

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

-----***-----



BÁO CÁO BÀI TẬP LỚN

MÔN : KĨ THUẬT VI XỬ LÝ

Giảng viên : Phạm Hoàng Duy

Nhóm : 10

Nhóm thực hiện : 04

Thành viên nhóm :

Nguyễn Trọng Khang – B18DCCN301

Chu Quang Ngọc - B18DCCN455

Lê Thành Luân – B18DCCN378

Phan Chính Quảng – B18DCCN488

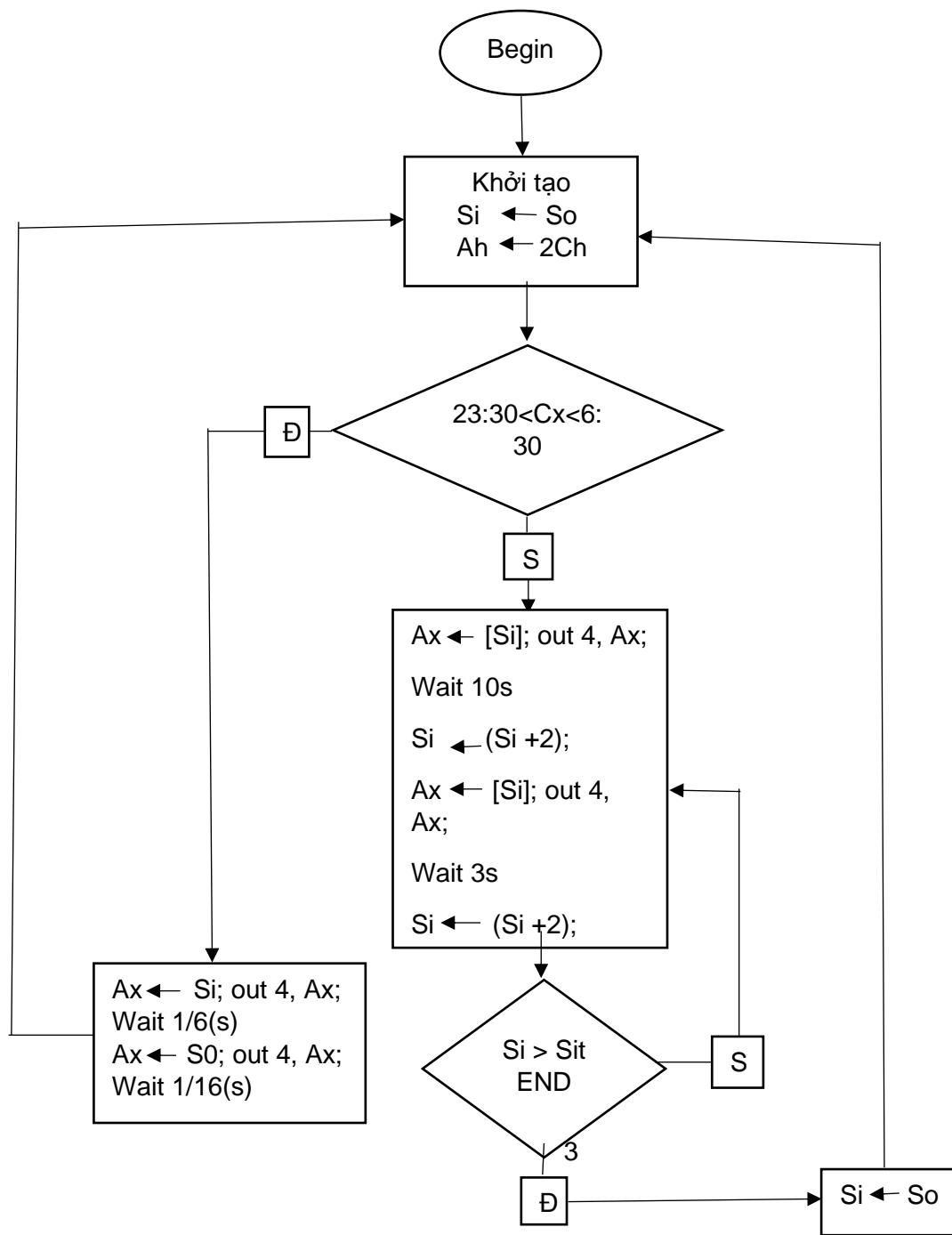
LỜI MỞ ĐẦU.....	Error! Bookmark not defined.
I.Bài tập 1 : Điều khiển đèn giao thông.....	Error! Bookmark not defined.
1.Cách thức hoạt động.....	Error! Bookmark not defined.
2. Lưu đồ	Error! Bookmark not defined.
3. Code.....	Error! Bookmark not defined.
II.Bài tập 2 : Themometer(Bếp thông minh).....	Error! Bookmark not defined.
1. Cách thức hoạt động	Error! Bookmark not defined.
2. Lưu đồ	Error! Bookmark not defined.
3. Code.....	Error! Bookmark not defined.
III.Bài tập 3 : Stepper motor(Động cơ bước).....	Error! Bookmark not defined.
1. Cách thức hoạt động	Error! Bookmark not defined.
2.Lưu đồ	Error! Bookmark not defined.
3. Code.....	Error! Bookmark not defined.
LỜI CẢM ƠN.....	Error! Bookmark not defined.

