

单片机原理及应用考试复习知识点

第 1 章 计算机基础知识

考试知识点：

1、各种进制之间的转换

(1) 各种进制转换为十进制数

方法：各位按权展开相加即可。

(2) 十进制数转换为各种进制

方法：整数部分采用“除基取余法”，小数部分采用“乘基取整法”。

(3) 二进制数与十六进制数之间的相互转换

方法：每四位二进制转换为一位十六进制数。

2、带符号数的三种表示方法

(1) 原码：机器数的原始表示，最高位为符号位（0‘+’1‘-’），其余各位为数值位。

(2) 反码：正数的反码与原码相同。负数的反码把原码的最高位不变，其余各位求反。

(3) 补码：正数的补码与原码相同。负数的补码为反码加 1。

原码、反码的表示范围：-127 ~ +127，补码的表示范围：-128 ~ +127。

3、计算机中使用的编码

(1) BCD 码：每 4 位二进制数对应 1 位十进制数。

(2) ASCII 码：7 位二进制数表示字符。0 ~ 9 的 ASCII 码 30H ~ 39H，A 的 ASCII 码 41H，a 的 ASCII 码 61H。

第 2 章 80C51 单片机的硬件结构

考试知识点：

1、80C51 单片机的内部逻辑结构

单片机是把 CPU、存储器、输入输出接口、定时 / 计数器和时钟电路集成到一块芯片上的微型计算机，主要由以下几个部分组成。

(1) 中央处理器 CPU

包括运算器和控制器。

运算电路以 ALU 为核心，完成算术运算和逻辑运算，运算结果存放于 ACC 中，运算结果的特征存放于 PSW 中。

控制电路是单片机的指挥控制部件，保证单片机各部分能自动而协调地工作。程序计数器 PC 是一个 16 位寄存器，PC 的内容为将要执行的下一条指令地址，具有自动加 1 功能，以实现程序的顺序执行。

(2) 存储器

分类：

随机存取存储器 RAM：能读能写，信息在关机后消失。可分为静态 RAM (SRAM) 和动态 RAM (DRAM) 两种。

只读存储器：信息在关机后不会消失。

掩膜 ROM：信息在出厂时由厂家一次性写入。

可编程 PROM：信息由用户一次性写入。

可擦除可编程 EPROM：写入后的内容可由紫外线照射擦除。

电可擦除可编程 EEPROM：可用电信号进行清除和改写。

存储容量：

存储容量指存储器可以容纳的二进制信息量，M 位地址总线、N 位数据总线的存储器容量为 $2^M \times N$ 位。

80C51 单片机的存储器有内部 RAM(128B ,高 128B 为专用寄存器)、外部 RAM(64KB)、内部 ROM (4KB 掩膜 ROM)、外部 ROM (64KB)。

(3) 输入输出接口

4 个 8 位并行 I/O 口 (P0、P1、P2、P3)

(4) 其它资源

一个全双工串行口、5 个中断源、2 个 16 位的定时 / 计数器、时钟电路。

2、80C51 单片机的信号引脚

(1) 电源部分：VCC 接+5V、VSS 接地。

(2) 时钟电路部分：XTAL1 和 XTAL2 接晶振。

1 个机器周期 =6 个状态 =12 个拍节

6MHZ 的晶体机器周期 2us，12MHZ 的晶体机器周期 1us。

(3) I/O 口部分：P0——8 位数据总线 / 地址总线低 8 位、P1——用户口、P2——地址高 8 位、P3——第二功能。

(4) 控制部分：

地址锁存控制信号 ALE，用于控制把 P0 口输出的低 8 位地址送入锁存器锁存地起来。

外部程序存储器读选通信号 PSEN，低电平有效，以实现外部 ROM 单元的读操作。

访问程序存储器控制信号 EA，低电平时只读外部 ROM，高电平时先读内部 ROM，再读外部 ROM。

复位信号 RST，当输入的复位信号延续 2 个机器周期以上高电平时即为有效。

复位值：PC=0000H，SP=07H，P0=0FFH。

3、内部 RAM 的基本结构与功能

80C51 的内部数据存储器低 128 单元区，称为内部 RAM，地址为 00 ~ 7FH。

(1) 寄存器区 (00 ~ 1FH)

共分为 4 组，组号依次为 0、1、2、3，每组有 8 个寄存器，在组中按 R7~R0 编号。由 PSW 中 RS1、RS0 位的状态组合来决定哪一组。

(2) 位寻址区 (20H ~ 2FH)

