

《单片机》考试题标准答案

2009 年 1 月 7 日

一、概念填空题（共 30 分，15 小题，每小题 2 分）

- 1、与单片机配套的晶振为 12MHz 时，则一个时钟周期（节拍 P）是（1） μs 、一个状态周期 S 为（2） μs 。
- 2、单片机的 P₀、P₁、P₂ 和 P₃ 四个端口作为通用 I/O 口使用时，（P₁）为双向 I/O 口，其余的（P₀、P₂ 和 P₃）为准双向 I/O 口。
- 3、单片机通过 TXD 向外串行地（发送）数据，通过 RXD 从外串行地（接收）数据。
- 4、对于 80C51 单片机，当 $\overline{\text{EA}} = (1)$ 时或在程序执行时当 PC 值超过（0FFFH 或 FFFH）H 时，就去执行外接 ROM 内的程序代码。
- 5、单片机内部 RAM 的 20H~（2FH）H 单元，既可作为一般 RAM 单元使用，进行“字节操作”；也可以对单元中的每一位进行“位操作”；位地址为 00H~（7FH）H。
- 6、CY 是（进 / 借位标志位）、TCON 是（定时器控制寄存器）、TMOD 是（定时器方式寄存器）、DPTR 是（16 位的地址指针寄存器）。
- 7、51 单片机的复位引脚是（RST），是（高）有效。
- 8、DPTR 可以分为 2 个 8 位的寄存器使用，它们是（DPH）和（DPL）。（注：DPH 和 DPL 前后顺序无所谓）
- 9、LCALL 和 LJMP 指令的区别是（LCALL 是长调用指令，往往需要以一个 RET 指令返回调用处，返回点只有一处；LJMP 是长跳转指令，往往是以另一条或多条跳转指令指令跳转到需要的地方，返回点根据条件可能有多处；注：学生答对意思即可！）。
- 10、单片机通过三总线与外部接口设备互联沟通信息，它们是（数据）总线、（地址）总线和（控制）总线，其中（地址）总线是 16 位的。（注：数据、地址、控制三个答空的顺序先后无所谓，只要答出即可）
- 11、在 51 系列单片机中，外部（ROM）和外部（RAM）是分别独立排址的，二者各占 64KB 的空间。（注：ROM 和 RAM 前后顺序无所谓）
- 12、在地址信号配合下，外部 ROM 是采用（PSEN），而外部 RAM 是采用（RD）信号来读出内容。（注：前后顺序无所谓，或答：PSEN 和 RD 也可）
- 13、51 系列有三类五个中断源，分别是（外部）中断 2 个、（内部定时计数器）中断 2 个和（串行口）中断 1 个。（注：答的前后顺序无所谓）
- 14、若要以运行控制位 TR0（或 1）来启动定时/计数器 0（或 1）时，需要让 GATE=（0）；若要以中断请求信号 $\overline{\text{INT0}}$ （或 $\overline{\text{INT1}}$ ）来启动定时/计数器 0（/1）时，需要让 GATE=（1）。
- 15、8255A 是可编程芯片，它有（3）个可编程的（8）位的并行接口。

二、基础知识题（共 20 分，5 小题，每小题 4 分）

1、8051 内部主要由那几部分组成？

答：主要由：中央处理单元（CPU）、程序存储器（ROM）、数据存储器（RAM）、定时 / 计数器（Timer / Counter）、并行接口（PIO）、串行接口（SIO）和中断控制系统（ICS）等几大部分组成。

（注：学生回答汉字或英文一种答案即可）

2、下表是对 P3 口的 $\overline{\text{INT0}}$ 、 $\overline{\text{WR}}$ 、RXD 和 T0 四个口线（引脚）第二功能的描述，请填写完整。

引脚位号	引脚名称	第二功能含义
P3.2	$\overline{\text{INT0}}$	外部中断申请端 0（输入、低有效）
P3.6	$\overline{\text{WR}}$	写外设控制信号端（输出、低有效）
P3.0	RXD	串行口接收数据端（输入）
P3.4	T0	外部计数脉冲端 0（输入）

