

WaitEnable5	[22]	等待使能控制，用于存储器 Bank5 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[21]	保留。	0
DataWidth5	[20]	数据总线宽度控制，用于存储器 Bank5 0=8 位 1=16 位	0
ByteEnable4	[19]	nWBE / nBE(用于UB/LB) 控制，用于存储器Bank4 0 = 不使用UB/LB (XrnWBE[1:0]专门的nWBE[1:0]) 1 =使用 UB/LB (XrnWBE[1:0] 专门的 nBE[1:0])	0
WaitEnable4	[18]	等待使能控制，用于存储器Bank4 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[17]	保留。	0
DataWidth4	[16]	数据总线宽度控制，用于存储器 Bank4 0=8 位 1=16 位	0
ByteEnable3	[15]	nWBE / nBE(用于UB/LB) 控制，用于存储器Bank3 0 = 不使用UB/LB (XrnWBE[1:0]专门的nWBE[1:0]) 1 =使用 UB/LB (XrnWBE[1:0]专门的 nBE[1:0])	0
WaitEnable3	[14]	等待使能控制，用于存储器Bank3 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[13]	保留	0
DataWidth3	[12]	数据总线宽度控制，用于存储器 Bank3 0=8 位 1=16 位	0
ByteEnable2	[11]	nWBE / nBE(用于UB/LB) 控制，用于存储器Bank2。 0 = 不使用UB/LB (XrnWBE[1:0]专门的nWBE[1:0]) 1 =使用 UB/LB (XrnWBE[1:0]专门的 nBE[1:0])	0
WaitEnable2	[10]	等待使能控制，用于存储器Bank2 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[9]	保留	0
DataWidth2	[8]	数据总线宽度控制，用于存储器 Bank2 0=8 位 1=16 位	0

ByteEnable1	[7]	nWBE / nBE(用于UB/LB) 控制, 用于存储器Bank1。 0 = 不使用UB/LB (XrnWBE[1:0]专门的nWBE[1:0]) 1 =使用 UB/LB (XrnWBE[1:0]专门的 nBE[1:0])	0
WaitEnable1	[6]	等待使能控制, 用于存储器Bank1 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[5]	保留	0
DataWidth1	[4]	数据总线宽度控制, 用于存储器 Bank1 0=8 位 1=16 位	0
ByteEnable0	[3]	nWBE / nBE(用于UB/LB) 控制, 用于存储器Bank0 0 = 不使用UB/LB (XrnWBE[1:0]专门的nWBE[1:0]) 1 =使用 UB/LB (XrnWBE[1:0]专门的 nBE[1:0])	0
WaitEnable0	[2]	等待使能控制, 用于存储器Bank0 0=WAIT 禁止 1=WAIT 使能	0
Reserved	[1]	保留	0
DataWidth0	[0]	数据总线宽度控制, 用于存储器 Bank0 0=8 位 1=16 位	H/W Set

6.3.2. SRAM 页控制寄存器(SRAM_BC: XrCSn0~XrCSn2)

寄存器	地址	读/写	描述	复位值
SRAM_BC0	0x70000004	读/写	SRAM Bank0 控制寄存器	0x000F_0000
SRAM_BC1	0x70000008	读/写	SRAM Bank1 控制寄存器	0x000F_0000
SRAM_BC2	0x7000000C	读/写	SRAM Bank2 控制寄存器	0x000F_0000
SRAM_BC3	0x70000010	读/写	SRAM Bank3 控制寄存器	0x000F_0000
SRAM_BC4	0x70000014	读/写	SRAM Bank4 控制寄存器	0x000F_0000
SRAM_BC5	0x70000018	读/写	SRAM Bank5 控制寄存器	0x000F_0000

		<p>.....</p> <p>1100=12 时钟 1101=13 时钟</p> <p>1110=14 时钟 1111=15 时钟</p>	
Tacp	[7:4]	<p>在页模式下, 页模式访问周期</p> <p>0000=0 时钟 0001=1 时钟</p> <p>0010=2 时钟 0011=3 时钟</p> <p>.....</p> <p>1100=12 时钟 1101=13 时钟</p> <p>1110=14 时钟 1111=15 时钟</p>	0000
Reserved	[3:2]	保留	
PMC	[1:0]	<p>页模式配置</p> <p>00=标准的 (1 位数据) 01=4 位数据</p> <p>10=8 位数据 11=16 位数据</p>	00

