Preface

With the recent advent of home video-game machines that offer a fast, responsive man-machine interface and high-resolution graphics at a low price, there is a demand for a similar revolution in graphics quality in the entire home information market, covering such products as car navigation systems and Internet TVs.

In addition to the need for a powerful and simple means of handling the necessary processing, there are growing demands for the preservation of upward-compatibility of software and data bases. At the same time, there is the question of how best to handle the increasingly complex graphics algorithms required for these applications.

In response to this demand, Hitachi has developed the Q Series of graphics accelerators, offering high-speed rendering and display processing in a chip set that includes a SuperH family microcomputer.

The Q (“Quick”) series is designed for use in a chip set with a SuperH family microcomputer to provide a compact system capable of the high-speed geometric, rendering, and display processing required by graphics display systems.

Q Series devices perform rendering and display processing compatible with the SuperH family, based on the concepts of simplicity, realtime operation, and upgradability.

The HD64412 (Q2i) is a 2D graphics renderer for minimum system configuration use within this series, featuring minimized graphics memory (one 4-Mbit memory), unified handling of graphics and natural images, realtime software 3D graphics drawing, and improved system bus utilization. The Q2i makes it possible to implement a compact graphics processing system with high-level drawing capabilities for a wide range of multimedia terminals, including car navigation systems, medium-definition OA products such as Internet computers, and display systems in industrial equipment, as well as Internet TVs, karaoke systems, and similar AV products.

For examples of the use of this LSI, see Q Series Application Notes—HD64412Q2i.

Section 1 Overview of Q2i (Quick 2D Graphics Renderer Improved)

1.1 Q2i Overview

The Q2i (Quick 2D Graphics Renderer Improved) is a 2D graphics renderer for minimum system configuration use in the SuperH microcomputer graphics accelerator “Quick” series (Q series), based on the concepts of simplicity, realtime operation, and upgradability.

The use of unified graphics memory and a double-buffering system that switches drawing and display buffers in frame units, providing a high-speed drawing performance of 60 screens per second, has made possible minimization of graphics memory (with the use of a single 4-Mbit memory), unified handling of graphics and natural images, and realtime software 3D graphics drawing. The clear separation of geometric operations (handled by the CPU) and rendering operations (handled by the Q2i) has also resulted in improved system bus utilization.

New developments and upgrades can be implemented easily by means of the Q2i’s interface software, allowing attention to be focused on the development of application software.

The Q2i is a high-performance graphics rendering LSI for multimedia applications, which provides both drawing and display functions integrated into a single chip.

Figure 1-1 shows an overview of the Q2i system.

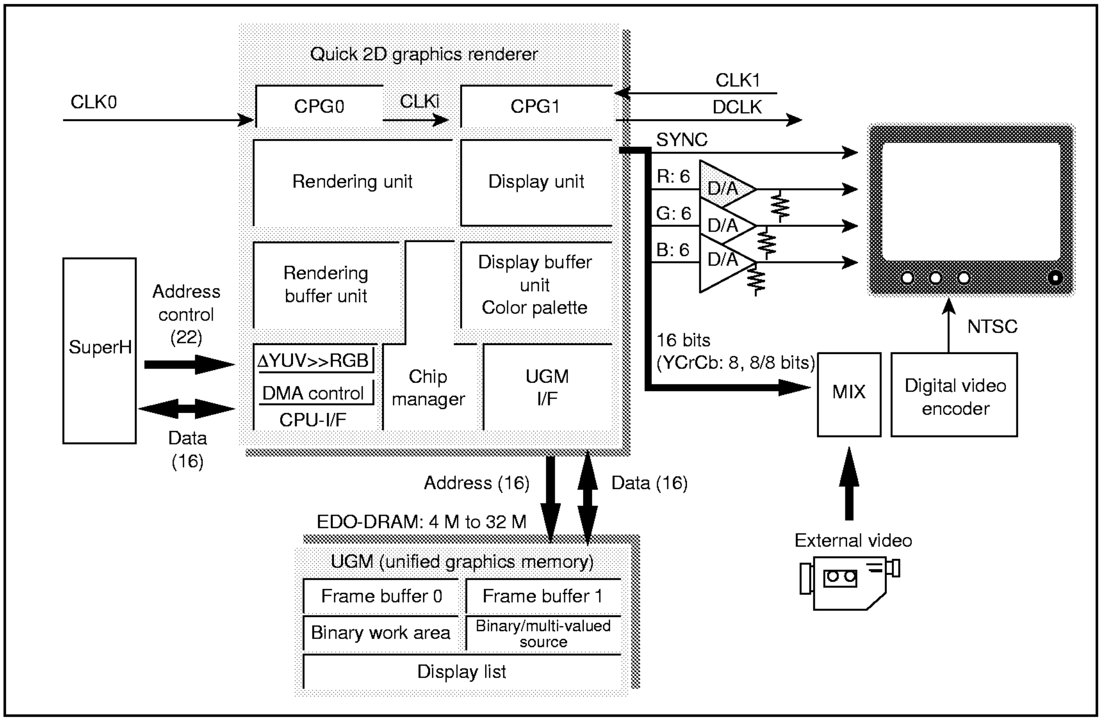


Figure 1-1 Overview of Q2i System

1.2 Block Diagram

Figure 1-2 shows a block diagram of the Q2i. The functions of the various blocks in figure 1-2 are as follows.

- Rendering unit

  Performs fetching and interpretation of the display list in the UGM, references the source data in the UGM, and outputs drawing data to the drawing-side frame buffer in the UGM.
- Rendering buffer unit

  Buffers data and addresses between the rendering unit and the UGM, and outputs them efficiently.
- CPU interface unit

  Performs control relating to connection to the CPU bus.
- Memory interface unit

  Performs control relating to connection to the UGM bus.
- Display unit

  Controls the control signals sent to the CRT device.

- Display buffer unit

  Reads data to be displayed on the CRT from the display-side frame buffer, and outputs the display data in accordance with the display timing.

- Color palette (6 bits per color, 64 gradation settings)

  When using 8 bits/pixel, performs conversion to display data of 256 colors out of 262,144, based on the color conversion table.

- (YUV (YUV): RGB conversion

  Converts input data (YUV (262,144 colors) or YUV (262,144 colors) to RGB data (65,536 colors), and stores it in the UGM.

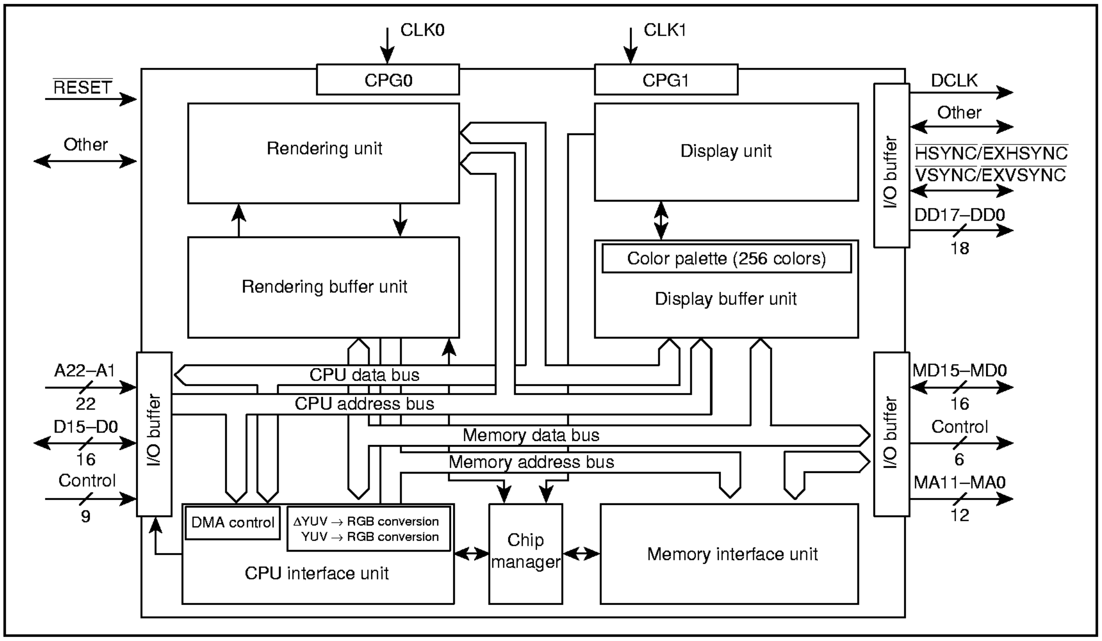


Figure 1-2 Internal Block Diagram

1.3 Concepts

1.3.1 Simplicity (Optimization of System Configuration)

Use of Unified Graphics Memory Architecture

- Unified handling of image data (unified graphics memory (UGM) architecture)

  Data in various formats can be stored and managed in the same unified graphics memory (figure 1-3).
- Minimum necessary UGM

  Minimum UGM configuration: One 16-bit-data-bus type 4-Mbit EDO page mode DRAM

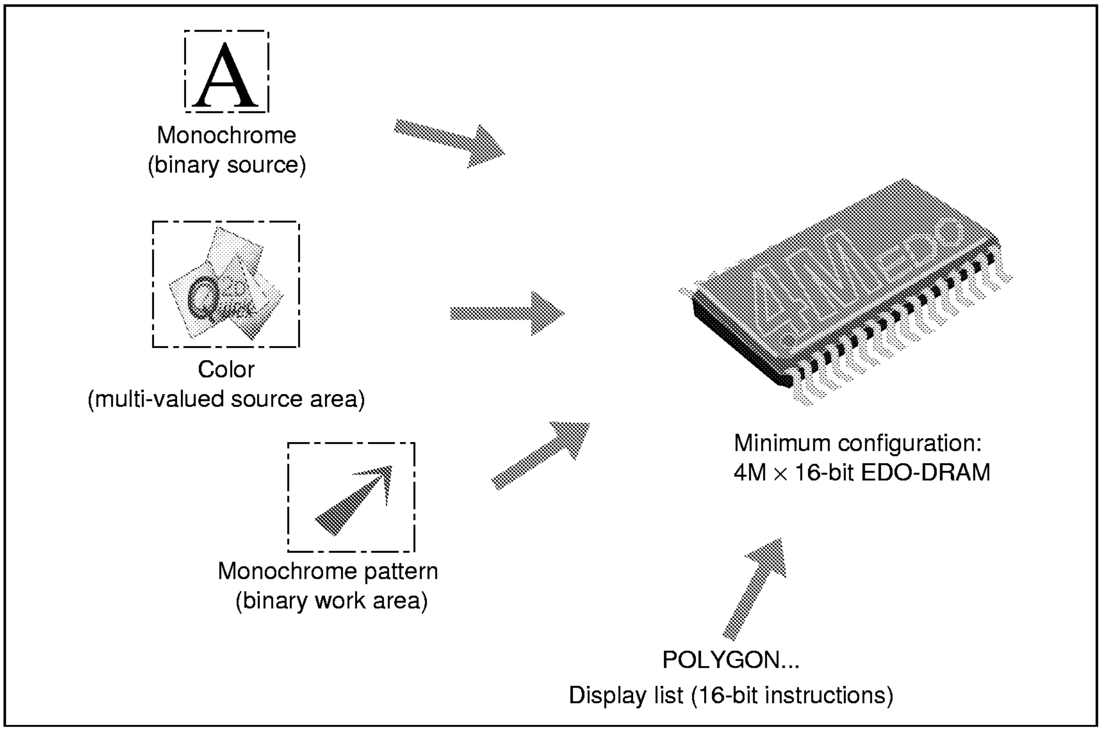


Figure 1-3 Reduced System Size Through Use of UGM Architecture

- Unified system bus interface

  A CPU interface circuit is incorporated to provide a unified interface. This enables unified graphics memory to be allocated in the SuperH's memory space without regard to the type of SuperH used. (See figure 1-4.)

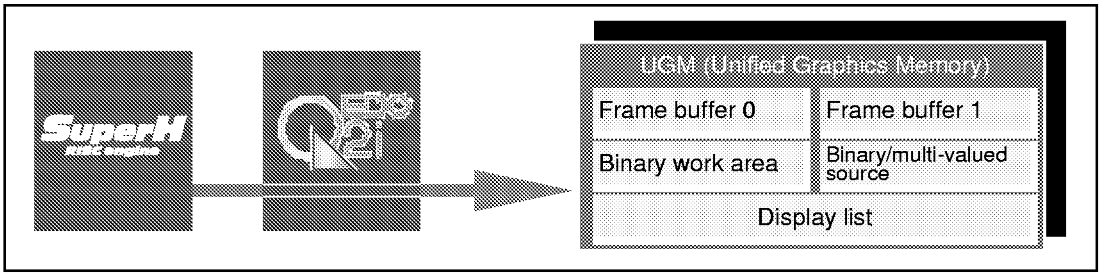


Figure 1-4 Unified System Bus Interface (UGM Directly Accessible by SuperH via Q2i)

1.3.2 Realtime Operation

Use of Double-Buffering Architecture: The use of a double-buffering architecture that allows switching between the drawing buffer and display buffer in frame or field units, together with the use of EDO page mode DRAM for the UGM, enables realtime operation by alternating display processing with high-speed drawing processing. (See figure 1-5.)

- Double-buffer control
  - Kinds of double-buffer control
    - **Auto display change mode:** Mode in which priority is given to display frame switching. If drawing is in progress when frame switching is to be performed, drawing is forcibly terminated midway.
    - **Auto rendering mode:** Mode in which display switching is not performed until drawing ends. If drawing does not end within one frame, drawing is continued without interruption and frame switching is performed at the frame boundary immediately after drawing is completed.
    - **Manual display change mode:** Mode in which display frame switching and the start of drawing are controlled by software. When the display area change bit (DC bit) is set after drawing is completed, frame switching is performed at the next frame boundary.
  - Double-buffer switching timing
    - **Non-interlace mode:** Scanning system in which one frame is composed of one field. Double-buffer switching is performed in units of a frame.
    - **Interlace mode:** Scanning system in which one frame is composed of two fields. Double-buffer switching is performed in units of a frame.
    - **Interlace sync & video mode:** Scanning system in which one frame is composed of two fields. Double-buffer switching is performed in units of a field.

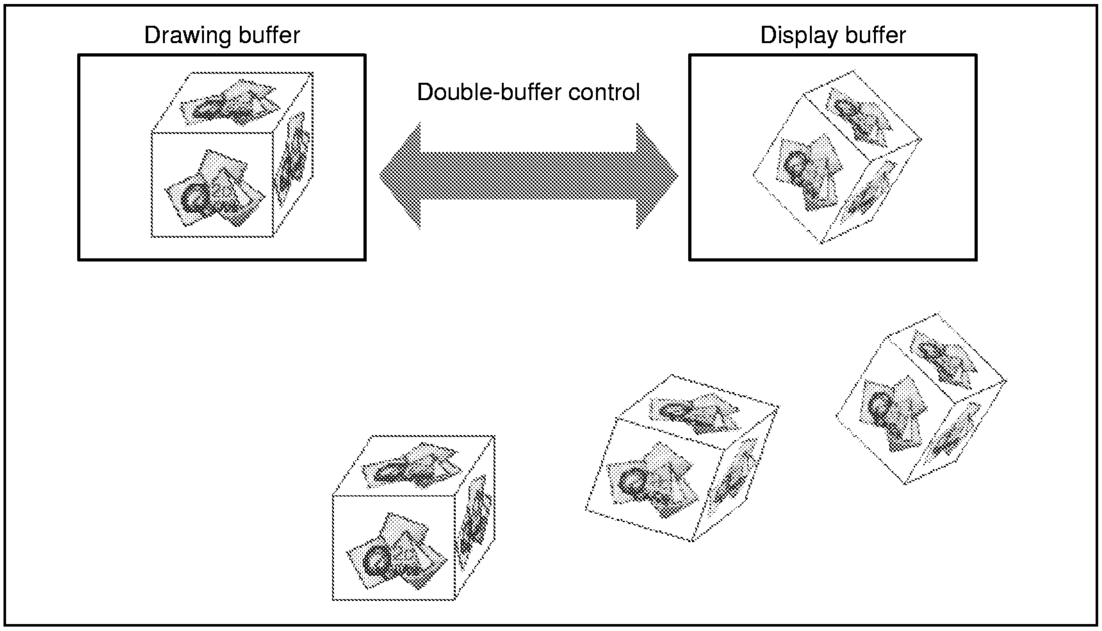


Figure 1-5 Double-Buffering Architecture

Support for EDO Page Mode DRAM: EDO page mode DRAM can be used for the UGM. This enables the Q2i to use burst access to the UGM and perform high-speed drawing.

Use of Write-Only Drawing: Write-only drawing (a drawing method using only write operations) is used to improve drawing performance.

1.3.3 Upgradability

Algorithm Upgrading: In the Q2i's drawing system, algorithms for coordinate conversion, etc., are executed by the SuperH, using a systematized data base containing coordinates and other data, and the results are represented in graphical form. Thus, the graphics for a variety of shapes can be implemented simply by upgrading the algorithms, without having to modify the data base. (See figure 1-6.)

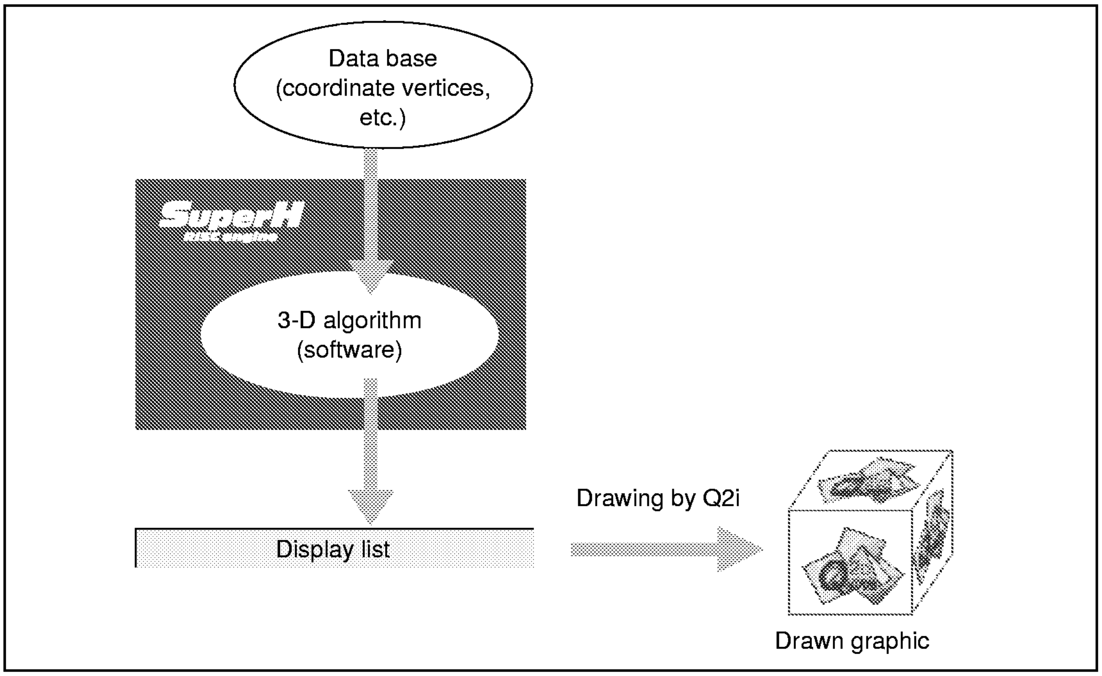


Figure 1-6 Data Flow when Using a 3D Algorithm

Drawing System Upgrading: The Q2i is available as a series—the Q Series—in the same way as the SuperH, enabling the user to select the most appropriate Q2i and SuperH models for his application. The user's drawing system can also be upgraded as necessary by changing the Q2i or SuperH combination.

Consistency of Application Interface: The Q2i's carefully selected drawing commands are of four kinds: four-vertex surface drawing, line drawing, work surface drawing, and work line drawing. This makes it possible to reduce the parts dependent upon drawing commands within an application, and so achieve a more consistent interface between applications.

1.4 Summary of Functions

Table 1-1 summarizes the functions of the Q2i.

Table 1-1 Summary of Q2i Functions

Item			Function/Performance
Maximum clock frequency	Drawing system internal operation (operating clock)		Multiplication on: 33 MHz × 1, 16.5 MHz × 2, 8.25 MHz × 4 Multiplication off: 33 MHz
	Display system internal operation (display dot clock)		Operating clock/2
Display functions	Sample screen size		320 × 240 dots (standard size in non-interlace operation) 640 × 480 dots (standard size in interlace & video operation)
	CRT scanning system		Non-interlace, interlace, interlace sync & video
	External synchronization		Master, TV synchronization
	Display colors		256 colors (selectable from 262,144) or 65,536 colors
Drawing functions	Drawing commands	Drawing related	4-vertex surface drawing, line drawing, work surface drawing, work line drawing
		Register setting related	Current pointer setting, local offset setting, clipping
		Sequence control related	Jump, Subroutine call
	Coordinate systems		Drawing coordinate system: Rendering coordinates, work coordinates Source coordinate system: Binary source coordinates, multi-valued source coordinates
	Color representation		Drawing coordinate system: 8 or 16 bits/pixel Source coordinate system: 1 bit/pixel, 8 or 16 bits/pixel

Table 1-1 Summary of Q2i Functions (cont)

Item			Function/Performance
Interface	SuperH	Command/data transfer	DMA transfer (single address), CPU memory address
		YUV → RGB conversion	16-bit input, 4:2:2 (8 bits each for Y, U, V) 16-bit output (R: 5, G: 6, B: 5 bits)
		(YUV → RGB conversion	8-bit input (4 bits each for d-Y, d-U, d-V) 16-bit output (R: 5, G: 6, B: 5 bits)
		Interrupt output	Sync detection, frame detection, DMA transfer end, command error, vertical blanking, command end, command abort
		Supported SuperH	Directly connectable to 3.3 V or 5 V operation SuperH (SH-1, SH-2 or SH-3)
	Unified graphics memory	16-bit-bus-width EDO-DRAM	Minimum 4 Mbits (choice of 4 Mbits × 1, 4 Mbits × 2, 16 Mbits × 1, 16 Mbits × 2)
	Display	RGB output	18-bit output (R: 6, G: 6, B: 6 bits)
Process/package			0.6-micron CMOS/144-pin QFP
Power supply voltage/temperature range			5.0 V ±5% / 0°C to 70°C (I-specification: 5.0 V ±10% / –40°C to 85°C)

1.5 Basic Functions

1.5.1 Interface Functions

DMA Transfer Function: Data can be transferred between memory connected to the CPU bus and the graphics memory using the DMAC. DMA transfer can be used for display list, YUV data, and (YUV data transfer. High-speed single address mode DMA transfer is possible, since the Q2i controls graphics memory addresses with its internal address counter.

YUV → RGB Conversion: Data comprising 8 bits for each of Y, U, and V, in a 4:2:2 format, is converted to RGB data (R: 5 bits, G: 6 bits, B: 5 bits).

(YUV → RGB Conversion: Data comprising 4 bits for each of (Y, U, and V is converted to RGB data (R: 5 bits, G: 6 bits, B: 5 bits).

Interrupt Output Function: Interrupt output to the CPU can be generated based on the vertical retrace line interval and by command. This feature is used for display list and source data transfer to the UGM, and blinking control.

EDO-DRAM Support: The UGM capacity can be selected to suit the display system; possible memory configurations are 4 Mbits × 1, 4 Mbits × 2, 16 Mbits × 1, and 16 Mbits × 2.

PLL Function: The Q2i allows duty free designation of external clock input (CLK0).

1.5.2 Rendering Functions

Coordinate Systems: The Q2i has four 2-dimensional coordinate systems (screen coordinates, rendering coordinates, multi-valued source coordinates, and work coordinates), and one 1-dimensional coordinate system (binary source coordinates).

Screen coordinates are display control coordinates. Screen coordinate X corresponds to the horizontal dimension of the display screen, and Y to the vertical dimension, and the origin is the display screen origin. The screen coordinate positive directions are right for the X-axis and down for the Y-axis. Either 16 bits (16-bits/pixel) or 8 bits (8-bits/pixel) can be selected as the data width of one screen coordinate.

Rendering coordinates are drawing control coordinates. Rendering coordinates are shifted horizontally and vertically with respect to screen coordinates by the offset amounts specified in drawing commands. Drawing commands perform drawing operations using these coordinates. However, drawing commands that specify clipping use screen coordinates. Either 16 bits (16-bits/pixel) or 8 bits (8-bits/pixel) can be selected as the data width of one rendering coordinate.

Multi-valued source coordinates are drawing control coordinates. When a drawing command is executed, these are the source (rectangle) coordinates specified by the command. Either 16 bits (16-bits/pixel) or 8 bits (8-bits/pixel) can be selected as the data width of one multi-valued source coordinate.

Binary source coordinates are drawing control coordinates. When a drawing command is executed, these are the source data (1-dimensional) coordinates specified by the command. The data width of one binary source coordinate is 1 bit (1-bit/pixel). For one binary source, one physical address (top-left) and the horizontal width and vertical height of the binary source are specified.

Work coordinates are drawing control coordinates that correspond one-to-one with the rendering coordinates. When a drawing command is executed, these are the work coordinates specified by the command. The data width of one work coordinate is 1 bit.

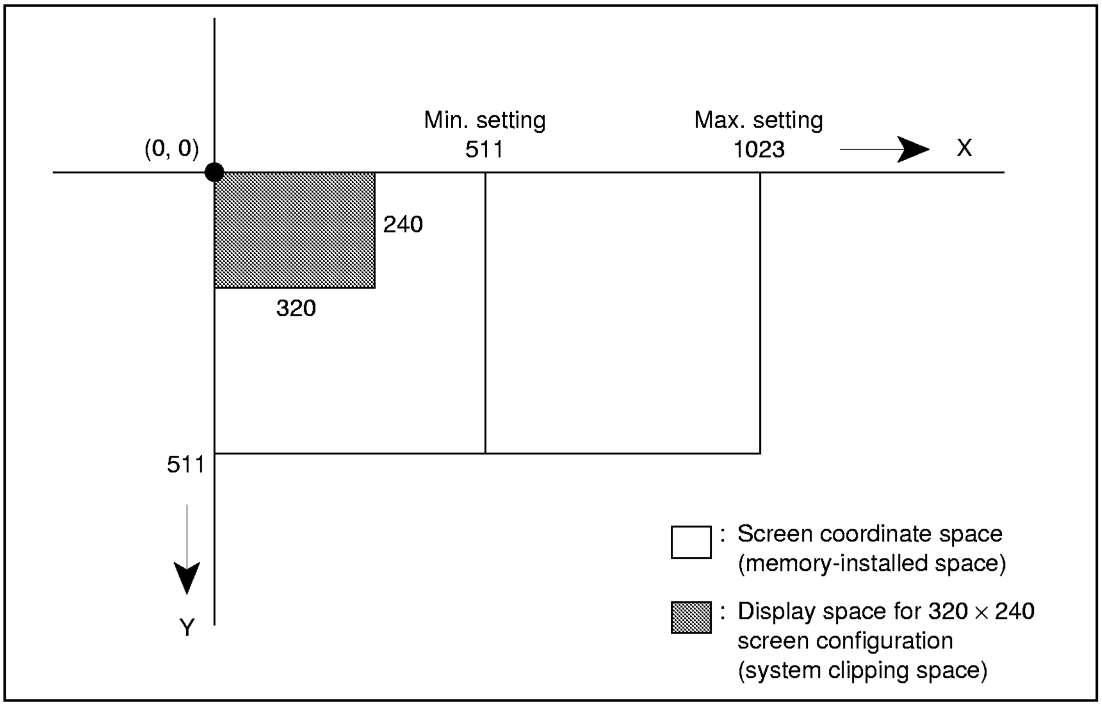
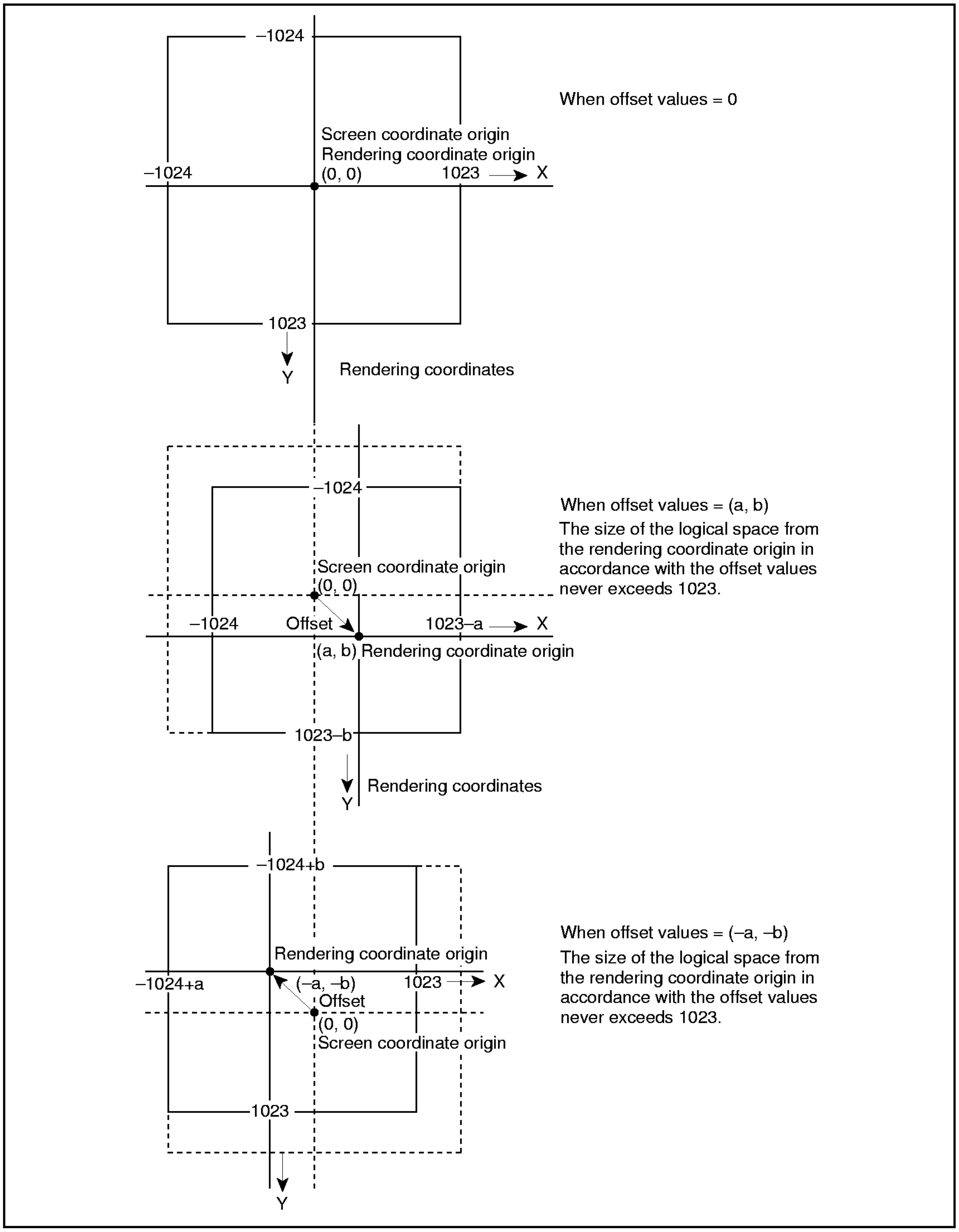

(0, 0)
Min. setting
511
Max. setting
1023
X
240
320
511
Y
□ : Screen coordinate space (memory-installed space)
▨ : Display space for 320 × 240 screen configuration (system clipping space)

Figure 1.7 Screen Coordinates



When offset values = 0

When offset values = (a, b)

The size of the logical space from the rendering coordinate origin in accordance with the offset values never exceeds 1023.

When offset values = (–a, –b)

The size of the logical space from the rendering coordinate origin in accordance with the offset values never exceeds 1023.

Figure 1-8 Rendering Coordinates

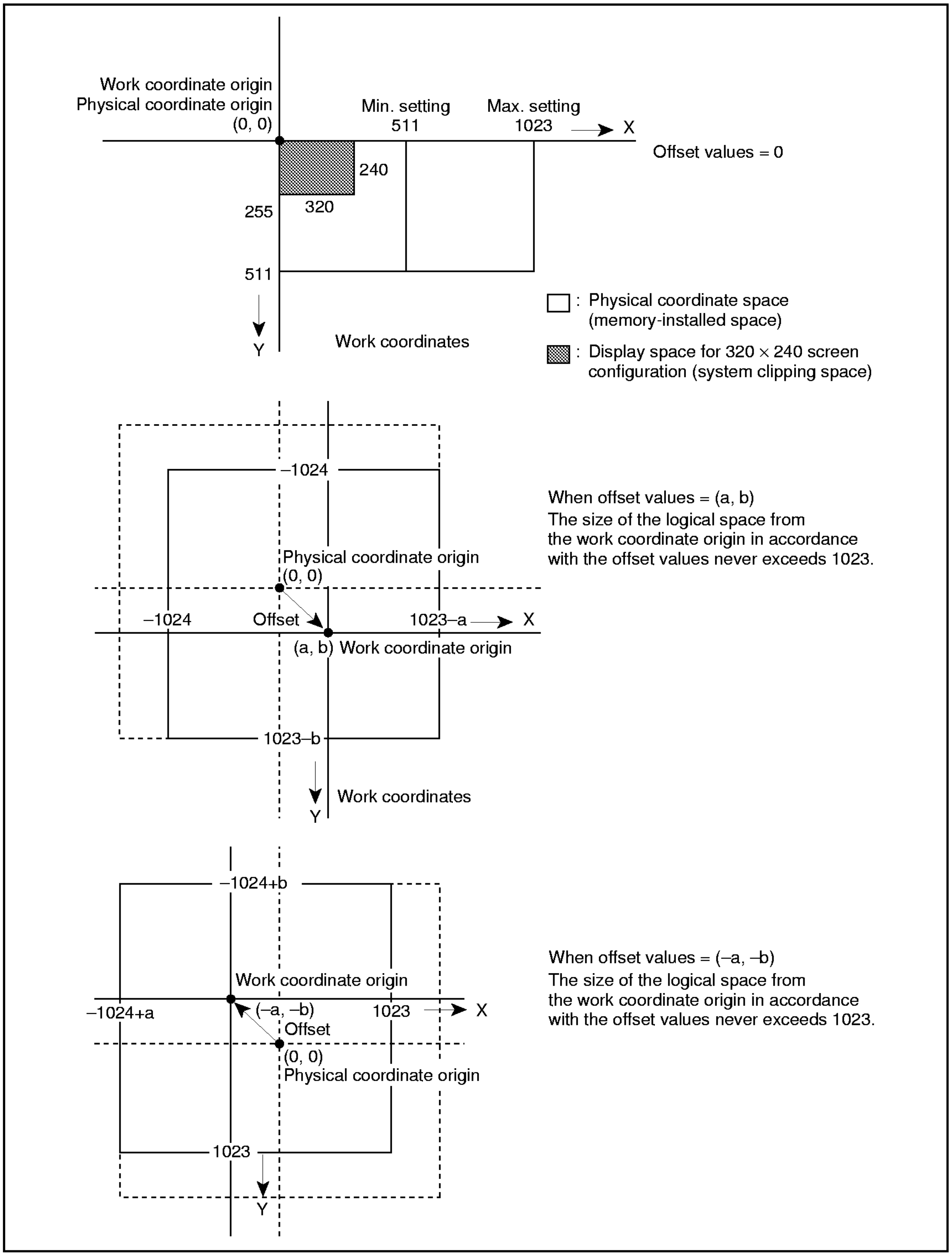

Work coordinate origin
Physical coordinate origin
(0, 0)
Min. setting
511
Max. setting
1023
X
Offset values = 0
240
320
255
511
Y
Work coordinates
□ : Physical coordinate space (memory-installed space)
▨ : Display space for 320 × 240 screen configuration (system clipping space)
–1024
Physical coordinate origin
(0, 0)
–1024
Offset
1023–a
X
(a, b) Work coordinate origin
1023–b
Work coordinates
Y
When offset values = (a, b)
The size of the logical space from the work coordinate origin in accordance with the offset values never exceeds 1023.
–1024+b
Work coordinate origin
–1024+a
(–a, –b)
Offset
1023
X
(0, 0)
Physical coordinate origin
1023
Y
When offset values = (–a, –b)
The size of the logical space from the work coordinate origin in accordance with the offset values never exceeds 1023.

Figure 1-9 Work Coordinates

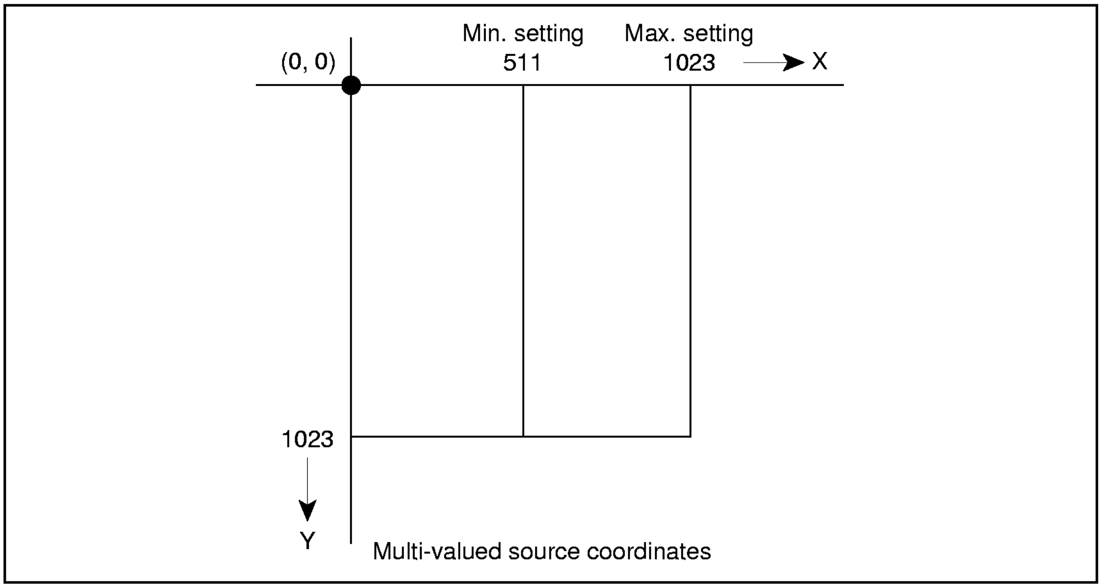


Figure 1-10 Multi-Valued Source Coordinates

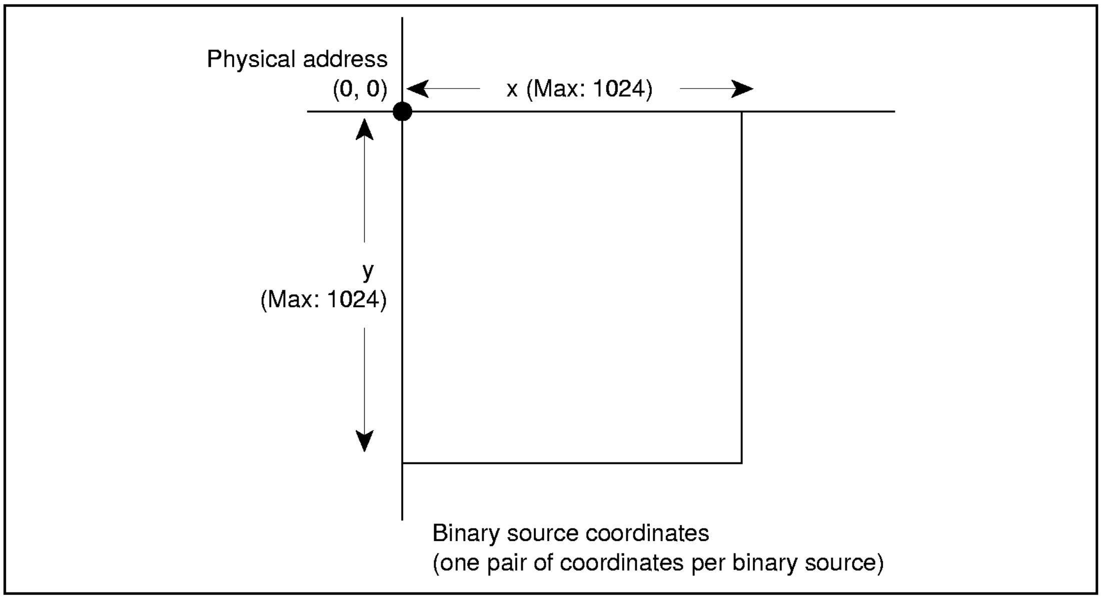

Physical address
(0, 0)
x (Max: 1024)
y
(Max: 1024)
Binary source coordinates
(one pair of coordinates per binary source)

Figure 1-11 Binary Source Coordinates
(One Pair of Coordinates Per Binary Source)

Drawing Functions: Drawing is performed at rendering coordinates or work coordinates by means of drawing commands. Whether or not source referencing is possible, and the drawing destination (rendering or work coordinates), depend on the individual drawing command.

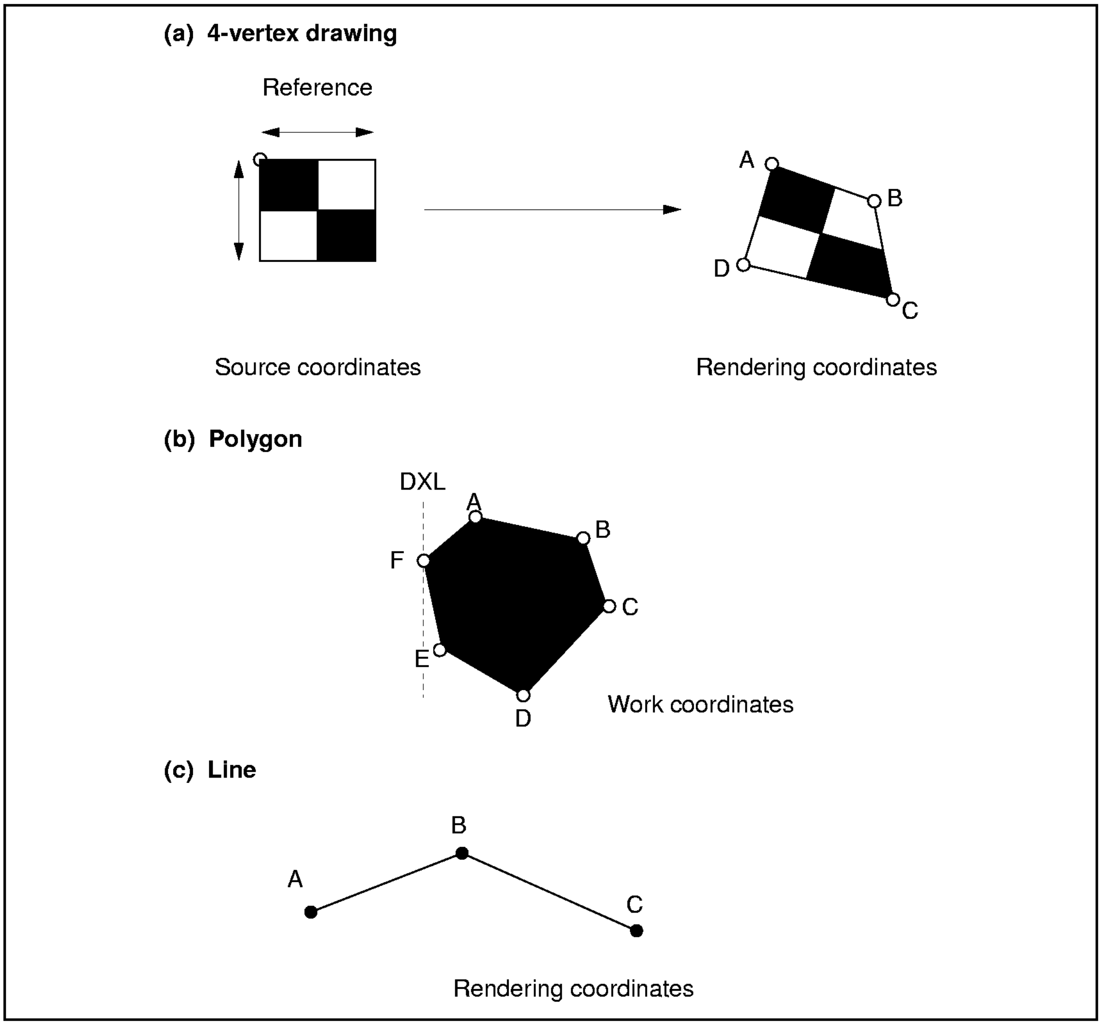


Figure 1-12 Drawing Functions

Jump: Changes the address specifying the display list (linked drawing command list) fetch destination (JUMP command).

Subroutines: Subroutines down to one nesting level can be used in display list control (GOSUB command).

Interrupts: The drawing operation is halted and an interrupt request is sent to the CPU.

Clipping: Two kinds of clipping can be specified: system clipping (SCLIP command) and user clipping (UCLIP command).

No Operation: No operation is performed. The next instruction is simply fetched, without any processing being executed (NOP3 command).

Drawing Suspension and Resumption: The Q2i supports a drawing suspend/resume function, synchronized with the VSYNC signal between the SuperH and Q2i. This function is mainly used when alternately using frame buffer and background screen to execute drawing.

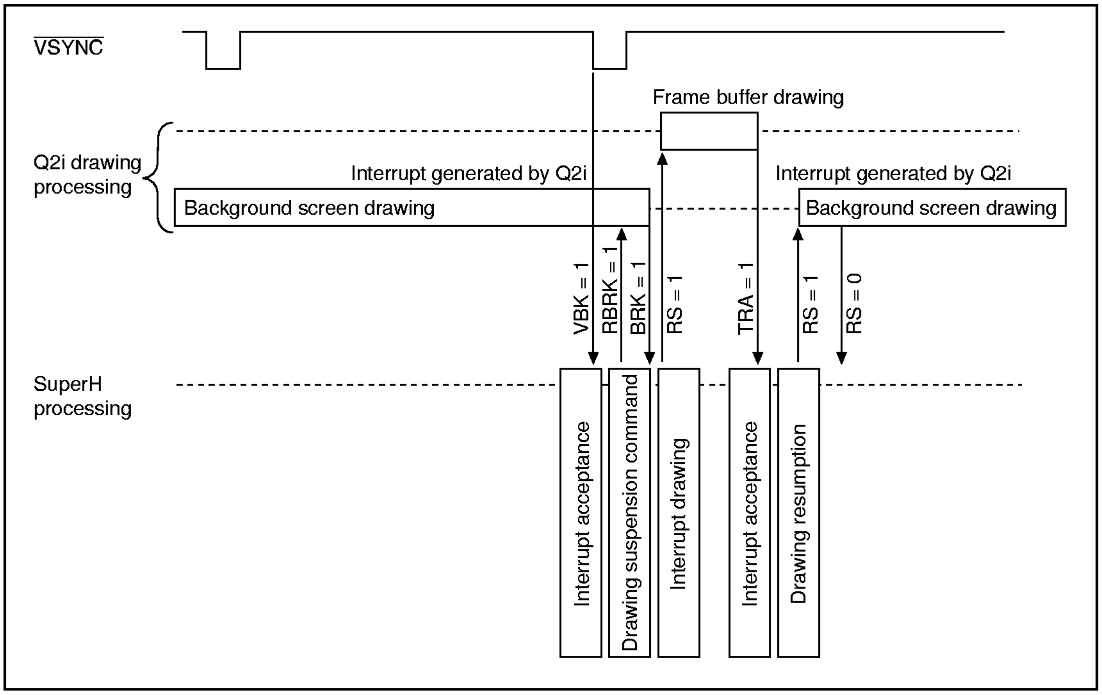


Figure 1-13 Example of Timing for Suspending and Resuming Background Screen Drawing

1.5.3 Display Functions

Display Size:
320 × 240 dots (standard size in non-interlace operation)
640 × 480 dots (standard size in interlace & video operation)

CRT Scan Modes: There are three scan modes:

- Non-interlace (vertical scan cycle examples: 1/60 sec, 1/30 sec)
- Interlace (vertical scan cycle example: 1/30 sec)
- Interlace sync & video (vertical scan cycle example: 1/30 sec)

External Synchronization mode (TV Sync Mode): In TV sync mode, the Q2i is synchronized and operated using the horizontal and vertical sync signals of a TV, video, or other external asynchronous system

In this mode, the TV, video, or other asynchronous system is treated as the master, and the Q2i as the slave. The Q2i can synchronize its display output with the external system.

Synchronization is performed every horizontal scan with the $\overline{\text{EXHSYNC}}$ input signal, and after every vertical scan with the $\overline{\text{EXVSYNC}}$ input signal (see section 3.3.4, Display Functions).

Built-In Color Palettes: The Q2i has a built-in color palette that enables simultaneous display of 256 colors out of 262,144. The palette is mapped onto the Q2i's register space, with 6 bits used for each of R, G, and B. The color palette is only valid in 8-bit/pixel mode.

Section 2 Pins

2.1 Pin Arrangement and Functions

2.1.1 Overview of Pins

Figure 2 shows an overview of the Q2i's pins.

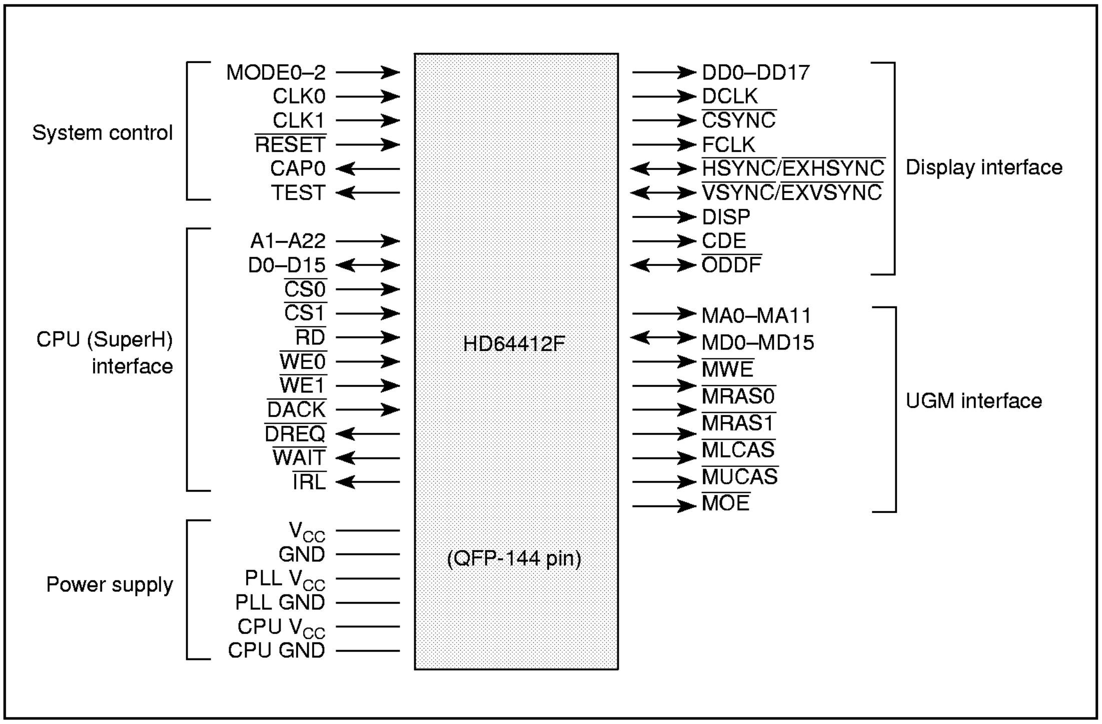


Figure 2-1 Overview of Q2i Pins

2.1.2 Pin Arrangement

Figure 2-2 shows the pin arrangement of the Q2i.

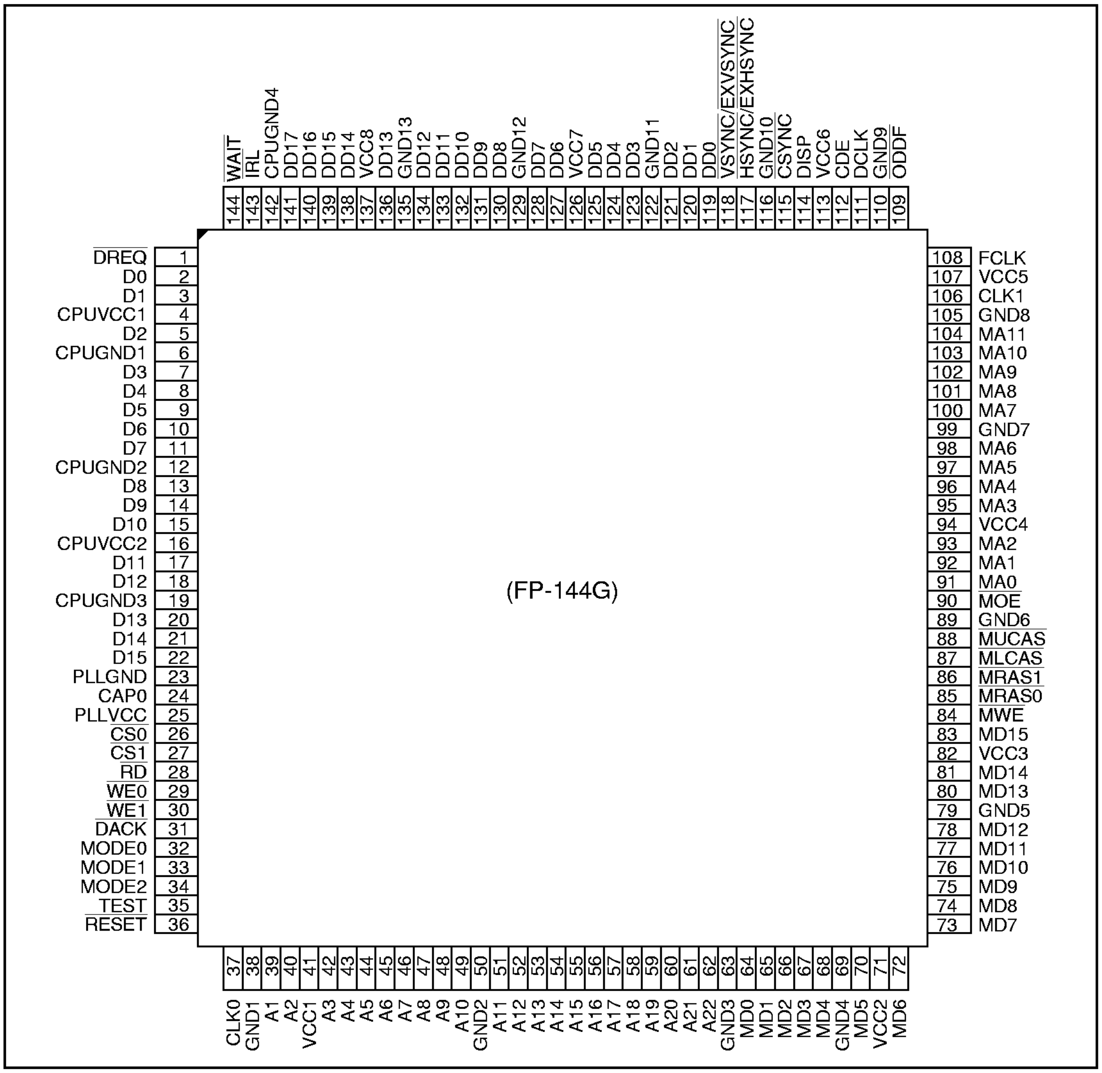


Figure 2-2 Pin Arrangement

2.1.3 Pin Functions

Table 2-1 summarizes the functions of the Q2i's pins.