I. Bài tập 1 : Điều khiển đèn giao thông

1. Cách thức hoạt động

- Các hướng được điều khiển độc lập với nhau.
- Chu trình bình thường 6:30 đến 23:30:
 - + Xanh trong 35s, vàng 3s, đỏ. Sau đó chuyển sang hướng kế tiếp bên phải.
 - + Khi đèn xanh còn 3s, thì đèn xanh sẽ nháy với tần số 3Hz (3 lần trong 1s).
- Chu trình giờ thấp điểm 23:30 đến 6:30(hôm sau): Tất cả các hướng đều bật đèn vàng nhấp nháy với tần số 3Hz.

2. Lưu đồ



3. Code

```
; controlling external device with 8086 microprocessor.  
; realistic test for c:\emu8086\devices\Traffic_Lights.exe
```

```
#start=Traffic_Lights.exe#
```

```
waiting macro i,j
```

```
    mov dx, j  
    mov cx, i  
    mov ah, 86h  
    int 15h  
endm
```

```
org 100h
```

```
jmp start
```

s0	dw	0000_0010_0100_1100b
s1	dw	0000_0010_0100_1010b
s2	dw	0000_0010_0110_0001b
s3	dw	0000_0010_0101_0001b
s4	dw	0000_0011_0000_1001b
s5	dw	0000_0010_1000_1001b

```
s6      dw    0000_1000_0100_1001b  
s7      dw    0000_0100_0100_1001b  
sit_end = $
```

Start:

```
mov si, offset s0
```

continue:

```
mov ah, 2Ch
```

```
int 21h
```

```
cmp cx, 0000_0110_0001_1110b
```

```
jl thapdiem
```

```
cmp cx, 0001_0111_0001_1110b
```

```
jg thapdiem
```

binhthuong:

```
mov ax, [si]
```

```
out 4, ax
```

```
waiting 0216h, 0EC0h
```

```
add si, 2
```

```
mov ax, [si]
```

```
out 4, ax
```

```
waiting 2Dh, 0C6C0h
```

```
add si, 2
```

```
cmp si, sit_end  
jb binhthuong  
mov si, offset s0  
jmp continue
```

thapdiem:

```
mov ax, 0000_0100_1001_0010b  
out 4, ax  
waiting 2h, 8B0Ah  
xor ax, ax  
out 4, ax  
waiting 2h, 8B0Ah  
jmp continue
```

ret

END

Giải thích:

- Marco được dùng để lưu lại hoạt động. Khi gọi các lệnh waiting i, j thì các lệnh bên dưới nó được thực hiện

- Mov si, offset s0; Si chỉ vào địa chỉ dữ liệu s0

Chu trình 1:

- Dùng hàm 2ch của ngắt 21h để đọc thời gian trên máy

- So sánh cx với 6_30, nếu nhỏ hơn thì nhảy đến Thấp Điểm hoặc lớn hơn thì nhảy đến Bình Thường

- Tiếp tục so sánh cx với 23_30, nếu lớn hơn thì nhảy đến thấp điểm hoặc nếu nhỏ hơn thì nhảy đến Bình Thường

- Tiếp dòng 41:

