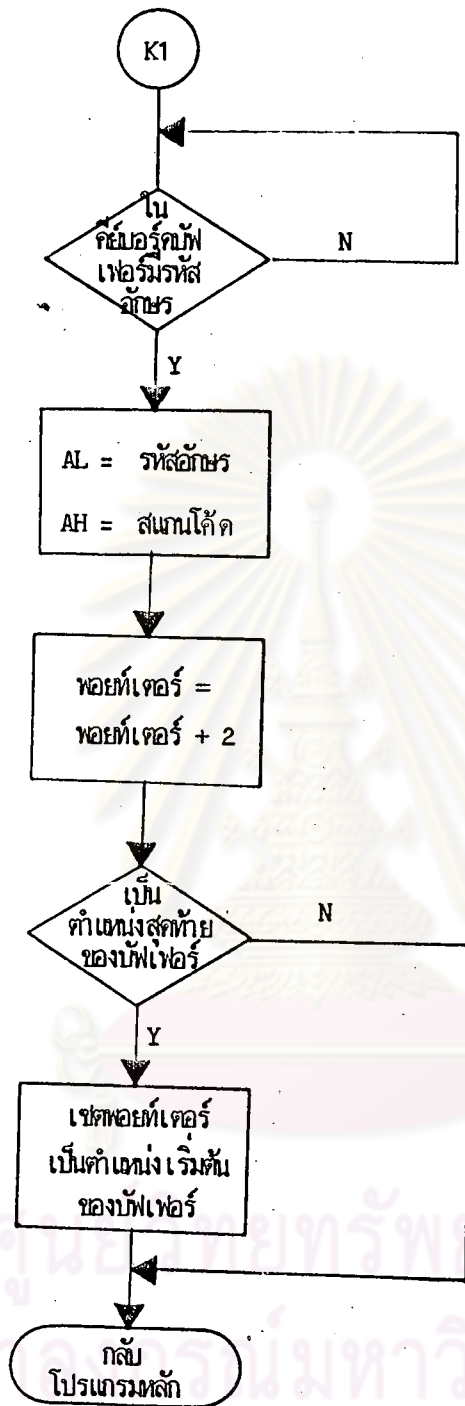


โปรแกรมคีย์บอร์ดไอโอ



รูปที่ 4.1 การทำงานของโปรแกรมคีย์บอร์ดไอโอ

ศูนย์วิจัยคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 (ต่อ)



การทำงานของจอภาพ

มีส่วนที่เกี่ยวข้องดังนี้

โมโนโครมดิสเพลย์อะแดปเตอร์ (Monochrome Display Adapter)

มีคุณสมบัติดังนี้

1. สามารถแสดงผลตัวอักษรบนจอภาพได้ 2000 ตัวอักษรหรือ 25 บรรทัด บรรทัดละ 80 ตัวอักษร ตัวอักษรมีขนาด 7X9 บนพื้นที่ 9X14
2. มีไมโครโวลตา 6845 ซีอาร์ทีคอนโทรลเลอร์ (Motorola 6845 CRT Controller) เป็นตัวควบคุมการสแกนตัวอักษรบนจอภาพ
3. มีแรมขนาด 4 กิโลไบต์เป็นบัฟเฟอร์ของจอภาพ
4. มีรอมขนาด 8 กิโลไบต์เป็นที่เก็บฟอนต์ตัวอักษร (Character font) จำนวน 256 ตัวเท่ากับจำนวนรหัสตัวอักษร

6845 ซีอาร์ทีคอนโทรลเลอร์

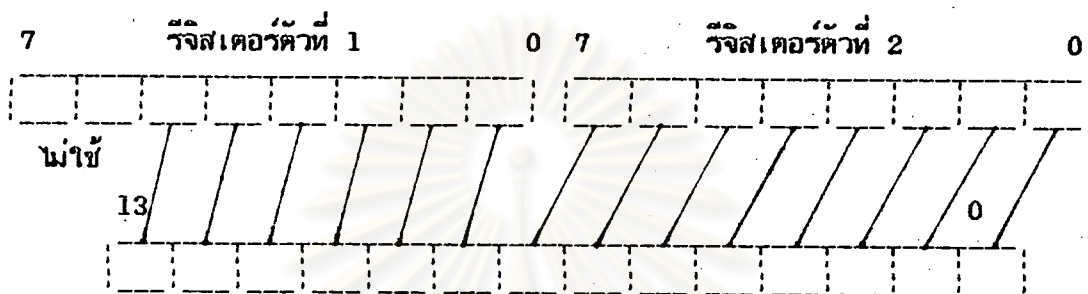
6845 ซีอาร์ทีคอนโทรลเลอร์ มีรีจิสเตอร์สำหรับใช้งานทั้งหมด 18 ตัว ดังนี้

รีจิสเตอร์

หน้าที่

- | | |
|---------|---|
| 0 - 3 | บอกลักษณะของตัวอักษรในแนวราบ จำนวนตัวอักษรต่อ 1 บรรทัด
ความกว้างของแต่ละตัวอักษร |
| 4 - 9 | บอกลักษณะของตัวอักษรในแนวดิ่ง จำนวนตัวอักษรต่อ 1 คอลัมน์
จำนวนแถวที่จะสแกนคือ 1 ตัวอักษร |
| 10 | แถวสแกนแถวแรกของเคอร์เซอร์ |
| 11 | แถวสแกนแถวสุดท้ายของเคอร์เซอร์ |
| 12 - 13 | ตำแหน่งของจอภาพที่จะปรากฏตัวอักษร |
| 14 - 15 | ตำแหน่งของจอภาพที่จะปรากฏเคอร์เซอร์ |
| 16 - 18 | ไม่ใช้ |

6845 สามารถอ้างตำแหน่งของจอภาพได้สูงสุด 16 กิโลไบต์ โดยวิธีการใช้รีจิสเตอร์ 2 ตัวร่วมกันเพื่อทำเป็น 14 บิต สำหรับจอภาพแบบโมโนโครม ตำแหน่งที่จะแสดงตัวอักษรหรือเคอร์เซอร์จะกระทำโดยตั้งแต่ตำแหน่งที่ 0 ถึงตำแหน่งที่ 1999



