

一、选择题

- 1、和 PC 机系统相比，下列哪点不是嵌入式系统所特有的：(C)
A、系统内核小 B、专用性强 C、可执行多任务 D、系统精简
- 2 ARM 公司专门从事 (A)
A、基于 RISC 技术芯片设计开发 B、ARM 芯片生产 C、软件设计 D、ARM 芯片销售
- 3、ADD R0, R1, #3 属于 (A) 寻址方式。
A、立即寻址 B、多寄存器寻址 C、寄存器直接寻址 D、相对寻址
- 4、GET 伪指令的含义是：(A)
A、包含一个外部文件 B、定义程序的入口 C、定义一个宏 D、声明一个变量
- 5、存储一个 32 位数 0x876165 到 2000H~2003H 四个字节单元中，若以小端模式存储，则 2000H 存储单元的内容为：(C)
A、0x00 B、0x87 C、0x65 D、0x61
- 6、μCOS-II 操作系统不属于：(C)
A、RTOS B、占先式实时操作系统 C、非占先式实时操作系统 D、嵌入式实时操作系统
- 7、若 R1=2000H, (2000H)=0x86, (2008H)=0x39, 则执行指令 LDR R0, [R1, #8]! 后 R0 的值为：(D)
A、0x2000 B、0x86 C、0x2008 D、0x39
- 8、寄存器 R13 除了可以做通用寄存器外，还可以做：(C)
A、程序计数器 B、链接寄存器 C、栈指针寄存器 D、基址寄存器
- 9、FIQ 中断的入口地址是：(A)
A、0x0000001C B、0x00000008 C、0x00000018 D、0x00000014
- 10、ARM 指令集和 Thumb 指令集分别是 (D) 位的。
A、8 位, 16 位 B、16 位, 32 位 C、16 位, 16 位 D、32 位, 16 位
- 11、ARM 寄存器组有 (D) 个寄存器。
A、7 B、32 C、6 D、37
- 12、在所有工作模式下，(A) 都指向同一个物理寄存器，即各模式共享
A、R0-R7 B、R0-R12 C、R8-R12 D、R13, R14
- 13、当异常发生时，寄存器 (A) 用于保存 CPSR 的当前值，从异常退出时则可由它来恢复 CPSR。
A、SPSR B、R13 C、R14 D、R15
- 14、能实现把立即数 0X3FF5000 保存在 R0 中的指令是 (A)
A、LDR R0, = 0X3FF5000 B、LDR R0, 0X3FF5000
C、MOV R0, 0X3FF5000 D、MOV R0, =0X3FF5000
- 15、ARM 指令 LDMIA R0, {R1, R2, R3, R4} 实现的功能是 ()
A、 $R1 \leftarrow [R0]$, $R2 \leftarrow [R0+4]$, $R3 \leftarrow [R0+8]$, $R4 \leftarrow [R0+12]$
B、 $R1 \leftarrow [R0]$, $R2 \leftarrow [R0-1]$, $R3 \leftarrow [R0-2]$, $R4 \leftarrow [R0-3]$
C、 $R1 \leftarrow [R0]$, $R2 \leftarrow [R0+1]$, $R3 \leftarrow [R0+2]$, $R4 \leftarrow [R0+3]$
D、 $R1 \leftarrow [R0]$, $R2 \leftarrow [R0-4]$, $R3 \leftarrow [R0-8]$, $R4 \leftarrow [R0-12]$
- 16、指令 MSR CPSR_C, R0 实现的功能是 (B)
A、传送 R0 的内容到 CPSR 中，更新 CPSR
B、传送 R0 的内容到 CPSR 中，但仅修改其控制域
C、传送 CPSR 的内容到 R0
D、传送 CPSR 的控制域内容到 R0
- 17、(D) 伪指令用于定义一个代码段或数据段
A、RLIST B、DCB C、MAP D、AREA
- 18、在上机实验中，ADS 开发环境下，调试程序用 (A) 软件

A、AXD B、Debug C、Gcc D、Make

19、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

20、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

21、按照 ARM 过程调用标准 (APCS)，栈指针使用 (B) 寄存器

- A. R0 B. R13 C. R14 D. R15

22、嵌入式系统应用软件一般在宿主机上开发，在目标机上运行，因此需要一个 (B) 环境。

- A. 交互操作系统 B. 交叉编译 C. 交互平台 D. 分布式计算

23、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

24、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

25、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

26、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。

27、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

28、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

29、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。

30、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

31、嵌入式 GUI 设计不包括下面哪项：(Z)

- 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

32、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

33、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

- 34、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 Δt ，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) Δt 。
- A. 8 B. 10 C. 12 D. 14
- 35、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
- 36、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
- 37、“冲击波”病毒属于(A)类型的病毒。
- A. 蠕虫 B. 文件 C. 引导区 D. 邮件
- 38、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。
- A. 8 B. 256 C. 512 D. 1024
- 39、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
- 40、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
- 41、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
- 42、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
- 43、下面哪种操作系统不属于商用操作系统。(B)
- A. windows xp B. Linux C. VxWorks D. WinCE
- 44、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
- 45、下面哪种不属于嵌入式系统的调试方法。(D)
- A. 模拟调试 B. 软件调试 C. BDM/JTAG 调试 D. 单独调试
- 46、在嵌入式 ARM 处理器中，下面哪种中断方式优先级最高。(A)
- A. Reset B. 数据中止 C. FIQ D. IRQ
- 47、NAND FLASH 和 NOR FLASH 的区别正确的是。(D)
- A. NOR 的读速度比 NAND 稍慢一些 B. NAND 的写入速度比 NOR 慢很多
C. NAND 的擦除速度远比 NOR 的慢 D. 大多数写入操作需要先进行擦除操作
- 48、下面哪种嵌入式操作系统很少用在手机上。(D)
- A. Symbian B. linux C. Win CE D. VxWorks
- 49、 $0x17 \& 0x11$ 的运算结果是。(A)
- (A) 0x01 (B) 0x11 (C) 0x17 (D) 0x07
- 50、以下哪种方式不属于文件系统的格式。(B)
- (A) FAT (B) DOS (C) NTFS (D) EXT
- 51、在将 uc/os 操作系统移植到 ARM 处理器上时，以下哪些文件不需要修改。(A)
- (A) OS_CORE.C (B) INCLUDE.H
(C) OS_CPU.H (D) OSTASKINIT