+ mov ax, [Si] chuyển nội dung ở ô nhớ ds:[si] vào ax và mã là s0

+ add si, 2

```
mov ax, [si]      ; chuyển nội dung ô nhớ ds: [si +2] vào ax, mã là s2
out 4, ax        ;ghi ra cổng
waiting 2dh, 0C6C0h ; chờ 3s = 002D_C6C0h(s)
add si, 2        ; si = si+2
cmp si, sit_end  ; so sánh si với sit_end($)
jb binhthuong   ; Nếu nhỏ hơn thì nhảy đến binhthuong
mov si, offset s0
jmp countine     ;nếu không thì nhảy đến countine
```

- Tiếp dòng 54: Chu trình 2

```
+ mov ax, 0000_0100_1001_0010b      ;bật tất cả đèn vàng
out 4, ax                      ; ghi ra cổng
waiting 2h, 8B0Ah           ; đợi 1/6(s) = 0002_8B0Ah
                                ; tắt 1/6(s), bật 1/6 đến 1s nháy 3 lần
xor, ax, ax                  ; ax=0, tắt toàn bộ đèn
out 4, ax                      ; ghi ra cổng
waiting 2h, 8B0Ah           ; đợi 1/6(s)
jmp countine    ; nhảy đến coutine tiếp tục đọc dữ liệu thời gian , nếu
                  ; thời gian còn ở giờ thấp điểm thì đèn vàng sáng lên
```

- Cứ như thế lặp đi lặp lại

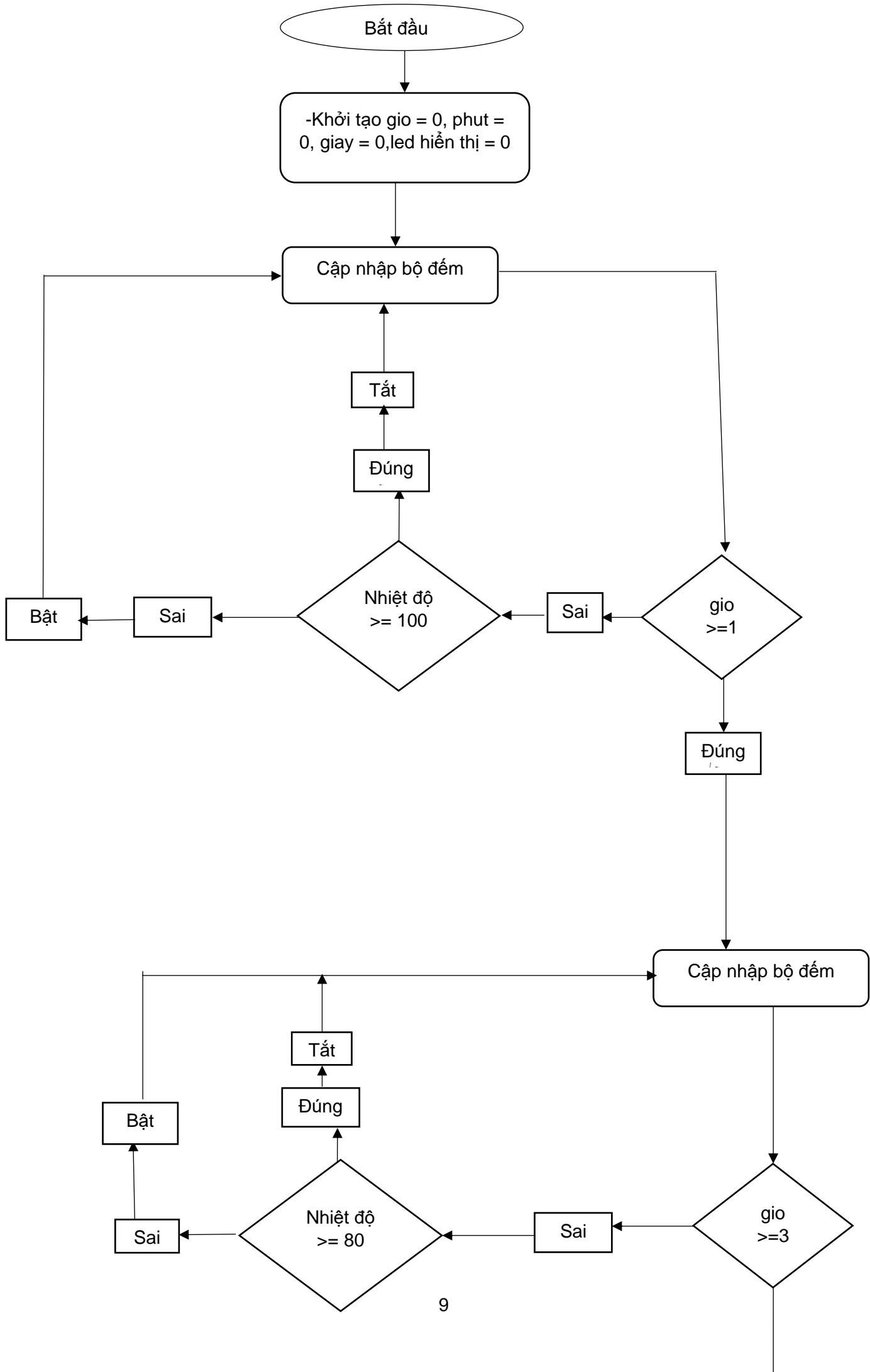
II. Bài tập 2 : Themometer (Bếp thông minh)

