MB86297A
Carmine
Hardware Manual

Revision 1.28　11th March 2009

FUJITSU

- The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU or any third party or does FUJITSU warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite). Please note that FUJITSU will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

Trademarks

Sun is a trademark of Sun Microsystems, Inc. in the United States.

Windows is a trademark of Microsoft Corporation in the United States and other countries

Verilog-XL is a trademark of Cadence Design Systems, Inc.

UNIX is a registered trademark, which is licensing X/Open Company Limited., in the United States and other countries.

Revision History

Rev.No	Date	Contents
01		Initial Release after translation (Note: this version was updated in more than one version)
02		Preliminary version
0.1	2005/07/25	First draft version after translation (Note: this version was updated in more than one version)
0.2	2005/08/22	Pin table updated
0.7	2005/12/08	General update, formating not yet finished Corrected “1.1 Overview”. Added “1.3 Signal”. Added “1.4 Pin Assignement”. Added “1.5 Pin Function”. Corrected “1.6 Address Map”. Added “2 DRAM Controller”. Added “3 PCI Interface”. Added “I2C Interface”. Added and corrected “5 CARMINE CONTROL”. Corrected “6 KOTTOS”. Added and corrected “7 Display Controller”. Added and corrected “8 Video Capture”. Added “9 Electrical Characteristics”. Overview Changed “I/O column”, in Pin Assignment, for pins used as output pins at test time. Added description of processing of unused pins for when the graphics memory interface is used via a 32-bit data bus. Added “Description of Test Pin (JTAG)”. Added “Description of Interrupt Registers”. DRAM Controller Added “DRAM CTRL DDRIF1 Register”. Video Capture Changed English text into Japanese. Deleted “Description of RGB555 mode”. Added “Description of RGB Input function” and “Register Description”. Electrical Characteristics Corrected “Recommended Operating Conditions for Standerd CMOS I/O”. Added “Standard CMOS I/O V-I Characteristics”. Added “PCI I/O DC Characteristics”.
0.71	2005/12/29	Content is equal to 0.7, content table was included
0.72	2006/02/06	Basing on version 0.7

		History table updated, list of modification from rev. 02 to rev. 0.7 Formatting finished and a few details were added after translation was finished
1.0	2006/07/21	Initial Release
1.1	2007/03/27	Chapter 2 is replaced with the order in Chapter 3. 6. KOTTOS(2D/3D Graphics) ➢ Bitmap data order in explanation of DrawBitmapP had been opposite. ➢ Added "Need setting 1 to CO field of MDR0 in BltCopyCompressedP and BltCopyCompAlphaMapP". ➢ Added the explanation about Z calculation of Shadow and Non-TopLeft mode. ➢ Added the EQUAL judgement isn't exact in when Z value has slope. ➢ Added the explanation of an anti-aliasing with blend function. And added the explanation of an anti-aliasing with texture. ➢ The value of command in DrawVertex2i had been wrong. 7. Display Controller ➢ VCCC register description change description about software reset and add WBsr. ➢ L1BLD register description: correct error description about L1AS-bit ➢ WBM register description: change description about functions of each bit ➢ Add description about display layer limit 8. Video Capture ➢ 8.2 Selection of input port describe that RGB input can be also used as 656 input ➢ 8.6.1, 8.6.2 : change paragraph number in them
1.20	2007/06/05	Minor English corrections For better understanding, corrected VCCC register, removed references to POM bit (DCM3 register). Correct HSP register (15 clocks latency).
1.21	2007/06/12	Added short description of Field Detection bit (4) in DCM0/DCM1 register
1.22	2007/06/25	Corrected DCM3 description (DCKD field) +17 to +16 PLL Clocks. Added: DCM0/1 bit 5 description.
1.23	2008/03/20	Corrected Register Overview (L5 for 08C, 090, 094, 098). Corrected image positioning section 6.5.9 (text was behind image)
1.24	2008/04/01	Corrected value for CHLPF_x (11) + K3 from 10/32 to 8/32 (section 8.4.3).
1.25	2008/04/03	Corrected typo in CSC (Capture SCale), corrected explanations in 7.11 'Simultaneous layer display restrictions'. Section 8.5.2 Interrupt Status, removed last sentence and changed clearing order. Corrected value of bit 16 in RGBS register. Changed description of RGBCMb register.
1.26	2008/04/30	Added width of SZP and DSP bitfields in section 6.4.4. (G_BitBlt).
1.27	2008/05/27	Corrected typo in description of DEVSEL timing (Status register IDSEL+06h). Chapter 3, replaced 'disabled' with word 'prohibited' Section 7.12.2 (VCCC) corrected address to 0x1ff8

1.28	2009/03/11	Added description of bits 15 + 14 (FIFO size) and renamed register from 'DRAM CTRL Reserve 1' to 'DRAM CTRL FIFO Size'. Reworked descriptions in section 9.3 Precautions at Power ON Corrections to description of Data Format in Display Controller

CONTENTS

1 Specification of Carmine Chip

1.1 Overview

Features

- Geometry engine

  Carmine contains floating point operation pipelines dedicated to geometry processing for graphics processing, which allows 2D/3D graphics vertex processing including lighting at max 10M vertices/sec.

- Rendering engine

  Carmine contains a completely newly designed 2D/3D rendering engine. Carmine's multiple hardware for pixel processing improves performance of Fujitsu GDC controller Coral Series, and also supports hardware acceleration of OpenGL-ES.

- Display controller

  Carmine contains display controllers compatible with Fujitsu GDC controller Coral Series. The maximum resolution is 1280 × 1024, and the display controllers has two independent channels. Carmine has also two independent digital video output ports. Each digital video output port allows transparent processing, such as overlay display up to eight layers and alpha planes for four layers.

- Digital video capturing

  Captures video images on television, etc., and displays them on the same screen as drawn images. Carmine captures video images from two independent channels.

- CPU interface

  Carmine is connected to a 32-bit PCI bus (Rev. 2.2) (the maximum operating frequency is 66 MHz).

- External memory interface

  DDR SDRAM can be used. Selectable data bus width (64-bit/32-bit). The maximum data transfer frequency is 266 Mbps (clock frequency is 133 MHz). ***Table 1.1.1*** shows the available memory configuration.

Table 1.1.1 External Memory Configuration

Memory type	Data bus width	Number of memories used	Total capacity
DDR SDRAM 128 Mbits (x16)	32 Bits	2	32 Mbytes
DDR SDRAM 256 Mbits (x16)	32 Bits	2	64 Mbytes
DDR SDRAM 128 Mbits (x16)	64 Bits	4	64 Mbytes
DDR SDRAM 256 Mbits (x16)	64 Bits	4	128 Mbytes

- Others

  CMOS 90nm technology, 8-layer
  TEBGA543 (Thermally Enhanced Ball Grid Array) package, θja =17°C/W (wind speed 0 mm)
  Power supply voltage: 1.2 V (internal logic)/2.5 V (memory interface)/3.3 V (I/O)
  Power consumption: 2.3 W

1.2 Entire Block Diagram

Fig. 1.1 shows the Carmine block diagram.

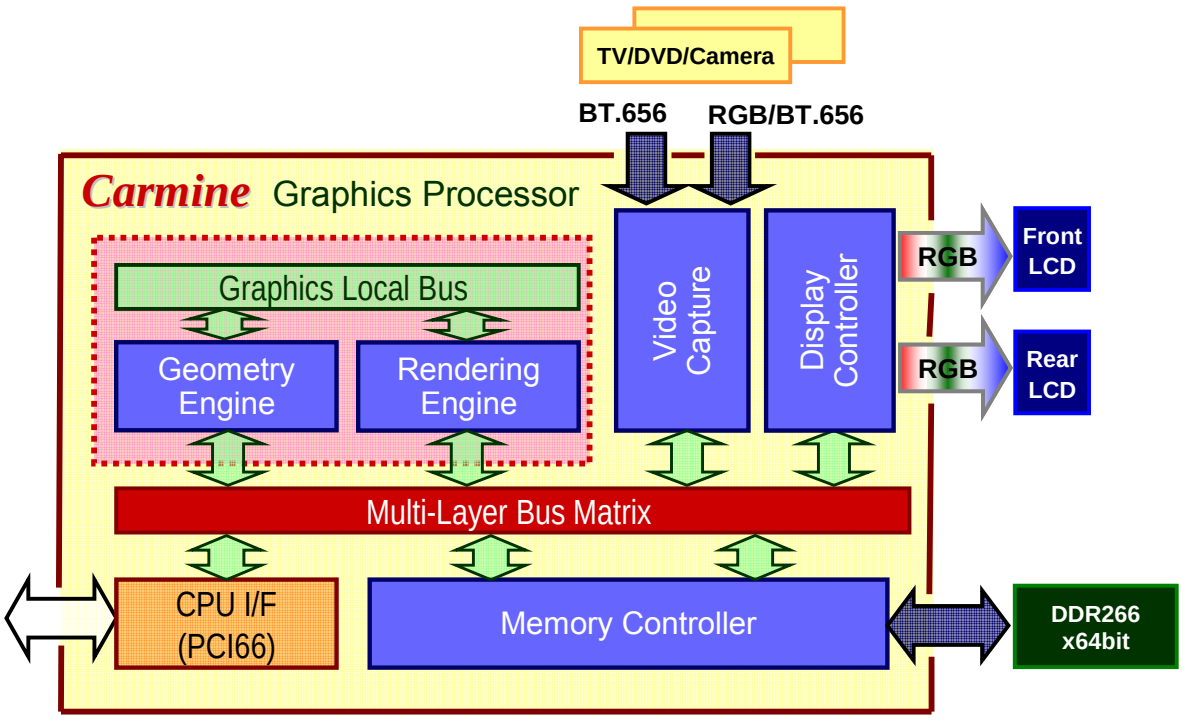


Fig. 1.1 Carmine Block Diagram

1.3 Signal

1.3.1 Signal Line

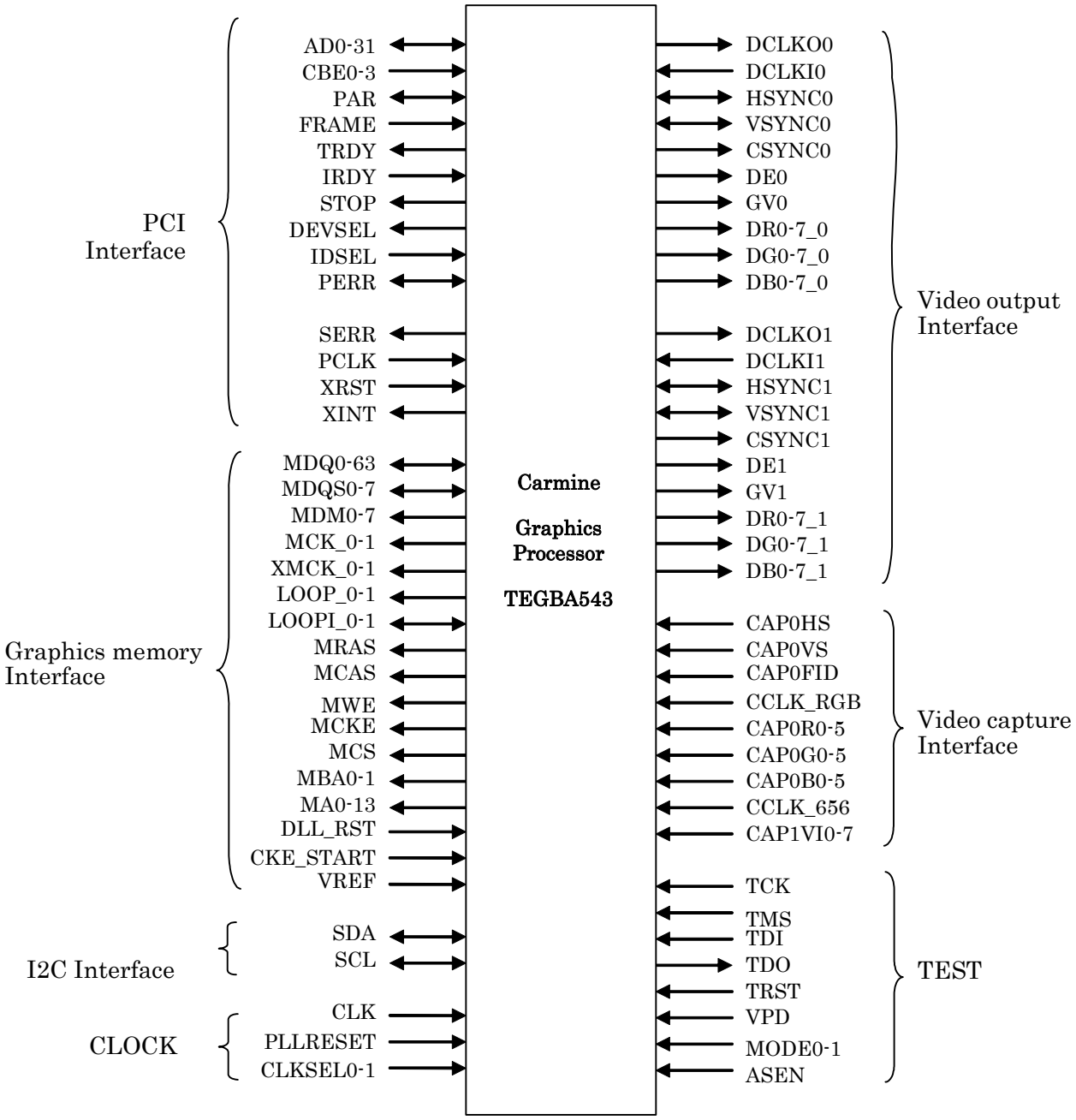


Fig. 1.2 Carmine Signal Line

1.4 Pin Assignment

1.4.1 TEBGA543 Package Pin Assignment (TOP_VIEW)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
A	VSS	VSS	DG3_0	DG0_0	DR4_0	DR0_0	TCK	CLK	VSS	PLLRESET	AD2	AD5	CBE0	AD10	AD13	SERR	VSS	PCLK	AD18	AD21	AD24	AD27	AD30	CBE3	VSS	VSS	A
B	VSS	VSS	DG4_0	DG1_0	DR5_0	DR1_0	TMS	ASEN	VSS	IDSEL	AD1	AD4	AD7	AD9	AD12	AD15	VSS	IRDY	AD17	AD20	AD23	AD26	AD29	AD31	VSS	VSS	B
C	DG6_0	DG5_0	VSS	DG2_0	DR6_0	DR2_0	TDI	XRST	MODE0	CLKSEL1	AD0	AD3	AD6	AD8	AD11	AD14	STOP	TRDY	AD16	AD19	AD22	AD25	AD28	VSS	KE_STAR	DLL_RST	C
D	DB1_0	DB0_0	DG7_0	VSS	DR7_0	DR3_0	TDO	TRST	MODE1	CLKSEL0	XINT	VDDE	VDDI	VDDI	CBE1	PAR	PERR	DEVSEL	FRAME	VDDE	VSS	CBE2	VSS	VPD	SCL	SDA	D
E	DB5_0	DB4_0	DB3_0	DB2_0	VSS	VSS	VDDE	VDDI	VDDI	VSS	VSS	VDDE	VDDI	VDDI	VDDE	VDDE	VSS	VDDI	VDDE	VDDE	VSS	VSS	VSS	VSS	MDQ14	MDQ15	E
F	DE0	CSYNC0	DB7_0	DB6_0	VSS	VSS	VDDE	VDDI	PLLVDD	PLLVSS	VSS	VDDE	VSS	VSS	VDDE	VSS	VSS	VDDI	VDDI	VDDE	VSS	VSS	VSS	VSS	MDQ12	MDQ13	F
G	DCLKI0	GV0	VSYNC0	HSYNC0	VDDE	VDDE															VDDE3	VDDE3	MDQ17	MDQ16	MDQ10	MDQ11	G
H	DCLKO1	DCLKO0	DR1_1	DR0_1	VDDE	VDDI															VDDI	VDDE3	MDQ19	MDQ18	MDQ8	MDQ9	H
J	VSS	VSS	DR3_1	DR2_1	VDDI	VDDI															VDDI	VDDI	MDQ21	MDQ20	MDM1	MDQS1	J
K	DR7_1	DR6_1	DR5_1	DR4_1	VSS	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VREF	VSS	MDQ23	MDQ22	VSS	VSS	K
L	DG3_1	DG2_1	DG1_1	DG0_1	VDDE	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VSS	VDDE3	MDM2	MDQS2	VSS	VSS	L
M	DG7_1	DG6_1	DG5_1	DG4_1	VDDE	VDDE				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VDDE3	VDDE3	VSS	VSS	MDQ1	MDQ0	M
N	DB2_1	DB1_1	DB0_1	VDDI	VDDI	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VSS	VDDI	VDDI	VDDI	MDQ3	MDQ2	N
P	DB5_1	DB4_1	DB3_1	VDDI	VDDI	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VSS	VDDI	MDQ30	MDQ31	MDQ5	MDQ4	P
R	DE1	CSYNC1	DB7_1	DB6_1	VDDE	VDDE				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VDDE3	VDDE3	MDQ28	MDQ29	MDQ7	MDQ6	R
T	DCLKI1	GV1	VSYNC1	HSYNC1	VDDE	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VSS	VDDE3	MDQ26	MDQ27	MDM0	MDQS0	T
U	VSS	VSS	CAP0VS	CAP0HS	VSS	VSS				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				VREF	VSS	MDQ24	MDQ25	VSS	VSS	U
V	CLK_RGB	CAP0R1	CAP0R0	CAP0FID	VDDI	VDDI															VDDI	VDDI	MDM3	MDQS3	VSS	VSS	V
W	CAP0R5	CAP0R4	CAP0R3	CAP0R2	VDDE	VDDI															VDDI	VDDE2	MA12	MA13	VSS	LOOPI_0	W
Y	CAP0G3	CAP0G2	CAP0G1	CAP0G0	VDDE	VDDE															VDDE2	VDDE2	MA11	MCKE	VSS	LOOP_0	Y
AA	CAP0B1	CAP0B0	CAP0G5	CAP0G4	VSS	VSS	VDDE1	VDDI	VDDI	VREF	VSS	VDDE1	VSS	VSS	VDDE1	VSS	VREF	VDDI	VDDI	VDDE2	VSS	VSS	MA8	MA9	VSS	XMCK_0	AA
AB	CAP0B5	CAP0B4	CAP0B3	CAP0B2	VSS	VSS	VDDE1	VDDE1	VDDI	VSS	VDDE1	VDDE1	VDDI	VDDI	VDDE1	VDDE1	VSS	VDDI	VDDE2	VDDE2	VSS	VSS	MA6	MA7	VSS	MCK_0	AB
AC	CAP1 VI2	CAP1 VI1	CAP1 VI0	VSS	MDQ62	MDQ60	MDQ58	MDQ56	MDM7	VSS	VSS	MDQ49	MDQ51	MDQ53	MDQ55	MDM6	VSS	VSS	MCAS	MCS	MBA1	MA0	VSS	MA4	MA5	VSS	AC
AD	CLK_656	CAP1 VI3	VSS	CAP1 VI6	MDQ63	MDQ61	MDQ59	MDQ57	MDQS7	VSS	VSS	MDQ48	MDQ50	MDQ52	MDQ54	MDQS6	VSS	VSS	MWE	MRAS	MBA0	MA10	MA2	VSS	VSS	VSS	AD
AE	VSS	VSS	CAP1 VI4	CAP1 VI7	VSS	VSS	MDQ33	MDQ35	MDQ37	MDQ39	MDM4	VSS	VDDI	MDQ46	MDQ44	MDQ42	MDQ40	MDM5	VSS	VSS	VSS	VSS	MA1	MA3	VSS	VSS	AE
AF	VSS	VSS	CAP1 VI5	VSS	VSS	VSS	MDQ32	MDQ34	MDQ36	MDQ38	MDQS4	VSS	VDDI	MDQ47	MDQ45	MDQ43	MDQ41	MDQS5	LOOPI_1	LOOP_1	XMCK_1	MCK_1	VSS	VSS	VSS	NOBALL	AF
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

TOP VIEW

PKG board power ring (V_{SS}, V_{DDE} and V_{DDI} are arranged in this order from inside toward outside)

1.4.2 Package Pin Assignment Table

JEDEC	Pin name	I/O	JEDEC	Pin name	I/O	JEDEC	Pin name	I/O	JEDEC	Pin name	I/O
A1	VSS	-	AF19	LOOPI_1	I/O	A15	AD13	I/O	AE8	MDQ35	I/O
B1	VSS	-	AF20	LOOP_1	Output	A14	AD10	I/O	AE9	MDQ37	I/O
C1	DG6_0	Output	AF21	XMCK_1	Output	A13	CBE0	Input	AE10	MDQ39	I/O
D1	DB1_0	Output	AF22	MCK_1	Output	A12	AD5	I/O	AE11	MDM4	I/O
E1	DB5_0	Output	AF23	VSS	-	A11	AD2	I/O	AE12	VSS	-
F1	DE0	Output	AF24	VSS	-	A10	PLLRESET	Input	AE13	VDDI	-
G1	DCLKI0	Input	AF25	VSS	-	A9	VSS	-	AE14	MDQ46	I/O
H1	DCLKO1	Output	AF26	NOBALL	-	A8	CLK	Input	AE15	MDQ44	I/O
J1	VSS	-	AE26	VSS	-	A7	TCK	Input	AE16	MDQ42	I/O
K1	DR7_1	Output	AD26	VSS	-	A6	DR0_0	Output	AE17	MDQ40	I/O
L1	DG3_1	Output	AC26	VSS	-	A5	DR4_0	Output	AE18	MDM5	I/O
M1	DG7_1	Output	AB26	MCK_0	Output	A4	DG0_0	Output	AE19	VSS	-
N1	DB2_1	Output	AA26	XMCK_0	Output	A3	DG3_0	Output	AE20	VSS	-
P1	DB5_1	Output	Y26	LOOP_0	Output	A2	VSS	-	AE21	VSS	-
R1	DE1	Output	W26	LOOPI_0	I/O	B2	VSS	-	AE22	VSS	-
T1	DCLKI1	Input	V26	VSS	-	C2	DG5_0	Output	AE23	MA1	Output
U1	VSS	-	U26	VSS	-	D2	DB0_0	Output	AE24	MA3	Output
V1	CCLK_RGB	Input	T26	MDQS0	I/O	E2	DB4_0	Output	AE25	VSS	-
W1	CAP0R5	Input	R26	MDQ6	I/O	F2	CSYNC0	Output	AD25	VSS	-
Y1	CAP0G3	I/O	P26	MDQ4	I/O	G2	GV0	Output	AC25	MA5	Output
AA1	CAP0B1	I/O	N26	MDQ2	I/O	H2	DCLKO0	Output	AB25	VSS	-
AB1	CAP0B5	I/O	M26	MDQ0	I/O	J2	VSS	-	AA25	VSS	-
AC1	CAP1VI2	Input	L26	VSS	-	K2	DR6_1	Output	Y25	VSS	-
AD1	CCLK_656	Input	K26	VSS	-	L2	DG2_1	Output	W25	VSS	-
AE1	VSS	-	J26	MDQS1	I/O	M2	DG6_1	Output	V25	VSS	-
AF1	VSS	-	H26	MDQ9	I/O	N2	DB1_1	Output	U25	VSS	-
AF2	VSS	-	G26	MDQ11	I/O	P2	DB4_1	Output	T25	MDM0	I/O
AF3	CAP1VI5	I/O	F26	MDQ13	I/O	R2	CSYNC1	Output	R25	MDQ7	I/O
AF4	VSS	-	E26	MDQ15	I/O	T2	GV1	Output	P25	MDQ5	I/O
AF5	VSS	-	D26	SDA	I/O	U2	VSS	-	N25	MDQ3	I/O
AF6	VSS	-	C26	DLL_RST	Input	V2	CAP0R1	Input	M25	MDQ1	I/O
AF7	MDQ32	I/O	B26	VSS	-	W2	CAP0R4	Input	L25	VSS	-
AF8	MDQ34	I/O	A26	VSS	-	Y2	CAP0G2	Input	K25	VSS	-
AF9	MDQ36	I/O	A25	VSS	-	AA2	CAP0B0	I/O	J25	MDM1	I/O
AF10	MDQ38	I/O	A24	CBE3	Input	AB2	CAP0B4	I/O	H25	MDQ8	I/O
AF11	MDQS4	I/O	A23	AD30	I/O	AC2	CAP1VI1	Input	G25	MDQ10	I/O
AF12	VSS	-	A22	AD27	I/O	AD2	CAP1VI3	Input	F25	MDQ12	I/O
AF13	VDDI	-	A21	AD24	I/O	AE2	VSS	-	E25	MDQ14	I/O
AF14	MDQ47	I/O	A20	AD21	I/O	AE3	CAP1VI4	I/O	D25	SCL	I/O
AF15	MDQ45	I/O	A19	AD18	I/O	AE4	CAP1VI7	Input	C25	CKE_ START	Input
AF16	MDQ43	I/O	A18	PCLK	Input	AE5	VSS	-	B25	VSS	-
AF17	MDQ41	I/O	A17	VSS	-	AE6	VSS	-	B24	AD31	I/O
AF18	MDQS5	I/O	A16	SERR	Output	AE7	MDQ33	I/O	B23	AD29	I/O
B22	AD26	I/O	AD9	MDQS7	I/O	C13	AD6	I/O	AC22	MA0	Output
B21	AD23	I/O	AD10	VSS	-	C12	AD3	I/O	AC23	VSS	-
B20	AD20	I/O	AD11	VSS	-	C11	AD0	I/O	AB23	MA6	Output
B19	AD17	I/O	AD12	MDQ48	I/O	C10	CLKSEL1	Input	AA23	MA8	Output
B18	IRDY	Input	AD13	MDQ50	I/O	C9	MODE0	Input	Y23	MA11	Output
B17	VSS	-	AD14	MDQ52	I/O	C8	XRST	Input	W23	MA12	Output
B16	AD15	I/O	AD15	MDQ54	I/O	C7	TDI	Input	V23	MDM3	I/O
B15	AD12	I/O	AD16	MDQS6	I/O	C6	DR2_0	Output	U23	MDQ24	I/O
B14	AD9	I/O	AD17	VSS	-	C5	DR6_0	Output	T23	MDQ26	I/O

B13	AD7	I/O	AD18	VSS	-	C4	DG2_0	Output	R23	MDQ28	I/O
B12	AD4	I/O	AD19	MWE	Output	D4	VSS	-	P23	MDQ30	I/O
B11	AD1	I/O	AD20	MRAS	Output	E4	DB2_0	Output	N23	VDDI	-
B10	IDSEL	Input	AD21	MBA0	Output	F4	DB6_0	Output	M23	VSS	-
B9	VSS	-	AD22	MA10	Output	G4	HSYNC0	I/O	L23	MDM2	I/O
B8	ASEN	Input	AD23	MA2	Output	H4	DR0_1	Output	K23	MDQ23	I/O
B7	TMS	Input	AD24	VSS	-	J4	DR2_1	Output	J23	MDQ21	I/O
B6	DR1_0	Output	AC24	MA4	Output	K4	DR4_1	Output	H23	MDQ19	I/O
B5	DR5_0	Output	AB24	MA7	Output	L4	DG0_1	Output	G23	MDQ17	I/O
B4	DG1_0	Output	AA24	MA9	Output	M4	DG4_1	Output	F23	VSS	-
B3	DG4_0	Output	Y24	MCKE	Output	N4	VDDI	-	E23	VSS	-
C3	VSS	-	W24	MA13	Output	P4	VDDI	-	D23	VSS	-
D3	DG7_0	Output	V24	MDQS3	I/O	R4	DB6_1	Output	D22	CBE2	Input
E3	DB3_0	Output	U24	MDQ25	I/O	T4	HSYNC1	I/O	D21	VSS	-
F3	DB7_0	Output	T24	MDQ27	I/O	U4	CAP0HS	I/O	D20	VDDE	-
G3	VSYNC0	I/O	R24	MDQ29	I/O	V4	CAP0FID	I/O	D19	FRAME	Input
H3	DR1_1	Output	P24	MDQ31	I/O	W4	CAP0R2	Input	D18	DEVSEL	Output
J3	DR3_1	Output	N24	VDDI	-	Y4	CAP0G0	Input	D17	PERR	Output
K3	DR5_1	Output	M24	VSS	-	AA4	CAP0G4	I/O	D16	PAR	I/O
L3	DG1_1	Output	L24	MDQS2	I/O	AB4	CAP0B2	I/O	D15	CBE1	Input
M3	DG5_1	Output	K24	MDQ22	I/O	AC4	VSS	-	D14	VDDI	-
N3	DB0_1	Output	J24	MDQ20	I/O	AC5	MDQ62	I/O	D13	VDDI	-
P3	DB3_1	Output	H24	MDQ18	I/O	AC6	MDQ60	I/O	D12	VDDE	-
R3	DB7_1	Output	G24	MDQ16	I/O	AC7	MDQ58	I/O	D11	XINT	Output
T3	VSYNC1	I/O	F24	VSS	-	AC8	MDQ56	I/O	D10	CLKSEL0	Input
U3	CAP0VS	I/O	E24	VSS	-	AC9	MDM7	I/O	D9	MODE1	Input
V3	CAP0R0	Input	D24	VPD	Input	AC10	VSS	-	D8	TRST	Input
W3	CAP0R3	Input	C24	VSS	-	AC11	VSS	-	D7	TDO	Output
Y3	CAP0G1	Input	C23	AD28	I/O	AC12	MDQ49	I/O	D6	DR3_0	Output
AA3	CAP0G5	I/O	C22	AD25	I/O	AC13	MDQ51	I/O	D5	DR7_0	Output
AB3	CAP0B3	I/O	C21	AD22	I/O	AC14	MDQ53	I/O	E5	VSS	-
AC3	CAP1VI0	I/O	C20	AD19	I/O	AC15	MDQ55	I/O	F5	VSS	-
AD3	VSS	-	C19	AD16	I/O	AC16	MDM6	I/O	G5	VDDE	-
AD4	CAP1VI6	I/O	C18	TRDY	Output	AC17	VSS	-	H5	VDDE	-
AD5	MDQ63	I/O	C17	STOP	Output	AC18	VSS	-	J5	VDDI	-
AD6	MDQ61	I/O	C16	AD14	I/O	AC19	MCAS	Output	K5	VSS	-
AD7	MDQ59	I/O	C15	AD11	I/O	AC20	MCS	Output	L5	VDDE	-
AD8	MDQ57	I/O	C14	AD8	I/O	AC21	MBA1	Output	M5	VDDE	-
N5	VDDI	-	E18	VDDI	-	U21	VREF	Input	K17	VSS	-
P5	VDDI	-	E17	VSS	-	T21	VSS	-	K16	VSS	-
R5	VDDE	-	E16	VDDE	-	R21	VDDE3	-	K15	VSS	-
T5	VDDE	-	E15	VDDE	-	P21	VSS	-	K14	VSS	-
U5	VSS	-	E14	VDDI	-	N21	VSS	-	K13	VSS	-
V5	VDDI	-	E13	VDDI	-	M21	VDDE3	-	K12	VSS	-
W5	VDDE	-	E12	VDDE	-	L21	VSS	-	K11	VSS	-
Y5	VDDE	-	E11	VSS	-	K21	VREF	Input	L11	VSS	-
AA5	VSS	-	E10	VSS	-	J21	VDDI	-	M11	VSS	-
AB5	VSS	-	E9	VDDI	-	H21	VDDI	-	N11	VSS	-
AB6	VSS	-	E8	VDDI	-	G21	VDDE3	--	P11	VSS	-
AB7	VDDE1	-	E7	VDDE	-	F21	VSS	-	R11	VSS	-
AB8	VDDE1	-	E6	VSS	-	F20	VDDE	-	T11	VSS	-
AB9	VDDI	-	F6	VSS	-	F19	VDDI	-	T12	VSS	-
AB10	VSS	-	G6	VDDE	-	F18	VDDI	-	T13	VSS	-
AB11	VDDE1	-	H6	VDDI	-	F17	VSS	-	T14	VSS	-
AB12	VDDE1	-	J6	VDDI	-	F16	VSS	-	T15	VSS	-
AB13	VDDI	-	K6	VSS	-	F15	VDDE	-	T16	VSS	-
AB14	VDDI	-	L6	VSS	-	F14	VSS	-	R16	VSS	-

AB15	VDDE1	-	M6	VDDE	-	F13	VSS	-	P16	VSS	-
AB16	VDDE1	-	N6	VSS	-	F12	VDDE	-	N16	VSS	-
AB17	VSS	-	P6	VSS	-	F11	VSS	-	M16	VSS	-
AB18	VDDI	-	R6	VDDE	-	F10	PLLVSS	-	L16	VSS	-
AB19	VDDE2	-	T6	VSS	-	F9	PLLVDD	-	L15	VSS	-
AB20	VDDE2	-	U6	VSS	-	F8	VDDI	-	L14	VSS	-
AB21	VSS	-	V6	VDDI	-	F7	VDDE	-	L13	VSS	-
AB22	VSS	-	W6	VDDI	-	K10	VSS	-	L12	VSS	-
AA22	VSS	-	Y6	VDDE	-	L10	VSS	-	M12	VSS	-
Y22	VDDE2	-	AA6	VSS	-	M10	VSS	-	N12	VSS	-
W22	VDDE2	-	AA7	VDDE1	-	N10	VSS	-	P12	VSS	-
V22	VDDI	-	AA8	VDDI	-	P10	VSS	-	R12	VSS	-
U22	VSS	-	AA9	VDDI	-	R10	VSS	-	R13	VSS	-
T22	VDDE3	-	AA10	VREF	Input	T10	VSS	-	R14	VSS	-
R22	VDDE3	-	AA11	VSS	-	U10	VSS	-	R15	VSS	-
P22	VDDI	-	AA12	VDDE1	-	U11	VSS	-	P15	VSS	-
N22	VDDI	-	AA13	VSS	-	U12	VSS	-	N15	VSS	-
M22	VDDE3	-	AA14	VSS	-	U13	VSS	-	M15	VSS	-
L22	VDDE3	-	AA15	VDDE1	-	U14	VSS	-	M14	VSS	-
K22	VSS	-	AA16	VSS	-	U15	VSS	-	M13	VSS	-
J22	VDDI	-	AA17	VREF	Input	U16	VSS	-	N13	VSS	-
H22	VDDE3	-	AA18	VDDI	-	U17	VSS	-	P13	VSS	-
G22	VDDE3	-	AA19	VDDI	-	T17	VSS	-	P14	VSS	-
F22	VSS	-	AA20	VDDE2	-	R17	VSS	-	N14	VSS	-
E22	VSS	-	AA21	VSS	-	P17	VSS	-			
E21	VSS	-	Y21	VDDE2	-	N17	VSS	-			
E20	VDDE	-	W21	VDDI	-	M17	VSS	-			
E19	VDDE	-	V21	VDDI	-	L17	VSS	-			

Notes:

V_{SS}/$PLLV_{SS}$	:	Ground
V_{DDE}	:	3.3 V power supply
V_{DDE1}, V_{DDE2}, V_{DDE3}	:	2.5 V/3.3V power supply for SDRAM
V_{DDI}	:	1.2 V power supply
$PLLV_{DD}$	:	PLL power supply (1.2 V)
V_{REF}	:	1/2 V_{DDE} (1-3) for SSTL2 mode; 0 V for LVCMOS mode

- Fujitsu recommends $PLLV_{DD}$ be separated on the board.
- Connect a bypass capacitor of good high frequency characteristics between power pin and ground pin. Place the capacitor as close as possible to the pins.
- The following pins are described as "I/O" in the I/O column. They are used as TEST output pins for Fujitsu shipping test of LSI, but used as input pins during normal operation:

  CAP0G5 to 3, CAP0B5 to 0, CAP0VS, CAP0HS, CAP0FID, CAP1VI0, CAP1VI6 to 4

1.5 Pin Function

1.5.1 PCI Interface

Table 1.5.1 PCI Interface Pins

Pin name	I/O	Function
AD0-31	In/Out	PCI address/data bus signal
CBE0-3	Input	PCI bus command/byte enable signal
FRAME	Input	PCI frame signal
IRDY	Input	PCI initiator ready signal
TRDY	Output	PCI target ready signal
DEVSEL	Output	PCI device selection signal
IDSEL	Input	PCI configuration device select signal
STOP	Output	PCI stop signal
PAR	In/Out	PCI parity signal
PERR	Output	PCI parity error signal
SERR	Output	PCI system error signal
XINT	Output	Interrupt output signal Note: This signal is output asynchronously with PCLK.
PCLK	Input	PCI clock signal
XRST	Input	System reset input signal

1.5.2 Video Output Interface

Table 1.5.2 Video Output Interface Pins

Pin name	I/O	Function
DCLKO0	Output	Display dot clock output signal
DCLKI0	Input	Dot clock input signal
HSYNC0	In/Out	Horizontal synchronization signal output For external synchronization mode, horizontal synchronization signal input
VSYNC0	In/Out	Vertical synchronization signal output For external synchronization mode, vertical synchronization signal input
CSYNC0	Output	Composite synchronization output signal
DE0	Output	Display valid period output signal
GV0	Output	Graphics/video switching output signal
DR7-0_0	Output	Digital image output signal (red)
DG7-0_0	Output	Digital image output signal (green)
DB7-0_0	Output	Digital image output signal (blue)
DCLKO1	Output	Display dot clock output signal
DCLKI1	Input	Dot clock input signal
HSYNC1	In/Out	Horizontal synchronization signal output For external synchronization mode, horizontal synchronization signal input
VSYNC1	In/Out	Vertical synchronization signal output For external synchronization mode, vertical synchronization signal input
CSYNC1	Output	Composite synchronization output signal
DE1	Output	Display valid period output signal
GV1	Output	Graphics/video switching output signal
DR7-0_1	Output	Digital image output signal (red)
DG7-0_1	Output	Digital image output signal (green)
DB7-0_1	Output	Digital image output signal (blue)

(Remarks)

- Adding an external circuit generates composite video signals.
- External video can be displayed synchronously with this LSI. Select the mode synchronous with DCLKI signal or the mode synchronous with the set dot clock, as with normal display.
- HSYNC and VSYNC signals enter input immediately after reset. Pull up these signal pins outside the LSI.
- GV signal is used to switch between graphics and external video when performing chroma-key. The GV signal outputs Low level when "video" is selected.
- Correspondence between 16-bit/pixel color mode and 8-bit/pixel color mode, and 8-bit digital RGB output pin is as follows.

(A) For 16-bit/pixel color mode:

RGB data in graphics memory R:G:B = 5:5:5	Digital RGB output R:G:B = 8:8:8
00000b	00000000b
00001b to 11111b	"111b" is added to lower 3 bits Calculation formula: X × 8 + 7

(B) For 8-bit/pixel color mode:

RGB data in color palette R:G:B = 6:6:6	Digital RGB output R:G:B = 8:8:8
000000b	00000000b
000001b to 111111b	"11b" is added to lower 2 bits Calculation formula: X × 4 + 3

In video capture YCbCr mode, when images are converted to RGB, they are converted to a full 8 bits of precision and then displayed.

Table 1.5.3 Processing When No Pins Used

Pin name	Pin state		Pin processing	Remarks
	At reset	At operation		
DCLKO0	Output	Output	OPEN	
DCLKI0	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
HSYNC0	Input	In/Out	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistor.
VSYNC0	Input	In/Out	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CSYNC0	Output	Output	OPEN	
DE0	Output	Output	OPEN	
GV0	Output	Output	OPEN	
DR7-0_0	Output	Output	OPEN	
DG7-0_0	Output	Output	OPEN	
DB7-0_0	Output	Output	OPEN	
DCLKO1	Output	Output	OPEN	
DCLKI1	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
HSYNC1	Input	In/Out	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
VSYNC1	Input	In/Out	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CSYNC1	Output	Output	OPEN	
DE1	Output	Output	OPEN	
GV1	Output	Output	OPEN	
DR7-0_1	Output	Output	OPEN	
DG7-0_1	Output	Output	OPEN	
DB7-0_1	Output	Output	OPEN	

*High resistance: about 4.7 kΩ

1.5.3 Video Capture Interface

Table 1.5.4 Video Capture Interface Pins

Pin name	I/O	Function
CCLK_RGB	Input	Digital video input clock signal (RGB)
CAP0R5 to 0	Input	Digital video data input signal (red)
CAP0G1 to 0/VI7 to 6	Input	Digital video data input signal (green)/RBT.656 input bits 7 to 6
CAP0G2	Input	Digital video data input signal (green)
CAP0G5-3	In/Out	
CAP0B5 to 0/VI5 to 0	In/Out	Digital video data input signal (blue)/ RBT.656 input bits 5 to 0
CAP0VS	In/Out	Digital video input vertical synchronization signal
CAP0HS	In/Out	Digital video input horizontal synchronization signal
CAP0FID	In/Out	Digital video input field identification signal
CCLK_656	Input	Digital video input clock signal (RBT.656)
CAP1VI0	In/Out	Digital video data input signal (RBT.656)
CAP1VI3 to 1	Input	
CAP1VI6 to 4	In/Out	
CAP1VI7	Input	

[Remarks]

The following pins are described as “I/O” in the I/O column. They are used as TEST output pins for Fujitsu shipping test of LSI, but used as input pins during normal operation:

CAP0G5 to 3, CAP0B5 to 0, CAP0VS, CAP0HS, CAP0FID, CAP1VI0, CAP1VI6 to 4

Table 1.5.5 Processing When No Pins Used

Pin name	Pin state		Pin processing	Remarks
	At reset	At operation		
CCLK_RGB	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CAP0R5 to 0	Input	Input	VDDE/GND	Pull up the pins to VDDE or pull down the pins to GND, via a high resistance.
CAP0G5 to 0	Input	Input	VDDE/GND	Pull up the pins to VDDE or pull down the pins to GND, via a high resistance.
CAP0B5 to 0	Input	Input	VDDE/GND	Pull up the pins to VDDE or pull down the pins to GND, via a high resistance.
CAP0VS	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CAP0HS	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CAP0FID	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CCLK_656	Input	Input	VDDE/GND	Pull up the pin to VDDE or pull down the pin to GND, via a high resistance.
CAP1VI7 to 0	Input	Input	VDDE/GND	Pull up the pins to VDDE or pull down the pins to GND, via a high resistance.

*High resistance: about 4.7 kΩ

1.5.4 I²C Interface

Table 1.5.6 I²C Interface Pins

Pin name	I/O	Function
SCL	In/Out	I²C serial clock
SDA	In/Out	I²C serial data

When not using the I2C interface, perform the following processing.

Table 1.5.7 Processing When No Pins Used

Pin name	Pin state	Processing for pin	Functionality
SCL	Input	VDDE	Pull up the pin to VDDE via a high resistance.
SDA	Input	VDDE	Pull up the pin to VDDE via a high resistance.

*High resistance: about 4.7 kΩ

1.5.5 Graphics Memory Interface

Table 1.5.8 Graphics Memory Interface Pins

Pin name	I/O	Function
MCK_1 to 0	Output	Graphics memory clock signal
XMCK_1 to 0	Output	Graphics memory clock signal
MDQ63 to 0	In/Out	Graphics memory bus data signal
MDQS7 to 0	In/Out	Graphics memory data strobe signal
MDM7 to 0	In/Out	Graphics memory data mask signal
MA13 to 0	Output	Graphics memory bus address signal
MBA1 to 0	Output	Graphics memory bank address signal
MCKE	Output	Graphics memory clock enable signal
MCS	Output	Graphics memory chip select signal
MRAS	Output	Graphics memory row address strobe signal
MCAS	Output	Graphics memory column address strobe signal
MWE	Output	Graphics memory write enable signal
LOOP1 to 0	Output	Carmine loop output signal
LOOPI1 to 0	In/Out	Carmine loop input signal
V_{REF}	Input	Reference power In SSTL mode: 1/2 V_{DDE} (1-3) input In LVCMOS mode: 0 V input
CKE_START (3,3V I/O)	Input	Sets the CKE state at reset. Follow the DRAM specification. (For DDR, Low is generally requested at power-on. Set the pin to "0".) 0: Outputs Low from MCKE at reset. 1: Outputs High from MCKE at reset.
DLL_RST (3.3V I/O)	Input	Carmine DLL reset input signal 0: Performs reset 1: Cancels reset

[Remarks]

Connect loop signals as follows: "LOOP0 to LOOPI0", or "LOOP1 to LOOPI1."

For details, refer to ***PCB Design Guideline***.

Table 1.5.9 and *Table 1.5.10* show the state and processing of unused pins for when the graphics memory interface is used via a 32-bit data bus.

Table 1.5.9 Processing of Unused Pins with VTT

Pin name	Pin state		Pin processing	Remarks
	Reset time	Normal state		
MCK_1	Output	Output	OPEN	
XMCK_1	Output	Output	OPEN	
MDQ63 to 32	Input	Input	VTT/GND	Pull up the pins to VTT by high resistance or pull down the pins by high resistance.
MDQS7 to 4	Input	Input	VTT/GND	Pull up the pins to VTT by high resistance or pull down the pins by high resistance.
MDM7 to 4	Output	Output	OPEN	
LOOP1	Output	Output		Connect to LOOPI1.
LOOPI1	Input	Input		Connect to LOOP1.

*High resistance: about 4.7 kΩ

Table 1.5.10 Processing of Unused Pins without VTT

Pin name	Pin state		Pin processing	Remarks
	Reset time	Normal state		
MCK_1	Output	Output	OPEN	
XMCK_1	Output	Output	OPEN	
MDQ63 to 32	Input	Output	OPEN	During reset period, the pins serve as input, but the through current is controlled so as not to flow. After reset is cancelled, the pins serve as output.
MDQS7 to 4	Input	Output	OPEN	
MDM7 to 4	Output	Output	OPEN	
LOOP1	Output	Output		Connect to LOOPI1.
LOOPI1	Input	Input		Connect to LOOP1.

1.5.6 Clock Input

Table 1.5.11 Clock Input Pins

Pin name	I/O	Function
CLK	Input	PLL reference clock input signal
PLLRESET	Input	PLL reset input signal When the pin is Low, reset is performed; when High, reset is cancelled. After turning on the power, be sure to input a signal that causes Low to High transition. See Section ***9.3 Precautions at Power ON***.
CLKSEL1 and 0	Input	Clock rate selection signal

Input a reference clock for the internal system clock and display dot clock to the CLK pin.

Select the internal system clock and display reference clock as follows depending on the CLKSEL pins and the built-in PLL. (The input clock frequency shown here is a typical value.)

CLKSEL1	CLKSEL0	Input clock frequency	Multiplication ratio	Display reference clock	System clock
L	L	Input 13.50 MHz	× 39	526.50 MHz	263.25 MHz
L	H	Input 14.32 MHz	× 37	529.84 MHz	264.92 MHz
H	L	Input 17.73 MHz	× 30	531.90 MHz	265.95 MHz
H	H	Input 33.33 MHz	× 16	533.33 MHz	266.66 MHz

(Caution)

- Use the crystal oscillator as CLK input.
- Use the display reference clock ranging from 400 to 533.33 MHz. If used outside this range, operation is not guaranteed.

1.5.7 Test

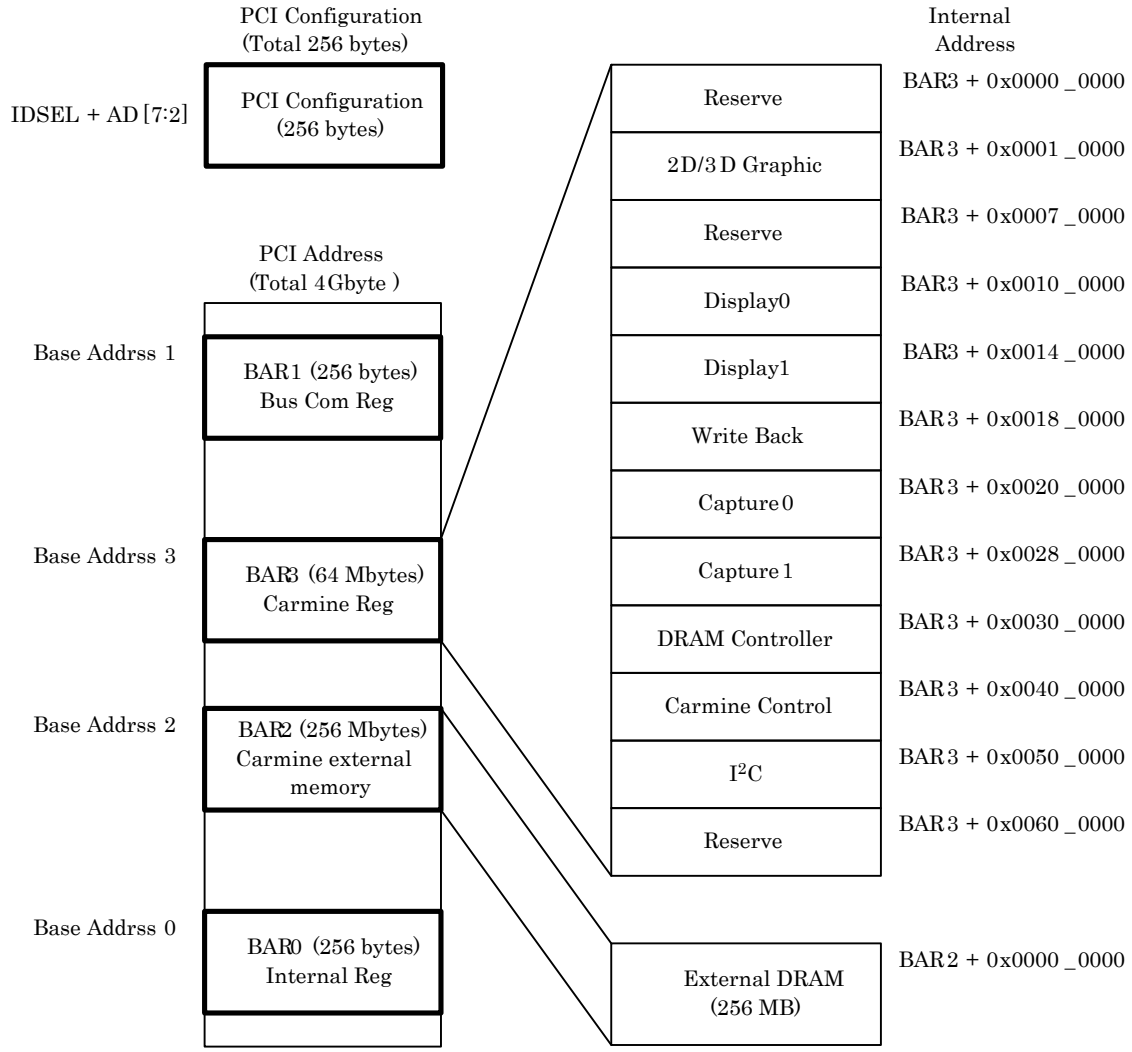
Table 1.5.12 Test Pins

Pin name	I/O	Function	Remarks
TCK	Input	JTAG	When the pin is not used, connect it to GND.
TMS	Input	JTAG	When the pin is not used, connect it to GND.
TDI	Input	JTAG	When the pin is not used, connect it to GND.
TDO	Output	JTAG	When the pin is not used, leave it open.
TRST	Input	JTAG	When this pin is not used, connect it to RESET or GND.
MODE1 and 0	Input	Mode switching pin 00: LVCMOS 11: SSTL2 Other settings are prohibited.	To switch between LVCMOS and SSTL2, the setting of DRAM controller is also needed. See Section *3.2 DRAM Initial Sequence*
ASEN	Input	Test pin	When this pin is not used, connect it to GND.
VPD	Input	Test pin	When this pin is not used, connect it to GND.

1.6 Address Map

Table 1.6.1 shows the internal address map of Carmine.

Table 1.6.1 Carmine Internal Address Map



1.7 Common Items

"Cremson", "Coral", and "Coral-PA" in the descriptions in this document refers to previous-generation graphics controllers. Cremson represents the MB86290A, Coral is the general term for the MB86293, MB86294, and MB86296, and Coral-PA represents the MB86296.

The following terms are used in register descriptions:

Register address

This is the address of the register.

Bit No.

This is the bit number.

Bit field name

This is each bit field name in the register.

R/W

This is the read/write access attribute of each bit field. The meaning of the symbols is as follows.

R0: The value read from the field is always "0". Write to the field is Don't Care.

W0: Only "0" can be written to the field.

W1: Only "1" can be written to the field.

R: Read from the field can be performed.

W: Write to the field can be performed.

RX: Read from the field can be performed. (The value read from the field is undefined.)

RW: Both read from and write to the field can be performed.

RW0: Read from and "0" write to the field can be performed.

Initial value

This is the initial value immediately after each bit field has been reset. When the initial value is not shown or when - is shown for the initial value, this means that the initial value is undefined.

Reserved bits

Use "0" as the value written to a reserved bit to maintain compatibility with future products.

Cycle of DRAM Contoroller

This is based on MCK(Graphics memory clock).

2 PCI Interface

2.1 Features

Main features are as follows:

PCI device interface

- Target function supporting PCI local bus specification Rev2.2
- 32 bits/33MHz and 32 bits/66MHz PCI interfaces
- 4 target spaces (BAR1 is used for debugging only. The user cannot use it)

  Memory space: 256 Mbytes, 64 Mbytes (BAR2, BAR3)
  I/O space: 256 bytes × 2 (BAR0, BAR1)
  Contains 32-byte FIFO bidirectionally

2.2 Function

2.2.1 Transfer to PCI Target

PCI target command

PCI commands received as target are as follows.

Command Type	Code (**C/BE[3:0]#**)
I/O Read	0010 (2_H)
I/O Write	0011 (3_H)
Memory Read	0110 (6_H)
Memory Write	0111 (7_H)
Memory Read Multiple	1100 (C_H)
Memory Read Line	1110 (E_H)
Memory Write and Invalidate	1111 (F_H)
Configuration Read	1010 (A_H)
Configuration Write	1011 (B_H)

Byte, Word, and Dword access is allowed for read/write access. All memory commands are treated as memory read ("0110") or memory write ("0111"). Burst transfer can be performed when memory command is executed. Burst transfer (address lower 2 bits) supports only linear increment ("00"). For other transfers, single transfer is performed. When I/O command or configuration command is executed, only single transfer is performed.

2.2.2 Generation of PCI Target Abort

Target abort is returned in the following cases:

In each case, set Signaled Target Abort (bit27) of the PCI Configuration Status register.

(1) Mismatch of the I/O access address and the byte enable

When the byte indicated by the lower 2 bits of the address phase does not match the byte enable of the data phase, target abort is returned.

(2) Address parity error

When address parity error is detected, target abort is returned to the access. At this time, when SERR Enable (bit8) of the PCI Configuration Status register is enabled, SERR# is asserted simultaneously.

2.3 Registers

2.3.1 Register Address Mapping

The configuration space (256 bytes), I/O space (256 bytes × 2) and memory space (256 Mbytes, 64 Mbytes) are provided as access space from the PCI bus.

Addressing from the PCI is arbitrarily determined by the setting of the Base Address register. Only the configuration space is specified by IDSEL+AD [7:2].

Address Offset	
IDSEL + 00_H : IDSEL + FF_H	Configuration register (256 bytes)
BAR0 + 00h : BAR0 + FF_H	Internal register (256 bytes) (Base Address 0 PCI I/O Space)
BAR1 + 00_H : BAR1 + FF_H	Internal register (256 bytes) Note: Internal Use Only (Base Address 1 PCI I/O Space)
BAR2 + 0000000_H : BAR2+ $FFFFFFF_H$	BAR2 memory space (256 Mbytes) (Base Address 2 PCI Mem Space)
BAR3 + 000000h : BAR3+$3FFFFFF_H$	BAR3 memory space (64 Mbytes) (Base Address 3 PCI Mem Space)

2.3.2 Configuration Register

Address Offset	31 24	23 16	15 8	7 0	Write
00H	Device ID		Vendor ID		N
04H	Status		Command		Y
08H	Class Code			Revision ID	N
0CH	BIST (Not Supported)	Header Type (Not Supported)	Latency Timer	Cache Line Size (Not Supported)	N
10H	**Base Address 0 for FPCI0x Internal Register(BAR0)**				Y *1
14H	**Base Address 1 for I/O Register (BAR1)**				Y *1
18H	**Base Address 2 for Mem Space (BAR2)**				Y *1
1CH	**Base Address 3 for Mem Space (BAR3)**				Y *1
20H	Base Address4 (Not Supported)				N
24H	Base Address5 (Not Supported)				N
28H	Cardbus CIS Pointer (Not Supported)				N
2CH	Subsystem ID		Subsystem Vendor ID		N
30H	Expansion ROM Base Address (Not Supported)				N
34H	Reserved			Cap_ptr	N
38H	Reserved				N
3CH	Max_Lat	Min_Gnt	Interrupt Pin	**Interrupt Line**	Y *2 [7:0]
40H : FFH	Reserved				N

The registers in the grayed-out area are the ones that are allowed to be written from the PCI.

*1 In the base addresses, only the set address bits can be rewritten. Rewrite of address ranges is not allowed.

*2 The set value of this register does not affect the operation of Carmine.

2.3.3 Internal Register (BAR0)

Address offset	31 24	23 16	15 8	7 0	Write
BAR0 + 00_H	Reserved		Reserved		N
BAR0 + 04_H	Reserved		Reserved		N
BAR0 + 08_H : BAR0 + $0F_H$	Reserved				N
BAR0 + 10_H	Target Access Address				N
BAR0 + 14_H	**Target Clear/Status**				Y
BAR0 + 18_H : BAR0 + BF_H	Reserved				N
BAR0 + $C0_H$	Reserved		Reserved		N
BAR0 + $C4_H$	Reserved			Reserved	N
BAR0 + $C8_H$	Reserved		Reserved		N
BAR0 + CC_H	Reserved	Reserved	Reserved		N
BAR0 + $D0_H$	Reserved	Reserved	Reserved	Reserved	N
BAR0 + $D4_H$	Reserved				N
BAR0 + $D8_H$	Reserved				N
BAR0 + DC_H	Reserved				N
BAR0 + $E0_H$	Reserved				N
BAR0 + $E4_H$: BAR0 + FF_H	Reserved				N

The registers in the grayed-out area are the ones that are allowed to be written from the PCI.

2.3.4 Configuration Register Details

Vendor ID register (Offset : IDSEL+00$_H$)

	Description	
Bit	15:0	Vendor ID. This field is used to identify device vendors. $FFFF_H$ cannot be assigned. $FFFF_H$ is returned when the device is not present
R/W	R	
Init	$10CF_H$	

Device ID register (Offset : IDSEL+02$_H$)

	Description	
Bit	31:16	Device ID. This field is used to identify particular devices.
R/W	R	
Init	$202B_H$	

Command register (Offset : IDSEL+04$_H$)

	Description	
Bit	0	IO space. Replies to access to I/O space when the value is "1",. Disables reply when the value is "0".
R/W	R/W	
Init	0	
Bit	1	Memory space. Replies to access to memory space when the value is "1",. Disables reply when the value is "0".
R/W	R/W	
Init	0	
Bit	2	Bus Master (Not supported)
R/W	R	
Init	0	
Bit	3	Special Cycles (Not supported)
R/W	R	
Init	0	
Bit	4	Memory Write and Invalidate Enable (Not supported)
R/W	R	
Init	0	
Bit	5	VGA Palette Snoop (Not supported)
R/W	R	
Init	0	
Bit	6	Parity Error Response. When the value is "1", PERR# is asserted when data parity error is detected. Disables PERR# reply when the value is "0".
R/W	R/W	
Init	0	
Bit	7	Reserved
R/W	R	
Init	0	
Bit	8	SERR# Enable. When the value is "1", SERR# is asserted when address parity error is detected. Disables SERR# reply when the value is "0".
R/W	R/W	
Init	0	
Bit	9	Fast Back-to-Back Enable (Not supported)
R/W	R	
Init	0	
Bit	15:10	Reserved
R/W	R	
Init	000000	

Status register (Offset : IDSEL+06H)

	Description	
Bit	18:16	Reserved
R/W	R	
Init	000	
Bit	19	Interrupt Status. Indicates interrupt status. This bit is set irrespective of the status of Interrupt Disable of the Command register.
R/W	R	
Init	0	
Bit	20	Capabilities List. The value "1" indicates that the Capabilities register exists.
R/W	R	
Init	0	
Bit	21	66 MHz Capable. The value "1" indicates that 66 MHz operation can be performed.
R/W	R	
Init	1	
Bit	22	Reserved
R/W	R	
Init	0	
Bit	23	Fast-back-to-back Capable. The value "1" indicates that it is possible to reply to the fast-back-to-back transaction for a different target.
R/W	R	
Init	1	
Bit	24	Master Data Parity Error (Not supported)
R/W	R	
Init	0	
Bit	26:25	DEVSEL Timing. Indicates timing to assert the DEVSEL#. 00: Fast Decode 01: Medium Decode 10: Slow Decode 11: Reserved Carmine uses Medium Decode ("01").
R/W	R	
Init	01	
Bit	27	Signaled Target Abort. Set "1" to this bit when Carmine ends the transaction by returning a target abort. Writing "1" to the bit clears the bit.
R/W	R/W	
Init	0	
Bit	28	Received Target Abort (Not supported)
R/W	R	
Init	0	
Bit	29	Received Master Abort (Not supported)
R/W	R	
Init	0	
Bit	30	Signaled System Error. Set "1" to this bit when Carmine asserts SERR#. Writing "1" to the bit clears the bit.
R/W	R/W	
Init	0	
Bit	31	Detected Parity Error. Set "1" to this bit when Carmine detects parity error. (The bit is set irrespective of the status of Parity Error Response of the Command register.) Writing "1" to the bit clears the bit.
R/W	R/W	
Init	0	

Revision ID register (Offset : IDSEL+08H)

	Description	
Bit	7:0	Revision ID.
R/W	R	
Init	03H	

Class Code register (Offset : IDSEL+09H)

	Description	
Bit	15:8	Programming Interface. This field is used to identify a programming interface included in class code.
R/W	R	
Init	00H	
Bit	23:16	Sub Class. This field is used to identify a subclass included in class code.
R/W	R	
Init	80H	
Bit	31:24	Base Class. This field is used to identify a base class included in class code.
R/W	R	
Init	03H	

Cache Line Size register (Offset : IDSEL+0CH)

	Description	
Bit	7:0	Cache Line Size (Not supported)
R/W	R	
Init	00H	

Latency Timer register (Offset : IDSEL+0DH)

	Description	
Bit	15:8	Header Type. (Not supported for multifunction devices)
R/W	R	
Init	00H	

Header Type register (Offset : IDSEL+0EH)

	Description	
Bit	23:16	BIST. (Not supported.))
R/W	R	
Init	00h	

BIST register (Offset : IDSEL+0FH)

	Description	
Bit	31:24	BIST (Not supported)
R/W	R	
Init	00H	

BaseAddress register0 (Offset : IDSEL+10H)

	Description	
Bit	0	Memory Space Indicator/IO Space Indicator. For Carmine, this bit is fixed for setting I/O space.
R/W	R	
Init	1	
Bit	2:1	Memory space: Type 00: Locate anywhere in 32-bit memory address space 01: Locate below 1MB memory address space 10: Locate anywhere in 64-bit memory address space 11: Reserved I/O space: bit 1: Reserved = fixed at "0". Bit 2: BaseAddress = fixed at "0".
R/W	R	
Init	00	
Bit	3	Memory space: Prefetchable The value "1" indicates that BAR0 space is prefetchable*. I/O space: Base Address = fixed at "0".
R/W	R	
Init	0	
Bit	31:8	BaseAddress. Base address for internal registers 256 bytes I/O space
R/W	R/W	
Init	-	

BaseAddress register1 (Offset : IDSEL+14H)

	Description	
Bit	0	Memory Space Indicator/IO Space Indicator. For Carmine, this bit is fixed for setting I/O space.
R/W	R	
Init	1	
Bit	2:1	Memory space: Type 00: Locate anywhere in 32-bit memory address space 01: Locate below 1MB memory address space 10: Locate anywhere in 64-bit memory address space 11: Reserved I/O space: bit 1 : Reserved = fixed at "0". Bit 2 : BaseAddress = fixed at "0".
R/W	R	
Init	00	
Bit	3	Memory space: Prefetchable The value "1" indicates that BAR1 space is prefetchable*. I/O space: Base Address = fixed at "0".
R/W	R	
Init	0	
Bit	31:8	BaseAddress. BAR1 base address 256 bytes I/O space
R/W	R/W	
Init	-	

BaseAddress Register2 (Offset : IDSEL+18$_H$)

	Description	
Bit	0	Memory Space Indicator/IO Space Indicator. For Carmine, this bit is fixed for setting memory space.
R/W	R	
Init	0	
Bit	2:1	Memory space: Type 00: Locate anywhere in 32-bit memory address space 01: Locate below 1MB memory address space 10: Locate anywhere in 64-bit memory address space 11: Reserved I/O space: bit 1 : Reserved = fixed at "0". Bit 2 : BaseAddress = fixed at "0".
R/W	R	
Init	00	
Bit	3	Memory space: Prefetchable The value "1" indicates that BAR2 space is prefetchable*. I/O space: Base Address = fixed at "0".
R/W	R	
Init	1	
Bit	31:28	BaseAddress. Base address for BAR2 memory space BAR2 is memory space.
R/W	R/W	
Init	-	

BaseAddress Register3 (Offset : IDSEL+1C$_H$)

	Description	
Bit	0	Memory Space Indicator/IO Space Indicator. For Carmine, this bit is fixed for setting memory space.
R/W	R	
Init	0	
Bit	2:1	Memory space: Type 00: Locate anywhere in 32-bit memory address space 01: Locate below 1MB memory address space 10: Locate anywhere in 64-bit memory address space 11: Reserved I/O space: bit 1 : Reserved = fixed at "0". Bit 2 : BaseAddress = fixed at "0".
R/W	R	
Init	00	
Bit	3	Memory space: Prefetchable The value "1" indicates that BAR3 space is prefetchable*. I/O space: Base Address = fixed at "0".
R/W	R	
Init	1	
Bit	31:26	BaseAddress. Base address for BAR3 memory space BAR3 is memory space.
R/W	R/W	
Init	-	

* Prefetchable: When the device meets the following conditions, memory space can be marked as "prefetchable".

(1) The device returns all bytes at read time irrespective of byte enable.

(2) Side effects such as data deletion, etc. will not occur even when read access is made to this space.

Cardbus CIS Pointer register (Offset : IDSEL+28H)

	Description	
Bit	31:0	Cardbus CIS Pointer (Not supported)
R/W	R	
Init	all0	

Subsystem Vendor ID register (Offset : IDSEL+2CH)

	Description	
Bit	15:0	Subsystem Vendor ID. This field is used to identify the vendor of the add-in board or subsystem on which the device is mounted. This field cannot be rewritten from the PCI.
R/W	R/W	
Init	00_H	

Subsystem ID register (Offset : IDSEL+2EH)

	Description	
Bit	31:16	Subsystem ID. This field is used to identify the device mounted to the add-in card or subsystem. This field cannot be rewritten from the PCI.
R/W	R/W	
Init	00_H	

Expansion ROM Base Address register (Offset : IDSEL+30H)

	Description	
Bit	31:0	Expansion ROM Base Address (Not supported)
R/W	R	
Init	All0	

Capabilities Pointer register (Offset : IDSEL+34h)

	Description	
Bit	7:0	Capabilities Pointer. This field indicates the offset where the Capabilities register exists, in the specific area of the Configuration register.
R/W	R	
Init	00_H	

Interrupt Line register (Offset : IDSEL+3CH)

	Description	
Bit	7:0	Interrupt Line. Register for interrupt routing
R/W	R/W	00_H -FE_H: Number of the interrupt line to which the device is connected
Init	00_H	FF_H: The interrupt line for the device is not connected to the interrupt controller of the system. The value written to this register does not affect the operation of Carmine. The initialization program writes the value to this field and the device driver and operating system reference the value.

Interrupt Pin register (Offset : IDSEL+3D$_H$)

		Description
Bit	15:8	Interrupt pin. This register is the interrupt pin register.
R/W	R	00_H: The device uses no interrupt pin.
Init	01_H	01_H: The device uses the interrupt pin INTA#.
		02_H: The device uses the interrupt pin INTB#.
		03_H: The device uses the interrupt pin INTC#.
		04_H: The device uses the interrupt pin INTD#.
		05_H -FF_H: Reserved.
		Carmine's PCI interrupt output supports only INTA# (01_H).

Min_Gnt register (Offset : IDSEL+3E$_H$)

		Description
Bit	23:16	Min_Gnt. Specify the burst period needed by the device (operation at 33MHz assumed), in increments of 250 ns (1/4 μs).
R/W	R	
Init	00_H	

Max_Lat register (Offset : IDSEL+3F$_H$)

		Description
Bit	31:24	Max_. Specify the interval at which the device issues transfer requests to the PCI bus (operation at 33MHz assumed), in increments of 250 ns (1/4 μs).
R/W	R	
Init	00_H	

2.3.5 Internal Register Details (BAR0)

Target Access Address register (Offset : BAR0 + 10_H)

	Description	
Bit	31:0	Target Access Address. Indicates the address of the last received PCI access. When access enters the locked state*, use this register to check the address of the last access. And, this register can be used to determine whether or not to use Target Clear (bit31) of the Target Clear/Status register to clear the access. Note that when the last access is a read access and is completed normally, an address to which 4h is added is shown because of lookahead.
R/W	R	
Init	all0	

Target Clear/Status register (Offset : BAR0 + 14_H)

	Description	
Bit	3:0	Target Access Command. Indicates the command for the last received PCI access. When access enters the locked state*, use this register to check the command for the last access. And, this register can be used to determine whether or not to use Target Clear (bit31) of the Target Clear/Status register to clear the access.
R/W	R	
Init	0_H	
Bit	7:4	Reserved
R/W	R	
Init	0_H	
Bit	11:8	Target Access Byte Enable. Indicates the byte enable for the last received access. When access enters the locked state*, use this register to check the byte enable for the last access. And, this register can be used to determine whether or not to use Target Clear (bit31) of the Target Clear/Status register to clear the access.
R/W	R	
Init	0_H	
Bit	15:12	Reserved
R/W	R	
Init	0_H	
Bit	23:16	Target FIFO Data Count. Indicates the number of remaining data in the target FIFO.
R/W	R	
Init	00_H	
Bit	30:24	Reserved
R/W	R	
Init	all0	
Bit	31	Target Clear. Writing "1" to the bit clears the current target access. Clearing access using this bit can be used to avoid the locked state*, etc. When access is cleared by this bit, the counter in the PCI block and the state machine in the AHB block are cleared. Read to the bit is disabled ("0").
R/W	W	
Init	0	

*: The "locked state" here means the state in which response cannot be made between PCI and the local bus due to something. It does not mean "PCI lock". Target Clear register is for avoiding the locked state, and so do not use it when transfer is performed correctly.

3 DRAM Controller

3.1 DRAM Refresh

Refresh is performed according to the setting of the DRAM CTRL REFRESH register.

3.2 DRAM Initial Sequence

3.2.1 I/O Mode Setting

Before executing DRAM initial sequence, set I/O mode (LVCMOS or SSTL2). Before DRAM controller is initialized, the receiving I/O transistor is OFF and so there is no problem. However, if Carmine is used in a mode other than these modes, IC may be destroyed in the worst case.

To set I/O mode, use the DRAM CTRL IO CONT0 register or DRAM CTRL IO CONT1 register.

Set the following depending on the mode used:

SSTL2: DRAM CTRL IO CONT0 register 0555_H
DRAM CTRL IO CONT1 register 0555_H

LVCMOS: DRAM CTRL IO CONT0 register 0111_H
DRAM CTRL IO CONT1 register 0111_H

3.2.2 Setting Procedures for DRAM Initial Sequence

Carmine executes the DRAM initial sequence after setting the DRAM CTRL STATES register of the DRAM controller. Figure 2.1 shows procedures for setting registers for the DRAM initial sequence.

Note that the setting procedure varies according to whether an externally connected DRAM requires DLL reset.

The initial sequence does not start until correct values are set to the bit field of DCTRL_STATES of the DRAM CTRL STATES register. Therefore, the following procedures *(1)* to *(7)* have no restrictions on its setting order because

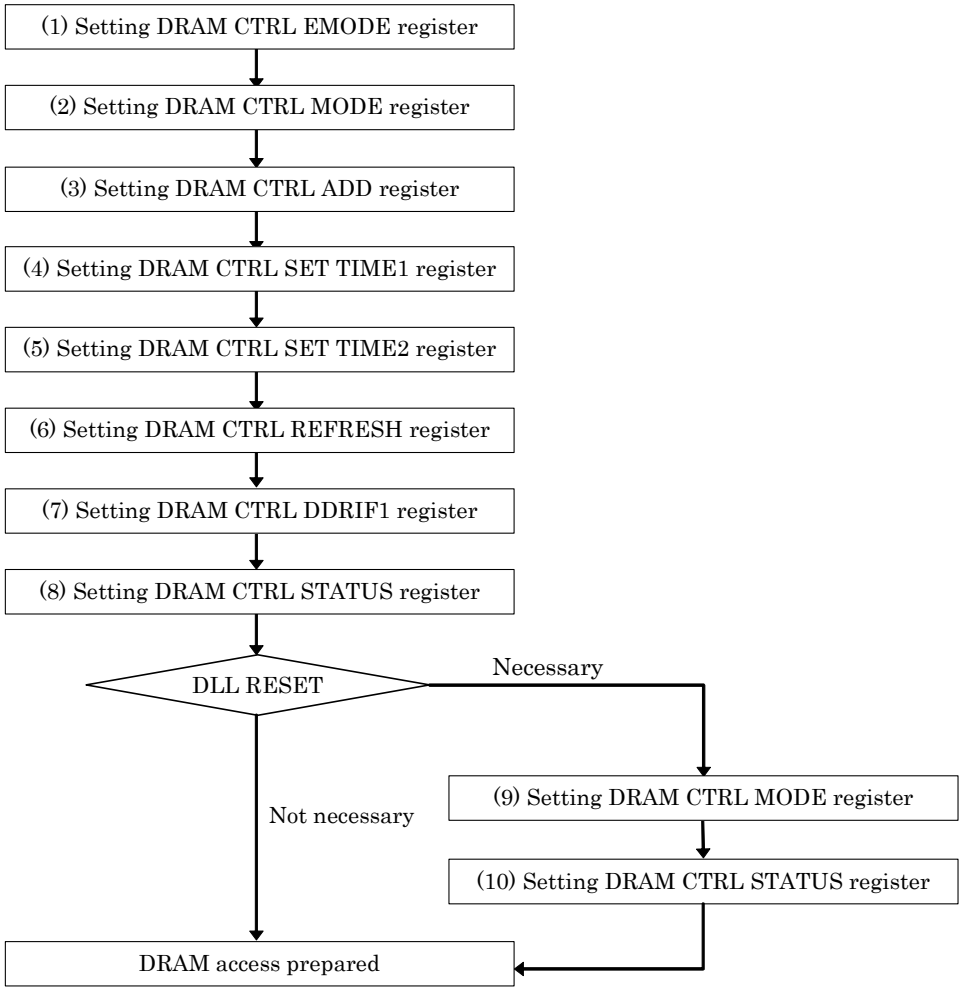


Fig. 3.1 Setting Procedures for DRAM Initial Sequence

[Remarks]

(1) Use this register when using DRAM that requires setting of extended mode register.

Set correct values to DRAM according to Specifications of the externally connected DRAM, and set "1" to the EBM bit field of the DRAM CTRL EMODE register.

(2) Use this register for setting the MODE register of the externally connected DRAM.

When DRAM requires DLL reset, set "0x000010" to the OPM bit field of the DRAM CTRL MODE register.

(8) When extended MODE register is set:

Set "0x0003" to the DCTRL_STATES bit field of the DRAM CTRL STATES register.

When extended MODE register is not set:

Set "0x0002" to the DCTRL_STATES bit field of the DRAM CTRL STATES register.

(9) Set "0x000000" to the DCTRL_STATES bit field of the DRAM CTRL MODE register after checking that "0x0000" is set to OPM bit field of the DRAM CTRL STATES register.

(10) Set "0002" to the STATES bit field of the DRAM CTRL STATES register.

3.2.3 Internal Status of DRAM Controller and Issued Command

DRAM controller used by Carmine operates according to the setting of the DRAM CTRL STATUS register. The following figure shows the transition order of the internal status of DRAM controller and commands issued at each status.

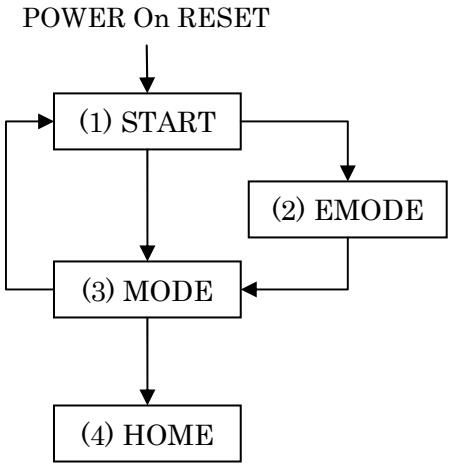


Table 3.2.1 Internal Status of DRAM Controller

	Status name	Issued command	Description
(1)	START	NOP	START status means a status after reset is performed. It continues to issue NOP command. The status remains unchanged until correct values are set to the DCTRL_STATES bit field of the DRAM CTRL STATES register. For setting values, see ***Register details***.
(2)	EMODE	EMRS	EMODE status issues EMRS command. It transits to MODE status after issuing the command.
(3)	MODE	MRS	MODE status issues MRS command. When resetting DLL embedded in DRAM, this status transits to START status after issuing the command. When not resetting DLL, it transits to HOME status.
(4)	HOME		HOME status accesses DRAM.

3.3 DRAM Controller Control Registers

3.3.1 Register List

Address	Register Name	Description
0x0030_0000	DRAM CTRL ADD	Address setting register
0x0030_0002	DRAM CTRL MODE	MODE setting register
0x0030_0004	DRAM CTRL EMODE	Extended MODE setting register
0x0030_0006	DRAM CTRL SET TIME1	Access timing setting register 1
0x0030_0008	DRAM CTRL SET TIME2	Access timing setting register 2
0x0030_000A	DRAM CTRL REFRESH	Refresh setting register
0x0030_000C	DRAM CTRL STATES	Status setting register
0x0030_000E	DRAM CTRL RESERVE0	Reserve0
0x0030_0010	DRAM CTRL RESERVE1	Reserve1
0x0030_0012	DRAM CTRL RESERVE2	Reserve2
0x0030_0014	DRAM CTRL DDRIF1	DDR I/F setting register
0x0030_0016	DRAM CTRL DDRIF2	DDR I/F setting register
0x0030_0018	DRAM CTRL RESERVE4	Reserve4
0x0030_001A	DRAM CTRL RESERVE5	Reserve5
0x0030_001C	DRAM CTRL RESERVE6	Reserve6
0x0030_001E	DRAM CTRL RESERVE7	Reserve7
0x0030_0020	DRAM CTRL RESERVE8	Reserve8
0x0030_0022	DRAM CTRL RESERVE9	Reserve9
0x0030_0024	DRAM CTRL IO CONT0	I/O control register 0
0x0030_0026	DRAM CTRL IO CONT1	I/O control register 1

3.3.2 Register Details

Base Address = 0x0030_0000

DRAM CTRL ADD

Set addresses to multiplexed and output Row and Column to DRAM, to this register.

Register address	**Base Address** + 00_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				BSEL				RSEL					CSEL		
R/W	–				RW				RW					RW		
Initial value	–															

Bit 2-0 CSEL (Column address select)
Indicates the starting bit of the column address that is output to DRAM.
010: Outputs address bits 11 to 2 (32-bit address).
011: Outputs address bits 12 to 3 (64-bit address).
Other settings: These are prohibited.

Bit 7-4 RSEL (Row address select)
Indicates the starting bit of the row address that is output to DRAM.
1010: Outputs address bits 23 to 10.
1011: Outputs address bits 24 to 11.
1100: Outputs address bits 25 to 12.
1101: Outputs address bits 26 to 13.
Other settings: These are prohibited.

Bit 11-8 BSEL (Bank address select)
Indicates the starting bit of the bank address that is output to DRAM.
0010: Outputs address bits 23 to 22.
0011: Outputs address bits 24 to 23.
0100: Outputs address bits 25 to 24.
0101: Outputs address bits 26 to 25.
0110: Outputs address bits 27 to 26.
0111: Outputs address bits 27 to 26
Other settings: These are prohibited.

Fig. 3.2 shows the relationship between the DRAM CTRL ADD register set values and internal addresses. For example, when connecting four "x16 DRAMs", each having four banks (2 bits), 8192 row addresses (13 bits) and 512 column addresses, the internal address of CSEL bit is 64 bits, meaning the set value of the CSEL bit is "011" and the enable column address is the internal address [11:3]. Row address must be set immediately after column address. The enable row address is the internal address [24:12], and the set value of the RSEL bit is "1100". Bank address also must be set immediately after row address. The enable bank address is the internal address [26:25] and "0101" is set to the BSEL bit.

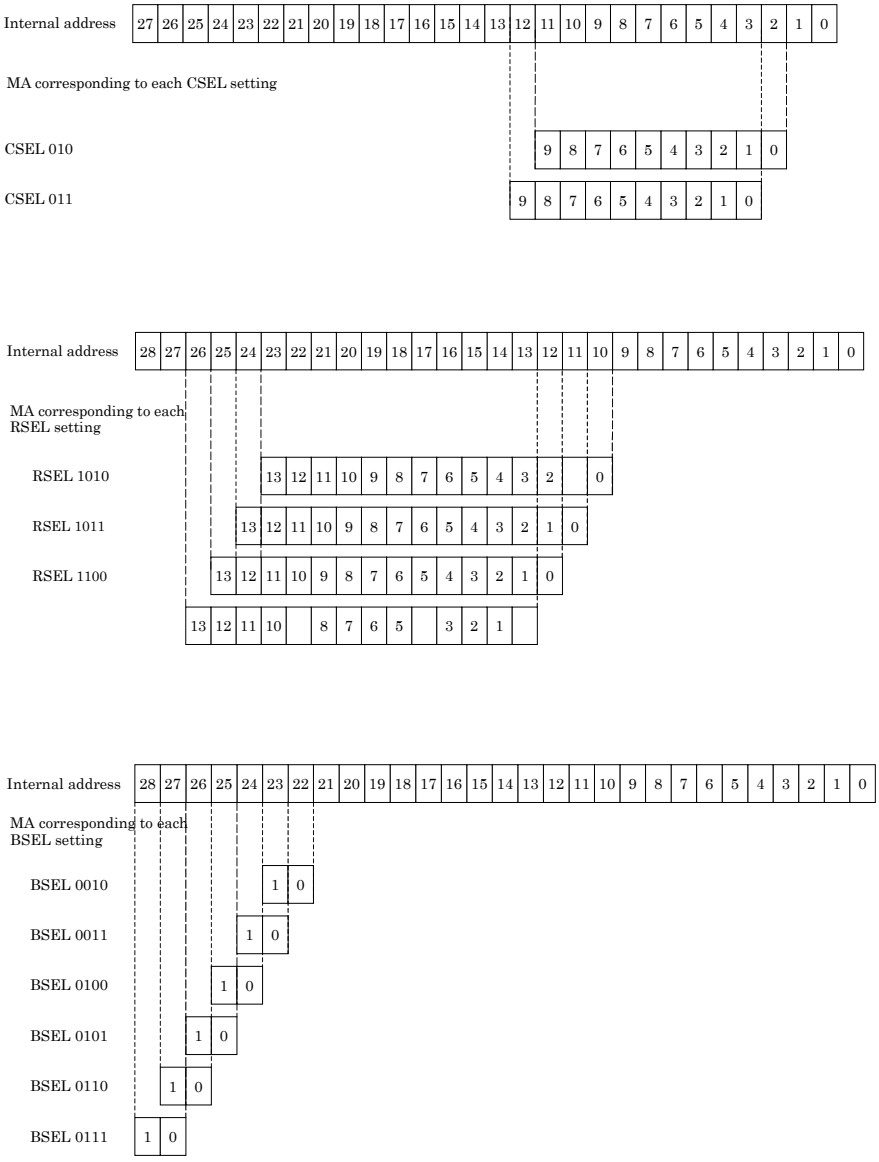


Fig. 3.2 DRAM CTRL ADD Register Setting and Internal Addresses

DRAM CTRL MODE

Set a value written to the DRAM MODE register to this register. The value of the bits 12 to 0 of this register is output as it is from MA12 to 0 when the mode register set (MRS) command is issued. In general, the mode shown below is assigned. Set correct values according to Section of "MODE register" in DRAM ***Specifications*** used. Note that when "1" is set to Bit 8 (that is, when DLL RESET is performed), the setting procedures for DRAM initial sequence differ from the case where the bit 8 is "0".

Register address	**Base Address** + 02_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved			OPM						CL			BT	BL		
R/W	R			RW						RW			RW	RW		
Initial value	0			0						0			0	0		

Bit 2-0 BL (Burst Length)

Sets a DRAM burst length. For SDR, only burst length 1 is supported; for DDR, only burst length 2 is supported.

000: Burst length 1 (for SDR)

001: Burst length 2 (for DDR)

Other settings: These are prohibited.

Bit 3 BT (Burst Type)

Sets a DRAM burst type. Only sequential is supported.

0: Sequential

Other settings: These are prohibited.

Bit 6-4 CL (CAS Latency)

Sets DRAM CAS latency. When using CAS Latency 2.5, change the setting of the DDRIF2 register.

010: CAS Latency 2

011: CAS Latency 3

110: CAS Latency 2.5

Other settings: These are prohibited.

Bit 12-7 OPM (Operating Mode)

Sets a DRAM operation mode.

000000: Normal operation

000010: Normal operation/DLL reset

Other settings: These are prohibited.

DRAM CTRL EMODE

Set a value written to the DRAM extended MODE register to this register. The value of the bits 12 to 0 of this register is output as it is from MA 12 to 0 when the extended mode register set (EMODE) command is issued. In general, the mode shown below is assigned. Set correct values according to Section of "EMODE register" in DRAM ***Specifications*** used.

Register address	**Base Address** + 04_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EME	Reserved		EOPM											DS	DL
R/W		RW		RW											RW	RW
Initial value	0	0		–											0	0

Bit 0 DL (DRAM DLL CONTROL)

0: Enables DLL embedded in DRAM.

1: Disables DLL embedded in DRAM.

Bit 1 DS (DRAM Drive strength)

0: DRAM pin output is normal.

1: DRAM pin output is weak.

Bit 12-2 EOPM (Extended Operating Mode)

00000000000: Normal

Others: Refer to Section of "EMODE register" in DRAM ***Specifications*** used.

Bit 15 EME (EMODE ENABLE)

When EME is not enabled, EMRS is not issued even when the status is EMODE. When the EMODE status is established with EME being disabled, a command is issued as an ordinary MRS, outputting the value of the EMODE register to MA12 to 0.

1: Enabled

0: Disabled

DRAM CTRL SET TIME1

Set access timing to DRAM to this register. Set appropriately based on the relationship between the specification of DRAM used and the internal frequency.

Register address	**Base Address** + 06_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	TRCD				TRAS			Reserved	TRP			TRC			
R/W	–	RW				RW			–	RW			RW			
Initial value	–	0				0			–	0			0			

Bit 3-0 TRC (RAS cycle time)
Sets the shortest DRAM RAS cycle time by the cycle.
0001: 1 cycle
0010: 2 cycles
0011: 3 cycles
0100: 4 cycles
0101: 5 cycles
0110: 6 cycles
0111: 7 cycles
1000: 8 cycles
1001: 9 cycles
1010: 10 cycles
1011: 11 cycles
Other settings: These are prohibited.

Bit 6-4 TRP (RAS Precharge time)
Sets the shortest DRAM RAS precharge time by the cycle.
001: 1 cycle
010: 2 cycles
011: 3 cycles
100: 4 cycles
Other settings: These are prohibited.

Bit 10-8 TRAS (RAS active time)
Sets the shortest DRAM RAS active time by the cycle.
001: 1 cycle
010: 2 cycles
011: 3 cycles
100: 4 cycles
101: 5 cycles
110: 6 cycles
111: 7 cycles
Other settings: These are prohibited.

Bit 14-12 TRCD (RAS to CAS delay time)
Sets the shortest time from the issue of DRAM RAS to the issue of DRAM CAS, by the cycle.
001: 1 cycle
010: 2 cycles
011: 3 cycles
100: 4 cycles
Other settings: These are prohibited.

DRAM CTRL SET TIME2

Set access timing to DRAM to this register. Set appropriately based on the relationship between the specification of DRAM used and the internal frequency.

Register address	**Base Address** + 08_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved		TWTR		TRFC				Reserved		TRRD		Reserved		TWR	
R/W	RW										RW		–		RW	
Initial value	–										0		–		0	

Bit 1-0 TWR (Write recovery time)

Sets the shortest DRAM write recovery time by the cycle.

00: Setting prohibited

01: 3 cycles

10: 4 cycles

11: 5 cycles

Bit 5-4 TRRD (RAS to RAS bank active delay time)

Sets the shortest interval at which the active command can be issued when making RAS of different banks active continuously, by the cycle.

00: Setting prohibited

01: 1 cycle

10: 2 cycles

11: 3 cycles

Bit 11-8 TRFC (Auto REFRESH command period)

Sets the shortest time from the issue of a refresh command to the issue of the next refresh command, by the cycle.

0001: 1 cycle

0010: 2 cycles

0011: 3 cycles

0100: 4 cycles

0101: 5 cycles

0110: 6 cycles

0111: 7 cycles

1000: 8 cycles

1001: 9 cycles

1010: 10 cycles

1011: 11 cycles

Other settings: These are prohibited.

Bit 13-12 TWTR (Write to Read command delay time)

Sets the shortest time from the issue of a write command to the issue of the read command, by the cycle.

00: Setting prohibited

01: 4 cycles

10: 5 cycles

11: 6 cycles

DRAM CTRL REFRESH

Sets auto refresh to DRAM. Set according to the specifications of DRAM used. The issuing interval of refresh command fluctuates around a set value. Set the value by the cycle.

Register address	**Base Address** + $0A_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								REF_CNT							
R/W	RW								RW							
Initial value	0								0							

Bit 7-0 REF_CNT (Refresh count)
Sets an issuing interval of auto refresh by the 16 cycles.
00000000: Continues issuing refresh.
00000001 to 11111110: Interval can be set within the range of 16 cycles to 4064 cycles.
11111111: Does not issue refresh

Example 1: 00000001: Issues the auto refresh command at interval of about 16 cycles.
Example 2: 00000010: Issues the auto refresh command at interval of about 32 cycles.

DRAM CTRL STATE

Sets the status of DRAM CTRL and others.

Register address	**Base Address** + $0C_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved					INTSTT	INTMSK	INTCL	Reserved				DCTRL_STATES			
R/W	R					RW	RW	RW	RW				RW			
Initial value	Undefined					0	0	0	0				Undefined			

Bit 3-0 DCTRL_STATES (STATES)

Reads or writes the status of DRAM CTRL.

0000: START Initial status. DRAM controller continues to issue NOP.

0010: MOD Status that sets the MODE register to EDRAM. After setting a value to the DRAM CTRL MODE register, set this value (MOD) and use DRAM.
Issues MRS command.

0011: EMODE Status that sets the extended MODE register to DRAM. Set a value appropriate to DDR SDRAM to the DRAM CTRL EMODE register beforehand. Issues EMRS command.

0100: HOME Home status. This status is set from the outside or transits to a status next to MODE. Normally, the DRAM controller remains in this status.

Others: Setting is disabled.

Bit 8 INTCL (INT CLEAR)

Clears interrupt.

0: Performs nothing.

1: Sets INTSTT to "0".

Bit 9 INTMSK (INT MASK)

Masks interrupt. Initial value "0" is masked.

0: Sets INT to "0" irrespective of INTSTT.

1: Outputs the INTSTT value to INT.

Bit 10 INTSTT (INT STATUS)

Interrupt state. Initial value "0" is masked. When INTCL is set to "1", this bit is set to "0". This bit is set to "1" when a request to the DRAM controller is issued when the status is not HOME.

0: Normal status

1: Interrupt status

DRAM CTRL Reserve 0

Register address	**Base Address** + $0E_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0020															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL FIFO DEPTH

Register address	**Base Address** + 10_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FIFOdepth		Reserved													
R/W	RW		RW													
Initial value	000F															

Bit 13-0 Reserved
Write is prohibited.
Bit 15+14 FIFO Depth (in commands)
00 - 8
01 - 4
10 - 16
11 - 48

DRAM CTRL Reserve 2

Register address	**Base Address** + 12_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL DDRIF1

Register address	**Base Address** + 14_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CKN2		CKP2		CKN1		CKP1		Reserved							
R/W	RW		RW		RW		RW		RW							
Initial value	01		10		01		10		46							

Bit 7-0 Reserved
Write is prohibited.

Bit 9-8 CKP1
Used to set MCK0.
00: Stops clock.
10: Outputs clock.

Bit 11-10 CKN1
Used to set XMCK0.
00: Stops clock.
01: Outputs clock.

Bit 13-12 CKP2
Controls MCK1.
00: Stops clock.
10: Outputs clock.

Bit 15-14 CKN2
Controls XMCK1.
00: Stops clock.
10: Outputs clock.

DRAM CTRL DDRIF2

Register address	**Base Address** + 16_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DDRIF2															
R/W	RW															
Initial value	0055															

Bit 15-0 DDRIF2
This field shifts the data receive timing by half a cycle. Set this field when using CAS Latency 2.5.
0055:DRAM: Set this when using CAS Latency 2 or CAS Latency 3.
55aa:DRAM: Set this when using CAS Latency 2.5.

DRAM CTRL Reserve 4

Register address	**Base Address** + 18_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL Reserve 5

Register address	**Base Address** + $1A_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0000															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL Reserve 6

Register address	**Base Address** + $1C_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0002															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL Reserve 7

Register address	**Base Address** + $1E_H$															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0db6															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL Reserve 8

Register address	**Base Address** + 20_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW															
Initial value	0db6															

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL Reserve 9

Register address	**Base Address** + 22_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															
R/W	RW												R			
Initial value	0												Undefined			

Bit 15-0 Reserved
Write is prohibited.

DRAM CTRL IO CONT0

This register controls an I/O. Always set this register before executing DRAM initial sequence. Set so that a value of this register matches the mode used (LVCMOS mode or SSTL2 mode).

Register address	**Base Address** + 24_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IO_CTRL0											
R/W	RW															
Initial value	0															

Bit 11-0 IO_CTRL0
Driving capability 1
SSTL2: 'h444
LVCMOS: 'h000
Driving capability 2 (recommended)
SSTL2: 'h555
LVCMOS: 'h111
Others: Setting inhibited
*When using a driving capability other than 2, contact our staff in charge.

DRAM CTRL IO CONT1

This register controls an I/O. Always set this register before executing DRAM initial sequence. Set so that a value of this register matches the mode used (LVCMOS mode or SSTL2 mode).

Register address	**Base Address** + 26_H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IO_CTRL1											
R/W	RW															
Initial value	0															

Bit11-0 IO_CTRL1

Driving capability 1

SSTL2: 'h444

LVCMOS: 'h000

Driving capability 2 (recommended)

SSTL2: 'h555

LVCMOS: 'h111

Others: Setting inhibited

*When using a driving capability other than 2, contact our staff in charge.

4 I²C Interface

4.1 Overview

This interface is a serial interface that supports the Inter IC Bus, and operates as a master/slave device on the I²C bus.

Note: Carmine does not support slave mode.

4.2 Features

This interface has the following features:

Master send/receive

Arbitration

Clock synchronization

Slave address detection

General call address detection

Transfer direction detection

Repetitive generation and detection of start condition

Bus error detection

Standard mode (max. 100 Kbps) / fast mode (max. 400 Kbps)

4.3 Block Diagram

Fig. 4.1 shows the block diagram of this module.

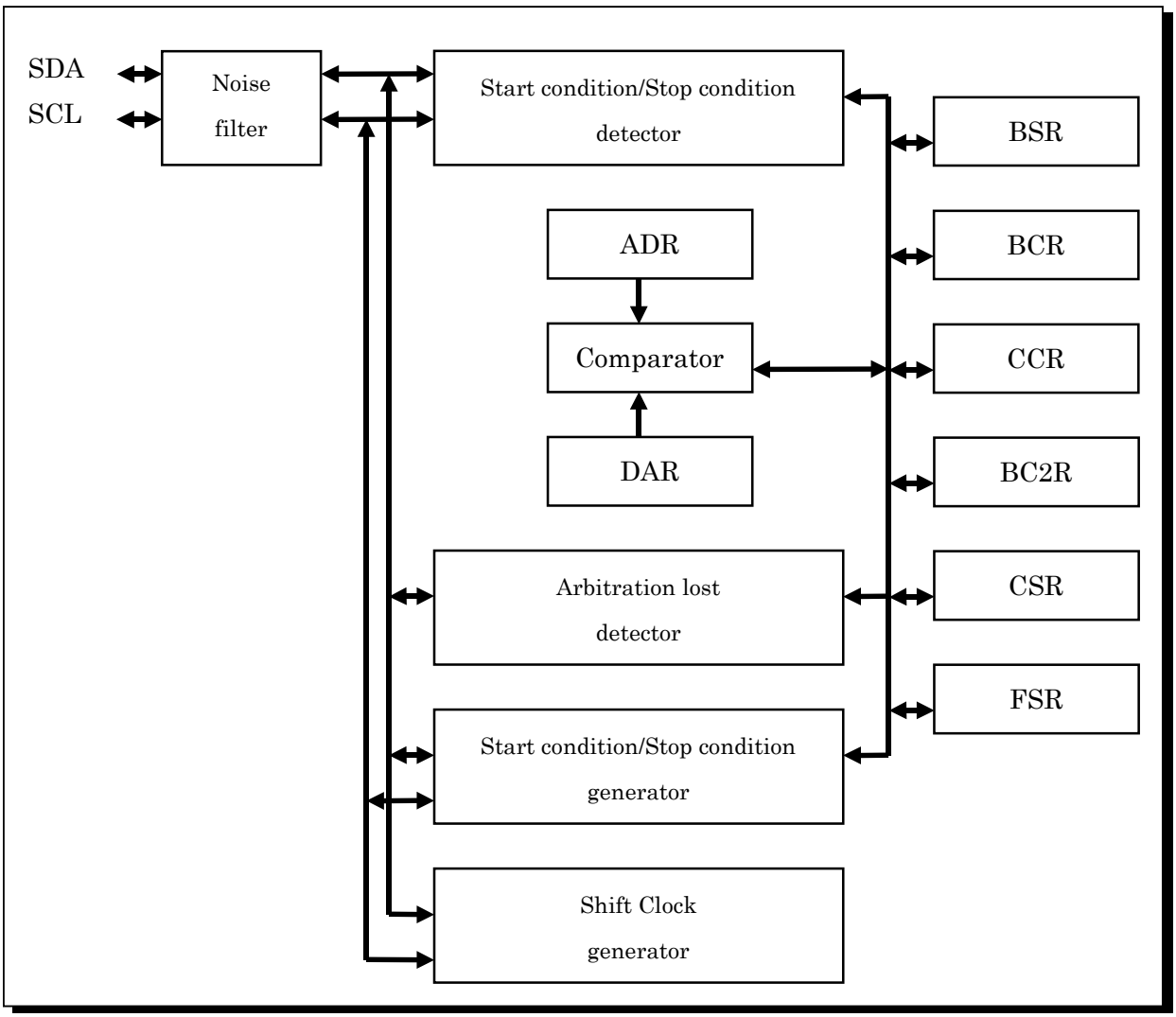


Fig. 4.1 Block Diagram

4.4 Description of Block Function

- Start condition/Stop condition detector

  Detects start condition and stop condition from the status changes of SDA and SCL.

- Start condition/Stop condition generator

  Changes the statuses of SDA and SCL to generate start condition and stop condition.

- Arbitration Lost detector

  When sending data, compares data output to the SDA line signal and data input from the SDA line signal, to check whether they match. If they do not match, the detector generates Arbitration Lost.

- Shift clock generator

  Counts the generation timing of serial data transfer clock and controls output of SCL clock signal according to the clock control register setting.

- Comparator

  Compares whether the received address and the own address specified for the address register are the same, or checks whether the received address is a global address.

- ADR

  A 7-bit register to specify a slave address.

- DAR

  An 8-bit register used for serial data transfer.

- BSR

  An 8-bit register that shows the statuses of the I²C bus, and has the following functions:
  - Detects repeated start conditions
  - Detects Arbitration Lost
  - Stores acknowledge bit
  - Detects data transfer direction
  - Detects addressing
  - Detects general call address
  - Detects first byte

- BCR

  An 8-bit register that controls the I²C bus and interrupt, and has the following functions:
  - Interrupt request / Interrupt permission
  - Generation of start condition
  - Master/slave selection
  - Permission of acknowledge generation

- CCR

  A 7-bit register that sets the clock frequency for serial data transfer.
  - Permission of operation
  - Setting of serial clock frequency
  - Standard mode and fast mode

- Noise filter

  The noise filter consists of a three-stage shift register circuit. When all three values of SCL/SDA input signal sampled continuously are “1s”, filter output is “1”; when all three values of SCL/SDA input signal sampled continuously are “0s”, filter output is “0”; and when all the 3 values are not “1” or “0”, the state of the signal 1 clock before is held.

- BC2R

  Checks the status of the line after the signal passes through the noise filter and to forcibly drive the line “low”.

- CSR

  Expands the CS bit in the CCR register.

- FSR

  Specifies the frequency range of the bus clock used.

4.5 Operation Description

The I²C bus performs communication using two bidirectional bus lines: One serial data line (SDA) and one serial clock line (SCL). This module has the SDA input (SDAI) and SDA output (SDAO) that are for the SDA line, and is connected to the SDA line via an open drain I/O cell. This module also has the SCL input (SCLI) and SCL output (SCLO) that are for the SCL line, and is connected to the SCL line via the open drain I/O cell. This module is connected to the SDA line and SCL line by wired logic.

4.5.1 Start Condition

When "1" is written to the MSS bit with the bus opened (BB=0), this module enters master mode, and at the same time, generates a start condition. In master mode, even when the bus is in use (BB=1), writing "1" to the SCC bit generates a start condition again.

A start condition is generated under the following two conditions:

(i) Writing "1" to the MSS bit with the bus not used (MSS=0 & BB=0 & INT=0 & AL=0).

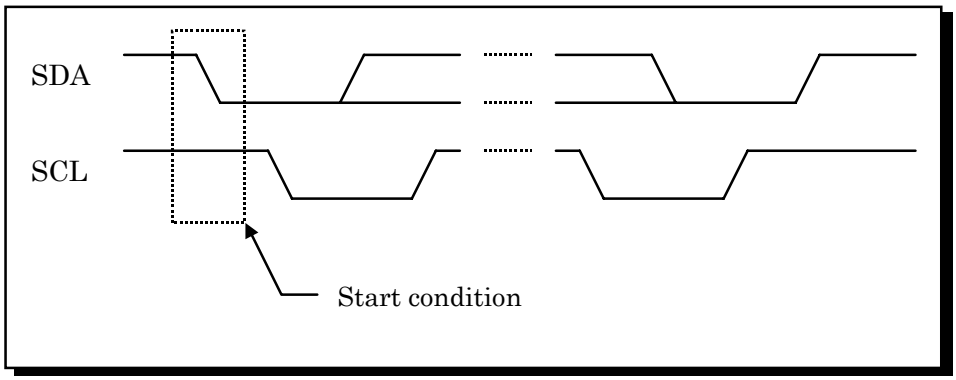
(ii) Writing "1" to the SCC bit with an interrupt occurred in bus master mode (MSS=1 & BB=1 & INT=1 & AL=0).

When "1" is written to the MSS bit during the idle state, the AL bit is set to "1".

In cases other than above (i) and (ii), writing "1" to the MSS bit and SCC bit respectively is ignored.

Start condition on the I²C bus

Start condition means that the SDA line changes from "1" to "0" with the SCL line being "1".



4.5.2 Stop Condition

In master mode (MSS=1), writing"0" to the MSS bit generates a stop condition, which causes the module to become a slave.

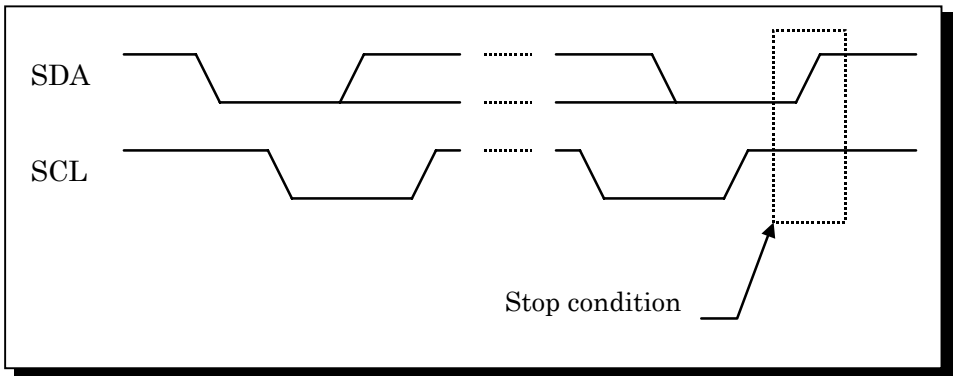
A stop condition is generated under the following condition:

(i) Writing "0" to the MSS bit with an interrupt occurred in bus master mode (MSS=1 & BB=1 & INT=1 & AL=0).

In a state other than above, writing "0" to the MSS bit is ignored.

Stop condition on the I²C bus

Stop condition means that the SDA line changes from "0" to "1" with the SCL line being "1".

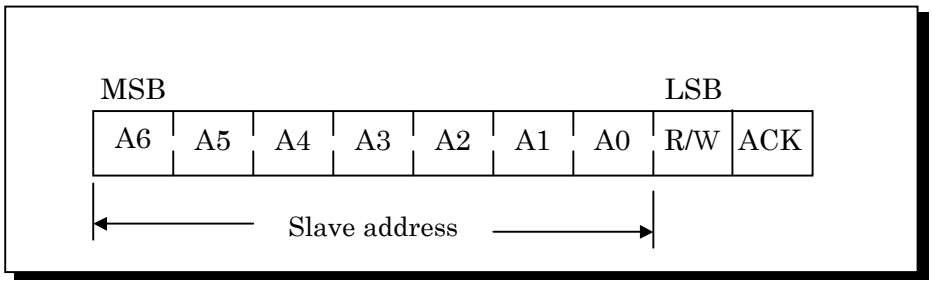


4.5.3 Addressing

In master mode, after a start condition has been generated, BB and TRX are set to 1 respectively, the data of the DAR register is output from the MSB. When an acknowledge is received from the slave after address data is sent, bit 0 of send data (bit 0 of the sent DAR register) is inverted and stored in the TRX bit.

Transfer format of slave address

The transfer format of slave address is shown below.



Slave address map

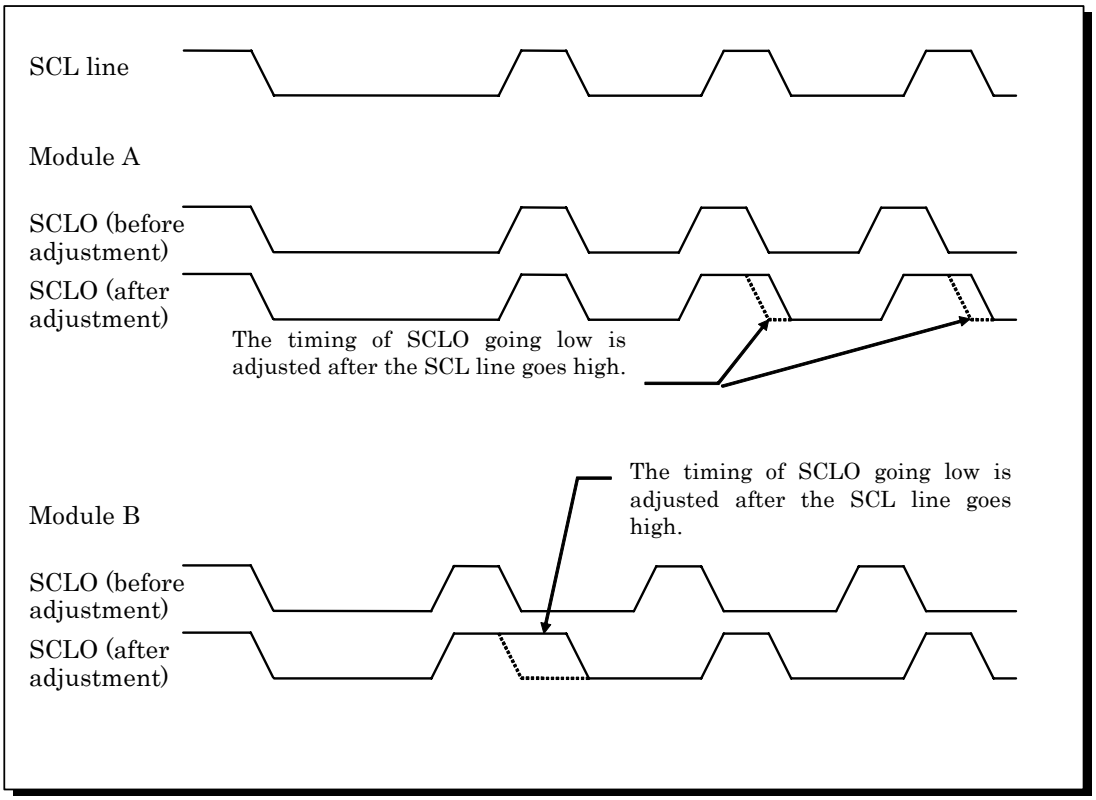
The slave address map is shown below.

Slave address	R/W	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Reserved
0000 011	X	Reserved
0000 1XX	X	
0001 XXX to 1110 XXX	X	Available slave addresses
1111 0XX	X	10-bit slave address*
1111 1XX	X	Reserved

*1: This module does not support 10-bit slave address.

4.5.4 Adjustment of SCL Synchronization

When multiple I²C devices become master devices almost at the same time and drives the SCL line, each I²C device senses the status of the SCL line and automatically adjusts the driving timing of the SCL line according to the timing of the slower device.



4.5.5 Arbitration

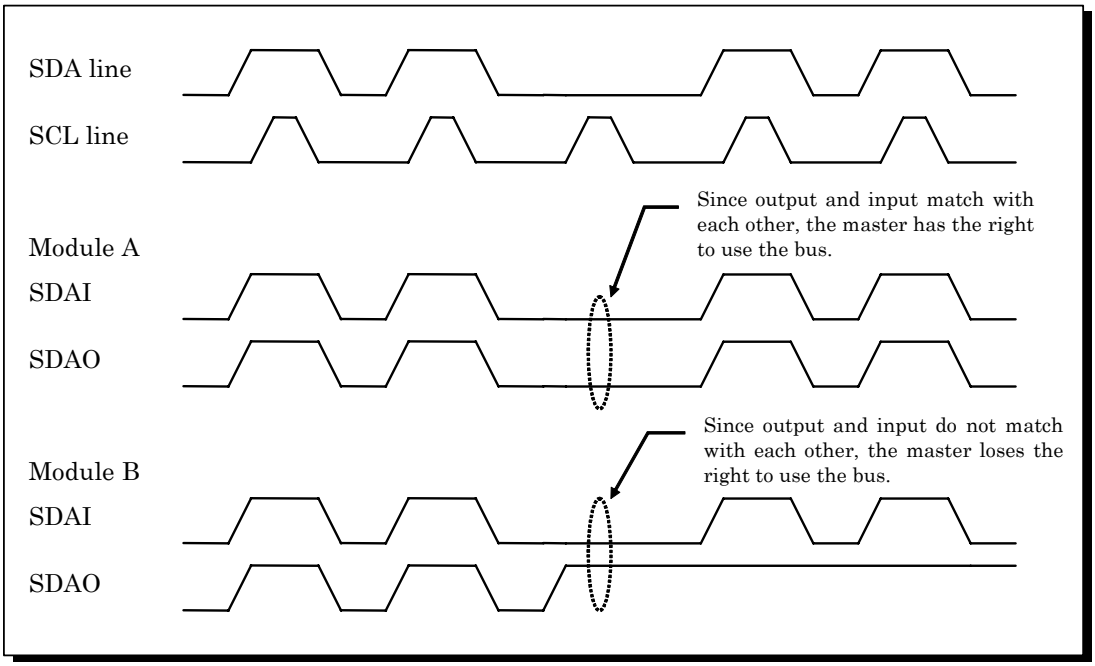
Arbitration occurs when one master and another master are sending data simultaneously. When a master's send data is "1" and data on the SDA line is "0", the master assumes that it has lost the arbitration and then sets 1 to AL.

Also, when the master tries to generate a start condition when another master is using the bus, the former master assumes that it has lost the arbitration and then sets 1 to AL.

Also, even when one master checks that the bus is not used and writes 1 to MSS, one master assumes that it has lost the arbitration and then sets 1 to AL when one master detects a start condition generated by another master before one master generates a start condition.

When "1" is set to the AL bit, MSS and TRX are set to "0s" respectively, which causes slave receive mode.

The master stops driving the SDA line at the point when the master loses the arbitration (the right to use the bus). However, the master does not stop driving the SCL line until 1-byte transfer ends and the interrupt is cleared.

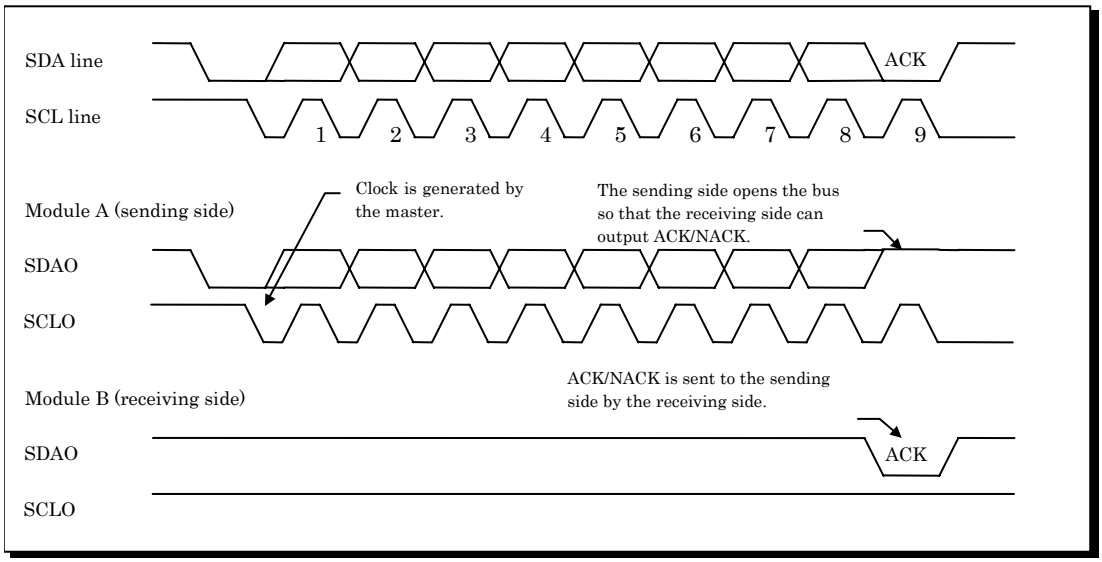


4.5.6 Acknowledge/Negative Acknowledge

The value of the 9th bit shows Acknowledge (ACK)/negative acknowledge (NACK). When the value of the 9th bit is "0", it shows ACK; when the value of the 9th bit is "1", it shows NACK.

