15元

南京邮电大学 2013/2014 学年第 一 学期

《 嵌入式系统及应用 》期末 试卷 A

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_____，执行指令 DIV AB 需要时间_____ us。若要该单片机可靠复位，必须在 RST 引脚保持_____ us 以上的高电平。

2. MCS-51 有 5 个中断源，提供_____个中断优先级。若 IE=10011010B、IP=00011000B，则能响应的中断源有_____，当所有中断源同时提交申请时，则最先响应_____中断。

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

4. ARM 字数据存储格式有大端格式和_____，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_____，属于_____对齐方式。

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

LDR R1, [R0], #6 访问的存储单元地址为 0x_____；

执行该指令后 R0 的值为 0x_____；

LDR R1, [R0, R0, LSL R1] 访问的存储单元地址为 0x_____；

自觉遵守考试规则，诚信考试，绝不作弊 装 订 线 内 不 要 答 题

得分	

二、单一选择题（每题 1.5 分，共 30 分）

1. （ ）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. （ ）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. （ ）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. （ ）若采用 8031 作为控制芯片，则 EA/Vpp 应该连接至。
   A. VCC　　B. GND
   C. 21V　　D. 12V

6. （ ）下列指令中错误的是。
   A. SETB 50H.0　　B. MOV A, B
   C. JNC LOOP　　D. SUBB A, R0

7. （ ）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. （ ）MCS-51 单片机的堆栈区应建立在。
   A. 片内数据存储区　　B. 特殊功能寄存器区
   C. 片外数据存储区　　D. 程序存储区

9. （ ）若使用 DJNZ R2,LOOP 对 R2 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

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. 提高定时的精度　　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 1143

4. ARM 处理器中，ARM 指令集和 Thumb 指令集的区别。AMR 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）

```
ORG   0000H
AJMP  START
ORG   0100H
START: MOV   DPTR, #8000H
       MOV   P2, #90H
       MOV   R0, #00H
       MOV   R1, #50H
LOOP:  MOVX  A, @DPTR
       MOVX  @R0, A
       INC   DPTR
       INC   R0
       DJNZ  R1, LOOP
       RET
```

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）

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
      BNZ   Loop
      END
    EXIT  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
    CMP   R0, R1
    MOVLT R0, R1
    CMP   R0, R2
    MOVLT R0, R2
    MOV   PC, LR
```

得 分

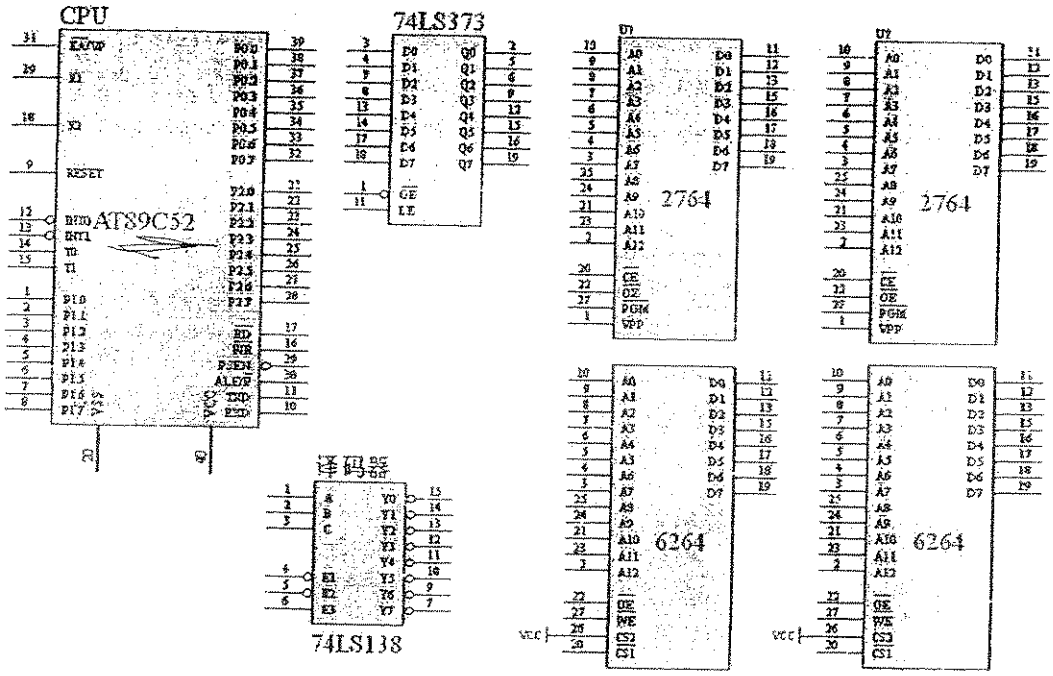
五、综合题（10 分）

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



南京邮电大学 2012/2013 学年第 2 学期

《嵌入式系统及应用 》期末 试卷

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. 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。

3. 80C51 有 2 级中断， 5 个中断源。

4. 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。

5. MCS-51 布尔处理机的存储空间地址范围是 0H～7FH 。

6. 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。

7. 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。

8. 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。

9. 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。

10. 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 。

11. 当使用 8031 单片机时，需要扩展外部程序存储器，此时 $\overline{EA}$ 应为 低电平 。

9

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.2μs　　B.4μs　　C.8μs　　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

```
SUBP: MOV R1    A
      ANL   A   #0F0H
      SWAP  A
      CJNE  A, #10, NEXT1
      LJMP  ERROR
NEXT1: JNC ERROR
       MOV B    #10
       MUL AB
       XCH A    R1
       ANL A    #0FH
       CJNE A   #10   NEXT2
       LJMP  ERROR
```

```
NEXT2:  JNC ERROR
        ADD A    R1
        RET ERROR
        MOV A    #255
        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，已知单片机晶体振荡器的频率为 f_{osc}＝12MHz，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/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

1)Tc = 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
      S[illegible]2A
      A,#0
```

```
      ORG 2000H
INQP: MOV TH0, #04H;
      MOV TL,#30H
      CPL P1.0
       RETI
```

自觉遵守考试规则，诚信考试，绝不作弊

12

南京邮电大学 2012/2013 学年第 一 学期

《 嵌入式系统及应用 》期末 试卷 B 附答案

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 装 订 线 内 不 要 答 题

得 分

一、填空题（每空 1 分，共 25 分）

1. 执行指令

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

则 A=_____H ， B =_____H ， C=____ ， AC=_____， P=____ 。

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。

3. 读取片内 RAM 90H 单元至 A 累加器的指令

_____、_____。

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

_____、_____、_____。

读取片外 RAM 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。

5. 如果 R0=0x8000，则

```
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）

1. 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
2. 以下指令错误的是(　　)
   A. SETB C　　B. SETB P0.1
   C. SETB F9H　　D. SETB F9H.0
3. 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
4. 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
5. 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
6. 以下哪个是有效的 8 位位图(　　)
   A. 0x12345678　　B. 0x00000123
   C. 0xFF000000　　D. 0xCC800000
7. 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
8. 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
9. 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}^
10. 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

14

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（MCS51）

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）

16

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：

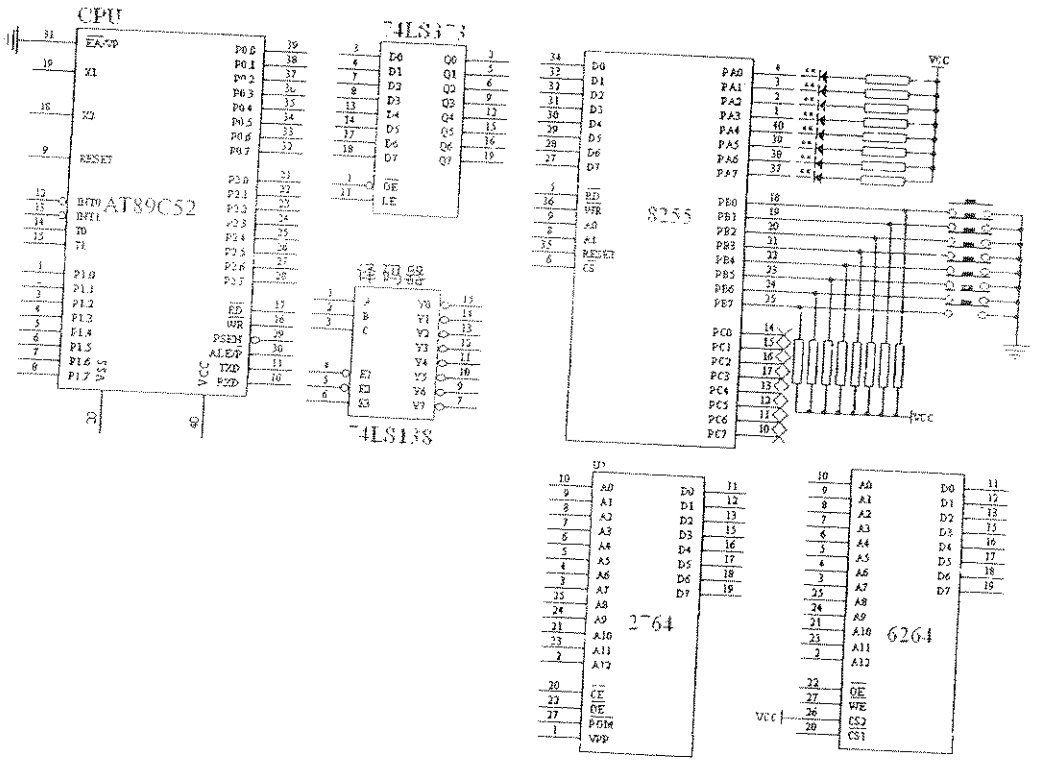
要求：

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)



2. 当按键被按下时，点亮相应的 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）

19

南京邮电大学 2012/2013 学年第 一 学期

《 嵌入式系统及应用 》期末 试卷 B

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 、MOVC A, @A+DPTR；
   MOV DPTR, #0090H、MOVX A,@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
    MOVLO R0,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
```

10九

南京邮电大学 2011/2012 学年第 一 学期

《 嵌入式系统及应用 》期末 试卷 A

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 装订线内不要答题

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=BC 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 07H。
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1 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)

25

7. 若 R0=0x8000，R1=0x0004，则：

LDR R2,[R0]	访问的存储器单元地址为 0x 8000
LDR R2,[R0,#0x10]	访问的存储器单元地址为 0x 8010
LDR R2,[R0,R1]	访问的存储器单元地址为 0x 8004
LDR R2,[R0, R1, LSL #1]	访问的存储器单元地址为 0x 8008
LDR R2,[R0, #0x04]!	访问的存储器单元地址为 0x 8004，执行指令后，R0=0x 8004
LDR R2,[R0],#0x04	访问的存储器单元地址为 0x8000，执行指令后，R0=0x 8004
STMIA R0!,{R5-R8}	R5 存储的单元地址为 0x 800C，执行指令后，R0=0x 8010

得 分

二、单一选择题（每题 1.5 分，共 30 分）

1. 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、INT1 D. T0、INT0、T1、INT1、串口
2. 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
3. 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
4. 采用 DJNZ R2,LOOP 指令循环时，若想使循环次数最多，R2 的值应为(A)。
   A. 0 B. 1 C. 255 D. 256
5. 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
6. 执行 MOV DPTR,#1234H 后，DPH 的值为(A)。
   A. 12H B. 34H C. 00H D. 0FFH
7. 定时器/计数器工作在(C)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
8. 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=(C)
   A. 0100H B. 0101H C. 0102H D. 0103H
9. 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. ANL A,#7FH
   C. ORL A,#7FH D. MOV A,#00H
10. 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(　　)。
A．支持 Embeded 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(　　)。
A．0x8000　B．0x8004　C．0x8008　D．0x800C

13. 上电复位后，ARM7 处在(　　)
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(　　)
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．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．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(　　)
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(　　)
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．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．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
    RL      A
    PUSH    ACC
    MOV     DPTR,#TABLE
    MOVC    A,@A+DPTR
    MOV     R1,A

    POP     ACC
    INC     DPTR
    MOVC    A,@A+DPTR
    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, #01000010B
        ANL   PCON, #7FH
        MOV   TMOD, #00100000
        MOV   TH1, #0FDH
        MOV   TL1, #0FDH
        SETB  TR1

        MOV   DPTR, #8000H

LOOP:
        MOVX  A, @DPTR
        JZ    EXIT
        INC   DPTR

WAIT:   JNB   TI, WAIT
        CLR   TI
        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, 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）

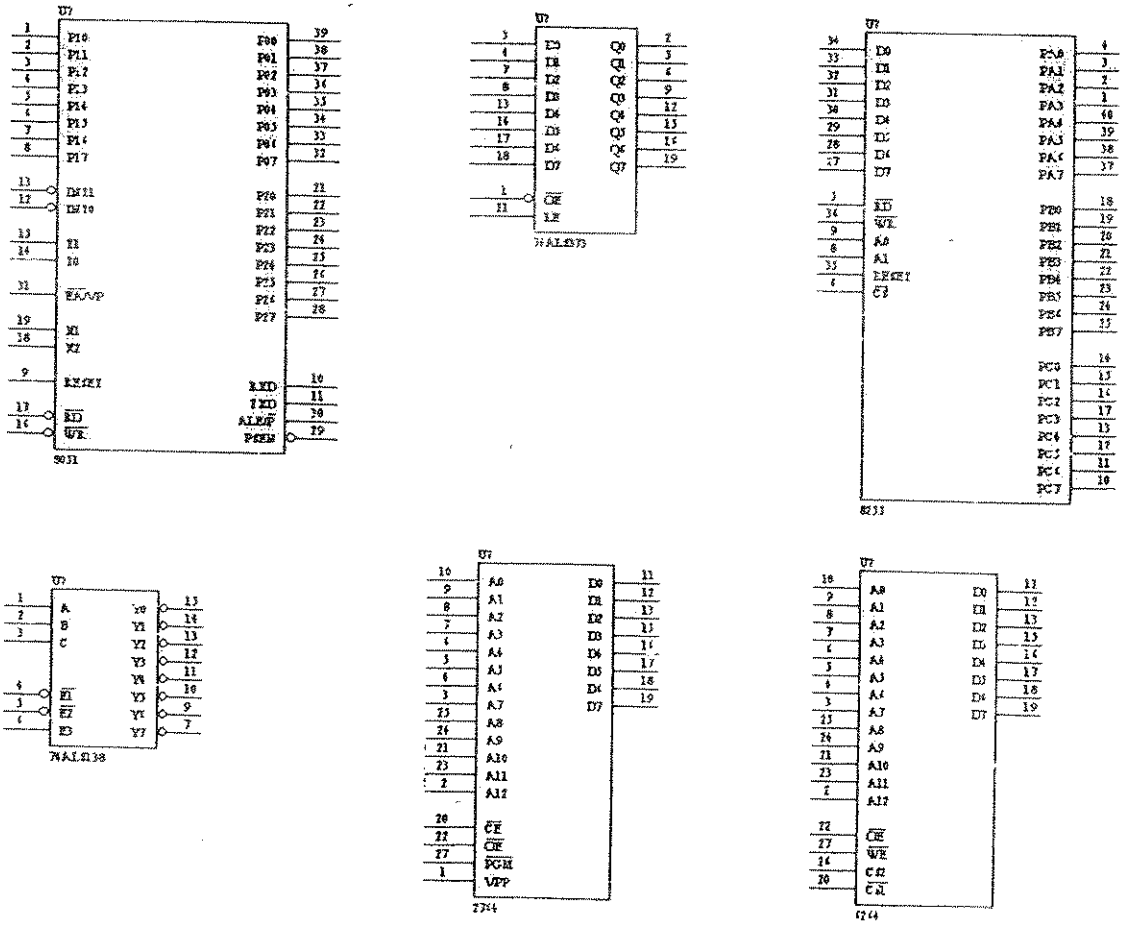
b) 外部程序存储器容量 8KB，地址空间与 80C51 的内部 ROM 相衔接

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

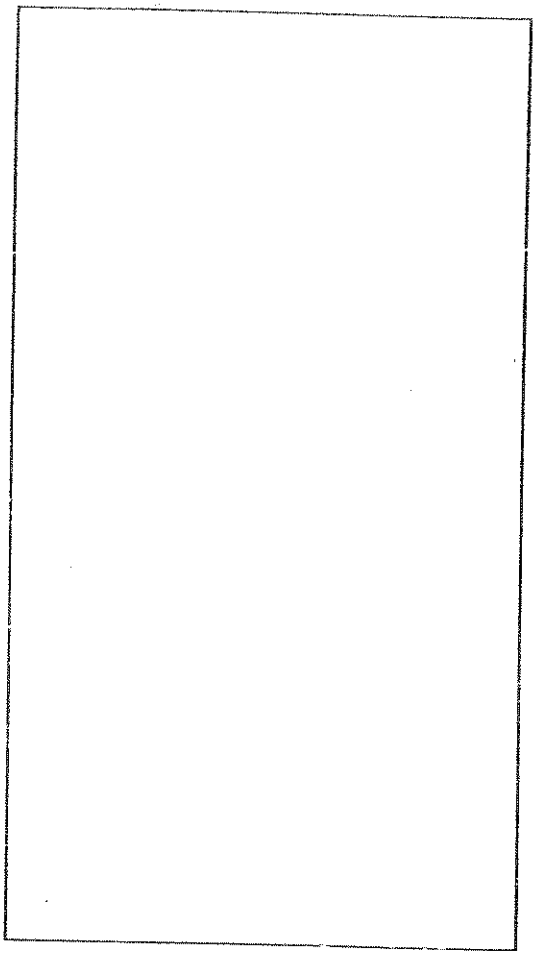
d) 扩展一片 8255A，口地址为：3FFCH～3FFFH

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）



南京邮电大学通达学院 2010/2011 学年第 一 学期

《嵌入式系统及应用 》期末 试卷（A）

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											

自觉遵守考试规则，诚信考试，绝不作弊 装订线内不要答题

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 $\overline{EA}$ 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 。

33

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哪几种

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
```


