

ARM 嵌入式体系结构与接口技术（Cortex-A53 刘洪涛版）（微课版） - 习题答案

第一章 嵌入式系统基础知识

1. 嵌入式系统的定义？

答：嵌入式系统是以应用为中心,以现代计算机技术为基础,能够根据用户需求(功能、可靠性、成本、体积、功耗、环境等灵活裁剪软硬件模块的专用计算机系统。

2. 嵌入式系统的特点？

答：软硬件可裁剪，对功能、可靠性、成本、体积、功耗要求严格，代码短小精悍，可固化，实时性，若交互性，需要专门的开发环境和开发工具。

3. 嵌入式系统的主要组成？

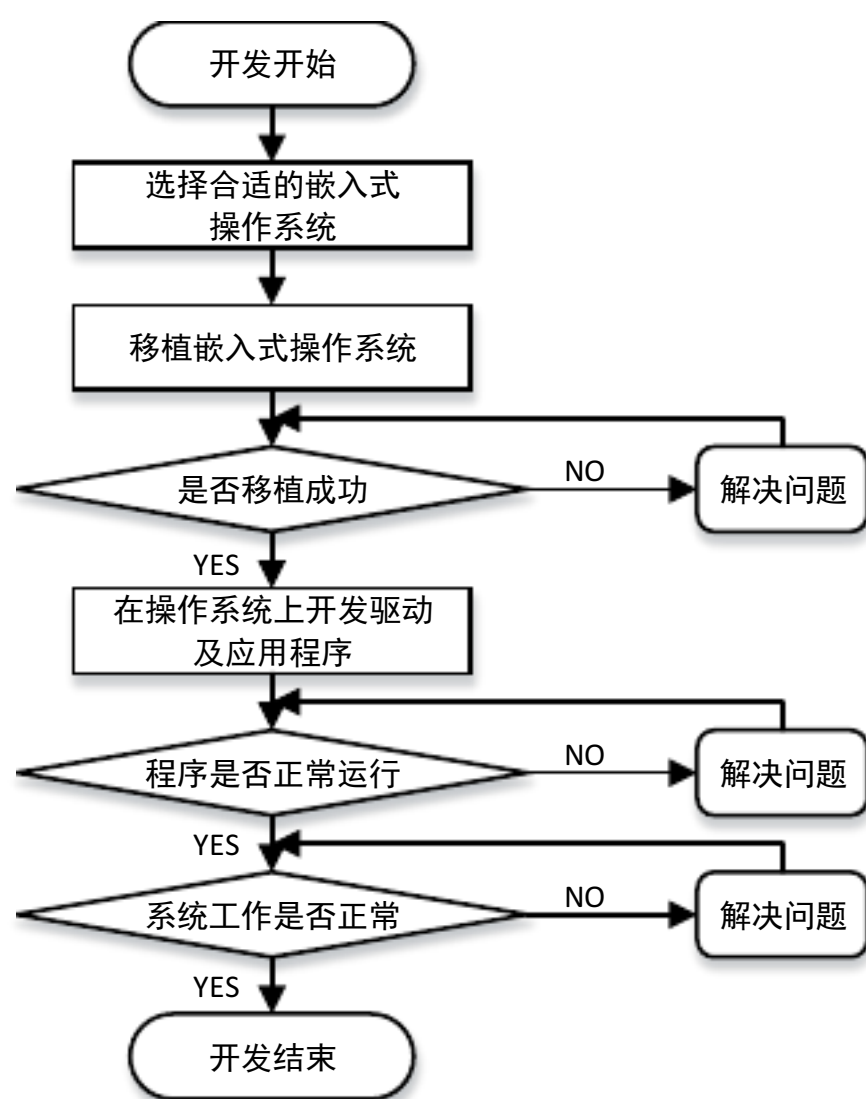
答：嵌入式系统由硬件部分和软件部分组成。硬件是基础，软件是灵魂与核心。“软硬兼施”才能综合提高嵌入式系统的性能。具体来说，一般嵌入式系统可以分为嵌入式处理器（CPU）、外围设备、嵌入式操作系统（可选）以及应用软件等四个部分。