7 ONENAND 控制器

本节主要介绍 S3C6410 RSIC 微处理器 OneNAND 控制器的功能和使用。S3C6410 支持外部 16 位总线，用于同步和异步 OneNAND 外部存储（通过平分端口 0）。通过使用两个控制器，最大可支持 2 页。Denali OneNAND Flash 存储控制器被用做 S3C6410 的 OneNAND 控制器使用。OneNAND 控制器是由 Denali 开发、测试和许可的，具有先进的微控制器总线架构 (AMBA 2)，可兼容的系统单晶片外部设备。该控制器同时支持两组存储器。每个存储页仅支持多路复用的 OneNAND。使用 OneNAND Flash 代替 NAND Flash，其 ‘XSELNAND’ 引脚必须置 0（低电平）。

7.1 ONENAND 控制器的特性

OneNAND 控制器的特性包括以下几个方面：

- (1) 通过使用两个 OneNAND 控制器，支持最大 2 页。
- (2) 支持同步/异步多路复用的 OneNAND 存储器。
- (3) 支持 16 位宽的外部存储器数据通道。
- (4) 支持 SINGLE/INCR4/INCR8 脉冲传输，用于 32 位 AHB 数据总线。
- (5) 支持单一字传输，用于 32 位 AHB SFR 总线。
- (6) 支持仅 ERROR/OKAY 响应，用于两个 AHB 总线。
- (7) 数据缓冲以达到最高性能。
- (8) 在 Flash 控制器和系统总线接口间异步的先进先出，用于速度的匹配。
- (9) 通过地址映射，支持擦除命令。
- (10) 支持复制模式作为寄存器命令。
- (11) 如果 OneNAND 设备 ID 是 0x0040、0x0048 和 0x0058，支持写同步模式。
- (12) 如果 OneNAND 设备 ID 是 0x0030, 0x0034 及 OneNAND 版本 ID 位[9:8] 不是 2' b00，支持写同步模式。
- (13) 当映射 01 页访问命令被使用时，支持 LDM4/STM4。如果设备密度是 128Mb 或 256Mb，推荐不超过一个字的访问，用于 01 页访问命令。

