CHAPTER 6: THE SUPER GAME BOY SYSTEM

1. OVERVIEW

1.1 What is Super Game Boy (SGB)?

SGB is a device that enables Game Boy software to be enjoyed on a TV screen. Game Boy software can be plugged into the SGB, which operates on the Super Nintendo Entertainment System (Super NES).

SGB consists of the basic Game Boy circuitry, and components such as an Intercommunication Device (ICD, with built-in SGB RAM), the system program, and a CIC.

Basic SGB operation involves conversion by the ICD of 2-bit, 4 grayscale image signals generated by the SGB CPU to SUPER NES character data and storage of these data in SGB RAM. The system program subsequently transfers this data by DMA to SUPER NES WRAM and then to VRAM. The above operations are performed repeatedly to display the Game Boy screen on a TV screen.

Unmodified sound output from the SGB CPU is linked to the SUPER NES sound mixing circuit and is output from the speaker on the TV.

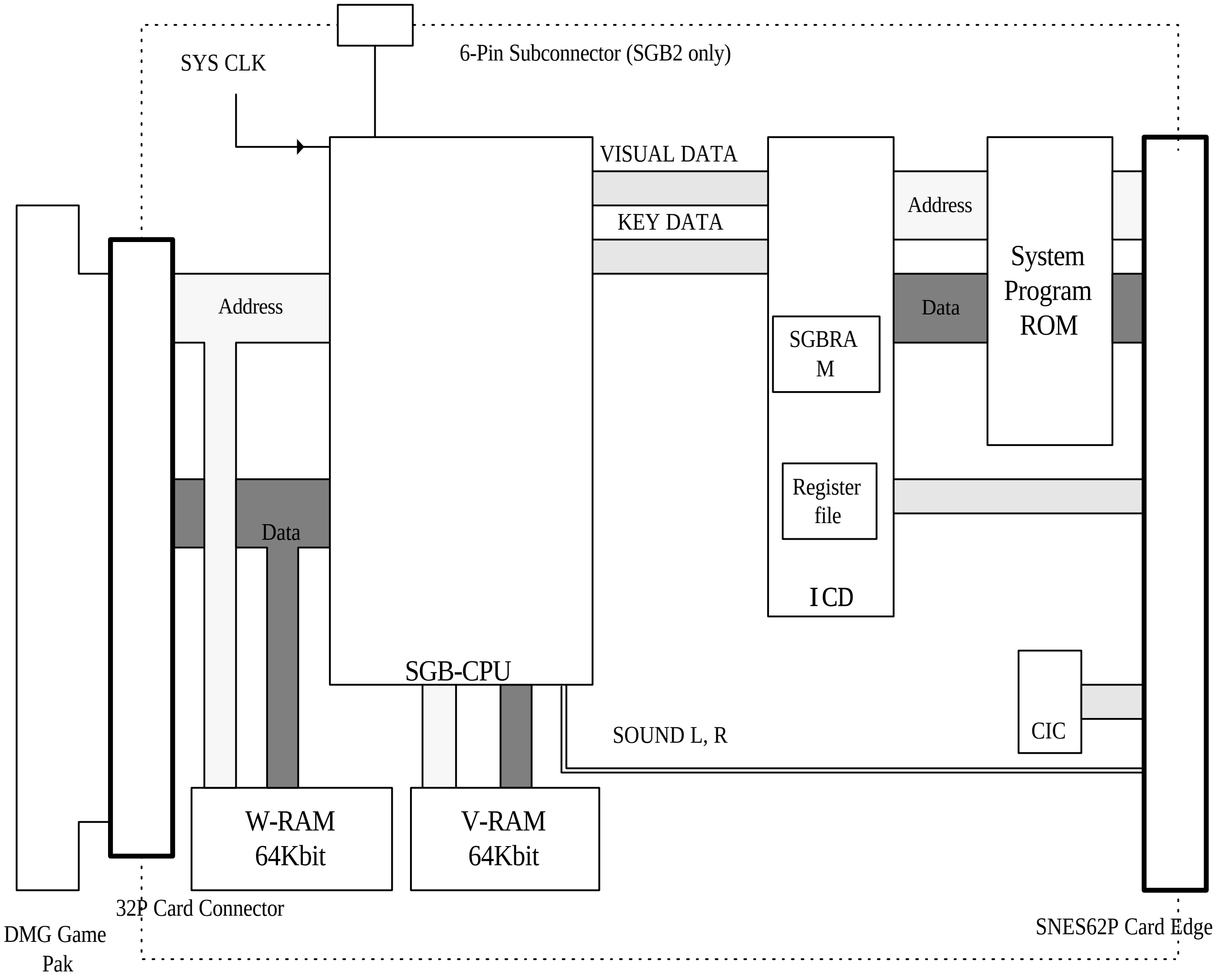
These operations are controlled by the SGB system program and therefore require no special consideration when programming for Game Boy.

Game Boy software not specifically created for SGB provides 4 colors in 4 grayscales. These colors are selected from several color patterns provided in the system program. Programming using the system commands described later allows a game to be represented using 4 palettes of 4 colors each per screen and SUPER NES functions such as SUPER NES sound.

Super Game Boy comes in 2 models: the 1994 model, which has no communication connector, and the 1998 model, which is equipped with a communication connector.

This manual uses the term SGB2 when discussing points that concern only the 1998 model. Descriptions that use the term Super Game Boy or SGB refer to both Super Game Boy models. SGB2 allows game representations that use SHVC functions for communication play. (SGB2 has not been released in the U.S. market.)

1.2 Block Diagram



1.3 Functions

The types of representations indicated below can be implemented using SUPER NES functions invoked by sending system commands.

For more information, please see Section 3 in this chapter, *System Commands*.

Image Functions

- Up to 4 palettes of 4 colors each can be represented on a single screen.
- Multiple areas can be specified for each screen, and separate color palette attributes can be specified for each area.
- Color palette attributes can be specified separately for each character (8 x 8 bits).

Sound Functions

- The rich variety of sound effects included the system program can be generated by the SUPER NES audio processing unit (APU).
- The sound generator included in the system program can be used by transferring music data.

Controller Functions

- Data from multiple SUPER NES controllers data can be read, providing for multiplayer games that can accommodate between 2 and 4 players.

Miscellaneous

- SUPER NES program data can be transferred.

1.4 System Program

The system program can provide the following features.

- On the T.V. screen, the system program displays the space outside the game screen (picture frame).

The picture frame has the following features.

- The frame can be selected from among 9 pre-loaded frames.
- A mode in which an image created by the game producer is transferred and displayed as the frame.
- A drawing mode that allows the user to create the frame.

Features of the color palette selection screen are as follows.

- Palettes can be selected from among 32 pre-loaded palettes.
- A mode that allows colors to be set from DMG in DMG games.

A mode is available that allows the user to arrange the colors on a palette.

A screen is provided for changing the key configuration of the controller.

- If the commands described in Section 3.2 in this chapter, *System Command Details*, are sent to the register file, Super NES functions, such as those described in Section 1.3, *Functions*, can be used by having the system program read these commands.

2. SENDING COMMANDS AND DATA TO SUPER NES

The following 2 methods can be used to send data from a DMG program to Super NES.

Send data to the register file using P14 and P15. The size of the register file is 128 bits; this is referred to as 1 packet.

Send data to SGB RAM using an image signal.

NOTE Data transfers from the register file and SGB RAM to SUPER NES are performed by the system program.

2.1 System Commands

Using the register file to transmit system commands allows the various SUPER NES functions described below to be used in games.

The system program receives the commands and performs the specified processing.

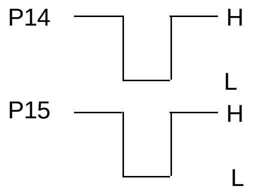
- Data Format of System Commands

1) Data Transmission Methods

Using 2 bits in SGB (P14 and P15 of SGB CPU), data is sent to the register file by serial transmission.

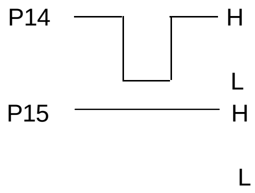
The system program reads the contents written to the register file.

1. Start write



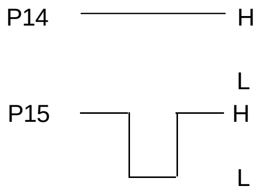
A LOW pulse is output to both P14 and P15.
This is required for transmission of each packet (128 bits) .

2. Write 0

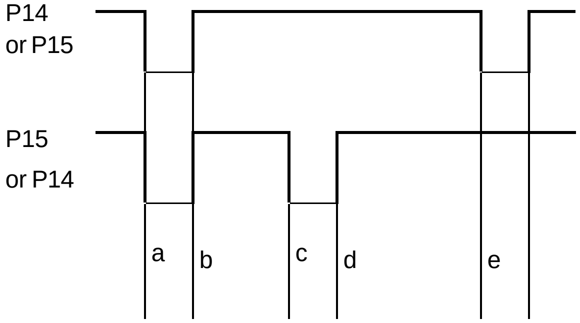


P15 is fixed at HIGH, and a
LOW pulse is output to P14.

3. Write 1

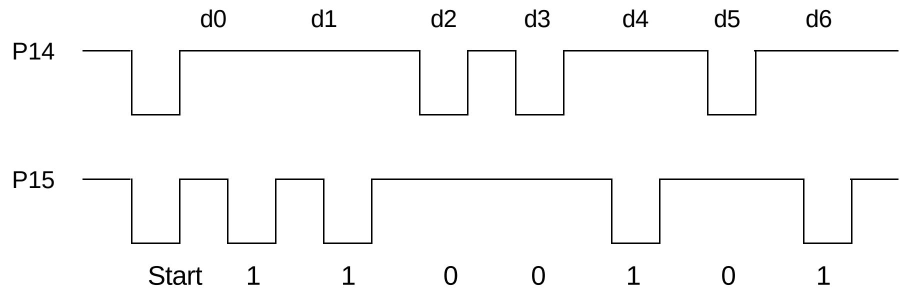


P14 is fixed at HIGH, and
a LOW pulse is output to P15.

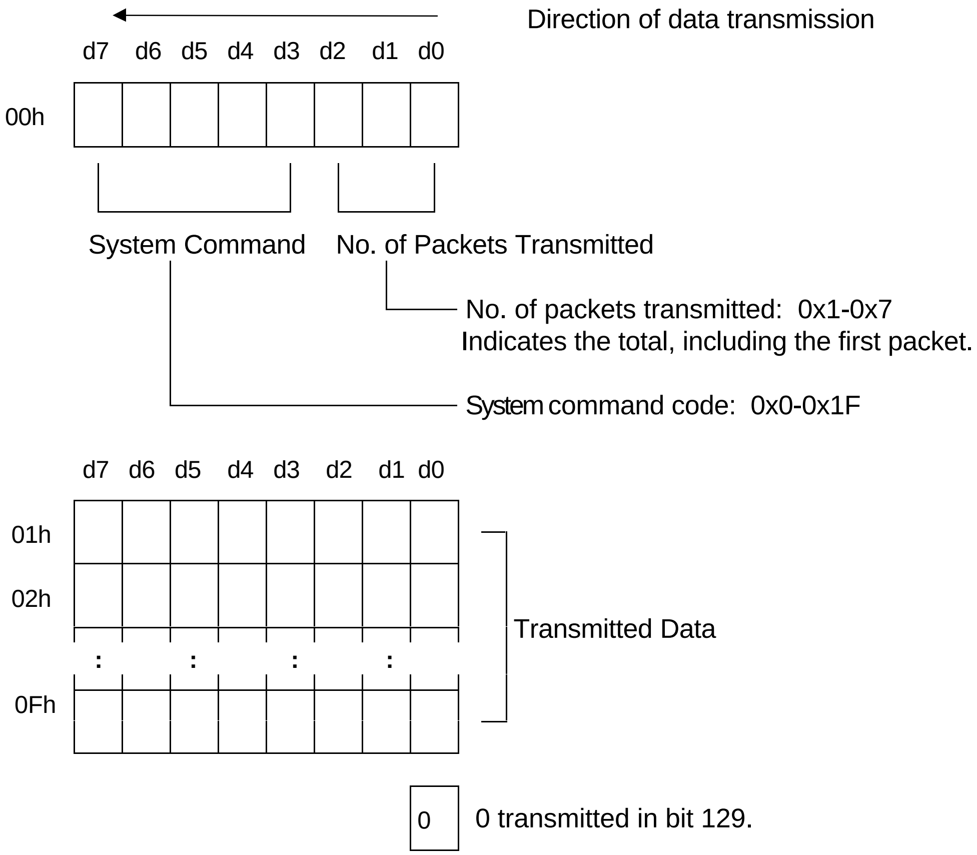


Pulse Width	
a, c, e	5 μ s (min)
b, d	15 μ s (min)

2) Write Example



3) Format of Data Transmitted to Register File



No. of packets transmitted: 0x1-0x7
Indicates the total, including the first packet.

System command code: 0x0-0x1F

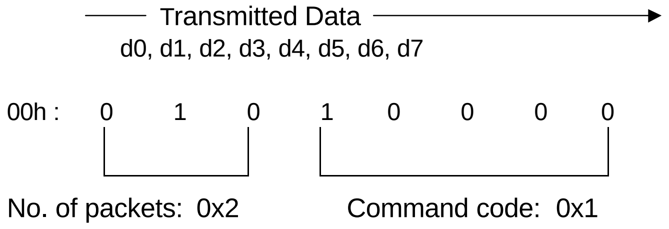
0 transmitted in bit 129.

If 2 or more packets are used for one system command, bits 0x00-0xF of the second packet onward are used for data.

Transmission Procedure

1. Start of write

2. Data transmission (example)



0x01: data
0x02: data
: :
: :
0xF: data

3. Transmission of 0 in bit 129

   Bit 129: 0

4. Start of write

5. Data transmission: second packet

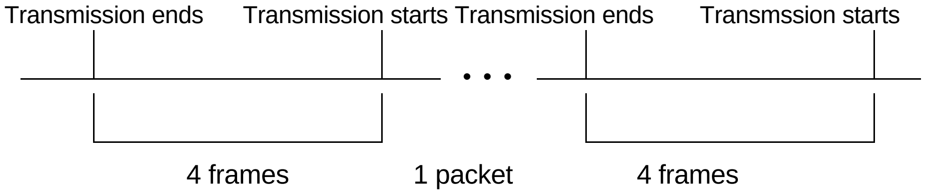
0x00: data
0x01: data
: :
: :
0xF: data

1. Transmission of 0 in bit 129

   Bit 129: 0

4) Transmission Interval

The interval between completion of transmission of one packet (128 bits + 1 bit) and transmission of the next packet is set at approximately 60 msec (4 frames).



5) Transmission Bit 129

The data in bit 129 marks the end of one packet, so it should always be transmitted.

2.2 Data Transfer Using an Image Signal

Data and programs stored in a cartridge can be transferred using the image signal transmission path (LD0, LD1).

Character data stored in DMG VRAM and displayed are then stored in SGB RAM. The system program usually transfers these data to SUPER NES VRAM as character data. However, when a specific command is received, the data is handled as data for command processing.

The displayed image signal is handled directly as data, so be careful to ensure that the OBJ display and window are set to OFF, the correct values are set for the DMB color palette, and the BG to be displayed is correctly transferred.

When data is transferred they are displayed to the screen, so the system command MASK_EN must be used to mask the screen.

For more information, see Section 3.2 in this chapter, *System Command Details*.