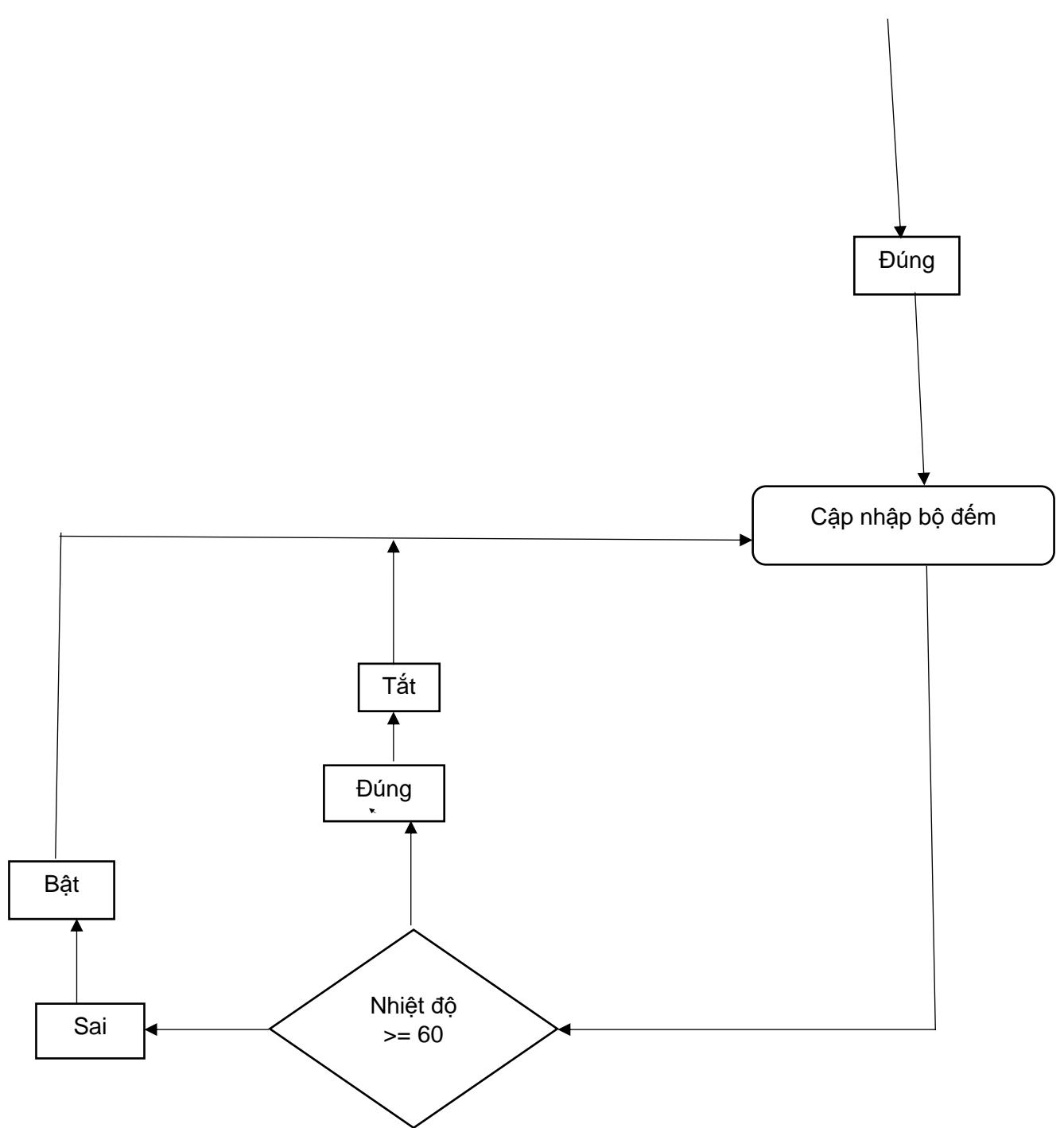
1. Cách thức hoạt động

- Bếp được bật và đun sôi từ 0 độ đến 100 độ
- Quá trình tiếp theo đun sôi bếp ở 100 độ
 - + Giữ nhiệt độ bằng cách nếu nhiệt độ > 100 thì tắt bếp
 - + Nếu nhiệt độ < 100 thì bật bếp
 - + Quá trình sẽ được wait 1s và sẽ được lặp lại 3600 lần
- Quá trình tiếp theo đun sôi bếp ở 80 độ

- + Bếp đang ở nhiệt độ 100 độ
 - + Giữ nhiệt độ bằng cách nếu nhiệt độ > 80 thì tắt bếp
 - + Nếu nhiệt độ < 80 thì bật bếp
 - + Quá trình sẽ được wait 1s và sẽ được lặp lại 7200 lần
- Quá trình tiếp theo đun sôi bếp ở 60 độ
 - + Bếp đang ở nhiệt độ 80 độ
 - + Giữ nhiệt độ bằng cách nếu nhiệt độ > 60 thì tắt bếp
 - + Nếu nhiệt độ < 60 thì bật bếp
 - + Quá trình được lặp vô hạn
 - Hiển thị đồng hồ

2. Lưu đồ





3. code

```

include 'emu8086.inc'
#start=thermometer.exe#
#make-bin#
#start=led_display.exe#
#make_bin#

```

```
name "thermo"
```

```
turn MACRO x
```

```
    mov al, x
```

```
    out 127, al
```

```
endm
```

```
waiting MACRO c, d
```

```
    mov cx, c