（注：含义一栏说的意思对即可，不用一字不差！）

3、程序状态字 PSW 各位如下表，请简要解释 PSW.7、PSW.6、PSW.4、PSW.3 各位的功能含义。

位序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	F0	RS1	RS0	OV	—	P

答：

CY：进位/借位标志位。若 ACC 在运算过程中发生了进位或借位，则 CY=1；否则=0。它也是布尔处理器的位累加器，可用于布尔操作。

AC：半进位/借位标志位。若 ACC 在运算过程中，D3 位向 D4 位发生了进位或借位，则 CY=1，否则=0。机器在执行“DA A”指令时自动要判断这一位。

RS1 和 RS0：工作寄存器组选择位。

RS1RS0 = 0 0 则选择了工作寄存器组 0 区 R0~R7 分别代表 00H ~07H 单元。

RS1RS0 = 0 1 则选择了工作寄存器组 1 区 R0~R7 分别代表 08H ~0FH 单元。

RS1RS0 = 1 0 则选择了工作寄存器组 2 区 R0~R7 分别代表 10H ~17H 单元。

RS1RS0 = 1 1 则选择了工作寄存器组 3 区 R0~R7 分别代表 18H ~1FH 单元。

4、解释伪指令：ORG、DB、EQU 和 BIT 的含义？

ORG：起始地址定位伪指令。用于规定目标程序段或数据块的起始地址，设置在程序段或数据块的开始处。

DB：定义字节数据伪指令。注：下面括号中内容可以不写，也算对的。

（格式：[标号:] DB 字节数据表

功能：字节数据表可以是 1 个或多个字节数据、字符串或表达式，它表示将字节数据表中的数据从左到右依次存放在指定地址单元。）

EQU：赋值伪指令。告诉汇编程序，将汇编语句操作数的值赋予本语句的标号。

BIT：位定义伪指令

5、IE 的各位如下表所示，请简要解释 EA、ES、ET0、EX0 各位的功能含义。

位地址	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H
位符号	EA	—	—	ES	ET1	EX1	ET0	EX0

EA：总的中断允许控制位（总开关）：EA=0 时禁止全部中断；EA=1 时允许中断。

ES：串行口的中断允许控制位：ES=0 时禁止中断；ES=1 时允许中断。

ET0：Timer0 的中断允许控制位：ET0=0 时禁止中断；ET0=1 时允许中断。

EX0：INT0 的中断允许控制位：EX0=0 时禁止中断；EX0=1 时允许中断。

三、汇编程序分析题（共 20 分，4 小题，每小题 5 分）

1、分析下面程序段，加上必要的注释，最后说明该程序段完成的功能。

```
MOV    DPTR,    #2000H    ;地址指针指向数据块首地址
MOV    R7,      #50      ;数据块长度为 50
CLR    A        ;要写入的数据为 0
LOOP0: MOVX    @DPTR, A    ;将数据写入
INC    DPTR      ;
DJNZ   R7,      LOOP0    ;未完，继续
LOOP1: SJMP    LOOP1      ;结束，等待
```

功能：将首地址为 2000H，长度为 50 的数据块全部写入数据“00H”。

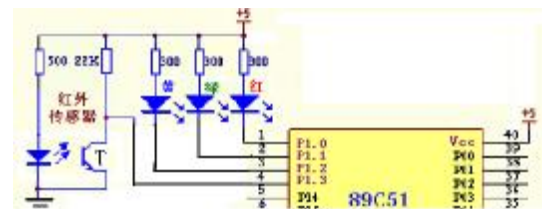
2、分析下面程序段，加上必要的注释，最后说明该程序段完成的功能。

```
MOV    R0,    #40H    ;数据指针指向内部 RAM 的 40H 单元
MOV    A,     @R0    ;从 40H 单元取出数据
MOV    R7,    A    ;将该数据暂存到 R7 中
SWAP   A        ;将该数据的高低半字节互换
ANL    A,     #0FH    ;取（或屏蔽）出该数据的高半字节
INC    R0      ;数据指针指向内部 RAM 的 41H 单元
MOV    @R0,   A    ;原数据的高半字节送 41H 单元保存
MOV    A,     R7    ;重取原数据
ANL    A,     #0FH    ;取（或屏蔽）出该数据的低半字节
INC    R0      ;数据指针指向内部 RAM 的 42H 单元
MOV    @R0,   A    ;原数据的低半字节送 42H 单元保存
SJMP   $        ;结束，等待
```