4. 市场上主流的嵌入式系统有哪些？

答：VxWorks, QNX, Palm OS, WinCE, Linux, Unix, Lynx, Nucleux, Hopen, Delta OS。

5. 简述嵌入式系统的开发流程？

答：



第二章 ARM 技术概述

1. ARM 的三种含义？

答：它是一个公司的名称，是一类微处理器的统称，还是一种技术的名称。

2. RISC 和 CISC 的区别？

答：精简指令集计算机（Reduced Instruction Set Computer, RISC），RISC 把着眼点放在如何使计算机

的结构更加简单和如何使计算机的处理速度更加快速上。RISC 选取了使用频率最高的简单指令，抛弃复杂指令，固定指令长度，减少指令格式和寻址方式，不用或少用微码控制。这些特点使得 RISC 非常适合嵌入式处理器。

复杂指令集计算机（Complex Instruction Set Computing, CISC），CISC 更侧重于硬件执行指令的功能性，使 CISC 指令及处理器的硬件结构变得更复杂。这些会导致成本、芯片体积的增加，影响其在嵌入式产品中的应用。

3. ARM 处理器的特点？

答：

- a) 体积小、低功耗、低成本、高性能；
- b) 支持 Thumb（16 位）/ARM（32 位）双指令集，能很好的兼容 8 位/16 位器件；
- c) 大量使用寄存器，指令执行速度更快；
- d) 大多数数据操作都在寄存器中完成；
- e) 寻址方式灵活简单，执行效率高；
- f) 指令长度固定。

第三章 Cortex-A53 编程模型

1. ARM-v8 架构支持的基本数据类型？

答：

- A. Byte：字节，8bit。
- B. Halfword：半字，16bit（半字必须与 2 字节边界对齐）。
- C. Word：字，32bit（字必须与 4 字节边界对齐）。
- D. DoubleWord：双字，64bit。
- E. QueaWord：四字，128bit。

2. 简述 ARM-v8 架构支持几种模式？

答：User mode, FIQ mode, IRQ mode, SVC mode, Abtrt mode, Undefined mod, System mode, Monitor mode, Hypervisor mode。

3. X30 寄存器的作用？

答：x30 别名 lr，链接寄存器用于保存返回地址。

4. 简述存储器的大端对齐和小端对齐？

答：

大端对齐：低地址存放数据的高有效位，高地址存放数据的低有效位；

小端对齐：低地址存放数据的低有效位，高地址存放数据的高有效位。

5. 简述指令的 3 级流水线？

答：到 ARM7 为止的 ARM 处理器使用简单的 3 级流水线，它包括下列流水线级。

- ① 取指令（fetch）：从寄存器装载一条指令。
- ② 译码（decode）：识别被执行的指令，并为下一个周期准备数据通路的控制信号。在这一级，指令占有译码逻辑，不占用数据通路。
- ③ 执行（excute）：处理指令并将结果写回寄存器。