Table 2-1 Pin Functions

Type	Symbol	Pin No.	I/O	Function	Notes
System control	MODE0	32	Input	Operating mode pin 0	5V input specification
	MODE1	33	Input	Operating mode pin 1	5V input specification
	MODE2	34	Input	Operating mode pin 2	5V input specification
	CLK0	37	Input	Q2i operating clock	5V input specification
	CLK1	106	Input	Display dot clock	5V input specification
	$\overline{\text{RESET}}$	36	Input	Reset	5V input specification
	CAP0	24	Output	Multiplication circuit external capacitance pin	
	TEST	35	Output	Test pin (leave open)	
CPU interface	A1	39	Input	CPU address 1	3V/5V-CPU I/F
	A2	40	Input	CPU address 2	3V/5V-CPU I/F
	A3	42	Input	CPU address 3	3V/5V-CPU I/F
	A4	43	Input	CPU address 4	3V/5V-CPU I/F
	A5	44	Input	CPU address 5	3V/5V-CPU I/F
	A6	45	Input	CPU address 6	3V/5V-CPU I/F
	A7	46	Input	CPU address 7	3V/5V-CPU I/F
	A8	47	Input	CPU address 8	3V/5V-CPU I/F
	A9	48	Input	CPU address 9	3V/5V-CPU I/F
	A10	49	Input	CPU address 10	3V/5V-CPU I/F
	A11	51	Input	CPU address 11	3V/5V-CPU I/F
	A12	52	Input	CPU address 12	3V/5V-CPU I/F
	A13	53	Input	CPU address 13	3V/5V-CPU I/F
	A14	54	Input	CPU address 14	3V/5V-CPU I/F
	A15	55	Input	CPU address 15	3V/5V-CPU I/F
	A16	56	Input	CPU address 16	3V/5V-CPU I/F
	A17	57	Input	CPU address 17	3V/5V-CPU I/F
	A18	58	Input	CPU address 18	3V/5V-CPU I/F
	A19	59	Input	CPU address 19	3V/5V-CPU I/F

Table 2-1 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Function	Notes
CPU interface	A20	60	Input	CPU address 20	3V/5V-CPU I/F
	A21	61	Input	CPU address 21	3V/5V-CPU I/F
	A22	62	Input	CPU address 22	3V/5V-CPU I/F
	D0	2	Input/output	CPU data 0	3V/5V-CPU I/F
	D1	3	Input/output	CPU data 1	3V/5V-CPU I/F
	D2	5	Input/output	CPU data 2	3V/5V-CPU I/F
	D3	7	Input/output	CPU data 3	3V/5V-CPU I/F
	D4	8	Input/output	CPU data 4	3V/5V-CPU I/F
	D5	9	Input/output	CPU data 5	3V/5V-CPU I/F
	D6	10	Input/output	CPU data 6	3V/5V-CPU I/F
	D7	11	Input/output	CPU data 7	3V/5V-CPU I/F
	D8	13	Input/output	CPU data 8	3V/5V-CPU I/F
	D9	14	Input/output	CPU data 9	3V/5V-CPU I/F
	D10	15	Input/output	CPU data 10	3V/5V-CPU I/F
	D11	17	Input/output	CPU data 11	3V/5V-CPU I/F
	D12	18	Input/output	CPU data 12	3V/5V-CPU I/F
	D13	20	Input/output	CPU data 13	3V/5V-CPU I/F
	D14	21	Input/output	CPU data 14	3V/5V-CPU I/F
	D15	22	Input/output	CPU data 15	3V/5V-CPU I/F
	$\overline{\text{CS0}}$	26	Input	Chip select 0 (UGM)	3V/5V-CPU I/F
	$\overline{\text{CS1}}$	27	Input	Chip select 1 (internal registers)	3V/5V-CPU I/F
	$\overline{\text{RD}}$	28	Input	Read strobe	3V/5V-CPU I/F
	$\overline{\text{WE0}}$	29	Input	Write pulse 0	3V/5V-CPU I/F
	$\overline{\text{WE1}}$	30	Input	Write pulse 1	3V/5V-CPU I/F
	$\overline{\text{DACK}}$	31	Input	DMA acknowledge	3V/5V-CPU I/F
	$\overline{\text{DREQ}}$	1	Output	DMA request	3V/5V-CPU I/F
	$\overline{\text{WAIT}}$	144	Output	CPU wait	3V/5V-CPU I/F
	$\overline{\text{IRL}}$	143	Output	Interrupt request	3V/5V-CPU I/F
Display interface	DD0	119	Output	Display data output 0	5V output specification
	DD1	120	Output	Display data output 1	5V output specification

Table 2-1 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Function	Notes
Display interface	DD2	121	Output	Display data output 2	5V output specification
	DD3	123	Output	Display data output 3	5V output specification
	DD4	124	Output	Display data output 4	5V output specification
	DD5	125	Output	Display data output 5	5V output specification
	DD6	127	Output	Display data output 6	5V output specification
	DD7	128	Output	Display data output 7	5V output specification
	DD8	130	Output	Display data output 8	5V output specification
	DD9	131	Output	Display data output 9	5V output specification
	DD10	132	Output	Display data output 10	5V output specification
	DD11	133	Output	Display data output 11	5V output specification
	DD12	134	Output	Display data output 12	5V output specification
	DD13	136	Output	Display data output 13	5V output specification
	DD14	138	Output	Display data output 14	5V output specification
	DD15	139	Output	Display data output 15	5V output specification
	DD16	140	Output	Display data output 16	5V output specification
	DD17	141	Output	Display data output 17	5V output specification
	DCLK	111	Output	Display clock output	5V output specification
	$\overline{\text{CSYNC}}$	115	Output	Composite sync signal output	5V output specification
	FCLK	108	Output	1/2 display dot clock	5V output specification
	$\overline{\text{HSYNC}}$/ $\overline{\text{EXHSYNC}}$	117	Input/output	Horizontal sync output/ external horizontal sync input	5V input/output specification
	$\overline{\text{VSYNC}}$/ $\overline{\text{EXVSYNC}}$	118	Input/output	Vertical sync output/ external vertical sync input	5V input/output specification
	DISP	114	Output	Signal indicating display synchronization (display sync high level)	5V output specification
	CDE	112	Output	Color detection (high in case of DD pin specific color output)	5V output specification
	$\overline{\text{ODDF}}$	109	Input/output	Signal indicating odd field (low when odd)	5V input/output specification

Table 2-1 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Function	Notes
UGM interface	MA0	91	Output	Memory address 0	5V output specification
	MA1	92	Output	Memory address 1	5V output specification
	MA2	93	Output	Memory address 2	5V output specification
	MA3	95	Output	Memory address 3	5V output specification
	MA4	96	Output	Memory address 4	5V output specification
	MA5	97	Output	Memory address 5	5V output specification
	MA6	98	Output	Memory address 6	5V output specification
	MA7	100	Output	Memory address 7	5V output specification
	MA8	101	Output	Memory address 8	5V output specification
	MA9	102	Output	Memory address 9	5V output specification
	MA10	103	Output	Memory address 10	5V output specification
	MA11	104	Output	Memory address 11	5V output specification
	MD0	64	Input/output	Memory data 0	5V input/output specification
	MD1	65	Input/output	Memory data 1	5V input/output specification
	MD2	66	Input/output	Memory data 2	5V input/output specification
	MD3	67	Input/output	Memory data 3	5V input/output specification
	MD4	68	Input/output	Memory data 4	5V input/output specification
	MD5	70	Input/output	Memory data 5	5V input/output specification
	MD6	72	Input/output	Memory data 6	5V input/output specification
	MD7	73	Input/output	Memory data 7	5V input/output specification
	MD8	74	Input/output	Memory data 8	5V input/output specification
	MD9	75	Input/output	Memory data 9	5V input/output specification
	MD10	76	Input/output	Memory data 10	5V input/output specification

Table 2-1 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Function	Notes
UGM interface	MD11	77	Input/output	Memory data 11	5V input/output specification
	MD12	78	Input/output	Memory data 12	5V input/output specification
	MD13	80	Input/output	Memory data 13	5V input/output specification
	MD14	81	Input/output	Memory data 14	5V input/output specification
	MD15	83	Input/output	Memory data 15	5V input/output specification
	$\overline{\text{MWE}}$	84	Output	Memory write pulse	5V output specification
	$\overline{\text{MRAS0}}$	85	Output	Row select signal 0	5V output specification
	$\overline{\text{MRAS1}}$	86	Output	Row select signal 1	5V output specification
	$\overline{\text{MLCAS}}$	87	Output	Lower column select signal	5V output specification
	$\overline{\text{MUCAS}}$	88	Output	Upper column select signal	5V output specification
	$\overline{\text{MOE}}$	90	Output	Memory read pulse	5V output specification
Power supply	VCC1	41	Power supply	Buffer/internal VDD	5V input specification
	VCC2	71	Power supply	Buffer/internal VDD	5V input specification
	VCC3	82	Power supply	Buffer/internal VDD	5V input specification
	VCC4	94	Power supply	Buffer/internal VDD	5V input specification
	VCC5	107	Power supply	Buffer/internal VDD	5V input specification
	VCC6	113	Power supply	Buffer/internal VDD	5V input specification
	VCC7	126	Power supply	Buffer/internal VDD	5V input specification
	VCC8	137	Power supply	Buffer/internal VDD	5V input specification
	GND1	38	Ground	Buffer VSS	
	GND3	63	Ground	Buffer VSS	

Table 2-1 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Function	Notes
Power supply	GND5	79	Ground	Buffer VSS	
	GND6	89	Ground	Buffer VSS	
	GND8	105	Ground	Buffer VSS	
	GND9	110	Ground	Buffer VSS	
	GND11	122	Ground	Buffer VSS	
	GND13	135	Ground	Buffer VSS	
	GND2	50	Ground	Internal VSS	
	GND4	69	Ground	Internal VSS	
	GND7	99	Ground	Internal VSS	
	GND10	116	Ground	Internal VSS	
	GND12	129	Ground	Internal VSS	
	PLL VCC	25	Power supply	Multiplication circuit VDD	5V input specification
	PLL GND	23	Ground	Multiplication circuit VSS	
	CPU VCC1	4	Power supply	CPU IO unit buffer VDD	3V/5V input specification
	CPU VCC2	16	Power supply	CPU IO unit buffer VDD	3V/5V input specification
	CPU GND1	6	Ground	Buffer VSS	
	CPU GND3	19	Ground	Buffer VSS	
	CPU GND2	12	Ground	Internal VSS	
	CPU GND4	142	Ground	Internal VSS	

2.2 Operating Mode Pins

These pins determine the Q2i's operating mode. The mode is fixed in a reset-startup.

1. MODE2 = L, MODE1 = L, MODE0 = L
   Normal operation state. Multiplication on. The external input clock is duty-free.
   The internal operating clock has the same frequency as the external input clock.
2. MODE2 = L, MODE1 = L, MODE0 = H
   Normal operation state. Multiplication on. The external input clock is duty-free.
   The internal operating clock has twice the frequency of the external input clock.
3. MODE2 = L, MODE1 = H, MODE0 = L
   Normal operation state. Multiplication on. The external input clock is duty-free.
   The internal operating clock has four times the frequency of the external input clock.
4. MODE2 = L, MODE1 = H, MODE0 = H
   Normal operation state. Multiplication off. The external input clock must have a 50% duty.
   An external input clock is used as the internal operating clock.
5. MODE2 = H, MODE1 = *, MODE0 = *: Setting prohibited

Notes: H: High level
L: Low level
*: Either high or low level

2.3 CPU Interface Pins

2.3.1 CPU Writes

The CPU can access the UGM or Q2i internal registers. In a UGM access, a low-level signal is input to $\overline{\text{CS0}}$; in a Q2i internal register access, a low-level signal is input to $\overline{\text{CS1}}$. $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$ should not be driven low at the same time. The UGM or Q2i internal register address is input to A22 to A1. The address is a word address. Only word (2-byte) access can be used with the Q2i. Input a low-level signal to either $\overline{\text{WE0}}$ or $\overline{\text{WE1}}$, or to both. If the CPU is an SH7040, SH7042, or SH7044 the $\overline{\text{WAIT}}$ pin is also used as A20, and therefore the UGM capacity is limited. The allocation of ROM, etc., in the memory space must be taken into consideration when determining the UGM capacity.

The Q2i uses an asynchronous interface for the CPU interface, and a delay in a CPU access due to a Q2i source is reported to the CPU by means of the $\overline{\text{WAIT}}$ signal. However, the high-level width specification for the $\overline{\text{WE0}}$ and $\overline{\text{WE1}}$ signals must be observed. See Section 6, Usage Notes, for detailed information concerning clocks. If the CPU is an SH-3, the clock output from the CKIO pin cannot be input directly to the Q2i's CLK0 pin, because the signal output from the CKIO pin is a 3.3 V TTL interface signal, while the Q2i's CLK0 pin signal is a 5 V CMOS interface signal. In this case, a signal obtained by level-shifting CKIO to 5 V should be input to the CLK0 pin.

To enable the SuperH to recognize the $\overline{\text{WAIT}}$ signal output by the Q2i, enable SuperH hardware waits and set a software wait cycle. The software wait cycle is determined by the relationship between the frequency of the clock output from the CKI0 pin and the frequency of the clock input to the CLK0 pin. Refer to the Q-Series Application Note, HD64412 Q2i Volume, for examples of software wait cycle setting. When an SH704X is used, a setting must be made to extend the $\overline{\text{CS}}$ assertion period.

Byte access to registers must not be used, since this will corrupt the accessed register and UGM data.

In some SuperH models, the initial setting of the $\overline{\text{CS}}$ pin is I/O port (input) mode. In this case, the $\overline{\text{CS}}$ pin level will temporarily be unstable in a power-on reset, and so the $\overline{\text{CS}}$ pin should be pulled up externally.

2.3.2 CPU Reads

A read operation is basically the same as a write operation. Reads are performed in word units.

2.3.3 DMA Writes

The DMA controller can perform display list, binary source, and ΔYUV data transfers using cycle stealing. To perform data transfer with the DMA controller, DMA mode settings must be made in the DMA transfer start address register (DMASR), DMA transfer word count register (DMAWR), and system control register (SYSR). After the DMA mode settings are made, the Q2i drives the $\overline{\text{DREQ}}$ signal low as soon as its preparations are completed. On receiving this signal, the DMA controller reads data from memory and places it on the data bus. The data on the data bus is then latched internally by the Q2i on the rise of the $\overline{\text{RD}}$ signal, and transferred to the UGM. When DMA writes are performed using a display list or binary source as the data, the DMA mode is set to 01. When DMA writes are performed using ΔYUV or YUV data as the data, the DMA mode is set to 11.

In DMA mode, the Q2i does not output hardware waits to the CPU.

For the DMA mode, set cycle-steal DMA mode edge detection and single address mode.

When an SH704X (SH-2) is used as the CPU, a setting must be made to enable extension of the CPU's $\overline{\text{CS}}$ assertion period.

UGM access by the CPU should not be performed if the DMA mode setting is 01 or 11.

The SuperH family includes models in which the initial DACK pin setting is active-high. In this case, leave the DACK pin at its initial setting (active-high) and use an external circuit to invert the DACK pin signal before input to the Q2i's $\overline{\text{DACK}}$ pin.

The initial setting of the DACK pin is active-high in the following SuperH models:

SH7014, SH7032, SH7034, SH7050, SH7020, SH7021, SH7040 Series

2.3.4 Interrupts

The Q2i interrupts the CPU by means of Q2i internal sources. Interrupt sources are set in the interrupt enable register (IER).

2.4 Power Supply Pins

2.4.1 Normal Power Supply and PLL Power Supply

The normal power supply and PLL power supply are connected to 5 V.

CAP0 is the external capacitance pin for the multiplication circuit. When multiplication is turned on, connect the specified capacitance to this pin. (See figure 2-3.) When multiplication is turned off, either leave the CAP0 pin open or connect the circuit shown inside the dotted lines.

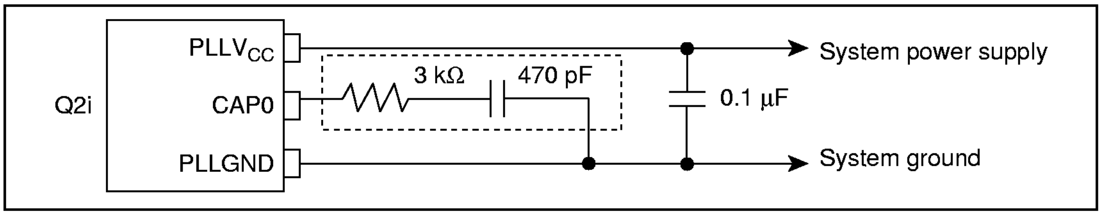


Figure 2-3 Sample CAP0 Pin Connection Circuit

2.4.2 CPU Power Supply

The Q2i can be connected to an SH-1, SH-2, or SH-3. Either 5 V or 3.3 V is connected to the CPU power supply (CPU Vcc1 and Vcc2), as appropriate. Connect 5 V for an SH703X (SH-1), SH7064 (SH-2), or SH704X (SH-2) CPU, and 3.3 V for an SH7708 (SH-3) CPU.

When a 3.3 V CPU power supply is used, the order of powering on is: normal power supply and PLL power supply (5 V) first, followed by the CPU power supply (3.3 V). The order of powering off is: CPU power supply (3.3 V) first, followed by the normal power supply and PLL power supply (5 V). (The device may be damaged if this order is not followed.)

2.5 Display Interface Pins

The signals output from the display interface pins are all synchronized with the dot clock (DCLK).

2.5.1 DAC Interface

Outputs the digital pixel data synchronized with the dot clock. The pixel data format is 6 bits each for R, G, and B. Outside the display period, DD0–DD17 all go low.

2.5.2 Video Encoder Interface

In the Q2i, a video encoder interface is implemented by setting the DOT bit to 1 in the display mode register. For example, when an NTSC encoder is used as the video encoder, 4FSC (14.31818 MHz) should be input to the CLK1 pin. As a result, the dot clock (7.15909 MHz) will be output to the DCLK pin, the digital composite sync signal to the $\overline{\text{CSYNC}}$ pin, and FSC (subcarrier frequency: 3.58 MHz) to the FCLK pin. In TV sync mode, $\overline{\text{CSYNC}}$ output is high.

The clocks output from DCLK and FCLK are synchronized. Therefore, if the clock output from FCLK is used as the subcarrier frequency, color drift may occur due to cross-colors, etc. The video encoder circuit should therefore include provisions to prevent color drift and to make $\overline{\text{HSYNC}}$ an odd multiple of DCLK.

2.5.3 CRT Interface

Outputs the horizontal sync signal and vertical sync signal, the DISP signal indicating display synchronization, the CDE signal for color detection, and the $\overline{\text{ODDF}}$ signal that indicates whether the current field is even or odd for interlace control. When synchronization is coordinated with an external device (TV or video recorder), the horizontal sync, vertical sync, and $\overline{\text{ODDF}}$ signals are input. In a reset, the $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{ODDF}}$ pins go to input mode, and therefore these pins must be fixed in a non-significant direction (pulled up).

2.6 UGM Interface Pins

2.6.1 UGM Access

The Q2i allows EDO page mode DRAM to be used as the UGM. EDO page mode DRAMs that can be used with the Q2i are the Hitachi HM51 (S) 4265 Series (4-Mbit capacity, 5 V supply voltage, 256 k × 16 memory configuration), the Hitachi HM5118165 Series (16-Mbit capacity, 5 V supply voltage, 1 M × 16 memory configuration), or an equivalent product. Two of these DRAMs can be connected to the Q2i. Basically, memory with an access time of 60 ns or less should be used.

Section 3 Unified Graphics Memory (UGM) Display Functions

3.1 Clocks

There are two Q2i clocks, CLK0 and CLK1. The clock used as the base for the operating clock is input at the CLK0 pin, and the clock used as the base for the display dot clock (DCLK) is input at the CLK1 pin.

The operating clock is the base clock for performing drawing operations, and is also used as the base clock for UGM access. The Q2i includes an operating clock multiplication circuit that enables a ×1, ×1/2, or ×1/4 multiple of the operating clock to be selected for input at the CLK0 pin.

The display dot clock is the base clock for display operations, and is used to control display data output and generate horizontal and vertical sync signals. The Q2i has a display dot clock divider that enables a ×1 or ×2 multiple of the dot rate to be input at the CLK1 pin.

The relationship between the clocks and operating frequencies is summarized in table 3-1.

Table 3-1 Input Clocks and Operating Frequencies

Clock Input Pin	Clock Type	Operating Mode	
CLK0	One of the clocks on the right is the operating clock.	Multiplication on	Clock with the CLK0 frequency, and duty adjusted to 50%
			Clock with twice the CLK0 frequency, and duty adjusted to 50%
			Clock with four times the CLK0 frequency, and duty adjusted to 50%
		Multiplication off	Clock with the CLK0 frequency
CLK1	One of the clocks on the right is the display dot clock.	Clock with the CLK1 frequency	
		Clock with 1/2 the CLK1 frequency	

The operating clock and display dot clock frequencies can be set to any values within the following range:

(1) Operating clock frequency > 2 × display dot clock frequency

(2) Operating clock frequency = 2 × display dot clock frequency and the operating clock and display dot clock are synchronized

Drawing operations can therefore be performed at maximum speed without being influenced by the characteristics of the display device.

3.2 UGM (Unified Graphics Memory)

3.2.1 Overview

The memory connected to the Q2i (graphics memory) is used for the following purposes.

1. Frame buffers

   Q2i drawing area and display area.
2. Display list (command list)

   Area that stores the Q2i drawing command list. The Q2i fetches commands from this area while carrying out drawing operations.
3. Source areas, work areas, etc.

   Used as the source area that stores painting patterns and font data, the FTRAP command drawing area, and so on.

The UGM can be allocated to part of the CPU's main memory area. Figure 3-1 shows a sample system configuration using UGM, and figure 3-2 shows an example of UGM mapping onto the CPU memory space.

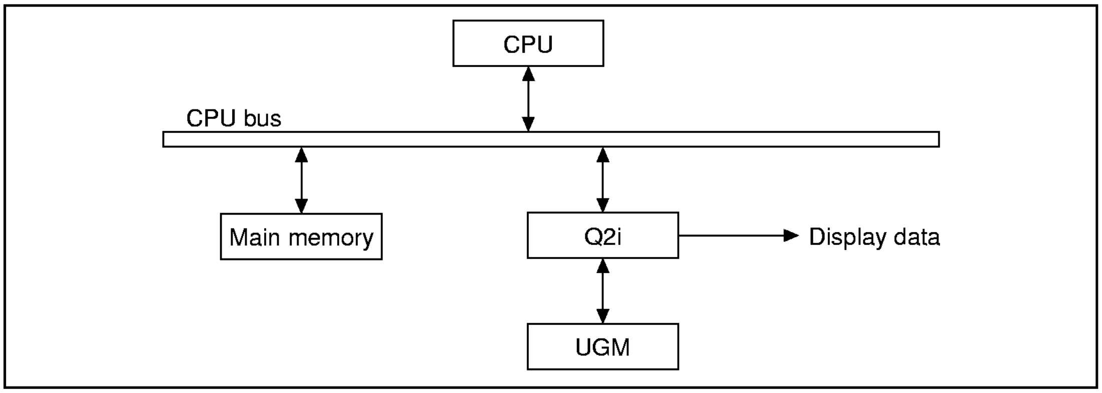


Figure 3-1 Example of System Configuration Using UGM

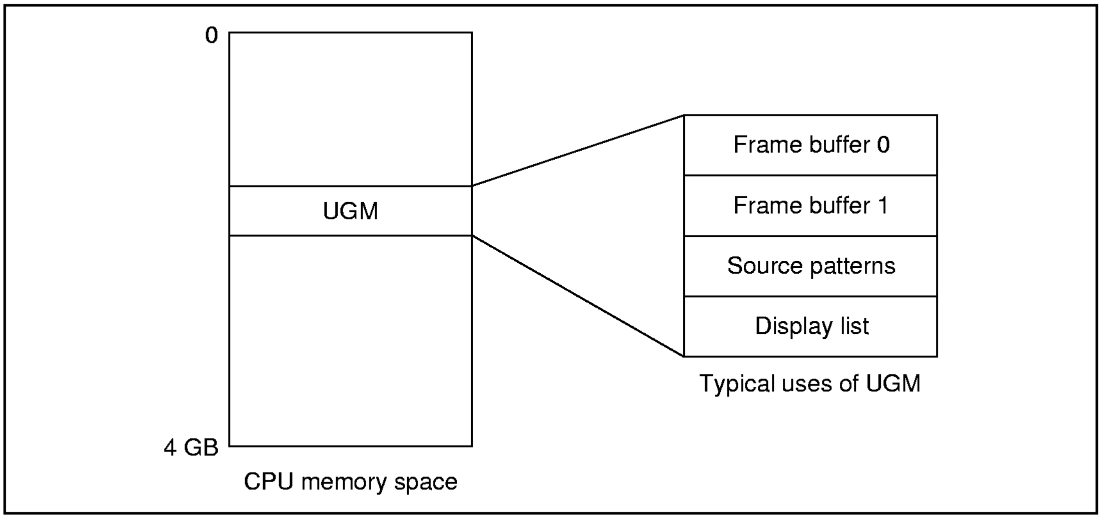


Figure 3-2 Example of UGM Mapping onto CPU Memory Space

3.2.2 Memory Access

The priority order for control of UGM access is as follows:

1. Refreshing
2. Display
3. CPU
4. Other (command fetches, drawing, source referencing, etc.)

To enable these different kinds of processing to be performed in parallel, after performing access for a fixed period, the Q2i passes the access right to another source. So if three sources are requesting access, for example, they will perform accesses alternately.

UGM Access by the CPU: The CPU can access the UGM in two ways, via CPU software or via the DMAC. When the CPU accesses the UGM, the UGM address is input directly to the Q2i's A1–A22 pins and the $\overline{\text{CS0}}$ pin is driven low. Therefore, a UGM address in the range specified by the memory mode register should be input to the Q2i's A1–A22 pins. For example, when using one 4-Mbit memory as the UGM, the Q2i's A19–A22 pins must go low when the UGM is accessed by the CPU.

Since a SuperH CPU is used, the UGM is mapped onto SuperH memory space. Data transfer between the CPU and Q2i is synchronized with the Q2i's operating clock.

For UGM access by the CPU, set initial values in the interface control register, memory control register, and display control register, and then start display synchronization operation before performing the access. If this is not done, the Q2i will output waits continuously when the CPU accesses the UGM.

- Access by software

  In access by software, the UGM is accessed as part of the main memory.

  In a write operation, no-wait access is possible if there is empty space in the Q2i's built-in 32-byte FIFO buffer.

  In a read operation, a number of wait cycles are inserted. The number of wait cycles varies greatly depending on the relationship between the operating clock and the display dot clock, and the screen size. For example, with a 33 MHz operating clock, a 7 MHz display dot clock, and a 320 × 240 (8 bits/pixel) screen size, the average number of wait cycles will be around 23.

- Access by DMA

  With a CPU that has a built-in DMAC (such as the SH-2 or SH704X), data in the memory connected to the CPU can be transferred to the UGM using the DMAC. DMA transfer can be used to transfer display list or YUV data.

  Single address mode can be used in DMA transfers, since graphics memory addresses are controlled by the Q2i's built-in address counter. However, only cycle-steal mode can be used as the bus mode.

UGM Access by Q2i: EDO page mode DRAM can be connected to the Q2i as UGM. Use of this memory enables the Q2i to perform memory access in one-cycle (operating clock) units.

The memory configuration consists of one or two 256-kword × 16-bit (4-Mbit) DRAMs, and one or two 1-Mword × 16-bit (16-Mbit) DRAMs.

With regard to row address and column address multiplex control, it is possible to use products with a 9, 10, 11, or 12-bit row address.

The type of memory is set in the memory mode register (MEMR)

3.2.3 Memory Map

The Q2i performs UGM address control. The UGM includes the display list area, binary source area, work area, 8-bit/pixel source and 16-bit/pixel source areas, and 8-bit/pixel rendering and 16-bit/pixel rendering areas. The UGM is configured in 512-byte units, and a different memory configuration is used for each area. The memory configuration for each of the areas is shown in figure 3-3.

Area settings are made according to the respective start addresses (see section 5.4, Memory Control Registers).

- 1-bit/pixel (work, binary source, display list)

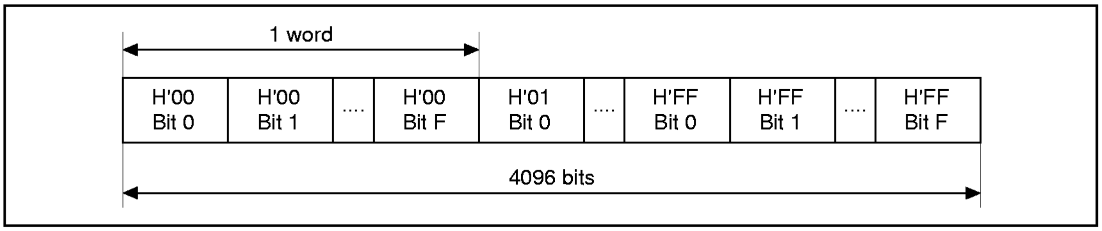


Figure 3-3 Configuration of One Memory Unit (512 Bytes) (1)

- 8 bits/pixel (multi-valued source, multi-valued destination)

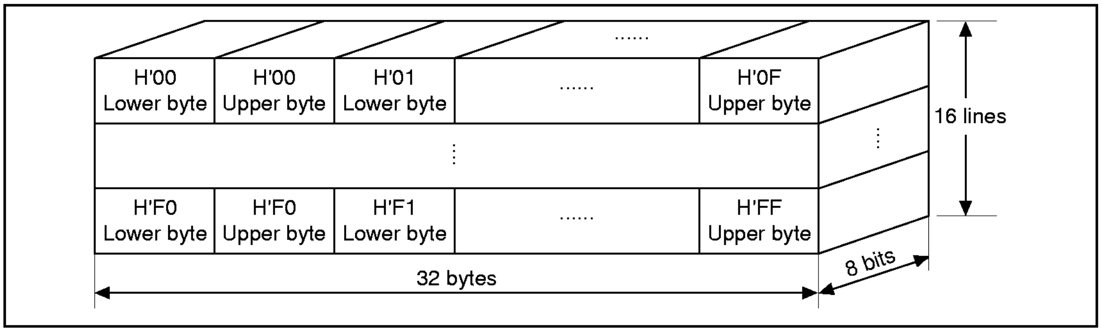


Figure 3-3 Configuration of One Memory Unit (512 Bytes) (2)

- 16 bits/pixel (multi-valued source, multi-valued destination)

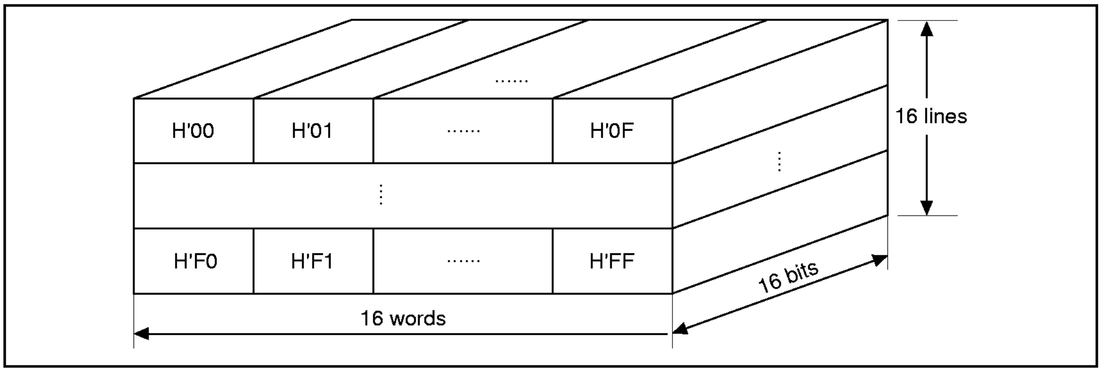
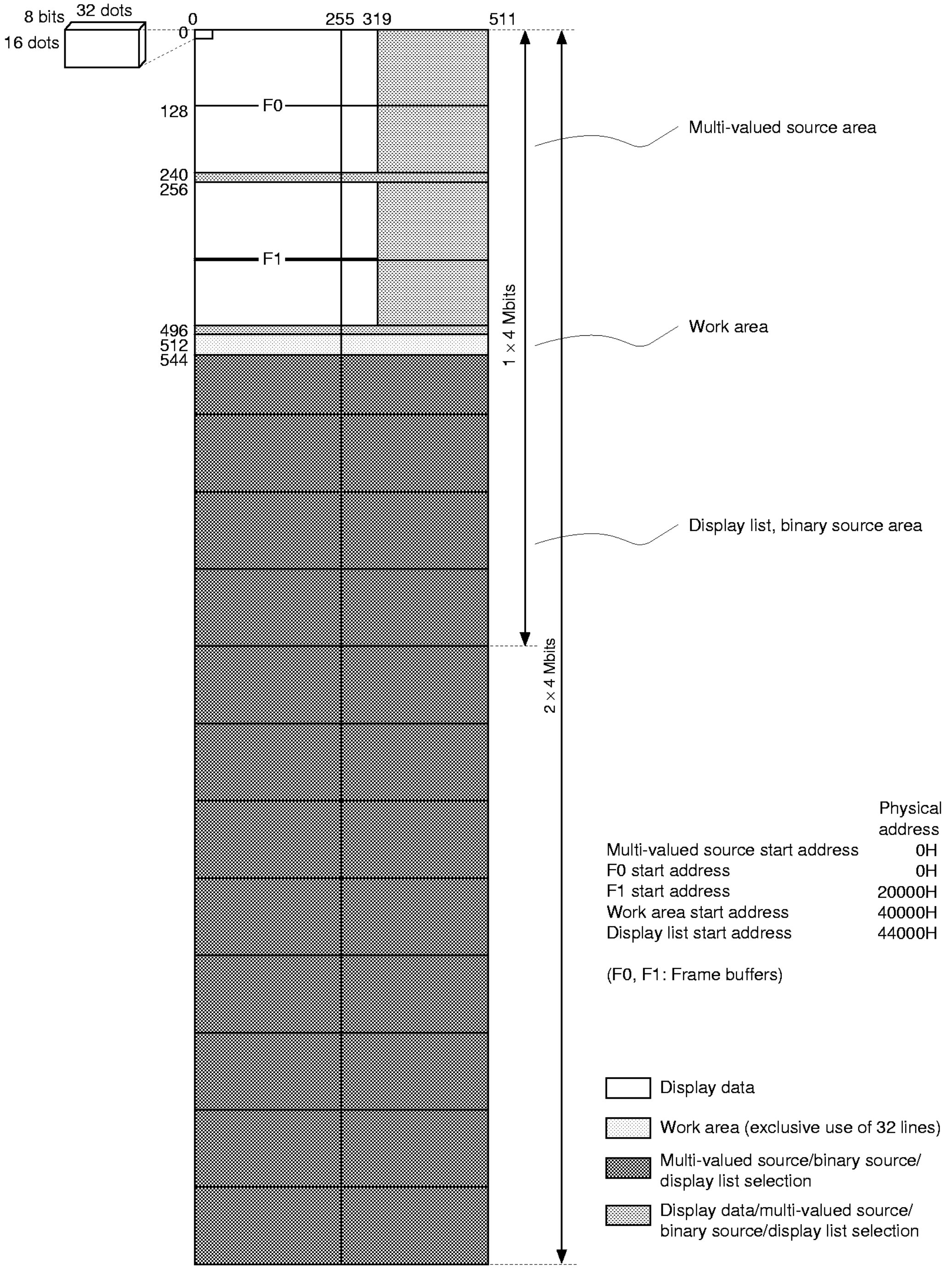


Figure 3-3 Configuration of One Memory Unit (512 Bytes) (3)

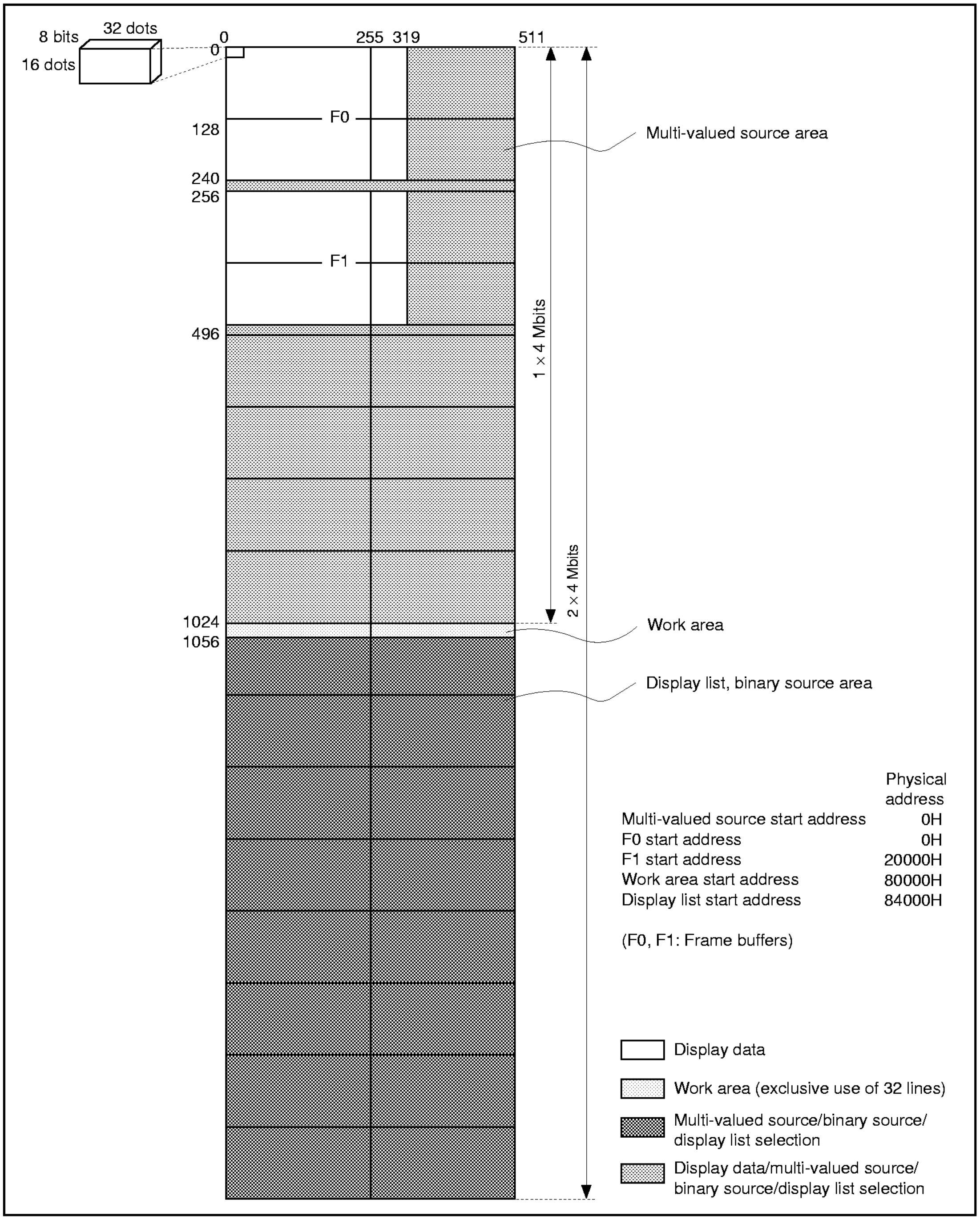
Figures 3-4 to 3-12 show sample UGM memory maps.


8 bits
32 dots
16 dots
0
255
319
511
0
128
F0
240
256
F1
496
512
544
1 × 4 Mbits
2 × 4 Mbits
Multi-valued source area
Work area
Display list, binary source area
☐ Display data
▨ Work area (exclusive use of 32 lines)
▨ Multi-valued source/binary source/ display list selection
▨ Display data/multi-valued source/ binary source/display list selection

	Physical address
Multi-valued source start address	0H
F0 start address	0H
F1 start address	20000H
Work area start address	40000H
Display list start address	44000H

(F0, F1: Frame buffers)

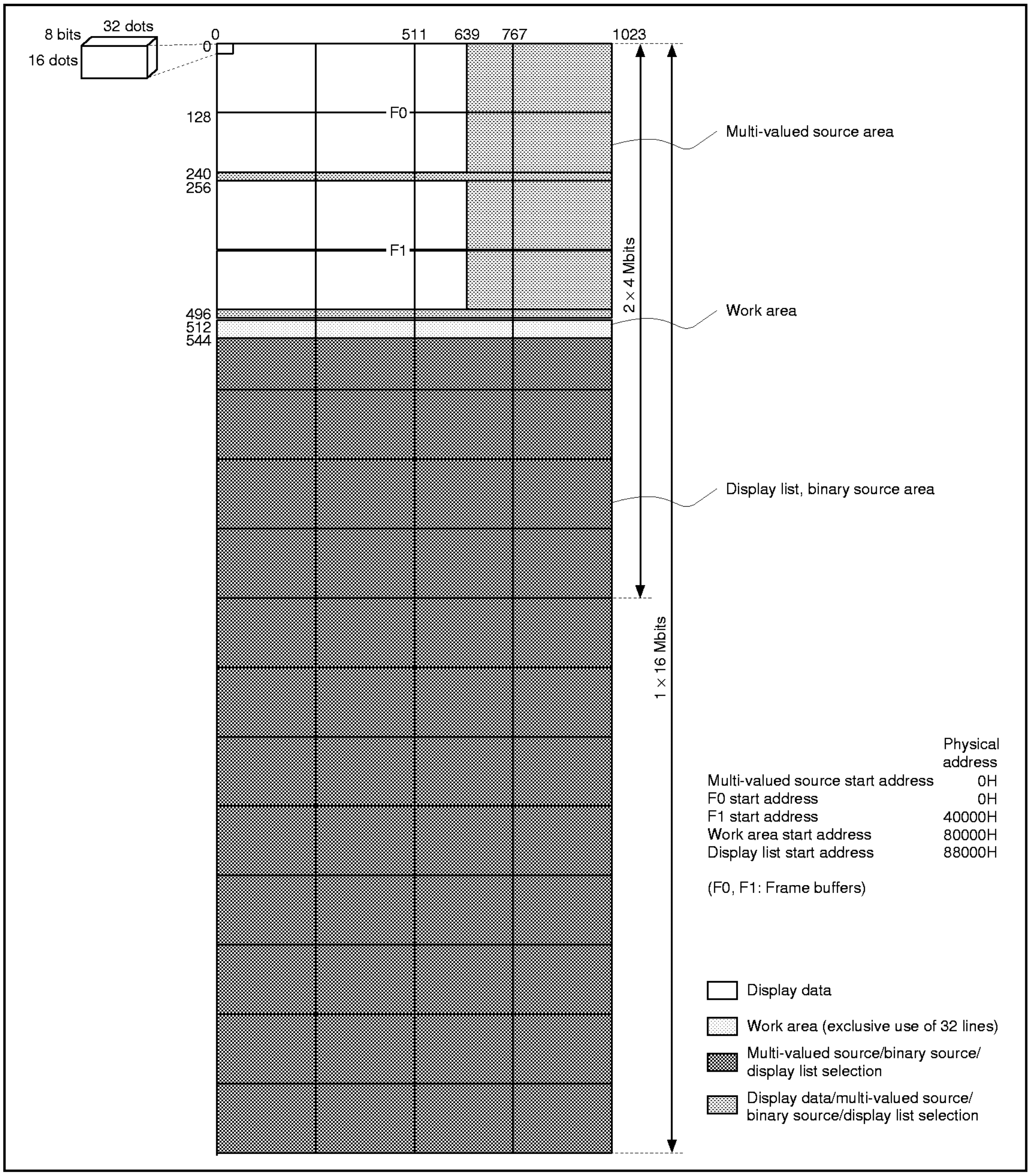
Figure 3-4 Memory Map Example 1
[Screen Size at 8 Bits/Pixel (320 × 240 Equivalent)]


8 bits
32 dots
16 dots
0
255
319
511
0
128
F0
240
256
F1
496
1024
1056
1 × 4 Mbits
2 × 4 Mbits
Multi-valued source area
Work area
Display list, binary source area
Display data
Work area (exclusive use of 32 lines)
Multi-valued source/binary source/ display list selection
Display data/multi-valued source/ binary source/display list selection

	Physical address
Multi-valued source start address	0H
F0 start address	0H
F1 start address	20000H
Work area start address	80000H
Display list start address	84000H

(F0, F1: Frame buffers)

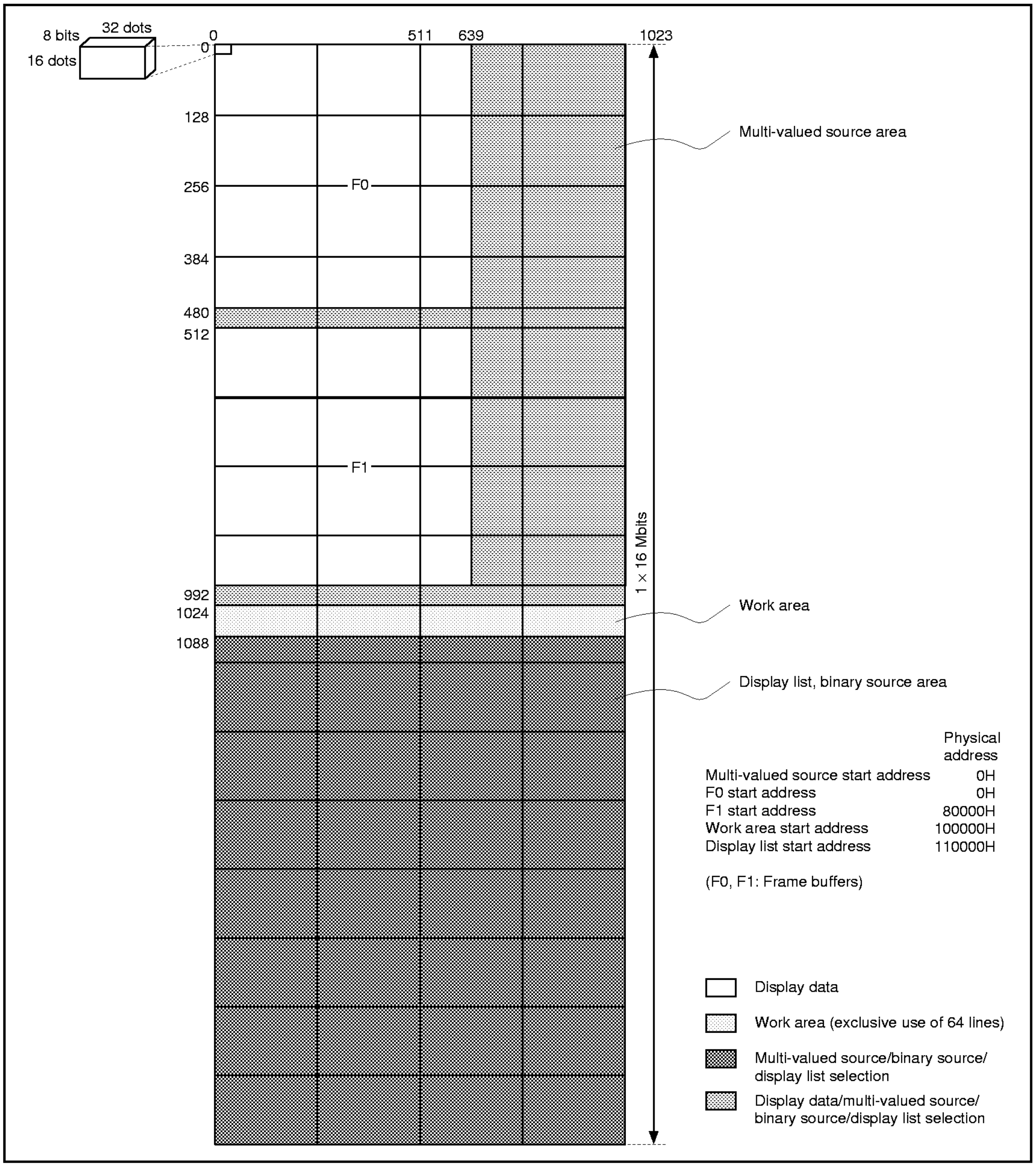
Figure 3-5 Memory Map Example 2
[Screen Size at 8 Bits/Pixel (320 × 240 Equivalent)]


8 bits
32 dots
16 dots
0
511
639
767
1023
0
128
F0
240
256
F1
496
512
544
2 × 4 Mbits
1 × 16 Mbits
Multi-valued source area
Work area
Display list, binary source area
□ Display data
▨ Work area (exclusive use of 32 lines)
▩ Multi-valued source/binary source/ display list selection
▨ Display data/multi-valued source/ binary source/display list selection

	Physical address
Multi-valued source start address	0H
F0 start address	0H
F1 start address	40000H
Work area start address	80000H
Display list start address	88000H

(F0, F1: Frame buffers)

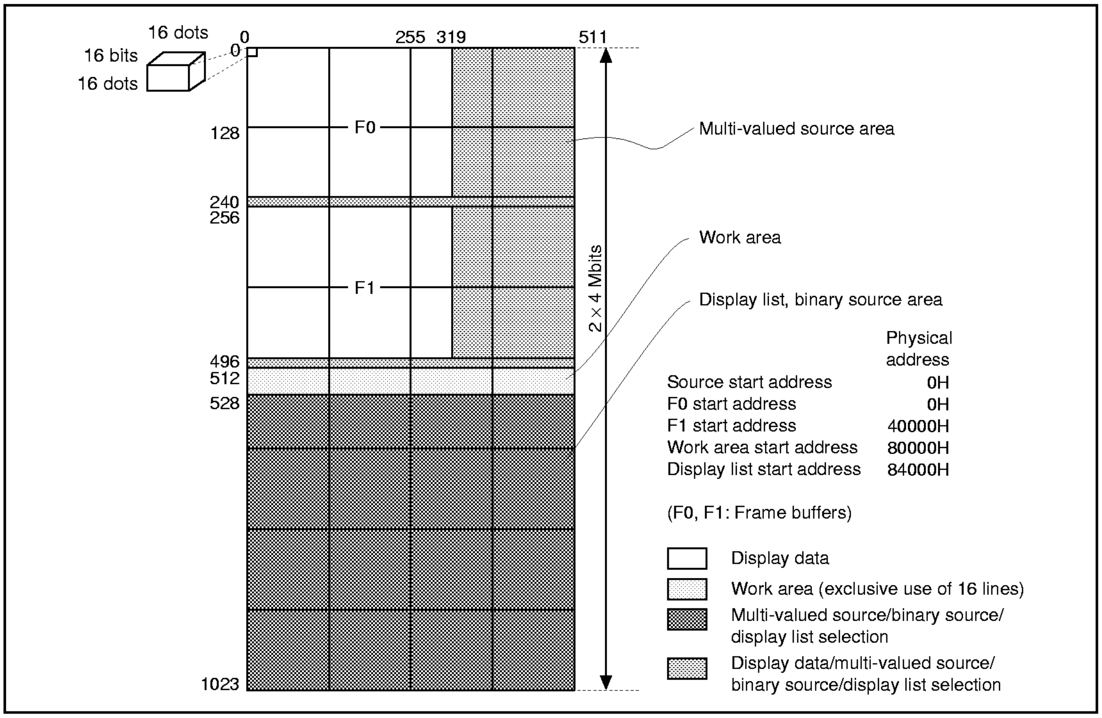
Figure 3-6 Memory Map Example 3
[Screen Size at 8 Bits/Pixel (640 × 240 Equivalent)]


8 bits
32 dots
16 dots
0
0
511
639
1023
128
256
384
480
512
992
1024
1088
F0
F1
1 × 16 Mbits
Multi-valued source area
Work area
Display list, binary source area
Display data
Work area (exclusive use of 64 lines)
Multi-valued source/binary source/ display list selection
Display data/multi-valued source/ binary source/display list selection

	Physical address
Multi-valued source start address	0H
F0 start address	0H
F1 start address	80000H
Work area start address	100000H
Display list start address	110000H

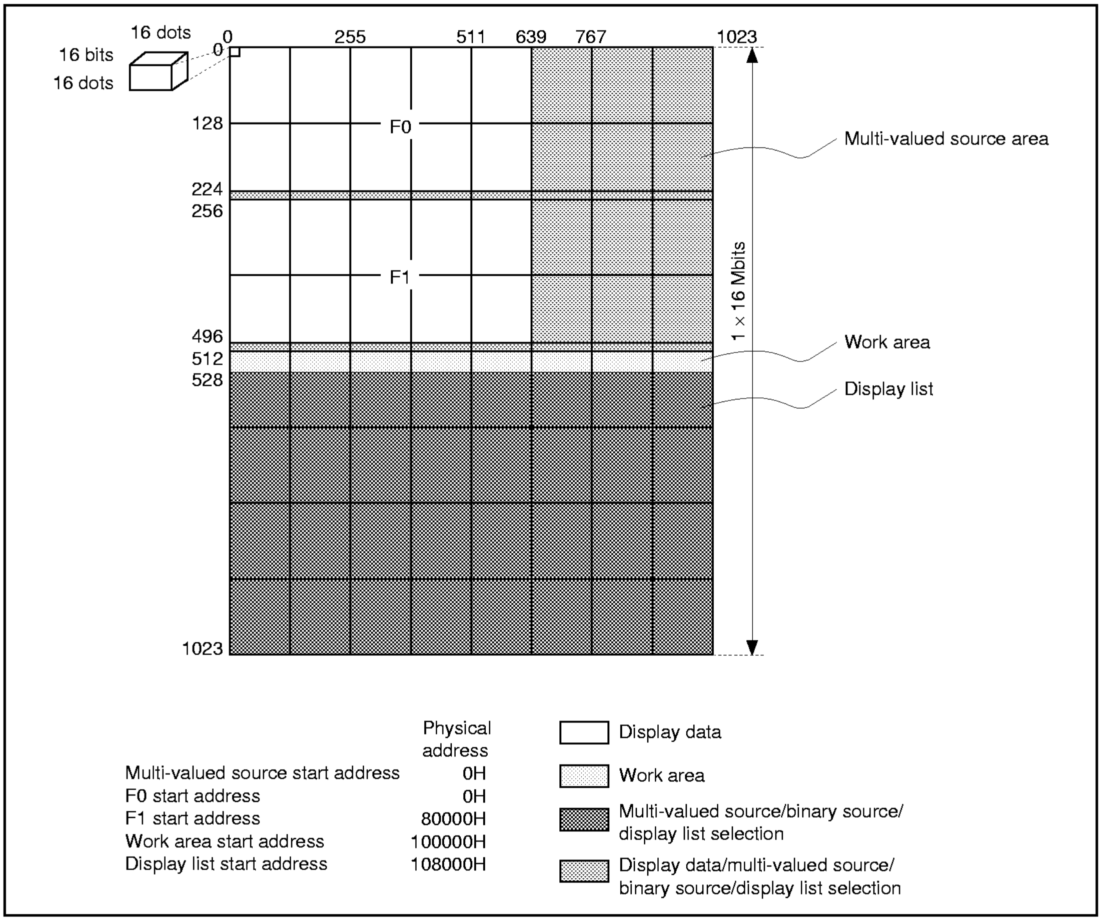
(F0, F1: Frame buffers)

Figure 3-7 Memory Map Example 4
[Screen Size at 8 Bits/Pixel (640 × 480 Equivalent)]


16 dots
16 bits
16 dots
0
0
255
319
511
128
F0
240
256
F1
496
512
528
1023
2 × 4 Mbits
Multi-valued source area
Work area
Display list, binary source area
(F0, F1: Frame buffers)
Display data
Work area (exclusive use of 16 lines)
Multi-valued source/binary source/ display list selection
Display data/multi-valued source/ binary source/display list selection

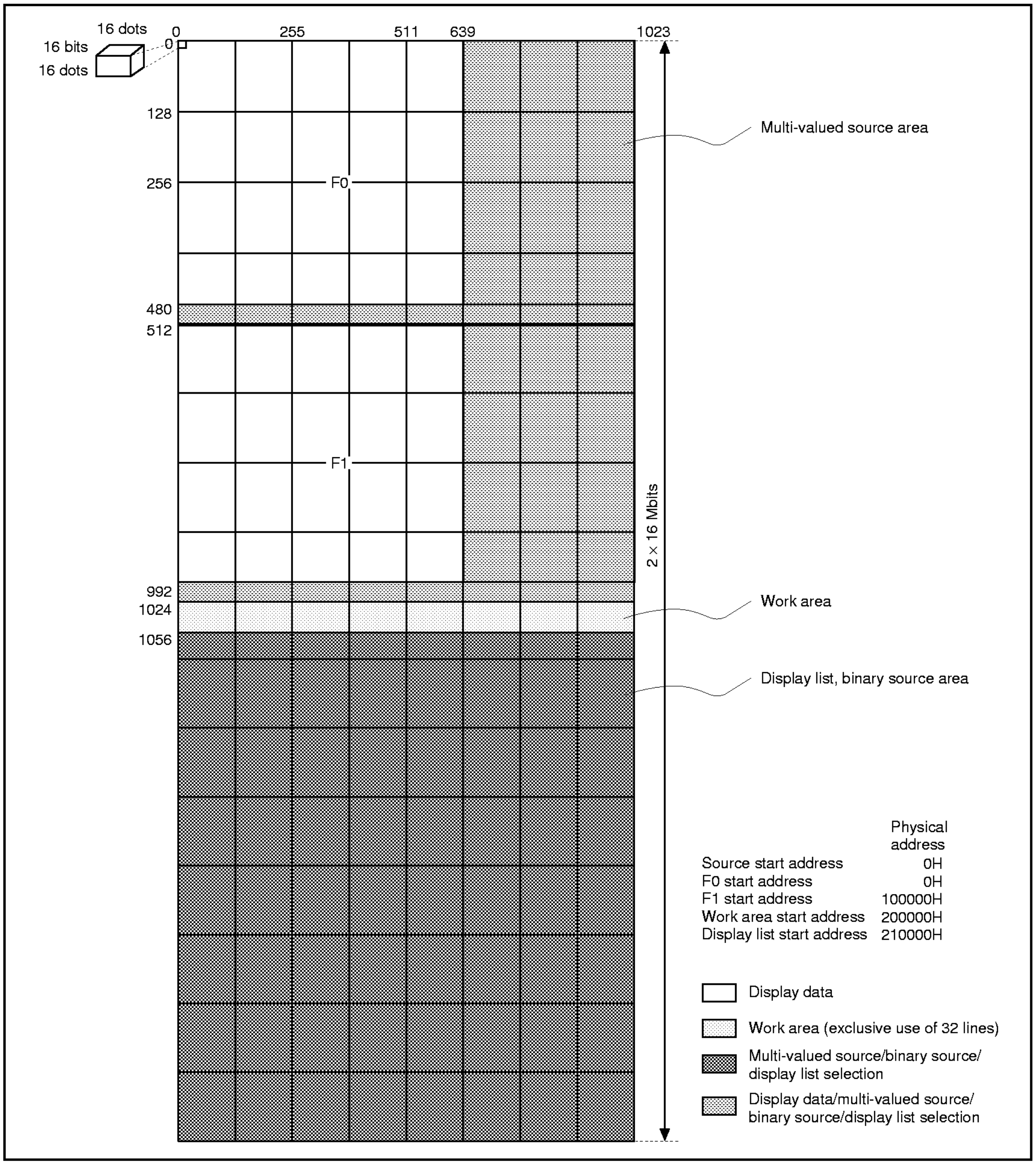
	Physical address
Source start address	0H
F0 start address	0H
F1 start address	40000H
Work area start address	80000H
Display list start address	84000H

Figure 3-8 Memory Map Example 5
[Screen Size at 16 Bits/Pixel (320 × 240 Equivalent)]



	Physical address
Multi-valued source start address	0H
F0 start address	0H
F1 start address	80000H
Work area start address	100000H
Display list start address	108000H

Figure 3-9 Memory Map Example 6
[Screen Size at 16 Bits/Pixel (640 × 240 Equivalent)]


16 dots
16 bits
16 dots
0
0
255
511
639
1023
128
256
F0
480
512
F1
992
1024
1056
2 × 16 Mbits
Multi-valued source area
Work area
Display list, binary source area
Display data
Work area (exclusive use of 32 lines)
Multi-valued source/binary source/ display list selection
Display data/multi-valued source/ binary source/display list selection

	Physical address
Source start address	0H
F0 start address	0H
F1 start address	100000H
Work area start address	200000H
Display list start address	210000H

Figure 3-10 Memory Map Example 7
[Screen Size at 16 Bits/Pixel (640 × 480 Equivalent)]

3.3 Display and Display Control

3.3.1 Overview

The Q2i has two screens, a drawing screen and a display screen, managed by means of rendering coordinates. Display is performed for both these screens in accordance with double-buffering control designated by the user.

A display function for controlling display timing is also provided, enabling the Q2i to coordinate with the display timing of a display device receiving output from it.

When the GBM bit is set to 1 in the Q2i's rendering mode register (REMR), a function can be used that converts YUV or (YUV data color images to RGB data. When the GBM bit is cleared to 0, the color palette can be used, enabling 256 colors to be specified out of a total of 262,144.

3.3.2 Double-Buffering Control

The Q2i uses double-buffering control to alternately switch the display and drawing areas located in the UGM. An area switching operation is called a frame change. There are three modes for double-buffering control: auto display change mode, auto rendering mode, and manual display change mode. In auto display change mode, a frame change is performed each time the Q2i detects a $\overline{\text{VSYNC}}$ synchronization pulse. In auto rendering mode, a frame change is performed once only when the Q2i detects a $\overline{\text{VSYNC}}$ synchronization pulse after drawing is completed. In manual display change mode, the Q2i performs a frame change once only at the $\overline{\text{VSYNC}}$ synchronization pulse following issuance by the SuperH of a frame change directive to the Q2i. These modes are specified by the double-buffering mode bits (DBM) in the system control register.

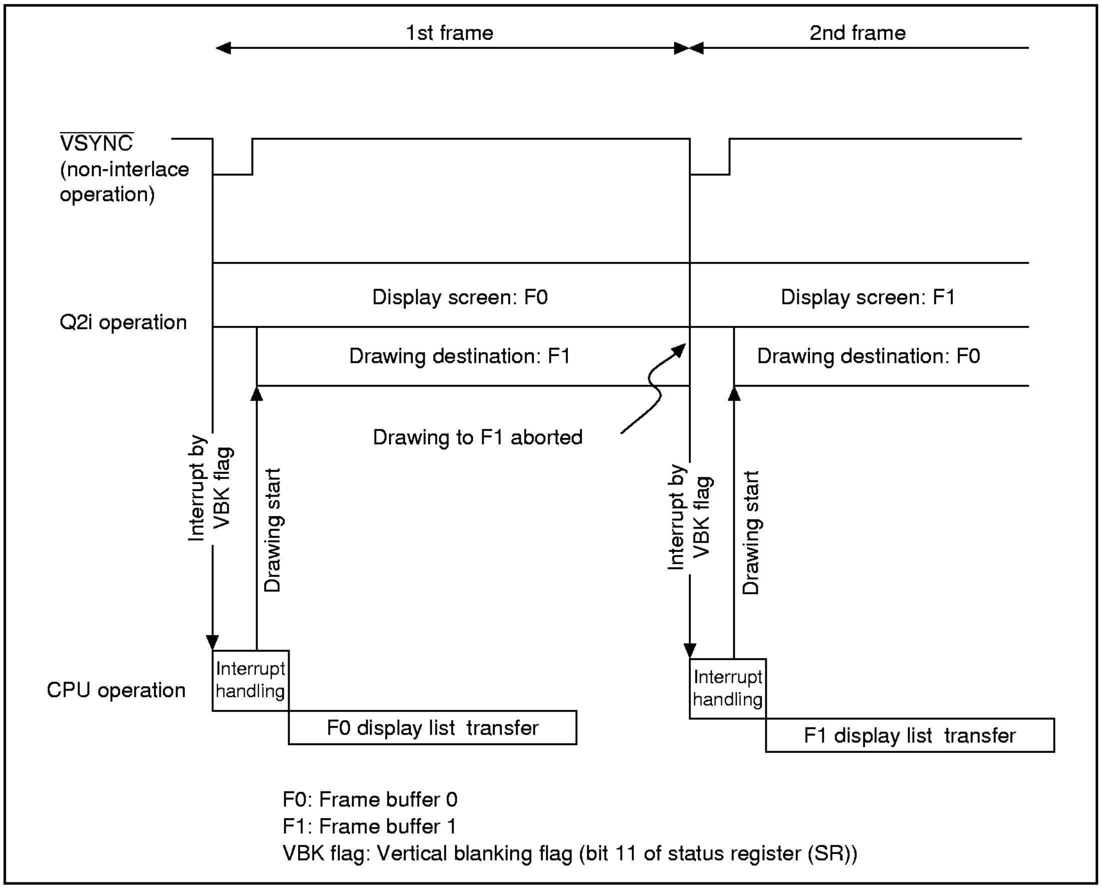
When double-buffering control is performed in a mode other than auto display change mode, frame changes can be performed according to the relevant mode by detection of a $\overline{\text{VSYNC}}$ synchronization pulse by the SuperH, followed by setting of the rendering start bit (RS) to 1.

Frame change timing in double-buffering control is performed in frame units when the Q2i is operated in non-interlace or interlace mode, and in field units when operated in interlace sync & video mode.

When the Q2i is operated in interlace mode, the status register frame flag (FRM) is used for $\overline{\text{VSYNC}}$ synchronization pulse detection by the SuperH. When the Q2i is operated in non-interlace mode, synchronization pulses are detected using the vertical blanking flag (VBK). When the Q2i is operating in interlace sync & video mode, since the first frame corresponds to the even field and the second frame to the odd field, synchronization pulses are detected using VBK or FRM.