功能：将一个数据的高低两个半字节拆分成两个字，分别存放。例如，设 40H 中数据为 56H，执行完该段程序后，41H 单元为 05H，42H 单元为 06H。

3、给程序加上必要的注释，并结合电路分析其完成的功能

```
JOB3:  SETB   P1.0    ; 红色 LED 灭
        CLR   P1.1    ; 绿色 LED 亮（表示正常）
REDO:  SETB   P1.3    ; 拉高检测口线
CHECK: JNB    P1.3, JOB3; 判断光线是否被遮断
LOOP:  CLR    P1.0    ; 若被遮断，使红色 LED 亮
        SETB  P1.1    ; 绿色 LED 灭，表示有人侵入
        AJMP  REDO    ; 循环，继续检测
```



4、设晶振=6MHz，给程序加上必要的注释，并分析其完成的功能

```
ORG    0000H
AJMP   MAIN    ; 转入主程序
ORG    001BH
AJMP   TINT     ; 转入中断服务子程序
MAIN:  MOV     TMOD, #10H    ; 设定 T1 为工作方式 1
        MOV     TH1, #3CH    ; 赋初值 3CB0H=15536（硬件定时 100ms）
        MOV     TL1, #0B0H   ;
        MOV     30H, #10     ; 设定软计数值为 10 次
        CLR     F0           ; 利用标志位 F0 作为 1s 定时到标志，开始时将 F0 清零
        SETB    ET1          ; 允许 T1 中断
```

```

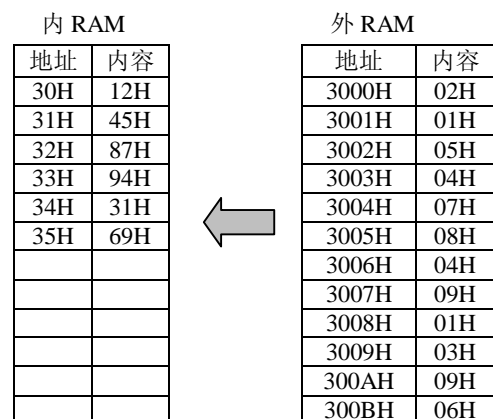
SETB EA          ; 开（或允许）总中断
SETB TR1         ; 允许 T1 开始计数
HERE: JBC F0, TIMEUP ; 在此循环等待中断和 1s 定时到，到了转 TIMEUP 去做 1s 定时相应工作
AJMP HERE
TIMEUP:(do something) ; 1s 定时到了以后要做的事情
AJMP HERE
ORG 2000H        ; 中断服务程序，100ms 进入一次
TINT: DJNZ 30H, RTN ; 若中断不到 10 次就返回
SETB F0          ; 已经中断够 10 次了“置 1”标志位
MOV 30H, #10     ; 为下一个 1 秒定时做准备
RTN: MOV TH1, #3CH ; 再赋定时 100ms 的初值 3CB0H
MOV TL1, #0B0H
RETI             ; 中断返回
END

```

四、汇编程序编程和设计题（共 15 分）

1（本小题 4 分）、如右图，请编程将外 RAM 中 3000H 开始的 12 个单元中的松散各式的 BCD 码变换（拼合）成六个紧凑格式的 BCD 存放在内 RAM 的 30H~35H 单元中。

判卷注意：复习题给定的转换方向是内 RAM→外 RAM，而此处考试题的转换方向是内 RAM←外 RAM，学生可能不注意而死记硬背照抄。那样就只能给一半分数：2 分。