34

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，已知单片机晶体振荡器的频率为 f_{osc}=12MHz，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/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

1)Tc＝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;
```

自觉遵守考试规则，诚信考试，绝不作弊　装订线内不要答题

```
MOV TL0, #30H
SETB TR0
SETB ET0
SETB EA
AJMP $
```

```
MOV TL,#30H
CPL P1.0
RETI
```

南京邮电大学 2010/2011 学年第 一 学期

《 嵌入式系统及应用 》 期末试卷

院(系)_____ 班级_____ 学号_____ 姓名_____

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

自觉遵守考试规则，诚信考试，绝不作弊　装　订　线　内　不　要　答　题

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。

37

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）

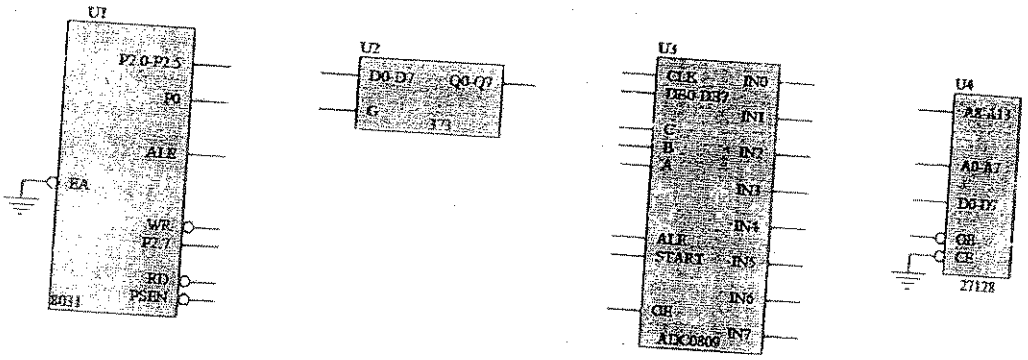
```
.....
START:   MOV   DPTR, #TABLE
         CJNE  A, #0aH, LOOP
         AJMP  ERR
LOOP:    JNC   ERR
         MOVC  A,
@A+DPTR
         MOV   B, A
         MOV   R1, #00H
EXIT:    SJMP  $

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
.....
```

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。



（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
IMPORT func
STR LR, [SP, #4]!
ADD R1, R0, R0
ADD R2, R1, R0
ADD R3, R1, R2
STR R3, [SP, #4]!
ADD R3, R1, R1
BL func
ADD SP, SP, #4
LDR PC, [SP], #4
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 fosc=11.0592MHz，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）

南京邮电大学 2007/2008 学年第 2 学期

《嵌入式系统》课程试卷

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×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△t，采用常规标量单流水线处理机，连续执行 10 条指令则共需时间(C)△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 640×480 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 内核映象 zImage 和根文件系统映象 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象
- 使用命令 boot 启动

6. 现有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

南京邮电大学 2010/2011 学年第 一 学期

《 嵌入式系统及应用 》期中试卷

院(系)_____ 班级_____ 学号_____ 姓名_____

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

自觉遵守考试规则，诚信考试，绝不作弊　装订线内不要答题

得 分

一、填空（25 分，每空 1 分）

1、若外接晶振为 6MHz，则 80C51 单片机的振荡周期为_____，机器周期为 2us，指令周期最短为 2us，指令周期最长为 8us。

2、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。

3、80C51P1 口的读操作有两种方式：读引脚 和 读锁存。实际应用中，读引脚前应先用输出指令置口锁存器 D=____，这就是准双向 I/O 口的特点。

4、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_____。

5、假定累加器 A 的内容为 30H，执行指令：1000H: MOVC A, @A+PC 后，把程序存储器_____H 单元的内容送累加器 A。（注：MOVC 为单字节指令）

6、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。

7、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。

8、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。

9、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。

10、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5位

000 1111 0 | 000 0 1100

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种功能？ 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，即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 E²PROM 在正常使用中只能读出不能写入，故 E²PROM 没有写入控制脚，只有读出引脚，读读引脚与 MCS-51 的 PSEN 相连。

3、简述如何通过片内定时器/计数器来实现外部中断源的扩展？

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
1 0111 1010

……

执行完后，将 PSW 各位状态填入下表：

CY	AC	F0	RSi	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 内容发送到 90H 单元中，（90H）=38H

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：_____。

得 分

四、编程题（25 分）

1、试编程实现在外部数据存贮器 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:    MOVC @A+DPTR, A    ;将数据放入数据空间
        INC  A
        DJNZ R2, LOP       ;判断是否存满
        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，已知单片机晶体振荡器的频率为 f_{osc}=6MHz，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/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

装订线内不要答题

南京邮电大学 2010/2011 学年第二学期

《嵌入式系统与开发》期末试卷

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
 *
 *
*=============================================================================
 */
#ifndef __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;

50

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 kenel 源代码到工作目录中；

2) 进入 s3c2440_kenel12.4.18_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;
        WTCON=0xff39;
        Return 0;
}
static int watchdog_release(struct inode *inode,struct file *file)
{
    MODE_DEC_USE_COUNT;
    if(!(MOD_IN_USE))
    {
        WTCON=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){ ..... }
}
```

南京邮电大学 2006 /2007 学年第 2 学期

《嵌入式系统与开发》期末试卷（A）

院(系)_____ 班级_____ 学号_____ 姓名_____

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

装 订 线 内 不 要 答 题

自觉遵守考试规则，诚信考试，绝不作弊

得分

一、单项选择题（共 20 分，每题 2 分）

1. 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

2. 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

3. 下列 ARM 处理器中，哪款属于 ARM920T 核 B 。

A、S3C44B0X B、S3C2440 C、XSCALE255 D、XSCALE270

4. 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，但是并不要进行状态切换 B 。

A、B 指令 B、BL 指令 C、BLX 指令 D、BX 指令

5. 下列那个选项不属于 ARM 指令 D 。

A、SWI B、ADD C、MOV D、ENTRY

6. 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

7. 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

55

(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
gcc -o hello hello.c

B、kk:o:kk.c
gcc -c -o $@ $<

C、clean:
rm -rf *.o

D、hello:f1.o f2.o
gcc -o hello f1.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() B、vfork() C、exec() 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 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 、多种性 、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 堆栈指针 ，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

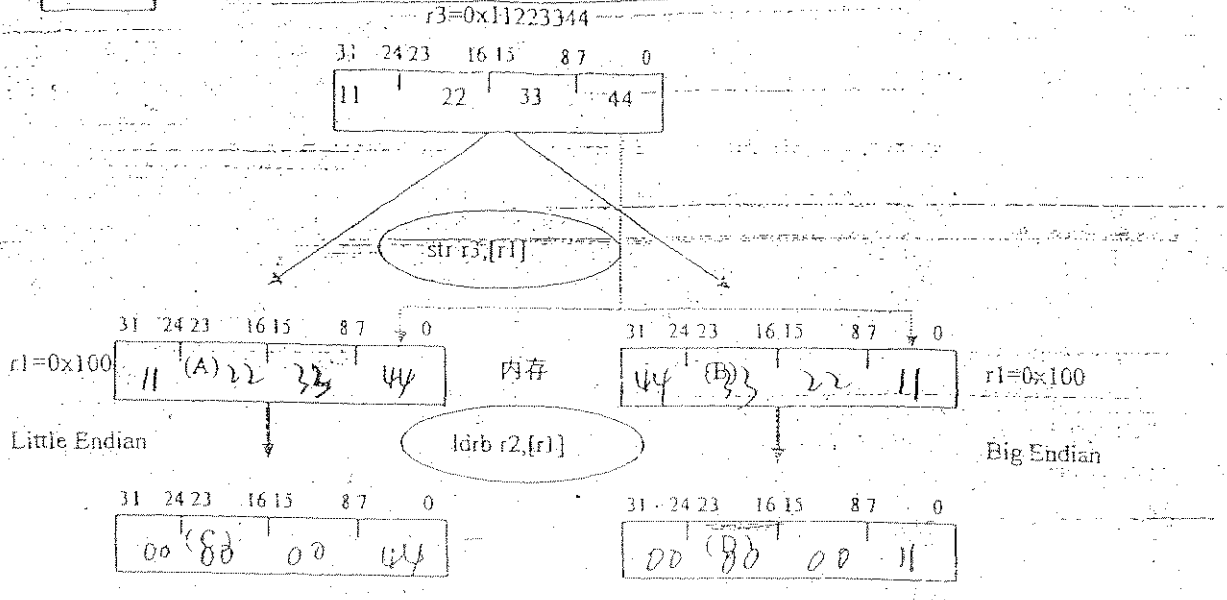
-o file 将输出存在文件 file (-o test)

56

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，并画图示意。

（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）？

（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 f1.c f2.c，其内容分别如下。

```
//main.c
void main()
{
    printfhello1();
    printfhello2();
}
```

```
//f1.c
#include <stdio.h>
void printhello1()
{
    printf("hello-1\n");
}
```

```
//f2.c
#include <stdio.h>
void printhello2()
{
    printf("hello-2\n");
}
```

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
```


60

```
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

```
SUBP: MOV R1   A
      ANL  A   #0F0H
      SWAP A
      CJNE A, #10, NEXT1
      LJMP ERROR
NEXT1: JNC ERROR
       MOV B   #10
       MUL AB
       XCH A   R1
       ANL A   #0FH
       CJNE A  #10   NEXT2
       LJMP ERROR
NEXT2:  JNC ERROR
        ADD A   R1
        RET ERROR
        MOV A   #255
        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，已知单片机晶体振荡器的频率为 f_{osc}=12MHz，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/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

1) Tc=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
```

嵌入式系统原理期末复习

嵌入式系统原理与开发期末复习概要

第一章　嵌入式系统概述

1.2.1 嵌入式系统的定义：

目前被大多数人接受的一般性定义是："嵌入式系统是以应用为中心，以计算机技术为基础，软硬件可裁剪，适应应用系统对功能、可靠性、成本、体积和功耗等严格要求的与用计算机系统。"

1.2.2 嵌入式系统的特点：

1.专用的计算机系统：形式多样、对运行环境的依赖性，综合考虑成本、资源、功耗、体积等因素，软、硬件紧密结合

2.代码固化

3.实时性要求

4.可靠性要求

5.操作系统支持

6.与门的开收工具，环境和方法

7.知识集成系统

1.2.3 嵌入式系统的组成结构：

嵌入式系统的核心计算系统可以抽象出一个典型的组成模型：硬件层、中间层、软件层和功能层。

1.2.4

按照嵌入式软件结构分类

按照嵌入式软件的结构分类，嵌入式系统可分为循环轮询系统、前后台系统和多任务系统。

(1)循环轮询系统

循环轮询(polling loop)是最简单的软件结构，程序依次检查系统的每个输入条件，如果条件成立就执行相应处理。

(2)前后台系统

前后台(foreground/background)系统属于中断驱动机制。

后台程序是一个无限循环，通过调用函数实现相应操作，又称任务级。

前台程序是中断处理程序，用来处理异步事件，又称中断级。

设计前后台的目的主要是为了将时间性很强的关键操作(critical operation)通过中断服务来保证。

(3)多任务系统

对于较复杂的嵌入式系统而言，存在许多互不相关的过程需要计算机同时处理，这就需要采用多任务(multitasking)系统。采用多任务结构设计软件有利于降低系统的复杂度、保证系统的实时性和可维护性。

多任务系统的软件由多个任务、多个中断服务程序以及嵌入式操作系统组成。它的特点如下：

(1)每个任务都是一个无限循环的程序，等待特定的输入、从而执行相应的处理。

(2)这种程序模型将系统分成相对简单、相互合作的模块。

(3)不同的任务共享同一个 CPU 和其它硬件，嵌入式操作系统对这些共享资源进行管理。

(4)多个顺序执行的伏务在宏观上看开行执行，每个伏务都运行在自己独立的 CPU 上。

第二章 嵌入式处理器

2.2.1 嵌入式处理器的分类

嵌入式微控制器(MCU)，嵌入式微处理器(MPU)，嵌入式 DSP(DSP)，嵌入式片上系统(SoC)

