

## 14. 复位控制器(RSTC)

### 14.1 描述

复位控制器(RSTC)，基于上电复位单元，不用任何外部元件即可处理系统所有的复位。它报告最近一次发生的复位。

复位控制器还可以独立或同时驱动外部复位和外设以及处理器复位。

### 14.2 方块图

图 14-1 复位控制器方块图

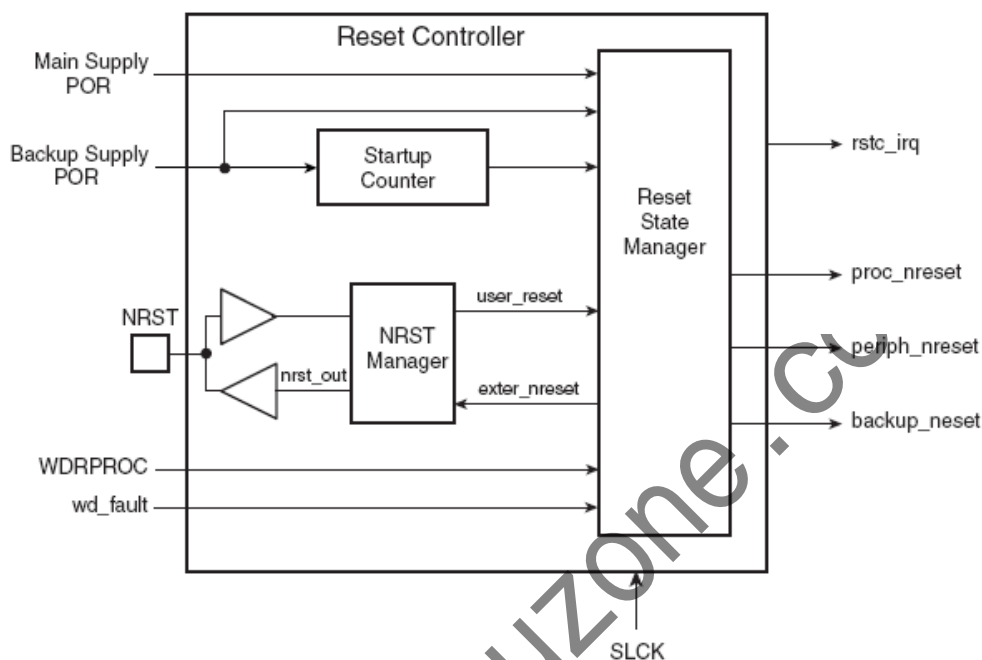
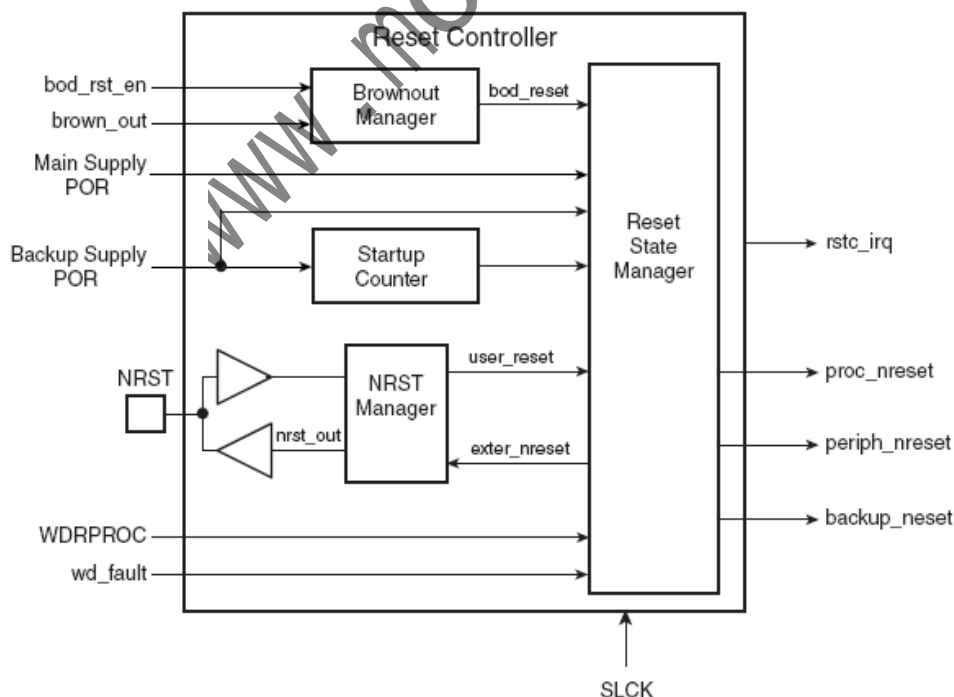


图 14-2 复位控制器方块图



### 14.3 功能描述

### 14.3.1 复位控制器概述

复位控制器由一个 NRST 管理器，一个启动计数器和一个复位状态管理器组成。运行在低速时钟并产生以下复位信号：

- proc\_nreset:处理器复位。还复位看门狗定时器
- backup\_nreset:影响所有由 VDDBU 提供电源的外设。
- periph\_nreset:影响所有的集成外设。
- nrst\_out:驱动 NRST 引脚。

这些复位信号在外部事件或软件作用时被复位控制器使能。当一个 NRST 信号需要输出时，复位状态管理器控制复位信号的发生并提供一个到 NRST 管理器的信号。

NRST 管理器在一个可编程的时间里形成 NRST 引脚上的有效信号，以此方式控制外部设备的复位。

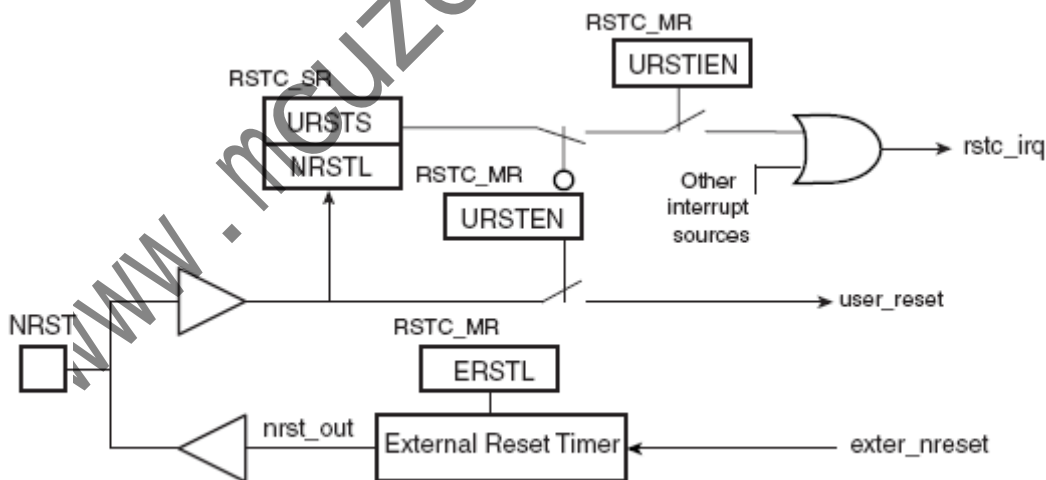
启动计数器等待晶振完全启动。等待晶振启动的时间的最大值可以查表获得，晶振启动时间最大值可以在产品文献的电特性段落中的晶振特性段落中找到。

复位控制器模式寄存器 (RSTC\_MR)，允许复位控制器的配置，RSTC\_MR 由 VDDBU 供电，所以只要 VDDBU 电压存在，其配置就会被保存。

### 14.3.2 NRST 管理器

NRST 管理器采样 NRST 输入引脚，并当复位状态管理器要求时，使此引脚为低电平。图 14-2 展示了 NRST 管理器的方块图。

图 14-3 NRST 管理器



#### 14.3.2.1 NRST 信号或中断

NRST 管理器在慢速时钟采样 NRST 引脚。当此口线被检测为低电平，一个用户复位被报告给复位状态管理器。

然而，当一个 NRST 的有效信号产生，NRST 管理器可以被编程为不触发一个复位。将 RSTC\_MR 中的 URSTEN 位写 0 则禁用用户复位发生。

NRST 引脚的电平可以在任何时间在 RSTC\_SR 中的 NRSTL 位 (NRST 电平) 被读取。只要 NRST 引脚被激活，RSTC\_SR 中的 URSTS 位被置位。此位仅在 RSTC\_SR 被读取时清零。

复位控制器还可以被编程用来发生一个中断来代替发生一个复位。这样做的

话，RSTC\_SR 中的 URSTIEN 位必须被写为 1。

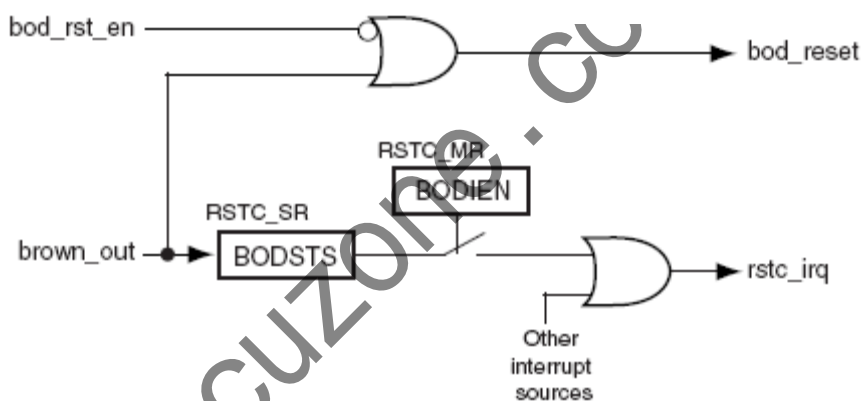
### 14.3.2.2 NRST 外部复位控制

复位状态管理器激活 ext\_nreset 信号来激活 NRST 引脚。当前面的情况发生，在一段通过 RSTC\_SR 中的 ERSTL 域编程的时间内，“nrst\_out”信号被 NRST 管理器驱动为低。此有效持续时间，名叫 EXTERNAL\_RESET\_LENGTH，持续了  $2^{(ERSTL+1)}$  个慢时钟周期。这给出了一个有效的在 60 微秒到 2 秒之间的持续时间。注意 ERSTL 设置为 0 对 NRST 脉冲表示持续一个 2 个周期时间。

此功能部件允许复位控制器形成 NRST 引脚电平，以此方式来保证 NRST 口线被驱使为低电平，为可能连接在系统复位上的外部设备提供符合的复位时间。

作为在可备份的 RSTC\_MR 内的域，对于需要一个比慢时钟振荡器长的启动时间的设备，此域可以被用于形成系统上电复位。

图 14-4. Brownout 管理器



### 14.3.3 复位状态

复位状态管理器处理不同的复位源并产生内部复位信号。它报告在状态寄存器(RSTC\_SR)的 RSTTYP 域中的复位状态。RSTTYP 域的更新在处理器复位释放时被执行。

#### 14.3.3.1 通用复位

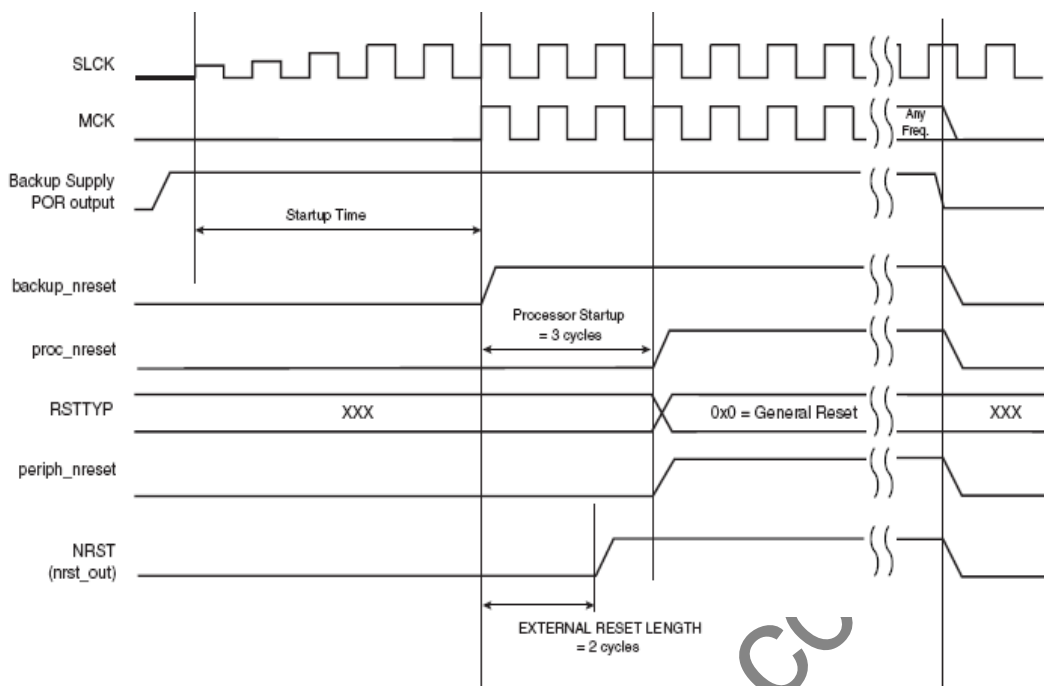
当 VDDBU 上电时产生一个通用复位。备用电源 POR 单元输出上升，用一个启动计数器过滤，此计数器由慢时钟操作。此计数器的目的是确保慢时钟晶振在启动设备前是稳定的。启动时间的长度严格编码到服从慢时钟振荡器启动时间。

此时间后，处理器时钟被使能为慢时钟，并且所有其他信号仍然在 2 个周期对适当的处理器和逻辑复位有效。接着，所有的复位信号被释放并且 RSTC\_SR 中的 RSTTYP 域报告一个通用的复位。随着 RSTC\_MR 复位，NRST 口线在 backup\_nreset 后上升 2 周期，如 ERSTL 在值 0x0 缺省。

当 VDDBU 被备用电源 POR 单元检测为低，所有复位信号立即被激活，即使主电源 POR 单元未报告主电源掉电。

图 14-3 展示了通用复位怎样影响复位信号。

图 14-5 通用复位状态



### 14.3.3.2 唤醒复位

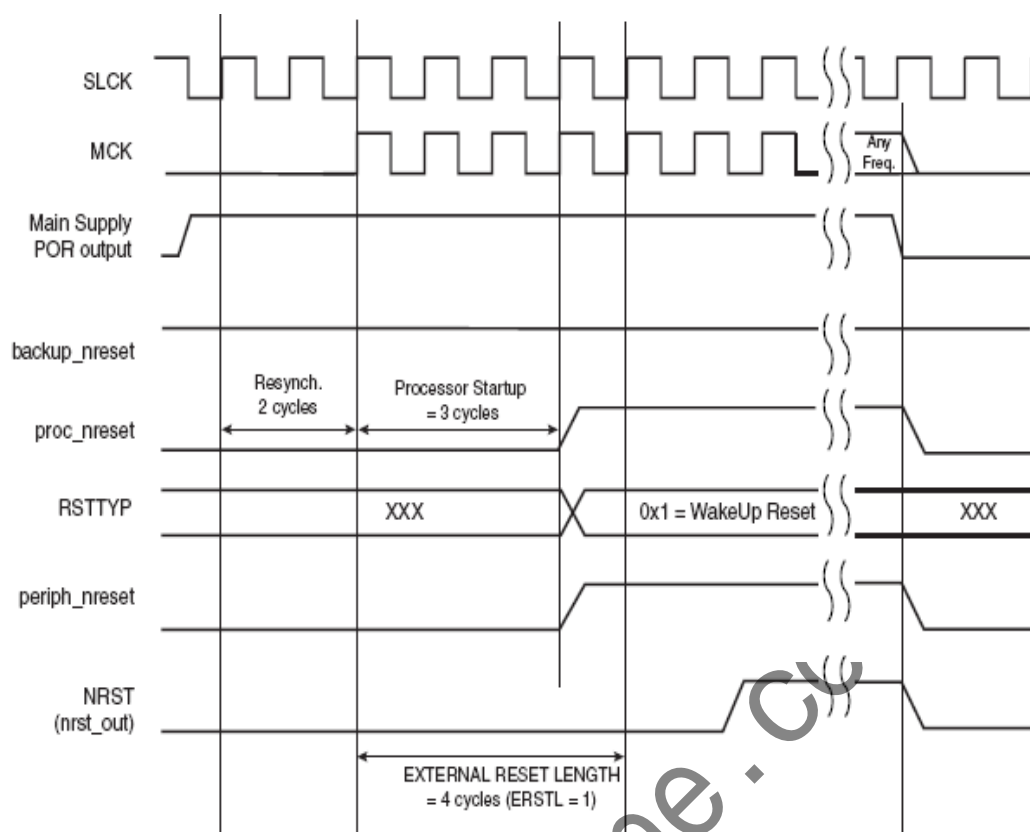
唤醒复位在主电源掉电时产生。当主电源 POR 输出有效，除了 backup\_nreset 所有的复位信号有效。当主电源上电，POR 输出在慢时钟上重新同步化。处理器时钟接着在 2 个慢时钟周期内重新使能，依赖于 ARM 处理器的要求。

在延迟结束时，处理器和其他复位信号升高。RSTC\_SR 中的 RSTTYP 域被更新为报告一个唤醒复位。

“nrst\_out”仍然对 EXTERNAL\_RESET\_LENGTH 周期有效。RSTC\_MR 被备份，编程了的周期数是可应用的。

当主电源被检测到下降（falling），立即使复位信号有效。此转换和主电源 POR 输出同步。

图 14-6 唤醒状态



### 14.3.3.3 用户复位

当在 NRST 引脚上检测到一个低电平时，并且 RSTC\_MR 中的 URSTEN 位是 1，则进入用户复位。NRST 输入信号和 SLCK 同步以确保合适的系统操作。