参考答案 1:

```

NRAMtable EQU 30H
WRAMtable EQU 3000H
DataLength EQU 6 ;或 06H
Main_S_J: MOV R0, #NRAMtable ;指针 2 指向紧凑 BCD 码数据首址
          MOV DPTR, #WRAMtable ;指针 1 指向松散 BCD 码数据首址
          MOV R7, #DataLength ;数据长度作为计数值
Main_S_J0: MOVBX A, @DPTR ;取一个要转换的松散 BCD 码数据
          MOV B, A ;暂存该数据（保护该值）
          INC DPTR ;源指针加一，指向松散格式下一位
          MOVBX A, @DPTR ;再取一个相邻的要转换的松散 BCD 码数据
          SWAP A ;该数据的高低半字节交换
          ORL A, B ;两个松散 BCD 码数据合并成一个紧凑 BCD 码数据
          MOV @R0, A ;保存转换后的紧凑 BCD 码数据
          INC R0 ;目的指针加 1，准备保存下一个紧凑 BCD 码数据
          INC DPTR ;源指针再加一，指向松散格式 BCD 码下一位
          DJNZ R7, Main_S_J0 ;未转换完继续
          RET ;转换结束退出

```

参考答案 2:

```

NRAMtable EQU 30H
WRAMtable EQU 3000H
DataLength EQU 6 ;或 06H
Main_S_J: MOV R0, #NRAMtable ;指针 2 指向紧凑 BCD 码数据首址

```

	MOV	DPTR,	# WRAMtable	;指针 1 指向松散 BCD 码数据首址
	MOV	R7,	# DataLength	;数据长度作为计数值
Main_S_J0:	LCALL	SBCD_JBCD		;将二个松散格式 BCD 码转换成一个紧凑 BCD 码
	DJNZ	R7,	Main_S_J0	;未转换完继续
	RET			;全部转换结束返回
SBCD_JBCD:	MOVX	A, @DPTR		;取一个要转换的松散 BCD 码数据
	MOV	B,	A	;暂存该数据（保护该值）
	INC	DPTR		;源指针加一，指向松散格式下一位
	MOVX	A, @DPTR		;再取一个相邻的要转换的松散 BCD 码数据
	SWAP	A		;该数据的高低半字节交换
	ORL	A,	B	;两个松散 BCD 码数据合并成一个紧凑 BCD 码数据
	MOV	@R0,	A	;保存转换后的紧凑 BCD 码数据
	INC	R0		;目的指针加 1，准备保存下一个紧凑 BCD 码数据
	INC	DPTR		;源指针再加一，指向松散格式 BCD 码下一位
	RET			;一次转换结束返回

2（本小题 4 分）、若单片机的晶振频率为 6 MHz，从 P1.0 输出周期为 1ms 的连续方波，定时器用 T0 工作于方式 2，试编程实现。

答案：

```

ORG 0000H
AJMP MAIN      ;转主程序
ORG 000BH      ;T0 的中断矢量
CPL P1.0       ;中断服务：P1.0 取非，形成方波
RETI           ; 中断返回
MAIN: MOV TMOD, #02H ; 设定 T0 为工作方式 2
MOV TH0, #6    ; 方波半周期为 500μS 的定时初值设置
MOV TL0, #6    ;
SETB ET0       ; 允许 T0 中断
SETB EA        ; 允许总中断
SETB TR0       ; 开始计时
HERE: AJMP HERE ; 原地等待定时中断
END