可对单元中的每一位进行位操作，16 个字节单元共 128 个可寻址位，位地址为 00 ~ 7FH。

位起始地址 D0= (字节地址 -20H) *8

(3) 用户 RAM 区 (30H ~ 7FH) 堆栈、缓冲区

堆栈是在内部 RAM 中开辟的，最大特点就是“后进先出”的数据操作原则。

两项功能：保护断点和保护现场。两种操作：进栈和出栈。

SP 堆栈指针，它的内容就是堆栈栈顶单元的地址。

4、专用寄存器 (内部数据存储器高 128 单元)

(1) 累加器 A (ACC)

(2) 寄存器 B

(3) 程序状态字 PSW

CY——进位标志位，最高位的进位或借位。

AC——半进位标志位，低 4 位向高 4 位的进位或借位。

OV——溢出标志位，同符号数相加，结果为异符号，有溢出；异符号数相减，结果和

减数符号相同，有溢出。

P—— A 中 1 的个数，奇数个 P=1，偶数个 P=0。

(4) 数据指针 DPTR：80C51 中惟一一个供用户使用的 16 位寄存器。高 8 位 DPH，低 8 位 DPL。

第 3 章 80C51 单片机指令系统

考试知识点：

1、寻址方式

- (1) 立即寻址 (#data, #data16) 例：MOV A, #00H
- (2) 直接寻址 (direct) 内部 RAM：00 ~ 7FH、特殊功能寄存器 例：MOV A, 00H
- (3) 寄存器寻址 (A、B、Rn、DPTR)
- (4) 寄存器间接寻址 (@Ri、@DPTR) 例：MOVX A, @DPTR
- (5) 变址寻址 (@A+DPTR, @A+PC) 例：MOVC A, @A+DPTR
- (6) 位寻址 (bit) 20 ~ 2FH：00 ~ 7FH、特殊功能寄存器 例：MOV C, 00H
- (7) 相对寻址 (rel) 例：JZ rel