ACK/NACK is sent to the sending side by the receiving side. When receiving data, the ACK/NACK is stored in the LRB bit.

When the slave does not receive ACK (or it receives NACK) from the master (the receiving side) when the slave sends data, TRX is set to 0 and enters slave receive mode. By this, the master can generate a stop condition when the slave opens the SCL line.



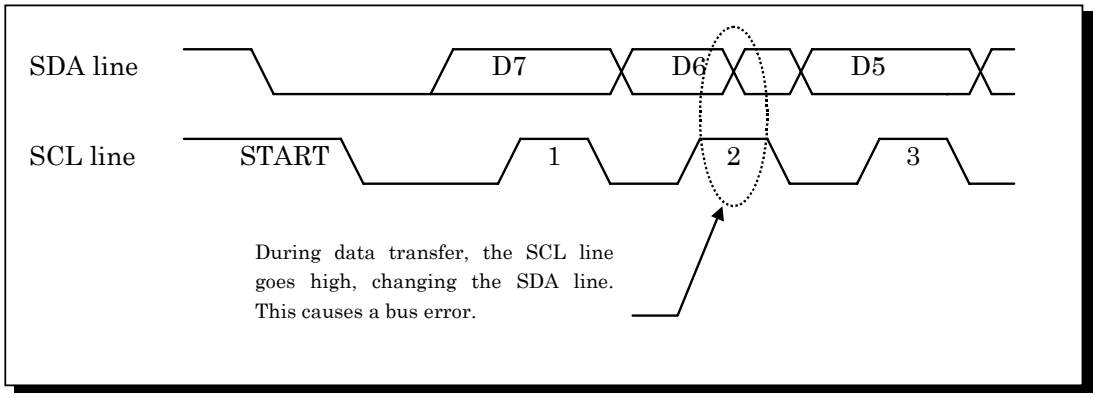
4.5.7 Bus Error

When the following conditions are met, this module assumes that a bus error has occurred, and enters stopped state.

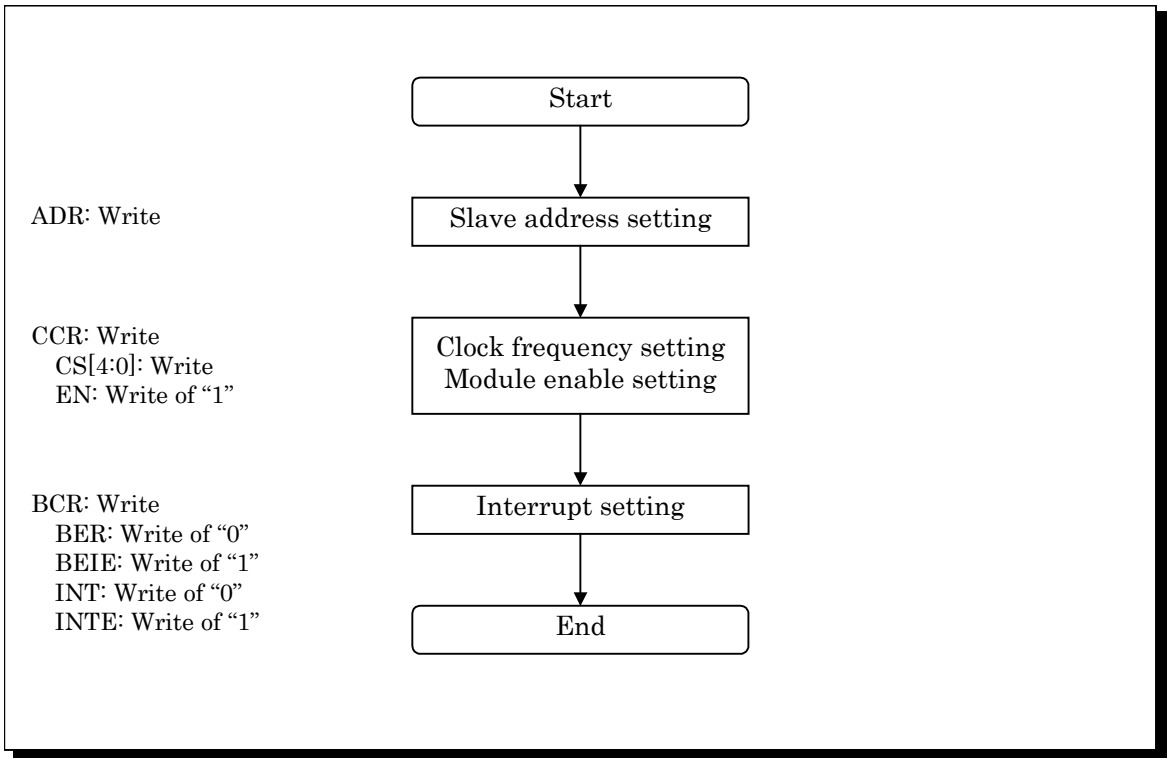
(1) Violation of the regulations on the I²C bus is detected during data transfer (including ACK bit).

(2) Stop condition is detected in master mode.

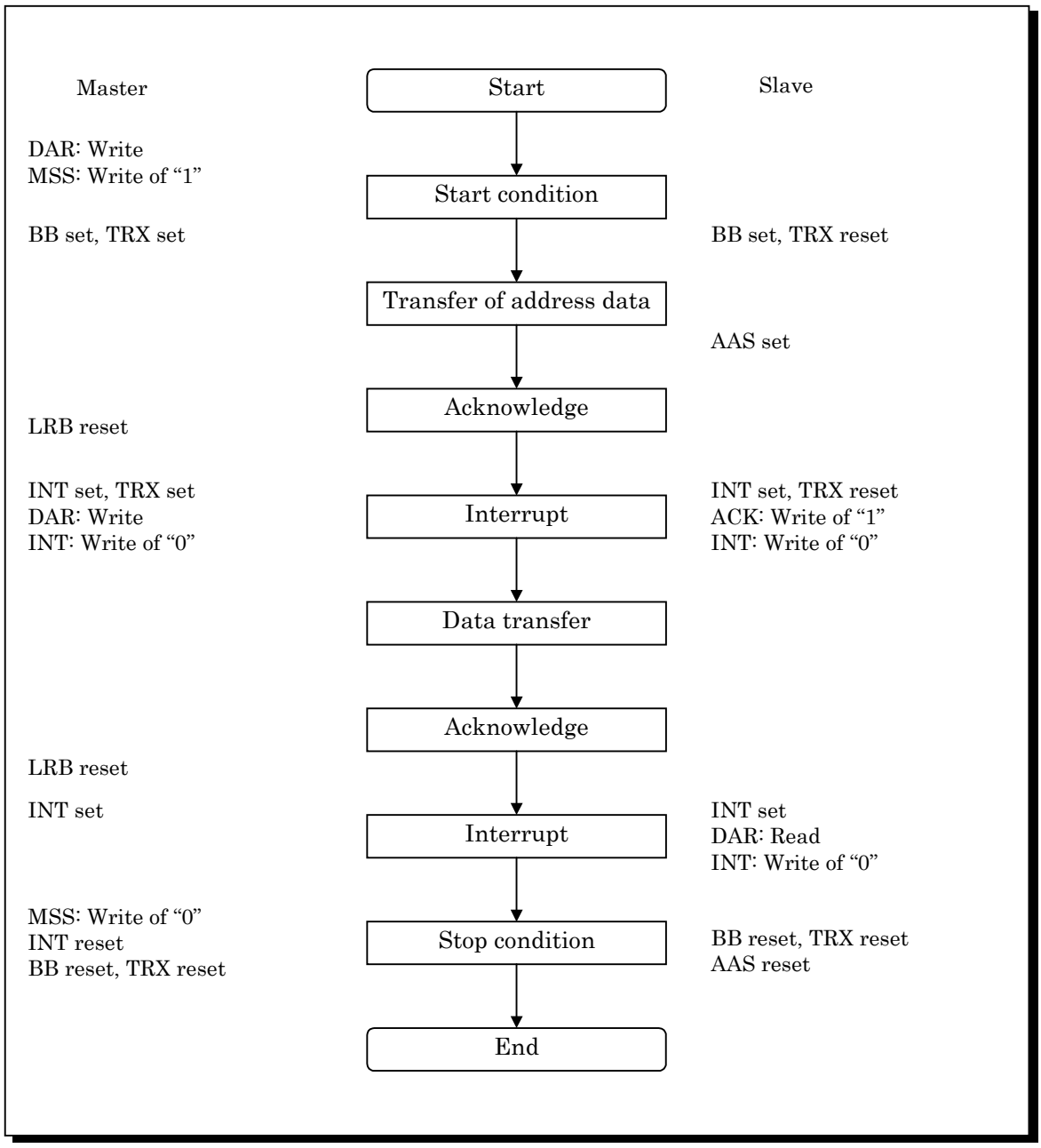
(3) Violation of the regulations on the I²C bus in bus idle mode is detected.



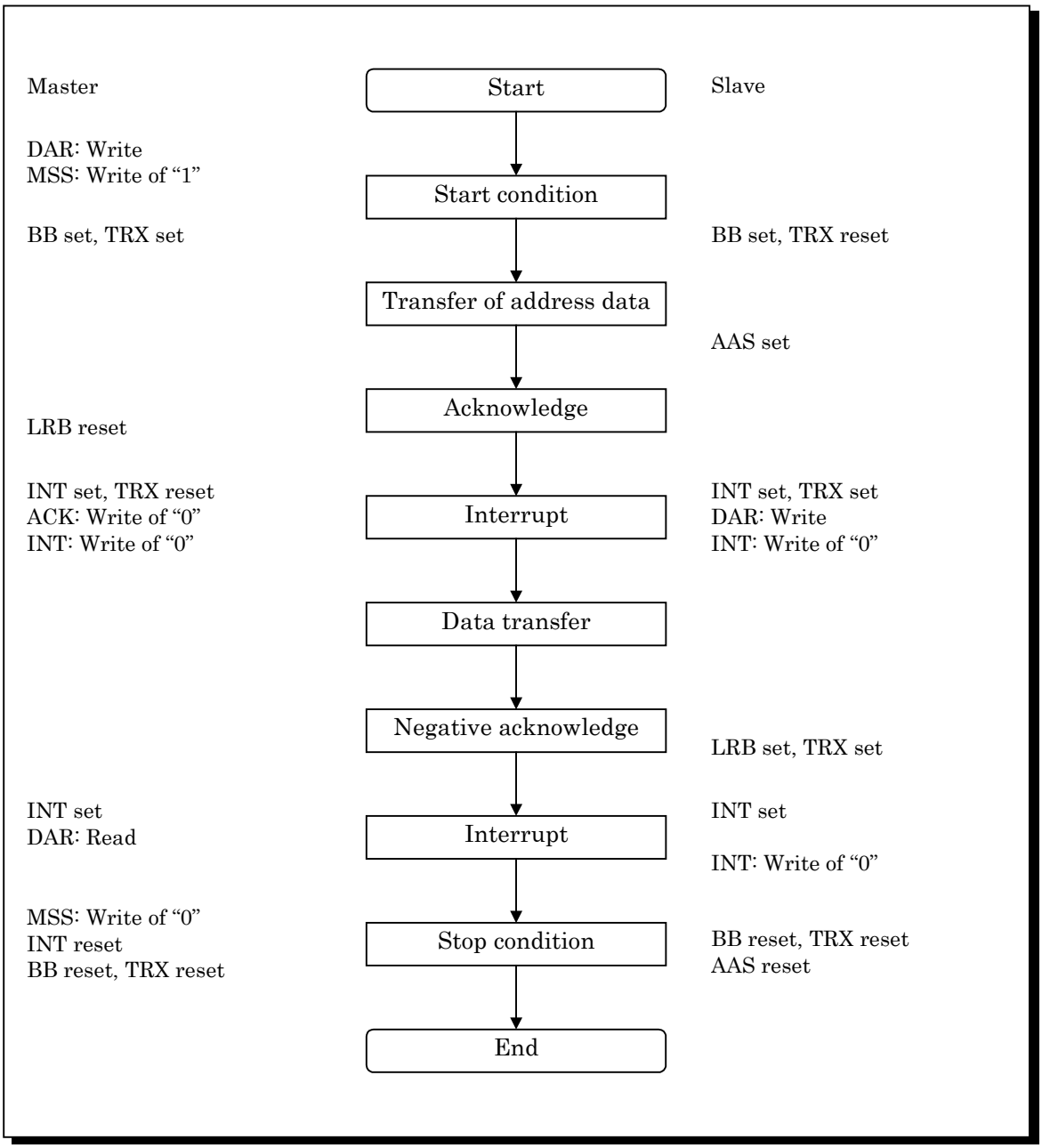
4.5.8 Initialization



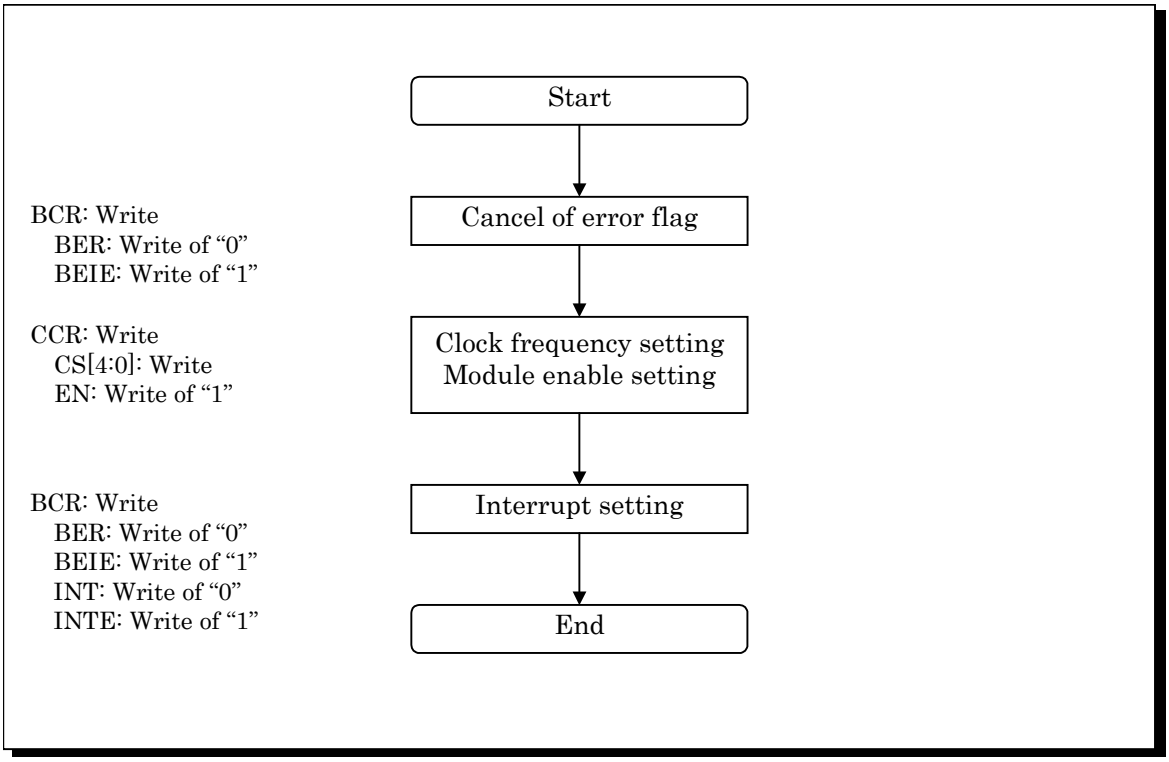
4.5.9 1-byte Transfer from Master to Slave



4.5.10 1-byte Transfer from Slave to Master



4.5.11 Return after Bus Error



4.5.12 Interrupt Processing and Wait Request to the Master Device

While the INT flag of the BCR register is “H” (while CPU is processing the interrupt generated by the module), the SCLO output is set to “L”. While the slave sets the SCL line to “L”, the master cannot generate the next transfer clock, and so the slave requests the master to wait.

4.6 Cautions

4.6.1 10-bit Slave Address

This module does not support 10-bit slave address. Do not specify slave addresses 78_H-$7B_H$ for this module. If specified incorrectly, ACK is returned when receiving 1byte, but normal transfer cannot be performed.

4.6.2 Conflict among SCC, MSS and INT Bits

When write is performed to the SCC, MSS and INT bits simultaneously, a conflict occurs with next byte transfer, generation of start condition, and generation of stop condition. Priority at this time is as follows:

(1) Priority between next byte transfer and generation of stop condition

When "0" is written to the INT bit and "0" is written to the MSS bit, writing to the MSS bit takes precedence over the writing to the INT bit, causing a stop condition.

(2) Priority between next byte transfer and generation of start condition

When "0" is written to the INT bit and "1" is written to the SCC bit, the writing to the SCC bit takes precedence over the writing to the INT, causing a start condition.

(3) Priority between generations of start condition and stop condition

It is prohibited to simultaneously write "1" to the SCC bit and "0" to the MSS bit.

4.6.3 Setting of Serial Transfer Clock

When the rising delay of the SCL pin is large or when the slave device expands the clock, the frequency may be smaller than the set value (the calculated value) due to the overhead.

4.6.4 Restrictions on Multimaster

When using this module as a multimaster, the following usage is prohibited: this module and another master sends a global call address simultaneously, and arbitration is lost in this module at the second byte and later.

This restriction does not apply to the following:

- To use this module in a single-master environment.
- To use this module in a multimaster environment. However, in this case, this module does not use the sending of a general call address.
- To use this module in a multimaster environment. However, in this case, but module other than this module does not use the sending of a general call address.
- To use this module in a multimaster environment and another master and this module send a global call address simultaneously. However, in this case, arbitration is not lost in this module at the second byte and later*.

  *: Arbitration is lost in the larger size of send data. Therefore, data value in the second byte and later must always be smaller than that in other masters.

4.7 Register Configuration

4.7.1 Register List

Offset Address	Register Name	Description
0x0050_0000	BSR	Bus status register
0x0050_0004	BCR	Bus control register
0x0050_0008	CCR	Clock control register
0x0050_000C	ADR	Address register
0x0050_0010	DAR	Data register
0x0050_0014	CSR	CS extension register
0x0050_0018	FSR	FS register
0x0050_001C	BC2R	Bus control 2 register

4.7.2 Register Details

Base Address = 0x0050_0000

Bus status register (BSR)

Address	Base Address+0000							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT
Initial	L	L	L	L	L	L	L	L

All bits of this register are cleared while the EN bit of CCR is “0”.

[Bit7: BB (bus busy)]

Indicates the state of the I²C bus.

BB	State
0	Detected a stop condition.
1	Detected a start condition (the bus is in use).

[Bit6: RSC (Repeated Start Condition)]

Detects repeated start conditions.

RSC	State
0	Does not detect repeated start conditions.
1	Detected a start condition again when the bus was in use.

When "0" is written to the INT bit and I²C device is not recognized in slave mode, this bit is cleared by detecting a start condition or a stop condition with the bus being stopped.

[Bit5: AL (Arbitration Lost)]

Detects Arbitration Lost.

AL	State
0	Arbitration Lost is not detected.
1	Arbitration Lost is detected when the master is sending data, or “1” is written to the MSS bit when another system is using the bus.

Writing “0” to the INT bit clears this bit.

[Restriction]

The following is prohibited:

To use this module in a multimaster environment and another master and this module send a global call address simultaneously and, arbitration is lost in this module at the second byte and later.

[Bit4: LRB (LAST Recieved Bit)]

Stores the 9th bit of data indicating ACK/ NACK.

LRB	State
0	Detected ACK.
1	Detected NACK.

This bit is cleared when a start condition or stop condition is detected.

[Bit3: TRX (Transfer/Recieve)]

Indicates the send/receive status of data transfer.

TRX	State
0	Send status
1	Receive status

[Bit2: AAS (Address As Slave)]

Detects addressing.

AAS	State
0	No addressing is performed in slave mode.
1	Addressing is performed in slave mode.

This bit is cleared when a start condition or stop condition is detected.

[Bit1: GCA (General Call Address)]

Detects a general call address (00_H).

GCA	State
0	Does not receive general call address in slave mode.
1	Received general call address in slave mode.

This bit is cleared when a start condition or stop condition is detected.

[Bit0: FBT (First Byte Transfer)]

Detects the 1st byte.

FBT	State
0	The receive data is other than the 1st byte.
1	The receive data is the 1st byte (address data).

Even when this bit is set to "1" by detecting a start condition, this bit is cleared when "0" is written to the INT bit or no addressing is performed in slave mode.

Bus control register (BCR)

Address	Base Address+0004							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT
Initial	L	L	L	L	L	L	L	L

Bits of this register other than bit7 and bit6 are cleared while the EN bit of the CCR register is "0".

[Bit7: BER (Bus ERror)]

This bit is a bus error interrupt request flag bit.
At write time

BER	State
0	Clears bus error interrupt request flag.
1	Not applicable

At read time

BER	State
0	Bus error is not detected.
1	Detected an invalid start or stop condition during data transfer.

When this bit is set, the EN bit of the CCR register is cleared, this module is stopped, and data transfer is suspended.

[Bit6: BEIE (Bus Error Interrupt Enable)]

This bit is a bus error interrupt enable bit.
At read/write time

BEIE	State
0	Bus error interrupt is disabled.
1	Bus error interrupt is enabled.

When this bit is "1", an interrupt is generated if the BER bit is "1".

[Bit5: SCC (Start Condition Continue)]

Generates a start condition.
At write time

SCC	State
0	Not applicable
1	Generates start condition again when the master transfers data.

This bit is automatically cleared after set to "1".

[Bit4: MSS (Master Slave Select)]

Selects master or slave.
At write time

MSS	State
0	Generates a start condition to enter slave mode after data transfer ends.
1	Enters master mode to generate a start condition and starts data transfer.

This bit is cleared when arbitration is lost during data sending by the master, entering slave mode.

[Restriction]

The following is prohibited:

To use this module in a multimaster environment, another master and this module send a general call address simultaneously, and arbitration is lost in this module at the second byte and later.

[Bit3: ACK (ACKnowledge)

Enables generation of ACK when data is received.
At read/write time

ACK	State
0	Does not generate ACK.
1	Generates ACK.

This bit is disabled when address data is received in slave mode.

[Bit2: GCAA (General Call Address Acknowledge)]

Enables generation of ACK when a general call address is received.

At read/write time

GCAA	State
0	Does not generate ACK..
1	Generates ACK.

[Bit1: INTE (INTerrupt Enable)]

Enables interrupt.
At read/write time

INTE	State
0	Interrupt is disabled.
1	Interrupt is enabled.

When this bit is “1”, an interrupt is generated if the INT bit is “1”.

[Bit0: INT (INTerrupt)]

This bit is a transfer end interrupt request flag bit.

At write time

INT	State
0	The transfer end interrupt request flag is cleared.
1	Not applicable

At read time

INT	State
0	Transfer does not end.
1	"1" is set when this module meets one of the following conditions when 1 byte transfer including the ACK bit ends: - This module is a bus master. - This module is an addressed slave. - This module receives a general call address (this is only when GCAA="1"). - Arbitration is lost in this module (this is only when the bus is acquired). - This module tries to generate a start condition when another system is using the bus.

When this bit is "1", the SCL line keeps "Low". Writing "0" to this bit clears this bit, opening the SCL line to transfer the next byte. In addition, when a start condition or stop condition is generated in master mode, this bit is reset to "0".

Conflict among SCC, MSS and INT bits

When write is performed to the SCC, MSS and INT bits simultaneously, a conflict occurs with next byte transfer, generation of start condition, and generation of stop condition. Priority at this time is as follows:

(1) Priority between next byte transfer and generation of stop condition

When "0" is written to the INT bit and "0" is written to the MSS bit, writing to the MSS bit takes precedence over the writing to the INT bit, causing a stop condition.

(2) Priority between next byte transfer and generation of start condition

When "0" is written to the INT bit and "1" is written to the SCC bit, the writing to the SCC bit takes precedence over the writing to the INT, causing a start condition.

(3) Priority between generations of a start condition and a stop condition

It is prohibited to simultaneously write "1" to the SCC bit and "0" to the MSS bit.

Clock control register (CCR)

Address	Base Address+0008							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	-	HSM	EN	CS4	CS3	CS2	CS1	CS0
Initial	-	L	L	-	-	-	-	-

[Bit7: unused]

At read time, this bit is always "1".

[Bit6: HSM (High Speed Mode)]

Sets standard/fast mode.
At read/write time

HSM	Status
0	Standard mode
1	Fast mode

[Bit5: EN (ENable)]

Enables operation.
At read/write time

EN	Status
0	Operation is disabled.
1	Operation is enabled.

When this bit is "0", each bit of the BSR and BCR registers (except the BER and BEIE bits) are cleared. Writing "1" to the BER bit clears this EN bit.

[Bit4-0: CS4 to 0 (Clock Period Select 4 to 0)]

Sets the frequency of serial transfer clock.

The frequency of the serial transfer clock is determined based on the value set here and the value set using the CSR register. For details, see the description of the CSR register.

Address register (ADR)

Address	Base Address+000C							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	-	A6	A5	A4	A3	A2	A1	A0
Initial	-	-	-	-	-	-	-	-

[Bit7: unused]

At read time, this bit is always "1".

[Bit6 to 0: A6 to 0 (Address 6 to 0)]

Stores a slave address.

In slave mode, this register is compared with the DAR register after receiving address data. When they match, this register sends ACK to the master.

Data register (DAR)

Address	Base Address+0010							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial	-	-	-	-	-	-	-	-

[Bit7 to 0: D7 to 0(Data 7 to 0)]

Stores serial data.

This is a data register used to perform a serial transfer, and its data is transferred from MSB. When it receives data (TRX=0), data output is "1".

The write side of this register has a double buffer; when the bus is in use (BB=1), write data is loaded to the serial transfer register at the time of transferring each byte. At read time, it reads the serial transfer register directly, and so receive data is valid only when "1" is set to the INT bit.

CS extension register (CSR)

Address	Base Address+0014							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TST1	TST0	CS10	CS9	CS8	CS7	CS6	CS5
Initial	L	L	L	L	L	L	L	L

[Bit7, 6: TST1, TST0]

TST1,TST0	State
00	Normal mode
Other than 00	Test mode

[Bit5 to 0: CS10 to 5(Clock Period Select 10 to 5)]

These bits set the frequency of the serial transfer clock (fscl). These bits extend CS4 to 0 in the CCR register. The initial value of CS10 to 5 is "000000", but when it is set to a value other than "000000", this setting is enabled.

CS10 to 5	Status
000000	Uses only CS4 to 0.
Other than "000000"	Uses CS10 to 0.

Set fscl not to exceed the following values::

Standard mode: 100 KHz

Fast mode: 400 KHz

Use the bus clock in the frequency range shown below. Usually, the frequency of the bus clock is 1/4 that of the system clock. (For example, when the system clock is set to 266 MHz, the bus clock is 66 MHz.) When using a bus clock outside this frequency range, operation is not guaranteed.

Bus clock ϕ: 14 MHz to 66 MHz

Standard mode

$$fscl = \frac{\phi}{(2 \times m) + 2}$$

ϕ: Bus clock
m: CS10 to 0 value + 1

Fast mode

$$fscl = \frac{\phi}{int(1.5 \times m) + 2}$$

ϕ: Bus clock
m: CS10 to 0 value + 1

Regarding the value of int( ), round off digits after the decimal point.

[Note]

The "+ 2" cycles is the minimum overhead to check that the output level of the SCL pin changed.

When the rising delay of the SCL pin is large or when the slave device expands the clock, the value may be greater than this "+2" value.

The "m" value when using the CS extension register is value of "CS10 to 0 +1".

Bus clock frequency register (FSR)

Address	Base Address+0018							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	-	-	-	-	FS3	FS2	FS1	FS0
Initial	-	-	-	-	L	L	L	H

[Bit7 to 4: unused]

At read time, this bit field is always "0000".

[Bit3 to 0: FS3 to 0(Bus Clock Frequency Select 3 to 0)]

This bit field selects the bus clock frequency used. When this register is set, characteristics of the noise filter, etc. are set. Standard set values are show in the table below. If necessary, adjust them depending on the characteristic of the I²C buffer used and the noise condition on the I²C bus.

FS3	FS2	FS1	FS0	Frequency [MHz]
0	0	0	0	Setting is disabled.
0	0	0	1	14 to less than 20
0	0	1	0	20 to less than 40
0	0	1	1	40 to less than 60
0	1	0	0	60 to less than 80
0	1	0	1	80 to less than 100
0	1	1	0	100 to less than 120
0	1	1	1	120 to less than 140
1	0	0	0	140 to less than 160
1	0	0	1	160 to less than 180
1	0	1	0	180 to less than 200
1	0	1	1	200 to less than 220
1	1	0	0	—
1	1	0	1	—
1	1	1	0	—
1	1	1	1	—

Bus control 2 register (BC2R)

Address	Base Address+0014							
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	-	-	SDAS	SCLS			SDAL	SCLL
Initial	-	-	0	-	-	-	L	L

[Bit7, 6: unused]

At read from this field, "00" is always read.

[Bit5: SDAS (SDA Status)]

This field indicates the SDA line signal level that is enabled after the signal has passed the noise filter.

This field is only enabled for read.

SDAS	State
0	SDA line is "0".
1	SDA line is "1".

[Bit4: SCLS (SCL Status)]

This field indicates the SCL line signal level that is enabled after the signal has passed the noise filter.

This field is only enabled for read.

SCLS	State
0	SCL line is "0".
1	SCL line is "1".

[Bit3, 2: unused]

At read from this field, "00" is always read.

[Bit1: SDAL (SDA Low drive)]

This field forcibly sets SDAL output to "L".

Both read from and write to the field can be performed.

SDAL	State
0	SDAL Output is normal operation.
1	Forcibly sets SDAL output to "L".

[Bit0: SCLL (SCL Low drive)]

This field forcibly sets SCLO output to "L".

Both read from and write to the field can be performed.

SCLL	State
0	SCLO output is normal operation.
1	Forcibly sets SCLO output to "L".

5 CARMINE CONTROL

5.1 Interrupt

Carmine has a function to report interrupt information from inside it to the outside.

It detects rising edge of the INT signal (level output) output from the each internal module and stores the rising edge in the Status register, and at the same time, outputs the logical sum (OR) of all INT information of the register to the external pin (XINT).

However, INT signal masked by the INT Mask register is not written to the Status register. The XINT signal is asynchronous to the PCLK clock.

To clear the interrupt, clear the interrupt factor and then clear the interrupt in the Status register.

5.1.1 Block Diagram

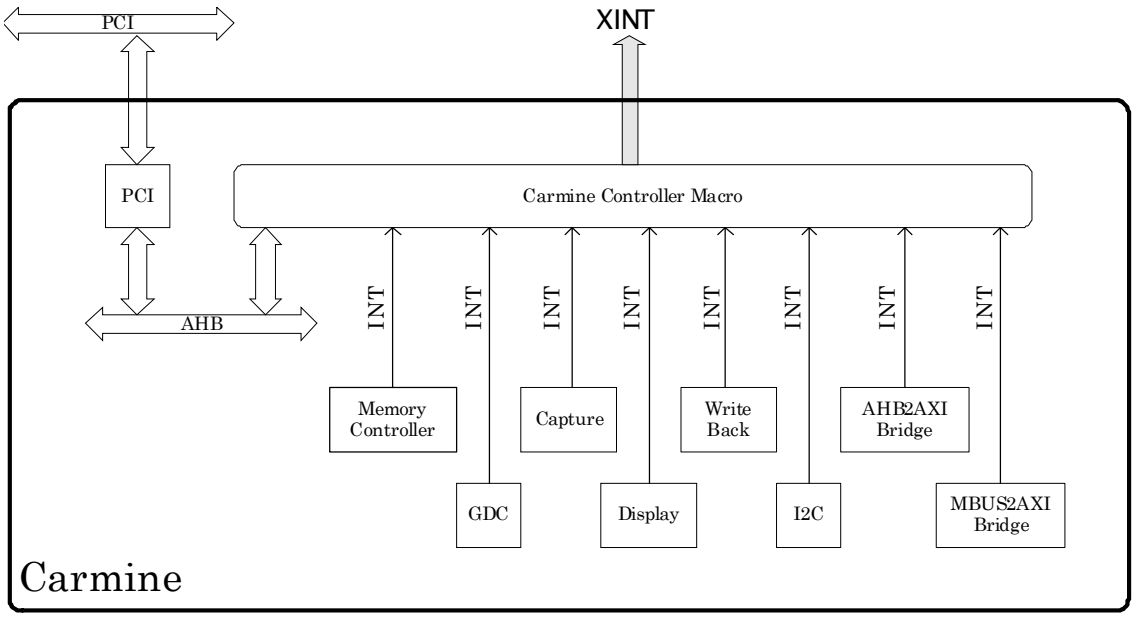


Fig. 5.1 Block Diagram

5.2 Reset Sequence

The CCNT module controls the following types of reset shown in ***Table 5.2.1***.

The reset types are as follows (***Table 5.2.1*** Reset support table).

Macro Soft Reset is a register for debugging. Be careful when using it. (See the description of ***Macro Soft Reset register***.)

AHB, APB, AXI, MBUS and HBUS are specific names for Carmine internal bus.

All resets in this chip are asynchronous resets.

Table 5.2.1 Reset Correspondence Table

	PCI	CLK GEN	CCNT	AHB	CAP0	CAP1	DSP0	DSP1	WB	2D/3D	MEC	I2C	AXI	AHB2 AXI	MBUS 2AXI	AHB2 HBUS	AHB2 APB
XRST	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
PLL RESET	×	○	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Soft Reset ALL	×	×	×	×	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
Macro Soft Reset	×	×	×	×	○	○	○	○	○	○	○	○	○	○	○	○	○

Simultaneous reset, ○: Individual reset, ×: Reset not supported

5.3 Carmine Control Registers

5.3.1 Register List

Address	Register Name	Description
0x0040_0000	Status register	Status register
0x0040_0004	INT Mask	INT Mask register
0x0040_0008	CCNT Reserve	Reserved register
0x0040_000C	Clock Enable	Clock Enable register
0x0040_0010	Soft reset	Chip Soft reset register
0x0040_1000	Macro Soft Reset	Macro Soft Reset register

5.3.2 Register Details

Base Address = 0x0040_0000

STATUS register

Register Address	**Base Address** + 0000															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved					INT26	INT25	INT24	INT23	INT22	Reserved	INT20	INT19	INT18	INT17	INT16
R/W	R					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0					0	0	0	0	0	0	0	0	0	0	0
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	INT15	INT14	INT13	INT12	INT11	INT10	Reserved	INT8	INT7	INT6	Reserved	INT4	INT3	INT2	INT1	INT0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 0 INT0 (CAPTURE0 VSYNC)

Indicates the state of a vertical synchronization interrupt of CAPTURE0 or the occurrence of video input error.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 0 of the INT Mask register (INIT0 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 1 INT1 (CAPTURE1 VSYNC)

Indicates the state of a vertical synchronization interrupt of CAPTURE1 or the occurrence of video input error.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 1 of the INT Mask register (INIT1 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 2 INT2 (DISPLAY0 VSYNC)

Indicates the state of a vertical synchronization interrupt of DISPLAY0.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 2 of the INT Mask register (INIT2 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 3 INT3 (DISPLAY0 FSYNC)

Indicates the state of a frame synchronization interrupt of DISPLAY0.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 3 of the INT Mask register (INIT3 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 4 INT4 (DISPLAY0 EXTSYNC ERROR)

Indicates the state of an external synchronization error of DISPLAY0.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 4 of the INT Mask register (INIT4 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 5 Reserved

Bit 6 INT6 (DISPLAY1 VSYNC)

Indicates the state of a vertical synchronization interrupt of DISPLAY1.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 6 of the INT Mask register (INIT6 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 7 INT7 (DISPLAY1 FSYNC)

Indicates the state of a vertical synchronization interrupt of DISPLAY1.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 7 of the INT Mask register (INIT7 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 8 INT8 (DISPLAY1 EXTSYNC ERROR)

Indicates the state of an external synchronization error of DISPLAY1.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 8 of the INT Mask register (INIT8 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 9 Reserved

Bit 10 INT10 (WRITE BACK)

Indicates the state of end interrupt of write back operation.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 10 of the INT Mask register (INIT10 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 11 INT11 (2D/3D GRAPHICS)

Indicates the state of end interrupt of drawing command.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 11 of the INT Mask register (INIT11 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 12 INT12 (2D/3D GRAPHICS ERROR)

Indicates the state of error interrupt of drawing command.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 12 of the INT Mask register (INIT12 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 13 INT13 (DRAM CONTROLLER)

Indicates the state of error interrupt of DRAM controller.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 13 of the INT Mask register (INIT13 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 14 INT14 (I^2C)

Indicates that an I^2C interrupt occurs.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 14 of the INT Mask register (INIT14 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 15 INT15 (MBUS2AXI READ ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 15 of the INT Mask register (INIT15 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 16 INT16 (MBUS2AXI WRITE ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 16 of the INT Mask register (INIT16 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 17 INT17 (AHB2AXI READ ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 17 of the INT Mask register (INIT17 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 18 INT18 (AHB2AXI WRITE ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 18 of the INT Mask register (INIT18 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 19 INT19 (AHB2HBUS ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 19 of the INT Mask register (INIT19 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 20 INT20 (PCI2AHB ERROR)

Indicates the protocol error of internal bus (for debugging).
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 20 of the INT Mask register (INIT20 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 21 Reserved

Bit 22 INT22 (PLL-U n Lock)

Indicates that PLL is unlocked during operation of PLL.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 22 of the INT Mask register (INIT22 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 23 INT23 (DLL – UnLock1)

Indicates that DLL for MDQS3 to 0 is unlocked during operation of DLL1.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 23 of the INT Mask register (INIT23 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 24 INT24 (DLL – ALM_1)

Indicates that the operation of DLL for MDQSs (3 to 0) is abnormal.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 24 of the INT Mask register (INIT24 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 25 INT25 (DLL – UnLock2)

Indicates that DLL for MDQS7 to 4 is unlocked during operation of DLL2.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 25 of the INT Mask register (INIT25 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

Bit 26 INT26 (DLL - ALM_2)

Indicates that the operation of DLL for MDQSs (7 to 4) is abnormal.
Writing "1" to this bit clears INT information to "0".
This bit is fixed at "0" when "0" is set to bit 26 of the INT Mask register (INIT26 Mask).

0: Interrupt not occurred (initial value).

1: Interrupt occurred.

INT Mask register

Register address	**Base Address** + 0004															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserve					INT26 Mask	INT25 Mask	INT24 Mask	INT23 Mask	INT22 Mask	INT21 Mask	INT20 Mask	INT19 Mask	INT18 Mask	INT17 Mask	INT16 Mask
R/W	R					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0					0	0	0	0	0	0	0	0	0	0	0
Bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	INT15 Mask	INT14 Mask	INT13 Mask	INT12 Mask	INT11 Mask	INT10 Mask	INT9 Mask	INT8 Mask	INT7 Mask	INT6 Mask	INT5 Mask	INT4 Mask	INT3 Mask	INT2 Mask	INT1 Mask	INT0 Mask
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 0 INT0 Mask (CAPTURE0 VSYNC)
Writing "1" to this bit enables INT information of the Status register bit 0.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 0 information is fixed at "0" (no interrupts occur).
1: Enables INT0.

Bit 1 INT1 Mask (CAPTURE1 VSYNC)
Writing "1" to this bit enables INT information of the Status register bit 1.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 1 information is fixed at "0" (no interrupts occur).
1: Enables INT1.

Bit 2 INT2 Mask 'DISPLAY0 VSYNC)
Writing "1" to this bit enables INT information of the Status register bit 2.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 2 information is fixed at "0" (no interrupts occur).
1: Enables INT2.

Bit 3 INT3 Mask (DISPLAY0 FSYNC)
Writing "1" to this bit enables INT information of the Status register bit 3.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 3 information is fixed at "0" (no interrupts occur).
1: Enables INT3.

Bit 4 INT4 Mask (DISPLAY0 EXTSYNC ERROR)
Writing "1" to this bit enables INT information of the Status register bit 4.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 4 information is fixed at "0" (no interrupts occur).
1: Enables INT4.

Bit 5 INT5 Mask (Reserved)
Writing "1" to this bit enables INT information of the Status register bit 5.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 5 information is fixed at "0" (no interrupts occur).
1: Enables INT5.

Bit 6 INT6 Mask (DISPLAY1 VSYNC)
Writing "1" to this bit enables INT information of the Status register bit 6.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 6 information is fixed at "0" (no interrupts occur).
1: Enables INT6.

Bit 7 INT7 Mask 'DISPLAY1 FSYNC)
Writing "1" to this bit enables INT information of the Status register bit 7.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 7 information is fixed at "0" (no interrupts occur).
1: Enables INT7.

Bit 8 INT8 Mask (DISPLAY1 EXTSYNC ERROR)
Writing "1" to this bit enables INT information of the Status register bit 8.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 8 information is fixed at "0" (no interrupts occur).
1: Enables INT8.

Bit 9 INT9 Mask (Reserved)
Writing "1" to this bit enables INT information of the Status register bit 9.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 9 information is fixed at "0" (no interrupts occur).
1: Enables INT9.

Bit 10 INT10 Mask (WRITE BACK)
Writing "1" to this bit enables INT information of the Status register bit 10.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 10 information is fixed at "0" (no interrupts occur).
1: Enables INT10.

Bit 11 INT11 Mask (2D/3D GRAPHICSA)
Writing "1" to this bit enables INT information of the Status register bit 11.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 11 information is fixed at "0" (no interrupts occur).
1: Enables INT11.

Bit 12 INT12 Mask (2D/3D GRAPHICS ERROR)
Writing "1" to this bit enables INT information of the Status register bit 12.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 12 information is fixed at "0" (no interrupts occur).
1: Enables INT12.

Bit 13 INT13 Mask (DRAM CONTROLLER)
Writing "1" to this bit enables INT information of the Status register bit 13.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 13 information is fixed at "0" (no interrupts occur).
1: Enables INT13.

Bit 14 INT14 Mask (I2C)
Writing "1" to this bit enables INT information of the Status register bit 14.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 14 information is fixed at "0" (no interrupts occur).
1: Enables INT14.

Bit 15 INT15 Mask (MBUS2AXI READ ERRR)
Writing "1" to this bit enables INT information of the Status register bit 15.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 15 information is fixed at "0" (no interrupts occur).
1: Enables INT15.

Bit 16 INT16 Mask (MBUS2AXI WRITE ERR)
Writing "1" to this bit enables INT information of the Status register bit 16.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 16 information is fixed at "0" (no interrupts occur).
1: Enables INT16.

Bit 17 INT17 Mask (AHB2AXI READ ERROR)

Writing "1" to this bit enables INT information of the Status register bit 17.

Information of Status register is cleared when this bit is set to "masked" from "enabled".

0: Mask ON (initial value). Status register bit 17 information is fixed at "0" (no interrupts occur).

1: Enables INT17.

Bit 18 INT18 Mask (AHB2AXI WRITE ERROR)
Writing "1" to this bit enables INT information of the Status register bit 18.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 18 information is fixed at "0" (no interrupts occur).
1: Enables INT18.

Bit 19 INT19 Mask (AHB2HBUS ERROR)
Writing "1" to this bit enables INT information of the Status register bit 19.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 19 information is fixed at "0" (no interrupts occur).
1: Enables INT19.

Bit 20 INT20 Mask (PCI2AHB ERROR)
Writing "1" to this bit enables INT information of the Status register bit 20.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 20 information is fixed at "0" (no interrupts occur).
1: Enables INT20.

Bit 21 INT21 Mask (Reserved)
Writing "1" to this bit enables INT information of the Status register bit 6.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 6 information is fixed at "0" (no interrupts occur).
1: Enables INT21.

Bit 22 INT22 Mask (PLL UnLock)
Writing "1" to this bit enables INT information of the Status register bit 22.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 22 information is fixed at "0" (no interrupts occur).
1: Enables INT22.

Bit 23 INT23 Mask (DLL UnLock)
Writing "1" to this bit enables INT information of the Status register bit 23.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 23 information is fixed at "0" (no interrupts occur).
1: Enables INT23.

Bit 24 INT24 Mask (DLL ALM_1)
Writing "1" to this bit enables INT information of the Status register bit 24.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 24 information is fixed at "0" (no interrupts occur).
1: Enables INT24.

Bit 25 INT25 Mask (DLL UnLock2)
Writing "1" to this bit enables INT information of the Status register bit 25.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 25 information is fixed at "0" (no interrupts occur).
1: Enables INT25.

Bit 26 INT26 Mask (DLL ALM_2)
Writing "1" to this bit enables INT information of the Status register bit 26.
Information of Status register is cleared when this bit is set to "masked" from "enabled".
0: Mask ON (initial value). Status register bit 26 information is fixed at "0" (no interrupts occur).
1: Enables INT26.

CCNT Reserved register

Register address	**Base Address** + 0008							
Bit No.	7	6	5	4	3	2	1	0
Bit field name	Reserved							
R/W	R							
Initial value	0							

Bit 7-0 Reserved
Write disabled.

Clock Enable register

Register address	**Base Address** + 000C															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						CKEN9	CKEN8	CKEN7	CKEN6	CKEN5	CKEN4	CKEN3	CKEN2	CKEN1	CKEN0
R/W	R						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0						0	0	0	0	0	0	0	0	0	0

Bit 0 CKEN0 (DRAM Controller Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 1 CKEN1 (2D/3D Graphic Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 2 CKEN2 (I^2C Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 3 CKEN3 (Capture0 Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 4 CKEN4 (Capture1 Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 5 CKEN5 (Display0 Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 6 CKEN6 (Display1 Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 7 CKEN7 (Write Back Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 8 CKEN8 (656 Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

Bit 9 CKEN9 (RGB Clock Enable)
Writing "1" to this bit enables clock supply.
0: Disables clock supply (initial value).
1: Enables clock supply.

(Cautions)

When controlling this clock, note the following points:

1. When stopping the clock supply, check in advance that the target module is in the idle state. For the idle state of the module, see the description of registers in each module.
2. Do not access the target module with clock supply stopped. When bit 20 of the INTMASK register is enabled (mask cancelled) and then a module with its clock stopped is accessed, bit 20 (PCI2AHB Error) of the Status register causes an interrupt.

Software reset register

Register Address	**Base Address** + 0010							
Bit No.	7	6	5	4	3	2	1	0
Bit field name	Reserved							SFTRST
R/W	R							RW
Initial value	0							0

Bit 0 SFTRST (Soft reset)

Writing "1" to this bit outputs a reset to all modules in the chip (except PCI module, Clock Generator module and this module).

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Reset (initial value)

1: Reset

Macro Soft Reset Register (for Debug)

Register address	**Base Address** + 1000															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve			AHB 2APB	AHB 2HBUS	MBUS 2AXI	AHB 2AXI	AXI	WB	Disp1	Disp0	Cap1	Cap0	I²C	2D/3D	DRAMC
R/W	R			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0			0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 DRAMC (DRAM Controller Soft Reset)

Writing "1" to this bit outputs a reset to the DRAM Controller macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit1 2D/3D (2D/3D Graphic Soft Reset)

Writing "1" to this bit outputs a reset to the 2D/3D Graphic macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit2 I2C (I2C Soft Reset)

Writing "1" to this bit outputs a reset to the I2C macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit3 Cap0 (Capture0 Soft Reset)

Writing "1" to this bit outputs a reset to the Capture0 macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit4 Cap1 (Capture1 Soft Reset)

When 1 is written to this bit, a reset is output to the Capture1 macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit5 Disp0 (Display0 Soft Reset)

Writing "1" to this bit outputs a reset to the Display0 macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit6 Disp1 (Display1 Soft Reset)

Writing "1" to this bit outputs a reset to the Display1 macro.

In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.

0: No Soft Reset (initial value)

1: Soft Reset

Bit7 WB (Write Back Soft Reset)
Writing "1" to this bit outputs a reset to the Write Back macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

Bit8 AXI (AXI Soft Reset)
Writing "1" to this bit outputs a reset to the AXI macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

Bit9 AHB2AXI (AHB2AXI Soft Reset)
Writing "1" to this bit outputs a reset to the AHB2AXI macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

Bit10 MBUS2AXI (MBUS2AXI Soft Reset)
Writing "1" to this bit outputs a reset to the MBUS2AXI macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

Bit11 AHB2HBUS (AHB2HBUS Soft Reset)
Writing "1" to this bit outputs a reset to the AHB2HBUS macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

Bit12 AHB2APB (AHB2APB Soft Reset)
Writing "1" to this bit outputs a reset to the AHB2APB macro.
In this case, the value of this register is output as it is (level output). To cancel the reset, set "0" to this bit again.
0: No Soft Reset (initial value)
1: Soft Reset

(Cautions)

1. This register is for debugging at software development. The register may not be restored correctly depending on the Reset timing, therefore do not use the register in products.
2. When using the register, note the following points:

   When performing this Reset while the macro to be Reset is using the BUS, the BUS may deadlock. Before executing the Reset, always check that the macro is in the idle state. For the idle state of each macro, see the description of registers in each macro.
3. When executing the Reset, the macro to be Reset returns to the initial state, so perform the various settings again.

6 KOTTOS (2D/3D Graphics)

6.1 Overview

"KOTTOS" is a module to draw 2D/3D graphics.

6.1.1 Interface

Endian

- Little Endian supported.

Interrupt

- Normal interrupt (Interrupt command)
- Abnormal interrupt (command error interrupt, internal error interrupt)

Display synchronization

- Vertical synchronization signal input (two displays supported)

6.1.2 Function

Command interpretation

- Interprets drawing commands (display list).
- Supported graphics include dot, straight line, triangle, quadrangle, concave/convex polygon (arbitrary polygon), and rectangle (BitBlt).

Coordinate transformation

- Performs matrix calculation operation on model coordinates to obtain clip space coordinates.
- Transforms view volume for the clip space coordinates to obtain device coordinates.
- Performs view volume clipping for graphics not fall into the view volume.
- Performs culling of back faces where not to be drawn.

Lighting

- Performs lighting according to the setting of light source and materials.
- Supports up to eight light sources and two types of light source: ambient light and diffuse light.

Interrupt processing

- Asserts interrupt signal when executing G_Interrupt (display list) or a command error occurs.

Drawing color

Supports 8-bit index color data (which assumes display palette), 16-bit color and 32-bit color as drawing input data.

Drawing clipping

Performs clipping in order not to draw outside a rectangle frame by setting rectangle frame on the drawing frame so that no drawing is performed outside the rectangle frame.

Drawing effect

Antialiasing

Processes line boundary in units of subpixels and blends pixel color and color prior to drawing, reducing (smoothing) jaggies. Antialiasing cannot be used in 8-bit color index mode.

Thick lines and broken lines

When drawing lines, specify line width and broken line. Supports verticalization of broken line pattern.

Alpha blending

Creates the effect of semi-transparency by blending the colors of two images. When alpha blending is used in 8-bit color index mode, blending is performed as 8-bit grayscale level.

Shading

Supports Gouraud shading. It allows for realistic shading and gradation of 3D objects can be expressed. When Gouraud shading is used in 8-bit color index mode, shading is performed as 8-bit grayscale level.

Texture mapping

Using texture mapping, a pattern can be drawn on the surface of a side using an image pattern. A texture pattern is placed in graphics memory. Texture of max "4096 × 4096" pixels can be used.

Attribute	Description
Perspective correction	Corrects texture coordinates for 3D objects.
MIP map	Controls the MIP map level of the texture referenced for each pixel.
Filtering	Point sampling, bilinear filtering, mip map, and trilinear filtering can be set.
Wrapping	CLAMP_TO_EDGE, REPEAT, BORDER, CLAMP, MIRRORED_REPEAT can be set.
Multi-texture	Up to two textures can be specified for a single object.
Expansion of compressed texture	Compressed texture can be used. Select lossless/lossy compression or palette format.

Fog

Supports fog function that makes distant view of object opaque depending on its depth. Fogging in units of pixels is possible.

Stencil test

References and updates 8-bit stencil buffers.

Write mask

Masks write of each component of RGBA to memory in 32-bit color and 16-bit color modes (Color mask).
Masks write to memory in bits in 8-bit index mode.
Masks write to Z buffers.
Masks write to stencil buffers in bits.

Register save/restore

This chip has functions to write the specified register to memory and to allow the register to read the written data.

6.2 Register List

This module has registers that can be accessed via PCI. Addresses in the following register lists are byte offsets added to the base address assigned to KOTTOS.

Space (byte address)	Description
0001_0000_H to 0001_FFFF_H	Host I/F space
0002_0000_H to 0002_7FFF_H	Rendering engine space
0002_8000_H to 0002_FFFF_H	Vertex reader space
0003_0000_H to 0003_FFFF_H	VL engine space
0004_0000_H to 0004_FFFF_H	Primitive engine space
0005_0000_H to 0005_FFFF_H	Clip engine space

Values in parenthesis in the Offset Address column are specified addresses for SaveRestoreReg.

Values in the ***SetReg*** column of the Rendering engine module are specified addresses in the SetRegister display list.

Table 6.2.1 Registers for Host I/F Module

Offset Address	Register Name	Description
0001_000C_H (0_4003_H)	INTR	Interrupt request
0001_0010_H (0_4004_H)	INTMASK	Interrupt mask
0001_0038_H (0_400E_H)	FRHALT	Stop/operation control of geometry processor
0001_003C_H (0_400F_H)	SRESET	Software reset control
0001_0040_H (0_4010_H)	ENDCHG	Internal I/F endian switching

Table 6.2.2 Registers of Rendering Engine Module (1)

Offset Address	Register Name	Set Reg	Description
0002_0000H (0_8000H)	Ys	0000H	Starting Y coordinate
0002_0004H (0_8001H)	Xs	0001H	Starting X coordinate
0002_0008H (0_8002H)	dXdy	0002H	Inclination value of long edge
0002_000CH (0_8003H)	XUs	0003H	Initial value of upper triangle of short edge
0002_0010H (0_8004H)	dXUdy	0004H	Inclination value of upper triangle of short edge
0002_0014H (0_8005H)	XLs	0005H	Initial value of lower triangle of short edge
0002_0018H (0_8006H)	dXLdy	0006H	Inclination value of lower triangle of short edge
0002_001CH (0_8007H)	USN	0007H	Span count of upper triangle
0002_0020H (0_8008H)	LSN	0008H	Span count of lower triangle
0002_0040H (0_8010H)	Rs	0010H	Initial value of red component of color
0002_0044H (0_8011H)	dRdx	0011H	X-direction incremental value of red component of color
0002_0048H (0_8012H)	dRdy	0012H	Y-direction incremental value of red component of color
0002_004CH (0_8013H)	Gs	0013H	Initial value of green component of color
0002_0050H (0_8014H)	dGdx	0014H	X-direction incremental value of green component of color
0002_0054H (0_8015H)	dGdy	0015H	Y-direction incremental value of green component of color
0002_0058H (0_8016H)	Bs	0016H	Initial value of blue component of color
0002_005CH (0_8017H)	dBdx	0017H	X-direction incremental value of blue component of color
0002_0060H (0_8018H)	dBdy	0018H	Y-direction incremental value of blue component of color
0002_0064H (0_8019H)	As	0019H	Initial value of α component of color
0002_0068H (0_801AH)	dAdx	001AH	X-direction incremental value of α component of color
0002_006CH (0_801BH)	dAdy	001BH	Y-direction incremental value of α component of color
0002_0080H (0_8020H)	Zs	0020H	Initial value of Z coordinate (only sign and decimal part when Z value is 32 bits)
0002_0084H (0_8021H)	dZdx	0021H	X-direction incremental value of Z coordinate (only sign and decimal part when Z value is 32 bits)
0002_0088H (0_8022H)	dZdy	0022H	Y-direction incremental value of Z coordinate (only sign and decimal part when Z value is 32 bits)
0002_008CH (0_8023H)	Z32s	0023H	Integer part of Zs when Z value is 32 bits
0002_0090H (0_8024H)	dZ32dx	0024H	Integer part of dZdx when Z value is 32 bits
0002_0094H (0_8025H)	dZ32dy	0025H	Integer part of dZdy when Z value is 32 bits
0002_00A0H (0_8028H)	Fs	0028H	Initial value of fog coordinate
0002_00A4H (0_8029H)	dFdx	0029H	X-direction incremental value of fog coordinate
0002_00A8H (0_802AH)	dFdy	002AH	Y-direction incremental value of fog coordinate
0002_00C0H (0_8030H)	S0s	0030H	Initial value of texture S coordinate of texture unit 0
0002_00C4H (0_8031H)	dS0dx	0031H	X-direction incremental value of texture S coordinate of texture unit 0
0002_00C8H (0_8032H)	dS0dy	0032H	Y-direction incremental value of texture S coordinate of texture unit 0
0002_00CCH (0_8033H)	T0s	0033H	Initial value of texture T coordinate of texture unit 0
0002_00D0H (0_8034H)	dT0dx	0034H	X-direction incremental value of texture T coordinate of texture unit 0
0002_00D4H (0_8035H)	dT0dy	0035H	Y-direction incremental value of texture T coordinate of texture unit 0
0002_00D8H (0_8036H)	Q0s	0036H	Initial value of texture Q coordinate of texture unit 0
0002_00DCH (0_8037H)	dQ0dx	0037H	X-direction incremental value of texture Q coordinate of texture unit 0
0002_00E0H (0_8038H)	dQ0dy	0038H	Y-direction incremental value of texture Q coordinate of texture unit 0
0002_00E4H (0_8039H)	S1s	0039H	Initial value of texture S coordinate of texture unit 1
0002_00E8H (0_803AH)	dS1dx	003AH	X-direction incremental value of texture S coordinate of texture unit 1
0002_00ECH (0_803BH)	dS1dy	003BH	Y-direction incremental value of texture S coordinate of texture unit 1
0002_00F0H (0_803CH)	T1s	003CH	Initial value of texture T coordinate of texture unit 1
0002_00F4H (0_803DH)	dT1dx	003DH	X-direction incremental value of texture T coordinate of texture unit 1
0002_00F8H (0_803EH)	dT1dy	003EH	Y-direction incremental value of texture T coordinate of texture unit 1
0002_00FCH (0_803FH)	Q1s	003FH	Initial value of texture Q coordinate of texture unit 1
0002_0100H (0_8040H)	dQ1dx	0040H	X-direction incremental value of texture Q coordinate of texture unit 1
0002_0104H (0_8041H)	dQ1dy	0041H	Y-direction incremental value of texture Q coordinate of texture unit 1

Table 6.2.3 Registers of Rendering Engine Module (2)

Offset Address	Register Name	Set Reg	Description
0002_0154H (0_8055H)	LZs	0055H	Register referenced as Z component initial value for DrawLine
0002_0158H (0_8056H)	LZde	0056H	Register referenced as Z component incremental value for DrawLine
0002_0280H (0_80A0H)	Tcolor	00A0H	Transparent color used in BitBlt transparent processing
0002_0284H (0_80A1H)	FormColor	00A1H	Forming color used in BitBlt die-cut processing
0002_0288H (0_80A2H)	LINEEXT	00A2H	Selects broken line pattern ("vertical to main axis" or "vertical to theoretical line")
0002_02A0H (0_80A8H)	BLDTU00	00A8H	Register referenced by PixelBlender when texture unit 0 is enabled
0002_02A4H (0_80A9H)	BLDTU01	00A9H	
0002_02A8H (0_80AAH)	BLDTU10	00AAH	Register referenced by PixelBlender when texture unit 1 is enabled
0002_02ACH (0_80ABH)	BLDTU11	00ABH	
0002_02F0H (0_80BCH)	BLDCONST	00BCH	CONST value of PixelBlender
0002_03E0H (0_80F8H)	BLPO	00F8H	Reference pointer of broken line pattern

Table 6.2.4 Registers of Rendering Engine Module (3)

Offset Address	Register Name	Set Reg	Description
0002_0420H (0_8108H)	MDR0	0108H	Setting related to basic mode of drawing, and character scaling setting
0002_0424H (0_8109H)	MDR1	0109H	Setting related to point and straight line
0002_0428H (0_810AH)	MDR2	010AH	Setting related to triangle and polygon
0002_0430H (0_810CH)	MDR4	010CH	Setting related to BitBlt
0002_0434H (0_810DH)	MDR5	010DH	Setting related to alpha blend
0002_0438H (0_810EH)	MDR6	010EH	Setting related to stencil test
0002_043CH (0_810FH)	MDR7	010FH	Setting related to fog coordinate
0002_0440H (0_8110H)	FBR	0110H	Base address of frame buffer
0002_0444H (0_8111H)	XRR	0111H	Horizontal pixel count of frame buffer
0002_0448H (0_8112H)	ZBR	0112H	Base address of Z buffer
0002_044CH (0_8113H)	TBR	0113H	Base address of texture (for upward compatibility)
0002_0450H (0_8114H)	PFBR	0114H	Base address of flag buffer for polygon
0002_0454H (0_8115H)	CXMIN	0115H	Upper left X coordinate of clip frame
0002_0458H (0_8116H)	CXMAX	0116H	Lower right X coordinate of clip frame
0002_045CH (0_8117H)	CYMIN	0117H	Upper left Y coordinate of clip frame
0002_0460H (0_8118H)	CYMAX	0118H	Lower right Y coordinate of clip frame
0002_0464H (0_8119H)	TXS	0119H	Texture size (for upward compatibility)
0002_0468H (0_811AH)	TIS	011AH	Tile size (for upward compatibility)
0002_0474H (0_811DH)	ABR	011DH	Base address of alpha map (for upward compatibility)
0002_0478H (0_811EH)	STCBR	011EH	Base address of stencil buffer
0002_047CH (0_811FH)	COLMASK	011FH	Color/stencil write mask control
0002_0480H (0_8120H)	FC	0120H	Foreground color
0002_0484H (0_8121H)	BC	0121H	Background color
0002_0488H (0_8122H)	ALF	0122H	Alpha blend value when alpha Gouraud is not used
0002_048CH (0_8123H)	BLP	0123H	Broken line pattern
0002_0490H (0_8124H)	ATR	0124H	Reference value of alpha test
0002_0498H (0_8126H)	STCR	0126H	Reference value of stencil test
0002_049CH (0_8127H)	FOGCOL	0127H	Target color of fog
0002_0540H (0_8150H)	LX0dc	0150H	Vertex 0 buffer: mirror of X coordinate register
0002_0544H (0_8151H)	LY0dc	0151H	Vertex 0 buffer: mirror of Y coordinate register
0002_0548H (0_8152H)	LX1dc	0152H	Vertex 1 buffer: mirror of X coordinate register
0002_054CH (0_8153H)	LY1dc	0153H	Vertex 1 buffer: mirror of Y coordinate register

Table 6.2.5 Registers of Vertex Reader Module (1)

Offset Address	Register Name	Description
0002_8000H (0_A000H)	GCTR	Control register
0002_8040H (0_A010H)	GMDR0	Mode setting register
0002_8044H (0_A011H)	GMDR1	Straight line mode setting register
0002_8048H (0_A012H)	GMDR2	Triangle mode setting register
0002_8050H (0_A014H)	GMDR2E	Triangle extended mode setting register
0002_8054H (0_A015H)	IDFOGL	Setting of numeric expression format of each element in OpenGL mode
0002_8058H (0_A016H)	IVAOGL	Enable/disable setting of vertex element in OpenGL mode
0002_8060H (0_A018H)	VRINT	Normal interrupt factor
0002_8064H (0_A019H)	VRINTM	Normal interrupt mask
0002_8068H (0_A01AH)	VRERR	Command error interrupt factor
0002_806CH (0_A01BH)	VRERRM	Command error interrupt mask
0002_8070H (0_A01CH)	DDLFIFO_STATUS	FIFO status in DirectDL mode
0002_8100H (0_A040H)	C_OXYO	Coral-compatible XY DC coordinate offset for shadow primitive
0002_8108H (0_A042H)	C_OZORG	Coral-compatible Z DC coordinate offset for body primitive
0002_810CH (0_A043H)	C_OZNTL	Coral-compatible Z DC coordinate offset for non-top-left primitive
0002_8114H (0_A045H)	C_OZSH	Coral-compatible Z DC coordinate offset for shadow primitive
0002_8200H (0_A080H)	C_MDR1	Coral-compatible MDR1 for body primitive (same as S_MDR1)
0002_820CH (0_A083H)	C_MDR2	Coral-compatible MDR2 for body primitive (same as S_MDR2)
0002_8210H (0_A084H)	C_MDR2S	Coral-compatible MDR2 for shadow primitive
0002_8214H (0_A085H)	C_MDR2TL	Coral-compatible MDR2 for non-top-left primitive
0002_8230H (0_A08CH)	C_FCC	Coral-compatible ForeColor
0002_8234H (0_A08DH)	C_BCC	Coral-compatible BackColor
0002_8238H (0_A08EH)	C_FCSC	Coral-compatible ForeColor for shadow primitive
0002_8248H (0_A092H)	C_LGA	Log write address for G_VertexNopLOG and G_VertexLOG
0002_8300H (0_A0C0H)	IDXBRCOORD	(X,Y,Z,W) base address of index DL
0002_8304H (0_A0C1H)	IDXSTRIDECOORD	(X,Y,Z,W) stride of index DL
0002_8308H (0_A0C2H)	IDXBRCOLF	(Rf,Gf,Bf,Af) base address of index DL
0002_830CH (0_A0C3H)	IDXSTRIDECOLF	(Rf,Gf,Bf,Af) stride of index DL
0002_8310H (0_A0C4H)	IDXBRCOLB	(Rb,Gb,Bb,Ab) base address of index DL
0002_8314H (0_A0C5H)	IDXSTRIDECOLB	(Rb,Gb,Bb,Ab) stride of index DL
0002_8318H (0_A0C6H)	IDXBRNORM	(Nx,Ny,Nz) base address of index DL
0002_831CH (0_A0C7H)	IDXSTRIDENORM	(Nx,Ny,Nz) stride of index DL
0002_8320H (0_A0C8H)	IDXBRTEX0	(S0,T0,Q0) base address of index DL
0002_8324H (0_A0C9H)	IDXSTRIDETEX0	(S0,T0,Q0) stride of index DL
0002_8328H (0_A0CAH)	IDXBRTEX1	(S1,T1,Q1) base address of index DL
0002_832CH (0_A0CBH)	IDXSTRIDETEX1	(S1,T1,Q1) stride of index DL
0002_8330H (0_A0CCH)	IDXBRF	(F) base address of index DL
0002_8334H (0_A0CDH)	IDFSTRIDEF	(F) stride of index DL

Table 6.2.6 Registers of Vertex Reader Module (2)

Offset Address	Register Name	Description
0002_8400H (0_A100H)	DDLFIFOG	Display list input FIFO
0002_8420H (0_A108H)	S_MDR0	Register to store MDR0 set value
0002_8424H (0_A109H)	S_MDR1	Register to store MDR1 set value
0002_8428H (0_A10AH)	S_MDR2	Register to store MDR2 set value
0002_8430H (0_A10CH)	S_MDR4	Register to store MDR4 set value
0002_8440H (0_A110H)	S_FBR	Register to store FBR set value
0002_8444H (0_A111H)	S_XRR	Register to store XRR set value
0002_8520H (0_A148H)	FR_ST	Status of geometry processor
0002_8524H (0_A149H)	CMDERR	Error display list. It is updated when command error occurs.
0002_8530h (0_A14Ch)	Access inhibited	Area used by hardware When this area is read, no response is made.
0002_8538H (0_A14EH)	DL_CNT	Display list counter. It is stopped when command error occurs.

0002_9000H to 0002_9FFFH is an access-inhibited area. When this area is read, no response is made

Table 6.2.7 Registers of VL Engine Module (1)

Offset Address	Register Name	Description
0003_0100H (0_C040H)	MVP00	Component of row 0 in column 0 in MVP matrix
0003_0104H (0_C041H)	MVP10	Component of row 0 in column 1 in MVP matrix
0003_0108H (0_C042H)	MVP20	Component of row 0 in column 2 in MVP matrix
0003_010CH (0_C043H)	MVP30	Component of row 0 in column 3 in MVP matrix
0003_0110H (0_C044H)	MVP01	Component of row 1 in column 0 in MVP matrix
0003_0114H (0_C045H)	MVP11	Component of row 1 in column 1 in MVP matrix
0003_0118H (0_C046H)	MVP21	Component of row 1 in column 2 in MVP matrix
0003_011CH (0_C047H)	MVP31	Component of row 1 in column 3 in MVP matrix
0003_0120H (0_C048H)	MVP02	Component of row 2 in column 0 in MVP matrix
0003_0124H (0_C049H)	MVP12	Component of row 2 in column 1 in MVP matrix
0003_0128H (0_C04AH)	MVP22	Component of row 2 in column 2 in MVP matrix
0003_012CH (0_C04BH)	MVP32	Component of row 2 in column 3 in MVP matrix
0003_0130H (0_C04CH)	MVP03	Component of row 3 in column 0 in MVP matrix
0003_0134H (0_C04DH)	MVP13	Component of row 3 in column 1 in MVP matrix
0003_0138H (0_C04EH)	MVP23	Component of row 3 in column 2 in MVP matrix
0003_013CH (0_C04FH)	MVP33	Component of row 3 in column 3 in MVP matrix
0003_0160H (0_C058H)	MV02	Component of row 2 in column 0 in MV matrix
0003_0164H (0_C059H)	MV12	Component of row 2 in column 1 in MV matrix
0003_0168H (0_C05AH)	MV22	Component of row 2 in column 2 in MV matrix
0003_016CH (0_C05BH)	MV32	Component of row 2 in column 3 in MV matrix
0003_0180H (0_C060H)	IMV00	Component of row 0 in column 0 in MV inverse matrix
0003_0184H (0_C061H)	IMV10	Component of row 0 in column 1 in MV inverse matrix
0003_0188H (0_C062H)	IMV20	Component of row 0 in column 2 in MV inverse matrix
0003_018CH (0_C064H)	IMV01	Component of row 1 in column 0 in MV inverse matrix
0003_0190H (0_C065H)	IMV11	Component of row 1 in column 1 in MV inverse matrix
0003_0194H (0_C066H)	IMV21	Component of row 1 in column 2 in MV inverse matrix
0003_0198H (0_C068H)	IMV02	Component of row 2 in column 0 in MV inverse matrix
0003_019CH (0_C069H)	IMV12	Component of row 2 in column 1 in MV inverse matrix
0003_01A0H (0_C06AH)	IMV22	Component of row 2 in column 2 in MV inverse matrix

Table 6.2.8 Registers of VL Engine Module (2)

Offset Address	Register Name	Description
0003_01C0$_H$ (0_C070$_H$)	VV_XMIN	Minimum X value of NDC coordinate of view volume
0003_01C4$_H$ (0_C071$_H$)	VV_XMAX	Maximum X value of NDC coordinate of view volume
0003_01C8$_H$ (0_C072$_H$)	VV_YMIN	Minimum Y value of NDC coordinate of view volume
0003_01CC$_H$ (0_C073$_H$)	VV_YMAX	Maximum Y value of NDC coordinate of view volume
0003_01D0$_H$ (0_C074$_H$)	VV_ZMIN	Minimum Z value of NDC coordinate of view volume
0003_01D4$_H$ (0_C075$_H$)	VV_ZMAX	Maximum Z value of NDC coordinate of view volume
0003_01D8$_H$ (0_C076$_H$)	VV_WMIN	Minimum Wd value of CC coordinate of view volume
0003_01E0$_H$ (0_C078$_H$)	VP_XScale	X scaling of NDC of viewport transformation
0003_01E4$_H$ (0_C079$_H$)	VP_XOffset	X offset of NDC of viewport transformation
0003_01E8$_H$ (0_C07A$_H$)	VP_YScale	Y scaling of NDC of viewport transformation
0003_01EC$_H$ (0_C07B$_H$)	VP_YOffset	Y offset of NDC of viewport transformation
0003_01F0$_H$ (0_C07C$_H$)	DR_ZScale	Z scaling of NDC of viewport transformation
0003_01F4$_H$ (0_C07D$_H$)	DR_ZOffset	Z offset of NDC of viewport transformation
0003_0200$_H$ (0_C080$_H$)	LG_AmbR	Red component of global ambient light
0003_0204$_H$ (0_C081$_H$)	LG_AmbG	Green component of global ambient light
0003_0208$_H$ (0_C082$_H$)	LG_AmbB	Blue component of global ambient light

Table 6.2.9 Registers of VL Engine Module (3)

Offset Address	Register Name	Description
0003_0220H (0_C088H)	L0_AmbR	Light source 0: Red component of ambient light
0003_0224H (0_C089H)	L0_AmbG	Light source 0: Green component of ambient light
0003_0228H (0_C08AH)	L0_AmbB	Light source 0: Blue component of ambient light
0003_0230H (0_C08CH)	L0_DiffR	Light source 0: Red component of diffuse light
0003_0234H (0_C08DH)	L0_DiffG	Light source 0: Green component of diffuse light
0003_0238H (0_C08EH)	L0_DiffB	Light source 0: Blue component of diffuse light
0003_0240H (0_C090H)	L0_PosX	Light source 0: X component of light source position
0003_0244H (0_C091H)	L0_PosY	Light source 0: Y component of light source position
0003_0248H (0_C092H)	L0_PosZ	Light source 0: Z component of light source position
0003_0270H (0_C09CH)	L1_AmbR	Light source 1: Red component of ambient light
0003_0274H (0_C09DH)	L1_AmbG	Light source 1: Green component of ambient light
0003_0278H (0_C09EH)	L1_AmbB	Light source 1: Blue component of ambient light
0003_0280H (0_C0A0H)	L1_DiffR	Light source 1: Red component of diffuse light
0003_0284H (0_C0A1H)	L1_DiffG	Light source 1: Green component of diffuse light
0003_0288H (0_C0A2H)	L1_DiffB	Light source 1: Blue component of diffuse light
0003_0290H (0_C0A4H)	L1_PosX	Light source 1: X component of light source position
0003_0294H (0_C0A5H)	L1_PosY	Light source 1: Y component of light source position
0003_0298H (0_C0A6H)	L1_PosZ	Light source 1: Z component of light source position
0003_02C0H (0_C0B0H)	L2_AmbR	Light source 2: Red component of ambient light
0003_02C4H (0_C0B1H)	L2_AmbG	Light source 2: Green component of ambient light
0003_02C8H (0_C0B2H)	L2_AmbB	Light source 2: Blue component of ambient light
0003_02D0H (0_C0B4H)	L2_DiffR	Light source 2: Red component of diffuse light
0003_02D4H (0_C0B5H)	L2_DiffG	Light source 2: Green component of diffuse light
0003_02D8H (0_C0B6H)	L2_DiffB	Light source 2: Blue component of diffuse light
0003_02E0H (0_C0B8H)	L2_PosX	Light source 2: X component of light source position
0003_02E4H (0_C0B9H)	L2_PosY	Light source 2: Y component of light source position
0003_02E8H (0_C0BAH)	L2_PosZ	Light source 2: Z component of light source position
0003_0310H (0_C0C4H)	L3_AmbR	Light source 3: Red component of ambient light
0003_0314H (0_C0C5H)	L3_AmbG	Light source 3: Green component of ambient light
0003_0318H (0_C0C6H)	L3_AmbB	Light source 3: Blue component of ambient light
0003_0320H (0_C0C8H)	L3_DiffR	Light source 3: Red component of diffuse light
0003_0324H (0_C0C9H)	L3_DiffG	Light source 3: Green component of diffuse light
0003_0328H (0_C0CAH)	L3_DiffB	Light source 3: Blue component of diffuse light
0003_0330H (0_C0CCH)	L3_PosX	Light source 3: X component of light source position
0003_0334H (0_C0CDH)	L3_PosY	Light source 3: Y component of light source position
0003_0338H (0_C0CEH)	L3_PosZ	Light source 3: Z component of light source position

Table 6.2.10 Registers of VL Engine Module (4)

Offset Address	Register Name	Description
0003_0360$_H$ (0_C0D8$_H$)	L4_AmbR	Light source 4: Red component of ambient light
0003_0364$_H$ (0_C0D9$_H$)	L4_AmbG	Light source 4: Green component of ambient light
0003_0368$_H$ (0_C0DA$_H$)	L4_AmbB	Light source 4: Blue component of ambient light
0003_0370$_H$ (0_C0DC$_H$)	L4_DiffR	Light source 4: Red component of diffuse light
0003_0374$_H$ (0_C0DD$_H$)	L4_DiffG	Light source 4: Green component of diffuse light
0003_0378$_H$ (0_C0DE$_H$)	L4_DiffB	Light source 4: Blue component of diffuse light
0003_0380$_H$ (0_C0E0$_H$)	L4_PosX	Light source 4: X component of light source position
0003_0384$_H$ (0_C0E1$_H$)	L4_PosY	Light source 4: Y component of light source position
0003_0388$_H$ (0_C0E2$_H$)	L4_PosZ	Light source 4: Z component of light source position
0003_03B0$_H$ (0_C0EC$_H$)	L5_AmbR	Light source 5: Red component of ambient light
0003_03B4$_H$ (0_C0ED$_H$)	L5_AmbG	Light source 5: Green component of ambient light
0003_03B8$_H$ (0_C0EE$_H$)	L5_AmbB	Light source 5: Blue component of ambient light
0003_03C0$_H$ (0_C0F0$_H$)	L5_DiffR	Light source 5: Red component of diffuse light
0003_03C4$_H$ (0_C0F1$_H$)	L5_DiffG	Light source 5: Green component of diffuse light
0003_03C8$_H$ (0_C0F2$_H$)	L5_DiffB	Light source 5: Blue component of diffuse light
0003_0340$_H$ (0_C0F4$_H$)	L5_PosX	Light source 5: X component of light source position
0003_0344$_H$ (0_C0F5$_H$)	L5_PosY	Light source 5: Y component of light source position
0003_0348$_H$ (0_C0F6$_H$)	L5_PosZ	Light source 5: Z component of light source position
0003_0400$_H$ (0_C100$_H$)	L6_AmbR	Light source 6: Red component of ambient light
0003_0404$_H$ (0_C101$_H$)	L6_AmbG	Light source 6: Green component of ambient light
0003_0408$_H$ (0_C102$_H$)	L6_AmbB	Light source 6: Blue component of ambient light
0003_0410$_H$ (0_C104$_H$)	L6_DiffR	Light source 6: Red component of diffuse light
0003_0424$_H$ (0_C105$_H$)	L6_DiffG	Light source 6: Green component of diffuse light
0003_0428$_H$ (0_C106$_H$)	L6_DiffB	Light source 6: Blue component of diffuse light
0003_0430$_H$ (0_C108$_H$)	L6_PosX	Light source 6: X component of light source position
0003_0434$_H$ (0_C109$_H$)	L6_PosY	Light source 6: Y component of light source position
0003_0438$_H$ (0_C10A$_H$)	L6_PosZ	Light source 6: Z component of light source position
0003_0450$_H$ (0_C114$_H$)	L7_AmbR	Light source 7: Red component of ambient light
0003_0454$_H$ (0_C115$_H$)	L7_AmbG	Light source 7: Green component of ambient light
0003_0458$_H$ (0_C116$_H$)	L7_AmbB	Light source 7: Blue component of ambient light
0003_0460$_H$ (0_C118$_H$)	L7_DiffR	Light source 7: Red component of diffuse light
0003_0464$_H$ (0_C119$_H$)	L7_DiffG	Light source 7: Green component of diffuse light
0003_0468$_H$ (0_C11A$_H$)	L7_DiffB	Light source 7: Blue component of diffuse light
0003_0470$_H$ (0_C11C$_H$)	L7_PosX	Light source 7: X component of light source position
0003_0474$_H$ (0_C11D$_H$)	L7_PosY	Light source 7: Y component of light source position
0003_0478$_H$ (0_C11E$_H$)	L7_PosZ	Light source 7: Z component of light source position

Table 6.2.11 Registers of VL Engine Module (5)

Offset Address	Register Name	Description
0003_0500H (0_C140H)	MF_AmbR	Material of front face: Red component of ambient light
0003_0504H (0_C141H)	MF_AmbG	Material of front face: Green component of ambient light
0003_0508H (0_C142H)	MF_AmbB	Material of front face: Blue component of ambient light
0003_0510H (0_C144H)	MF_DiffR	Material of front face: Red component of diffuse light
0003_0514H (0_C145H)	MF_DiffG	Material of front face: Green component of diffuse light
0003_0518H (0_C146H)	MF_DiffB	Material of front face: Blue component of diffuse light
0003_051CH (0_C147H)	MF_DiffA	Material of front face: α component of diffuse light
0003_0520H (0_C148H)	MF_EmisR	Material of front face: Red component of emitted light
0003_0524H (0_C149H)	MF_EmisG	Material of front face: Green component of emitted light
0003_0528H (0_C14AH)	MF_EmisB	Material of front face: Blue component of emitted light
0003_0550H (0_C154H)	MB_AmbR	Material of rear face: Red component of ambient light
0003_0554H (0_C155H)	MB_AmbG	Material of rear face: Green component of ambient light
0003_0558H (0_C156H)	MB_AmbB	Material of rear face: Blue component of ambient light
0003_0560H (0_C158H)	MB_DiffR	Material of rear face: Red component of diffuse light
0003_0564H (0_C159H)	MB_DiffG	Material of rear face: Green component of diffuse light
0003_0568H (0_C15AH)	MB_DiffB	Material of rear face: Blue component of diffuse light
0003_056CH (0_C15BH)	MB_DiffA	Material of rear face: α component of diffuse light
0003_0570H (0_C15CH)	MB_EmisR	Material of rear face: Red component of emitted light
0003_0574H (0_C15DH)	MB_EmisG	Material of rear face: Green component of emitted light
0003_0578H (0_C15EH)	MB_EmisB	Material of rear face: Blue component of emitted light
0003_05A0H (0_C168H)	VL_VERTEXSET	Enable/disable setting of vertex element
0003_05B0H (0_C16CH)	MATRIXSET	Setting related to MVP processing
0003_05C0H (0_C170H)	NVSF	F value used to perform normal vector scaling
0003_05D0H (0_C174H)	LIGHTSET	Setting related to lighting
0003_05E0H (0_C178H)	CLIPSET	Setting related to clipping

Table 6.2.12 Registers of Primitive Engine Module (1)

Offset Address	Register Name	Description
0004_0024H (1_0009H)	PO_FACTOR	Stores factor component of PolygonOffset
0004_0028H (1_000AH)	PO_UNITS	Stores units component of PolygonOffset
0004_002CH (1_000BH)	LINE_SET_REG	Stores the set value of G_LineSetting
0004_0030H (1_000CH)	POLYGON_SET_REG	Stores the set value of G_PolygonSetting
0004_0034H (1_000DH)	PR_VERTEXSET	Stores the set value of G_VertexSetting
0004_005CH (1_0017H)	DC-OFFSET-PX	Sets X coordinate of DC-offset for point
0004_0060H (1_0018H)	DC-OFFSET-PY	Sets Y coordinate of DC-offset for point
0004_0064H (1_0019H)	DC-OFFSET-LX	Sets X coordinate of DC-offset for line
0004_0068H (1_001AH)	DC-OFFSET-LY	Sets Y coordinate of DC-offset for line
0004_006CH (1_001BH)	DC-OFFSET-TX	Sets X coordinate of DC-offset for triangle
0004_0070H (1_001CH)	DC-OFFSET-TY	Sets Y coordinate of DC-offset for triangle

6.3 Register Details

This section describes the control registers of "XXXX module".

This module supports only 32-bit word access.

6.3.1 HOSTIF Module

Base Address = 0001_0000_H

INTR

Register address	**Base Address** + $000C_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																															AHBER
RW	R0																															RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 0 AHBER

This bit is for internal protocol error.

Read: Indicates the status of internal protocol.

Write: Clears the error status of internal protocol.

0 Protocol is normal.

1 Protocol error.

[Restriction]

- A signal that ORed error interrupt signal from AHBER and VertexReader is output as an error interrupt signal from the host I/F.

INTMASK

Register address	**Base Address** + 0010_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																															MAHBER
RW	R0																															RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit 0 MAHBER

Masks an interrupt generated when an error is detected in internal protocol.

0 Mask OFF.

1 Mask ON.

FRHALT

Register address	**Base Address** + 0038_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																															FRHALT
RW	R0																															RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit 0 FRHALT

Halts geometry processor.

0 Enables geometry processor boot.

1 HALT

SRESET

Register address	**Base Address** + $003C_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																														SRSTF	SRSTK
RW	R0																														RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit 0 SRSTK

Performs a software reset for KOTTOS. Writing "0" performs a reset. Writing "1" cancels the reset.

0 Performs a software reset.

1 Does not perform a software reset.

Bit 1 SRSTF

Performs a software reset for geometry processor. Writing "0" performs a reset. Writing "1" cancels the reset.

0 Performs a software reset.

1 Does not perform a software reset.

[Restriction]

- Both SRSTK and SRSTF are level resets. Write "0" → Write "1" must be performed from the external host.

ENDCHG

Register address	**Base Address** + 0040H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																														FIED	RGED
RW	R0																														RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 0 RGED

Specifies endian mode of register access.

0 Little endian

1 Big endian

Bit 1 FIED

Specifies endian mode of FIFO access.

0 Little endian

1 Big endian

6.3.2 Rendering Engine Module

Base Address = 0002_0000H

Ys register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0000H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Xs register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0004H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

dXdy register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0008 H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XUs register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 000CH																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

dXUdy register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0010H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XLs register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0014H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

dXLdy register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0018H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac															
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USN register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 001CH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name					Int																											
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LSN register

This register is for hardware debugging. It checks the operating status of setup processing.

Register address	**Base Address** + 0020H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name					Int																											
R/W	R				R												R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Rs register

Register address	**Base Address** + 0040H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the red component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dRdx register

Register address	**Base Address** + 0044H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the red component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dRdy register

Register address	**Base Address** + 0048 H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the red component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

Gs register

Register address	**Base Address** + 004CH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	R							R	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the green component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dGdx register

Register address	**Base Address** + 0050H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the green component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dGdy register

Register address	**Base Address** + 0054H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the green component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

Bs register

Register address	**Base Address** + 0058_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	R							R	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the blue component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dBdx register

Register address	**Base Address** + $005C_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the blue component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dBdy register

Register address	**Base Address** + 0060_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the blue component of a color. Set this register when using Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

As register

Register address	**Base Address** + 0064_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	R							R	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the α component of a color. Set this register when using α Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dAdx register

Register address	**Base Address** + 0068_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int								Frac															
R/W	RW							RW	RW								RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the α component of a color. Set this register when using α Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

dAdy register

Register address	**Base Address** + 006C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Don't care							S	Int																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the α component of a color. Set this register when using α Gouraud shading by DrawTrap. Specify this register using the 8-bit integer part and 16-bit decimal part irrespective of color mode.

Zs register

Register address	**Base Address** + 0080$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	0	Int															Frac															
R/W	RW	RW															RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the Z component. Set this register when using depth test by DrawPixel, DrawTrap, and polygon drawing. In Z value 32-bit mode, only the decimal part is enabled, and specify the integer part by Z32s.

dZdx register

Register address	**Base Address** + 0084$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S	Int															Frac															
R/W	RW	RW															RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the Z component. Set this register when using depth test by DrawTrap and polygon drawing. In Z value 32-bit mode, only the sign part and decimal part are enabled, and specify the integer part by dZ32dx.

dZdy register

Register address	**Base Address** + 0088$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S	Int															Frac															
R/W	RW	RW															RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the Z component. Set this register when using depth test by DrawTrap and polygon drawing. In Z value 32-bit mode, only the sign part and decimal part are enabled, and specify the integer part by dZ32dy.

Z32s register

Register address	**Base Address** + 008C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Int																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used in Z value 32-bit mode.
DrawTrap (triangle), Draw:PolygonEnd (polygon): Specify the decimal part by Zs.
DrawLine (straight line): Specify the decimal part by LZs.

dZ32dx register

Register address	**Base Address** + 0090H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Int																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used in Z value 32-bit mode. Specify the sign part and decimal part by dZdx.

dZ32dy register

Register address	**Base Address** + 0094H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Int																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used in Z value 32-bit mode.

DrawTrap (triangle), Draw:PolygonEnd (polygon): Specify the sign part and decimal part by dZdy.

DrawLine (straight line): Specify the sign part and decimal part by LZde.

Fs register

Register address	**Base Address** + 00A0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	0	Int																Frac														
R/W	RW	RW																RW														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the fog component. Set this register when using fog by DrawPixel, DrawTrap, and polygon drawing.

dFdx register

Register address	**Base Address** + 00A4H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S	Int																Frac														
R/W	RW	RW																RW														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the fog component. Set this register when using fog by DrawTrap and polygon drawing.

dFdy register

Register address	**Base Address** + 00A8H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S	Int																Frac														
R/W	RW	RW																RW														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the fog component. Set this register when using fog by DrawTrap and polygon drawing.

S0s register

Register address	**Base Address** + $00C0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the S component of texture 0. Set this register when using texture 0 by DrawPixel, DrawTrap, and polygon drawing.

dS0dx register

Register address	**Base Address** + $00C4_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the S component of texture 0. Set this register when using texture 0 by DrawTrap and polygon drawing.

dS0dy register

Register address	**Base Address** + $00C8_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the S component of texture 0. Set this register when using texture 0 by DrawTrap and polygon drawing.

T0s register

Register address	**Base Address** + $00CC_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the T component of texture 0. Set this register when using texture 0 by DrawPixel, DrawTrap, and polygon drawing.

dT0dx register

Register address	**Base Address** + $00D0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the T component of texture 0. Set this register when using texture 0 by DrawTrap and polygon drawing.

dT0dy register

Register address	**Base Address** + 00D4H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int													Frac															
R/W	RW			RW													RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the T component of texture 0. Set this register when using texture 0 by DrawTrap and polygon drawing.

Q0s register

Register address	**Base Address** + 00D8H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	0							Int	Frac																							
R/W	R							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the Q component of texture 0. Set this register when using perspective correction of texture 0 by DrawPixel, DrawTrap, and polygon drawing.

dQ0dx register

Register address	**Base Address** + 00DCH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S							Int	Frac																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the Q component of texture 0. Set this register when using perspective correction of texture 0 by DrawTrap and polygon drawing.

dQ0dy register

Register address	**Base Address** + 00E0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S							Int	Frac																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the Q component of texture 0. Set this register when using perspective correction of texture 0 by DrawTrap and polygon drawing.

S1s register

Register address	**Base Address** + 00E4H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int													Frac															
R/W	RW			RW													RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the S component of texture 1. Set this register when using texture 1 by DrawPixel, DrawTrap, and polygon drawing.

dS1dx register

Register address	**Base Address** + $00E8_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the S component of texture 1. Set this register when using texture 1 by DrawTrap and polygon drawing.

dS1dy register

Register address	**Base Address** + $00EC_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the S component of texture 1. Set this register when using texture 1 by DrawTrap and polygon drawing.

T1s register

Register address	**Base Address** + $00F0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the T component of texture 1. Set this register when using texture 1 by DrawPixel, DrawTrap, and polygon drawing.

dT1dx register

Register address	**Base Address** + $00F4_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the T component of texture 1. Set this register when using texture 1 by DrawTrap and polygon drawing.

dT1dy register

Register address	**Base Address** + $00F8_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S			Int																			Frac										
R/W	RW			RW																			RW										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the T component of texture 1. Set this register when using texture 1 by DrawTrap and polygon drawing.

Q1s register

Register address	**Base Address** + 00FC$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	0							Int	Frac																							
R/W	R							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an initial value of the Q component of texture 1. Set this register when using perspective correction of texture 1 by DrawPixel, DrawTrap, and polygon drawing.

dQ1dx register

Register address	**Base Address** + 0100$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S							Int	Frac																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is an X direction inclination value of the Q component of texture 1. Set this register when using perspective correction of texture 1 by DrawTrap and polygon drawing.

dQ1dy register

Register address	**Base Address** + 0104$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S							Int	Frac																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a Y direction inclination value of the Q component of texture 1. Set this register when using perspective correction of texture 1 by DrawTrap and polygon drawing.

LZs register

Register address	**Base Address** + 0154$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S							Int	Frac																							
R/W	RW							RW	RW																							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Use this register when setting the Z initial value by DrawLine.
In Z value 32-bit mode, only the decimal part is enabled, and specify the integer part by Z32s.

LZde register

Register address	**Base Address** + 0158$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	S	Int															Frac															
R/W	RW	RW															RW															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Use this register when setting the Z inclination value (main axis direction) by DrawLine.
In Z value 32-bit mode, only the sign part and decimal part are enabled, and specify the integer part by dZ32dy.

Tcolor register

Register address	**Base Address** + 0280H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Color																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31-0 Color

Sets the transparent color for BitBlt. For 16-bit color, sets a transparent color to lower 16 bits; for 8-bit color, sets a transparent palette code to lower 8 bits. Sets arrangement of the color components in the color data by CO of the MDR0 register (change of the arrangement is the same as for FC).

*All-bit match including the A component is achieved at transparency.

FormColor register

Register address	**Base Address** + 0284H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Color																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31-0 Color

Sets the forming color for BitBlt. For 16-bit color, sets a transparent color to lower 16 bits; for 8-bit color, sets a transparent palette code to lower 8 bits. Sets arrangement of the color components in the color data by CO of the MDR0 register (change of the arrangement is the same as for FC).

*All-bit match including the A component is achieved at transparency.

LINEEXT register

Register address	**Base Address** + 0288H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																BMP																
R/W	R															RW	R															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 16 BPM (Broken Pattern Mode)

Sets broken line pattern drawing mode

0 Draws a broken line pattern vertical to the main axis (this operation is compatible with MB86290A).

1 Draws a broken line pattern vertical to the theoretical line.

BLDTU00 register

Register address	**Base Address** + $02A0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name		SRC2A				SRC2RGB				SRC1A				SRC1RGB				SRC0A				SRC0RGB			FUNCA				FUNCRGB			
R/W	R	RW			R	RW			R	RW			R	RW				RW			R	RW			RW				RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Sets the blend processing of texture unit 0.

Bit 3-0 FUNCRGB (Blender Function for RGB)

Specifies operation mode of the pixel blender.

0000	REPLACE	Arg0
0001	MODULATE	Arg0 * Arg1
0010	ADD	Arg0 + Arg1
0011	ADD_SIGNED	Arg0 + Arg1 – 0.5 (0.5 is the value for when the Arg max value is “1”. This value varies with each format.)
0100	INTERPOLATE	Arg0 * Arg2 + Arg1 * (1 – Arg2)
0101	SUBTRACT	Arg0 – Arg1
0110	DOT3_RGB	4 * ((Arg0r – 0.5) * (Arg1r – 0.5) + (Arg0g – 0.5) * (Arg1g – 0.5) +(Arg0b – 0.5) * (Arg1b – 0.5))
0111	DOT3_RGBA	

Bit 7-4 FUNCA (Blender Function for A)

Specifies operation mode of the pixel blender.

0000	REPLACE	Arg0
0001	MODULATE	Arg0 * Arg1
0010	ADD	Arg0 + Arg1
0011	ADD_SIGNED	Arg0 + Arg1 - 0.5 (0.5 is the value for when the Arg max value is “1”. This value varies with each format.)
0100	INTERPOLATE	Arg0 * Arg2 + Arg1 * (1 – Arg2)
0101	SUBTRACT	Arg0 – Arg1

Bit 8-10 SRC0RGB (Source Arg0 RGB)

Specifies which of these to assign as the RGB component of the pixel blender Arg0.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

Bit 12-14 SRC0A (Source Arg0 A)

Specifies which of these to assign as the “A” component of the pixel blender Arg0.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

Bit 16-18 SRC1RGB (Source Arg1 RGB)

Specifies which of these to assign as the RGB component of the pixel blender Arg1.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

Bit 20-22 SRC1A (Source Arg1 A)

Specifies which of these to assign as the “A” component of the pixel blender Arg1.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

Bit 24-26 SRC2RGB (Source Arg2 RGB)

Specifies which of these to assign as the RGB component of the pixel blender Arg2.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

Bit 28-30 SRC2A (Source Arg2 A)

Specifies which of these to assign as the A component of the pixel blender Arg2.

000	CONSTANT	Uses the BLDRCONST value.
001	PRIMARY_COLOR	Uses the fragment color value before the pixel blender is activated.
010	BUFFER_COLOR	Uses the pixel color value written in the frame buffer.
011	PREVIOUS	Uses output of blender at previous stage When BLDTU00 is set to TU0, the setting is equivalent to the setting achieved as when PRIMARY_COLOR is specified. When BLDTU00 is set to TU1, uses output of TU0.
100	TEXTURE0 *1	Uses the texture assigned to texture unit 0.
101	TEXTURE1 *1	Uses the texture assigned to texture unit 1.

*1 TEXTURE0 and TEXTURE1 are restricted as follows:

When specifying TEXTURE0 as the source, enable TU0.

When specifying TEXTURE1 as the source, enable TU1.

BLDTU01 register

Register address	**Base Address** + 02A4H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name											OP2A				OP2 RGB				OP1A				OP1 RGB				OP0A				OP0 RGB	
R/W	R										RW		R		RW		R		RW		R		RW		R		RW		R		RW	
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Sets the blend processing of texture unit 0.

Bit 0-1 OP0RGB (Blender Operation for RGB of Arg0)

Specifies the transformation expression for the RGB component of the pixel blender Arg0.

00 SRC_COLOR (R, G, B)
01 ONE_MINUS_SRC_COLOR (1-R, 1-G, 1-B)
10 SRC_ALPHA (A, A, A)
11 ONE_MINUS_SRC_ALPHA (1-A, 1-A, 1-A)

Bit 4-5 OP0A (Blender Operation for A of Arg0)

Specifies the transformation expression for the A component of the pixel blender Arg0.

10 SRC_ALPHA (A)
11 ONE_MINUS_SRC_ALPHA (1-A)

Bit 8-9 OP1RGB (Blender Operation for RGB of Arg1)

Specifies the transformation expression for the RGB component of the pixel blender Arg1.

00 SRC_COLOR (R, G, B)
01 ONE_MINUS_SRC_COLOR (1-R, 1-G, 1-B)
10 SRC_ALPHA (A, A, A)
11 ONE_MINUS_SRC_ALPHA (1-A, 1-A, 1-A)

Bit 12-13 OP1A (Blender Operation for A of Arg1)

Specifies the transformation expression for the “A” component of the pixel blender Arg1.

10 SRC_ALPHA (A)
11 ONE_MINUS_SRC_ALPHA (1-A)

Bit 16-17 OP2RGB (Blender Operation for RGB of Arg2)

Specifies the transformation expression for the RGB component of the pixel blender Arg2.

00 SRC_COLOR (R, G, B)
01 ONE_MINUS_SRC_COLOR (1-R, 1-G, 1-B)
10 SRC_ALPHA (A, A, A)
11 ONE_MINUS_SRC_ALPHA (1-A, 1-A, 1-A)

Bit 20-21 OP2A (Blender Operation for A of Arg2)

Specifies the conversion expression for the “A” component of the pixel blender Arg2.

10 SRC_ALPHA (A)
11 ONE_MINUS_SRC_ALPHA (1-A)

BLDTU10 register

Register address	**Base Address** + 02A8H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name		SRC2A				SRC2RGB				SRC1A				SRC1RGB				SRC0A				SRC0RGB			FUNCA				FUNCRGB			
R/W	R	RW			R	RW			R	RW			R	RW				RW			R	RW			RW				RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Sets the blend processing of texture unit 1.

This register has the same fields as BLDTU00, but its initial value is different from that of the BLDTU00 register.

BLDTU11 register

Register address	**Base Address** + 02ACH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name											OP2A			OP2 RGB			OP1A			OP1 RGB			OP0A			OP0 RGB					
R/W	R										RW		R	RW		R	RW		R	RW		R	RW		R	RW						
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Sets the blend processing of texture unit 1.

This register has the same fields as BLDTU01.

BLDCONST register

Register address	**Base Address** + 02F0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	R								G								B							A								
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is referenced when the pixel blender selects CONSTANT as the source.

Specify each component using 8 bits irrespective of color mode.

BLPO register (Broken Line Pattern Offset)

Register address	**Base Address** + 03E0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																													BCR			
R/W																													RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores the bit number of a broken line pattern that is set to the BLP register when drawing a broken line. This register is incremented or decremented (specified using BPD of MDR1) every time one pixel is drawn. Setting a value to this register allows drawing a broken line pattern from an arbitrary starting position. When no write is performed to this register, the position of the broken line pattern is kept.

MDR0 register

Register address	**Base Address** + 0420H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	PXCM							CO			ZP					CF							CY	CX					BSV		BSH	
R/W	RW	R	R						R		RW		R			RW		R					RW	RW	R				RW		RW	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 1-0 BSH (Bitmap Scale Horizontal)

Sets the horizontal double size/half size function when drawing a bit pattern for BltDraw.

* When setting DrawBitmapP: BltDraw or DrawBitmapLargeP: BltDraw, always set this field to "00 (initial value)".

00:	Magnification is 1 (initial value)	Performs neither double size processing nor half size processing.
01:	Magnification is 2	Performs double size processing.
10:	Magnification is 1/2	Performs half size processing.

Bit 3-2 BSV (Bitmap Scale Vertical)

Sets the vertical double size/half size function when drawing a bit pattern for BltDraw.

* When setting DrawBitmapP: BltDraw or DrawBitmapLargeP: BltDraw, always set this field to "00 (initial value)".

00:	Magnification is 1 (initial value)	Performs neither double size processing nor half size processing.
01:	Magnification is 2	Performs double size processing.
10:	Magnification is 1/2	Performs half size processing.

Bit 8 CX (Clip X enable)

Sets whether to enable or disable the X direction clipping function.

0:	Disable (initial value)	Does not perform X direction clipping.
1:	Enable	Performs X direction clipping.

Bit 9 CY (Clip Y enable)

Sets whether to enable or disable the Y direction clipping function.

0:	Disable (initial value)	Does not perform Y direction clipping.
1:	Enable	Performs Y direction clipping.

Bit 16-15 CF (Color Format)

Sets the color mode of the frame buffer.

00:	8 bits/pixel (initial value)	Mode in which it is assumed that color is used as palette code
01:	16 bits/pixel	Mode in which each color component of RGB is 5 bits and the "A" component is 1 bit
10:	32 bits/pixel	Mode in which each color component of RGBA is 8 bits

Bit 21-20 ZP (Z Precision)

Sets the precision of the Z value used for hidden surface removal.

00:	16 bits/Z (initial value)	Mode in which the Z value is 16 bits
01:	8 bits/Z	Mode in which the Z value is 8 bits
10:	32 bits/Z	Mode in which the Z value is 32 bits

Bit 24 CO (Color Order)

Sets how to arrange the color components in color data and pixel provided by the display list and BitBlt function. This bit is enabled when drawing by 16-bit color and 32-bit color.

Register whose write operation is changed by CO, is FC and BC. When CO=0, input value "ARGB" is rearranged to "RGBA" and written to the register. When CO=1, input value "ARGB" is written to the register without being rearranged.

Display list whose color component arrangement is changed by CO, is the BorderColor field of RegTexture and the RGBA field of SetVertex/DrawVertex.

0:	ARGB (initial value)	Format compatible with Coral
1:	RGBA	Format used by OpenGL

Bit 31 PXCM (Pixel Center Mode)

Be sure to set "1" to this bit. The initial value is "0".

MDR1 register

Register address	**Base Address** + 0424H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name				LW					TU0	TU1	BP		BL	BW	BPD	FOG			LOG					BM		ZW	ZCL			ZC	AS	SM
R/W	R			RW					RW	RW	RW		RW	RW	RW	RW	R			RW				RW		RW	RW			RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bit 0 SM (Shading Mode)

Sets shading mode.

0:	Flat (initial value)	Flat shading
1:	Gouraud	Gouraud shading

Bit 1 AS (Alpha shading mode)

Performs the same interpolation processing as Gouraud shading, for alpha blend ratio.

0:	Flat (initial value)	Alpha flat shading
1:	Gouraud	Alpha Gouraud shading

Bit 2 ZC (Z Compare mode)

Sets whether to enable or disable Z value comparison mode. When the command is DrawPixelZ, "Z" comparison is performed irrespective of the ZC value. The ZC value does not change when forcibly comparing.

0:	Disable (initial value)	Does not perform Z comparison processing.
1:	Enable	Performs Z comparison processing.

Bit 5-3 ZCL (Z Compare Logic)

Sets the comparison expression when comparing Z value.

000:	NEVER
001:	ALWAYS
010:	LESS
011:	LEQUAL
100:	EQUAL
101:	GEQUAL
110:	GREATER
111:	NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets whether to mask the write of "Z" value, in Z value comparison mode. Even when the write mask of "Z" value is specified, comparison itself is performed.

0:	Does not mask (initial value).	Writes the "Z" value
1:	Masks.	Does not write the "Z" value.

Bit 8-7 BM (Blend Mode)

Sets blend mode for the line function.

00:	Normal (initial value)	Does not perform blend processing.
01:	Alpha blend	Performs alpha blend processing.
10:	Drawing with logical operation	For antialias line, operates as alpha blend mode When using this mode, set BFE of MDR5 to "0".

Bit 12-9 LOG (Logical operation)

Specifies the logical operation type when performing the drawing with logical operation.

0000: CLEAR
0001: AND
0010: AND REVERSE
0011: COPY (initial value)
0100: AND INVERTED
0101: NOP
0110: XOR
0111: OR
1000: NOR
1001: EQUIV
1010: INVERT
1011: OR REVERSE
1100: COPY INVERTED
1101: OR INVERTED
1110: NAND
1111: SET

Bit 16 FOG (Fog Enable)

Sets whether to enable or disable the fog function.

0: Disable (initial value)
1: Enable

Bit 17 BPD (Broken Pattern Direction)

Sets whether to set the reference starting direction to "upper bit -> lower bit" or "lower bit -> upper bit" when drawing broken line.

0:	Reference from upper bit (initial value)	Setting compatible with existing products
1:	Reference from lower bit	Setting compatible with OpenGL

Bit 18 BW (BC MSB Write)

Sets a value written to the MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1) of a pixel drawn by BC when drawing broken line.

The MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1) of the BC is used to enable/disable transparency, so use this BW to set the MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1).

This bit changes in conjunction with the BW bit of the MDR4 register.

0: Writes 0 (initial value).
1: Writes 1.

Bit 19 BL (Broken Line)

Sets whether to draw a broken line or solid line. This bit is ignored when a command is DrawPixel or DrawPixelZ.

0: Broken line (initial value)
1: Solid line

Bit 21-20 BP (Broken line Period)

Selects the cycle of the broken line pattern. This bit field is ignored when a command is DrawPixel or DrawPixelZ.

00: 32 bits (initial value)
01: 24 bits
10: 16 bits

Bit 22 TU1 (Texture Unit 1 enable)

Sets whether to enable or disable texture unit 1.

0: Disable (initial value)

1: Enable

Bit 23 TU0 (Texture Unit 0 enable)

Sets whether to enable or disable texture unit 0.

0: Disable (initial value)

1: Enable

Bit 28-24 LW (Line Width)

Sets the line width at the time of line drawing. This bit field is ignored when a command is DrawPixel or DrawPixelZ.

00000: 1 pixel (initial value).

00001: 2 pixels.

: :

11111: 32 pixels.

MDR2 register

Register address	**Base Address** + 0428H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			TU0	TU1												FOG				LOG				BM		ZW	ZCL				ZC	AS	SM
R/W	R		RW	RW	R											RW	R			RW				RW		RW	RW				RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bit 0 SM (Shading Mode)

Sets shading mode.

0:	Flat (initial value)	Flat shading
1:	Gouraud	Gouraud shading

Bit 1 AS (Alpha shading mode)

Performs the same interpolation processing as Gouraud shading, for alpha blend ratio.

0:	Flat (initial value)	Alpha flat shading
1:	Gouraud	Alpha Gouraud shading

Bi t2 ZC (Z Compare mode)

Sets whether to enable or disable Z value comparison mode.

0:	Disable (initial value)	Does not perform “Z” comparison processing.
1:	Enable	Performs “Z” comparison processing.

Bit 5-3 ZCL (Z Compare Logic)

Sets the comparison expression when comparing “Z” value.

000:	NEVER
001:	ALWAYS
010:	LESS
011:	LEQUAL
100:	EQUAL
101:	GEQUAL
110:	GREATER
111:	NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets whether to mask the write of the “Z” value, in Z value comparison mode. Even when the write mask of “Z” value is specified, comparison itself is performed.

0:	Does not mask (initial value).	Writes the “Z” value.
1:	Masks.	Does not write the “Z” value.

Bit 8-7 BM (Blend Mode)

Sets blend mode.

00:	Normal (initial value)	Does not perform blend processing. Performs alpha blend processing.
01:	Alpha blend	Performs drawing with logical operation in the logical mode set to LOG of MDR2.
10:	Drawing with logical operation	When using this mode, set BFE of MDR5 to 0.

Bit 12-9 LOG (Logical operation)

Specifies the logical operation type when performing the drawing with logical operation.

0000: CLEAR
0001: AND
0010: AND REVERSE
0011: COPY (initial value)
0100: AND INVERTED
0101: NOP
0110: XOR
0111: OR
1000: NOR
1001: EQUIV
1010: INVERT
1011: OR REVERSE
1100: COPY INVERTED
1101: OR INVERTED
1110: NAND
1111: SET

Bit 16 FOG (Fog Enable)

Sets whether to enable or disable the fog function.

0: Disable (initial value)
1: Enable

Bit 28 TU1 (Texture Unit 1 enable)

Sets whether to enable or disable texture unit 1.

0: Disable (initial value)
1: Enable

Bit 29 TU0 (Texture Unit 0 enable)

Sets whether to enable or disable texture unit 0.

0: Disable (initial value)
1: Enable

MDR4 register

Register address	**Base Address** + 0430H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name														BW						LOG				BM				FE			TE	AS
R/W	R													RW	RW					RW				RW		R		RW	RW		RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bit 0 AS (Alpha Select 32BPP)

Sets whether to use the ALF value or the "A" component of the pixel as the alpha factor when performing alpha blending. This bit can be used only for 32-bit color mode. When the BM is other than alpha blending, this AS is ignored.

* When setting DrawRectP(ClearPolyFlag), always set this field to "0 (initial value)".

* When enabling AS, set BM of this register to 01.

0: ALF (initial value).

1: "A" component of the pixel.

Bit 1 TE (Transparent Enable)

Sets whether to enable or disable transparent mode.

* When setting DrawRectP(BltFill/ClearPolyFlag), always set this field to "0 (initial value)".

0:	Disable (initial value)	Does not perform transparent processing.
1:	Enable	Does not draw the pixels that match the transparent color set to Tcolor, using the Blt function.

Bit 4 FE (Forming Enable)

Sets whether to enable or disable the forming mode.

* When setting DrawRectP(ClearPolyFlag), always set this field to "0 (initial value)".

0:	Disable (initial value)	Does not perform forming processing.
1:	Enable	Draws only the pixels in the drawing target positions that match the forming color set to FormColor, using the Blt function.

Bit 8-7 BM (Blend Mode)

Sets blend mode for the Blt function.

* When setting DrawRectP(ClearPolyFlag), always set this field to "00 (initial value)".

00:	Normal (initial value)	Does not perform blend processing.
01:	Alpha blend	Performs alpha blend processing using the transparency ratio set to the ALF register.
10:	Drawing with logical operation	Performs drawing with logical operation by the operation mode set to LOG of the MDR2 resgister.

Bit 12-9 LOG (Logical operation)

Specifies the logical operation type when performing the drawing with logical operation.

* When setting DrawRectP(ClearPolyFlag), always set this field to "0000 (CLEAR)".

0000: CLEAR
0001: AND
0010: AND REVERSE
0011: COPY (initial value)
0100: AND INVERTED
0101: NOP
0110: XOR
0111: OR
1000: NOR
1001: EQUIV
1010: INVERT
1011: OR REVERSE
1100: COPY INVERTED
1101: OR INVERTED
1110: NAND
1111: SET

Bit 18 BW (BC MSB Write)

Sets a value written to the MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1) of a pixel drawn by BC when drawing character.

The MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1) of the BC is used to enable/disable transparency, and so use this BW to set the MSB (CO of MDR0 = 0) or LSB (CO of MDR0 = 1).

This bit changes in conjunction with the BW bit of the MDR1 register.

0: Writes 0 (initial value).

1: Writes 1.

MDR5 register

Register address	**Base Address** + 0434H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name														ATFUNC			BLFUNCDST				BLFUNCSRC										ATE	BFE
R/W	R													RW			RW				RW				R						RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit 0 BFE (Blend Function Enable)

Sets blend function mode. Alpha blend must be enabled.

0: Disable (initial value) Does not perform the blend function processing.

1: Enable Performs the blend function processing.

Bit 1 ATE (Alpha Test Enable)

Sets alpha test mode. Alpha blend must be enabled.

0: Disable (initial value) Does not perform alpha test processing.

1: Enable Performs alpha test processing.

Bit 8-11 BLFUNCSRC (Blend Function of Source)

Sets how to calculate the source blend ratio when performing alpha blending.

0000 ZERO
0001 ONE (initial value)
0010 DST_COLOR
0011 SRC_COLOR
0100 ONE_MINUS_DST_COLOR
0101 ONE_MINUS_SRC_COLOR
0110 SRC_ALPHA
0111 ONE_MINUS_SRC_ALPHA
1000 DST_ALPHA
1001 ONE_MINUS_DST_ALPHA
1010 SRC_ALPHA_SATURATE

Bit 15-12 BLFUNCDST (Blend Function of Destination)

Sets how to calculate the destination blend ratio when performing alpha blending.

0000 ZERO (initial value)
0001 ONE
0010 DST_COLOR
0011 SRC_COLOR
0100 ONE_MINUS_DST_COLOR
0101 ONE_MINUS_SRC_COLOR
0110 SRC_ALPHA
0111 ONE_MINUS_SRC_ALPHA
1000 DST_ALPHA
1001 ONE_MINUS_DST_ALPHA
1010 SRC_ALPHA_SATURATE

Bit 18-16 ATFUNC (Alpha Test Function)

Specifies how to test when performing alpha blending.

000 NEVER
001 ALWAYS (初期値)
010 LESS
011 LEQUAL
100 EQUAL
101 GEQUAL
110 GREATER
111 NOTEQUAL

MDR6 register

Register address	**Base Address** + 0438H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name														DPPASS				DPFAIL				SFAIL				STFUNC						STCE
R/W	R													RW			R	RW			R	RW			R	RW			R			RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit 0 STCE (Stencil Test Enable)

Sets whether to enable or disable stencil test.

0: Disable (initial value) Does not perform stencil test processing and stencil buffer update.

1: Enable Performs stencil test processing and stencil buffer update.

Bit 4-6 STFUNC (Stencil Test Function)

Specifies how to compare with the reference value when performing stencil test.

000 NEVER
001 ALWAYS (initial value)
010 LESS
011 LEQUAL
100 EQUAL
101 GEQUAL
110 GREATER
111 NOTEQUAL

Bit 8-10 SFAIL (Stencil Test Fail function)

Specifies how to update the stencil buffer for when the STFUNC condition is not met when performing stencil test.

000 KEEP (initial value)
001 ZERO
010 REPLACE
011 INCR
100 DECR
101 INVERT
110 INCR_WRAP
111 DECR_WRAP

Bit 12-14 DPFAIL (Depth Test Fail Function)

Specifies how to update the stencil buffer for when the condition is not met when performing depth test.

