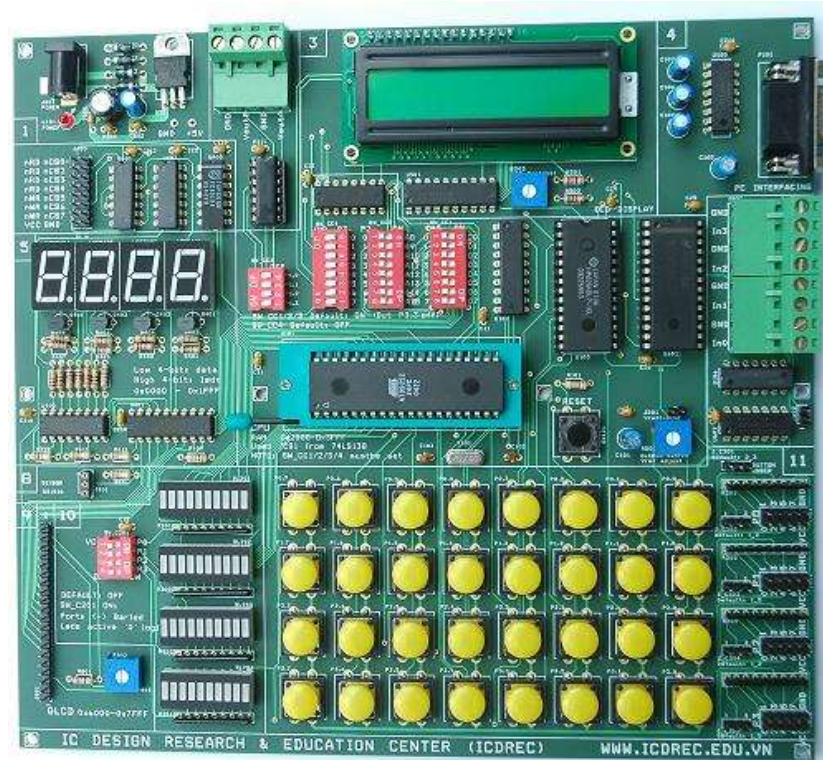


Chương 2: Họ vi điều khiển 8051 (lập trình hợp ngữ)



- Hiểu được tập lệnh bên trong vđk 8051
- Biết được cách sử dụng các lệnh ASM trên 8051
- Biết được cách sử dụng các cấu trúc, các dữ liệu, hàm ASM
- Vận dụng để viết các chương trình đơn giản trên 8051

- Tập lệnh hợp ngữ
- Cấu trúc chương trình hợp ngữ

COMPUTER ENGINEERING

- RAM nội được phân chia như thế nào?
- Tác dụng của các bank thanh ghi, bit nào dùng để chọn bank thanh ghi?
- Nêu một số thanh ghi có chức năng đặc biệt?
- Các kiểu định địa chỉ nhớ?

COMPUTER ENGINEERING

Cấu trúc chung của lệnh:

- Mã lệnh Toán hạng 1 Toán hạng 2 Toán hạng 3
- Trong đó:
 - Mã lệnh: tên gọi nhớ chức năng của lệnh
 - Các toán hạng của lệnh có thể có hoặc không

COMPUTER ENGINEERING

Ví dụ:

- ❑ RET (không có toán hạng)
- ❑ JZ TEMP (có 1 toán hạng)
- ❑ ADD A, R3 (có 2 toán hạng)
- ❑ CJNE A, #20, LOOP (có 3 toán hạng)

COMPUTER ENGINEERING

Phân loại tập lệnh:

- Các lệnh toán học
- Các lệnh logic
- Các lệnh vận chuyển dữ liệu
- Các lệnh thao tác bit
- Các lệnh đọc cổng
- Các lệnh điều khiển chương trình

Các lệnh toán học

Lệnh	Chức năng
ADD	Lệnh cộng
ADDC	Lệnh cộng có cờ nhớ
SUBB	Lệnh trừ
INC	Tăng 1
DEC	Giảm 1
MUL	Nhân
DA	Hiệu chỉnh thập phân
DIV	Chia

