Technology Tarena 达内科技

ARM 异常处理程序设计

ARM FUNDAMENTAL DAY09

知识讲解

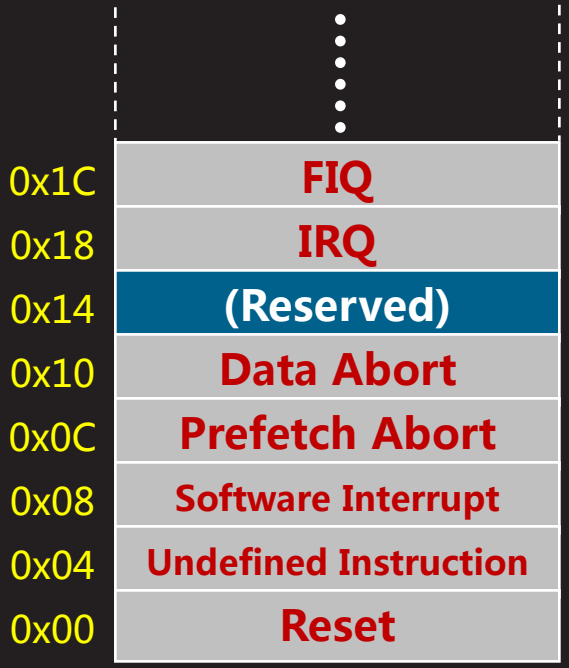
异常处理

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
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：

从 SPSR_<mode>恢复CPSR
从LR_<mode>恢复PC
Note：这些操作只能在 ARM 态执行。



Vector Table

Vector table can be at 0xFFFF0000 on ARM720T and on ARM9/10 family devices

Technology Tarena 达内科技

知识讲解

异常优先级

异常在当前指令执行完成之后才被响应
多个异常可以在同一时间产生

异常指定了优先级和固定的服务顺序:

- Reset
- Data Abort
- FIQ
- IRQ
- Prefetch Abort
- SWI
- Undefined instruction

知识讲解

异常返回指令

异常返回：

使用一数据处理指令：相应的指令取决于什么样的异常

指令带有 “S” 后缀
PC做为目的寄存器

在特权模式不仅仅更新PC，而且拷贝SPSR 到 CPSR

从SWI 和 Undef异常返回

```
MOVS  pc, lr
```

从FIQ, IRQ 和 预取异常(Prefect Abort)返回

```
SUBS  pc, lr, #4
```

从数据异常(Data Abort)返回

```
SUBS  pc, lr, #8
```

如果 LR之前被压栈的话使用LDM “ ^”

```
LDMFD  sp!, {pc}^
```

异常返回地址

ARM 状态:

在异常产生的时候内核设置 LR_mode = PC - 4.
处理程序需要调整 LR_mode (取决于是哪一个异常发生了)，以便返回到正确的地址

Thumb 状态:

处理器根据发生的异常自动修改存在 LR_mode 中的地址
不论异常产生时的状态如何，处理器确保处理程序的ARM返回指令能返回到正确的地址(和正确的状态)

知识讲解

从SWIs和未定义指令返回

Technology Tarena 达内科技

异常是由指令本身引起的，因此内核在计算 LR 时的 PC 值并没有被更新.

		ARM	Thumb	
SWI		pc-8	pc-4	←;Exception taken here
xxx	⌫	pc-4	pc-2	←;lr = next instruction
yyy		pc	pc	

因此返回指令为:

MOVS pc, lr

Note : ⌫ 表示异常返回后将执行的那条指令

知识讲解

从FIQs和IRQs返回

Technology Tarena 达内科技

异常在当前指令执行完成后才被响应，因此内核在计算 LR 时的 PC 值已被更新。

		ARM	Thumb	
www		pc-12	pc-6	←Interrupt occurred during execution
xxx	⌫	pc-8	pc-4	
yyy		pc-4	pc-2	←ARM lr = next instruction
zzz		pc	pc	←Thumb lr = two instructions ahead

