

16.00元

# 《嵌入式系统及应用》

本试卷共 4 页； 考试时间 110 分钟；

专业 \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分

## 一、填空题（20 分，每空 1 分）

- MCS-51 系列单片机为 8 位单片机。
- MCS-51 的堆栈只可设置在 片内数据 RAM ，堆栈寄存器 SP 是 8 位寄存器。
- 80C51 有 2 级中断， 5 个中断源。
- 定时器/计数器的工作方式 3 是指将 T0 拆成两个独立的 8 位计数器。而另一个定时器/计数器此时通常只可作为 波特率发生器 使用。
- MCS-51 布尔处理机的存储空间地址范围是 0H~7FH 。
- 在 80C51 单片机中，由 2 个振荡周期组成 1 个状态，由 6 个状态组成 1 个机器周期。
- 对于 80C51 无嵌套的单级中断，响应时间至少 3 个机器周期，最多 8 个机器周期。
- MCS-51 单片机有 4 个并行输入/输出口，当系统扩展外部存储器或扩展 I/O 口时， P0 口作地址低 8 位和数据传送总线， P2 口作地址总线高 8 位输出， P3 口的相应引脚会输出控制信号。
- 计算机有 CISC 和 RISC 两种类型，以 ARM 微处理器为核心的计算机属于 RISC 类型，其指令长度是定长的。
- ARM 处理器有两种总线架构，数据和指令使用同一接口的是 冯·诺伊曼 结构，数据和指令分开使用不同接口的是 哈佛结构 。
- 当使用 8031 单片机时，需要扩展外部程序存储器，此时 EA 应为 低电平 。

得分

## 二、选择题 (10 分, 每题 2 分)

1. MCS-51 单片机的复位信号是 A 有效。

- A. 高电平      B. 低电平      C. 脉冲      D. 下降沿

2. 若 MCS-51 单片机使用晶振频率为 6MHz 时, 其复位持续时间应该超过 B。

- A. 2 $\mu$ s      B. 4 $\mu$ s      C. 8 $\mu$ s      D. 1ms

3. 若 PSW.4=0, PSW.3=1, 要想把寄存器 R0 的内容入栈, 应使用 D 指令。

- A. PUSH R0      B. PUSH @R0      C. PUSH 00H      D. PUSH 08H

4. 定时器/计数器工作方式 1 是 D。

- A. 8 位计数器结构      B. 2 个 8 位计数器结构  
C. 13 位计数结构      D. 16 位计数结构

5. 下面哪点不是嵌入式操作系统的特点。 C

- A. 内核精简      B. 专用性强  
C. 功能强大      D. 高实时性

得分

## 三、简答题 (16 分, 每题 4 分)

1. 中断服务子程序与普通子程序有何异同之处?

答: 当中断产生的时候进入中断服务程序, 不需要调用; 而普通子程序只有被调用了才能执行。

2. MCS-51 单片机片内 256B 的数据存储器可分为几个区? 分别作什么用?

答:

4 个区

工作寄存器区: 从 00H~1FH 安排了 4 组工作寄存器, 每组占用 8 个 RAM 字节, 记为 R0~R7; 位寻址区: 地址从 20H~2FH, 共 16 字节, 128 位; 用户 RAM 区: 地址 30H~7FH, 共 80 字节, 这是正在给用户使用的一般 RAM 区, 该区主要用来存放随机数据和运算的结果, 另外也常常把堆栈开辟在该区域中; 剩下的区域 80H~FFH, 存放 21 个特殊功能寄存器, 它们离散分部在该区域中, 未占用的地址单元无定义, 用户不可以使用, 如果对未定义单元进行读/写操作, 得到的是随机数, 而写入的数据将会丢失。

3. 简述 80C51 单片机指令系统的寻址方式。

答: 立即数寻址; 直接寻址; 寄存器寻址; 寄存器间接寻址; 相对寻址; 变址寻址; 位寻址

4. MCS-51 外扩的程序存储器和数据存储器可以有相同的地址空间, 但不会发生数据冲突, 为什么?

答: 不发生数据冲突的原因是:

MCS-51 中访问程序存储器和数据存储器的指令不一样;

程序存储器访问指令为 MOVC;

数据存储器访问指令为 MOVX;

选通信号不同,前者为/PSEN,后者为/WR 与/RD。

得分

#### 四、程序分析题 (30 分, 每空 2 分)

1. 执行下列程序段中第一条指令后, (1)(P1.7)= 0 (P1.3)= 0 ,

(P1.2)= 0 ; 执行第二条指令后, (2)(P1.5)= 1 ,

(P1.4)= 1 , (P1.3)= 1 。

```
ANL P1, #73H
```

```
ORL P1, #38H
```

2. 下列程序段执行后, (A)= 0EH , (B)= 00H 。

```
MOV A, #0FCH
```

```
MOV B, #12H
```

```
DIV AB
```

3. 下列程序段执行后, (R0)= 7FH , (7EH)= 00H , (7FH)= 41H 。

```
MOV R0, #7FH
```

```
MOV 7EH, #0
```

```
MOV 7FH, #40H
```

```
DEC @R0
```

```
DEC R0
```

```
DEC @R0
```

4. 已知(SP)=09H, (DPTR)=4567H, 在执行下列指令后, (SP)= 0BH , 内部

RAM(0AH)= 67H , (0BH)= 45H 。

```
PUSH DPL
```

```
PUSH DPH
```

5. 下列程序中注释的数字为执行该指令所需的机器周期数, 若单片机的晶振频率为

6MHz, 问执行下列程序需要时间为 1006 微秒 。

```
MOV R3, #100; 1
```

```
LOOP: NOP ; 1
```

```
NOP
```

```
NOP
```

```
DJNZ R3, LOOP ; 2
```

```
RET ; 2
```

得分

### 五、设计题 (24 分)

1、编写一段子程序 将二位压缩的 BCD 码转换为二进制数 入口、出口均是 A。若是非法的 BCD 码 则 A 返回值为 255。 共 8 分

<pre> SUBP: MOV R1  A       ANL A  #0F0H       SWAP A       CJNE A, #10, NEXT1       LJMP ERROR NEXT1: JNC ERROR       MOV B  #10       MULAB       XCH A  R1       ANLA  #0FH       CJNE A  #10  NEXT2       LJMP ERROR           </pre>	<pre> NEXT2: JNC ERROR       ADD A  R1       RET ERROR       MOVA  #255       RET           </pre>
---	--

2、要求使用定时器/计数器实现在 P1.0 引脚上产生周期为 4ms 的方波输出，已知单片机晶体振荡器的频率为  $f_{osc}=12\text{MHz}$ ，请使用定时器/计数器 T0 的方式 0。(16 分)

(1) 计算求解出定时常数 TC?

(2) 根据计算结果，编写程序在 P1.0 引脚上产生周期为 4ms 的方波输出。

方式寄存器 TMOD:

GATE	C/T	M1	M0	GATE	C/T	M1	M0
------	-----	----	----	------	-----	----	----

1)  $T_c = 4192$

2) ORG 0000H

AJMP MAIN

ORG 000BH

AJMP INQP

ORG 0030H

MAIN: MOV TMOD, #00H

MOV TH0, #04H

MOV TL0, #30H

SETB TR0

SETB ET0

SETB EA

AJMP \$

ORG 2000H

INQP: MOV TH0, #04H;

MOV TL, #30H

CPL P1.0

RETI

# 《嵌入式系统及应用》

本试卷共 4 页； 考试时间 110 分钟；

专业 \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分

## 一、填空题（20 分，每空 1 分）

1. MCS-51 系列单片机为 8 位单片机。
2. 80C51 有 2 级中断， 5 个中断源。
3. 定时器/计数器的工作方式 3 是指将 T0 拆成两个独立的 8 位计数器。而另一个定时器/计数器此时通常只可作为 波特率发生器 使用。
4. 在 80C51 单片机中，由 2 个振荡周期组成 1 个状态，由 6 个状态组成 1 个机器周期。
5. MCS-51 单片机有 4 个并行输入/输出口，当系统扩展外部存储器或扩展 I/O 口时，P0 口作地址低 8 位和数据传送总线，P2 口作地址总线高 8 位输出，P3 口的相应引脚会输出控制信号。
6. 在中断服务程序中现场保护和现场恢复期间，中断系统应处在 关中断 状态。
7. 当使用 8031 单片机时，需要扩展外部程序存储器，此时 EA 应为 低电平 。
8. 根据嵌入式系统使用的微处理器，可以将嵌入式系统分为 嵌入式微处理器 ，嵌入式 DSP，嵌入式微控制器 以及片上系统。
9. 哈佛体系结构数据空间和地址空间分开存储，ARM920T 采用 哈佛结构 的内核架构。
10. 按操作系统的分类可知，Dos 操作系统属于顺序执行操作系统，Unix 操作系统属于分时操作系统，VxWorks 属于 嵌入式实时 操作系统。
11. ARM7TDMI 采用 3 级流水线结构，ARM920TDMI 采用 5 级流水线。
12. ARM7TDMI 中，T 表示支持 16 位 Thumb 指令集，D 表示 JTAG 调试器，M 表示快速乘法器，I 表示嵌入式跟踪宏单元，支持在线断点和调试。

得分

## 二、选择题（10 分，每题 2 分）

1. MCS-51 单片机的复位信号是 A 有效。  
A. 高电平 B. 低电平 C. 脉冲 D. 下降沿
2. 若 PSW.4=0, PSW.3=1，要想把寄存器 R0 的内容入栈，应使用 D 指令。  
A. PUSH R0 B. PUSH @R0 C. PUSH 00H D. PUSH 08H
3. 已知 1 只共阴极 LED 显示器，其中 a 笔段为字形代码的最低位，若需显示数字 1，它的字形代码应为 A。

- A.06H      B.F9H      C.30H      D.CFH
- 4.下面 B 操作系统不属于商用操作系统。
- A. windows xp      B. Linux      C. VxWorks      D. WinCE
5. 在嵌入式 ARM 处理器中, 下面 A 中断方式优先级最高。
- A. Reset      B. 数据中止      C. FIQ      D. IRQ

得分

### 三、简答题 (16 分, 每题 4 分)

1. 中断服务子程序与普通子程序有何异同之处?