Các lệnh logic

Lệnh	Chức năng
ANL	Lệnh AND logic
ORL	Lệnh OR logic
XRL	Lệnh XOR logic
CLR	Lệnh xóa logic
CPL	Lệnh đảo giá trị logic
RL	Lệnh quay trái
RR	Lệnh quay phải
RLC	Lệnh quay trái qua cờ C
RRC	Lệnh quay phải qua cờ C
SWAP	Hoán đổi 2 nửa của A

Các lệnh vận chuyển dữ liệu

STT	Cú pháp		Mô tả	Số byte	Số chu kỳ
	Mã lệnh	Toán hạng			
1	MOV	A,Rn	Copy giá trị của toán hạng bên phải cho vào toán hạng bên trái (các toán hạng đều là 8bit)	1	1
2	MOV	A,direct		2	1
3	MOV	A,@Ri		1	1
4	MOV	A,#data		2	1
5	MOV	Rn,A		1	1
6	MOV	Rn,direct		2	2
7	MOV	Rn,#data		2	1
8	MOV	Direct,A		2	1
9	MOV	Direct,Rn		2	2
10	MOV	Direct,direct		3	2
11	MOV	Direct,@Ri		2	2
12	MOV	Direct,#data		3	2
13	MOV	@Ri,A		1	1
14	MOV	@Ri,direct		2	1
15	MOV	@Ri,#data		2	1



Các lệnh vận chuyển dữ liệu (tt)



STT	Cú pháp		Mô tả	Số byte	Số chu kỳ
	Mã lệnh	Toán hạng			
16	MOV	Dptr,#data16	Đưa giá trị 16bit vào thanh ghi DPTR	3	2
17	MOVC	A,@A+dptr	Đọc giá trị bộ nhớ chương trình tại địa chỉ = A + DPTR, cất kết quả vào A	1	2
18	MOVC	A,@A+PC	Đọc giá trị bộ nhớ chương trình tại địa chỉ = A + PC, cất kết quả vào A	1	2
19	MOVX	A,@Ri	Đọc vào A giá trị của bộ nhớ ngoài tại địa chỉ = Ri	1	2
20	MOVX	A,@dptr	Đọc vào A giá trị của bộ nhớ ngoài tại địa chỉ = DPTR	1	2
21	MOVX	@dptr,A	Ghi giá trị của A vào bộ nhớ ngoài tại địa chỉ = DPTR	1	2



Các lệnh vận chuyển dữ liệu (tt)



COMPUTER ENGINEERING

Các lệnh thao tác bit

Lệnh	Chức năng
SETB BIT	Thiết lập bit = 1
CLR BIT	Xóa bit = 0
CPL BIT	Bù bit
JB BIT, ĐÍCH	Nhảy về đích nếu bit = 1
JNB BIT, ĐÍCH	Nhảy về đích nếu bit = 0
JBC BIT, ĐÍCH	Nhảy về đích nếu bit = 1 và xóa bit

Lệnh	Ví dụ	Mô tả
MOV A, PX	MOV A, P2	Chuyển dữ liệu P2 vào ACC
JNB PX.Y, ĐÍCH	JNB P2.1, ĐÍCH	Nhảy tới đích nếu P2.1 = 0
JB PX.Y, ĐÍCH	JNB P2.1, ĐÍCH	Nhảy tới đích nếu P2.1 = 1
MOV C, PX.Y	MOV C, P2.4	Copy trạng thái chân P2.4 vào CY

Đọc trạng thái cổng vào

Các lệnh đọc cổng (tt)

Lệnh	Ví dụ
ANL PX	ANL P1, A
ORL PX	ORL P2, A
XRL PX	XRL P0, A
JBC PX.Y, ĐÍCH	JBC P1.1, ĐÍCH
CPL PX	CPL P1.2
INC PX	INC P1
DEC PX	DEC P2
DJN2 PX	DJN2 P1, ĐÍCH
MOV PX	MOV P1.2, C
CLR PX	CLR P2.3
SETB PX	SETB P2.3

