CHAPTER 2: DISPLAY FUNCTIONS

1. GENERAL DISPLAY FUNCTIONS

1.1 Character Composition

- The basic character size is an 8 x 8-dot composition.
- With characters of the basic size:
  128 OBJ-only characters are available (256 with CGB)
  128 characters can be registered both as OBJ and BG characters (256 with CGB)
  128 BG-only characters are available (256 with CGB)

- On DMG, characters can be represented using 4 shades of gray (including transparent).
- On CGB, characters can be represented using 32 shades for each color of RGB.

- The basic character size can be switched to an 8 x 16-dot composition for OBJ characters only. In this case, however, only even-numbered character codes can be specified.
  Even if an odd-numbered character code is specified, the display will be the same as that seen with an even-numbered code.

- Up to 40 OBJ characters can be displayed in a single screen, and up to 10 characters can be displayed on each horizontal line. (Stored in OAM (Display RAM: 0xFE00-0xFE9F))

- The display data for OBJ characters is stored in OAM (Display RAM: 0xFE00~0xFE9F) in the following order:
  y-axis coordinate
  x-axis coordinate
  Character code
  attribute data

- Data is written to OAM from working RAM by DMA transfer.

- OBJ characters are automatically displayed to the screen using the data written to OAM.

- Data specification ranges for OBJ characters:
  $0x00 \leq$ character code $\leq 0xFF$
  $0x00 \leq X \leq 0xFF$
  $0x00 \leq Y \leq 0xFF$

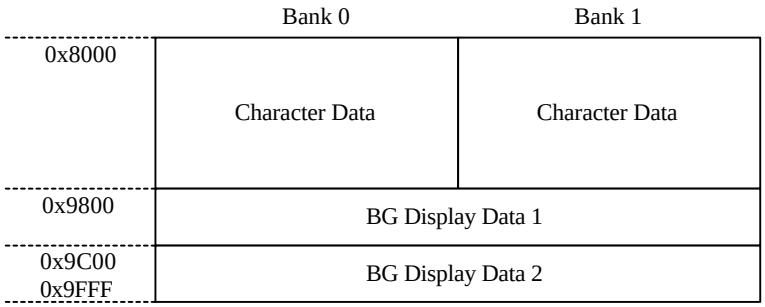
1.2 LCD Display RAM

The DMG CPU has 8 KB (64 Kbits) of built-in LCD display RAM.
The CGB CPU has 16 KB (128 Kbits) of built-in LCD display RAM.

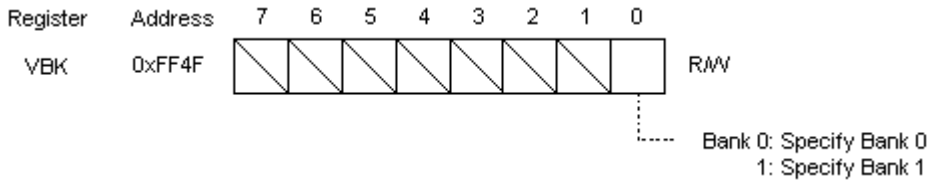
In CGB, 16 KB of memory can be joined in the 8 KB (64-Kbit) memory area (0x8000-0x9FFF) by bank switching using the register VBK (0xFF4F). Bank switching is used exclusively in CGB and cannot be used in DMG mode.

- Mapping of LCD Display RAM

The 16 KB of memory in CGB is partitioned into 2 x 8 KB by register VBK.



- Bank Register (CGB) for LCD Display RAM



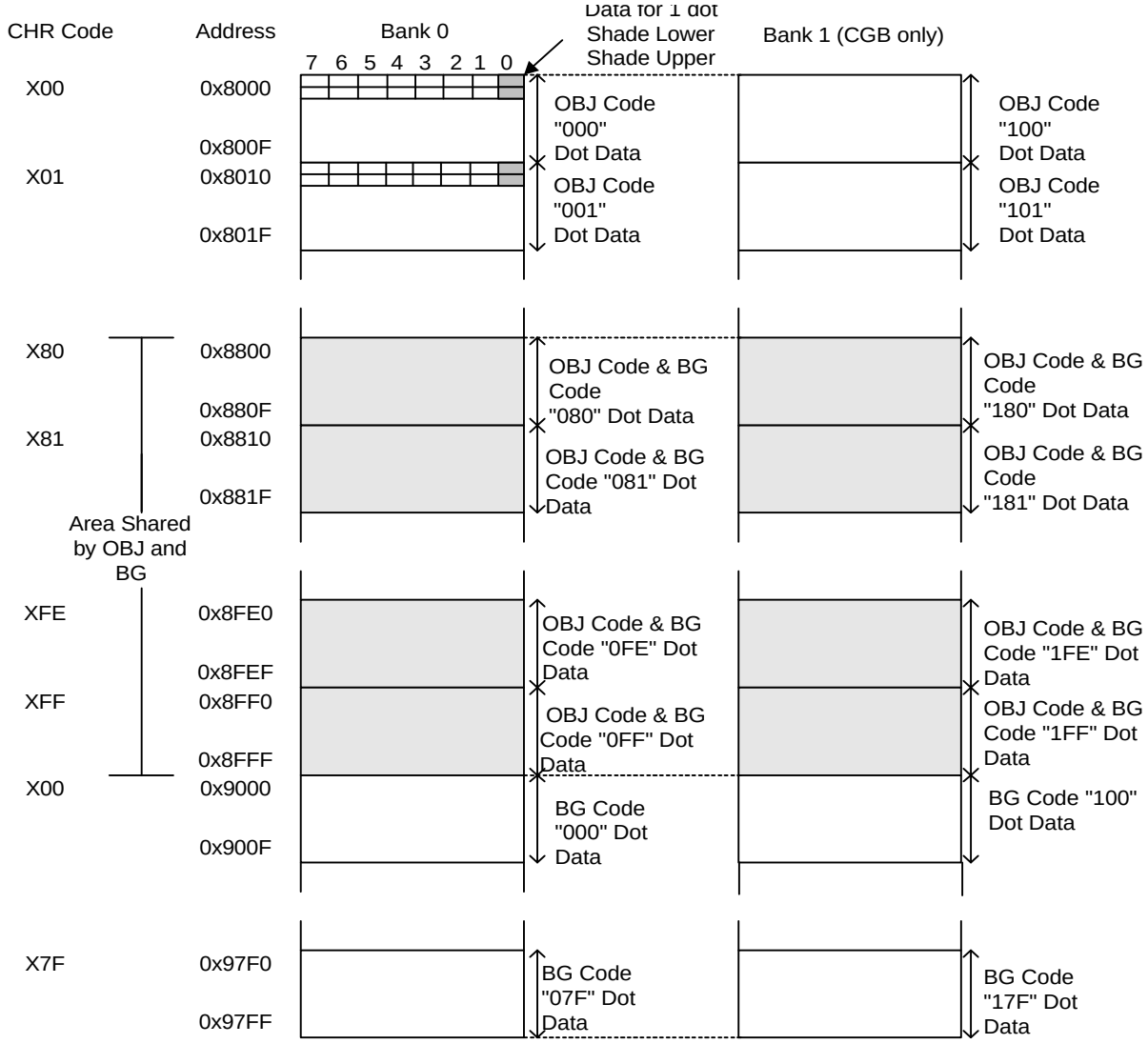
Bank 0 is selected immediately after cancellation of a reset signal.
This function is available only in CGB. In DMG mode, bit 0 is forcibly set to 0, and its value cannot be changed to 1.

1.3 Character RAM

- Character data can be written to the 6144 bytes from 0x8000 to 0x97FF.
- By default, the area from 0x8000 to 0x8FFF is allocated for OBJ character data storage.
- The register LCDC can be used to select either 0x8000-0x8FFF or 0x8800-0x97FF as the area for storing BG and window character data.
- If the BG character data is allocated to 0x8000-0x8FFF, this data shares an area with OBJ data, and the character dot data that corresponds to the CHR codes is also the same.
- By means of bank switching, CGB can store twice the amount of character data in LCD display RAM that DMG can store. In this case, both Bank 1 and Bank 0 have the same mapping as the area in DMG.