答:

当中断产生的时候进入中断服务程序,不需要调用;而普通子程序只有被调用了才能执行。

2. ARM 微处理器对 IRQ 的中断响应过程

答: 当发生异常时, ARM 处理器对异常中断的响应过程如下: (1) 将 CPSR 的内容保存到将要执行的异常中断模式的 SPSR 中。此时, 异常的类型为 IRQ, 则 SPSR\_IRQ=CPSR。 (2) 设置当前程序状态寄存器 CPSR 中的模式字段位。即 CPSR[4:0]=0b10011 (3) 将异常发生时程序的下一条指令地址保存到新的异常模式的 R14 (也就是 LR) 寄存器。 (4) 强制对程序计数器赋值, 使程序从异常所对应的向量地址开始执行中断服务子程序。

3. MCS-51 外扩的程序存储器和数据存储器可以有相同的地址空间, 但不会发生数据冲突, 为什么?

答: 不发生数据冲突的原因是:

MCS-51 中访问程序存储器和数据存储器的指令不一样:

程序存储器访问指令为 MOVC;

数据存储器访问指令为 MOVX;

选通信号不同,前者为/PSEN,后者为/WR 与/RD。

4. ARM 处理器的工作模式有几种

答 1 用户模式 usr      2 快速中断模式 fiq      3 外部中断模式 irq      4 操作系统保护模式 svc 或 管理模式      5 数据访问中止模式 abt      6 处理未定义指令的未定义模式 und

得分

### 四、程序分析题 (30 分, 每空 2 分)

1. 执行下列程序段中第一条指令后, (1)(P1.7)=0 (P1.3)=0, (P1.2)=0;

执行第二条指令后, (2)(P1.5)=1, (P1.4)=1, (P1.3)=1。

ANL P1, #73H

ORL P1, #38H

2. 下列程序段执行后, (A)=8BH, (CY)=  不影响  。

MOV A #C5H

RL A

3. 下列程序段执行后 (R0)=7FH, (7EH)=00H, (7FH)=41H。

MOV R0 #7EH

MOV 7EH #0FFH

MOV 7FH #40H

INC @R0

INC R0

INC @R0

4. 已知 (SP)=09H, (DPTR)=4567H, 在执行下列指令后, (SP)=\_\_0BH\_\_, 内部 RAM(0AH)=\_\_67H\_\_, (0BH)=\_\_45H\_\_

PUSH DPL

PUSH DPH

5. 下列程序中注释的数字为执行该指令所需的机器周期数, 若单片机的晶振频率为 6MHz, 问执行下列程序需要时间为\_\_1006 微秒\_\_。

```
MOV R3,#100; 1
LOOP: NOP      ; 1
      NOP
      NOP
      DJNZ R3,LOOP ; 2
      RET      ; 2
```

得分

### 五、设计题 (24 分)

1. 已知内部 RAM 30H 单元开始存放 20H 个数据 将其传送到外部 RAM 的 4000H 单元开始的存储区 请编程实现。(8 分)

```
MOV R0 #30H
MOV DPTR #4000H
MOV R2 #20H
LOOP: MOV A @R0 取数
      MOVX @DPTR A 存数
      INC R0
      INC DPTR
      DJNZ R2
      LOOP
      RET
```

2. 要求使用定时器/计数器实现在 P1.0 引脚上产生周期为 4ms 的方波输出, 已知单片晶体振荡器的频率为  $f_{osc}=12\text{MHz}$ , 请使用定时器/计数器 T0 的方式 0。(16 分)

(1) 计算求解出定时常数 TC?

(2) 根据计算结果, 编写程序在 P1.0 引脚上产生周期为 4ms 的方波输出。

方式寄存器 TMOD:

GATE	C/T	M1	M0	GATE	C/T	M1	M0
------	-----	----	----	------	-----	----	----

1)  $T_c=4192$

2) ORG 0000H

AJMP MAIN

ORG 000BH

AJMP INQP

ORG 0030H

MAIN: MOV TMOD, #00H

MOV TH0, #04H

ORG 2000H

INQP: MOV TH0, #04H;

MOV TL0, #30H

SETB TR0

SETB ET0

SETB EA

AJMP \$

MOV TL, #30H

CPL P1.0

RETI

84



# 《 嵌入式系统及应用 》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

自觉遵守考场规则，诚信考试，绝不作弊

得分

## 一、填空 (25 分, 每空 1 分)

- 若外接晶振为 6MHz, 则 80C51 单片机的振荡周期为 60ns, 机器周期为 245, 指令周期最短为 245, 指令周期最长为 845。
- 80C51 的位寻址区包括 片内数据 RAM 32-47 和 可位寻址的特殊功能寄存器。
- 80C51 P1 口的读操作有两种方式: 读引脚 和 读锁存。实际应用中, 读引脚前应先输出指令置口锁存器 D=       , 这就是准双向 I/O 口的特点。
- 80C51 单片微机中, 堆栈的地址指针为       , 程序存储器指令地址为       , 外部数据存储器使用的 16 位地址指针为       。
- 假定累加器 A 的内容为 30H, 执行指令: 1000H: MOVCA, @A+PC 后, 把程序存储器        H 单元的内容送累加器 A。(注: MOVCA 为单字节指令)
- 80C51 单片微机中共有 5 个中断源, 包括 2 个外部中断源。每个中断源可通过寄存器设置为 2 个优先级。
- 80C51 内部定时器/计数器的定时和计数功能分别对 T0, T1 或 T2 引脚输入的 1-0 跳变 和 对微机内部机器周期 进行计数。
- 80C51 串行口的工作方式中, 波特率可变的是工作方式 1 和工作方式 3, 这两种方式选用 定时计数器 作为串行口的波特率发生器。
- 单片机的外部中断触发方式有 电平有效方式和跳变有效方式 两种。
- 定时器 T0 采用工作方式 0, 若计算得到的定时常数为 1E0CH, 则计数寄存器的内容 TH0= 0 TL0= 0。

TH 高 8 位  
TL 低 8 位

00011110/00001100

得分

## 二、简答题 (20 分, 每题 5 分)

1、说明 80C51 单片机的引脚 EA 的作用, 该引脚接高电平和接低电平时各有何功能? EA: 片外程序存储器访问允许信号, 低电平有效。

EA=1: 选择片内程序存储器

EA=0, 则程序存储器全部在片外, 而不管片内是否有程序存储器

2、在 80C51 单片微机系统中, 外接程序存储器和数据存储器共用 16 位地址线和 8 位数据线, 为什么不会发生冲突?

因为控制信号线的不同: 外扩的 RAM 芯片既能读出又能写入, 所以通常都有读写控制引脚, 即为 OE 和 WE, 外扩 RAM 的读写控制引脚分别与 MCS-51 的 RD 和 WR 引脚相连, 外扩的 EPROM 在正常使用中只能读出不能写入, 故 EPROM 没有写控制脚, 又有读出引脚, 该引脚与 MCS-51 的 PSEN 相连。

4、80C51 共有哪几种寻址方式? 简述寄存器间接寻址的寻址范围。

得分

## 三、程序分析题 (30 分)

1、如果 DPTR=507BH, SP=32H, (30H)=50H, (31H)=5FH, (32H)=3CH, 试分析在执行下列指令后, DPH、DPL、SP 单元中的内容。(5 分)

POP DPH

POP DPL

POP SP

DPH= 3CH, DPL= 5FH, SP= 50H

2、读程序, 填写 PSW 中内容。(6 分)

MOV A, #0FBH

MOV PSW, #10H

ADD A, #7FH

1111 1011  
0111 1111  
1011 1010

执行完后，将 PSW 各位状态填入下表：

CY	AC	FO	RS1	RS0	OV	FI	P
1	1	0	1	0	1	0	1

3. 在 8051 片内 RAM 中，已知 (30H) = 38H, (38H) = 40H, (40H) = 48H, (48H) = 90H，试分析下段程序，按照示例格式给每条指令添加注释，说明该指令的作用以及执行完该程序指令后的目标操作数单元的结果。(7 分)

例：MOV 88H, #30H : 将立即数 30H 发送至单元 88H 中，(88H) = 30H

MOV A, 40H : 将 40H 单元的内容发送至 A 中，A = (40H) = 48H

MOV R1, A : 将 A 中的内容送到 R1，R1 = 48H

MOV @R1, 30H : 将 30H 单元的内容发送至 @R1 单元中，(48H) = 38H

MOV DPTR, #1234H : 将立即数 1234H 发送至 DPTR，DPTR = 1234H

MOV 40H, 38H : 将 38H 单元的内容发送至 40H 单元中，(40H) = 40H

MOV R1, 30H : 将 30H 单元的内容发送至 R1，R1 = 38H

MOV 90H, R1 : 将 R1 内容发送至 90H 单元中，(90H) = 30H

4. 读下面程序回答问题：(12 分)

```

START:  MOV  DPTR, #TABLE
        CJNE  A, #0aH, LOOP
        AJMP  ERR
LOOP:   JNC   ERR
        MOVC  A, @A+DPTR
        MOV  B, A
        MOV  R1, #00H
EXIT:   SJMP  $
ERR:    MOV  R1, #0FFH
        MOV  B, R1
        SJMP EXIT
ORG     1000H : 数字 0~9 的 ASCII 码表
TABLE:  DB    30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H, 38H, 39H

```

- (1) 表中 32H 的地址是 DPTR+2。
- (2) 执行这段程序前，如果 A 中为 0AH，则执行后，(R1) = (0FFH)。
- (3) 执行这段程序前，如果 A 中为 07H，则执行后，(R1) = (00H)，B 中为 37H。
- (4) 这段程序的功能是：将 A 中的内容送至 R1，并将 R1 中的内容送至 B。

得分

#### 四、编程题 (25 分)

1、试编程实现在外部数据存储器 2000H~200FH 16 个单元中按顺序存入数据 00H~0FH，并加上必要的注释。(10 分)

```

ORG 0000H
MOV R2, #10H ; 要存的数据个数
MOV DPTR, #2000H
MOV A, 00H
LOP: MOV @A+DPTR, A ; 将数据放入数据空间
INC A
DJNZ R2, LOP ; 判断是否存完
SJMP $

```

2、要求使用定时器/计数器实现在 P1.0 引脚上产生周期为 8ms 的方波输出，已知单片机晶振频率为  $f_{osc}=6\text{MHz}$ ，请使用定时器/计数器 T0 的方式 0。(15 分)

- (1) 计算求解出定时常数 TC?
- (2) 根据计算结果，编写程序在 P1.0 引脚上产生周期为 8ms 的方波输出。

方式寄存器 TMOD:

GATE	C/T	M1	M0	GATE	C/T	M1	M0
------	-----	----	----	------	-----	----	----

装订线内不要答题

# 《 嵌入式系统及应用 》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分	
----	--

一、填空题 (每空 1 分, 共 20 分)

1. 若 MCS51 系统采用 6MHz 晶振, 则机器周期为 2us, 执行指令 DIV AB 需要时间 2 us。若要该单片机可靠复位, 必须在 RST 引脚保持 4 us 以上的高电平。

MCS-51 有 5 个中断源, 提供 4 个中断优先级。若 IE=10011010B, IP=00011000B, 则能响应的中断源有 定时/计数器 0, 定时器 1, 串行通信, 当所有中断源同时提交申请时, 则最先响应 定时器 1 中断。

3. 若定时器 T0 工作在方式 0, 要产生 5ms 的定时, 则 TMOD= 100 B, TCON=          B, TH0=          B, TL0=          B。(振荡频率为 12MHz)

4. ARM 字数据存储器格式有大端格式和 小端格式, 若存储单元 0x80000002-0x80000005 分别存储 0x01, 0x02, 0x03, 0x04, 则若按大端格式读取 0x80000002 的字是 0x 04, 属于 半字 对齐方式。

5. 若 R1=0x87654321, 执行 BX R1 指令后, 进入 ARM 状态, 此时若遇上一条不能识别的指令, 则会进入 Thumb 模式, 且状态切换成 Thumb 状态。

6. 如果 R0=0x0002, R1=0x0001 则  
LDR R1, [R0, #6] 访问的存储单元地址为 0x 0008;  
LDR R1, [R0], #6 访问的存储单元地址为 0x 0002;  
执行该指令后 R0 的值为 0x 0008;  
LDR R1, [R0, R0, LSL R1] 访问的存储单元地址为 0x 0006;