Examples are given below for Q2i non-interlace operation, with a description of the operation in each mode.

Auto Display Change Mode: In auto display change mode, display frame switching has priority. If drawing is in progress when the frame is switched, drawing is aborted midway through that display list. It is therefore essential for drawing to be finished before the arrival of a $\overline{\text{VSYNC}}$ synchronization pulse. An outline of operation in this mode is shown in figure 3-11.



F0: Frame buffer 0
F1: Frame buffer 1
VBK flag: Vertical blanking flag (bit 11 of status register (SR))

Figure 3-11 Operation in Auto Display Change Mode

Auto Rendering Mode: In auto rendering mode, display switching is not performed until drawing ends. If drawing does not end within one frame, it is continued without interruption. An outline of operation in this mode is shown in figure 3-12.

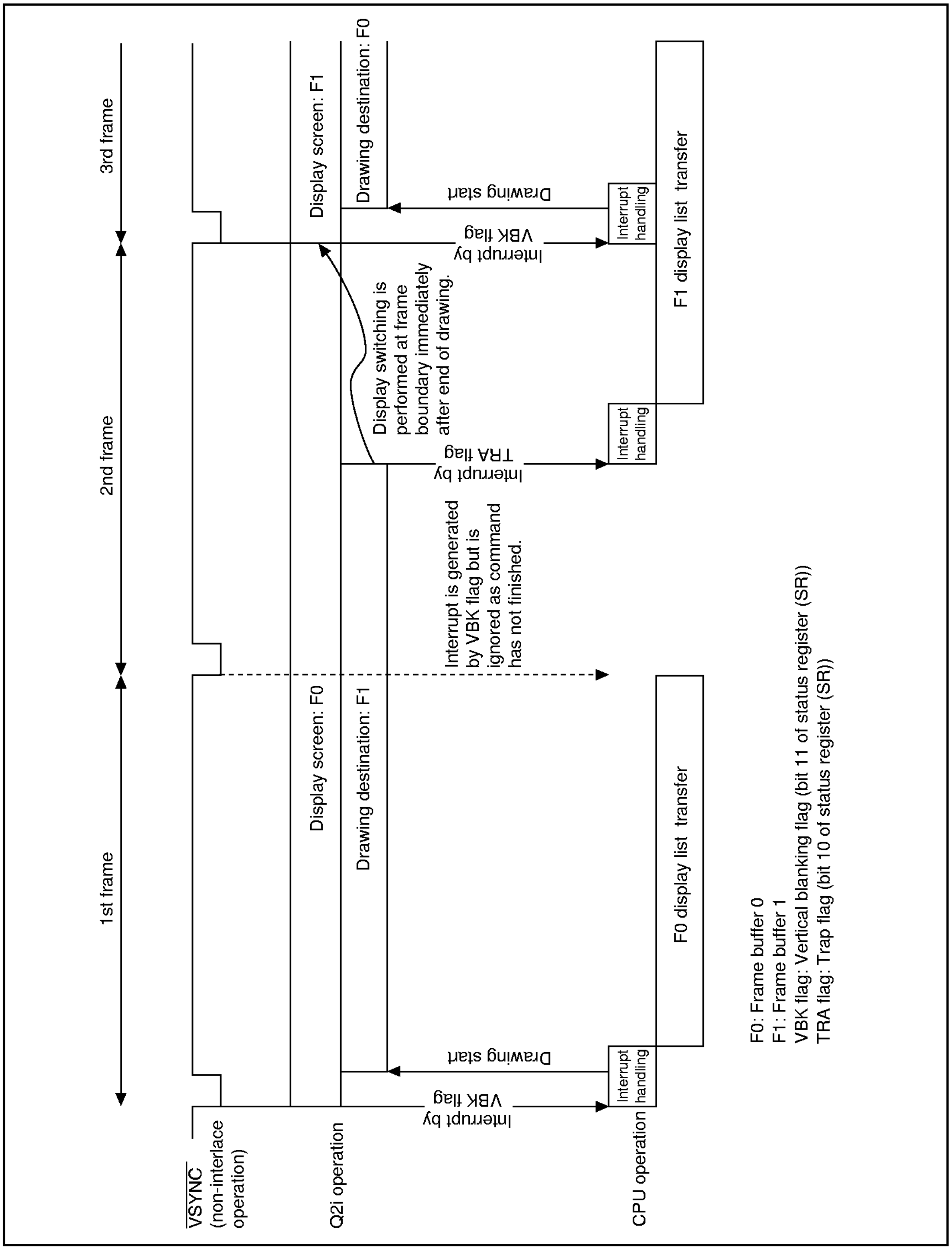

1st frame
2nd frame
3rd frame
VSYNC (non-interlace operation)
Q2i operation
CPU operation
Display screen: F0
Drawing destination: F1
Display screen: F1
Drawing destination: F0
Interrupt by VBK flag
Drawing start
Interrupt handling
F0 display list transfer
Interrupt is generated by VBK flag but is ignored as command has not finished.
Interrupt by TRA flag
Display switching is performed at frame boundary immediately after end of drawing.
F1 display list transfer
F0: Frame buffer 0
F1: Frame buffer 1
VBK flag: Vertical blanking flag (bit 11 of status register (SR))
TRA flag: Trap flag (bit 10 of status register (SR))

Figure 3-12 Operation in Auto Rendering Mode

Manual Display Change Mode: In manual display change mode, display frame switching and the start of drawing are controlled independently by software. Display switching can be performed either by performing F0/F1 switching with an SYSR DC bit setting by software, or by setting the start address of F0 or F1 in the display start address register indicated by DBR in SR. The start of drawing is controlled by the RS bit in SYSR. Interrupts by means of the VBK flag and TRA flag in SR are used for the control timing. An outline of operation when using the DC bit in this mode is shown in figure 3-13. When switching from this mode to another double-buffer control mode, it is essential to set the DC bit to 1 and make a frame change first.

Make sure that the DC bit is cleared to 0 before setting it to 1.

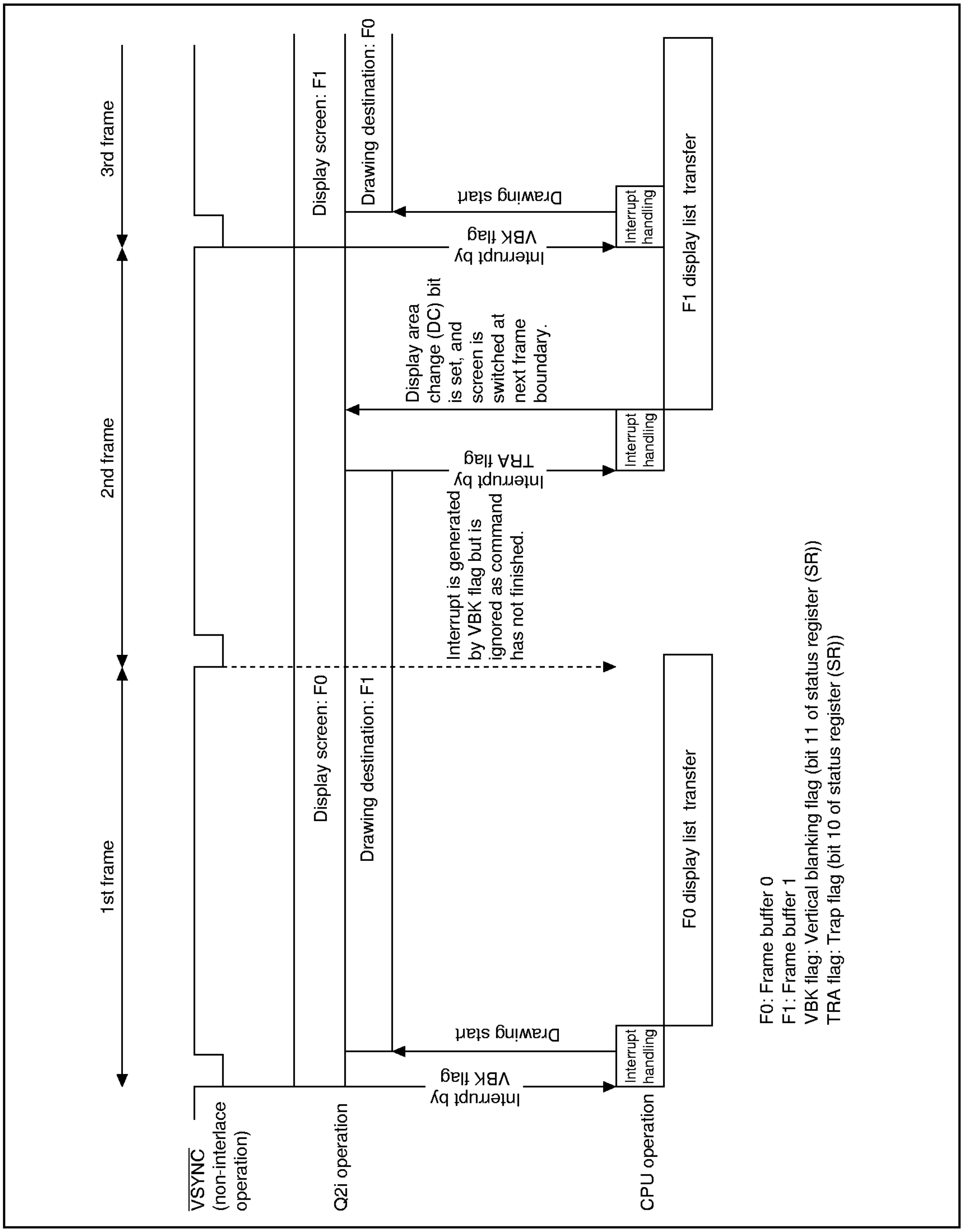
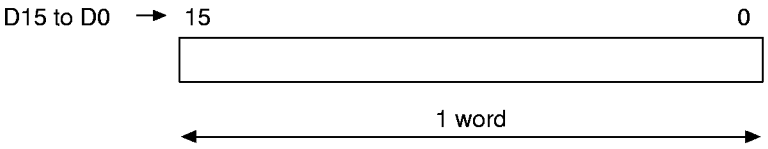


Figure 3-13 Operation in Manual Display Change Mode

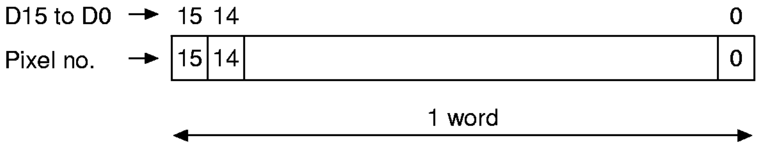
3.3.3 Color Data Formats

Input Color Data Configurations: Input color data configurations are shown below.

1. 16-bit data

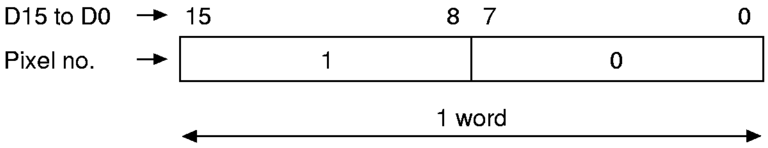


2. 1-bit/pixel data

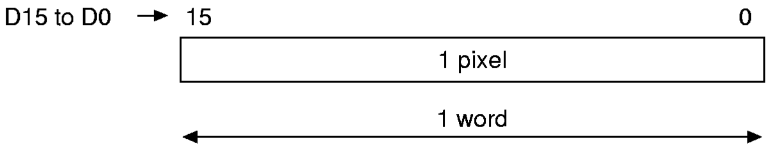


Note: The pixel number runs from 0 upward from the left to right side of the screen.

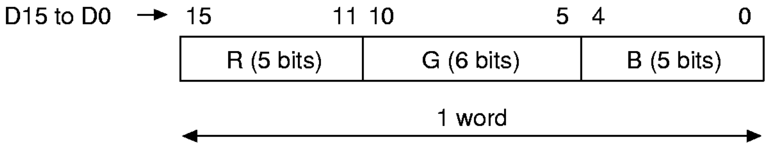
3. 8-bit/pixel data



4. 16-bit/pixel data

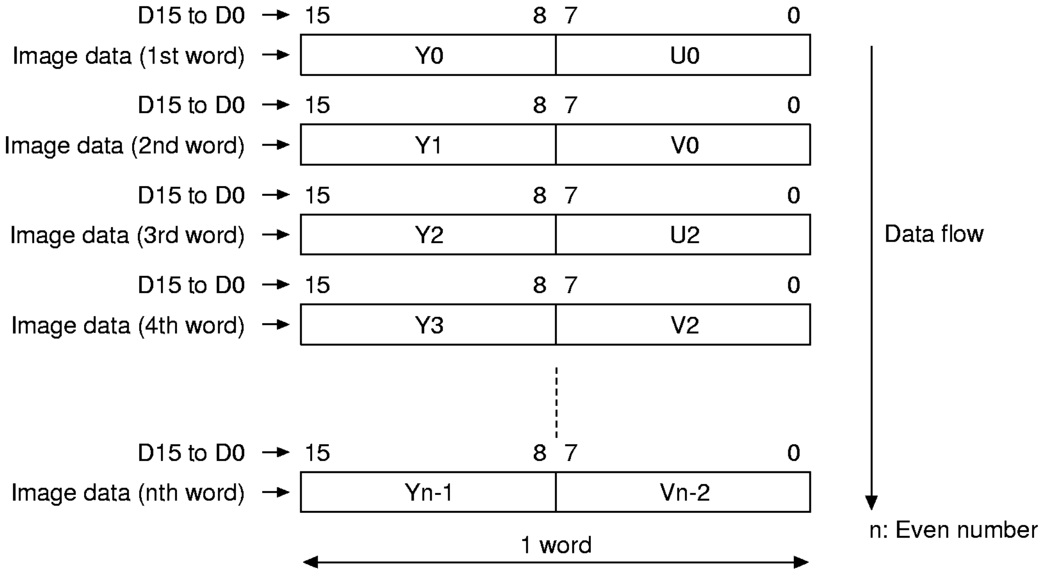


5. RGB data (16-bit/pixel data)



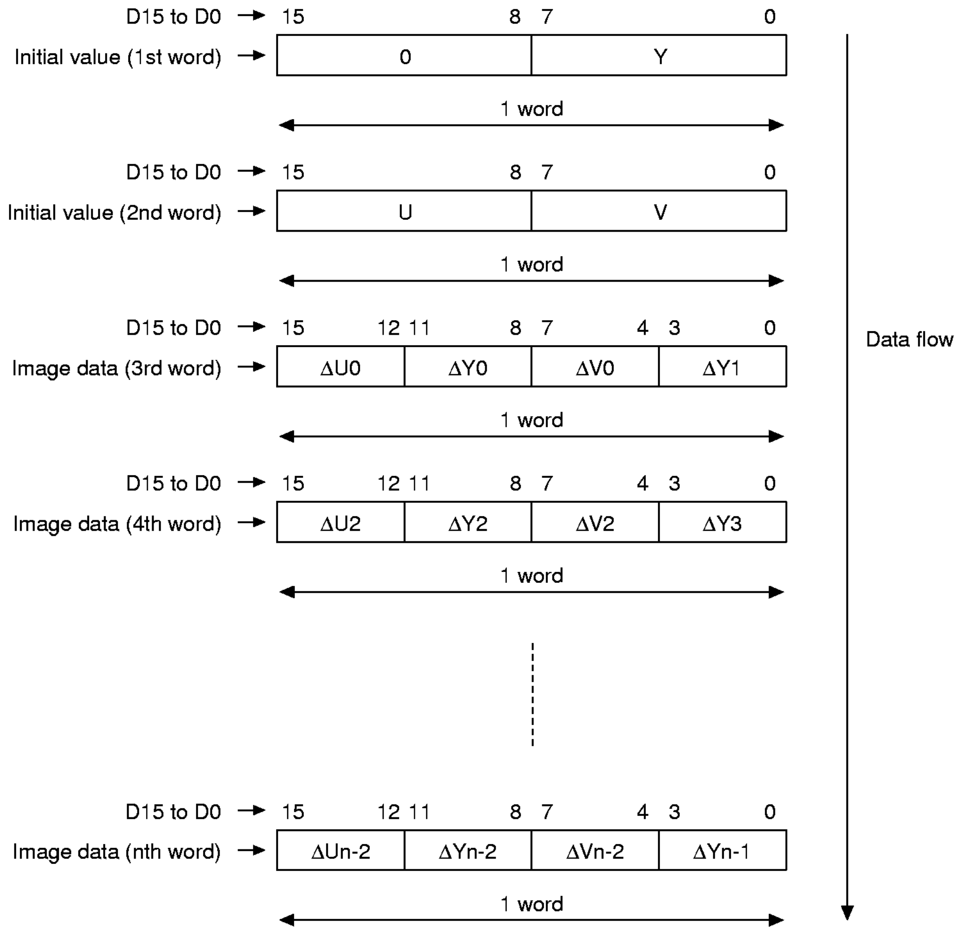
6. YUV data

YUV data uses a 4:2:2 format. The U and V data is horizontally reduced data.

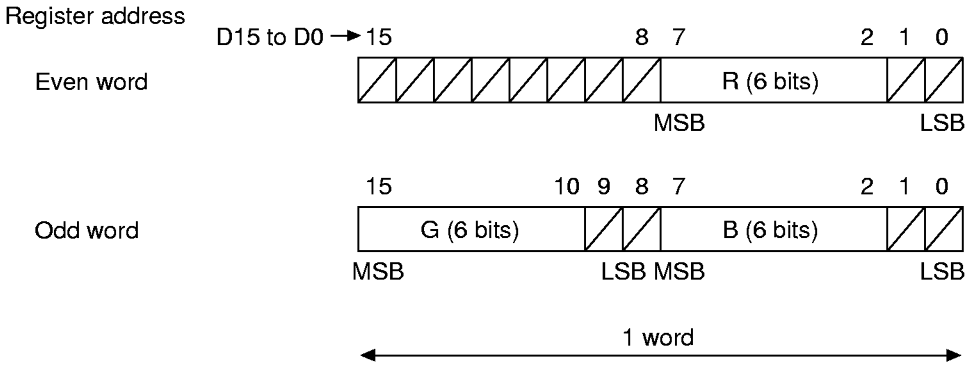


7. (YUV data

(YUV data uses a raster as the basic unit. The data configuration for one raster consists of the initial value in the first two words and compressed image data in the remaining words.

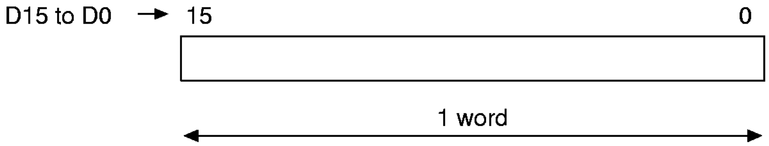


Color Palette Register Color Data Configuration: The color palette register color data configuration is shown below.

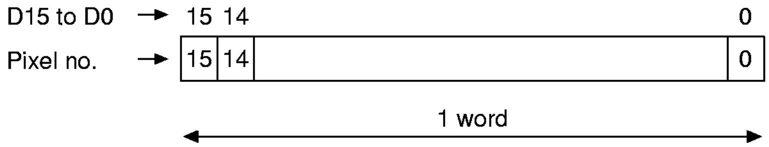


Configurations of Data in UGM: UGM data configurations are shown below.

1. 16-bit data

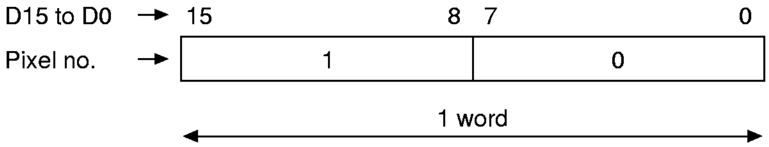


2. 1-bit/pixel data

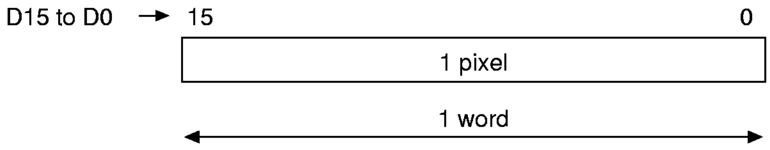


Note: The pixel number runs from 0 upward from the left to right side of the screen.

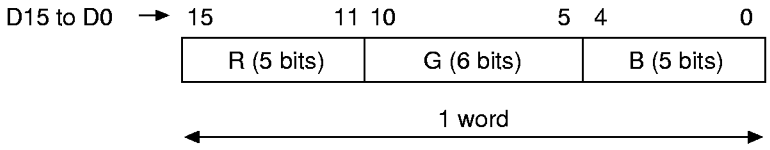
3. 8-bit/pixel data



4. 16-bit/pixel data

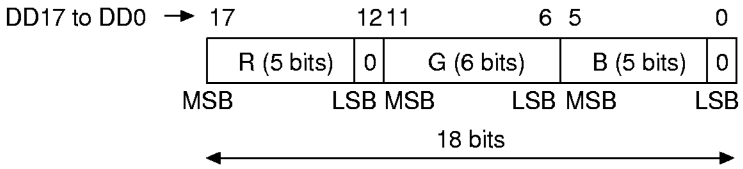


5. RGB data (16-bit/pixel data)

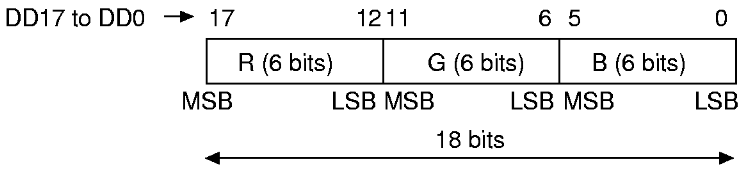


Output Color Data Configurations (Q2i → Display Monitor): Output data configurations are shown below.

1. RGB data
   a. When the frame buffer is 16 bits/pixel and a color palette is not used

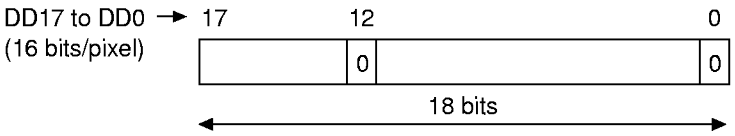


   b. When the frame buffer is 8 bits/pixel and a color palette is used



2. 18-bit data (in case of independent format)

   Frame buffer bits 15 to 11 are output to DD17 to DD13, and bits 10 to 0 are output to DD11 to DD1.



Color Data Format Conversion

- (YUV/YUV-to-RGB conversion

  The specifications of the function for converting (YUV or YUV data in main memory to RGB data are shown in figure 3-14.

  The Q2i converts (YUV or YUV data transferred from the CPU to RGB data and stores it in the UGM source area. The converted RGB data is used as source data for quadrilateral-drawing commands.

  Q2i registers that should be set by the CPU are as follows. Make the register settings in the order shown in figure 3-15.

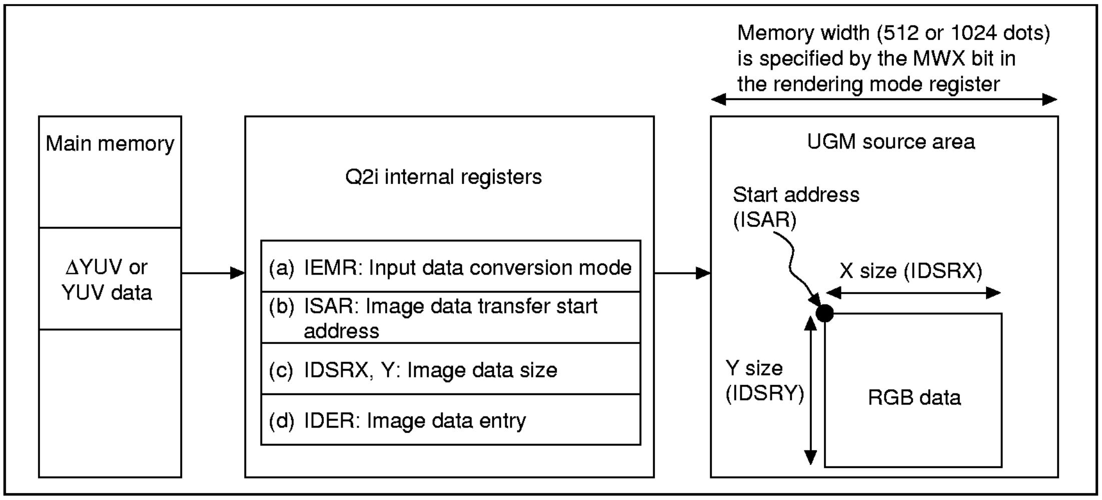

Memory width (512 or 1024 dots) is specified by the MWX bit in the rendering mode register
Main memory
ΔYUV or YUV data
Q2i internal registers
(a) IEMR: Input data conversion mode
(b) ISAR: Image data transfer start address
(c) IDSRX, Y: Image data size
(d) IDER: Image data entry
UGM source area
Start address (ISAR)
X size (IDSRX)
Y size (IDSRY)
RGB data

Figure 3-14 Specifications of ΔYUV/YUV-to-RGB Conversion Functions

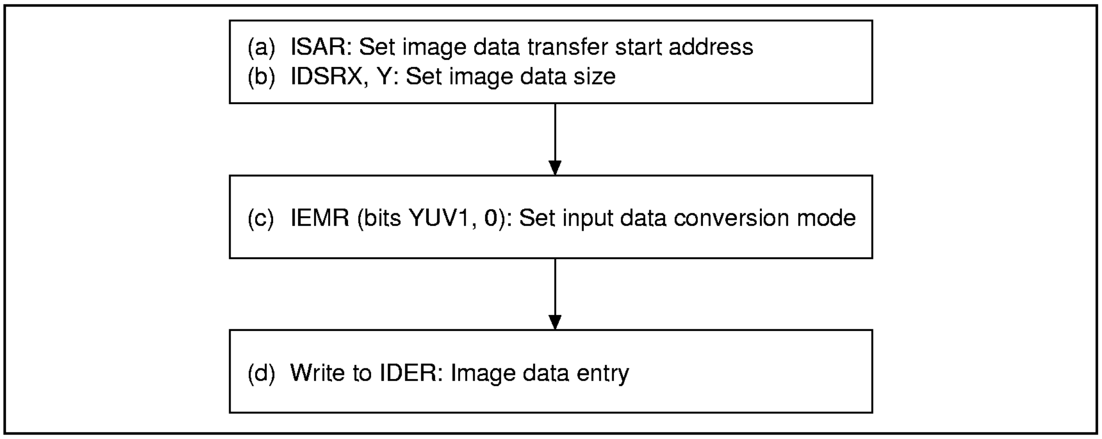

(a) ISAR: Set image data transfer start address
(b) IDSRX, Y: Set image data size
(c) IEMR (bits YUV1, 0): Set input data conversion mode
(d) Write to IDER: Image data entry

Figure 3-15 Register Setting Procedure for ΔYUV/YUV-to-RGB Conversion

3.3.4 Display Functions

The Q2i has functions for outputting image data, drawn in the UGM, in synchronization with externally or internally generated display timing.

Register and Display Screen: In the Q2i, horizontal and vertical display timing for the display screen is set in the display parameter register (see section 5.5, Display Control Registers).

The display control registers settings depend on the scanning and synchronization systems used. The calculations shown in table 3-2 should therefore be carried out before making the display parameter register settings.

Figure 3-16 shows the display timing in non-interlace mode. The display screen is defined by the variables shown in table 3-3. Set the number of rasters within one VSYNC cycle for each of vc, vsw, ys, and yw, regardless of the display mode register scan mode.

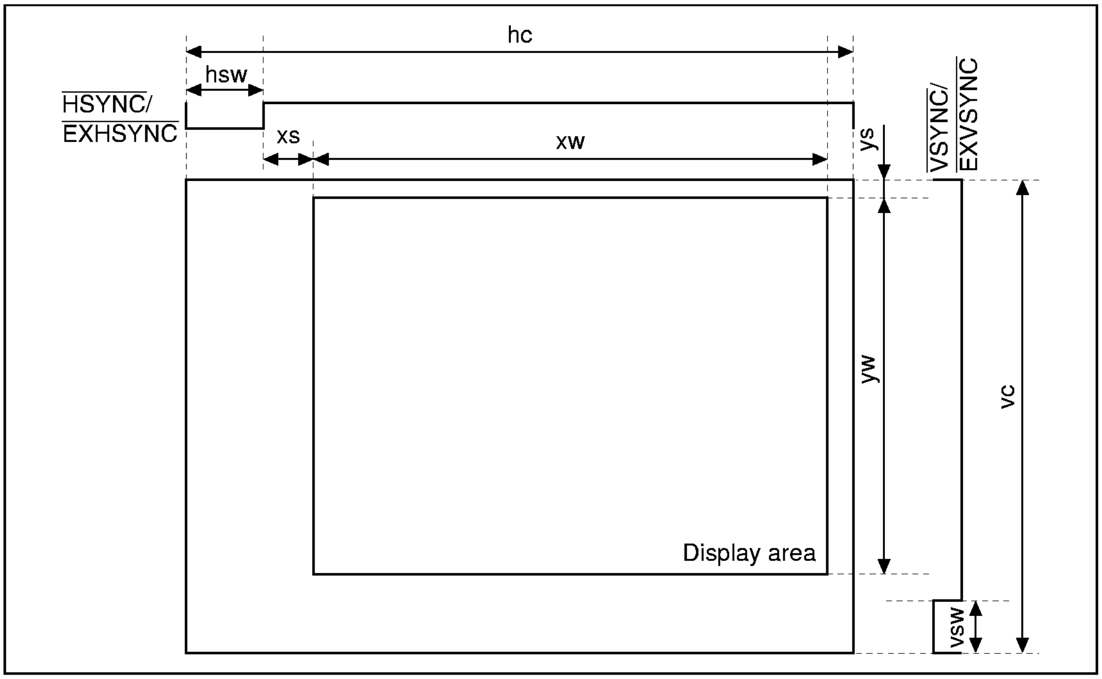
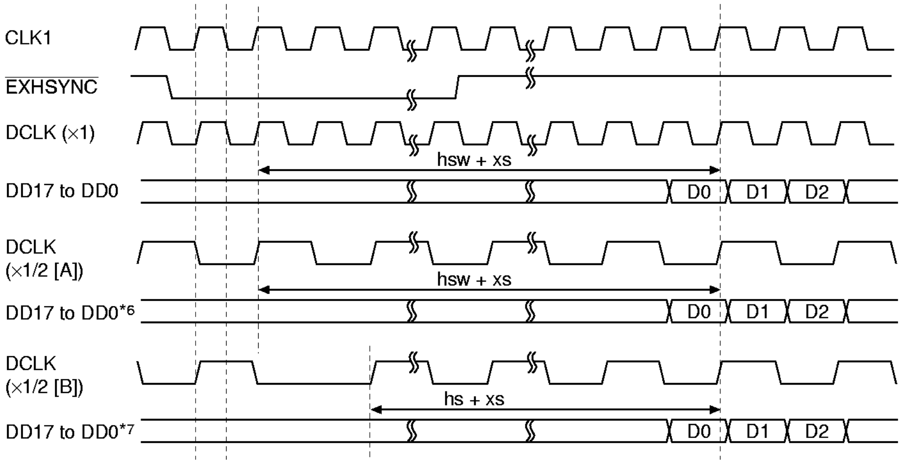


Figure 3-16 Display Timing (Non-Interlace Mode)

Table 3-2 Register Settings*[1]

				Operating Mode	
Register No. (Address)	**Register Name**		**Bit Names**	**Master Mode**	**TV Sync Mode**
008	Display size register X (DSRX)		DSX	xw-1	xw-1*[5]
009	Display size register Y (DSRY)		DSY	yw-1	yw-1
013	Display windows	Horizontal display start position register (DSWR-HDS)	HDS	hsw+xs-3	hsw+xs-8*[2], *[3]
014		Horizontal display end position register (DSWR-HDE)	HDE	hsw+xs-3+xw	hsw+xs-8+xw*[2]
015		Vertical display start position register (DSWR-VDS)	VDS	ys	ys*[4]
016		Vertical display end position register (DSWR-VDE)	VDE	ys+yw	ys+yw
017	Horizontal sync pulse width register (HSWR)		HSW	hsw-1	hsw-1
018	Horizontal scan cycle register (HCR)		HC	hc-1	hc
019	Vertical sync position register (VSPR)		VSP	vc-vsw	vc-vsw
01A	Vertical scan cycle register (VCR)		VC	vc	vc+2

Notes:
1. In all scanning modes the settings of bits VDS, VDE, VSP, and VC are made for a one-frame unit.
2. The HDS and HDE bit specifications are the values from the rise of DCLK after the fall of $\overline{\text{EXHSYNC}}$ is latched at the rise of CLK1, and then latched at the next fall of CLK1.
3. HDS bit lower limit values are shown in the table below. This value should be set as a number of dot clock cycles. Access from the CPU may be kept waiting during this period.
4. In interlace and interlace sync & video modes, the setting is: VDS ≥ 2. In non-interlace mode, the setting is: VDS ≥ 0.
5. The lower limit of bits DSX9 to DSX0 is: DSX9–0 ≥ 4.



6. When $\overline{\text{EXHSYNC}}$ cycle is an even multiple of CLK1
7. When $\overline{\text{EXHSYNC}}$ cycle is an odd multiple of CLK1

Table 3-3 HDS Lower Limit Values (t_{cyc0} Units)

Clock Ratio	8-Bits/Pixel (GBM = 0)		16-Bits/Pixel (GBM = 1)
CLK0/DCLK	**BG = 0**	**BG = 1**	**BG = 0**
2 to less than 4	64 t_{cyc0}	191 t_{cyc0}	96 t_{cyc0}
4 or more	114 t_{cyc0}	151 t_{cyc0}	126 t_{cyc0}

Table 3-4 Variables Defined by Display Screen

Variable	Description	Unit
hc*[8]	Horizontal scan cycle	Dot clock
hsw	Horizontal sync pulse width	Dot clock
xs	Interval between $\overline{\text{HSYNC}}$ rise and display screen horizontal display start position	Dot clock
xw	Display screen display width per raster	Dot clock
vc*[9]	Vertical scan cycle	Raster lines
vsw	Vertical sync pulse width	Raster lines
ys	Interval between $\overline{\text{VSYNC}}$ rise and display screen vertical display start position	Raster lines
yw	Display screen vertical display interval	Raster lines

8. $hsw + xs + xw < hc – 10$
9. $vsw + ys + yw < vc$

Screen Display: In the Q2i, the DEN (display enable) bit in the system control register (SYSR) can be used to select whether or not display data is to be output to the screen. When display data is not output, the display off output register (DOOR) settings are displayed.

The frame flag (FRM) and vertical blanking flag (VBK) in the status register indicate the position of the fall of the vertical sync signal ($\overline{\text{VSYNC}}$) determined by the set value (VSP9–0) in the vertical sync position register (VSPR) regardless of the synchronization method.

Scanning Systems: The Q2i allows selection of non-interlace mode, interlace mode, or interlace sync & video mode as the scanning system. The mode setting is made in the SCM (scan mode) bits in the display mode register (DSMR). In non-interlace mode, one frame is composed of one field. In interlace mode, one frame is composed of two fields, even and odd, in which the same data is displayed. In interlace sync & video mode, also, one frame is composed of two fields, even and odd, but in this mode different data is displayed in these two fields. In master mode, the Q2i outputs a high-level signal from the $\overline{\text{ODDF}}$ pin during even field display, and a low-level signal during odd field display. In TV sync mode, a high-level signal is input at the $\overline{\text{ODDF}}$ pin to display the even field, and a low-level signal to display the odd field. Figure 3-17 shows examples of raster scan control display.

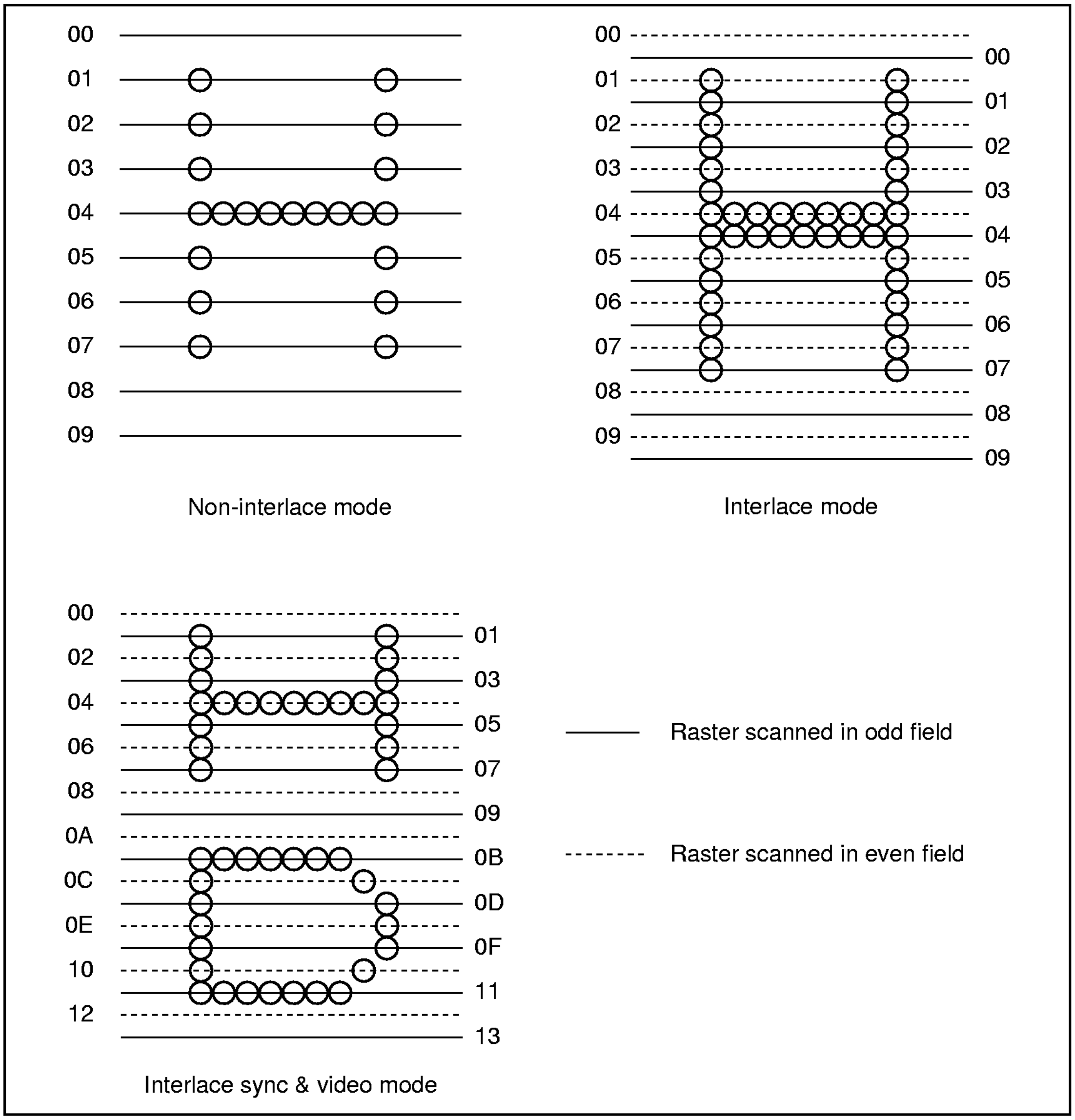


Figure 3-17 Examples of Raster Scan Control Display

Synchronization Systems: The Q2i is provided with a TV sync function in addition to master mode to simplify synchronization with an external device.

The TVM (TV sync mode) bits in the display register (DSMR) are used to select master mode or TV sync mode.

- Internal Synchronization Mode (Master Mode)

  Setting the horizontal and vertical sync signal ($\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$) cycles and pulse widths in the display control register outputs the corresponding waveforms, and display data is output in synchronization with these signals.

  In interlace mode and interlace sync & video mode, a signal indicating odd field or even field is output at the $\overline{\text{ODDF}}$ pin.

- External Synchronization Mode (TV Sync Mode)

  In TV mode, display data is output in synchronization with vertical and horizontal sync signals ($\overline{\text{EXHSYNC}}$ and $\overline{\text{EXVSYNC}}$) input from an external source. The Q2i outputs display data on the basis of the fall of the $\overline{\text{EXHSYNC}}$ signal and the rise of the $\overline{\text{EXVSYNC}}$ signal.

  In this mode, the horizontal sync signal, vertical sync signal, and clock from the sync signal generator should be input at the $\overline{\text{EXHSYNC}}$, $\overline{\text{EXVSYNC}}$, and CLK1 pins, respectively. Signals that do not contain equivalent pulses should be used for the sync signals.

  In interlace mode and interlace sync & video mode, a signal indicating odd field or even field should be input at the $\overline{\text{ODDF}}$ pin.

  In non-interlace mode, the $\overline{\text{ODDF}}$ pin should be fixed high or low to prevent an unstable input level at this pin.

  The Q2i performs UGM refreshing based on $\overline{\text{EXHSYNC}}$ and $\overline{\text{EXVSYNC}}$. Therefore, $\overline{\text{EXHSYNC}}$ and $\overline{\text{EXVSYNC}}$ must be input to enable UGM refreshing to be carried out.

  The signal flow in TV sync mode is shown in figure 3-18.

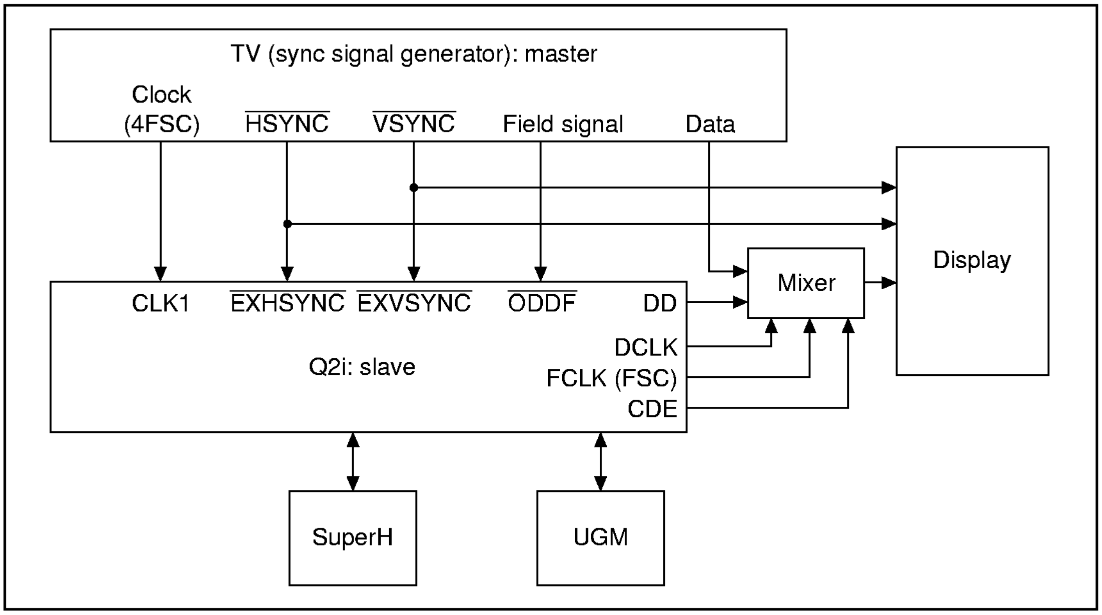


Figure 3-18 Signal Flow in TV Sync Mode

- Synchronization System Switching Mode

  This mode is used to switch to master mode if the external sync signal generator malfunctions during operation in TV sync mode.

  The processing sequence in this case is as follows: detection of malfunction, set this mode, switch CLK1 to a clock supplied by a different system, then set master mode.

Refresh Control: The number of refresh cycles for the UGM connected to the Q2i is set in bits REF3–0 (refresh cycle count) in the display mode register (DSMR).

Bit 3: REF3	Bit 2: REF2	Bit 1: REF1	Bit 0: REF0	Operation
0	0	0	0	Refresh timing is not output.
*	*	*	*	Refresh timing is set to any value from 1 to 15 cycles, and output.

The setting made in bits REF3–0 is the number of refreshes per raster.

For example, if the refreshing specification of the memory used is 1/60 sec for one field at 512 cycles/8 ms, the necessary number of cycles in one field is 1065. The number of refresh cycles (times/raster) should be set so that the number of cycles required for one field is the number of lines.

The Q2i supports CAS-before-RAS refresh mode.

The refresh cycles set in bits REF3–0 are executed from the fall of the DISP signal. Table 3-5 shows some sample settings.

Table 3-5 Estimated Number of Refresh Cycles (for 1/60s Field)

Sample Display		Number of Refresh Cycles (Per Raster)			
Screen Sizes	**Memory Size**	**4 Mbits × 1**	**4 Mbits × 2**	**16 Mbits × 1**	**16 Mbits × 2**
320 × 240		5	5	—	—
640 × 240		—	5	5	5
640 × 480		—	—	3	3

Two-Screen Combination/Scroll Display: Setting the BG bit to 1 in the display mode register (DSMR) enables the background screen (BG) to be superimposed on a frame screen (FB0 or FB1). This function is valid in 8-bit/pixel mode. There is no mode in which BG alone is displayed. the BG size and bit configuration are the same for FB0 and FB1. The number of display colors is 256 for FB and BG combined.

The BG start position is specified by a 2-dimensional virtual address using background coordinate start registers X and Y (BGSX, BGSY). The origin of these 2-dimensional virtual coordinates is the start of the UGM. Set BG so that it does not overlap FB0 and FB1.

Superimposition is achieved by outputting background screen pixels if the frame screen pixels are of a transparent color, or by outputting frame screen pixels otherwise. The setting for a transparent color is all 0s, as with drawing attributes.

Register updating should be carried out in the register update interval shown in section 5.2, Register Updating. The display contents differ according to the setting of the background screen wraparound mode (WRAP) bit.

Table 3-6 Background Screen Related Register Settings

Register		Field			
Address	**Name**	**Bit No.**	**Name**	**Set Value**	**Notes**
H'005	DSMR	10	BG	1/0	Combination on/off
		11	WRAP	1/0	Wraparound on/off
H'006	REMR	0	GBM	0	8-bits/pixel
H'026	BGSRX	9–0	BGSX9–0	BG starting point X coordinate	Coordinate origin is start of UGM
H'027	BGSRY	13–0	BGSY13–0	BG starting point Y coordinate	

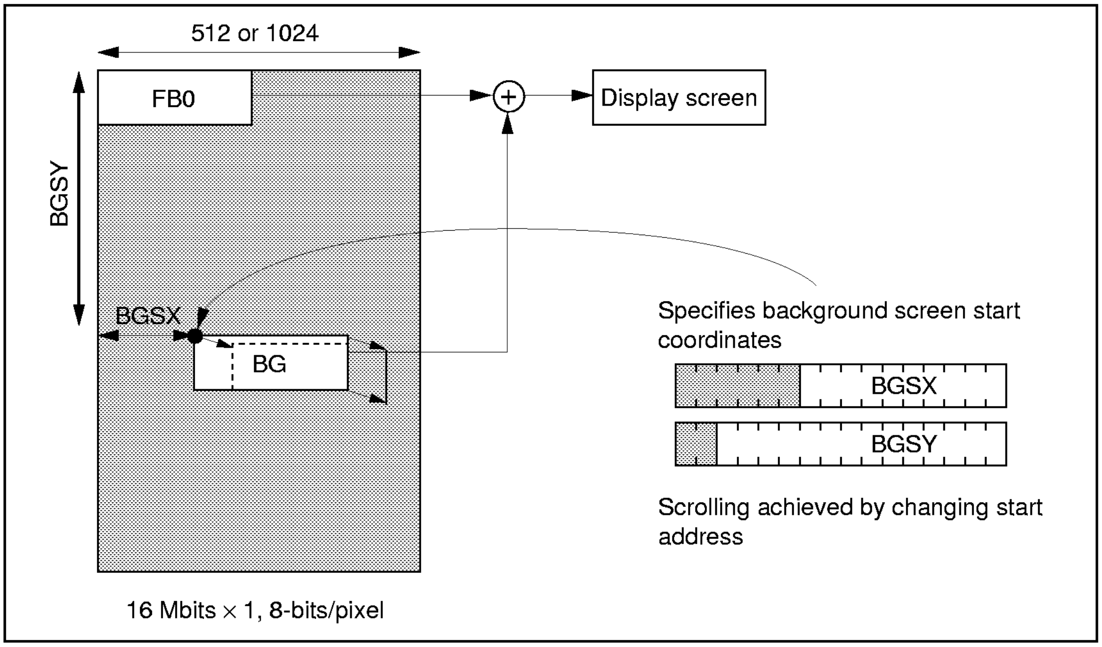

512 or 1024
FB0
Display screen
BGSY
BGSX
BG
Specifies background screen start coordinates
BGSX
BGSY
Scrolling achieved by changing start address
16 Mbits × 1, 8-bits/pixel

Figure 3-19 Example of Background Screen Simple Scroll (WRAP = 0)

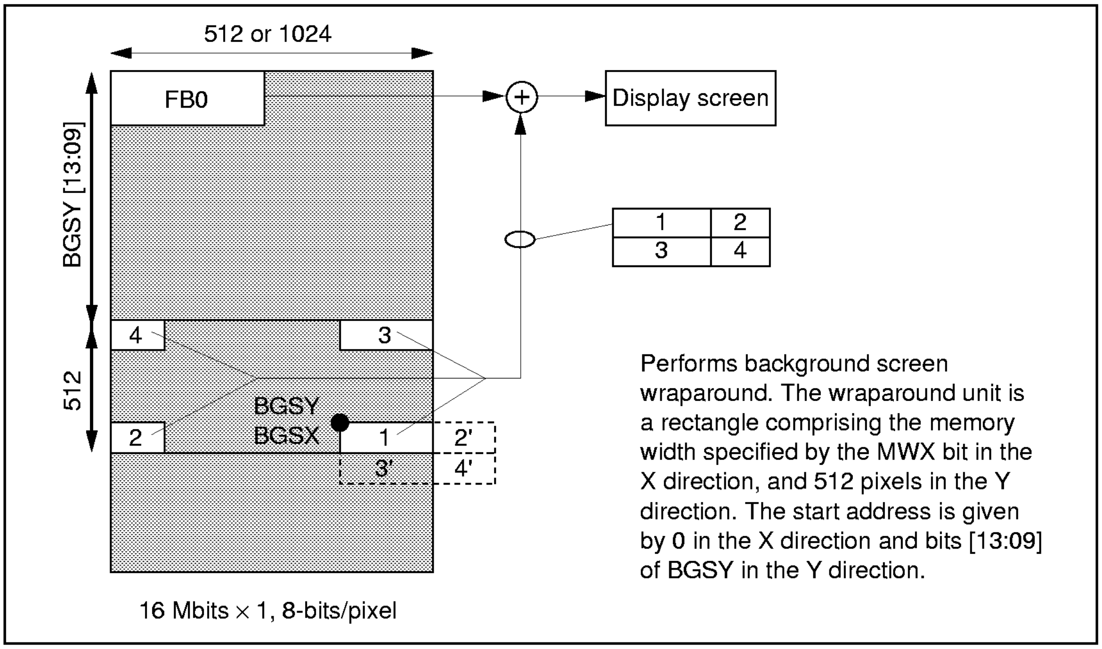

512 or 1024
FB0
Display screen
BGSY [13:09]
512
1
2
3
4
4
3
2
BGSY
BGSX
1
2'
3'
4'
Performs background screen wraparound. The wraparound unit is a rectangle comprising the memory width specified by the MWX bit in the X direction, and 512 pixels in the Y direction. The start address is given by 0 in the X direction and bits [13:09] of BGSY in the Y direction.
16 Mbits × 1, 8-bits/pixel

Figure 3-20 Example of Background Screen Wraparound Scroll (WRAP = 1)

Display Timing: The relationship between the display control register settings and the display signals is shown in figure 3-21.

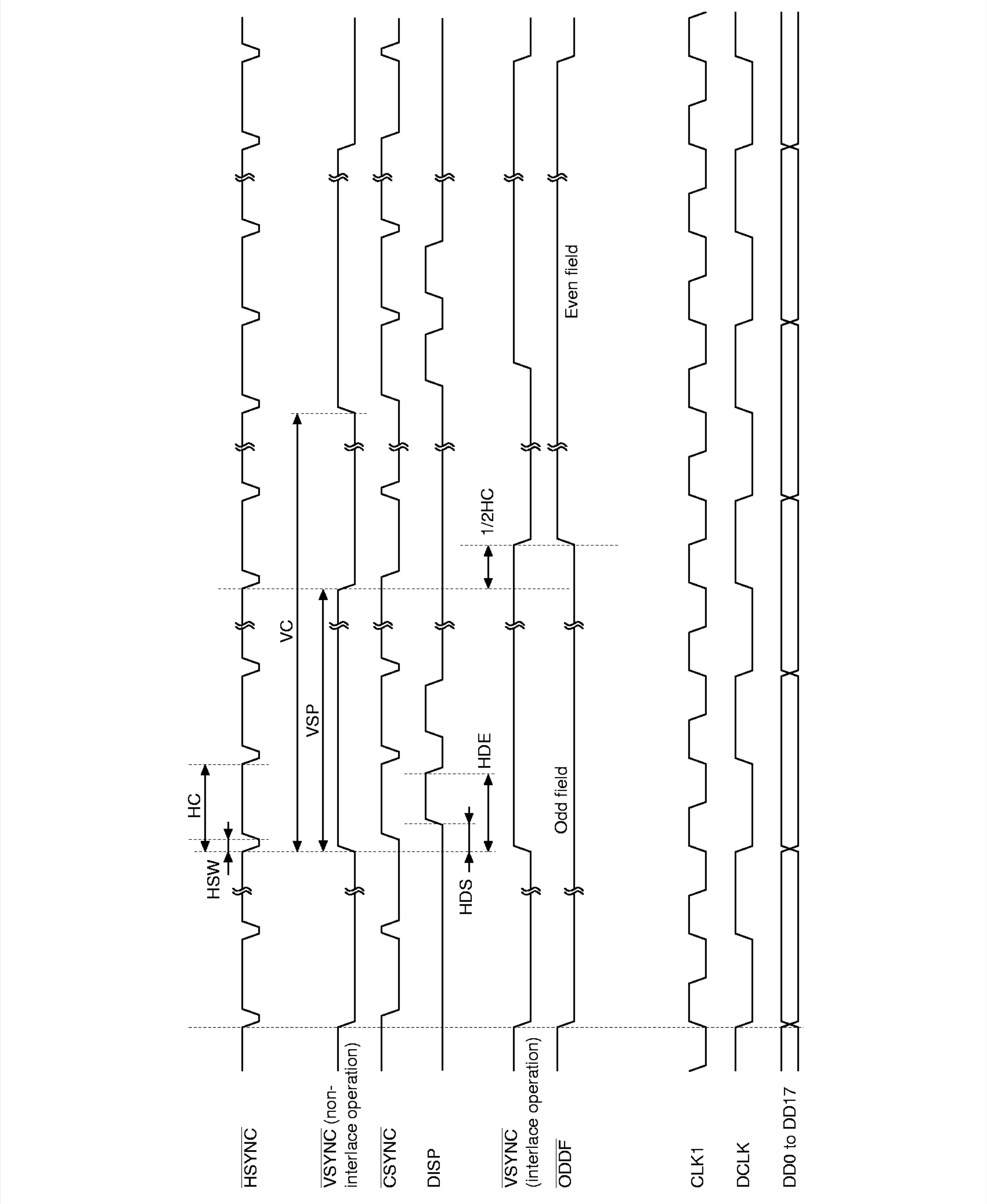

$\overline{\text{HSYNC}}$
$\overline{\text{VSYNC}}$ (non-interlace operation)
$\overline{\text{CSYNC}}$
DISP
$\overline{\text{VSYNC}}$ (interlace operation)
$\overline{\text{ODDF}}$
CLK1
DCLK
DD0 to DD17
HSW
HC
VC
VSP
HDS
HDE
1/2HC
Odd field
Even field

Figure 3-21 Display Timing

3.4 Initial States

3.4.1 Initial States (when Specied Power is Turned On)

Initial states are undefined.

Registers: undefined
I/O pins: undefined
Output pins: low/high output

3.4.2 Reset State (when Low Level is Input at $\overline{\text{RESET}}$ Pin)

Registers: After a reset, the Q2i's internal registers are initialized as shown in table 3-7.

Table 3-7 Initial Register Values after Reset

Register Address		R/W	Register Name		Abbreviation	Data															
CS1	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	—	—	—		—	—															
0	000	R/W	System control		SYSR	1	1	0	▨	▨	▨	0	0	DBM		0	0	▨	▨	▨	▨
	001	R	Status		SR	0	0	0	0	0	0	0	DBF	▨	▨	▨	▨	0	0	1	1
	002	W	Status register clear		SRCR	TVCL	FRCL	DMCL	CECL	VBCL	TRCL	CSCL	▨	▨	▨	▨	▨	▨	▨	▨	▨
	003	R/W	Interrupt enable		IER	0	0	0	0	0	0	0	▨	▨	▨	▨	▨	▨	▨	▨	▨
	004	R/W	Memory mode		MEMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	MES			MEA		▨	▨
	005	R/W	Display mode		DSMR	▨	▨	▨	▨	▨	▨	0	DOT	1	0	SCM		1	0	0	0
	006	R/W	Rendering mode		REMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	MWX	▨	▨	▨	▨	▨	GBM
	007	R/W	Input data conversion mode		IEMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	0	0

Memory Control Registers

CS1	A [10:1]	R/W	Register Name		Abbreviation	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	008	R/W	Display size	X	DSR	▨	▨	▨	▨	▨	▨	DSX									
	009	R/W		Y		▨	▨	▨	▨	▨	▨	▨	DSY								
	00A	R/W	Displav start address	0	DSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	DSA0						
	00B	R/W		1		▨	▨	▨	▨	▨	▨	▨	▨	▨	DSA1						
	00C	R/W	Display list start address	H	DLSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	DLSAH						
	00D	R/W		L		DLSAL												▨	▨	▨	▨
	00E	R/W	Multi-valued source area start address		SSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	SSAH						▨
	00F	R/W	Work area start address		WSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	WSAH						
	010	R/W	DMA transfer start address	H	DMASR	▨	▨	▨	▨	▨	▨	▨	▨	▨	0	0	0	0	0	0	0
	011	R/W		L		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	▨
	012	R/W	DMA transfer word count		DMAWR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Input Data Control Registers

CS1	A [10:1]	R/W	Register Name		Abbreviation	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	021	R/W	Image data transfer start address	H	ISAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	0	0	0	0	0	0
	022	R/W		L		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	▨
	023	R/W	Image data size	X	IDSR	▨	▨	▨	▨	▨	0	0	0	0	0	0	0	0	0	0	0
	024	R/W		Y		▨	▨	▨	▨	▨	▨	0	0	0	0	0	0	0	0	0	0
	025	W	Image data entry		IDER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Only bits marked □ are affected by a reset. Registers other than those shown above are not affected by a reset.

Pins: Table 3-8 shows the Q2i pin states after a reset.

Table 3-8 Pin States After Reset

I/O pins	Input state	D0–D15, $\overline{\text{VSYNC/EXVSYNC}}$, $\overline{\text{HSYNC/EXHSYNC}}$, $\overline{\text{ODDF}}$
	Output state (low-level output)	MD0–MD15
Output pins	Low-level output	DISP, CDE, DD0–DD17
	High-level output	$\overline{\text{DREQ}}$, $\overline{\text{IRL}}$, $\overline{\text{WAIT}}$
	Low/high-level output	$\overline{\text{CSYNC}}$, DCLK, FCLK, MA0–MA11, $\overline{\text{MWE}}$, $\overline{\text{MRAS0}}$, $\overline{\text{MRAS1}}$, $\overline{\text{MLCAS}}$, $\overline{\text{MUCAS}}$, $\overline{\text{MOE}}$

UGM Refreshing: UGM refreshing is not performed immediately after a reset.

Section 4 Display List

4.1 Overview

The display list is an area used for linked drawing command descriptions. Drawing commands comprise four-vertex surface drawing and line drawing commands which draw at rendering coordinates, and work surface drawing and work line drawing commands which draw at work coordinates. There are also register setting, sequence control, and drawing end commands.

In addition, line drawing and trapezoid files have absolute coordinate and relative coordinate specification commands.

Table 4-1 lists the drawing commands.