Character Code Mapping

With BG character data allocated to 0x8800-0x97FF:



- The case of 8 x 8 dots/block for both BG and OBJ:

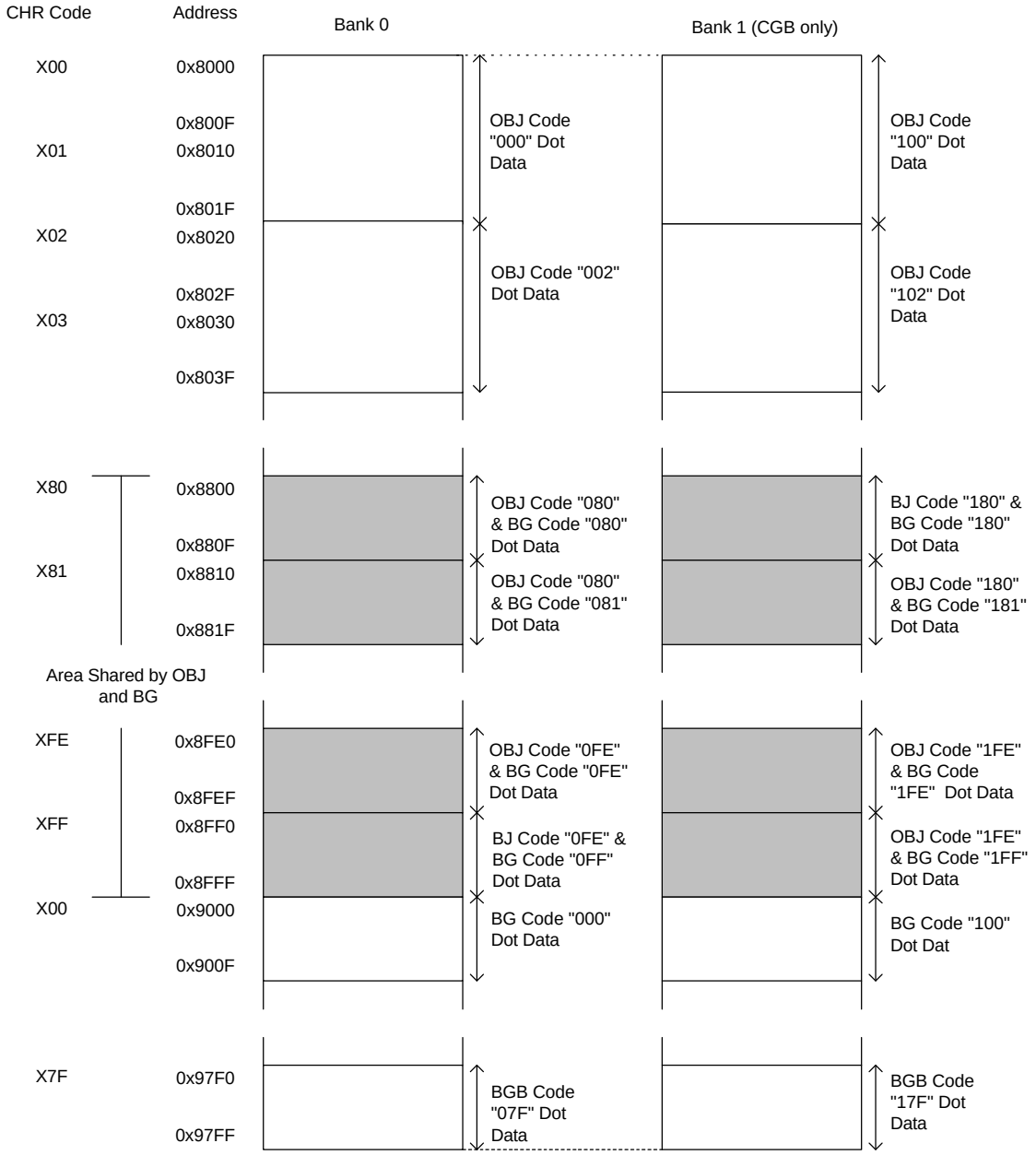
CHR Codes:

<DMG>
OBJ: 256 x 1
BG: 256 x 1

<CGB>
OBJ: 256 x 2
BG: 256 x 2

Note	***Because bank switching is not available in DMG mode, Bank 1 on the right side of the figure is not available in this mode.***

- The case of 8 x 16 dots/block (OBJ) and 8 x 8 dots/block (BG):



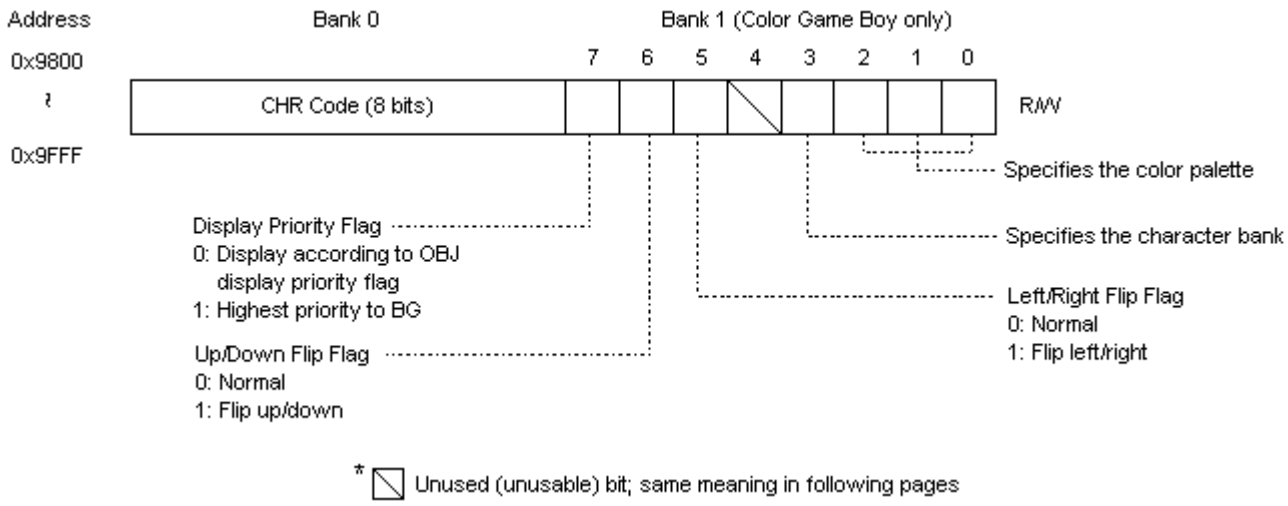
CHR Codes:

<DMG>	<CGB>
OBJ: 128 x 1	OBJ: 128 x 2
BG: 256 x 1	BG: 256 x 2

2) If BG character data is allocated to 0x8000-0x8FFF, these data share an area with OBJ data, and the dot data that correspond to the CHR codes also are the same.

Note	***Because bank switching is not available in DMG mode, Bank 1 on the right side of the figure is not available in this mode.***

1.4 BG Display



Two screens of BG display can be held, Data 1 or Data 2.
Whether the BG display data is allocated to 0x9800-0x9BFF or to 0x9C00-0x9FFF is determined by bit 3 of the LCDC register (0xFF40).

