SH7268/SH7269 Group

Video Display Controller 4 Driver User's Manual

R01AN0778EJ0100
Rev.1.00
Apr 18, 2013

Introduction

This document explains the usage of SH7268/SH7269 Video Display Controller 4 (hereafter referred to as 'VDC4') Driver.

Target Device

SH7268/SH7269

Contents

1. Overview

1.1 Environment

This program was developed and tested in the following environments.

- CPU
  - SH7269
- Development Environment
  - HEW (SuperH RISC engine microcomputer software integrated development environment) Version 4.09.00.007
  - Renesas SuperH RISC engine Standard Toolchain Version 9.4.1.0
    - SH C/C++ Compiler Version 9.04.01
    - SH Assembler Version 7.01.02
    - SH C/C++ Standard Library Generator Version 3.00.03
    - Optimizing Linkage Editor Version 10.01.00
- Evaluation Board
  - SH7269 CPU board (Part number: R0K572690C000BR)
  - Optional board for graphic display (Part number: M3A-HS64G02)
  - SH7269 VDC4 board (Part number: R0K572690B000BR)

1.2 Features

The following table lists the features supported by this driver program.

Table 1 Features of VDC4 Driver

Input Video	Image Specification	• ITU-R BT.656 standard 8bit (27MHz) • ITU-R BT.601 standard 8bit (27MHz, interlace signal) • ITU-R BT.601 standard 8bit (54MHz, progressive signal) • RGB888, RGB666 and RGB565 video image
	Image Quality Adjustment	• Horizontal noise reduction • B lack stretch • Sharpness / LTI
	Picture Recording	• YCC422, RGB565, RGB888 format A rate of 1/1, 1/2, 1/4 or 1/8 field
	Scaling / Rotation	• Scaling input video image to output image Vertical / Horizontal: x1/8 to x8 • 0, 90, 180 or 270 degree rotation and horizontal mirroring
Graphics	Layer	• 3 planes Graphics 1 / Video[1] Graphics 2 Graphics 3
	Image Format	• Prog ressive format RGB565, RGB888, ARGB1555, ARGB4444, ARGB8888 CLUT8, CLUT4, CLUT1, YCC422[2]
	Image Synthesizer	• Alpha blending in rectangular area (fade-in and fade-out functions) • Ch roma-key • Alpha blending in one pixel units
Panel Output	Size	• QVGA, WQVGA, VGA, WVGA, SVGA
	Format	• Progressive RGB565 / RGB666 / RGB888 • Progressive RGB888 (serial output format)
	Adjustment	• Brightness / Contrast / Dither processing • RGB gamma correction

Notes: 1. Graphics 1 and Video are exclusive.
2. Only for graphics 1.

1.3 File Configuration

The file configuration of this driver is shown below.

Table 2 File Configuration

File Name	Description
vdc4_api.c	Source file for VDC4 driver functions.
vdc4_api.h	Header file including the prototype declarations for the VDC4 driver calls and definitions of constants.
vdc4_clk.c	Source file for checking panel clock and software reset.
vdc4_int.c	Source file interrupt handlers.
vdc4_local.h	Header file including local definitions.
vdc4_para.c	Source file checking arguments.
vdc4_reg.c	Source file controlling registers.
vdc4_user.h	Header file for compilation option.

This driver requires external header files as below.

Table 3 External File Dependencies

File Name	Description
typedefine.h	Header file including the typedef declarations for the basic types.
iodefine.h	Header file including IO definitions.

1.4 Program Size and Section

Table 4 shows program size and section used by the VDC4 driver.

Table 4 Program Size and Section

"Renesas SuperH RISC engine Standard Toolchain 9.4.1.0"
"Speed & size optimization enabled"

Type	Section name	Size [byte]	Description
ROM	P_VDC4	10.3K (14.4K)	Program area
	C_VDC4	0.9K (1.0K)	Constant area
	D_VDC4	40 (40)	Initialized data area
RAM	B_VDC4	204 (204)	Uninitialized data area

Note: Program size does not include VRAM size and Input Video Buffer size.
Values in the parentheses are program size when parameter checking is defined.

1.5 Layer and Surface

VDC4 has three layers (graphics plane). This driver uses these layers as graphics resource and creates three graphics surfaces and one video surface. Graphics surface 1 and Video surface are mutually exclusive, because both surfaces use layer 1 as resource. Thus, those surfaces can not be created simultaneously.

The features of VDC4 module are implemented in this driver by controlling the surfaces.

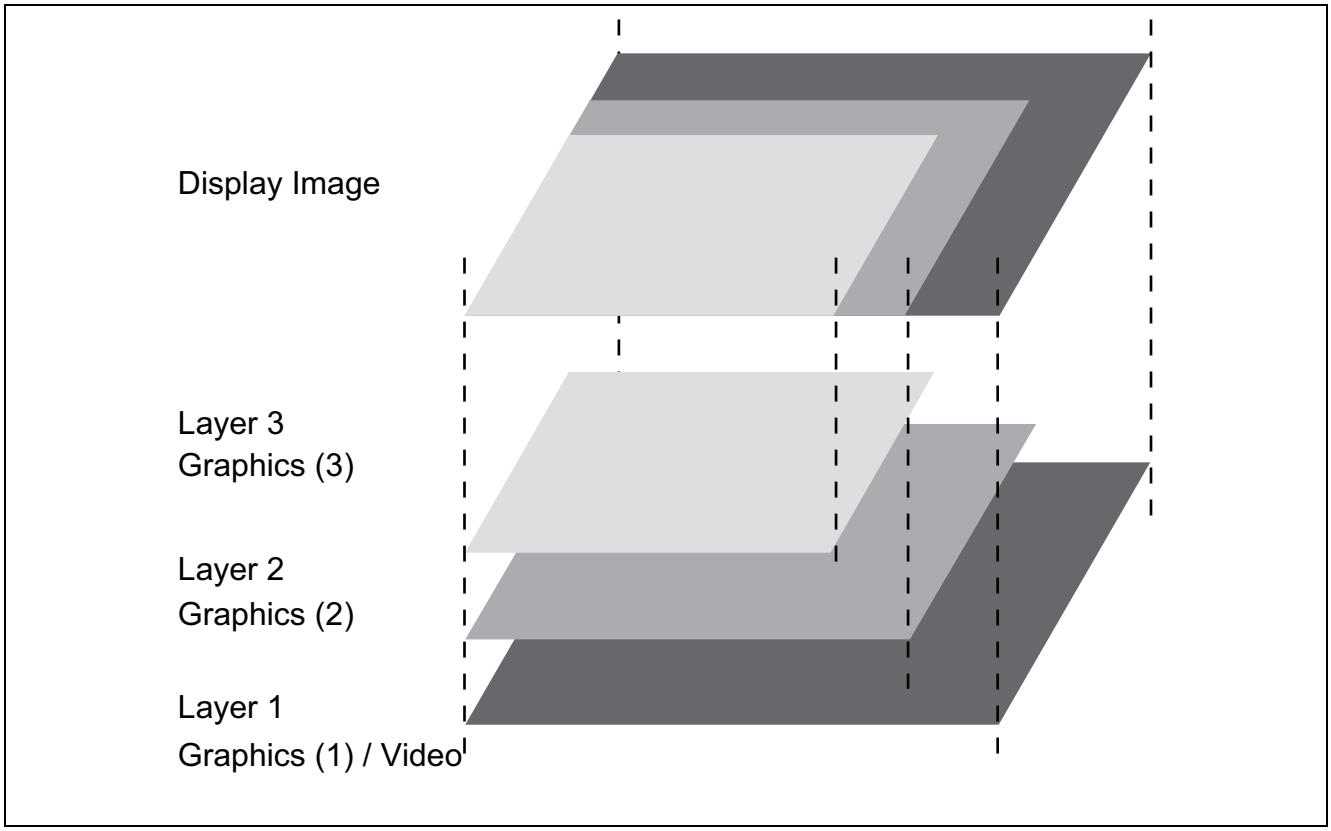
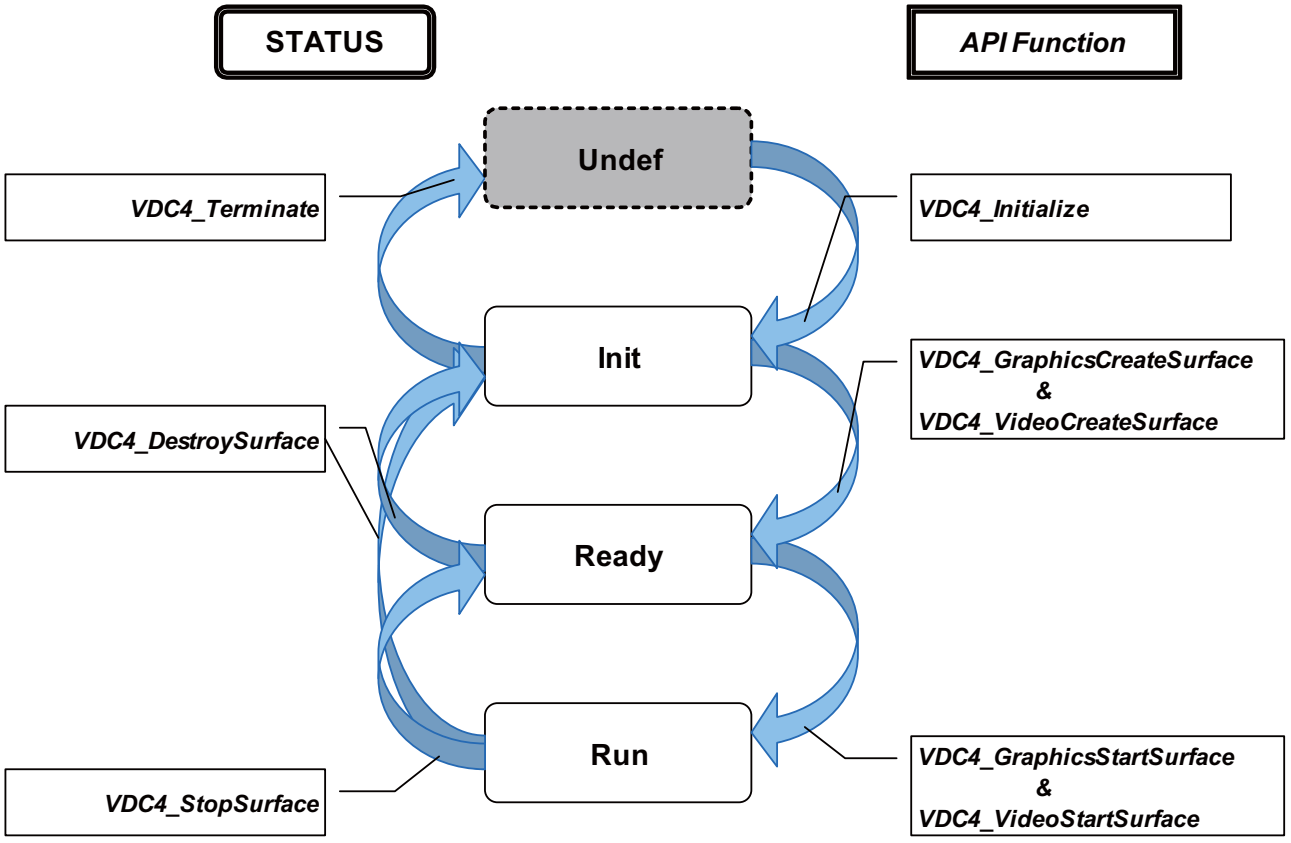


Figure 1 Layer and Surface

1.6 State Transitions

All surfaces have a state. The state transitions occur when a particular API function is used. The state transition of the surface is shown below.



Note: The state of the surface is independent of each other with the exception that the states of all surfaces are changed by VDC4_Initialize or VDC4_Terminate.

Figure 2 VDC4 Driver State Transitions

1.7 Interrupt Handler

Interrupt handlers are implemented in this driver. Table 5 shows the list of interrupt handlers.

Table 5 Interrupt Handler

Interrupt Source	Interrupt Vector		Interrupt Handler
	Number	Address	
VI_VSYNC Vsync signal before scaling	171	0x00000002AC ~ 0x00000002AF	void vdc4_Int_VI_VSYNC(void) ;
LO_VSYNC Vsync signal after scaling	172	0x00000002B0 ~ 0x00000002B3	void vdc4_Int_LO_VSYNC(void) ;
VSYNCERR Missing Vsync signal for scaling	173	0x00000002B4 ~ 0x00000002B7	void vdc4_Int_VSYNCERR(void) ;
VLINE Specified line signal for panel output	174	0x00000002B8 ~ 0x00000002BB	void vdc4_Int_VLINE(void) ;
VFIELD Field end signal for recording function	175	0x00000002BC ~ 0x00000002BF	void vdc4_Int_VFIELD(void) ;
VBUFERR1 Frame buffer write overflow signal	176	0x00000002C0 ~ 0x00000002C3	void vdc4_Int_VBUFERR1(void) ;
VBUFERR2 Graphics 1 frame buffer read underflow signal	177	0x00000002C4 ~ 0x00000002C7	void vdc4_Int_VBUFERR2(void) ;
VBUFERR3 Graphics 2 frame buffer read underflow signal	178	0x00000002C8 ~ 0x00000002CB	void vdc4_Int_VBUFERR3(void) ;
VBUFERR4 Graphics 3 frame buffer read underflow signal	179	0x00000002CC ~ 0x00000002CF	void vdc4_Int_VBUFERR4(void) ;

If it is necessary to use VDC4 interrupts, functions in Table 5 should be defined as interrupt handler. If API function provided by OS or equal with it to define the interrupt handler is not used, interrupt handler in Table 5 should be set to appropriate entry in the vector table (see example below).

```
1  /* Vector table example */
2  /* Register the interrupt handler in the vector table */
3  void (*VectorTable[])( void ) = {
4     …
5     vdc4_Int_VI_VSYNC,
6     vdc4_Int_LO_VSYNC,
7     vdc4_Int_VSYNCERR,
8     vdc4_Int_VLINE,
9     vdc4_Int_VFIELD,
10    vdc4_Int_VBUFERR1,
11    vdc4_Int_VBUFERR2,
12    vdc4_Int_VBUFERR3,
13    vdc4_Int_VBUFERR4,
14    …
15 } ;
```

1.8 Compilation Switches

This driver has compilation switches in vdc4_user.h.

1.8.1 Parameter Checking

When _VDC4_PARAMETER_CHECK is defined, the driver checks the arguments of driver functions. Arguments are checked and error codes returned if there are errors. For error codes, see 2.2 Error and 2.3 API Function.

1.8.2 Interrupt Handler Definition

In this driver, some functions for interrupt handlers are implemented (see 1.7 Interrupt Handler and Table 5). To use these functions for interrupt function that will guarantee register values before and after processing, _VDC4_DEFINE_INTHDL should be defined. When _VDC4_DEFINE_INTHDL is defined, the declaration "#pragma interrupt" is enabled and the compiler assumes that the functions in Table 5 are interrupt functions.

If some function that support interrupt function (e.g. the API function provided by OS to register the interrupt handler) is used for the registration of the interrupt handler, _VDC4_DEFINE_INTHDL should be undefined.

1.9 Restriction

1.9.1 Reserved Word

To separate from the other program, the prefix 'VDC4' is appended to the API names, variable names and other symbols in this driver. Therefore, the names started with 'VDC4' (in both uppercase and lowercase letters) should not be used in the other program.

1.9.2 Register Update

A lot of VDC4 registers are updated at the rising edge of the Vsync signal. Therefore, it will take a period of time equal to 1 Vsync period time at the most to reflect the set values.

1.9.3 Reentrancy

API functions in this driver are not reentrant. If API functions are called asynchronously from multiple tasks or interrupt service routines, it can cause unexpected behavior. Pay attention to where and when API functions are called.

2. API

2.1 Common Definition

2.1.1 Typedef

In this driver, data types in Table 6 are used. These declarations are defined in typedefine.h (see 1.3File Configuration).

Table 6 Typedef Declarations

Typedef T	ype
_SBYTE	signed char
_UBYTE	unsigned char
_SWORD	signed short
_UWORD	unsigned short
_SINT	signed int
_UINT	unsigned int
_SDWORD	signed long
_UDWORD	unsigned long
_SQWORD	signed long long
_UQWORD	unsigned long long

2.1.2 Definition of Enumeration Constants

The `vdc4_OnOff` enumeration defines ON and OFF.

```
typedef enum {
    VDC4_OFF = 0,
    VDC4_ON = 1
} vdc4_OnOff ;
```

Enum Value		Description
VDC4_OFF	0	OFF
VDC4_ON	1	ON

The `vdc4_Edge` enumeration defines the edge of the signal.

```
typedef enum {
    VDC4_EDGE_RISING = 0,
    VDC4_EDGE_FALLING = 1
} vdc4_Edge ;
```

Enum Value		Description
VDC4_EDGE_RISING	0	rising edge of the signal
VDC4_EDGE_FALLING	1	falling edge of the signal

The `vdc4_LayerID` enumeration defines the surface ID.

```
typedef enum {
    VDC4_SURFACE_GRAPHICS_1 = 0,
    VDC4_SURFACE_GRAPHICS_2 = 1,
    VDC4_SURFACE_GRAPHICS_3 = 2,
    VDC4_SURFACE_GRAPHICS_NUM = 3,
    VDC4_SURFACE_VIDEO_1 = 3,
```

```
    VDC4_SURFACE_VIDEO_NUM = 1,
    VDC4_SURFACE_NUM = 4
} vdc4_LayerID ;
```

Enum Value		Description
VDC4_SURFACE_GRAPHICS_1	0	Graphics 1 surface ID
VDC4_SURFACE_GRAPHICS_2	1	Graphics 2 surface ID
VDC4_SURFACE_GRAPHICS_3	2	Graphics 3 surface ID
VDC4_SURFACE_GRAPHICS_NUM	3	Number of Graphics surfaces
VDC4_SURFACE_VIDEO_1	3	Video surface ID
VDC4_SURFACE_VIDEO_NUM	1	Number of Video surfaces
VDC4_SURFACE_NUM	4	Number of surfaces

2.1.3 Rectangle Structure

This driver has some timing data expressed by a rectangle, so structure `vdc4_AreaRect` that express rectangle is defined as shown in Figure 3.