รูปที่ 4.2 การคำนวณหาตำแหน่งบนจอภาพ

ในการโปรแกรม 6845 ซีพียูคอนโทรลเลอร์ ชั้นแรกจะต้องเลือกหมายเลขของรีจิสเตอร์ที่ต้องใช้ โดยการส่งค่าหมายเลขของรีจิสเตอร์ในเมมโมรี่พอร์ท 3B4H แล้วจึงส่งค่าที่ต้องการเซตไปยังพอร์ท 3B5H

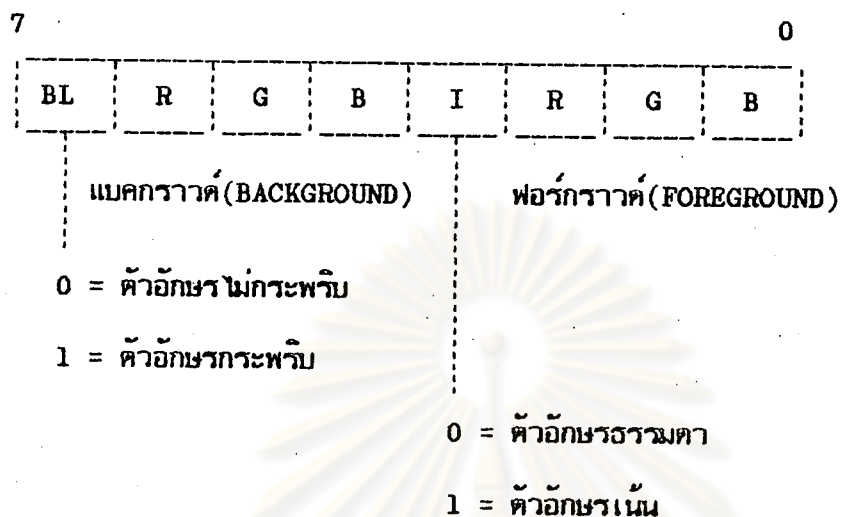
ตัวอักษร (Character)

จากรหัสตัวอักษร 256 ตัว สามารถเปลี่ยนเป็นตัวอักษรในพอนท์ตัวอักษรที่สมนัยกัน นอกจากรหัสตัวอักษรบางตัวที่ใช้ในการควบคุมการแสดงผลของจอภาพ รหัสตัวอักษรเหล่านี้ได้แก่

1. รหัส 07 ใช้ในการสร้างเสียงเตือน
2. รหัส 08 กอຍหลังเคอร์เซอร์และลบตัวอักษรบนจอภาพ ณ.ตำแหน่งนั้น 1 ตัว
3. รหัส 10 เลื่อนตำแหน่งเคอร์เซอร์ลงมา 1 บรรทัด
4. รหัส 13 เลื่อนตำแหน่งของเคอร์เซอร์ให้กลับไปขึ้นต้นบรรทัดนั้น

แอททริบิวต์ (Attribute)

จะเป็นตัวกำหนดลักษณะของตัวอักษรที่จะแสดงบนจอภาพ มีขนาด 1 ไบต์ มีความหมาย



ฟอร์กราวด์ R G B	แบกราวด์ R G B	ความหมาย
0 0 0	0 0 0	ไม่มีตัวอักษรปรากฏ
1 1 1	1 1 1	ตำแหน่งของตัวอักษรเป็นพื้นขาว
0 0 0	1 1 1	ตำแหน่งของตัวอักษรเป็นพื้นปกติ
0 0 0	0 0 1	ขีดเส้นใต้ตรงตัวอักษร
1 1 1	0 0 0	ตัวอักษรถูกแสดงในลักษณะรีเวอร์ส วิดีโอ (REVERSE VIDEO)

รูปที่ 4.3 ลักษณะของแอททริบิวต์

การแสดงตัวอักษรบนจอภาพ

ในการแสดงผลบนจอภาพ ตำแหน่งของจอภาพที่เป็นเลขคู่จะเป็นตำแหน่งของตัวอักษร ส่วนตำแหน่งที่เป็นเลขคี่จะเป็นตำแหน่งของแอดเดรส ตำแหน่งของจอภาพแบบโมโนโครมจะสามารถอ้างถึงได้ตั้งแต่ตำแหน่งของหน่วยความจำที่ B0000H จนถึง B0F9FH มีขนาด 4000 ไบต์ หรือ 1 หน้าสำหรับจอภาพแบบกราฟิก หน่วยความจำที่ใช้เป็นบัฟเฟอร์ของจอภาพเริ่มตั้งแต่ตำแหน่งที่ B8000H มีขนาด 16 กิโลไบต์แบ่งเป็น 4 หน้า หน้าละ 4 กิโลไบต์ เริ่มตั้งแต่หน้าที่ 0 ถึงหน้าที่ 3

