单片机.

第1章.

1. 微处理器：CPU或MPU
   微控制器：MCU
   存储器：RAM（随机存取存储器）
   ROM（只读存储器）
   输入/输出（I/O）接口.
   总线：地址总线AB.
   数据总线DB
   控制总线CB
2. 程序存储器中连续顺序存放
3. PC会自动+1
4. 十进制（D）. 二进制（B）
   十六进制（H）
5. 机器数：01001001
   真值：+1001001
6. 带符号数：原码
   反码
   补码（默认）

   把减法变成加法运算
   [负数]补 = 100 - [绝对值]补
   如 [90]补 = 100H - 90H = 63H
   范围：-128 ~ +127
10. BCD和ASCII
11. 位：1位二进制数
    字节：8位二进制数
    字：16位二进制数.

第2章.

1. CPU — 运算器
   控制器.
2. 运算器：算术逻辑运算部件ALU
   完成各种运算（8位）
   寄存器：累加器ACC.（8位）
   寄存器B（8位）用于乘除法

PSW程序状态寄存器：

C	AC	F0	RS1	RS0	OV	-	P
D7	D6	D5	D4	D3	D2	D1	D0

C：第7位进位（进，1）
AC：半进位（低半字节向高字节进位，1）
P：奇偶（A有奇个1，1）
OV：发生溢出（1）. $OV = D_{7C} \oplus D_{6C}$
乘法：OV=1，表示超过255.
乘积分别在B、A中.
反之，存在A.
除法：OV=1 → 除数为0.
PSW.1：未定义
RS1.RS0：选择工作寄存器组
3. 控制器：程序计数器PC（16位）
   表示内容：下一条要执行的指令
   自动加1，顺序执行.
   堆栈指针SP.（8位）
   由低向高，指向栈顶
   进栈时，自动+1
   出栈时，自动-1
   数据指针DPTR.（16位）
   用于存放16位地址.
   间接寄存器（变址、基址）
   范围：64K
   高字节DPH，低字节DPL

4. 引脚：
   (1) 主电源：Vcc：+5V电源
   Vss：接地
   (2) 时钟电路：XTAL1 } ← 振荡器
   XTAL2 }
   (3) RST：复位：上电复位
   手动复位

(4). ALE：地址锁存允许输出。
以6m时钟振荡频率固定输出正脉冲。

(5) $\overline{PSEN}$：外部程序存储器读选通信号。

(6) $\overline{EA}$：外部访问允许端。

EA输入高电平，(0000H—0FFFH)
片内程序存储器 → 超过 → 片外

EA输入低电平，片外。

80引→低(无片内)

8051 → 21V

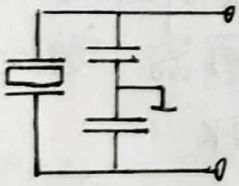
(7). P0.0～P0.7：片外 → 低8位地址
8位双向数据总线

P1.0～P1.7：EPROM编程和程序验证
→低8位地址

P2.0～P2.7：片外 → 高8位地址
EPROM ↗

P3.0～P3.7：功能与SP1相同
每个引脚可单独定义为I/O

5. 振荡

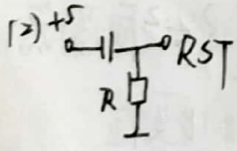


12MHz.
振荡周期：$\frac{1}{12}$ μs
时钟周期：$\frac{1}{6}$ μs
机器周期：1 μs
指令周期：1—4 μs

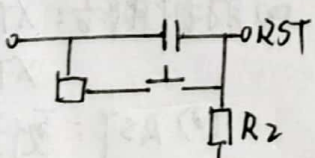
6. 复位

(1). 内部寄存器复位
Sp：07H. P0-P3：FFH，其余都是00H

(2)



上电复位.



复合复位

7. 程序存储器 ← MOVC

ROM片内	EPROM片外
0000H—0FFFH	0000H—FFFFH
$\overline{EA}$=1	$\overline{EA}$=0

8. 数据存储器 RAM.

MOV → 片内RAM
MOVX → 片外RAM.