如图 2-4 所示为 3 级流水线指令的执行过程。

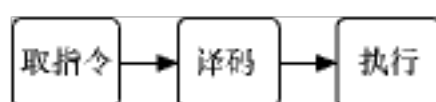


图 3-3 3 级流水线

当处理器执行简单的数据处理指令时，流水线使得平均每个时钟周期能完成 1 条指令。但 1 条指令需

要 3 个时钟周期来完成，因此，有 3 个时钟周期的延时（latency），但吞吐率（throughput）是每个周期 1 条指令。

6. PSTATE.NZCV 位分别是什么作用？

答：

① N

本位设置成当前指令运行结果的 bit[31]的值。当两个由补码表示的有符号整数运算时，N=1 表示运算的结果为负数，N=0 表示结果为正数或零。

② Z

Z=1 表示运算的结果为零，Z=0 表示运算的结果不为零。

③ C

下面分 4 种情况讨论 C 的设置方法。

在加法指令中（包括比较指令 CMN），当结果产生了进位，则 C=1，表示无符号数运算发生上溢出；其他情况下 C=0。

在减法指令中（包括比较指令 CMP），当运算中发生错位（即无符号数运算发生下溢出），则 C=0；其他情况下 C=1。

对于在操作数中包含移位操作的运算指令（非加/减法指令），C 被设置成被移位寄存器最后移出去的位。

对于其他非加/减法运算指令，C 的值通常不受影响。

④ V

下面分两种情况讨论 V 的设置方法。

对于加/减运算指令，当操作数和运算结果都是以二进制的补码表示的带符号的数时，且运算结果超出了有符号运算的范围是溢出。V=1 表示符号位溢出。

对于非加/减法指令，通常不改变标志位 V 的值（具体可参照 ARM 指令手册）。

尽管以上 C 和 V 的定义看起来颇为复杂，但使用时在大多数情况下用一个简单的条件测试指令即可，不需要程序员计算出条件码的精确值即可得到需要的结果。

第四章 ARM 开发及环境搭建

1. 熟悉 eclipse 开发环境

答：参考 4.2 章节

第五章 ARM 处理器的指令系统

1. 用 ARM 汇编实现下面列出的操作。

X0=15

答：MOV X0, #15

X0=X1/16（有符号数）

答：MOV X2, #16

SDIV X0, X1, X2

X1=X2*3

答：MOV X0, #3

MUL X1, X2, X0

X0=-X0

答：MOV X1, #-1
MOV X2, R0
MUL X0,X1,X2

2 . BIC 指令的作用是什么？

答：将一个寄存器中的某一位或者某几位清零。

3 . B、BL 指令的区别有哪些？

答：B 与 BL 都可以使程序跳转到指定的地址（label）处执行程序

B 跳转指令 pc←label

BL 带返回的连接跳转 pc←label，同时将 BL 下一条指令地址保存到 LR

4 . 写一个程序，如果 X0 的值大于 0x50，则将 X1 的值减去 0x10，并把结果送给 X0？

答：

```
CMP X0, #0x50
BL.GT func_sub
...
```

```
func_sub:
    SUB X0,X1,#0x10
    RET
```

5 . 编写一段 ARM 汇编程序，实现数据块复制，将 X0 指向的 8 个字的连续数据保存到 X1 指向的一段连续的内存单元中？

答：

```
LDP X2, X3, [X0]
LDP X4, X5, [X0, #8]
STP X2, X3, [X1]
STP x4, X5, [X1, #4]
```

第六章 ARM-v8 汇编语言程序设计

1 . 在 ARM 汇编中如何定义一个全局的数字变量？

答：使用.global 伪操作

语法格式：.global 全局变量名 并且被初始化为 0

2 . AAPCS64 中规定的 ARM 寄存器的使用规则是什么？

答：前 8 个寄存器 X0-X7 用于向子程序传递参数值并从函数返回结果值。它们也可以用来保存程序中的中间值(但是，通常只在子程序调用之间使用)。

寄存器 X16 (IP0) 和 X17 (IP1) 可以被链接寄存器用作例程和它调用的任何子程序之间的擦除寄存器。它们还可以在例程中用于保存子程序调用之间的中间值。

子调用调用必须保留寄存器 X19-X29 和 SP 的内容。存储在 X19-X29 中的每个值的所有 64 位都必须保留，即使在使用 ILP32 数据模型时也是如此。