1. ONENAND 控制器框图

如图 7-1 所示，显示了 OneNAND 控制器的结构框图。

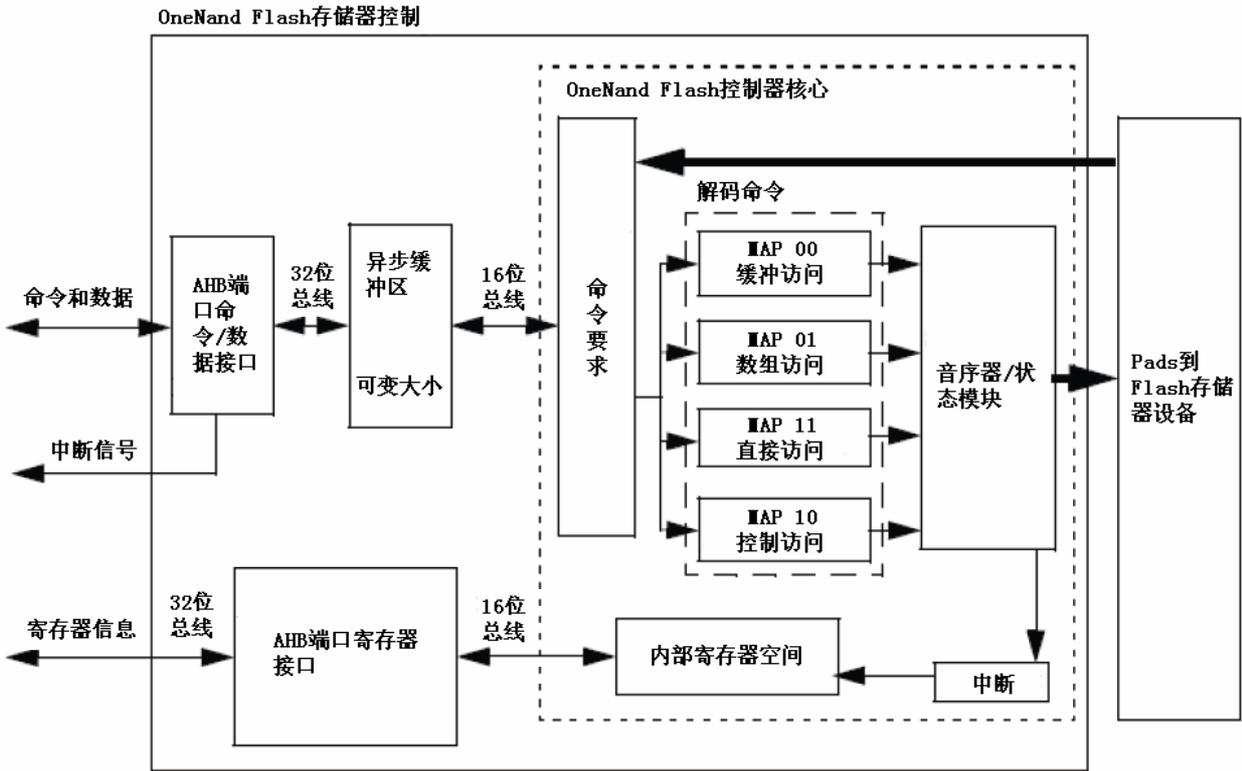


图 7-1 OneNAND 控制器结构框图

2. 信号描述

外部存储器接口的具体信号描述，如表 7-1 所示。

表 7-1 外部存储器接口信号描述

信号	I/O	描述
XmODATA	I/O	在内存读/写地址段期间，XmODATA[15:0]（数据总线）输出地址。内存读数据段时输入数据，内存写数据段时输出数据
XmOCSn[1:0]	0	XmOCSn[1:0]（可选芯片）被激活时，内存地址不超出每个页地址区域。通过系统控制 SFR 的设置，XmOCSn[1:0]可以被分配到 SROMC 或 OneNAND 控制器。 低有效
XmOWEn	0	XmOWEn（写使能）说明当前的总线周期是写周期 低有效

Xm00En	0	Xm00en（输出使能）说明当前的总线周期是读周期 低有效
Xm0INTsm0_FWEn Xm0INTsm1_FREn	I	从 OneNAND 存储页 0, 1 中断输入 如果 OneNAND 存储器没有被使用，这些信号必须置 0
Xm0ADDRVALID	0	地址有效输出。在 POP 产品中，地址和数据复用。Xm0ADDRVALID 说明总线被使用时， 用于地址 低有效
XmORPn	0	系统复位输出，OneNAND 存储器 低有效
XmORDY0_ALE XmORDY1_CLE	I	XmORDY 是同步脉冲等待输入，外部设备使用延迟同步脉冲转移。XmORDY 在同步读模 式下，说明数据有效。当 XmOCSn 为低时，XmORDY 被激活
Xm0SMCLK	0	静态存储器时钟，用于同步静态存储器设备

3. 输入时钟

OneNAND 控制器有三个时钟源输入。总线系统接口获得 AHB 总线时钟，即 HCLK。Flash 控制器核心获得两个 Flash 时钟，即 mclk 和 mclk_flash。mclk 的频率必须是 mclk_flash 的两倍，由 OneNAND flash 存储器提供。

可以设置系统控制器 SFR 的频率。当更改时钟频率的比率时，必须按照下面的程序：

- (1) 确保没有存储器传输。
- (2) 在系统控制器 SFR 内，转换时钟比率。
- (3) 写时钟比率寄存器。
- (4) 开始存储器访问。

7.2 存储器地址映射

OneNAND 控制器读内存设备 dev_id 的大小区域，以决定地址映射。自动配置 MEM_ADDR 区域的地址映射到支持的设备。如表 7-2 所示，“MEM_ADDR 区域”决定区域的大小，用于几个 OneNAND 存储器设备。

表 7-2 MEM_ADDR 区域

dev_id 区域 大小	密度	所属 块	页 大 小	映射 位置	MEM_ADDR 区域					
					保留	DFS_DBS	FBA	FPA	FSA	保留
0000	128Mb	256	1KB	[23:22]	[21:17]	N/A	[16:9]	[8:3]	[2]	[1:0]
0001	256Mb	512	1KB	[23:22]	[21:18]	N/A	[17:9]	[8:3]	[2]	[1:0]
0010	512Mb	512	2KB	[23:22]	[21:19]	N/A	[18:10]	[9:4]	[3:2]	[1:0]
0011	1Gb Dual Die	1024	2KB	[23:22]	[21:20]	[19]	[18:10]	[9:4]	[3:2]	[1:0]
0011	1Gb	1024	2KB	[23:22]	[21]	N/A	[19:10]	[9:4]	[3:2]	[1:0]
0100	2Gb Dual Die	2048	2KB	[23:22]	[21]	[20]	[19:10]	[9:4]	[3:2]	[1:0]
0100	2Gb	2048	2KB	[23:22]	N/A	N/A	[20:10]	[9:4]	[3:2]	[1:0]
0101	4Gb Dual Die	4096	2KB	[23:22]	N/A	[21]	[20:10]	[9:4]	[3:2]	[1:0]
0101	4Gb	4096	2KB	[23:22]	N/A	N/A	[21:10]	[9:4]	[3:2]	[1:0]

