

## Các nhóm lệnh 8051

- Nhóm lệnh chuyển dữ liệu
- Nhóm lệnh số học
- Nhóm lệnh logic
- Nhóm lệnh xử lý bit
- Nhóm lệnh rẽ nhánh



## Nhóm lệnh chuyển dữ liệu<sup>2</sup>

Mnemonic	Operation	Execution Time (μs) for 12MHz operation
MOV A, <src>	A = <src>	1
MOV <dest>, A	<dest> = A	1
MOV <dest>, <src>	<dest> = <src>	2
MOV DPTR, #data16	DPTR = 16-bit immediate constant	2
PUSH <src>	INC SP : MOV "@SP", <src>	2
POP <dest>	MOV <dest>, "@SP" : DEC SP	2
XCH A, <byte>	ACC and <byte> exchange data	1
XCHD a, @Ri	ACC and @Ri exchange low nibble	1



## Truyền dữ liệu với RAM ngoài

Address width	Mnemonic	Operation	Execution Time ( $\mu$ s) for 12MHz operation
8 bits	MOVX A,@Ri	Read external RAM @Ri	2
8 bits	MOVX @Ri,A	Write external RAM @ Ri	2
16 bits	MOVX A, @DPTR	Read external RAM @DPTR	2
16 bits	MOVX @DPTR, A	write external RAM @DPTR	2



## Lệnh số học

Mnemonic	Operation	Execution Time ( $\mu$ s) for 12MHz operation
ADD A, <byte>	$A = A + \text{<byte>}$	1
ADDC A, <byte>	$A = A + \text{<byte>} + C$	1
SUBB A, <byte>	$A = A - \text{<byte>} - C$	1
INC A	$A = A + 1$	1
INC <byte>	$\text{<byte>} = \text{<byte>} + 1$	1
INC DPTR	$\text{DPTR} = \text{DPTR} + 1$	2
DEC A	$A = A - 1$	1
DEC <byte>	$\text{<byte>} = \text{<byte>} - 1$	1
MUL AB	$B:A = B \times A$	4
DIV AB	$A = \text{Int}[A/B]$ $B = \text{Mod}[A/B]$	4
DA A	Decimal adjust	1



## Lệnh xử lý logic

Mnemonic	Operation	Execution Time (μs) for 12MHz operation
ANL A, <byte>	A = A .AND. <byte>	1
ANL <byte>, A	<byte> = <byte> .AND. A	1
ANL <byte>, #data	<byte> = <byte> .AND. #data	2
ORL A, <byte>	A = A .OR. <byte>	1
ORL <byte>, A	<byte> = <byte> .OR. A	1
ORL <byte>, #data	<byte> = <byte> .OR. #data	2
XRL A, <byte>	A = A .XOR. <byte>	1
XRL <byte>, A	<byte> = <byte> .XOR. A	1
XRL <byte>, #data	<byte> = <byte> .XOR. #data	2
CLR A	A = 00H	1
CPL A	A = .NOT. A	1

## Lệnh xử lý logic (tiếp)

CLR A	A = 00H	1
CPL A	A = .NOT. A	1
RL A	Rotate ACC Left 1 bit	1
RLC A	Rotate Left through Carry	1
RR A	Rotate ACC Right 1 bit	1
RRC A	Rotate Right through Carry	1
SWAP A	Swap Nibbles in A	1



## Các lệnh nhảy không điều kiện

Mnemonic	Operation	Execution Time ( $\mu$ s) for 12MHz operation
JMP addr	Jump to addr	2
JMP @A+DPTR	Jump to A+DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1



## Lệnh nhảy có điều kiện

Mnemonic	Operation	Execution Time ( $\mu$ s) for 12MHz operation
JZ rel	Jump if A = 0	2
JNZ rel	Jump if A $\neq$ 0	2
JC rel	Jump if C = 1	2
JNC rel	Jump if C = 0	2



## Lệnh nhảy với cờ

Mnemonic	Operation	Execution Time (μs) for 12MHz operation
DJNZ <byte>,rel	Decrement and jump if not zero	2
CJNE A,<byte>,rel	Jump if A $\neq$ <byte>	2
CJNE <byte>,#data,rel	Jump if <byte> $\neq$ #data	2