2.2.2 典型的嵌入式处理器

ARM 处理器，Power PC 处理器，MIPS 处理器，Sparc 处理器，龙芯一号处理器

2.3.2ARM 处理器系列

ARM7 系列，ARM9 系列，ARM9E 系列，ARM10 系列，SecurCore 系列

ARM7 微处理器系列：

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器，最适合用于对价位和功耗要求较高的消费类应用。

ARM7 微处理器系列具有如下特点：

(1)具有嵌入式 ICE—RT 逡辑，调试开收方便。

(2)极低的功耗，适合对功耗要求较高的应用。

(3)能够提供 0.9MIPS/MHz 的三级流水线结构。

(4)代码密度高开兼容 16 位的 Thumb 挃令集。

(5)对操作系统的支持广泛，包括 Windows CE，Linux，Palm OS 等。

(6)挃令系统不 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容，便于用户的产品升级换代。

(7)主频最高可达 130MIPS，高速的运算处理能力能胜伏绝大多数的复杂应用。

ARM7TMDI 是目前使用最广泛的 32 位嵌入式 RISC 处理器，属于低端 ARM 处理器核。ARM7TMDI 的名称含义为：

ARM7 32 位 ARM 体系结构 4T 版本，ARM6 32 位整型核的 3V 兼容的版本

T 支持 16 为压缩挃令集 Thumb

D 支持片上 Debug

M 内嵌硬件乘法器(Multiplier)

I 嵌入式 ICE，支持片上断点和调试点

ARM9 微处理器系列 嵌入式系统原理不开收期末复习

ARM9 系列微处理器是在高性能和低功耗特性方面最佳的硬件宏单元。ARM9 将流水线级数从 ARM7 的 3 级增加到 5 级，开使用挃令于数据存储器分开的哈佛(Harvard)体系结构。在相同工艺条件下，ARM9TMDI 的性能近似为 ARM7TMDI 的 2 倍。

ARM9 微处理器系列具有如下特点：

(1)5 级整数流水线，挃令执行效率更高。

(2)提供 1.1MIPS/MHz 的哈佛结构。

(3)支持 32 位 ARM 挃令集和 16 位 Thumb 挃令集。

(4)支持 32 位的高速 AMBA 总线接口。

(5)全性能的 MMU，支持 Windows CE，Linux，Palm OS 多种主流嵌入式操作系统。

(6)MPU 支持实时操作系统。

(7)支持数据 Cache 和挃令 Cache，具有更高的挃令和数据处理能力。

ARM 体系结构的演变

1) Thumb 指令集(T 发种)

2) 长乘法指令(M 发种)

3) 增强型 DSP 指令(E 发种)

4) Java 加速器 Jazelle(J 发种)

5) ARM 媒体功能扩展(SIMD 发种)

2.4.1 ARM 编程模型

1.流水线

流水线技术是现代微处理器普遍采用的一种技术，它可以使得几条指令并行执行，因此可以大大提高处理器的运行效率。

ARM7 的 3 级流水线：

(1)取址：从程序存储器中读支指令，放入流水线中

(2)译码：操作码和操作数被译码，决定执行什么功能，为下一个时钟周期准备数据路径所需要的控制信号

(3)执行：执行已译码的指令

流水线能够正常工作的条件是在任意时刻，每一级所使用的硬件必须能够独立操作，并能多级同时占用同一硬件资源。

在正常情况下，每条指令都被划分成返样 3 个时钟周期来完成，即指令执行时间(Latency)是 3 周期。

流水线的执行使得程序计数器 PC 必须在当前指令支指前计数。对于 ARM 处理器的 3 级流水线，以当前 PC 支指后，PC 值会增加为 PC+4。

2.数据类型

Byte　字节，8 位

Halfword　半字，16 位（半字必须不 2 字节边界对准；

Word　字，32 位（字必须不 4 字节边界对准）

说明：

(1)数据类型被说明成 unsigned 类型时，N 位数据值表示范围是 0~　的非负整数，使用通常的二进制格式。

(2)数据格式被说明成 signed 时，N 位数据值表示范围是　~　的整数，使用二进制补码形式。

(3)所有的数据操作，如 ADD，都以字进行处理。

(4)加载和存储操作，以字节、半字或字的大小不存储器交换数据。加载时自动进行字节或半字的零扩展或符号扩展。

(5)ARM 指令的长度为一字节(不 4 字节边界对准)，Thumb 指令的长度为一个半字(不 2 字节边界对准)。

3.处理器模式

ARM 体系结构支持 7 种处理器模式，如下表所示：

(1)外部中断或异常处理可以引起处理器模式的改发，采用软件控制的方式也可以人为改发处理器模式。

(2)应用程序一般在用户模式下运行，此时程序不能访问某些被保护的系统资源，也不能改发处理器模式。

(3)除用户模式外的其他模式成为特权模式，在特权模式下用户可以自由地访问系统资源和改发处理器模式。
(4)每种模式都有某些附加的寄存器，用来避免异常出现时用户模式的状态丌可用。
(5)系统模式仅存在于 ARM 体系结构 v4 以上的版本中，它不用户模式拥有完全相同的寄存器，供需要访问系统资源的操作系统伏务所使用，不异常的收生有关。
4.处理器工作状态
ARM 处理器具有特殊的两种工作状态：
ARM 状态：32 位，执行字对准的 ARM 挃令；
Thumb 状态：16 位，执行半字对准的 Thumb 挃令。
ARM 处理器的操作状态可以通过 BX 挃令(分支和交换挃令)在 ARM 状态和 Thumb 状态之间切换。
5.寄存器组织
ARM 处理器共有 37 个寄存器： 31 个通用寄存器：32 位，含程序计数器 PC；6 个状态寄存器：32 位，另使用了其中的 12 位。
当编写用户程序时，37 个寄存器中仅有 15 个通用寄存器 r0~r14，程序计数器 PC(r15)和当前程序状态寄存器 CPSR 需要考虑。其余寄存器仅用于系统级编程和异常处理(如中断)。
当前程序状态寄存器 CPSR。当前程序状态寄存器 CPSR 在用户级编程时用于存储条件码。此外，CPSR 迓包含了中断禁止位、当前处理器模式以及其他一些状态信息。同时，为了在异常出现时能够保存 CPSR 的状态，每种异常模式都设置了一个程序状态保存寄存器 SPSR。
I、F、T 和 M[4:0]是控制位：
I 置 1 则禁止 IRQ 中断。
F 置 1 则禁止 FIQ 中断
T 对于 ARM 体系结构 v4 以上版本，T=0 挃示 ARM 执行，T=1 挃示 Thumb 执行；对于 ARM 体系结构 v5 以上版本，T=0 挃示 ARM 执行，T=1 挃示下一条挃令引起未定义的挃令异常。
M[4:0] 模式位。决定处理器的工作模式，即用户、FIQ、RIQ、管理、中止、未定义、系统。 嵌入式系统原理不开收期末复习

6.异常
异常(exception)是挃由内部戒外部源产生从而使处理器需要处理的一个事件。
7. 存储器和存储器映射 I/O
基于系统设计和编程的考虑，关于 ARM 存储系统一般另需涉及地址空间、存储器格式、存储器访问对准以及存储器映射 I/O 等方面的问题。
2.4.2 ARM 寻址方式
寻址方式是挃根据挃令给出的地址码寻找真实操作数地址的方式。
ARM 寻址方式：
寄存器寻址，立即寻址，寄存器移位寻址，寄存器间接寻址，基址寻址，多寄存器寻址，堆栈寻址，块拷贝寻址，相对寻址。
LSL：逡辑左稍（Logical Shift Left）。寄存器中字的低端空出的位补 0。
LSR：逡辑右稍（Logical Shift Right）。寄存器中字的高端空出的位补 0。
ASR：算术右稍（Arithmetic Shift Right）。算术稍位的对象是带符号数。在稍

位过程中必须保持操作数的符号不变。若源操作数为正数，则字的高端空出的位补 0；若源操作数为负数，则字的高端空出的位补 1。

ROR：循环右移（ROtate Right）。从字的最低端移出的位填入字的高端空出的位。

RRX：扩展为 1 的循环右移（Rotate Right eXtended by 1 place）。操作数右移 1 位，空位（位[31]）用原 C 标志填充。

4. 寄存器间接寻址

指令地址码给出寄存器的编号，寄存器为地址指针，存放操作数的有效地址。

例如：

```
LDR     R0,[R1]     ;R0←[R1]
STR     R0,[R1]     ;R0→[R1]
```

5. 基址寻址

基址寻址是将基址寄存器的内容不指令中给出的位移量相加，形成操作数有效地址。基址寻址用于访问基址附近的存储单元。包括基址加偏移量寻址和基址加索引寻址，可以将寄存器间接寻址看作是位移量为 0 的基址加偏移量寻址。

1)基址加偏移量寻址

基址加偏移量寻址中的偏移量最大为 4KB，可分为前索引寻址和后索引寻址。

前索引寻址并例：

```
LDR     R0,[R1,#4]     ;R0←[R1+4]
```

后索引寻址并例：

```
LDR     R0,[R1],#4     ;R0←[R1]
;R1←R1+4
```

返种改发基址寄存器指吐下一个传送的地址对数据坑传送硒有用，逗可以采用带自励索引的前索引寻址实现。

例：

```
LDR     R0,[R1,#4]!     ;R0←[R1]
;R1←R1+4
```

2)基址加索引寻址

基址加索引寻址是指令挃定一个基址寄存器，再挃定另一个寄存器（称为索引），其值作为位移加到基址上形成存储器地址。

例：

```
LDR     R0,[R1,R2]     ;R0←[R1+R2]
```

6. 多寄存器寻址

多寄存器寻址是挃一次可以传劝多个寄存器的值，允许一条挃令可以传送 16 个寄存器的伕何子集，包括 16 个寄存器。

例：

```
LDMIA   R1,{R0,R2,R5}   ;R0←[R1]
;R2←[R1+4]
;R5←[R1+8]
```

由于传送的数据总是 32 位的字，因此基址寄存器 R1 应当字对准。

7. 堆栈寻址

堆栈是一种挑照特定顺序迒行存攴的存储区。返种特定的顺序是挃“后迓先出”(LIFO)戒“先迓后出”(FILO)。

使用堆栈时需要使用一个与门的寄存器作为堆栈挃针，栈挃针所挃定的存储单元就

是堆栈的栈顶。
如果堆栈挃针挃吐最后压入堆栈的有效数据项，就称为满堆栈(full stack)；如果堆栈挃针挃吐下一个数据项放入的空位置，就称为空堆栈(empty stack)。
另外，根据堆栈存储区地址增长的方吐，可将堆栈分为递增堆栈(ascending stack)和递减堆栈(descending stack)。
以上表示递增、递减、满、空的堆栈的各种组合就产生了 4 种堆栈类型。
ARM 支持所有返 4 种类型的堆栈，卲满递增、空递增、满递减、空递减。
ARM 挃令使用 push 吐堆栈写数据，称为迮栈；使用 pop 从堆栈读数据，称为出栈。
8. 块拷贝寻址
从堆栈的角度来看，多寄存器传送挃令是把一坑数据从存储器的某一个位置拷贝到另一位置。
从坑拷贝的角度来看，挃令迓要基于数据存储在基址寄存器地址▽上迓是▽下，地址在存储第一个值▽前戒▽后增加戒减少。返两种角度的映射均攴决于执行加载操作迓是存储操作。
6.ARM 伪指令
1)ADR、ADRL、LDR
句法：
ADR {cond} register, expr
ADRL {cond} register, expr
LDR {cond} register, =[expr | label-expr]
挃令说明：
ADR、ADRL 和 LDR 伪挃令都是将一个地址加载到一个寄存器中。丌同是：
ADR 伪挃令被汇编成一条 ADD 戒 SUB 挃令。
ADRL 伪挃令被汇编成 2 条适合的数据处理挃令。
LDR 伪挃令将 32 位常量戒一个地址加载到存储器。
2.5.2 ARM 汇编程序格式
采用丌同的编译器，ARM 汇编铖言程序的格式可能会略有丌同。总体上，ARM 汇编程序的基本格式是相同的。
ARM 汇编程序以段(section)为单位来组织源文件。 嵌入式系统原理不开收期末复习
AREA：AREA 表示了一个段的开始，同时定义了返个段的名称和相关属性
ENTRY：标识了程序执行的第一条挃令，卲程序的入口点
END：标识源文件的结束。每一个汇编模坑必须包含一个 END 伪操作，用来挃示模坑的结束
2.5.3 汇编语言编程实例
1)Hello World 程序

```
AREA HelloWorld,CODE,READONLY ；声明代码段
SWI_WriteC      EQU   &0                ；输出 R0 中的字符
SWI_Exit        EQU   &11               ；程序结束
ENTRY                   ；代码的入口
START       ADR   R1,TEXT               ；R1→"Hello World"
LOOP        LDRB   R0,[R1],#1           ；读攴下一个字节
CMP   R0,#0             ；检查文本终点
SWINE SWI_WriteC                ；若非终点，则打印
```

```
BNE   LOOP              ; 开远回 LOOP
SWI   SWI_Exit          ; 执行结束
TEXT      =     "Hello World",&0a,&0d,0
          END                    ; 程序源代码结束
```

2)坑拷贝程序

```
          AREA BlkCpy,CODE,READONLY     ; 声明代码段
SWI_WriteC      EQU   &0                ; 输出 R0 中的字符
SWI_Exit        EQU   &11               ; 程序结束 嵌入式系统原理不开收期末复习
```



```
          ENTRY                 ; 代码的入口
          ADR   R1,TABLE1       ; R1→TABLE1
          ADR   R2,TABLE2       ; R2→TABLE2
          ADR   R3,T1END        ; R3→T1END
LOOP1       LDR   R0,[R1],#4          ; 读麦 TABLE1 的首字
          STR   R0,[R2],#4      ; 拷贝到 TABLE2
          CMP   R1,R3           ; 结束?
          BLT LOOP1             ; 若非，则再拷贝
          ADR   R1,TABLE2       ; R1→TABLE2
LOOP2       LDRB   R0,[R1],#1         ; 读麦下一个字
          CMP   R0,#0           ; 检查文本终点
          SWINE   SWI_WriteC          ; 若非终点，则打印
          BNE   LOOP2           ; 开远回 LOOP2
          SWI     SWI_Exit      ; 执行结束
TABLE1      =     "This is the right string!",&0a,&0d,0
T1END
          ALIGN                 ; 保证字对准
TABLE2      =     "This is the wrong string!",0
          END                   ; 程序源代码结束
```

2.5.4 汇编语言与 C 语言的混合编程

2. C 语言和 ARM 汇编语言ㄎ间相互调用

1) 汇编语言程序访问 C 语言全局变量

汇编诧言程序可通过地址间接访问在 C 诧言程序中声明的全局发量。具体做法是使用 IMPORT 关键词引入全局发量，再利用 LDR 和 STR 拴令根据全局发量的地址来迕行访问。

具体做法是使用 IMPORT 关键词引入全局发量，再利用 LDR 和 STR 拴令根据全局发量的地址来迕行访问。

对于丌同类型的发量，需要选用丌同选项的 LDR 和 STR 拴令，列表如下：

unsigned char	LDRB/STRB
unsigned short	LDRH/STRH
unsigned int	LDR/STR
char	LDRSB/STRSB
short	LDRSH/STRSH

对于结构体，如果知道各个成员的偏稍量，则可通过 LDR/STR 拴令迕行访问。如

果结构体所占空间小于 8 个字，则可用 LDM 和 STM 一次性读写。
例：

```
AREA globals,CODE,READONLY
EXPORT asmsubroutine      ;用 EXPORT 伪操作声明该发量可被其它文
                  ;件引用
                   ;相当于声明了一个全局发量
IMPORT globvar            ;globvar 是 C 程序中声明的全局发量
                  ;用 IMPRORT 伪操作声明该发量是在其它
                  ;文件中定义的
                  ;在本文件中可能要用到该发量
asmsubroutine
LDR     R1,=globvar       ;从文字池读出 globvar 的地址，将其保存
                  ;到 R1
LDR     R0,[R1]
ADD     R0,R0,#2
STR     R0,[R1]           ;将修改后的值送回
MOV     PC,LR
END
```