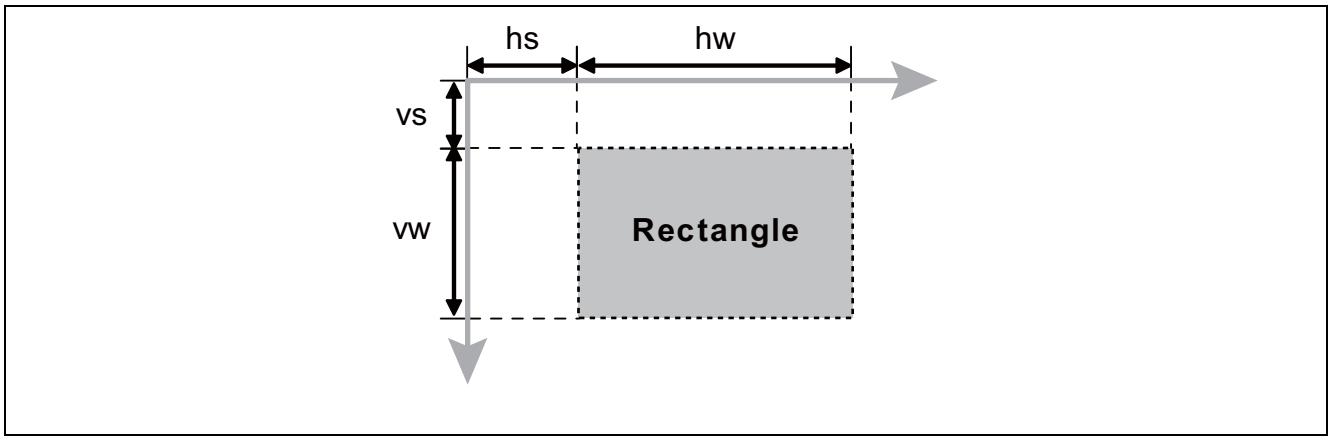


Figure 3 Rectangle Data

Members of the structure `vdc4_AreaRect` are shown below.

```
typedef struct {
    _UWORD hs ;
    _UWORD hw ;
    _UWORD vs ;
    _UWORD vw ;
} vdc4_AreaRect ;
```

Type Member Name	Description
_UWORD hs	Horizontal start position [pixel clock cycles]
_UWORD hw	Horizontal width [pixel clock cycles]
_UWORD vs	Vertical start position [lines]
_UWORD vw	Height (vertical width) [lines]

2.2 Error

API function calls returns an error code. The error codes are shown in Table 7.

Table 7 Error Code List

Error Value		Description
VDC4_ERR_NONE	0	Normal end.
VDC4_ERR_SURFACE_BAD	0x2000	Surface error. The function cannot be called with the surface.
VDC4_ERR_SURFACE_STATUS	0x2001	Surface error. The function cannot be called in a current state.
VDC4_ERR_PARAM_RANGE	0x4001	Parameter error. The specified value is out of range.
VDC4_ERR_PARAM_UNDEF	0x4002	Parameter error. Null should not be specified.
VDC4_ERR_PARAM_INVALID	0x4003	Parameter error. The specified parameter is invalid.
VDC4_ERR_PARAM_OTHERS	0x4000	Parameter error. Others.
VDC4_ERR_SYSTEM_PNLCLK	0x8000	System error. The abnormality of the panel clock is detected.

Parameter errors are returned from API function call only when the compilation switch of parameter checking is defined.

2.3 API Function

Table 8 shows the states in which each driver call can be made.

Table 8 State Matrix

Driver Call Name	Surface	States			
		Undef Init		Ready	Run
VDC4_Initialize All	surface	OK	NG	NG	NG
VDC4_Terminate All	surface	NG	OK	NG	NG
VDC4_GraphicsCreateSurface	Video	NG	NG	NG NG	
	Graphics1/2/3		OK		
VDC4_VideoCreateSurface	Video	NG	OK	NG NG	
	Graphics 1/2/3		NG		
VDC4_DestroySurface All	surface	NG	NG	OK	OK
VDC4_RegistCallbackFunc All	surface	NG	OK		
VDC4_GraphicsStartSurface	Video	NG NG		NG	NG
	Graphics 1/2/3			OK	
VDC4_VideoStartSurface	Video	NG NG		OK	NG
	Graphics 1/2/3			NG	
VDC4_GraphicsChangeParam	Video	NG NG		NG	NG
	Graphics 1/2/3				OK
VDC4_VideoChangeParam	Video	NG NG		NG	OK
	Graphics 1/2/3				NG
VDC4_StopSurface All	surface	NG	NG	NG	OK
VDC4_ImageColorMatrix	Video	NG NG		OK	NG
	Graphics 1			OK	
	Graphics 2/3			NG	
VDC4_VideoNoiseReduction	Video	NG NG		OK	NG
	Graphics 1/2/3			NG	
VDC4_ImageEnhancement	Video OK	NG NG			OK
	Graphics 1			OK	OK
	Graphics 2/3			NG NG	
VDC4_GraphicsAlphaBlending	Video NG	NG NG			NG
	Graphics 1			NG	NG
	Graphics 2/3			OK OK	
VDC4_GraphicsChromakey	Video NG	NG NG			NG
	Graphics 1			NG	NG
	Graphics 2/3			OK OK	
VDC4_GraphicsSetCLUT	Video NG	NG NG			NG
	Graphics 1/2/3			OK OK	
VDC4_DisplayCalibration All	surface	NG	OK		
VDC4_DisplaySetGammaCorrectionTable All	surface	NG	OK		
VDC4_SwitchVsync All	surface	NG	OK		
VDC4_CheckClock All	surface	OK	NG		
VDC4_Reset All	surface	OK	NG		

2.3.1 VDC4_Initialize

Syntax

```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_Initialize( const vdc4_InitAttr *attr,
                                void (*init_func)( _UDWORD ),
                                _UDWORD user_num ) ;
```

Parameters	• [in]const vdc4_InitAttr *attr	Initialization attribute parameter
	• [in]void (*init_func)(_UDWORD)	Pointer to the user-defined function
	• [in]_UDWO RD user_num	User-defined number
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_UNDEF	Parameter error. Undefined.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.
	VDC4_ERR_PARAM_OTHERS	Parameter error. Others.

Description

In this function, the operations are performed as below.

- The specified user-defined function is executed.
- The VDC4 driver internal variables are initialized.
- The VDC4 registers are initialized.
- The initialization for LCD panel drive is performed.
- The operation of the panel clock is enabled.
- The VDC4 interrupts are enabled.
- All states of surfaces are transitioned to 'Init'.

Before VDC4 driver initialization, user-defined function specified by `init_func` is called. For information about the process of user-defined function, see 3.1.

It is not always necessary to specify the user-defined function pointer. If user-defined function is not specified, the operation stated below should be performed before this VDC4 driver initialization.

- Supply the clock to the VDC4.
- Set the VDC4 interrupt levels.
- Set I/O port for the VDC4.
- Set the environment specific parameters (e.g. related to input video image or LCD panel).

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_InitAttr * attr	in	Initialization attribute parameter NULL should not be specified.
void (*init_func)(_UDWORD)	in	Pointer to the user-defined function If '0' is not specified, user-defined function will be called with an argument specified by user-defined number (user_num). If necessary, user-defined function must be implemented by the user. Syntax: void Init_Func(_UDWORD User_Num) ; Parameters: • [in]_UDWORD User_Num — User-defined number Return Values: • void Description: Processing implemented by the user is performed.
_UDWORD user_num	in	User-defined number This parameter is used as an argument to the user-defined function. This parameter is ignored, when pointer to the user-defined function (init_func) is set to '0'.

Members of the structure `vdc4_InitAttr` are shown below.

```
typedef struct {
    vdc4_VsyncSigSel Vsel ;
    vdc4_SyncCtrl *sync ;
    vdc4_PanelClockSel panel_icksel ;
    vdc4_PanelClkDCDR panel_dcdr ;
    _UWORD tcon_half ;
    _UWORD tcon_offset ;
    vdc4_Edge outcnt_lcd_edge ;
    vdc4_LcdTconTim *outctrl[ VDC4_LCD_TCONSIG_NUM ] ;
    vdc4_LcdOutput *settings ;
    _UDWORD IntEnable_1 ;
    _UDWORD IntEnable_2 ;
} vdc4_InitAttr ;
```

Type Member Name	Input / Output	Description
vdc4_VsyncSigSel Vsel	in	Vsync Signal Output Select
vdc4_SyncCtrl * sync	in	Synchronization Control If NULL is specified, the frequent Vsync signal masking and the missing Vsync signal compensation will be set to turn off.
vdc4_PanelClockSel panel_icksel	in	Panel clock source select • VDC4_LCDPANEL_CLKSEL_IMG:

		Video image clock (VIDEO_X1) • VDC4_LCDPANEL_CLKS EL_EXT: External clock (LCDEXT_CLK) • VDC4_LCDPANEL _CLKSEL_PERI: Peripheral clock 1 • VDC4_LCDPANEL_C LKSEL_IMG_DV: Video image clock (DV_CLK)
vdc4_PanelClkDCDR panel_dcdr	in	Panel clock frequency division ratio • VDC4_L CDPANEL_CLKDIV_1_1: 1/1 • VDC4_L CDPANEL_CLKDIV_1_2: 1/2 • VDC4_L CDPANEL_CLKDIV_1_3: 1/3 • VDC4_L CDPANEL_CLKDIV_1_4: 1/4 • VDC4_L CDPANEL_CLKDIV_1_5: 1/5 • VDC4_L CDPANEL_CLKDIV_1_6: 1/6 • VDC4_L CDPANEL_CLKDIV_1_7: 1/7 • VDC4_L CDPANEL_CLKDIV_1_8: 1/8 • VDC4_L CDPANEL_CLKDIV_1_9: 1/9 • VDC4_L CDPANEL_CLKDIV_1_12: 1/12 • VDC4_L CDPANEL_CLKDIV_1_16: 1/16 • VDC4_L CDPANEL_CLKDIV_1_24: 1/24 • VDC4_LCDPANEL_CLKDIV_1_32: 1/32
_UWORD tcon_half	in	TCON reference timing, 1/2fH timing The clock cycle count from the Hsync rising edge. 0x0000 ~ 0x07FF
_UWORD tcon_offset	in	TCON reference timing, offset Hsync signal timing The clock cycle count from the Hsync rising edge. 0x0000 ~ 0x07FF
vdc4_Edge outcnt_lcd_edge	in	Output phase control of LCD_DATA23 to LCD_DATA0 pin • VDC4_EDGE_RISING: Rising edge • VDC4_EDGE_FALLING: Falling edge
vdc4_LcdTconTim * outctrl[VDC4_LCD_TCONSIG_NUM]	in	LCD TCON timing setting • outc trl[VDC4_LCD_TCONSIG_STVA_VS]: STVA / VS • outc trl[VDC4_LCD_TCONSIG_STVB_VE]: STVB / VE • outctrl[VDC4 _LCD_TCONSIG_STH_SP_HS]: STH / SP / HP • outctrl[VDC4 _LCD_TCONSIG_STB_LP_HE]: STB / LP / HE • outctrl[VDC4 _LCD_TCONSIG_CPV_GCK]: CPV / GCK • outctrl[VDC4 _LCD_TCONSIG_POLA]: POLA • outctrl[VDC4 _LCD_TCONSIG_POLB]: POLB • outctrl[VDC4 _LCD_TCONSIG_DE]: DE The settings of unnecessary signal should be set to NULL.
vdc4_LcdOutput * settings	in LCD	Output Control NULL should not be specified.
_UDWORD IntEnable_1	in	Interrupt enable bits 1, INT 0 - INT 7 Specify the interrupt to be used. To use more than one type of interrupt, the OR should be taken to specify the value.

		• VDC4_INT _BIT_NONE: None • VDC4_INT_B IT_VI_VSYNC: VI_VSYNC • VDC4_INT_B IT_LO_VSYNC: LO_VSYNC • VDC4_INT_B IT_VSYNCERR: VSYNCERR • VDC4_INT _BIT_VLINE: VLINE • VDC4_INT _BIT_VFIELD: VFIELD • VDC4_INT _BIT_VBUFERR1: VBUFFER1 • VDC4_INT _BIT_VBUFERR2: VBUFFER2 • VDC4_INT _BIT_VBUFERR3: VBUFFER3
_UDWORD IntEnable_2	in	Interrupt enable bits 2, INT 8 If necessary, specify VBUFFER4 interrupt. • VDC4_INT _BIT_NONE: None • VDC4_INT_BIT_VBUFERR4: VBUFFER4

Members of the structure `vdc4_VsyncSigSel` are shown below.

```
typedef struct {
    vdc4_OnOff FreeRunVsync ;
    _UWORD res_fv ;
    _UWORD res_fh ;
} vdc4_VsyncSigSel ;
```

Type Member Name	Input/ Output	Description
vdc4_OnOff FreeRunVsync	in	Free-running Vsync ON/OFF • VDC4_OF F • VDC4_O N
_UWORD res_fv	in	Free-running Vsync period setting Vsync period = (res_fv + 1) x Hsync period 0x0000 ~ 0x07FF [lines] This parameter is ignored, when Free-running Vsync (FreeRunVsync) is set to VDC4_OFF.
_UWORD res_fh	in	Hsync period setting Hsync period = (res_fh + 1) / pixel clock frequency 0x0000 ~ 0x07FF [pixel clock cycles]

Members of the structure `vdc4_SyncCtrl` are shown below.

```
typedef struct {
    vdc4_OnOff res_vmask_on ;
    _UWORD res_vmask ;
    vdc4_OnOff res_vlack_on ;
    _UWORD res_vlack ;
} vdc4_SyncCtrl ;
```

Type Member Name	Input/ Output	Description
vdc4_OnOff res_vmask_on	in	Repeated Vsync signal masking control • VDC4_OF F • VDC4_O N
_UWORD res_vmask	in	Repeated Vsync signal masking period res_vmask should be specified in terms of 128 pixel clock cycles.

vdc4_OnOff res_vlack_on	in	Missing Vsync signal compensation • VDC4_OF F • VDC4_O N
_UWORD res_vlack	in	Missing-Sync Compensating pulse output wait time res_vlack sh ould be spe cified in terms of 128 pixel clock cycles.

Members of the structure `vdc4_LcdTconTim` are shown below.

```
typedef struct {
    _UWORD tcon_hsvs ;
    _UWORD tcon_hwvw ;
    vdc4_LcdTconPolMode tcon_md ;
    _UWORD tcon_hs_sel ;
    _UWORD tcon_inv ;
    vdc4_LcdTconPin tcon_pin ;
    vdc4_Edge outcnt_edge ;
} vdc4_LcdTconTim ;
```

Type Member Name	Input/ Output	Description
_UWORD tcon_hsvs	in	Signal pulse start position First changing timing. Starts pul se output after the time specified by the value of tcon_hsvs. 0x0000 ~ 0x07FF [pixel clock cycles or 1/2fH cycles] This parameter sh ould be greater than or equ al to 1 when tcon_ md of POLA/POLB sign al is not set to normal mode.
_UWORD tcon_hwvw	in Pulse	width Second cha nging timin g. Outputs a pulse of the duration of the value of tcon_hwvw. 0x0000 ~ 0x07FF [pixel clock cycles or 1/2fH cycles]
vdc4_LcdTconPolMode tcon_md	in	POLA/POLB signal generation mode select • VDC 4_LCD_TCON_POLMD_NORMAL: Normal mode • VDC4_L CD_TCON_POLMD_1X1REV: 1x1 reverse mode • VDC4_L CD_TCON_POLMD_1X2REV: 1x2 reverse mode • VDC4_L CD_TCON_POLMD_2X2REV: 2x2 reverse mode
_UWORD tcon_hs_sel	in	Horizontal signal operating reference select • 0: Hsync signal reference • 1: Offset Hsync signal reference Offset value is spe cified b y the tcon_offset membe r of the vdc4_InitAttr structure.
_UWORD tcon_inv	in	Polarity inversion control of signal • 0: Not inverted • 1: Inverted
vdc4_LcdTconPin tcon_pin	in	Output pin for LCD driving signal • VDC 4_LCD_TCON_PIN_NON: None • VDC4_L CD_TCON_PIN_0: LCD_TCON0 • VDC4_L CD_TCON_PIN_1: LCD_TCON1 • VDC4_L CD_TCON_PIN_2: LCD_TCON2

		• VDC4_L CD_TCON_PIN_3: LCD_TCON3 • VDC4_L CD_TCON_PIN_4: LCD_TCON4 • VDC4_L CD_TCON_PIN_5: LCD_TCON5 • VDC4_L CD_TCON_PIN_6: LCD_TCON6
vdc4_Edge outcnt_edge	in	Output phase control of signal • VDC4_EDGE_RISING: Rising edge • VDC4_EDGE_FALLING: Falling edge This parameter is not referenced when the output pin (tcon_pin) is set to VDC4_LCD_TCON_PIN_NON.