得分

二、单一选择题（每题 1.5 分，共 30 分）

1. ( D ) 下面哪个系统不属于嵌入式系统。  
 A. MP3 播放器      B. GPS 接收机  
 C. “银河玉衡”核心路由器      D. “天河一号”计算机系统
2. ( A ) 下面关于哈佛结构描述正确的是。  
 A. 程序存储空间与数据存储空间分离      B. 存储空间与 IO 空间分离  
 C. 程序存储空间与数据存储空间合并      D. 存储空间与 IO 空间合并
3. ( B ) I2C 协议中有几根线。  
 A. 1      B. 2  
 C. 3      D. 4
4. (    ) 下面哪种嵌入式操作系统很少用于手机终端设备上。  
 A. Symbian      B. WinCE  
 C. Android      D. us/os
5. ( B ) 若采用 8031 作为控制芯片，则 EA/V<sub>pp</sub> 应该连接至。  
 A. VCC      B. GND  
 C. 21V      D. 12V
6. ( A ) 下列指令中错误的是。  
 A. SETB 50H.0      B. MOV A, B  
 C. JNC LOOP      D. SUBB A, R0  
 C<sub>y=0</sub> 故, C<sub>y=1</sub> 故 JC
7. ( D ) 在下列信号中，不是给程序存储器扩展使用的是  
 A. PSEN ✓      B. EA ✓  
 C. ALE ✓      D. WR ✓
8. ( A ) MCS-51 单片机的堆栈区建立在。  
 A. 片内数据存储器区      B. 特殊功能寄存器区  
 C. 片外数据存储器区      D. 程序存储器区
9. (    ) 若使用 DJNZ R2, LOOP 来执行循环，则循环次数最多的是。  
 A. R2=0      B. R2=1  
 C. R2=255      D. R2=256

D2: 丁及王

10. ( ) 中断查询确认后, 在下列各种单片机运行情况下, 能很快进行中断响应的是。

- A. 当前正在进行高优先级中断处理      B. 当前正在执行 RETI 指令  
C. 当前正在执行对 IE 进行修改的指令      D. 当前正在执行 MOVX 指令

11. ( ) 执行 MOV IE, #81H 指令的意义是。

- A. 屏蔽中断源      B. 开放外部中断源 0  
C. 开放外部中断源 1      D. 开放外部中断源 0 和 1

12. ( ) 在 MCS51 中, 实现 P0 口线的数据和低位地址利用复用, 应使用。  
A. 地址锁存器      B. 地址寄存器  
C. 地址缓冲器      D. 地址译码器

13. ( ) 在 MCS51 中, 与定时工作方式 0 和 1 相比较, 定时工作方式 2 不具备的特点是。

- A. 计数溢出后能自动恢复计数初值      B. 增加计数器的位数  
C. 提高了计时的精度      D. 适于循环定时

14. ( ) 下面哪一种工作模式不属于 ARM 特权模式。

- A. 用户模式      B. 管理模式  
C. 软中断模式      D. FIQ 模式

15. ( ) 对寄存器 R1 的内容乘以 4 的正确指令是。

- A. LSR R1, #2      B. LSL R1, #2  
C. MOV R1, R1, LSL #2      D. MOV R1, R1, LSR #2

16. ( ) 下面指令执行后, 改变 R1 寄存器内容的指令是。

- A. TST R1, #2      B. ORR R1, R1, R1  
C. CMP R1, #2      D. EOR R1, R1, R1

17. ( ) 寄存器 R14 除了可以做通用寄存器外, 还可以做。

- A. 程序计数器      B. 链接寄存器  
C. 堆栈指针寄存器      D. 基址寄存器

18. ( ) FIQ 中断的入口地址是。

- A. 0x0000001C      B. 0x00000008  
C. 0x00000018      D. 0x00000014

19. ( ) 以下哪个是有效的 8 位位图。

- A. 0x00123400      B. 0x80000012  
C. 0x50000080      D. 0x00808000

20. ( ) LDMED/STMED 指令用于。

- A. 满递增堆栈      B. 空递增堆栈  
C. 满递减堆栈      D. 空递减堆栈

得 分

### 三、问答题（每题 5 分，共 20 分）

1. 什么是嵌入式系统？其特点是什么？

答：

2. 在 MCS51 外扩的程序存储器和数据存储器可以有相同的地址空间，但不会发生数据冲突，为什么？

答：



3. 8051 在什么条件下可以响应中断?

答:

- ① 中断源申请中断
- ② 中断源已被允许中断, 且 CPU 也已允许中断
- ③ 没有同级或高优先级中断在执行中断服务程序
- ④ 143

4. ARM 处理器中, ARM 指令集和 Thumb 指令集的区别。ARM 指令带条件执行的优点是什么?

答:

得分

四、编程题（每题 5 分，共 20 分）

1. 把外部数据存储器 8000H 开始的 50 个字节分别送至外部存储器 9000H 开始的单元中（MCS51）

2. 串口编程：以 19200bps 的速率把片外存储器中 8000H 开始的 10 个字节发送出去。

（MCS51，晶振 11.0592MHz，无校验，查询方式或中断方式均可）

```
ORG 0000H
AJMP START
ORG 0000H

START: MOV DPTR, #8000H
      MOV P2, #90H
      MOV R0, #50H
      MOV R1, #50H
      30T
loop: MOVX A, @DPTR
      MOVX @R0, A
      INC DPTR
      INC R0
      DJNZ R1, loop
      RET
```

3. 用 ARM 指令实现下面从 1 加到 n 的求和函数。

```
int abc(int n)
{
    int i;
    int sum;

    sum = 0;
    for(i=n; n>0; i--)
        sum = sum + i;
    return(sum);
}
```

提示：参数 n 传送至 R0，返回值通过 R0 返回。

ARM 指令：

ABC

```
MOV R3, #n
MOV R0, #0
MOV R1, #1
Loop
    ADD R0, R0, R1
    ADD R1, R1, #1
    CMP R1, R3
    BNE Loop
    BXIP
    MOV PC, LR
```

4. 用 ARM 指令实现下面求三个数中最大值的 C 语言函数

```
int max(int a,b,c)
{
    int d;

    d = a;
    if(d < b)
        d = b;
    if(d < c)
        d = c;
    return(d);
}
```

提示：参数 a, b, c 分别传送至 R0, R1, R2，返回值通过 R0 返回。

ARM 指令：

MAX

```
MOV R0, R0
CMP R0, R1
MOVLT R0, R1
CMP R0, R2
MOVLT R0, R2
MOV PC, LR
```

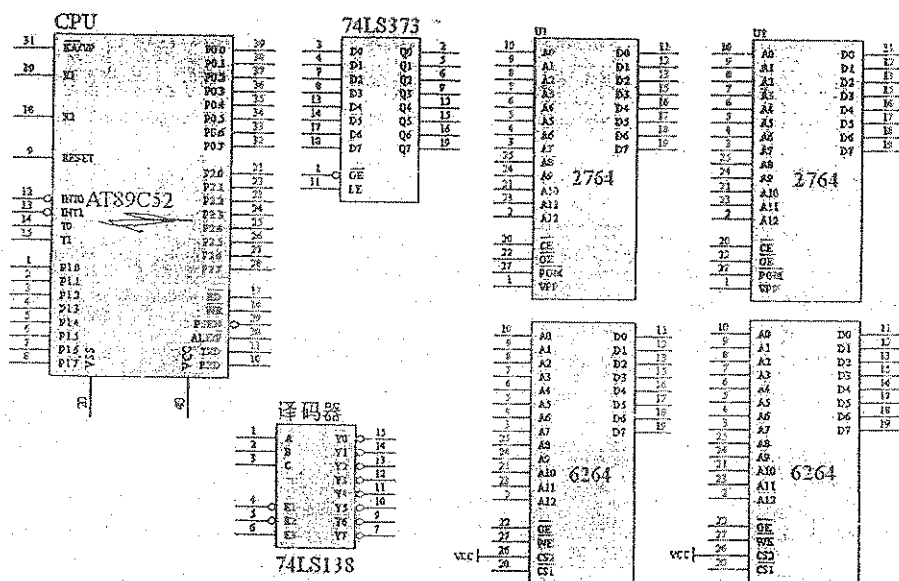
得分

### 五、综合题 (10 分)

采用 AT89C52(8KROM, 256B RAM) 芯片, 设计一个单片机应用系统。完成原理图, 并包括使单片机工作的最小电路。

要求:

- 1) 外扩 2 片数据存储器 6264(8K), 其地址范围分别为: 0000H~1FFFH, 2000H~3FFFH
- 2) 使用 AT89C52 内部 ROM, 并外扩 2 片程序存储器 2764(8K), 地址范围分别为: 2000H~3FFFH、4000H~5FFFH



装订线内不要答题

[illegible]

得分

```
SETB      C
MOV       A, #10110110B
MOV       B, #10111111B
SUBB     A, B
```

2. 若 MCS51 系统采用 12MHZ 晶振, 则状态周期为 \_\_\_\_\_ us, 机器周期为 \_\_\_\_\_ us, 执行时间最长的指令是 \_\_\_\_\_, 它的指令周期是 \_\_\_\_\_ us。

读取特殊功能寄存器 90H 单元至 A 累加器的指令  
读取 ROM 0090H 单元至 A 累加器的指令

4. ARM7 处理器有两种状态是\_\_\_\_\_和\_\_\_\_\_, 可以通过\_\_\_\_\_指令进行状态切换; ARM7 处理器共有\_\_\_\_\_种处理器模式, 上电复位后进入\_\_\_\_\_模式, 当遇到无法识别的指令时进入\_\_\_\_\_模式, 当访问存储器异常时进入\_\_\_\_\_模式, 操作系统一般工作在\_\_\_\_\_模式下, 用户可以通过\_\_\_\_\_指令去调用操作系统所提供功能。

LDR R1, [R0]	访问的存储单元地址为_____;
LDR R1, [R0, #0X10]!	访问的存储单元地址为_____;
LDR R1, [R0], #0X10	访问的存储单元地址为_____;

得分

二、单一选择题（每题 1.5 分，共 15 分）

- 89C51 上电复位后执行 SETB RS1, 则此时 R0 对应的片内 RAM 地址是( )  
A. 00H B. 08H  
C. 10H D. 18H
- 以下指令错误的是( )  
A. SETB C B. SETB P0.1  
C. SETB F9H D. SETB F9H.0
- 当 IE 设为 10001110B, IP 设为 00010100B 时, 当所有中断源同时提出中断请求时, 最先响应的中断源是( )  
A. INT0 中断 B. T0 中断  
C. INT1 中断 D. T1 中断
- 外扩 8K 的 ROM 与 89C51 片内的 ROM 相衔接以构成较大的 ROM 空间, 则 EA/VPP 引脚应该连接至( )  
A. VCC B. GND  
C. 12V D. 21V
- 在 MCS51 串口多机通信中, 从机初始化时把 SM2 设置为 1, 则此时从机能接收( )  
A. 地址帧 B. 数据帧  
C. 地址帧和数据帧 D. 以上均不是
- 以下哪个是有效的 8 位位图( )  
A. 0x12345678 B. 0x00000123  
C. 0xFF000000 D. 0xCC800000
- 在采用三级流水线 ARM7 中, 在 ARM 状态下执行地址为 0x40000000 的指令 ADD R0,PC,#10 后, R0 的值为( )  
A. 0x40000000 B. 0x40000010  
C. 0x40000018 D. 0x40000020
- 若 R1=0x40000000 则执行指令 STMIB R1!,{R5-R7}后, R5 存储至( )  
A. 0x40000000 B. 0x40000004  
C. 0x40000008 D. 0x40000010
- 以下不能实现程序跳转的指令是( )  
A. MOV PC,LR B. SUBS PC,R14,#4  
C. ADD R0,PC,#4 D. LDMFD SP!,{R0-R7,PC}^
- LDMFA/STMFA 指令用于( )  
A. 满递增堆栈 B. 空递增堆栈  
C. 满递减堆栈 D. 空递减堆栈