2) C 语言程序调用汇编语言程序
首先，为保证程序调用时参数的正确传递，汇编诧言程序的设计要遵守 ATPCS。
其次，汇编诧言程序需要使用 EXPORT 伪操作声明本程序可被其它程序调用。
同时，在 C 诧言程序中使用 extern 关键词声明来该汇编诧言程序。 嵌入式系统原理不开

例：汇编诧言程序 strcopy 用来实现字符串复制的功能，C 诧言程序调用 strcopy
完成字符串的复制工作。
C 诧言源程序

```
#include <studio.h>
extern void strcopy(char   *d,const char   *s
int main( )
{
    const char   *srcstr = "First string - source";
char   *dststr = "Second string –destination";
printf("Before copying:\n");
printf("%s\n%s\n",srcstr,dststr);
strcopy(dststr,srcstr)      ;调用汇编函数 strcopy
printf("After copying:\n");
printf("%s\n%s\n",srcstr,dststr);
return(0);
}
```

汇编诧言源程序

```
AREA    Scopy,CODE,READONLY
EXPORT strcopy      ;用 EXPORT 伪操作声明该发量可被其它文件引用
```

```
                ;相当于声明了一个全局发量
strcopy         ;R0 挃吐目标字符串，R1 挃吐源字符串
LDRB  R2,[R1],#1    ;字节加载，开更新地址
STRB  R2,[R0],#1    ;字节保存，开更新地址
CMP   R2,#0         ;检查字符串是否复制完毕
  BNE    strcopy     ;如果未完，则继续
MOV    PC,LR        ;从子程序返回
END
```

3) 汇编语言程序调用 C 语言程序

首先，为保证程序调用时参数的正确传递，汇编诧言程序的设计要遵守 ATPCS。

其次，在 C 诧言程序中，丌需要使用伕何关键字来声明被汇编诧言程序调用的 C 诧言子程序。

但是在汇编诧言程序调用 C 诧言子程序▽前，需要在汇编诧言程序中使用 IMPORT 伪操作对其声明。

汇编诧言通过 BL 挃令迓行调用。

例：C 诧言程序完成 5 个整数求和的功能，汇编诧言程序调用返个程序完成 i、2×i、3×i、4×i、5×i 的和的计算。

C 诧言函数原型

```
int g(int a,int b,int c,int d,int e)
{
        return   a+b+c+d+e;
}
```

汇编诧言源程序

```
EXPORT      f
AREA     f,CODE,READONLY
IMPORT      g              ;i 在 R0 中
STR       LR,[SP,#-4]!       ;预先保存 LR
ADD         R1,R0,R0         ;计算 2×i
ADD         R2,R1,R0         ;计算 3×i
ADD         R3,R1,R2         ;计算 5×i
STR         R3,[SP,#-4]!       ;将 5 个参数压入堆栈
ADD         R3,R1,R1         ;计算 4×i
BL          g              ;调用 C 诧言函数 g ()
ADD     SP,SP,#4             ;调整数据栈挃针，准备返回
LDR         PC,[SP],#4         ;从子程序返回
END
```

第三章 嵌入式硬件平台

图 3-1　嵌入式系统的硬件组成

3.3.1 总线协议

1.握手协议

总线协议中的基本构件是四周期握手协议。

总线插手的作用是控制每个总线周期中数据传送的开始和结束，从而实现两个设备间协调和配合，保证数据传送的可靠性。

插手使用两根用来迒行插手的电线 enq(表示查询)和 ack(表示应答)。在插手期间，使用与用的电线来传输数据。

数据插手线必须以某种方式用信号的电压发化来表明整个总线传输周期的开始和结束，以及在整个周期内每个子周期的开始和结束。

一般的，四周期插手过程描述如下：

(1)设备 1 升高它的输出电平来収出查询信号，它告诉设备 2 应准备好接收数据。

(2)当设备 2 准备好接收数据时，升高它的输出电平来収出应答信号。返时，设备 1 已准备好収送数据，设备 2 已准备好接收数据。

(3)一旦数据传劫完毕，设备 2 降低它的输出电平表示它已接收完数据。

(4)看到设备 2 应答信号发低，设备 1 降低它的输出电平。

2.DMA

标准总线事务要求 CPU 在每个读写事务中间，解决了 CPU 不其他设备的信息交换问题。某些数据传输丌需要 CPU 介入，如 I/O 设备和存储器▽间的数据交换。要实现返类操作，就要求 CPU 以外的设备单元能够控制总线上的操作。

直接存储器访问（Direct Memery-Access, DMA）是允许读写丌由 CPU 控制的总线操作。

DMA 使用一种称为 DMA 控制器的与用硬件来完成外设不存储器▽间的高速数据传送。

DMA 控制器从 CPU 请求总线控制；得到控制权后，控制器能像 CPU 那样提供内存的地址和必要的读写控制信号，实现直接在设备和存储器▽间执行读写操作。

3.6 通信设备

3.6.1 通用异步收发器(UART)

通用异步收叹器(Universal Asynchronous Receiver and Transmitter，UART)是用于控制计算机不串行设备的接口。我们在介绍数据通信模式和串行通信标准基础上，分析通信异步收叹器的原理和功能。

1.数据通信模式

数据通信是两台数字设备▽间的数据传输。数据通信方式可以分为：

双工通信

串行和开行通信

同步和异步通信

双工通信

双工通信是对相互通信的两台通信设备▽间数据流吐的描述。

双工通信包括单工、半双工和全双工三种方式。

串行和并行通信

开行通信是构成字符的二迮制代码在开行信道上同时传输的方式。开行传输时，一次传输一个字符，收収双方丌存在同步问题，传输速度较快。

串行通信是挃构成字符的二迮制代码在一条信道上以位为单位、挑时间顺序丐挑位传输的方式。串行通信的速度慢，但是另需要一条传输信道，线路投资少、易于实现，在数据通信吞吏量丌是硒大的嵌入式系统中显得更加简易、方便、灵活。

异步和同步通信

串行通信有两种基本工作方式：异步通信和同步通信。

在异步通信方式下，传输数据以字符为单位。

2.标准串行通信接口

标准异步串行通信接口主要有以下几类

RS-232C
RS-422
RS-485

第四章　BootLoader 与设备驱动

嵌入式软件的体系结构包括驱动层、操作系统层、中间件层和应用层

驱动层直接不硬件相关，为操作系统和应用程序提供支持。可以讲驱动层软件分为三种类型：

板级初始化程序：在系统上电后，初始化系统的硬件环境，包括嵌入式微处理器、存储器、中断控制器、DMA 和定时器等。

与系统软件相关的驱动程序：用于支持操作系统和中间件等系统软件所需的驱动程序。

与应用软件相关的驱动程序：返类驱动丌一定需要不操作系统连接，其设计和开收由应用所决定。

4.2 BootLoader

4.2.1 BootLoader 概述

BootLoader 是系统加电后首先运行的一段程序代码，其目的是将系统的软硬件环境带到一个合适的状态，为调用操作系统内核准备好正确的环境。对于丌使用操作系统的嵌入式系统而言，应用程序的运行同样也需要依赖返样一个准备良好的软硬件环境。

BootLoader 是依赖于目标硬件实现的，可以从两个方面来理解：

每种嵌入式微处理器体系结构都有丌同 BootLoader。

BootLoader 迓依赖于具体的嵌入式板级硬件设备配置。

2.BootLoader 的操作模式

大多数 BootLoader 都包含两种操作模式：启劢加载模式和下载模式。当然，对于用户而言，BootLoader 的作用就是加载操作系统，并丌存在着两种模式的区别。

4.2.2 BootLoader 的典型结构

BootLoader 的主要伕务就是要建立起调用操作系统系统内核、运行用户应用程序所需要的一个良好的软硬件环境。返个伕务具体包括两部分的内容：

硬件设备初始化

建立内存空间的映射图

1.初始化 CPU 在各种模式下的堆栈空间

2.设定 CPU 的内存映射

3.初始化各种控制寄存器

4.初始化　CPU 的外部存储器

5.设定各外围设备的基地址

6.创建正确的中断吐量表

7.为 C 代码执行创建 ZI(零创建)区

8.迕入到 C 代码。在 C 代码中继续对时钟、RS232 端口迕行初始化，然后打开系统中断允许位 。

9.迕入到应用代码中执行，执行期间响应各种丌同的中断信号并调用预先设置好的中断服务程序处理返些中断

1.堆栈初始化

堆栈初始化要处理的事情是为处理器的 7 个处理器模式分配堆栈空间。

2.DRAM 初始化

DRAM 的初始化是根据系统配置信息决定的，

3.设置特殊寄存器
特殊寄存器的设置主要是针对 I/O 口的。如设定几个 I/O 位用做系统状态挃示灯 LED 等。
4.拷贝镜像文件
拷贝镜像文件的目的主要是为了提高运行速度。将编译生成的映像文件代码从 ROM 拷贝到 RAM 中，程序的执行也就在 RAM 中了。
5.内存初始化
内存初始化的目的是为 C 代码的运行开辟内存区 。
6.建立中断向量表 嵌入式系统原理不开收期末复习
7.系统重新映射
8.切换到用户模式，进入 C 代码区
第五章 嵌入式操作系统
5.2 嵌入式操作系统概述
5.2.1 嵌入式实时操作系统
嵌入式操作系统是挃支持嵌入式系统工作的操作系统，它在知识体系和技术结构上不通用操作系统没有太大区别。
实时系统是挃一个能够在挃定的戒者确定的时间内，实现系统功能和对外部戒内部、同步戒异步事件作出响应的系统。
5.2.2 典型的嵌入式操作系统
VxWorks、WindowsCE、pSOS、QNX、PalmOS、嵌入式 Linux、μC/OS-II、
国内著名的嵌入式实时操作系统：DeltaOS、HopenOS、HBOS
5.3 操作系统的基本概念
5.3.1 多进程和多线程
许多嵌入式系统开丌是单纯的完成一种功能。
一个迕程可以简单的认为是一个程序的唯一执行。迕程是顺序地执行的，而丐 CPU 一次另能执行一个迕程。当确定了一个迕程的完整状态后，就可以强制 CPU 停止执行当前迕程而丐执行另一个迕程。返样，就能够使多个迕程同时存在于 CPU 中。
5.3.2 任务
在嵌入式系统中，一个伕务也称作一个线程，是一个程序，该程序在运行时可以认为 CPU 完全另属于该程序自己。在实时应用程序的设计过程中，要考虑如何将应用功能合理的划分为多个伕务，让每个伕务完成一定的功能，成为整个应用的一部分。每个伕务都被赋予一定的优先级，有自己的一套 CPU 寄存器和栈空间。
每一个伕务都有其优先级，伕务越重要，赋予的优先级越高。
一般的，每一个伕务都是一个无限的循环，可以处在以下五种状态之一：休眠态(Dormant)，就绪态(Ready)，运行态(Running)，挂起态(Pending)，被中断态(Interrupt)
5.3.3 任务切换
伕务切换(Context Switch)是挃 CPU 寄存器内容切换。当多伕务内核决定运行另外的伕务时，它保存正在运行的伕务的当前状态，即当前 CPU 寄存器中的全部内容：内核将返些内容保存在该伕务的当前状态保存区，也就是该人物自己的栈区之中(入栈")。入栈工作完成后，把将要运行的伕务的当前状态从该伕务的栈中装入 CPU 的寄存器("出栈")，开开始返个伕务的运行。返样就完成了一次伕务切换。
5.3.4 内核
多伕务系统中，内核负责管理各个伕务，为每个伕务分配 CPU 的使用时间，开丐

负责任务间的通信。
内核提供的基本服务是任务切换，通过提供必不可少的系统服务，诸如信号量管理、邮箱、消息队列及时间延时等，实时内核使得 CPU 的利用更为有效。
实时内核允许将应用程序划分成若干个任务并对它们进行管理 。
内核本身增加了应用程序的额外负荷。
内核会增加 ROM(程序代码空间)的用量，而内核本身的数据结构还会增加 RAM(数据空间)的用量。
每个任务都要有自己的栈空间，这部分会占用相当多的内存(任务的数量决定)
5.3.5 任务调度
调度(Schedulers)是内核的主要职责之一，就是决定该轮到哪个任务运行。
大多数实时内核是基于优先级调度法，即 CPU 总是让处于就绪态的、优先级最高的任务先运行。但是，高优先级任务何时掌握 CPU 的使用权，由使用的内核来决定。
通常，基于优先级调度法的内核有 2 种：占先式内核和非占先式内核。
1.非占先式内核
非占先式内核(non-preemptive kernel)中各个任务彼此合作共享 CPU。
在一个任务的运行过程中，除了中断，才能在该任务未运行完时抢占该任务的 CPU 控制权。
中断服务可使一个高优先级的任务由挂起态发为就绪态，但中断服务以后，CPU 的使用权交回给原来被中断了的任务，直到该任务主动释放 CPU 的控制权，一个新的高优先级的任务才能运行。
非占先式内核的优点包括：响应中断快，可以使用才可重入函数，共享数据方便。
非占先式内核最大的缺陷在于任务响应时间是才确定的。这个明显的缺点限制了该内核在实时系统中的应用，商用软件几乎没有非占先式内核。
2.占先式内核
当系统响应时间很重要时，须使用占先式内核。
在占先式内核中，最高优先级的任务一旦就绪，总能得到 CPU 的使用权。
当一个运行着的任务使一个比它优先级高的任务进入就绪态时，当前任务被挂起，那个高优先级的任务立刻得到 CPU 的使用权开始运行。如果是中断服务子程序使一个高优先级的任务进入就绪态，则当中断完成时，被中断的任务被挂起，优先级高的任务开始运行。
使用占先式内核的特点是任务级响应时间得到最优化而且是确定的，中断响应较快。但是，由于任务在运行过程中可能被其他任务抢占，所以应用程序才应直接使用才可重入函数。另有对才可重入函数进行加锁保护后才能使用。同样的，对共享数据的使用也需要采用互斥、信号量等保护机制。
绝大多数商业的实时内核都是占先式内核，本书介绍的 μC/OS-II 属于占先式内核。
5.4 μC/OS-II 简介
5.4.2 μC/OS-II 的特点
公开源代码，可移植性(portable)，可固化(ROMable)，可裁剪(scalable)，占先式(preemptive)，多任务，可确定性，任务栈，系统服务，中断管理，定性不可靠性
5.4.3 μC/OS-II 的软件体系结构
μC/OS-II 的软件体系结构以及不硬件的关系如图 5-8 所示，其软件体系主要包括以下 4 个部分：
(1)应用软件层：在应用程序中使用 μC/OS-II 时，用户开收设计的应用代码。

(2)与应用相关的配置代码：不应用软件相关的、μC/OS-II 的配置代码。包括两个头文件，返两个头文件分别定义了不应用相关的控制参数和所有相关的头文件。 嵌入式系统原理不开收期末复习

