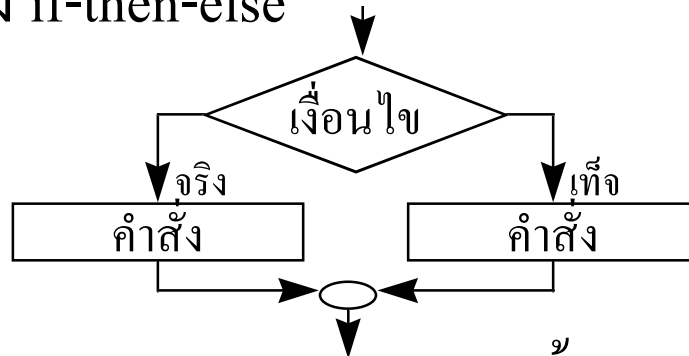


บทที่ 10 โครงสร้างควบคุม

- โครงสร้างการตัดสินใจแบบ if-then-else
- โครงสร้างการกระทำซ้ำแบบ repeat-until
- โครงสร้างการกระทำซ้ำแบบ while
- โครงสร้างการกระทำซ้ำแบบ for

โครงสร้าง if-then-else

- โครงสร้าง if-then-else



- รูปแบบโปรแกรมภาษาแอสเซมบลี

```
if condition is false then jump to else_label
then_actions
jump to endif_label
else_label:
    else_actions
endif_label:
```

EXAMPLE

```
if AL<10 then
```

```
    DL:=AL+'0'
```

```
else
```

```
    DL:=AL+'A'-10;
```

```
cmp    al,10
```

```
jae    abovenine
```

```
mov     dl,al
```

```
add     dl,'0'
```

```
jmp     endif
```

```
abovenine:
```

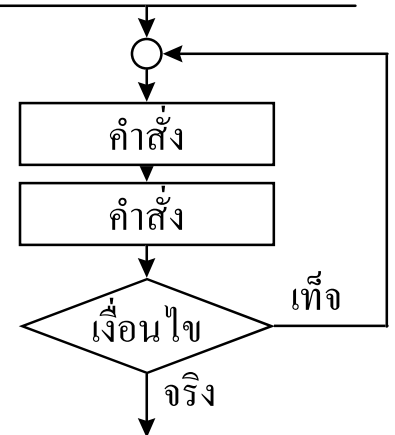
```
mov     dl,al
```

```
add     al,'A'-10
```

```
endif:
```

โครงสร้าง repeat-until

- โครงสร้าง repeat-until



- รูปแบบโปรแกรมภาษาแอสเซมบลี

```
startlabel:
    actions
    ...
    actions
    if condition is false then jump to startlabel
```

EXAMPLE

repeat

```
DX:=DX+BL*BL;
```

```
BL:=BL+1;
```

```
CX:=CX+1;
```

until (DX>100);

startlabel:

```
mov    al,bl
```

```
mul     bl
```

```
add     dx,ax
```

```
inc     bl
```

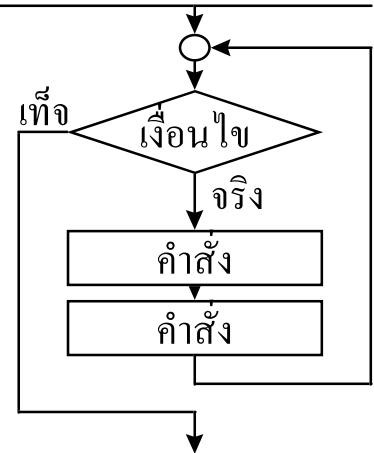
```
inc     cx
```

```
cmp     dx,100
```

```
jbe     startlabel
```

โครงสร้าง while

- โครงสร้าง while



- รูปแบบโปรแกรมภาษาแอสเซมบลี

startlabel:

if *condition is false* then jump to endlabel
actions

...

actions

jump to startlabel

endlabel:

EXAMPLE

```
while (DL<>13) and
      (CX<20) do
```

```
begin
```

```
AX:=AX+DL;
BX:=BX+1;
DL:=DATA[BX]
CX:=CX+1;
```

```
end;
```

startloop:

```
cmp    dl,13
jz      endloop
cmp    cx,20
jae     endloop
```

```
add     al,dl
adc     ah,0
inc     bx
mov     dl,data[bx]
inc     cx
```

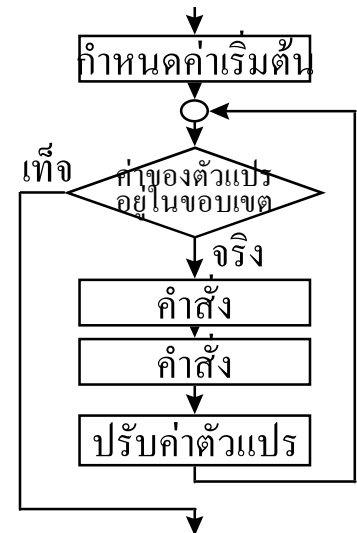
```
jmp     startloop
```

endloop:

โครงสร้าง for

- โครงสร้าง for

- รูปแบบโปรแกรมภาษาแอสเซมบลี
 - ไม่ใช่คำสั่ง LOOP : ยุ่งยากในการใช้



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

- ใช้คำสั่ง LOOP : สะดวกกว่า แต่ไม่สามารถใช้ในการวนรอบที่ซับซ้อนได้

```
set the value of CX
startloop:
  actions
  LOOP startloop
```

โครงสร้าง for

EXAMPLE

```
CX:=0;
for DL:=1 to 100 do
begin
    if DL mod 7 = 0
    then
        CX:=CX+1;
    end;
end;
```

```
mov     cx,0
mov     dl,1
startloop:
cmp     dl,100
ja      endloop
```

```
mov     al,dl
mov     ah,0
mov     bl,7
div     bl
```

```
cmp     ah,0
jne     endif
```

```
inc     cx
```

```
endif:
```

```
inc     dl
jmp     startloop
endloop:
```

- โครงสร้างการทำงานของคำสั่ง **LOOPZ** และ **LOOPNZ**
มีลักษณะปนกันระหว่างโครงสร้าง for และ repeat

EXAMPLE

```
AX:=0;
CX:=100;
repeat
    AX:=AX+data[BX];
    BX:=BX+2;
    CX:=CX-1;
until (data[BX]=0) or
      (CX=0);
```

```
mov     ax,0
mov     cx,100
startloop:
add     ax,data[bx]
add     bx,2

cmp     data[bx],0
loopnz  startloop
```