- 52、下面哪个特点不属于 USB 设备的特点。 (B)
(A) 串行通信方式 (B) 不可热拨插
(C) 分 HOST、DEVICE 和 HUB (D) 通信速率比 RS232 快
- 53、下列操作系统不属于嵌入式操作系统的是: (A)。
(A) WinXP (B) uCOS (C) Linux (D) WinCE
- 54、ARM 体系结构中 3 级流水结构的第 2 阶段是 (A)。
(A) 取指 (B) 译码 (C) 执行 (D) 编译
- 55、ARM 一个字占 (B) 个字节。
(A) 1 (B) 4 (C) 2 (D) 8
- 56、作为堆栈指针寄存器的是 (A)。
(A) R12 (B) R13 (C) R14 (D) R15
- 57、具有独立 R8—15 寄存器的工作模式是 (D)。
(A) 用户 (B) 系统 (C) 中断 (D) 快中断
- 58、可以切换到 Thumb 状态的指令是: (A)。
(A) LDR R0, =LABEL+1 BX R0
(B) LDR R0, =LABEL BL R0
(C) LDR R0, =LABEL+1 BX R1
(D) LDR R0, LABEL+1 BL R0
- 59、可以切换到 ARM 状态的指令是: (B)。
(A) LDR R0, =LABEL+1 BX R0 (B) LDR R0, =LABEL BX R0
(C) LDR R0, =LABEL+1 BL R1 (D) LDR R0, LABEL+1 BL R0
- 60、IRQ 中断返回指令是: (A)。
(A) MOV PC, R14 (B) MOV PC, R14_SVC
(C) SUBS PC, R14_IRQ, #4 (D) SUBS PC, R14_FIQ, #4
- 61、下列中断优先级最低的是: (D)。
(A) FIQ (B) IRQ (C) 中止 (D) SWI
- 62、R1 内容是 4000H, 4000H (4 字节对齐) 开始向上依次存放数据 1, 2, 3, 4, 5, 执行指令 LDMIA R1!, {R0, R1, R2, R3} 后, R1 中数据是: (A)。
(A) 2 (B) 3 (C) 4 (D) 5
- 63、R1 内容是 4000H, 4000H (4 字节对齐) 开始向上依次存放数据 1, 2, 3, 4, 5, 执行指令 LDMIB R1!, {R0, R1, R2, R3} 后, R1 中数据是: (B)。
(A) 2 (B) 3 (C) 4 (D) 5
- 64、下列不属于 ARM 体系结构特点的是: (D)。
(A) 大而统一的寄存器文件。 (B) 数据的加载和存储结构。
(C) 地址的自动增加和减少。 (D) 不固定的指令长度。
- 65、ARM7TDMI 中, T 的含义是 (A)。
(A) Thumb 指令扩展 (B) 支持片上调试 (C) 硬件乘法指令 (D) 观察点硬件
- 66、R0 内容是 1000H, 执行 STR R1, [R0, #4]! 后, R0 内容是 (B)。
(A) 1000H (B) 1004H (C) 1008H (D) 100CH
- 67、作为链接寄存器的是 (C)。
(A) R12 (B) R13 (C) R14 (D) R15
- 68、不属于异常模式的是 (B)。
(A) 管理 (B) 系统 (C) 中断 (D) 快中断
- 69、FIQ 和 IRQ 分别为禁止和允许时, CPSR 中 F 和 I 的设置是: (A)。

- (A) 10 (B) 01 (C) 00 (D) 11
- 70、FIQ 和 IRQ 分别为允许和禁止时，CPSR 中 F 和 I 的设置是：(B)。
- (A) 10 (B) 01 (C) 00 (D) 11
- 71、FIQ 中断返回指令是：(D)。
- (A) MOVS PC, R14 (B) MOVS PC, R14_SVC
(C) SUBS PC, R14_IRQ, #4 (D) SUBS PC, R14_FIQ, #4
- 61、下列中断优先级最高的是：(B)。
- (A) FIQ (B) 未定义指令 (C) 中止 (D) 复位
- 62、ARM9 体系结构采用了 (C)。
- A、3 级流水 B、4 级流水 C、5 级流水 D、6 级流水
- 63、ARM 处理器总共有 (D)。
- A、6 个寄存器 B、16 个寄存器 C、31 个寄存器 D、37 个寄存器
- 64、ADS1.2 集成开发环境中 ARM linker 选项卡中 R0 Base 用来设置 (A)。
- A、程序代码的存放的起始地址 B、程序数据存放的起始地址
C、程序入口地址 D、程序跳转地址
- 65、ADS1.2 集成开发环境中 ARM linker 选项卡中 RW Base 用来设置 (B)。
- A、程序代码的存放的起始地址 B、程序数据存放的起始地址
C、程序入口地址 D、程序跳转地址
- 66、ARM 汇编器伪指令中 # 是 (C) 伪指令的简写。
- A、SPACE B、MAP C、FIELD D、DCD
- 67、ARM 汇编器伪指令中 % 是 (A) 伪指令的简写。
- A、SPACE B、MAP C、FIELD D、DCD
- 68、ARM 要求中断向量表必须放在从 (A) 开始的连续的 32 字节空间内。
- A、0x00000000 B、0x30000000 C、0x30100000 D、0x0000FFFF
- 69、ARM 可以有 (B) 种运行状态。
- A、6 B、7 C、3 D、8
- 70、|Image\$\$R0\$\$Limit| 表示 (A)。
- A、R0 区末地址后面的地址 B、R0 区首地址后面的地址
C、ZI 区在 RAM 中的起始地址 D、ZI 区在 RAM 结束地址后面的地址
- 71、|Image\$\$ZI\$\$Limit| 表示 (D)。
- A、R0 区末地址后面的地址 B、R0 区首地址后面的地址
C、ZI 区在 RAM 中的起始地址 D、ZI 区在 RAM 结束地址后面的地址
- 72、C 语言编程中，ATPCS 规定堆栈采用 (A)。
- A、满递减堆栈 (FD) 型 B、空递减堆栈 (ED) 型
C、满递增堆栈 (FA) 型 D、空递增堆栈 (EA) 型
- 73、ATPCS 规定堆栈采用满递减堆栈类型，即 (A)。
- A、堆栈通过减小存储器地址而向下增长 B、堆栈通过减小存储器地址而向上增长
C、堆栈通过增大存储器地址而向下增长 D、堆栈通过增大存储器地址而向上增长
- 74、S3C2410A 的内核供电是 (B)。
- A、1.9V B、1.8V C、5V D、3.3V
- 75、S3C2410A 的存储器供电是 (D)。
- A、1.9V B、1.8V C、5V D、3.3V
- 76、S3C2410A 的内部缓冲器 SRAM 存储空间为 (A)。
- A、4KB B、16KB C、32MB D、64MB