2、数据传送类指令

(1) 内部 RAM 数据传送指令

MOV 目的, 源; 目的 源

交换指令：

XCH A, direct/Rn/@Ri ; A 和源交换

XCHD A, @Ri ; 只换低 4 位

SWAP A ; A 的高低 4 位交换

注意：A 作目的的操作数会影响 P。

PUSH direct

POP direct

(2) 外部 RAM 数据传送指令

MOVX A, @Ri/@DPTR ; 外部地址内容 A

MOVX @Ri/@DPTR, A ; A 外部地址内容

(3) ROM 数据传送指令

MOVC A, @A+DPTR/@A+PC ; 查表指令

3、算术运算指令

(1) 加法指令

ADD/ADDC A, #data/ direct/ Rn/@Ri ; 会影响 CY、AC、OV、P

INC A/ direct/ Rn/@Ri/DPTR ; 加 1, P

DA A ; 十进制调整, 大于 9 加 6

(2) 减法指令

SUBB A, #data/ direct/ Rn/@Ri ; 会影响 CY、AC、OV、P

DEC A/ direct/ Rn/@Ri ; 减 1

(3) 乘除指令

MUL AB ; (A)*(B) BA, 会影响 CY=0, OV, P

DIV AB ; (A)/(B) 的商 A, 余数 B

4、逻辑运算及移动指令

(1) 逻辑运算指令

ANL/ORL/XRL A , #data/ direct/ Rn/@Ri

ANL/ORL/XRL direct , A/#data

与 清 0, 或 置 1, 异或 取反

CLR/CPL A ; 清 0 和取反

(2) 移位指令

RL/RR/RLC/RRC A

注意：每左移一位相当于乘 2，每右移一位相当于除 2，带进位的移会影响 CY 和 P。

5、控制转移类指令

(1) 无条件转移指令

LJMP addr16 ; addr16 PC , 64KB

AJMP addr11 ; (PC)+2 PC , addr11 PC10~0 , 2KB

SJMP rel ; (PC)+2+rel PC , 256B

JMP @A+DPTR ; (A)+(DPTR) PC , 64KB

(2) 条件转移指令

累加器 A 判 0 转移指令

JZ rel ; A 为 0

JNZ rel ; A 不为 0

比较不相等转移指令

CJNE A/Rn/@Ri , #data , rel

CJNE A , direct , rel

注意：第一操作数和第二操作数不相等，程序转移，若第一大于第二，CY=0，第一小于第二，CY=1。第一操作数和第二操作数相等，程序顺序执行，CY=0。

减 1 不为 0 转移指令

DJNZ Rn/direct , rel ; (Rn/direct) -1 不为 0，程序转移。

(3) 调用和返回指令

LCALL addr16 ; (PC)+3 PC , 先入低 8 位，再入高 8 位，addr16 PC

ACALL addr11 ; (PC)+2 PC , 先入低 8 位，再入高 8 位，addr11 PC10~0

RET ; 先出高 8 位，再出低 8 位

6、位操作类指令

(1) 位传送指令

MOV C , bit

MOV bit , C

(2) 位赋值指令

CLR C/bit

SETB C/bit

(3) 位逻辑运算指令

ANL/ORL C , bit 或/bit

CPL C/bit

注意：实现逻辑表达式

(4) 位控制转移指令

JC rel ; (CY) =1

JNC rel ; (CY) =0

JB bit , rel ; (bit)=1

JNB bit , rel ; (bit)=0
JBC bit , rel ; (bit)=1 , 转移 , 清 0

第 4 章 80C51 单片机汇编语言程序设计

考试知识点：

1、汇编语言的语句格式

【标号：】 操作码 【操作数】 【；注释】

标号：语句地址的标志符号。

操作码：语句执行的操作内容，用指令助记符表示。

操作数：为指令操作提供数据。

注释：对语句的解释说明。

2、伪指令

起始地址 ORG、结束 END、赋值 EQU、字节 DB、字 DW、空 DS、位 BIT

3、汇编语言程序的基本结构形式

(1) 顺序结构

(2) 分支结构

(3) 循环结构：数据传送问题、求和问题

4、定时程序

例：延时 100ms 的子程序，设晶振频率 6MHZ。

```
DELAY : MOV    R5 , #250
LOOP2 : MOV    R4 , #49
LOOP1 : NOP
        NOP
        DJNZ   R4 , LOOP1
        DJNZ   R5 , LOOP2
        RET
```

5、查表程序

- (1) 要查找的数据在表中的位置给 A
- (2) 表的首地址给 DPTR
- (3) MOVC A , @A+DPTR
- (4) 数据表

第 5 章 80C51 单片机的中断与定时

考试知识点：

1、中断源和中断请求标志位

中断名称	中断请求标志	中断向量
外部中断 0	IE0	0003H
T0 中断	TF0	000BH
外部中断 1	IE1	0013H
T1 中断	TF1	001BH
串行发送中断	TI	0023H
串行接收中断	RI	0023H

2、和中断相关的寄存器的设置

(1) 定时器控制寄存器 TCON

格式如下：

位	D7	D6	D5	D4	D3	D2	D1	D0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
位地址	8F	8E	8D	8C	8B	8A	89	88

IT0=0，为电平触发方式。 $\overline{\text{INT0}}$ 低电平有效。

IT0=1，为边沿触发方式。 $\overline{\text{INT0}}$ 输入脚上电平由高到低的负跳变有效。

IE0=1，说明有中断请求，否则 IE0=0。

(2) 中断允许控制寄存器 IE

其各位的定义如下：

位	D7	D6	D5	D4	D3	D2	D1	D0
代号	EA	—	—	ES	ET1	EX1	ET0	EX0

EA：开放或禁止所有中断。 ES：开放或禁止串行通道中断。 ET1：开放或禁止定时 / 计数器 T1 溢出中断。 EX1：开放或禁止外部中断源 1。 ET0：开放或禁止定时 / 计数器 T0 溢出中断。 EX0：开放或禁止外部中断源 0。

(3) 中断优先级控制寄存器 IP

各位的定义如下：

位	D7	D6	D5	D4	D3	D2	D1	D0
代号	—	—	—	PS	PT1	PX1	PT0	PX0

1 为高优先级、0 为低优先级。如果同级的多个中断请求同时出现，则按 CPU 查询次序确定哪个中断请求被响应。查询次序为：外部中断 0、T0 中断、外部中断 1、T1 中断、串行中断。

3、响应中断的必要条件

(1) 有中断源发出中断请求。

(2) 中断允许寄存器 IE 相应位置“1”，CPU 中断开放 (EA=1)。

(3) 无同级或高级中断正在服务。

(4) 现行指令执行到最后一个机器周期且已结束。若现行指令为 RETI 或需访问特殊功能寄存器 IE 或 IP 的指令时，执行完该指令且其紧接着的指令也已执行完。

中断响应的主要内容是由硬件自动生成一条长调用指令，指令格式为“LCALL addr16”。这里的 addr16 就是程序存储器中断区中相应中断的入口地址。

4、中断程序设计

(1) 在 0000H 处存放一条无条件转移指令转到主程序。

(2) 在入口地址处存放一条无条件转移指令转到中断服务子程序。

(3) 设置触发方式 (IT0/IT1)