只要在 NRST 上检测到一个低电平，用户复位就会被送入。处理器复位和外设复位有效。

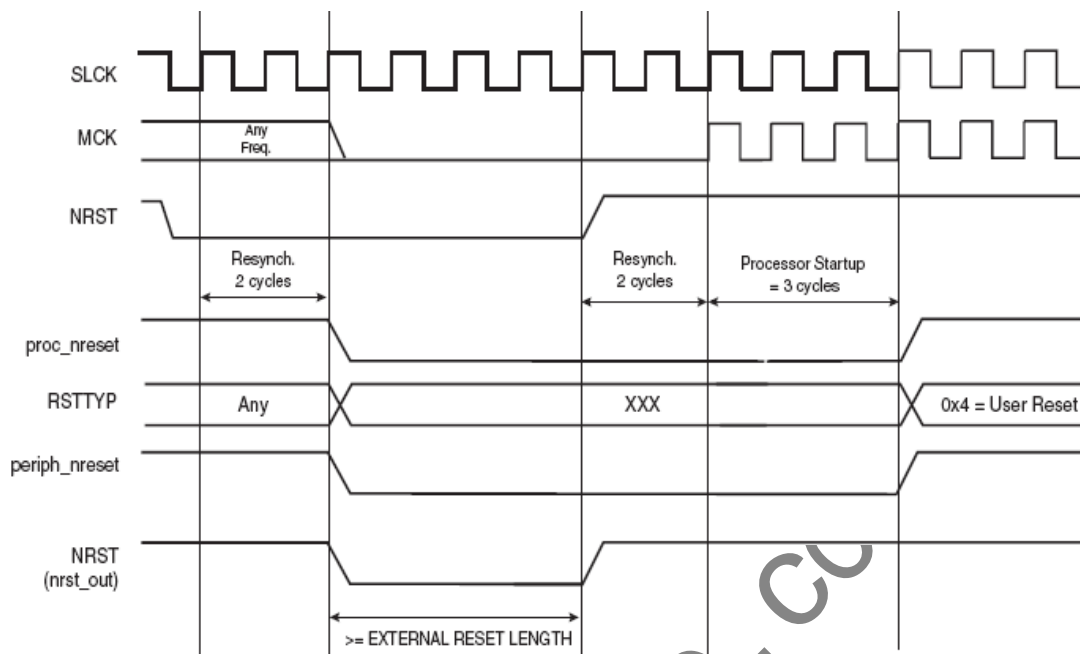
用户复位在一个双周期同步时间和一个三周期处理器启动时间后，当 NRST 上升时消失。

当处理器复位信号被释放，状态寄存器的 RSTTYP 域被用 0x4 的值装载，表示一个用户复位。

NRST 管理器保证 NRST 块对 EXTERNAL\_RESET\_LENGTH 慢时钟周期有效，当在 ERSTL 域中编程。

然而，如果 NRST 在 EXTERNAL\_RESET\_LENGTH 后未上升，比如是因为被外部驱使为低电平，内部复位仍然有效直到 NRST 确切的上升。

图 14-8 用户复位状态



#### 14.3.3.4 软件复位

复位控制器提供若干命令用于激活不同的复位信号。这些命令通过向控制寄存器(RSTC\_CR)中的以下位来执行：

- PROCRST: 写 PROCRST 为 1 来复位处理器和看门狗定时器。
- PERRST: 写 PERRST 为 1 来复位所有的嵌入式外设，包括存储器系统，并且，特别是重映射命令。外设复位通常用作调试目的。
- EXTRST: 写 EXTRST 为 1 使得在被模式寄存器(RSTC\_MR)中的 ERSTL 域定义的一个时间里 NRST 引脚为低电平。

如果这些位中的至少一个通过软件被置位，就会进入软件复位。所有这些命令可以被独立或同时执行。软件复位持续 2 个时钟周期。

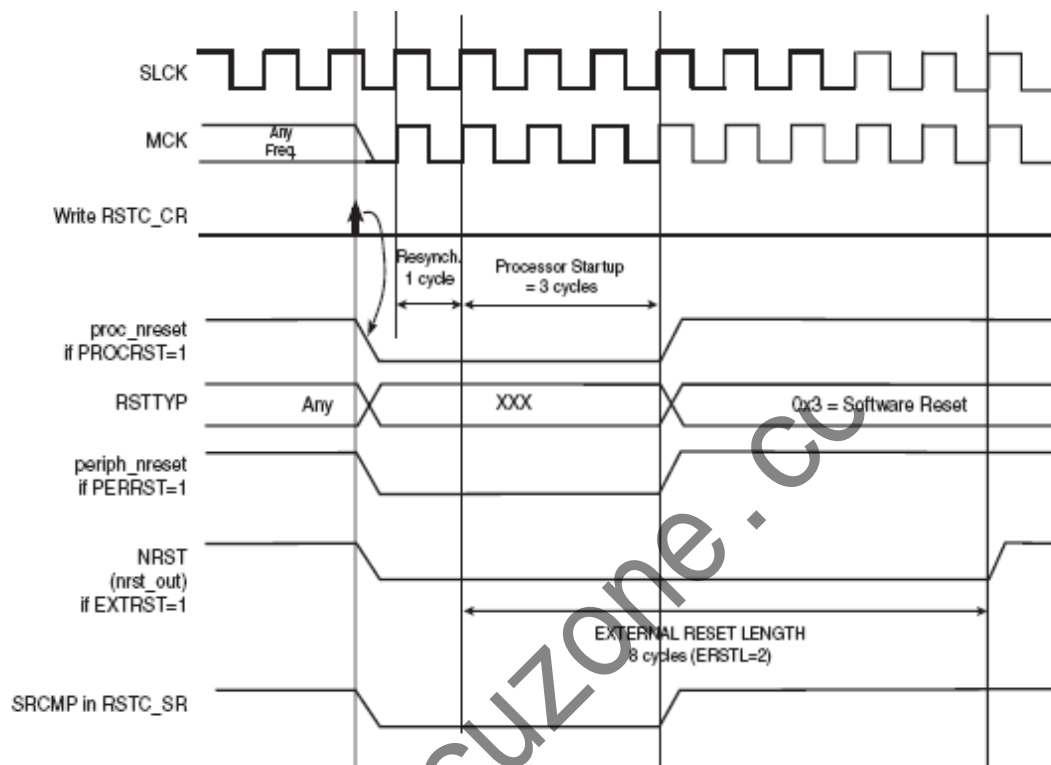
只要寄存器一被写入，内部复位信号就被激活。这在主控时钟 (MCK) 上可以检测到。当软件复位消失他们被释放，就是说，同步到 SLCK。

如果 EXTRST 置位，nrst\_out 信号的激活取决于 ERSTL 域的编程。然而在 NRST 上的下降沿不会导致一个用户复位。

仅当 PROCRST 位被置位，复位控制器才报告在状态寄存器的 RSTTYP 域中的软件状态。其他复位在 RSTTYP 中不报告。

只要一个软件操作被检测到，则 SRCMP 位 (进行中的软件复位命令) 在状态寄存器 (RSTC\_SR) 中被置位。只要软件复位一消失，SRCMP 就被清零。在 SRCMP 置位期间，没有其他软件复位会被执行，；并且向 RSTC\_CR 中写任何值都无效。

图 14-9 软件复位



#### 14.3.3.5 看门狗复位

当看门狗产生超时，则进入看门狗复位。此状态持续 2 个时钟周期。

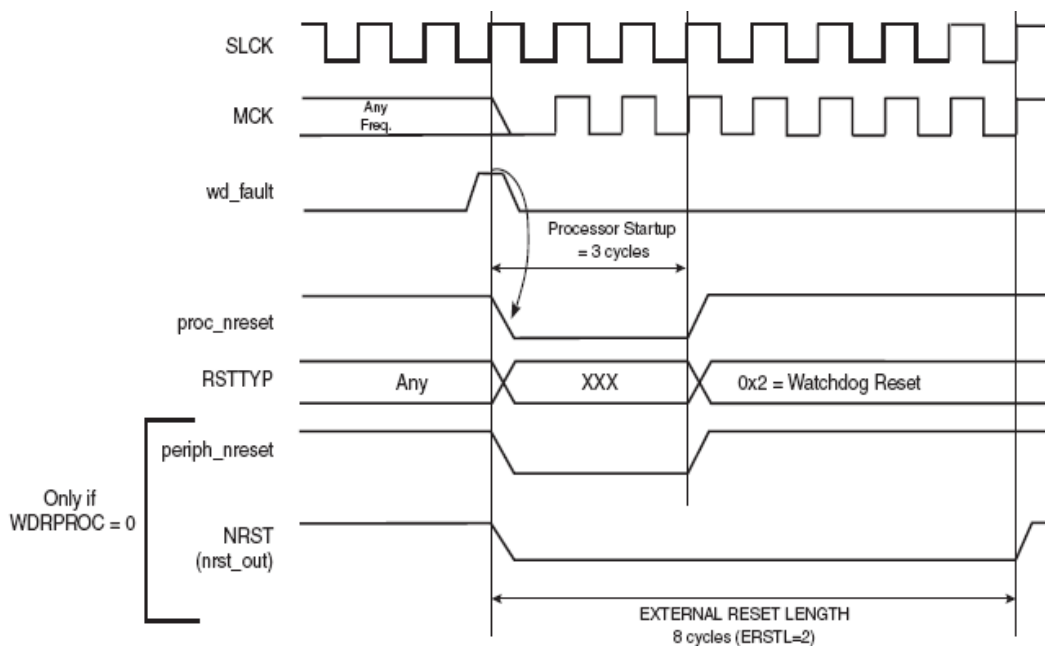
当在看门狗复位时，复位信号的有效依赖于 WDT\_MR 中的 WDRPROC 位：

- 如果 WDRPROC 是 0，处理器复位和外设复位有效。NRST 口线也有效，依赖于 ERSTL 域的编程。然而，NRST 上的低电平并不导致一个用户复位状态。
- 如果 WDRPROC 是 1，仅处理器复位有效。

看门狗定时器被 proc\_nreset 信号复位。因为如果 WDRSTEN 被置位看门狗超时总是导致一个处理器复位，所以看门狗定时器总是在一个看门狗复位后复位，并且看门狗被缺省使能并设置为最大周期。

当 WDT\_MR 位中的 WDRSTEN 被复位，看门狗超时在复位控制器上无影响。

图 14-10 看门狗复位



#### 14.3.4 复位状态优先级

复位状态管理器在不同的复位源间管理以下优先级，以降序给出：

- 备份复位
- 唤醒复位
- 看门狗复位
- 软件复位
- 用户复位

特殊情况如下：

- 当在用户复位：
  - 看门狗事件不可能发生，因为看门狗定时器正被 `proc_nreset` 信号复位。
  - 软件复位不可能发生，因为处理器复位正被激活。
- 当在软件复位：
  - 看门狗事件比当前状态优先级高。
  - NRST 无效
- 当在看门狗复位：
  - 处理器复位被激活并因此软件复位不可能被编程。
  - 用户复位不可能输入。

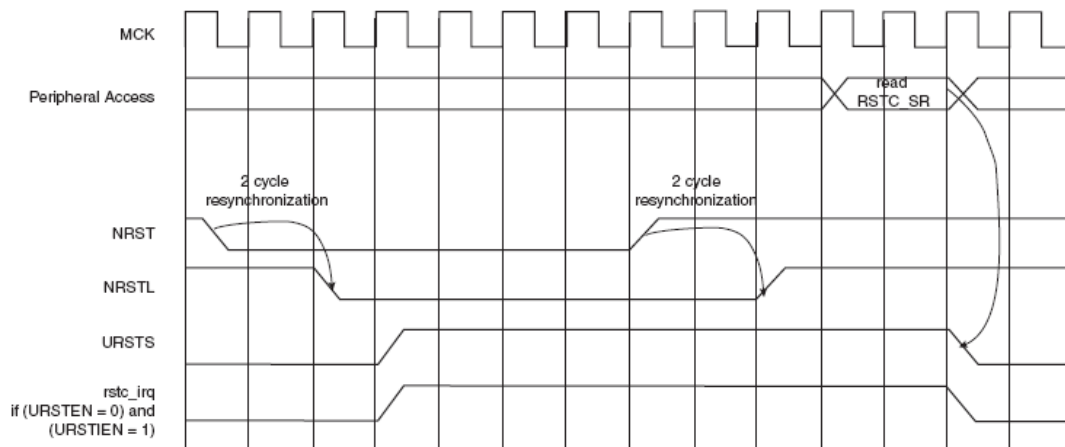
#### 14.3.5 复位控制器状态寄存器

复位控制器状态寄存器(RSTC\_SR)提供以下若干状态域：

- RSTTYP 域：此域给出了最后复位类型，如前段解释。
- SRCMP 位：此域表示软件复位命令在进进行中并且直到当前复位结束没有更远的软件复位应当被执行。此位在当前软件复位结束时自动清零。
- NRSTL 位：状态寄存器的 NRSTL 位给出了 NRST 引脚在每个 MCK 上升边沿采样的电平。
- URSTS 位：NRST 引脚 high-to-low 的转变置位 RSTC\_SR 寄存器的 URSTS 位。此转变还可以在主控时钟(MCK)上升沿被检测到(见图 14-11)。如果用户复位被使无效并且如果中断被 RSTC\_MR 寄存器中的 URSTIEN 位

使能，则 URSTS 位触发一个中断。读 RSTC\_SR 状态寄存器复位 URSTS 位并清零中断。

图 14-11 复位控制器状态和中断



#### 14.4 复位控制器(RSTC)用户接口

表 14-1 复位控制器 (RSTC) 寄存器映射

偏移量	寄存器	名称	访问	复位值	备份复位值
0x00	控制寄存器	RSTC_CR	只写	-	
0x04	状态寄存器	RSTC_SR	只读	0x0000_0001	0x0000_0000
0x08	模式寄存器	RSTC_MR	读/写	-	0x0000_0000

注意：1. RSTC\_SR 的复位值报告为一个通用复位或一个唤醒复位，取决于最后上升的电源。

##### 14.4.1 复位控制器控制寄存器

寄存器名称：RSTC\_CR

访问类型：只写

31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	EXTRST	PERRST	-	PROCRST

- PROCRST：处理器复位  
0=无效。  
1=如果 KEY 正确，复位处理器。
- OERRST：外设复位  
0 =无效。  
1 =如果 KEY 正确，复位外设。
- EXTRST：外部复位  
0=无效  
1=如果 KEY 正确，NRST 引脚有效。
- KEY：口令

应该写入值 0xA5。在此域写任何其他值将退出操作。

#### 14.4.2 复位控制器状态寄存器

寄存器名称: RSTC\_SR

访问类型: 只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	SRCMP	NRSTL
15	14	13	12	11	10	9	8
-	-	-	-	-	RSTTYP		
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	URSTS

- URSTS: 用户复位状态

0 = 从最近一次读 RST\_SR 到现在, 在 NRST 上无 high-to-low 边沿发生。

1 = 从最近一次读 RST\_SR 到现在 NRST 上至少有一次 high-to-low 转变被检测到。

- RSTTYP: 复位类型

报告最近一次处理器复位的原因。读此 RSTC\_SR 不复位此域。

表 1

RSTTYP			复位类型	备注
0	0	0	通用复位	VDDCORE 和 VDDBU 两个都上升
0	0	1	唤醒复位	VDDCORE 上升
0	1	0	看门狗复位	看门狗超时发生
0	1	1	软件复位	有软件命令处理器复位
1	0	0	用户复位	NRST 引脚检测到低电平

- NRSTL: NRST 引脚电平

在主控时钟(MCK)同步下的 NRST 引脚电平

- SRCMP: 进程中的软件复位命令

0 = 无软件命令正被复位控制器执行。复位控制器等待软件命令。

1 = 一个软件复位命令正被复位控制器执行。复位控制器忙。

#### 14.4.3 复位控制器模式寄存器

寄存器名称: RSTC\_MR

访问类型: 读/写

31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	ERSTL			
7	6	5	4	3	2	1	0
-	-	-	URSTIEN	-	-	-	URSTEN

- URSTEN: 用户复位使能

0=NRST 引脚上低电平的检测不会产生一个用户复位。

1=NRST 引脚上低电平的检测触发一个用户复位。

- URSTIEN: 用户复位中断使能

0=RSTC\_SR 中的 USRTS 位为 1 在 rstc\_irq 上无效。

1=如果 URSTEN=0, RSTC\_SR 中的 USRTS 位为 1 激活 rstc\_irq。

- ERSTL: 外部复位长度

此域定义外部复位长度。外部复位在  $2^{(ERSTL+1)}$  个慢时钟周期的时间里被激活。这允许有效延迟

时间编程为 60 微秒和 2 秒之间。

- KEY: 口令

应该写入值 0xA5。在此域写任何值都将退出操作。

www.mcuzone.cc

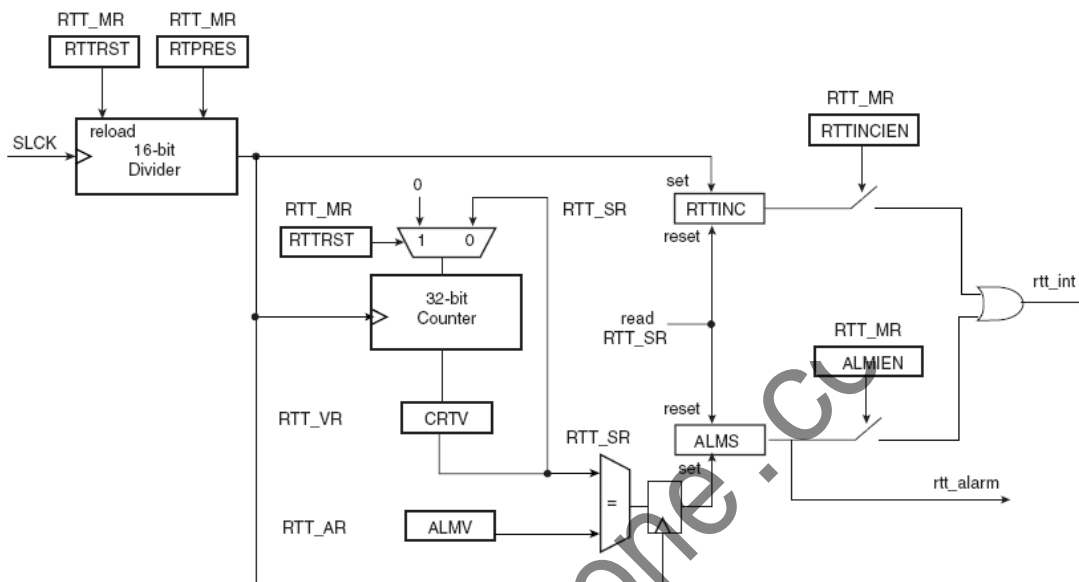
## 15 实时定时器(RTT)