Table 4-1 Drawing Commands

Type	Command Name	Function
Four-vertex surface drawing	POLYGON4 Quadrilateral paint	Draws quadrilateral with four coordinates as vertices. Painting can be performed with source tiling and specified color
	POLYGON4A	Four-vertex surface drawing with multi-valued source as transfer source
	POLYGON4B	Four-vertex surface drawing with binary source as transfer source
	POLYGON4C	Four-vertex surface drawing using specified color
Line drawing	LINE Polygonal line	Draws solid polygonal line from start coordinates through nodal coordinates.
	LINE	Polygonal line drawing (absolute coordinate specification)
	RLINE	Polygonal line drawing (relative coordinate specification)
	PLINE Polygonal line with line-type specification	Draws polygonal line with line type (pattern) from start coordinates through nodal coordinates.
	PLINE	Pattern-reference polygonal line drawing (absolute coordinate specification)
	RPLINE	Pattern-reference polygonal line drawing (relative coordinate specification)
Work surface drawing	FTRAP Trapezoid paint	Performs binary EOR painting of trapezoid with left side parallel to Y-axis.
	FTRAP	Binary EOR trapezoid fill (absolute coordinate specification)
	RFTRAP	Binary EOR trapezoid fill (relative coordinate specification)
	CLRW Rectangle zero-clear	Performs zero-painting of rectangle with diagonal designated by two coordinate points.
Work line drawing	LINEW Polygonal line	Draws solid polygonal line from start coordinates through nodal coordinates.
	LINEW	Binary polygonal line drawing (absolute coordinate specification)
	RLINEW	Binary polygonal line drawing (relative coordinate specification)

Table 4-1 **Drawing Commands (cont)**

Type	Command Name	Function
Register setting	MOVE	Current pointer setting (absolute coordinate specification)
	RMOVE	Current pointer setting (relative coordinate specification)
	LCOFS	Local offset value setting (absolute coordinate specification)
	RLCOFS	Local offset value setting (relative coordinate specification)
	SCLIP	Sets rectangle with diagonal designated by origin and specified coordinate point as clipping area.
	UCLIP	Sets rectangle with diagonal designated by two coordinate points as clipping area.
Sequence control	JUMP	Command sequence jump (branch)
	GOSUB	Subroutine call (branch)
	RET	Subroutine return
	NOP3	No operation: no processing executed.
Drawing end	TRAP	Ends drawing processing and generates CPU interrupt.

The following items (basic functions) are available for drawing command parameter specifications. The items that can be specified depend on the command.

- Rendering coordinate system

  The position of all coordinate system origins, excluding binary source coordinates, can be assigned to the pixel position at which x = 0 and y is a multiple of 128, counting from the start of the unified graphics memory (UGM) in pixel units.

  — Rendering coordinates (2-dimensional coordinate system)

  These are the coordinates at which the Q2i performs drawing. For rendering coordinates, a 512-pixel × 512-pixel or 1024-pixel × 1024-pixel coordinate system can be selected according to the size of the UGM.

  — Multi-valued source coordinates (2-dimensional coordinate system)

  These are the coordinates for natural image and other color map data. A maximum of 1024 × 1024 size positive coordinates can be used. The size of the coordinates that can be used is determined by the size of the UGM. Also, the maximum number of colors that can be handled with multi-valued source coordinates is normally the same as for rendering coordinates. Multi-valued source coordinates can be superimposed on rendering coordinates.

- Work coordinates (2-dimensional coordinate system)

  These are the coordinates for managing graphics used when rendering attribute work specification is performed. Work coordinates are managed by the Q2i so that there is a 1-to-1 correspondence for each rendering coordinate pixel. Clipping processing is also handled in the same way as for rendering coordinates.

- Binary source coordinates (1-dimensional coordinate system)

  These are coordinates for storing character patterns and line patterns. Character patterns and line patterns are stored in order from any address in the UGM, and pattern sizes are managed with a 2-dimensional coordinate system using command parameters TDX and TDY.

- Rendering reference data
  - In drawing that references the transfer source, the referenced data format may be multi-valued source data or binary source data, defined as individual multi-valued source coordinates and binary source coordinates.
  - In drawing that does not reference the transfer source, specified color data may be referenced.
  - With a POLYGON4 type command, work data may be referenced.
- Rendering attributes
  - Ten kinds of attribute specifications can be made: work (WORK), clipping (CLIP), transparent (TRNS), source style (STYL), net drawing (NET), source half (HALF), even/odd select (EOS), bold line drawing (FWUL, W2UL, FWDR, W2DR), and source linear address (LNi) and drawing by two pixels (FST). The attributes that can be specified depend on the command.

4.2 Command Fetching

The Q2i carries out drawing operations while performing fetches from the display list stored in the UGM. The display list consist of a number of linked Q2i drawing commands.

The Q2i performs sequential fetches in low-to-high address order, starting at the address set in the display list start address register (DLSAR). The fetch address can be changed midway, using a JUMP or GOSUB command. Q2i fetching can be terminated by placing a TRAP command at the end of the display list.

The Q2i has a 16-word dedicated command buffer, and an equivalent area of the UGM is accessed at one time. When processing of the commands in this buffer is completed, another command fetch is performed.

If the commands include a JUMP, GOSUB, or other command that changes the flow, the Q2i continues its fetch operation from the address indicated by that command.

Figure 4-1 shows an example of the display list.

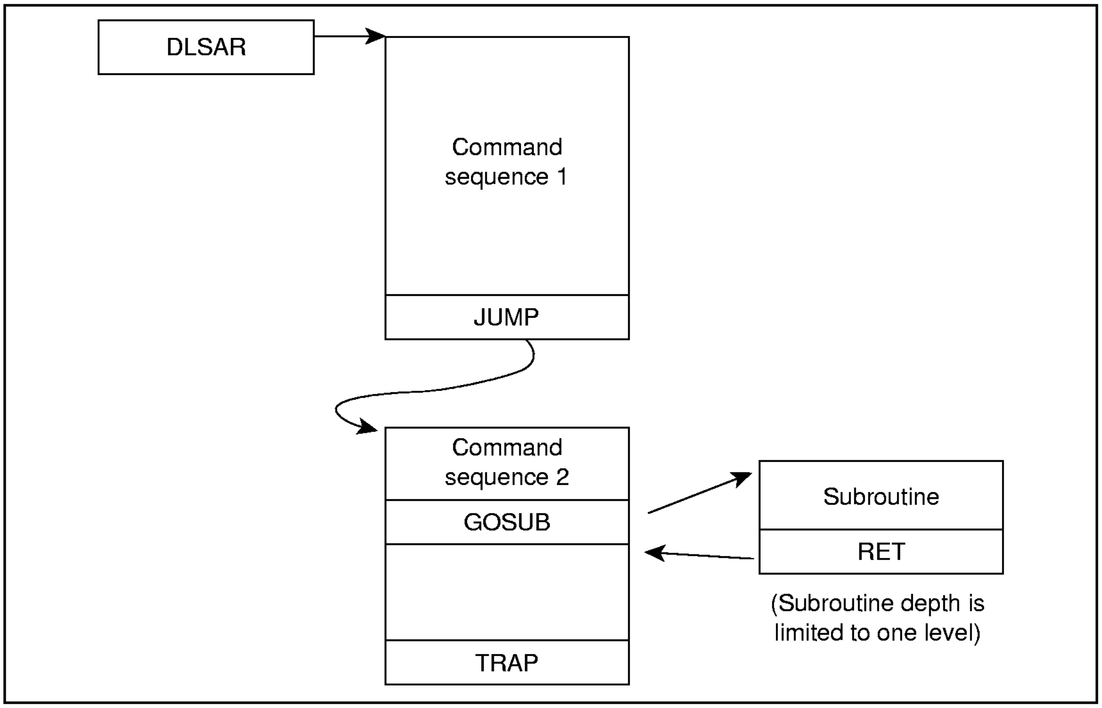


Figure 4-1 Display List Example

4.3 Basic Functions

4.3.1 Rendering Coordinate Systems

The Q2i controls three 2-dimensional coordinate systems, for rendering coordinates, 8-bit/pixel or 16-bit/pixel (multi-valued) source coordinates, and work coordinates, and one 1-dimensional coordinate system, for 1-bit/pixel (binary) source coordinates.

Rendering Coordinates: The size of the coordinate system is fixed, as shown in figure 4-2. The correspondence to the frame buffers is also fixed, but depends on the installed memory capacity and screen size. Make the appropriate selection with the rendering mode register. In an area other than one containing a frame buffer, although drawing operations are performed, nothing is written.

When drawing is performed using the LCOFS command, coordinates after addition of the offset (XO, YO) set by the LCOFS command must be within the range shown as follows.

$$-1020 \leq X + XO \leq 1020$$
$$-1020 \leq Y + YO \leq 1020$$

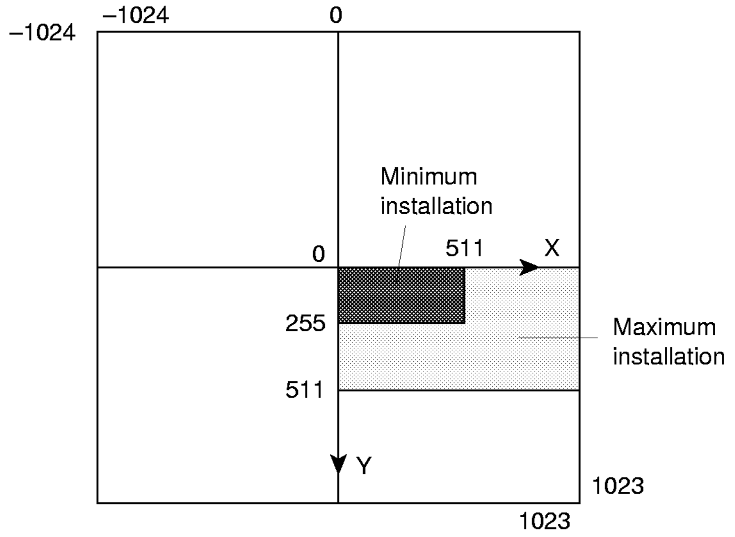


Figure 4-2 Rendering Coordinates

Multi-Valued Source Coordinates: The coordinate origin is specified by the multi-valued source area start address. The maximum coordinate system size is represented by 1024 × 1024 positive coordinates, as shown in figure 4-3, but the size depends on the installed memory capacity, screen size, and multi-valued source area start address. Depending on the multi-valued source start address, this coordinate system may entirely or partially overlap another coordinate system.

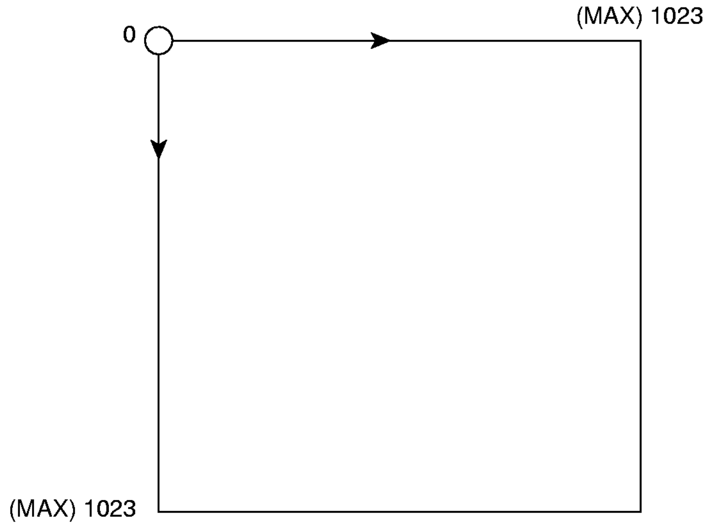


Figure 4-3 Multi-Valued Source Coordinates (LNi = 0)

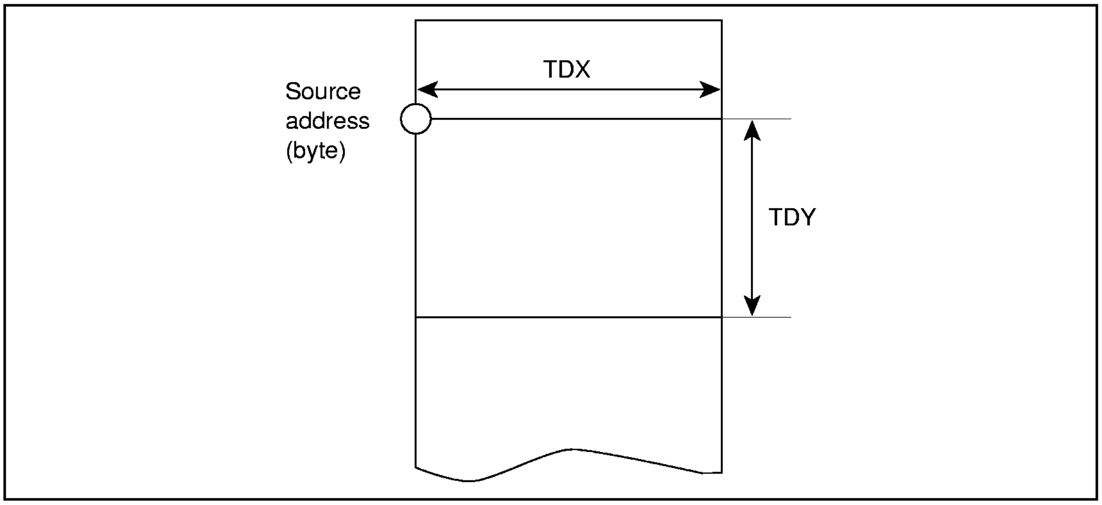


Figure 4-4 Multi-Valued Source Coordinates with LNi = 1 Specified (Linear Address)

Binary Source Coordinates: The binary (1-bit/pixel) source coordinate system is mapped directly onto 1-dimensional memory space. Any area and location can be used, and this coordinate system can overlap display list space. However, the start address of a source figure is always a byte address. The size of the figure is specified by POLYGON4B command parameters TDX and TDY. The TDX setting can only be made in multiples of 8 pixels.

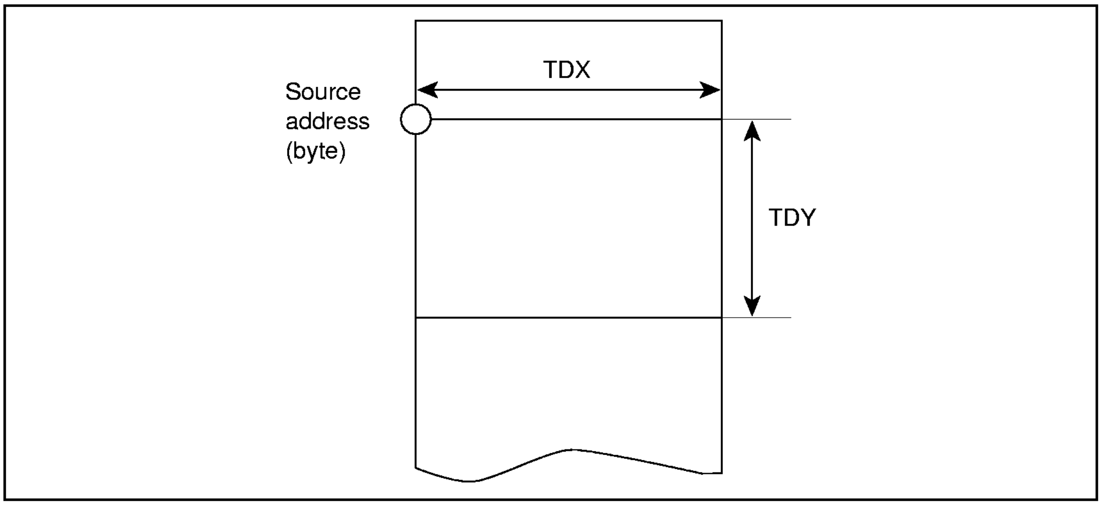


Figure 4-5 Binary Source Coordinates

Binary Work Coordinate System: The work coordinate system corresponds on a one-to-one basis to the rendering coordinate system. Therefore, clipping is also handled in the same way as for the rendering coordinate system.

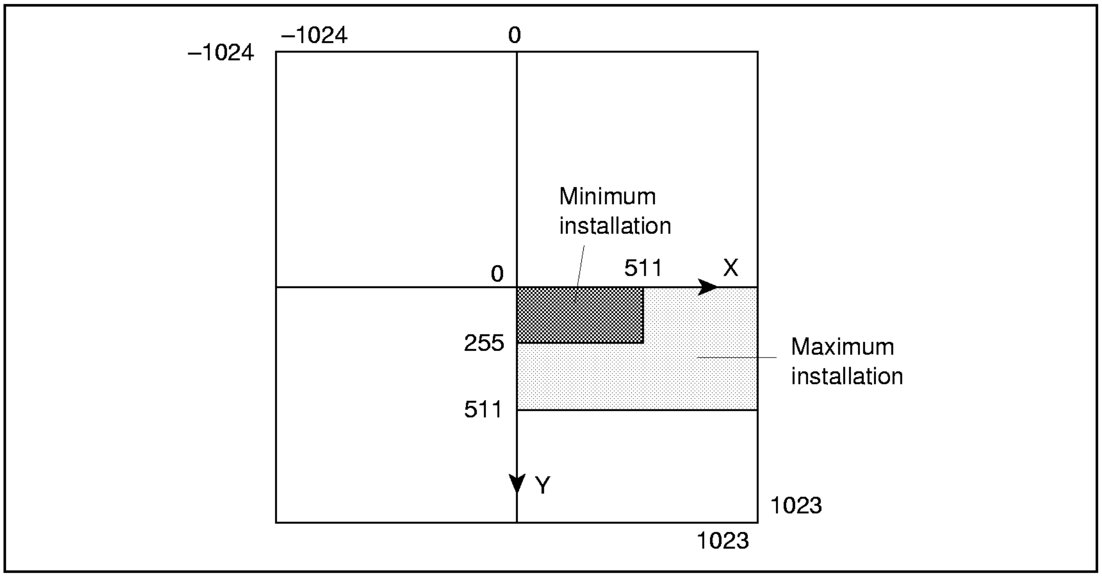
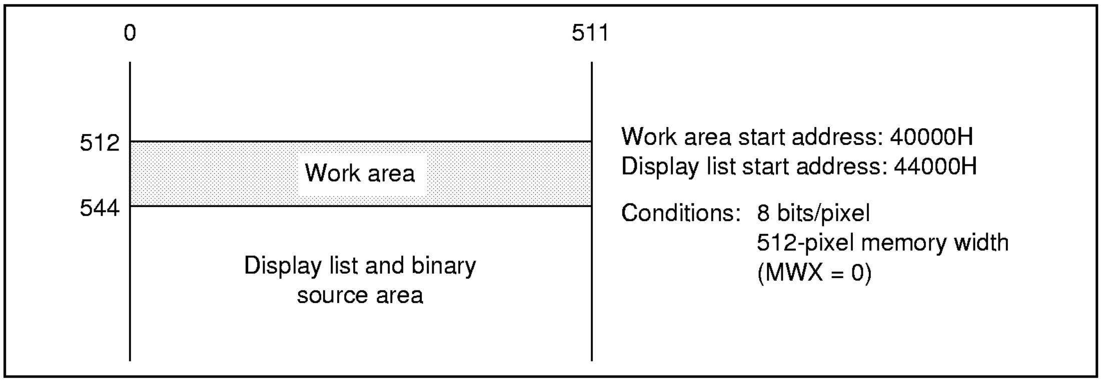


Figure 4-6 Work Coordinate System

Relationship between Binary Work Coordinates and Addresses: Work coordinates are linear coordinates that start from the work area start address. Work coordinates comprise 2-dimensional coordinates reflected at each pixel (512 or 1024 pixels) specified by the MWX bit in the rendering mode register (REMR). Examples are shown in figures 4-7 and 4-8.

The memory capacity required for the binary work area is (the number of pixels specified by the MWX bit) × (the number of pixels along the Y-axis)/8 [bytes].



Work area start address: 40000H
Display list start address: 44000H

Conditions: 8 bits/pixel
512-pixel memory width
(MWX = 0)

Figure 4-7 Example of Relationship between Work Coordinates and Physical Addresses

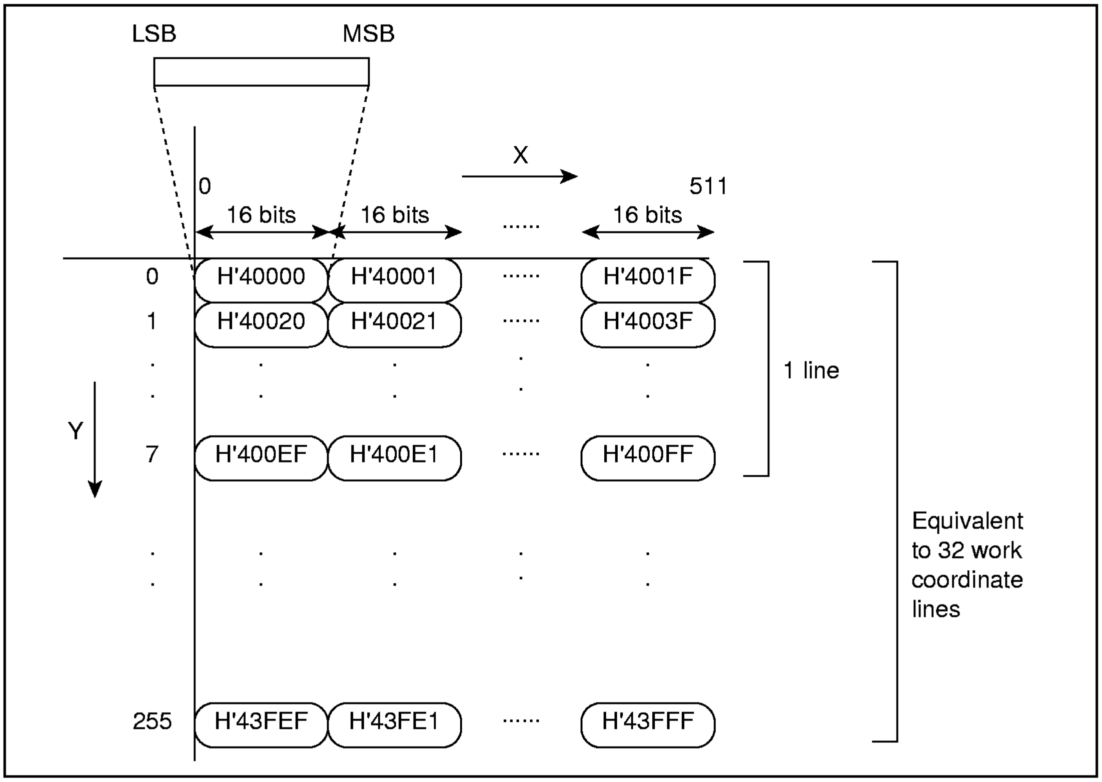


Figure 4-8 Relationship between Work Coordinates and Addresses

4.3.2 Rendering Reference Data

Q2i drawing operations can be broadly divided into those that reference the transfer source and those that do not. Drawing commands that reference the transfer source are POLYGON4A, POLYGON4B, PLINE, and RPLINE. Drawing commands that do not reference the transfer source are POLYGON4C, LINE, RLINE, FTRAP, RFTRAP, CLRW, LINEW, and RLINEW.

With drawing operations that reference the transfer source, there are two reference data formats: multi-valued source data and binary source data.

Of the commands that do not reference the transfer source, POLYGON4C, LINE, RLINE, LINEW, and RLINEW reference the specified color data.

With POLYGON4 commands, it is possible to reference a combination of multi-valued source data and binary source data, binary source data and binary work data, or specified color data and binary work data (figure 4-9).

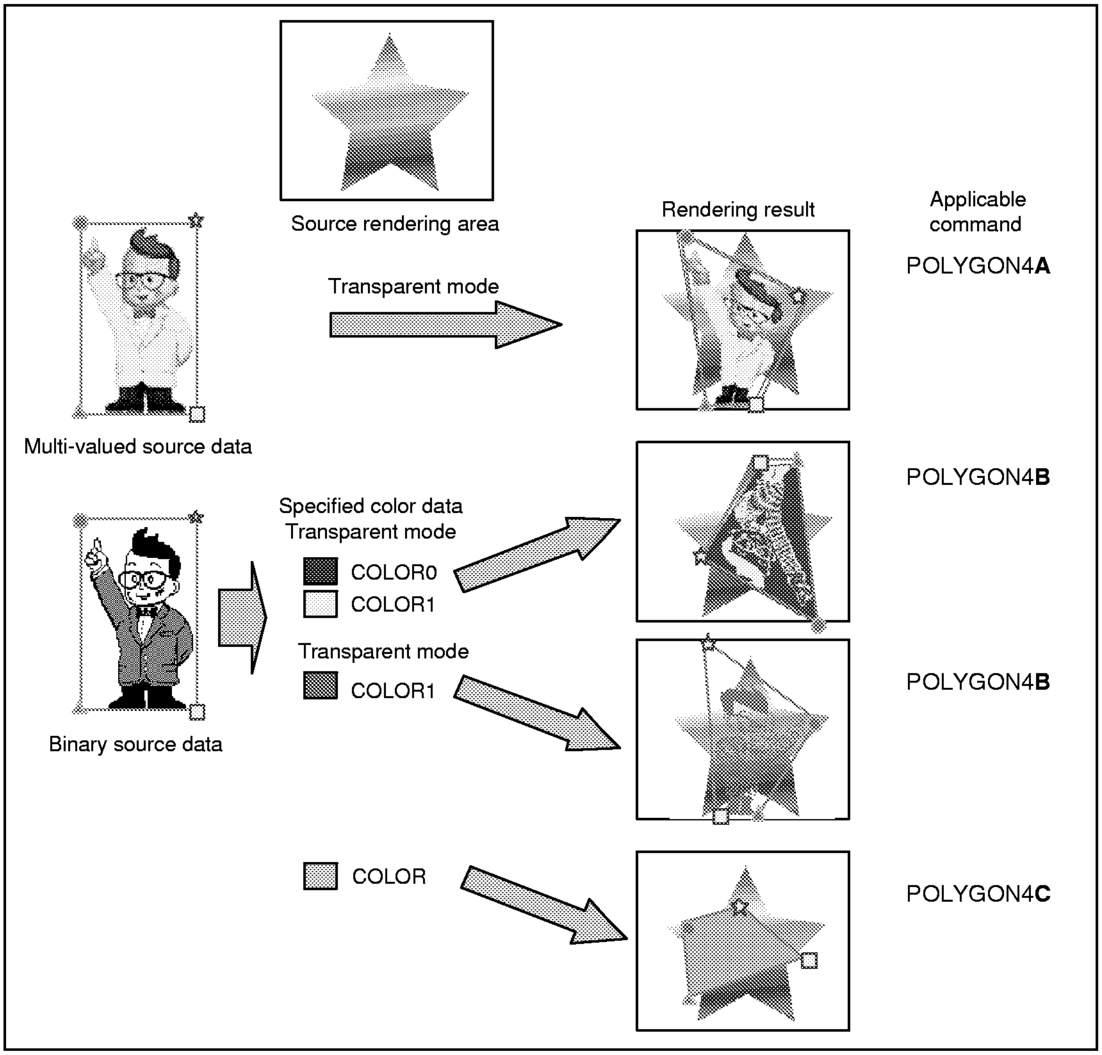


Figure 4-9 Example of POLYGON4 Transfer Data Combinations

Multi-Valued Source Data: Multi-valued source data is defined as multi-valued source coordinates (2-dimensional coordinates).

However, the horizontal width (TDX) is specified in multiples of 8 pixels. The configuration of multi-valued source data is shown in figure 4-10.

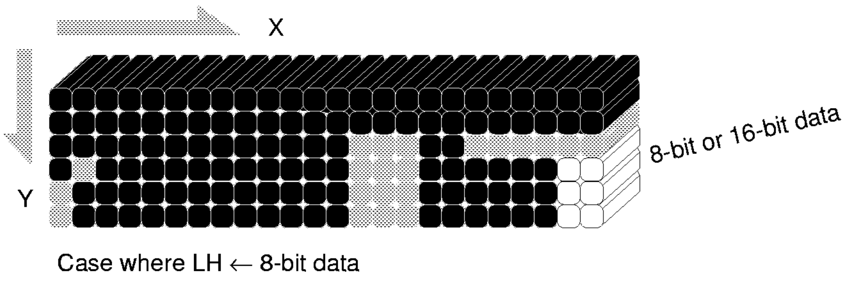
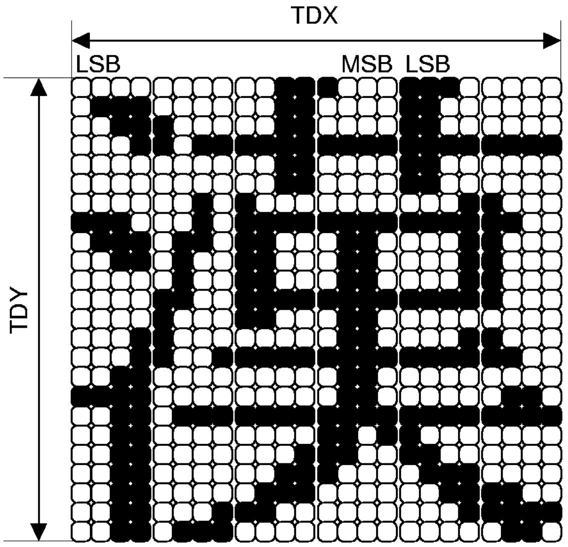


Figure 4-10 Multi-Valued Source Data Configuration

Binary Source Data: Binary source data is arranged in linear fashion in the binary source area in the UGM, and is managed as 2-dimensional coordinates (binary source coordinates) by TDX and TDY in the POLYGON4B command. The left-hand screen pixel must be located at the LSB of the binary source data when the binary source data area is viewed from the Q2i.

However, the horizontal width (TDX) is specified in multiples of 8 pixels (1-byte units). An example of binary source data is shown in figure 4-11.

A binary source is used for the definition of character data and line-type data. When drawing, 0s are converted to COLOR0 data, and 1s to COLOR1 data (in transparent mode, only 1s are converted to COLOR1 data for drawing).



Example of kanji font as binary source
TDX=24, TDY=24
Data:
H'1C00, H'0E07, H'430C,
H'0C1C, H'D8E3, H'FFFF,
H'0C00, H'0003, H'030C,
H'0140, H'4718, H'3FFF,
H'636E, H'2C18, H'1863,
H'6320, H'3018, H'1FFF,
H'6310, H'1800, H'1860,
H'FF98, H'0C3F, H'0060,
H'600F, H'EC60, H'FFFF,
H'B00C, H'0C01, H'0338,
H'1C0C, H'0C0E, H'3C0E,
H'038C, H'ECF8, H'6000

Figure 4-11 Example of Kanji Font as Binary Source (TDX = 24, TDY = 24)

Binary Work Data: Binary work data is defined as binary work coordinates (2-dimensional coordinates). Work data is used to implement polygon painting. Polygon outline data is created with the FTRAP command, etc., and the created figure data is used to delineate the rendering figure. For example, if the POLYGON4C command is used jointly for work, the work area polygon can be drawn in the rendering area with the specified color value. The configuration of binary work data is shown in figure 4-12.

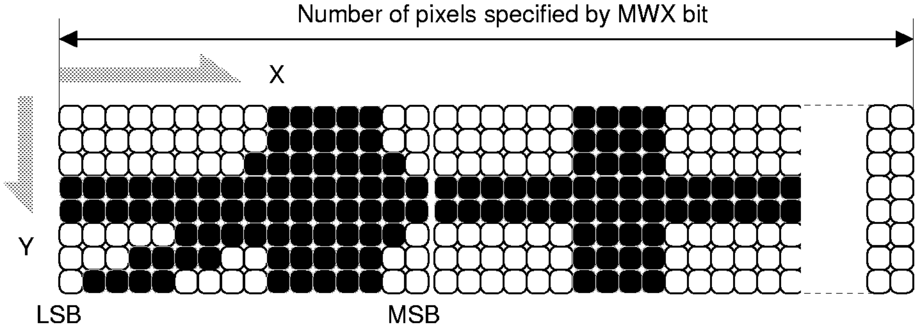


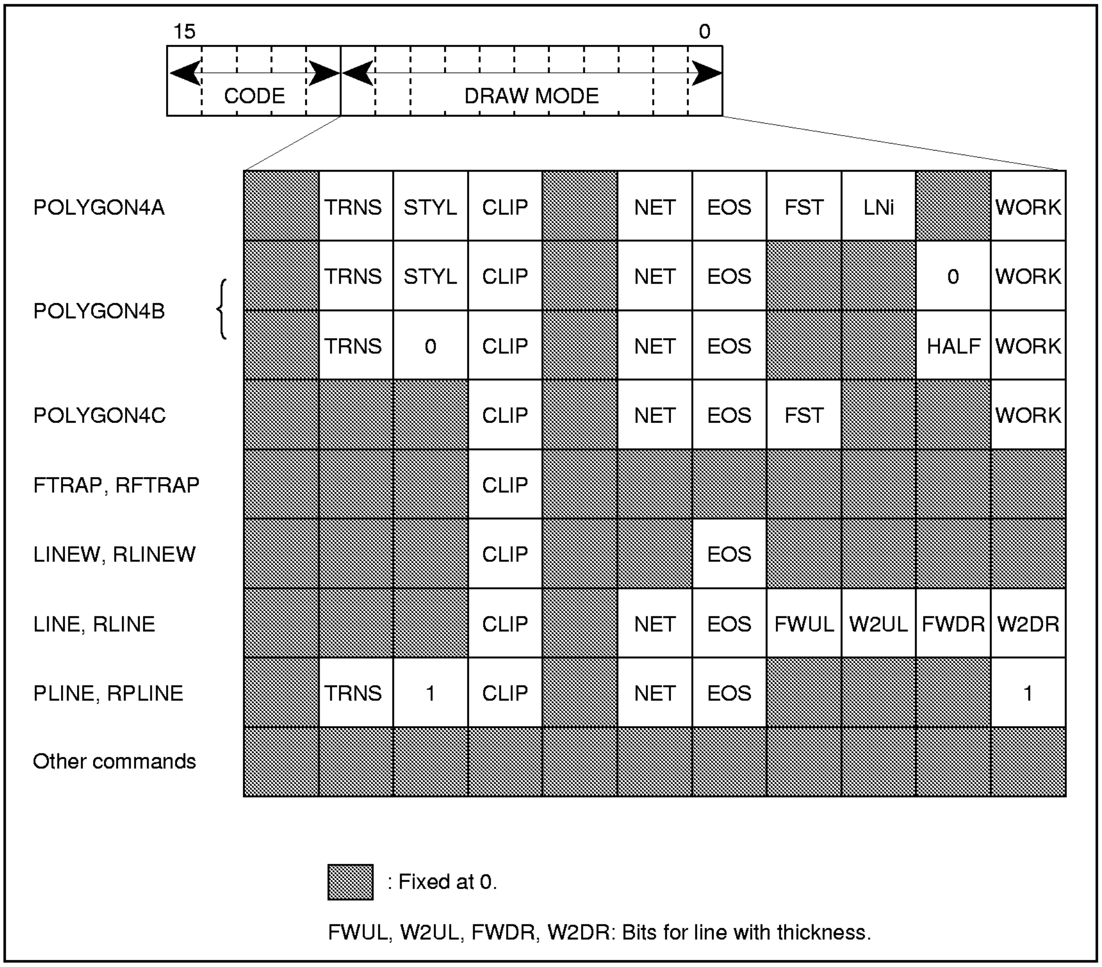
Figure 4-12 Binary Source Data Configuration

Specified Color Data: Specified color data is defined directly in drawing parameters COLOR, COLOR0, COLOR1, LINE COLOR0, and LINE COLOR1. When the Q2i is used for 8-bit/pixel operation, the same color palette number is defined in the upper 8 bits and lower 8 bits by the drawing parameter color specification. When the Q2i is used for 16-bit/pixel operation, the R, G, and B values are defined directly by the drawing parameter color specification.

However, with LINEW and RLINEW, 0 or 1 is defined in the rendering attribute EOS bit.

4.3.3 Rendering Attributes

With the Q2i, seven rendering attributes can be specified. The rendering attributes are embedded in the commands, and can be specified on an individual command basis. Figure 4-13 shows the bit arrangement for rendering attributes.



POLYGON4A	▨	TRNS	STYL	CLIP	▨	NET	EOS	FST	LNi	▨	WORK
POLYGON4B	▨	TRNS	STYL	CLIP	▨	NET	EOS	▨	▨	0	WORK
	▨	TRNS	0	CLIP	▨	NET	EOS	▨	▨	HALF	WORK
POLYGON4C	▨	▨	▨	CLIP	▨	NET	EOS	FST	▨	▨	WORK
FTRAP, RFTRAP	▨	▨	▨	CLIP	▨	▨	▨	▨	▨	▨	▨
LINEW, RLINEW	▨	▨	▨	CLIP	▨	▨	EOS	▨	▨	▨	▨
LINE, RLINE	▨	▨	▨	CLIP	▨	NET	EOS	FWUL	W2UL	FWDR	W2DR
PLINE, RPLINE	▨	TRNS	1	CLIP	▨	NET	EOS	▨	▨	▨	1
Other commands	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

▨ : Fixed at 0.

FWUL, W2UL, FWDR, W2DR: Bits for line with thickness.

Figure 4-13 Rendering Attribute Bit Arrangement

Transparency Specification (TRNS): When color expansion of binary source data is performed, transparency or non-transparency can be selected on an individual drawing command basis with the TRNS bit. When transparency is selected, a 0 in the binary source data is transparent and a 1 has the value of the COLOR1 parameter. When non-transparency is selected, a binary data 0 has the value of the COLOR0 parameter, and a 1 has the value of the COLOR1 parameter. With multi-valued source data, “all-0” data becomes a transparent color, and those pixels are not drawn. The transparency specification can be used with the POLYGON4A, POLYGON4B, PLINE, and RPLINE commands; in other commands, the TRNS bit should be cleared to 0.

Source Style Specification (STYL): When drawing a rectangle, the STYL bit can be used to select, on an individual drawing command basis, whether the source data is to be enlarged or reduced, or referenced repeatedly. If no style specification is made, the source data is enlarged or reduced in proportion to the size of the rendering area. When a style specification is made, the source data is referenced repeatedly in proportion to the size of the rendering area. This attribute is therefore used when drawing repeated patterns such as hatch patterns. The source style specification can be used with the POLYGON4A, POLYGON4B, PLINE, and RPLINE commands; in other commands, the STYL bit should be cleared to 0. If a source style specification is made, do not make a source half specification.

An example of a source style specification is shown in figure 4-14.

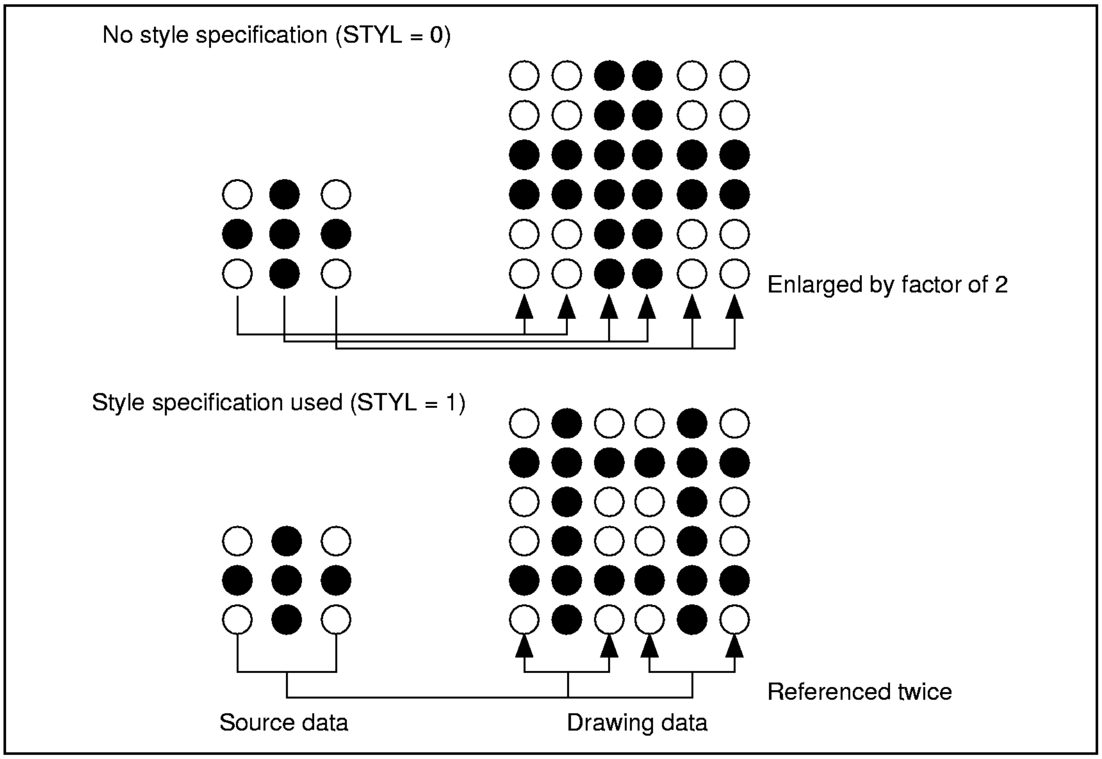

No style specification (STYL = 0)
Enlarged by factor of 2
Style specification used (STYL = 1)
Referenced twice
Source data
Drawing data

Figure 4-14 Example of Source Style Specification

Clipping Specification (CLIP): The Q2i can perform clipping area management. There are two kinds of clipping area: a system clipping area designated by the SCLIP command, and a user clipping area designated by the UCLIP command.

The system clipping area is a rectangular area for prescribing a valid drawing range as the display range when the Q2i performs double-buffering control. Thus the system clipping area is used to prohibit drawing outside the area. The system clipping area is valid when the CLIP bit is cleared to 0.

The user clipping area is valid when the CLIP bit is set to 1. When the user clipping area is made valid, the system clipping area becomes invalid. Thus, to prohibit drawing outside the system clipping area, ensure that the user clipping area fits inside the system clipping area.

In both the system clipping area and the user clipping area, drawing points are drawn even when on the clipping area boundary.

An example of a clipping specification is shown in figure 4-15.

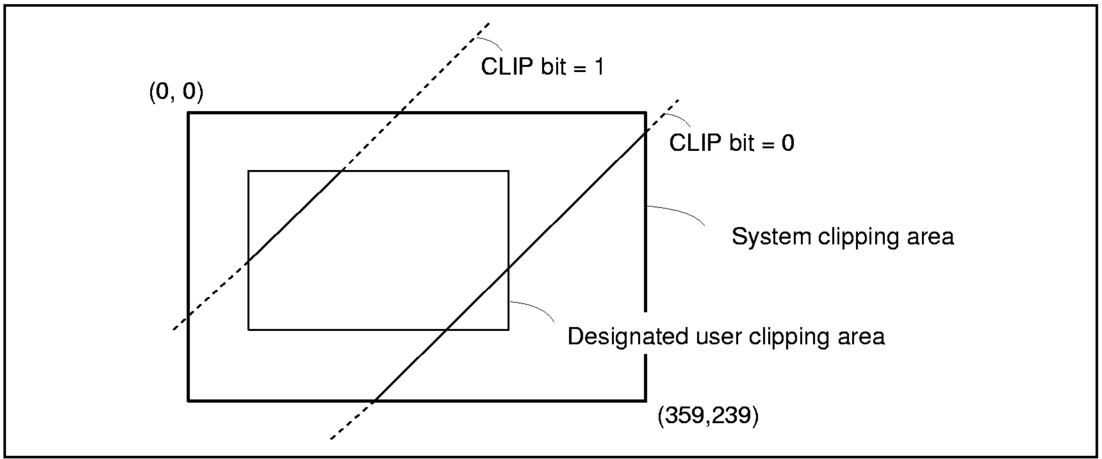


Figure 4-15 Example of Clipping Specification

Net Drawing Specification (NET): The NET bit can be used to select, on an individual drawing command basis, whether or not net drawing is to be performed. Net drawing is a function for drawing only pixels at coordinates for which the condition “rendering coordinates X + Y = EOS (0: even number, 1: odd number)” is true. For example, if EOS = 0, drawing will only be performed at coordinates Y = 0, X = 0, 2, 4, 6, 8, ..., Y = 1, X = 1, 3, 5, 7, 9, ...

This function enables the drawn figure and ground to be mutually semi-superimposed.

The net drawing specification can be used with the POLYGON4 commands, and the LINE, RLINE, PLINE, and RPLINE commands; in other commands the NET bit should be cleared to 0.

Even/Odd Select Specification (EOS): Even pixels are selected when EOS = 0, and odd pixels when EOS = 1.

The even/odd select specification is used together with the net specification or source half specification.

With the LINEW and RLINEW commands, drawing is performed at the work coordinates with 0 when EOS = 0, and with 1 when EOS = 1.

Examples of even/odd select specifications are shown in figure 4-16.

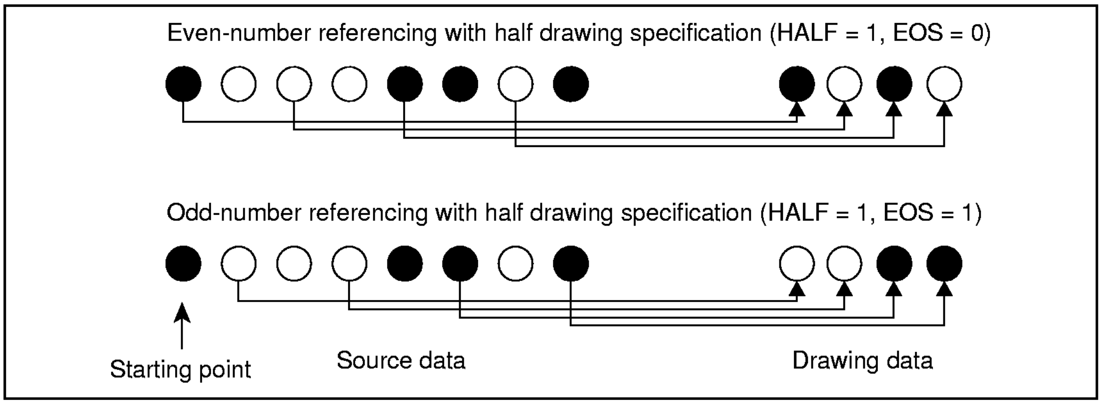

Even-number referencing with half drawing specification (HALF = 1, EOS = 0)
Odd-number referencing with half drawing specification (HALF = 1, EOS = 1)
Starting point
Source data
Drawing data

Figure 4-16 Examples of Even/Odd Select Specifications

Source Half Drawing Specification (HALF): The HALF bit can be used to select whether all or only half of the source data is to be referenced. When the source half drawing specification is selected, only EOS (0: even number, 1: odd number) data is referenced from the source starting point. Thus only half of the source data in the horizontal direction is referenced.

The source half drawing specification can only be used with the POLYGON4B (binary source) command; in other commands, the HALF bit should be cleared to 0. If a source half specification is made, do not make a source style specification.

Work Specification (WORK): When drawing is performed at rendering coordinates with POLYGON4 commands, the WORK bit can be used to select, on an individual drawing command basis, whether or not binary work data is to be referenced.

When binary work data referencing is selected, drawing is performed if the work data for the pixel corresponding to the rendering coordinates is 1, but not if the work data is 0. Drawing can thus be performed at the rendering coordinates in the same form as the figure drawn at the work coordinates. Drawing at work coordinates can be performed by either of two methods: drawing by means of the FTRAP command or drawing by the SuperH. Ensure that UGM drawing access by means of the FTRAP command and UGM drawing access by the SuperH are not performed simultaneously. The commands that can be used are POLYGON4A, POLYGON4B, and POLYGON4C. In other commands, this specification is invalid.

With the PLINE and RPLINE commands, this attribute is specified but work references are not performed. For other commands, the work specification bit should be cleared to 0.

Bold Line Drawing Specification: Taking individual line segments of a polygonal line specified by parameters as reference lines, the FWUL bit and FWDR bit, respectively, can be used to select, for the upper-left direction and lower-right direction independently, whether or not the reference lines are to be made bold lines. The width of a bold line can be selected from line widths 1 to 5 by a combination of bits W2UL and W2DR.

This function is valid for each segment of a polygonal line. Using the segment line main scanning axes, lines with the same slope in the up (left) and down (right) directions, and of the same length, are drawn repeatedly. Therefore, the shape of the segment linkage parts is not considered. This function can be used with the LINE and RLINE commands. In other commands, the FWUL, W2UL, FWDR, and W2DR bits should all be cleared to 0.

Table 4-2 Bold Line Drawing Settings

FWUL	W2UL	FWDR	W2DR	Line Width (Direction, Magnification)
0	0	0	0	1 (no magnification)
			1	1 (no magnification)
		1	0	2 (lower right 1)
			1	3 (lower right 2)
	1	0	0	1 (no magnification)
			1	1 (no magnification)
		1	0	2 (lower right 1)
			1	3 (lower right 2)
1	0	0	0	2 (upper left 1)
			1	2 (upper left 1)
		1	0	3 (upper left 1, lower right 1)
			1	3 (upper left 1, lower right 2)
	1	0	0	3 (upper left 2)
			1	3 (upper left 2)
		1	0	4 (upper left 2, lower right 1)
			1	5 (upper left 2, lower right 2)

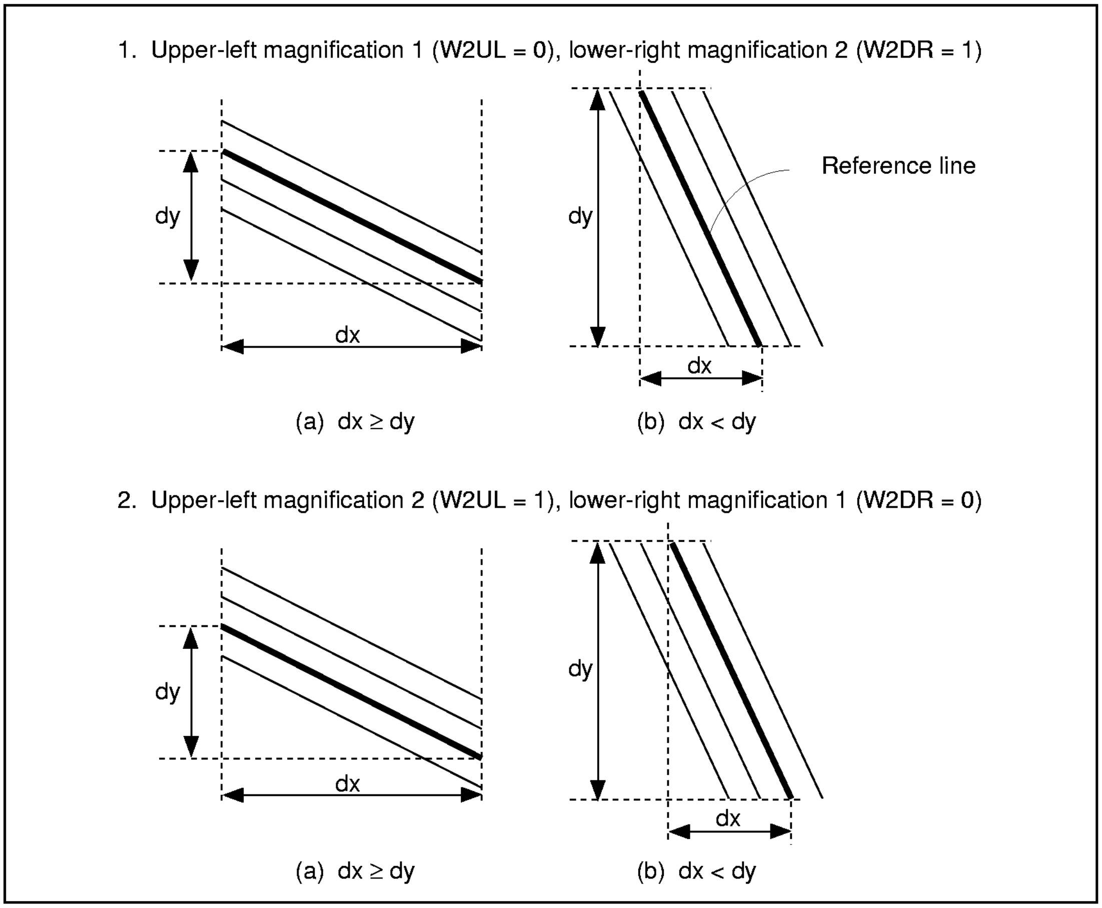

1. Upper-left magnification 1 (W2UL = 0), lower-right magnification 2 (W2DR = 1)
dy
dx
dy
dx
Reference line
(a) dx ≥ dy
(b) dx < dy
2. Upper-left magnification 2 (W2UL = 1), lower-right magnification 1 (W2DR = 0)
dy
dx
dy
dx
(a) dx ≥ dy
(b) dx < dy

Figure 4-17 Examples of Bold Line Drawing (Line Width 4 Drawing)

Source Address Linear Specification (LNi): Use of a 2-dimensional virtual address or a linear address as the source address can be selected, on an individual drawing command basis, by means of the LNi bit. To use a linear address, set this bit to 1.

This function can be used with the POLYGON4A command. In other commands, the LNi bit should be cleared to 0. For details of command operation, see section 4.4.1, POLYGON4A.

2-Pixel-Unit Processing (FST): Whether or not 2-pixel-unit processing is performed can be specified for individual drawing commands by means of the FST bit. To perform 2-pixel-unit processing, set the FST bit to 1. In this case, no other drawing attributes except CLIP can be used. This function can be used with the POLYGON4A and POLYGON4C commands. In other commands, the FST bit should be cleared to 0.

For a rectangle in which the source and destination are the same size, make settings so that TXS = even number, DX1 = DX4 = even number, DX2 = DX3 = odd number, DY1 = DY2, DY3 = DY4, and DX2 – DX1 = 16n – 1 (where n is a natural number).

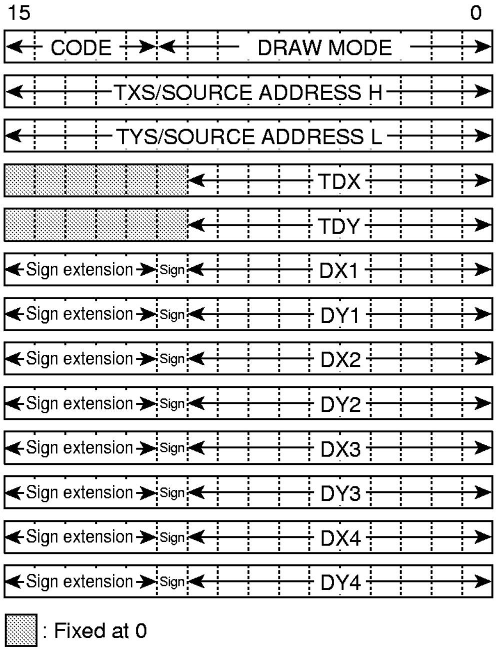
4.4 Drawing Commands

4.4.1 POLYGON4A

Function

Performs any four-vertex drawing while referencing a multi-valued (8- or 16-bit/pixel) source.

Command Format



1. Code
   B'00000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
O	×	Δ	×	O	×

Drawing Modes										
Reserved	TRNS	STYL	CLIP	Reserved	NET	EOS	FST	LNi	Reserved	WORK
Fixed at 0	O	O	O	Fixed at 0	O	O	O	O	Fixed at 0	O

O : Can be used
Δ : Referenced depending on mode (valid when WORK = 1)
× : Cannot be used

3. Command Parameters
   TXS, TYS: Source starting point
   SOURCE ADDRESS H: Source start upper address
   SOURCE ADDRESS L: Source start lower address
   TDX, TDY: Source size
   DXn, DYn (n = 1 to 4): Rendering coordinates, work coordinates

Description

Transfers multi-valued (8- or 16-bit/pixel) source data to any quadrilateral rendering coordinates. The source is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps.

When repeated source referencing is selected as a rendering attribute (STYL = 1), the source is not enlarged or reduced, but is referenced repeatedly.

When work referencing is selected as a rendering attribute (WORK = 1), transfer is performed while referencing work area data for the same coordinates as the rendering coordinates.

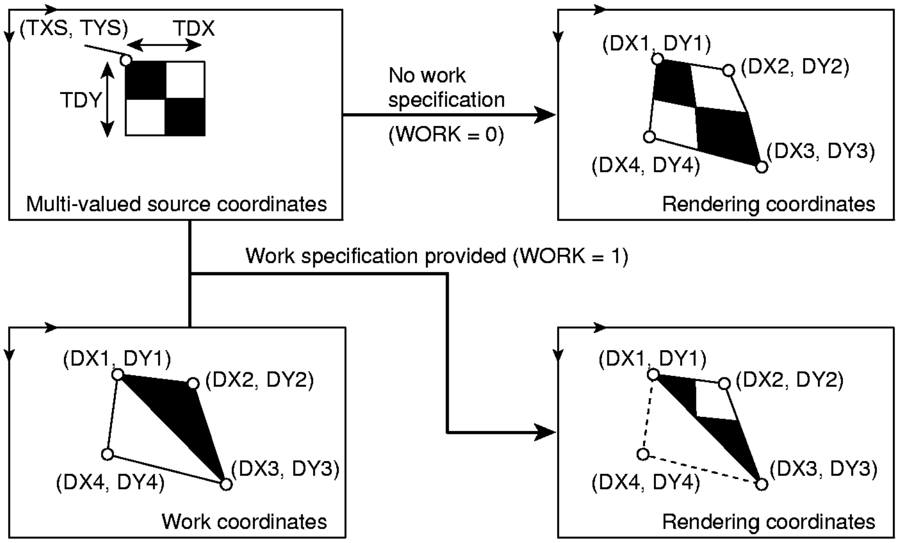
The multi-valued source is referenced taking 8 pixels in the multi-valued source coordinate X-direction as one unit, and enlargement or compression is carried out. The TDX value must therefore be specified in 8-pixel units in order for multi-valued source referencing to be performed normally.

Setting the LNi rendering attribute to 1 in 8-bit/pixel mode enables the linear address space in the UGM to be treated as multi-valued source coordinate space.

When LNi = 1, set the upper bits of the source address in SOURCE ADDRESS H, and the lower bits in SOURCE ADDRESS L.

When LNi = 0, make TXS and TYS settings in pixel units.

Example

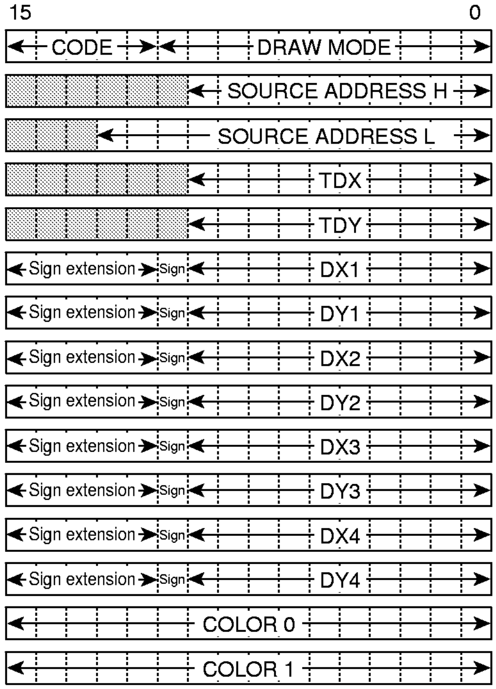


4.4.2 POLYGON4B

Function

Performs any four-vertex drawing while referencing a binary (1-bit/pixel) source.

Command Format



▨ : Fixed at 0

1. Code

   B'00001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	O	Δ	×	O	×

Drawing Modes										
Reserved	TRNS	STYL	CLIP	Reserved	NET	EOS	Reserved		HALF	WORK
Fixed at 0	O	◊	O	Fixed at 0	O	O	Fixed at 0	Fixed at 0	◊	O

O : Can be used
Δ : Referenced depending on mode (valid when WORK = 1)
× : Cannot be used
◊ : Cannot be specified simultaneously

3. Command Parameters

   SOURCE ADDRESS H: 1-bit/pixel source start upper address
   SOURCE ADDRESS L: 1-bit/pixel source start lower address
   TDX, TDY: Source size
   DXn, DYn (n = 1 to 4): Rendering coordinates, work coordinates
   COLOR0, COLOR1: 8 or 16-bit/pixel color specifications

Description

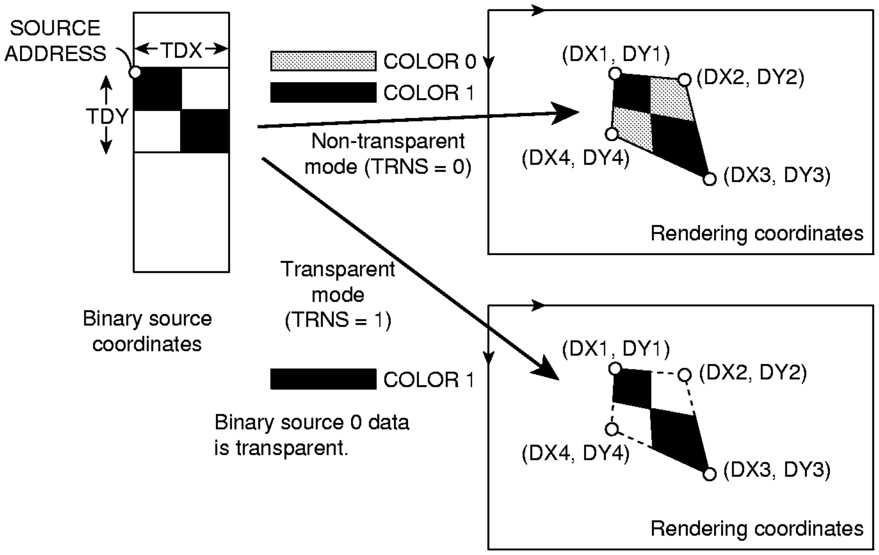
Transfers binary (1-bit/pixel) source data to any quadrilateral rendering area, expanding the data to the colors specified by parameters COLOR0 and COLOR1. The source is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps.

