



合肥工业大学

HEFEI UNIVERSITY OF TECHNOLOGY

2022217587 党存远

嵌入式系统原理 第二次作业.

1. ARM 微处理器支持哪几种运行模式? 各运行模式有什么特点。

答: ARM 微处理器支持模式如下:

1. USR (用户模式): ARM 处理器正常的程序执行状态, 在此模式下某些被保护的系统资源是不能被访问的, 应用程序也不能直接进行处理器模式的切换。在此模式下, 当需要进行处理模式切换时, 应用程序可以产生异常处理, 在异常处理中进行处理器模式的切换。

2. FIQ (快速中断模式): 用于高速数据传输或通道处理。

3. IRQ (外部中断模式): 用户通用的中断处理。

4. SVC (管理模式): 操作系统使用的保护模式, 处理软件中断。

5. ABT (数据访问中止模式): 当数据或指令预取中止时进入该模式, 可用于虚拟存储器与存储器保护。

6. UND (未定义指令模式): 当未定义的指令执行时进入该模式, 可用于支持硬件协处理器的软件仿真。

7. SYS (系统模式): 运行具有特权的操作系统任务。

2. 已知 ARM920T 微处理器的存储块类型为极小页, 请给出其虚拟地址到物理地址的转换过程。

答: ARM920T 微处理器首先使用虚拟地址的高位部分来访问一级页表, 之后从一级页表得到的基地址与虚拟地址中的下一部分结合, 用来访问二级页表, 在二级页表中每个条目对应一个物理页面基地址, 之后从二级页表得到的基地址与虚拟地址的页内偏移地址相加得到完整的物理地址。





合肥工业大学

HEFEI UNIVERSITY OF TECHNOLOGY

LSL R0, R0, #24 // R0 = D, 0, 0, 0

ORR R0, R0, R3 // R0 = D, C, 0, 0

ORR R0, R0, R2 // R0 = D, C, B, 0

ORR R0, R0, R1 // R0 = D, C, B, A.

解释: 首先将 R0 的值分别赋值给 R3, R2, R1

之后将 R3, R2, R1 与 R0 中值变为 0, 0, 0, A / 0, 0, B, 0 / 0, 0, C, 0 / 以及 D, 0, 0, 0

最后进行或操作即可。

3. 使用 ARM 汇编完成下列 C 的数组赋值。

for(i=0; i<=10; i++) [a[i]=b[i]+c]

要求: 写出 ARM 汇编程序, 并解释执行过程

a: R1 b: R2 c: R3.

MOV R0, #0 // 初始化 i

loop:

CMP R0, #10

BGT end // 若 R0 > 10 则跳出循环

LDR R4, [R2, R0, LSL #2] // 将 R4 赋予 b[i] 的数据

ADD R4, R4, R3 // R4 = R4 + R3

STR R4, [R1, R0, LSL #2] //

将 R4 中的数据赋值

给 a[i]

ADD R0, R0, #1 // i = i + 1

B Loop

end





程序的执行过程:

首先对 i 进行赋值值为 0, 之后比较 i 与 10 的大小判断循环条件.
若 $i > 10$ 则退出循环, 否则继续执行 Loop 下操作.
当 $i \leq 10$ 时: 将 $b[i]$ 的值加载到 R_4 寄存器中, $R_4 = b[i]$
利用 ADD 指令将 R_4 赋值于 $b[i] + c$, 之后将 R_4 的数据传回 $a[i]$ 中
最后对 i 进行递增操作然后重复执行 Loop.

1. 试结合嵌入式存储系统的层次结构, 说明存储器的类型特点与用途.

答: 存储器可分为: 内部寄存器: 用于存储数据, 加速指令的执行速度.

