

## บทที่ 10 โครงสร้างควบคุม

เราได้ศึกษาคำสั่งกระโดดในบทที่ 9 โดยคำสั่งกลุ่มนี้ทำให้เราสามารถเขียนโปรแกรมให้มีการทำงานที่ซับซ้อนขึ้นได้. แต่การใช้คำสั่งเหล่านี้ยังไม่เป็นระบบทำให้โปรแกรมที่เขียนขึ้นนั้นทำความเข้าใจได้ยาก และมีลักษณะเหมือนเส้นสปาเก็ตตี้ได้. ในบทนี้เราจะพิจารณาการใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมรูปแบบต่าง ๆ . การใช้คำสั่งกระโดดในลักษณะนี้จะทำให้โปรแกรมของเรามีความเป็นโครงสร้างมากขึ้น.

### การสร้างโครงสร้างการตัดสินใจแบบ if-then-else

รูปแบบของโครงสร้างที่ง่ายที่สุดคือโครงสร้างแบบ if-then-else. รูปแบบของโปรแกรมภาษาแอสเซมบลีให้มีโครงสร้างแบบ if-then-else มีลักษณะดังนี้

```

if condition is false then jump to elselabel
    then_actions
    jump to endif_label
elselabel:
    else_actions
endif_label:

```

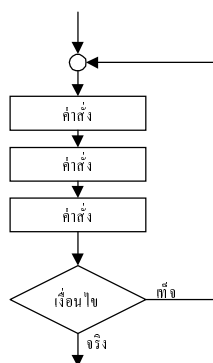
ตัวอย่างของโปรแกรมมีลักษณะโปรแกรมที่ 10.1

(a)	(b)
if AL<10 then	cmp al,10
	jae abovenine
DL:=AL+'0'	mov dl,al
	add dl,'0'
	jmp endif
else	abovenine:
DL:=AL+'A'-10;	mov dl,al
	add al,'A'-10
	endif:

**โปรแกรมที่ 10.1** โปรแกรม (a) แสดงตัวอย่างของโปรแกรมที่เขียนด้วยโครงสร้างของภาษา pascal. โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

### การสร้าง repeat until loop

โครงสร้างของ repeat until loop ในภาษาระดับสูงทั่วไปมีลักษณะดังรูปที่ 10.1



รูปที่ 10.1 โครงสร้างควบคุมแบบ repeat until

เราสามารถเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลีโดยมีโครงสร้างแบบ repeat until ได้ ดังตัวอย่าง

(a)	(b)
BL:=1;	mov bl,1
CX:=0;	mov cx,0
DX:=0;	mov dx,0
repeat	startloop:
DX:=DX+BL*BL;	mov al,bl
	mul bl ;result in ax
	add dx,ax
BL:=BL+1;	inc bx
CX:=CX+1;	inc cx
	cmp dx,100
until (DX>100);	jbe startloop

**โปรแกรมที่ 10.2** โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา pascal ที่ใช้โครงสร้างแบบ repeat until loop. โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

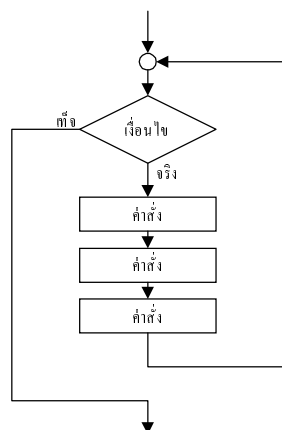
รูปแบบของโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากับ repeat until loop มีลักษณะเป็นดังนี้

```

startlabel:
    action;
    ...
    action;
    if condition is false then jump to startlabel
  
```

### การสร้าง while loop

โครงสร้างของ while loop ในภาษาระดับสูงทั่วไปมีลักษณะเป็นดังนี้



**รูปที่ 10.2** โครงสร้างควบคุมแบบ while

เราสามารถเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลีโดยมีโครงสร้างแบบ whil loop ได้ ดังตัวอย่าง

(a)	(b)
while (DL<>13) and	startloop:
(CX<20) do	cmp dl,13
	jz endloop
	cmp cx,20
	jae endloop
begin	add al,dl
AX:=AX+DL;	adc ah,0

```

BX:=BX+1;          inc    bx
DL:=DATA[BX];      mov    dl,data[bx]
CX:=CX+1;          inc    cx

end;                jmp    startloop
endloop:

```

โปรแกรมที่ 10.3 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา pascal ที่ใช้โครงสร้างแบบ while loop. โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

รูปแบบของโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากับ while loop มีลักษณะเป็นดังนี้

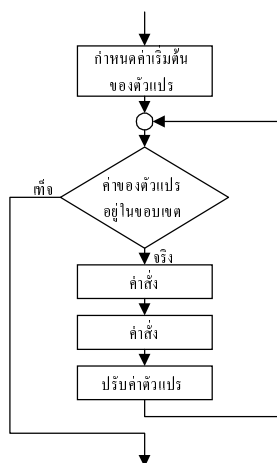
```

startlabel:
    if condition is false then jump to endlabel
    action
    ...
    action
    jump to startlabel
endlabel:

```

### การสร้าง for loop

เราสามารถใช้คำสั่งกระโดดในการสร้างโครงสร้างแบบ for loop ได้ นอกจากนั้นเรายังสามารถใช้คำสั่งกลุ่ม LOOP ในการสร้างโครงสร้างแบบ for loop ได้เช่นเดียวกัน. โครงสร้างของ for loop มีลักษณะดังรูปที่ 10.3.



รูปที่ 10.3 โครงสร้างของ for loop

รูปแบบของโปรแกรมภาษาแอสเซมบลีที่มีโครงสร้างเป็นแบบ for loop มีลักษณะดังนี้

```

    initialize index variables
startloop:
    if index value is not in the range then jump to endloop
    action
    ...
    action
    update index variable
    jump to startloop
endloop:

```

ตัวอย่างของโปรแกรมที่เขียนโดยใช้รูปแบบลักษณะนี้เป็นดังโปรแกรมที่ 10.4.

(a)	(b)
CX:=0;	mov cx,0
for DL:=1 to 100 do	mov dl,1
begin	startloop:
if DL mod 7=0 then	cmp dl,100
	ja endloop
	mov al,dl
	mov ah,0
	mov bl,7
	div bl
	cmp ah,0
	jne endif
CX:=CX+1;	inc cx
	endif:
end;	inc dl
	jmp startloop
	endloop:

**โปรแกรมที่ 10.4** โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา pascal ที่ใช้โครงสร้างแบบ for loop. โปรแกรม (b) แสดงโปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน

เรายังสามารถใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ได้ เช่นเดียวกัน ดังต่อไปนี้

```

      set the value of CX
startloop:
      actions
      LOOP startloop

```

แต่การใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ไม่สามารถสร้างโครงสร้างการกระทำซ้ำที่มีความซับซ้อนมาก ๆ ได้ เช่นการกระทำซ้ำที่มีวงรอบของการกระทำซ้ำซ้อนกันหลาย ๆ วง.

สำหรับคำสั่ง LOOPZ และ LOOPNZ นั้น เราสามารถนำมาใช้ในการสร้างโครงสร้างควบคุมที่มีความซับซ้อนขึ้นได้ โดยโครงสร้างดังกล่าวจะมีลักษณะปนกันระหว่าง for loop และ while loop หรือ repeat until. นั่นคือเงื่อนไขควบคุมการกระทำซ้ำจะขึ้นกับทั้งค่าของรีจิสเตอร์ (มีลักษณะคล้ายโครงสร้างแบบ for loop) และเป็นเงื่อนไขจริง ๆ (คล้าย repeat until loop และ while loop.) ดังในตัวอย่างโปรแกรมต่อไปนี้

(a)	(b)
AX:=0;	mov ax,0
CX:=100;	mov cx,100
repeat	startloop:
AX:=AX+data[BX];	add ax,data[BX]
BX:=BX+2;	add bx,2
CX:=CX-1;	cmp data[BX],0
until (data[BX]=0) or	loopnz startloop
(CX=0);	

**โปรแกรมที่ 10.5** โปรแกรม (a) โปรแกรมที่เขียนด้วยภาษา pascal. โปรแกรม (b) โปรแกรมภาษาแอสเซมบลีที่เทียบเท่ากัน