### 15.1 概论

实时定时器围绕一个 32 位计数器构造并用于计数经历时间。它产生一个周期中断和/或根据一个编程了的值触发一个定时警报。

### 15.2 方块图

图 15-1 实时定时器



### 15.3 功能描述

实时定时器被用于计数经历时间。基于一个 32 位计数器构建，此计数器由一个可编程的 16 位分配器分频后的慢时钟提供时钟。其值可以在 RTT 模式寄存器 (RTT\_MR) 的 RTPRES 域中被编程。

将 RTPRES 编程为 0x00008000 相当于用一个 1Hz 信号(如果慢时钟是 32.738Hz) 馈送实时计数器。32 位计数器可以加到  $2^{32}$  秒，相当于 136 多年，然后转到 0。

实时定时器还可以被用作一个带较低时基的自由运行定时器。通过将 RTPRES 编程为 3 可以获得最好的精度。将 RTPRES 编程为 1 或 2 也是可以的，但有可能导致丢失状态事件，因为状态寄存器在读操作 2 个慢时钟周期后被清零。因此如果 RTT 被配置来触发一个中断，此中断在读 RTT\_SR 后 2 个慢时钟周期内产生。为了阻止若干中断处理程序的执行，中断必须在中断处理程序中禁用并当状态寄存器清零时重新使能。

实时定时器值 (CRTV) 可以在寄存器 RTT\_VR (实时值寄存器) 中任何时间被读取。由于此值可以从主控时钟异步更新，因此建议连续读两次该寄存器，如果活得相同的值再返回，以提高返回值的准确度。

计数器的当前值和报警寄存器 RTT\_AR (实时报警寄存器) 中写入的值作比较。如果计数器值符合报警值，则 RTT\_SR 中的 SLM 位被置位。复位后，

报警信号寄存器被置位到其最大值，相当于 0xFFFF\_FFFF。

RTT\_SR 中的 RTTINC 位在实时计数器每次递增后被置位。此位可被用于开启一个周期中断，当 RTPRES 编程为 0x8000，且慢时钟是 32.768Hz 的话，周期为 1 秒。

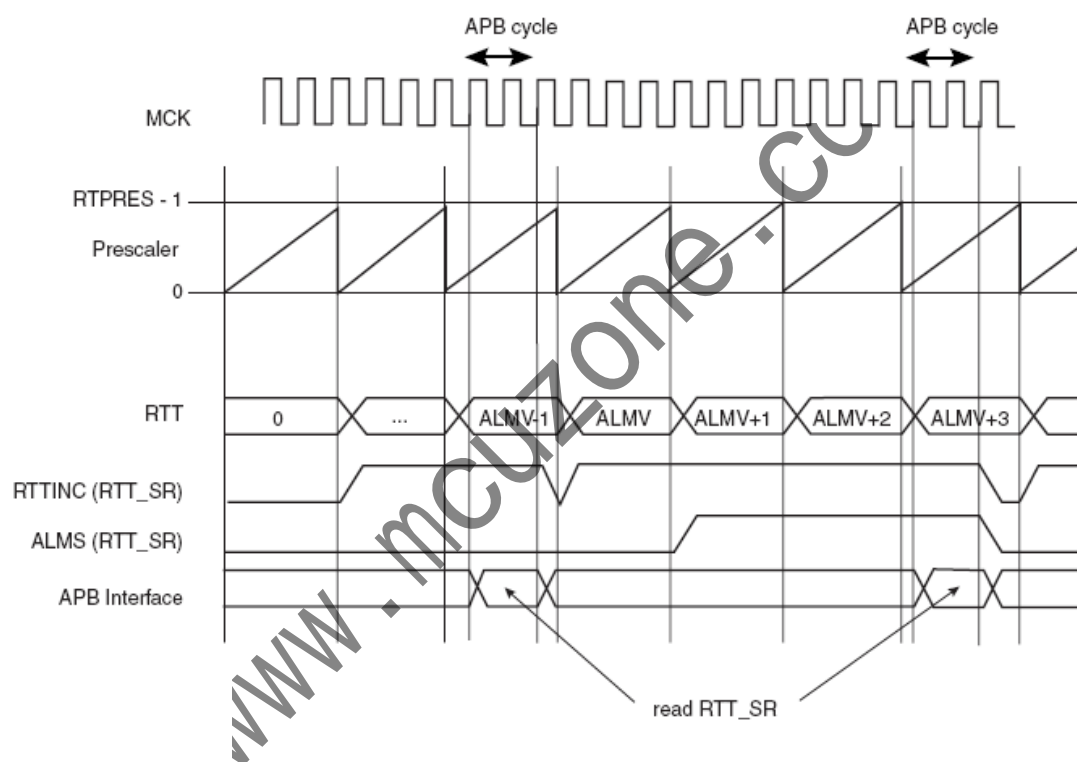
读 RTT\_SR 状态寄存器复位 RTTINC 和 ALMS 域。

写 RTT\_MR 中的 RTTRST 位会立即用新的编程值重装载和重开启时钟分频器。同时也复位 32 位计数器。

注意：由于慢时钟（SCLK）和系统时钟(MCK)间的不同步：

1. 在 RTT\_MR 寄存器中的 RTTRST 位写以后，计数器的重开启和 RTT\_VR 当前值寄存器的复位仅在 2 个慢时钟周期后是有效的。
2. 状态寄存器标志位在读 RTT\_SR（状态寄存器）后 2 个慢时钟周期复位。

图 15-2 RTT 计数



## 15.4 实时定时器(RTT)用户接口

### 15.4.1 寄存器映射

表 15-1 实时定时器寄存器映射

偏移量	寄存器	名称	访问	复位值
0x00	模式寄存器	RTT_MR	读/写	0x0000_8000
0x04	报警寄存器	RTT_AR	读/写	0xFFFF_FFFF
0x08	当前值寄存器	RTT_VR	只读	0x0000_0000
0x0C	状态寄存器	RTT_SR	只读	0x0000_0000

### 15.4.2 实时定时器模式寄存器

寄存器名称: RTT\_MR

访问类型: 读/写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	RTRST	RTTINCIEN	ALMIEN
15	14	13	12	11	10	9	8
RTPRES							
7	6	5	4	3	2	1	0
RTPRES							

- RTPRES: 实时定时器预分频值

定义实时计数器递增需要的 SLCK 周期数。RTPRES 被定义如下:

RTPRES=0: 预定标器周期等于  $2^{16}$

RTPRES≠0: 预定标器周期等于 RTPRES。

- ALMIEN: 报警中断使能

0=RTT\_SR 中的 ALMS 位在发生中断时无效。

1=RTT\_SR 中的 ALMS 位激活中断。

- RTTINCIEN: 实时定时器递增中断使能

0=RTT\_SR 中的 RTTINC 位发生中断时无效。

1=RTT\_SR 中的 RTTINC 激活中断。

- RTRRST: 实时定时器重启

- 1=用新编程值重装载并重启动时钟分频器。并复位 32 位计数器。

### 15.4.3 实时定时器报警寄存器

寄存器名称: RTT\_AR

访问类型: 读/写

31	30	29	28	27	26	25	24
ALMV							
23	22	21	20	19	18	17	16
ALMV							
15	14	13	12	11	10	9	8
ALMV							
7	6	5	4	3	2	1	0
ALMV							

- ALMV: 报警值

定义和实时定时器比较的报警值

### 15.4.4 实时定时器值寄存器

寄存器名称: RTT\_VR

访问类型: 只读

31	30	29	28	27	26	25	24
CRTV							
23	22	21	20	19	18	17	16
CRTV							
15	14	13	12	11	10	9	8
CRTV							
7	6	5	4	3	2	1	0
CRTV							

- CRTV: 当前实时值

返回实时定时器的当前值。

### 15.4.5 实时定时器状态寄存器

寄存器名称: RTT\_SR

访问类型: 只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	RTTINC	ALMS

- ALMS: 实时报警状态

0=从最后一次读 RTT\_SR 为止，实时报警未产生。

1=从最后一次读 RTT\_SR 为止，产生实时报警。

- RTTINC: 实时定时器递增

0 = 从最后一次读 RTT\_SR 为止，实时定时器未递增。

1 = 从最后一次读 RTT\_SR 为止，实时定时器递增。

[www.mcuzone.cc](http://www.mcuzone.cc)

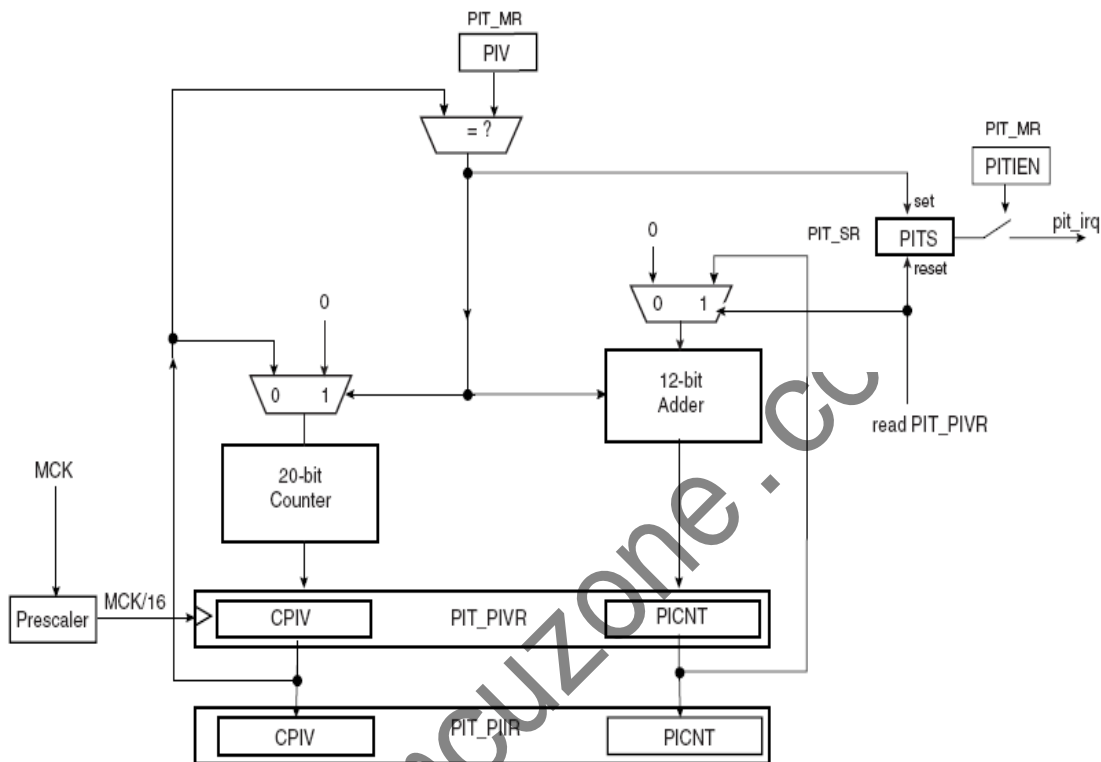
## 16 周期间断定时器(PIT)

### 16.1 概论

周期间断定时器 (PIT) 提供操作系统使用的程序调度中断。被设计来提供最大的准确度和最有效的管理，甚至对响应时间很长的系统。

### 16.2 方框图

图 16-1 周期间断定时器



### 16.3 功能描述

周期间断定时器为操作系统提供周期间断。

PIT 提供一个可编程的溢出计数器和一个 **reset-on-read** 功能部件。围绕两个计数器构建：一个 20 位 CPIV 计数器和一个 12 位 PICNT 计数器。两个计数器都工作在主控时钟的十六分之一。

第一个 20 位 CPIV 计数器从 0 递增到一个设置在模式寄存器 (PIT\_MR) 中的 PIV 域的可编程溢出值。当计数器 CPIV 达到此值，计数器复位到 0 并递增周期间断计数器，PICNT。如果中断被使能 (PIT\_MR 中的 PITIEN)，状态寄存器 (PIT\_SR) 中的状态位 PIT 置位并触发一个中断。

在 PIT\_MR 中写一个新 PIV 值不复位/重启计数器。

当 CPIV 和 PICNT 值通过读周期间断值寄存器 (PIT\_PIVR) 得到，溢出计数器 (PICNT) 复位并且 PITS 清零，以此方式响应中断。PICNT 的值给出了周期间断从最后一次读 PIT\_PIVR 到当前经过的周期数。

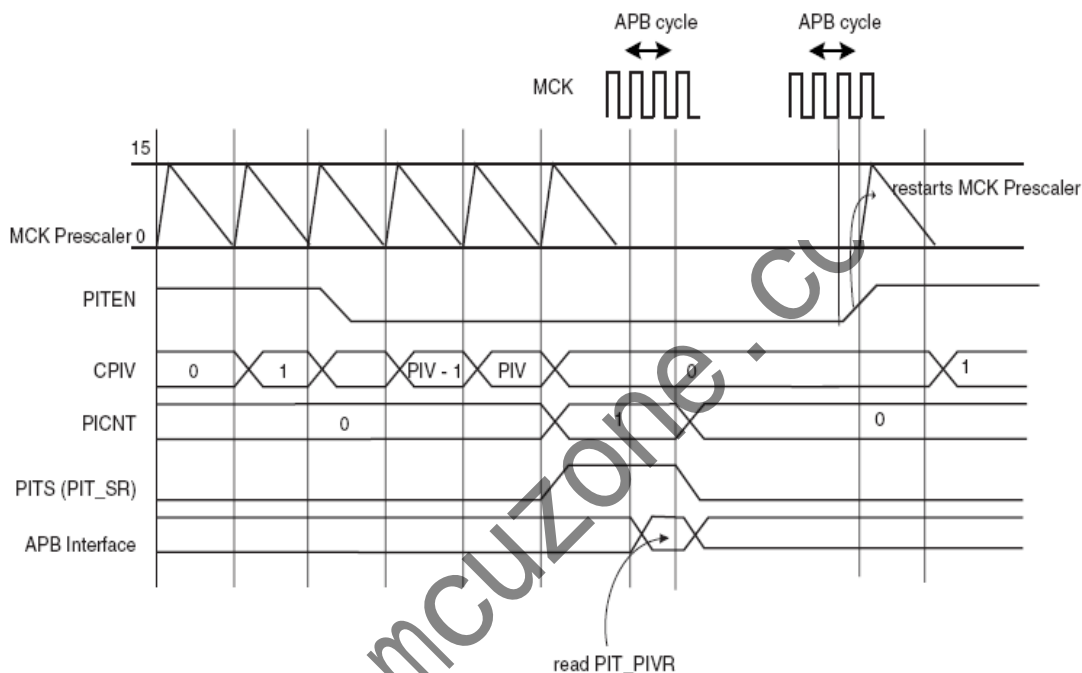
当 CPIV 和 PICNT 值通过读周期间断映射寄存器 (PIT\_PIIR) 获得，则不会

影响计数器 CPIV 和 PICNT，也不会影响 PITS 位。例如，一个分析器可以读没有清零任何未决的中断，然而一个定时器中断通过读 PIT\_PIVR 清零中断。

PIT 可能被 PIT\_MR 寄存器中的 PITEN 位使能或禁用（复位时禁止）。当 CPIV 值是 0，仅 PITEN 位变成有效。图 16-2 说明了 PIT 计数。PIT 位复位 (PITEN=0) 后，CPIV 继续计数直到达到 PIV 值，并接着复位。只有 PITEN 再次设置，PIT 才重启计数。

当内核进入调试状态，则 PIT 停止。

图 16-2 用 PITEN 使能/使失效 PIT



#### 16.4 周期间接定时器 (PIT) 用户接口

表 16-1 周期间断定时器(PIT)寄存器映射

偏移量	寄存器	名称	访问	复位值
0x00	模式寄存器	PIT_MR	读/写	0x000F_FFFF
0x04	状态寄存器	PIT_SR	只读	0x0000_0000
0x08	周期间断值寄存器	PIT_PIVR	只读	0x0000_0000
0x0C	周期间断映射寄存器	PIT_PIIR	只读	0x0000_0000

### 16.4.1 周期间断定时器模式寄存器

寄存器名称: PIT\_MR

访问类型: 读/写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	PITIEN	PITEN
23	22	21	20	19	18	17	16
-	-	-	-	PIV			
15	14	13	12	11	10	9	8
PIV							
7	6	5	4	3	2	1	0
PIV							

- PIV: 周期间断值

定义和周期间断定时器(CPIV)的主要的 20 位计数器比较的值。周期等于 (PIV+1)。

- PITEN: 周期间断定时器使能

0=当计数达到 PIV 值, 周期间断定时器被禁用。

1=周期间断定时器使能。

- PITIEN: 周期间断定时器中断使能

0=PIT\_SR 中的 PITS 位对中断无效

1=PIT\_SR 中的 PITS 位激活中断

### 16.4.2 周期间断定时器状态寄存器

寄存器名称: PIT\_SR

访问类型: 只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	PITS

- PITS: 周期间断定时器状态

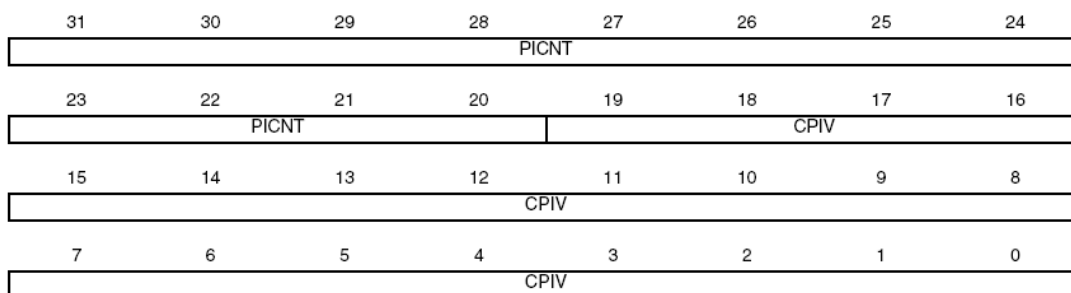
0=周期间断定时器从最近一次读 PIT\_PIVR 到当前未计数到达 CPIV。