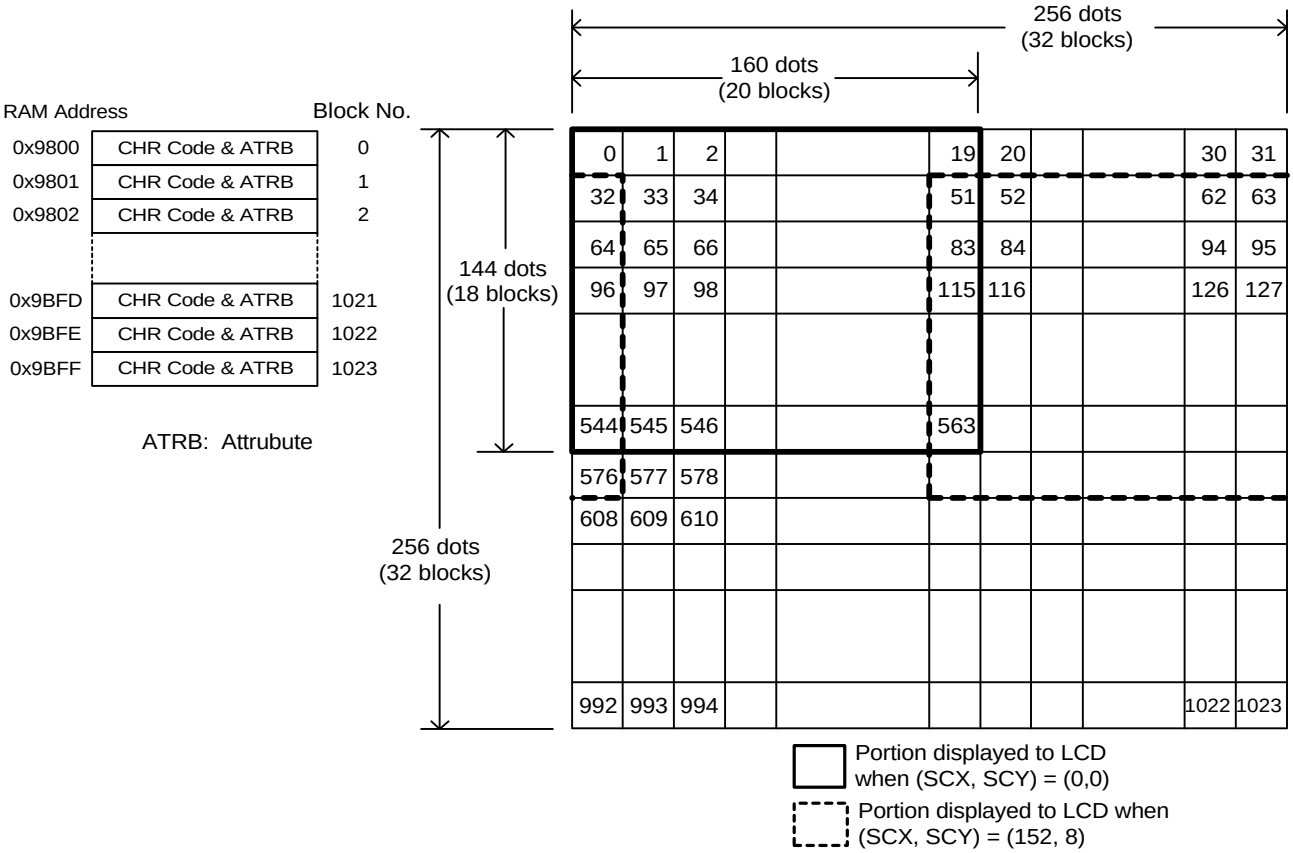
Because bank switching is not available in DMG mode, Bank 1 on the right side of the figure is not present in this mode.

	Bank 0	Bank 1 (CGB only)
0x9800	BG Display Data 1	
0x9C00 0x9FFF	BG Display Data 2	

Data for 32 x 32 character codes (256 x 256 dots) can be specified from 0x9800 or 0x9C00 as BG display data. Of these, data for 20 x 18 character codes (160 x 144 dots) are displayed to the LCD screen.

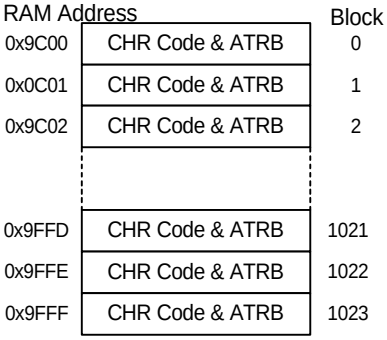
The screen can be scrolled vertically or horizontally one dot at a time by changing the values of scroll registers SCX and SCY.

1) With BG display data allocated to 0x9800-0x9BFF:



Note: Attributes specified only with CGB

2) With BG display data allocated to 0x9C00-0x9FFF:



ATRB: Attribute

Note: Attributes specified only with CGB.

Correspondence between LCD screen and block numbers as shown in preceding figure.

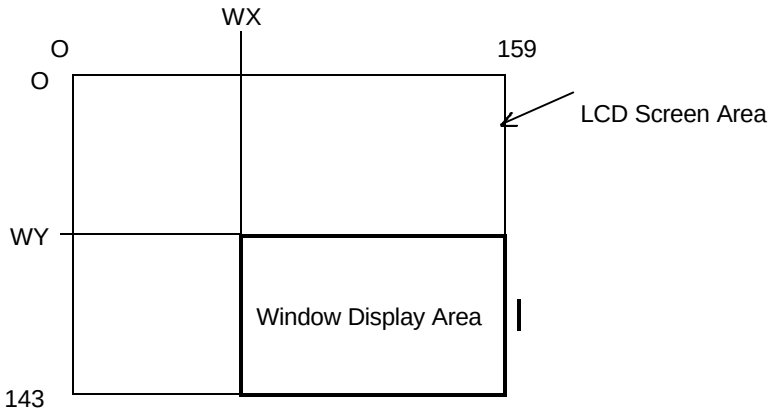
1.5 LCD Screen

- Window Display

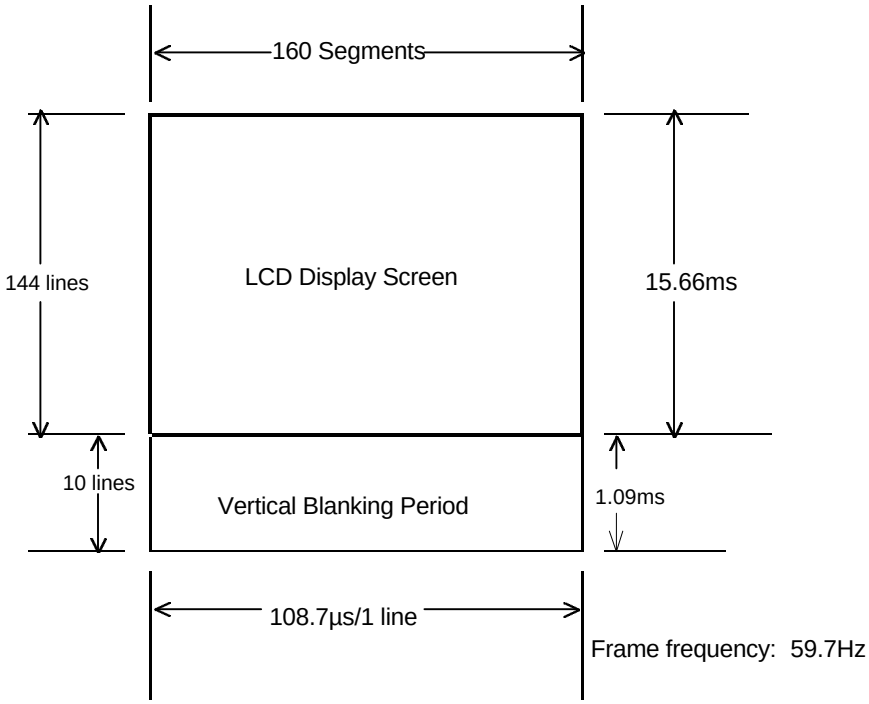
  Specifying a position on the LCD screen using registers WX and WY causes the window to open downward and to the right beginning from that position.