Note	***Commands that transfer data using image signals are indicated by the heading, Data Transfer Using VRAM.***

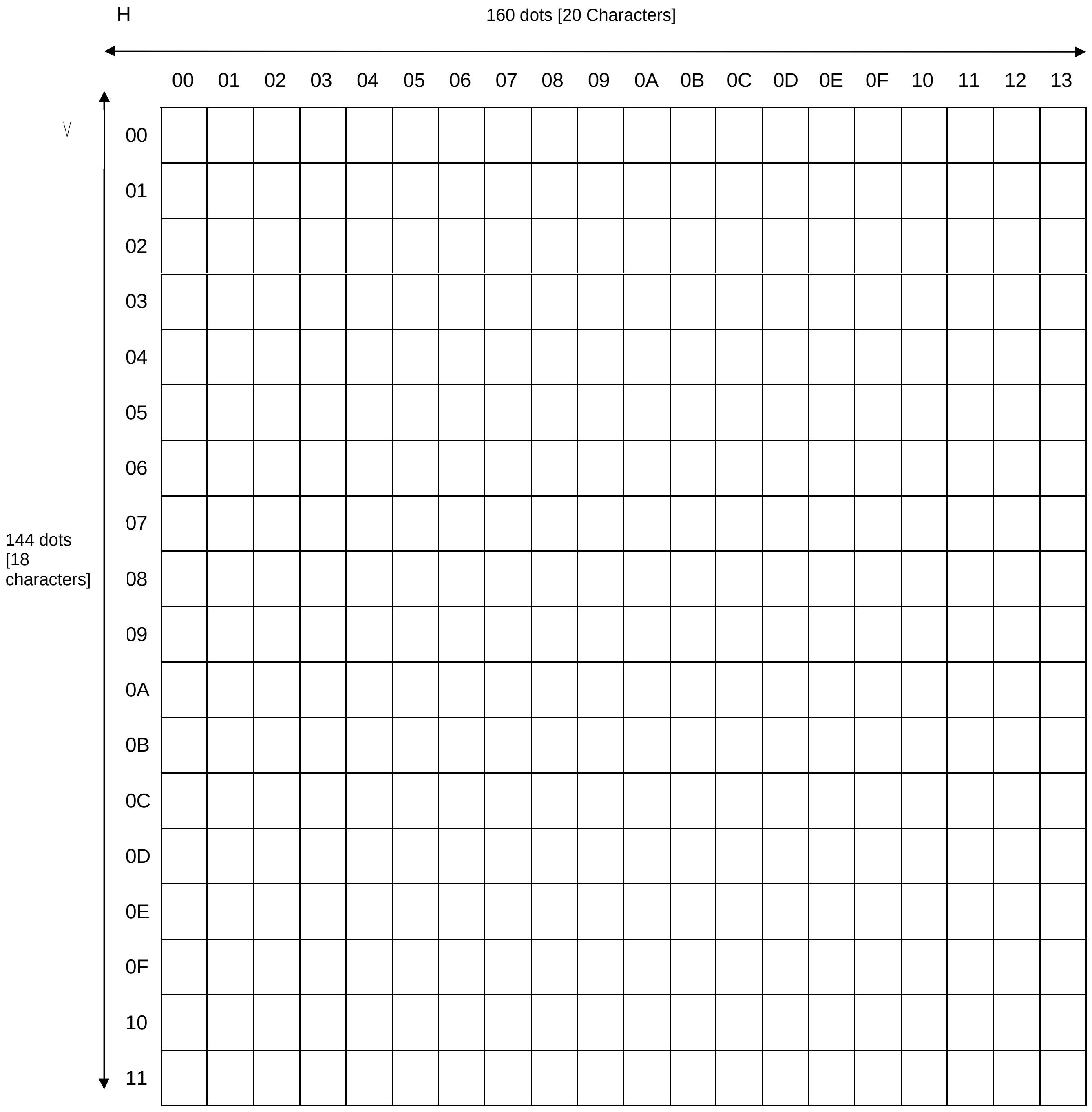
3. SYSTEM COMMANDS

3.1 System Command Summary

Command	Command Code	Command	Command Code
PALO1	00	DATA_TRN	10
PAL23	01	MLT_REQ	11
PAL03	02	JUMP	12
PAL12	03	CHR_TRN	13
ATTR_BLK	04	PCT_TRN	14
ATTR_LIN	05	ATTR_TRN	15
ATTR_DIV	06	ATTR_SET	16
ATTR_CHR	07	MASK_EN	17
SOUND	08	PAL_PRI	19
SOU_TRN	09	Use prohibited	0D
PAL_SET	0A	Use prohibited	18
PAL_TRN	0B		
ATRC_EN	0C		
ICON_EN	0E		
DATA_SND	0F		

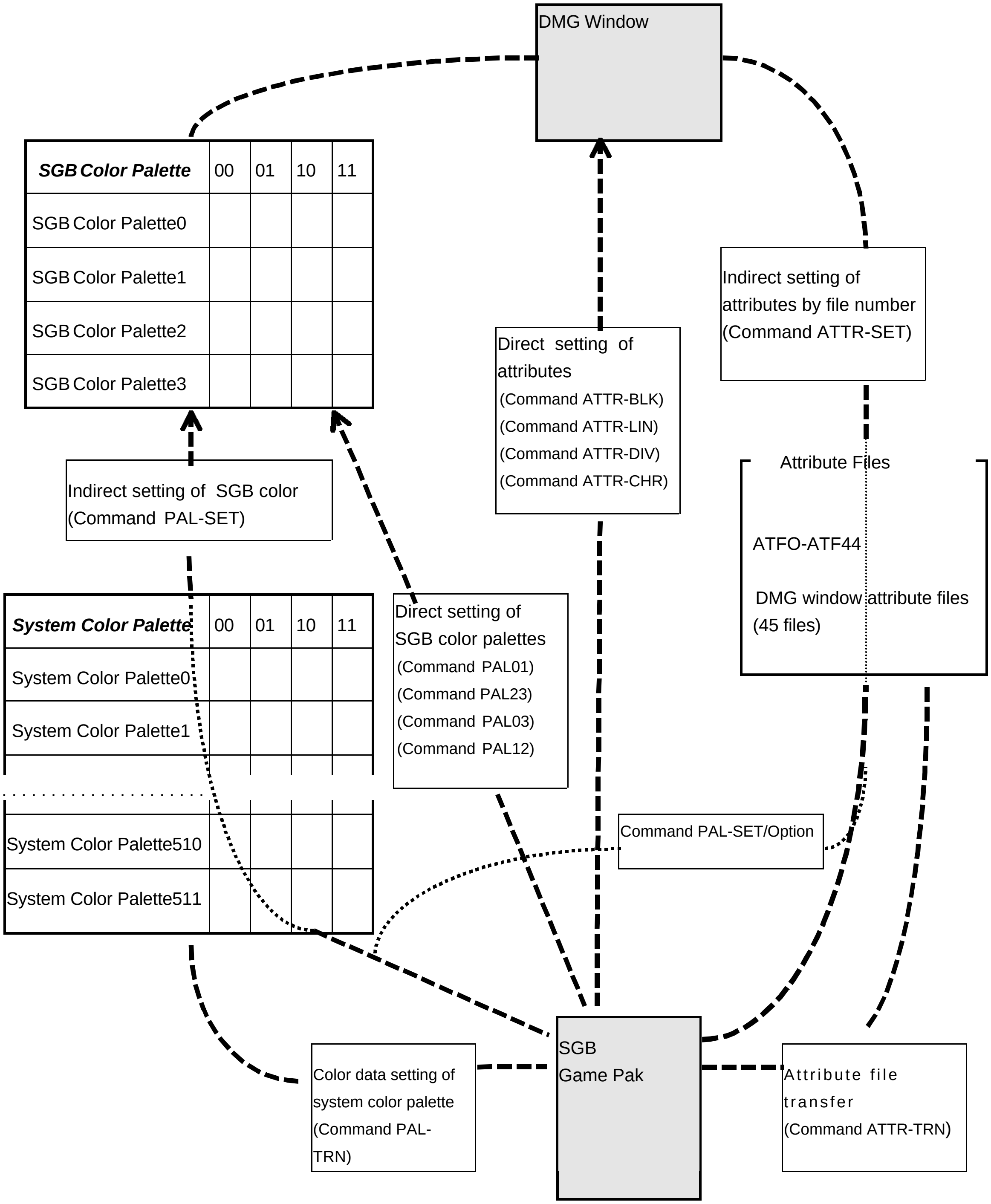
3.2 System Command Details

Please refer to the following map in the discussion of coordinate settings and color palette area specifications in the description of the system command functions.


H
160 dots [20 Characters]
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
V
144 dots [18 characters]
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11

DMG Window

Setting the Color Palettes and Attributes



SGB Color Palette	00	01	10	11
SGB Color Palette0				
SGB Color Palette1				
SGB Color Palette2				
SGB Color Palette3				

System Color Palette	00	01	10	11
System Color Palette0				
System Color Palette1				
. . . .				
System Color Palette510				
System Color Palette511				

DMG Window
Indirect setting of SGB color (Command PAL-SET)
Direct setting of attributes (Command ATTR-BLK) (Command ATTR-LIN) (Command ATTR-DIV) (Command ATTR-CHR)
Indirect setting of attributes by file number (Command ATTR-SET)
Attribute Files
ATFO-ATF44
DMG window attribute files (45 files)
Direct setting of SGB color palettes (Command PAL01) (Command PAL23) (Command PAL03) (Command PAL12)
Command PAL-SET/Option
SGB Game Pak
Color data setting of system color palette (Command PAL-TRN)
Attribute file transfer (Command ATTR-TRN)

Note	***Bit 00 of SGB color palettes 0 – 3 have the same color. The color setting in effect for this bit is the most recent setting.***

DMG Color Palettes and SGB Color Palettes

With DMG screen data representations, colors in SGB are converted from the grayscale data registered in the DMG color palettes, rather than being converted from the bit data for the character.

	00	01	10	11 (m,n)
DMG palette	00	01	10	11

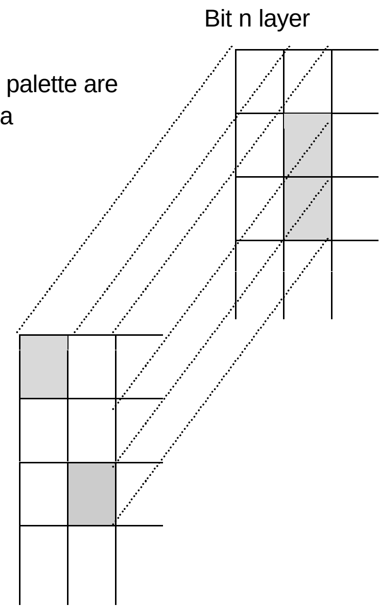
* 00-11 in the palette are grayscale data

light 00> 01> 10> 11 dark

Table 1

	00	01	10	11
SGB palette	Red	Green	Gray	Black

Table 2



Example: When the grayscale data shown in Table 1 are specified for the DMG palette, the character represented on the DMG LCD is as shown in the DMG character image figure below and to the Accordingly, when the color data shown in Table 2 are specified for the SGB palette, the character image represented on SUPER NES is as shown in the SGB character image figure below and to the right.

DMG Character Image

10	00	
00	01	
00	11	

00-11: grayscale data

However, if bit 11 of the DMG palette is set to grayscale 00, the ▨ portion of the DMG character image is displayed with a 00 grayscale, and the ▨ portion of the SGB character image is displayed as red rather than black.

SGB character image

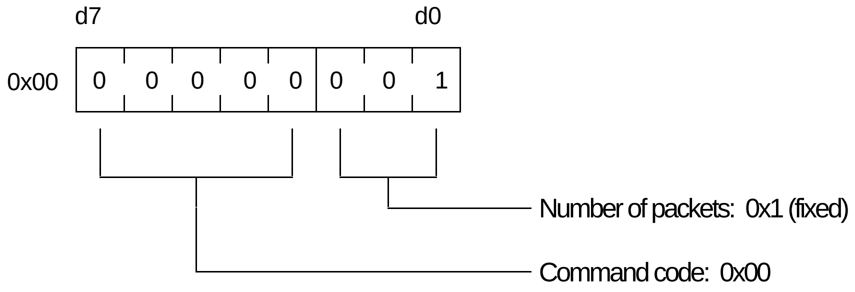
Gray	Red	
Red	Green	
Red	Black	

Thus, in this case, when character data display using all of the colors on the SGB palette is desired, a separate grayscale palette (DMG palette) for SGB must be provided, DMG and SGB must be distinguished, and the program must be made to branch accordingly.
(See Section 4.2, *Recognizing SGB*.)

When representing DMG grayscale on SGB, the image can be faithfully represented if 00 of the SGB palette is set to a light color and 11 to a dark color.

Command: PAL01 (Code: 0x00)

Function: Sets the color data of SGB color palettes 0 and 1.



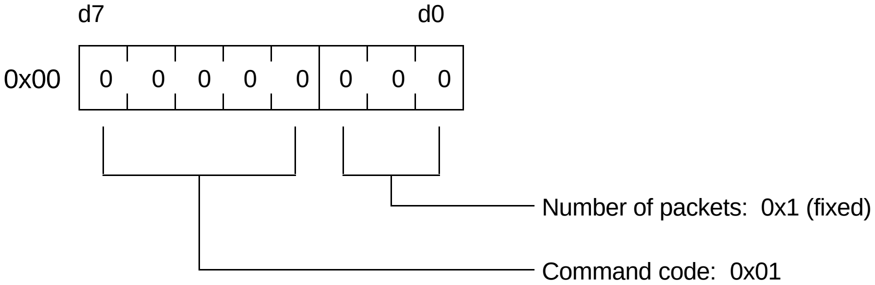
	d7 d0		d7	d0
0x01	Palette0 Color00 Data LOW 8bit	0x02	--	Palette0 Color00 Data HIGH 7bit
0x03	Palette0 Color01 Data LOW 8bit	0x04	--	Palette0 Color01 Data HIGH 7bit
0x05	Palette0 Color10 Data LOW 8bit	0x06	--	Palette0 Color10 Data HIGH 7bit
0x07	Palette0 Color11 Data LOW 8bit	0x08	--	Palette0 Color11 Data HIGH 7bit

	d7 d0		d7	d0
0x09	Palette1 Color01 Data LOW 8bit	0x0A	--	Palette1 Color01 Data HIGH 7bit
0x0B	Palette1 Color10 Data LOW 8bit	0x0C	--	Palette1 Color10 Data HIGH 7bit
0x0D	Palette1 Color11 Data LOW 8bit	0x0E	--	Palette1 Color11 Data HIGH 7bit

0x0F	0	0	0	0	0	0	0	0

Command: PAL23 (Code: 0x01)

Function: Sets the color data for SGB color palettes 2 and 3.



	d7 d0		d7	d0
0x01	Palette2 Color00 Data LOW 8bit	0x02	--	Palette2 Color00 Data HIGH 7bit
0x03	Palette2 Color01 Data LOW 8bit	0x04	--	Palette2 Color01 Data HIGH 7bit
0x05	Palette2 Color10 Data LOW 8bit	0x06	--	Palette2 Color10 Data HIGH 7bit
0x07	Palette2 Color11 Data LOW 8bit	0x08	--	Palette2 Color11 Data HIGH 7bit

	d7 d0		d7	d0
0x09	Palette3 Color01 Data LOW 8bit	0x0A	--	Palette3 Color01 Data HIGH 7bit
0x0B	Palette3 Color10 Data LOW 8bit	0x0C	--	Palette3 Color10 Data HIGH 7bit
0x0D	Palette3 Color11 Data LOW 8bit	0x0E	--	Palette3 Color11 Data HIGH 7bit

0x0F	0	0	0	0	0	0	0	0

Command: PAL03 (Code: 0x02)

Function: Sets the color data for SGB color palettes 0 and 3.

	d7							d0
0x00	0	0	0	0	1	0	0	1

Number of packets: 0x1 (fixed)

Command code: 0x02

	d7 … d0
0x01	Palette0 Color00 Data LOW 8bit
0x03	Palette0 Color01 Data LOW 8bit
0x05	Palette0 Color10 Data LOW 8bit
0x07	Palette0 Color11 Data LOW 8bit

	d7	… d0
0x02	--	Palette0 Color00 Data HIGH 7bit
0x04	--	Palette0 Color01 Data HIGH 7bit
0x06	--	Palette0 Color10 Data HIGH 7bit
0x08	--	Palette0 Color11 Data HIGH 7bit

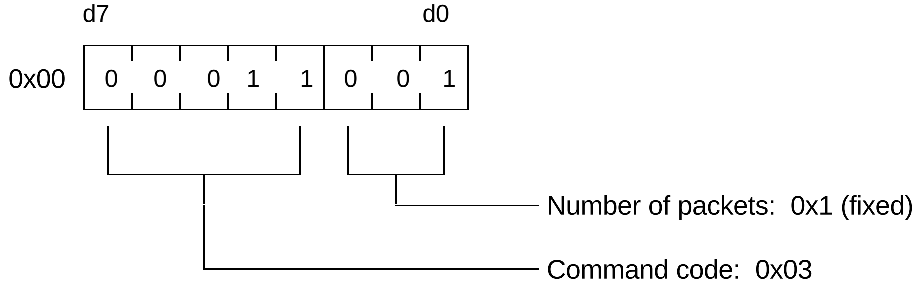
	d7 … d0
0x09	Palette3 Color01 Data LOW 8bit
0x0B	Palette3 Color10 Data LOW 8bit
0x0D	Palette3 Color11 Data LOW 8bit

	d7	… d0
0x0A	--	Palette3 Color01 Data HIGH 7bit
0x0C	--	Palette3 Color10 Data HIGH 7bit
0x0E	–	Palette3 Color11 Data HIGH 7bit