Note: Specify tcon_hsvs and tcon_hwvw in pixel clock cycles when the signal is used as horizontal signal.
Specify tcon_hsvs and tcon_hwvw in 1/2fH cycles when the signal is used as vertical signal.

It is not necessary to set all member of the structure `vdc4_LcdTconTim`. Members that must be set are different depending on the each signal.

Table 9 Structure vdc4_LcdTconTim and Timing Signal

Signal / Member	STVA	STVB	STH	STB	CPV	POLA	POLB	DE
tcon_hsvs	R	R	R	R	R	R	R	I
tcon_hwvw	R	R	R	R	R	R	R	I
tcon_md	I	I	I	I	I	R	R	I
tcon_hs_sel	I	I	R	R	R	R	R	I
tcon_inv	R	R	R	R	R	R	R	R
tcon_pin	R	R	R	R	R	R	R	R
outcnt_edge	R	R	R	R	R	R	R	R

Note: 'R' in the table means that the member is referred.
'I' in the table means that the member is ignored.

Members of the structure `vdc4_LcdOutput` are shown below.

```
typedef struct {
    vdc4_AreaRect res_f ;
    vdc4_OnOff out_endian_on ;
    vdc4_OnOff out_swap_on ;
    vdc4_LcdOutFormat out_format ;
    vdc4_LcdClkFrqSel out_frq_sel ;
    _UWORD out_dir_sel ;
    vdc4_LcdClkPhase out_phase ;
} vdc4_LcdOutput ;
```

Type Member Name	Input/ Output	Description
vdc4_AreaRect res_f	in	Full-screen enable control See section 2.1.3 for the structure vdc4_AreaRect. res_f.hs should be 16 clock cycles or greater. res_f.hs + res_f.hw should be 2015 clock cycles or less. res_f.vs should be 4 lines or greater. res_f.vs + res_f.vw should be 2039 lines or less.
vdc4_OnOff out_endian_on	in	Bit endian change ON/OFF control • VDC4_OFF • VDC4_ON
vdc4_OnOff	in	B/R signal swap ON/OFF control

out_swap_on		• VDC4_OF F • VDC4_O N
vdc4_LcdOutFormat out_format	in	LCD output format select • VDC4_L CD_OUTFORMAT_RGB888: RGB888 • VDC4_L CD_OUTFORMAT_RGB666: RGB666 • VDC4_L CD_OUTFORMAT_RGB565: RGB565 • VDC4_LCD_OUTFORMAT_SERIAL_RGB: Serial RGB
vdc4_LcdClkFrqSel out_frq_sel	in	Clock frequency control • VDC4_LCD_SERIAL_CLKFRQ_3: x 3, serial RGB • VDC4_LCD_SERIAL_CLKFRQ_4: x 4, serial RGB This parameter is referenced only when LCD output format select (out_format) is set to serial RGB.
_UWORD out_dir_sel	in	Scan direction select • 0: Forward scan • 1: Reverse scan This parameter is referenced only when LCD output format select (out_format) is set to serial RGB.
vdc4_LcdClkPhase out_phase	in	Clock Phase Adjustment for Serial RGB Output Triple speed mode (x3) • VDC4_L CD_SERIAL_CLKPHASE_0: 0[clk] • VDC4_L CD_SERIAL_CLKPHASE_1: 1[clk] • VDC4_L CD_SERIAL_CLKPHASE_2: 2[clk] Quadruple speed mode (x4) • VDC4_L CD_SERIAL_CLKPHASE_0: 0[clk] • VDC4_L CD_SERIAL_CLKPHASE_1: 1[clk] • VDC4_L CD_SERIAL_CLKPHASE_2: 2[clk] • VDC4_L CD_SERIAL_CLKPHASE_3: 3[clk] This parameter is referenced only when LCD output format select (out_format) is set to serial RGB.

2.3.2 VDC4_Terminate

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_Terminate(void (*quit_func)(_UDWORD), _UDWORD user_num) ;	
Parameters	• [in]void (*quit_func)(_UDWORD)	Pointer to the user-defined function
	• [in]_UDWO RD user_num	User-defined number
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.

Description

In this function, the operations are performed as below.

- The specified user-defined function is executed.
- The operation of the panel clock is disabled.
- The VDC4 interrupts are disabled.
- All states of surfaces are transitioned to 'Undef'.

After VDC4 driver termination, user-defined function specified by `quit_func` is called. For information about the process of user-defined function, see 3.2.

It is not always necessary to specify the user-defined function pointer. If user-defined function is not specified, the operation stated below should be performed after this VDC4 driver termination.

- Halt the clock supply to the VDC4.
- Clear the VDC4 interrupt levels.
- Set the environment specific parameters (e.g. related to input video image or LCD panel).

Arguments Settings

Type Parameter Name	Input / Output	Description
void (*quit_func)(_UDWORD)	in	Pointer to the user-defined function If '0' is not specified, user-defined function will be called with an argument specified by user-defined number (user_num). If necessary, user-defined function must be implemented by the user. Syntax void Quit_Func(_UDWORD User_Num) ; Parameters • [in]_UDWORD User_Num User-defined number Return Values • void Description Processing implemented by the user is performed.
_UDWORD user_num	in	User-defined number This parameter is used as an argument to the user-defined function. This parameter is ignored, when pointer to the user-defined function (quit_func) is set to '0'.

2.3.3 VDC4_GraphicsCreateSurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsCreateSruface(vdc4_LayerID id, const vdc4_GraphicsAttr *attr) ;	
Parameters	• [in]vdc4_Lye rID id	Layer ID
	• [in]const vdc4_GraphicsAttr *attr	Graphics surface attribute parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_UNDEF	Parameter error. Undefined.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- The graphics surface specified by id is created.
- The color format of the graphics surface is set.
- The frame buffer of the graphics surface is set.
- The state of the graphics surface specified by id is transitioned from 'Init' to 'Ready'.
- Scaling up parameters are set only when graphics surface 1 is created.

If CLUT1, CLUT4 or CLUT8 color format is specified, it is necessary to set CLUT. If ARGB1555 color format is specified, the alpha value should be specified. To set CLUT and the alpha value for ARGB1555 format are performed by VDC4_GraphicsSetCLUT (see 2.3.17).

Note that graphics surface 1 and video surface are exclusive.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_GraphicsAttr * attr	in	Graphics surface attribute parameter NULL should not be specified.

Members of the structure `vdc4_GraphicsAttr` are shown below.

```
typedef struct {
    vdc4_AreaRect gr_grc ;
    vdc4_FrameBuff buff ;
} vdc4_GraphicsAttr ;
```

Type Member Name	Input/ Output	Description
vdc4_AreaRect gr_grc	in	Display area See section 2.1.3 for the structure vdc4_AreaRect. gr_grc.hs should be 16 clocks or greater. gr_grc.hs + gr_grc.hw should be 2015 clocks or less. gr_grc.hw should be 3 clocks or greater. gr_grc.vs should be 4 lines or greater. gr_grc.vs + gr_grc.vw should be 2039 lines or less.
vdc4_FrameBuff buff	in	Graphics frame buffer parameter

Members of the structure `vdc4_FrameBuff` are shown below.

```
typedef struct {
    _SINT gr_bst_md ;
    void *gr_base ;
    void *buff_2nd ;
    _UWORD gr_ln_off ;
    _SINT gr_ln_off_dir ;
    _UWORD gr_flm_loop ;
    vdc4_GrFormat gr_format ;
    vdc4_OnOff gr_endian_on ;
    _UDWORD bg_color ;
    vdc4_GraphicsExt *GraEx ;
} vdc4_FrameBuff ;
```

Type Member Name	Input/ Output	Description
_SINT gr_bst_md	in	Frame buffer burst transfer mode • 0: 32-byte transfer • 1: 128-byte transfer
void * gr_base	in	Frame buffer base address NULL must not be specified.

void * buff_2nd	in	2nd frame buffer base address (only double buffering mode) When single buffer mode, NULL must be specified. For more information about memory allocation for the second buffer, see Figure 4.
_UWORD gr_ln_off	in	Frame buffer line offset address [bytes] When gr_bst_md is set to 0: A multiple of 32 When gr_bst_md is set to 1: A multiple of 128
_SINT gr_ln_off_dir	in	Line offset address direction of the frame buffer • 0: Increments the address by the line offset address. • 1: Decrements the address by the line offset address. If gr_ln_off_dir is set to '1', the displayed image will be flipped vertically.
_UWORD gr_flm_loop	in	Number of lines when reading the addresses repeatedly The number of lines is gr_flm_loop + 1. 0x0000 (0) ~ 0x03FF (1023) If this feature is not necessary, it's recommended that gr_flm_loop is set to a large value enough (e.g. 1023).
vdc4_GrFormat gr_format	in	Graphics format of the frame buffer read signal • VDC4_FORMAT_RGB565: RGB565 • VDC4_FORMAT_RGB888: RGB888 • VDC4_FORMAT_ARGB1555: ARGB1555 • VDC4_FORMAT_ARGB4444: ARGB4444 • VDC4_FORMAT_ARGB8888: ARGB8888 • VDC4_FORMAT_CLUT8: CLUT8 • VDC4_FORMAT_CLUT4: CLUT4 • VDC4_FORMAT_CLUT1: CLUT1 • VDC4_FORMAT_YCC422: YCbCr422, only for graphics 1
vdc4_OnOff gr_endian_on	in	Endian control of data read from buffer (ON/OFF) • VDC4_OFF • VDC4_ON
_UDWORD bg_color	in	Background color When gr_format is set to CLUT1, CLUT4, CLUT8 or YCbCr422, specify value in RGB888 format. Except as noted above, specify value in the same color format specified by gr_format. LSB-justified format. See Figure 5 for the background color format.
vdc4_GraphicsExt * GraEx	in	Graphics Extension parameter This is valid only when Graphics 1 is specified. If it is unnecessary, GraEx can be NULL.

When double buffering is performed, note the limitations stated below about the second frame buffer (`buff_2nd`) memory (see Figure 4).

- The second buffer has the same width (`gr_arc.hw`), height (`gr_arc.vw`) and line offset (`gr_ln_off`) as the first buffer has.
- The second frame buffer should be allocated behind the first buffer.
- The address offset between two frame buffer (Frame offset) should be less than '0x00800000' (8.0Mbyte).
- The address offset between two frame buffer (Frame offset) should be as follows:
  - When `gr_bst_md` (frame buffer burst transfer mode) is set to 0:
    A multiple of 32
  - When `gr_bst_md` is set to 1:
    A multiple of 128

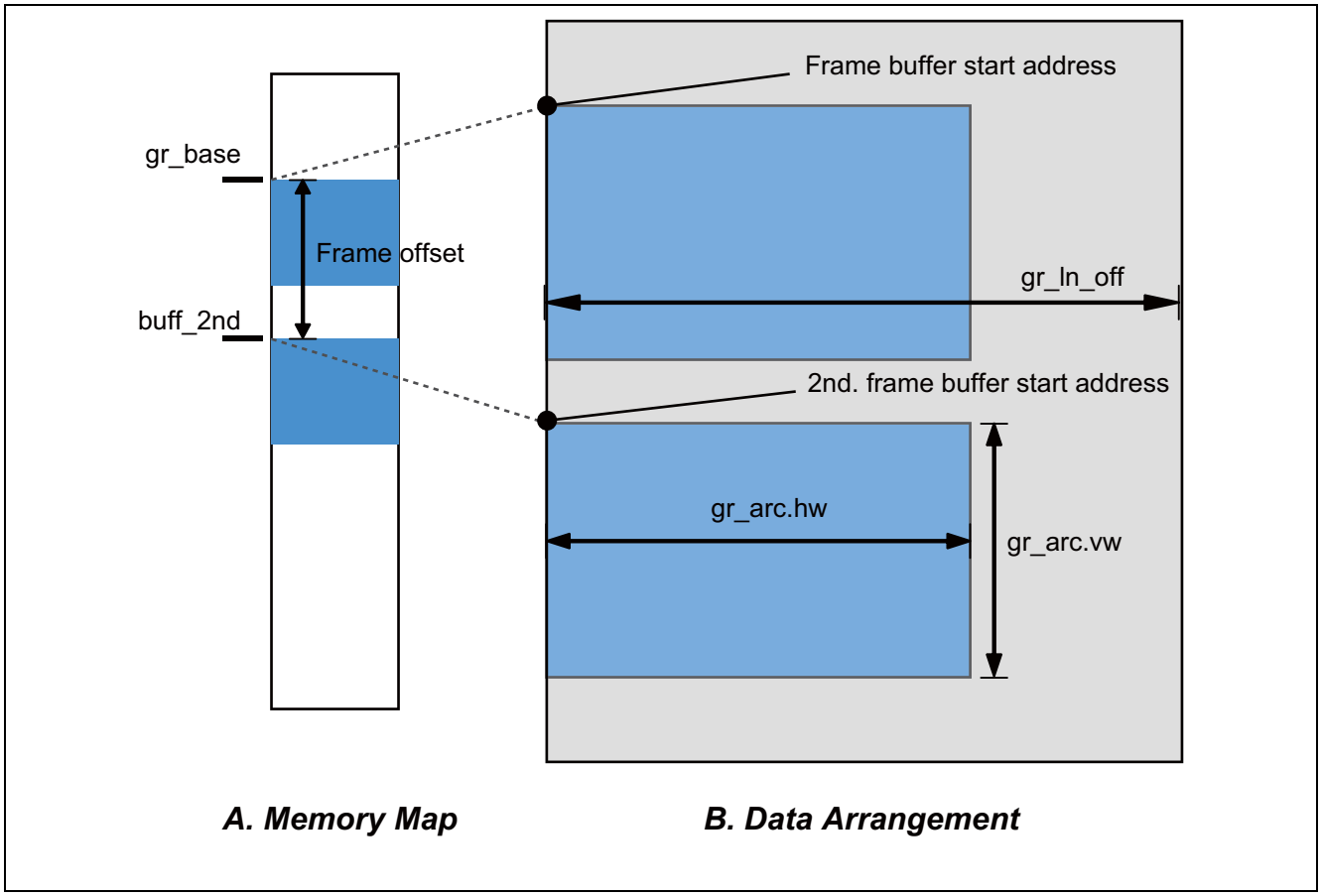
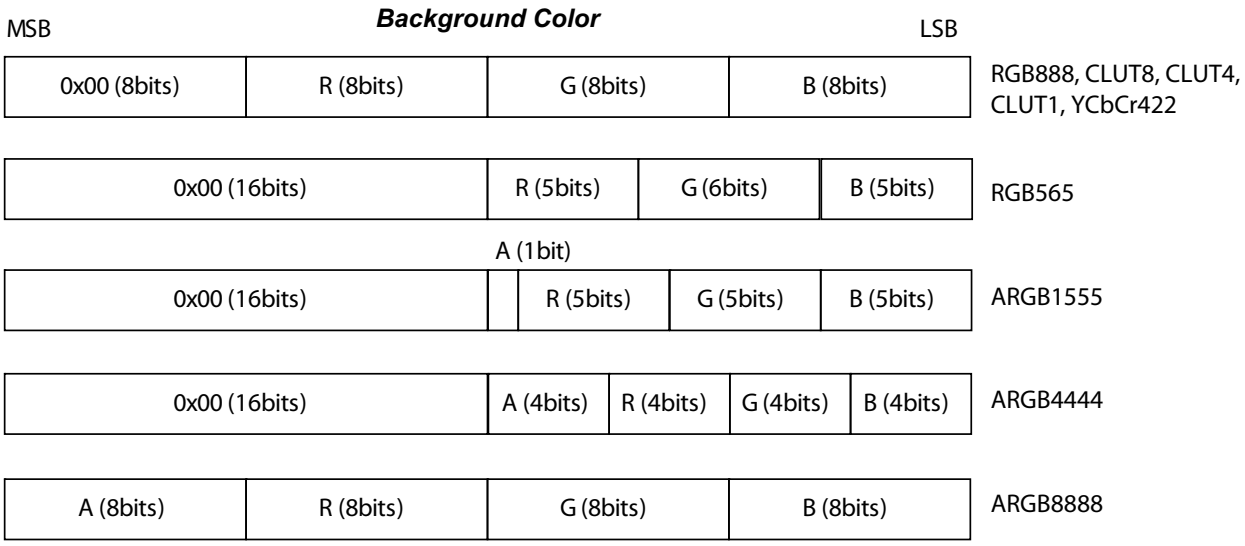


Figure 4 Memory Allocation and Data Arrangement for the 2nd Buffer



Note: Alpha value in ARGB1555, ARGB4444 and ARGB8888 format has no effect on the background color.

