GAME BOY® ADVANCE

Programming Manual

Version 1.35

AGB-06-0001-002-B13
Released: May 27, 2005

Table of Contents

Contents

Tables

Figures

Equations

Revision History

Revision No.	Date Revised	Description
1.35	3/7/2005	• Added "Precaution regarding memory map" to "3.1 Overall Memory Map" on page 7.
1.34a	1/26/2004	• Corrected note in "10.8 Sound PWM Control" on page 87.
1.34	10/6/2003	• Specified that for color special effects, α blending cannot be performed between OBJs. • Corrected an erroneous entry for the brightness adjustment calculation method (G) in color special effects. • Corrected an erroneous entry in the normal serial communications flow diagram. • Added that the IF register's H-Blank interrupt request can also occur during V-Blank.
1.32	1/30/2003	• Changed explanation of the start bit and busy flag in the `SIOCNT` register under "13.2 16-Bit Multi-player Communication" on page 113. • Changed the explanation of the `JOYSTAT` register under "13.5 JOY Bus Communication" on page 126. • Added recommended text strings to "16.1.5 Guidelines for Use of the Stop Function" on page 142. • In Tables 24, 25, and 26, added that the H-Blank interval startup for DMA can only occur during the H-Blank for the display interval.
1.30	3/18/2002	• Added a JOY Bus communication caution • Corrected the section about normal serial communications to read that the state of the master's SD terminal "outputs a LO" instead of "pull-up input." • Added guidelines on use of the Stop (Sleep) feature.
1.25	10/26/2001	• Fixed the bit contents of general-purpose communication register R. • Specified the number of lost cycle of OBJ outside of rendering line in "OBJ Function Overview". • Changed the command names of "JOY Bus Communication. (Device Reset -> JOY Bus Reset, GBA Data Write -> JOY Bus Data Write, GBA Data Read -> JOY Bus Data Read) • Modified conditions for canceling Stop Mode. • Added to [Stop Function Cautions]. • Changed the explanation of Device Type and added a diagram of ROM registration data.

Revision No.	Date Revised	Description
1.22	8/10/2001	• Fixed section explaining Game Pak interrupts in the “Interrupt Control” chapter. • Added additional information to explanations about the DISPCNT register’s Individual Screens Display Flag and Forced Blank.
1.20	7/24/2001	• Modified the Complete Block Diagram for the system architecture. • Modified the cautions for priority in the “Display Priority of OBJ and BG” section. • Fixed the explanation at the beginning of the “DMA Transfer” chapter. • Added section explaining Game Pak interrupts to the “Interrupt Control” chapter. • Added cautions to the “Interrupt Control” chapter. • Fixed the Multi-Play communication flowchart. • Fixed the BLDALPHA register so that reading and writing are both possible.
1.15	5/7/2001	• Added cautions to “Communication Function” chapter. • Added cautions to “Key Input” chapter. • Added cautions to “Stop Function” chapter. • Added to explanation of functions at the beginning of “DMA Transfer” chapter. • Modified the flowchart for Multi-Play communication.

Revision No.	Date Revised	Description
1.1	4/2/2001	• Changed the picture in the Game Boy Advance introduction in the beginning paragraph. • Added a caution regarding clearing of IME and IE in the chapter “Interrupt Control”. • Added additional description of an error flag and ID flag for multi-play communication. • Added additional description of communication error flag of multi-play communication control register. • Modified the host side example in the description of JOY bus communication from NUS to DOL. Added DOL to the abbreviation in “Using This Manual”. • Modified the SIO timing chart for normal serial communication. • Revised the number of colors from 256 to 32,768 in the description of Display Synchronization DMA of DMA3. • Modified the description of general-purpose communication mode. • Revised the caution for normal serial communication. • Revised the caution for communication function. • Revised the summary of normal serial communication in the communication function chapter, and added additional description. • Added additional description in the caution for the selection of communication function in the communication function chapter. • Emphasized that unless general-purpose communication mode, the cancellation condition SIO for System Call Stop will not work. • Changed LPU to LCD controller in system calls Halt and Stop. • Deleted the first item in Sound 3 Usage Note.

Revision No.	Date Revised	Description
1.1 (Cont.)	4/2/2001	• Changed the names of following registers according to header files provided by Nintendo. **--Wait Control--** 204h WSCNT _ WAITCNT **--Color Special Effects--** 050h BLDMOD _ BLDCNT 052h COLEV _ BLDALPHA 054h COLY _ BLDY **--Sound Related--** 080h~ SGCNT0_(L H) _ SOUNDCNT_(L H) ** Combined multiple names 084h SGCNT1 _ SOUNDCNT_X 088h SG_BIAS _ SOUNDBIAS 060h~ SG10_(L H) _ SOUND1CNT_(L H) ** 064h SG11 _ SOUND1CNT_X 068h SG20 _ SOUND2CNT_L 06Ch SG21 _ SOUND2CNT_H 070h~ SG30_(L H) _ SOUND3CNT_(L H) ** 074h SG31 _ SOUND3CNT_X 078h SG40 _ SOUND4CNT_L 07Ch SG41 _ SOUND4CNT_H 090h~ SGWR(0-3)_L _ WAVE_RAM(0-3)_L ** 092h~ SGWR(0-3)_H _ WAVE_RAM(0-3)_H ** 0A0h~ SG_FIFOA_(L H) _ FIFO_A_(L H) ** 0A4h~ SG_FIFOB_(L H) _ FIFO_B_(L H) ** **--DMA Related--** 0B0h~ DM(0-3)SAD_L _ DMA(0-3)SAD_L ** 0B2h~ DM(0-3)SAD_H _ DMA(0-3)SAD_H ** 0B4h~ DM(0-3)DAD_L _ DMA(0-3)DAD_L ** 0B6h~ DM(0-3)DAD_H _ DMA(0-3)DAD_H ** 0B8h~ DM(0-3)CNT_L _ DMA(0-3)CNT_L ** 0Bah~ DM(0-3)CNT_H _ DMA(0-3)CNT_H ** **--Timer Related--** 100h~ TM(0-3)D _ TM(0-3)CNT_L ** 102h~ TM(0-3)CNT _ TM(0-3)CNT_H ** **--Communication Related--** 134h R _ RCNT 128h SCCNT_L _ SIOCNT 12Ah SCCNT_H _ SIODATA8 (Normal serial, UART communication) SIOMLT_SEND (Multi-play communication) 120h SCD0 _ SIODATA32_L (Normal serial communication) SIOMULTI0 (Multi-play communication) 122h SCD1 _ SIODATA32_H (Normal serial communication) SIOMULTI1 (Multi-play communication) 124h~ SCD(2 3) _ SIOMULTI(2 3) ** 140h HS_CTRL _ JOYCNT 158h JSTAT _ JOYSTAT 150h~ JOYRE_(L H) _ JOY_RECV_(L H) ** 154h~ JOYTR_(L H) _ JOYTRANS_(L H) ** **--Key Related--** 130h P1 _ KEYINPUT 132h P1CNT _ KEYCNT

Revision No.	Date Revised	Description

1.04	3/1/2001	• Specified the method to control the OBJ display individually in the description of the double size flag and the rotation/scaling flag for OAM attribute 0.
		• Added the description of display synchronization DMA to DMA3.
		• Added the description of the DMA problem and how to avoid it at the end of the chapter on DMA.
		  • Added the restrictions to the description of the repeat flag in DMA3.
		  • Updated the timing chart and the cable connection diagram for the multi-play communication.
		  • Revised the description of the normal serial communication cautions.

1.02	2/13/2001	• Modified the description of “8-Bit/32-Bit Normal Communication Function” summary in “Communication” chapter. • Added a paragraph to “Selecting Communication Function” in “Communication” chapter.
1.01	2/01/2001	• Modified the description of pin 31 in the Game Pak bus. • Revised the cancel conditions for the Stop function in the power-down mode. • Added additional descriptions and cautions for the initialization flag of Sound 1.
1.0	12/01/2000	• Deleted the checksum of ROM registration data and revised the • diagram. • Revised the diagram for “Communication Cable” in the “Communication Function”. • Revised the number of DMG sold from tens of millions to a hundred million in the Game Boy Advance introduction. • Revised the hours you can play continuously from “about 20 hours” to “about 15 hours”. • Revised the illustrations of the Game Boy Advance hardware and the Multi Player Communication Cable in the multi play communication diagram. • Added the description of the timing chart for normal SIO communication. • Added a caution in the DMA valid flag of all the DMA control registers. • Added a caution in the master start bit of the multi-play control register. • Revised the multi-play timing chart. • Revised the memory map for system reserve area in the work RAM. • Added a caution to “Communication Function”. • Revised the first sentence in “UART Communication”. Added “Relation between Data register, FIFO and Shift register”.

Revision No.	Date Revised	Description
1.0 (Cont.)	12/01/2000	• Revised the expression of [Cautions] to a more specific expression [Cautions for ~~]. • Added a description of X coordinate and Y coordinate for OAM. Added the diagram to Y coordinate. • Revised the description of the pre-fetch buffer flag in the Game Pak memory wait control register. • Added cautions to the description of the input/output select flag in the R register of general communication. • Deleted the checksum of ROM registration data and revised the diagram. • Revised the diagram for “Game Boy Advance Game Link Cable” in the “Communication Function”. • Revised the number of DMG sold from tens of millions to a hundred million in the Game boy Advance introduction. • Revised the hours you can play continuously from “about 20 hours” to “about 15 hours”. • Revised the illustrations of the Game Boy Advance hardware and the Multi Player Game Boy Advance Game Link cable in the multi-play communication diagram. • Added the description of the timing chart for normal SIO communication. • Added a caution in the DMA valid flag of all the DMA control registers. • Added a caution in the master start bit of the multi-play control register. • Revised the multi-play timing chart. • Revised the memory map for system reserve area in the work RAM. • Added a caution to “Communication Function”. • Revised the first sentence in “UART Communication”. Added “Relation between Data register, FIFO and Shift register”. • Revised the expression of [Cautions] to a more specific expression [Cautions for ~~]. • Added a description of X coordinate and Y coordinate for OAM. Added the diagram to Y coordinate. • Revised the description of the pre-fetch buffer flag in the Game Pak memory wait control register. • Added cautions to the description of the input/output select flag in the R register of general communication.

Revision No.	Date Revised	Description
0.4.1.8	10/16/2000	• Added cautions to the priority setting of OBJ. • Added a description and cautions to Sound 1,2,3, and 4. • Added the description to “Mapping of character data”. • Revised the description in SCCNT_L[d14] and [06] of UART communication register. • Revised the connection diagram of 16 bit multi-play communication. • Added a description to all sound operation modes of the sound control register. • Revised the itemized description of Chapter 10 “Sound”.
0.4.1.7	08/10/2000	• Modified the description of an error flag for the multi-play control register. • Modified the description of a valid flag for all the DMA control registers. • Added the number of transfer when 0 is set for the DMA word count register.
0.4.1.6	06/26/2000	• Modified the connection diagram of the multi-play cable. • Added the transition diagram of the multi-play communication data. • Modified the description of “16-Bit Multi-play Communication”.
0.4.1.3	05/25/2000	• Changed the diagram in System-Allocated Area in Working RAM, and deleted “(Tentative)”. • Revised ROM registration data. • Corrected the description of internal shift clock of normal SIO control register. • Newly added the description of “Communication Cable” in the chapter of Communication Functions. • Corrected Overview of Screen Sizes for Text BG Screens in “Rendering Functions”.
	05/16/2000	• Added the diagram of Multi Player Communication Cable connection.
	05/08/2000	• Corrected [Sound 1 Usage Notes]. • In 1) Normal Communication of Communication Functions, mentioned not to use a cable.
0.4.1.2	04/06/2000	• Added the description of UART system communication.
0.4.1.1	03/10/2000	• Improved the description of interrupt and multiple interrupt process. • Improved the description of system call and multiple system call process.
	03/08/2000	• Added the description of ROM registration data.

Revision No.	Date Revised	Description
0.4.1	02/25/2000	• Changed the method to specify OBJ size. • Corrected misprints in the communication control register.
	02/24/2000	• Added the PWM sampling cycle control function.
	02/22/2000	• Modified the description of Direct Sounds, and corrected register R bit structure.
0.4.0	02/09/2000	• Added the Complete Block Diagram.
	01/25/2000	• Changed CPU internal working RAM memory capacity, and created CPU external working RAM. • Changed the bit structures of DMA control registers. • Deleted Infrared Communication functions. • Created the interrupt IME register, and changed the bit structures of IE and IF registers. • Changed the number of colors that can be displayed to 32,768. • Changed the specifications of Normal Serial Communication (bit width, communication speed) • Changed the specifications of Multi SIO Communication (UART system). • Changed the center coordinate of OBJ Rotation to dot boundary. • Added UART system communication function.
0.3.6.3	01/05/2000	• Minor modification. • Corrected BG Offset Registers diagrams • Corrected the diagrams of Registers for Setting the Direction Parameters of BG data. • Corrected diagram of the Sound 1 Duty Cycle. • Corrected the name of d05 bit for the DISPCNT Register. • Added the description of Bit map BG mode. • Corrected the SIO Timing Chart of Normal Serial Communication. • Changed the diagrams and descriptions of the Sound Control Registers. • Added the formula for calculating the number of OBJs that can be displayed on 1 line.
0.3.6.2	12/21/1999	• Minor modification. (Numbering for items P81,P82,P149; reference to chapter removed) • Deleted 14.3.

Introduction

Game Boy Advance (GBA, or sometimes AGB) stresses portability and focuses on 2D rather than 3D image processing functions, resulting in a cutting-edge portable game device with revolutionary capabilities.

It provides window-like functions, rotation, scaling, _ blending, and fade-in/fade-out features that can be combined to produce exactly the image representations desired.

Additionally, the bitmap image-rendering function, with its two modes (double buffering mode for rewriting full-screen images in real time and single buffering mode for stills), can be used to handle realistic images that are indistinguishable from actual photographs.

The 2.9-inch-wide reflective TFT color LCD screen provides a clear display with little afterimage.

In addition to Game Boy Color compatible sound, Game Boy Advance has a PCM stereo sound generator. Multiple tracks can be played simultaneously by overlapping them using the CPU. L and R buttons have been added to the Controller. The broader range of control provided also expands the breadth of game designs possible.

Although Game Boy Advance uses a 32-bit RISC CPU whose computing performance and data processing capabilities far surpass those of Game Boy Color, it consumes little power, allowing approximately 15 hours of continuous play. This is made possible by the inclusion of the various types of RAM on a single custom chip.

Furthermore, software for Game Boy Advance can be developed using the C language, minimizing the cost of development equipment. This favorable development environment and the high level of freedom of the system configuration allow one to build a profound world of play in which anyone can become absorbed.

With its extremely high-performance computational and data processing capabilities as a foundation, Game Boy Advance provides greater image and sound representation capabilities, making the pursuit of fun its essential aim.

The purpose of this high level of performance is to bring unique game ideas fully to life.

Game Boy Advance is an innovation born from experience. While providing backwards compatibility with the enormous software resources available for the 100 million Game Boy units in use worldwide, it also breaks new ground for portable game devices.

Using This Manual

Important terms and symbols used in this manual are defined below.

1. Terms

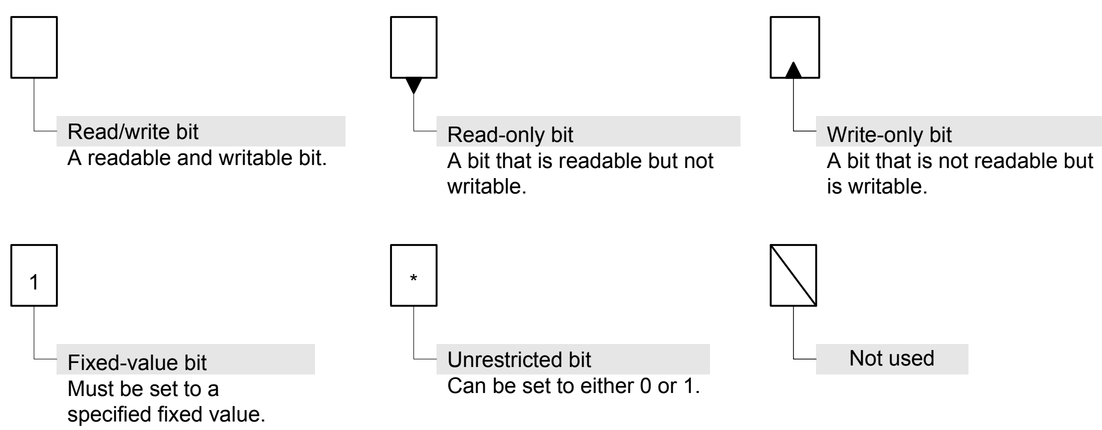
- The term “user” in this manual refers to the software developer, not to the general consumer.
- Bit lengths in this manual are expressed as follows:

Bit Length	Term Used
8 bits	byte
16 bits	half-word
32 bits	word

2. Symbols

The attributes of bits used in bit operations are represented as follows.

Figure 1 - Bit Operation Attribute Symbols



3. Abbreviations

Nintendo's game hardware is abbreviated as follows:

- DMG: Game Boy
- CGB: Game Boy Color
- GBA or AGB: Game Boy® Advance
- GCN: Nintendo GameCube™

1 The Game Boy Advance System

1.1 System Overview

Game Boy Advance is a portable game device that maintains backward compatibility with Game Boy Color (CGB) and provides higher performance.

The Game Boy Advance's 2.9-inch-wide reflective TFT color LCD and 32-bit RISC CPU enable production of games that match or surpass the Super Nintendo Entertainment System (Super NES) in performance.

The Game Boy Advance CPU

32-bit RISC CPU (ARM7TDMI)/16.78 MHz

Downward Compatibility with CGB

Integral 8-bit CISC CPU for compatibility.

(However, it cannot operate at the same time as the Game Boy Advance CPU.)

Memory

System ROM	16 KB (and 2 KB for CGB System ROM)
Working RAM	32 KB + CPU External 256 KB (2 wait)
VRAM	96 KB
OAM	64 bits x 128
Palette RAM	16 bits x 512 (256 colors for OBJ; 256 colors for BG)
Game Pak memory	Up to 32 MB: mask ROM or flash memory (& EEPROM) + Up to 512 kilobits: SRAM or flash memory

Display

- 240 x 160 x RGB pixels
- 32,768 colors simultaneously displayable
- Special effects features (rotation/scaling, α blending, fade-in/fade-out, and mosaic)
- 4 image system modes

Operation

Operating keys (A, B, L, R, START, SELECT, and Control Pad)

Sound

4 sounds (corresponding to CGB sounds) + 2 CPU direct sounds (PCM format)

Communication

Serial communication (8 bit/32 bit, UART, Multi-player, General-purpose, JOY Bus)

Game Pak

Like the original Game Boy and Game Boy Color, Game Boy Advance is equipped with a 32-pin connector for Game Pak connection. When a Game Pak is inserted, Game Boy Advance automatically detects its type and switches to either Game Boy Color or Game Boy Advance mode.

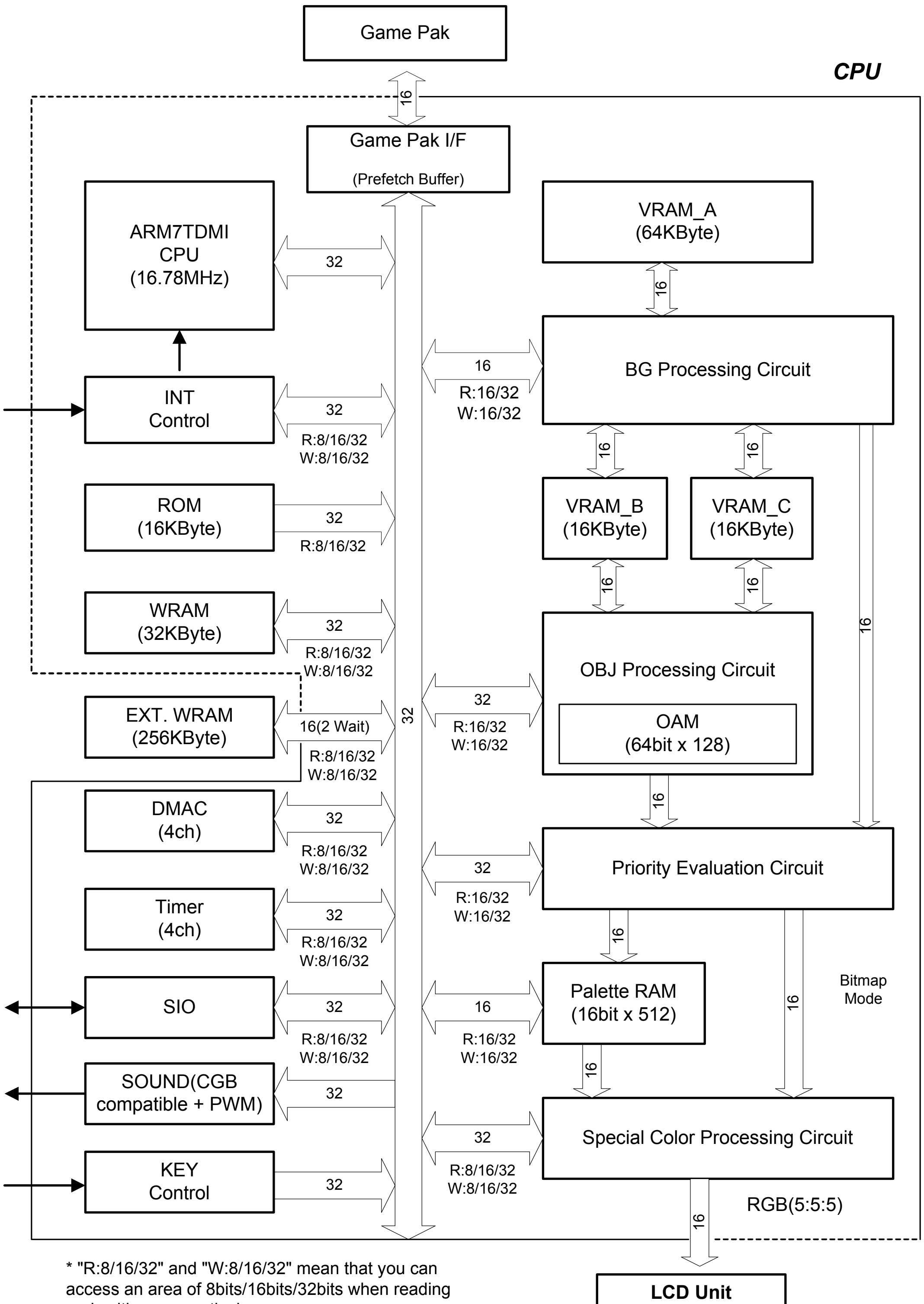
The following Game Paks operate on the Game Boy Advance system.

1. DMG Game Paks, DMG/CGB dual mode Game Paks, and CGB dedicated Game Paks
2. GBA-dedicated Game Paks (Game Paks that only function with Game Boy Advance)

2 System Configuration

2.1 CPU Block Diagram

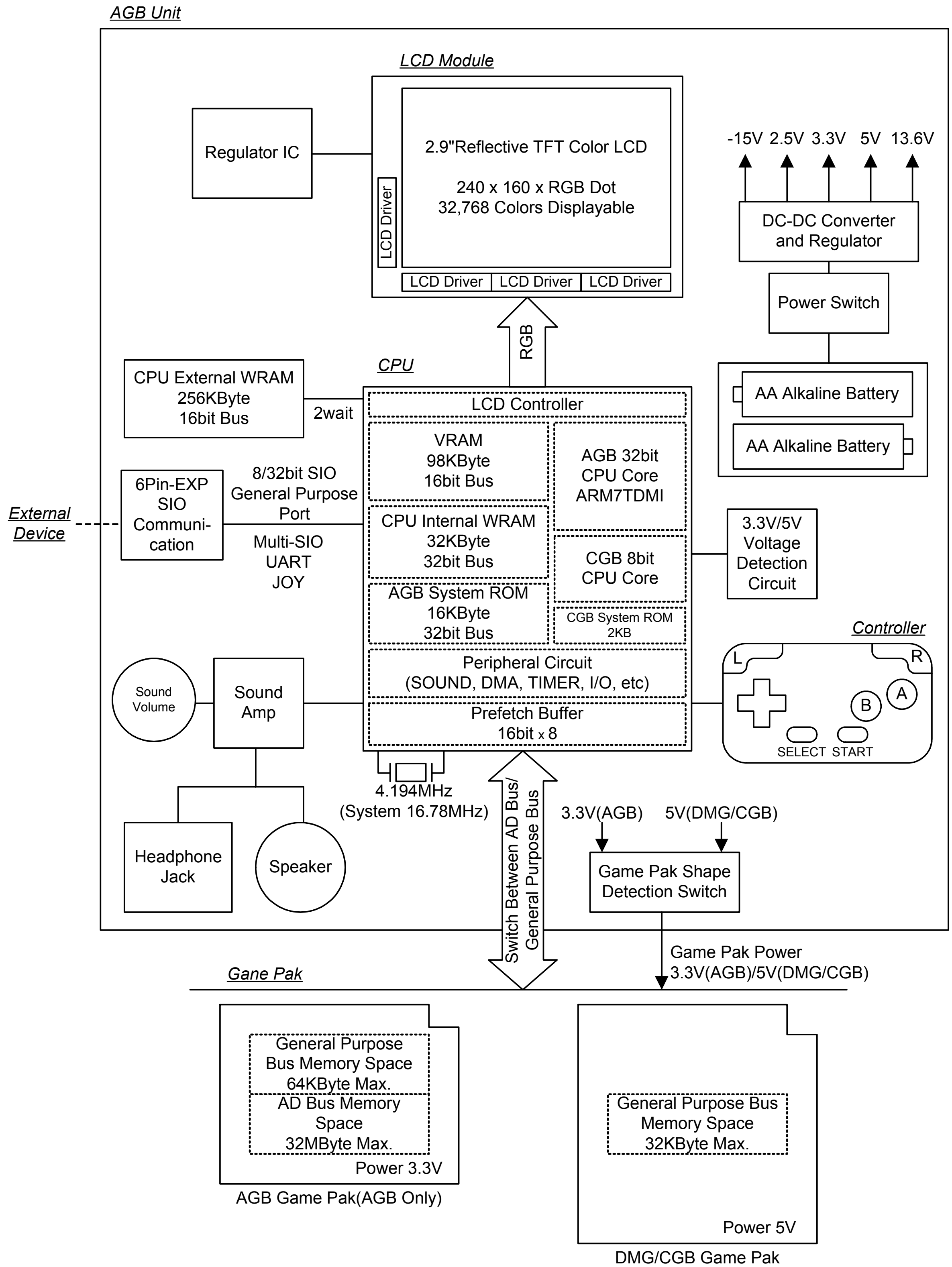
Figure 1 - Game Boy Advance CPU Block Diagram



* "R:8/16/32" and "W:8/16/32" mean that you can access an area of 8bits/16bits/32bits when reading and writing, respectively.

2.2 Complete Block Diagram

Figure 2 - Complete Game Boy Advance System Block Diagram



2.3 Memory Configuration and Access Widths

Table 1 - Game Boy Advance Memory Configuration and Access Widths

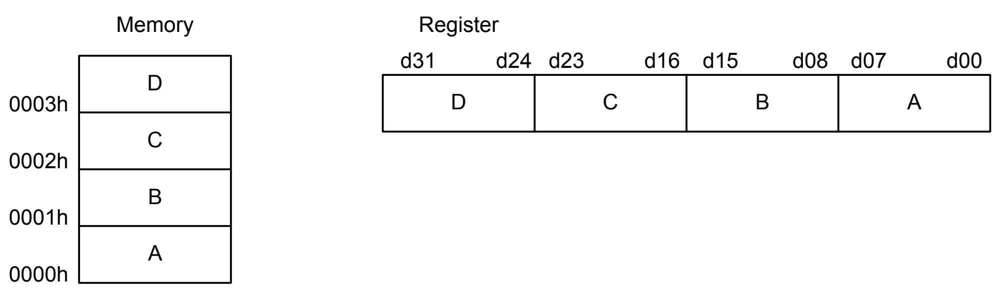
Memory Type	Bus Width	DMA		CPU	
		Read Width	Write Width	Read Width	Write Width
OAM	32	16/32	16/32	16/32	16/32
Palette RAM	16	16/32	16/32	16/32	16/32
VRAM	16	16/32	16/32	16/32	16/32
CPU Internal Working RAM	32	16/32	16/32	8/16/32	8/16/32
CPU External Working RAM	16	16/32	16/32	8/16/32	8/16/32
Internal registers	32	16/32	16/32	8/16/32	8/16/32
Game Pak ROM (Mask ROM, Flash Memory)	16	16/32	16/32	8/16/32	16/32
Game Pak RAM (SRAM, Flash Memory)	8	—	—	8	8

Good execution efficiency is obtained when programs that operate from the Game Pak use 16-bit instructions (16-bit compiler), and those that operate from CPU Internal Working RAM use 32-bit instructions (32-bit compiler).

2.4 Little-Endian

In the Game Boy Advance CPU, memory addresses are allocated in 8-bit increments, and little-endian format is used in implementing the 8-, 16-, and 32-bit access widths.

Figure 3 - Game Boy Advance CPU Memory Addresses (Little-Endian)

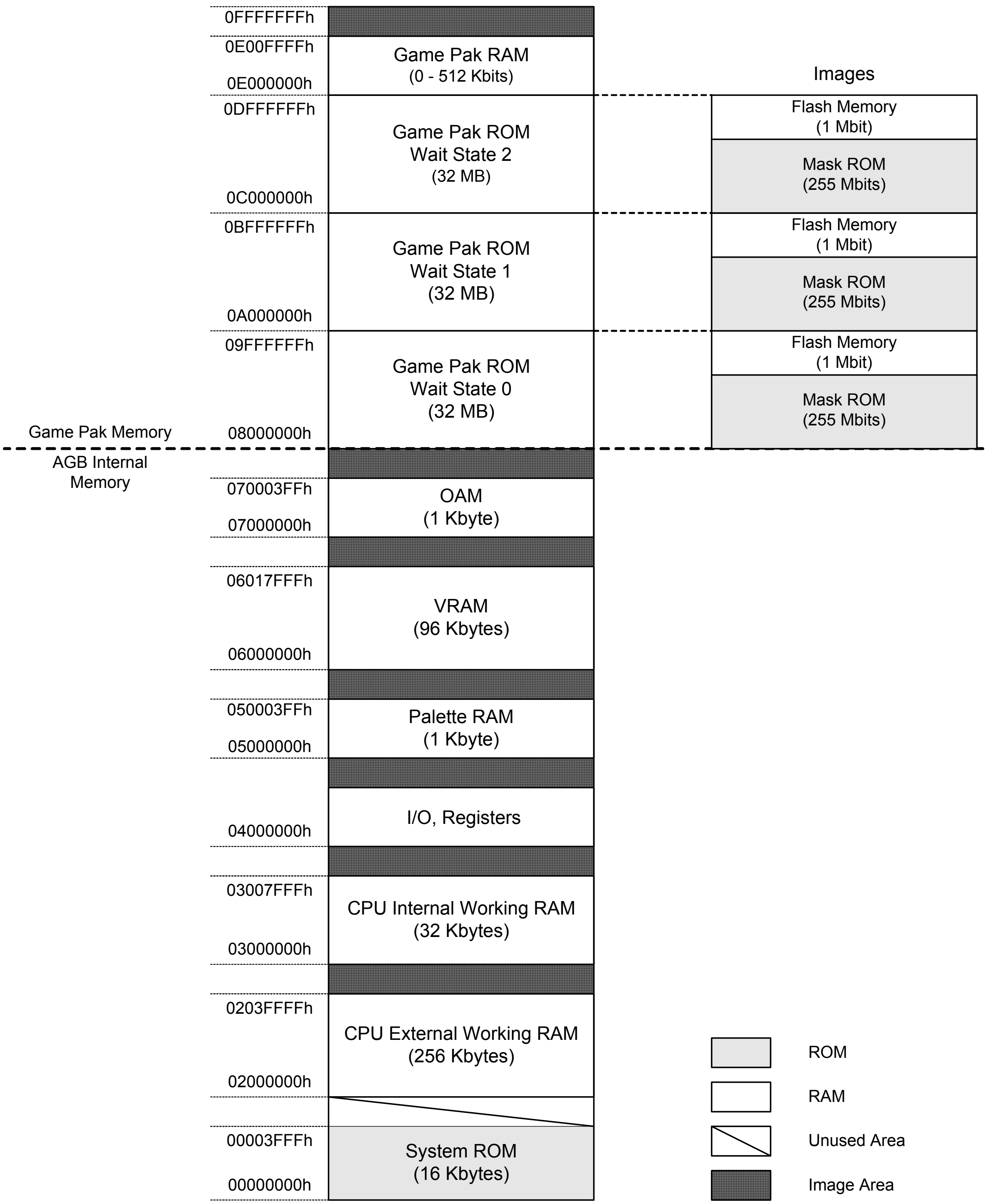


3 Game Boy Advance Memory

3.1 Overall Memory Map

The following is the overall memory map of the Game Boy Advance system.

Figure 4 - Game Boy Advance System Memory Map



Precaution Regarding Memory Map

Because the results obtained from accessing unmapped areas (memory and register addresses that are not mapped) are undefined in Game Boy Advance programming, do not perform processing based on these results.

Be aware that the memory access methods in Nintendo DS differ from those in Game Boy Advance, Game Boy Advance-SP, and Game Boy Player and therefore, the values obtained when reading unmapped areas are very likely to differ.

3.2 Memory Configuration

In broad terms, the area 00000000h-07FFFFFFh is allocated as Game Boy Advance internal memory, and 08000000-0EFFFFFFh is allocated as Game Pak memory.

3.2.1 Game Boy Advance Internal Memory

3.2.1.1 System ROM

The 16 KB from 000000000h is the system ROM.

Various types of System Calls can be used.

3.2.1.2 CPU External Working RAM

The 256 KB from 02000000h is CPU External Working RAM. Its specifications are 2 Wait 16 bit Bus.

3.2.1.3 CPU Internal Working RAM

The 32 KB from 03000000h is CPU Internal Working RAM. It is used to store programs and data.

3.2.1.4 I/O and Registers

This area is used for various registers.

3.2.1.5 Palette RAM

The 1 KB from 05000000h is palette RAM. It is used to assign palette colors.

3.2.1.6 VRAM

The 96 KB from 06000000h is the VRAM area. This area is for BG and OBJ data.

3.2.1.7 OAM

The 1 KB from 07000000h is Object Attribute Memory (OAM). It holds the objects to be displayed and their attributes.

3.2.2 Game Pak Memory

3.2.2.1 Game Pak ROM

Three 32 MB Game Pak ROM spaces are allocated to the area beginning from 08000000h.

The access speed of each of these spaces can be set individually. Thus, they are named Wait State 0, Wait State 1, and Wait State 2.

This specification enables memory of varying access speeds in Game Pak ROM to be accessed optimally.

The base addresses of the 3 spaces are 08000000h for Wait State 0, 0A000000h for Wait State 1, and 0C000000h for Wait State 2.

In addition, the upper 1 megabit of each space is allocated as flash memory.

This area is used primarily for saving data.

3.2.2.2 Game Pak RAM

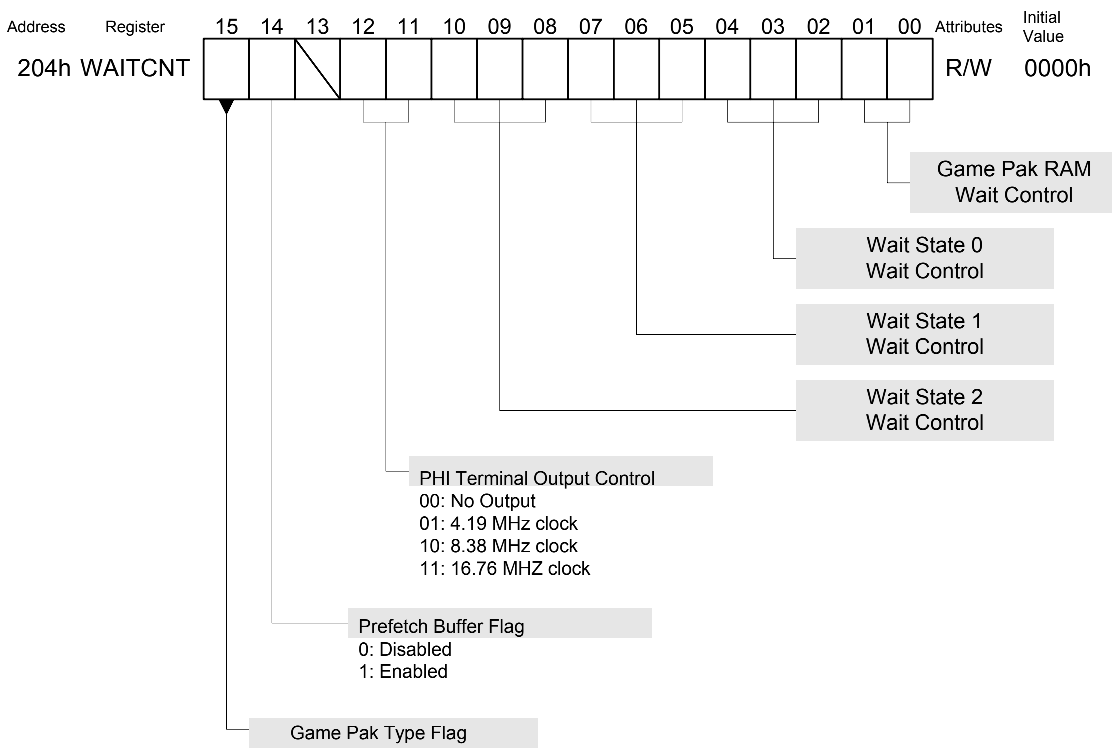
The area beginning from 0E000000h is the Game Pak RAM area. Up to 512 kilobits of SRAM or Flash Memory can be stored here. However, it is an 8 bit data bus. Due to the specifications, any Game Pak device other than ROM must be accessed using Nintendo's library.

3.3 Game Pak Memory Wait Control

Although the 32 MB Game Pak memory space is mapped to the area from 08000000h onward, the 32 MB spaces beginning from 0A000000h and 0C000000h are images of the 32 MB space that starts at 08000000h.

These images enable memory to be used according to the access speed of the Game Pak memory (1-4 wait cycles).

Figure 5 - The WAITCNT Register



WAITCNT [d15] Game Pak Type Flag

The System ROM uses this.

WAITCNT [d14] Prefetch Buffer Flag

When the Prefetch Buffer Flag is enabled and there is some free space, the Prefetch Buffer takes control of the Game Pak Bus during the time when the CPU is not using it, and reads Game Pak ROM data repeatedly. When the CPU tries to read instructions from the Game Pak and if it hits the Prefetch Buffer, the fetch is completed with no wait in respect to the CPU. If there is no hit, the fetch is done from the Game Pak ROM and there is a wait based on the set wait state.

If the Prefetch Buffer Flag is disabled, the fetch is done from the Game Pak ROM. There is a wait based on the wait state associated with the fetch instruction to the Game Pak ROM in respect to the CPU.

WAITCNT [d12 - 11] PHI Terminal Output Control

Controls the output from the PHI terminal. This should always be set to 00(No Output).

WAITCNT [d10 - 08],[d07 - 05],[d04 - 02] Wait State Wait Control

Individual wait cycles for each of the three areas (Wait States 0-2) that occur in Game Pak ROM can be set. The relation between the wait control settings and wait cycles is as follows. Use the appropriate settings for the device you are using.

Table 2 - Game Pak Memory Wait Control Values

Wait Control Value	Wait Cycles			
	1st Access	2nd Access		
		Wait State 0	Wait State 1	Wait State 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
101	3	1	1	1
110	2	1	1	1
111	8	1	1	1

After executing the System ROM (when the User Program is started) the Wait Control Value is 000. In the Game Pak Mask ROM used with the actual manufactured product the specifications are 1st Access/3 Wait, 2nd Access/1 Wait. In this case, set the Wait Control Value to 101.

WAITCNT [d01 - 00] Game Pak RAM Wait Control

Wait cycles for the Game Pak RAM can be set. The relation between the wait control settings and wait cycles is as follows. Use the appropriate settings for the device you are using.

Table 3 - Wait Control Values and Wait Cycles

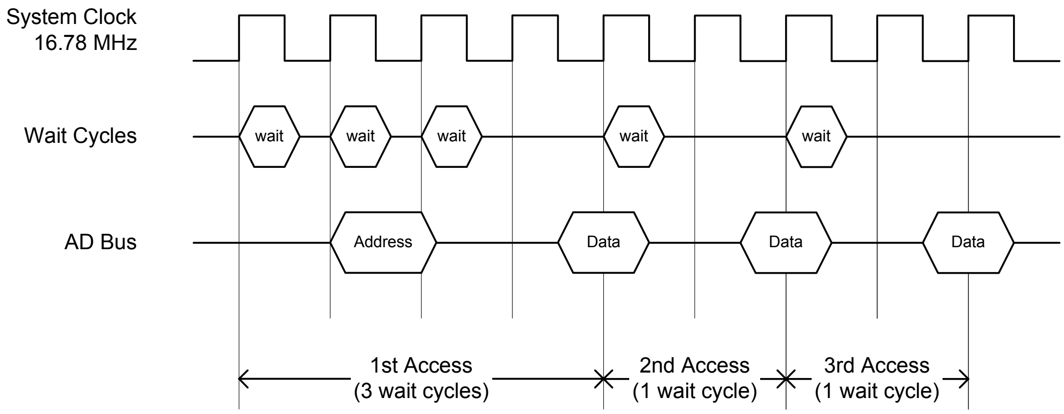
Wait Control Value	Wait Cycles
00	4
01	3
10	2
11	8

3.3.1 Access Timing

The following timing charts illustrate Game Pak ROM access with 3 wait cycles on the first access and 1 wait cycle on the second.

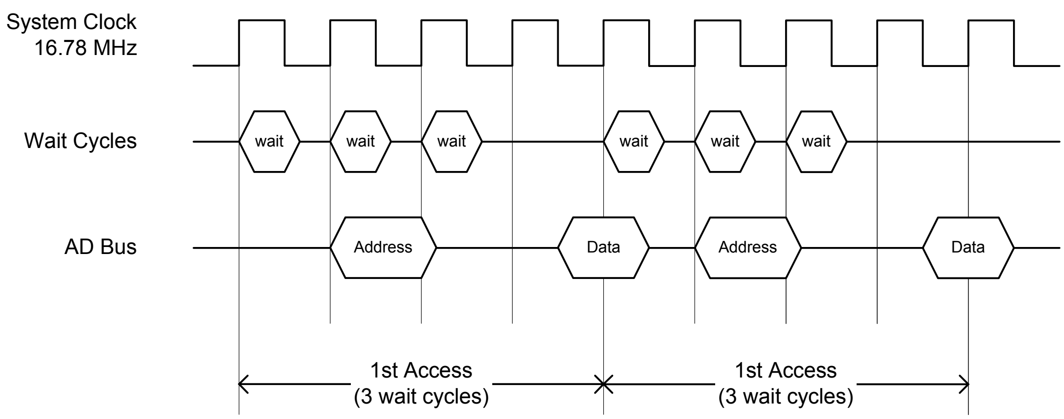
3.3.1.1 Sequential Access

Figure 6 - Sequential Access Timing Chart



3.3.1.2 Random Access

Figure 7 - Random Access Timing Chart



3.3.2 Game Pak Bus

The Game Pak bus has a total of 32 terminals, which are described in the following table.

Table 4 - Game Pak Bus Terminals

No.	Game Pak ROM Access		Game Pak RAM Access	
	Terminal	Use	Terminal	Use
1	VDD (3.3V)		VDD (3.3V)	
2	PHI		PHI	
3	/WR	Write Flag	/WR	Write Flag
4	/RD	Read Flag	/RD	Read Flag
5	/CS	ROM Chip Selection	/CS	ROM Chip Selection
6	AD0	Terminals used for both address (lower) and data	A0	Address
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	

Table 4 - Game Pak Bus Terminals (Continued)

No.	Game Pak ROM Access		Game Pak RAM Access	
	Terminal	Use	Terminal	Use
22	A16	Address (upper)	D0	Data
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 Chip Selection
31	/IREQ and DREQ	Terminal used for IREQ and DREQ	/IREQ and DREQ	Terminal used for IREQ and DREQ
32	GND		GND	

4 LCD

The Game Boy Advance uses a 2.9-inch-wide reflective TFT color LCD screen.

The vertical blanking interval of Game Boy Advance is longer than that of DMG and CGB, and its horizontal blanking interval is fixed.