000 KEEP (initial value)
001 ZERO
010 REPLACE
011 INCR
100 DECR
101 INVERT
110 INCR_WRAP
111 DECR_WRAP

Bit 16-18 DPPASS (Depth Test Pass Function)

Specifies how to update the stencil buffer for when the condition is met when performing depth test.

000 KEEP (initial value)
001 ZERO
010 REPLACE
011 INCR
100 DECR
101 INVERT
110 INCR_WRAP
111 DECR_WRAP

MDR7 register

Register address	**Base Address** + 043C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																								FOGCRD								
R/W	R																							RW	R							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 8 FOGCRD (Fog Coordinates select)

Sets whether to use the "Z" value or fog coordinate as a value used to calculate the fog factor.

0: Z (initial value) Uses the "Z" value.

1: FogCoord Uses the fog coordinate.

FBR register (Frame buffer Base)

Register address	**Base Address** + 0440$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			FBASE																													
R/W	R		RW																								R					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 29-6 FBASE (Frame buffer Base address)

Base address of the drawing frame. Even when a value is written to bit 5-0 and bit31-30, these values are ignored and treated as value "0".

XRR register (X Resolution)

Register address	**Base Address** + 0444$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																				XRES												
R/W	R																			RW												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 12-0 XRES (X Resolution)

Sets the horizontal resolution of the drawing frame using a pixel count. Set the value of "4096" or smaller in increments of 16. Be sure to set "0" to bit 0-3.

When using the BitBlt function, the horizontal length of the drawing frame must be aligned on an 8-byte boundary. With color mode in mind, set a horizontal pixel count that is 8-byte aligned.

ZBR register (Z buffer Base)

Register address	**Base Address** + 0448$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			ZBASE																													
R/W	R		RW																								R					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 29-6 ZBASE (Z buffer Base address)

Base address of the "Z" buffer. Even when a value is written to bit 5-0 and bit 31-30, these values are ignored and treated as value "0".

TBR register (Texture memory Base)

Register address	**Base Address** + 044C$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			TBASE																													
R/W	R		W																								R					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for upward compatibility; Fujitsu recommends that this register be specified by the RegTexture command in KOTTOS.

Bit 29-6 TBASE (Texture memory Base address)

Base address of texture. Even when a value is written to bit 5-0 and bit 31-30, these values are ignored and treated as value "0". Specification by this register is only reflected in 2 entries: texture ID:0 and texture ID:1.

PFBR register (Polygon Flag-Buffer Base)

Register address	**Base Address** + 0450$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			PFBASE																													
R/W	R		RW																								R					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 29-6 PFBASE (Polygon Flag- buffer Base address)

Base address of the flag buffer used for polygon drawing. Even when a value is written to bit 5-0 and bit 31-30, these values are ignored and treated as value "0".

CXMIN register (Clip X minimum)

Register address	**Base Address** + 0454$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPXMIN											
R/W	R																				RW											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 11-0 CLIPXMIN (Clip X minimum)

Specifies the upper left X coordinate in the clip frame.

CXMAX register (Clip X maximum)

Register address	**Base Address** + 0458$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPXMAX											
R/W	R																				RW											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 11-0 CLIPXMAX (Clip X maximum)

Specifies the lower right X coordinate in the clip frame.

CYMIN register (Clip Y minimum)

Register address	**Base Address** + 045C$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPYMIN											
R/W	R																				RW											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 11-0 CLIPYMIN (Clip Y minimum)

Specifies the upper left Y coordinate in the clip frame.

CYMAX register (Clip Y maximum)

Register address	**Base Address** + 0460_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPYMAX											
R/W	R																				RW											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 11-0 CLIPYMAX (Clip Y maximum)

Specifies the lower right Y coordinate in the clip frame.

TXS register

Register address	**Base Address** + 0464_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name				TXSN															TXSM														
R/W				W															W													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for upward compatibility; Fujitsu recommends that this register be specified by the RegTexture command in KOTTOS.

Bit 12-0 TXSM (Texture Size M)

Indicates the X direction texture size. A power of "2" value in the range of "1 to 4096" can be specified. Do not specify values other than these values.

Specification by this bit is reflected only in texture ID:0.

0_0000_0000_0001:	M=1	0_0000_1000_0000:	M=128
0_0000_0000_0010:	M=2	0_0001_0000_0000:	M=256
0_0000_0000_0100:	M=4	0_0010_0000_0000:	M=512
0_0000_0000_1000:	M=8	0_0100_0000_0000:	M=1024
0_0000_0001_0000:	M=16	0_1000_0000_0000:	M=2048
0_0000_0010_0000:	M=32	1_0000_0000_0000:	M=4096
0_0000_0100_0000:	M=64	Other than the above	Setting is disabled.

Bit 28-16 TXSN (Texture Size N)

Indicates the Y direction texture size. A power of "2" value in the range of "1 to 4096" can be specified. Do not specify values other than these values.

Specification by this bit is reflected only in texture ID:0.

0_0000_0000_0001:	M=1	0_0000_1000_0000:	M=128
0_0000_0000_0010:	M=2	0_0001_0000_0000:	M=256
0_0000_0000_0100:	M=4	0_0010_0000_0000:	M=512
0_0000_0000_1000:	M=8	0_0100_0000_0000:	M=1024
0_0000_0001_0000:	M=16	0_1000_0000_0000:	M=2048
0_0000_0010_0000:	M=32	1_0000_0000_0000:	M=4096
0_0000_0100_0000:	M=64	Other than the above	Setting is disabled.

TIS register

Register address	**Base Address** + 0468H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name				TISN															TISM													
R/W				W															W													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is provided to maintain backward compatibility; Fujitsu recommends that, for KOTTOS, this register be specified by the RegTexture command.

Bit 12-0 TISM (Tile Size M)

Indicates the X direction tile size. A power of "2" value in the range of "1 to 4096" can be specified. Do not specify values other than these values. Specification by this bit is reflected only in texture ID:1.

0_0000_0000_0001:	M=1	0_0000_1000_0000:	M=128
0_0000_0000_0010:	M=2	0_0001_0000_0000:	M=256
0_0000_0000_0100:	M=4	0_0010_0000_0000:	M=512
0_0000_0000_1000:	M=8	0_0100_0000_0000:	M=1024
0_0000_0001_0000:	M=16	0_1000_0000_0000:	M=2048
0_0000_0010_0000:	M=32	1_0000_0000_0000:	M=4096
0_0000_0100_0000:	M=64	Other than the above	Setting is disabled.

Bit 28-16 TISN (Tile Size N)

Indicates the Y direction tile size. A power of "2" value in the range of "1 to 4096" can be specified. Do not specify values other than these values. Specification by this bit is reflected only in texture ID:1.

0_0000_0000_0001:	M=1	0_0000_1000_0000:	M=128
0_0000_0000_0010:	M=2	0_0001_0000_0000:	M=256
0_0000_0000_0100:	M=4	0_0010_0000_0000:	M=512
0_0000_0000_1000:	M=8	0_0100_0000_0000:	M=1024
0_0000_0001_0000:	M=16	0_1000_0000_0000:	M=2048
0_0000_0010_0000:	M=32	1_0000_0000_0000:	M=4096
0_0000_0100_0000:	M=64	Other than the above	Setting is disabled.

ABR register (Alpha map Base)

Register address	**Base Address** + 0474H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			ABASE																													
R/W	R		RW																											R		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for upward compatibility; Fujitsu recommends that this register be specified by the BltCopyAltAlphaMapP, BltCopyCompAlphaMapP and DrawRectAlphaMapP commands in KOTTOS.

Bit 29-6 ABASE (Alpha-map Base address)

Sets the base address of the alpha map. Unlike existing models, bit5-3 is also enabled.

STCBR register (Stencil buffer Base)

Register address	**Base Address** + 0478H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			STCBASE																													
R/W	R		RW																											R		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bi t29-6 STCBASE (Stencil-buffer Base address)

Sets the base address of the stencil buffer when using stencil test.

COLMASK register (Color Mask)

Register address	**Base Address** + 047C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name									STLMSK									IDXMSK											RM	GM	BM	AM
R/W	R								RW									RW							R				RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1

Bit 0 AM (Alpha factor write Mask)

Sets whether to enable or disable write of the "A" component of RGBA when drawing pixel. This setting can be used for 16-bit color mode and 32-bit color mode.

0: Disable Does not write the "A" component.

1: Enable (initial value) Writes the "A" component.

Bit 1 BM (Blue factor write Mask)

Sets whether to enable or disable write of the "B" component of RGBA when drawing pixel. This setting can be used for 16-bit color mode and 32-bit color mode.

0: Disable Does not write the "B" component.

1: Enable (initial value) Writes the "B" component.

Bit 2 GM (Green factor write Mask)

Sets whether to enable or disable write of the "G" component of RGBA when drawing pixel. This setting can be used for 16-bit color mode and 32-bit color mode.

0: Disable Does not write the "G" component.

1: Enable (initial value) Writes the "G" component.

Bit 3 RM (Red factor write Mask)

Sets whether to enable or disable write of the "R" component of RGBA when drawing pixel. This setting can be used for 16-bit color mode and 32-bit color mode.

0: Disable Does not write the "R" component.

1: Enable (initial value) Writes the "R" component.

Bit 8-15 IDXMSK (Index write Mask)

Sets whether to enable or disable write to each bit when drawing pixel. This setting can be used for 8-bit color mode.

0: Disable Does not write to the relevant bit.

1: Enable (initial value) Writes to the relevant bit.

Bit 16-23 STLMSK (Stencil write Mask)

Sets whether to enable or disable write to each bit when writing stencil.

0: Disable Does not write to the relevant bit.

1: Enable (initial value) Writes to the relevant bit.

FC register (Foreground Color)

Register address	**Base Address** + 0480_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FGC8 / FGC16 / FGC32																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Interpretation of the value to be written changes due to the value of CO bit of the MDR0 register.

8-bit color mode:

Bi t7-0 FGC8 (Foreground 8bit Color)
Sets the index color (color index code) for foreground.

Bit 31-8 Unused bits

16-bit color mode (when the CO bit of the MDR0 register is "0"):

Bit 15-0 FGC16 (Foreground 16bit Color)
16-bit color data used as foreground color.
Bit 0 to 14 of an input value is written to bit1 to 15 of FC, and bit 15 of the input value is written to bit 0 of FC. For the rest, this register is the same as when the CO bit of the MDR0 register is "1".
Other bits are the same as when the CO bit of MDR0 is "1".

Bit 31-16 Unused bits

32-bit color mode (when CO bit of MDR0 is 0)

Bit 31-0 FGC32 (Foreground 32bit Color)
32-bit color code used as foreground color
Bits 0 to 24 of the input value are written to bits 8 to 31 of FC, and bits 25 to 31 of the input value are written to bits 0 to 7 of FC. Other bits are the same as when the CO bit of MDR0 is "1".

16-bit color mode (when CO bit of MDR0 is 1)

Bit 15-0 FGC16 (Foreground 16bit Color)
16-bit color code used as foreground color
At Gouraud shading, bits 1 to 15 are not used, but bit 0 is used as bit 0 of the drawing color written to the memory.

Bit 31-16 Unused bits

32-bit color mode (when the CO bit of the MDR0 is "1"):

Bit 31-0 FGC32 (Foreground 32bit Color)
32-bit color data used as foreground color
In Gouraud shading mode, bit 8 to 31 is not used, but bit 0 to 8 is used as bit 0 to 8 of the drawing color written to memory.

BC register

Register address	**Base Address** + 0484H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGC8 / BGC16 / BGC32																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8-bit color mode:

Bit 7-0 BGC8 (Background 8bit Color)

8-bit palette data used as the background color when drawing broken line and character.

Bit 14-8 Unused bits

Bit 15 BT (Background Transparency)

Sets whether to make the background color transparent.

0:	Does not make the background color transparent (initial value).	Draws the background in a color set to BC.
1:	Makes the background color transparent.	Does not draw the background.

Bit1 6-31 Unused bits

16-bit color mode (when the CO bit of the MDR0 register is "0"):

Bit 14-0 BGC16 (Background 16bit Color)

16-bit color data used as the background color when drawing broken line and character.

Bit 0 to 14 of an input value is written to bit 1 to 15 of BC.

The MSB (bit15) of an input value is used as BT, so set the bit 0 of BC using the MDR1 or MDR4 register.

Bit 15 BT (Background Transparency)

Sets whether or not to make the background color transparent.

0:	Does not make the background color transparent (initial value).	Draws the background in a color set to BC.
1:	Makes the background color transparent.	Does not draw the background.

Bit 31-16 Unused bits

32-bit color mode (when the CO bit of the MDR0 register is "0"):

Bit 30-0 BGC32 (Background 32bit Color)

32-bit color data used as the background color when drawing broken line and character.

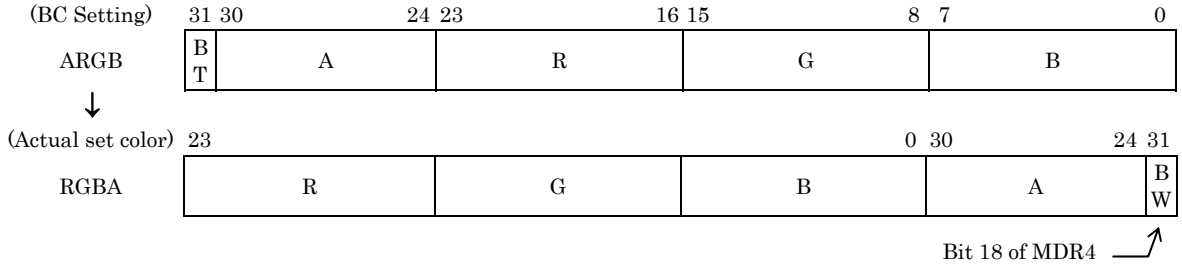
Bit 0 to 23 of an input value is written to bit 8 to 31 of BC, and bit 24 to 30 of the input value is written to bit 1 to 7 of BC. The MSB (bit31) of an input value is used as BT, and so set the bit 0 of BC using the MDR1 or MDR4 register.

Bit 31 BT (Background Transparency)

Sets whether to make the background color transparent.

0:	Does not make the background color transparent (initial value).	Draws the background in a color set to BC.
1:	Makes the background color transparent.	Does not draw the background.

Example of correspondence between BC setting and actual set color (32-bit color; CO of MDR0 = 0)



16-bit color mode (when the CO bit of the MDR0 register is "1"):

Bit 0 BT (Background Transparency)

Sets whether to make the background color transparent.

0:	Does not make the background color transparent (initial value).	Draws the background in a color set to BC.
1:	Makes the background color transparent.	Does not draw the background.

Bit 15-1 BGC16 (Background 16bit Color)

16-bit color data used as the background color when drawing broken line and character.

The LSB (bit0) of an input value is used as BT, and so set the bit 0 of BC using the MDR1 or MDR4 register.

Bit 31-16 Unused bits

32-bit color mode (when the CO bit of the MDR0 register is "1"):

Bit 0 BT (Background Transparency)

Sets whether to make the background color transparent.

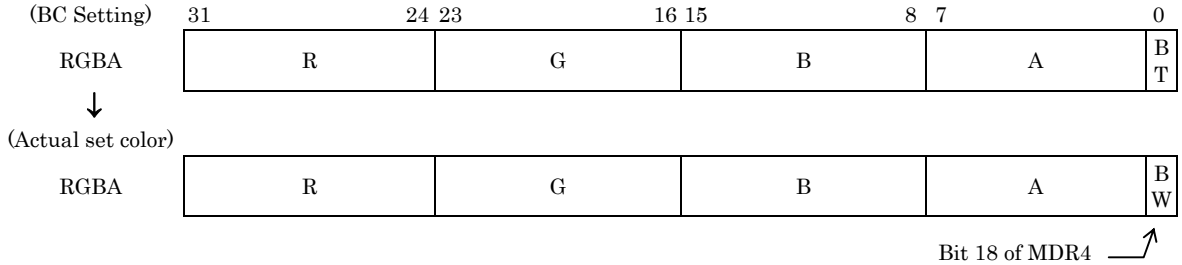
0:	Does not make the background color transparent (initial value).	Draws the background in a color set to BC.
1:	Makes the background color transparent.	Does not draw the background.

Bit 31-1 BGC32 (Background 32bit Color)

32-bit color data used as the background color when drawing broken line and character.

The LSB (bit0) of an input value is used as BT, and so set the bit 0 of BC using the MDR1 or MDR4 register.

Example of correspondence between BC setting and actual set color (32-bit color; CO of MDR0 = 1)



* The transparency of the background color is determined using all-bit matching including the A component.

ALF register

Register address	**Base Address** + 0488_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																									A							
R/W	R																								RW							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 7-0 A

Sets an alpha blending factor. 00_H represents 0%; FF_H represents 100%.

BLP register

Register address	**Base Address** + $048C_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BLP																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31-0 BLP

Sets a broken line pattern. When bit value is "1", foreground color is used for drawing; when bit value is "0", background color is used for drawing. 1 bit of a broken line pattern corresponds to 1 pixel of a line. To select the repeated length and reference direction of a broken line pattern, use BP and BPD of the MDR1 register. For bit arrangement of pattern data in each setting, see ***Table 6.3.1*** below.

Table 6.3.1 Bit Arrangement of Broken Line Pattern

MDR1		Bit arrangement of pattern
BP	BPD	
32 bits (00_H)	MSB → LSB (0_H)	Bit 31-0
	LSB → MSB (1_H)	
24 bits (01_H)	MSB → LSB (0_H)	Bit 31-8
	LSB → MSB (1_H)	Bit 23-0
16 bits (10_H)	MSB → LSB (0_H)	Bit 31-16
	LSB → MSB (1_H)	Bit15-0

ATR register (Alpha Test Reference factor)

Register address	**Base Address** + 0490_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																									REFA							
R/W	R																								RW							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit7-0 REFA

Sets an alpha factor value used as a reference for when using the alpha test function.

STCR register (Stencil Test Reference)

Register address	**Base Address** + 0498_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name									STCMASK																STCREF							
R/W	R								RW								R								RW							
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 7-0 STCREF (Stencil Reference)
Sets a stencil value used as a reference for when using the stencil test function.

Bit 16-23 STCMASK (Stencil Mask)
Both the STCREF value and the stencil buffer value are compared with STCMASK after being ANDed.

FOGCOL register (Fog Color)

Register address	**Base Address** + $049C_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FOGCOL8/16/32																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Specifies the fog target color.
Interpretation of the value written changes due to the value of CO bit of the MDR0 register.

8-bit color mode:

Bit 7-0 FOGCOL8 (Fog 8bit Color)
Color index code used as a fog color.

Bit 31-8 Unused bits

16-bit color mode (when the CO bit of the MDR0 register is “0”):

Bit 15-0 FOGCOL16 (Fog 16bit Color)
16-bit color data used as a fog color.
Bit 0 to 14 of an input value is written to bit 1 to 15 of FOGCOL, and bit15 of the input value is written to bit 0 of the FOGCOL register.
For the rest, this register is the same as when the CO bit of the MDR0 register is “1”.

Bit 31-16 Unused bits

32-bit color mode (when the CO bit of the MDR0 register is “0”):

Bit 31-0 FOGCOL32 (Fog 32bit Color)
32-bit color data used as a fog color.
Bit 0 to 24 of an input value is written to bit 8 to 31 of FOGCOL, and bit 25 to 31 of the input value is written to bit 0 to 7 of the FOGCOL register. For the rest, this register is the same as when the CO bit of the MDR0 register is “1”.

16-bit color mode (when the CO bit of the MDR0 register is “1”):

Bit 15-0 FOGCOL16 (Fog 16bit Color)
16-bit color data used as a fog color

Bit 31-16 Unused bits

32-bit color mode (when the CO bit of the MDR0 register is “1”):

Bit 31-0 FOGCOL32 (Fog 32bit Color)
32-bit color data used as a fog color.

REV register (Revision)

Register address	**Base Address** + 04B4H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	REV																															
R/W	R																															
Initial value	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

This register returns a fixed value indicating the revision of the rendering engine. It returns 24240200H.

LX0dc register

Register address	**Base Address** + 0540H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac				0											
R/W	RW				RW												RW				R											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is the mirror of the "X" component of the vertex 0 buffer.

This register is for the compatibility with Coral; Fujitsu recommends that the SetVertex command be used.

LY0dc register

Register address	**Base Address** + 0544H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac				0											
R/W	RW				RW												RW				R											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is the mirror of the "Y" component of the vertex 0 buffer.

This register is for the compatibility with Coral; Fujitsu recommends that the SetVertex command be used.

LX1dc register

Register address	**Base Address** + 0548H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac				0											
R/W	RW				RW												RW				R											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is the mirror of the "X" component of the vertex 1 buffer.

This register is for the compatibility with Coral; Fujitsu recommends that the SetVertex command be used.

LY1dc register

Register address	**Base Address** + 054CH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Sign				Int												Frac				0											
R/W	RW				RW												RW				R											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is the mirror of the "Y" component of the vertex 1 buffer.

This register is for the compatibility with Coral; Fujitsu recommends that the SetVertex command be used.

6.3.3 Vertex Reader Module

Base Address = 0002_8000_H

GCTR (Geometry Control Register)

Register address	Base Address + 0000_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name								FO			FCNT							NF	FF	FE											ST	
R/W	RX							RX	RX		RX							RX	RX	RX	RX										R	
Initial value	X	X	X	X	X	X	X	0	X	X	1	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	0	0

This is a control register in which flag and status information of KOTTOS is reflected.

Bit 24 FO (FIFO Overflow)
Indicates that a DDL-FIFO overflow occurs
0 Normal
1 Detected an overflow.

Bit 21-15 FCNT (FIFO Counter)
Indicates the number (0 to 1000000b) of empty steps of DDL-FIFO.

Bit 14 NF (FIFO Near Full)
Indicates that the number of vacant stages of DDL-FIFO is less than 8.
0 The number of empty steps of DDL-FIFO is 8 or greater.
1 The number of empty steps of DDL-FIFO is less than 8.

Bit 13 FF (FIFO Full)
Indicates that DDL-FIFO is full.
0 Not full
1 Full

Bit 12 FE (FIFO Empty)
Indicates that there is no data in DDL-FIFO.
0 Data present
1 Data absent

Bit 1, 0 ST (Status)
Indicates the status of KOTTOS.
00 Idle
01 Processing in progress
10 Reserved
11 Suspend (in the suspend status, when a Flush display list is input, this state changes to the idle state)

GMDR0 (Geometry Mode Register for Vertex)

Register address	Base Address + 0040 H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	IDFM																								CF	DF			ST	Z	C	F
R/W	RW																								RW	RW			RW	RW	RW	RW
Initial value	0																								0	0	0	0	0	0	0	0

Sets the switching between OpenGL mode and Coral mode, and the attribute and data format of vertex parameters to be input in Coral mode.

Note that when the data format is fixed-decimal format, only the range in which a value can be represented by a 32-bit fixed point (16-bit integer part, 16-bit decimal part) can be specified

Bits other than IDFM of this register are enabled when IDFM=0 (Coral mode is on). In OpenGL mode, this register is disabled and IDFOGL and IVAOGL are enabled.

When updating this register, the setting of G_MatrixSetting is updated by the F field. Issue G_MatrixSetting after updating this register.

Bit 31 IDFM (Input Data Format Mode)

Switches between OpenGL mode and Coral mode, regarding the vertex parameter attribute and the input data format.

0 Coral mode (the setting of the GMDR0 register is enabled)

1 OpenGL mode (the setting of the IDFOGL and IVAOGL registers is enabled)

Bit 7 CF (Color Format)

Specifies the color data format.

0 RGB independent / Packed RGB format

1 Reserved

Bit 6-5 DF (Data Format)

Specifies the format of vertex coordinate data and G_LoadMatrixMVP/IMV/MV.

00 Uses the floating point format (only RGB independent is possible for Color Format).

01 Uses the fixed point format (only packed RGB format is possible for Color Format).

10 Reserved

11 Packed integer format (only packed RGB format is possible for Color Format)

CF	DF	Input data format
0	00	Floating point format + RGB independent
	01	Fixed point format + Packed RGB format
	10	Reserved
	11	Packed integer format + Packed RGB format
1	00	Reserved
	01	Reserved
	10	Reserved
	11	Reserved

Bit 3 ST (texture S and T data enable)

Sets the presence or absence of the texture ST coordinate of G_Vertex, G_VertexLOG, and G_VertexNopLOG.

0 Does not use the texture ST coordinate.

1 Uses the texture ST coordinate.

Bit 2 Z (Z data enable)

Sets the presence or absence of the Z coordinate of G_Vertex, G_VertexLOG, and G_VertexNopLOG.

0 Does not use the Z coordinate.
1 Uses the Z coordinate.

Bit 1 C (Color data enable)

Sets the presence or absence of the vertex color of G_Vertex, G_VertexLOG, and G_VertexNopLOG.

0 Does not use the vertex color.
1 Uses the vertex color.

Bit 0 F (Frustum mode)

Sets projection conversion mode.

0 Performs orthogonal projection transformation.
1 Performs perspective projection transformation.

GMDR1 (Geometry Mode Register for Line)

Register address	Base Address + 0044_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												BO		EP		AA
R/W																												RW		RW		RW
Initial value																												0		0		0

Sets the geometry mode for when drawing line.

Bit 4 BO (Broken line Offset)
Sets the broken line pattern reference position.
0 Does not clear the reference position of the broken line pattern.
1 Clears the reference position of the broken line pattern.

Bit 2 EP (End Point mode)
Sets the endpoint drawing mode.
For line strip, no endpoint drawing is performed at all times.
0 Does not perform endpoint drawing.
1 Performs endpoint drawing.

Bit 0 AA (Anti Alias mode)
Sets the antialiasing mode.
0 Does not perform antialiasing.
1 Performs antialiasing.

GMDR2 (Geometry Mode Register for Triangle)

Register address	Base Address + 0048_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																														FD		CF
R/W																														RW		RW
Initial value																														0		0

Sets the geometry mode for when drawing triangle/polygon. This register is used together with the GMDR2E register. When either one of them is updated, CF and FD of the GMDR2 and GMDR2E registers are updated.
Use this register in Coral mode. In OpenGL mode, use G_PolygonSetting.

Bit 2 FD (Face Definition)
Sets the definition of face.
0 Defines face to be vertices arranged counterclockwise.
1 Defines face to be vertices arranged clockwise.

Bit 0 CF (Cull Face)
Sets the drawing mode of rear face.
0 Draws rear face.
1 Does not draw rear face (this bit value is disabled for polygon).

GMDR2E (Geometry Mode Register for Triangle Extension)

Register address	Base Address + 0050_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																						TL		SP						FD		CF
R/W	RW																					RW		RW						RW		RW
Initial value	0																					0		0						0		0

Sets the geometry mode for when drawing triangle/polygon. This register is used together with the GMDR2 register. When either one of them is updated, CF and FD of the GMDR2 and GMDR2E registers are updated.

The pixels at the border drawn when TL = 1 are generated by an approximate algorithm. Consequently, the pixel drawing position may be different from that of other sides of graphics.

Bit 10 TL (Top-Left rule mode)
Drawing algorithm
0 Applies top left rule.
1 Does not apply top left rule.

Bit 8 SP (Shadow Primitive)
Drawing mode of shadow primitive
0 Does not draw shadow primitive.
1 Draws shadow primitive.

Bit 2 FD (Face Definition)
Sets the definition of face.
0 Defines face to be vertices arranged counterclockwise.
1 Defines face to be vertices arranged in clockwise.

Bit 0 CF (Cull Face)
Sets the drawing mode of rear face.
0 Draws rear face.
1 Does not draw rear face (this bit value is disabled for polygon).

IDFOGL (Input vertex Data Format on OpenGL mode)

Register address	Base Address + 0054H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name		DFIDX												DFF				DFN				DFT				DFC				DFV		
R/W		RW												RW				RW				RW				RW				RW		
Initial value		0	0	0										1	1	0		1	1	0		1	1	0		1	1	0		1	1	0

Sets the vertex data format input in OpenGL mode.

This register is enabled when GMDR0 [31] = 1 (OpenGL mode is on). When OpenGL mode is off (or when Coral mode is on), this register is disabled and the GMDR0 register is enabled.

Note that when the data format is fixed-decimal format, only the range in which a value can be represented by a 32-bit fixed point (16-bit integer part, 16-bit decimal part) can be specified.

Bit 30-28 DFIDX (Data Format for Index)
Specifies the index format.
000 Ub (GLubyte)
001 Us (GLushort)
010 Ui GLuint)
011 Reserved

Bit 18-16 DFF (Data Format for Fog)
Specifies the fog format.
110 F (GLfloat)
111 fixed

Bit 13-11 DFN (Data Format for Normal)
Specifies the format of normal (Nx, Ny, Nz).
110 f(GLfloat)
111 fixed

Bit 10-8 DFT (Data Format for Texture)
Specifies the format of vertex texture (S, T, Q).
110 F (GLfloat)
111 fixed

Bit 6-4 DFC (Data Format for Color)
Specifies the format of vertex color (R, G, B, A).
001 Ub (GLubyte) //32-bit packed data; 8 bits for each of RGBA
110 f (GLfloat)
111 fixed

Bit 2-0 DFV (Data Format for Vertex)
Specifies the format of vertex coordinate data (X, Y, Z, W) and G_LoadMatrixMVP/IMV/MV.
110 f (GLfloat)
111 fixed

IVAOGL (Input Vertex Attribute on OpenGL mode)

Register address	Base Address + 0058H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	FO	RPC			BA	BC	FOG	N	Q1	ST1	Q0	ST0	FA	FC	W	Z
R/W																	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value																	0	0			0	0	0	0	0	0	0	0	0	0	0	0

Sets the vertex parameter attribute input in OpenGL mode.
This register is enabled when GMDR0 [31] = 1 (OpenGL mode is on). When OpenGL mode is off, Coral mode is on and this register is disabled and the GMDR0 register is enabled.

Bit 15 FO (Fog coordinate Output enable)

Sets whether to enable or disable the output fog coordinate.

This bit must be enabled when the FOG bit of this register is enabled or when the FOGZ bit of the MATRIXSET register is enabled.

0: Disable

1: Enable

Bit 14 RPC (Rendering Perspective Correction enable)

Sets whether to enable or disable perspective correction of texture mapping.

This bit must also be enabled when using texture information in which PC (Perspective Correction) is enabled in a texture information table.

0: Disable

1: Enable

Bit 11 BA (Alpha data for Background color enable)

Sets the presence or absence of the background color alpha value of the G_Vertex parameter. When BC = 0, set "0" to this bit.

0 Does not include the background color alpha value in the parameter.

1 Includes the background color alpha value in the parameter.

Bit 10 BC (RGB data for Background color enable)

Sets the presence or absence of the background color RGB value of the G_Vertex parameter.

0 Does not include the background color RGB value in the parameter.

1 Includes the background color RGB value in the parameter.

Bit 9 FOG (Fog data enable)

Sets the presence or absence of the fog coordinate of the G_Vertex parameter.

0 Does not include the fog coordinate in the parameter.

1 Includes the fog coordinate in the parameter.

Bit 8 N (Normal data enable)

Sets the presence or absence of the normal coordinate (Nx, Ny, Nz) of the G_Vertex parameter.

0 Does not include the normal coordinate in the parameter.

1 Includes the normal coordinate in the parameter.

Bit 7 Q1 (texture Q1 data enable)

Sets the presence or absence of the texture 1 component "Q" coordinate of the G_Vertex parameter. When ST1 = 0, set "0" to this bit.

0 Does not include the texture 1 component "Q" coordinate in the parameter.

1 Includes the texture 1 component "Q" coordinate in the parameter.

Bit 6 ST1 (texture S1 and T1 data enable)

Sets the presence or absence of the texture 1 component ST coordinate of the G_Vertex parameter.

0 Does not include the texture 1 component ST coordinate in the parameter.

1 Includes the texture 1 component ST coordinate in the parameter.

Bit 5 Q0 (texture Q0 data enable)

Sets the presence or absence of the texture 0 component "Q" coordinate of the G_Vertex parameter. When ST0 = 0, set "0" to this bit.

0 Does not include the texture 0 component "Q" coordinate in the parameter.

1 Includes the texture 0 component "Q" coordinate in the parameter.

Bit 4 ST0 (texture S0 and T0 data enable)

Sets the presence or absence of the texture 0 component ST coordinate of the G_Vertex parameter.

0 Does not include the texture 0 component ST coordinate in the parameter.

1 Includes the texture 0 component ST coordinate in the parameter.

Bit 3 FA (Alpha data for Foreground color enable)

Sets the presence or absence of the foreground color alpha value of the G_Vertex parameter. When FC = 0, set "0" to this bit.

0 Does not include the foreground color alpha value in the parameter.

1 Includes the foreground color alpha value in the parameter.

Bit 2 FC (RGB data for Forground color enable)

Sets the presence or absence of the foreground color RGB value of the G_Vertex parameter.

0 Does not include the foreground color RGB value in the parameter.

1 Includes the foreground color RGB value in the parameter.

Bit 1 W (W data enable)

Sets the presence or absence of "W" of the G_Vertex parameter.

0 Does not include "W" in the parameter.

1 Includes "W" in the parameter.

Bit0 Z (Z data enable)

Sets the presence or absence of the "Z" coordinate of the G_Vertex parameter.

0 Does not include the "Z" coordinate in the parameter.

1 Includes the "Z" coordinate in the parameter.

[Caution]

- When drawing points, set the BC and BA bits to "0".

VRINT (Interrupt Register of Vertex Reader)

Register address	Base Address + 0060_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																															DFIN	GFIFO
R/W																															RW	RW
Initial value																															0	0

Indicates the normal interrupt factor of the vertex reader.

Writing “0” clears normal interrupt factor. Writing “1” is ignored.

Bit 1 DFIN (Draw Finish interrupt)

Indicates that a drawing end wait is detected after detection of the G_Interrupt command.

0 Indicates the status in which no drawing end wait interrupt occurs.

1 Indicates the status in which a drawing end wait interrupt occurs.

Bit 0 GFIFO (Geometry FIFO interrupt)

Indicates that the G_Interrupt command is detected.

0 Indicates the state in which no G_Interrupt command detection interrupt occurs.

1 Indicates the state in which a G_Interrupt command detection interrupt occurs.

VRINTM (Interrupt Mask Register of Vertex Reader)

Register address	Base Address + 0060_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																															MDFIN	MGFIFO
R/W																															RW	RW
Initial value																															1	1

Masks the normal interrupt factor of the Vertex Reader.

Bit 1 MDFIN (Draw Finish interrupt Mask)

Masks the interrupt of the DFIN bit of the VRINT register.

0 Does not mask the interrupt of the DFIN bit of the VRINT register.

1 Masks the interrupt of the DFIN bit of VRINT.

Bit 0 MGFIFO (Geometry FIFO interrupt Mask)

Masks the interrupt of the GFIFO bit of the VRINT register.

0 Does not mask the interrupt of the GFIFO bit of the VRINT register.

1 Masks the interrupt of the GFIFO bit of the VRINT register.

VRERR (Error Register of Vertex Reader)

Register address	Base Address + 0068_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																																CERR
R/W																																RW
Initial value																																0

Indicates the abnormal interrupt factor of the vertex reader

Writing “0” clears the abnormal interrupt factor. Writing “1” is ignored.

Bit 0 CERR (Display List Command Error)

Indicates that an undefined display list command is detected.

0 Indicates the status in which no display list command error occurs.

1 Indicates the status in which a display list command error occurs.

VRERRM (Error Mask Register of Vertex Reader)

Register address	Base Address + $006C_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																																MCERR
R/W																																RW
Initial value																																1

Masks the abnormal interrupt factor of the vertex reader.

Bit 0 MCERR (Display List Command Error Mask)

Masks the interrupt of the CERR bit of the VRERR register.

0 Does not mask the interrupt of the CERR bit of the VRERR register.

1 Masks the interrupt of the CERR bit of the VRERR register.

DDLFIFO_STATUS

Register address	Base Address + 0070H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name								FO						NF	FF	FE											FCNT					
R/W								R						R	R	R											R					
Initial value	X	X	X	X	X	X	X	0	X	X	X	X	X	0	0	1	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0

Bit 24 FO (FIFO Overflow)
Indicates that a DDL-FIFO overflow occurs. During normal operation, no FIFO overflow occurs.
0 Normal
1 Overflow is detected.

Bit 18 NF (FIFO Near Full)
Indicates that the number of empty steps of DDL-FIFO is less than 8.
0 The number of empty steps of DDL-FIFO is 8 or greater.
1 The number of empty steps of DDL-FIFO is less than 8.

Bit 17 FF (FIFO Full)
Indicates that DDL-FIFO is full. During normal operation, FIFO does not become full.
0 Not full
1 Full

Bit 16 FE (FIFO Empty)
Indicates that there is no data in DDL-FIFO.
0 There is data.
1 There is no data.

Bit 6-0 FCNT (FIFO Counter)
Indicates the number (0 to 1000000b) of empty steps of DDL-FIFO.

C_OXYO

Register address	Base Address + 0100H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	OXYO																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used by the DC coordinate XY offset of shadow primitive. This register is updated by the display list OverlapXYOfft.

C_OZORG

Register address	Base Address + 0108H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	OZORG																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used by the DC coordinate Z offset of body primitive.
This register is updated by OverlapZOfft.

C_OZNTL

Register address	Base Address + 010C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	OZNTL																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used by the DC coordinate Z offset of top-left not applicable primitive.

This register is updated by OverlapZOfft.

C_OZSH

Register address	Base Address + 0114$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	OZSH																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used by the DC coordinate Z offset of shadow primitive.

This register is updated by OverlapZOfft.

C_MDR1

Register address	Base Address + 0200$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR1																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR1 register of body primitive.

The entity of this register is the register S_MDR1. When either one of these registers is updated, the result is reflected in both of them.

This register is updated by SetRegister and SetModeRegister.

C_MDR2

Register address	Base Address + 020C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR2																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR2 register of body primitive.

The entity of this register is the register S_MDR2. When either one of these registers is updated, the result is reflected in both of them.

This register is updated by SetRegister and SetModeRegister.

C_MDR2S

Register address	Base Address + 0210_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR2																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR2 register of shadow primitive.

This register is updated by SetModeRegister.

C_MDR2TL

Register address	Base Address + 0214_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR2																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR2 register of top-left not applicable primitive.

This register is updated by SetModeRegister.

C_FCC

Register address	Base Address + 0230_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FC																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store FC of body primitive.

This register is updated by SetColorRegister. (Unlike MDR*, this register is not updated by SetRegister.)

C_BCC

Register address	Base Address + 0234_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BC																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store BC of body primitive.

This register is updated by SetColorRegister. (Unlike MDR*, this register is not updated by SetRegister.)

C_FCSC

Register address	Base Address + 0238H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FC																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store FC of shadow primitive.

This register is updated by SetColorRegister.

C_LGA

Register address	Base Address + 0248 H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	C_LGA																															
R/W	RW																															
Register address	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the addresses where logs are written by G_VertexNopLOG and G_VertexLOG.

This register is updated by DC_LogOutAddr.

IDXBRCOORD

Register address	Base Address + 0300H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDECOORD

Register address	Base Address + 0304H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRCOLF

Register address	Base Address + 0308H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDECOLF

Register address	Base Address + 030C$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRCOLB

Register address	Base Address + 0310$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDECOLB

Register address	Base Address + 0314$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRNORM

Register address	Base Address + 0318$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDENORM

Register address	Base Address + 031C$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRTEX0

Register address	Base Address + 0320$_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDETEX0

Register address	BaseAddress + 0324H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRTEX1

Register address	Base Address + 0328H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDETEX1

Register address	Base Address + 032CH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

IDXBRF

Register address	Base Address + 0330h																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Base Address																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexBaseAddr.

IDXSTRIDEF

Register address	Base Address + 0334H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Stride																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated by SetIndexStride.

DDLFIFOG (Direct Displaylist FIFO of Geometry)

Register address	Base Address + 400H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DDLFIFOG																															
R/W	W																															
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

This register is an FIFO register for direct DL transfer.

S_MDR0

Register address	Base Address + 0420_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR0																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR0 register.
This register is updated by SetRegister.

S_MDR1

Register address	Base Address + 0424_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR1																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR1 register.
The entity of this register is the C_MDR1 register. When either one of these registers is updated, the result is reflected in both of them.
This register is updated by SetRegister and SetModeRegister.

S_MDR2

Register address	Base Address + 0428_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR2																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR2 register.
The entity of this register is the C_MDR2 register. When either one of these registers is updated, the result is reflected in both of them.
This register is updated by SetRegister and SetModeRegister.

S_MDR4

Register address	Base Address + 0430_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDR4																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the MDR4 register.

The entity of this register is the C_MDR4 register. When either one of these registers is updated, the result is reflected in both of them.
This register is updated by SetRegister.

S_FBR

Register address	Base Address + 0440H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FBR																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the FBR register. This register is updated by SetRegister.

S_XRR

Register address	Base Address + 0444H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	XRR																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is for the compatibility with Coral. This register is used to store the XRR register. This register is updated by SetRegister

FR_ST

Register address	Base Address + 0520H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																																ST
R/W																																RW
Initial value																																1

This register is used to manage the status of the geometry processor. This register is updated from the idle status ⇒ the busy status by KOTTOS; from the busy status ⇒ the idle status by the geometry processor.

Bit 0 ST (Geometry processor Status)
Indicates the status of the geometry processor.
0 Idle
1 Busy

CMDERR

Register address	Base Address + 0524h																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CMDERR																															
R/W	R																															
Initial value	Undefined																															

When a command error occurred, this register holds the value of the display list where the error occurred.

DL_CNT

Register address	Base Address + 0538h
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	DL_CNT
R/W	R
Initial value	Undefined

Each time one word is input to the display list, the value of this register is incremented by 1. When a command error occurs (VRERR = 1), the register stops incrementing. When the register reaches the maximum value, input of the next word returns the register to “0”.

6.3.4 VL Engine Module

Base Address = 0003_0000_H

MVP, MV*, IMV**

Register address	Base Address + 0100_H to $01A0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_LoadMatrixMVP, G_LoadMatrixMV, or G_LoadMatrixIMV.

Bit 0-31 VALUE

Value of each element of the matrix. This value is stored in the 32-bit single precision floating point format.

VV_XMIN, VV_XMAX, VV_YMIN, VV_YMAX

Register address	Base Address + $01C0_H$ to $01CC_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_ViewVolumeXYClip.

Bit 0-32 VALUE

Each NDC coordinate where a view volume is set. The coordinate is stored in the 32-bit single precision floating point format.

VV_ZMIN, VV_ZMAX

Register address	Base Address + $01D0_H$ to $01D4_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_ViewVolumeZClip.

Bit 0-32 VALUE

Each NDC coordinate where a view volume is set. The coordinate is stored in the 32-bit single precision floating point format.

VV_WMIN

Register address	Base Address + $01D8_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_ViewVolumeWClip.

Bit 0-32 VALUE

Each NDC coordinate where a view volume is set. The coordinate is stored in the 32-bit single precision floating point format.

VP_XScale, VP_XOffset, VP_YScale, VP_Yoffset

Register address	Base Address + 01E0H to 01ECH																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_Viewport.

Bit 0-32 VALUE

Scale and offset (NDC coordinate) of view port transformation. They are stored in the 32-bit single precision floating point format.

DR_ZScale, DR_Zoffset

Register address	Base Address + 01F0H to 01F4H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_DepthRange.

Bit 0-32 VALUE

Scale and offset (NDC coordinate) of the depth range (Z coordinate). They are stored in the 32-bit single precision floating point format.

LG_AmbR, LG_AmbG, LG_AmbB

Register address	Base Address + 0200H to 0208H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_GlobalLight.

Bit 0-32 VALUE

Color component of global ambient light. The color component is stored in the 32-bit single precision floating point format.

*L[0-7]_**

Register address	Base Address + 0220H to 0478H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_Light.

Bit 0-32 VALUE

Color component of each light source 0 to 7. The color component is stored in the 32-bit single precision floating point format.

*MF_**

Register address	Base Address + 0500_H to 0528_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_Material as the material of the face.

Bit 0-32 VALUE

Each reflection component of face material. The reflection component is stored in the 32-bit single precision floating point format.

*MB_**

Register address	Base Address + 0550_H to 0578_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VALUE																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_Material as the material of the rear face.

Bit 0-32 VALUE

Each reflection component of rear face material. The reflection component is stored in the 32-bit single precision floating point format.

VL_VERTEXSET

Register address	Base Address + $05A0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																FO	—			BA	BC	FOG	N	Q1	ST1	Q0	ST0	FA	FC	W	Z
R/W	R0																RW	R0			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sets whether to enable or disable input vertex elements. This register stores a value set by the GMDR0 register (Coral mode), the IVAOGL register (OpenGL mode), or G_VertexSetting.

Bit 0 Z (Z enable)
Enables/disables Z element
0: Disables
1: Enables

Bit 1 W (W enable)
Enables/disables W element
0: Disables
1: Enables

Bit 2 FC (Front Color enable)
Enables/disables RGB element of front face color
0: Disables
1: Enables

Bit 3 FA (Front color alpha enable)
Enables/disables A element of front face color
0: Disables
1: Enables

Bit 4 ST0 (Texture 0 ST enable)
Enables/disables ST element of texture 0
0: Disables
1: Enables

Bit 5 Q0 (Texture 0 SQenable)
Enables/disables Q element of texture 0
0: Disables
1: Enables

Bit 6 ST1 (Texture 1 ST enable)
Enable/disable of ST element of texture 1
0: Disable
1: Enable

Bit 7 Q1 (Texture 1 SQenable)
Enables/disables Q element of texture 1
0: Disables
1: Enables

Bit 8 N (Normal vector enable)
Enables/disables normal vector element (Nx, Ny, Nz)
0: Disables
1: Enables

Bit 9 FOG (Fog coordinate enable)
Enables/disables fog coordinate
0: Disables
1: Enables

Bit 10 BC (Back Color enable)
Enables/disables RGB element of rear face color
0: Disables
1: Enables

Bit 11 BA (Back color alpha enable)
Enables/disables A element of rear face color
0: Disables
1: Enables

Bit 15 FO (Fog coordinate Output enable)
Enables/disables output fog coordinate
This bit must be enabled when the FOG bit of this register is enabled or when the FOGZ bit of the MATRIXSET register is enabled.
0: Disables
1: Enables

MATRIXSET

Register address	Base Address + $05B0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																											PROJ	IMV	NVS	NVN	FOGZ
R/W	R0																											RW	RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by the GMDR0 register or G_MatrixSetting.

Bit 0 FOGZ (Fog eye coordinate Z Calculattion Enable)
Enable/disable of fog coordinate operation processing
0: Disables (sets fog of the input vertex as it is)
1: Enables (sets Z value of the eye coordinate as fog)

Bit 1 NVN (Normal Vector Normalize enable)
Enables/disables normal vector normalization processing
0: Disables
1: Enables

Bit 2 NVS (Normal Vector Scaling enable)
Enables/disables normal vector scaling processing
0: Disables
1: Enables (uses NVSF value to perform scaling)

Bit 3 IMV (Inverted Model View transfromation enable)
Enables/disables MV inverse matrix transformation processing of normal vector
0: Disables
1: Enables

Bit 4 PROJ (Projection enable)
Enables/disables perspective transformation processing
0: Disables
1: Enables

NVSF

Register address	Base Address + 05C0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	F																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_NormalScale.

Bit 0-32 F

This value is referenced as scaling factor "f" when normal vector scaling is enabled (NVS of MATRIX_SET = 1).
This value is stored in the 32-bit single precision floating point format.

LIGHTSET

Register address	Base Address + 05D0H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	—																LE7	LE6	LE5	LE4	LE3	LE2	LE1	LE0	—		INIC	SIDE	—			LEN
R/W	R0																RW	RW	RW	RW	RW	RW	RW	RW	R0		RW	RW	R0			RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by G_LightSetting.

Bit 0 LEN (Lighting Enable)

Sets whether to enable or disable light source processing. To disable this bit, set "0" to LE0 to LE7.

0: Disables

1: Enables

Bit 4 SIDE (Lighting Side)

Sets whether to perform light source processing for single face or both faces.

0: Single face

1: Both faces

Bit 5 INIC (Initial Color)

Sets whether to use the vertex color as the initial value for light source processing.

0: Does not use the vertex color as the initial value.

1: Uses the vertex color as the initial value.

Bit 8 LE0 (Light Enable for Light No.0)

Sets whether to enable or disable light source 0.

0: Disables

1: Enables

Bit 9 LE1 (Light Enable for Light No.1)

Sets whether to enable or disable light source 1.

0: Disables

1: Enables

Bit 10 LE2 (Light Enable for Light No.2)

Sets whether to enable or disable light source 2.

0: Disables

1: Enables

Bit 11 LE3 (Light Enable for Light No.3)

Sets whether to enable or disable light source 3.

0: Disables

1: Enables

Bit 12 LE4 (Light Enable for Light No.4)
Sets whether to enable or disable light source 4.
0: Disables
1: Enables

Bit 13 LE5 (Light Enable for Light No.5)
Sets whether to enable or disable light source 5.
0: Disables
1: Enables

Bit 14 LE6 (Light Enable for Light No.6)
Sets whether to enable or disable light source 6.
0: Disables
1: Enables

Bit 15 LE7 (Light Enable for Light No.7)
Sets whether to enable or disable light source 7.
0: Disables
1: Enables

CLIPSET

Register address	Base Address + $05E0_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	–																															CE
R/W	R0																															R W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores the setting of the presence or absence of clipping specified by G_Begin.

Bit 0 CE (Clipping Enable)
Enables/disables clipping processing
0: Disables (sets the clip flag forcibly to “0”)
1: Enables

6.3.5 Primitive Engine Module

Base Address = 0004_0000$_H$

PO_FACTOR

Register address	**Base Address** + 024$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Factor																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores the value of a factor component set by G_PolygonOffset.

	Factor
Bit 31-0	Stores a PolygonOffset parameter, factor. This value is stored in the 32-bit single precision floating point format.

PO_UNITS

Register address	**Base Address** + 028$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Units																															
R/W	RW																															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores the value of the units component set by G_PolygonOffset.

	Units
Bit 31-0	Stores a parameter of PolygonOffset, **`factor`**. This value is stored in the 32-bit single precision floating point format.

Note: The units here is the value units multiplied by r; and the multiplication result is set by the driver.

LINE_SET_REG

Register address	**Base Address** + $02C_H$																																
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												BLPC		ENDP		AA
R/W	R0																											RW		RW		RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by the GMDR1 register or G_LineSetting.

Bit 0 AA
Antialias
0: Disable
1: Enable

Bit 2 ENDP
Whether to draw end point or not
0: Disable
1: Enable

Bit 4 BLPC
Whether to initialize broken line pointer or not
0: Disable
1: Enable

POLYGON_SET_REG

Register address	**Base Address** + 030_H																																
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	–		POMB		POMF		POBF	POBL	POBP	POFF	POFL	POFP	–	CLD	CLB	CLF
R/W	R0																		RW		RW		RW	RW	RW	RW	RW	RW		RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by the GMDR2/GMDR2E register or G_PolygonSetting.

Bit 0 CLF
Front face culling setting
0: Does not perform culling.
1: Performs culling.

Bit 1 CLB
Back face culling setting
0: Does not perform culling.
1: Performs culling.

Bit 2 CLD
Culling direction
0: The face with the counterclockwise culling direction is the front face.
1: The face with the clockwise culling direction is the front face.

Bit 4 POFP
Whether to enable or disable front face PolygonOffset when point mode enabled
0: Disable
1: Enable

Bit 5 POFL
Whether to enable or disable front face PolygonOffset when line mode enabled
0: Disable
1: Enable

Bit 6 POFF
Whether to enable or disable front face PolygonOffset when fill mode enabled
0: Disable
1: Enable

Bit 7 POBP
Whether to enable or disable back face PolygonOffset when point mode enabled
0: Disable
1: Enable

Bit 8 POBL
Whether to enable or disable back face PolygonOffset when line mode enabled
0: Disable
1: Enable

Bit 9 POBF
Whether to enable or disable back face PolygonOffset when fill mode enabled
0: Disable
1: Enable

Bit 11-10 POMF
Specification of front face PolygonMode
00: POINT
01: LINE
10: FILL

[Caution]

- When the graphics is point or line type, this register is not referenced.
- When the graphics is a polygon, PolygonMode and PolygonOffset cannot apply, so disable the POFP, POFL, POFF, POBP, POBL and POBF bits, and set the POMF and POMB bits to FILL (10b).

PR_VERTEXSET

Register address	**Base Address** + 034H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	FOPC				BA	BC	FOG	N	Q1	ST1	Q0	ST0	FA	FC	W	Z
R/W																	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Stores a value set by the GMDR0 register (Coral mode), the IVAOGL register (OpenGL mode), or G_VertexSetting.

Bit 0 Z

Presence or absence of the input Z component

0: Absence

1: Presence

Bit 1 W

Presence or absence of the input W

0: Absence

1: Presence

Bit 2 FC

Presence or absence of the input front face color initial value RGB component (Rf, Gf, Bf)

0: Absence

1: Presence

Bit 3 FA

Presence or absence of the input front face color initial value A component (Af)

0: Absence

1: Presence

Bit 4 ST0

Presence or absence of the ST component (S0, T0) of input texture 0

0: Absence

1: Presence

Bit 5 Q0

Presence or absence of the Q component (Q0) of input texture 0

0: Absence

1: Presence

Bit 6 ST1

Presence or absence of the ST component (S1, T1) of input texture 1

0: Absence

1: Presence

Bit 7 Q1

Presence or absence of the Q component (Q1) of input texture 1

0: Absence

1: Presence

Bit 8 N

Presence or absence of the input normal vector component (Nx, Ny, Nz)

0: Absence

1: Presence

Bit 9 FOG

Presence or absence of the input fog coordinate (fog)

00: Absence

01: Presence

Bit 10 BC

Presence or absence of the input rear face color initial value RGB component (Rb,Gb,Bb)

0: Absence

1: Presence

Bit 11 BA

Presence or absence of the input rear face color initial value A component (Ab)

0: Absence

1: Presence

Bit 14 RPC

Presence or absence of perspective correction by the rendering engine

0: Absence

1: Presence

When (ST0|ST1)==0, this bit becomes “0”, too.

Bit 15 FO

Presence or absence of the fog coordinate by the input from VL Engine

0: Absence

1: Presence

DC-OFFSET-PX

Register address	**Base Address** + 05C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOPX															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting value (X coordinate) of DC-OFFSET for point

This register sets values used when inverting the sign of the X coordinate of the Point and when adding the offset.

When the value of this register is other than “0”, perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the X coordinate for a point

0: Does not perform sign inversion

1: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

DC-OFFSET-PY

Register address	**Base Address** + 060$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOPY															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting value of the DC-OFFSET for a point (Y coordinate)

This register sets the value used when inverting the sign of the X coordinate of the Point and when adding the offset.

When the value of this register is other than “0”, perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the Y coordinate for a point

0: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

DC-OFFSET-LX

Register address	**Base Address** + 064_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOLX															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting value of the DC-OFFSET for a line (X coordinate)

This register sets the value used when inverting the sign of the LINE and when adding the offset.

When the value of this register is other than "0", perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the X coordinate for a line

0: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

DC-OFFSET-LY

Register address	**Base Address** + 068_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOLY															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Set value of the DC-OFFSET for a line (Y coordinate)

This register sets values used when inverting the sign of the LINE and when adding the offset.

When the value of this register is other than "0", perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the Y coordinate for a line

0: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

DC-OFFSET-TX

Register address	**Base Address** + 06C$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOTX															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Set value of the DC-OFFSET for a triangle (X coordinate)

This register sets values used when inverting the sign of the Triangle and when adding the offset.

When the value of this register is other than “0”, perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the X coordinate for a triangle

0: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

DC-OFFSET-TY

Register address	**Base Address** + 070$_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TOTY															S	Int												Frac			
R/W	RW															RW	RW												RW			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Set value of the DC-OFFSET for a triangle (Y coordinate)

This register sets values used when inverting the sign of the Triangle and when adding the offset.

When the value of this register is other than “0”, perform sign inversion first and then add the offset.

Bit 31 TOPX

Sign inversion bit of the Y coordinate for TRIANGLE

0: Does not perform sign inversion

1: Performs sign inversion

Bit 16 S

Sign bit of Fixed

Bit 15-4 Int

Integer part of Fixed

Bit 3-0 Frac

Decimal part of Fixed

6.4 Display List

6.4.1 Overview

Display list is a collection of display list commands, parameters and pattern data. The display list commands stored in FIFO are processed sequentially.

Display lists are transferred to the display list FIFO.

They can be transferred to the display list FIFO in either of the following ways:

(1) Write to FIFO from outside via PCI

(2) Write to FIFO from external memory via the rendering unit

In this manual, the method above *(1)* is called "direct DL transfer", and the method above *(2)* "indirect DL transfer". The method above *(2)* has a mode called "index reference DL transfer".

6.4.2 Display List Transfer Mode

Direct DL transfer

It writes a display list consisting of commands and data to the display list FIFO of KOTTOS via PCI.

Displaylist Command-1
Data1-1
Data1-2
Data1-3
Displaylist Command-2
Data2-1
Data2-2
Data2-3
…

Display list

Indirect DL transfer

When it detects an indirect DL start command in a display list transferred using direct DL transfer, it transfers the display list for the specified word count from the specified address. When the transfer is completed, it returns to direct DL transfer mode.

Displaylist Command-1
Data1-1
Data1-2
Data1-3
G_IndirectDL
Base Address
Word count
Displaylist Command-3
Data3-1
Data3-2
Data3-3
…

Index reference DL transfer

When executing a command indicating index reference, specify a coordinate data that is usually specified as a display list, using a vertex number instead of coordinate data itself. "Vertex number" is an index from the external memory base address storing vertex data specified separately.

Displaylist Command-1
Data1-1
Data1-2
Data1-3
G_VertexIndex
Index Count
Vertex Number-1
Vertex Number-2
Vertex Number-3
…
Displaylist Command-2
Data2-1
Data2-2
Data2-3
…

6.4.3 Header Format

The display list header format is shown below.

Table 6.4.1 Display List Header Format

31 24	23 16	15 0
Type	Command	Parameter

Table 6.4.2 Description of Each Field of Header

Type	Display list type
Command	Command
Parameter	Field where various conditions that cannot be specified using Command above are specified. For example, specify vertex numbers to be specified for a Vertex type command for this Parameter.

6.4.4 Geometry Display List

A list of geometry processing display lists

Table 6.4.3 shows a list of display lists used for geometry processing.

Table 6.4.3 Geometry Processing Display List (1)

Type	Code	Description	Command	Description
SaveRestoreReg	$1C_H$	Saves/restores registers.	00_H	Save
			01_H	Restore
G_Nop	20_H	No operation	00_H	-
G_Begin	21_H	Specifies primitive type and performs pre-processing.	10_H	Points
			11_H	Lines
			12_H	Polygon
			13_H	Triangles
			14_H	Quads
			15_H	Line_Strip
			17_H	Triangle_Strip
			18_H	Triangle_Fan
			19_H	Quads_Strip
			30_H	nclip_Points
			31_H	nclip_Lines
			32_H	nclip_Polygon
			33_H	nclip_Triangles
			34_H	nclip_Quads
			35_H	nclip_Line_Strip
			37_H	nclip_Triangle_Strip
			38_H	nclip_Triangle_Fan
			39_H	nclip_Quads_Strip
G_End	23_H	Ends primitives.	00_H	-
G_Vertex	30_H	Sets vertex parameters and draws.	00_H	-
G_IndirectDL	34_H	Reads display lists from external memory.	00_H	-
G_VertexIndex	38_H	Reads vertex data from external memory by specifying indices.	00_H	DrawElement
			01_H	DrawArray
G_BitBlt	$3C_H$	Packs BitBlt type commands	00_H	-

Table 6.4.4 Geometry Processing Display List (2)

Type	Code	Description	Command	Description
G_Viewport	41_H	Scales the screen coordinate (X, Y) and sets the origin offset.	00_H	-
G_DepthRange	42_H	Scales the screen coordinate (Z) and sets the origin offset.	00_H	-
G_LoadMatrixMVP	43_H	Sets the ModelViewProjection transformation matrix.	00_H	RowCol
			01_H	ColRow
G_ViewVolumeXYClip	44_H	Sets the view volume clip boundary values X, Y.	00_H	-
G_ViewVolumeZClip	45_H	Sets the view volume clip boundary value “Z”.	00_H	-
G_ViewVolumeWClip	46_H	Sets the view volume clip boundary value “W”.	00_H	-
G_LoadMatrixMV	49_H	Sets the ModelView transformation matrix.	00_H	-
G_LoadMatrixIMV	$4A_H$	Sets the ModelView inverse transformation matrix.	00_H	RowCol
			001_H	ColRow
G_LineSetting	60_H	Sets the line drawing effect.	00_H	-
G_PolygonSetting	64_H	Sets the polygon drawing mode.	00_H	-
G_PolygonOffset	65_H	Sets each PolygonOffset offset component.	00_H	-
G_VertexSetting	68_H	Sets which element is valid as vertex element.	00_H	
G_GlobalLight	86_H	Sets the global ambient light source.	00_H	-
G_Light	87_H	Sets the light source (ambient light, diffuse light, and light source position).	00_H	-
G_Material	88_H	Sets the material characteristics (ambient light, diffuse light, and emitted light)	00_H	-
G_MatrixSetting	96_H	Sets the matrix calculation method.	00_H	-
G_NormalScale	97_H	Sets the normal vector scaling ratio.	00_H	-
G_LightSetting	98_H	Sets the light source processing.	00_H	-

Table 6.4.5 Geometry Processing Display List (3)

Type	Code	Description	Command	Description
SetIndexBaseAddr	$D0_H$	Sets base address of each element in index DL mode.	00_H	COORD
			01_H	COLF
			02_H	COLB
			03_H	NORM
			04_H	TEX0
			05_H	TEX1
			06_H	F
SetIndexStride	$D1_H$	Sets stride of each element in index DL mode.	00_H	COORD
			01_H	COLF
			02_H	COLB
			03_H	NORM
			04_H	TEX0
			05_H	TEX1
			06_H	F
G_Sync	DC_H	Waits for event synchronization.	00_H	-
G_Interrupt	DD_H	Reports interrupt.	00_H	-

Table 6.4.6 Geometry Processing Display List for Compatibility with Coral

Type	Code	Description	Command	
G_VertexLOG	32_H	Sets vertex parameters and draws. Outputs device coordinates.	00_H	-
G_VertexNopLOG	33_H	Outputs only device coordinates.	00_H	-
SetModeRegister	$C0_H$	Sets drawing extended mode registers.	00_H	MDR1
			01_H	MDR2
			03_H	MDR2S
			07_H	MDR2TL
SetGModeRegister	$C1_H$	Sets geometry extended mode registers.	00_H	GMDR0
			01_H	GMDR1
			02_H	GMDR2
			20_H	GMDR2E
			80_H	IDFOGL
			90_H	IVAOGL
OverlapXYOfft	$C8_H$	Sets the XY offset when shadowing.	00_H	ShadowXY
			01_H	ShadowXYcompsition
OverlapZOfft	$C9_H$	Sets the Z offset when drawing 2D in a mode where shadowing, bordering, and the top left rule are not applied.	00_H	Origin
			01_H	NonTopLeft
			03_H	Shadow
			07_H	Packed_ONBS
DC_LogOutAddr	CC_H	Sets the starting address for device coordinate output.	00_H	-
SetColorRegister	CE_H	Sets the body, shadow and edge colors.	00_H	ForeColor
			01_H	BackColor
			02_H	ForeColorShadow
			03_H	BackColorShadow

Table 6.4.7 Coral-compatible Display List Unnecessary for KOTTOS

Type	Code	Description	Command	Description
G_BeginCont	22_H	Interpreted as G_Begin	Omitted	-
G_Init	40_H	Interpreted as NOP.	00_H	-
G_BeginE	$E1_H$	Interpreted as G_Begin.	Omitted	-
G_BeginECont	$E2_H$	Interpreted as G_Begin.	Omitted	-
G_EndE	$E3_H$	Interpreted as G_End.	Omitted	-
SetLVertex2i	72_H	Operates as SetVertex2i.	Omitted	-
SetLVertex2iP	73_H	Operates as SetVertex2iP.	Omitted	-

Description of geometry display list

SaveRestoreReg

[Format]

31 28	24 23 20	16 15 12 8 4 0
SaveRestoreReg ($1C_H$)	Save (00_H)	Reserved
MEMADDR		
REGADDR		
REGCOUNT		

SaveRestoreReg ($1C_H$)	Restore (01_H)	Reserved
MEMADDR		

[Parameter description]

Field name	Contents	Valid range
MEMADDR	Save/restore destination memory address	00000000_H to $3FFFFFF8_H$ (only 64-bit boundary)
REGADDR	Save register start address	0_4000_H to 1_001F_H (only addresses with register)
REGCOUNT	Save register count	1 to 65535

[Processing]

This command saves the contents of the register into the PCI memory space specified using MEMADDR. REGADDR is specified using the address parenthesized in the register list.

When saving the contents of the register, both the save register start address (REGADDR) and the save register count (REGCOUNT) are automatically saved. At restore, the register is restored only by specifying MEMADDR.

Operation is not guaranteed when a nonexistent register is saved or restored.

Memory storage format:

63 48	47 32	31 15 0
REGADDR		REGCOUNT
Reg value 1		Reg value 0
…		
Reg value m-1		Reg value m-2

G_Nop

[Format]

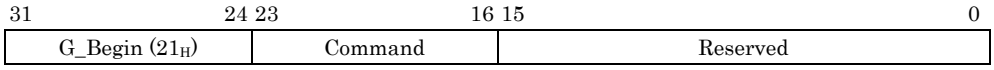
31 24	23 16	15 0
G_Nop (20_H)	Reserved	Reserved

[Processing]

This command performs nothing.
This command can be used to align 64-bit boundary when a display list is written to FIFO by a 64-bit access.

G_Begin

[Format]



[Description of command]

Command		Clipping	Description
Points	10 H	Performed	Point
Lines	11H		Independent straight line
Polygon	12H		Polygon
Triangles	13H		Independent triangle
Quads	14H		Independent quadrangle
Line_Strip	15H		Continuous straight line
Triangle_Strip	17H		Continuous triangle (strip format)
Triangle_Fan	18H		Continuous triangle (fan format)
Quads_Strip	19H		Continuous quadrangle (strip format)
nclip_Points	30H	Not performed	Point
nclip_Lines	31H		Independent straight line
nclip_Polygon	32H		Polygon
nclip_Triangles	33H		Independent triangle
nclip_Quads	34H		Independent quadrangle
nclip_Line_Strip	35H		Continuous straight line
nclip_Triangle_Strip	37H		Continuous triangle (strip format)
nclip_Triangle_Fan	38H		Continuous triangle (fan format)
nclip_Quads_Strip	39H		Continuous quadrangle (strip format)

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_05E0H	CLIPSET	Updates this register by enabling/disabling clipping

[Processing]

This command sets the internal status to drawing mode of the graphics specified for Command. After this, this command processes the graphics as the one where vertex data to be input is set, until G_End is input.

The display list that can be input between G_Begin and G_End is as follows.

- G_Vertex
- G_VertexIndex
- G_Material

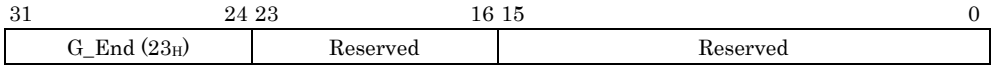
[Caution]

When setting the command to Points(10_H) or nclip_Points(30_H), note the following points:

- Disable the vertex elements BC and BA set by IVAOGL or G_VertexSetting.
- When the vertex element FC has been enabled by IVAOGL or G_VertexSetting, the value of the FC register (0002_0480_H) is updated with the vertex color. For point drawing, when the A component of the 16-bit pixel is greater than a value close to 0.502, it is converted to 1; when the A component of the 16-bit pixel is smaller than a value close to 0.502, it is converted to 0. The operation method on the A component of the 16-bit pixel for point drawing is different from that for straight-line drawing and triangle drawing.

G_End

[Format]



[Processing]

This command ends the graphics processing started by G_Begin.

G_Vertex

(1) Input data format = Coral mode (GMDR0[31] = 0)

[Format]

Vertex parameters that can be combined by the GMDR0 register are input.

(1-1) Data format is the floating point format (GMDR0[7:5] = 000_B)

31 – 24	23 – 16	15	0	
G_Vertex (30_H)	Reserved	EF	Reserved	Always exists.
X.float				Always exists.
Y.float				Always exists.
Z.float				Dependent on GMDR0[2].
R.float				Dependent on GMDR0[1].
G.float				Dependent on GMDR0[1].
B.float				Dependent on GMDR0[1].
S.float				Dependent on GMDR0[3].
T.float				Dependent on GMDR0[3].

(1-2) Data format is the fixed point format (GMDR0[7:5] = 001_B)

31 – 24	23 – 16	15	0	
G_Vertex (30_H)	Reserved	EF	Reserved	Always exists.
X.fixed				Always exists.
Y.fixed				Always exists.
Z.fixed				Dependent on GMDR0[2].
A.int	R.int	G.int	B.int	Dependent on GMDR0[1].
S.fixed				Dependent on GMDR0[3].
T.fixed				Dependent on GMDR0[3].

(1-3) Data format is the packed integer format (GMDR0[7:5] = 011_B)

31 – 24	23 – 16	15	0	
G_Vertex (30_H)	Reserved	EF	Reserved	Always exists
Y.int		X.int		Always exists
Z.fixed				Dependent on GMDR0[2].
A.int	R.int	G.int	B.int	Dependent on GMDR0[1].
S.fixed				Dependent on GMDR0[3].
T.fixed				Dependent on GMDR0[3].

(2) Input data format = OpenGL mode (GMDR0[31] = 1)

[Format]

Vertex parameters that can be combined by the IDFOGL and IVAOGL registers are input.

(2-1) Data format is the floating point format or the fixed point format (IDFOGL[6:4] ≠ 001_B)

31 24	23 16	15	0	
G_Vertex (30_H)	Reserved	EF	Reserved	Always exists.
X.float/fixed				Always exists.
Y.float/fixed				Always exists.
Z.float/fixed				Dependent on IVAOGL[0].
W.float/fixed				Dependent on IVAOGL[1].
Rf.float/fixed				Dependent on IVAOGL[2].
Gf.float/fixed				Dependent on IVAOGL[2].
Bf.float/fixed				Dependent on IVAOGL[2].
Af.float/fixed				Dependent on IVAOGL[3].
S0.float/fixed				Dependent on IVAOGL[4].
T0.float/fixed				Dependent on IVAOGL[4].
Q0.float/fixed				Dependent on IVAOGL[5].
S1.float/fixed				Dependent on IVAOGL[6].
T1.float/fixed				Dependent on IVAOGL[6].
Q1.float/fixed				Dependent on IVAOGL[7].
Nx.float/fixed				Dependent on IVAOGL[8].
Ny.float/fixed				Dependent on IVAOGL[8].
Nz.float/fixed				Dependent on IVAOGL[8].
FOG.float/fixed				Dependent on IVAOGL[9].
Rb.float/fixed				Dependent on IVAOGL[10].
Gb.float/fixed				Dependent on IVAOGL[10].
Bb.float/fixed				Dependent on IVAOGL[10].
Ab.float/fixed				Dependent on IVAOGL[11].

(2-1) Data format is the floating point format or the fixed point format. RGBA is an unsigned byte (IDFOGL[6:4] = 001_B)

31 24	23 16	15	0	
G_Vertex (30_H)	Reserved	EF	Reserved	Always input.
X.float/fixed				Always input.
Y.float/fixed				Always input.
Z.float/fixed				Dependent on IVAOGL[0].
W.float/fixed				Dependent on IVAOGL[1].
Rf.ubyte	Gf.ubyte	Bf.ubyte	Af.ubyte	Dependent on IVAOGL[2][3].
S0.float/fixed				Dependent on IVAOGL[4].
T0.float/fixed				Dependent on IVAOGL[4].
Q0.float/fixed				Dependent on IVAOGL[5].
S1.float/fixed				Dependent on IVAOGL[6].
T1.float/fixed				Dependent on IVAOGL[6].
Q1.float/fixed				Dependent on IVAOGL[7].
Nx.float/fixed				Dependent on IVAOGL[8].
Ny.float/fixed				Dependent on IVAOGL[8].
Nz.float/fixed				Dependent on IVAOGL[8].
FOG.float/fixed				Dependent on IVAOGL[9].
Rb.ubyte	Gb.ubyte	Bb.ubyte	Ab.ubyte	Dependent on IVAOGL[10][11].

[Processing]

This command performs coordinate transformation and lighting processing for the input vertex coordinate. This command draws the processing result as the graphics specified by G_Begin.

Processing by Quads is the same as processing by Triangle_Fan. Processing by Quad_Strip is the same as processing by Triangle_Strip.

EF field:

The EF field is referenced by geometry processing firmware when the Polygon Mode function is set to LINE (POMF==LINE or POMB==LINE of G_PolygonSetting). When the EF bit of vertex of the start point for the side of triangle is set to "1", this command draws the side as LINE. When the QUADS or QUAD_STRIP is executed, this field is used to detect edges.

Usually set "0" to the field.

G_IndirectDL

[Format]

31 24	23 16	15 0
G_IndirectDL (34_H)	Reserved	Reserved
Base Address		
Word Count		

[Parameter description]

Field name	Contents	Valid range
Base Address	Display list start PCI address	00000000_H to $FFFFFFF8_H$ (only 64-bit boundary)

[Setting register affecting processing]

Module	Register address	Register name	Description
Vertex Reader	0002_806C_H	VRERRM	Error type interrupt mask

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8068_H	VRERR	CERR Update

[Processing]

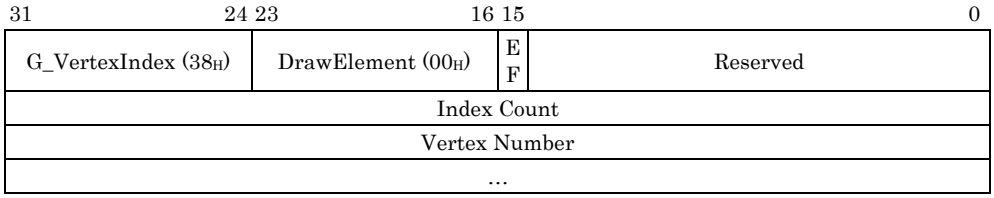
This command reads and executes the display list specified for Word Count from the address specified by Base Address. After the execution, the command re-executes the display list from the DDL-FIFO. This execution image is like that of the subroutine.

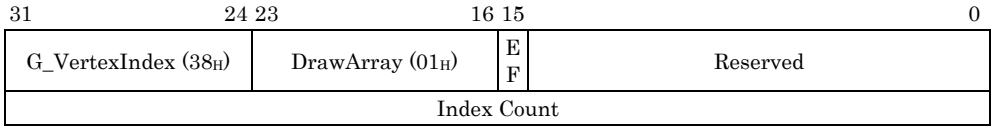
[Caution]

A nested structure where the display list to be transferred using G_IndirectDL includes G_IndirectDL, is inhibited. When a nested structure is detected, the system treats it as a command error, and sets the CERR bit of the VRERR register to 1.

G_VertexIndex

[Format]





[Description of parameter]

Field name	Content	Effective range
Index Count	Vertex number count	00000001_H to $3FFFFFFF_H$
Vertex Number	Vertex number (only for DrawElement)	00000000_H to $3FFFFFFF_H$
EF	Edge detection flag when drawing QUADS and QUAD_STRIP	0, 1

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFIDX field as well as DFV, C, T, N and F fields affect processing.
	0002_8300_H	IDXBRCOORD	Affects the coordinate data read address.
	0002_8304_H	IDXSTRIDECOORD	
	0002_8308_H	IDXBRCOLF	Affects the front face color read address.
	0002_830C_H	IDXSTRIDECOLF	
	0002_8310_H	IDXBRCOLB	Affects the rear face color read address.
	0002_8314_H	IDXSTRIDECOLB	
	0002_8318_H	IDXBRNORM	Affects the normal vector read address.
	0002_831C_H	IDXSTRIDENORM	
	0002_8320_H	IDXBRTEX0	Affects the texture 0 read address.
	0002_8324_H	IDXSTRIDETEX0	
	0002_8328_H	IDXBRTEX1	Affects the texture 1 read address.
	0002_832C_H	IDXSTRIDETEX1	
	0002_8330_H	IDXBRF	Affects the fog read address.
	0002_8334_H	IDXSTRIDEF	

The index format is determined by the DFIDX field of the IDFOGL register.

The vertex data format is determined by the DFV, C, T, N and F fields of the IDFOGL register.

[Processing]

DrawElement

DrawElement reads vertex data for Index Count from memory, assuming the subsequent display list as the index of vertex data. Specify the starting address storing vertex data by SetIndexBaseAddress, and specify the stride of vertex data by SetIndexStride.

The index format is determined by the DFIDX field of the IDFOGL register.

The vertex data format is determined by the DFV, C, T, N and F fields of the IDFOGL register.

DrawArray

Unlike DrawElement, DrawArray consists of two words, having no word indicating Vertex Number. The index of vertex data always starts at vertex number "0", and is automatically incremented by "1" internally in the geometry engine. How to reference vertex data from memory is the same as that of DrawElement.

EF field:

The EF field is used to detect edges when executing QUADS and QUAD_STRIP. This flag is referenced by geometry processing firmware when the Polygon Mode function is enabled (POMF! or POMB! of G_PolygonSetting is FILL). Usually set "0" to the field.

Assignment of vertex data

Real address in memory associated with each vertex number is calculated as follows:

Real address = Base address + Stride * Vertex number (0, 1, 2,…) + Offset (X, Y, Z,...)

The byte/halfword ordering in the Vertex Number field is little endian.

Offset is executed as follows according to the attribute setting of the IVAOGL register.

Base=IDXBRCOORD

Parameter	Z=0, W=0	Z=1, W=0	Z=1, W=1
X	0_H	0_H	0_H
Y	4_H	4_H	4_H
Z	-	8_H	8_H
W	-	-	C_H

Base=IDXBRCOLF

Parameter	FC=0,FA=0	FC=1,FA=0	FC=1,FA=1
R	-	0_H	0_H
G	-	1_H (Byte)/4_H (Word)	1_H (Byte)/4_H (Word)
B	-	2_H (Byte)/8_H (Word)	2_H (Byte)/8_H (Word)
A	-	-	3_H (Byte)/C_H (Word)

Note: The offset of COLF is byte (8 bits) or word (32 bits) depending on the data format of IDFOGL.

Base=IDXBRCOLB

Parameter	BC = 0,BA = 0	BC = 1,BA = 0	BC = 1,BA = 1
R	-	0_H	0_H
G	-	1_H(Byte)/4_H(Word)	1H(Byte)/4_H(Word)
B	-	2_H(Byte)/8_H(Word)	2H(Byte)/8_H(Word)
A	-	-	3H(Byte)/C_H(Word)

Note: The offset of COLB is byte (8 bits) or word (32 bits) depending on the data format of IDFOGL.

Base=IDXBRNORM

Parameter	N = 0	N = 1	
Nx	-	0_H	
Ny	-	4_H	
Nz	-	8_H	

Base=IDXBRTEX0

Parameter	ST0 = 0,Q0 = 0	ST0 = 1,Q0 = 0	ST0 = 1,Q0 = 1
S0	-	0_H	0_H
T0	-	4_H	4_H
R0	-	-	-
Q0	-	-	C_H

Base=IDXBRTEX1

Parameter	ST1 = 0,Q1 = 0	ST1 = 1,Q1 = 0	ST1 = 1,Q1 = 1
S1	-	0_H	0_H
T1	-	4_H	4_H
R1	-	-	-
Q1	-	-	C_H

Base=IDXBRF

Parameter	F = 0	F = 1	
F	-	0_H	

[Caution]

1. G_VertexIndex is supported only in OpenGL mode. In Coral input mode, it is treated as a command error and no G_VertexIndex processing is performed.
2. Vertex data sent by G_VertexIndex is aligned on a 32-bit address boundary.

G_BitBlt

[Format]

31 24	23 16	15 12	11 8	7 0
G_BitBlt ($3C_H$)	Reserved	SZP	DSP	Reserved
Length				
X				
Y				
Z				
W				
BLT displaylist				
…				

[Description of parameter]

Field name	Content	Effective range
SZP	BLT displaylist size specification word offset	See the table below.
DSP	BLT displaylist drawing coordinate specification word offset	See the table below.
Length	BLT displaylist word count. The length from G_BitBlt to W is not included.	00000000_H to $3FFFFFFF_H$
X	"X" component of model coordinate	Floating point value
Y	"Y" component of model coordinate	Floating point value
Z	"Z" component of model coordinate	Floating point value
W	"W" component of model coordinate	Floating point value

[Processing]

This command executes the BLT command of the rendering display list to draw graphics to a position centered on the coordinate after the MVP-transformation. The center is calculated by dividing the "size by 2", and the fraction is rounded off. Various BLT commands can be used as "BLT displaylist". Set SZP, DSP and Length as shown in the table below, depending on the BLT command used. The destination coordinate (DRXs, DRYs) of the BLT displaylist is replaced with the post-MVP-transformation coordinate, and so does not affect the drawing.

BLT Type	Command	SZP	DSP	Length
DrawRectP	-	2	1	3
DrawRectAlphaMapP	-	3	2	4
DrawBitmapP	-	2	1	Dependent on Count.
DrawBitmapLargeP	-	3	2	Dependent on Count.
BltCopyP	-	3	2	4
BltCopyAlternateP	-	7	6	8
BltCopyAltAlphaMapP	Normal	6	5	7
	ABR	7	6	8
BltCopyCompressedP	-	5	4	6
BltCopyCompAlphaMapP	-	4	3	5

G_Viewport

[Format]

31 24	23 16	15 0
G_Viewport (41_H)	Reserved	Reserved
X_Scaling.float/fixed		
X_Offset.float/fixed		
Y_Scaling.float/fixed		
Y_Offset.float/fixed		

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_01E0_H	VP_XScale	Updated by X_Scaling.
	0003_01E4_H	VP_XOffset	Updated by X_Offset
	0003_01E8_H	VP_YScale	Updated by Y_Scaling.
	0003_01EC_H	VP_YOffset	Updated by Y_Offset.

[Processing]

This command updates the above update registers and sets the update result as the view port transformation setting of the XY coordinate. When the value is a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_DepthRange

[Format]

31 24	23 16	15 0
G_DepthRange (42_H)	Reserved	Reserved
Z_Scaling.float/fixed		
Z_Offset.float/fixed		

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_01F0_H	DR_ZScale	Updated by Z_Scaling.
	0003_01F4_H	DR_ZOffset	Updated by Z_Offset.

[Processing]

This command updates the above update registers and sets the update result as the view port transformation setting of the Z coordinate.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_LoadMatrixMVP

[Format]

31 24	23 16	15 0
G_LoadMatrixMVP(43_H)	RowCol (00_H)	Reserved
m00.float/fixed		
m10.float/fixed		
m20.float/fixed		
m30.float/fixed		
m01.float/fixed		
m11.float/fixed		
m21.float/fixed		
m31.float/fixed		
m02.float/fixed		
m12.float/fixed		
m22.float/fixed		
m32.float/fixed		
m03.float/fixed		
m13.float/fixed		
m23.float/fixed		
m33.float/fixed		

31 24	23 16	15 0
G_LoadMatrixMVP(43_H)	ColRow (01_H)	Reserved
m00.float/fixed		
m01.float/fixed		
m02.float/fixed		
m03.float/fixed		
m10.float/fixed		
m11.float/fixed		
m12.float/fixed		
m13.float/fixed		
m20.float/fixed		
m21.float/fixed		
m22.float/fixed		
m23.float/fixed		
m30.float/fixed		
m31.float/fixed		
m32.float/fixed		
m33.float/fixed		

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFV field affects processing

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_0100H	MVP00	Updated by m00.
	0003_0104H	MVP10	Updated by m10.
	0003_0108H	MVP20	Updated by m20.
	0003_010CH	MVP30	Updated by m30.
	0003_0110H	MVP01	Updated by m01.
	0003_0114H	MVP11	Updated by m11.
	0003_0118H	MVP21	Updated by m21.
	0003_011CH	MVP31	Updated by m31.
	0003_0120H	MVP02	Updated by m02.
	0003_0124H	MVP12	Updated by m12.
	0003_0128H	MVP22	Updated by m22.
	0003_012CH	MVP32	Updated by m32.
	0003_0130H	MVP03	Updated by m03.
	0003_0134H	MVP13	Updated by m13.
	0003_0138H	MVP23	Updated by m23.
	0003_013CH	MVP33	Updated by m33.

[Processing]

This command updates the above update registers and sets the update result as the MVP transformation matrix. When the value is in a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_ViewVolumeXYClip

[Format]

31	24 23	16 15	0
G_ViewVolumeXYClip(44_H)	Reserved	Reserved	
XMIN.float/fixed			
XMAX.float/fixed			
YMIN.float/fixed			
YMAX.float/fixed			

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_01C0_H	VV_XMIN	Updated by XMIN.
	0003_01C4_H	VV_XMAX	Updated by XMAX.
	0003_01C8_H	VV_YMIN	Updated by YMIN.
	0003_01CC_H	VV_YMAX	Updated by YMAX.

[Processing]

This command updates the above update registers and sets the update result as the setting of the XY component of view volume clipping. When the value is a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

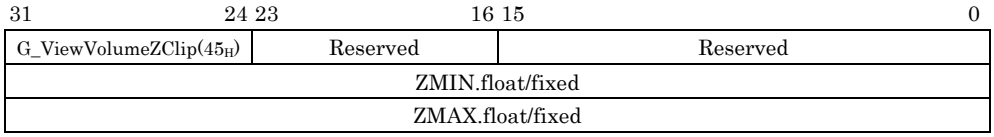
Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_ViewVolumeZClip

[Format]



[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040H	GMDR0	IDFM and D fields affect processing.
	0002_8054H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_01D0H	VV_ZMIN	Updated by ZMIN.
	0003_01D4H	VV_ZMAX	Updated by ZMAX.

[Processing]

This command updates the above update registers and sets the update result as the setting of the Z component of view volume clipping. When the value is a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

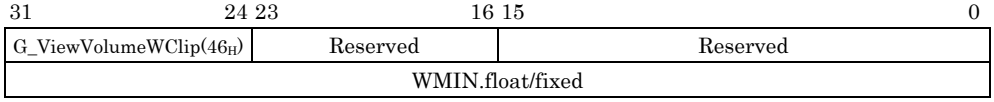
Register	Field	Value	Description
IDFOGL	DFV	111B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01B	fixed: fixed point
		11B	fixed: fixed point
		Others	Float: floating point

G_ViewVolumeWClip

[Format]



[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040H	GMDR0	IDFM and D fields affect processing.
	0002_8054H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_01D8H	VV_WMIN	Updated by WMIN.

[Processing]

This command updates the above update register and sets the update result as the setting of the W component of view volume clipping. When the value is a "fixed" point value, it is converted to a "floating" point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_LoadMatrixMV

[Format]

31 24	23 16	15 0
G_LoadMatrixMV (49_H)	Reserved	Reserved
m02.float/fixed		
m12.float/fixed		
m22.float/fixed		
m32.float/fixed		

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	IDFM and D fields affect processing.
	0002_8054_H	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_0160_H	MVP02	Updated by m02.
	0003_0164_H	MVP12	Updated by m12
	0003_0168_H	MVP22	Updated by m22.
	0003_016C_H	MVP32	Updated by m32.

[Processing]

This command updates the above update registers and sets the update result as the coordinate transformation matrix for fog coordinate calculation by the eye coordinate Z component. When the value is a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_LoadMatrixIMV

[Format]

31 24	23 16	15 0
G_LoadMatrixIMV(4A$_H$)	RowCol (00$_H$)	Reserved
m00.float/fixed		
m10.float/fixed		
m20.float/fixed		
m01.float/fixed		
m11.float/fixed		
m21.float/fixed		
m02.float/fixed		
m12.float/fixed		
m22.float/fixed		

31 24	23 16	15 0
G_LoadMatrixIMV(4A$_H$)	ColRow (01$_H$)	Reserved
m00.float/fixed		
m01.float/fixed		
m02.float/fixed		
m10.float/fixed		
m11.float/fixed		
m12.float/fixed		
m20.float/fixed		
m21.float/fixed		
m22.float/fixed		

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040$_H$	GMDR0	IDFM and D fields affect processing
	0002_8054$_H$	IDFOGL	DFV field affects processing.

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_0180$_H$	IMV00	Updated by m00.
	0003_0184$_H$	IMV10	Updated by m10.
	0003_0188$_H$	IMV20	Updated by m20.
	0003_018C$_H$	IMV01	Updated by m01.
	0003_0190$_H$	IMV11	Updated by m11.
	0003_0194$_H$	IMV21	Updated by m21.
	0003_019C$_H$	IMV02	Updated by m02.
	0003_01A0$_H$	IMV12	Updated by m12.
	0003_01A4$_H$	IMV22	Updated by m22.
	0003_01A8$_H$	IMV03	Updated by m03.
	0003_01AC$_H$	IMV13	Updated by m13.
	0003_01B0$_H$	IMV23	Updated by m23.

[Processing]

This command updates the above update registers and sets the update result as the normal vector coordinate transformation matrix. When the value is a “fixed” point value, it is converted to a “floating” point value and then stored.

Data type in the parameter section following the header is interpreted as follows.

(1) When input vertex = OpenGL mode (GMDR0[31]=1)

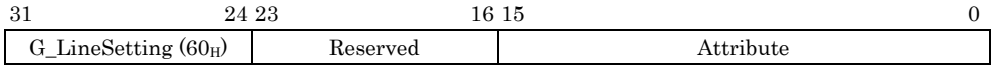
Register	Field	Value	Description
IDFOGL	DFV	111_B	fixed: fixed point
		Others	Float: floating point

(2) When input vertex = Coral mode (GMDR0[31]=0)

Register	Field	Value	Description
GMDR0	D	01_B	fixed: fixed point
		11_B	fixed: fixed point
		Others	Float: floating point

G_LineSetting

[Format]



Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	—											BLPC	—	ENDP	—	AA

[Description of parameter]

Field name	Content	Effective range
AA	Antialias	0: Absent, 1: Present
ENDP	Presence/absence of endpoint drawing	0: Absent, 1: Present
BLPC	Presence/absence of initialization of broken line pointer	0: Absent, 1: Present

[Update register]

Module	Register address	Register name	Description
Primitive Engine	0004_002CH	LINE_SET_REG	Updated by Attribute.

[Processing]

This command updates the above update register for setting of straight line. LINE_SET_REG is referenced when Lines, Line_Strip, nclip_Lines, or nclip_Line_Strip is specified by G_Begin.

G_PolygonSetting

[Format]

31	24 23	16 15	0
G_PolygonSetting (64_H)	Reserved	Attribute	

Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	—		POMB		POMF		POBF	POBL	POBP	POFF	POFL	POFP	—	CLD	CLB	CLF

[Description of parameter]

Field name	Content	Effective range
CLF	Front face culling setting	0: Does not perform culling 1: Performs culling
CLB	Rear face culling setting	0: Does not perform culling 1: Performs culling
CLD	Culling direction	0: Counterclockwise for front face, 1: Clockwise for front face
POFP	Enable/disable of front face PolygonOffset in POINT mode	0: Disable, 1: Enable
POFL	Enable/disable of front face PolygonOffset in LINE mode	0: Disable, 1: Enable
POFF	Enable/disable of front face PolygonOffset in FILL mode	0: Disable, 1: Enable
POBP	Enable/disable of rear face PolygonOffset in POINT mode	0: Disable, 1: Enable
POBL	Enable/disable of rear face PolygonOffset in LINE mode	0: Disable, 1: Enable
POBF	Enable/disable of rear face PolygonOffset in FILL mode	0: Disable, 1: Enable
POMF	Specification of front face PolygonMode	00_B:POINT, 01_B:LINE, 10_B:FILL
POMB	Specification of rear face PolygonMode	00_B:POINT, 01_B:LINE, 10_B:FILL

[Update register]

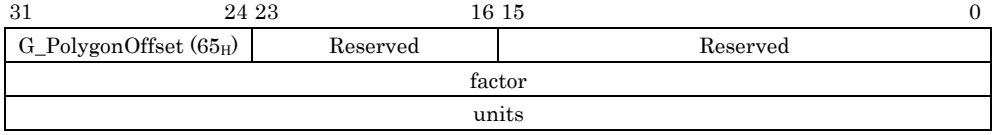
Module	Register address	Register name	Description
Primitive Engine	0004_0030_H	POLYGON_SET_REG	Updated by Attribute.

[Processing]

This command updates the above update register for setting of culling and PolygonMode. POLYGON_SET_REG is referenced when graphics other than Point, Lines, Line_Strip, nclip_Point, nclip_Lines and nclip_Line_Strip is specified by G_Begin.

G_PolygonOffset

[Format]



[Update register]

Module	Register address	Register name	Description
Primitive Engine	0004_0024H	PO_FACTOR	Updated by factor
	0004_0028H	PO_UNITS	Updated by units.

[Processing]

This command updates the above update registers for setting of PolygonOffset. PO_FACTOR and PO_UNITS are referenced when the setting of PolygonOffset is enabled by PolygonSetting.

G_VertexSetting

[Format]

31 — 24	23 — 16	15 — 0
G_VertexSetting (68_H)	Reserved	Attribute

Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FO	RPC	—		BA	BC	FOG	N	Q1	ST1	Q0	ST0	FA	FC	W	Z

[Description of parameter]

Field name	Content	Effective range
Z	Enable/disable of "Z" component operation	0: Disable, 1: Enable
W	Enable/disable of "W" component operation	0: Disable, 1: Enable
FC	Enable/disable of front face color initial value RGB component (Rf, Gf, Bf) calculation	0: Disable, 1: Enable
FA	Enable/disable of front face color initial value "A" component (Af) calculation	0: Disable, 1: Enable
ST0	Enable/disable of texture 0 "ST" component (S0, T0) calculation	0: Disable, 1: Enable
Q0	Enable/disable of texture 0 "Q" component (Q0) calculation	0: Disable, 1: Enable
ST1	Enable/disable of texture 1 "ST" component (S1, T1) calculation	0: Disable, 1: Enable
Q1	Enable/disable of texture 1 "Q" component (Q1) calculation	0: Disable, 1: Enable
N	Enable/disable of normal vector component (Nx, Ny, Nz) calculation	0: Disable, 1: Enable
FOG	Enable/disable of fog coordinate (Fog) calculation	0: Disable, 1: Enable
BC	Enable/disable of rear face color initial value RGB component (Rb, Gb, Bb) calculation	0: Disable, 1: Enable
BA	Enable/disable of rear face color initial value A component (Ab) calculation	0: Disable, 1: Enable
RPC	Enable/disable of texture perspective correction calculation	0: Disable, 1: Enable
FO	Enable/disable of fog coordinate calculation	0: Disable, 1: Enable

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_05A0_H	VL_VERTEXSET	Updated by Attribute
Primitive Engine	0004_0034_H	PR_VERTEXSET	Updated by Attribute.

[Processing]

This command switches between enable and disable of vertex elements in coordinate transformation and lighting processing, irrespective of enable or disable of vertex elements which the user inputs. For example, to perform lighting processing using only material and without vertex element color, specify "enable" of color element by G_VertexSetting after disabling the color element by IVAOGL.

RPC must be set to "enable" when using a texture ID whose perspective correction is enabled. FO must be set to "enable" when the FOG field is enabled or when FOGZ is enabled by G_MatrixSetting.

[Caution]

Even when G_VertexSetting is input, enable/disable of each element of input vertex does not change, and the content of GMDR0, IDFOGL and IVAOGL also does not change.

Setting GMDR0 and IVAOGL updates VL_VERTEXSET and PR_VERTEXSET. When performing G_VertexSetting setting different from setting of GMDR0 and IVAOGL, be sure to issue G_VertexSetting after GMDR0 and IVAOGL have been set.

When drawing points, set the BC and BA bits to “0”.

G_GlobalLight

[Format]

31	24 23	16 15	0
G_GlobalLight (86_H)	Reserved	Reserved	
AmbR			
AmbG			
AmbB			
AmbA			

[Description of parameter]

Field name	Content	Effective range
AmbR	"R" component of global ambient light	Floating point value
AmbG	"G" component of global ambient light	Floating point value
AmbB	"B" component of global ambient light	Floating point value
AmbA	"A" component of global ambient light	Floating point value

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_0200_H	LG_AmbR	Updated by AmbR.
	0004_0204_H	LG_AmbG	Updated by AmbG.
	0004_0208_H	LG_AmbB	Updated by AmbB.

[Processing]

This command updates the above update registers for setting of global ambient light. LG_AmbR, LG_AmbG and LG_AmbB are used for lighting processing. LG_AmbA is not used for lighting processing, and so does not affect processing.

G_Light

[Format]

31 24	23 16	15 0	
G_Light (87_H)	Reserved	Reserved	ID
AmbR			
AmbG			
AmbB			
AmbA			
DiffR			
DiffG			
DiffB			
DiffA			
PosX			
PosY			
PosZ			

[Description of parameter]

Field name	Content	Effective range
ID	Light source ID	0 to 7
AmbR	Light source (ambient light) “R” component	Floating point value
AmbG	Light source (ambient light) “G” component	Floating point value
AmbB	Light source (ambient light) “B” component	Floating point value
AmbA	Light source (ambient light) “A” component	Floating point value
DiffR	Light source (diffuse light) “R” component	Floating point value
DiffG	Light source (diffuse light) “G” component	Floating point value
DiffB	Light source (diffuse light) “B” component	Floating point value
DiffA	Light source (diffuse light) “A” component	Floating point value
PosX	Light source position “X” component	Floating point value
PosY	Light source position “Y” component	Floating point value
PosZ	Light source position “Z” component	Floating point value

[Update register]

Module	Register address	Register name	Description	Description
VL Engine	0003_0220H	L0_AmbR	When ID=0	Updated by AmbR.
	0003_0224H	L0_AmbG		Updated by AmbG.
	0003_0228H	L0_AmbB		Updated by AmbB.
	0003_0230H	L0_DiffR		Updated by DiffR.
	0003_0234H	L0_DiffG		Updated by DiffG.
	0003_0238H	L0_DiffB		Updated by DiffB.
	0003_0240H	L0_PosX		Updated by PosX.
	0003_0244H	L0_PosY		Updated by PosY.
	0003_0248H	L0_PosZ		Updated by PosZ.
	0003_0270H	L1_AmbR	When ID=1	Updated by AmbR.
	…	…		Omitted (same as for L0)
	0003_0298H	L1_PosZ		Updated by PosZ.
	…	…	ID=2 to 5	Omitted
	0003_400H	L6_AmbR	When ID=6	Updated by AmbR.
	…	…		Omitted (same as for L0)
	0003_0428H	L6_PosZ		Updated by PosZ.
	0003_0450H	L7_AmbR	When ID=7	Updated by AmbR.
	…	…		Omitted (same as for L0)
	0003_0478H	L7_PosZ		Updated by PosZ.

[Processing]

This command updates the above update registers for setting of light source. Set up to eight light sources, light source 0 to light source 7 by specifying ID.

AmbA and DiffA are not used for lighting processing, and so does not affect processing.

G_Material

[Format]

31 24	23 16	15 0
G_Material (88_H)	Reserved	Atribute
R		
G		
B		
A		

Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	—					TYP			—						BAK	FRT

[Parameter description]

Field name	Contents	Valid range
FRT	Selects whether or not to apply material setting to front face	0: Does not apply material setting. 1: Applies material setting.
BAK	Selects whether or not to apply material setting to back face	0: Does not apply material setting. 1: Applies material setting.
TYP	Selects light source type for set material	001b: Sets material to only Ambient 010b: Sets material to only Diffuse 011b: Sets material to both Ambient and Diffuse 100b: Sets material to Emission
R	R component of material	Float value
G	G component of material	Float value
B	B component of material	Float value

[Update register]

Module	Register address	Register name	Description	
VL Engine	0003_0500H	MF_AmbR	FRT=1 and (TYP=001b or 011b)	Updated with R
	0003_0504H	MF_AmbG		Updated with G
	0003_0508H	MF_AmbB		Updated with B
	0003_0510H	MF_DiffR	When FRT = 1 and (TYP = 010b or 011b)	Updated with R
	0003_0514H	MF_DiffG		Updated with G
	0003_0518H	MF_DiffB		Updated with B
	0003_051CH	MF_DiffA		Updated with A
	0003_0520H	MF_EmisR	When FRT = 1 and TYP = 100b	Updated with R
	0003_0524H	MF_EmisG		Updated with G
	0003_0528H	MF_EmisB		Updated with B

Module	Register address	Register name	Description	
VL Engine	0003_0550H	MB_AmbR	BAK=1 and (TYP=001b Or 011)	Updated with R
	0003_0554H	MB_AmbG		Updated with G
	0003_0558H	MB_AmbB		Updated with B
	0003_0560H	MB_DiffR	When BAK = 1 and (TYP = 010b or 011b)	Updated with R
	0003_0564H	MB_DiffG		Updated with G
	0003_0568H	MB_DiffB		Updated with B
	0003_056CH	MB_DiffA		Updated with A
	0003_0570H	MB_EmisR	When BAK = 1 and TYP = 100b	Updated with R
	0003_0574H	MB_EmisG		Updated with G
	0003_0578H	MB_EmisB		Updated with B

[Processing]

This command updates the above update registers as the material setting. The material is referenced when lighting has been enabled.

G_MatrixSetting

[Format]

31 24	23 16	15 0
G_MatixSetting (96_H)	Reserved	Attribute

Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	–											PROJ	IMV	NVS	NVN	FOGZ

[Description of parameter]

Field name	Content	Effective range
FOGZ	Enable/disable of fog eye coordinate calculation	0: Disable, 1: Enable
NVN	Enable/disable of normalization of normal vector	0: Disable, 1: Enable
NVS	Enable/disable of scaling of normal vector	0: Disable, 1: Enable
IMV	Enable/disable of MV inverse conversion	0: Disable, 1: Enable
PROJ	Enable/disable of perspective transformation	0: Disable, 1: Enable

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_05B0_H	MATRIXSET	Updated by Attribute.

[Processing]

This command updates the above update register.

[Caution]

When NVN is enabled, normal vector scaling (NVS) is not performed.

G_NormalScale

[Format]

31 24	23 16	15 0
G_NormalScale (97_H)	Reserved	Reserved
F		

[Update register]

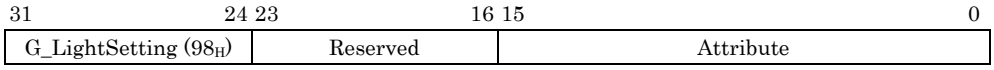
Module	Register address	Register name	Description
VL Engine	0003_05C0_H	NVSF	Updated by F

[Processing]

This command updates the above update register as the scaling ratio of normal vector. NVSF is referenced when NVN of MATRIXSET is disabled and NVS is enabled.

G_LightSetting

[Format]



Attribute:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LE7	LE6	LE5	LE4	LE3	LE2	LE1	LE0	—		INIC	SIDE	—			LEN

[Description of parameter]

Field name	Content	Effective range
LEN	Enable/disable of light source processing	0: Disable, 1: Enable
SIDE	Selects between lighting of single side and lighting of both sides.	0: Lighting of single side, 1: Lighting of both sides
INIC	Presence/absence of vertex color lighting initial value	0: Absent, 1: Present
LE0	Enable/disable of light source ID 0	0: Disable, 1: Enable
LE1	Enable/disable of light source ID 1	0: Disable, 1: Enable
LE2	Enable/disable of light source ID 2	0: Disable, 1: Enable
LE3	Enable/disable of light source ID 3	0: Disable, 1: Enable
LE4	Enable/disable of light source ID 4	0: Disable, 1: Enable
LE5	Enable/disable of light source ID 5	0: Disable, 1: Enable
LE6	Enable/disable of light source ID 6	0: Disable, 1: Enable
LE7	Enable/disable of light source ID 7	0: Disable, 1: Enable

[Update register]

Module	Register address	Register name	Description
VL Engine	0003_05D0H	LIGHTSET	Updated by Attribute.

[Processing]

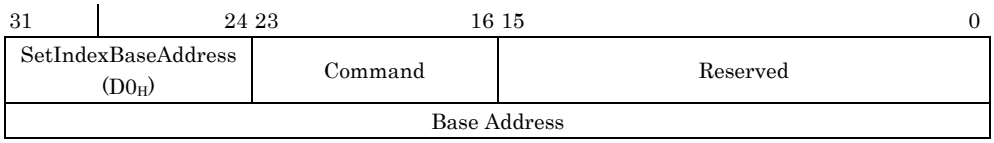
This command updates the above update register. The lighting initial value referenced by INIC=1 is the color of vertex element (element enabled by FC and FA of IVAOGL).

[Caution]

When disabling lighting processing (LEN=0), also set all the other fields in Attribute to “0”. When either one of the other fields in Attribute is “1”, which decreases module performance.

SetIndexBaseAddress

[Format]



[Description of command]

Command		Description
COORD	00_H	Specification for the coordinate element (X, Y, Z, W)
COLF	01_H	Specification for the front face color (Rf, Gf, Bf, Af)
COLB	02_H	Specification for the rear (=back) face color (Rb, Gb, Bb, Ab)
NORM	03_H	Specification for the normal vector (Nx, Ny, Nz)
TEX0	04_H	Specification for texture 0 (S0, T0,00)
TEX1	05_H	Specification for texture 1 (S1, T1, 01)
F	06_H	Specification for fog (F)

[Description of parameter]

Field name	Content	Effective range
Base Address	Starting AXI address of element data	00000000_H to $FFFFFFFC_H$ (only 32-bit boundary))

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8300_H	IDXBRCOORD	Updated by Command = 00_H (coordinate).
	0002_8308_H	IDXBRCOLF	Updated by Command = 01_H (front face color).
	0002_8310_H	IDXBRCOLB	Updated by Command = 02_H (rear face color).
	0002_8318_H	IDXBRNORM	Updated by Command = 03_H (normal vector).
	0002_8320_H	IDXBRTEX0	Updated by Command = 04_H (texture 0).
	0002_8328_H	IDXBRTEX1	Updated by Command = 05_H (texture 1).
	0002_8330_H	IDXBRF	Updated by Command = 06_H (fog).

[Processing]

This command updates the above update registers. Set the starting address (Base Address) where the vertex parameters are stored when executing index reference DL (G_VertexIndex), to this command. This starting address is different from the starting address of indirect DL.

This command performs no update processing when Command is other than those in the above update register table.

SetIndexStride

[Format]

31	24 23	0
SetIndexStride ($D1_H$)	Command	Reserved
Stride		

[Description of command]

Same as SetIndexBaseAddress.

[Description of parameter]

Field name	Content	Effective range
Stride	AXI address interval of element data	00000004_H to $0003FFFC_H$ (only 32-bit boundary)

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8304_H	IDXSTRIDECOORD	Updated by Command = 00_H (coordinate).
	0002_830C_H	IDXSTRIDECOLF	Updated by Command = 01_H (front face color).
	0002_8314_H	IDXSTRIDECOLB	Updated by Command = 02_H (rear face color).
	0002_831C_H	IDXSTRIDENORM	Updated by Command = 03_H (normal vector).
	0002_8324_H	IDXSTRIDETEX0	Updated by Command = 04_H (texture 0).
	0002_832C_H	IDXSTRIDETEX1	Updated by Command = 05_H (texture 1).
	0002_8334_H	IDXSTRIDEF	Updated by Command = 06_H (fog).

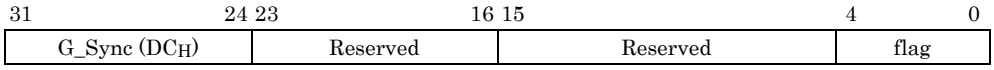
[Processing]

This command updates the above update registers. Set an AXI address interval (byte address) between vertex parameters when executing index reference DL, to this command. However, the address must be a 32-bit boundary address.

This command performs no update processing when Command is other than those in the above update register table.

G_Sync

[Format]



flag:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	—														V1BLANK	V0BLANK

[Description of parameter]

Field name	Function	Effective range
V0BLANK	Waits for vertical blank signal of display 0.	0: Disable 1: Waits for KTS_v0blank signal.
V1BLANK	Waits for vertical blank signal of display 1.	0: Disable, 1: Waits for KTS_v1blank signal.

[Processing]

This command stops the subsequent display list processing until the event set by Flag is detected.

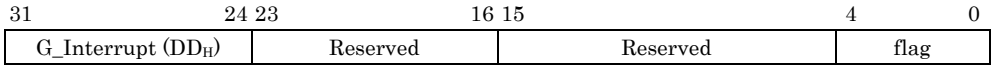
When Bit 0=0 and Bit 1=0, the same processing as G_Nop is performed.

When Bit 0=1 or Bit 1=1, this command stops DDL/IDL-FIFO read processing.

When Bit 0=1 and Bit 1=1, this command stops DDL/IDL-FIFO read processing until these conditions occur simultaneously.

G_Interrupt

[Format]



[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8064$_H$	VRINTM	Normal interrupt mask

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8060$_H$	VRINT	The DFIN and GFIFO fields are updated.

[Processing]

This command generates interrupts to the host CPU according to the condition set by Flag.
This command reflects Flag information to the VRINT register.

Flag:

Bit number	4	3	2	1	0
Bit field name	Reserved	Reserved	DRAWFIN	Reserved	GFIFO

Bit 0 GFIFO

GFIFO interrupt

0 Generates an interrupt at the point when this command reaches GFIFO.

1 Masks the GFIFO interrupt. Generates no interrupts.

Bit 2 DRAWFIN

Drawing end interrupt

0 Generates an interrupt at the point when the execution of the command sent just before this command ends.

1 Masks the drawing end interrupt. Generates no interrupts.

G_VertexLOG

[Format]

31	24 23	16 15	0
G_VertexLOG (32_H)	Reserved	Reserved	
X.float/fixed			
Y.float/fixed			
Z.float/fixed			

The above format is just an example. As with G_Vertex, word configuration varies with the setting of the GMDR0 and IVAOGL registers.

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	Sets the vertex format.
	0002_8054_H	IDFOGL	
	0002_8058_H	IVAOGL	
	0002_8248_H	C_LGA	Holds the log write address.
	0002_8440_H	S_FBR	The setting of the rendering engine changes due to the log write. This register is used to save this original setting so that it will be restored
	0002_8444_H	S_XRR	
	0002_8420_H	S_MDR0	
	0002_8430_H	S_MDR4	

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8248_H	C_LGA	Holds the log write address.

[Processing]

This command writes the result of the coordinate transformation to the address set by the C_LGA register. After writing the log, the C_LGA register is incremented for the number of written words. Set the C_LGA register setting using DC_LogOutAddr.

Write processing is the same as G_VertexNopLog.

After writing the log, the same graphics processing as G_Vertex is performed.

[Restriction]

This command must be present between G_Begin and G_End. There is no restriction on graphics type.

G_VertexNopLOG

[Format]

31	24 23	16 15	0
G_VertexNopLOG (33_H)	Reserved	Reserved	
X.float/fixed			
Y.float/fixed			
Z.float/fixed			

The above format is just an example. As with G_Vertex, word configuration varies with the setting of the GMDR0 and IVAOGL registers.

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	Sets the vertex format
	0002_8054_H	IDFOGL	
	0002_8058_H	IVAOGL	
	0002_8248_H	C_LGA	Holds the log write address.
	0002_8440_H	S_FBR	The setting of the rendering engine changes due to the log write. This register is used to save this original setting so that it will be restored
	0002_8444_H	S_XRR	
	0002_8420_H	S_MDR0	
	0002_8430_H	S_MDR4	

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8248_H	C_LGA	Holds the log write address.

[Processing]

This command writes the result of the coordinate transformation to the address set using the C_LGA register. After writing the log, the C_LGA register is incremented for the number of written words. Set the C_LGA register setting using DC_LogOutAddr.

[Restriction]

This command must be present between G_Begin (Points/nclip_Points) and G_End. This command cannot be used for graphics other than point.

SetModeRegister

[Format]

31 24	23 16	15 0
SetModeRegister ($C0_H$)	Command	Reserved
MDR1*/MDR2*		

[Description of Command]

Command		Description
MDR1	00_H	Updates the set MDR1 of the body.
MDR2	01_H	Updates the set MDR2 of the body.
MDR2S	03_H	Updates the set MDR2 of the shadow.
MDR2TL	07_H	Updates the set MDR2TL of the top-left c.

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8200_H	C_MDR1	Updated by Command = 00_H
	0002_820C_H	C_MDR2	Updated by Command = 01_H.
	0002_8210_H	C_MDR2S	Updated by Command = 03_H.
	0002_8214_H	C_MDR2TL	Updated by Command = 07_H.

[Processing]

This command updates the above update registers as the setting of the shadowed graphics and top-left rule not applicable graphics.

This command performs no update processing when Command is other than those in the above update register table.

C_MDR2TL sets a bordering line drawn in top-left rule not applicable mode. It sets a drawing effect same as the body setting (C_MDR2).

SetGModeRegister

[Format]

31	24 23	16 15	0
SetGModeRegister($C1_H$)	Command	Reserved	
Parameter=GMDR1E/GMDR2E/IDFOGL/IVAOGL			

[Description of Command]

Command		Description
GMDR0	00_H	Updates GMDR0.
GMDR1	01_H	Updates GMDR1.
GMDR2	02_H	Updates GMDR2.
GMDR2E	20_H	Updates GMDR2E.
IDFOGL	80_H	Updates IDFOGL.
IVAOGL	90_H	Updates IVAOGL.

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8040_H	GMDR0	Updated by Command = 00_H.
	0002_8044_H	GMDR1	Updated by Command = 01_H.
	0002_8048_H	GMDR2	Updated by Command = 02_H.
	0002_8050_H	GMDR2E	Updated by Command = 02_H or 20_H.
	0002_8054_H	IDFOGL	Updated by Command = 80_H.
	0002_8058_H	IVAOGL	Updated by Command = 90_H.
VL Engine	0003_05A0_H	VL_VERTEXSET	Updated by Command = 00_H or 90_H.
Primitive Engine	0004_002C_H	LINE_SET_REG	Updated by Command = 01_H or 10_H.
	0004_0030_H	POLYGON_SET_REG	Updated by Command = 02_H or 20_H.
	0004_0034_H	VERTEX_SET_REG	Updated by Command = 00_H or 90_H

[Processing]

This command updates the above update registers.
This command performs no update processing when Command is other than those in the above update register table.

[Caution]

When the update processing occurs, wait for the processing in the geometry engine to end before updating the register.
Update of these registers may decrease module performance.　Reduce updating.

OverlapXYOfft

[Format]

31 24	23 16	15 0
OverlapXYOfft ($C8_H$)	C_OXYO (00_H)	Reserved
Y Offset		X Offset

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8100_H	C_OXYO	Updated by command 00h.

[Processing]

This command updates the above update register. C_OXYO is referenced as an XY coordinate offset when drawing a shadowed graphics (GMDR2E:SP=1).

OverlapZOfft

[Format]

31 24	23 16	15 0
OverlapZOfft ($C9_H$)	Command	Reserved
don't care		Z Offset

Note: When MDR0 ZP=1, only lower 8 bits are enabled.

31 24	23 16	15 8	7 0
OverlapZOfft ($C9_H$)	Packed_ONBS (07_H)	Reserved	
S_Z Offset	Reserved	N_Z Offset	O_Z Offset

[Description of Command]

Command		Description
Origin	00_H	"Z" value offset of the body
NonTopLeft	01_H	"Z" value offset of the non-top-left surrounding straight line
Shadow	03_H	"Z" value offset of the shadow
Packed_ONBS	07_H	Sets a "Z" value offset of Origin, NonTopLeft and Shadow respectively in 8 bits.

