

一、 简答题 (6 题*8)

1、 简述 Arm 指令集 6 种移位操作各自的作用

- LSL: 逻辑左移, 空出的最低有效位用 0 填充。
- LSR: 逻辑右移, 空出的最高有效位用 0 填充。
- ASL: 算术左移, 由于左移空出的有效位用 0 填充, 因此它与 LSL 同义。
- ASR: 算术右移, 算术移位的对象是带符号数, 移位过程中必须保持操作数的符号不变。如果源操作数是正数, 空出的最高有效位用 0 填充, 如果是负数用 1 填充。
- ROR: 循环右移, 移出的字的最低有效位依次填入空出的最高有效位。
- RRX: 带扩展的循环右移。将寄存器的内容循环右移 1 位, 空位用原来 C 标志位填充。

2、 简述 S5PV210 中 GPIO 的作用

GPIO (General-Purpose Input/Output Ports) 全称是通用编程 I/O 端口。它们是 CPU 的引脚, 可以通过它们向外输出高低电平, 或者读入引脚的状态, 这里的状态也是通过高电平或低电平来反应的, 所以 GPIO 接口技术可以说是 CPU 众多接口技术中最为简单、常用的一种。

3、 请说出嵌入式系统有哪 2 种状态寄存器, 它们各自有什么作用作用; 请回答现在市面上有哪些主流的 ARM 处理器系列

a)

当前程序状态寄存器 CPSR (Current Program Status Register) 和备份的程序状态寄存器 SPSR (Saved Program Status Register) ;

b)

保存最近执行的算术或逻辑运算的信息;

控制中断的允许或禁止;

设置处理器工作模式。

c) (本答案由 deepseek 给出, 请谨慎参考)

- Cortex-A 系列
- Cortex-R 系列
- Cortex-M 系列
- Neoverse 系列
- Ethos 系列

4、 简述嵌入式操作系统有哪些作用; 嵌入式 linux 操作系统有哪些特点

- 补平硬件差异的界面或是说隐藏硬件, 让应用程序可以在上面运行。

- 程序设计人员无须考虑到不同硬件所造成的差异，可专注于所擅长领域的开发。
- Linux 是一个内核运行在单独的内存地址空间的**单内核**，但是汲取了**微内核的精华**如模块化设计、抢占式内核、支持内核线程以及动态装载内核模块等特点。

5、Arm-Linux 进程调度依据分为哪几个部分

调度程序运行时，要在所有可运行的进程中选择最值得运行的进程。选择进程的依据主要有：

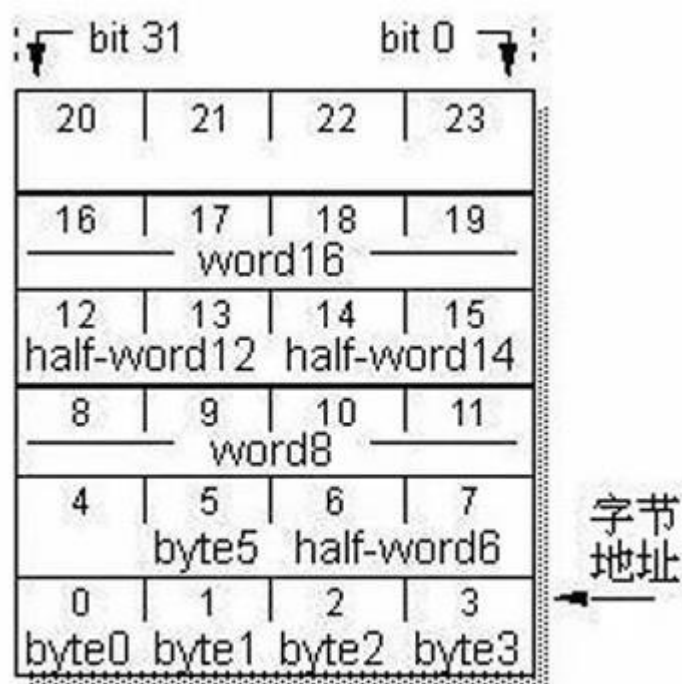
进程的调度策略（policy）

静态优先级（priority）

动态优先级（counter）

实时优先级（rt-priority）

6、简述大端存储器组织是什么结构



(b)大端存储器组织

二、 看语句写作用（2 题*5）（本题答案由 deepseek 给出，请谨慎参考）

1、MRS R1, CPSR

将 当前程序状态寄存器（CPSR）的值读取到通用寄存器 R1 中。

2、ORR R1, [R1,#1]

将 R1 的值与立即数 1 进行按位或操作，结果存回 R1。

3、MSR CPSR,R1

将 R1 的值写入 当前程序状态寄存器 (CPSR)。

4、BLX FUNC1

带链接和状态切换的分支跳转，跳转到函数 FUNC1 的地址，并保存返回地址到 LR。

5、STRH R4, [R1,R2]!