When repeated source referencing is selected as a rendering attribute (STYL = 1), the source is not enlarged or reduced, but is referenced repeatedly.

When work referencing is selected as a rendering attribute (WORK = 1), transfer is performed while referencing work area data for the same coordinates as the rendering coordinates.

The TDX value is specified in multiples of 8 pixels.

Example

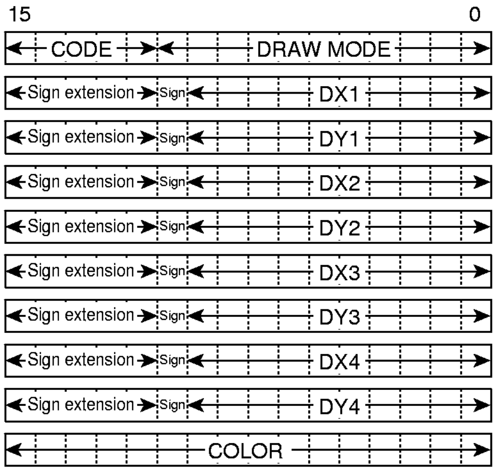


4.4.3 POLYGON4C

Function

Performs any four-vertex drawing with a monochrome specification.

Command Format



1. Code
   B'00010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	Δ	O	O	×

Drawing Modes											
Reserved			CLIP	Reserved	NET	EOS	FST	Reserved			WORK
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	O	O	O	Fixed at 0	Fixed at 0	Fixed at 0	O

O : Can be used
Δ : Referenced depending on mode (valid when WORK = 1)
× : Cannot be used

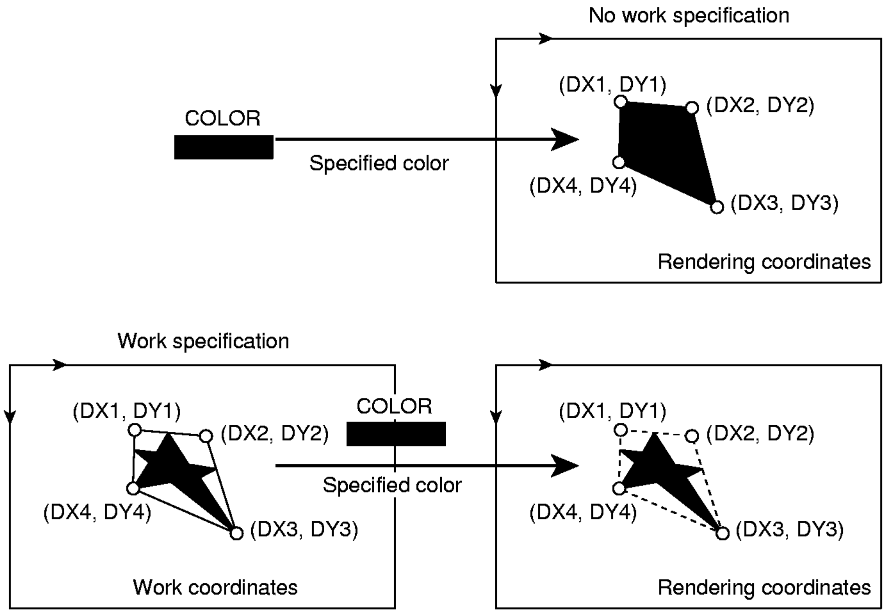
3. Command Parameters
   DXn, DYn (n = 1 to 4): Rendering coordinates, work coordinates
   COLOR: 8 or 16-bit/pixel color specification

Description

Draws any quadrilateral in the rendering area in the single color specified by the COLOR parameter.

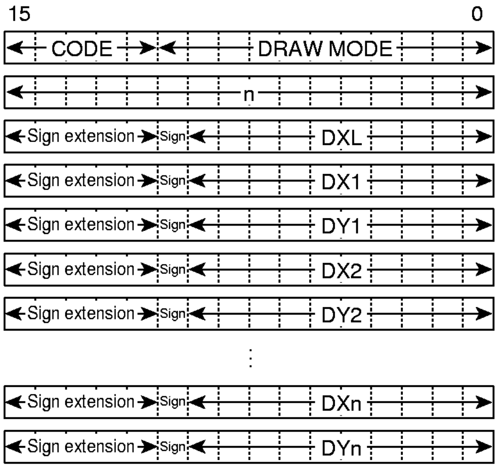
When work referencing is selected as a rendering attribute (WORK = 1), transfer is performed while referencing work area data for the same coordinates as the rendering coordinates.

Example



4.4.4 FTRAP

Command Format



1. Code
   B'01000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	O

Drawing Modes										
Reserved			CLIP	Reserved						
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

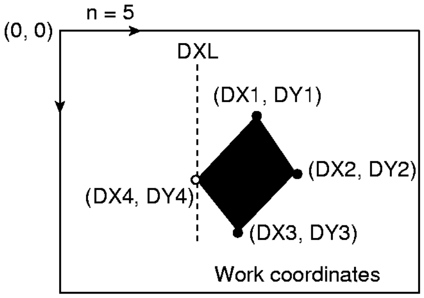
O : Can be used
× : Cannot be used

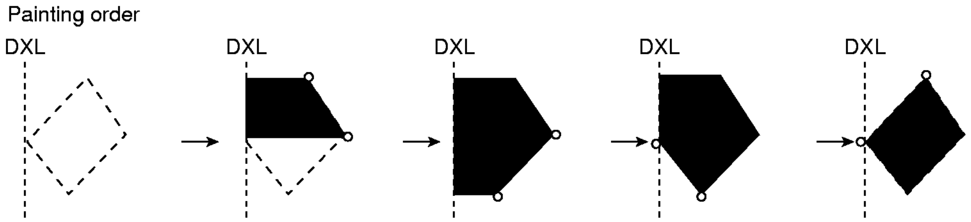
3. Command Parameters
   n (n = 2 to 65,535): Number of vertices
   DXL: Lefthand side coordinate
   DXn (n = 2 to 65,535): Absolute coordinate
   DYn (n = 2 to 65,535): Absolute coordinate

Description

Draws a polygon with n-1 vertices at work coordinates. Paints n-1 trapezoids with X = DXL as the left-hand side, and line segments (DX1, DY1) – (DX2, DY2), (DX2, DY2) – (DX3, DY3), ..., (DXn-1, DYn-1) – (DXn, DYn) as the right-hand sides, and with top and bottom bases parallel to the X-axis, at work coordinates using binary EOR. Bottom base drawing is not performed. Set the minimum value of DX1 to DXn as DXL.

Example





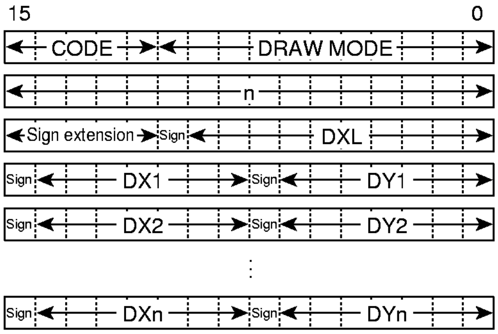
FTRAP Parameter List Order

n
DXL

DX1
DY1
DX2
DY2
DX3
DY3
DX4
DY4
DX1 } Add start point at end.
DY1

4.4.5 RFTRAP

Command Format



1. Code
   B'01001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	O

Drawing Modes										
Reserved			CLIP	Reserved						
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O : Can be used
× : Cannot be used

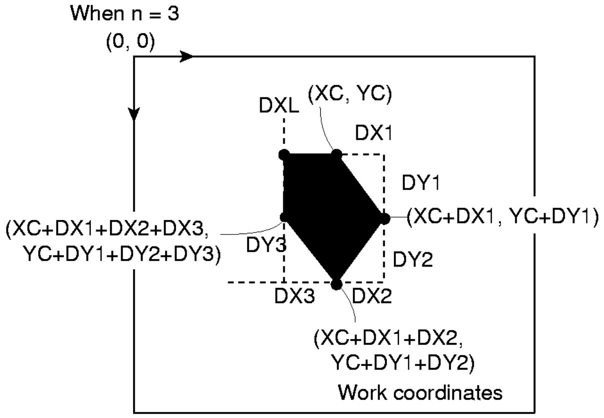
3. Command Parameters
   n (n = 1 to 65,535): Number of vertices
   DXL: Lefthand side coordinate
   DXn, DYn (n = 1 to 65,535): Relative coordinates

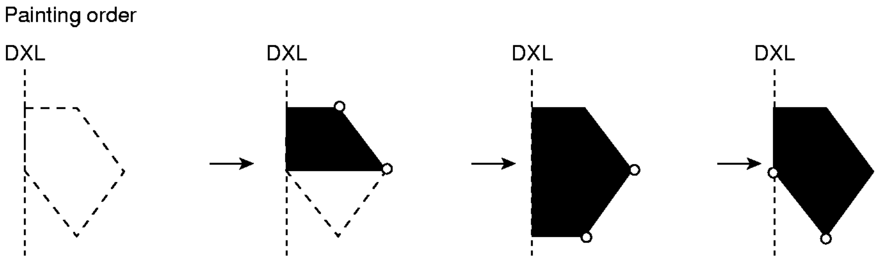
Description

Paints n trapezoids with X = DXL as the left-hand side, and line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer (XC, YC) as the right-hand sides, and with top and bottom bases parallel to the X-axis, at work coordinates using binary EOR. Bottom base drawing is not performed.

The final coordinate point is stored as the current pointer (XC, YC).

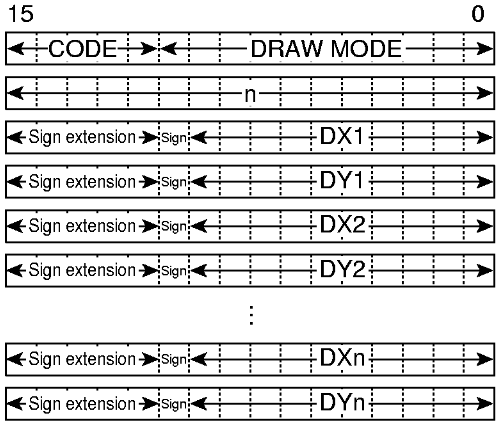
Example





4.4.6 LINEW

Command Format



1. Code
   B'01010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	∇	×	O

Drawing Modes											
Reserved			CLIP	Reserved		EOS	Reserved				
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O : Can be used
∇ : Can be used (EOS reference: specified color is binary EOS bit value)
× : Cannot be used

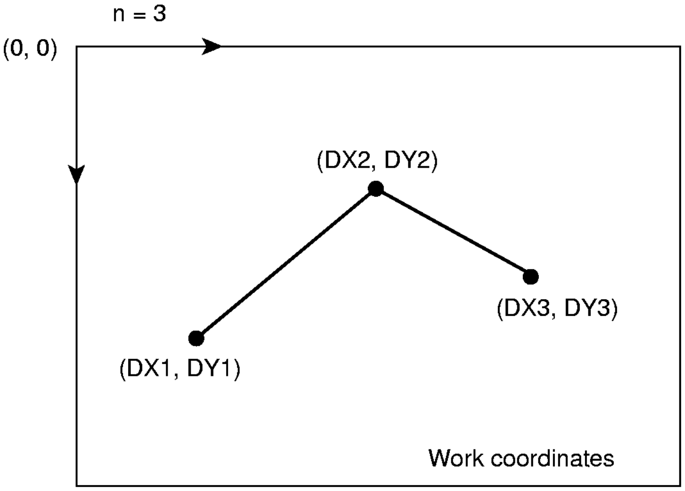
3. Command Parameters
   n (n = 2 to 65,535): Number of vertices
   DXn (n = 2 to 65,535): Absolute coordinate
   DYn (n = 2 to 65,535): Absolute coordinate

Description

Performs binary drawing at work coordinates of a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn). 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when EOS = 0, and at work coordinates with 1 when EOS = 1. (Used to draw the border of a polygon-paint figure at work coordinates.)

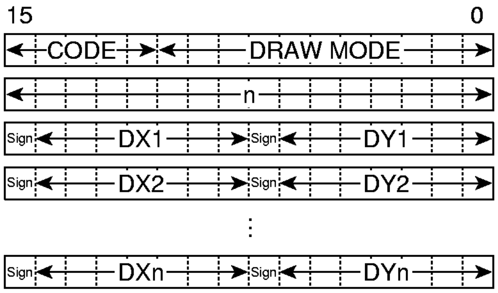
Note: 8-point drawing is used.

Example



4.4.7 RLINEW

Command Format



1. Code
   B'01011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	∇	×	O

Drawing Modes										
Reserved			CLIP	Reserved		EOS	Reserved			
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O : Can be used
∇ : Can be used (EOS reference: specified color is binary EOS bit value)
× : Cannot be used

3. Command Parameters
   n (n = 1 to 65,535): Number of vertices
   DXn, DYn (n = 1 to 65,535): Relative coordinates

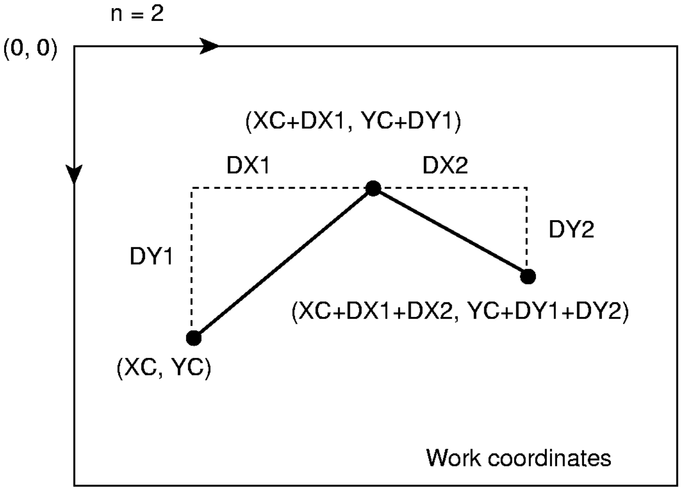
Description

Performs binary drawing at work coordinates of a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer (XC, YC). 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when EOS = 0, and at work coordinates with 1 when EOS = 1.

The final coordinate point is stored as the current pointer (XC, YC). (Used to draw the border of a polygon-paint figure at work coordinates.)

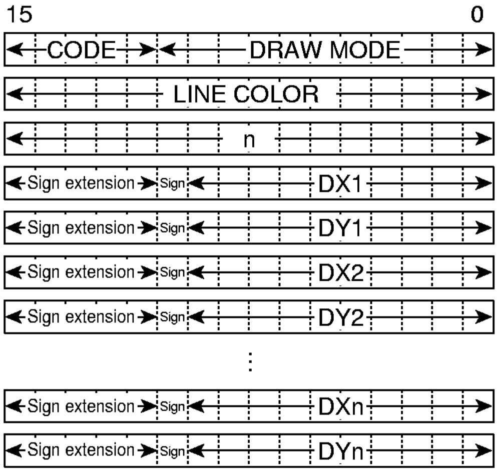
Note: 8-point drawing is used.

Example



4.4.8 LINE

Command Format



1. Code
   B'01100

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	O	O	×

Drawing Modes										
Reserved			CLIP	Reserved	NET	EOS	FWUL	W2UL	FWDR	W2DR
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	O	O	0000–1111			

O : Can be used

× : Cannot be used

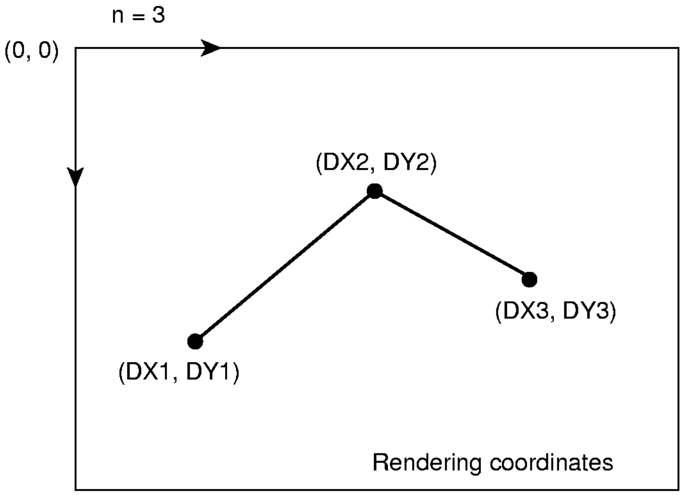
3. Command Parameters
   LINE COLOR0: 16-bit/pixel color specification
   n (n = 2 to 65,535): Number of vertices
   DXn (n = 2 to 65,535): Absolute coordinate
   DYn (n = 2 to 65,535): Absolute coordinate

Description

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn).

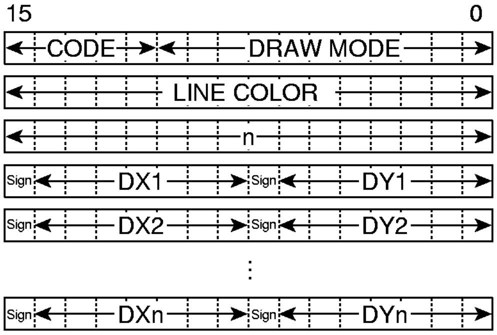
Note: 8-point drawing is used.

Example



4.4.9 RLINE

Command Format



1. Code
   B'01101

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	O	O	×

Drawing Modes										
Reserved			CLIP	Reserved	NET	EOS	FWUL	W2UL	FWDR	W2DR
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	O	O	0000–1111			

O : Can be used

× : Cannot be used

3. Command Parameters
   LINE COLOR: 16-bit/pixel color specification
   n (n = 1 to 65,535): Number of vertices
   DXn, DYn (n = 1 to 65,535): Relative coordinates

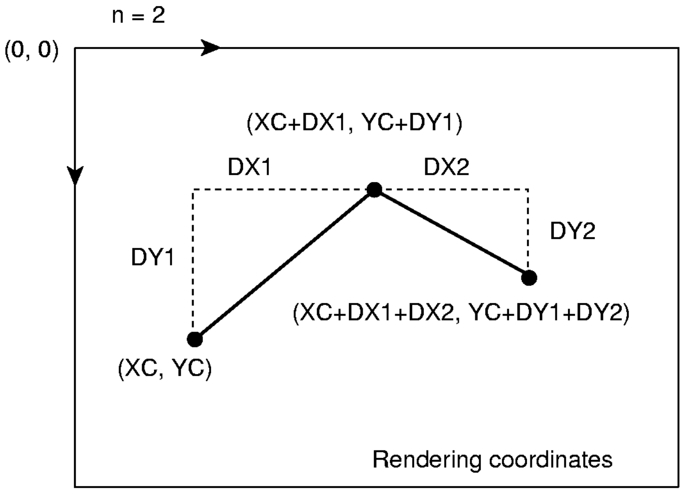
Description

Draws a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer (XC, YC).

The final coordinate point is stored as the current pointer (XC, YC).

Note: 8-point drawing is used.

Example

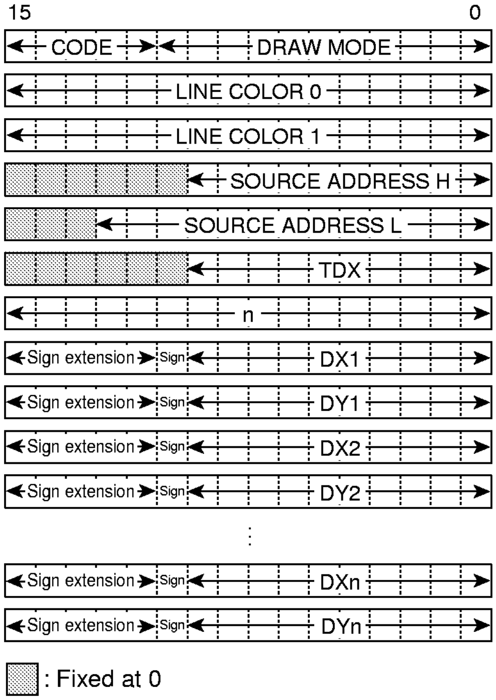


4.4.10 PLINE

Function

Draws a polygonal line at rendering coordinates while referencing a binary source.

Command Format



1. Code
   B'01110

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	O	×	×	O	×

Drawing Modes										
Reserved	TRNS	Reserved	CLIP	Reserved	NET	EOS	Reserved			
Fixed at 0	O	Fixed at 1	O	Fixed at 0	O	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 1

O : Can be used
× : Cannot be used

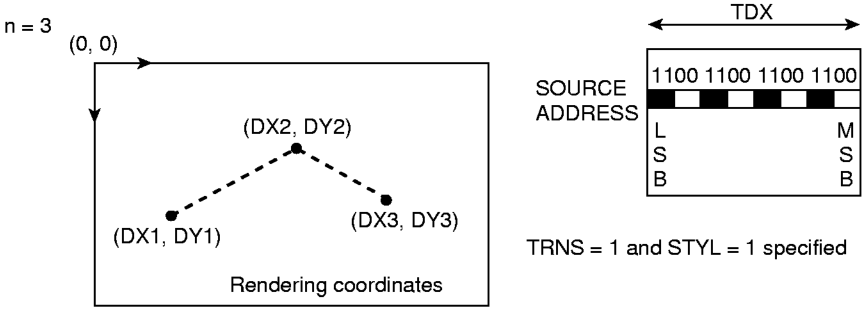
3. Command Parameters
   LINE COLOR0: 8 or 16-bit/pixel color specification
   LINE COLOR1: 8 or 16-bit/pixel color specification
   SOURCE ADDRESS H: 1-bit/pixel source start upper address
   SOURCE ADDRESS L: 1-bit/pixel source start lower address
   TDX: Source size
   n (n = 2 to 65,535): Number of vertices
   DXn (n = 2 to 65,535): Absolute coordinate
   DYn (n = 2 to 65,535): Absolute coordinate

Description

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn).

Note: 4-point drawing is used.

Example

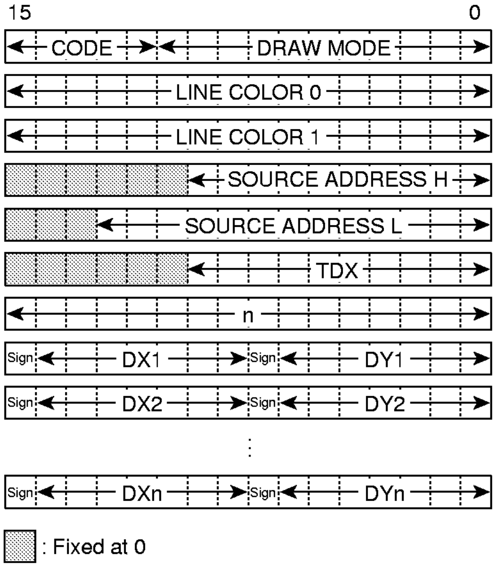


4.4.11 RPLINE

Function

Draws a polygonal line at rendering coordinates while referencing a binary source.

Command Format



1. Code
   B'01111

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	O	×	×	O	×

Drawing Modes										
Reserved	TRNS	Reserved	CLIP	Reserved	NET	EOS	Reserved			
Fixed at 0	O	Fixed at 1	O	Fixed at 0	O	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 1

O : Can be used
× : Cannot be used

3. Command Parameters

LINE COLOR0: 8 or 16-bit/pixel color specification

LINE COLOR1: 8 or 16-bit/pixel color specification

SOURCE ADDRESS H: 1-bit/pixel source start upper address

SOURCE ADDRESS L: 1-bit/pixel source start lower address

TDX: Source size

n (n = 1 to 65,535): Number of vertices

DXn, DYn (n = 1 to 65,535): Relative coordinates

Description

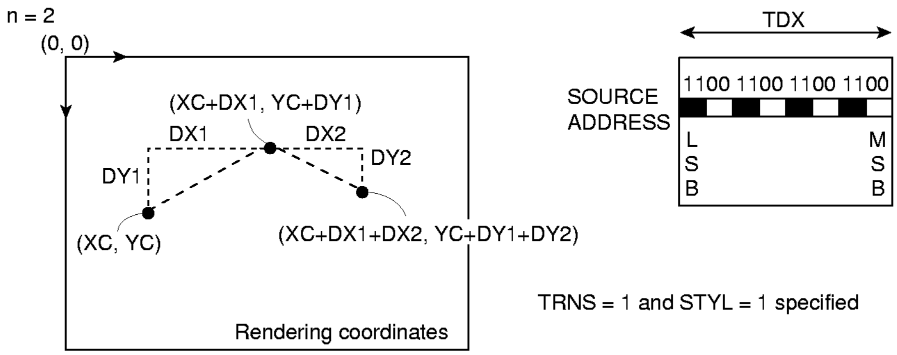
Draws a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer (XC, YC).

The final coordinate point is stored as the current pointer (XC, YC).

Commands used with WORK = 1 do not perform work referencing.

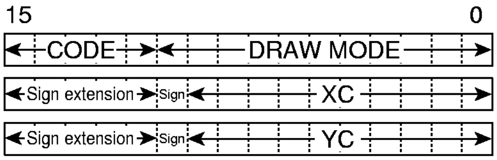
Note: 4-point drawing is used.

Example



4.4.12 MOVE

Command Format



1. Code
   B'10000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

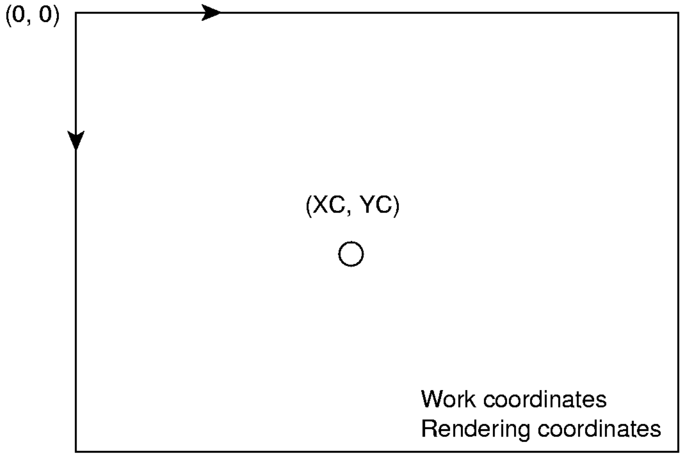
3. Command Parameters
   XC, YC: Absolute coordinates

Description

Sets the current pointer of the rendering coordinate system and the current pointer of the work coordinate system with absolute coordinates. The current pointer is used only by relative drawing commands.

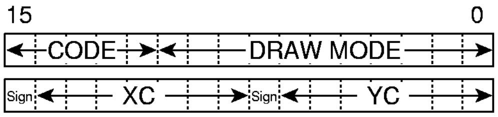
After issuing a MOVE command, use relative drawing commands in succession. If an absolute drawing command is used midway, the current pointer will be used as an operational register and its value will be overwritten. Therefore, a MOVE command must be issued before using relative drawing commands again.

Example



4.4.13 RMOVE

Command Format



1. Code

   B'10001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

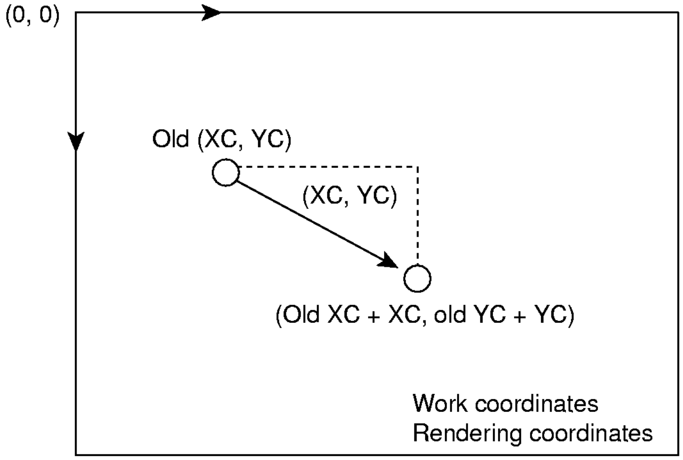
3. Command Parameters

   XC, YC: Relative coordinates

Description

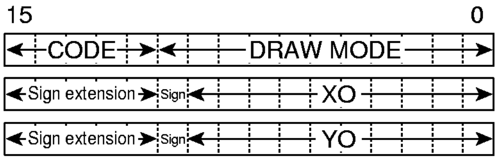
Sets the current pointer of the rendering coordinate system and the current pointer of the work coordinate system with relative coordinates from the old current pointer.

Example


(0, 0)
Old (XC, YC)
(XC, YC)
(Old XC + XC, old YC + YC)
Work coordinates
Rendering coordinates

4.4.14 LCOFS

Command Format



1. Code
   B'10010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

3. Command Parameters
   XO, YO: Local offset absolute specification

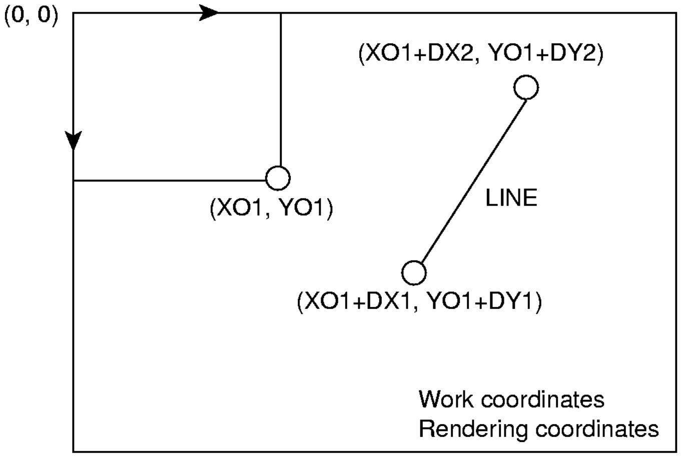
Description

Sets the local offset of the rendering coordinate system and the local offset of the work coordinate system with absolute coordinates. After this setting is made, this offset value is added in all subsequent coordinate specifications.

The start of the display list must be set (the initial value is undefined).

If the local offset is to be reflected in the current pointer, issue a MOVE command after the LCOFS command.

Example


(0, 0)
(XO1+DX2, YO1+DY2)
LINE
(XO1, YO1)
(XO1+DX1, YO1+DY1)
Work coordinates
Rendering coordinates

4.4.15 RLCOFS

Command Format

15			0
CODE	DRAW MODE		
Sign	XO	Sign	YO

1. Code
   B'10011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

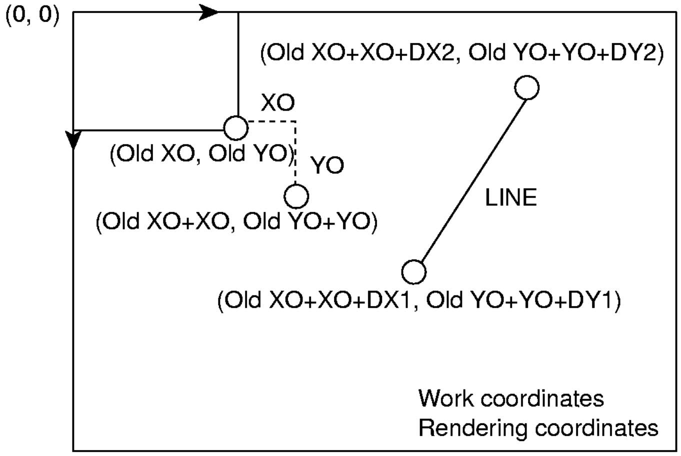
3. Command Parameters
   XO, YO: Local offset relative specification

Description

Sets the local offset of the rendering coordinate system and the local offset of the work coordinate system with relative coordinates from the old local offset. After this setting is made, this offset value is added in all subsequent coordinate specifications.

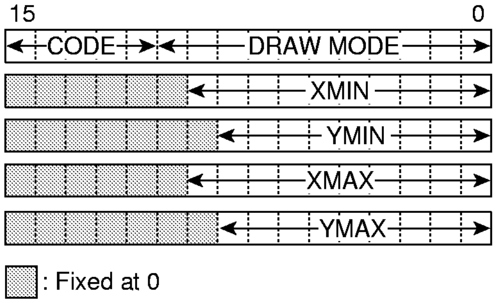
The old local offset refers to the local offset value set by the LCOFS command. If the local offset is to be reflected in the current pointer, execute a MOVE command after setting the offset with an LCOFS or RLCOFS command.

Example


(0, 0)
(Old XO+XO+DX2, Old YO+YO+DY2)
XO
(Old XO, Old YO)
YO
(Old XO+XO, Old YO+YO)
LINE
(Old XO+XO+DX1, Old YO+YO+DY1)
Work coordinates
Rendering coordinates

4.4.16 UCLIP

Command Format



1. Code

   B'10101

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

3. Command Parameters

   XMIN, XMAX: Left and right X coordinates

   YMIN, YMAX: Upper and lower Y coordinates

Description

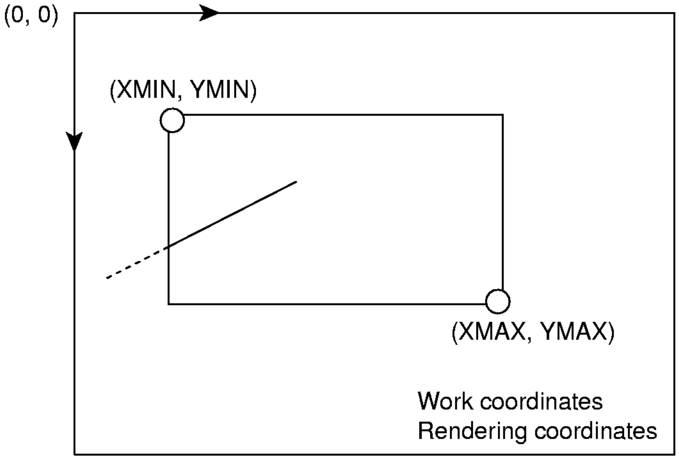
Designates the area specified by top-left coordinates (XMIN, YMIN) and bottom-right coordinates (XMAX, YMAX) in the rendering coordinate and work coordinate systems as a user clipping area.

The local offset value set by the LCOFS or RLCOFS command is not added to the coordinates set by this command.

When making this setting, ensure that the system clipping area is not exceeded.

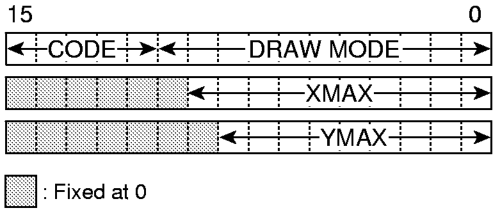
This setting is valid when CLIP = 1.

Example



4.4.17 SCLIP

Command Format



1. Code
   B'10111

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

3. Command Parameters
   XMAX: Horizontal X coordinate
   YMAX: Vertical Y coordinate

Description

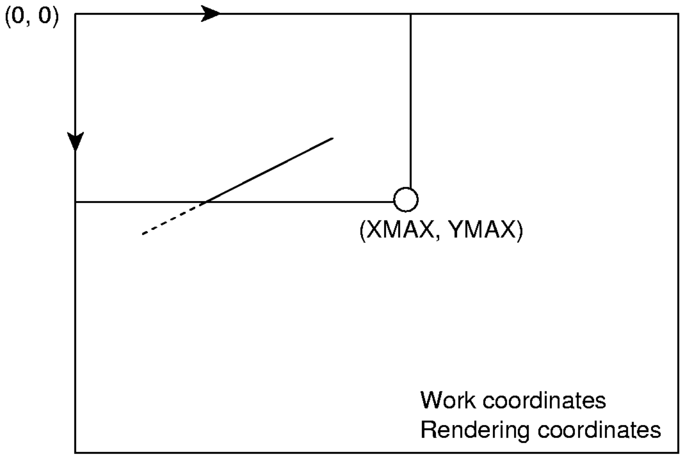
Designates the area specified by top-left coordinates (0, 0) and bottom-right coordinates (XMAX, YMAX) in the rendering coordinate and work coordinate systems as the system clipping area.

The local offset value set by the LCOFS or RLCOFS command is not added to the coordinates set by this command.

Make this setting according to the screen size. The start of the display list must be set (the initial value is undefined).

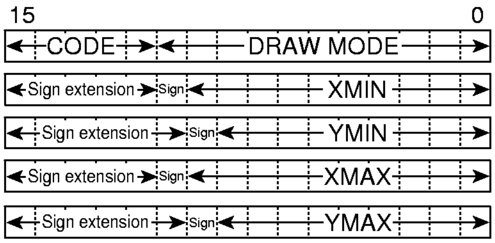
The set values are screen coordinates. This setting is valid when CLIP = 0.

Example



4.4.18 CLRW

Command Format



1. Code
   B'10100

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	O

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

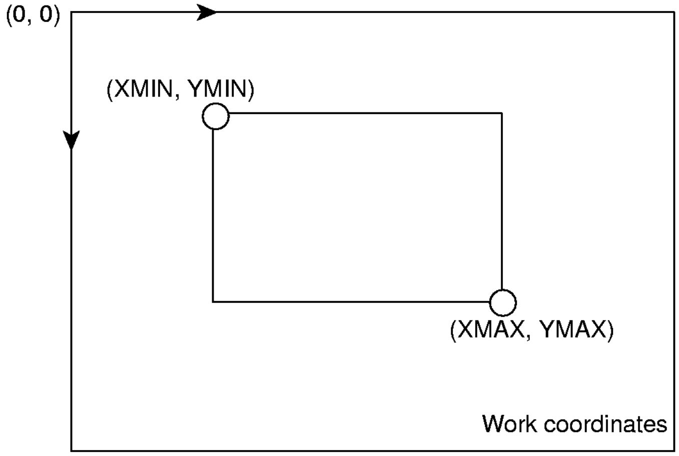
O : Can be used
× : Cannot be used

3. Command Parameters
   XMIN, XMAX: Left and right X coordinates
   YMIN, YMAX: Upper and lower Y coordinates

Description

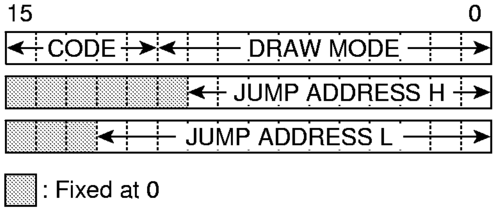
Zero-clears the area specified by top-left coordinates (XMIN, YMIN) and bottom-right coordinates (XMAX, YMAX) in the work coordinate system.

Example



4.4.19 JUMP

Command Format



1. Code
   B'11000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

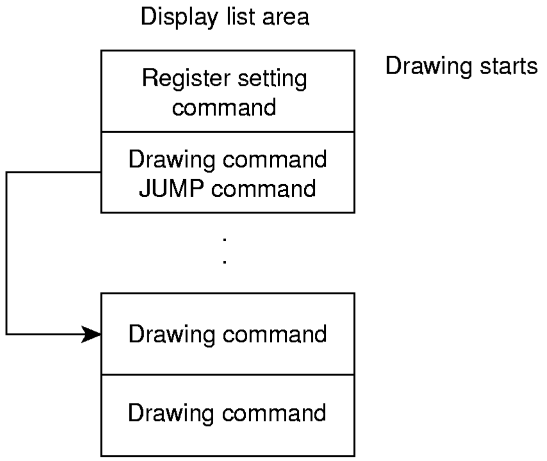
× : Cannot be used

3. Command Parameters
   JUMP ADDRESS H: Jump destination upper address
   JUMP ADDRESS L: Jump destination lower address

Description

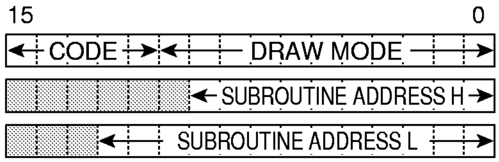
Changes the display list fetch destination to the specified address.

Example


Display list area
Register setting command
Drawing starts
Drawing command JUMP command
.
.
Drawing command
Drawing command

4.4.20 GOSUB

Command Format



1. Code

   B'11001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

3. Command Parameters

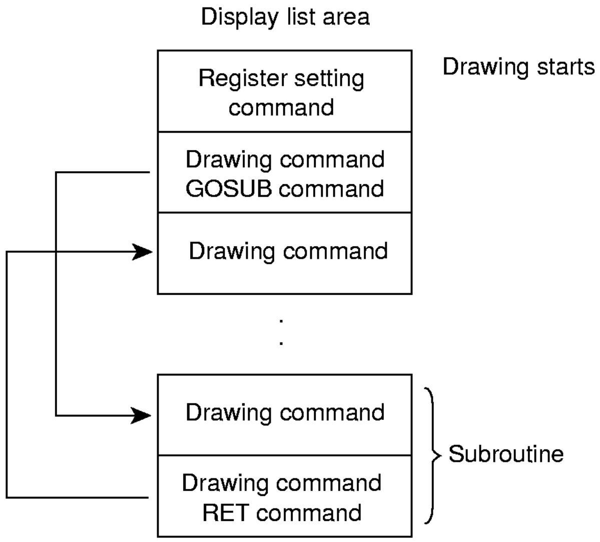
   SUBROUTINE ADDRESS H: Subroutine upper address

   SUBROUTINE ADDRESS L: Subroutine lower address

Description

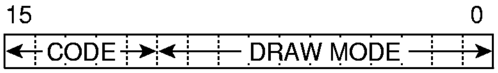
Changes the display list fetch destination to the specified subroutine address. The fetch address is restored by an RET instruction. As only one level of nesting is permitted, it will not be possible to return if a subroutine call is issued within the subroutine.

Example


Display list area
Register setting command
Drawing starts
Drawing command
GOSUB command
Drawing command
Drawing command
Subroutine
Drawing command
RET command

4.4.21 RET

Command Format



1. Code
   B'11011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

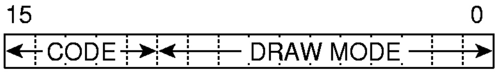
× : Cannot be used

Description

Restores the display list fetch destination to the address following the source of the subroutine call.

4.4.22 TRAP

Command Format



1. Code
   B'11111

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

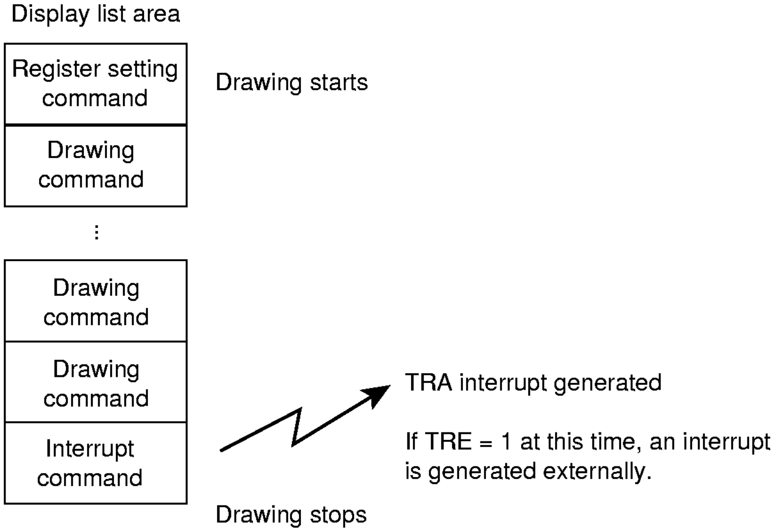
× : Cannot be used

Description

Halts the drawing operation and sends an interrupt to the CPU.

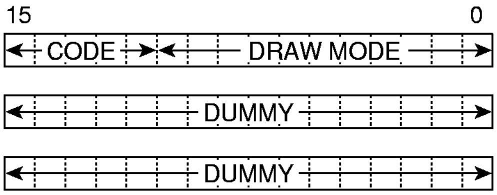
This command must be placed at the end of the display list.

Example


Display list area
Register setting command
Drawing command
⋮
Drawing command
Drawing command
Interrupt command
Drawing starts
TRA interrupt generated
If TRE = 1 at this time, an interrupt is generated externally.
Drawing stops

4.4.23 NOP3

Command Format



1. Code
   B'11110

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
×	×	×	×	×	×

Drawing Modes										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

× : Cannot be used

Description

The NOP3 command does not perform any operation. This command, which consists of three words including the command word, simply fetches the next instruction without executing any processing.

The NOP3 command can be used instead of a JUMP or GOSUB command.

Section 5 Registers

5.1 Overview

The Q2i has address-mapped registers mapped onto the address space (H'000 to H'2FF). These registers are divided into eight groups—interface control registers, memory control registers (two sets), display control registers, rendering control registers (two sets), input data control registers, and color palette registers. Word access is used on all of these registers. The address specification is made by inputting the address from pins A10 to A1 while the $\overline{\text{CS1}}$ pin is in the 0 state.

Addresses A10–A1 = H'028–H'03F and H'04C–H'0FF are reserved, and should not be read or written to. Reading or writing to these addresses may result in the loss of address-mapped register values, and unpredictable operation by the Q2i.

To enable the Q2i to manage UGM access rights, initial values must be set in the address-mapped registers by the SuperH before it accesses the UGM. If the UGM is accessed without setting these initial values, the Q2i may output a continuous wait signal to the SuperH. The setting procedure is shown in 1 to 3 below.

1. Set initial values in the system control register. Set SRES = 0, DRES = 1, DEN = 0.
2. Set initial values in registers 002–025.
3. Set SRES = 0, DRES = 0.

5.2 Register Updating

External Updating: Writing to an address-mapped register from the CPU is called external updating.

If external updating is performed in the interval from the raster following the end of screen display until immediately before the rise of $\overline{\text{VSYNC}}$, a register can be rewritten without causing display flicker.

As the VBK flag and FRM flag in the status register (SR) are set to 1 at the start of vertical blanking, external updating can be carried out using these flags.

Figures 5-1 (a) and (b) show the external update interval.

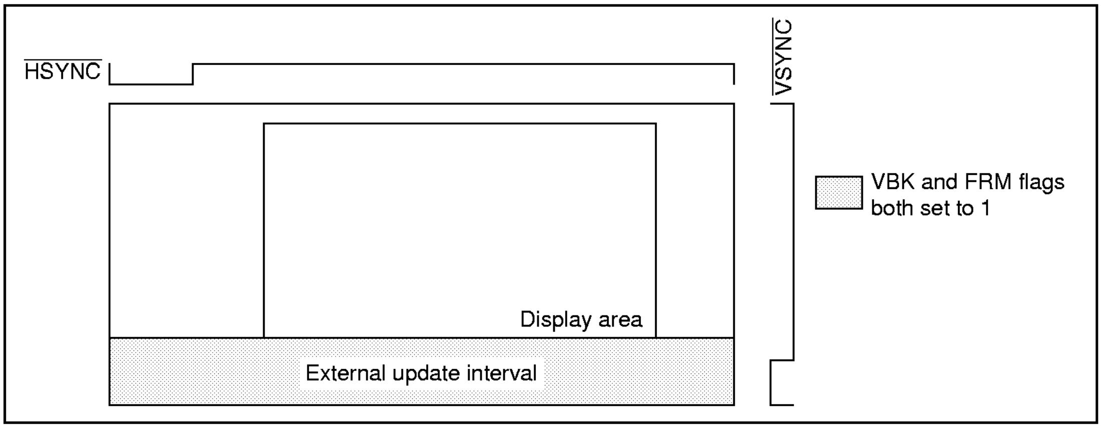

$\overline{\text{HSYNC}}$
$\overline{\text{VSYNC}}$
VBK and FRM flags both set to 1
Display area
External update interval

Figure 5-1 (a) External Update Interval (Interlace Mode)

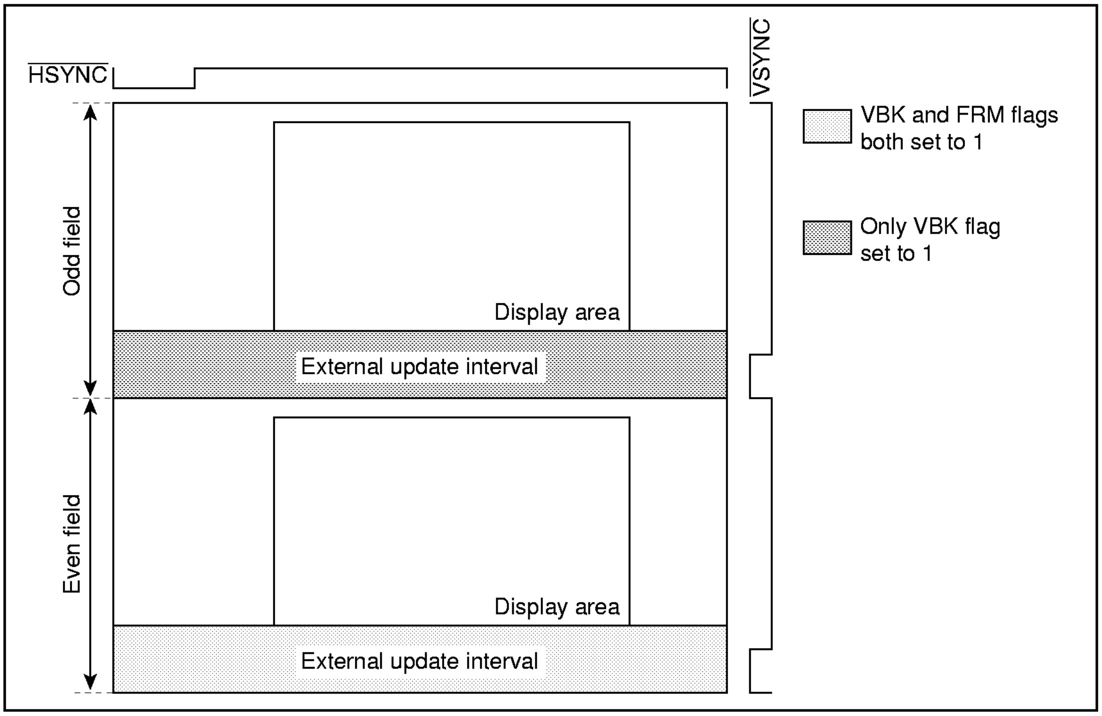


Figure 5-1 (b) External Update Interval (Interlace Mode and Interlace Sync & Video Mode)

Internal Updating: Some address-mapped registers have an internal update function. The internal update function is provided to prevent display flicker when the CPU modifies address-mapped registers relating to display operations without being aware of the display timing.

The display controller references address-mapped registers in coordination with display timing, and latches data in an internal register. This data transfer is called internal updating. Internal updating is carried out while the DRES bit is set to 1 in the system control register (SYSR) and at the beginning of each frame. The update is performed on setting of the fall of $\overline{\text{VSYNC}}$ when TVM1 = 0 and TVM0 = 0 in the display mode register (DSMR) (master mode), and on detection of the fall of $\overline{\text{EXVSYNC}}$ when TVM1 = 1 and TVM0 = 0 (TV mode). Internal updating is not performed when TVM1 = 0 and TVM0 = 1.

The address-mapped registers provided with the internal update function are shown in tables 5-1 (a) to (c). The initial values of these registers should be set while the DRES bit is set to 1. Internal updating is performed for display start address register 0 and display start address register 1 in display operations. In drawing operations, external updating is used.

Table 5-1 Registers with Internal Update Function

(a) Interface Control Registers

Address A[10:1]	Name	Abbreviation	Bits with Internal Update Function
000	System control register	SYSR	DEN (bit 13)
005	Display mode register	DSMD	WRAP (bit 11) BG (bit 10)

(b) Memory Control Registers

Address A[10:1]	Name	Abbreviation	Bits with Internal Update Function
008	Display size register X	DSXR	All bits
009	Display size register Y	DSYR	All bits
00A	Display start address register 0	DSAR0	All bits
00B	Display start address register 1	DSAR1	All bits
026	Background start coordinate register X	BGSRX	All bits
027	Background start coordinate register Y	BGSRY	All bits

(c) Display Control Registers

Address A[10:1]	Name	Abbreviation	Bits with Internal Update Function
013	Display window register (horizontal display start position)	DSWR (HDS)	All bits
014	Display window register (horizontal display end position)	DSWR (HDE)	All bits
015	Display window register (vertical display start position)	DSWR (VDS)	All bits
016	Display window register (vertical display end position)	DSWR (VDE)	All bits
017	Horizontal sync pulse width register	HSWR	All bits
018	Horizontal scan cycle register	HCR	All bits
019	Vertical sync position register	VSPR	All bits
01A	Vertical scan cycle register	VCR	All bits
01D	Color detection register H	CDERH	All bits
01E	Color detection register L	CDERL	All bits

5.3 Interface Control Registers

The interface control registers comprise eight 16-bit registers related to overall Q2i control, mapped onto addresses (A10–A1) H'000 to H'007.

5.3.1 System Control Register (SYSR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRES	DRES	DEN	—	—	RBRK	DC	RS	DBM1	DBM0	DMA1	DMA0	—	—	—	—
Initial value:	1	1	0	—	—	0	0	0	*	*	0	0	—	—	—	—
Read/Write:	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	—	—	—

Note: * Value is retained.

The system control register (SYSR) is a 16-bit readable/writable register that specifies Q2i system operation.

SYSR is initialized as follows in a reset:

- Bits SRES and DRES are set to 1.
- Bits DEN, RBRK, RS, DMA1, and DMA0 are cleared to 0.
- Bits DBM1, DBM0, and CCM3 to CCM0 retain their values.

Bit 15—Software Reset (SRES): Controls execution and suspension of command processing.

Bit 15: SRES	Description	
0	Command processing execution is enabled.	
1	SRES is set to 1 when a hardware reset is performed. Clear to 0 in initialization. When this bit is set to 1 by software, a reset is performed for drawing operations only. In this case, the bit must be set to 1 for at least 16 input clock cycles. When SRES is set to 1, the command error flag (CER), trap flag (TRA), command suspend flag (CSF), rendering break bit (RBRK), and drawing break flag (BRK) are cleared to 0.	(Initial value)

Bit 14—Display Reset (DRES)

Bit 13—Display Enable (DEN): These bits control starting and stopping of display synchronous operation.

Bit 14:	Bit 13:	
DRES	DEN	Description
0	0	Display operation is started. The DRES bit cannot be cleared to 0 while the $\overline{\text{RESET}}$ pin is low. When using the Q2i from the initial state, make all control register settings before clearing the DRES bit to 0. When the DEN bit is 0, display data from pins DD17 to DD0 has the value set in display off output registers H and L (DRORH, L).
	1	Display operation is started. The DRES bit cannot be cleared to 0 while the $\overline{\text{RESET}}$ pin is low. When using the Q2i from the initial state, make all control register settings, clear the DRES bit to 0, and then set the DEN bit to 1. Display data from pins DD17 to DD0 has the value stored in the UGM from the next frame.
1	0	Display synchronous operation is started. (Initial value) The Q2i only performs UGM refresh operations. With these settings, the Q2i operates as follows: 1. Drawing is not performed even if the RS bit is set to 1 in SYSR. 2. Display data from pins DD17 to DD0 is all-0 output. 3. The VBK flag is cleared to 0 in SR. 4. Waits are output continuously when a UGM access is performed by the CPU. When switching from DRES, DEN = 01 to DRES, DEN = 10, DRES, DEN = 11 is set temporarily for reasons relating to internal updating; this does not affect operation.
	1	Setting prohibited

Bits 12 and 11—Reserved: Only 0 should be written to these bits.

Bit 10—Rendering Break (RBRK): Controls rendering (drawing) breaks. This bit should only be set when the BRK bit is cleared to 0.

Bit 10: RBRK	Description
0	The TRA bit in the status register (SR) is set to 1 by TRAP command execution, and drawing switches is terminated. (Initial value)
1	The BRK bit in the status register (SR) is set to 1 when the currently executing command ends, and drawing is terminated. After suspension, the start address of the next command is placed in the command status register (CSTR). This bit is cleared to 0 after a drawing break.

Bit 9—Display Area Change (DC): Controls frame buffer switching in manual display change mode.

Bit 9: DC	Description
0	Switching of the frame buffer for display is not performed in manual display change mode. When the DC bit is 0, it can be set to 1. (Initial value)
1	Switching of the frame buffer for display is performed in manual display change mode. This bit can be set to 1 only when it is 0. Switching is performed in frame units in non-interlace and interlace modes, and in field units in interlace sync & video mode. This bit is cleared to 0 after frame buffer switching. Therefore, DC should not be cleared to 0 by the SuperH.

Bit 8—Rendering Start (RS): Specifies the start of rendering.

Bit 8: RS	Description
0	Rendering is not started. (Initial value)
1	Rendering is started. This bit is cleared to 0 after rendering starts. When starting rendering, have a UGM dummy read performed by the CPU, clear the internal FIFO, and then set this bit to 1. The internal FIFO is cleared automatically 64 CLK0 cycles later, after which drawing can be performed by setting this bit to 1. When drawing is performed after being suspended using the RBRK bit, check that the RS bit is cleared to 0 by the SuperH and ensure that the state in which the Q2i performs drawing is entered.

Bits 7 and 6—Double-Buffer Mode 1 and 0 (DBM1, DBM0): These bits select double-buffer control.

Bit 7:	Bit 6:	
DBM1	**DBM0**	**Description**
0	0	Auto display change mode is set.
	1	Auto rendering mode is set.
1	0	Manual display change mode is set.
	1	Setting prohibited

Bits 5 and 4—DMA Mode (DMA1, DMA0): These bits specify DMA transfer. Use the DMA flag (DMF) in SR to check for the start and end of DMA mode.

Bit 5:	Bit 4:	
DMA1	**DMA0**	**Description**
0	0	Normal mode is set. (Initial value)
	1	The mode for DMA transfer to memory (UGM) corresponding to $\overline{CS0}$ is set. When the remaining DMA transfer count reaches 0, this bit is automatically cleared and normal mode is entered. The initial value of the remaining DMA transfer count is determined by the setting in the DMA transfer word count register (DMAWR). The remaining DMA transfer count is an internal value in the LSI, and is decremented by 1 each time a word is processed. UGM access by the CPU is disabled in this mode.
1	0	Setting prohibited
	1	The mode for DMA transfer to the register [image data entry register (IDER)] corresponding to $\overline{CS1}$ is set. In this mode, register address incrementing is not performed and all writes are to IDER. When the remaining DMA transfer count reaches 0, this bit is automatically cleared and normal mode is entered. The initial value of the remaining DMA transfer count is determined by the setting in the DMA transfer word count register (DMAWR). The remaining DMA transfer count is an LSI internal value, and is decremented by 1 each time a word is processed. UGM access by the CPU is disabled in this mode.

Bits 3 to 0—Reserved: Only 0 should be written to these bits.

5.3.2 Status Register (SR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVR	FRM	DMF	CER	VBK	TRA	CSF	DBF	BRK	—	—	—	Q3	Q2	Q1	Q0
Initial value:	0	0	0	0	0	0	0	*	0	—	—	—	0	0	1	1
Read/Write:	R	R	R	R	R	R	R	R	R	—	—	—	R	R	R	R

Note: * Value is retained.

The status register (SR) is a 16-bit read-only register used to read the internal status of the Q2i from outside.

SR is initialized as follows in a reset:

- Flag DBF retains its values.
- The Q flags are set to 0011.
- All other flags are cleared to 0.

Bit 15—TV Sync Signal Error Flag (TVR): Flag that indicates that $\overline{\text{EXVSYNC}}$ has been detected within the vertical cycle.

Bit 15: TVR	Description
0	The rise of $\overline{\text{EXVSYNC}}$ has been detected each time within the vertical cycle determined by the vertical scan cycle register (VCR) setting after the TVR flag has been cleared by the DRES bit in SYSR or the TVCL bit in SRCR. (Initial value)
1	In TV sync mode (bits TVM1 and TVM0 = 10 in DSMR), a rise of $\overline{\text{EXVSYNC}}$ has not been detected within the vertical cycle determined by the VCR set value. The TVR flag retains its state until cleared by a reset or by software.