因此返回指令为:

SUBS pc, lr, #4

Note : ⌫表示异常返回后将执行的那条指令

知识讲解

从预取异常返回

Technology Tarena 达内科技

当指令到达执行阶段时异常才产生，因此内核在计算 LR 时的 PC 值已被更新。

需要重新执行导致异常的指令

	ARM	Thumb	
www ⌫	pc-8	pc-4	← Prefetch Abort occurred here
xxx	pc-4	pc-2	← ARM lr = next instruction
yyy	pc	pc	← Thumb lr=two instructions ahead

因此返回指令为:

SUBS pc, lr, #4

Note : ⌫表示异常返回后将执行的那条指令

知识讲解

从数据异常返回

Technology Tarena 达内科技

异常发生（和计算 LR）在 PC 被更新之后，需要重新执行导致异常的指令。

	ARM	Thumb	
www ⌫	pc - 12	pc - 6	←Data abort occurred here
xxx	pc - 8	pc – 4	
yyy	pc - 4	pc - 2	←ARM lr = two instructions ahead
zzz	pc	pc	
aaa	pc + 4	pc + 2	←Thumb lr = four instructions ahead

因此返回指令为:

SUBS pc, lr, #8

Note : ⌫表示异常返回后将执行的那条指令

中断处理

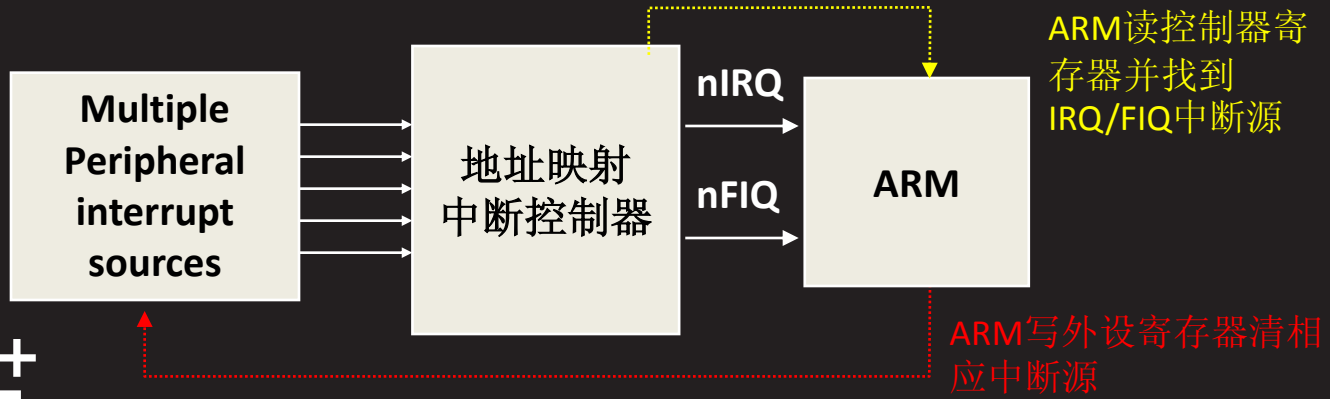
ARM 有两级外部中断 FIQ，IRQ.

可是大多数的基于ARM 的系统有 >2个的中断源!

因此需要一个中断控制器(通常是地址映射的)来控制中断是怎样传递给ARM的。

在许多系统中，一些中断的优先级比其它中断的优先级高，他们要抢先任何正在处理的低优先级中断。

Note: 通常中断处理程序总是应该包含清除中断源的代码。



知识讲解

FIQ vs IRQ 回顾

Technology Tarena 达内科技

FIQ 和 IRQ 提供了非常基本的优先级级别。

FIQs有高于IRQs的优先级，表现在下面2个方面:

- 当多个中断产生时，CPU优先处理FIQ.
- 处理 FIQ时禁止 IRQs.
- IRQs 将不会被响应直到 FIQ处理完成.