โปรแกรมระบบที่ควบคุมการทำงานของจอภาพ

โปรแกรมวิดีโอไอโอ ใช้อินเตอร์รัพ 10H

มีทั้งหมด 16 ฟังก์ชัน ดังนี้

1. เมื่อ AH = 0 กำหนดลักษณะของจอภาพ โดยที่ถ้า

AL = 0 จอภาพขาวดำ แบบ 40X25

AL = 1 จอภาพสี แบบ 40X25

AL = 2 จอภาพขาวดำ แบบ 80X25

AL = 3 จอภาพสี แบบ 80X25

AL = 4 จอภาพกราฟิกสี แบบ 320X200

AL = 5 จอภาพกราฟิกขาวดำ แบบ 320X200

AL = 6 จอภาพกราฟิกขาวดำ แบบ 640X200

2. เมื่อ AH = 1 กำหนดลักษณะของเคอร์เซอร์ โดยที่

CH = ตำแหน่งแถวเริ่มต้นสแกนของเคอร์เซอร์

CL = ตำแหน่งแถวสุดท้ายของเคอร์เซอร์

3. เมื่อ AH = 2 กำหนดตำแหน่งของเคอร์เซอร์ โดยที่

DH = ตำแหน่งแถว



DL = ตำแหน่งคอลัมน์

BH = หน้าของจอภาพ

4. เมื่อ AH = 3 หาค่าแห่งของเคอร์เซอร์ โดยที่

BH = หน้าของจอภาพ

ค่าของตำแหน่งเคอร์เซอร์ที่ได้จะอยู่ใน DX โดย

DH = ตำแหน่งแถว

DL = ตำแหน่งคอลัมน์

5. เมื่อ AH = 4 หาค่าแห่งของไลท์เพน

6. เมื่อ AH = 5 เขตหน้าของจอภาพที่จะใช้

7. เมื่อ AH = 6 สโกกรภาพขึ้น 1 บรรทัด โดยที่

AL = จำนวนแถวที่จะถูกลบทิ้ง

CH, CL = ตำแหน่งของแถวและคอลัมน์ ณ จุดบนด้านซ้ายสุดที่จะสโกกร

DH, DL = ตำแหน่งของแถวและคอลัมน์ ณ จุดกลางด้านขวาสุดที่จะสโกกร

BH = แอทริบิวที่จะใช้กับบรรทัดที่จะลบทิ้ง

8. เมื่อ AH = 7 สโกกรภาพลง 1 บรรทัด โดยที่

BL = จำนวนแถวที่จะถูกลบทิ้ง

CH, CL = ตำแหน่งของแถวและคอลัมน์ ณ จุดบนด้านซ้ายสุดที่จะสโกกร

DH, DL = ตำแหน่งของแถวและคอลัมน์ ณ จุดบนด้านขวาสุดที่จะสโกกร

BH = แอทริบิวที่จะใช้กับบรรทัดที่จะลบทิ้ง

9. เมื่อ AH = 8 อ่านค่าแอทริบิวและรหัสตัวอักษร ณ ตำแหน่งของเคอร์เซอร์ โดยที่

BH = หน้าของจอภาพ

ค่าที่ได้รับจะเป็นดังนี้

AL = รหัสตัวอักษร

AH = แอทริบิว

10. เมื่อ AH = 9 แสดงตัวอักษรและแอทริบิว ณ.ตำแหน่งของเคอร์เซอร์

BH = หน้าของจอภาพ

BL = ค่าแอทริบิวของตัวอักษรที่จะแสดง

CX = จำนวนของตัวอักษรที่จะแสดง

AL = รหัสตัวอักษรที่จะแสดง

11. เมื่อ AH = 10 แสดงเฉพาะตัวอักษร ณ.ตำแหน่งของเคอร์เซอร์

BH = หน้าของจอภาพ

CX = จำนวนของตัวอักษรที่จะแสดง

AL = รหัสตัวอักษรที่จะแสดง

12. เมื่อ AH = 11 กำหนดสีสำหรับกราฟฟิก

13. เมื่อ AH = 12 เขียนจุดบนจอแบบกราฟฟิก

DX = ตำแหน่งแถว

CX = ตำแหน่งคอลัมน์

AL = สีที่ใช้

14. เมื่อ AH = 13 อ่านจุดบนจอแบบกราฟฟิก

DX = ตำแหน่งแถว

CX = ตำแหน่งคอลัมน์

AL = จุดที่อ่านได้

15. เมื่อ AH = 14 แสดงตัวอักษรบนจอภาพและเลื่อนเคอร์เซอร์ไป 1 คอลัมน์

AL = รหัสตัวอักษรที่จะแสดง

BH = หน้าของจอภาพ

16. เมื่อ AH = 15 ตรวจสอบสถานะของจอภาพ

ค่าที่ได้จะเป็นดังนี้

AL = ลักษณะของจอภาพที่ใช้

AH = จำนวนของตัวอักษรต่อคอลัมน์

BH = หน้าของจอภาพ

โปรแกรมวิดีโอ ไอโอ ฟังก์ชัน AH = 14

ในการวิจัยครั้งนี้จะขอก้าวเฉพาะรายละเอียดของฟังก์ชันที่ 14 ซึ่งเป็นฟังก์ชันที่ทำหน้าที่ในการแสดงตัวอักษรบนจอภาพ และเลื่อนเคอร์เซอร์ไปยังตำแหน่งถัดไป ในการทำงานของฟังก์ชันที่ 14 ตัวอักษรที่ส่งมาในรีจิสเตอร์ AL จะถูกนำไปแสดงบนจอภาพ ณ ตำแหน่งของเคอร์เซอร์ในขณะนั้นหลังจากนั้นเคอร์เซอร์จะเลื่อนไปยังตำแหน่งถัดไป ถ้าเคอร์เซอร์เลื่อนไปจนกระทั่งคอลัมน์สุดท้าย ค่าของคอลัมน์จะถูกกำหนดเป็นศูนย์ ค่าของแถวจะเพิ่มขึ้นอีกหนึ่ง และถ้าค่าของแถวเพิ่มจนกระทั่งถึงแถวที่ 25 เคอร์เซอร์จะถูกกำหนดไปยังแถวสุดท้ายคอลัมน์แรก จอภาพจะสโกรลขึ้นไป 1 บรรทัด ค่าของแอดเดรสไบวจะเป็นค่าเดิม ส่วนของโปรแกรมจะเป็นดังนี้

WRITE_TY	PROC	NEAR	
	PUSH	AX	; เก็บค่าในรีจิสเตอร์
	PUSH	AX	; เก็บค่าของรหัสตัวอักษร
	MOV	AH,3	; อ่านตำแหน่งของเคอร์เซอร์
	INT	10H	; DH - แถว, DL - คอลัมน์
	POP	AX	; ค่าของรหัสตัวอักษรที่เก็บไว้
	CMP	AL,08	; เป็นแบคสเปซ ?
	JE	U8	; แบคสเปซ
	CMP	AL,0DH	; เป็นแครีเอจรีเทอร์น ?

```

JE      U9          ; แครี่เอจรีเทอร์น
CMP     AL,0AH     ; เป็นไลน์ฟีด ?
JE      U10       ; ไลน์ฟีด
CMP     AL,07     ; เป็นรหัสเสียง
JE      U11       ; เสียง