0x0F	0	0	0	0	0	0	0	0

Command: PAL12 Code: 0x03

Function: Sets the color data for SGB color palettes 1 and 2.



	d7							d0
0x00	0	0	0	1	1	0	0	1

Number of packets: 0x1 (fixed)

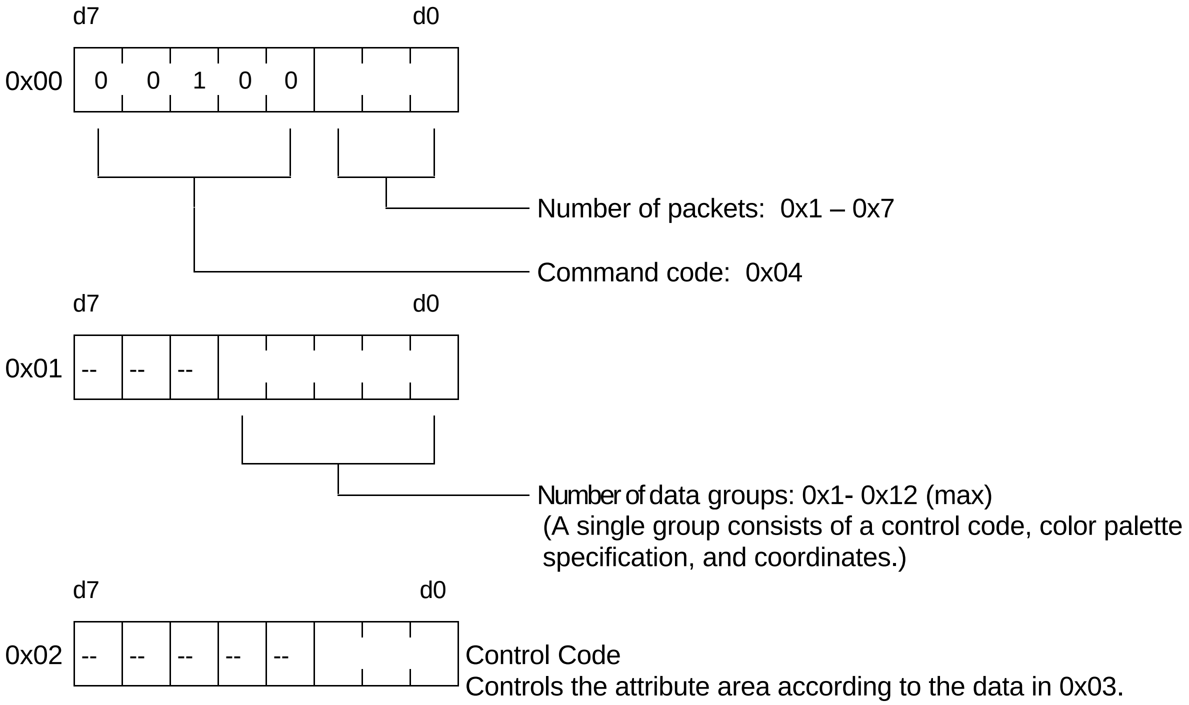
Command code: 0x03

	d7	d0		d7	d0
0x01	Palette1 Color00 Data LOW 8bit		0x02	--	Palette1 Color00 Data HIGH 7bit
0x03	Palette1 Color01 Data LOW 8bit		0x04	--	Palette1 Color01 Data HIGH 7bit
0x05	Palette1 Color10 Data LOW 8bit		0x06	--	Palette1 Color10 Data HIGH 7bit
0x07	Palette1 Color11 Data LOW 8bit		0x08	--	Palette1 Color11 Data HIGH 7bit
	d7	d0		d7	d0
0x09	Palette2 Color01 Data LOW 8bit		0x0A	--	Palette2 Color01 Data HIGH 7bit
0x0B	Palette2 Color10 Data LOW 8bit		0x0C	--	Palette2 Color10 Data HIGH 7bit
0x0D	Palette2 Color11 Data LOW 8bit		0x0E	--	Palette2 Color11 Data HIGH 7bit

0x0F	0	0	0	0	0	0	0	0

Command Code: ATTR_BLK (Code: 0x04)

Function: Applies the specified color palette attributes to areas inside and outside the square.



Number of packets: 0x1 – 0x7

Command code: 0x04

Number of data groups: 0x1- 0x12 (max)
(A single group consists of a control code, color palette specification, and coordinates.)

Control Code
Controls the attribute area according to the data in 0x03.

Control Codes

000	No control occurs.
001	Applies the attributes specified by d1 and d0 of 0x03 only to the area within the square (including the CHR border).
010	Applies the color palette attributes specified by d3 and d2 of 0x03 only on the square CHR border.
011	Applies the color palette attributes specified by d1 and d0 of 0x03 only to the area within the square, and applies the color palette attributes specified by d3 and d2 of 0x03 only to the border of the square.
100	Applies the attributes specified by d5 and d4 of 0x03 only to the area outside the square (including the CHR border).
101	Applies the color palette attributes specified by d1 and d0 of 0x03 to the area within the square, and applies the color palette attributes specified by d5 and d4 of 0x03 to the area outside of the CHR border. (CHR border is unchanged.)
110	Applies the color palette attributes specified by d5 and d4 of 0x03 only to the area outside of the square, and applies the color palette attributes specified by d3 and d2 of 0x03 to the CHR border .
111	Applies the specified color palette attributes to the area inside the square, to the CHR border line, and to the area outside the CHR border .

The color palette attributes of areas not specified are not changed.

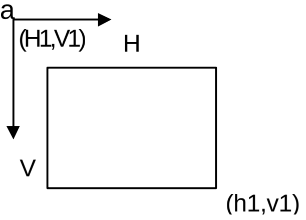
	d7							d0	
0x03	--	--							Specifies the color palette

- Bits d1–d0: Color palette number for the area inside the square.
- Bits d3–d2: Color palette number for character area on the square
- Bits d5–d4: Color palette number for area outside the square.

	d7							d0
0x04	--	--	--	Starting point H1				
0x05	--	--	--	Starting point V1				
0x06	--	--	--	Ending point h1				
0x07	--	--	--	Ending point v1				

0x04–0x07: Starting (upper left) and ending (lower right) points of the square.

Coordinate data



	d7							d0	
0x08	--	--	--	--	--				Control Code
0x09	--	--							Specifies the color palette
0x0A	--	--	--	Starting point H2					Coordinate data
0x0B	--	--	--	Starting point V2					
0x0C	--	--	--	Ending point h2					
0x0D	--	--	--		Ending point v2				
0x0E	0	0	0	0	0	0	0	0	
0x0F	0	0	0	0	0	0	0	0	

Note ***If the number of packets is 1, 0x00 is written to 0x0E and 0x0F. If the number of packets exceeds 1, the control code and color palette specification code of the next data item are written to 0x0E and 0x0F, respectively.***

When the number of packets exceeds 1:

	d7							d0	
0x0E	--	--	--	--	--				Control Code
0x0F	–	--							Color palette specification ↑Remainder of previous/first packet ↓Second packet
0x00	--	--	--	Starting point H3					Coordinate data
0x01	--	--	--	Starting point V3					
0x02	--	--	--	Ending point h3					
0x03	--	--	--	Ending point v3					

	d7							d0	
0x04	--	--	--	--	--				Control code
0x05	--	--							Color palette specification
0x06	--	--	--	Starting point H4					Coordinate data
0x07	--	--	--	Starting point V4					
0x08	--	--	--	Ending point h4					
0x09	--	--	--	Ending point v4					

	d7							d0	
0x0A	--	--	--	--	--				Control code
0x0B	--	--							Color palette specification
0x0C	--	--	--	Starting point H5					Coordinate data
0x0D	--	--	--	Starting point V5					
0x0E	--	--	--	Ending point h5					
0x0F	--	--	--	Ending point v5					

0x00	--	--	--	--	--				Control code
0x01	--	--							Color palette specification
0x02				* *					

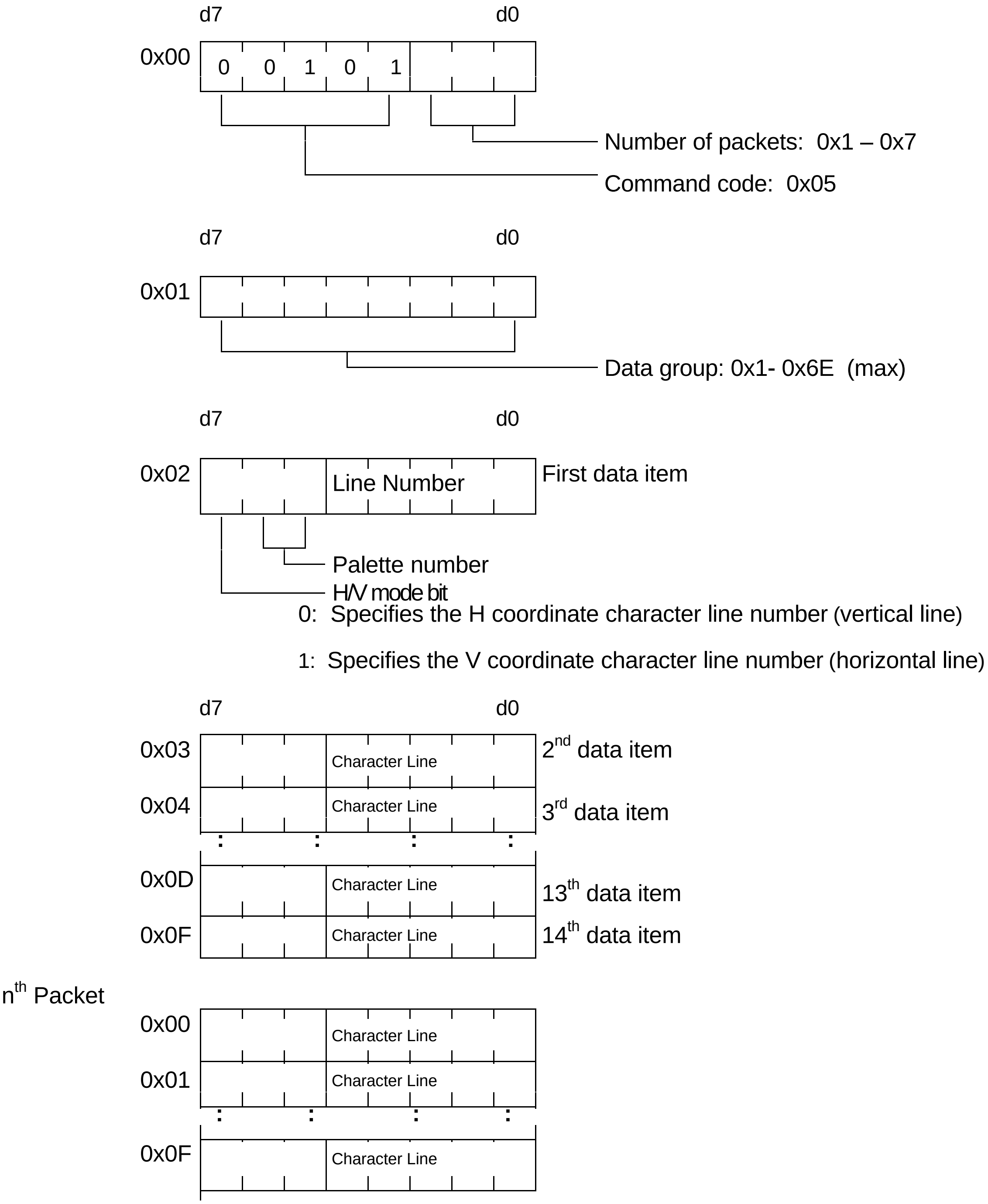
* The empty area of the packet is filled with 0x00.

Note ***When there is no area inside the square border (e.g., h1 = H1 + 1), a control code such as one that sets the color attribute for the area inside the border cannot be used.***

Please note that when ATTR_BLK, ATTR_LIN, ATTR_DIV, or ATTR_CHR are used, the data that is sent are valid even if MASK_EN (freezes screen immediately before masking) is selected.
When using MASK_EN before these commands, use 0x10 or 0x11 as the argument. If 0x01 is used as the MASK_EN argument, ATTR_TRN and ATTR_SET should be used.

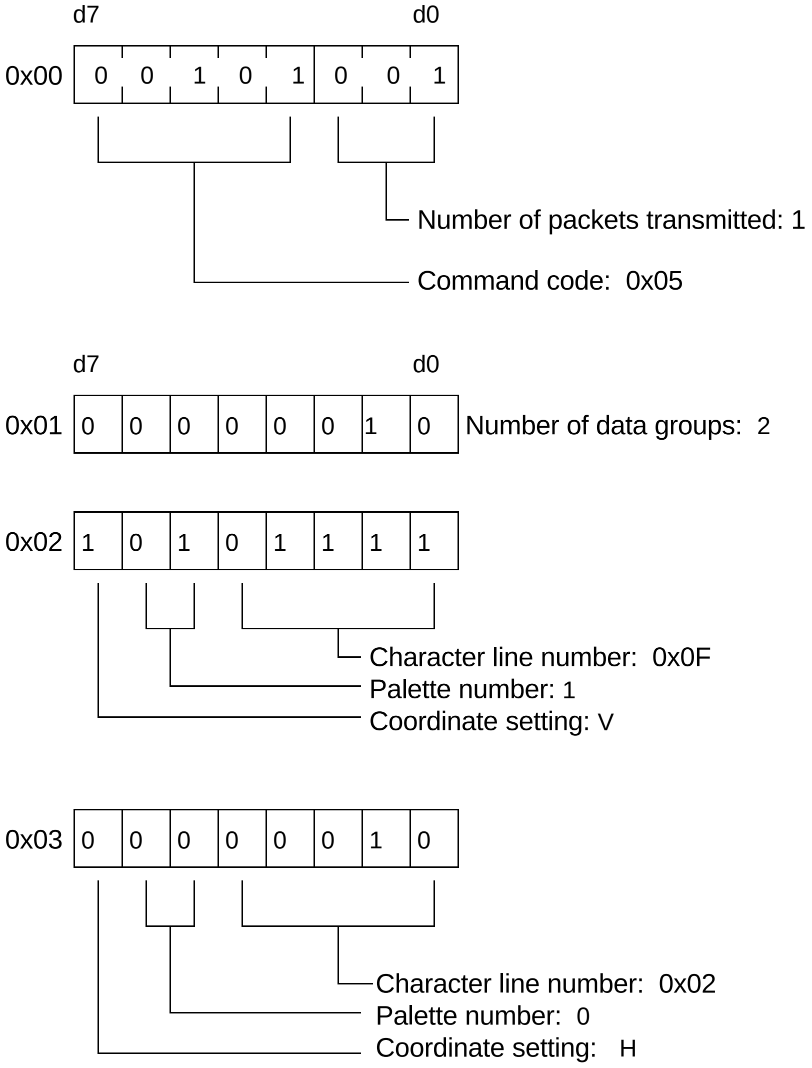
Command: ATTR_LIN (Code: 0x05)

Function: Applies the specified color palette attribute to a coordinate line.

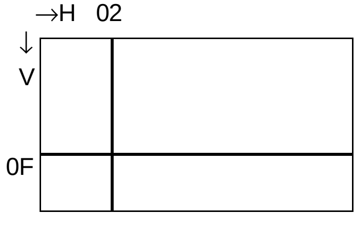


* See the note on ATTR_BLK.

Example



Applies the Palette 0 attribute to this line.