Figure 5 Background Color Setting

Members of the structure `vdc4_GraphicsExt` are shown below.

```
typedef struct {
    vdc4_YccSwapFormat gr_ycc_swap ;
    vdc4_YccExchgMode gr_cnv444_md ;
    _UWORD gr_in_hw ;
    _UWORD gr_in_vw ;
    vdc4_Interpolation res_interpotyp ;
    vdc4_OnOff adj_sel ;
} vdc4_GraphicsExt ;
```

Type Member Name	Input/ Output	Description
vdc4_YccSwapFormat gr_ycc_swap	in	Controls swappin g of d ata read fro m buffer in the YCbCr422 format • VDC4_YCCS WAP_CBY0CRY1: CbY0CrY1 • VDC4_YCCS WAP_Y0CBY1CR: Y0CbY1Cr • VDC4_YCCS WAP_CRY0CBY1: CrY0CbY1 • VDC4_YCCS WAP_Y0CRY1CB: Y0CrY1Cb • VDC4_YCCS WAP_Y1CRY0CB: Y1CrY0Cb • VDC4_YCCS WAP_CRY1CBY0: CrY1CbY0 • VDC4_YCCS WAP_Y1CBY0CR: Y1CbY0Cr • VDC4_YCCS WAP_CBY1CRY0: CbY1CrY0 This i s refere nced o nly when graphics fo rmat (gr_format) is set to YCbC r422. And this parameter is valid only when endia n control (gr_endian_on) is set to VDC4_ON.
vdc4_YccExchgMode gr_cnv444_md	in	Interpolation mode fo r YCb Cr422 to YCb Cr444 conversion • VDC4_YCC_ 444_HOLD: Hold interpolation • VDC4_YCC_ 444_AVERAGE: Average interpolation This i s refere nced o nly when graphics fo rmat

		(gr_format) is set to YCbCr422.
_UWORD gr_in_hw	in	Horizontal wi dth of grap hics 1 i nput to scale r [pixel clock cycles] When gr_in_hw is shorter than the horizontal width of display area specified by gr_grc and graphics format (gr_format) is set to RGB565, RGB888 or YCbCr422, enlargement process is done.
_UWORD gr_in_vw	in	Vertical width of graphics 1 input to scaler [lines] When gr_in_vw is shorter than the vertical width of display area specified by gr_grc and graphics format (gr_format) is set to RGB565, RGB888 or YCbCr422, enlargement process is done.
vdc4_Interpolation res_interpotyp	in	Interpolation mode select • VDC4_INTERPOLATION_HOLD: Hold interpolation • VDC4_INTE RPOLATION_LINEAR: Linear interpolation This is referenced only when enlargement process is done.
vdc4_OnOff adj_sel	in	Measures to decre ase th e influen ce b y folding pix els (ON/OFF) • VDC4_OF F • VDC4_O N This is referenced only when enlargement process is done.

2.3.4 VDC4_VideoCreateSurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoCreateSruface(vdc4_LayerID id, const vdc4_VideoAttr *attr) ;	
Parameters	• [in]vdc4_Lye rID id	Layer ID
	• [in]const vdc4_VideoAttr *attr	Video surface attribute parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_UNDEF	Parameter error. Undefined.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- The video surface specified by id is created.
- Scaling and rotation parameters are set.
- IP conversion and field determination signal (RES_FLD_DLY_SEL) are controlled.
- The state of the video surface specified by id is transitioned from 'Init' to 'Ready'.

The procedure for the scaling and rotation of the input video image is as below (see Figure 6).

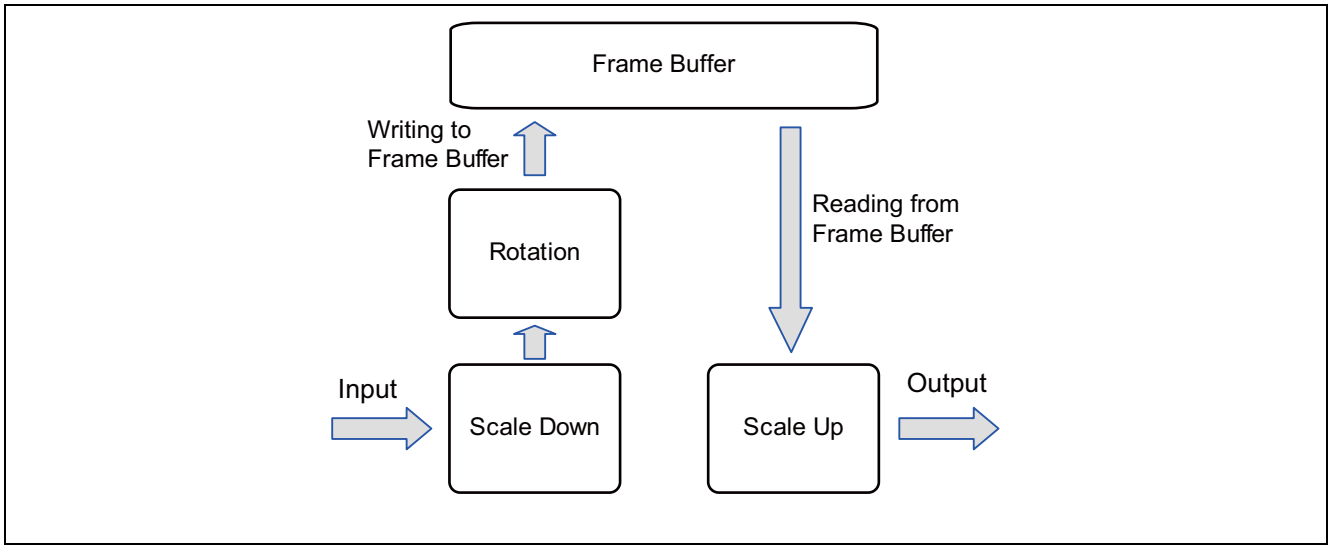
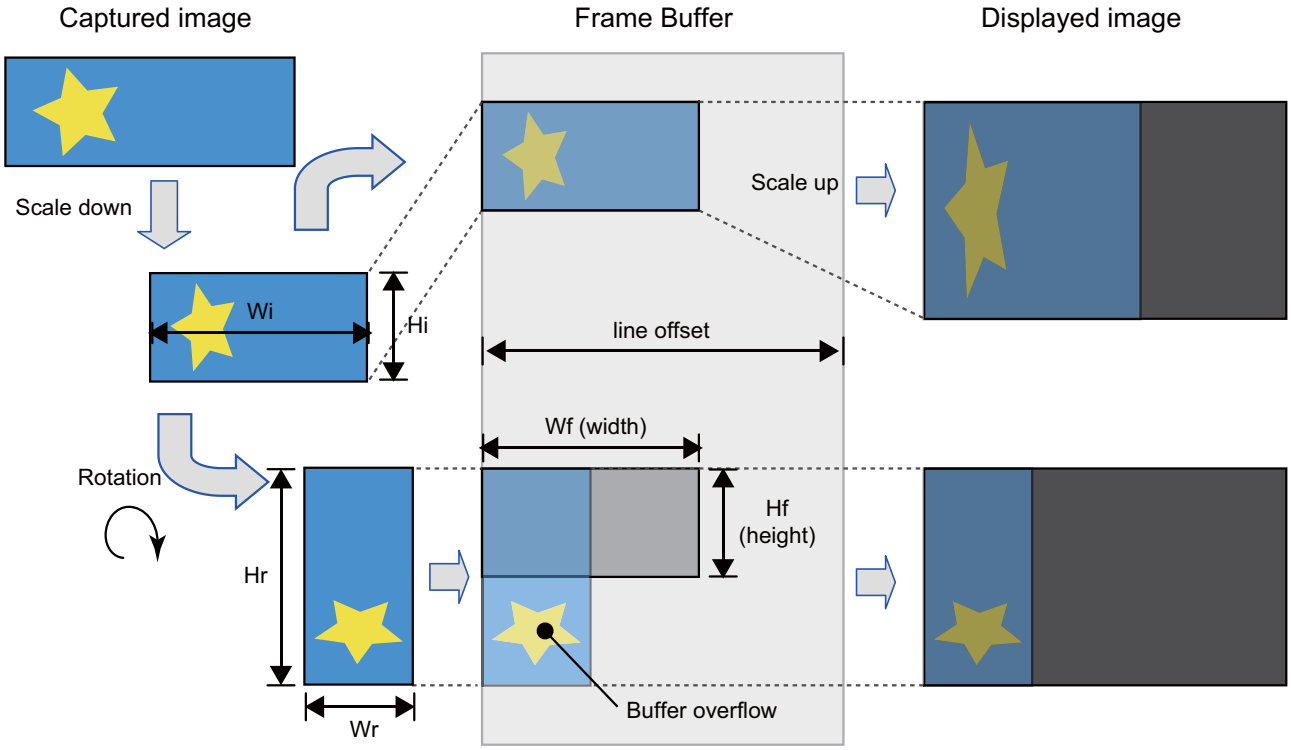


Figure 6 Scaling and Rotation Processing Flow

As shown in Figure 6, the writing to frame buffer is performed before scaling up. Therefore, it is not necessary to consider the scaled up size to decide the size of frame buffer. But the writing to frame buffer is performed after rotation. When the input video image is rotated by 90 degrees or 270 degrees, width and height are exchanged.

Figure 7 shows that the required size of frame buffer is changed by scaling and rotation. In the top of the figure, the image to be captured is scaled down and written to frame buffer. Enough memory is allocated to frame buffer to write image data. In the bottom of the figure, width and height are exchanged after rotation. Therefore, buffer overflow occurs.

It is necessary to consider the scaling down and rotation to decide the size of frame buffer for the image to be captured. The values of frame buffer line offset address (`res_ln_off`) and frame buffer frame offset address (`res_flm_off`) are affected by the scaling down and rotation. For `res_ln_off` and `res_flm_off`, see the description of the structure `vdc4_Res_FrameBuff`.



Note: Wi, Hi: The width and height of the captured image after scaling down.
Wr, Hr: The width and height of the captured image after rotation. Wr is equal to Hi, Hr is equal to Wi.
Wf, Hf: The width and height of the frame buffer. Hf is equal to Hi but smaller than Hr.

Figure 7 Video Frame Buffer

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_VIDEO_1: Video surface
vdc4_VideoAttr * attr	in	Video surface attribute parameter NULL should not be specified.

Members of the structure `vdc4_VideoAttr` are shown below.

```
typedef struct {
    vdc4_VideoType type ;
    _SINT inp_sel ;
    _UWORD inp_fh50 ;
    _UWORD inp_fh25 ;
    vdc4_ExtInSig *ExtSig ;
    vdc4_InpDlay *dly ;
    vdc4_AreaRect res ;
    vdc4_AreaRect res_p ;
    vdc4_ScalingRot *ScaleRot ;
    vdc4_ResFrameBuff frame ;
    vdc4_VideoDisplay *Disp ;
} vdc4_VideoAttr ;
```

Type Member Name	Input/ Output	Description
vdc4_VideoType type	in	Type of video surface • VDC4_VIDEO_TYPE_DISPLAY: Display • VDC4_VIDEO_TYPE_RECORD: Record
_SINT inp_sel	in	Input select • 0: Video decoder output signals • 1: Signals supplied via the external input pins When panel clock source select (panel_icksel) is set to VDC4_LCDPANEL_CLKSEL_IMG in VDC4_Initialize, inp_sel should be set to '0'. When panel clock source select is set to VDC4_LCDPANEL_CLKSEL_IMG_VG in VDC4_Initialize, inp_sel should be set to '1'.
_UWORD inp_fh50	in	Vsync signal 1/2fH phase timing 0x0000 ~ 0x03FF [clock cycles]
_UWORD inp_fh25	in	Vsync signal 1/4fH phase timing 0x0000 ~ 0x03FF [clock cycles]
vdc4_ExtInSig * ExtSig	in	External input signal parameter When input select (inp_sel) is set to '1', NULL should not be specified. When input select is set to '0', ExtSig is ignored.
vdc4_InpDlay * dly	in	Sync signal delay adjustment parameter If it is unnecessary, NULL can be set.
vdc4_AreaRect res	in	Image area to be captured See section 2.1.3 for the structure vdc4_AreaRect. res.hs should be 16 clock cycles or greater. res.hs + res.hw should be 2015 clock cycles or less. And res.hw

		should be 4 clock cycles alignment. res.vs should be 4 lines or greater. res.vs + res.vw should be 2039 lines or less. And res.vw should be 4 lines alignment.
vdc4_AreaRect res_p	in	Image output enable signal See section 2.1.3 for the structure vdc4_AreaRect. res_p.hs should be 16 clock cycles or greater. res_p.hs + res_p.hw should be 2015 clock cycles or less. And res_p.hw should be 4 clock cycles alignment. res_p.vs should be 4 lines or greater. res_p.vs + res_p.vw should be 2039 lines or less. And res_p.vw should be 4 lines alignment.
vdc4_ScalingRot * ScaleRot	in	Scaling and rotation parameter If it is unnecessary, NULL can be set. When pointer ScaleRot is set to NULL, scaling and rotation parameters shall be set as following Table 10.
vdc4_ResFrameBuff frame	in	Video surface frame buffer parameter
vdc4_VideoDisplay * Disp	in	Video surface display parameter NULL should not be specified when type of video surface (type) is set to display.

Members of the structure `vdc4_ExtInSig` are shown below.

```
typedef struct {
    vdc4_InpFormat format ;
    vdc4_OnOff inp_endian_on ;
    vdc4_OnOff inp_swap_on ;
    vdc4_Edge inp_pxd_edge ;
    vdc4_Edge inp_vs_edge ;
    vdc4_Edge inp_hs_edge ;
    _SINT inp_vs_inv ;
    _SINT inp_hs_inv ;
    _SINT inp_f525_625 ;
    _SINT inp_h_edge_sel ;
    vdc4_InpHpos inp_h_pos ;
} vdc4_ExtInSig ;
```

Type **Member Name**	**Input/ Output**	**Description**
vdc4_InpFormat format	in	External input format select • VDC4_INFORMAT_RGB888: RGB888 • VDC4_INFORMAT_RGB666: RGB666 • VDC4_INFORMAT_RGB565: RGB565 • VDC4_INFORMAT_BT656: BT656 • VDC4_INFORMAT_BT601: BT601
vdc4_OnOff inp_endian_on	in	External input bit endian change ON/OFF control • VDC4_OFF • VDC4_ON
vdc4_OnOff inp_swap_on	in	External input B/R signal swap ON/OFF control • VDC4_OFF • VDC4_ON
vdc4_Edge inp_pxd_edge	in	Clock edge select for capturing external input video signals

		• VDC4_EDGE_RISING: Rising edge • VDC4_EDGE_FALLING: Falling edge
vdc4_Edge inp_vs_edge	in	Clock e dge select for ca pturing external input Vsync signal • VDC4_EDGE_RISING: Rising edge • VDC4_EDGE_FALLING: Falling edge
vdc4_Edge inp_hs_edge	in	Clock edge select for capturing external inp ut Hsync signal • VDC4_EDGE_RISING: Rising edge • VDC4_EDGE_FALLING: Falling edge
_SINT inp_vs_inv	in	External input Vsync signal inversion control • 0: Not inverted (positive polarity) • 1: Inverted (negative polarity)
_SINT inp_hs_inv	in	External input Hsync signal inversion control • 0: Not inverted (positive polarity) • 1: Inverted (negative polarity)
_SINT inp_f525_625	in	Number of lines for BT. 656 inp ut of external in put system • 0: 525 lines • 1: 625 lines
_SINT inp_h_edge_sel	in	Hsync sig nal referen ce select for BT.656 format of external input system • 0: EAV • 1: SAV
vdc4_InpHpos inp_h_pos	in	Y/Cb/Y/Cr data string start timing with res pect to Hsync reference • VDC4_INP_H_POS_CBY CRY: Cb/Y/Cr/Y • VDC4_INP_ H_POS_YCRYCB: Y/Cr/Y/Cb • VDC 4_INP_H_POS_CRYCBY: Cr/Y/Cb/Y • VDC4_INP_H_POS_YCBYCR: Y/Cb/Y/Cr

Members of the structure `vdc4_InpDlay` are shown below.

```
typedef struct {
    _UWORD inp_vs_dly_l ;
    _UWORD inp_vs_dly ;
    _UWORD inp_hs_dly ;
    _UWORD inp_fld_dly ;
} vdc4_InpDlay ;
```

Type Member Name	Input/ Output	Description
_UWORD inp_vs_dly_l	in	Number of lines for dela ying Vsync sign al and field differentiation signal 0 ~ 7 [lines]
_UWORD inp_vs_dly	in	Vsync signal delay amount 0 ~ 254 [clock cycles]
_UWORD inp_hs_dly	in	Hsync signal delay amount 0 ~ 254 [clock cycles]
_UWORD inp_fld_dly	in	Field differentiation signal delay amount 0 ~ 254 [clock cycles]

Members of the structure `vdc4_ScalingRot` are shown below.

```
typedef struct {
    vdc4_OnOff res_pfil_sel ;
    vdc4_Interpolation res_interpotyp ;
    vdc4_RotationType rot ;
    vdc4_OnOff adj_sel ;
} vdc4_ScalingRot ;
```

Type Member Name	Input/ Output	Description
vdc4_OnOff res_pfil_sel	in	Prefilter mode select for brightness signals (ON/OFF) • VDC4_OF F • VDC4_O N
vdc4_Interpolation res_interpotyp	in	Interpolation mode select • VDC4_INTERPOLATION_HOLD: Hold interpolation • VDC4_INTE RPOLATION_LINEAR: Linear interpolation
vdc4_RotationType rot	in	Rotation control and horizontal mirroring • VD C4_ROT_NORMAL: Normal • VDC4_ROT_MIRROR: Horizontal mirroring • VDC4_ROT_90_DEG: 90 degree rotation • VDC4_ROT_180_DEG: 180 degree rotation • VDC4_ROT_270_DEG: 270 degree rotation
vdc4_OnOff adj_sel	in	Measures to decrease the influence by the lack of last-input line, folding lines and folding pixels (ON/OFF) • VDC4_OF F • VDC4_O N This parameter affects the scaling ratio except for the horizontal scale-down ratio. In the case of scal e-down, measures to decrease the influence by lack of last-input pixel are always taken.