  Window display data also can be specified as character codes, beginning from 0x9800 or 0x9C00 in external SRAM.

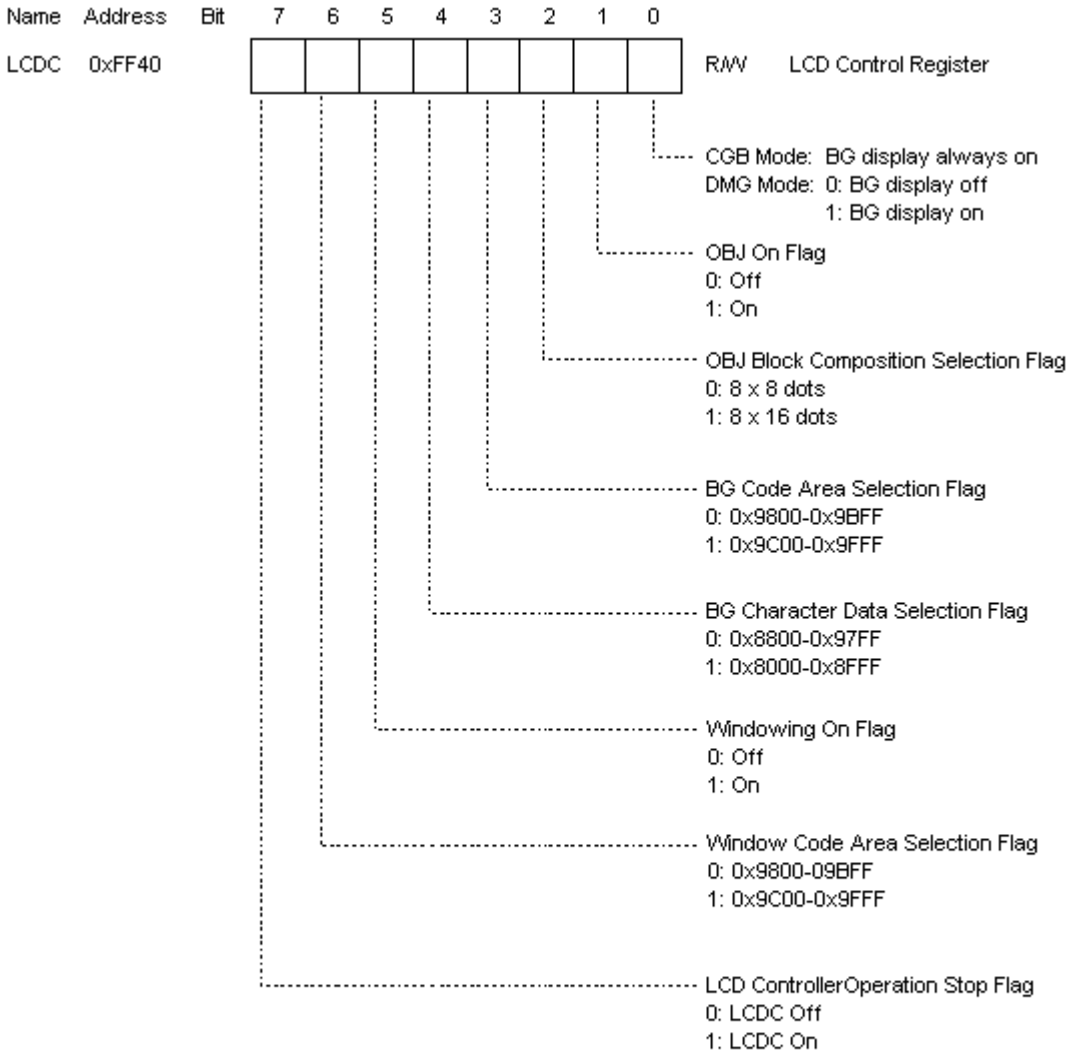
  OBJ character data is displayed in the window in the same way as the BG screen.



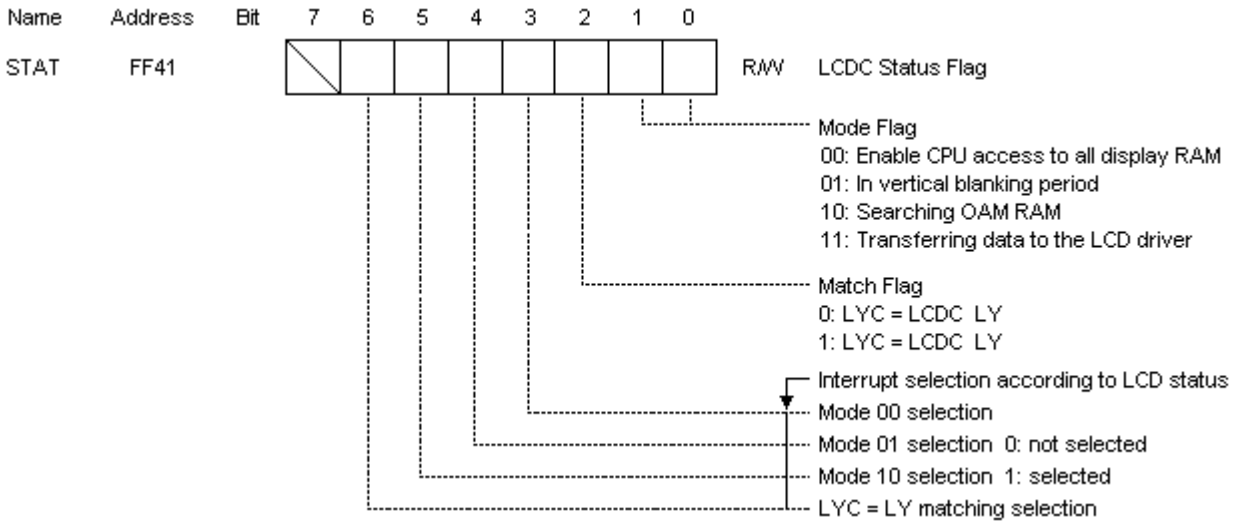
◆ Screen Timing



1.6 LCD Display Registers



* In CGB, the liquid crystal protection circuit functions when the LCDC is turned on. Consequently, a white screen is displayed for up to 2 frames. In DMG, the LCDC should be off during vertical blanking periods.



STAT indicates the current status of the LCD controller.

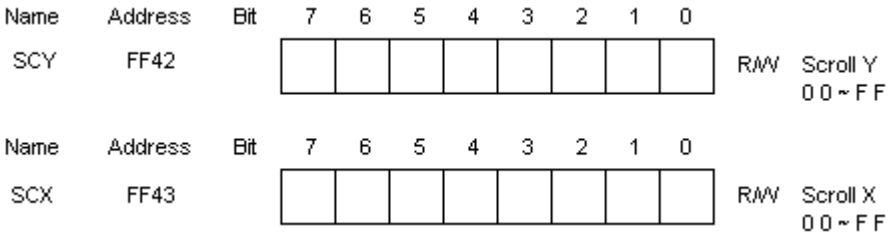
Mode 00: A flag value of 1 represents a horizontal blanking period and means that the CPU has access to display RAM (0x8000-0x9FFF).
When the value of the flag is 0, display RAM is in use by the LCD controller.

Mode 01: A flag value of 1 indicates a vertical blanking period and means that the CPU has access (approximately 1 ms) to display RAM (0x8000-0x9FFF).

Mode 10: A flag value of 1 means that OAM (0xFE00-0xFE90) is being used by the LCD controller and is inaccessible by the CPU.

Mode 11: A flag value of 1 means that the LCD controller is using OAM (0xFE00-0xFE90) and display RAM (0x8000-0x9FFF). The CPU cannot access either of these areas.

In addition, the register allows selection of 1 of the 4 types of interrupts from the LCD controller. Executing a write instruction for the match flag resets that flag but does not change the mode flag.



Changing the values of SCY and SCX scrolls the BG screen vertically and horizontally one dot (or pixel) at a time.