(4) 设置 IE 和 IP。

(5) 设置 SP。

(6) 原地踏步。

(7) 中断服务子程序。最后 RETI。

5、定时计数的基本原理

(1) 定时功能：每个机器周期计数器加 1。

(2) 计数功能：T0 (P3.4) 和 T1 (P3.5) 输入计数脉冲，每一来一个脉冲计数器加 1。

6、用于定时计数的寄存器的设置

(1) 定时器控制寄存器 TCON

格式如下：

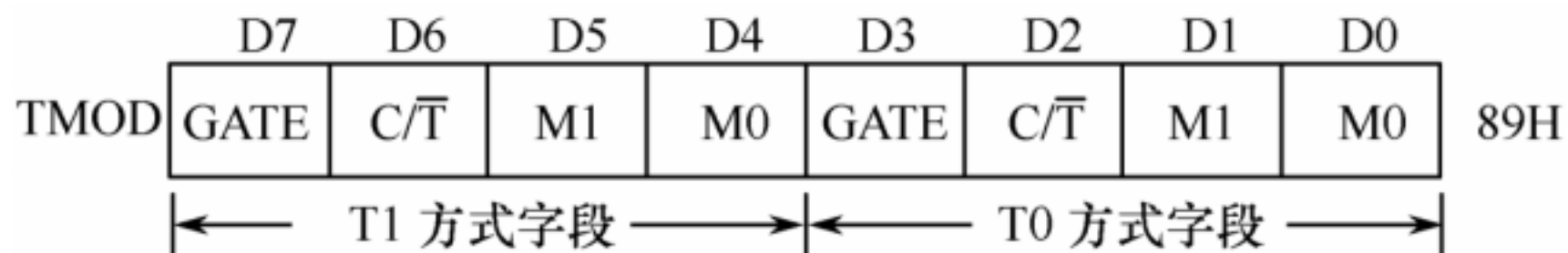
位	D7	D6	D5	D4	D3	D2	D1	D0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
位地址	8F	8E	8D	8C	8B	8A	89	88

TF1、TF0——计数溢出标志位。当计数器产生计数溢出时，由硬件置 1。采用查询方式，它是供查询的状态位。采用中断方式，作为中断请求信号。

TR1、TR0——计数运行控制位。为 1 时，启动定时器 / 计数器工作；为 0 时，停止定时器 / 计数器工作。

(2) 工作方式控制寄存器 TMOD

其格式如下：



GATE：门控位。当 GATE=1 时，同时 INTx 为高电平，且 TRx 置位时，启动定时器，外部启动。当 GATE=0 时，每当 TRx 置位时，就启动定时器，是内部启动方式。

C/T：选择定时器功能还是计数器功能。该位置位时选择计数器功能；该位清零时选择定时器功能。

M1M0：这两位指定定时 / 计数器的工作方式，可形成四种编码，对应四种工作方式：

M1	M0	方式	说明
0	0	0	TLx 低 5 位与 THx 中 8 位构成 13 位计数器
0	1	1	TLx 与 THx 构成 16 位计数器
1	0	2	可自动再装入的 8 位计数器，当 TLx 计数溢出时，THx 内容自动装入 TLx。
1	1	3	对定时器 0, 分成两个 8 位的计数器；对定时器 1, 停止计数。

7、各种工作方式计数初值计算公式

方式 0：

定时时间 $T = (8192 - \text{计数初值}) \times \text{机器周期}$

计数次数 $C = 8192 - X$

方式 1 :

定时时间 $T = (65536 - \text{计数初值}) \times \text{机器周期}$

计数次数 $C = 65536 - X$

方式 2 :

定时时间 $T = (256 - \text{计数初值}) \times \text{机器周期}$

计数次数 $C = 256 - X$

8、定时器程序设计

查询方式 :

(1) 在 0000H 处存放一条无条件转移指令, 转到主程序。

(2) 设置工作方式 TMOD。

(3) 设置计数初值。

(4) 启动定时计数。

(5) 等待时间到或计数计满。

LOOP : JBC TF0/TF1 , LOOP1

SJMP LOOP

LOOP1 : ,,

(6) 重新设置计数初值 (除方式 2), 再转第 5 步。

中断方式 :