[Setting register that affects processing]

Module	Register address	Register name	Description
Vertex Reader	0002_8420_H	S_MDR0	ZP field affects processing.

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8108_H	C_OZORG	Updated by Commands = 00_H, 07_H.
	0002_810C_H	C_OZNTL	Updated by Commands = 01_H, 07_H.

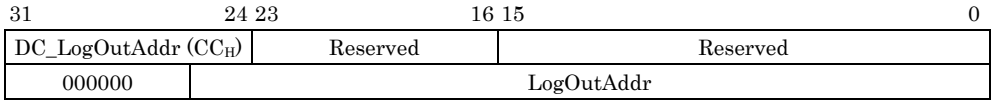
[Processing]

This command updates the above update registers.
The set value of the above update registers is added to the "Z" value after geometry coordinate transformation has been performed, when drawing a shadowed graphics or a non-top-left graphics. The calculating method of each figure part is shown below.

Object	Method of Z coordinate calculation
main body of figure	Main body's Z coordinate + "Origin" offset
Border in non-toplext rule mode	Main body's Z coordinate + "Origin" offset + "NonTopLeft" offset
Shadow part	Main body's Z coordinate + "Origin" offset + "Shadow" offset
Border of shadow part in non-toplext rule mode	Main body's Z coordinate + "Origin" offset + "Shadow" offset + "NonTopLeft" offset

DC_LogOutAddr

[Format]



[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8248$_H$	C_LGA	Holds the log write address.

[Processing]

This command updates data in the C_LGA register. The C_LGA register is used by G_VertexLOG and G_VertexNopLOG.

SetColorRegister

[Format]

31	24 23	16 15	0
SetColorRegister (CE_H)	Command	Reserved	
FGC8/16/24			

[Description of Command]

Command		Description
ForeColor	00_H	Updates the body foreground color C_FCC.
BackColor	01_H	Updates the body background color C_BCC.
ForeColorShadow	02_H	Updates the shadow foreground color C_FCC.
BackColorShadow	03_H	Updates the shadow background color C_BCC.

[Update register]

Module	Register address	Register name	Description
Vertex Reader	0002_8230_H	C_FCC	Updated by Command = 00_H (ForeColor).
	0002_8234_H	C_BCC	Updated by Command = 01_H (BackColor).
	0002_8238_H	C_FCSC	Updated by Command = 02_H (ForeColorShadow).
	0002_823C_H	C_BCSC	Updated by Command = 03_H (BackColorShadow).

[Processing]

This command updates the above update registers for setting of the color of the shadowed graphics.

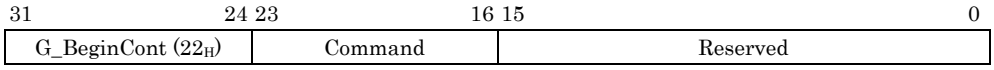
This command performs no update processing when Command is other than those in the above update register table.

[Caution]

When the update processing occurs, wait for the processing in the geometry engine to end before updating the register.
Update of these registers may decrease module performance.　Reduce updating.

G_BeginCont

[Format]

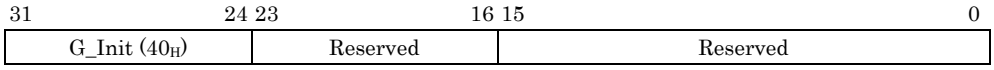


[Processing]

This command performs the same processing as G_Begin.

G_Init

[Format]

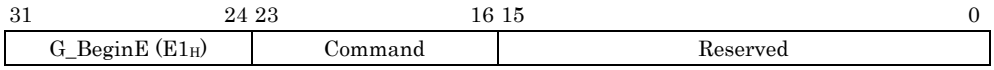


[Processing]

This command performs nothing.

G_BeginE

[Format]

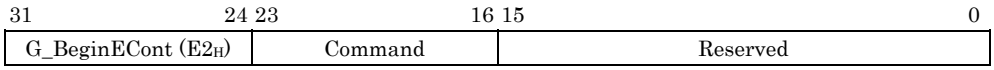


[Processing]

This command performs the same processing as G_Begin.

G_BeginECont

[Format]

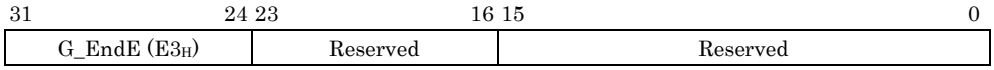


[Processing]

This command performs the same processing as G_Begin.

G_EndE

[Format]



[Processing]

This command performs the same processing as G_End.

SetLVertex2i

[Format]

31 24	23 16	15 0
SetLVertex2i (72_H)	Reserved	Reserved
LX0dc		
LY0dc		

[Processing]

This command is processed as SetVertex2i.

[Caution]

This command is for compatibility with existing GDCs.

Fujitsu recommends that coordinate setting be performed using SetVertex.

SetLVertex2iP

[Format]

31 24	23 16	15 0
SetLVertex2iP (73_H)	Reserved	Reserved
LY0dc		LX0dc

[Processing]

This command is processed as SetVertex2iP.

[Caution]

This command is for compatibility with existing GDCs.

Fujitsu recommends that coordinate setting be performed using SetVertex.

6.4.5 Rendering Display List

Command list

The rendering display list of KOTTOS, and each command code are shown below.

Rendering Display List Code Table (1)

Type	Code	Command	Code
DrawPixel	00_H	Pixel	00_H
DrawPixelZ	01_H	PixelZ	01_H
DrawLine	02_H	Xvector	20_H
		Yvector	21_H
		XvectorNoEnd	22_H
		YvectorNoEnd	23_H
		XvectorBlpClear	24_H
		YvectorBlpClear	25_H
		XvectorNoEndBlpClear	26_H
		YvectorNoEndBlpClear	27_H
		AntiXvector	28_H
		AntiYvector	29_H
		AntiXvectorNoEnd	$2A_H$
		AntiYvectorNoEnd	$2B_H$
		AntiXvectorBlpClear	$2C_H$
		AntiYvectorBlpClear	$2D_H$
		AntiXvectorNoEndBlpClear	$2E_H$
		AntiYvectorNoEndBlpClear	$2F_H$
DrawLine2i/DrawLine2iP	03_H/04_H	ZeroVector	30_H
		Onevector	31_H
		ZeroVectorNoEnd	32_H
		OnevectorNoEnd	33_H
		ZeroVectorBlpClear	34_H
		OnevectorBlpClear	35_H
		ZeroVectorNoEndBlpClear	36_H
		OnevectorNoEndBlpClear	37_H
		AntiZeroVector	38_H
		AntiOnevector	39_H
		AntiZeroVectorNoEnd	$3A_H$
		AntiOnevectorNoEnd	$3B_H$
		AntiZeroVectorBlpClear	$3C_H$
		AntiOnevectorBlpClear	$3D_H$
		AntiZeroVectorNoEndBlpClear	$3E_H$
		AntiOnevectorNoEndBlpClear	$3F_H$

Draw	$F0_H$	Flush	$C1_H$
		PolygonEnd	$E1_H$
DrawTrap	05_H	TrapRight	60_H
		TrapLeft	61_H
SetVertex2i/ SetVertex2iP	$70_H/71_H$	Normal	FF_H
		PolygonBegin	$E0_H$
DrawVertex2i/ DrawVertex2iP	06h/07_H	TriangleFan	62_H
		FlagTriangleFan	63_H
SetVertex	14_H	Fixed	00_H
		PackedInt	FF_H
DrawVertex	15_H	TriangleFan Fixed	00_H
		TriangleFan PackedInt	62_H
		Line Fixed ZeroVector	10_H
		Line Fixed Onevector	11_H
		Line Fixed ZeroVectorNoEnd	12_H
		Line Fixed OnevectorNoEnd	13_H
		Line Fixed ZeroVectorBlpClear	14_H
		Line Fixed OnevectorBlpClear	15_H
		Line Fixed ZeroVectorNoEndBlpClear	16_H
		Line Fixed OnevectorNoEndBlpClear	17_H
		Line Fixed AntiZeroVector	18_H
		Line Fixed AntiOnevector	19_H
		Line Fixed AntiZeroVectorNoEnd	$1A_H$
		Line Fixed AntiOnevectorNoEnd	$1B_H$
		Line Fixed AntiZeroVectorBlpClear	$1C_H$
		Line Fixed AntiOnevectorBlpClear	$1D_H$
		Line Fixed AntiZeroVectorNoEndBlpClear	$1E_H$
		Line Fixed AntiOnevectorNoEndBlpClear	$1F_H$
		Line PackedInt ZeroVector	30_H
		Line PackedInt Onevector	31_H
		Line PackedInt ZeroVectorNoEnd	32_H
		Line PackedInt OnevectorNoEnd	33_H
		Line PackedInt ZeroVectorBlpClear	34_H
		Line PackedInt OnevectorBlpClear	35_H
		Line PackedInt ZeroVectorNoEndBlpClear	36_H
		Line PackedInt OnevectorNoEndBlpClear	37_H
		Line PackedInt AntiZeroVector	38_H
		Line PackedInt AntiOnevector	39_H
		Line PackedInt AntiZeroVectorNoEnd	$3A_H$
		Line PackedInt AntiOnevectorNoEnd	$3B_H$
		Line PackedInt AntiZeroVectorBlpClear	$3C_H$
		Line PackedInt AntiOnevectorBlpClear	$3D_H$
		Line PackedInt AntiZeroVectorNoEndBlpClear	$3E_H$
		Line PackedInt AntiOnevectorNoEndBlpClear	$3F_H$

DrawRectP	09_H	BltFill	41_H
		ClearPolyFlag	$E2_H$
DrawRectAlphaMapP	$1E_H$	BltFill	41_H
DrawBitmapP	$0B_H$	BltDraw	42_H
		DrawBitmap	43_H
DrawBitmapLargeP	$2F_H$	BltDraw	42_H
BltCopyP	$0D_H$	TopLeft	44_H
		TopRight	45_H
		BottomLeft	46_H
		BottomRight	47_H
BltCopyAlternateP	$0F_H$	TopLeft	44_H
BltCopyAltAlphaMapP	$1F_H$	Normal	01_H
		ABR	00_H
BltCopyCompressedP	$2D_H$	TopLeft	44_H
BltCopyCompAlphaMapP	$2E_H$	TopLeft	44_H
SetRegister	$F1_H$	No command	
LoadFirm	18_H	Display list	00_H
		Memory	01_H
RegTexture	19_H	Base	00_H
		State	01_H
BindTexture	$1A_H$	Reserved	00_H
SetFog	$1B_H$	Table	00_H

Description of rendering display list

Parameter for a command are stored in the corresponding register. For the meaning of parameters, see the description of each register.

DrawPixel

[Format]

31 24	23 16	15 0
DrawPixel (00_H)	Pixel (00_H)	Reserved
PXs		0
PYs		0

[Setting register that affects processing]

Module	Register address	Register name	Description
Rendering Engine	0002_0420_H	MDR0	CF and ZP fields affect processing.
	0002_0424_H	MDR1	Each field affects processing.
	0002_0040_H	Rs	Initial value of color red component
	0002_004C_H	Gs	Initial value of color green component
	0002_0058_H	Bs	Initial value of color blue component
	0002_0064_H	As	Initial value of color α component
	0002_0080_H	Zs	Initial value of Z coordinate
	0002_008C_H	Z32s	Integer part of initial value of 32-bit Z coordinate
	0002_00A0_H	Fs	Initial value of fog coordinate
	0002_00C0_H	S0s	Initial value of S coordinate of texture 0
	0002_00CC_H	T0s	Initial value of T coordinate of texture 0
	0002_00D8_H	Q0s	Initial value of Q coordinate of texture 0
	0002_00E4_H	S1s	Initial value of S coordinate of texture 1
	0002_00F0_H	T1s	Initial value of T coordinate of texture 1
	0002_00FC_H	Q1s	Initial value of Q coordinate of texture 1

[Description of parameter]

Field name	Content	Effective range
PXs	X coordinate of point	-4096 to 4095
PYs	Y coordinate of point	-4096 to 4095

Data format

Data	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PXs	S			Int													Frac			
PYs	S			Int													Frac			

[Processing]

This command draws points.

To add various drawing effects, set enable/disable of drawing effect by using the MDR1 register. Set each parameter value to the register for initial value of drawing parameters.

DrawPixelZ

[Format]

When ZP of MDR0 is 00b or 01b:

31 24	23 16	15 0
DrawPixel (01_H)	PixelZ (01_H)	Reserved
PXs		0
PYs		0
PZ16s		0

When ZP of MDR0 is 10b:

31 24	23 16	15 0
DrawPixel (01_H)	PixelZ (01_H)	Reserved
PXs		0
PYs		0
PZ32s		

[Description of parameter]

Field name	Content	Effective range
PXs	X coordinate of point	-4096 to 4095
PYs	Y coordinate of point	-4096 to 4095
PZ16s	Z value of point (MDR0:ZP= 00_B or 01_B)	0 to 65535
PZ32s	Z value of point (MDR0:ZP= 10_B)	0 to 4294967295

Data format:

Data	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PXs	S			Int													Frac			
PYs	S			Int													Frac			
PZ16s	0			Int																

[Processing]

This command draws points having Z value.

When DrawPixelZ is used, depth test is performed even when the ZC field of the MDR1 register is disabled. It is also possible to set various drawing effects in the same way as DrawPixel. This command is for upward compatibility; Fujitsu recommends that DrawPixel be used.

DrawLine

[Format]

31 – 24	23 – 16	15 – 12	11 – 8	7 – 0
DrawLine (02_H)	Command	Reserved		
LPN				
LXs				
LXde				
LYs				
LYde				

[Description of Command]

Command		Description
Xvector	20_H	Draws lines (the main axis is X).
Yvector	21_H	Draws lines (the main axis is Y).
XvectorNoEnd	22_H	Draws lines (the main axis is X; draws no endpoint).
YvectorNoEnd	23_H	Draws lines (the main axis is Y; draws no endpoint).
XvectorBlpClear	24_H	Draws lines (the main axis is X; clears the broken line pattern reference position before starting drawing).
YvectorBlpClear	25_H	Draws lines (the main axis is Y; clears the broken line pattern reference position before starting drawing).
XvectorNoEndBlpClear	26_H	Draws lines (the main axis is X; draws no endpoint; clears the broken line pattern reference position before starting drawing).
YvectorNoEndBlpClear	27_H	Draws lines (the main axis is Y; draws no endpoint; clears the broken line pattern reference position before starting drawing).
AntiXvector	28_H	Draws antialiased lines (the main axis is X).
AntiYvector	29_H	Draws antialiased lines (the main axis is Y).
AntiXvectorNoEnd	$2A_H$	Draws antialiased lines (the main axis is X; draws no endpoint).
AntiYvectorNoEnd	$2B_H$	Draws antialiased lines (the main axis is Y; draws no endpoint).
AntiXvectorBlpClear	$2C_H$	Draws antialiased lines (the main axis is X; clears the broken line pattern reference position before starting drawing).
AntiYvectorBlpClear	$2D_H$	Draws antialiased lines (the main axis is Y; clears the broken line pattern reference position before starting drawing).
AntiXvectorNoEndBlpClear	$2E_H$	Draws antialiased lines (the main axis is X; draws no endpoint; clears the broken line pattern reference position before starting drawing).
AntiYvectorNoEndBlpClear	$2F_H$	Draws antialiased lines (the main axis is Y; draws no endpoint; clears the broken line pattern reference position before starting drawing).

[Description of parameter]

Field name	Content	Effective range
LPN	Number of pixels for straight line	1 to 8191
LXs	X coordinate of the starting point	-4096 to4095
LXde	When drawing with Y set as the main axis, input a DX/DY value (only decimal number). When drawing with X set as the main axis, input "1" or "-1" depending on the drawing direction.	-1.0～1.0
LYs	Y coordinate of the starting point	-4096 to 4095
LYde	When drawing with X set as the main axis, input a DX/DY value (only decimal number). When drawing with Y set as the main axis, input "1" or "-1" depending on the drawing direction.	-1.0 to 1.0

Data format:

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPN	0			Int													Frac						0									
LXs	S			Int													Frac						0									
LXde	S															Int	Frac															
LYs	S			Int													Frac															
LYde	S															Int	Frac															

[Setting register that affects processing]

Module	Register address	Register name	Description
Rendering Engine	0002_0420H	MDR0	CF and ZP fields affect processing.
	0002_0424H	MDR1	Each field affects processing.
	0002_0154H	LZs	Initial value of Z coordinate
	0002_0158H	LZde	Inclination value of Z coordinate
	0002_008CH	Z32s	Integer part of initial value of 32-bit Z coordinate
	0002_0090H	dZ32dx	Integer part of X direction incremental value of 32-bit Z coordinate
	0002_0094H	dZ32dy	Integer part of Y direction incremental value of 32-bit Z coordinate

[Processing]

This command draws lines by specifying an initial value and an inclination value directly. It starts drawing after parameters are set to the register for line drawing.

It can also draw line having Z comparison value, by setting the initial value and incremental value of Z coordinate to the LZs and LZde registers respectively using SetRegister. In 32-bit "Z" value mode, the LZs and LZde registers are valid only in terms of the sign and the decimal part, and the integer part used is the one in Z32s and dZ32dy.

When performing alpha blending, value of the ALF register is used as a blend ratio.

This display list is for upward compatibility; Fujitsu recommends that drawing be performed using SetVertex and DrawVertex.

DrawLine2i

[Format]

31 24	23 16	15 0	
DrawLine2i (03_H)	Command	Reserved	Vertex
LFXs		0	
LFYs		0	

[Description of Command]

Command		Description
ZeroVector	30_H	Draws a line from vertex 0 to vertex 1.
OneVector	31_H	Draws a line from vertex 1 to vertex 0.
ZeroVectorNoEnd	32_H	Draws a line from vertex 0 to vertex 1 (draws no endpoint).
OneVectorNoEnd	33_H	Draws a line from vertex 1 to vertex 0 (draws no endpoint).
ZeroVectorBlpClear	34_H	Draws a line from vertex 0 to vertex 1 (clears the broken line pattern reference position before starting drawing).
OneVectorBlpClear	35_H	Draws a line from vertex 1 to vertex 0 (clears the broken line pattern reference position before starting drawing).
ZeroVectorNoEndBlpClear	36_H	Draws a line from vertex 0 to vertex 1 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
OneVectorNoEndBlpClear	37_H	Draws a line from vertex 1 to vertex 0 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
AntiZeroVector	38_H	Draws an antialiased line from vertex 0 to vertex 1.
AntiOneVector	39_H	Draws an antialiased line from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	$3A_H$	Draws an antialiased line from vertex 0 to vertex 1 (draws no endpoint).
AntiOneVectorNoEnd	$3B_H$	Draws an antialiased line from vertex 1 to vertex 0 (draws no endpoint).
AntiZeroVectorBlpClear	$3C_H$	Draws an antialiased line from vertex 0 to vertex 1 (clears the broken line pattern reference position before starting drawing).
AntiOneVectorBlpClear	$3D_H$	Draws an antialiased line from vertex 1 to vertex 0 (clears the broken line pattern reference position before starting drawing).
AntiZeroVectorNoEndBlpClear	$3E_H$	Draws an antialiased line from vertex 0 to vertex 1 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
AntiOneVectorNoEndBlpClear	$3F_H$	Draws an antialiased line from vertex 1 to vertex 0 (draws no endpoint; clears the broken line pattern reference position before starting drawing).

[Description of parameter]

Field name	Content	Effective range
Vertex	Vertex number to be set	0, 1, 2
LFXs	X coordinate of vertex	-4096 to 4095
LFYs	Y coordinate of vertex	-4096 to 4095

Data format

Data	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFXs	S										Int								Frac	
LFYs	S										Int								Frac	

[Processing]

This command draws an XY setup line.　It starts drawing after an endpoint has been set to the register for drawing the XY setup line.

When performing alpha blending, value of the ALF register is used as a blend ratio.

This display list is for upward compatibility; Fujitsu recommends that drawing be performed using SetVertex and DrawVertex.

DrawLine2iP

[Format]

31 24	23 16	15 0	
DrawLine2iP (04_H)	Command	Reserved	Vertex
LFYs		LFXs	

[Description of Command]

Same as DrawLine2i.

[Description of parameter]

Same as DrawLine2i.

Data format

Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFXs	S	Int														
LFYs	S	Int														

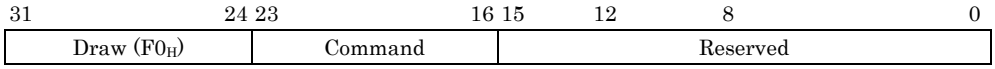
[Processing]

As with DrawLine2i, this command draws an XY setup line. Only integer (packed format) can be used for coordinate.

This display list is for upward compatibility; Fujitsu recommends that drawing be performed using SetVertex and DrawVertex.

Draw

[Format]



[Description of Command]

Command		Description
Flush	$C1_H$	Performs flush processing.
PolygonEnd	$E1_H$	Draws closed polygons.

[Setting register that affects processing]

Module	Register address	Register name	Description
Rendering Engine	0002_0420_H	MDR0	The CF and ZP fields affect processing.
	0002_0428_H	MDR2	Each field affects processing.
	0002_0080_H	Zs	Initial value of Z coordinate
	0002_0084_H	dZdx	X direction incremental value of Z coordinate
	0002_0088_H	dZdy	Y direction incremental value of Z coordinate
	0002_008C_H	Z32s	Integer part of initial value of 32-bit Z coordinate
	0002_0090_H	dZ32dx	Integer part of X direction incremental value of 32-bit Z coordinate
	0002_0094_H	dZ32dy	Integer part of Y direction incremental value of 32-bit Z coordinate
	0002_00A0_H	Fs	Initial value of fog coordinate
	0002_00A4_H	dFdx	X direction incremental value of fog coordinate
	0002_00A8_H	dFdy	Y direction incremental value of fog coordinate
	0002_00C0_H	S0s	Initial value of S coordinate of texture 0
	0002_00C4_H	dS0dx	X direction incremental value of S coordinate of texture 0
	0002_00C8_H	dS0dy	Y direction incremental value of S coordinate of texture 0
	0002_00CC_H	T0s	Initial value of T coordinate of texture 0
	0002_00D0_H	dT0dx	X direction incremental value of T coordinate of texture 0
	0002_00D4_H	dT0dy	Y direction incremental value of T coordinate of texture 0
	0002_00D8_H	Q0s	Initial value of Q coordinate of texture 0
	0002_00DC_H	dQ0dx	X direction incremental value of Q coordinate of texture 0
	0002_00E0_H	dQ0dy	Y direction incremental value of Q coordinate of texture 0
	0002_00E4_H	S1s	Initial value of S coordinate of texture 1
	0002_00E8_H	dS1dx	X direction incremental value of S coordinate of texture 1
	0002_00EC_H	dS1dy	Y direction incremental value of S coordinate of texture 1
	0002_00F0_H	T1s	Initial value of T coordinate of texture 1
	0002_00F4_H	dT1dx	X direction incremental value of T coordinate of texture 1

	0002_00F8H	dT1dy	Y direction incremental value of T coordinate of texture 1
	0002_00FCH	Q1s	Initial value of Q coordinate of texture 1
	0002_0100H	dQ1dx	X direction incremental value of Q coordinate of texture 1
	0002_0104H	dQ1dy	Y direction incremental value of Q coordinate of texture 1

[Processing]

PolygonEnd:

This command draws a polygon formed by SetVertex2i/2iP:FlagTriangleFan and DrawVertex2i/2iP:FlagTriangleFan, using SetVertex2i/2iP:PolygonBegin as the first vertex. After drawing, this command automatically clears the flag for the circumscribed quadrangle area for the polygon.

When using depth comparison, texture, and fog, registers to set the initial value and inclination value of each component must be set.

Flush:

This command waits for the drawing processing to end and writes the drawing result and then changes the status of KOTTOS to idle.

DrawTrap

[Format]

31 24	23 16	15 12	8 0
DrawTrap (05_H)	Command	Reserved	
Ys		0	
Xs			
DXdy			
XUs			
DXUdy			
XLs			
DXLdy			
USN		0	
LSN		0	

[Description of Command]

Command		Description
TrapRight	60_H	Draws a right triangle.
TrapLeft	61_H	Draws a left triangle.

[Description of parameter]

Field name	Content	Effective range
Ys	Starting Y coordinate	-4096 to 4095
Xs	Starting X coordinate	-4096 to 4095
DXdy	Inclination value of long side	-4096 to 4095
XUs	Starting X coordinate of short side upper triangle section	-4096 to 4095
DXUdy	Inclination value of short side upper triangle section	-4096 to 4095
XLs	Starting X coordinate of short side lower triangle section	-4096 to 4095
DXLdy	Inclination value of short side lower triangle section	-4096 to 4095
USN	Number of spans in upper triangle section	0 to 8191
LSN	Number of spans in lower triangle section	0 to 8191

Data format

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ys	S				Int												0															
Xs	S				Int												Frac				0											
DXdy	S				Int												Frac															
XUs	S				Int												Frac															
DXUdy	S				Int												Frac															
XLs	S				Int												Frac															
DXLdy	S				Int												Frac															
USN	-																0				Int											
LSN	-																0				Int											

[Setting register that affects processing]

Module	Register address	Register name	Description
Rendering Engine	0002_0420H	MDR0	CF and ZP fields affect processing.
	0002_0428H	MDR2	Each field affects processing.
	0002_0488H	ALF	Blend ratio at α flat shading time
	0002_0480H	FC	Color when performing flat shading
	0002_0040H	Rs	Initial value of color red component when performing Gouraud shading
	0002_0044H	dRdx	X direction incremental value of color red component when performing Gouraud shading
	0002_0048H	dRdy	Y direction incremental value of color red component when performing Gouraud shading
	0002_004CH	Gs	Initial value of color green component when performing Gouraud shading
	0002_0050H	dGdx	X direction incremental value of color green component when performing Gouraud shading
	0002_0054H	dGdy	Y direction incremental value of color green component when performing Gouraud shading
	0002_0058H	Bs	Initial value of color blue component when performing Gouraud shading
	0002_005CH	dBdx	X direction incremental value of color blue component when performing Gouraud shading
	0002_0060H	dBdy	Y direction incremental value of color blue component when performing α Gouraud shading
	0002_0064H	As	Initial value of color α component when performing Gouraud shading
	0002_0068H	dAdx	X direction incremental value of color α component when performing α Gouraud shading
	0002_006CH	dAdy	Y direction incremental value of color α component when performing α Gouraud shading
	0002_0080H	Zs	Initial value of Z coordinate
	0002_0084H	dZdx	X direction incremental value of Z coordinate
	0002_0088H	dZdy	Y direction incremental value of Z coordinate
	0002_008CH	Z32s	Integer part of initial value of 32-bit Z coordinate
	0002_0090H	dZ32dx	Integer part of X direction incremental value of 32-bit Z coordinate
	0002_0094H	dZ32dy	Integer part of Y direction incremental value of 32-bit Z coordinate
	0002_00A0H	Fs	Initial value of fog coordinate
	0002_00A4H	dFdx	X direction incremental value of fog coordinate
	0002_00A8H	dFdy	Y direction incremental value of fog coordinate
	0002_00C0H	S0s	Initial value of S coordinate of texture 0
	0002_00C4H	dS0dx	X direction incremental value of S coordinate of texture 0
	0002_00C8H	dS0dy	Y direction incremental value of S coordinate of texture 0
	0002_00CCH	T0s	Initial value of T coordinate of texture 0
	0002_00D0H	dT0dx	X direction incremental value of T coordinate of texture 0
	0002_00D4H	dT0dy	Y direction incremental value of T coordinate of

		texture 0
0002_00D8H	Q0s	Initial value of Q coordinate of texture 0
0002_00DCH	dQ0dx	X direction incremental value of Q coordinate of texture 0
0002_00E0H	dQ0dy	Y direction incremental value of Q coordinate of texture 0
0002_00E4H	S1s	Initial value of texture 1 S coordinate
0002_00E8H	dS1dx	X direction incremental value of S coordinate of texture 1
0002_00ECH	dS1dy	Y direction incremental value of S coordinate of texture 1
0002_00F0H	T1s	Initial value of T coordinate of texture 1
0002_00F4H	dT1dx	X direction incremental value of T coordinate of texture 1
0002_00F8H	dT1dy	Y direction incremental value of T coordinate of texture 1
0002_00FCH	Q1s	Initial value of Q coordinate of texture 1
0002_0100H	dQ1dx	X direction incremental value of Q coordinate of texture 1
0002_0104H	dQ1dy	Y direction incremental value of Q coordinate of texture 1

[Processing]

This command draws triangles by setting an initial value and an inclination value directly. It executes DrawTrap after necessary parameters are set to the register for drawing triangle using SetRegister. The registers to be set for triangle vary with the MDR2 register.

This display list is for upward compatibility; Fujitsu recommends that drawing be performed using SetVertex and DrawVertex.

SetVertex2i

[Format]

31 24	23 16	15 4	3 2 1 0
SetVertex2i (70_H)	Command	Reserved	vertex
Xdc		0	
Ydc		0	

[Description of Command]

Command		Description
Normal	FF_H	Sets vertex data.
PolygonBegin	$E0_H$	Sets vertex data and the start of polygon simultaneously.

[Description of parameter]

Field name	Content	Effective range
Vertex	Vertex number to be set	0, 1, 2
Xdc	Vertex X coordinate	-4096 to 4095
Xdc	Vertex Y coordinate	-4096 to 4095

Data format

Data	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Xdc	S										Int								Frac	
Ydc	S										Int								Frac	

[Processing]

This command sets vertex data for XY setup primitive drawing to registers.
When the PolygonBegin command is specified, the display list of until the PolygonEnd command is input is treated as the one for polygon drawing.

SetVertex2iP

[Format]

31 24	23 16	15 4	3 2 1 0
SetVertex2iP (71_H)	Command	Reserved	vertex
Ydc		Xdc	

[Description of Command]

Same as SetVertex2i.

[Description of parameter]

Same as SetVertex2i.

Data format

Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Xdc	Sign						Int									
Ydc	Sign						Int									

[Processing]

Same as SetVertex2i. Only integer (packed format) can be used for coordinate.

DrawVertex2i

[Format]

31 24	23 16	15 0	
DrawVertex2i (06_H)	Command	Reserved	Vertex
Xdc		0	
Ydc		0	

[Description of Command]

Command		Description
TriangleFan	62_H	Draws an XY setup triangle.
FlagTriangleFan	63_H	Draws an XY setup triangle for polygon drawing to the flag buffer.

[Description of parameter]

Field name	Content	Effective range
Xdc	Starting X coordinate	4096 to 4095
Ydc	Starting Y coordinate	4096 to 4095

Data format:

Data	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Xdc	S										Int								Frac	
Ydc	S										Int								Frac	

[Processing]

This command draws XY setup triangles. It starts drawing after parameters are set to the register for drawing XY setup triangle.

DrawVertex2iP

[Format]

31 24	23 16	15 0	
DrawVertex2iP (07_H)	Command	Reserved	Vertex
Ydc		Xdc	

[Description of Command]

Same as DrawVertex2i.

[Description of parameter]

Data format:

Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Xdc	S				Int											
Ydc	S				Int											

[Processing]

Same as DrawVertex2i. Packed integer format is used for vertex.

SetVertex

[Format]

When ZP of MDR0 is 00_B or 01_B:

31 24	23 16	15 9	8 7 6 5 4 3 2 1	0
SetVertex (14_H)	PackedInt (FF_H)	Reserve	ParamType	Vertex
Ydc.int		Xdc.int		
Z16.fixed				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q0.fixed				
F.fixed				

31 24	23 16	15 9	8 7 6 5 4 3 2 1	0
SetVertex (14_H)	Fixed (00_H)	Reserve	ParamType	Vertex
Xdc.fixed				
Ydc.fixed				
Z16.fixed				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q1.fixed				
F.fixed				

When ZP of MDR0 is 10_B:

31 24	23 16	15 9	8 7 6 5 4 3 2 1 0
SetVertex (14_H)	PackedInt (FF_H)	Reserve	ParamType / Vertex
Ydc.int		Xdc.int	
Z32.fixed (Lower)			
Z32.fixed (Upper)			
Color			
S0.fixed			
T0.fixed			
Q0.fixed			
S1.fixed			
T1.fixed			
Q1.fixed			
F.fixed			

31 24	23 16	15 9	8 7 6 5 4 3 2 1 0
SetVertex (14_H)	Fixed (00_H)	Reserve	ParamType / Vertex
Xdc.fixed			
Ydc.fixed			
Z32.fixed (Lower)			
Z32.fixed (Upper)			
Color			
S0.fixed			
T0.fixed			
Q0.fixed			
S1.fixed			
T1.fixed			
Q1.fixed			
F.fixed			

[Description of Command]

Command		Description
PackedInt	FF_H	Sets vertex data having an integer coordinate packed into one word.
Fixed	00_H	Sets vertex data having a fixed decimal coordinate. The coordinate consists of two words.

[Description of parameter]

Field name	Content	Effective range
ParamType	Specifies which parameter is provided.	00_H to FF_H
Vertex	Specifies which vertex data to set among three vertex buffers. For triangle, either 0, 1 or 2 is set; for straight line, either 0 or 1 is set.	0,1,2
Xdc,Ydc	Coordinate of a vertex. When the command is PackedInt, the coordinate consists of one word. When the command is Fixed, the coordinate consists of two words, that is, Xdc and Ydc respectively consisting of one word.	-4096 to 4095
Z	Z coordinate of a vertex. When ZP of MDR0 is 00_B (=16bit/Z) or 01_B (=8bit/Z), the coordinate consists of one word. When ZP of MDR0 is 10_B, the coordinate consists of two words.	0 to 255 (8bit/Z) 0 to 65535 (16bit/Z) 0 to 4294967295 (32bit/Z)
Color	RGBA components of vertex color. In 16BPP mode, upper 5 bites of each component are used. In 8BPP mode, color index code is set to R. When CO of MDR0 is 0, the color data are arranged in order of ARGB; when CO of MDR0 is 1, the color data are arranged in order of RGBA. For the arrangement of the color data, see "data format"(two pages later).	00000000_H to $FFFFFFFF_H$
S0, T0	ST coordinate of texture 0 of vertex	-8192 to 8191
Q0	Q coordinate of texture 0 of vertex	0.00001526 to 1.0
S1, T1	ST coordinate of texture 1 of vertex	-8192 to 8191
Q1	Q coordinate of texture 1 of vertex	0.00001526 to 1.0
F	Fog coordinate of vertex	0 to 65535

Table 6.4.8 Relationship between ParamType and Necessary Parameters

Parameter name	Required/not required	Function
Xdc.int	Always required	
Ydc.int	Always required	
Z.fixed	Required when ParamType[0] is "1"	Used to specify when using Z comparison.
Color	Required when ParamType[1] is "1" or when ParamType[2] is "1"	ParamType[1] is for the specification for A component; it is specified when using alpha value Gouraud shading. Specify this parameter using 8 bits in 8-bit color mode, 16-bit color mode and 32-bit color mode.
		ParamType[2] is for the specification for RGB component; it is specified when using Gouraud shading. Specify this parameter using 8 bits in 8-bit color mode, 16-bit color mode and 32-bit color mode.
S0.fixed, T0.fixed	Required when ParamType[3] is "1"	Specify this parameter when using texture 0.
Q0.fixed	Required when ParamType[3] is "1" or when ParamType[4] is "1"	Specify this parameter when using perspective correction for texture coordinate.
S1.fixed, T1.fixed	Required when ParamType[5] is "1"	Specify this parameter when using texture 1.
Q1.fixed	Required when ParamType[5] is "1" or when ParamType[6] is "1"	Specify this parameter when using perspective correction for texture coordinate.
F.fixed	Required when ParamType[7] is "1"	Specify this parameter when using fog coordinate.

Data format:

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Xdc.int	-																			S	Int											
Ydc.int	-																			S	Int											
Xdc.fixed				S	Int															Frac												
Ydc.fixed				S	Int															Frac												
Z16.fixed	0	Int																		Frac												
Z32.fixed (Lower)	0																			Frac												
Z32.fixed (Upper)	Int																															
Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0.fixed			S	Int													Frac															
T0.fixed			S	Int													Frac															
Q0.fixed	0								Int	Frac																						
S1.fixed			S	Int													Frac															
T1.fixed			S	Int													Frac															
Q1.fixed	0								Int	Frac																						
F.fixed	0	Int																		Frac												

Color: When CO bit of MDR0 is “1”

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Color	R.Int								G.Int								B.Int								A.Int							

Color: When CO bit of MDR0 is “0”

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Color	A.Int								R.Int								G.Int								B.Int							

[Processing]

This command sets vertex data for triangle/line drawing when using the setup interface. Integer (packed format) and fixed point number can be used for XY coordinate. Use this command together with DrawVertex. Coordinates Xdc and Ydc consist of one word when the command is PackedInt; they consist of two words when the command is Fixed. The number of necessary parameters depends on the ParamType field, the command, and the Z value precision (MDR0:ZP).

Value used as the "A" component when drawing is determined as shown in *Table 6.4.9.*

Table 6.4.9 Value used as "A" component by SetVertex/DrawVertex

	ParamType[1]	MDR1/ MDR2 BM	0 CF	Value to be set as "A" component
A	0	0	0(8BPP)	FF_H
			(16BPP)	00_H when the "A" component (bit0) of FC register is set to "0"; FF_H when set to "1"
			2(32BPP)	The "A" component (bit7-0) of FC register
		1	–	Value of ALF register
	1	–	–	The "A" component of Color field

* For point and straight line, BM of MDR1 is referenced; for triangle and polygon, BM of MDR2 is referenced. However, the setting at the point DrawVertex (not SetVertex) is input is referenced.

DrawVertex

[Format]

When MDR0's ZP is 00b or 01b:

31 – 24	23 – 16	15 – 12	9 – 2	1 – 0
DrawVertex (15_H)	* PackedInt	Reserve	ParamType	Vertex
Ydc.int		Xdc.int		
Z.fixed				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q1.fixed				
F.fixed				

31 – 24	23 – 16	15 – 12	9 – 2	1 – 0
DrawVertex (15_H)	* Fixed	Reserve	ParamType	Vertex
Xdc.fixed				
Ydc.fixed				
Z.fixed				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q1.fixed				
F.fixed				

When MDR0's ZP is 10_B:

31 24	23 16	15 9	8 7 6 5 4 3 2	1 0
DrawVertex (15_H)	PackedInt	Reserve	ParamType	Vertex
Ydc.int		Xdc.int		
Z32.fixed (Lower)				
Z32.fixed (Upper)				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q1.fixed				
F.fixed				

31 24	23 16	15 9	8 7 6 5 4 3 2	1 0
DrawVertex (15_H)	Fixed	Reserve	ParamType	Vertex
Xdc.fixed				
Ydc.fixed				
Z32.fixed (Lower)				
Z32.fixed (Upper)				
Color				
S0.fixed				
T0.fixed				
Q0.fixed				
S1.fixed				
T1.fixed				
Q1.fixed				
F.fixed				

[Description of Command]

Command		Description
TriangleFan Fixed / Triangle Fan PackedInt	$00_H/62_H$	Draws a setup triangle.
Line Fixed ZeroVector / Line PackedInt ZeroVector	10_H /30_H	Draws a line from vertex 0 to vertex 1.
Line Fixed OneVector / Line PackedInt OneVector	11_H /31_H	Draws a line from vertex 1 to vertex 0.
Line Fixed ZeroVectorNoEnd / Line PackedInt ZeroVectorNoEnd	12_H /32_H	Draws a line from vertex 0 to vertex 1 (draws no endpoint).
Line Fixed OneVectorNoEnd / Line PackedInt OneVectorNoEnd	13_H /33_H	Draws a line from vertex 1 to vertex 0 (draws no endpoint).
Line Fixed ZeroVectorBlpClear / Line PackedInt ZeroVectorBlpClear	14_H /34_H	Draws a line from vertex 0 to vertex 1 (clears the broken line pattern reference position before starting drawing).
Line Fixed OneVectorBlpClear / Line PackedInt OneVectorBlpClear	15_H /35_H	Draws a line from vertex 1 to vertex 0 (clears the broken line pattern reference position before starting drawing).
Line Fixed ZeroVectorNoEndBlpClear / Line PackedInt ZeroVectorNoEndBlpClear	16_H /36_H	Draws a line from vertex 0 to vertex 1 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
Line Fixed OneVectorNoEndBlpClear / Line PackedInt OneVectorNoEndBlpClear	17_H /37_H	Draws a line from vertex 1 to vertex 0 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
Line Fixed AntiZeroVector / Line PackedInt AntiZeroVector	18_H /38_H	Draws an antialiased line from vertex 0 to vertex 1.
Line Fixed AntiOneVector / Line PackedInt AntiOneVector	19_H /39_H	Draws an antialiased line from vertex 1 to vertex 0.
Line Fixed AntiZeroVectorNoEnd / Line PackedInt AntiZeroVectorNoEnd	$1A_H$ /$3A_H$	Draws an antialiased line from vertex 0 to vertex 1 (draws no endpoint).
Line Fixed AntiOneVectorNoEnd / Line PackedInt AntiOneVectorNoEnd	$1B_H$ /$3B_H$	Draws an antialiased line from vertex 1 to vertex 0 (draws no endpoint).
Line Fixed AntiZeroVectorBlpClear / Line PackedInt AntiZeroVectorBlpClear	$1C_H$ /$3C_H$	Draws an antialiased line from vertex 0 to vertex 1 (before starting drawing, the broken line pattern reference position is cleared).
Line Fixed AntiOneVectorBlpClear / Line PackedInt AntiOneVectorBlpClear	$1D_H$ /$3D_H$	Draws an antialiased line from vertex 1 to vertex 0 (clears the broken line pattern reference position before starting drawing).
Line Fixed AntiZeroVectorNoEndBlpClear / Line PackedInt AntiZeroVectorNoEndBlpClear	$1E_H$ /$3E_H$	Draws an antialiased line from vertex 0 to vertex 1 (draws no endpoint; clears the broken line pattern reference position before starting drawing).
Line Fixed AntiOneVectorNoEndBlpClear / Line PackedInt AntiOneVectorNoEndBlpClear	$1F_H$ /$3F_H$	Draws an antialiased line from vertex 1 to vertex 0 (draws no endpoint; clears the broken line pattern reference position before starting drawing).

[Description of parameter]

See description of the SetVertex.

[Processing]

This command draws triangle/line when using the setup interface. Use this command together with SetVertex.

As with SetVertex, DrawVertex updates the vertex buffer and draws the graphics specified by Command.

Coordinates Xdc and Ydc consist of one word when the command is PackedInt; they consist of two words when the command is Fixed.

The number of necessary parameters depends on the ParamType field. See the SetVertex.
Slope calculation is performed internally using fixed point number, and the calculated value may not be complete precision. To get a result of higher precision, perform slope calculation using CPU, etc. and draw using DrawTrap.

DrawRectP

[Format]

31 24	23 16	15 0
DrawRectP (09_H)	Command	Reserved
RYs		RXs
RsizeY		RsizeX

[Description of Command]

Command		Description
BltFill	41_H	Fills in a rectangular area using the current color (single color). Drawing is performed to the frame buffer specified by FBR.
ClearPolyFlag	$E2_H$	Fills in a polygon drawing flag buffer area specified by PFBR, with value "0". RsizeX and Y are used to specify the size of a drawing frame.

[Description of parameter]

Field name	Content	Effective range
RXs	Filling starting X coordinate in the drawing frame	-4096 to 4095
RYs	Filling starting Y coordinate in the drawing frame	-4096 to 4095
RsizeX	X direction pixel count in a rectangular area to be filled in	1 to 4096
RsizeY	Y direction pixel count in a rectangular area to be filled in	1 to 4096

Data format:

Data	15 14 13	12 11 10 9 8 7 6 5 4 3 2 1 0
RXs	S	Int
RYs	S	Int
RsizeX	0	Int
RsizeY	0	Int

[Processing]

This command draws rectangles. It fills rectangular area using a foreground color.

To execute this command, set an 8-byte aligned value to XRES.

[When Command is ClearPolyFlag]

No clipping is performed.

Filling is performed in units of bytes. Therefore, an area larger than a specified coordinate range may be cleared. It is defined that a sufficient flag buffer must be allocated. When a sufficient flag buffer is allocated following this definition, the user needs not be concerned with this matter.

PFBR is used as the base register.

[Caution]

Transparency (TE of MDR4) and background color transparency (BT of BC) affect the processing.

When using DrawRectP, set “0” to TE (bit1) and BW (bit18) of MDR4 register and the BT bit of the BC register.

When using ClearPolyFlag, disable rendering clipping (CX, CY of MDR0).

DrawRectAlphaMapP

[Format]

31 24	23 16	15 0
DrawRectAlphaMapP (1E$_H$)	Command	Reserved
AMADDR		
RYs		RXs
RsizeY		RsizeX

[Description of Command]

Command		Description
BltFill	41$_H$	Fills in a rectangular area using the current color (single color).

[Description of parameter]

Field name	Content	Effective range
AMADDR	Alpha map storage address A value specified for this field rewrites ABR.	00000000$_H$ to 3FFFFFF8$_H$ (only 64-bit boundary)
RXs	Filling starting X coordinate in the drawing frame	-4096 to 4095
RYs	Filling starting Y coordinate in the drawing frame	-4096 to 4095
RsizeX	X direction pixel count in a rectangular area to be filled in	1 to 4096
RsizeY	Y direction pixel count in a rectangular area to be filled in	1 to 4096

Data format:

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AMADDR	0						Int																								0	
Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
RXs	S			Int																												
RYs	S			Int																												
RsizeX	0			Int																												
RsizeY	0			Int																												

[Processing]

This command draws a rectangle by specifying alpha blending by alpha map. It fills a rectangular area using a foreground color while performing alpha blending for each pixel. To execute this command, set an 8-byte aligned value to XRES. The stride of alpha map data is the same as RsizeX, and the alpha map data has no restriction on 8-byte alignment.

[Caution]

Transparency (TE of MDR4) and background color transparency (BT of BC) affect the processing.

Set TE (bit 1) and BW (bit 18) of MDR4 and the BT bit of the BC register to “0”.

DrawBitmapP

[Format]

31 24	23 16	15 0
DrawBitmapP ($0B_H$)	Command	Count
RYs		RXs
RsizeY		RsizeX
(Pattern 0)		
(Pattern 1)		
…		
(Pattern n)		

[Description of Command]

Command		Description
BltDraw	42_H	Draws an 8/16/32-bit pixel pattern
DrawBitmap	43_H	Draws a binary bit map pattern. "0" is drawn by a color to be set for the transparent or background color register; "1" is drawn by a color to be set for the foreground color register.

[Description of parameter]

Field name	Content	Effective range
Count	Parameter word count including "RYs, RXs", "RsizeY, RsizeX", and pattern.	18 to 65535
RXs	Drawing starting X coordinate in the drawing frame	-4096 to 4095
RYs	Drawing starting Y coordinate in the drawing frame	-4096 to 4095
RsizeX	X direction pixel count in a pattern	8 to 4096
RsizeY	Y direction pixel count in a pattern	8 to 4096

Data format:

Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Count	Int															
RXs	S				Int											
RYs	S				Int											
RsizeX	0				Int											
RsizeY	0				Int											

Pattern data format:

BltDraw

For 8 bits/pixel

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	Pixel 3								Pixel 2								Pixel 1								Pixel 0							
Word m	Pixel n-1								Pixel n-2								Pixel n-3								Pixel n-4							

For 16 bits/pixel

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	Pixel 1																Pixel 0															
Word m	Pixel n-1																Pixel n-2															

For 32 bits/pixel

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	Pixel 0																															
Word m	Pixel n-1																															

When the number of remaining pixels in the last word is less than 32 bits, data on the MSB is ignored in KOTTOS internally.

[Caution]

Background color transparency (BT of BC) affects the processing. Set BW (bit 18) of MDR4 and the BT bit of the BC register to "0".

Set BSH and BSV of MDR0 to "0".

Set so that "the drawing starting coordinate + Rsize" becomes greater than 4065.

When the drawing starting coordinate is a negative value, always enable rendering clipping (CX, CY of MDR0). When rendering clipping is disabled, set a positive value to the drawing start coordinate.

DrawBitmap

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31

Unlike BltDraw, DrawBitmap must create data whose word boundary is aligned horizontally in a line. When the number of remaining pixels in the last word in one line is less than 32 bits, the remaining part is ignored in KOTTOS internally.

The following shows data format in which 40-pixel wide characters are drawn using DrawBitmap.

[Data format (for first two lines) in which 40-pixel wide characters are drawn using DrawBitmap]

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31
Word 1	P32	P33	P34	P35	P36	P37	P38	P39																								
Word 2	P40	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	P51	P52	P53	P54	P55	P55	P56	P57	P58	P59	P60	P61	P62	P63	P64	P65	P66	P67	P68	P69	P70
Word 1	P71	P72	P73	P74	P75	P76	P77	P78																								

[Processing]

This command draws rectangular patterns. Specify the word count not including only header words (including RYs/RXs words and RsizeY/RsizeX words) for the Count.

To execute this command, set an 8-byte aligned value to XRES.

[Caution]

Set so that "the drawing starting coordinate + Rsize" becomes greater than 4065.

When the drawing starting coordinate is a negative value, enable rendering clipping (CX, CY of MDR0). When rendering clipping is disabled, set a positive value to the drawing start coordinate.

DrawBitmapLargeP

[Format]

31 24	23 16	15 0
DrawBitmapLargeP($2F_H$)	BltDraw (42_H)	Reserved
Count		
RYs		RXs
RsizeY		RsizeX
(Pattern 0)		
(Pattern 1)		
. . .		
(Pattern n)		

[Description of parameter]

Field name	Content	Effective range
Count	Parameter word count including "RYs,RXs", "RsizeY,RsizeX", and pattern.	18 to 16777218
RXs	Drawing starting X coordinate in the drawing frame	-4096 to 4095
RYs	Drawing starting Y coordinate in the drawing frame	-4096 to 4095
RsizeX	X direction pixel count in a pattern	8 to 4096
RsizeY	Y direction pixel count in a pattern	8 to 4096

Data format:

Data	31:30:29:28:27:26:25:24:23:22:21:20:19:18:17:16	15:14:13:12	11:10: 9 : 8 : 7 : 6 : 5 : 4 : 3 : 2 : 1 : 0
Count	0	Int	
RXs	-	S	Int
RYs	-	S	Int
RsizeX	-	0	Int
RsizeY	-	0	Int

[Processing]

This command draws rectangular patterns. Use this command when drawing a large pattern whose Count value cannot fit in 16 bits.

To execute this command, set an 8-byte aligned value to XRES.

[Caution]

Background color transparency (BT of BC) affects the processing. Set BW (bit 18) of MDR4 and the BT bit of the BC register to "0".

Set so that "the drawing starting coordinate + Rsize" becomes greater than 4065.

When the drawing starting coordinate is a negative value, enable rendering clipping (CX, CY of MDR0). When rendering clipping is disabled, set a positive value to the drawing start coordinate.

BltCopyP

[Format]

31 24	23 16	15 0
BltCopyP ($0D_H$)	Command	Reserved
SRYs		SRXs
DRYs		DRXs
BRsizeY		BRsizeX

[Description of Command]

Command		Description
TopLeft	44_H	Starts BitBlt transfer starting at the upper left coordinate.
TopRight	45_H	Performs operation equivalent to transfer from the upper right coordinate. (For SRXs=DRXs, TopLeft operation is performed; for SRYs=DRYs, BottomRight operation is performed.)
BottomLeft	46_H	Performs operation equivalent to transfer from the lower left coordinate. (For SRXs=DRXs, BottomRight operation is performed; for SRYs=DRYs, TopLeft operation is performed.)
BottomRight	47_H	Starts BitBlt transfer from the lower right coordinate.

[Description of parameter]

Field name	Content	Effective range
SRXs	Transfer source starting X coordinate in the drawing frame	0 to 4095
SRYs	Transfer source starting Y coordinate in the drawing frame	0 to 4095
DRXs	Transfer destination starting X coordinate in the drawing frame	-4096 to 4095
DRYs	Transfer destination starting Y coordinate in the drawing frame	-4096 to 4095
BRsizeX	X direction pixel count in a pattern	1 to 4096
BRsizeY	Y direction pixel count in a pattern	1 to 4096

Data format:

Data	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
SRXs	0	Int
SRYs	0	Int
DRXs	S	Int
DRYs	S	Int
BRsizeX	0	Int
BRsizeY	0	Int

[Processing]

This command copies a rectangular pattern within a drawing frame.
To execute this command, set an 8-byte aligned value to XRES.
In such a case that the copy source area and the copy destination area overlap even partially, the user must select Command appropriately. When copying a rectangular pattern in a manner in which no overlap occurs between the said areas, Fujitsu recommends that TopLeft be used.

[Caution]

Background color transparency (BT of BC) affects the processing. Set BW (bit 18) of MDR4 and the BT bit of the BC register to “0”.

BltCopyAlternateP

[Format]

31 24	23 16	15 0
BltCopyAlternateP ($0F_H$)	TopLeft (44_H)	Reserved
SADDR		
Sstride		
SRYs		SRXs
DADDR		
Dstride		
DRYs		DRXs
BRsizeY		BRsizeX

[Description of parameter]

Field name	Content	Effective range
SADDR	Transfer source drawing frame starting address (base address)	00000000h to 3FFFFFF8h
Sstride	Transfer source drawing frame horizontal pixel count	2 to 4096 (8-byte aligned)
SRXs	Transfer starting X coordinate in the transfer source drawing frame	0 to 4095
SRYs	Transfer starting Y coordinate in the transfer source drawing frame	0 to 4095
DADDR	Transfer destination drawing frame starting address	00000000h to 3FFFFFF8h
Dstride	Transfer destination drawing frame horizontal pixel count	2 to 4096 (8-byte aligned)
DRXs	Transfer starting X coordinate in the transfer destination drawing frame	-4096 to 4095
DRYs	Transfer starting Y coordinate in the transfer destination drawing frame	-4096 to 4095
BRsizeX	X direction pixel count in the rectangular area to be transferred	1 to 4096
BRsizeY	Y direction pixel count in the rectangular area to be transferred	1 to 4096

Data format:

Data	31	30	29	28	27	26	25:24:23:22:21:20:19:18:17:16	15	14	13	12	11:10: 9 : 8 : 7 : 6 : 5 : 4 : 3	2	1	0
SADDR	0						Int								0
Sstride	0											Int			
SRXs	-								0			Int			
SRYs	-								0			Int			
DADDR	0						Int								0
Dstride	0											Int			
DRXs	-								S			Int			
DRYs	-								S			Int			
BRsizeX	-								0			Int			
BRsizeY	-								0			Int			

[Processing]

This command copies a rectangular pattern between different drawing frames. In a case that the copy source area and the copy destination area overlap even partially, copying cannot be performed correctly.
To execute this command, set an 8-byte aligned value to XRES.
Set an 8-byte aligned value to SStride and DStride.

[Caution]

Background color transparency (BT of BC) affects the processing.
Set BW (bit 18) of MDR4 and the BT bit of the BC register to 0.

BltCopyAltAlphaMapP

[Format]

31 24	23 16	15 0
BltCopyAltAlphaMapP(1F$_H$)	Normal (01$_H$)	Reserved
SADDR		
Sstride		
SRYs		SRXs
AMADDR		
DRYs		DRXs
BrsizeY		BRsizeX

31 24	23 16	15 0
BltCopyAltAlphaMapP(1F$_H$)	ABR (00$_H$)	Reserved
SADDR		
Sstride		
SRYs		SRXs
BlendStride		
BlendRYs		BlendRXs
DRYs		DRXs
BrsizeY		BRsizeX

[Description of Command]

Command		Description
Normal	01$_H$	Can specify an alpha map storage address. Stride of the Alpha map is set to the same value as the transfer horizontal pixel count (BRsizeX). The stride has no restriction on 8-byte alignment.
ABR	00$_H$	Specifies an alpha map frame.

[Description of parameter]

Field name	Content	Effective range
SADDR	Transfer source drawing frame starting address (base address)	00000000$_H$ to 3FFFFFF8$_H$
Sstride	Transfer source drawing frame horizontal pixel count	2 to 4096
SRXs	Transfer starting X coordinate in the transfer source drawing frame	0 to 4095
SRYs	Transfer starting Y coordinate in the transfer source drawing frame	0 to 4095
AMADDR	Specifies an alpha map storage address. A value specified in this field rewrites ABR.	00000000$_H$ to 3FFFFFF8$_H$ (64-bit aligned)
Blendstride	Alpha map frame horizontal pixel count (The base address is set for ABR.)	2 to 4096
BlendRXs	Reference starting X coordinate in the alpha map drawing frame	0 to 4095
BlendRYs	Reference starting Y coordinate in the alpha map drawing frame	0 to 4095
DRXs	Transfer starting X coordinate in the transfer destination drawing frame	-4096 to 4095
DRYs	Transfer starting Y coordinate in the transfer destination drawing frame	-4096 to 4095
BRsizeX	X direction pixel count in the rectangular area to be transferred	1 to 4096
BRsizeY	Y direction pixel count in the rectangular area to be transferred	1 to 4096

Data format:

Data	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13	12	11 10 9 8 7 6 5 4 3	2	1	0
SADDR	0	Int					0
Sstride	0			Int			
SRXs	-		0	Int			
SRYs	-		0	Int			
AMADDR	0	Int					0
Blend stride	0			Int			
BlendRXs	-		0	Int			
BlendRYs	-		0	Int			
DRXs	-		S	Int			
DRYs	-		S	Int			
BRsizeX	-		0	Int			
BRsizeY	-		0	Int			

[Processing]

This command performs alpha blending for the source specified for SADDR, SStride, SRXs, SRXy and the alpha map specified for ABR (alpha base address), BlendStride, BlendRXs, BlendRYs, and copies the result of the alpha blending to the destination specified by FBR (frame buffer base address), XRES (X resolution), DRXs, DRYs. Copying is performed using TopLeft. In a case that the copy source area and the copy destination area overlap even partially, copying cannot be performed correctly.

When logical operation is set to the display list BltCopyAltAlphaMapP, it is ignored.

To execute this command, set an 8-byte aligned value to XRES.

Set an 8-byte aligned value to SStride and BlendStride.

[Caution]

Background color transparency (BT of BC) affects the processing.

Set BW (bit 18) of MDR4 and the BT bit of the BC register to “0”.

BltCopyCompressedP

[Format]

31 24	23 16	15 0
BltCopyCompressedP($2D_H$)	TopLeft (44_H)	Reserved
SADDR		
DADDR		
Dstride		
DRYs		DRXs
BRsizeY		BRsizeX

[Description of parameter]

Field name	Content	Effective range
SADDR	Storage address for transfer source and compressed pattern data	00000000_H to $3FFFFFF8_H$
DADDR	Transfer destination drawing frame starting address	00000000_H to $3FFFFFF8_H$
Dstride	Transfer destination drawing frame horizontal pixel count	2 to 4096
DRXs	Transfer starting X coordinate in the transfer destination drawing frame	-4096 to 4095
DRYs	Transfer starting Y coordinate in the transfer destination drawing frame	-4096 to 4095
BRsizeX	X direction pixel count in the rectangular area to be transferred after decompression of pattern data.	8 to 4096 (multiple of 8)
BRsizeY	Y direction pixel count in the rectangular area to be transferred after decompression of pattern data.	to 4096 (multiple of 8)

[Processing]

This command decompresses and copies compressed pattern data. In a case that the copy source area and the copy destination area overlap even partially, copying cannot be performed correctly. Data is compressed in units of 8 pixels, and so the data size must be a multiple of "8".

To execute this command, set an 8-byte aligned value to XRES. Set an 8-byte aligned value to DStride.

[Caution]

Background color transparency (BT of BC) affects the processing.

Set BW (bit 18) of MDR4 and the BT bit of the BC register to "0".

When the pattern data is the compressed format, the CO bit of MDR0 (bit24) is only " 1" .

Set the CO bit to "1" when this command is used.

BltCopyCompAlphaMapP

[Format]

31 24	23 16	15 0
BltCopyCompAlphaMapP(2E$_H$)	TopLeft (44$_H$)	Reserved
SADDR		
AMADDR		
DRYs		DRXs
BRsizeY		BRsizeX

[Description of parameter]

Field name	Content	Effective range
SADDR	Storage address for transfer source and compressed pattern data	00000000$_H$ to 3FFFFFF8$_H$
AMADDR	Specifies an alpha map storage address. A value specified in this field rewrites ABR.	00000000$_H$ to 3FFFFFF8$_H$ (64-bit aligned)
DRXs	Transfer starting X coordinate in the transfer destination drawing frame	-4096 to 4095
DRYs	Transfer starting Y coordinate in the transfer destination drawing frame	-4096 to 4095
BRsizeX	X direction pixel count in the rectangular area to be transferred after decompression of pattern data.	8 to 4096 (multiple of 8)
BRsizeY	Y direction pixel count in the rectangular area to be transferred after decompression of pattern data.	8 to 4096 (multiple of 8)

Data format:

Data	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR	0	Int																												0		
AMADDR	0	Int																												0		
DRXs	-																S	Int														
DRYs	-																S	Int														
BRsizeX	-																0	Int														
BRsizeY	-																0	Int														

[Processing]

This command decompresses and copies compressed pattern data. In a case that the copy source area and the copy destination area overlap even partially, copying cannot be performed correctly. Data is compressed in units of 8 pixels, and so the data size must be a multiple of "8".

To execute this command, set an 8-byte aligned value to XRES. Stride of the alpha map data is the same as BRsizeX, and has no restriction on 8-byte alignment.

[Caution]

Background color transparency (BT of BC) affects the processing.

Set BW (bit 18) of MDR4 and the BT bit of the BC register to "0".

When the pattern data is the compressed format, the CO bit of MDR0 (bit24) is only " 1" .

Set the CO bit to "1" when this command is used.

SetRegister

[Format]

31 24	23 16	15 0
SetRegister ($F1_H$)	Count	Address
(Val 0)		
(Val 1)		
…		
(Val n)		

[Description of parameter]

Field name	Content	Effective range
Count	Number of registers that are set	01_H to FF_H (only the range in which registers exist)
Address	Address of the starting register that is set	0000_H to 0153_H (only addresses where registers exist)

[Register updated by other than the specified registers]

Module	Register address	Register name	Description
Vertex Reader	0002_8420_H	S_MDR0	Updated when MDR0 setting is included.
	0002_8424_H	S_MDR1	Updated when MDR1 setting is included.
	0002_8428_H	S_MDR2	Updated when MDR2 setting is included.
	0002_8430_H	S_MDR4	Updated when MDR4 setting is included.
	0002_8440_H	S_FBR	Updated when FBR setting is included.
	0002_8444_H	S_XRR	Updated when XRR setting is included.

[Processing]

This command sets data to successive registers in the rendering engine.

When setting data to two or more registers, the register address of data of the second or later is incremented by 1. Operation is not guaranteed when data is written to undefined addresses.

LoadFirm

[Format]

31 24	23 16	15 14 8	7 0
LoadFirm (18_H)	Displaylist (00_H)	Reserved	
InstRAM Address		Count	
Code1		Code0	
…			
Code n		Code n-1	

31 24	23 16	15 14 8	7 0
LoadFirm (18_H)	Memory (01_H)	Reserved	
InstRAM Address		Count	
LocalMemAddress			

[Description of Command]

Command		Description
Displaylist	00_H	Transfers the subsequent display list for 32-bit word count as firmware.
Memory	01_H	Transfers the subsequent words for 32-bit word count as memory address and as firmware.

[Description of parameter]

Field name	Content	Effective range
InstRAM Address	Write starting address of RAM that stores pixel processing firmware	0 to 510 (64-bit boundary address in units of 32-bit words)
Count	Number of 32-bit words transferred as pixel processing firmware (write is performed in units of 64 bits)	1 to 512
LocalMemAddress	AXI space address where pixel processing firmware is stored	00000000_H to $3FFFFFF8_H$

Data format:

Data	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
InstRAM Address	0	Int	0	
Count	0	Int		

Data	31 1 0	
LocalMemAddress	Address	0

Code format in display list (in units of 32 bits):

31 16	15 0
Code 1	Code 0
Code 3	Code 2
…	
Code n-1	Code n-2

Code format in local memory (in units of 64 bits):

63 48	47 32	31 15	0
Code 3	Code 2	Code 1	Code 0
Code 7	Code 6	Code 5	Code 4
…			
Code n-1	Code n-2	Code n-3	Code n-4

[Processing]

This command loads the pixel processing program to the instruction RAM inside KOTTOS. This command selects between the mode where code is transferred from the display list and the mode where code is read and transferred from local memory.

The instruction RAM of the pixel processing module is accessed in units of 64 bits. If the bit count is less than 64 bits, padding (undefined data) is written.

RegTexture

[Format]

31 28	24 23 20	16 15 12	8 4	0
RegTexture (19_H)	Base (00_H)	Reserved		TexID
0	Base address			0
0	SizeT	0	SizeS	
TL	Reserved	FMT	BA	Reserved	CMP	Reserved	BPP

31 28	24 23 20	16 15 12	8 6 4	0
RegTexture (19_H)	State (01_H)	Reserved		TexID
WRAPS	WRAPT	0	MAGFL 0 MINFL	0	PC
BDRCOL				

[Description of Command]

Command		Description
Base	00_H	Sets base information of texture. Up to eight pieces of information can be registered using TexID.
State	01_H	Sets state information of texture mapping. Up to eight pieces of information can be registered using TexID.

[Description of parameter]

Field name		Effective range
TexID	Texture information table entry number	0 to 7

Data format:

Data	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Base Address	-0 Address 0
SizeS/ SizeT	0 Int 0 Int

Parameter for base information:

BPP	(Bit Per Pixel) Specifies the pixel format of a texture.		
	000:	8 bits / pixel	8-bit palette code assuming palette code
	001:	RGB5_A1	16-bit color data: 5 bits for each of RGB components, 1 bit for "A" component
	010:	RGBA8	32-bit color data: 8 bits for each of RGBA components
	011:	RGB5_6_5	16-bit color data: 5 bits for R, 6 bits for G, and 5 bits for B
	100:	RGBA4	16-bit color data: 4 bits for each of RGBA components
	101:	16 bits / pixel	16-bit value assuming LUMINANCE_ALPHA
CMP	(Compress) Specifies the compressed format of a texture.		
	00:	PLAIN	Uncompressed format
	01:	COMPRESSED	Compressed format
	10:	PALETTE4	4-bit palette code format
	11:	PALETTE8	8-bit palette code format
BA	(Bilinear Accelerate) Specifies the special format used to perform fast access using bilinear filtering. This parameter is enabled only for uncompressed format (PLAIN).		

	0:	Disable	Normal format
	1:	Enable	Format for bilinear fast mode
FMT	(Format) Specifies the type of a texture.		
	000:	ALPHA	Assumes 8 bits/pixel
	001:	LUMINANCE	Assumes 8 bits/pixel
	010:	LUMINANCE_ALPHA	Assumes 16 bits/pixel: 8 bits/pixel for LUMINANCE and ALPHA respectively.
	011:	INTENSITY	Assumes 8 bits/pixel
	100:	RGB	Assumes 16-bit color data or 32-bit color data
	101:	RGBA	Assumes 16-bit color data or 32-bit color data
Reserved	Specify 0 for future compatibility.		
TL	(Tile pattern) Indicates that the pattern is used for tiling drawing.		
	0:	Texture	Pattern is referenced as texture.
	1:	Tile	Pattern is referenced as tile.

Parameters for state information:

PC	(Texture coordinates Perpective Correction) Performs perspective correction for a texture when drawing 3D.		
	0:	Disable (initial value)	Does not correct.
	1:	Enable	Corrects
MINFL	(Minificate texture Filtering) Sets how to interpolate when minifying texture.		
	000:	NEAREST (initial value)	Point sampling
	001:	LINEAR	Bilinear filtering
	010:	NEAREST_MIPMAP_NEAREST	Mipmap
	011:	LINEAR_MIPMAP_NEAREST	Bilinear filtering, mipmap
	110:	NEAREST_MIPMAP_LINEAR	Performs interpolation between mipmaps.
	111:	LINEAR_MIPMAP_LINEAR	Trilinear filtering
MAGFL	(Magnificate texture Filtering) Sets how to interpolate when magnifying texture.		
	000:	NEAREST (initial value)	Point sampling
	001:	LINEAR	Bilinear filtering
WRAPT	(Texture Wrapping mode for T) Sets how to wrap a texture coordinate for T direction.		
	000:	CLAMP_TO_EDGE (initial value)	
	001:	REPEAT	
	010:	BORDER	
	011:	CLAMP	
	100:	MIRRORED_REPEAT	
WRAPS	(TextureWrapping mode for S) Sets how to wrap a texture coordinate for S direction.		
	000:	CLAMP_TO_EDGE (initial value)	
	001:	REPEAT	
	010:	BORDER	
	011:	CLAMP	
	100:	MIRRORED_REPEAT	

Format of BDRCOL

8-bit color mode:

Bit 7-0	BC8 (Border 8 bits Color)
	Sets a texture border color using an 8-bit indirect color (color index code).
Bit 31-8	Unused bits

16-bit color mode (when CO bit of MDR0 is "0"):

Bit 15-0	BC16 (Border 16 bits Color)
	Sets a texture border color using a 16-bit direct color.
Bit 31-16	Unused bits

32-bit color mode (when CO bit of MDR0 is "0"):

Bit 31-0	BC32 (Border 32 bits Color)
	Sets a texture border color using a 32-bit direct color.

16-bit color mode (when CO bit of MDR0 is "1"):

Bit 15-0	BC16 (Border 16 bits Color)
	Sets a texture border color using a 16-bit direct color.
Bit 31-16	Unused bits

32-bit color mode (when CO bit of MDR0 is "1"):

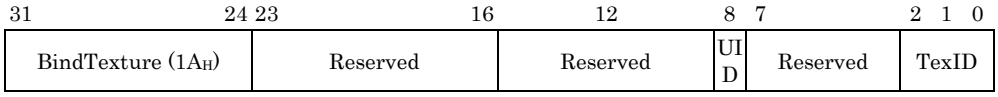
Bit 31-0	BC32 (Border 32 bits Color)
	Sets a texture border color using a 32-bit direct color.

[Processing]

This command registers texture pattern information used as texture, to the texture information table. It registers two types of information: base information and state information.

BindTexture

[Format]



[Description of parameter]

Field name		Effective range
UID	ID of texture unit for setting	0, 1
TexID	Texture information table entry number of the texture assigned to the texture unit specified by UnitID	0t o 7

[Processing]

This command specifies the texture pattern used as texture 0 or texture 1 (texture ID registered by RegTexture).

SetFog

[Format]

31 28	24 23 20	16 15 12	8 4 0
SetFog ($1B_H$)	Table (00_H)	Reserved	Count
fs0			
df0			
fs1			
dF1			
.			
.			
fs31			
df31			

[Description of parameter]

Field name		Effective range
Count	Number of entries (number of initial value/coefficient pairs) Specify 32.	32
fs	Initial value	-8192 to 8191
df	Incremental value	-8192 to 8191

Data format:

Data	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Fs	S Int	Frac
dF	S Int	Frac

[Processing]

This command registers the fog inclination value to the texture information table. It divides the min-max range of the Z value or of the fog coordinate (selected by FOGCRD of MDR7) into 32 units, and then sets the initial value and incremental value of the fog factor of each divided unit. When performing operation equivalent to OpenGL, calculate the initial value and incremental value based on the fog mode (LINEAR, EXP, EXP2) and Density specified by OpenGL, and then set the result in 32 entries (elements in table).

Regarding the fog factor, the minimum value is clamped to 0% (0.0); the maximum value to 100% (1.0). When implementing START and END using LINEAR mode of OpenGL, set the table so that 1.0 is set in START and 0.0 in END. When a value exceeding 1.0 is set as the initial value, a point (START) where the coefficient starts decreasing at an arbitrary position can be set.

For the fog factor f, set 100% (1.0) as FF_H, and 0% (0.0) as 00_H.

6.5 Application Note

6.5.1 Host Interface

Endian

KOTTOS uses little endian in principle. Register addresses described in this document are little endian byte addresses. When using a big endian CPU, note that addresses are different from those shown in this document.

6.5.2 Initialization Procedure for Hardware

Loading of firmware for geometry processing

Load the geometry processing firmware to PCI memory space.

This operation is not required when using only rendering display list.

(1) Place geometry processing firmware in PCI memory space beforehand. KOTTOS does not get involved in this placement.

(2) Set a reset vector used for the geometry processor. Write the address where geometry processing firmware is placed, to 000F_FFFC$_H$ in PCI memory space BAR2. KOTTOS does not get involved in this setting of the reset vector.

(3) Write to the FRHALT register to start up the geometry processor.

(4) Reference the FR_ST register and check that the status becomes IDLE.

Table 6.5.1 Registers Used to Start Up Geometry Processor

Setting	Register	Field
Start-up of geometry processor	FRHALT	FRHALT
Determination of end of start-up of geometry processor	FR_ST	ST

Loading of firmware for pixel processing

Always load the firmware for pixel processing beforehand when using geometry display list and rendering display list.

Pixel processing firmware is loaded by the display list LoadFirm. The LoadFirm contains Displaylist command and Memory command. For easier loading, use Displaylist command for the first loading. To replace to use multiple pieces of Pixel processing firmware stored in memory beforehand, use Memory command.

6.5.3 Basic Drawing Procedure

Setting of frame buffer

KOTTOS uses memory space connected to Carmine to draw. Pixel drawing space necessary for drawing is called “frame buffer”.

Frame buffer is a rectangular image data area for drawing.

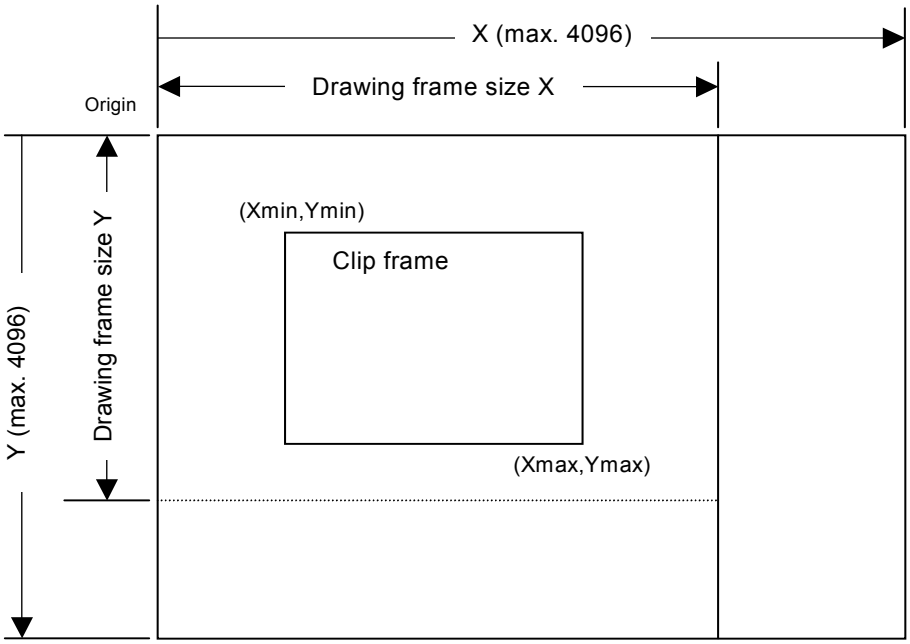


Fig. 6.1 Concept of Frame Buffer

Drawing frame is treated as a 2D coordinate system with the upper left point set as the origin, as shown in the figure above. Coordinate of max. “4096 × 4096” can be treated. The horizontal pixel count must be a multiple of “16”.

Frame buffer is allocated in graphics memory by setting the drawing frame origin address and the X direction resolution (pixel count) to a register. The size of Y direction needs not be set, but the user must be careful not to cause the maximum Y coordinate to overlap with another area when drawing.

Drawing address = Origin address + (Drawing frame size X × Y coordinate) + X coordinate

The above expression is the one for 8 bits/pixel. For 16 bits/pixel, multiply the value of the drawing frame size X and the X coordinate by 2; for 32 bits/pixel, multiply them by 4. This is because 1 pixel is 2 bytes and 4 bytes, respectively.

Table 6.5.2 Byte Count per Pixel

Color mode	Byte count/pixel
8-bit color index	1
16-bit color	2
32-bit color	4

A function called “clipping” can be used that does not draw outside the specified rectangular area. Specify the clip frame to the upper left coordinate and lower right coordinate.

When clipping is enabled, a negative coordinate can also be specified. This allows KOTTOS to draw such a graphics as the one, that is, only a part of which is stored in the frame buffer. When drawing a graphics onto a negative coordinate without specifying clipping, the drawing is performed on a horizontally wrapped around position or in a vertically protruded memory area.

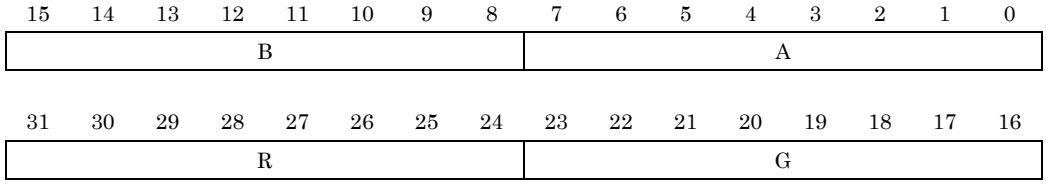
Table 6.5.3 Registers to Set Frame Buffer

Setting	Register	Field
Base address of frame buffer	FBR	-
Horizontal pixel count of frame buffer	XRR	-
Color mode	MDR0	CF
Enable/disable of clipping	MDR0	CX, CY
Clip frame upper left X coordinate	CXMIN	-
Clip frame upper left Y coordinate	CYMIN	-
Clip frame lower right X coordinate	CXMAX	-
Clip frame lower right Y coordinate	CYMAX	-

[Memory data format]

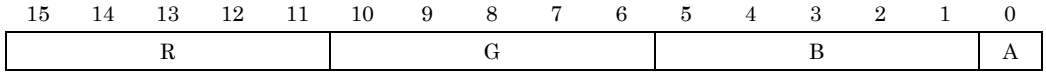
32-bit direct color (32 bits/pixel)

Color data represented using each 8-bit R, G, B, and A. Bits 7 to 0 are normally meaningless, but they are used as a blend value or display block processing control bits when alpha blend or texture is used. This color data is always stored in RGBA format irrespective of the CO bit of the MDR0 register.



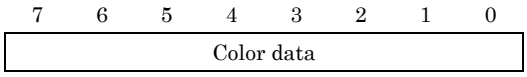
16-bit direct color (16 bits/pixel)

Color data represented using each 5-bit RGB. Bit 0 is normally meaningless, but it is used as the stencil processing control bit for texture data. This color data is always stored in RGBA format irrespective of the CO bit of the MDR0 register.



8-bit color index (8 bits/pixel)

8-bit color code. R, G and B are not distinguished. This color code is assumed to be treated as palette reference index code in external display modules.



Setting of polygon flag buffer

When using a concave polygon, a polygon flag buffer must be prepared as the drawing work area. Concave polygon means a polygon drawn using G_Begin (Polygon or nclip_Polygon) to G_End or using SetVertex:FlagTriangleFan to Draw:PolygonEnd.

Polygon drawing flag buffer is, as with Z buffer, of the same shape as drawing frame. A 1-bit memory area is needed per pixel, and additionally an area for X resolution must be allocated before and after the memory area. Polygon drawing flag buffer must be cleared before drawing after allocating memory area.

DrawRectP contains ClearPolyFlag, a command dedicated to clearing polygon drawing flag buffer. As with normal DrawRectP, ClearPolyFlag also clears only the specified range from the specified coordinate, and so the area for XRES size allocated before and after the memory area must also be cleared. Note that when executing ClearPolyFlag, disable rendering clipping.

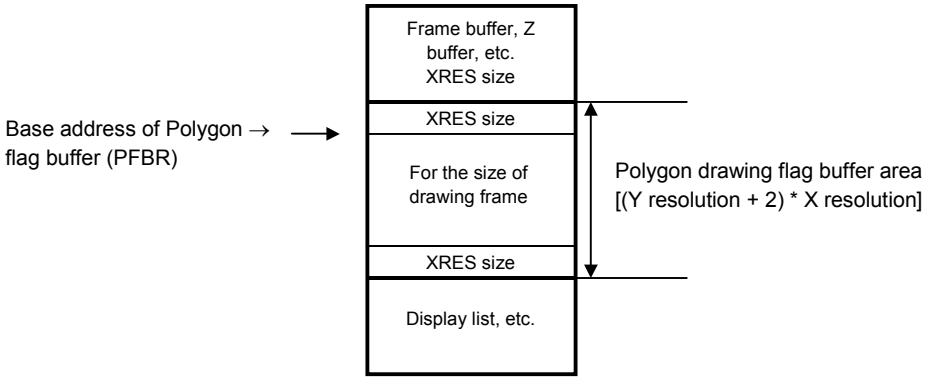


Fig. 6.2 Setting Example of Polygon Flag Buffer

[Memory data format]

Polygon drawing flag

Binary data that represents 1 pixel using 1 bit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16

Table 6.5.4 Registers to Set Polygon Flag Buffer

Setting	Register	Field
Base address of polygon flag buffer	PFBR	-
Horizontal pixel count of frame buffer	XRR	-

Table 6.5.5 Display List to Clear Polygon Flag Buffer

Setting	Display list	Command
Clearing of polygon flag buffer	DrawRectP	ClearPolyFlag

Coral mode and OpenGL mode

KOTTOS has two modes: an OpenGL mode in which the function can be fully used, and a Coral-compatible mode, which is an existing Graphics Display Controller. In the Coral and OpenGL modes, how to set the function is partially different. Using IDFM of the GMDR0 register switches these modes.

Table 6.5.6 Difference between Coral and OpenGL Mode Settings

	Coral mode	OpenGL mode
Setting of vertex element	GMDR0: C,Z,ST	IVAOGL, G_VertexSetting
Setting of element format	GMDR0: DF,CF	IDFOGL
Setting of projection transformation	GMDR0: F	G_MatrixSetting
Setting of culling	GMDR2	G_PolygonSetting

After mode switching, reset the contents shown in the above table. When using the Coral mode setting for the OpenGL mode, or vice versa, the setting may not be correctly reflected.

Each mode has functions that cannot be used.

- Functions that cannot be used in OpenGL mode
  – Top-left rule not-applicable mode (GMDR2E: TL)
  – Shadow primitive drawing mode (GMDR2E: SP)

- Functions that cannot be used in Coral mode
  - Quad graphics (G_Begin:Quads, Quads_Strip, nclip_Quads, nclip_Quads_Strip)
  - Function that uses vertex elements that cannot be set using GMDR0. (It can be set using IVAOGL or IDFOGL.)
    Example: Back face color, lighting processing, index mode
  - PolygonMode (G_PolygonSetting: POMF, POMB)
  - PolygonOffset (G_PolygonSetting: POFP, POFL, POFF, POBP, POBL, POBF)

The sequence of elements for color R, G, B, and A is different in the Coral and OpenGL modes. When switching between the modes, the CO bit of the MDR0 register is also usually changed concurrently.

Table 6.5.7 Register for setting color element sequence format

Setting	Register	Field
Sequence format of color elements	MDR0	CO

Setting of coordinate transformation matrix

Coordinate of an original graphics is called “object coordinate (OC)”; coordinate after MVP (Model-View-Projection) transformation is called “clip coordinate (CC)”. Set an MVP matrix used to transform from the “object coordinate” to the “clip coordinate”. As the numerical format of each element of the matrix, either floating point number or fixed point number can be selected. The expression to calculate the MVP transformation is as follows.

$$\begin{pmatrix} Xcc \\ Ycc \\ Zcc \\ Wcc \end{pmatrix} = \begin{pmatrix} m00 & m10 & m20 & m30 \\ m01 & m11 & m21 & m31 \\ m02 & m12 & m22 & m32 \\ m03 & m13 & m23 & m33 \end{pmatrix} \begin{pmatrix} Xoc \\ Yoc \\ Zoc \\ Woc \end{pmatrix}$$

Table 6.5.8 Registers to Set Data Format

Setting	Register	Field
OpenGL/Coral mode	GMDR0	IDFM
Data format (OpenGL mode)	IDFOGL	DFV
Data format (Coral mode)	GMDR0	DF

Table 6.5.9 Display List to Set MVP Matrix

Setting	Display list	Command
MVP matrix (OpenGL-compatible arrangement sequence)	G_LoadMatrixMVP	ColRow
MVP matrix (Coral-compatible arrangement sequence)		RowCol

Setting of coordinate transformation

By dividing X, Y and Z components of the clip coordinate by "W" component, a graphics is transformed to a 3-dimensional and perspective graphics. This transformation is called "perspective transformation", and the coordinate after the perspective transformation is called "Normalized Device Coordinate (NDC)". Perspective transformation is not necessary only when rotating or translating.

$$\begin{pmatrix} Xndc \\ Yndc \\ Zndc \end{pmatrix} = 1/Wcc \begin{pmatrix} Xcc \\ Ycc \\ Zcc \end{pmatrix}$$

Table 6.5.10 Display List to Set Perspective Transformation (OpenGL Mode)

Setting	Display list	Command	Field
Enable/disable of perspective transformation	G_MatrixSetting	-	PROJ

Table 6.5.11 Register to Set Perspective Transformation (Coral Mode)

Setting	Register	Field
Enable/disable of perspective transformation (Coral mode)	GMDR0	F

[Caution]

In the OpenGL mode, use the G_MatrixSetting; in the Coral mode, use GMDR0. When using G_MatrixSetting and GMDR0 together, the setting may not be correctly reflected.

Setting of clipping

Drawing a part out of the screen causes writing to graphics memory outside the assumed range and performance degradation. Clipping is a function to prevent this. Two types of clipping are supported: view volume clipping, and rendering clipping.

Setting of view volume clipping

View volume clipping is for setting a drawing range in clip coordinate space. This means a setting of a viewable range and the range is called "view volume". When perspective transformation is enabled, view volume is a space spread toward the depth direction.

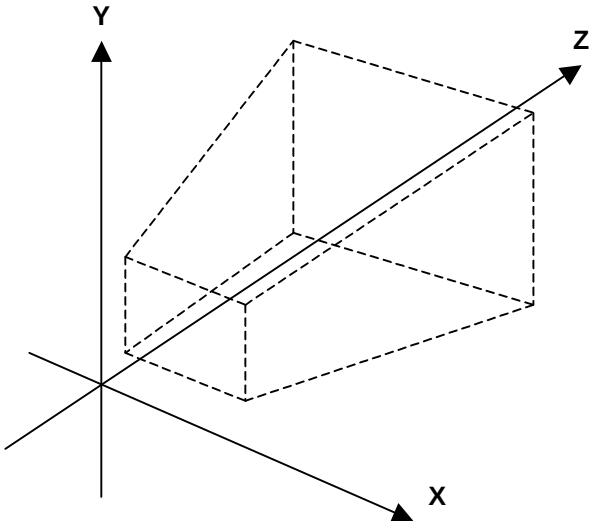


Fig. 6.3 View Volume in Clip Coordinate Space

For example, for the X coordinate, whether or not clipping is required is determined as follows:

Clip occurrence condition on the X minimum value → $Xcc/Wcc < XMIN$

Clip occurrence condition on the X maximum value→ $Xcc/Wcc < XMAX$

Set the XMIN, XMAX, YMIN,and YMAX using G_ViewVolumeXYClip. Similarly, set "Z" using G_ViewVolumeZClip.

"W" component gets involved with view volume, and so set the "W" range, too. Division is performed by transformation of X_{ndc}=Xcc/Wcc, and so "W" must be greater than "0" ("W>0"). In this case, if too small a value is allowed for "W", NDC coordinate becomes too great. To prevent this, set the minimum value using G_ViewVolumeWClip. For the "W", the maximum value needs not be set.

Whether or not to perform view volume clipping is specified when drawing graphics described later.

Table 6.5.12 Display Lists to Set View Volume

Setting	Display list	Command	Field
XY maximum/minimum clip coordinate	G_ViewVolumeXYClip	-	-
Z maximum/minimum clip coordinate	G_ViewVolumeZClip	-	-
W minimum clip coordinate	G_ViewVolumeWClip	-	-

Setting of rendering clipping

View volume clipping is performed in clip coordinate space by performing floating point operation. Therefore, it is difficult to set so that the view volume clipping will fit perfectly the frame buffer on the device coordinate (see "view port transformation" described later). In this case, separately from view volume clipping, set rendering clipping. The rendering clipping is to clip on the device coordinate.

View volume is usually set larger than the frame buffer, and on the other, rendering clipping is set to a range that fits perfectly the frame buffer.

Rendering clipping can be performed faster than view volume clipping, but when the device coordinate range is exceeded when performing view port transformation described later, clipping cannot be performed correctly. Set view volume clipping so that the device coordinate range will not be exceeded.

Table 6.5.13 Registers to Set Rendering Clipping

Setting	Register	Field
Enable/disable of X direction of rendering clipping	MDR0	CX
Enable/disable of Y direction of rendering clipping		CY
X minimum value of rendering clipping	CXMIN	CLIPXMIN
X maximum value of rendering clipping	CXMAX	CLIPXMAX
Y minimum value of rendering clipping	CYMIN	CLIPYMIN
Y maximum value of rendering clipping	CYMAX	CLIPYMAX

Setting of culling

As a 3D object is usually created as convex polygon, it is unnecessary to draw a triangle whose rear face can be seen from the view point. Processing to erase a triangle of rear face before drawing it is called "culling". It can be set whether or not to perform culling (erasing) for front face and rear face respectively.

Which face is front is set based on the sequence to specify vertex. The user can select between the face where vertices are specified counterclockwise as front face and the face where vertices are specified clockwise as front face. As with Coral, it is defined by the direction as viewed from XY plane in the space where greater Y coordinate is shown downward. Note that specification of front and rear faces changes when using a space where smaller Y coordinate such as OpenGL is shown downward.

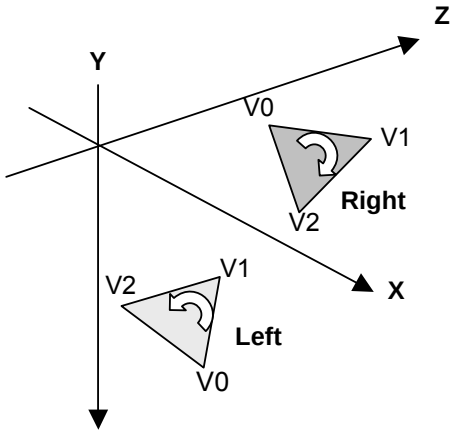


Fig. 6.4 Definition of Clockwise/Counterclockwise Rotation

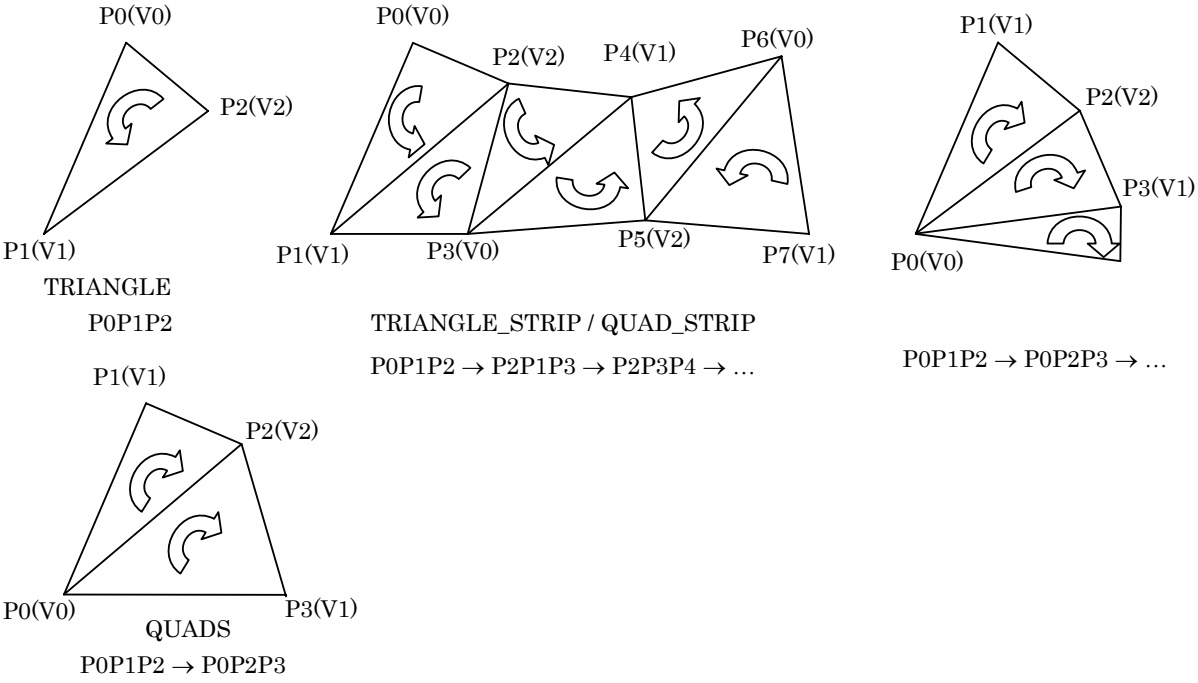


Fig. 6.5 Definition of Vertex Sequence for Each Triangle

Culling is set using G_PolygonSetting, but it can also be set using the GMDR2/GMDR2E register for compatibility with Coral. In point and straight line drawing, culling is not necessary, and the setting for culling is ignored.

[Caution]

In the OpenGL mode, use G_PolygonSetting; in the Coral mode, use the GMDR2/GMDR2E register. When using G_PolygonSetting and the GMDR2/GMDR2E register together, the setting may not be correctly reflected.

Table 6.5.14 Display List to Set Culling (OpenGL Mode)

Setting	Display list	Command	Field
Enable/disable of front face culling	G_PolygonSetting	-	CLF
Enable/disable of rear face culling			CLB
Definition of front face			CLD

Table 6.5.15 Register to Set Culling (Coral Mode)

Setting	Register	Field
Enable/disable of front face culling	GMDR2/GMDR2E	CF
Definition of front face		FD

Setting of view port transformation

Transformation to adjust NDC to a coordinate that fits the frame buffer is called “view port transformation”. A coordinate after the transformation is called “Device Coordinates (DC)”.

Setting of data format is the same as MVP matrix (***Table 6.5.8***).

Xdc = X_Scaling*Xndc + X_Offset
Ydc = Y_Scaling*Yndc + Y_Offset
Zdc = Z_Scaling*Zndc + Z_Offset

Table 6.5.16 Display Lists to Set View Port Transformation

Setting	Display list	Command	Field
View port transformation of XY	G_Viewport	-	-
View port transformation of Z	G_DepthRange	-	-

Setting of DC OFFSET

Usually, DC OFFSET setting is not needed. When using a coordinate system where smaller Y coordinate is shown "downward" and greater Y coordinate is shown "upward", as DC coordinate system, set the DC OFFSET. To specify a coordinate system where Y direction height is 100 pixels, as inversed "Y" device coordinate system, set "100" to "Y" OFFSET and set sign inversion to "ON", which allows drawing a graphics of inversed Y direction.

X = OFFSET + (sign inversion/non-inversion) Xdc

Y = OFFSET + (sign inversion/non-inversion) Ydc

DC-OFFSET registers are special registers, and cannot be set using display list such as SetRegister. Set a value by accessing directly from the host CPU after checking that the status register (GCTR) in KOTTOS is IDLE.

Table 6.5.17 Registers to Set DC OFFSET

Setting	Register	Field
DC X coordinate transformation of POINT	DC-OFFSET-PX	TOPX,OFFSET
DC Y coordinate transformation of POINT	DC-OFFSET-PY	TOPY,OFFSET
DC X coordinate transformation of LINE	DC-OFFSET-LX	TOLX,OFFSET
DC Y coordinate transformation of LINE	DC-OFFSET-LY	TOLY,OFFSET
DC X coordinate transformation of TRIANGLE	DC-OFFSET-TX	TOTX,OFFSET
DC Y coordinate transformation of TRIANGLE	DC-OFFSET-TY	TOTY,OFFSET

Setting of vertex element for object coordinate data

For object vertex data, color, texture, normal vector, and fog coordinate as well as object coordinate (X, Y, Z, W) can be set (each element is described later). Formats such as floating point number and fixed point number can be selected for each vertex element, but the settable format depends on the vertex element.

Table 6.5.18 Registers to Set Vertex Element Format and Enable/Disable of Element

Setting	Register	Field
Data format (OpenGL mode)	IDFOGL	DFV
Enable/disable of Vertex element (OpenGL mode)	IVAOGL	All fields
Data format (Coral mode)	GMDR0	DF
Enable/disable of vertex element (Coral mode)	GMDR0	C, Z, ST, CF

Table 6.5.19 Display List to Set Enable/Disable of Vertex Element (Not Recommended)

Setting	Display list	Command
Vertex element	G_VertexSetting	-

Input of drawing graphics

Table 6.5.20 Display Lists to Input Drawing Graphics

Input content	Display list	Command
Drawing graphics and enable/disable of view volume clipping	G_Begin	Various commands
Object vertex data	G_Vertex	-
Each element of object vertex data	Element value	-
End of graphics	G_End	-

There are several points to note when applying the depth test, texture and fog described later in the polygon (Polygon, nclip_Polygon). The parameter change in each function is calculated from several vertex parameters extracted as representative points from the entire polygon that is treated as one face. Therefore, there are the following restrictions:

1. All vertex parameters must be values that can be represented by "an arbitrary one vertex value + single rate of change". Mathematically, this means that all vertex parameters are on the same plane.
2. When the difference between vertex parameter values is large, the calculated rate of change becomes large and may exceed the parameter setting range.
3. Gouraud shading cannot be applied.

When using the above drawing effects for the polygon, adjust G_DepthRange (depth test), G_LoadMatrixMV (fog), and texture coordinate and use them in the range where no problems occur.

Also, because the color of the polygon is set using the FC register, lighting cannot be applied.

6.5.4 Lighting

In 3D drawing, shadow is represented to make objects 3-dimensional. A lighting is used to represent shadow. To represent shadow, set a light source and calculate the front face color based on the positional relationship between the object and the light source.

Setting of light source

There are two types of light sources: point light source and global ambient light source. The point light source has three elements: Ambient light (Ambient), diffuse light (Diffuse), and position (Position). Ambient light is the light component that reaches objects evenly, irrespective of the position of the point light source. Diffuse light is the light component that has direction and changes, depending on the positional relationship between the point light source and the object. The global ambient light source has only ambient light reaching all objects evenly.

Eight point light sources can be set. The position vector of the point light source must be specified using a normalized vector (vector with length of "1").

Setting	Display list	Command	Field
Enable/disable of each point light source	G_LightSetting	-	LE0 to LE7
Each point light source	G_Light	-	ID and others
Global ambient light source	G_GlobalLight	-	-

Setting of material

As the material, set the object reflectivity for the ambient light (Ambient) and diffuse light (Diffuse) and set the emission light (Emission) of the object itself. A different material can be set for the front and back faces.

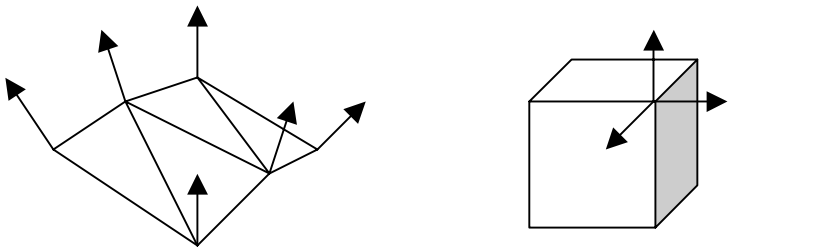
Setting	Display list	Command	Field
Material	G_Material	-	FRT and others

Setting of normal vector

To perform lighting, information on the direction in which each element is facing is required. It is normal vector that has the directional information. Normal vector is set using three elements (Nx, Ny, Nz) for each vertex.

In a smooth face, one vertex corresponds to one normal vector, in angular face, even for the same vertex, a different normal vector corresponds to the vertex for each face. That is, for an angular face, a different normal vector must be input to the same vertex for the number of faces.

To draw a smooth face, enable Gouraud shading (see Section ***6.5.6***).



Normal vector for a smooth face　　Normal vector for an angular face

Fig. 6.6 Normal Vector

When performing MVP transformation for vertex, coordinate transformation must also be performed for normal vector. To perform this transformation, set the inverse matrix of MV matrix using G_LoadMatrixIMV. MV matrix is usually the upper left "3 × 3" part of MVP matrix, and thus set inverse matrix of this part.

Normal vector is specified using a normalized vector (vector with length of "1"), but when an unnormalized vector must be specified, normalization can be performed by hardware. Also, when the user wants to use unnormalized vectors intentionally, disable normalization and then use the normal vector scaling function.

Table 6.5.21 Register to Set Normal Vectors

Setting	Register	Field
Enable/disable of vertex normal vector element	IVAOGL	N

Table 6.5.22 Display Lists to Set Normal Vectors

Setting	Display list	Command	Field
Normal vector transformation matrix	G_LoadMatrixIMV	ColRow/RowCol	-
Enable/disable of normal vector matrix operation	G_MatrixSetting	-	IMV
Enable/disable of normal vector normalization			NVN
Enable/disable of normal vector scaling			NVS
Normal vector scaling ratio	G_NormalScale	-	-

Setting of lighting

To perform lighting, vertex color is calculated based on the relationship between light source and normal vector. When both sides lighting is selected, lighting is performed for both front and rear sides.

When addition of the initial value at lighting calculation is enabled (G_LightSetting: INIC), the color component of the vertex is added as the initial value of lighting operation. The calculation expression of lighting is as follows. Σ term indicates that the color component for the number of enabled point light sources is added.

Vertex color component (R, G, B) = *InitCol + M_Emis + LG_Amb * M_Amb*
*Σ(M_Amb * L_Amb + max(l・n, 0) * M_Diff * L_Diff)+*

Vertex α component (A) = *M_Diff*

InitCol	Initial color (vertex color component)
M_Emis	Emitted light component of material
M_Amb	Ambient light component of material
M_Diff	Diffuse light component of material
LG_Amb	Global ambient light
L_Amb	Ambient light of each point light source
L_Diff	Diffuse light of each point light source
l	Position vector of each point light source
n	Vertex normal vector

Use InitCol when processing light source not supported by hardware, such as specular. InitCol calculates lighting to be added by the host CPU in advance and passes it to hardware as vertex color. When not using InitCol, vertex color is not needed. In such a case, after disabling the color elements (FC, FA, BC, and BA) of the vertex using the IVAOGL register, enable the lighting-related color elements (FC, FA, BC, and BA) using G_VertexSetting. Always issue G_VertexSetting after setting the IVAOGL register because setting the IVAOGL register also changes the setting of the color element. At this time, always disable the addition of the initial value to lighting calculation using G_LightSetting (INIC).

Table 6.5.23 Display Lists to Set Lighting

Setting	Display list	Command	Field
Enable/disable of lighting	G_LightSetting	-	LEN
Switching between single side lighting and both sides lighting			SIDE
Enable/disable of addition of initial value to lighting calculation			INIC
Enable/disable of color elements related to lighting	G_VertexSetting	-	FC, FA ,BC, BA

6.5.5 Flat shading

Flat shading is a function to draw a graphic using a single color. Gouraud shading described later can be selected to draw straight lines, triangles, and quadrangles, but select only flat shading for points and polygons. Disable Gouraud shading to select the flat shading.

Table 6.5.24 Color Setting for Flat Shading

Graphics (nclip included)	Vertex element (FC,FA / BC,BA)	Referenced color
Point (Points)	Enable	The color of each vertex *The contents of the FC register are changed.
	Disable	The FC register
Straight line or triangle (Lines, LineStrip, Triangles, Triangle_Fan, Triangle_Strip)	Enable	The color of each graphic's vertex that is finally input
	Disable	The FC register
Quadrangle (Quads, Quads_Strip)	Enable	The referenced color is different in triangular areas created by dividing a quadrangle by a diagonal. For one triangular area, the color of the third vertex is referenced. For the other triangular area, the color of the fourth vertex is referenced.
	Disable	The FC register
Polygon (Polygon)	Enable	The color of one of the vertices. To obtain the correct drawing result, set the same color for all vertices.
	Disable	The FC register

Table 6.5.25 Registers to Set Flat Shading

Setting	Register	Field
RGB component flat shading for points and straight lines	MDR1	SM
A component flat shading for points and straight lines		AS
RGB component flat shading for triangles	MDR2	SM
A component flat shading for triangles		AS

6.5.6 Gouraud shading

Gouraud shading is a function to smoothly change the graphics color by using straight-line interpolation, based on the color specified for each vertex. Use Gouraud shading when drawing smooth faces.

To add color information to each vertex, enable color components of vertex element (FC, FA, BC, and BA) using GMDR0/IVAOGL or G_VertexSetting.

When Gouraud shading is specified in 8-bit color index mode, Gouraud shading is performed using color index value as 8-bit color component. This function is useful to create alpha map.

Gouraud shading cannot be applied to polygons.

Table 6.5.26 Registers to Set Gouraud Shading

Setting	Register	Field
Enable/disable of RGB component Gouraud shading for point or straight line	MDR1	SM
Enable/disable of A component Gouraud shading for point or straight line		AS
Enable/disable of RGB component Gouraud shading for triangle	MDR2	SM
Enable/disable of A component Gouraud shading for triangle		AS

6.5.7 Alpha blending

Alpha blending is a function to perform semi-transparent drawing. It blends the pixels to be drawn and the pixels already written to the frame buffer at the specified alpha blend ratio. Operation processing of the alpha blending depends on whether or not to enable the function of the blend function (MDR5). When the function is enabled, the operation processing is performed based on the blend function. When disabled, drawing color "C" is calculated as follows (in this case, color of drawn pixel is C_P, frame buffer color is C_F, and alpha value is "A").

$$C = C_P \times A + (1 - A) \times C_F$$

Alpha value "A" is expressed using 8 bits; 00_H indicates blend ratio 0% and FF_H indicates blend ratio 100%.

There are two types of alpha blending: one is blending the entire graphics using a uniform blend ratio, and the other is setting a blend ratio for each vertex to change the transparency ratio. When "A" component is enabled in vertex element, vertex "A" component is used as blend ratio. At this time, when Gouraud shading for "A" component is set to "Enable", the blend ratio can be changed. When the "A" component is disabled, the ALF register is used as blend ratio.

Bit 0 in 16-bit direct color and bits 0 to 7 in 32-bit direct color, graphics is drawn as follows. For 32-bit direct color, eventually the alpha value used for alpha blend is written. (The alpha value includes the following: ALF in normal mode; alpha value of each pixel in alpha Gouraud mode, etc.)

Table 6.5.27 Content of Drawing A Component for Alpha Blending

Color mode	Blend function	Content of A component
16-bit color	None/provided	0: when alpha blend ratio of each pixel is "0" 1: when alpha blend ratio of each pixel is other than "0"
32-bit color	None	Alpha blend ratio of each pixel
	Provided	"A" component after execution of blend function

Table 6.5.28 Registers to Set Alpha Blending

Setting	Register	Field
Enable/disable of alpha blending for point or straight line	MDR1	BM
Enable/disable of alpha blending for triangle or polygon	MDR2	BM
Specification of blend ratio when blend ratio is fixed	ALF	-

Alpha test

Alpha test is a function to compare the reference alpha value set to the ATR register and the alpha value and then, based on the comparison result, to select whether or not to draw pixels. The alpha test operates even when alpha blending is disabled.

Alpha value to be compared is used as alpha value in drawing mode at that time. In normal alpha blending, the alpha value set to the ALF register is used; when Gouraud shading is performed for the alpha value, shaded alpha value is used. Perform the alpha test before performing the blend function processing described later.

Table 6.5.29 Alpha Test Comparison Functions

Comparison function	Description
NEVER	Always does not draw.
ALWAYS	Always draws.
LESS	Draws when “alpha value < reference alpha value”.
LEQUAL	Draws when “alpha value ≤ reference alpha value”.
EQUAL	Draws when “alpha value = reference alpha value”.
GEQUAL	Draws when “alpha value ≥ reference alpha value”.
GREATER	Draws when “alpha value > reference alpha value”.
NOTEQUAL	Draws when “alpha value ≠ reference alpha value”.

Table 6.5.30 Registers to Set Alpha Test

Setting	Register	Field
Enable/disable of alpha test	MDR5	ATE
Alpha test comparison function		ATFUNC
Alpha test reference value	ATR	-

Blend function

Using the function of blend function allows the module to select how to calculate the alpha blend ratio independently for the source and destination. It works only when alpha blending is enabled. Functions using the alpha value of destination pixel can be used only in 32-bit/pixel mode.

Table 6.5.31 Blend Functions

Blend function	Blend ratio
ZERO	(0%, 0%, 0%, 0%)
ONE	(100%, 100%, 100%, 100%)
DST_COLOR	(R_d, G_d, B_d, A_d)
SRC_COLOR	(R_s, G_s, B_s, A_s)
ONE_MINUS_DST_COLOR	(1-R_d, 1-G_d, 1-B_d, 1-A_d)
ONE_MINUS_SRC_COLOR	(1-R_s, 1-G_s, 1-B_s, 1-A_s)
SRC_ALPHA	(A_s, A_s, A_s, A_s)
ONE_MINUS_SRC_ALPHA	(1-A_s, 1-A_s, 1-A_s, 1-A_s)
DST_ALPHA	(A_d, A_d, A_d, A_d)
ONE_MINUS_DST_ALPHA	(1-A_d, 1-A_d, 1-A_d, 1-A_d)
SRC_ALPHA_SATURATE	(f, f, f, 1); f=min(A_s, 1-A_d)

Notes: Parenthesized value refers to the blend ratio of each element of R, G, B, and A.

When the mixture result exceeds the maximum value of each element of R, G, B, and A, each element is clamped to the maximum value.

Table 6.5.32 Register to Set Blend Function

Setting	Register	Field
Enable/disable of blend function processing	MDR5	BFE
Blend function for the source		BLFUNCSRC
Blend function for the destination		BLFUNCDST

6.5.8 Logical operation drawing

Logical operation is performed between the pixels to be drawn and the pixels already written to the frame buffer.

Table 6.5.33 Logical Operation Functions and Operation

Operation format	LOG	Operation	Operation format	LOG	Operation
CLEAR	0000b	0	AND	0001b	S & D
COPY	0011b	S	OR	0111b	S | D
NOP	0101b	D	NAND	1110b	!(S & D)
SET	1111b	1	NOR	1000b	!(S | D)
COPY INVERTED	1100b	!S	XOR	0110b	S xor D
INVERT	1010b	!D	EQUIV	1001b	!(S xor D)
AND REVERSE	0010b	S & !D	AND INVERTED	0100b	!S & D
OR REVERSE	1011b	S | !D	OR INVERTED	1101b	!S | D

Table 6.5.34 Registers to Set Logical Operation Drawing

Setting	Register	Field
Enable/disable of logical operation drawing for point and straight line	MDR1	BM
Specification of logical operation function for point and straight line		LOG
Enable/disable of logical operation drawing for triangle and polygon	MDR2	BM
Specification of logical operation function for triangle and polygon		LOG

6.5.9 Depth test

In 3D drawing, depth test is used when removing hidden surface using the Z buffer method. To compare depth, always set the Z buffer.

Setting of Z buffer

Z buffer is of a shape of which vertical and horizontal pixel counts are the same as those of the drawing frame. Z value can be selected among 32 bits/pixel, 16 bits/pixel and 8 bits/pixel. Always clear the Z value before drawing each frame using DrawRectP. At this time, be careful about color order (MDR0:CO).

The Z value of the Z buffer is a positive integer. It is clamped to "0" when the Z value becomes a negative value due to MVP transformation and view port transformation.

Table 6.5.35 Registers to Set Z Buffer

Setting	Register	Field
Base address of Z buffer	ZBR	-
Z value mode	MDR0	ZP

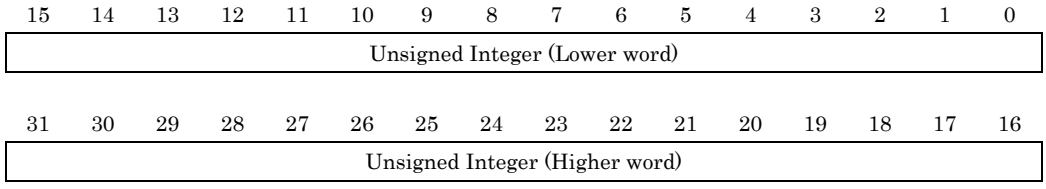
Table 6.5.36 Z Value Byte Count/Pixel

Z value mode	Byte count per pixel
8 bits	1
16 bits	2
32 bits	4

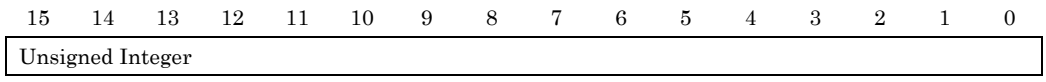
[Memory data format]

The Z value can be used as 32 bits, 16 bits, or 8 bits per pixel.

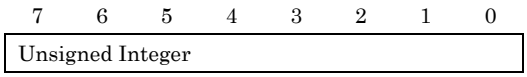
(1) Unsigned 32-bit integer data



(2) Unsigned 16-bit integer data



(3) Unsigned 8-bit integer data



Setting of depth test

Depth test compares "Z" value of pixel and "Z" value of Z buffer to control whether to draw (PASS) or not to draw (FAIL). Using this function allows the module to remove the hidden surface, which hides an inner object by an outer object.

How to compare is set using the MDR1 register (point and straight line) or the MDR2 register (triangle and polygon). Even in the case of PASS, the depth value write mask function is available that does not update the Z buffer.

When Z coordinate of each vertex is different, Z value has possibility to have difference from ideal value. Therefore, EQUAL, LEQUAL and GEQUAL that are including equal judgement has possibility that it doesn't judge exact equality when Z coordinate has a difference.

For the cautions in applying the depth test to polygons, see ***Input of drawing graphics***.

Table 6.5.37 Registers to Set Depth Test

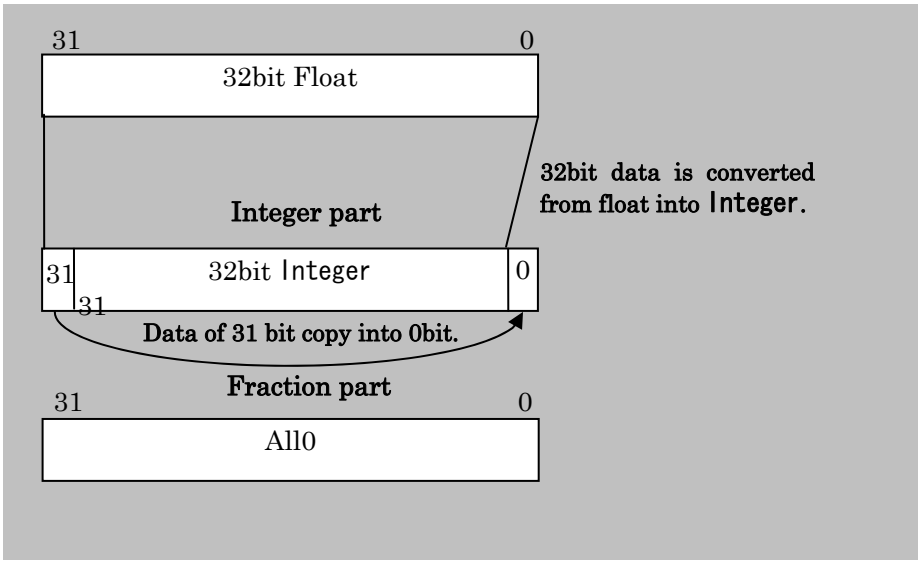
Setting	Register	Field
Enable/disable of depth test for point and straight line	MDR1	ZC
Depth comparison function for point and straight line		ZCL
Depth value write mask for point and straight line		ZW
Enable/disable of depth test for triangle and polygon	MDR2	ZC
Depth test function for triangle and polygon		ZCL
Depth value write mask for triangle and polygon		ZW

Table 6.5.38 Depth Test Functions using MDR1/MDR2 Register

Depth test function	Code	Condition
NEVER	000_B	Always does not draw.
ALWAYS	001_B	Always draws.
LESS	010_B	Draws when "pixel Z value < Z buffer value".
LEQUAL	011_B	Draws when "pixel Z value ≤ Z buffer value".
EQUAL	100_B	Draws when "pixel Z value = Z buffer value".
GEQUAL	101_B	Draws when "pixel Z value ≥ Z buffer value".
GREATER	110_B	Draws when "pixel Z value > Z buffer value".
NOTEQUAL	111_B	Draws when "pixel Z value != Z buffer value".

