

BUF_AMOUNTn	位	描述	初始状态
Reserved	[31:16]		
AMOUNT	[15:0]	该值依赖实际使用的存储设备。复位后，该寄存器通过闪存控制器被设置。 只读	

7.7.13. 技术寄存器

寄存器	地址	读/写	描述	复位值
TECH0	0x701000C0	读	Bank0 技术寄存器	由存储器决定的
TECH1	0x701800C0			

TECHn	位	描述	初始状态
Reserved	[31:16]		
TECHNOLOGY	[15:0]	该值依赖实际使用的存储设备。复位后，该寄存器通过闪存控制器。只读	

7.7.14. FBA 宽度寄存器

寄存器	地址	读/写	描述	复位值
FBA_WIDTH0	0x701000D0	读/写	Bank0 FBA 宽度寄存器	0x000A
FBA_WIDTH1	0x701800D0			

FBA_WIDTHn	位	描述	初始状态
Reserved	[31:5]		0
FBA	[4:0]	设置块的数量。默认值为 0x0A。初始化期间设置通过软件	0x0A

7.7.15. FPA 宽度寄存器

寄存器	地址	读/写	描述	复位值
FPA_WIDTH0	0x701000E0	读/写	Bank0FPA 宽度寄存器	0x0006
FPA_WIDTH1	0x701800E0			

FPA_WIDTHn	位	描述	初始状态
Reserved	[31:5]	保留。	0
FPA	[4:0]	设置数值，将用来代表页的数值。默认值为 6。初始化期间通过软件设置	0x06

7.7.16. FSA 宽度寄存器

寄存器	地址	读/写	描述	复位值
FSA_WIDTH0	0x701000F0	读/写	Bank0 FSA 宽度寄存器	0x0002
FSA_WIDTH1	0x701800F0			

FSA_WIDTHn	位	描述	初始状态
Reserved	[31:3]		0
FSA	[2:0]	设置位的数值，将用来表示扇区的数值。默认值为 0x2。初始化期间通过软件设置	0x2

7.7.17. 修正寄存器

寄存器	地址	读/写	描述	复位值
REVISION0	0x70100100	读	Bank0 修正寄存器	0x00000002
REVISION1	0x70180100			

REVISIONn	位	描述	初始状态
Reserved	[31:16]		0x0000
REVISION	[15:0]	保存控制器修正数。默认值为 0x1。只读	0x0002

7.7.18. DATARAM0 代码寄存器

寄存器	地址	读/写	描述	复位值
DATARAM00	0x70100110	读/写	Bank0 Dataram0 代码寄存器	0x0002
DATARAM01	0x7018110			

DATARAMn	位	描述	初始状态
Reserved	[31:16]		0
DATARAM0	[3:0]	ram0 设置数据的非扇区部分。默认值为 0x2。在初始化期间通过软件设置	2

7.7.19. DATARAM1 代码寄存器

寄存器	地址	读/写	描述	复位值
DATARAM10	0x70100120	读/写	Bank0 Dataram1 代码寄存器	0x0003
DATARAM11	0x7018120			

DATARAMn	位	描述	初始状态
Reserved	[31:4]		0
DATARAM1	[3:0]	ram1 设置数据的非扇区部分。默认值为 0x3。在初始化期间通过软件设置	3

7.7.20. 同步模式寄存器

寄存器	地址	读/写	描述	复位值
SYNC_MODE0	0x70100130	读	Bank0 同步模式寄存器	由存储器决定的
SYNC_MODE1	0x70180130			

SYNC_MODE _n	位	描述	初始状态
Reserved	[31:1]		0
RM	[1]	为读操作设置同步或异步传输模式。默认值为0x0。在初始化期间通过软件设置。该值从MEM_CFG寄存器[15]拷贝。只读 <ul style="list-style-type: none"> • 0 = 异步模式 • 1 = 激活同步模式 	0
WM	[0]	为写操作设置同步或异步传输模式。默认值为0x0。在初始化期间通过软件设置。该值从MEM_CFG寄存器[1]拷贝。只读 <ul style="list-style-type: none"> • 0 = 异步模式 • 1 = 激活同步模式 	0

7.7.21. 传输备用寄存器

寄存器	地址	读/写	描述	复位值
TRANS_SPARE0	0x70100140	读/写	Bank0 传输大小寄存器	0x0000
TRANS_SPARE1	0x70180140			

TRANS_SPARE _n	位	描述	初始状态
Reserved	[31:1]		0
TSRF	[0]	通过映射01, 涉及到所有的读写指令。如果该位	0

		被设置, 存储器备用区的数据将被存储控制器异步FIFO。备用区的大小依赖扇区数 <ul style="list-style-type: none"> • 0 = 只传输数据 • 1 = 增加扇区大小 页的主要数据区首先传输, 然后是备用区	
--	--	---	--

7.7.22. DBS-DFS 宽度寄存器

寄存器	地址	读/写	描述	复位值
DBS_DFS_WIDTH0	0x70100160	读/写	Bank0 页 DBS_DFS 宽度	由存储器决定
DBS_DFS_WIDTH1	0x70180160		寄存器	的

DBS_DFS_WIDTHn	位	描述	初始状态
Reserved	[31:2]		0
WIDTH	[1:0]	设置 DBS 和 DFS 宽度。在初始化期间通过软件设置。如果没关系则忽略	0

7.7.23. 页计数寄存器

寄存器	地址	读/写	描述	复位值
PAGE_CNT0	0x70100170	读	Bank0 页计数寄存器	0x0000
PAGE_CNT1	0x70180170			

PAGE_CNTn	位	描述	初始状态
Reserved	[31:8]		0
PAGE_COUNT	[7:0]	通过当前执行的多页回拷贝指令保存回拷贝页数。只读	0