## Lệnh làm việc với bit

Mnemonic	Operation	Execution Time (μs) for 12MHz operation
ANL C, bit	C = C .AND. bit	2
ANL C,/bit	C = C .AND. .NOT. bit	2
ORL C, bit	C = C .OR. bit	2
ORL C,/bit	C = C .OR. .NOT. bit	2
MOV C,bit	C = bit	1
MOV bit,C	bit = C	2
CLR C	C = 0	1
CLR bit	bit = 0	1



## Lệnh làm việc với bit (tiếp)

SETB C	C = 1	1
SETB bit	bit = 1	1
CPL C	C = .NOT. C	1
CPL bit	bit = .NOT. bit	1
JC rel	Jump if C = 1	2
JNC rel	Jump if C = 0	2
JB bit,rel	Jump if bit = 1	2
JNB bit,rel	Jump if bit = 0	2
JBC bit,rel	Jump if bit = 1; CLR bit	2



Electrical Engineering

## Lệnh làm việc với bảng dữ liệu

Mnemonic	Operation	Execution Time (μs) for 12MHz operation
MOVC A,@A+DPTR	Read Program memory at (A+DPTR)	2
MOVC A,@A+PC	Read Program memory at (A+PC)	2

MOVC, nghĩa là move constant



Electrical Engineering

12

## Ví dụ bảng tìm kiếm

```
MOV A, Index
CALL look_up
.....
look_up:  inc A
          movc A, @A + PC
          ret
TABLE:   DB data1, data2, ...
```



## Lập trình có cấu trúc

- Lập trình tuần tự (sequential programming)
- Lập trình cấu trúc
- Lập trình hướng đối tượng



## Các đặc trưng lập trình cấu trúc

- Dữ liệu + giải thuật = chương trình
- Chương trình
  - Chương trình con
  - Có 3 loại cú pháp cơ bản
    - Lệnh gán
    - Lệnh if .... then
    - Lệnh do while



## Phát biểu While/do

- while [condition] Do
- các lệnh chương trình
- do
- các lệnh chương trình
- while (...)





## Ví dụ tính tổng

- Tính tổng dãy số
- Chiều dài của dãy số chứa trong thanh ghi R7
- Địa chỉ bắt đầu dãy số trong thanh ghi R0



## Ví dụ

- $[\text{sum}] = 0$
- WHILE ( $\text{length} > 0$ ) Do
  - $[\text{sum} = \text{sum} + @\text{pointer}]$
  - $[\text{pointer} = \text{pointer} + 1]$
  - $[\text{length} = \text{length} - 1]$
- End



## Phát biểu case

```
case [ expression] of  
  0: do0  
  1: do1  
  .....  
  n: don  
  default: do_default  
End_case
```



## Ví dụ case

- Yêu cầu, chương trình thực đơn, người sử dụng nhập vào giá trị 0, 1, 2, và chương trình sẽ nhập giá trị tương ứng



## Ví dụ case (tiếp)

```
Call InChar
CJNE A, #'0', Skip1
Act0: .....
      JMP Exit
Skip1: CJNE A, #'1', Skip2
Act1: .....
      JMP Exit
.....
Exit:
      DoN
```



## Các định nghĩa phụ

- Dấu ; dùng để chú thích
  - Mov @R0, A; Lưu giá trị vào vùng đệm
- ORG
  - Org 100h, điểm xuất phát chương trình
- Equ
  - ESC Equ 27h, lệnh gán giá trị



## Các định nghĩa phụ

- DB (define byte)
  - table:DB 0,1,2,4
  - message: DB, 'Login:', 0
- DW (define word)
  - temp: DW 'A', 1342H,
- DS (define storage)
  - Length EQU 25H
  - Buffer: DS Length



## Chương trình mẫu

- Chuyển đổi ký tự sang số Hex
  - ASCII – Hex
- Bài tập về nhà
  - Hex -> ASCII