```

แสดงผลตัวอักษรบนจอภาพ

```

MOV     BH,ACTIVEPAGE ; ค่าของหน้าในขณะนั้น
MOV     AH,10        ; ฟังก์ชันแสดงตัวอักษร
MOV     CX,1         ; 1 ตัวอักษรเท่านั้น
INT     10H          ;

```

กำหนดตำแหน่งของเคอร์เซอร์ไปยังตำแหน่งถัดไป

```

INC     DL           ; เพิ่มค่าของคอลัมน์อีกหนึ่ง
CMP     DL,BYTE PTR CRT_COLS; เป็นคอลัมน์สุดท้ายของแถว ?
JNZ     U7           ; ไปกำหนดตำแหน่งเคอร์เซอร์
MOV     DL,0         ; กำหนดค่าคอลัมน์เป็น 0
CMP     DH,24        ; เป็นแถวสุดท้าย ?
JNZ     U6           ; ไม่ใช่ เพิ่มค่าของแถว

```

สโการลจอภาพขึ้น 1 บรรทัด

```

U1:     MOV     AH,2   ; ฟังก์ชันกำหนดตำแหน่งของเคอร์เซอร์
        MOV     BH,0   ; หน้า
        INT     10H
U2:     MOV     AH,8   ; ฟังก์ชันอ่านค่าแอดทริบิวและตัวอักษร

```

```

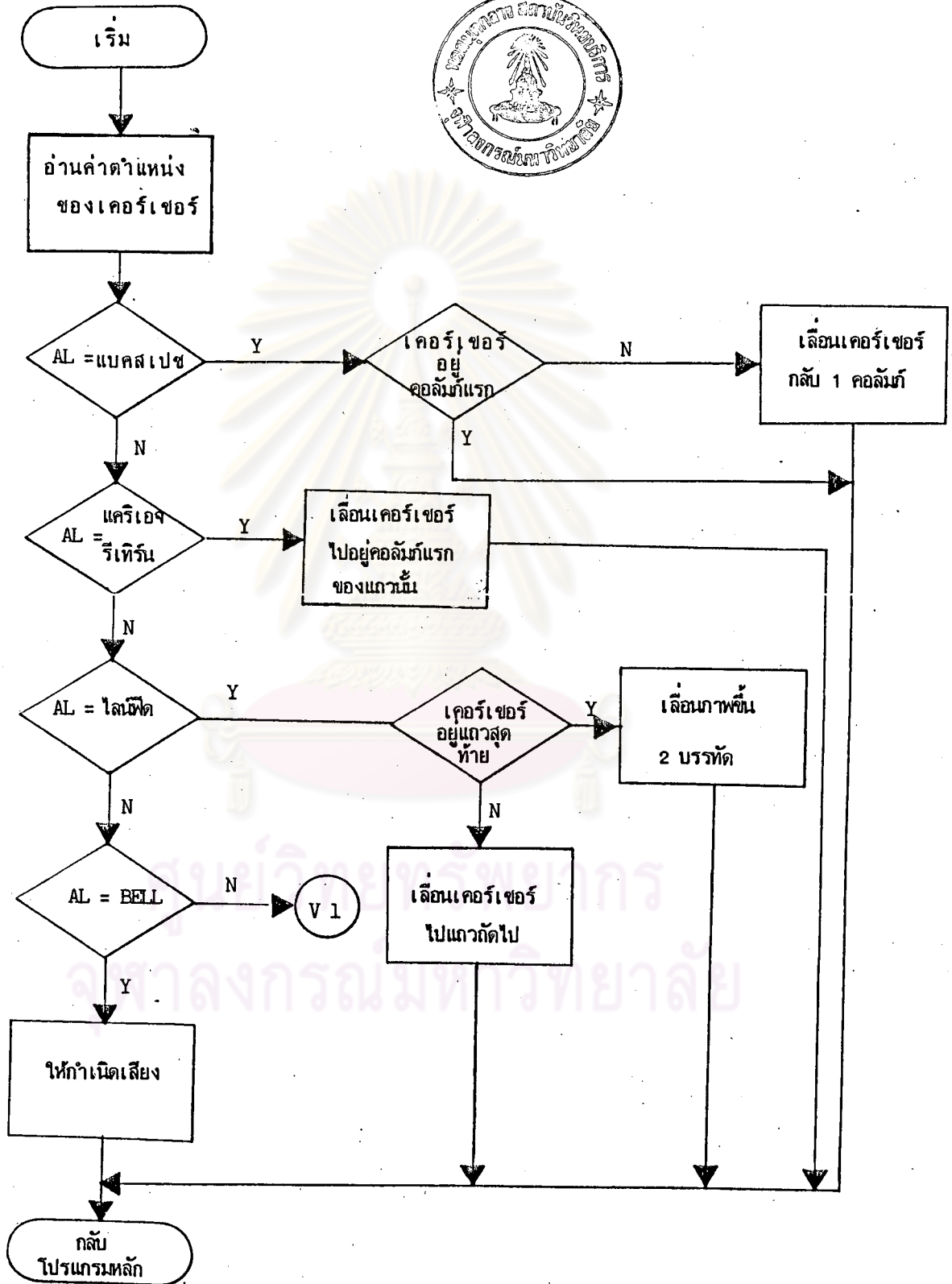
INT      10H      ; AH - แอททริบิว, AL - รหัสตัวอักษร
MOV      BH, AH   ; เก็บค่าแอททริบิวไว้ใน BH
U3:      ; สโกวลจอภาพ
MOV      AH, 6    ; ฟังก์ชันสโกวลจอภาพขึ้น
MOV      AL, 1    ; 1 บรรทัด
MOV      CX, 0    ; ตำแหน่งบนซ้ายที่สโกวล
MOV      DH, 24   ; แถวล่างสุด
MOV      DL, BYTE PTR CRT_COLS ; คอัมม่าขวาสุด
DEC      DL
U4:      INT      10H      ;
U5:      POP      AX      ;
        JMP      VIDEO_RETURN ;
U6:      INC      DH      ; เพิ่มค่าของแถว
U7:      MOV      AH, 2    ; ฟังก์ชันกำหนดตำแหน่งเคอร์เซอร์
        JMP      U4      ; กำหนดตำแหน่ง
; การชี้เป็นแบคสเปซ
U8:      CMP      DL, 0    ; เป็นคอัมม่าแรกสุด ?
        JE       U7      ; ใช่กำหนดตำแหน่งเคอร์เซอร์
        DEC      DL      ; ไม่ใช่ ลดตำแหน่งคอัมม่าลง 1 คอัมม่า
        JMP      U7      ; กำหนดตำแหน่ง
; การชี้เป็นแครริเอจรีเทอาน
U9:      MOV      DL, 0    ; กำหนดเคอร์เซอร์ไปที่คอัมม่าแรก

