

## [说明](#)

### [一、GBA系统](#)

### [二、系统配置](#)

### [三、GBA内存](#)

### [四、LCD屏幕](#)

### [五、图像系统](#)

### [六、描绘功能](#)

[字符模式背景](#)

[位图模式背景](#)

[精灵](#)

[精灵和BG优先权](#)

### [七、调色板](#)

### [八、窗口特性](#)

### [九、颜色特效](#)

### [十、声音](#)

### [十一、定时器](#)

### [十二、DMA传送](#)

### [十三、通讯功能](#)

### [十四、按键输入](#)

### [十五、中断控制](#)

### [十六、节能功能](#)

### [十七、系统调用](#)

### [十八、ROM数据](#)

### [附一、数据格式](#)

### [附二、寄存器表](#)



简介

GBA是一款注重便携性和2D图形性能的游戏机，它提供了类似Window的旋转、缩放、α混合、浅入浅出等特性。它有两种位图模式：双倍缓冲模式用于实时重写全屏幕影响，单缓冲模式用于显示静止图像。显示屏是2.9英尺宽的反射TFT彩色液晶。声音方面，除了兼容于GBC之外，GBA有一个PCM立体声发生器，可以通过CPU重叠模拟播放多个轨道。控制器增加了L和R按键。虽然GBA使用的32位RISC CPU在性能和数据处理能力方面都远超过GBC，但是耗电量却少很多，可以连续游戏20小时，因为它混合使用了多种RAM在一个特制的芯片里。GBA的软件可以用C语言编写。总体性能相当于或超过SFC。

手册说明

术语：	（位长度）	
	位长度	术语
	8	字节
	16	半字
图例：	32	字
	（位操作里的位属性）	
	读/写位	只读位
	只写位	
缩写：	固定值位	无限制位
	未使用位	
	（任天堂掌机的缩写）	
	GB	GameBoy
	GBC	GameBoy Color
	GBA	GameBoy Advance

烧录电路：



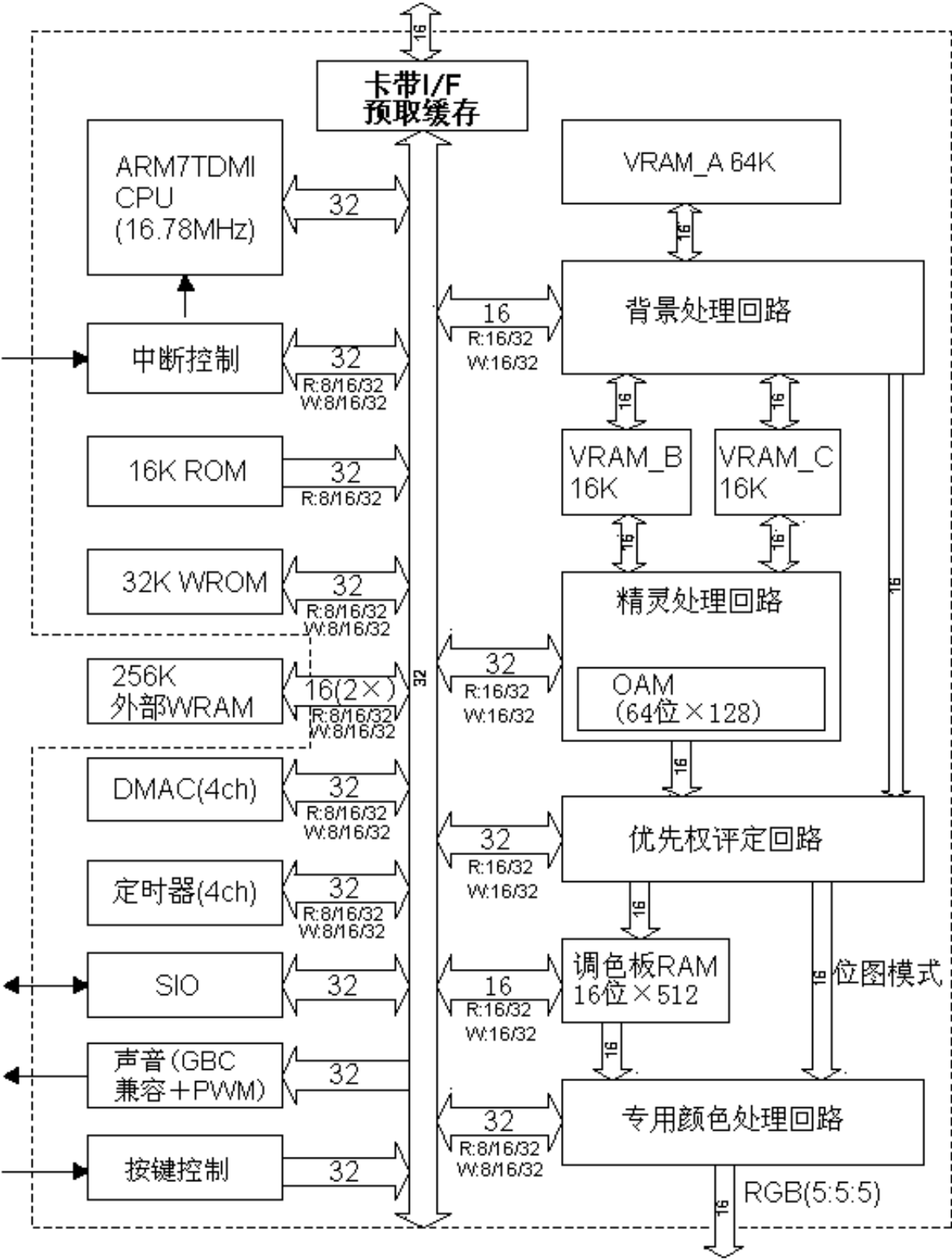
# 一、GBA系统

CPU:	32位RISC CPU（ARM7TDMI）/16.78MHz	
兼容性:	集成8位CISC CPU兼容于GBC，但是不能和GBA的CPU同时工作	
内存:	系统ROM	16K字节（对于GBC是2K）
	工作RAM	32K字节+CPU外部256K字节（2倍周期）
	VRAM	96K字节
	OAM	64位×128
	调色板RAM	16位×512（256色用于精灵，256色用于背景）
	卡带记忆体	最多32M ROM或闪存+最多512Kbit SRAM或闪存
显示:	240×160×RGB点、32,768色模拟显示、特效（旋转、缩放、α混合、浅入浅出和马赛克）、4图像系统模式	
操作:	控制键（A、B、L、R、START、SELECT和方向键）	
声音:	4声道（相应于GBC的声道）+2个CPU直接声道（PCM格式）	
通讯:	串口通讯（8位/32位、UART、多玩家、多用途、JOY总线）	
卡带:	同DMG和GBC一样，GBA的卡带使用32针接口，GBA自动检测插入卡带的类型并切换GBC或GBA模式。GB卡带、GB/GBC双重模式卡带、GBC专用卡带、GBA专用卡带都可以在GBA系统上使用。	

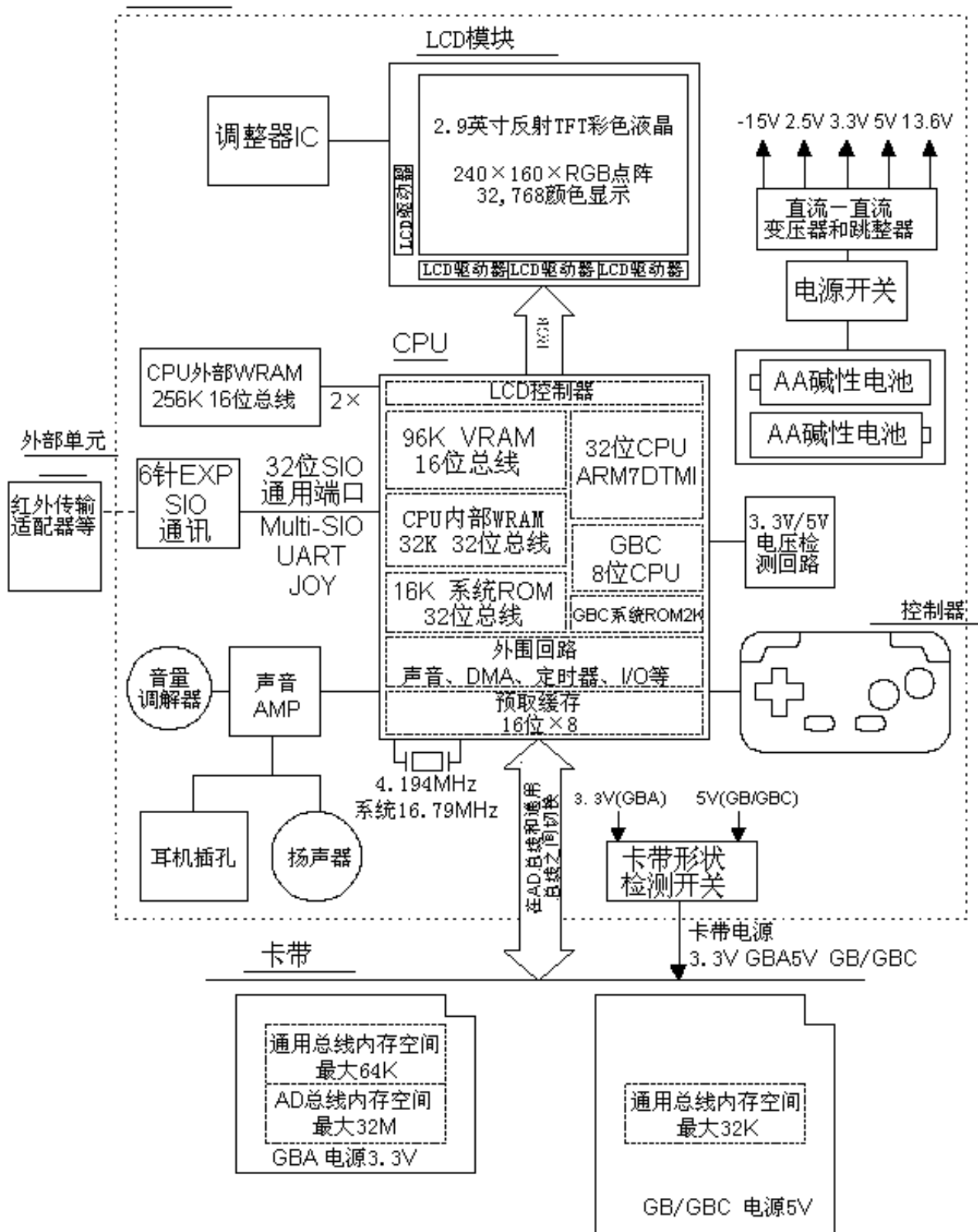


## 二、系统配置

CPU结构图：（R/W： 8/16/32表示可以以8、16、32位的方式读取或写入这个区域）



主机结构图：



## 内存配置和访问带宽

内存类型	总线带宽	DMA		CPU	
		读 带宽	写 带宽	读 带宽	写 带宽
OAM	32	16/32	16/32	16/32	16/32
调色板RAM	16	16/32	16/32	16/32	16/32
VRAM	16	16/32	16/32	16/32	16/32
CPU内部WRAM	32	16/32	16/32	8/16/32	8/16/32
CPU外部WRAM	16	16/32	16/32	8/16/32	8/16/32
内部寄存器	32	16/32	16/32	8/16/32	8/16/32
卡带ROM	16	16/32	16/32	8/16/32	16/32





### 三、GBA记忆体

全部的记忆体映象：

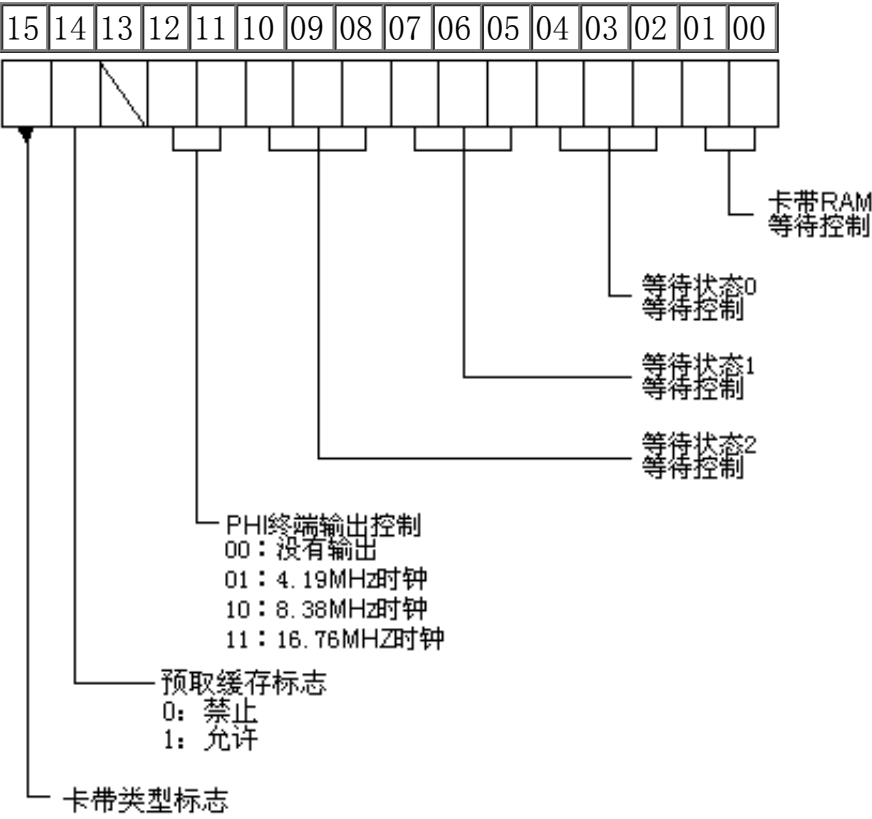
卡带记忆体  （分别设置各段空间的访问速度， 用以优化对卡带ROM的存取。 闪存用于保存数据）	0FFFFFFFh	镜像区域	镜像
	0E00FFFFh	卡带RAM（0—512Kbit）	
	0E000000h		
	0DFFFFFFFh	卡带ROM 等待状态2（32M）	闪存（1Mbit）
	0C000000h		ROM（255Mbit）
	0BFFFFFFFh	卡带ROM 等待状态1（32M）	闪存（1Mbit）
	0A000000h		ROM（255Mbit）
	09FFFFFFFh	卡带ROM 等待状态0（32M）	闪存（1Mbit）
	08000000h		ROM（255Mbit）
	GBA内部记忆体		镜像区域
070003FFh		OAM(1K)精灵属性内存，储存精灵及其属性	
07000000h			
		镜像区域	
06017FFFh		VRAM(96K)	
06000000h			
		镜像区域	
050003FFh		调色板RAM（1K）	
50000000h			
		镜像区域	
04000000h		I/O，寄存器	
		镜像区域	
03007FFFh		CPU内部WRAM（32K）用于储存程序和数据	
03000000H			
		镜像区域	
0203FFFFh		CPU外部WRAM（256K）	
02000000h			
	未使用		
00003FFFh	系统ROM（16K）包含多种系统调用		
00000000h			

卡带记忆体等待控制：

32M的卡带记忆体分配的地址空间是从08000000h开始的，但是从0A000000h和0C000000h开始都是08000000h的镜像。这些镜像可以分别设置卡带记忆体的访问速度（1—4等待周期），用来

和GBA配合。

地址：204h； 寄存器名称：WSCNT； 属性：读/写； 初始值：0000h



说明：

[d12-11]	应恒设为0				
[d10-8]/[d7-5]/[d4-2]	等待状态控制和等待周期的关系如下：				
	等待状态控制的值	等待周期			
		第一次访问	第二次访问		
			等待状态0	等待状态1	等待状态2
	000	4	2	4	8
	001	3	2	4	8
	010	2	2	4	8
	011	8	2	4	8
	100	4	1	1	1
[d1-d0]	101	3	1	1	1
	110	2	1	1	1
	111	8	1	1	1
	卡带RAM等待控制和等待周期的关系如下：				
	等待控制的值	等待周期			
	00	4			
	01	3			
	10	2			
	11	8			

卡带总线：

卡带有32个引脚，如下表所示：

卡带ROM访问		卡带RAM访问	
序号			

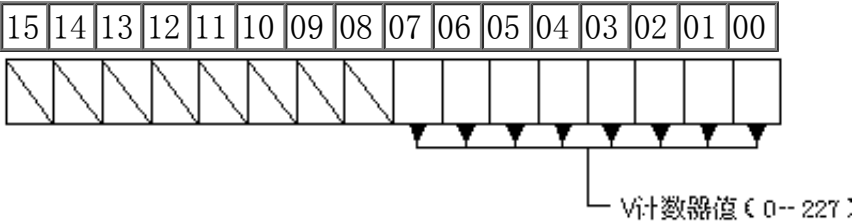
	引脚	用途	引脚	用途
1	VDD (3. 3V)		VDD3. 3V	
2	PHI		PHI	
3	/WR	写标志	/WR	写标志
4	/RD	读标志	/RD	读标志
5	/CS	ROM芯片选择	/CS	
6	AD0	用于数据和低位地址	A0	地址
7	AD1		A1	
8	AD2		A2	
9	AD3		A3	
10	AD4		A4	
11	AD5		A5	
12	AD6		A6	
13	AD7		A7	
14	AD8		A8	
15	AD9		A9	
16	AD10		A10	
17	AD11		A11	
18	AD12		A12	
19	AD13		A13	
20	AD14		A14	
21	AD15		A15	
22	A16	高位地址	D0	数据
23	A17		D1	
24	A18		D2	
25	A19		D3	
26	A20		D4	
27	A21		D5	
28	A22		D6	
29	A23		D7	
30	/CS2		/CS2	RAM芯片选择
31	/IREQ /DREQ	用于 IREQ和DREQ	/IREQ /DREQ	用于 IREQ和DREQ
32	GND		GND	

## 四、LCD屏幕

### V计数器:

V计数器指示了目前哪条扫描线正在描绘。0—159表示正在描绘之中，160—227表示处于垂直空白期间。

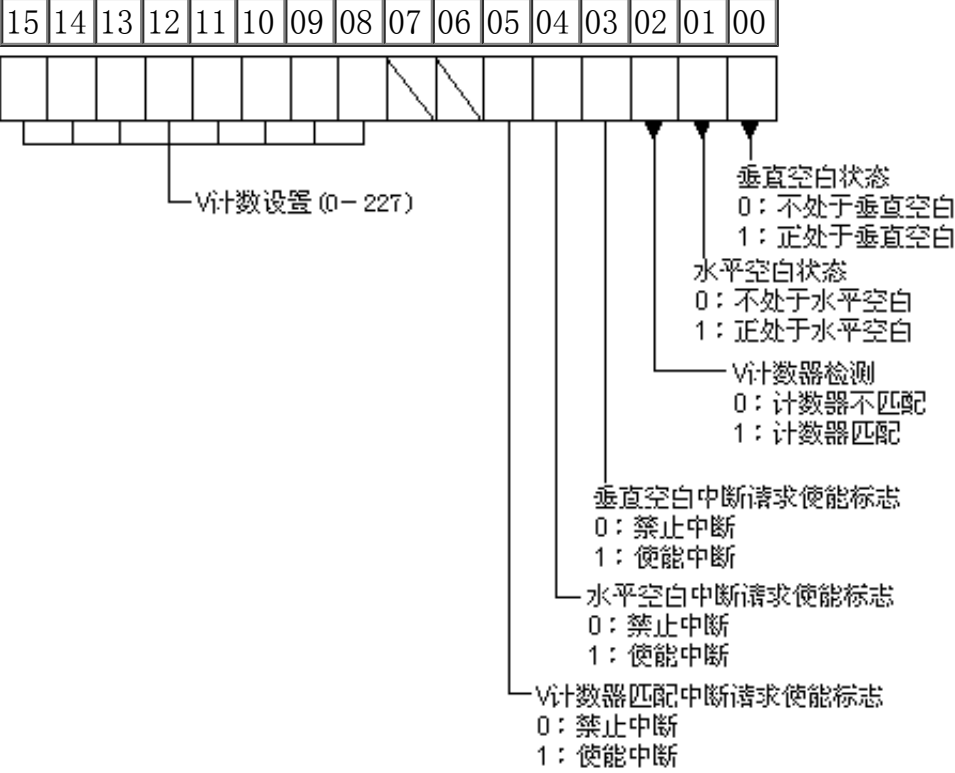
地址：006h；寄存器：VCOUNT；属性：只读；初始值：0000h



### LCD状态:

LCD的状态可以通过读取DISPSTAT寄存器的0—5位来取得。

地址：004h；寄存器：DISPSTAT；属性：读/写；初始值：0000h



### 说明:

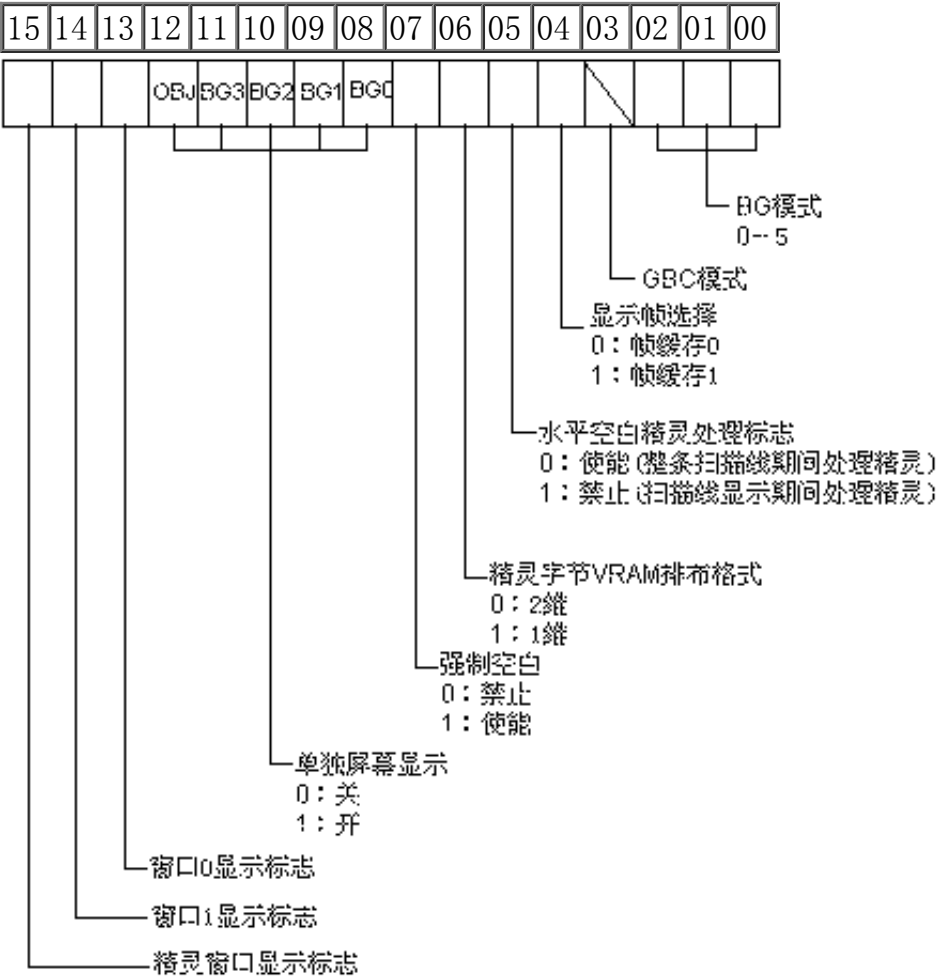
[d15-8]	V计数设置，用于V计数器检测和V计数器匹配中断，数值范围是0—227
[d5]	V计数器匹配中断，当V计数的值和VCOUNT的值相同时产生一个中断
[d4]/[d3]	水平/垂直空白中断，当进入水平/垂直空白时产生一个中断
[d2]/[d1]/[d0]	分别检测V计数器/水平空白/垂直空白状态



五、图像系统

GBA可以根据软件的目的使用不同的图像系统。这些显示相关的项目主要是通过DISPCNT寄存器来改变的。

地址：0000h； 寄存器：DISPCNT； 属性：读/写； 初始值：0080h



说明：

[d15] [d14] [d13]	各种窗口显示的主要标志，请参阅后面的章节
[d12-8]	单独屏幕显示标志，允许单独设置背景0、背景1、背景2、背景3和精灵是否显示
[d7]	强制空白使得CPU强制停止图像处理回路，允许访问VRAM，调色板，OAM和内部寄存器。LCD在强制空白期间显示白色，但是内部的HV同步计数器继续工作。当内部HV同伴计数器在显示期间取消一个强制空白，显示屏会在3条垂直线之后从最开始的地方重新显示
[d6]	精灵字节VRAM排布格式指定了精灵字节在VRAM里的排布格式，请参阅后面的章节
[d5]	水平空白精灵处理标志，当被设为0时，精灵会在整条水平扫描线期间（包括水平空白）被处理；被设为1时，只会在显示扫描线期间被处理，而在水平空白期间不处理。因此，如果要在水平空白期间访问OAM或精灵VRAM就要设置该位，然而就算这样还是不能得到最大精灵显示的性能
[d4]	显示帧选择，当显示位图模式时如果有2个帧缓存（背景模式4和5），这个位允许选择显示哪个帧
[d3]	这个位由系统控制，不能被用户访问
[d2]	进行背景模式选择，从0—5，请参阅后面章节

背景模式：

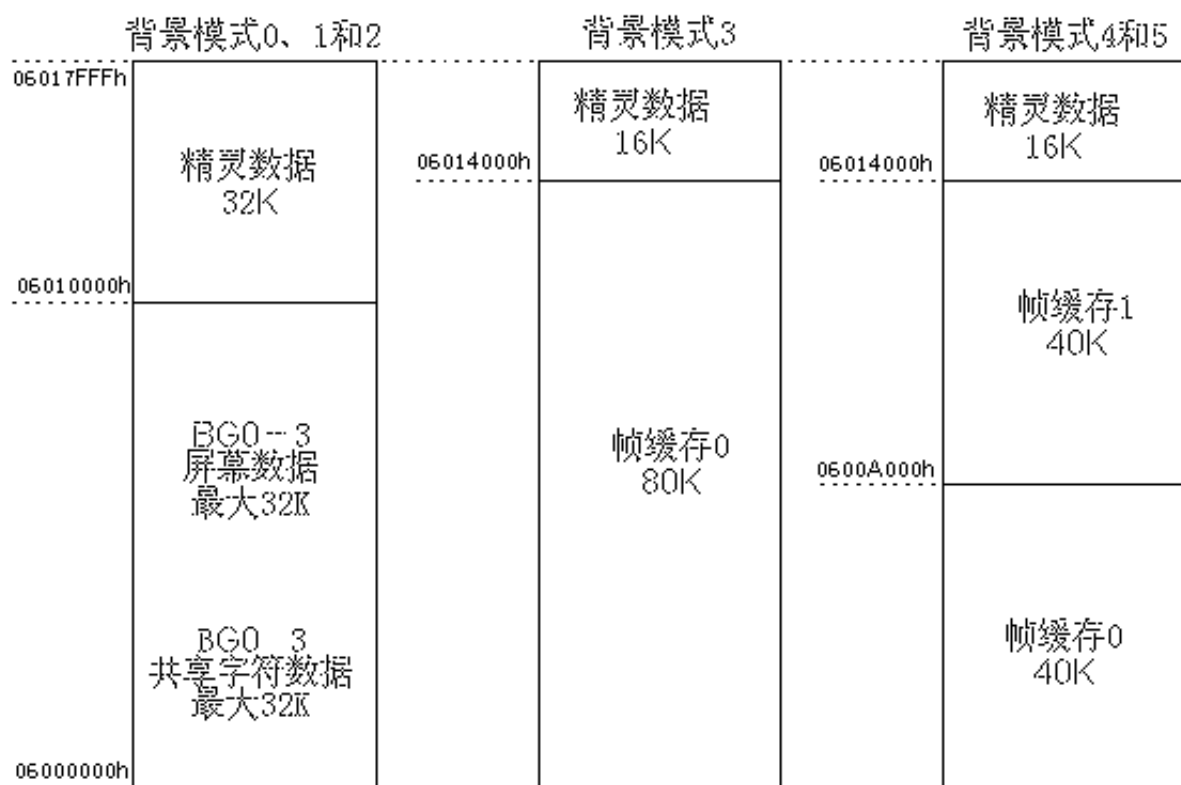
GBA允许选择字符格式和位图格式的背景模式。模式0、1和2使用字符格式描绘LCD屏幕，适合于游戏；模式3、4和5使用位图格式描绘LCD屏幕，适合显示图像。在模式3中，只有一个帧内存，可以显示32,768种颜色，适合于描绘静止图像。模式4和5有两个帧内存，适合描绘动态影像。控制文本背景滚动的方法与背景旋转/缩放和位图背景滚动是不同的，参阅后面章节。

背景模式	字符格式背景屏幕			可列举的字符数	颜色/调色板数	特性					
	旋转/缩放	屏幕个数	大小			HV滚动 (分立屏幕)	HV翻转 (分立字符)	马赛克 (16级)	半透明 (16级)	浅入浅出	屏幕优先级指定
0	否	4	256×256至512×512	1024	16/16 256/1	○	○	○	○	○	○
1	否	2	256×256至512×512	1024	16/16 256/1	○	○	○	○	○	○
	是	1	128×128至1024×1024	256	256/1	○	×	○	○	○	○
2	是	2	128×128至1024×1024	256	256/1	○	×	○	○	○	○
背景模式	位图格式背景屏幕			帧内存	颜色数	特性					
	旋转/缩放	屏幕个数	大小			HV滚动 (分立屏幕)	HV翻转 (分立字符)	马赛克 (16级)	半透明 (16级)	浅入浅出	屏幕优先级指定
3	是	1	241×160	1	32,768	○	×	○	○	○	○
4	是	1	241×160	2	256	○	×	○	○	○	○
5	是	1	160×128	2	32,768	○	×	○	○	○	○

VRAM记忆体映象：

96K的VRAM在不同的背景模式下排布如下：





在0、1和2背景模式时，用户可以映射屏幕和字符数据区域到64K背景区域，参阅后面章节。

六、描绘功能

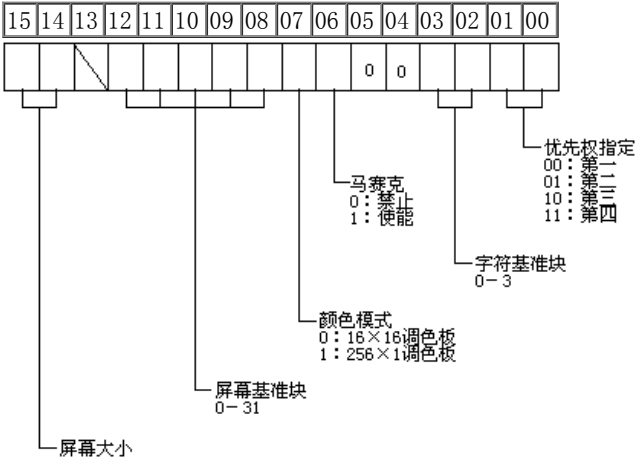
GBA有96K内建VRAM，它可以描绘背景和精灵。描绘背景的方式根据背景模式的不同而不同，如下所述：