Table 6.5.39 Depth Write Mask using MDR1/MDR2 Register

Depth write mask	1	Does not write "Z" value.
	0	Writes "Z" value (when Z comparison mode is ON).



When Z value mode is 32bit/pixel, Conversion from Float of Z value to Integer.（Integer part）

When you draw the figure except polygon(Polygon,nclip_Polygon) by using geometry at Z value mode of 32bit/pixel, 31bit and 0bit are set the same value like Figure above .

Therefore, When Z value is 0x80000000(= 2147483648) or more, 0bit is set to 0x1.

When Z value is 0x7FFFFFFF(= 2147483647) or less, 0bit is set to 0x0.

In that case, "Fraction part" is set to 0x00000000.

6.5.10 Texture Mapping

Texture mapping is a function to read the texel corresponding to the texture coordinate specified for vertex and then paste it to graphics. In KOTTOS, two textures can be used. Also, KOTTOS has two texture units to blend texture.

For the cautions in applying the texture to polygons, see ***Input of drawing graphics***.

Texture coordinate

Texture coordinate is a 2D coordinate system where horizontal and vertical positions are represented using ST coordinate. Upper left ST coordinate of the texture is (0.0, 0.0); lower right (1.0, 1.0).

As texture pattern, up to "4096 × 4096" pixels can be used. Pattern size to be used is set using **`RegTexture`**. When the ST coordinate exceeds the pattern range, several processing methods such as the processing to repeat the texture pattern (Repeat) and the processing to extend texel at the edge (Clamp) can be selected.

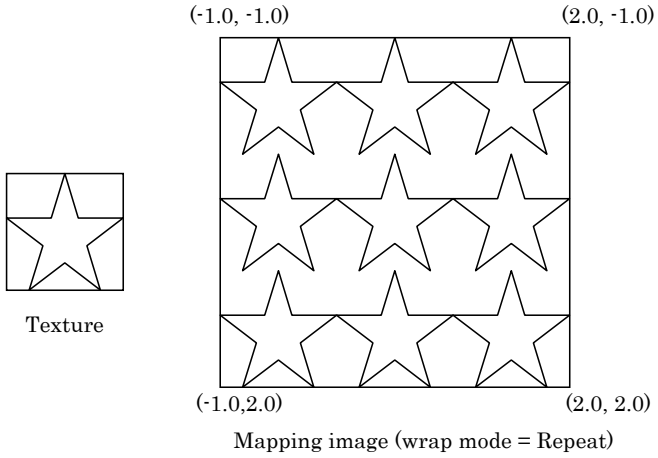
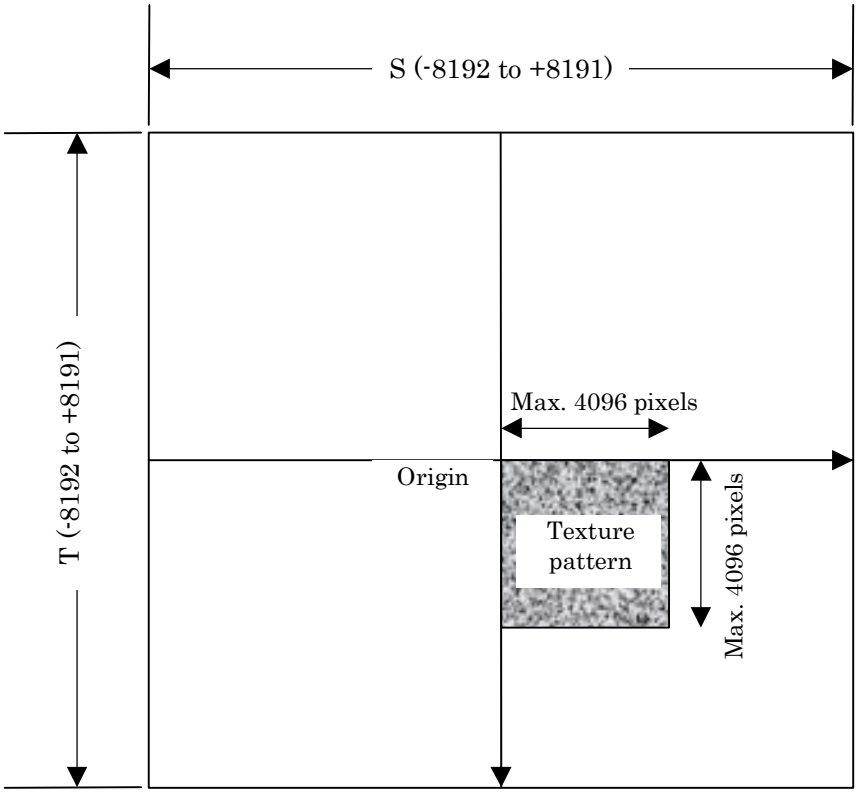


Fig. 6.7 Texture Coordinate and Mapping Image

The specifiable range of texture coordinate varies with the texture size. The texture model coordinate given to vertex is multiplied by the texture size and transformed to texture device coordinate. When a texture of "64 × 64" pixel is used, texture model coordinate (1.0) is transformed to texture device coordinate (64.0).

A value from "-8192" to "+8191" can be used for texture device coordinate. Texture device coordinate is added to the vertex of face primitive, thereby associating the face and the texture pattern.



Registration of texture

KOTTOS stores up to eight pieces of texture information. Changing the entry specification ID when drawing graphics allows the KOTTOS to use multiple textures. To use eight textures or more, rewrite the texture information table.

Texture information consists of base information, which is the information on the texture itself, and state information, which specifies how to map.

[Base information]

Storage address

Size

Bit per pixel (BPP)

Specification of compressed/uncompressed format

Specification of bilinear fast mode format

Format

Table 6.5.40 Display List to Register Texture

Input content	Display list	Command
Registration of texture information	RegTexture	Base
Registration of texture mapping method		State

Texture size

Regarding each of S and T, selectable texture data size is a value from 1 to 4096 pixels that is expressed as power of "2".

Table 6.5.41 Field to Register Texture Size

Input content	Display list	Command	Field
Registration of texture size	RegTexture	Base	SizeS, SizeT

Texture format

The formats shown in

Table 6.5.43 can be used as texture format. "Base Format" in the left column is a texture format specified using the RegTexture command. "Derived Source Color" in the right column represents how to transform each format to RGBA during pixel processing. "Texture Bit Per Pixel" in the middle column specifies the texel bit length and the bit arrangement.

Textures in uncompressed format and palette format are interpreted as ARGB or RGBA depending on color order setting (CO of MDR0), and, as with pixel, is transformed to RGBA format when drawing. Texture in the compressed format supports only RGBA format. Therefore, the color order must be set to RGBA format. Convert the texture in the compressed format in advance to set to RGBA format at compression.

Table 6.5.42 Field to Register Texture Format and BPP

Input content	Display list	Command	Field
Registration of texture format	RegTexture	Base	FMT
Registration of texture BPP			BPP

Table 6.5.43 RegTexture Format and Settable BPP, and Assignment when Blending

Base format (FMT)	Texture Bit Per Pixel (BPP)	Derived Source Color (R, G, B, A)
ALPHA	8BPP	(0, 0, 0, A)
LUMINANCE	8BPP	(L, L, L, 1)
LUMINANCE_ALPHA	16BPP	(L, L, L, A)
INTENSITY	8BPP	(I, I, I, I)
RGB	8BPP/RGB5_A1/RGBA8/R5_G6_B5/RGBA4	(R, G, B, 1)
RGBA	8BPP/RGB5_A1/RGBA8/R5_G6_B5/RGBA4	(R, G, B, A)

Note: When the frame buffer is other than 8 bits/pixel, the texture with "BPP = 8BPP" cannot be used with "FMT = RGB" and "FMT = RGBA".

Texture BPP

Table 6.5.44 shows the bit formats supported as texture format. For the umcompressed format and palette format, whether to place "A" component on the lower bit or on the upper bit can be selected by setting the ColorOrder bit of the MDR0 register. The compressed format supports only RGBA format.

Table 6.5.44 Bit Format of Texture

	RegTexture(Base)		Texture Bit Per Pixel	Color Order = 1	Color Order = 0
	CMP	BPP			
Uncompressed format	PLAIN	8BPP	8 bits	Color data 8 bits	Same as left
		RGB5_A1	16 bits	R5:G5:B5:A1	A1:R5:G5:B5
		RGBA8	32 bits	R8:G8:B8:A8	A8:R8:G8:B8
		R5_G6_B5	16 bits	R5:G6:B5	Same as left
		RGBA4	16 bits	R4:G4:B4:A4	A4:R4:G4:B4
		16BPP	16 bits	16bit code	Same as left
Palette format	PALETTE4	RGB5_A1	4 bits + Palette table	R5:G5:B5:A1	A1:R5:G5:B5
		RGBA8		R8:G8:B8:A8	A8:R8:G8:B8
		R5_G6_B5		R5:G6:B5	Same as left
		RGBA4		R4:G4:B4:A4	A4:R4:G4:B4
		16BPP		16bit code	Same as left
	PALETTE8	RGB5_A1	8 bits + Palette table	R5:G5:B5:A1	A1:R5:G5:B5
		RGBA8		R8:G8:B8:A8	A8:R8:G8:B8
		R5_G6_B5		R5:G6:B5	Same as left
		RGBA4		R4:G4:B4:A4	A4:R4:G4:B4
		16BPP		16bit code	Same as left
Compressed format	COMPRESSED	8BPP	8 bits	Color data 8bit	–
		RGB5_A1	16 bits	R5:G5:B5:A1	–
		RGBA8	32 bits	R8:G8:B8:A8	–
		16BPP	16 bits	16-bit code	–

Note: Palette format above refers to the color code and format of palette table.

The memory allocation of each format is shown. KOTTOS supports little endian and pixels are allocated starting at the LSB in the 64-bit unit of the bus width.

Fig. 6.8 shows an example of the memory storage format in uncompressed format (PLAIN, RGBA8). "P" means a pixel. For mipmap format, pixels are stored starting at the highest resolution level 0 for the size up to 1 × 1.

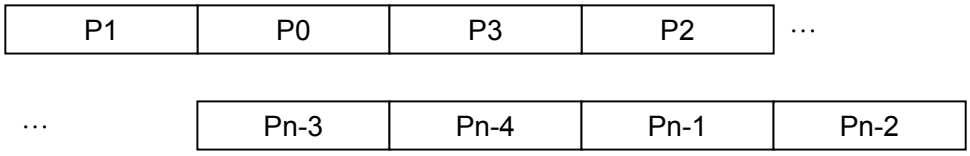


Fig. 6.8 Memory Storage Format of Uncompressed Format (PLAIN, RGBA8)

Fig. 6.9 shows an example of memory storage format of palette format (PALETTE8, RGBA8). "P" means a palette code making up texture/tile pattern, and "Color" means a color code table corresponding to P0 to P255. For mipmap format, the color code table is common to each level. The palette code is stored starting at the highest resolution level 0 for the size up to 1 × 1.

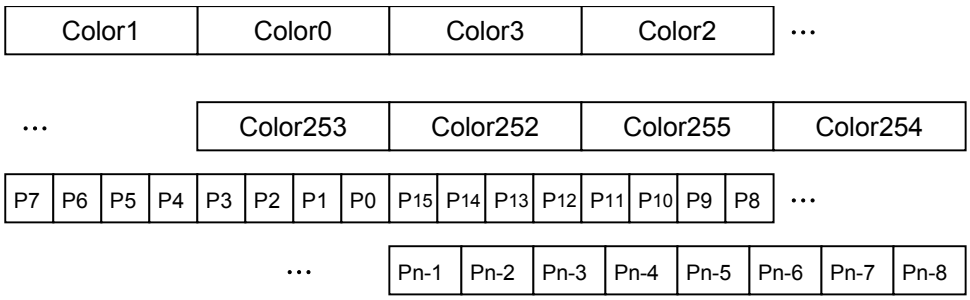


Fig. 6.9 Memory Storage Format of Palette Format (PALETTE8, RGBA8)

When the texture in the compressed format is mipmap format, one data item containing each level is created by performing compression and conversion. Consequently, there is no need to store each mipmap level. ***Fig. 6.10*** shows an example of the memory storage format. "D" means 1-byte data.

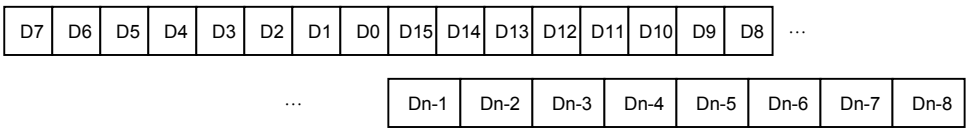
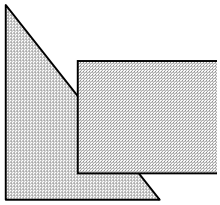


Fig. 6.10 Memory Storage Format of Compressed Format (COMPRESSED)

Tiling

Drawing is performed using the pixel read from the tiling pattern corresponding to the specified coordinate. The tiling determines pixel on the pattern to be read using the coordinate of the pixel to be drawn, irrespective of position and size of the primitive.



Example of tiling

Table 6.5.45 Display List to Specify Tiling

Input content	Display list	Command	Field
Specification of tiling	RegTexture	Base	TL

Format for bilinear fast mode

This format speeds up bilinear filtering by creating texture data in advance in an arrangement where 4-pixel redundancy is provided per pixel in a normal texture arrangement.

This processing requires information for 4 pixels per pixel, therefore it uses an area 4 times greater than the normal one. Data in this format can be used only when performing bilinear filtering, and cannot be used when performing point sampling.

Color mode is limited to 16-bit direct color.

	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	08	09	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

Normal texture arrangement (“8 × 8” pixels)

	0				1					6				7			
0	00	01	08	09	01	02	09	10	～	06	07	14	15	07	00	15	08
1	08	09	16	17	09	10	17	18	～	14	15	12	13	15	08	23	16
2	16	17	24	25	17	18	25	26	～	22	23	30	31	23	16	31	24
3	24	25	32	33	25	26	33	34	～	30	31	38	39	31	24	39	32
4	32	33	40	41	33	34	41	42	～	38	39	46	47	39	32	47	40
5	40	41	48	49	41	42	49	50	～	46	47	54	55	47	40	55	48
6	48	49	56	57	49	50	57	58	～	54	55	62	63	55	48	63	56
7	56	57	00	01	57	58	01	02	～	62	63	06	07	63	56	07	00

Texture arrangement in bilinear fast mode (“8 × 8” pixels)

Table 6.5.46 Display List to Specify Bilinear Fast Mode

Input content	Display list	Command	Field
Specification of bilinear fast mode	RegTexture	Base	BA

Texture wrapping

This processing specifies operation when a negative value or a value greater than the texture size is specified for value of S and T.

Function	Description
Wrap	CLAMP_TO_EDGE REPEAT BORDER CLAMP MIRRORED_REPEAT

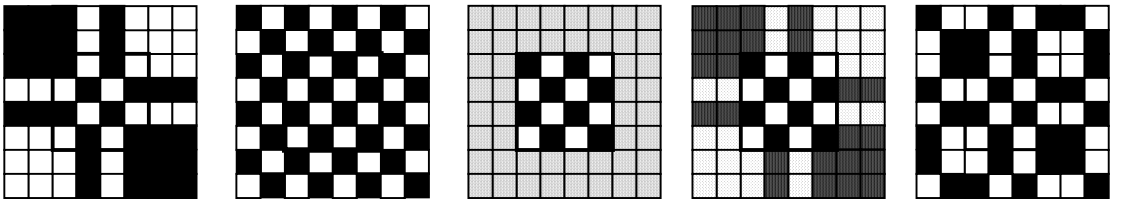


Fig. 6.11 Image of Wrapping Mode

[CLAMP_TO_EDGE]

When the given S, T coordinate is a negative value or a value greater than the texture size, the S, T coordinate is fixed as follows.

S < 0	S = 0
S > Texture X size – 1	S = Texture X size - 1

[REPEAT]

It simply masks the upper x bit of the given S, T coordinate. When the texture size is 64 pixels, lower 6 bits of the integer part of ST is used for the S, T coordinate.

[BORDER]

When the given ST coordinate is a negative value or greater than the texture size, it draws using border color instead of texture.

[CLAMP]

It is the same operation as CLAMP_TO_EDGE when filtering mode is NEAREST. When performing bilinear filtering (LINEAR, LINEAR_MIPMAP_NEAREST, LINEAR_MIPMAP_LINEAR), CLAMP is different from CLAMP_TO_EDGE in that CLAMP blends the range where the ST coordinate is greater than the texture size with the border color.

[MIRRORED_REPEAT]

It maps the range where the ST coordinate is greater than the texture size while inversing the S, T coordinate.

Table 6.5.47 Display List to Specify Wrapping Mode

Input content	Display list	Command	Field
Specification of wrapping mode	RegTexture	State	WRAPS, WRAPT

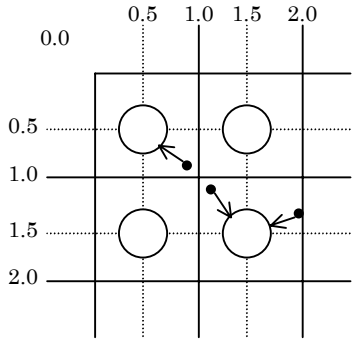
Texture filtering

When mapping a texture to a graphics smaller than the size of the original texture or greater than the size of the original texture, various filtering modes can be specified to improve the quality of the mapping result. Processing time increases as the quality of mode is higher.

The user can select how to map when mapping a texture to a graphics greater than the size of the original texture (Magnification) or smaller than the size of the original texture (Minification).

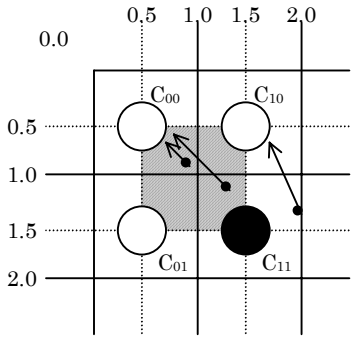
[Point sampling]

This is the simplest mode that uses the texture pixel (texel) specified by (S, T) for drawing as it is. It selects the pixel nearest the calculated ST coordinate.



[Bilinear filtering]

This mode blend texture pixel at four points near the texture pixel specified fby (S, T) according to the distance from the specified point and uses the blend result for drawing.



[Mipmap]

When mapping texture, this mode maps texture at the mipmap level corresponding to the reduction ratio. Texture of reduced version must be prepared. The texture whose ST size is halved is reduced to become a texture whose ST size is "1 × 1".

Perspective correction described later must be enabled when using mipmap.

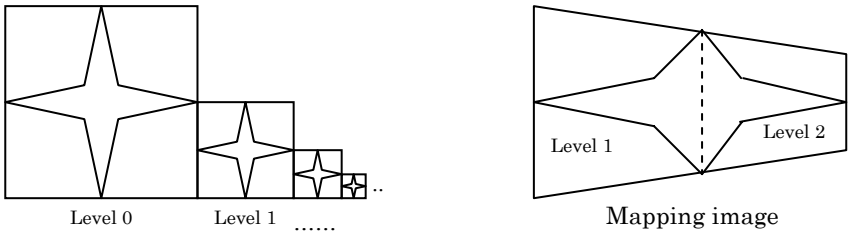


Fig. 6.12 Mipmap Texture

[Trilinear filtering]

This mode samples texels from two mipmap levels to blend them according to the distance from the two mipmap levels. The pixel at the center of the mapping image shown in ***Fig. 6.12*** is situated right in the middle of level 1 and level 2, meaning the two texels are blended at 50%.

Table 6.5.48 Setting of Texture Filtering

Input content	Display list	Command	Field
Specification of filtering mode (magnification)	RegTexture	State	MAGFL
Specification of filtering mode (minification)			MINFL

Table 6.5.49 Relationship between Filtering and MINFL Setting

MINFL	Bilinear	Mipmap	Trilinear
NEAREST	×	×	×
LINEAR	○	×	×
NEAREST_MIPMAP_NEAREST	×	○	×
NEAREST_MIPMAP_LINEAR	×	○	○
LINEAR_MIPMAP_NEAREST	○	○	×
LINEAR_MIPMAP_LINEAR	○	○	○

○: Enable ×: Disable

Perspective correction

When simply mapping texture to the graphics for which perspective transformation has been performed, it is mapped distorted. Perspective correction eliminates this distortion.

Perspective correction must be set at two timings: when registering texture and when using texture. When registering texture, use RegTexture to specify whether or not to perform perspective correction for each texture. And, to specify texture for which perspective correction is enabled using BindTexture, enable RPC using the IVAOGL register (OpenGL mode) or G_VertexSetting (Coral mode).

Perspective correction does not function correctly when both the PC bit of RegTexture and the RPC bit of the IVAOGL/G_VertexSetting are not enabled.

Table 6.5.50 Display Lists to Set Perspective Correction

Input content	Display list	Command	Field
Enable/disable of perspective correction (at registration time)	RegTexture	State	PC
Enable/disable of perspective correction for Coral mode	G_VertexSetting	-	RPC

Table 6.5.51 Register to Set Perspective Correction

Setting	Register	Field
Enable/disable of perspective correction for OpenGL mode	IVAOGL	RPC

Use of texture mapping

KOTTOS maps two textures, texture 0 and texture 1 to graphics. At this time, the user can select an arbitrary texture from the texture information table, as texture 0 and texture 1 respectively.

KOTTOS has two blend units; texture unit 0 and 1, to blend the mapped two textures. Either texture unit can select texture 0 or texture 1 as source.

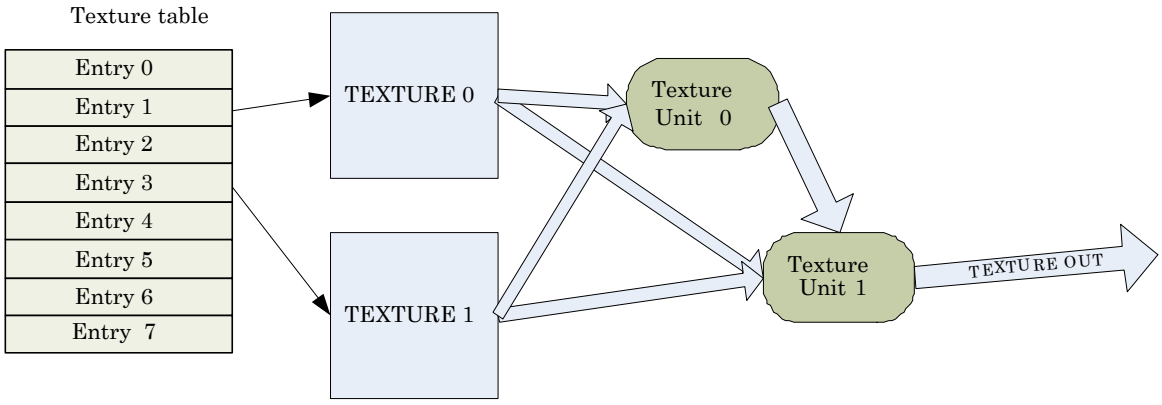


Fig. 6.13 Relationship among Texture Information Table, Textures and Texture Units

Table 6.5.52 Display List Used to Assign Textures

Input content	Display list	Command	Field
Assignment of table entry to texture 0 and 1	BindTexture	-	

Table 6.5.53 Registers to Set Texture Assignment

Setting	Register	Field
Addition of texture coordinate to vertex (OpenGL mode)	IVAOGL	ST0,Q0,ST1,Q1
Addition of texture coordinate to vertex (Coral mode)	GMDR0	ST
Enable/disable of texture unit 0 for point and straight line	MDR1	TU0
Enable/disable of texture unit 1 for point and straight line		TU1
Enable/disable of texture unit 0 for triangle and polygon	MDR2	TU0
Enable/disable of texture unit 1 for triangle and polygon		TU1

Texture blending

Texture unit performs blend operation processing for three terms: Arg0, Arg1 and Arg2.

Set the blend function for each of R, G, and B components and for "A" component.

Table 6.5.54 Operation of Texture Blending and Whether to Enable/Disable Texture Blending

COMBINE FUNCTION	Expression	For RGB factor	For A factor
REPLACE	Arg0	○	○
MODULATE	Arg0 *Arg1	○	○
ADD	Arg0 +Arg1	○	○
ADD_SIGNED	Arg0 +Arg1 -0.5	○	○
INTERPOLATE	Arg0 *Arg2 + Arg1 *(1-Arg2)	○	○
SUBTRACT	Arg0 -Arg1	○	○
DOT3_RGB	4 * ((Arg0r-0.5)*(Arg1r-0.5)+	○	×
DOT3_RGBA	(Arg0g-0.5)*(Arg1g-0.5)+ (Arg0b-0.5)*(Arg1b-0.5))	○	×

Set what is used as Arg0 to Arg2 as follows.
As with blend function, it can be set separately for R, G, and B component and for "A" component.

Table 6.5.55 Source Selection for Texture Blending

SOURCEi (i=0 to 2 e.g. Arg0 to 2)	Source type	Texture Unit	
SOURCEi_RGB / SOURCEi_ALPHA	CONSTANT	0/1	BLDCONST
	PRIMARY_COLOR	0/1	Fragment color before texturing
	PREVIOUS	0	PRIMARY_COLOR
		1	TU0's output
	TEXTURE0	0/1	TEXTURE0 *
	TEXTURE1	0/1	TEXTURE1 *

In addition, operation processing can be set before assigning the source (Arg) to each component.
Operation processing can also be set separately for R, G, and B component and for "A" component.

* There are the following restrictions:

- When specifying TEXTURE0 as the source, enable TU0.
- When specifying TEXTURE1 as the source, enable TU1.

Table 6.5.56 Operation for Source during Texture Blending

OPERANDi (i=0 to 2 e.g. Arg0 to 2)	Operand	Expression
OPERANDi_RGB	SRC_COLOR	(R, G, B)
	ONE_MINUS_SRC_COLOR	(1-R, 1-G, 1-B)
	SRC_ALPHA	(A, A, A)
	ONE_MINUS_SRC_ALPHA	(1-A, 1-A, 1-A)
OPERANDi_A	SRC_ALPHA	(A)
	ONE_MINUS_SRC_ALPHA	(1-A)

Table 6.5.57 Registers to Set Texture Unit Operation Processing

Setting	Register	Field
Source selection for texture unit 0	BLDTU00	SRC0RGB, SRC0A, SRC1RGB, SRC1A, SRC2RGB, SRC2A
Specification of blend operation for texture unit 0		FUNCRGB, FUNCA
Specification of source operation for texture unit 0	BLDTU01	OP0RGB, OP0A, OP1RGB, OP1A, OP2RGB, OP2A
Source selection for texture unit 1	BLDTU10	SRC0RGB, SRC0A, SRC1RGB, RC1A, SRC2RGB, SRC2A
Specification of blend operation for texture unit 1		FUNCRGB, FUNCA
Specification of source operation for texture unit 1	BLDTU11	OP0RGB, OP0A, OP1RGB, OP1A, OP2RGB, OP2A

6.5.11 Fog

Fog is a function to haze over the object as it goes away further. When fogging, this fog function blends fragment color and fog color using fog factor (f).

$C = fC_r + (1-f)C_f$: (C_r : fragment color, C_f : for color, f : fog factor)

As the fog factor becomes nearer "0", the fog color element becomes stronger and thus the haze effect for remote area becomes stronger.
Fog factor (f) is calculated from the fog factor table. ***Fig. 6.14***. shows an example of fog factor table.

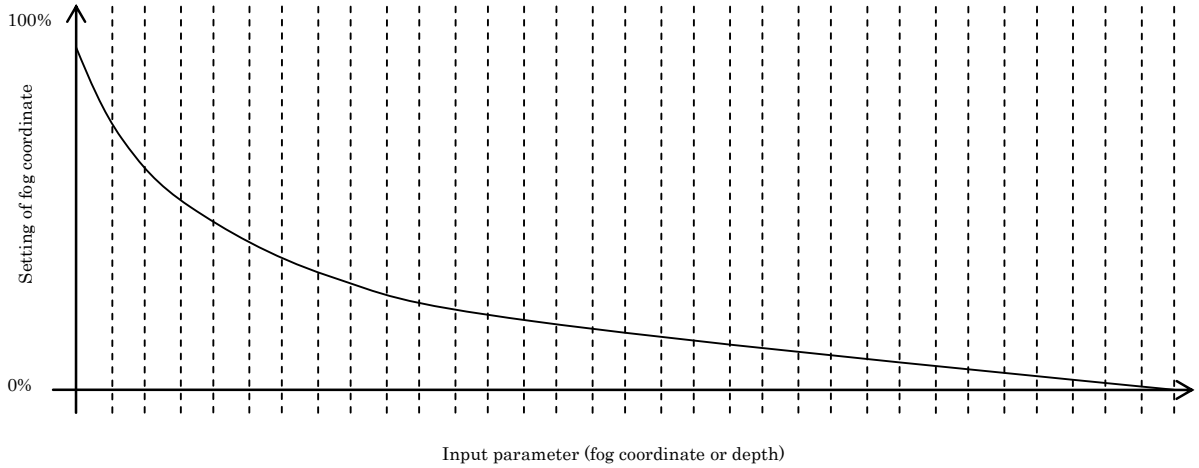


Fig. 6.14 Example of Fog Factor Table

Setting of fog coordinate

Input for when referencing the fog factor table is fog coordinate. Fog factor is usually calculated from the depth (Z coordinate). Fog coordinate can be calculated automatically by setting the transformation matrix for fog coordinate using G_LoadMatrixMV and enabling fog coordinate operation using G_MatrixSetting. At this time, the FO field of the IVAOGL register must be set to "Enable". When fog coordinate operation is enabled, fog coordinate needs not be given as vertex element, and so set the FOG field of the IVAOGL register to "Disable".
The matrix set using G_LoadMatrixMV is the third line (for MVP matrix, the line used to calculate Z coordinate) of MV transformation matrix. When the same value as the third line of MVP transformation matrix is set, fog coordinate becomes the same value as "Z" value. Changing this matrix allows the module to calculate the fog coordinate appropriate to the fog processing.

$$F = m02 \times X + m12 \times Y + m22 \times Z + m32 \times W$$

When the user specifies fog coordinate for each vertex instead of hardware fog coordinate, set fog coordinate calculation to "Disable" using G_MatrixSetting and set the FOG field to "Enable" using the IVAOGL register and then directly specify fog coordinate for vertex. In this case, the FO field must also be set to "Enable".

In both the case where automatic calculation is performed and the case where fog coordinate is specified directly, set FogCoord to the FOGCRD field of the MDR7 register.

For the cautions in applying the fog to polygons, see ***Input of drawing graphics***

Table 6.5.58 Display Lists to Set Fog Coordinate Calculation

Input content	Display list	Command	Field
Setting of transformation matrix for fog coordinate calculation	G_LoadMatrixMV	-	-
Enable/disable of fog coordinate calculation	G_MatrixSetting	-	FOGZ

Table 6.5.59 Registers to Set Fog Coordinate

Setting	Register	Field
Enable/disable of fog coordinate of vertex element	IVAOGL	F
Enable/disable of fog coordinate in operation processing		FO
Selection of horizontal axis input for fog table	MDR7	FOGCRD

When accurate fog processing is not needed, the user can select the "Z" value for depth comparison (Z test) as the horizontal axis input parameter for the fog factor table. In this case, set "Z" to the FOGCRD field of the MDR7 register.

Setting of the fog factor table

Set input parameter and fog factor function to the fog factor table. The vertical axis is fog factor; the horizontal axis is input parameter (fog coordinate or "Z" value).

Divide this fog factor table along the horizontal axis evenly into 32 sections and set initial value and inclination value in each section. This means approximating the fog factor function that is originally a curve by a line graph divided into 32 sections. To set the table, use the SetFog display list.

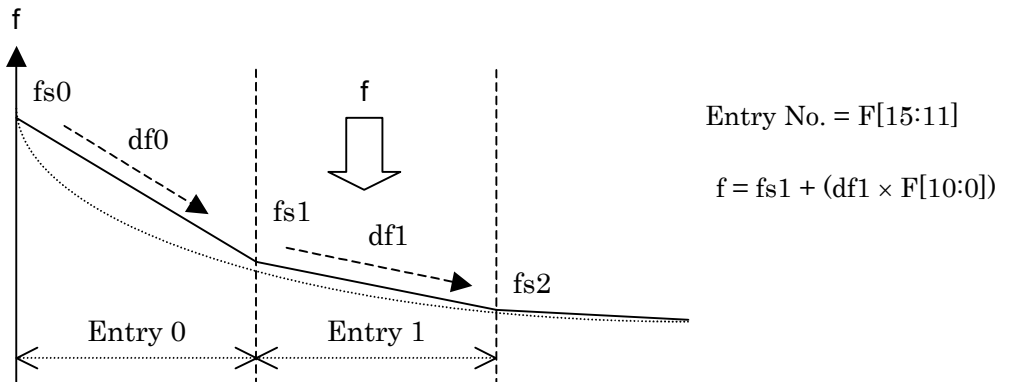


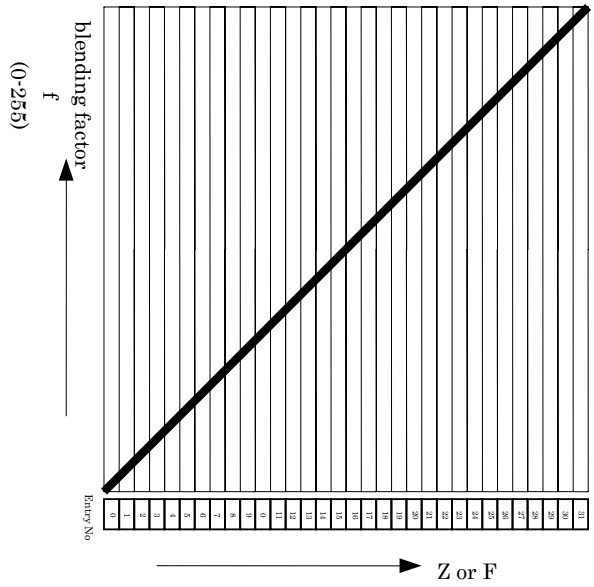
Fig. 6.15 Fog Factor Calculation When Fog Coordinate Used (When Entry 1 Selected)

Table 6.5.60 Selection of Input Parameter for Fog Factor Table

MDR7:FOGCRD	
FogCoord	Set the fog coordinate as an input parameter. The upper 5 bits of fog coordinate are used to select the table section. The lower bits of fog coordinate are used to obtain a fog factor from the initial value and incremental value.
Z	Set the "Z" value as an input parameter. The upper 5 bits of "Z" values are used to select the table section. The lower bits of "Z" value are used to obtain a fog factor from the initial value and incremental value. For 32 bits/Z, the integer part [31:16] of Z is used as the input parameter value; for other Z value modes, the integer part [15:0] of Z is used as the input parameter value. (For 8 bits/Z, only space in Entry0 can be used.)

6.5.12 Example of Fog Factor Table Setting

An example of a linear setting is shown.



The line segment in each entry can be arbitrarily specified for the initial value and inclination. Note that line segments adjacent to each other become discontinuous depending on the line segment specification. The bottom left in each entry is the origin in the coordinate system.

Z or F can be selected as the axis in the horizontal direction when setting. When the axis is F, the integer part [15:0] of F is used. When the axis is Z, if Z is 32 bits (set by ZP bit of MDR0), the integer part [31:16] of Z is used; if Z is not 32 bits, the integer part [15:0] of Z is used. (If Z is 8 bits, only the space in Entry0 can be used.)

An example of the display list is shown below.

1b000020 // setFOG
00000000 //0
00000100
00080000 //1
00000100
00100000 //2
00000100
00180000 //3
00000100
00200000 //4
00000100
00280000 //5
00000100
00300000 //6
00000100
00380000 //7
00000100
00400000 //8
00000100
00480000 //9
00000100
00500000 //10
00000100
00580000 //11
00000100
00600000 //12
00000100
00680000 //13
00000100
00700000 //14
00000100
00780000 //15
00000100
00800000 //16
00000100
00880000 //17
00000100
00900000 //18
00000100
00980000 //19
00000100
00a00000 //20
00000100
00a80000 //21
00000100
00b00000 //22
00000100
00b80000 //23
00000100
00c00000 //24
00000100
00c80000 //25
00000100
00d00000 //26
00000100
00d80000 //27
00000100
00e00000 //28
00000100
00e80000 //29
00000100
00f00000 //30
00000100
00f80000 //31
00000100

For your reference, a schematic diagram of the fog factor table in OpenGL mode is shown in ***Fig. 6.16***.

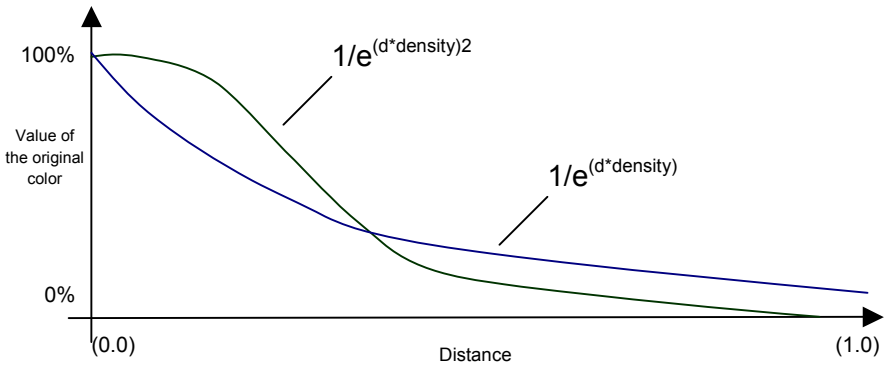


Fig. 6.16 Schematic Diagram of Fog Factor Table in OpenGL Mode

Table 6.5.61 Display List to Set Fog Factor Table

Input content	Display list	Command	Field
Setting of fog factor table	SetFog	Table	-

Use of fog function

After calculating the fog coordinate and setting the fog table, enable the fog function.

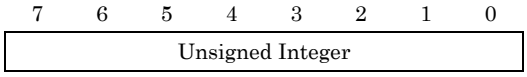
Table 6.5.62 Registers to Set Fog Function

Setting	Register	Field
Enable/disable of fog for point and straight line	MDR1	FOG
Enable/disable of fog for triangle and polygon	MDR2	FOG
Fog color	FOGCOL	-

6.5.13 Stencil Test

Stencil test controls whether or not to draw pixels by comparing the reference stencil value and the stencil buffer value. The user can also specify various operation processing for the stencil buffer depending on the comparison result. The stencil buffer consists of 8 bits per pixel.

Stencil value: 8 bits/pixel unsigned integer



For the stencil buffer, how to update can be specified in three cases respectively: a case where the stencil buffer does not pass the stencil test, a case where the stencil buffer does not pass the depth test (Z test), and a case where the stencil buffer passes the depth test.

Table 6.5.63 Comparison Functions for Stencil Test

Comparison function for stencil test	Code	Condition
NEVER	000	Always does not draw.
ALWAYS	001	Always draws.
LESS	010	Draws when “reference stencil value < stencil buffer value”.
LEQUAL	011	Draws when “reference stencil value <= stencil buffer value”.
EQUAL	100	Draws when “reference stencil value = stencil buffer value”.
GEQUAL	101	Draws when “reference stencil value >= stencil buffer value”.
GREATER	110	Draws when “reference stencil value > stencil buffer value”.
NOTEQUAL	111	Draws when “reference stencil value != stencil buffer value”.

Table 6.5.64 Update Conditions for Stencil Test

Stencil update condition	Description
SFAIL	When stencil test shows “does not draw”: * The stencil test and depth test are performed in this order. If the stencil buffer does not pass the stencil test, the depth test is not performed and the result of the depth test is undefined.
DPFAIL	When depth test shows “does not draw”: * Since the depth test has been performed, it means the stencil buffer have passed the stencil test.
DPPASS	When depth test shows “draws”: * Since the depth test has been performed, it means the stencil buffer have passed the stencil test.

Table 6.5.65 Update Functions for Stencil Test

	Stencil operation function	Code	Condition
SFAIL/ DPFAIL/ DPPASS	KEEP	000	Does not update the stencil buffer.
	ZERO	001	Writes "0" to the stencil buffer.
	REPLACE	010	Writes a reference stencil value to the stencil buffer.
	INCR	011	Increments the stencil buffer value by "1" (with clamping).
	DECR	100	Decrements the stencil buffer value by "1" (with clamping).
	INVERT	101	Performs bit inversion for the stencil buffer value.
	INCR_WRAP	110	Increments the stencil buffer value by "1" (without clamping).
	DECR_WRAP	111	Decrements the stencil buffer value by "1" (without clamping).

Notes:

1. "With clamping" refers to the processing that ignores "-1" for minimum value 00_H and ignores "+1" for maximum value FF_H.
2. "Without clamping" refers to the processing that treats "-1" for minimum value 00_H as FFH and treats "+1" for maximum value FF_H as 00_H.

Table 6.5.66 Registers to Set Stencil Test

Setting	Register	Field
Address of stencil buffer	STCBR	-
Reference value when performing stencil test	STCR	STCREF
Stencil buffer write mask		STCMASK
Enable/disable of stencil test	MDR6	STCE
Stencil test function		STFUNC
Specification of stencil buffer update processing when stencil test FAILs		SFAIL
Specification of stencil buffer update processing when depth test FAILs		DPFAIL
Specification of stencil buffer update processing when depth test PASSes		DPPASS

6.5.14 PolygonMode

The PolygonMode is a function to draw Triangle type graphics or Quad type graphics using only line segments or only points. When PolygonMode = Line, specify the line segment drawn for which vertex. Set "1" to the EF bit of the header (G_Vertex) of the vertex that becomes the start point of the line segment.

[Notes]

- The PolygonMode does not support Point type graphics, Line type graphics, and Polygon type graphics.
- G_VertexIndex can be used only when PolygonMode = Fill.

Table 6.5.67 Display lists to set PolygonMode

Input	Display list	Command	Field
Setting of input data format to OpenGL mode	SetGModeRegistaer	00h	IDFM
Setting of PolygonMode	G_PolygonSetting	-	POMF,POMB
Specification of vertex for which line segment drawn when PolygonMode = Line	G_Vertex	-	EF

6.5.15 PolygonOffset

PolygonOffset is a function to add offset to the Z value after coordinate transformation is performed. The expression to calculate the offset value is shown below.

Offset = m × factor + r × units

The user sets the "factor" and "r × units". "m" is the inclination after coordinate transformation and is calculated automatically by KOTTOS. Set a separate value for the front and back faces when PolygonMode is Point, Line, and Fill, respectively.

Table 6.5.68 Display Lists to Set PolygonOffset

Input	Display list	Command	Field
Setting of input data format to OpenGL mode	SetGModeRegistaer	00h	IDFM
Enable/disable of PolygonOffset when PolygonMode = Point	G_PolygonSetting	-	POFP,POBP
Enable/disable of PolygonOffset when PolygonMode = Line			POFL,POBL
Enable/disable of PolygonOffset when PolygonMode = Fill			POFF,POBF
Setting of "factor" and "r × units"	G_PolygonOffset	-	factor,units

6.5.16 BitBlt (Bit Block Transfer)

This is a function to transfer a rectangle in units of pixels. When using the BitBlt function, horizontal width (XRES) of the frame buffer must be aligned in units of 8 bytes.

BitBlt processing includes the following types.

Table 6.5.69 BitBlt Processing Types

Processing type	Function
FILL	Fills a rectangular area in the specified color.
DRAW	Draws data supplied by display list, to a rectangular area.
COPY	Copies a rectangular area in the same frame buffer.
COPYALT	Copies a rectangular area between two different frame buffers. It is possible to copy between frame buffers whose shape and size are different.
COPYCOMP	Draws compressed data to the frame buffer while expanding it.

When there is an overlap area between the source and destination, the starting point for transfer must be set correctly.

When using BitBlt, specify the following attributes.

Table 6.5.70 BitBlt Drawing Attributes

Drawing attribute	Function
Transparent mode	Does not draw pixels of a color specified as transparent color. This mode cannot be used during Fill processing.
Forming mode	Draws only the pixels in which drawing target area matches the specified forming color.
Alpha map mode	Draws using alpha blending according to the alpha map. This mode can apply only to Fill or Copy, CopyCompressed. The commands for them are DrawRectAlphaMapP, BltCopyAltAlphaMapP, and BltCopyCompAlphaMapP, respectively.
Alpha blending mode	Draws while performing alpha blending at the blend ratio set by ALF. Only in 32-bit color mode, pixel alpha mode where the source pixel "A" component is used as the blend ratio can be used.
Logical operation mode	Performs binary logical operation between source data and destination data to write the result.

Transparent mode and logical operation mode cannot be specified simultaneously. In addition, logical operation mode is disabled in alpha blend mode and alpha map mode.

When alpha map mode or alpha blend mode and transparent mode are specified simultaneously, transparency determination is performed first, not drawing pixels in transparent color.

In 8-bit index color mode, blend processing is performed assuming 8 bits as one color component.

No update processing is performed for pixels including "A" component, whose blend ratio is "0" in alpha blend and alpha map mode. When the blend ratio is other than "0", "A" component of each pixel is as follows.

Table 6.5.71 Content of Drawing "A" Component When BitBlt Alpha Blending

Color mode	Content of upper bits
16-bit color	"A" component of pixel of the copy source image (1 bit)
32-bit color	"A" component of pixel of the copy source image (8 bits)

Table 6.5.72 Processing Types and Whether to Specify Drawing Attributes

Processing type	Logical operation	Transparent	Alpha blend	Alpha map	Forming	Background color transparent
FILL	○	×	○	○	○	×
CLEARPOLYFLAG	×	×	×	×	×	×
DRAW	○	○	○	×	○	×
BINARY	○	○	○	×	○	○
COPY	○	○	○	×	○	×
COPYALT	○	○	○	○	○	×
COPYCOMP	○	○	○	○	○	×

Table 6.5.73 Display Lists to Use BitBlt

Input content	Display list	Command	Processing type
Filling of rectangular area	DrawRectP	BltFill	FILL
Clear of polygon flag buffer		ClearPolyFlag	
Filling of rectangular area (alpha map provided)	DrawRectAlphaMapP	BltFill	
Drawing of rectangular pattern	DrawBitmapP	BltDraw	DRAW
Drawing of bit pattern		DrawBitmap	
Drawing of rectangular pattern (huge data supported)	DrawBitmapLargeP	BltDraw	
Copy of rectangular area (in the same frame buffer)	BltCopyP	TopLeft	COPY
		TopRight	
		BottomLeft	
		BottomRight	
Copy of rectangular area (between different frame buffers)	BltCopyAlternateP	TopLeft	COPYALT
Copy of rectangular area (alpha map provided)	BltCopyAltAlphaMapP	Normal	
		ABR	
Expansion and copy of compressed data	BltCopyCompressedP	TopLeft	COPYCOMP
Expansion and copy of compressed data (alpha map provided)	BltCopyCompAlphaMapP	TopLeft	

Table 6.5.74 Registers to Set Drawing Effect of BitBlt

Setting	Register	Field
Enable/disable of logical operation mode	MDR4	BM
Setting of logical operation function		LOG
Enable/disable of transparent mode	MDR4	TE
Setting of transparent color	Tcolor	-
Enable/disable of forming mode	MDR4	FE
Setting of forming color	FormColor	
Enable/disable of alpha blending mode	MDR4	BM
Setting of blend ratio of alpha blending	ALF	-
Selection of pixel alpha mode	MDR4	AS
Enable/disable of background color transparent	BC	BT
Setting of background color transparent	BC	-

Alpha map

Alpha map is the alpha blend ratio data having the same XY size as transfer data. It consists of 8-bit alpha blend ratio corresponding to each pixel of transfer data. Position of alpha map is specified by display list AMADDR or the ABR register.

Bit pattern drawing

This draws pixel of bitmap "1" in foreground color and pixel of bitmap "0" in background color using the binary bit map. Setting the BC register allows to make the background color transparent.

When both alpha blend and background transparency are specified, transparency determination is performed first, not drawing pixels in background color. Other alpha blend specifications are the same as that of BltDraw.

* The max. settable pixel size is 2096 when either height or width is set to the double character size.

Table 6.5.75 Registers to Set Bit Pattern Drawing

Setting	Register	Field
Horizontal direction scaling of bit pattern	MDR0	BSH
Vertical direction scaling of bit pattern		BSV
Specification of color corresponding to bit pattern "1"	FC	-
Specification of color corresponding to bit pattern "0"	BC	-
	MDR4	BW
Whether or not to make the pixel corresponding to bit pattern "0" transparent	BC	BT

BitBlt with MVP transformation

G_BitBlt is provided as the display list to perform BitBlt for the coordinate for which MVP transformation has been performed. A rectangle is drawn with the post-MVP transformation coordinate set as the center.

G_BitBlt behaves like a header with model coordinate. When G_BitBlt is added to the beginning of BitBlt for which the user wants to perform MVP transformation, BitBlt is performed.

Unlike in the case of graphics, DC_OFFSET does not apply to coordinate transformation performed using G_BitBlt.

Table 6.5.76 Display List to Perform BitBlt with MVP Transformation

Input content	Display list	Command	Field
BitBlt with MVP transformation	G_BitBlt	-	-

6.5.17 Drawing Effect of Straight Line

How to draw straight lines complies with the OpenGL Diamond-exit rule.

Thick line

Thickness can be specified when drawing straight line. Specify the width using the pixel count on the device coordinate irrespective of MVP transformation.

Table 6.5.77 Register to Set Width of Thick line

Setting	Register	Field
Width (pixel count) of straight line	MDR1	LW

Control of endpoint drawing

When controlling minutely the connection between straight lines, the user can forcibly draw to the coordinate specified as the endpoint. Usually, the endpoint coordinate is not drawn to avoid double input at the connection between straight lines.

Table 6.5.78 Register to Set Endpoint Drawing Control

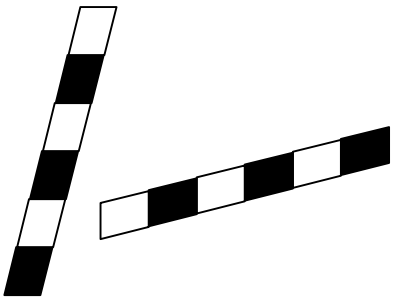
Setting	Register	Field
Whether or not to draw endpoint	GMDR1	ENDP

Broken line pattern

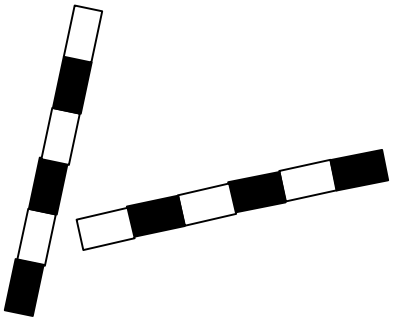
Broken line can be drawn by specifying bit broken line pattern. It is also possible to select transparent drawing for pixels of the broken line pattern "0".

How to draw broken line pattern can be selected from between the following two formats depending on the LINEEXT setting:

Broken line pattern vertical to the main axis (compatible with MB86290A (CREMSON))



Broken line pattern vertical to the theoretical line



Note the following points when using alpha blending at broken line drawing:

- The A component of the BC register is used as the blend ratio of the back color part.
- In the 16-bit color mode, the A component of the BC register can only be set to 00h or 01h. In the 16-bit color mode, when applying alpha blend to the back color part, set the blend ratio by inputting the following display list. Because this display list changes the value of the BLDCONST register, change the BLDCONST register back to the original value after drawing broken line.

```
F10100A6
00223344
F10100BC        // BLDCONST register
0000007F        // Set the same value as ALF register.
.
.               // Processing to draw broken line
.
F10100A6
33223344
```

Table 6.5.79 Registers to Set Broken Line Drawing

Setting	Register	Field
Enable/disable of broken line drawing	MDR1	BL
Broken line pattern repeat cycle		BP
Broken line pattern reference direction		BPD
Broken line pattern	BLP	-
Whether or not to initialize broken line pattern reference pointer	GMDR1	BLPC
Reference/setting of broken line pattern reference pointer	BLPO	-
Color corresponding to broken line pattern "1"	FC	-
Color corresponding to broken line pattern "0"	BC	-
	MDR1	BW
Transparency setting of pixel corresponding to broken line pattern "0"	BC	BT
Selection of broken line type	LINEEXT	BPM

Antialiasing

Antialiasing can smooth straight line's jaggies.

When smoothing a face boundary of triangle, etc., overwrite the face boundary with antialiased line. The boundary part is blended with the background color at the time of drawing, and it must be drawn from a farther primitive, if possible.

Do not specify this function and logical operation drawing simultaneously. When they are specified simultaneously, the drawing result is not guaranteed.

The "A" component of pixel written when antialiasing is used is treated in the same way as alpha blend. When only antialiasing is specified, the blend ratio for antialiasing is used as "A" component instead of alpha blend ratio. When both alpha blend and antialiasing are specified, a blend ratio obtained by multiplexing the blend ratio for antialiasing and the alpha blend ratio, is used. When the function of blend function is used with both of these blend ratios specified, the blend function is executed for the said resulting blend ratio.

Antialiasing does not work correctly when the blend function is ON (MD5.BFE = 0). When using antialiasing, set the blend function to OFF.

Table 6.5.80 Register to Set Antialiased Straight Line

Setting	Register	Field
Enable/disable of antialias for straight line	GMDR1	AA

Blend ratio in blend function and antialiasing

Alpha blending	Blend function	Blend ratio
Disable	Disable	Blend ratio of antialiasing
Enable		Alpha blend ratio × Antialias blend ratio
	Enable	Blend function is applied to "Alpha blend ratio × Antialias blend ratio"

Example of register setting in the case of using texture and antialiasing

Texture unit 0	Texture unit 1	Setting of BLDTU00 and BLDTU10
Disable	Disable	Default setting
Enable	Disable	[In the case of ignoring A factor of texture] BLDTU00:SRC0A=PRIMARY, BLDTU00:FUNCA=REPLACE [In the case of applying A factor of texture to blend ratio] BLDTU00:SRC0A=PRIMARY, BLDTU00:SRC1A=TEXTURE0, BLDTU00:FUNCA=REPLACE
Disable	Enable	[In the case of ignoring A factor of texture] BLDTU10:SRC0A=PRIMARY, BLDTU10:FUNCA=REPLACE [In the case of applying A factor of texture to blend ratio] BLDTU10:SRC0A=PRIMARY, BLDTU10:SRC1A=TEXTURE1, BLDTU10:FUNCA=REPLACE
Enable	Enable	

6.5.18 Indirect Display List

Indirect display list is a function to store a display list in graphics memory in advance and to make KOTTOS automatically read the display list. After this function is executed, the indirect display list returns to normal direct display list.

Inputting G_IndirectDL as indirect display list is prohibited. If it is input, a command error occurs.

Input content	Display list	Command	Field
Execution of display list stored in graphics memory	G_IndirectDL	-	-

6.5.19 Index Mode

Index mode is a function to store each element of vertex in graphics memory in advance and to reference them using index number. As with normal G_Vertex, when G_VertexIndex is input between G_Begin and G_End, drawing starts.

The user can select between two modes: one is DrawElement where the user specifies vertex number for each vertex, and the other is DrawArray where vertex number is automatically incremented from "0".

Table 6.5.81 Display Lists to Use Index Mode

Input content	Display list	Command	Field
Setting of base address of coordinate	SetIndexBaseAddress	COORD	-
Setting of base address of front face color		COLF	
Setting of base address of rear face color		COLB	
Setting of base address of normal vector		NORM	
Setting of base address of texture 0		TEX0	
Setting of base address of texture 1		TEX1	
Setting of base address of fog coordinate		F	
Setting of stride of coordinate	SetIndexStride	COORD	-
Setting of stride of front face color		COLF	
Setting of stride of rear face color		COLB	
Setting of stride of normal vector		NORM	
Setting of stride of texture 0		TEX0	
Setting of stride of texture 1		TEX1	
Setting of stride of fog coordinate		F	
Specification of vertex number	G_VertexIndex	DrawElement	-
Automatic setting of vertex number		DrawArray	

Table 6.5.82 Register to Specify Vertex Number Format of Index Mode

Setting	Register	Field
Specification of index number format	IDFOGL	DFIDX

6.5.20 Detection of end of drawing

G_Interrupt display list is provided to detect the end of the drawing. G_Interrupt generates a normal interrupt (KTS_int) and notifies the user that drawing has ended. The interrupt generation timing can be selected between the following two types:

Generate interrupt when interpreting G_Interrupt command:

When display list is stored in graphics memory, this interrupt is used to determine that the area used for storing becomes the used one. Drawing may still continue.

Generate interrupt after drawing has ended:

This interrupt is used to detect that drawing has ended. When a display list is input immediately after G_Interrupt, no interrupt occurs until drawing by this display list also ends.

Table 6.5.83 Display List to Detect End of Drawing

Input content	Display list	Command	Field
Controlling of interrupt generation when interpreting command	G_Interrupt	-	GFIFO
Controlling of interrupt generation when drawing ends			DRAWFIN

6.5.21 Debug function

KOTTOS has a function to detect undefined display list. When a command error (undefined display list is detected) occurs, an abnormal interrupt is output. Even when a command error occurs, processing by display list itself continues.

To locate the location where the error occurs, use the CMDERR and DL_CNT registers. The DL_CNT register is counted up each time display list is input, but when a command error occurs, it stops counting up. Display list that causes an error is recorded in the CMDERR register.

Table 6.5.84 Registers used by Debug Function

Setting	Register	Field
Generation of command error	VRERR	CERR
Command error interrupt mask	VRERRM	MCERR
Content of display list that causes an error	CMDERR	-
Display list counter	DL_CNT	-

6.5.22 2D Drawing Function

This function draws graphics by specifying device coordinates directly when no geometry processing is needed.

Drawing primitive

KOTTOS 2D drawing primitive includes the following:

Point drawing

Line drawing

Triangle drawing

Polygon drawing

Line and triangle drawings include two types of interfaces; setup interface and draw interface. The setup interface draws graphics by giving vertex coordinate of graphics, and the draw interface draws by giving a start point coordinate and inclination value.

The setup interface gives coordinates only, which reduces load of host CPU. The draw interface, on the other hand, is a mode for upward compatibility, so the inclination value must be calculated on the host CPU.

Point drawing

Draws a point consisting of a pixel.

Table 6.5.85 Used Display Lists

Type	Command
DrawPixel	Pixel
DrawPixelZ	PixelZ

DrawPixelZ performs Z comparison without setting of ZC of the MDR1 register.

Line drawing

Draws a line. Specify the following attributes to draw a line.

Table 6.5.86 Line Drawing Attributes

Attribute	Functions
Line width	Specifies line width ranging from 1 to 32 pixels.
Broken line	Specifies broken line pattern among 32/24/16-bit patterns
Antialiasing	Draws smooth and antialiased line when an antialiasing command is selected.
End-point drawing control	Sets draw/not-draw the last one pixel. Setting "not-draw" prevents the connection part of consecutive straight lines from being drawn twice.
Shading	Specifies Gouraud shading or flat shading.
Alpha-blending	Sets alpha-blending/not-alpha-blending. Specifies transmission by surfaces or pixels.
Logical operation	Sets perform logical operation/not perform logical operation and sets how to operate.
Z comparison	Sets perform hidden surface removal/not perform hidden surface removal and sets how to compare.
Texture/tiling	Sets perform texture or tiling/ not perform texture or tiling.
Fog	Sets perform fog processing/ not perform fog processing according to depth.

To clear broken line pointer, to perform antialiasing, and to control end-point drawing, use a display list command instead of mode registers.

Drawing parameter of setup interface

The setup interface draws graphics by specifying coordinate of each vertex and parameter. When the thick line is specified, the depth of the line increases toward both directions centering on the ideal line. At this point, parameters with inclination such as texture coordinate and "Z" value change towards its principal axis and the same parameters are used to change towards sub axis.

Display lists that can be specified only X and Y coordinates include Drawline2i/2iP. These display lists are only for compatibility, so Fujitsu recommends SetVertex/DrawVertex be used.

Table 6.5.87 Used Display Lists

Type	Command
SetVertex/DrawVertex	Line PackedInt ZeroVector Line Fixed ZeroVector
	Line PackedInt OneVector Line Fixed OneVector
	Line PackedInt ZeroVectorNoEnd Line Fixed ZeroVectorNoEnd
	Line PackedInt OneVectorNoEnd Line Fixed OneVectorNoEnd
	Line PackedInt ZeroVectorBlpClear Line Fixed ZeroVectorBlpClear
	Line PackedInt OneVectorBlpClear Line Fixed OneVectorBlpClear
	Line PackedInt ZeroVectorNoEndBlpClear Line Fixed ZeroVectorNoEndBlpClear
	Line PackedInt OneVectorNoEndBlpClear Line Fixed OneVectorNoEndBlpClear
	Line PackedInt AntiZeroVector Line Fixed AntiZeroVector
	Line PackedInt AntiOneVector Line Fixed AntiOneVector
	Line PackedInt AntiZeroVectorNoEnd Line Fixed AntiZeroVectorNoEnd
	Line PackedInt AntiOneVectorNoEnd Line Fixed AntiOneVectorNoEnd
	Line PackedInt AntiZeroVectorBlpClear Line Fixed AntiZeroVectorBlpClear
	Line PackedInt AntiOneVectorBlpClear Line Fixed AntiOneVectorBlpClear
	Line PackedInt AntiZeroVectorNoEndBlpClear Line Fixed AntiZeroVectorNoEndBlpClear
	Line PackedInt AntiOneVectorNoEndBlpClear Line Fixed AntiOneVectorNoEndBlpClear
DrawLine2i/DrawLine2iP	ZeroVector
	OneVector
	ZeroVectorNoEnd
	OneVectorNoEnd
	ZeroVectorBlpClear
	OneVectorBlpClear
	ZeroVectorNoEndBlpClear
	OneVectorNoEndBlpClear
	AntiZeroVector
	AntiOneVector
	AntiZeroVectorNoEnd
	AntiOneVectorNoEnd
	AntiZeroVectorBlpClear
	AntiOneVectorBlpClear
	AntiZeroVectorNoEndBlpClear
	AntiOneVectorNoEndBlpClear

Drawing parameter of draw interface

The draw interface draws graphics by specifying the start point and incremental value. Unlike the setup interface, when the thick line is specified, the depth of the line increases downwards (for X principal axis) or increases rightwards (for Y principal axis) from the ideal line. The draw interface is a command only for compatibility, so Fujitsu recommends the setup interface be used.

Table 6.5.88 Used Display List

Type	Command
DrawLine	Xvector
	Yvector
	XvectorNoEnd
	YvectorNoEnd
	XvectorBlpClear
	YvectorBlpClear
	XvectorNoEndBlpClear
	YvectorNoEndBlpClear
	AntiXvector
	AntiYvector
	AntiXvectorNoEnd
	AntiYvectorNoEnd
	AntiXvectorBlpClear
	AntiYvectorBlpClear
	AntiXvectorNoEndBlpClear
	AntiYvectorNoEndBlpClear

Triangle drawing

To draw a triangle, specify the following attributes.

Table 6.5.89 Triangle Drawing Attributes

Attribute	Function
Shading	Specifies Gouraud shading or flat shading.
Alpha-blending	Sets alpha-blending/not-alpha-blending. Specifies transmission by surfaces or pixels.
Logical operation	Sets perform logical operation/not perform logical operation and sets how to operate.
Z comparison	Sets perform hidden surface removal/not perform hidden surface removal and sets how to compare
Texture/tiling	Sets perform texture or tiling/ not perform texture or tiling.
Fog	Sets perform fog processing/ not perform fog processing according to depth.

To prevent double inputting of pixels in drawing the vertex-shared triangle, a triangle is drawn in accordance with the following rules:

Top and left sides are drawn including the specified coordinate.

Bottom and right sides are drawn inside the specified coordinate.

Coordinate whose width or height is less than 1 pixel is not drawn.

Drawing parameter of setup interface

The setup interface draws graphics by specifying the coordinates of each vertex, color code, "Z" value, texture coordinate, and fog coordinate. The setup interface uses SetVertex and DrawVertex. When specifying only the X and Y coordinates, XY setup interface (DrawVertex2i and DrawVertex2iP) can be used. However, the X and Y coordinates can be specified by the setup interface, Fujitsu recommends SetVertex and DrawVertex be used.

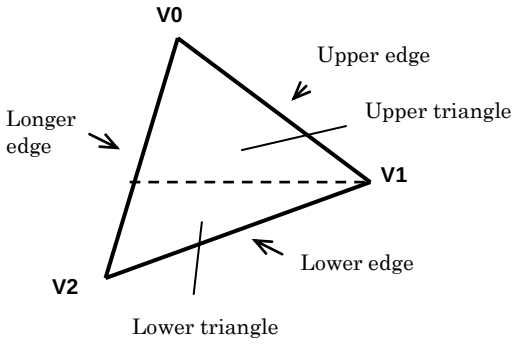
Table 6.5.90 Used Display Lists

Type	Command
DrawVertex2i/DrawVertex2iP	TriangleFan
SetVertex	Normal
DrawVertex	TriangleFan

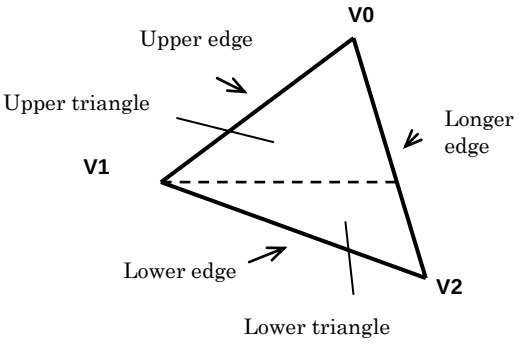
Drawing parameter of draw interface

The draw interface uses the following parameters to draw graphics. The draw interface is only for compatibility, and Fujitsu recommends the setup interface be used.

The draw interface differentiates right triangle and left triangle for longer edge in the direction of Y axis, as follow.

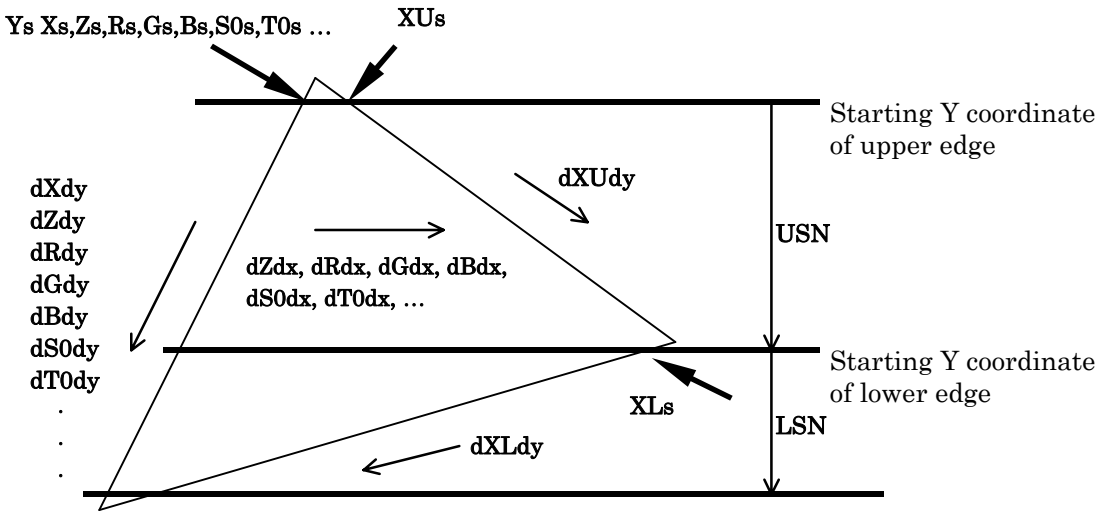


Right triangle



Left triangle

Parameters required to draw triangle by the draw interface are as follows:



Notes:

For relationship between Xs, XUs, and XLs

For example, in the figure above, when drawing the right triangle with parameters of the coordinate relations of “Xs (Starting X coordinate of upper edge) > XUs” or “Xs (Starting X coordinate of lower edge) > XLs”, desired graphics cannot be drawn.

Table 6.5.91 Parameter Setting Registers of Draw Interface

Ys	Starting Y coordinate of longer edge when drawing triangle
Xs	Starting X coordinate of longer edge corresponding to Ys
XUs	Starting X coordinate of upper edge
XLs	Starting X coordinate of lower edge
Zs	Starting Z coordinate of longer edge corresponding to Ys
Rs	Starting color value R of longer edge corresponding to Ys
Gs	Starting color value G of longer edge corresponding to Ys
Bs	Starting color value B of longer edge corresponding to Ys
S0s	Starting texture 0 coordinate value S of longer edge corresponding to Ys
T0s	Starting texture 0 coordinate value T of longer edge corresponding to Ys
Q0s	Starting texture 0 coordinate value Q of longer edge corresponding to Ys
S1s	Starting texture 1 coordinate value S of longer edge corresponding to Ys
T1s	Starting texture 1 coordinate value T of longer edge corresponding to Ys
Q1s	Starting texture 1 coordinate value Q of longer edge corresponding to Ys
Fs	Starting fog coordinate value F of longer edge corresponding to Ys
dXdy	X incremental value in the direction of longer edge
dXUdy	X incremental value in the direction of upper edge
dXLdy	X incremental value in the direction of lower edge
dZdy	Z incremental value in the direction of longer edge
dRdy	R incremental value in the direction of longer edge
dGdy	G incremental value in the direction of longer edge
dBdy	B incremental value in the direction of longer edge
dS0dy	S incremental value of texture 0 in the direction of longer edge
dT0dy	T incremental value of texture 0 in the direction of longer edge
dQ0dy	Q incremental value of texture 0 in the direction of longer edge
dS1dy	S incremental value of texture 1 in the direction of longer edge
dT1dy	T incremental value of texture 1 in the direction of longer edge
dQ1dy	Q incremental value of texture 1 in the direction of longer edge
dFdy	F incremental value in the direction of longer edge
USN	Span count of upper triangle
LSN	Span count of lower triangle
dZdx	Z incremental value in the horizontal direction
dRdx	R incremental value in the horizontal direction
dGdx	G incremental value in the horizontal direction
dBdx	B incremental value in the horizontal direction
dS0dx	S incremental value of texture 0 in the horizontal direction
dT0dx	T incremental value of texture 0 in the horizontal direction
dQ0dx	Q incremental value of texture 0 in the horizontal direction
dS1dx	S incremental value of texture 1 in the horizontal direction
dT1dx	T incremental value of texture 1 in the horizontal direction
dQ1dx	Q incremental value of texture 1 in the horizontal direction
dFdx	F incremental value in the horizontal direction

Table 6.5.92 Used Display List

Type	Command
DrawTrap	TrapRight
	TrapLeft

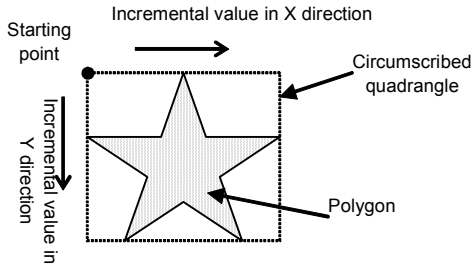
Polygon drawing

Hardware can draw polygon including concave shape. Follow the procedures below.

1. Executing the SetVertex2i/2iP:PolygonBegin command
   Initialize the hardware to draw polygon and set the first vertex.
2. Setting the second vertex (SetVertex2i/2iP:Normal)
   Set the coordinate of the second vertex of the polygon.
3. Drawing vertex array (DrawVertex2i/2iP:FlagTriangleFan)
   Set the vertex coordinate of polygon using the DrawVertex2i/2iP:FlagTriangleFan.
4. Executing Draw:PolygonEnd command
   Draws polygon for the drawing frame by referencing result of drawing into polygon flag buffer. After drawing, the used polygon flag buffer is cleared automatically.

- How to set Z comparison, fog and texture

Use the MDR2 register to set drawing attribute for polygon drawing, like the case with triangle. Set drawing parameters of each attribute (Z value, and starting value and incremental value of texture coordinate) to the circumscribed quadrangle of polygon, instead of values of each vertex.



Circumscribed quadrangle becomes a little larger than rectangle area than that calculated from the maximum and minimum values of all vertex coordinates of polygon. Use this function only when the error can be allowed.

For the "Z" value, only a single value can be set for all polygons. Set the "Z" incremental value to "0".

Specify Z value, fog coordinate and the starting value and increment value of texture coordinate before setting the PolygonEnd command, like the draw interface of triangle. Registers to be used are triangle parameter register shown in *Table 6.5.91*.

As the starting value, set the top left coordinate value of the circumscribed quadrangle area of polygon. As the incremental value, in the case of X direction, set the incremental value in the X direction of circumscribed quadrangle; in the case of Y direction (longer edge direction), set the incremental value in the Y direction of circumscribed quadrangle.

Because the operation for Z value, Fog and Texture coordinate in the polygon is performed for the circumscribed quadrangle area, parameters of the starting value (top left on the circumscribed quadrangle area) may be greater than that of vertex of the polygon itself.

Parameters given to the polygon are limited to the originally-specifiable maximum value and minimum value, in the range between the starting point (top left on the circumscribed quadrangle) and the end point (lower right on the circumscribed quadrangle). If the value exceeds the range, drawing is not performed correctly.

Gouraud shading cannot be used for polygon drawing.

Table 6.5.93 Used Display Lists

Type	Command
SetVertex2i/SetVertex2iP	PolygonBegin
DrawVertex2i/DrawVertex2iP	FlagTriangleFan
Draw	PolygonEnd

6.5.23 Top-left Rule Not-applicable Primitive Drawing

In drawing triangle and polygon, the inside of sides connecting vertices is drawn. However, to prevent double inputting of pixels, a special rule called "top-left rule" applies to parts on the sides of graphics. Due to this rule, sometimes, the top and left sides of graphics are drawn and the right and lower sides of graphics are not drawn.

The top-left rule not-applicable drawing is a function to not apply the top-left rule and draw the graphics exactly according to the coordinates specified by the user. In ***Fig. 6.17*** below, there is a gap between the lower side of the graphics A and the top side of the graphics B. However, when using coordinate data created on the assumption that there is no gap between graphics A and B, these two graphics must be tangential to one another. In such a case, use this function to get the precise drawing result intended by the data creator.

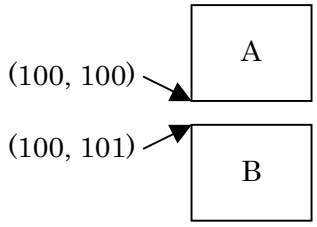


Fig. 6.17 Top-left Rule Not-applicable Drawing

Do not use coordinate transformation when using the top-left rule not-applicable drawing. When performing processing involving some coordinate transformation such as rotation, scaling, etc., the distance changes between graphics that do not share vertices.

To get the correct drawing result by not applying the top-left rule, input a coordinate with a device coordinate passing through the center point of each pixel. Because the decimal part of the coordinate of the center point of the pixel is ".5", input accordingly, that is, use a coordinate with a device coordinate decimal part of ".5". When the original coordinate data is an integer, use the G_Viewport offset or the DC_OFFSET register to automatically add "0.5".

To set top-left rule not-applicable drawing, use the following procedures:

(1) Use the SetGModeRegister to set "0" to the IDFM bit of the GMDR0 register (Coral mode).

(2) Use SetGModeRegister to set "1" to the TL bit of the GMDR2E register (top-left rule does not apply).

(3) Use the OverlapZOfft to set the Z offset of the body.

(4) Use the OverlapZOfft to set the Z value offset used for the top-left rule not-applicable mode.

Table 6.5.94 Display Lists to Set Top-left Rule Not-applicable Drawing

Input	Display list	Command	Field
Setting of input data format: Coral/OpenGL	SetGModeRegistaer	GMDR0	IDFM
Application/not-application of top-left rule	SetGModeRegistaer	GMDR2E	TL
Setting of Z offset of body	OverlapZOfft	Origin	Z Offset
Setting of Z offset used in top-left rule not-applicable mode	OverlapZOfft	NonTopLeft	Z Offset
* When setting Z offset of body and Z offset used in top-left rule not-applicable mode using 8 bits	OverlapZOfft	Packed_ONBS	O_Z Offset/ N_Z Offset

6.5.24 Shadow Primitive Drawing

The shadow primitive drawing function provides shadows to graphics when drawing triangle and polygon. This function can be executed only in the Coral mode, and cannot be applied to point and straight line drawing. When using this function, add offset to the body value so as not to exceed each value range.

To set shadow primitive drawing, use the following procedures:

(1) Use the SetGModeRegister to set "0" to the IDFM bit of the GMDR0 register (Coral mode).

(2) Use the SetGModeRegister to set "1" to the SP bit of the GMDR2E register (shadow primitive drawing).

(3) Use the OverlapXYOfft to set the XY offset of the shadow.

(4) Use the OverlapZOfft to set the Z offset of the body.

(5) Use the OverlapZOfft to set the Z offset of the shadow.

(6) Use the SetColorRegister to set the color (front face, back face) of the body.

(7) Use the SetColorRegister to set the color of the shadow.

Table 6.5.95 Display Lists to Set Shadow Primitive Drawing

Input	Display list	Command	Field
Setting of input data format: Coral/OpenGL	SetGModeRegistaer	GMDR0	IDFM
Enable/disable of shadow primitive drawing mode	SetGModeRegistaer	GMDR2E	SP
Setting of XY offset of shadow	OverlapXYOfft	C_OXYO	X Offset/ Y Offset
Setting of Z offset of body	OverlapZOfft	Origin	Z Offset
Setting of Z offset of shadow	OverlapZOfft	Shadow	Z Offset
* When setting Z offset of body and shadow using 8 bits	OverlapZOfft	Packed_ONBS	O_ZOffset/ S_ZOffset
Setting of foreground color of body primitive	SetColorRegister	ForeColor	FGC8/16/24
Setting of background color of body primitive	SetColorRegister	BackColor	FGC8/16/24
Setting of color of primitive shadow	SetColorRegister	ForeColorShadow	FGC8/16/24

6.5.25 Coral-compatible Drawing Function

In Coral, the function (such as shadowed drawing, top-left rule not-applicable drawing, polygon with Z or texture) that must be started using G_BeginE has been implemented by a simplified algorithm, so the intended result may not be obtained.

For example, the front and back relationship between the shadow graphics and the body graphics is determined by the OverlapZ setting, but when the body reaches the maximum value of the Z value range, the value of the shadow cannot be represented (due to overflow), resulting in failure.

For a polygon with Z or texture, representative points are selected, and all pixel values of the face are determined based on the value of these vertices. Consequently, the operation accuracy may not be maintained. In principle, the entire computer graphics uses value approximation, so reduced accuracy can occur in the entire graphics. Note that because the above function is implemented by a simplified algorithm, it tends to cause more reduced accuracy than normal.

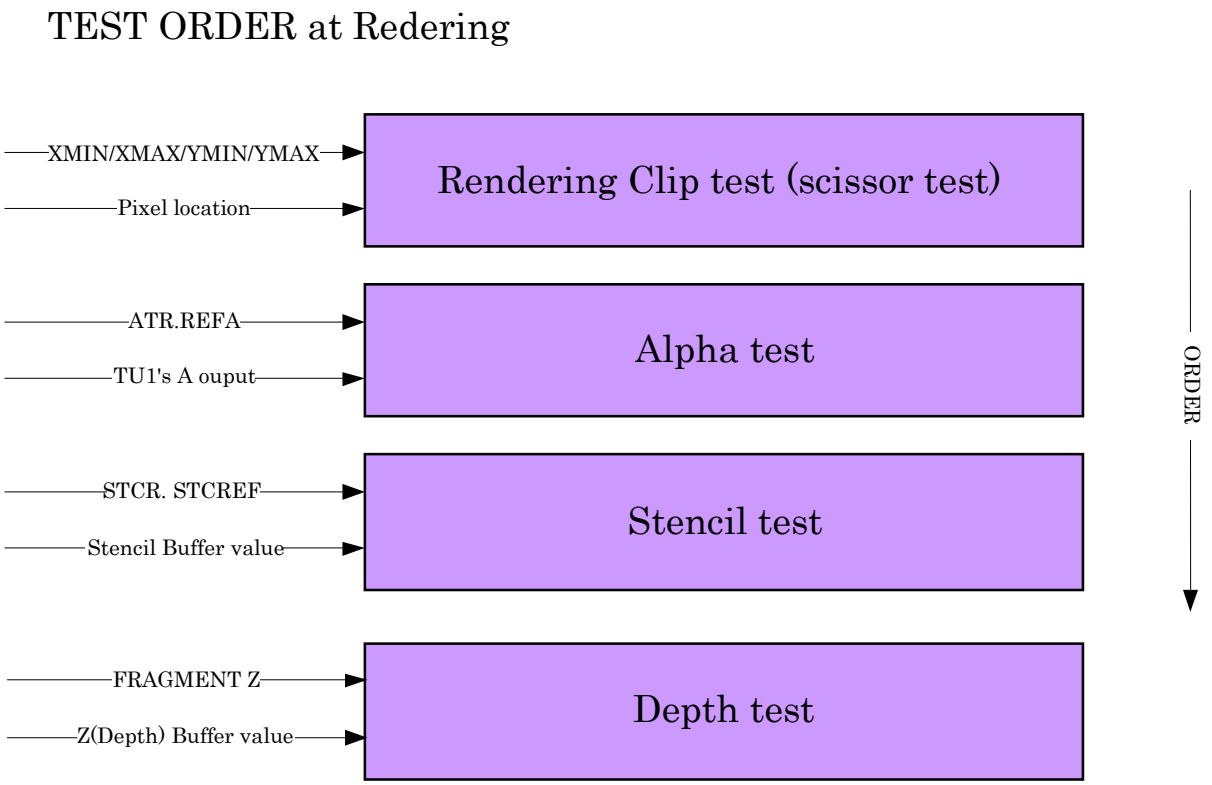
6.5.26 Processing Sequence

Pixel processing is performed by the pixel processing program. The processing is dependent on the pixel processing program. Typical processing sequences are as follows:

Texture mapping -> fog -> antialiasing -> clipping -> alpha test -> stencil test -> color mask -> depth buffer test -> alpha blending or logical operation

6.5.26.1 Test Processing Order

The test execution order is shown below.



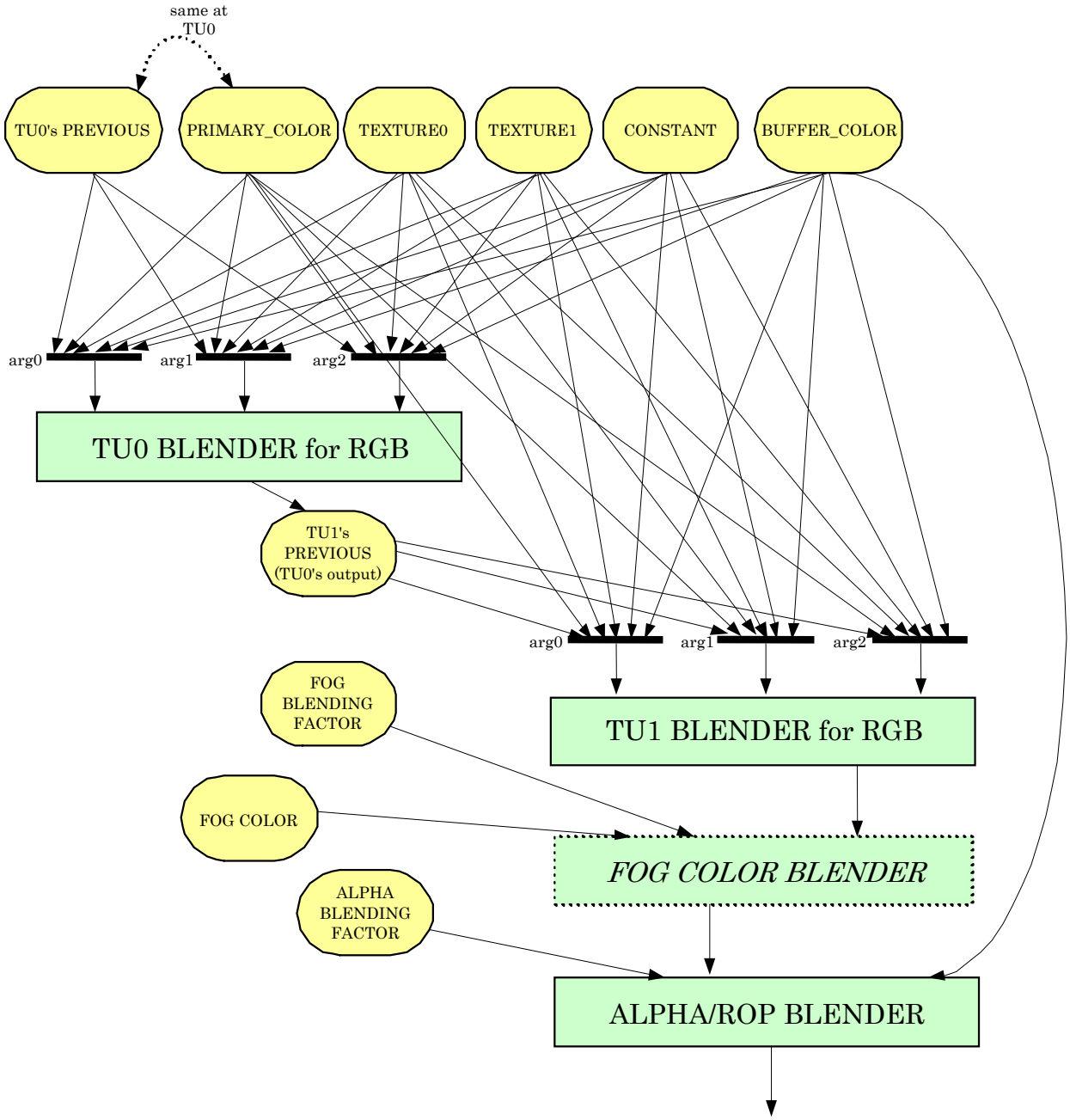
The tests are executed in the above order.
(Pixels of interest are drawn only when the pixels have passed all the tests.)
For example, when pixels fail the stencil test, the depth test is not executed.
Therefore, operation of stencil buffer in the stencil test can have only the following three definitions:

StencilFail
StencilPass && DepthFail
StencilPass && DepthPass

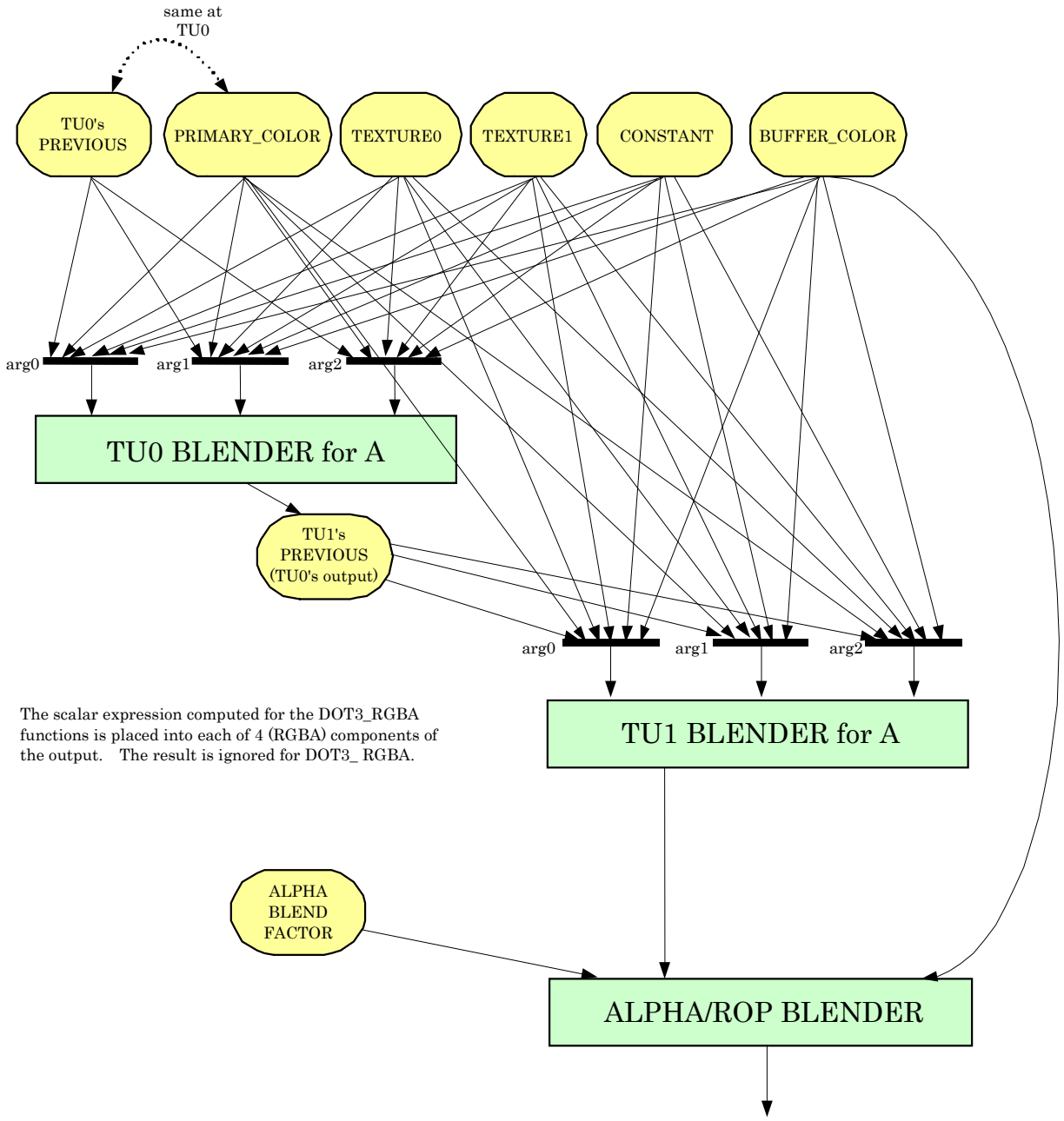
6.5.26.2 Sequence of blend processing

The blend processing sequence and the relationship between each source and its output are shown below.

BLEND SOURCE SELECT & BLEND ORDER (for RGB)



BLEND SOURCE SELECT & BLEND ORDER (for A)



Blend is performed in the above order.
TU0 and TU1 can arbitrarily specify the source and operation function using the BLDTUxx register.
At specification of the source, PREVIOUS means the output of the blender at the previous stage.
When the blender at the previous stage is disabled, the output at the stage preceding the previous stage is enabled. PREVIOUS of TU0 is a fragment color (a color obtained by interpolating the vertex color).
In addition, blending with FOGCOL depending on the fog processing is performed at the above position, but the blend is implicitly controlled by setting whether the FOG function is enabled or disabled.

The TEXTURE0/1 as source is pre-blend texel data (filter processing at the texture engine level, such as bilinear interpolation, is complete) itself. When TU1 requires output of TU0, specify PREVIOUS, not TEXTURE0.

Also, note the following restrictions on texture blend processing:

When specifying TEXTURE0 as the source, enable TU0.
When specifying TEXTURE1 as the source, enable TU1.

7 Display Controller

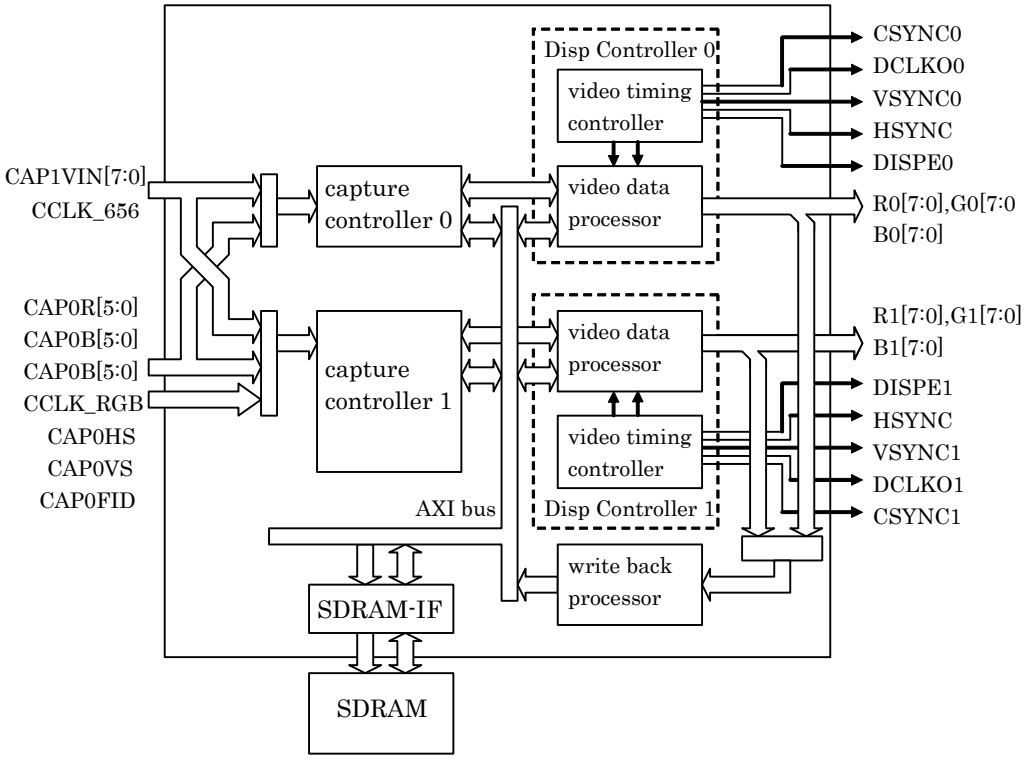
7.1 Overview

7.1.1 Entire Configuration

Carmine has two display controllers that are backward compatible with Fujitsu existing product MB86296 (Coral-PA). The display controllers generate synchronization signals independently based on individual pixel clocks. The two display controllers can be used as, for example, XGA and VGA. In addition, this chip can be synchronized with separate video capture sources; one can be synchronized with 50Hz, and the other 60Hz. Using these controllers and the dual display function of the Coral-PA simultaneously, up to four screens can be displayed.

Using the writeback function allows to store one of two video outputs in graphics memory.

This chip has two capture controllers corresponding to the two display controllers. The display controller and the capture controller correspond one-to-one. For the capture input, there are two types of switchable format: 656 format and RGB format. RGB-format input pins can be used as 656-format ones.



7.1.2 Individual functions of display controller

Display control

Displays eight-layer window. Also, it can perform screen scrolling.

Upward compatibility

Supports upward compatibility with existing products when used in four-layer and split display (for left/right) mode.

Dual display by single display controller

A single display controller can output two different screens. The controller can output any layer to two screens.

Alpha dedicated layer

Provides four 8 bits/pixel layers for alpha processing.

Video timing generator

Generates video timing supporting the screen resolution of “320 × 240” to “1280 × 1024”.

Color lookup table

Incorporates four color lookup tables (palette RAM) for frame display in index color mode (8 bits/pixel).

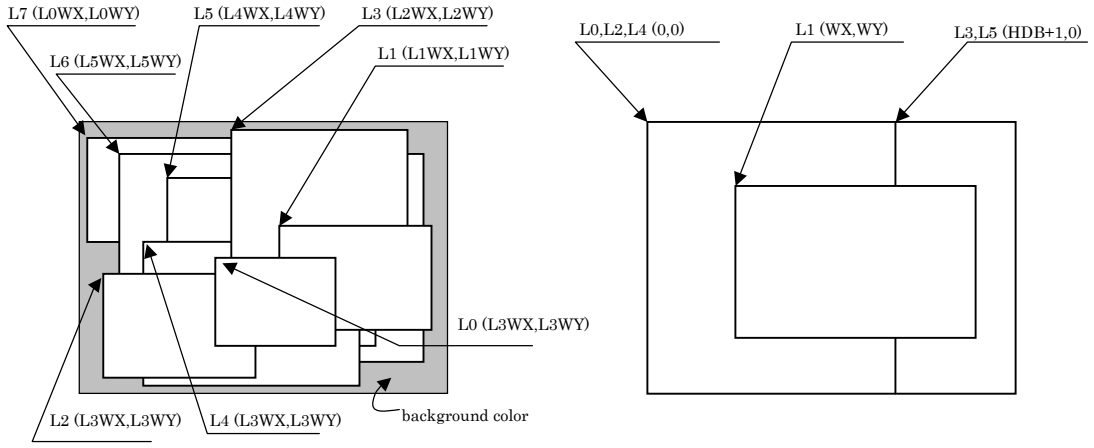
Cursor

Two “64 × 64” pixel cursors can be used in 8-bit index color mode.

7.2 Display Function

7.2.1 Screen Structure

The display controller displays windows using eight-layered window display. The layer stacking sequence can be set arbitrarily. Also, the controller has four-layer and split display (for left/right) mode, supporting upward compatibility with existing products.



(a) Eight layerd window display (b) Four layered display for backward compatibility

Fig. 7.1 Structure of Display Screen

Correspondence of the display layers to existing products is as shown below.

Layer correspondence		Coordinates of the starting point		Width and height	
		(a) Window	(b) Compatibility	(a) Window	(b) Compatibility
L0	C	(L0WX,L0WY)	(0,0)	(L0WW,L0WH+1)	(HDP+1,VDP+1)
L1	W	(L1WX,L1WY)	(WX,WY)	(L1WW,L1WH+1)	(WW,WH+1)
L2	ML	(L2WX,L2WY)	(0,0)	(L2WW,L2WH+1)	(HDB+1,VDP+1)
L3	MR	(L3WX,L3WY)	(HDB,0)	(L3WW,L3WH+1)	(HDP-HDB,VDP+1)
L4	BL	(L4WX,L4WY)	(0,0)	(L4WW,L4WH+1)	(HDB+1,VDP+1)
L5	BR	(L5WX,L5WY)	(HDB,0)	(L5WW,L5WH+1)	(HDP-HDB,VDP+1)
L6	---	(L6WX,L6WY)	(0,0)	(L6WW,L6WH+1)	(HDP+1,VDP+1)
L7	---	(L7WX,L7WY)	(0,0)	(L7WW,L7WH+1)	(HDP+1,VDP+1)

C, W, ML, MR, BL and BR above are layers of existing products. Selection between window mode and compatibility mode can be performed on a layer by layer basis. When a mixture of display modes is allowed instead of separating them completely, new functions can be used by making minor changes to the programs that run on the existing products. When displaying an image with a high resolution, the number of layers or pixel data that can be displayed concurrently may be limited by the data supply capability of graphics memory.

7.2.2 Stacking

7.2.2.1 Overview

Image data for eight layers (L0 to L7) is processed as follows.

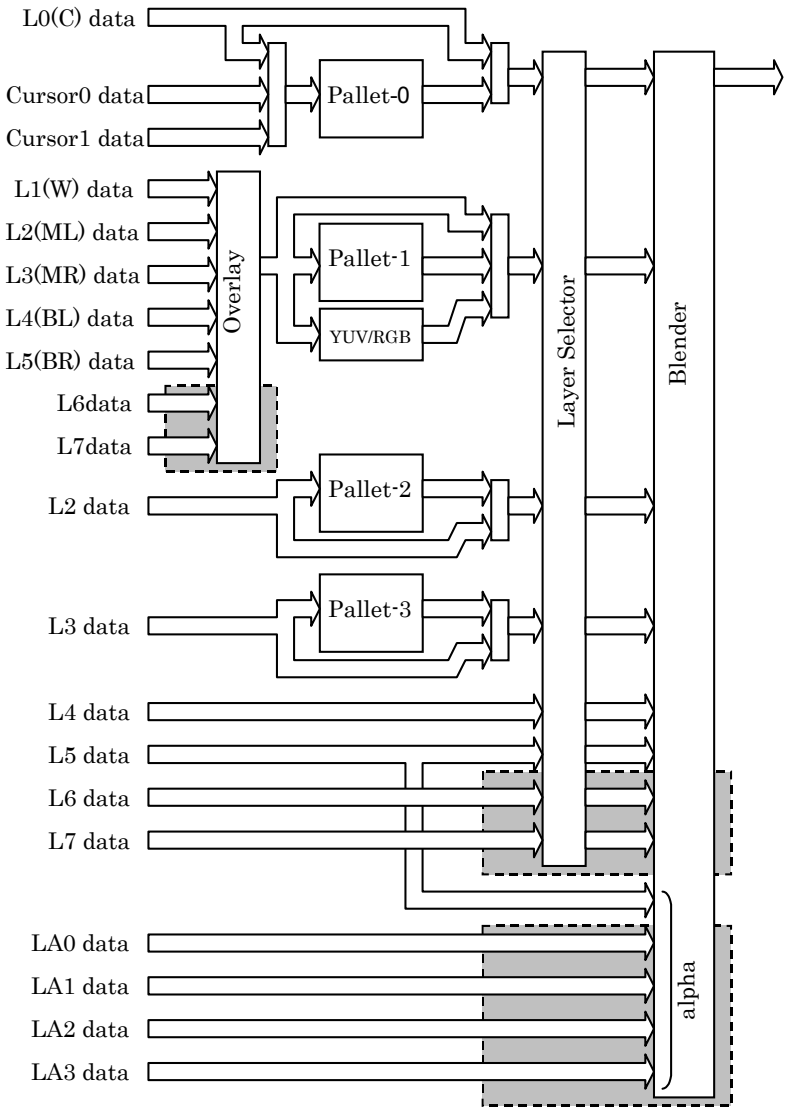


Fig. 7.2 Structure of Display Layer

The basic flow is "palette → layer selector → blender". Palette converts 8-bit color data to RGB format. In the layer selector, the stacking sequence is changed arbitrarily. The blender performs blending using a blend factor defined for each layer, or performs stacking in accordance with the transparent color definition. L0 layer corresponds to the C layer for existing products, and shares the Palette with the cursor. So, L0 layer and the cursor are designed to be stacked before blend operation is performed.

L1 layer corresponds to the W layer for existing products. For upward compatibility with existing products, L1 layer is stacked with the lower layer before blend operation is performed.

L2 to L7 layers are processed in 2 ways: one is “inputting them individually into the blender”, and the other is “stacking them with L1 layer and then inputting the result into the blender”. When stacking in extended mode, select the former method; when stacking in the same way as existing products, select the latter method. The user can specify which to choose on a layer by layer basis.

7.2.2.2 Stacking Mode

There are two modes to stack image layers: Simple priority mode and the blend mode.

In simple priority mode, superimposition is performed in accordance with the transparent color defined on a layer by layer basis, and when a layer’s color is a transparent color, the value of the layer lower than this layer is set as the image value for the next stage; when a layer’s color is not a transparent color, its own value is set as the image value for the next stage.

$D_{view} = D_{new}$ (when D_{new} does not match the transparent color)

$= D_{lower}$ (when D_{new} matches the transparent color)

When L1 layer is in YCbCr mode, it is always processed as not being transparent, not determining whether or not it's color is a transparent color.

In blend mode, the following calculation is performed, specifying blend ratio r defined on a layer by layer basis, in 8-bit precision: $D_{view} = D_{new} \times r + D_{lower} \times (1 - r)$

Blend is enabled by mode setting on a layer by layer basis. And at this time, the A (alpha) field for the pixel must be set to 1. For 8 bits/pixel, the bit specifying that blend be enabled is MSB of the Palette RAM data; for 16 bits/pixel, it is MSB of the layer’s own data; for 24 bits/pixel, it is MSB of the word. This bit position is for ARGB format; the bit position for RGBA format is different from that for ARGB format (see data format).

In Carmine, a mode is added where the A field for the pixel is ignored and blend is always enabled. This mode is selected using LnID (Ln Ignore bit of blended Data).

LnID = 0: Blend target data’s A field is enabled.

LnID = 1: Blend target data’s A field is ignored.

7.2.2.3 Blend Factor Layer

In ordinary blend mode, the blend factor is fixed to a certain value on a layer by layer basis, but there is also a mode where the value of a blend factor dedicated layer (LA0 to LA3) can be set as the blend factor. For blend factor dedicated layers, only 8 bits/pixel can be specified. In this mode, the blend factor can be specified on a pixel by pixel basis; for example, gradation can be provided. The blend factor is also called the alpha coefficient.

To ensure compatibility with existing products, L5 layer can also be set as the blend factor layer. In this case, L5 layer is set to 8 bits/pixel window display mode and superimposition extended mode.

7.2.2.4 Control of Blend Factor Layer

Whether the blend factor dedicated layer is enabled or disabled is controlled using the DCM1 register's LA0E to LA3E bits. Each blend target layer can select from among LA0 to LA3 as the blend factor.

The selection is performed using the LnBLD register's LnAL (Ln Alpha Layer, 2 bits) field.

LnAL = 00: LA0 layer is selected.

LnAL = 01: LA1 layer is selected.

LnAL = 10: LA2 layer is selected.

LnAL = 11: LA3 layer is selected.

When using L5 layer as the alpha layer, as with existing products, use LnAL=00. Otherwise, an undefined value is enabled as the alpha field.

When using an alpha dedicated layer, select it using the LnBLD register's LnAS (Ln alpha source) bit.

LnAS=0: L5 layer is used as the alpha layer.

LnAS=1: An alpha dedicated layer is used.

This bit is common to all layers. That is, when a value is set for this bit for one layer, this value is used for all the other layers.

7.2.3 Display Parameter

A display area is defined using the parameters shown below. Each parameter is set as a register value.

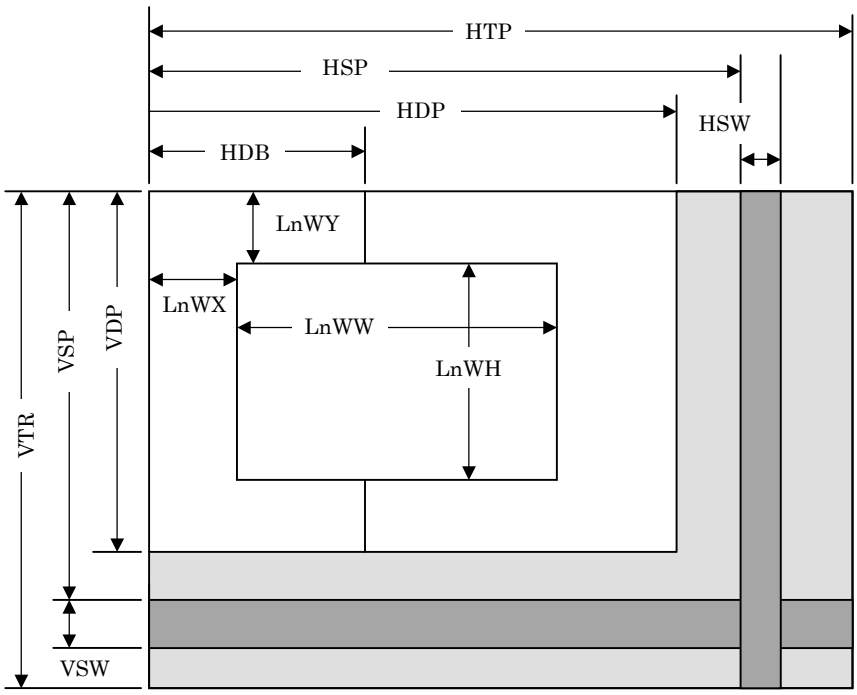


Fig. 7.3 Display Parameter

Note: The above setting of display parameters is slightly different from the actual one. For details, see the display timing.

HTP	Horizontal Total Pixels
HSP	Horizontal Synchronize pulse Position
HSW	Horizontal Synchronize pulse Width
HDP	Horizontal Display Period
HDB	Horizontal Display Boundary
VTR	Vertical Total Raster
VSP	Vertical Synchronize pulse Position
VSW	Vertical Synchronize pulse Width
VDP	Vertical Display Period
LnWX	Layer n Window position X
LnWY	Layer n Window position Y
LnWW	Layer n Window Width
LnWH	Layer n Window Height

When not performing screen split, set HDP = HDB. Then, only the left half of the screen is displayed.

The set values must meet the following magnitude relations:

$0 < HDB \leq HDP < HSP < HSP + HSW + 1 < HTP$

$0 < VDP < VSP < VSP + VSW + 1 < VTR$

7.2.4 Control of display position

Image data to be displayed is placed in a logical 2-dimensional coordinate space (a logical image space) in graphics memory. There are the following 8 logical image spaces as spaces holding display images.

L0 layer, L1 layer, L2 layer, L3 layer

L4 layer, L5 layer, L6 layer, L7 layer

Also, there are the following layers as data spaces not directly displayed. These layers are used as coefficients for performing display blend.

LA0 layer, LA1 layer, LA2 layer, LA3 layer

The relation between logical image space and display position is defined as shown below.

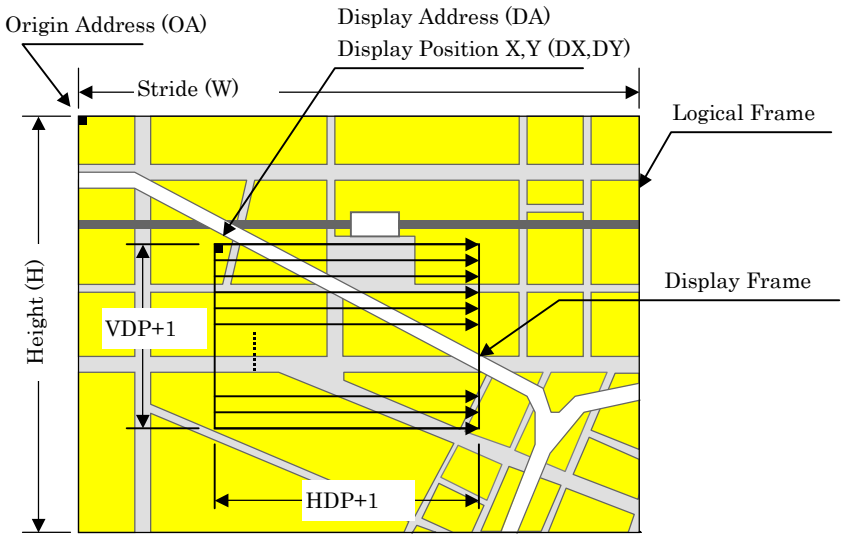


Fig. 7.4 Display Position Setting Parameter

OA	Origin Address	Logical space origin address Memory address of the pixel, which is the origin (upper left) of the logical frame.
W	Stride	Logical space (memory) width Width of the logical frame in units of 64 bytes
H	Height	Logical space height Raster count of the logical space
DA	Display Address	Display origin address Address of the origin (upper left) of the display frame
DX DY	Display Position	Display origin coordinates Coordinates of the origin of the display frame, in the logical frame space

The logical space is subjected to display scan both horizontally and vertically as if it were connected cyclically. When displaying an image beyond the boundary of a logical space by using this function, smooth scroll is made possible by additionally drawing the continued graphics in the part that goes beyond the boundary.

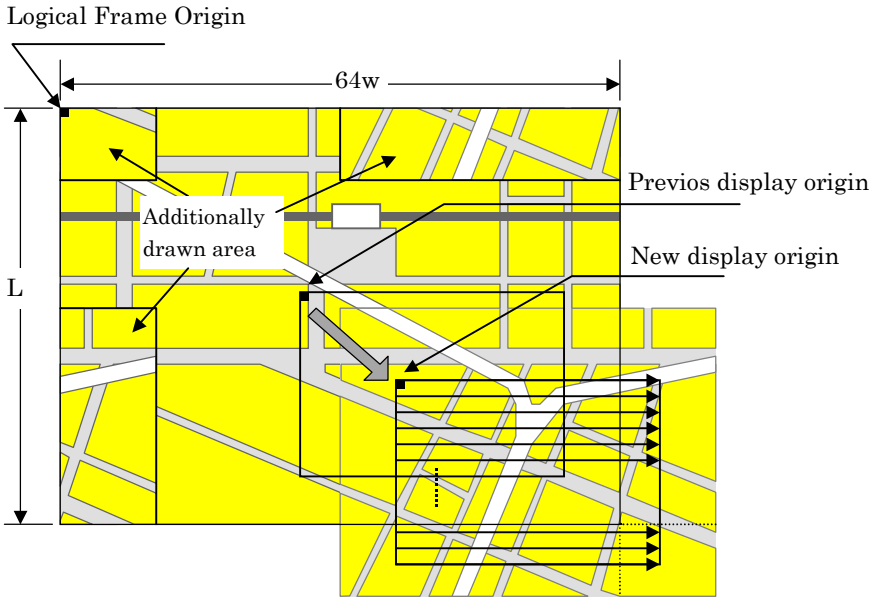


Fig. 7.5 Display Wraparound Processing

The relational expression between x,y coordinates in the frame and the corresponding linear address (in bytes) is as follows:

$A(x, y) = x \times bpp/8 + 64wy \quad (bpp = 8,16,32)$

The display coordinates origin must be present in the frame. Specifically, parameters has the following setting constraint:

$0 \leq DX < w \times 64 \times 8/bpp \ (bpp = 8,16,32)$

$0 \leq DY < H$

(DX,DY) and DA must indicate the same point in the frame. That is, the following relation must be established:

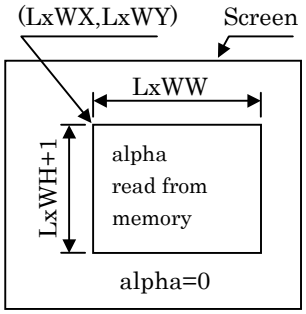
$DA = OA + DX \times bpp/8 + 64w \times DY \quad (bpp = 8,16,32)$

Note: L1 layer and LA0 to LA3 layers has no wraparound function.

L5 layer and LA0 to LA3 layers can be used as alpha coefficients for performing display blend. The alpha value corresponds to a layer at an absolute position in the display screen, and is calculated on a pixel by pixel basis.

When a window matching the display screen is defined, the alpha value read from memory is applied to the pixels in all areas of the display screen.

When a window smaller than the display screen is defined, the alpha value read from memory is applied to the inside the window, and alpha value 0 is applied to the outside the window.



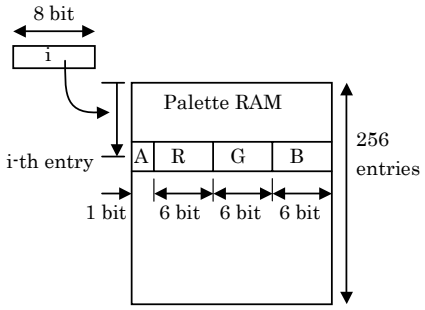
Since L5 layer is also used as display data, it has the wraparound function, but alpha dedicated layers LA0 to LA3 has no wraparound function.