Figure 8 - Display Screen Horizontal and Vertical Blanking Intervals

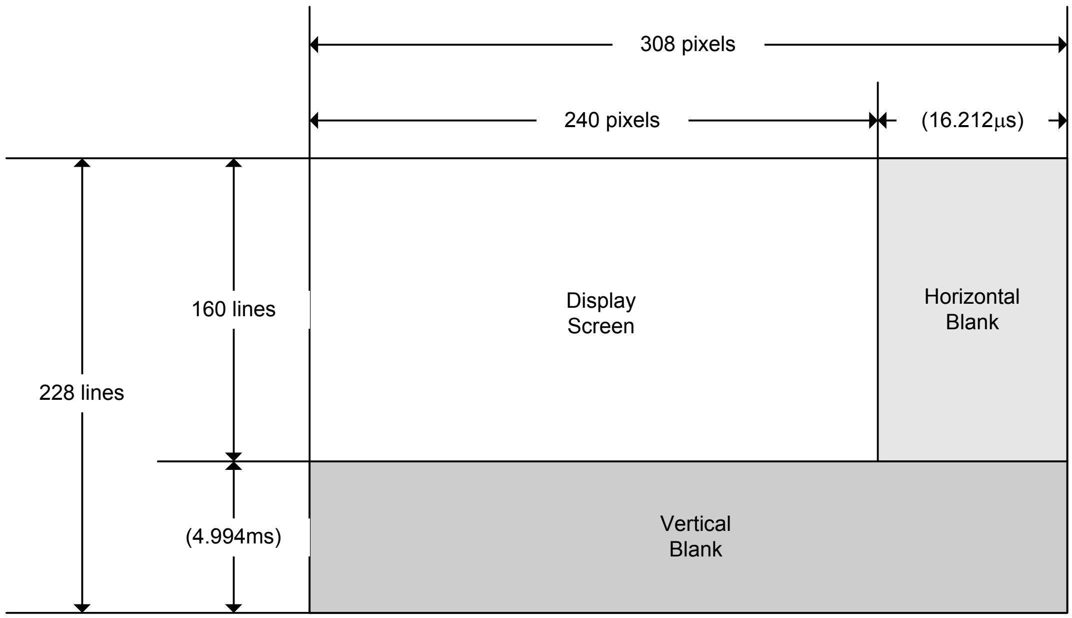


Table 5 - Game Boy Advance Display Screen Features

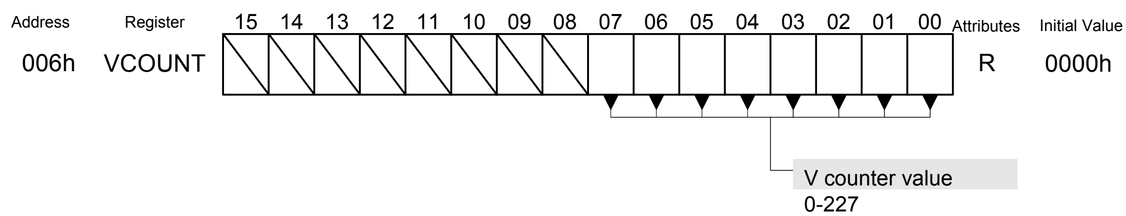
Item		Value	Interval
Display screen size	Number of pixels per horizontal line	240 pixels	57.221 µs
	Number of horizontal lines	160 lines	11.749 ms
Total number of pixels	Number of pixels per horizontal line	308 pixels	73.433 µs
	Number of horizontal lines	228 lines	16.743 ms
Blanking	Number of pixels per horizontal blank	68 pixels	16.212 µs
	Number of horizontal lines per vertical blank	68 lines	4.994 ms
Scanning cycle	H interval frequency	13.618 KHz	73.433 µs
	V interval frequency	59.727 Hz	16.743 ms

4.1 LCD Status

4.1.1 V Counter

The VCOUNT register can be used to read which of the total of 228 LCD lines (see "Figure 8 - Display Screen Horizontal and Vertical Blanking Intervals" on page 15) is currently being rendered.

Figure 9 - The VCOUNT Register



A value of 0-227 is read.

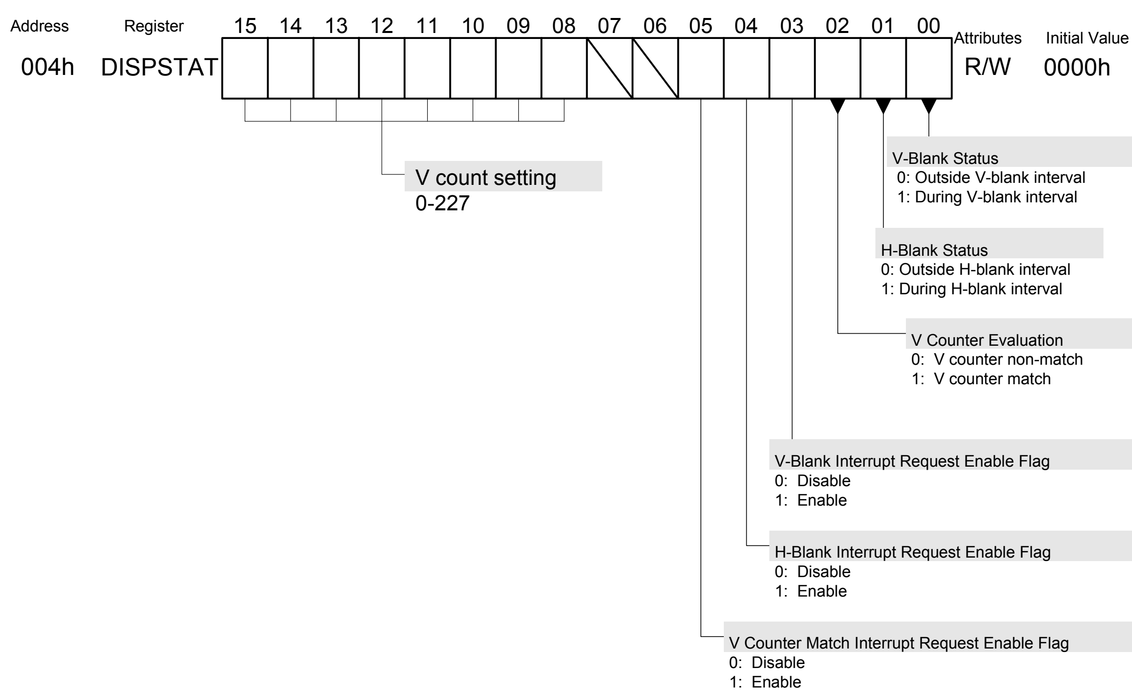
A value of 0-159 indicates that rendering is in progress; a value of 160-227 indicates a vertical blanking interval.

4.1.2 General LCD Status

General LCD status information can be read from bits 0-5 of the DISPSTAT register.

In addition, 3 types of interrupt requests can be generated by the LCD controller.

Figure 10 - The DISPSTAT Register



DISPSTAT [d15-08] V Count Setting

Can be used to set the value used for V counter evaluation and V counter match interrupts. The range for this setting is 0-227.

DISPSTAT [d05] V Counter Match Interrupt Request Enable Flag

Allows an interrupt request to be generated when the value of the V counter setting and the value of the line actually rendered (VCOUNT register value) agree.

DISPSTAT [d04] H-Blank Interrupt Request Enable Flag

Allows an interrupt request to be generated during horizontal blanking.

DISPSTAT [d03] V-Blank Interrupt Request Enable Flag

Allows an interrupt request to be generated during vertical blanking.

DISPSTAT [d02] V Counter Evaluation

Flag indicating whether the V count setting and the V count register value match. It is set while they match and automatically reset when they no longer match.

DISPSTAT [d01] H-Blank Status

Can check whether a horizontal blanking interval is currently in effect.

DISPSTAT [d00] V-Blank Status

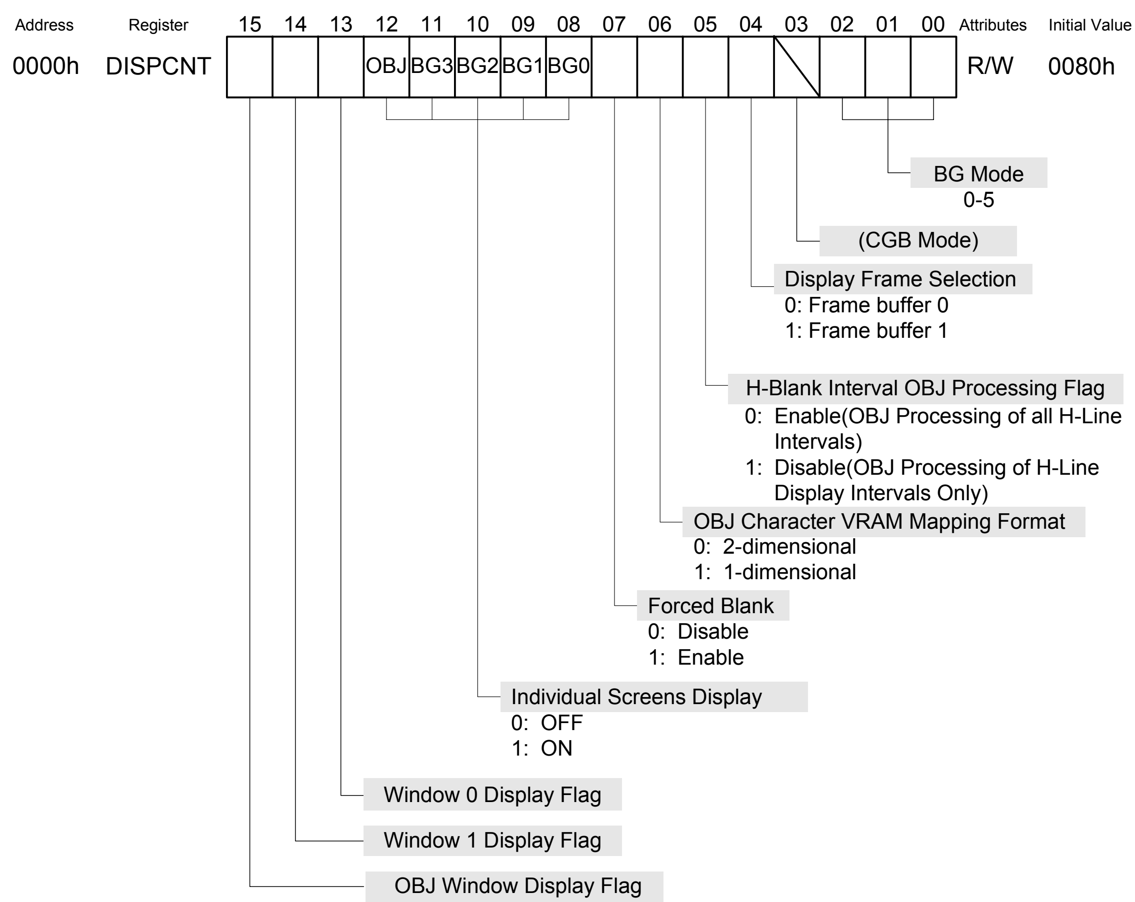
Can check whether a vertical blanking interval is currently in effect.

5 Image System

Game Boy Advance can use different image systems depending on the purpose of the software.

These display-related items are changed mainly using the DISPCNT register.

Figure 11 - The DISPCNT Register



DISPCNT [d15] OBJ Window Display Flag

Master flag that controls whether the OBJ window is displayed.

For information on the OBJ window, see "6.3 OBJ (Object)" on page 47.

DISPCNT [d14][d13] Display Flags for Windows 0 and 1

Master flag that controls whether windows 0 and 1 are displayed.

For information on windows, see "8 Window Feature" on page 65.

DISPCNT [d12-08] Individual Screens Display Flag

Allows individual control of whether BG0, BG1, BG2, BG3, and OBJ, respectively, are displayed. When settings are changed during the display period, the screen is hidden immediately if switching from ON to OFF, but the screen is displayed only after displaying three vertical lines when switching from OFF to ON.

DISPCNT [d07] Forced Blank

Setting this bit causes the CPU to forcibly halt operation of the image processing circuit, allowing access to VRAM, color palette RAM, OAM, and the internal registers. The LCD screen displays white during a forced blank. However, the internal HV synchronous counter continues to operate even during a forced blank. When the internal HV synchronous counter modifies the forced blank setting during a display period, the forced blank occurs immediately when switching from ON to OFF, but the forced blank only occurs after displaying three vertical lines when switching from OFF to ON.

DISPCNT [d06] OBJ Character VRAM Mapping Format

Specifies the VRAM mapping format for an OBJ character.

A setting of 0 causes the OBJ character to be handled in memory mapped 2-dimensional. A setting of 1 causes the OBJ character to be handled in memory mapped 1-dimensional.

For information on OBJ character VRAM mapping formats, see "6.3.2 Character Data Mapping" on page 48.

DISPCNT [d05] H-Blank Interval OBJ Processing Flag

A setting of 0 executes OBJ Render Processing with all H-Line intervals (including H-Blank intervals).

A setting of 1 executes OBJ Render Processing with the display intervals only and not for H-Blank intervals. Thus, when the user accesses OAM or OBJ VRAM during an H-Blank interval, this bit needs to be set. However, also in this situation, maximum OBJ display performance cannot be obtained.

DISPCNT [d04] Display Frame Selection

When rendering in bitmap format in a mode in which there are 2 frame buffers (BG modes 4 and 5), this bit allows selection of one of the frame buffers for rendering. A setting of 0 selects the contents of frame buffer 0 for rendering; a setting of 1 selects the contents of frame buffer 1 for rendering.

DISPCNT [d03] (CGB Mode)

Game Boy Advance is equipped with two CPUs. In Game Boy Advance mode, a 32-bit RISC CPU starts, and in CGB mode, an 8-bit CISC CPU starts.

Note: Because this bit is controlled by the system, it cannot be accessed by the user.

DISPCNT [d02-00] BG Mode

Selects the BG mode from a range of 0-5.

For more information on BG modes, see "5.1 BG Modes" on page 21.

5.1 BG Modes

5.1.1 Details of BG Modes

In Game Boy Advance, changing the BG mode allows character format and bitmap format to be used selectively, as appropriate.

In modes 0, 1, and 2, rendering to the LCD screen is performed in a character format suitable for the game.

In modes 3, 4, and 5, rendering to the LCD screen is performed in bitmap format.

Table 6 - Background Mode Details (Character Format BG Screen)

BG Mode	Character Format BG Screen			Number of Characters Specifiable	Number of Colors/ Palettes	Features					
	Rotation /Scaling	No. of Screens	Size			1	2	3	4	5	6
0	No	4	256 x 256 to 512 x 512	1024	16 / 16 256 / 1	O	O	O	O	O	O
1	No	2	256 x 256 to 512 x 512	1024	16 / 16 256 / 1	O	O	O	O	O	O
	Yes	1	128 x 128 to 1024 x 1024	256	256 / 1	O	X	O	O	O	O
2	Yes	2	128 x 128 to 1024 x 1024	256	256 / 1	O	X	O	O	O	O

Table 7 - Background Mode Details (Bitmap Format BG Screen)

BG Mode	Bitmap Format BG Screen			Frame Memory	No. of Colors	Features					
	Rotation /Scaling	No. of Screens	Size			1	2	3	4	5	6
3	Yes	1	240 x 160	1	32,768	O	X	O	O	O	O
4	Yes	1	240 x 160	2	256	O	X	O	O	O	O
5	Yes	1	160 x 128	2	32,768	O	X	O	O	O	O

Features:
*1 HV Scroll (individual screens)
*2 HV Flip (individual characters)
*3 Mosaic (16 levels)
*4 Semitransparent(16 levels)
*5 Fade-in/Fade-out
*6 Screen priority specification (2 bits)

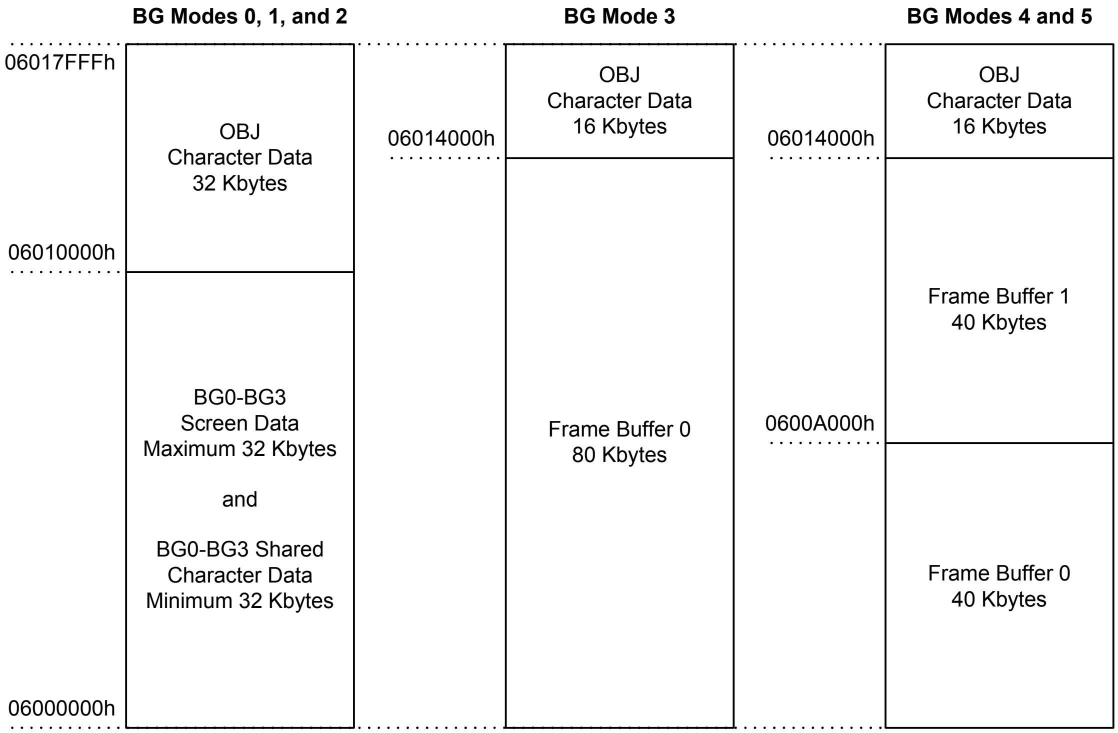
Note: In mode 3, one frame memory is available that can display 32,768 colors, which is suitable for rendering still images. Modes 4 and 5 allow double buffering using two frame memories, and are thus suitable for rendering animated video.

The method of controlling text BG scrolling is different from that of BG rotation/scaling and bitmap BG scrolling. (See "6.1.8 BG Scrolling" on page 42 and "6.1.7 BG Rotation and Scaling Features" on page 40.)

5.1.2 VRAM Memory Map

The VRAM (96 KB) memory maps in the BG modes are as shown in the following figure.

Figure 12 - Background Mode Memory Maps



Users can map the screen and character data areas in the 64 KB BG area in BG modes 0, 1, and 2. For more information, see "6.1.3 VRAM Address Mapping of BG Data" on page 30.

In addition, see the descriptions in subsequent sections of this document for more information on the memory areas and the data formats for each area.

6 Rendering Functions

The Game Boy Advance CPU has 96 KB of built-in VRAM.

Its rendering functions include BG and OBJ display capability. The method used for BG rendering varies with the BG mode, as described below.

6.1 Character Mode BG (BG Modes 0-2)

In character mode, the components of the BG screen are basic characters of 8 x 8 pixels.

6.1.1 BG Control

There are 4 BG control registers, corresponding to the maximum number of BG screens (registers BG0CNT, BG1CNT, BG2CNT, and BG3CNT).

Registers BG0CNT and BG1CNT are exclusively for text BG control, while BG2CNT and BG3CNT also support BG rotation and scaling control.

The registers used by the BG modes are as follows.

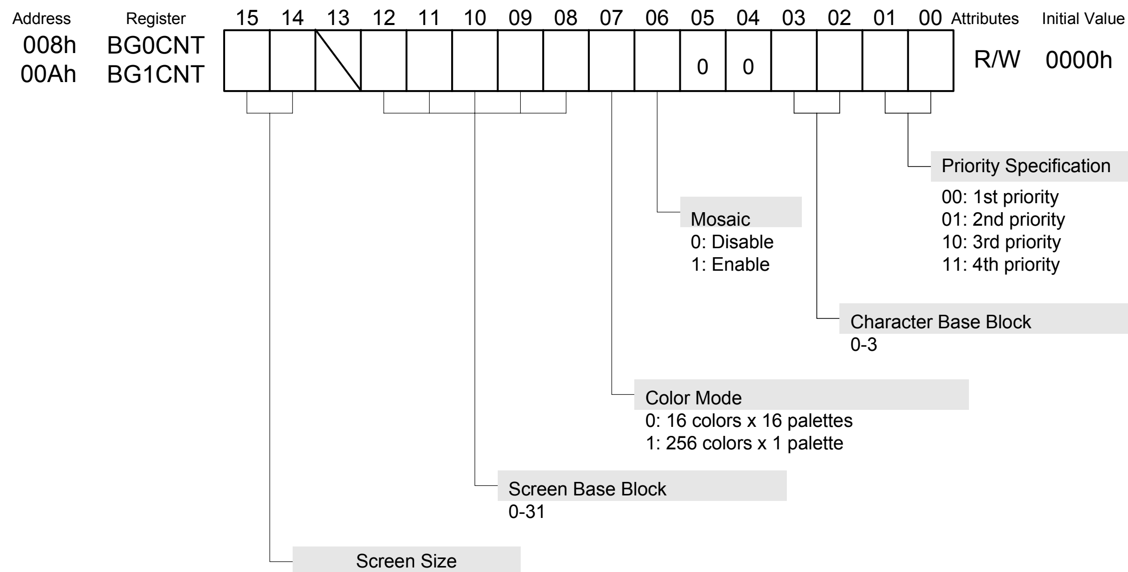
Table 8 - Background Mode Registers

BG Mode	BG Control Register			
	BG0CNT	BG1CNT	BG2CNT	BG3CNT
0	BG0 (text)	BG1 (text)	BG2 (text)	BG3 (text)
1	BG0 (text)	BG1 (text)	BG2 (rotation/scaling)	
2			BG2 (rotation/scaling)	BG3 (rotation/scaling)

The contents of the BG control registers are shown below.

6.1.1.1 Text BG Screen Control (BG0, BG1)

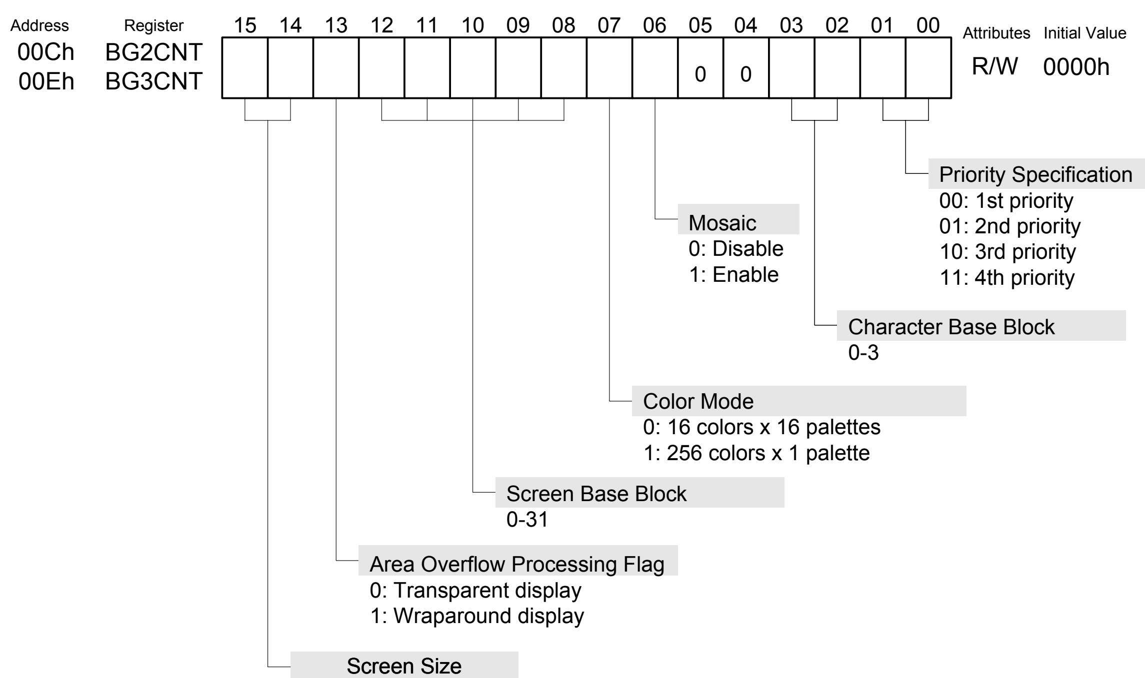
Figure 13 - Background Screen Control Registers



6.1.1.2 Text BG and Rotation/Scaling BG Screen Control (BG2 and BG3)

Whether the screen is a text screen or a scaling/rotation screen varies with the BG mode.

Figure 14 - Text Background and Rotation/Scaling Background Screen Control Registers



BG*CNT [d15-14] Screen Size

Allows the screen size for the BG as a whole to be specified.

When a value other than the maximum is specified, the remaining VRAM area can be used as a character data area.

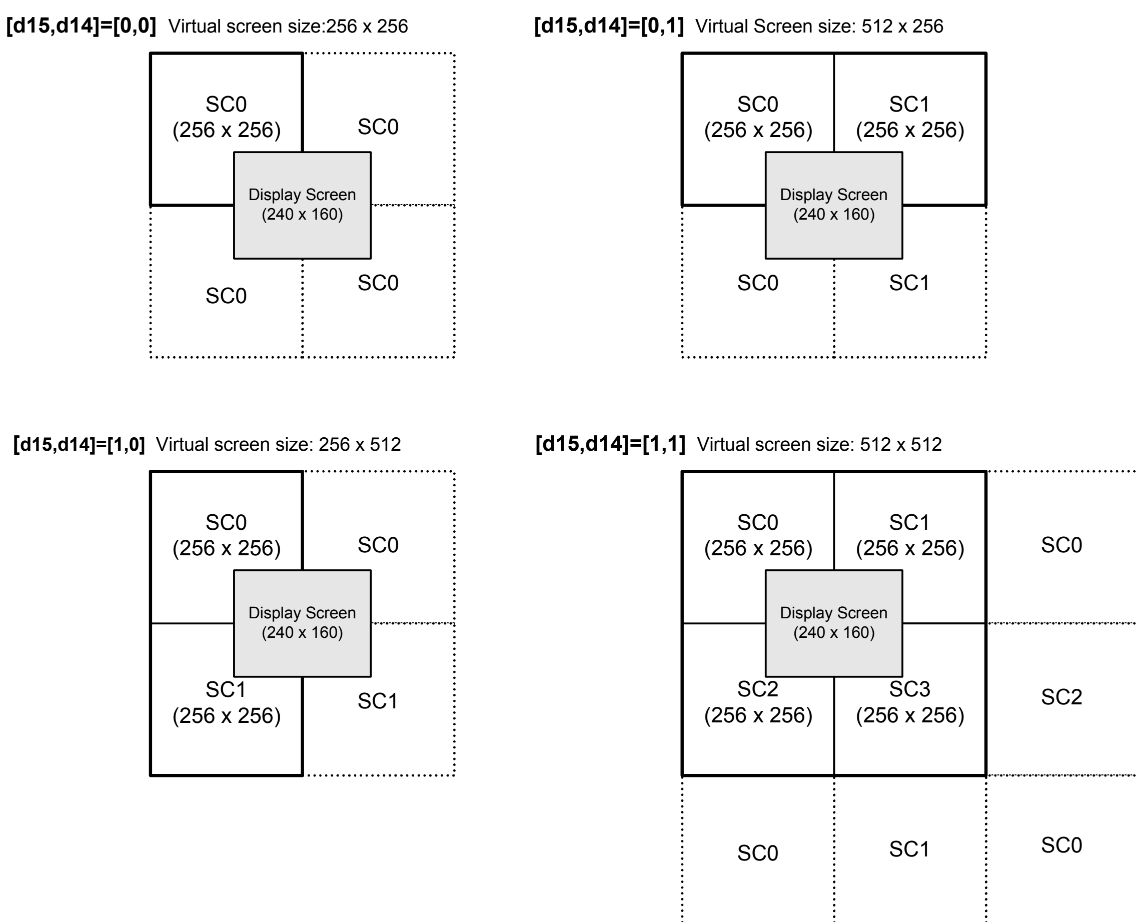
Refer to the table below and the VRAM Memory Map figure above.

Table 9 - Screen Size Settings

Screen Size Setting	Text Screen		Rotation/Scaling Screen	
	Screen Size	Screen Data	Screen Size	Screen Data
00	256 x 256	2 KB	128 x 128	256 Bytes
01	512 x 256	4 KB	256 x 256	1 KB
10	256 x 512	4 KB	512 x 512	4 KB
11	512 x 512	8 KB	1024 x 1024	16 KB

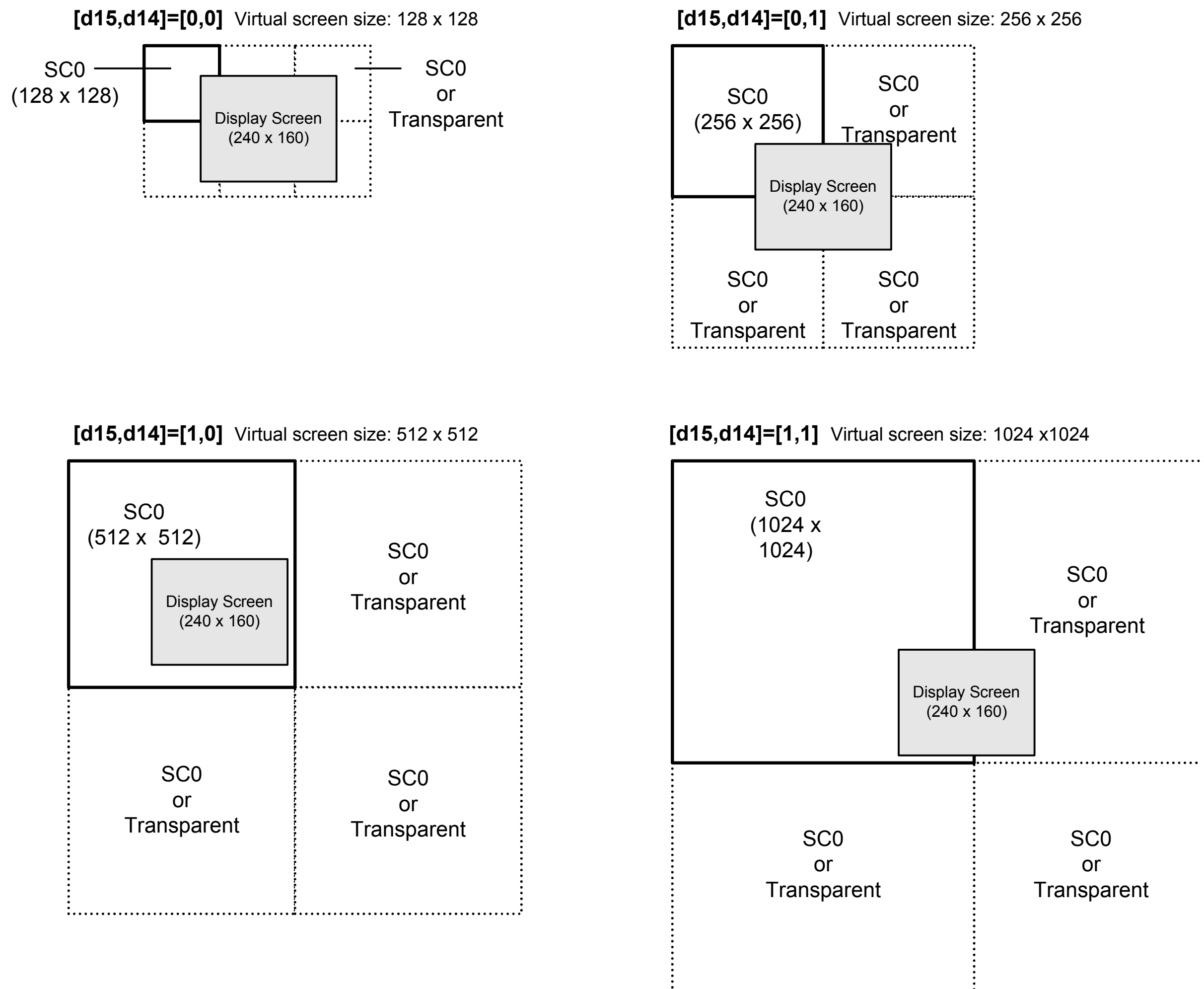
(1) Overview of Screen Sizes for Text BG Screens

Figure 15 - Text Background Screen Sizes



(2) Illustration of Screen Sizes for Rotation/Scaling BG Screens

Figure 16 - Rotation/Scaling Background Screen Sizes



BG2CNT,BG3CNT [d13] Area Overflow Processing

When the display screen overflows the boundaries of the virtual screen due to a rotation/scaling operation, this bit can be used to choose whether the area of the screen into which the overflow occurs is displayed as transparent or wraps around the display screen.

For information on scaling, see "6.1.7 BG Rotation and Scaling Features" on page 40.

BG*CNT [d12-08] Screen Base Block Specification

Specifies the starting block in VRAM where screen data are stored.

(32 steps: 0-31; 2-KB increments).

See "6.1.3 VRAM Address Mapping of BG Data" on page 30.

BG*CNT [d07] Color Mode

Specifies whether to reference BG character data in 16 color x 16 palette format or 256 color x 1 palette format.

BG*CNT [d06] Mosaic

Turns mosaic processing for BG on and off.

BG*CNT [d03-02] Character Base Block Specification

Specifies the starting block in VRAM where the character data to be displayed in the BG is stored.

(4 steps: 0-3; 16-KB increments)

See "6.1.3 VRAM Address Mapping of BG Data" on page 30.

BG*CNT [d01-00] Priority Among BGs

With the default value (same priority value specified for all), the order of priority is BG0, BG1, BG2, and BG3. However, this order can be changed to any desired.

Values of 0 (highest priority) to 3 can be specified.

When the BG priority has been changed, care should be taken in specifying the pixels used for color special effects.

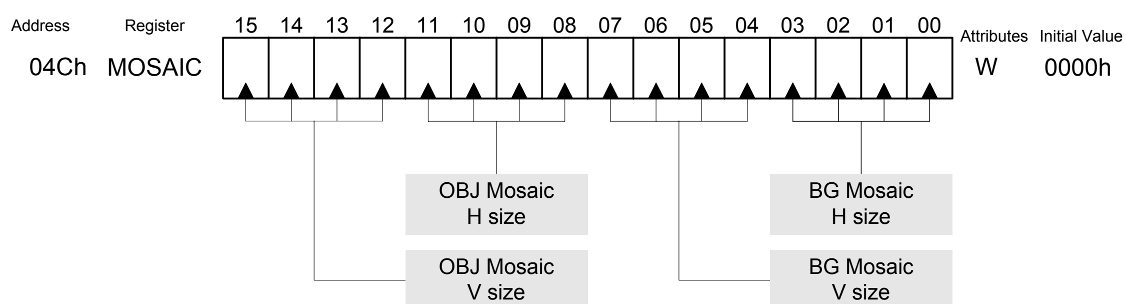
6.1.2 Mosaic Size

Mosaic size is set in the MOSAIC register.

Turning mosaic on/off for each BG is accomplished by the mosaic flag of the BG control register.

For information on the mosaic flag, see "6.1.1 BG Control" on page 25.

Figure 17 - The MOSAIC Register

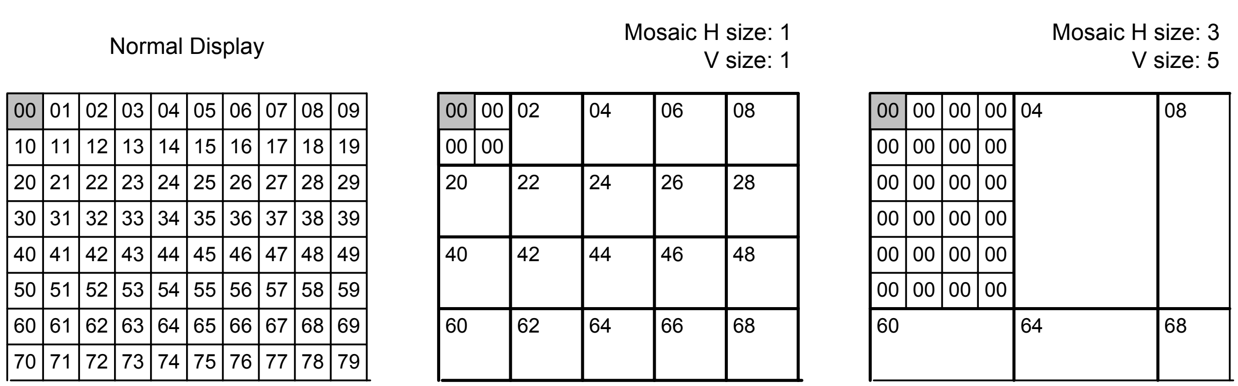


The mosaic value specifies how many pixels of a normal display should comprise each large pixel displayed.

Counting from the upper left-most pixel on the screen, the number of pixels equal to the mosaic size are used in the mosaic display. The other pixels are overwritten by the mosaic. Please refer to the figure below.

If the mosaic size value is 0, a normal display is seen even if mosaic is turned on.

Figure 18 - Mosaic Schematic



6.1.3 VRAM Address Mapping of BG Data

BG data (BG character and screen data) are stored in the 64-KB BG area of VRAM.

6.1.3.1 BG Character Data

The starting address for referencing BG character data can be specified using the character base block specification of the BG control register.

The amount of data depends on the number of character data items stored and the data format (color formats: 256 colors x 1 palette or 16 colors x 16 palettes).

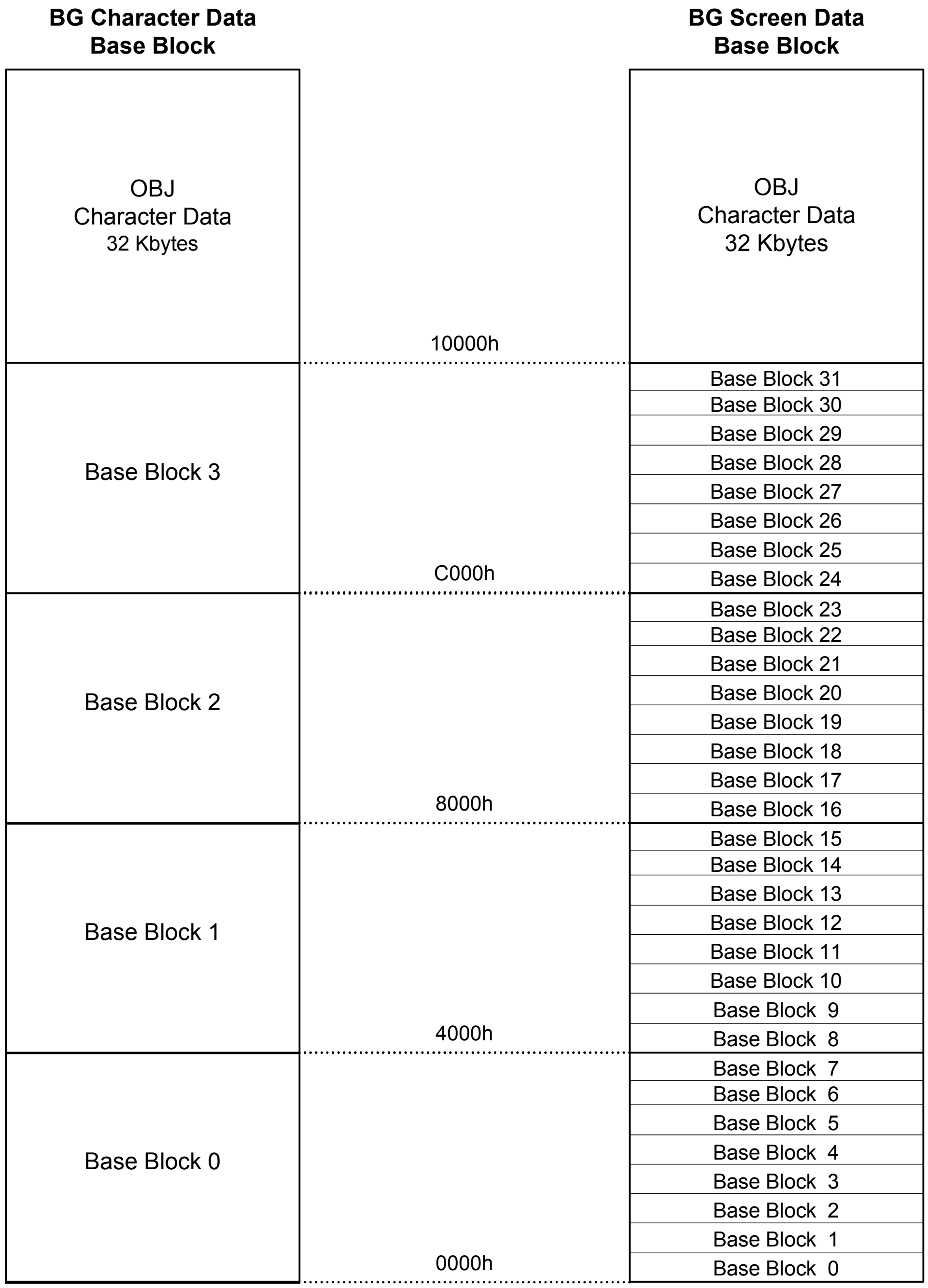
6.1.3.2 BG Screening Data

The starting address for referencing BG screen data can be set using the screen base block specification of the BG control register.

The amount of data depends on the type of BG screen (text or rotation/scaling) and the screen size. These can be set by the BG control register.

6.1.3.3 Illustration of VRAM Base Blocks for BG Data

Figure 19 - VRAM Base Blocks for Background Data



6.1.4 Character Data Format

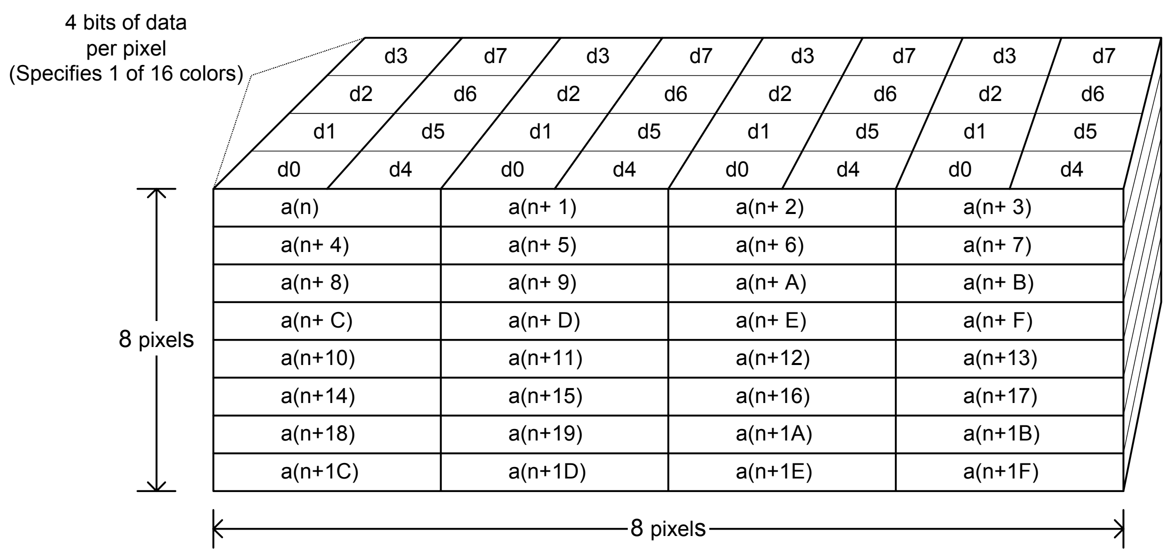
There are two formats for character pixel data, 16 color x 16 palettes and 256 colors x 1 palette. The same format is used for OBJ and BG.

The data are held in VRAM in the form indicated below.

6.1.4.1 16 Colors x 16 Palettes

There are 2 pixels per address. Thus, the amount of data for each basic character is 20H x 8 bits.

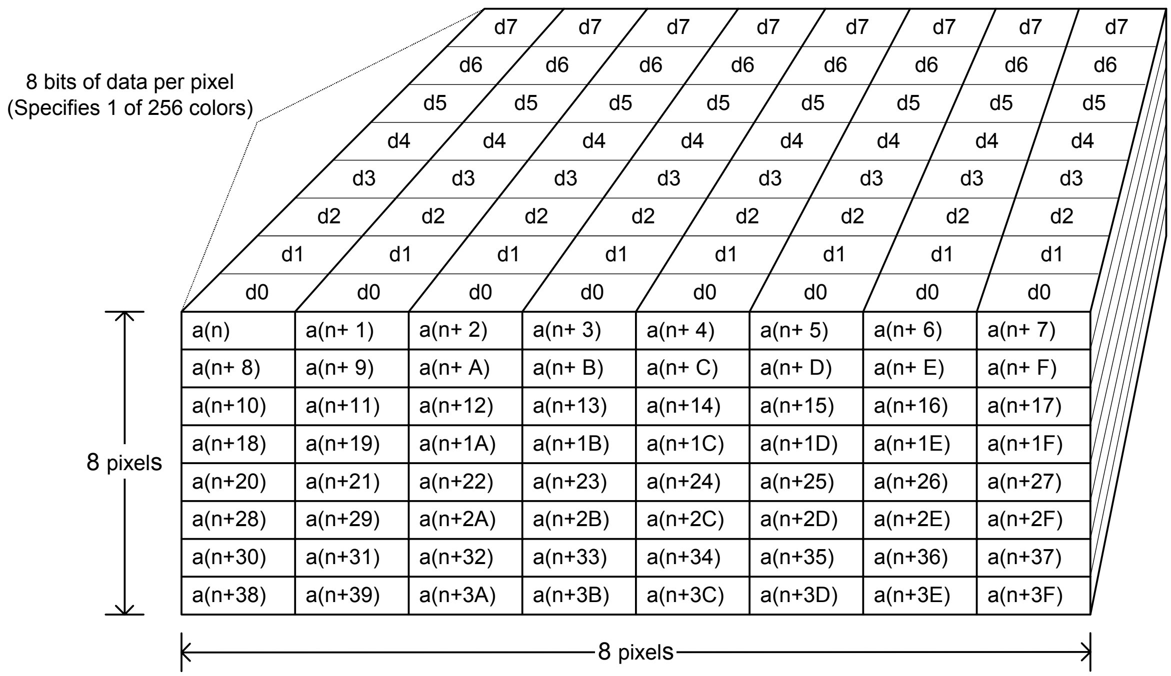
Figure 20 - 16-Color x 16-Palette Character Data Format



6.1.4.2 256 Colors x 1 Palette

There is 1 pixel specified per address. Thus, the amount of data for each basic character is 40H x 8 bits.

Figure 21 - 256-Color x 1-Palette Character Data Format



6.1.5 BG Screen Data Format

A BG screen is considered to be the 8 x 8 pixel unit that represents the size of the basic character, and the BG screen data specifies the characters that are arranged.

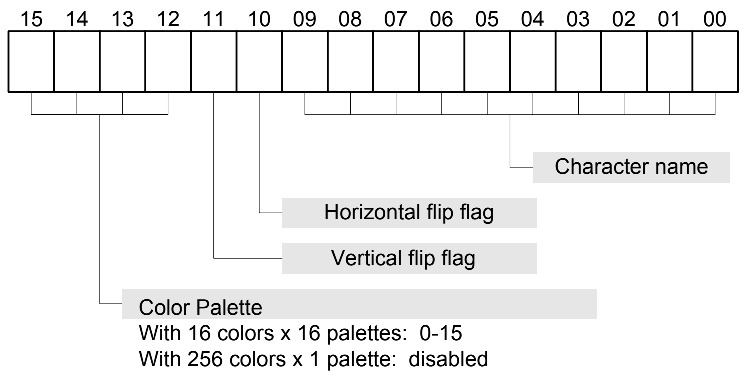
BG screen data should be stored, beginning from the starting address of the BG screen base block specified in the BG control register. The number of screen data items specified per BG depends on the screen size setting in the BG control register.