7.7.24. 错误页地址寄存器

寄存器	地址	读/写	描述	复位值
ERR_PAGE_ADDRO	0x70100180	读	Bank0 错误页地址寄存器	0x0000
ERR_PAGE_ADDR1	0x70180180			

ERR_PAGE_ADDRn	位	描述	初始状态
Reserved	[31:6]		0
FAIL_PAGE_ADDR	[5:0]	执行后, 加载或者擦除错误中断, 该寄存器将保存没有成功操作的页的地址。只读	0

7.7.25. 脉冲读出等待寄存器

寄存器	地址	读/写	描述	复位值
BURST_RD_LAT0	0x70100190	读	Bank0 脉冲读出延迟寄存器	由存储器决定的
BURST_RD_LAT1	0x70180190			

BURST_RD_LATn	位	描述	初始状态
Reserved	[31:3]		0
BURST_RD_LAT	[2:0]	在周期中设置脉冲读出。默认值是 0x6。该值从 MEM_CFG[14:12]复制。只读	寄存器

7.7.26. 中断引脚使能寄存器

寄存器	地址	读/写	描述	复位值
INT_PIN_ENABLE0	0x701001A0	读/写	Bank0 中断引脚使能寄存器。	0x0000
INT_PIN_ENABLE1	0x701801A0			

INT_PIN_ENABLEn	位	描述	初始状态
Reserved	[31:1]		0
INT	[0]	<p>中断引脚有效。</p> <p>如果需要使用中断引脚或者控制器要从状态寄存器查看中断信息，该位有效。初始化期间通过软件设置。</p> <ul style="list-style-type: none"> • 0 = 使用状态寄存器 • 1 = 使用中断引脚 	0

7.7.27. 中断监控周期寄存器

寄存器	地址	读/写	描述	复位值
INT_MON_CYC0	0x701001B0	读/写	Bank0 中断监控寄存器	0x01F4
INT_MON_CYC1	0x701801B0			

INT_MON_CYCn	位	描述	初始状态
Reserved	[31:12]	保留。	0
INT_MON_CYC	[11:0]	在 INT_ERR_STA 寄存器检测和存储状态寄存器之间设置周期值。如果闪存配置寄存器位 IOBE 被清除，则使用该寄存器	500

7.7.28. 存取时钟寄存器

寄存器	地址	读/写	描述	复位值
ACC_CLOCK0	0x701001C0	读/写	Bank0 存取时钟寄存器	0x0003
ACC_CLOCK1	0x701001C1			

ACC_CLOCKn	位	描述	初始状态
Reserved	[31:3]	保留。	0
ACCESS_CLK	[2:0]	设置周期值，要求来完成闪存设备存取的时间。遵循下面的公式： (35ns/时钟周期+1) 系 统 时 钟 闪存时钟 (MHz) 访问时钟 (MHz) 166 83 3 134 67 3 100 50 2 60 30 2	3

7.7.29. 慢读路径寄存器

寄存器	地址	读/写	描述	复位值
SLOW_RD_PATH0	0x701001D0	读/写	Bank0 缓慢读通道寄存器	0x0000
SLOW_RD_PATH1	0x701801D0			

SLOW_RD_PATHn	位	描述	初始状态
Reserved	[31:1]	保留	0
SRP	[0]	默认值是 0x0	0

7.7.30. 错误块地址寄存器

寄存器	地址	读/写	描述	复位值
ERR_BLK_ADDR0	0x701001E0	读	Bank0 错误块地址寄存器	0x0000
ERR_BLK_ADDR1	0x701801E0			

ERR_BLK_ADDRn	位	描述	初始状态
Reserved	[31:12]		0
FAIL_BLK_ADDR	[11:0]	执行后，载入或者擦除中断，寄存器将保存操作失败的时钟地址。只读	0

7.7.31. 闪存版本 ID 寄存器

寄存器	地址	读/写	描述	复位值
FLASH_VER_ID0	0x701001F0	读	Bank0 闪存版本 ID 寄存器	由存储器决定的
FLASH_VER_ID1	0x701801F0			

FLASH_VER_IDn	位	描述	初始状态
Reserved	[31:16]	保留	
FLASH_VER	[15:0]	该值依赖实际使用的存储器设备。复位后，该寄存器被闪存控制器设置。只读	

7.8 ONENAND 控制器应用举例

下面是几段实现 ONENAND 部分功能的程序，通过对程序进行分析，有助于对该控制器的理解。

7.8.1 ONENAND 初始化程序

```
//输出为:TRUE – 存储器设备复位; FALSE: 由于错误, 存储器设备不复位
bool ONENAND_Init(u32 Controller)
{
    u32 uBaseAddress;
// Controller — OneNand 控制器端口数
```

```
if(Controller == 0)
{
    uBaseAddress = ONENAND0_BASE;
}
else if(Controller == 1)
{
    uBaseAddress = ONENAND1_BASE;
}
else
{
    return FALSE;
}

ONENAND_pBase[Controller] = (void *)uBaseAddress;
```

```
#if (ONENAND_VERSION == ONENAND_EVT0)
//存储器端口 0 驱动激增(OneNand 控制信号)
```

```
GPIO_SetMem0DrvStrength(0x55000411);
```

```
#endif
```

```
ONENAND_SetFlashClock(Controller, eDiv2_HCLKx2);
```

```
ONENAND_SetAccClock(Controller);
```

```
ONENAND_GetDevInformation(Controller);
```

```
ONENAND_SetMemSpace(Controller);
```

```
ONENAND_EnableAllInterrupt(Controller);
```

```
ONENAND_DisableInterrupt(Controller, eOND_RDYACT);
```