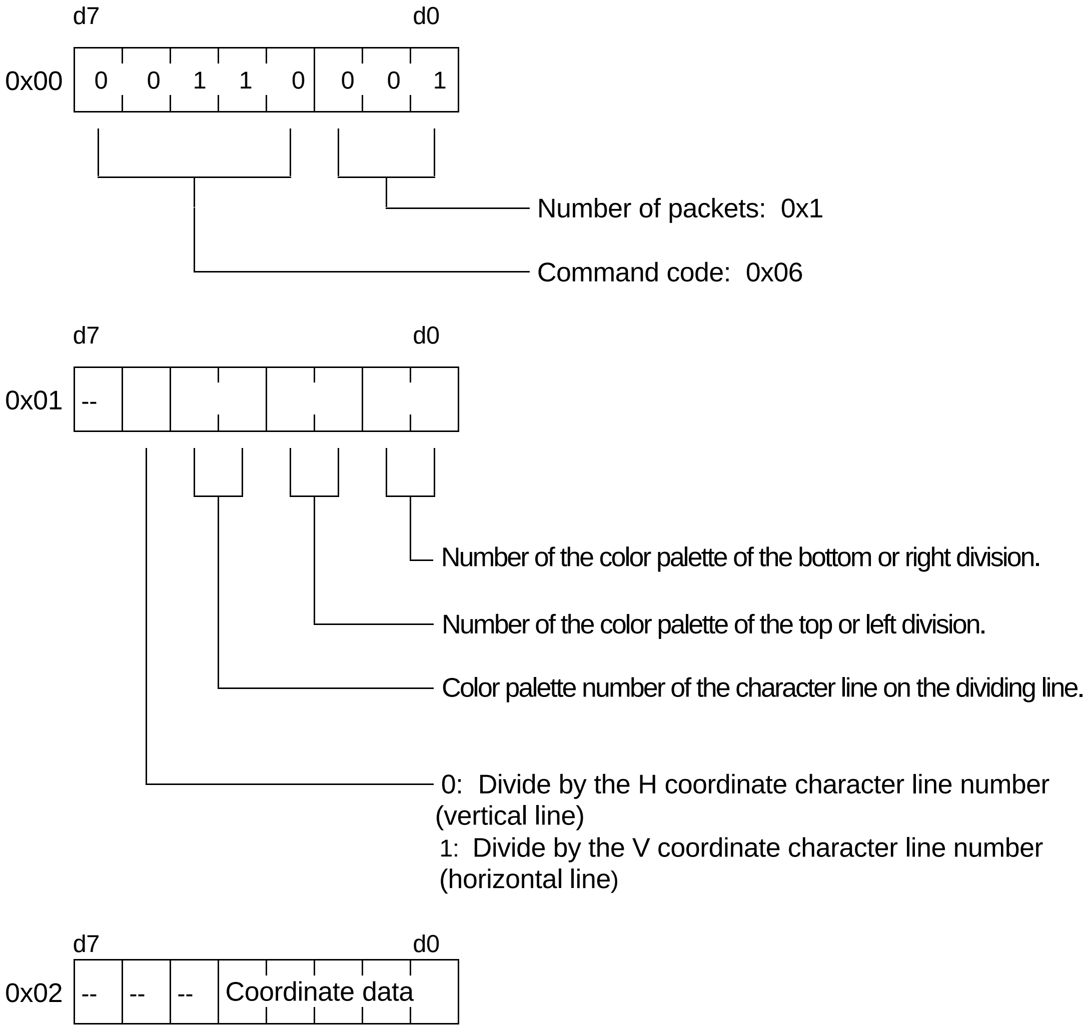


Applies the Palette 1 attribute to this line.

* The color of intersection of the two lines is decided by the last line color.

Command: ATTR_DIV (Code: 0x06)

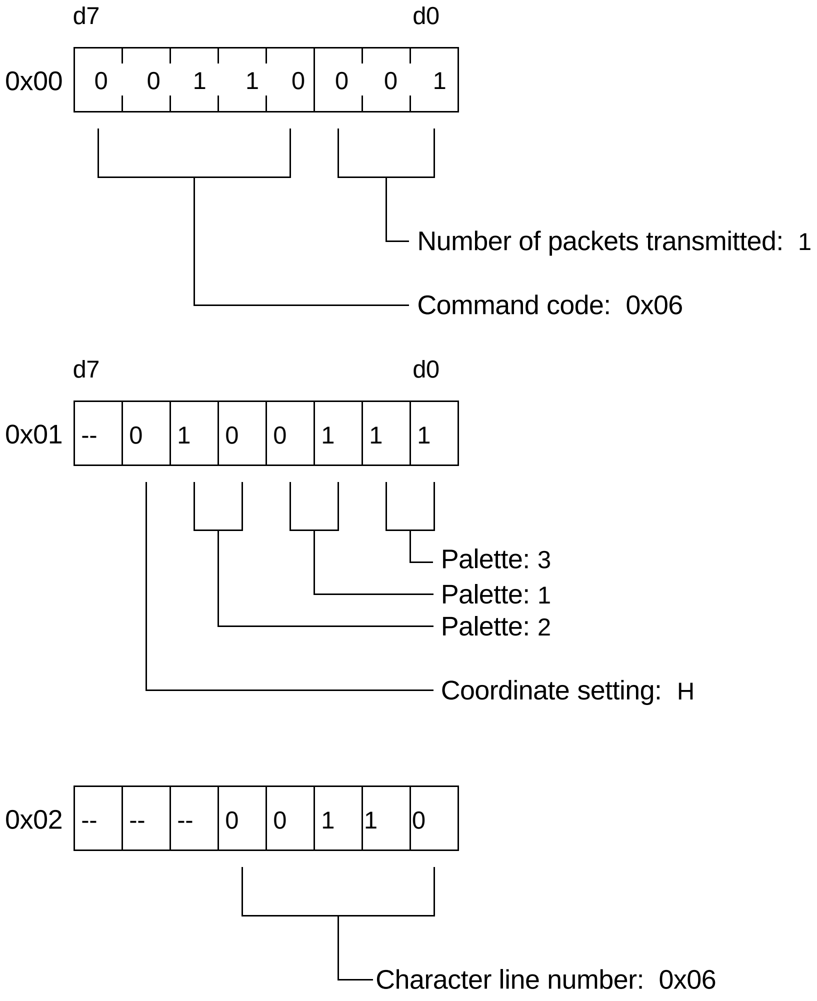
Function: Divides the color palette attributes of the screen by the specified coordinates.


d7
d0
0x00
0 0 1 1 0 0 0 1
Number of packets: 0x1
Command code: 0x06
d7
d0
0x01
--
Number of the color palette of the bottom or right division.
Number of the color palette of the top or left division.
Color palette number of the character line on the dividing line.
0: Divide by the H coordinate character line number (vertical line)
1: Divide by the V coordinate character line number (horizontal line)
d7
d0
0x02
-- -- --
Coordinate data

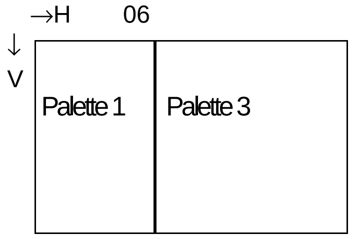
* 0x03 - 0x0F should be filled with 0x00.

* See note on ATTR_BLK.

Example



Sets this character line to the Palette 2 attribute.



Command: ATTRIBUTE_CHR (Code: 0x07)

Function: Specifies a color palette for each character.

	d7							d0
0x00	0	0	1	1	1			

Number of packets: 0x1 – 0x6

Command code: 0x07

	d7							d0
0x01	--	--	--					

H coordinate of start of write

	d7							d0
0x02	--	--	--					

V coordinate of start of write

	d7							d0
0x03								

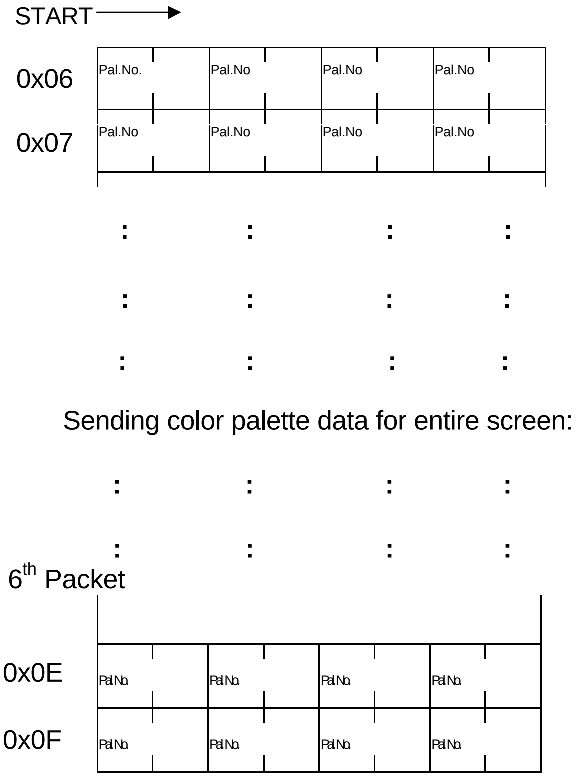
Number of data items to send
Each data item (2 bits) specifies a color palette.

	d7							d0
0x04	--	--	--	--	--	--	--	

Most significant bit of number of data items sent, specified in 0x03
(The maximum number of data items required is 360.)

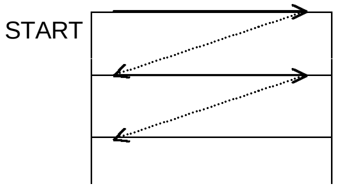
	d7							d0
0x05	--	--	--	--	--	--	--	

Write horizontally: 0; Write vertically

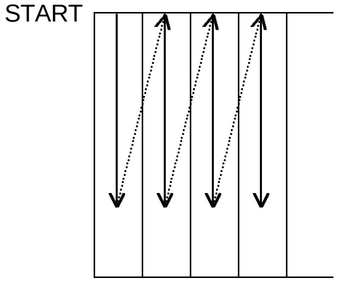


Sending color palette data for entire screen:

Horizontal write (H direction)



Vertical write (V direction)

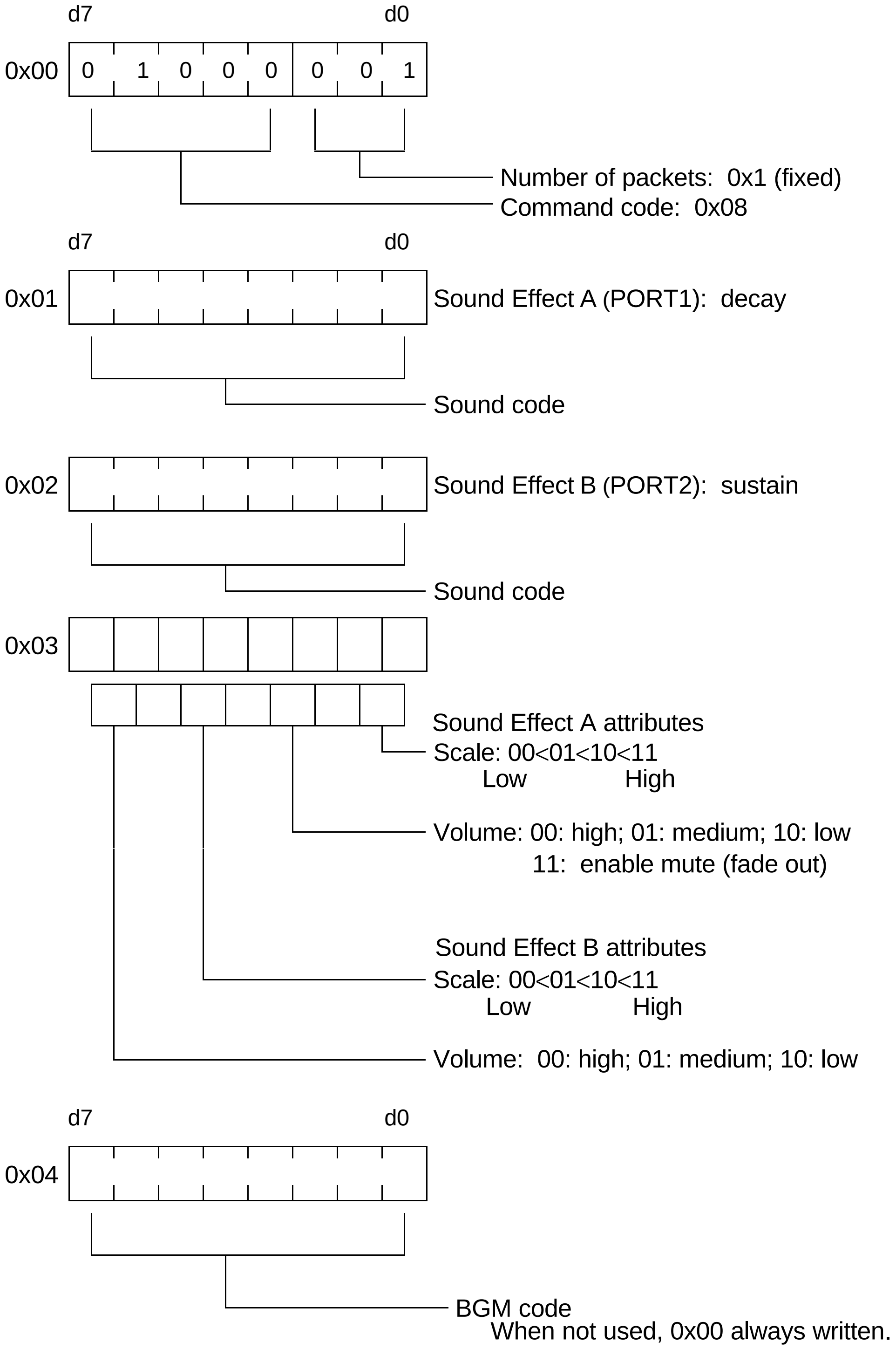


← Data items nos. 357, 358, 359, and 360.

* See note on ATTR_BLK.

Command: SOUND (Code: 0x08)

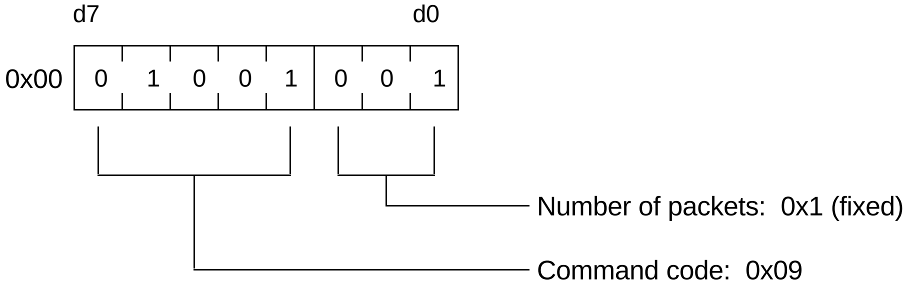
Function: Generates and halts internal sound effects and sounds that use internal tone data.


d7
d0
0x00
0 1 0 0 0 0 0 1
Number of packets: 0x1 (fixed)
Command code: 0x08
d7
d0
0x01
Sound Effect A (PORT1): decay
Sound code
0x02
Sound Effect B (PORT2): sustain
Sound code
0x03
Sound Effect A attributes
Scale: 00<01<10<11
Low High
Volume: 00: high; 01: medium; 10: low
11: enable mute (fade out)
Sound Effect B attributes
Scale: 00<01<10<11
Low High
Volume: 00: high; 01: medium; 10: low
d7
d0
0x04
BGM code
When not used, 0x00 always written.

* For more information, see Section 3.4, *Sound Flag Lists*.

Command: SOU_TRN (Code: 0x09)
(Data transfer using VRAM)

Function: Sends a sound program and sound data to the APU.



* The 4 Kbytes of SGB RAM data immediately following the command is sent to APU RAM.

The data to be transferred must be prepared prior to the frame preceding issuance of the command.

The transfer ends 6 frames after the command is issued (not counting the frame in which the command is issued).

The beginning of the data for transfer contains a 16-bit representation of the number of data items and the transfer destination address, and the end contains an ending code and the starting address of the program. For more information, see Chapter 7: *Super Game Boy Sound*.