7.3 Data Format

Data processed by the display controller is of the following formats:

7.3.1 Indirect Color (8 bits/pixel)

Palette RAM's index. The index is displayed after being converted by Palette RAM to the image data where each of RGB (R, G and B) is 6 bits. The palette that can be used depends on the layer.



When the pixel value is i, the RGB output value is determined by the i-th entry of the pallet.

The precision of each color element of the palette is 6 bits. The basic precision of display output is 8 bits for each of RGB, and each color element of the palette is output to be displayed with 2 bits shifted toward the MSB side.

7.3.2 Direct Color (16 bits/pixel)

The level of each of RGB is expressed by 5 bits. The basic precision of display output is 8 bits for each of RGB, and the value of each color element is output to be displayed with 3 bits shifted toward the MSB side.

There are the ARGB and RGBA formats.

format	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGB	A	R					G					B				
RGBA	R						G					B				A

The A bit determines whether display blend is enabled or disabled.

7.3.3 Direct Color (24 bits/pixel)

The level of each of RGB is expressed using 8 bits. In fact, 1 pixel is expressed by 32 bits.

There are the ARGB and RGBA formats.

format	31	30	…	25	24	23	22	…	17	16	15	14	…	9	8	7	6	…	1	0
ARGB	A					R					G					B				
RGBA	R					G					B					A				

In ARGB format, whether A = 0 or A ≠ 0 (whether blend enabled or disabled) is determined using a 1-bit value.

In ARGB format, whether A = 0 or A ≠ 0 (whether blend enabled or disabled) is determined using an 8-bit integer value.

Concerning the L5 layer: the “A” field of this format can be used as a blending factor. The following setup is required:

Blend factor dedication : L5 is invalidated in DLS setting (data is not displayed directly)

ARGB with blend : L5 is validated in DLS setting, L5ID=1, L5AF=11

RGBA with blend : L5 is validated in DLS setting, L5ID=1, L5AF=00

7.3.4 YCbCr Color (16 bits/pixel)

Image data of "YcbCr = 4:2:2" format. This image data is displayed after being converted by the calculation circuit to the image data where each of RGB is 8 bits. 2 pixels, each being RGB 24 bits, are expressed by 32 bits. As a result, the image data can be processed as 16 bits/pixel.

format	31	30	…	25	24	23	22	…	17	16	15	14	…	9	8	7	6	…	1	0
YCbCr	Y					Cr					Y					Cb				

7.3.5 Alpha Factor (8 bits/pixel)

The factor for display blend is held. When the value is t, t/256 is expressed as a binary fraction. During display blend, the following calculation is performed for each color element of each pixel:

$$c' = c0*t/256 + c1(1-t/256)$$

7.3.6 Alpha Factor (32 bits/pixel)

The factors for display blend are held. Four values packed into one pixel are applicable for four layers respectively.

format	31	30	…	25	24	23	22	…	17	16	15	14	…	9	8	7	6	…	1	0
Alpha	A3					A2					A1					A0				

One value is selected by LnAF parameter in each layer.

7.3.7 Layer Dependency

The display color that can be used for each layer is as shown below.

Layer	Compatible mode	Extended mode
L0	Direct (16,24), indirect (P0)	Direct (16,24), indirect (P0)
L1	Direct (16,24), indirect (P1), YCbCr	Direct (16,24), indirect (P1), YCbCr
L2	Direct (16,24), indirect (P1)	Direct (16,24), indirect (P2)
L3	Direct (16,24), indirect (P1)	Direct (16,24), indirect (P3)
L4	Direct (16,24), indirect (P1)	Direct (16,24)
L5	Direct (16,24), indirect (P1)	Direct (16,24), alpha
L6	Direct (16,24), indirect (P1)	Direct (16,24)
L7	Direct (16,24), indirect (P1)	Direct (16,24)
LA0	Alpha	
LA1	Alpha	
LA2	Alpha	
LA3	Alpha	

Pn above means the corresponding palette RAM. 4 palettes are used in the following manner:

Palette 0 (P0): This palette corresponds to the C layer palette for existing products. It is used an L0 palette. It can also be used as a cursor palette.

Palette 1 (P1): This palette corresponds to the MB layer palette for existing products. In compatible mode, it is used as a palette common to L1 to L5. In extended mode, it is used as an L1 dedicated palette.

Palette 2 (P2): This palette is an L2 dedicated palette. It can only be used in extended mode.

Palette 3 (P3): This palette is an L2 dedicated palette. It can only be used in extended mode.

7.4 Cursor

7.4.1 Cursor Display Function

Carmine can display 2 cursors. Their sizes are "64 × 64" pixels, and their cursor patterns are placed in graphics memory. The color used is an indirect color (8 bits/pixel), and the palette used is an L0 layer palette. However, transparency control (handling of transparent code and code "0") is independent of the L0 layer. No blend with the lower layer is performed.

7.4.2 Cursor Control

During cursor display, priority for the L0 layer (that is, whether to display the cursor above the L0 layer or below it) can be set. The priority can be set independently for the 2 cursors.

When a cursor goes beyond the screen, the part that goes beyond the screen is not displayed. Normally, cursor 0 is displayed in preference to cursor 1. However, when cursor 1 is displayed above the L0 layer and cursor 0 is displayed below the L0 layer, cursor 1 is displayed in preference to cursor 0.

7.5 Control of Display Scan

7.5.1 Supported Display

The typical resolution of a usable display and the frequency of the display's synchronization signal are shown. The pixel clock frequency is determined by setting the frequency division ratio of the display reference clock. The display reference clock is the built-in PLL (533 MHz at 33 MHz input) or is the clock given to the DCLKI input pin. The table below shows a setting example in a case where the built-in PLL is 533 MHz.

Table 7.5.1 Resolution and Display Frequency

Resolution	Frequency division ratio of reference clock	Pixel frequency	Horizontal total pixel count	Horizontal frequency	Vertical total raster count	Vertical frequency
320 × 240	1/84(*1)	6.35 MHz	403	15.75 kHz	263	59.9 Hz
400 × 240	1/63	8.47 MHz	538	15.73 kHz	263	59.8 Hz
480 × 240	1/53	10.1 MHz	638	15.76 kHz	263	60.0 Hz
640 × 480	1/21	25.4 MHz	800	31.4 kHz	525	60.5 Hz
854 × 480	1/16	33.3 MHz	1062	31.7 kHz	525	59.8 Hz
800 × 600	1/13	41.0 MHz	1060	38.7 kHz	628	61.6 Hz
1024 × 768	1/8	66.7 MHz	1380	48.3 kHz	806	59.9 Hz
1280 × 1024	1/5	106.7 MHz	1664	64.1 kHz	1066	60.1 Hz

(*1): LCS = 1

Pixel frequency = 33.3 MHz × 16 × Frequency division ratio of reference clock (when the built-in PLL is selected)

= Frequency of the DCLKI input pin x Frequency division ratio of reference clock (when DCLKI is selected)

Horizontal frequency = Pixel frequency/Horizontal total pixel count

Vertical frequency = Horizontal frequency/Vertical total raster count

7.5.2 Interlace Display

Carmine can perform interlace display in addition to non-interlace display.

When the DCM0 register's synchronous mode setting is interlace video (11), the odd rasters and even rasters of images in graphics memory are output alternately on a field by field basis, and 1 screen displayed consists of 1 frame (odd field + even field).

When the DCM0 register's synchronous mode setting is interlace (01), images in graphics memory are output in raster order. The same image data is output to an odd field and an even field. So, the raster count in the screen becomes half of that for interlace video. However, unlike non-interlace mode, odd fields and even fields are distinguished from each other owing to the phase relation between horizontal synchronization signal and vertical synchronization signal.

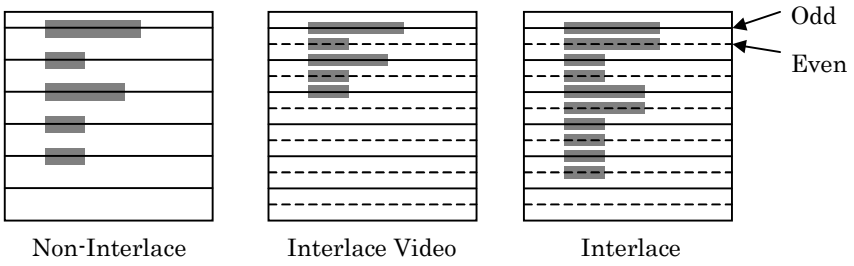
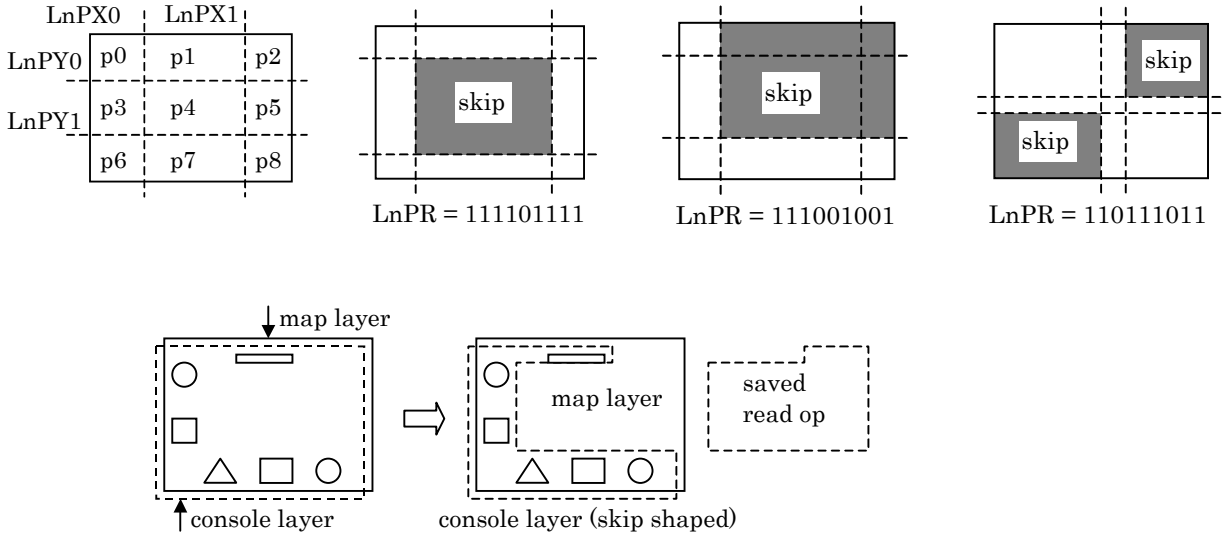


Fig. 7.6 Display Difference by Synchronous Mode Setting

7.6 Read Skip

Data read (= data reading) of an area hidden completely by the upper layer is inhibited, thus being able to lower the reading traffic for memory. Also, when a top-layer enabled image (such as operation menu) is localized in a specific area of the screen, extra data read can be inhibited.

It is possible to split the display area into 9 partitions by 2 horizontal straight lines and 2 vertical straight lines, and to define whether or not to read each of these partitions. Areas not to be read are subjected to transparent processing.



Whether to enable or disable each partition is set for the LnPR (Ln Partition Read, 9 bits) field. The LnPR field's each bit corresponds to each partition. The partition corresponding to the set value 0 is not read.

Note: L1 layer is not relevant.

7.7 External Synchronization

Display scan can be performed in synchronization with horizontal/vertical synchronization signals coming from outside.

When external synchronous mode is selected by a register, Carmine samples HSYNC signals, to perform display in synchronization with external video signals. The built-in PLL clock or the DCLKI signal input can be selected as the sampling clock. Also, superimposition can be performed using chroma-key processing. A sample circuit for performing external synchronization is shown below.

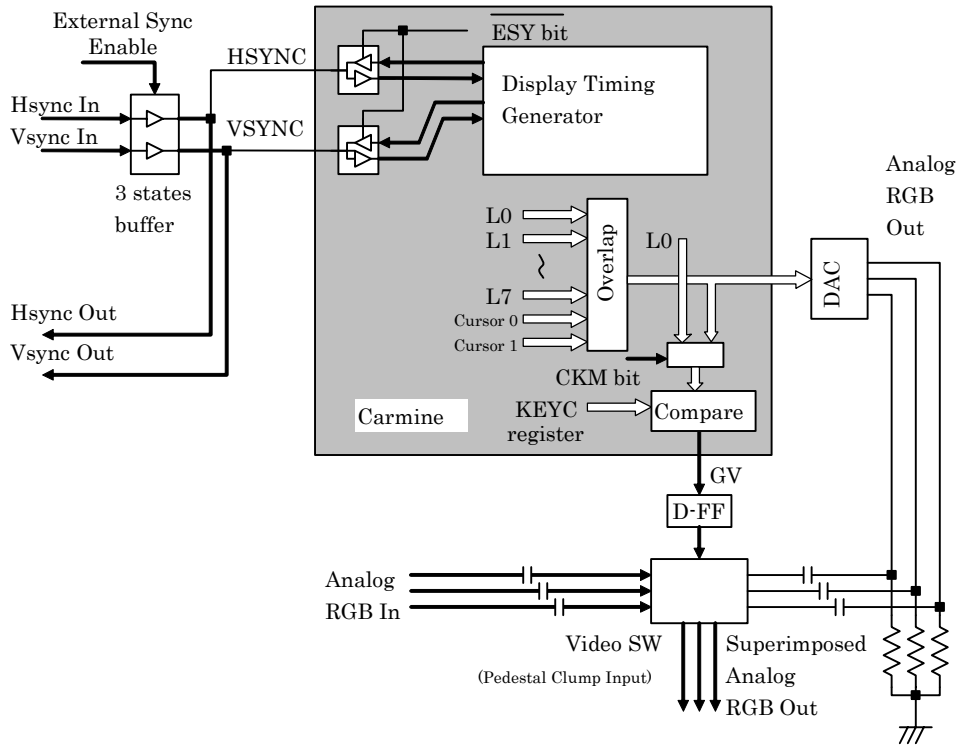


Fig. 7.7 Example of Use of External Synchronization

External synchronous mode is set by the DCM0 (1) register's ESY bit. When the mode is set, Carmine's HSYNC and VSYNC pins go into input mode. After this, supply external synchronization signals via the 3-state buffer. When exiting external synchronous mode, turn off external synchronizing input and then set the ESY bit inside Carmine to OFF.

Turning ON the buffer for external synchronization signals with Carmine's synchronization signal output turned ON must be avoided. Perform control so that no period where they are concurrently ON exists, following the above procedure.

When performing external synchronization by using a display clock based on the built-in PLL, Carmine extends the clock cycle immediately after a horizontal synchronizing pulse is input, to be in phase with the horizontal synchronization signal. At this point, a caution is needed. That is, when a high-speed serial transfer transmitter such as LVDS is previously connected to the digital RGB output, the phase matching makes PLL incorporated in the transmitter unstable temporarily; and so do not perform external synchronization based on the built-in PLL with a high-speed serial transfer transmitter connected to the digital RGB output.

Horizontal synchronization is controlled by the following state transitions:

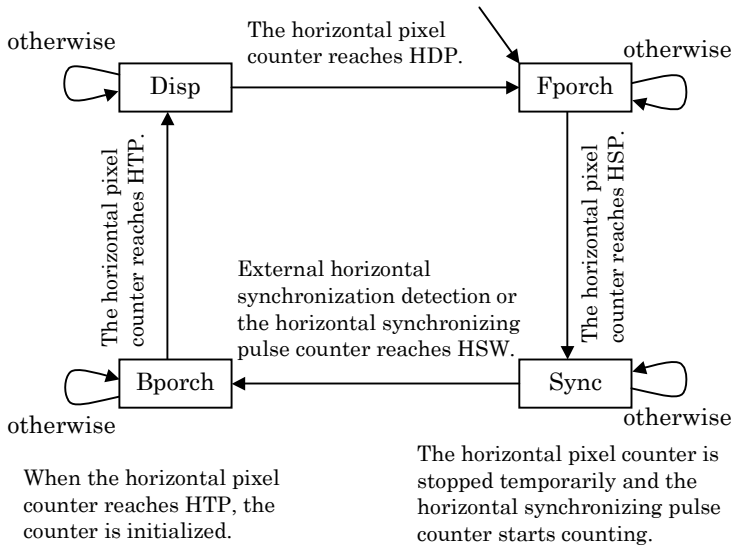


Fig. 7.8 State Transition Diagram for Horizontal Synchronization

State transition is mainly controlled by the horizontal pixel counter's count value. The display period corresponds to the Disp state. When the horizontal pixel counter's count value reaches the HDP register's set value, the display period ends, making a transition from the Disp state to the Fporch (front porch) state. When, in the Fporch state, the horizontal pixel counter's count value reaches the HSP register's set value, the Sync state occurs. In this state, an external horizontal synchronization signal is waited for. Carmine detects and synchronizes the negation edge of the external horizontal synchronizing pulse. When the external horizontal synchronization signal is detected, a transition is made to the Bporch (back porch) state. In the Sync state, the horizontal pixel counter stops. Instead, the horizontal synchronizing pulse counter performs counting starting with 0. When this count value reaches the HSW register's set value, a transition is made to the Bporch state, not detecting the external horizontal synchronization signal. When, in the Bporch state, the horizontal pixel counter's value reaches the HTP register's set value, the counter is reset, and at the same time, a transition is made to the Disp state, starting displaying the next raster.

Vertical synchronization is controlled by the following state transitions:

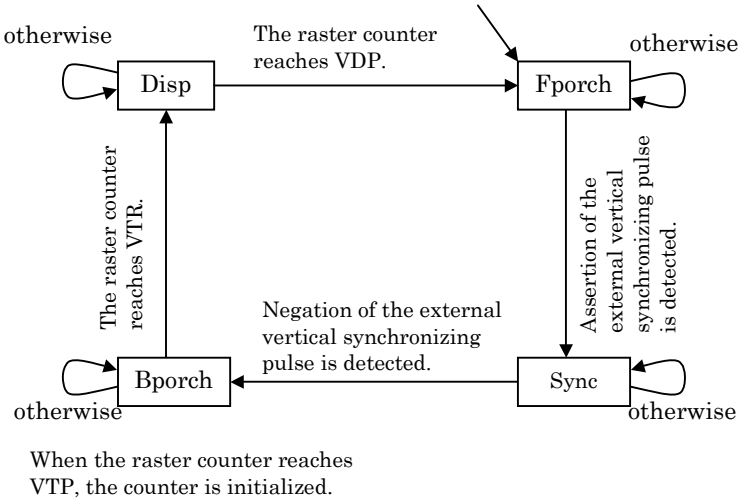


Fig. 7.9 State Transition Diagram for Vertical Synchronization

For vertical control, the raster counter's count value is the basis for state transition. The display period corresponds to the Disp state. When the raster counter's value reaches the VDP register's set value, the display period ends, making a transition from the Disp state to the Fporch (front porch) state. In the Fporch state, assertion of an external vertical synchronizing pulse is waited for. When the assertion is detected, a transition is made to the Sync state. In this state, negation of the external vertical synchronization signal is waited for. When the negation is detected, a transition is made to the Bporch (back porch) state. In the Bporch state, when the raster counter's value reaches the VTR register's set value, the counter is reset, and at the same time, a transition is made to the Disp state, starting displaying the next field.

7.8 Variable Parameters Used for Conversion from YCbCr to RGB for L1 Layer

For L1 layer, data of YCbCr format can be displayed after being converted to RGB, and conversion parameters are variable.

YCbCr data is converted using the following expressions:

$R = a_{11}*Y + a_{12}*(Cb - 128) + a_{13}*(Cr - 128) + b_1$

$G = a_{21}*Y + a_{22}*(Cb - 128) + a_{23}*(Cr - 128) + b_2$

$B = a_{31}*Y + a_{32}*(Cb - 128) + a_{33}*(Cr - 128) + b_3$

a_{ij} ---- 11 bit signed real (lower 8 bit is fraction, two's complement)

b_i ----- 9 bit signed integer (two's complement)

These expressions can be represented as follows as matrix calculation:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \mathbf{A} \begin{pmatrix} Y \\ Cb\text{-}128 \\ Cr\text{-}128 \end{pmatrix} + \boldsymbol{b} \qquad \text{where } \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

These parameters are set in the following registers:

L1YCR0 (a_{12}, a_{11}), L1YCR1 (b_1, a_{13})

L1YCG0 (a_{22}, a_{21}), L1YCG1 (b_2, a_{23})

L1YCB0 (a_{32}, a_{31}), L1YCB1 (b_3, a_{33})

The same conversion as existing products is performed using the initial value that is set in the registers immediately after reset. The registers' values immediately after reset are as follows:

a_{11} = 0x12b (299/256), a_{12} = 0x0, a_{13} = 0x198 (408/256)

a_{21} = 0x12b (299/256), a_{22} = 0x79c (-100/256), a_{23} = 0x72f (-209/256)

a_{31} = 0x12b (299/256), a_{32} = 0x204 (516/256), a_{33} = 0x0

$b_1 = b_2 = b_3$ = 0x1f0 (-16)

Luma, contrast, chroma, and color saturation degree can be controlled by changing these conversion parameters.

Adding a constant to $\boldsymbol{b}$ means an increase of luma.

Multiplying $\mathbf{A}$ by a scalar constant greater than "1" means an increase of contrast.

2-dimensional rotation of Cb-128 and Cr-128 means a change of chroma.

Color saturation degree is the relative color strength compared to luma.

New conversion coefficients in which these changes are reflected are as shown in the following expression:

$$\mathbf{A} = c_1 \mathbf{A}_0 \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t) & \sin(t) \\ 0 & -\sin(t) & \cos(t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_2 \end{pmatrix} = \mathbf{A}_0 \begin{pmatrix} c1 & 0 & 0 \\ 0 & \cos(t)c_1c_2 & \sin(t)c_1c_2 \\ 0 & -\sin(t)c_1c_2 & \cos(t)c_1c_2 \end{pmatrix}$$

$$\boldsymbol{b} = \boldsymbol{b}_0 + \begin{pmatrix} c_3 \\ c_3 \\ c_3 \end{pmatrix}$$

$\mathbf{A}_0$, $\mathbf{b}_0$: Initial value

c_1: Contrast parameter. "1" is the standard value. "1.2", for example, means that the contrast is little stronger than the standard contrast.

c_2: Color saturation degree parameter. "1" is the standard value, and "0" indicates monochrome image.

c_3: Luma parameter. "0" is the standard value.

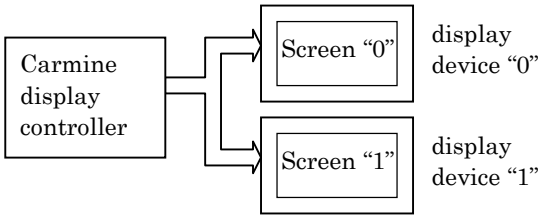
t: Chroma rotation parameter. "0" degree is the standard value.

Note: When a_{ij} and b_i are calculated anew, clip and set them in the effective value range of the relevant register.

7.9 Dual Display by Single Display Controller

7.9.1 Overview

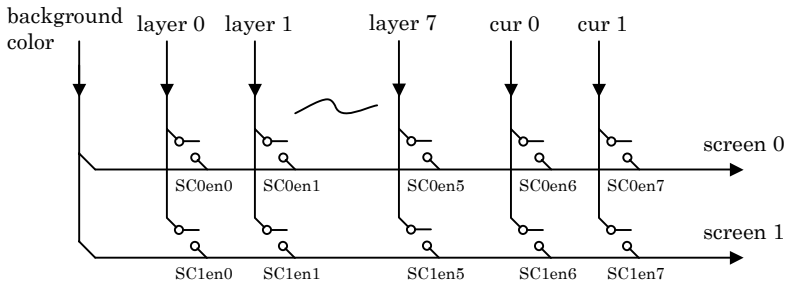
A single display controller has the Coral-PA-compatible dual display function. This function can output 2 multiplexed screens, to display them on 2 display devices. Carmine has 2 display controllers, and so using this function, up to 4 independent screens can be displayed. It can be controlled which layer is output to which screen. It is assumed here that the screen 0 is output to the display device 0 and the screen 1 is output to the display device 1.



7.9.2 Layer Destination Control

It is possible that arbitrary layers or cursors are included in both screens or in one screen. Layers not included in the screens are treated as transparent. When all outputs are OFF, the background color is displayed.

The destination control can be considered as the following virtual crosspoint switches.



The MDen (multi display enable) bit is in the MDC (multi display control) register, enabling the dual display operation.

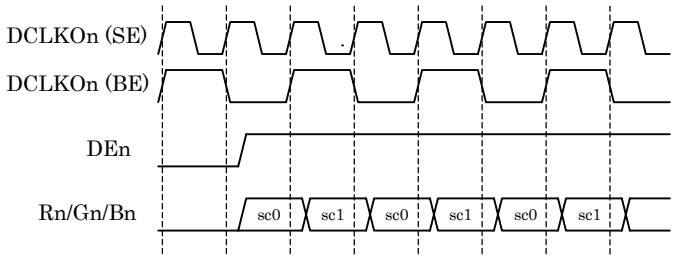
The SC0en (screen "0" enable) field is in the MDC register, specifying that the layer or cursor be output to which screen 0.

The SC1en (screen "1" enable) field is in the MDC register, specifying that the layer or cursor be output to which screen 1.

bit-0 ---- L0 is included.
bit-1 ---- L1 is included.
:
bit-5 ---- L5 is included.
bit-6 ---- Cursor0 is included.
bit-7 ---- Cursor1 is included.

7.9.3 Output Signal Control

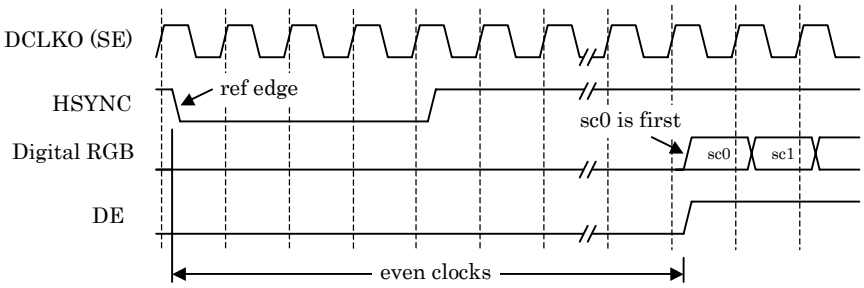
There are 2 output modes: one is parallel mode and the other is multiplexing mode; however, Carmine uses only multiplexing mode. 2 screens are multiplexed and output to the RGB output.



There are two clock output modes.

In BE (bi-edge) DCLKO mode, 2 output phases can be identified by both edges.

In SE (single-edge) DCLKO mode, 2 output phases are identified by HSYNCn or DEn.

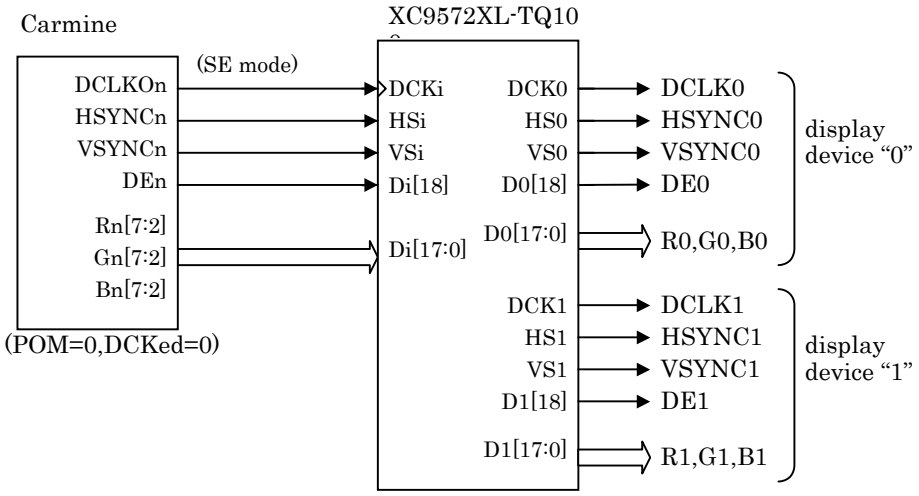


The CKed (clock edge) bit is in the DCM3 register, selecting between the 2 DCLKO modes. Cked = 0 indicates that SE (single-edge) DCLKO mode is established; Cked = 1 indicates that BE (bi-edge) DCLKO mode is established.

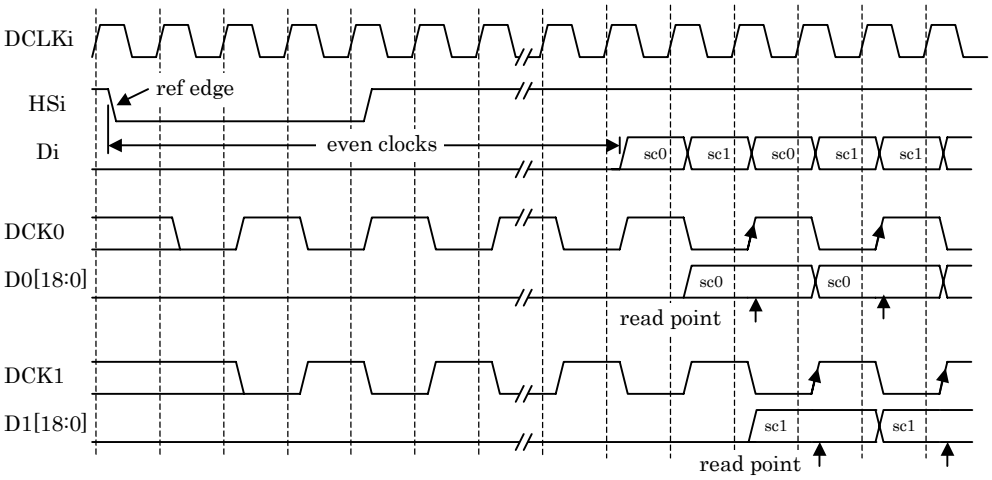
7.9.4 Sample Output Circuit

(1) SE mode

This example shows that screen data in low-cost CPLD is separated using SE mode's DCLKOn clock and DEn output.

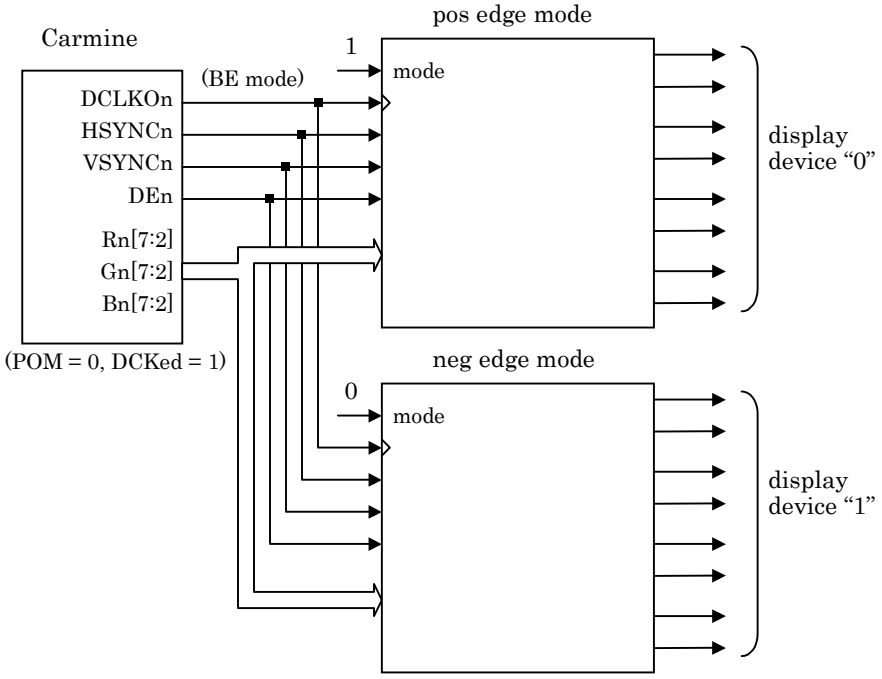



```
module XC9572XL ( DCKi, HSi,VSi,Di,    DCK0,HS0,VS0,D0,    DCK1,HS1,VS1,D1 );
        input DCKi,HSi,VSi;
        input[18:0] Di;
        output DCK0,HS0,VS0, DCK1,HS1,VS1;
        output[18:0] D0,D1;
        reg HS0,HS1, VS0,VS1, DCK0,DCK1;
        reg[18:0] D0,D1;
        always @(posedge DCKi) begin
                HS0 <= HSi; HS1 <= HS0;
                VS0 <= VSi; VS1 <= VS0;
                DCK0 <= (HS0&!HSi)? 0: !DCK0; // sync to ref edge : flip
                DCK1 <= DCK0;
                if(DCK0) D0 <= Di;
                if(DCK1) D1 <= Di;
        end
endmodule
```



(2) BE mode

When a device can select between positive edge and negative edge as an active edge of the clock, the device can receive data in 2 screens separately through the DCLKO clock of BE mode.



7.9.5 Display Clock and Timing

When performing dual display, twice the frequency for performing normal display must be supplied. VGA display needs 25 MHz display clock, but dual display needs 50 MHz display clock. For both displays, timing setting such as HTP except SC (scaling ratio) parameter is the same. The maximum display clock frequency determines the maximum usable resolution. SVGA (800 × 600) resolution can be used for performing dual display, and in this case, the display clock is 80 MHz.

Note: When supplying 80 MHz from the built-in PLL, the PLL's oscillation frequency must be reviewed.

7.9.6 Restrictions

(1) The scan speed and resolution of the two display devices must be the same under the synchronization signals common to the display devices.

(2) Dual display cannot be performed concurrently with external synchronization.

(3) The BE mode cannot be used when using the DCLKI clock input as the display clock.

7.10 Writeback

7.10.1 Overview

A display output image where multiple layers are superimposed can be written to graphics memory as 24 bits/pixel image data. When parameter setting described later is performed and then the WBEN bit of the WBC (Writeback control) register is set to 1, the display output image equivalent to 1 screen is captured and written to memory.

Writeback operation is performed concurrently with display operation. When performing writeback operation, attention must be paid to the bandwidth of graphics memory.

7.10.2 Source Selection

When performing dual display, it is selected which display is used as the source. The selection is performed using the WBM (Writeback Model) register's VSEL (video select) field.

Vsel	Single display	Coral-PA dual display
00	Display section 0	Display section 0 / Screen 0
10	Unused	Display section 0 / Screen 1
01	Display section 1	Display section 1 / Screen 0
11	Unused	Display section 1 / Screen 1

7.10.3 Image Area Definition

(1) Write starting address

WBOA0: Non-interlace display output is written. During interlace display, odd fields are written.

WBOA1: Even fields for interlace display output are written.

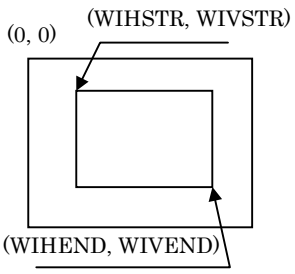
(2) Stride

WBW: Specify stride in units of 64 bytes. A value greater than the data amount equivalent to 1 line of the display screen must be set.

(3) Clip frame

A clip frame can be specified for the following register, and only the area inside the clip frame is subjected to writeback operation.

WIHSTR, WIVSTR, WIHEND, WIVEND



7.10.4 Data Format

Four data formats are available.

C24	RGBA	Data format
0	0	ARGB 16bit/pixel
0	1	RGBA 16bit/pixel
1	0	ARGB 24bit/pixel
1	1	RGBA 24bit/pixel

Also, it can be selected using the BLEN bit whether each pixel's blend bit is set to 1 or 0.

BLEN = 0: Blend bit = 0 (for ARGB, MSB; for RGBA, LSB)
BLEN = 1: Blend bit = 1 (for ARGB, MSB; for RGBA, LSB)

7.10.5 Field Selection

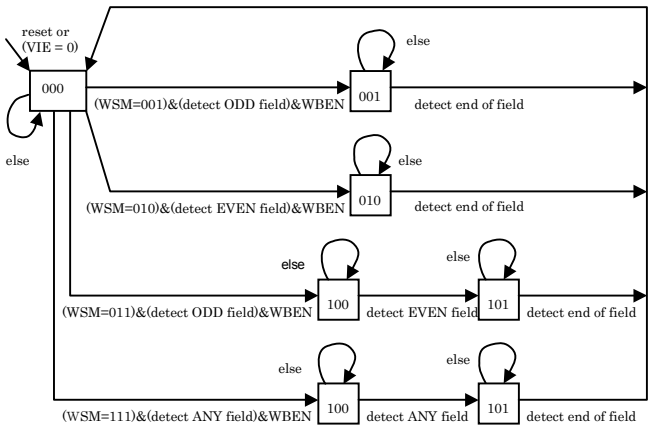
Writeback for non-interlace display output simply captures 1 frame. For interlace display output, fields can be selected according to the specified mode.

001: Odd field mode (non-interlace)
010: Even field mode
011: Both fields mode
111: Both fields mode (no field discrimination is performed)

Field selection is performed using the WBM register's FM (Field Mode) field.

7.10.6 State Transition

The following state transitions occur.



000: Initial state
001: Odd field mode / Writeback in progress
010: Even field mode / Writeback in progress
100: Both fields mode / Writeback in progress for first field

These states can be read using the WBM register's ST (Status) field.

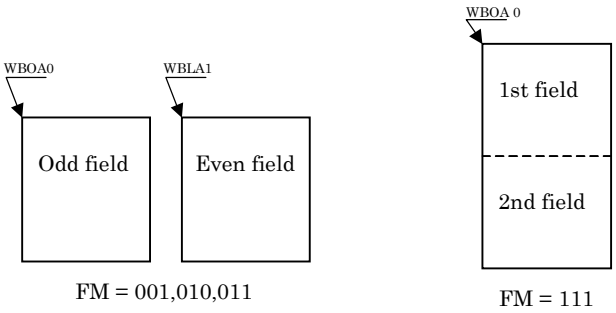
When 1 writeback operation ends, the WBE flag is automatically cleared, causing the stopped state.

7.10.7 Transfer Destination Selection

The starting address of the transfer destination of an image is as shown below.

WBOA0: Odd field
WBOA1: Even field

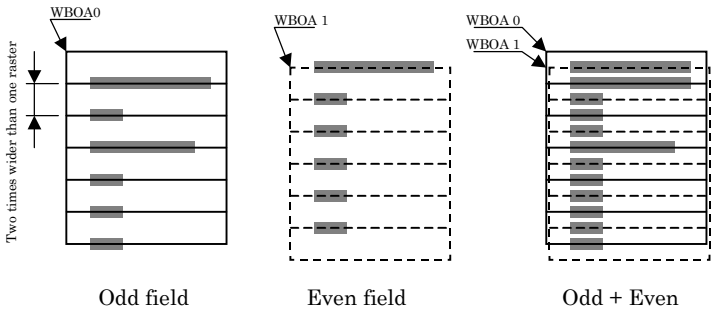
However, when FM = 111, odd fields and even fields are not distinguished from each other, and first field is written to the area beginning at WBOA0 and second field is written after the first field.



To create an image where even fields are captured after odd fields and they are arranged in graphics memory in non-interlace (progressive) scan order, perform the following setting (FM = 011).

(1 The starting address of even field is increased for the amount of 1 raster compared to the starting address of odd field.

(2) The stride is set to twice 1 raster.



When spacing for the amount of 1 raster is kept between the above starting addresses and data is transferred with this spacing kept between the succeeding addresses for even fields and the succeeding addresses for odd fields, even fields and odd fields are synthesized.

7.11 Simultaneous layer display restrictions

The number of layers that can be simultaneously displayed is limited. If the limit is exceeded, improper data is used for the display on screen, which is visible as noise or other errors. The number of layers that can be simultaneously displayed depends on the conditions listed below.

1) Display pixel clock and resolution

Generally speaking, a higher display pixel clock is required for higher display resolutions. This correspondingly means that more data must be read from external SDRAM within a specific time period. This means that memory bandwidth plays an increasingly important role for higher resolutions.

2) Pixel size of layer (bit/pixel)

Display errors are less likely to cause memory bandwidth problems with less bits per pixel.

3) Data supply capability of SDRAM

The bandwidth capabilities of external SDRAM are critical. Components that have higher clock and faster timing conditions for RAS and CAS have higher bandwidth capabilities and this means that the number of layers that can be simultaneously displayed increases accordingly.

4) Layer Allocation

Carmine's display data buffers are structured and managed in pairs (1 odd number layer and 1 even number layer) as shown below:

L0:L1, L2:L3, ... , L6:L7

If only one layer of a pair is used, it can use a data buffer with double capacity. As a result, display failure occurs less often. If 32bit/pixel la y er is used in SVGA or higher resolution, it is recommended that one layer is left unused in a pair.

Numbers of layers displayed simultaneously for representative conditions are shown in next page. Common conditions are as follow.

- Each layers are 32bit/pixel primally except that L1 is 16bit/pixel if capture is included.
- Draw operation works at the same time.
- "8+4" in table means 8 layers and 4 layers are displayed by disp0 and disp1, respectively.
- Only one layer is used in a pair if less than 5 layers are displayed
- "656+656" in capture column means dual capture operation.
- SDRAM is clocked at 133MHz and following timing parameters are used.

  fast ------ DRAM_CTRL_SET_TIME1 = 0x2628

  std ------- DRAM_CTRL_SET_TIME1 = 0x4749

DRAM speed	Resolution	pixel clk [MHz]	Capture	Displayed Layers (32bit/pixel)
fast	640x480	530/21	no	8+6
	800x480	530/16	no	8+3
	800x600	530/13	no	8+1, 5+4
	1024x768	530/8	no	4+3
	1280x1024	530/5	no	3+0
std	640x480	530/21	no	8+2
	800x480	530/16	no	8+0, 4+4
	800x600	530/13	no	6+0, 4+3
	1024x768	530/8	no	4+0
	1280x1024	530/5	no	2+0
fast	640x480	530/21	656	8+6
	800x480	530/16	656	8+3
	800x600	530/13	656	8+1, 5+4
	1024x768	530/8	656	4+3
	1280x1024	530/5	656	3+0
std	640x480	530/21	656	8+2
	800x480	530/16	656	8+0, 4+4
	800x600	530/13	656	6+0, 4+3
	1024x768	530/8	656	4+0
	1280x1024	530/5	656	2+0
fast	640x480	530/21	656+656	8+5
	800x480	530/16	656+656	8+2, 5+5
	800x600	530/13	656+656	7+1, 4+4
	1024x768	530/8	656+656	4+2
std	640x480	530/21	656+656	7+1, 4+4
	800x480	530/16	656+656	6+1, 4+4
	800x600	530/13	656+656	4+1
	1024x768	530/8	656+656	3+1

7.12 Register

7.12.1 List of Registers

Newly added registers are shown shaded.

There is a possibility that their addresses and bit fields may be changed.

Base = VdisCapBase(= WbackBase + 0x1ff8)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VCCC (Video/Capture Common Control)																															
																			C1sel	C0sel								WBsr	C1sr	C0sr	V1sr	V0sr
1	VCSR(Video/Capture Software reset)																															

Base = WbackBase

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000	WBC (Writeback Control)																															
	WBE																															
004	WBM (Writeback Mode)																															
									WBW (stride)								BLEN	C24	RGBA				ST		FM				VSEL			
008	WBOA0 (Writeback Origine Address 0)																															
00C	WBOA1 (Writeback Origine Address1)																															
010	WIVSTR																				WIHSTR											
014	WIVEND																				WIVEND											

Base = DisplayBase0 or DisplayBase1

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000	DCM0 (Display Control Mode 0)																															
	DEN										L7E	L6E	L45E	L23E	L1E	L0E	CKS	LCS		SC					EEQ	EGV	FF	Fdet	SF	ESY	SYNC	
100	DCM1 (Display Control Mode 1)																															
	DEN				LA3E	LA2E	LA1E	LA0E	L7E	L6E	L5E	L4E	L3E	L2E	L1E	L0E	CKS	LCS	SC						EEQ	EGV	FF	Fdet	SF	ESY	SYNC	
104	DCM2 (Display Control Mode 2)																															
																												RUM1			RUF	RUM0
108	DCM3 (Display Control Mode 3)																															
																							CKddr	CKinv	CKDe			CKDn				
004	HTP (H Total Pixels)																															
008	HDB (H Display Boundary)																				HDP (H Display Period)											
00C	VSWH		VSW						HSW												HSP (H Sync pulse Position)											
010	VTR (V Total Rasters)																															
014	VDP (V Display Period)																				VSP (V Sync pulse Position)											
018	WY (Window Y)																				WX (Window X)											
01C	WH (Window Height)																				WW (Window Width)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
020	L0M (L0 Mode)																															
	L0C				L0W (L0 width)																L0H (L0 Height)											
024	L0OA (L0 Origin Address)																															
028	L0DA (L0 Display Address)																															
02C	L0D Y (L0 Display Y)																				L0DX (L0 Display X)											
110	L0EM (L0 Extend Mode)																															
	L0EC								L0PB																							L0WP
114	L0WY (L0 Window Y)																				L0WX (L0 Window X)											
118	L0WH (L0 Window Height)																				L0WW (L0 Window Width)											
1880																	L0SE							L0PR							
1884	L0PX1 (L0 Partition X1)																				L0PX0 (L0 Partition X0)											
1888	L0PY1 (L0 Partition Y1)																				L0PY0 (L0 Partition Y0)											
030	L1M (L1 Mode)																															
	L1C	L1YC	L1CS	L1IM					L1S (L1 Stride)																							
034	L1OA0 (L1 Origin Address 0) / CBDA0 (Capure Buffer Display Address 0)																															
038	CBDA1 (Capture Buffer Display Address 1)																															
120	L1EM (L1 Extend Mode)																															
	L1EC			L1CP			VMAG		L1PB																							
124	L1WY (L1 Window Y)																				L1WX (L1 Window X)											
128	L1WH (L1 Window Height)																				L1WW (L1 Window Width)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
040	L2M (L2 Mode)																															
	L2C	L2FLP							L2S (L2 Stride)												L0H (L0 Height)											
044	L2OA0 (L2 Origin Address 0)																															
048	L2DA0 (L2 Display Address 0)																															
04C	L2OA1 (L2 Origin Address 1)																															
050	L2DA1 (L2 Display Address 1)																															
054	L2DY (L2 Display Y)																				L2DX (L2 Display X)											
130	L2EM (L2 Extend Mode)																															
	L2EC								L2PB																						L2OM	L2WP
134	L2WY (L2 Window Y)																				L2WX (L2 Window X)											
138	L2WH (L2 Window Height)																				L2WW (L2 Window Width)											
18A0																	L2SE							L2PR							
18A4	L2PX1 (L2 Partition X1)																				L2PX0 (L2 Partition X0)											
18A8	L2PY1 (L2 Partition Y1)																				L2PY0 (L2 Partition Y0)											
058	L3M (L3 Mode)																															
	L3C	L3FLP							L3S (L3 Stride)												L3H (L3 Height)											
05C	L3OA0 (L3 Origin Address 0)																															
060	L3DA0 (L3 Display Address 0)																															
064	L3OA1 (L3 Origin Address 1)																															
068	L3DA1 (L3 Display Address 1)																															
06C	L3DY (L3 Display Y)																				L3DX (L3 Display X)											
140	L3EM (L3 Extend Mode)																															
	L3EC								L3PB																						L3OM	L3WP
144	L3WY (L3 Window Y)																				L3WX (L3 Window X)											
148	L3WH (L3 Window Height)																				L3WW (L3 Window Width)											
18B0																	L3SE							L3PR							
18B4	L3PX1 (L3 Partition X1)																				L3PX0 (L3 Partition X0)											
18B8	L3PY1 (L3 Partition Y1)																				L3PY0 (L3 Partition Y0)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
070	L4M (L4 Mode)																															
	L4C	L4FLP							L4S (L4 Stride)												L4H (L4 Height)											
074	L4OA0 (L4 Origin Address 0)																															
078	L4DA0 (L4 Display Address 0)																															
07C	L4OA1 (L4 Origin Address 1)																															
080	L4DA1 (L4 Display Address 1)																															
084	L4DY (L4 Display Y)																				L4DX (L4 Display X)											
150	L4EM (L4 Extend Mode)																															
	L4EC																														L4OM	L4WP
154	L4WY (L4 Window Y)																				L4WX (L4 Window X)											
158	L4WH (L4 Window Height)																				L4WW (L4 Window Width)											
18C0																	L4SE								L4PR							
18C4	L4PX1 (L4 Partition X1)																				L4PX0 (L4 Partition X0)											
18C8	L4PY1 (L4 Partition Y1)																				L4PY0 (L4 Partition Y0)											
088	L5M (L5 Mode)																															
	L5C	L5FLP							L5S (L5 Stride)												L5H (L5 Height)											
08C	L5OA0 (L5 Origin Address 0)																															
090	L5DA0 (L5 Display Address 0)																															
094	L5OA1 (L5 Origin Address 1)																															
098	L5DA1 (L5 Display Address 1)																															
09C	L5DY (L5 Display Y)																				L5X (L5 Display X)											
160	L5EM (L5 Extend Mode)																															
	L5EC																														L5OM	L5WP
164	L5WY (L5 Window Y)																				L5WX (L5 Window X)											
168	L5WH (L5 Window Height)																				L5WW (L5 Window Width)											
18D0																	L5E								L5PR							
18D4	L5PX1 (L5 Partition X1)																				L5PX0 (L5 Partition X0)											
18D8	L5PY1 (L5 Partition Y1)																				L5PY0 (L5 Partition Y0)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1900	L6M (L6 Mode)																															
	L6C	L6FLP							L6S (L6 Stride)												L6H (L6 Height)											
1904	L6OA0 (L6 Origin Address 0)																															
1908	L6DA0 (L6 Display Address 0)																															
190c	L6OA1 (L6 Origin Address 1)																															
1910	L6DA1 (L6 Display Address 1)																															
1914	L6DY (L6 Display Y)																				L6X (L6 Display X)											
1918	L6EM (L6 Extend Mode)																															
	L6EC																														L6OM	L6WP
191C	L6WY (L6 Window Y)																				L6WX (L6 Window X)											
1920	L6WH (L6 Window Height)																				L6WW (L6 Window Width)											
1924																	L6SE							L6PR							
1928	L6PX1 (L6 Partition X1)																				L6PX0 (L6 Partition X0)											
192C	L6PY1 (L6 Partition Y1)																				L6PY0 (L6 Partition Y0)											
1940	L7M (L7 Mode)																															
	L7C	L7FLP							L7S (L7 Stride)												L7H (L7 Height)											
1944	L7OA0 (L7 Origin Address 0)																															
1948	L7DA0 (L7 Display Address 0)																															
194C	L7OA1 (L7 Origin Address 1)																															
1950	L7DA1 (L7 Display Address 1)																															
1954	L7DY (L7 Display Y)																				L7X (L7 Display X)											
1958	L7EM (L7 Extend Mode)																															
	L7EC																														L7OM	L7WP
195C	L7WY (L7 Window Y)																				L7WX (L7 Window X)											
1960	L7WH (L7 Window Hieght)																				L7WW (L7 Window Windth)											
1964																	L7SE							L7PR							
1968	L7PX1 (L7 Partition X1)																				L7PX0 (L7 Partition X0)											
196C	L7PY1 (L7 Partition Y1)																				L7PY0 (L7 Partition Y0)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1A00	LA0M (LA0 Mode)																															
									LA0S (L0 Stride)																							
1A04	LA0DA (LA0 Display Address)																															
1A08	LA0WY (LA0 Window Y)																LA0WX (LA0 Window X)															
1A0C	LA0WH (LA0 Window Height)																LA0WW (LA0 Window Width)															
1A10	LA1M (LA1 Mode)																															
									LA1S (L0 Stride)																							
1A14	LA1DA (LA1 Display Address)																															
1A18	LA1WY (LA1 Window Y)																LA1WX (LA1 Window X)															
1A1C	LA1WH (LA1 Window Height)																LA1WW (LA1 Window Width)															
1A20	LA2M (LA2 Mode)																															
									LA2S (L0 Stride)																							
1A24	LA2DA (LA2 Display Address)																															
1A28	LA2WY (LA2 Window Y)																LA2WX (LA2 Window X)															
1A2C	LA2WH (LA2 Window Height)																LA2WW (LA2 Window Width)															
1A30	LA3M (LA3 Mode)																															
									LA3S (L0 Stride)																							
1A34	LA3DA (LA3 Display Address)																															
1A38	LA3WY (LA0 Window Y)																LA3WX (LA3 Window X)															
1A3C	LA3WH (LA0 Window Height)																LA3WW (LA3 Window Width)															

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0A0					CSIZE				CPM								CUTC (Cursor Transparent Control)															
					CSIZ1		CSIZ0				CUE1	CUE0			CUO1	CUO0								CUZT	CUTC							
0A4					CUOA0 (CUrsor0 Origin Address)																											
0A8					CUY0 (Cursor0 Position Y)																CUX0 (Cursor0 Position X)											
0AC					CUOA1 (CUrsor1 Origin Address)																											
0B0					CUY1 (Cursor1 Position Y)																CUX1 (Cursor1 Position X)											
170	MDC (Multi Display Control)																															
	MDEN								SC1xEN				SC0xEN				SC1EN								SC0EN							
180	DLS (Display Layer Select)																															
	DLS7				DLS6				DLS5				DLS4				DLS3				DLS2				DLS1				DLS0			
184	DBGC (Display Back Ground Color)																															
0B4	L0BLD (L0 Blend)																															
															L0BE	L0BS	L0BI	L0BP	L0ID	L0AF	L0AS		L0BR								
188	L1BLD (L1 Blend)																															
															L1BE	L1BS	L1BI	L1BP	L1ID	L1AF	L1AS		L1BR								
18C	L2BLD (L2 Blend)																															
															L2BE	L2BS	L2BI	L2BP	L2ID	L2AF	L2AS		L2BR								
190	L3BLD (L3 Blend)																															
															L3BE	L3BS	L3BI	L3BP	L3ID	L3AF	L3AS		L3BR								
194	L4BLD (L4 Blend)																															
															L4BE	L4BS	L4BI	L4BP	L4ID	L4AF	L4AS		L4BR								
198	L5BLD (L5 Blend)																															
															L5BE	L5BS	L5BI	L5BP	L5ID	L5AF	L5AS		L5BR								
1990	L6BLD (L6 Blend)																															
															L6BE	L6BS	L6BI	L6BP	L6ID	L6AF	L6AS		L6BR								
1994	L7BLD (L7 Blend)																															
															L7BE	L7BS	L7BI	L7BP	L7ID	L7AF	L7AS		L7BR								

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0BC	L0TC (L0 Transparent Control)																															
																	L0ZT	L0TC (L0 Transparent Color)															
0C0	L2TR (L2 Transparent Control)																L3TR (L3 Transparent Control)															
	L2ZT	L2TC (L2 Transparent Color)															L0ZT	L3TR (L3 Transparent Color)															
1A0	L0TEC (L0 Extend Transparency Control)																															
	L0EZT								L0ETC (L0 Extend Transparent Color)																							
1A4	L1TEC (L1 Transparent Extend Control)																															
	L1EZT								L1ETC (L1 Extend Transparent Color)																							
1A8	L2TEC (L2 Transparent Extend Control)																															
	L2EZT								L2ETC (L2 Extend Transparent Color)																							
1AC	L3TEC (L3 Transparent Extend Control)																															
	L0EZT								L3ETC (L3 Extend Transparent Color)																							
1B0	L4ETC (L4 Extend Transparent Control)																															
	L4EZT								L4ETC (L4 Extend Transparent Color)																							
1B4	L5ETC (L5 Extend Transparent Control)																															
	L5EZT								L5ETC (L5 Extend Transparent Color)																							
1998	L6ETC (L6 Extend Transparent Control)																															
	L6EZT								L6ETC (L6 Extend Transparent Color)																							
199C	L7ETC (L7 Extend Transparent Control)																															
	L7EZT								L7ETC (L7 Extend Transparent Color)																							
1E0	L1YCR0 (L1 YC to Red Coefficient 0)																															
						a12																	a11										
1E4	L1YCR1 (L1 YC to Red Coefficient 1)																															
								b1															a13										
1E8	L1YCG0 (L1 YC to Green Coefficient 0)																															
						a22																	a21										
1EC	L1YCG1 (L1 YC to Green Coefficient 1)																															
								b2															a23										
1F0	L1YCB0 (L1 YC to Blue Coefficient 0)																															
						a32																	a31										
1F4	L1YCB0 (L1 YC to Blue Coefficient 0)																															
								b3															a33										

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
400	L0PAL0																															
	A								R								G								B							
404	L0PAL1																															
:	:																															
7FC	L0PAL255																															
800	L1PAL0																															
	A								R								G								B							
804	L1PAL1																															
:	:																															
BFC	L1PAL255																															
1000	L2PAL0																															
	A								R								G								B							
1004	L2PAL1																															
:	:																															
13FC	L2PAL255																															
1400	L0PAL0																															
	A								R								G								B							
1404	L0PAL1																															
:	:																															
17FC	L0PAL255																															

7.12.2 Common Control Register

VCCC (Video display/Capture Common Control)

VCCC

Register address	WbackBaseAddress + $1ff8_H$																															
Bit No.	31	30	29			16	15	14	13			12	11	10	9	8	7	6	5	4		3		2		1		0				
Bit field name	reserve								C1sel			C0sel	reserve							WBsr		C1sr		C0sr		V1sr		V0sr				
R/W	RW0								RW				RW0							RW												
Initial value	0								1			0	0							00000												

Performs overall control of display controllers, capture controllers and writeback.

Bit 0 V0sr (Vdisp0 software reset)

Specifies whether or not to perform software reset for display controller 0. Reset action is triggered by write of VCSR register. It is only specifying that this bit is written.

0 Performs no software reset.

1 Performs software reset.

Bit 1 V1sr (Vdisp1 software reset)

Specifies whether or not to perform software reset for display controller 1. Reset action is triggered by write of VCSR register. It is only specifying that this bit is written.

0 Performs no software reset.

1 Performs software reset.

Bit 2 C0sr (Capture0 software reset)

Specifies whether or not to perform software reset for capture controller 0. Reset action is triggered by write of VCSR register. It is only specifying that this bit is written.

0 Performs no software reset.

1 Performs software reset.

Bit 3 C1sr (Capture1 software reset)

Specifies whether or not to perform software reset for capture controller 1. Reset action is triggered by write of VCSR register. It is only specifying that this bit is written.

0 Performs no software reset.

1 Performs software reset.

Bit 4 WBsr (Capture1 software reset)

Specifies whether or not to perform software reset for Writeback controller. Reset action is triggered by write of VCSR register. It is only specifying that this bit is written.

0 Performs no software reset.

1 Performs software reset.

Bit 12 C0sel (Capture 0 select)

Selects an input of capture controller 0.

0 656 dedicated port

1 RGB/656 shared port

Bit 13 C1sel (Capture 1 select)

Selects an input of capture controller 1

0 656 dedicated port

1 RGB/656 shared port

VCSR (Video display/Capture Software Reset)

VCSR

Register address	WbackBaseAddress + 1ffc$_H$																													
Bit No.	31	30	29	28	27	26	25	24					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Bit field name	VCSR																													
R/W	R0WX																													
Initial value	0																													

This register executes software reset (= soft reset). Write data is ignored. Write operation causes a single-shot reset pulse to be generated internally. Registers subjected to software reset are selected by the VCCC register.

7.12.3 Writeback Register

WBC (Writeback Control)

Register address	WbackBaseAddress + 0$_H$																													
Bit No.	31	30	29	28	27	26	25	24					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Bit field name	WBE	reserve																												
R/W	RW	R0																												
Initial value	0	0																												

Bit 31 WBE (Writeback Enable)
When 1 is written to this bit, an instruction is issued to start writeback.
When writeback ends, this bit returns to 0 automatically.

WBM (Writeback Mode)

Register address	WbackBaseAddress + 04$_H$																													
Bit No.	31	30				24	23	22				18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve						WBW								BLEN	C24	RGBA	resv	ST				FM			rsv	Vsel		resv	
R/W	R0						RW								RW	RW	RW	R0	R				RW			R0	RW		R0	
Initial value	0						X								X	X	X	0	000				X			0	X		0	

Bit 3-2 Vsel (Synchronize)
Selects a writeback source.
00 Display controller 0 (dual display screen 0)
01 Display controller 1 (dual display screen 0)
10 Display controller 0 (dual display screen 1)
11 Display controller 1 (dual display screen 1)

Bit 7-5 FM (Field Mode)
Specifies a field selection mode.
001 Odd field mode (non-interlace)
010 Even field mode
011 Both fields mode
100 Both fields mode (no field discrimination is performed)

Bit 10-8 ST (Status)

Indicates an execution state.

000 Initial state

001 Odd field mode / Writeback in progress

010 Even field mode / Writeback in progress

100 Both fields mode / Writeback in progress for first field

Bit 13 RGBA

Specifies a pixel data format.

0 ARGB format

1 RGBA format

Bit 14 C24 (Color 24bit)

Specifies the data size of 1 pixel.

0 16bit/pixel

1 24bit/pixel

Bit 15 BLEN（Field Mode）

Specifies value of A field of write data

0 Disables blend.

1 Enables blend.

Bit 23-16 WBW (Writeback Width)

Specifies the memory width (the stride) of the write destination image area in units of 64 bytes.

WBOA0 (Writeback Origin Address 0)

Register address	WbackBaseAddress + 08_H																															
Bit No.	31	30	29	28	27	26	25	24					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Bit field name	reserve				WBOA0																												
R/W	R0				RW																										R0		
Initial value	0				Undefined																										0		

Bit 27-0 WBOA0 (Writeback Origin Address 0)

Specifies the starting address of write destination image area 0.

WBOA1 (Writeback Origin Address 1)

Register address	WbackBaseAddress + $0C_H$																															
Bit No.	31	30	29	28	27	26	25	24					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Bit field name	reserve				WBOA1																												
R/W	R0				RW																										R0		
Initial value	0				Undefined																										0		

Bit 27-0 WBOA0 (Writeback Origin Address 1)

Specifies the starting address of write destination image area 1.

WBSTR (Writeback Start)

Register address	WbackBaseAddress + 10_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve				WIVSTR												reserve				WIHSTR											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies the coordinates of the upper left point in the writeback target area in the display screen.

Bit 11-0 WIHSTR (Writeback Image Horizontal Start)
Specifies the X coordinate.

Bit 27-16 WIVSTR (Writeback Image Vertical Start)
Specifies the Y coordinate.

WBEND (Writeback End)

Register address	WbackBaseAddress + 14_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve				WIVEND												reserve				WIHEND											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies the coordinates of the lower right point in the writeback target area in the display screen.

Bit 11-0 WIHEND (Writeback Image Horizontal END)
Specifies the X coordinate.

Bit 27-16 WIVEND (Writeback Image Vertical END)
Specifies the Y coordinate.

7.12.4 Display Controller Register

DCM0(Display Control Mode 0)/ DCM1(Display Control Mode1)

DCM0

Register address	DisplayBaseAddress + 0x0																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserve											L45E	L23E	L1E	L0E	CKS	Resv	SC						EEQ	EGV	FF	Fdet	SF	ESY	SYNC	
R/W	RW	W0	RX										RW	RW	RW	RW	RW	R0	RW						RW	RW	RW	R0	RW	RW	RW	
Initial value	0	0											0	0	0	0	0	0	1110						0	0	0	0	0	0	00	

DCM1

Register address	DisplayBaseAddress + 0x100																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserve			LA3E	LA2E	LA1E	LA0E	L7E	L6E	L5E	L4E	L3E	L2E	L1E	L0E	CKS	LCS	SC						EEQ	EGV	FF	Fdet	SF	ESY	SYNC	
R/W	RW	W0	RX		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW						RW	RW	RW	R0	RW	RW	RW	
Initial value	0	0	X		0	0	0	0	0	0	0	0	0	0	0	0	0	0	11101						0	0	0	0	0	0	00	

These registers set a display control mode. The entity of the DCM0 and DCM1 registers is a single register, but they are separated because of the format difference of partial bit fields. The 2 registers never hold a value different from each other. These registers are not initialized by software reset.

Bit 1-0 SYNC (Synchronize)
Sets a synchronous mode.
00 Non-interlace mode
01 Interlace mode
11 Interlace video mode

Bit 2 ESY (External Synchronize)
Sets external synchronous mode.
0 Disables external synchronization.
1 Enables external synchronization.

Bit 3 SF (Synchronize signal Format)
Sets the format of the synchronization signal (VSYNC, HSYNC).
0 Negative logic
1 Positive logic

Bit 4 Fdet (Field detection on external sync)
0 Field detection on ext. sync off
1 Field detection on ext. sync on

Bit 5 FF (Field Format)
Used when Bit 4 (Fdet) is 1 to determine the polarity of a filed (odd/even)
0 Normal field polarity
1 Inverted field polarity

Bit 6 Reserved

Bit 7 EEQ (Enable Equalizing pulse)
Sets the mode of the CCYNC signal.
0 Inserts no equalization pulse into the CCYNC signal.
1 Inserts an equalization pulse into the CCYNC signal.

Bit 13-8 SC (Scaling)

Divides the display reference clock by the set ratio, generating dot clocks.

DCM0		DCM1	
x00000	Does not divide the clock.	000000	Does not divide the clock.
x00001	Divides the clock to 1/4.	000001	Divides the clock to 1/2.
x00010	Divides the clock to 1/6.	000010	Divides the clock to 1/3.
x00011	Divides the clock to 1/8.	000011	Divides the clock to 1/4.
:		:	
x11111	Divides the clock to 1/64.		
		111111	Divides the clock to 1/64.

When the field in DCM0 is set to n, the frequency division ratio is 1/(2n + 2).

When the field in DCM1 is set to m, the frequency division ratio is 1/(m + 1).

Both of the registers are basically the setting parameters of the same function, and 2n + 2 = m + 1. Therefore, m = 2n + 1. When the SC field of DCM0 is set to n, 2n + 1 is reflected in DCM1.

When PLL is selected as the reference clock, the SC field does not function when the SC field is set to 1/1 to 1/5. In this case, the DC field is set to a ratio other than these.

Bit 14 LCS (Lower frequency Clock Select)

Selects the frequency of the built-in PLL clock.

0 Standard frequency (533 MHz)

1 Lower frequency (266 MHz)

Bit 15 CKS (Clock Source)

Selects the reference clock.

0 Sets the built-in PLL output as the reference clock.

1 Sets the DCLKI signal input as the reference clock.

Bit 16 L0E (L0-layer Enable)

Enables L0 layer display.

0 Performs no L0 layer display.

1 Performs L0 layer display.

Bit 17 L1E (L1-layer Enable)

Enables L1 layer display.

0 Performs no L1 layer display.

1 Performs L1 layer display.

Bit 18 L23E (L2 & L3-layer Enable)-------- DCM0

Enables L2 layer display and L3 layer display at the same time. These layers correspond to the M layer for existing products.

0 Performs neither L2 layer display nor L3 layer display.

1 Performs L2 layer display and L3 layer display.

L2E (L2-layer Enable) ------- DCM1

Enables L2 layer display.

0 Performs no L2 layer display.

1 Performs L2 layer display.

Bit 19 L45E (L4 & L5-layer Enable)-------- DCM0

Enables L4 layer display and L5 layer display at the same time. These layers correspond to the B layer for existing products.

0 Performs neither L4 layer display nor L5 layer display.

1 Performs L4 layer display and L5 layer display.

L3E (L3-layer Enable)------- DCM1

Enables L3 layer display.

0 Performs no L3 layer display.

1 Performs L3 layer display.

Bit 20 L4E (L4-layer Enable)

Enables L4 layer display

0 Performs no L4 layer display.

1 Performs L4 layer display.

Bit 21 L5E (L5-layer Enable)

Enables L5 layer display.

0 Performs no L5 layer display.

1 Performs L5 layer display.

Bit 22 L6E (L6-layer Enable)

Enables L6 layer display.

0 Performs no L6 layer display.

1 Performs L6 layer display.

Bit 23 L7E (L5-layer Enable)

Enables L7 layer display.

0 Performs no L7 layer display.

1 Performs L7 layer display.

Bit 24 La0E (La0-layer Enable)

Enables the La0 layer.

0 Performs no LA0 layer display.

1 Performs LA0 layer display.

Bit 25 La1E (La1-layer Enable)

Enables the La1 layer.

0 Performs no LA1 layer display.

1 Performs LA1 layer display.

Bit 26 La2E (La2-layer Enable)

Enables the La2 layer.

0 Performs no LA2 layer display.

1 Performs LA2 layer display.

Bit 27 La3E (La3-layer Enable)

Enables the La3 layer.

0 Performs no LA3 layer display.

1 Performs LA3 layer display.

Bit 31 DEN (Display Enable)

Enables display.

0 Performs no output of the display signal.

1 Performs output of the display signal.

DCM2 (Display Control Mode 2)

Register address	DisplayBaseAddress + 0x104															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved														RUF	RUM
R/W	R0														RW	RW
Initial value	0														0	0

Bit 0 RUM (Register Update Mode)

Selects the mode where register values are reflected in synchronization with vertical synchronization.

0 Reflects register update in the internal control circuit in real time. When update is performed during display period, display is distorted.

1 Notifies register values to the internal control circuit in synchronization with vertical synchronization. Simultaneity is controlled by the RUF flag described later.

Bit 1 RUF (Register Update Flag)

When "1" is written to this flag, an instruction is issued to update the value at the next vertical synchronization. When the update ends, this flag returns to "0".

0 Indicates the Initial state or that update ends.

1 Indicates that vertical synchronization is waited for.

DCM3 (Display Control Mode 3)

Register address	DispBaseAddress + 0x108															
Bit No.	31	30	29	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve						reserve		DCKed	DCKinv	reserve		DCKD					
R/W	RO						RW0		RW	RW	RW0		RW					
Initial value	0						00		0	0	00		000000					

Bit 5-0 DCKD (Display Clock Delay)

Defines an additional delay in units of internal PLL clocks.

00000 No additional delay
00010 +2 PLL clock
00100 +3 PLL clock
00110 +4 PLL clock
: :
11110 +16 PLL clock
xxxx1 reserved

Bit 8 DCKinv (Display Clock inversion)

0: DCLKO output signal is not inverted

1: DCLKO output signal is inverted.

Bit 9 DCKed (Display clock edge)

Defines an edge or edges used.

0: Single edge mode. RGB output is performed at the rising edge.

1: Both edges mode. RGB output is performed at the positive and negative edges.

HTP (Horizontal Total Pixels)

Register address	DisplayBaseAddress + 0x04															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				HTP											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the horizontal total pixel count. "Set value +1" is the horizontal total pixel count.

HDP (Horizontal Display Period)

Register address	DisplayBaseAddress + 0x08															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HDP											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the horizontal display period in units of pixel clocks. "Set value +1" is the pixel count for the display period.

HDB (Horizontal Display Boundary)

Register address	DisplayBaseAddress + 0x08															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				HDB											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the display period for the left screen in units of pixel clocks. "Set value +1" is the pixel count for the display period for the left screen. When performing no lateral split display, set the same value as HDP.

HSP (Horizontal Synchronize pulse Position)

Register address	DisplayBaseAddress + 0x0C															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HSP											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the pulse position of the horizontal synchronization signal in units of pixel clocks. When the clock count from the start of the display period (offset (15 clocks) is included) reaches "set value + 1", the horizontal synchronization signal is asserted.

HSW (Horizontal Synchronize pulse Width)

Register address	DisplayBaseAddress + 0x0C							
Bit No.	23	22	21	20	19	18	17	16
Bit field name	HSW							
R/W	RW							
Initial value	Undefined							

Specifies the pulse width of the horizontal synchronization signal in units of pixel clocks. "Set value + 1" is the clock count of the pulse width.

VSW (Vertical Synchronize pulse Width)

Register address	DisplayBaseAddress + 0x0C							
Bit No.	31	30	29	28	27	26	25	24
Bit field name	VSWH	Reserved	VSW					
R/W	RW	R0	RW					
Initial value	0	0	Undefined					

Bit 5-0 VSW (Vertical Synchronize pulse Width)
Specifies the pulse width of the vertical synchronization signal in units of rasters. "Set value + 1" is the raster count of the pulse width.

Bit 7 VSWH (VSW Half)
Extends the pulse width of the vertical synchronization signal by half of 1 raster.

VTR (Vertical Total Rasters)

Register address	DisplayBaseAddress + 0x10															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				VTR											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the vertical total raster count. "Set value + 1" is the vertical total raster count. For interlace display, "set value + 1.5" is the vertical total raster count for 1 field, and "2 × set value + 3" is the vertical total raster count for 1 frame. (See Section *7.5.2 "Interlace Display"*.)

VSP (Vertical Synchronize pulse Position)

Register address	DisplayBaseAddress + 0x14															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VSP											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the pulse position of the vertical synchronization signal in units of rasters. The vertical synchronizing pulse is asserted at the (set value + 1)-th raster relative to the display starting raster.

VDP (Vertical Display Period)

Register address	DisplayBaseAddress + 0x14															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				VDP											
R/W	R0				RW											
Initial value	0				Undefined											

Specifies the vertical display period in units of rasters. Set value + 1 is the display raster count.

L0M (L0-layer Mode)

Register address	DisplayBaseAddress + 0x20																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L0C	Reserve							L0W								Reserve				L0H											
R/W	RW	R0							RW								R0				RW											
Initial value	0	0							Undefined								0				Undefined											

Bit 11-0 L0H (L0-layer Height)

Specifies the height of the L0 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L0W (L0-layer memory Width)

Sets the memory width (the stride) of the L0 layer logical frame in units of 64 bytes.

Bit 31 L0C (L0-layer Color mode)

Sets the L0 layer color mode.

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) ARGB mode

L0EM (L0-layer Extended Mode)

Register address	DisplayBaseAddress + 0x110																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	-----		4	3	2	1	0			
Bit field name	L0EC		Reserve						L0PB				Reserve																L0WP			
R/W	RW		R0						RW				R0																RW			
Initial value			0										0																0			

Bit 0 L0 WP (L0-layer Window Position enable)

Selects the L0 layer display position.

0 Compatible mode display (compatible with C layer)

1 Window display

Bit 23-20 L0PB (L0-layer Palette Base)

Indicates the value added to the index when drawing the L0 layer palette. A value 16 times the set value is added to the index.

Bit 31-30 L0EC (L0-layer Extended Color mode)

Sets the L0 layer extended color mode.

00 Depends on L0C.

01 Direct color (24 bits/pixel) mode

10 Direct color (16 bits/pixel) RGBA mode

11 Direct color (24 bits/pixel) RGBA mode

L0OA (L0-layer Origin Address)

Register address	DisplayBaseAddress + 0x24																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L0OA																							
R/W	R0								RW																						RW0	
Initial value	0								Undefined																						0000	

Sets the logical frame origin address of L0 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L0DA (L0-layer Display Address)

Register address	DisplayBaseAddress + 0x28																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L0DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of L0 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L0DP (L0-layer Display Position)

Register address	DisplayBaseAddress + 0x2c																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0DY												Reserved				L0DX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display starting position coordinates (DX,DY) of L0 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L0DX (L0-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L0DY (L0-layer Display Position Y)
Specifies the Y coordinate.

L0WP (L0-layer Window Position)

Register address	DisplayBaseAddress + 0x114																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WY												Reserved				L0WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L0 layer window. The origin is the upper left point of the display screen.

Bit 11-0 L0WX (L0-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L0WY (L0-layer Window Position Y)
Specifies the Y coordinate.

L0WS (L0-layer Window Size)

Register address	DisplayBaseAddress + 0x116																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WH												Reserved				L0WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L0 layer window.

Bit 11-0 L0WW (L0-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L0WY (L0-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L0RM (L0-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x1880																															
Bit No.	31	30	29	28									19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve																L0SE	Reserve							L0RP							
R/W	R0																RW	R0							RW							
Initial value	0																0	0							111111111							

Controls read skip operation.

Bit 0-8 L0RP (L0-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L0SE (L0-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L0PX (L0-layer Partition X)

Register address	DisplayBaseAddress + 0x1884																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0PX1												Reserved				L0PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L0 layer when performing read skip.

Bit 11-0 L0PX0 (L0-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L0PX1 (L0-layer Partition X1)
Specifies the X coordinate.

L0PY (L0-layer Partition Y)

Register address	DisplayBaseAddress + 0x1888																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0PY1												Reserved				L0PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L0 layer when performing read skip.

Bit 11-0 L0PX0 (L0-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L0PX1 (L0-layer Partition Y1)
Specifies the Y coordinate.

L1M (L1-layer Mode)

Register address	DisplayBaseAddress + 0x30																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	-----			5	4	3	2	1	0
Bit field name	L1C	L1YC	L1CS	L1IM	Reserve				L1W								Reserve												
R/W	RW	RW	RW	RW	R0				RW								R0												
Initial value	0	0	0	0	0				Undefined								0												

Bit 23-16 L1W (L1-layer memory Width)
Sets the memory width (the stride) of the L1 layer logical frame in units of 64 bytes.

Bit 28 L1IM (L1-layer Interlace Mode)
Sets a video capture operation mode when L1CS is in capture mode.
0 Normal mode
1 For non-interlace display, performs display in WEAVE mode. For interlace display and interlace video display, performs buffer management in units of frames (a frame is a pair of an odd field and an even field).

Bit29 L1CS (L1-layer Capture Syncronize)
Sets whether to use L1 layer as a normal display layer or as a video capture layer.
0 Normal mode
1 Capture mode

Bit 30 L1YC (L1-layer YC mode)
Sets an L1 layer color format.
0 RGB mode
1 YC mode

Bit 31 L1C (L1-layer Color mode)
Sets an L1 layer color mode.
0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L1EM (L1-layer Extended Mode)

Register address	DisplayBaseAddress + 0x120																												
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	...	4	3	2	1	0
Bit field name	L1EC		Resv		Resv		VMAG		L1PB				Reserve			Reserve												
R/W	RW		R0	RW0	R0		RW		RW				R0			RW0												
Initial value	0		0	0	0		0		0				0			0	0				0							

Bit 23-20 L1PB (L1-layer Palette Base)
Indicates the value added to the index when drawing the L1 layer palette. A value 16 times the set value is added to the index.

Bit 25-24 VMAG (Video Magnify)
Specifies that the capture image be enlarged.
00 Does not use the enlarge function.
01 Reserved.
10 Uses the enlarge function.
11 Reserved.

Bit 31-30 L1EC (L1-layer Extended Color mode)
Sets the L1 layer extended color mode.
00 Depends on L1C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L1DA (L1-layer Display Address)

Register address	DisplayBaseAddress + 0x34																												
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L1DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of L1 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

Wraparound processing is not performed for L1 layer, and so the X,Y coordinates of the frame origin linear address and the display position are not specified.

This register is assigned at the same address as the CBDA0 register described later. Which register of the two is enabled is determined by the L1M register's L1CS bit.
L1CS=0: Enables the L1DA register.

L1CS=1: Enables the CBDA0 register.

CBDA0 (Capture Buffer Display Address 0)

Register address	DisplayBaseAddress + 0x34																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				CBDA0																											
R/W	R0				R																											
Initial value	0				Undefined																											

This register is a read-only register which can be accessed when the L1M register's L1CS bit is 1. This register indicates the starting address of the displayed capture image. When the L1CS bit is 1 and the L1IM bit is also 1, this register indicates the starting address of an odd field of the capture screen.

CBDA1 (Capture Buffer Display Address 1)

Register address	DisplayBaseAddress + 0x38																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				CBDA1																											
R/W	R0				R																											
Initial value	0				Undefined																											

This register is a read-only register which is only enabled when the L1CS bit is 1 and the L1IM bit is also 1. This register indicates the starting address of an even field of the capture screen.

L1WP (L1-layer Window Position)

Register address	DisplayBaseAddress + 0x124 （DispplayBaseAddress + 18h）																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L1WY												Reserved				L1WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L1 layer window. The origin is the upper left point of the display screen. This register is mapped into two addresses.

Bit 11-0 L1WX (L1-layer Window Position X)
Specifies the X coordinate.

Bit 23-16 L1WY (L1-layer Window Position Y)
Specifies the Y coordinate.

L1WS(L1-layer Window Size)

Register address	DisplayBaseAddress + 0x128 （DispplayBaseAddress + 1Ch）																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L1WH												Reserved				L1WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L1 layer window. This register has been mapped into two addresses.

Bit 11-0 L1WW (L1-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 23-16 L1WY (L1-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L2M (L2-layer Mode)

Register address	DisplayBaseAddress + 0x40																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	--	4	3	2	1	0			
Bit field name	L2C	L2FLP		Reserve					L2W								Reserve				L2H											
R/W	RW	RW		RW0	R0				RW								R0				RW											
Initial value	0	00		0	0000				Undefined								0000				Undefined											

Bit 11-0 L2H (L2-layer Height)

Specifies the height of the L2 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L2W (L2-layer memory Width)

Sets the memory width (the stride) of the L2 layer logical frame in units of 64 bytes.

Bit 30-29 L2FLP (L2-layer Flip mode)

Sets a flipping mode of L2 layer.

00 Displays side 0.
01 Displays side 1.
10 Displays side 0 and side 1 alternately on a frame by frame basis.
11 Reserved.

Bit 31 L2C (L2-layer Color mode)

Sets a color mode of L2 layer.

0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L2EM (L2-layer Extended Mode)

Register address	DisplayBaseAddress + 0x130																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10			4	3	2	1	0			
Bit field name	L2EC		Reserve						L2PB				Reserve															L2OM	L2WP			
R/W	RW		R0						RW				R0															RW	RW			
Initial value	00		0						0				0															0	0			

Bit 0 L2 WP (L2-layer Window Position enable)

Selects the display position of L2 layer.

0 Compatible mode display (compatible with ML layer)
1 Window display

Bit 1 L2OM (L2-layer Overlay Mode)

Selects a superimposition mode of L2 layer.

0 Compatible mode
1 Extended mode

Bit 23-20 L2PB (L2-layer Palette Base)

Indicates the value added to the index when drawing the L2 layer palette. A value 16 times the set value is added to the index.

Bit 31-30 L2EC (L2-layer Extended Color mode)

Sets the L2 layer extended color mode.

00 Depends on L2C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L2OA0 (L2-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x44																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L2OA0																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 0 of L2 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L2DA0 (L2-layer Display Address 0)

Register address	DisplayBaseAddress + 0x48																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L2DA0																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of side 0 of L2 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L2OA1 (L2-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x4C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L2OA1																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L2 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L2DA1 (L2-layer Display Address 1)

Register address	DisplayBaseAddress + 0x50																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L2DA0																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of side 1 of L2 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L2DP (L2-layer Display Position)

Register address	DisplayBaseAddress + 0x54																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2DY												Reserved				L2DX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display starting position coordinates (DX,DY) of L2 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L2DX (L2-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L2DY (L2-layer Display Position Y)
Specifies the Y coordinate.

L2WP (L2-layer Window Position)

Register address	DisplayBaseAddress + 0x134																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WY												Reserved				L2WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L2 layer window. The origin is the upper left point of the display screen.

Bit 11-0 L2WX (L2-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L2WY (L2-layer Window Position Y)
Specifies the Y coordinate.

L2WS (L2-layer Window Size)

Register address	DisplayBaseAddress + 0x138																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WH												Reserved				L2WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L2 layer window.

Bit 11-0 L2WW (L2-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L2WY (L2-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L2RM (L2-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x18A0																			
Bit No.	31	30	29	28		19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve									L2SE	reserve							L2RP							
R/W	R0									RW	R0							RW							
Initial value	0									0	0							111111111							

Controls read skip operation.

Bit 0-8 L2RP (L2-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L2SE (L2-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L2PX (L2-layer Partition X)

Register address	DisplayBaseAddress + 0x18A4																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2PX1												Reserved				L2PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L2 layer when performing read skip.

Bit 11-0 L2PX0 (L2-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L2PX1 (L2-layer Partition X1)
Specifies the X coordinate.

L2PY (L2-layer Partition Y)

Register address	DisplayBaseAddress + 0x18A8																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2PY1												Reserved				L2PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L2 layer when performing read skip.

Bit 11-0 L2PY0 (L2-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L2PY1 (L2-layer Partition Y1)
Specifies the Y coordinate.

L3M (L3-layer Mode)

Register address	DisplayBaseAddress + 0x58																															
Bit No.	31	30	29	28	27	--	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L3C	L3FLP		Reserve				L3W								Reserve				L3H												
R/W	RW	RW		R0				RW								R0				RW												
Initial value	0	0		0				Undefined								0				Undefined												

Bit 11-0 L3H (L3-layer Height)
Specifies the height of the L3 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L3W (L3-layer memory Width)
Sets the memory width (the stride) of the L3 layer logical frame in units of 64 bytes.

Bit 30-29 L3FLP (L3-layer Flip mode)
Sets the L3 layer flipping mode.
- 00 Displays side 0.
- 01 Displays side 1.
- 10 Displays side 0 and side 1 alternately on a frame by frame basis.
- 11 Reserved.

Bit 31 L3C (L3-layer Color mode)
Sets a color mode of L3 layer.
- 0 Indirect color (8 bits/pixel) mode
- 1 Direct color (16 bits/pixel) ARGB mode

L3EM (L3-layer Extended Mode)

Register address	DisplayBaseAddress + 0x140																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10		4	3	2	1	0				
Bit field name	L3EC		Reserve						L3PB				Reserve														L3OM	L3WP					
R/W	RW		R0						RW				R0														RW	RW					
Initial value	00		0						0				0														0	0					

Bit 0 L3 WP (L3-layer Window Position enable)
Selects the display position of L3 layer.
- 0 Compatible mode display (compatible with MR layer)
- 1 Window display

Bit 1 L3OM (L3-layer Overlay Mode)
Selects a superimposition mode of L3 layer.
- 0 Compatible mode
- 1 Extended mode

Bit 23-20 L3PB (L3-layer Palette Base)
Indicates the value added to the index when drawing the L3 layer palette. A value 16 times the set value is added to the index.

Bit 31-30 L3EC (L3-layer Extended Color mode)
Sets the L3 layer extended color mode.
- 00 Depends on L3C.
- 01 Direct color (24 bits/pixel) ARGB mode
- 10 Direct color (16 bits/pixel) RGBA mode
- 11 Direct color (24 bits/pixel) RGBA mode

L3OA0 (L3-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x5C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L3OA0																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the logical frame origin address of side 0 of L3 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L3DA0 (L3-layer Display Address 0)

Register address	DisplayBaseAddress + 0x60																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L3DA0																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 0 of L3 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L3OA1 (L3-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x64																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L3OA1																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L3 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L3DA1 (L3-layer Display Address 1)

Register address	DisplayBaseAddress + 0x68																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L3DA1																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 1 of L3 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L3DP (L3-layer Display Position)

Register address	DisplayBaseAddress + 0x6c																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3DY												Reserved				L3DX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display starting position coordinates (DX,DY) of L3 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L3DX (L3-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L3DY (L3-layer Display Position Y)
Specifies the Y coordinate.

L3WP (L3-layer Window Position)

Register address	DisplayBaseAddress + 0x144																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WY												Reserved				L3WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L3 layer window. The origin is the upper left point of the display screen.

Bit 11-0 L3WX (L3-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L3WY (L3-layer Window Position Y)
Specifies the Y coordinate.

L3WS (L3-layer Window Size)

Register address	DisplayBaseAddress + 0x148																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WH												Reserved				L3WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L3 layer window.

Bit 11-0 L3WW (L3-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L3WY (L3-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L3RM (L3-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x18B0																			
Bit No.	31	30	29	28	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve								L3SE	reserve							L3RP							
R/W	R0								RW	R0							RW							
Initial value	0								0	0							111111111							

Controls read skip operation.

Bit 0-8 L3RP (L3-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L3SE (L3-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L3PX (L3-layer Partition X)

Register address	DisplayBaseAddress + 0x18B4																			
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3PX1												Reserved				L3PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L3 layer when performing read skip.

Bit 11-0 L3PX0 (L3-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L3PX1 (L3-layer Partition X1)
Specifies the X coordinate.

L3PY (L3-layer Partition Y)

Register address	DisplayBaseAddress + 0x18B8																			
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3PY1												Reserved				L3PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L3 layer when performing read skip.

Bit 11-0 L3PX0 (L3-layer Partition Y0)
Specifies the Y coordinate.

Bi t27-16 L3PX1 (L3-layer Partition Y1)
Specifies the Y coordinate.

L4M (L4-layer Mode)

Register address	DisplayBaseAddress + 0x70																															
Bit No.	31	30	29	28	27	--	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L4C	L4FLP	Reserve					L4W								Reserve				L4H												
R/W	RW	RW	R0					RW								R0				RW												
Initial value	0	0	0					Undefined								0				Undefined												

Bit 11-0 L4H (L4-layer Height)
Specifies the height of the L4 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L4W (L4-layer memory Width)
Sets the memory width (the stride) of the L4 layer logical frame in units of 64 bytes.

Bit 30-29 L4FLP (L4-layer Flip mode)
Sets a flipping mode of L4 layer.
00 Displays side 0.
01 Displays side 1.
10 Displays side 0 and side 1 alternately on a frame by frame basis.
11 Reserved.

Bit 31 L4C (L4-layer Color mode)
Sets a color mode of L4 layer.
0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L4EM (L4-layer Extended Mode)

Register address	DisplayBaseAddress + 0x150																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10					4	3	2	1	0	
Bit field name	L4EC								Reserve																					L4OM	L4WP	
R/W	RW								R0																					RW	RW	
Initial value	00								0																					0	0	

Bit 0 L4 WP (L4-layer Window Position enable)
Selects the display position of L4 layer.
0 Compatible mode display (compatible with BL layer)
1 Window display

Bit 1 L4OM (L4-layer Overlay Mode)
Selects a superimposition mode of L4 layer.
0 Compatible mode
1 Extended mode

Bit 31-30 L4EC (L4-layer Extended Color mode)
Sets the L4 layer extended color mode.
00 Depends on L4C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L4OA0 (L4-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x74																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L4OA0																											
R/W	R0				RW																										RW0	
Initial value	0				不定																											

Sets the logical frame origin address of side 0 of L4 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L4DA0 (L4-layer Display Address 0)

Register address	DisplayBaseAddress + 0x78																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L4OA0																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 0 of L4 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L4OA1 (L4-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x7C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L4OA1																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L4 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L4DA1 (L4-layer Display Address 1)

Register address	DisplayBaseAddress + 0x80																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L4DA1																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 1 of L4 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L4DP (L4-layer Display Position)

Register address	DisplayBaseAddress + 0x84																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4DY												Reserved				L4DX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display starting position coordinates (DX,DY) of L4 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L4DX (L4-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L4DY (L4-layer Display Position Y)
Specifies the Y coordinate.

L4WP (L4-layer Window Position)

Register address	DisplayBaseAddress + 0x154																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WY												Reserved				L4WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L4 layer window. The origin is the upper left point of the display screen.

Bit 11-0 L4WX (L4-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L4WY (L4-layer Window Position Y)
Specifies the Y coordinate.

L4WS (L4-layer Window Size)

Register address	DisplayBaseAddress + 0x158																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WH												Reserved				L4WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L4 layer window.

Bit 11-0 L4WW (L4-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L4WY (L4-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L4RM (L4-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x18C0																															
Bit No.	31	30	29	28									19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve																L4SE	reserve							L4RP							
R/W	R0																RW	R0							RW							
Initial value	0																0	0							111111111							

Controls read skip operation.

Bit 0-8 L4RP (L4-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L4SE (L4-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L4PX (L4-layer Partition X)

Register address	DisplayBaseAddress + 0x18C4																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4PX1												Reserved				L4PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L4 layer when performing read skip.

Bit 11-0 L4PX0 (L4-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L4PX1 (L4-layer Partition X1)
Specifies the X coordinate.

L4PY (L4-layer Partition Y)

Register address	DisplayBaseAddress + 0x18C8																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4PY1												Reserved				L4PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L4 layer when performing read skip.

Bit 11-0 L4PY0 (L4-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L4PY1 (L4-layer Partition Y1)
Specifies the Y coordinate.

L5M (L5-layer Mode)

Register address	DisplayBaseAddress + 0x88																															
Bit No.	31	30	29	28	27	--	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L5C	L5FLP		Reserve				L5W								Reserve				L5H											
R/W	RW	RW		R0				RW								R0				RW											
Initial value	0	0		0				Undefined								0				Undefined											

Bit 11-0 L5H (L5-layer Height)
Specifies the height of the L5 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L5W (L5-layer memory Width)
Sets the memory width (the stride) of the L5 layer logical frame in units of 64 bytes.

Bit 30-29 L5FLP (L5-layer Flip mode)
Sets a flipping mode of L5 layer.
00 Displays side 0.
01 Displays side 1.
10 Displays side 0 and side 1 alternately on a frame by frame basis.
11 Reserved.

Bit 31 L5C (L5-layer Color mode)
Sets a color mode of L5 layer.
0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L5EM (L5-layer Extended Mode)

Register address	DisplayBaseAddress + 0x160																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10					4	3	2	1	0	
Bit field name	L5EC		Reserve																											L5OM	L5WP	
R/W	RW		R0																											RW	RW	
Initial value	00		0																											0	0	

Bit 0 L5 WP (L5-layer Window Position enable)
Selects the display position of L5 layer.
0 Compatible mode display (compatible with BR layer)
1 Window display

Bit 1 L5OM (L5-layer Overlay Mode)
Selects a superimposition mode of L5 layer.
0 Compatible mode
1 Extended mode

Bit 31-30 L5EC (L5-layer Extended Color mode)
Sets the L5 layer extended color mode.
00 Depends on L5C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L5OA0 (L5-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x8C																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L5OA0																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 0 of L5 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L5DA0 (L5-layer Display Address 0)

Register address	DisplayBaseAddress + 0x90																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L5DA0																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 0 of L5 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L5OA1 (L5-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x94																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L5OA1																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L5 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L5DA1 (L5-layer Display Address 1)

Register address	DisplayBaseAddress + 0x98																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L5DA1																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 1 of L5 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L5DP (L5-layer Display Position)

Register address	DisplayBaseAddress + 0x9C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WY												Reserved				L5DX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display starting position coordinates (DX,DY) of L5 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L5DX (L5-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L5DY (L5-layer Display Position Y)
Specifies the Y coordinate.

L5WP (L5-layer Window Position)

Register address	DisplayBaseAddress + 0x164																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WY												Reserved				L5WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the L5 layer window. The origin is the upper left point of the display screen

Bit 11-0 L5WX (L5-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L5WY (L5-layer Window Position Y)
Specifies the Y coordinate.

L5WS (L5-layer Window Size)

Register address	DisplayBaseAddress + 0x168																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WH												Reserved				L5WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the L5 layer window.

Bit 11-0 L5WW (L5-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L5WY (L5-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L5RM (L5-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x18D0																															
Bit No.	31	30	29	28				19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Bit field name	reserve											L5SE	reserve							L5RP												
R/W	R0											RW	R0							RW												
Initial value	0											0	0							111111111												

Controls read skip operation.

Bit 0-8 L5RP (L5-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L5SE (L5-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L5PX (L5-layer Partition X)

Register address	DisplayBaseAddress + 0x18D4																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5PX1												Reserved				L5PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L5 layer when performing read skip.

Bit 11-0 L5PX0 (L5-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L5PX1 (L5-layer Partition X1)
Specifies the X coordinate.

L5PY (L5-layer Partition Y)

Register address	DisplayBaseAddress + 0x18D8																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5PY1												Reserved				L5PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L5 layer when performing read skip.

Bit 11-0 L5PX0 (L5-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L5PX1 (L5-layer Partition Y1)
Specifies the Y coordinate.

L6M (L6-layer Mode)

Register address	DisplayBaseAddress + 0x1900																															
Bit No.	31	30	29	28	27	--	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L6C	L6FLP	Reserve					L6W								Reserve				L6H												
R/W	RW	RW	R0					RW								R0				RW												
Initial value	0	0	0					Undefined								0				Undefined												

Bit 11-0 L6H (L6-layer Height)
Specifies the height of the L6 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L6W (L6-layer memory Width)
Sets the memory width (the stride) of the L6 layer logical frame in units of 64 bytes.

Bit 30-29 L6FLP (L6-layer Flip mode)
Sets a flipping mode of L6 layer.
00 Displays side 0.
01 Displays side 1.
10 Displays side 0 and side 1 alternately on a frame by frame basis.
11 Reserved.

Bit 31 L6C (L6-layer Color mode)
Sets a color mode of L6 layer.
0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L6EM (L6-layer Extended Mode)

Register address	DisplayBaseAddress + 0x1918																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10					4	3	2	1	0	
Bit field name	L6EC								Reserve																					L6OM	L6WP	
R/W	RW								R0																					RW	RW	
Initial value	00								0																					0	0	

Bit 0 L6 WP (L6-layer Window Position enable)
Selects the display position of L6 layer.
0 Compatible mode display
1 Window display

Bit 1 L6OM (L6-layer Overlay Mode)
Selects a superimposition mode of L6 layer.
0 Compatible mode
1 Extended mode

Bit 31-30 L6EC (L6-layer Extended Color mode)
Sets the L6 layer extended color mode.
00 Depends on L6C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L6OA0 (L6-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x1904																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L6OA0																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the logical frame origin address of side 0 of L6 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L6DA0 (L6-layer Display Address 0)

Register address	DisplayBaseAddress + 0x1908																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L6DA0																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 0 of L6 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L6OA1 (L6-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x190C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L6OA1																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L6 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L6DA1 (L6-layer Display Address 1)

Register address	DisplayBaseAddress + 0x1910																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						L6DA1																									
R/W	R0						RW																									
Initial value	0						Undefined																									

Sets the display origin address of side 1 of L6 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L6DP (L6-layer Display Position)

Register address	DisplayBaseAddress + 0x1914			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Bit field name	Reserved	L6WY	Reserved	L6DX
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the display starting position coordinates (DX,DY) of L6 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L6DX (L6-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L6DY (L6-layer Display Position Y)
Specifies the Y coordinate.

L6WP (L6-layer Window Position)

Register address	DisplayBaseAddress + 0x191c			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Bit field name	Reserved	L6WY	Reserved	L6WX
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the display position coordinates (WX,WY) of the L6 layer window. The origin is the upper left point of the display screen.

Bi 11-0 L6WX (L6-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L6WY (L6-layer Window Position Y)
Specifies the Y coordinate.

L6WS (L6-layer Window Size)

Register address	DisplayBaseAddress + 0x1920			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Bit field name	Reserved	L6WH	Reserved	L6WW
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the size of the L6 layer window.

Bit 11-0 L6WW (L6-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L6WY (L6-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L6RM (L6-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x1924			
Bit No.	31-19	18-17-16	15	14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
Bit field name	reserve		L6SE	reserve	L6RP
R/W	R0		RW	R0	RW
Initial value	0		0	0	111111111

Controls read skip operation.

Bit 0-8 L6RP (L6-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L6SE (L6-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L6PX (L6-layer Partition X)

Register address	DisplayBaseAddress + 0x1928			
Bit No.	31-30-29-28-27-26-25-24-23-22-21-20-19-18-17-16	15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0		
Bit field name	Reserved	L6PX1	Reserved	L6PX0
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Specifies 2 X coordinates, which are the splitting boundary of L6 layer when performing read skip.

Bit 11-0 L6PX0 (L6-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L6PX1 (L6-layer Partition X1)
Specifies the X coordinate.

L6PY (L6-layer Partition Y)

Register address	DisplayBaseAddress + 0x192C			
Bit No.	31-30-29-28-27-26-25-24-23-22-21-20-19-18-17-16	15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0		
Bit field name	Reserved	L6PY1	Reserved	L6PY0
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Specifies 2 Y coordinates, which are the splitting boundary of L6 layer when performing read skip.

Bit 11-0 L6PY0 (L6-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L6PY1 (L6-layer Partition Y1)
Specifies the Y coordinate.

L7M (L7-layer Mode)

Register address	DisplayBaseAddress + 0x1940																															
Bit No.	31	30	29	28	27	--	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L7C	L7FLP		Reserve				L7W								Reserve				L7H												
R/W	RW	RW		R0				RW								R0				RW												
Initial value	0	0		0				Undefined								0				Undefined												

Bit 11-0 L7H (L7-layer Height)
Specifies the height of the L7 layer logical frame in units of pixels. Set value + 1 is the height.

Bit 23-16 L7W (L7-layer memory Width)
Sets the memory width (the stride) of the L7 layer logical frame in units of 64 bytes.

Bit 30-29 L7FLP (L7-layer Flip mode)
Sets a flipping mode of L7 layer.
00 Displays side 0.
01 Displays side 1.
10 Displays side 0 and side 1 alternately on a frame by frame basis.
11 Reserved.

Bit 31 L7C (L7-layer Color mode)
Sets a color mode of L7 layer.
0 Indirect color (8 bits/pixel) mode
1 Direct color (16 bits/pixel) ARGB mode

L7EM (L7-layer Extended Mode)

Register address	DisplayBaseAddress + 0x1958																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10					4	3	2	1	0	
Bit field name	L7EC		Reserve																											L7OM	L7WP	
R/W	RW		R0																											RW	RW	
Initial value	00		0																											0	0	

Bit 0 L7 WP (L7-layer Window Position enable)
Selects the display position of L7 layer.
0 Compatible mode display (compatible with BR layer)
1 Window display

Bit 1 L7OM (L7-layer Overlay Mode)
Selects a superimposition mode of L7 layer.
0 Compatible mode
1 Extended mode

Bit 31-30 L7EC (L7-layer Extended Color mode)
Sets the L7 layer extended color mode.
00 Depends on L7C.
01 Direct color (24 bits/pixel) ARGB mode
10 Direct color (16 bits/pixel) RGBA mode
11 Direct color (24 bits/pixel) RGBA mode

L7OA0 (L7-layer Origin Address 0)

Register address	DisplayBaseAddress + 0x1944																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L7OA1																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 0 of L7 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L7DA0 (L7-layer Display Address 0)

Register address	DisplayBaseAddress + 0x1948																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L7A0																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of side 0 of L7 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L7OA1 (L7-layer Origin Address 1)

Register address	DisplayBaseAddress + 0x194C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				L7OA1																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the logical frame origin address of side 1 of L7 layer. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

L7DA1 (L7-layer Display Address 1)

Register address	DisplayBaseAddress + 0x1950																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								L7A0																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of side 1 of L7 layer. For direct color mode (16 bits/pixel), it is assumed that the lower 1 bit is 0 and alignment is performed in units of 2 bytes.

L7DP (L7-layer Display Position)

Register address	DisplayBaseAddress + 0x1954			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Bit field name	Reserved	L7DY	Reserved	L7DX
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the display starting position coordinates (DX,DY) of L7 layer relative to the logical frame origin in units of pixels.

Bit 11-0 L7DX (L7-layer Display Position X)
Specifies the X coordinate.

Bit 27-16 L7DY (L7-layer Display Position Y)
Specifies the Y coordinate.

L7WP (L7-layer Window Position)

Register address	DisplayBaseAddress + 0x195c			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Bit field name	Reserved	L7WY	Reserved	L7WX
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the display position coordinates (WX,WY) of the L7 layer window. The origin is the upper left point of the display screen.

Bit 11-0 L7WX (L7-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 L7WY (L7-layer Window Position Y)
Specifies the Y coordinate.

L7WS (L7-layer Window Size)

Register address	DisplayBaseAddress + 0x1960			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Bit field name	Reserved	L7WH	Reserved	L7WW
R/W	R0	RW	R0	RW
Initial value	0	Undefined	0	Undefined

Sets the size of the L7 layer window.

Bit 11-0 L7WW (L7-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 L7WY (L7-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

L7RM (L7-layer Readskip Mode)

Register address	DisplayBaseAddress + 0x1964																			
Bit No.	31	30	29	28		19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserve									L7SE	reserve							L7RP							
R/W	R0									RW	R0							RW							
Initial value	0									0	0							111111111							

Controls read skip operation.

Bit 0-8 L7RP (L7-layer Read Partition)
Specifies whether or not to perform read operation of the partition corresponding to each bit.
0 Performs no data read.
1 Performs data read.

Bit 15 L7SE (L7-layer Skip Enable)
Specifies whether or not to enable the read skip function.
0 Disables read skip.
1 Enables read skip.

L7PX (L7-layer Partition X)

Register address	DisplayBaseAddress + 0x1968																										
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L7PX1												Reserved				L7PX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 X coordinates, which are the splitting boundary of L7 layer when performing read skip.

Bit 11-0 L7PX0 (L7-layer Partition X0)
Specifies the X coordinate.

Bit 27-16 L7PX1 (L7-layer Partition X1)
Specifies the X coordinate.

L7PY (L7-layer Partition Y)

Register address	DisplayBaseAddress + 0x196C																										
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L7PY1												Reserved				L7PY0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Specifies 2 Y coordinates, which are the splitting boundary of L7 layer when performing read skip.

Bit 11-0 L7PX0 (L7-layer Partition Y0)
Specifies the Y coordinate.

Bit 27-16 L7PX1 (L7-layer Partition Y1)
Specifies the Y coordinate.

LA0M (LA0-layer Mode)

Register address	DisplayBaseAddress + 0x1A00																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA0W								Reserve															
R/W	R0								RW								R0															
Initial value	0								Undefined								0															

Bit 23-16 LA0W(LA0-layer memory Width)
Sets the memory width (the stride) of the LA0 layer logical frame in units of 64 bytes.

LA0DA (LA0-layer Display Address)

Register address	DisplayBaseAddress + 0x1A04																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA0DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of LA0 layer.

LA0WP (LA0-layer Window Position)

Register address	DisplayBaseAddress + 0x1A08																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA0WY												Reserved				LA0WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the LA0 layer window. The origin is the upper left point of the display screen.

Bit 11-0 LA0WX (LA0-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 LA0WY (LA0-layer Window Position Y)
Specifies the Y coordinate.

LA0WS (LA0-layer Window Size)

Register address	DisplayBaseAddress + 0x1A0C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA0WH												Reserved				LA0WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the LA0 layer window.

Bit 11-0 LA0WW (LA0-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 LA0WY (LA0-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

LA1M (LA1-layer Mode)

Register address	DisplayBaseAddress + 0x1A10																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA1W								Reserve															
R/W	R0								RW								R0															
Initial value	0								Undefined								0															

Bit 23-16 LA1W (LA1-layer memory Width)
Sets the memory width (the stride) of the LA1 layer logical frame in units of 64 bytes.

LA1DA (LA1-layer Display Address)

Register address	DisplayBaseAddress + 0x1A14																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA1DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of LA1 layer.

LA1WP (LA1-layer Window Position)

Register address	DisplayBaseAddress + 0x1A18																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA1WY												Reserved				LA1WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the LA1 layer window. The origin is the upper left point of the display screen.

Bit 11-0 LA1WX (LA1-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 LA1WY (LA1-layer Window Position Y)
Specifies the Y coordinate.

LA1WS (LA1-layer Window Size)

Register address	DisplayBaseAddress + 0x1A1C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA1WH												Reserved				LA1WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the LA1 layer window.

Bit 11-0 LA1WW (LA1-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit27-16 LA1WY (LA1-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

LA2M (LA2-layer Mode)

Register address	DisplayBaseAddress + 0x1A20																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA2W								Reserve															
R/W	R0								RW								R0															
Initial value	0								Undefined								0															

Bit 23-16 LA2W (LA2-layer memory Width)
Sets the memory width (the stride) of the LA2 layer logical frame in units of 64 bytes.

LA2DA (LA2-layer Display Address)

Register address	DisplayBaseAddress + 0x1A24																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA2DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of LA2 layer.

LA2WP (LA2-layer Window Position)

Register address	DisplayBaseAddress + 0x1A28																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA2WY												Reserved				LA2WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the LA2 layer window. The origin is the upper left point of the display screen.

Bit 11-0 LA2WX (LA2-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 LA2WY (LA2-layer Window Position Y)
Specifies the Y coordinate.

LA2WS (LA2-layer Window Size)

Register address	DisplayBaseAddress + 0x1A2C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA2WH												Reserved				LA2WW											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the LA2 layer window.

Bit11-0 LA2WW (LA2-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit27-16 LA2WY (LA2-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

LA3M (LA3-layer Mode)

Register address	DisplayBaseAddress + 0x1A30																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA3W								Reserve															
R/W	R0								RW								R0															
Initial value	0								Undefined								0															

Bit 23-16 LA3W (LA3-layer memory Width)
Sets the memory width (the stride) of the LA3 layer logical frame in units of 64 bytes.

LA3DA (LA3-layer Display Address)

Register address	DisplayBaseAddress + 0x1A34																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								LA3DA																							
R/W	R0								RW																							
Initial value	0								Undefined																							

Sets the display origin address of LA3 layer.

LA3WP (LA3-layer Window Position)

Register address	DisplayBaseAddress + 0x1A38																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA3WY												Reserved				LA3WX											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (WX,WY) of the LA3 layer window. The origin is the upper left point of the display screen.

Bit 11-0 LA3WX (LA3-layer Window Position X)
Specifies the X coordinate.

Bit 27-16 LA3WY (LA3-layer Window Position Y)
Specifies the Y coordinate.

LA3WS (LA3-layer Window Size)

Register address	DisplayBaseAddress + 0x1A3C																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LA3WH												Reserved				LA3WW											
R/W	RW				R0												RW				RW											
Initial value	0				Undefined												0				Undefined											

Sets the size of the LA3 layer window.

Bit 11-0 LA3WW (LA3-layer Window Width X)
Specifies the width in units of pixels. Do not set 0.

Bit 27-16 LA3WY (LA3-layer Window Height Y)
Specifies the height. Set value + 1 is the height.

CUTC (CUrsor Transparent Control)

Register address	DisplayBaseAddress + 0xA0															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							CUZT	CUTC							
R/W	R0							RW	RW							
Initial value	0							Un-defined	Undefined							

Bit7-0 CUTC (Cursor Transparent Code)
Specifies color data treated as transparent.

Bit8 CUZT (Cursor Zero Transparency)
Sets treatment of code 0.
0 Treats code 0 as transparent.
1 Does not treat code 0 as transparent.

CPM (Cursor Priority Mode)

Register address	DisplayBaseAddress + 0xA2							
Bit No.	7	6	5	4	3	2	1	0
Bit field name	Reserved		CEN1	CEN0	Reserved		CUO1	CUO0
R/W	R0		RW	RW	R0		RW	RW
Initial value	0		0	0	0		0	0

Sets the priority of cursor display. Cursor 0 is displayed in preference to cursor 1.

Bit 0 CUO0 (Cursor Overlap 0)
Sets the display priority of cursor 0 and C layer (L0 layer).
0 Performs screen superimposition, placing cursor 0 below L0 layer.
1 Performs screen superimposition, placing cursor 0 above L0 layer.

Bit 1 CUO1 (Cursor Overlap 1)
Sets the display priority of cursor 1 and L0 layer.
0 Performs screen superimposition, placing cursor 1 below L0 layer.
1 Performs screen superimposition, placing cursor 1 above L0 layer.

Bit 4 CEN0 (Cursor Enable 0)
Sets display/non-display of cursor 0.
0 Does not display cursor 0.
1 Displays cursor 0.

Bit 5 CEN1 (Cursor Enable 1)
Sets display/non-display of cursor 1.
0 Does not display cursor 1.
1 Displays cursor 1.

CUOA0 (Cursor-0 Origin Address)

Register address	DisplayBaseAddress + 0xA4																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				CUOA0																											
R/W	R0				RW																								RW0			
Initial value	0				Undefined																											

Sets the starting address of the cursor 0 pattern. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

CUP0 (Cursor-0 position)

Register address	DisplayBaseAddress + 0xA8																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY0												Reserved				CUX0											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (CUX0,CUY0) of cursor 0 in units of pixels. The coordinate reference point is the upper left point of the cursor pattern.

Bit 11-0 CUX0 (Cursor0 Position X)
Specifies the X coordinate.

Bit 23-16 CUX0 (Cursor0 Position X)
Specifies the X coordinate.

CUOA1 (Cursor-1 Origin Address)

Register address	DisplayBaseAddress + 0xAC																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve				CUOA1																											
R/W	R0				RW																										RW0	
Initial value	0				Undefined																											

Sets the starting address of the cursor 1 pattern. Lower 4 bits are fixed to 0, and so 16-byte alignment is performed.

CUP1 (Cursor-1 position)

Register address	DisplayBaseAddress + 0xB0																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY1												Reserved				CUX1											
R/W	R0				RW												R0				RW											
Initial value	0				Undefined												0				Undefined											

Sets the display position coordinates (CUX1,CUY1) of cursor 1 in units of pixels. The coordinate reference point is the upper left point of the cursor pattern.

Bit 11-0 CUX1 (Cursor1 Position X)
Specifies the X coordinate.

Bit 23-16 CUX1 (Cursor1 Position X)
Specifies the X coordinate.

DLS (Display Layer Select)

Register address	DisplayBaseAddress + 0x180							
Bit No.	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Bit field name	DLS7	DLS6	DLS5	DLS4	DLS3	DLS2	DLS1	DSL0
R/W	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0111	0110	0101	0100	0011	0010	0001	0000

Defines the layer superimposition order.

Bit 3-0 DLS0 (Display Layer Select 0)
Selects the top layer.

0000	L0 layer
0001	L1 layer
:	:
0111	L7 layer
1000	Reserved.
:	:
1110	Reserved.
1111	Non-selection

Bit 7-4 DLS1 (Display Layer Select 1)
Selects the second layer. The content of this field is the same as DSL0.

Bit 11-8 DLS2 (Display Layer Select 2)
Selects the third layer. The content of this field is the same as DSL0.

Bit 15-12 DLS3 (Display Layer Select 3)
Selects the forth layer. The content of this field is the same as DSL0.

Bit 19-16 DLS4 (Display Layer Select 4)
Selects the fifth layer. The content of this field is the same as DSL0.

Bit 23-20 DLS5 (Display Layer Select 5)
Selects the sixth layer. The content of this field is the same as DSL0.

Bit 27-24 DLS6 (Display Layer Select 6)
Selects the seventh layer. The content of this field is the same as DSL0.

Bit 31-28 DLS7 (Display Layer Select 7)
Selects the bottom layer. The content of this field is the same as DSL0.

DLS6 and DLS7 are forcibly regarded as not being selected (0) under the following conditions:

Both L6EN and L7EN of the DCM1 register are 0.

This processing is for ensuring compatibility during operation of L6 layer. In general, the L6 layer display program for existing products specifies 0 for the field corresponding to DLS6/DLS7. To disable this setting, such a processing is performed.

MDC (Multi Display Control)

Register address	DisplayBaseAddress + 0x170																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MDen	reserve							SC1Xen				SC0Xen				SC1en								SC0en							
R/W	RW	R0							RW				RW				RW								RW							
Initial value	0	0							Undefined				Undefined				Undefined								Undefined							

Bit 0 SC0en0 (screen 0 enable 0)
0: L0 is not included in screen 0.
1: L0 is included in screen 0.

Bit 1 SC0en1 (screen 0 enable 1)
0: L1 is not included in screen 0.
1: L1 is included in screen 0.

～

Bit 5 SC0en5 (screen 0 enable 5)
0: L5 is not included in screen 0.
1: L5 is included in screen 0.

Bit 6 SC0en6 (screen 0 enable 6)
0: Cursor 0 is not included in screen 0.
1: Cursor 0 is included in screen 0.

Bit 7 SC0en7 (screen 0 enable 7)
0: Cursor 1 is not included in screen 0.
1: Cursor 1 is included in screen 0.

Bit 8 SC1en0 (screen 1 enable 0)
0: L0 is not included in screen 1.
1: L0 is included in screen 1.