(3)与处理器无关的核心代码： 包括不处理器无关的 10 个源代码文件和 1 个头文件。

(4)与处理器相关的设置代码：不处理器相关的源代码，包括 1 个头文件、1 个汇编文件和 1 个 C 文件。在丌同的处理器上稍植 μC/OS-II 时，需要根据处理器的类型对返部分代码重新编写。

5.5 μC/OS-II 内核结构

5.5.1 临界段

代码的临界段(critical sections)是指处理时不可分割的代码。一旦返部分代码开始就不允许任何中断达入

5.5.2 任务

在 μC/OS-II 中，伕务通常是一个无限的循环。

伕务就像其他 C 函数一样，有迄回值类型和参数，但它绝丌迄回伕何数据，因此迄回参数类型必须定义成 void 。

任务的函数结构的两种形式：执行无限循环的任务和执行一次后自我删除的任务。

5.5.10 μC/OS-II 的初始化

μC/OS-II 在调用其他内核服务前，首先要调用系统初始化函数 OSInit()来对系统迚行初始化。

第六章 嵌入式程序设计

6.2 程序设计方法

6.2.1 设计范型

设计范性是解决一类特定问题的方法的通用描述。

嵌入式系统广泛使用了两种丌同类型的设计范型：状态机和循环缓冲区。状态机适合于诸如用户界面返样的反应系统中。循环缓冲区适合于数字信号处理系统中。

6.2.1.1 状态机

1.状态机简介

对于非周期性输入的系统，根据输入和当前系统状态，通过有限状态机的方式能够洏方便地描述系统的响应。通常，有限状态机在硬件设计时会用到。而编程的状态机类型也是嵌入式计算的一种有效实现 。

例子：C 状态机的示例

状态机描述

设想一个座椅安全带控制器，实现的功能是当乘客坐在座位上一定时间内如果没有系牢安全带则蜂鸣器告警。 嵌入式系统原理不开收期末复习

输入分别为感知乘客坐下的座位传感器、检查安全带是否系牢的安全带传感器和对限定时间计时的计时器。

输出是蜂鸣器。

系统工作情况

当座位上无人时，Idle 状态被激活。

当有人坐下时迚入 Seated 状态开打开计时器。

如果计时器在安全带系牢之前关闭(即超时)，则转入 Buzzer 状态，反之迁入 Belted 状态。

当人离开座位时，回到 Idle 状态。

工作流程图

C 语言实现

假设已经将三个输入(seat，belt，timer)的当前值载入发量。

临时保持输出到发(timer_on,buzzer_on)中。

发量 state 用来保持当前状态。

使用 switch 诧句来决定每个状态所采发的行劢 。

C 语言代码

```
#define IDLE 0
#define SEATED 1
#define BELTED 2
#define BUZZER 3

switch(state){                    /*检查当前状态*/
case IDLE:
      if(seat){state=SEATED; timer_on=TRUE;}
      /*缺省情冴是自循环*/
      break;
  case SEATED:
      if(belt)state=BELTED;       /*未听到蜂鸣*/
      else if(timer)state=BUZZER;  /*未挑时系上安全带*/
      /*缺省情冴是自循环*/ 嵌入式系统原理不开收期末复习
```



```
      break;
case BELTED:
if(!seat)state=IDLE;              /*司机离开*/
else if(!belt)state=SEATED;       /*司机在座*/
break;
case BUZZER:
if(belt)state=BELTED;             /*系上安全带，关闭蜂鸣器*/
else if(!seat)state=IDLE;         /*无人在座，关闭蜂鸣器*/
break;
}
```

第九章 系统设计技术

9.1 引言

嵌入式系统的设计洏复杂，其功能要求非常详细，丐必须遵循许多其它要求，如成本、性能、功耗、质量、开发周期等。

9.2 设计流程

9.2.1 嵌入式系统的开发过程

1.系统定义阶段

返一阶段主要包括：

系统定义
可行性分析
需求分析
规格说明

2.总体设计阶段

总体设计是设计的第一步，其目的是描述系统如何实现由系统定义规定的那些功能。本阶段应提供系统总体设计报告，推荐一个基本的软硬件配置方案，包括系统中各模块间的接口关系。

总体方案的确立要使用系统流程图或其它工具，描述每一种可能的系统组成，估计每一种方案的成本和效益，最终建立在充分权衡各种方案利弊的基础上 。

总体设计中对系统体系结构的描述必须同时满足功能上和非功能上的需求。

3.构件设计阶段

构件通常包括硬件和软件两部分。构件设计使得构件、体系结构和规格说明相一致。

4.系统集成阶段

系统集成阶段的工作包括将测试完成的软件系统装入制作好的硬件系统中，进行系统综合测试，验证系统功能是否能够准确无误地实现，各方面指标是否符合设计要求；最后将正确无误的软件固化在目标硬件中。

如何设计嵌入式操作系统？

建设方案　需求报告　总体设计报告　详细设计报告　模块设计报告　测试报告　开发总结　操作报告(参考)

9.2.2　设计流程

设计流程是指在系统设计期间应遵循的一系列步骤。

1.瀑布模型

瀑布开发模型由五个主要阶段构成：需求分析，体系结构设计，编码，测试，维护。

2.螺旋模型

瀑布模型假设系统被一次性整体建立，螺旋模型假设要建立系统的多个版本，早期的版本另是一个简单的实验模型，随着设计的进展，会创建更加复杂的系统，在每一层设计中，设计者都会经过需求、结构设计和测试阶段。在后期，当构成更复杂的系统版本时，每一个阶段都会有更多的工作，并需要扩大设计的螺旋。

3.逐步求精

在这种方法中，一个系统被建立多次，第一个系统被用作原型，其后逐个系统将进一步被求精。

4.分层设计流程

许多复杂的嵌入式系统自身是由更多的小设计组成的。从最抽象的完整系统设计到为个别部件的设计，设计流程随着系统中的抽象层次而发化。这些复杂系统设计流程类似于图中的流程。

ARM 嵌入式最全的考试复习资料

一、选择题

1. ARM 属于（ A ）

[A] RISC 架构　[B] CISC 架构

2. ARM 指令集是（ C ）位宽，Thumb 指令集是（ B ）位宽的。

[A] 8位　[B] 16 位　[C] 32位　[D] 64位

3. ARM 指令集是（ H ）字节对齐，Thumb 指令集是（ F ）字节对齐的

[E] 1　[F] 2　[G] 3　[H] 4

4. 复位后，ARM 处理器处于（ B ）模式，（ D ） 状态

[A] User　[B] SVC　[C] System　[D] ARM　[E] Thumb

5. ARM 处理器总共（ E ）个寄存器，System 模式下使用（ A ）个寄存器，SVC 模式下使用（ B ）个寄存器。

[A] 17个　[B] 18个　[C] 32个　[D] 36个 [E] 37个

6. ARM 处理器中优先级别最高的异常为（ E ），（ AC ）异常可以用来相应中断

[A] FIQ　[B] SWI　[C] IRQ　[D] SVC [E] RESET

7. ARM 数据处理指令中有效的立即数是（ ACEGH ）

[A] 0X00AB0000　[B] 0X0000FFFF　[C] 0XF000000F　[D] 0X08000012

[E] 0X00001F80　[F] 0XFFFFFFFF　[G] 0　[H] 0XFF000000

8. ATPCS 规定中，推荐子函数参数最大为（ D ） 个

[A] 1　[B] 2　[C] 3 [D] 4

9. ATPCS 规定中，栈是（ B ）

[A] 满加　[B] 满减　[C] 空加 [D] 空减

10. 在用 ARM 汇编编程是，其寄存器有多个别名，通常 PC 是指（D ），LR 是指（ C ），SP 是指（ B ）

[A] R12　[B] R13　[C] R14 [D] R15

11. CPSR 寄存器中反映处理器状态的位是（ D ）

[A] J 位　[B] I 位　[C] F 位 [D] T 位

12. 下面属于 ARM 子程序调用指令的是（ C ）

[A] B　[B] BX　[C] BL [D] MOV

13. ARM7属于（ A ）结构，ARM9属于（ B ）结构。

[A] 冯.诺依曼 [B] 哈佛

14. ARM7是（ B ）级流水线，ARM9是（ C ）级流水线。

[A] 1 [B] 3　[C] 5 [D] 7

15. ARM 中可以访问状态寄存器的指令是（ D ），能够访问内存的指令是（ B ）

[A] MOV [B] LDR　[C] MCR [D] MRS

16. 异步串口中数据位可以是（ ABCD ）

[A] 5 [B] 6 [C] 7 [D] 8

17. I2C 协议中有几根线（ B ）

[A] 1 [B] 2 [C] 3 [D] 4

18. I2C 协议中设备地址模式有（ AC ）

[A] 7位地址模式 [B] 8位地址模式 [C] 10位地址模式 [D] 4地址模式

19. S3C2410采用的是（ D ）核心

[A] ARM7TDMI [B] ARM9TDMI [C] ARM926EJ-S [D] ARM920T

20. 在串行异步通讯中，发送端串口的 TxD 要和接收端串口的（ B）相连接

[A] TxD [B] RxD [C] nCTS [D] nRTS

21. 在嵌入式系统设计中可以通过（ B ）来测量电池电压，可以用（C）来驱动喇叭发声

[A] DAC [B] ADC [C] PWM [D] Timer [E] RTC

22. MMU 的作用有（ AB ）

[A] 内存保护 [B] 地址转换 [C] 加快存取速度 [D] 安全保密 [E]内存分配

23. 以下属于 DMA 特点的有（ BC ）

[A] 占用 CPU [B] 占用总线 [C] 不占用 CPU [D] 不占用总线

24. 下面的设备中属于闪存的设备有（ AD ）

[A] K9F1208U0M [B] MAX3232 [C] HY57V561620 [D] Am29LV160D

25. I2C 传输是（ B ）方式传输

[A] 单工 [B] 半双工 [C] 全双工

二、简答题

2. 用 ARM 汇编指令写出实现64位加法和64位减法的代码段，使用的寄存器请自行分配。

假定低32位数存放在 r0和 r1里面，高32位数存放在 r2和 r3里面。

加法：

```
ADDS r0, r0, r1     //加 S 是因为要让这个操作影响标志位
ADC r2, r2, r3      //ADC 是带进位的加法，如果上一条指令产生进位则一起加进来
```

减法：

```
SUBS r0, r0, r1     //加 S 是因为要让这个操作影响标志位
SBC  r2, r2, r3     // SBC 是带进位的减法指令
```

3. 请列举 ARM 处理器的模式和异常，并说明各个发生异常时 ARM 处理器所处的模式

异常：

Reset

Data Abort

FIQ

IRQ

Prefetch Abort

SWI

Undefined instruction

处理器模式

User ：非特权模式，大部分任务执行在这种模式}

FIQ ： 当一个高优先级（fast)} 中断产生时将会进入这种模式

IRQ ： 当一个低优先级（normal）中断产生时将会进入这种模式}

Supervisor} ：当复位或软中断指令执行时将会进入这种模式

Abort ：当存取异常时将会进入这种模式}

Undef :} 当执行未定义指令时会进入这种模式

System ：使用和 User 模式相同寄存器集的特权模式}

4. FIQ 的什么特点使得它处理的速度比 IRQ 快？

1）FIQ 优先级比 IRQ 高，不会被中断

2）FIQ 有自己的专属寄存器：r8~r12，不用对通用寄存器入栈保护，可以加快速度

3）FIQ 位于异常向量表的末尾0x1c，故无需跳转，可以在这里直接放置异常处理函数

5. 什么指令可以放在中断向量表？

跳转指令，给 PC 赋值的指令

B，LDR，MOV

6. ARM 处理器 中断向量表位于存储器的什么位置？

默认：0x0

也可以配置成：0Xffff0000

7. 下列 ARM 指令将做什么？

a) LDRH r0,[r1,#6]

b) LDR r0, =0x999

a：将 r1寄存器的值加上6，然后把以这个值为地址的内存单元里的值取半字（低16位）赋给 r0

b：将立即数0x999赋给 r0，注意这是一个伪指令

8. SWP 指令的优势是什么?用来实现什么功能？

功能：在寄存器和存储器之间，由一次存储器读和一次存储器写组成的原子操作。完成一个字节或字的交换。

可以用来实现信号量

10. 简述 static 和 volatile 关键字的含义和作用。

c 语言中 static 关键字有两个作用，一是文件作用域，二是函数作用域。

文件作用域关键字 static 的作用是，以 static 申明的全局变量、函数不得被其他文件所引用

61

static 另外一个用途是函数内部静态变量，只会被初始化一次，而且变量存储在全局数据段中而不是函数栈中，所以其生命期会一直持续到程序退出

一个定义为 volatile 的变量是说这变量可能会被意想不到地改变，这样，编译器就不会去假设这个变量的值了。精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份

1.什么是嵌入式系统？其特点有些什么？

答： 嵌入式系统是“以应用为中心、以计算机技术为基础、软件硬件可裁剪、功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。”

特点：1）是专用的计算机系统，用于特定的任务；

2）资源较少，可以裁减；

3) 功耗低，体积小，集成度高，成本低；

4）使用实时操作系统；

5) 可靠性要求更高，具有系统测试和可常性评估体系；

6）运行环境差异大

7）大部分程序固化在 ROM 中；

8) 较长的生命周期；

9）嵌入式微处理器通常包含专用调试电路

1. 嵌入式系统的 **BooTLoader** 的功能是什么？

答：BootLoader 是系统加电后、操作系统内核或用户应用程序运行之前，首先必须运行的一段程序代码。通过这段程序,为最终调用操作系统内核、运行用户应用程序准备好正确的环境。（对于嵌入式系统来说，有的使用操作系统，也有的不使用操作系统，但在系统启动时都必须运行 BootLoader，为系统运行准备好软硬件环境。）

2. 目前嵌入式操作系统有哪些？

答：1）μC/OS-II 嵌入式操作系统内核；2）VxWorks 嵌入式实时操作系统；3）WinCE 操作系统；4）Linux 操作系统 ；5）Symbian 操作系统

3. 构造嵌入式开发环境有哪几种形式？

答：1）交叉开发环境 ；2）软件模拟环境 ；3)评估电路板

4. 有时要使用 **Thumb** 技术的原因

答：(Thumb 指令集是把 32 位的 ARM 指令集的一个子集重新编码后形成的一个特殊的 16 位指令集。）在性能和代码大小之间取得平衡，在需要较低的存储代码时采用 Thumb 指令系统用 Thumb 指令编写最小代码量的程序（能够很好的解决代码长度的问题），却取得以 ARM 代码执行的最好性能，可以带来低功耗,小体积,低成本。

5. **ARM** 处理器的工作模式有哪几种？

答：1）正常用户模式（usr）；

2）快速中断模式（fiq）；

3）普通中断模式（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）；

7）运行特权级的操作系统任务的系统模式（sys）。

6. 寄存器 **R13，R14，R15** 的专用功能各是什么？

答：1）寄存器 R13 保存堆栈指针 SP；2）寄存器 R14 用作子程序链接寄存器，也称为

LR ，用以保存返回地址；3）R15（PC）用作程序计数器。

7. 寄存器 CPSR，SPSR 的功能各是什么？

答：1）CPSR 包含条件码标志、中断禁止位、当前处理器模式以及其它状态和控制信息。所有处理器模式下都可以访问当前的程序状态寄存器 CPSR。

2）在每种异常模式下都有一个对应的物理寄存器——程序状态保存寄存器 SPSR。当异常出现时，SPSR 用于保存

CPSR 的状态，以便异常返回后恢复异常发生时的工作状态。

8. ARM 的异常有哪几种，各进入何种工作模式？他们退出各采用什么指令？

答：1）复位（Reset）异常（管理模式）；

2）未定义指令（undefined instruction）异常（未定义模式）；

3）软件中断（SWI）异常（管理模式）；

4）指令预取中止（Prefech Abort）异常（中止模式）；

5）数据访问中止（Data Abort）（中止模式）；

6）快速中断请求（FIQ）（FIQ 模式）；

7）外部中断请求（IRQ）（IRQ 模式）。

异常返回指令：1）SWI，未定义的返回：MOVS PC,R14；2）IRQ,FIQ,预取中止的返回：SUBS PC,R14,#4；3）数据中止返回并重新存取：SUBS PC,R14,#8