Bit 14—Frame Flag (FRM): Flag that indicates the vertical blanking interval after frame display.

Bit 14: FRM	Description
0	Indicates the interval from FRM flag clearing by the DRES bit in SYSR or the FRCL bit in SRCR until the end of the next display in non-interlace mode, or until the end of the next even field display in interlace mode or interlace sync & video mode. (Initial value)
1	Indicates the interval from the first even field vertical blanking interval after FRM flag clearing by the DRES bit in SYSR or the FRCL bit in SRCR until the FRM flag is cleared again (frame units).

Bit 13—DMA Flag (DMF): Flag that indicates that DMA transfer mode has been initiated and transfer has been completed.

Bit 13: DMF	Description
0	DMA transfer mode has not been initiated at all since DMF flag clearing by the DMCL bit in SRCR, or the next DMA transfer mode (bits DMA1 and DMA0 = 01 or 11 in SYSR) has been initiated and the remaining transfer count has not yet reached 0. (Initial value)
1	DMA transfer mode has been initiated and the transfer word count has reached 0. The DMF flag retains its state until cleared by a reset or by software.

Bit 12—Command Error Flag (CER): Flag that indicates that an illegal command has been fetched.

Bit 12: CER	Description
0	Normal state. An illegal command has not been fetched since CER flag clearing by the SRES bit in SYSR or the CECL bit in SRCR. (Initial value)
1	Drawing operation halt state. Drawing operation remains halted because an illegal command was fetched after CER flag clearing by the SRES bit in SYSR or the CECL bit in SRCR. The CER flag retains its state until cleared by a reset or by software.

Bit 11—Vertical Blanking Flag (VBK): Flag that indicates the vertical blanking interval.

Bit 11: VBK	Description
0	Indicates the interval from VBK flag clearing by the DRES bit in SYSR or the VBCL bit in SRCR until the end of the next display. (Initial value)
1	Indicates the interval from the first vertical blanking interval after VBK flag clearing by the DRES bit in SYSR or the VBCL bit in SRCR until the VBK flag is cleared again (field units).

Bit 10—Trap Flag (TRA): Flag that indicates the end of command execution.

Bit 10: TRA	Description
0	Indicates the interval from TRA flag clearing by the SRES bit in SYSR or the TRCL bit in SRCR until the end of execution of the next command. (Initial value)
1	Command execution has ended, or the current command is not being executed. The TRA flag retains its state until cleared by a reset or by software.

Bit 9—Command Suspend Flag (CSF): Flag that indicates that command execution has been suspended by a frame change in auto display change mode or manual display change mode.

Bit 9: CSF	Description
0	Normal operation (Initial value)
1	A rendering end interrupt has not been generated in the interval from CSF flag clearing by the SRES bit in SYSR or the CSCL bit in SRCR until the next frame change. The CSF flag retains its state until cleared by a reset or by software.

Bit 8—Display Buffer Frame (DBF): Flag that indicates the frame buffer being displayed.

Bit 8: DBF	Description
0	Frame buffer 0 (F0) is being displayed.
1	Frame buffer 1 (F1) is being displayed.

Bit 7—Drawing Break Flag (BRK): Flag that indicates a drawing break.

Bit 7: BRK	Description
0	Indicates the interval until the next drawing break occurs after the BRK flag is cleared by the SRES bit in SYSR or the BRCL bit in SRCR. (Initial value)
1	Indicates that a command is not currently being executed due to a drawing break directive. The BRK flag retains its state until cleared by a reset or by software.

Bits 6 to 4—Reserved: These bits always read 0.

Bits 3 to 0—Q Flags (Q3 to Q0): Flags used for Q Series product identification. In the Q2i, 0011 is read from these flags.

Bit 3: Q3	Bit 2: Q2	Bit 1: Q1	Bit 0: Q0	Description
0	0	1	0	HD64411F (Q2)
0	0	1	1	HD64412F (Q2i)

5.3.3 Status Register Clear Register (SRCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVCL	FRCL	DMCL	CECL	VBCL	TRCL	CSCL	—	BRCL	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	—	*	—	—	—	—	—	—	—
Read/Write:	W	W	W	W	W	W	W	—	W	—	—	—	—	—	—	—

Note: * Value is retained.

The status register clear register (SRCR) is a 16-bit write-only register that clears the corresponding flags in SR. Writing 1 to one of bits 15 to 9 or 7 in SRCR will clear the corresponding flag in SR to 0. When SR clearing is completed, the value of SRCR is cleared to all-0 internally (a read will return 0).

Bit	Bit Name	Abbreviation	Description
15	TV sync signal error flag clear	TVCL	Writing 1 to the TVCL bit clears the TVR flag to 0 in SR.
14	Frame buffer clear	FRCL	Writing 1 to the FRCL bit clears the FRM flag to 0 in SR.
13	DMA flag clear	DMCL	Writing 1 to the DMCL bit clears the DMF flag to 0 in SR.
12	Command error flag clear	CECL	Writing 1 to the CECL bit clears the CER flag to 0 in SR.
11	Vertical blanking flag clear	VBCL	Writing 1 to the VBCL bit clears the VBK flag to 0 in SR.
10	Trap flag clear	TRCL	Writing 1 to the TRCL bit clears the TRA flag to 0 in SR.
9	Command suspend flag clear	CSCL	Writing 1 to the CSCL bit clears the CSF flag to 0 in SR.
7	Drawing break flag clear	BRCL	Writing 1 to the BRCL bit clears the BRK flag to 0 in SR.
8, 6–0	Reserved	—	Only 0 should be written to these bits.

5.3.4 Interrupt Enable Register (IER)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVE	FRE	DME	CEE	VBE	TRE	CSE	—	BRE	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	—	0	—	—	—	—	—	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W	—	—	—	—	—	—	—

The interrupt enable register (IER) is a 16-bit readable/writable register that enables or disables interrupts by the corresponding flags in SR. When a bit in SR is set to 1 and the bit at the corresponding bit position in IER is also 1, $\overline{\text{IRL}}$ is driven low and an interrupt request is sent to the CPU.

The interrupt generation condition is as follows.

$$\text{Interrupt generation condition} = \overline{\text{IRL}} = \overline{a+b+c+d+e+f+g}$$

$$a = \text{TVR} \cdot \text{TVE}$$
$$b = \text{FRM} \cdot \text{FRE}$$
$$c = \text{DMF} \cdot \text{DME}$$
$$d = \text{CER} \cdot \text{CEE}$$
$$e = \text{VBK} \cdot \text{VBE}$$
$$f = \text{TRA} \cdot \text{TRE}$$
$$g = \text{CSF} \cdot \text{CSE}$$
$$h = \text{BRK} \cdot \text{BRE}$$

Bit 15—TV Sync Signal Error Flag Enable (TVE): Enables or disables interrupts initiated by the TVR flag in SR.

Bit 15: TVE	Description	
0	Interrupts initiated by the TVR flag in SR are disabled.	(Initial value)
1	Interrupts initiated by the TVR flag in SR are enabled. When TVR·TVE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.	

Bit 14—Frame Flag Enable (FRE): Enables or disables interrupts initiated by the FRM flag in SR.

Bit 14: FRE	Description	
0	Interrupts initiated by the FRM flag in SR are disabled.	(Initial value)
1	Interrupts initiated by the FRM flag in SR are enabled. When FRM·FRE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.	

Bit 13—DMA Flag Enable (DME): Enables or disables interrupts initiated by the DMF flag in SR.

Bit 13: DME	Description
0	Interrupts initiated by the DMF flag in SR are disabled. (Initial value)
1	Interrupts initiated by the DMF flag in SR are enabled. When DMF·DME = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.

Bit 12—Command Error Flag Enable (CEE): Enables or disables interrupts initiated by the CER flag in SR.

Bit 12: CEE	Description
0	Interrupts initiated by the CER flag in SR are disabled. (Initial value)
1	Interrupts initiated by the CER flag in SR are enabled. When CER·CEE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.

Bit 11—Vertical Blanking Flag Enable (VBE): Enables or disables interrupts initiated by the VBK flag in SR.

Bit 11: VBE	Description
0	Interrupts initiated by the VBK flag in SR are disabled. (Initial value)
1	Interrupts initiated by the VBK flag in SR are enabled. When VBK·VBE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.

Bit 10—Trap Flag Enable (TRE): Enables or disables interrupts initiated by the TRA flag in SR.

Bit 10: TRE	Description
0	Interrupts initiated by the TRA flag in SR are disabled. (Initial value)
1	Interrupts initiated by the TRA flag in SR are enabled. When TRA·TRE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.

Bit 9—Command Suspend Flag Enable (CSE): Enables or disables interrupts initiated by the CSF flag in SR.

Bit 9: CSE	Description	
0	Interrupts initiated by the CSF flag in SR are disabled.	(Initial value)
1	Interrupts initiated by the CSF flag in SR are enabled. When CSF·CSE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.	

Bit 7—Drawing Break Flag Enable (BRE)

Bit 7: BRE	Description	
0	Interrupts by the BRK flag in SR are disabled.	(Initial value)
1	Interrupts by the BRK flag in SR are enabled. When BRK·BRE = 1, an $\overline{\text{IRL}}$ interrupt request is sent to the CPU.	

Bits 8 and 6 to 0—Reserved: Only 0 should be written to these bits.

5.3.5 Memory Mode Register (MEMR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	MES2	MES1	MES0	MEA1	MEA0	—	—
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	—	—

Note: * Value is retained.

The memory mode register (MEMR) is a 16-bit readable/writable register that specifies the size of UGM used and the number of row address lines.

If the value of this register is modified during a memory access, operation will be temporarily unstable.

MEMR bits MES2 to MES0, MEA1, and MEA0 retain their values in a reset.

Bits 15 to 7—Reserved: Only 0 should be written to these bits.

Bits 6 to 4—Memory Size (MES2 to MES0): These bits select the size and quantity of memories used for the UGM.

Bit 6: MES2	Bit 5: MES1	Bit 4: MES0	Description
0	0	0	Memory size: 4 Mbits × 1
		1	Memory size: 4 Mbits × 2
	1	0	Memory size: 16 Mbits × 1
		1	Memory size: 16 Mbits × 2
1	*	*	Setting prohibited

*: Don’t care

Bits 3 and 2—Memory Address Mode (MEA1, MEA0): These bits select the number of row address lines for the memory used for the UGM.

Bit 3: MEA1	Bit 2: MEA0	Description
0	0	9 row address lines
	1	10 row address lines
1	0	11 row address lines
	1	12 row address lines

Bits 1 and 0—Reserved: Only 0 should be written to these bits.

5.3.6 Display Mode Register (DSMR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WRAP	BG	—	DOT	TVM1	TVM0	SCM1	SCM0	REF3	REF2	REF1	REF0
Initial value:	—	—	—	—	0	0	—	*	1	0	*	*	1	0	0	0
Read/Write:	—	—	—	—	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The display mode register (DSMR) is a 16-bit readable/writable register that specifies the Q2i display operation.

If the value of this register is modified during a display operation, operation will be temporarily unstable.

DSMR is initialized as follows in a reset:

Bits WRAP and BG are initialized to 0, bits TVM1 and TVM0 to 10, and bits REF3 to REF0 to 1000.

The DOT, SCM1, and SCM0 bits retain their values.

Bits 15 to 12—Reserved: Only 0 should be written to these bits.

Bit 11—Background Screen Wraparound Mode Configuration (WRAP)

Bit 11: WRAP	Description
0	Background screen wraparound is not performed. Display contents are not guaranteed if the display area extends beyond the memory installation space. (Initial value)
1	Background screen wraparound is performed. The wraparound units are the number of pixels specified by the MWX bit in the rendering mode register (REMR) in the X direction, and 512 pixels in the Y direction. The start coordinates of this area are indicated by bits 13 to 9 of the background start address register (BGSR).

Bit 10—Background Screen Combination (BG)

Bit 10: BG	Description
0	Background screen combination is not performed. (Initial value)
1	Background screen combination is performed.

Bit 9—Reserved: Only 0 should be written to this bit.

Bit 8—Dot Clock Mode (DOT): Specifies settings for the dot clock, the basic clock for the Q2i's display block.

Bit 8: DOT	Description
0	The clock input from the CLK1 pin is used as the display dot clock. The frequency of the clock output from the DCLK pin is the same as that of CLK1. The frequency of the clock output from the FCLK pin is 1/2 that of CLK1.
1	A clock with 1/2 the frequency of the clock input from the CLK1 pin is used as the display dot clock. The frequency of the clock output from the DCLK pin is 1/2 that of CLK1. The frequency of the clock output from the FCLK pin is 1/4 that of CLK1.

Figure 5-2 shows the display clock timing.

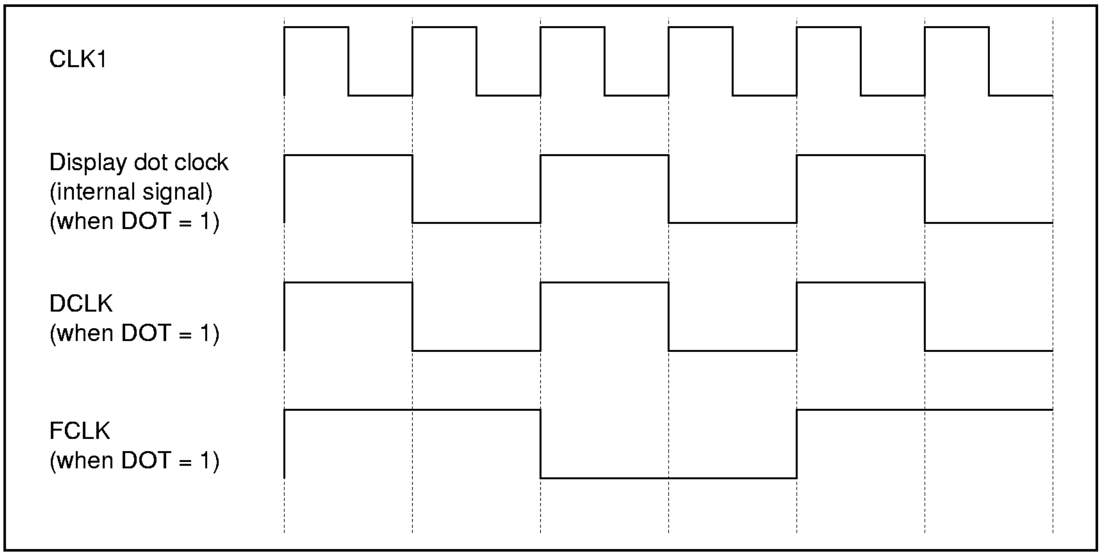


Figure 5-2 Display Clock Timing (DOT = 1)

Bits 7 and 6—TV Sync Mode (TVM1, TVM0): These bits specify TV sync mode, in which synchronous operation is performed by means of $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ input from an external source, or master mode, in which $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ are output.

Bit 7: TVM1	**Bit 6: TVM0**	**Description**
0	0	Master mode is set. The Q2i outputs $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{ODDF}}$ signals.
	1	Synchronization system switching mode is set. Switching is performed from TV sync mode to master mode, or vice versa, via this mode. In this mode, display operations are forcibly halted and the DISP pin output goes low. The clock supply to the CLK1 pin can also be stopped (input invalidated) (fixed high within the chip). The $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{ODDF}}$ pins are inputs.
1	0	TV sync mode is set. $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{ODDF}}$ signals are input to the Q2i. (Initial value)
	1	Setting prohibited

Bits 5 and 4—Scan Mode (SCM1, SCM0): These bits specify the display output scan mode and the unit of display switching.

Bit 5: SCM1	Bit 4: SCM0	Description
0	0	Non-interlace mode: Frame buffer switching can be performed in 1-VC units.
	1	Setting prohibited
1	0	Interlace mode: Frame buffer switching can be performed in 2-VC units.
	1	Interlace sync & video mode: Frame buffer switching can be performed in 1-VC units.

Bits 3 to 0—Refresh Cycles (REF3 to REF0): These bits specify the number of cycles for which refreshing is performed within one raster in the display screen area.

Bit 3: REF3	Bit 2: REF2	Bit 1: REF1	Bit 0: REF0	Description
0	0	0	0	Refresh timing is not output
			1	Number of refresh cycles = 1
		1	0	Number of refresh cycles = 2
			1	Number of refresh cycles = 3
	1	0	0	Number of refresh cycles = 4
			1	Number of refresh cycles = 5
		1	0	Number of refresh cycles = 6
			1	Number of refresh cycles = 7
1	0	0	0	Number of refresh cycles = 8 (Initial value)
			1	Number of refresh cycles = 9
		1	0	Number of refresh cycles = 10
			1	Number of refresh cycles = 11
	1	0	0	Number of refresh cycles = 12
			1	Number of refresh cycles = 13
		1	0	Number of refresh cycles = 14
			1	Number of refresh cycles = 15

5.3.7 Rendering Mode Register (REMR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	MWX	—	—	—	—	—	GBM
Initial value:	—	—	—	—	—	—	—	—	—	*	—	—	—	—	—	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	—	—	—	—	—	R/W

Note: * Value is retained.

The rendering mode register (REMR) is a 16-bit readable/writable register that specifies Q2i rendering operations.

If the value of this register is modified during a drawing operation, operation will be temporarily unstable.

REMR bits MWX and GBM retain their values in a reset.

Bits 15 to 7—Reserved: Only 0 should be written to these bits.

Bit 6—Memory Width (MWX): Specifies the X-direction logical coordinate space of the UGM connected to the Q2i.

Bit 6: MWX	Description
0	X-direction logical coordinate space is 512 pixels.
1	X-direction logical coordinate space is 1024 pixels.

Bits 5 to 1—Reserved: Only 0 should be written to these bits.

Bit 0—Graphic Bit Mode (GBM): Specifies the bit configuration of the rendering data handled by the Q2i.

Bit 0: GBM	Description
0	Rendering data bit configuration is 8 bits/pixel.
1	Rendering data bit configuration is 16 bits/pixel.

Figure 5-3 shows the correspondence between memory physical addresses (bytes) and rendering and multi-valued source coordinates. The X upper coordinate and X lower coordinate signify the values when the memory map example X value is divided into the respective bus widths. Similarly, the Y upper coordinate and Y lower coordinate are values obtained by dividing the Y value.

GBM = 0 (8 bits/pixel), MWX = 0 (512 pixels)

<table>
<tr><td>A22</td><td>A21</td><td>A20</td><td>A19</td><td>A18</td><td>A17</td><td>A16</td><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td><td>A7</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>A0</td></tr>
<tr><td colspan="10">Y upper coordinate</td><td colspan="4">X upper coordinate</td><td colspan="4">Y lower coordinate</td><td colspan="5">X lower coordinate</td></tr>
</table>

GBM = 0 (8 bits/pixel), MWX = 1 (1024 pixels)

<table>
<tr><td>A22</td><td>A21</td><td>A20</td><td>A19</td><td>A18</td><td>A17</td><td>A16</td><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td><td>A7</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>A0</td></tr>
<tr><td colspan="9">Y upper coordinate</td><td colspan="5">X upper coordinate</td><td colspan="4">Y lower coordinate</td><td colspan="5">X lower coordinate</td></tr>
</table>

GBM = 1 (16 bits/pixel), MWX = 0 (512 pixels)

<table>
<tr><td>A22</td><td>A21</td><td>A20</td><td>A19</td><td>A18</td><td>A17</td><td>A16</td><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td><td>A7</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>A0</td></tr>
<tr><td colspan="9">Y upper coordinate</td><td colspan="5">X upper coordinate</td><td colspan="4">Y lower coordinate</td><td colspan="4">X lower coordinate</td><td>0</td></tr>
</table>

GBM = 1 (16 bits/pixel), MWX = 1 (1024 pixels)

<table>
<tr><td>A22</td><td>A21</td><td>A20</td><td>A19</td><td>A18</td><td>A17</td><td>A16</td><td>A15</td><td>A14</td><td>A13</td><td>A12</td><td>A11</td><td>A10</td><td>A9</td><td>A8</td><td>A7</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>A0</td></tr>
<tr><td colspan="8">Y upper coordinate</td><td colspan="6">X upper coordinate</td><td colspan="4">Y lower coordinate</td><td colspan="4">X lower coordinate</td><td>0</td></tr>
</table>

Upper line: Memory physical addresses (bytes) A22 to A1, A0
Lower line: Logical coordinates (X, Y)

A0 is an LSI internal signal, indicating the LSB of the byte address.
When GBM = 0, the X lower coordinate value must be an even number.

Figure 5-3 Correspondence between Memory Physical Addresses (Bytes) and Rendering Coordinates and Multi-Valued Source Coordinates

5.3.8 Input Data Conversion Mode Register (IEMR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	YUV1	YUV0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	R/W	R/W

Note: * Value is retained.

The input data conversion mode register (IEMR) is a 16-bit readable/writable register that specifies the conversion format for input data from the CPU.

If the value of this register is modified during data conversion, operation will be temporarily unstable.

IEMR bits YUV1 and YUV0 retain their values in a reset.

Bits 15 to 2—Reserved: Only 0 should be written to these bits.

Bits 1 and 0—YUV Mode (YUV1, YUV0): These bits specify conversion to RGB format, and storage in the UGM, of data input in YUV or (YUV format.

Bit 1: YUV1	Bit 0: YUV0	Description
0	0	Normal mode is set. Data conversion is not performed.
	1	YUV-RGB conversion is performed. When the total number of data conversion pixels reaches 0, this bit is automatically cleared and normal mode is entered. The total number of data conversion pixels is the product of the image data size register X and Y (IDSRX, IDSRY) set values. The total number of data conversion pixels is decremented by 1 in the LSI each time a pixel is processed. UGM access by the CPU is disabled in this mode.
1	0	(YUV-RGB conversion is performed. When the total number of data conversion pixels reaches 0, this bit is automatically cleared and normal mode is entered. The total number of data conversion pixels is the product of the image data size register X and Y (IDSRX, IDSRY) set values. The total number of data conversion pixels is decremented by 1 in the LSI each time a pixel is processed. UGM access by the CPU is disabled in this mode.
	1	Setting prohibited

5.4 Memory Control Registers

The memory control registers comprise eleven 16-bit registers related to the UGM (unified graphics memory) configuration, mapped onto addresses (A10–A1) H'008 to H'012.

5.4.1 Display Size Registers X and Y (DSRX, DSRY)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSRX	—	—	—	—	—	—	DSX	DSX	DSX	DSX	DSX	DSX	DSX	DSX	DSX	DSX
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSRY	—	—	—	—	—	—	—	DSY	DSY	DSY	DSY	DSY	DSY	DSY	DSY	DSY
Initial value:	—	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

Display size registers X and Y (DSRX, DSRY) are 16-bit readable/writable registers that specify the size of the display screen. The number of dots in the horizontal direction is set in DSRX, and the number of dots in the vertical direction in DSRY.

The set value of the DSX bits (H'0000 to H'03FF) corresponds to the number of horizontal dots, from 1 to 1024.

The set value of the DSY bits (H'0000 to H'01FF) corresponds to the number of vertical dots, from 1 to 512.

Bits 15 to 10 of DSRX and bits 15 to 9 of DSRY are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The DSX bits in DSRX and the DSY bits in DSRY retain their values in a reset.

5.4.2 Display Start Address Registers 0 and 1 (DSAR0, DSAR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAR0	—	—	—	—	—	—	—	—	—	DSA0 (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAR1	—	—	—	—	—	—	—	—	—	DSA1 (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

Display address registers 0 and 1 (DSAR0, DSAR1) are 16-bit readable/writable registers that specify the memory areas to be used as UGM frame buffers.

Only the upper 6 bits (A22 to A16) of the start physical address of frame buffer 0 (F0) are set in the DSA0 field in DSAR0, and only the upper 6 bits (A22 to A16) of the start physical address of frame buffer 1 (F1) are set in the DSA1 field in DSAR1.

The display start address register whose contents are actually valid as the display start address is the register indicated by bit DBF in SR. The display start address register whose contents are not valid as the display start address indicates the rendering coordinate origin. When these registers are modified, the new set value becomes valid when an internal update is performed in the case of the display start address register whose contents are valid as the display start address, and when an external update (rewrite) is performed in the case of the display start address register that indicates the rendering coordinate origin.

Bits 15 to 7 of DSAR0 and DSAR1 are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The DSA0 field in DSAR0 and the DSA1 field in DSAR1 retain their values in a reset.

5.4.3 Display List Start Address Registers H and L (DLSARH, DLSARL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLSARH	—	—	—	—	—	—	—	—	—	DLSAH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLSARL	DLSAL (address A15–A5 setting)											—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*	*	*	*	—	—	—	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	—	—	—	—

Note: * Value is retained.

Display list start address registers H and L (DLSARH, DLSARL) are 16-bit readable/writable registers that specify the memory area to be used as the display list.

The DLSAH field in DLSARH and the DLSAL field in DLSARL contain a total of 18 bits, and only the upper bits (A22 to A5) of the start physical address of the display list are set in these fields.

Bits 15 to 7 of DLSARH and bits 4 to 0 of DLSARL are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The DLSAH field in DLSARH and the DLSAL field in DLSARL retain their values in a reset.

5.4.4 Multi-Valued Source Area Start Address Register (SSAR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SSAH (address A22–A17 setting)						—
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—

Note: * Value is retained.

The multi-valued source area start address register (SSAR) is a 16-bit readable/writable register that specifies the memory area to be used as the multi-valued source area. Only the upper bits (A22 to A17) of the start physical address of the source area are set in the SSAH field.

Bits 15 to 7 and 0 of SSAR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The SSAH field in SSAR retains its value in a reset.

5.4.5 Work Area Start Address Register (WSAR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WSAH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The work area start address register (WSAR) is a 16-bit readable/writable register that specifies the memory area to be used as a work area. Only the upper bits (A22 to A16) of the start physical address of the work area are set in the WSAH field.

The work area varies depending on the installed memory and the screen size. Some sample calculations are shown below (the work area is the same for 16-bit/pixel and 8-bit/pixel modes).

1. Work coordinate system memory capacity for 320 × 240 screen size
   512 pixels × 256 lines = 131,072 bits: 131,072 bits/8 bits/1024 = 16 kB
2. Work coordinate system memory capacity for 640 × 240 screen size
   1024 pixels × 256 lines = 262,144 bits: 262,144 bits/8 bits/1024 = 32 kB
3. Work coordinate system memory capacity for 640 × 480 screen size
   1024 pixels × 512 lines = 524,288 bits: 524,288 bits/8 bits/1024 = 64 kB

Bits 15 to 7 of WSAR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The WSAH field in WSAR retains its value in a reset.

5.4.6 DMA Transfer Start Address Registers H and L (DMASRH, DMASRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASRH	—	—	—	—	—	—	—	—	—	DMASH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASRL	DMASL (address A15–A1 setting)															—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

DMA transfer start address registers H and L (DMASRH, DMASRL) are 16-bit readable/writable registers that specify the start address of the transfer destination UGM in a DMA transfer.

The upper bits (A22 to A16) of the start address are set in the DMASH field in DMASRH, and the lower bits (A15 to A1) in the DMASL field in DMASRL.

If the value of these registers is modified during a series of DMA operations from the time bits DMA1 and DMA0 in SYSR are set to 10 by the CPU until they are cleared automatically by the Q2i, operation will be unstable.

When bits DMA1 and DMA0 are set to 11, the value in these registers is not referenced. Transfer data passes via the image data entry register (IDER), is converted, and stored sequentially starting at the data transfer start address indicated by the image data transfer start address register (ISAR).

The address (A22 to A1) indicated by the DMASH and DMASL fields is a word address.

Bits 15 to 7 of DMASRH and bit 0 of DMASRL are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The values of the DMASH field in DMASRH and the DMASL field in DMASRL are initialized to all-0 by a reset.

5.4.7 DMA Transfer Word Count Register (DMAWR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW	DMAW
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DMA transfer word count register (DMAWR) is a 16-bit readable/writable register that specifies the number of words (1 word = 16 bits) to be transferred in DMA transfer.

If the value of this register is modified during a series of DMA operations from the time bits DMA1 and DMA0 in SYSR are set to 10 or 11 by the CPU until they are cleared automatically by the Q2i, operation will be unstable.

When bits DMA1 and DMA0 are set to 11, ensure that the total number of pixels in the data set in IDSRX and IDSRY is the same as the total number of pixels corresponding to the transfer word count set in this register. If these values are not the same, DMA transfer will end at the smaller of the two values, with bits DMA1 and DMA0, and bits YUV1 and YUV0, being cleared to 00.

The value of the DMAW bits in DMAWR is initialized to all-0 by a reset.

5.5 Display Control Registers

The display control registers comprise twelve 16-bit registers for setting the display timing, mapped onto addresses (A10–A1) H'013 to H'01E.

5.5.1 Display Window Registers (DSWR (HDS/HDE/VDS/VDE))

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSWR (HDS)	—	—	—	—	—	—	—	HDS	HDS	HDS	HDS	HDS	HDS	HDS	HDS	HDS
Initial value:	—	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSWR (HDE)	—	—	—	—	—	—	HDE	HDE	HDE	HDE	HDE	HDE	HDE	HDE	HDE	HDE
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSWR (VDS)	—	—	—	—	—	—	—	VDS	VDS	VDS	VDS	VDS	VDS	VDS	VDS	VDS
Initial value:	—	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSWR (VDE)	—	—	—	—	—	—	VDE	VDE	VDE	VDE	VDE	VDE	VDE	VDE	VDE	VDE
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The display window registers (DSWR (HDS/HDE/VDS/VDE)) are 16-bit readable/writable registers that specify the horizontal and vertical output timing for the display screen.

1. Horizontal Display Start Position (HDS Bits): Field that specifies the horizontal display start position in dot-clock units.
2. Horizontal Display End Position (HDE Bits): Field that specifies the horizontal display end position in dot-clock units.
3. Vertical Display Start Position (VDS Bits): Field that specifies the vertical display start position in raster line units.

4. Vertical Display End Position (VDE Bits): Field that specifies the vertical display end position in raster line units.

Bits 15 to 9 of DSWR (HDS) and DSWR (VDS) and bits 15 to 10 of DSWR (HDE) and DSWR (VDE) are reserved. Only 0 should be written to these bits (a read will return an undefined value).

DSWR (HDS/HDE/VDS/VDE) bits HDS, HDE, VDS, and VDE retain their values in a reset.

5.5.2 Horizontal Sync Pulse Width Register (HSWR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	HSW	HSW	HSW	HSW	HSW	HSW	HSW
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The horizontal sync pulse width register (HSWR) is a 16-bit readable/writable register that specifies the horizontal signal low-level pulse width in dot-clock units.

Bits 15 to 17 of HSWR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The HSW bits in HSWR retain their values in a reset.

5.5.3 Horizontal Scan Cycle Register (HCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	HC	HC	HC	HC	HC	HC	HC	HC	HC	HC	HC
Initial value:	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The horizontal scan cycle register (HCR) is a 16-bit readable/writable register that specifies the horizontal scan cycle in dot-clock units. In TV sync mode (bits TVM1 and TVM0 set to 10 in DSMR), this register setting must be made so that the $\overline{\text{HSYNC}}$ cycle specified by this register is the same as or greater than the $\overline{\text{EXHSYNC}}$ cycle.

Bits 15 to 11 of HCR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The HC bits in HCR retain their values in a reset.

5.5.4 Vertical Sync Position Register (VSPR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VSP	VSP	VSP	VSP	VSP	VSP	VSP	VSP	VSP	VSP
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The vertical sync position register (VSPR) is a 16-bit readable/writable register that specifies the vertical sync signal start position in raster-line units. In TV sync mode (bits TVM1 and TVM0 set to 10 in DSMR), this register setting must be made so that the $\overline{\text{VSYNC}}$ fall setting position specified by this register is the same as or later than the fall of $\overline{\text{EXVSYNC}}$.

Bits 15 to 10 of VSPR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The VSP bits in VSPR retain their values in a reset.

5.5.5 Vertical Scan Cycle Register (VCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VC	VC	VC	VC	VC	VC	VC	VC	VC	VC
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

The vertical scan cycle register (VCR) is a 16-bit readable/writable register that specifies the vertical scan interval, including the vertical retrace line interval, in raster-line units. In TV sync mode (bits TVM1 and TVM0 set to 10 in DSMR), the $\overline{\text{EXVSYNC}}$ rise detection time limit should be set. If a rise is not detected within the time limit, the result is indicated by the TVR flag in SR.

Bits 15 to 10 of VCR are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The VC bits in VCR retain their values in a reset.

5.5.6 Display Off Output Registers H and L (DOORH, DOORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOORH:	—	—	—	—	—	—	—	—	DOR	DOR	DOR	DOR	DOR	DOR	—	—
Initial value:	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOORL:	DOG	DOG	DOG	DOG	DOG	DOG	—	—	DOB	DOB	DOB	DOB	DOB	DOB	—	—
Initial value:	*	*	*	*	*	*	—	—	*	*	*	*	*	*	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Note: * Value is retained.

Display off output registers H and L (DOORH, DOORL) are 16-bit readable/writable registers that specify the display data to be output when display is off. A 6-bit setting is made for each of the RGB components, in bits DOR, DOG, and DOB.

Bits 15 to 8, 1, and 0 of DOORH, and bits 9, 8, 1, and 0 of DOORL are reserved. Only 0 should be written to these bits.

DOORH/L bits DOR, DOG, and DOB retain their values in a reset.

5.5.7 Color Detection Registers H and L (CDERH, CDERL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDERH:	—	—	—	—	—	—	—	—	CDR	CDR	CDR	CDR	CDR	CDR	—	—
Initial value:	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDERL:	CDG	CDG	CDG	CDG	CDG	CDG	—	—	CDB	CDB	CDB	CDB	CDB	CDB	—	—
Initial value:	*	*	*	*	*	*	—	—	*	*	*	*	*	*	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Note: * Value is retained.

Color detection registers H and L (CDERH, CDERL) are 16-bit readable/writable registers. When the display data output from pins DD17 to DD0 matches the values set in these registers, 1 is output from the CDE pin.

Bits 15 to 8, 1, and 0 of CDERH, and bits 9, 8, 1, and 0 of CDERL are reserved. Only 0 should be written to these bits.

CDERH/L bits CDR, CDG, and CDB retain their values in a reset.

5.6 Rendering Control Registers

The rendering control registers comprise two 16-bit registers related to rendering control, mapped onto addresses (A10–A1) H'01F to H'020.

5.6.1 Command Status Registers H and L (CSTRH, CSTRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSTRH	—	—	—	—	—	—	—	—	—	CSTH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSTRL	CSTL (address A15–A1 setting)															—
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	—
Read/Write:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	—

Note: * Value is retained.

Command status registers H and L (CSTRH, CSTRL) are 16-bit read-only registers that store the address of the command word (op code word) being executed when frame switching is performed.

The upper bits (A22 to A16) of the command word address are indicated by the CSTH field, and the lower bits (A15 to A1) by the CSTL field. The address indicated by the CSTH and CSTL fields is a word address.

Bits 15 to 7 of CSTRH and bit 0 of CSTRL are reserved. These bits always read 0.

The CSTH field in CSTRH and the CSTL field in CSTRL retain their values in a reset.

5.7 Input Control Registers

The input control registers comprise five 16-bit registers related to the control of input data conversion, mapped onto addresses (A10–A1) H'021 to H'025. The settings in these registers are valid when the setting of bits YUV1 and YUV0 in the input data conversion mode register (IEMR) is 01 or 10.

5.7.1 Image Data Transfer Start Address Registers H and L (ISARH, ISARL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISARH	—	—	—	—	—	—	—	—	—	ISAH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISARL	ISAL (address A15–A1 setting)															—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Note: * Value is retained.

Image data transfer start address registers H and L (ISARH, ISARL) are 16-bit readable/writable registers that specify the image data transfer destination as a physical address when the setting of bits YUV1 and YUV0 is 01 or 10. The upper bits (A22 to A16) of the start address are set in the ISAH field, and the lower bits (A15 to A1) in the ISAL field. The address indicated by the ISAH and ISAL fields is a word address.

If the value of these registers is modified during a series of data conversion operations from the time bits YUV1 and YUV0 are set to 01 or 10 by the CPU until YUV mode is cleared automatically by the Q2i, operation will be unstable.

Bits 15 to 7 of ISARH and bit 0 of ISARL are reserved. Only 0 should be written to these bits.

The values of the ISAH field in ISARH and the ISAL field in ISARL are initialized to all-0 by a reset.

5.7.2 Image Data Size Registers X and Y (IDSRX, IDSRY)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDSRX	—	—	—	—	—	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX	IDSX
Initial value:	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0
Read/Write:	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDSRY	—	—	—	—	—	—	IDSY	IDSY	IDSY	IDSY	IDSY	IDSY	IDSY	IDSY	IDSY	IDSY
Initial value:	—	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Image data size registers X and Y (IDSRX, IDSRY) are 16-bit readable/writable registers that specify the image data X size and Y size in pixel units when the setting of bits YUV1 and YUV0 is 01 or 10. An even number should be set for the X size (IDSX0 bit = 0).

If the value of these registers is modified during a series of data conversion operations from the time bits YUV1 and YUV0 are set to 01 or 10 by the CPU until YUV mode is cleared automatically by the Q2i, operation will be unstable.

Bits 15 to 11 of IDSRX and bits 15 to 10 of IDSRY are reserved. Only 0 should be written to these bits.

The values of the IDSX and IDSY bits in IDSRX/Y are initialized to all-0 by a reset.

5.7.3 Image Data Entry Register (IDER)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE	IDE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

The image data entry register (IDER) is a 16-bit write-only register that comprises the entry in which image data is input when the setting of bits YUV1 and YUV0 is 01 or 10.

IDER is initialized to H'0000 by a reset.

5.8 Memory Control Registers 2

Memory control registers 2 comprise two 16-bit registers related to background screen control, mapped onto addresses (A10–A1) H'026 and H'027.

5.8.1 Background Start Coordinate Registers X and Y (BGSRX, BGSRY)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BGSRX	—	—	—	—	—	—	BGSX	BGSX	BGSX	BGSX	BGSX	BGSX	BGSX	BGSX	BGSX	BGSX
Initial value:	—	—	—	—	—	—	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BGSRY	—	—	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY	BGSY
Initial value:	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Value is retained.

Background start coordinate registers X and Y (BGSRX, BGSRY) are 16-bit readable/writable registers that specify the background screen start coordinates. The settings should be made so that the background screen does not overlap the frame buffers. Bits 15 to 10 in BGSRX and bits 15 and 14 in BGSRY are reserved. Only 0 should be written to these bits.

The BGSX bits in BGSRX and BGSY bits in BGSRY retain their values in a reset.

5.9 Rendering Control Registers 2

Rendering control registers 2 comprise twelve 16-bit registers related to rendering control, mapped onto addresses (A10–A1) H'040 to H'04B

When reading these registers, first set BRCL to 1 in the status clear register (SRCR), then set RBRK to 1 in the system control register (SYSR), and wait until BRK is set to 1 in the status register (SR) before performing the read. If this procedure is not followed, an undefined value will be read.

5.9.1 Current Pointer Registers X and Y (CURRX, CURRY)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRX	—	—	—	XC	XC	XC	XC	XC	XC	XC	XC	XC	XC	XC	XC	XC
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRY	—	—	—	YC	YC	YC	YC	YC	YC	YC	YC	YC	YC	YC	YC	YC
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Value is retained.

Current pointer registers X and Y (CURRX, CURRY) are 16-bit read-only registers that indicate the current pointer coordinates.

Bits 15 to 13 of CURRX and CURRY are reserved. These bits always read 0.

The XC bits in CURRX and YC bits in CURRY retain their values in a reset.

5.9.2 Local Offset Registers X and Y (LCOX, LCOY)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCOX	—	—	—	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO	XO
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCOY	—	—	—	YO	YO	YO	YO	YO	YO	YO	YO	YO	YO	YO	YO	YO
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Value is retained.

Local offset registers X and Y (LCOX, LCOY) are 16-bit read-only registers that indicate the offset coordinates.

Bits 15 to 13 of LCOX and LCOY are reserved. These bits always read 0.

The XO bits in LCOX and YO bits in LCOY retain their values in a reset.

5.9.3 User Clipping Area Registers [UCL (UXMIN/UYMIN/UXMAX/UYMAX)]

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCL Upper-left X:	—	—	—	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN	UXMIN
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCL Upper-left Y:	—	—	—	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN	UYMIN
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCL Lower-right X:	—	—	—	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX	UXMAX
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCL Lower-right Y:	—	—	—	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX	UYMAX
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Value is retained.

The user clipping area registers [UCL (UXMIN/UYMIN/UXMAX/UYMAX)] are 16-bit read-only registers that indicate the user clipping area.

Bits 15 to 13 of UCL (UXMIN/UYMIN/UXMAX/UYMAX) are reserved. These bits always read 0.

The UXMIN, UYMIN, UXMAX, and UYMAX bits in the UCL registers retain their values in a reset.

5.9.4 System Clipping Area Registers [SCL (SXMAX/SYMAX)]

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCL Lower-right X:	—	—	—	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX	SXMAX
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCL Lower-right Y:	—	—	—	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX	SYMAX
Initial value:	—	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*
Read/Write:	—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Value is retained.

The system clipping area registers [SCL (SXMAX/SYMAX)] are 16-bit read-only registers that indicate the system clipping area.

Bits 15 to 13 of SCL (SXMAX/SYMAX) are reserved. These bits always read 0.

The SXMAX and SYMAX bits in the SCL registers retain their values in a reset.

5.9.5 Return Address Registers H, L (RTNH, RTNL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTNH	—	—	—	—	—	—	—	—	—	RTNH (address A22–A16 setting)						
Initial value:	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
Read/Write:	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTNL	RTNL (address A15–A1 setting)															—
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Note: * Value is retained.

Return address registers H and L (RTNH, RTNL) are 16-bit readable/writable registers specify the return address.

The upper bits (A22 to A16) of the start address are set in the RTNH field, and the lower bits (A15 to A1) in the RTNL field.

The address (bits A22 to A1) indicated by the RTNH and RTNL fields is a word address.

Bits 15 to 7 of RTNH and bit 0 of RTNL are reserved. Only 0 should be written to these bits (a read will return an undefined value).

The RTNH field in RTNH and the RTNL field in RTNL retain their values in a reset.

5.10 Color Palette

The color palette is mapped onto addresses (A10–A1) H'100 to H'2FF. Settings can be made for 256 pixels, with 6 bits each for R, G, and B. The color palette is only valid when the GBM bit in the rendering mode register (REMR) is 0 (8 bits/pixel). When the GBM bit is set to 1 (16 bits/pixel), the color palette values set when the GBM bit is 0 (8 bits/pixel) are lost.

5.10.1 Color Palette Registers H, L000–255 (CP000RH, L–CP255H, L)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP000RH	—	—	—	—	—	—	—	—	R000 (Red: 6 bits)						—	—
Initial value:	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP000RL	G000 (Green: 6 bits)						—	—	B000 (Blue: 6 bits)						—	—
Initial value:	*	*	*	*	*	*	—	—	*	*	*	*	*	*	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP001RH	—	—	—	—	—	—	—	—	R001 (Red: 6 bits)						—	—
Initial value:	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP001RL	G001 (Green: 6 bits)						—	—	B001 (Blue: 6 bits)						—	—
Initial value:	*	*	*	*	*	*	—	—	*	*	*	*	*	*	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

⋮

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP255RH	—	—	—	—	—	—	—	—	R255 (Red: 6 bits)						—	—
Initial value:	—	—	—	—	—	—	—	—	*	*	*	*	*	*	—	—
Read/Write:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP255RL	G255 (Green: 6 bits)						—	—	B255 (Blue: 6 bits)						—	—
Initial value:	*	*	*	*	*	*	—	—	*	*	*	*	*	*	—	—
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Note: * Value is retained.

Color Palette Registers H, L000–255 (CP000RH, L to CP255RH, L) are 32-bit readable/writable registers. These registers can be used as a color palette capable of simultaneously displaying 256 colors out of 262,144. The settings are valid when the GBM bit is 0.

The color palette is controlled in 2-word units comprising one pixel. The same units must therefore be used for accesses to the color palette registers.

When writing to color palette registers, first write to the R register, then to the G and B registers. The R register value is reflected as the new color palette set value when the G and B registers are set.

When reading color palette registers, first read the R register, then the G and B registers.

When accessing color palette registers, it is not possible to access another Q2i register between the R register and the G and B registers.

In modes in which GBM = 1 (16 bits/pixel), these registers are used as part of the internal display circuitry. Therefore, color palette register set values are lost when the GBM bit is set to 1.

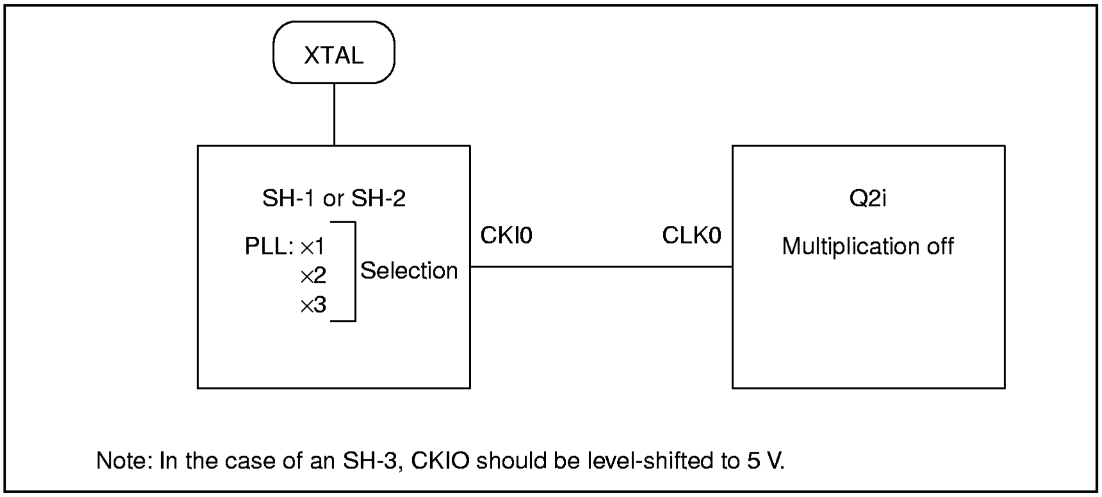
For the same reason, if a color palette is accessed when GBM = 1, the color palette access will not be completed and the Q2i will output a continuous CPU wait signal.

Section 6 Usage Notes

6.1 CPU Clock and Q2i-CLK0

Input a clock that satisfies the following condition to the CLK0 pin. CK and CLK0 can be asynchronous.

High-level interval (t_{RDHW} or t_{WRHW}) of $\overline{RD}$ or $\overline{WE0}$, $\overline{WE1}$ input to Q2i ≥ Q2i operating clock cycle



Note: In the case of an SH-3, CKIO should be level-shifted to 5 V.

Figure 6-1 Example of Connection for Synchronous Operation

6.2 Note on Use of Auto Display Change Mode

When using auto display change mode, if Q2i drawing is aborted due to a frame change, invalid drawing of 1 to 4 dots may be performed when the next display list is executed, depending on when drawing is halted (because of invalid data remaining in the drawing unit).

Permanent Remedy: Adjust the display list in the system design stage so that drawing processing is always completed before a frame change.

6.3 Power-On Sequence

The CLK0, CLK1, and RESET signal timing when powering on is shown in the figure 6-2. The time from the rise of VCCn until the rise of CLK0 and CLK1 should be a maximum of 100 ms, and the time from the rise of VCCn until the rise of RESET, a minimum of 100 ms. If CLK0 and CLK1 are stopped for a long period (100 ms or more) after powering on, the chip may be damaged.

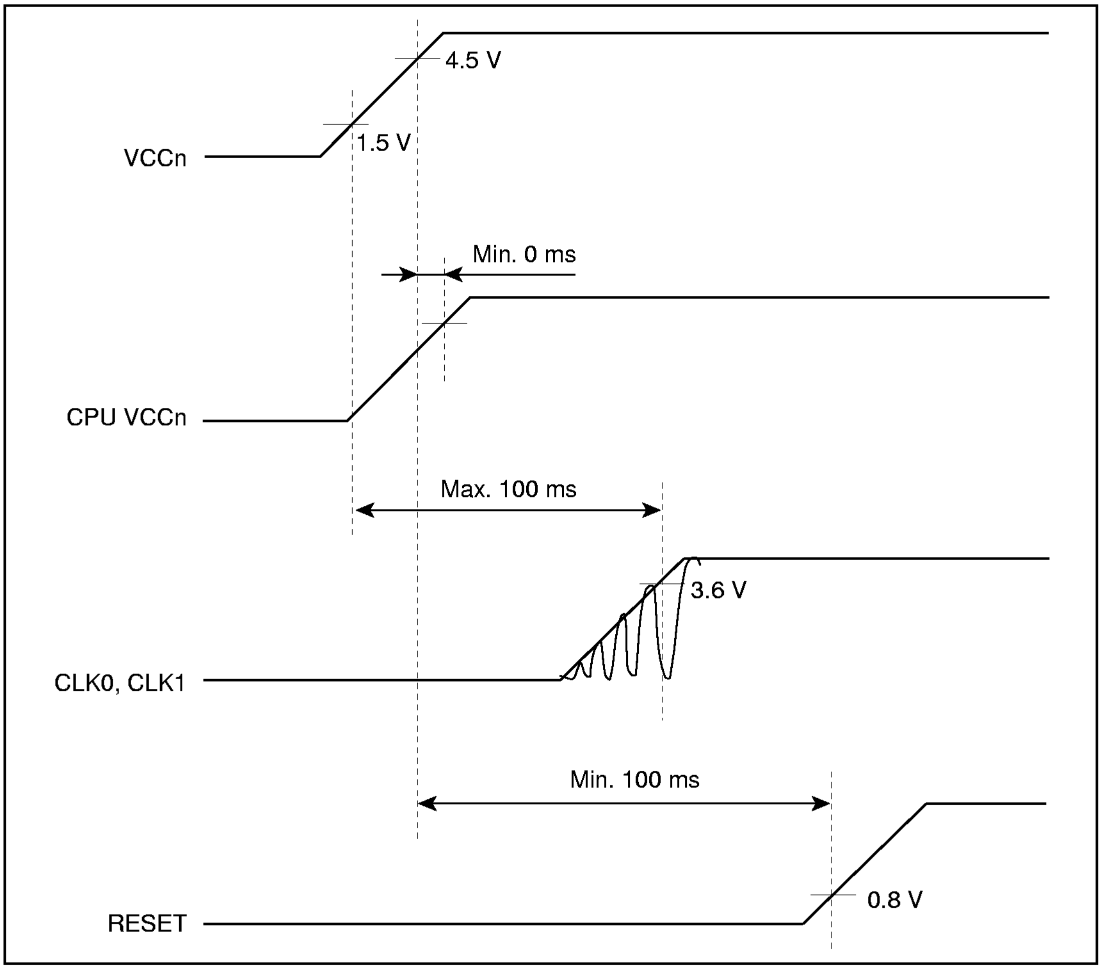


Figure 6-2 Power-On Sequence

6.4 Q2i Internal Buffers

The Q2i has three internal buffers—a command buffer, source buffer, and work buffer—as shown in figure 6-3.

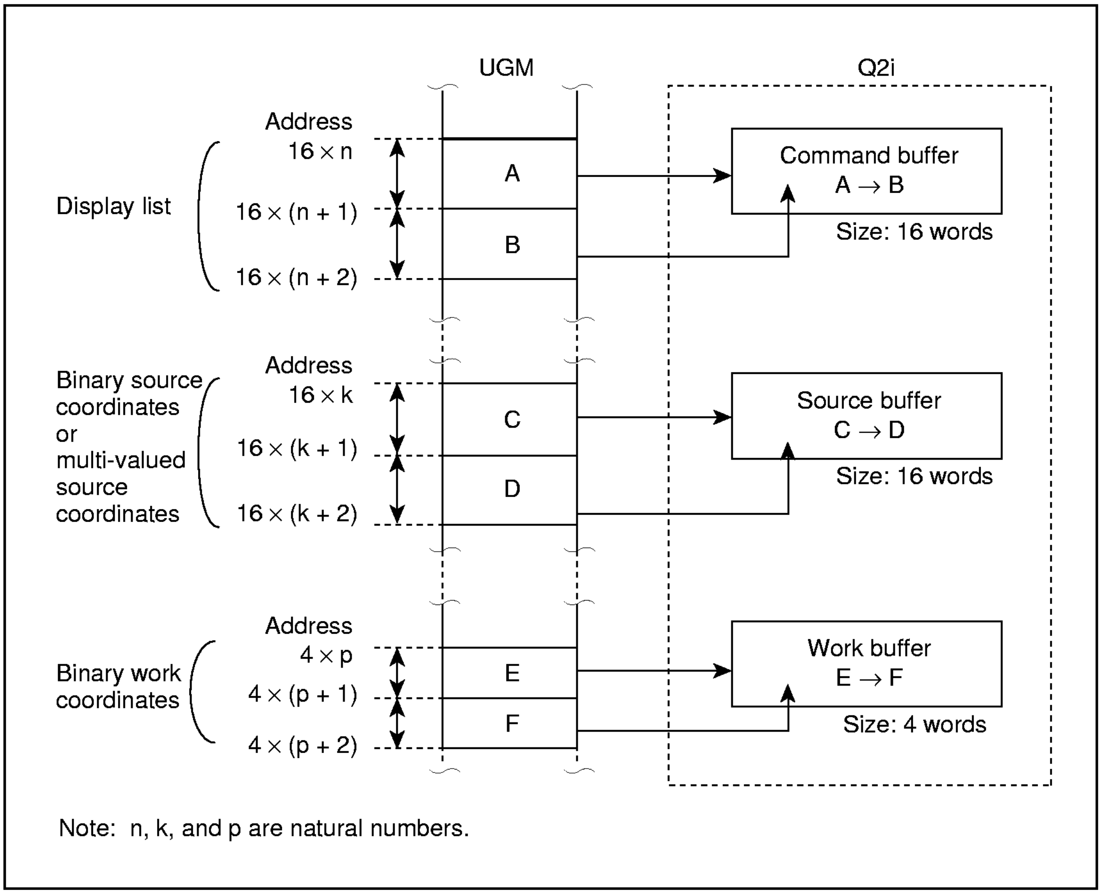


Figure 6-3 Updating of Q2i's Internal Buffers

These buffers are used by the Q2i to temporarily store data held in the UGM. The Q2i uses the data stored in these buffers when executing drawing.

The functions of these buffers are shown in (a) to (c) below.

(a) Command buffer: Used by the Q2i to store a display list held in the UGM
(b) Source buffer: Used by the Q2i to store a binary source or multi-valued source held in the UGM
(c) Work buffer: Used by the Q2i when performing drawing at binary work coordinates in the UGM

The size of these buffers is fixed: 16 words each for the command buffer and source buffer, and 4 words for the work buffer.

Therefore, when the Q2i fetches data in the UGM into a buffer, it performs buffer updating by managing the read destination address of each buffer.

An outline of buffer updating is given in (d) to (f) below.

(d) When the Q2i uses the command buffer, the buffer contents are updated each time the UGM address value indicated by the Q2i exceeds a 16-word boundary.
(e) When the Q2i uses the source buffer, the buffer contents are updated each time the UGM address value indicated by the Q2i exceeds a 16-word boundary.
(f) When the Q2i uses the work buffer, the buffer contents are updated each time the UGM address value indicated by the Q2i exceeds a 4-word boundary.

Problems: The following problems occur depending on the commands used.

(a) When using the POLYGON4B, PLINE, and RPLINE commands
If a binary source within a 16-word boundary is used without updating command parameters SOURCE ADDRESSH and SOURCE ADDRESSL, the source buffer is not updated since the address indicated by the Q2i does not exceed a 16-word boundary. In a drawing operation, drawing is performed using the binary source stored in the source buffer.

(b) When using the POLYGON4A command
If a multi-valued source within a 16-word boundary is used without updating command parameters TXS and TYS or the source area start address, the source buffer is not updated since the address indicated by the Q2i does not exceed a 16-word boundary. In a drawing operation, drawing is performed using the multi-valued source stored in the source buffer.

(c) When using the FTRAP, RFTRAP, CLRW, LINEW, and RLINEW commands
If a binary source within a 4-word boundary is used without updating the work area start address, the work buffer is not updated since the address indicated by the Q2i does not exceed a 4-word boundary. In a drawing operation, drawing is performed using the work data stored in the work buffer.

Remedy: As a method of updating the source buffer contents, either use a source pattern that exceeds 16 words or use a source pattern at a location that exceeds a 16-word boundary.

As a method of updating the work buffer, either use a work pattern that exceeds 4 words or use a work pattern at a location that exceeds a 4-word boundary.

6.5 Note on Changing TV Synchronization Mode

When B'01 is set in the TV synchronization mode bits (TVM) in the display mode register (DSMR) and a transition is made to synchronization system switching mode, set the display reset bit (DRES) to 1 and clear the display enable bit (DEN) to 0 in the system control register before making the transition to synchronization system switching mode.

This procedure provides for the HD64412 to perform UGM refreshing in synchronization system switching mode.

The procedure is shown below. The procedure is performed in order from step 1 to 3. The HD64412 performs UGM refreshing immediately.

1. Set DRES = 1 and DEN = 0.
2. Set TVM1 = 0 and TVM0 = 1.

The procedure for switching from synchronization system switching mode to another TV synchronization mode is shown in 3 to 6 below.

3. Input the clock to CLK1. When setting TVM1 = 1, TVM0 = 0, also input signals to the $\overline{\text{EXHSYNC}}$, $\overline{\text{EXVSYNC}}$, and $\overline{\text{ODDF}}$ pins.
4. If the display size is to be changed, set values in the Q2i's address-mapped registers.
5. The input clock from the CLK1 pin becomes valid on setting TVM1 = 0, TVM0 = 0, or TVM1 = 1, TVM0 = 0.
6. Set DRES = 0, DEN = 1. When an internal update is performed, the Q2i begins display.

6.6 POLYGON4A Source Reference Position

Problem: If the POLYGON4A command is used under the following conditions, a source reference error occurs, and the data drawn at rendering coordinate pixel $X = (64 \times t) + 1$ is the same as that drawn at the next pixel (where $t \geq 1$).

- Conditions
  Rendering attribute: WORK = 1 or STYL = 1
  Source start point TXS: $(32 \times p) + 1$ (where $p \geq 0$)

Remedy: When using the POLYGON4A command under these conditions, set a value other than $(32 \times p) + 1$ for source start point TXS.

Section 7 Electrical Characteristics

7.1 Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} *1	–0.3 to +7.0	V
Input voltage	V_{in} *1	–0.3 to V_{CC} + 0.3	V
Permissible output low current	$\| I_{OL} \|$ *2	2.0	mA
Total permissible output low current	$\| \_I_{OL} \|$ *3	90	mA
Permissible output high current	$\| -I_{OH} \|$ *2	2.0	mA
Total permissible output high current	$\| \_ (-I_{OH}) \|$ *3	90	mA
Operating temperature	T_{opr}	0 to 70	°C
Storage temperature	T_{stg}	–55 to +125	°C

Notes: 1. Value based on GND = 0 V. Includes $CPUV_{CC}$ and $PLLV_{CC}$.

2. The permissible output current is the maximum value of the current drawn in or flowing out from one output pin and one input/output pin.

3. The total permissible output current is the sum of currents drawn in or flowing out from output pins and input/output pins.

Usage Note: Permanent damage to the chip may result if absolute maximum ratings are exceeded. In normal operation, it is advisable to observe the recommended operating conditions. Exceeding these conditions may adversely affect the reliability of the chip.

7.2 Recommended Operating Conditions

7.2.1 Recommended Operating Conditions

Table 7-2 Recommended Operating Conditions

Item	Symbol		Min	Typ	Max	Unit
Power supply voltage	V_{CC}*1, $PLLV_{CC}$*2		4.75	5.0	5.25	V
	$CPUV_{CC}$*3	5 V operation	4.75	5.0	5.25	V
		3.3 V operation	3.0	3.3	3.6	V
Input low voltage (except CLK0, CLK1)	V_{ILT}*1		0	—	0.8	V
Input low voltage (CLK0, CLK1)	V_{ILC}*1		0	—	0.8	V
Input high voltage (except CLK0, CLK1)	V_{IHT}*1		2.2	—	V_{CC}	V
Input high voltage (CLK0, CLK1)	V_{IHC}*1		$0.8 \times V_{CC}$	—	V_{CC}	V
Operating temperature	T_{opr}		0	25	70	°C

Notes: 1. Value based on GND = 0 V.

2. Value based on PLLGND = 0 V.

3. Value based on CPUGND = 0 V.

7.3 Electrical Characteristics Test Methods

7.3.1 Timing Testing

The output low voltage for timing testing is 1.5 V. The output high voltage for timing testing is also 1.5 V.