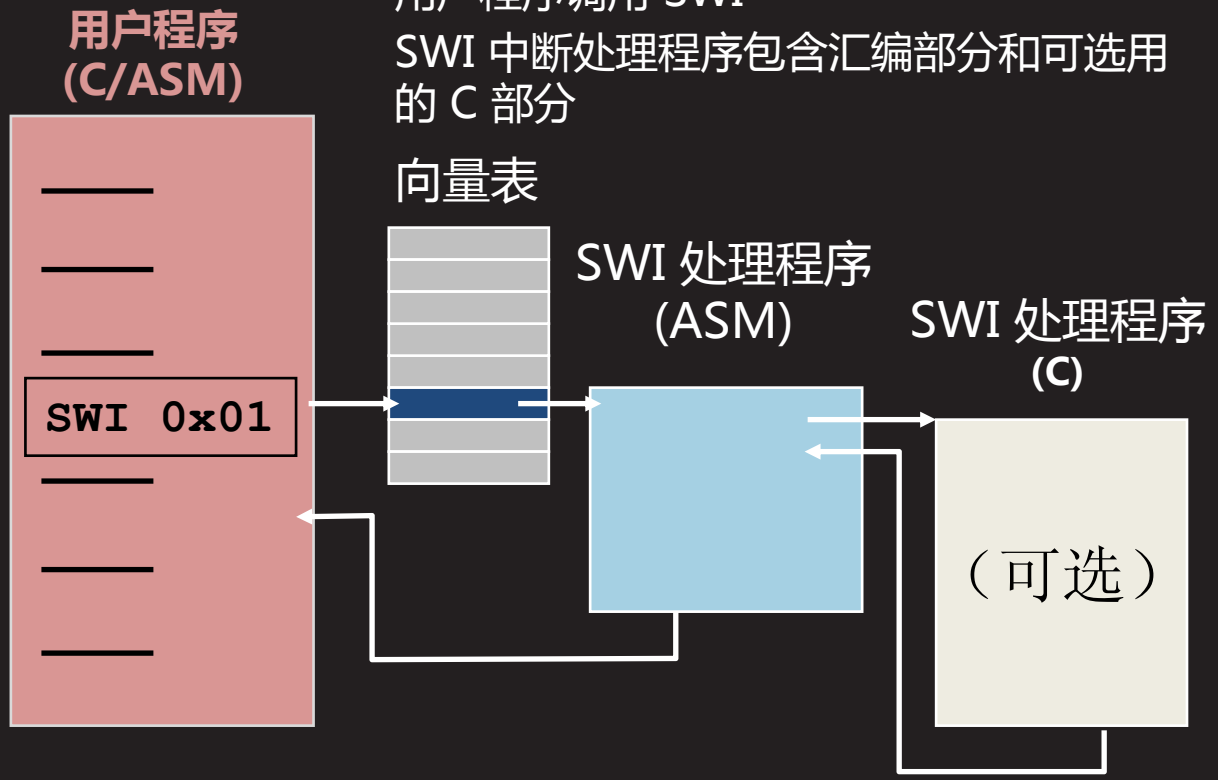
FIQs 的设计使中断响应尽可能的快.

- FIQ 向量位于中断向量表的最末.中断处理程序可从中断向量处连续执行
- FIQ 模式有5个额外的私有寄存器 (r8-r12)中断处理必须保护其使用的非私有寄存器
- 可以有多个FIQ中断源,但是考虑到系统性能应避免嵌套。

软中断

用户程序调用 SWI

SWI 中断处理程序包含汇编部分和可选用的 C 部分



知识讲解

SWI 调用

Technology Tarena 达内科技

汇编中, SWI 调用使用“SWI　中断号”实现, e.g:

SWI 0x24

小心在汇编中如果SWI 调用时处于Supervisor模式将会冲掉LR_svc.