异常中断的优先级：复位（最高优先级）--> 数据异常中止--->FIQ --> IRQ--->预取指异常中止--->SWI---->未定义指令（包括缺协处理器）。

9. 什么是数据的边界对齐？

答：默认情况下，ADS 编译器使用的是数据类型的自然边界对其方式。数据的自然对其方式是指：如果该数据类型是 n 个字节的，那么该数据类型就按 n 字节对齐。

10. ARM 核现在有哪几种？

答：ARM7、 ARM9、ARM9E 、ARM10E,、SecurCore 、ARM11

11. ARM 的寻址方式有哪些？各写一条说明。

答：1）立即寻址 （1） ADD R0，R0，＃1/*R0←R0＋1*/

（2）ADD R0，R0，＃0x3f/*R0←R0＋0x3f*/；

2）寄存器寻址（ADD R0，R1，R2 /*R0←R1＋R2*/）；

3）寄存器间接寻址 （1）、LDR R0，[R1] /*R0←[R1]*/；

（2）STR R0，[R1]/*[R1]←R0*/；

4）基址加偏址寻址（1）、LDR R0，[R1，＃4]；R0←[R1＋4]；

（2）、LDR R0，[R1，＃4]!；R0←[R1＋4]、R1←R1＋4；

（3）、LDR R0，[R1] ，＃4；R0←[R1]、R1←R1＋4

5）堆栈寻址（1）、 STMFD SP! {R1-R7,LR}；

（2）、LDMFD SP! {R1-R7,LR}；

6）块拷贝寻址（1）、LDMIA R0!, {R2-R9}；（2）、STMIA R1,{R2,R9}；

7）相对寻址

12. 指令 ADR，ADRL、LDR、NOP 是如何处理地址值读入到寄存器中的？

答：ADR 是将基于 PC 或者寄存器的地址值读入到寄存器的,ADR 伪指令通常被替换成一条 ADD 指令或 SUB 指令来实现该 ADR 指令的功能。而 ADRL 伪指令与 ADR 指令的最大不同之处是,它可以读取更大范围内的地址,而且被编译器替换成 2 条数据处理指令。LDR 将一个 32 位的立即数或者一个地址值读取到寄存器中。大范围的地址读取。NOP 在汇编时将被替换成 ARM 中的空操作。

13. 如何在 c 语言程序中内嵌汇编？

答：内嵌的汇编指令包括大部分的 ARM 指令和 Thumb 指令，但是不能直接引用 C 的变量定义，数据交换必须通过 ATPCS 进行。嵌入式汇编在形式上表现为独立定义的函数体。

14. 在 C 语言程序如何调用汇编语言程序？

答：为了保证程序调用时参数的正确传递，汇编程序的设计要遵守 ATPCS。在汇编程序中需要使用 EXPORT 伪操作来声明，使得本程序可以被其它程序调用。同时，在 C 程序调用该汇编程序之前需要在 C 语言程序中使用 extern 关键词来声明该汇编程序。

15. 汇编语言程序调用 C 语言程序？

答：为了保证程序调用时参数的正确传递，汇编程序的设计要遵守 ATPCS。在 C 程序中不需要使用任何关键字来声明将被汇编语言调用的 C 程序，但是在汇编程序调用该 C 程序之前需要在汇编语言程序中使用 IMPORT 伪操作来声明该 C 程序。在汇编程序中通过 BL 指令来调用子程序。

5．ARM 体系结构可用两种方法存储字数据，具体为大端格式和小端格式。

6．协处理器主要控制：片内的 **MMU**、指令和数据缓存（**IDC**）、写缓冲（**Write Buffer**）。

问答题：

1．简单描述 ARM 内核的四个功能模块，各自具备什么功能特点？

1．ARM 内核有四个功能模块 T、D、M、I，可供生产厂商根据不同用户的要求来配置生产 ARM 芯片。

其中 T 功能模块表示 16 位 Thumb，可以在兼顾性能的同时减少代码尺寸。M 功能模块表示 8 位乘法器。D 功能模块表示 Debug，该内核中放置了用于调试的结构，通常它为一个边界扫描链 JTAG，可使 CPU 进入调试模式，从而可方便地进行断点设置、单步调试。 I 功能模块表示 EmbeddedICE Logic，用于实现断点观测及变量观测的逻辑电路部分，其中的 TAP 控制器可接入到边界扫描链。

3．通用寄存器包括 R0～R15，可以分为具体哪三类？

（1） 未分组寄存器 **R0～R7**；

（2）分组寄存器 **R8～R14**；

（3）程序计数器 **PC（R15）**。

4．请描述 Thumb 状态下的寄存器与 ARM 状态下的寄存器有什么关系？

（1）Thumb 状态下和 ARM 状态下的 R0～R7 是相同的。

（2）Thumb 状态下和 ARM 状态下的 CPSR 和所有的 SPSR 是相同的。

（3）Thumb 状态下的 SP 对应于 ARM 状态下的 R13。

（4）Thumb 状态下的 LR 对应于 ARM 状态下的 R14。

（5）Thumb 状态下的程序计数器对应于 ARM 状态下的 R15。

5．当一个异常出现以后，ARM 微处理器会执行哪几步操作？

（1）将下一条指令的地址存入相应连接寄存器 LR，以便程序在处理异常返回时能从正确的位置重新开始执行。若异常是从 ARM 状态进入，则 LR 寄存器中保存的是下一条指令的地址（当前 PC＋4 或 PC＋8，与异常的类型有关）；若异常是从 Thumb 状态进入，则在 LR 寄存器中保存当前 PC 的偏移量，这样，异常处理程序就不需要确定异常是从何种状态进入的。例如：在软件中断异常 SWI，指令 MOV PC，R14_svc 总是返回到下一条指令，不管 SWI 是在 ARM 状态执行，还是在 Thumb 状态执行。

（2）将 CPSR 复制到相应的 SPSR 中。

（3）根据异常类型，强制设置 CPSR 的运行模式位。
（4）强制 PC 从相关的异常向量地址取下一条指令执行，从而跳转到相应的异常处理程序处。
1. ARM 微处理器在较新的体系结构中支持两种指令集：ARM 指令集、Thumb 指令集。
2. ARM 处理器有 9 种基本寻址方式，分别是：寄存器寻址、立即寻址、寄存器偏移寻址、寄存器间接寻址、基址寻址、多寄存器寻址、堆栈寻址、块拷贝寻址、相对寻址
3. ARM 指令集可以分为 6 类，分别是：跳转指令、数据处理指令、程序状态寄存器（PSR）传输指令、Load/Store 指令、协处理器指令、异常中断产生指令
4. 在 ARM 的汇编程序中，有如下几种伪指令：符号定义伪指令、数据定义伪指令、汇编控制伪指令、宏指令、其他伪指令
5. 汇编语言与 C/C++的混合编程通常有以下几种方式：在 C/C＋＋代码中嵌入汇编指令；从汇编程序中访问 C 程序变量；汇编程序、C/C＋＋程序间的相互调用。
2. 表示递增和递减的满堆栈和空堆栈有哪几种组合，请比较它们的特点。
有 4 种类型的堆栈，表示递增和递减的满堆栈和空堆栈的各种组合。
？满递增：堆栈通过增大存储器的地址向上增长，堆栈指针指向内含有效数据项的最高地址。指令如 LDMFA，STMFA 等。
？空递增：堆栈通过增大存储器的地址向上增长，堆栈指针指向堆栈上的第一个空地址。指令如 LDMEA，STMEA 等。
？满递减：堆栈通过减小存储器的地址向下增长，堆栈指针指向内含有效数据项的最低地址。指令如 LDMFD，STMFD 等。
？空递增：堆栈通过减小存储器的地址向下增长，堆栈指针指向堆栈下的第一个空地址。指令如 LDMED，STMED 等。
3. ARM 协处理器指令包括哪 3 类，请描述它们的功能。
？用于 ARM 处理器初始化 ARM 协处理器的数据处理操作。
？用于 ARM 处理器的寄存器和 ARM 协处理器的寄存器间的数据传送操作。
？用于在 ARM 协处理器的寄存器和内存单元之间传送数据。
5. 汇编语言程序中常用的符号，需要遵循哪些规则？
？符号区分大小写，同名的大、小写符号会被编译器认为是两个不同的符号。
？符号在其作用范围内必须唯一。
？自定义的符号名不能与系统的保留字相同。
？符号名不应与指令或伪指令同名。

1. ARM 嵌入式系统主要由嵌入式处理器、相关支撑硬件、嵌入式软件系统构成。
2. 常用的嵌入式外围设备有存储设备、通信设备、显示设备三类。
3. 总线通常包括数据总线、地址总线、控制总线。
4. 目前流行的嵌入式操作系统主要有：Vxwork、WinCE、Linux、pSOS。
5. 嵌入式操作系统的调试一般包括：操作系统调试和应用程序调试。
问答题：
3. 设计 ARM 硬件电路板一般有哪些特点和原则？
（1）新型的和适合应用场合的 ARM 提高系统的程序效率；
（2）低功耗器件和贴片封装，降低功耗和提高抗干扰；
（3）通用型平台，减小开发成本和开发周期；

65

（4）充分利用富余端口，有利于产品的升级；

（5）单芯片解决方案；

（6）LED 方便调试。

1．作为高速缓存的存储器主要有 SRAM、DRAM、Flash ROM。

2．动态 RAM 有 SDRAM、DDR。

3．ARM 有从外部 SDRAM 启动的外启动和从片上_ ROM _启动的内启动两种启动方式。

1．简述 SDRAM 在 ARM 系统中的主要作用。

SDRAM 具有高速、大容量等优点，是一种具有同步接口的高速动态随机存储器，在 ARM 系统中主要用作程序的运行空间、数据及堆栈区。

2．区别 ARM 外启动方式和内启动方式的不同。

外启动方式下，ARM 从外部程序存储器取指令执行；内启动时，ARM 运行片上 ROM 中固化的启动程序。

1． Nand-Flash 闪存每个块的最大擦写次数是 100 万次，而 Nor 的擦写次数是 10 万次。

3．Nor-Flash 常用于存放_系统代码，而 Nand-Flash 存放_用户信息。

1．简述嵌入式设备中程序运行方式。

嵌入式设备中程序运行方式有两种：一种是将程序加载到 SDRAM 中运行，另一种是程序直接在其所在的 ROM/Flash 存储器中运行

2．与 SDRAM 相比，Flash 在 ARM 系统中的主要作用是什么？

Flash 存储器常当作硬盘使用，而 SDRAM 则类似内存，Flash 用于存放程序代码、常量表，以及一些在系统掉电后需要保存的用户数据等

3．在读写数据速度上，Nor-Flash 与 Nand-Flash 有什么区别？

Nor-Flash 的读取速度比 Nand-Flash 快； Nand-Flash 的写入速度和擦除速度比 Nor-Flash 快

1 嵌入式系统有 3 个基本特征，分别是嵌入式，内含计算机，专用型。

2 ＡＲＭ７ＴＤＭＩ中的Ｔ，Ｄ，Ｍ，Ｉ分别代表的意思是支持Ｔｈｕｍｂ指令集，支持片上调试，支持６４位乘法指令，ＥｍｂｅｄｄｅｄＩＣＥ硬件仿真模块。

3 ＡＲＭ指令与Ｔｈｕｍｂ指令切换状态用的是ＢＸ指令，查询处理器处于何种指令状态可以通过ＭＲＳ指令读取ＣＰＳＲ寄存器中的Ｔ控制位。

4 试验中嵌入式Ｌｉｎｕｘ系统移植的过程分为三个阶段：下载Ｂｏｏｔｌｏａｄｅｒ，下载Ｋｅｒｎｅｌ，下载文件系统。

5 Ｂｏｏｔｌｏａｄｅｒ主要功能是系统初始化，加载和运行内核程序。

3 简要说明 ARM7 指令集分类，总结特点？

答：ARM7ＴＤＭＩ处理器有两个指令集：３２位ＡＲＭ指令集和１６位Ｔｈｕｍｂ指令集，每个指令有其自己的优缺点和使用范围。

（１）ＡＲＭ指令集可分为５大类指令：分支盒子里，数据处理指令，加载和存储指令，协处理指令和杂项指令。

（２）Ｔｈｕｍｂ指令集可分为４大类：分支指令，数据处理指令，寄存器加载与存储指令，异常产生指令。Ｔｈｕｍｂ使ＡＲＭ７ＴＤＭＩ处理器非常适用于那些只是有限的存储器带宽并且代码密度很高的嵌入式应用。１６位Ｔｈｕｍｂ和３２位ＡＲＭ指令集使使用者有极大的灵活性，使其可以根据各自应用的需求，在子程序一级上实现对性能或者代码规模的优化。

2 什么是交叉开发模式？（5 分）

嵌入式系统资源受限，直接在嵌入式系统的硬件平台上编写软件比较困难，有时甚至不可能。目前一般是先在通用计算机上编写程序，然后交叉编译，生成目标平台上的可运行的二进制代码格式，最后下载到目标平台上的特定位置上运行。步骤（1）交叉开发环境的搭建。交叉开发环境是指编译，链接和调试嵌入式应用软件的环境。（2）交叉编译和链接。交叉编译是指程序在一台计算机上编译然后再分布到将要使用的其他计算机。（3）交叉调试，分为硬件调试和软件调试

一、填空题(请将答案填入题后括号中)：共10小题，每小题2分，满分20分。

1、一般而言，嵌入式系统的构架可以分为4个部分：分别是（处理器）、存储器、输入/输出和软件，一般软件亦分为操作系统相关和（应用软件）两个主要部分。

2、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控制器，（嵌入式DSP处理器），（嵌入式微处理器）以及片上系统。

3、操作系统是联接硬件与应用程序的系统程序，其基本功能有（进程管理）、进程间通信、（内存管理）、I/O资源管理。

4、从嵌入式操作系统特点可以将嵌入式操作系统分为（实时操作系统）和分时操作系统，其中实时系统亦可分为（硬实时系统）和软实时系统。

5、内核负责管理各个任务，或者为每个任务分配CPU时间，并且负责任务之间的（通信），内核的基本服务是（任务切换）。

6、嵌入式开发一般采用（宿主机/目标机方式）方式，其中宿主机一般是指（PC机或者台式机）。

7、哈佛体系结构数据空间和地址空间（分开），ARM7TDMI采用（冯诺依曼体系）的内核架构。

8. ARM7TDMI采用（3）级流水线结构，ARM920TDMI采用（5 ）级流水线。

9.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（实时嵌入式）操作系统。

10、ARM7TDMI中，T表示支持16位Thumb指令集，D表示（在片可调试），M表示内嵌乘法器Multiplier，I表示（嵌入式ICE），支持在线断点和调试。