字符模式背景（背景模式0—2）：  
在字符模式中，背景屏幕是又8×8点阵的基本字符组成的。

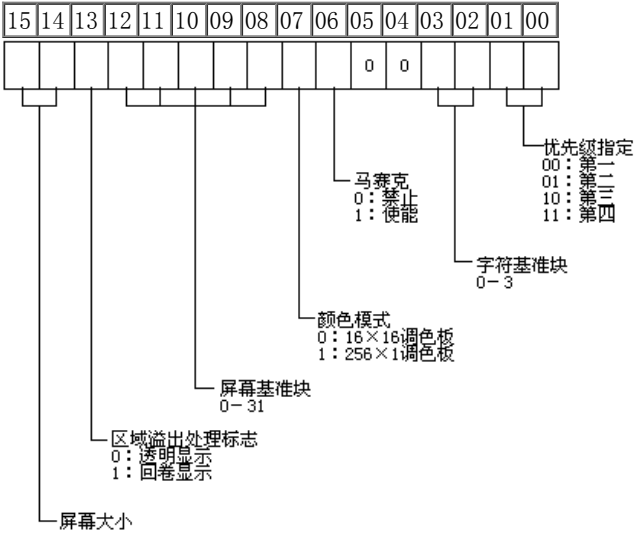
背景控制：  
总共有4个背景控制寄存器，对应于最大4个背景屏幕（寄存器BG0CNT、BG1CNT、BG2CNT和BG3CNT）。寄存器BG0CNT和BG1CNT是文本背景专用的，而BG2CNT和BG3CNT同时支持背景旋转和缩放控制。各种背景模式下寄存器的使用如下：

背景模式	背景控制寄存器			
	BG0CNT	BG1CNT	BG2CNT	BG3CNT
0	BG0 (TEXT)	BG1 (TEXT)	BG2 (TEXT)	BG3 (TEXT)
1	BG0 (TEXT)	BG1 (TEXT)	BG2 (旋转/缩放)	
2			BG2 (旋转/缩放)	BG3 (旋转/缩放)

文本背景控制（BG0和BG1）：  
地址：008h/00Ah； 寄存器：BG0CNT/BG1CNT； 属性：读/写； 初始值：0000h



文本背景和旋转/缩放背景控制（BG2和BG3）：  
地址：00Ch/00Eh； 寄存器：BG2CNT/BG3CNT； 属性：读/写； 初始值：0000h



说明：  
分配背景屏幕大小。当不是最大值时，剩下的VRAM可以用作字符数据区域，参考前面的VRAM内存映像图

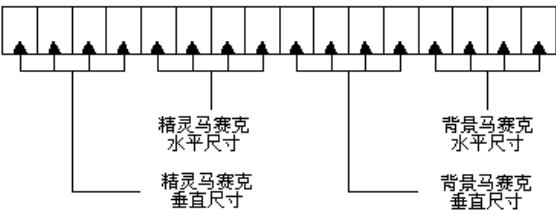
屏幕大小设置	文本屏幕		旋转/缩放屏幕	
	屏幕大小	屏幕数据	屏幕大小	屏幕数据
0	256×256	2K	128×128	256
1	512×256	4K	256×256	1K
2	256×512	4K	512×512	4K
3	512×512	8K	1024×1024	16K

文本背景屏幕大小一览：  
00，虚拟屏幕大小256×256      01，虚拟屏幕大小是512×256

[d15-14]	<p>10, 虚拟屏幕大小是256×512</p>	<p>11, 虚拟屏幕大小是512×512</p>
	<p>10, 虚拟屏幕大小是256×512</p>	<p>11, 虚拟屏幕大小是512×512</p>
旋转/缩放背景屏幕大小示例:		
	<p>00, 虚拟屏幕大小128×128</p>	<p>01, 虚拟屏幕大小256×256</p>
	<p>10, 虚拟屏幕大小512×512</p>	<p>11, 虚拟屏幕大小1024×1024</p>
[d13]	区域溢出处理, 当显示屏幕因为旋转/缩放操作而溢出虚拟屏幕的边界时, 这个位用来选择显示透明还是回卷	
[d12-8]	指定屏幕基准块, 用来指定VRAM里那个块是屏幕的开始位置 (32段: 0-31; 增量是2K字节)	
[d7]	颜色模式, 指定背景字符是用16色×16调色板还是256色×1调色板	
[d6]	打开或关闭马赛克	
[d3-2]	指定VRAM里字符开始显示的块 (4段: 0-3; 增量是16K字节)	
[d1-0]	指定背景优先权, 缺省时 (全部优先权相等) 的优先顺序是BG0、BG1、BG2和BG3。可以改变的值是0(最高)-3。改变优先权时注意指定使用像素的特殊效果	

**马赛克大小:**  
 用MOSAIC寄存器来设置马赛克大小, 为每个背景打开/关闭马克是通过设置背景控制寄存器上的马赛克标志实现的。  
 地址: 04Ch; 寄存器: MOSAIC; 属性: 只写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



**说明：**  
马赛克值指定了多少个普通点组成一个大点显示。从屏幕左上角点开始，和马赛克尺寸相当的点数会用于马赛克显示，除了第一点，其他点都会被马赛克改写，如下图所示。如果马赛克尺寸值是0，则正常显示。

正常显示										马赛克 水平大小:1 垂直大小:1						马赛克 水平大小:3 垂直大小:5																							
00	01	02	03	04	05	06	07	08	09	<table><tr><td>00</td><td>00</td><td rowspan="2">02</td><td rowspan="2">04</td><td rowspan="2">06</td><td rowspan="2">08</td></tr><tr><td>00</td><td>00</td></tr><tr><td>20</td><td>22</td><td>24</td><td>26</td><td>28</td></tr><tr><td>40</td><td>42</td><td>44</td><td>46</td><td>48</td></tr><tr><td>60</td><td>62</td><td>64</td><td>66</td><td>68</td></tr></table>	00	00	02	04	06	08	00	00	20	22	24	26	28	40	42	44	46	48	60	62	64	66	68	00	00	00	00	04	08
00	00	02	04	06	08																																		
00	00																																						
20	22	24	26	28																																			
40	42	44	46	48																																			
60	62	64	66	68																																			
10	11	12	13	14	15	16	17	18	19																														
20	21	22	23	24	25	26	27	28	29																														
30	31	32	33	34	35	36	37	38	39																														
40	41	42	43	44	45	46	47	48	49																														
50	51	52	53	54	55	56	57	58	59																														
60	61	62	63	64	65	66	67	68	69																														
70	71	72	73	74	75	76	77	78	79	<table><tr><td>00</td><td>00</td><td>00</td><td>00</td><td rowspan="5">64</td><td rowspan="5">68</td></tr><tr><td>00</td><td>00</td><td>00</td><td>00</td></tr><tr><td>00</td><td>00</td><td>00</td><td>00</td></tr><tr><td>00</td><td>00</td><td>00</td><td>00</td></tr><tr><td>00</td><td>00</td><td>00</td><td>00</td></tr><tr><td colspan="4">60</td><td></td><td></td></tr></table>	00	00	00	00	64	68	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	60						
00	00	00	00	64	68																																		
00	00	00	00																																				
00	00	00	00																																				
00	00	00	00																																				
00	00	00	00																																				
60																																							

**背景数据的VRAM地址映射：**  
背景数据（背景字符和屏幕数据）储存在VRAM里的64K背景区域里。

**背景字符数据：**  
背景字符数据的开始地址可以通过背景控制寄存器里的字符基准块指定。数据总量由字符个数和数据格式（颜色格式：256色×1调色板或16色×16调色板）确定。

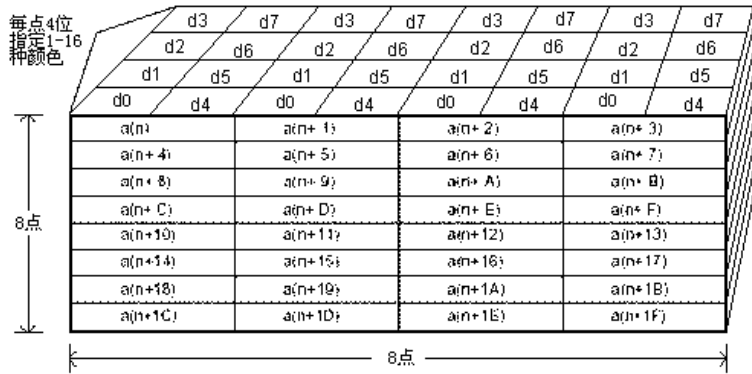
**背景屏幕数据：**  
背景屏幕数据的开始地址可以通过背景控制寄存器里的屏幕基准块指定。数据总量由背景屏幕类型（文本或旋转/缩放）和屏幕尺寸确定，这些都可以通过背景控制寄存器来设定。

背景数据VRAM基准块示意图：  
背景字符数据基准块                      背景屏幕数据基准块

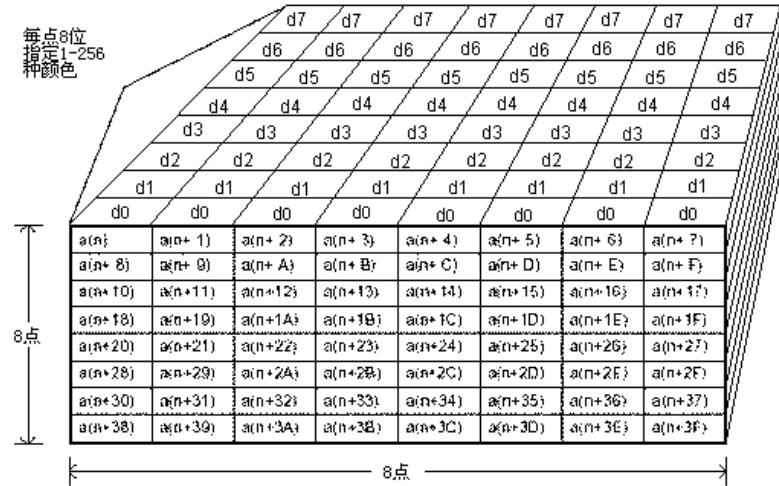
精灵字符数据 32K		精灵字符数据 32K
	10 000h	基准块31
基准块3		基准块30
	C 000h	基准块29
基准块2		基准块28
	8 000h	基准块27
基准块1		基准块26
	4 000h	基准块25
基准块0		基准块24
	0 000h	基准块23
		基准块22
		基准块21
		基准块20
		基准块19
		基准块18
		基准块17
		基准块16
		基准块15
		基准块14
		基准块13
		基准块12
		基准块11
		基准块10
		基准块9
		基准块8
		基准块7
		基准块6
		基准块5
		基准块4
		基准块3
		基准块2
		基准块1
		基准块0

**字符数据格式：**  
有两种字符点数据，16色×16调色板和256色×1调色板。精灵和背景使用同样的格式，在VRAM里的数据如下所示：

16色×16调色板：  
每个地址有2个点，这样，每个基本字符的数据量是20H×8位。



256色×1调色板：  
每个地址有1个点，这样，每个基本字符的数据量是40H×8位。

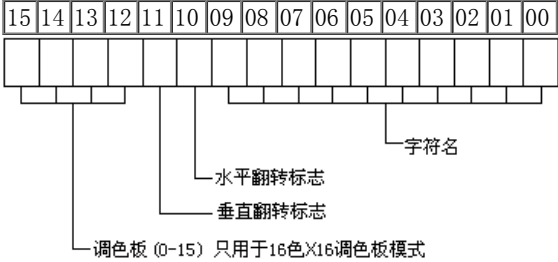


背景屏幕数据格式：

一个背景屏幕是有8×8点阵的基本字符组成的，背景屏幕数据指定了字符的排列。背景屏幕数据应该储存在背景控制寄存器指定的背景屏幕基准块的地址开始的内存里。每个背景屏幕数据条目的个数由背景控制寄存器里的屏幕尺寸指定。文本屏幕和旋转/缩放屏幕格式如下：

文本背景屏幕：

一个文本背景屏幕的每个基本字符包含2个字节的屏幕数据，一共可以指定1024个字符类型。

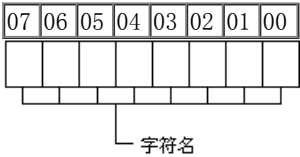


说明：

[d15-12]	如果颜色模式是16色×16调色板的，这些位用来指定字符使用哪个（0—15号）调色板
[d11]/[d10]	设为1则显示垂直/水平反转的字符
[d9-0]	指定从字符基准块开始的字符名（字符编码）

旋转/缩放背景屏幕：

旋转/缩放背景屏幕的每个基本字符包含1个字节的屏幕数据，一共可以指定256个字符类型。字符数据必须被指定为256色×1调色板模式。



注：

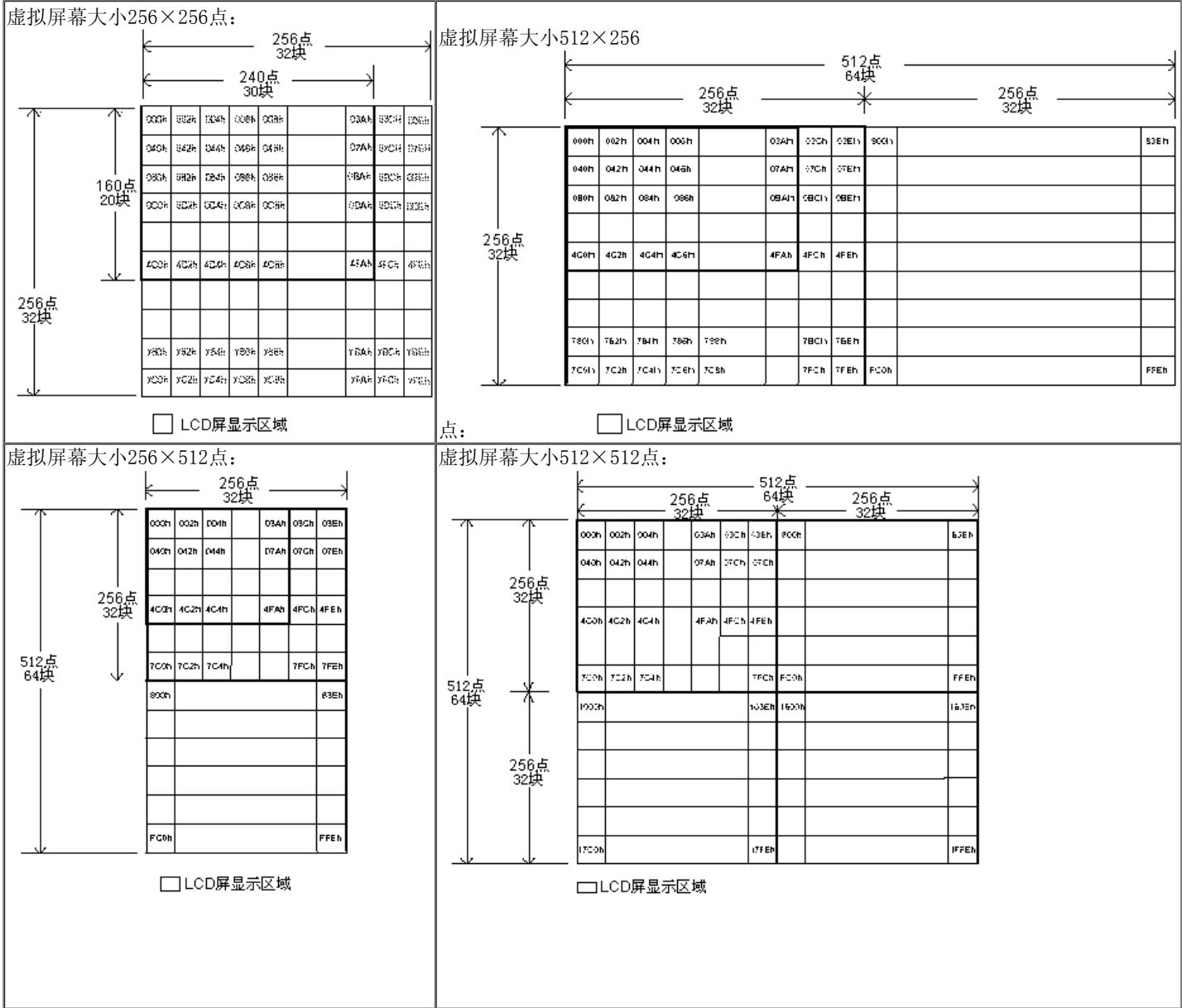
GBA在背景VRAM的使用上提供了很高的自由度，在安排VRAM是要注意一下几点：

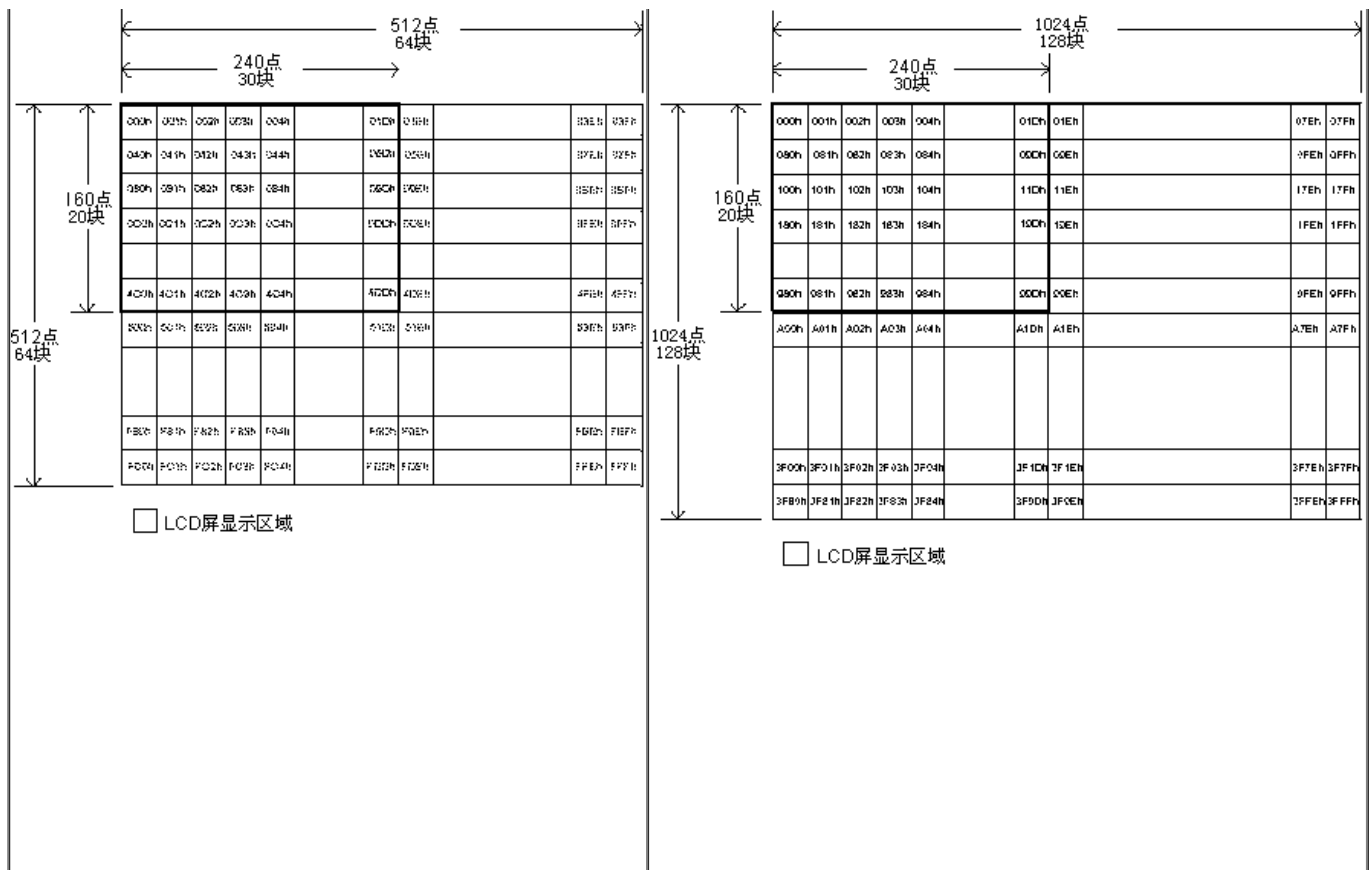
1	有两种格式的背景字符数据（被定义16和256色），这两种格式可以一起使用
2	背景字符数据基准块可以从4个块里选择（通过背景控制寄存器）
3	背景屏幕数据基准块可以从32个块里选择（通过背景控制寄存器）
4	每个背景屏幕尺寸（使用的VRAM中）可以单独设置（通过背景控制寄存器）

文本和旋转/缩放背景可以在一个背景屏幕里同时使用。在安排VRAM时要特别注意背景模式1，因为文本背景屏幕（可以同时处理256色×1调色板和16色×16调色板的背景字符数据）和旋转/缩放背景屏幕（只能处理256色×1调色板）可以同时使用。因此，在编程时要充分理解VRAM映射状态。

LCD屏背景屏幕数据地址映射：

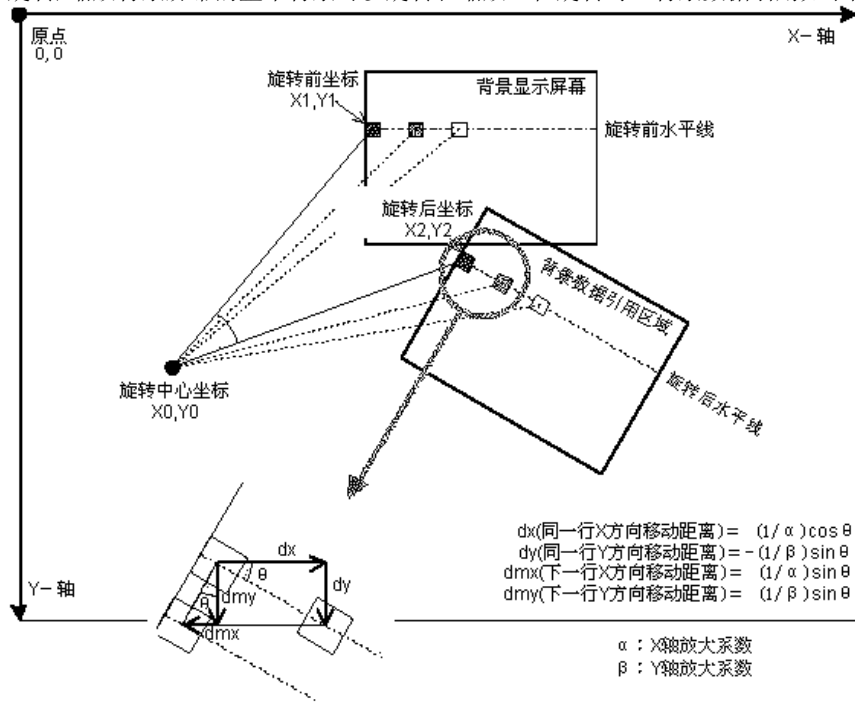
文本背景：





### 背景旋转和缩放特性:

旋转/缩放背景屏幕的整个背景可以旋转和缩放。在旋转时，背景数据引用如下图所示:



背景旋转和缩放在GBA里隐含使用如下数学表达式:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

$$A = \frac{1}{\alpha}\cos\theta, \quad B = \frac{1}{\alpha}\sin\theta, \quad C = -\frac{1}{\beta}\sin\theta, \quad D = \frac{1}{\beta}\cos\theta$$

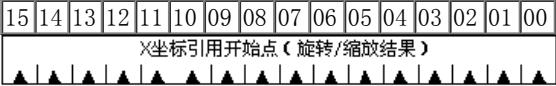
$$x_2 = A(x_1 - x_0) + B(y_1 - y_0) + x_0$$

$$y_2 = C(x_1 - x_0) + D(y_1 - y_0) + y_0$$

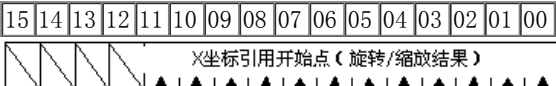
背景2和背景3旋转和缩放时的参数是在一下寄存器里指定的。当旋转/缩放背景和位图模式背景是偏移显示(滚动)时，也使用了背景数据引用寄存器。(文本背景也有一个偏移寄存器)

设置背景数据开始点的寄存器：

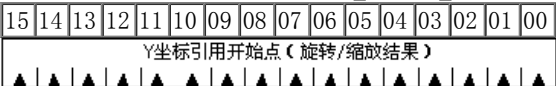
地址：028h/038h； 寄存器：BG2X\_L/BG3X\_L； 属性：只写； 初始值：0000h



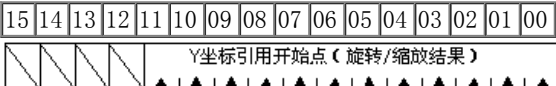
地址：02Ah/03Ah； 寄存器：BG2X\_H/BG3X\_H； 属性：只写； 初始值：0000h



地址：02Ch/03Ch； 寄存器：BG2Y\_L/BG3Y\_L； 属性：只写； 初始值：0000h

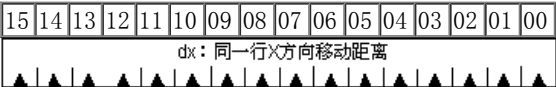


地址：02Eh/03Eh； 寄存器：BG2Y\_H/BG3Y\_H； 属性：只写； 初始值：0000h

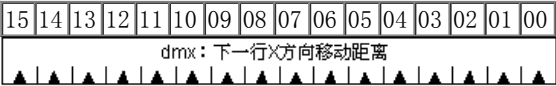


设置背景数据方向参数的寄存器：

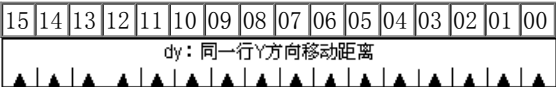
地址：020h/030h； 寄存器：BG2PA/BG3PA； 属性：只写； 初始值：0100h



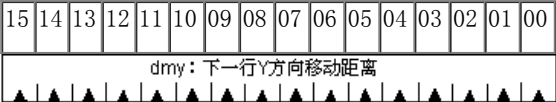
地址：022h/032h； 寄存器：BG2PB/BG3PB； 属性：只写； 初始值：0000h



地址：024h/034h； 寄存器：BG2PC/BG3PC； 属性：只写； 初始值：0000h



地址：026h/036h； 寄存器：BG2PD/BG3PD； 属性：只写； 初始值：0100h



背景旋转/缩放处理的操作：

1	用户使用软件确定显示屏幕左上角坐标的旋转缩放操作结果，并将这些值作为背景数据引用的开始点写到寄存器里（BG2X_L、GB2X_H、BG2Y_L、BG2Y_H、BG3X_L、BG3X_H、BG3Y_L、BG3Y_H）。这些值是带符号的定点数（8位小数，19位整数和1位符号，共28位）。 背景数据引用方向在BG2PA、BG2PB、BG2PC、BG2PD、BG3PA、BG3PB、BG3PC和BG3PD寄存器里设置。这些值是带符号定点数（8位小数，7位整数和1位符号，共16位）。
2	图像处理电路计算X方向（dx、dy），相对于在上述寄存器里设置的，背景数据引用开始点的增量总和，来计算X坐标。
3	要开始下一行时，计算相对于开始点的Y方向增量（dmx，dmy）的总和，来计算下一行描绘开始点的坐标。然后转到第二步继续。
4	但是，如果如果在水平空白期间改写了背景数据引用开始点寄存器，该寄存器Y方向不会被计入总和。CPU使用这种模式来改变每条线的旋转/缩放中心坐标。

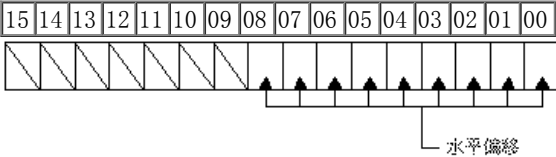
区域溢出处理：

当显示屏幕因为旋转/缩放操作溢出虚拟屏幕边界时，可以用这个背景控制寄存器选择发生溢出的屏幕区域是透明还是回卷。见前面。

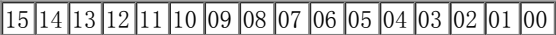
背景滚动：

对每个文本背景屏幕，可以以1点为增量指定显示屏幕的偏移。偏移寄存器只在文本背景时有效。要偏移显示旋转/缩放背景和位图模式背景，请参阅前面章节。

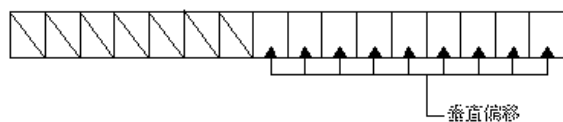
地址：010h/014h/018h/01Ch； 寄存器：BG0H0FS/BG1H0FS/BG2H0FS/BG3H0FS； 属性：只写； 初始值：0000h



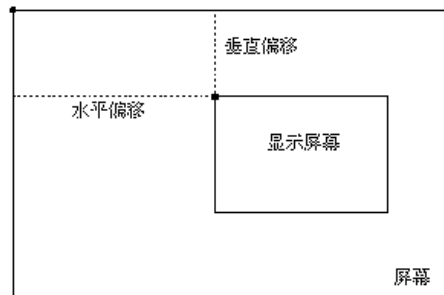
地址：012h/016h/01Ah/0ECh； 寄存器：BG0V0FS/BG1V0FS/BG2V0FS/BG3V0FS； 属性：只写； 初始值：0000h







偏移示意图：



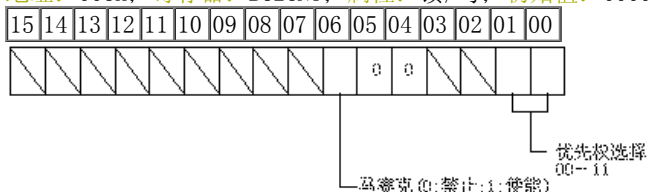
### 位图模式背景：

在位图模式里，背景屏幕以像素为单位处理的，VRAM（帧缓存）的内容以颜色数据显示在屏幕的每一点上。

### 背景控制：

位图背景即BG2。为了显示帧缓存的内容到LCD屏幕，必须将DISPCNT寄存器的BG2显示标志打开。BG2CNT寄存器用来控制BG。

地址：00Ch；寄存器：BG2CNT；属性：读/写；初始值：0000h



### 说明：

[d6]	马赛克开关
[d1-0]	因为位图模式只有一个背景平面，所以在背景里没有优先权可言，但是还是可以设置精灵优先权，参阅后面章节

### 背景旋转/缩放：

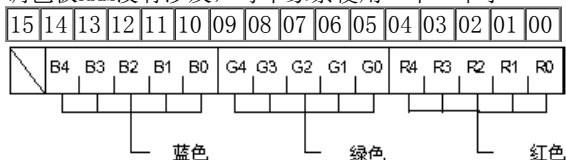
位图背景旋转/缩放参数使用和GB2相关的寄存器（BG2X\_L、BG2X\_H、BG2Y\_L、BG2Y\_H、BG2PA、BG2PB、BG2PC和BG2PD）。参阅前面章节。对于位图BG，如果显示部分因为旋转/缩放操作超出屏幕边界，那部分区域会变成透明。