BG screen data for text and rotation/scaling screens are specified in the following formats.

6.1.5.1 Text BG Screen

A text BG screen consists of 2 bytes of screen data per basic character; 1,024 character types can be specified.

Figure 22 - Text Background Screen Format



[d15-12] Color Palette

If the color mode specification in the BG control register is 16 colors x 16 palettes, these bits specify palette 0-15 as the palette to be applied to the character.

This is disabled when the color mode specification is 256 x 1 palette.

[d11] Vertical Flip Flag

Enables the BG character to be flipped vertically.

A setting of 1 produces the vertical-flip display.

[d10] Horizontal Flip Flag

Enables the BG character to be flipped horizontally.

A setting of 1 produces the horizontal-flip display.

[d09-00] Character Name

Specify the number of the character that has character base block starting address specified in the BG control register as its starting point.

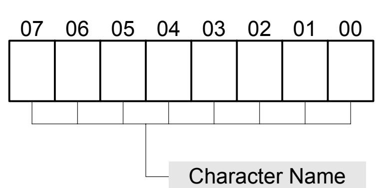
6.1.5.2 Rotation/Scaling BG Screen

The rotation/scaling BG screen consists of 1 byte of screen data per basic character; 256 character types can be specified.

The character data must be classified as 256 colors x 1 palette.

The color mode specification in the BG control register is disabled for a rotation/scaling screen.

Figure 23 - Rotation/Scaling Background Screen Format



Cautions for VRAM

Game Boy Advance provides a high degree of freedom in using the BG area of VRAM.

Consequently, in managing VRAM, the following points deserve particular attention.

1. There are 2 formats for BG character data (defined by 16 and 256 colors), and these can be used together.
2. The BG character data base block can be selected from among 4 blocks (BG control register).
3. The BG screen data base block can be selected from among 32 blocks (BG control register).
4. The screen size (amount of VRAM used) can be set for each BG (BG control register).
5. Text and rotation/scaling BGs can be present and used together in a BG screen.

In managing VRAM, particular care is required in BG mode 1, because text BG screens (which can handle BG character data in both 256 colors x 1 palette and 16 colors x 16 palettes) and rotation/scaling BG screens (which can handle only 256 colors x 1 palette) may be used together.

Therefore, the VRAM mapping status should be sufficiently understood when programming.

6.1.6 BG Screen Data Address Mapping for the LCD Screen

6.1.6.1 Text BG

Figure 24 - Virtual Screen Size of 256 x 256 Pixels (Text Background)

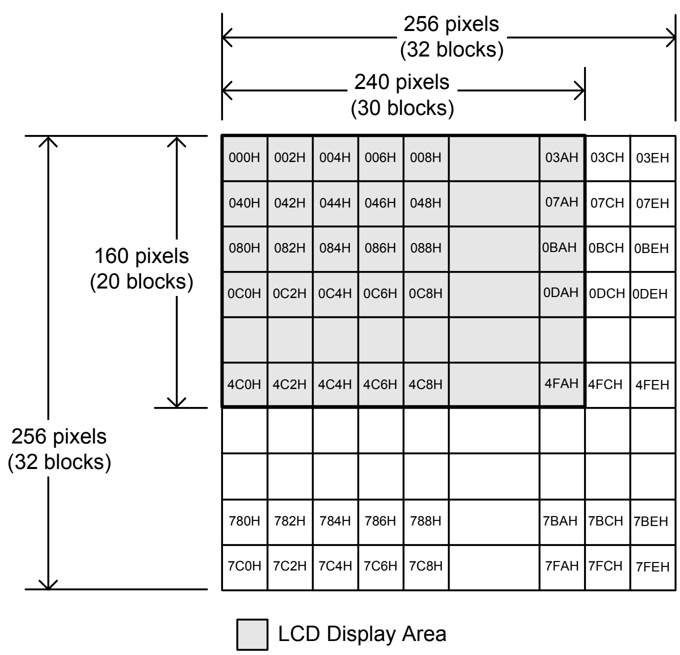


Figure 25 - Virtual Screen Size of 512 x 256 Pixels (Text Background)

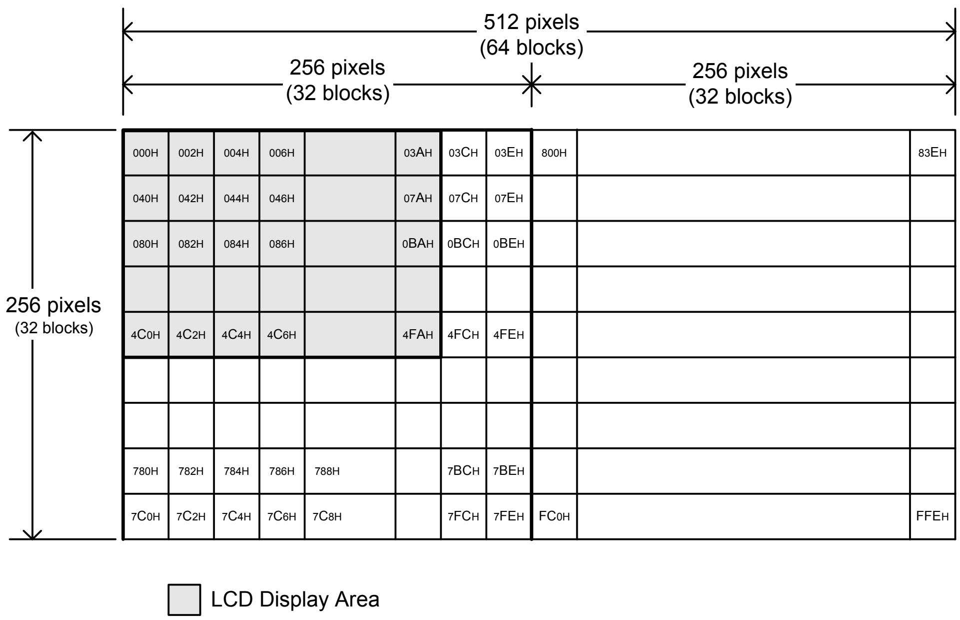


Figure 26 - Virtual Screen Size of 256 x 512 Pixels (Text Background)

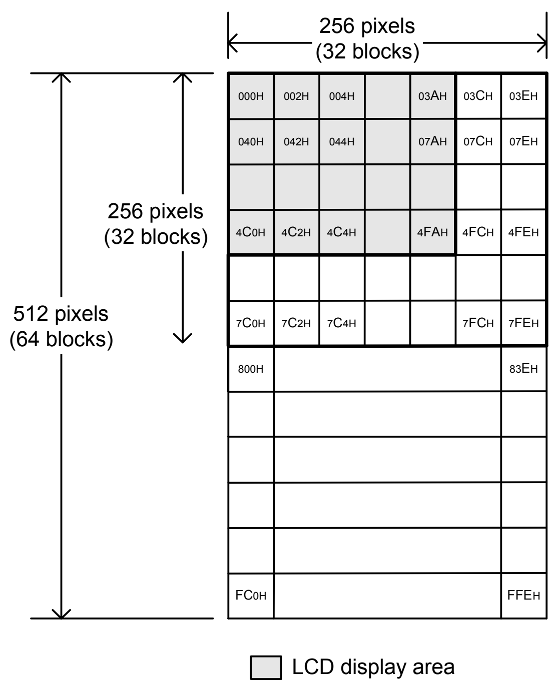
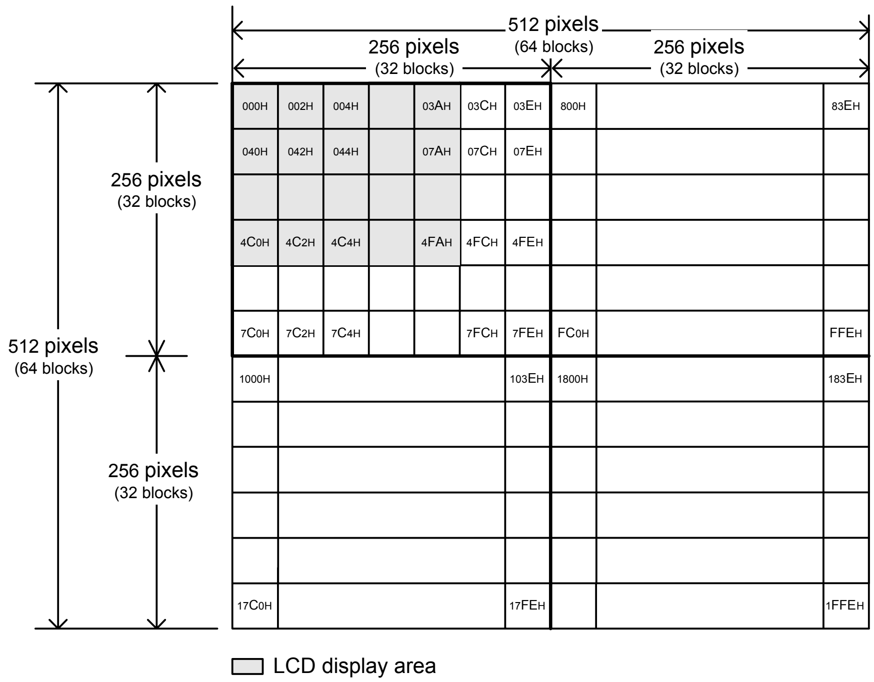


Figure 27 - Virtual Screen Size of 512 x 512 Pixels (Text Background)



6.1.6.2 Rotation/Scaling BG

Figure 28 - Virtual Screen Size of 128 x 128 Pixels (Rotation/Scaling Background)

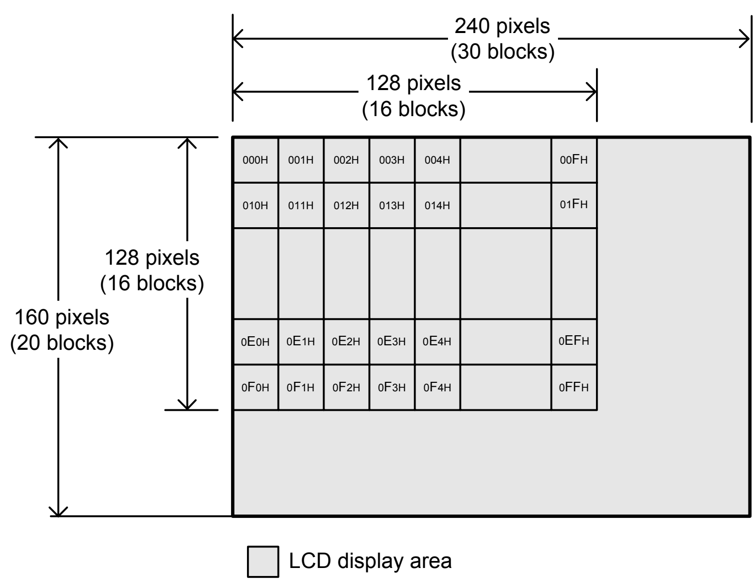


Figure 29 - Virtual Screen Size of 256 x 256 Pixels (Rotation/Scaling Background)

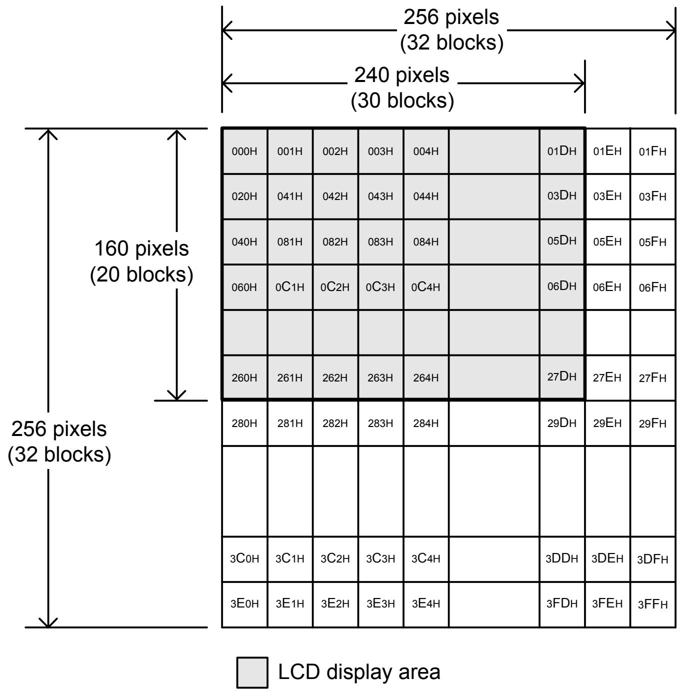


Figure 30 - Virtual Screen Size of 512 x 512 Pixels (Rotation/Scaling Background)

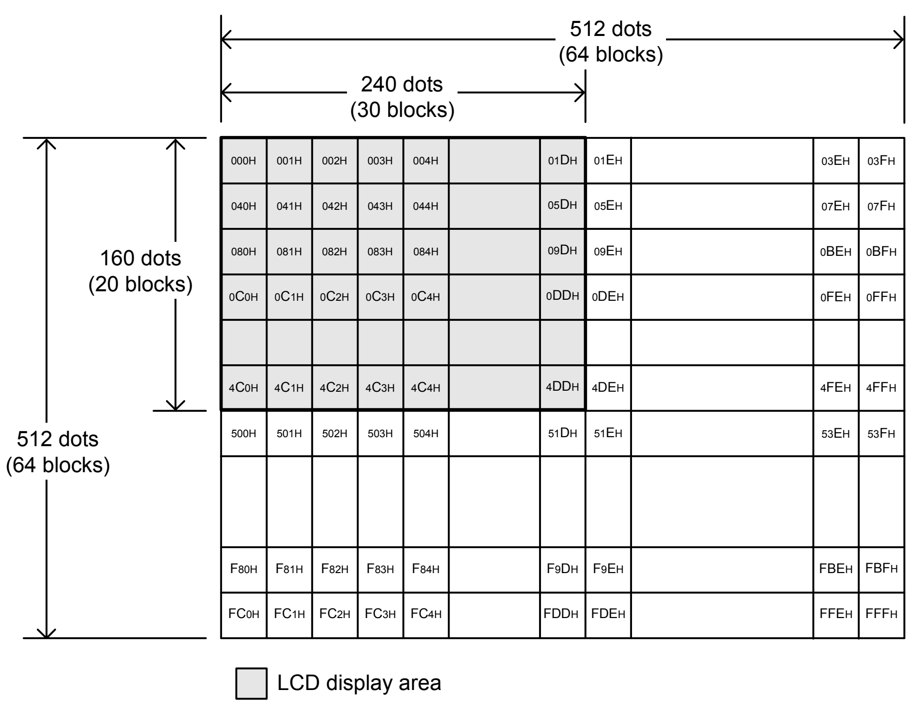
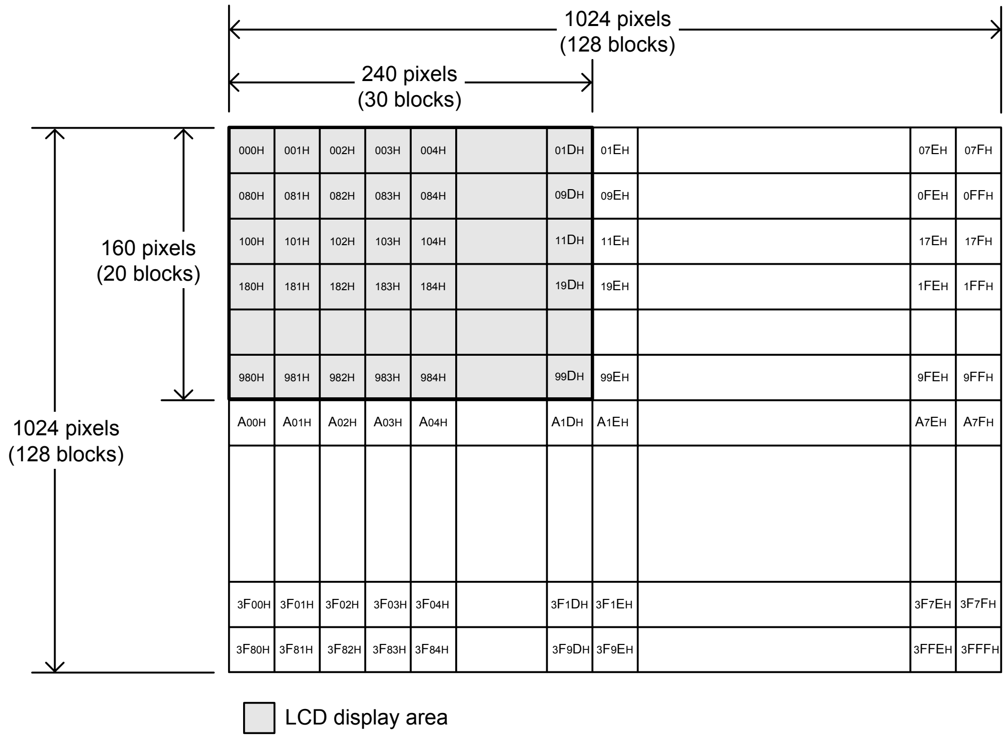


Figure 31 - Virtual Screen Size of 1024 x 1024 Pixels (Rotation/Scaling Background)

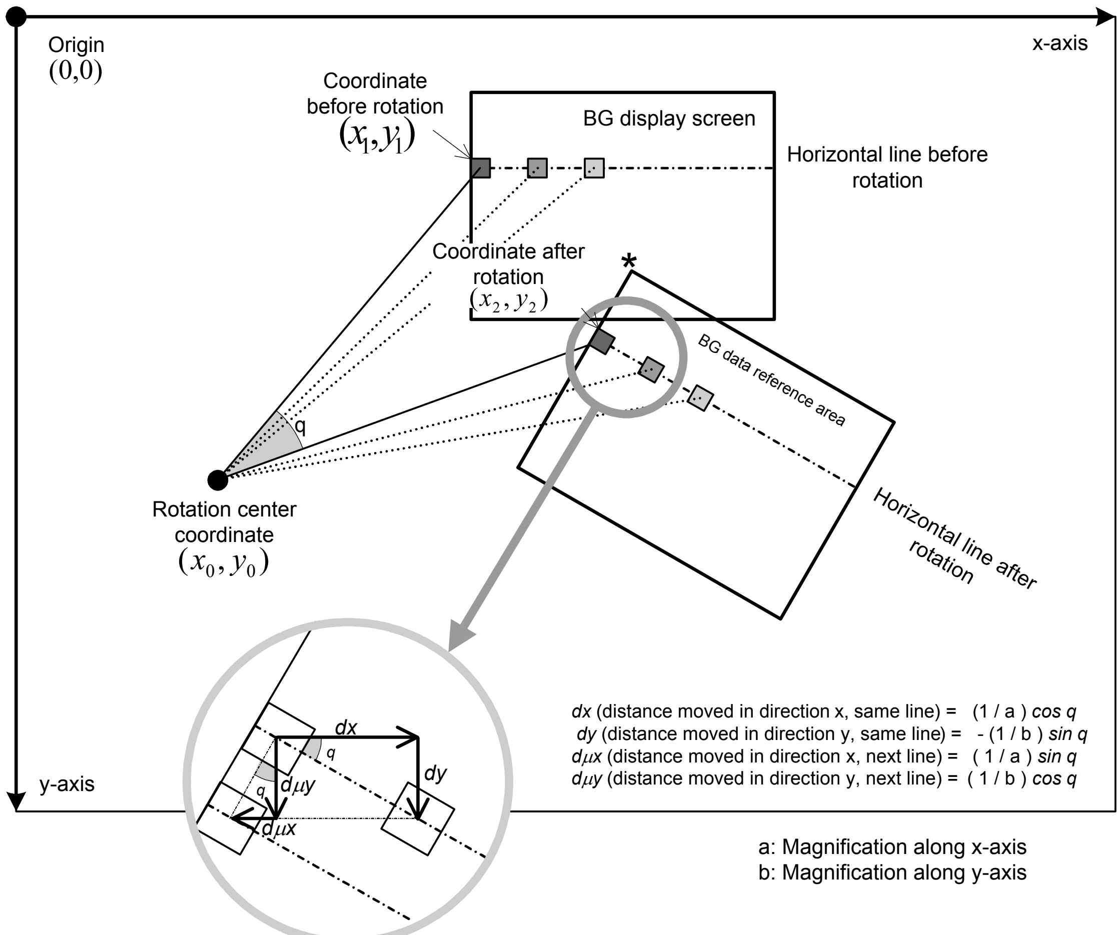


6.1.7 BG Rotation and Scaling Features

Rotation and scaling of the BG as a whole can be performed in a rotation/scaling BG screen.

With rotation, BG data is referenced as shown in the following figure.

Figure 32 - Referencing Rotated Background Data



BG rotation and scaling are implemented in Game Boy Advance using the following arithmetic expressions.

Equation 1 - Background Rotation and Scaling

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

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

$$x_2 = A(x_1 - x_0) + B(y_1 - y_0) + x_0$$
$$y_2 = C(x_1 - x_0) + D(y_1 - y_0) + y_0$$

Parameters used in rotation and scaling operations are specified for BG2 and BG3 in the following registers. Registers for Starting Point of BG Data Reference are also used when Scaling/Rotation BG and Bitmap Mode BG are offset displayed (scrolled). (There is also an offset register for Text BG.)

Figure 33 - Registers for Setting the Starting Point of BG Data

Address	Register	Bits 15–12	Bits 11–00	Attributes	Initial Value
028h 038h	BG2X_L BG3X_L	X-coordinate of reference starting point (rotation/scaling results)		W	0000h
02Ah 03Ah	BG2X_H BG3X_H	(unused)	X-coordinate of reference starting point (rotation/scaling results)	W	0000h
02Ch 03Ch	BG2Y_L BG3Y_L	Y-coordinate of reference starting point (rotation/scaling results)		W	0000h
02Eh 03Eh	BG2Y_H BG3Y_H	(unused)	Y-coordinate of reference starting point (rotation/scaling results)	W	0000h

Figure 34 - Registers for Setting the Direction Parameters of BG Data

Address	Register	Bits 15–00	Attributes	Initial Value
020h 030h	BG2PA BG3PA	dx: distance of movement in x direction along same line	W	0100h
022h 032h	BG2PB BG3PB	dμx: distance of movement in x direction along next line	W	0000h
024h 034h	BG2PC BG3PC	dy: distance of movement in y direction along same line	W	0000h
026h 036h	BG2PD BG3PD	dμy: distance of movement in y direction along next line	W	0100h

6.1.7.1 Operations Used in BG Rotation/Scaling Processing

1. Using software, the user determines the results of the rotation/scaling operation for the left-upper coordinate of the display screen and sets this as the starting point of the BG data reference in registers BG2X_L, BG2X_H, BG2Y_L, BG2Y_H, BG3X_L, BG3X_H, BG3Y_L, and BG3Y_H. The set value is a signed fixed-point number (8 bits for fractional portion, 19 bits for integer portion, and 1 bit for sign, for a total of 28 bits).

   The BG data reference direction is set in BG2PA, BG2PB, BG2PC, BG2PD, BG3PA, BG3PB, BG3PC, and BG3PD. The set value is a signed fixed-point number (8 bits for fractional portion, 7 bits for integer portion, and 1 bit for sign, for a total of 16 bits).

2. The image processing circuit sums the increases in the x direction (dx, dy) in relation to the BG data reference starting point set in the above registers, and calculates the x-coordinate.

3. When the line is advanced, the increases in the y direction (dμx, dμy) are summed in relation to the reference starting point, and the coordinate of the rendering starting point for the next line is calculated. The processing in step 2) is then performed.

4. However, if a register for the BG data reference starting point is rewritten during an H-blanking interval, the y-direction summation for that register is not calculated. The CPU uses this mode to change the center coordinate and the rotation/scaling parameters for each line.

6.1.7.2 Area Overflow Processing

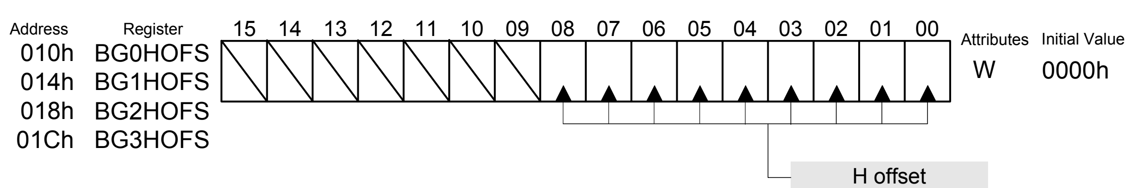
When the display screen overflows the boundaries of the virtual screen due to a rotation/scaling operation, this BG control register can be used to select whether the area of the screen into which the overflow occurs is transparent or wraps around the display screen.

For information on BG control, see "6.1.1 BG Control" on page 25.

6.1.8 BG Scrolling

For each text BG screen, the offset on the display screen can be specified in 1-pixel increments. Offset register is only valid for Text BG. In order to offset display Scaling/Rotation BG and Bitmap Mode BG set the BG Reference Starting Point. See "6.1.7 BG Rotation and Scaling Features" on page 40.

Figure 35 - Offset Settings Registers



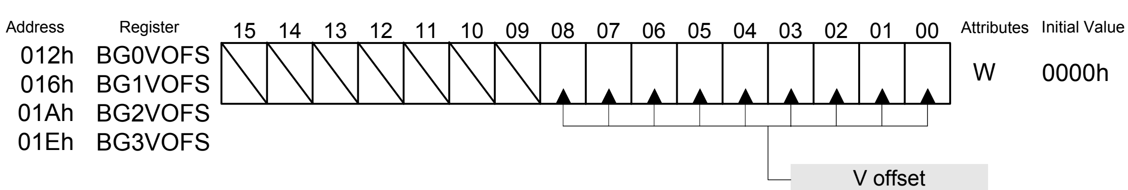
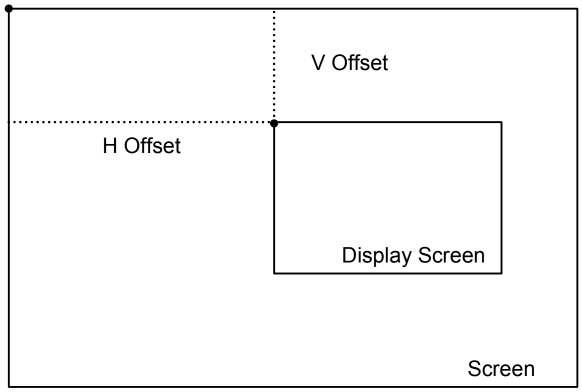


Figure 36 - Offset Illustration



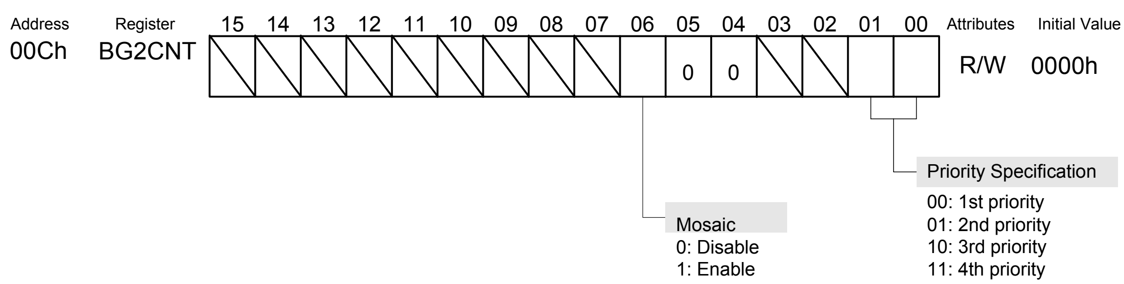
6.2 Bitmap Mode BGs (BG Modes 3-5)

In the bitmap modes, the components of the BG screen are handled in pixel units, and the contents of VRAM (frame buffer) are displayed as color data for each pixel on the screen.

6.2.1 BG Control

The bitmap BG will be treated as BG2. Therefore, in order to display the content of the frame buffer on the LCD screen, you need to set the BG2 display flag to ON in the DISPCNT Register. For BG Control the BG2CNT Register is used.

Figure 37 - The BG2CNT Register



BG2CNT [d06] Mosaic

> This controls the ON/OFF of mosaic processing for BG2. When ON, the settings for the Mosaic Size Register, MOSAIC, are referenced. For information on Mosaic, see "6.1.2 Mosaic Size" on page 29.

BG2CNT [d01-00] Priority Among BGs

> Due to the fact that in Bitmap Mode there is only one BG plane (other than the backdrop plane), there is no priority relationship among BGs, but you can set up priorities with OBJ. For information on this, see "6.4 Display Priority of OBJ and BG" on page 58.

6.2.2 BG Rotation/Scaling

The parameters for Bitmap BG Rotation/Scaling use BG2 related registers(BG2X_L, BG2X_H, BG2Y_L, BG2Y_H, BG2PA, BG2PB, BG2PC, and BG2PD).

For information on rotation/scaling parameters, see "6.1.7 BG Rotation and Scaling Features" on page 40.

With Bitmap BG, if the displayed portion exceeds the edges of the screen due to the rotation/scaling operation, that area becomes transparent.

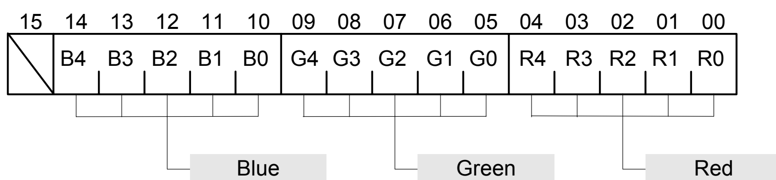
6.2.3 Pixel Data

In the bitmap modes, only the amount of pixel data corresponding to the size of the display screen can be stored in VRAM. Available bitmap modes allow the simultaneous display of 32,768 colors (BG modes 3 and 5) and the display of 256 of the 32,768 colors (BG mode 4). The format of the data in the frame buffer differs between the modes as described below.

6.2.3.1 32,768-Color Simultaneous Display Format (BG Modes 3 and 5)

- Palette RAM is not referenced.
- Each pixel uses a half-word.

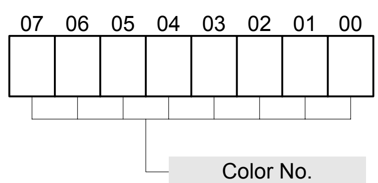
Figure 38 - 32,768-Color Simultaneous Display Format



6.2.3.2 256-Color (of 32,768) Display Format (BG Mode 4)

- Palette RAM color data (256 of the 32,768 colors storable) are referenced.
- Each pixel uses 1 byte.

Figure 39 - 256-Color Display Format



6.2.4 Pixel Data Address Mapping for the LCD Screen

The different address mappings for the different BG modes are shown below.

The frame buffer (VRAM) starts at address 06000000h. Thus, to see the addresses used by the CPU, add 06000000h to the addresses shown below.

6.2.4.1 BG Mode 3 (32,768 colors, 240X160 pixels, 1 frame buffer)

Because there is a single frame buffer, this mode is used mainly for still images. However, it enables 32,768 colors to be displayed simultaneously over the full screen.

Table 10 - BG Mode 3

	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

Table 10 - BG Mode 3 (Continued)

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 address (+06000000h)

6.2.4.2 BG Mode 4 (256 colors, 240X160 pixels, 2 frame buffers)

Two frame buffers are allocated in VRAM, making this mode suitable for full-motion video. Of the total of 32,768 colors, 256 can be displayed simultaneously over the full screen.

(1) Frame 0

Table 11 - BG Mode 4 (Frame 0)

	0	1	2	3	4		236	237	238	239
0	0h	1h	2h	3h	4h		ECh	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

VRAM address (+06000000h)

(2) Frame 1

Table 12 - BG Mode 4 (Frame 1)

	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

Table 12 - BG Mode 4 (Frame 1)

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

VRAM address (+06000000h)

6.2.4.3 BG Mode 5 (32,768 colors, 160X128 pixels, 2 frame buffers)

Although there are 2 frame buffers, the display area is limited in this mode to enable simultaneous display of 32,768 colors.

(1) Frame 0

Table 13 - BG Mode 5 (Frame 0)

	0	1	2	3	4		156	157	158	159
0	0h	2h	4h	6h	8h		138h	13Ah	13Ch	13Eh
1	140h	142h	144h	146h	148h		298h	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	9D80h	9D82h	9D84h	9D86h	9D88h		9EB8h	9EBAh	9EBCh	9EBEh
127	9EC0h	9EC2h	9EC4h	9EC6h	9EC8h		9FF8h	9FFAh	9FFCh	9FFEh

VRAM Address (+06000000h)

(2) Frame 1

Table 14 - BG Mode 5 (Frame 1)

	0	1	2	3	4		156	157	158	159
0	A000h	A002h	A004h	A006h	A008h		A138h	A13Ah	A13Ch	A13Eh
1	A140h	A142h	A144h	A146h	A148h		A298h	A29Ah	A29Ch	A29Eh
2	A2A0h	A2A2h	A2A4h	A2A6h	A2A8h		A3B8h	A3BAh	A3BCh	A3BEh
3	A3C0h	A3C2h	A3C4h	A3C6h	A3C8h		A4F8h	A4FAh	A4FCh	A4FEh

Table 14 - BG Mode 5 (Frame 1) (Continued)

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	13D80h	13D82h	13D84h	13D86h	13D88h		13EB8h	13EBAh	13EBCh	13EBEh
127	13EC0h	13EC2h	13EC4h	13EC6h	13EC8h		13FF8h	13FFAh	13FFCh	13FFEh

VRAM address (+06000000h)

6.3 OBJ (Object)

6.3.1 OBJ Function Overview

Objects are in character format regardless of the BG mode. However, the number of basic characters that can be defined varies depending on the BG mode.

Table 15 - OBJ Function Features

Item	Function
Number of display colors	16 colors/16 palettes or 256 colors/1 palette (mixed display possible)
Number of characters (8 x 8 pixels)	1,024 (16 colors x 16 palettes): in BG modes 0-2 512 (256 colors x 1 palette): in BG modes 0-2 512 (16 colors x 16 palettes): in BG modes 3-5 256 (256 colors x 1 palette): in BG modes 3-5
Character size	8x8 - 64x64 pixels (12 types)
Max. number per screen	128 (64x64 pixel conversion)
Max. number per line	128 (8x8 pixel conversion)
Color special effects	HV flip, semi-transparency, mosaic, priority specification, OBJ windows

6.3.1.1 OBJ Display Capability on a Single Line

The single-line OBJ display capability shown in the table above, is the capability at maximum efficiency.

When the displayed OBJ are arranged continuously from the start of OAM, you can calculate the OBJ display capability on a single line using the following formula:

(Number of H Pixels x 4 **-** 6) / Number of Rendering Cycles =

OBJ Displayable on a single line (Max. of 128)

The “Number of H Pixels” is usually 308 pixels, but when the H-Blank Interval OBJ Processing Flag for Register DISPCNT is set to 1, there are 240 pixels (see "4 LCD" on page 15).

“x 4” expresses the number of cycles that the OBJ Rendering Circuit can use per one pixel. “- 6” represents the number of cycles needed for processing before OBJ rendering at the start of the H Line.

The “Number of Rendering Cycles” and the corresponding number of OBJ displayable for a single line is expressed in the table below.

Table 16 - Rendering Cycles and the Corresponding Number of Displayable Objects

OBJ H Size	Number of Rendering Cycles		Number of OBJ Displayable on a Single Line	
	Normal OBJ	Rotation/ Scaling OBJ	Normal OBJ	Rotation/ Scaling OBJ
8	8	26	128	47
16	16	42	76	29
32	32	74	38	16
64	64	138	19	8
128 (double the size of 64)	X	266	X	4

The table above expresses capabilities at maximum efficiency. In reality, the OAM also contains OBJs which are outside the rendered screen. Therefore, the efficiency will drop. OBJs outside the rendered screen lose 2 cycles.

6.3.2 Character Data Mapping

With OBJ character data, the basic character is 8 x 8 pixels, and characters between 8 x 8 and 64 x 64 pixels can be handled (total of 12 types). The base address of OBJ character data is a fixed VRAM base address. The OBJ character data capacity allocated is either 32 KB or 16 KB, depending on the BG mode (see "5.1.2 VRAM Memory Map" on page 23).

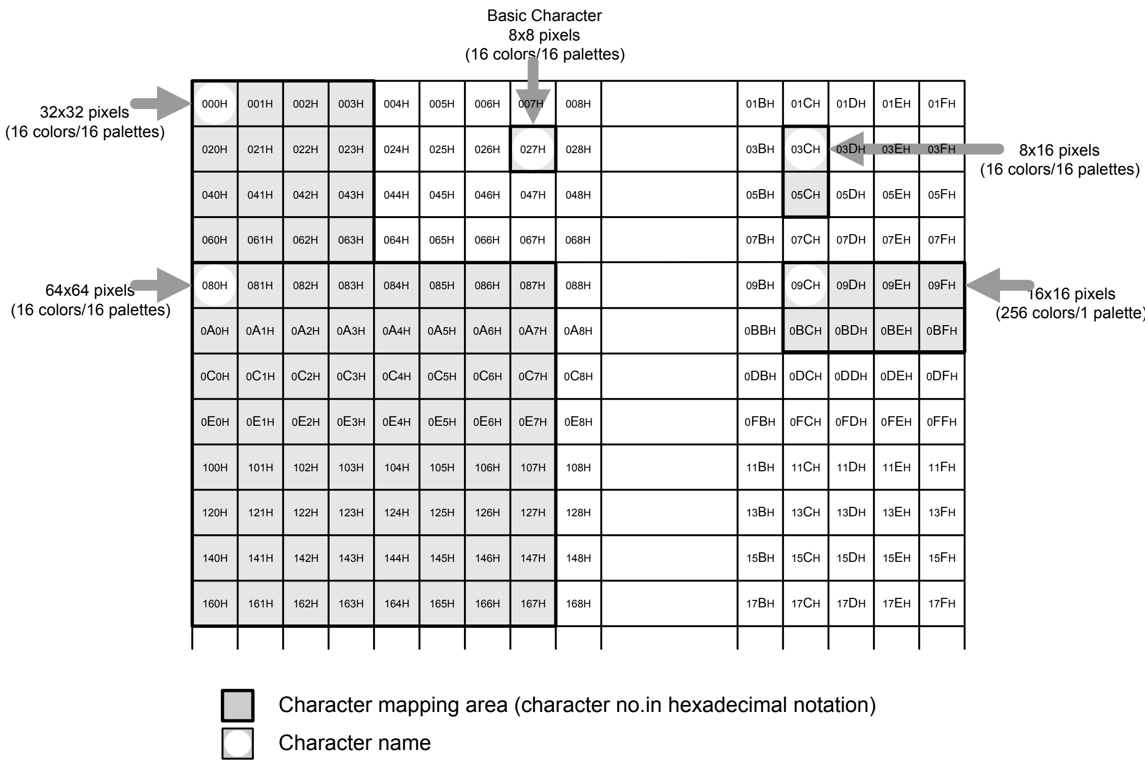
There are 2 types of mapping to the character area, and they can be specified in bit [d06] of the DISPCNT register.

OBJ is managed by character numbers that are divided by 32 bytes starting with OBJ character database address. The required capacity to define 1 basic character of 16 colors x 16 palettes is 32 bytes. The required capacity to define 1 basic character of 256 colors x 1 palette is 64 bytes.

6.3.2.1 VRAM 2-Dimensional Mapping for OBJ Characters

Setting the DISPCNT register bit [d06] to 0 results in the 2-dimensional mapping mode shown in the following figure.

Figure 40 - VRAM 2-Dimensional Mapping for OBJ Characters



Cautions for Character Name

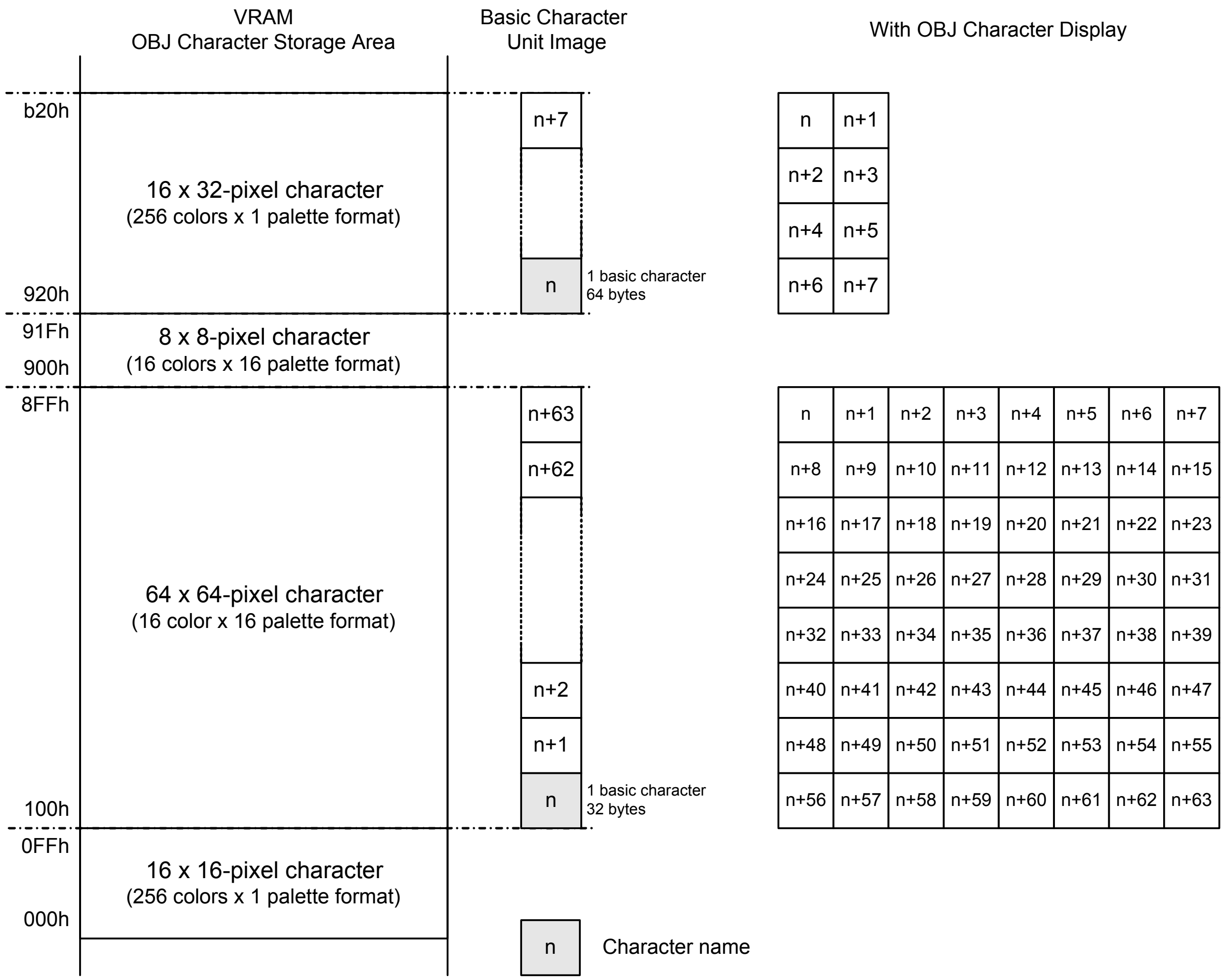
When a character of 256 colors x 1 palette is displayed during 2 dimensional mapping mode, specifying a character name is limited to even numbers (see "Figure 48 - OBJ Attribute 2" on page 56). So, in most cases when defining a character of 256 colors x 1 palette during 2 dimensional mapping mode, you define it so that a character name is an even number.

6.3.2.2 VRAM 1-Dimensional Mapping for OBJ Characters

Setting DISPCNT register bit [d06] to 1 results in the 1-dimensional mapping mode shown in the following figure.

The data that comprise a character are stored in contiguous addresses.

Figure 41 - VRAM 1-Dimensional Mapping for OBJ Characters



6.3.3 OAM

OBJs are displayed by placing data in OAM.

OBJ data for 128 OBJs can be written to internal CPU OAM (addresses 07000000h-070003FFh), and 128 OBJ characters of an arbitrary size can be displayed on the LCD.

6.3.3.1 OAM Mapping

OBJ attributes occupying 48 bits x 128 OBJs can be written to OAM.

In addition, when rotation/scaling are performed for an OBJ, a total of 32 instances of rotation/scaling parameter combinations (PA, PB, PC, and PD) can be written to OAM, as shown in the following figure.