1=周期间断定时器从最近一次读 PIT\_PIVR 到当前已计数到达 CPIV。

### 16.4.3 周期间断定时器值寄存器

寄存器名称: PIT\_PIVR

访问类型: 只读



读此寄存器清零 PIT\_SR 中的 PITS 位。

- CPIV: 当前周期间断值

返回周期间断定时器的当前值。

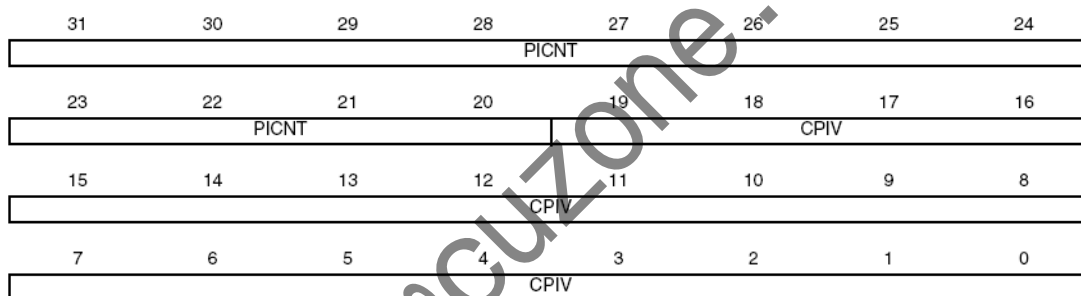
- PICNT: 周期间断计数器

返回从最近一次读 PIT\_PIVR 到当前周期间断发生的次数。

#### 16.4.4 周期间断定时器映射寄存器

寄存器名称: PIT\_PIRR

访问类型: 只读



- CPIV: 当前周期间断值

返回周期间断定时器的当前值

- PICNT: 周期间断计数器

返回从最近一次读 PIT\_PIVR 到当前周期间断发生的次数。

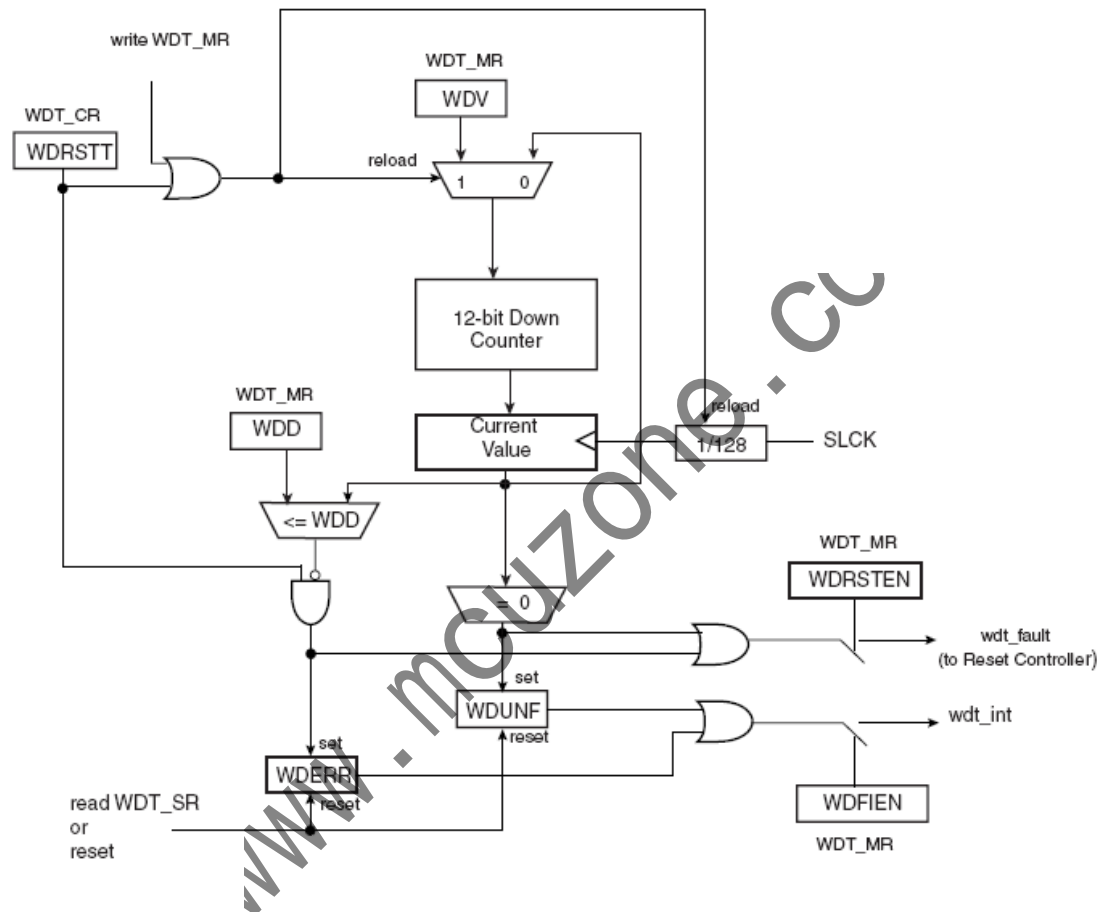
## 17 看门狗定时器 (WDT)

### 17.1 概论

如果用户程序进入死锁状态，看门狗定时器可以防止系统锁定 (lock-up)。它以一个 12 位的递减计数器组成。此计数器允许一个达 16 秒 (慢时钟 32.768KHz) 的看门狗周期。它还可仅产生一个通用复位或一个处理器复位。另外，当处理器在调试模式或空闲模式时可以停止看门狗。

### 17.2 方框图

图 17-1 看门狗定时器方块图



### 17.3 功能描述

如果用户程序进入死锁状态，看门狗定时器可以防止系统锁定 (lock-up)。看门狗由 VDDCORE 供电。当处理器复位时将加载初始值。

看门狗围绕一个 12 位递减计数器构造，此计数器用定义在模式寄存器 (WDT\_MR) 的 WV 域中的值装载。看门狗定时器用慢时钟除以 128 来作为时钟，最大看门狗周期为 16 秒 (用一个典型的 32.768KHz 的慢时钟)。处理器复位后，WV 的值为 0xFFFF，相当于用外部复位产生使能 (一个备份复位后，WDRSTEN 域为 1) 的最大值。这就意味着看门狗复位后缺省就运行，例如上电复位。如果用户不想用它就必须禁用它 (通过置位 WDT\_MR 中的 WDDIS 位) 或必须重编程它来满足走大看门狗周期的应用需求。看门狗模式寄存器 (WDT\_MR) 仅可以被写一次。仅能用一个处理器复位复

位。写 WDT\_MR 寄存器将用新编程的模式参数重新装载定时器。

普通操作中，用户通过一定时间间隔向控制寄存器的 WDRSTT 位写 1，在定时器下溢产生前重新装载看门狗。看门狗计数器接着立即从 WDT\_MR 重新装载并重启，慢时钟 128 分频器被复位和重启。WDT\_CR 寄存器被写保护。如果没有适当的硬编码密钥（hard-coded key），写 WDT\_CR 则无效。如果产生一个下溢，如果 WDRSTEN 位在模式寄存器被置位则复位控制器的“wdt\_fault”信号被激活。此外，WDUNF 位在看门狗状态寄存器(WDT\_SR)中被置位。

为阻止软件死锁时连续的触发看门狗，当看门狗计数器在 0 和 WDD 间的窗口，看门狗必须重新装载，WDD 在看门狗模式寄存器 WDT\_MR 中定义。

当看门狗计数器在 WDV 和 WDD 间任何重启看门狗的尝试都导致看门狗错误，即时看门狗在使无效状态。WDERR 位在 WDT\_SR 中刷新，到复位控制器的“wdt\_fault”信号是有效的。

注意此功能部件可以通过编程一个大于或等于 WDV 值的 WDD 值来禁用。在这样的配置中，重启看门狗定时器在整个 [0;WDV] 范围内是允许的并且不会出错。这是复位后的缺省配置（WDD 和 WDV 值相等）。

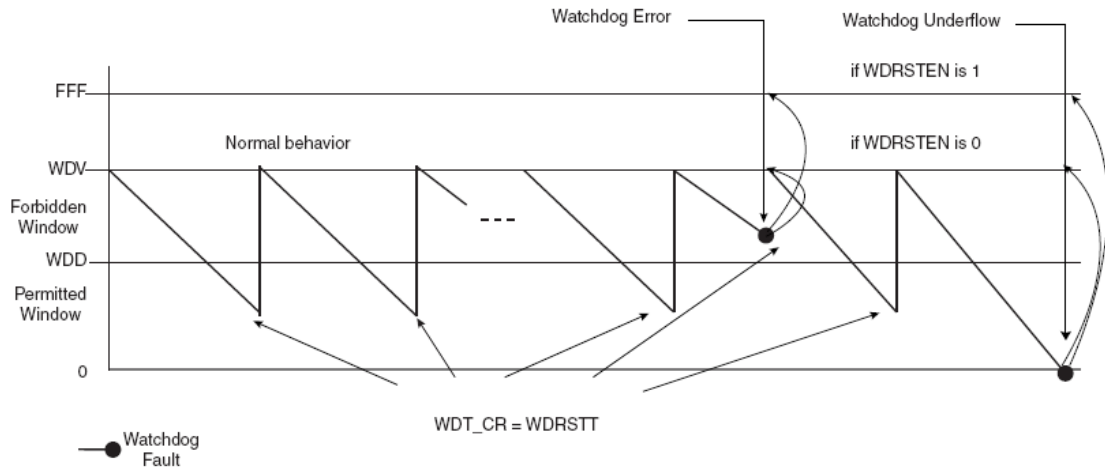
如果 WDFIEN 位在模式寄存器中置位，状态位 WDUNF（看门狗下溢）和 WDERR（看门狗错误）触发一个中断。如果 WDRSTEN 位置位，就像已在复位控制器编程器 Datasheet 中所解释的那样，则“wdt\_fault”信号在复位控制器导致看门狗复位。此情况下，处理器和看门狗定时器被复位，WDERR 和 WDUNF 标志位被复位。

如果复位发生或如果 WDT\_SR 已读，状态位被复位，中断被清零，并且到复位控制器的“wdt\_fault”信号被使无效。

写 WDT\_MR 重新装载和重启递减计数器。

当处理器在调试状态或空闲模式，计数器可能被停止，取决于对 WDT\_MR 中的 WDIDLEHLT 和 WDDBGHLR 位的编程值。

图 17-2 看门狗行为



www.mcuzone.com

## 17.4 看门狗定时器 (WDT) 用户接口

表 17-1 看门狗定时器(WDT)寄存器映射

偏移量	寄存器	名称	访问	复位值
0x00	控制寄存器	WDT_CR	只写	-
0x04	模式寄存器	WDT_MR	一次性读/写	0x3FFF_2FFF
0x08	状态寄存器	WDT_SR	只读	0x0000_0000

### 17.4.1 看门狗定时器控制寄存器

寄存器名称: WDT\_CR

访问类型: 只写

31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	WDRSTT

- WDRSTT: 看门狗重启

0: 无效。

1: 重启看门狗。

- KEY: 口令

应写入 0xA5.在此域写任何其他值异常结束写操作。

### 17.4.2 看门狗定时器模式寄存器

寄存器名称：WDT\_MR

访问类型：一次性读/写

31	30	29	28	27	26	25	24
-	-	WDIDLEHLT	WDDBGHLT	WDD			
23	22	21	20	19	18	17	16
WDD							
15	14	13	12	11	10	9	8
WDDIS	WDRPROC	WDRSTEN	WDFIEN	WDV			
7	6	5	4	3	2	1	0
WDV							

- **WDV: 看门狗计数器值**

定义装载于 12 位看门狗计数器中的值

- **WDFIEN: 看门狗故障中断使能**

0: 看门狗故障（下溢或错误）对中断无效

1: 看门狗故障（下溢或错误）激活中断

- **WDRSTEN: 看门狗复位使能**

0: 看门狗故障（下溢或错误）对复位无效

1: 看门狗故障（下溢或错误）触发一个看门狗复位

- **WDRPROC: 看门狗复位处理器**

0: 如果 WDRSTEN 是 1，看门狗故障（下溢或错误）激活所有的复位

1: 如果 WDRSTEN 是 1，看门狗故障（下溢或错误）激活处理器复位

- **WDD: 看门狗增量**

定义重装载看门狗定时器的允许范围

如果看门狗定时器值小于或等于 WDD，用 WDRSTT = 1 写 WDT\_CR 重启定时器。

如果看门狗定时器值大于 WDD，用 WDRSTT = 1 写 WDT\_CR 导致看门狗错误。

- **WDDBGHLT: 看门狗调试停止**

0: 当处理器处在调试状态，看门狗运行

1: 当处理器处在调试状态，看门狗停止

- **WDIDLEHLT: 看门狗空闲停止**

0: 当系统处于空闲模式，看门狗运行

1: 当系统处于空闲模式，看门狗停止

- **WDDIS: 看门狗使无效**

0: 使能看门狗定时器

1: 是无效看门狗定时器

### 17.4.3 看门狗定时器状态寄存器

寄存器名称: WDT\_SR

访问类型: 只读

1

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	WDERR	WDUNF

- **WDUNF: 看门狗下溢**

0: 从最近一次读 WDT\_SR 到当前无看门狗下溢产生。

1: 从最近一次读 WDT\_SR 到当前至少一个看门狗下溢产生

- **WDERR: 看门狗错误**

0: 从最近一次读 WDT\_SR 到当前无看门狗错误产生

1: 从最近一次读 WDT\_SR 到当前至少一个看门狗错误产生

www.mcuzone.cn

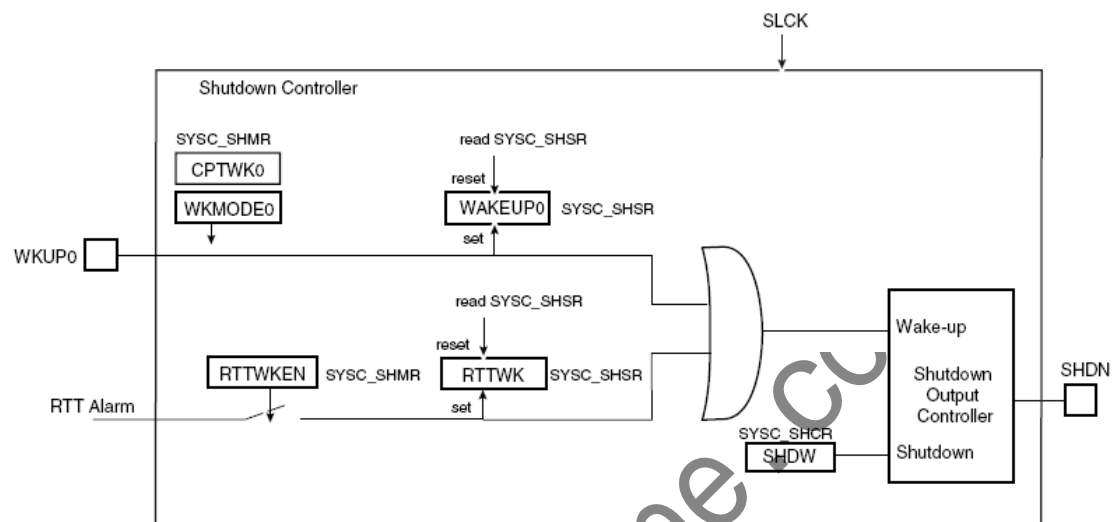
## 18 掉电控制器 (SHDWC)

### 18.1 描述

掉电控制器控制电源 VDDIO 和 VDDCORE 和防抖动 (debounced) 输入口线上的唤醒检测。

### 18.2 方框图

图 18-1 掉电控制器方框图



### 18.3 I/O 口线描述

表 18-1 I/O 口线描述

名称	描述	类型
WKUP0	唤醒 0 输入	输入
SHDN	关机输出	输出

### 18.4 产品相关

#### 18.4.1 电源管理

掉电控制器由慢时钟提供时钟。掉电管理控制器在关机控制器运行时无效。

#### 18.5 功能描述

掉电控制器管理主电源。为实现此目的，其由 VDDBU 供电并且管理唤醒输入引脚和一个输出引脚，SHDN。

一个典型应用是把 SHDN 引脚连接到 DC/DC 转换器的输入来给系统提供主电源，特别是 VDDCORE 和/或 VDDIO。唤醒输入 (WKUP0) 连接到任何按钮或唤醒系统的信号。

软件可以通过向掉电控制寄存器 (SHDW\_CR) 的 SHDW 位写 1 来控制 SHDN 引脚。向 SHDN\_CR 写入后只需 2 个慢时钟周期即进入掉电。此寄存器受口令保护，所以必须有正确的口令才能向其写入值。如果正确写入，系统即进入掉电状态。

WKUP0 上的电平改变可以用来唤醒。唤醒在掉电模式寄存器 (SHDN\_MR) 中进行配置。转换检测器可以被编程为检测正/负边沿或 WKUP0 上的任何电平改变。此检测也可以禁用。程序可通过定义 WKMODE0 来设置。