### 像素数据：

在位图模式，只有和显示屏幕相当数量的像素数据能存在VRAM里。可用的位图模式允许同时显示32,768色（BG模式3和5）和显示32,768里的256色（BG模式4）。不同模式时帧内存里的数据格式也不同，如下所述：

### 32,768色同时显示格式（BG模式3和5）：

调色板RAM没有涉及，每个像素使用一个‘半字’



### 256色（32,768选）显示格式（BG模式4）：

引用调色板RAM的颜色数据（储存了32,768色里的256种），每个像素使用一个字节



### LCD屏幕的像素数据地址映射：

不同的BG模式有不同的地址映射，如下所示。帧缓存（VRAM）开始地址是06000000h。用CPU访问VRA就要对下列地址加上06000000h

### BG模式3（32,768色，240×160点阵，1帧缓存）：

因为是单帧缓存，这个模式主要用来显示静止图像。它能够在全屏幕用32,768色同时显示。

	0	1	2	3	4		236	237	238	239
0	0h	2h	4h	6h	8h		1D8h	1Dah	1DCh	1DEh
1	1E0h	1E2h	1E4h	1E6h	1E8h		3B8h	3Bah	3BCh	3BEh
2	3C0h	3C2h	3C4h	3C6h	3C8h		598h	59Ah	59Ch	59Eh
3	5A0h	5A2h	5A4h	5A6h	5A8h		778h	77Ah	77Ch	77Eh
4	780h	782h	784h	786h	788h		958h	95Ah	95Ch	95Eh
156	12480h	12482h	12484h	12486h	12488h		12658h	1265Ah	1265Ch	1265Eh
157	12660h	12662h	12664h	12666h	12668h		12838h	1283Ah	1283Ch	1283Eh
158	12840h	12842h	12844h	12846h	12848h		12A18h	12A1Ah	12A1Ch	12A1Eh
159	12A20h	12A22h	12A24h	12A26h	12A28h		12BF8h	12BFAh	12BFCh	12BFEh

VRAM地址 + 06000000h

BG模式4（256色，240×160点阵，2帧缓存）：

在VRAM里分配了2帧缓存，使得这个模式适合全动作影像。整个屏幕能同时显示全部32,768色里的256色。

	0	1	2	3	4		236	237	238	239
0	0h	1h	2h	3h	4h		EDh	EDh	EEh	EFh
1	F0h	F1h	F2h	F3h	F4h		1DCh	1DDh	1DEh	1DFh
2	1E0h	1E1h	1E2h	1E3h	1E4h		2CCh	2CDh	2CEh	2CFh
3	2D0h	2D1h	2D2h	2D3h	2D4h		3BCh	3BDh	3BEh	3BFh
4	3C0h	3C1h	3C2h	3C3h	3C4h		4ACh	4ADh	4AEh	4AFh
156	9240h	9241h	9242h	9243h	9244h		932Ch	932Dh	932Eh	932Fh
157	9330h	9331h	9332h	9333h	9334h		941Ch	941Dh	941Eh	941Fh
158	9420h	9421h	9422h	9423h	9424h		950Ch	950Dh	950Eh	950Fh
159	9510h	9511h	9512h	9513h	9514h		95FCh	95FDh	95FEh	95FFh

帧0

VRAM地址 + 06000000h

帧1

VRAM地址 + 06000000h

	0	1	2	3	4		236	237	238	239
0	A000h	A001h	A002h	A003h	A004h		A0ECh	A0EDh	A0EEh	A0EFh
1	A0F0h	A0F1h	A0F2h	A0F3h	A0F4h		A1DCh	A1DDh	A1DEh	A1DFh
2	A1E0h	A1E1h	A1E2h	A1E3h	A1E4h		A2CCh	A2CDh	A2CEh	A2CFh
3	A2D0h	A2D1h	A2D2h	A2D3h	A2D4h		A3BCh	A3BDh	A3BEh	A3BFh
4	A3C0h	A3C1h	A3C2h	A3C3h	A3C4h		A4ACh	A4ADh	A4AEh	A4AFh
156	13240h	13241h	13242h	13243h	13244h		1332Ch	1332Dh	1332Eh	1332Fh
157	13330h	13331h	13332h	13333h	13334h		1341Ch	1341Dh	1341Eh	1341Fh
158	13420h	13421h	13422h	13423h	13424h		1350Ch	1350Dh	1350Eh	1350Fh
159	13510h	13511h	13512h	13513h	13514h		135FCh	135FDh	135FEh	135FFh

BG模式5（32,768色，160×128点阵，2帧缓存）：

有2帧缓存，能显示32,768色，但是显示区域受到了限制。

	0	1	2	3	4		156	157	158	159
0	0h	2h	4h	6h	8h		136h	13Ah	13Ch	13Eh
1	140h	142h	144h	146h	148h		296h	29Ah	29Ch	29Eh
2	2A0h	2A2h	2A4h	2A6h	2A8h		3B8h	3BAh	3BCh	3BEh
3	3C0h	3C2h	3C4h	3C6h	3C8h		4F8h	4FAh	4FCh	4FEh
4	500h	502h	504h	506h	508h		638h	63Ah	63Ch	63Eh
124	9B00h	9B02h	9B04h	9B06h	9B08h		9C38h	9C3Ah	9C3Ch	9C3Eh
125	9C40h	9C42h	9C44h	9C46h	9C48h		9D78h	9D7Ah	9D7Ch	9D7Eh
126	9D90h	9D92h	9D94h	9D96h	9D98h		9EB8h	9EBAh	9EBCh	9EBEh
127	9EC0h	9EC2h	9EC4h	9EC6h	9EC8h		9FF8h	9FFAh	9FFCh	9FFEh

帧0VRAM地址 + 06000000h

帧1VRAM地址 + 06000000h

	0	1	2	3	4		156	157	158	159
0	A000h	A002h	A004h	A006h	A008h		A136h	A13Ah	A13Ch	A13Eh
1	A140h	A142h	A144h	A146h	A148h		A296h	A29Ah	A29Ch	A29Eh
2	A2A0h	A2A2h	A2A4h	A2A6h	A2A8h		A3B8h	A3BAh	A3BCh	A3BEh
3	A3C0h	A3C2h	A3C4h	A3C6h	A3C8h		A4F8h	A4FAh	A4FCh	A4FEh
4	A500h	A502h	A504h	A506h	A508h		A638h	A63Ah	A63Ch	A63Eh
124	13B00h	13B02h	13B04h	13B06h	13B08h		13C38h	13C3Ah	13C3Ch	13C3Eh
125	13C40h	13C42h	13C44h	13C46h	13C48h		13D78h	13D7Ah	13D7Ch	13D7Eh
126	13D90h	13D92h	13D94h	13D96h	13D98h		13EB8h	13EBAh	13EBCh	13EBEh
127	13EC0h	13EC2h	13EC4h	13EC6h	13EC8h		13FF8h	13FFAh	13FFCh	13FFEh

精灵：

精灵功能一览：

精灵使用字符格式，而不管BG模式是什么。但是根据BG模式的不同，基本字符的数量也是变化的。

条目	功能
显示颜色	16色×16调色板或256色×1调色板，可以混合使用
字符数（8×8点）	1024（16色×16调色板）：BG模式0－2 512（256色×1调色板）：BG模式0－2 512（16色×16调色板）：BG模式3－5 256（256色×1调色板）：BG模式3－5
字符尺寸	8×8到64×64（12种）
屏幕最大数目	128（64×64点转换）
单线最大数目	128（8×8点转换）
颜色特殊效果	垂直水平翻转，半透明，马赛克，优先权指定，精灵窗口

单线精灵显示能力：

上表表示的单线显示能力是最大功效的能力。当显示的精灵从OAM开始的地方连续排列时，可用下面的公式计算精灵单线显示能力：  
（水平点数×4－6）/描绘周期数＝单线可显示精灵数[最大128]

水平点数通常是308点，但当DISPCNT寄存器的水平空白精灵处理标志被设为1时，是240点。‘×4’表示精灵描绘电路对每一点可以使用的周期。‘－6’表示在水平线开始描绘精灵前所需要的处理周期。描绘周期数和相应的单线可显示精灵数如下表所示：

精灵水平尺寸	描绘周期数		单线可显示精灵数	
	普通精灵	旋转/缩放精灵	普通精灵	旋转/缩放精灵
8	8	26	128	47
16	16	42	76	29
32	32	74	38	16
64	64	138	19	8
128（64的两倍尺寸）	X	266	X	4

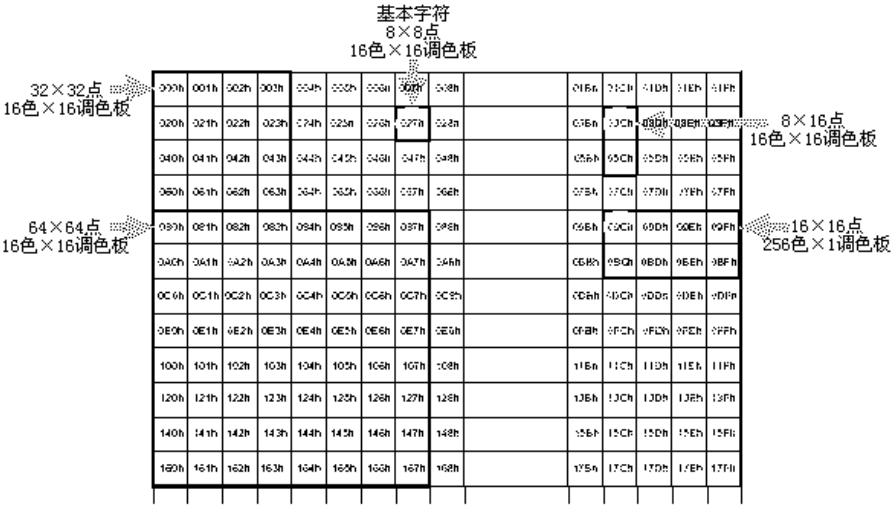
如果OAM里非显示（屏幕之外）精灵的号码低于要显示的精灵，那么非显示精灵的尺寸越大，描绘效率就越低，请注意这点。

精灵数据映射：

精灵字符数据的基本字符是8×8点，8×8点到64×64点之间的字符都可以处理。精灵字符的基址是固定的VRAM基址。精灵字符数据的容量分配或者是32K或者是16K，根据BG模式不同。字符区域有两种映射类型，可以通过DISPCNT寄存器的[d6]位指定。

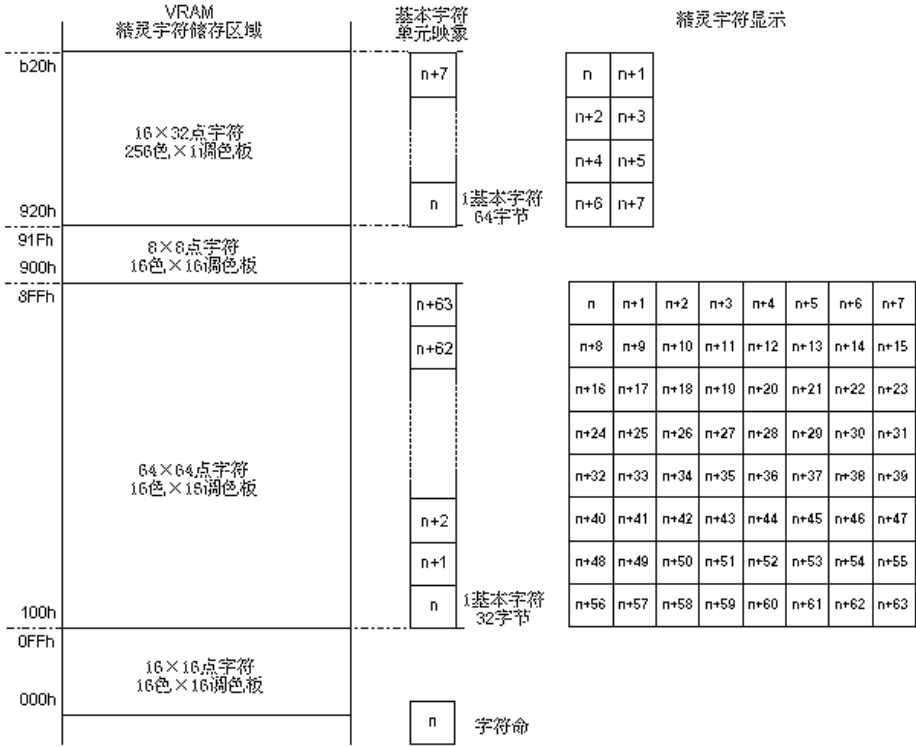
精灵字符VRAM的2维映射：

设置DISPCNT寄存器的[d6]位为0，设置2维映射模式，如下图所示：



■ 字符映射区域（字符码：16进制号码）  
□ 字符码

精灵字符VRAM的1维映射：  
设置DISPCNT寄存器的[d6]位为1，设置1维映射模式，如下图。组成一个字符的数据是储存在连续地址里的。

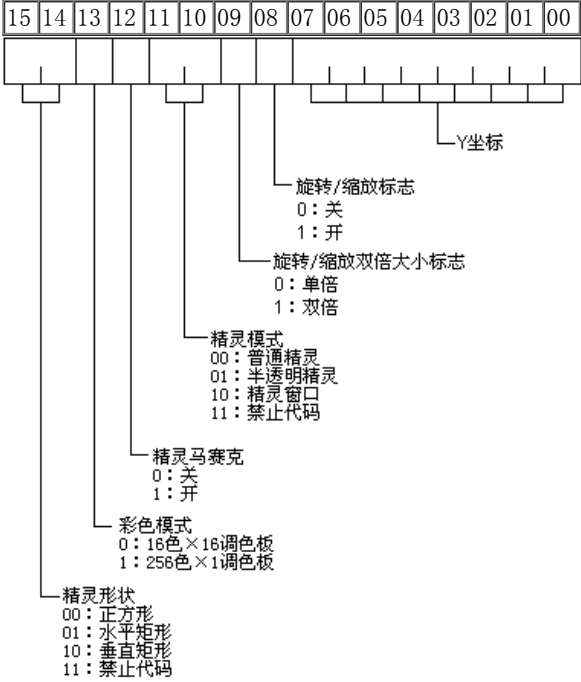


**OAM:**  
精灵是通过OAM里的数据显示的。128个精灵的数据可以写到CPU内部的OAM里（地址是07000000h—070003FFh），任意大小的128个精灵字符可以显示在LCD上。

**OAM映射:**  
精灵属性占据OAM里48（位）×128（精灵）大小空间。另外，当精灵执行旋转/缩放操作时，总共32个旋转/缩放参数结合体实例（PA，PB，PC和PD）可以写到OAM，如下图所示：

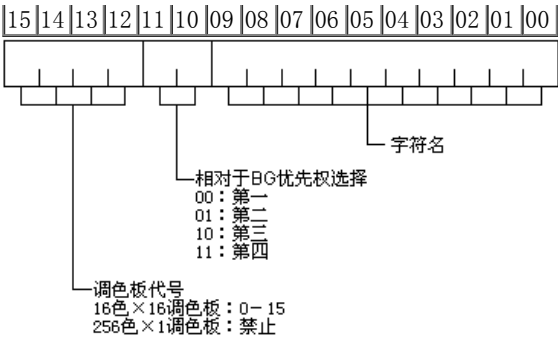


精灵属性0:



[d15-14]	精灵形状，旋转精灵字符的形状：正方形，水平矩形，垂直矩形。11是禁止的代码，请同时参考精灵属性1的精灵尺寸
[d13]	颜色模式标志，指定精灵数据格式是16色×16调色板还是256色×1调色板
[d12]	精灵马赛克标志，打开或关闭精灵马赛克
[d11-10]	精灵模式，指定是普通精灵、半透明精灵还是精灵窗口。11是禁止代码。 当指定半透明精灵时，可以执行颜色特殊效果处理，详细请见后面颜色章节； 当指定精灵窗口时，不显示普通精灵，显示非0字符数据点。
	旋转/缩放双倍大小标志，精灵是限制在精灵区域（8×8—64×64点阵）内的，当旋转时字符数据可能超出这个区域的边界。通过设置双倍大小标志为1，可以虚拟一个双倍大小的精灵区域，来避免出现问题。设置了之后，即使精灵显示被放大到2倍大小也不会越界。示例：64×64点精灵区域→128×128点区域显示，带旋转处理。注意，精灵显示位置偏移了。如果双倍大小标志设为0，突出边界的部分会被砍掉。见下图：

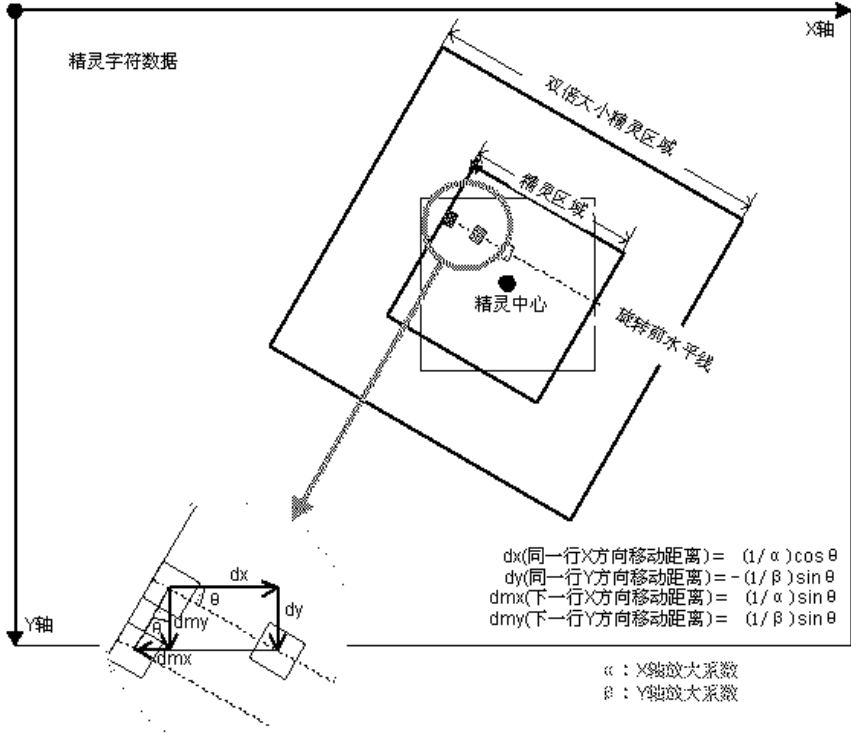




[d15-12]	调色板代号，当精灵颜色模式是16色×16调色板时，用这个指定调色板号；在256色×1调色板时不使用
[d11-10]	相当于BG优先权选择，指定精灵相对于BG的优先权，参阅前面章节
[d9-0]	写入排列在VRAM里的精灵字符数据的开始基本字符号码 16色×16调色板（颜色模式=1）：允许选择1024个字符 256色×1调色板（颜色模式=0）：允许选择512个字符 在2维映射模式，0位固定为0 BG模式3-5时，精灵字符数据RAM平分为16KB，所以0-511号码禁用，使用512和大于512的号码

精灵旋转/缩放特性：

精灵的旋转/缩放特性和BG的旋转/缩放基本一样。下图为精灵旋转示例：



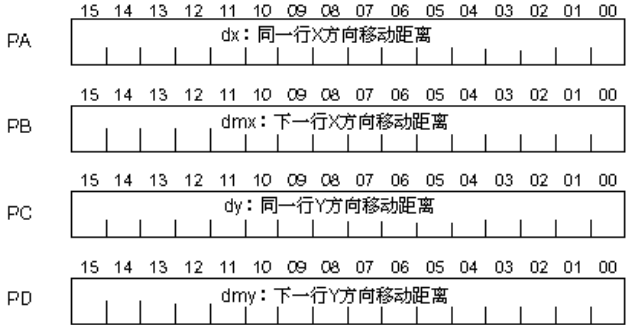
当显示一个精灵时，从左上角位置开始，精灵字符数据按水平方向被调用。旋转显示可以通过增加一个角度到调用方向来实现。旋转中心被固定在精灵区域的中心。如果调用点超出指定的精灵大小就会变成透明。

精灵旋转/缩放处理的操作：

1	指定OAM里的旋转/缩放参数号码到精灵属性1
2	图像处理电路计算相对于旋转中心（精灵区域中心）X方向（dx, dy）的总增量，旋转中心作为参考点，计算X方向坐标
3	当一行完了后，累加相对于旋转中心Y方向（dmx, dmy）的增量，来计算下一个描绘行开始点的坐标，然后执行第2步

旋转/缩放参数：

在精灵旋转/缩放处理中指定字符数据引用方向（见后面章节）。PA、PB、PC和PD里的值是带符号的定点数（8位小数，7位整数和1位符号，共16位）。4个参数作为一个组一齐使用，可以放在OAM里32个指定区域的任何一个里。

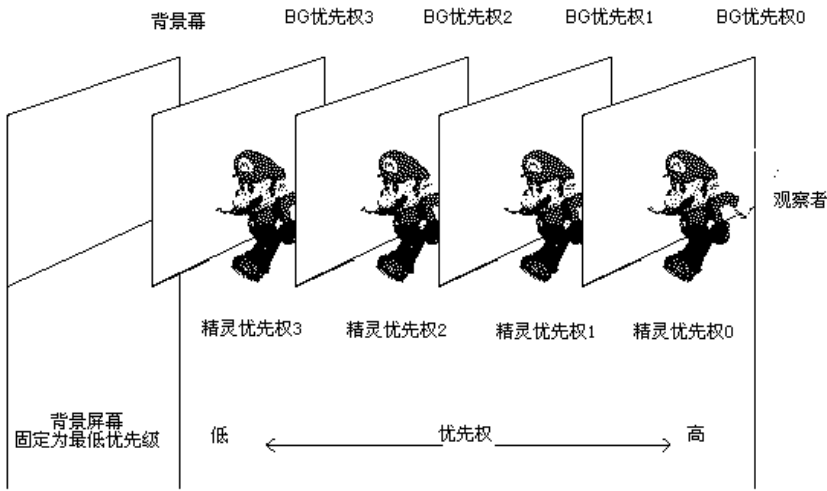


精灵和BG显示优先权：

BG间的优先权：  
BG之间的优先权可以被设为4个层次，当BG优先权相同时，BG号码最低的优先权高。

精灵间的优先权：  
精灵之间的优先权可以被设为4个层次，当精灵优先权相同时，精灵号码最低的优先权高。

BG和精灵间的优先权：  
精灵相对于BG的优先权有4个层次，参见下图：





# 七、调色板

GBA的LCD单元可以显示32级红色，32级绿色和32级蓝色，共32768色。一次能显示的颜色数依赖于BG模式，参加前面章节。调色板用来定义字符格式BG和精灵。位图格式BG模式3和5不是调色板格式。参加前面章节。调色板以下面两种形式出现：

## 16色×16调色板：

这种模式提供16个调色板，每个包含16色。精灵和BG调色板的0号颜色强制分配为透明（不能指定颜色）。

## 256色×1调色板：

这个模式分配全部256色到一个调色板。颜色数据由15位（5位红色、5位绿色和5位蓝色）组成。颜色可以从总共32,768色里选择。精灵0号颜色和0号BG颜色强制分配为透明（不能指定颜色）。

## 0号透明色：

0号透明色用来描绘低优先级精灵和BG的象素为透明。0号BG调色板的0号颜色用于指定背景幕，背景幕的优先权是最低的。

## 调色板RAM：

精灵和BG使用不同的调色板。调色板RAM有512字节大，足够容纳16位（32,768色）的256种颜色。精灵和BG调色板的内存映射如下图所示：



精灵和BG都可以选择两种模式（16色×16调色板和256色×1调色板）之一。这些模式引用调色板RAM如下图所示：

16色×16调色板

调色板RAM

调色板0		颜色0
调色板1		颜色1
调色板2		颜色2
调色板3		颜色3
调色板4		...
调色板5		...
调色板6		...
调色板7		颜色13
调色板8		颜色14
调色板9		颜色15
调色板10		
调色板11		
调色板12		
调色板13		
调色板14		
调色板15		

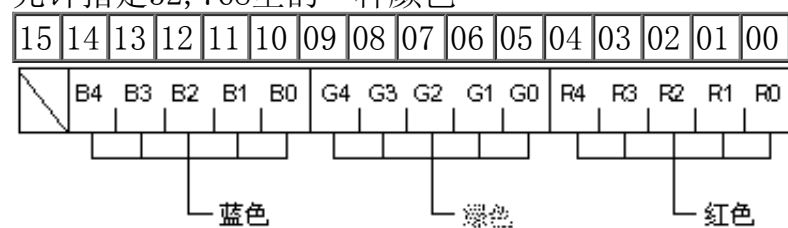
256色×1调色板

调色板RAM

调色板0	颜色0
	颜色1
	颜色2
	颜色3
	颜色4
	...
	...
	...
	颜色252
	颜色253
	颜色254
	颜色255

颜色数据格式:

允许指定32,768里的一种颜色



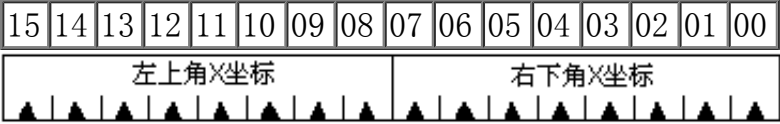
# 八、窗口特性

GBA系统可以同时显示2个窗口。可以单独控制是否显示窗口内和窗口外的东西。并且，每个窗口可以单独执行滚动和颜色特效如旋转，α混合，浅入浅出等。

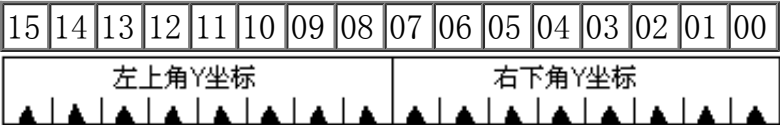
## 设置窗口位置：

窗口位置设置一个矩形区域的左上角和右下角坐标。这些设置指定了窗口的位置和大小。当显示一个非矩形窗口时，这些寄存器的值在水平空白期间被更新。

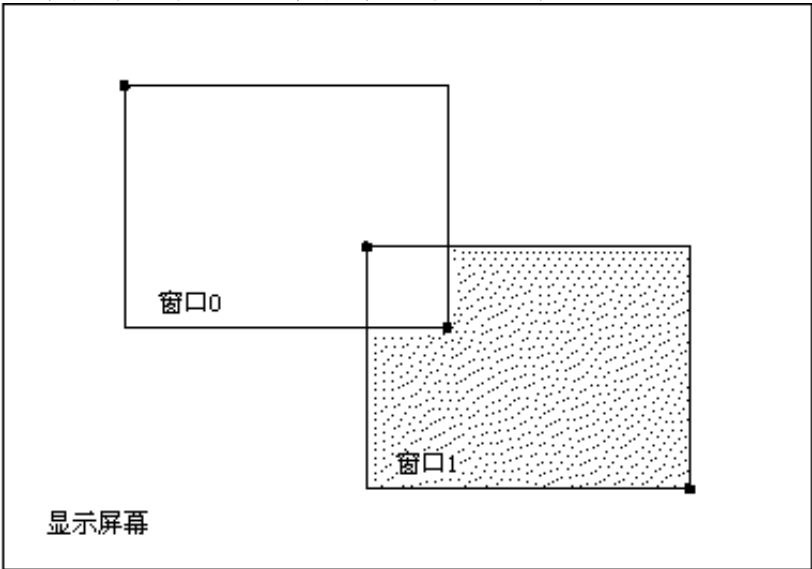
地址：040h/042h；寄存器：WIN0H/WIN1H；属性：只写；初始值：0000h



地址：044h/046h；寄存器：WIN0V/WIN1V；属性：只写；初始值：0000h



显示例子：窗口0的优先权比窗口1高。



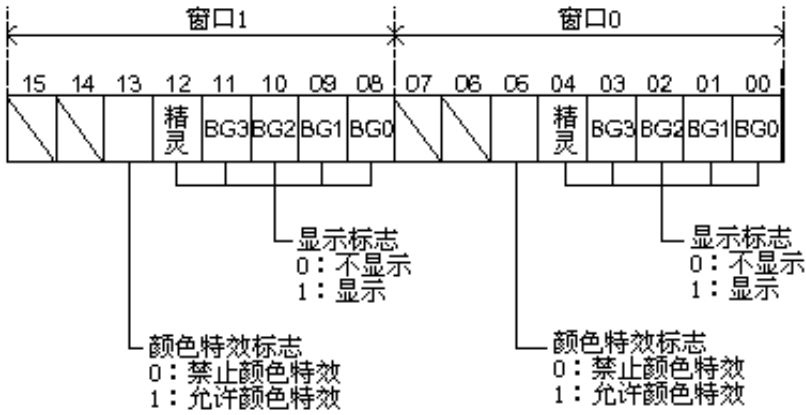
## 窗口控制：

窗口控制寄存器控制窗口显示和关闭之类的操作。但是DISPCNT寄存器里的主窗口显示标志比WININ和WINOUT寄存器优先级高。关于DISPCNT寄存器参加前面章节。

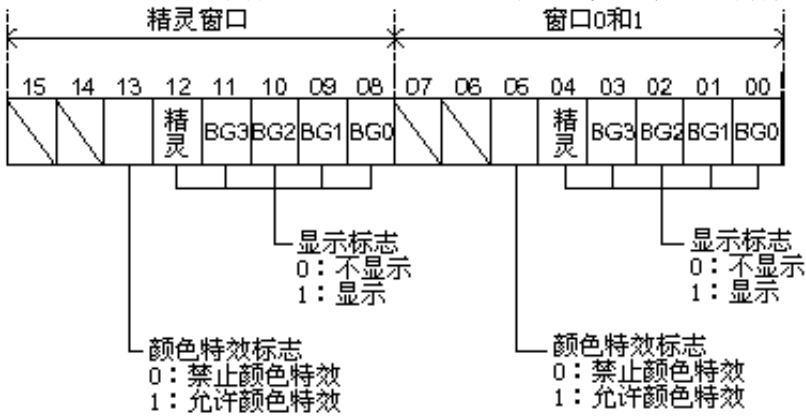
## 窗口内部的控制：

WININ寄存器控制窗口0和1内部区域的显示。高位（d13-8）控制窗口1，地位（d5-0）控制窗口0。

地址：048h；寄存器：WININ；属性：读/写；初始值：0000h



窗口外部和精灵窗口内部控制：  
WINOUT寄存器控制窗口外部区域的显示，同时控制窗口0和1。并且控制精灵窗口内部的区域。  
地址：04Ah； 寄存器：WINOUT； 属性：读/写； 初始值：0000h



WININ[d12-8][d4-0]、WINOUT[d12-8][d4-0]	显示标志，打开（1）或关闭（0）精灵和BG 3—0
WININ[d13][d5]、WINOUT[d13][d5]	颜色特效标志，禁止（0）或允许（1）颜色特效，参见后面章节