APU RAM program area: 0x0400 – 0x2AFF/9.75 Kbytes
APU RAM music data area: 0x2B00 – 0x4AFF/8 Kbytes
APU RAM sampling data area: 0x4DB0-0xEEFF/40.25 Kbytes

Note ***When SOU_TRN is used, 5 packets of SOU_TRN initialization should be sent to the register file. For more information, see Section 5.3,*** **SOU_TRN Initialization Data*.***

Command: PAL_SET (Code: 0x0A)

Function: Applies system color palettes to SGB color palettes.

	d7							d0
0x00	0	1	0	1	0	0	0	1

Number of packets: 0x1 (fixed)

Command code: 0x0A

	d7							d0
0x01								

Number of the system color palette to apply to SGB color (LOW)

	d7							d0
0x02	--	--	--	--	--	--	--	

Number of the system color palette to apply to SGB color (HIGH)

* The system color palettes selected are palettes 000-511.

Number of the system color palette applied to SGB color palette 1.

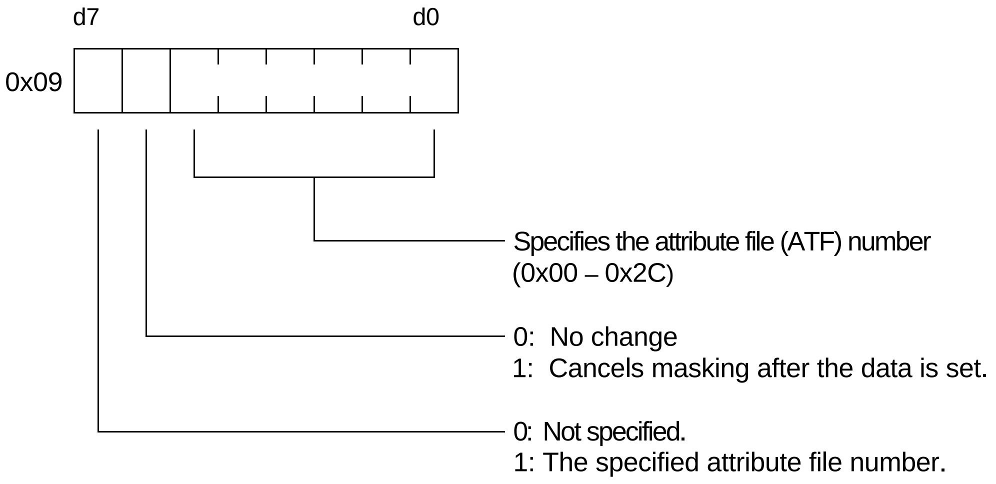
	d7							d0	
0x03									LOW
0x04	--	--	--	--	--	--	--		HIGH

Number of the system color palette applied to SGB color palette 2.

	d7							d0	
0x05									LOW
0x06	--	--	--	--	--	--	--		HIGH

Number of the system color palette applied to SGB color palette 3.

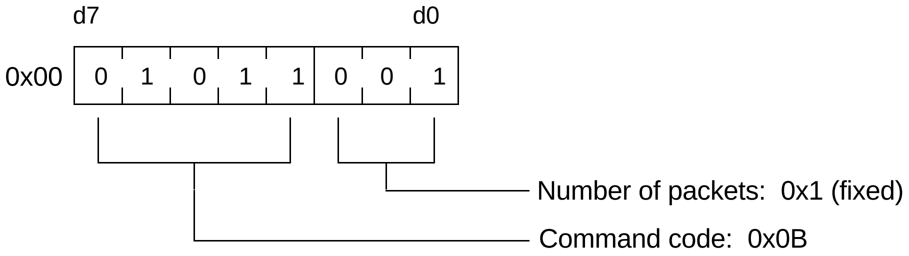
	d7							d0	
0x07									LOW
0x08	--	--	--	--	--	--	--		HIGH


d7
d0
0x09
Specifies the attribute file (ATF) number
(0x00 – 0x2C)
0: No change
1: Cancels masking after the data is set.
0: Not specified.
1: The specified attribute file number.

* 0x0A - 0x0F should be filled with 0x00.

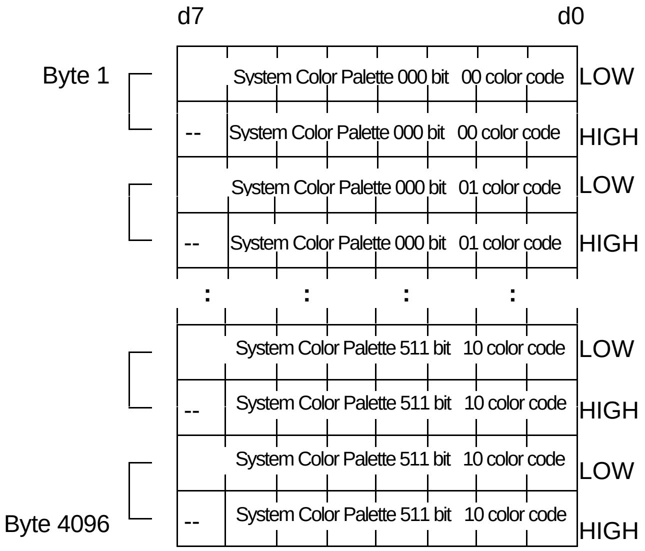
Command: PAL_TRN (Code: 0x0B)
(Data Transfer using VRAM)

Function: Transfers color data to the system color palette.



* The 4 Kbytes of SGB RAM data immediate following the command is handled as system color palette data and stored in SUPER NES WRAM as data for system color palettes 000 – 511.

The format of data storage in SGB RAM is as follows.

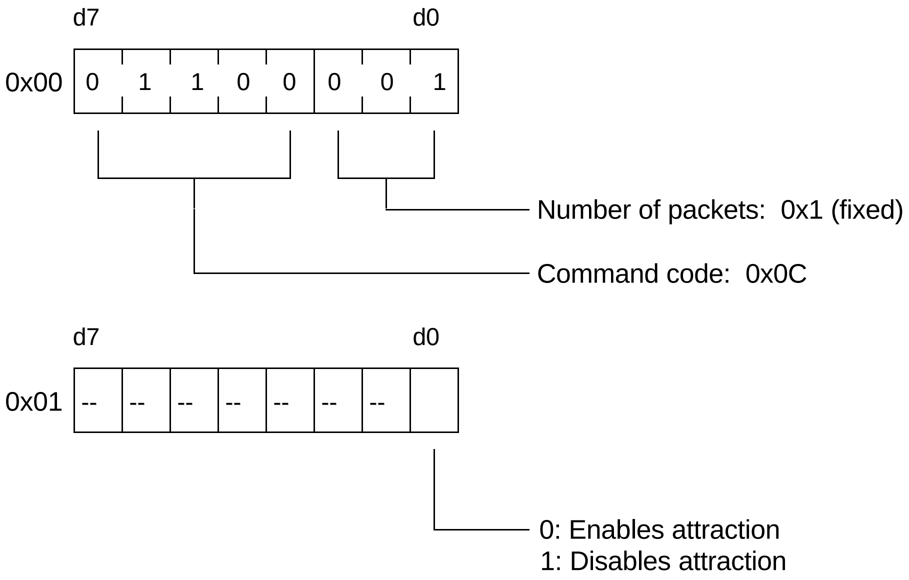

d7
d0
Byte 1
System Color Palette 000 bit 00 color code
LOW
--
System Color Palette 000 bit 00 color code
HIGH
System Color Palette 000 bit 01 color code
LOW
--
System Color Palette 000 bit 01 color code
HIGH
: : : :
System Color Palette 511 bit 10 color code
LOW
--
System Color Palette 511 bit 10 color code
HIGH
System Color Palette 511 bit 10 color code
LOW
Byte 4096
--
System Color Palette 511 bit 10 color code
HIGH

The storage addresses are 0x3000 - 0x3FFF.

Command: ATRC_EN (Code: 0x0C)

Function: Enables and disables attraction mode.

Enables and disables attraction on the picture frame.
The default setting is enabled (0x00).
If the command is issued during attraction, attraction is stopped.



	d7							d0
0x00	0	1	1	0	0	0	0	1
0x01	--	--	--	--	--	--	--	

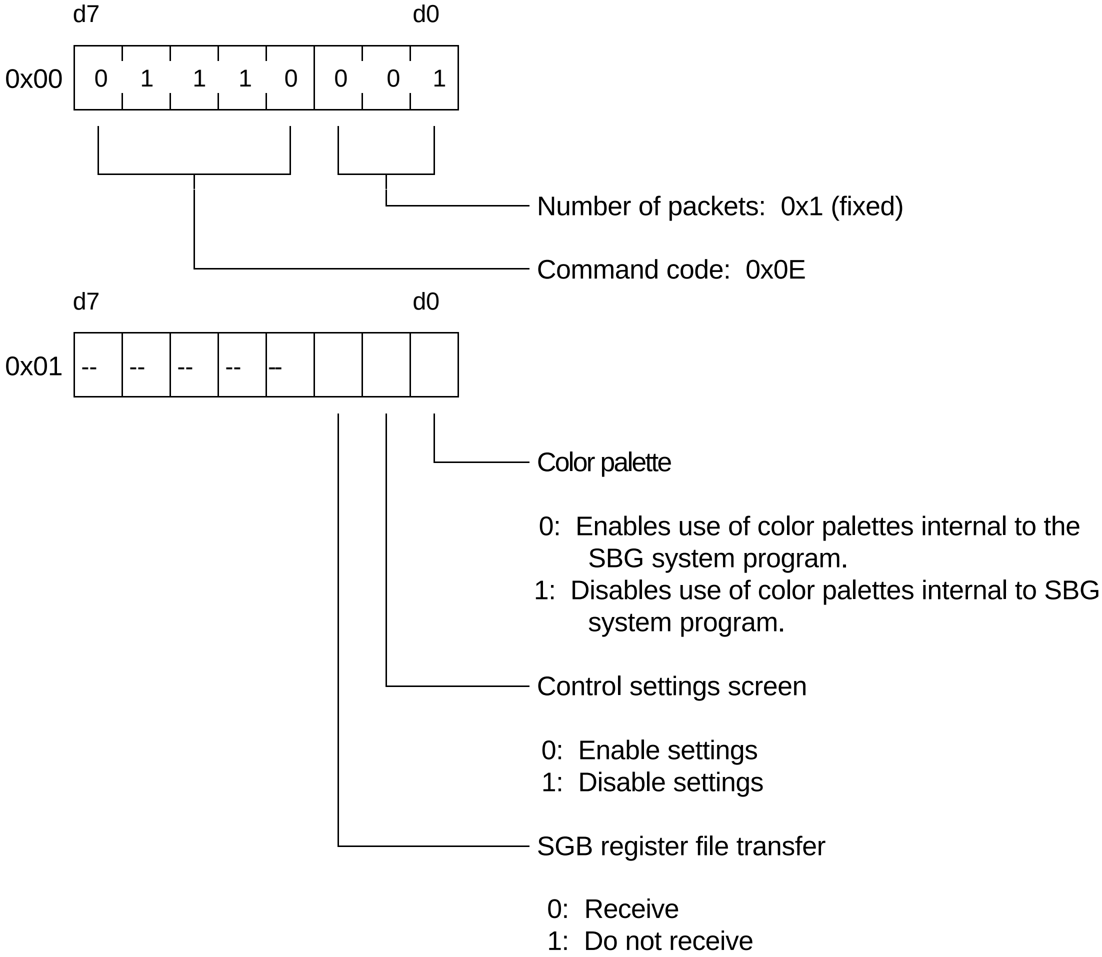
Example: Attraction start duration for a model (type without communication connector).

The time required for attraction to start for each picture frame is as follows.
(Times shown in parentheses are times required to start attraction a second time.)

Mario	7 min. (5 min.)
Cork	3 min. (5 min.)
Landscape	1 min. (1 min.)
Cinema	3 mins.
Cats	3 mins.
Pencils	3 mins. (5 mins.)
Escher art	7 mins. (5 mins.)

Command: ICON_EN (Code: 0x0E)

Function: Enables and disables the icon function.



	d7							d0
0x00	0	1	1	1	0	0	0	1

Number of packets: 0x1 (fixed)

Command code: 0x0E

	d7							d0
0x01	--	--	--	--	--			

Color palette

0: Enables use of color palettes internal to the SBG system program.
1: Disables use of color palettes internal to SBG system program.

Control settings screen

0: Enable settings
1: Disable settings

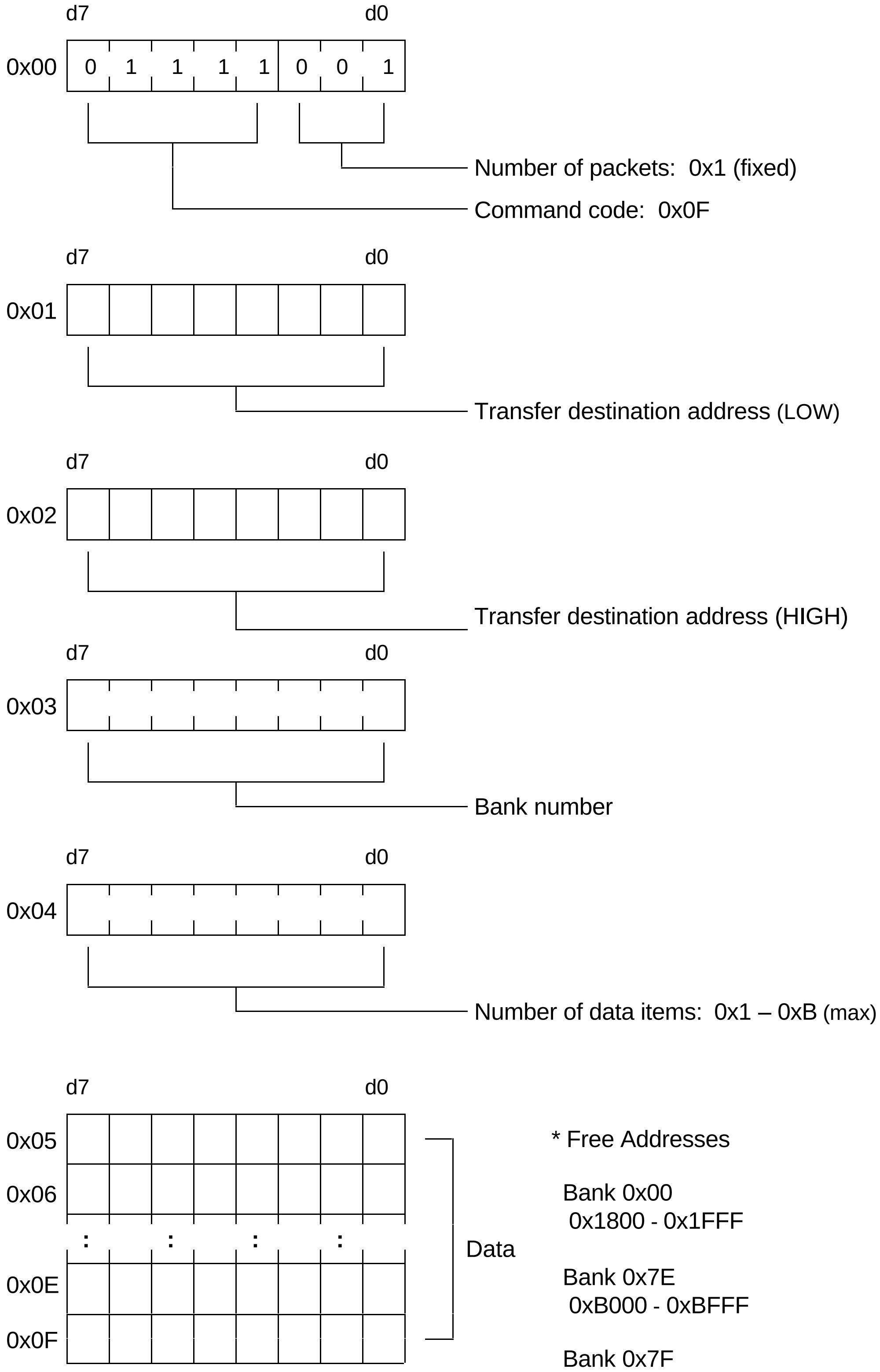
SGB register file transfer

0: Receive
1: Do not receive

* The default value is 0x00.