而且，防抖动电路可以编程为 WKUP0 所使用。防抖动电路过滤 WKUP0 上比编程在 HDW\_MR 寄存器的 CPTWK0 中的 16SLCK 的周期要短的脉冲。如果某个引脚上的电平改变被检测到，则计数器启动。当计数器达到在对应的域中编程值，CPTWK0，则 SHDN 引脚被释放。如果在计数器到达相应值之前一个新的输入改变被检测到，则计数器被停止并清零。状态寄存器 (SHDW\_SR) 的位 WAKEUP0 报告 WKUP0 上的编程事件的检测，并在读 SHDW\_SR 后复位。

掉电控制器可以被编程为用 RTT 警报 (RTT 警报上升沿的检测和 SLCK 同步) 激活唤醒。这是通过写 SHDW\_MR 寄存器的 RTTWKEN 域完成。当使能，RTT 警报状态报告在 SHDW\_SR 状态寄存器的 RTTWK 域。当读 SHDW\_SR 后复位。当使用 RTT 警报来唤醒系统，用户必须保证 RTT 警报状态标志在系统掉电前清零。否则，将没有状态标志位的上升沿会被检测到并导致唤醒失败。

www.mcuzone.cn

## 18.6 掉电控制器(SHDWC)用户接口

### 18.6.1 寄存器映射

表 18-2 掉电控制器(SHDWC)寄存器

偏移量	寄存器	名称	访问	复位值
0x00	掉电控制寄存器	SHDW_CR	只写	-
0x04	掉电模式寄存器	SHDW_MR	读写	0x0000_0103
0x18	掉电状态寄存器	SHDW_SR	只读	0x0000_0000

### 18.6.2 掉电控制器寄存器

寄存器名称: SHDW\_CR

访问类型: 只写

31	30	29	28	27	26	25	24
KEY							
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	SHDW

- SHDW: 掉电命令

0 = 无效

1 = 如果 KEY 正确, 激活 SHDN 引脚。

- KEY: 口令

应被写为 0xA5。在此域中写任何值将退出操作。

### 18.6.3 掉电模式寄存器

寄存器名称: SHDW\_MR

访问类型: 读/写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	RTTWKEN
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
CPTWK0				-	-	WKMODE0	

- WKMODE 0 : 唤醒模式 0

WKMODE[1: 0]		唤醒输入转换选择
0	0	无, 在唤醒输入上不执行检测
0	1	低到高电平
1	0	高到低电平
1	1	上两列电平转变

- CPTWK0: Wake-up0 上的计数器

定义 16 慢时钟周期数, 对应输入引脚上的电平检测在唤醒事件发生前应该持续。因为 WKUP0 的内部同步化, SHDN 引脚在 WKUP 上事件发生后经过  $(CPTWK*16+1)$  个慢时钟周期被释放。

- RTTWKEN: 实时定时器唤醒使能

0 =RTT 警报信号在掉电控制器上无效

1=RTT 警报信号使得 SHDN 引脚释放。

#### 18.6.4 掉电状态寄存器

寄存器名称：SHDW\_SR

访问类型：只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	RTTWK
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	WAKEUP0

- WAKEUP0: Wake-up0 状态

0=从最近一次读 SHDW\_SR 到当前对应的唤醒输入上无唤醒事件发生。

1=从最近一次读 SHDW\_SR 到当前对应的唤醒输入上至少一次唤醒事件发生。

- RTTWK: 实时定时器唤醒

0 =从最近一次读 SHDW\_SR 到当前没有 RTT 唤醒事件发生。

1=从最近一次读 SHDW\_SR 到当前至少有一次 RTT 唤醒事件发生。

www.mcuzone.com

## 19 总线矩阵

### 19.1 概论

总线矩阵实现一个多层 AHB，基于 AHB-Lite 协议，并使能系统中多层 AHB 主控和从控间的并行访问路径，以此方式增加总带宽。总线矩阵互联了 5 个 AHB 主控和 5 个 AHB 从控。总线矩阵用户接口和 ARM 先进的外围总线兼容并且提供 5 个特殊功能寄存器(MATRIX\_SFR)，这些寄存器允许总线矩阵支持特定应用部件。

### 19.2 存储器映射

总线矩阵给每个 AHB 主控接口提供一个解码器。解码器给每个 AHB 主控若干存储器映射。取决于产品，每个存储器区被指定为若干从控。当用不同的 AHB 从控（就是，外部 RAM，内部 ROM，内部 Flash,等）在相同的地址启动变得可能。

总线矩阵用户接口提供一个主控配置寄存器（MATRIX\_MCFG）以对每个主控独立的执行一重映射操作。

### 19.3 特殊总线授权技术

总线矩阵提供一些推测性总线授权技术来提前预测一些主控的访问请求。此技术减小了初次访问的延迟时间。总线授权技术给每个从控设置一个缺省的主控。在当前访问的结尾，若无其他请求待定，则从控仍然连接其关联的缺省主控。一个从控可以关联 3 种缺省的主控：无缺省主控，最近一次访问主控和固定缺省主控。

#### 19.3.1 无缺省主控

当前访问的结尾，若无其他请求待定，从控被从所有主控切断。无缺省主控适用于低功耗模式。

#### 19.3.2 最近一次访问主控

当前访问的结尾，若无其他请求待定，从控仍然连接最近执行访问请求的主控。

#### 19.3.3 固定缺省主控

当前访问的结尾，若无请求待定，从控仍然连接其固定缺省主控。不像最近访问主控，固定主控不改变 除非用户通过软件操作改变它。

从一种缺省主控变到另一种，总线矩阵用户接口提供 5 个从控配置寄存器，每个从控一个，给每个从控设置缺省主控。从控配置寄存器包含 2 个域：DEFMSTR\_TYPE 和 FIXED\_DEFMSTR。2 位 DEFMSTR\_TYPE 标志位选择缺省主控类型（无缺省主控，最近访问主控，固定缺省主控）然而 3 位 FIXED\_DEFMSTR 标志位选择一个固定缺省主控，假设 DEFMSTR\_TYPE 被设置为一个固定缺省主控。见 133 页“总线矩阵用户接口”。

### 19.4 仲裁

总线矩阵提供仲裁功能，当冲突情况发生，此功能减小延迟时间，就是说，当两个或更多主控试图在同一时间访问相同的从控。总线矩阵仲裁方法用轻微的改进了的循环算法，依赖于从控的从控配置寄存器中的参数，此算法对第一次访问特定主控授权总线。

有三种循环算法类型：

- 无缺省主控的循环算法
- 最近访问主控的循环算法
- 固定缺省主控的循环算法

#### 19.4.1 无缺省主控的循环仲裁

这是被总线矩阵仲裁使用的主算法。在一个纯循环算法方式，它允许总线矩阵从不同的主控调度请求到相同的从控。在当前访问的结尾，若无其他请求待定，从控从所有的主控断开。此配置对第一次访问产生一个延迟时间周期。无缺省主控的仲裁可以被用作执行突发操作较多的主控。

#### 19.4.2 最近访问主控的循环仲裁

这是一个被总线矩阵仲裁使用的边缘循环算法。它允许总线矩阵对最近一次访问从控的主控缩短一个延迟时间周期。事实上，在当前传输的结尾，若无其他主控请求待定，从控仍然连接至执行访问的最近一次主控。如果其他主控访问同样的从控，这些非特权主控仍然可获一个延迟时间周期。此技术可被用于主要执行单一访问的主控。

#### 19.4.3 固定缺省主控的循环仲裁

这是一个边缘的循环算法。它允许总线矩阵仲裁对从控的固定主控缩短一个延迟时间周期。在当前访问的结尾，从控仍然连接它的固定缺省主控。通过此固定缺省主控尝试的任何请求不会导致任何延迟时间，然而其他非特权主控仍获得一个延迟周期。此技术可被用作主要执行单一访问的主控。

### 19.5 总线矩阵用户接口

表 19-1 寄存器映射

偏移量	寄存器	名称	访问	复位值
0x0000	主控配置寄存器	MATRIX_MCFG	只写	0x00000000
0x0004	从控配置寄存器 0	MATRIX_SCFG0	读/写	0x00000010
0x0008	从控配置寄存器 1	MATRIX_SCFG0	读/写	0x00000010
0x000C	从控配置寄存器 2	MATRIX_SCFG0	读/写	0x00000010
0x0010	从控配置寄存器 3	MATRIX_SCFG0	读/写	0x00000010
0x0014	从控配置寄存器 4	MATRIX_SCFG0	读/写	0x00000010
0x0018-0x0020	保留	-	-	-
0x0024	MATRIX TCM 配置寄存器	MATRIX_TCR	读/写	0x00000000
0x0028-0x002C	保留	-	-	-
0x0030	EBI 片选分配寄存器	EBI_CSA	读/写	0x00000000
0x0034	USBPAD 上拉控制寄存器	USB_PUCR	读/写	0x00000000

### 19.5.1 总线矩阵主控配置寄存器

寄存器名称: MATRIX\_MCFG

访问类型: 只写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	RCB1	RCB0

- RCBx: AHB 主控 x 重映射命令位

0: 无效

1: 此命令位用于取反设置: 写 1 交替的取消和使能主控 x 存储寻址从控的重映射。

www.mcuzone.com

### 19.5.2 总线矩阵从控配置寄存器

寄存器名称: MATRIX\_SCFG0...MATRIX\_SCFG4

访问类型: 读/写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	FIXED_DEFMSTR			DEFMSTR_TYPE	
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
SLOT_CYCLE							

- **SLOT\_CYCLE**: 对一个突发访问允许的最大周期数

当一个突发访问到达 **SLOT\_CYCLE** 界限，此访问可能被另外试图访问此从控的主控打断。

该限制被设置用于避免当突发周期很长时锁定速度非常慢的设备。

此界限不应该很小。一个不合理的小值会破坏每个突发访问并使得总线矩阵没有执行任何数据传送而花时间在仲裁。对 **SLOT\_CYCLE** 而言 16 个周期是个合理的值。

- **DEFMASTR\_TYPE**: 缺省主控类型

0: 无缺省主控

在当前从控访问结尾，若无其他主控请求等待，从控切断所有主控连接。

这对一个脉冲串的第一次传输将导致一个周期延迟。

1: 上一个缺省主控

在当前从控访问结尾，若无其他主控请求等待，从控仍然连接访问过的上一个主控。

这导致当上一个主控试图再次访问从控时没有一个周期延迟。

2: 固定缺省主控

在当前从控访问结尾，若无其他主控请求等待，从控同在 **FIXED\_DEFMSTR** 寄存器中有其索引的固定主控连接。

这导致当固定主控试图再次访问从控而没有一个周期延迟。

- **FIXED\_DEFMSTR**: 缺省主控的固定索引

这是对此值固定缺省主控的索引

### 19.5.3 总线矩阵 TCM 配置寄存器

寄存器名称: MATRIX\_TCR

访问类型: 读/写

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
DTCM_SIZE				ITCM_SIZE			

- ITCM\_SIZE: ITCM 使能存储器块容量  
0000 0KB(无 ITCM 存储器)  
0101: 16KB  
0110: 32KB  
0111: 64KB  
其他: 保留
- DTCM\_SIZE: DTCM 使能存储器块容量  
0000 0 KB (无 DTCM 存储器)  
0101: 16 KB  
0110: 32 KB  
0111: 64 KB  
其他: 保留

www.mcuzone.com

### 19.5.4 EBI 片选分配寄存器

寄存器名称: EBI\_CSA

访问类型: 读/写

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	EBI_DBPUC
7	6	5	4	3	2	1	0
-	-	EBI_CS5A	EBI_CS4A	EBI_CS3A	-	EBI_CS1A	-

- EBI\_CS1A: EBI 片选 1 指定  
0=EBI 片选 1 被分配到静态存储器控制器  
1=EBI 片选 1 被分配到 SDRAM 控制器
- EBI\_CS3A: EBI 片选 3 指定  
0=EBI 片选 3 仅分配到静态存储器控制器并且 EBI\_NCS3 按 SMC 定义运转。  
1=EBI 片选 3 被分配到静态存储器控制器并且 SmartMedia 逻辑被激活。
- EBI\_CS4A: EBI 片选 4 指定  
0=EBI 片选 4 仅分配到静态存储器控制器并且 EBI\_NCS4 按 SMC 定义运转。  
1=EBI 片选 4 被分配到静态存储器控制器并且 CompactFlash (第一个槽) 被激活。
- EBI\_CS5A: EBI 片选 5 指定  
0=EBI 片选 5 仅分配到静态存储器控制器并且 EBI\_NCS5 按 SMC 定义运转。  
1=EBI 片选 5 被分配到静态存储器控制器并且 CompactFlash (第二个槽) 被激活。
- EBI\_DBPUC: EBI 数据总线上拉配置  
0=EBI D0-D15 数据总线位被内部上拉到 VDDIOM0 电源。  
1=EBI D0-D15 数据总线位不内部上拉。

### 19.5.5 USB PAD 上拉控制寄存器

寄存器名称: USB\_PUCR

访问类型: 读/写

复位值: 0x0000\_0000

31	30	29	28	27	26	25	24
Reserved	UDP_PUP_ON	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

0: PAD 上拉使无效

1: PAD 上拉使能

www.mcuzone.cc

## 20 外部总线接口 (EBI)

### 20.1 概论

外部总线接口 (EBI) 被设计来确保若干外部设备和基于 ARM 设备的嵌入式存储器控制器之间数据成功传输。静态存储器和 SDRAM 控制器都被 EBI 上的外部存储器控制器所支持。这些外部存储器控制器可以处理若干类型的外部存储器和外设, 像 SRAM, PROM, EPROM, EEPROM, Flash, 和 SDRAM。

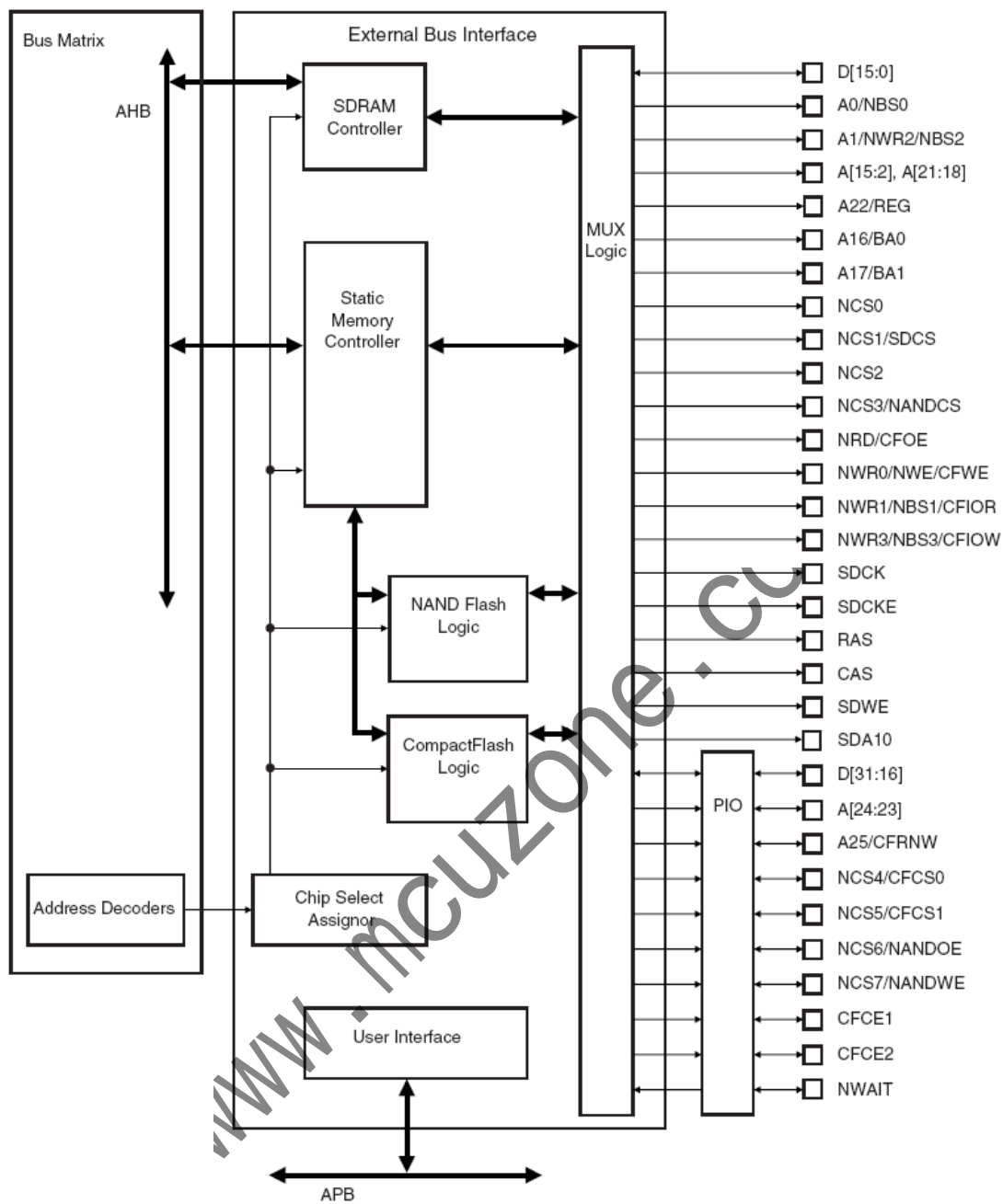
EBI 还通过集成的电路支持 CompactFlash 和 NAND Flash 协议, 使得外部元件大量减少。此外, EBI 用多达八个外部设备处理数据传输, 每个外部设备被指定到八个被嵌入式存储控制器定义的地址空间。数据传输通过一个 16 位或 32 位数据总线执行, 地址总线多达 26 位, 多达八个片选块 (NCS[7:0]) 和若干控制引脚, 这些引脚通常在不同外部存储器控制器间多路复用。