```

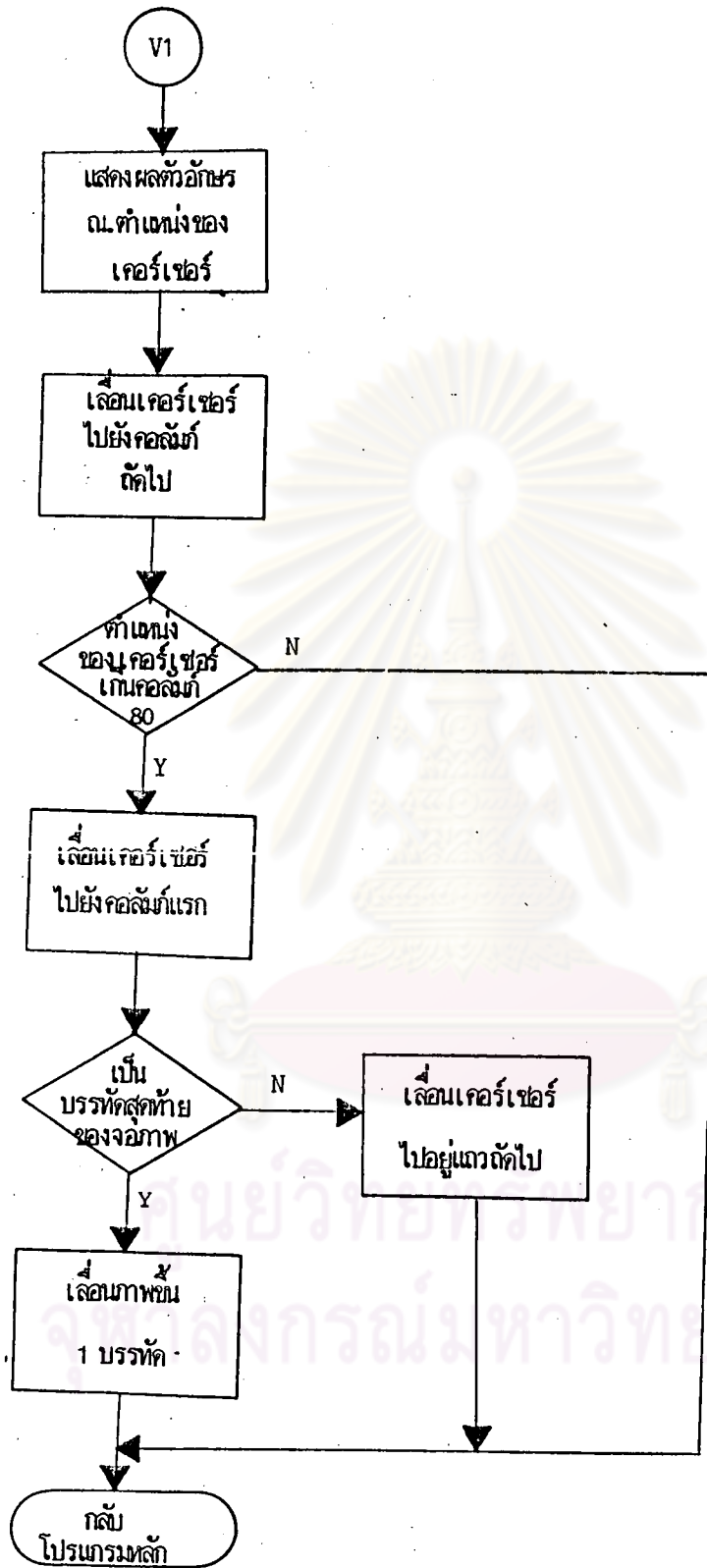


```
JMP          U7          ; กำหนดตำแหน่งเคอร์เซอร์
;
; การพิมพ์เป็นไลน์ฟีด
;
U10:        CMP          DH,24      ; เป็นแถวสุดท้ายของจอภาพ ?
           JNE          U6          ; ไม่ใช่เพิ่มตำแหน่งของแถว
           JMP          U1          ; ใช้สโตร์จอภาพขึ้น 1 บรรทัด
;
; การพิมพ์เป็นรหัสของเสียงเตือน
;
U11:        MOV          BL,2       ; ความถี่
           CALL         BEEP        ; โพรแกรมย่อยในการทำเสียง
           JMP          U5          ; กลับโปรแกรมหหลัก
WRITE_TTY  ENDP
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.4 การทำงานของโปรแกรมวีทีโอไอโอ



รูปที่ 4.4 (ต่อ)

การทำงานของเครื่องพิมพ์

เครื่องพิมพ์ที่ใช้ต่อกับเครื่องไอเอ็มพีสามารถต่อได้ทั้งแบบขนาน (Parallel Interface) และแบบซีเรียล (Serial Interface) แต่ในที่นี้จะขอลำถึงการต่อแบบขนาน เท่านั้น

โมโนโครมคิสเพลย์ปริ้นท์เตอร์อะแดปเตอร์และพาราล์เลลปริ้นท์เตอร์อะแดปเตอร์

(Monochrome Display / Printer Adapter and Parallel Printer Adapter)

เป็นตัวกลางในการติดต่อระหว่างหน่วยประมวลผลกลางกับเครื่องพิมพ์ การติดต่อจะเป็นแบบขนาน ลักษณะของการติดต่อจะเป็นดังตารางที่ 4.3 เครื่องพิมพ์จะพิมพ์ได้ก็ต่อเมื่อ สายซีเล็ก อินพุตถูกกำหนด ทุกครั้งที่เปิดเครื่องคอมพิวเตอร์ จะมีการกำหนดค่าให้กับเครื่องพิมพ์ โดยผ่านทางสายอินพุตของปริ้นท์เตอร์ ในการพิมพ์ตัวอักษรแต่ละครั้งมีขั้นตอนดังนี้ รหัสตัวอักษรที่จะพิมพ์จะถูกส่งมาจากสายคาต้าบิตทั้ง 8 สาย เครื่องพิมพ์จะรับรหัสตัวอักษรมาหาค่าของตัวอักษรในเครื่องพิมพ์ที่สัมพันธ์กันมาพิมพ์ และส่งสัญญาณแอกในเลขจํายังคอมพิวเตอร์ รหัสตัวอักษรตัวถัดไปจะถูกส่งไปยังสายคาต้าบิต การตรวจสอบว่าเครื่องพิมพ์พร้อมที่จะพิมพ์หรือไม่สามารถตรวจได้จากสายบีวี ถ้าสายนี้ ไม่ถูกกำหนดคอมพิวเตอร์ก็สามารถส่งรหัสตัวอักษรไปให้เครื่องพิมพ์ได้