将 R4 的低 16 位（半字）存储到内存地址 [R1 + R2]，并更新基址寄存器 $R1 = R1 + R2$ 。

三、 程序题 (2*5+12)

1、给你一段冒泡排序程序挖空，让你填 5 个空

题目场景如下：无符号数据字块存储在 0x400004，无符号数据字块数目字存储在 0x400000（别问我为什么要用这么复杂的表述方法，我当初看了半天才看懂这是讲的什么，你们也必须感受我的痛苦！/(T o T)/~~）

代码如下：

```
AREA SORT, CODE, READONLY
```

```
ENTRY
```

```
START
```

```
    MOV    R1, #0x400000
```

```
LP
```

```
    SUBS   R1, R1, #1
```

```
    BEQ    EXIT
```

```
    MOV    R7, R1
```

```
    LDR    R0, = (1)
```

```
LP1
```

```
    LDR    R2, [R0], #4
```

```
    LDR    R3, [R0]
```

```
    CMP    R2, R3
```

```
    (2)
```

```
    (3)
```

```
    SUBS   R7, R7, #1
```

```
    BNE    (4)
```

B (5)

EXIT

END

(1) 0x400004

(2) STRLO R3, [R0, # -4]

(3) STRLO R2, [R0]

(4) LP1

(5) LP

2. 给定一个初始地址为 0x400000 的有 100 个单元的**有符号**字符串，要求将字符串内部所有的**所有大写字母**转为小写字母，其他字符不变；(要求使用汇编语言)

(以下答案由 deepseek 给出，请谨慎参考)

AREA FUNC1, CODE, READONLY

ENTRY

start

LDR R0, =0x400000 ; 加载字符串起始地址
MOV R1, #100 ; 初始化循环计数器

; 通过字符字面量获取 ASCII 值

LDR R2, ='A' ; R2 存储'A'的 ASCII 值
LDR R3, ='Z' ; R3 存储'Z'的 ASCII 值
MOV R4, ='a' ; R4 暂时存储'a'的 ASCII 值
SUB R5, R4, R2 ; R5='a'-'A',大小写转换差值

loop

LDRB R4, [R0] ; 加载当前字符

; 判断是否大写

CMP R4, R2 ; 比较当前字符与'A'
BLT not_upper ; 小于'A'则跳过
CMP R4, R3 ; 比较当前字符与'Z'
BGT not_upper ; 大于'Z'则跳过

; 执行大小写转换

ADD R4, R4, R5 ; 大写转小写 (加 0x20)
STRB R4, [R0] ; 存回原内存地址

not_upper

ADD R0, R0, #1 ; 指针移动到下一个字符
SUBS R1, R1, #1 ; 计数器递减并更新标志
BNE loop ; 循环直到处理完所有字符

END

EXIT

四、 回答题 (4 题*5)