- 77、S3C2410A 的每个 BANK 区为 (C)。
- A、32M B、64M C、128M D、1G
- 78、S3C2410A 中的中断控制寄存器能够接收来自 (B) 个中断源的请求。
- A、2 个 B、56 个 C、5 个 D、32 个
- 79、S3C2410A 共有 (D) 个中断请求信号。
- A、2 个 B、56 个 C、5 个 D、32 个
- 80、异常向量中保存的通常是 (A)。
- A、异常处理程序的地址 B、初始化为 0000H 的值
C、初始化为 FFFFH 的值 D、未经初始化的 ZI 空间
- 81、下列不属于异步串行通信可使用的停止位为 (D)。
- A、1 位 B、1.5 位 C、2 位 D、2.5 位
- 82、LCD 显示缓冲中，332RGB 格式是指 (C)。
- A、显示缓冲总计 332KB B、红、绿、蓝三个颜色分量各占 332 字节
C、红、绿、蓝三个颜色分量占 3 位、3 位、2 位。 D、红、绿、蓝三个颜色分量各占 332 位
- 83、RedHat 9.0 linux 的目录显示命令是 (A)。
- A、ls B、cp C、cd D、gvim
- 84、RedHat 9.0 linux 的设备挂载命令是 (B)。
- A、ls B、mount C、umount D、gvim
- 85、利用 VIVI 向 NAND FLASH 烧写内核的命令是 (B)。
- A、load flash vivi x B、load flash kernel x
C、load flash root x D、load flash param x
- 86、下述汇编语言哪条属于伪指令 (C)。
- A、STR R0, [R7], #8 B、STRB R0, [R9, #const-struct] C、LDR R3, =0xFF0 D、B .
- 87、寄存器 (B) 作为堆栈指针 SP。
- A、R12 B、R13 C、R14 D、R15
- 88、当上电或复位后，VIVI 启动，位于 NAND Flash 中的前 4KB 程序由 S3C2410A 自动复制到 (B) 中。
- A、BANK6 区的 SDRAM B、S3C2410A 内部的 RAM
C、NAND Flash 的未使用区域 D、SDRAM 的 RW 区域
- 89、在下列 ARM 处理器模式中，(D) 模式有自己独立的 R8-R14 寄存器。
- A. 系统模式 B. 终止模式 C. 中断模式 D. 快速中断模式
- 90、下面语句执行后，实现了 R0=[R1+R2*4] 功能的是 (A)。
- A、LDR R0, [R1, R2, LSL #2] B、LDR R0, [R1, R2, LSL #2]!
C、LDR R0, [R1], R2, LSL #2 D、LDR R0, [R2, R1, LSL #2]
- 91、下列哪条指令逆向减法指令？ (C)
- A、SUB R0, R1 B、SUBS R0, R1 C、SBC R0, R1 D、RSB R0, R1;
- 92、下列哪条指令能实现清除 R0 中的低 8 位，其他位保持不变？ (D)
- A、ADD R0, R0, #0xFF B、ORR R0, R0, #0xFF
C、EOR R0, R0, #0xFF D、BIC R0, R0, #0xFF
- 93、下列哪条指令是带状态切换的跳转指令 (D)
- A、B B、BL C、BX D、BLX
- 94、某总线有 104 根信号线，其中数据总线 (DB) 32 根，若总线工作频率为 33MHz，则其理论最大传输率为 C。
- (注：本题答案中的 B 表示 Byte)
- A. 33 MB/s B. 64MB/s C. 132 MB/s D. 164 MB/s
- 95、BL 和 BX 的指令的区别是 (D)。

- A. BL 是跳转并切换状态；BX 带返回的跳转并切换状态。
 B. BL 是带返回的连接跳转；BX 带返回的跳转并切换状态。
 C. BL 是跳转并切换状态；BX 是带返回的连接跳转。
 D. BL 是带返回的连接跳转；BX 是跳转并切换状态。
- 96、下列 ARM 寄存器与其用途的对应关系错误的是（ D ）。
- A. R13 (the Stack Point, SP, 堆栈指针)
 B. R14 (the Link Register, LR, 链路寄存器)；
 C. R15 (PC) D. R12 CPSR
- 97、ARM 汇编语句“ADD R0, R2, R3, LSL#1”的作用是（ A ）。
- A. $R0 = R2 + (R3 \ll 1)$ B. $R0 = (R2 \ll 1) + R3$
 C. $R3 = R0 + (R2 \ll 1)$ D. $(R3 \ll 1) = R0 + R2$
- 98、S3C2410X 的 ARM 处理器的状态寄存器包括：（ B ）。
- A. 1 个 SPSR 和 5 个 CPSR B. 1 个 CPSR 和 5 个 SPSR
 C. 2 个 SPSR 和 4 个 CPSR D. 2 个 CPSR 和 4 个 SPSR
- 99、以下说法正确的是（ C ）。
- A. C 语言程序总是从第一个的函数开始执行
 B. 在 C 语言程序中，要调用函数必须在 main() 函数中定义
 C. C 语言程序总是从 main() 函数开始执行
 D. C 语言程序中的 main() 函数必须放在程序的开始部分
- 100、现代电子设计方法包含了可测试技术，其中（ B ）接口是 IC 芯片的测试标准接口。
- A. BIST B. JTAG C. UART D. USB
- 101、以下叙述正确的是（ C ）。
- A. 宿主机与目标机之间只需建立逻辑连接即可
 B. 在嵌入式系统中，调试器与被调程序一般位于同一台机器上
 C. 在嵌入式系统开发中，通常采用的是交叉编译器
 D. 宿主机与目标机之间的通信只有串口和并口两种。
- 102、ARM9 处理器是（ C ）位处理器。
- A. 8 位 B. 16 位 C. 32 位 D. 64 位
- 103、IRQ 中断的入口地址是（ C ）。
- A. 0x00000000 B. 0x00000008 C. 0x00000018 D. 0x00000014
- 104、实现无失真放大，应使晶体三极管工作在（ B ）区域。
- A. 截止 B. 放大 C. 饱和 D. 上升
- 105、采用 RS232-C 串行通信至少需要三根线，其中不包括（ A ）。
- A. 电源线 B. 地线 C. 发送数据线 D. 接收数据线
- 106、在微型计算机中，采用中断方式的优点之一是（ C ）。
- A. 简单且容易实现 B. CPU 可以不工作 C. 可实时响应突发事件 D. 传送速度最快
- 107、若 4 输入或非门的输出完成，则多余输入端应（ C ）。
- A. 全部接高电平 B. 只需一个接高电平即可 C. 全部接地 D. 只需一个接地即可
- 108、实时操作系统中，两个任务并发执行，一个任务要等待其合作伙伴发来信息，或建立某个条件后再向前执行，这种制约性合作关系被成为（A）。
- A. 同步 B. 互斥 C. 调度 D. 执行
- 109、下列关于存储管理单元（MMU）说法错误的是（B）。
- A. MMU 提供的一个关键服务是使各个任务作为各自独立的程序在其自己的私有存储空间中运行。
 B. 在带 MMU 的操作系统控制下，运行的任务必须知道其他与之无关的任务的存储需求情况，这就简化了各