If settings scaling and rotation parameters is unnecessary and `ScaleRot` is set to NULL, settings can be omitted. In this case, scaling and rotation parameters shall be set as Table 10.

Table 10 Scaling and Rotation Default Value

Parameter De	fault Value
res_pfil_sel	VDC4_OFF
res_interpotyp	VDC4_INTERPOLATION_HOLD
rot	VDC4_ROT_NORMAL
adj_sel	VDC4_OFF

In this driver, captured image size (`res`) and output image size (`res_p`) are compared automatically and if necessary, scaling processing is performed. In comparison process, the change of size by rotation (i.e., width and height are exchanged) is considered.

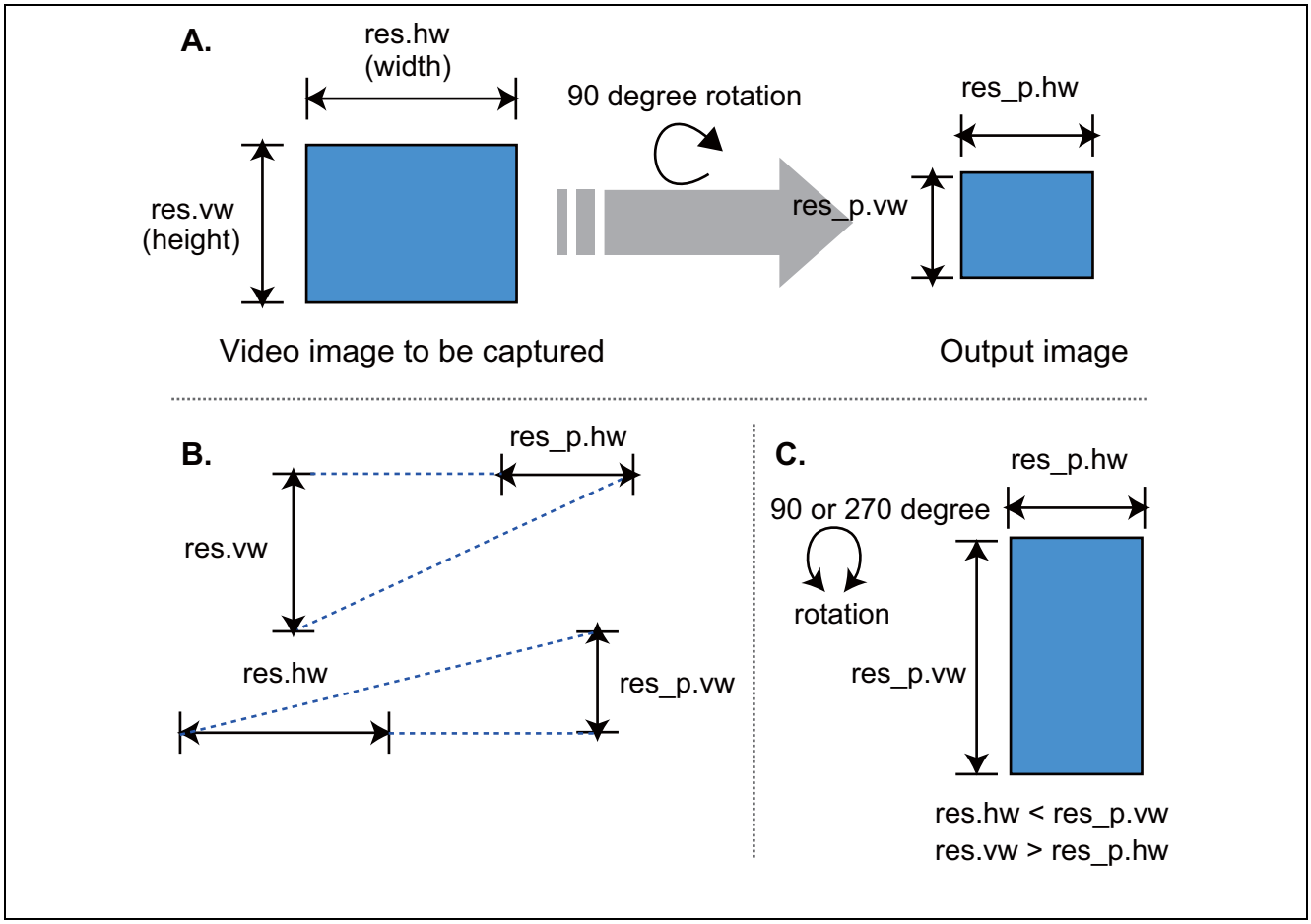


Figure 8 Scaling and Rotation

Figure 8 shows an example of scaling and rotation processing. 'A' in the Figure 8, input video image is scaled and rotated by 90 degrees. 'B' in the Figure 8 shows that horizontal size and vertical size to be compared are exchanged by 90 degree rotation. 'C' in the Figure 8 shows prohibited processing. It is prohibited setting the scaling and rotation parameters to fulfill all conditions as follows:

- 90 or 270 degree rotation processing is performed.
- The width of input image is shorter than the height of output image.
- The height of input image is longer than the width of output image.

In this driver, initial phase control for IP conversion is performed in the scaling and rotation process. For this processing, there is nothing that users have to do. Registers and parameters set in the initial phase control processing are as below (see Table 11).

Table 11 Initial Scaling Phase Settings for IP Conversion

Rotation	Horizontal Scaling	Vertical Scaling	Register	Value
Normal	Scale up or down	Scale up or down	RES_TOP_INIPASE	2048
horizontal mirroring	Scale up or down	Scale up or down	RES_TOP_INIPASE	2048
90 degree rotation	Scale down	Scale down	RES_TOP_INIPASE	2048
	Scale up and down	-	RES_TOP_INIPASE	2048
	Scale up	Scale up	RES_US_HB_INIPHASE	2048
180 degree rotation	Scale up or down	Scale down	RES_TOP_INIPASE	2048
	Scale up or down	Scale up	RES_BTM_INIPASE	2048
270 degree rotation	Scale down	Scale down	RES_TOP_INIPASE	2048
	Scale up and down	-	RES_TOP_INIPASE	2048
	Scale up or down	Scale up	RES_US_HT_INIPHASE	2048

Members of the structure `vdc4_ResFrameBuff` are shown below.

```
typedef struct {
    _SINT res_bst_md ;
    void *res_base ;
    _UWORD res_ln_off ;
    _UDWORD res_flm_off ;
    vdc4_ResFormat res_md ;
    vdc4_OnOff res_dth_on ;
    vdc4_ResFsRate res_fs_rate ;
    _SINT res_fld_sel ;
    _SINT res_inter ;
    _UWORD res_flm_num ;
    _SINT res_loop ;
    _UDWORD bg_color ;
} vdc4_ResFrameBuff ;
```

Type **Member Name**	**Input/ Output**	**Description**
_SINT res_bst_md	in	Transfer burst length fo r frame b uffer writing a nd reading • 0: 32-byte transfer • 1: 128-byte transfer
void * res_base	in	Frame buffer base address NULL must not be specified. When res_bst_md is set to 0: Set address on 32-byte alignment. When res_bst_md is set to 1: Set address on 128-byte alignment.
_UWORD res_ln_off	in	Frame buffer line offset address [byte] When res_bst_md is set to 0: A multiple of 32 When res_bst_md is set to 1:

		A multiple of 128
_UDWORD res_flm_off	in	Frame buffer frame offset address [byte] When res_bst_md is set to 0: A multiple of 32 When res_bst_md is set to 1: A multiple of 128
vdc4_ResFormat res_md	in	Video-signal writing format • VDC4_RES_MD_YCC422: YCbCr422 • VDC4_RES_MD_RGB565: RGB565 • VDC4_RES_MD_RGB888: RGB888
vdc4_OnOff res_dth_on	in	Dither correction ON/OFF • VDC4_OFF • VDC4_ON
vdc4_ResFsRate res_fs_rate	in	Writing rate • VDC4_RES_FS_RATE_PER1: 1/1 • VDC4_RES_FS_RATE_PER2: 1/2 • VDC4_RES_FS_RATE_PER4: 1/4 • VDC4_RES_FS_RATE_PER8: 1/8
_SINT res_fld_sel	in	Write field select • 0: Top field • 1: Bottom field This parameter is valid when res_fs_rate is not set to VDC4_RES_FS_RATE_PER1.
_SINT res_inter	in	Field operating mode select • 0: Progressive • 1: Interlace
_UWORD res_flm_num	in	Number of frames of buffer (defined by res_flm_num + 1) When type of video surface (type) is set to: recording 0x0000 ~ 0x03FF display 0 or 1 (one or two planes) For rotated image Setting res_flm_num to 0 is prohibited.
_SINT res_loop	in	Frame buffer write mode select • 0: Frame mode • 1: Line mode (read as ring buffer)
_UDWORD bg_color	in	Background color setting When res_md is set to YCbCr422, specify value in CrYCb format. Except as noted above, specify value in RGB888 format. LSB-justified format. See Figure 9 for the background color format.

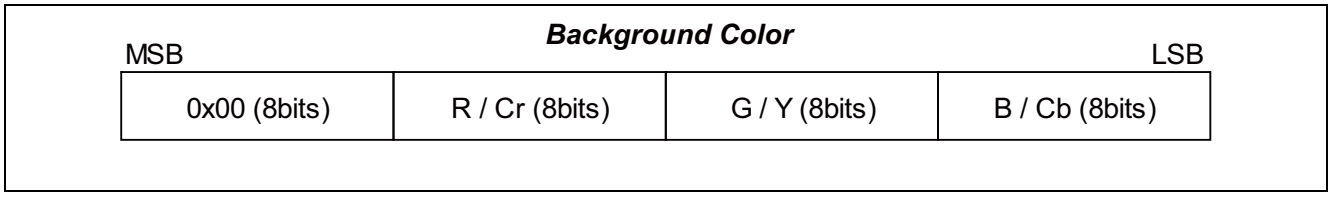


Figure 9 Background Color Setting

The equation to convert between RGB and YCbCr are as below (Typical value for standard SMPTE 293M).

Y = 0.299R + 0.587G + 0.114B
Cb = -0.169R – 0.331G + 0.500B + 128.0
Cr = 0.500R – 0.419G – 0.081B + 128.0

Members of the structure `vdc4_VideoDisplay` are shown below.

```
typedef struct {
    _UWORD res_vsdly ;
    vdc4_OnOff gr_endian_on ;
    vdc4_YccSwapFormat gr_ycc_swap ;
    vdc4_YccExchgMode gr_cnv444_md ;
} vdc4_VideoDisplay ;
```

Type Member Name	Input/ Output	Description
_UWORD res_vsdly	in	Vsync signal delay control 0x0000 ~ 0x00FF delay[usec]: res_vsdly x output Hsync period[usec]
vdc4_OnOff gr_endian_on	in	Endian control of data read from buffer (ON/OFF) • VDC4_OF F • VDC4_O N
vdc4_YccSwapFormat gr_ycc_swap	in	Controls swappin g of d ata read fro m buffer in the YCbCr422 format • VDC4_YCCS WAP_CBY0CRY1: CbY0CrY1 • VDC4_YCCS WAP_Y0CBY1CR: Y0CbY1Cr • VDC4_YCCS WAP_CRY0CBY1: CrY0CbY1 • VDC4_YCCS WAP_Y0CRY1CB: Y0CrY1Cb • VDC4_YCCS WAP_Y1CRY0CB: Y1CrY0Cb • VDC4_YCCS WAP_CRY1CBY0: CrY1CbY0 • VDC4_YCCS WAP_Y1CBY0CR: Y1CbY0Cr • VDC4_YCCS WAP_CBY1CRY0: CbY1CrY0 This i s refe renced o nly when video-signal writing format (res_md) is set to YCC42 2. And this para meter is valid o nly when endian control (gr_endian_on) is set to VDC4_ON.
vdc4_YccExchgMode gr_cnv444_md	in	Interpolation mode fo r YCb Cr422 to YCb Cr444 conversion • VDC4_YCC_ 444_HOLD: Hold interpolation • VDC4_YCC_ 444_AVERAGE: Average interpolation This i s refe renced o nly when video-signal writing format (res_md) is set to YCbCr422.

2.3.5 VDC4_DestroySurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_DestroySruface(vdc4_LayerID id) ;	
Parameters	• [in]vdc4_LyerID id	Layer ID
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.

Description

In this function, the operations are performed as below.

- The surface specified by id is destroyed.
- If the surface specified by id is running (the state of the surface is 'Run'), the surface is destroyed after stop.
- The state of the surface specified by id is transitioned to 'Init'.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_ 3: Graphics surface 3 • VDC4_SURFACE_VIDEO_1: Video surface

2.3.6 VDC4_RegistCallbackFunc

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_RegistCallbackFunc(vdc4_IntType type, void (*callback)(vdc4_IntType), _UWORD line_num) ;	
Parameters	• [in]vdc4_IntT ype type	Type of interrupt
	• [in]void (*callback)(vdc4_IntType)	Pointer to the callback function
	• [in]_UWORD line_num	Line interrupt set
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- The specified callback function is registered.

The only one of callback function can be registered for each interrupt type. When the interrupt type for which the callback function has already registered is specified, the registration of the callback function is overwritten. When the callback function pointer (`callback`) is sett to '0', the callback function is unregistered. Number of lines determining the timing to generate an interrupt (`line_num`) is valid only when line signal for panel output is specified as interrupt type.

The callback function is executed in the interrupt handler and returns interrupt type to the argument. When interrupt handler is not defined, the callback function registered by this function is not executed. For definition of interrupt handler, see 1.7. And for the generation of interrupt, it is necessary to enable the interrupt in VDC4_Initialize.

The registered callback function is executed depending on surface states and interrupt types (see Table 12). The surfaces not shown in Table 12 means that the callback function is not executed (e.g. When graphics surface 1 is created, the callback function of VI_VSYNC interrupt is not executed.).

Table 12 Surface States and Callback

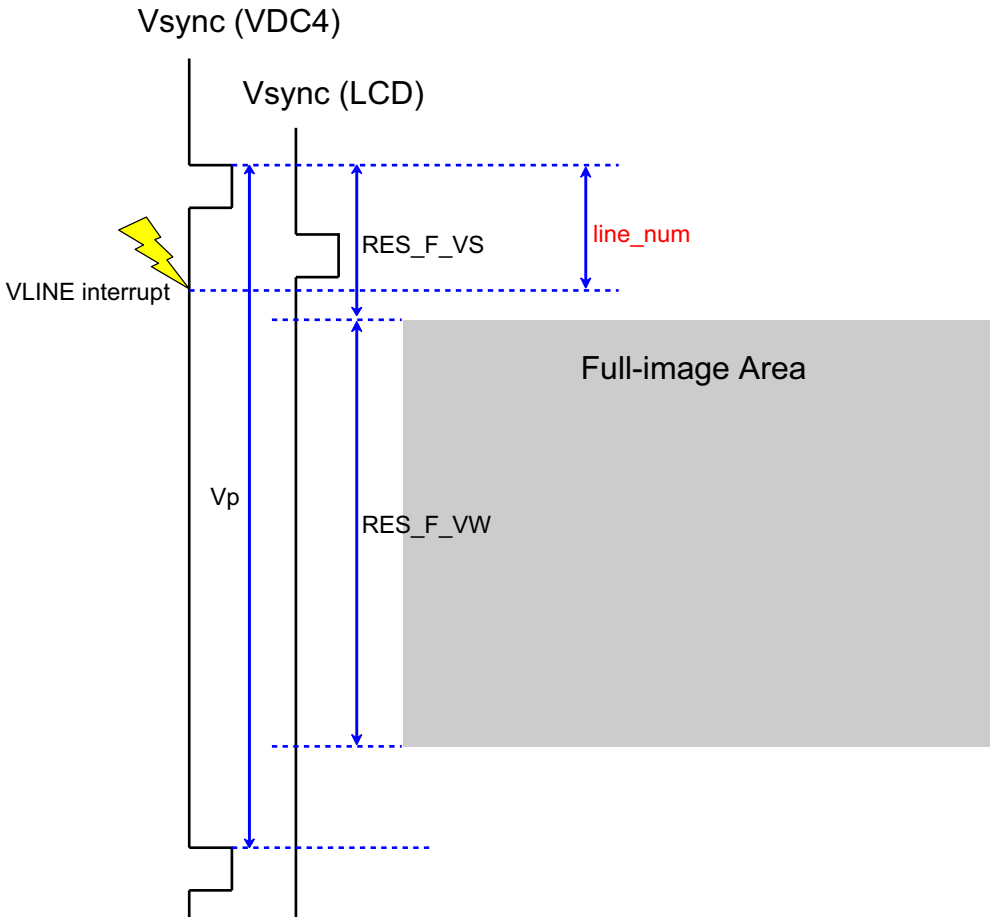
Type of Interrupt	Surface	Surface States		
		Init	Ready	Run
VI_VSYNC	Video	x	o	o
LO_VSYNC	All surface	x	o	o
VSYNCERR	All surface	x	o	o
VLINE	All surface	x	x	o
VFIELD	Video	x	x	o
VBUFERR1	Video	x	x	o
VBUFERR2	Gra phics 1	x	x	o
VBUFERR3	Gra phics 2	x	x	o
VBUFERR4	Gra phics 3	x	x	o

Note: 'o' in the table means that the callback function is called.
'x' in the table means that the callback function is not called.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_IntType type	in	Type of interrupt • VDC4_INT_TYPE_VI_VSY NC: VI_VSYNC • VDC4_INT _TYPE_LO_VSYNC: LO_VSYNC • VDC4_INT_TYPE_VSYNCERR: VSYNCERR • VDC4_INT_TYPE_VLINE: VLINE • VDC4_INT_TYPE_VFIELD: VFIELD • VDC4_INT_TYPE_VBU FERR1: VBUFERR1 • VDC4_INT_TYPE_VBU FERR2: VBUFERR2 • VDC4_INT_TYPE_VBU FERR3: VBUFERR3 • VDC4_INT_TYPE_VBUFERR4: VBUFERR4
void (*callback)(vdc4_IntType)	in	Pointer to the callback function If it is unnecessary, 0 should be set. Syntax void CallbackFunc(vdc4_IntType cb_type) ; Parameters • [in]vdc4_IntT ype Type of interrupt cb_type Return Values • void Description This callback function is called when the specified type of interrupt (type) is generated. And this function returns type of interrupt to the argument.
_UWORD line_num	in	Line number timing to generate VLINE interrupt signal This parameter is valid only when the type of interrupt (type) is VDC4_INT_TYPE_VLINE.