### 20.2 方框图

图 20-1 展示了外部总线接口的组织。

www.mcuzone.cn

图 20-1 展示了外部总线接口的组织



### 20.3 I/O 口线描述

20.4 表 20-1 I/O 口线描述

名称	功能	类型	有效电平
<b>EBI</b>			
D0-D31	数据总线	I/O	
A0-A25	地址总线	输出	
NWAIT	外部等待信号	输入	低
<b>SMC</b>			
NCS0-NCS7	片选线	输出	低
NWR0-NWR3	写信号	输出	低
NRD	读信号	输出	低
MWE	写使能	输出	低
MBS0-NBS3	字节屏蔽信号	输出	低
<b>EBI for CompactFlash Support</b>			
CFCE1-CFCE2	CompactFlash 芯片使能	输出	低
CFOE	CompactFlash 输出使能	输出	低
CFWE	CompactFlash 写使能	输出	低
CFIOR	CompactFlash I/O 读使能	输出	低
CFIOW	CompactFlash I/O 写使能	输出	低
CFRNW	CompactFlash 读非写信号	输出	
CFCS0-CFCS1	CompactFlash 片选线	输出	低
<b>EBI for NAND Flash Support</b>			
NANDCS	NAND Flash 片选线	输出	低
NANDOE	NAND Flash 输出使能	输出	低
NANDWE	NAND Flash 写使能	输出	低
<b>SDRAM 控制器</b>			
SDCK	SDRAM 时钟	输出	
SDCKE	SDRAM 时钟使能	输出	高
SDCS	SDRAM 控制器片选线	输出	低
BA0-BA1	存储体选择	输出	
SDWE	SDRAM 写使能	输出	低
RAS-CAS	行和列信号	输出	低
NWR0-NWR3	写信号	输出	低
NBS0-NBS3	字节屏蔽信号	输出	低
SDA10	SDRAM A10 线	输出	

若干信号的连接不是直接通过 MUX 逻辑并依赖于此刻在使用的存储控制器。

142 页表 20-2 详述了两个存储控制器和 EBI 引脚的连接。

表 20-2 EBI 引脚和存储控制器 I/O 口线的连接

EBI 引脚	SDRAMC I/O 口线	SMC I/O 口线
NWR1/NBS1/CFIOR	NBS1	NWR1/NUB
A0/NBS0	不支持	SMC_A0/NLB
A1/NBS2/NWR2	不支持	SMC_A1
A[11:2]	SDRAMC_A[9:0]	SMC_A[11:2]
SDA10	SDRAMC_A10	不支持
A12	不支持	SMC_A12
A[14:13]	SDRAMC_A[12:11]	SMC_A[14:13]
A[25:15]	不支持	SMC_A[25:15]
D[31:16]	D[31:16]	D[31:16]
D[15:0]	D[15:0]	D[15:0]

## 20.4 应用举例

### 20.4.1 硬件接口

表 20-3 和表 20-4 详述了对每个存储控制器被用在 EBI 引脚和外部设备间的连接。

表 20-3 EBI 引脚和外部静态设备连接

引脚	接口设备的引脚					
	8 位静态设备	2×8 位静态设备	16 位静态设备	4×8 位静态设备	2*16 位静态设备	32 位静态设备
控制器	SMC					
D0-D7						
D8-D15						
D16-D23	D0-D7	D0-D7	D0-D7	D0-D7	D0-D7	D0-D7
D24-D31	-	D8-D15	D8-D15	D8-D15	D8-D15	D8-D15
A0/NBS0	-	-	-	D16-D23	D16-D23	D16-D23
A1/NWR2/NBS2	A0	-	NLB	-	NLB	BE0
A2-A25	A1	A0	A0	WE	NBL	BE2
NCS0	A[2:25]	A[1:24]	A[1:24]	A[0:23]	A[0:23]	A[0:23]
NCS1/SDCS	CS	CS	CS	CS	CS	CS
NCS2	CS	CS	CS	CS	CS	CS
NCS3/NANDCS	CS	CS	CS	CS	CS	CS
NCS4/CFCS0	CS	CS	CS	CS	CS	CS
NCS5/CFCS1	CS	CS	CS	CS	CS	CS
NCS6/NAND0E	CS	CS	CS	CS	CS	CS
MCS7/NANDWE	CS	CS	CS	CS	CS	CS
NRD/CFOE	OE	OE	OE	OE	OE	OE
NWR0/NEW	WE	WE	WE	WE	WE	WE
NWR1/NBS1	-	WE	NUB	WE	NUB	BE1
NWR3/NBS3	-	-	-	WE	NUB	BE3

- 注意：
1. NWR1 使能高位字节写。NWR0 使能低位字节写。
  2. NWRx 使能对应字节 x 写。（x=0,1,2 或 3）
  3. NBS0 和 NBS1 分别使能低 16 位字的低位和高位字节

4. NBS2 和 NBS3 分别使能高 16 位字的低位和高位字节
5. BEx:字节 x 使能 (x=0, 1, 2, 或 3)

表 20-4EBI 引脚和外部设备连接

引脚	接口的设备引脚			
	SDRAM	CompactFlash	CompactFlash True IDE Mode	NAND Flash
<b>控制器</b>	<b>SDRAMC</b>	<b>SMC</b>		
D0 - D7	D0 - D7	D0 - D7	D0 - D7	I/O0-I/O7
D8 - D15	D8 - D15	D8 - D15	D8 - D15	I/O8-I/O15
D16 - D31	D16 - D31	A0	A0	
A0/NBS0	DQM0	A1	A1	
A2 - A10	A[0:8]	A[2:10]	A[2:10]	
A11	A9			
SDA10	A10			
A12				
A13 - A14	A[11:12]			
A15				
A16/BA0	BA0			
A17/BA1	BA1			
A18 - A20				
A21				CLE
A22		REG	REG	ALE
A23 - A24				
A25		CFRNW	CFRNW	
NCS0				
NCS1/SDCS	CS			
NCS2				CE
NCS3/NANDCS		CFCS0	CFCS0	
NCS4/CFCS0		CFCS1	CFCS1	
NCS5/CFCS1				RE
NCS6/NANDOE				WE
NCS7/NANDWE		-		
NRD/CFOE		OE	-	
NWR0/NWE/CFWE		WE	WE	
NWR1/NBS1/CFIOR	DQM1	IOR	IOR	
NWR3/NBS3/CFIOW	DQM3	IOW	IOW	
CFCE1		CE1	CS0	
CFCE2		CE2	CS1	
SDCK	CLK			
SDCKE	CKE			
RAS	RAS			
CAS	CAS			

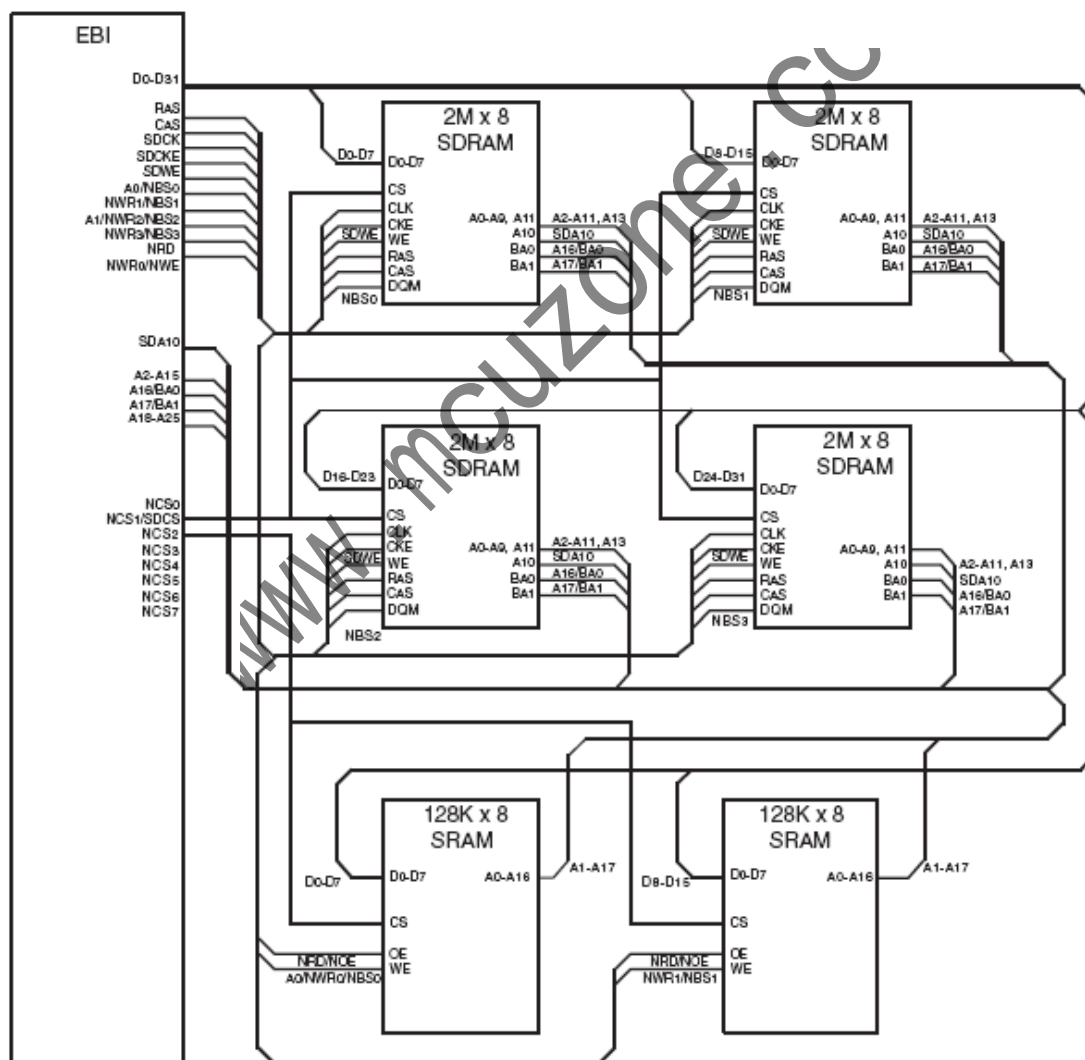
SDWE	WE			
NWAIT		WAIT	WAIT	
Pxx		CD1 or CD2	CD1 or CD2	
Pxx				CE
Pxx				RDY

- 注意：
1. 不是直接连接到 CompactFlash 槽。使能控制 EBI 数据总线和 CompactFlash 槽之间的双向缓冲。
  2. 任何 PIO 线
  3. CE 连接依赖于 NAND Flash。对标准的 NAND Flash 设备，必须被连接到任意空闲 PIO 线。对“CE don't care” NAND Flash 设备，可以连接到 NCS3/NANDCS 或任何空闲 PIO 线。

### 20.4.2 连接举例

图 20-2 展示了 EBI 和外部设备连接的例子

图 20-2 EBI 连接到存储设备



### 20.5 产品相关性

### 20.5.1 I/O 口线

引脚用于接外部总线接口可以和 PIO 块多路复用。编程者必须首先编程 PIO 控制器来指定外部总线接口引脚到他们的外设功能。如果外部总线接口的 I/O 口线未被应用程序使用，则 I/O 口线可以被 PIO 控制器用作其他目的。

## 20.6 功能描述

EBI 在内部 AHB 总线（由总线矩阵驱动）和外部存储或外设间传输数据。控制波形和外部地址的参数，数据和控制总线。它由以下元素组成：

- 静态存储控制器（SMC）
- SRAM 控制器(SDRAMC)
- 片选分配部件 分配一个 AHB 地址空间到外部设备。
- 一个多路复用控制器电路共享不同控制器间的引脚。
- 可编程的 CompactFlash 支持逻辑
- 可编程的 NAND Flash 支持逻辑

### 20.6.1 总线多路复用

EBI 提供一个完整的控制信号集，此信号集通过存储器区请求功能的一个多路复用逻辑操作共享 32 位数据块，多达 26 位的地址块和控制信号。

当无外部访问正在被执行，多路复用被特别组织来在稳定状态保证地址和输出块的维护。多路复用还被指定来维护在存储控制器中定义的数据浮动时间。此外，SDRAM 的刷新周期在没有延迟其他外部存储控制器访问的情况下被 SDRAM 控制器独立的刷新。

### 20.6.2 上拉电阻控制

总线矩阵用户接口中的 EBI\_CSA 寄存器可以使能不和 PIO 控制器块多路复用的数据总线块上的片上上拉电阻。上拉电阻在复位后被使能。置位 DBPUC 位禁用 D0-D15 块上的上拉电阻。使能 D16-D31 上的上拉电阻可通过编程适当的 PIO 控制器执行。

### 20.6.3 静态存储控制器

对静态存储控制器上的信息，参考静态存储控制器章节

### 20.6.4 SDRAM 控制器

对 SDRAM 控制器上的信息，参考 SDRAM 章节

### 20.6.5 CompactFlash 支持

外部总线接口集成了连接 CompactFlash 设备的电路。

CompactFlash 逻辑被 NCS4 和/或 NCS5 地址空间上的静态存储控制器（SMC）驱动。编程 EBI\_CSA 寄存器上的 CS4A 和/或 CS5A 位到适当的值使能此逻辑。此寄存器的细节，请参考总线矩阵用户接口章节。访问一个外部 CompactFlash 设备通过访问保留到 NCS4 和/或 NCS5 的地址空间（就是说，对 NCS4 在 0x5000 0000 和 0x5FFF FFFF 之间，对 NCS5 在 0x6000 0000 和 0x6FFF FFFF 之间）完成。

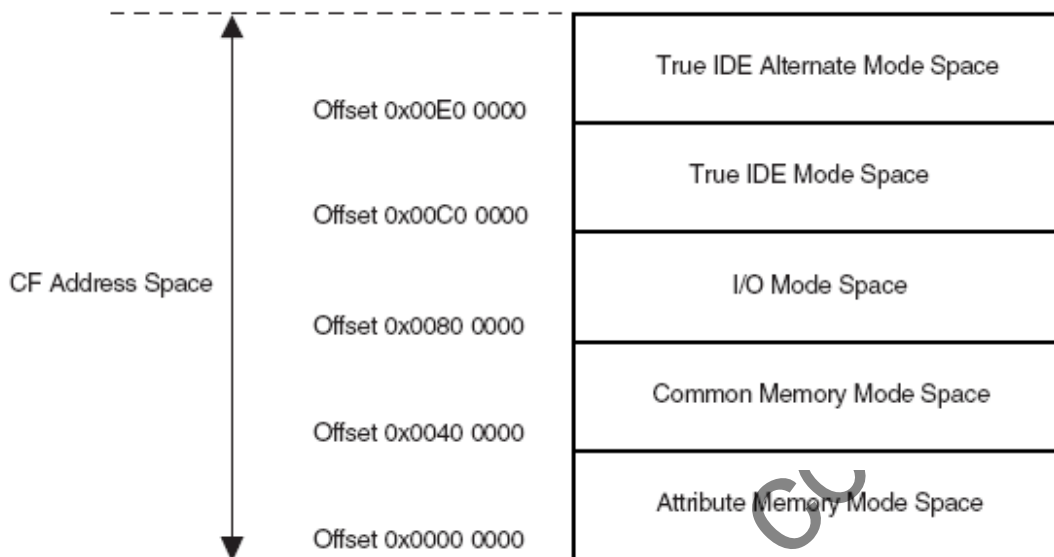
所有的 CompactFlash 模式(Attribute Memory, Common Memory, I/O and True IDE)是被支持的，但是信号 \_IOIS16 (I/O 和 True IDE 模式) 和 \_ATA SEL (True IDE 模式)不被处理。

#### 20.6.5.1 I/O 模式, Common Memory 模式, Attribute Memory 模式和 True IDE 模式

在 NCS4 和/或 NCS5 地址空间，当前传输地址区别 I/O 模式, Common Memory 模式, Attribute Memory 模式和 True IDE 模式。

不同模式通过一个如图 20-3 所说明的特殊存储映射被访问。传输地址的 A[23:21]位被用于选择需要的模式，如表 20-5 描述。

图 20-3 CompactFlash 存储映射



注意：EBI 的 A22 引脚被用于驱动 CompactFlash 设备（除了在 True IDE 模式）的 REG 信号。

表 20-5 CompactFlash 模式选项

A[23:21]	模式基本地址
000	Attribute Memory
010	Common Memory
100	I/O 模式
110	True IDE 模式
111	交替 True IDE 模式

#### 20.6.5.2 CFCE1 和 CFCE2 信号

为了覆盖所有访问类型，SMC 必须交替的置位去驱动 8 位数据总线或 16 位数据总线。D[7:0]总线上的奇字节访问仅当 SMC 被配置用来驱动对应的 NCS 引脚（NCS4 和 或 NCS5）上的 8 位存储器。NCS4 和/或 NCS5 地址空间的片选寄存器(对应片选模式中的的 DBW 域)必须置位，如表 20-6 所示，去使能需要的访问类型。

NBS1 和 NBS0 是 SMC 的字节选择信号并当 SMC 在对应片选设置为字节选择模式时可用。

CFCE1 和 CFCE2 波形与相应的 NCSx 波形相同。关于这些波形和定时的细节，参考静态存储控制器选项。

表 20-6 CFCE1 和 CFCE2 真值表

模式	CFCE2	CFCE1	DBW	备注	SMC 访问模式
Attribute Memory	NBS1	NBS0	16 位	访问 D[7:0]偶字节	字节选择
Common Memory	NBS1	NBS0	16 位	访问 D[7:0]偶字节	字节选择

				访问 D[15:8]奇字节	
	1	0	8 位	访问 D[7:0]奇字节	无关紧要
I/O 模式	NBS1	NBS0	16 位	访问 D[7:0]偶字节 访问 D[15:8]奇字节	字节选择
	1	0	8 位	访问 D[7:0]奇字节	无关紧要
True IDE 模式					
Task file	1	0	8 位	访问 D[7:0]偶字节 访问 D[7:0]奇字节	无关紧要
数据寄存器	1	0	16 位	访问 D[7:0]偶字节 访问 D[15:8]奇字节	字节选择
交替 True IDE 模式					
控制寄存器 交替状态读	0	1	无 关 紧 要	访问 D[7:0]偶字节 访问 D[7:0]奇字节	无关紧要
驱动地址	0	1	8 位		无关紧要
True IDE Standby 模 式或地址空间未指定 到 CF	1	1	无 关 紧 要	无关紧要	无关紧要