个任务的设计。

C. MMU 提供了一些资源以允许使用虚拟存储器。

D. MMU 作为转换器，将程序和数据的虚拟地址（编译时的连接地址）转换成实际的物理地址，即在物理主存中的地址。

110、寄存器 R15 用作（ B ）。

A. 栈指针寄存器 B. 程序计数器 C. 当前程序状态寄存器 D. 基地址寄存器

111、下列描述不属于 RISC 计算机的特点的是（C）。

A. 流水线每周期前进一步。 B. 更多通用寄存器。

C. 指令长度不固定，执行需要多个周期。

D. 独立的 Load 和 Store 指令完成数据在寄存器和外部存储器之间的传输。

112、用图形点阵的方式显示一个 16*16 点阵汉字需要（B）字节。

A、8 B、32 C、16 D、64

113、下面关于Shell 的说法，不正确的是：（D）

A. 操作系统的外壳 B. 用户与Linux 内核之间的接口程序

C. 一个命令语言解释器 D. 一种和C 类似的程序语言

114、存储一个 32 位数 0x2168465 到 2000H~2003H 四个字节单元中，若以大端模式存储，则 2000H 存储单元的内容为（ D）。

A、0x21 B、0x68 C、0x65 D、0x02

115、S3C2420X I/O 口常用的控制器是（D）。

A、端口控制寄存器（GPACON-GPHCON）。 B、端口数据寄存器（GPADAT-GPHDAT）。

C、外部中断控制寄存器（EXTINTN）。 D、以上都是。

116、嵌入式系统有硬件和软件部分构成，以下（C）不属于嵌入式系统软件。

A. 系统软件 B. 驱动 C. FPGA 编程软件 D. 嵌入式中间件

117、Cache用于存放主存数据的部分拷贝，主存地址与Cache 单元地址间的转换用（A）完成。

A. 硬件 B. 软件 C. 用户 D. 程序员

118、每种嵌入式操作系统都有自身的特点以吸引相关用户，下列说法错误的是（D）。

A. 嵌入式Linux 提供了完善的网络技术支持；

B. CLinux 是专门为没有MMU 的ARM 芯片开发的；

C. C/OS-II 操作系统是一种实时操作系统（RTOS）；

D. WinCE 提供完全开放的源代码。

119、下列CPSR 寄存器标志位的作用说法错误的是（D）。

A. N：负数 B. Z：零 C. C：进位 D. V：借位

120、在指令系统的各种寻址方式中，获取操作数最快的方式是（1）B。若操作数的地址包含在指令中，则属于（2）A 方式。

（1）A. 直接寻址 B. 立即寻址 C. 寄存器寻址 D. 间接寻址

（2）A. 直接寻址 B. 立即寻址 C. 寄存器寻址 D. 间接寻址

121、下面关于 DMA 方式的描述，不正确的是 C。

A. DMA 方式使外设接口可直接与内存进行高速的数据传输

B. DMA 方式在外设与内存进行数据传输时不需要 CPU 干预

C. 采用 DMA 方式进行数据传输时，首先需要进行现场保护

D. DMA方式执行I/O交换要有专门的硬件电路

122、在嵌入式处理器中，CACHE 的主要功能由（1）A 实现；某 32 位计算机的 CACHE 容量为 16KB（B：字节），CACHE 块的大小为 16B（B：字节），若主存与 CACHE 地址映像采用直接映像方式，则主存地址为 0x1234E8F8 的单元装入 CACHE 的地址（2）C；在下列 CACHE 替换算法中，平均命中率最高的是（3）D。

- (1) A. 硬件 B. 软件 C. 操作系统 D. 应用程序
- (2) A. 00010001001101 B. 01000100011010 C. 10100011111000 D. 11010011101000
- (3) A. 先入后出算法 (FILO) B. 随机替换算法 (RAND)
C. 先入先出算法 (FIFO) D. 近期最少使用算法 (LRU)
- 123、在嵌入式系统的存储结构中,存取速度最快的是 B。
- A. 内存 B. 寄存器组 C. Flash D. Cache
- 124、为了提高嵌入式软件的可移植性,应注意提高它的 D。
- A. 易用性 B. 简洁性 C. 可靠性 D. 设备无关性
- 125、嵌入式系统的开发通常是在交叉开发环境实现的,交叉开发环境是指 (A)
- A 在宿主机上开发,在目标机上运行 B 在目标机上开发,在宿主机上运行
C 在宿主机上开发,在宿主机上运行 D 在目标机上开发,在目标机上运行
- 126、指令ADD R2,R1,R1,LSR #2 中,LSR 的含义是 (B)。
- A 逻辑左移 B 逻辑右移 C 算术右移 D 循环右移
- 127、波特率9600bps 是指数据每秒传输 (B)
- A 9600 个字节 B 9600 个比特 C 9600 个字 D 9600 个字符
- 128、32 位体系结构的ARM 处理器有 B 种不同的处理器工作模式,和 B 个主要用来标识CPU 的工作状态和程序的运行状态的状态寄存器。
- A、7、7 B、7、6 C、6、6 D、6、7
- 129、ARM9 和ARM7 的重要区别是 (A)
- A ARM9 带有MMU 功能 B ARM9 支持Thumb 指令集
C ARM9 带有Cache 功能 D ARM9 是哈佛结构
- 130、在寄存器间接寻址方式中,指定寄存器中存放的是 (B)。
- A、操作数 B、操作数地址 C、转移地址 D、地址偏移量
- 131、嵌入式微控制器相比嵌入式微处理器的最大特点 (B)。
- A、体积大大减小 B、单片化 C、功耗低 D、成本高
- 132、中断向量是指 (C) 。
- A、中断断点的地址 B、中断向量表起始地址 C、中断处理程序入口地址 D、中断返回地址
- 133、(B) 不是进程和程序的区别。
- A. 程序是一组有序的静态指令,进程是一次程序的执行过程
B. 程序只能在前台运行,而进程可以在前台或后台运行
C. 程序可以长期保存,进程是暂时的
D. 程序没有状态,而进程是有状态的
- 134、在ARM 体系构建的嵌入式系统中,由电平模式触发的中断,其对应的中断标准应该在何时被清除? A
- A、当中断处理程序结束以后,才可以清除 B、进入相应的中断处理程序,即可以清除
C、产生IRQ 中断的时候,处理器自动清除 D、任何时候都可以清除
- 135、在ARM 体系结构中,从主动用户模式 (User) 切换到超级用户模式 (Supervisor),应采用何种方法? C
- A、直接修改CPU 状态寄存器 (CPSR) 对应的模式
B、先修改程序状态备份寄存器 (SPSR) 到对应的模式,再更新CPU 状态
C、使用软件中断指令 (SWI)
D、让处理器执行未定义指令

二、填空题

- 1、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。
- 2、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。