Figure 42 - Writing Rotation/Scaling Parameters to OAM

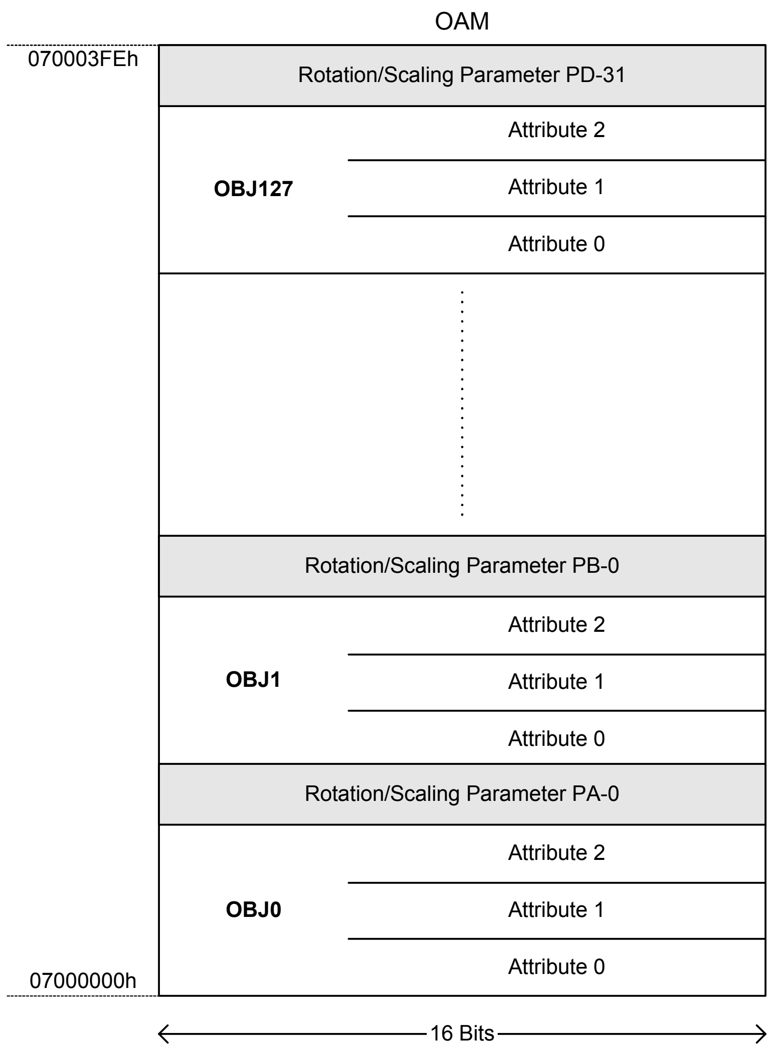
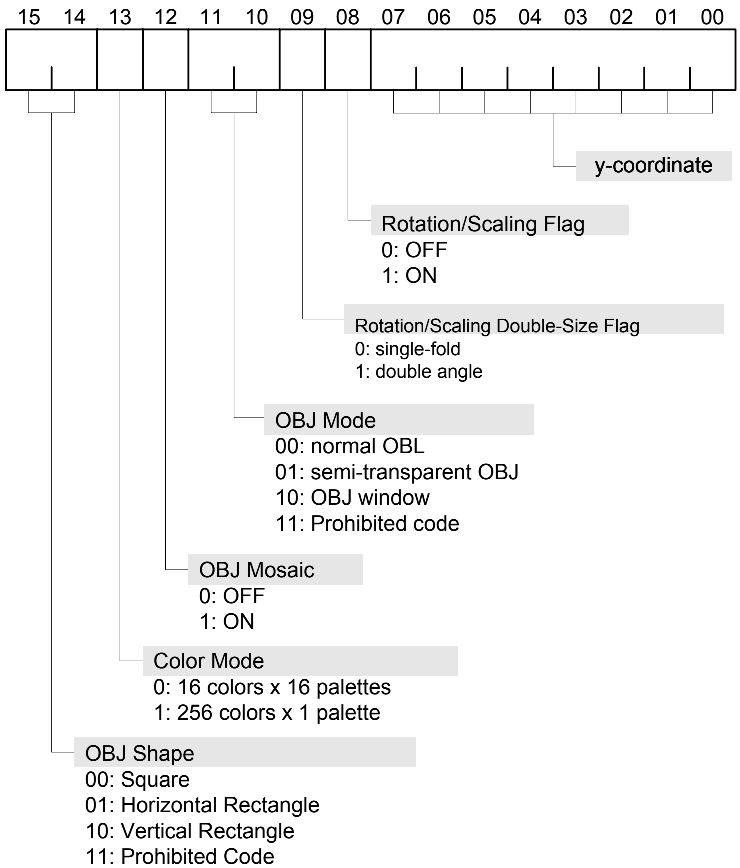


Figure 43 - OBJ Attribute 0



[d15-14] OBJ Shape

Selects the OBJ Character Shape: Square, Horizontal Rectangle, or Vertical Rectangle.

11 is a prohibited code.

Please also refer to the OBJ size specification for OBJ Attribute 1 (see "Figure 47 - Object Sizes" on page 55).

[d13] Color Mode Flag

Specifies whether the OBJ data format is 16 colors x 16 palette mode or 256 colors x 1 palette mode.

[d12] OBJ Mosaic Flag

Turns mosaic for OBJs on and off.

[d11-10] OBJ Mode

Specifies whether an OBJ is a normal OBJ or a semitransparent OBJ.

A normal OBJ is specified by 00, a semi-transparent OBJ by 01, and an OBJ window by 10.

A value of 11 is a prohibited code, so care should be taken to prevent this setting.

When a semi-transparent OBJ is specified, color special effects processing can be performed. For information on color special effects, see "9 Color Special Effects" on page 67.

OBJs for which an OBJ window specification is used are not displayed as normal OBJs; pixels with non-zero character data are used as the OBJ window.

[d09] Rotation/Scaling Double-Size Flag

OBJs are limited in size by the OBJ field (8x8 - 64x64 pixels), and the character data may surpass the boundaries of this field when rotated.

This problem can be avoided by implementing a pseudo double-size for the OBJ field, by setting the double-size flag to 1.

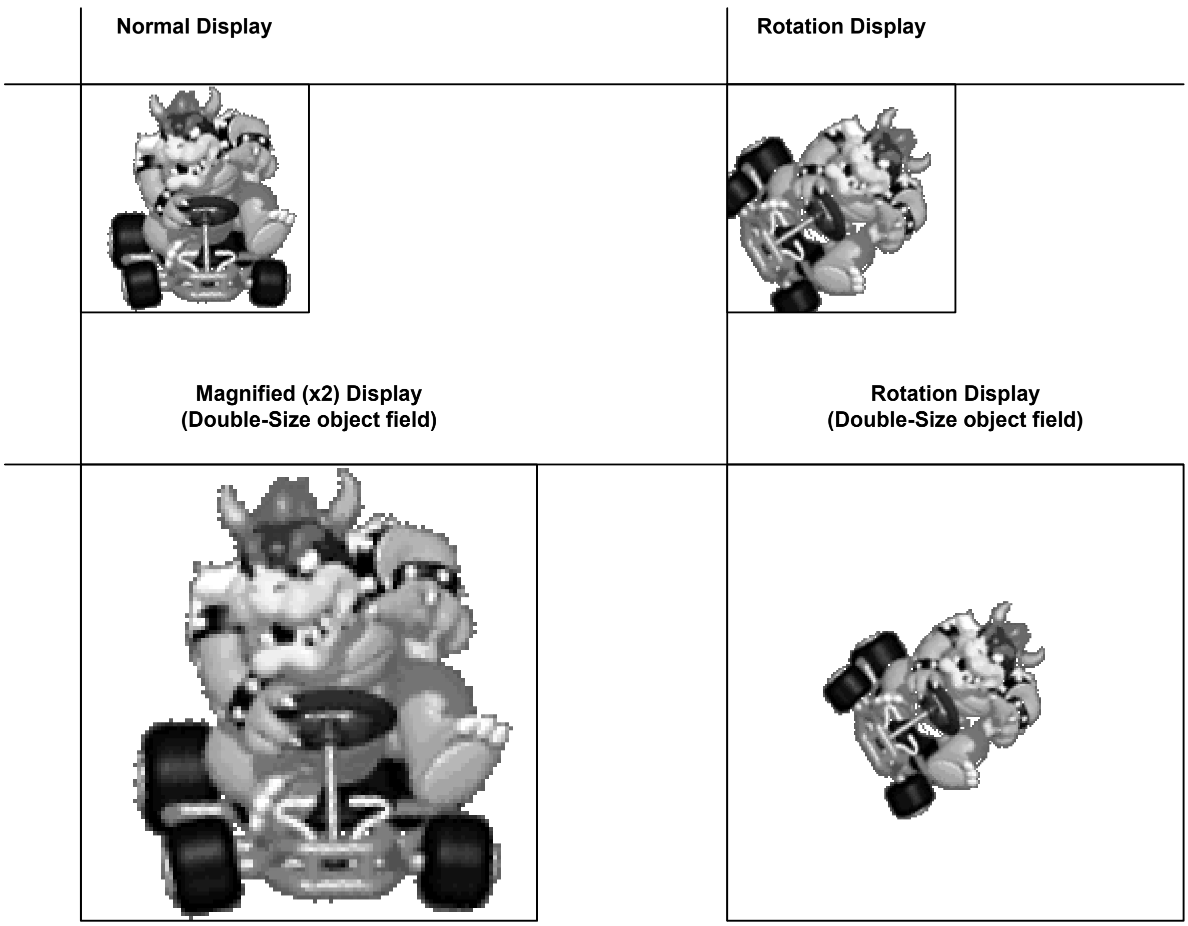
With this setting, the OBJ does not surpass the boundaries of the OBJ field even if the OBJ display is magnified by up to two-fold.

Example: A 64x64 pixel OBJ field increased to a 128x128 pixel field displayed with rotation processing. Note, however, that the OBJ display position is shifted.

With the double-size flag set to 0, display of the portion protruding from the edges is cut off.

Please refer to the following figure.

Figure 44 - Cropping when Displaying a Scaled or Rotated Object



Individual Control of OBJ display

It is possible to control the ON and OFF functions of the OBJ display individually by setting in the combination of this double size flag and the rotation/scaling flag of [d08].

In case of (double size flag, rotation/scaling flag) = (1, 0), OBJ is not displayed, but is displayed in other cases.

[d08] Rotation/Scaling Flag

Allows rotation processing for the OBJ to be enabled and disabled.

With the OBJ rotation/scaling feature enabled by setting this bit to 1, the maximum number of OBJs displayed per line is decreased. Please refer to the description in "6.3.1.1 OBJ Display Capability on a Single Line" on page 47.

Individual Control of OBJ display

It is possible to control the ON and OFF functions of the OBJ display individually by setting in the combination of the double size flag for [d09] and this rotation/scaling flag.

In case of (double size flag, rotation/scaling flag) = (1, 0), OBJ is not displayed, but is displayed in other cases.

[d07-00] Y-Coordinate

Allows the y-coordinate of the OBJ in the display screen to be specified.

Cautions

160 pixels in total (0 - 159) are inside the display screen, and 96 pixels in total (160 - 255) are outside the display screen (virtual screen).

When the vertical size displays a 64 pixel OBJ by a double size of character, the size is 128 pixels, exceeding the vertical 96 pixels for the virtual screen.

Therefore, in the range of Y coordinate values of 129 - 159, the lower part of OBJ that is pushed out upwards is displayed. The upper part of OBJ in the lower screen is not displayed (see the figure below).

Figure 45 - Cropping when Displaying a Magnified Character

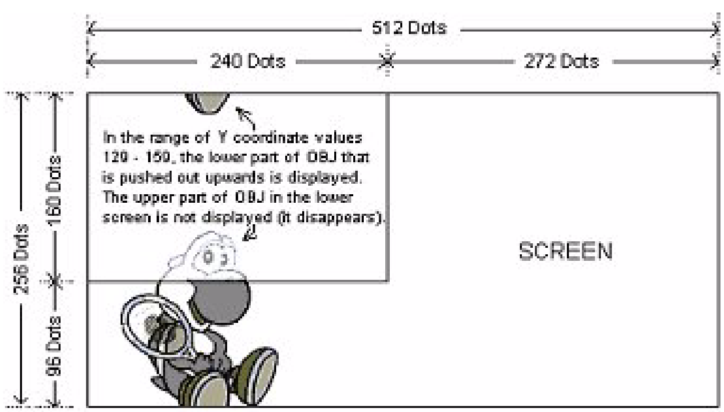
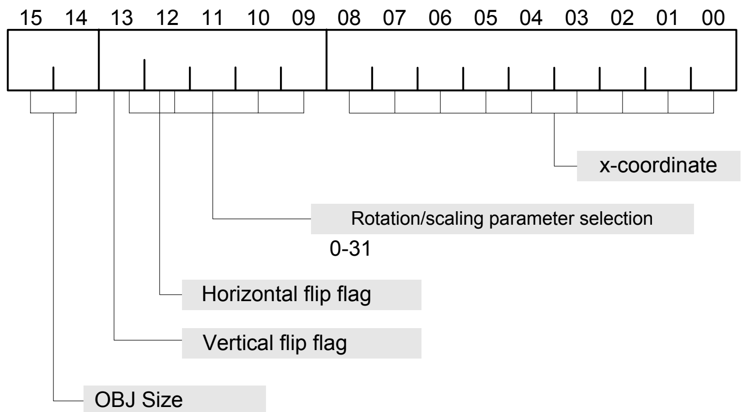


Figure 46 - OBJ Attribute 1



[d15-14] OBJ Size

Linked to the specification of the OBJ size for Attribute 0, the size for the OBJ Character is also specified. For each of the three OBJ shapes, you can set four sizes.

Figure 47 - Object Sizes

OBJ Shape		OBJ Size 00	01	10	11
00	Square	A 8x8	B 16x16	C 32x32	D 64x64
01	Horizontal Rectangle	E 16x8	F 32x8	G 32x16	H 64x32
10	Vertical Rectangle	I 8x16	J 8x32	K 16x32	L 32x64
11		Prohibited Code			

[d13] [d12] Vertical and Horizontal Flip Flags

Allows the OBJ to be flipped horizontally and vertically.

A normal display is produced by a setting of 0 and a flip display by a setting of 1.

When the rotation/scaling flag ([d08] of OBJ Attribute 0) is enabled, these bits also can be used as the high-order bits of the rotation/scaling parameter selection.

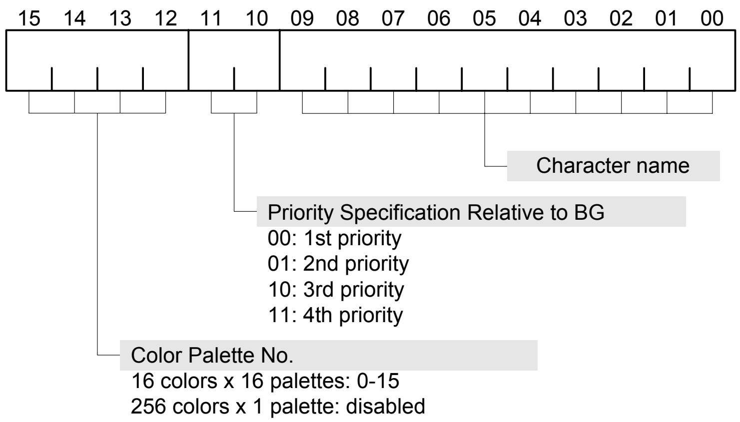
[d13-09] Rotation/Scaling Parameter Selection

The parameters used in OBJ rotation/scaling processing are selected from the 32 parameters registered in OAM.

[d08-00] X-Coordinate

Specifies the x-coordinate of the OBJ on the display screen in the range of 0~511.

Figure 48 - OBJ Attribute 2



[d15-12] Color Palette No.

When 16 colors x 16 palette format is specified in the color mode bit, these bits specify 1 of the 16 palettes to apply to the character data.

When 256 colors x 1 palette format is specified in the color mode bit, these bits are disabled.

[d11-10] Priority Relative to BG

Specifies the display priority of the OBJ relative to BG. For information on priority, see "6.4 Display Priority of OBJ and BG" on page 58.

[d09-00] Character Name

Writes the number of the basic character located at the start of the OBJ character data mapped in VRAM. (See "6.3.2 Character Data Mapping" on page 48).

16 colors x 16 palettes (color mode=1)

Allows selection of 1,024 characters.

256 colors x 1 palette (color mode=0)

Allows selection of 512 characters.

Bit 0 fixed at 0 in 2-dimensional mapping mode.

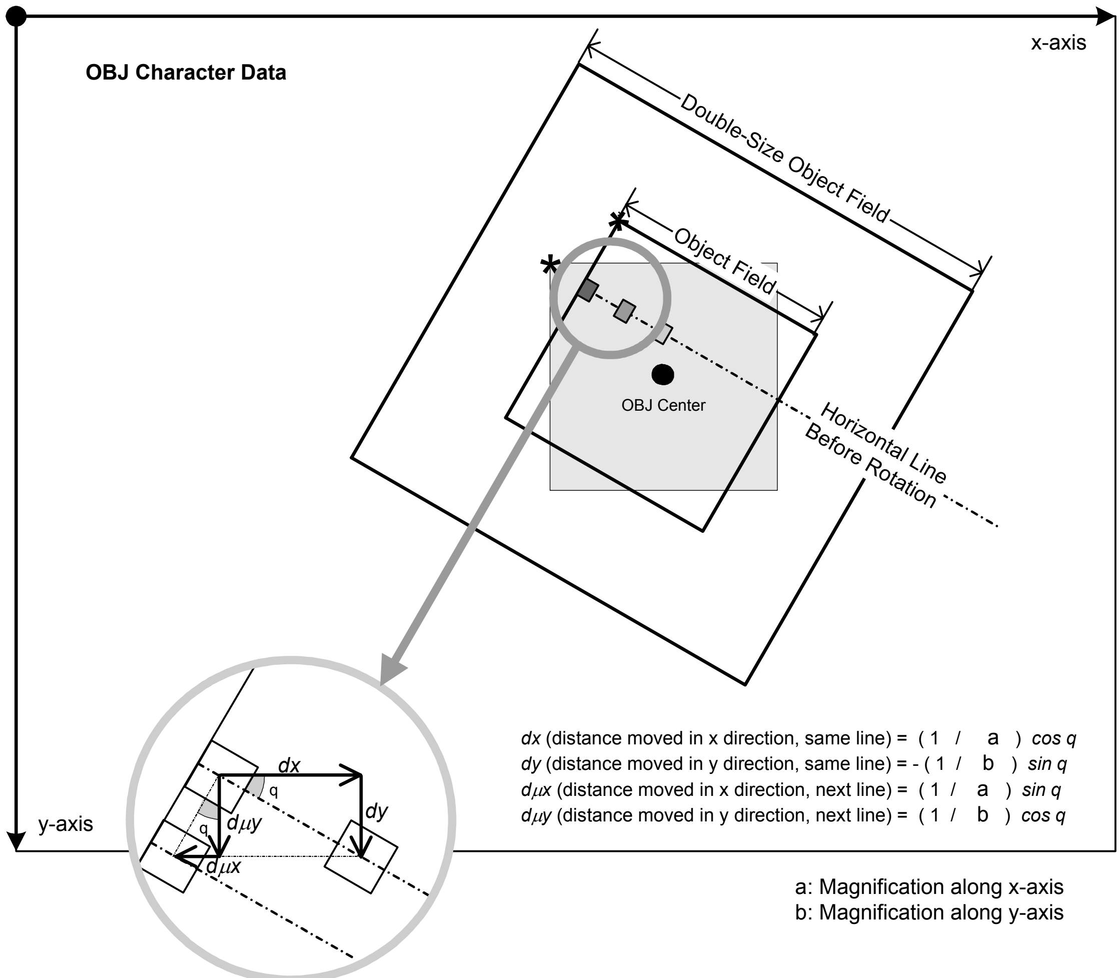
BG Mode is 3~5 (Bitmap Mode)

OBJ character data RAM is halved to 16 KB, so character name numbers 0-511 are disabled and numbers 512 and greater are used.

6.3.4 OBJ Rotation/Scaling Feature

The rotation and scaling feature for OBJ is essentially the same as that for BG.

Figure 49 - OBJ Character Data Referenced with Rotation



When an OBJ is displayed, the OBJ character data are referenced horizontally, beginning from the left-uppermost position. Rotation display can be achieved by adding an angle to the reference direction. The center of rotation is fixed at the center of the OBJ field. If a reference point surpasses the specified OBJ size, it becomes transparent.

6.3.4.1 Operations Used in OBJ Rotation/Scaling Processing

1. Specify the rotation/scaling parameter number to be applied in OBJ Attribute 1 of the OAM.
2. The image-processing circuit sums the increases in the x direction (dx, dy) in relation to the center of rotation (OBJ field center), which serves as reference point, to calculate the x-direction coordinates.
3. When the line is advanced, the increases in the y-direction (dμx, dμy) in relation to the reference point, are summed to calculate the coordinate of the starting point for rendering the next line. The processing in step 2) above, is then performed.

6.3.4.2 Rotation/Scaling Parameters

Specifies the direction of character data reference in OBJ rotation/scaling processing.

The values set for PA, PB, PC, and PD are signed, fixed-point numbers (8-bit fractional portion, 7-bit integer portion, 1-bit sign, for a total of 16 bits).

These 4 parameters are used together as a single group, which can be placed in any of 32 areas in OAM.

Figure 50 - Object Rotation/Scaling Parameters

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PA	dx: distance moved in x direction along same line															
PB	dμx: distance moved in x direction along next line															
PC	dy: distance moved in y direction along same line															
PD	dμy: distance moved in y direction along next line															

6.4 Display Priority of OBJ and BG

6.4.1 Priority Among BGs

Priority among BGs can be set to any of 4 levels.

When BGs have the same priority setting, the BG with the lowest BG number is given priority.

6.4.2 Priority Among OBJs

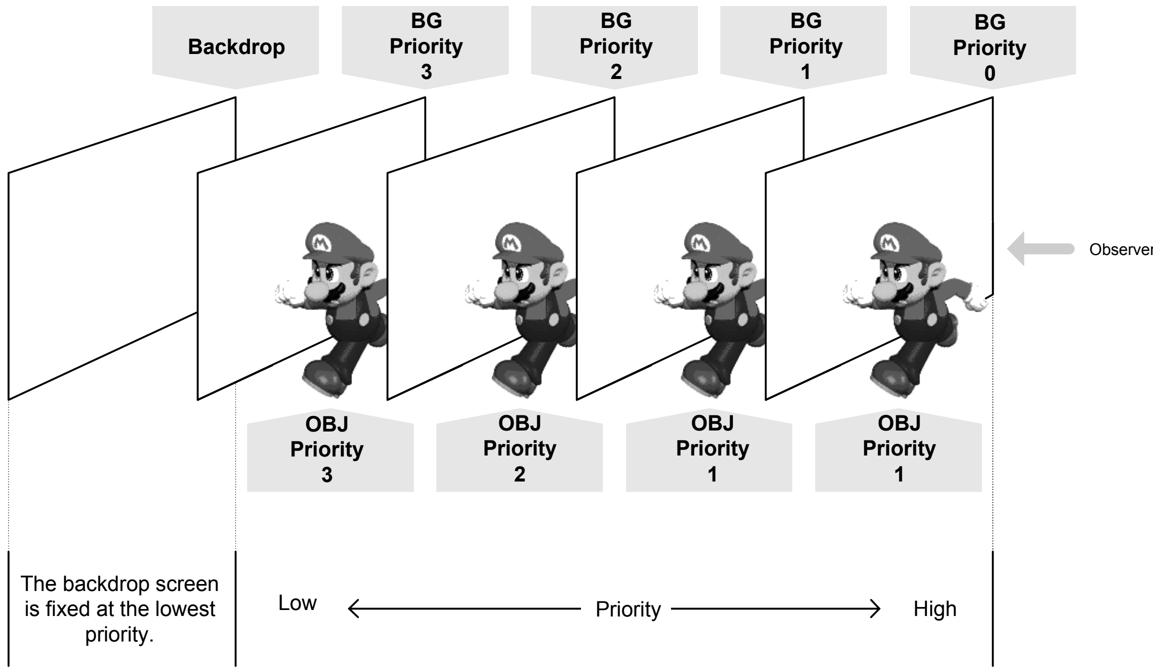
Priority among OBJs can be set to any of 4 levels.

When OBJs have the same priority setting, the OBJ with the lowest OBJ number is given priority.

6.4.3 Priority Among BGs and OBJs

The priority of each OBJ in relation to the BG can be set to 4 levels. Please refer to the following figure.

Figure 51 - Background and Object Priority



Cautions for priority

When orders of OBJ number and OBJ priority are reversed, the display is not right. Please be cautious not to let this situation occur.

Examples of when display is not right:

- OBJ-No.0 (OBJ priority 2)
- BG (BG priority 1)
- OBJ-No.1 (OBJ priority 0)

7 Color Palettes

7.1 Color Palette Overview

The LCD unit of Game Boy Advance can display 32 levels of red, 32 levels of green, and 32 levels of blue, for a total of 32,768 colors.

The number of colors that can be displayed at once varies with the BG mode. See "5.1.1 Details of BG Modes" on page 21. Color palettes are used in defining character-format BGs and OBJs.

Note: Bitmap-format BG modes 3 and 5 are not palette formats. See "6.2 Bitmap Mode BGs (BG Modes 3-5)" on page 43.

Color palettes come in the following two forms.

7.1.1 16 Colors x 16 Palettes

This mode provides 16 color palettes, each consisting of 16 colors.

Color 0 for OBJ and BG palettes is forcibly allocated to transparent (color specification disabled).

7.1.2 256 Colors x 1 Palette

This mode allocates all 256 of its colors to 1 palette.

Color data are represented by 15 bits (5 for Red, 5 for Green, and 5 for Blue). Colors can be selected from the total of 32,768.

OBJ color 0 and BG color 0 are forcibly allocated to transparent (color specification disabled).

7.1.3 Color 0 Transparency

Color 0 transparency is used to render the pixels of low-priority OBJs or BGs as transparent.

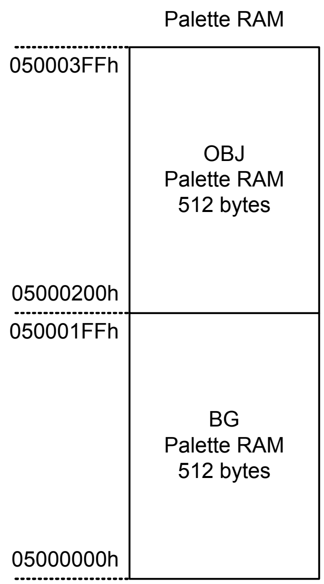
Note: The color specified for color 0 of BG palette 0 is applied to the backdrop, which has the lowest priority.

7.2 Color Palette RAM

OBJs and BGs use separate palettes.

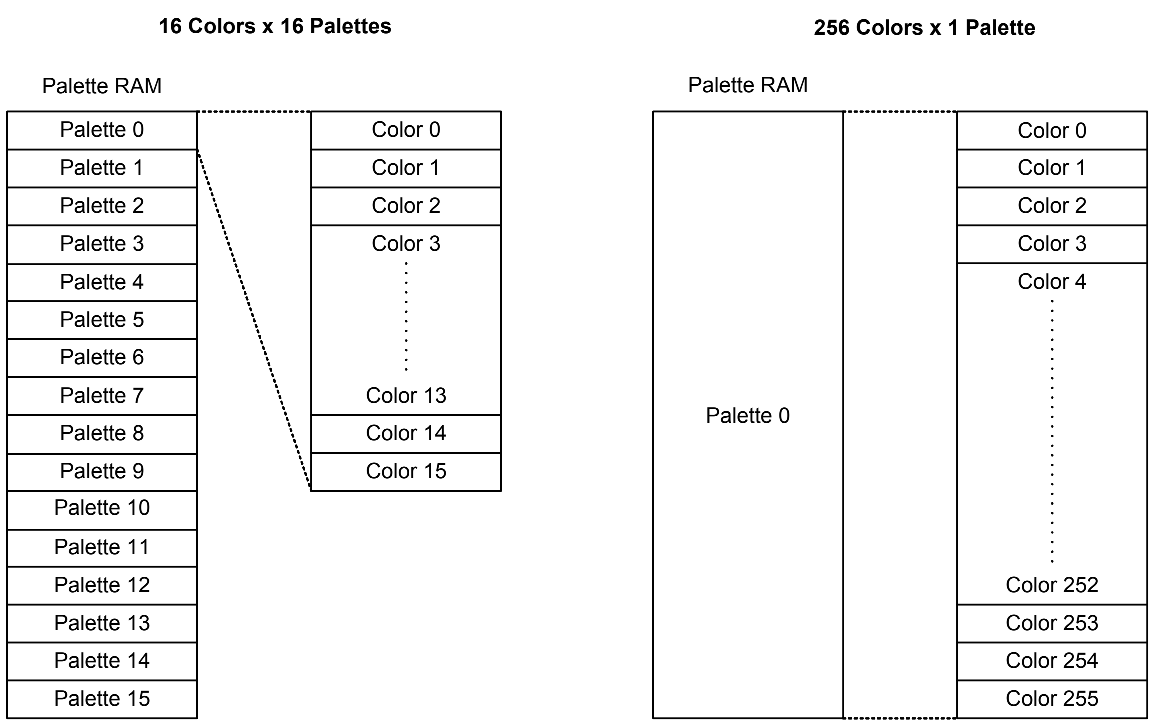
The size of palette RAM is large enough (512 bytes) to hold data (16-bit) for up to 256 colors (of 32,768) that can be specified. The memory map of the OBJ and BG palettes is shown in the follow figure.

Figure 52 - Color Palette RAM Memory Map



Either of 2 modes (16 colors x 16 palette and 256 colors x 1 palette) can be selected for OBJ and BG. Palette RAM for these modes is referenced as shown in the following figure.

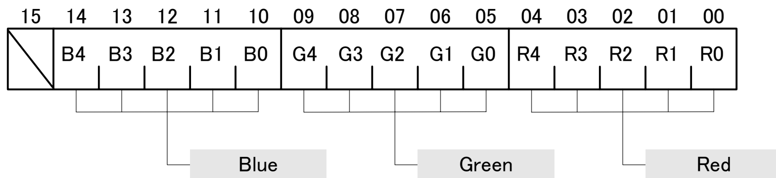
Figure 53 - Referencing Palette RAM for OBJ and BG Modes



7.3 Color Data Format

Allows 1 of 32,768 colors to be specified.

Figure 54 - Color Data Format



8 Window Feature

The Game Boy Advance system can display 2 windows simultaneously.

Display of the areas inside and outside the windows can be separately turned on and off.

In addition, scrolling and color special effects such as rotation, α blending, and fade-in/fade-out can be performed for each window.

8.1 Window Position Setting

The Window Position Setting specifies the upper-left and lower-right coordinates of a rectangular area.

These settings specify the window's position and size.

When a non-rectangular window is displayed, the values of these registers are updated during H-blanking intervals.

Figure 55 - Window Position Setting Registers

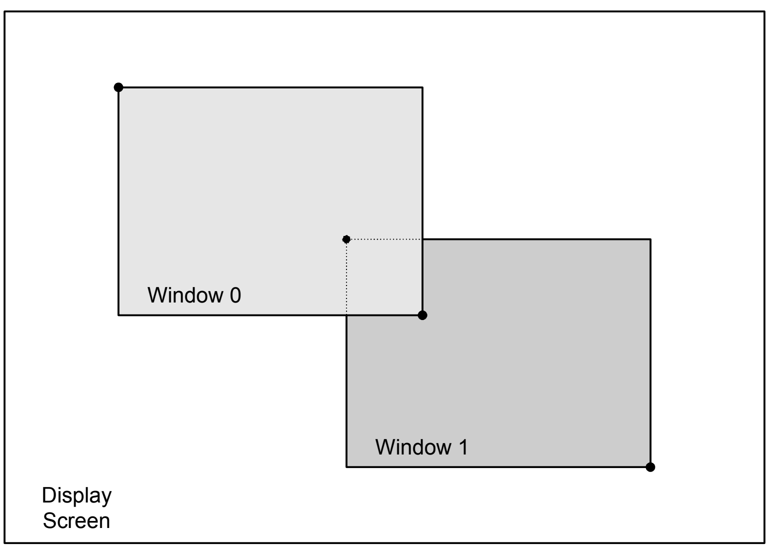
Address	Register	15–08	07–00	Attributes	Initial Value
040h 042h	WIN0H WIN1H	Left-upper x-coordinate of window	Right-lower x-coordinate of window	W	0000h

Address	Register	15–08	07–00	Attributes	Initial Value
044h 046h	WIN0V WIN1V	Left-upper y-coordinate of window	Right-lower y-coordinate of window	W	0000h

Window Display Example

Window 0 has a higher display priority than Window 1.

Figure 56 - Window Display Priority Example



8.2 Window Control

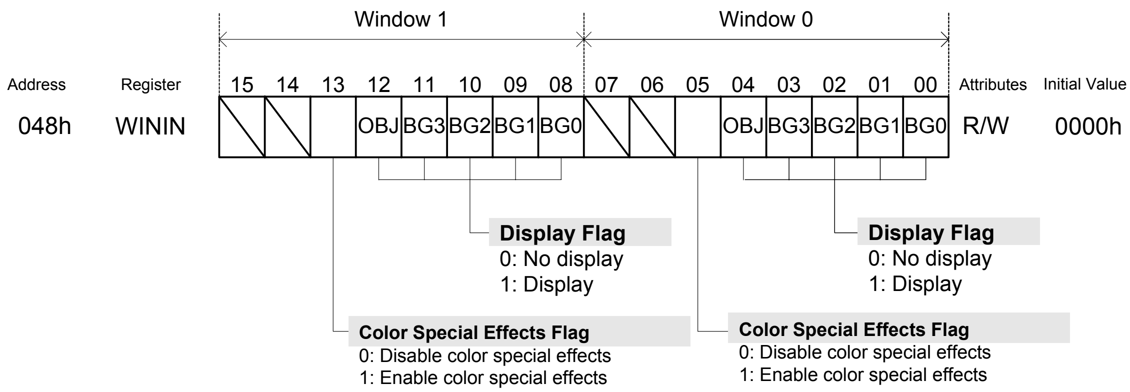
The window control registers control operations such as turning window display on and off.

However, the master window display flag of the DISPCNT register has a higher priority than the WININ and WINOUT registers. For information concerning the DISPCNT register, see "5 Image System" on page 19.

8.2.1 Control of Inside of Window

The WININ register controls display of the area inside windows 0 and 1. The high-order bits (d13-8) control Window 1, while the low-order bits (d5-0) control Window 0.

Figure 57 - The WININ Register

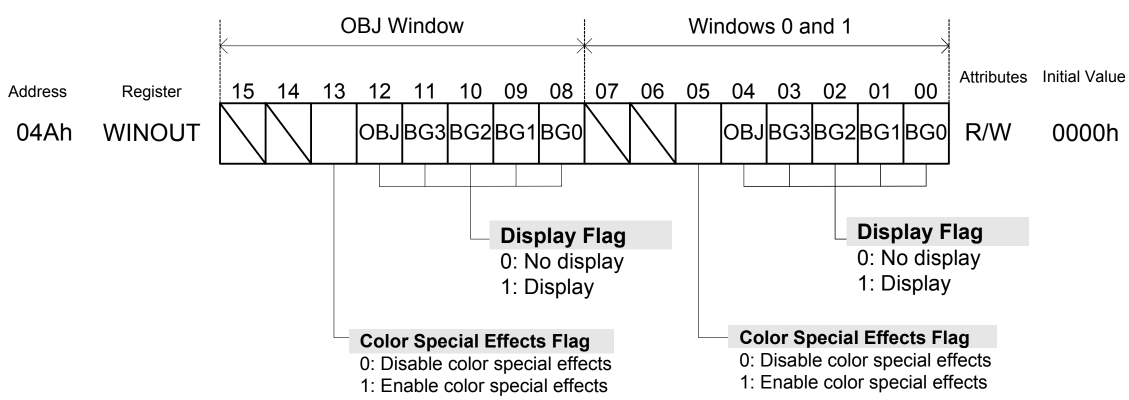


8.2.2 Control of Outside of Window and Inside of OBJ Window

The WINOUT register controls display of the area outside the window. It controls both windows 0 and 1.

In addition, it controls display of the area inside the OBJ window.

Figure 58 - The WINOUT Register



WININ [d12-08][d04-00], WINOUT[d12-08][d04-00] Display Flags

Turns display of the OBJ and BG 3-0 on and off.

A setting of 0 turns display off, and 1 turns display on.

WININ [d13][d05], WINOUT[d13][d05] Color Special Effects Flags

A setting of 0 disables color special effects; 1 enables them. For information on color special effects, see "9 Color Special Effects" on page 67.

9 Color Special Effects

The Game Boy Advance provides the following color special effects. The area where these effects are applied can be limited using a window.

1. α Blending

   Performs arithmetic operations on 2 selected surfaces and implements processing for 16 levels of semi-transparency.

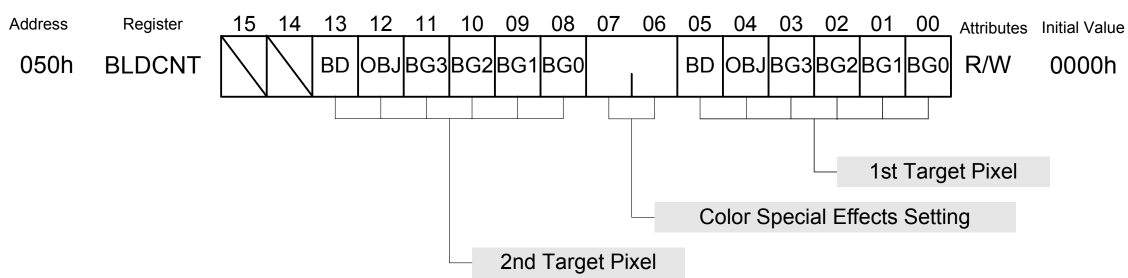
2. Fade-in/Fade-out

   Performs arithmetic operations on 1 selected surface and implements processing for 16 levels of brightness.

9.1 Selection of Color Special Effects

The types of color special effects and the target pixels, are determined by the BLDCNT register.

Figure 59 - The BLDCNT Register



Although color special effects are specified by the BLDCNT register, for α blending, which involves processing between surfaces, the 2 target surfaces must have suitable priorities.

In addition, semi-transparent OBJs are individually specified in OAM, and color special effects for the OBJ as a whole, are specified in the BLDCNT register. These specifications are summarized in the following table.

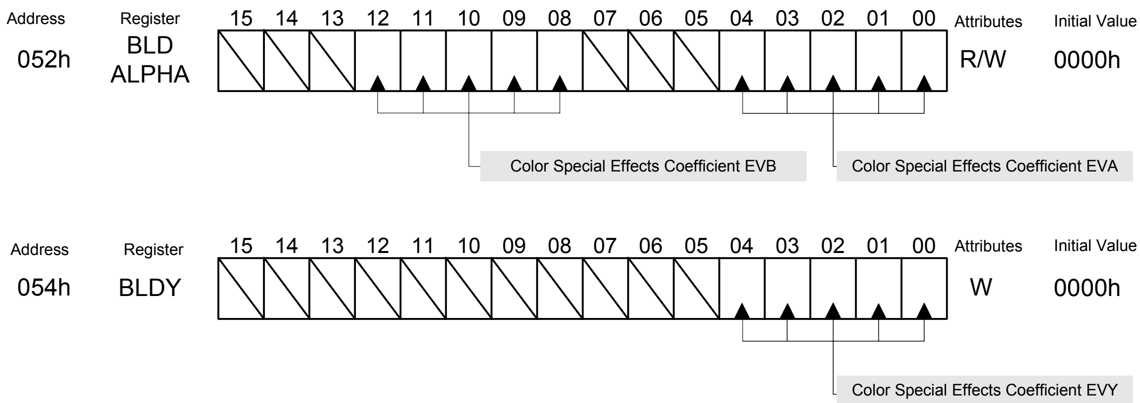
Table 17 - Specifications for the BLDCNT Register

BLDCNT		Type	Color Special Effects Processing
d07	d06		
0	0	No special effects	Normally, color special effects processing is not performed. 16-level semi-transparency processing (α blending) is performed only when a semi-transparent OBJ is present and is followed immediately by a 2nd target screen.
0	1	α blending (Semi-transparency processing)	If the 1st target screen is followed immediately by a 2nd target screen, 16-level semi-transparency processing (α blending) is performed. The bits of the backdrop of the 1st target screen should be turned off ([d05]=0). When OBJ = 1 for the 1st target pixel, processing is executed for all OBJs regardless of the OBJ type. When OBJ=0, processing is executed only if the OBJ is semi-transparent.
1	0	Brightness Increase	Gradually increases brightness for 1st target screen. The entire screen can gradually be made whiter by setting all bits of the specification for the 1st target screen to 1. When OBJ=1 for the 1st target screen, processing for increased brightness is executed only for normal objects. If a semi-transparent OBJ is the 1st target screen, α blending processing is always executed.
1	1	Brightness Decrease	Gradually decreases brightness for 1st target screen. The entire screen can gradually be made darker by setting all bits of the specification for the 1st target screen to 1. When OBJ=1 for the 1st target screen, processing for decreased brightness is executed only for normal objects. If a semi-transparent OBJ is the 1st target screen, α blending processing is always executed.

9.2 Color Special Effects Processing

9.2.1 Coefficients for Color Special Effects

Figure 60 - The BLDALPHA and BLDY Registers



Coefficients used in α blending processing are specified in EVA and EVB of the BLDALPHA register.

The coefficient used in processing brightness changes is specified in EVY of the BLDY register.

The values of EVA, EVB, and EVY are numbers less than 1 and are obtained by multiplying 1/16 by an integer.

Table 18 - EVA, EVB, and EVY Values

EVA, EVB, EVY					Coeff.	EVA, EVB, EVY					Coeff.
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

The color special effects arithmetic expressions that use the coefficients are shown below.

9.2.1.1 Alpha Blending (16 levels of semi-transparency) Operations

- Display color (R) = 1st pixel color (R) x EVA + 2nd pixel color (R)×EVB
- Display color (G) = 1st pixel color (G) x EVA + 2nd pixel color (G) ×EVB
- Display color (B) = 1st pixel color (B) x EVA + 2nd pixel color (B) ×EVB

9.2.1.2 Brightness Increase Operations

- Display color (R) = 1st pixel (R) + (31 - 1st pixel (R)) ×EVY
- Display color (G) = 1st pixel (G) + (31 - 1st pixel (G)) ×EVY
- Display color (B) = 1st pixel (B) + (31 - 1st pixel (B)) ×EVY

9.2.1.3 Brightness Decrease Operations

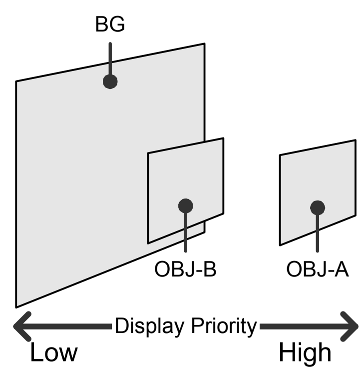
- Display color (R) = 1st pixel (R) - 1st pixel (R) ×EVY
- Display color (G) = 1st pixel (G) - 1st pixel (G) ×EVY
- Display color (B) = 1st pixel (B) - 1st pixel (B) ×EVY

Note:

There is no method for α blending between OBJs.

In Figure 61, the OBJ is designated in the first target screen, and the BG and OBJ are designated in the second target screen. In this case OBJ-B is ignored as the α blending's target pixel, and it is considered that there is a BG immediately after OBJ-A. If this is the result, α blending is conducted on OBJ-A and the BG.

Figure 61 - α Blending between OBJ and BG



10 Sound

In addition to 4 channels of CGB-compatible sound, Game Boy Advance has 2 channels of direct sound.

1. Direct Sounds A and B
   - Provides playback of linear 8-bit audio data.
   - Uses the timer and DMA.
2. Sound 1

   Allows generation of rectangular waveforms with sweep (frequency change) and envelope (volume change) functions.
3. Sound 2

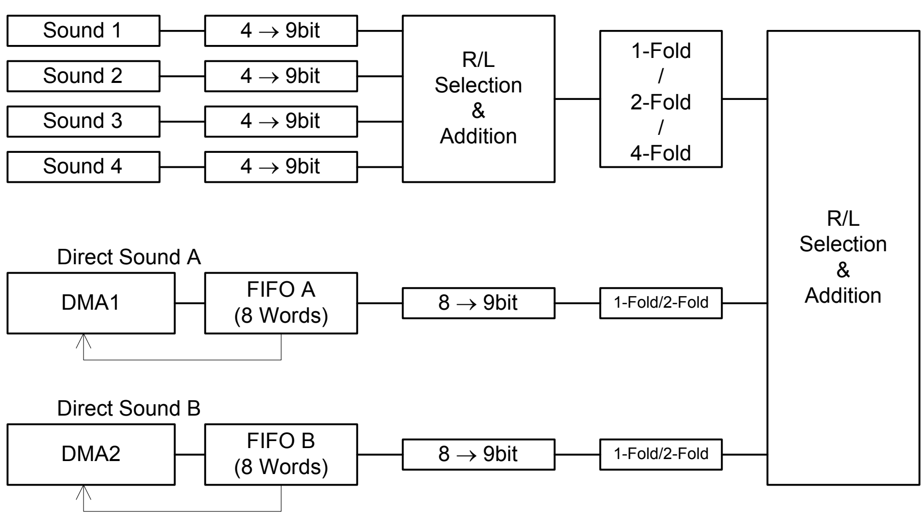
   Allows generation of rectangular waveforms with envelope functions.
4. Sound 3
   - Allows playback of any waveform recorded in waveform RAM.
   - Waveform RAM in Game Boy Advance has double the capacity of that in CGB.
5. Sound 4

   Can generate white noise with the envelope function.

The synthesis ratio of sounds 1-4 to direct sound can be specified.

10.1 Sound Block Diagram

Figure 62 - Game Boy Advance Sound System Block Diagram



10.2 Direct Sounds A and B

Direct sounds have 2 channels, A and B. Linear 8-bit audio data can be played back.

The audio data are set to a bias level of 00h and are 8-bit data (+127 to -128), obtained by 2's complement.

Audio data are transferred sequentially to the sound FIFO (8-word capacity), using the sound FIFO transfer mode of DMA 1 and 2.

The sampling rate can be set to an arbitrary value using timers 0 and 1.

10.2.1 Sound FIFO Input Registers

Figure 63 - Sound FIFO Input Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0A0h 0A4h	FIFO_A_L FIFO_B_L	Sound Data 1								Sound Data 0								W	-

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0A2h 0A6h	FIFO_A_H FIFO_B_H	Sound Data 3								Sound Data 2								W	-

10.2.1.1 Sound Data

All sounds are PWM modulated (refer to "10.8 Sound PWM Control" on page 87) at the final portion of the Sound Circuit. Therefore, if you match the 8 bit audio data sampling frequency and the timer settings with the PWM modulation sampling frequency, a clean sound can be produced.

The following operations are repeated for direct sound.

10.2.1.2 Preparing to Use Direct Sound

1. Using sound control register SOUNDCNT_H (refer to "10.7 Sound Control" on page 84), select the timer channel to be used (0 or 1).
2. Using sound control register SOUNDCNT_H, do a 0 clear with FIFO A and FIFO B, and initialize the sequencer.
3. In cases of producing a sound immediately after starting the direct sound, write the first 8 bits of linear audio data to the FIFO with a CPU write.
4. Specify the transfer mode for DMA 1 or 2 (see "12.2 DMA 1 and 2" on page 95).
5. Specify the direct sound outputs settings in the sound control register.
6. Start the timer.

With the preceding preparations, direct sound is executed as follows.

10.2.1.3 Direct Sound Execution

1. When the specified timer overflows due to a count up, the audio data are passed from the FIFO to the sound circuit.
2. If 4 words of data remain in the FIFO as the transfer count progresses, the FIFOs for direct sounds A and B output a data transfer request to the specified DMA channel.
3. If the DMA channel receiving the request is in sound FIFO transfer mode, 4 words of data are provided to the sound FIFO (the DMA WORD COUNT is ignored).