7.3. 命令映射

有四种命令支持 OneNand flash 存储控制器。这些命令通过引入地址的第 23 位和 22 位的值被选择。命令映射决定低 22 位的地址被使用的方式。

1. “00” =Map 00 命令。

Map 00 命令是用来存取控制器选定的缓冲区,在 Flash 存储设备里,包括引导,dataram0 和 dataram1。地址总是从 0x0 开始。内存控制器将映射地址到适当的缓冲区。通过 boot_buffer_size 和 data_buffer_size 寄存器,用户可以知道缓冲器的最大地址。

在大多数情况下,缓冲区主要用于引导缓冲,用户将使用这个命令从这个缓冲区读取数据。然而,Map

00 命令也用于读/修改/写操作。读/修改/写命令是有顺序的，Map 00 命令在缓冲区可以用于读或写任意字，内存控制器将自动选择 dataram 0 或 dataram 1。如表 7-3 所示，对 Map 00 地址映射进行了详细的描述。

表 7-3 Map 00 地址映射

地址位	名称	描述
31:24	AHB_int_add	AHB 端口地址
23:22	CMD_MAP	00=读或写 XIP 缓冲区
21:17	RESV	保留空间
16:1	MEM_ADDR	存储设备上的缓冲地址
0	BYTE	必须设置为 0

2. “01” =Map 01 命令。

Map 01 命令是用于标准的高速访问存储器数组。通过 FBA 和 FPA 命令的指定，用户可以读和写数据到特殊的数组页。一旦 OneNAND Flash 控制器仅支持页地址（FSA 必须设置为 0）。一个完整的页必须读或写一段时间。命令的实际使用数量，要看数据传输的大小。甚至多个命令被请求，必须使用相同的地址，直到整个块都被发送完。

每读或写数据字都将占用 32 位的大小。异步缓冲器将写入数据分成两个 16 位字词或连接读取数据转化为一个 32 位字。如表 7-4 所示，对 Map 01 地址映射进行了详细的描述。

表 7-4 Map 01 地址映射

地址位	名称	描述
31:24	AHB_int_add	AHB 端口地址
23:22	CMD_MAP	01=读或写到内存设备
21:0	MEM_ADDR	参照表 6-27

3. “10” =Map 10 命令。

Map 10 命令是用于控制特殊功能的存储器设备。这是一个数据通过路径的命令，目的是内存控制器，而不是内存设备。不像其他的命令类型，数据（输入或输出）的这些相关处理，并不影响内存的容量，而是用于指定和执行内存控制器的确切命令。输入和输出数据流将始终为 32 位。但只有低 16 位的数据通道包含相关的信息。如表 7-5 所示，对 Map 10 地址映射进行了详细的描述。

表 7-5 Map 10 地址映射

地址位	名称	描述
31:24	AHB_int_add	AHB 端口地址
23:22	CMD_MAP	10=启动一个特殊功能的 Flash 设备或读状态内存控制器
21:0	MEM_ADDR	参照表 6-27

(1) 擦除操作

OneNand Flash 控制器支持单块或多块擦除。如果内存设备支持多块擦除，擦除是并行执行的。如果内存设备不支持多块擦除，则擦除操作将按顺序执行。当使用多块擦除动作时，用户将指定每一块的地址，然后发出一个单块擦除命令，用于最终启动整个块擦除操作。

实际擦除命令的使用是由 AHB 总线（读或写）和低字节数据输入总线上的处理类型来决定的。如表 7-6 所示。

表 7-6 擦除操作

地址	命令类型	数据输入	功能
[23:22]	写	0x00	保存当前的擦除操作状态到内存控制器
=10 DFS_DBS和 FBA被使用。 FPA 和 FSA 不被使用， 擦除操作 必须被清除。	读	-	如果前面的命令是 Map 10 写入 0x00，则这个命令返回擦除状态。如果前面的命令不是 Map 10 写入 0x00，则如擦除已经完成，返回为 0 0=没有擦除操作在进行，或擦除操作已经完成 1=擦除操作正在进行
	写	0x01	保存多块擦除的块地址。没有启动擦除操作。
	写	0x03	保存单块擦除的块地址，启动擦除操作。也用于指定最终块的多块擦除并启动多块擦除操作。

下面是 ARM11 处理器中，OneNand Flash 控制器擦除块操作的部分代码，参考如下：

//ONENAND_EraseBlock 函数：主要功能实现擦除存储器块。