得 分

三、问答题（每题 6 分，共 30 分）

1. IEEE 对嵌入式系统所做的定义是什么？以及嵌入式系统应该具备以下哪 4 个特性？

2. 80C51 的中断与子程序调用有什么异同点？

3. 在 MCS51 中，试叙述 INT0 中断的响应过程。

4. 在 ARM 中，试叙述 IRQ 异常的响应过程。

5. 列举嵌入式系统采用 RTOS 的好处, 并说明抢占式调度算法和非抢占式调度算法的区别。

得 分

四、编程题 (每题 5 分, 共 20 分)

1. 找出外部数据存储器中 8000H---8050H 单元中的最大数(字节)送至内部 RAM 的 20H 单元, 编写程序 (MC:S51)

--

2. 找出数据存储器中 0x8000---0x8050 单元中的最大数 (字节) 并送至 R0 中。(ARM)

--



3. 串口编程：以 9600bps 的速率把字符串 “Hello world” 发送出去。(MCS51, 晶振 11.0592MHZ, 查询方式或中断方式均可)

4. 用 ARM 指令实现下面求绝对值的 C 语言函数

```
int abs(int a)
{
    int c;

    if ( a >= 0)
        c = a;
    else
        c = -a;
    return(c);
}
```

提示：用 R0 代替 a, 返回值 c 也用 R0 代替。

得分

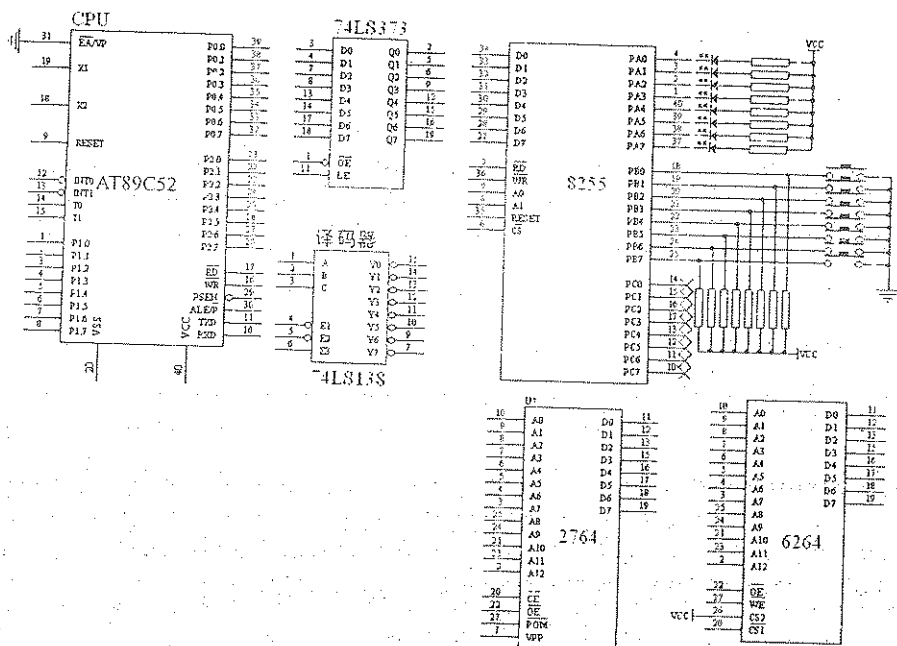
### 五、综合题 (10 分)

采用 AT89C52 (8KROM, 256B RAM) 芯片, 设计一个单片机应用系统, 如下图所示:

要求:

- 1) 外扩一片数据存储器 6264 (8K), 其地址范围唯一: 0000H~1FFFH
- 2) 外扩一片程序存储器 2764 (8K), 其地址与片内 ROM 相衔接
- 3) 外接一片 8255 以扩展 I/O 口, 其端口地址为: 8000-8003H。如图所示, 8255 的 PA 口连接 8 个 LED 灯, PB 口与 8 个按键相连。

1. 完成原理图, 并包括使单片机工作的最小电路 (7 分)



白 觉 遵 守 考 试 规 则, 诚 信 考 试, 绝 不 作 弊

2. 当按键被按下时，点亮的 LED 灯。例如：按下与 B 口第 0 位相连的按键时点亮与 A 口第 0 位相连的 LED 灯。写出 8255 的初始化程序（3 分）



《 嵌入式系统及应用 》

标准答案和评分标准

一、填空题（每空 1 分，共 25 分）

1. 0F6 、 0BF、 1 、 1 、 0

2. 1/6us、 1us 、 乘除指令、 4us

3. MOV R0, #90H 、 MOV A, @R0;

MOV A, 90H;

MOV DPTR, #0090H 、 MOV A, #0 、 MOVCA, @A+DPTR;

MOV DPTR, #0090H、 MOVXA, @DPTR

4. ARM 状态、Thumb 状态、 BX/BLX、 7、 管理、 未定义、 中止、 系统、 SWI

5. 0x8000、 0x8010、 0x8000

二、选择题（每题 1.5 分，共 15 分）

1. CDCAA 、 CDBCA

三、简答题（每题 6 分，共 30 分）

1. 根据英国电机工程师协会（IEEE）所做的定义： devices used to control, monitor, or assist the operation of equipment, machinery or plants 嵌入式系统是“控制、监视或辅助某个设备、机器甚至工厂运行的设备”。

嵌入式系统应该具备以下 4 个特性：

执行特定的功能、以微处理器与外围设备构成核心、需要严格的时序与稳定性、全自动操作。

2 答：

相同点：

- 1) 都是中断当前正在执行的程序，转去执行其它子程序或中断服务子程序
- 2) 都需要保护断点地址以便执行完子程序或中断服务子程序返回断点出继续执行
- 3) 执行完成后，完成现场恢复
- 4) 两者都可以用嵌套调用

不同点：

- 1) 中断请求是由外部设备发出的，是随机的
- 2) 中断由固定的矢量地址转入中断服务程序，子程序是由软件设定
- 3) 中断响应是受控的，响应时间受影响，子程序响应时间是固定的。

3 答：

CPU 每个周期极其采样 INT0 引脚信号一次，当有中断请求时，响应中断，由硬件生成长调用指令 LCALL 0003H；将当前程序计数器 PC 压入堆栈进行保护；将对应的中断源的中断矢量地址装入 PC，转向中断服务程序，直至 RETI 为止；撤销中断请求，弹出断点处地址至 PC；恢复源程序的断点执行，恢复中断触发源状态；

4 答：

- 1) 保存当前状态，PC->LR 中
- 2) CPSR 存入 SPSR\_irq
- 3) 强制 M[4:0]为 10010B 进入 IRQ 模式

- 4) 将 CPSR 中断标志 I 禁止, F 不变
- 5) 将 T 标志清 0, 处于 ARM 状态
- 6) 强制 PC 的值为特定的值, 去执行异常响应程序
- 7) 执行异常响应返回指令, 返回被中断的程序继续执行

5 答:

RTOS 好处: 便于设计可靠性高、提高计算机的运行速度、有效支持高级语言程序、便于用户程序移植。

非抢占式调度算法: 中断服务可以使一个高优先级的任务由阻塞状态变为就绪状态。但中断服务以后控制权还是回到原来被中断了的那个任务, 直到该任务主动放弃 CPU 的使用权时, 那个高优先级的任务才能获得 CPU 的使用权。

抢占式调度算法: 最高优先级的任务一旦就绪, 总能得到 CPU 的控制权。即当一个运行着的任务使一个比它优先级高的任务进入了就绪状态, 当前任务的 CPU 使用权就被剥夺了, 或者说被挂起了, 那个高优先级的任务立刻得到了 CPU 的控制权。

#### 四、编程题 (每题 5 分, 共 20 分)

1.

- |                     |     |
|---------------------|-----|
| 有递增寻址寄存器指令          | 1 分 |
| 有间接寻址方式访问片外 RAM 的指令 | 1 分 |
| 有 MOVX 指令           | 1 分 |
| 有循环结构               | 1 分 |
| 程序基本正确              | 1 分 |

```

ORG    0000H
MAIN:  MOV    B, #0
        MOV    R2, #51H
        MOV    DPTR, #8000H
LOOP:  MOVX   A, @DPTR
        CLR    C
        PUSH   ACC
        SUBB   A, B
        POP    ACC
        JC     NEXT
        MOV    B, A
NEXT:
        INC    DPTR
        DJNZ   R2, LOOP
        MOV    20H, B
        SJMP   $

```

2.

- |          |     |
|----------|-----|
| 有 LDR 指令 | 1 分 |
| 有 ADD 指令 | 1 分 |
| 有多字节操作指令 | 1 分 |
| 有地址赋值指令  | 1 分 |
| 程序基本正确   | 1 分 |

```

MOV    R0, #0
LDR     R1, =0x8000
LDR     R2, =0x8050
LOOP
LDRB   R2, [R1]
CMP     R0, R2
MOVLOR0, R2
ADD     R1, R1, #1
CMP     R1, R2
BNE     LOOP

```

3.

- 有初始化串口寄存器指令 1 分
- 有初始化定时器指令 1 分
- 有间接寻址的指令 1 分
- 有中断框架/查询框架 1 分
- 程序基本正确 1 分

```

ORG    0000H

MAIN:
MOV     SCON,#01010010B
ANL     PCON,#7FH
MOV     TMOD,#00100000B
MOV     TH1,#0FDH
MOV     TL1,#0FDH
SETB    TR1
MOV     DPTR,#STR

```

```

LOOP:
JNB     TI,LOOP
CLR     TI
CLR     A
MOVC    A,@A+DPTR
JZ      EXIT
MOV     SBUF,A
INC     DPTR
SJMP    LOOP

EXIT:   SJMP    $
STR:    DB      "Hello world",0

```

4.

- 有比较指令 1 分
- 有跳转指令 1 分
- 跳转指令带条件 1 分
- 有减法指令 1 分
- 程序基本正确 1 分

```

abc
CMP      R0, #0
MOVGE    R0,R0
RSBLT    R0,R0,#0
MOV      PC,LR

```

五、

综合题 (10 分)

1.

- 373 的 D 端——P0 正确 1 分
- 373 的 Q 端——6264 的 A0~A7 正确 1 分
- 译码电路正确 1 分
- RD——6264 的 OE 正确 1 分
- WR——6264 的 WE 正确 1 分
- 有最小工作外围电路 1 分
- 8255 连接基本正确 1 分

2.

- 控制字正确 1 分
- 程序基本正确 2 分

```

PORTCON EQU 3FFFH
MOV      DPTR,#PORTCON
MOV      A,#10001001B
MOVX     @DPTR,A

```





# 《 嵌入式系统及应用 》

院(系) 自动化 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

自觉遵守考试规则，诚信考试，绝不作弊

得分
----

一、填空题 (每空 1 分, 共 30 分)

1. 执行指令 SETB C  
MOV A, #10110110B  
MOV B, #10110000B  
ADDC A, B

则 A = 67 H, B = 5BC H, C = 1, AC = 0, OV = 1, P = 1.