The VLINE interrupt timing is specified by line_num in the line number. This line number is counted from VDC4 internal vertical sync signal (see Figure 10). The timing relationship between VDC4 internal Vsync and LCD Vsync is defined by the setting of LCD TCON signals to drive the LCD panel. And the timing relationship between VDC4 internal Vsync and the valid period of the full screen to be displayed is defined by the setting of full-screen enable control. For more information about these settings, see 2.3.1 VDC4_Initialize and 4.3 Setting Example for LCD.



Vsync (VDC4): VDC4 internal vertical sync signal
Vsync (LCD): LCD vertical sync signal
RES_F_VS: Vertical enable signal start position for full screen
RES_F_VW: Vertical enable signal width for full screen
Vp: Vertical period
line_num: Line number timing to generate VLINE interrupt signal

Figure 10 VLINE Interrupt Timing

2.3.7 VDC4_GraphicsStartSurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsStartSurface(vdc4_LayerID id, const vdc4_ShowGraphics *Graphics) ;	
Parameters	• [in]vdc4_Lye rID id • [in]const vdc4_ShowGraphics * Graphics	Layer ID Graphics display settings
Return Values	• vdc4_E rrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_INVALID	Error code Normal end. Surface error. Illegal surface. Surface error. Illegal state. Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- The display of the graphics surface specified by id is started.
- The specified parameters for graphics are set.
- The state of the graphics surface specified by id is transitioned from 'Ready' to 'Run'.

Before the settings are reflected, it will take a period of time equal to 1 Vsync period time at the most.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_ShowGraphics * Graphics	in	Graphics display settings NULL should not be specified.

Members of the structure `vdc4_ShowGraphics` are shown below.

```
typedef struct {
    vdc4_CtrlFrameBuff *CtrlFrameBuff ;
    vdc4_CtrlDispSel *CtrlDispSel ;
    vdc4_AreaRect *Gr_AreaRect ;
} vdc4_ShowGraphics ;
```

Type Member Name	Input/ Output	Description
vdc4_CtrlFrameBuff * CtrlFrameBuff	in	Graphics frame buffer control Settings will be left unchanged if NULL is specified.
vdc4_CtrlDispSel * CtrlDispSel	in	Graphics display selection Settings will be left unchanged if NULL is specified.
vdc4_AreaRect * Gr_AreaRect	in	Display area Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect. Gr_AreaRect->hs should be 16 clock cycles or greater. Gr_AreaRect->hs + Gr_AreaRect->hw should be 2015 clock cycles or less. Gr_AreaRect->hw should be 3 clock cycles or greater. Gr_AreaRect->vs should be 4 lines or greater. Gr_AreaRect->vs + Gr_AreaRect->vw should be 2039 lines or less.

Members of the structure `vdc4_CtrlFrameBuff` are shown below.

```
typedef struct {
    _SINT ChgFrameNum ;
    void *Buffer ;
} vdc4_CtrlFrameBuff ;
```

Type Member Name	Input/ Output	Description
_SINT ChgFrameNum	in	Frame number of frame buffer 0 or 1 Only valid when double buffering is enabled. Frame number is 0 initialized when graphics surface is created.
void * Buffer	in	Frame buffer base address Ignored in this driver call.

Members of the structure `vdc4_CtrlDispSel` are shown below.

```
typedef struct {
    vdc4_DispSel gr_disp_sel ;
} vdc4_CtrlDispSel ;
```

Type Member Name	Input/ Output	Description
vdc4_DispSel gr_disp_sel	in	Graphics display mode • VDC4_DISPSEL_BAC K: Background • VDC4_DISPSEL_LOWER: Lower-layer graphics • VDC4_DISPSEL_CURRE NT: Current graphics • VDC4_DISPSEL_BLEND : Blended current graphics with lower-layer graphics This setting is prohibited for the graphics 1 process. Before starting display, graphi cs display mode is as below (see Table 13).

The graphics display mode is transitioned depending on the state of the surfaces as below.

Table 13 Graphics Display Mode and Surface

State / Surface	Init Read	y	Run
Video Backgroun	d	Background	Lower-layer graphics
Graphics 1	Background	Background	User-specified value
Graphics 2	Lower-layer graphics	Lower-layer graphics	User-specified value
Graphics 3	Lower-layer graphics	Lower-layer graphics	User-specified value

For the enlarged graphics display in the graphics 1 process, the graphics display mode should be set to 'Lower-layer graphics' by user.

2.3.8 VDC4_VideoStartSurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoStartSurface(vdc4_LayerID id) ;	
Parameters	• [in]vdc4_LyerID id	Layer ID
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.

Description

In this function, the operations are performed as below.

- The video surface specified by id is started.
- The state of the video surface specified by id is transitioned from 'Ready' to 'Run'.

Before the settings are reflected, it will take a period of time equal to 1 Vsync period time at the most.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_VIDEO_1: Video surface

2.3.9 VDC4_GraphicsChangeParam

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsChangeParam(vdc4_LayerID id, const vdc4_ShowGraphics *Graphics) ;	
Parameters	• [in]vdc4_Lye rID id • [in]const vdc4_ShowGraphics * Graphics	Layer ID Graphics display settings
Return Values	• vdc4_E rrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_INVALID	Error code Normal end. Surface error. Illegal surface. Surface error. Illegal state. Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- The frame buffer is exchanged for specified buffer.
- The graphics display mode is changed.
- The display area is changed.

Before the settings are reflected, it will take a period of time equal to 1 Vsync period time at the most.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_ShowGraphics * Graphics	in	Graphics display settings NULL should not be specified.

Members of the structure `vdc4_ShowGraphics` are shown below.

```
typedef struct {
    vdc4_CtrlFrameBuff *CtrlFrameBuff ;
    vdc4_CtrlDispSel *CtrlDispSel ;
    vdc4_AreaRect *Gr_AreaRect ;
} vdc4_ShowGraphics ;
```

Type Member Name	Input/ Output	Description
vdc4_CtrlFrameBuff * CtrlFrameBuff	in	Graphics frame buffer control Settings will be left unchanged if NULL is specified.
vdc4_CtrlDispSel * CtrlDispSel	in	Graphics display selection Settings will be left unchanged if NULL is specified. See section 2.3.7 for the structure vdc4_CtrlDispSel.
vdc4_AreaRect * Gr_AreaRect	in	Display area Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect. Gr_AreaRect->hs should be 16 clock cycles or greater. Gr_AreaRect->hs + Gr_AreaRect->hw should be 2015 clock cycles or less. Gr_AreaRect->hw should be 3 clock cycles or greater. Gr_AreaRect->vs should be 4 lines or greater. Gr_AreaRect->vs + Gr_AreaRect->vw should be 2039 lines or less.

Members of the structure `vdc4_CtrlFrameBuff` are shown below.

```
typedef struct {
    _SINT ChgFrameNum ;
    void *Buffer ;
} vdc4_CtrlFrameBuff ;
```

Type Member Name	Input/ Output	Description
_SINT ChgFrameNum	in	Frame number of frame buffer 0 or 1 Only valid when double buffering is enabled.
void * Buffer	in	Frame buffer base address Only valid when single buffering is enabled.

2.3.10 VDC4_VideoChangeParam

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoChangeParam(vdc4_LayerID id, const vdc4_ChangeVideo *ChgVideo) ;	
Parameters	• [in]vdc4_Lye rID id • [in]const vdc4_ChangeVideo * ChgVideo	Layer ID Change video setting
Return Values	• vdc4_E rrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	Error code Normal end. Surface error. Illegal surface. Surface error. Illegal state. Parameter error. Out of range. Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- Image area to be captured is changed.
- Image output enable signal is changed.
- Scaling and rotation parameters are changed.

When scaling and/or rotation parameters are changed, note frame buffer overflow. If enough memory is not allocated to the video surface, the setting parameters in this function might cause incorrect buffer access. For information about frame buffer of video image, see 2.3.4.

Before the settings are reflected, it will take a period of time equal to 1 Vsync period time at the most.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_VIDEO_1: Video surface
vdc4_ChangeVideo * ChgVideo	in	Change video setting NULL should not be specified.

Members of the structure `vdc4_ChangeVideo` are shown below.

```
typedef struct {
    vdc4_AreaRect *res ;
    vdc4_AreaRect *res_p ;
    vdc4_ScalingRot *ScaleRot ;
} vdc4_ChangeVideo ;
```

Type Member Name	Input/ Output	Description
vdc4_AreaRect * res	in	Image area to be captured Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect. res->hs should be 16 clock cycles or greater. res->hs + res->hw should be 2015 clock cycles or less. And res->hw should be 4 clock cycles alignment. res->vs should be 4 lines or greater. res->vs + res->vw should be 2039 lines or less. And res->vw should be 4 lines alignment.
vdc4_AreaRect * res_p	in	Image output enable signal Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect. res_p->hs should be 16 clock cycles or greater. res_p->hs + res_p->hw should be 2015 clock cycles or less. And res_p->hw should be 4 clock cycles alignment. res_p->vs should be 4 lines or greater. res_p->vs + res_p->vw should be 2039 lines or less. And res_p->vw should be 4 lines alignment.
vdc4_ScalingRot * ScaleRot	in	Scaling and rotation parameter Settings will be left unchanged if NULL is specified. See section 2.3.4 for the structure vdc4_ScalingRot.

2.3.11 VDC4_StopSurface

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_StopSurface(vdc4_LayerID id) ;	
Parameters	• [in]vdc4_LyerID id	Layer ID
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.

Description

In this function, the operations are performed as below.

- The surface specified by id is stopped.
- The graphics display mode is changed.
- The state of the surface specified by id is transitioned from 'Run' to 'Ready'.

When video surface is stopped, write and read access to the frame buffer is disabled. When graphics surface is stopped, read access to the frame buffer is disabled. For information about the change of the graphics display mode in the stop process, see Table 13.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_ 3: Graphics surface 3 • VDC4_SURFACE_VIDEO_1: Video surface

2.3.12 VDC4_ImageColorMatrix

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_ImageColorMatrix(const vdc4_ColorMatrix *mtx) ;	
Parameters	• [in]const vdc4_ColorMatrix *mtx	Color matrix parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_OTHERS	Parameter error. Others.

Description

In this function, the operations are performed as below.

- Specified color matrix is set.

Two color matrices are available. One is in the input controller and the other is in the image quality improver (see Figure 11).

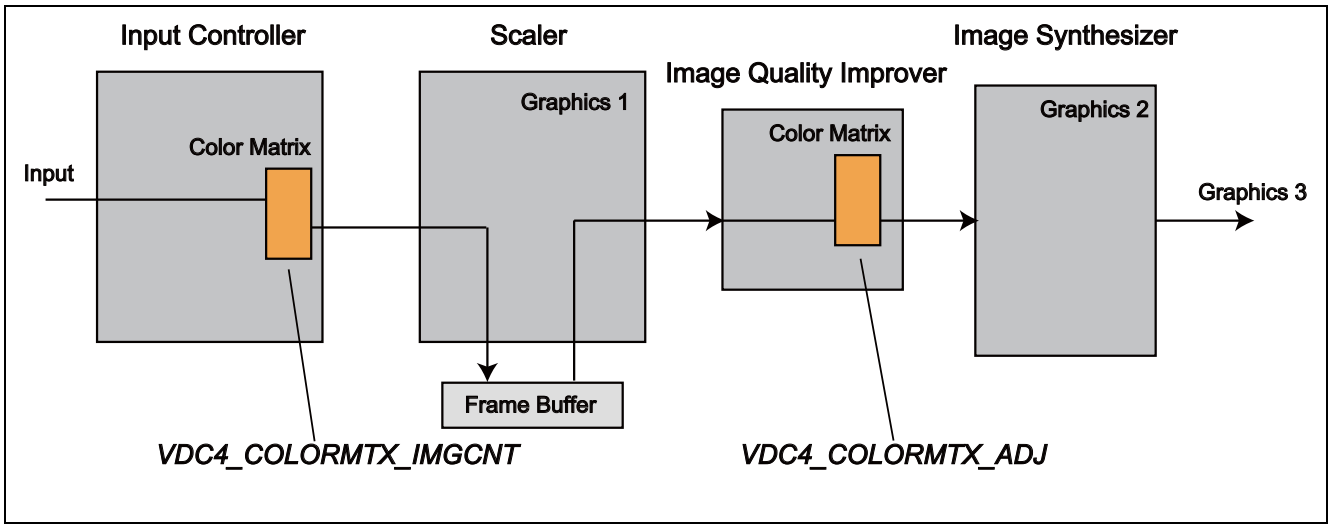


Figure 11 Color Matrix

As shown in Figure 11, color matrix `VDC4_COLORMTX_IMGCNT` acts on input video signal and color matrix `VDC4_COLORMTX_ADJ` acts on input video signal and graphics 1 output signal.

When video surface or graphics 1 surface is used, it is not necessary to set color matrix value. The color matrices are initialized when video surface and graphics 1 surface are created. For information about color matrix initialization, see Table 14 and Table 15.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_ColorMatrix * mtx	in Colo	r matrix parameter NULL should not be specified.

Members of the structure `vdc4_ColorMatrix` are shown below.

```
typedef struct {
    vdc4_ColorMtxModule module ;
    vdc4_ColorMtxMode mode ;
    _UWORD DcOffset[ VDC4_COLORMTX_OFS_NUM ] ;
    _UWORD matrix[ VDC4_COLORMTX_GAIN_NUM ] ;
} vdc4_ColorMatrix ;
```

Type Member Name	Input/ Output	Description
vdc4_ColorMtxModule module	in	Target module of a color matrix (see Figure 11) • VDC4_COLORMTX_IMGCNT: Input controller Output from noise reduction • VDC4_COLORMTX_A DJ: Image quality improver Output from scaler (graphics 1)
vdc4_ColorMtxMode mode	in	Color matrix operating mode • VDC4_COLORMTX_GBR_GBR: GBR => GBR • VDC4_COLORMTX_GBR_YCBCR: GBR => YCbCr • VDC4_COLORMTX_Y CBCR_GBR: YCbCr => GBR • VDC4_COLORMTX_Y CBCR_YCBCR: YCbCr => YCbCr When modul e is set to VDC4_COLORMTX_ADJ, VDC4_COLORMTX_GBR_YCBCR and VDC4_COLORMTX_YCBCR_YCBCR should not be specified.
_UWORD DcOffset[VDC4_COLORMTX_OFS_NUM]	in	Offset adjustment of Y/G, B and R signal Unsigned (0 (-128) ~ 0x00FF (+127))
_UWORD matrix[VDC4_COLORMTX_GAIN_NUM]	in	Gain adjustment of GG, GB, GR, BG, BB, BR, RG, RB and RR Signed (two's complement) (-1024 ~ +1023, 256 = 1.0 [times])

Table 14 shows which color matrix operating mode in two color matrices should be initialized to, when video surface or graphics surface is created. Color matrix operating mode depends on the signal format of video surface and graphics 1 surface.

When the color matrix operating mode is specified, the color matrix value should be set to as shown in Table 15. The conversion between YCbCr and RGB follows SMPTE 293M.