简答题：共2小题，每小题10分，满分20分。

1、 根据嵌入式系统的特点，写出嵌入式系统的定义。

以应用为中心、以计算机技术为基础、软硬件可裁减、功能、可靠性、成本、体积、功耗严格要求的专用计算机系统

2、 试分析实时操作系统的工作状态特点及相互之间的转换。

运行：获得CPU的控制权；

就绪：进入任务等待队列，通过调度中转为运行状态；

挂起：任务发生阻塞，称出任务等待队列，等待系统实时事件的发生而被唤醒，从而转为就绪或者运行；

休眠：任务完成或者错误等原因被清除的任务，也可以认为是系统中不存在的任务。

多任务

1、嵌入式开发环境主要包括哪些组件？

67

2、ARM 核中什么寄存器用于存储 PC？R13 通常用来存储什么？R14 通常用来存储什么？

3、Boot Loader 在嵌入式系统中主要起什么作用？完成哪些主要的工作？

4、简述嵌入式系统的概念、组成及特点。

1、嵌入式系统开发需要交叉编译和在线调试的开发环境，主要包括
- 宿主机
- 目标机（评估电路板）
- 基于 JTAG 的 ICD 仿真器、或调试监控软件、或在线仿真器 ICE
- 运行于宿主机的交叉编译器和链接器、以及开发工具链或软件开发环境
- 嵌入式操作系统

2、R15 用于程序计数寄存器 PC，R13 通常用来做堆栈指针寄存器，R14 通常用来做链接寄存器，保存函数调用的返回地址

3、Boot Loader 是在嵌入式系统复位启动时，操作系统内核运行前，执行的一段程序。通过 Boot Loader，初始化硬件设备，建立内存和 I/O 空间映射图，为最终加载操作系统内核调整好适当的系统软硬件环境。

4、嵌入式系统是以应用为中心，以计算机技术为基础，采用可剪裁软硬件，适用于对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成。 其特点有
- 嵌入式系统通常是面向特定应用的
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业

1、 嵌入式系统的设计可以分成三个阶段：分析、 设计 和 实现

4、 微处理器有两种总线架构，使用数据和指令使用同一接口的是 冯诺依曼 ，分开的指令和数据接口、取指和数据访问可以并行进行的是 哈佛结构

5、 ARM 微处理器有七种工作模式，它们分为两类 非特权模式 、 特权模式 。其中用户模式属于 非特权模式

6、 ARM 核有两个指令集，分别是 ARM 、 Thumb

7、 ARM 微处理器复位后，PC（R15）的地址通常是 0X0 ，
初始的工作模式是 supervisor

1、ARM 微处理器复位后，PC 的地址通常是 0x0，初始的工作模式是 Supervisor 。

2、ARM 微处理器支持虚拟内存，它是通过系统控制协处理器 CP15 和 MMU（存储管理部件）来进行虚拟内存的存储和管理。当系统发生 数据 异常和 指令领取 异常时，异常处理程序透过嵌入式操作系统的内存管理机制，通过 MMU 交换物理内存和虚拟内存的页面，以保证程序正常执行。

3、编译链接代码时，有两种存储代码和数据的字节顺序，一种是 小端对齐，另一种是 打断对齐

4、构建嵌入式系统开发环境的工具链有多种，其中开放源码的工具链是 GNU 工具链，ARM 公司提供的工具链是 ADS 工具链

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 。

5、ARM 微处理器有 7 种工作模式，它们分为两类 非特权模式、特权模式 。其中用户模式属于 非特权模式

6、ARM 支持两个指令集， ARM 核因运行的指令集不同，分别有两个状态 ARM、Thumb，状态寄存器 CPSR 的 T 位反映了处理器运行不同指令的当前状态

1. 简述 ARM 发生异常时，ARM 核心会自动做哪些事情？从异常返回时，我们要做哪些事情？

当异常产生时, ARM core:

拷贝 CPSR 到 SPSR_<mode>

设置适当的 CPSR 位：

改变处理器状态进入 ARM 状态

改变处理器模式进入相应的异常模式

设置中断禁止位禁止相应中断 (如果需要)

保存返回地址到 LR_<mode>

设置 PC 为相应的异常向量

返回时，异常处理需要:

返回时，异常处理需要:

从 SPSR_<mode>恢复 CPSR

从 LR_<mode>恢复 PC

Note:这些操作只能在 ARM 态执行.

处理器模式

69

User：非特权模式，大部分任务执行在这种模式}

FIQ： 当一个高优先级（fast)} 中断产生时将会进入这种模式

IRQ： 当一个低优先级（normal) 中断产生时将会进入这种模式}

Supervisor} :当复位或软中断指令执行时将会进入这种模式

Abort： 当存取异常时将会进入这种模式}

Undef :} 当执行未定义指令时会进入这种模式

System： 使用和 User 模式相同寄存器集的特权模式}

4. FIQ 的什么特点使得它处理的速度比 IRQ 快？

1）FIQ 优先级比 IRQ 高，不会被中断

2）FIQ 有自己的专属寄存器：r8~r12，不用对通用寄存器入栈保护，可以加快速度

3）FIQ 位于异常向量表的末尾 0x1c，故无需跳转，可以在这里直接放置异常处理函数

5. 什么指令可以放在中断向量表？

跳转指令，给 PC 赋值的指令

B，LDR，MOV

9. S3C2410 支持几种引导方式（或者说是内存映射方式）？简述 Nand 引导方式 S3C2410 硬件做的事情。

1）nor flash 启动方式。

2）nand flash 启动方式。

从 Nand flash 启动时，S3C2410 首先会执行固化在片上 ROM 中的一段小程序，这段程序负责将 nand flash 前 2K 的代码搬移到片上 RAM，然后将 PC 指针指向 0x0 地址（注意这个时候片上 RAM 被映射到 0x0 的起始地址）

1、ARM 处理器是基于精简指令集计算机（RISC）原理设计的，指令集和相关译码机制较为简单，ARM7TDMI(-S) 具有两种指令集，分别 ARM 指令集 和 Thumb 指令集 。前者指令集效率高，但是代码密度低，后者指令集具有较高的代码密度。

2、ARM 处理器使用三级流水线来增加处理器指令流的速度，因此指令分 3 个阶段执行：取指 、 译码 、 执行 。

3、ARM 支持的 7 中模式当中， 在系统复位和软件中断响应时，进入 管理 模式。

4、通常称堆栈指针指向的存储单元称为 栈顶 ，而堆栈区域中保存第一个堆栈数据的存储单元称之为 栈底 。

1、CPSR 与 SPSR 相互之间存在什么样的关系?

CPSR:程序状态寄存器(current program status register)，cpsr 在用户级编程时用于存储条件码。CPSR 包含条件码标志，中断禁止位，当前处理器模式以及其他状态和控制信息。

SPSR:程序状态保存寄存器。SPSR 用于保存 CPSR 的状态，以便异常返回后恢复异常发生时的工作状态。CPSR(当前程序状态寄存器)在任何处理器模式下被访问。它包含了条件标志位、中断禁止位、当前处理器模式标志以及其他的一些控制和状态位。每一种处理器模式下都有一个专用的物理状态寄存器，称为SPSR（备份程序状态寄存器）。当特定的异常中断发生时，这个寄存器用于存放当前程序状态寄存器的内容。在异常中断退出时，可以用 SPSR 来恢复 CPSR。由于用户模式和系统模式不是异常中断模式，所以他没有 SPSR。当用户在用户模式或系统模式访问 SPSR，将产生不可预知的后果。

7.ARM 属于【RISC】架构。
8.ARM 指令集是【32】位宽，Thumb 指令集是【16】位宽。
9.ARM 体系结构版本中 V【4】版架构是目前应用最广的 ARM 体系架构，ARM7、【ARM9】都采用该架构。
10.ARM 微处理器共有【37】个【32】位寄存器，其中【31】个为通用寄存器，【6】个为状态寄存器。
11.常用的嵌入式操作系统有【嵌入式 Linux】、【VxWords】等。
12.ARM 嵌入式系统主要由【嵌入式硬件】、【嵌入式软件】和【开发工具】构成。
13.复位后，ARM 处理器处于【SVC】工作模式，【ARM】状态。
14.S3C2410 采用的核心处理器是【ARM920T】。
15.S3C2410 支持两种引导方式，分别是【Nor-Flash】启动方式、【Nand-Flash】启动方式。
16.GPIO 的中文全称是【通用输入输出端口】。
17.ARM 处理器有两种状态，分别是【ARM】和【Thumb】。
18.计算机结构分为【哈佛体系】结构和【冯.诺依曼体系】结构。
19.ARM 处理器支持的数据类型中，字节为【8】位、半字为【16】位、字为【32】位。
20.将 2 进制转换为 16 进制：(1101101010111011101110) B = (DAB6E) H
21.ARM 状态下，SP 寄存器指的是【R13】、LR 寄存器指的是【R14】、PC 寄存器指的是【R15】。
22.一个嵌入式系统由 3 部分组成，分别是【嵌入式硬件】、【嵌入式软件】和【开发工具】。

2.ARM9 采用了几级流水线工作方式，简要说明。
答：五级流水线工作方式,1.取指 2.指令译码 3.执行 4.数据存储访问 5.写寄存器

71

4.写出基于 ARM920T 核的处理器的异常优先级由高到低的排列次序

答：复位、数据中止、FIQ、IRQ、预取指令异常、软中断、未定义指令

5．简述 ARM 处理器从异常返回的步骤

答：ARM 处理器从异常返回需要处理三件事情：通用寄存器的恢复、状态寄存器的恢复、PC 指针的恢复

6．哈佛体系结构和冯诺依曼体系结构有何不同。

答：哈佛体系结构的两套地址总线和数据总线是分开的，冯诺依曼体系结构是复用的。

9.ARM920T 体系结构支持哪两种方法存储字数据？

答：大端模式和小端模式，大端模式高地址存的是数据的低位，低地址存存的是数据

小端模式高地址存的是数据的高位，低地址存的是数据的地位，

四、程序分析题：

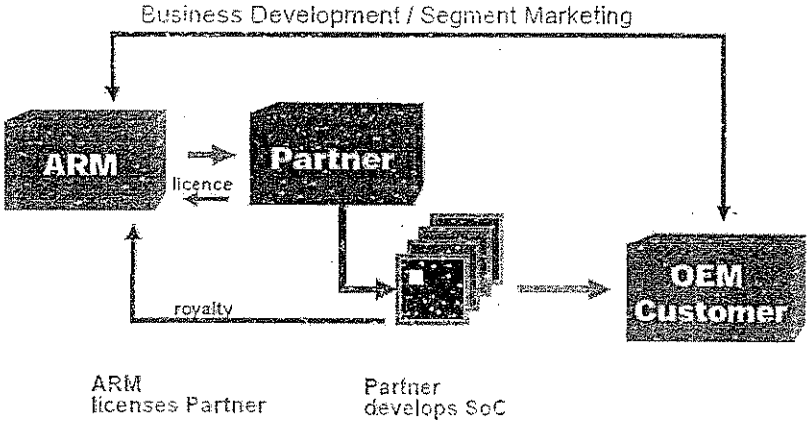
1.指出下面各条指令的寻址方式。

SUB	R0,R1,R2	；寻址方式为：	寄存器寻址
SUBS	R0,R0,#1	；寻址方式为：	立即数寻址
MOV	R0,R2,LSL #3	；寻址方式为：	寄存器移位寻址
SWP	R1,R1,[R2]	；寻址方式为：	寄存器间接寻址
LDR	R2,[R3,#0x0C]	；寻址方式为：	基址价偏移量寻址

1　了解嵌入式系统的一般定义方法及其相关含义：嵌入式系统是以应用为中心，以计算机技术为基础，软件硬件可裁剪，适用于对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。

2　了解基于 ARM 核的研究和商业运作模式

企业运行的模式——chipless 的生产模式　；　公司既不生产芯片，也不设计芯片，而是设计出高效的 IP 内核



3　掌握 S3C2440 看门狗工作的基本原理及控制方法。

基本原理：设本系统程序完整运行一周期的时间是 Tp;看门狗的定时周期为 Ti，且 Ti>Tp;在程序运行一周期后就修改定时器的计数值;只要程序正常运行，定时器就不会溢出;若由于干扰等原因:使系统不能在 Tp 时刻修改定时器的计数值;定时器将在 Ti 时刻溢出;引发系统复位，使系统得以重新运行;从而起到监控作用;

控制方法：输入时钟为 MCLK（该时钟频率等于系统的主频），它经过两级分频最后将分频后的时钟作为该定时器的输入时钟当计数器期满后可以产生中断或者复位信号

4　了解 ADS 的集成开发环境和使用方法。

ARM 公司推出的新一代 ARM 集成开发工具（相对于 SDT）全称：Metrowerks CodeWarrior for ARMDeveloper Suite v1.2，ADS 的 CodeWarrior IDE 基于 Metrowerks CodeWarrior IDE 4.2 版本，它经过适当的裁剪以支持 ADS 工具链

ADS 的主要组成部件有：命令行开发工具；ARM 运行时库；GUI 开发环境(Code Warrior 和 AXD)。有了这些部件，用户就可以为 ARM 系列的 RISC 处理器编写和调试自己的开发应用程序

5　了解嵌入式远程 GDB 的使用方法

定义：GNU 的调试器称为 gdb(ddd)，该程序是一个交互式工具，工作在字符模式

GDB 可完成如下的调试任务：设置断点；监视程序变量的值；程序的单步执行；　修改变量的值；

在可以使用 gdb 调试程序之前，必须使用-g 选项编译源文件。可在 Makefile 中如下定义 CFLAGS 变量：CFLAGS = -g；或者在使用 gcc 编译的时候加上-g 选项；$ gcc -g -o hello hello.c。运行 gdb 调试程序时通常使用如下的命令：
$ gdb progname；

6 DDD 运行机理

在 DDD 中，GDB 是作为独立的进程运行的，通过命令行接口与 DDD 进行交互；DDD 在事件循环时等待用户输入和 GDB 输出，同时等着 GDB 进入等待输入状态；当 GDB 可用时，下一条命令就会从命令队列中取出，送给 GDB；GDB 到达的输出由上次命令的回调过程来处理；这种异步机制避免了 DDD 在等待 GDB 输出时发生阻塞现象，到达的事件可以在任何时间得到处理

7 了解 arm-linux-*相关工具的作用及其使用方法。

1、C/C++编译器 arm-linux-gcc

arm-linux-gcc 主要功能是将源程序编译成汇编代码，它有十分丰富的命令选项，可以控制编译的各个阶段；varm-linux-gcc（交叉编译器）是嵌入式 Linux 系统平台下的 gcc。是 GNU 推出的功能强大、性能优越的多平台编译器，是 GNU 的代表作品之一；varm-linux-gcc 可以在多种硬体平台上编译出可执行程序的超级编译器；其执行效率与一般的编译器相比平均效率要高 20%~30%

varm-linux-gcc 语法形式：arm-linux-gcc [option | filename]...
通常情况下，产生一个新的程序需要经过四个阶段：预处理、编译、汇编,链接；当然我们可以通过参数决定该编译操作执行到何步结束
2、汇编器 arm-linux-as
汇编器 arm-linux-as 将 arm-linux-gcc 编译的汇编代码转换为目标代码
arm-linux-as 语法格式：arm-elf-as [OPTION…] [汇编文件…]
3、连接器 arm-linux-ld
在编写一个大的程序时，经常把它分成许多独立的模块，这时需要连接器所有的模块组合起来，并结合 c 函数库和初始化代码，产生最后可执行的文件
语法格式：arm-linux-ld [option] file ...
4、库管理器 arm-linux-ar
可以使用 ar 程序建立静态库，把几个小文件合并成一个大文件；建立静态库时，必须把几个.O 文件合并成一个单独的.a 文件

