

# 三星系列ARM处理器开发技术

# S3C44B0芯片特点

---

2.5伏内核电压，8KB Cache, 最高主频66MHz

内置LCD控制器，支持256色彩色/16级灰度/黑白显示

内置IIC总线控制器

内置IIS总线控制器

71个通用I/O端口，支持最多8个外部扩展中断

8路10位ADC

二个串口，支持IrDA1.0

5个定时器，支持日历功能的实时时钟

8个外部存储控制器bank，支持ROM/SRAM/DRAM等

# 外部存储器控制接口

支持Little/Big Endian模式

共8个Bank，其中

6个支持ROM、SRAM(nGCS0-nGCS5)

2个支持ROM、SRAM、FD/EDO/SDRAM(nGCS6-nGCS7)

每个Bank均支持最大32MByte大小空间，总共256MBytes

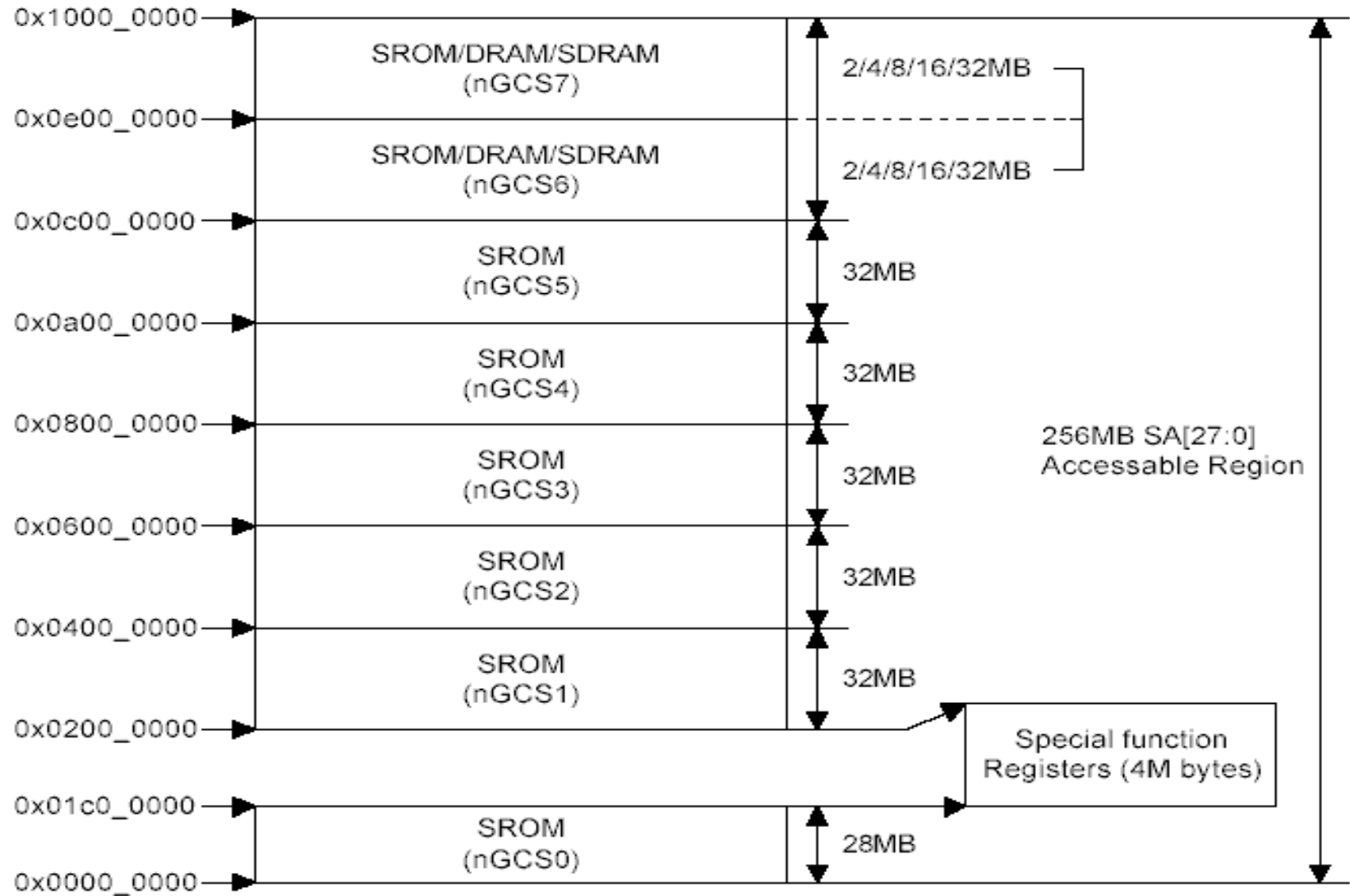
7个Bank内存起始地址固定，大小可编程(nGCS0- nGCS6)