Name	Address	Bit	7	6	5	4	3	2	1	0		
LY	FF44										R	LCDC y-coordinate

LY indicates which line of data is currently being transferred to the LCD driver. LY takes a value of 0-153, with 144-153 representing the vertical blanking period.

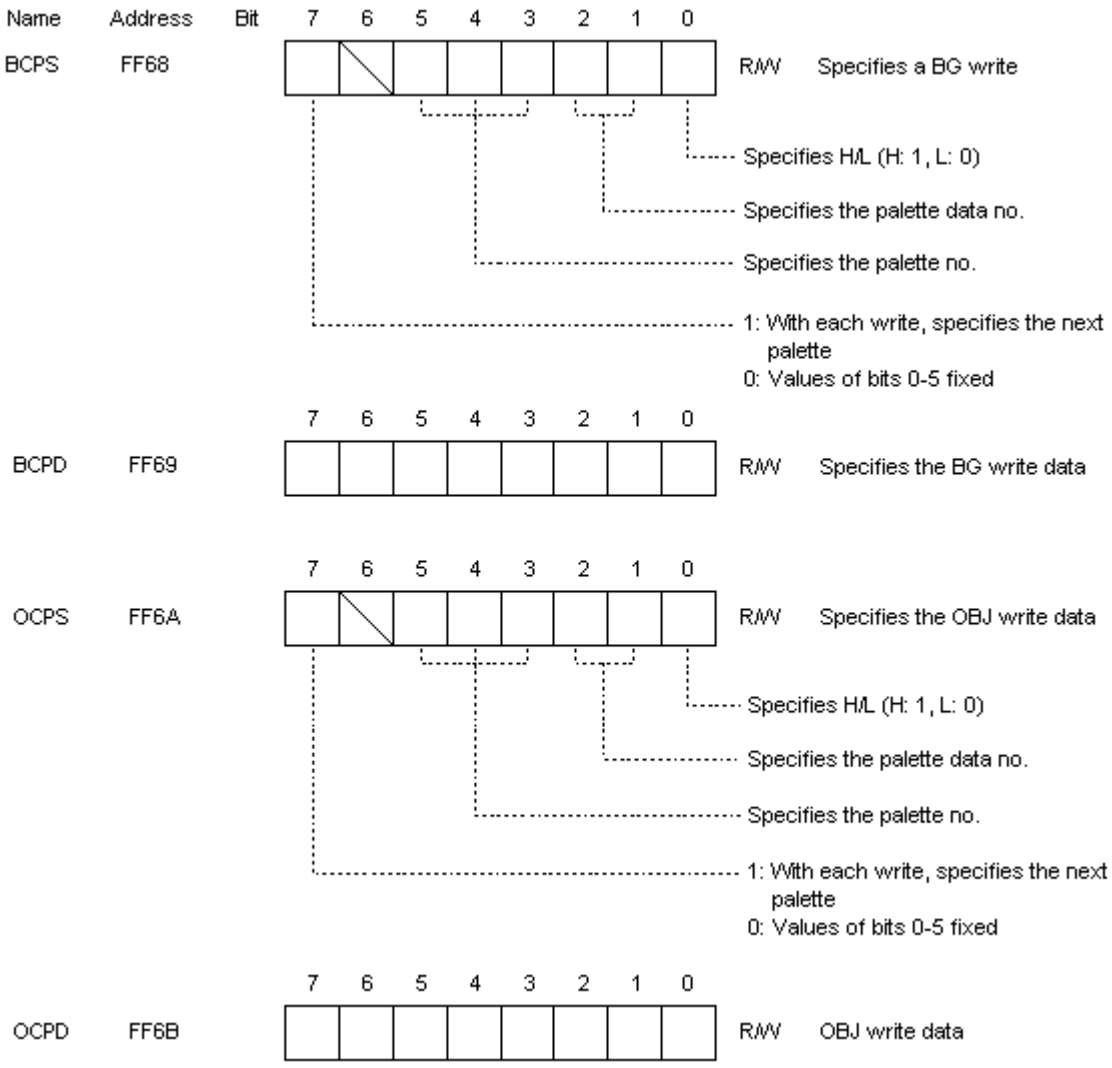
When the value of bit 7 of the LCDC register is 1, writing 1 to this again does not change the value of register LY.

Writing a value of 0 to bit 7 of the LCDC register when its value is 1 stops the LCD controller, and the value of register LY immediately becomes 0. (Note: Values should not be written to the register during screen display.)

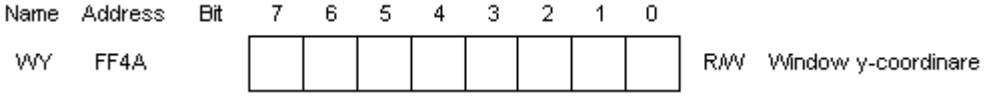
Name	Address	Bit	7	6	5	4	3	2	1	0		
LYC	FF45										R/W	LY Compare

Register LYC is a register compared with register LY. If they match, the Matchflag of the STAT register is set.

NOTE	***The following 3 registers (BGP, OBP0, and OBP1) are valid in DMG and DMG mode of CGB. For information on CGB color palette settings, see Section 3,* LCD Color Display.**

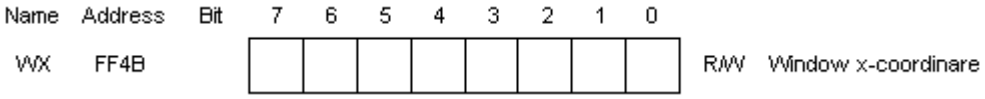


The grayscales (2 bit) for the character dot data is converted by the palette data (BG: register BGP; OBJ: OBP0 or OBP1) and output to the LCD driver as data representing 4 shades (including transparent).



$0 \leq WY \leq 143$

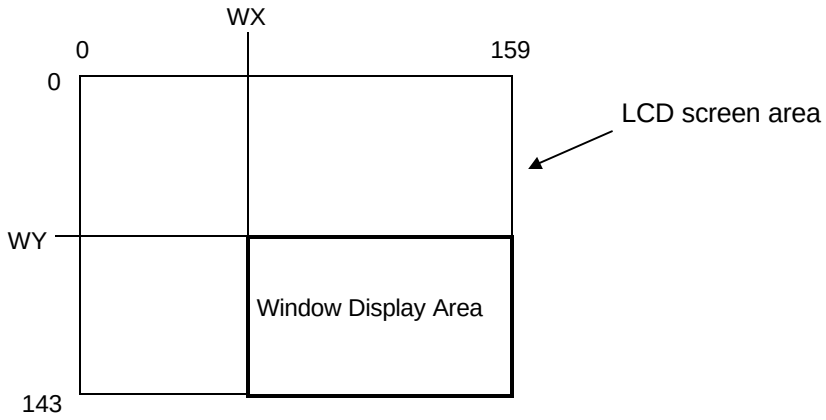
With WY = 0, the window is displayed from the top edge of the LCD screen.



$7 \leq WX \leq 166$

With WX = 7, the window is displayed from the left edge of the LCD screen.

Values of 0-6 should not be specified for WX.