### 20.6.5.3 读/写信号

在 I/O 模式和 True IDE 模式，CompactFlash 逻辑驱动 CFIOR 和 CFLOW 上的 SMC 读和写命令信号，当 CFOE 和 CFWE 信号无效。同样的，在 Common Memory 模式和 Attribute Memory 模式，SMC 信号在 CFOE 和 CFWE 信号上被驱动，当 CFIOR 和 CFLOW 无效。149 页图 20-4 展示了此逻辑的含义。通过置位地址设置和保持 NCS4(和/或 NCS5)片选上的时间到适当值支持属性存储模式，公用存储模式和 I/O 模式。

图 20-4 CompactFlash 读/写控制信号

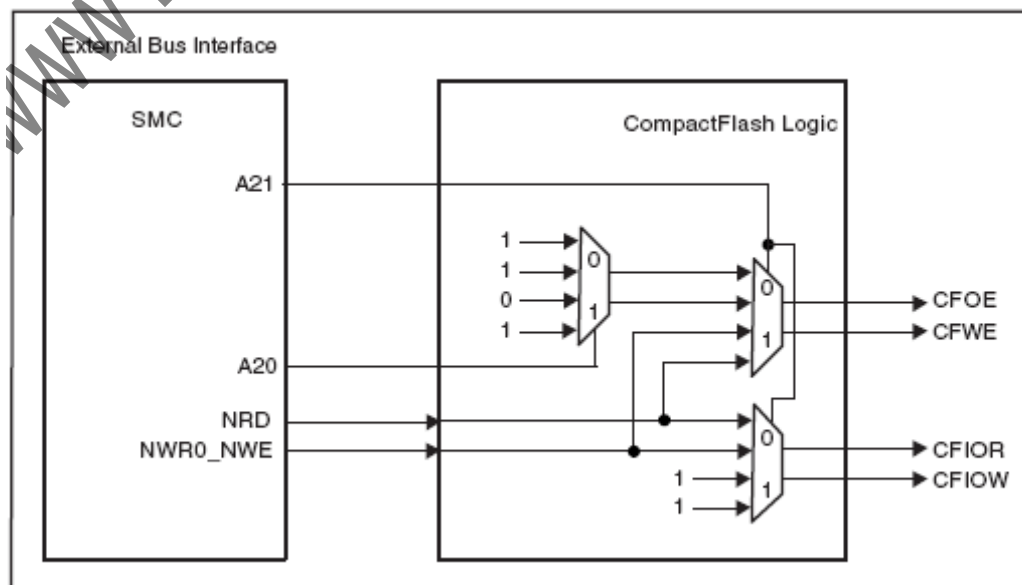


表 20-7 CompactFlash 模式选项

模式基地址	CFOE	CFWE	CFIOR	CFLOW
-------	------	------	-------	-------

<b>Attribute Memory Common Memory</b>	NRD	NWR0_NWE	1	1
<b>I/O 模式</b>	1	1	NRD	NWR0_NWE
<b>True IDE 模式</b>	0	1	NRD	NWR0_NWE

#### 20.6.5.4 EBI 引脚上 CompactFlash 信号的多路复用

149 页表 20-8 和 150 页表 20-9 揭示了 CompactFlash 逻辑信号和 EBI 引脚上其他 EBI 信号的多路复用。只要 EBI\_CSA 寄存器的 CS4A 和/或 CS5A 域置位，表 20-8 中 EBI 引脚严格的专用于 CompactFlash 接口。

当对应的 CompactFlash 接口使能（CS4A = 1 和/或 CS5A = 1），150 页表 20-9 中 EBI 引脚在所有存储区仍然可共享。

表 20-8 专用的 CompactFlash 接口多路复用表

引脚	CompactFlash 信号		EBI 信号	
	CS4A = 1	CS5A = 1	CS4A = 0	CS5A = 0
NCS4/CFCS0	CFCS0		NCS4	
NCS5/CFCS1		CFCS1		NCS5

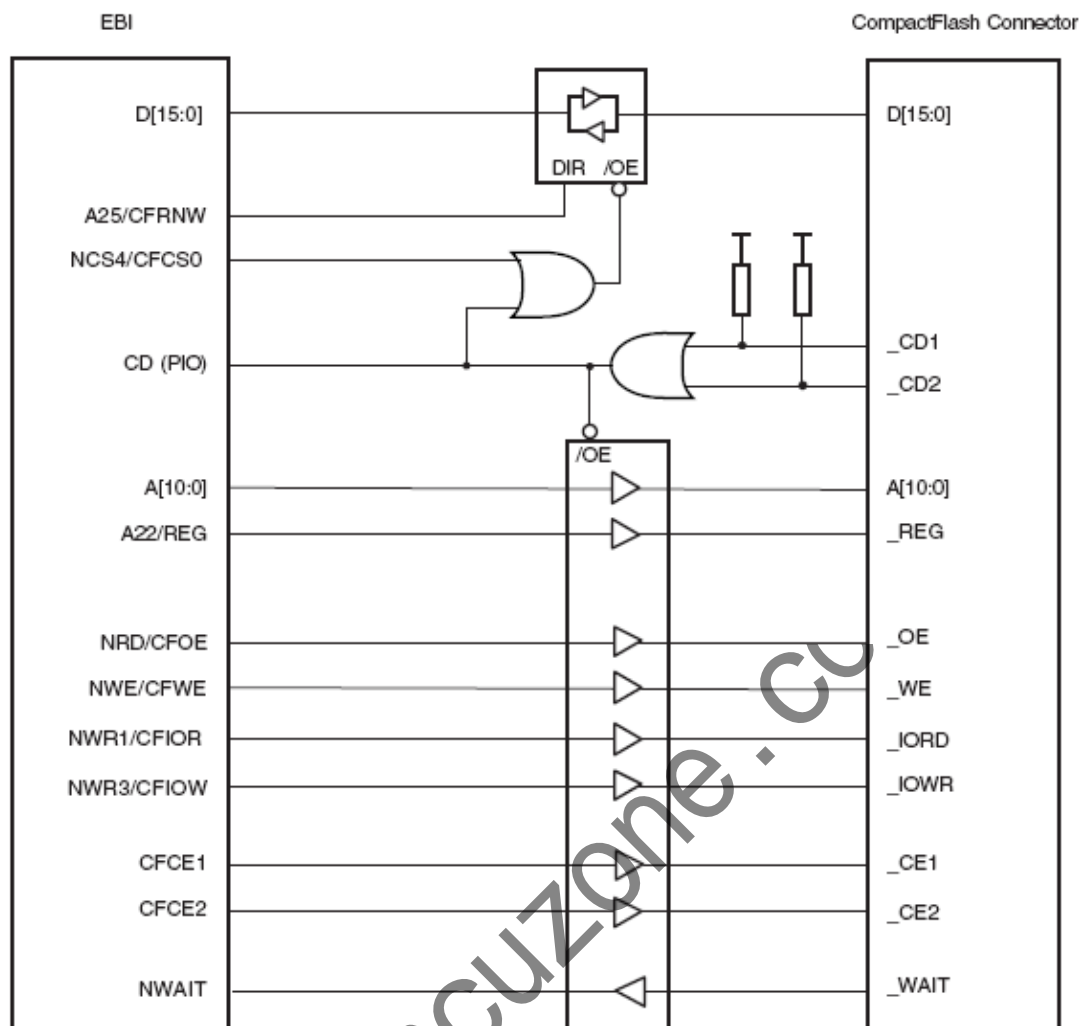
表 20-9 共享的 CompactFlash 接口多路复用表

引脚	访问 CompactFlash 设备	访问其他 EBI 设备
	CompactFlash 信号	EBI 信号
NRD/CFOE	CFOE	NRD
NWR0/NWE/CFWE	CFWE	NWR0/NWE
NWR1/NBS1/CFIOR	CFIOR	NWR1/NBS1
NWR3/NBS3/CFIOW	CFIOW	NWR3/NBS3
A25/CFRNW	CFRNW	A25

#### 20.6.5.5 应用举例

150 页图 20-5 说明了 CompactFlash 应用。CFCS0 和 CFRNW 信号未直接连接到 CompactFlash 槽 0，但是控制在 EBI 和 CompactFlash 设备间的缓冲的方向与输出使能。CFCS0 信号的时序与 NCS4 信号相同。而且，CFRNW 信号和地址总线一样，在整个传输中始终有效。CompactFlash\_WAIT 信号连接于静态存储控制器的 NWAIT 输入。关于这些波形和时序的细节，参考静态存储控制器章节。

图 20-5 CompactFlash 应用程序举例



### 20.6.6 NAND Flash支持

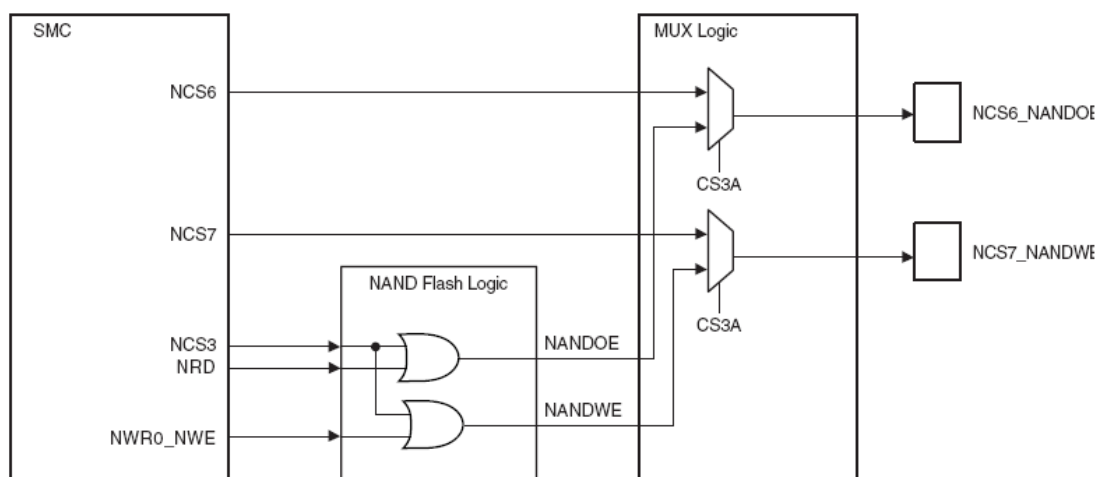
EBI集成了连接NAND Flash设备的电路。

NAND Flash逻辑被NCS3地址空间上的静态存储控制器驱动。编程总线矩阵用户接口中EBI\_CSA寄存器中的CS3A到适当的值使能NAND Flash逻辑。关于此寄存器的细节，参考总线矩阵用户接口段。通过访问NCS3（就是，在0x40000000和0x4FFFFFFF之间）的地址空间来访问外部NAND Flash设备。

当NCS3信号有效，NAND Flash逻辑驱动NANDOE和 NANDWE信号上SMC的读和写命令。只要访问NCS3地址空间失败，NANDOE 和 NANDWE就被使无效。关于这些波形的更多细节，参考静态存储控制器章节。

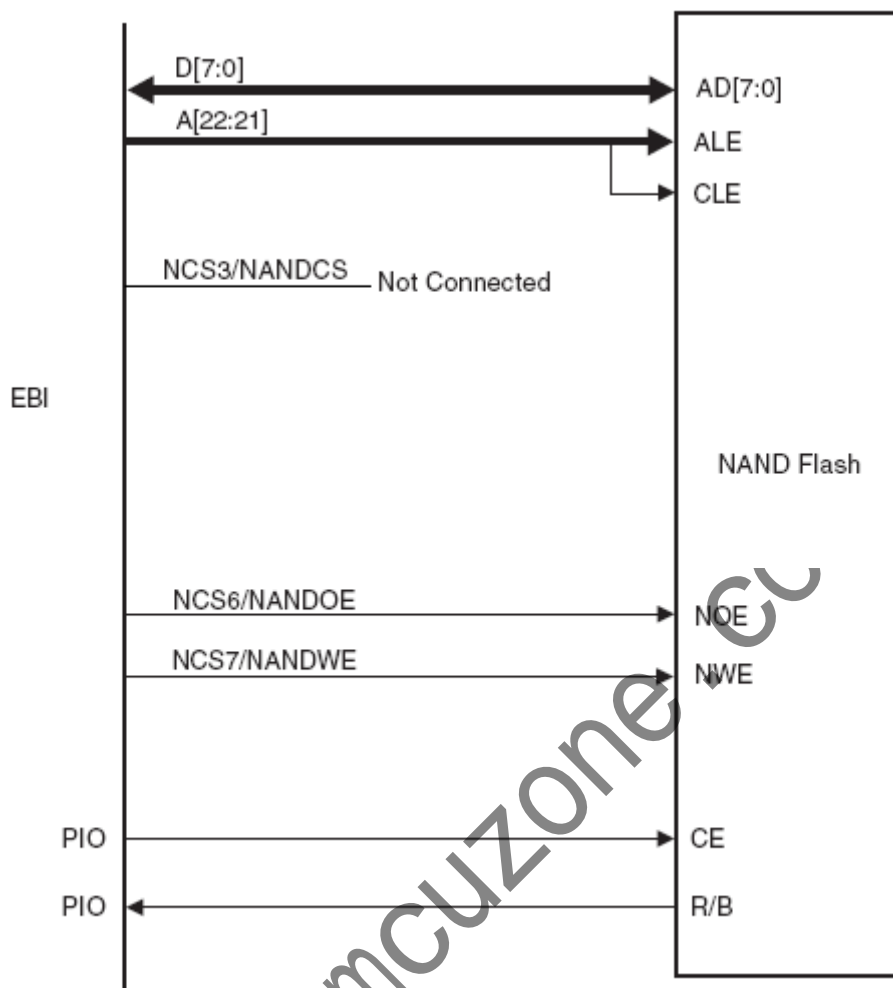
NANDOE 和 NANDWE信号同静态存储控制器的NCS6和NCS7多路复用。此多路复用在EBI的MUX逻辑被EBI\_CSA寄存器中的CS3A位 控制。关于此寄存器的细节，参考总线矩阵用户接口章节。此时NCS6 和 NCS7变得不可用。执行在保留到NCS6 和 NCS7（就是说，在0x70000000和0x8FFFFFFF之间）的地址空间的一个访问可能导致一个不可预期的结果。

图20-6 EBI引脚上NAND Flash信号多路复用



NAND Flash设备上的地址锁存使能和命令锁存使能信号通过EBI地址总线的地址位A22和A21驱动。用户应该注意EBI地址总线上的任何位也可能被用作此目的。NAND Flash设备的数据总线上的命令，地址或数据字通过用他们在NCS3地址空间的地址来区别。设备的芯片使能（CE）信号和准备/忙（R/B）信号被连接到PIO口线。就算NCS3未被选择，CE信号仍然有效，防止此设备进入待机状态。

图20-7 NAND Flash应用程序举例



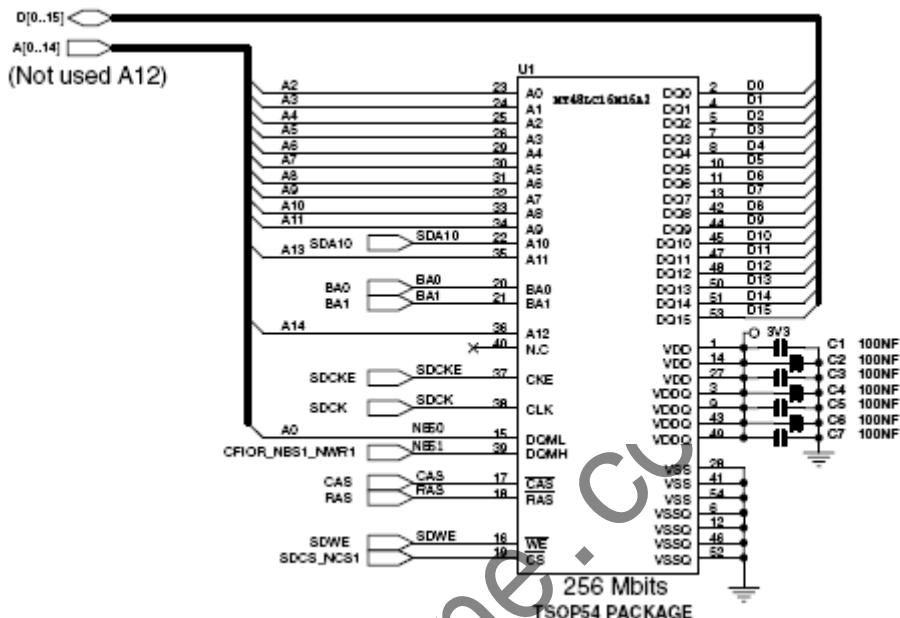
注意：外部总线接口还可支持16位设备。

## 20.7 实现举例

所有硬件配置仅给出说明。用户应该参考存储器制造商网站去确认设备可用性。

### 20.7.1 16位SDRAM

#### 20.7.1.1 硬件配置



#### 20.7.1.2 软件配置

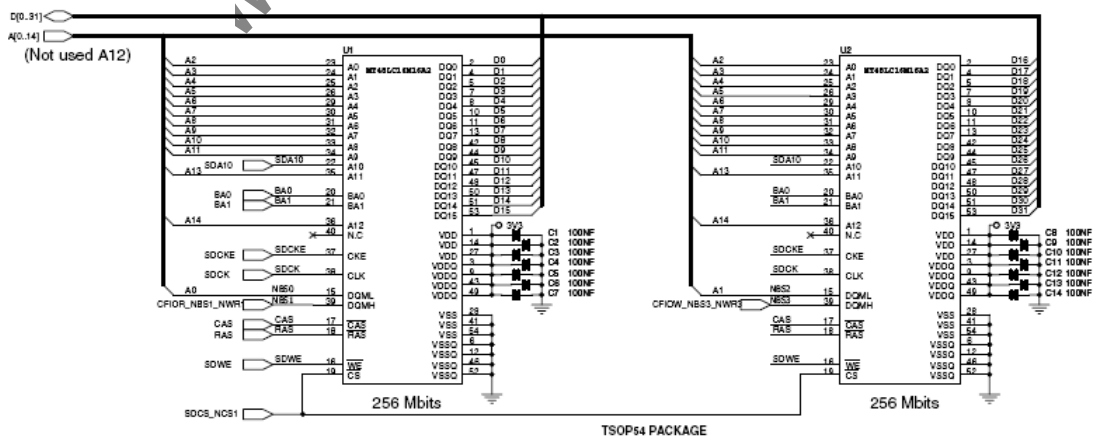
以下配置必须被执行：

- 通过置位放置在总线矩阵存储器空间EBI片选制定寄存器中的EBI\_CS1A位指定EBI CS1到SDRAM控制器。
- 初始化SDRAM控制器，依赖于SDRAM设备和系统总线频率数据总线宽度编程为16位宽。