1个Bank内存起始地址和大小均可编程(nGCS7)

所有Banks访问周期均可编程控制

所有Banks均支持8/16/32位数据访问宽度

# 存储地址映射表



# 存储控制相关寄存器

名字	地址	描述	R/W
BWSCON	0x01C80000	各Bank数据宽度和等待周期控制寄存器	R/W
BANKCON0	0x01C80004	Bank 0访问周期控制寄存器	R/W
...	...	...	
BANKCON5	0x01C80018	Bank 5访问周期控制寄存器	
BANKCON6	0x01C8001C	Bank 6访问周期和存储类型控制寄存器	R/W
BANKCON7	0x01C80020	Bank 7访问周期和存储类型控制寄存器	
REFRESH	0x01C80024	DRAM/SDRAM刷新周期控制寄存器	R/W
BANKSIZE	0x01C80028	Bank Size控制寄存器	R/W
MRSRB6	0x01C8002C	Bank 6模式控制寄存器	R/W
MRSRB7	0x01C80030	Bank 7模式控制寄存器	

# 中断控制

---

- 1、支持30个中断源，包括8个外部中断源
- 2、中断控制器支持26个中断信号，其中外部中断4/5/6/7复用到一个中断信号上(读EXTINTPND寄存器来判断)
- 3、每个中断信号处理均可独立设置成FIQ或者IRQ模式
- 4、中断信号处理均可采用矢量处理方法(vector interrupt mode)或者非矢量处理方法
- 5、支持中断优先级排队

# 非矢量中断处理方式

---

非矢量中断处理模式是一种传统的中断处理方法。当系统产生中断的时候，系统将interrupt pending寄存器中对应中断的标志位置位，然后跳转到位于0x18处的统一中断处理函数中，该中断函数通过读取interrupt pending寄存器来判断中断源，并根据优先级关系再跳转到对应中断源的处理代码内。

# 矢量中断处理方式

矢量中断处理模式是当CPU读取位于0x18处的IRQ中断处理指令时候，系统自动读取对应于该中断源的确定地址上的指令取代0x18处的指令，通过跳转指令，系统就直接跳转到对应的中断处理函数中，节省了中断处理时间，提高了中断处理速度。

例如，ADC中断的矢量地址为0xC0，则在0xC0处放置以下代码：

```
ldr pc, =HandlerADC /* mGKB */
```

当ADC中断产生的时候，系统会自动调转到HandlerADC函数代码中。



# 动态改变中断处理函数方法

在矢量中断处理方式中，可以非常方便的动态随时改变中断处理函数。方法是让中断对应的确定地址代码调转到RAM的固定地址处，就可以随时通过替换RAM固定地址处的代码来实现动态改变中断处理函数。