Đọc chốt trong của cổng ra

- Nhảy có điều kiện
- Nhảy vô điều kiện
- Các lệnh gọi

COMPUTER ENGINEERING

Lệnh	Hoạt động
JZ	Nhảy nếu $A = 0$
JNZ	Nhảy nếu $A \neq 0$
DJNZ	Giảm và nhảy nếu $A = 0$
CJNE A, byte	Nhảy nếu $A \neq \text{byte}$
CJNZ re, #data	Nhảy nếu $\text{byte} = \text{data}$
JC	Nhảy nếu $CY = 1$
JNC	Nhảy nếu $CY = 0$
JB	Nhảy nếu $\text{bit} = 1$
JNB	Nhảy nếu $\text{bit} = 0$
JBC	Nhảy nếu $\text{bit} = 1$ và xóa nó

Ví dụ: Hãy tìm tổng của các giá trị 79H, F5H và E2H. Đặt vào trong các thanh ghi R0 (byte thấp) và R5 (byte cao).

```
MOV    A, #0          ; Xoá thanh ghi A = 0
MOV    R5, A          ; Xoá R5
ADD    A, #79H        ; Cộng 79H vào A (A = 0 + 79H = 79H)
JNC    N-1            ; Nếu không có nhớ cộng kế tiếp
INC    R5              ; Nếu CY = 1, tăng R5

N-1:   ADD    A, #0F5H  ; Cộng F5H vào A (A = 79H + F5H = 6EH)
                        ; và CY = 1
        JNC    N-2      ; Nhảy nếu CY = 0
        INC    R5        ; Nếu CY = 1 tăng R5 (R5 = 1)
N-2:   ADD    A, #0E2H  ; Cộng E2H vào A (A = 6E + E2 = 50)
                        ; và CY = 1
        JNC    OVER     ; Nhảy nếu CY = 0
        INC    R5        ; Nếu CY = 1 tăng R5
OVER:  MOV    R0, A      ; Bây giờ R0 = 50H và R5 = 02
```

- ❑ Nhảy xa LJMP: Nhảy xa LJMP là một lệnh 3 byte trong đó byte đầu tiên là mã lệnh còn hai byte còn lại là địa chỉ 16 bit của đích. Địa chỉ đích 02 byte có phép một phép nhảy đến bất kỳ vị trí nhớ nào trong khoảng 0000 - FFFFH.
- ❑ Nhảy gần SJMP: Trong 2 byte này thì byte đầu tiên là mã lệnh và byte thứ hai là chỉ tương đối của địa chỉ đích. Đích chỉ tương đối trong phạm vi 00 - FFH được chia thành các lệnh nhảy tới và nhảy lùi: Nghĩa là -128 đến +127 byte của bộ nhớ tương đối so với địa chỉ hiện thời của bộ đếm chương trình. Nếu là lệnh nhảy tới thì địa chỉ đích có thể nằm trong khoảng 127 byte từ giá trị hiện thời của bộ đếm chương trình. Nếu địa chỉ đích ở phía sau thì nó có thể nằm trong khoảng -128 byte từ giá trị hiện hành của PC.

- ❑ Lệnh gọi xa LCALL: Trong lệnh 3 byte này thì byte đầu tiên là mã lệnh, còn hai byte sau được dùng cho địa chỉ của chương trình con đích..
- ❑ Lệnh gọi tuyệt đối ACALL (Absolute call): Lệnh ACALL là lệnh 2 byte khác với lệnh LCALL dài 3 byte. Do ACALL chỉ có 2 byte nên địa chỉ đích của chương trình con phải nằm trong khoảng 2k byte địa chỉ vì chỉ có 11bit của 2 byte được sử dụng cho địa chỉ.