Table 14 Color Matrix Operating Mode Initial Setting

Surface	Input Select	External Input System Format Select	Frame Buffer Video-Signal Writing Format	Frame Buffer Signal Reading Format	VDC4_COLOR-MTX_IMGCNT	VDC4_COLOR-MTX_ADJ
Video	Video decoder output signal	-	YCbCr (YCbCr422)	-*	YCbCr to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	YCbCr to RGB	RGB to RGB
	Signal supplied via the external input pins	YCbCr (BT656, BT601)	YCbCr (YCbCr422)	-*	YCbCr to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	YCbCr to RGB	RGB to RGB
		RGB (RGB888, RGB666, RGB565)	YCbCr (YCbCr422)	-*	RGB to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	RGB to RGB	RGB to RGB
Graphics 1	-	-	-	YCbCr (YCbCr422)	-	YCbCr to RGB
				RGB (RGB565, RGB888, ARGB1555, ARGB4444, ARGB8888, CLUT8, CLUT4, CLUT1)	-	RGB to RGB

Note: * The frame buffer signal writing and reading format of the video surface are the same format.

Table 15 Color Matrix Coefficient (Initial Value)

Color Matrix Parameters	GBR to GBR	GBR to YCbCr	YCbCr to GBR	YCbCr to YCbCr
Offset adjustment of Y/G signal	128	128	128	128
Offset adjustment of B signal	128	128	128	128
Offset adjustment of R signal	128	128	128	128
Y/G signal gain adjustment for Y/G signal output	256	150	256	256
Cb/B signal gain adjustment for Y/G signal output	0	29	1960	0
Cr/R signal gain adjustment for Y/G signal output	0	77	1865	0
Y/G signal gain adjustment for Cb/B signal output	0	1963	256	0
Cb/B signal gain adjustment for Cb/B signal output	256	128	454	256
Cr/R signal gain adjustment for Cb/B signal output	0	2005	0	0
Y/G signal gain adjustment for Cr/R signal output	0	1941	256	0
Cb/B signal gain adjustment for Cr/R signal output	0	2027	0	0
Cr/R signal gain adjustment for Cr/R signal output	256	128	359	256

2.3.13 VDC4_VideoNoiseReduction

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoNoiseReduction(vdc4_OnOff nr1d_on, const vdc4_NoiseReduction *nr) ;	
Parameters	• [in]vdc4_O nOff nr1d_on	Noise reduction ON/OFF control
	• [in]const vdc4_NoiseReduction *nr	Noise reduction parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- Horizontal noise reduction parameters are set.

The operating mode in horizontal noise reduction (i.e., R/G/B mode and Y/Cb/Cr mode) is set automatically depending on the input signals specified in VDC4_VideoCreateSurface.

If this function is not called, the horizontal noise reduction parameters will be left unchanged since a reset. For information about initial value after a reset, see "SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048EJ)".

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_OnOff nr1d_on	in	Noise reduction ON/OFF control • VDC4_OF F • VDC4_ON
vdc4_NoiseReduction * nr	in	Noise reduction parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_NoiseReduction` are shown below.

```
typedef struct {
    vdc4_NRparam y ;
    vdc4_NRparam cb ;
    vdc4_NRparam cr ;
} vdc4_NoiseReduction ;
```

Type Member Name	Input/ Output	Description
vdc4_NRparam y	in	Noise Reduction parameter of Y/G signal
vdc4_NRparam cb	in	Noise Reduction parameter of Cb/B signal
vdc4_NRparam cr	in	Noise Reduction parameter of Cr/R signal

Members of the structure `vdc4_NRparam` are shown below.

```
typedef struct {
    vdc4_NRTap nr1d_tap ;
    _UWORD nr1d_th ;
    vdc4_NRGain nr1d_gain ;
} vdc4_NRparam ;
```

Type Member Name	Input/ Output	Description
vdc4_NRTap nr1d_tap	in TAP	select • VDC4_NR_TAPSEL_1: Adjacent pixel • VDC4_NR_TAPSEL_2: 2 adjacent pixels • VDC4_NR_TAPSEL_3: 3 adjacent pixels • VDC4_NR_TAPSEL_4: 4 adjacent pixels
_UWORD nr1d_th	in	Maximum value of signal coring (absolute value) 0 ~ 0x007F
vdc4_NRGain nr1d_gain	in	Noise Reduction gain adjustment • VDC4_ NR_GAIN_1_2: 1/2 • VDC4_ NR_GAIN_1_4: 1/4 • VDC4_ NR_GAIN_1_8: 1/8 • VDC4_ NR_GAIN_1_16: 1/16

2.3.14 VDC4_ImageEnhancement

Syntax

```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_ImageEnhancement( vdc4_OnOff bkstr_on,
                                      const vdc4_Black *black,
                                      vdc4_OnOff shp_h_on,
                                      const vdc4_EnhanceSharp *sharp,
                                      vdc4_OnOff lti_h_on,
                                      const vdc4_EnhanceLTI *lti,
                                      const vdc4_AreaRect *EnhArea ) ;
```

Parameters	• [in]vdc4_O nOff bkstr_on	Black stretch ON/OFF control
	• [in]const vdc4_Black *black	Black stretch parameter
	• [in]vdc4_O nOff shp_h_on	Sharpness ON/OFF control
	• [in]const vdc4_EnhanceSharp *sharp	Sharpness parameter
	• [in]vdc4_O nOff lti_h_on	LTI ON/OFF control
	• [in]const vdc4_EnhanceLTI *lti	LTI parameter
	• [in]const vdc4_AreaRect *EnhArea	Enhancement area of a rectangle
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.
	VDC4_ERR_PARAM_OTHERS	Parameter error. Others.

Description

In this function, the operations are performed as below.

- Black stretch is turned on/off.
- Black stretch parameters are set.
- Sharpness is turned on/off.
- Sharpness parameters are set.
- LTI (Luminance Transient Improvement) is turned on/off.
- LTI parameters are set.
- Enhancer area is set.

In this function, each parameter and turning on/off control can be set individually. If NULL pointer to set parameters is specified, parameters for image quality improver will be left unchanged. If this function is not called, the parameters will be left unchanged since a reset. For information about initial value after a reset, see "SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048EJ)". When using sharpness or LTI, the enhancer area should be set at least once.

If this function is called in case of RGB signal input, the parameter error (VDC4_ERR_PARAM_OTHERS) will be returned.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_OnOff bkstr_on	in	Black stretch ON/OFF control • VDC4_OF F • VDC4_ON
vdc4_Black * black	in	Black stretch parameter Settings will be left unchanged if NULL is specified.
vdc4_OnOff shp_h_on	in	Sharpness ON/OFF control • VDC4_OF F • VDC4_O N
vdc4_EnhanceSharp * sharp	in	Sharpness parameter Settings will be left unchanged if NULL is specified.
vdc4_OnOff lti_h_on	in	LTI ON/OFF control • VDC4_OF F • VDC4_O N
vdc4_EnhanceLTI * lti	in	LTI parameter Settings will be left unchanged if NULL is specified.
vdc4_AreaRect * EnhArea	in	Enhancement area of a rectangle Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect. EnhArea->hs should be 4 clock cycles or greater. EnhArea->vs should be 2 lines or greater.

Members of the structure `vdc4_Black` are shown below.

```
typedef struct {
    _UWORD bkstr_st ;
    _UWORD bkstr_t1 ;
    _UWORD bkstr_t2 ;
    _UWORD bkstr_d ;
} vdc4_Black ;
```

Type Member Name	Input/ Output	Description
_UWORD bkstr_st	in	Black stretch start point 0 (low) ~ 15 (high)
_UWORD bkstr_t1	in	Black stretch time constant (T1) 0 (small) ~ 31 (large)
_UWORD bkstr_t2	in	Black stretch time constant (T2) 0 (small) ~ 30 (large)
_UWORD bkstr_d	in	Black stretch depth 0 (shallow) ~ 15 (deep)

Members of the structure vdc4_EnhanceSharp are shown below.

```
typedef struct {
    _SINT shp_h2_lpf_sel ;
    vdc4_SharpnessCtrl HrzSharp[ VDC4_IMGENH_SHARP_NUM ] ;
} vdc4_EnhanceSharp ;
```

Type Member Name	Input/ Output	Description
_SINT shp_h2_lpf_sel	in	LPF selection for folding prevention before H2 edge detection • 0: LPF not selected • 1: LPF selected
vdc4_SharpnessCtrl HrzSharp[VDC4_IMGENH_SHARP_NUM]	in	Horizontal Sharpness (H1, H2, H3) HrzSharp[VDC4_IMGENH_SHARP_H1]: H1 HrzSharp[VDC4_IMGENH_SHARP_H2]: H2 HrzSharp[VDC4_IMGENH_SHARP_H3]: H3

Members of the structure vdc4_SharpnessCtrl are shown below.

```
typedef struct {
    _UWORD shp_clip_o ;
    _UWORD shp_clip_u ;
    _UWORD shp_gain_o ;
    _UWORD shp_gain_u ;
    _UWORD shp_core ;
} vdc4_SharpnessCtrl ;
```

Type Member Name	Input/ Output	Description
_UWORD shp_clip_o	in	Sharpness correction value clipping (on the overshoot side) 0x0000 ~ 0x00FF
_UWORD shp_clip_u	in	Sharpness correction value clipping (on the undershoot side) 0x0000 ~ 0x00FF
_UWORD shp_gain_o	in	Sharpness edge amplitude value gain (on the overshoot side) 0x0000 (0 times) ~ 0x0040 (1 times) ~ 0x00FF (approx. 4times)
_UWORD shp_gain_u	in	Sharpness edge amplitude value gain (on the undershoot side) 0x0000 (0 times) ~ 0x0040 (1 times) ~ 0x00FF (approx. 4times)
_UWORD shp_core	in	Active sharpness range 0x0000 ~ 0x007F

Members of the structure vdc4_EnhanceLTI are shown below.

```
typedef struct {
    _UWORD lti_h2_inc_zero ;
    _SINT lti_h2_lpf_sel ;
    _UWORD lti_h2_gain ;
    _UWORD lti_h2_core ;
    _UWORD lti_h4_inc_zero ;
```

```
    _SINT lti_h4_median_tap_sel ;
    _UWORD lti_h4_gain ;
    _UWORD lti_h4_core ;
} vdc4_EnhanceLTI ;
```

Type Member Name	Input/ Output	Description
_UWORD lti_h2_inc_zero	in	Median filter LTI correction threshold (H2) 0x0000 ~ 0x00FF
_SINT lti_h2_lpf_sel	in	LPF sel ection for foldin g preventio n before H2 edge detection • 0: LPF not selected • 1: LPF selected
_UWORD lti_h2_gain	in	LTI edge amplitude value gain (H2) 0x0000 (0 ti mes) ~ 0x0 040 (1 time s) ~ 0x00F F (approx. 4times)
_UWORD lti_h2_core	in	LTI coring (maximum core value of 255) (H2) 0x0000 ~ 0x00FF
_UWORD lti_h4_inc_zero	in	Median filter LTI correction threshold (H4) 0x0000 ~ 0x00FF
_SINT lti_h4_median_tap_sel	in	Median filter reference pixel select • 0: Second adjacent pixel selected as reference • 1: Adjacent pixel selected as reference
_UWORD lti_h4_gain	in	LTI edge amplitude value gain (H4) 0x0000 (0 ti mes) ~ 0x0 040 (1 time s) ~ 0x00F F (approx. 4times)
_UWORD lti_h4_core	in	LTI coring (maximum core value of 255) (H4) 0x0000 ~ 0x00FF

2.3.15 VDC4_GraphicsAlphaBlending

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsAlphaBlending(vdc4_LayerID id, vdc4_OnOff gr_arc_on, const vdc4_AlphaAttr *attr) ;	
Parameters	• [in]vdc4_Lye rID id	Layer ID
	• [in]vdc4_O nOff gr_arc_on	Alpha blending in a rectangular area ON/OFF
	• [in]const vdc4_AlphaAttr *attr	Alpha blending attribute parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- Parameters for alpha blending in a rectangular area are set.
- Rectangular area for alpha blending is set.

Rectangular area for alpha blending is initialized to the same area that is specified as display area when graphics surface is created.

When alpha blending without fade-in/out in rectangular area has already turned on, alpha value can not be changed. To change alpha value, it is necessary to turn off alpha blending at least once. Before the settings are reflected, it will take a period of time equal to 1 Vsync period time at the most.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_OnOff gr_arc_on	in	Alpha blending in a rectangular area ON/OFF • VDC4_OF F • VDC4_O N
vdc4_AlphaAttr * attr	in	Alpha blending in a rectangular area attribute parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_AlphaAttr` are shown below.

```
typedef struct {
    vdc4_AreaRect *gr_arc ;
    vdc4_AlphaFade *fade ;
} vdc4_AlphaAttr ;
```

Type Member Name	Input/ Output	Description
vdc4_AreaRect * gr_arc	in	Rectangular area parameter for alpha blending Settings will be left unchanged if NULL is specified. See section 2.1.3 for the structure vdc4_AreaRect.
vdc4_AlphaFade * fade	in	Alpha blending/fading parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_AlphaFade` are shown below.

```
typedef struct {
    _UWORD gr_arc_def ;
    _SWORD gr_arc_coef ;
    _UWORD gr_arc_rate ;
} vdc4_AlphaFade ;
```

Type Member Name	Input/ Output	Description
_UWORD gr_arc_def	in	Initial alpha value for alp ha blendi ng in a rectan gular area 0 ~ 255
_SWORD gr_arc_coef	in	Alpha coefficient for alpha blendin g in a rectang ular area -255 ~ 255
_UWORD gr_arc_rate	in	Frame rate for alpha blending in a rectangular area The number of Vsync signal rising edges. Each time the Vsync sign al rise s for the numbe r of times set in the gr_arc_ rate + 1, th e value of the gr_arc_coef is added to the alpha value. 0 ~ 255

2.3.16 VDC4_GraphicsChromakey

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsChromakey(vdc4_LayerID id, vdc4_OnOff gr_ck_on, const vdc4_ChromaKeyAttr *attr) ;	
Parameters	• [in]vdc4_Lye rID id	Layer ID
	• [in]vdc4_O nOff gr_ck_on	Chroma-key processing ON/OFF
	• [in]const vdc4_ChromaKeyAttr *attr	Chroma-key attribute parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- Chroma-key parameters are set.

Chroma-key processing contains two colors: signals for chroma-key processing and replaced signals. In this driver, the color value should be specified in the same color format that target surface contains, with the exception of the following items:

- When CLUT4 or CLUT8 is used in the target surface, replaced signals should be specified in ARGB8888 format.
- When RGB565 or RGB888 is used in the target surface, replaced alpha signal should be specified separately.
- When ARGB1555 is used in the target surface, alpha signal is not supported.

In this driver, if alpha blending in a rectangular area is performed on the same graphics surface, alpha blending is performed in a specified rectangular area. In the other area, chroma-key processing is performed. And chroma-key processing is prohibited when the color format is specified as CLUT1.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_OnOff gr_ck_on	in	CLUT-index/ RGB-index chroma-key processing ON/OFF • VDC4_OF F • VDC4_O N
vdc4_ChromaKeyAttr * attr	in	Chroma-key attribute parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_ChromaKeyAttr` are shown below.

```
typedef struct {
    _UDWORD ck_color ;
    _UDWORD rep_color ;
    _UDWORD rep_alpha ;
} vdc4_ChromaKeyAttr ;
```

Type Member Name	Input/ Output	Description
_UDWORD ck_color	in	RGB/CLUT signal for chroma-key processing Specify value in the same co lor format (LSB-justified) that target surface contains. For more information about color format, see Figure 12.
_UDWORD rep_color	in	Replaced ARGB signal When the col or format that target surface contains is set to CL UT4 or CL UT8, spe cify value in ARGB8 888 format. Exce pt as n oted above, spe cify value in the same col or format (LSB-j ustified) that target su rface contains. For more info rmation ab out color fo rmat, see Figure 12.
_UDWORD rep_alpha	in	Replaced alpha signal Specify repl aced alpha signal (8bit LSB-ju stified) when sign al format in targ et surfa ce is RGB56 5 or RGB888 whi ch do es n ot have alph a value. For m ore information about color format, see Figure 12.