数据总线初始化顺序在SDRAM控制器的“SDRAM设备初始化”章节有描述。

### 20.7.2 32位SDRAM

#### 20.7.2.1 硬件配置



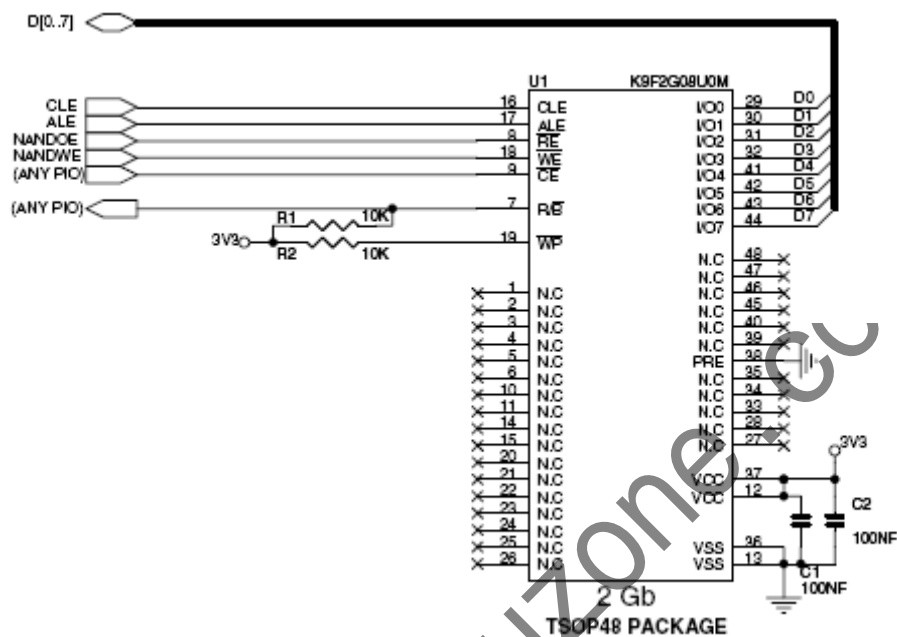
#### 20.7.2.2 软件配置

以下配置必须被执行：

- 通过置位位于在总线矩阵存储器空间EBI片选制定寄存器中的EBI\_CS1A位指定EBI CS1到SDRAM控制器。
- 初始化SDRAM控制器，依赖于SDRAM设备和系统总线频率  
数据总线宽度被编程为32位。数据口线D[16..31]和PIO块多路复用，因此专用的PIO必须在PIO控制器中被编程为相应的外设模式。  
数据总线初始化顺序在SDRAM控制器的“SDRAM设备初始化”章节有描述。

### 20.7.3 8位NANDFlash

#### 20.7.3.1 硬件配置



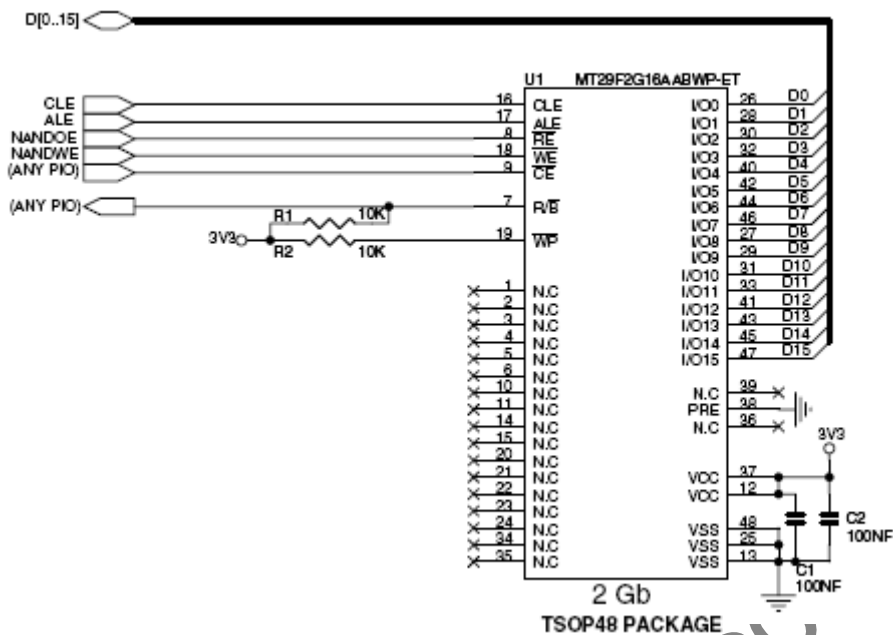
#### 20.7.3.2 软件配置

以下配置必须被执行：

- 通过置位放置在总线矩阵存储器空间EBI片选制定寄存器中的EBI\_CS3A位指定EBI CS3到SDRAM控制器。
- 给ALE / CLE功能保留A21 / A22。地址和命令锁存在访问期间分别通过置位地址位A21和A22来控制。
- NANDOE 和 NANDWE信号同PIO块多路复用，因此专用的PIO必须在PIO控制器中被编程为相应的外设模式。
- 配置一个PIO口线作为输入来管理准备/忙信号
- 根据NAND Flash时序，数据总线宽度和系统总线频率配置静态存储控制器CS3的建立，脉冲，周期参数和模式。

### 20.7.4 16位NANDFlash

#### 20.7.4.1 硬件配置

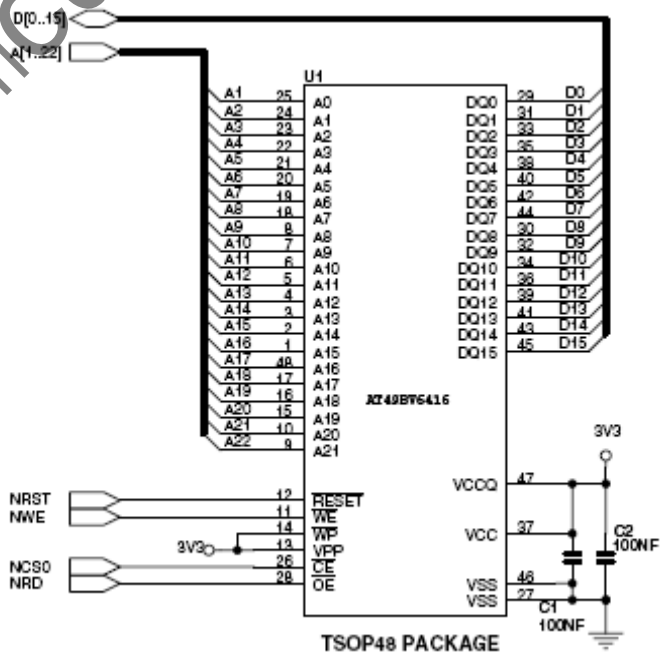


#### 20.7.4.2 软件配置

除了在静态存储控制器的模式寄存器中被编程的数据总线宽度，软件配置和8位NANDFlash一样。

### 20.7.5 NOR Flash on NCS0

#### 20.7.5.1 硬件配置



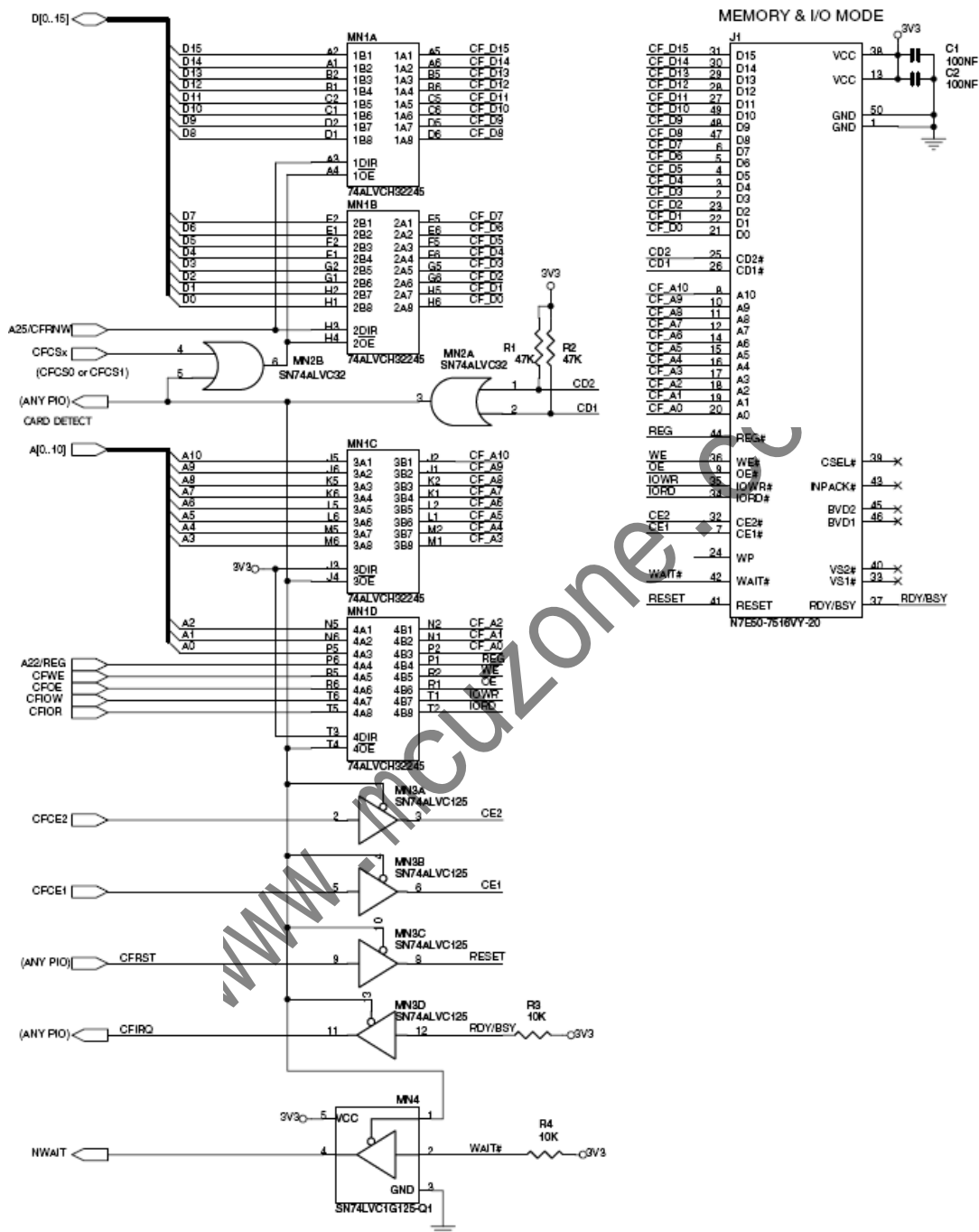
#### 20.7.5.2 软件配置

对静态存储控制器的缺省配置，字节选择模式，16位数据总线，读/写被片选控制，允许以慢时钟在16位非易失性存储器上启动。

对其他配置，根据Flash定时和系统总线频率配置静态存储控制器CS0建立，脉冲，周期参数和模式。

## 20.7.6 Compact Flash

### 20.7.6.1 硬件配置



### 20.7.6.2 软件配置

以下配置必须被执行：

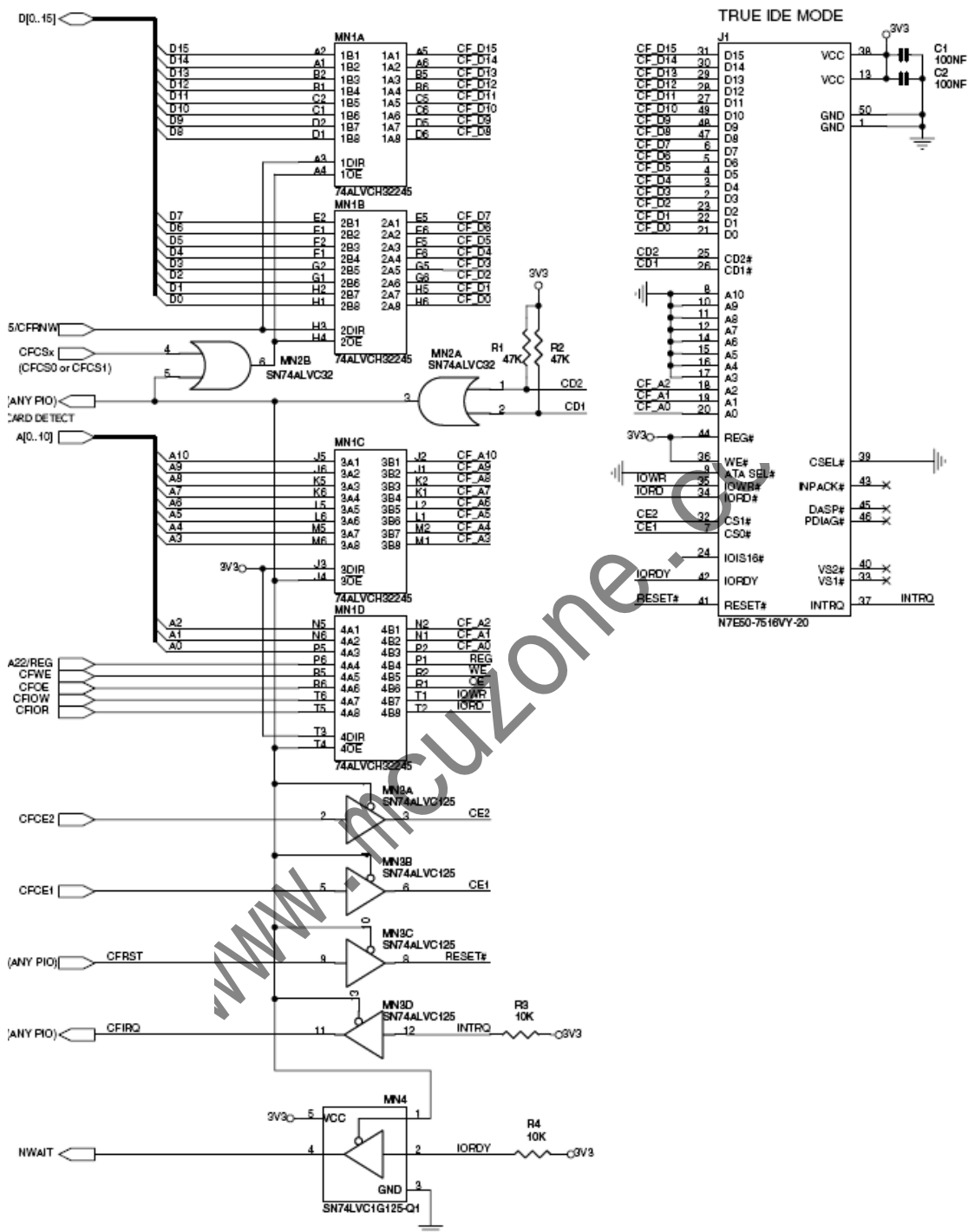
- 通过置位放置在总线矩阵存储空间中的EBI片选指定寄存器中的EBI\_CS4A 或和 EBI\_CS5A位，指定EBI CS4 和/或 EBI\_CS5到CompactFlash 槽 0 或/和 槽 1。
- 地址线A23选择I/O (A23=1) 或存储模式(A23=0)，A22用于REG功能。

- A23, CFRNW, CFS0, CFCS1, CFCE1 和 CFCE2信号同PIO口线多路复用，因此专用的PIO必须在PIO控制器外设模式下被编程。
- 为CFRST配置一个PIO口线作为输入，分别为CFIRQ和CARD DETECT配置其他的两个PIO口线作为输入。
- 根据Compact Flash时序和系统总线频率，配置SMC CS4 和/或 SMC\_CS5 (给 slot 0 或 1)建立，脉冲，周期参数和模式。

[www.mcuzone.com](http://www.mcuzone.com)

## 20.7.7 Compact Flash True IDE

### 20.7.7.1 硬件配置



### 20.7.7.2 软件配置

以下配置必须被执行：

- 通过置位放置在总线矩阵存储空间的EBI片选指定寄存器中的EBI\_CS4A 或/和 EBI\_CS5A位，指定EBI CS4 和/或 EBI\_CS5到CompactFlash slot 0

或/和 slot 1。

- 地址线A21用于选择交替True IDE (A21=1) 或 True IDE (A21=0)模式。
- CFRNW, CFS0, CFCS1, CFCE1 和 CFCE2信号同PIO块多路复用，因此专用的PIO必须在PIO控制器外设模式下被编程。
- 为CFRST配置一个PIO口线作为输入，分别为CFIRQ和CARD DETECT配置其他的两个PIO口线作为输入。
- 根据Compact Flash时序和系统总线频率，配置SMC CS4 和/或 SMC\_CS5 (给 slot 0 或 1)的建立，脉冲，周期参数和模式。

www.mcuzone.cn