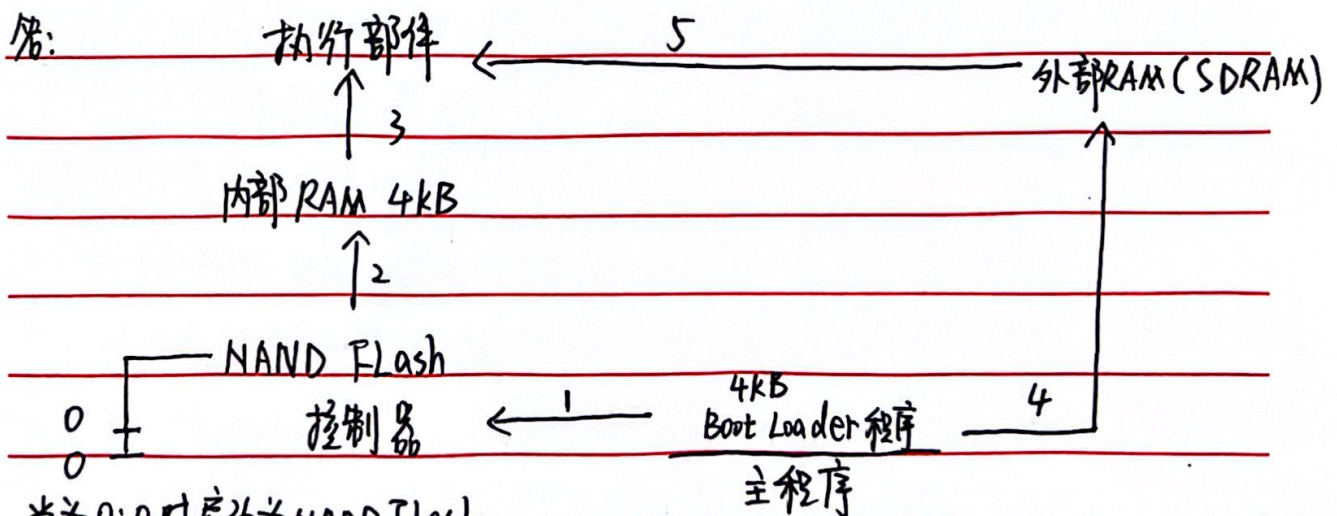
芯片的高速缓存: 存储将要执行的指令和数据, 提高指令的执行速度.

主存储器 1: 存储将要执行的指令和数据.

主存储器 2: 存储固化的系统程序, 运行初始化程序等.

外部存储器: 存储大量的数据.

2. 试分析使用 NAND flash 作为引导 ROM 时系统的启动流程.



当为 0:0 时启动为 NAND Flash

首先通过 NAND Flash 读取 Boot Loader 到 ~~临时RAM上~~ 一个特定区域.





之后将 NAND Flash 中数据写入 RAM 中, 该 RAM 称为 Stepping Stone, 是一个临时 RAM, 当 Boot Loader 被加载到临时 RAM 后, 处理器开始执行该内存中的代码, 之后将更完整的 Boot Loader 加载到主内存中即 (SDRAM) 最后进行执行操作。

3. 请说明使用 8 位存储器芯片构成 8、16 和 32 位存储器系统时, 存储器芯片的地址线与微处理器芯片地址总线的连接方法。

a. 8 位: 存储器的 8 位数据线 $DA_7 \sim DA_0$ 与 S3C2440 的低 8 位数据总线 $DATA_7 \sim DATA_0$ 相连; 存储器的地址线 A_1, A_0 与 S3C2440 的地址总线 $ADDR_1, ADDR_0$ 相连; 存储器的允许输出信号 nOE 端连接 S3C2440 的 nOE 引脚, 写信号 nWE 连 nWB 引脚; 片选信号 nCE 连接 S3C2440 的 $nGCS_n$ 引脚。

b. 16 位: 低 8 位存储器的 8 位数据线 $DA_7 \sim DA_0$ 与 S3C2440 的低 8 位数据总线 $DATA_7 \sim DATA_0$ 相连; 高 8 位存储器的 8 位数据线 $DA_7 \sim DA_0$ 与 S3C2440 的高 8 位数据总线 $DATA_{15} \sim DATA_8$ 相连。两片存储器地址线 A_1, A_0 与 S3C 中 $ADDR_2$ 与 $ADDR_1$ 相连。两片允许输出信号 nOE 接芯片 nOE , 低 8 位存储器片选信号 nWE 接 $nWBE_0$, 高 8 位片选信号 nWE 接 $nWBE_1$ 引脚, 两个片选信号接入 $nGCS_n$ 。

c. 32 位: 低 8 位 $DA_7 \sim DA_0 \rightarrow$ 芯片 $DATA_7 \sim DATA_0$

次低 8 位 $DA_7 \sim DA_0 \rightarrow$ 芯片 $DATA_{15} \sim DATA_8$

次高 8 位 $DA_7 \sim DA_0 \rightarrow$ 芯片 $DATA_{23} \sim DATA_{16}$

高 8 位 $DA_7 \sim DA_0 \rightarrow$ 芯片 $DATA_{31} \sim DATA_{24}$

四片存储器地址线分别与 $ADDR_3, ADDR_2$ 与 A_1, A_0 相连。

允许输出信号 $nOE \rightarrow nOE$, 片选信号 $nCE \rightarrow nGCS_n$

低 8 位, 次低 8 位, 次高 8 位与高 8 位分别与 $nWBE_0, nWBE_1, nWBE_2, nWBE_3$ 相连。