```

```
    mov dx, d
```

```
    mov ah, 86h
```

```
    int 15h
```

```
endm
```

```
org 100h
```

```
turn 1
```

```
mov bx,0
```

```
init:
```

```
    in al, 125
```

```
    cmp al, 100
```

```
    jl init
```

```
mov cx, 3600
```

```
start100:
```

```
    mov dl,100
```

```
inc bx
push bx
mov ax,bx
mov bl,60
div bl
mov dh,ah
mul dl
add al,dh
out 199,ax
pop bx
pusha
mov dl, 100
call GuiAm
waiting 0fh, 4240h
```

```
popa
loop start100
```

```
mov cx, 7200
mov bx,0
start80:
    mov dl,100
    inc bx
    push bx
    mov ax,bx
    mov bl,60
    div bl
```

```
mov dh,ah  
mul dl  
add al,dh  
out 199,ax  
pop bx  
pusha  
mov dl, 80  
call GuiAm  
waiting 0fh, 4240h  
popa  
loop start80
```

start60:

```
mov dl,100  
inc bx  
push bx  
mov ax,bx  
mov bl,60  
div bl  
mov dh,ah  
mul dl  
add al,dh  
out 199,ax  
pop bx  
mov dl, 60  
call GuiAm
```

jmp start60

ret

GiuAm PROC

in al, 125

cmp al, dl

jl low

cmp al, dl

jle ok

jg high

low:

turn 1

jmp ok

high:

turn 0

jmp ok

ok:

ret

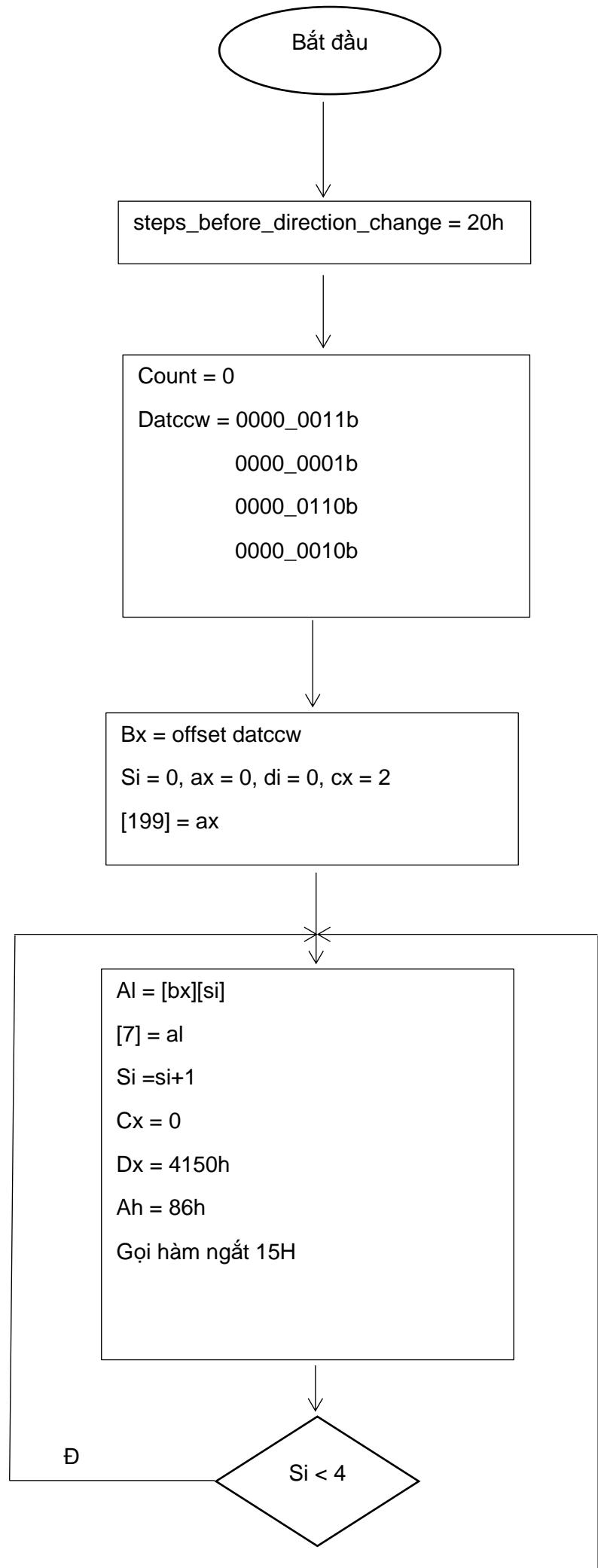
endp

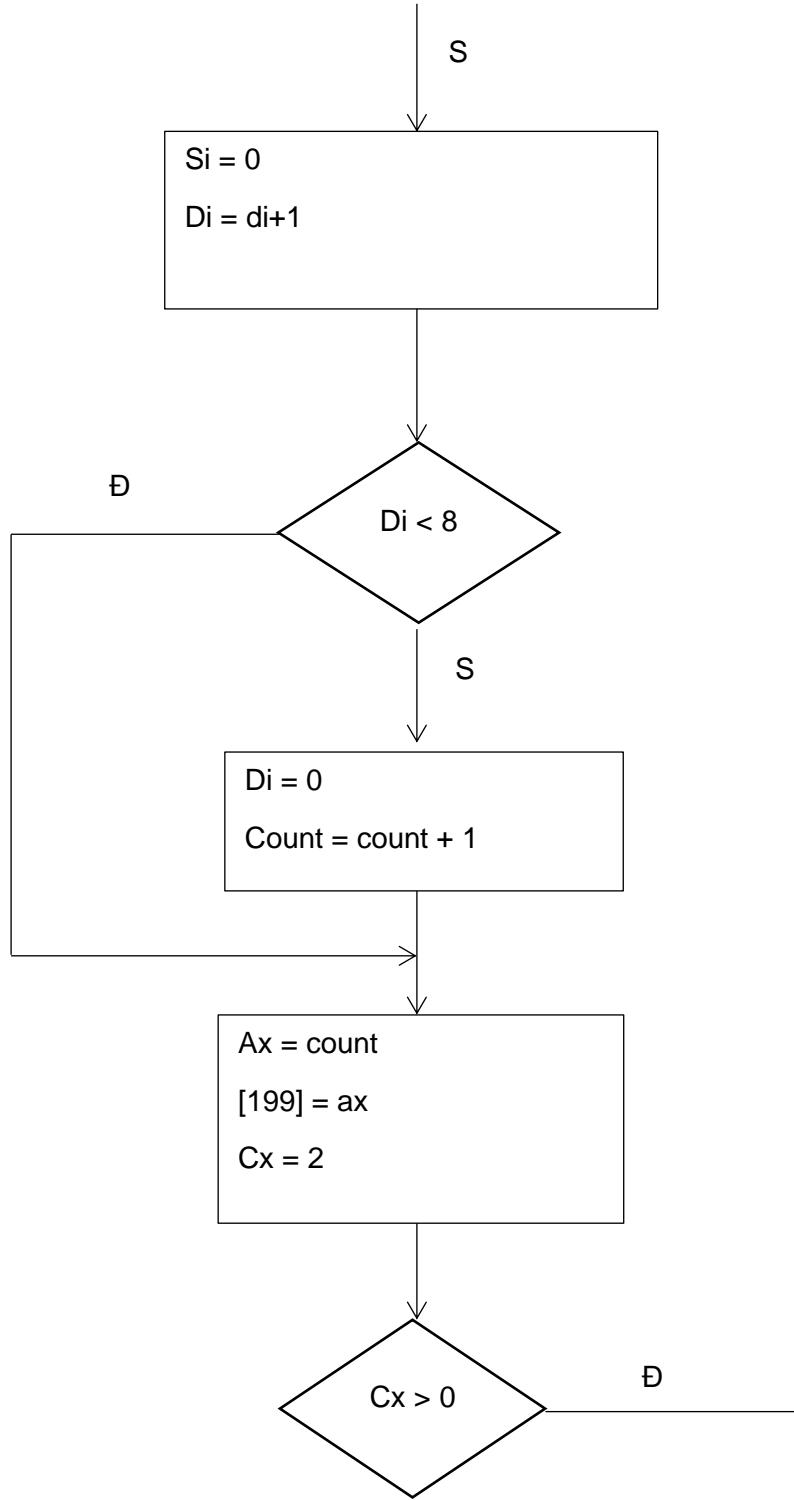
III. Bài tập 3 : Stepper motor (Động cơ bước)

1. Cách thức hoạt động

- Trong chương trình ta sử dụng một mô tơ bước 3 pha, nó có 3 nam châm được điều khiển bởi các bít 0, 1, và 2
- Nam châm sẽ hoạt động khi nó có màu đỏ và mũi tên ở góc trên bên phải biểu thị cho chiều quay của mô-tơ. Đường kẻ màu xanh để ta nhận biết mô-tơ đang quay.
- Trong chương trình mô-tơ bước sẽ được điều khiển thông qua cổng dữ liệu I/O7
- Nạp số vòng quay vào cổng 199(LED hoạt động trên cổng 199) để hiển thị số vòng mô tơ chạy được lên đèn LED. Ta sử dụng lại thanh ghi ax để lấy dữ liệu rồi chuyển ra cổng 199. Sau đó lại tiếp tục vòng lặp mới để mô tơ thực hiện những vòng quay tiếp.

2. Lưu đồ





3. Code

```
#start=stepper_motor.exe#
```

```
#start=led_display.exe#
```

```
name "stepper"
```

```
#make_bin#
```

```
steps_before_direction_change = 20h ; 32 (decimal)
```

```
jmp start
```

```
cout dw 0
```

```
datccw db 0000_0011b
```

```
db 0000_0001b
```

```
db 0000_0110b
```

```
db 0000_0010b
```

```
start:
```

```
mov bx, offset datccw ; start from clock-wise half-step.
```

```
mov si, 0
```

```
mov ax, 0
```

```
mov di, 0
```

```
mov cx, 2 ; step counter
```

```
out 199, ax
```

```
next_step:
```

```
mov al, [bx][si]
```

```
out 7, al
```

```
inc si
```

```
mov cx, 0h  
mov dx, 4150h  
mov ah,86h  
int 15h
```

```
cmp si, 4  
jb next_step  
mov si, 0
```

```
inc di  
cmp di, 8  
jb nhay  
mov di, 0  
inc cout  
nhay:  
mov ax, cout  
out 199,ax  
mov cx,2  
loop next_step
```