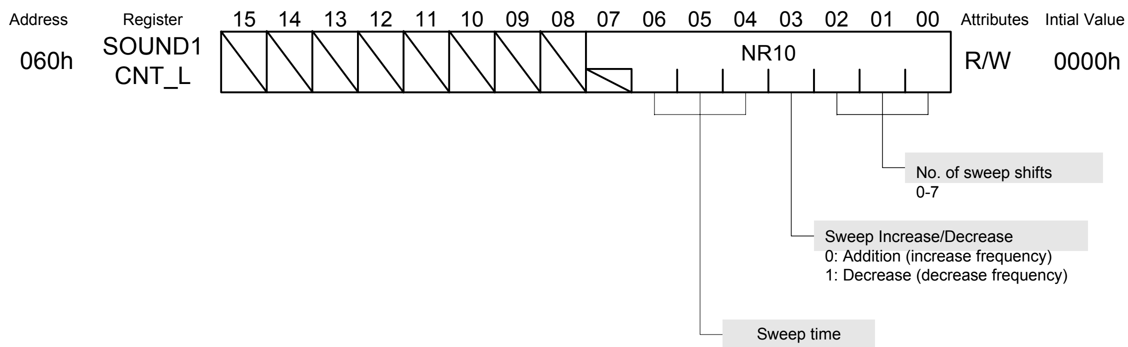
The preceding is repeated starting from 1.

10.3 Sound 1

Sound 1 is a circuit that generates rectangular waveforms with sweep (frequency change) and envelope (volume change) functions.

The contents of NR10, NR11, NR12, NR13, and NR14 for Sound 1, conform with those of CGB.

Figure 64 - The SOUND1CNT_L Register



SOUND1CNT_L [d06 - 04] Sweep Time

These bits specify the interval for frequency change.

Table 19 - Sound 1 Frequency Change Bits

Setting	Sweep Time
000	Sweep OFF
001	1/f128 (7.8 ms)
010	2/f128 (15.6 ms)
011	3/f128 (23.4 ms)
100	4/f128 (31.3 ms)
101	5/f128 (39.1 ms)
110	6/f128 (46.9 ms)
111	7/f128 (54.7 ms)

(f128=128Hz)

SOUND1CNT_L [d03] Sweep Increase/Decrease

Specifies whether the frequency increases or decreases.

When the sweep function is not used, the increase/decrease flag should be set to 1.

SOUND1CNT_L [d02 - 00] Number of Sweep Shifts

Specifies the number of sweeps.

The frequency data with a single shift are determined according to the following formula, with $f_{(t)}$ signifying the frequency after a shift and $f_{(t-1)}$ the frequency before the shift.

Equation 2 - Determining Single-Shift Frequency Data

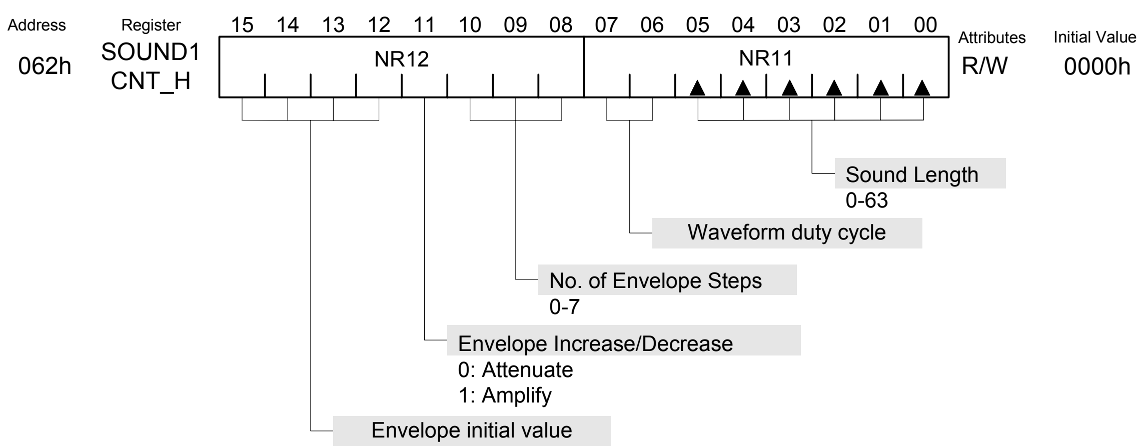
$$f_{(t)} = f_{(t-1)} \pm \frac{f_{(t-1)}}{2^n}$$

$$f_{(0)} = \text{Initial frequency data}$$

If the addition according to this formula produces a value consisting of more than 11 bits, sound output is stopped and the Sound 1 ON flag (bit 0) of NR52 is reset.

With subtraction, if the subtrahend is less than 0, the pre-subtraction value is used. However, if the specified setting is 0, shifting does not occur and the frequency is unchanged.

Figure 65 - The SOUND1CNT_H Register



SOUND1CNT_H [d15 - 12] Envelope Initial-Value

Allows specification of any of 16 levels ranging from maximum to mute.

SOUND1CNT_H [d11] Envelope Increase/Decrease

Specifies whether to increase or decrease the volume.

SOUND1CNT_H [d10 - 08] Number of Envelope Steps

Sets the length of each step of envelope amplification or attenuation.

With n the specified value, the length of 1 step (steptime) is determined by the following formula.

Equation 3 - Determining the Length of 1 Step (steptime)

$$steptime = n \times \frac{1}{64} (\text{sec})$$

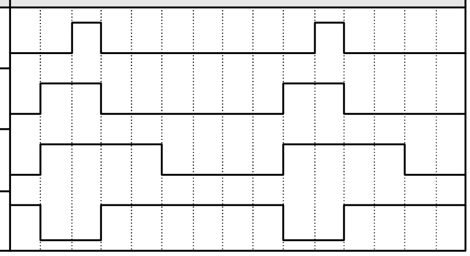
When n = 0, the envelope function is turned off.

SOUND1CNT_H [d07 - 06] Waveform Duty Cycle

Specifies the proportion of amplitude peaks for the waveform.

Figure 66 - Waveform Amplitude Peak Proportions

Setting	Duty Cycle	Waveform
00	12.5%	
01	25.0%	
10	50.0%	
11	75.0%	



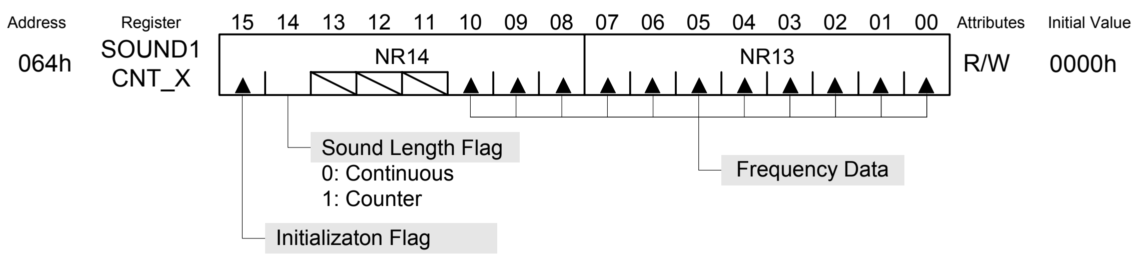
SOUND1CNT_H [d05 - 00] Sound Length

With st signifying the sound length, the length of the output sound is determined by the following formula.

Equation 4 - Determining the Length of the Output Sound

$$time = (64 - st) \times \frac{1}{256}(\sec)$$

Figure 67 - The SOUND1CNT_X Register



SOUND1CNT_X [d15] Initialization Flag

A setting of 1 causes Sound 1 to restart.

When the sweep function is used, set the initialization flag again after an interval of 8 clocks or more.

SOUND1CNT_X [d14] Sound Length Flag

When 0, sound is continuously output.

When 1, sound is output for only the length of time specified for the sound length in NR11.

When sound output ends, the Sound 1 ON flag of NR52 is reset.

SOUND1CNT_X [d10-00] Frequency Data

With fdat signifying the frequency, the output frequency (f) is determined by the following formula.

Equation 5 - Determining the Output Frequency

$$f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)}(Hz)$$

Thus, the specifiable range of frequencies is 64 to 131.1 KHz.

Sound 1 Usage Notes

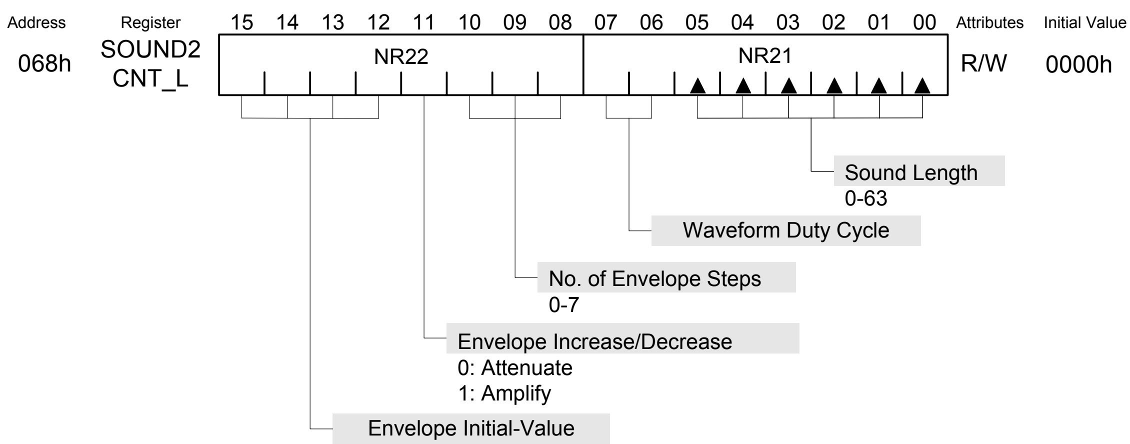
1. When the sweep function is not used, the sweep time should be set to 0 and the sweep increase/decrease flag should be set to 1.
2. If sweep increase/decrease flag of NR10 is set to 0, the number of sweep shifts set to a non-zero value, and sweep OFF mode is set, sound production may be stopped.
3. When a value is written to the envelope register, sound output becomes unstable before the initialization flag is set. Therefore, set initialization flag immediately after writing a value to the envelope register.
4. For sound 1, if you change the frequency when selecting a consecutive operation mode (sound length flag of NR14 is 0), always set 0 for the data of sound length (lower 6 bits of NR11) after setting the frequency data. If 0 is not set, sound may stop prematurely.
5. If the Sound 1 initialization flag is set when the sweep function is used, always set the initialization flag again after an interval of 8 clocks or more. Unless the initialization flag is set twice with an interval of 8 clocks or more, the sound may not be heard.

10.4 Sound 2

Sound 2 is a circuit that generates rectangular waveforms with envelope functions.

The contents of NR21, NR22, NR23, NR24 for Sound 2, conform with those of CGB.

Figure 68 - The SOUND2CNT_L Register



SOUND2CNT_L [d15 - 12] Envelope Initial-Value

Allows specification of any one of 16 levels ranging from maximum to mute.

SOUND2CNT_L [d11] Envelope Increase/Decrease

Specifies whether volume will increase or decrease.

SOUND2CNT_L [d10 - 08] Number of Envelope Steps

Sets the length of 1 step of envelope amplification or attenuation.

With n signifying the value specified, the length of 1 step (step time) is determined by the following formula.

Equation 6 - Determining the Length of 1 Step (steptime)

$$steptime = n \times \frac{1}{64}(\sec)$$

When n=0, the envelope function is turned off.

SOUND2CNT_L [d07 - 06] Waveform Duty Cycle

Specifies the proportion of waveform amplitude peaks.

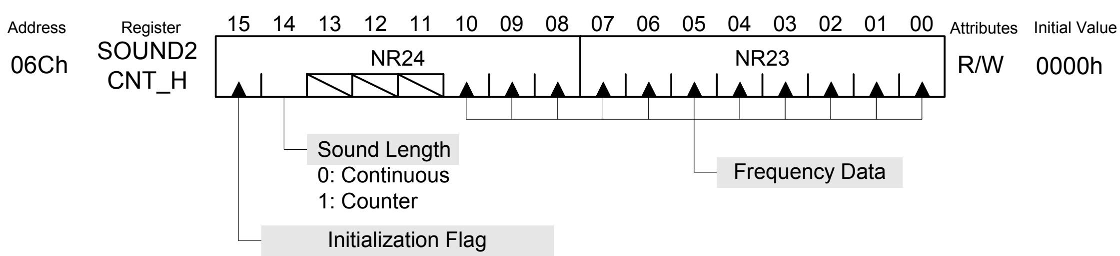
SOUND2CNT_L [d05 - 00] Sound Length

With st signifying the sound length data, the length of the output sound is determined by the following formula.

Equation 7 - Determining the Length of the Output Sound

$$time = (64 - st) \times \frac{1}{256}(\sec)$$

Figure 69 - The SOUND2CNT_H Register



SOUND2CNT_H [d15] Initialization Flag

A setting of 1 causes Sound 2 to be restarted.

SOUND2CNT_H [d14] Sound Length

Continuous sound output with 0; with 1, sound output only for the time specified in the sound length data of NR21.

When sound output ends, the Sound 2 ON flag of NR52 is reset.

SOUND2CNT_H [d10-00] Frequency Data

With fdat signifying the frequency data, the output frequency is determined by the following formula.

Equation 8 - Determining the Output Frequency

$$f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)}(Hz)$$

Thus, the frequency range that can be specified is 64 to 131.1 KHz.

Sound 2 Usage Notes

1. When a value is written to the envelope register, sound output becomes unstable before the initialization flag is set. Therefore, set initialization flag immediately after writing a value to the envelope register.
2. For sound 2, if you change the frequency when selecting a consecutive operation mode (Reset the sound length flag of NR24), always set 0 for the data of sound length (lower 6 bits of NR21) after setting the frequency data. If 0 is not set, sound may stop prematurely.

10.5 Sound 3

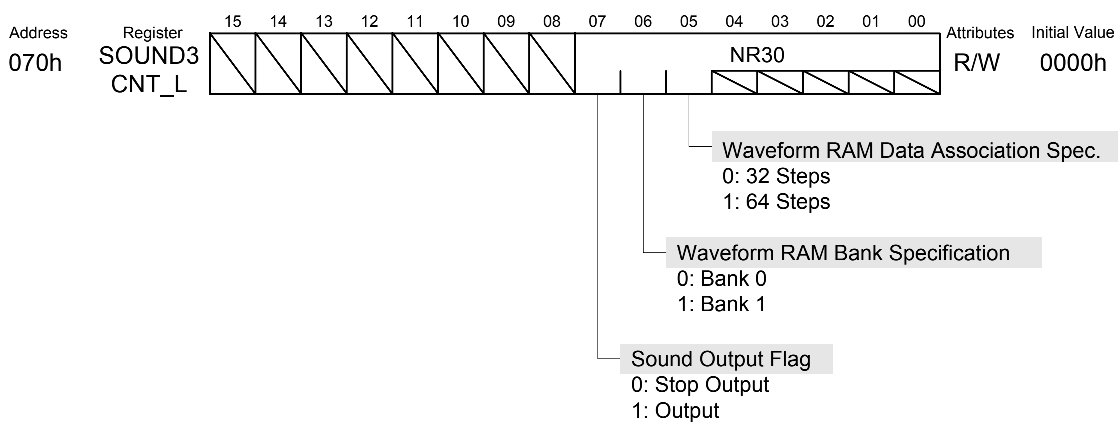
The Sound 3 circuit outputs arbitrary waveforms and can automatically read waveform patterns (1 cycle) in waveform RAM and output them while modifying their length, frequency, and level.

The capacity of the waveform RAM of Sound 3 in Game Boy Advance (total of 64 steps) is twice that in CGB, and can be used as 2 banks of 32 steps or as 64 steps.

In addition, a new output level of 3/4 output can now be selected.

The contents of NR30, NR31, NR32, NR33, NR34 for Sound 3, add the functionality listed above to those of CGB.

Figure 70 - The SOUND3CNT_L Register



SOUND3CNT_L [d07] Sound Output Flag

Sound output stops when 0; sound output occurs when 1.

SOUND3CNT_L [d06] Waveform RAM Bank Specification

Two banks of waveform RAM are provided, banks 0 and 1. The Sound 3 circuit plays the waveform data in the specified bank.

When waveform RAM is accessed by the user, the bank not specified is accessed.

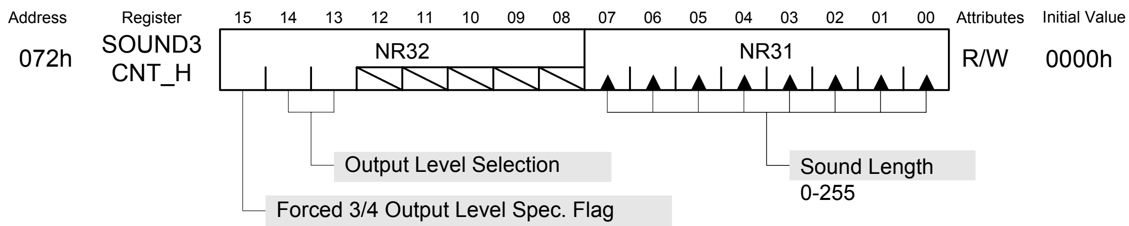
SOUND3CNT_L [d05] Waveform RAM Data Association Specification

When 0 is specified, 32-step waveform pattern is constructed under normal operation.

With a setting of 1, the data in the bank specified by NR30 [d06] (waveform RAM bank specification) is played, followed immediately by the data in the back bank.

The front bank 32 steps and the back bank 32 steps combine to form a waveform pattern with a total of 64 steps.

Figure 71 - The SOUND3CNT_H Register



SOUND3CNT_H [d15] Forced 3/4 Output Level Specification Flag

With 0 specified, the output level specified in NR32 [d14-13] is used.

A setting of 1 forces a 3/4 output level regardless of the setting in NR32 [d14-13].

SOUND3CNT_H [d14 - 13] Output Level Selection

The Sound 3 output-level selections are as shown in the following table.

Table 20 - Sound 3 Output Level Selections

Setting	Output Level
00	Mute
01	Outputs the waveform RAM data unmodified.
10	Outputs the waveform RAM data with the contents right-shifted 1 bit (1/2).
11	Outputs the waveform RAM data with the contents right-shifted 2 bits (1/4).

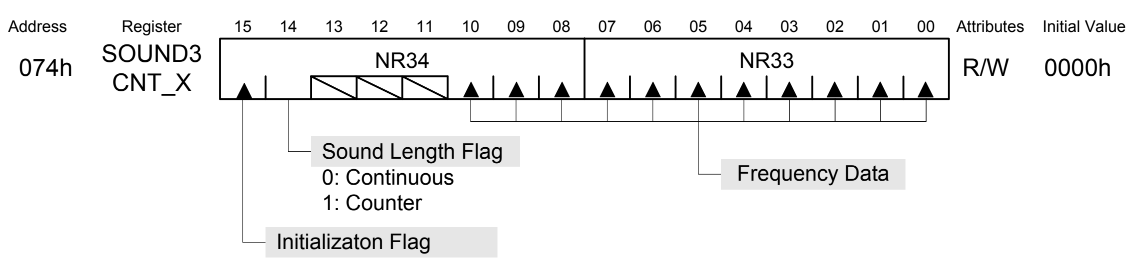
SOUND3CNT_H [d07-00] Sound Length

The sound length, *time*, is determined by the following formula, with *st* signifying the sound-length setting.

Equation 9 - Determining the Length of the Output Sound

$$time = (256 - st) \times \frac{1}{256}(sec)$$

Figure 72 - The SOUND3CNT_X Register



SOUND3CNT_X [d15] Initialization Flag

When SOUND3CNT_L [d07] is 1, a setting of 1 in this bit causes Sound 3 to restart.

SOUND3CNT_X [d14] Sound Length Flag

When 0, sound is continuously output.

When 1, sound is output for only the length of time specified for the sound length in NR31.

When sound output ends, the Sound 2 ON flag of NR52 is reset.

SOUND3CNT_X [d10 - 00] Frequency Data

With *fdat* signifying the frequency, the output frequency (*f*) is determined by the following formula.

Equation 10 - Determining the Output Frequency

$$f = \frac{4194304}{4 \times 2^3 \times (2048 - fdat)} (Hz)$$

Thus, the specifiable range of frequencies is 64 to 131.1 KHz.

Sound 3 Usage Notes

1. When changing the frequency during Sound 3 output, do not set the initialization flag. The contents of waveform RAM may be corrupted. With sounds 1, 2, and 4, the initialization flag can be set without problems.
2. For sound 3, if you change the frequency when selecting a consecutive operation mode (Reset the sound length flag of NR34), always set 0 for the data of sound length (NR31) after setting the frequency data. If 0 is not set, sound may stop prematurely.

10.5.1 Waveform RAM

Waveform RAM consists of a 4-bit x 32-step waveform pattern. It has 2 banks, with [d06] of SOUND3CNT_L used for bank specification.

The Sound 3 circuit plays the waveform data specified by the bank setting, while the waveform RAM not specified is the waveform RAM accessed by the user.

Figure 73 - Waveform RAM Registers

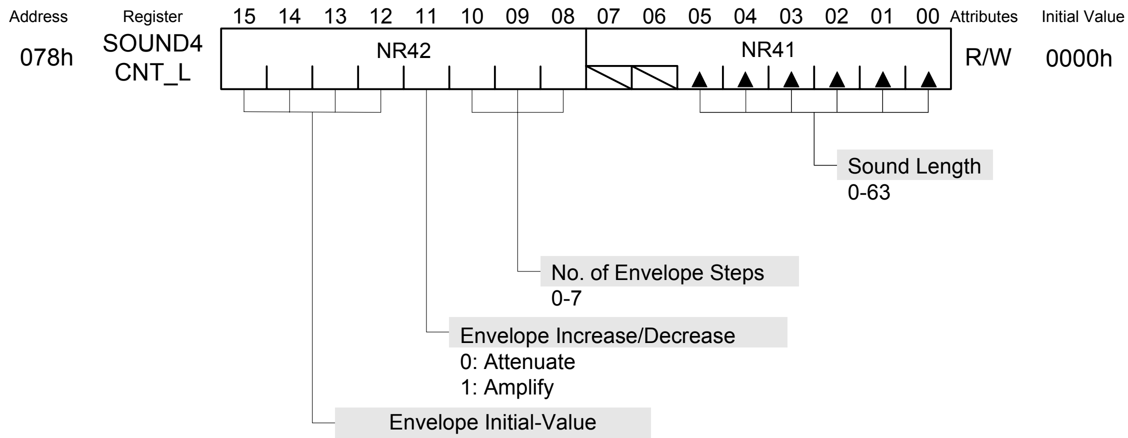
Address	Register	15–12	11–08	07–04	03–00	Attributes	Initial Value
090h	WAVE_RAM0_L	Step 2	Step 3	Step 0	Step 1	R/W	-
092h	WAVE_RAM0_H	Step 6	Step 7	Step 4	Step 5	R/W	-
094h	WAVE_RAM1_L	Step 10	Step 11	Step 8	Step 9	R/W	-
096h	WAVE_RAM1_H	Step 14	Step 15	Step 12	Step 13	R/W	-
098h	WAVE_RAM2_L	Step 18	Step 19	Step 16	Step 17	R/W	-
09Ah	WAVE_RAM2_H	Step 22	Step 23	Step 20	Step 21	R/W	-
09Ch	WAVE_RAM3_L	Step 26	Step 27	Step 24	Step 25	R/W	-
09Eh	WAVE_RAM3_H	Step 30	Step 31	Step 28	Step 29	R/W	-

10.6 Sound 4

Sound 4 is a circuit that generates white noise with the envelope function.

The contents of NR41, NR42, NR43, and NR44 for Sound 4 conform with those of CGB.

Figure 74 - The SOUND4CNT_L Register



SOUND4CNT_L [d15 - 12] Envelope Initial-Value

Allows specification of any of 16 levels ranging from maximum to mute.

SOUND4CNT_L [d11] Envelope Increase/Decrease

Specifies whether to increase or decrease the volume.

SOUND4CNT_L [d10 - 08] Number of Envelope Steps

Sets the length of each step of envelope amplification or attenuation.

With n the specified value, the length of 1 step (steptime) is determined by the following formula.

Equation 11 - Determining the Length of 1 Step (steptime)

$$steptime = n \times \frac{1}{64}(\mathrm{sec})$$

When n = 0, the envelope function is turned off.

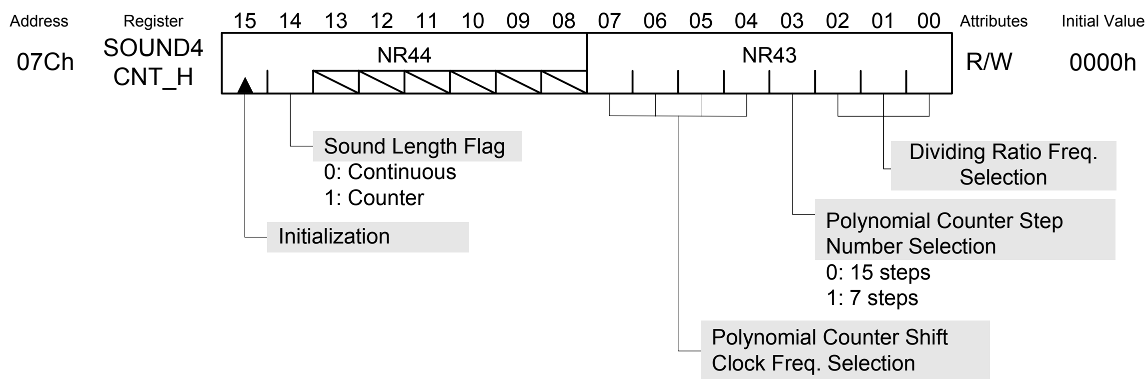
SOUND4CNT_L [d05 - 00] Sound Length

With st signifying the sound length, the length of the output sound is determined by the following formula.

Equation 12 - Determining the Length of the Output Sound

$$time = (64 - st) \times \frac{1}{256}(\mathrm{sec})$$

Figure 75 - The SOUND4CNT_H Register



SOUND4CNT_H [d15] Initialization Flag

A setting of 1 causes Sound 4 to be restarted.

SOUND4CNT_H [d14] Sound Length

Continuous sound output with 0; with 1, sound output only for the time specified in the sound length data of NR41.

When sound output ends, the Sound 4 ON flag of NR52 is reset.

SOUND4CNT_H [d07 - 04] Polynomial Counter Shift Clock Frequency Selection

With n signifying the specified value, the shift clock frequency (shiftfreq) is selected as shown in the following formula.

Equation 13 - Selecting the Shift Clock Frequency

$$shiftfreq = \text{dividing ratio frequency} \times \frac{1}{2^{(n+1)}}$$

However, %1110 and %1111 are prohibited codes.

SOUND4CNT_H [d03] Polynomial Counter Step Number Selection

A value of 0 selects 15 steps; 1 selects 7 steps.

SOUND4CNT_H [d02 - 00] Dividing Ratio Frequency Selection

Selects a 14-step prescalar input clock to produce the shift clock for the polynomial counter.

With f=4.194304 MHz, selection is as shown in the following table.

Table 21 - Sound 4 Prescalar Input Clock Selection

Setting	Dividing Ratio Frequency
000	$fx1/2^3x2$
001	$fx1/2^3x1$
010	$fx1/2^3x(1/2)$
011	$fx1/2^3x(1/3)$
100	$fx1/2^3x(1/4)$
101	$fx1/2^3x(1/5)$
110	$fx1/2^3x(1/6)$
111	$fx1/2^3x(1/7)$

Sound 4 Usage Note

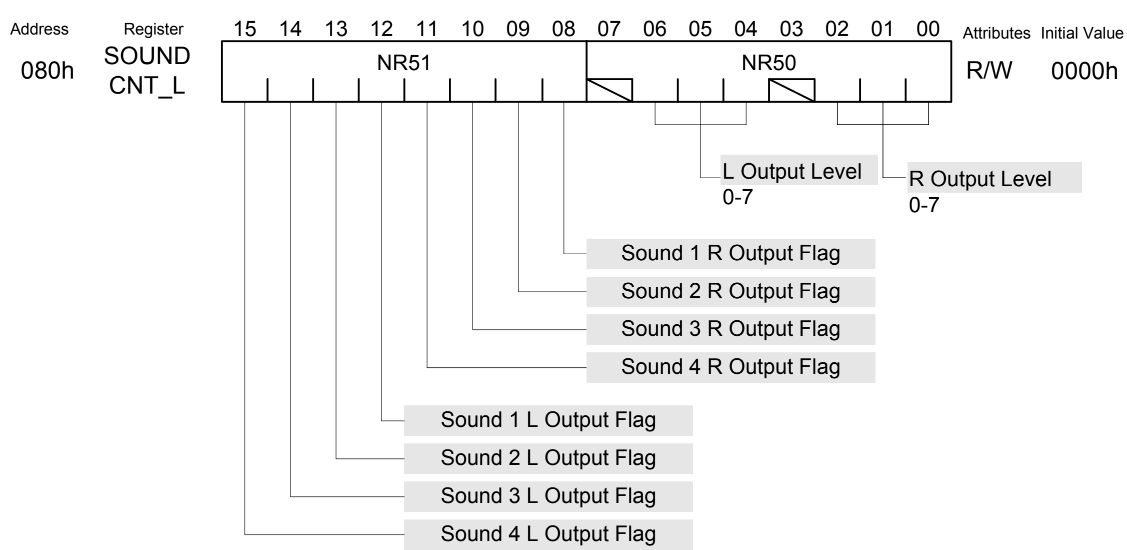
When a value is written to the envelope register, sound output becomes unstable before the initialization flag is set. Therefore, set initialization flag immediately after writing a value to the envelope register.

10.7 Sound Control

The output ratio for direct sound and sound can be set using the SOUNDCNT_H register. Final sound control can be achieved with the SOUNDCNT_L register.

NR50 and NR51 are each based on their counterparts in CGB.

Figure 76 - The SOUNDCNT_L Register



SOUNDCNT_L [d15 - 12] L Output Flag for each Sound

No output of that sound to L when 0.

Output of that sound to L when 1.

SOUNDCNT_L [d11 - 08] R Output Flag for each Sound

No output of that sound to R when 0.

Output of that sound to R when 1.

SOUNDCNT_L [d06 - 04] L Output Level

L output level can be set to any of 8 levels.

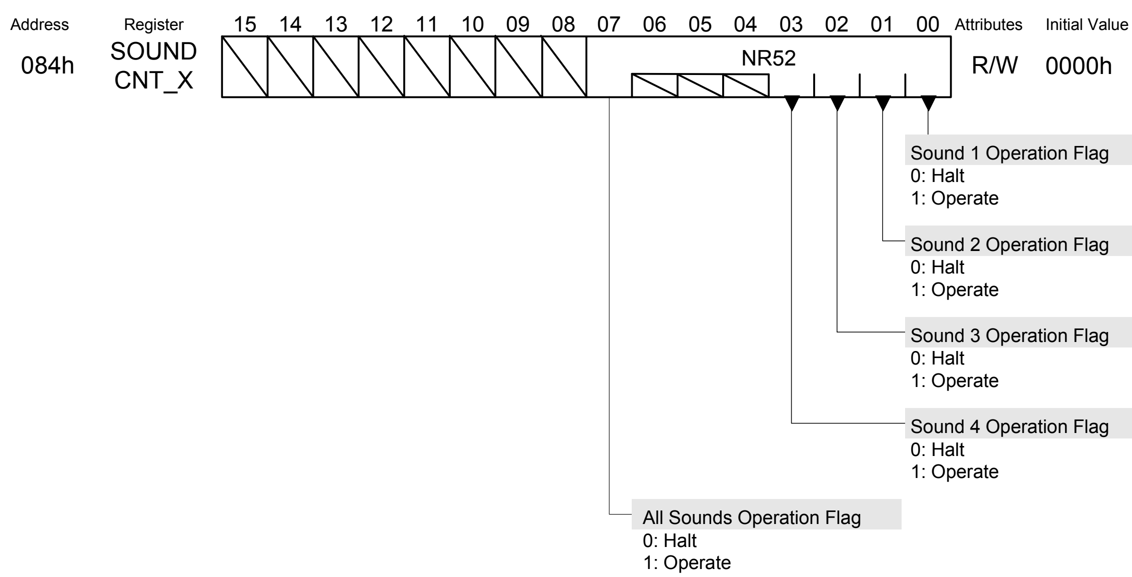
However, there is no effect on direct sound.

SOUNDCNT_L [d02 - 00] R Output Level

R output level can be set to any of 8 levels.

However, there is no effect on direct sound.

Figure 77 - The SOUNDCNT_X Register



SOUNDCNT_X [d07] All Sounds Operation Flag

The master flag that controls whether sound functions as a whole are operating.

A setting of 0 halts all sound functions including direct sound, producing a mute state.

In this situation, the contents of all the Sound mode registers are reset.

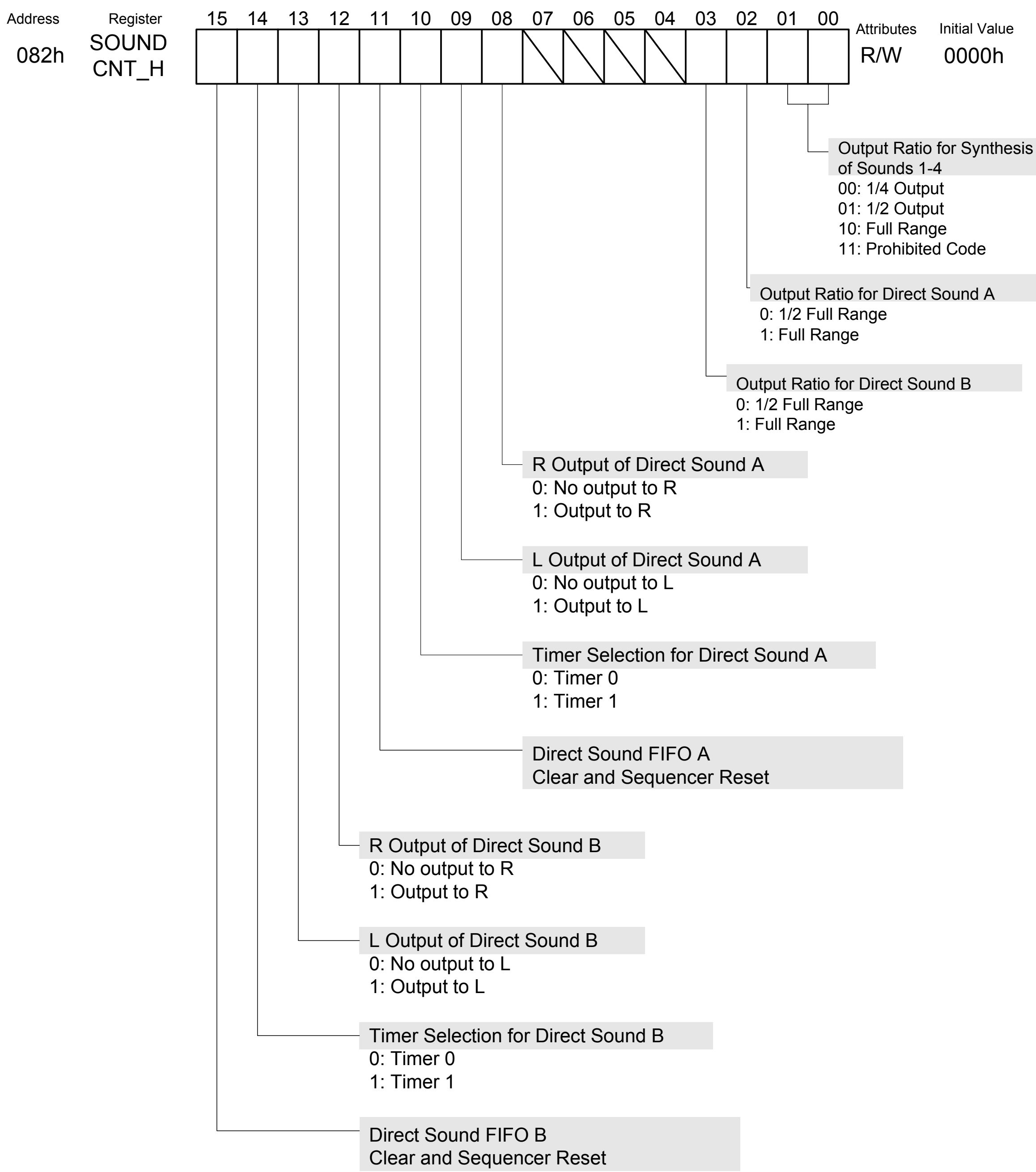
Note: Always set all the sound operation flags to 1 when setting each sound mode register. You cannot set each sound mode register when all the sound is stopped.

SOUNDCNT_X [d03, d02, d01, d00] Sound Operation Flags

Each sound circuit's status can be referenced.

Each sound is set during output, and when in counter mode it is reset after the time passes which was set up with the length data.

Figure 78 - The SOUNDCNT_H Register



SOUNDCNT_H [d15],[d11] FIFO Clear and Sequencer Reset for Each Direct Sound

With direct sound the sequencer counts the number of times data is transmitted from FIFO to the mixing circuit. A setting of 1 resets the FIFO and sequencer used for each direct sound. When this bit is read, 0 is returned.

SOUNDCNT_H [d14],[d10] Timer Selection for Each Direct Sound

Specifies the timer used for each direct sound.

A setting of 0 selects timer 0, and 1 selects timer 1.

The same timer can be specified for both direct sounds (A and B).

SOUNDCNT_H [d13],[d09] L Output for Each Direct Sound

Controls the output to L for each direct sound. A setting of 0 results in no output to L; a setting of 1 causes output to L.

SOUNDCNT_H [d12],[d08] R Output for Each Direct Sound

Controls the output to R for each direct sound. A setting of 0 results in no output to R; a setting of 1 causes output to R.

SOUNDCNT_H [d03],[d02] Output Ratio for Each Direct Sound

Selects the output level for each direct sound.

A setting of 0 produces output that is 1/2 of full range. A setting of 1 results in full-range output.

SOUNDCNT_H [d01 - 00] Output Ratio for Synthesis of Sounds 1-4

Specifies the output level for the synthesis of sounds 1-4.

A setting of 00 results in output that is 1/4 of full range.

A setting of 01 results in output that is 1/2 of full range.

A setting of 10 results in full-range output.

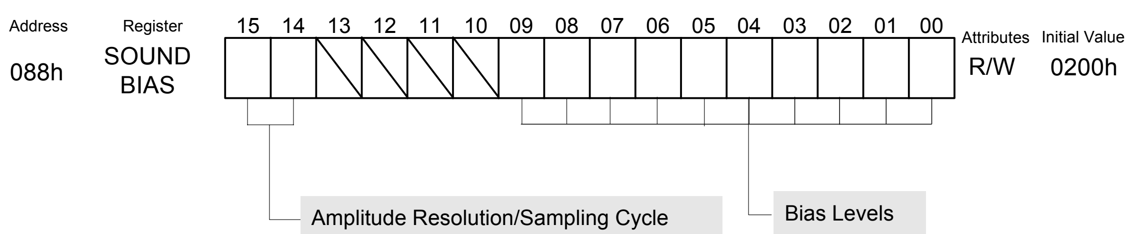
A setting of 11 is a prohibited code.

10.8 Sound PWM Control

Bit modulation format PWM is used in the Game Boy Advance sound circuit. When no sound is produced, the duty waveform is output, and bias voltage is provided. The PWM circuit is stopped when the setting for duty is 0h.

Note: The bias level uses system ROM. This can be the cause of errors, therefore be careful not to change the bias level value.

Figure 79 - The SOUNDBIAS Register



SOUNDBIAS [d15 - 14] Amplitude Resolution/Sampling Cycle

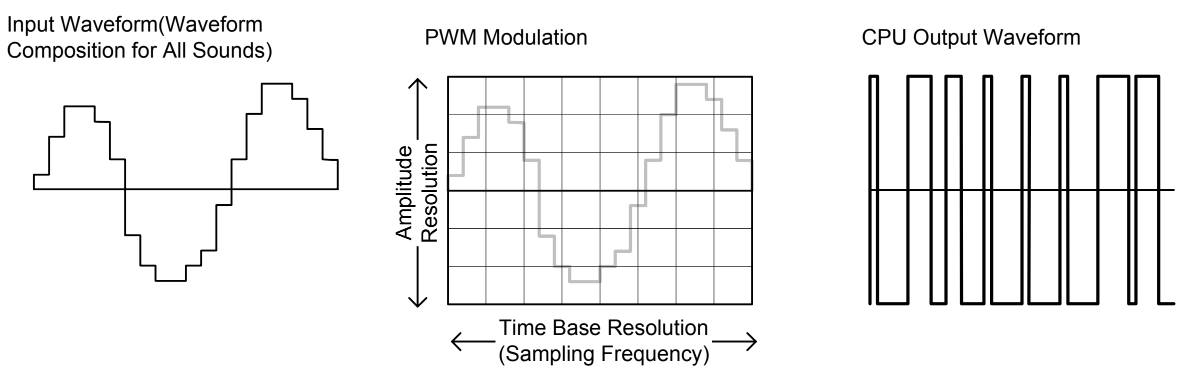
This sets the amplitude resolution and sampling cycle frequency during PWM modulation.

The DMG compatible sound is input at 4 bits/130.93KHz so in order to have accurate modulation the sampling frequency must be set high. Direct sound will arbitrarily decide the sampling frequency based on the timer setting. By using the sampling frequencies listed in the table below, an accurate modulation can be done. Thus, in order to increase authenticity of sound, the amplitude resolution needs to be set higher. When producing both compatible sound and direct sound find a value that will work for both and set this.

Table 22 - PWM Modulation Amplitude Resolution and Sampling Cycle Frequency

Setting	Amplitude Resolution	Sampling Frequency
00	9-bit	32.768 KHz
01	8-bit	65.536 KHz
10	7-bit	131.072 KHz
11	6-bit	262.144 KHz

Figure 80 - PWM Conversion Image



SOUNDBIAS [d09 - 00] Bias Level

This is used by system ROM. Please do not change this value, as it may cause errors.

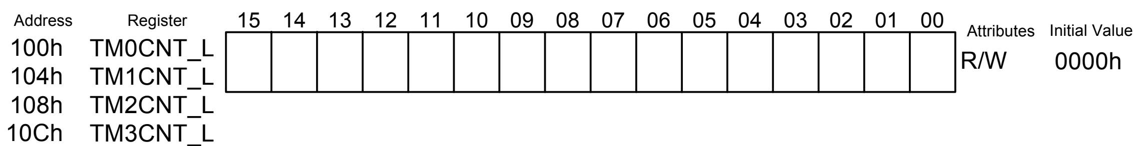
11 Timer

Game Boy Advance is equipped with 4 channels of 16-bit timers.

Of these, timers 0 and 1 can be used to set the interval for the supply of data from the FIFO(s) for direct sounds A and B. This interval is set by timer overflow.

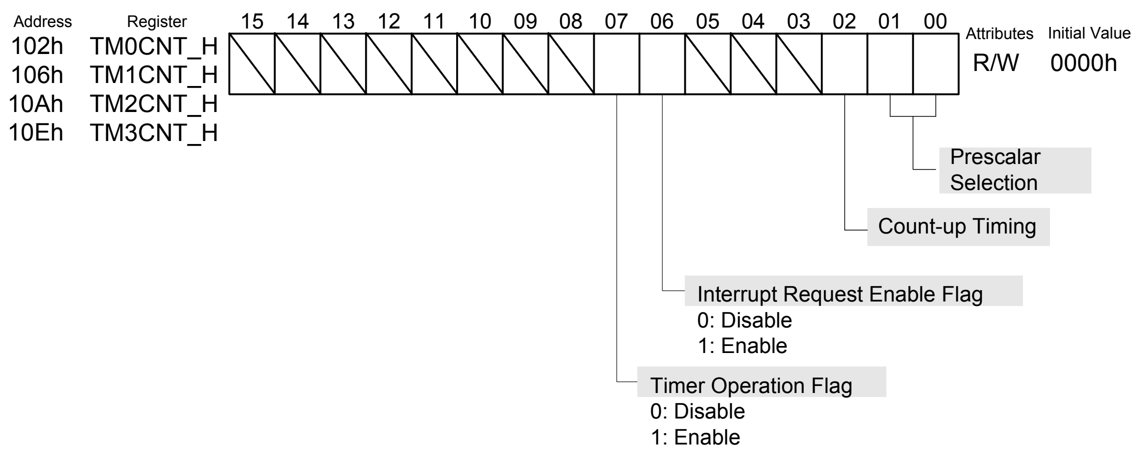
11.1 Timer Setting

Figure 81 - Timer Setting Registers



11.2 Timer Control

Figure 82 - Timer Control Registers



TM*CNT_H [d07] Timer Operation Flag

Starts and stops the timer. A setting of 0 stops the timer, and a setting of 1 starts it.

TM*CNT_H [d06] Interrupt Request Enable Flag

Controls whether an interrupt request flag is generated by an overflow. No interrupt is generated with a setting of 0. An overflow does generate an interrupt if the setting is 1.

TM*CNT_H [d02] Count-Up Timing

With a setting of 0, count-up is performed in accordance with the prescalar specification in [d01-00]. With a setting of 1, overflow of the timer channel one number lower starts a count-up regardless of the prescalar specification.

This mode is suitable for purposes such as time measurement over relatively long periods.

The count-up timing specification is disabled for Timer 0, which counts up in accordance with the prescalar specification.

TM*CNT_H [d01 - 00] Prescalar Selection

Allows selection of a prescalar based on the system clock (16.78MHz).

Table 23 - Timer Control Prescalar Selection

Setting	Prescalar (Count-Up Interval)	
00	System clock	(59.595 ns)
01	64 cycles of system clock	(3.814 μs)
10	256 cycles of system clock	(15.256 μs)
11	1024 cycles of system clock	(61.025 μs)

12 DMA Transfer

DMA uses the DMA controller to transfer data at a high speed between memories without going through the Game Boy Advance CPU.

(In order to prevent conflict with the external bus, the CPU stops when the DMA controller is working.)

Game Boy Advance has 4 DMA transfer channels.

The highest priority of these channels is DMA0, followed in order by DMA1, DMA2, and DMA3.

If a DMA with a higher priority than the currently executing DMA begins execution, the execution of the current DMA is temporarily halted, and the DMA with the higher priority is executed. Once this DMA finishes, the original DMA resumes execution from where it was halted.

Thus, the most appropriate uses of each DMA channel are those described below.

- DMA 0

  Because this has the highest priority, it is not interrupted by other DMA channels. Thus, it is used for reliable processing over a limited period, as is required for purposes such as horizontal-blanking DMA.