- 00H～1FH：工作寄存器
一旦设置了其中一组的当前工作寄存器，则其余的就只能同作缓冲区。
- 20H～2FH：位寻址区.
16个单元，128位(00H—7FH)
30H～7FH：数据缓冲区
(堆栈)
80～FFH：SFR.特殊功能寄存器.

80～FFH：高端RAM.
(8052有).

第三章.

1. 寻址

(1) 立即寻址：#... 指令自动+1

(2). 直接寻址：① SFR(可以是字符地址)
② RAM低128字节

MOV A, 95H.

(3) 指明累加器时用ACC 或PUSH ACC

(3) 寄存器寻址.

MOV A, R3. (B. DPTR. CY)
(R0～R7)

(4) 寄存器间接寻址

MOV A, @R0. (R1, SP. DPTR)

(5) 变址寻址.

MOVC A, @A+DPTR(PC)
程序空间rom内容

(6) 相对寻址.

D=PC+rel
当前PC值(下一条指令首地址

(7)位寻址
SETB 3AH
片内RAM中位寻址区.
某些可位寻址的SFR区.
Cy

2. 数据传送
• MOV A, Rn
direct.
@Ri
#data.

• MOV Rn, A
direct
#data

• MOV direct, A
Rn
direct
@Ri
#data

• MOV @Ri, A
direct
#data.

• MOV DPTR, #data16

• MOVC A, @A+PC (DPTR)
64K
2字节

• 累加器A与外部数据存储器
MOVX A, @Ri (00H~FFH, 256字节)
@DPTR (64K)

MOVX @Ri, A
MOVX @DPTR, A
Ri: P0口输入/输出
DPTR: P0低8位, P2高八位.

3. 堆栈操作
PUSH direct. ; sp←sp+1, (sp)←(direct)
POP direct ; (direct)←(sp), sp←sp-1.

4. 交换:
XCH A, Rn
direct 全交换.
Ri

XCHD A, @Ri 半交换.
SWAP A 高低位交换

5. 加法
ADD A, Rn
direct
@Ri
#data.
同时刷新PSW中Cy. AC. OV. P.

ADDC A, Rn ; A←A+Rn+Cy
direct.
@Ri
#data.

INC A
Rn
direct +1.
@Ri
DPTR
不影响Cy, AC, OV.

6. 减法.
SUBB A, Rn (direct, @Ri, #data)
前要先CLR C.

DEC (A) (Rn, direct, @Ri)
影响奇偶标志位.
其他均不影响PSW.

7. 二十进制调整
DA A ; 得到BCD码
只跟在ADD和ADDC后. 不适用于减法

8. 乘法 不支持负数
MUL AB ; A×B→BA.
乘积大于255. (即B≠0). 则OV置1,
否则清0. Cy总是0

9. 除法 不支持负数.
DIV AB. ; A/B中商→A, 余数→B
B为0, OV=1. 否则为0

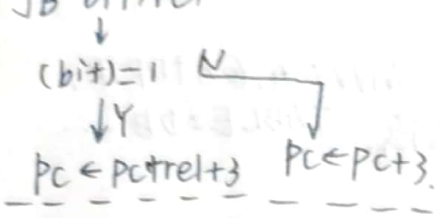
10. 逻辑运算.
(1) 与.
ANL A, Rn (direct, @Ri, #data)
ANL direct, A (#data)
可用于屏蔽某些位