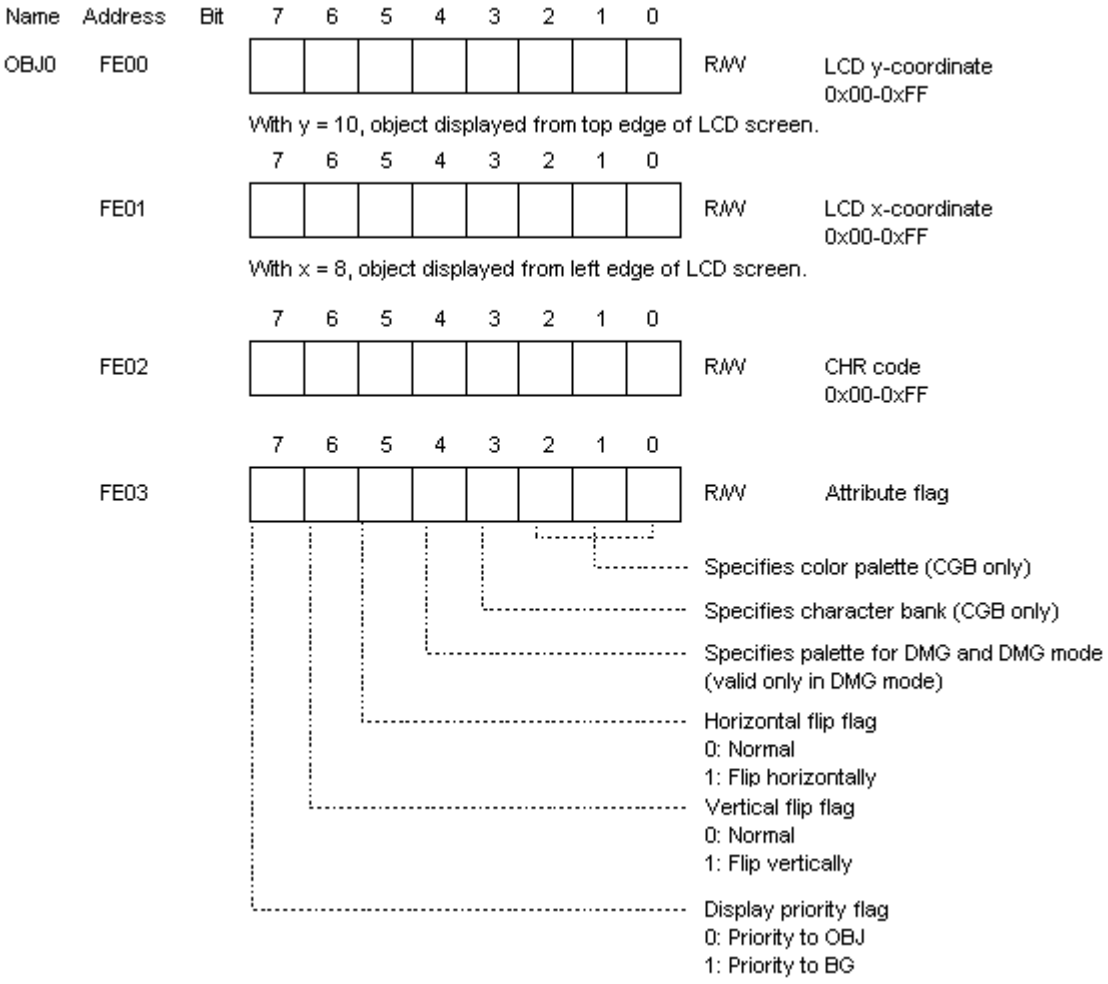
OBJ characters are displayed in the same manner in the window as on BG.

1.7 OAM Registers

OBJ (Object)

- Data for 40 objects (OBJ) can be loaded into internal OAM RAM in the CPU (0xFE00-OxFE9F), and 40 objects can be displayed to the LCD. Up to 10 objects can be displayed on the same Y line.
- Each object consists of a y-coordinate (8 bits), x-coordinate (8 bits), and CHR code (8 bits) and specifications for BG and OBJ display priority (1 bit), vertical flip (1bit), horizontal flip (1 bit), DMG-mode palette, (1 bit), character bank (1bit), and color palette (3 bits), for a total of 32 bits.
- An 8 x 8- or 8 x 16-dot block composition can be specified for an OBJ using bit 2 of the LCDC register. With an 8 x 16-dot composition, the CHR code is specifed as an even number, as in DMG.

OAM Register



OBJ1-OBJ39 have the same composition as OBJ0.

Note	***In DMG mode, the lower 4 bits of the attribute flag are invalid; only the flags in the upper 4 bits including the palette flag are valid.***

1.8 DMA Registers

1.8.1 DMA Transfers in DMG

DMA transfers of 40 x 32 bits of data can be performed from the RAM area (0x8000-0xDFFF) to OAM (OxFE00-0xFE9F). The transfer time is 160 μs.

Note that in DMG, data cannot be transferred by DMA from ROM area 0x0000-0x7FFF.

The starting address of a DMA transfer can be specified as 0x8000-0xDFFF in increments of 0x100.

Note that the method used for transfers from 0x8000-0x9FFF (display RAM) is different from that used for transfers from other addresses.

Example 1

The following example shows how to perform a DMA transfer of 40 x 32 bits from the expansion RAM area (0xC000-0xC09F) to OAM (0xFE00-0xFE9F).

During DMA, the CPU is run using the internal RAM area (0xFF80-0xFFFE) to prevent external bus conflicts.

1. The program writes the following instructions to internal RAM (0xFF80-0xFFFE):

Address	Machine Code	Label	Instruction		Comment
FF80	3E C0		LD	A, 0C0H	
	E0 46		LD	(DMA) , A	;C000-C09F→OAM
	3E 28		LD	A, 40	;160-cycle wait
	3D	L1:	DEC	A	
	20 FD		JR	NZ, L1	
	C9			RET	

2. Example of program that writes the above instructions to internal RAM starting from 0xFF80:

```
                Label   Instruction

                LD      C, 80H
                LD      B, 10
                LD      HL, DMADATA
L2:             LD      A, (HLI)
                LD      (C), A
                INC     C
                DEC     B
                JR      NZ, L2
                 •
                 •
                 •

DMADATA         DB      3EH,  0C0H,  0E0H,  46H,  3EH
                DB      28H,  3DH,  20H,  0FDH,  0C9H
```

3. When the DMA transfer is performed, the subroutine written to internal RAM shown in Step 1 above is executed:

```
•
CALL    0FF80H   :DMA transfer
•
```

> ***Note*** ***The preceding program is used for DMA transfers performed within routines for processing interrupts implemented by vertical blanking. In all other cases, however, the program written to internal RAM should be as shown below to prevent interrupts during a transfer.***

```
Address  Machine Code  Label  Instruction          Comment
FF80     F3                   DI                   :Interrupt disabled
         3E C0                LD  A, 0C0H
         E0 46                LD (DMA), A          :C000~C09F→0AM
         3E 28                LD  A, 40            :160-cycle wait
         3D           L1:     DEC A
         20 FD                JR  NZ, L1
         FB                   EI                   :Interrupt enabled
         C9                   RET
```

Example 2

The example below shows a DMA transfer of 40 x 32 bits of data from the display RAM area (0x9F00-0x9F9F) to OAM (0xFE00-0xFE9H).

```
Machine Code  Label  Instruction          Comment

3E 9F         LD     A, 9FH
E0 46         LD     (DMA), A             :9F00~9F9F→0AM
```

Data can be transferred by DMA from 0x8000-0x9F9F to OAM either by the method shown in Example 1 or by using only the above instructions.

1.8.2 DMA Transfers in CGB

Using the Earlier DMA Transfer Method

This DMA method transfers only 40 x 32 bits of data from 0-0xDFFF to OAM (0xFE00-0xFE9F). The transfer starting address can be specified as 0-0xDFFF in increments of 0x100. The transfer method is the same as that used in DMG, but when data is transferred from 0x8000-0x9FFF (LCD display RAM area), the data transferred are those in the bank specified by bit 0 of register VBK. When transferring data from 0xD000-0xDFFF (unit working RAM area), the data transferred are those in the bank specified by the lower 3 bits of register SVBK.

> ***Note*** ***When the CPU is operating at double-speed, the transfer rate is also doubled.***

Using the New DMA Transfer Method

The DMA transfer method provided for DMG has been augmented in CGB with the following DMA transfer functions.