2. 指令 MOV R0, #90H

MOV A, @R0 访问的是 片内 RAM 的 90H 单元;

指令 MOV A, 90H 访问的是 SFR 区 的 90H 单元;

指令 MOV DPTR, #0090H

MOVX A, @DPTR 访问的是 片外 RAM 的 90H 单元;

指令 CLR A

MOV DPTR, #0090H

MOVC A, @A+DPTR 访问的是 ROM 的 90H 单元。

(选填 SFR 区、片内 RAM, ROM, 片外 RAM)。

3. 在 RST 引脚出现 10ms 以上的高电平单片机就会复位, 复位后程序计数器 PC 值为 0000 H, SP 值为 07 H。

4. 若 IE=10011010B、IP=00011000B, 则能响应的中断源有 定时器 0 定时器 1 串行通信, 当所有中断同时提交申请时, 则最先响应 定时器 0 中断。

5. ARM7 处理器, 在 ARM 状态下指令长度为 4 字节, 在 Thumb 状态下指令长度为 1 字节。

6. 嵌入式处理器可以分为以下几大类: 嵌入式微处理器、嵌入式微控制器、嵌入式 DSP 处理器、嵌入式片上系统 SOC。

7. 若 R0=0x8000, R1=0x0004, 则:

LDR R2,[R0]	访问的存储器单元地址为 <u>0x8000</u>
LDR R2,[R0,#0x10]	访问的存储器单元地址为 <u>0x8010</u>
LDR R2,[R0,R1]	访问的存储器单元地址为 <u>0x8004</u>
LDR R2,[R0, R1, LSL #1]	访问的存储器单元地址为 <u>0x8008</u>
LDR R2,[R0, #0x04]!	访问的存储器单元地址为 <u>0x8004</u> , 执行指令后, R0= <u>0x8004</u>
LDR R2,[R0],#0x04	访问的存储器单元地址为 <u>0x8000</u> , 执行指令后, R0= <u>0x8004</u>
STMIA R0!,{R5-R8}	R5 存储的单元地址为 <u>0x8000</u> , 执行指令后, R0= <u>0x8004</u>

得分

## 二、单一选择题 (每题 1.5 分, 共 30 分)

- 同一优先级下, CPU 响应中断由先到后的顺序是( B )  
A. T0、T1、INT0、INT1、串口 B. INT0、T0、INT1、T1、串口  
C. 串口、T1、INT1、T0、INT0 D. T0、INT0、T1、INT1、串口
- 下列指令中错误的是( C )  
A. ADD A,B B. MOVX @DPTR,A C. PUSH R0 D. MOV ACC,7,C
- 在晶振频率相同的情况下, 定时间隔最长的方式是( B )。  
A. 0 B. 1 C. 2 D. 3
- 采用 DJNZ R2,LOOP 指令循环时, 若想使循环次数最多, R2 的值应为( 255 )。  
A. 0 B. 1 C. 255 D. 256
- 在下列哪种情况下, 不影响 SP 值( C )。  
A. 中断 B. 复位 C. 数据传送指令 D. 压栈指令
- 执行 MOV DPTR,#1234H 后, DPH 的值为( 12H )。  
A. 12H B. 34H C. 00H D. 0FFH
- 定时器/计数器工作在( 方式 0 )时, 计数初值自动重装。  
A. 方式 0 B. 方式 1 C. 方式 2 D. 方式 3
- 若指令 MOV A,#0 (指令长度为 2) 的地址是 0100H, 则执行完该指令后 PC=( 0101H )  
A. 0100H B. 0101H C. 0102H D. 0103H
- 无法把 A 中的最高位 (即 ACC.7) 置为 0 的指令是( C )。  
A. CLR ACC.7 B. ANLA,#7FH  
C. ORLA,#7FH D. MOVA,#00H
- 与中断有关的说法错误的是( C )。  
A. 需用 RETI 返回 B. 应保护现场

- C. 发生时刻是固定的 D. 可选用不同的工作寄存器组
11. ARM7TDMI-S 的后缀 D 的含义是( C )。
- A. 支持 Embedded ICE 观察硬件 B. 支持 64 位乘法
- C. 支持片上调试 D. 支持高密度 16 位的 Thumb 指令集
12. 在采用三级流水线 ARM7 中, 在 ARM 状态下执行地址为 0x8000 的指令 ADD R0,PC,#4 后 R0 的值为( D )。
- A. 0x8000 B. 0x8004 C. 0x8008 D. 0x800C
13. 上电复位后, ARM7 处在( A )
- A. ARM 状态和管理模式 B. ARM 状态和系统模式
- C. Thumb 状态和管理模式 D. Thumb 状态和系统模式
14. 当处理器访问存储器失败时, 进入( B )
- A. 管理模式 B. 中止模式 C. 未定义模式 D. 系统模式
15. 下面哪种不属于嵌入式操作系统( A )
- A. Windows xp B. WinCE
- C. VxWorks D. ucLinux
16. 在嵌入式 ARM 处理器中, 下面哪种中断方式优先级最高( A )
- A. 复位 B. 数据中止 C. FIQ D. IRQ
17. 在下面哪种操作系统中, 高优先级任务能得到及时响应( C )
- A. 分时操作系统 B. 非抢占式操作系统 C. 抢占式操作系统
18. 芯片的 JTAG 接口不能用于( D )
- A. 芯片测试 B. ISP 在线编程 C. 程序调试 D. 软件仿真
19. 若 CPSR 的值为 0x000000FF, 则下面不正确的是( A )
- A. 允许 FIQ 异常 B. 处于 Thumb 状态
- C. 禁止 IRQ 异常 D. 处于系统模式
20. LDMFA/STMFA 指令用于( A )
- A. 满递增堆栈 B. 空递增堆栈
- C. 满递减堆栈 D. 空递减堆栈

得分

### 三、编程题（每题 6 分，共 24 分）

1. 编写计算 0~100 间任意一个数的平方的子程序，要计算的数字通过 R0 传递给子程序，结果由 R1（存放高字节）、R2（存放低字节）返回，采用查表方式实现。（MCS51）

SQR:

```

    PUSH    PSW
    PUSH    ACC
    MOV     A, R0
    R1      A
    PUSH    ACC
    MOV     PPTR, #TABLE
    MOVL    A, @A+PPTR
    MOV     R1, A

    POP     ACC
    INC     PPTR
    MOVL    A, @A+PPTR
    MOV     R2, A

    POP     ACC
    POP     PSW
    RET
  
```

TABLE: DW 0,1,4,9,16,25,...10000

2. 编写计算 0~100 间任意一个数的平方的子程序，要计算的数字通过 R0 传递给子程序，结果由 R1 返回，采用查表方式实现。（ARM）

SQR

```

    STMFD   SP!, {R4}
    LDR     R4, =TABLE
    LDRH    R1, [R4, R0, LSL#1]

    LDMFD   SP!, {R4}
    MOV     PC, LR
  
```

TABLE DCW 0,1,4,9,16,25,...10000

3. 以 9600bps 的速率把片外 8000H 开始的数据发送出去, 遇 0 停止, 无奇校验位。  
(MCS51, 晶振 11.0592MHz, 查询方式或中断方式均可)

```

ORG 0000H
MAIN:
    MOV    SCON, #01000000B
    ALC    PCON, #7FH
    MOV    TMOD, #00100000
    MOV    TH1, #0F0H
    MOV    TL1, #0F0H
    SETB   TR1

    MOV    DPTR, #8000H

LOOP:
    MOVA   A, @DPTR
    JZ     EXIT
    INC    DPTR
WALT: JABTL, WALT
    CLR    TL
    MOV    SBUF, A
    SJMP  LOOP

EXIT
    SJMP  $
    
```

4. 用 ARM 指令实现下面的 C 语言函数。

```

int abc(int a)
{
    int b;
    if(a<0)
        b = -a;
    else
    {
        if(a>=5)
            b = a-5;
        else
            b = a;
    }
    return(b);
}
    
```

提示: 用 R0 代替 a, 返回值 b 也用 R0 代替。

ARM 指令:

```

ABC
    CMP    R0, #0
    BGE    PLUS
    RSB    R0, R0, #0
    B      EXIT

PLUS
    CMP    R0, #5
    BLT    EXIT
    SUB    R0, R0, #5
EXIT
    MOV    PC PC, LR
    
```

得分

#### 四、综合题 (16 分)

1. 设计一个单片机应用系统, 要求:

a) 采用 89C51 芯片 (有 8K 的片内 ROM)

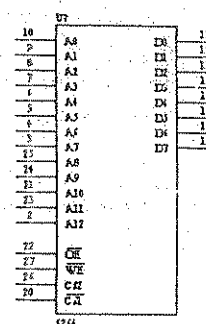
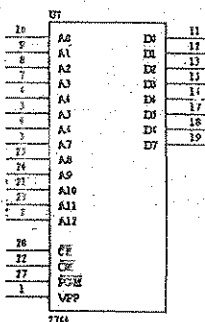
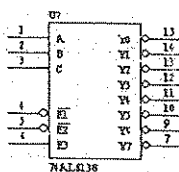
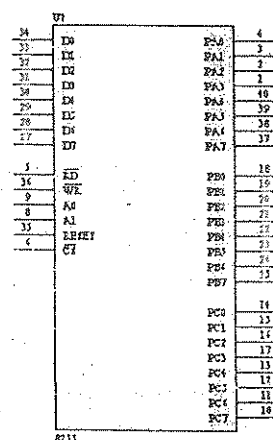
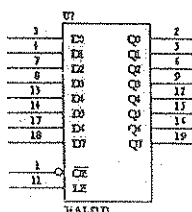
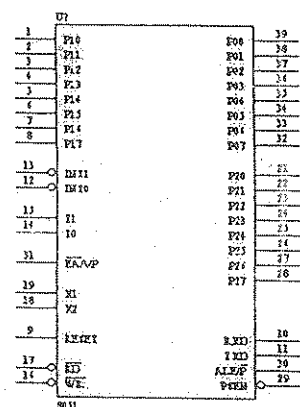
b) 外部程序存储器容量 8KB, 地址空间与 80C51 的内部 ROM 相衔接

c) 外部数据存储器 8KB, 地址空间为: 0000H~1FFFFH (地址范围唯一)

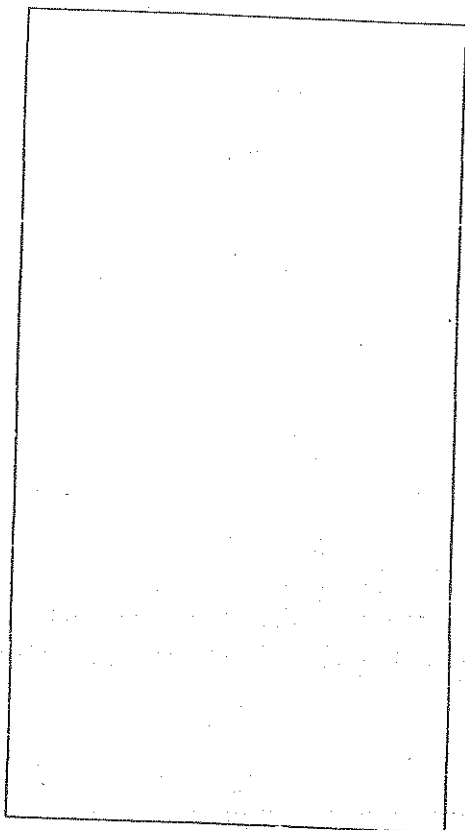
d) 扩展一片 8255A, 口地址为: 3FFCH~3FFFH

按照上述要求完成完整电路图, 包括能使单片机工作的最小电路。(12 分)

说明: 可以直接把线连上; 也可在引脚上写网络标号, 相同网络标号的引脚是连在一起的。



2. 在上图中，8255 的 A 口连接 8 个 LED 灯，8255 的 C 口连接 8 个按键。  
当某一按键按下时，点亮对应的 LED 灯，编写完整程序。（4 分，键盘去抖动不考虑）