例如：在SWI处理程序中的二级调用
解决方法: 在SWI调用之前对LR_svc 压栈保护

知识讲解

获取 SWI 指令号

Technology Tarena 达内科技

ARM 内核不提供直接传递软中断(SWI)号到处理程序的机制：
SWI 处理程序必须定位SWI 指令，并提取SWI指令中的常数域
为此, SWI 处理程序必须确定SWI 调用是在哪一种状态(ARM/Thumb).
检查 SPSR 的 T-bit
SWI 指令在ARM 状态下在 LR-4 位置, Thumb 状态下在 LR-2位置
SWI 指令按相应的格式译码：
ARM 态格式:

Thumb 态格式:

知识讲解

软中断(SWI)处理示例

Technology Tarena 达内科技

寄存器压栈，
设置堆栈指针

```
SWI_Handler :
    STMFD   sp!, {r0-r3,r12,lr}
    MOV     r1, sp
```

```
    MRS     r0, spsr
    STMFD   sp!, {r0}
```

取出 spsr 并压栈保存

提取SWI 指令的常量域(24-bits: 如果从ARM中调用,
8-bits: 如果从Thumb中调用)

```
    TST     r0, #0x20
    LDRNEH  r0, [lr,#-2]
    BICNE   r0, r0, #0xff00
    LDREQ   r0, [lr,#-4]
    BICEQ   r0, r0, #0xff000000
```

```
; r0 now contains SWI number
; r1 now contains pointer to parameters on stack
```

调用 C SWI 处理程序

```
    BL      C_SWI_Handler
```

```
    LDMFD   sp!, {r1}
    MSR     spsr_csxf, r1
```

```
    LDMFD   sp!, {r0-r3,r12,pc}^
```

恢复寄存器并返回

知识讲解

C SWI 处理程序示例

Technology Tarena 达内科技

```
// Memory mapped registers
volatile unsigned parallel_output, parallel_input;
void C_SWI_Handler (unsigned number, int *param)
// r0 = SWI number
// r1 = pointer to SWI parameters in memory
{
        switch (number)
        {
                case 0:
                        parallel_output = param[0];
                        break;
                case 1:
                        param[0] = parallel_input;
                        break;
                default :
                        break;
        }
}
```