- DMA 1 and DMA 2

  These are used for direct sound functions, which require relatively high priority, or for general-purpose transfers.

- DMA 3

  This is used for the most general types of transfers.

Perform the following settings when using DMA.

1. Specify the transfer source address in the source address register.
2. Specify the transfer destination address in the destination address register.
3. Set the number of data items in the word-count register.
4. Specify the transfer method to be used in the DMA control register.

Cautions for DMA

When transferring data to OAM or OBJ VRAM by DMA during H-blanking, the H-blank must first be freed from OBJ display hardware processing periods using the DISPCNT register. (See "5 Image System" on page 19.)

12.1 DMA 0

DMA 0 allows different areas of internal memory in the main unit to access one another. It has the highest priority of the DMA channels.

12.1.1 Source Address

Specifies the source address using 27 bits.

The area 00000000h-07FFFFFFh (internal memory area of main unit) can be specified.

Figure 83 - DMA 0 Source Address Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0B0h	DMA0 SAD_L																	W	0000h
0B2h	DMA0 SAD_H																	W	0000h

12.1.2 Destination Address

Specifies the destination address using 27 bits.

The area 00000000h-07FFFFFFh (internal memory area of main unit) can be specified.

Figure 84 - DMA 0 Destination Address Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0B4h	DMA0 DAD_L																	W	0000h
0B6h	DMA0 DAD_H																	W	0000h

12.1.3 Word Count

Specifies the number of bytes transferred by DMA0, using 14 bits. The number can be specified in the range 0001h~3FFFh~0000h (when 0000h is set, 4000h bytes are transferred).

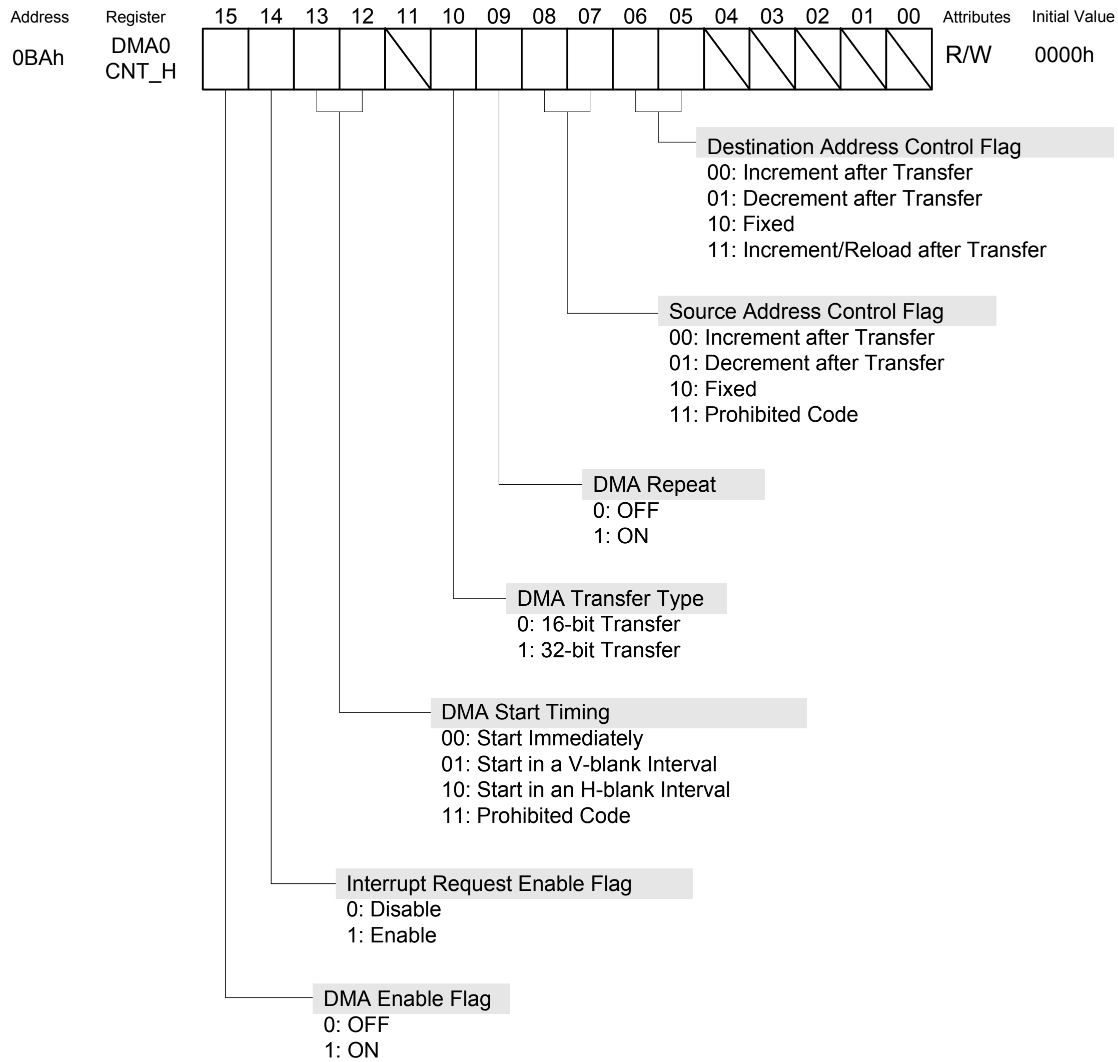
Thus, in 16-bit data transfer mode, up to 4000h x 2=8000h bytes can be transferred, and in 32-bit data transfer mode, up to 4000h x 4=10000h bytes can be transferred.

Figure 85 - The DMA0CNT_L Register

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0B8h	DMA0 CNT_L																	W	0000h

12.1.4 DMA Control

Figure 86 - The DMA0CNT_H Register



DMA0CNT_H [d15] DMA Enable Flag

A setting of 0 disables DMA.

A setting of 1 enables DMA, and after the transfer is completed the source and destination registers are restored to their last values.

Note: Delay of 2 waits will occur before DMA is activated after this flag is set. Accessing DMA related registers during this time may cause a DMA malfunction. Do another process or insert a dummy load command instead.

DMA0CNT_H [d14] Interrupt Request Enable Flag

Enables an interrupt request to be generated when DMA transfer of the specified word count has been completed.

No request is generated with a setting of 0; a request is generated with a setting of 1.

DMA0CNT_H [d13 - 12] DMA Startup Timing

The timing of the DMA transfer can be selected from the following options.

Table 24 - DMA Transfer Timing Selections (DMA 0)

Setting	DMA Startup Timing
00	**Start immediately**
01	**Start during a V-blanking interval** Starts at the beginning of a V-blanking interval (approximately 4.993 ms).
10	**Start during a H-blanking interval** Starts at the beginning of a H-blanking interval (approximately 16.212 μs). If this accompanies OAM access, the H-blanking interval must first be freed of OBJ display hardware processing periods. (See "5 Image System" on page 19.)
11	**Prohibited Code**

DMA0CNT_H [d10] DMA Transfer Type

Sets the bit length of the transfer data.

With a setting of 0, the data are transferred by DMA in 16-bit (half-word) units. With a setting of 1, the data are transferred by DMA in 32-bit (word) units.

DMA0CNT_H [d09] DMA Repeat

With the DMA repeat function set to ON, if V-blanking or H-blanking intervals are selected as the timing of DMA startup, DMA is restarted when the next startup condition occurs (a V-blank or H-blank).

Note: In this mode, restarting will continue as long as the DMA enable flag is not set to 0.

When the DMA repeat function is set to OFF, DMA halts as soon as the amount of data specified by the value in the word-count register has been transferred.

DMA0CNT_H [d08] Source Address Control Flag

Control of the source address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

11 is a prohibited code.

DMA0CNT_H [d07] Destination Address Control Flag

Control of the destination address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

A setting of 11 causes an increment and after all transfers end, a reload (the setting is returned to what it was when the transfer started) is performed.

12.2 DMA 1 and 2

DMA channels 1 and 2 provide access between the Game Pak bus/internal memory of the main unit and internal memory of the main unit, or between the Game Pak bus/internal memory of the main unit and the direct sound FIFO. Transfers to direct-sound FIFO can be accomplished only by using DMA 1 and 2.

12.2.1 Source Address

Specifies the source address using 28 bits.

The area 00000000h-0FFFFFFFh can be specified.

Figure 87 - DMA 1 and 2 Source Address Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0BCh 0C8h	DMA1SAD_L DMA2SAD_L																	W	0000h

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0BEh 0CAh	DMA1SAD_H DMA2SAD_H																	W	0000h

12.2.2 Destination Address

Specifies the destination address using 27 bits.

The area 00000000h-07FFFFFFh (internal memory area of main unit) can be specified.

Figure 88 - DMA 1 and 2 Destination Address Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0C0h 0CCh	DMA1DAD_L DMA2DAD_L																	W	0000h

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0C2h 0CEh	DMA1DAD_H DMA2DAD_H																	W	0000h

12.2.3 Word Count

Specifies the number of bytes transferred by DMA 1 and DMA 2, using 14 bits. The number can be specified in the range 0001h~3FFFh~0000h (when 0000h is set, 4000h bytes are transferred).

Thus, in 16-bit data transfer mode, up to 4000h x 2=8000h bytes can be transferred, and in 32-bit data transfer mode, up to 4000h x 4=10000h bytes can be transferred.

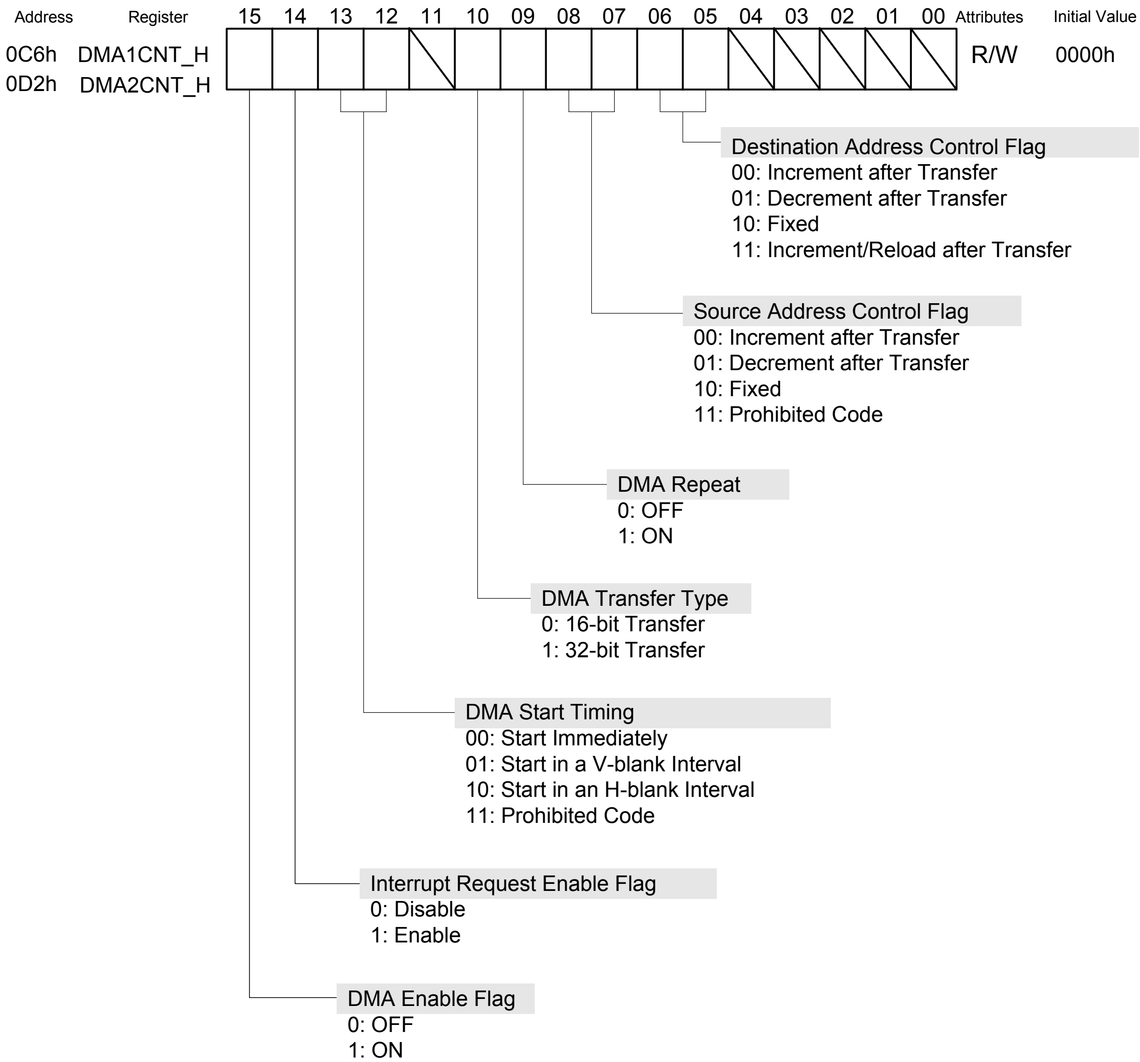
Figure 89 - DMA 1 and 2 Word Count Registers

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0C4h 0D0h	DMA1CNT_L DMA2CNT_L																	W	0000h

The word-count register setting is disabled in direct-sound FIFO transfer mode. With each request received from sound FIFO, 32 bits x 4 words of sound data are transferred.

12.2.4 DMA Control

Figure 90 - DMA 1 and 2 Control Registers



DMA(1,2)CNT_H [d15] DMA Enable Flag

A setting of 0 disables the DMA function.

A setting of 1 enables DMA, and after the transfer is completed the source and destination registers are restored to their last values.

Note: Delay of 2 waits will occur before DMA is activated after this flag is set. Accessing DMA related registers during this time may cause a DMA malfunction. Do another process or insert a dummy load command instead.

DMA(1,2)CNT_H [d14] Interrupt Request Enable Flag

Enables an interrupt request to be generated when DMA transfer of the specified word count has been completed.

No request is generated with a setting of 0; a request is generated with a setting of 1.

DMA(1,2)CNT_H [d13-12] DMA Startup Timing

The timing of the DMA transfer can be selected from the following options.

Table 25 - DMA Transfer Timing Selections (DMA 1 and 2)

Setting	DMA Startup Timing
00	**Start Immediately**
01	**Start During a V-blanking interval** Starts at the beginning of a V-blanking interval (approximately 4.993 ms).
10	**Start During a H-blanking interval** Starts at the beginning of a H-blanking interval (approximately 16.212 µs). If this accompanies OAM access, the H-blanking interval must first be freed of OBJ display hardware processing periods. (See "5 Image System" on page 19.)
11	**Start When Request Generated by Direct-Sound FIFO** Starts when a request is received form direct-sound FIFO. Specify sound FIFO as the destination address. Also, set the DMA repeat function [d09] to ON.

DMA(1,2)CNT_H [d10] DMA Transfer Type

Sets the bit length of the transfer data.

With a setting of 0, the data are transferred by DMA in 16-bit (half-word) units. With a setting of 1, the data are transferred by DMA in 32-bit (word) units.

In direct-sound FIFO transfer mode, the data are transferred in 32-bit units.

DMA(1,2)CNT_H [d09] DMA Repeat

With the DMA repeat function set to ON, if V-blanking or H-blanking intervals are selected as the timing of DMA startup, DMA is restarted when the next startup condition occurs (a V-blank or H-blank).

Note: In this mode, restarting will continue as long as the DMA enable flag is not set to 0.

When the DMA repeat function is set to OFF, DMA halts as soon as the amount of data specified by the value in the word-count register has been transferred.

Note: Set this bit to 1 in direct-sound FIFO transfer mode.

DMA(1,2)CNT_H [d08] Source Address Control Flag

Control of the source address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

11 is a prohibited code.

Note: When the Game Pak Bus has been set to the source address, make sure you select increment.

DMA(1,2)CNT_H [d07] Destination Address Control Flag

Control of the destination address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

A setting of 11 causes an increment to be carried out and then a reload (returned to setting at start of transfer) is performed after every transfer is completed.

However, when in direct sound FIFO transfer mode, the destination address is fixed and unrelated to the setting.

12.3 DMA 3

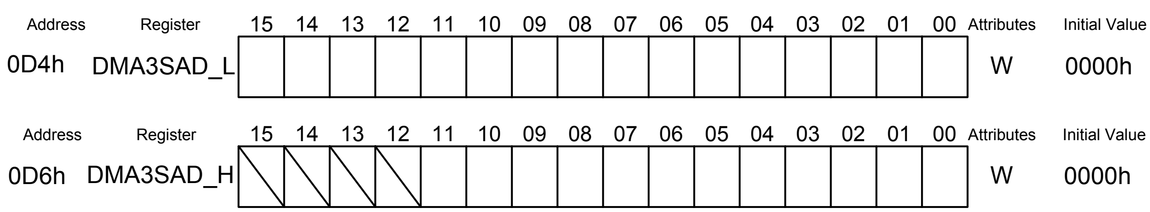
DMA 3 provides memory access between the Game Pak bus and internal memory of the main unit, or between different areas of internal memory of the main unit.

12.3.1 Source Address

Specifies the source address using 28 bits.

The area 00000000h-0FFFFFFFh (internal memory of main unit and Game Pak memory area) can be specified.

Figure 91 - DMA 3 Source Address Registers

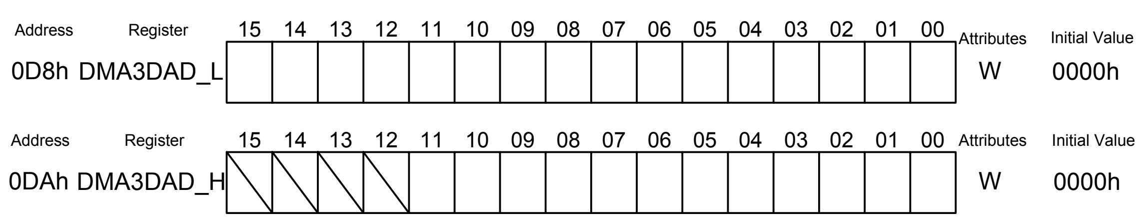


12.3.2 Destination Address

Specifies the destination address using 28 bits.

The area 00000000h-0FFFFFFFh (internal memory area of main unit and Game Pak memory area) can be specified.

Figure 92 - DMA 3 Destination Address Registers



12.3.3 Word Count

Specifies the number of bytes transferred by DMA 3, using 16 bits. The number can be specified in the range 0001h~FFFFh~0000h (when 0000h is set, 10000h bytes are transferred).

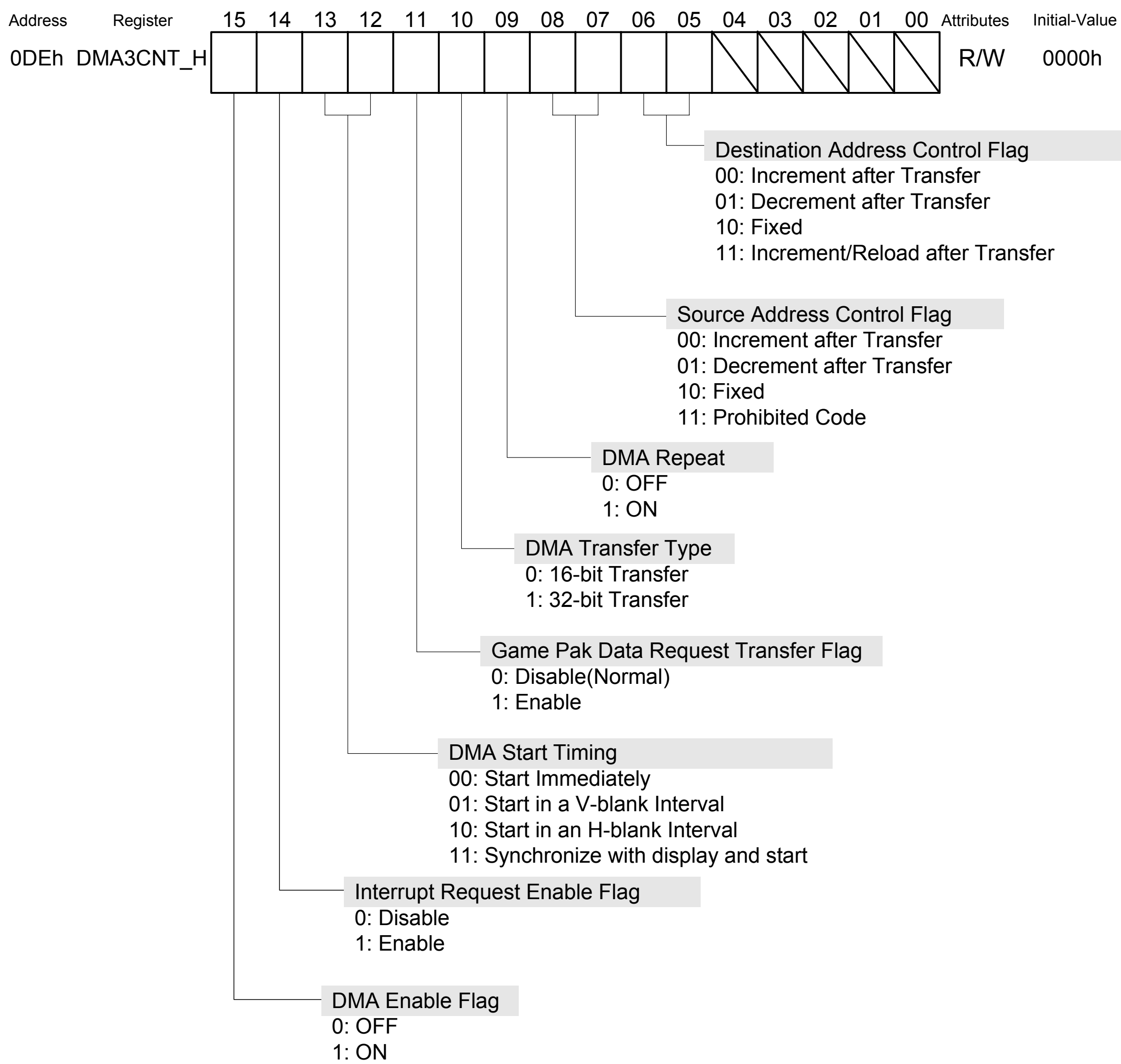
Thus, in 16-bit data transfer mode, up to 10000h x 2=20000h bytes can be transferred, and in 32-bit data transfer mode, up to 10000h x 4=40000h bytes can be transferred.

Figure 93 - The DMA3CNT_L Register

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
0DCh	DMA3CNT_L																	W	0000h

12.3.4 DMA Control

Figure 94 - The DMA3CNT_H Register



DMA3CNT_H [d15] DMA Enable Flag

A setting of 0 disables DMA.

A setting of 1 enables DMA, and after the transfer is completed the source and destination registers are restored to their last values.

Note: Delay of 2 waits will occur before DMA is activated after this flag is set. Accessing DMA related registers during this time may cause a DMA malfunction. Do another process or insert a dummy load command instead.

DMA3CNT_H [d14] Interrupt Request Enable Flag

Enables an interrupt request to be generated when DMA transfer of the specified word count has been completed.

No request is generated with a setting of 0; a request is generated with a setting of 1.

DMA3CNT_H [d13-12] DMA Startup Timing

The timing of the DMA transfer can selected from the following options.

Table 26 - DMA Transfer Timing Selections (DMA 3)

Setting	DMA Startup Timing
00	**Start Immediately**
01	**Start During a V-blanking Interval** Starts at the beginning of a V-blanking interval (approximately 4.993 ms).
10	**Start During a H-blanking Interval** Starts at the beginning of a H-blanking interval (approximately 16.212 μs). If this accompanies OAM access, the H-blanking interval must first be freed of OBJ display hardware processing periods. (See "5 Image System" on page 19.)
11	**Synchronize with display and start.** Synchronize with start of H-Line rendering during a display interval and start.

DMA3CNT_H [d11] Game Pak Data Request Transfer Flag

Should normally be set to 0.

When set to 1, DMA transfer is performed in response to a data request from the Game Pak.

Note: A Game Pak that supports this transfer mode is required in order to use it. In addition, it cannot be used at the same time as a Game Pak interrupt.

DMA3CNT_H [d10] DMA Transfer Type

Sets the bit length of the transfer data.

With a setting of 0, the data are transferred by DMA in 16-bit (half-word) units. With a setting of 1, the data are transferred by DMA in 32-bit (word) units.

DMA3CNT_H [d09] DMA Repeat

With the DMA repeat function set to ON, if V-blanking or H-blanking intervals are selected as the timing of DMA startup, DMA is restarted when the next startup condition occurs (a V-blank or H-blank).

Note: In this mode, restarting will continue as long as the DMA enable flag is not set to 0.

When the DMA repeat function is set to OFF, DMA halts as soon as the amount of data specified by the value in the word-count register has been transferred.

However, in Game Pak data request mode do not use the repeat function.

DMA3CNT_H [d08] Source Address Control Flag

Control of the source address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

11 is a prohibited code.

Note: When the Game Pak Bus has been set to the source address, make sure you select increment.

DMA3CNT_H [d07] Destination Address Control Flag

Control of the destination address is specified after each DMA transfer.

A setting of 00 causes an increment.

A setting of 01 causes a decrement.

A setting of 10 causes it to be fixed.

A setting of 11 causes an increment to be carried out and then a reload (returned to setting at start of transfer) is performed after every transfer is completed.

12.3.5 Display Synchronization DMA

This function is used to transfer frame data from peripheral equipment, such as a camera, to a frame buffer in BG mode 3. Since BG mode 3 has only one frame buffer, this function is designed so that the next frame data transfer will not overwrite the current screen data that is displayed.

When transferring one (1) frame of data composed of 240 x 160 pixels with 32,768 colors (16-bit/pixel), use the following settings:

- Word count register

Word count	The number of transfers per horizontal line (For 32-bit DMA transfer, set to 78h).

- DMA control register

DMA repeat	:1

You can enable this DMA anytime. Set the DMA enable flag to 1 after making the above settings. If the DMA enable flag is 1 when the V count value is 162, DMA transfers will be executed in the next frame. Synchronizing with the horizontal line, DMA, which transfers the "word count" data per horizontal line, will be executed 160 times, from line 2 to line 161.

Data is always DMA transferred to the frame buffer address located 2 horizontal lines before the line being drawn, so currently displayed graphics will never be affected by transferred data. When the V count value becomes 162, the DMA enable flag is reset to 0 automatically and the DMA stops.

If the DMA enable flag is cleared manually, there is a possibility of a malfunction. Always wait until the DMA enable flag is reset to 0.

Although the DMA repeat flag is ON, this DMA will be disabled after the transfer of 1 frame's worth of data. Therefore, it is necessary to re-enable the DMA enable flag for every frame to be transferred.

12.4 DMA Problems: How to Avoid Them

With DMA transfer it is possible to synchronize with H-blank, V-blank, Direct sound (DMA1,2), and Display (DMA3) (DMA repeat). However, there are some problems with this function as discussed below.

With a DMA repeat, the DMA begins when the start trigger is sent. When the word count's data transfer is finished, DMA stops is repeated.

If the DMA enable flag is cleared by the CPU at the same time as the DMA start trigger, the DMA locks up. Therefore, be careful when stopping the DMA during a DMA repeat.

12.4.1 When the DMA Repeat Function is not Used

The DMA automatically stops after it has been executed one time, so do not clear the DMA-enabled flag with the user program until it becomes 0.

12.4.2 When the DMA Repeat Function is Used

Maintain a spacing of 4 clocks or more between the DMA start trigger and the timing to clear the DMA-enabled flag by the CPU. For example, it is possible to stop the DMA safely by clearing an enabled flag before the next start trigger is sent by using an interrupt that occurs at the end of the DMA. When this method cannot be used, stop the DMA as shown below.

12.4.2.1 How to Stop DMA Repeat in H-blank and V-blank Mode

DMA is not in progress and the DMA start trigger is not sent during the V-blank. Therefore, you can clear a DMA enable flag safely. If this method cannot be used, follow the procedures shown below.

(1) Write the following settings in 16-bit width to the DMA control register:

- DMA-enabled flag: 1 (Enabled)
- DMA start timing: 00 (Start Immediately)
- Data request transfer flag of the Game Pak side: 0 (Disabled) (DMA3 only)
- DMA repeat: 0 (OFF)
- Other control bits: No change

(2) Run a process for 4 clocks or more.

Example:

(Three NOP commands or one LDR command) + the 1st clock of the STR command using the following procedure (Section 3) makes 4 clocks total.

(Data is actually written at the 2nd clock of the STR command.)

(3) Write the following settings in 16-bit blocks to the DMA control register and stop the DMA:

- DMA-enabled flag: 0 (Disabled)
- DMA start timing: 00
- Data request transfer flag of the Game Pak side: 0 (DMA3 only)
- DMA repeat: 0
- Other control bits: No change

Note: Please note that the DMA may be started one extra time due to procedure 1 above.

12.4.2.2 How to Stop a DMA Repeat in the Direct Sound FIFO Transfer Mode

(1) Write the following settings in 32-bit blocks to the DMA control register and Word count register:

- DMA Word count register
  - Word count: 0004h
- -DMA control register
  - DMA-enabled flag: 1 (Enabled)
- DMA start timing: 00 (Start immediately)
- DMA transfer type: 1 (32 bit transfer mode)
- DMA repeat: 0 (OFF)
- Destination address control flag: 10 (Fixed)
- Other control bits: No change

However, when the value of the DMA word count register is already set to 0004h, the procedure is executed by writing in 16-bit width to the DMA control register. *

It is possible to disable the next repeated DMA by setting the DMA to start immediately; however, Direct Sound FIFO Transfer mode will be cancelled so that the value of the Word count register will be used. Therefore, the value of the Word count register needs to be set to 0004h.*

Similarly, the setting of destination address control flag will be used, so the value of 10 (destination address fixed) needs to be set, too.

Note: We recommend that the transfer type, destination address control flag, and the word count are initially set to the above setting. (i.e., transfer type = 1, destination address control flag = 10, and word count = 0004h).

(2) Run a process of 4 clocks or more.

Example:

(Three NOP commands or one LDR command) + the 1st clock of the STR command by the following procedure (3) equals a total of 4 clocks.

(Data is actually written at the 2nd clock of the STR command.)

(3) Write the following settings in the 16-bit width to the DMA control register and stop the DMA:

- DMA-enabled flag: 0 (Disabled)
- DMA start timing: 00
- DMA transfer type: 1
- DMA repeat: 0
- Destination address control flag: 10
- Other control bits: No change

Note: Please note that the DMA may be started one extra time due to procedure 1 above.

13 Communication Functions

Game Boy Advance provides the following five communication functions.

1. 8-Bit/32-Bit Normal Communication Function

   The use of Game Link cable for the previous DMG/MGB/CGB is prohibited for normal communication. It is possible to communicate at 256KHz and 2MHz with peripheral equipment that does not use cables.

   Always set the communication speed at 256KHz when performing normal communication using a Game Boy Advance Game Link Cable. Communication cannot be done properly at 2MHz. Also, please note it will be a one-way communication due to cable connection of multi-play communication.

   Due to differences in voltage, communication with DMG/MGB/CGB is not possible. Similarly, communication with previous DMG/MGB/CGB compatible hardware (pocket printer, etc.) which connects to an extension connector is not possible.

2. 16-Bit Multi-player Communication Function

   This multiple/simultaneous communication function uses UART system to enable communication of up to 4 Game Boy Advance units.

   A special cable for Multi-player communication is necessary.

3. UART Communication Function

   Enables high-speed communication by UART system.

4. General-Purpose Communication Function

   Enables communication by any protocol through direct control of the communication terminal.

5. JOY Bus Communication Function

   Enables communication using Nintendo's standardized Joy bus.

Selecting Communication Function

All the communication functions use an external expansion 6-pin connector. Communication functions are switched by the communication function set flag of the communication control register RCNT (2-bit) and the communication mode set flag of the serial communication control register SIOCNT (2-bit), which are described later.

Table 27 - Communication Functions

Communication Functions	RCNT		SIOCNT	
	d15	d14	d13	d12
General-Purpose	1	0	*	*
JOY Bus	1	1	*	*
8-Bit Serial	0	*	0	0
32-Bit Serial	0	*	0	1
16-Bit Serial	0	*	1	0
UART	0	*	1	1

(* … any)

Do not change or reset a communication mode during communication, as this may cause a communication malfunction.

When changing communication modes, change only the communication mode flag first. Do not start a communication at the same time the mode is changed. This may cause a malfunction.

Even if you think you have set different bits of same register separately using the C language, sometimes they are optimized by a compiler and changed to codes that are set simultaneously.

When this happens, attach the type qualifier, volatile, in order to prevent optimization.

Example:

```
*(volatile unsigned short int*)REG_AGB =0x8000;      /*REG_AGB:register name*/
*(volatile unsigned short int*)REG_AGB = 0x0040;
```

If communication is not finished (SIO interrupt does not occur) after a certain period of time, or if there is a communication error after retries, enter another communication mode once and then re-enter the communication mode once again. By doing this, the communication circuit will be reset.

Cautions for Communication Function

For communication, take into consideration a case in which unexpected data is received. Be careful so that a lock up, destruction of saved data, or malfunction do not occur.

(Example: To permit cancellation of communication by pressing a key.)

The following situations are examples of communication problems.

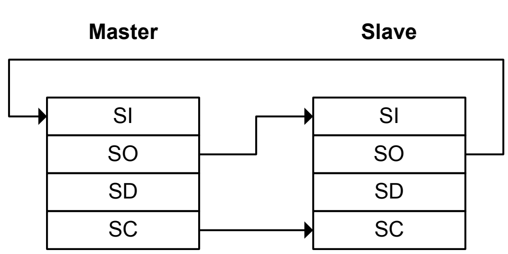
- When a peripheral device that is not supported is connected
- When different software is connected to other device
- When the communication mode is different from the other device
- When the Game Boy Advance Game Link cable is connected incorrectly
- When an error occurs in data due to noise

13.1 8-Bit/32-Bit Normal Serial Communication

Serial transfer sends/receives simultaneously.

If data is stored in the data register and the serial transfer is started, received data is stored in the data register when the transfer is complete.

Figure 95 - Connecting during Normal Serial Communication:



The master (internal clock mode) will output the shift clock from the SC terminal. The SD terminal outputs a LO.

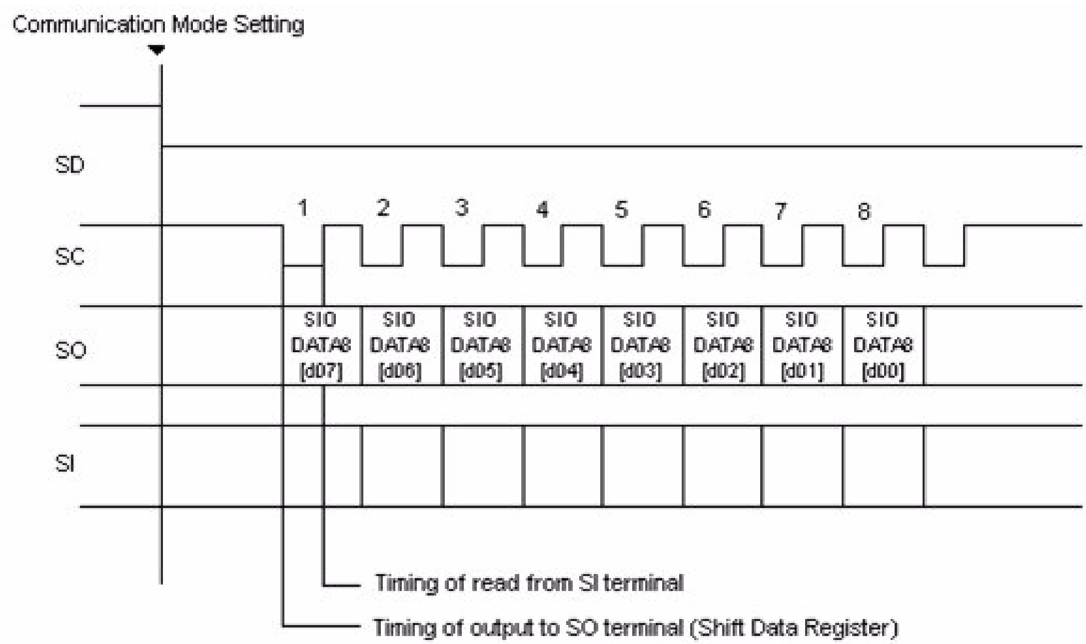
In the case of a slave (external clock mode), the SC terminal becomes an input terminal, with pull-up. The SD terminal outputs a LO.

The stored data will be left-shifted by the falling edge of the shift clock, and will be output from the SO terminal in order, starting from the most significant bit. The data input from the SI terminal will be input to the least significant bit, with the rising edge of the shift clock.

13.1.1 SIO Timing Chart

The figure below illustrates 8-bit communication. In 32-bit communication, the shift clock sends and receives 32 bits of data.

Figure 96 - SIO Timing Chart (8-bit Communication)



8 bit Normal Serial Communication Data Register

8-bit transfer mode uses SIODATA8 as a data register. The upper 8-bits will become disabled.

(This data register is used for 16 bit multi-play communication as well.)

Figure 97 - The SIODATA8 Register

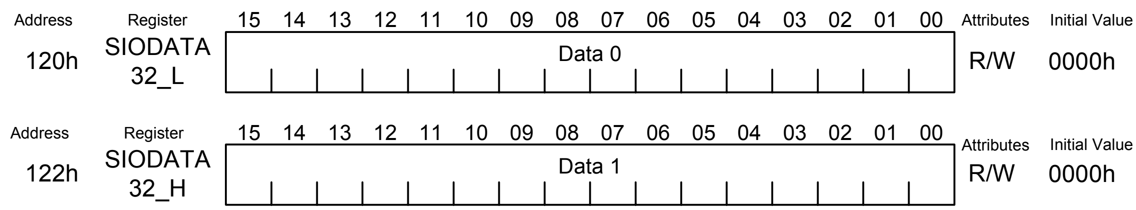
Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
12Ah	SIODATA8																	R/W	0000h

13.1.2 32-Bit Normal Serial Communication Data Registers

32-bit transfer mode uses [120h:SIODATA32_L] and [122h:SIODATA32_H] as data registers.(These data registers are used for 16-bit multi-player communication also.)

The most significant bit will be d15 in the register SIODATA32_H, and the least significant bit will be d0 in the register SIODATA32_L.

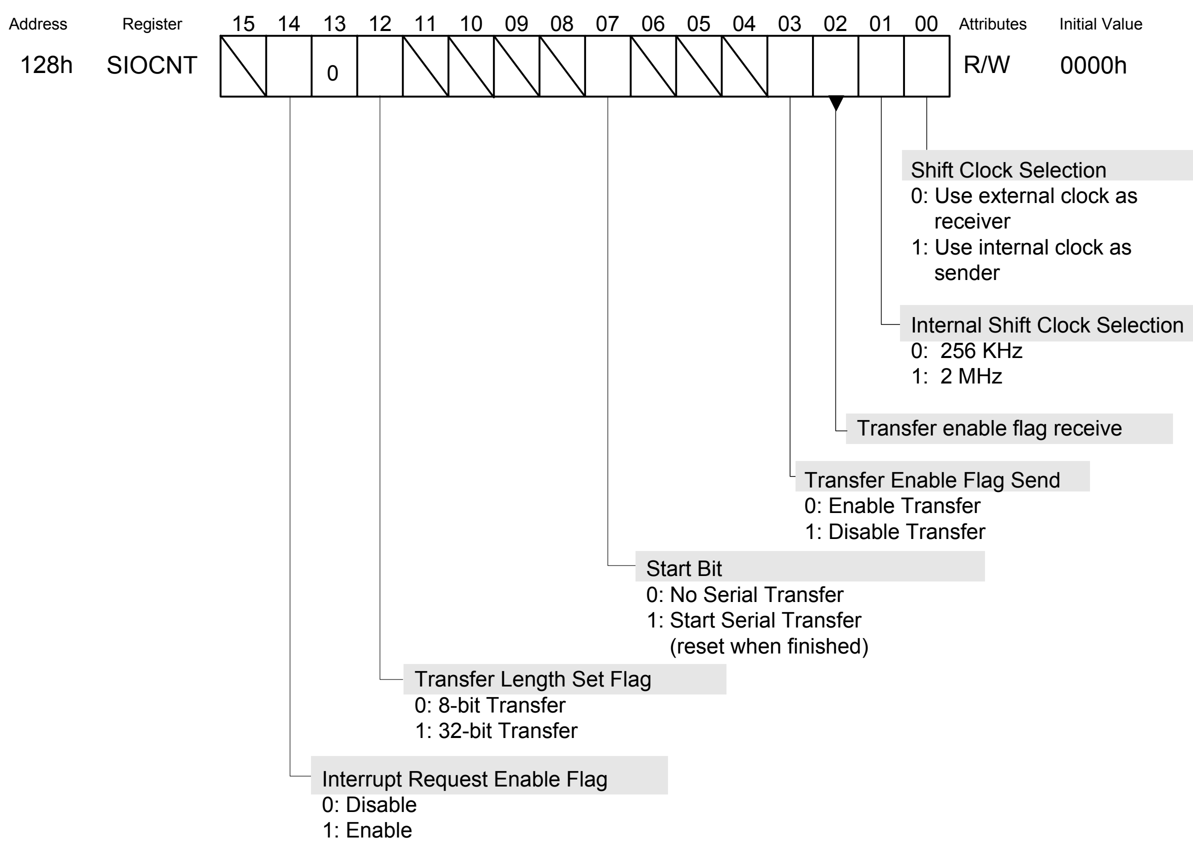
Figure 98 - 32-Bit Normal Serial Communication Data Registers



13.1.3 Control Register

When Register RCNT (d15) = (0), the mode will be 8-bit normal serial communication mode by setting to Register SIOCNT (d13, d12) = (0,0), and the mode will be 32-bit normal serial communication mode by setting to SIOCNT (d13, d12) = (0, 1).

Figure 99 - The SIOCNT Register (32-Bits)



SIOCNT [d14] Interrupt Request Enable Flag

If 0 is set, an interrupt request will not be made.

If 1 is set, an interrupt request will be made immediately after transfer is complete.

SIOCNT [d12] Transfer Length Setting Flag

Sets bit length of transfer data.

If 0, 8-bit transfer is carried out. If 1, 32-bit transfer is carried out.

SIOCNT [d07] Start Bit

With a setting of 1, a serial transfer starts. The bit is automatically reset after transfer completion.

SIOCNT [d03] Transfer Enable Flag Send

A setting of 0 enables transfer; 1 disables it.

This flag is output from the SO terminal until the start of a transfer. When the transfer starts, serial data are output from the SO terminal.

SIOCNT [d02] Transfer Enable Flag Receive

It is possible to read the status of SI terminal (transfer-enable flag transmitting of the other party's hardware) before communication starts.

It becomes invalid after communication has started.(receive data bit during communication is reflected.)

SIOCNT [d01] Internal Shift Clock Selection

If 0, 256KHz is selected for the shift clock.

If 1, 2MHz is selected for the shift clock.

SIOCNT [d00] Shift Clock Selection

If 0, an external clock is used as a shift clock. (slave)

The external clock is input by the SC terminal from another hardware unit. SD terminal will go to LO output.

If 1, an internal clock is used as a shift clock. (master)

The internal clock is output from the SC terminal, and SD terminal will be in the pull-up input status.

Cautions for Normal Serial Communications

The shift clock should be selected before the start bit of the SIOCNT register is set. Extra shift operations may result if the serial transfer is started before or at the same time as the shift clock is selected.

Do not use a value of the transfer enable flag receiving bit for SIOCNT register while the start bit of SIOCNT register is being set. (Because it transforms to a receiving data bit that is being communicated.)

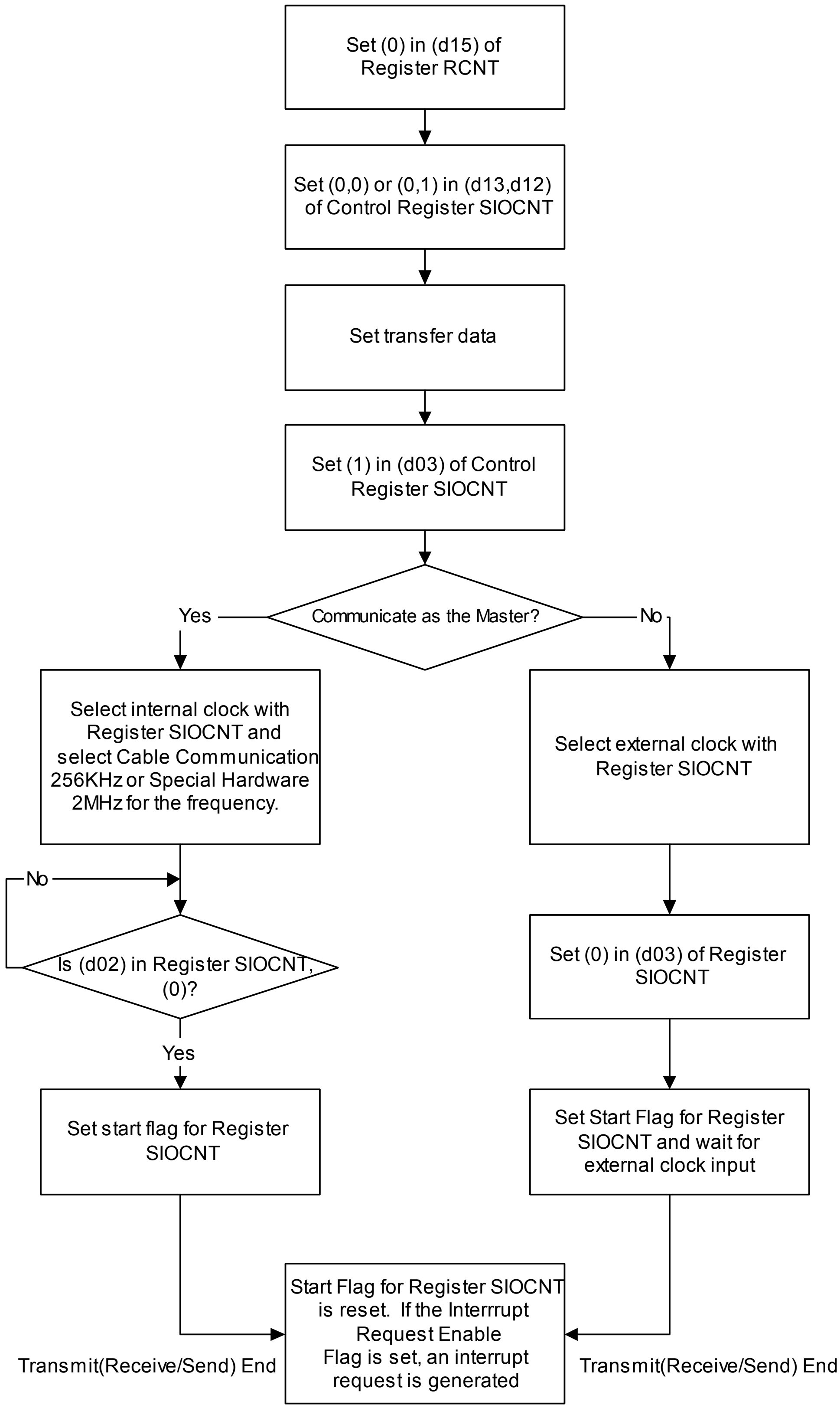
The 8 bit transfer mode is compatible in terms of modes with DMG/CGB, but the voltage with the communication terminal varies. Therefore, communication between Game Boy Advance and DMG/CGB is not possible.