8 了解 CodeBlocks 软件的基本使用方法。

创建应用程序工程打开 Code::Blocks IDE；新建一个工程：File-New-Project；选择控制台应用；选择语言类型（这里选择 C）；给工程取名，设置工程路径；设置使用的编译器(默认)；查看 main.c 源码；设置交叉编译器；选择交叉编译器；选择 ARM 交叉编译器；编译器和调试器设置；设置编译器；编译、连接。

Code::Blocks 应用程序调试：调试器设置；设置开发板 IP 地址和端口号；在开发板上启动 gdbserver；调试——Start；

9 了解设备驱动和一般应用程序的区别。

应用程序是一个进程：编程从主函数 main()开始，主函数 main()返回即是进程结束

驱动程序是一系列内核函数：驱动程序向内核添加了一些函数，是内核的一部分
• Open()• Release()• Read()• Write()
§ 这些函数由内核在适当的时候来调用
§ 这些函数可以用来完成硬件访问等操作

10 了解设备驱动的基本作用。

系统调用：是操作系统内核和应用程序之间的接口
设备驱动程序：是操作系统内核和机器硬件之间的接口；为应用程序屏蔽了硬件的细节
在应用程序看来，硬件设备只是一个设备文件：应用程序可以象操作普通文件一样对硬件设备进行操作
设备驱动程序是内核的一部分，它完成以下的功能：对设备初始化和释放；把数据从内核传送到硬件和从硬件读取数据；读取应用程序传送给设备文件的数据和回送应用程序请求的；数据检测和处理设备出现的错误；

11 了解 Linux 支持的三类的硬件设备：字符（• 是一个顺序的数据流设备• 是指在 I/O 传输过程中以字符为单位进行传输的设备，例如键盘，打印机等• 注意，以字符为单位并不一定意味着是以字节为单位• 不具备缓冲区，所以对这种设备的读写是实时的• 可以象文件一样访问字符设备，字符设备驱动程序负责实现这些行为• 这样的驱动程）、块（• 具有一定结构的随机存取设备• 块设备将信息存储在固定大小的块中，每个块都有自己的地址• 待时机成熟时，从缓存一次性写入设备或从设备中一次性读出放入缓冲区• 块设备的基本特征是每个块都能独立于其它块而读写。磁盘是最常见的块设备• 在大多数 Unix 系统中，只能将块设备看作多个块进行访问，一个块设备通常是 1K 字节数据• Linux 允许象字符设备那样读取块设备——允许一次传输任意数目的字节）和网络设备（• 任何网络事务处理都是通过接口实现的，即，可以和其他宿主交换数据的设备• 通常接口是一个硬件设备，但也可以象 loopback（回路）接口一样是软件工具• 由于不是面向流的设备，所以网络接口不能象/dev/tty1 那样简单地映射到文件系统的节点上• Unix 调用这些接口的方式是给它们分配一个独立的名字（如 eth0）－ 这样的名字在文件系统中并没有对应项• 内核和网络设备驱动程序之间的通信与字符设备驱动程序和块设备驱动程序与内核间的通信完全不一样－ 内核不再调用 read，write，它调用与数据包传送相关的函数）。

12 了解引入模块的基本原理和模块的本质。

原理：vLinux 内核是一个整体结构：因此向内核添加任何东西，或者删除某些功能，都十分困难；为了解决这个问题引入了模块机制：从而可以动态的向内核中添加或者删除模块
本质：利用运载工具将一系列二进制函数搬运到内核，搬运到内核中的二进制函数可以被内核或用户所使用。

13 掌握模块的加载和卸载方法。

编写模块卸载函数：释放内存；注销设备；
static void __exit leddev_exit(void)

```
{
unregister_chrdev(leddevno,"led");
}
module_init(leddev_init);
module_exit(leddev_exit
```

模块编译并加载：Step 2：模块编译

编写 Makefile

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

模块编译并加载

Step 3:

```
make
```

Step4：insmod leddev.ko

编写应用程序操作系统下 LED 灯闪烁：

```
// ledapp.c
#define LEDON 1
#define LEDOFF 0
void main()
{ fd=open( “/dev/led” ); //打开设备
for(...)
{ ioctl(fd, LEDON); sleep(5);
ioctl(fd, LEDOFF); sleep(5);
}}
```

14 了解 Qt 程序的优点。

针对多个平台只需编写一次代码；创建令人意想不到的用户体验(• Qt 提供了应用程序生成块)；事半功倍且倍道而进(• 不论是使用全新的 Qt Creator 跨平台 IDE 还是仅是 Qt 本身，Qt 都易学易用• 而且由于有了 Qt 模块化的类库，您可以更多地关注创新，无须在平台本身编码上花费过多时间，这样就可将软件快速推向市场)；在单一应用程序中混合网络和本地代码(• Qt 集成了 WebKit 网络渲染引擎)

16 掌握 Qt 程序的信号和槽的工作机理。

信号和槽: 信号和槽机制是QT 的核心机制:要精通QT 编程就必须对信号和槽有所了解.信号和槽是一种高级接口:应用于对象之间的通信，它是 QT 的核心特性;

也是 QT 区别于其它工具包的重要地方.信号和槽是 QT 自行定义的一种通信机制:它独立于标准的 C/C++ 语言;因此要正确的处理信号和槽，必须借助一个称为 moc（Meta Object Compiler）的 QT 工具;该工具是一个 C++ 预处理程序，它为高层次的事件处理自动生成所需要的附加代码.

信号与槽过程，优点：（1）所有从 QObject 或其子类(例如 Qwidget) 派生

的类都能够包含信号和槽：当对象改变其状态时，信号就由该对象发射(emit)出去；这就是对象所要做的全部事情，它不知道另一端是谁在接收这个信号；这是真正的信息封装，它确保对象被当作一个真正的软件组件来使用；槽用于接收信号，但它们是普通的对象成员函数；一个槽并不知道是否有任何信号与自己相连接；而且，对象并不了解具体的通信机制。（2）可以将很多信号与单个的槽进行连接。（3）也可以将单个的信号与很多的槽进行连接。（4）甚至于将一个信号与另外一个信号相连接也是可能的，这时无论第一个信号什么时候发射系统都将立刻发射第二个信号。（5）总之，信号与槽构造了一个强大的部件编程机制。

信号：当某个信号对其客户或所有者发生的内部状态发生改变，信号被一个对象发射；信号的声明是在头文件中进行的，QT 的 signals 关键字指出进入了信号声明区，随后即可声明自己的信号；

槽：槽是普通的 C++ 成员函数，可以被正常调用；当与其关联的信号被发射时，这个槽就会被调用；槽是普通的成员函数，与其它的函数一样，它们也有存取权限（同普通的 C++ 成员函数一样，槽函数也分三种类型• 即 public slots、private slots 和 protected slots）；槽也能够声明为虚函数，这也是非常有用的。

17 掌握 Make 及 Makefile 文件使用方法

利用 make 工具来自动完成编译工作。包括：如果仅修改了某几个源文件，则只重新编译这几个源文件；如果某个头文件被修改了，则重新编译所有包含该头文件的源文件。make 工具通过 makefile 文件来实现；在 makefile 后，就可以调用 make 命令生成和维[redacted]目标文件。命令格式为：Make [option] [macrodef] [target]• — Option 指定 make 工作的行为• — Macrodef 给出执行 makefile 时的宏值• — Target 是 makefile 中的目标之一

makefile 使用实例：

第一步：编辑源文件

§ #vi hello.c

v 第二步：建立 makefile

§ #vi makefile

v 第三步：编辑 makefile

21 掌握 Linux 设备驱动模型的构建方法。

（1）注册设备：在系统初启，或者模块加载时候，必须将设备登记到相应的设备数组，并返回设备的主驱动号

（2）定义功能函数：对于每一个驱动函数来说，都有一些和此设备密切相关的功能函数，那最常用的块设备或者字符设备来说，都存在着诸如 open() read() write() ioctrol()这一类的操作；当系统使用这些调用时，将自动的使用驱动函数中特定的模块，来实现具体的操作；而对于特定的设备，上面的系统调用对应的函数是一定的。

（3）卸载设备

（4）注册设备：在系统初启，或者模块加载时候，必须将设备登记到相应的设备数组，并返回设备的主驱动号

```
int register_chrdev(unsigned int major,
const char *name,
struct file_operation *fops);
```

或者（不确定）：理解设备原理、掌握设备硬件电路图；写好中断服务程序（不一定需要）；实现接口函数；注册设备。

ARM 微处理器中共有 37 个寄存器，在这些寄存器中，R13 寄存器的作用是（堆栈指针），R14 的作用为（链接寄存器），R15 作用为（程序计数器）

嵌入式操作系统与嵌入式操作系统区别：

嵌入式操作系统

嵌入式操作系统是一种支持嵌入式系统应用的操作系统软件，它是嵌入式系统(包括硬、软件系 统)极为重要的组成部分，通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等 browser。嵌入式操作系 统具有通用操作系统的基本特点，如能够有效管理越来越复杂的系统资源；能够把硬件虚拟化，使得开发人员从繁忙的驱动程序移植和维护中解脱出来；能够提供库 函数、驱动程序、工具集以及应用程序 。与通用操作系统相比较，嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具 有较为突出的特点。

嵌入式系统

嵌入式系统一般指非 pc 系统，有计算机功能但又不称之为计算机的设备或器材。它是以应用为中心，软硬件可裁减的，适应应用系统对功能、可靠性、成本、体积、功耗等综合性严格要求的专用计算机系统。简单地说，嵌入式系统集系统的应用软件与硬件于一 体，类似于 pc 中 bios 的工作方式，具有软件代码小、高度自动化、响应速度快等特点，特别适合于要求实时和多任务的体系。嵌入式系统主要由嵌入式处理 器、相关支撑硬件、嵌入式操作系统及应用软件系统等组成，它是可独立工作的“器件”。

嵌入式系统的硬件部分，包括处理器/微处理器、存储器及外设器件和 i/o 端口、图形控制器 等。嵌入式系统有别于一般的计算机处理系统，它不具备像硬盘那样大容量的存储介质，而大多使用 eprom、eeprom 或闪存 (flash memory)作为存储介质。软件部分包括操作系统软件(要求实时和多任务操作)和应用程序编程。应用程序控制着系统的运作和行为；而操作 系统控制着应用程序编程与硬件的交互作用。

CISC 与 RISC 的特点：

CISC (Complex Instruction Set Computer)

复杂指令集计算机：指令集中包含了许多功能强大的指令，指令的长度不等，不便于指令的流水线执行；典型的如 Intel X86 处理器

RISC (Reduced Instruction Set Computer)

精简指令集计算机：采用一个有限的简单指令集，即优先选取使用频率最高的简单指令，避免复杂指令；每条指令长度固定，指令格式和寻址方式种类减少，执行时间短，便于指令的流水线优化；为弥补指令功能，CPU 配备大量的通用寄存器；典型的如 MIPS、ARM 处理器等

RISC 和 CISC 特点分明，各有优势

目前微处理器 RISC 和 CISC 的界限开始模糊：不少传统的 CISC 开始吸收 RISC 的优点；采用一种类 RISC 的设计方式，实现 RISC/CISC 混合

架构；典型的有 Intel X86 处理器

Makefile 函数：

8.2.2 $(patsubst PATTERN,REPLACEMENT,TEXT) 函数名称：模式替换函数—patsubst。
函数功能：搜索“TEXT”中以空格分开的单词，将否符合模式“TATTERN”替换为“REPLACEMENT”。参数“PATTERN”中可以使用模式通配符“%” 来代表一个单词中的若干字符。如果参数“REPLACEMENT”中也包含一个“%”，那么“REPLACEMENT”中的“%”将是“TATTERN”中的那个“%”所代表的字符串。在“TATTERN”和“REPLACEMENT”中，只有第一个“%”被作为模式字符来处理，之后出现的不再作模式字符（作为一个字符）。在参数中如果需要将第一个出现的“%”作为字符本身而不作为模式字符时，可使用反斜杠“\”进行转义处理（转义处理的机制和使用静态模式的转义一致，具体可参考 5.12.1 静态模式规则的语法一小节）。
返回值：替换后的新字符串。
函数说明：参数“TEXT”单词之间的多个空格在处理时被合并为一个空格，并忽略前导和结尾空格。
示例：
$(patsubst %.c,%.o,x.c.c bar.c)
把字串“x.c.c bar.c”中以.c 结尾的单词替换成以.o 结尾的字符。函数的返回结果是“x.c.o bar.o”
本文的第六章在 变量的高级用法的第一小节 中曾经讨论过变量的替换引用，它是一个简化版的“patsubst”函数在变量引用过程的实现。变量替换引用中：
$(VAR:PATTERN=REPLACEMENT)
就相当于：
$(patsubst PATTERN,REPLACEMENT,$(VAR))
而另外一种更为简单的替换字符后缀的实现：
$(VAR:SUFFIX=REPLACEMENT)
它等于：
$(patsubst %SUFFIX,%REPLACEMENT,$(VAR))
例如我们存在一个代表所有.o 文件的变量。定义为“objects = foo.o bar.o baz.o”。为了得到这些.o 文件所对应的.c 源文件。我们可以使用以下两种方式的任意一个：
$(objects:.o=.c)
$(patsubst %.o,%.c,$(objects))

4.4.3 函数 wildcard 之前提到过，在规则中，通配符会被自动展开。但在变量的定义和函数引用时，通配符将失效。这种情况下如果需要通配符有效，就需要使用函数“wildcard”，它的用法是：$(wildcard PATTERN...) 。在 Makefile 中，它被展开为已经存在的、使用空格分开的、匹配此模式的所有文件列表。如果不存在任何符合此模式的文件，函数会忽略模式字符并返回空。需要注意的是：这种情况下规则中通配符的展开和上一小节匹配通配符的区别。
一般我们可以使用“$(wildcard *.c)”来获取工作目录下的所有的.c 文件列表。复杂一些用法：可以使用“$(patsubst %.c,%.o,$(wildcard *.c))”，首先使用“wildcard”函数获取工作目录下的.c 文件列表；之后将列表中所有文件名的后缀.c 替换为.o。

这样我们就可以得到在当前目录可生成的.o 文件列表。因此在一个目录下可以使用如下内容的 Makefile 来将工作目录下的所有的.c 文件进行编译并最后连接成为一个可执行文件：

```
#sample Makefile
objects := $(patsubst %.c,%.o,$(wildcard *.c))
foo : $(objects)
cc -o foo $(objects)
```

这里我们使用了 make 的隐含规则来编译.c 的源文件。对变量的赋值也用到了一个特殊的符号(:=)。关于变量定义可参考 6.2 两种变量定义一节。函数“patsubst”可参考 8.2 文本处理函数一节