3 . 什么是内联汇编？

答：在 C 代码中插入汇编语言的方法称为内联汇编。

4 . 汇编代码中如何调用 C 代码中定义的函数？

答：（1）调用的 C 带有参数，需要使用 R0~R3 寄存器进行参数的传递，如果参数多于四个

多出来的部分使用堆栈传递。

(2) 调用 C 程序时，使用 BL 跳转指令，格式为： BL 函数名

(3) 如果 C 程序带有返回值，使用寄存器 R0 传递。

第七章 ARMv8 异常处理及编程

1. ARMv8 架构异常的种类？

答：FIQ, IRQ, Undef, SVC, Monitor, Hyp, Abort。

2. 什么类型的异常优先级最高？

答：Reset

3. 什么指令可以放在异常向量表中？

答：跳转指令

第八章 通用 GPIO 接口

1. 什么是 GPIO？

答：GPIO 的英文全称为 General Purpose Input Output ports，也就是通用输入输出接口。

2. S5P6818 芯片将 GPIO 引脚分成几组？分别是什么？

答：总共分成 5 组，每组 32 个引脚，总共 160 个引脚，分别为 GPIOA, GPIOB, GPIOC, GPIOD, GPIOE。

3. 编程实现利用 S5P6818 的 GPIO 控制 RGB 三色灯循环闪烁？

答：具体代码在 part08.3 文件夹中 led.c, led.h, main.c 文件中。

第九章 ARM 外部中断

1. S5P6818 中断源有哪几种？

答：软件产生中断（SGI），专有外设中断（PPI），共享外设中断（SPI）。

2. 简述 S5P6818 中断机制？

答：ARM 内核只有两个外部中断输入信号 nFIQ 和 nIRQ。但对于一个系统来说，中断源可能多达几十个。为此，在系统集成时，一般都会有一个中断控制器来处理异常信号。这时候用户程序可能存在多个 IRQ/FIQ 的中断处理函数。为了使从向量表开始的跳转始终能找到正确的处理函数入口，需要设置处理机制和方法。不同的中断控制器处理方法不同。

4. 编写完成 VOL+和 VOL-按键中断实验？

答：具体代码在 part09.3 文件夹中 key_interrupt.c, key_interrupt.h, s5p6818-irq.c, main.c 文件中。

第十章 UART 串行通信接口

1. 串行通信与并行通信的概念是什么？

答：串行通信是指计算机与 I/O 设备之间数据传输的各位是按顺序依次一位接一位进行传送。并行通信是指计算机与 I/O 设备之间通过多条传输线交换数据，数据的各位同时进行传送。

2. 同步通信与异步通信的概念及区别是什么？

答：概念：

异步通信，是指数据传送以字符为单位，字符与字符间的传送是完全异步的，位与位之间的传送基本上是同步的。

同步通信,是指数据传送是以数据块(一组字符)为单位,字符与字符之间、字符内部的位与位之间都同步。

区别:

异步通信:

- (1) 以字符为单位传送信息。
- (2) 相邻两字符间的间隔是任意长。
- (3) 因为一个字符中的波特位长度有限, 所以需要的接收时钟和发送时钟只要相近就可以。
- (4) 异步方式特点就是: 字符间异步, 字符内部各位同步。

同步通信:

- (1) 以数据块为单位传送信息。
- (2) 在一个数据块(信息帧)内, 字符与字符间无间隔。
- (3) 因为一次传输的数据块中包含的数据较多, 所以接收时钟与发送时钟严格同步, 通常要有同步时钟。

3. RS-232C 串口通信接口规范是什么?

答: 参照本章 10.1.2 小结

4. 在 S5P6818 串口控制器中, 哪个寄存器用来设置串口波特率?

答: UBRDIVn 和 UFRACVALn

第十一章 定时器和 PWM

1. PWM 输出波形的特点是什么?

答: 若令频率不变, 直接改变脉冲的宽度, 亦即控制开关元件的导通时间; 比如现在是高电平导通, 那么高电平时间越长, 低电平时间越短, 导通时间就长; 否则就越短。

2. 编程实现输出占空比为 70%、波形周期为 10ms 的 PWM 波形?

答: 具体代码在 part11.2 文件夹中 pwm.h, pwm.c, main.c 文件中。

3. 使用 PWM0 定时器中断实现 1s 的精确定时, 控制 led 灯每隔 1s 亮灭一次?

答: 具体代码在 part11.3 文件夹中 pwm-timer.h, pwm-timer.c, s5p6818-irq.c, main.c 文件中。

4. 在控制系统中为何要加入看门狗功能?