1) Horizontal Blanking DMA Transfer

Sixteen bytes of data can be automatically transferred from the user program area (0-0x7FFF) or external and unit working RAM area (0xA000-0xDFFF) to the LCD display RAM area (0x8000-0x9FFF) during each horizontal blanking period. The number of lines transferred by DMA in a horizontal blanking period can be specified as 1-128 by setting register HDMA5. CPU processing is halted during a DMA transfer period.

2) General-Purpose DMA Transfers
Between 16 and 2048 bytes (specified in 16-byte increments) are transferred from the user program area (0-0x7FFF) or external and unit working RAM area (0xA000-0xDFFF) to the LCD display RAM area (0x8000-0x9FFF). As with horizontal blanking DMA transfers, CPU operation is halted during the DMA transfer period.

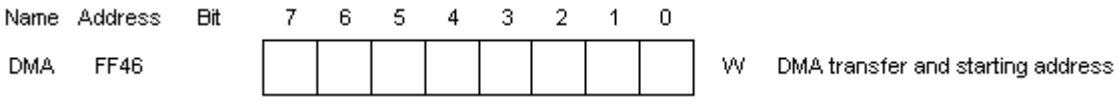
The unit working RAM area (0xD000-0xDFFF) selected as the transfer source is the bank specified by register SVBK.

The LCD display RAM area (0x8000-0x9FFF) selected as the transfer destination is the bank specified by register VBK.

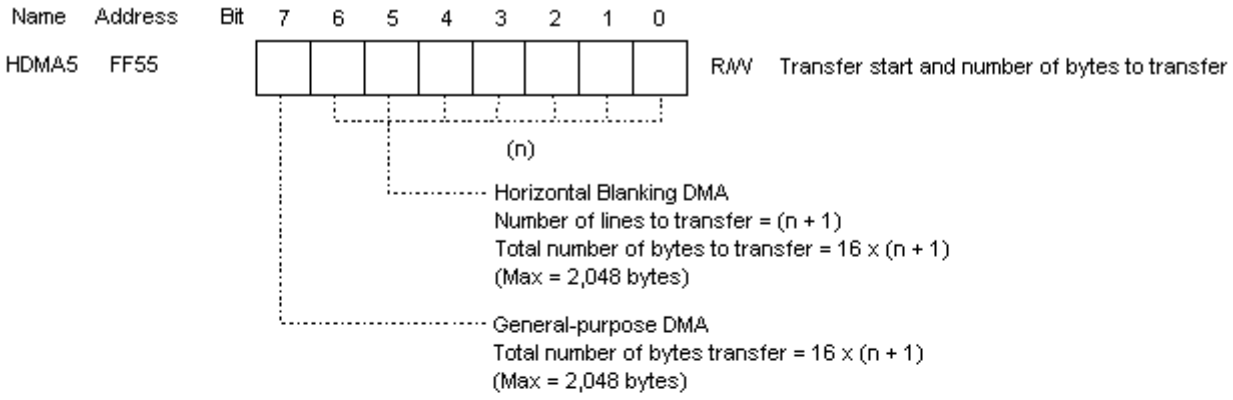
Special Notes

- The number of bytes transferred by the new DMA method must be specified in 16-byte increments; byte counts that are not a multiple of 16 cannot be transferred.
- With the new DMA transfer method, transfers are performed at a fixed rate regardless of whether the CPU is set to operate at normal or double-speed.
- Horizontal blanking DMA transfer should always be started with the LCDC on and the STAT mode set to a value other than 00.
- General-purpose DMA transfer should be performed with the LCDC off or during a vertical blanking period.
- When the new DMA transfer method is used to transfer data from the user program area (0-0x7FFF), mask ROM and MBC for double-speed mode are **required**.

1.8.3 DMA Control Register: For both DMG and CGB



1.8.4 New DMA Control Registers: CGB only



Horizontal blanking DMA

Value of 1 written:
After 1 is written, horizontal blanking DMA transfer is started from the first horizontal blanking period.
(DMA should always be started with LCDC on and value other than 00 for STAT mode.)

* When a value of 0 has been subsequently written, DMA transfer stops beginning with the next horizontal blanking period.

General-purpose DMA

Value of 0 written (the following applies only when the bit is already 0):
General-purpose DMA starts
(DMA should be started with LCDC off or during a horizontal blanking period. Ensure that the transfer period does not overlap with STATE mode settings of 10 or 11.)

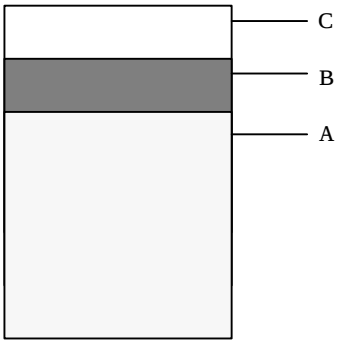
* Only input of a reset signal can halt a general-purpose DMA transfer in progress.

Note These registers cannot be written to in DMG mode.

1.9 OBJ Display Priority

As a rule, when objects overlap, the one with the lower OBJ number is given priority. In DMG or DMG mode of CGB, among overlapping objects with different x-coordinates, priority is given to the object with the smallest x-coordinate.

1) The case with the same x-coordinate: For both DMG and CGB



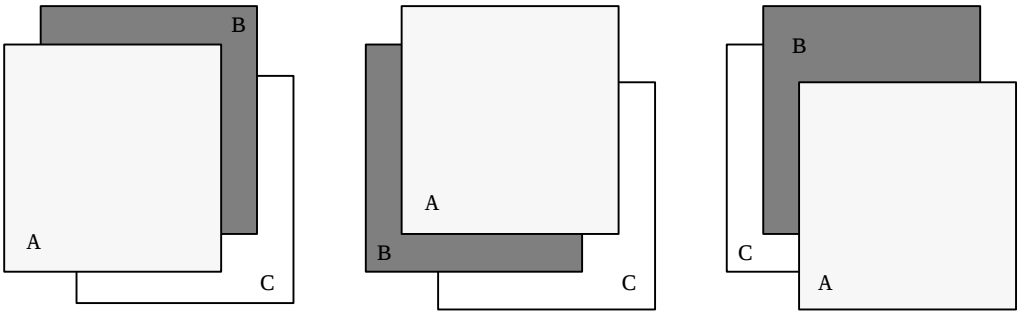
a = No. of OBJ A
b = No. of OBJ B
c = No. of OBJ C

When a < b < c, objects are displayed as indicated in the figure at left.

2) The case with different x-coordinates: CGB only

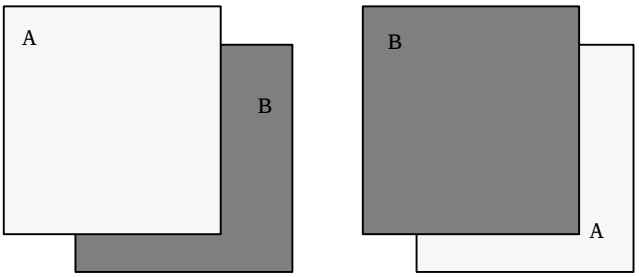
a = No. of OBJ A
b = No. of OBJ B
c = No. of OBJ C

When a < b < c, objects are is displayed as indicated in the figures below.



3) Different x-coordinates: DMG/CGB in DMG mode

In DMB mode and with objects with different x-coordinates, the object with the smallest x-coordinate is given priority.



a = No. of OBJ A
b = No. of OBJ B

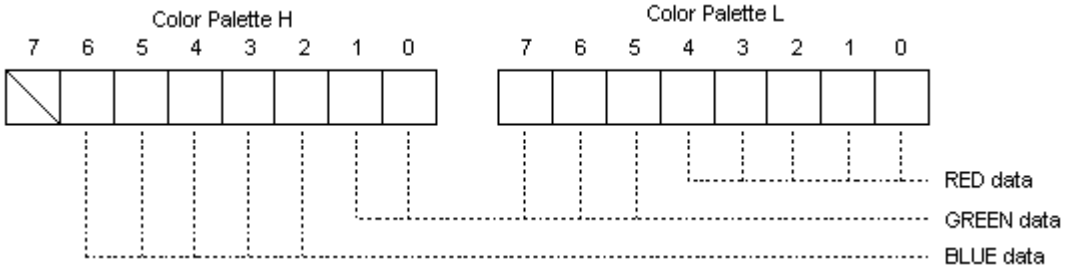
When a < b, objects are displayed as indicated in the figure at left.