Command: DATA_SND (Code: 0x0F)

Function: Transfers data to SUPER NES WRAM using the register file.


d7
d0
0x00
0 1 1 1 1 0 0 1
Number of packets: 0x1 (fixed)
Command code: 0x0F
0x01
Transfer destination address (LOW)
0x02
Transfer destination address (HIGH)
0x03
Bank number
0x04
Number of data items: 0x1 – 0xB (max)
0x05
0x06
0x0E
0x0F
Data

* Free Addresses

Bank 0x00
0x1800 - 0x1FFF

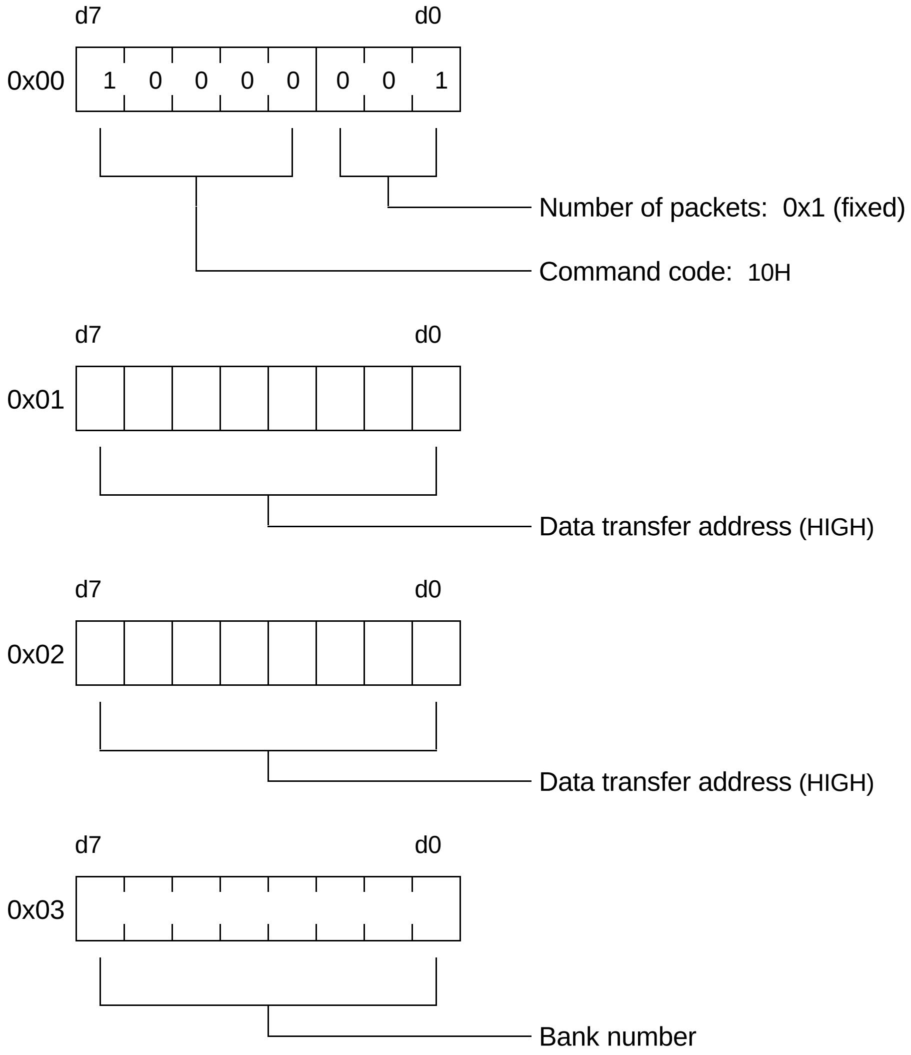
Bank 0x7E
0xB000 - 0xBFFF

Bank 0x7F
0x0000 - 0xFFFF

Command: DATA_TRN (Code: 0x10)

(Data Transfer using VRAM)

Function: Transfers data in SGB RAM to SUPER NES WRAM.


d7
d0
0x00
1 0 0 0 0 0 0 1
Number of packets: 0x1 (fixed)
Command code: 10H
d7
d0
0x01
Data transfer address (HIGH)
d7
d0
0x02
Data transfer address (HIGH)
d7
d0
0x03
Bank number

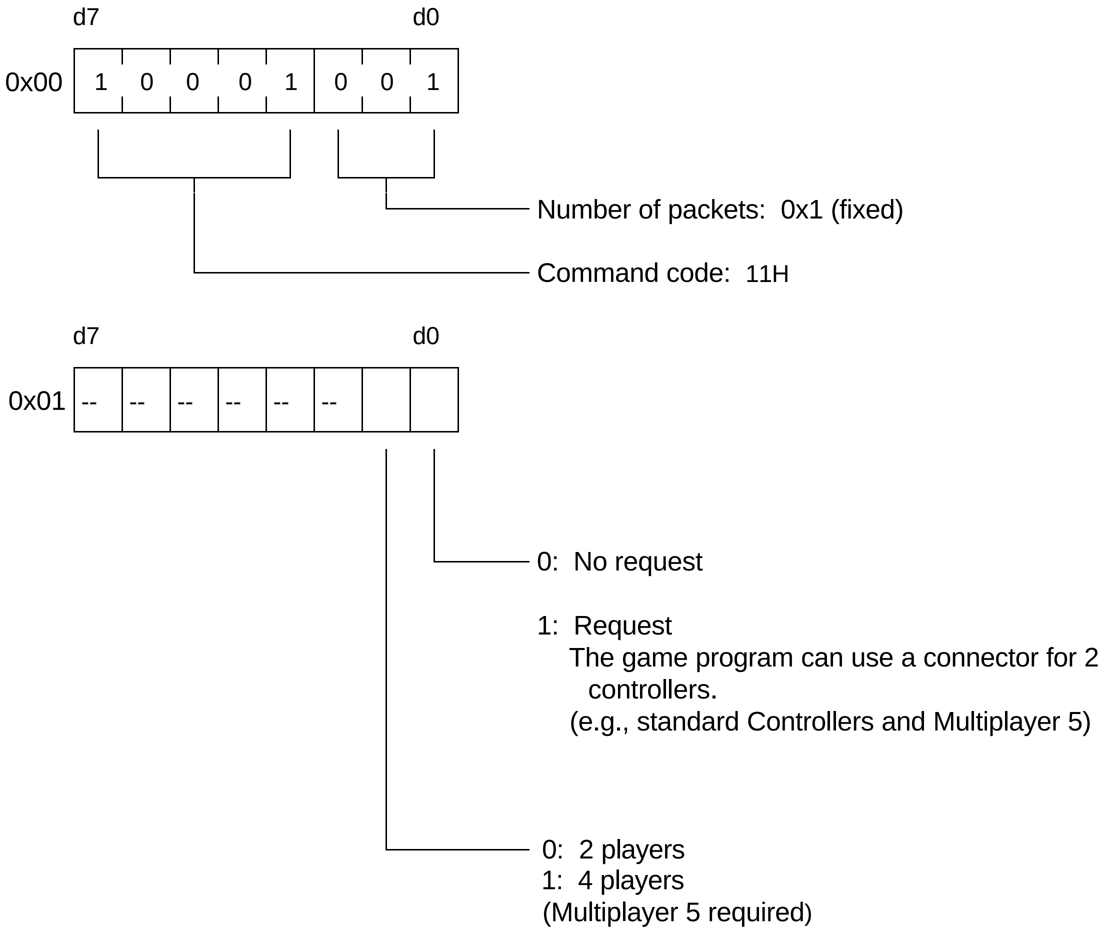
* Free Addresses

Bank 0x00	0x1800 – 0x1FFF
Bank 0x7E	0xB000 – 0xBFFF
Bank 0x7F	0x0000 – 0xFFFF

Note ***When an SHVC program is tranferred to WRAM and executed, 0x00 should be written to 0x1700 of bank 00. This can be written either by using DATA_SND or DATA_TRN or by using the transferred program.***

Command: MLT_REQ (Code: 0x11)

Function: Requests multiplayer mode.



Number of packets: 0x1 (fixed)

Command code: 11H

0: No request

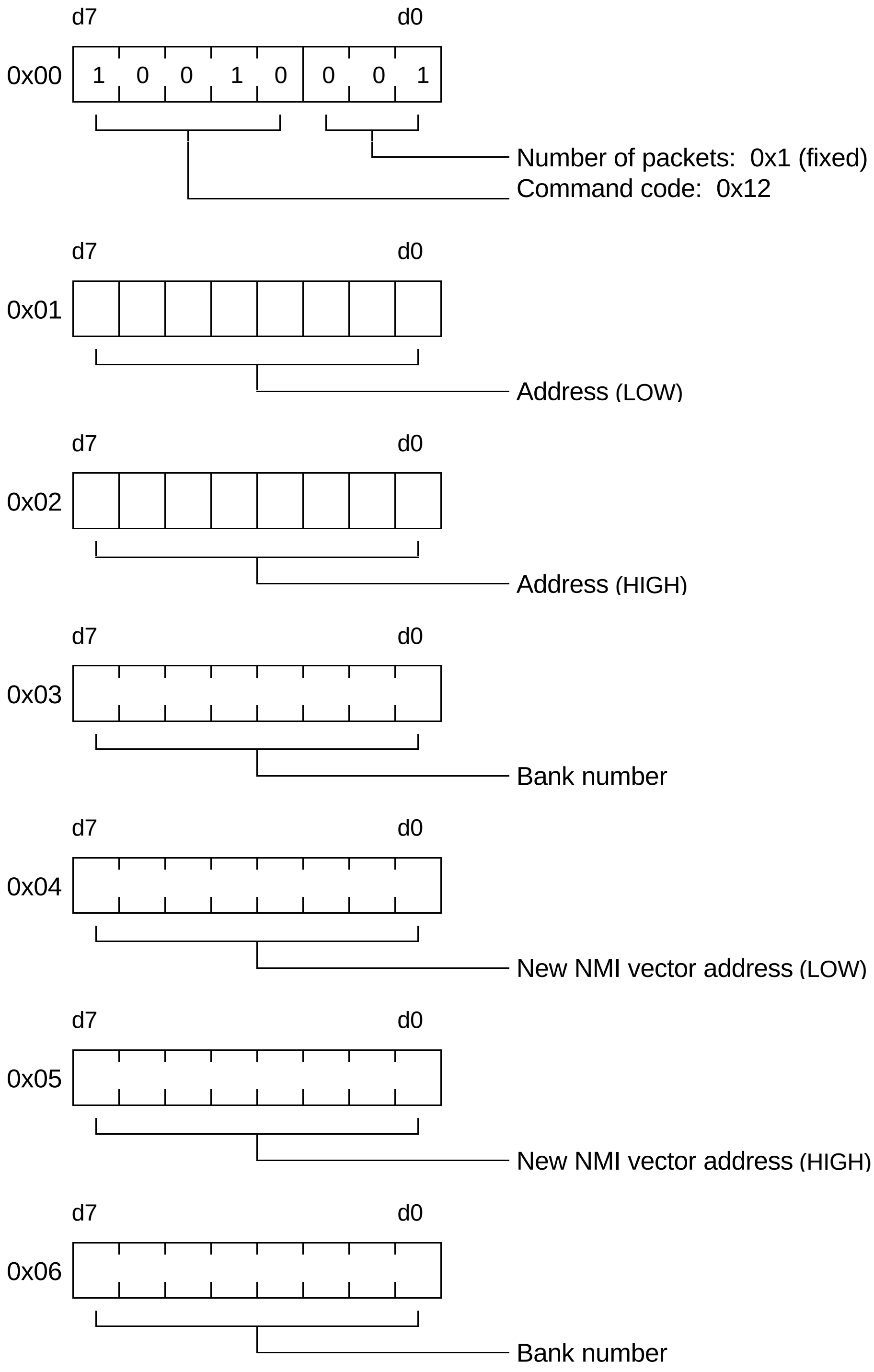
1: Request
The game program can use a connector for 2 controllers.
(e.g., standard Controllers and Multiplayer 5)

0: 2 players
1: 4 players
(Multiplayer 5 required)

* The default value is 0x00.

Command: JUMP (Code: 0x12)

Function: Sets the SUPER NES program counter to the specified address.



Note	***If all addresses from 0x04 to 0x06 are set to 0, the NMI jumps to the original vector. NMI is disabled in the system program, so it must be enabled to be used.***

Command: CHR_TRN (Code: 0x13)
(Data Transfer using VRAM)

Function: Transfers SUPER NES character format data.

	d7							d0
0x00	1	0	0	1	1	0	0	1

Number of packets: 0x1 (fixed)
Command code: 0x13

0x01	--	--	--	--	--	--	0	

0: Data for characters 0x00 – 0x7F (BG)

1: Data for characters 0x80 – 0xFF (BG)

The characters are in 16-color (4-bit) mode.

Note ***The 4 Kbytes of SGB RAM data immediately following this command is handled as SUPER NES character data and transferred to SUPER NES VRAM.***

The format of the tranferred data is based on the SUPER NES character data format.

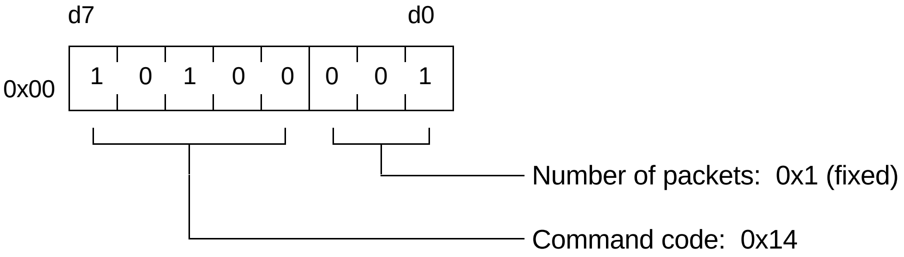
The BG character names are allocated to 0x00 – 0xFF.

When character data is used for the picture frame, characters with a character name setting of 0x00 should consist of null bits, and all dots of characters with a name setting of 0x01 should be represented by non-null bits.

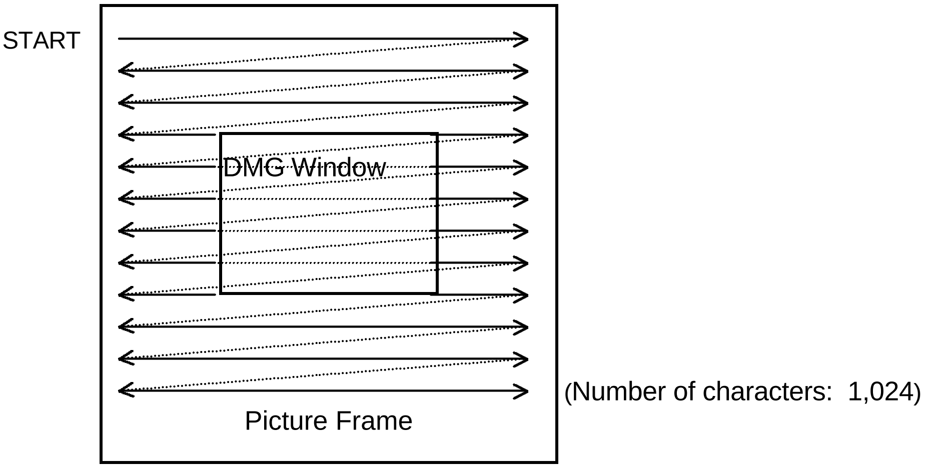
Command: PCT_TRN (Code: 0x14)

(Data Transfer using VRAM)

Function: Transfers screen data and color data for picture frames created by the software developer.



* The 4 Kbytes of SGB RAM immediately following this command are handled as screen data and transferred to SUPER NES VRAM.

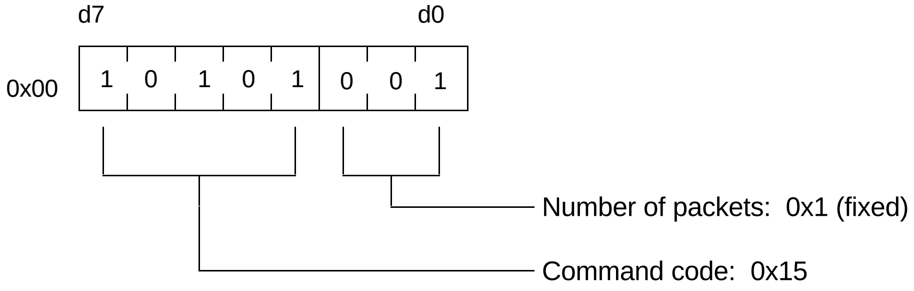


There should be 1,024 uncompressed characters of screen data.
The inside of the DMG window should be filled with null characters.
Three color palettes, 4-6, are transferred.
The initial data consists of 2,048 bytes of screen data. This is followed by by 3 palettes of color data (2 bytes x 16 x 3).