1、 简述嵌入式系统硬件基本结构；简述嵌入式系统软件的基本结构

- 以嵌入式处理器为中心，配置存储器、I/O 设备、通信模块以及电源等必要的辅助接口组成。

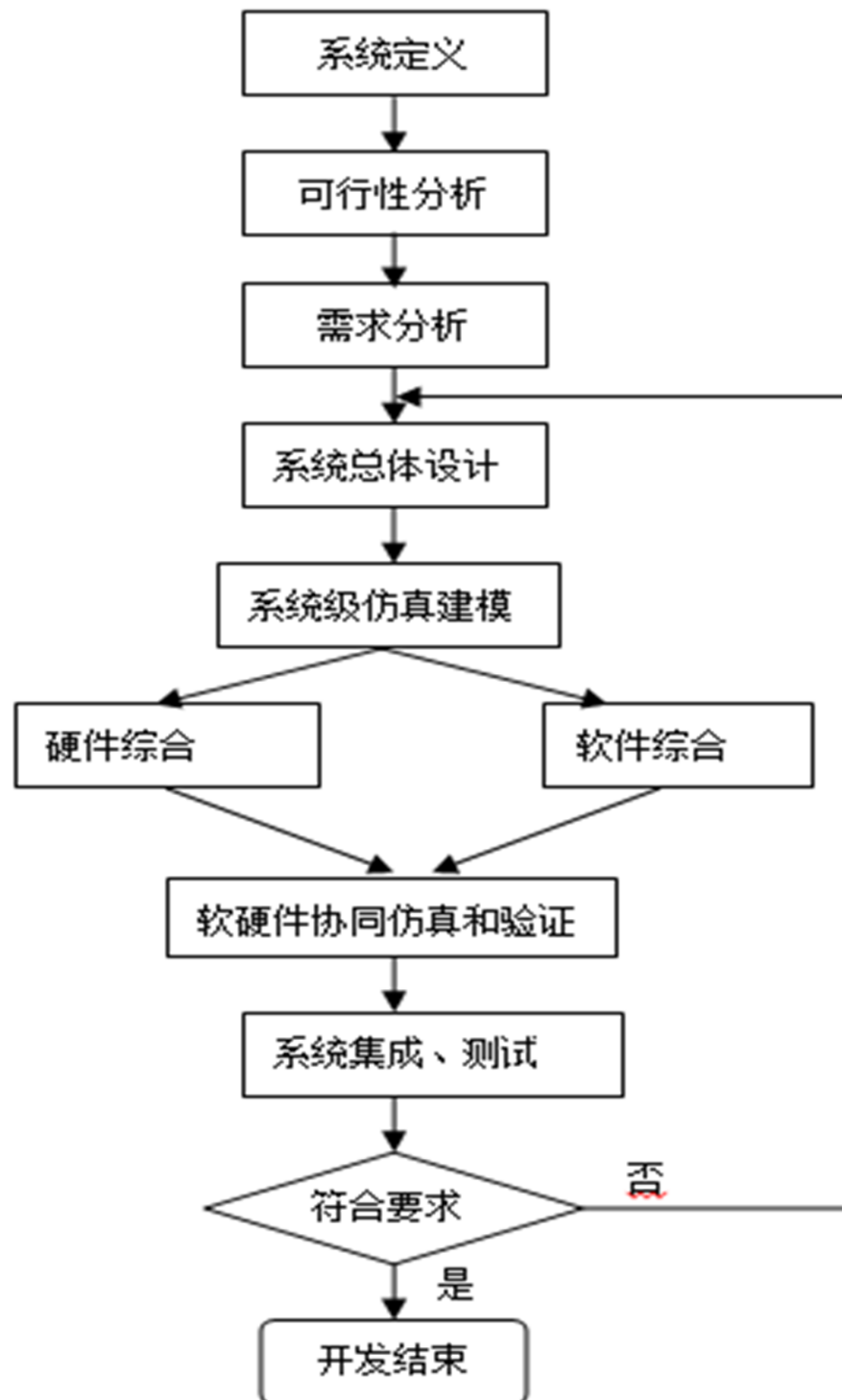
- 嵌入式系统软件结构一般包含四个层面：

- ◆ 设备驱动层
- ◆ 实时操作系统（RTOS）
- ◆ 应用程序接口（API）层
- ◆ 实际应用程序层

2、请说出 BootLoader 的作用

嵌入式系统的开机程序，为系统准备好软硬件环境。对应通用计算机中的 BIOS。系统加电复位后，CPU 从由复位地址上取指令，将首先执行 Boot Loader 程序。

3、请简要说明嵌入式协同设计开发流程



4、简述 Arm-Linux 系统调用原理及其作用

原理：Linux 系统利用 SWI 指令来从用户空间进入内核空间。SWI 指令用于产生软件中断，从而实现从用户模式到管理模式的变换，CPSR 保存到管理模式的 SPSR，执行转移到 SWI 向量。在其他模式下也可使用 SWI 指令，处理器同样地切换到管理模式。

(xdm 这个不是我想用 deepseek，实在是我真没办法从复习资料和 PPT 里面找到原

文，我严重怀疑这个知识点就没有教过，只能交给 deepseek 了。请谨慎参考)

意义：

- 安全性：通过限制用户程序直接访问硬件资源和操作系统内核，系统调用提供了安全屏障，防止恶意软件直接操作硬件或篡改操作系统。
- 封装性：系统调用为硬件操作提供了抽象层，使得上层应用程序无需了解底层的硬件细节即可进行复杂的操作。
- 高效性：系统调用机制允许操作系统内核优化和管理资源，提高系统的整体效率和性能。
- 可移植性：尽管不同的硬件架构（如 ARM 与 x86）在具体实现上有所不同，但系统调用的概念和接口在各种平台上保持一致，这有助于提高软件的跨平台兼容性。