答: 检测程序的正常运行, 当程序跑死后, 看门狗产生一个强制系统复位。这样可以使程序重新运行, 减小程序跑死的危害。

5. 编程实现 1s 内不对看门狗实现喂狗操作, 看门狗会自动复位?

答: 具体代码在 part11.5 文件夹中 wdt.h, wdt.c, main.c 文件中。

第十二章 A/D 转换器

1. A/D 转换器选型时需要考虑哪些指标?

答: 分辨率、转换速率、量化误差、偏移误差、满刻度误差、线性度等。

2. 根据 A/D 的基本原理, 可以将 A/D 控制器分为哪些种类?

答: 积分型 A/D 转换器、逐次逼近性 A/D、并行比较/串行比较 A/D、电容阵列逐次比较型、压频变换型。

3. 编程实现 ADC 中断处理程序测量模拟电压值?

答: 具体代码在 part12.3 文件夹中 adc.h, adc.c, s5p6818-irq.c, main.c 文件中。

第十三章 SPI 总线接口

1 . SPI 总线和 I2C 总线的区别是什么？

答：从名称上：

SPI(Serial Peripheral Interface：串行外设接口);

I2C(INTER IC BUS：意为 IC 之间总线)

从硬件连接上：

SPI 总线是四线制（MISO，MOSI，CLK，CS）如果主机和从机是一对一的可以省略 CS 片选线

IIC 总线是两线制（SDA，CLK）

从通信方式上：

I2C 是半双工，SPI 是全双工

从传输速度上：

SPI 的传输速度（8Mbit/s）高于 IIC 的传输速度（最快 3.4Mbit/s）

2 . 编写程序先擦除 M25P32 闪存中的数据，在写入数据？

答：M25P32 闪存操作数据的函数在 part13.2 文件夹的 source/spi.c 文件中，具体代码如下

```
void erase_sector(void)
```

```
{  
    soft_reset();  
    slave_enable();  
    send_byte(0x06);  
    slave_disable();  
  
    soft_reset();  
    slave_enable();  
    send_byte(0xD8);  
    send_byte(0x00);  
    send_byte(0x00);  
    send_byte(0xF0);  
    slave_disable();  
  
}
```

更加详细的源码可以查看 part13.2 文件夹中的 spi.c，spi.h，main.c 文件。

第十四章 I2C 总线接口

1 . 请画出 I2C 总线的时序图？

答：

主机发送起始信号启用总线

主机发送一字节数据指明从机地址与下一字节传送方向

对应的从机应答

发送器发送一字节数据

接收方应答

... ..

数据传送完成后，主机发送停止信号释放总线使用权

更多的时序可以操作 14.1.2 章节。

2. 根据 MMA8451 芯片手册，编写实现连续读时序和连续写时序？

答：略

3. 请简述 I2C 总线的特点和缺点？

答：

特点：

布线简单：一条串行数据线 SDA、一条串行时钟线 SCL。

每个连接到总线的器件都可以通过唯一的地址。

真正的多主机总线，支持冲突检测和仲裁，防止数据被破坏。

串行的 8 位双向数据传输位。

速率在标准模式下可达 100Kbit/s、快速模式下可达 400Kbit/s、高速模式下可达 3.4Mbit/s。

片上的滤波器可以滤去总线数据线上的毛刺波保证数据完整。

连接到相同总线的 IC 数量只受到总线的最大电容 400pF 限制。

缺点：半双工、传输速度低

第十五章 温度监控系统综合案例

1. 在项目中添加按键驱动代码，可以通过按键调整温度报警的范围？

答：具体代码参考 part15.1 工程文件。

注：VOL+按键按下时，进入报警温度最大值的设置，VOL+按键每按一次最大报警温度加 1，VOL-按键每按下一次最大报警温度减 1。

VOL+按键按下时，进入报警温度最低值的设置，VOL+按键每按一次最低报警温度加 1，VOL-按键每按下一次最低报警温度减 1。

超过 5s 没有任何按下时，自动退出温度报警值范围设置的功能。