Using a Game Link cable for DMG/MGB is prohibited in normal serial communication mode. It is possible to communicate at 256 KHz and 2 MHz with peripheral equipment that does not use a cable.

Always set a communication speed of 256 KHz when performing normal communication with the Game Boy Advance Game Link cable. Communication cannot be done properly at 2 MHz.

Also, please note it will be a one-way communication due to cable connection of multi-play communication.

Figure 100 - Normal Serial Communication Flow (Example)



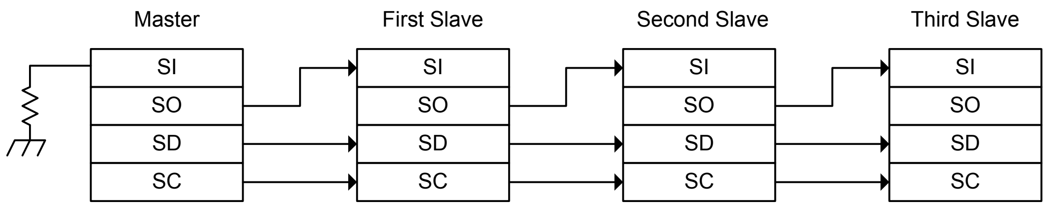
13.2 16-Bit Multi-player Communication

Game Boy Advance enables multi-player communication between up to 4 units using a special cable.

Depending on the connection status, 1 unit is established as the master and transfers data to slaves in order, one after another.

13.2.1 Connection Status during Multi-player Communication

Figure 101 - Multi-Player Communication Connection Status



In multi-player communication mode the SC and SD become pull-up input terminals. Immediately following a reset or in another communication mode, LO is output from the SD terminal. Once the SD terminal becomes HI, you can tell that all connected terminals have entered multi-player communication mode.

The SI terminal is in pull-up input, but due to the multi player Game Boy Advance Game Link cable it becomes pull-down. Thus, once all of the terminals are in multi-player mode, the terminal that is LO input to the SI terminal becomes the master. The terminal that is HI input to the SI terminal becomes the slave.

If you set the start bit of Register SIOCNT of the master, the data registers SIOMULTI0, SIOMULTI1, SIOMULTI2, and SIOMULTI3 of the master are initialized to FFFFh.

Additionally, the “SYNC signal” (LO level) is output from the SC terminal. At the same time, the “Start bit” (LO level) is output from the SD terminal. Next, the data from Register SIOMLT_SEND is output and a ““Stop bit” (HI level) is output.

After this is done, the master makes the SD terminal become pull-up input, and LO is output from the SO terminal.

Each slave detects the “SYNC Signal” output from the master and initializes all of the data registers (SIOMULTI0, SIOMULTI1, SIOMULTI2, and SIOMULTI3) to FFFFh. The data output from the master is stored in the master and each slave’s SIOMULTI0 register.

If LO is input to the SI terminal of the slave which was connected immediately following the master, a “Start bit” (LO level) is output from the SD terminal. Next, data from Register SIOMLT_SEND is output, and lastly a “Stop bit” (HI level) is output.

After this, the SD terminal goes to pull-up input and LO is output from the SO terminal.

At this point, the data output from the first slave is stored in the master and each slave’s SIOMULTI1 Register.

In this way, each slave is sent and all transmissions are carried out.

In the following situations the master produces a “SYNC Signal” (pull-up input after the output of a 5 cycle HI interval of source oscillation) and the transmission ends:

- After the master outputs its own “Stop bit”, the next “Start bit” is not input after a certain period of time.
- After a “Stop bit” is received from the first or second slave, a “Start bit” is not input after a certain period of time.
- A “Stop bit” is received from the third slave.

Once the transmission ends, the received data is stored in each of the data registers (SIOMULTI0, SIOMULTI1, SIOMULTI2, and SIOMULTI3). If there is a terminal that is not connected the initial data FFFFh is stored.

Figure 102 - Multi-player Communication Timing Chart

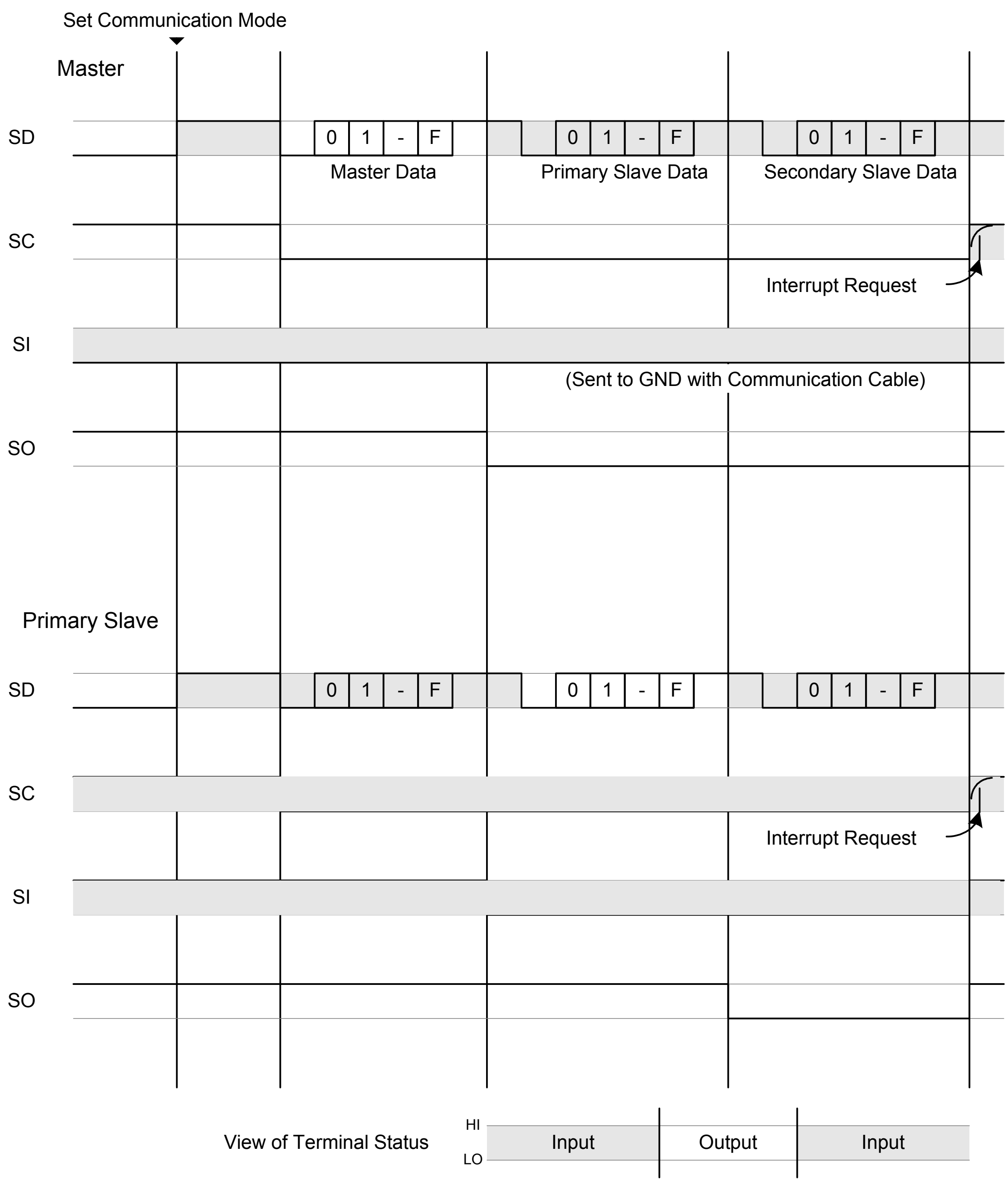
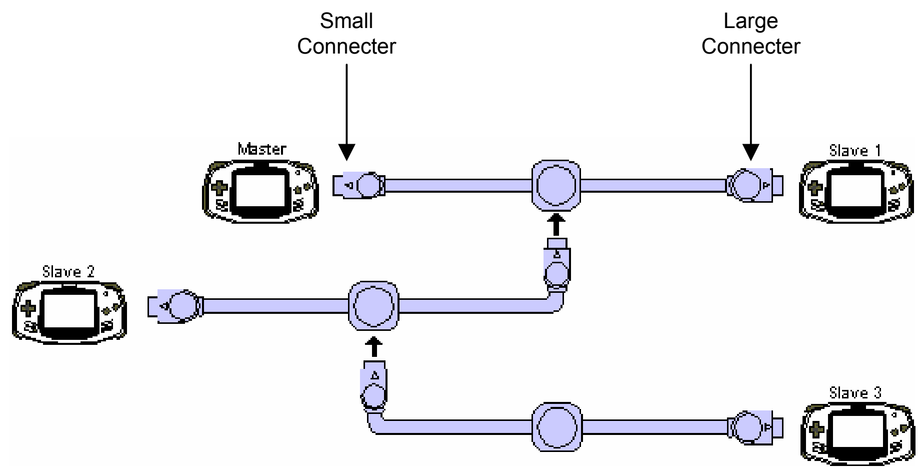


Figure 103 - Multi Player Game Boy Advance Game Link Cable Connecting Diagram



13.2.2 Data Registers

The data send is stored in the Register SIOMLT_SEND.

Figure 104 - The SIOMLT_SEND Register

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
12Ah	SIOMLT_ SEND																	R/W	0000h

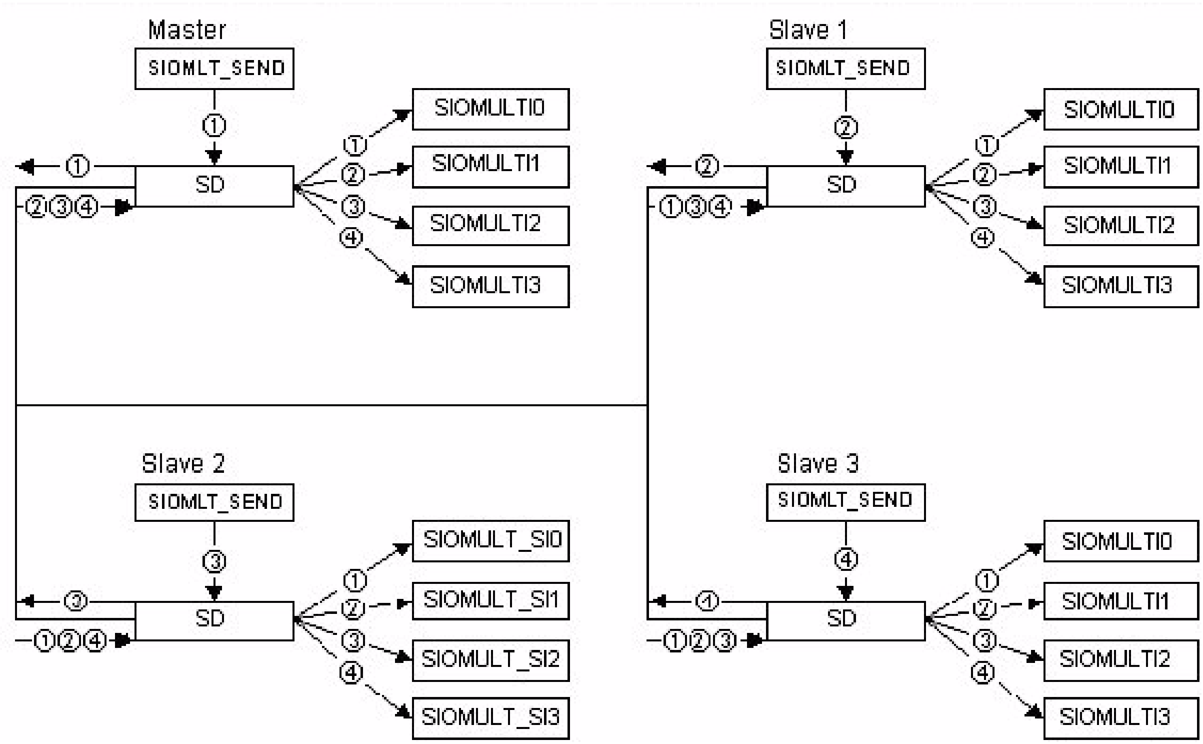
After multi-player communication is finished, a send data of Master is in SIOMULTI0. Send data of First slave, Second slave, and Third slave are in SIOMULTI1, SIOMULTI2, and SIOMULTI3 respectively.

Figure 105 - Multi-Player Data Registers

Address	Register	15–00	Attributes	Initial Value
120h	SIO MULTI0	Data 0	R/W	0000h
122h	SIO MULTI1	Data 1	R/W	0000h
124h	SIO MULTI2	Data 2	R/W	0000h
126h	SIO MULTI3	Data 3	R/W	0000h

13.2.3 Data Transition Diagram

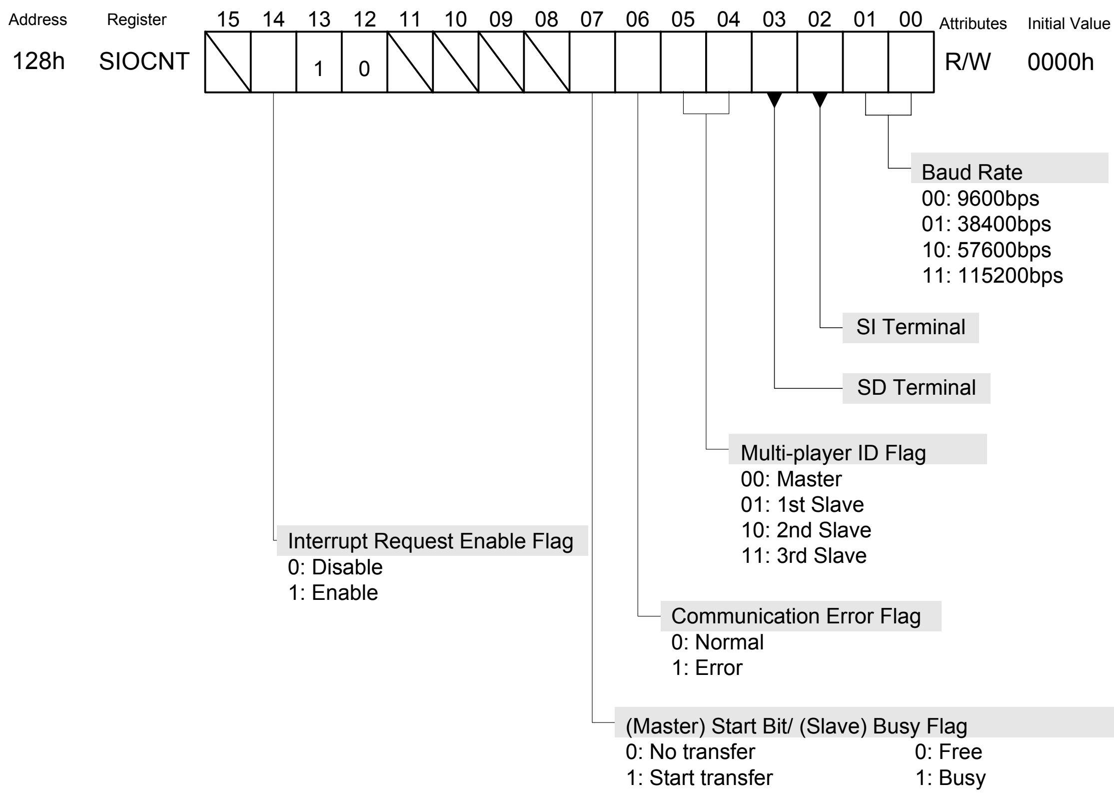
Figure 106 - Multi-Player Data Transitions



13.2.4 Control Register

If you set Register SIOCNT (d13,d12) = (1,0) when Register RCNT (d15) = (0), you will go to 16-bit multi-player communication mode.

Figure 107 - The SIOCNT Register (16-Bit)



SIOCNT [d14] Interrupt Request Enable Flag

When set to 0, no interrupt request is generated.

When set to 1, an interrupt request is generated upon the completion of multi-player communication.

SIOCNT [d07] Start Bit/Busy Flag

1. Master (d02 is 0)

   When set to 0, no data is transferred.

   When set to 1, a data transfer is started. Upon completion of the data transfer, it is automatically reset.

   Caution

   Due to individual differences in Game Boy Advance hardware, there is a variation in the timing of interrupt occurrences. Always use a timer when sending data, so that you have an adequate interval between communications (minimum send interval + 600 clocks). 600 clocks is the guaranteed value that considers the variation in timing of interrupt occurrence. Refer to Game Boy Advance Programming Cautions 5.1.”16-Bit MultiPlayer Communications.”

2. Slave (d02 is 1)

   Set during input of transmit start bit (LO source oscillation cycle × 3 (approximately 180ns)), and reset when the transfer is complete.

SIOCNT [d06] Communication Error Flag

The communication status can be confirmed at the end of a communication. (During communication, it is not reflected properly.)

If the status for this bit is 0, there is no error. If it is 1, it means an error has occurred.

This error flag is automatically set in the following situations:

- The SI Terminal does not become LO during the interval when the "SYNC signal" is being input (the master is outputting).

  Example: When connected to the fifth slave or after that, or when the previous slave is not connected.
- The stop bit for the receive data is not HI (Framing Error)

However, communication continues even when an error occurs, and invalid data is stored in SIOMULTI0 - SIOMULTI3.

Confirm error flags when communicating so there are no problems created in case of an incorrect cable connection.

SIOCNT [d05 - d04] Multi-player ID Flag

When multi-player communication ends, an ID code will be stored which specifies the order that each particular machine was connected.

Confirm ID code when communicating so there are no problems created in case of an incorrect cable connection.

SIOCNT [d03] SD Terminal

The status of the SD Terminal can be read. If all of the connected terminals enter multi-player communication mode, it becomes HI status.

SIOCNT [d02] SI Terminal

The status of the SI Terminal can be read.

When all of the connected terminals are in multi-player communication mode, this shows that the terminal which is LO input to the SI terminal is the master. HI input means that it is a slave.

Prior to communication starting, it is not possible to determine the number order of a particular slave.

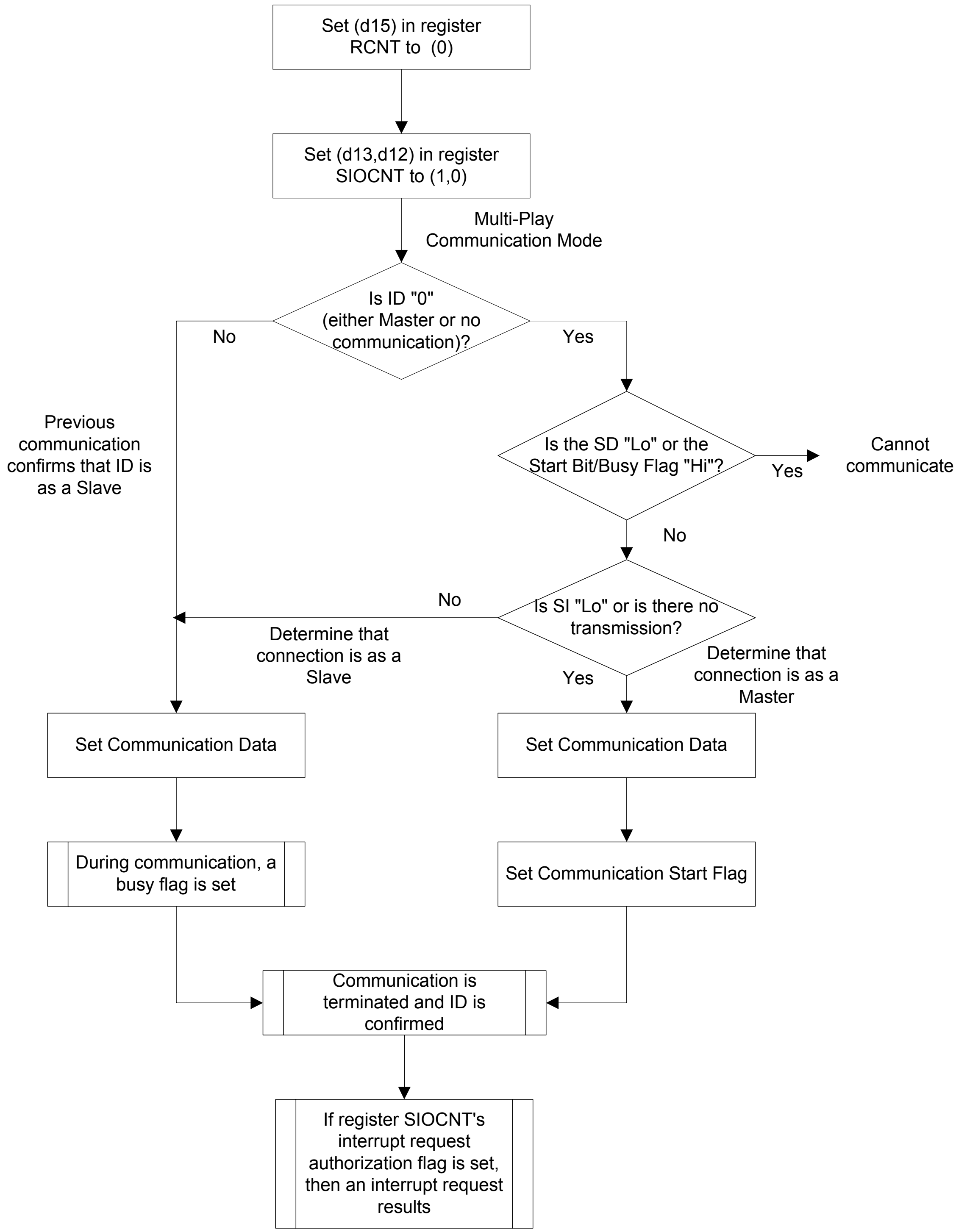
SIOCNT [d01 - d00] Baud Rate

Sets the communication baud rate.

Table 28 - Normal Serial Communication Baud Rates

Setting	Baud Rate
00	9600 bps
01	38400 bps
10	57600 bps
11	115200 bps

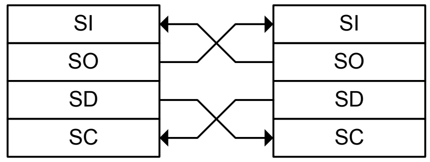
Figure 108 - Multi-player Communication Flow (Example)



13.3 UART Communication Functions

UART communications can be illustrated using the following drawing.

Figure 109 - UART Communication



In UART communication mode, a HI level is output from the SD terminal.

When the receive data register (or the receive FIFO) is full, a HI is output from the SD terminal. When it is not full, a LO is output from the SD terminal if the receive enable flag is set. A HI is output if it is reset.

The output of the SD terminal of the other machine is input to the SC terminal.

Once data is written to the send data register, data is sent after a “Start bit” (1 bit) is sent from the SO terminal. However, when the CTS flag for the Control Register is set, data can be sent only when there is a LO input to the SC terminal. The Stop bit is a fixed 1 bit.

13.3.1 Data Register

Figure 110 - The SIODATA8 Register

Address	Register	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Attributes	Initial Value
12Ah	SIODATA8	\	\	\	\	\	\	\	\									R/W	0000h

13.3.2 Relations Between Data Register, FIFO, and Shift Register

When sending or receiving, there are 4 bytes of FIFO. By using the FIFO enable flag for the control register SIOCNT, you can select whether to use or not use FIFO.

13.3.3 When FIFO is not Used

If written to a data register SIODATA8, data is written to a send shift register, and if read, data is read from a receive shift register. (Only the lower 8 bits are valid.)

Figure 111 - Serial Communication without FIFO

13.3.4 When FIFO is Used

If written to a data register SIODATA8, data is written to a send FIFO. If all the contents of the send shift register are shifted out, data is transferred from a send FIFO to a shift register, immediately.

Please note when using this operation, that data is immediately transferred to a shift register when the first data is written to the data register, and the interrupt request condition is met as a send FIFO becomes empty. Also, when read, data is read from a receive FIFO. (Only the lower 8 bits are valid.)

Figure 112 - Serial Communication with FIFO

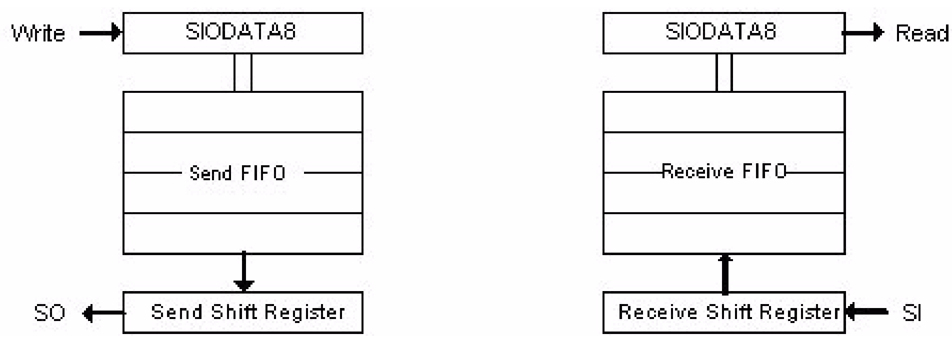
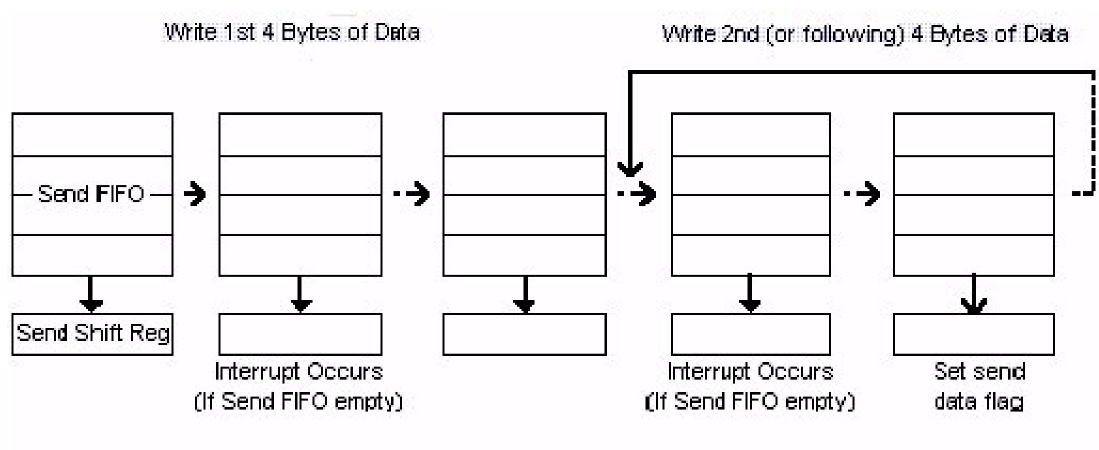


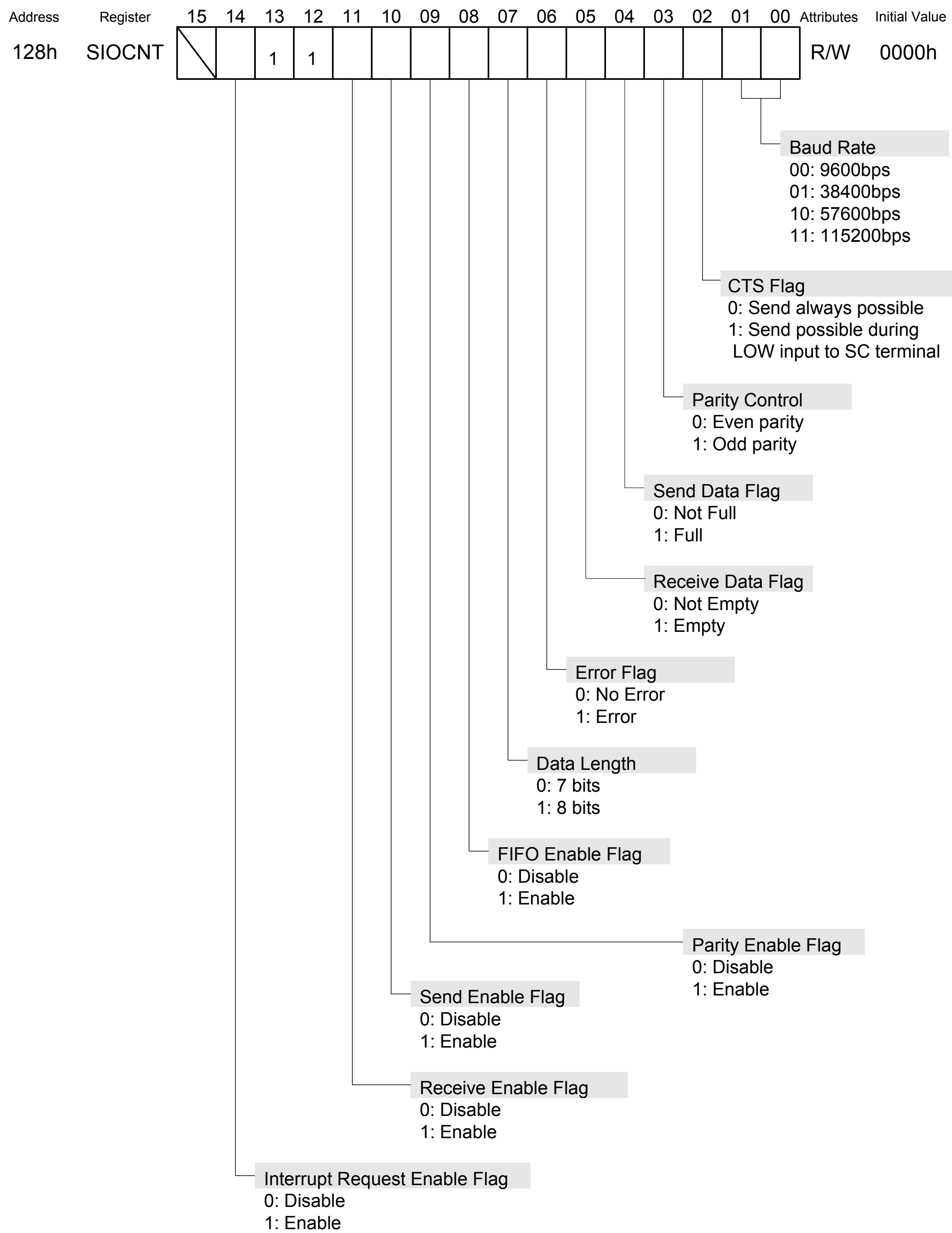
Figure 113 - Example: Writing Data Registers



13.3.5 Control Register

If Register SIOCNT (d13,d12) = (1,1) is set when Register RCNT (d15) = (0), you will go to UART communication mode.

Figure 114 - The SIOCNT Register (UART)



SIOCNT [d14] Interrupt Request Enable Flag

When set to 0, an interrupt request is not generated.

When set to 1 and FIFO is invalid, an interrupt request is generated when a communication error occurs or when the transmission (send/receive) ends.

When set to 1 and FIFO is valid, an interrupt request is generated when a communication error occurs, when a send FIFO is emptied, or a receive FIFO becomes full.

SIOCNT [d11] Receive Enable Flag

Controls the receive enable/disable.

If the receive enable flag is set when the receive data register (or the receive FIFO) is not full, a LO is output from the SD terminal, and a HI is output if it is reset.

Note: You must first set the receive enable flag and send enable flag to 0 [Disable] before going from UART communication mode to a different communication mode.

SIOCNT [d10] Send Enable Flag

Controls the send enable/disable.

Note: You must first set the receive enable flag and send enable flag to 0 [Disable] before going from UART communication mode to a different communication mode.

SIOCNT [d09] Parity Enable Flag

Controls the parity enable/disable.

SIOCNT [d08] FIFO Enable Flag

Controls the send of the 8 bit wide × 4 depth and the receive FIFO enable/disable.

Note: When using FIFO, first you need to go into UART mode in a status of 0 [FIFO Disable]. By disabling FIFO in UART mode the FIFO sequencer is initialized.

SIOCNT [d07] Data Length

Select data length as 8 bits or 7 bits.

SIOCNT [d06] Error Flag

By referring to this error flag, the status of communication errors can be determined.

When it is 0, no errors have occurred. When it is set to 1, an error has occurred. By reading Register SIOCNT, this error flag is reset.

Additionally, when there has been an error, the data from the Receive Shift Register is not written to the Receive Data Register. The conditions associated with each error are described below.

Table 29 - UART Communication Error Conditions

Error Name	Condition
Framing Error	The receive data stop bit is not 0
Parity Error	When parity is enabled, there is an error in the parity for the receive data
Overrun Error	When FIFO is invalid, if the receive data is not empty (SIOCNT [d05] = 0) and next receive has ended (detect stop bit). Or when FIFO is valid, if receive FIFO is full and next communication has ended (detect stop bit).

SIOCNT [d05] Receive Data Flag

When set to 0, there is still data present.

When set to 1, it is empty.

SIOCNT [d04] Send Data Flag

When set to 0, it is not full.

After one send operation ends this is reset.

When set to 1, it is full.

Set during a write of data to the lower 8 bits of the Send Data Register SIODATA8.

SIOCNT [d03] Parity Control

Switches between even parity and odd parity.

SIOCNT [d02] CTS Flag

The SD terminal of the other machine (receive enable/disable) is input to the SC terminal.

When set to 0, a send is always possible independent of the SC Terminal.

When set to 1, a send is only possible when a LO is being input to the SC Terminal.

SIOCNT [d01 - d00] Baud Rate

Sets communication baud rate.

Table 30 - UART Communication Baud Rates

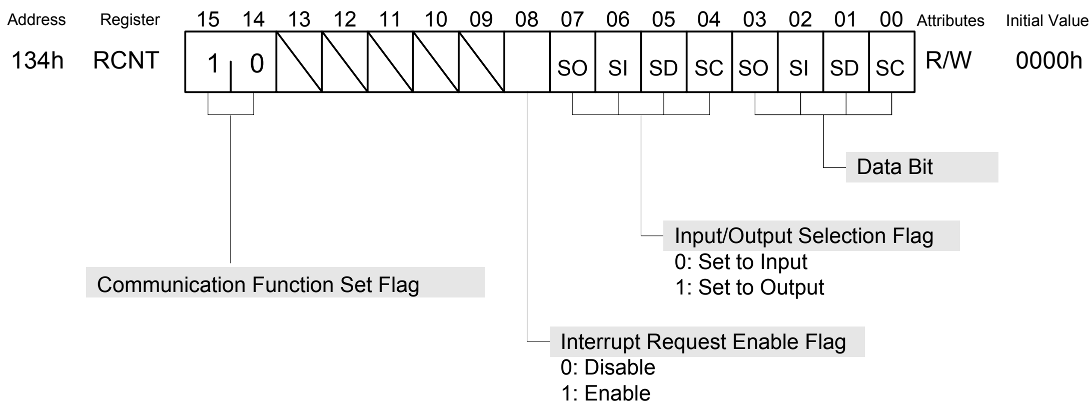
Setting	Baud Rate
00	9600 bps
01	38400 bps
10	57600 bps
11	115200 bps

13.4 General-Purpose Communication

By setting (d15, d14) = (1, 0) for RCNT register, it will change to a general-purpose communication mode.

In this mode, all of the terminals SI, SO, SC, and SD become pull-up and operate as general-purpose input/output terminals. Each of the communication terminals SI, SO, SC, and SD can be directly controlled.

Figure 115 - The RCNT Register (General-Purpose Communication)



RCNT [d15 - d14] Communication Function Set Flag

When set to 00 or 01, operates as a serial communication(8-bit/16-bit serial communication, multi-player communication, UART communication function) terminal.

When set to 10, can be used as a general-purpose input/output terminal.

When set to 11, can be used as a JOY Bus communication terminal.

RCNT [d08] Interrupt Request Enable Flag

When general-purpose input/output is set(R[d15,d14]=[1,0]) with the communication function set flag, a 1 causes an interrupt request to be generated with the falling of the SI Terminal (edge detect).

When set to 0, no interrupt request is generated.

RCNT [d07 - d04] Input/Output Selection Flag

When general-purpose input/output is set (R[d15,d14]=[1,0]) with the communication function set flag, a setting of 0 allows the corresponding terminal to be used as an input terminal. A setting of 1 allows the corresponding terminal to be used as an output terminal.

Caution

Always set the SI terminal to an input. If it is set to an output, a problem may occur with some connecting equipment.

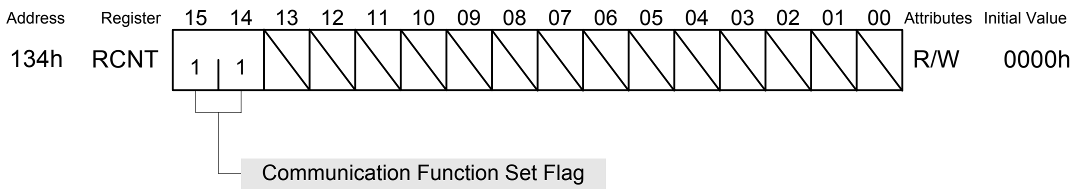
RCNT [d03 - d00] Data Bit

When the corresponding terminal is set for input, the status (HI/LO) of the terminal can be confirmed. If the corresponding terminal is set for output, the status of the set bit is output.

13.5 JOY Bus Communication

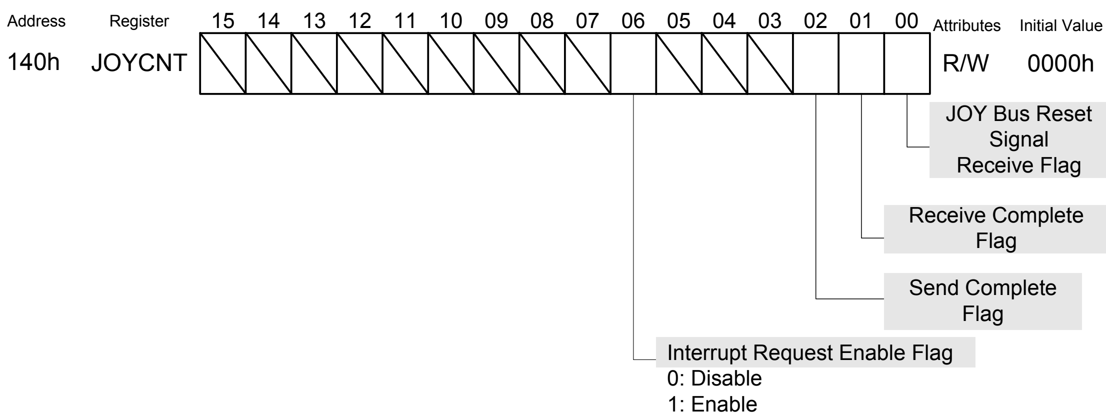
By setting the communication function set flag to 11 for Register RCNT, JOY Bus communication mode is selected. In JOY Bus communication mode, the SI Terminal is for input, and SO Terminal is for output. SD and SC Terminals go to LO output.

Figure 116 - The RCNT Register (JOY Bus Communication)



13.5.1 JOY Bus Communication Control

Figure 117 - The JOYCNT Register



JOYCNT [d05] Interrupt Request Enable Flag

When set to 0, an interrupt request is not generated.

When set to 1, an interrupt request is generated if JOY Bus reset command is received.

JOYCNT [d02] Send Complete Flag

Set upon completion of send operation.

When this is set, if you write a 1, a reset can be done.

JOYCNT [d01] Receive Complete Flag

Set upon completion of receive operation.

When this is set, if you write a 1, a reset can be done.

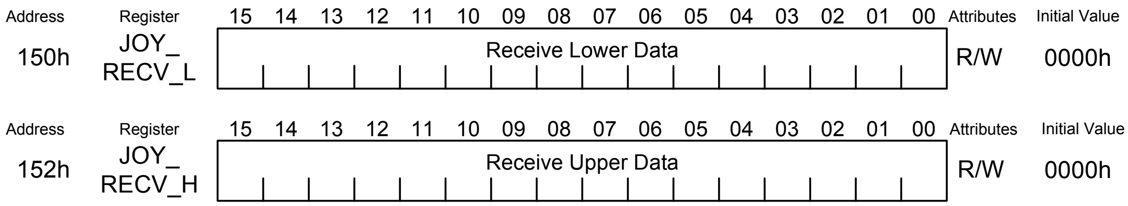
JOYCNT [d00] JOY Bus Reset Signal Receive Flag

Set when a JOY Bus reset command is received.

When this is set, if you write a 1, a reset can be done.

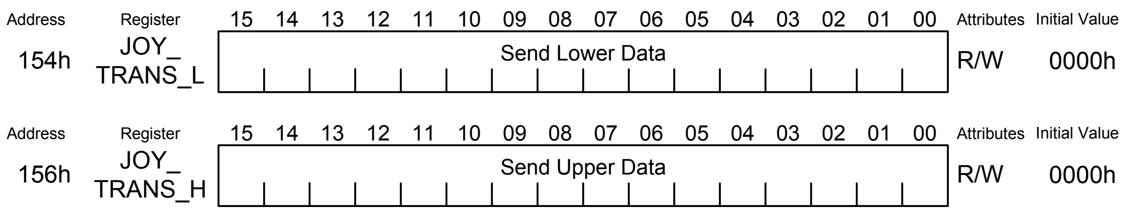
13.5.2 Receive Data Registers

Figure 118 - JOY Bus Receive Data Registers



13.5.3 Send Data Registers

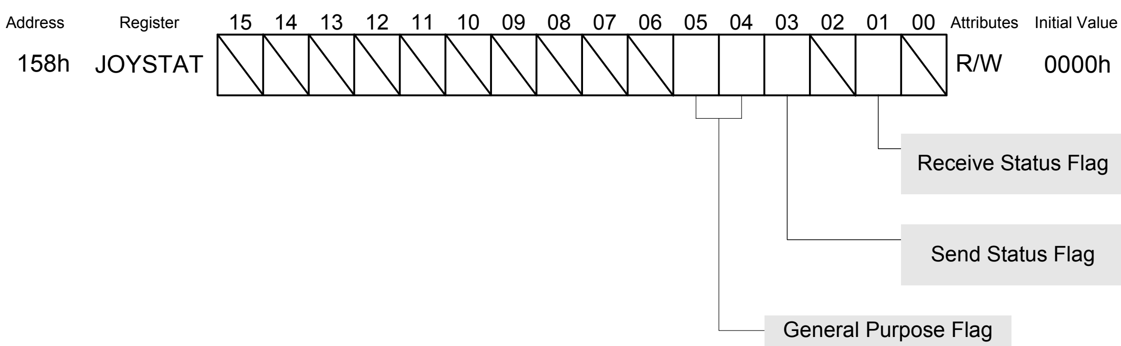
Figure 119 - JOY Bus Send Data Registers



13.5.4 Receive Status Register

The lower 8-bits of the receive status register JOYSTAT is returned as the communication status.

Figure 120 - The JOYSTAT Register



JOYSTAT [d05,d04] General-Purpose Flag

This flag is not assigned.

The user can set the use of this flag arbitrarily.

JOYSTAT [d03] Send Status Flag

Set this bit if the JOY_TRANS register is written by word. Reset this bit if a JOY Bus Data Read Signal is received.

JOYSTAT [d01] Receive Status Flag

Set this bit if a JOY Bus Data Write Signal is received. Reset this bit if the JOY_RECV register is read by word.

13.5.5 JOY Bus Communication Operations

Game Boy Advance JOY Bus communication recognizes four commands sent from the host (for example, Nintendo GameCube). These commands are: “JOY Bus Reset”, “Type/Status Data Request”, “JOY Bus Data Write”, and “JOY Bus Data Read.” Game Boy Advance operates based on the particular signal received.

The transfer of the bit data for JOY Bus communication is done in units of bytes and in the order of MSB first.

13.5.6 [JOY Bus Reset] Command(FFh) Received

The JOY Bus reset signal receive flag of Register JOYCNT is set.

If the interrupt request enable flag for the same register is also set, a JOY Bus interrupt request is generated.

Table 31 - The JOY Bus Reset Command

Direction	Order	d7	d6	d5	d4	d3	d2	d1	d0	Remarks
Receive	1	1	1	1	1	1	1	1	1	Command 255(FFh)
Send	1	0	0	0	0	0	0	0	0	Type Number 0400h
	2	0	0	0	0	0	1	0	0	
	3	Lower 8 bits of Register JOYSTAT								Communication Status

13.5.7 [Type/Status Data Request] Command(00h) Received

Returns 2 byte type number(0004h) and 1 byte communication status.

Table 32 - The Type/Status Data Request Command

Direction	Order	d7	d6	d5	d4	d3	d2	d1	d0	Remarks
Receive	1	0	0	0	0	0	0	0	0	Command 0(00h)
Send	1	0	0	0	0	0	0	0	0	Type Number 0400h
	2	0	0	0	0	0	1	0	0	
	3	Lower 8 bits of Register JOYSTAT								Communication Status

13.5.8 [JOY Bus Data Write] Command(15h) Received

Receives the 4 bytes of data sent following this command, and stores them in Register JOY_RECV. Once the receive is completed a 1 byte communication status is returned, and the receive complete flag for Register JOYCNT is set. Also, if the interrupt request enable flag for the same register is set, a JOY Bus interrupt request is generated.

Table 33 - The JOY Bus Data Write Command

Direction	Order	d7	d6	d5	d4	d3	d2	d1	d0	Remarks
Receive	1	0	0	0	1	0	1	0	1	Command 21(15h)
Receive	2	Lower 8 bits of receive data Register JOY_RECV_L								Receive Data
	3	Upper 8 bits of receive data Register JOY_RECV_L								
	4	Lower 8 bits of receive data Register JOY_RECV_H								
	5	Upper 8 bits of receive data Register JOY_RECV_H								
Send	6	Lower 8 bits of Register JOYSTAT								Communication Status

13.5.9 [JOY Bus Data Read] Command(14h) Received

4 bytes of data stored in Register JOY_TRANS and the 1 byte communication status are sent, and the send complete flag for Register JOYCNT is set.