(1) 在 0000H 处存放一条无条件转移指令, 转到主程序。

(2) 在入口地址处存放一条无条件转移指令转到中断服务子程序。

(3) 设置工作方式 TMOD。

(4) 设置计数初值。

(5) 启动定时计数。

(6) 设置 IE 和 IP。

(7) 设置 SP。

(8) 原地踏步。

(9) 中断服务子程序。重新设置计数初值 (除方式 2), 最后 RETI。

例 选用定时器 /计数器 T1 工作方式 0 产生 500 μ S 定时, 在 P1.1 输出周期为 1ms 的方波, 设晶振频率 =6MHZ。

(1) 根据定时器 /计数器 1 的工作方式, 对 TMOD 进行初始化。

按题意可设: GATE=0 (用 TR1 位控制定时的启动和停止), $\frac{C}{T}=0$ (置定时功能), M1M0=00 (置方式 0), 因定时器 /计数器 T0 不用, 可将其置为方式 0 (不能置为工作方式 3), 这样可将 TMOD 的低 4 位置 0, 所以 (TMOD) = 00H。

(2) 计算定时初值

$$(2^{13} - X) \times 2 = 500$$

$$X = 7942D = 1111100000110B$$

将低 5 位送 TL1, 高 8 位送 TH1 得: (TH1) = F8H, (TL1) = 06H

(3) 编制程序 (查询方式)

ORG 0000H

LJMP MAIN


```

        ORG 0300H
MAIN :  MOV  TMOD , #00H      ; TMOD 初始化
        MOV  TH1 , #0F8H     ; 设置计数初值
        MOV  TL1 , #06H
        SETB TR1              ; 启动定时
LOOP :  JBC  TF1 , LOOP1      ; 查询计数溢出
        AJMP LOOP
LOOP1 : CPL  P1.1              ; 输出取反
        MOV  TL1 , #06H      ; 重新置计数初值
        MOV  TH1 , #0F8H
        AJMP LOOP            ; 重复循环
        END

```

例 用定时器 /计数器 T1 以工作方式 2 计数，要求每计满 100 次进行累加器加 1 操作。

(1) TMOD 初始化

M1M0=10 (方式 2) $\overline{C/T}=1$ (计数功能)，GATE=0 (TR1 启动和停止)，因此 (TMOD) =60H。

(2) 计算计数初值

$$2^8 - 100 = 156D = 9CH \quad \text{所以 TH1}=9CH$$

(3) 编制程序 (中断方式)

```

        ORG 0000H
        AJMP MAIN          ; 跳转到主程序

        ORG 001BH          ; 定时 /计数器 1 中断服务程序入口地址
        AJMP INSERT1

        ORG 0030H
MAIN :  MOV  TMOD #60H      ; TMOD 初始化
        MOV  TL1 , #9CH    ; 首次计数初值
        MOV  TH1 , #9CH    ; 装入循环计数初值
        SETB TR1           ; 启动定时 /计数器 1
        SETB EA            ; 开中断
        SETB ET1
        SETB PT1           ; T1 为高优先级
        MOV  SP , #40H
        SJMP $              ; 等待中断
INSERT1 : INC  A
        RETI
        END

```

第 6 章 单片机并行存储器扩展

考试知识点：

1、单片机并行扩展总线的组成

- (1) 地址总线：传送地址信号
- (2) 数据总线：传送数据、状态、指令和命令
- (3) 控制总线：控制信号

2、80C51 单片机并行扩展总线

- (1) 以 P0 口的 8 位口线充当低位地址线 / 数据线
- (2) 以 P2 口的口线作高位地址线
- (3) 控制信号：

使用 ALE 作地址锁存的选通信号，以实现低 8 位地址锁存。

以 PSEN 信号作为扩展程序存储器的读选通信号。

以 EA 信号作为内外程序存储器的选择信号。

以 RD 和 WR 作为扩展数据存储器和 I/O 端口的读 / 写选通信号。

3、单片机并行存储器扩展的方法

各种外围接口电路与单片机相连都是利用三总线实现。

- (1) 地址线的连接

将外围芯片的低 8 位地址线 (A7~A0) 经锁存器与 P0 口相连，高 8 位地址线 (A15~A8) 与 P2 口相连。如果不足 16 位则按从低至高的顺序与 P0、P2 口的各位相连。

- (2) 数据线的连接

外围芯片的数据线 (D7~D0) 可直接与 P0 口相连。

- (3) 控制线的连接

ROM：OE—PSEN

RAM：OE—RD、WE—WR

片选信号 CE 的连接方法：

- (1) 接地，适用于扩展一块存储器芯片。
- (2) 线选法