以上页的ADC中断处理为例。

定义HandlerADC为RAM中固定地址：

```
.equ HandlerADC, 0x0C7fff20
```

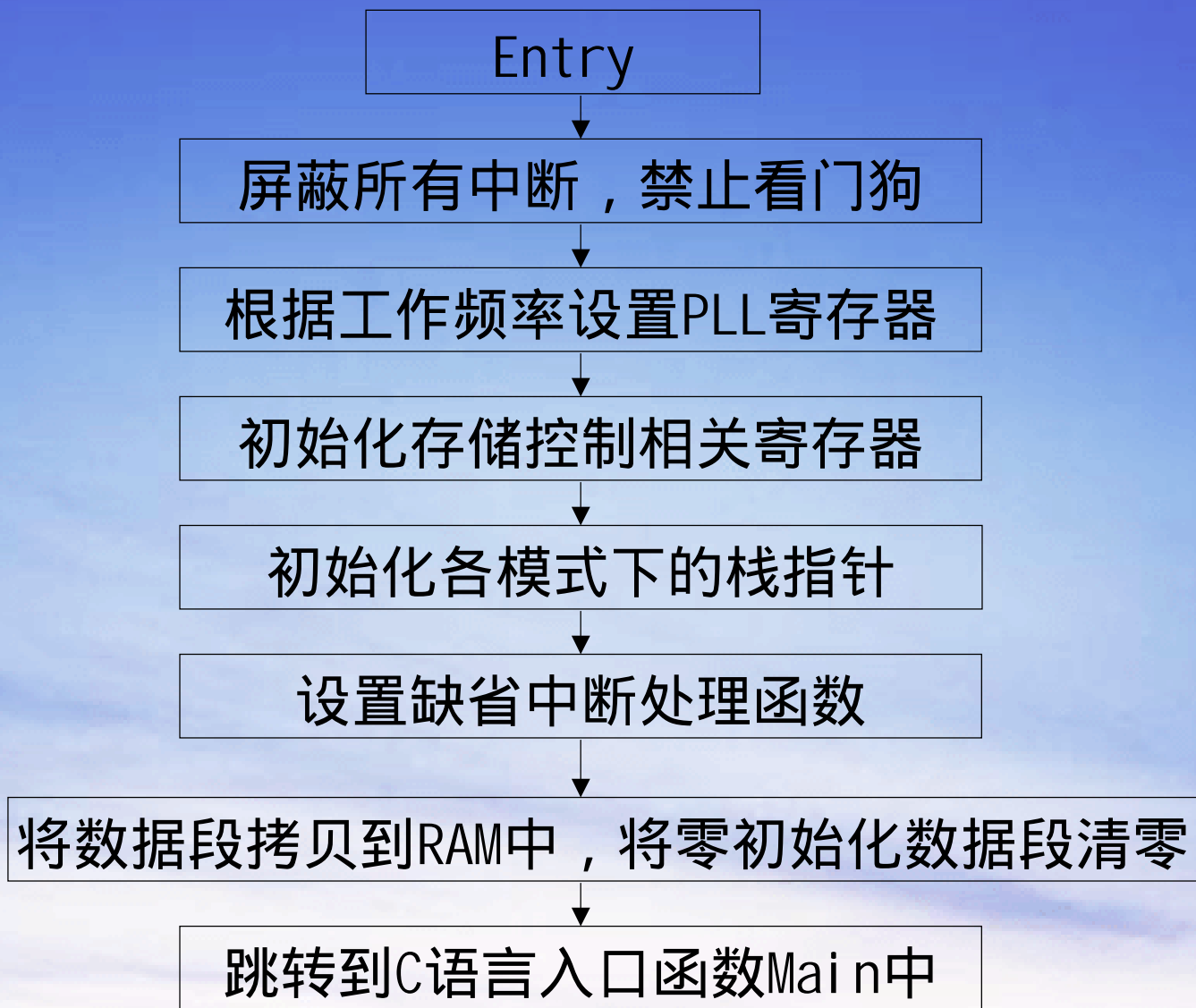
定义一个函数指针指向该固定地址：

```
#define pISR_ADC (*(unsigned *) (0x0C7fff20))
```

就可以在代码中随时改变中断处理函数了：

```
pISR_ADC = (unsigned)MyADCInt;
```

# 启动流程



# 软件开发注意事项

---

- 要正确设置PLL寄存器，要满足三星手册内的约束条件，并使SDIV值越大越好
- 各处理器模式的栈指针初始化值一定要确保各模式栈有足够空间，不会相互重叠导致数据破坏
- 中断处理尽可能使用动态改变方法，便于跟踪调试和根据需要设置中断函数