```

3（本小题 7 分）、三位七段 LED 串行静态数据显示驱动电路设计与编程

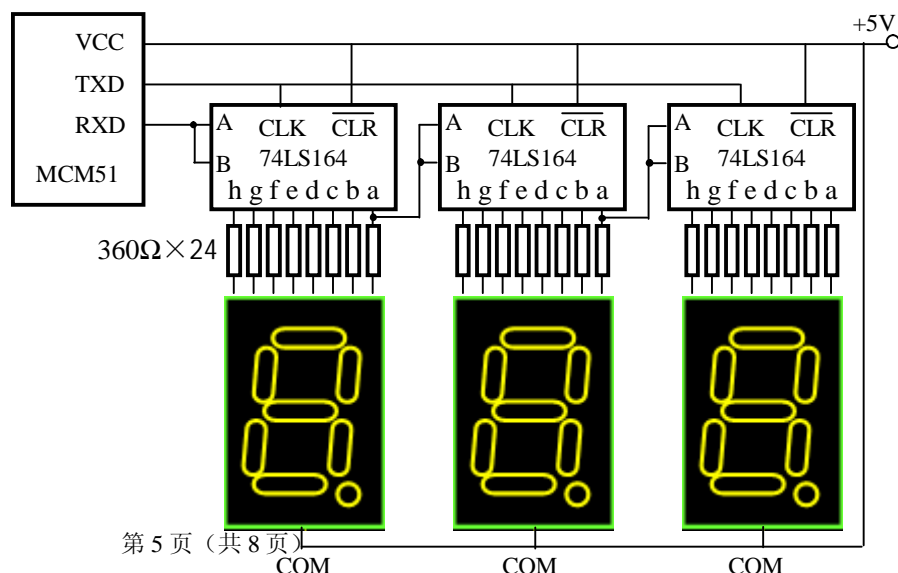
（1）硬件设计（画出电路图）：采用三个 74LS164（通用移位寄存器）芯片并利用单片机串行口（RXD 和 TXD）设计一个三位静态的 LED 显示驱动电路（LED 为共阳极，每段额定电流为 10mA）。

（2）软件编程：配合该硬件系统编制软件程序（采用 51 汇编语言）。

答：

（1）硬件电路设计（3 分）如图。

注意：学生不画电阻也算对了不扣分，其余酌情均分。



a、显示子程序（2分）:

MOVC A, @A+DPTR ;将 BCD 码转换成字形码

MOV	SBUF, A	:通过串行口输出显示字形码数据
-----	---------	-----------------

```
JNB      TI, $          ;串行输出完毕么？， 未完继续
```