- 3、ARM 核有多个寄存器，其中大部分用于通用寄存器，有小部分作为专用寄存器，R15 寄存器用于存储 PC，R13 通常用来存储 SP。
- 4、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依曼，数据和指令分开使用不同接口的是哈佛结构。
- 5、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。
- 6、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。
- 7、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，另一种是大端对齐。
- 8、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。
- 9、一般而言，嵌入式系统的构架可以分为 4 个部分：处理器、（存储器）、输入/输出和软件，一般软件亦分为（操作系统）和应用软件两个主要部分。
- 10、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，（嵌入式微处理器）以及片上系统。
- 11、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。
- 12、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（硬实时操作系统）和软实时系统。
- 13、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（任务调度）。
- 14、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）。
- 15、嵌入式 CPU 有哈佛体系结构和（冯诺依曼体系结构），其中哈佛体系结构数据空间和地址空间（分开），ARM920T 采用（哈佛体系）的内核架构。
- 16、ARM7TDMI 采用（3）级流水线结构，ARM920TDMI 采用（5）级流水线。
- 17、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。
- 18、ARM 内核有四个功能模块，即_____、_____、_____、_____，可供生产厂商根据不同用户的要求来配置生产 ARM 芯片。
- 19、ARM7TDMI 与 ARM720T 的一个重要区别在于后者有_____、_____。
- 20、ARM 微处理器共有_____个_____位寄存器，其中_____个为通用寄存器，_____个为状态寄存器。
- 21、ARM 体系结构可用两种方法存储字数据，具体为_____、_____。
- 22、协处理器主要控制：_____、_____、_____。
- 23、ARM 微处理器在较新的体系结构中支持两种指令集：_____、_____。
- 24、ARM 处理器有 9 种基本寻址方式，分别是：____、____、____、____、____、____、____、____、____。
- 25、ARM 指令集可以分为 6 类，分别是：____、____、____、____、____、____。
- 26、在 ARM 的汇编程序中，有如下几种伪指令：____、____、____、____、____。
- 27、汇编语言与 C/C++ 的混合编程通常有以下几种方式：____、____、____。
- 28、ARM C/C++ 编译器支持和产生以下几种格式的文件：____、____、____、____、____。
- 29、ARM 嵌入式系统主要由____、____和____构成。
- 30、常用的嵌入式外围设备有____、____和____三类。
- 31、总线通常包括____、____和____。
- 32、目前流行的嵌入式操作系统主要有：____、____、____和____。
- 33、嵌入式操作系统的调试一般包括：____调试和____调试。
- 34、作为高速缓存的存储器主要有____、____和____。

35. 目前动态 RAM 主要有_____和_____。
36. ARM 有从外部_____启动的外启动和从片上_____启动的内启动两种启动方式。
37. Nand-Flash 闪存每个块的最大擦写次数是_____万次，而 Nor 的擦写次数是_____万次。
38. MTD 核心层分为：_____层、_____层和_____层。
39. Nor-Flash 常用于存放_____，而 Nand-Flash 存放_____。
40. 一个嵌入式系统结构有典型的四层组成，包括：_____、_____、_____和_____。
41. 假设 R1=0x31, R2=0x2 则执行指令 ADD R0, R1, R2, LSL #2 后, R0 的值是 (0x39)。
18. T、D、M、I
19. 存储空间管理单元 (MMU) 和 8K Cache 片内缓存 (提高 CPU 性能)
20. 37、32、31、6
21. 大端格式和小端格式
22. 片内的 MMU、指令和数据缓存 (IDC)、写缓冲 (Write Buffer)
23. ARM 指令集、Thumb 指令集
24. 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
25. 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
26. 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
27. 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
28. filename.c 源文件、filename.h 头文件、filename.o 目标文件、filename.s 汇编代码文件、filename.lst 列表文件
29. 嵌入式处理器、相关支撑硬件、嵌入式软件系统
30. 存储设备、通信设备、显示设备
31. 数据总线、地址总线、控制总线
32. Vxwork、WinCE、Linux、pSOS
33. 操作系统、应用程序
34. SRAM、DRAM、Flash ROM
35. SDRAM、DDR
36. SDRAM、ROM
37. 100、10
38. 用户模块接口、MTD 抽象、MTD 设备驱动模块
39. 系统代码、用户信息
40. 硬件层、中间层、软件层、功能层。

三、简答题

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，具有系统测试和可靠性评估体系；

- 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

2、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。）

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？

答：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、请详细描述 bootloader 启动代码的设计流程。

答：bootloader 的设计流程：（1）初始化看门狗（2）关中断（3）设置处理器时钟（4）初始化 SDRAM 器件（5）初始化串口（6）实现 shell 命令解释器功能，（7）提供通过串口的下载功能 download 和执行功能 go。

5、目前嵌入式操作系统有哪些？

答：1) μ C/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

6、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

7、嵌入式系统开发的基本流程？

答：1) 系统定义与需求分析；2) 系统设计方案的初步确立；
3) 初步设计方案性价比评估与方案评审论证；4) 完善初步方案、初步方案实施；
5) 软硬件集成测试；6) 系统功能性能测试及可靠性测试。

8、什么是可编程片上系统？

答：用可编程逻辑技术把整个系统放到一块硅片上，称作可编程片上系统 SOPC。它是一种特殊的嵌入式系统，首先它是 SOC，即由单个芯片实现整个系统的主要逻辑功能，具有一般 SOC 基本属性；其次，它又具备软硬件在系统可编程的功能，是可编程系统，具有可裁剪、可扩充、可升级等灵活的设计方式。

9、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。

10、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)。

11、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。

11、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。

12、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 (Prefetch 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：MOVSP, PC, R14；2) IRQ, FIQ, 预取中止的返回：SUBSP, PC, R14, #4；

3) 数据中止返回并重新存取：SUBSP, 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)。

13、什么是小端和大端存储器组织？

答：1) 小端存储器组织是较高的有效字节存放在较高的存储器地址，较低的有效字节存放在较低的存储器地址 (同小)。2) 大端存储器组织是较高的有效字节存放在较低的存储器地址，较低的有效字节存放在较高的存储器地址 (正反大)。

14、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齐方式。数据的自然对齐方式是指：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。

15、ARM 核现在有哪几种？

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

16、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 1) BEQ LOOP 2) BL ROUTE1

17、在 ARM 的立即寻址方式下其立即数如何在指令编码中表示？

答：在 ARM 数据处理指令中，当参与操作的第二操作数为立即数型时，每个立即数都是采用一个 8 位的常数循环右移偶数位而间接得到。其中循环右移的位数由一个 4 位二进制的两倍表示，如果立即数记作 <immediate>，8 位常数记作 immed_8，4 位的循环右移值记作 rotate_imm，有效的立即数是由一个 8 位的立即数循环右移偶数位得到。因此有效立即数 immediate 可以表示成：<immediate>=immed_8 循环右移 (2×rotate_imm)

18、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。

19、指令 LDR R1,=0xffff 是如何将立即数 0xffff 读入到 R1 的？

答：由于需要读取的寄存器的数据超过了 MOV 及 MVN 指令可操作的范围，编译器将该常数放在数据缓冲区中，同时用一条基于 PC 的 LDR 指令读取该常数。

LDR R1, [PC, offset_to_LP00L].

... ..