(2)或
ORL A,Rn(direct,@Ri,#data)
ORL direct,A(#data)
实现对某些位置位
(3)异或指令 不进位加法
XRL A,Rn(direct,@Ri,#data)
XRL direct,A(#data)
对目的操作数的某些位取反,与1相异或
11.清零
CLR A
12.取反 仅限于A
CPL A
13.累加器A循环移位.
RL A: $D_7 \leftarrow D_0$
RR A: $D_7 \rightarrow D_0$
RLC A: $CY\ D_7 \leftarrow D_0$
RRC A: $CY\ D_7 \rightarrow D_0$
14.控制转移类指令
(1)转移类指令
①无条件转移
LJMP addr16 ;PC←addr16 64kB 3
AJMP addr11 ;PC←PC+2
2kB PC.10~PC.0←addr11 2
JMP @A+DPTR ;PC←A+DPTR A加25的五.
SJMP rel ;PC←PC+rel 2
②条件转移指令
1)判0
JZ rel
JNZ rel
JZ rel
A=0 N
Y
PC=PC+2. 2
PC=PC+2+rel
2)判CY
JC rel
JNC rel
JC rel
CY=1 N
Y
PC←PC+2
PC←PC+rel+2
3)比较转移指令
CJNE A,direct,rel
(#data)
CJNE Rn,#data,rel 3
CJNE @Ri,#data,rel
CJNE S_1,S_2,rel
S_1-S_2
$S_1 \neq S_2$ N
Y
PC←PC+rel+2
PC←PC+2
4)循环转移指令
DJNZ Rn,rel 2
DJNZ direct,rel 3
DJNZ S_1,rel
$S_1-1 \rightarrow S_1$
$S_1=0$ N
Y
PC←PC+2
PC←PC+rel+2
③调用子程序指令
LCALL addr16;
ACALL addr11;
④返回指令
RET
RETI 只用于中断服务.
15.位操作指令
①位数据传送
MOV C,bit
MOV bit,C
②位状态修改
CLR C(bit) ;清零
CPL C(bit) ;取反
SETB C(bit) ;置1
③位逻辑运算指令 CY∧(bit)
与:ANL C,bit+C/bit)

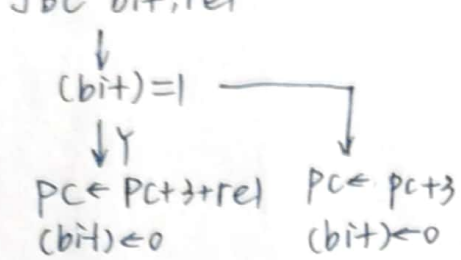
或：ORL C,bit(/bit)
没有异或

④位条件转移指令

JB bit,rel 3
JNB bit,rel



JBC bit,rel 3



第4章.

1. 程序设计语言：①机器语言
②汇编语言
③高级语言

2. 标号：①标号必须以字母开头
②不能使用指令助记符、伪指令或寄存器名来作标号
③建议使用能说明程序段功能的标号

3. 伪指令：起某种控制作用

- ORG
1) 可多次使用
2) 不能有重叠
3) 若不带操作数，则起始地址为0000H
- END：表示源程序结束，END后面的程序都不能运行

SJMP $：原地踏步

- EQU：赋值指令
把操作数段中的地址或数据赋值给字符名称

COUNT EQU 16H. ; COUNT=16H.

- DB：定义字节数据（8位）（ROM）
定义（存储）若干个字节的数据或ASCII码字符.
- DW：定义字数据（16位）（ROM）
定义（存储）若干字数据.
- DS：定义存储区
从指定的地址开始，保留一定数量的内存单元，以备使用

```
ORG 1000H
DS  5
DB  23H
```

从1000H.保留5个字节单元，即(1005H)=23H

- BIT：位地址符号指令
把位地址赋给所规定的字符名称.
- DATA：数据地址赋值指令
DATA定义的标识符在汇编时作为标号登记在符号表中，可以先使用后定义.
而EQU不可以

4. 程序

- 顺序程序.
- 分支程序：双分支.
多分支
称作转移指令和有条件转移指令.

多分支←分支表法.←利用转移指令 JMP@A+DPTR.

散转指令表：

```
MOV DPTR,#JMP_TBL
JMP @A+DPTR
JMP_TBL: AJMP LABEL1
         AJMP LABEL2
         ...
```

·循环程序 DJNZ. JB
R6. R7用于循环次数的控制
延时程序←双循环
·查表程序 MOV DPTR, #TABLE, MOVC A, @A+DPTR.
顺序查表法. 计算查表法. 对分查表法. TABLE: DB ...
·子程序 RET. LCALL