ในการโปรแกรมไปยังเครื่องพิมพ์ สามารถกระทำได้ 3 ฟังก์ชัน ผ่านทางพอร์ตของ 6845 ซีอาร์ทีคอนโทรลเลอร์ ดังนี้

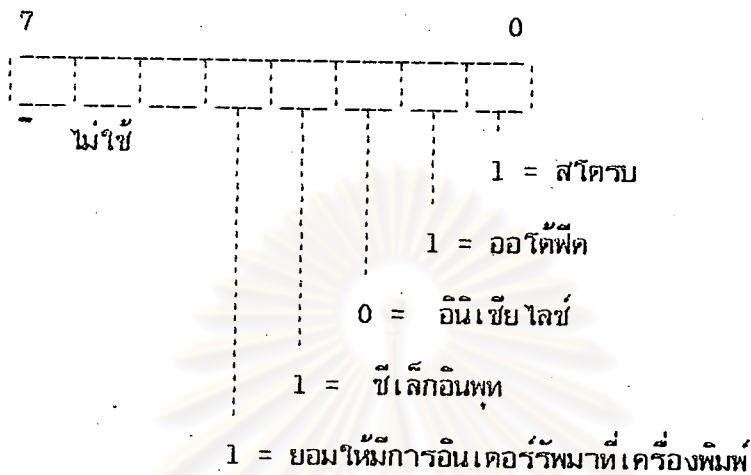
ฟังก์ชัน	โมโนโครมคิสเพลย์ ปริ้นท์เตอร์อะแดปเตอร์	พาราล์เลลปริ้นท์เตอร์ อะแดปเตอร์
1. ส่งรหัสตัวอักษรไปพิมพ์	3BCH	378H
2. ส่งรหัสควบคุม	3BEH	37AH
3. อ่านค่าสถานะของเครื่องพิมพ์	3BDH	379H

ตารางที่ 4.2 ปริ้นท์เตอร์อะแดปเตอร์พอร์ทัล

สัญญาณ	ปลายทาง	พิน
คาตัมบิท 0	เครื่องพิมพ์	2
คาตัมบิท 1	เครื่องพิมพ์	3
คาตัมบิท 2	เครื่องพิมพ์	4
คาตัมบิท 3	เครื่องพิมพ์	5
คาตัมบิท 4	เครื่องพิมพ์	6
คาตัมบิท 5	เครื่องพิมพ์	7
คาตัมบิท 6	เครื่องพิมพ์	8
คาตัมบิท 7	เครื่องพิมพ์	9
สโตรบ (Strobe)	เครื่องพิมพ์	1
ซีเล็กอินพุท (Select Input)	เครื่องพิมพ์	13
ออโตฟีด (Auto Feed)	เครื่องพิมพ์	14
อินิเชียไลซ์ปริ้นเตอร์ (Initialize Printer)	เครื่องพิมพ์	16
แอกโนลเลจ (Acknowledge)	คอมพิวเตอร์	10
บิวซี (Busy)	คอมพิวเตอร์	11
เ้าท์ออฟเปเปอร์ (Out of Paper)	คอมพิวเตอร์	12
ซีเล็ก (Select)	คอมพิวเตอร์	13
เออเรอร์ (Error)	คอมพิวเตอร์	15

ตารางที่ 4.3 สัญญาณการติดต่อแบบขนาน

รหัสควบคุมเครื่องพิมพ์ขนาด 8 บิต มีความหมายดังนี้



รูปที่ 5.4 รหัสควบคุมเครื่องพิมพ์

โปรแกรมระบบที่ควบคุมการทำงานของเครื่องพิมพ์

โปรแกรมปริ้นต์เตอร์ไอโอ ใช้อินเตอร์รัพท์ 17H

มีทั้งหมด 3 ฟังก์ชัน ดังนี้

1. เมื่อ AH = 0 พิมพ์ตัวอักษรบนเครื่องพิมพ์ โดยที่

AL = รหัสตัวอักษรที่จะพิมพ์

ผลลัพธ์ที่ได้รับคือ AH จะบอกถึงผลของการพิมพ์ ถ้า

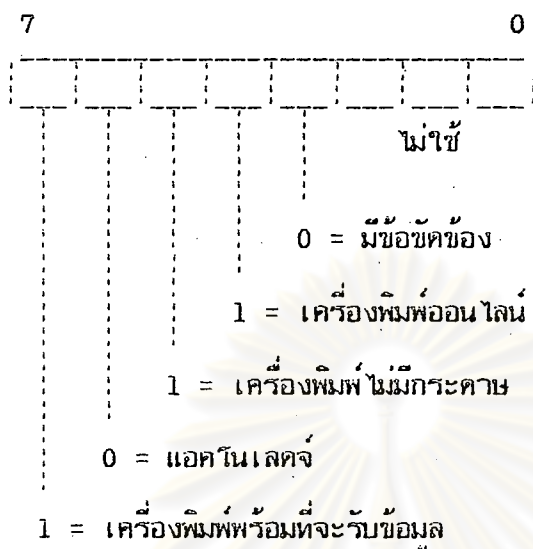
AH = 1 รหัสตัวอักษรนั้นพิมพ์ไม่ได้

2. เมื่อ AH = 1 กำหนดค่าหมายเลขของเครื่องพิมพ์ โดยที่

DX = หมายเลขของเครื่องพิมพ์

3. เมื่อ AH = 2 อ่านค่าสถานะของเครื่องพิมพ์

AH จะเป็นผลลัพธ์ที่ได้รับ มีความหมายดังนี้



รูปที่ 4.6 แสดงค่าสถานะของเครื่องพิมพ์

โปรแกรมปริ้นต์เตอร์ไอโอ ฟังก์ชัน AH = 0

จะส่งรหัสตัวอักษรใน AL ไปยังเครื่องพิมพ์ตั้งส่วนของโปรแกรม