LTORG ;声明数据缓冲池

LP00L DCD 0xFFFF; 0xFFFF 放在数据缓冲池中。

20、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。

21、如何在汇编语言程序访问 C 语言的全局变量？

答：汇编程序可以通过地址间接访问在 C 语言程序中声明的全局变量。通过使用 IMPORT 关键词引入全局变量，并利用 LDR 和 STR 指令根据全局变量的地址可以访问它们。

对于不同类型的变量，需要采用不同选项的 LDR 和 STR 指令，如下所示：

1) unsigned char—— LDRB/STRB;

2) unsigned short——LDRH/STRH;

3) unsigned int——LDR/STR;

4) char ——LDRSB/STRSB;

5) Short——LDRSH/STRSH

22、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。

23、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。

24、请简述 ARM 处理器的特点，至少说出 5 个以上的特点。（5 分）

答：低功耗；低成本，高性能，RISC 结构；指令定长；支持 Thumb（16 位）/ARM（32 位）双指令集；体积小；

25、请写出 ARM 内核和 ARM SoC 处理器的异同，并举例进行说明。（5 分）

答：ARM 内核是 ARM SoC 处理器中的核心部分，所有 ARM SoC 都采用 ARM 的体系结构和指令集，ARM SoC 主要是在 ARM 内核基础上继承了 Memory Controller, Interrupt Controller, Timer, DMA Controller 以及像 GPIO, USB, IIC, LCD 等外设控制器。

26、ARM 内核有多少个寄存器，请列举出这些寄存器的名字和数量。（5 分）

答：ARM 有 37 个寄存器，（1）未分组寄存器：R0-R7，共 8 个；（2）分组寄存器 R8-R14，其中 FIQ 模式下有单独的一组 R8-R12 共 5 个，另外 6 种模式共用一组 R8-R12，共 5 个，USR 和 SYS 模式共用一组 R13-R14，共 2 个，另外 5 种模式下各有独自的一组 R13-R14，共 10 个；（3）程序计数器 PC 即 R15 寄存器，共 1 个；（4）状态寄存器 CPSR，和 5 个备份状态寄存器 SPSR，共 6 个；ARM 总计 37 个寄存器。

27、ARM 通用寄存器中，有 3 个寄存器有特殊功能和作用，请写出它们的名字和作用。（6 分）

答：R13：SP 栈指针寄存器，用来保存程序执行时的栈指针位置；R14：LR 返回链接寄存器，用来保存程

序执行 BL 指令或模式切换时的返回原程序继续执行的地址；R15：PC 程序计数器，保存程序执行的当前地址。

28、请描述一下 CPSR 寄存器中相关 Bit 的情况和作用。（5 分）

答：条件位（指令进行算术运算后的结果是否有进位，借位等），I 位（IRQ 异常允许位），F 位（FIQ 异常允许位），T 位（ARM/Thumb 工作状态），模式位（处理器工作模式）

29、请写出以下相关 ARM 指令语句的注释：（7 分）

MOV R0, PC 把 PC 的值传送到 R0 寄存器，此时 PC 地址为当前指令位置+8
ADD R0, R1, #1 把 R1 加上 1 的结果给 R0 寄存器
LDR R0, =0x56000010 是伪指令，把 0x56000010 放到 R0 寄存器中，采用文字池技术
LDR PC, [PC, #4] 把当前 PC 值加 4 位置的内容赋值给 PC，实现绝对跳转
MRS R0, CPSR 把 CPSR 的值传送给 R0
BL LOOP 带返回地址的跳转指令，把下一条指令给 LR 后，跳转到 Loop 地址
STMFD R13!, {R0-R12} 压栈指令，把 R0 到 R12 寄存器的值通过 R13 栈指针进行压栈保存，同时这个栈是满递减方式的栈。

30、什么是立即数？请简要描述立即数在使用时有什么注意要点。（2 分）

立即数，主要是指寻址时直接在指令中出现的数，在使用时注意（1）立即数前需要加#（2）ARM 指令只有 32 位长，立即数在指令中占 12 位存储空间，ARM 用这 12 位空间 8 位表示有效数字—基数 B，4 位表示译为的数 M，按照把 B 循环右移 M*2 位，构造成为一个新的 32 位的数，其它位补 0，所以在用立即数时，要注意其是否合法。

31、请问 BL 指令跳转时 LR 寄存器保存的是什么内容？并请简述原因。（3 分）

BL 跳转时，LR 中保存的是执行 BL 跳转指令的下一条指令的地址，考虑流水线的情况，即当前的 PC-4。LR 用来在需要返回程序时从 LR 中还原程序执行的位置继续执行。

32、请描述一下什么是处理器现场，如何进行保存现场？（5 分）

每种工作模式下都包含 R0-R15，CPSR 这 17 个寄存器，程序的执行当前状态就保存在这些寄存器中，称为处理器现场。当发生模式切换时，由于其中的一些寄存器是多种模式下共用的寄存器，为了防止共用处理器寄存器中的值被破坏，所以需要保存原模式下的处理器现场，利用 STM 批量存储指令，把处理器现场对应的寄存器保存到栈上，待还原时再出栈恢复（模式和返回地址）。其中保存现场的工作，硬件完成了 CPSR 模式的保存和 PC 返回地址的保存，其他寄存器的保存工作主要依靠软件压栈完成，其中 LR 因为可能被异常处理程序中的 BL 跳转指令修改，所以一般都需要软件压栈再保存。

33、请说明 SRAM, SDRAM, Nor Flash 和 Nand Flash 的异同，并详细描述各自编程上的注意要点。

答：异同见教材，编程上要注意说明 SRAM 和 NorFlash 无须初始化就可以使用，SDRAM 和 NandFlash 都需要初始化驱动代码才可以工作，NorFlash 的写需要靠芯片手册规定的时序来做，STR 指令不能够直接写入 NorFlash 和 NandFlash，但可以直接写入 SRAM 和 SDRAM。NandFlash 器件是属于 IO 方式进行读，和其他 3 种器件的总线读方式不一样，因此读写 NandFlash 需要相应的驱动。

34、叙述 S3C2410A 存储控制器的特点，并回答每个片选信号对应的地址范围。

S3C2410A 的存储器控制器提供访问外部 ROM、SRAM、SDRAM 存储器所需要的数据信号、地址信号和控制信号，所以 S3C2410A 与外部存储器进行连接时不需要增加额外的器件。存储器控制器有以下的特性：

- 1) 小/大端工作模式(可通过软件选择)。
- 2) 地址空间：共有 8 个 bank，名称分别为 bank7~bank0，每 bank 有 128MB 空间，总共 1G 字节空间。
- 3) 除 bank0 只能是 16/32 位宽之外，其他 bank 都具有可编程的访问大小(可以是 8/16/32 位宽)。
- 4) 总共 8 个存储器 bank：6 个是 ROM、SRAM 等类型存储器 bank。剩下的 2 个可以作为 ROM、SRAM、SDRAM 等存储器 bank。
- 5) 7 个固定的存储器 bank 的起始地址；最后一个 bank 的起始地址是可调整的。
- 6) 最后两个 bank 的大小是可编程的。
- 7) 所有存 bank 的访问周期都是可编程的；总线访问周期可以通过插入外部 wait 来延长。
- 8) 支持 SDRAM 的自刷新和掉电模式。

表7.1 片选信号对应的地址范围

片选信号名称	地址范围	地址空间大小
nGCS0	0X0000-0000~0X07FF-FFFF	128M字节
nGCS1	0X0800-0000~0X0FFF-FFFF	128M字节
nGCS2	0X1000-0000~0X17FF-FFFF	128M字节
nGCS3	0X1800-0000~0X1FFF-FFFF	128M字节
nGCS4	0X2000-0000~0X27FF-FFFF	128M字节
nGCS5	0X2800-0000~0X2FFF-FFFF	128M字节
nGCS6	0X3000-0000~0X37FF-FFFF	128M字节
nGCS7	0X3800-0000~0X3FFF-FFFF (起始地址可变)	128M字节

四、指令及程序设计题

1、写一条 ARM 指令，完成操作 $r1 = r2 * 4$ 。

```
MOV R1, R2, LSL #2
```

2、初始值R2=5, R3=4, R4=3, 执行指令SUBS R2, R3, R4, LSR #2后, 寄存器R2, R3的值分别是多少?