# 《 嵌入式系统及应用 》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

自觉遵守考场规则，诚信考试，绝不作弊

得分

## 一、填空题（每空 1 分，共 25 分）

1. 80C51 包括 5 个中断源，分别为外部中断 0、定时器/计数器 0、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
2. 51 单片机程序存储器指令地址使用计数器为 \_\_\_\_\_，外接数据存储器 16 位地址指针为 \_\_\_\_\_，堆栈的地址指针为 \_\_\_\_\_。
3. 80C51 的位寻址区包括 \_\_\_\_\_ 和 \_\_\_\_\_。
4. 假定  $(SP) = 62H$ ， $(61H) = 30H$ ， $(62H) = 70H$ 。下列指令：POP DPH；POP DPL；执行后，DPTR 的内容为 \_\_\_\_\_，SP 的内容为 \_\_\_\_\_。
5. 若外接晶振为 6MHz，则 80C51 单片机的振荡周期为 \_\_\_\_\_，机器周期为 \_\_\_\_\_，指令周期最短为 \_\_\_\_\_。
6. 80C51 内部定时器/计数器的定时和计数功能分别对 \_\_\_\_\_ 和 \_\_\_\_\_ 进行计数。
7. ARM 内核提供符合 IEEE 1149.1 标准的 JTAG 调试接口，JTAG 调试使用 \_\_\_\_\_ 技术。
8. 从嵌入式操作系统能否满足实时性的要求来分类，VxWorks 属于 \_\_\_\_\_ 操作系统。
9. 在嵌入式处理器的分类中，单片机属于 \_\_\_\_\_。
10. ARM 微处理器支持 7 种运行模式，除用户模式外，其他 6 种模式又称为 \_\_\_\_\_ 模式，用户模式和系统模式之外的其它 5 种模式又称为 \_\_\_\_\_ 模式。
11. 在 ARM 的 C 语言编程中，内嵌的汇编语句采用符号 \_\_\_\_\_ 进行注释。
12. ARM RISC 支持两种指令集，包括 \_\_\_\_\_ 和 \_\_\_\_\_。
13. 嵌入式交叉开发环境由 \_\_\_\_\_ 和 \_\_\_\_\_ 组成。

得分

## 二、简答题（每小题 5 分，共 25 分）

1、80C51 有哪两种低功耗工作方式？写出它们的控制寄存器和控制位。

答：

2、在 80C51 单片机系统中，外接程序存储器和数据存储器共用 16 位地址线和 8 位数据线，为什么不会发生冲突？

答：

3、简述嵌入式系统的基本特征。

答：

4、与 CISC 架构相比，RISC 架构具有哪些优点？能否认为 RISC 可以取代 CISC 架构？

答：

5、如何进行嵌入式操作系统的选型？

答：

得分

## 三、程序分析题（2 小题，共 20 分）

1. 读下面程序回答问题：（10 分）

```

.....
START:  MOV DPTR, #TABLE
        CJNE A, #0aH, LOOP
        AJMP ERR
LOOP:   JNC ERR
        MOVC A,
        @A+DPTR
        MOV B, A
        MOV R1, #00H
EXIT:   SJMP $

```

```

ERR:    MOV R1, #0FFH
        MOV B, R1
        SJMP EXIT
        ORG 1000H
        : 数字 0~9 的 ASCII 码表
TABLE:  DB
        30H, 31H, 32H, 33H, 34H, 35H, 36H,
        37H, 38H, 39H
        .....

```

- (1) 表中 32H 的地址是\_\_\_\_\_。
- (2) 执行这段程序前, 如果 A 中为 0AH, 则执行后, (R1)=\_\_\_\_\_。
- (3) 执行这段程序前, 如果 A 中为 07H, 则执行后, (R1)=\_\_\_\_\_, B 中为\_\_\_\_\_。
- (4) 这段程序的功能是:\_\_\_\_\_。

2. 在嵌入式设计中, 假设在 C 语言程序文件中定义了如下函数:

```
int func(int x1, int x2, int x3, int x4, int x5)
{
    return x1+x2*x3-x4/x5;
}
```

阅读如下汇编程序, 并根据要求回答问题。(10 分)

AREA f, CODE, READONLY	STR R3, [SP, #4]!
IMPORT func	ADD R3, R1, R1
STR LR, [SP, #4]!	BL func
ADD R1, R0, R0	ADD SP, SP, #4
ADD R2, R1, R0	LDR PC, [SP], #4
ADD R3, R1, R2	END

- (1) 程序中 IMPORT func 是用于说明\_\_\_\_\_;
- (2) 汇编程序中用于调用 C 程序的语句为\_\_\_\_\_;
- (3) STR LR, [SP, #4]! 的含义是\_\_\_\_\_;
- (4) ARM 的汇编程序设计中如何完成多个参数的传递? 说明程序中调用 C 语言函数时各参数的对应关系。

得分

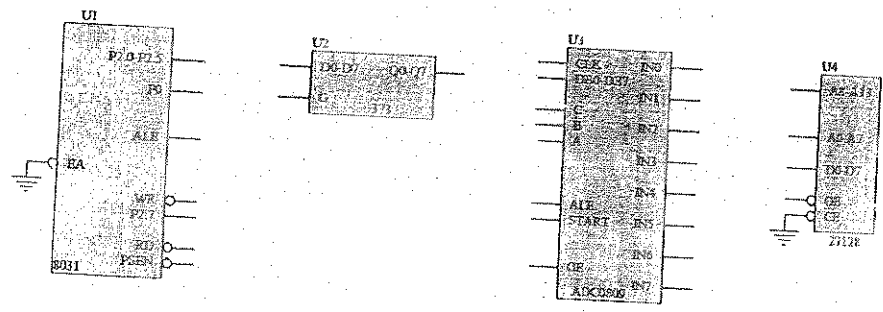
#### 四、综合设计题 (2 小题, 共 30 分)

1. 单片机串口采用中断法接收数据, 设  $f_{osc}=11.0592\text{MHz}$ , 波特率为 2400bps, SMOD=0, 中断优先级为高级, 奇偶校验位放在接收数据第 9 位。请编写初始化程序。(15 分)

自觉遵守考试规则，诚信考试，绝不作弊

2、设计一个 8031 单片机巡回检测系统，拟外扩一片 ADC0809，一片 EPROM (16K\*8) 27128，如图所示。要求编写每隔 50ms 依次检测一路模拟信号的源程序，采样转换值依次存入内存 20H~27H。假设 A/D 转换延时 0.128ms 的子程序已存在，入口标号 D128。晶振 6MHz。(15 分)

- 要求：(1) 补充必要的器件，完成单片机系统电路，连接各芯片。  
 (2) 列出 ADC0809 IN0~IN7 地址和 EPROM 地址。  
 (3) 编写依次巡回检测 8 路模拟信号的源程序，加上必要的伪指令。采用 A/D 转换结束实现中断的方式读入采样转换值，采样转换值依次存入 40H 开始的 8 个内存单元后重复运行。



# 《嵌入式系统》

考试时间: 120 分钟 开课学院 计算机 任课教师 \_\_\_\_\_

姓名 \_\_\_\_\_ 学号 \_\_\_\_\_ 班级 \_\_\_\_\_

## 一. 单项选择题 (2 × 25):

- 1 以下哪个不是嵌入式系统的设计的三个阶段之一: ( D )  
A 分析 B 设计 C 实现 D 测试
- 2 以下哪个不是 RISC 架构的 ARM 微处理器的一般特点: ( C )  
A 体积小、低功耗 B 大量使用寄存器  
C 采用可变长度的指令格式, 灵活高效 D 寻址方式灵活简单
- 3 Xscale 中, DMA 控制器具有多少个有优先级的通道, 可为内部外设和外部芯片提供服务?: ( B )  
A 15 B 16 C 17 D 18
- 4 BOOTP 主要是用于无磁盘的客户机从服务器得到: ( D )  
A 目标板的 IP 地址 B 服务器的 IP 地址 C 网关 IP 地址 D ABC
- 5 通常所讲的交叉编译就是在 X86 架构的宿主机上生成适用于 ARM 架构的 ( A ) 格式的可执行代码。  
A elf B exe C pe D sh
- 6 下面不属于 Boot Loader 阶段 1 所完成的步骤的是: ( C )  
A 硬件设备初始化。  
B 拷贝 Boot Loader 的阶段 2 到 RAM 空间中。  
C 将 kernel 映像和根文件系统映像从 Flash 读到 RAM 空间中。  
D 设置堆栈。
- 7 以下哪个不是 ARM 的 7 种运行状态之一: ( B )  
A 快中断状态 B 挂起状态 C 中断状态 D 无定义状态
- 8 在 x86 处理器上, Linux 系统调用是通过自陷指令 ( A ) 实现的。  
A INT 0x80 B INT 0x40 C INT 0x20 D INT 0x10
- 9 用以下的哪个命令可以把 server 的 /tmp mount 到 client 的 /mnt/tmp 并且是 read only ( A )  
A mount -o ro server:/tmp /mnt/tmp  
B mount -o ro /mnt/tmp server:/tmp  
C mount -o ro client:/mnt/tmp server:/tmp

- D mount -o ro server/tmp client/mnt/tmp
- 10 以下对 GDB 可以完成的任务描述正确的是: ( D )
- A 运行程序,可以给程序加上所需的调试任何条件
  - B 在给定的条件下让程序停止
  - C 检查程序停止时的运行状态
  - D ABC 中的任务都可以完成
- 11 以下哪个不是 GDB 中断点的四种状态之一: ( C )
- A 有效 B 禁止 C 指定次数有效 D 有效后删除
- 12 Linux 操作系统支持多种设备,这些设备的驱动程序不包括以下的那一项特点 ( C )
- A 设备驱动可以使用标准的内核服务如内存分配、中断和等待队列等。
  - B 大多数 Linux 设备驱动可以在需要的时候加载到内核,同时在不再使用时被卸载。
  - C 当系统启动及设备驱动初始化后,驱动程序将维护其控制的设备。如果一个特有的设备驱动程序所控制的物理设备不存在,将会影响整个系统的运行。
  - D Linux 设备驱动程序可以集成为内核的一部分。在编译内核的时候,可以选择把哪些驱动程序直接集成到内核里面。
- 13 以下哪个 GUI 是由中国人主持的一个自由软件项目: ( A )
- A MinuGUI B OpenGUI C MicroWindows D Qt/Embedded
- 14 嵌入式 GUI 设计不包括下面哪项: ( B )
- A 驱动程序设计 B 程序逻辑设计
  - C 用户界面程序设计 D 硬件设计
- 15 下面不属于使用 CPLD/FPGA 可编程逻辑器件来开发数字电路的优点的是 ( C )
- A 大大缩短设计时间
  - B 减少 PCB 面积
  - C 增加开发费用
  - D 提高系统的可靠性
- 16 若内存按字节编址,用存储容量为  $32K \times 8$  比特的存储器芯片构成地址编号 A0000H 至 DFFFFH 的内存空间,则至少需要 ( C ) 片。
- A. 4 B. 6 C. 8 D. 10
- 17 设指令由取指、分析、执行 3 个子部件完成,每个子部件的工作周期均为  $\Delta t$ ,采用常规标量单流水线处理机,若连续执行 10 条指令则共需时间 ( C )  $\Delta t$ 。
- A. 8 B. 10 C. 12 D. 14
- 18 在下列调度算法中, ( A ) 算法不会出现任务“饥饿 (starvation)”的情形。
- A. 时间片轮转算法 B. 先来先服务算法
  - C. 可抢占的短作业优先算法 D. 静态优先级算法
- 19 以下不属于网络安全控制技术的是 ( D )。
- A. 防火墙技术 B. 访问控制技术