```

PRINTER_IO  PROC      FAR
              STI
              PUSH     DS      ; เก็บค่าในรีจิสเตอร์ต่าง ๆ
              PUSH     DX
              PUSH     SI
              PUSH     CX
              PUSH     BX
              MOV      SI,DATA ; ตำแหน่งค่าตัวของไบออส
              MOV      DS,SI
              MOV      SI,DX   ; หมายเลขเครื่องพิมพ์

```

```

MOV     DX,PRINTER_BASC[SI] ;
OR      DX,DX                ; ตรวจสอบว่ามีเครื่องพิมพ์ต่ออยู่หรือไม่
JZ      B1                   ; กลับไปรแกรมหลักถ้าไม่มี
JMP     B2                   ; ส่งรหัสตัวอักษร ไปพิมพ์
B1 :    POP     BX            ;
        POP     CX            ;
        POP     SI            ;
        POP     DX            ;
        POP     DS            ;
        IRET

```

;; ส่งรหัสตัวอักษรใน AL ไปยังเครื่องพิมพ์

```

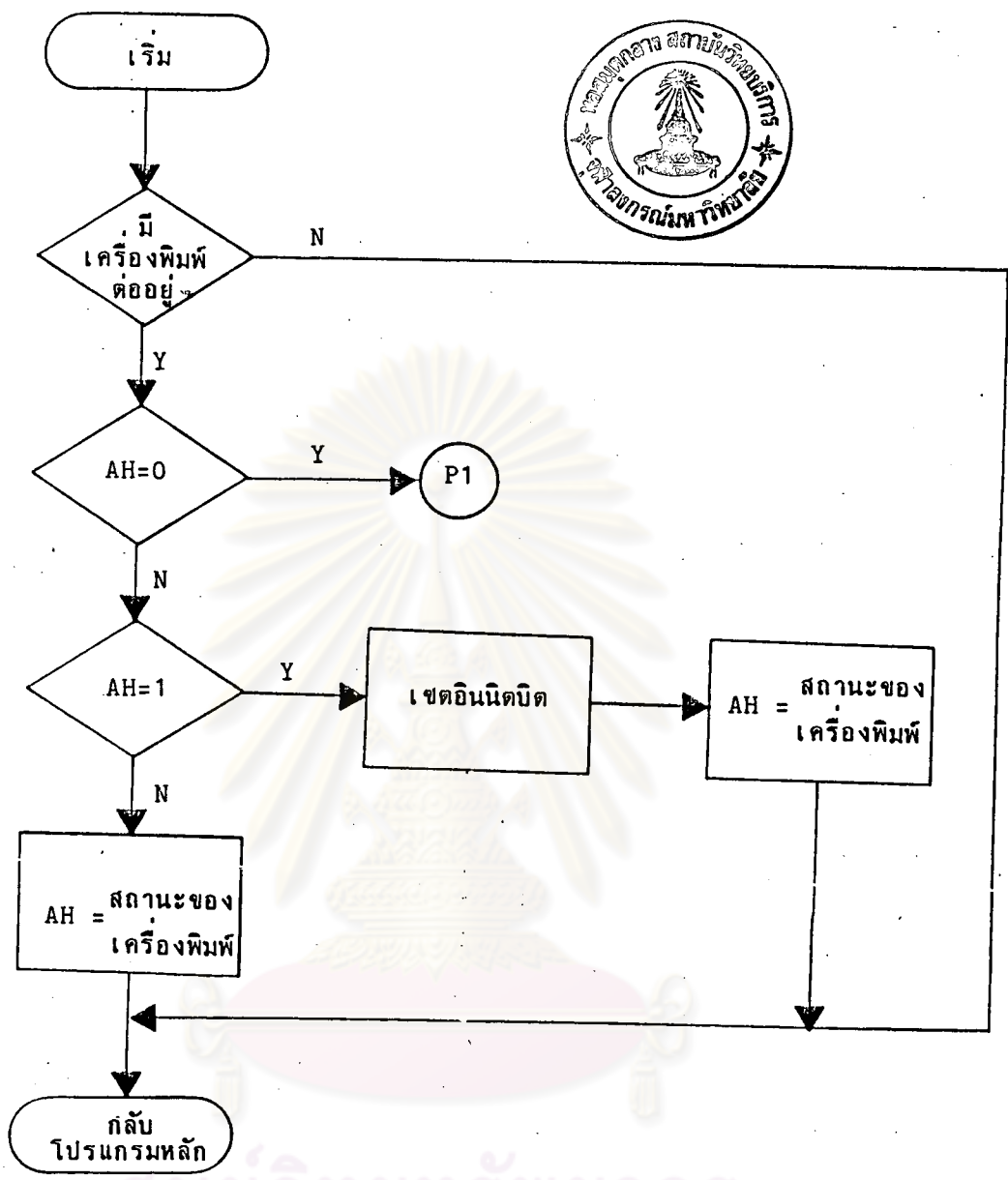
B2 :    PUSH   AX            ; เก็บค่ารหัสตัวอักษร
        MOV    BL,10         ; ตั้งเวลาเช็ค 10 ครั้ง
        XOR   CX,CX         ; ตั้งเคาน์เตอร์เป็น 0
        OUT   DX,AL         ; ส่งรหัสตัวอักษร ไปยังสายคาตาบิต
        INC   DX             ; DX - พอร์ตสถานะของเครื่องพิมพ์
B3 :    IN     AL,DX         ; AL = สถานะของเครื่องพิมพ์
        MOV   AH,AL         ; เก็บสถานะไว้ใน AH
        TEST  AL,80H        ; เครื่องพิมพ์พร้อมจะรับข้อมูลหรือไม่
        JNZ   B4            ; พร้อม
        LOOP  B3            ;
        DEC   BL            ; ลดเวลาเช็คลง

```