COMPUTER ENGINEERING

- Các thành phần cơ bản của Assembly
- Khai báo trong lập trình ASM
- Cấu trúc một chương trình ASM

COMPUTER ENGINEERING

Tên	Lệnh	Toán hạng	Chú thích
A:	Mov	AH, 10h	; Đưa giá trị 10h vào thanh ghi AH

- Lables: Nhãn - đánh dấu cho một đoạn lệnh
- Orders: Lệnh
- Directives: Định hướng cho chương trình dịch
- Comments: các lời chú thích

COMPUTER ENGINEERING

- ❑ Mỗi dòng lệnh không quá 255 ký tự
- ❑ Mỗi dòng lệnh phải bắt đầu bằng 1 ký tự, nhân, lệnh hoặc chỉ thị định hướng
- ❑ Các thành phần của mỗi dòng lệnh cách biệt nhau ít nhất bằng 1 dấu cách
- ❑ Sau ";" là comments

COMPUTER ENGINEERING

Khai báo biến

Tên biến DB Giá trị khởi tạo

DATA1:	DB	2D	; Số thập phân
DATA2:	DB	00110101B	; Số nhị phân (35 ở dạng Hex)
DATA3:	DB	39H	; Số dạng Hex
DATA4 DB		"Ky thuat may tinh"	; Các ký tự ASCII

Khai báo hằng

Tên hằng EQU Giá trị

```
COUNT EQU 25  
MOV R3, #count ; Khi thực hiện lệnh "MOV R3, #COUNT"  
;thì thanh ghi R3 sẽ được nạp giá trị 25
```

COMPUTER ENGINEERING

Ký hiệu	Thực hiện	Ví dụ	Kết quả
+	Cộng	10+5	15
-	Trừ	25-17	8
*	Nhân	7*4	28
/	Chia nguyên	7/4	1
MOD	Chia lấy dư	7 MOD 4	3
SHR	Dịch phải	1000B SHR 2	0010B
SHL	Dịch trái	1010B SHL 2	101000B
NOT	Đảo	NOT 1	1111111111111110B
AND	And bit	1101B AND 0101B	0101B
OR	Or bit	1101B OR 0101B	1101B
XOR	Xor	1101B XOR 0101B	1000B
LOW	Lấy byte thấp	LOW(0AADDH)	0DDH
HIGH	Lấy byte cao	HIGH(0AADDH)	0AAH
EQ, =	So sánh bằng	7 EQ 4 or 7=4	0 (false)
NE, <>	SS Không bằng	7 NE 4 or 7<>4	0FFFFH (true)
GT, >	SS lớn hơn	7 GT 4 or 7>4	0FFFFH (true)
GE, >=	SS nhỏ hơn hoặc bằng	7 GE 4 or 7>=4	0FFFFH (true)
LT, <	SS nhỏ hơn	7 LT 4 or 7<4	0 (false)
LE, <=	SS nhỏ hơn hoặc bằng	7 LE 4 or 7<=4	0 (false)

- A-Z, 0-9 ? _
- 32 ký tự để phân biệt
- Phải bắt đầu bằng ký tự
- Không được trùng với các từ khóa

COMPUTER ENGINEERING

PC →

```
ORG (Vị trí bắt đầu con trỏ chương trình )  
....  
<đoạn chương trình chính>  
....  
<các chương trình con>  
....  
END. (Kết thúc chương trình)
```

COMPUTER ENGINEERING

PC →

Ví dụ:

```
ORG 00H ; (con trỏ chương trình bắt đầu từ 00h
LJMP MAIN ; nhảy tới vị trí có nhãn là MAIN)
; (vị trí bắt đầu chương trình chính MAIN):
ORG 0030H
MAIN:
MOV R1, #10 ; ( nạp cho R1 giá trị là 10) .
LAP1:
DJNZ R1, LAP1
END ; (Kết thúc chương trình.)
```

Nhấn:

.....

Các câu lệnh

.....

RET

Ví dụ:

```
ORG 00H
LJMP MAIN
ORG 0030H
MAIN:
MOV R1,#10
LCALL LAP1           ;gọi chương trình con
LAP1:
DJNZ R1,LAP1
RET                  ; kết thúc chương trình con
END
```

- Cấu trúc bên trong RAM
- Các bank thanh ghi, stack
- Các thanh ghi chức năng đặc biệt
- Kết nối ROM, RAM ngoài
- Các kiểu định địa chỉ

COMPUTER ENGINEERING

Kết thúc chương 2-2