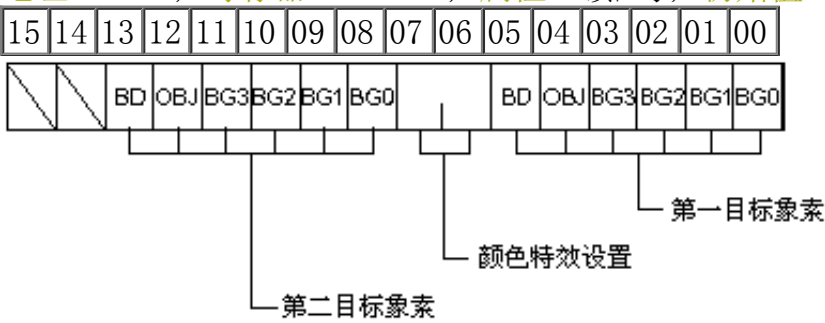
九、颜色特效

GBA提供下面这些颜色特效，可以用窗口指定这些特效应用的区域。

α 混合：对两个选择的表面做算术操作，执行16级的半透明处理。

浅入浅出：对一个选择的表面做算术操作，执行16级亮度处理。

选择颜色特效：  
颜色特效的类型和目标像素由BLDMOD寄存器指定  
地址：050h； 寄存器：BLDMOD； 属性：读/写； 初始值：0000h

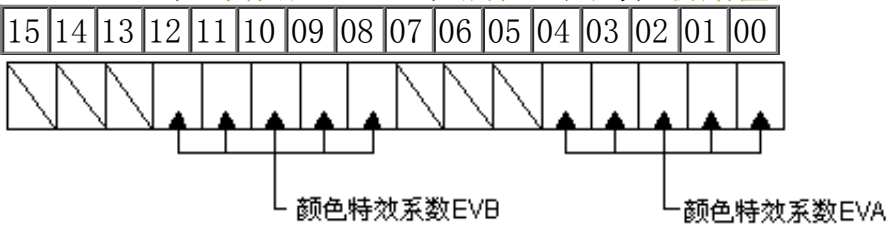


虽然通过BLDMOD寄存器指定了 α 混合颜色特效的2个目标表面，但是两个表明必须要有合适的优先级才能处理成功。

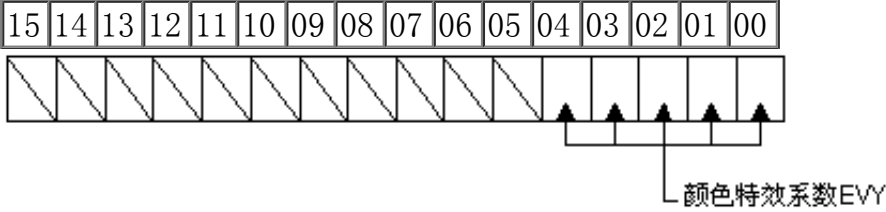
另外，OAM里的半透明精灵是单独指定的，但是BLDMOD寄存器里的颜色特效是对全部精灵的。详细说明见下表：

DLBMOD		类型	颜色特效处理
d7	d6		
0	0	没有特效	通常不执行颜色特效，只有存在半透明精灵并且是紧随在2个目标屏幕之后才执行16级半透明处理(α 混合)
0	1	α 混合 (半透明处理)	如果第一目标屏幕紧随在第二目标屏幕之后，16级半透明处理(α 混合)就执行，第一目标屏幕的底幕位应该被关掉([d5]=0)。当第一目标像素的精灵位=1，对全部精灵进行处理而不够精灵类型；当精灵位=0，只对半透明精灵执行处理。
1	0	亮度增加	逐渐增加第一目标屏幕的亮度。设置第一目标屏幕指定的全部位为1会使得屏幕逐渐变白。当第一目标屏幕精灵位=1，只对普通精灵增加亮度；当一个半透明精灵位于第一目标屏幕，总会执行 α 混合的。
1	1	亮度减少	第一目标屏幕的亮度逐渐减少。设置第一目标屏幕指定的全部位为1会使得屏幕逐渐变黑。当第一目标屏幕精灵位=1，只对普通精灵减少亮度；当一个半透明精灵位于第一目标屏幕，总会执行 α 混合的。

颜色特效处理：  
颜色特效系数  
地址：052h； 寄存器：COLEV； 属性：只写； 初始值：0000h



地址：054h；寄存器：COLY；属性：只写；初始值：0000h



α 混合的系数由COLEB寄存器里的EVA和EVB指定。亮度变化系数由COLY寄存器里的EVY指定。  
EVA、EVB和EVY的值是小于1的数，由一个整数乘以1/16得到。

EVA, EVB, EVY					系数	EVA, EVB, EVY					系数
0	0	0	0	0	0	0	1	0	0	0	8/16
0	0	0	0	1	1/16	0	1	0	0	1	9/16
0	0	0	1	0	2/16	0	1	0	1	0	10/16
0	0	0	1	1	3/16	0	1	0	1	1	11/16
0	0	1	0	0	4/16	0	1	1	0	0	12/16
0	0	1	0	1	5/16	0	1	1	0	1	13/16
0	0	1	1	0	6/16	0	1	1	1	0	14/16
0	0	1	1	1	7/16	0	1	1	1	1	15/16
						1	X	X	X	X	16/16

使用系数的颜色特效算术表达式如下所示：

α 混合（16级半透明）操作：

显示颜色（R）＝第一象素颜色（R）×EVA＋第二象素颜色（R）×EVB

显示颜色（G）＝第一象素颜色（G）×EVA＋第二象素颜色（G）×EVB

显示颜色（B）＝第一象素颜色（B）×EVA＋第二象素颜色（B）×EVB

亮度增加操作：

显示颜色（R）＝第一象素颜色（R）＋（31－第一象素颜色（R））×EVY

显示颜色（G）＝第一象素颜色（G）＋（63－第一象素颜色（G））×EVY

显示颜色（B）＝第一象素颜色（B）＋（31－第一象素颜色（B））×EVY

亮度减少操作：

显示颜色（R）＝第一象素颜色（R）－第一象素颜色（R）×EVY

显示颜色（G）＝第一象素颜色（G）－第一象素颜色（G）×EVY

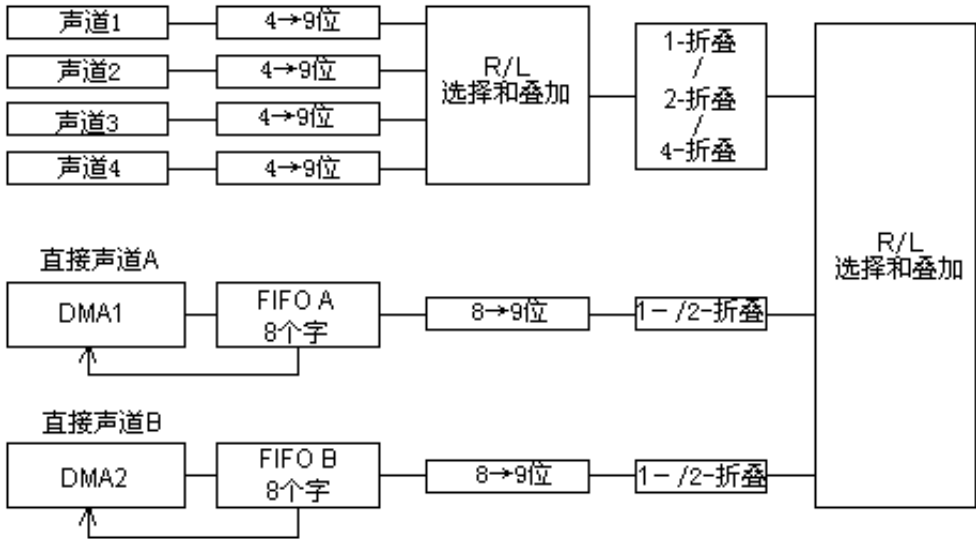
显示颜色（B）＝第一象素颜色（B）－第一象素颜色（B）×EVY

十、声音

除了GBC的4个声道外，GBA还有2个直接声道。

直接声道A和B	提供线性8位声音数据回放，使用定时器和DMA
声道1	可以发生带扫掠（改变频率）和包络（改变幅值）的方波
声道2	可以产生带包络功能的方波
声道3	可以回放记录在波形RAM里的任何波形，GBA的波形RAM容量是GBC的2倍
声道4	可以产生白噪声，可以指定声道1－4到直接声道的合成率

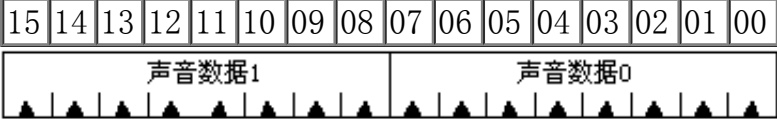
声音结构图：



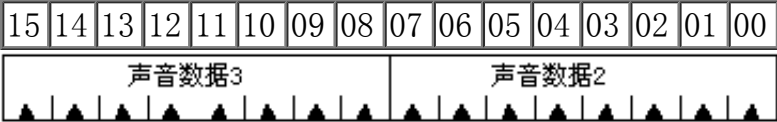
直接声道A和B：

直接声音有2个声道：A和B。可以回放8位线性声音数据。声音数据被设置为00h偏压，是8位数据（+127到-128），即2的补数。声音数据循序地传送到声音FIFO（8个字的容量，P注：FIFO为先入先出缓存），使用DMA1和2的声音FIFO传送模式。采样率可以使用定时器0和1设置为任意值。声音FIFO输入寄存器：

地址：0A0/0A4h；寄存器：SGFIFOA\_L/SGFIFOB\_L；属性：只写；初始值：----



地址：0A2/0A6h；寄存器：SGFIFOA\_H/SGFIFOB\_H；属性：只写；初始值：----



声音数据：

所有的声音都是在声音电路的最后部分采用PWM调制的（参见后面章节）。因此，如果你把8位声音数据采样率和在定时器里设置的PWM调制采样率匹配好，就可以产生一个清晰的声音。下面的操作是直接声音里反复出现的：

准备使用直接声音：

1	使用声音控制寄存器SGCNT0_H（参见后面章节），选择使用的定时器通道（0或1）
2	使用声音控制寄存器SGCNT0_H，对FIFO A和FIFO B清零，并初始音序器。
3	为了在开始直接声音之后立刻产生一个声音，用CPU写第一个8位线性声音数据到FIFO里
4	指定DMA1和2的传输模式（参加后面章节）
5	在声音控制寄存器里指定直接声音输出设置

6开始定时器

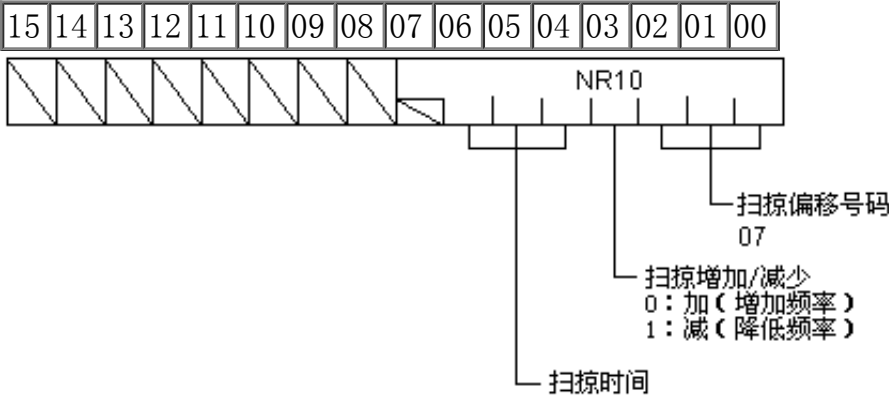
在前面的准备完成后，直接声音如下执行：

1	当指定的定时器因为累计溢出，声音数据从FIFO通到声音电路里
2	在执行传输计数时，如果FIFO里剩下4个字，直接声音A和B的FIFO输出一个数据传输请求到指定的DMA通道
3	如果DMA通道接收一个声音FIFO传输模式的请求，4个字的数据就会提供给声音FIFO（DMA字计算被忽略），然后返回第一步

声道1：

声道1是一个发生带扫掠（调频）和包络（调幅）方波的电路。声道1的NR10、NR11、NR12、NR13和NR14的内容是基于GBC的。

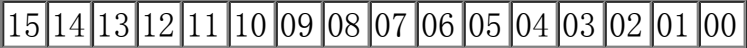
地址：060h；寄存器：SG10\_L；属性：读/写；初始值：0000h



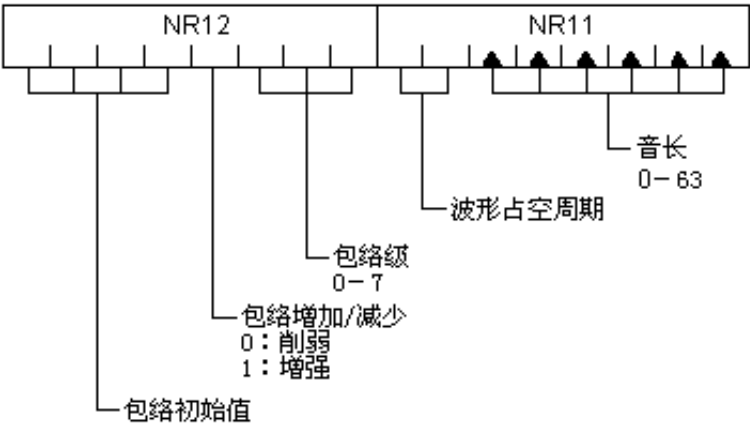
说明：

SG10_L[d6-4]	扫掠时间，这些位指定了频率改变的间隔，f128=128Hz	
	设置	扫掠时间
	000	扫掠停止
	001	1/f128（7.8毫秒）
	010	2/f128（15.6毫秒）
	011	3/f128（23.4毫秒）
	100	4/f128（31.3毫秒）
	101	5/f128（39.1毫秒）
	110	6/f128（46.9毫秒）
	111	7/f128（54.7毫秒）
SG10_L[d3]	扫掠增加/减少，指定频率是升高还是降低，当扫掠功能没有使用时，增/减标志应该设为1	
SG10_L[d2-0]	扫掠偏移号码，带单偏移的的频率数据是根据下面的公式确定的，其中f(t)表示偏移后的频率，f(t-1)表示偏移前的	
	$f_{(t)} = f_{(t-1)} \pm \frac{f_{(t-1)}}{2^n}$	
	f <sub>(t0)</sub> = 初始频率数据	
	如果这个公式的加法产生了一个11位以上的值，声音输出停止，NR52的声道1打开标志（0位）被复位； 在减法时，如果减数小于0，就使用被减数的值。但是，如果指定设置为0，偏移不会发生，频率不改变。	

地址：062h；寄存器：SG10\_H；属性：读/写；初始值：0000h



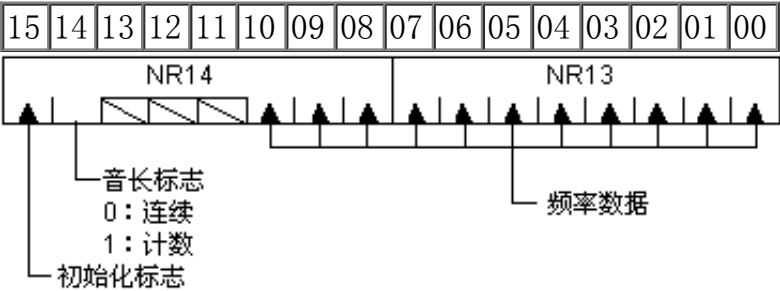




说明：

SG10_H[d15-12]	包络初始值，可以指定从最大值到无声的任意16级															
SG10_H[d11]	包络增/减，指定音量增加或减少															
SG10_H[d10-8]	包络步数，设置包络增强或削弱每步的长度。其中n时指定值，1步（步时间）由下面公式确定： $\text{步时间} = n \times \frac{1}{64} (\text{秒})$ 当n=0时，包络功能被关闭															
SG10_H[d7-6]	波形占空周期，指定波形幅值峰值的比例。 <table><tr><th>设置</th><th>占空周期</th><th>波形</th></tr><tr><td>00</td><td>12.5%</td><td></td></tr><tr><td>01</td><td>25.0%</td><td></td></tr><tr><td>10</td><td>50.0%</td><td></td></tr><tr><td>11</td><td>75.0%</td><td></td></tr></table>	设置	占空周期	波形	00	12.5%		01	25.0%		10	50.0%		11	75.0%	
设置	占空周期	波形														
00	12.5%															
01	25.0%															
10	50.0%															
11	75.0%															
SG10_H[d5-0]	音长，其中st表示音长，输出声音的长度由下面的公式指定： $\text{时间} = (64 - st) \times \frac{1}{256} (\text{秒})$															

地址：064h； 寄存器：SG11； 属性：读/写； 初始值：0000h



说明：

SG11[d15]	初始化表示，设为1则引起声道1复位
SG11[d14]	音长标志，0表示连续输出；1表示只输出NR11里的音长指定时间。声音输出结束时，NR52的声道1打开标志被复位
SG11[d10-0]	频率数据，其中fdat表示频率，输出频率(f)由下面公式确定： $f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)} \text{ Hz}$ ，因此，频率的指定范围是63到131.1千赫

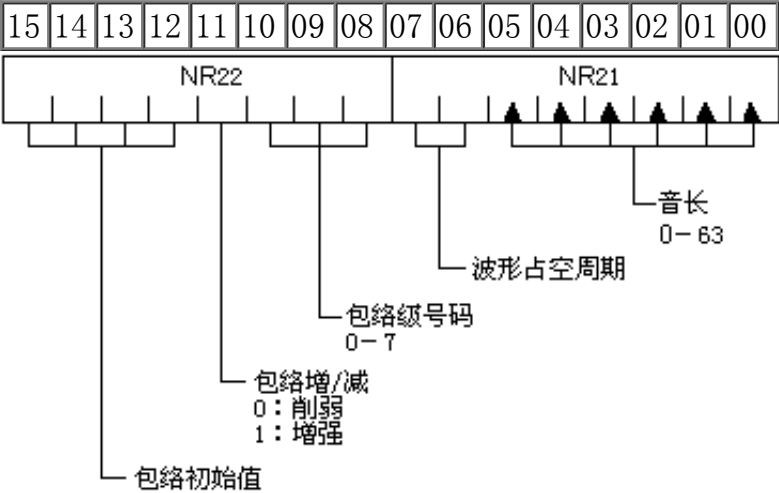
声道1使用注意：

1	当扫掠功能不使用时，扫掠时间应该设为0，扫掠增/减标志应该设为1
2	如果NR10的扫掠增减标志被设为0，扫掠偏移号码应设为非0值，在扫掠关闭模式，声音发生可能被停止
3	当声音执行中（打开标志是1）改变包络寄存器（NR12）时，在包络寄存器中设置数值之后应设置初始化标志

声道2：

声道2是一个产生带包络功能的方波电路。声道2的NR21、NR22、NR23和NR24内容是基于GBC的。

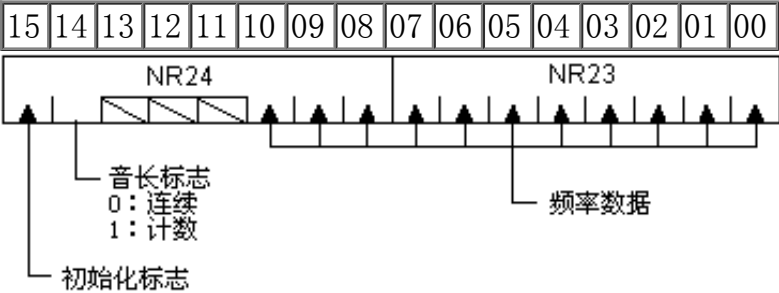
地址：068h； 寄存器：SG20； 属性：读/写； 初始值：0000h



说明：

SG20[d1-12]	包络初始值，可以指定从最大值到无声的16级
SG20[d11]	包络增/减，指定音量增或减
SG20[d10-8]	包络步数，设置包络增强或削弱的一步长度。其中n表示指定值，1步长度（步时间）由下面公式确定： $\text{步时间} = n \times \frac{1}{64} \text{ (秒)}$ 当n=0，包络功能关闭
SG20[d7-6]	波形占空周期，指定波形幅值峰值的比例
SG20[d5-0]	音长，其中st表示音长，输出声音的长度由下面的公式指定： $\text{时间} = (64 - st) \times \frac{1}{256} \text{ (秒)}$

地址：06Ch； 寄存器：SG21； 属性：读/写； 初始值：0000h



说明：

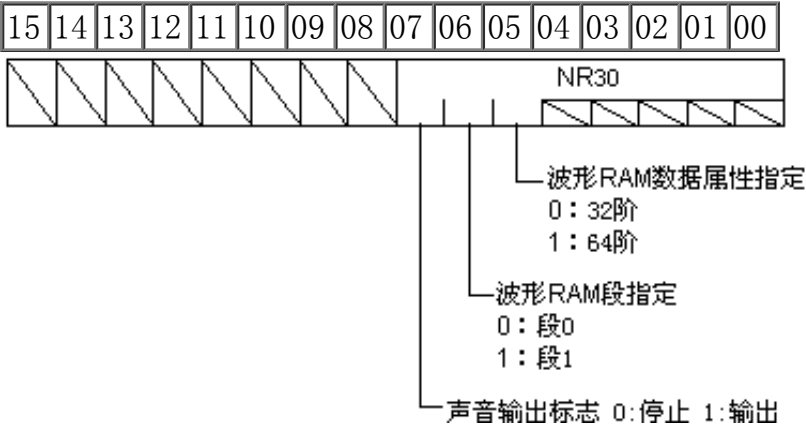
SG21[d15]	初始化标志，设为1使得声道2复位
-----------	------------------

SG21[d14]	音长，0是连续声音输出，1是只输出NR21里音长数据指定的时间。声音输出结束时，NR52的声道2打开标志被复位
SG21[d10-0]	频率数据，其中fdat表示频率数据，输出频率(f)由下面公式确定： $f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)} \text{ Hz}$ ，因此，频率的指定范围是63到131.1千赫

声道2使用注意：  
在声音执行中（打开标志为1）改变包络寄存器（NR22）的内容时，应该在指定包络寄存器的值后设置初始化标志。

声道3：  
声道3是一个可以输出任意波形的电路，并且可以自动从波形RAM里读取波形表（1周期）和修改他们的长度、频率、级别后输出。GBA里声道3的波形RAM的容量（共64阶）是GBC的2倍，同时可以用作2段32阶或1段46阶。并且，现在可以选择一个新的输出级别：3/4输出。

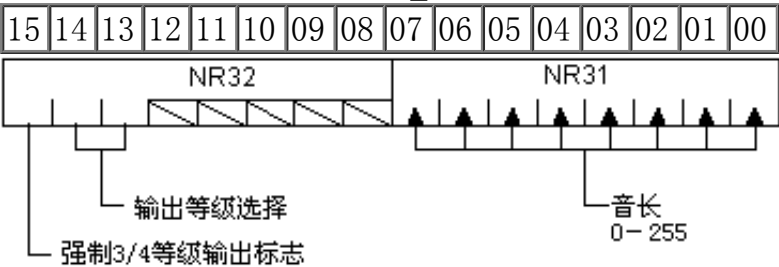
地址：070h； 寄存器：SG30\_L； 属性：读/写； 初始值：0000h



说明：

SG30_L[d7]	声音输出标志，0是停止声音输出，1是输出声音
SG30_L[d6]	波形RAM段选择，有2个波形RAM段：0和1。声道3的电路播放指定段的波形表。当用户访问波形RAM，被访问的是没有指定的那个段。
SG30_L[d5]	波形RAM数据属性指定，指定0时，在普通操作里建立32阶的波形表；设置为1时，NR30[d6]（波形RAM段指定）里的数据被播放，紧随着的是后段。前段32阶和后段32阶联合产生共64阶的波形表。

地址：072h； 寄存器：SG30\_H； 属性：读/写； 初始值：0000h

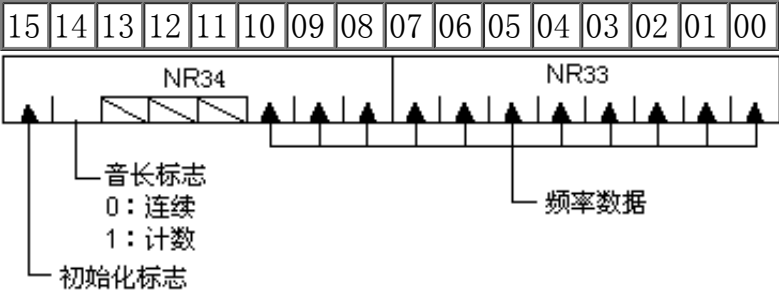


说明：

SG30_H[d15]	强制3/4等级输出标志，为0时，使用NR32[d14-13]指定的等级输出；1时强制输出3/4等级不管NR32[d14-13]的设置	
SG30_H[d14-13]	输出等级选择，声道3的输出等级选择入下表所示：	
	设置	输出等级
	00	静音
	01	不修改而直接输出波形RAM的数据

	10	将波形RAM的数据右移1位（1/2）输出
	11	将波形RAM的数据右移2位（1/4）输出
SG30_H[d7-0]	音长，音长用下面的公式确定，其中st表示音长设置： $\text{时间} = (256 - st) \times \frac{1}{256} \text{ (秒)}$	

地址：074h； 寄存器：SG31； 属性：读/写； 初始值：0000h



说明：

SG31[d15]	初始化标志，当SG30_L[d7]是1，这个位设为1则引起声道3复位
SG31[d14]	音长标志，0时声音连续输出，1时声音只输出NR31指定的音长时间。声音输出结束时NR52的声道3标志被复位
SG31[d10-0]	频率数据，其中fdat表示频率数据，输出频率(f)由下面公式确定： $f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)} \text{ Hz}$ ，因此，频率的指定范围是63到131.1千赫

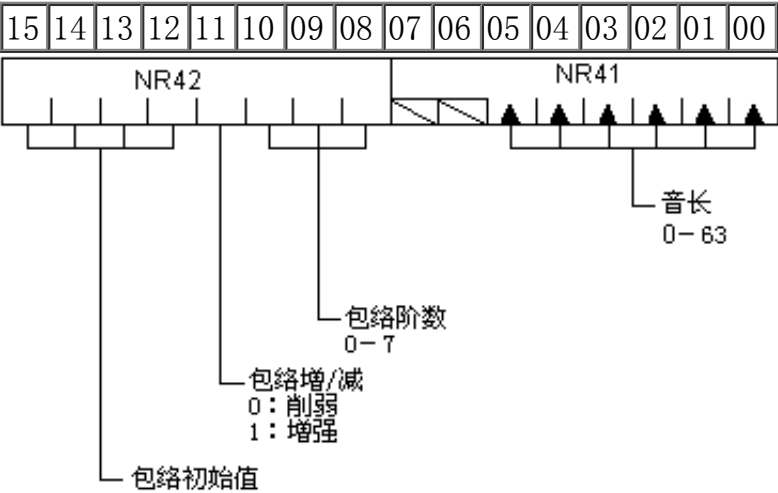
声道3使用注意：  
在声道3输出时改变频率不要设置初始化标志，因为波形RAM里的内容可能被破坏。而在声道1、2和4里，初始化标志可以被设置而不出现问题。

波形RAM：  
波形RAM包含一个4位×32阶的波形表。它有两个段，SG30\_L的[d6]位用来选择段。声道3的电路播放指定段的波形数据的时候，未指定的波形RAM可以被用户访问。

地址	寄存器	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	属性	初始值
090h	SGWR0...L	Step 2				Step 3				Step 0				Step 1				R/W	-
092h	SGWR0...H	Step 6				Step 7				Step 4				Step 5				R/W	-
094h	SGWR1...L	Step 10				Step 11				Step 8				Step 9				R/W	-
096h	SGWR1...H	Step 14				Step 15				Step 12				Step 13				R/W	-
098h	SGWR2...L	Step 18				Step 19				Step 16				Step 17				R/W	-
09Ah	SGWR2...H	Step 22				Step 23				Step 20				Step 21				R/W	-
09Ch	SGWR3...L	Step 26				Step 27				Step 24				Step 25				R/W	-
09Eh	SGWR3...H	Step 30				Step 31				Step 28				Step 29				R/W	-

声道4:  
声道4是一个产生白噪声的电路。声道4的NR41、NR42、NR43和NR44的内容是基于GBC的。

地址：078h；寄存器：SG40；属性：读/写；初始值：0000h



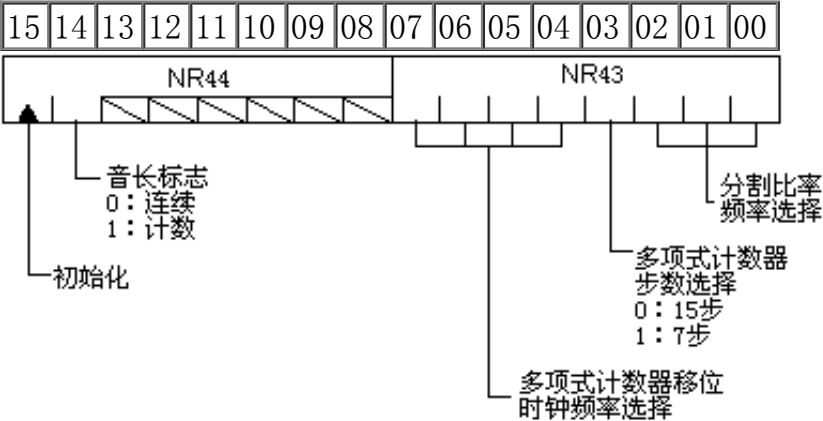
说明:

SG40[d15-12]	包络初始值，可以指定从最大到静音的1给级值
SG40[d11]	包络增/减，指定音量增或减
SG40[d10-8]	包络步数，设置包络增强或削弱的一步长度。其中n表示指定值，1步长度（步时间）由下面公式确定： $\text{步时间} = n \times \frac{1}{64} (\text{秒})$ 当n=0，包络功能关闭
SG40[d5-4]	音长，其中st表示音长，输出声音的长度由下面的公式指定：