R2=4, R3=4, R4=3

3、编写程序，将存储器从 0x400000 开始的 200 个字节的数据，传送到 0x400800 开始的区域。

```
解:      MOV      R0, #0x400000
          LDR      R1, =0x400800
          MOV      R7, #200
LP        LDRB     R2, [R0], #1
          STRB     R2, [R1], #1
          SUBS     R7, R7, #1
          BNE     LP
HERE
          B        HERE
```

4、有如下程序段，画出程序流程图，并回答执行程序以后 R0 的值是多少。

```
num      EQU      2
start
          MOV      r0, #1
          MOV      r1, #5
          MOV      r2, #2
          BL       func
stop
          B        stop
func
          CMP      r0, #num
          MOVHS    pc, lr
          ADR      r3, JumpTable
          LDR      pc, [r3, r0, LSL#2]
```


JumpTable

DCD DoA

DCD DoS

DoA

ADD r0, r1, r2

MOV pc, lr

DoS

SUB r0, r1, r2

MOV pc, lr

END

答：当 R0 为 0 执行 DoA，R0 为 1 时执行 DoS，R0 大于 2 直接返回。流程图略。R0=3。

5、有程序段如下，实现下面流程图，试补充编写 ARM 汇编代码。

程序段：

MOV r0, #15

MOV r1, #9

start

:补充代码

stop

B stop

END

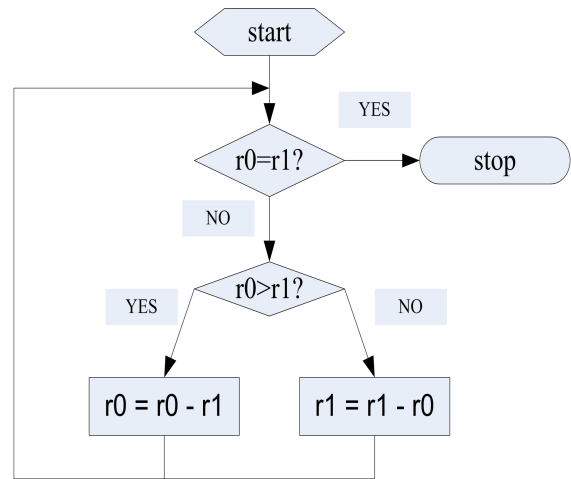
答：

CMP r0, r1

SUBLT r1,r1,r0

SUBGT r0,r0,r1

BNE start



6、编写程序，比较存储器中 0x400000 和 0x400004 两无符号字数据的大小，并且将比较结果存于 0x400008 的字中，若两数相等其结果记为 0，若前者大于后者其结果记为 1，若前者小于后者其结果记为-1。

解：MOV R0, #0x400000

LDR R1, [R0] ; 取第 1 个数

LDR R2, [R0, #4] ; 取第 2 个数

CMP R1, R2 ; 两个数相比较

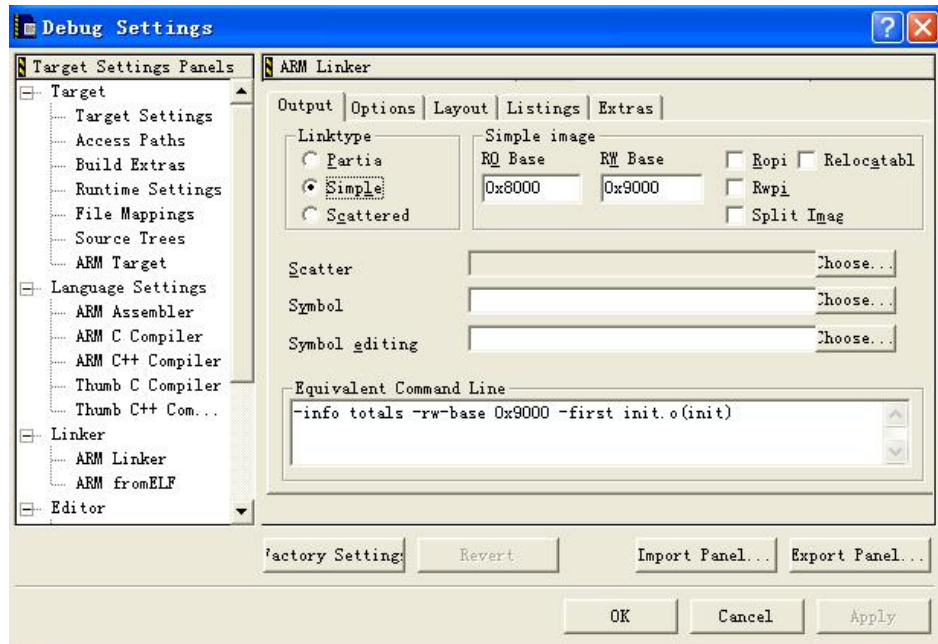
MOVHI R1, #1 ; R1 大

MOVLO R1, # -1 ; R1 小

MOVEQ R1, #0 ; 两个数相等

STR R1, [R0, #8]

7、如下在 ADS 中对工程的链接器（ARM Linker）进行了配置



并编写了如下的汇编程序：

```

N      EQU 5

      AREA Test1, CODE, READONLY
      ENTRY
      MOV R0, #0
      MOV R1, #1
      REPEAT ADD R0, R0, R1
      ADD R1, R1, #1
      CMP R1, #N
      BLE REPEAT
      LDR R2, =RESULT
      STR R0, [R2]
      HERE B HERE

      AREA DataSpace, DATA, READWRITE
      RESULT DCD 0
      END
  
```

问：现执行上述程序，请列出 R0,R1,R2 的最终值，并告知整个程序的功能。（5分）

答：R0=1 R1=2 R2=1

功能：对 R0 和 R1 分别赋值 0，1，在 R1 为 5 的条件下循环给 R0 赋值 R0 和 R1 之和，给 R1 加 1，取得 RESULT 的地址赋值于 R2，将 R0 的值保存在 RESULT 中

8、阅读如下程序

test.s:

```

      area test2,code,readonly
      export s_program
s_program    sub r0,r0,r1
  
```

```
mov pc,lr
end
```

test.c:

```
#include <stdio.h>
extern int s_program (int a,int b);
int main()
{
    int num;
    num= s_program(54,23);
    printf("%d",num);
    return 0;
}
```

请列出上述程序执行后 R0, R1 的值, 并说明上述程序的作用。(5 分)

答: R0=31 R1=23

功能: test.c 调用 s_program 函数, 将 54 赋值 R0, 23 赋值 R1, 两者相减后, 赋值给 R0, R0 作为返回值赋值给 num, 然后输出 num。

9、现有如下 C 语言定义的函数 mypro

```
int mypro (int a, int b)
{
    a=a*b;
    return (a);
}
```