Also, if the interrupt request enable flag for the same register is set, a JOY Bus interrupt request is generated.

Table 34 - The JOY Bus Data Read Command

Direction	Order	d7	d6	d5	d4	d3	d2	d1	d0	Remarks
Receive	1	0	0	0	1	0	1	0	0	Command 20(14h)
Send	2	Lower 8 bits of send data Register JOY_TRANS_L								Send Data
	3	Upper 8 bits of send data Register JOY_TRANS_L								
	4	Lower 8 bits of send data Register JOY_TRANS_H								
	5	Upper 8 bits of send data Register JOY_TRANS_H								
	6	Lower 8 bits of Register JOYSTAT								Communication Status

JOY Bus Communication Caution

If the Nintendo GameCube Game Boy Advance Cable is disconnected during JOY Bus communication, the Game Boy Advance's JOY Bus communication circuitry may malfunction and JOY Bus communication would no longer be possible, even if the cable is reconnected. For this reason, if around 0.16 seconds (approximately 10 frames) passes and no SIO interrupt has been generated in the Game Boy Advance, reset the communication circuitry to time out the process.

If the GBA cable is disconnected and then reconnected, the Game Boy Advance may behave the same way it does when it receives a JOY Bus reset command (i.e., the JOY_IF_RESET bit of the REG_JOYCNT register is set). In this situation, an interrupt request will be generated if SIO interrupts are enabled.

When a Game Boy Advance is connected to Nintendo GameCube and continuous communication is executed, the Game Boy Advance will experience frequent SIO interrupts.

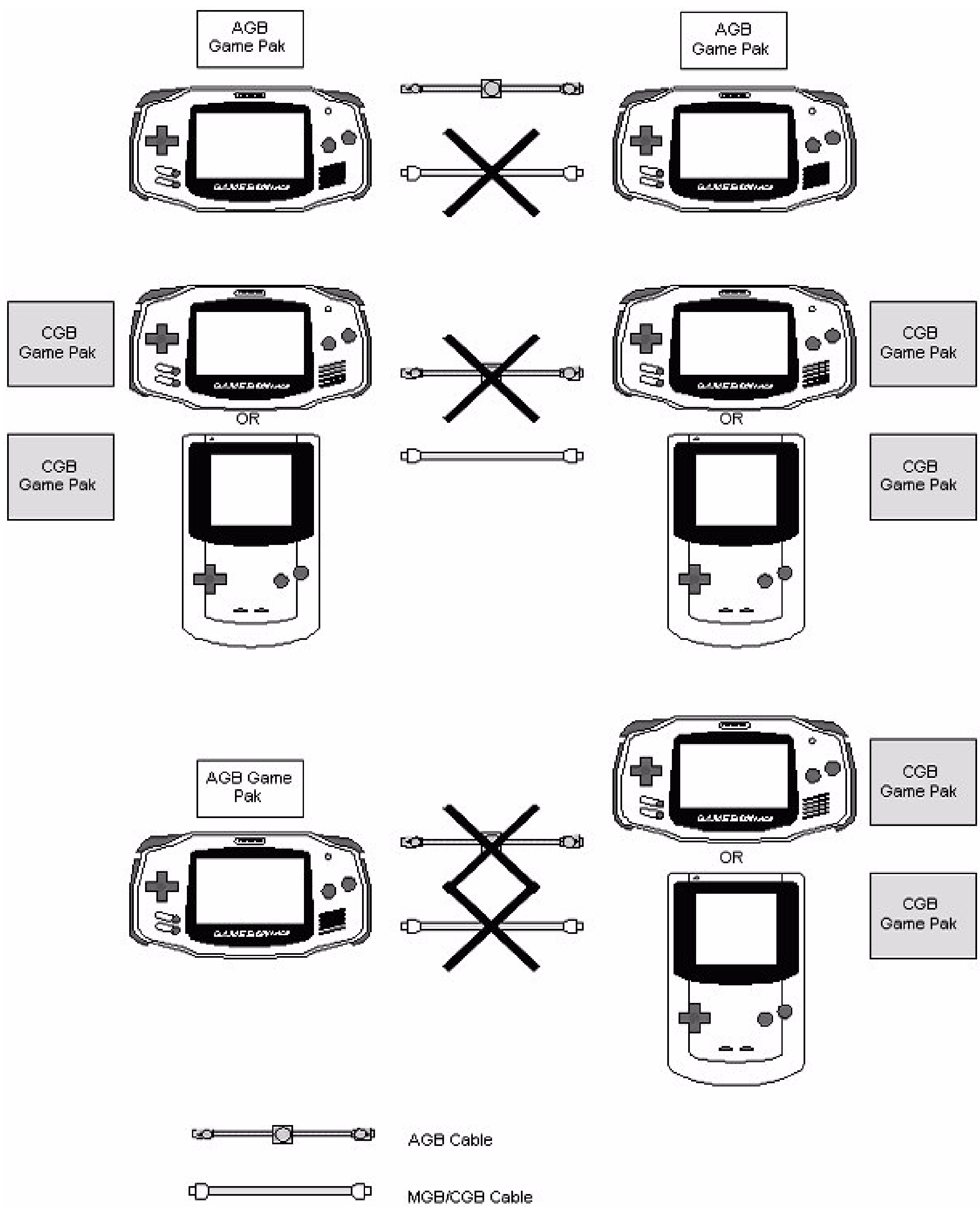
If the Game Boy Advance cannot respond to this series of SIO interrupts (due to such factors as a large DMA, an extended interrupt process, or a prolonged disabling of interrupts), then data communication will fail and incorrect values will be exchanged.

Thus, you should design your application so that the progress of the game will not be affected due to the exchange of incorrect values during communication. For example, do not disable interrupts for long periods of time, and either divide up large DMA tasks or use some other method such as substituting them with CpuFastCopy().

13.6 Game Boy Advance Game Link Cable

When communicating between Game Boy Advance units, the Game Boy Advance Game Link cable to be used will vary depending upon the type of Game Pak used.

Figure 121 - Game Boy Advance Game Link Cable Connection Types



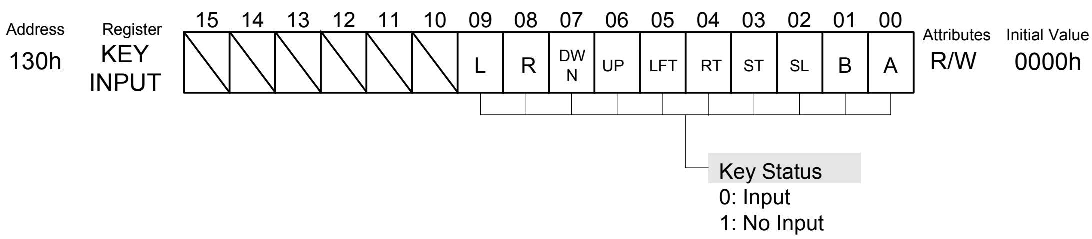
14 Key Input

14.1 Key Status

Game Boy Advance allows input with the L and R Buttons, as well as with the START and SELECT buttons, the +Control Pad, and A and B Buttons.

The status of each of these buttons can be checked by reading the individual bits of Register KEYINPUT.

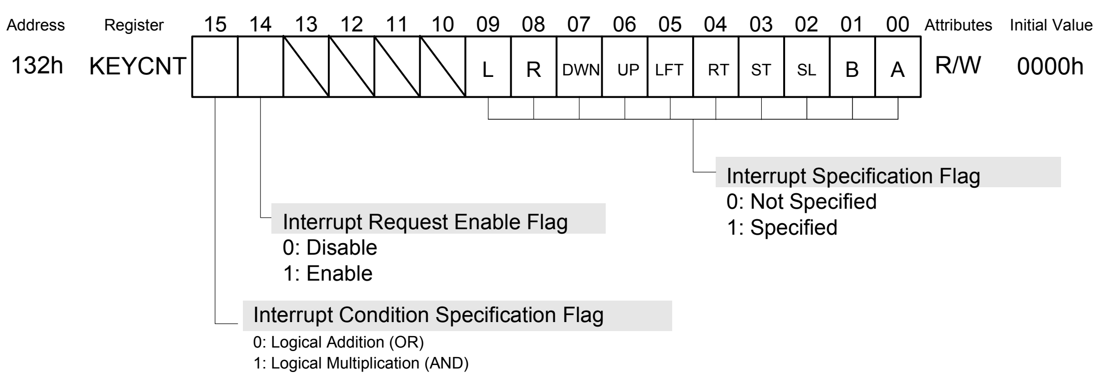
Figure 122 - The KEYINPUT Register



14.2 Key Interrupt Control

When an interrupt is performed for key input, this register enables a target key combination or condition for the interrupt to be specified.

Figure 123 - The KEYCNT Register



14.2.1 Interrupt Conditions

Specifies interrupt generation conditions when the interrupt enable request flag is true.

The conditions for buttons selected with the key interrupt specification flag can be selected as follows.

1. Logical Addition (OR) Operation

   The conditions for interrupt request generation occur when there is input for any of the buttons specified as interrupts.

2. Logical Multiplication (AND) Operation

   The conditions for interrupt request generation occur when there is simultaneous input for all of the keys specified as interrupt keys.

Key Input Cautions

Key bounce (chattering) may occasionally occur when a user presses a button. In order to prevent a button's function from being called multiple times, we recommend that you program an interval (i.e., once per frame) when reading a key.

15 Interrupt Control

Game Boy Advance can use 14 types of maskable hardware interrupts. If an interrupt request signal is received from a hardware item, the corresponding interrupt request flag is set in the IF register. Masking can be performed individually for interrupt request signals received from each hardware item by means of the interrupt request flag register IE.

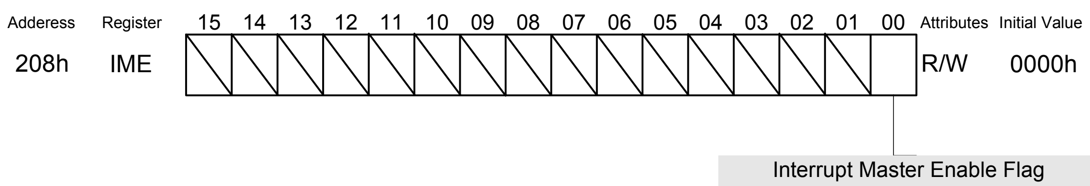
15.1 Interrupt Master Enable Register

The entire interrupt can be masked.

When this flag is 0, all interrupts are disabled.

When 1, the setting for interrupt enable register IE is enabled.

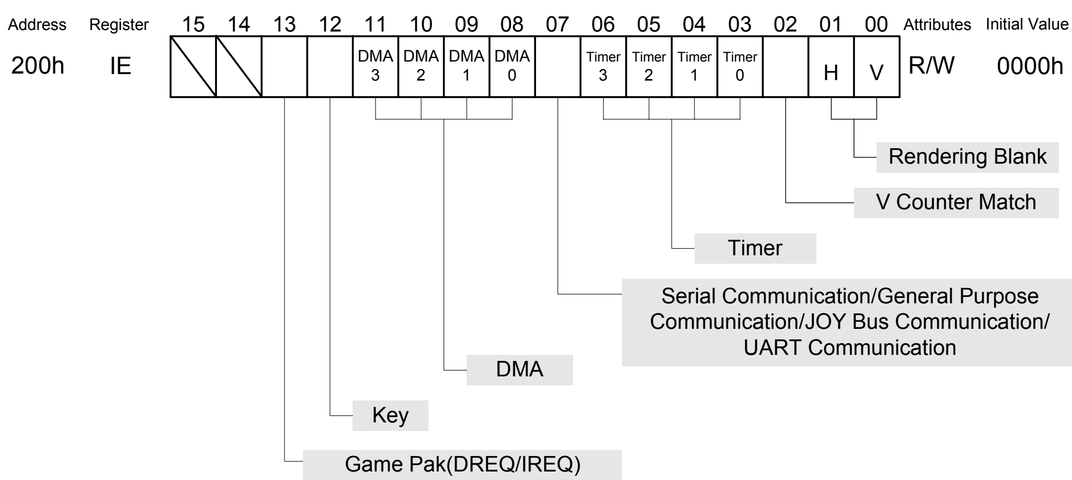
Figure 124 - The IME Register



15.2 Interrupt Enable Register

With the interrupt enable register, each hardware interrupt can be individually masked.

Figure 125 - The IE Register

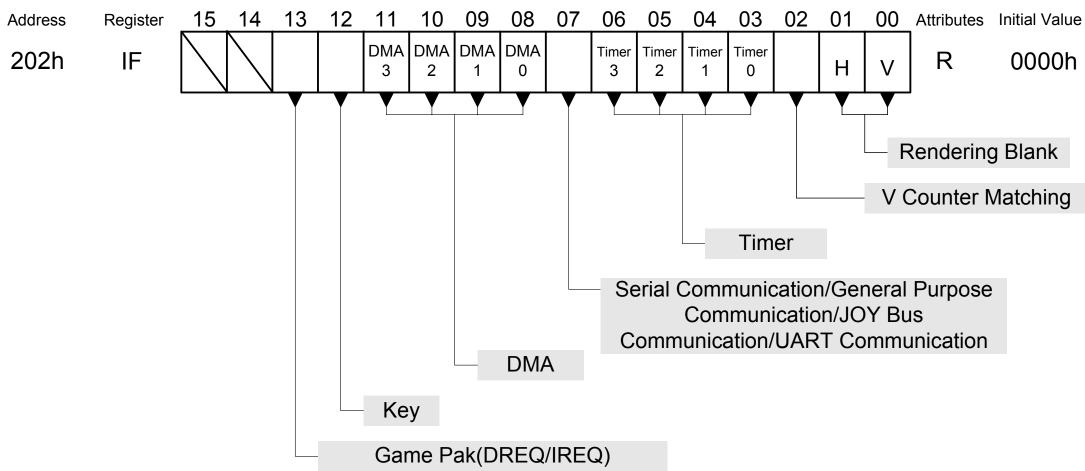


By resetting the bit, the corresponding interrupt can be prohibited. Setting this to 1 enables the corresponding interrupt.

15.3 Interrupt Request Register

When an interrupt request signal is generated from each hardware device, the corresponding interrupt request flag is set in the IF Register.

Figure 126 - The IF Register



If a 1 is written to the bit which the interrupt request flag is set in, that interrupt request flag can be reset.

15.3.1 About H-Blank Interrupts

H-Blank interrupt requests also occur during V-Blank.

15.3.2 About Game Pak Interrupts

An interrupt request occurs when the IREQ terminal is “High”.

Although the IREQ terminal is pulled “High” in the Game Boy Advance hardware, the IREQ terminal is set to “Lo” when a normal Game Pak is installed. Therefore, the IREQ terminal is pulled “High” and an interrupt request occurs when the Game Pak is removed from the Game Boy Advance.

15.3.3 Cautions Regarding Clearing IME and IE

A corresponding interrupt could occur even while a command to clear IME or each flag of the IE register is being executed.

When clearing a flag of IE, you need to clear IME in advance so that mismatching of interrupt checks will not occur.

When Multiple Interrupts are Used

When the timing of clearing of IME and the timing of an interrupt agree, multiple interrupts will not occur during that interrupt. Therefore, set (enable) IME after saving IME during the interrupt routine.

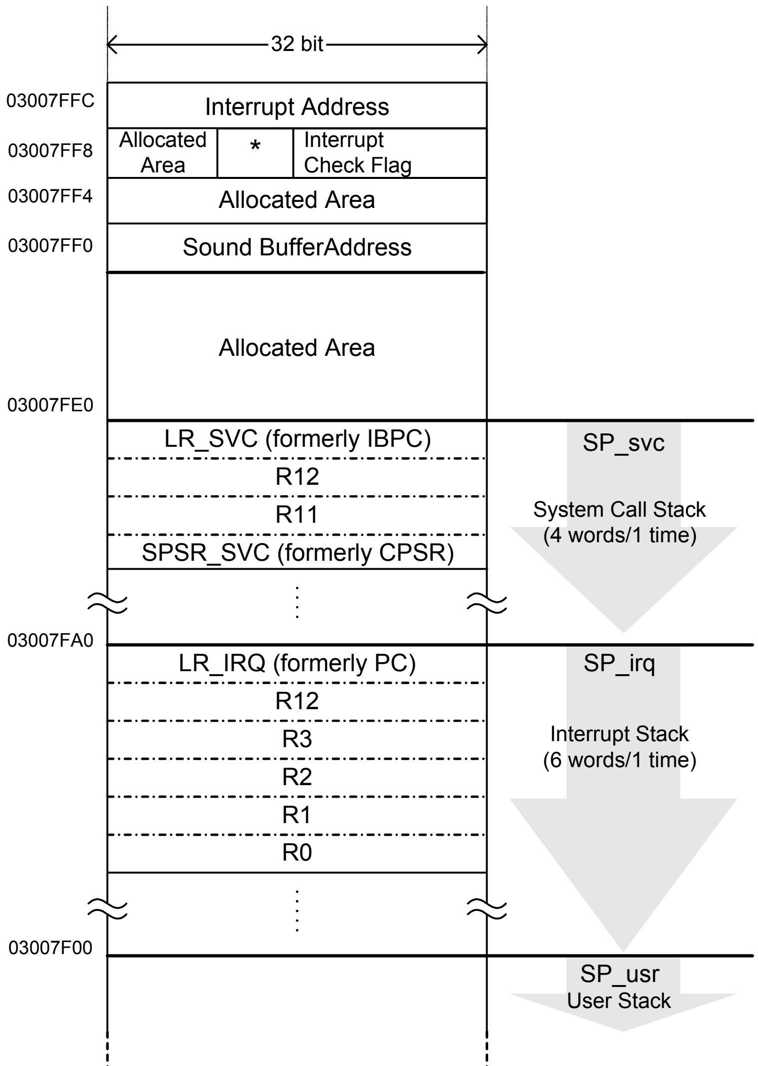
15.3.4 Other Cautions

The CPU stops during DMA transfer. Therefore, interrupts cannot take place until the DMA transfer has completed (there is a delay until the DMA has finished).

15.4 System-Allocated Area in Work RAM

Controlling interrupts entails, along with clearing the IF register and setting the IE register, first writing an interrupt jump address at addresses $7FFC-$7FFF (total of 32 bits; see figure below) in the system allocated area of Work RAM. Processing is executed in 32-bit mode for the user interrupt. To return control from the interrupt routine to the user program, the instruction “BX LR” is used.

Figure 127 - System-Allocated Area in Work RAM



* Specify where to return for SoftReset( ) System Call
If 0h:08000000h
If not 0h:02000000h

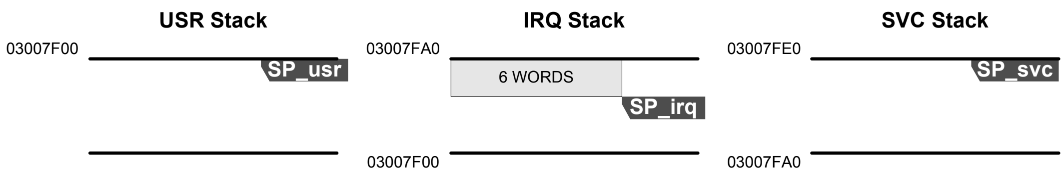
By changing each CPU Mode SP Initial-value, they can be set to an arbitrary memory map.

15.5 Interrupt Operation

The user can arbitrarily define the Interrupt Processing Routine, but as a general rule, the Monitor ROM handles this processing. For further details on each register, please refer to “ARM7TDMI Data Sheet”.

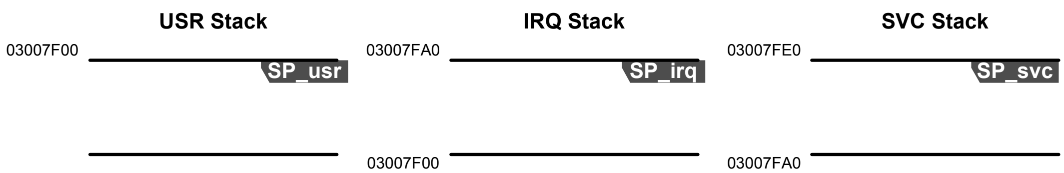
15.5.1 Normal Interrupt

1. If an interrupt occurs, the CPU enters IRQ mode and control shifts to the Monitor ROM. In Monitor ROM, save each register (R0~R3, R12, LR_irq (former PC)) to the Interrupt Stack. The total is 6 words. Next, call the user interrupt processing set up in 03007FFCh. Commands called from the monitor directly must be in 32bit code format.



2. User interrupt processing is done (you can reference the cause of the interrupt with the IF Register). Also solve* problems with a stack, if necessary.

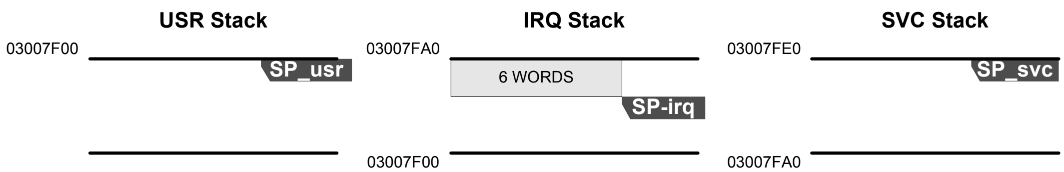
3. Restore the registers (total of 6 words) saved to the Interrupt Stack and return to user main processing.



Note: Only the interrupt stack is used for normal interrupt processing. Therefore, there is a possibility of stack overflow in some cases. To solve this problem, you can either allocate a larger interrupt stack by moving SP_usr in advance or use user stack for both, by switching the CPU mode to the user mode in user interrupt processing. For the latter method, see "15.5.2 Multiple Interrupts" on page 139.

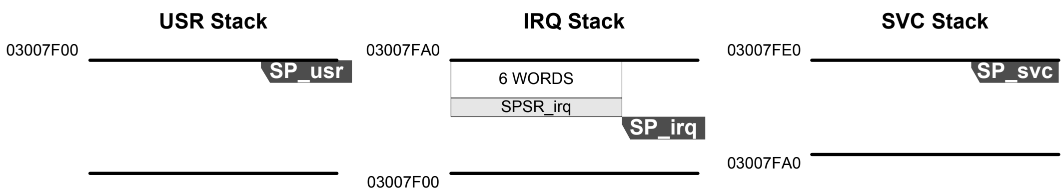
15.5.2 Multiple Interrupts

1. If an interrupt occurs, the CPU enters IRQ mode and control shifts to the Monitor ROM. In Monitor ROM, save each register (R0~R3, R12, LR_irq (former PC)) to the Interrupt Stack. The total is 6 words. Next, call the user interrupt processing set up in 03007FFCh. Commands called from the monitor directly must be in 32bit code format.

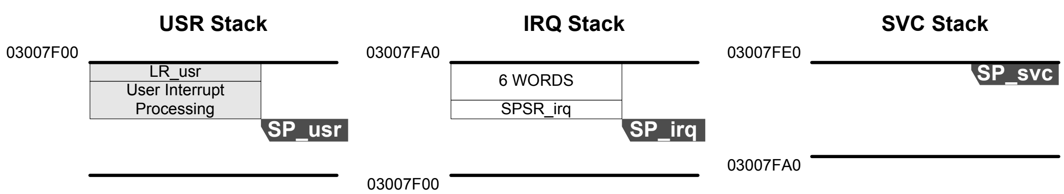


2. User interrupt processing is done (you can reference the cause of the interrupt with the IF Register).

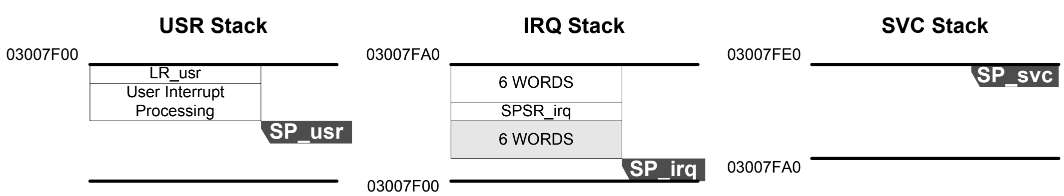
   - If multiple interrupts occur, SPSR_irq will be overwritten, so you must save before enabling IRQ.



   - The Stack problem is solved* (CPU mode is changed to user mode with system mode = privilege here.) and IRQ is enabled.
   - With user interrupt processing, user stack is used because the CPU is in system mode. When calling the subroutine, save LSR_usr as well.



   - When an interrupt occurs, Monitor ROM does the processing (1) again, and loads each register to the interrupt stack.



   - Continue processing (2).

16 Power-Down Functions

16.1 Stop Function

16.1.1 Stop Function Summary

During periods when the LCD display is not done and CPU processing is not considered essential you can reduce power consumption greatly if used efficiently.

The content of each type of RAM are maintained.

16.1.2 Implementing Stop

1. Implementation of Stop Mode

   Game Boy Advance is placed in stop mode by executing the system call [SWI <3>] instruction (Stop( )).

2. Canceling Stop Mode

   If the corresponding flag of the interrupt enable register IE is set for various interrupt requests of Key, Game Pak, and SIO (general-purpose communication mode only), Stop Mode is canceled. For example, if key input conditions set by the KEYCNT register are met, or by Hi input to IREQ/DREQ terminal or Lo input to SI terminal in general-purpose communication mode. However, because Clock is stopped, the applicable flag of Interrupt Request Register IF will not be set.

   Note: Canceling stop status requires a brief wait until the system clock stabilizes.

16.1.3 System Working Status in Stop Mode

The working status of each block of the Game Boy Advance system during a stop is shown in the following table.

Table 35 - System Status while in Stop Mode

Block	Working	Status
GBA CPU	X	Wait status resulting from wait signal
LCD Controller	X	Stopped because no clock provided*
Sound	X	Stopped*
Timer	X	Stopped
Serial Communication	X	Stopped
Key	X	Stopped
System Clock	X	Stopped
Infrared Communication	X	Stopped

Note: The LCD controller stops so turn OFF the LCD display before entering Stop Mode. Sound stops in Stop Mode, therefore noise may result. So stop the sound first, then enter Stop Mode.

16.1.4 Stop Function Cautions

- The only way to distinguish the stopped state from the power off state is to observe the Power lamp. Therefore, it is possible that a user may fail to notice the stopped state.

  Specifically, it is possible that a battery may die as a result of being left in the stopped state for long hours, or that power is turned off while in the stopped state.
- Please ensure that a problem does not occur in this situation.
- Do not use the stop function when writing to a backup device.
- Do not use the stop function during communication. (Use it after terminating communication.)
- When using the stop function, enable the Game Pak interrupt first. When Stop is cancelled by removing the Game Pak, stop the processing (i.e., infinite loop, etc.).
- Be sure to adhere the guidelines described in the next section when using this function.

16.1.5 Guidelines for Use of the Stop Function

Use the term “Sleep Mode” or “Sleep” when describing the Stop Function in Instruction Booklets and game screen text.

16.1.5.1 Entering Sleep Mode

- Entering Sleep Mode from the Menu Screen:

  Follow the examples listed in the table below for both on-screen text and Instruction Booklet text:

Table 36 - Terminology for Entering Sleep Mode from the Menu Screen

Language	Term
Japanese	スリープ
English	Sleep
German	Standby
French	Veille
Spanish	Salvapantallas
Italian	Riposo
Dutch	Standby

Be sure to display a way to exit Sleep Mode. For example, “To show the menu again, press SELECT + L Button + R Button.”

- Entering Sleep Mode via a Button Shortcut

  Follow the examples listed in the table below for both on-screen text and Instruction Booklet text:

Table 37 - Terminology for Entering Sleep Mode Using a Buttons Shortcut

Language	Term
Japanese	スリープコマンド
English	Easy Sleep
German	Standby-Bereitschaft
French	Veille rapide
Spanish	Salvapantallas directo
Italian	Riposo Rapido
Dutch	Standby

  For the button shortcut, always use the following button combination: SELECT + L Button + R Button.

  This is the only acceptable button combination for entering Sleep Mode. Design your application so that it does not go into Sleep Mode when the user presses other buttons.

  By waiting to enter into Sleep Mode until all three pressed buttons (SELECT + L + R) have been released, you can avoid an unexpected exit from Sleep Mode.

- Automatically Entering Sleep Mode

  Follow the examples listed in the table below for both on-screen text and Instruction Booklet text:

Table 38 - Terminology for Automatically Entering Sleep Mode

Language	Term
Japanese	オートスリープ
English	Auto Sleep
German	Auto-Standby
French	Veille automatique
Spanish	Auto-Salvapantallas
Italian	Autoriposo
Dutch	Auto Standby

  Create a Menu Screen Option and enable this setting to be toggled ON and OFF.

  Set the initial setting to OFF (The user may mistakenly think their system is broken if the Game Boy Advance goes into sleep mode automatically without their prior knowledge.)

  On the Menu Screen, display the way to exit Sleep Mode.

16.1.5.2 Exiting Sleep Mode

Exit from sleep mode via the simultaneous pressing of SELECT + L + R.

By waiting until all three pressed buttons have been released before exiting sleep mode, you can avoid an unexpected entry into Sleep Mode again.

16.2 Halt Function

16.2.1 Halt Function Summary

During periods when CPU processing is not considered essential you can reduce power consumption if used efficiently.

16.2.2 Halt Transition Method

1. Transition to Halt Mode

   Game Boy Advance is placed in halt mode by executing the system call [SWI <2> instruction (Halt( )).

   Game Boy Advance enters Halt status.

2. Cancel Halt Mode

   Halt is canceled when the interrupt enable register IE's corresponding flag is set with any type of interrupt request.

16.2.3 System Working Status in Halt Mode

The working status of each block of the Game Boy Advance system during a semi-stop is shown in the following table.

Table 39 - System Status while in Halt Mode

Block	Working	Status
GBA CPU	X	Wait status resulting from wait signal
LCD Controller	O	Normal operation
Sound	O	Normal operation
Timer	O	Normal operation
Serial Communication	O	Normal operation
Key	O	Normal operation
System Clock	O	Normal operation

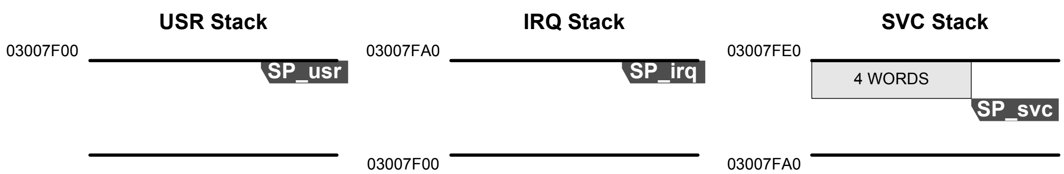
17 Game Boy Advance System Calls

Please refer to the *AGB System Call Reference Manual* for Game Boy Advance system calls.

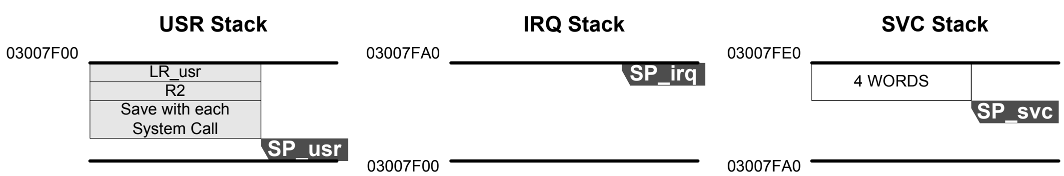
17.1 System Call Operation

17.1.1 Normal Calls

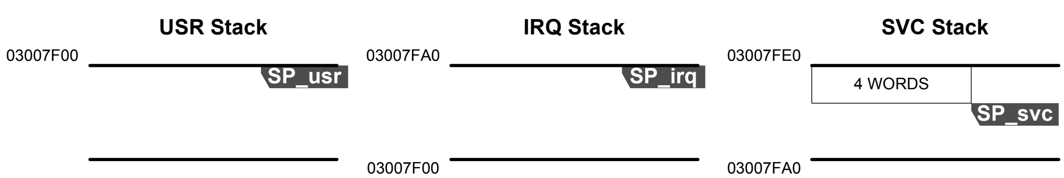
1. When an argument is required for the system call used, after writing to registers R0-R3 call the monitor ROM system call with the “SWI<Number>”. The CPU mode changes to Supervisor Mode.
2. Save the registers, SPSR_svc (formerly CPSR), R 11, R12, LR_svc (formerly PC) to the system call stack with the monitor ROM.



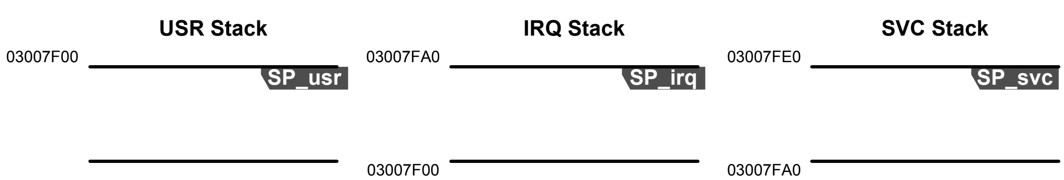
3. Switch from CPU mode to system mode. Call the IRQ disable flag with monitor ROM. The previous status will continue.
4. Save the R2 and LR_usr registers to the user stack. Other registers will be saved with each system call.



5. Complete processing using each system call.

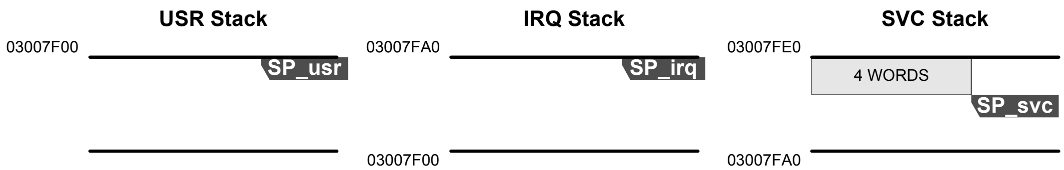


6. Return value to registers R0, R1, and R3, in cases where a system call provides a return value, and then return to the user program.

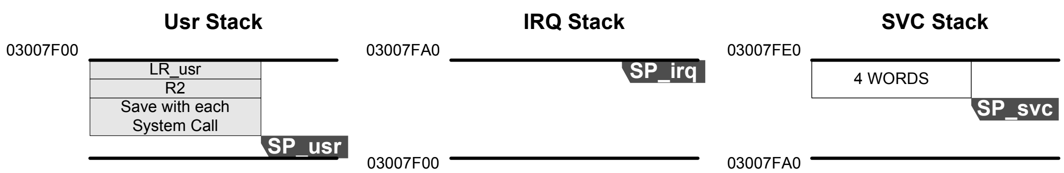


17.1.2 Multiple Calls

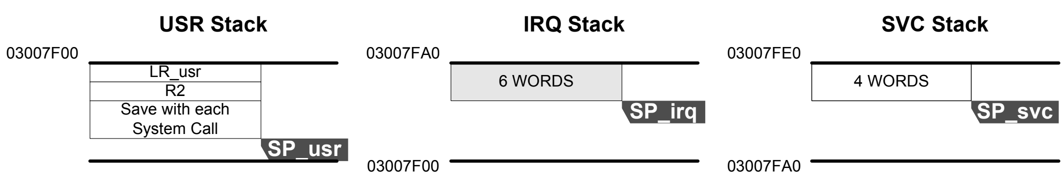
1. When an argument is required for the system call used, after reading to the registers, R0-R3, call the monitor ROM system call with the “SWI<Number>”.
2. Save the registers, SPSR_svc (formerly CPSR), R12, LR_svc (formerly PC) to the system call stack with the monitor ROM.



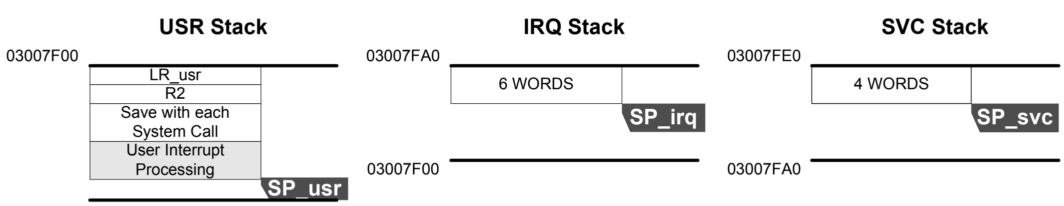
3. Switch from CPU mode to system mode. The status of the IRQ Disable Flag prior to the call is kept in System ROM. The previous conditions will be continued.
4. Save the R2 and LR_usr registers to the user stack. Other registers will be saved with each system call.



5. Interrupt occurs while executing system call.

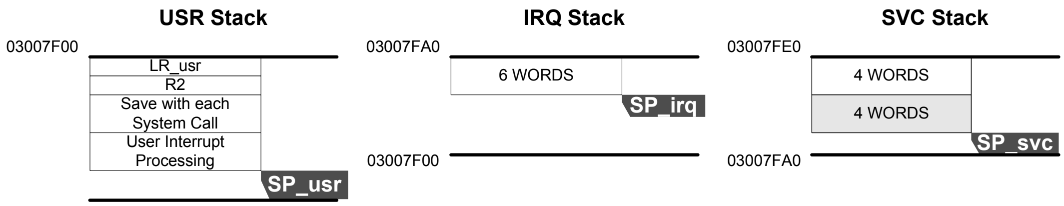


6. User interrupt processing is done. (You can reference the cause of the interrupt with the IF Register.) The CPU mode is changed to System Mode (User Mode with privilege) in order to solve the problem with stacks (to reference interrupt processing).

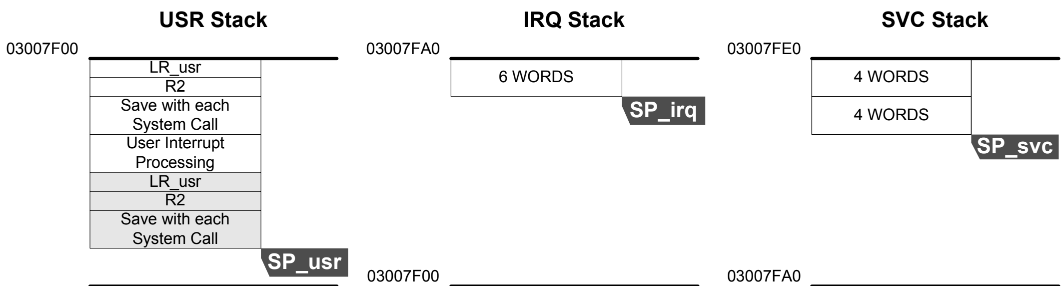


- If the System Call occurs during User interrupt processing, the System Call is called using Multiple Calls.

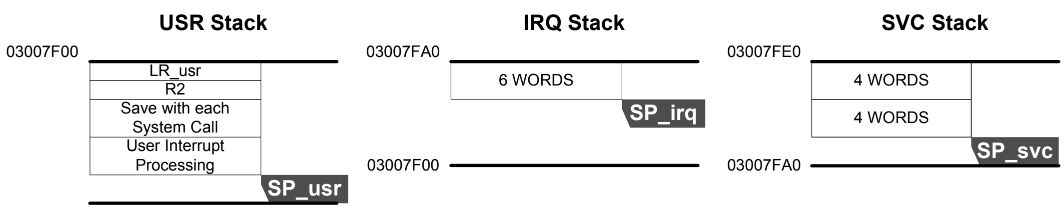
7. Monitor ROM does the system call operation (1), and loads to the system call stack.



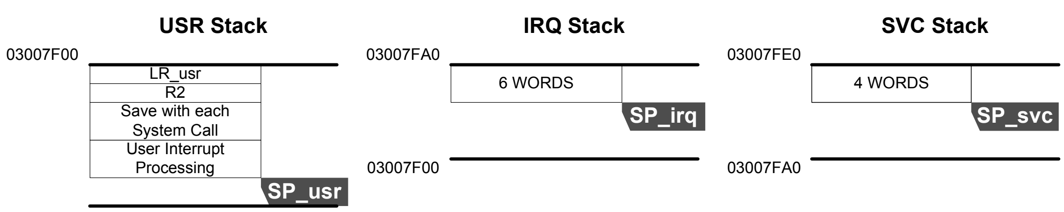
8. Switch the CPU Mode to System Mode (privileged user mode).
9. Monitor ROM does the same operation as (3), and loads to the user stack.



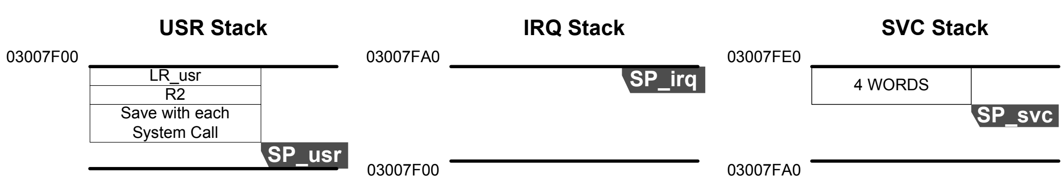
10. Complete processing with each system call.



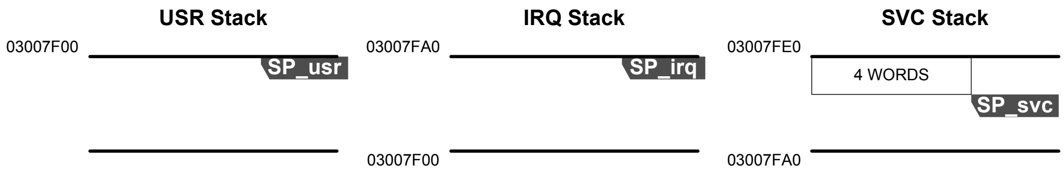
11. Return value to registers R0, R1, and R3, in cases where a system call provides a return value, and then return to the user interrupt processing.



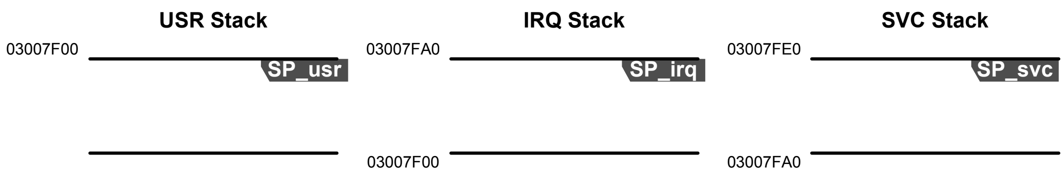
12. Complete the user interrupt processing and return to the previous system call.



13. Complete processing with each system call.



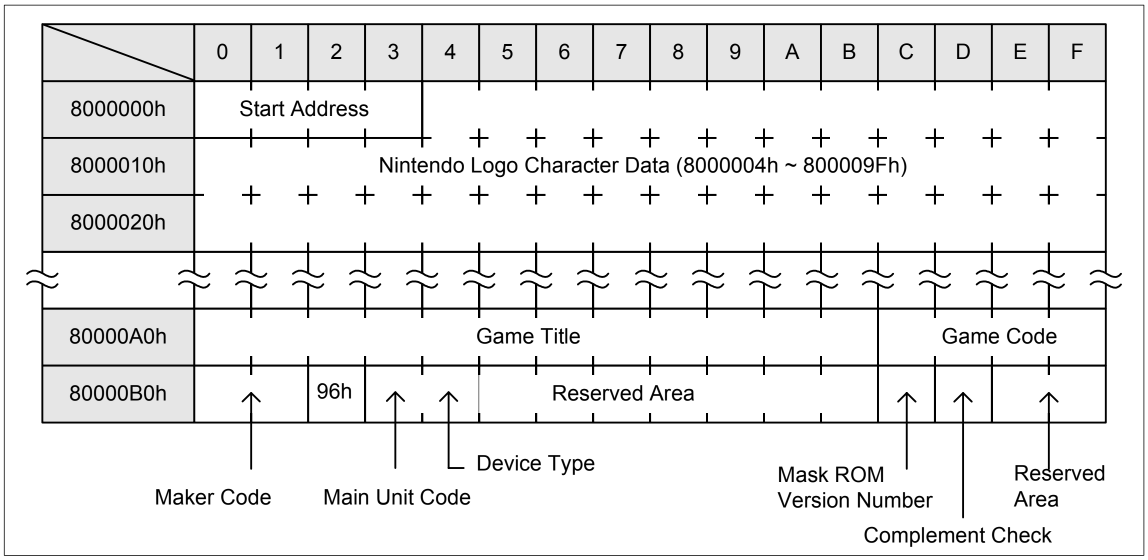
14. Return value to registers R0, R1, and R3, in cases where a system call provides a return value, and then return to the user program.



18 ROM Registration Data

As with software for CGB, it is necessary to register information about the game in the program area for Game Boy Advance software.

Figure 128 - Game Boy Advance ROM Registration Data



18.1 Start Address

Store the 32-bit ARM command “B<User program start address>”.

18.2 Nintendo Logo Character Data

The Nintendo logo/character data, which is displayed when the game is started, is stored here. The Monitor ROM checks this data at start-up, therefore always store the data provided by Nintendo.

18.3 Game Title

Store the Game title in this area.

18.4 Game Code

Store the Game Code provided by Nintendo in this area.

18.5 Maker Code

The Maker Code, determined by the “maker” of the software and Nintendo, is stored here.

18.6 96h

Store the fixed code “96h”.

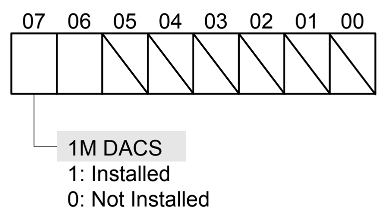
18.7 Main Unit Code

Store the code for the hardware on which the software is intended to run.

18.8 Device Type

Store the type of device that is installed in the Game Pak. If there is a 1-megabit flash DACS (Debugging And Communication System) (=custom 1Mbit flash Memory with security and patch functions) in a Game Pak, set the applicable bit to 1. Otherwise it is reset (see the illustration below). Other bits are system allocated area.

Figure 129 - Device Type Bits



18.9 Mask ROM Version No.

Store the ROM version number here.

18.10 Complement Check

The 2’s complement of the total of the data stored in address 80000A0h ~ 80000BCh plus 19h is stored in this location.

18.11 Reserved Area

This is a system allocated area. Set this area to 00h.