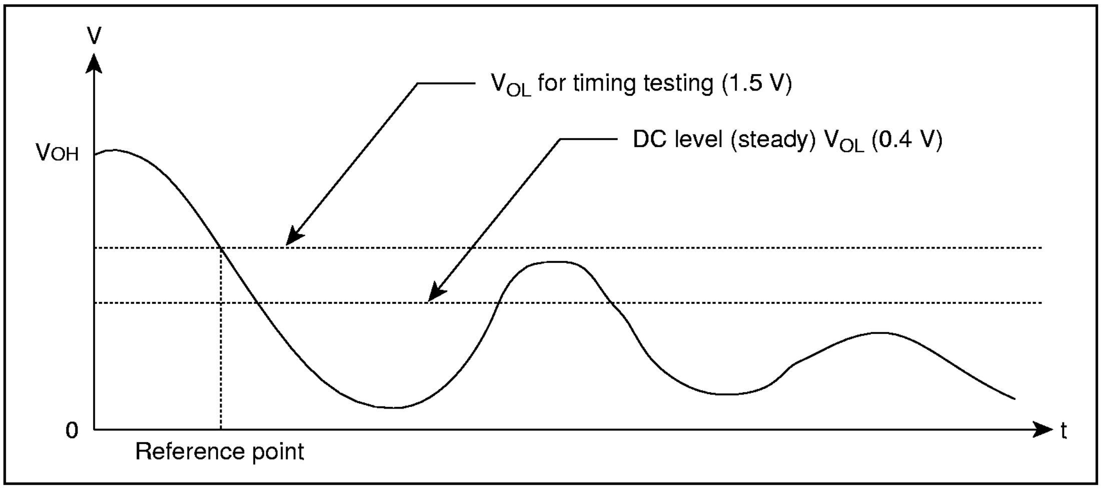
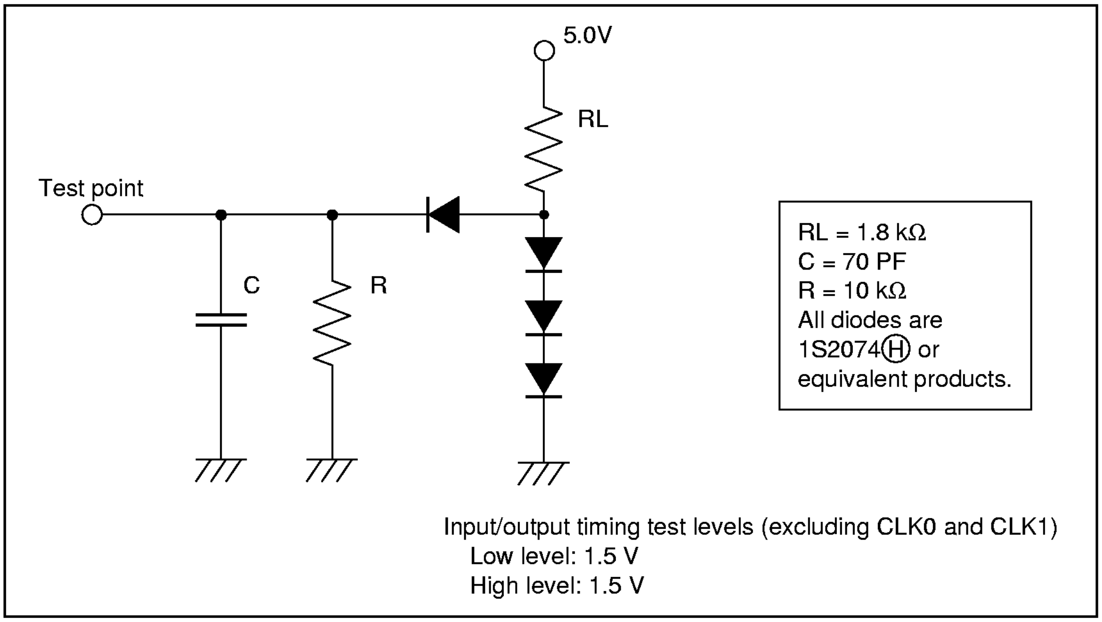

V
VOH
0
V_{OL} for timing testing (1.5 V)
DC level (steady) V_{OL} (0.4 V)
Reference point
t

Figure 7-1 Basis of V_{OL} Timing Testing

7.3.2 Test Load Circuit (All Output and Input/Output Pins)



RL = 1.8 kΩ
C = 70 PF
R = 10 kΩ
All diodes are
1S2074Ⓗ or
equivalent products.

Input/output timing test levels (excluding CLK0 and CLK1)
Low level: 1.5 V
High level: 1.5 V

Figure 7-2 Test Load Circuit

7.4 Electrical Characteristics

7.4.1 DC Characteristics

Table 7-3 DC Characteristics

(Unless otherwise indicated, V_{CC} = CPUV_{CC} = PLLV_{CC} = 5.0 V ±5%, GND = CPUGND = PLLGND = 0 V, Ta = 0 to +70°C)

Item	Pin Names	Symbol	Min	Max	Unit	Test Conditions
Input high voltage (CMOS level)	I1	V_{IHC}	0.8 × V_{CC}	V_{CC} + 0.3	V	
Input low voltage (CMOS level)		V_{ILC}	–0.3	0.8		
Input high voltage (TTL level)	I2, IO1, IO2	V_{IHT}	2.2	V_{CC} + 0.3	V	
Input low voltage (TTL level)		V_{ILT}	–0.3	0.8		
Input leakage current	I1, I2	I_{in}	–2.5	2.5	µA	V_{in} = 0 to V_{CC}
Three-state leakage current (off state)	IO1, IO2	I_{TSI}	–10	10		V_{in} = 0.4 to V_{CC}
Output high voltage (5 V)	IO1, O1	V_{OH}	V_{CC} – 1.0	—	V	I_{OH} = –400 µA
Output high voltage	IO2, O2	V_{OH}	CPUV_{CC} – 1.0	—		I_{OH} = –400 µA CPUV_{CC} = 5.0 V ±5%
			CPUV_{CC} – 0.5	—		I_{OH} = –200 µA CPUV_{CC} = 3.0 V to 3.6 V
Output low voltage	IO1, IO2, O1, O2	V_{OL}	—	0.4		I_{OL} = 2.0 mA
Input capacitance	IO1, IO2	C_{in}	—	20	pF	V_{in} = 0 V
	I1, I2		—	20		Ta = 25°C f = 1.0 MHz
Current dissipation		I_{CC}	—	180	mA	Data bus operation in progress/display operation in progress/command execution in progress

Note: The symbols used in table 7-3 are explained below.

Item	Input	Output	High-Z	Pull-up	Pin Names
I1	CMOS	—	—	—	CLK1, CLK0
I2	TTL	—	—	—	MODE2–0, $\overline{\text{RESET}}$, A22–A1, $\overline{\text{CS1–0}}$, $\overline{\text{RD}}$, $\overline{\text{WE1–0}}$, $\overline{\text{DACK}}$
IO1	TTL	CMOS	Yes	—	$\overline{\text{HSYNC/EXHSYNC}}$, $\overline{\text{VSYNC/EXVSYNC}}$, $\overline{\text{ODDF}}$, MD15–0
IO2	TTL	CMOS	Yes	—	D15–D0
O1	—	CMOS	—	—	CDE, DD17–0, DCLK, FCLK, DISP, MA11–0, $\overline{\text{MWE}}$, $\overline{\text{MRAS1–0}}$, $\overline{\text{MLCAS}}$, $\overline{\text{MUCAS}}$, $\overline{\text{MOE}}$, $\overline{\text{CSYNC}}$
O2	—	CMOS	—	—	$\overline{\text{DREQ}}$, $\overline{\text{IRL}}$, $\overline{\text{WAIT}}$

7.4.2 AC Characteristics

Unless otherwise specified, Vcc = CPUVcc = PLLVcc = 5.0 V ±5%, GND = CPUGND = PLLGND = 0 V, Ta = 0 to +70°C

(1) Input Clocks

Table 7-4 Input Clocks (1) (Pins MODE2 to MODE0 = 011: Multiplication Off)

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
CLK0 cycle time	t_{cyc0}	30.3	50	ns	Figure 7-3	
CLK0 high-level pulse width	t_{C0PWH}	10.1	—	ns		
CLK0 low-level pulse width	t_{C0PWL}	10.1	—	ns		
CLK0 duty	t_{C0DT}	$0.5t_{cyc0} - 1.7$	$0.5t_{cyc0} + 1.7$	ns		
CLK1 cycle time	t_{cyc1}	60.6	200	ns		
CLK1 high-level pulse width	t_{C1PWH}	25.3	—	ns		
CLK1 low-level pulse width	t_{C1PWL}	25.3	—	ns		
CLK1 duty	t_{C1DT}	$0.5t_{cyc0} - 1.7$	$0.5t_{cyc0} + 1.7$	ns		
CLK rise time	t_{cr}	—	5	ns		
CLK fall time	t_{cf}	—	5	ns		

Table 7-4 Input Clocks (2) (Pins MODE2 to MODE0 = 000, 001, 010: Multiplication On)

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
CLK0 cycle time	t_{cyc}	30.3	50	ns	Figure 7-4	×1
CLK0 cycle time	t_{cyc}	60.6	100	ns		×2
CLK0 cycle time	t_{cyc}	121.2	200	ns		×4
CLK0 high-level pulse width	t_{CPWH}	10.1	—	ns		
CLK0 low-level pulse width	t_{CPWL}	10.1	—	ns		
CLK delay time 1	t_{CLKD1}	—	13	ns		
CLK delay time 2	t_{CLKD2}	—	$t_{cyc}/4 + 14.7$	ns		
CLK delay time 3	t_{CLKD3}	—	$t_{cyc}/2 + 14.7$	ns		
CLK delay time 4	t_{CLKD4}	—	$3t_{cyc}/4 + 14.7$	ns		
CLKi cycle time	t_{cyc0}	30.3	50	ns		
CLKi high-level pulse width	t_{CiPWH}	10.1	—	ns		
CLKi low-level pulse width	t_{CiPWL}	10.1	—	ns		
CLK1 cycle time	t_{cyc1}	60.6	200	ns		
CLK1 high-level pulse width	t_{C1PWH}	25.3	—	ns		
CLK1 low-level pulse width	t_{C1PWL}	25.3	—	ns		
CLK1 duty	t_{C1DT}	$0.5t_{cyc1} - 1.7$	$0.5t_{cyc1} + 1.7$	ns		
CLK rise time	t_{cr}	—	5	ns		
CLK fall time	t_{cf}	—	5	ns		

(2) Reset

Table 7-5 Reset

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
$\overline{\text{RESET}}$ low pulse width	t_{RESW}	40	—	tcyc1	Figure 7-5	
$\overline{\text{RESET}}$ acceptance indeterminate time 1	t_{RES1}	5	—	ns		
$\overline{\text{RESET}}$ acceptance indeterminate time 2	t_{RES2}	5	—	ns		
DCLK rise delay time from CLK1	t_{DCRD}	—	30	ns		
DCLK fall delay time from CLK1 1	t_{DCFD1}	—	30	ns		DOT = 1
DCLK fall delay time from CLK0 0	t_{DCFD0}	—	30	ns		DOT = 0
FCLK rise delay time from CLK1	t_{FCRD}	—	30	ns		
FCLK fall delay time from CLK1	t_{FCFD}	—	30	ns		
DCLK cycle time	$t_{cyc\ D}$	$2t_{cyc1}$	$2t_{cyc1}$	ns		DOT = 1
DCLK cycle time	$t_{cyc\ D}$	$1t_{cyc1}$	$1t_{cyc1}$	ns		DOT = 0

(3) CPU Read Cycle

Table 7-6 CPU Read Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
Address setup time	t_{ADS}	0	—	ns	Figure 7-6	
Address hold time	t_{ADH}	0	—	ns		
$\overline{\text{CSn}}$ setup time	t_{CSS}	0	—	ns		1
$\overline{\text{CSn}}$ hold time	t_{CSH}	0	—	ns		2
$\overline{\text{WAIT}}$ cycle start time 1	t_{WAS1}	—	$3t_{cyc0} + 25$	ns		
$\overline{\text{RD}}$ high-level width	t_{RDHW}	t_{cyc0}	—	ns		
Read data setup time for $\overline{\text{WAIT}}$	t_{RDDWS}	10	—	ns		
$\overline{\text{WAIT}}$ drive time	t_{WAD}	t_{cyc0}	—	ns		
Read data turn-on time	t_{RDDON}	0	—	ns		
Read data hold time	t_{RDDH}	4	—	ns		
Read data turn-off time	t_{RDDOF}	4	—	ns		
$\overline{\text{WE}}$ high-level width	t_{WEHW}	t_{cyc0}	—	ns		

Notes: 1. If the fall of $\overline{\text{CSn}}$ is later than the fall of $\overline{\text{RD}}$, the specifications of t_{ADS}, t_{WAS1}, t_{RDDON}, and t_{WEHW} are from the fall of $\overline{\text{CSn}}$.

2. If the rise of $\overline{\text{CSn}}$ is earlier than the rise of $\overline{\text{RD}}$, the specifications of t_{ADH}, t_{RDDH}, t_{RDDOF}, and t_{WEHW} are from the rise of $\overline{\text{CSn}}$.

(4) CPU Write Cycle

Table 7-7 CPU Write Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
Address setup time	t_{ADS}	0	—	ns	Figure 7-7	
Address hold time	t_{ADH}	0	—	ns		
$\overline{\text{CSn}}$ setup time	t_{CSS}	0	—	ns		1
$\overline{\text{CSn}}$ hold time	t_{CSH}	0	—	ns		2
$\overline{\text{RD}}$ high-level width	t_{RDHW}	t_{cyc0}	—	ns		
$\overline{\text{WAIT}}$ drive time	t_{WAD}	t_{cyc0}	—	ns		
$\overline{\text{WAIT}}$ cycle start time 2	t_{WAS2}	—	$3t_{cyc0} + 25$	ns		
$\overline{\text{WE}}$ high-level width	t_{WEHW}	t_{cyc0}	—	ns		
Write data setup time for $\overline{\text{WE}}$	t_{WRDES}	2_{tcyc0}	—	ns		
Write data hold time	t_{WRDH}	0	—	ns		
Write data turn-off time	t_{WRDOF}	—	30	ns		

Notes: 1. If the fall of $\overline{\text{CSn}}$ is later than the fall of $\overline{\text{WE}}$, the specifications of t_{ADS}, t_{RDHW}, and t_{WAS2} are from the fall of $\overline{\text{CSn}}$.

2. If the rise of $\overline{\text{CSn}}$ is earlier than the rise of $\overline{\text{WE}}$, the specifications of t_{ADH}, t_{RDHW}, t_{WRDES}, t_{WRDH}, and t_{WRDOF} are from the rise of $\overline{\text{CSn}}$.

(5) DMA Write Cycle

Table 7-8 DMA Write Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
$\overline{\text{RD}}$ high-level width	t_{RDHW}	t_{cyc0}	—	ns	Figure 7-8	
$\overline{\text{RD}}$ low-level width	t_{RDLW}	$3t_{cyc0}$	—	ns		
Write data hold time	t_{WRDH}	0	—	ns		
Write data turn-off time	t_{WRDOF}	—	30	ns		
Write data setup time for $\overline{\text{RD}}$	t_{WRDRS}	$2t_{cyc0}$	—	ns		
$\overline{\text{DREQ}}$ negate time	t_{DAN}	—	$3t_{cyc0} + 25$	ns		
$\overline{\text{DREQ}}$ assert time	t_{DDA}	$3t_{cyc0} + 25$	—	ns		
$\overline{\text{DACK}}$ setup time	t_{DAS}	0	—	ns		1
$\overline{\text{DACK}}$ hold time	t_{DAH}	0	—	ns		2

Notes: 1. If the fall of $\overline{\text{DACK}}$ is later than the fall of $\overline{\text{RD}}$, the specification of t_{RDLW} is from the fall of $\overline{\text{DACK}}$.

2. If the rise of $\overline{\text{DACK}}$ is earlier than the rise of $\overline{\text{RD}}$, the specifications of t_{RDLW}, t_{WRDH}, t_{WRDOF}, and t_{WRDRS} are from the rise of $\overline{\text{DACK}}$.

(6) Interrupt Output

Table 7-9 Interrupt Output

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
$\overline{\text{IRL}}$ delay time	t_{IRD}	—	25	ns	Figure 7-9	
$\overline{\text{IRL}}$ low-level width	t_{IRLW}	$2t_{cyc0}$	—	ns		

(7) UGM Read Cycle

Table 7-10 UGM Read Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
RAS delay time	t_{RASD}	—	25	ns	Figure 7-10	
CAS delay time	t_{CASD}	—	25	ns		
Row address setup time	t_{ROWS}	0	—	ns		
Row address hold time	t_{ROWH}	15	—	ns		
Column address setup time	t_{COMS}	6	—	ns		
Column address hold time	t_{COMH}	10	—	ns		
OE delay time	t_{OED}	—	25	ns		
MD turn-on time	t_{MDON}	0	—	ns		
MD turn-off time	t_{MDOF}	—	35	ns		
MD input setup time	t_{MDIS}	5	—	ns		
MD input hold time	t_{MDIH}	3	—	ns		
MD input time 1	t_{MDI1}	—	$t_{cyc0} - 5$	ns		
MD input hold time 1	t_{MDH1}	3	—	ns		
Column address delay time	t_{CADD}	—	20	ns		
OE rise delay time from RAS rise	t_{OERD}	0	—	ns		

(8) UGM Write Cycle

Table 7-11 UGM Write Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
RAS delay time	t_{RASD}	—	25	ns	Figure 7-11	
CAS delay time	t_{CASD}	—	25	ns		
Row address setup time	t_{ROWS}	0	—	ns		
Row address hold time	t_{ROWH}	15	—	ns		
Column address setup time	t_{COMS}	6	—	ns		
Column address hold time	t_{COMH}	10	—	ns		
MD turn-on time	t_{MDON}	0	—	ns		
$\overline{\text{WE}}$ delay time	t_{WED}	—	25	ns		
MD output setup time	t_{MDOS}	0	—	ns		
MD output hold time	t_{MDOH}	18	—	ns		

(9) UGM Refresh Cycle

Table 7-12 UGM Refresh Cycle

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
RAS delay time	t_{RASD}	—	25	ns	Figure 7-12	
CAS delay time	t_{CASD}	—	25	ns		

(10) Master Display Mode

Table 7-13 Master Display Mode

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
DCLK rise delay time from CLK1	t_{DCRD}	—	30	ns	Figure 7-13	
FCLK rise delay time from CLK1	t_{FCRD}	—	30	ns		
FCLK fall delay time from CLK1	t_{FCFD}	—	30	ns		
DD setup time for DCLK	t_{DDS}	9	—	ns		
DD hold time for DCLK	t_{DDH}	5	—	ns		
$\overline{\text{HSYNC}}$ delay time from DCLK	t_{HSDD}	—	25	ns		
$\overline{\text{VSYNC}}$ delay time from DCLK	t_{VSDD}	—	25	ns		
ODDF delay time from DCLK	t_{ODDD}	—	25	ns		
$\overline{\text{CSYNC}}$ delay time from DCLK	t_{SYDD}	—	25	ns		
DISP delay time from DCLK	t_{DIDD}	—	25	ns		
CDE delay time from DCLK	t_{CDEDD}	—	25	ns		

(11) TV Sync Display Mode

Table 7-14 TV Sync Display Mode

Item	Symbol	Min	Max	Unit	Test Conditions	Notes
DCLK rise delay time from CLK1	t_{DCRD}	—	30	ns	Figure 7-14	
DCLK fall delay time from CLK1 1	t_{DCFD1}	—	30	ns		DOT = 1
DCLK fall delay time from CLK0 0	t_{DCFD0}	—	30	ns		DOT = 0
FCLK rise delay time from CLK1	t_{FCRD}	—	30	ns		
FCLK fall delay time from CLK1	t_{FCFD}	—	30	ns		
DCLK cycle time	t_{cycD}	$2t_{cyc1}$	$2t_{cyc1}$	ns		DOT = 1
		t_{cyc1}	t_{cyc1}			DOT = 0
DD setup time for DCLK	t_{DDS}	9	—	ns		
DD hold time for DCLK	t_{DDH}	5	—	ns		
DISP delay time from DCLK	t_{DIDD}	—	25	ns		
CDE delay time from DCLK	t_{CDEDD}	—	25	ns		
$\overline{\text{EXHSYNC}}$ high-level width	t_{EXHHW}	$2t_{cyc1}$	—	ns		
$\overline{\text{EXHSYNC}}$ low-level width	t_{EXLLW}	$4t_{cyc1}$	—	µs		
$\overline{\text{EXHSYNC}}$ acceptance indeterminate time 1	t_{EXH1}	5	—	ns		
$\overline{\text{EXHSYNC}}$ acceptance indeterminate time 2	t_{EXH2}	5	—	ns		
DISP start time for $\overline{\text{EXHSYNC}}$	t_{DIEXH}	hds-1	hds-1	t_{cycD}		1
$\overline{\text{EXVSYNC}}$ low-level width	t_{EXVLW}	1HC	—	t_{cycD}		
$\overline{\text{EXVSYNC}}$ acceptance indeterminate time 1	t_{EXV1}	5	—	ns		
$\overline{\text{EXVSYNC}}$ acceptance indeterminate time 2	t_{EXV2}	5	—	ns		
ODDF acceptance indeterminate time 1	t_{OD1}	$4t_{cyc1}$	—	ns		
ODDF acceptance indeterminate time 2	t_{OD2}	$1t_{cyc1}$	—	ns		

Note: 1. hds = hsw + xs

7.5 Timing Charts

7.5.1 Input Clocks

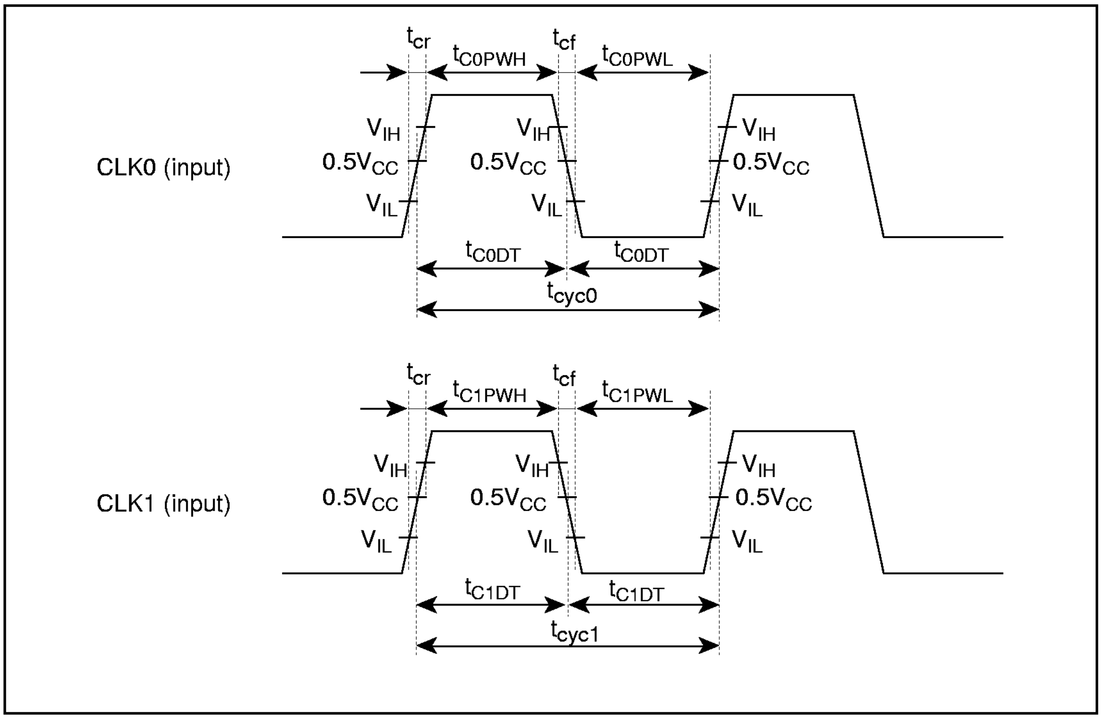


Figure 7-3 Input Clocks (Pins MODE2 to MODE0 = 011)

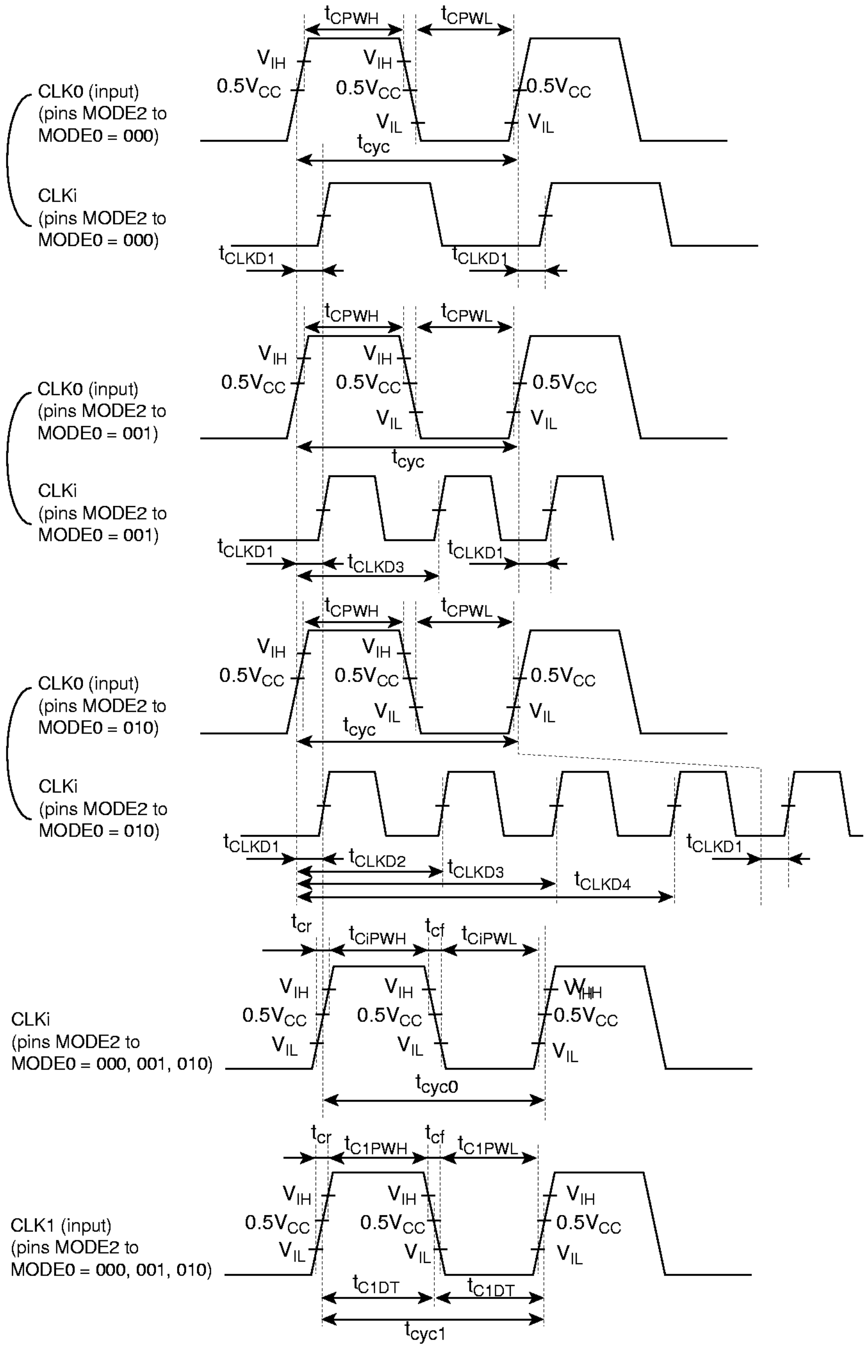
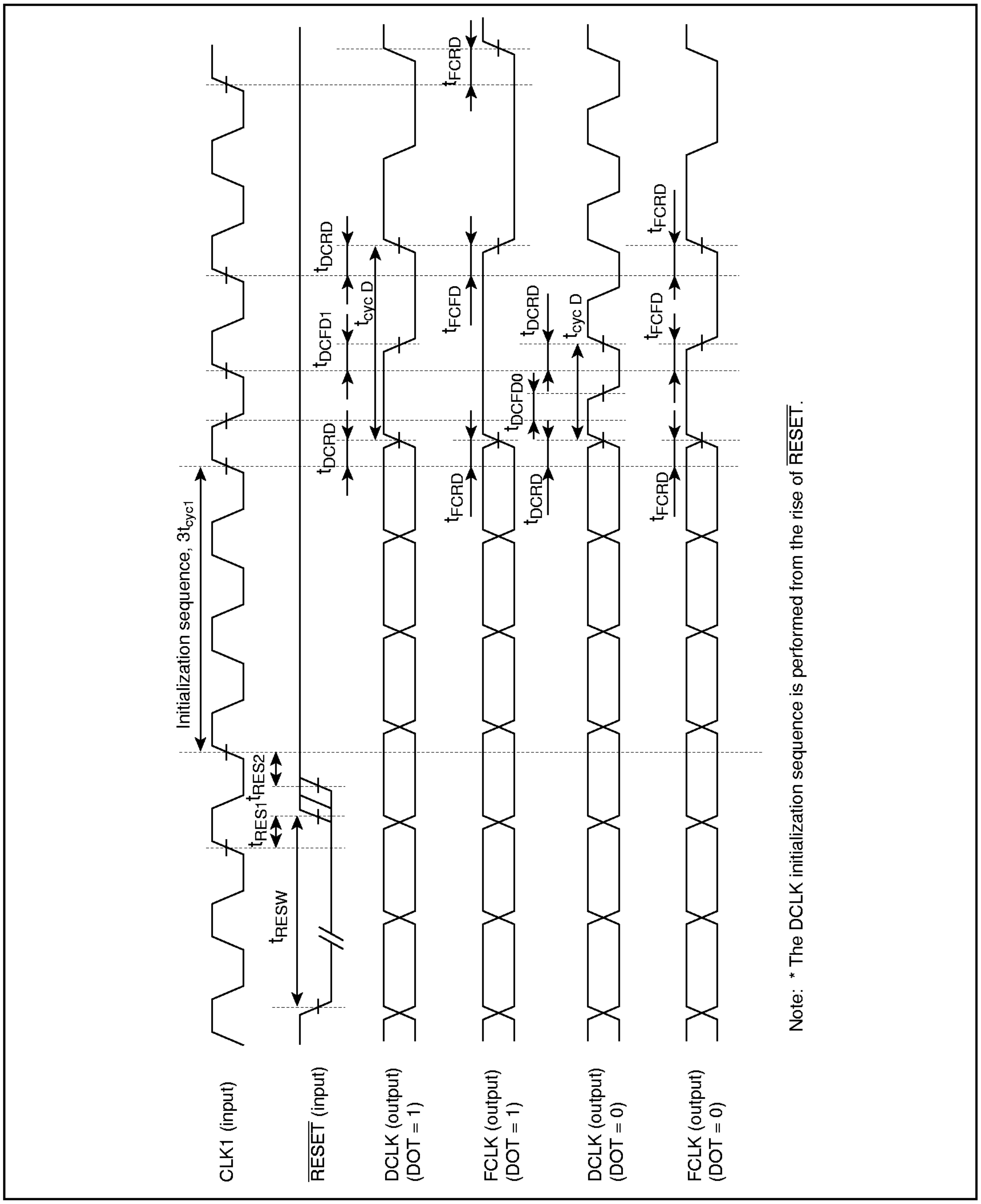


Figure 7-4 Input Clocks (Pins MODE2 to MODE0 = 000, 001, 010)

7.5.2 Reset Timing



Note: * The DCLK initialization sequence is performed from the rise of $\overline{\text{RESET}}$.

Figure 7-5 Reset Timing

7.5.3 CPU Read Cycle Timing

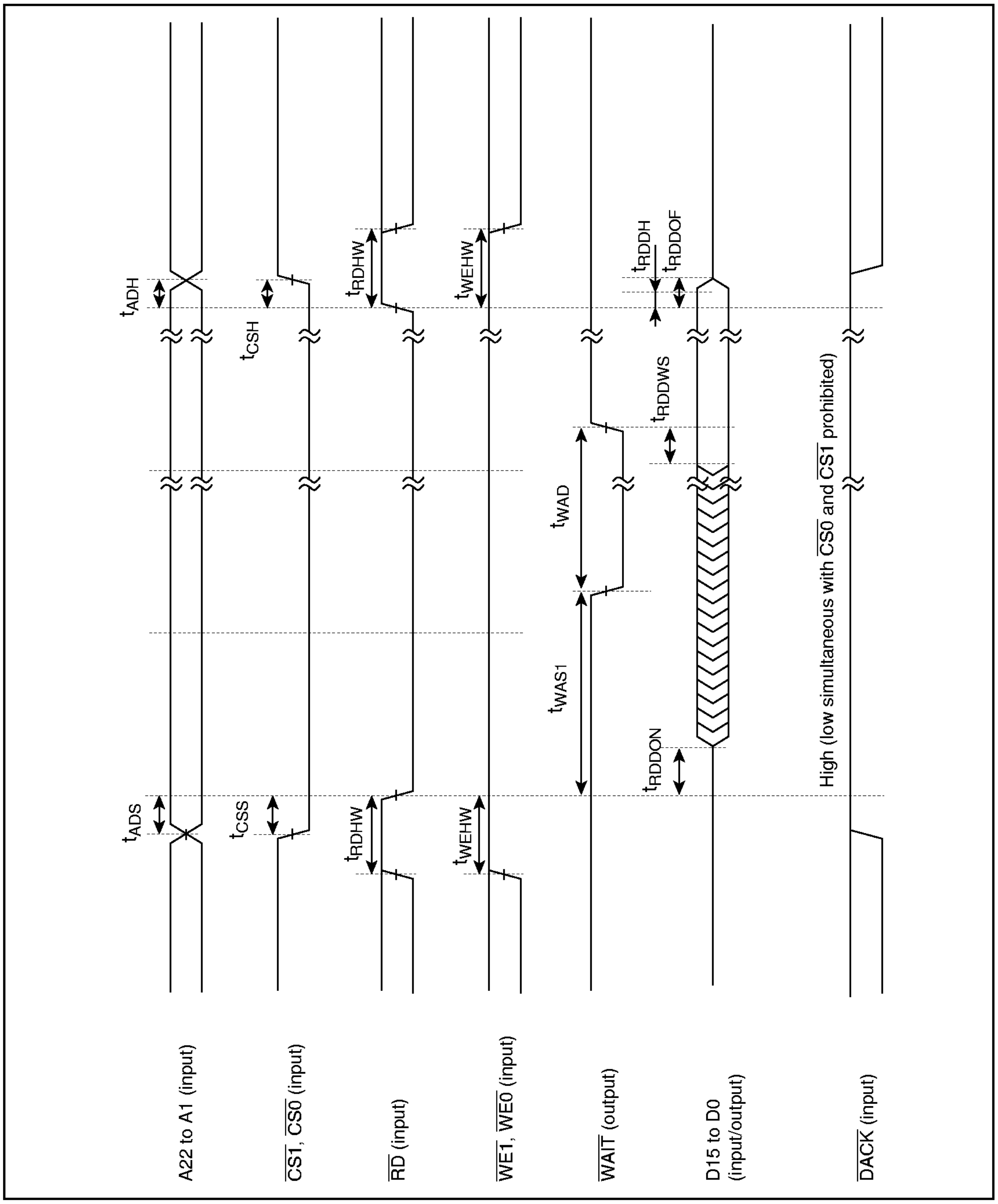

A22 to A1 (input)
$\overline{CS1}$, $\overline{CS0}$ (input)
$\overline{RD}$ (input)
$\overline{WE1}$, $\overline{WE0}$ (input)
$\overline{WAIT}$ (output)
D15 to D0 (input/output)
$\overline{DACK}$ (input)
t_{ADS}
t_{ADH}
t_{CSS}
t_{CSH}
t_{RDHW}
t_{RDHW}
t_{WEHW}
t_{WEHW}
t_{WAS1}
t_{WAD}
t_{RDDON}
t_{RDDWS}
t_{RDDH}
t_{RDDOF}
High (low simultaneous with $\overline{CS0}$ and $\overline{CS1}$ prohibited)

Figure 7-6 CPU Read Cycle Timing (CPU ← Q2i) with Hardware Wait

7.5.4 CPU Write Cycle Timing

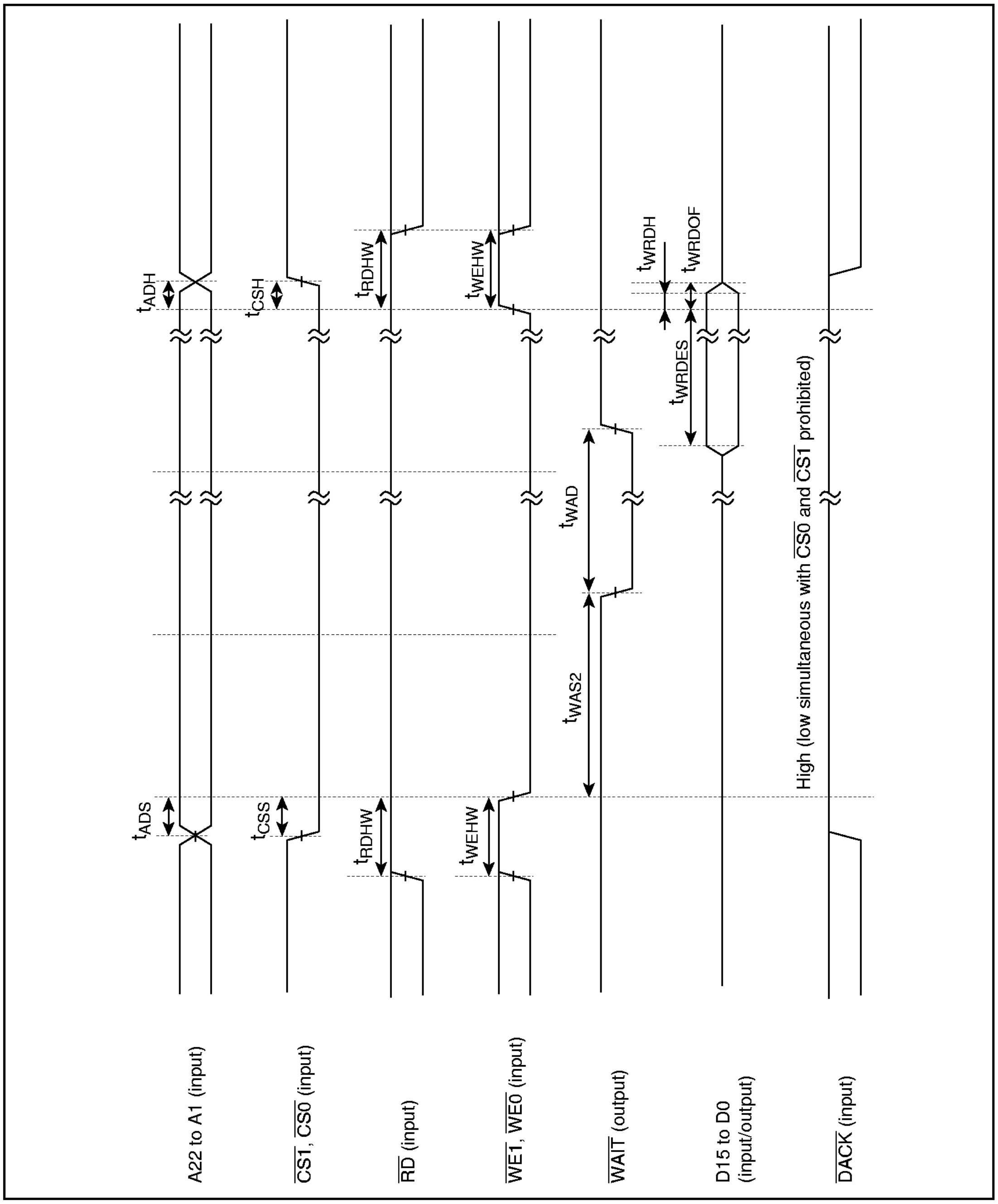

A22 to A1 (input)
$\overline{\text{CS1}}$, $\overline{\text{CS0}}$ (input)
$\overline{\text{RD}}$ (input)
$\overline{\text{WE1}}$, $\overline{\text{WE0}}$ (input)
$\overline{\text{WAIT}}$ (output)
D15 to D0 (input/output)
$\overline{\text{DACK}}$ (input)
t_{ADS}
t_{CSS}
t_{RDHW}
t_{WEHW}
t_{WAS2}
t_{WAD}
t_{WRDES}
t_{ADH}
t_{CSH}
t_{RDHW}
t_{WEHW}
t_{WRDH}
t_{WRDOF}
High (low simultaneous with $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$ prohibited)

Figure 7-7 CPU Write Cycle Timing (CPU → Q2i) with Hardware Wait

7.5.5 DMA Write Cycle Timing (DMAC → Q2i)

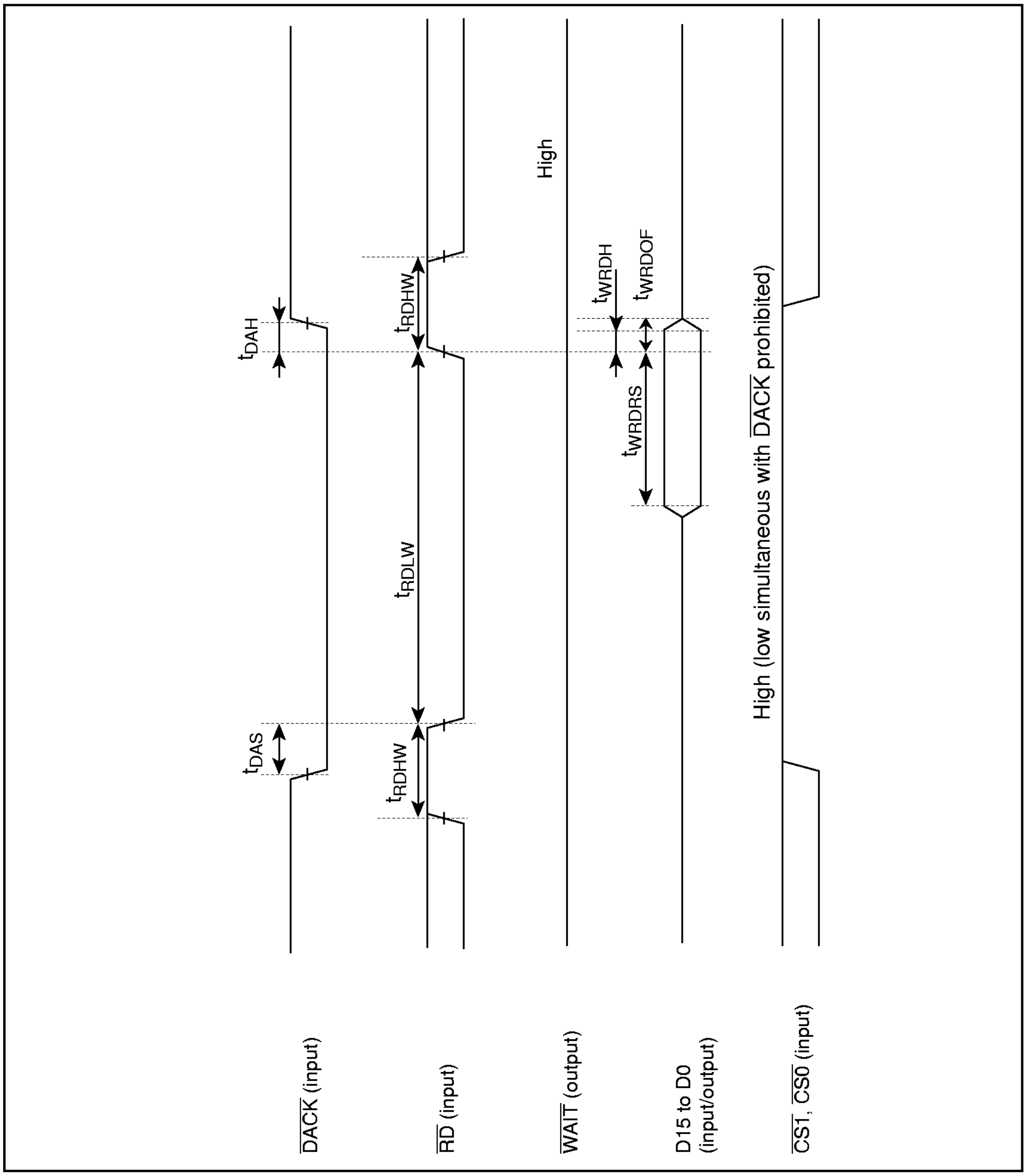


Figure 7-8 (1) DMA Write Cycle Timing (DMAC → Q2i)

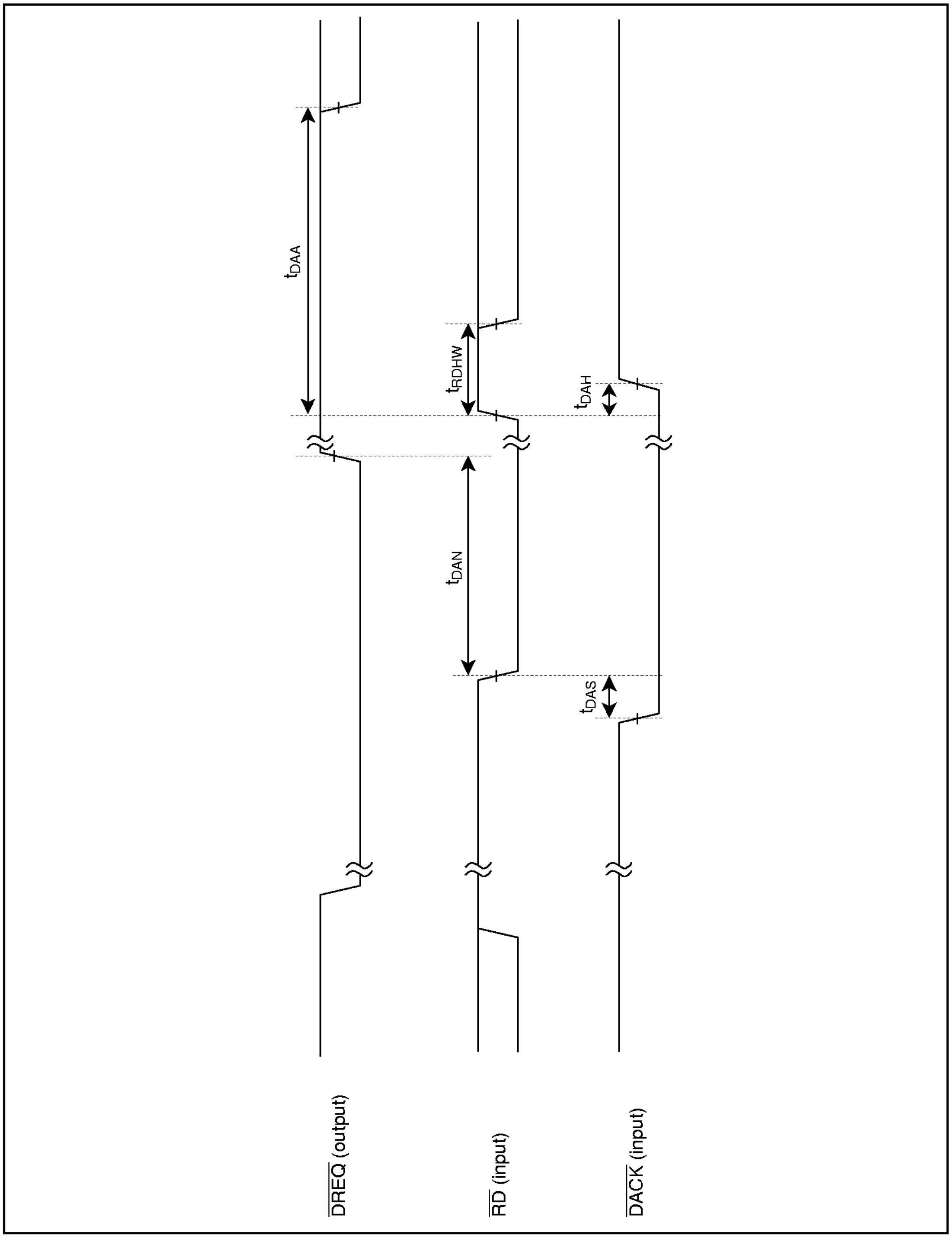

$\overline{\text{DREQ}}$ (output)
$\overline{\text{RD}}$ (input)
$\overline{\text{DACK}}$ (input)
t_{DAA}
t_{RDHW}
t_{DAH}
t_{DAN}
t_{DAS}

Figure 7-8 (2) DMA Write Cycle Timing (DMAC → Q2i)