SGCNT0[0]

$$\text{时间} = (64 - sf) \times \frac{1}{256} \text{ (秒)}$$

地址：07Ch； 寄存器：SG41； 属性：读/写； 初始值：0000h



说明：

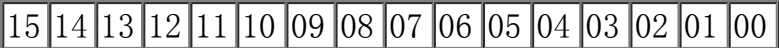
SG41[d15]	设为1将引起声道4复位																		
SG41[d14]	音长，0是连续输出，1只输出NR41指定的音长数据时间。当声音输出结束，NR52的声道4打开标志被复位																		
SG41[d7-4]	多项式计数器移位时钟频率选择，其中n是指定值，移位时钟频率（移位频率）由下面的公式计数得到： $\text{移位频率} = \text{分割比率频率} \times \frac{1}{2^{(n+1)}}$ 但是，%1110和%1111是禁止的代码																		
SG41[d3]	多项式计数器步数选择，0选择15步，1选择7步																		
SG41[d2-0]	分割比率频率选择，选择一个14阶的输入时钟预引比例来产生多项式计数器的移位时钟。 当f = 4.1943.4MHz时，选择如下表所示： <table><tr><th>设置</th><th>分割比率频率</th></tr><tr><td>000</td><td><math>fx1/2^3 \times 2</math></td></tr><tr><td>001</td><td><math>fx1/2^3 \times 1</math></td></tr><tr><td>010</td><td><math>fx1/2^3 \times (1/2)</math></td></tr><tr><td>011</td><td><math>fx1/2^3 \times (1/3)</math></td></tr><tr><td>100</td><td><math>fx1/2^3 \times (1/4)</math></td></tr><tr><td>101</td><td><math>fx1/2^3 \times (1/5)</math></td></tr><tr><td>110</td><td><math>fx1/2^3 \times (1/6)</math></td></tr><tr><td>111</td><td><math>fx1/2^3 \times (1/7)</math></td></tr></table>	设置	分割比率频率	000	$fx1/2^3 \times 2$	001	$fx1/2^3 \times 1$	010	$fx1/2^3 \times (1/2)$	011	$fx1/2^3 \times (1/3)$	100	$fx1/2^3 \times (1/4)$	101	$fx1/2^3 \times (1/5)$	110	$fx1/2^3 \times (1/6)$	111	$fx1/2^3 \times (1/7)$
设置	分割比率频率																		
000	$fx1/2^3 \times 2$																		
001	$fx1/2^3 \times 1$																		
010	$fx1/2^3 \times (1/2)$																		
011	$fx1/2^3 \times (1/3)$																		
100	$fx1/2^3 \times (1/4)$																		
101	$fx1/2^3 \times (1/5)$																		
110	$fx1/2^3 \times (1/6)$																		
111	$fx1/2^3 \times (1/7)$																		

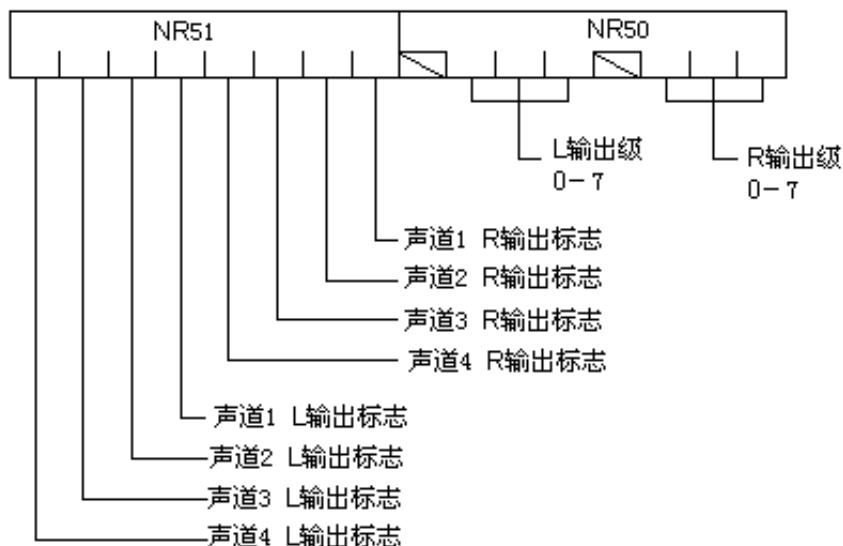
声道4使用注意：  
要在声音在执行时（打开标志为1）改变包络寄存器（NR52）的内容，应该在指定包络寄存器的值后设置初始化标志。

声音控制：

直接声道和声道的输出比率可以在SGCNT0\_H寄存器里设置。最后的声道控制是在SGCNT0\_L寄存器里。NR50和NR51分别是基于GBC里类似的寄存器。

地址：080h； 寄存器：SGCNT0\_L； 属性：读/写； 初始值：0000h

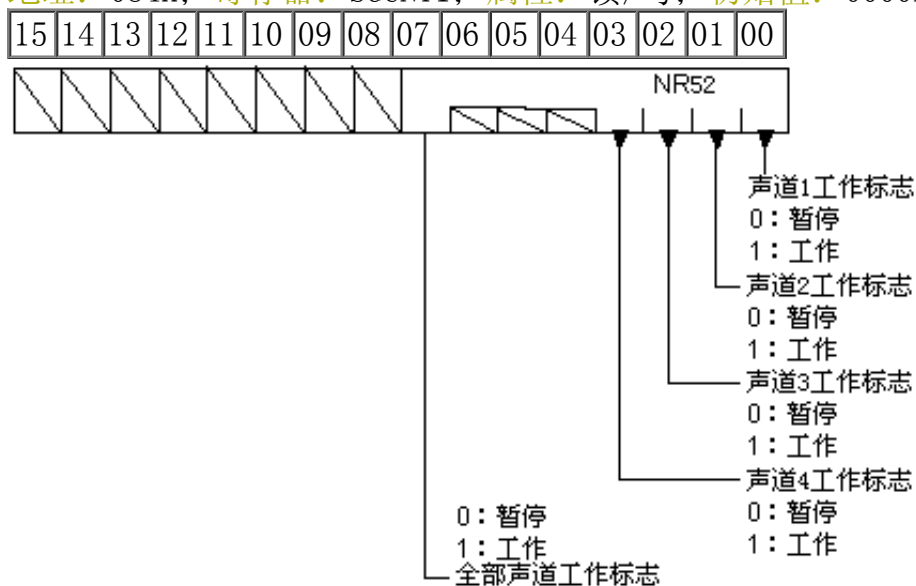




说明:

SGCNT0_L[d15-12]	每个声道的L输出标志, 0时该声道的L没有输出, 1时输出
SGCNT0_L[d11-8]	每个声道的R输出标志, 0时该声道的R没有输出, 1时输出
SGCNT0_L[d6-4]	L输出级, 可以指定8个输出级, 但是对直接声道没有效果
SGCNT0_L[d2-0]	R输出级, 可以指定8个输出级, 但是对直接声道没有效果

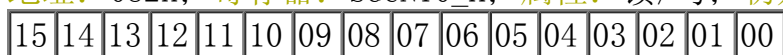
地址: 084h; 寄存器: SGCNT1; 属性: 读/写; 初始值: 0000h

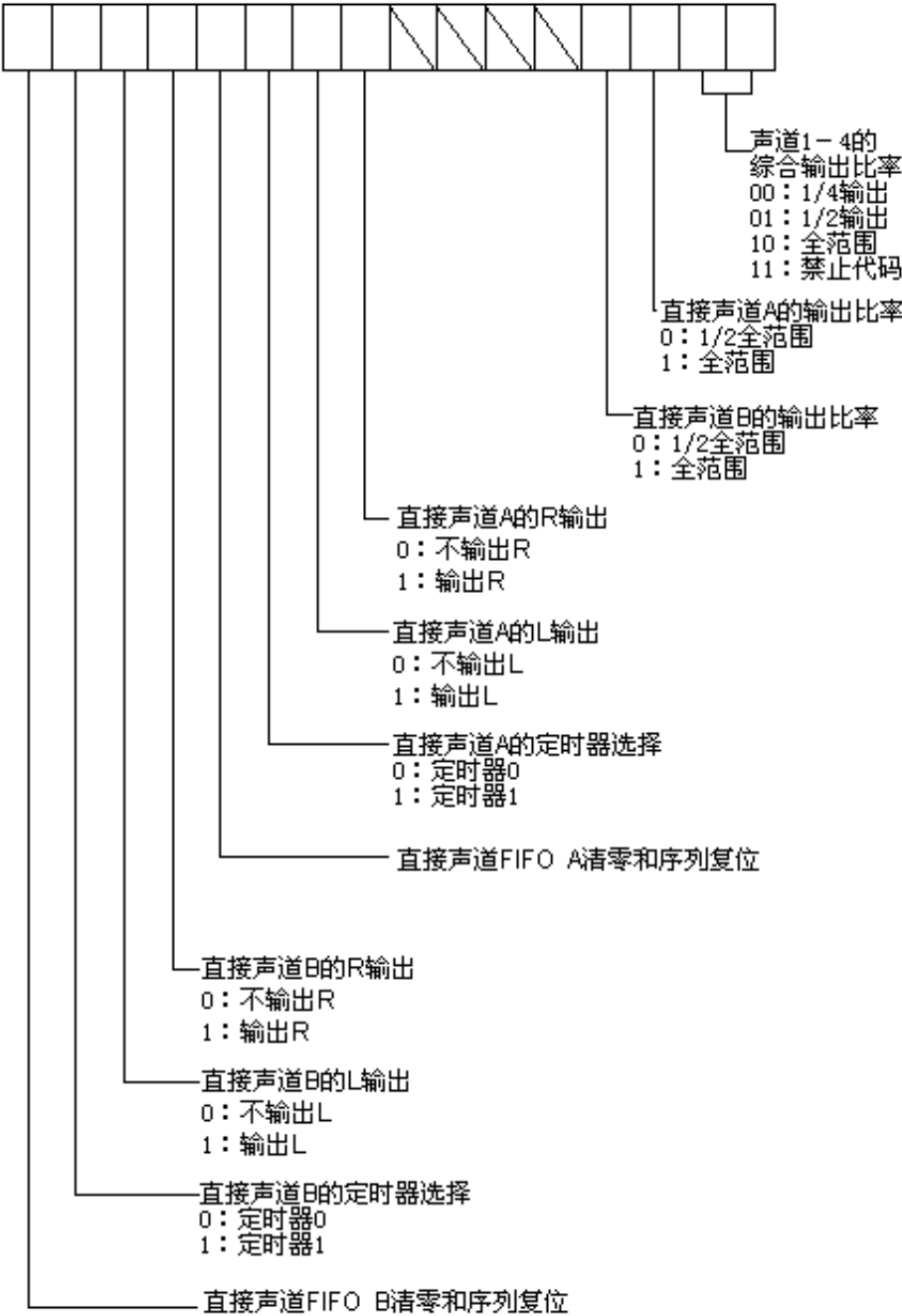


说明:

SGCNT1[d7]	全部声道工作标志, 控制全部声道是否工作的主标志。0时, 所有声道包括直接声道都暂停, 静音状态。这种情况下, 所有声道相关寄存器的内容都复位。1时使能全部声道。
SGCNT1[d3][d2][d1][d0]	声道工作标志, 涉及每个声道电路的状态。在输出时每个声道都被设置, 在计数器模式时经过预设的长度数据时间后复位。

地址: 082h; 寄存器: SGCNT0\_H; 属性: 读/写; 初始值: 0000h





说明:

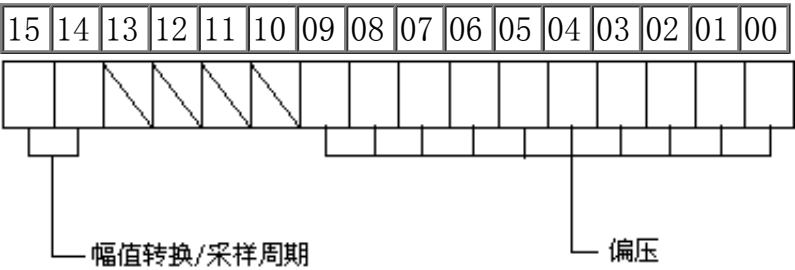
SGCNT0_H[d15] [d11]	FIFO每个直接声道的清零和序列复位。对于直接声道，音序器计算数据从FIFO传送到混音电路的次数。 设为1时，对应的直接声道复位FIFO和音序器。当这个位被读取时，返回0。
SGCNT0_H[d14] [d10]	每个直接声道的定时器选择，指定每个直接声道使用的定时器，0选择0号定时器，1选择1号定时器。 同一个定时器可以被两个声道同时指定。
SGCNT0H[d13] [d9]	每个直接声道的L输出，控制每个直接声道的L输出，0时没有L输出，1是输出L。
SGCNT0H[d18] [d8]	每个直接声道的R输出，控制每个直接声道的R输出，0时没有R输出，1是输出R。
SGCNT0H[d3] [d2]	每个直接声道的输出比率，0产生1/2全范围的输出，1产生全范围输出。
SGCNT0H[d1] [d0]	声道1－4的综合输出比率，00是1/4输出，01是1/2输出，10是全输出，11是禁止的代码

声音PWM控制:

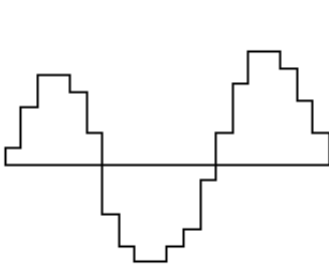
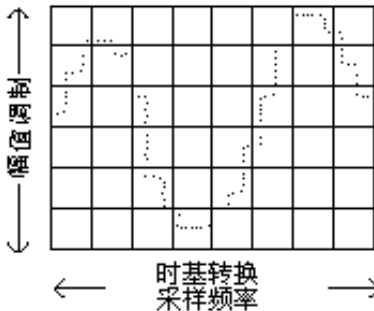
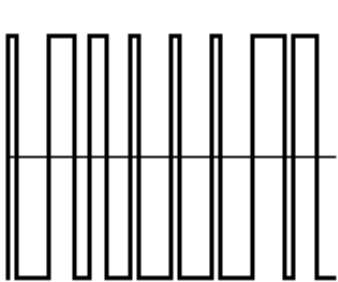


GBA声音电路使用PWM位调制格式。当没有声音产生时，输出占空波形，提供偏压。当设置占空为0h时PWM电路停止。这个寄存器使用系统ROM。这可能引起错误，因此要注意不能写这个寄存器。

地址：088h； 寄存器：SGBIAS； 属性：读/写； 初始值：0200h



说明：

SGBIAS[d15-14]	<p>幅值转换/采样周期，这个设置在PWM调制时的幅值转换和采样周期频率。DMG兼容的声音输入是4位/130.93KHz的，所以为了准确调制，采用率必须设得高。直接声道会被基于定时器设置的采样率随意决定。通过使用下面的采用率列表，准确的调制就会被完成。因此，为了增加声音的真是性，幅值转换需要设得更高。当同时产生兼容声音和直接声音时，请找一个能够使两个声音同时工作的值并设置它。</p>		
	设置	幅值转换	采样率
	00	9位	32.768KHz
	01	8位	65.536KHz
	10	7位	131.072KHz
11	6位	262.144KHz	
<p>PWM转换图像：</p>			
输入波形（由全部声道组成的波形）		PWM调制	CPU输出波形
			
SGBIAS[d9-0]	<p>偏压，这个值是由系统ROM使用的，请不要改变它，因为可能会导致错误</p>		

## 十一、定时器

GBA配备了4个通道的16位定时器。这里面，定时器0和1是用来设置为直接声道A和B的FIFO提供数据的时间间隔的。间隔是通过定时器溢出设置的。

### 定时器设置:

地址：100h/104h/108h/10Ch；寄存器：TM0D/TM1D/TM2D/TM3D；属性：读/写；初始值：0000h

[illegible]

### 定时器控制:

地址：102h/106h/10Ah/10Eh；寄存器：TM0CNT/TM1CNT/TM2CNT/TM3CNT；属性：读/写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00



说明:

TM*CNT[d7]	定时器工作标志，开始和停止定时器。0停止定时器，1开始定时器。
TM*CNT[d6]	中断使能请求标志，控制是否在溢出时发生一个中断请求。0不发生中断，1发生中断。
TM*CNT[d2]	计数间隔，设为0时，根据[d1-0]的预引比例计数。设为1时，忽略预引比例，低一个通道的定时器溢出时计一个数。 这种模式适合于作相对比较长的时间测量。定时器0的计数间隔不能被设置，被指定为预引比例模式。
TM*CNT[d1-0]	预引比例选择，可以选择基于系统时钟（16.78MHz）的预引比例。
	设置 预引比例（计数间隔）
	00 系统时钟（59.95纳秒）
	01 64个系统时钟周期（3.814微秒）
	10 256个系统时钟周期（15.256微秒）
	11 1024个系统时钟周期（61.025微秒）



## 十二、DMA传送

GBA有4个DMA传输通道。优先权最高的是DMA0，往后依次为DMA1、DMA2和DMA3。如果有一个优先权比正在执行的DMA高的DMA开始执行，当前执行的DMA会暂停，在完成高优先权的DMA后，原来的那个从暂停的地方继续执行。对于每个DMA通道最适当的用法如下：

DMAO:

因为它有最高的优先权，不会被其他的DMA通道中断。因此，它适合在一个有限的时间内执行重要的操作，例如水平空白DMA。

DMA1和DMA2:

这些在在直接声道功能上，因为它需要一个相对较高的优先权，或者用在普通目的的传输。

DMA3:

这个用在最通用的传输类型中。

在使用DMA时执行下面这些设置:

1	在源地址寄存器里指定传输源地址
2	在目标地址寄存器里指定传输目标地址
3	在字计数寄存器里设置数据项的数目
4	在DMA控制寄存器里指定使用的传输方法

注意:

当在水平空白期间使用DMA向OAM或精灵VRAM传输的数据时，水平空白必须首先用DISPCNT寄存器从精灵显示硬件处理期间空闲出来，参见前面章节。

DMAO:

DMA0允许主单元内部不同区域的内存互相访问，在DMA通道中具有最高的优先权。

源地址:

使用27位指定源地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址: 0B0h; 寄存器: DMOSAD L; 属性: 只写; 初始值: 0000h

[illegible]

地址: 0B2h; 寄存器: DMOSAD H; 属性: 只写; 初始值: 0000h

[illegible]

目标地址:

使用27位指定目标地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址：0B4h；寄存器：DMODAD L；属性：只写；初始值：0000h

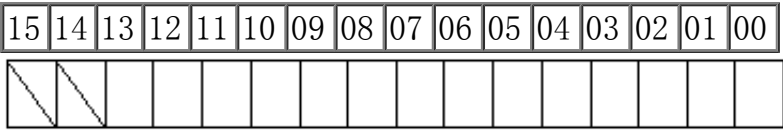
[illegible]

地址：0B6h；寄存器：DMODAD H；属性：只写；初始值：0000h

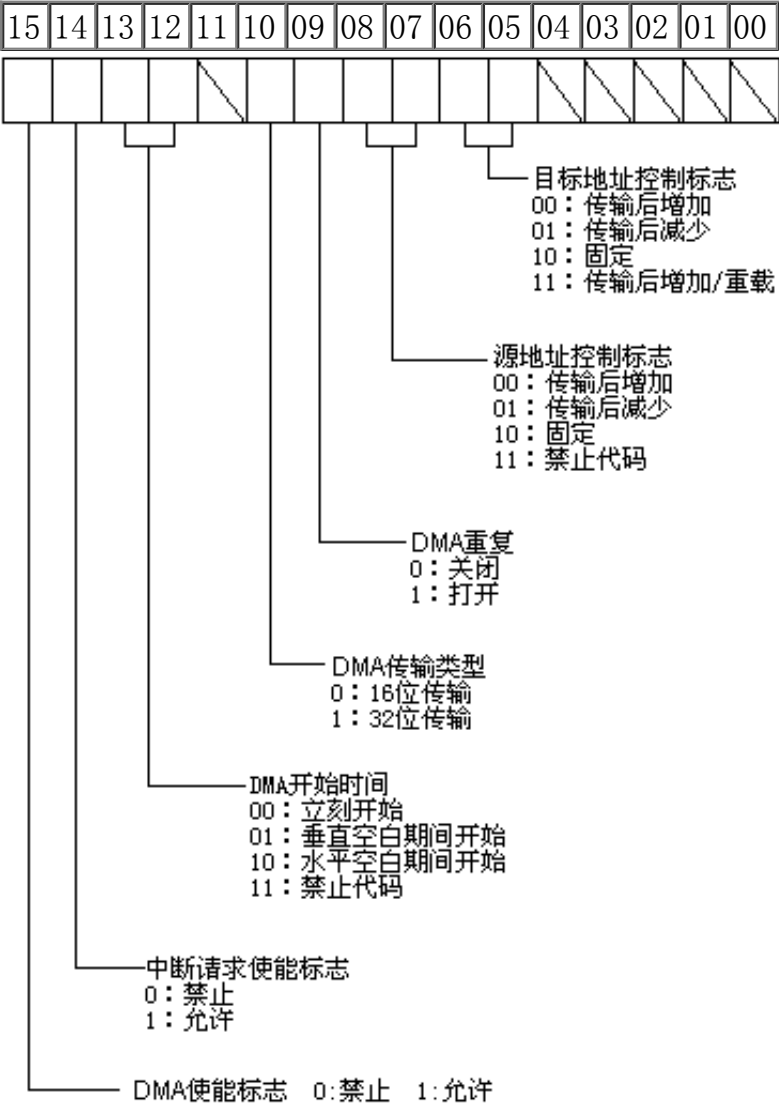
[illegible]

字计数：  
使用14位指定DMA0传输的字节数。数字范围是0000h—3FFFh。这样在16位数据传输模式，最多可以传输32,766字节；在32位数据传输模式，最多可以传输65,532字节。

地址：0B8h； 寄存器：DMOCNT\_L； 属性：只写； 初始值：0000h



DMA控制：  
地址：0BAh； 寄存器：DMOCNT\_H； 属性：读/写； 初始值：0000h



说明：

DMOCNT_H[d15]	设置为0则禁止DMA；1允许DMA，传输后会被复位
DMOCNT_H[d14]	中断请求标志，当指定的DMA传输字数完成后发生一个中断请求，0时没有请求，1时发生请求
DMOCNT_H[d13-12]	DMA传输时刻，DMA传输时刻可以从下面的选项里选择：
	设置 DMA开始时刻
	00 立刻开始
	01 垂直空白开始时开始（大约4.993毫秒）
	10 水平空白开始时开始（大约16.212毫秒）如果是访问OAM，水平空白必须先从精灵处理硬件里空闲出来。
	11 禁止代码

DMOCNT_H[d10]	DMA传输类型，设置传输数据的位长度，0是16位（半字），1是32位（字）
DMOCNT_H[d9]	DMA重复，打开时（1），如果选择垂直空白或水平空白作为DMA开始时刻，则在每次空白期间DMA都重新开始。 关闭时（0），DMA在传输完指定的数据量之后停止。
DMOCNT_H[d8-7]	源地址控制标志，控制每次DMA传输的源地址，00是增加，01是减少，10是固定，11是禁止代码
DMOCNT_H[d6-5]	目标地址控制标志，00是增加，01是减少，10是固定，11是增加，但在全部传输完后，重载传输开始时的设置

### DMA1和2:

DMA通道1和2提供卡带总线/主单元内部内存和主单元内部内存/直接声道FIFO。直接声道FIFO的传输只能用DMA1和2来完成。

#### 源地址:

使用27位指定源地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址：0BCh/0C8h；寄存器：DM1SAD\_L/DM2SAD\_L；属性：只写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

地址：0BEh/0CAh；寄存器：DM1SAD\_H/DM2SAD\_H；属性：只写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

#### 目标地址:

使用27位指定目标地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址：0C0h/0CCh；寄存器：DM1DAD\_L/DM2DAD\_L；属性：只写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

地址：0C2h/0CCh；寄存器：DM1DAD\_H/DM2DAD\_H；属性：只写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

#### 字计数:

使用14位指定DMA0传输的字节数。数字范围是0000h—3FFFh。这样在16位数据传输模式，最多可以传输32,766字节；在32位数据传输模式，最多可以传输65,532字节。

地址：0C4h/0D0；寄存器：DM1CNT\_L/DM2CNT\_L；属性：只写；初始值：0000h

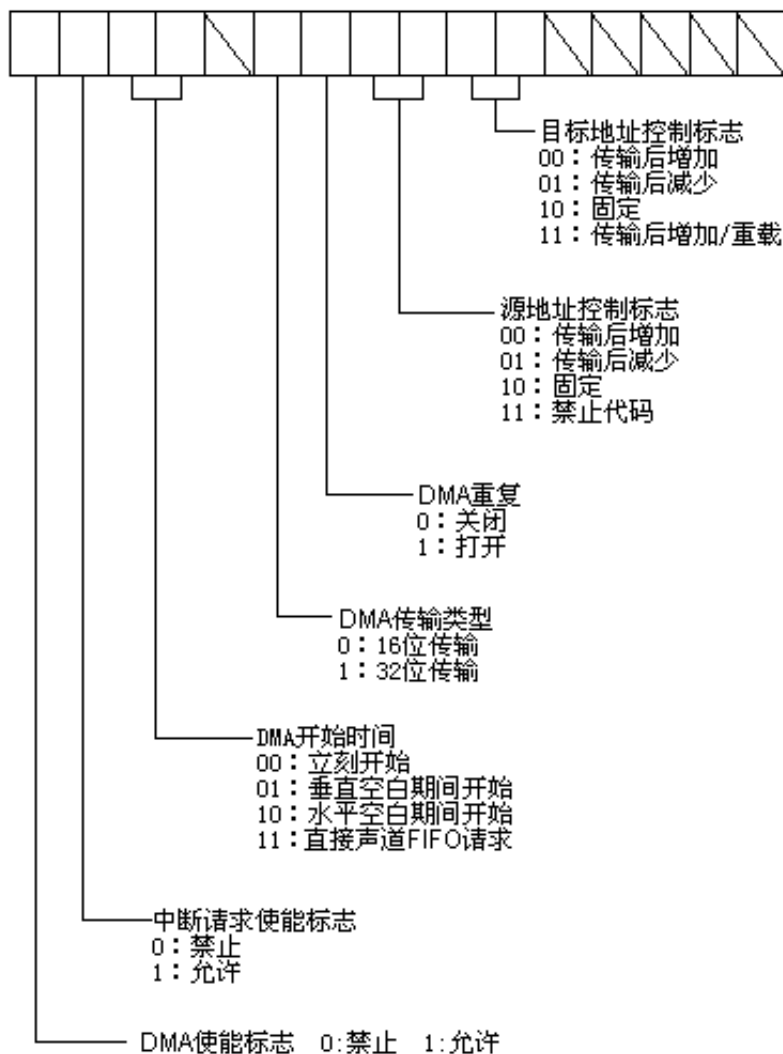
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

这个字计数寄存器在直接声道FIFO传输模式中不能用。从声道FIFO每接收到一个请求，就传输32位×4字。

#### DMA控制:

地址：0C6h/0D2h；寄存器：DM1CNT\_H/DM2CNT\_H；属性：读/写；初始值：0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



#### 说明:

DM(1, 2)CNT_H[d15]	设置为0则禁止DMA；1允许DMA，传输后会被复位
DM(1, 2)CNT_H[d14]	中断请求标志，当指定的DMA传输字数完成后发生一个中断请求，0时没有请求，1时发生请求
DM(1, 2)CNT_H[d13-12]	DMA传输时刻，DMA传输时刻可以从下面的选项里选择：
	设置 DMA开始时刻
	00 立刻开始
	01 垂直空白开始时开始（大约4.993毫秒）
	10 水平空白开始时开始（大约16.212毫秒）如果是访问OAM，水平空白应先从精灵处理硬件里空闲出来
	11 直接声道FIFO发生请求时开始。指定声道FIFO为目标地址，同时设置DMA重复功能[d9]为开。
DM(1, 2)CNT_H[d10]	DMA传输类型，设置传输数据的位长度，0是16位（半字），1是32位（字），直接声道FIFO传输模式是32位
DM(1, 2)CNT_H[d9]	DMA重复。 打开时（1），如果选择垂直空白或水平空白作为DMA开始时刻，则在每次空白期间DMA都重新开始。 关闭时（0），DMA在传输完指定的数据量之后停止。 在直接声道FIFO传输模式设置这位为1。
DM(1, 2)CNT_H[d8-7]	源地址控制标志，控制每次DMA传输的源地址，00是增加，01是减少，10是固定，11是禁止代码 选择卡带总线作为源地址时，要选择增加。

DM(1, 2)CNT_H[d6—5]	目标地址控制标志，00是增加，01是减少，10是固定，11是增加，但在全部传输完后，重载传输开始时的设置。但在直接声道FIFO传输模式，目标地址是固定的，和设置无关。
---------------------	---

### DMA3:

DMA3提供卡带总线/主单元内部内存和不同主单元内部内存之间的访问。

源地址:

使用27位指定源地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址: 0D4h; 寄存器: DM3SAD\_L; 属性: 只写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

地址: 0D6h; 寄存器: DM3SAD\_H; 属性: 只写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

目标地址:

使用27位指定目标地址。可以指定00000000h—07FFFFFFh（主单元内部内存）区域。

地址: 0D8h; 寄存器: DM3DAD\_L; 属性: 只写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

地址: 0DAh; 寄存器: DM3DAD\_H; 属性: 只写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

字计数:

使用14位指定DMA0传输的字节数。数字范围是0000h—3FFFh。这样在16位数据传输模式，最多可以传输32,766字节；在32位数据传输模式，最多可以传输65,532字节。

地址: 0DCh; 寄存器: DMOCNT\_L; 属性: 只写; 初始值: 0000h

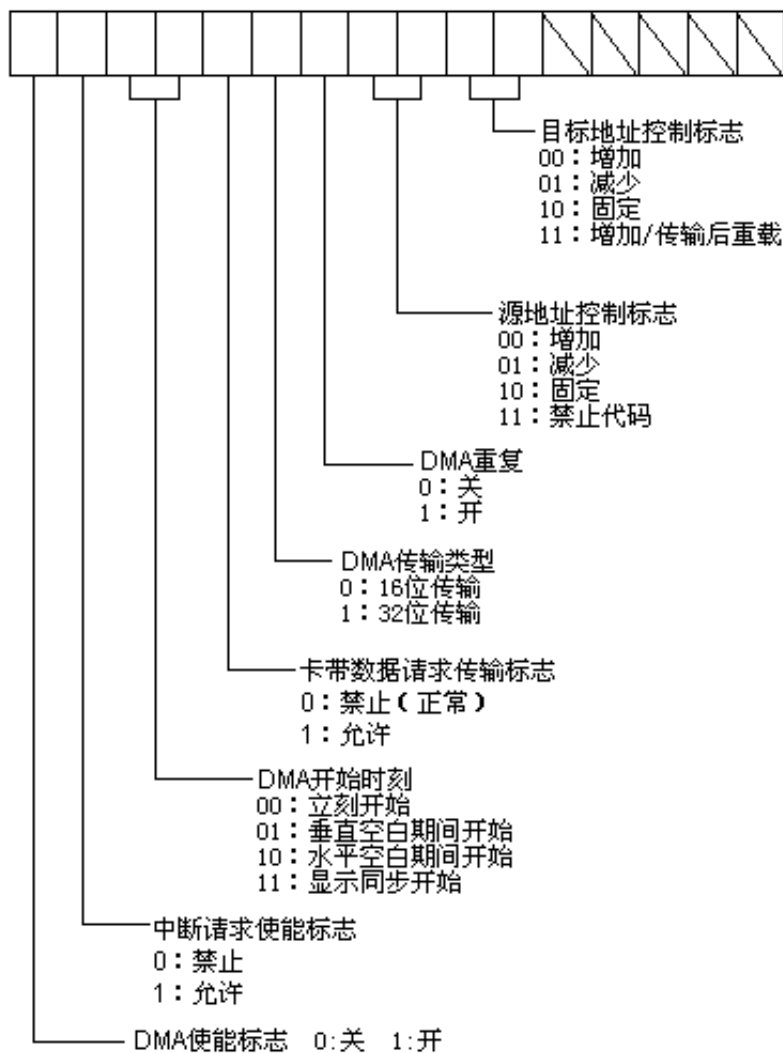
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

DMA控制:

地址: 0DEh; 寄存器: DMOCNT\_H; 属性: 读/写; 初始值: 0000h

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----





#### 说明:

DM3CNT_H[d15]	设置为0则禁止DMA; 1允许DMA, 传输后会被复位										
DM3CNT_H[d14]	中断请求标志, 当指定的DMA传输字数完成后发生一个中断请求, 0时没有请求, 1时发生请求										
DM3CNT_H[d13-12]	DMA传输时刻, DMA传输时刻可以从下面的选项里选择: <table border="1"> <thead> <tr> <th>设置</th><th>DMA开始时刻</th></tr> </thead> <tbody> <tr> <td>00</td><td>立刻开始</td></tr> <tr> <td>01</td><td>垂直空白开始时开始 (大约4.993毫秒)</td></tr> <tr> <td>10</td><td>水平空白开始时开始 (大约16.212毫秒) 如果是访问OAM, 水平空白应先从精灵处理硬件里空闲出来</td></tr> <tr> <td>11</td><td>显示同步开始, 显示期间和水平扫描线描绘开始同步。</td></tr> </tbody> </table>	设置	DMA开始时刻	00	立刻开始	01	垂直空白开始时开始 (大约4.993毫秒)	10	水平空白开始时开始 (大约16.212毫秒) 如果是访问OAM, 水平空白应先从精灵处理硬件里空闲出来	11	显示同步开始, 显示期间和水平扫描线描绘开始同步。
设置	DMA开始时刻										
00	立刻开始										
01	垂直空白开始时开始 (大约4.993毫秒)										
10	水平空白开始时开始 (大约16.212毫秒) 如果是访问OAM, 水平空白应先从精灵处理硬件里空闲出来										
11	显示同步开始, 显示期间和水平扫描线描绘开始同步。										
DM3CNT_H[d11]	卡带数据请求传输标志, 正常情况下应设为0, 设为1时, DMA响应卡带的请求执行传输。 注意: 使用这种模式需要卡带支持, 并且不能和卡带中断同时使用。										
DM3CNT_H[d10]	DMA传输类型, 设置传输数据的位长度, 0是16位(半字), 1是32位(字)										
DM3CNT_H[d9]	DMA重复。打开时(1), 如果选择垂直空白或水平空白作为DMA开始时刻, 则在每次空白期间DMA都重新开始。 关闭时(0), DMA在传输完指定的数据量之后停止。										
DM3CNT_H[d8-7]	源地址控制标志, 控制每次DMA传输的源地址, 00是增加, 01是减少, 10是固定, 11是禁止代码 选择卡带总线作为源地址时, 要选择增加。										

DM3CNT_H[d6-5]	目标地址控制标志，00是增加，01是减少，10是固定，11是增加，但在全部传输完后，重载传输开始时的设置。
----------------	---

# 十三、通讯功能

GBA提供下面5个功能：

1	8位/32位普通通讯功能：使用普通通讯的电缆，也可以不用电缆而是附件来通讯（当使用电缆通讯时，使用多玩家通讯）。因为电压不同，它不可能和GBC通讯。8位通讯处理8位数据，32位处理32位数据。
2	16位多玩家通讯功能：使用UART系统的多路同时通讯最多可以进行4台GBA的通讯，需要特殊的多玩家通讯电缆。
3	UART通讯功能：可以通过UART系统进行高速通讯。
4	通用通讯功能：通过直接控制通讯引脚可以用任何协议进行通讯。
5	JOY总线通讯功能：可以使用任天堂的标志JOY总线进行通讯。

选择通讯功能：  
全部的通讯功能使用一个外部的扩展6针接头。通讯功能通过通讯控制寄存器R的通讯功能设置标志（2位）和串行通讯控制寄存器SCCNT\_L的通讯模式设置标志（2位）来切换。如下：

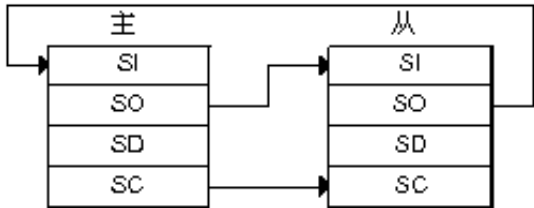
通讯功能	R		SCCNT_L	
	d15	d14	d13	d12
通用目的	1	0	*	*
JOY总线	1	1	*	*
8位串行	0	*	0	0
32位串行	0	*	0	1
16位串行	0	*	1	0
UART	0	*	1	1

当改变通讯模式时，先只改变通讯模式设置标志。不要在改变模式的同时开始通讯，可能会导致故障。

## 8位32位普通串行通讯：

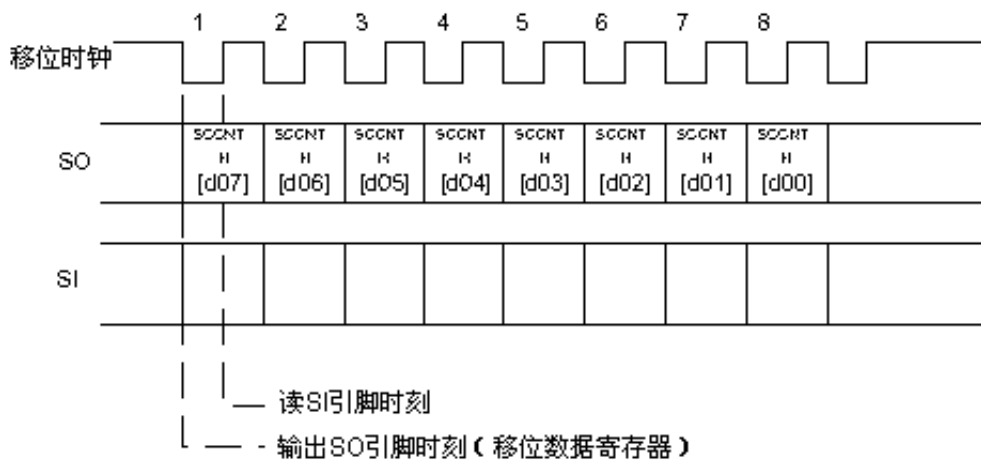
串行通讯是同时发送和接收的。在数据寄存器里设置数据，开始串行通讯，在通讯结束时，接受的数据就在数据寄存器里。

在普通传唤通讯时连接：



主机（内部时钟模式）会从SC引脚输出移位时钟，SD引脚变成上拉输入。在副机模式（外部时钟模式），SC引脚变成上拉输入，SD引脚转为低输出。为了从最高有效位开始，设置的数据会被移位时钟的下降沿左移，并从SO引脚输出。从SI引脚输入的数据会在移位时钟的上升沿输入到最低有效位。

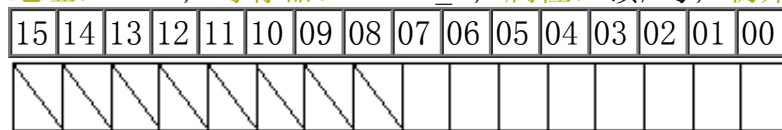
SI0时刻表：



### 8位普通串行通讯数据寄存器

8位传输模式使用SCCNT\_H作为数据寄存器，高8位变成禁止的。

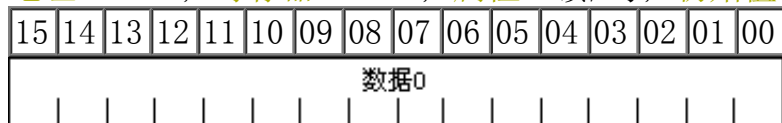
地址：12Ah；寄存器：SCCNT\_H；属性：读/写；初始值：0000h



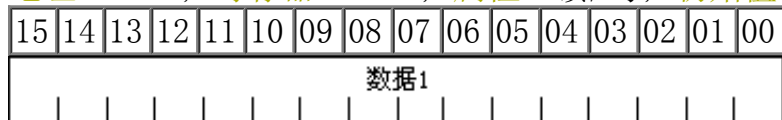
### 32位普通串行通讯数据寄存器

32位通讯模式使用[120h:SCD0]和[122h:SCD1]作为数据寄存器（这些数据寄存器也用在16位多玩家通讯里）。最高有效位在SCD1寄存器的d15，最低有效位在SCD0的d0。

地址：120h；寄存器：SCD0；属性：读/写；初始值：0000h



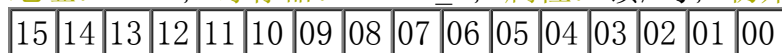
地址：122h；寄存器：SCD1；属性：读/写；初始值：0000h

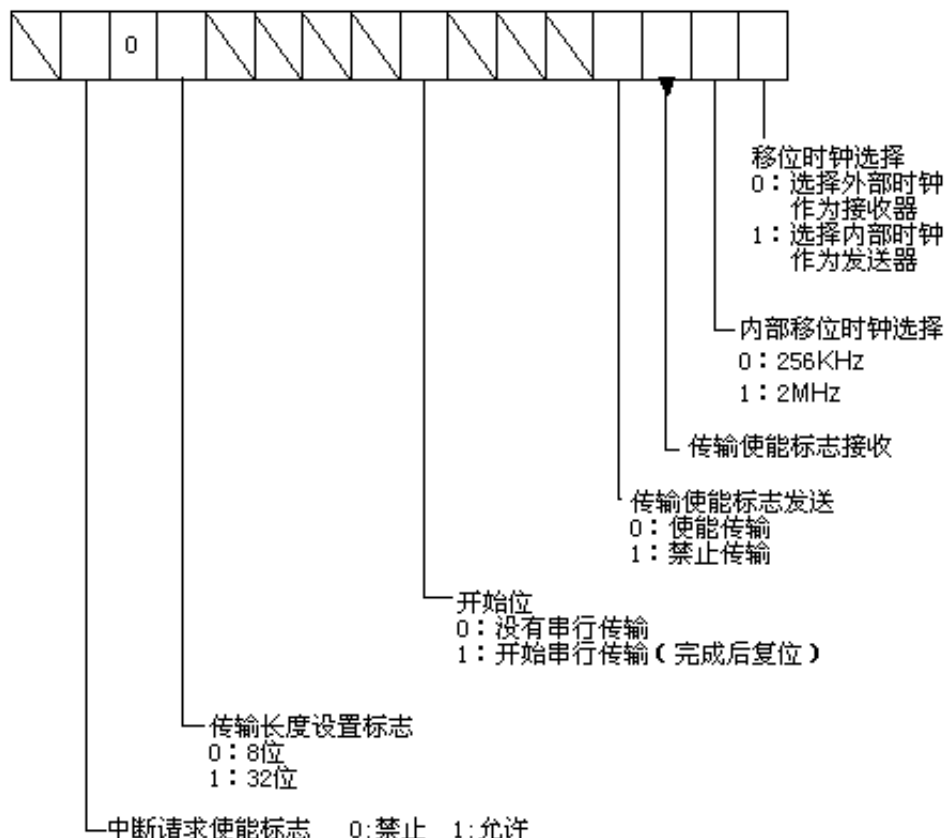


### 控制寄存器

当寄存器R(d15) = (0)，设置SCCNT\_L(d13, d12) = (0, 0)就是8位普通串行通讯模式；设置SCCNT\_L(d13, d12) = (0, 1)就是32位模式

地址：128h；寄存器：SCCNT\_L；属性：读/写；初始值：0000h





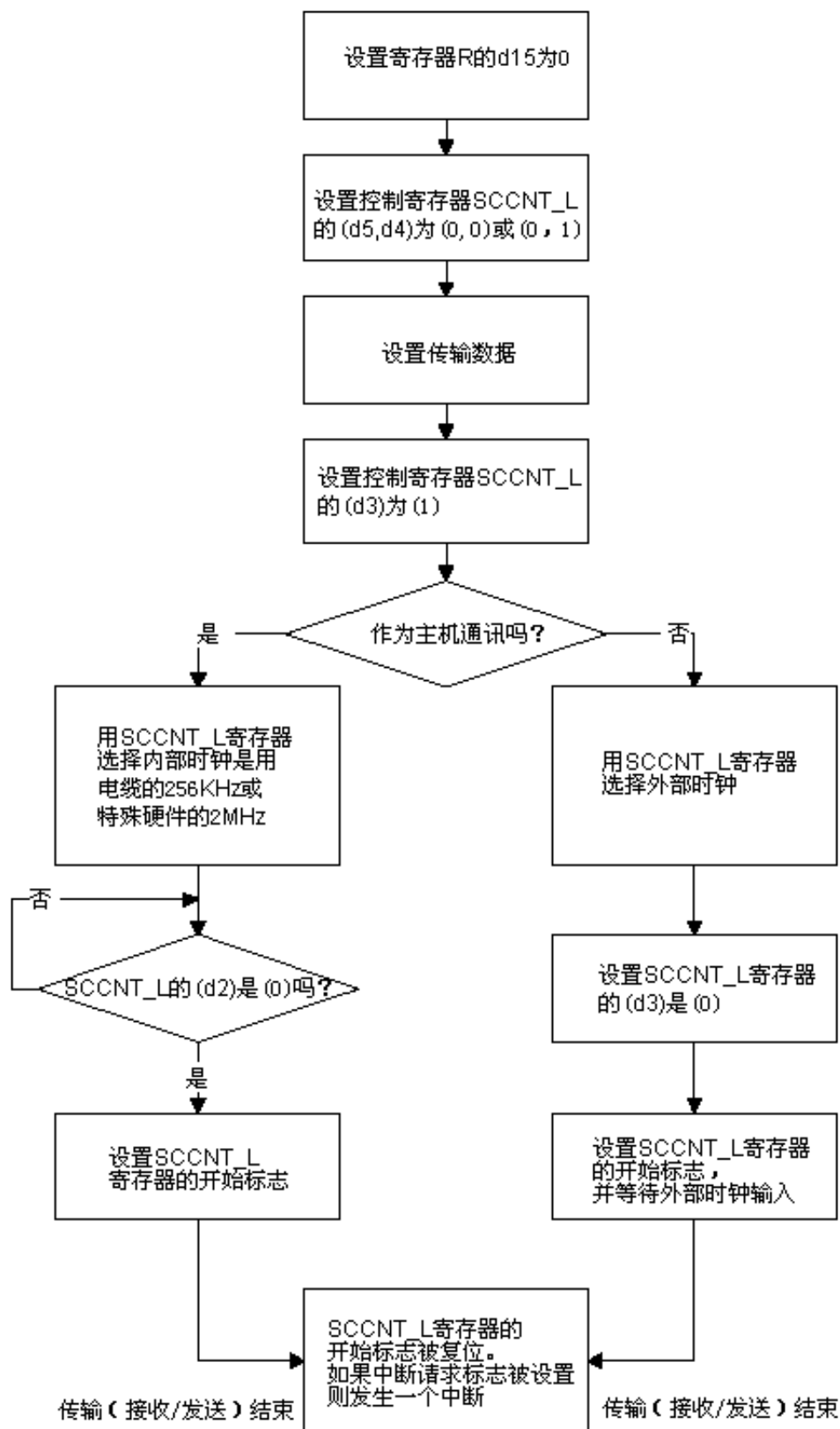
#### 说明：

SCCNT_L[d14]	中断请求使能标志，0没有请求，1在传输后立刻发生请求
SCCNT_L[d12]	传输长度设置标志，0是8位传输，1是32位传输
SCCNT_L[d7]	开始位，1时开始串行传输。该位在传输完成后自动复位
SCCNT_L[d3]	传输使能表示发送，0使能传输，1禁止。在传输开始时这个标志从S0引脚输出；传输时，串行数据也从S0引脚输出
SCCNT_L[d2]	传输使能标志接收，在通讯开始前可以去读SI引脚的状态（从另一方硬件传来的传输使能标志）。通讯开始后变成无效的（通讯中接收的数据位被反射回去）
SCCNT_L[d1]	内部移位时钟选择，0是选择256KHz移位时钟，1是选择2MHz移位时钟
SCCNT_L[d0]	移位时钟选择，0时使用外部时钟移位（副机），外部时钟从另一个硬件单元经SC引脚输入，SD引脚转到低输出。 1时，使用内部移位时钟（主机），内部时钟从SC引脚输出，SD引脚会处于上拉输入状态

#### 注意：

移位时钟应该在SCCNT\_L寄存器的开始位被设置之前选择好。额外的移位操作会导致串行传输在选择移位时钟之前或同时开始。8位通讯模式和GB/GBC的模式兼容，但是通讯引脚电压不同，因此GBA和GB/GBC通讯是不可能的。在普通通讯中是禁止使用电缆的，为了在GBA之间通讯，使用多玩家通讯功能，在后面再讨论。

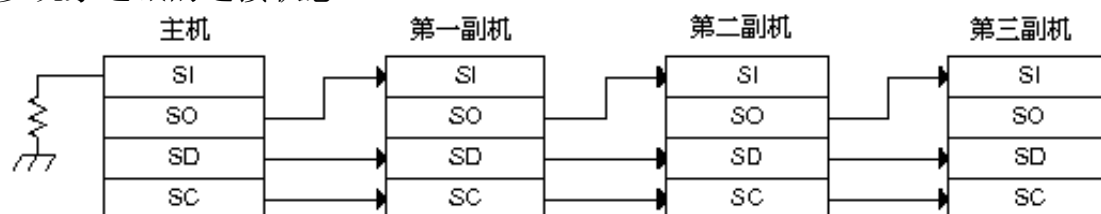
#### 普通串行通讯流程（示例）：



### 16位多玩家通讯:

GBA可以通过特殊的电缆让最多4个单元进行多玩家通讯。根据连接状态，依次设置1个单元作为主机向副机传输数据。

多玩家通讯的连接状态:



在多玩家通讯模式中，SC和SD作为上拉输入引脚。在复位之后或在其他通讯模式，从SD引脚输出的是低电平。一旦SD引脚变为高，你就可以说全部连接终端进入了多玩家通讯模式。SI引脚是上拉输入，但是因为多玩家通讯电缆，它变成下拉。因此，当全部终端处于多玩家模式，输入到SI引脚为低电平的终端变成主机，输入到SI引脚为高电平的变成副机。如果设置了SCCNT\_L寄存器的开始位，主机的数据寄存器SCD0、SCD1、SCD2和SCD3被初始化为FFFFh。另外，“同步信号”（低电平）从SC引脚输出，同时，SD引脚输出“开始位”（低电平）。然后，数据从SCCNT\_H寄存器输出，再然后是“停止位”（高电平）。这些完了之后，主机使SD引脚变成上拉输入，S0引脚输出低电平。每个副机检测从主机输出的“同步信号”并初始化全部的数据寄存器（SCD0、SCD1、SCD2和SCD3）为FFFFh。从主机输出的数据储存在主机和每个副机的SCD0寄存器。如果紧随在主机后面的副机的SI引脚输入低电平，一个“开始位”（低电平）即从SD引脚输出。然后SCCNT\_H寄存器的数据输出，最后是“停止位”（高电平）。这些之后SD引脚转为上拉输入，从S0引脚输出低电平。这时，第一副机输出的数据就存在主机和每个副机的SCD1寄存器里。这样每个副机都被发送了，全部的通讯被执行。在下面的情况下，主机产生一个“同步信号”（在输出5个周期高电平间隔的源振动之后上拉输入），传输结束。

- 在主机输出它自己的“停止位”之后，在相当一段时间里没有输入“开始位”；
- 在接收到第一和第二副机的“停止位”后，在相当一段时间里没有输入“开始位”；
- 从第三副机接收到一个“停止位”。

如果传输结束，接收的数据储存在各个数据寄存器里(SCD0、SCD1、SCD2和SCD3)。如果有一个终端没有连接，则储存初始值FFFFh。

多玩家通讯时刻表：





地址：120h； 寄存器：SCD0； 属性：读/写； 初始值： 0000h



地址：122h； 寄存器：SCD1； 属性：读/写； 初始值： 0000h



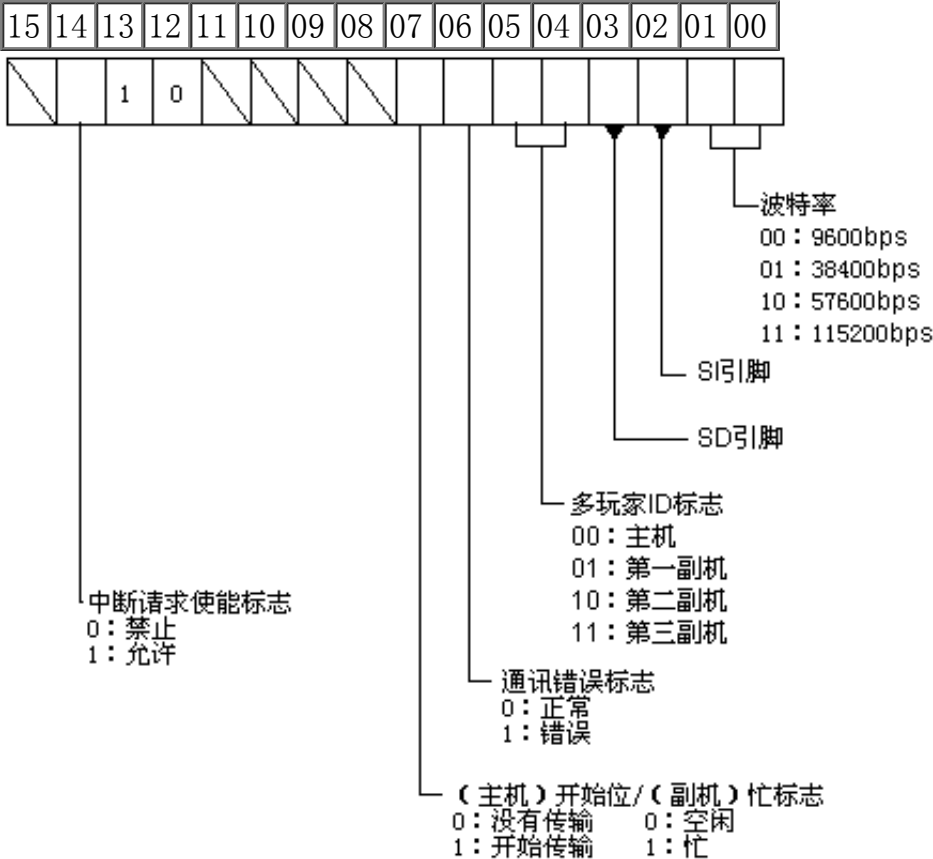
地址：124h； 寄存器：SCD2； 属性：读/写； 初始值： 0000h



地址：126h； 寄存器：SCD3； 属性：读/写； 初始值： 0000h



控制寄存器：  
如果在R(d15)=(0)时， 设置SCCNT\_L(d13, d12)=(1, 0)， 就会转到16位多玩家通讯模式  
地址：128h； 寄存器：SCCNT\_L； 属性：读/写； 初始值： 0000h

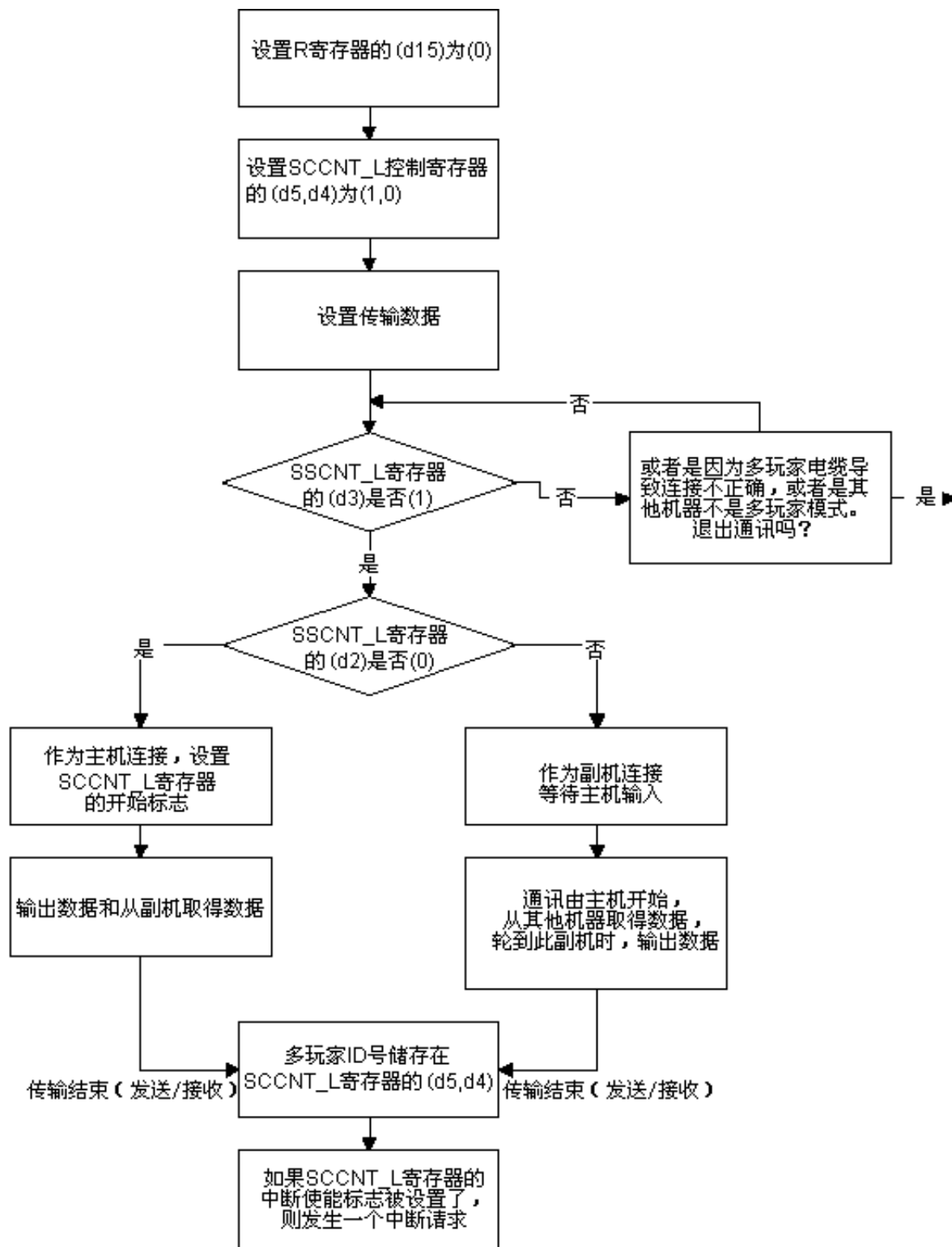


说明：

SCCNT_L[d14]	中断请求使能标志：0时没有发生中断请求，1时在完成多玩家通讯后发生一个中断请求
SCCNT_L[d7]	主机(d00是1)：开始位，1时开始串行传输。该位在传输完成后自动复位 副机(d00是0)：在输入传输开始位（低电平源振动周期×3，约180纳秒）时被设置，传输结束后复位

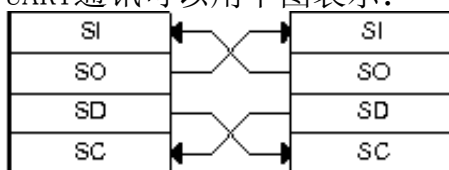
SCCNT_L[d6]	通讯错误标志：确定通讯状态，如果此位是0，没有错误；如果是1，表示有错误发生。 在下面这些情况，错误标志会自动被设置： <ul style="list-style-type: none"><li>在（主机输出的）“同步信号”输入时，SI引脚没有变成低电平</li><li>接收数据的停止位不是高电平（帧错误）</li></ul> 然而，即时有错误，通讯也会继续，无效数据储存在SCD0—SCD3。											
SCCNT_L[d5-4]	多玩家ID标志：多玩家通讯结束时，会储存一个ID代码用来指定每一台机器连接的顺序											
SCCNT_L[d3]	SD引脚：可以读取SD引脚的状态，如果所以终端进入多玩家通讯模式，它会变成高电平											
SCCNT_L[d2]	SI引脚：可以读取SI引脚的状态，当全部终端进入多玩家通讯模式，SI引脚低输入的是主机，高输入是副机											
SCCNT_L[d1-0]	波特率：设置通讯波特率 <table><tr><td>设置</td><td>波特率</td></tr><tr><td>00</td><td>9600bps</td></tr><tr><td>01</td><td>38400bps</td></tr><tr><td>10</td><td>57600bps</td></tr><tr><td>11</td><td>115200bps</td></tr></table>		设置	波特率	00	9600bps	01	38400bps	10	57600bps	11	115200bps
设置	波特率											
00	9600bps											
01	38400bps											
10	57600bps											
11	115200bps											

多玩家通讯流程（示例）：



## UART通讯功能：

UART通讯可以用下图表示：

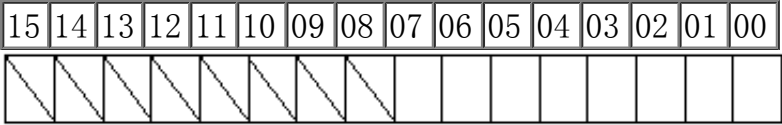


在UART通讯模式，SD引脚输出一个高电平。如果数据接收寄存器（或者接收FIFO）满了，SD引脚输出一个高电平；没有满的时候，如果接收使能标志被设置，SD引脚就输出低电平。复位后输出高电平。另一部机器的SD引脚输出输入到SC引脚。如果数据写入了数据发送寄存器，则从SO引脚输出一个“开始位”（1位）之后接着输出数据。但是，如果控制寄存器的CTS标志被设置了，数据可以只在SC引脚有低输入时才发送。停止位固定是1位。当发送或接收时，FIFO有4个字节。通过使用SCCNT\_L控制寄存器，可以选择是否使用FIFO。如果不使用FIFO，写SCCNT\_H

寄存器时，数据是写到数据发送寄存器；读取完成时，数据是从接收寄存器读取的（只有低8位有效）。如果使用FIFO，写SCCNT\_H寄存器时，数据是写到数据发送FIFO；读取完成时，数据是从接收FIFO读取的（只有低8位有效）。

数据寄存器：

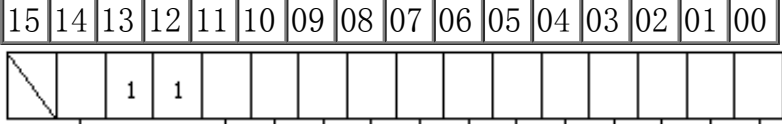
地址：12Ah； 寄存器：SCCNT\_H； 属性：读/写； 初始值：0000h