The format of the transferred data is based on that of SUPER NES screen and color data. However, the BG priority bit is set to 0, the color palettes to palette numbers 4-6, the higher-order 2 bits of the character name to 00b, and the characters to 8 x 8-bit mode.

Command: ATTR_TRN (Code: 0x15)
(Data Transfer using VRAM)

Function: Transfers attribute files.



* The 4 Kbytes of SGB RAM immediately following this command are transferred to WRAM as attribute files. (The capacity of each attribute file is 2 x 20 x 18/8 = 90 bytes. Thus, 45 attribute files occupy 4,050 bytes, 0xATF0-0xATF44.

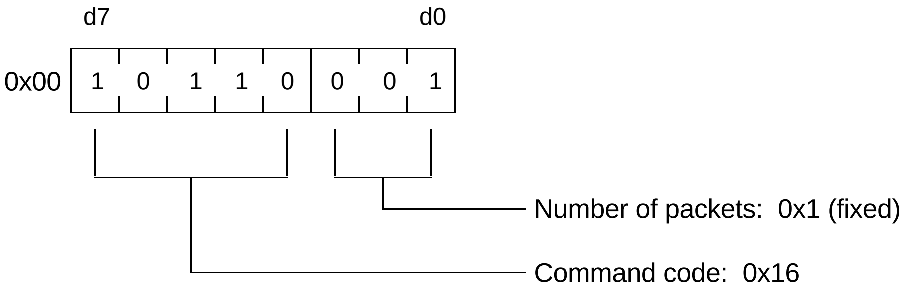
The ATF data format (90 bytes total) -- written horizontally from the left edge of the DMG window.

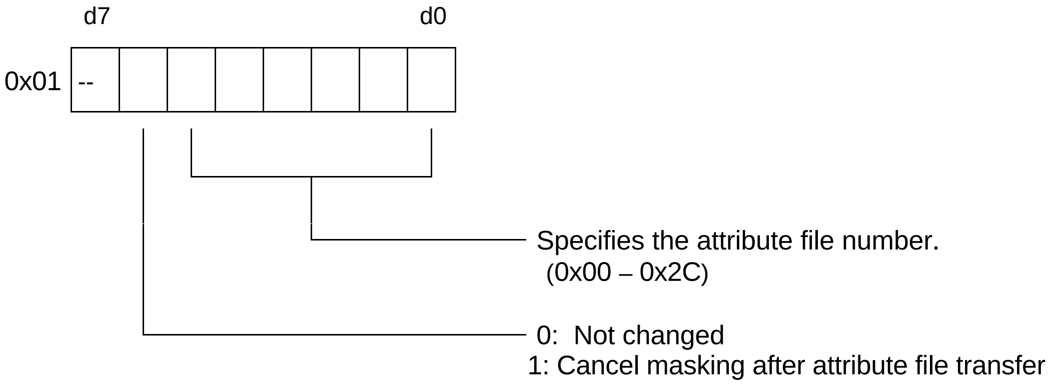
d7,d6	Byte 1 d5,d4	d3,d2	d1,d0	d7,d6	Byte 2 d5,d4	d3,d2	d1,d0	. .
d7,d6	Byte 6 d5,d4	d3,d2	d1,d0	d7,d6	Byte 7 d5,d4	d3,d2	d1,d0	. .
:	:	:	:	:	:	:	:	
d7,d6	Byte 81 d5,d4	d3,d2	d1,d0	d7,d6	Byte 82 d5,d4	d3,d2	d1,d0	. .
d7,d6	Byte 86 d5,d4	d3,d2	d1,d0	d7,d6	Byte 87 d5,d4	d3,d2	d1,d0	. .

(The figure depicts a DMG window with 20 x 18 characters.)

Command: ATTR_SET (Code: 0x16)

Function: Applies the specified attribute file to the DMG window.

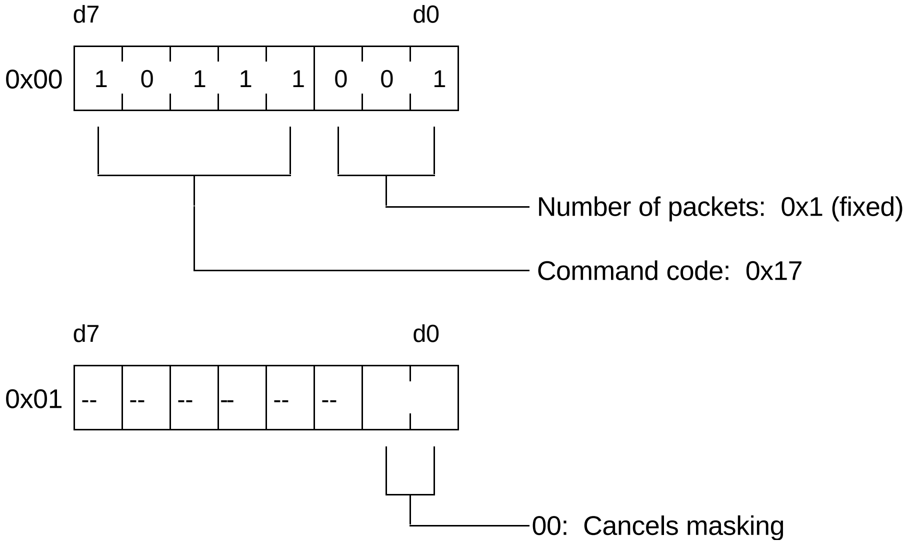



d7
d0
0x01
--
Specifies the attribute file number.
(0x00 – 0x2C)
0: Not changed
1: Cancel masking after attribute file transfer

* 0x02 - 0x0F filled with 0x00.

Command: MASK_EN (Code: 0x17)

Function: Masks the DMG window.



01: Freezes the screen immediately before masking. (No transfers to SUPER NES VRAM are performed from after the command is issued until masking is canceled.)

10: Masks by setting all SGB color palette color codes to black.

11: Masks by setting all SGB color palette color codes to the same color (color of bit 00).

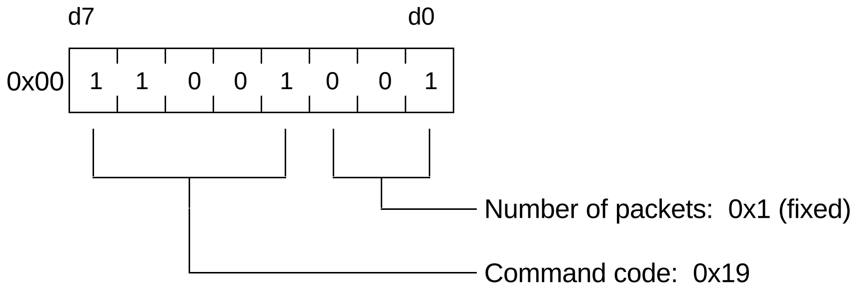
* 0x2- 0xF filled with 0x00.

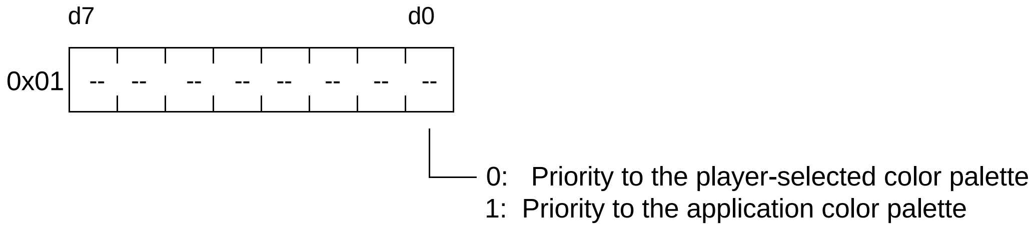
Note	***When masking is performed at the start of the game, it should be performed after the DMG reset is canceled and around the time that the DMG screen is displayed on SGB. (The timing of the command should be adjusted so that it is issued after a wait of several frames.) Without this timing, the screen may be momentarily be displayed in monochrome.***

Note	***Masking with an argument (0X01) of 0x10 or 0x11 is prohibited during a game.***

Command: PAL_PRI (Code: 0x19)

Function: Specifies the priority of the color palette for the application and the color palette selected by the player.




d7
d0
0x01
-- -- -- -- -- -- -- --
0: Priority to the player-selected color palette
1: Priority to the application color palette

* Default is 0.

Priority to Player-Selected Color Palette
When a screen that uses a player-selected color palette is displayed, any color or attribute settings commands that were sent have no effect on the DMG window.

Priority to Application Color Palette
When a screen that uses a player-selected color palette is displayed and a color or attribute setting command was sent, the sent colors are displayed in the DMG window.

* The corresponding commands are as follows.

00, 01, 02, 03, 04, 05, 06, 07, 0A, 16 (Code value)

3.3 Cautions Regarding Sending Commands

- After each packet (128 bits) is sent, 0 must always be sent in bit 129.
- If a data sequence covers more than 1 packet, byte 1 of each packet after the first is a continuation of the data of the previous packet.
- 0x00 is written to the unused areas in each packet.
- Note that there are two modes of data transfer: register-file mode and a mode in which 4 Kbytes are tranferred using SGB RAM.
- Controller key input should not be read while a command is being sent.

3.4 Sound Flag Summary

- Pre-loaded sound effects A and B can be played simultaneously using system commands.
- The A sound effects are formants, primarily action sounds, and the B sound effects are looping sounds, primarily ambient sounds.
- The interval (frequency) for these sound effects can be set to 4 levels.
- Changing the interval A allows a completely different sound effect to be obtained with the same sound source. In addition, the volume can be set to 3 levels.

3.4.1 Sound Effect A Flags

SOUND Command			
0x01	d7-d0 [bit]	0x03 d1-d0 [bit]	
Code	Flag Meaning	Recommended Interval Value	Voices Used
0x00	Dummy flag for retriggering		• • 6 • 7
0x80	Sound effect A stop (mute)		• • 6 • 7
0x01	Nintendo	d1 = 1 • d0 = 1	• • • 7
0x02	Game over sound	d1 = 1 • d0 = 1	• • 6 • 7
0x03	Falling sound	d1 = 1 • d0 = 1	• • • 7
0x04	Predetermined sound • • • A	d1 = 1 • d0 = 1	• • 6 • 7
0x05	Predetermined sound • • • B	d1 = 1 • d0 = 1	• • 6 • 7
0x06	Selected sound • • • A	d1 = 1 • d0 = 1	• • 6 • 7
0x07	Selected sound • • • B	d1 = 1 • d0 = 1	• • • 7
0x08	Selected sound • • • C	d1 = 1 • d0 = 0	• • 6 • 7
0x09	Error sound • • • buzzer	d1 = 1 • d0 = 0	• • • 7
0x0A	Item-catch sound	d1 = 1 • d0 = 0	• • 6 • 7
0x0B	One knock on door	d1 = 1 • d0 = 0	• • 6 • 7
0x0C	Explosion • • • small	d1 = 1 • d0 = 1	• • 6 • 7
0x0D	Explosion • • • medium	d1 = 1 • d0 = 1	• • 6 • 7
0x0E	Explosion • • • large	d1 = 1 • d0 = 1	• • 6 • 7
0x0F	Defeat sound • • • A	d1 = 1 • d0 = 1	• • • 7
0x10	Defeat sound • • • B	d1 = 1 • d0 = 1	• • 6 • 7
0x11	Striking sound (attack) • • • A	d1 = 1 • d0 = 0	• • 6 • 7
0x12	Striking sound (attack) • • • B	d1 = 1 • d0 = 0	• • 6 • 7
0x13	Air-sucking sound	d1 = 1 • d0 = 0	• • 6 • 7
0x14	Rocket launcher • • • A	d1 = 1 • d0 = 1	• • 6 • 7
0x15	Rocket launcher • • • B	d1 = 1 • d0 = 1	• • 6 • 7
0x16	Bubbling sound (in water)	d1 = 1 • d0 = 0	• • • 7

SOUND Command			
0x01	d7-d0 [bit]	0x03 d1-d0 [bit]	
Code	Flag Meaning	Recommended Interval Value	Voices Used
0x17	Jump	d1 = 1 • d0 = 1	• • • 7
0x18	Fast jump	d1 = 1 • d0 = 1	• • • 7
0x19	Jet (rocket) firing	d1 = 1 • d0 = 0	• • • 7
0x1A	Jet (rocket) landing	d1 = 1 • d0 = 0	• • • 7
0x1B	Cup breaking	d1 = 1 • d0 = 0	• • 6 • 7
0x1C	Glass breaking	d1 = 1 • d0 = 1	• • 6 • 7
0x1D	Level up	d1 = 1 • d0 = 0	• • 6 • 7
0x1E	Air injection	d1 = 1 • d0 = 1	• • • 7
0x1F	Sword wielding	d1 = 1 • d0 = 1	• • • 7
0x20	Falling in water	d1 = 1 • d0 = 0	• • • 7
0x21	Fire	d1 = 1 • d0 = 1	• • • 7
0x22	Breaking wall	d1 = 1 • d0 = 1	• • 6 • 7
0x23	Cancellation sound	d1 = 1 • d0 = 0	• • 6 • 7
0x24	Stepping	d1 = 1 • d0 = 1	• • 6 • 7
0x25	Block-hitting sound	d1 = 1 • d0 = 1	• • 6 • 7
0x26	Sound of picture floating into view	d1 = 1 • d0 = 1	• • 6 • 7
0x27	Screen fade-in	d1 = 1 • d0 = 0	• • 6 • 7
0x28	Screen fade-out	d1 = 1 • d0 = 0	• • 6 • 7
0x29	Window opening	d1 = 1 • d0 = 1	• • 6 • 7
0x2A	Window closing	d1 = 1 • d0 = 0	• • 6 • 7
0x2B	Large laser sound	d1 = 1 • d0 = 1	• • 6 • 7
0x2C	Sound of stone door closing (opening)	d1 = 1 • d0 = 0	• • 6 • 7
0x2D	Teleportation	d1 = 1 • d0 = 1	• • • 7
0x2E	Thunder	d1 = 1 • d0 = 0	• • 6 • 7
0x2F	Earthquake	d1 = 1 • d0 = 0	• • 6 • 7
0x30	Small laser sound	d1 = 1 • d0 = 0	• • 6 • 7

3.4.2 Sound Effect B Flags

SOUND Command			
0x02 d7-d0 [bit]		0x03 d5-d4 [bit]	
Code	Flag Meaning	Recommended Interval Value	Voices Used
0x00	Dummy flag for retriggering		0 • 1 • 4 • 5
0x80	Sound Effects B stop (mute)		0 • 1 • 4 • 5
0x01	Applause • • • small crowd	d5 = 1 • d4 = 0	• • • 5
0x02	Applause • • • medium crowd	d5 = 1 • d4 = 0	• • 4 • 5
0x03	Applause • • • large crowd	d5 = 1 • d4 = 0	0 • 1 • 4 • 5
0x04	Wind	d5 = 0 • d4 = 1	• • 4 • 5
0x05	Rain	d5 = 0 • d4 = 1	• • • 5
0x06	Storm	d5 = 0 • d4 = 1	• 1 • 4 • 5
0x07	Hurricane	d5 = 1 • d4 = 0	0 • 1 • 4 • 5
0x08	Thunder	d5 = 0 • d4 = 0	• • 4 • 5
0x09	Earthquake	d5 = 0 • d4 = 0	• • 4 • 5
0x0A	Lava flow	d5 = 0 • d4 = 0	• • 4 • 5
0x0B	Wave	d5 = 0 • d4 = 0	• • • 5
0x0C	River	d5 = 1 • d4 = 1	• • 4 • 5
0x0D	Waterfall	d5 = 1 • d4 = 0	• • 4 • 5
0x0E	Small character running	d5 = 1 • d4 = 1	• • • 5
0x0F	Horse galloping	d5 = 1 • d4 = 1	• • • 5
0x10	Warning sound	d5 = 0 • d4 = 1	• • • 5
0x11	Futuristic car running	d5 = 0 • d4 = 0	• • • 5
0x12	Jet flying	d5 = 0 • d4 = 1	• • • 5
0x13	UFO flying	d5 = 1 • d4 = 0	• • • 5
0x14	Electromagnetic waves	d5 = 0 • d4 = 0	• • • 5
0x15	Sound of score being raised	d5 = 1 • d4 = 1	• • • 5

SOUND Command			
0x02	d7-d0 [bit]	0x03 d5-d4 [bit]	
Code	Flag Meaning	Recommended Interval Value	Voices Used
0x16	Fire	d5 = 1 • d4 = 0	• • • 5
0x17	Camera shutter (formant)	d5 = 0 • d4 = 0	0 • 1 • 4 • 5
0x18	Writing (formant)	d5 = 0 • d4 = 0	• • • 5
0x19	Erasing (formant)	d5 = 0 • d4 = 0	• • • 5
0x81	Use prohibited (used by system)		
0x82	Use prohibited (used by system)		

3.4.2 Attributes of A and B Sound Effects

	SOUND Command								
	0x03								
	d7	d6	d5	d4	d3	d2	d1	d0	
A Sound Effects					×	×	0	0	Interval (short)
					×	×	0	1	Interval (med-short)
					×	×	1	0	Interval (med-long)
					×	×	1	1	Interval (long)
					0	0	×	×	Volume (high)
					0	1	×	×	Volume (med)
					1	0	×	×	Volume (low)
B Sound Effects	×	×	0	0					Interval (short)
	×	×	0	1					Interval (med-short)
	×	×	1	0					Interval (med-long)
	×	×	1	1					Interval (long)
	0	0	×	×					Volume (high)
	0	1	×	×					Volume (med)
	1	0	×	×					Volume (low)
					1	1			Mute ON

- Mute takes effect only when both bits d2 and d3 are set to 1. If the volume is set for either the A or B sound effect, mute is turned off.