Bit 9 SC1en1 (screen 1 enable 1)
0: L1 is not included in screen 1.
1: L1 is included in screen 1.

Bit 13 SC1en5 (screen 1 enable 5)
0: L5 is not included in screen 1.
1: L5 is included in screen 1.

Bit 14 SC1en6 (screen 1 enable 6)
0: Cursor 0 is not included in screen 1.
1: Cursor 0 is included in screen 1.

Bit 15 SC1en7 (screen 1 enable 7)
0: Cursor 1 is not included in screen 1.
1: Cursor 1 is included in screen 1.

Bit 16 SC0Xen0 (screen 0 extend enable 0)
0: L6 is not included in screen 0.
1: L6 is included in screen 0.

Bit 17 SC0Xen1 (screen 0 extend enable 1)
0: L7 is not included in screen 0.
1: L7 is included in screen 0.

Bit 20 SC1Xen0 (screen 1 extend enable 0)
0: L6 is not included in screen 1.
1: L6 is included in screen 1.

Bit 21 SC1Xen1 (screen 1 extend enable 1)
0: L7 is not included in screen 1.
1: L7 is included in screen 1.

Bit 31 MDen (multi display enable)
Enables dual (multi) display mode.
0: Single display mode
1: Dual display mode

DBGC (Display Background Color)

Register address	DisplayBaseAddress + 0x184																													
Bit No.	31	30	29	-----	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						DBGR								DBGG								DBGB							
R/W	RW0		R0				RW								RW								RW							
Initial value	0		0				0								0								0							

Specifies a color displayed for an area other than the display area in the screen for each layer.

Bit 7-0 DBGB (Display Background Blue)
Specifies the blue level background color.

Bit 15-8 DBGG (Display Background Green)
Specifies the green level background color.

Bit 23-16 DBGR(Display Background Red)
Specifies the red level background color.

L0BLD (L0 Blend)

Register address	DisplayBaseAddress + 0xB4																			
Bit No.	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L0BE	L0BS	L0BI	L0BP	L0ID	L0AL		L0AS	resv	L0BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L0 layer. This register corresponds to BRATIO/BMODE for existing products.

Bit 7-0 L0BR (L0-layer Blend Ratio)
Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L0AS (L0-layer Alpha Select)
elects an alpha layer. This selection bit is common to all layers. When L0AS=1, the LnAS bit for other layers is also regarded as 1.
0 Treats L5 layer as the alpha layer.
1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L0AL (L0-layer Alpha Layer)
Selects an alpha dedicated layer.
00 Treats LA0 as the alpha layer.
01 Treats LA1 as the alpha layer.
10 Treats LA2 as the alpha layer.
11 Treats LA3 as the alpha layer.

Bit 12 L0ID (L0-layer Ignore Data)
Specifies whether or not the A field of display data affects.
0 Only performs blend when the A field of display data is 1.
1 Ignores the A field of display data.

Bit 13 L0BP (L0-layer Blend Plane)
Selects whether a constant value or an alpha layer is used as the blend ratio.
0 Uses the L0BR value as the blend ratio.
1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L0BI (L0-layer Blend Increment)
Selects whether or not 1/256 is added when the blend ratio is not 0.
0 Does not add 1/256.
1 Adds 1/256.

Bit 15 L0BS (L0-layer Brend Select)
Selects a blend operation formula.
0 Upper image × blend ratio + lower image × (1 – blend ratio)
1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L0BE (L0-layer Blend Enable)
Enables blend
0 Performs superimposition that uses transparent color.
1 Performs superimposition that uses blend.

When performing blend, L0BE must specify blend mode and also the A field of L0 layer display data must be 1. When L0ID=0, the A field of L0 layer display data is ignored.

L1BLD (L1 Blend)

Register address	DisplayBaseAddress + 0x188																			
Bit No.	31	30 : 18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve			L1BE	L1BS	L1BI	L1BP	L1ID	L1AF		L1AS	resv	L1BR							
R/W	R0			RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0			0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L1 layer.

Bit 7-0 L1BR (L1-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L1AS (L1-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L1AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L1AL (L1-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L1ID (L1-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L1BP (L1-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L1BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L1BI (L1-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L1BS (L1-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L1BE (L1-layer Blend Enable)

Enables blend

0 Performs superimposition that uses transparent color.

1 Performs superimposition that uses blend.

When performing blend, L1BE must specify blend mode and also the A field of L1 layer display data must be 1. When L1ID=0, the A field of L1 layer display data is ignored.

L2BLD (L2 Blend)

Register address	DisplayBaseAddress + 0x18C																			
Bit No.	31	30	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L2BE	L2BS	L2BI	L2BP	L2ID	L2AF		L2AS	resv	L2BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L2 layer.

Bit 7-0 L2BR (L2-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit9 L2AS (L2-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L2AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L2AL (L2-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L2ID (L2-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L2BP (L2-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L2BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L2BI (L2-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L2BS (L2-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L2BE (L2-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color.

1 Performs superimposition that uses blend.

When performing blend, L2BE must specify blend mode and also the A field of L2 layer display data must be 1. When L2ID=0, the A field of L2 layer display data is ignored.

L3BLD (L3 Blend)

Register address	DisplayBaseAddress + 0x190																			
Bit No.	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L3BE	L3BS	L3BI	L3BP	L3ID	L3AF		L3AS	resv	L3BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L3 layer.

Bit 7-0 L3BR (L3-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L3AS (L3-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L3AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L3AL (L3-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L3ID (L3-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L3BP (L3-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L3BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit14 L3BI (L3-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L3BS (L3-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L3BE (L3-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color

1 Performs superimposition that uses blend.

When performing blend, L3BE must specify blend mode and also the A field of L3 layer display data must be 1. When L3ID=0, the A field of L3 layer display data is ignored.

L4BLD (L4 Blend)

Register address	DisplayBaseAddress + 0x194																			
Bit No.	31	30	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L4BE	L4BS	L4BI	L4BP	L4ID	L4AF		L4AS	resv	L4BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L4 layer.

Bit 7-0 L4BR (L4-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L4AS (L4-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L4AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L4AL (L4-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L4ID (L4-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L4BP (L4-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L4BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L4BI (L4-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256

Bit 15 L4BS (L4-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L4BE (L4-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color

1 Performs superimposition that uses blend.

When performing blend, L4BE must specify blend mode and also the A field of L4 layer display data must be 1. When L4ID=0, the A field of L4 layer display data is ignored.

L5BLD (L5 Blend)

Register address	DisplayBaseAddress + 0x198																			
Bit No.	31	30	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L5BE	L5BS	L5BI	L5BP	L5ID	L5AF		L5AS	resv	L5BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L5 layer.

Bit 7-0 L5BR (L5-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L5AS (L5-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L5AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L5AL (L5-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L5ID (L5-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L5BP (L5-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L5BR value as the blend ratio.

1 Uses the pixel of LA0 to LA3 layers as the blend ratio.

Bit 14 L5BI (L5-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L5BS (L5-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L5BE (L5-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color.

1 Performs superimposition that uses blend.

When performing blend, L5BE must specify blend mode and also the A field of L5 layer display data must be 1. When L5ID=0, the A field of L5 layer display data is ignored.

L6BLD (L6 Blend)

Register address	DisplayBaseAddress + 0x1990																			
Bit No.	31	30	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L6BE	L6BS	L6BI	L6BP	L6ID	L6AF		L6AS	resv	L6BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L6 layer.

Bit 7-0 L6BR (L6-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L6AS (L6-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L6AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L6AL (L6-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L6ID (L6-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit13 L6BP (L6-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L5BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L6BI (L6-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L6BS(L6-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L6BE (L6-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color.

1 Performs superimposition that uses blend.

When performing blend, L6BE must specify blend mode and also the A field of L6 layer display data must be 1. When L6ID=0, the A field of L6 layer display data is ignored.

L7BLD (L7 Blend)

Register address	DisplayBaseAddress + 0x1994																			
Bit No.	31	30	...	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve					L7BE	L7BS	L7BI	L7BP	L7ID	L7AF		L7AS	resv	L7BR							
R/W	R0					RW	RW	RW	RW	RW	RW		RW	R0	RW							
Initial value	0					0	0	0	0	0	0		0	0	0							

Specifies the blend parameters of L7 layer.

Bit 7-0 L7BR (L7-layer Blend Ratio)

Sets the blend ratio. Basically, set value/256 is the blend ratio.

Bit 9 L7AS (L7-layer Alpha Select)

Selects an alpha layer. This selection bit is common to all layers. When L0AS=1, the L7AS bit is also regarded as 1.

0 Treats L5 layer as the alpha layer.

1 Treats LA0 to LA3 layers as the alpha layers.

Bit 11-10 L7AL (L7-layer Alpha Layer)

Selects an alpha dedicated layer.

00 Treats LA0 as the alpha layer.

01 Treats LA1 as the alpha layer.

10 Treats LA2 as the alpha layer.

11 Treats LA3 as the alpha layer.

Bit 12 L7ID (L7-layer Ignore Data)

Specifies whether or not the A field of display data affects.

0 Only performs blend when the A field of display data is 1.

1 Ignores the A field of display data.

Bit 13 L7BP (L7-layer Blend Plane)

Selects whether a constant value or an alpha layer is used as the blend ratio.

0 Uses the L5BR value as the blend ratio.

1 Uses the pixel of L5 layer or of LA0 to LA3 layers as the blend ratio.

Bit 14 L7BI (L7-layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not 0.

0 Does not add 1/256.

1 Adds 1/256.

Bit 15 L7BS (L7-layer Brend Select)

Selects a blend operation formula.

0 Upper image × blend ratio + lower image × (1 – blend ratio)

1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit 16 L7BE (L7-layer Blend Enable)

Enables blend.

0 Performs superimposition that uses transparent color.

1 Performs superimposition that uses blend.

When performing blend, L7BE must specify blend mode and also the A field of L7 layer display data must be 1. When L7ID=0, the A field of L7 layer display data is ignored.

L0TC (L0-layer Transparency Control)

Register address	DisplayBaseAddress + 0xBC															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L0ZT	L0TC														
R/W	RW	RW														
Initial value	0	0														

Sets the L0 layer transparent color. In blend mode also, the color set by this register is displayed as transparent. When L0TC=0 and L0ZT=0, color 0 is displayed as black (opaque). This register corresponds to the CTC register for traditional products.

Bit 14-0 L0TC (L0-layer Transparent Color)

Sets the color value (code) displayed as transparent color for L0 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 15 L0ZT (L0-layer Zero Transparency)

Sets treatment of color value (code) 0 for C layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L2TC (L2-layer Transparency Control)

Register address	DisplayBaseAddress + 0xC0															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	L2ZT	L2TC														
R/W	RW	RW														
Initial value	0	0														

Sets the L2 layer transparent color. When L2TC=0 and L2ZT=0, color 0 is displayed as black (opaque). This register corresponds to the MLTC register for traditional products

Bit 30-16 L2TC (L2-layer Transparent Color)

Sets the color value (code) displayed as transparent color for L2 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L2ZT (L2-layer Zero Transparency)

Sets treatment of color value (code) 0 for L2 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L3TC (L3-layer Transparency Control)

Register address	DisplayBaseAddress + 0xC0															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L3ZT	L3TC														
R/W	RW	RW														
Initial value	0	0														

Sets the L3 layer transparent color. When L3TC=0 and L3ZT=0, color 0 is displayed as black (opaque). This register corresponds to the MRTC register for traditional products.

Bit 14-0 L3TC (L3-layer Transparent Color)

Sets the color value (code) displayed as transparent color for L3 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 15 L3ZT (L3-layer Zero Transparency)

Sets treatment of color value (code) 0 for L3 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L0ETC (L0-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1A0																											
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L0EZT	Reserve					L0TEC																							
R/W	RW	R0					RW																							
Initial value	0						0																							

Sets the L0 layer transparent color. 24 bits/pixel transparent color is set using this register. Lower 15 bits are physically the same as L0TC. Also, L0EZT is physically the same as L0ZT.

When L0ETC=0 and L0EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L0ETC (L0-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L0 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L0EZT (L0-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L0 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L1ETC (L1-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1A4																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L1EZT	Reserve					L1TEC																							
R/W	RW	R0					RW																							
Initial value	0	0					0																							

Sets the L1 layer transparent color. When L1ETC=0 and L1EZT=0, color 0 is displayed as black (opaque). For YCbCr display, whether or not color data is transparent is not determined, being always processed as opaque.

Bit 23-0 L1ETC (L1-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L1 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L1EZT (L1-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L1 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L2ETC (L2-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1A8																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L2EZT	Reserve					L2TEC																							
R/W	RW	R0					RW																							
Initial value	0	0					0																							

Sets the L2 layer transparent color. 24 bits/pixel transparent color is set using this register. Lower 15 bits are physically the same as L2TC. Also, L2EZT is physically the same as L2ZT.

When L2ETC=0 and L2EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L2ETC (L2-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L2 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L2EZT (L2-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L2 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L3ETC (L3-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1AC																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L3EZT	Reserve					L3TEC																							
R/W	RW	R0					RW																							
Initial value	0	0					0																							

Sets the L3 layer transparent color. 24 bits/pixel transparent color is set using this register. Lower 15 bits are physically the same as L3TC. Also, L3EZT is physically the same as L3ZT.

When L3ETC=0 and L3EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L3ETC (L3-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L3 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L3EZT (L3-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L3 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L4ETC (L4-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1B0																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L4EZT	Reserve					L4ETC																							
R/W	RW	R0					RW																							
Initial value	0	0					0																							

Sets the L4 layer transparent color. When L4ETC=0 and L4EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L4ETC (L4-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L4 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L4EZT (L4-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L4 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L5ETC (L5-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1B4																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L5EZT	Reserve					L5ETC																									
R/W	RW	R0					RW																									
Initial value	0	0					0																									

Sets the L5 layer transparent color. When L5ETC=0 and L5EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L5ETC (L5-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L5 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L5EZT (L5-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L5 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L6ETC (L6-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x1998																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L6 EZT	Reserve					L67ETC																									
R/W	RW	R0					RW																									
Initial value	0	0					0																									

Sets the L6 layer transparent color. When L6ETC=0 and L6EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L6ETC (L6-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L5 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L6EZT (L6-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L6 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

L7ETC (L7-layer Extend Transparency Control)

Register address	DisplayBaseAddress + 0x199C																															
Bit No.	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L7 EZT	Reserve													L7ETC																	
R/W	RW	R0													RW																	
Initial value	0	0													0																	

Sets the L7 layer transparent color. When L7ETC=0 and L7EZT=0, color 0 is displayed as black (opaque).

Bit 23-0 L7ETC (L7-layer Extend Transparent Color)

Sets the color value (code) displayed as transparent color for L7 layer. For index color mode (8 bits/pixel), bits 7 to 0 are used.

Bit 31 L7EZT (L5-layer Extend Zero Transparency)

Sets treatment of color value (code) 0 for L7 layer.

0 Does not treat code 0 as transparent.

1 Treats code 0 as transparent.

CKC (Chroma Key Control)

Register address	DisplayBaseAddress + 0xB8																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															KCS	KYEN	KYC														
R/W	R0															RW	RW	RW														
Initial value	0															0	0	Undefined														

Bit 14-0 KYC (Key Color)

Sets the key color used to perform chroma-key processing. When index color mode (8 bits/pixel) is established and chroma-key mode is set to C layer color, bits 7 to 0 are used.

Bit 15 KYEN (chroma-Key Enable)

Sets whether or not to perform chroma-key processing.

0 Performs no chroma-key processing (the GV pin always outputs H).

1 Performs chroma-key processing.

Bit 16 KCS (Key Color Select)

Selects whether display color or C layer color is used as the key color used to perform chroma-key processing.

0 Uses display color as the key color.

1 Uses C layer color as the key color.

L1YCR0 (L1 layer YC to Red coefficient 0)

Register address	DisplayBaseAddress + 0x1E0			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	a12	Reserved	a11
R/W	R0	RW	R0	RW
Initial value	0	000 0000 0000	0	001 0010 1011

This register defines the parameter for the red component at YCbCr/RGB conversion.

Bit 10-0 a11

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 26-16 a12

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

L1YCR1 (L1 layer YC to Red coefficient 1)

Register address	DisplayBaseAddress + 0x1E4			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	b1	Reserved	a13
R/W	R0	RW	R0	RW
Initial value	0	1 1111 0000	0	001 1001 1000

This register defines the parameter for the red component at YCbCr/RGB conversion time.

Bit 10-0 a13

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 24-16 b1

9-bit signed integer. The value is complement representation for 2.

L1YCG0 (L1 layer YC to Green coefficient 0)

Register address	DisplayBaseAddress + 0x1E8			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3 2 1 0
Bit field name	Reserved	a22	Reserved	a21
R/W	R0	RW	R0	RW
Initial value	0	111 1001 1100	0	001 0010 1011

This register defines the parameter for the green component at YCbCr/RGB conversion time.

Bit 10-0 a21

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 26-16 a22

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

L1YCG1 (L1 layer YC to Green coefficient 1)

Register address	DisplayBaseAddress + 0x1EC			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3 2 1 0
Bit field name	Reserved	b2	Reserved	a23
R/W	R0	RW	R0	RW
Initial value	0	1 1111 0000	0	111 0010 1111

This register defines the parameter for the green component at YCbCr/RGB conversion time.

Bit 10-0 a23

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 24-16 b2

9-bit signed integer. The value is complement representation for 2.

L1YCB0 (L1 layer YC to Blue coefficient 0)

Register address	DisplayBaseAddress + 0x1F0			
Bit No.	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Bit field name	Reserved	a32	Reserved	a31
R/W	R0	RW	R0	RW
Initial value	0	010 0000 0100	0	001 0010 1011

This register defines the parameter for the blue component at YCbCr/RGB conversion.

Bit 10-0 a31

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 26-16 a32

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

L1YCB1 (L1 layer YC to Green coefficient 1)

Register address	DisplayBaseAddress + 0x1F4			
Bit No.	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Bit field name	Reserved	b3	Reserved	a33
R/W	R0	RW	R0	RW
Initial value	0	1 1111 0000	0	000 0000 0000

This register defines the parameter for the blue component at YCbCr/RGB conversion.

Bit 10-0 a33

11-bit signed fixed-point value. The lower 8 bits of the value are placed after the decimal point. The value is complement representation for 2.

Bit 24-16 b3

The value is a 9-bit signed integer. The value is complement representation for 2.

L0PAL0-255 (L0-layer Palette 0-255)

Register address	DisplayBaseAddress + 0x400$_H$ -- DisplayBaseAddress + 0x7FF																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A								R								G								B							
R/W	RW	R0							RW							R0	RW							R0	RW							R0
Initial value	Undefined	0000000							Undefined							00	Undefined							00	Undefined							00

This register is the color palette register for L0 layer and for the cursors. When performing display in index color mode, color data for the display frame is used as a palette register number, and the color that is set in that register is used as the pixel display color. This register corresponds to CPALn for existing products.

Bit 7-2 B (Blue)
Sets the blue component of the color.

Bit 15-10 G (Green)
Sets the green component of the color.

Bit 23-18 R (Red)
Sets the red component of the color.

Bit 31 A (Alpha)
When blend mode is enabled, specifies whether or not blend with the lower layer is performed.

0	Performs no blend even when blend mode is enabled. Performs superimposition using transparent color
1	Performs blend.

L1PAL0-255 (L1-layer Palette 0-255)

Register address	DisplayBaseAddress + 0x800 – DisplayBaseAddress + 0xBF																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A								R								G								B							
R/W	RW	R0							RW							R0	RW							R0	RW							R0
Initial value	Undefined	0000000							Undefined							00	Undefined							00	Undefined							00

This register is the color palette register for L1 layer and for the cursors. When performing display in index color mode, color data for the display frame is used as a palette register number, and the color that is set in that register is used as the pixel display color. This register corresponds to MBPALn for existing products.

Bit 7-2 B (Blue)
Sets the blue component of the color.

Bit 15-10 G (Green)
Sets the green component of the color.

Bit 23-18 R (Red)
Sets the red component of the color.

Bit 31 A (Alpha)
When blend mode is enabled, specifies whether or not blend with the lower layer is performed.

0	Performs no blend even when blend mode is enabled. Performs superimposition using transparent color.
1	Performs blend.

L2PAL0-255 (L2-layer Palette 0-255)

Register address	DisplayBaseAddress + 0x1000 -- DisplayBaseAddress + 0x13FF																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A								R								G								B							
R/W	RW	R0							RW							R0	RW							R0	RW							R0
Initial value	Undefined	0000000							Undefined							00	Undefined							00	Undefined							00

This register is the color palette register for L2 layer and for the cursors. When performing display in index color mode, color data for the display frame is used as a palette register number, and the color that is set in that register is used as the pixel display color.

Bit 7-2 B (Blue)
Sets the blue component of the color.

Bit 15-10 G (Green)
Sets the green component of the color.

Bit 23-18 R (Red)
Sets the red component of the color.

Bit 31 A (Alpha)
When blend mode is enabled, specifies whether or not blend with the lower layer is performed.

0 Performs no blend even when blend mode is enabled. Performs superimposition using transparent color.

1 Performs blend.

L3PAL0-255 (L3-layer Palette 0-255)

Register address	DisplayBaseAddress +0x1400$_H$ -- DisplayBaseAddress + 0x17FF																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A								R								G								B							
R/W	RW	R0							RW							R0	RW							R0	RW							R0
Initial value	Undefined	0000000							Undefined							00	Undefined							00	Undefined							00

This register is the color palette register for L3 layer and for the cursors. When performing display in index color mode, color data for the display frame is used as a palette register number, and the color that is set in that register is used as the pixel display color.

Bit 7-2 B (Blue)
Sets the blue component of the color.

Bit 15-10 G (Green)
Sets the green component of the color.

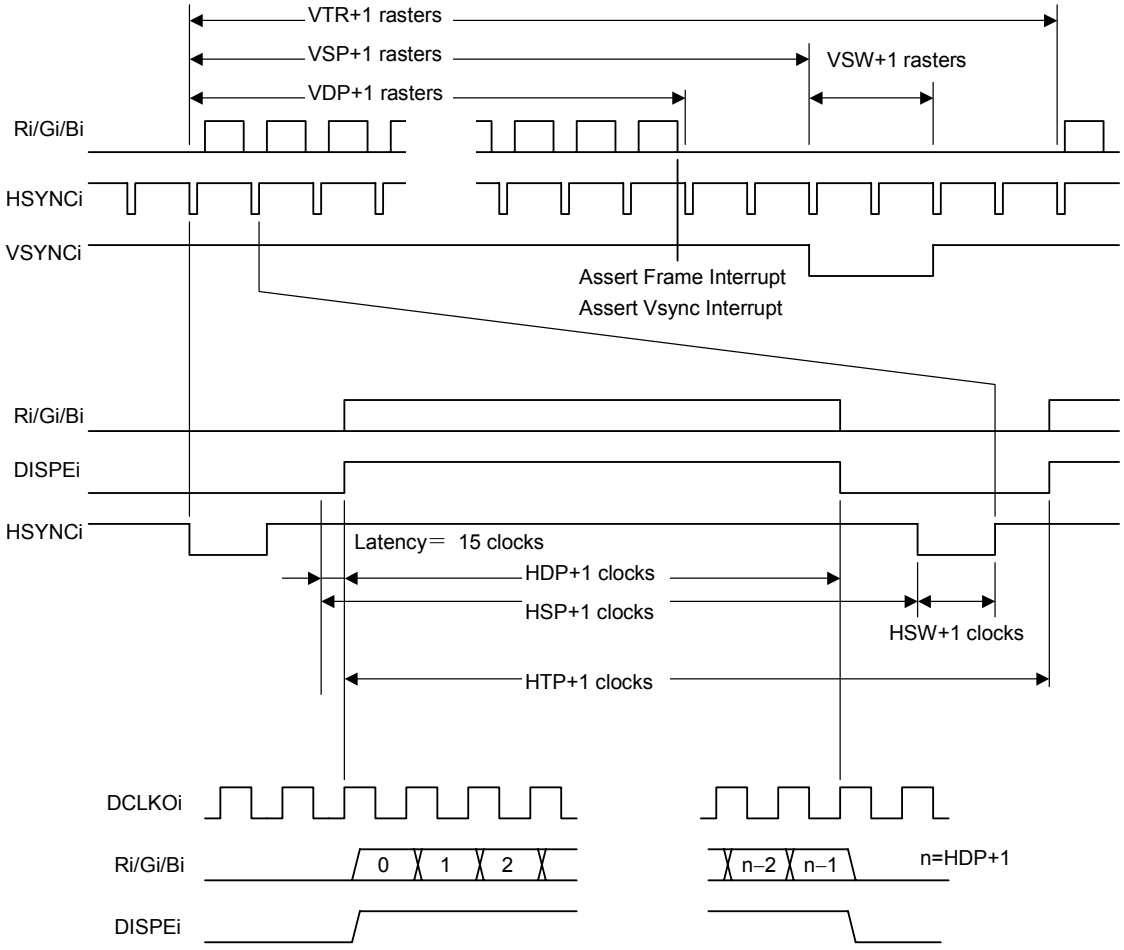
Bit 23-18 R (Red)
Sets the red component of the color.

Bit 31 A (Alpha)
When blend mode is enabled, specifies whether or not blend with the lower layer is performed.

0	Performs no blend even when blend mode is enabled. Performs superimposition using transparent color.
1	Performs blend.

7.13 Display Timing

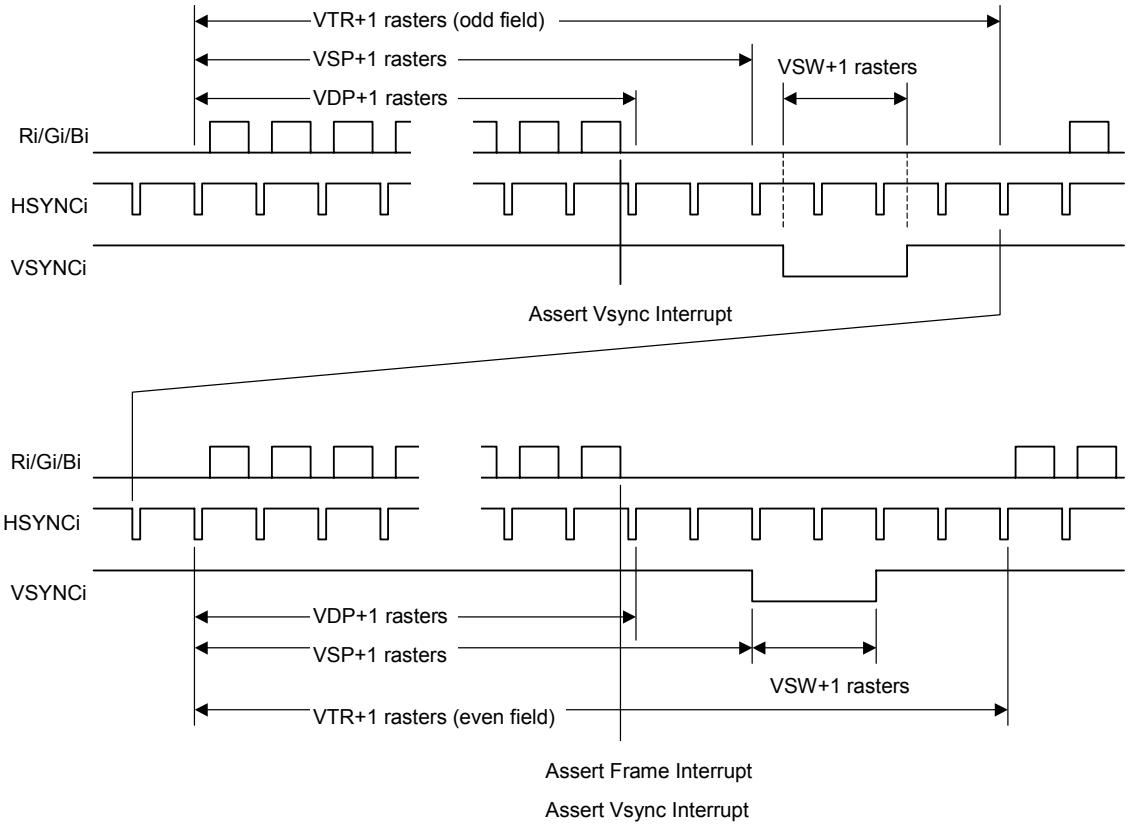
7.13.1 Noninterlace



VTR, HDP, etc., here indicate the setting value of the corresponding register.

VSYNC/frame interrupt is asserted when display of the last raster has ended. When the display parameter is updated in synchronization with the frame interrupt, display is not distorted. Since calculation of the next frame starts immediately after assertion of the vertical synchronizing pulse, update the parameter before the calculation.

7.13.2 Interlace Video

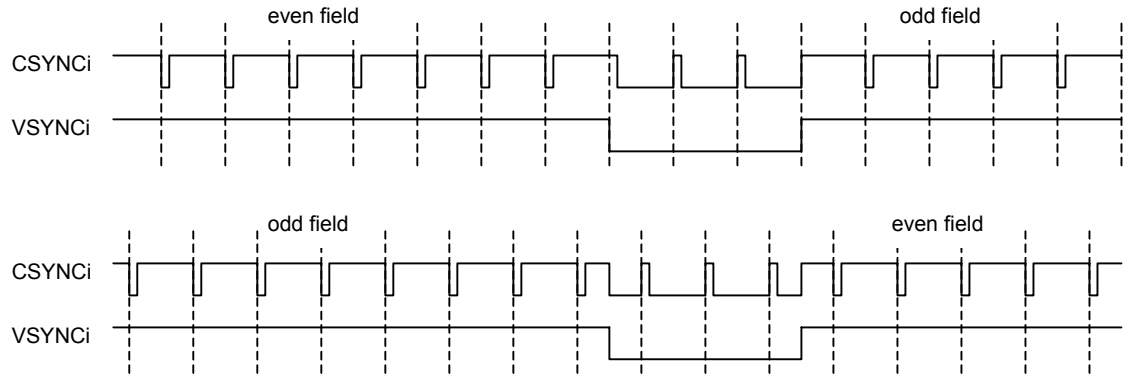


VTR, HDP, etc., here indicate the setting value of the corresponding register.

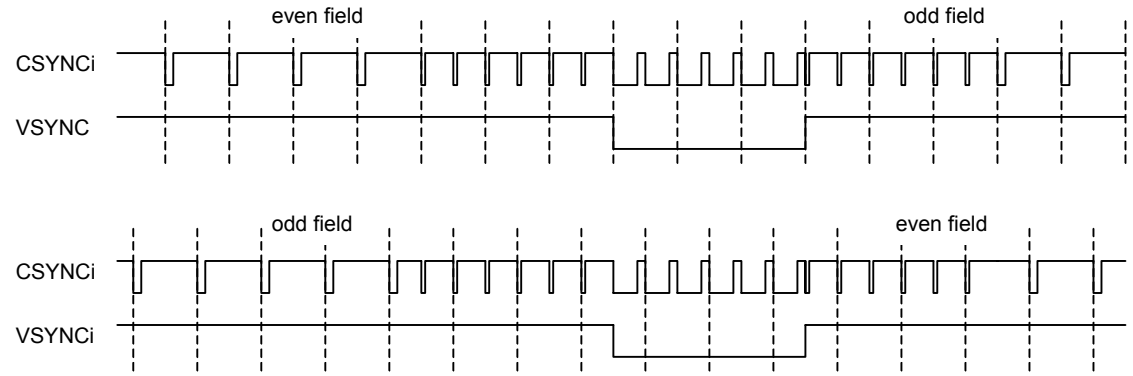
Interlace mode operates at the same timing as the noninterlace mode. The interlace mode is different from the noninterlace mode only in terms of output image data.

7.13.3 Composite Synchronization signal

When the EEQ bit of the DCM register is "0", the waveform of output of the CSYNC signal is as shown below.



When the EEQ bit of the DCM register is "1", an equalizing pulse is inserted into the CSYNC signal, resulting in the following waveform.



The equalizing pulse is inserted from when the vertical retrace period stars. Also, the equalizing pulse is inserted three times after the vertical synchronization period ended.

8 Video Capture

8.1 Video Capture Function

8.1.1 Input Data Format

– The Input data format conforms to ITU RBT-656 format (for details, see ***8.6 External Video Signal Input***). The signal mode is compatible with NTSC and PAL.

– The Input data format is compatible with digital RGB666 input.

8.1.2 Capture of Video Signal

When VIE of the Video Capture Mode register (VCM) is 1, Carmine is enabled, capturing video stream data from the video data input pins in synchronization with the CCLK clock.

8.1.3 Conversion to Non-interlace

Non-interlace display can be performed for captured video images. When performing non-interlace display, the user can select between 2 modes: BOB mode and WEAVE mode.

– BOB mode

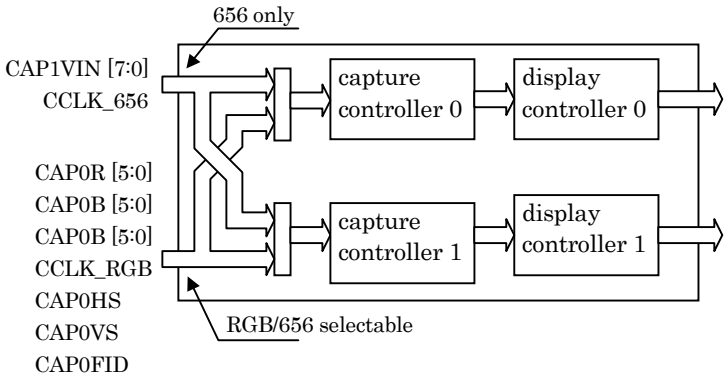
When a field is an odd field, raster of an even field is generated by average interpolation, and the raster is added to the odd field, generating 1 frame. When a field is an even field, raster of an odd field is generated by average interpolation, and the raster is added to the even field, generating 1 frame.
To select BOB mode, enable vertical interpolation by using the VI bit of the VCM (Video Capture Mode) register, and at the same time, set the L1IM bit of the L1M (L1-layer Mode) register to 0.

– WEAVE mode

An odd field and an even field are merged in the video capture buffer, to generate 1 frame. Vertical resolution is relatively high compared to BOB mode, but raster displacement is visible in motion scenes.
To select WEAVE mode, disable vertical interpolation by using the VI bit of the VCM (Video Capture Mode) register, and at the same time, set the L1IM bit of the L1M (L1-layer Mode) register to 1.

8.2 Selection of Input Port

Carmine has two independent capture sections. Input port has two systems, between which it can be switched.



Of the 2 capture input ports, one is for only 656 format, and the other can be selected between RGB format and 656 format.

When performing 656 input by using the port used both as RGB and 656, pin correspondence is as shown below.

RGB	656
CAP0G1	bit-7
CAP0G0	bit-6
CAP0B5	bit-5
CAP0B4	bit-4
CAP0B3	bit-3
CAP0B2	bit-2
CAP0B1	bit-1
CAP0B0	bit-0

Whether to use RGB or 656 is selected using the VIS bit and VF bit of the VCM register.

Input port is selected using the Csel0/1 bit of the VCCC (Video/Capture Common Control) register.

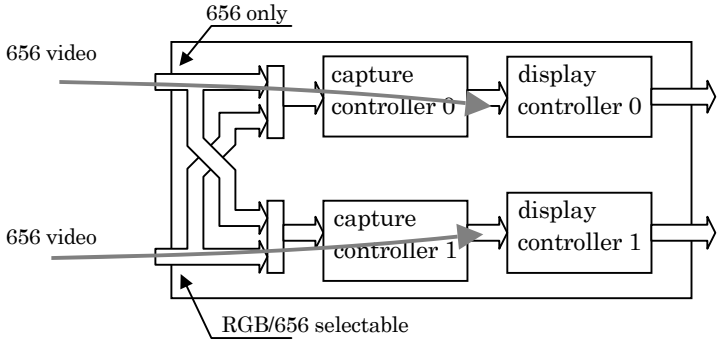
Csel0 = 0: 656 input → Capture 0

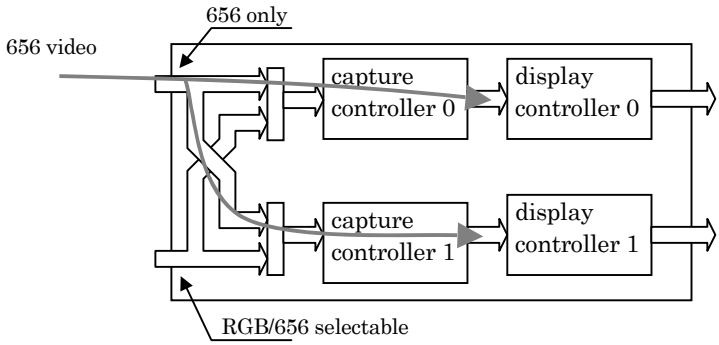
Csel0 = 1: RGB/656 input → Capture 0

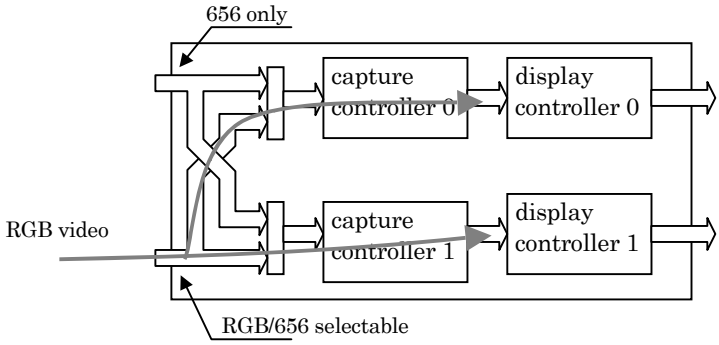
Csel1 = 0: 656 input → Capture 1

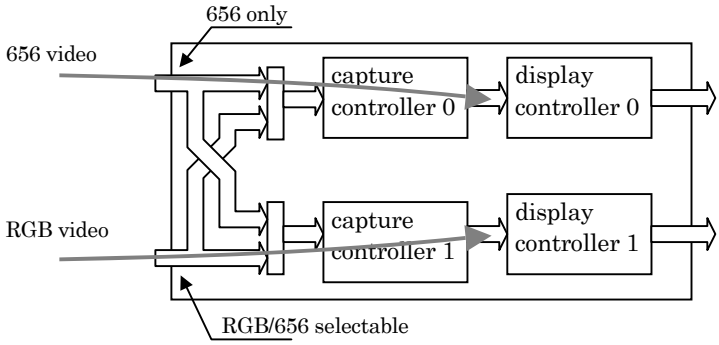
Csel1 = 1: RGB/656 input → Capture 1

Selection examples are shown below.


656 only
656 video
capture controller 0
display controller 0
656 video
capture controller 1
display controller 1
RGB/656 selectable


656 only
656 video
capture controller 0
display controller 0
capture controller 1
display controller 1
RGB/656 selectable


656 only
capture controller 0
display controller 0
RGB video
capture controller 1
display controller 1
RGB/656 selectable


656 only
656 video
capture controller 0
display controller 0
RGB video
capture controller 1
display controller 1
RGB/656 selectable

8.3 Video Buffer

8.3.1 Data Format

Data is stored in the capture buffer, basically in the 16 bits/pixel format. 2 color components (Cb, Cr) are respectively half the resolution of luma data (Y component) in the horizontal direction, resulting in being expressed in the 16 bits/pixel format. In L1 layer, data is converted to the RGB format and displayed.

format	31	30	…	25	24	23	22	…	17	16	15	14	…	9	8	7	6	…	1	0
YCbCr 16 bit/pixel	Y					Cr					Y					Cb				

Data can also be stored in the RGB format, assuming that the drawing section uses data as texture. There exist the following formats.

format	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGB 16 bits/pixel	A	R					G					B				
RGBA 16 bits/pixel	R					G					B					A

format	31	30	…	25	24	23	22	…	17	16	15	14	…	9	8	7	6	…	1	0
ARGB 24 bits/pixel	A					R					G					B				
RGBA 24 bits/pixel	R					G					B					A				

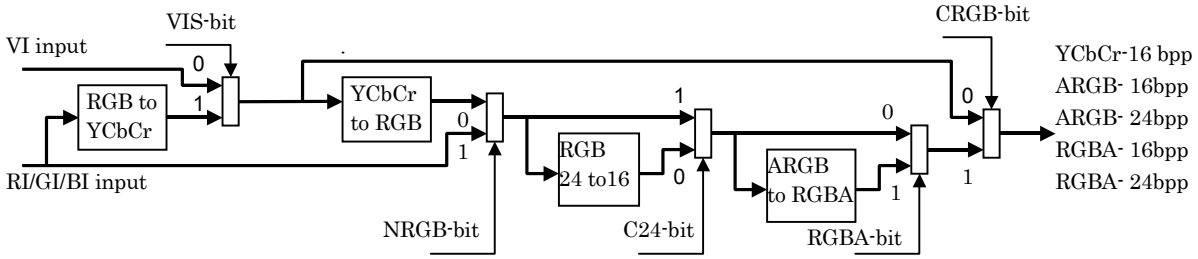
The relation between the capture data formats and the data format control bits in registers is as shown below.

NRGB	CRGB	C24	RGBA	Capture data format	Enlarged display
0	0	0	0	YcbCr 16 bits/pixel	○
0	0	0	1	Unused	
0	0	1	x	Unused	
0	1	0	0	ARGB 16 bits/pixel (YCbCr → RGB conversion)	
0	1	0	1	RGBA 16 bits/pixel (YCbCr → RGB conversion)	
0	1	1	0	ARGB 24 bits/pixel (YCbCr → RGB conversion)	
0	1	1	1	RGBA 24 bits/pixel (YCbCr → RGB conversion)	
1	0	0	0	ARGB 16 bits/pixel	○
1	0	0	1	RGBA 16 bits/pixel	
1	0	1	0	ARGB 24 bits/pixel	○
1	0	1	1	RGBA 24 bits/pixel	
1	1	x	x	Unused	

Data formats added from Coral-PA is shown shaded.
○ in the rightmost column of the above table indicates that enlarged display of capture data is supported.

"Unused" above means writing data which is meaningless as image data.
The NRGB bit is bit2 of the VCM register. The other bits are in the CBM register.

Selection of data format of capture image is shown below diagrammatically.



For the ARGB or RGBA format, whether to set the blend bit to 1 or 0 at pixel write time can be selected using the BLEN bit of the CBM register.

BLEN = 0 (blend bit = 0) (for ARGB, BLEN is MSB; for RGBA, BLEN is LSB)

BLEN = 1 (blend bit = 1) (for ARGB, BLEN is MSB; for RGBA, BLEN is LSB)

8.3.2 Synchronization Control

Write of video image data to graphics memory and scan to display the data are performed independently. Graphics memory for video capture is controlled using the ring buffer mode, and when image data equivalent to 1 frame is prepared in memory, the frame is displayed.

When the frame rate of video capture and that of display are different, frame dropping occurs or continuous display of the same frame occurs.

8.3.3 Area Allocation

Allocate an area for about 2.2 frames as the video capture buffer. This size is equivalent to the double buffer margin for the frame. Set the area's starting address and upper bound address in the CBOA and CBLA registers, respectively. The upper bound address specifies the raster starting position.

When allocating n rasters as the video capture buffer, set the upper bound address as follows:

$$CBLA = CBOA + 64 (n - 2) \times CBW$$

The starting address of the (n + 1)-th raster is CBOA + 64n × CBW (= CBLA + 64 × 2 × CBW), and the address range of the capture buffer area is as shown below.

$$CBOA \leq Address < CBOA + 64n \times CBW$$

When performing reduced display, allocate the buffer area using the reduced frame size.

8.3.4 Window Display

Captured video images are displayed using L1 layer. The whole or a part of a captured image can be displayed as the whole or a window of a screen.

When performing capture display, set L1 layer in capture synchronizing mode (L1CS = 1). In this mode, L1 layer displays the most recent frame that is in the video capture buffer. The display address used in normal mode is ignored.

The stride of L1 layer must match that of the video capture buffer. When they do not match, slanted, distorted images are displayed.

Make the display size of L1 layer the same as the reduced video capture image size. When the display size of L1 layer is made greater than the reduced video capture image size, invalid (= ineffective) data is displayed.

For L1 layer, the user can select between RGB display and YCbCr display, but when performing video capture, select the YCbCr format (L1YC = 1).

8.3.5 Interlace Display

It is possible to perform interlace display for images captured into the video capture buffer in WEAVE mode. To perform the interlace display, enable WEAVE mode and select interlace & video display for display scan.

However, when display scan is asynchronous, flicker occurs in motion scenes. To prevent flicker, set the OO (Odd Only) bit of the CBM (Capture Buffer Mode) register to 1.

8.4 Scaling

8.4.1 Video Reduction Function

When CM of the video capture mode register (VCM) is 11, Carmine reduces the video screen. A reduction scale is set for vertical direction in units of lines and for horizontal direction in units of 2 pixels; it can be set for vertical direction and horizontal direction independently. The set value for a reduction scale is defined as an input/output value, and is a 16-bit fixed decimal consisting of a 5-bit integer part and a 11-bit decimal part. The valid (= effective) set value is 0800_H to $FFFF_H$. Set the scale for vertical direction using bit 31-16 of the capture scale register (CSC), and set the scale for horizontal direction using bit 15-00. The initial value of this register is 08000800_H (the scale factor is 1). A sample calculation expression to set the scale is shown below.

Reduction in vertical direction:

576 ==> 490 lines 576/490 = 1.176
1.176 × 2048 = 2408 => 0968_H

Reduction in horizontal direction:

720 ==> 648 pixels 720/648 = 1.111
1.111 × 2048 = 2275 => $08E3_H$

Therefore, CSC is set to $096808E3_H$.

The capture horizontal pixel register (CHP) limits the pixel count during scaling processing, and is not a set value for scaling. The register performs clamp processing for video streaming data exceeding the value set for CHP. Normally, the register can be used with the initial value set to it.

8.4.2 Video Expansion Function

Carmine can expand a video screen independently in horizontal direction and vertical direction. This function can be used when, for example, displaying video stream input whose display resolution is less than the one to be used, on the full screen. Also, this function can be used to expand (zoom in) a part of video stream input. The initialization procedure is shown below.

- Set the magnify flag of the L1-layer mode register of the display controller.
- Set the size of the pre-expansion source image to CMSHP and CMSVL.
- Set the size of the post-expansion output image to CMDHP and CMDVL.

As a sample setting, setting of each register in the following case is shown.

Size of input image: 480 × 360 pixels

Size of display image: 640 × 480 pixels

HSCALE = (480/640)*2048=0x0600
VSCALE = (360/480)*2048=0x0600
CMSHP = 0x00f0
CMSVL = 0x0168
CMDHP = 0x0140
CMDVL = 0x01e0
L1WW = 0x0280
L1WH = 0x01df

Note1: When switching from expansion to reduction by using the display size change function, insert a setting to keep scale factor 1 (CSC=08000800h) for 2V period (vertical synchronization of display) or longer, before the switching. This is a restriction due to the fact that the complementary filter is shared by reduction and expansion. When switching from reduction to expansion, there is no restriction.

Note2: When reduced display and enlarged display are performed using scale factor continuously, images are somewhat distorted. This is due to the video capture function system of Carmine.

8.4.3 Image Processing Flow

Image processing of capture images displayed in the L1 layer window is performed in the following sequence.

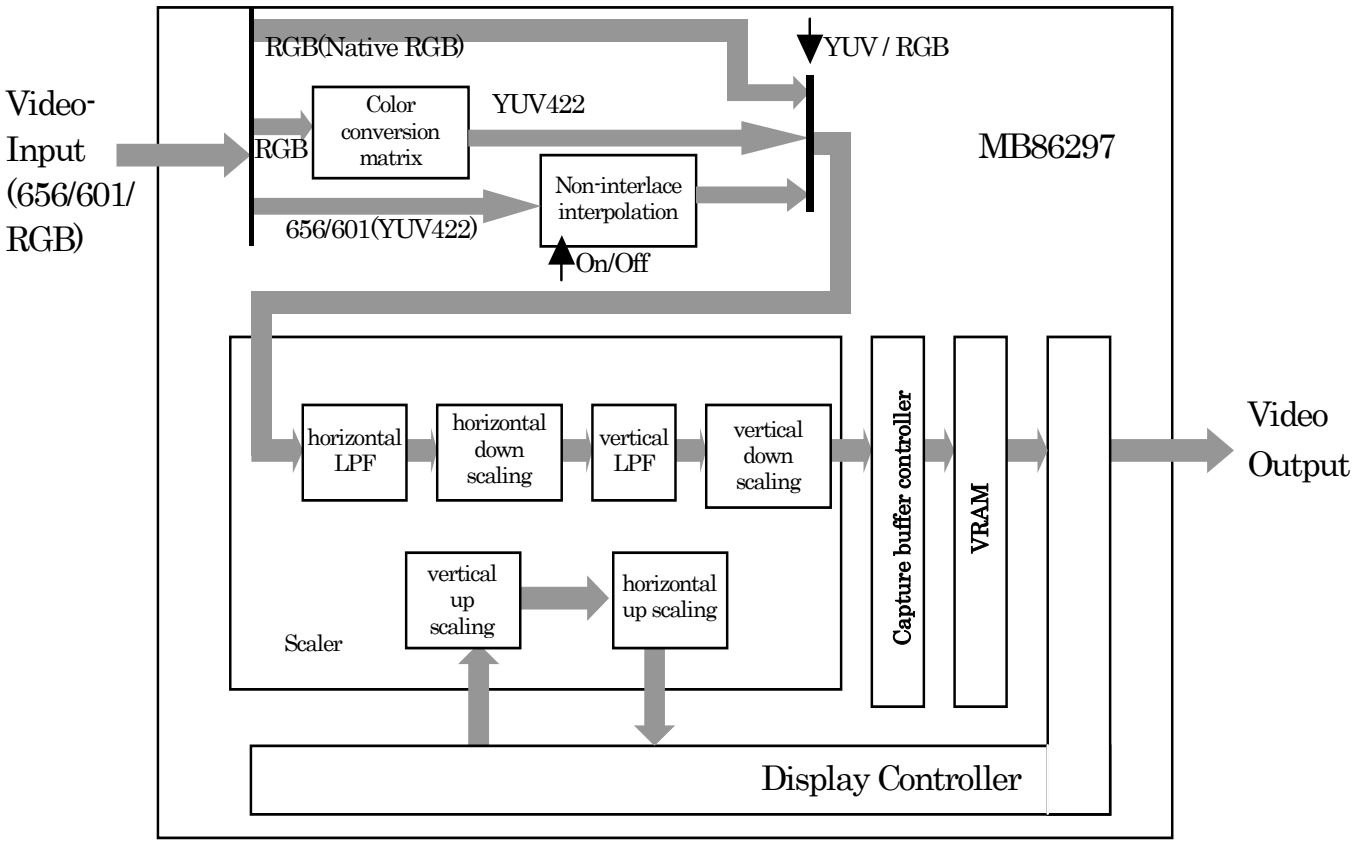


Fig. 8.1 Image Processing Flow

- Non-interlace interpolation processing

When VI of the video capture mode register (VCM) is 0, the interlace screen is vertically interpolated using data in the same field. The screen is doubled vertically. When VI is 1, no vertical interpolation is performed.

- Horizontal low-pass filter processing

Low-pass filter processing can be performed horizontally as pre-processing for reducing images horizontally. The horizontal low-pass filter (horizontal LPF) can be set to ON/OFF regardless of image expansion or reduction in horizontal direction.
The horizontal low-pass filter is a 5-tap FIR filter. Coefficients are specified using the following registers:

CHLPF_Y: Horizontal LPF luma component coefficient code

CHLPF_C: Horizontal LPF chroma component coefficient code

The coefficient for the luma (Y) signal and chroma (C) signal is specified independently using 2-bit coefficient code. The coefficient is symmetric.

CHLPF_x	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	8/32	10/32	3/32

When coefficient code "00" is set, the horizontal LPF is set to OFF (through).

Note : In native RGB mode (NRGB=1), only the coefficient code specified for CHLPF_Y is enabled.

- Reduction/expansion processing in horizontal direction

To perform reduction/expansion processing in horizontal direction, perform setting for bit 15-00 of the capture scale register (CSC).
Reduction in horizontal direction is performed before writing to VRAM. Expansion in horizontal direction is performed after reading from VRAM.
Interpolation filter processing for luma (Y) signal is performed using the Cubic Interpolate method; interpolation filter processing for chroma (C) signal performed using the BiLinear Interpolate method.

- Vertical low-pass filter processing

Low-pass filter processing can be performed vertically as pre-processing for reducing images vertically. The vertical low-pass filter (vertical LPF) can be set to ON regardless of image expansion or reduction in vertical direction.
The vertical low-pass filter is a 3-tap FIR filter. Coefficients are specified using the following registers:

CVLPF_Y: Vertical LPF luma component coefficient code

CVLPF_C: Vertical LPF chroma component coefficient code

The coefficient for the luma (Y) signal and chroma (C) signal is specified independently using 2-bit coefficient code. The coefficient is symmetric.

CVLPF_ x	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting is disabled.		

When coefficient code "00" is set, the vertical LPF is set to OFF (through).

Note: In native RGB mode (NRGB=1), only the coefficient code specified for CVLPF_Y is enabled.

- Reduction/expansion processing in vertical direction

To perform reduction/expansion processing in vertical direction, perform setting for bit 31-16 of the capture scale register (CSC).
Reduction in vertical direction is performed before writing to VRAM. Expansion in vertical direction is performed after reading from VRAM.
Interpolation filter processing for luma (Y) signal is performed using the Cubic Interpolate method; interpolation filter processing for chroma (C) signal performed using the BiLinear Interpolate method.

8.5 Interrupt

8.5.1 Overview

The interrupts generated from the Carmine capture section are shown below.

(1) Error detection

This interrupt occurs due to video input error.

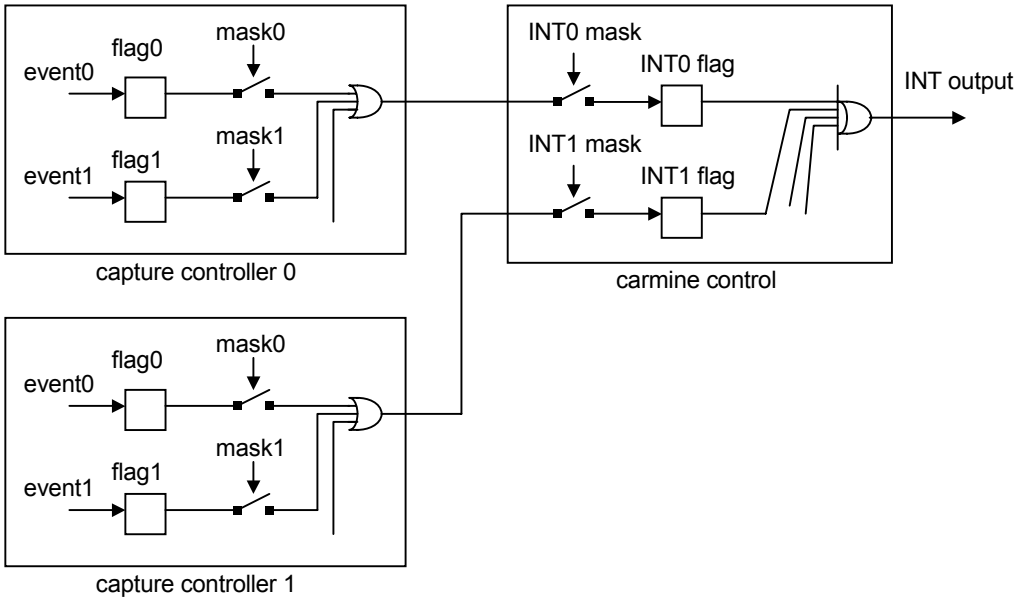
(2) Capture VSYNC

This interrupt occurs in synchronization with the video input vsync signal.

8.5.2 Interrupt Status

The interrupt status is controlled at low two steps. The interrupt of the capture controller occurs in multiple events, and these events are brought together into one signal and conveyed to the status flag of Carmine control. Events can be ignored by the mask setting.

When detecting events by polling, turn OFF propagation by the mask setting and read the status flag in the capture controller. The flag value can be read without being affected by the mask setting.



When the interrupt handler recognizes an interrupt event in the capture controller, you must first clear the Carmine control status flag and then the capture controller status flag.

8.5.3 Error Detection

An error occurs when an expected control code or synchronization signal cannot be detected in the input video data. The status register in the corresponding capture controller is the SYNC_err register.

8.5.4 Capture VSYNC interrupt

This interrupt notifies the VSYNC timing of capture. The status register in the corresponding capture controller is the CINT register.

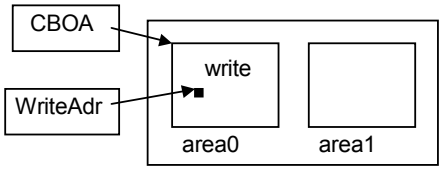
The following uses are assumed for this interrupt:

(1) Detection of valid image input
(2) Frame management by host CPU

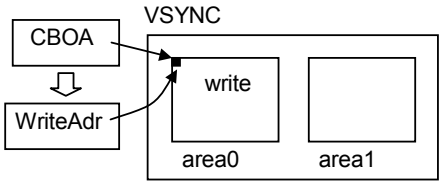
The single buffer mode is used when the host CPU manages frame. The CBOA and CBLA registers that specify the starting address of the buffer can be rewritten when the VSYNC interrupt occurs.

The frame immediately after the interrupt has occurred is written to the address in effect before the update is performed. A new address becomes effective from the next frame. In other words, the captured image at the previous address can be used as a still image.

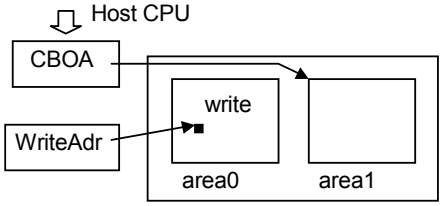
The following shows the state in which writing is being performed to area0. The starting address of area0 is held by the CBOA register.



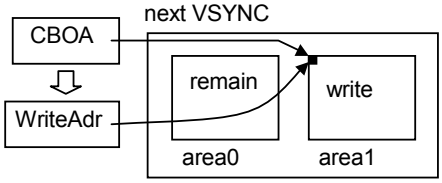
The following shows that the write address moves to the starting address of area0 in synchronization with VSYNC. A VSYNC interrupt occurs.



The following shows that, after the VSYNC interrupt has occurred, CBOA is changed to the starting of area1 by the host CPU. Writing is unaffected until the next VSYNC occurs.



The following shows that writing to area1 is started after the next VSYNC has occurred and that the image in area0 is stored.



Note: In single-buffer operating mode, set the CBOA and CBLA registers to the same value.

8.6 External Video Signal Input

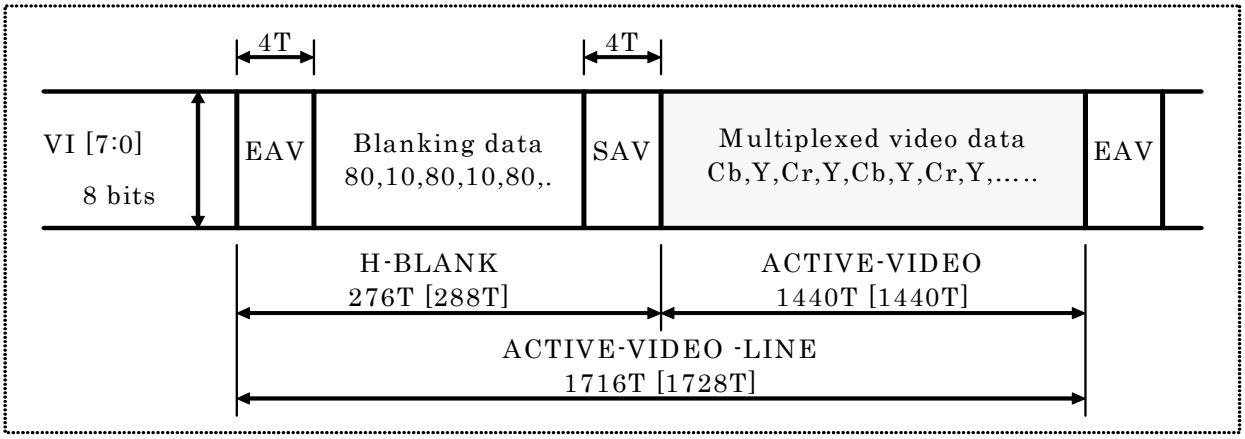
8.6.1 R.BT656YUV422 Input Format

The ITU R.BT-656 format is widely used for NTSC and PAL digital transmission. It is compatible with YUV422. Carmine can capture input interlace video signals after converting them to non-interlace ones by using linear complementarity.

In Carmine, when VIE of the video capture mode register (VCM) is 1, capture is enabled, capturing video stream data from the 8-bit VI pin in synchronization with the CCLK clock. Video streams that can be processed are limited to digital video streams conforming to ITU-RBT656, meaning that they are in the YCbCr4:2:2 format with timing reference code added. Since streams are captured using timing reference code, the format is automatically made compatible with both NTSC and PAL, but to detect error in code if it occurs, set VS of VCM to NTSC or PAL. When it is set to NTSC, the data count of the capture data count register (CDCN) is referenced; when set to PAL, the data count of the capture data count register (CDCP) is referenced. If the data count does not match the stream data count, bit 4-0 of the video capture status register (VCS) is set to a value other than 0000.

8.6.1.1 R.BT656 input format CAP0VI [7:0]

Synchronizing code and image data (Cb,Y,Cr,Y) are multiplexed in synchronization with 27 MHz clock and are input as 8-bit data. Valid pixels are sent with them sandwiched between synchronizing codes SAV and EAV.



SAV: Starting code (4 bytes) of active video data

EAV: Ending code (4 bytes) of active video data

T: 27 MHz

[ ]: 625/50 system (PAL)

BLANKING PERIOD			TIMING REF-CODE				720 PIXELS YUV4:2:2 DATA									TIMING REF-CODE				BLANKING PERIOD		
…	80	10	FF	00	00	SAV	Cb0	Y0	Cr0	Y1	Cb2	Y2	…	Cr718	Y719	FF	00	00	EAV	80	10	…

8.6.1.2 Format of R.BT656 synchronizing code (4 bytes)

Word / Bit	SYNC code (fixed)			EAV/SAV
	First byte	Second byte	Third byte	Fourth byte
7	1	0	0	1 (fixed)
6	1	0	0	F 0: First field 1: Second field
5	1	0	0	V 0: ACTIVE-VIDEO 1: VBI
4	1	0	0	H 0: SAV 1: EAV
3	1	0	0	P3 Protection bit
2	1	0	0	P2 Protection bit
1	1	0	0	P1 Protection bit
0	1	0	0	P0 Protection bit

8.6.1.3 SAV/EAV timing reference signal

Bit	7	6	5	4	3	2	1	0
Function	Fixed	F	V	H	P3	P2	P1	P0
80	1	0	0	0	0	0	0	0
9D	1	0	0	1	1	1	0	1
AB	1	0	1	0	1	0	1	1
B6	1	0	1	1	0	1	1	0
C7	1	1	0	0	0	1	1	1
DA	1	1	0	1	1	0	1	0
EC	1	1	1	0	1	1	0	0
F1	1	1	1	1	0	0	0	1

80: SAV code during valid pixel period (Active-video) in the first field

9D: EAV code during valid pixel period (Active-video) in the first field

AB: SAV code during vertical flyback period in the first field

B6: EAV code during vertical flyback period in the first field

C7: SAV code during valid pixel period (Active-video) in the second field

DA: EAV code during valid pixel period (Active-video) in the second field

EC: SAV code during vertical flyback period in the second field

F1: EAV code during vertical flyback period in the second field

8.6.1.4 R.BT656 synchronizing code (EAV) timing (525/60 system)

LINE-No,	522	523	524	525	1	2	3	4	5	6	7	8	9
1st_field	ACTIVE-VIDEO				Serration pulse			Vertical synchronization			Serration pulse		
EAV	DA	DA	DA	DA	F1	F1	F1	B6	B6	B6	B6	B6	B6
F	1	1	1	1	1	1	1	0	0	0	0	0	0
V	0	0	0	0	1	1	1	1	1	1	1	1	1
LINE-No,	260	261	262	263	264	265	266	267	268	269	270	271	272
2nd_field	ACTIVE-VIDEO				Serration pulse			Vertical synchronization			Serration pulse		
EAV	9D	9D	9D	9D	B6	B6	F1	F1	F1	F1	F1	F1	F1
F	0	0	0	0	0	0	1	1	1	1	1	1	1
V	0	0	0	0	1	1	1	1	1	1	1	1	1

LINE-No,	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1st_field	VBI-lines 1st_field												Act-video	
EAV	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	9D	9D	9D	9D
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	1	1	1	1	1	1	1	0	0	0	0
LINE-No,	273	274	275	276	277	278	279	280	281	282	283	284	285	286
2nd_field	VBI-lines 2nd_field												Act-video	
EAV	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	DA	DA	DA	DA
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1	1	1	0	0	0	0

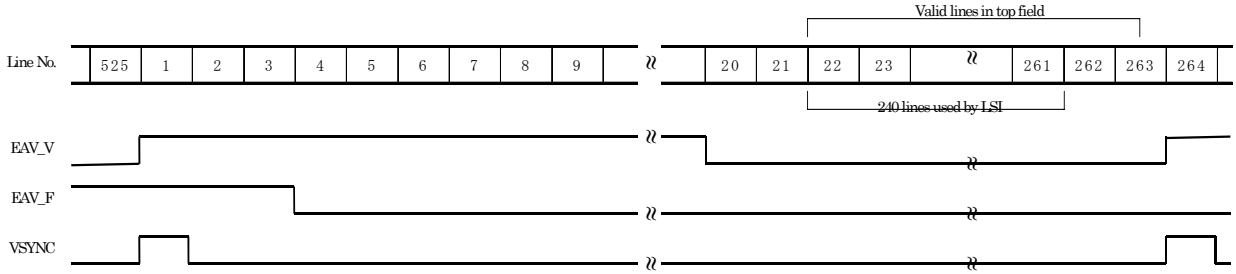
8.6.1.5 R.BT656 synchronizing code (EAV) timing (625/50 system)

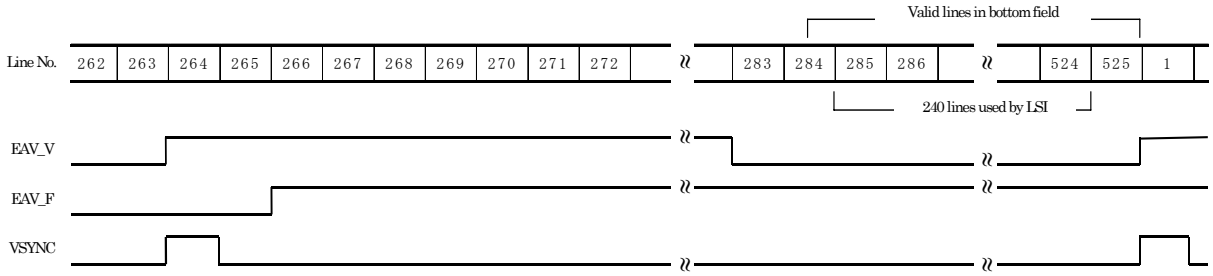
LINE-No,	620	621	622	623	624	625	1	2	3	4	5	6	7	8	9
1st_field	ACTIVE-VIDEO			Serration pulse			Vertical synchronization			Serration pulse		VBI-lines 1st_field			
EAV	DA	DA	DA	DA	F1	F1	B6	B6	B6	B6	B6	B6	B6	B6	B6
F	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
V	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
LINE-No,	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322
2nd_field	ACTIVE-VIDEO			Serration pulse		Vertical synchronization			Serration pulse			VBI-lines 2nd_field			
EAV	9D	9D	9D	B6	B6	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
F	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
V	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

LINE-No,	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1st_field	VBI-lines 1st_field														ACTIVE-VIDEO	
EAV	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	9D	9D	9D
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
LINE-No,	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338
2nd_field	VBI-lines 2nd_field														ACTIVE-VIDEO	
EAV	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	DA	DA	DA
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

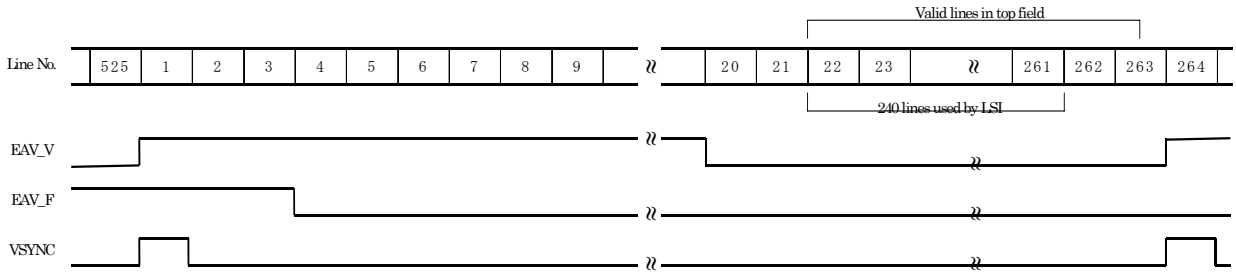
8.6.1.6 R.BT656 valid line

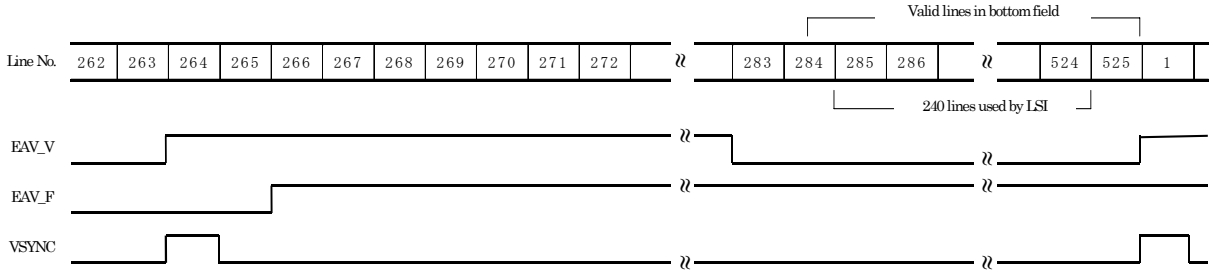
For R.BT656 (525/60 system)



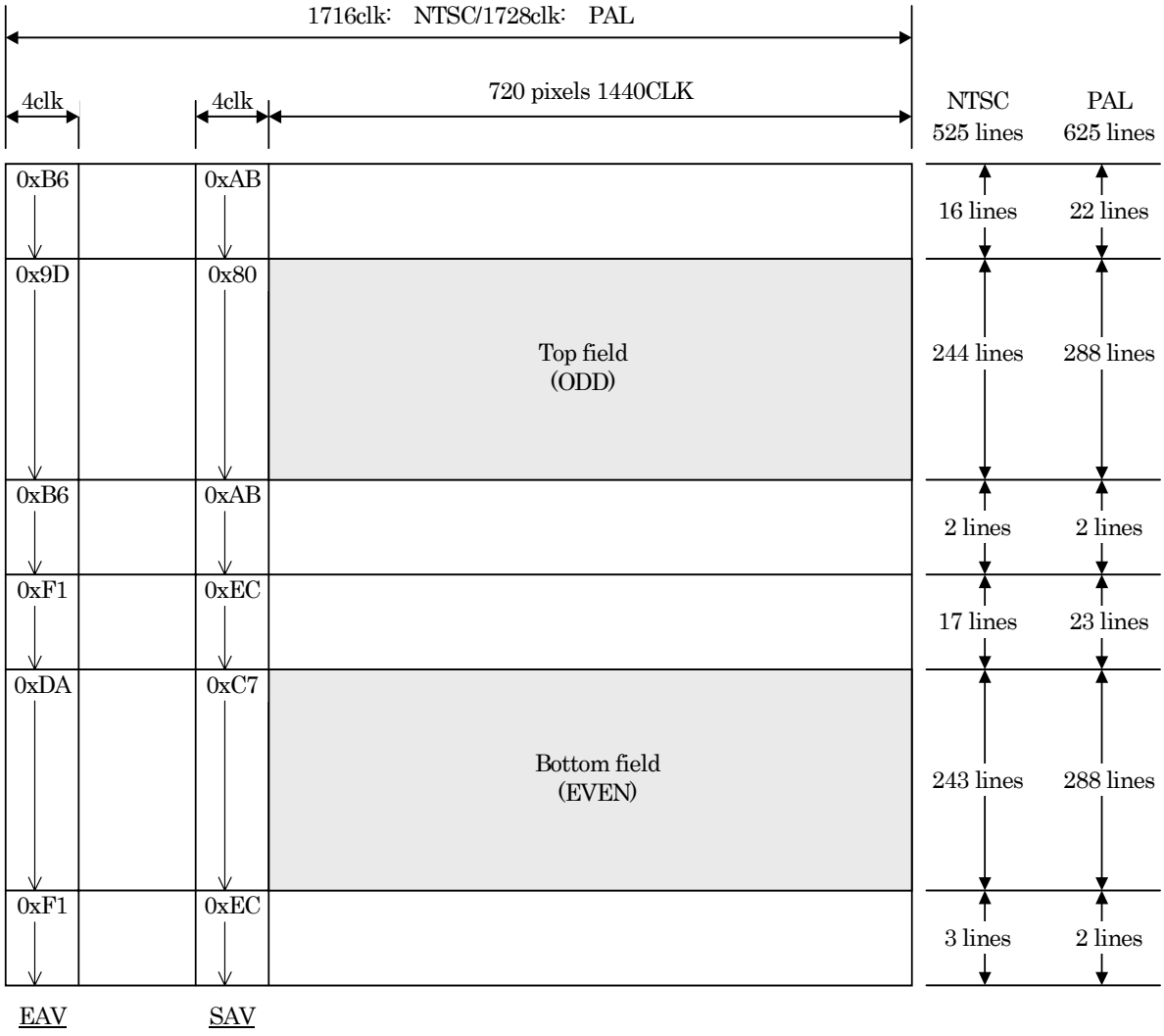


For R.BT656 (625/50 system)



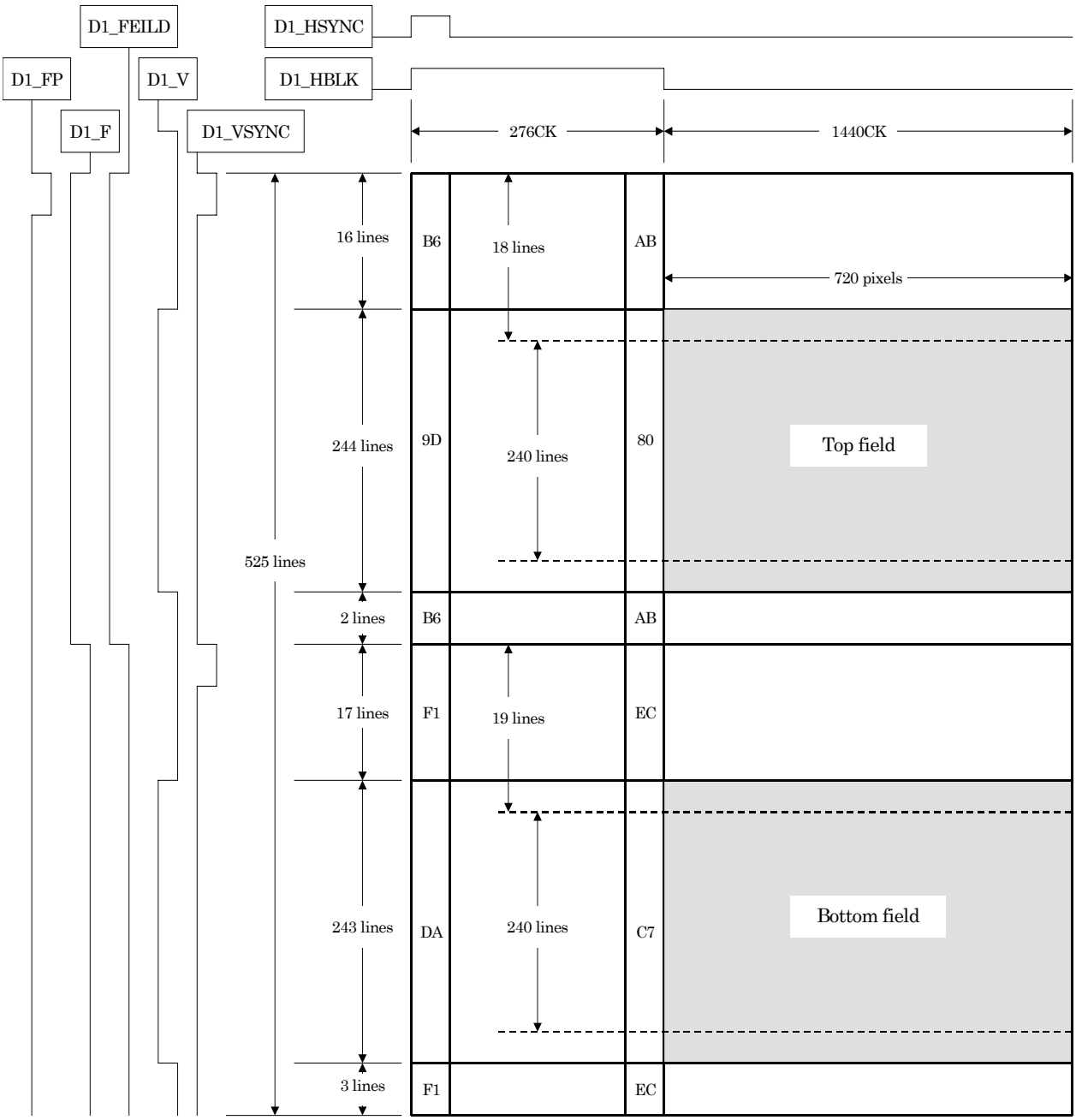


Various synchronization signals are detected from EAV and SAV, and the position of valid pixels is detected according to parameter setting.


1716clk: NTSC/1728clk: PAL
4clk
4clk
720 pixels 1440CLK
NTSC
525 lines
PAL
625 lines
0xB6
0xAB
16 lines
22 lines
0x9D
0x80
Top field
(ODD)
244 lines
288 lines
0xB6
0xAB
2 lines
2 lines
0xF1
0xEC
17 lines
23 lines
0xDA
0xC7
Bottom field
(EVEN)
243 lines
288 lines
0xF1
0xEC
3 lines
2 lines
EAV
SAV

8.6.1.7 R.BT656 frame format

R.BT656 format input [525/60 system]

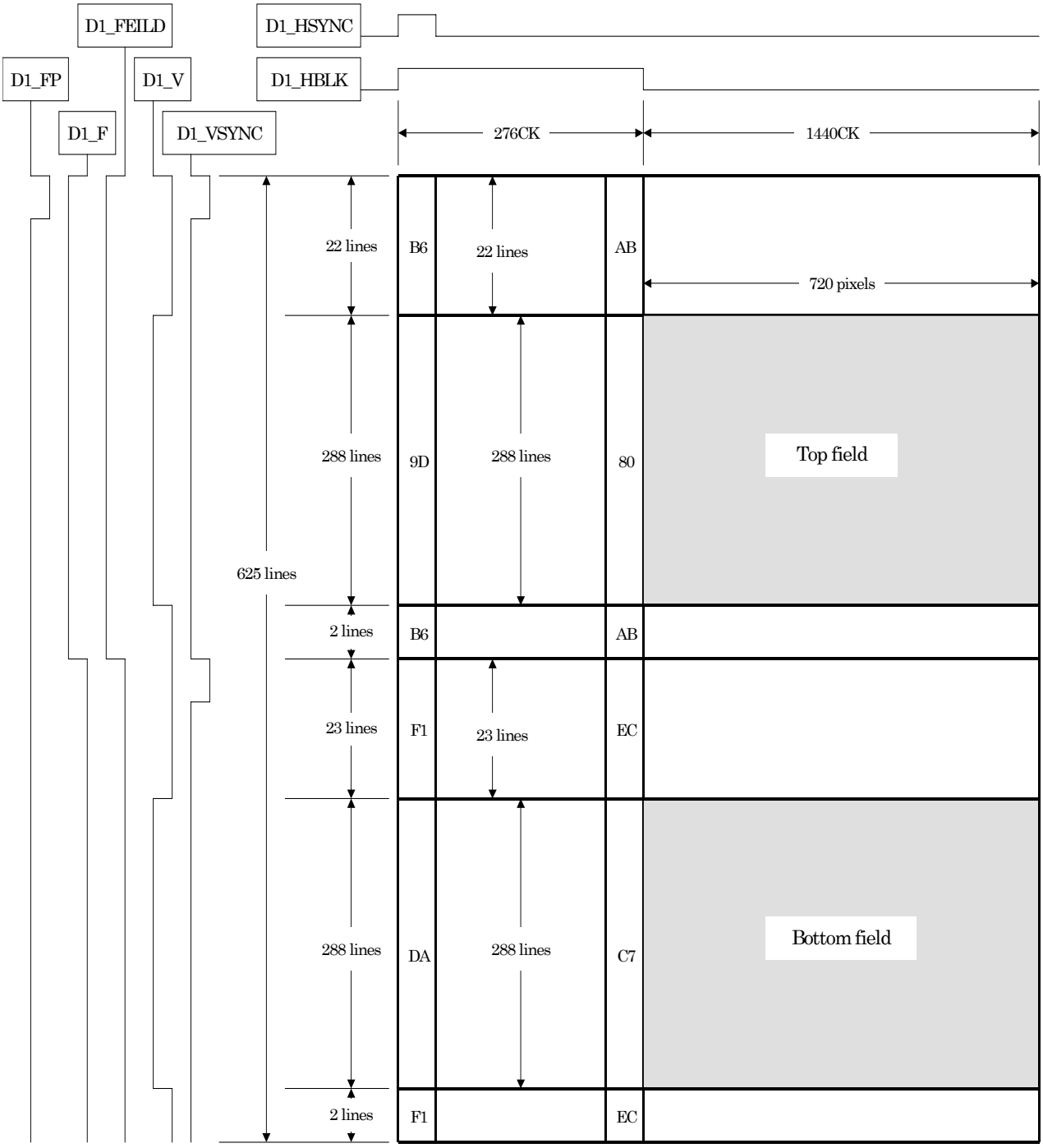


Note 1: CCLK_RGB = 27 MHz

Note 2: SAV and EAV shown are actually 4 bytes ('FF 00 00 XX').

Note 3: D1_FP, D1_F, D1_FEILD, D1_V, D1_VSYNC, D1_HSYNC, and D1HBLK are the images of internal signals generated from SAV and EAV.

R.BT656 format input [625/50 system]



Note 1: CCLK_RGB = 27 MHz

Note 2: SAV and EAV shown are actually 4 bytes ('FF 00 00 XX').

Note 3: D1_FP, D1_F, D1_FEILD, D1_V, D1_VSYNC, D1_HSYNC, and D1HBLK are the images of internal signals generated from SAV and EAV.

8.6.2 RGB Input Format

The RGB input video capture function has 2 data processing modes. One is the data processing mode as Native RGB mode, and the other is the one where RGB is converted to YUV422 by the internal RGB processor and then data processing is performed.

The RGB input function supports progressive video input. It does not support the interlace/progressive conversion function. It supports input of max 66Mpixel/sec. RGB component data is 6 bits.

Note: To set Native RGB mode, set NRGB to 1.

8.6.2.1 RGB input signal

Name	I/O	Function
CCLK_RGB	Input	Clock for RGB input
CAP0R0-5	Input	Red component value
CAP0G0-5	Input	Green component value
CAP0BI0-5	Input	Blue component value
CAP0VS	Input	Vertical sync for RGB capture
CAP0HS	Input	Horizontal sync for RGB capture

Note: Selection between YUV422 input (R.BT656) and RBT input is performed using the VIS bit of the VCM (video capture mode) register.

8.6.2.2 Setting of capture range

The RGB input function sets the following registers

(a) Setting of RGB input mode

- The RGB666 input flag (VIS) of the VCM register is set
- In Native RGB mode, the NRGB bit of the VCM register is set to “1”.

(b) Setting of HSYNC cycle

The HSYNC cycle is set for RGBHC.

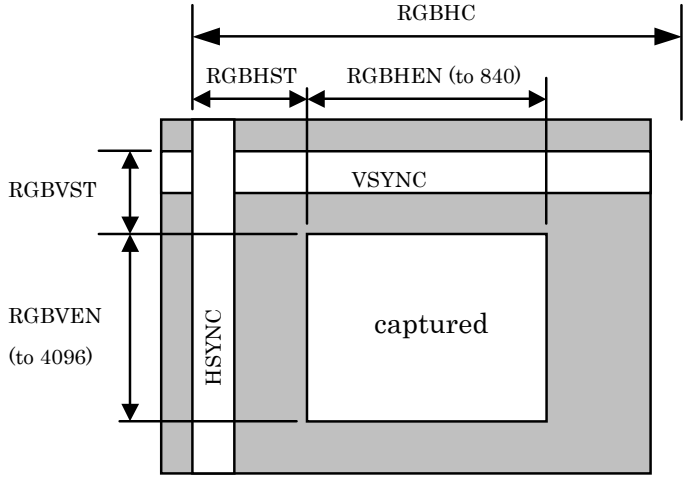
(c) Setting of horizontal valid pixel range

The starting position and size of valid pixel are set for RGBHST and RGBHEN.

(d) Setting of vertical valid pixel range

The starting position and size of valid pixel are set for RGBVST and RGBVEN.

The video capture range is defined as shown below.



RGBHC	RGB input Hsync Cycle
RGBHST	RGB input Horizontal enable area STart position
RGBHEN	RGB input Horizontal enable area size
RGBVST	RGB input Vertical ENable area STart position
RGBVEN	RGB input Vertical ENable area size

Note: In fact, setting of the display parameters is slightly different from the above. For details, see *8.7.2 Video Capture Register*.

(e) Conversion matrix coefficient

Set RGBCMY, RGBCMCb, RGBCMCr and RGBCMb as color conversion matrix coefficient.

Note: The horizontal valid pixel count that is set (RGBHEN) is max 840. This is due to the restriction by the line buffer size in the video capture module.

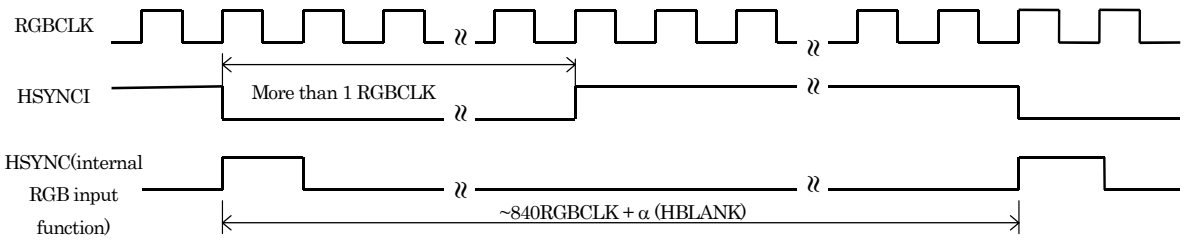
8.6.2.3 RGB Input Format

At RGB input time, data synchronization is performed using VSYNCI and HSYNCI that are input together with data RI, GI and BI.

1) Input regulation for HSYNCI

The rising edge or falling edge of CAP0HS is set as horizontal synchronization by setting the HP register.

Input signal is 1 clock (CCLK_RGB) or greater.



Note1: The horizontal valid pixel count that is set (RGBHEN) is max 840. This is due to the restriction by the line buffer size in the video capture module.

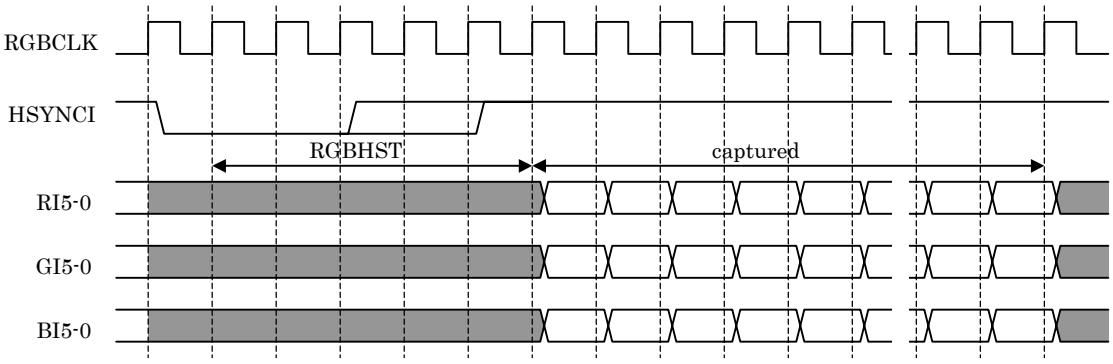
Note2: RGB_HSYNC is the image of an internal signal generated from CAP0HS.

2) Input regulation for valid pixel data for HSYNC

The input regulation for valid image data for HSYNC is shown.

Input data is input in synchronization with HSYNC of each line.

The distance from HSYNC to the beginning of valid pixel in data can be changed by setting the RGBHST register.

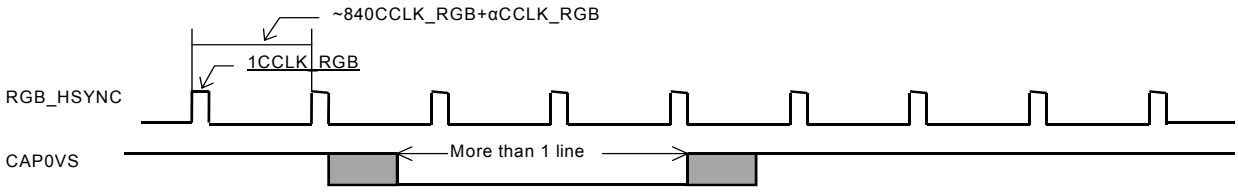


3) Input regulation for VSYNC

According to register setting, HSYNC samples CAP0VS to treat it as an internal HSYNC signal.

At this time, the width of the CAP0VS signal input from the outside is at least one line or greater.

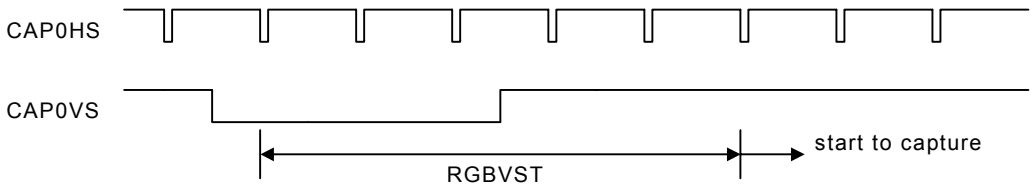
Either the rising or falling edge of CAP0VS is set to the VSYNC signal using the VP register setting.



Note： RGB_HSYNC is the image of the internal signal generated from CAP0HS.

4) Input regulation on valid pixel data for VSYNC

The input regulation on valid pixel data for VSYNC is shown below.



Note: The RGB input mode does not support input of interlace video. Set RGBVST_T_0 to “00”.

5) Color space conversion

Conversion from RGB to YCbCr is performed using the following matrix expression:

$$Y = a11*R + a12*G + a13*B + b1$$
$$Cb = a21*R + a22*G + a23*B + b2$$
$$Cr = a31*R + a32*G + a33*B + b3$$

a_{ij} : 10 bits signed real (lower 8 bits is fraction)

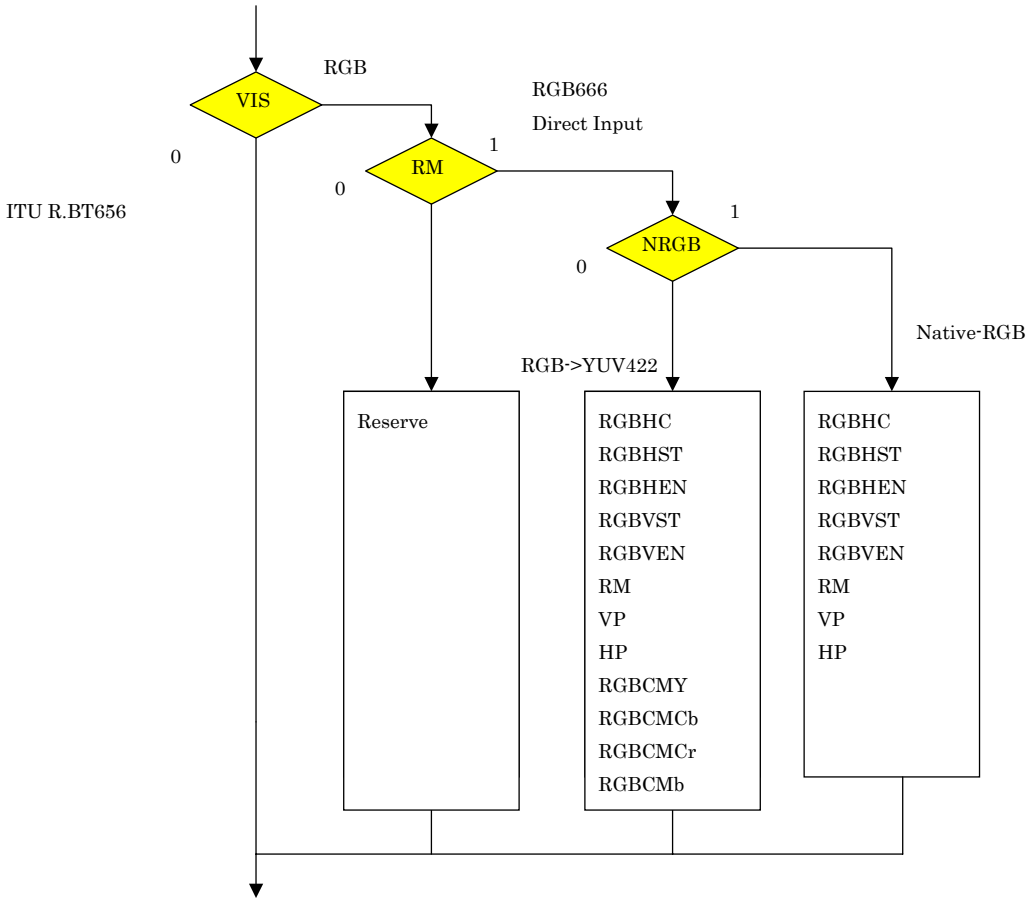
b_i: 8 bits unsigned integer

Note1: Each coefficient can be defined by setting the relevant register.

Note2: The converted YCbCr signal is further converted to the 4:2:2 format and then internal image processing is performed.

8.6.3 RGB Video Input Parameter Setting Chart

Registers requiring parameter setting vary with video input mode. See the following chart.



Register Setting Chart by RGB Video Input Mode

8.7 Registers

8.7.1 Register List

New registers added in MB86296 (Coral-PA) are shown shaded.

Base = CaptureBase0 or CaptureBase1

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000	VCM (Video Capture Mode)																															
	VIE	VIS		VIC			CM					VI																		NRGB	VS	
004	CSC (Capture SCale)																															
	VSCI				VSCF												HSCI				HSCF											
008	VCS_0 (Video Capture Status 0)																															
																									CE							
010	CBM (Capture Buffer Mode)																															
	OOM	SBUF	CRGB						CBW (stride)								BLEN	C24	RGBA													
014	CBOA (Capture Buffer Origin Address)																															
018	CBLA (Capture Buffer Limit Address)																															
01C	CISTR (Capture Image Start)																															
							CIVSTR																CIHSTR									
020	CIEND (Capture Image End)																															
							CIVEND																CIHEND									
028	CHP (Capture Horizontal Pixel)																															
																							CHP									
040	CLPF (Capture Low Pass Filter)																															
					CVLPF								CHLPF																			
048	CMSS (Capture Magnify Source Size)																															
					CMSHP																CMSVL											
04C	CMDS (Capture Magnify Display Size)																															
					CMDHP																CMDVL											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
080	RGBHC(RGB input HSYNC Cycle)/VIN_HSSIZE																															
																			RGBHC / VIN_HSSIZE													
084	RGBHEN(RGB input Horizontal Enable Area)																															
			RGBHST																RGBHEN													
088	RGBVEN(RGB input Vertical Enable Area) / VIN_HEN																															
			RGBVST_T_O						RGBVST / VIN_HVAL										RGBVEN / VIN_HSIZE													
08C	VIN_VSAMP																															
																							VJITFLT									FLDREV
090	RGBS(RGB input SYNC) / VIN_SS																															
																RM															HP	VP
0C0	RGBCMY(RGB Color convert Matrix Y coefficient)																															
	a11												a12												a13							
0C4	RGBCMCb(RGB Color convert Matrix Cb coefficient)																															
	a21												a22												a23							
0C8	RGBCMCr(RGB Color convert Matrix Cr coefficient)																															
	a31												a32												a33							
0CC	RGBCMb(RGB Color convert Matrix b coefficient)																															
			b1										b2												b3							
178	CINT																															
																															VS	
17C	CIMSK																															
																															VS	

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
180	SYNC_err																															
							VSL_err	VSS_err								VS_err							HSL_err	HSS_err								HS_err
184	SYNC_error_MSK																															
							MVSL_err	MVSS_err								MVS_err							MHSL_err	MHSS_err								MHS_err
300	CVCNT																															
1000	CDCN(Capture Data Count for NTSC)																															
	BDCN																				VDCN											
1004	CDCP(Capture Data Count for PAL)																															
	BDCP																				VDCP											
1014	MDS																															
																											YCMIM					
104C	VINLC_kep																															
																					VIN_LINE_NO_kep											
1054	VHSLS (Video Input HSYNC Long/Short)																															
			VIN_HS_LONG																		VIN_HS_SHORT											
1058	VHSDC (Video HSYNC Down Count)																															
	VIN_EHSYNC_MSK																								VIN_HDOWN_CNT							
1060	VVSLS (Video VSYNC long/short)																															
			VIN_HS_LONG																		VIN_HS_SHORT											
1064	VVSDC (Video VSYNC Down Count)																															
	VIN_EVSYNC_MSK																								VIN_VDOWN_CNT							

8.7.2 Video Capture Register

VCM (Video Capture Mode)

Register address	CaptureBaseAddress + 00H																												
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VIE	VIS	Reserve	VICE	Reserve	CM		Reserve		VI	Reserve										VF	Reserve								NRGB	VS	Rsv
R/W	RW	RW	RX	RW	RX	RW		RX		RW	RX										RW	RX								RW	RW	RX
Initial value	0	0		0	X	00		X		0	X										0	X								0	0	X

Sets video capture mode. This register is not initialized by software (= soft) reset.

Bit 1 VS (Video Select)
Selects NTSC or PAL, to detect code error (only at R.BT656 input time).
0 NTSC
1 PAL

Bit 2 NRGB (Native RGB input on).
Native RGB mode is set up
0 YUV 4:2:2
1 Native RGB

Bit 11 VF (Video Format select)
Selects a video input format.
0 R.BT656 input format
1 YC multiplex input format

Bit 20 VI (Vertical Interpolation)
Sets vertical direction interpolation processing.
0 Performs interpolation processing in vertical direction. Images are enlarged twice as much in vertical direction.
1 Performs no interpolation processing in vertical direction.

Bit 25-24 CM (Capture Mode)
Sets vide capture mode. When performing capture, set the field to 11.
00 Initial value
01 Reserved.
10 Reserved.
11 Uses capture.

Bit 28 VICE (Video Input Clock Enable)
Enables capture lock.
0 Enables capture lock.
1 Disables capture lock.

Bit 30 VIS (Video Input Select)
0 RBT656
1 RGB

Bit 31 VIE (Video Input Enable)
Enables the video capture function.
0 Performs no video capture.
1 Performs video capture.

Procedure to stop video capture clock

■ Write 0 to bit 31 (VIE) of the VCM register, to disable the video capture function.

■ Write 1 to bit 28 (VICE) of the VCM register, to stop the video capture clock.

Procedure to start video capture clock

■ Write 0 to bit 28 (VICE) of the VCM register, to enable the video capture clock.

■ Write 1 to bit 31 (VIE) of the VCM register, to enable the video capture function.

CSC (Capture SCale)

Register address	CaptureBaseAddress + 04_H			
Bit No.	31 30 29 28 27	26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
Bit field name	VSCI	VSCF	HSCI	HSCF
R/W	RW	RW	RW	RW
Initial value	00001	00000000000	00001	00000000000

Sets a video capture scaling ratio.

Bit 10-0 HSCF (Horizontal SCale Fraction)
Sets the decimal part of the horizontal direction scaling ratio.

Bit 15-11 HSCI (Horizontal Scale Integer)
Sets the integer part of the horizontal direction scaling ratio.

Bit 26-16 VSCF (Vertical SCale Fraction)
Sets the decimal part of the vertical direction scaling ratio.

Bit 31-27 VSCI (Vertical SCale Integer)
Sets the integer part of the vertical direction scaling ratio.

CBM (video Capture Buffer Mode)

Register address	CaptureBaseAddress + 10_H																					
Bit No.	31	30	29	28	27…24	23 22…17 16	15	14	13	12	11	10 9 8	7 6 5	4	3 2 1	0
Bit field name	OO	SBUF	CRGB		Reserve	CBW	resv	C24	resv	CSW	resv				reserve	CBST
R/W	RW	RW	RW	RW	RX	RW	RX	RW	RX	RW	RX	RW	RW	RW	RX	RW
Initial value	0	X	X	0	X	X	X	0	X	0	X	000	000	0	X	0

Bit 0 CBST (Capture Burst)

Specifies the burst length that is at capture write time. When burst is longer, access efficiency becomes higher, and so Fujitsu recommends that the bit be set to 1.

0 Standard burst write (4words)

1 Long burst write (8words)

Bit 12 CSW (Color Swap)

Interchanges the byte position of the color components.

0 Performs no interchanging.

1 Performs interchanging.

Bit 14 C24 (Color 24bit/pixel)

Selects between 24 bits/pixel and 16 bits/pixel when performing RGB input capture.

This function is enabled when Native RGB capture (NRGB = 1) or converted RGB capture (CRGB = 1) is set.

0 16bit/pixel

1 24bit/pixel

Bit 23-16 CBW (Capture Buffer memory Width)

Sets the memory width (the stride) of the capture buffer in units of 64 bytes.

Bit 29 CRGB (Capture RGB write)

Specifies that data in YCbCr format be converted to RGB5:5:5 (16 bits/pixel) and then written.

0 YCbCr format (no conversion)

1 RGB format

Bit 30 SBUF (Single Buffer)

Specifies that the capture buffer be managed using the single buffer mode.

0 Normal mode (ring buffer mode)

1 Single buffer mode

Bit 31 OO (Odd Only mode)

Specifies that only odd fields be captured.

0 Normal mode

1 Odd-only mode

Note: This register is not initialized by software reset.

CBOA (video Capture Buffer Origin Address)

Register address	CaptureBaseAddress + 14_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								CBOA																							
R/W	RX								RW																						R0	
Initial value	Undefined								Undefined																						0	

Specifies the starting address of the video capture buffer.

CBLA (video Capture Buffer Limit Address)

Register address	CaptureBaseAddress + 18_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								CBLA																							
R/W	RX								RW																						R0	
Initial value	Undefined								Undefined																						0	

Specifies the ending address of the video capture buffer.

Set CBLA to be greater than CBOA.

CISTR (Capture Image STaRt)

Register address	CaptureBaseAddress + $1C_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CIVSTR												Reserved				CIHSTR											
R/W	RX				RW												RX				RW											
Initial value	Undefined				Undefined												Undefined				Undefined											

Sets the range of image written to the video capture buffer. Specifies the upper left coordinates (CIHSTR, CIVSTR) that are in the write range, relative to the upper left point of the image. When an image is reduced, this function applies to the coordinates of the reduced image.

Bit 11-0 CIHSTR (Capture Image Horizontal STaRt)
Specifies the X coordinate.

Bit 27-16 CIVSTR (Capture Image Vertical STaRt)
Specifies the Y coordinate

CIEND (Capture Image END)

Register address	CaptureBaseAddress + 20H																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CIVEND												Reserved				CIHEND											
R/W	RX				RW												RX				RW											
Initial value	Undefined				Undefined												Undefined				Undefined											

Sets the range of image written to the video capture buffer. Specifies the lower right coordinates (CIHEND, CIVEND) that are in the write range, relative to the upper left point of the image. When an image is reduced, this function applies to the coordinates of the reduced image.

When the raster count of the input image is smaller than this set range, only the data equivalent to the size of the input image is written.

Bit 11-0 CIHEND (Capture Image Horizontal END)
Specifies the X coordinate.

Bit 27-16 CIVEND (Capture Image Vertical END)
Specifies the Y coordinate.

CVCNT (Capture Vertical Count)

Register address	CaptureBaseAddress + 300H															
Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CVCNT											
R/W	R0				R											
Initial value	0				Undefined											

Indicates the Y coordinate of the currently captured raster. This register can only be read.

CHP (Capture Horizontal Pixel)

Register address	CaptureBaseAddress + 28H																											
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																								CHP							
R/W	RX																								RW							
Initial value	X																								0x168 (360)							

Sets the horizontal pixel count of the image that is output to the capture buffer. Specifies the value in units of 2 pixels. The maximum value is 840 pixels (the set value is 0x1A4).

CLPF (Capture Low Pass Filter)

Register address	CaptureBaseAddress + 40_H				
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6 5 4 3 2	1 0
Bit field name	Reserved	CVLPF	Reserved	CHLPF	Reserved
R/W	RX	R/W	RX	R/W	RX
Initial value	0	0	0	0	X

Sets the coefficient of the low-pass filter. The vertical low-pass filter is a 3-tap FIR filter; the horizontal low-pass filter is a 5-tap FIR filter. The coefficient for the luma (Y) signal and chroma (C) signal is specified independently using 2-bit coefficient code. When coefficient code "00" is set, the low-pass filter is set to OFF (through).

Bit 17 to 16 CHLPF_C (Capture Horizontal LPF coefficient C)

CHLPF_C	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

Bit 19 to 18 CHLPF_Y (Capture Horizontal LPF coefficient Y)

CHLPF_Y	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

Bit 25 to 24 CVLPF_C (Capture Vertical LPF coefficient C)

CVLPF_C	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting is disabled		

Bit 27 to 26 CVLPF_Y (Capture Vertical LPF coefficient Y)

CVLPF_Y	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting is disabled		

CMSS (Capture Magnify Source Size)

Register address	CaptureBaseAddress + 48_H			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	CMSHP	Reserved	CMSVL
R/W	RX	RW	RX	RW
Initial value	X	X	X	X

Bit 9-0 CMSVL (Capture Magnify Source Vertical Line)

Sets the vertical line count of the image for which enlargement scaling processing is not yet performed.

Bit 25-16 CMSHP (Capture Magnify Source Horizontal Pixel)

Sets the horizontal pixel count of the image for which enlargement scaling processing is not yet performed. Specifies the value in units of 2 pixels.

CMDS (Capture Magnify Display Size)

Register address	CaptureBaseAddress + $4C_H$			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	CMDHP	Reserved	CMDVL
R/W	RX	RW	RX	RW
Initial value	X	X	X	X

Bit 9-0 CMDVL (Capture Magnify Display Vertical Line)

Sets the vertical line count of the image for which enlargement scaling processing is performed.

Bit 26-16 CMDHP (Capture Magnify Display Horizontal Pixel)

Sets the horizontal pixel count of the image for which enlargement scaling processing is performed. Specifies the value in units of 2 pixels.

RGBHC (RGB input Hsync Cycle)

Register address	CaptureBaseAddress + 80H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																						RGBHC									
R/W	RX																						RW									
Initial value	X																						X									

Bit 13-0 RGBHC

Inputs the horizontal cycle that is at video input time. The field is used when sampling VSYNC by the VIN_VLSAMP register or when detecting disconnected HSYNC that is at YC multiplex input time or at RGB/R.BT656 input time. Set value + 1 is the horizontal cycle.

RGBHEN (RGB input Horizontal Enable area)

Register address	CaptureBaseAddress + 84H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								RGBHST								Reserved						RGBHEN									
R/W	RX								RW								RX						RW									
Initial value	X								X								X						X									

This register is the parameter to determine the valid pixel data in horizontal direction. This register is used for YC multiplex input format (R.BT601).

Bit 12-0 RGBHEN

Sets valid pixel data size in 2-pixel units.

Bit 25-16 RGBHST

Sets the starting position of valid pixel data. Set value - 4 is the starting position.

RGBVEN (RGB input Vertical Enable area)

Register address	CaptureBaseAddress + 88H					
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0			
Bit field name	Reserved	RGB_VST_O	Reserved	RGBVST	Reserved	RGBVEN
R/W	RX	RW	RX	RX	RX	RX
Initial value	X	1	X	X	X	X

This register is the parameter to determine the valid pixel data in vertical direction. This register is used when using RGB input format.

Bit 12-0 RGBVEN (RGB input Vertical Enable area Size)
Sets the size of valid line that is in vertical direction.

Bit 25-16 RGBVST (RGB input Vertical Enable area Start position)
Sets the starting position of valid line data. Set value - 1 is the starting position.

Bit 29-28 RGBVST_T_O (RGB input Vertical Enable area Start position for Top field)
2bit signed integer
Start position of valid line = RGBVST_O + RGBVST_BOTTOM. Since Carmine supports only progressive RGB input, set RGBVST_O to "0".

RGBS (RGB input Sync)

Register address	CaptureBaseAddress + 90H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																RM	Reserved													HP	VP
R/W	RX																R/W	RX													R/W	R/W
Initial value	X																0	X													0	0

Sets edge detection of synchronization signal. This register is used when RGB is input.

Bit 0 VP (VINVSYNC Polarity)
0 Sets the falling edge of CAP0VS as VSYNC.
1 Sets the rising edge of CAP0VS as VSYNC.

Bit 1 HP(HSYNC Polarity)
0 Sets the falling edge of CAP0VS as HSYNC.
1 Sets the rising edge of CAP0VS as HSYNC.

Bit 16 RM (RGB Input Mode select)
Sets RGB666 direct input mode.
0 Reserved
1 RGB666 Direct input Mode

Color space conversion

RGB input data is converted to YCbCr format by the following matrix expression:

Y = a11*R + a12*G + a13*B + b1
Cb= a21*R + a22*G + a23*B + b2
Cr= a31*R + a32*G + a33*B + b3

a_{ij} 10bit signed real (lower 8bit is fraction)
b_i 8bit unsigned integer

- Each coefficient is set by the following registers.
- The chroma signal CbCr is subjected to 2-4 conversion filter processing to become YCbCr:4:2:2 format, before color space conversion is performed.

RGBCMY (RGB Color convert Matrix Y coefficient)

Register address	CaptureBaseAddress + $C0_H$																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	a11										Reserved	a12										Reserved	a13								
R/W	RW										RX	RW										RX	RW								
Initial value	0001000010										x	0010000000										x	0000011001								

This register sets the coefficient that calculates the Y signal when converting RGB to YCbCr.

Bit 9-0 a13
10bit signed real (lower8bit is fraction)

Bit 20-11 a12
10bit signed real (lower8bit is fraction)

Bit 31-22 a11
10bit signed real (lower8bit is fraction)

RGBCMCb (RGB Color convert Matrix Cb coefficient)

Register address	CaptureBaseAddress + C4H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	a21										Reserved	a22										Reserved	a23								
R/W	RW										RX	RW										RX	RW								
Initial value	1111011010										X	1110110110										X	0001110000								

This register sets the coefficient that calculates the Cb signal when converting RGB to YCbCr.

Bit 9-0 a23
10bit signed real (lower8bit is fraction)

Bit 20-11 a22
10bit signed real (lower8bit is fraction)

Bit3 1-22 a21
10bit signed real (lower8bit is fraction)

RGBCMCr (RGB Color convert Matrix Cr coefficient)

Register address	CaptureBaseAddress + C8H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	a31											a32											a33									
R/W	RW										RX	RW										RX	RW								
Initial value	0001110000										X	1110100010										X	1111101110								

This register sets the coefficient that calculates the Cr signal when converting RGB to YCbCr.

Bit 9-0 a33
10bit signed real (lower8bit is fraction)

Bit 20-11 a32
10bit signed real (lower8bit is fraction)

Bit 31-22 a31
10bit signed real (lower8bit is fraction)

RGBCMb (RGB Color convert Matrix b coefficient)

Register address	CaptureBaseAddress + CC_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	b1									Reserved	b2									Reserved	b3										
R/W	RX	RW									RX	RW									RX	RW										
Initial value	X	000010000									X	010000000									X	010000000										

This register sets an ITU R.BT-656 offset correction.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = A \times \begin{bmatrix} Y - b1 \\ Cb - b2 \\ Cr - b3 \end{bmatrix}$$

Bit 8-0 b3
9bit unsigned integer

Bit 19-11 b2
9bit unsigned integer

Bit 30-22 b1
9bit unsigned integer

See also section 'RGB Input Format'.

VIN_SS (Video INput Sync Set)

Register address	CaptureBaseAddress + 90_H																													
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															RM	Reserved														HP	VP
R/W	RX															RW	RX														RW	RW
Initial value	X															0	X														0	0

This register sets edge detection of the synchronization signal. This register is used in the RGB mode.

Bit 0 VP (CAP0VS Polarity)
- 0 Sets falling edge of CAP0VS as VSYNC
- 1 Sets rising edge of CAP0VS as VSYNC

Bit 1 HP (CAP0HS Polarity)
- 0 Sets falling edge of CAP0HS as HSYNC.
- 1 Sets rising edge of CAP0HS as HSYNC.

Bit 16 RM(RGB Input Mode select)

Always set this bit to "1" in the RGB input mode.
- 0 Reserved
- 1 RGB666 Direct input Mode

VIN_VSAMP (Video INput Vsync SAMPling mode)

Register address	CaptureBaseAddress + 8C$_H$																																
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																						VJITFLT		Reserved							FLDREV
R/W	RX																						RW		RX							RW
Initial value	X																						00		X							0

Bit 0 FLDREV(FieLD REVerse)
Inverts logic of CAP0FID input.
0 Does not invert.
1 Inverts.

Bit 9-8 VJITFLT(Vsync JITter FiLTer)
Sets how to sample HSYNC signal input from CAP0HS.
Select 00.
00 Samples via HSYNC.
01 Reserved
10 Reserved
11 Reserved

MDS (MoDe Select)

Register address	CaptureBaseAddress + 1014_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																										YCMIM		Reserved			
R/W	RX																										RW		RX			
Initial value	X																										00		X			

Bit 5-4 YCMIM (YC Multiplex video Input Mode)

Use by default.

00 Fixed

01 Reserved

10 Reserved

11 Setting prohibited

[Detection of synchronization error] (optional function)

<This function is common to RGB/R.BT656>

VINLC (Video INput Line Count)

Register address	CaptureBaseAddress + $104C_H$																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																				VIN_LINE_NO_kep											
R/W	RX																				R											
Initial value	X																				0											

Bit 12-0 VIN_LINE_NO_kep

Indicates the line count of 1 frame (field) (the blanking period is also included). Displayed value + 1 is the line count.

VHSLS (Video Input HSYNC Long/Short)

Register address	CaptureBaseAddress + 1054_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VIN_HS_LONG												Reserved				VIN_HS_SHORT											
R/W	RX				RW												RX				RW											
Initial value	X				X												X				X											

Bit 13-0 VIN_HS_SHORT Sets video input short interval HSYNC monitoring.

When the input CAP0HS is equal to or less than the set interval, HSS_err is set to 1. Set value + 1 is the cycle.

Bit 29-16 VIN_HS_LONG Sets video input long interval HSYNC monitoring.