- C. 入侵检测技术                      D. 差错控制技术
- 20 “冲击波”病毒属于( A )类型的病毒,它利用 Windows 操作系统的( )漏洞进行快速传播。
- (1) A. 蠕虫                      B. 文件                      C. 引导区                      D. 邮件
- (2) A. CGI 脚本                      B. RPC                      C. DNS                      D. IMAP
- 21 某幅图像具有  $640 \times 480$  个像素点,若每个像素具有 8 位的颜色深度,则可表示( A )种不同的颜色,经 5:1 压缩后,其图像数据需占用( ) Byte 的存储空间。
- (1) A. 8                      B. 256                      C. 512                      D. 1024
- (2) A. 61440                      B. 307200                      C. 384000                      D. 3072000
- 22 在下面的叙述中, ( D )不是嵌入式图形用户接口 (GUI) 的主要特点。
- A. 运行时占用的系统资源少                      B. 模块化结构,便于移植和定制
- C. 可靠性高                      D. 美观华丽,图形算法复杂
- 23 一个 4 位的二进制计数器,由 0000 状态开始,经过 25 个时钟脉冲后,该计数器的状态为( C )。
- A. 1100                      B. 1000                      C. 1001                      D. 1010
- 24 以下叙述中,不符合 RISC 指令系统特点的是( B )。
- A. 指令长度固定,指令种类少
- B. 寻址方式种类丰富,指令功能尽量增强
- C. 设置大量通用寄存器,访问存储器指令简单
- D. 选取使用频率较高的一些简单指令
- 25 通常所说的 32 位微处理器是指( C )。
- A. 地址总线的宽度为 32 位                      B. 处理的数据长度只能为 32 位
- C. CPU 字长为 32 位                      D. 通用寄存器数目为 32 个

## 二. 简答与名词解释 (5 × 6):

1. 简述 Linux 在嵌入式系统市场上取得辉煌的成果的原因。

- 广泛的硬件支持
- 内核高效稳定
- 开放源码,软件丰富
- 优秀的开发工具
- 完善的网络通信和文件管理机制

2. 简述嵌入式系统平台移植所需要的步骤。

- 硬件平台的移植
- 引导/装载程序的移植
- 内核的修改配置编译
- 相关驱动程序的移植
- 文件系统的移植
- 开发环境的移植
- 应用程序的移植

3. arm 系列处理器 arm9ejs 中的 ejs 三个字母的含义。

S: 可综合的软核 Softcore

E: 具有 DSP 的功能

J: Jazeller, 允许直接执行 Java 字节码

4. MMU 的含义及主要工作。

MMU, 也就是“内存管理单元”(memory management unit)。

其主要作用是两个方面: 一是对地址映射; 二是对地址访问的保护和限制。

5. 现在有一个空的 XSBASE255 开发板, 要将已制作好的 Linux 内核映像 zImage 和根文件系统映像 rootfs.img 传输到开发板上并启动 Linux 需要经过哪些步骤。

- 连接 JTAG 线和 COM 口

- 利用 JTAG 烧写 BootLoader, 并启动 BootLoader

- 在宿主机上配置 BOOTP 服务和 TFTP 服务

- 在目标板上使用 bootp 命令获取 ip, 使用 tftp zImage kernel 和 tftp zImage rootfs.img 分别传输内核和根文件系统映像

- 使用命令 boot 启动

6. 现在在宿主机上已编译好的 gdbserver 和测试程序 test, 宿主机和目标机的 ip 地址分别为 192.168.0.100 和 192.168.0.50。简述利用 GDB 进行远程调试的步骤及命令。

- 利用 zmodem 传输 gdbserver 和 test 到目标板上。

- 在宿主机上启动 gdbserver 进行监听, 命令为:

```
/gdbserver 192.168.100.216:1234 test
```

- 拷贝 test 程序到宿主机的 arm-gdb/bin 目录下, 并执行命令:

```
/arm-linux-gdb test
```

- 连接开发板, 命令为:

```
(gdb) target remote 192.168.100.50:1234
```

1) 1. 嵌入式系统的三要素是嵌入、专用、计算机。

2. 从嵌入式系统设计的角度来看, 嵌入式软件结构可以分为循环轮询系统、前后台系统、单处理器多任务系统以及多处理器多任务系统等几大类。

3. 衡量系统实时性的主要指标有: 响应时间、生存时间、吞吐量。

4. 软件一般包括: 程序、数据和文档。

5. 嵌入式软件的体系结构通常包括: 驱动层、操作系统层、中间件层和应用层。

6. 嵌入式系统中的任务管理主要包括: 创建任务、删除任务、改变任务状态和查询任务状态



## 《嵌入式系统与开发》

### 一. 填空题

1. 嵌入式系统以应用为中心, 以计算机技术为基础, 软, 硬件可裁减, 适应应用系统对功能、体积、功耗等严格要求的专用计算机系统。

2. S3C2440 开发板内部集成看门狗部件, 其是 A (A. 硬件 B. 软件) 部件设备。从本质上来说看门狗是定时器, 但也可以用来监视系统的运行, 在 B (A. 硬件 B. 软件) 发生故障时, 产生复位信号, 以使系统进行重启。

3. Linux 设备分为 字符 设备、块 设备和 网络 设备。

4. 如果把嵌入式设备启动看作接力赛跑, 系统刚启动时运行的是 引导程序 bootloader 代码, 然后运行 内核 代码, 最后加载文件系统, 完成系统的各项初始化。

### 二. 简单题

#### 1. 简述非操作系统和操作系统开发的优缺点

答: 非操作系统下程序员拥有更大的自由操纵系统资源, 但是由此也会带来开发效率低下, 不正确的操纵系统资源;

操作系统下, 程序员使用资源需在操作系统的约束下使用, 但是, 可以使用操作系统通过的各种功能, 开发效率提高了。

2. 在 linux 环境下编写一模块, 要求加载模块时打印字符串 “hello module is inserted into kernel”, 卸载模块时打印字符串 “hello module is unloaded from kernel”。编写模块代码文件、Makefile 并说明如何加载进内核。

```
/*
*
*=====
*=====
*
*      Filename:  hello.c
*
*      Description:  This is the module hello for linux kernel (my box is
*      2.6.38 version ,mageia 1).
*
*      Version:  1.0
*      Company:  NJUPT
*
*=====
*=====
*/
#ifdef __KERNEL__
#define __KERNEL__
#endif
```

```

#ifndef MODULE
#define MODULE
#endif
#include<linux/module.h>
#include<linux/kernel.h>
#include<linux/init.h>
static int __init hello_init()
{
    printk("hello,module is inserted into kernel\n");
    return 0;
}
void __exit hello_exit()
{
    printk("hello, module is unloaded form kernel\n");
}
module_init(hello_init);
module_exit(hello_exit);
MODULE_LICENSE("BSD");
# filename : Makefile
# This is the Makefile for hello.ko module
obj-m:=hello.o
KDIR:=/lib/modules/`uname -r`/build
PWD:=$(shell pwd)
all:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules
[root@localhost tmp]# insmod hello.ko
[root@localhost tmp]# dmesg | tail -1
hello,module is inserted into kernel
[root@localhost tmp]# rmmod hello.ko
[root@localhost tmp]# dmesg | tail -1
hello, module is unloaded form kernel

```

3.简述 Linux 开发板烧写 bootloader、内核和文件系统过程，并简单描述应用程序编辑、编译、下载和运行过程。

答:

bootloader 的建立（以 vivi 为例说明）:

1) 拷贝 vivi 源码到目录下;

- 2) 根据开发板进行配置: 选择“Load on Alternate Configuration File”菜单载入配置文件, 在输入框写入“arch/def-configs/smdk2440”→选择“OK”→选择 Exit→选择 Yes, 然后退出
- 3) 运行 make 编译: #make
- 4) 烧写 vivi 到开发板 (使用仿真器完成): 连接 SinoSys-ICE16 到实验箱和 PC, 打开实验箱电源。将/Bootloader/BurnFlash/目录下的程序 Jflash-s3c2440 拷贝出来。然后通过 Jflash-s3c2440 烧写 vivi 到实验箱的 NandFlash 中。
- 5) 启动开发板后, vivi 就已运行。

内核烧写:

- 1) 拷贝 kernel 源代码到工作目录中;
- 2) 进入 s3c2440\_kernel12.4.18\_rel, 并且编译: 选择“Load on Alternate Configuration File”菜单载入配置文件, 在输入框写入“arch/def-configs/smdk2440”, 选择“OK”, 打开菜单各个页, 查看配置文件的默认选项。可用空格键或回车键来改变选项。
- 3) 运行 make 编译, 生成内核映像文件。
- 4) 利用 JTAG 对 bootloader 进行烧写: 连接好 SinoSys-ICE16。打开实验箱电源。将/Bootloader/BurnFlash/下的程序 Jflash-s3c2440 拷贝出来。然后通过 NandFlash 192K 地址开始进行烧写。

构建文件系统:

- 1) 拷贝 busybox 源代码到工作目录中;
- 2) 编译 busybox: 运行 make menuconfig 可以打开它的编译界面。
- 3) 增加必要的文件;
- 4) 建立文件系统的配置目录;
- 5) 测试新的文件系统: 1.修改/etc/exports 文件; 2.激活 portmap 与 nfs 服务; 3.进行连接
- 6) 再次裁剪库文件;

### 三. 设计题

1、以 S3C2440 开发板为例, 编写基于 linux 平台下的看门狗硬件驱动, 并编写应用程序, 要求在应用程序中打开看门狗, 开发板上看门狗最长看门时间为 5 秒, 保证系统不会因为看门狗到时而造成自动重启。

具体要求:

- 编写设备驱动及简述编译、加载过程
- 编写应用程序并简述编译运行步骤

答:

设备驱动源码

//硬件接口函数

int WATCHDOG\_DEV=0;

```

static int watchdog_open(struct inode *inode, struct file *file)
{
    printk("watchdog device will be opened");
    if (WATCHDOG_DEV)
        Return -EBUSY;
        WATCHDOG_DEV++;
        MOD_INC_USE_COUNT;
        WTCNT=65535;
        WTCN=0xff39;
        Return 0;
}

static int watchdog_release(struct inode *inode, struct file *file)
{
    MODE_DEC_USE_COUNT;
    if(!(MOD_IN_USE))
    {
        WTCN=0x0;
    }
    return 0;
}

static int watchdog_write(struct file * file, const char * buffer, size_t count, loff_t
*ppos)
{
    int wdtcnt_val;
    copy_from_user(&wdtcnt_val, buffer, sizeof(int));
    WTCNT=wdtcnt_val;
    return 0;
}

//建立文件系统与设备驱动程序的接口定义
static struct file_operations watchdog_fops = {
    open: watchdog_open,
    release: watchdog_release,
    write: watchdog_write,
};

//注册、注销设备
#define DEVICE_NAME "watchdog"
#define WATCHDOG_MAJOR 234
static int __init watchdog_init(void)
{
    int ret;
    ret = register_chrdev(WATCHDOG_MAJOR,
        DEVICE_NAME, &watchdog_fops);
    if (ret < 0) {
        printk(DEVICE_NAME " Can't initial the watchdog device\n");
    }
}

```

```

        return ret;
    }
    return 0;
}
static void __exit watchdog_exit(void)
{
    int ret;
    ret=unregister_chrdev(WATCHDOG_MAJOR, DEVICE_NAME);
    if(ret<0)
    {
        printk(DEVICE_NAME " Can't exit the watchdog device\n");
    }
    return 0;
}
module_init(watchdog_init);
module_exit(watchdog_exit);
MODULE_LICENSE("GPL");

```

模块编译并加载:

1) 编写 Makefile

```

obj-m := watchdog.o
KDIR := 内核源码目录
PWD := $(shell pwd)
all:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

```

2) Make

3) insmod watchdog.ko

应用程序源码:

```

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <errno.h>
#include <pthread.h>
void * feeddogthread() {
    int feeddogvalue;
    int returnval;
    feeddogvalue=65535;
    while(1) {
        //每隔 5 秒，将重载看门狗计数寄存器的值