知识讲解

复位（Reset）

Technology Tarena 达内科技

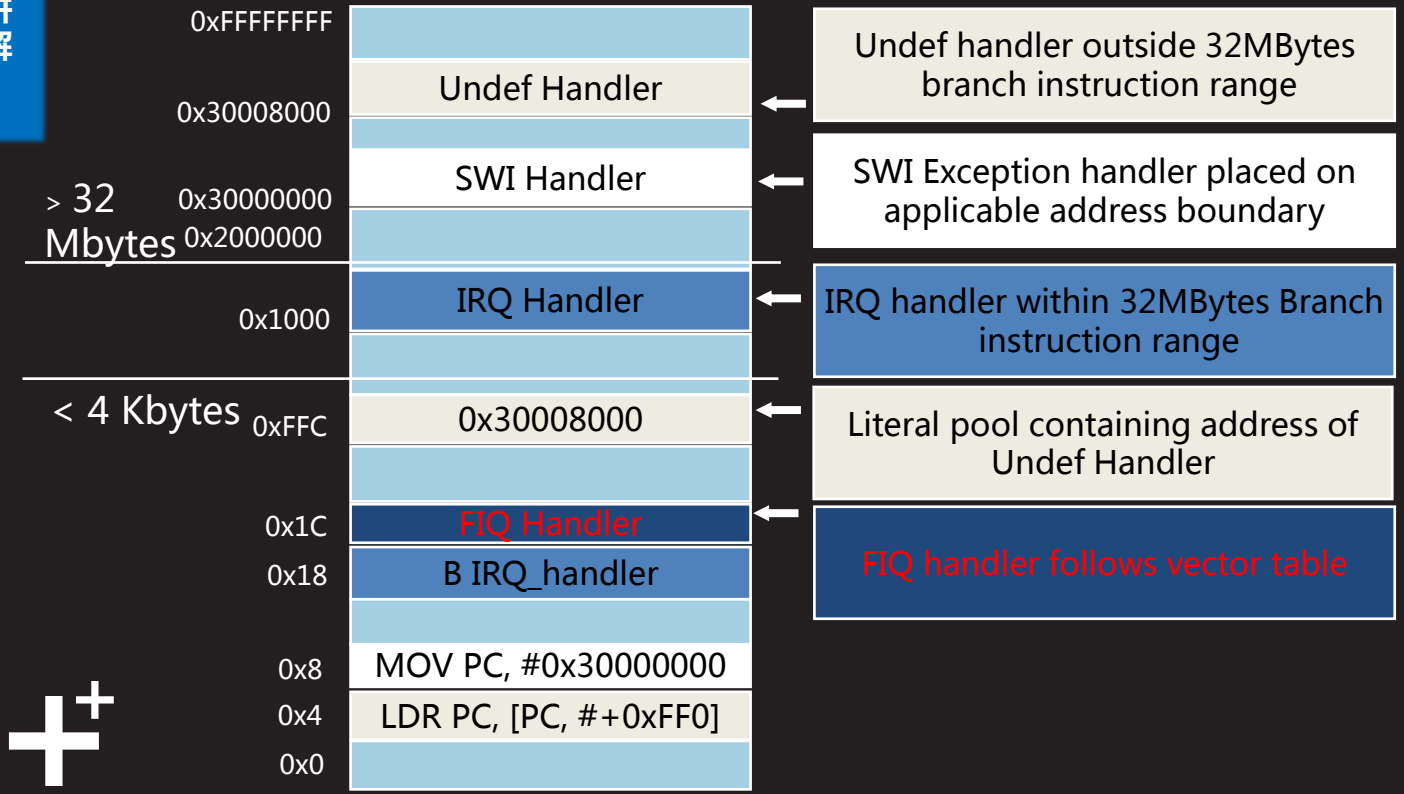
Reset 处理程序执行的动作取决于不同的系统。例如它可以：

- 设置异常向量
- 关闭所有中断及watchdog
- 初始化存储器系统 (e.g. MMU/PU)
- cpu频率设置等
- 初始化所有需要的模式的堆栈和寄存器
- 初始化所有 C 所需的变量
- 初始化所有I/O设备
- 改变处理器模式或/和状态
- 调用主应用程序

知识讲解

异常向量表&异常处理程序入口

Tarena Technology 达内科技



知识讲解

异常向量表的处理

异常向量表的处理方法

使用B指令，示例：

```
_start:
        b reset_start          @ handlerReset
        b except_undef         @ handlerUndef
        b except_svc           @ SWI interrupt handler
```

知识讲解

异常向量表的处理（续）

Technology Tarena 达内科技

异常向量表的处理方法

使用LDR指令，示例：

```
start:
        ldr pc, _start
        ldr pc, _undefined_instruction
        ldr pc, _software_interrupt
        ldr pc, _prefetch_abort
 _undefined_instruction:
          .word undefined_instruction
 _software_interrupt:
         .word software_interrupt
 _prefetch_abort:
         .word prefetch_abort
```

知识讲解

ARM MMU 原理简介

Technology Tarena 达内科技

MMU（内存管理单元）

* 内存变换
* 内存保护

代码实践

课堂练习

ARM 异常向量表安装 & 软中断异常程序设计

1、编写异常向量表代码
2、编写软中断异常处理程序
3、编写安装异常向量代码

Technology Tarena 达内科技

代码实践

课堂练习

中断程序设计

1、编写基于S5PV210 的中断处理程序

Technology Tarena 达内科技