```

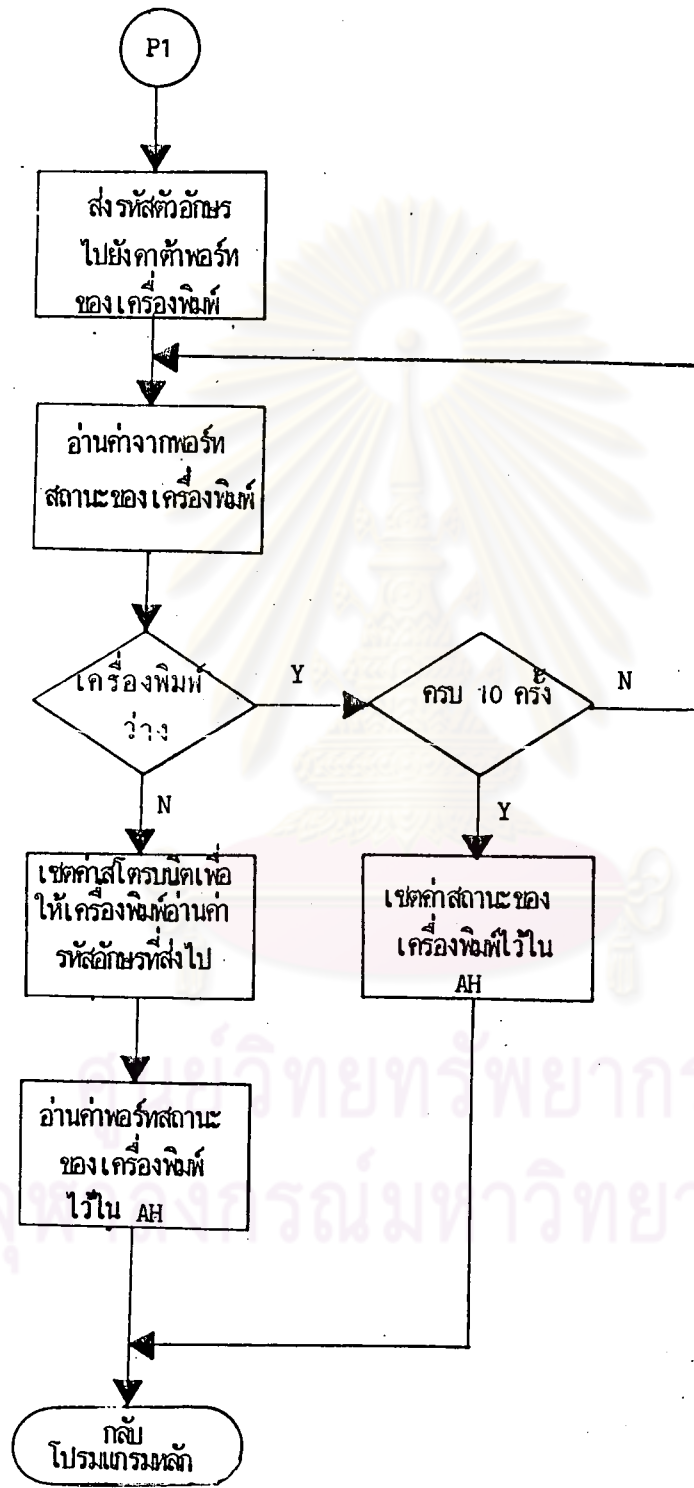
JNZ     B3             ; กลับ ไปอ่านค่าสถานะ เครื่องพิมพ์
OR      AH,1          ; กำหนดบิต เออ เรอร์
AND     AH,0F9H       ;
JMP     SHORT B7      ; กลับ ไปโปรแกรมหลัก
B4:     MOV     AL,0DH  ; กำหนดค่าสไตรบ เพื่อให้ เครื่องพิมพ์อ่านรหัส
        INC     DX      ; พอร์ตควบคุม
        OUT    DX,AL    ;
        MOV     AL,0CH  ; เลิกกำหนดค่าสไตรบ
        OUT    DX,AL    ;
        POP    AX       ; ค่าของรหัสตัวอักษรยังคงเดิม
;
; สถานะของปอร์ตเตอร์
B5:     PUSH   AX       ;
        MOV    DX,PRINTER_BASE(SI) ;
        INC   DX       ; พอร์ตสถานะ
        IN    AL,DX    ; อ่านค่าสถานะ
        MOV   AH,AL    ; เก็บสถานะใน AH
        AND  AH,0F8H   ;
B7:     POP    DX       ;
        MOV   AL,DL    ;
        XOR  AH,48H    ;
        JMP  B1        ; กลับ ไปโปรแกรมหลัก
PRINTER_IO ENDP

```

รูปที่ 4.7 การทำงานของโปรแกรมปริ้นเตอร์ไอโอ

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.7 (ต่อ)