控制寄存器：

当寄存器R(d15) = (0)，设置SCCNT\_L(d13, d12)=(1, 1)就是UART通讯模式

地址：128h； 寄存器：SCCNT\_L； 属性：读/写； 初始值：0000h



波特率  
00：9600bps  
01：38400bps  
10：57600bps  
11：115200bps

CTS标志  
0：始终可以发送  
1：在SC引脚低输入时发送

校验控制  
0：偶校验  
1：奇校验

发送数据标志  
0：非满  
1：满

接收数据标志  
0：非空  
1：空

错误标志  
0：没有错误  
1：错误

数据长度  
0：7位  
1：8位

FIFO使能标志  
0：禁止  
1：允许

校验使能标志  
0：禁止  
1：允许

发送使能标志  
0：禁止  
1：允许

接收使能标志  
0：禁止  
1：允许

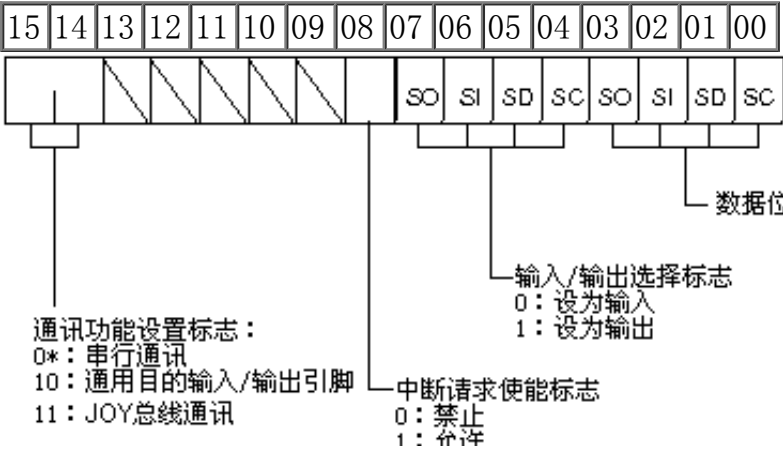
0：禁止  
1：允许  
中断请求使能标志

说明：

SCCNT_L[d14]	中断请求标志：0时不发生中断，1时如果通讯发生错误或传输结束则发生一个中断请求										
SCCNT_L[d11]	接收使能标志：设置此标志，当数据接收寄存器（或接收FIFO）不满时，SD引脚输出低电平，复位时输出高电平。 要从UART通讯模式转到不同的通讯模式前，必须首先设置此标志为0（禁止）。										
SCCNT_L[d10]	发送使能标志：要从UART通讯模式转到不同的通讯模式前，必须首先设置此标志为0（禁止）。										
SCCNT_L[d9]	校验使能标志：控制是否进行奇偶校验。										
SCCNT_L[d8]	FIFO使能标志：控制8位×4层发送和接收FIFO是否工作。要使用FIFO先要进入无FIFO的UART模式。 FIFO序列器的初始化通过在UART模式中禁止FIFO来实现。										
SCCNT_L[d7]	数据长度：选择数据是8位还是7位。										
SCCNT_L[d6]	错误标志：参考这个错误标志，可以确定通讯错误的状态。0表示没有错误，1有错误，读取SCCNT_L，标志被复位。 另外，当错误发生时，从接收移位寄存器的数据不写道数据接收寄存器里。每种错误的条件如下所示： <table border="1"> <tr> <td>帧错误</td><td>接收数据的停止位不是0</td></tr> <tr> <td>奇偶校验错误</td><td>使能奇偶校验时，在校验中发生错误</td></tr> <tr> <td>超载错误</td><td>接收数据非空时，紧随的接收结束（检测到停止位）</td></tr> </table>	帧错误	接收数据的停止位不是0	奇偶校验错误	使能奇偶校验时，在校验中发生错误	超载错误	接收数据非空时，紧随的接收结束（检测到停止位）				
帧错误	接收数据的停止位不是0										
奇偶校验错误	使能奇偶校验时，在校验中发生错误										
超载错误	接收数据非空时，紧随的接收结束（检测到停止位）										
SCCNT_L[d5]	接收数据标志：0表示还有数据，1表示空。										
SCCNT_L[d4]	发送数据标志：0表示未满，一个发送操作结束时复位；1表示满，写到数据发送寄存器SCCNT_H低8位时被设置。										
SCCNT_L[d3]	奇偶校验控制：切换奇校验和偶校验。										
SCCNT_L[d2]	CTS标志：另一台机器的SD引脚输入到SC引脚。0时总允许发送，1时只在SC引脚有低输入时发送。										
SCCNT_L[d1-0]	波特率：设置通讯波特率 <table border="1"> <tr> <th>设置</th><th>波特率</th></tr> <tr> <td>00</td><td>9600bps</td></tr> <tr> <td>01</td><td>38400bps</td></tr> <tr> <td>10</td><td>57600bps</td></tr> <tr> <td>11</td><td>115200bps</td></tr> </table>	设置	波特率	00	9600bps	01	38400bps	10	57600bps	11	115200bps
设置	波特率										
00	9600bps										
01	38400bps										
10	57600bps										
11	115200bps										

**通用目的通讯：**  
 设置(d13, d12)=(1, 0)初始化通用目的通讯模式。在这种模式，全部的引脚SI、S0、SC和SD变成上拉，作为通用目的输入/输出引脚工作。每个通讯引脚SI、S0、SC和SD可以直接控制。

**地址：**134h； **寄存器：**R； **属性：**读/写； **初始值：**0000h



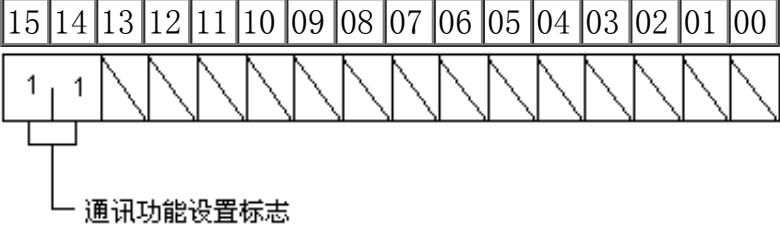
说明：

R[d15-d14]	通讯功能设置标志：00或者01，作为为串行通讯（8位/16位串行通讯，多玩家通讯，UART通讯功能）引脚；10，作为通用目的输入/输出引脚；11，用作JOY总线通讯引脚。
R[d8]	中断请求使能标志：当通讯功能设置标志设置成通用目的输出/输入(R[d15, d14]=[1, 0])时，此位设为1，则在SI引脚（边检测）的下降沿发生一个中断请求；0时没有中断请求发生。
R[d7-4]	输入/输出选择标志：当通讯功能设置标志设置成通用目的输出/输入(R[d15, d14]=[1, 0])时，0使得相应的引脚作为输入，1使得相应的引脚作为输出。
R[d3-0]	数据位：当响应的引脚被设置为输入，可以确定引脚的状态（高/低）。如果作为输出，则输出该位的状态。

JOY总线通讯：

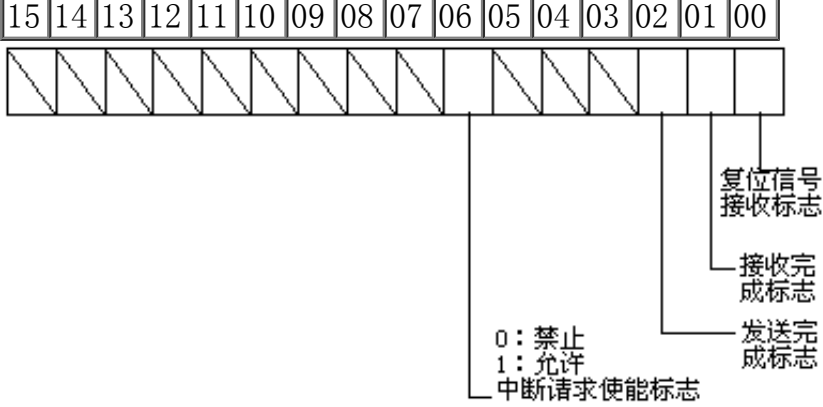
通过设置R寄存器的通讯功能设置标志为11，选择JOY总线通讯模式。在JOY总线通讯模式，SI引脚作为输入，SO引脚作为输出，SD和SC引脚转为低输出。

地址：134h； 寄存器：R； 属性：读/写； 初始值：0000h



JOY总线通讯控制：

地址：140h； 寄存器：HS\_CTRL； 属性：读/写； 初始值：0000h



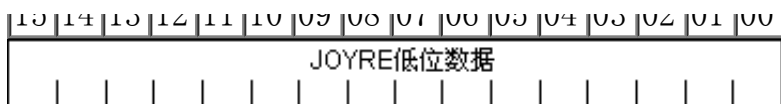
说明：

HS_CTRL[d5]	中断请求使能标志：0不发生中断请求，1在设备接收到一个设备复位信号后发生中断请求
HS_CTRL[d2]	发送完成标志：在发送操作完成时设置，当这个位被设置时，写一个1就会使它复位。
HS_CTRL[d1]	接收完成标志：在接收操作完成时设置，当这个位被设置时，写一个1就会使它复位。
HS_CTRL[d0]	设备复位信号接收标志，接收到一个设备复位命令时设置，当这个位被设置时，写一个1就会使它复位。

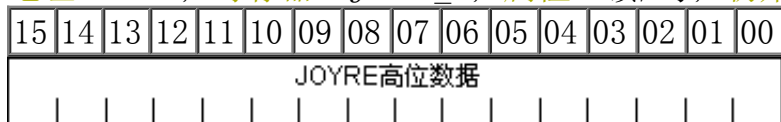
数据接收寄存器：

地址：150h； 寄存器：JOYRE\_L； 属性：读/写； 初始值：0000h



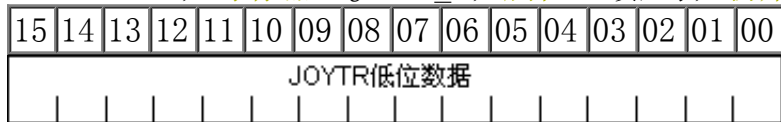


地址: 152h; 寄存器: JOYRE H; 属性: 读/写; 初始值: 0000h

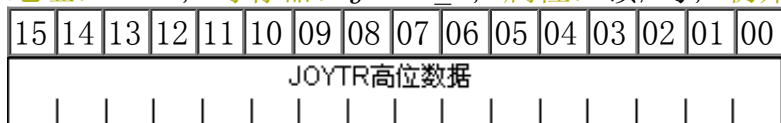


数据发送寄存器:

地址: 154h; 寄存器: JOYTR L; 属性: 读/写; 初始值: 0000h



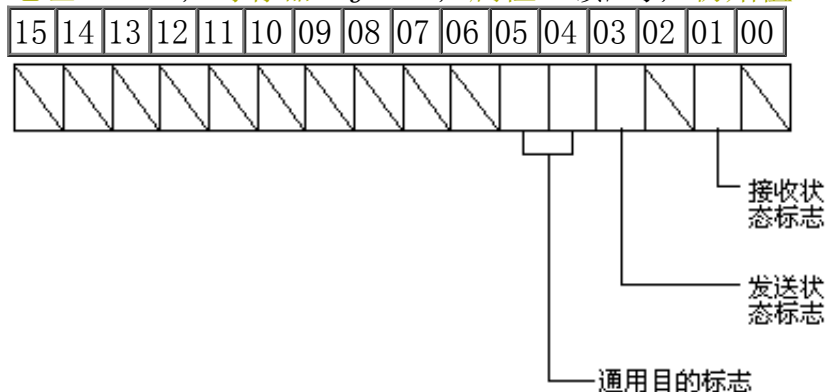
地址: 156h; 寄存器: JOYTR H; 属性: 读/写; 初始值: 0000h



接收状态寄存器:

接收状态寄存器 ISTAT 的低 8 位返回通讯状态。

地址: 158h; 寄存器: JSTAT; 属性: 读/写; 初始值: 0000h



说明:

JSTAT[d5-4]	这个标志没有定义，用户可以任意设置这个标志的定义
JSTAT[d3]	发送状态标志：当接收到一个GBA写数据信号，这个位被设置。如果用JOYRE寄存器读取完一个字，它被复位。
JSTAT[d1]	接收状态标志：当用JOYTR寄存器写完一个字，这个位被设置。如果接收到一个GBA读数据信号，它被复位。

### JOY总线通讯操作:

GBA的JOY总线通讯可以识别从主机（N64等）发出的4个命令：[设备复位]、[类型/状态数据请求]、[GBA数据写]和[GBA数据读]。GBA操作基于接收到的特殊信号。JOY总线通讯传输的位数据是以字节为单位的，并从最高有效位开始。

- 接收[设备复位]命令 (FFh)

HS\_CTRL寄存器的设备复位信号接收标志被设置。如果同一个寄存器的中断请求使能标志也设置了, 就发出一个JOY总线中断请求。

方向	顺序	d7	d6	d5	d4	d3	d2	d1	d0	注释
接收	1	1	1	1	1	1	1	1	1	命令255 (FFh)
发送	1	0	0	0	0	0	0	0	0	类型号码 0400h
	2	0	0	0	0	0	1	0	0	
	3	JSTAT寄存器低8位								通讯状态

- 接收[类型/状态数据请求]命令（00h）

返回2个字节类型号码（0004h）和一个字节通讯状态。

方向	顺序	d7	d6	d5	d4	d3	d2	d1	d0	注释
接收	1	0	0	0	0	0	0	0	0	命令0 (00h)
发送	1	0	0	0	0	0	0	0	0	类型号码 0400h
	2	0	0	0	0	0	1	0	0	
	3	JSTAT寄存器低8位								通讯状态

- 接收[GBA数据写]命令（15h）

接收紧随这个命令的4字节数据，储存在JOYRE寄存器里。接收完成后返回一个1字节的通讯状态，HS\_CTRL寄存器的接收完成标志被设置。同时，如果同一寄存器的中断请求标志设置了，发生一个JOY总线中断请求。

方向	顺序	d7	d6	d5	d4	d3	d2	d1	d0	注释
接收	1	0	0	0	1	0	1	0	1	命令21 (15h)
接收	2	数据接收寄存器JOYRE_L低8位								接收数据
	3	数据接收寄存器JOYRE_L高8位								
	4	数据接收寄存器JOYRE_H低8位								
	5	数据接收寄存器JOYRE_H高8位								
发送	6	JSTAT寄存器低8位								通讯状态

- 接收[GBA数据读]命令（14h）

储存在JOYTR寄存器里的4个字节数据和一个字节通讯状态被发送出去，并设置HS\_CTRL寄存器的发送完成标志。同时，如果同一寄存器的中断请求标志设置了，发生一个JOY总线中断请求。

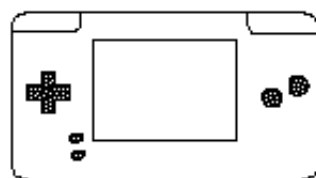
方向	顺序	d7	d6	d5	d4	d3	d2	d1	d0	注释
接收	1	0	0	0	1	0	1	0	0	命令20 (14h)
发送	2	数据发送寄存器JOYTR_L低8位								发送数据
	3	数据发送寄存器JOYTR_L高8位								
	4	数据发送寄存器JOYTR_H低8位								
	5	数据发送寄存器JOYTR_H高8位								
	6	JSTAT寄存器低8位								通讯状态

## 通讯电缆:

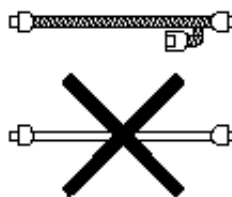
在GBA单元之间通讯时，根据使用卡带的类型而使用不同的电缆。



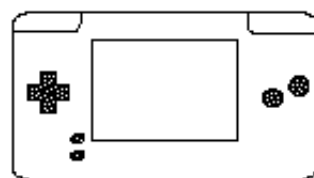
GBA卡带



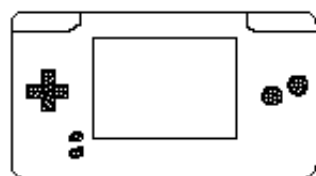
GBA的设计同前作不一样



GBA卡带

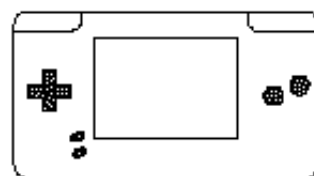
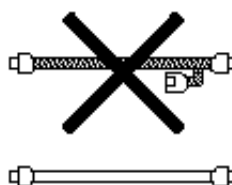
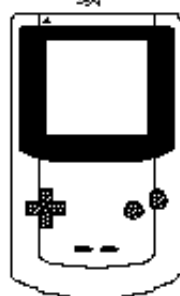


GBC卡带



或

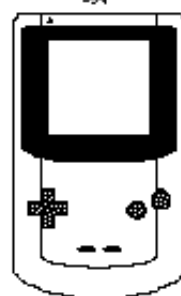
GBC卡带



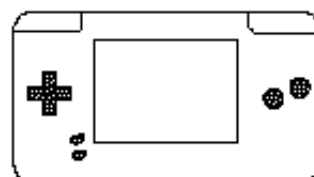
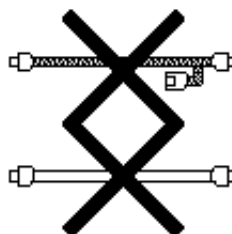
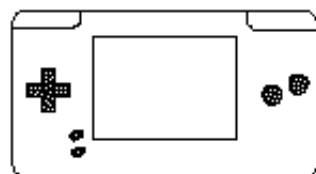
GBC卡带

或

GBC卡带



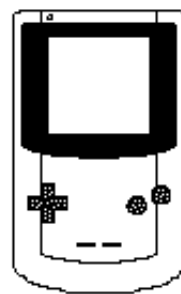
GBA卡带



GBC卡带

或

GBC卡带



GBA电缆



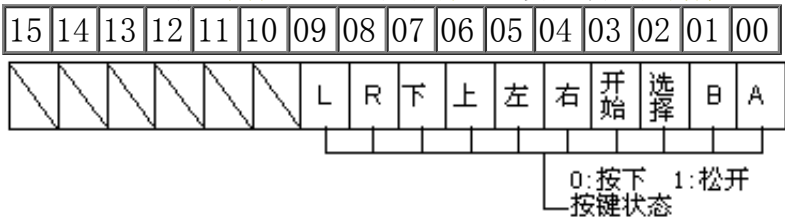
GB/GBC电缆

# 十四、按键输入

## 按键状态：

GBA允许用L和R键同START和SELECT键、方向键和A、B键输入。每个键的状态可以通过检查P1寄存器的分别对应的位取得。

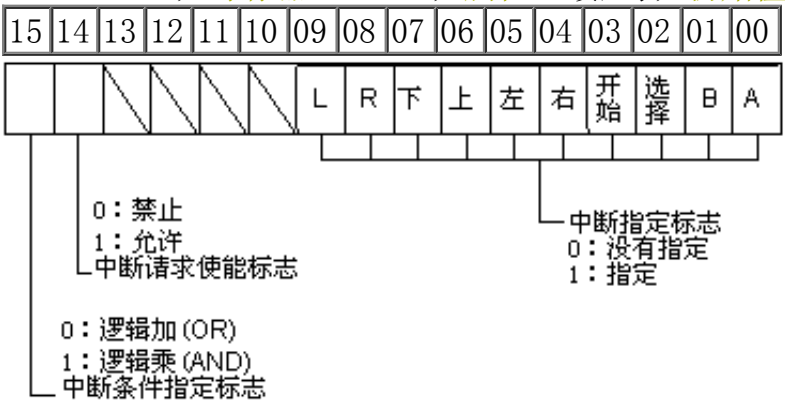
地址：130h； 寄存器：P1； 属性：读/写； 初始值：0000h



## 按键中断控制：

当执行一个按键输入中断时，这个寄存器可以指定中断的条件和联合目标按键。

地址：132h； 寄存器：P1CNT； 属性：读/写； 初始值：0000h



## 中断条件：

当中断使能请求标志是真时，指定发生中断的条件。在按键中断指定标志被指定时，可以为按钮选择下面这些条件：

### 逻辑加（OR）操作：

当有任何一个指定为中断的按键被输入时，触发中断请求条件。

### 逻辑乘（AND）操作：

当全部指定为中断的按键被输入时，触发中断请求条件。



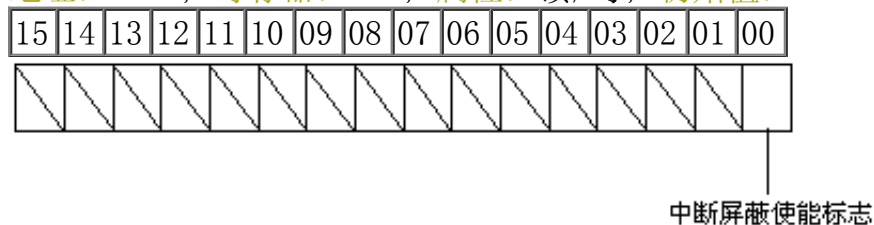
## 十五、中断控制

GBA可以使用14种可屏蔽硬件中断。从硬件部分接收到一个中断请求信号时，IF寄存器里相应的中断请求标志被设置。可以通过中断请求标志寄存器IE的方法屏蔽单独每个硬件部分的中断请求。

中断屏蔽使能寄存器:

全部的中断都可以被屏蔽。当这个标志是0，全部的中断被禁止。是1，则使能中断使能寄存器IE。

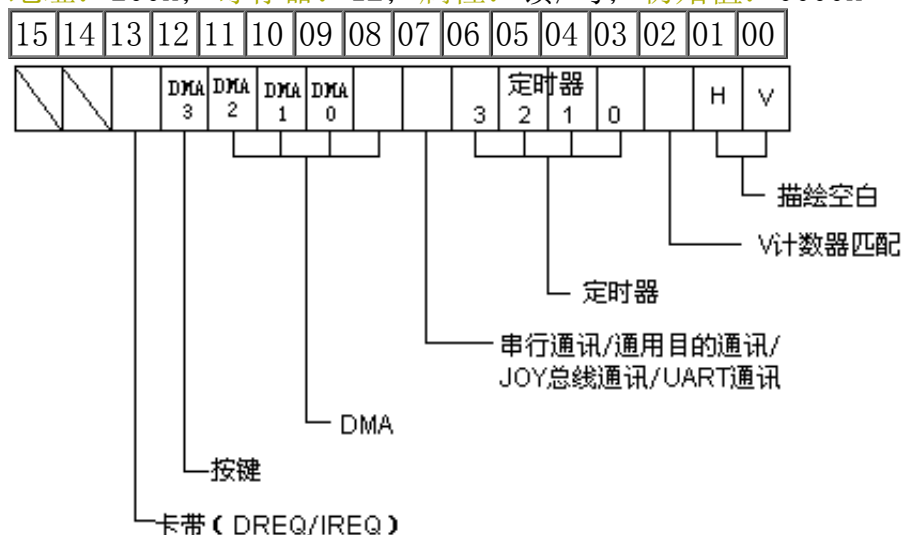
地址：208h；寄存器：IME；属性：读/写；初始值：0000h



中断使能寄存器:

使用中断使能寄存器，每个硬件中断可以单独被屏蔽。

地址: 200h; 寄存器: IE; 属性: 读/写; 初始值: 0000h

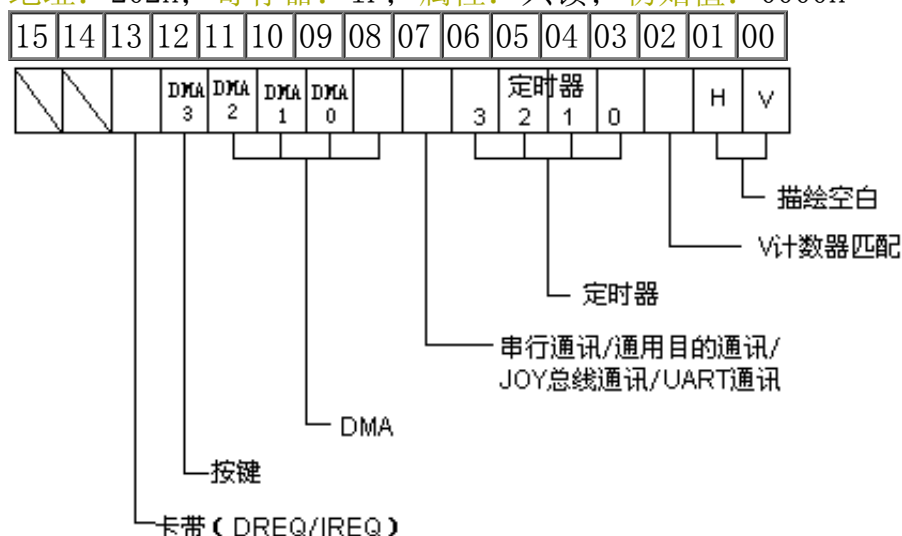


通过复位这些位，响应的中断可以被禁止，设为1使能响应的中断。

中断请求寄存器:

当每个硬件设备发生一个中断请求信号时，IF寄存器里响应的中断请求标志被设置。

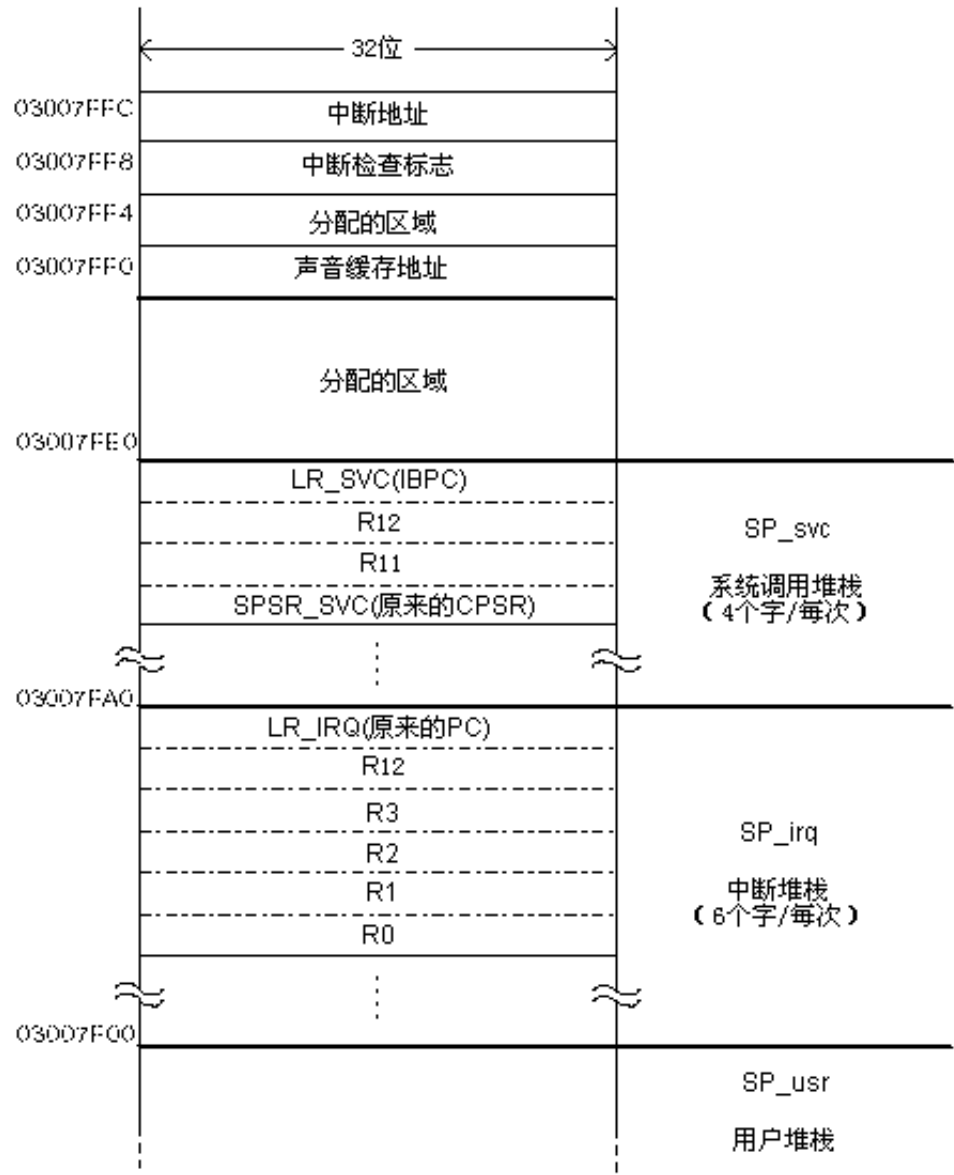
地址：202h；寄存器：IF；属性：只读；初始值：0000h



向中断请求标志位写1，使中断请求标志复位。

工作RAM内系统分配区域:

要控制中断继承，除了要清除IF寄存器和设置IE寄存器，首先要写一个中断跳转地址到工作RAM系统分配区域内的\$7FFC-\$7FFF（共32位，见下图），用户中断是以32位模式执行的。要从中断例程返回用户程序，使用“BX LR”指令。通过改变每个CPU模式SP初始值，他们可以被设置到任意的内存映射。

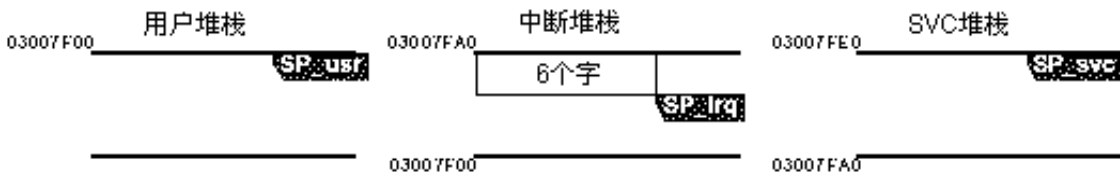


中断操作:

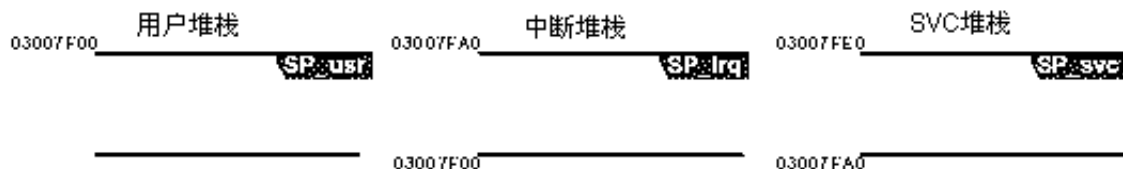
用户可以随意定义中断处理例程，但有一个通用规则，‘监控ROM’处理这些操作。每个寄存器的细节，请参阅“ARM7TDMI数据单”

普通中断:

如果发生一个中断，CPU进入IRQ模式，控制权转移到监视ROM。在监控ROM里，保存每个寄存器（R0~R3, R12, LR\_irq（原PC））到中断堆栈，共6个字。然后调用用户设在03007FFCh的中断例程。从监控器直接调用的命令必须是32位格式的。