When the input CAP0VS exceeds the set interval, HSL_err is set to 1. Set value + 1 is the cycle.

VHSDC (Video input HSync Down Count)

Register address	CaptureBaseAddress + 1058H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VIN_EHSYNC_MSK	Reserved							Reserved								Reserved								Reserved							
																									VIN_HDOWN_CNT							
R/W	RW	RX			RW	RX																										
																									RW							
Initial value	0	X			0	X																			0							

Bit 7-0 VIN_HDOWN_CNT Sets the video input HSYNC disconnection monitoring cycle.
When VSYNC disconnection exceeds the set cycle, HS_err is set to 1. Set value + 1 is the cycle.

Bit 31 VIN_EHSYNC_MSK Video input error HSYNC mask (only YC multiplex input mode)
Masks the range equal to or smaller than the value set for VIN_HS_SHORT. When this range is masked, CAP0HS input during this period is not detected.
0： No mask 1： Mask

VVSLS (Video input VSync Long/Short)

Register address	CaptureBaseAddress + 1060H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																Reserved								VIN_VS_SHORT							
									VIN_VS_LONG																									
R/W	RX								RW								RX								RW							
Initial value	X								X								X								X							

Bit 12-0 VIN_VS_SHORT Sets video input short interval VSYNC monitoring.
When the input CAP0VS is equal to or less than the set interval, VSS_err is set to 1. Set value + 1 is the cycle.

Bit 28-16 VIN_VS_LONG Sets video input long interval VHSYNC monitoring.
When the input CAP0VS exceeds the set interval, VSL_err is set to 1. Set value + 1 is the cycle.

VVSDC (Video Input VSync Down Count)

Register address	CaptureBaseAddress + 1064_H																															
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VIN_EVSYNC_MSK	Reserved																							VIN_VDOWN_CNT							
R/W	RW	R																							RW							
Initial value	0	0																							0							

Bit 7-0	VIN_VDOWN_CNT	Sets the video input VSYNC disconnection monitoring cycle. When VSYNC disconnection exceeds the set cycle, VS_err is set to 1. Set value + 1 is the cycle.
Bit 31	VIN_EVSYNC_MSK	Video input error VSYNC mask (only YC multiplex input mode) Masks the range equal to or smaller than the value set for VIN_VS_SHORT. When this range is masked, the input CAP0VS during this period is not detected. 0: No mask 1: Mask

CINT(Capture Interrupt)

Register address	CaptureBaseAddress + 178_H																															
Bit No.	31	30	29	28	27	26					21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																														VS	
R/W	R																														RW0	
Initial value	0																														0	

This register is the video synchronization signal interrupt status register. This register is cleared by writing "0" to it.

Bit 1 VS (VSYNC)
1: VSYNC 0: No VSYNC

CIMSK(Capture Interrupt Mask)

Register address	CaptureBaseAddress + $17C_H$																															
Bit No.	31	30	29	28	27	26					21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																														VS	
R/W	R																														RW	
Initial value	0																														0	

This register masks the video synchronization signal interrupt

Bit 1 VS (VSYNC)
1: Does not mask interrupt. 0: Masks interrupt.

SYNC_err (SYNC error)

Register address	CaptureBaseAddress + 180_H																										
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						VSL_err	VSS_err	Reserved							VS_err	Reserved						HSL_err	HSS_err	Reserved							HS_err
R/W	RX						RW0	RW0	RX							RW0	RX						RW0	RW0	RX							RW0
Initial value	X						0	0	X							0	X						0	0	X							0

This register is error interrupt status register of the video synchronization signal. This register is cleared by writing "0" to it.

Bit 0 HS_err (Hsync error)
1: Video input HSYNC disconnection error 0: No error

Bit 8 HSS_err (HSync Short error)
1: Video input short interval HSYNC error 0: No error

Bit 9 HSL_err (Hsync Long error)
1: Video input long interval HSYNC error 0: No error

Bit 16 VS_err (Vsync down error)
1: Video input VSYNC disconnection error 0: No error

Bit 24 VSS_err (Vsync Short error)
1: Video input short interval VSYNC error 0: No error

Bit 25 VSL_err (Vsync Long error)
1: Video input long interval VSYNC error 0: No error

SYNC_err_MSK (SYNC error MaSK)

Register address	CaptureBaseAddress + 184_H																										
Bit No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						MVSL_err	MVSS_err	Reserved							MVS_err	Reserved						MHSL_err	MHSS_err	Reserved							MHS_err
R/W	RX						RW	RW	RX							RW	RX						RW	RW	RX							RW
Initial value	X						0	0	X							0	X						0	0	X							0

Masks the interrupt of video synchronization signal.

Bit 0 MHS_err (Mask HSync error)
1: No mask 0: Mask

Bit 8 MHSS_err (Mask HSync Short error)
1: No mask 0: Mask

Bit 9 MHSL_err (Mask HSync Long error)
1: No mask 0: Mask

Bit 16 MVS_err (Mask VSync error)
1: No mask 0: Mask

Bit 24 MVSS_err (Mask VSync Short error)
1: No mask 0: Mask

Bit 25 MVSL_err (Mask VSync Long error)
1: No mask 0: Mask

[Detection of code error]

(Only for R.BT656 format input)

CDCN (Capture Data Count for NTSC)

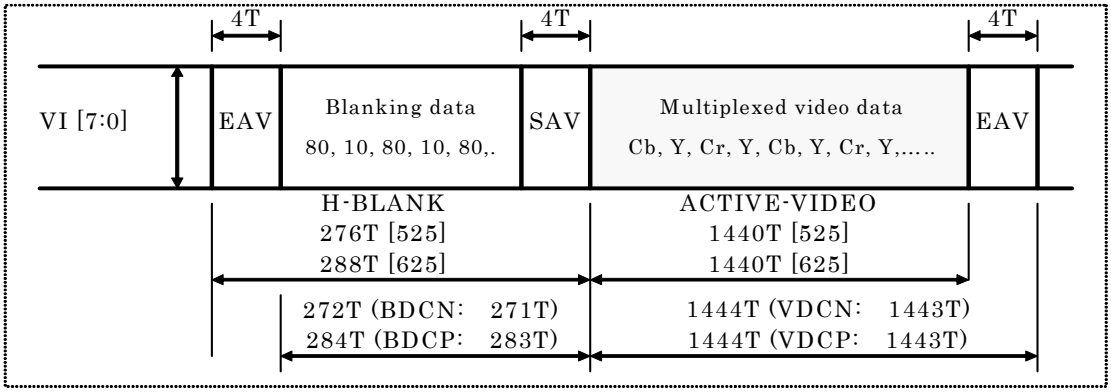
Register address	CaptureBaseAddress + 1000_H			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Bit field name	Reserved	BDCN	Reserved	VDCN
R/W	RX	RW	RX	RW
Initial value	X	0x10f (271)	X	0x5A3 (1443)

Sets the input video stream data count for when NTSC format is used.
(This register is only enabled when R.BT656 format is used.)

Bit 12-0 VDCN (Valid Data Count for NTSC)
Sets the data count for the valid period for when NTSC format is used. Set value + 1 is the data count.

Bit 28-16 BDCN (Blanking Data Count for NTSC)
Sets the data count for the blanking period for when NTSC format is used. Set value + 1 is the data count.

The range of VDCN and BDCN is as shown in the figure below.



SAV: start of active video timing reference code
EAV: end of active video timing reference code
T: clock period 37 ns nom.

CDCP (Capture Data Count for PAL)

Register address	CaptureBaseAddress + 1004_H			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	BDCP	Reserved	VDCP
R/W	RX	RW	RX	RW
Initial value	X	0x11B(283)	X	0x5A3(1443)

Sets the input video stream data count for when PAL format is used.
(This register is only enabled when R.BT656 format is used.)

Bit 12-0 VDCP (Valid Data Count for PAL)
Sets the data count for the valid period for when PAL format is used. Set value + 1 is the data count.

Bit 28-16 BDCP (Blanking Data Count for PAL)
Sets the data count for the blanking period for when PAL format is used. Set value + 1 is the data count.

CDCNS (Capture Data Count for NTSC Short)

Register address	CaptureBaseAddress + 1018_H			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit field name	Reserved	BDCN_S	Reserved	VDCN_S
R/W	RX	RW	RX	RW
Initial value	X	0x10f(271)	X	0x5A3(1443)

Sets the short interval input video stream data count for when NTSC format is used.
(This register is only enabled when R.BT656 format is used.)

Bit 12-0 VDCN_S (Valid Data Count for NTSC Short)
Sets the data count for the valid period for when NTSC format is used. Set value + 1 is the data count.

Bit 28-16 BDCN_S (Blanking Data Count for NTSC Short)
Sets the data count for the blanking period for when NTSC format is used. Set value + 1 is the data count.

CDCPS (Capture Data Count for PAL Short)

Register address	CaptureBaseAddress + 101C$_H$			
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Bit field name	Reserved	BDCP_S	Reserved	VDCP_S
R/W	RX	RW	RX	RW
Initial value	X	0x11B (283)	X	0x5A3 (1443)

Sets the short interval input video stream data count for when PAL format is used.
(This register is only enabled when R.BT656 format is used.)

Bit 12-0 VDCP_S (Valid Data Count for PAL Short)

Sets the data count for the valid period for when PAL format is used. Set value + 1 is the data count.

Bit 28-16 BDCP_S (Blanking Data Count for PAL Short)

Sets the data count for the blanking period for when PAL format is used. Set value + 1 is the data count.

VCS (Video Capture Status)

Register address	CaptureBaseAddress + 08_H	
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Bit field name	Reserve	CE0
R/W	RX	RW0
Initial value	X	00000

This register is the status register to show that error occurs in the input R.BT656 code. To detect error in code if it occurs, set VS of VCM to NTSC or PAL. When it is set to NTSC, the data count of the capture data count register (CDCN, CDCN_S) is referenced; when set to PAL, the data count of the capture data count register (CDCP, CDCP_S) is referenced. If the data count does not match the stream data count or if undefined code is detected in the fourth word of SAV/EAV, bit6-0 of the video capture status register (VCS) is set as shown below. (For code definition, see ***8.6.1 R.BT656YUV422 Input Format***.)

Bit 0	1:	R.BT656 undefined error (code bit7)	0:	No error
Bit 1	1:	R.BT656 undefined error (code bit7-4)	0:	No error
Bit 2	1:	R.BT656 undefined error (code bit7-0)	0:	No error
Bit 3	1:	R.BT656 long interval H code error (SAV)	0:	No error
Bit 4	1:	R.BT656 long interval H code error (EAV)	0:	No error
Bit 5	1:	R.BT656 short interval H code error (SAV)	0:	No error
Bit 6	1:	R.BT656 short interval H code error (EAV)	0:	No error

VCS_MSK (Video Capture Status MaSK)

Register address	CaptureBaseAddress + $0C_H$	
Bit No.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Bit field name	Reserve	MSK_CE0
R/W	RX	RW
Initial value	X	00000

Bit 0	1:	R.BT656 undefined error (code bit 7) mask absent	0:	R.BT656 undefined error (code bit 7) mask present
Bit 1	1:	R.BT656 undefined error (code bit 7-4) mask absent	0:	R.BT656 undefined error (code bit 7-4) mask present
Bit 2	1:	R.BT656 undefined error (code bit 7-0) mask absent	0:	R.BT656 undefined error (code bit 7-0) mask present
Bit 3	1:	R.BT656 long interval H code error (SAV) mask absent	0:	R.BT656 long interval H code error (SAV) mask present
Bit 4	1:	R.BT656 long interval H code error (EAV) mask absent	0:	R.BT656 long interval H code error (EAV) mask present
Bit 5	1:	R.BT656 short interval H code error (SAV) mask absent	0:	R.BT656 short interval H code error (SAV) mask present
Bit 6	1:	R.BT656 short interval H code error (EAV) mask absent	0:	R.BT656 short interval H code error (EAV) mask present

9 Electrical Characteristics

9.1 Maximum Rating

Table 9.1.1 shows the maximum ratings.

Table 9.1.1 Maximum Ratings

Parameter	Symbol	Maximum rating	Unit
Power supply voltage	V_{DDI}(*1) V_{DDE} V_{DDE}1,2,3	-0.5 < VDDI < 1.8(*1) -0.5 < VDDE <4.0 -0.5 < VDDE1,2,3 < 4.0	V
Input voltage	V_I	-0.5 < VI < VDDI+0.5 (≤ 1.8) -0.5 < VI < VDDE +0.5 (≤ 4.0) -0.5 < VI < VDDE1,2,3 +0.5 (≤ 4.0)	V
Output voltage	V_O	-0.5 < VO < VDDI+0.5 (≤ 1.8) -0.5 < VO < VDDE +0.5 (≤ 4.0) -0.5 < VO < VDDE1,2,3 +0.5 (≤ 4.0)	V
Storage ambient temperature	TST	-55 < TST < +125	°C
Operating junction temperature	Tj	-40 < Tj < 125	°C

(*1): Internal power supply

Notes:

- Applying stress exceeding the maximum ratings (voltage, current, temperature, etc.) may cause damage to semiconductor devices. Never exceed the ratings above.
- Never connect IC outputs or I/O pins directly, or connect them to V_{DD} or V_{SS} directly; otherwise thermal destruction of elements will result, but which does not apply to pins designed to prevent signal collision.
- Provide ESD protection, such as grounding when handling the product; otherwise externally-charged electric charge flows inside the IC and discharges, which may result in damage to the circuit.
- Applying voltage higher than V_{DD} or lower than V_{SS} to I/O pins of CMOS IC, or applying voltage higher than the ratings between V_{DD} and V_{SS} may cause latch up. The latch up increases supply current, resulting in thermal destruction of elements. When handling the product, never exceed the maximum ratings.

9.2 Recommended Operating Conditions

9.2.1 3.3 V Standard CMOS I/O

Table 9.2.1 shows the recommended operating conditions for the standard CMOS.

Table 9.2.1 Recommended Operating Conditions

Parameter		Symbol	Rating			Unit
			Min.	Typ.	Max.	
Power supply voltage		V_{DDE} V_{DDI}	3.0 1.1	3.3 1.2	3.6 1.3	V
Input voltage (High level)	3.3 V CMOS	V_{IH}	2.0		V_{DDE} + 0.3	V
Input voltage (Low level)	3.3 V CMOS	V_{IL}	-0.3		0.8	V
Operating ambient temperature		T_A	-40		85	°C
Junction temperature		T_J	-40		125	°C

9.2.2 Graphics Memory I/O

Table 9.2.2 SSTL2 Recommended Operating Conditions (JEDEC JESD8-9B compliant)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	V_{DDE}1,2,3	2.3	2.5	2.7	V
	V_{DDI}	1.1	1.2	1.3	V
Reference voltage	V_{REF}	1.13	1.25	1.38	V
Termination voltage	V_{TT}	V_{REF} - 0.04	V_{REF}	V_{REF} + 0.04	V
H level input voltage (DC)	V_{IH}(DC)	V_{REF} + 0.15	—	V_{DDE} + 0.3	V
L level input voltage (DC)	V_{IL}(DC)	-0.3	—	V_{REF} - 0.15	V
H level input voltage (AC)	V_{IH}(AC)	VREF + 0.31	—	V_{DDE} + 0.3	V
L level input voltage (AC)	V_{IL}(AC)	-0.3	—	V_{REF} - 0.31	V
Junction temperature	Tj	-40	—	125	°C

Table 9.2.3 2.5 V LVCMOS Recommended Operating Conditions (JEDEC JESD8-5 compliant)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	V_{DDE}1,2,3	2.3	2.5	2.7	V
	V_{DDI}	1.1	1.2	1.3	V
H level input voltage	V_{IH}	1.7	—	V_{DDE}+0.3	V
L level input voltage	V_{IL}	-0.3	—	0.7	V
Junction temperature	Tj	-40	—	125	°C

Table 9.2.4 3.3 V LVCMOS Recommended Operating Conditions (JEDEC JESD8-5 compliant)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	V_{DDE}1,2,3	3.0	3.3	3.6	V
	V_{DDI}	1.1	1.2	1.3	V
H level input voltage	V_{IH}	2.0	—	VDDE+0.3	V
L level input voltage	V_{IL}	-0.3	—	0.8	V
Junction temperature	Tj	-40	—	125	°C

Notes:

The recommended operating conditions are primarily intended to assure the normal operation of semiconductor device. The values of electrical characteristics are guaranteed under the requirements above, so use the product accordingly. Using the product without observing the conditions may affect the product's reliability. Performance of this product is not guaranteed if used under unspecified conditions and by an unspecified combination of logic. Be sure to contact Fujitsu when using the product under such conditions.

9.3 Precautions at Power ON

9.3.1 Recommended Power ON/OFF Sequence

Follow the power ON/OFF sequence as shown below:

<ON>: V_{DDI} (including internal, PLLVDD) → V_{DDE} 1,2,3 (external) → V_{DDE} (external) → Signal

<OFF>: Signal → V_{DDE} (external) → V_{DDE} 1,2,3 (external) →V_{DDI} (including internal, PLLVDD)

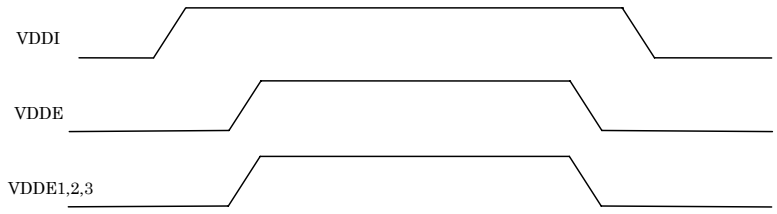


Fig. 9.1 Recommended Power ON/OFF Sequence (1)

There are no constraints for the power ON/OFF sequence of V_{DDI}, V_{DDE}, and V_{DDE} 1, 2, 3 as long as the following conditions are met: (***Fig. 9.2***)

- Do not apply V_{DDE} and V_{DDE} 1, 2, 3 (external) for more than a second if V_{DDI} (internal) is OFF.

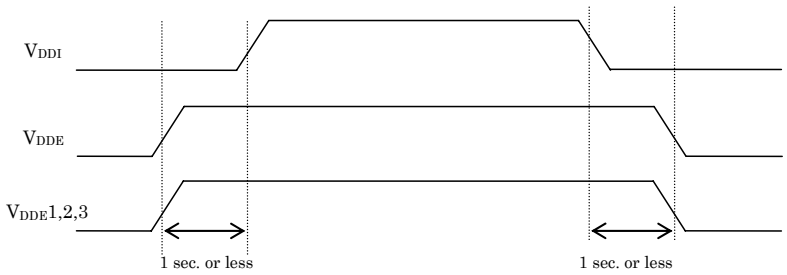


Fig. 9.2 Recommended Power ON/OFF Sequence (2)

Perform power ON/OFF for VREF and VTT according to the DDR-SDRAM specifications.

Perform power ON/OFF so that the power for PLLVDD(PLL) does not exceed VDDI.

Turn ON all powers described, do not leave some OFF.

The CMOS IC is unstable immediately after power ON. Perform a reset immediately after power ON.

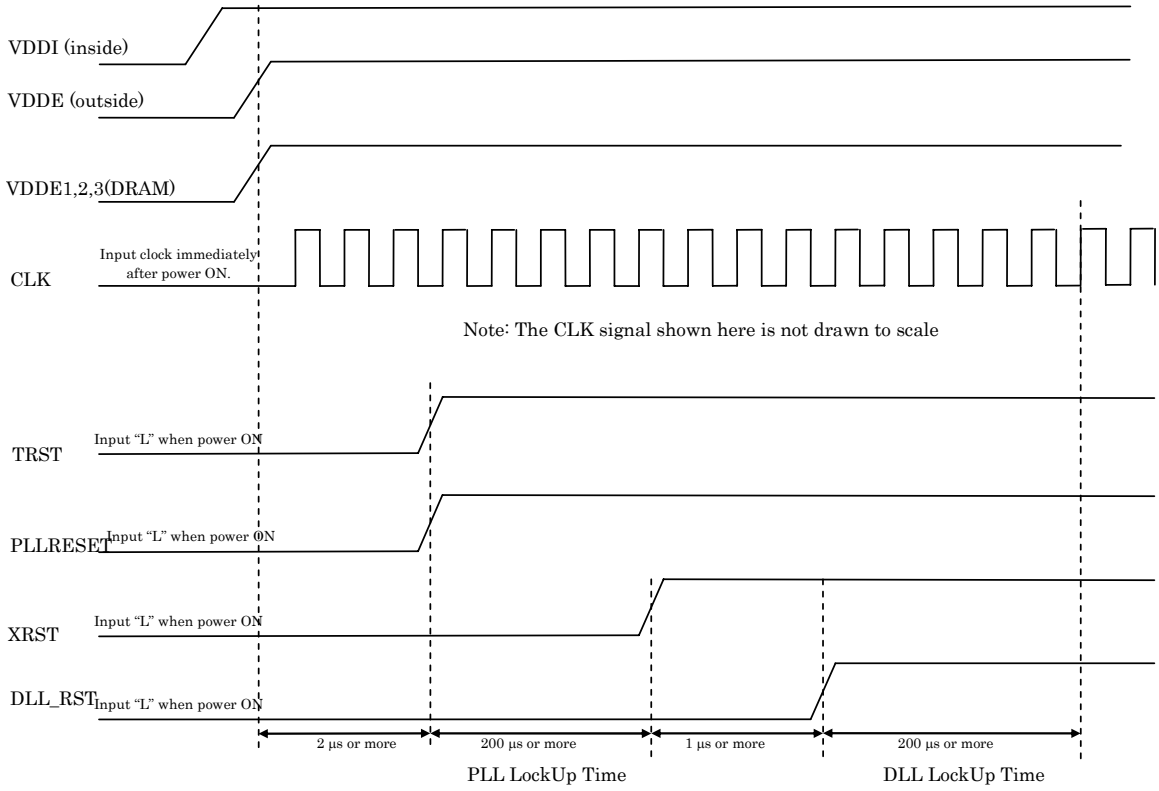
Set the reset pins (PLLRESET, TRST, XRST, and DLL_RST) to Low when powering ON.

Input a clock to CLK pin immediately after a power ON.

The XRST pin must be held Low for a minimum of 100 CLK cycles (PLL reference clock).

For stable oscillation, supply a stable clock to the PLL before cancelling the reset of the PLLRESET pin.

9.3.2 Power ON Reset



Keep the TRST, PLLRESET, XRST, DLL_RST pins Low when powering ON.

Keep the TRST and PLLRESET pins High for at least 2 µs.

XRST must be kept Low for at least 200 µs after PLLRESET has been released/deasserted.

Keep the DLL_RST pin High for at least 1 µs after the XRST signal has been released.

After this sequence, it is possible to access all registers (other than those in the memory controller).

The memory controller or external memory can be accessed after 200 µs have elapsed.

9.3.3 Reset at Normal Operation

When performing a reset via PLLRESET during normal operation, perform the reset described in ***9.3.2 Power ON Reset*** again.

XRST and DLL_RST are interdependent. There is no restriction on the reset sequence (high level to low level). However, when canceling the reset (low level to high level), follow the above power-on reset (see ***9.3.2 Power ON Reset*).**

TRST is the boundary scan pin. Do not perform a reset using TRST at normal operation.

9.4 DC Characteristics

9.4.1 3.3 V Standard CMOS I/O

Table 9.4.1 shows 3.3 V Standard CMOS I/O DC characteristics.

Table 9.4.1 Standard CMOS I/O DC Characteristics

Measurement condition: V_{DDE} = 3.3 ±0.3 V, V_{SS} = 0 V, Tj = -40 to 125°C

Parameter	Symbol	Condition		Rating			Unit
				Min.	Typ.	Max.	
H level input voltage	V_{IH}			2.0		V_{DDE} +0.3	V
L level input voltage	V_{IL}			-0.3		0.8	V
H level output voltage	V_{OH}	IOH = -100 μA		V_{DDE}-0.2		V_{DDE}	V
L level output voltage	V_{OL}	IOL = 100 μA		0		0.2	V
H level output V-I characteristic	—	Driving capability 1	IOH = 4 mA	See ***Fig. 9.3*** to ***Fig. 9.5*** V-1 characteristics.			—
		Driving capability 2	IOH = 6 mA				
		Driving capability 3	IOH = 8 mA				
L level output V-I characteristic	—	Driving capability 1	IOL = 4 mA				—
		Driving capability 2	IOL = 6 mA				
		Driving capability 3	IOL = 8 mA				
Input leakage current	I_L			—	—	±4	μA

Notes: The following shows external pin names that use standard CMOS I/O.

CLK, PLLRESET, DLL_RST, XRST, CLKSEL1-0, CKE_START,

CSYNC1-0, HSYNC1-0, VSYNC1-0,

DB0_1-0, DB1_1-0, DB2_1-0, DB3_1-0, DB4_1-0, DB5_1-0, DB6_1-0, DB7_1-0,

DG0_1-0, DG1_1-0, DG2_1-0, DG3_1-0, DG4_1-0, DG5_1-0, DG6_1-0, DG7_1-0,

DR0_1-0, DR1_1-0, DR2_1-0, DR3_1-0, DR4_1-0, DR5_1-0, DR6_1-0, DR7_1-0,

GV1-0, DE1-0, CCLK_RGB, CAP0B5-0, CAP0G5-0, CAP0R5-0, CAP0HS, CAP0VS, CAP0FID, CCLK_656, CAP1VI7-0, TCK, TRST, MODE1-0, ASEN, TCK, TDI, TDO, TMS

Driving capabilities 1 to 3 in the table above indicate the following external pins:

Driving capability 1: CAP0B5 to 0, CAP0G5 to 3, CAP0HS, CAP0VS, CAP1VI0

Driving capability 2: CAP0FID, CAP1VI6 to 4, HSYNC1, 0, VSYNC1, 0

Driving capability 3: DCLKO1-0, CSYNC1-0,

DB0_1 to 0, DB1_1 to 0, DB2_1 to 0, DB3_1 to 0, DB4_1 to 0, DB5_1 to 0, DB6_1 to 0, DB7_1 to 0, DE1 to 0, DG0_1 to 0, DG1_1 to 0, DG2_1 to 0, DG3_1 to 0, DG4_1 to 0, DG5_1 to 0, DG6_1 to 0, DG7_1 to 0,

DR0_1 to 0, DR1_1 to 0, DR2_1 to 0, DR3_1 to 0, DR4_1 to 0, DR5_1 to 0, DR6_1 to 0, DR7_1 to 0, GV1 to 0

3.3 V Standard CMOS I/O V-I Characteristic (Driving capability 1)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 3.0 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 3.3 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 3.6 V

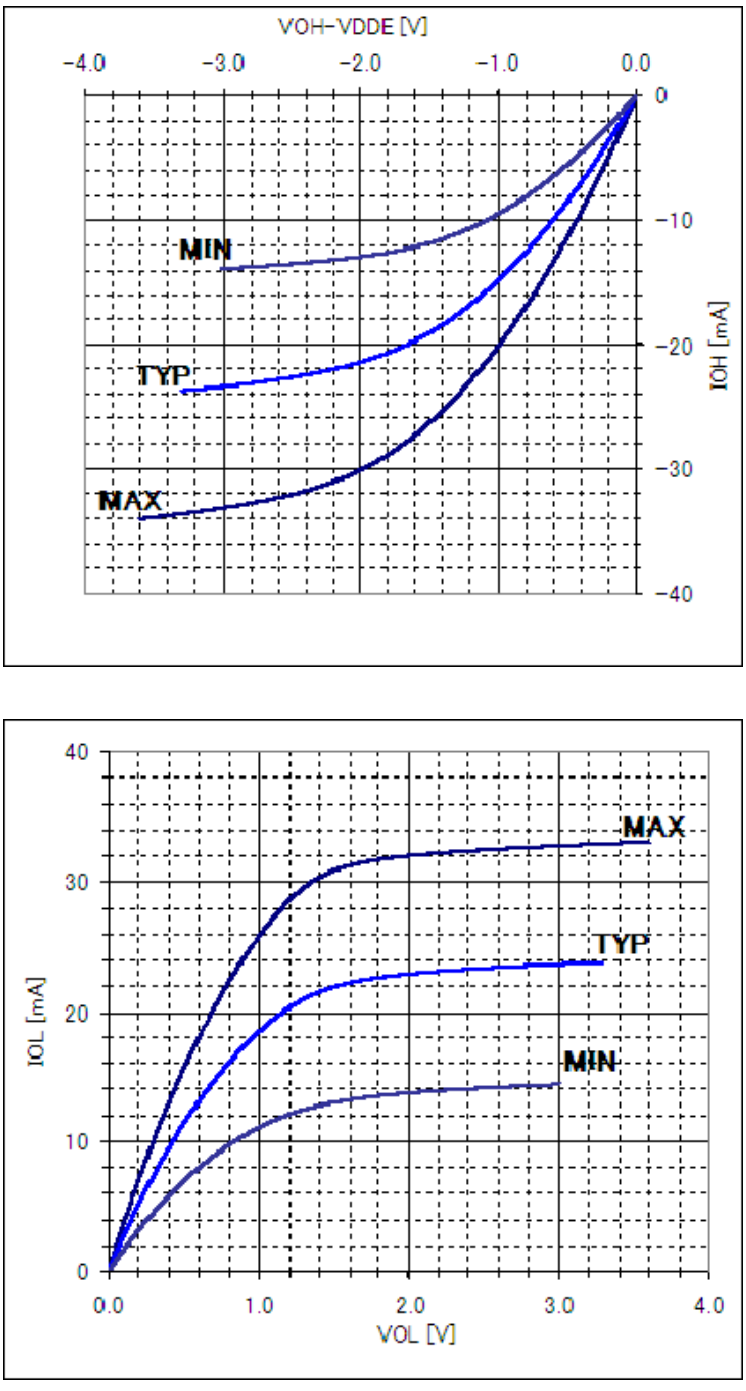


Fig. 9.3 3.3 V Standard CMOS I/O V-I Characteristic (Driving capability 1)

3.3 V Standard CMOS I/O V-I Characteristic (Driving capability 2)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 3.0 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 3.3 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 3.6 V

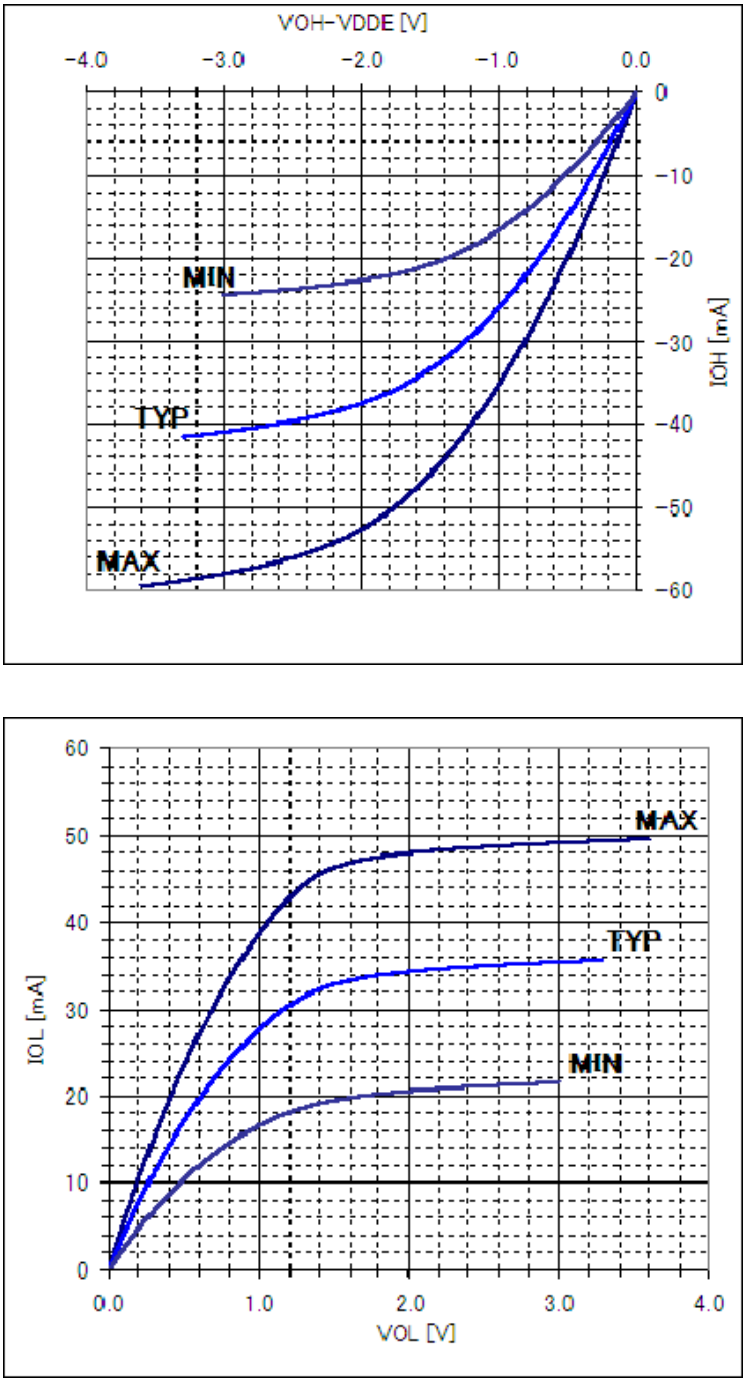


Fig. 9.4 3.3 V Standard CMOS I/O V-I Characteristic (Driving capability 2)

3.3 V Standard CMOS I/O V-I Characteristics (Driving capability 3)

Conditions	MIN: Process = Slow	Tj = 125°C	VD_{DE} = 3.0 V
	TYP: Process = Typical	Tj = 25°C	V_{DDE} = 3.3 V
	MAX: Process = Fast	Tj = -40°C	V_{DDE} = 3.6 V

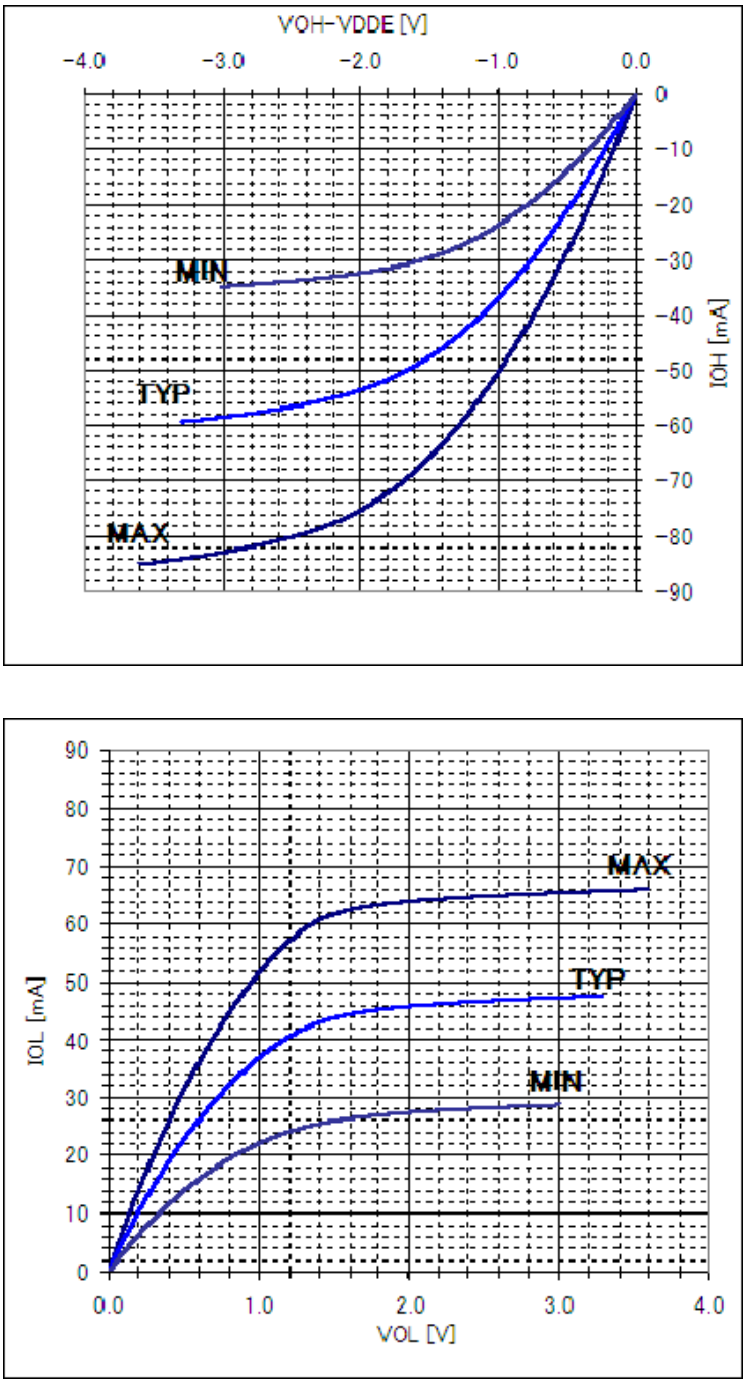


Fig. 9.5 3.3 V Standard CMOS I/O V-I Characteristic (Driving capability 3)

9.4.2 3.3 V 66 MHz PCI I/O

Table 9.4.2 shows 3.3 V 66 MHz PCI I/O DC characteristics.

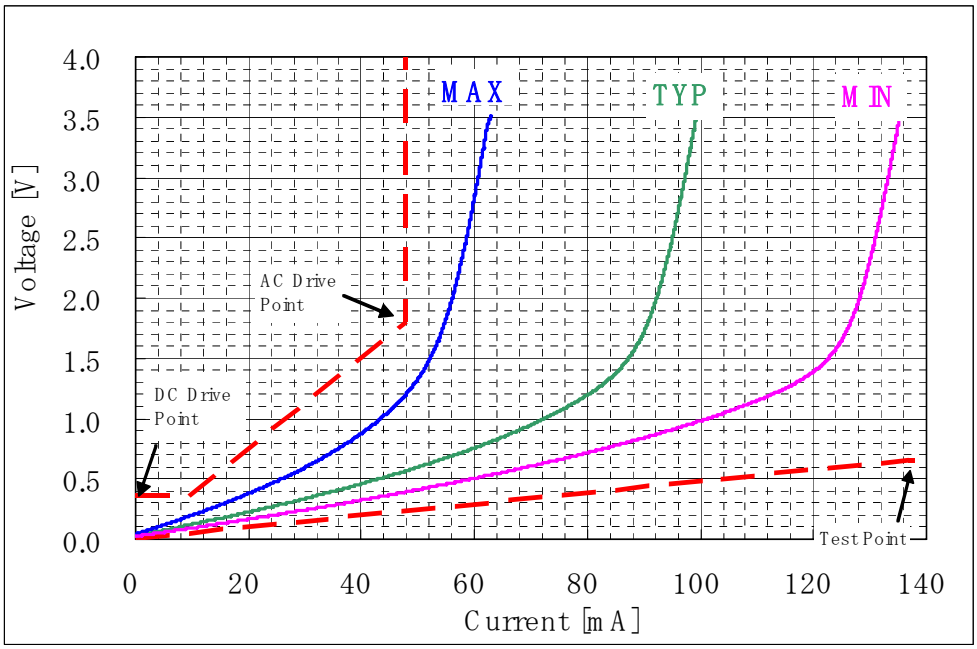
Table 9.4.2 3.3 V 66 MHz PCI I/O DC characteristics

Parameter	Symbol	Condition	Rating		Unit
			Min.	Max.	
Supply Voltage	V_{CC}		3.0	3.6	V
Input High Voltage	V_{IH}		0.5 V_{CC}	Vcc +0.5	V
Input Low Voltage	V_{IL}		-0.5	0.3 V_{CC}	V
Input Pull-up Voltage	V_{IPU}		0.7 V_{CC}		V
Input Leakage Current	I_{IL}	$0 < V_{IN} < V_{CC}$		±10	μA
Output High Voltage	V_{OH}	Iout = -0.5 mA	0.9 Vcc		V
Output Low Voltage	V_{OL}	Iout = 1.5 mA		0.1 V_{CC}	V

Note: External pins for PCI I/O are as follows.

AD31-0, PAR, DEVSEL, PERR, SERR, STOP, TRDY, XINT

PCI I/O I-V Characteristics



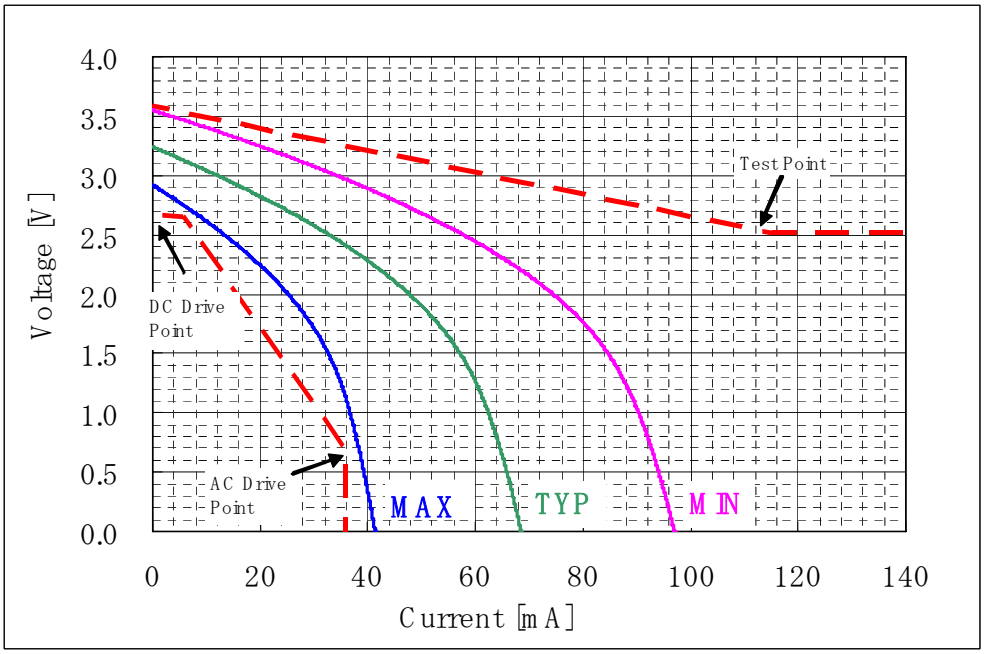


Fig. 9.6 3.3 V 66 Mz PCI I/O V-I Characteristic

9.4.3 Graphics Memory I/O

Table 9.4.3 shows SSTL2, ***Table 9.4.4*** shows 2.5V LVCMOS and ***Table 9.4.5*** shows 3.3V LVCMOS DC characteristics respectively.

Table 9.4.3 SSTL2 DC Characteristic

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
H level output voltage	V_{OH}	Driving capability 1 (∗1)	I_{OH} =-8.1 mA	1.74	—	V_{DDE}	V
		Driving capability 2	I_{OH} =-12.0 mA	1.84			
				1.94			
L level output voltage	V_{OL}	Driving capability 1 (∗1)	I_{OL} = 8.1 mA	0	—	0.56	V
		Driving capability 2	I_{OL} = 12.0 mA			0.46	
						0.36	

(∗1): Equivalent to JEDEC JESD8-9B SSTL2 CLASS-I

Table 9.4.4 2.5 V LVCMOS DC Characteristic

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
H level output voltage	V_{OH}	Driving capability 1	Equivalent to 8 mA buffer	See ***Fig. 9.7*** to ***Fig. 9.8*** V-1 characteristics.			V
		Driving capability 2	Equivalent to 12 mA buffer				
		Driving capability 3	Equivalent to 16 mA buffer				
L level output voltage	V_{OL}	Driving capability 1	Equivalent to 8 mA buffer				V
		Driving capability 2	Equivalent to 12 mA buffer				
		Driving capability 3	Equivalent to 16 mA buffer				

Table 9.4.5 3.3 V LVCMOS DC Characteristic

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
H level output voltage	V_{OH}	Driving capability 1	Equivalent to 8 mA buffer	See ***Fig. 9.10*** V-1 characteristic figures.			V
		Driving capability 2	Equivalent to 12 mA buffer				
L level output voltage	V_{OL}	Driving capability 1	Equivalent to 8 mA buffer				V
		Driving capability 2	Equivalent to 12 mA buffer				

Notes:

1. External pins for graphic memory IO Buffer are as follows.

   MCK_1 to 0, XMCK_1 to 0, MDQ63 to 0, MDQS7 to 0, MDM7 to 0, MA13 to 0, MBA1 to 0, MCKE, MCS, MRAS, MCAS, MWE, LOOP1 to 0, LOOPI1 to 0, CKE_START, DLL_RST

2. Fujitsu recommends “driving capability 2”. (For details, refer to ***Guidelines for PCB Design***)
3. Driving capability of each external pin cannot be changed.
4. When using with other than “driving capability 2”, contact Fujitsu.

2.5 V LVCMOS V-I Characteristic (Driving capability 1)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 2.3 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 2.5 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 2.7 V

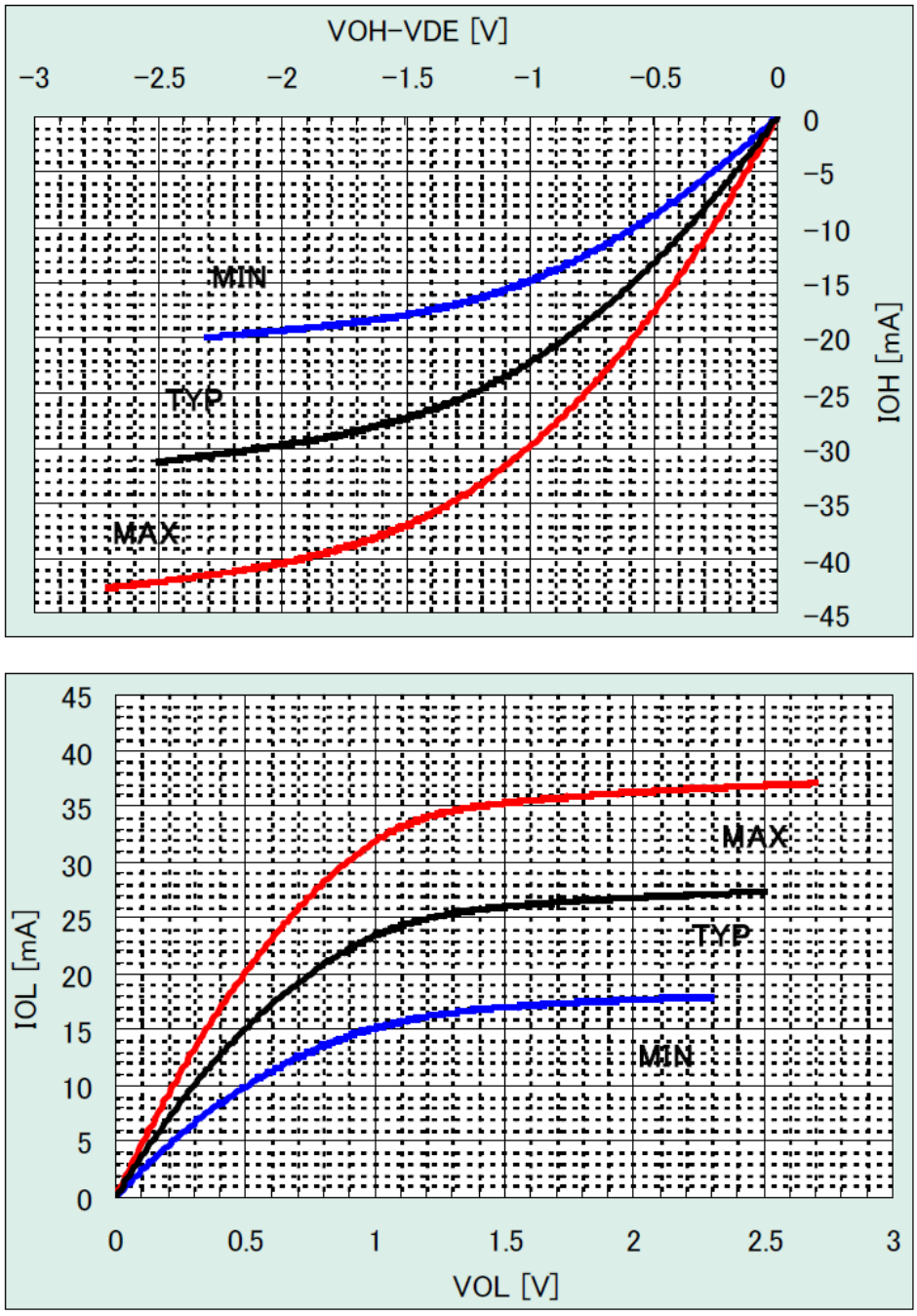


Fig. 9.7 2.5 V LVCMOS V-I Characteristic (Driving capability 1)

2.5 V LVCMOS V-I Characteristic (Driving capability 2)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 2.3 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 2.5 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 2.7 V

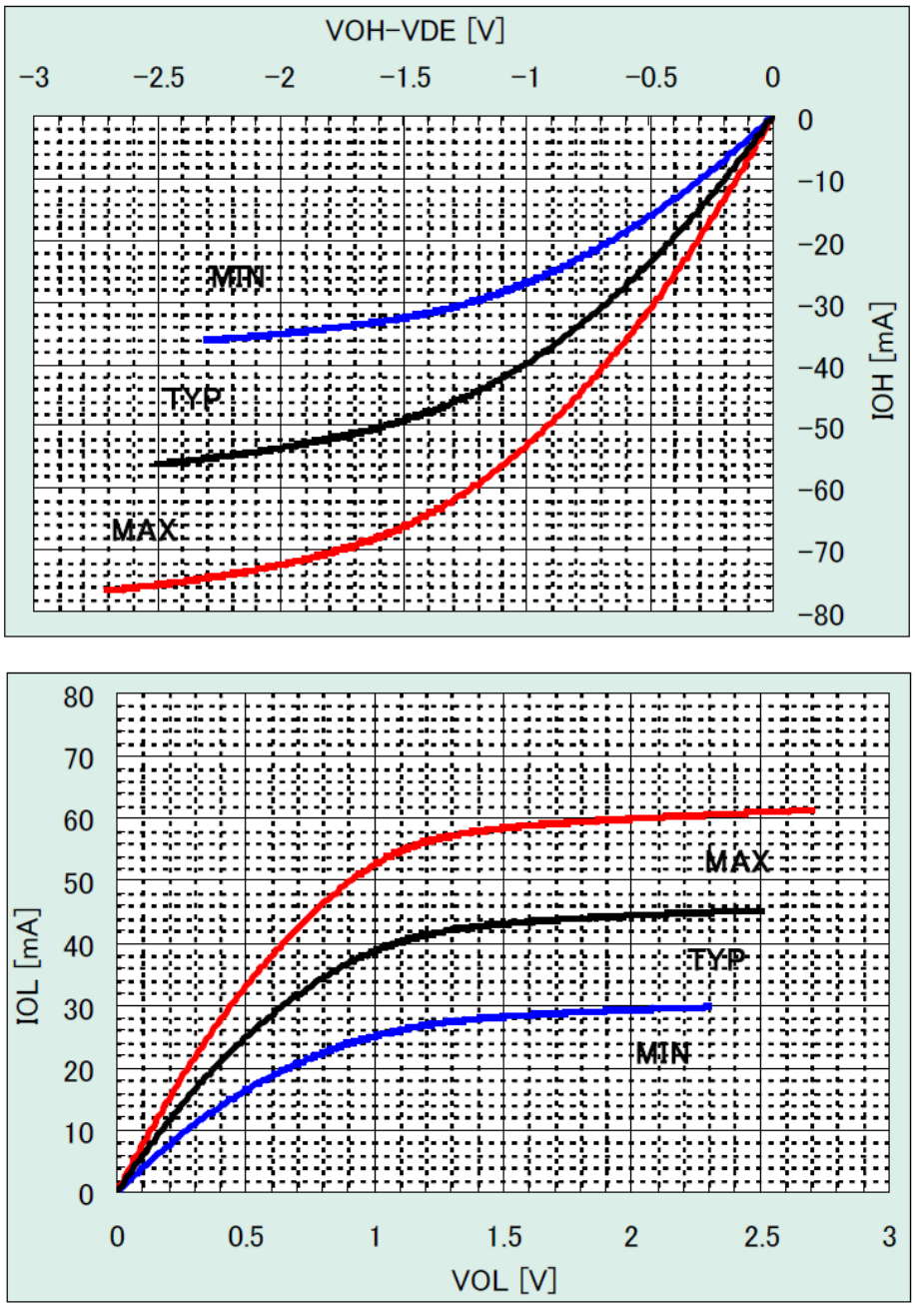


Fig. 9.8 2.5 V LVCMOS V-I Characteristic (Driving capability 2)

3.3 V LVCMOS V-I Characteristic (Driving capability 1)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 3.0 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 3.3 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 3.6 V

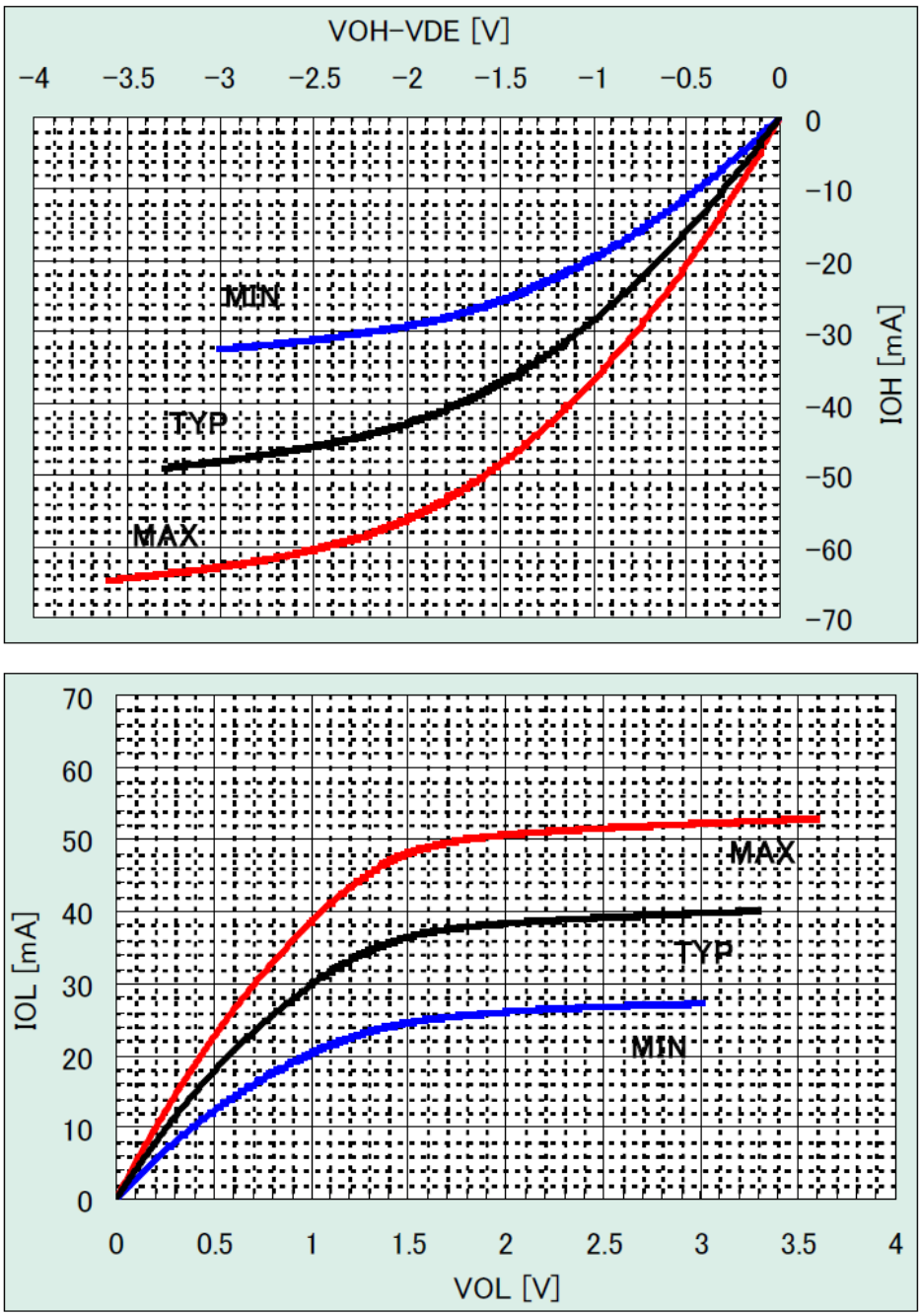


Fig. 9.9 3.3 V LVCMOS V-I Characteristic (Driving capability 1)

3.3 V LVCMOS V-I Characteristic (Driving capability 2)

Conditions	MIN:	Process = Slow	Tj = 125°C	V_{DDE} = 3.0 V
	TYP:	Process = Typical	Tj = 25°C	V_{DDE} = 3.3 V
	MAX:	Process = Fast	Tj = -40°C	V_{DDE} = 3.6 V

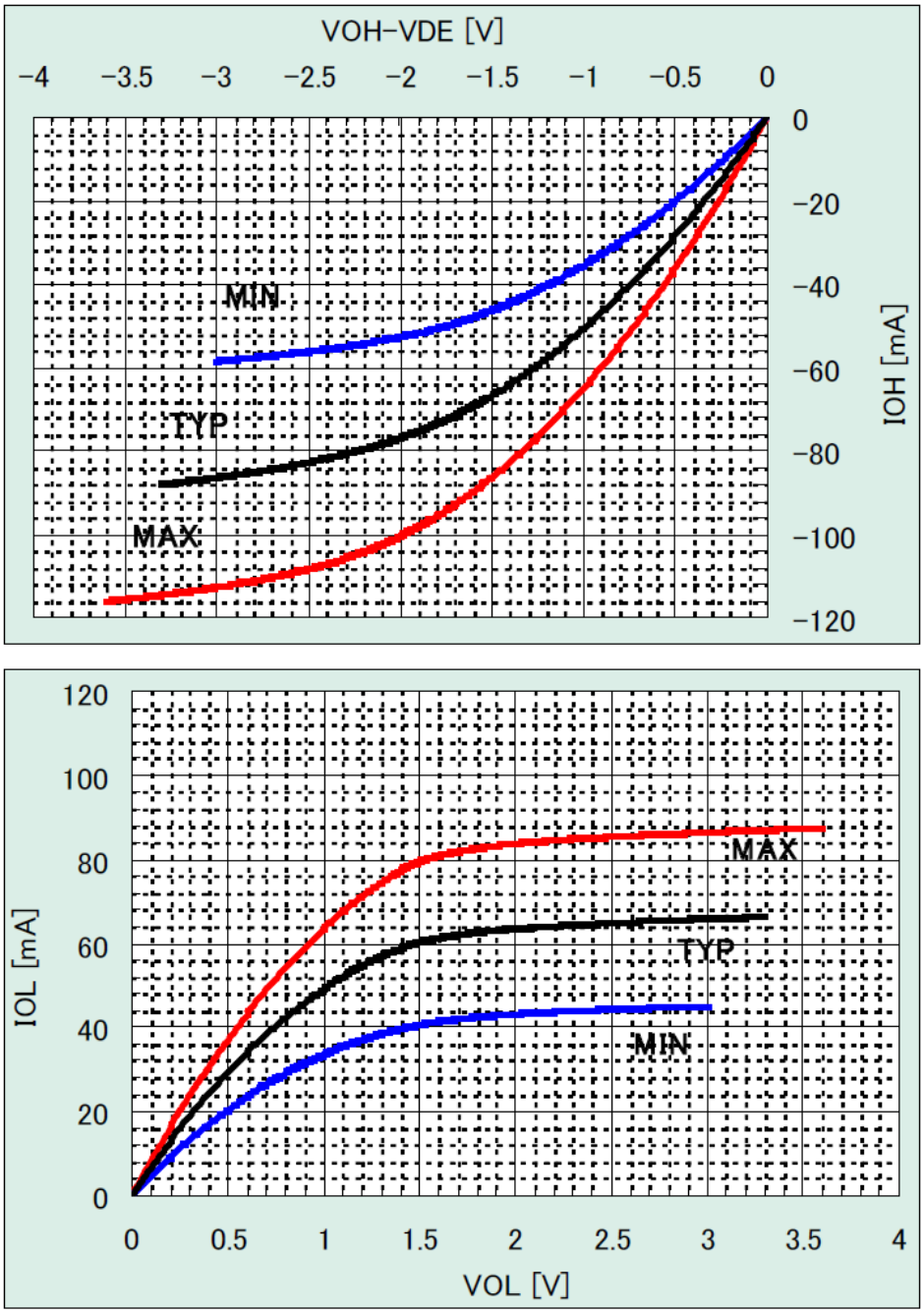


Fig. 9.10 3.3 V LVCMOS V-I Characteristic (Driving capability 2)

9.4.4 I²C Bus Fast Mode I/O

Table 9.4.6 I²C I/O DC Characteristic

Parameter & Condition	Symbol	Standard Mode(1)		Fast Mode		単位
		MIN	MAX	MIN	MAX	
"L" Level Input Voltage	VIL	-0.5	0.3VDDE	-0.5	0.3VDDE	V
"H" Level Input Voltage	VIH	0.7VDDE	(2)	0.7VDDE	(2)	V
Schmitt Trigger Hysterisys VDDE > 2[v]	Vhys	n/a	n/a	0.05VDDE	-	V
"L" Level Output Voltage Sink Current 3 [mA] VDDE > 2[v]	VOL1	0	0.4	0	0.4	V
Output Slew Rate (Tfall) Bus Capacitance 10[pF] ~ 400[pF] VIHmin to VILmax	tof	-	250 (3)	20+0.1Cb (4)	250 (3)	ns
Data Line Leakage Input Voltage 01. ~ 0.9VDDEmax	Ii	-10	10	-10	10	μA
I/O pin Capacitance	Ci	-	10	-	10	pF

(1) The I²C Bus Fast Mode I/O Buffer is downward compatible with Standard Mode.

(2) 90 nm Technology: Complies with the maximum ratings 4 [v].

(3) The maximum Tf (300 ns) of SDA and SCL bus lines shown in ***Table 9.5.16*** is greater than the maximum tof (250 ns) at output level. In this case, the series protection resistance can be connected between bus lines of SDA and SCL without exceeding the maximum rating Tf.

(4) Cb : Capacitance for one bus line (Unit: pF).

(5) The I²C Bus Fast Mode I/O Buffer itself has no function to prevent a spike of 50 ns pulse width (max.). Therefore, provide any input filter to prevent a spike for both internal or external semiconductor device.

Remark:

External pins for I²C IO Buffer are as follows.

SCL, SDA

I²C IO V-1 Characteristic Figure

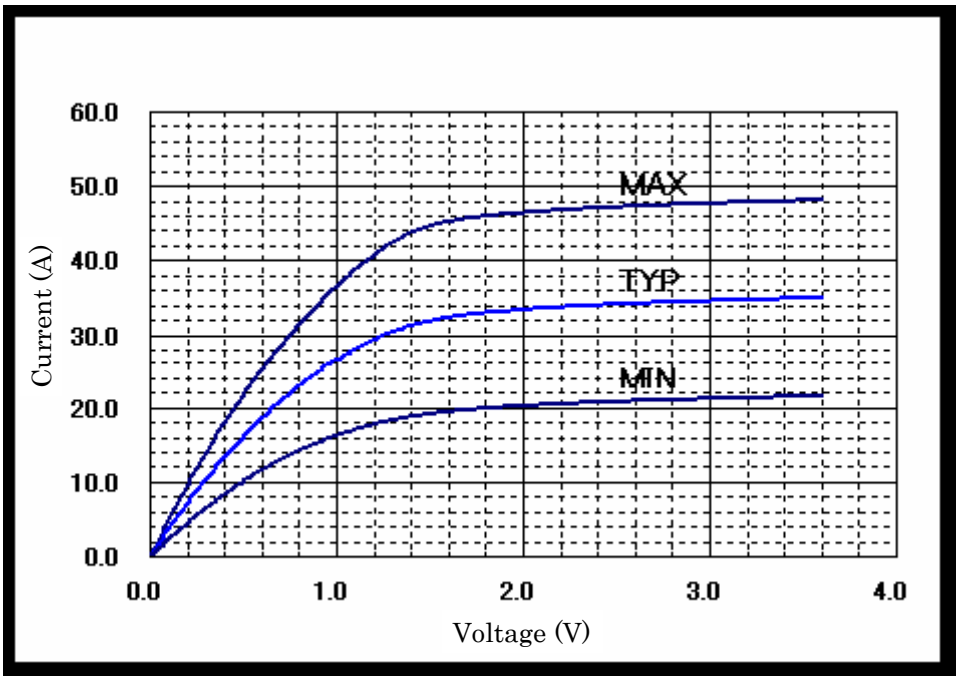


Fig. 9.11 I²C V-I Characteristic Figure

9.5 Alternate Current (AC) Characteristics

9.5.1 PCI Interface

Table 9.5.1 PCI Interface

Parameter	Signal	Abbrev.	Values			Unit
			Min	Typ	Max	
PCI Clock Period	PCLK	t_{cyc}	15			ns
PCI Clock Low Time	PCLK	t_{low}	6			ns
PCI Clock High Time	PCLK	t_{hi}	6			ns
PCI Input Setup (bussed signals)	AD[31:0], C/BE[3:0], PAR, FRAME, IRDY, IDSEL	t_{su}	3			ns
PCI Input Hold	AD[31:0], C/BE[3:0], PAR, FRAME, IRDY, IDSEL	t_h	0			ns
PCI Output Delay	AD[31:0], PAR, TRDY, STOP, DEVSEL, PERR, SERR	t_{val} *1)	1.3		6	ns
Output load conditions		-	10	-	30	pF

*1. Caution when using pins according to PCI-33 MHz specification

The PCI-66 MHz specification also requires Tval = 2.0 ns (Min.) to maintain compatibility with the 33 MHz specification. This is because the 33 MHz specification allows up to 2.0 ns for clock skew between PCI devices, although the 66 MHz specification defines the clock skew as 1.0 ns.

Carmine does not yet support this specification. To use according to the PCI-33 MHz specification, reduce the clock skew between PCI devices to less than 1.3 ns or insert a delay to satisfy Tval = 2.0 ns.

9.5.2 Display Interface

Common to UNIT0, UNIT1.

Table 9.5.2 Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
DCLKI frequency	f_{DCLKI}	*1			106	MHz
DCLKI H duration	t_{HDCLKI}		3			ns
DCLKI L duration	t_{LDCLKI}		3			ns
DCLKO frequency	f_{DCLKO}	*1			106	MHz

(*1) Depends on display resolution.

Table 9.5.3 Input Signal

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
HSYNC input pulse width	$t_{WHSYNC0}$	*1	3			clock
	$t_{WHSYNC1}$	*2	3			clock
HSYNC input setup time	t_{SHSYNC}	*2	6			ns
HSYNC input hold time	t_{HHSYNC}	*2	1			ns
VSYNC input pulse width	$t_{WHSYNC1}$		1			HSYNC 1 cycle

(*1) Applied for only PLL synchronization mode. Reference clock is internal PLL output.

(*2) Applied for only DCLKI synchronization mode. Reference clock is DCLKI.

Table 9.5.4 Output Signal (typical)

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
RGB output delay time	t_{RGB1}	*1				ns
		*2	1		7.2	
DISPE output delay time	t_{DEO1}	*1				ns
		*2	1		7.2	
HSYNC output delay time	$t_{DHSYNC1}$	*1				ns
		*2	1		7.2	
VSYNC output delay time	$t_{DVSYNC1}$	*1				ns
		*2	1		7.2	
CSYNC output delay time	$t_{DCSYNC1}$	*1				ns
		*2	1		7.2	
GV output delay time	t_{DGV1}	*1				ns
		*2	1		7.2	

Commonly applicable conditions: single display (Mden = 0) & (DCKed = 0) & (DCKinv = 0)

(*1) Load capacitance 10 pF

(*2) Load capacitance 20 pF

Table 9.5.5 Output Signal (inversion)

Item	Symbol	Conditions	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time	t_{RGB2}	*1				ns
		*2	1		7.4	
DISPE Output delay time	t_{DEO2}	*1				ns
		*2	1		7.4	
HSYNC Output delay time	$t_{DHSYNC2}$	*1				ns
		*2	1		7.4	
VSYNC Output delay time	$t_{DVSYNC2}$	*1				ns
		*2	1		7.4	
CSYNC Output delay time	$t_{DCSYNC2}$	*1				ns
		*2	1		7.4	
GV Output delay time	t_{DGV2}	*1				ns
		*2	1		7.4	

Commonly applicable conditions:

Single display (MDen = 0) & (DCKed = 0) & (DCKinv = 1)

Dual display (MDen = 1) & (DCKed = 0) & (DCKinv = 1)

(*1) Load capacitance 10 pF

(*2) Load capacitance 20 pF

Table 9.5.6 Output Signal (both edges)

Item	Symbol	Conditions	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time	t_{RGB3}	*1				ns
		*2	0		7.6	
DISPE Output delay time	t_{DEO3}	*1				ns
		*2	0		7.6	
HSYNC Output delay time	$t_{DHSYNC3}$	*1				ns
		*2	0		7.6	
VSYNC Output delay time	$t_{DVSYNC3}$	*1				ns
		*2	0		7.6	
CSYNC Output delay time	$t_{DCSYNC3}$	*1				ns
		*2	0		7.6	
GV Output delay time	t_{DGV3}	*1				ns
		*2	0		7.6	
RGB Output delay time	t_{RGB4}	*1				ns
		*2	0		7.6	
DISPE Output delay time	t_{DEO4}	*1				ns
		*2	0		7.6	
HSYNC Output delay time	$t_{DHSYNC4}$	*1				ns
		*2	0		7.6	
VSYNC Output delay time	$t_{DVSYNC4}$	*1				ns
		*2	0		7.6	
CSYNC Output delay time	$t_{DCSYNC4}$	*1				ns
		*2	0		7.6	
GV Output delay time	t_{DGV4}	*1				ns
		*2	0		7.6	

Commonly applicable conditions: dual display (MDen = 1) & (DCKed = 1)

(*1) Load capacitance 10 pF

(*2) Load capacitance 20 pF

9.5.3 Video Capture Interface

UNIT0, UNIT1 are common.

Table 9.5.7 Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
CCLK_656, CCLK_RGB frequency	F_{CCLK}	(*1)			80	MHz
CCLK_656, CCLK_RGB H period	t_{HCCLK}		3			ns
CCLK_656, CCLK_RGB L period	t_{LCCLK}		3			ns

(*1) Depends on video source resolution.

Table 9.5.8 Input Signal

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
VI,RGB input setup time	t_{SVI}		6			ns
VI,RGB input hold time	t_{HVI}		1			ns
CAP0HS input setup time	t_{SHSI}		6			ns
CAP0HS input hold time	t_{HHSI}		1			ns
CAP0VS input setup time	t_{SVSI}		6			ns
CAP0VS input hold time	t_{HVSI}		1			ns
CAP0FID input setup time	t_{HCEN}		6			ns
CAP0FID input hold time	t_{HCEN}		1			ns

9.5.4 Graphics Memory Interface

Connection to DDR-SDRAM conforming to JEDEC (JESD79D) DDR266 is supported.

The timing regulation in Carmine is described below. The output load conditions become the Guidelines for PCB Design.

Table 9.5.9 Write Spec (1, 2): CK-CMD/ADD,CK-DQS

Item	Symbol	SPEC calculation expression	Rating (*1)			Unit
			Min.	Typ.	Max.	
CMD/ADD Setup Valid-Data from CK↑	tVD_setup_CMD	(tCK/2)-1891	1859			ps
CMD/ADD Hold Valid-Data from CK↑	tVD_hold_CMD	(tCK/2)-661	3089			ps
Skew between DQS↑.vs. CK↑	tSkew_DQS_CK	Not dependent on tCK	-778		998	ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

Table 9.5.10 Write Spec (3): DQ-DQS

Item	Symbol	SPEC calculation expression	Rating (*1)			Unit
			Min.	Typ.	Max.	
DQ/DM Setup Valid-Data from DQS	tVD_setup_DQ	(tCK/4)-1170	705			ps
DQ/DM Hold Valid-Data from DQS	tVD_hold_DQ	(tCK/4)-1132	743			ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

Table 9.5.11 Read Spec (1): DQ-DQS

Item	Symbol	SPEC calculation expression	Rating (*1)			Unit
			Min.	Typ.	Max.	
tSETUP DQ from DQS	tSETUP_DQ	- (0.1875*tCK - 525)	-881			ps
tHOLD DQ from DQS	tHOLD_DQ	0.1875*tCK + 979	2385			ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

Table 9.5.12 Read Spec (2): DQ-R.T.T(RoundTrip Time)

Item	Symbol	SPEC calculation expression	Rating (*1)(*2)			Unit
			Min.	Typ.	Max.	
DQS RoundTripTime @CL=2or3 (CK_out ⇒ DRAM ⇒ DQS_in)	tRTT_DQS	<Max> tRTT_LBCK@Max+ tSkew_LP_CK @Max <Min> tRTT_LBCK@Min+ tSkew_LP_CK @Min (However, Min. when CL = 2 or 3 is fixed value)	-1740		+2813	ps
DQS RoundTripTime @CL=2.5	tRTT_DQS		-2469		+2813	ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

(*2) The spec indicates the total delay value including the tDQSCK delay of DRAM.

Table 9.5.13 Read Spec (3):

Item	Symbol	SPEC calculation expression	Rating (*1)			Unit
			Min.	Typ.	Max.	
LOOP RountTripTime (LOOP ⇒ LOOPI)	tRTT_LBCK	Min: - (0.25*tCK - 778) Max: 0.75*tCK - 4141	-1097		+1484	ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

Table 9.5.14 Read Spec (4):

Item	Symbol	SPEC calculation expression	Rating (*1)			Unit
			Min.	Typ.	Max.	
Skew Time between LOOP_RTT.vs. DQS_RTT	tSkew_LP_CK @Max = tRTT_DQS(Max) - tRTT_LBCK (Min) tSkew_LP_CK @Min = tRTT_DQS(Min) - tRTT_LBCK (Max)	Min: - (0.25*tCK - 503) Max: 0.75*tCK - 4296	-1372		+1329	ps

(*1) This is the SPEC for when tck = 7.5 ns (266 Mbps).

9.5.5 I²C Interface

Table 9.5.15 I²C Bus Timing

Parameter	Symbol		Minimum	Maximum	Unit
T_{S2SDAI}	SDA(I) setup time	standard	250		ns
		high-speed	100		ns
T_{H2SDAI}	SCL(I) hold time	standard	0		ns
		high-speed	0		ns
T_{CSCLI}	SCL(I) cycle time	standard	10.0		us
		high-speed	2.5		us
T_{WHSCLI}	SCL(I) H period	standard	4.0		us
		high-speed	0.6		us
T_{WLSCLI}	SCL(I) L period	standard	4.7		us
		high-speed	1.3		us
T_{CSCLO}	SCL(O) cycle time	standard	2*m+2 (*2)		PCLK *1
		high-speed	int(1.5*m)+2 (*2)		PCLK *1
T_{WHSCLO}	SCL(O) H period	standard	m+2(*2)		PCLK *1
		high-speed	int(0.5*m)+2 (*2)		PCLK *1
T_{WLSCLO}	SCL(O) L period	standard	m(*2)		PCLK *1
		high-speed	m(*2)		PCLK *1
T_{W2SCLI}	SCL(I) setup time	standard	4.0		us
		high-speed	0.6		us
T_{H2SCLI}	SCL(I) hold time	standard	4.7		us
		high-speed	1.3		us
T_{WBFI}	bus free time	standard	4.7		us
		high-speed	1.3		us
T_{S2SCLO}	SCL(O) set up time	standard	m+2(*2)		PCLK *1
		high-speed	int(0.5*m)+2 (*2)		PCLK *1
T_{H2SCLO}	SCL(O) hold time	standard	m-2(*2)		PCLK *1
		high-speed	int(0.5*m)-2 (*2)		PCLK *1
T_{H2SDAO}	SDA(O) hold time		5		PCLK *1

*1: PLCK is the bus clock to the I2C functional module. (PLCK is 66 MHz when PLL = 533 MHz.)

*2: See the m value of the extended CS register (CSR) of the I2C interface.

Table 9.5.16 SDA/SCL Bus Line

Item	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
SCL Clock frequency	fSCL	0	100	0	400	kHz
SDA, SCL Signal Rise Time	tr	-	1000	20+0.1Cb (*1)	300	ns
SDA, SCL Signal Fall Time	tf	-	300	20+0.1Cb (*1)	300	ns
SDA, SCL Bus Capacitance	Cb	-	400	-	400	pF

(*1) Cb: Total capacitance (unit: pF) of one bus line

9.5.6 Clock Reset

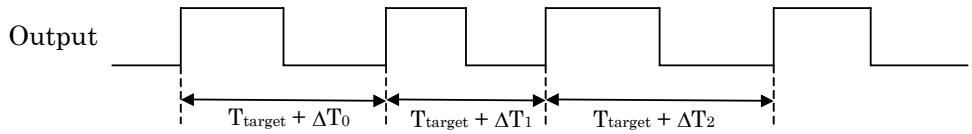
Table 9.5.17 PLL Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
CLK clock frequency	t_{PLLA1}		13.5		33.33	MHz
CLK pulse width	t_{PLLW}		5			ns
CLK Rising/falling time	tr/tf					ns

Table 9.5.18 PLL Specification

Parameter	Rating	Description
Input frequency	13.5 to 33 MHz (Max)	
Output frequency	400 to 533 MHz (Max)	
Duty ratio	45 to 55%	Duty=(High pulse width)/(Period)*100 [%]
Jitter	±47ps	Period Jitter（output cycle variation）
Lock up time	200 μs	

- The specification is for when the power is in an ideal state (no noise).
- The jitter is the maximum value of variation in the PLL output clock cycle. Jitter is defined as follows. It is the maximum value of the time variation ΔTn (n = 0, 1, 2 …) of the required cycle T_{target}.



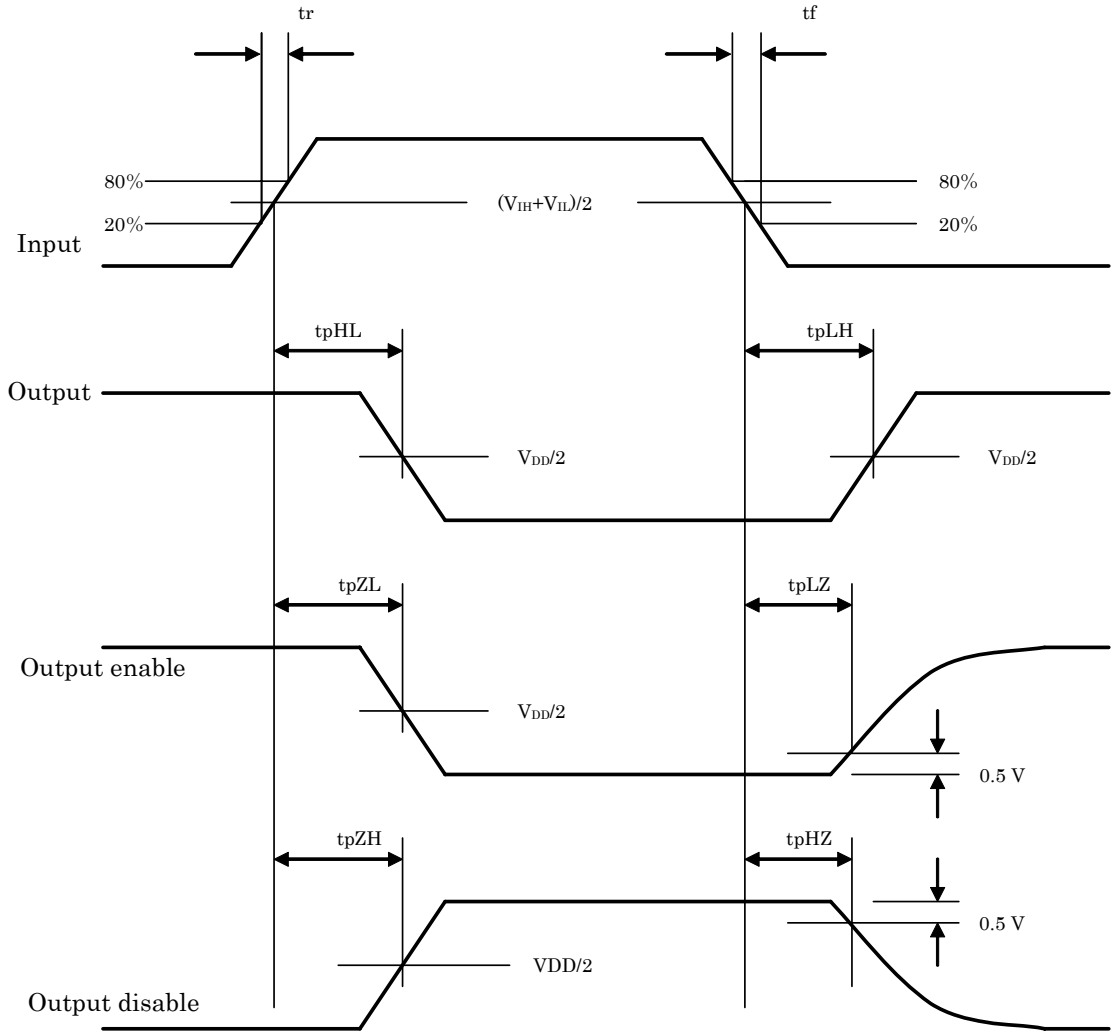
- The N-cycle jitter is calculated using the following expression: jitter × √N.

Table 9.5.19 Reset

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
PLLRESET enable time	t_{PLLRST}	(*1)	2			μs
TRST enable time	t_{TRST}	(*1)	1			μs
XRST enable time	t_{XRST}	(*1)	100			CLK

(*1) Follow the precautions in ***9.3 Precautions at Power ON*** when power is on.

9.6 AC Characteristics Measurement Conditions



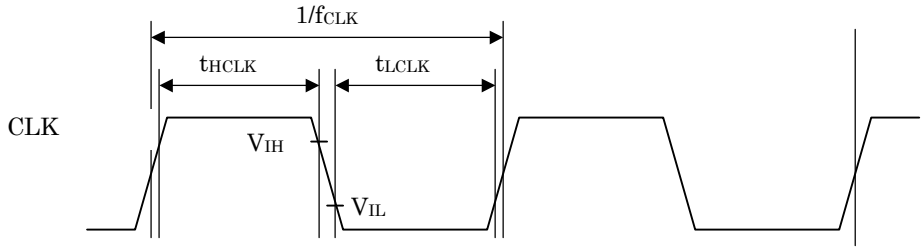
tr, tf ≤ 5 ns

V_{IH} = 2.0 V, V_{IL} = 0.8 V (3.3 V CMOS interface input)

9.7 Timing Diagram

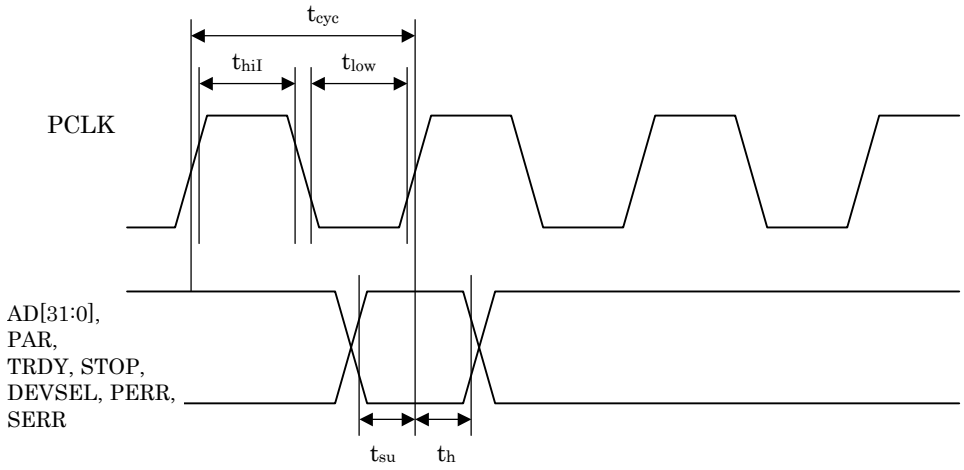
9.7.1 PLL Clock

Clock

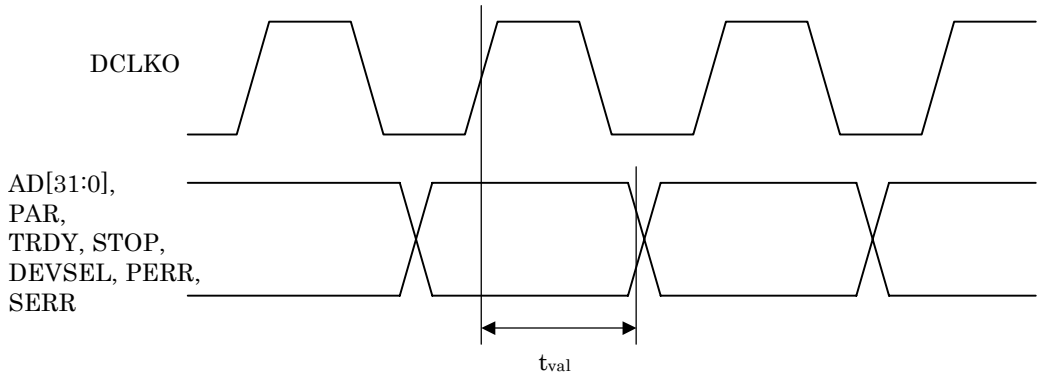


9.7.2 PCI Interface

PCI Clock/Input

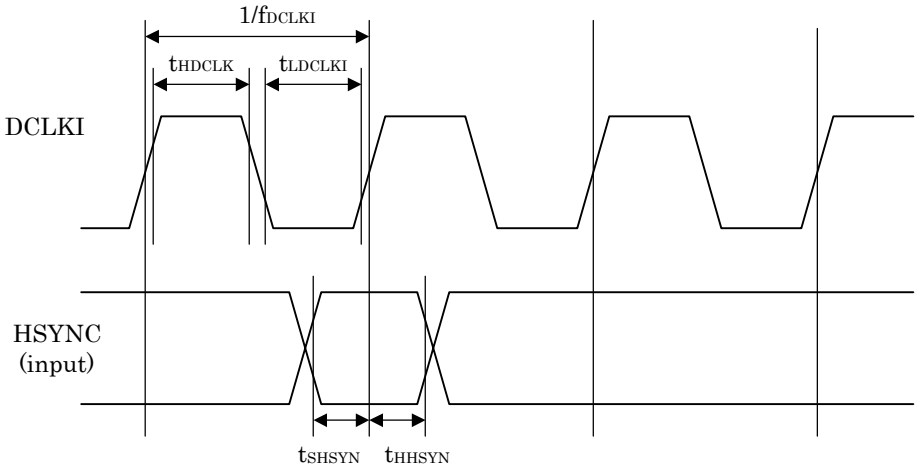


PCI Output

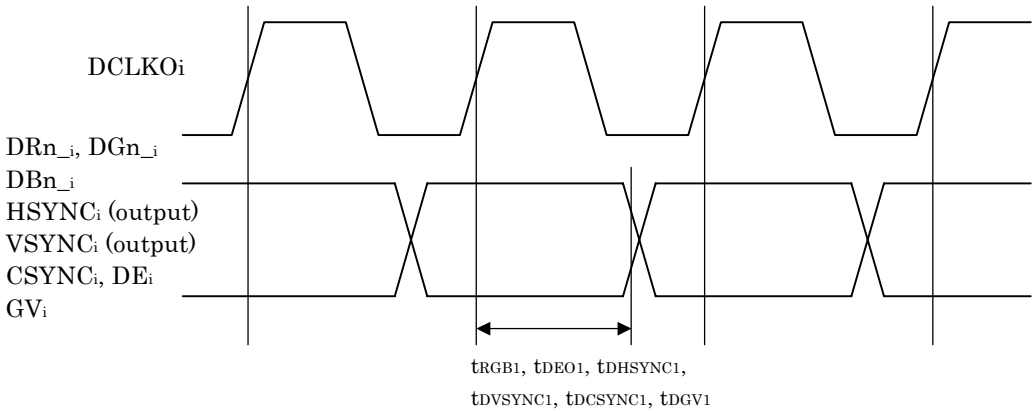


9.7.3 Display Interface

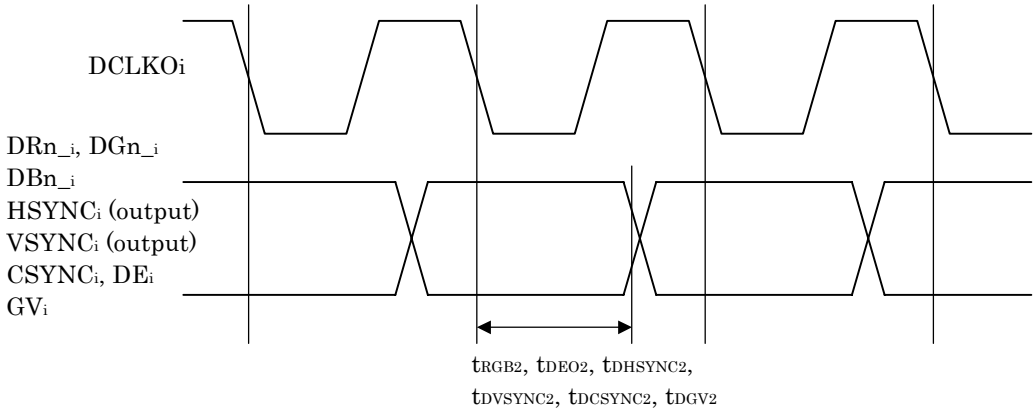
HSYNC signal setup/hold



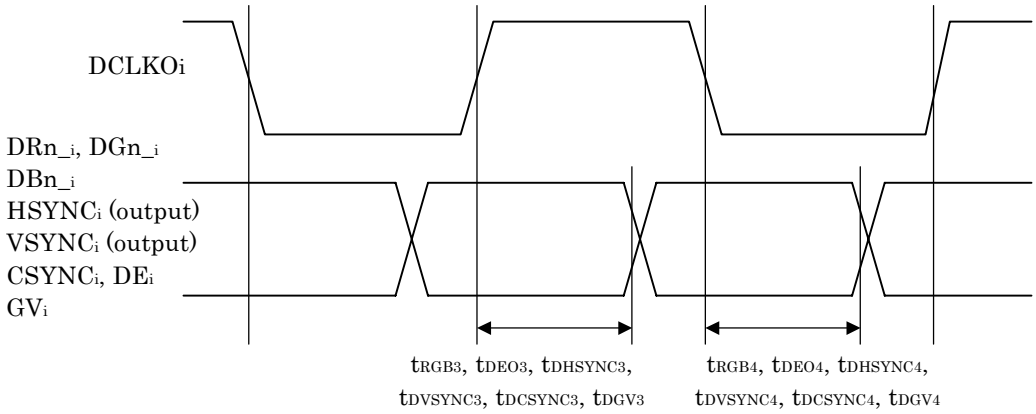
Output signal delay (standard)



Output signal delay (inverted)

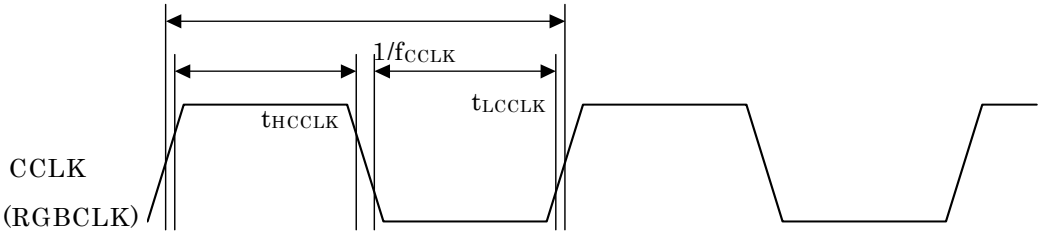


Output signal delay (bi-edge)

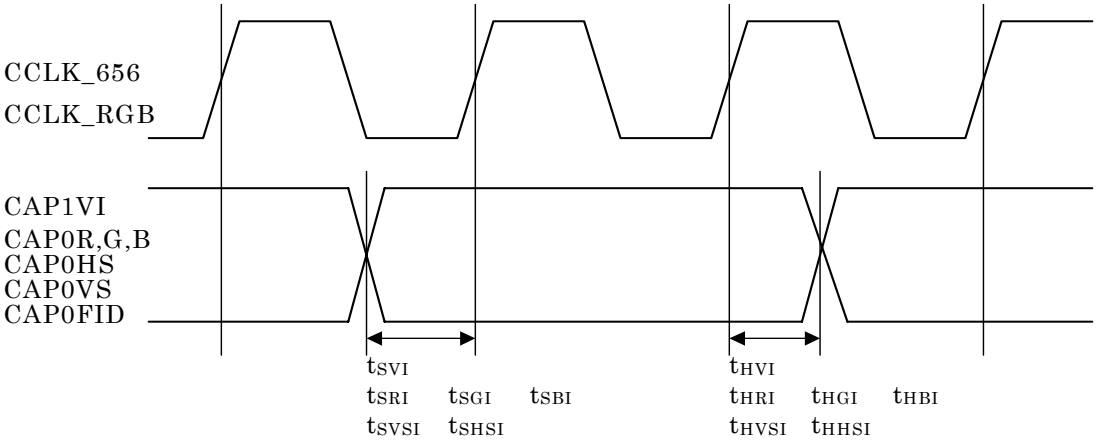


9.7.4 Video Capture Interface

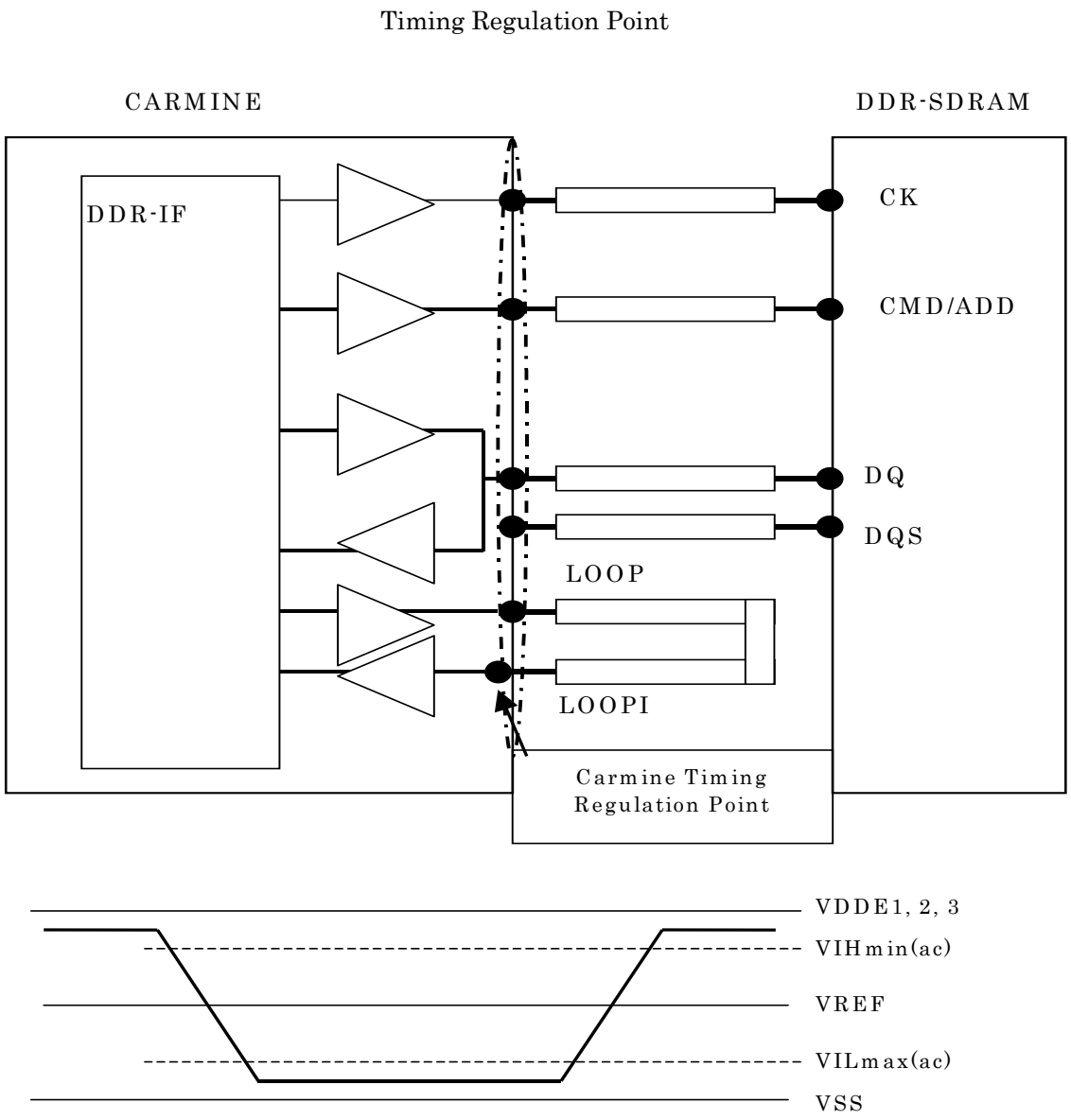
clock



Video input

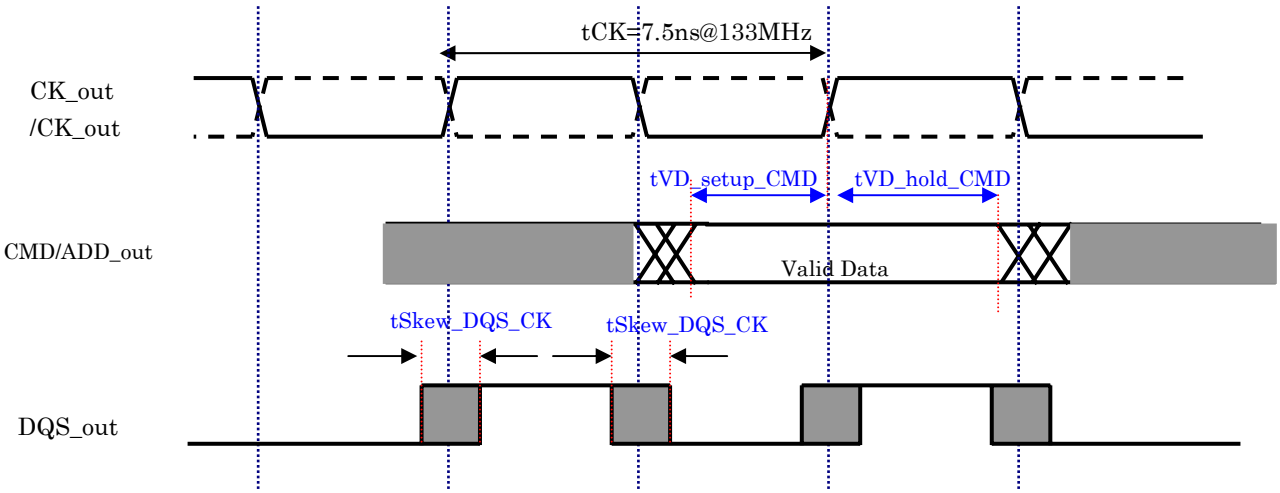


9.7.5 Graphics Memory Interface

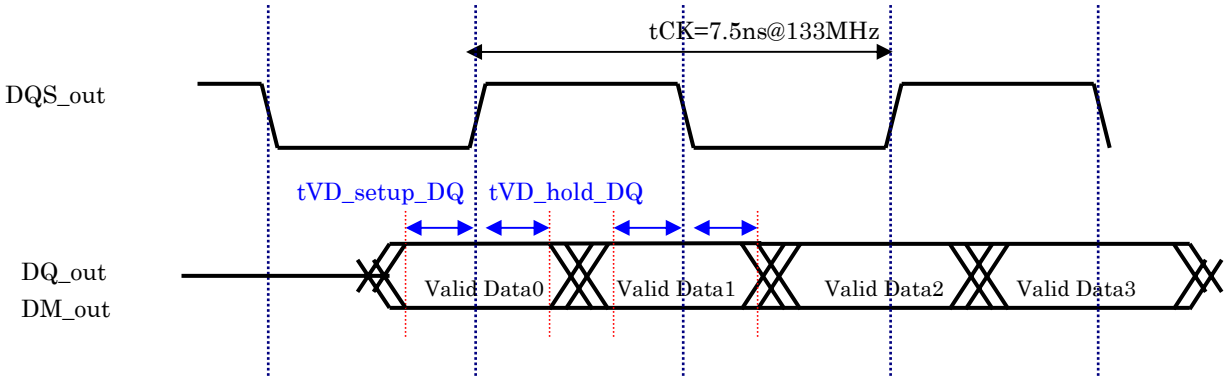


External load conditions: Guidelines for PCB Design

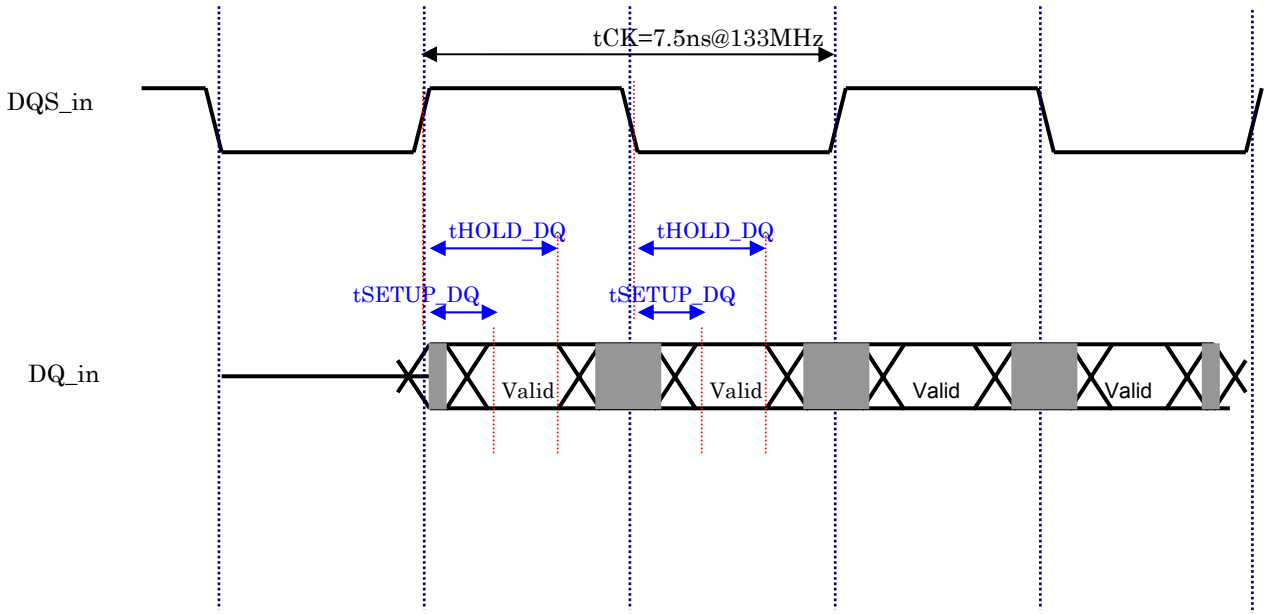
Write Spec (1,2) : CK-CMD/ADD,CK-DQS



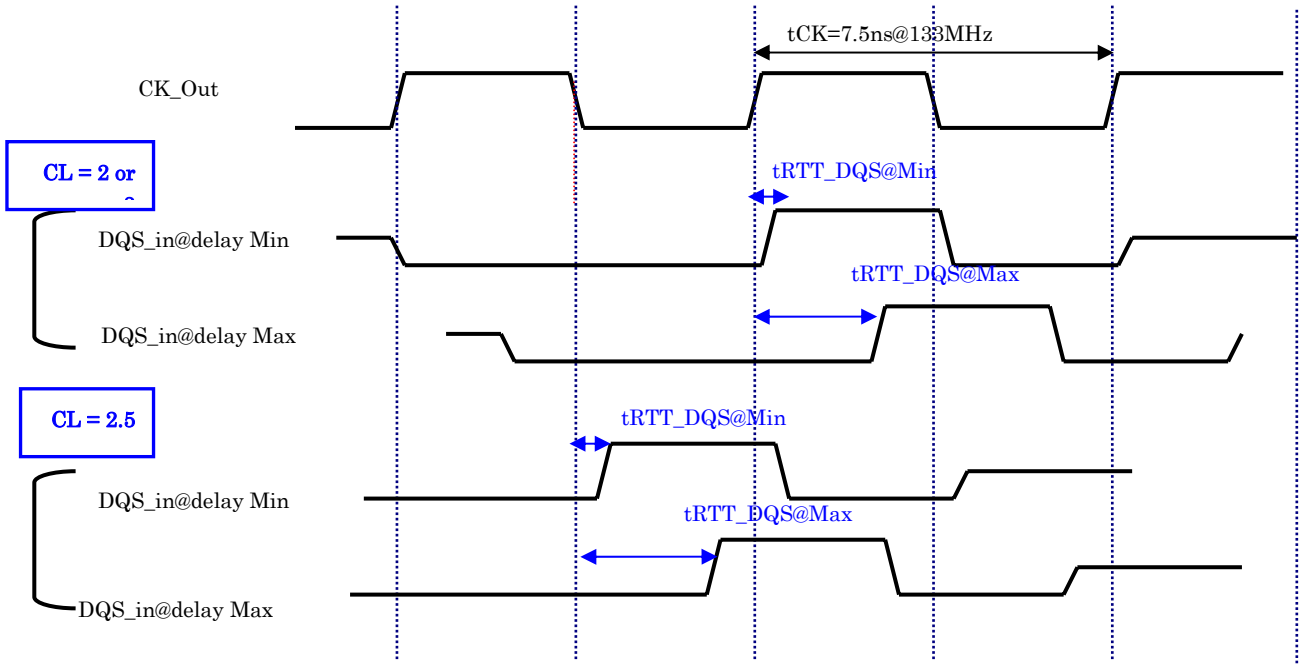
Write Spec (3) : DQ-DQS



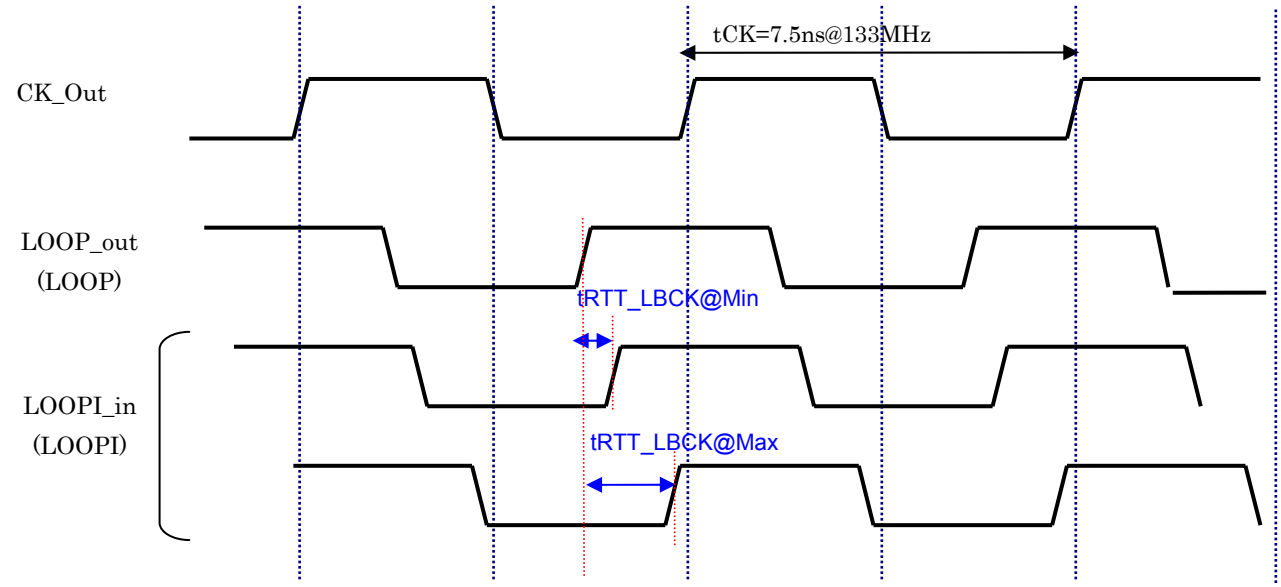
Read Spec (1)： DQ-DQS



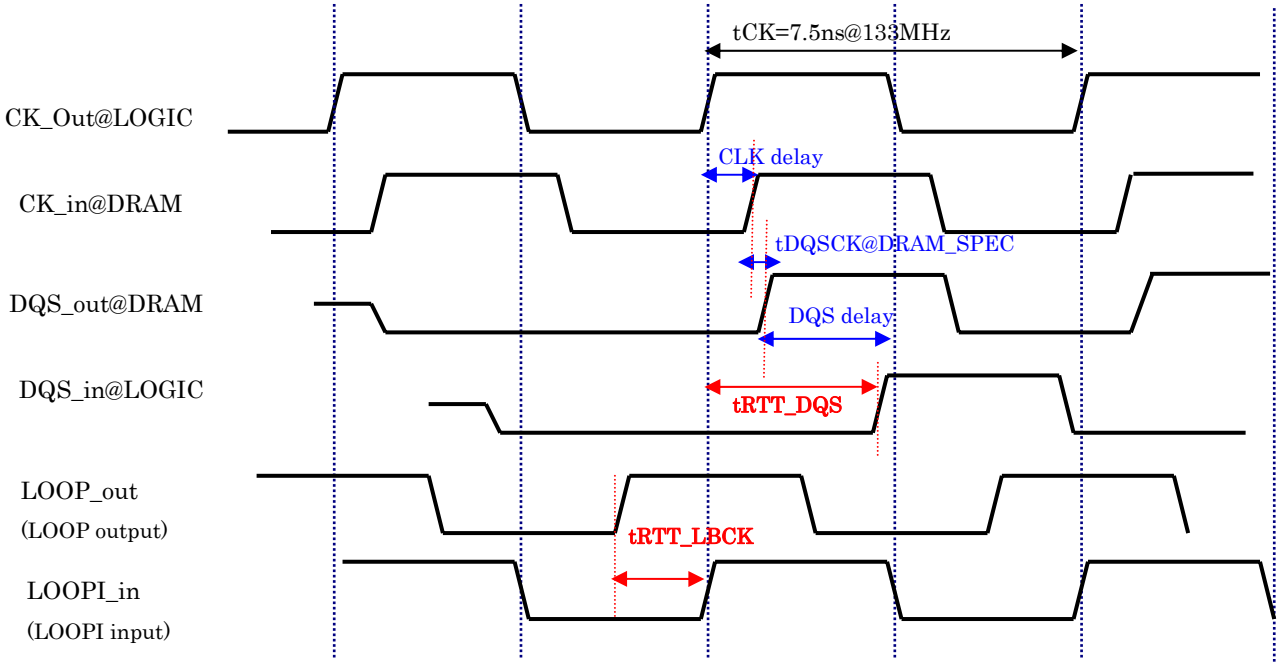
Read Spec (2)： DQS-R.T.T (RoundTrip Time)



Read Spec (3) : LOOP-R.T.T (RoundTrip Time)



Read Spec (4) : Skew DQS-LOOP



9.7.6 I²C Interface

I²C Bus Timing

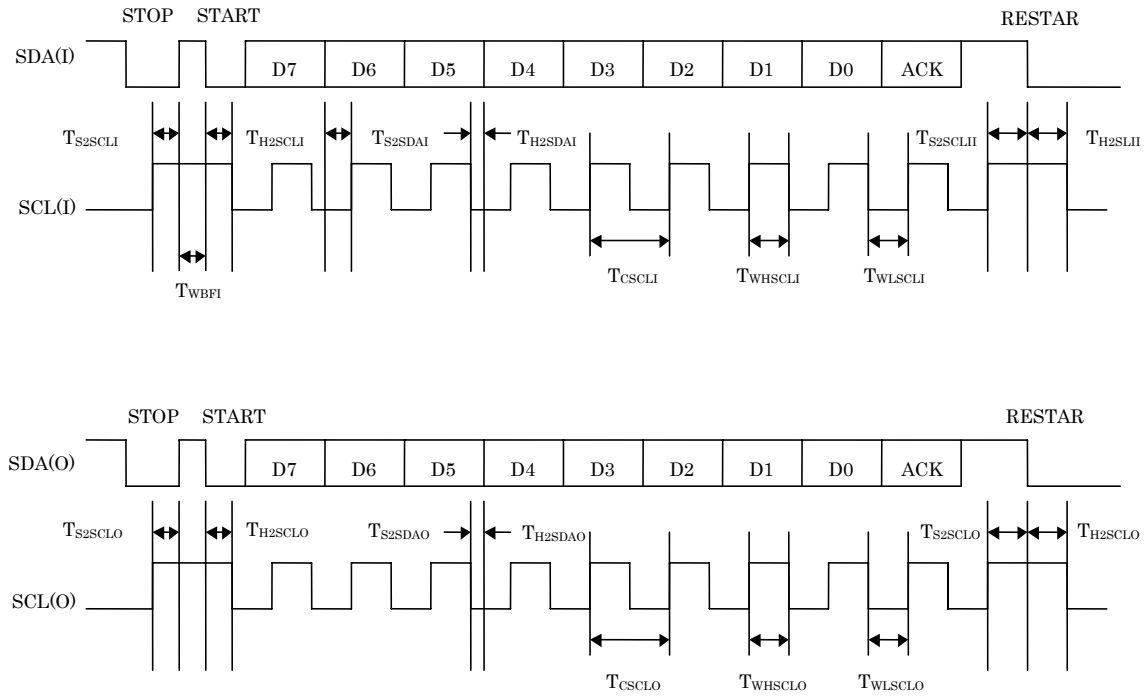
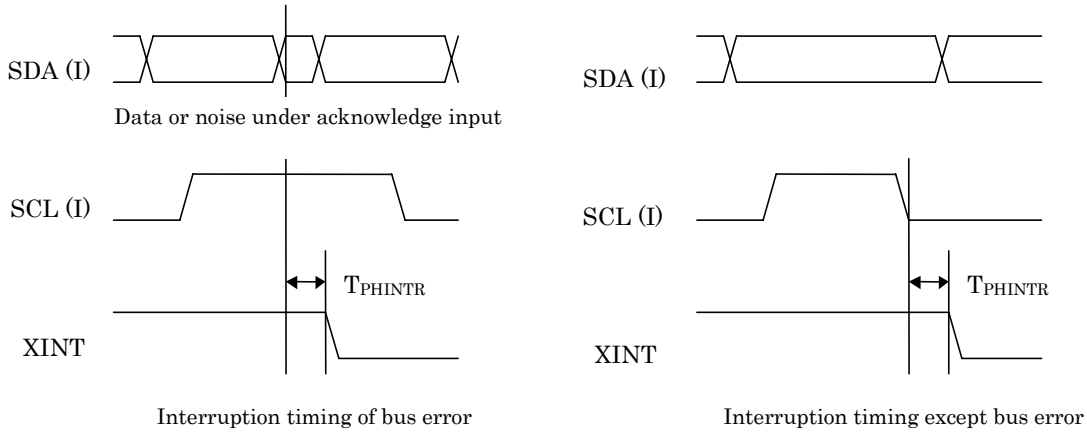


Fig. 9.12 I²C Bus Timing

Interruption Timing




FUJITSU