CLR TI ;完毕,清 TI

RET	;返回
-----	-----

b、主程序 2 分

```
XSLED: MOV     SCON, #00H    ;串行口设置为工作方式0
```

MOV R0, #30H ;指针指向三个 BCD 码存储区首址

MOV R7, #3 ;传送三个数据

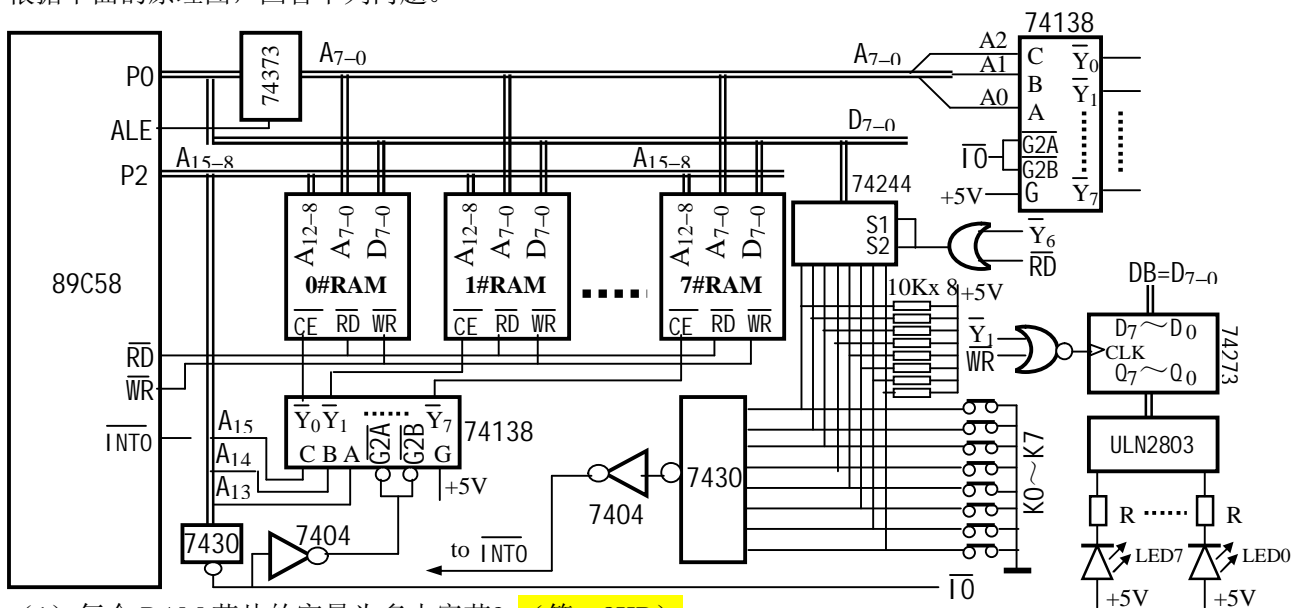
XSLED0:MOV A, @R0 ;从 BCD 码存储区取一个数据

LCALL DSPLY ; 通过串行口输出该显示字形码数据

DJNZ R7, XSLED0 ;三个数据输出完毕么？未完继续！

SJMP \$;完事，等待

根据下面的原理图，回答下列问题。



- (1) 每个 RAM 芯片的容量为多少字节？（答：8KB）
- (2) 整个 RAM 的地址范围是从多少到多少（16 进制）？（答：0000H~FEFFH，即前 255 页）
- (3) 3#RAM 的地址范围是从多少到多少（16 进制）？（答：6000H~7FFFH）
- (4) 6#RAM 的地址范围是从多少到多少（16 进制）？（答：C000H~DFFFH）
- (5) 整个 I/O 口的地址范围是从多少到多少（16 进制）？（答出：FF00H~FFFFH 即可）

可取其中的：1111111100000000B~1111111100000111B（16 进制的 FF00H~FF07H）

的地址范围是 64KBRAM 空间的最后一页：**FF00H~FFFFH**。

- (6) 键盘接口 (\overline{Y}_6) 的地址是多少? (答出: FF06H 即可)

(7) 8 个发光二极管 (LED7~LED0) 的驱动接口 ($\overline{Y_1}$) 的地址是多少? (答出: FF01H 即可)

本质上为：11111111XXXX001B，可取为：1111111100000001B，即 16 进制的 FF01H

(8) 计算和确定发光二极管的限流电阻 R 的阻值是多少? (设 LED 导通压降 $U_D=1.65V$, 额定工作电流为 $10mA$, ULN2803 内部的 OC 门反相器输出低电平为 $U_{ces}=0.35V$) (给出公式填入数据, 得出结果即可)

根据回路电压定律: $I_D * R + U_D + U_{ces} = VCC$, 即 $10mA * R = VCC - (U_D + U_{ces})$

$$\text{有: } R = \frac{V_{CC} - (U_D + U_{ces})}{I_D} = \frac{[5 - (1.65 + 0.35)](V)}{10(mA)} = \frac{3V}{10mA} = 300\Omega$$

(9) 编写一段键盘(状态)数据的读入程序(NT0中断服务程序)的主要程序部分以说明如何响应某按键按下并将其状态数据读入累加器 A 的过程。

答: 主程序初始化中相应部分不用编写, 仅写出中断入口和中断服务子程序中的主要部分(即中断服务子程序中的现场保护和恢复现场的两段均不用写)

```

KeyPort EQU 0FF06H
Key EQU 03H
...
ORG 0003H
LJMP KeyInput
...
ORG 1000H
KeyInput: ...
... ; 保护现场
MOV DPTR, # KeyPort ; 指向键盘接口
MOVBX A, @DPTR ; 读取键值
MOV Key, A ; 键值存内 RAM 的 Key 中, 供后续处理用
...
... ; 恢复现场
RETI

```

如果要求完整的程序, 则如下:

```

KeyPort EQU 0FF06H
Key EQU 03H
ORG 0000H
LJMP MAIN
ORG 0003H
LJMP KeyInput
ORG 2000H
MAIN: SETB EX0 ; 允许 INTO 中断
      SETB EA ; 允许总中断, 该两句亦可用 MOV IE, #81H 完成
WAIT: NOP ; 等待键盘中断
      SJMP WAIT
      ORG 2100H
KeyInput: ; 键盘中断服务子程序
          PUSH PSW
          PUSH DPH
          PUSH DPL
          PUSH ACC ; 保护现场
          MOV DPTR, # KeyPort ; 指向键盘接口
          MOVBX A, @DPTR ; 读取键值
          MOV Key, A ; 键值保存到 RAM 的 Key 中, 供后续处理用

```


POP	ACC	;恢复现场
POP	DPL	
POP	DPH	
POP	PSW	
RETI		