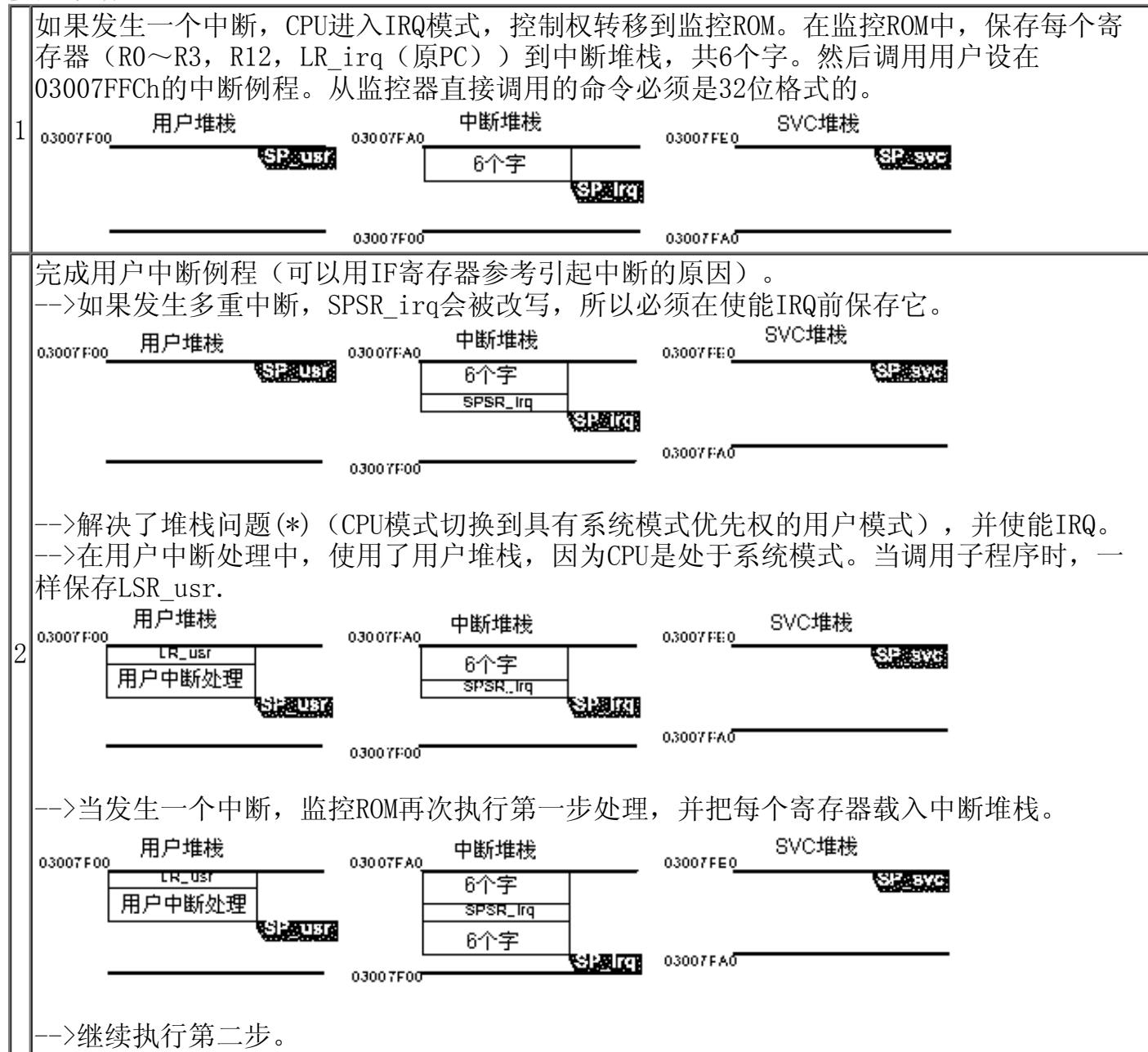
完成用户中断（可以用IF寄存器参考引起中断的原因）。如果必要，也可以用堆栈解决问题（\*）。重载保存到中断堆栈的寄存器（共6个字）并返回用户主例程。



注(\*):

只有中断堆栈用于普通中断处理。因此，在某些情况下有可能发生堆栈溢出。要解决这个问题，可以在用户中断例程中通过切换CPU模式到用户模式，或者事先移动SP\_usr分配大一点的中断堆栈，或同时使用用户堆栈。对于后一种办法，参加后面章节。

多重中断:



# 十六、节能功能

## 停止功能:

在LCD显示屏没有工作和CPU基本上没有运算的时候，如果有效地使用节能功能，就会大大减少电能的消耗。

## 执行停止:

执行停止模式: 执行系统调用[SWI<3>]指令(Stop()), 把GBA置于停止模式。

取消停止模式: 如果中断使能寄存器IE相应的标志被设置，当按键、卡带或SIO有一个中断请求时，停止被取消。

注意: 取消停止模式状态需要等待一小段时间，知道系统时钟稳定下来。

## 停止模式的系统工作状态:

GBA系统每个模块在停止模式的工作状态如下表所示:

模块	工作	状态
GBA的CPU	X	等待信号导致的等待状态
LPU	X	停止，因为没有时钟提供。因LPU停止，关闭LCD显示屏后进入停止模式
声音	X	停止，因声道停止，可能听到一声噪音
定时器	X	停止
串行通讯	X	停止
按键	X	停止
系统时钟	X	停止
红外通讯	X	停止

## 悬挂功能:

在CPU基本上没有运算的时候，如果有效地使用节能功能，就会大大减少电能的消耗。

## 执行悬挂:

调用悬挂: 执行系统调用[SWI<2>]指令(Halt()), 把GBA置于悬挂模式。

取消悬挂: 当中断使能寄存器IE设置的标志中任一个有中断请求时，取消悬挂。

## 悬挂模式的系统工作状态:

GBA系统在半停止时的工作状态如下表所示:

模块	工作	状态
GBA的CPU	X	等待信号导致的等待状态
LPU	○	正常工作
声音	○	正常工作
定时器	○	正常工作
串行通讯	○	正常工作
按键	○	正常工作
系统时钟	○	正常工作



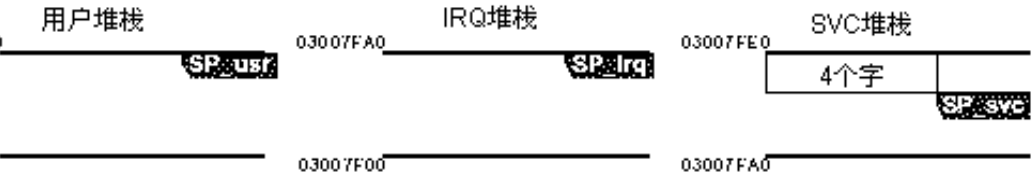
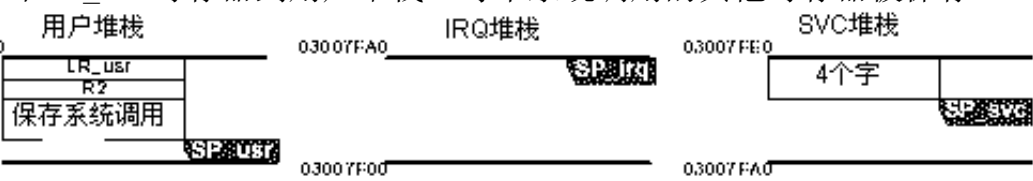




# 十七、GBA系统调用


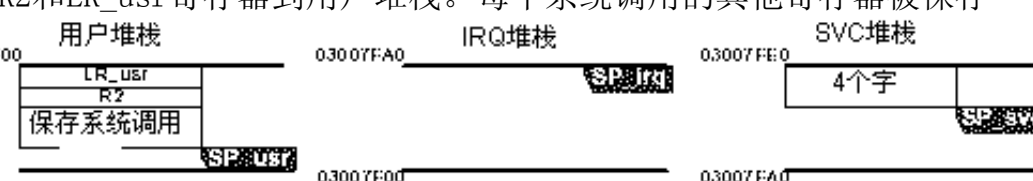
请参考GBA系统调用手册。

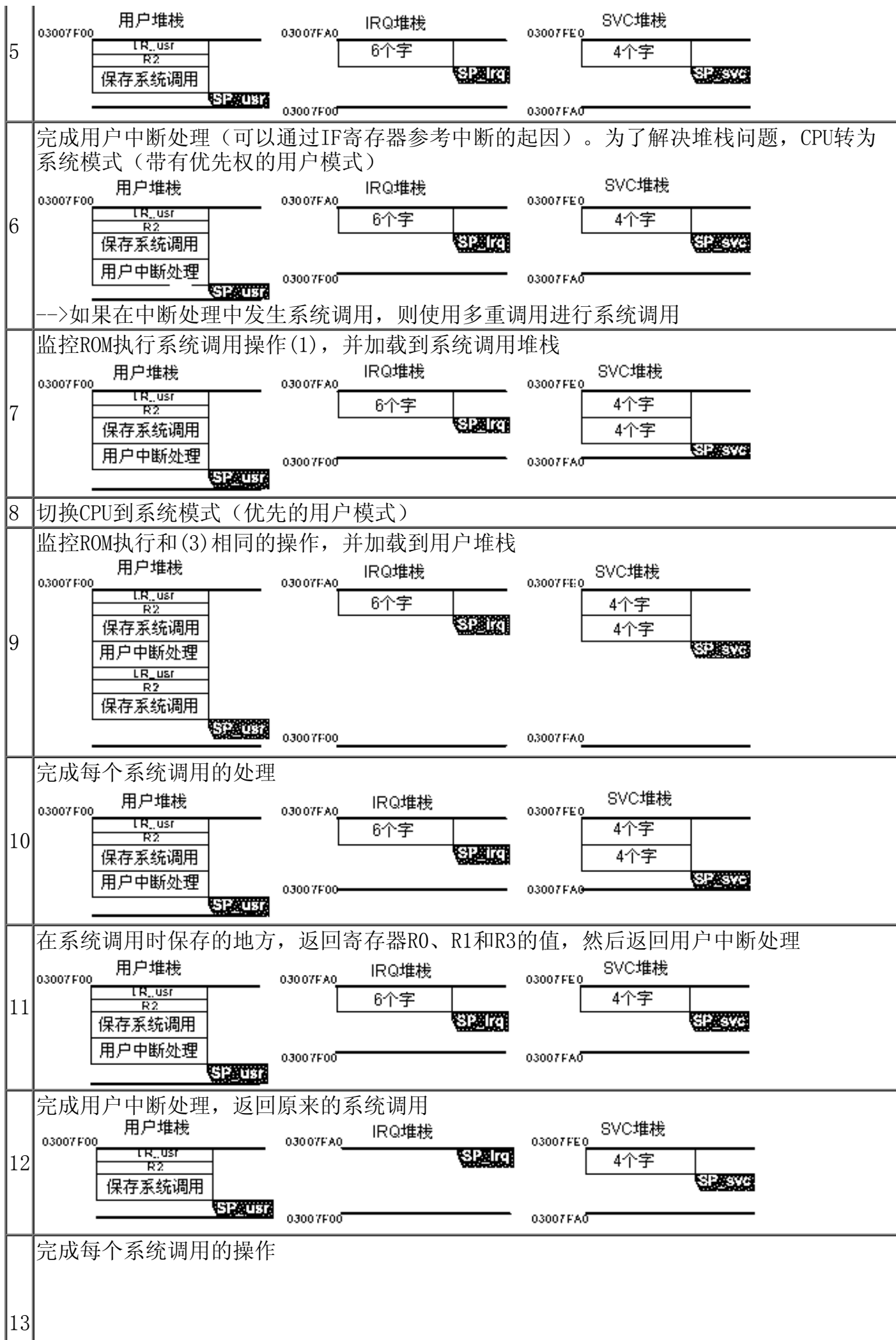
## 系统调用操作：

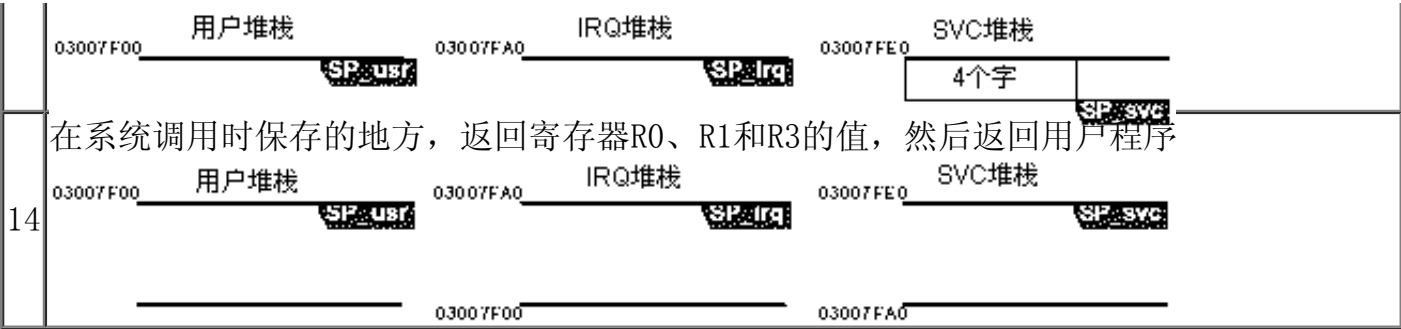
普通调用：

1	当系统调用需要使用一个参数时，在写到寄存器R0—R3之后用“SWI<Number>”调用监控ROM的系统调用。CPU切换到管理模式
2	用监控ROM保存寄存器SPSR_svc（原来的CPSR）、R11、R12、LR_svc（原来的PC）到系统调用堆栈 
3	从CPU模式切换到系统模式。用监控ROM调用IRQ禁止标志。继续原来的状态。
4	保存R2和LR_usr寄存器到用户堆栈。每个系统调用的其他寄存器被保存 
5	完成每个系统调用的处理 
6	在系统调用时保存的地方，返回寄存器R0、R1和R3的值，然后返回用户程序 

多重调用：

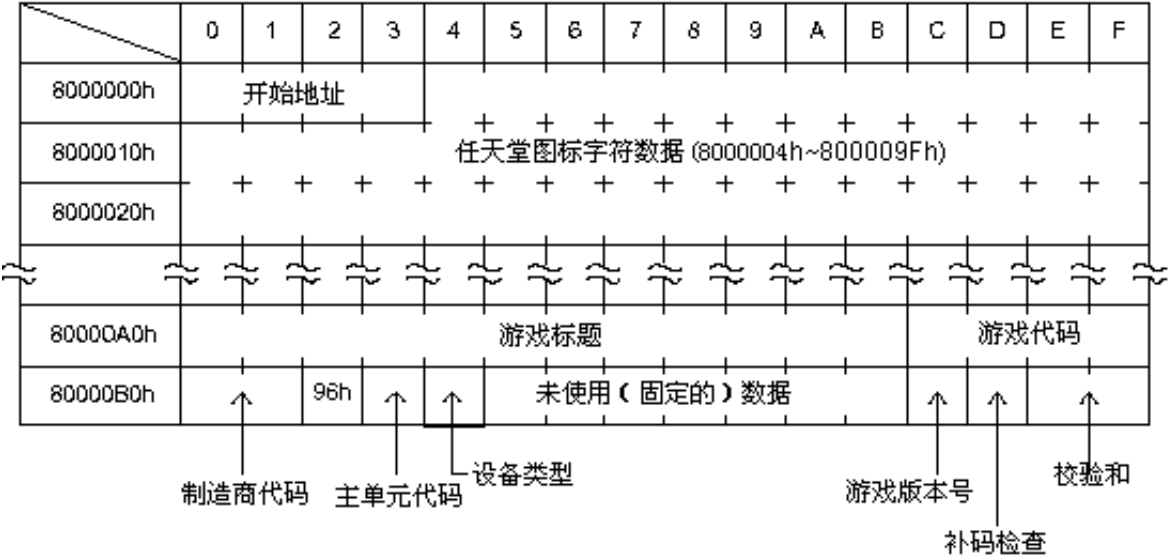
1	当系统调用需要使用一个参数时，在写到寄存器R0—R3之后用“SWI<Number>”调用监控ROM的系统调用
2	用监控ROM保存寄存器SPSR_svc（原来的CPSR）、R11、R12、LR_svc（原来的PC）到系统调用堆栈 
3	从CPU模式切换到系统模式。用监控ROM调用IRQ禁止标志。继续原来的状态。
4	保存R2和LR_usr寄存器到用户堆栈。每个系统调用的其他寄存器被保存 
	在执行系统调用时发生中断





# 十八、ROM寄存数据

对于GBC的软件，必须在GBA软件的程序区域驻留一些关于游戏的信息。



开始地址：  
储存32位ARM命令“B<用户程序开始地址>”。

任天堂图标字符数据：  
在游戏开始时显示的任天堂图标字符数据储存在这里。在起动时监控ROM检查这里，因此总是存放任天堂提供的数据。

游戏标题：  
在这里储存游戏标题。

游戏代码：  
在这里储存由任天堂提供的游戏代码。

制造商代码：  
在这里储存由软件制造商和任天堂确定的代码。

固定值：  
储存固定代码“96h”。

主单元代码：  
储存可运行软件的硬件代码。

设备类型：  
储存卡带的设备类型。如果在卡带里有一个1M位的快速DACS（调试和通讯系统）（=消耗1M位带安全和补丁功能的闪存），设置最高有效位为1，否则位0。其他位是分配区域。

软件版本号：  
储存ROM版本号。

补码检查：  
储存在地址80000A0h～80000BCh的数据加上19h的的和的2的补码储存在这里。

校验和：  
分别在地址80000BEh和80000BFh储存0。然后把ROM的数据一个字节一个字节地加起来确定一个总二进制数。用计算结果的低2个字节替换地址80000BEh和80000BFh的数据。



## GBA系统参考

(张晓波 译、编 2002-10-12)

## 附一、GBA数据格式:

调色板RAM: 05000000h-050001FFh(BG调色板)、05000200h-050003FF(OBJ调色板)

地址	数据	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	注释
任意	颜色数据	—	蓝色					绿色			红色							

VRAM: 06000000h-06017FFFh

### BG模式0-2（字符模式）

地址	数据	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	注释	
设定	16色字符数据	颜色号				颜色号				颜色号				颜色号				4位	
设定	256色字符数据	颜色号								颜色号								8位	
设定	文本GB屏幕数据	调色板号				垂直翻 转	水平翻 转	字符名											
设定	旋转/滚动BG屏幕数据	字符名							字符名										8位

### BG模式3—5（位图模式）

地址	数据	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	注释
任意	32, 768色像素数据	—	蓝色						绿色				红色					
任意	256色像素数据	颜色号								颜色号								8位

OAM: 07000000h—070003FFh

地址	数据	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	注释	
指定	精灵属性0	精灵格式		颜色模式	马赛克	精灵模式		双倍大小标志	旋转/缩放	Y坐标									
指定	精灵属性1	精灵尺寸		(垂直翻转) 选择		(水平翻转) 选择/缩放参数选择								X坐标					
指定	精灵属性2	调色板号				优先权		字符名											
指定	PA	同一行X方向移动距离																	
指定	PB	下一行X方向移动距离																	
指定	PC	同一行Y方向移动距离																	
指定	PD	下一行Y方向移动距离																	

### 附二、寄存器表:

[illegible]

地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值
00	DISPCNT	窗口显示标志			显示标志					强制空白	精灵数据映射格式	精灵水平扫描关闭	帧缓存号	GBC模式	BG模式			读写	0080h
		精灵	窗口1	窗口0	精灵	BG3	BG2	BG1	BG0										
04	DISPSTAT	V计数器设置								—	—	V计数器匹配中断	水平空白中断	垂直空白中断	V计数器相等	水平空白状态	垂直空白状态	读写	0000h
06	VCOUNT	—	—	—	—	—	—	—	—	V计数器值								读	0000h
08	BG0CNT	尺寸		—	屏幕基准块					颜色模式	马赛克	0	0	字符基准块	优先权		读写	0000h	
0A	BG1CNT	尺寸		—	屏幕基准块					颜色模式	马赛克	0	0	字符基准块	优先权		读写	0000h	
0C	BG2CNT	尺寸		区域溢出	屏幕基准块					颜色模式	马赛克	0	0	字符基准块	优先权		读写	0000h	
0E	BG3CNT	尺寸		区域溢出	屏幕基准块					颜色模式	马赛克	0	0	字符基准块	优先权		读写	0000h	
10	BG0H0FS	—	—	—	—	—	—	—	—	水平偏移								写	0000h
12	BG0V0FS	—	—	—	—	—	—	—	—	垂直偏移								写	0000h
14	BG1H0FS	—	—	—	—	—	—	—	—	水平偏移								写	0000h
16	BG1V0FS	—	—	—	—	—	—	—	—	垂直偏移								写	0000h
18	BG2H0FS	—	—	—	—	—	—	—	—	水平偏移								写	0000h
1A	BG2V0FS	—	—	—	—	—	—	—	—	垂直偏移								写	0000h
1C	BG3H0FS	—	—	—	—	—	—	—	—	水平偏移								写	0000h
1E	BG3V0FS	—	—	—	—	—	—	—	—	垂直偏移								写	0000h
20	BG2PA	dx： 同一行X方向移动距离																写	0100h
22	BG2PB	dmx： 下一行X方向移动距离																写	0000h
24	BG2PC	dy： 同一行Y方向移动距离																写	0100h
26	BG2PD	dmy： 下一行Y方向移动距离																写	0000h
28	BG2X_L	X方向引用开始点（旋转/缩放结果）																写	0000h
2A	BG2X_H	—	—	—	—	X方向引用开始点（旋转/缩放结果）											写	0000h	

2C	BG2Y_L	Y方向引用开始点（旋转/缩放结果）																写	0000h
2E	BG2Y_H	Y方向引用开始点（旋转/缩放结果）																写	0000h
地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值
30	BG3PA	dx： 同一行X方向移动距离																写	0100h
32	BG3PB	dmx： 下一行X方向移动距离																写	0000h
34	BG3PC	dy： 同一行Y方向移动距离																写	0000h
36	BG3PD	dmy： 下一行Y方向移动距离																写	0100h
38	BG3X_L	X方向引用开始点（旋转/缩放结果）																写	0000h
3A	BG3X_H	—	—	—	—	X方向引用开始点（旋转/缩放结果）											写	0000h	
3C	BG3Y_L	Y方向引用开始点（旋转/缩放结果）																写	0000h
3E	BG3Y_H	—	—	—	—	Y方向引用开始点（旋转/缩放结果）											写	0000h	
40	WIN0H	窗口0左上角X坐标								窗口0右下角X坐标								写	0000h
42	WIN1H	窗口1左上角X坐标								窗口1右下角X坐标								写	0000h
44	WIN0V	窗口0左上角Y坐标								窗口0右下角Y坐标								写	0000h
46	WIN1V	窗口1左上角Y坐标								窗口1右下角Y坐标								写	0000h
48	WININ	—	—	窗口1内部控制						—	—	窗口0内部控制						写	0000h
				特效	精灵	BG3	BG2	BG1	BG0			特效	精灵	BG3	BG2	BG1	BG0	写	0000h
4A	WINOUT	—	—	精灵窗口控制						—	—	窗口0和窗口1外部控制						写	0000h
				特效	精灵	BG3	BG2	BG1	BG0			特效	精灵	BG3	BG2	BG1	BG0	写	0000h
4C	MOSAIC	精灵马赛克								BG马赛克								写	0000h
		垂直尺寸				水平尺寸				垂直尺寸				水平尺寸					
50	BLDMOD	—	—	第二目标像素						颜色特效类型	第一目标像素						读写	0000h	
				背景幕	精灵	BG3	BG2	BG1	BG0		背景幕	精灵	BG3	BG2	BG1	BG0			
52	COLEV	—	—	—	EVB颜色特效系数						—	—	—	EVA颜色特效系数				写	0000h
54	COLEY	—	—	—	—	—	—	—	—	—	—	—	EVY颜色特效系数				写	0000h	
60	SG10_L	—	—	—	—	—	—	—	—	NR10								读写	0000h
62	SG10_H	NR12								NR11								读写	0000h
64	SG11	NR14								NR13								读写	0000h
68	SG20	NR22								NR21								读写	0000h
6C	SG21	NR24								NR23								读写	0000h
70	SG30_L	—	—	—	—	—	—	—	—	NR30								读写	0000h
72	SG30_H	NR32								NR31								读写	0000h
74	SG31	NR34								NR33								读写	0000h
78	SG40	NR42								NR41								读写	0000h



7C	SG41	NR44								NR43								读写	0000h
地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值
80	SGCNT0_L	NR51								NR50								读写	
82	SGCNT0_H	直接声道B				直接声道A				—	—	—	—	直接声道B输出比例	直接声道B输出比例	声道1—4混合比例	读写	0000h	
		FIFO B复位	定时器	L 输出	R 输出	FIFO A复位	定时器	L 输出	R 输出										
84	SGCNT1	—	—	—	—	—	—	—	—	NR52								读写	0000h
88	SGBIAS	幅值采样转换周期		—	—	—	—	偏压级										读写	0000h
90	SGWR0_L	Step2				Step3				Step0			Setp1			读写	—		
92	SGWR0_H	Step6				Step7				Step4			Setp5			读写	—		
94	SGWR1_L	Step10				Step11				Step8			Setp9			读写	—		
96	SGWR1_H	Step14				Step15				Step12			Setp13			读写	—		
98	SGWR2_L	Step18				Step19				Step16			Setp17			读写	—		
9A	SGWR2_H	Step22				Step23				Step20			Setp21			读写	—		
9C	SGWR3_L	Step26				Step27				Step24			Setp25			读写	—		
9E	SGWR3_H	Step30				Step31				Step28			Setp29			读写	—		
A0	SGFIFOA_L	声音数据1								声音数据0								写	—
A2	SGFIFOA_H	声音数据3								声音数据2								写	—
A4	SGFIFOB_L	声音数据1								声音数据0								写	—
A6	SGFIFOB_H	声音数据3								声音数据2								写	—
B0	DMOSAD_L	DMA0源地址																写	0000h
B2	DMOSAD_H	—	—	—	—	—	DMA0源地址										写	0000h	
B4	DMODAD_L	DMA0目标地址																写	0000h
B6	DMODAD_H	—	—	—	—	—	DMA0目标地址										写	0000h	
B8	DMOCNT_L	—	—	字计数														写	
BA	DMOCNT_H	DMA0控制																读写	0000h
		使能	中断	开始时刻	—	带宽	连续	源地址控制	目标地址控制	—	—	—	—	—					

BC	DM1SAD_L	DMA1源地址																写	0000h
BE	DM1SAD_H	—	—	—	—	—	DMA1源地址											写	0000h
地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值
C0	DM1DAD_L	DMA1目标地址																写	0000h
C2	DM1DAD_H	—	—	—	—	—	DMA1目标地址											写	0000h
C4	DM1CNT_L	—	—	字计数														写	
C6	DM1CNT_H	DMA1控制																读写	0000h
		使能	中断	开始时刻	—	带宽	连续	源地址控制	目标地址控制	—	—	—	—	—					
C8	DM2SAD_L	DMA2源地址																写	0000h
CA	DM2SAD_H	—	—	—	—	—	DMA2源地址											写	0000h
CC	DM2DAD_L	DMA2目标地址																写	0000h
CE	DM2DAD_H	—	—	—	—	—	DMA2目标地址											写	0000h
D0	DM2CNT_L	—	—	字计数														写	
D2	DM2CNT_H	DMA2控制																读写	0000h
		使能	中断	开始时刻	—	带宽	连续	源地址控制	目标地址控制	—	—	—	—	—					
D4	DM3SAD_L	DMA3源地址																写	0000h
D5	DM3SAD_H	—	—	—	—	—	DMA3源地址											写	0000h
D8	DM3DAD_L	DMA3目标地址																写	0000h
DA	DM3DAD_H	—	—	—	—	—	DMA3目标地址											写	0000h
DC	DM3CNT_L	—	—	字计数														写	0000h
DE	DM3CNT_H	DMA3控制																读写	0000h
		使能	中断	开始时刻	DREQ	带宽	连续	源地址控制	目标地址控制	—	—	—	—	—					
100	TM0D	定时器0设置																写	0000h
102	TM0CNT	定时器0控制																读写	0000h
		—	—	—	—	—	—	—	—	工作	中断	—	—	—	计数开始时刻	预引比例			
104	TM1D	定时器1设置																写	0000h
106	TM1CNT	定时器1控制																读写	0000h
		—	—	—	—	—	—	—	—	工作	中断	—	—	—	计数开始时刻	预引比例			
108	TM2D	定时器2设置																写	0000h
		定时器2控制																	
															计				

10A	TM2CNT	—	—	—	—	—	—	—	—	工作	中断	—	—	—	数 开 始 时 刻	预 引 比 例	读写	0000h	
10C	TM3D	定时器3设置																写	0000h
10E	TM3CNT	定时器3控制																读写	0000h
		—	—	—	—	—	—	—	—	工作	中断	—	—	—	计 数 开 始 时 刻	预 引 比 例			
地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值
120	SCD0	32位普通SIO通讯数据和多玩家通讯数据0																读写	0000h
122	SCD1	32位普通SIO通讯数据和多玩家通讯数据1																读写	0000h
124	SCD2	多玩家通讯数据2																读写	0000h
126	SCD3	多玩家通讯数据3																读写	0000h
128	SSCNT_T	端口控制				SIO控制													
	普通SIO 通讯	—	中 断 使 能	0	传 输 位 长	—	—	—	—	开 始	—	—	—	发 送 使 能 标 志	接 收 使 能 标 志	移 位 时 钟 频 率	移 位 时 钟	读写	0000h
	多玩家 通讯	—	中 断 使 能	1	0	—	—	—	—	开 始 (主 机) 忙 (副 机)	通 讯 错 误 标 志	多 玩 家 ID		SD 引 脚 监 视	SI 引 脚 监 视	波特率		读写	0000h
	UART通讯	—	中 断 使 能	1	1	接 收 使 能 标 志	发 送 使 能 标 志	校 验 使 能 标 志	FIFO 使 能 标 志	数 据 长 度	错 误 标 志	接 收 数 据 标 志	发 送 数 据 标 志	校 验 控 制	CTS 标 志	波特率		读写	0000h
12A	SSCNT_H	通讯数据																	
	普通SIO 通讯	—	—	—	—	—	—	—	—	8位普通SIO通讯数据								读写	0000h
130	P1	—	—	—	—	—	—	L	R	下	上	坐	右	开 始	选 择	B	A	读写	0000h
132	P1CNT	按键中断控制																读	0000h
		中 断	中 断												开	选			

		条件	使能	—	—	—	—	L	R	下	上	坐	右	开始	选择	B	A	写		
134	R	通讯功能选择		—	—	—	—	—	中断使能	输入输出选择				数据位				读写	0000h	
										S0	SI	SD	SC	S0	SI	SD	SC			
140	HS_CTRL	JOY总线通讯控制																	读写	0000h
		—	—	—	—	—	—	—	—	—	中断	—	—	—	发送完成	接收完成	接受复位			
150	JOYRE_L	JOY总线通讯接收数据0																	读写	0000h
152	JOYRE_H	JOY总线通讯接收数据1																	读写	0000h
154	JOYTR_L	JOY总线通讯发送数据0																	读写	0000h
156	JOYTR_H	JOY总线通讯发送数据1																	读写	0000h
158	JSTAT	JOY总线通讯接收状态																	读写	0000h
		—	—	—	—	—	—	—	—	—	—	通用的标志	发送状态标志	—	接收状态标志	—				
200	IE	中断使能标志																	读写	0000h
		—	—	卡带	按键	DMA3	DMA2	DMA1	DMA0	全部SIO	定时器3	定时器2	定时器1	定时器0	V计数匹配	水平空白	垂直空白			
202	IF	中断请求标志																	读写	0000h
		—	—	卡带	按键	DMA3	DMA2	DMA1	DMA0	全部SIO	定时器3	定时器2	定时器1	定时器0	V计数匹配	水平空白	垂直空白			
204	WSCNT	卡带	预取	—	PHI引脚输出控制			等待状态2等待控制			等待状态1等待控制			等待状态0等待控制			SRAM等待控制		读写	0000h
208	IME	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	中断屏蔽标志	读写	0000h
300	PAUSE	节能	模式	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000h
地址	寄存器	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	读写	初值	