请补充完成调用上述函数的汇编程序, 实现 11 与 22 相乘, 并要求将结果保存在地址为 0x40001000 处: (10 分)

```
area test3, code, readonly
import mypro
entry
code32

start
    mov r0,#11                (1)
    mov r1,#22                (2)
    BL mypro                  (3)
    Ldr r3,=0x40001000        (4)
    Str r0,[r3]               (5)
END
```

10、写一段 ARM 汇编程序: 循环累加队列 myarray 中的所有元素, 直到碰上零值元素, 结果放在 r4 中。程序框架如下, 补充代码完成上述功能。(8 分)

```
AREA total, CODE READONLY
ENTRY
```

```

start
    MOV     r4, #0
    ADR     r0, myarray
    ; 在此补充代码
loop
    LDR r1, [r0], #4
    ADD     r4, r4, r1
    CMP     r1, #0
BNE loop

loop
    LDR r1, [r0], #4
    ADD     r4, r4, r1
    CMP     r1, #0
    BNE     loop

```

11、编写一简单 ARM 汇编程序，实现 $1+2+\dots+100$ 的运算。

解：

```

    MOV R2, #100
    MOV R1, #0
LOOP
    ADD R1, R1, R2      ; R1 中为累加和
    SUBS R2, R2, #1     ; R2 控制循环
    BNE LOOP
    B    $

```

12、用 ARM 汇编语言设计程序实现求 $20!$ (20 的阶乘)，并将其 64 位结果放在 $[R9: R8]$ 中 ($R9$ 中存放高 32 位)。

```

AREA Fctrl, CODE, READONLY
ENTRY
CODE32
STARY
    MOV R8, #20
    MOV R9, #0
    SUB R0, R8, #1
LOOP
    MOV R1, R9
    UMULL R8, R9, R0, R8
    MLA R9, R1, R0, R9
    SUBS R0, R0, #1
    BNE LOOP
STOP
    B STOP
END

```

13、先对内存地址 $0x3000$ 开始的 100 个内存单元填入 $0x10000001 \sim 0x10000064$ 字数据，然后将每个字单元进行 64 位累加结果保存于 $[R9: R8]$ ($R9$ 中存放高 32 位)

```

AREA Fctrl, CODE, READONLY
ENTRY
CODE32
STARTY

```

```

    MOV    R0, #3000
    MOV    R1, #0x10000001
    MOV    R2, #0x100

```

```

LOOP_1
    STR    R1, [R0], #4
    ADD    R1, R1, #1
    SUBS   R2, R2, #1
    BNE    LOOP_1

```

```

    MOV    R0, #0x3000
    MOV    R2, #0x100
    MOV    R9, #0
    MOV    R8, #0

```

```

LOOP_2
    LDR    R1, [R0], #4
    ADDS   R8, R1, R8

    ADC    R9, R9, #0
    SUBS   R2, R2, #1
    BNE    LOOP_2

```

```

STOP
    B     STOP
END

```

- 14、已知 32 位变量 X、Y 存放在存储器的地址 0x90010、0x90014，要求编写完整 ARM 汇编程序实现 $Z=X+Y$ ，其中 Z 的值存放在 0x90018 中。

```

        AREA EX4_41, CODE, READONLY
        ENTRY
        CODE32
START    LDR R0, =0x90010      ;变量 X 的地址送入 R0
        LDR R1, [R0], #4      ;变量 X 的值读入 R1
        LDR R2, [R0], #4      ;变量 Y 的值读入 R2
        ADD R1, R1, R2        ;X+Y 结果存入 R1
        STR R1, [R0]          ;结果存入 z 中
        B START
        END

```

- 15、已知 32 位有符号数 X 存放在存储器的地址 0x90010 中，要求编写完整 ARM 汇编程序实现：

$$Y = \begin{cases} X & (X \geq 0) \\ -X & (X < 0) \end{cases}$$

```

AREA EX4_42, CODE, READONLY
    ENTRY
    CODE32
START  LDR R1, =0x90010      ;加载变量 X 的地址->R1
        MOV R0, #0          ;0->R0
        LDR R2, [R1]        ;将 X 的值加载到 R2
        CMP R2, #0          ;X 与 0 比较, 影响标志位
        SUBLT R2, R0, R2     ;X<0 执行该语句, 提到-X
        STR R2, [R1]        ;保存结果
        B START
    END

```

16、使用ARM汇编指令实现if条件执行

17、. 使用ARM汇编指令实现for循环结构

18、. 使用ARM汇编指令实现while循环结构

19、. 使用ARM汇编指令实现do...while循环结构

20、. 使用ARM汇编指令实现switch开关结构 答案见实验三的源程序

AREA	test4, CODE, READONLY	ADD	R0, R0, #1
ENTRY		ADD	R2, R2, #1
CODE32			
		B	FOR_LP
START			
;if(x>y) z = 100	FOR_END		
;else z = 50	NOP		
;r0 = x, r1 = y, r2 = z			
MOV R0, #56	;while(x <= y)		
MOV R1, #233	;{		
CMP R0, R1	; x *= 2;		
MOVHI R2, #100	;}		
MOVLS R2, #50	;r0 = x, r1 = y		
	MOV R0, #1		
	MOV R1, #20		
;for(i = 0; i < 10; i++)	B	WHILE_LP2	
;{			
; x++;	WHILE_LP1		
;}	MOV R0, R0, LSL #1 ;x		
;r0 = x, r2 = i	*= 2;		
MOV R0, #0			
MOV R2, #0	WHILE_LP2		
FOR_LP	CMP R0, R1 ;x		
CMP R2, #10	<= Y ?		
BHS FOR_END	BLS WHILE_LP1		

```

WHILE_END
    NOP

    ;do
    ;{
    ;    x--;
    ;}while(x > 0);
    ; r0 = x;

    MOV        R0,#5
DOWHILE_LP1
    ADD        R0,R0,#-1    ;x--

DOWHILE_LP2
    MOVS       R0,R0
    BNE        DOWHILE_LP1
;R0 != 0 CONTINUE

DOWHILE_END
    NOP

    ;switche(key & 0x0f)
    ;{
    ;    case 0:
    ;    case 2:
    ;    case 3: x = key + y;
    ;    break;
    ;    case 5:    x = key - y;
    ;    break;
    ;    case 7:    x = key * y;
    ;    break;
    ;    default:x = 163;
    ;    break;
    ;}
    ; r0 = x,r1 = y r2 = key

    MOV        R1,#3
    MOV        R2,#2

SWITCH
    AND        R2,R2,#0x0F

CASE_0  CMP        R2,#0
CASE_2  CMP        R2,#2

CASE_3  CMP        R2,#3
        BNE        CASE_5
        ADD        R0,R2,R1
        B          SWITCH_END

CASE_5  CMP        R2,#5
        BNE        CASE_7
        SUB        R0,R2,R1
        B          SWITCH_END

CASE_7  CMP        R2,#7
        BNE        DEFAULT
        MUL        R0,R2,R1
        B          SWITCH_END

DEFAULT
        MOV        R0,#163

SWITCH_END
    NOP

HALT    B          HALT

END

```