```

```

        printf("feed dog \n");
        returnval=write(watchdogfd, &feeddogvalue, sizeof(int));
        sleep(5); }
}

int main()
{
    pthread_t watchdogThd;
    int watchdogfd;
    int returnval;
    char ch;
    // 打开看门狗设备
    if((watchdogfd=open("/dev/watchdog", O_RDWR|O_NONBLOCK))<0) {
        printf("cannot open the watchdog device\n");
        exit(0);
    }
    // 创建喂狗线程
    returnval=pthread_create(&watchdogThd, NULL, feeddogthread, NULL);
    if(returnval<0)
        printf("cannot create feeddog thread\n");
    while(1){ .....}
}

```

```

        return ret;
    }
    return 0;
}
static void __exit watchdog_exit(void)
{
    int ret;
    ret=unregister_chrdev(WATCHDOG_MAJOR, DEVICE_NAME);
    if(ret<0)
    {
        printk(DEVICE_NAME " Can't exit the watchdog device\n");
    }
    return 0;
}
module_init(watchdog_init);
module_exit(watchdog_exit);
MODULE_LICENSE("GPL");

```

模块编译并加载:

- 1) 编写 Makefile
 

```

obj-m := watchdog.o
KDIR := 内核源码目录
PWD := $(shell pwd)
all:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

```
- 2) Make
- 3) insmod watchdog.ko

应用程序源码:

```

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <errno.h>
#include <pthread.h>
void * feeddogthread() {
    int feeddogvalue;
    int returnval;
    feeddogvalue=65535;
    while(1) {
        //每隔 5 秒，将重载看门狗计数寄存器的值

```

```

        printf("feed dog \n");
        returnval=write(watchdogfd, &feeddogvalue, sizeof(int));
        sleep(5); }
}

int main()
{
    pthread_t watchdogThd;
    int watchdogfd;
    int returnval;
    char ch;
    // 打开看门狗设备
    if((watchdogfd=open("/dev/watchdog", O_RDWR|O_NONBLOCK))<0) {
        printf("cannot open the watchdog device\n");
        exit(0);
    }
    // 创建喂狗线程
    returnval=pthread_create(&watchdogThd, NULL, feaddogthread, NULL);
    if(returnval<0)
        printf("cannot create feaddog thread\n");
    while(1){ .....}
}

```



# 《嵌入式系统与开发》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

得分

## 一、单项选择题 (共 20 分, 每题 2 分)

- 下列哪一项不是 ARM 处理器的典型特点 C。  
A. 体积小、功耗低、高性能、低成本 B. 具有大量寄存器  
C. 采用 CISC 架构 D. 采用 LOAD/STORE 指令完成内存和寄存器间数据的传输
- ARM 处理器最主要的应用场合为 A。  
A. 无线通讯领域 B. 图像处理领域 C. 安全领域 D. 存储设备领域
- 下列 ARM 处理器中, 哪款属于 ARM920T 核 B。  
~~ARM920T~~  
A. S3C440X B. S3C2440 C. XSCALE255 D. XSCALE270
- 下列哪条指令能够完成跳转功能, 同时可以将返回地址保存到 R14 寄存器中, 但是并不需要进行状态切换 B。  
A. B 指令 B. BL 指令 C. BLX 指令 D. BX 指令
- 下列那个选项不属于 ARM 指令 D。  
A. SWI ✓ B. ADD ✓ C. MOV ✓ D. ENTRY
- 下列那个选项不是 S3C2440 处理器内部集成的接口部件 D。  
A. 看门狗 ✓ B. RTC 实时时钟 ✓ C. LCD 控制器 ✓ D. 网络控制器
- 在 LINUX 系统中, 要将 hello.c 文件编译成 hello.o 目标文件 (注意不是可执行文件), 应该采用下列哪条命令来完成 B。  
A. gcc -o hello.o hello.c B. gcc -c -o hello.o hello.c  
C. gcc -S hello.c D. gcc -E hello.c

8. 下列哪一个选项中的规则不是实目标生成规则 C

A. hello:hello.c

B. kk.o:kk.c

gcc -o hello hello.c

gcc -c -o \$@ \$<

C. clean:

D. hello-fl.o f2.o

rm -rf \*.o

gcc -o hello fl.o f2.o

9. 下列哪个函数用于派生子进程，并且派生的子进程一定优先于父进程被执行 B

A. fork() B68 B. vfork() B68 C. exec() B68 D. system()

10. 如果编写了一个设备驱动模块，应该采用下列哪条命令将该驱动模块加载到linux内核中 A

A insmod B. rmmod C. lsmod D. locate

得分

二、填空题 (共 20 分, 每空 1 分)

1. ARM 是 Advanced RISC Machine 的英文缩写。

2. 嵌入式系统以 芯片 为中心, 以计算机技术为基础, 软硬件 可裁减, 适应应用系统对 功耗、成本、体积、功耗 严格要求的专用计算机系统。

3. 在 ARM 存储系统中, 支持的数据类型为 word、half-word 和 byte 三种类型。

4. ARM 微处理器中共有 37 个寄存器, 在这些寄存器中, R13 寄存器的作用为 程序计数器, R14 寄存器的作用为 链接寄存器, R15 寄存器的作用为 程序计数器。

5. 假设模块名为 test\_char.o, 请问如果要将该模块加载到 Linux 内核中, 使用命令为 insmod。

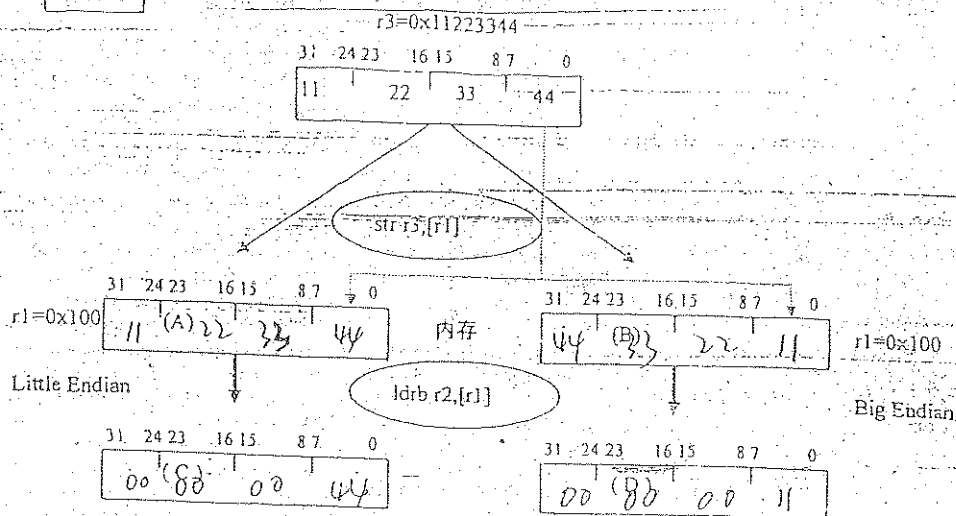
6. gcc 的整个编译过程分为 预处理、编译、汇编 和 链接 四步。如果有一个 C 源文件 (名称为 test.c), 编译后的可执行文件名为 test, 写出相应的 gcc 编译命令 gcc -o test test.c

-o file 将输出放在文件 file (-o test)

得分

### 三、简答题（共 30 分，每题 10 分）

1. ARM 存储体系支持大小端模式，根据下图回答问题。



(1) 简述大小端模式，并画示意图。

(2) 在上图中，假设寄存器  $r3$  的值为  $0x11223344$ ， $r1$  寄存器的值为  $0x100$ ，使用命令  $str\ r3, [r1]$  后，内存  $0x100$  开始的字单元存放的内容为什么（根据大小端模式分别回答）？

答：大端 44 33 22 11  
小端 11 22 33 44

(3) 假设  $r2$  寄存器的值为  $0x0$ ，在上图中使用  $ldrb\ r2, [r1]$  后， $r2$  的值分别为什么（根据大小端模式分别回答）？

2. 假设 temp 目录下有三个 C 源码文件：main.c fl.c f2.c，其内容分别如下。

```
//main.c
void main()
{
    printf("hello1");
    printf("hello2");
}
```

```
//fl.c
#include <stdio.h>
void printfhello1()
{
    printf("hello-1\n");
}
```

```
//f2.c
#include <stdio.h>
void printfhello2()
{
    printf("hello-2\n");
}
```

1. 读入所有的 makefile
2. 读入读 include 的其他 makefile
3. 初始化文件中的变量
4. 按与依赖规则, 并重新所有规则
5. 为所有的目标文件中创建依赖关系链
6. 根据依赖关系, 决定哪些目标需要重新生成
7. 执行生成命令

回答下列问题:

(1) 简述 make 工具执行的流程

(2) 编写 Makefile 文件, 要求:

a) 首先将上述几个源码文件首先编译成目标文件

b) 然后将目标文件链接成可执行文件 hello

c) 执行 make clean 时能够将所有目标文件和可执行文件删除

3. 假设要编写两个整数相减函数和两个整数大小比较函数, 并编写 main 函数, 完成下列功能: 在 main 函数中调用相减函数, 相减函数首先完成两个整数相减, 如果结果为负数, 再调用求绝对值函数, 在求绝对值函数中, 将传进来的参数求绝对值并返回, 要求 main 函数和相减函数用 C 代码实现, 求绝对值函数用 ARM 汇编代码实现。

```

1) Objects = main.o f1.o f2.o
hello: $(Objects)
gcc -o hello $(Objects)

main.o: main.c
gcc -c main.c

f1.o: f1.c
gcc -c f1.c

f2.o: f2.c
gcc -c f2.c

clean:
rm hello $(Objects)

```

```

2. #include <stdio.h>
extern void ABS(int *a)
int sub(int a, int b)
{ return a-b; }
int main(void)
{ int m=4, n=5;
  int num;
  num = sub(m,n);
  if (num < 0) ABS(&num);
  printf("%d\n", num);
  return 0; }

AREA ABS, CODE, READONLY
EXPORT ABS
MOV R1, #0
MOV R0, #0
MOV PC, #0
END

```

得分

四、设计题 (共 30 分, 每题 15 分)

1. 编写一字符设备驱动, 要求以模块的形式加载, 加载模块时打印字符串 "I LOAD A CHAR DEVICE", 卸载模块时打印字符串 "I UNLOAD A CHAR DEVICE". 编写 Makefile 并说明如何加载。

2. 在嵌入式 S3C2440 处理器中, 有看门狗模块, 要求完成下列内容:

- (1) 在无 OS 支持情况下, 要求编写接口代码, 初始化看门狗, 并打开看门狗; 编写喂狗函数。
- (2) 在 LINUX 下, 编写看门狗驱动代码, 并编写应用程序, 在应用程序中首先打开看门狗, 然后创建子进程, 子进程调用喂狗函数, 父进程中则每隔 5 秒中输出下列一句话: I am alive now

```

1. /* device.c */
#include <linux/kernel.h>
#include <linux/module.h>
int init_module()
{ printk("I LOAD A CHAR DEVICE\n");
  return 0; }

int cleanup_module()
{ printk("I UNLOAD A CHAR DEVICE\n");
  return 0; }

```

```

// Makefile
cc = gcc -Wall -DMODULE -DKERNEL -DLinux
device.o: device.c
cc -c -o device.o device.c

install:
  insmod device.o

uninstall:
  rmmod device.o

```

第 4 页共 4 页  
加载时, 执行 make  
卸载时, 执行 make uninstall

信  
考  
试  
绝  
不  
作  
弊