- Fade-out and fade-in take effect with mute-on and mute-off, respectively. Mute-on and mute-off are implemented for BGM played by A and B sound effects and by the APU.

- There is no independent mute-off flag.

- When the mute flag is set, the volume and interval data for the A (Port 1) and B (Port 2) sound effects also should be set.

4. MISCELLANEOUS

4.1 Reading Input from Multiple Controllers

After a multiplayer request (Command MLT_REQ) is sent, data from controllers 1, 2, 3, and 4 automatically become readable.

In 2-player mode, data from controller 1 is read first, followed by data from controller 2, then data from controller 1 again, and so on. In 4-player mode, the order is controller 1, controller 2, controller 3, controller 4, controller 1 again, and so on.

In these cases, the next controller for which data is to be read must be determined beforehand by reading P10-P13 with P14 and P15 high.

P10 - P13	Next Controller to Read
0xF	Controller 1
0xE	Controller 2
0xD	Controller 3
0xC	Controller 4

Note ***Controller data cannot be read if Multiplayer 5 and SUPER NES Mouse are connected at the same time.***

4.2 RECOGNIZING SGB

4.2.1 Distinguishing between Game Boy types (DMG, MGB/MGL, SGB, and SGB2)

The program uses the following methods to determine which of the 4 types is operating.

- Checks the initial value of the internal accumulator of the CPU. (distinguishes between previous/new versions of CPU).

01	→ DMG or SGB
FF	→ MGB/MGL or SGB2

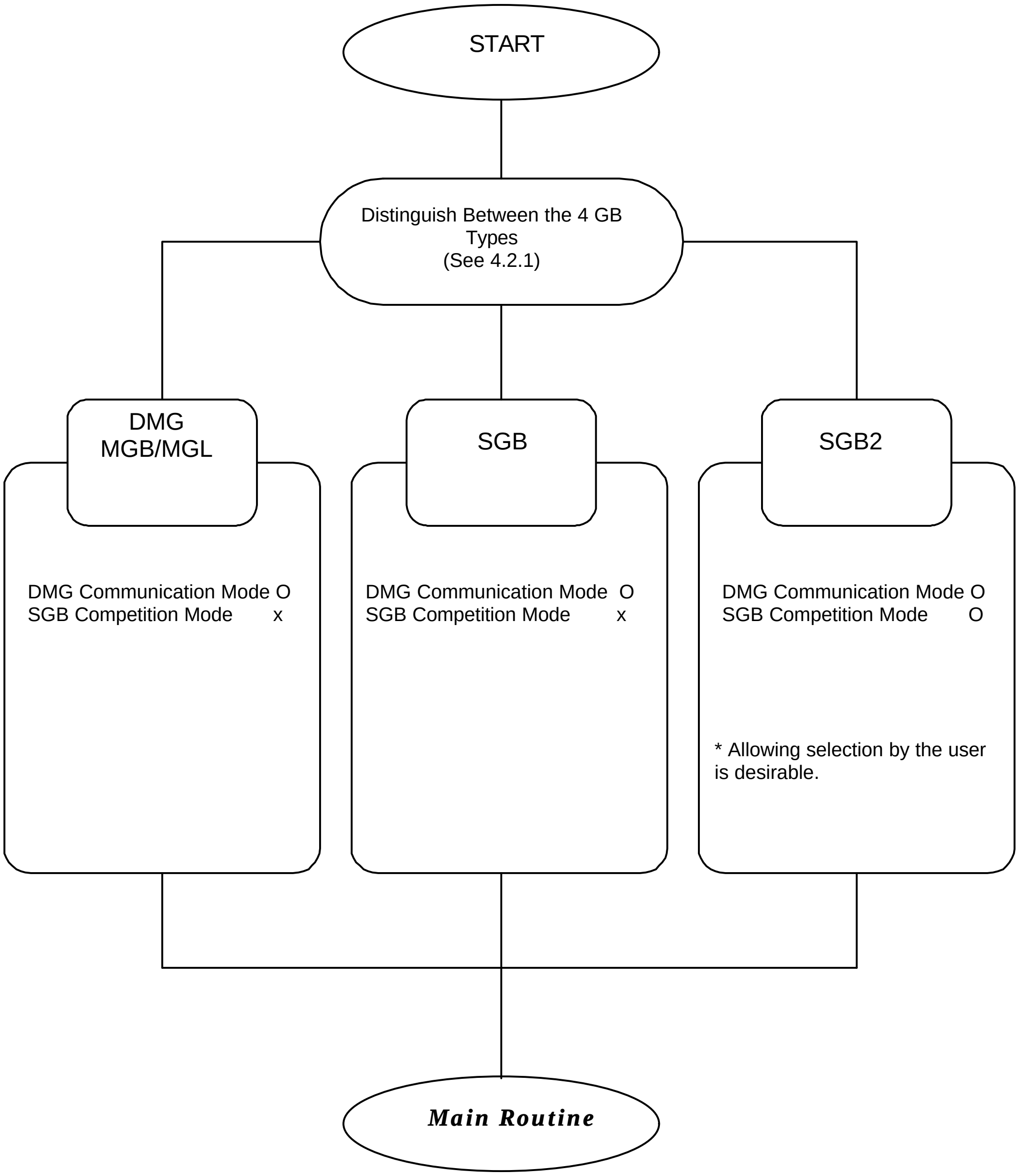
- Sends a muliplayer request (Command MLT_REQ) and determines whether there is a switch to multiplayer mode.

No switch	→ DMG or MGB/MGL
Switch	→ SGB or SGB2

* The following table summarizes these methods.

Initial Value of CPU Internal Accumulator	Switch/No Switch to Multiplayer Mode	Game Boy Type
01	No switch	DMG
	Switch	SGB
FF	No switch	MGB/MGL
	Switch	SGB2

4.2.2 Usage Example: Distinguishing Between the 4 Game Boy Types



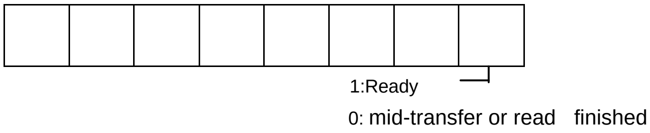
* A sample program for distinguishing between GB types is provided.

4.3 SGB Register Summary

The following registers can be used to perform functions such resetting the SGB CPU from a program transferred to SUPER NES WRAM and receiving and passing data to a DMG program.

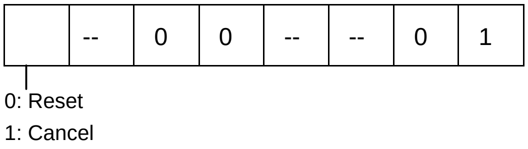
Register File Status
[RFS] 0x6002 (RD)

Reads the status of the register file



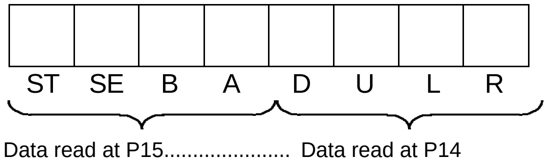
DMG Reset Register
[DRR] 0x6003 (WR)

Resets the SGB CPU



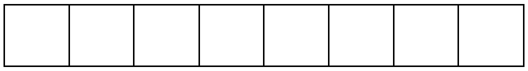
Controller Register1
[CR1] 0x6004 (WR)

Writes data from controller 1



Register File
[RF$_0$ - RF$_F$] 0x7000 - 0x700F (RD)

Register file for communication



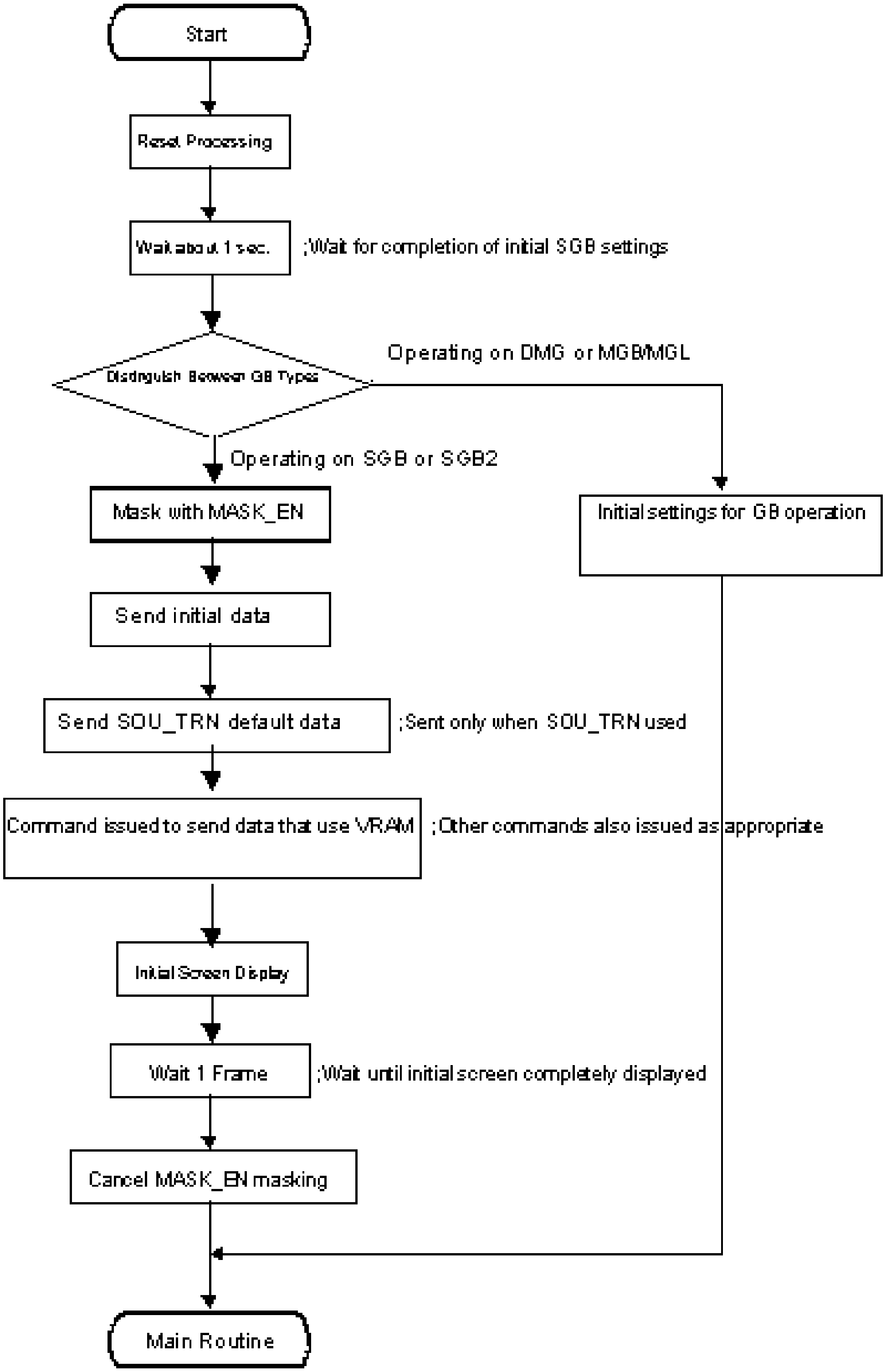
* The RFS flag (0x6002) is cleared to 0 by the reading of 0x7000.

The SGB CPU can be reset using DRR.

Using RF$_0$ - RF$_F$ and RFS allows data sent to the register file by the DMG program to be received by the SUPER NES program.

CR1 is a register used by the original SGB system program for writing keypad data from controller 1. The SUPER NES program can use the controller-reading routine of the DMG program to receive data written to this register.

4.4 Flowchart of Initial Settings Routine


Start
Reset Processing
Wait about 1 sec.
;Wait for completion of initial SGB settings
Distinguish Between GB Types
Operating on DMG or MGB/MGL
Operating on SGB or SGB2
Mask with MASK_EN
Initial settings for GB operation
Send initial data
Send SOU_TRN default data
;Sent only when SOU_TRN used
Command issued to send data that use VRAM
;Other commands also issued as appropriate
Initial Screen Display
Wait 1 Frame
;Wait until initial screen completely displayed
Cancel MASK_EN masking
Main Routine

5. PROGRAMMING CAUTIONS

5.1 ROM Registration Data

To use SGB functions (system commands), the following values must be stored at the ROM addresses indicated.

x146 ← 0x03 and 0x14B ← 0x33

5.2 Initial Data

When writing programs that use the system commands of SGB and SGB2, use the initialization routine of the game program to send the following 8 packets of default data to the register file.

```
INIT1   DEFB   $79,$5D,$08,$00,$0B,$8C,$D0,$F4,$60,$00,$00,$00,$00,$00,$00,$00

INIT2   DEFB   $79,$52,$08,$00,$0B,$A9,$E7,$9F,$01,$C0,$7E,$E8,$E8,$E8,$E8,$E0

INIT3   DEFB   $79,$47,$08,$00,$0B,$C4,$D0,$16,$A5,$CB,$C9,$05,$D0,$10,$A2,$28

INIT4   DEFB   $79,$3C,$08,$00,$0B,$F0,$12,$A5,$C9,$C9,$C8,$D0,$1C,$A5,$CA,$C9

INIT5   DEFB   $79,$31,$08,$00,$0B,$0C,$A5,$CA,$C9,$7E,$D0,$06,$A5,$CB,$C9,$7E

INIT6   DEFB   $79,$26,$08,$00,$0B,$39,$CD,$48,$0C,$D0,$34,$A5,$C9,$C9,$80,$D0

INIT7   DEFB   $79,$1B,$08,$00,$0B,$EA,$EA,$EA,$EA,$EA,$A9,$01,$CD,$4F,$0C,$D0

INIT8   DEFB   $79,$10,$08,$00,$0B,$4C,$20,$08,$EA,$EA,$EA,$EA,$EA,$60,$EA,$EA
```

5.3 SOU_TRN initial data

When using the SOU_TRN system command, send the following 5 packets of SOU_TRN default data to the register file before SOU_TRN is used.

```
STI
DB      $79, $00, $09, $00, $0B
DB      $AD, $C2, $02, $C9, $09, $D0, $1A, $A9, $01, $8D, $00

ST2
DB      $79, $0B, $09, $00, $0B
DB      $42, $AF, $DB, $FF, $00, $F0, $05, $20, $73, $C5, $80

ST3
DB      $79, $16, $09, $00, $0B
DB      $03, $20, $76, $C5, $A9, $31, $8D, $00, $42, $68, $68

ST4
DB      $79, $21, $09, $00, $01
DB      $60, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00

ST5
DB      $79, $00, $08, $00, $03
DB      $4C, $00, $09, $00, $00, $00, $00, $00, $00, $00, $00
```

THIS PAGE WAS INTENTIONALLY LEFT BLANK.