7.5.6 Interrupt Output Timing

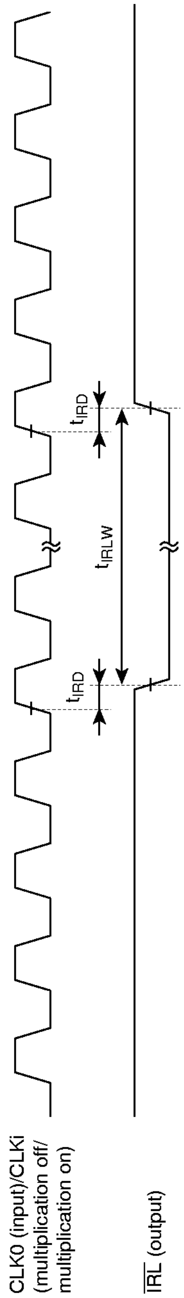


Figure 7-9 Interrupt Output Timing

7.5.7 UGM Read Cycle Timing

UGM Single Read Cycle Timing

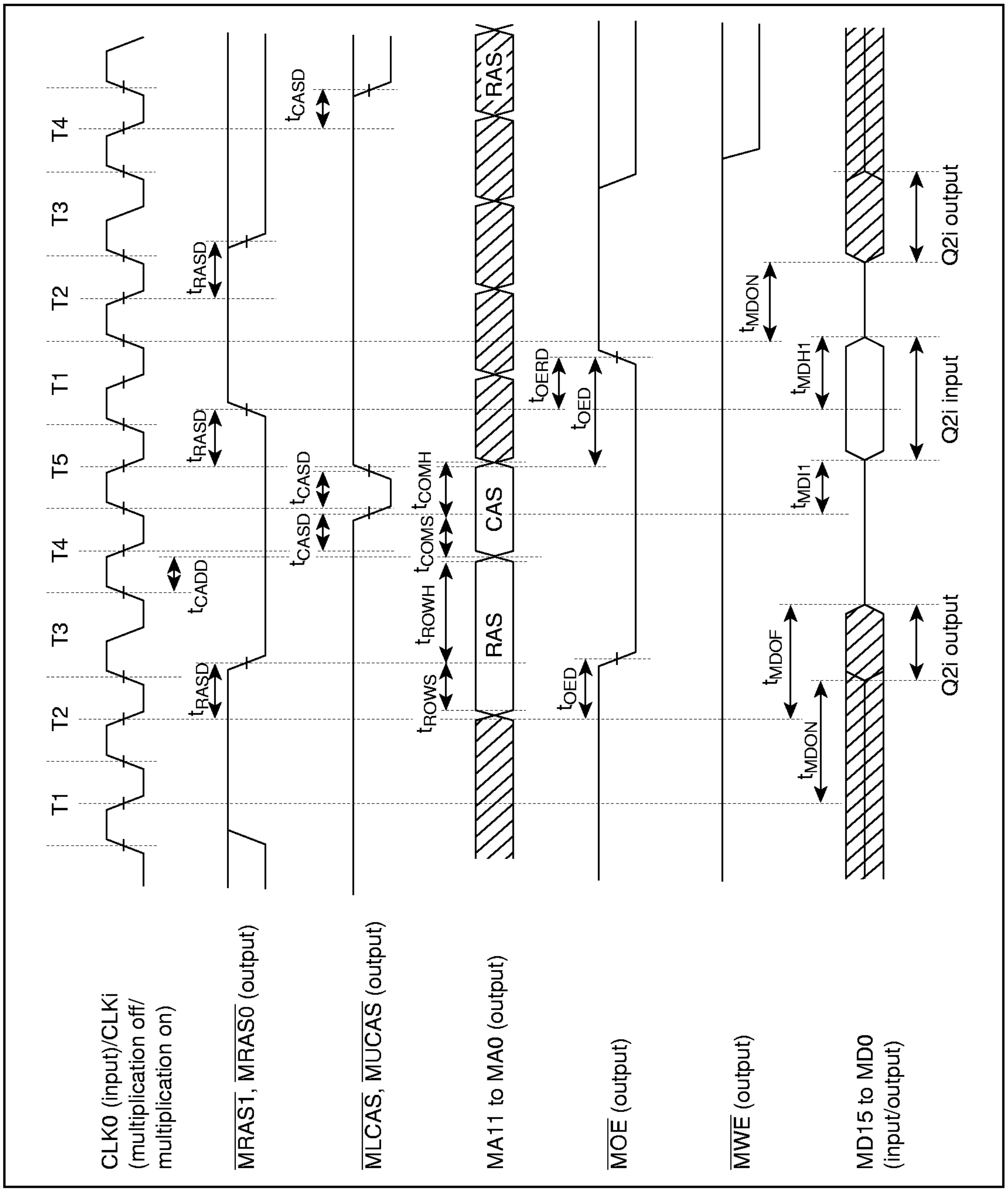


Figure 7-10 (1) UGM (EDO-DRAM) Single Read Cycle Timing

UGM Burst Read Cycle Timing

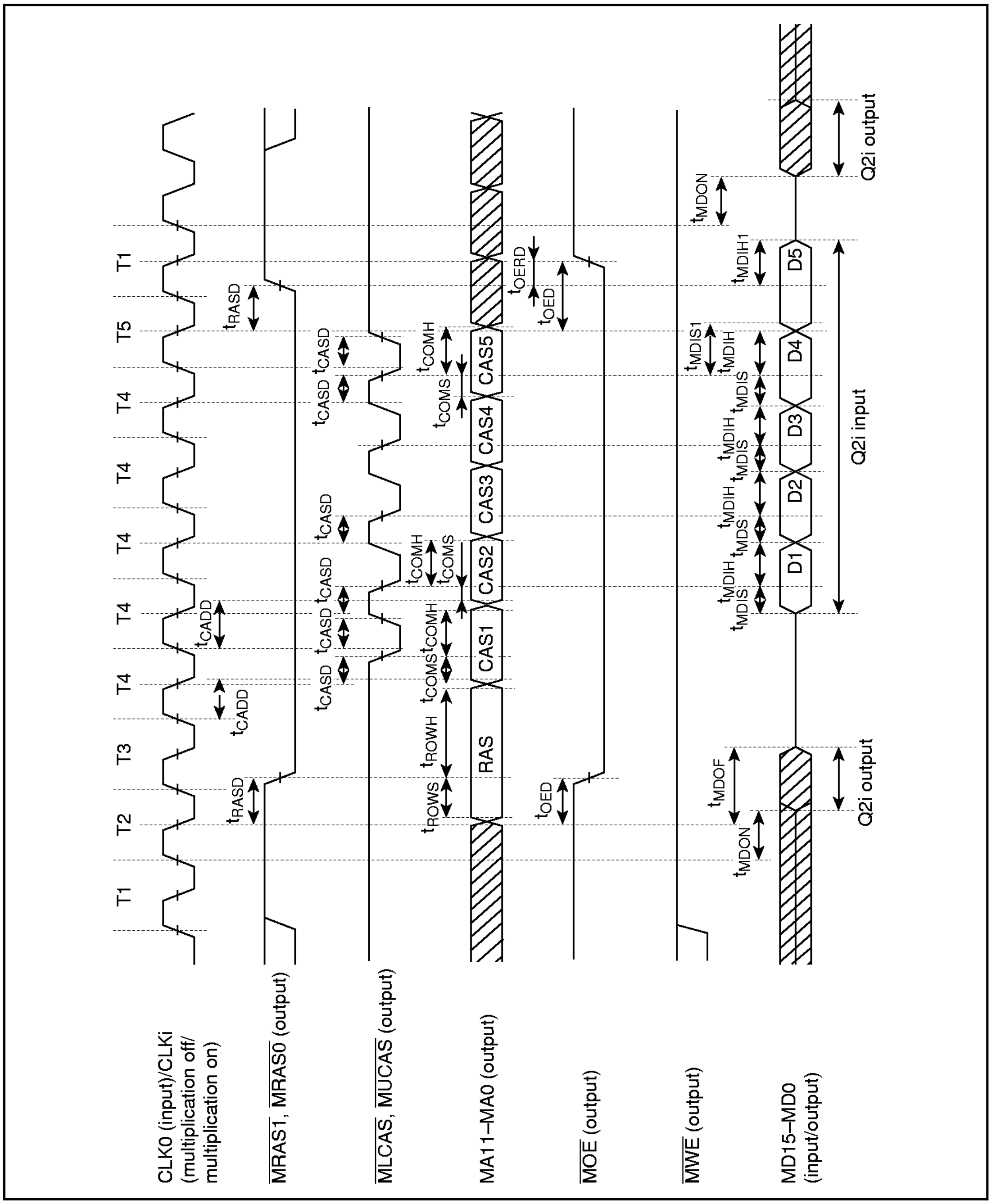


Figure 7-10 (2)   UGM (EDO-DRAM) Burst Read Cycle Timing

7.5.8 UGM Write Cycle Timing

UGM Single Write Cycle Timing

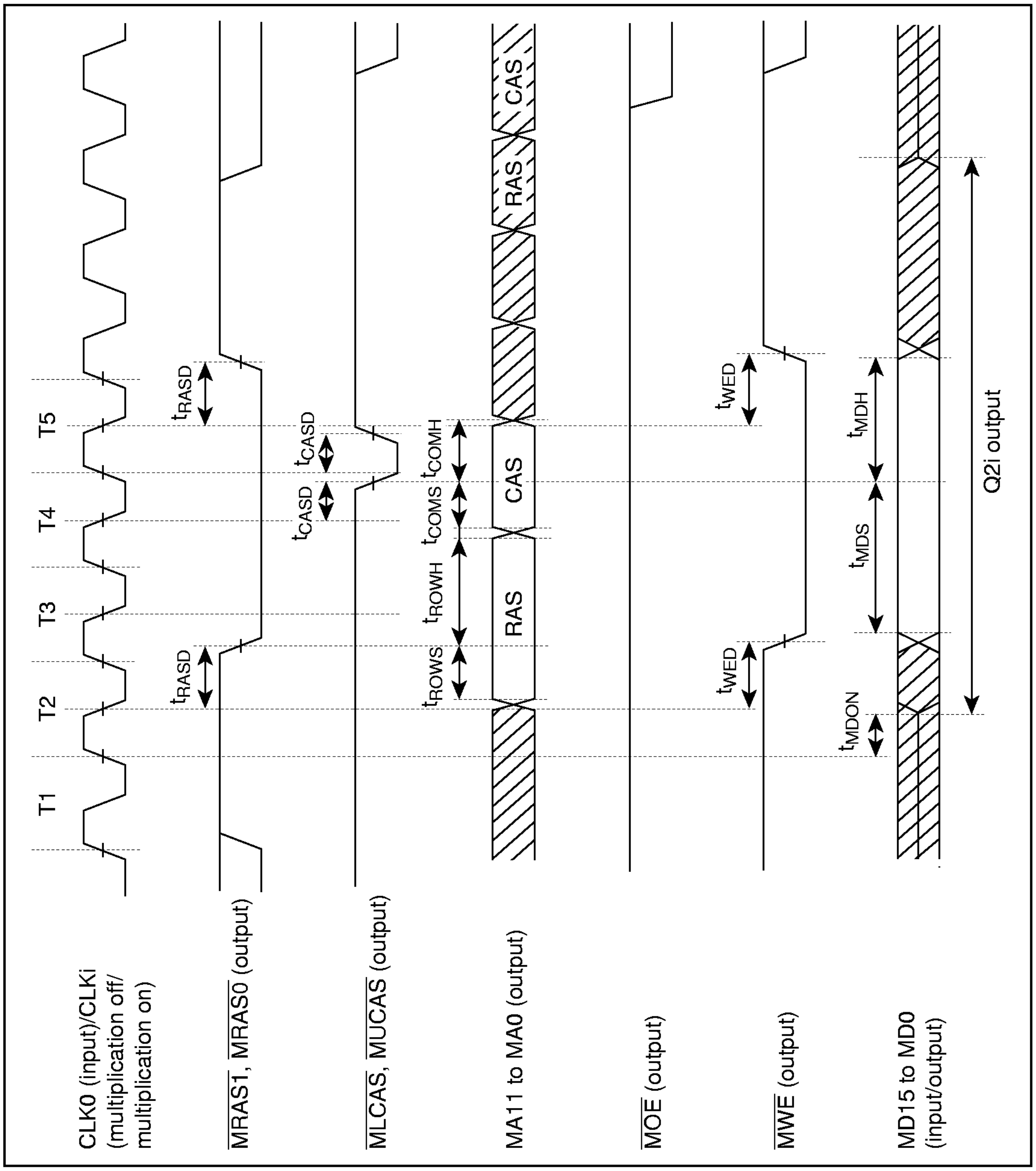


Figure 7-11 (1) UGM (EDO-DRAM) Single Write Cycle Timing

UGM Burst Write Cycle Timing

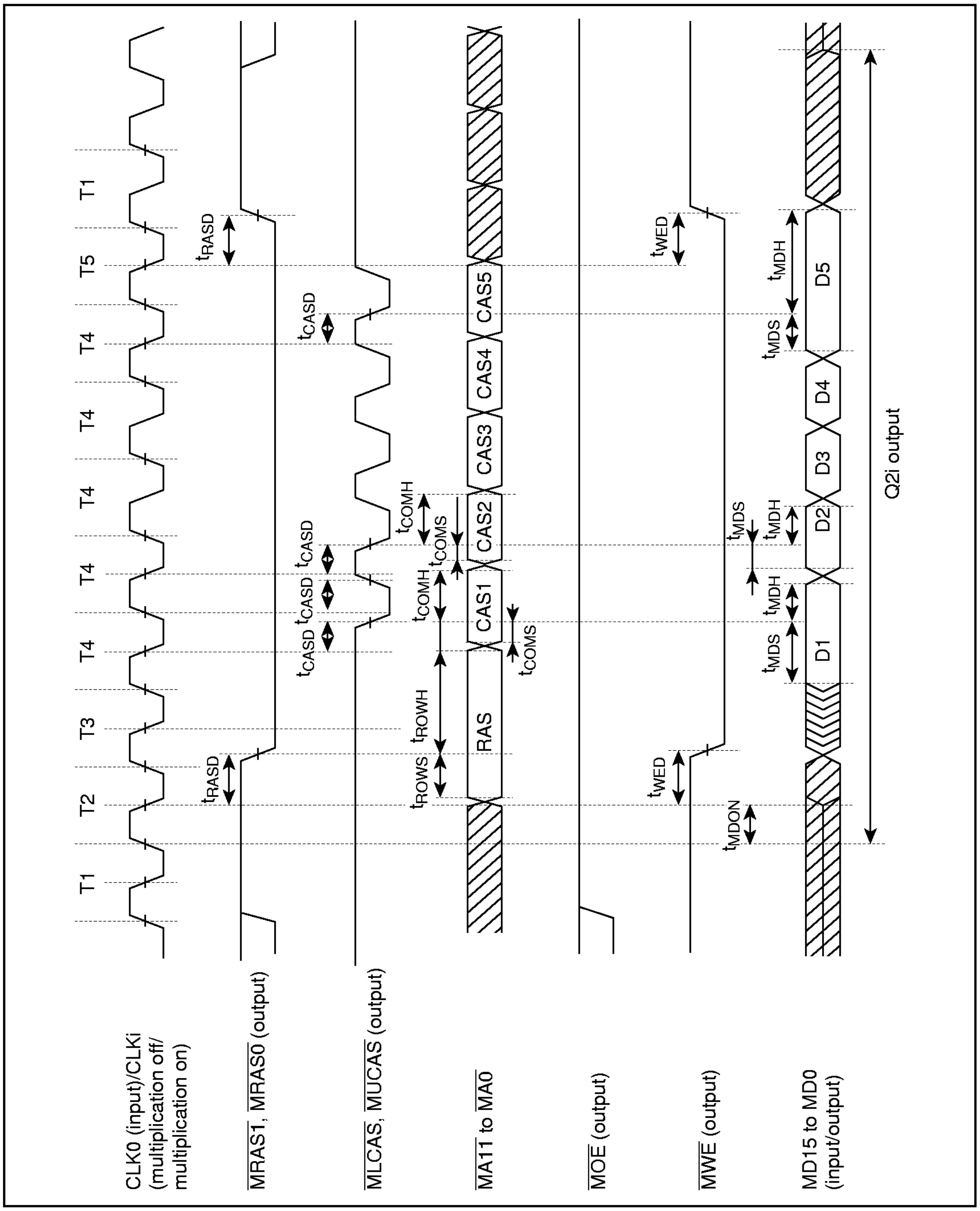


Figure 7-11 (2) UGM (EDO-DRAM) Burst Write Cycle Timing

7.5.9 UGM Refresh Cycle Timing

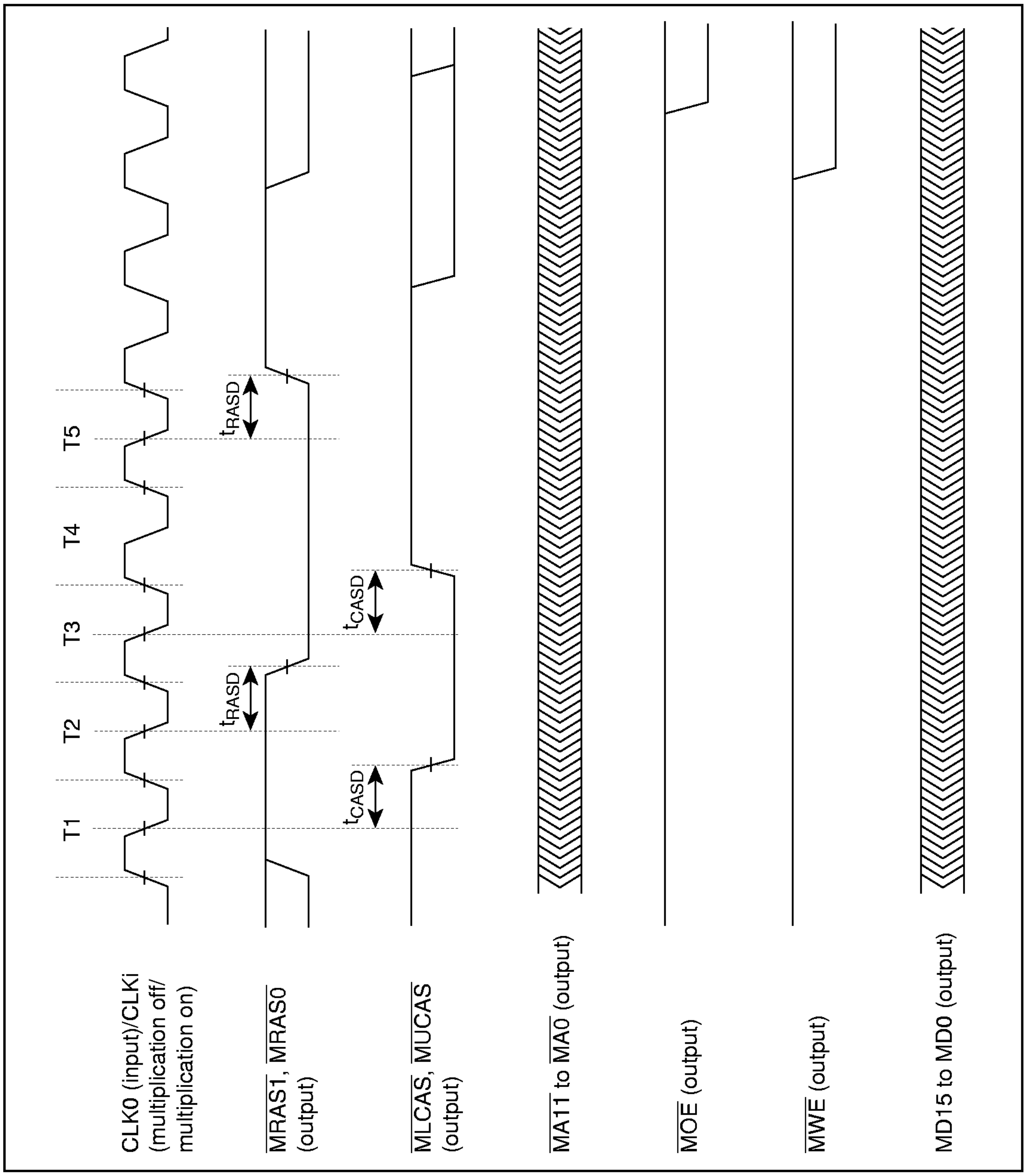


Figure 7-12 UGM (EDO-DRAM) Refresh Cycle Timing

7.5.10 Master Mode Display Timing

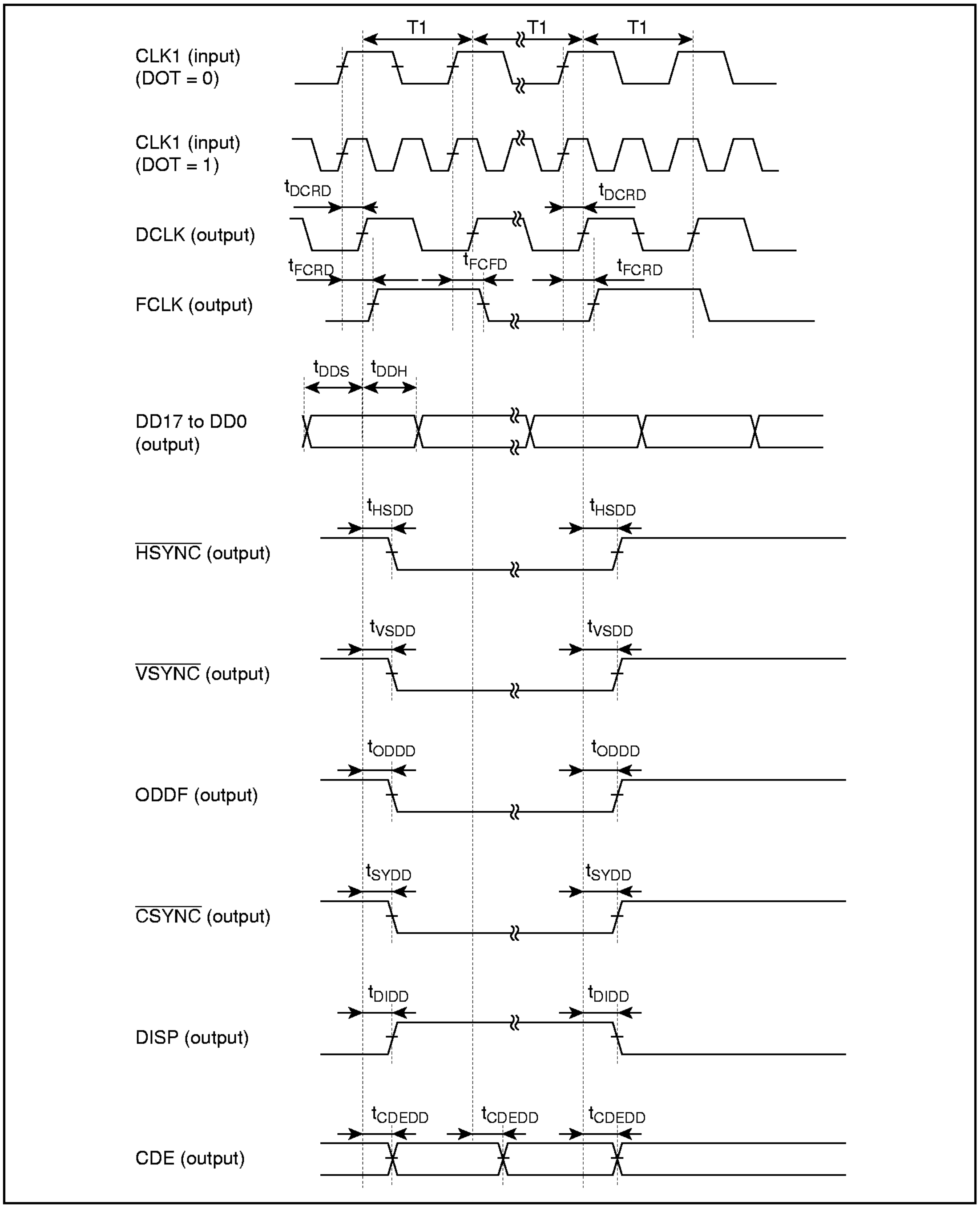


Figure 7-13 Master Mode Display Timing

7.5.11 TV Sync Mode Display Timing

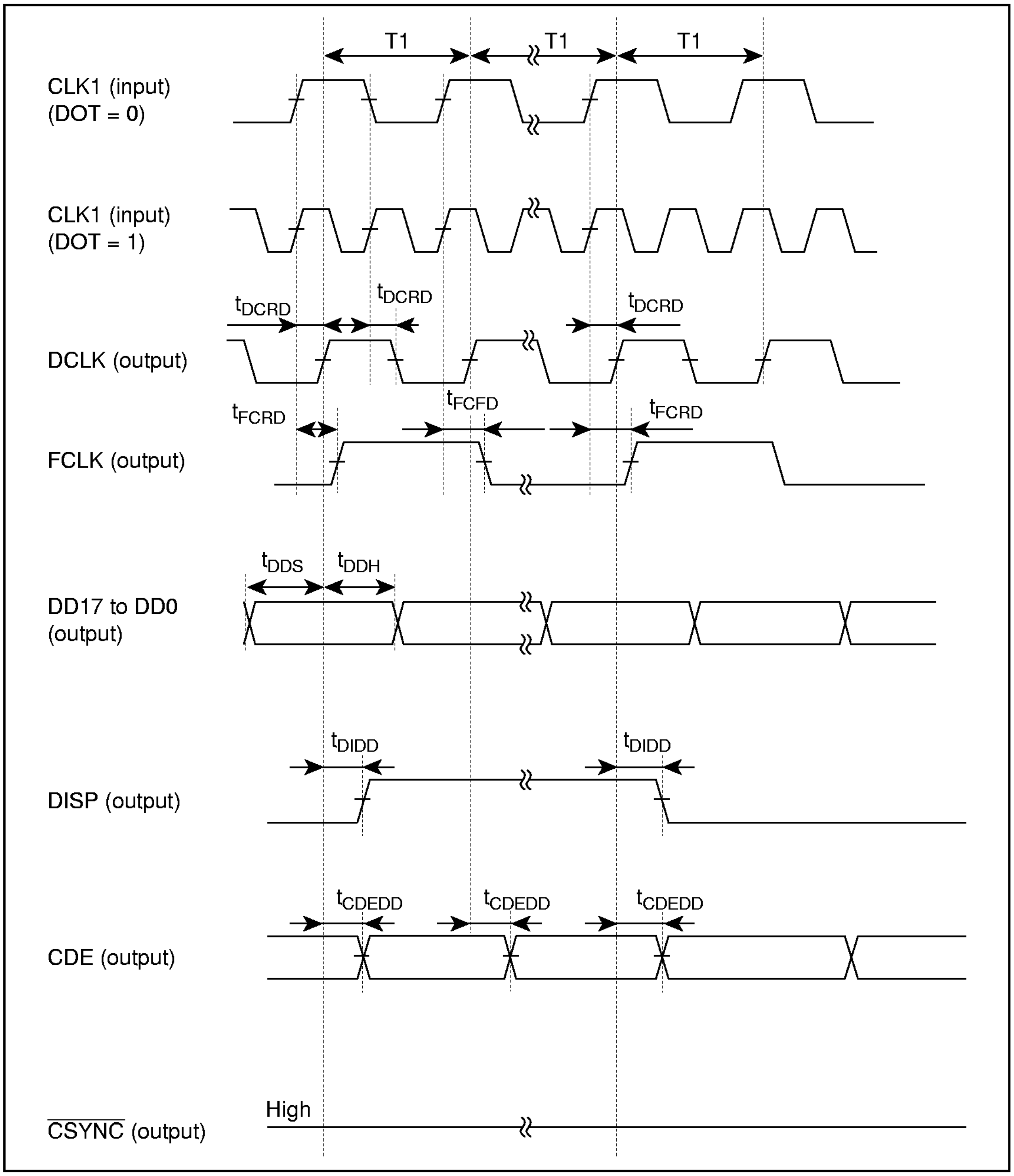
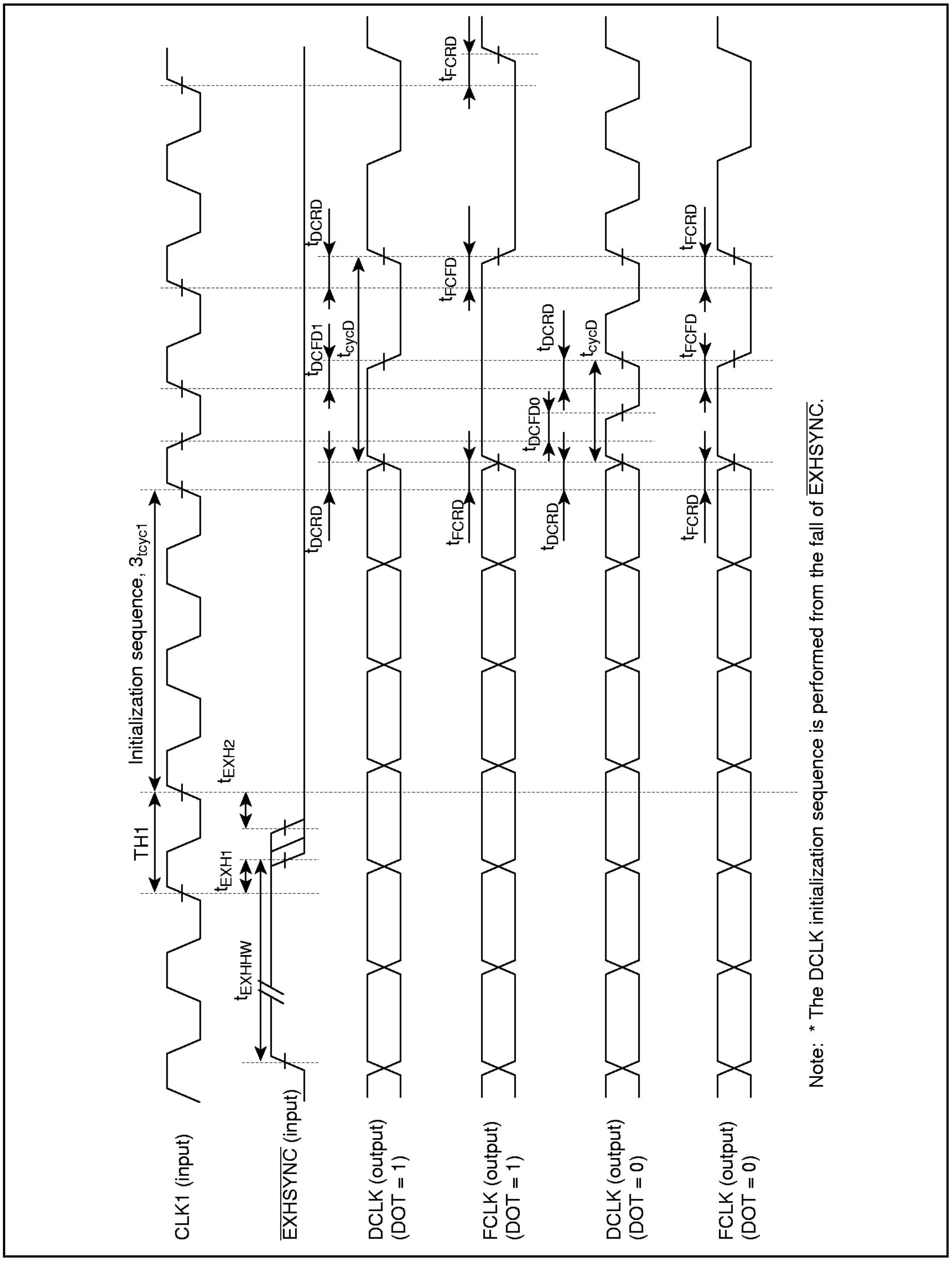


Figure 7-14 (1) TV Sync Mode Display Timing



Note: * The DCLK initialization sequence is performed from the fall of EXHSYNC.

Figure 7-14 (2) TV Sync Mode Display Timing

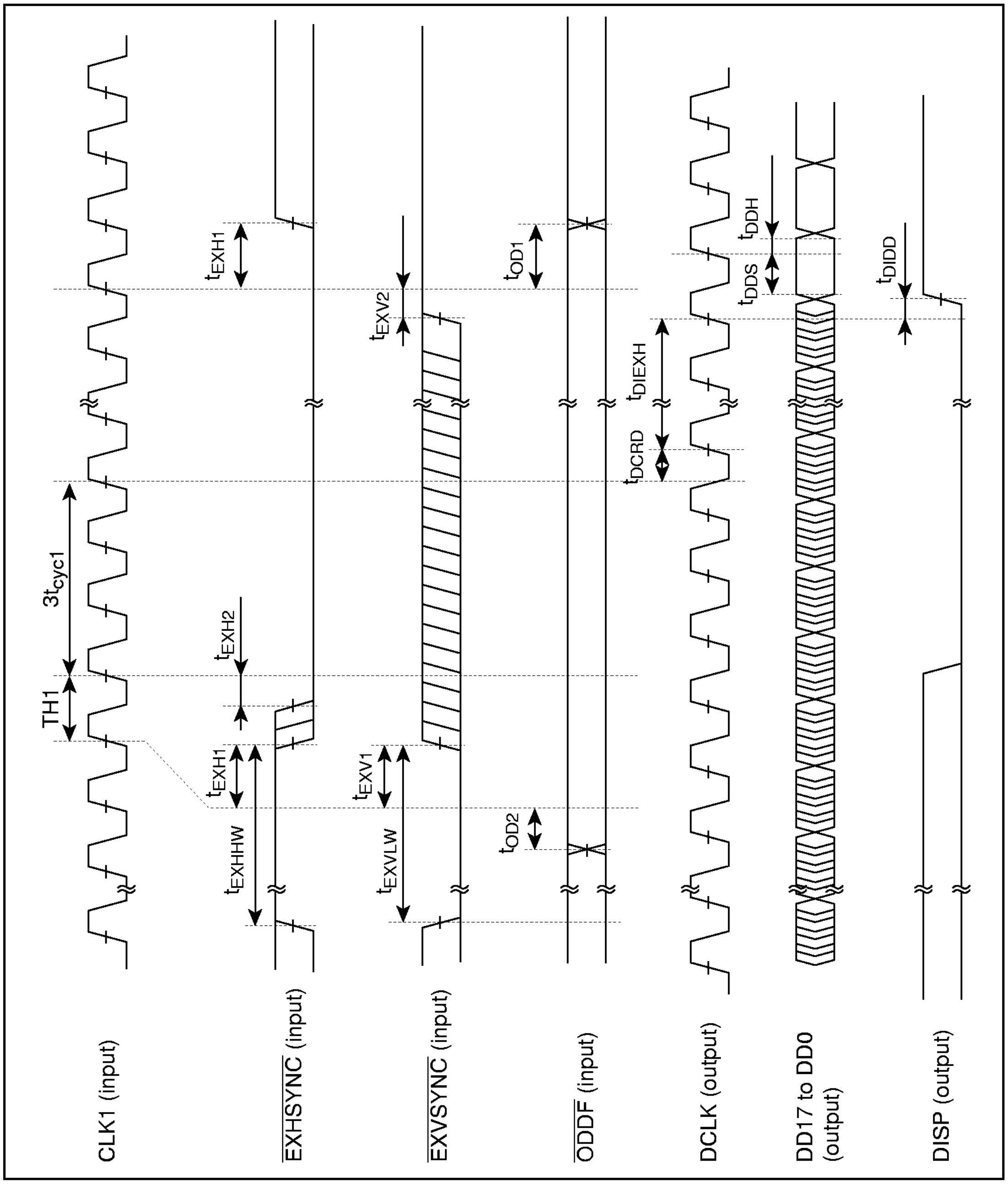


Figure 7-14 (3) TV Sync Mode Display Timing
(When DOT = 1 and $\overline{\text{EXHSYNC}}$ cycle is odd multiple of CLK1 cycle)

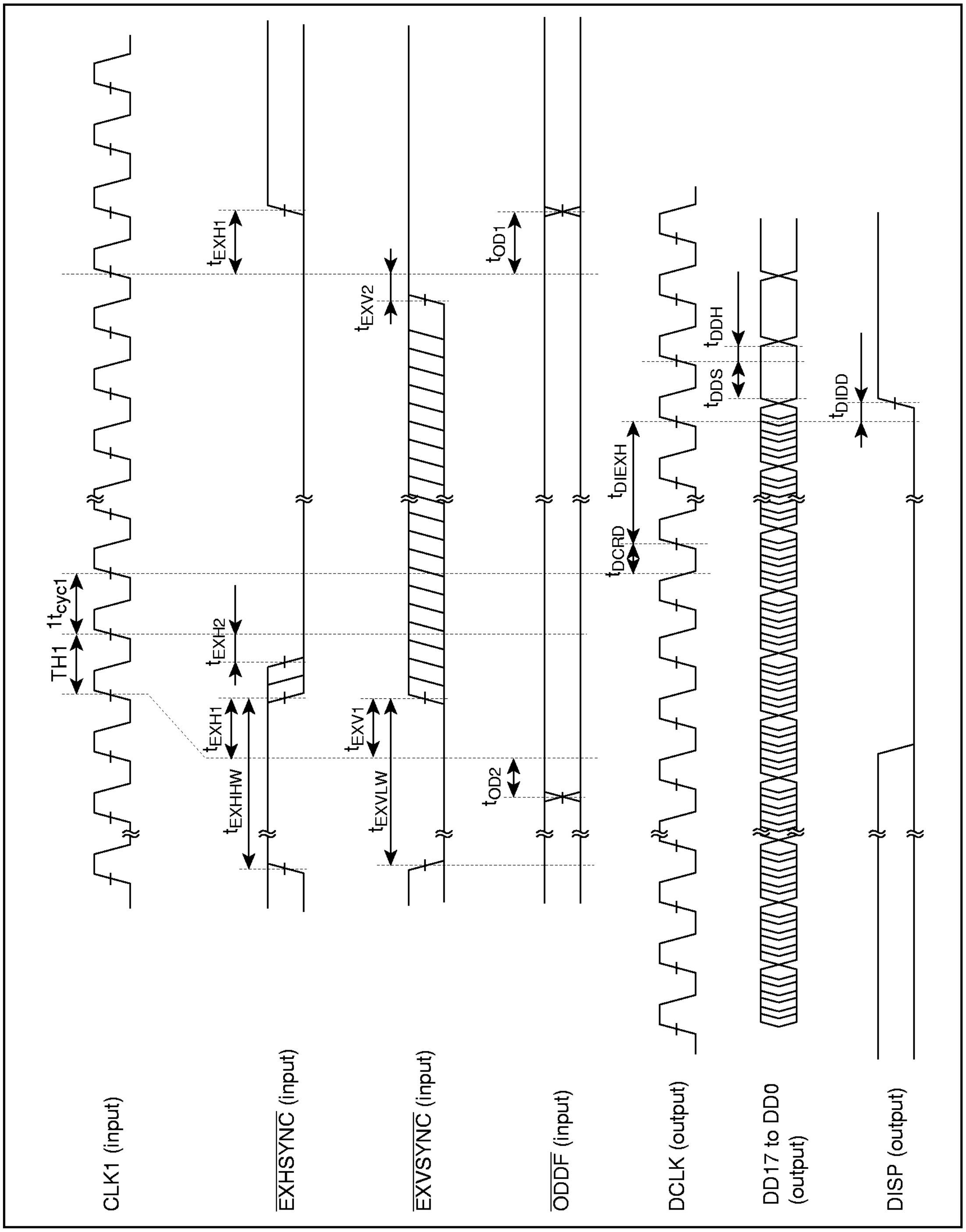


Figure 7-14 (4) TV Sync Mode Display Timing
(When DOT = 0, or DOT = 1 and $\overline{\text{EXHSYNC}}$ cycle is even multiple of CLK1 cycle)

Appendix A Registers

Table A-1 Registers

1. Interface Control Registers

Register address		R/W	Register name		Abbrevia-tion	Data															
$\overline{\text{CS1}}$	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	—	—	—		—	—															
0	000	R/W	System control		SYSR	SRES	DRES	DEN	▨	▨	RBRK	DC	RS	DBM		DMA		▨	▨	▨	▨
	001	R	Status		SR	TVR	FRM	DMF	CER	VBK	TRA	CSF	DBF	BRK	▨	▨	▨	Q			
	002	W	Status register clear		SRCR	TVCL	FRCL	DMCL	CECL	VBCL	TRCL	CSCL	▨	BRCI	▨	▨	▨	▨	▨	▨	▨
	003	R/W	Interrupt enable		IER	TVE	FRE	DME	CEE	VBE	TRE	CSE	▨	BRE	▨	▨	▨	▨	▨	▨	▨
	004	R/W	Memory mode		MEMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	MES			MEA		▨	▨
	005	R/W	Display mode		DSMR	▨	▨	▨	▨	WRAP	BG	▨	DOT	TVM		SCM		REF			
	006	R/W	Rendering mode		REMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	MWX	▨	▨	▨	▨	▨	GBM
	007	R/W	Input data conversion mode		IEMR	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	YUV	

2. Memory Control Registers

0	008	R/W	Display size	Y	DSR	▨	▨	▨	▨	▨	▨	DSX									
	009	R/W		X		▨	▨	▨	▨	▨	▨	▨	DSY								
	00A	R/W	Display start address	0	DSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	DSA0						
	00B	R/W		1		▨	▨	▨	▨	▨	▨	▨	▨	▨	DSA1						
	00C	R/W	Display list start address	H	DLSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	DLSAH						
	00D	R/W		L		DLSAL											▨	▨	▨	▨	▨
	00E	R/W	Multi-valued source area start address		SSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	SSAH						▨
	00F	R/W	Work area start address		WSAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	WSAH						
	010	R/W	DMA transfer start address	H	DMASR	▨	▨	▨	▨	▨	▨	▨	▨	▨	DMASH						
	011	R/W		L		DMASL															▨
	012	R/W	DMA transfer word count		DMAWR	DMAW															

3. Display Control Registers

0	013	R/W	Display windows	Horizontal display start address	DSWR	▨	▨	▨	▨	▨	▨	▨	HDS								
	014	R/W		Horizontal display end address		▨	▨	▨	▨	▨	▨	HDE									
	015	R/W		Vertical display start address		▨	▨	▨	▨	▨	▨	▨	VDS								
	016	R/W		Vertical display end address		▨	▨	▨	▨	▨	▨	VDE									
	017	R/W	Horizontal sync pulse width		HSWR	▨	▨	▨	▨	▨	▨	▨	▨	▨	HSW						
	018	R/W	Horizontal scan cycle		HCR	▨	▨	▨	▨	▨	HC										
	019	R/W	Vertical sync position		VSPR	▨	▨	▨	▨	▨	▨	VSP									
	01A	R/W	Vertical scan cycle		VCR	▨	▨	▨	▨	▨	▨	VC									
	01B	R/W	Display off output	H	DOOR	▨	▨	▨	▨	▨	▨	▨	▨	DOR						▨	▨
	01C	R/W		L		DOG						▨	▨	DOB						▨	▨
	01D	R/W	Color detection	H	CDER	▨	▨	▨	▨	▨	▨	▨	▨	CDR						▨	▨
	01E	R/W		L		CDG						▨	▨	CDB						▨	▨

4. Rendering Control Registers

0	01F	R	Command status	H	CSTR	▨	▨	▨	▨	▨	▨	▨	▨	▨	CSTH						
	020	R		L		CSTL															▨

Table A-1 Registers (cont)

5. Input Data Control Registers

Register Address		R/W	Register Name		Abbreviation	Data															
CS1	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	021	R/W	Image data transfer start address	H	ISAR	▨	▨	▨	▨	▨	▨	▨	▨	▨	ISAH						
	022	R/W		L		ISAL															▨
	023	R/W	Image data size	X	IDSR	▨	▨	▨	▨	▨	IDSX										
	024	R/W		Y		▨	▨	▨	▨	▨	▨	IDSY									
	025	W	Image data entry		IDER	IDE															

6. Memory Control Registers (2)

Register Address		R/W	Register Name		Abbreviation	Data															
CS1	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	026	R/W	Background start coordinate	X	BGSR	▨	▨	▨	▨	▨	▨	BGSX									
	027	R/W		Y		▨	▨	BGSY													
	028 to 0FF	—				Reserved															

7. Rendering Control Registers (2)

Register Address		R/W	Register Name		Abbreviation	Data															
CS1	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	040	R	Current pointer	X	CURR	▨	▨	▨	XC												
	041	R		Y		▨	▨	▨	YC												
	042	R	Local offset	X	LCOR	▨	▨	▨	XO												
	043	R		Y		▨	▨	▨	YO												
	044	R	User clipping area	XMIN	UCLR	▨	▨	▨	YXMIN												
	045	R		YMIN		▨	▨	▨	UYMIN												
	046	R		XMAX		▨	▨	▨	UXMAX												
	047	R		YMAX		▨	▨	▨	UYMAX												
	048	R	System clipping area	XMAX	SCLR	▨	▨	▨	SXMAX												
	049	R		YMAX		▨	▨	▨	SYMAX												
	04A	R/W	Return address	H	RTNR	▨	▨	▨	▨	▨	▨	▨	▨	▨	RTNH						
	04B	R/W		L		RTNL															▨

8. Reserved

Register Address		R/W	Register Name		Abbreviation	Data															
0	04C to 0FF	—				Reserved															

9. Color Palette

Register Address		R/W	Register Name		Abbreviation	Data															
CS1	A [10:1]					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	100	R/W	Color palette registers	000H	CP000R	▨	▨	▨	▨	▨	▨	▨	▨	R000						▨	▨
	101	R/W		000L		G000						▨	▨	B000						▨	▨
	102	R/W		001H	CP001R	▨	▨	▨	▨	▨	▨	▨	▨	R001						▨	▨
	103	R/W		001L		G001						▨	▨	B001						▨	▨
	104	R/W		002H	CP002R	▨	▨	▨	▨	▨	▨	▨	▨	R002						▨	▨
	105	R/W		002L		G002						▨	▨	B002						▨	▨
	⋮	⋮			⋮	⋮															
	2FE	R/W		255H	CP255R	▨	▨	▨	▨	▨	▨	▨	▨	R255						▨	▨
	2FF	R/W		255L		G255						▨	▨	B255						▨	▨

Appendix B Drawing Commands and Parameters

B.1 Relationship Between Drawing Commands and Rendering Attributes

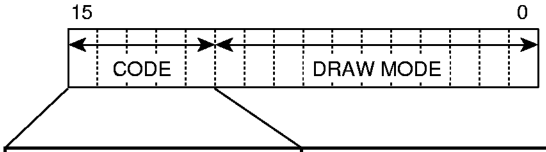
Table B-1 Relationship Between Drawing Commands and Rendering Attributes

Command	Reference Data				Drawing Destination		Rendering Attributes									
	Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work	TRNS	STYL	CLIP	NET	EOS	HALF	WORK	Bold Line Drawing	LNi	FST
POLYGON4A	O	×	Δ	×	O	×	O	O	O	O	O	×	O	×	O	O
POLYGON4B	×	O	Δ	×	O	×	O	◊	O	O	O	◊	O	×	×	×
POLYGON4C	×	×	Δ	O	O	×	×	×	O	O	O	×	O	×	×	×
LINE	×	×	×	O	O	×	×	×	O	O	O	×	×	O	×	×
RLINE	×	×	×	O	O	×	×	×	O	O	O	×	×	O	×	×
PLINE	×	O	×	×	O	×	O	O	O	O	O	×	×	×	•	×
RPLINE	×	O	×	×	O	×	O	O	O	O	O	×	×	×	•	×
FTRAP	×	×	×	×	×	O	×	×	O	×	×	×	×	×	×	×
RFTRAP	×	×	×	×	×	O	×	×	O	×	×	×	×	×	×	×
CLRW	×	×	×	×	×	O	×	×	O	×	×	×	×	×	×	×
LINEW	×	×	×	∇	×	O	×	×	O	×	O	×	×	×	×	×
RLINEW	×	×	×	∇	×	O	×	×	O	×	O	×	×	×	×	×
MOVE	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
RMOVE	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
LCOFS	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
RLCOFS	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
UCLIP	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
SCLIP	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
JUMP	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
GOSUB	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
RET	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
NOP3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
TRAP	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

O : Can be used
∇ : Can be used (EOS reference: specified color is binary EOS bit value)
Δ : Referenced depending on mode (valid when WORK = 1)
× : Cannot be used (clear to 0)
• : Fixed at 1
◊ : Cannot be specified simultaneously

B.2 Drawing Command Codes

Table B.2 Drawing Command Codes

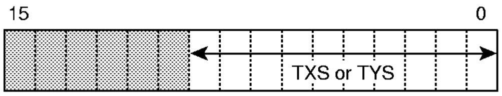


CODE					COMMAND
0	0	0	0	0	POLYGON4A
0	0	0	0	1	POLYGON4B
0	0	0	1	0	POLYGON4C
0	1	0	0	0	FTRAP
0	1	0	0	1	RFTRAP
0	1	0	1	0	LINEW
0	1	0	1	1	RLINEW
0	1	1	0	0	LINE
0	1	1	0	1	RLINE
0	1	1	1	0	PLINE
0	1	1	1	1	RPLINE
1	0	0	0	0	MOVE
1	0	0	0	1	RMOVE
1	0	0	1	0	LCOFS
1	0	0	1	1	RLCOFS
1	0	1	0	0	CLRW
1	0	1	0	1	UCLIP
1	0	1	1	1	SCLIP
1	1	0	0	0	JUMP
1	1	0	0	1	GOSUB
1	1	0	1	1	RET
1	1	1	1	1	TRAP
1	1	1	1	0	NOP3

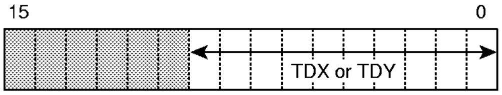
B.3 Drawing Command Parameter Specifications

POLYGON4 Commands

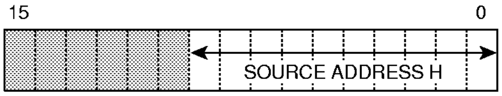
▒ : Fixed at 0



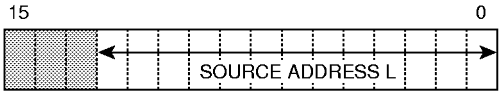
Source starting points TXS, TYS
Given as unsigned max. 10-bit data.
Specify correctly according to source area size.



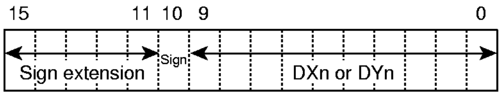
Source size TDX, TDY
Given as unsigned max. 10-bit data.
TDX can only be set in 8-pixel units.



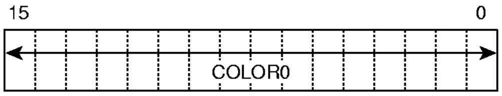
1-bit/pixel source start upper address
Given as upper 10 bits.



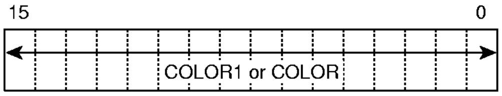
1-bit/pixel source start lower address
Given as lower 13 bits.
Source address is set as a byte address.



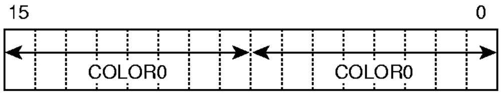
Rendering, work coordinates
Vertex coordinates DXn, DYn (1 ≤ n ≤ 4)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.



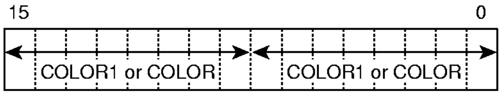
16-bit/pixel color specification
Color data 0 given as 16-bit data.



16-bit/pixel color specification
Color data 1 given as 16-bit data.

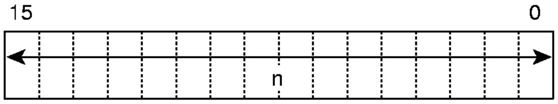


8-bit/pixel color specification
Color data 0 given as repeated 8-bit data.

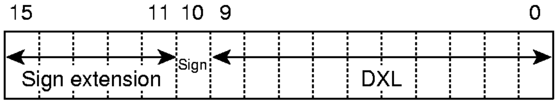


8-bit/pixel color specification
Color data 1 given as repeated 8-bit data.

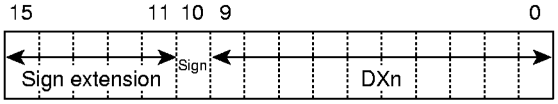
FTRAP, RFTRAP



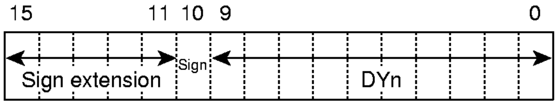
Number of vertices (2 ≤ n ≤ 65,535), absolute,
(1 ≤ n ≤ 65,535), relative
Given as unsigned 16-bit data.



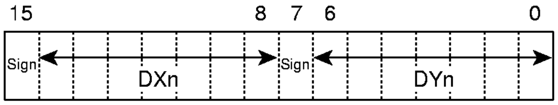
Left-hand side coordinate DXL
Given as signed 11-bit data.
Use sign extension in upper vacant bits.



Absolute coordinate
Vertex coordinate DXn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

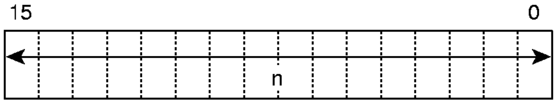


Absolute coordinate
Vertex coordinate DYn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

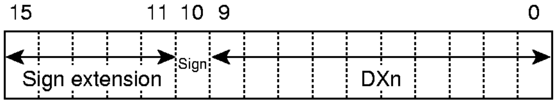


Relative coordinates
Vertex coordinates DXn, DYn (1 ≤ n ≤ 65,535)
Given as signed 8-bit data.

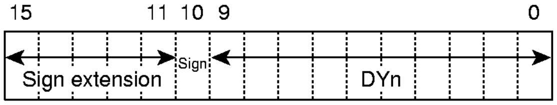
LINEW, RLINEW



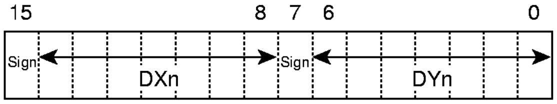
Number of vertices (2 ≤ n ≤ 65,535), absolute,
(1 ≤ n ≤ 65,535), relative
Given as unsigned 16-bit data.



Absolute coordinate
Vertex coordinate DXn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

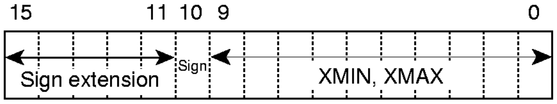


Absolute coordinate
Vertex coordinate DYn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

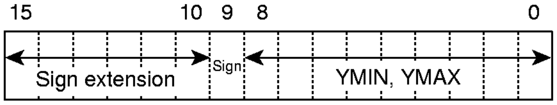


Relative coordinates
Vertex coordinates DXn, DYn (1 ≤ n ≤ 65,535)
Given as signed 8-bit data.

CLRW

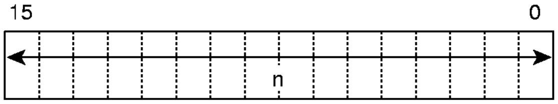


Left and right X coordinates XMIN, XMAX
Given as signed 11-bit data.

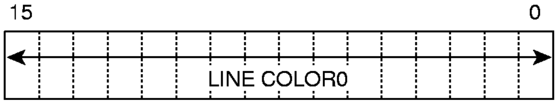


Upper and lower Y coordinates YMIN, YMAX
Given as signed 10-bit data.

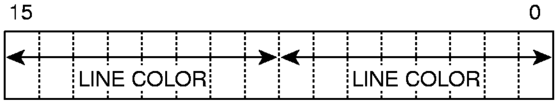
LINE, RLINE



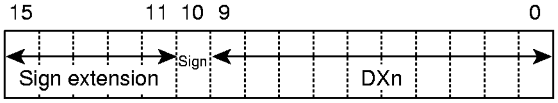
Number of vertices (2 ≤ n ≤ 65,535), absolute,
(1 ≤ n ≤ 65,535), relative
Given as unsigned 16-bit data.



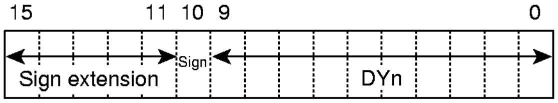
16-bit/pixel color specification
Color data given as 16-bit data.



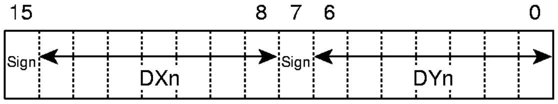
8-bit/pixel color specification
Color data given as repeated 8-bit data.



Absolute coordinate
Vertex coordinate DXn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.



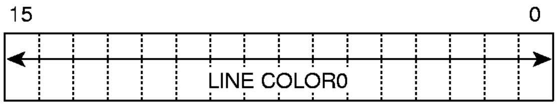
Absolute coordinate
Vertex coordinate DYn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.



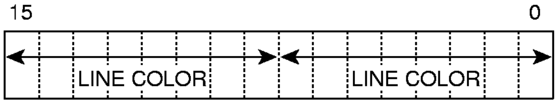
Relative coordinates
Vertex coordinates DXn, DYn (1 ≤ n ≤ 65,535)
Given as signed 8-bit data.

PLINE, RPLINE

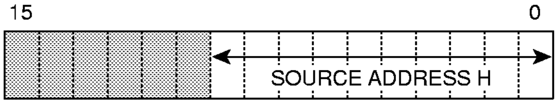
▨ : Fixed at 0



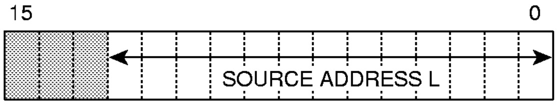
16-bit/pixel color specification
Color data given as 16-bit data.



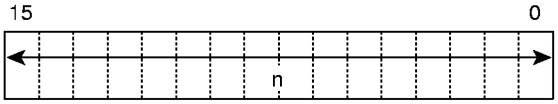
8-bit/pixel color specification
Color data given as repeated 8-bit data.



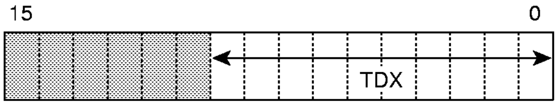
1-bit/pixel source start upper address
Given as upper 10 bits.



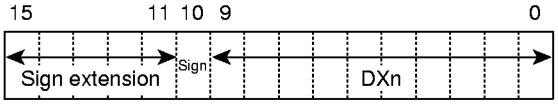
1-bit/pixel source start lower address
Given as lower 13 bits.
Source address is set as a byte address.



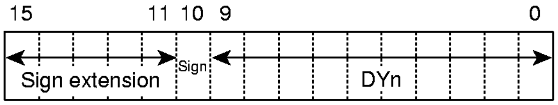
Number of vertices (2 ≤ n ≤ 65,535), absolute,
(1 ≤ n ≤ 65,535), relative
Given as unsigned 16-bit data.



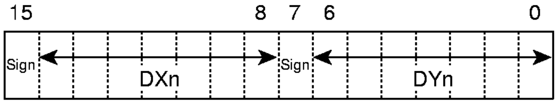
Source size TDX
Given as unsigned max. 10-bit data.
TDX can only be set in 8-pixel units.



Absolute coordinate
Vertex coordinate DXn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

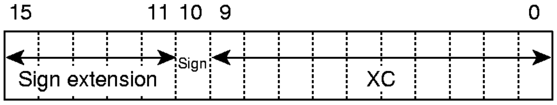


Absolute coordinate
Vertex coordinate DYn (2 ≤ n ≤ 65,535)
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

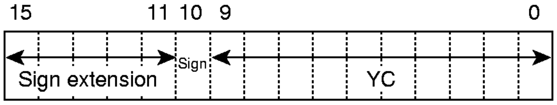


Relative coordinates
Vertex coordinates DXn, DYn (1 ≤ n ≤ 65,535)
Given as signed 8-bit data.

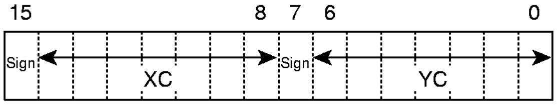
MOVE, RMOVE



Absolute coordinate
Vertex coordinate XC
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

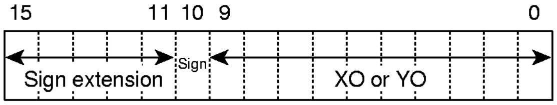


Absolute coordinate
Vertex coordinate YC
Given as signed 11-bit data.
Use sign extension in upper vacant bits.

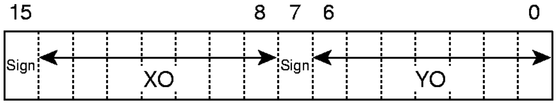


Relative coordinates
Vertex coordinates XC, YC
Given as signed 8-bit data.

LCOFS, RLCOFS



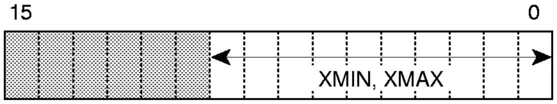
Relative specification
Local offset values XO, YO
Given as signed 11-bit data.
Use sign extension in upper vacant bits.



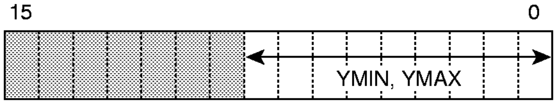
Relative specification
Local offset values XO, YO
Given as signed 8-bit data.

UCLIP, SCLIP

▨ : Fixed at 0



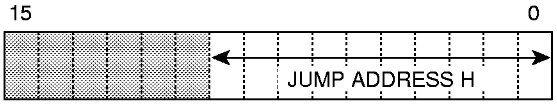
Left and right X coordinates XMIN, XMAX
Given as unsigned 10-bit data.



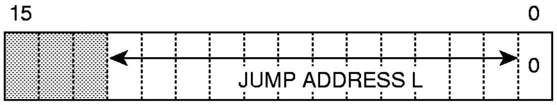
Upper and lower Y coordinates YMIN, YMAX
Given as unsigned 9-bit data.

JUMP

▨ : Fixed at 0



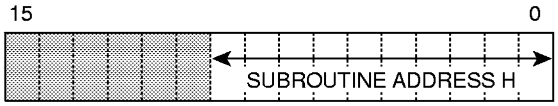
Jump destination upper address
Given as upper 10 bits.



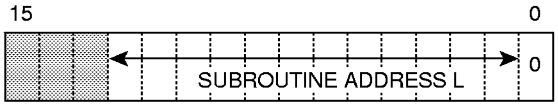
Jump destination lower address
Given as lower 13 bits.
Jump destination address is set as a word address.

GOSUB

▨ : Fixed at 0



Subroutine upper address
Given as upper 10 bits.



Subroutine lower address
Given as lower 13 bits.
Subroutine destination address is set as a word address.

Appendix C Drawing Algorithms

Straight Line Drawing Algorithms

- 8-point drawing and 4-point drawing

Figures C-1 (a) and (b) show examples of straight lines plotted on a bit-mapped display. Circles in the figures represent pixels. Due to the characteristics of a bit-mapped display, a straight line is drawn with the pixels arranged in a path differing slightly from an actual straight line. The same line is drawn in figures C-1 (a) and (b), but the algorithms are different, and so the pixel arrangements are also different. In both figures the line starts at the bottom left of the figure and is drawn dot by dot toward the top right corner. With the method shown in figure C-1 (a), the next dot drawn is to the right, or diagonally to the upper right, of the current dot. With the method shown in figure C-1 (b), on the other hand, the next dot drawn is to the right of, or directly above, the current dot.

For the sake of convenience, the method in figure C-1 (a) is here called 8-point drawing, and that in figure C-1 (b), 4-point drawing.

The difference between 8-point and 4-point drawing is illustrated in figure C-2. With 4-point drawing, the move to draw the next dot can be made in one of only four directions, up, down, left, or right (figure C-2 (b)). With 8-point drawing, moves can also be made in the four diagonal directions (figure C-2 (a)).

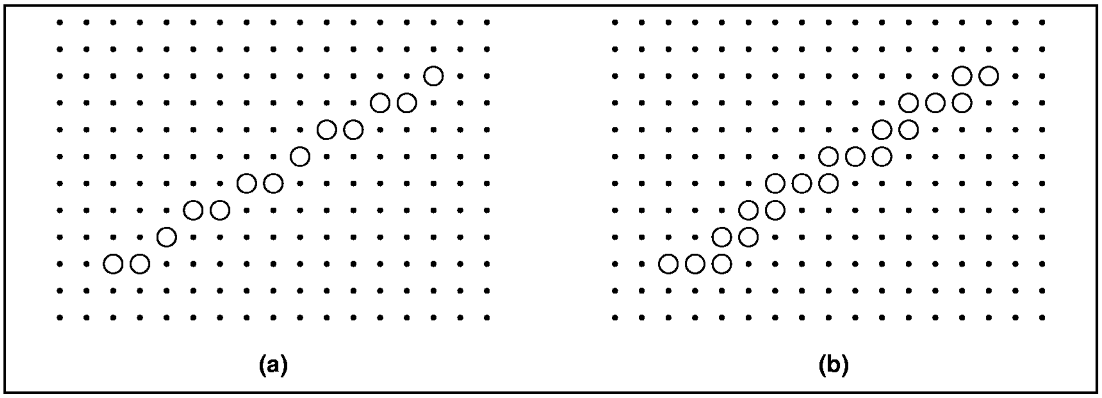


Figure C-1 Two Representations of a Straight Line on a Raster Display

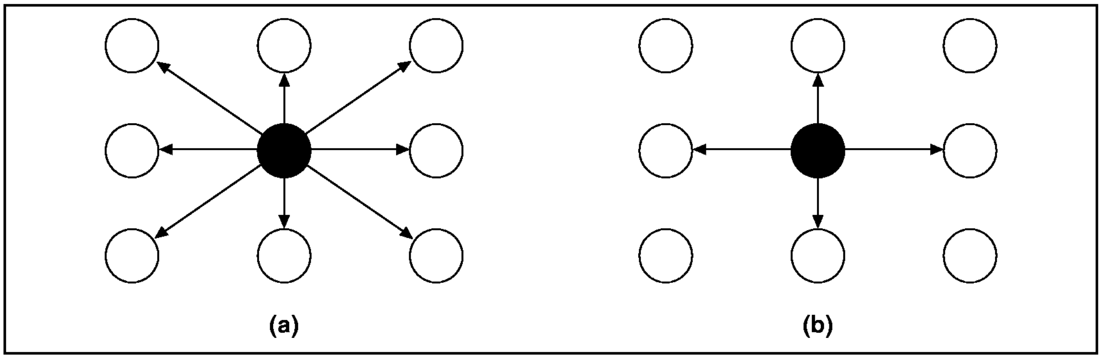


Figure C-2 Comparison of (a) 8-Point Drawing and (b) 4-Point Drawing

Next, 8-point drawing straight line approximation is described, using figure C-3 (a). After pixel A is drawn, either pixel B or pixel C is selected; the basis for selection is proximity to an actual straight line. The same approach is also used in 4-point drawing (figure C-3 (b)).

A comparison between 8-point drawing and 4-point drawing shows that closer approximation to a straight line can be achieved with 8-point drawing. However, the algorithm is correspondingly complex, requiring longer processing time.

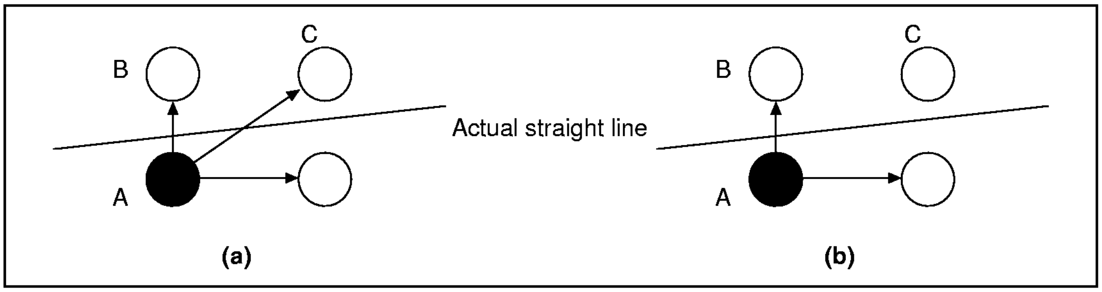


Figure C-3 Drawing Dot Determination Process in (a) 8-Point Drawing and (b) 4-Point Drawing

Readers interested in drawing algorithms can find further information in the sources listed below.

1. Jerry van Aken: “Curve-Drawing Algorithms for Raster Display,” ACM Trams. Graph. Vol. 4, No. 2 (April, 1985), 147–169
2. J.E. Bresenham: “Algorithm for Computer Control of a Digital Plotter,” IBM Syst. J. Vol. 4, No. 1 (1965), 25–30
3. J.E. Bresenham: “A Liner Algorithm for Incremental Digital Display of Digital Arcs,” Commum. ACM. Vol. 20, No. 2 (February 1977), 100–106
4. P.E. Danielsson: “Incremental Curve Generation,” IEEE Trans. Comput. Vol. C-19 (September 1970), 783–793
5. W.J.Jr. Bernard: “An Improved Algorithm for the Generation of Nonparametric Curves,” IEEE Trans. Comput. Vol. C-22, No. 12 (December 1973), 1052–1060
6. Jerry van Aken: “An Efficient Ellipse-Drawing Algorithm,” IEEE Comput. Graph & Appl. Vol. 4, No. 9 (September 1984), 24–35
7. Y. Suenaga: “A High-Speed Algorithm for the Generation of Straight Lines and Circular Arcs,” IEEE Trans. Comput. Vol. C-28, No. 10 (October 1979), 728–736

Appendix D Package Dimensions

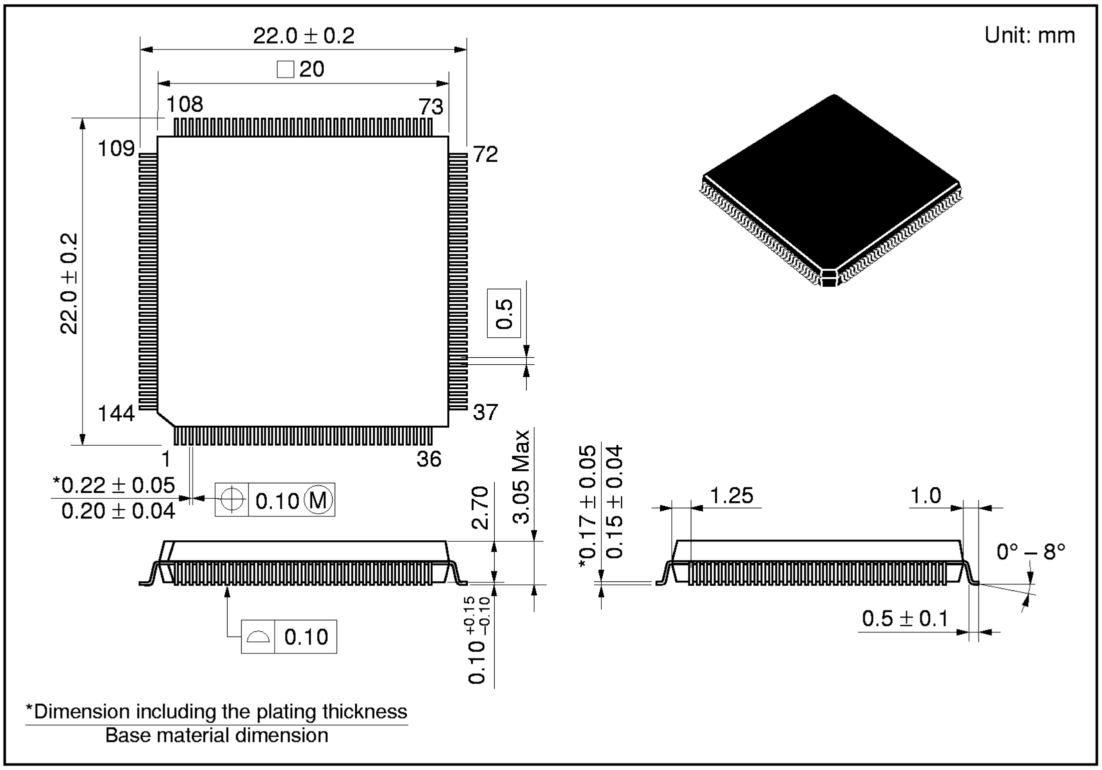


Figure D-1 Package Dimensions (FP-144G)