2. LCD COLOR DISPLAY (CGB ONLY)

The LCD unit of the CGB system can display 32 shades each for RGB, for a total 32,768 colors. A single color palette consists of 4 colors selected from among these 32,768 colors. One of 8 palettes can be selected for each BG and OBJ character. However, because each OBJ includes transparent data, each OBJ color palette consists of 3 colors. The color palettes for BG and OBJ are independent of one another.

2.1 Color Palettes

- Eight palettes each are provided for BG and OBJ.
- Each palette consists of 4 colors and is specified by the display dot data (2 bits) (Palette data numbers 0-3).
- The color palettes represent each color with 2 bytes, with 5 bits of data for each color of RGB (32,768 displayable colors).



2.2 Color Palette Composition

1. BG Color Palettes

Color Palette No.			Palette Data No.
Color palette 0	Color palette H00	Color palette L00	0
	Color palette H01	Color palette L01	1
	Color palette H02	Color palette L02	2
	Color palette H03	Color palette L03	3
Color palettes 1-7			

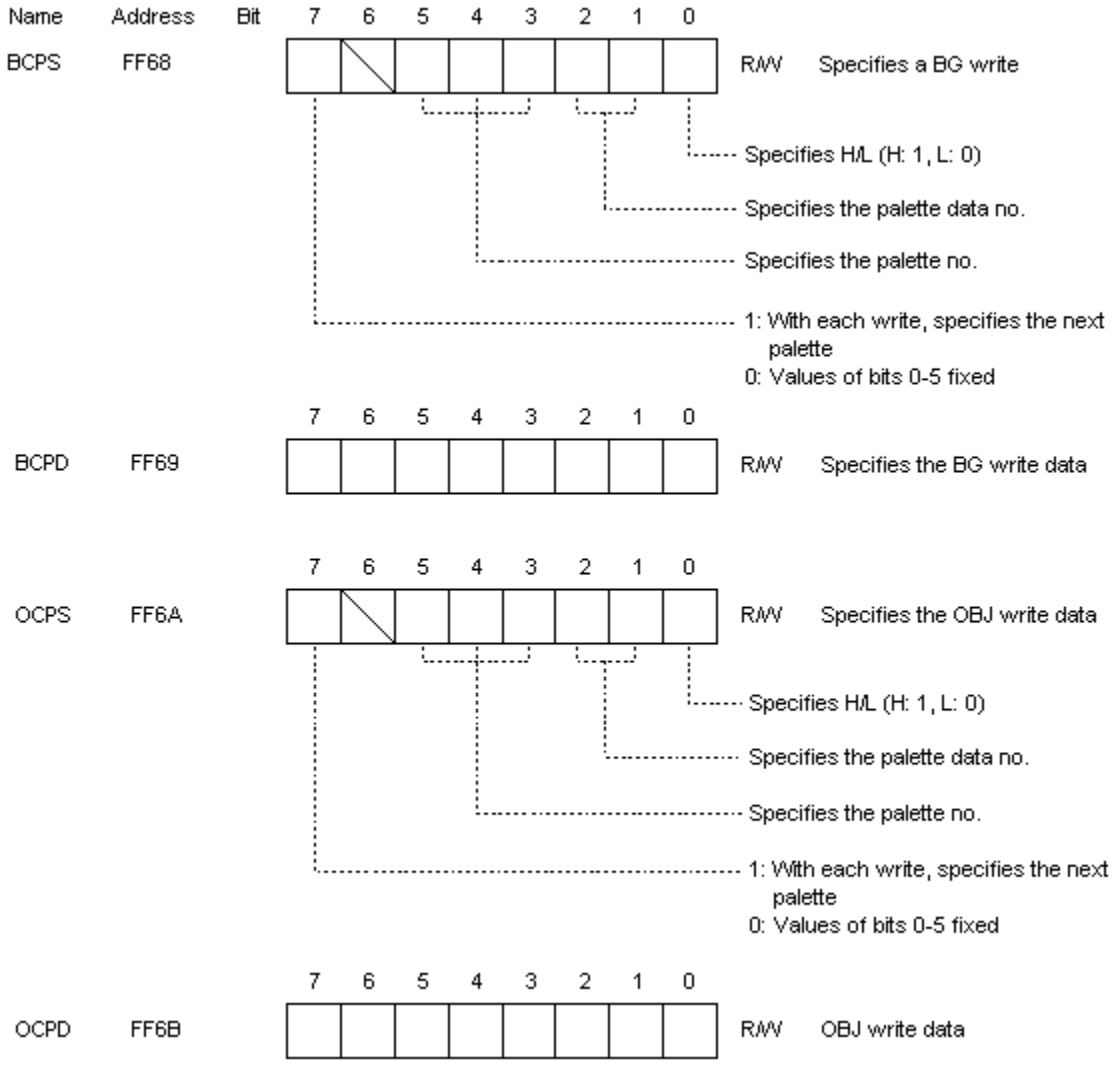
2. OBJ Color Palettes

OBJ color palettes have the same composition as shown in the figure above.

2.3 Writing Data to a Color Palette

Data is written to color palettes using the write-specification and write-data registers. The lower 6 bits of the write-specification register specifies the write address. When data is written to the write-data register, the data will be written to the address specified by the write-specification register. If the highest bit of the write-specification register is set to 1, the write address is then automatically incremented to specify the next address. (The next address is read from the lower 6 bits of the write-specification register.)

The write-specification and write-data registers also are used to read data from color palettes. When the write-data register is read, the data at the address specified by the write-specification register is read. When data is read, the specified address is not incremented even if the most-significant bit of the write-specification register is set to 1.



Note ***These registers cannot be written to in DMG mode.***

2.4 Overlapping OBJ and BG

When objects are displayed, overlapping objects and background are displayed according to the display priority flags for OBJ and BG, as indicated below. The BG display priority flag can be used to assign BG display priority to individual characters.

Display Priority Flag		Dot Data		Screen Display	
BG	***OBJ***	OBJ	***BG***	***Palette***	Data
0: Use OBJ priority	0: Priority to OBJ	00 00 obj obj	00 bg 00 bg	BG BG OBJ OBJ	00 bg obj obj
	1: Priority to BG	00 00 obj obj	00 bg 00 bg	BG BG OBJ BG	00 bg obj bg
1: Highest priority to BG (by character)	0 1	00 00 obj obj	00 bg 00 bg	BG BG OBJ BG	00 bg obj bg

* obj and bg represent dot data (01, 10, 11) for OBJ and BG, respectively.

2.5 Display Using Earlier DMG Software (DMG mode)

When earlier DMG software is used, coloring is performed automatically by the system using registers BGP, OBP0, and OBP1. However, the display uses 3 palettes, 1 for BG, with 4 colors, and 2 for OBJ, each with 3 colors (excluding transparent; maximum of 10 colors in 1 screen).

1. BG Display

   Colors specified in BG color palette No. 0 are displayed by the dot data (2 bits) whose grayscales are specified by register BGP.

2. OBJ Display

   Colors specified in OBJ color palettes No. 0 and No. 1 are displayed by the dot data (2 bits) whose grayscales are specified by registers OBP0 and OBP1.

The CGB unit automatically selects the display color according to the color palette pre-registered in the CGB (cannot be changed by a program). However, when turning on power to the CGB, the player can select from a combination of the 12 colors registered in the unit. This function is available only in DMB mode.

THIS PAGE WAS INTENTIONALLY LEFT BLANK.