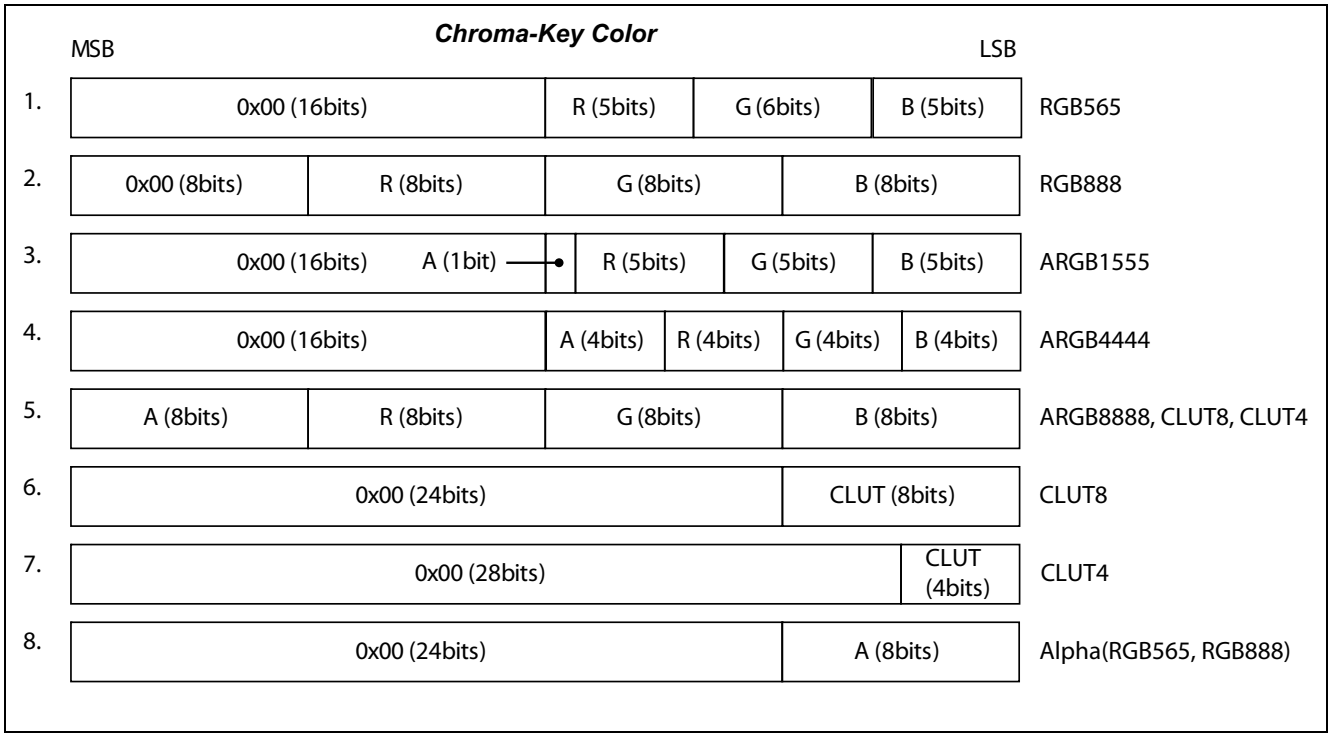


Figure 12 Chroma-Key Colors

Figure 12 shows color data format for chroma-key. When the color format is RGB565, RGB888, ARGB1555, ARGB4444 or ARGB8888, `ck_color` and `rep_color` should be specified as above (see from no.1 to no.5 in Figure 12). In addition, when the color format is RGB565 or RGB888 that have no alpha value, alpha value of replaced signal should be specified by `rep_alpha` as no.8. In case of CLUT8 and CLUT4, `ck_color` should be specified as no.6 and 7, and `rep_color` should be specified in ARGB8888 format as no.5.

2.3.17 VDC4_GraphicsSetCLUT

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsSetCLUT(vdc4_LayerID id, const vdc4_CLUT *table) ;	
Parameters	• [in]vdc4_Lye rID id	Layer ID
	• [in]const vdc4_CLUT *table	Pointer to CLUT parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_BAD	Surface error. Illegal surface.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_UNDEF	Parameter error. Undefined.
	VDC4_ERR_PARAM_OTHERS	Parameter error. Others.

Description

In this function, the operations are performed as below.

- CLUT data for the graphics surface specified by id is set.
- When CLUT format is used in the target surface, CLUT data is set.
- When ARGB1555 is used in the target surface, alpha value is set.

When using CLUT format or ARGB1555 format, this function should be called at least once.

After the graphics surface is destroyed by calling VDC4_DestroySurface, CLUT data is not guaranteed. Subsequently, when the same graphics surface is created, CLUT data should be set again by this CLUT function call.

When the graphics surface that has CLUT format is stopped displaying, CLUT data is left unchanged by calling VDC4_StopSurface. After that, the same graphics surface is displayed again by calling VDC4_GraphicsStartSurface (without calling this CLUT function), the same color can be displayed as before.

Before the CLUT data settings are reflected, it will take a period of time equal to 1 Vsync period time at the most. Note the CLUT data settings when LCD is displayed. If the function shown below is called, wait 1 Vsync period time before CLUT data settings.

- VDC4_GraphicsStartSurface
- VDC4_StopSurface
- VDC4_GraphicsSetCLUT

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_LayerID id	in	Layer ID • VDC4_SURFACE_GRAPHICS_ 1: Graphics surface 1 • VDC4_SURFACE_GRAPHICS_ 2: Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3: Graphics surface 3
vdc4_CLUT * table	in	Pointer to CLUT parameter NULL should not be specified.

Members of the structure `vdc4_CLUT` are shown below.

```
typedef struct {
    _SINT color_num ;
    _UDWORD *clut ;
} vdc4_CLUT ;
```

Type Member Name	Input/ Output	Description
_SINT color_num	in	Number of colors in CLUT. CLUT4 0 ~ 16 CLUT8 0 ~ 256 CLUT1 / ARGB1555 Ignored. 2 colors should be specified as CLUT data (clut) when CLUT1 or ARGB1555 is used.
_UDWORD * clut	in	Pointer to CLUT data in ARGB8888 format (see Figure 13) NULL should not be specified.

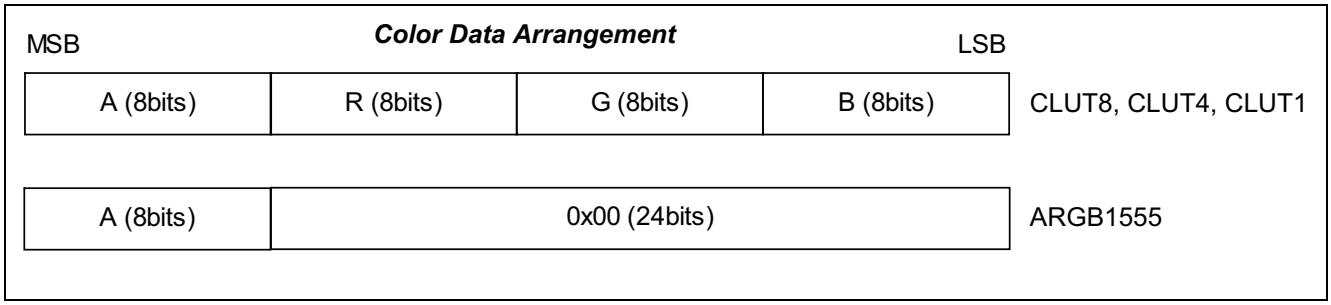


Figure 13 CLUT Data Arrangement

Figure 13 shows CLUT data arrangement. When ARGB1555 format is used, 8 bits alpha value is set in the MSB.

When CLUT1 or ARGB1555 format is used, `clut` is set to a pointer to the buffer in which two colors are stored. The first color represents a color data when CLUT1 equals '0', and the second color represents a color data when CLUT1 equals '1'. In ARGB1555 format, the first color represents a alpha value when alpha equals '0', and the second color represents a alpha value when alpha equals '1'.

2.3.18 VDC4_DisplayCalibration

Syntax
```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_DisplayCalibration( const vdc4_DispCalibration *param ) ;
```

Parameters	• [in]const vdc4_DispCalibration *param	Display Calibration parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.

Description

In this function, the operations are performed as below.

- Gamma correction is turned on/off.
- Panel brightness adjustment parameters are set.
- Contrast adjustment parameters are set.
- Panel dither correction parameters are set.
- The correction circuit sequence control is set.

In this function, it is only able to turn on/off gamma correction. To set gamma correction parameters, call VDC4_DisplaySetGammaCorrectionTable.

The panel dither output format select is set automatically depending on the output format specified in VDC4_Initialize.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_DispCalibration * param	in	Display Calibration parameter NULL should not be specified.

Members of the structure `vdc4_DispCalibration` are shown below.

```
typedef struct {
    vdc4_CalibrationPath path ;
    vdc4_OnOff gam_on ;
    vdc4_Calibr_Bright *bright ;
    vdc4_Calibr_Contrast *contrast ;
    vdc4_Calibr_Dither *panel_dither ;
} vdc4_DispCalibration ;
```

Type Member Name	Input/ Output	Description
vdc4_CalibrationPath path	in	Correction circuit sequence control • VDC4_CALIBR_B CG: Brightness -> contrast -> gamma correction • VDC4_CALIBR_GB C: Gamma correction -> brightness -> contrast
vdc4_OnOff gam_on	in	Gamma correction ON/OFF control • VDC4_OF F • VDC4_O N
vdc4_Calibr_Bright * bright	in	Brightness DC parameter Settings will be left unchanged if NULL is specified.
vdc4_Calibr_Contrast * contrast	in	Contrast gain parameter Settings will be left unchanged if NULL is specified.
vdc4_Calibr_Dither * panel_dither	in	Panel Dithering parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_Calibr_Bright` are shown below.

```
typedef struct {
    _UWORD pbrt_g ;
    _UWORD pbrt_b ;
    _UWORD pbrt_r ;
} vdc4_Calibr_Bright ;
```

Type Member Name	Input/ Output	Description
_UWORD pbrt_g	in	Brightness (DC) adjustment of G signal 0x0000 (-512) ~ 0x03FF (+511)
_UWORD pbrt_b	in	Brightness (DC) adjustment of B signal 0x0000 (-512) ~ 0x03FF (+511)
_UWORD pbrt_r	in	Brightness (DC) adjustment of R signal 0x0000 (-512) ~ 0x03FF (+511)

Members of the structure `vdc4_Calibr_Contrast` are shown below.

```
typedef struct {
    _UWORD cont_g ;
    _UWORD cont_b ;
    _UWORD cont_r ;
} vdc4_Calibr_Contrast ;
```

Type Member Name	Input/ Output	Description
_UWORD cont_g	in	Contrast (Gain) adjustment of G signal 0x0000 (0/128 [times]) ~ 0x00FF (255/128 [times])
_UWORD cont_b	in	Contrast (Gain) adjustment of B signal 0x0000 (0/128 [times]) ~ 0x00FF (255/128 [times])
_UWORD cont_r	in	Contrast (Gain) adjustment of R signal 0x0000 (0/128 [times]) ~ 0x00FF (255/128 [times])

Members of the structure `vdc4_Calibr_Contrast` are shown below.

```
typedef struct {
    vdc4_PDthMode pdth_sel ;
    _UWORD pdth_pa ;
    _UWORD pdth_pb ;
    _UWORD pdth_pc ;
    _UWORD pdth_pd ;
} vdc4_Calibr_Contrast ;
```

Type Member Name	Input/ Output	Description
vdc4_PDthMode pdth_sel	in	Panel dither operation mode • VD C4_PDTH_MD_TRU: Truncation • VDC4_PDTH_MD_RDOF: Round off • VDC4_PDTH_MD_2X2: 2 x 2 pattern dither • VD C4_PDTH_MD_RAND: Random pattern dither
_UWORD pdth_pa	in	Pattern value (A) of 2x2 Pattern dither 0 ~ 3

		This member is referenced when pdth_sel is set to VDC4_PDTH_MD_2X2.
_UWORD pdth_pb	in	Pattern value (B) of 2x2 Pattern dither 0 ~ 3 This member is referenced when pdth_sel is set to VDC4_PDTH_MD_2X2.
_UWORD pdth_pc	in	Pattern value (C) of 2x2 Pattern dither 0 ~ 3 This member is referenced when pdth_sel is set to VDC4_PDTH_MD_2X2.
_UWORD pdth_pd	in	Pattern value (D) of 2x2 Pattern dither 0 ~ 3 This member is referenced when pdth_sel is set to VDC4_PDTH_MD_2X2.

2.3.19 VDC4_DisplaySetGammaCorrectionTable

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_DisplaySetGammaCorrectionTable(const vdc4_GammaCorrection *table) ;	
Parameters	• [in]const vdc4_GammaCorrection * table	Gamma correction parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.

Description

In this function, the operations are performed as below.

- The gamma correction parameters are individually set for RGB signals.

In this function, the parameters referred to by gamma correction process are set. To turn on/off gamma correction, it is necessary to call VDC4_DisplayCalibration.

If this function is not called, the parameters for gamma correction will be left unchanged since a reset. For information about initial value after a reset, see "SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048EJ)".

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_GammaCorrection * table	in	Gamma correction parameter NULL should not be specified.

Members of the structure `vdc4_GammaCorrection` are shown below.

```
typedef struct {
    vdc4_GamCrrct *gam_g ;
    vdc4_GamCrrct *gam_b ;
    vdc4_GamCrrct *gam_r ;
} vdc4_GammaCorrection ;
```

Type Member Name	Input/ Output	Description
vdc4_GamCrrct * gam_g	in	Gamma correction parameter of G signal Settings will be left unchanged if NULL is specified.
vdc4_GamCrrct * gam_b	in	Gamma correction parameter of B signal Settings will be left unchanged if NULL is specified.
vdc4_GamCrrct * gam_r	in	Gamma correction parameter of R signal Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_GamCrrct` are shown below.

```
typedef struct {
    _UWORD gam_th[ VDC4_GAM_TH_NUM ] ;
    _UWORD gam_gain[ VDC4_GAM_GAIN_NUM ] ;
} vdc4_GamCrrct ;
```

Type Member Name	Input/ Output	Description
_UWORD gam_th[VDC4_GAM_TH_NUM]	in	Start threshold of area 1 to 31 of signal 0 ~ 255 The symbol VDC4_GAM_TH_NUM is defined as 31.
_UWORD gam_gain[VDC4_GAM_GAIN_NUM]	in	Gain adjustment of area 0 to 31 of signal 0x0000 ~ 0x07FF (0x0400 =1.0[times]) The symbol VDC4_GAM _GAIN_NUM is defined as 32.

2.3.20 VDC4_SwitchVsync

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_SwitchVsync(vdc4_OnOff FreeRunVsync, const vdc4_SyncPeriod *SyncPeriod) ;	
Parameters	• [in]vdc4_O nOff FreeRunVsync	Free-running Vsync ON/OFF
	• [in]con st vdc4_SyncPeriod * SyncPeriod	Sync period parameter
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_RANGE	Parameter error. Out of range.

Description

In this function, the operations are performed as below.

- The Vsync signal to be output from the scaler is switched.
- The sync period parameters are set.

In this function, the Vsync signal to be output from the scaler can be selected and the sync period parameter can be set. These settings are performed as part of the initialization in the function VDC4_Initialize (see 2.3.1), therefore it is not always necessary.

Even though the displayed image may be distorted by calling this function, it is not considered in this VDC4 driver. Users have to consider when the Vsync should be switched by calling this function.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_OnOff FreeRunVsync	in	Free-running Vsync ON/OFF • VDC4_OFF: External input Vsync signal • VDC4_ON: Internally generated free-running Vsync signal
vdc4_SyncPeriod * SyncPeriod	in	Sync period parameter Settings will be left unchanged if NULL is specified.

Members of the structure `vdc4_SyncPeriod` are shown below.

```
typedef struct {
    _UWORD res_fv ;
    _UWORD res_fh ;
} vdc4_SyncPeriod ;
```

Type Member Name	Input/ Output	Description
_UWORD res_fv	in	Free-running Vsync period setting Vsync period = (res_fv + 1) x Hsync period 0x0000 ~ 0x07FF [lines] This parameter is ignored, when Free-running Vsync (FreeRunVsync) is set to VDC4_OFF.
_UWORD res_fh	in	Hsync period setting Hsync period = (res_fh + 1) / pixel clock frequency 0x0000 ~ 0x07FF [pixel clock cycles]

2.3.21 VDC4_CheckClock

Syntax

```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_CheckClock( vdc4_PanelClockSel PanelClockSel,
                                vdc4_PanelClkDCDR Dcdr,
                                _UDWORD SrcClkFreq,
                                void (*WaitMsec)( _UDWORD )) ;
```

Parameters	• [in]vdc4_PanelClockSel PanelClockSel	Panel clock source select
	• [in]vdc4_Pa nelClkDCDR Dcdr	Panel clock frequency division ratio
	• [in]_UDWO RD SrcClkFreq	Source clock frequency
	• [in]void (*WaitMsec)(_UDWORD)	Pointer to the user-defined wait function
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.
	VDC4_ERR_PARAM_UNDEF	Parameter error. Undefined.
	VDC4_ERR_PARAM_INVALID	Parameter error. Invalid parameter.
	VDC4_ERR_SYSTEM_PNLCLK	System error. Abnormal panel clock.

Description

In this function, the operations are performed as below.

- The panel clock source select and the panel clock frequency division ratio are set.
- Free-running sync periods are set.
- Masking repeated Vsync signals and compensating for missing Vsync signals are disabled.
- TCON register update control register (TCON_UPDATE) is set.
- The wait time is calculated from free-running sync periods and source clock frequency, and wait processing is performed by user-defined wait function.
- After wait processing, the abnormal operation of the panel clock is detected by checking TCON_UPDATE.

This function is used to check that the panel clock in VDC4 is normal or not. For checking the panel clock and retrying when abnormal operation is detected, see section 4.2.2. The abnormal operation of the panel clock does not occur on the following conditions.

Table 16 Conditions of Normal Operation

Panel Clock Frequency Division Ratio	VIDEO_X1 Pin	Panel Clock Source Select
1/1, 1/2, 1/5, 1/9	All	All
1/7	Fixed	Video image clock (VIDEO_X1)
		Peripheral clock 1
Others	Clock Input	Video image clock (VIDEO_X1)
	Fixed	Video image clock (VIDEO_X1)
		Peripheral clock 1

This function should be called first after reset. Therefore, clock and I/O port settings as follows should be done before calling this function. However, I/O port settings for the sync signal of the LCD module should not be done. To check the panel clock operation, free-running sync signal settings are done in this function. If I/O port related to sync signal is set as output pin, illegal signal may be output to the LCD module.

- Supply the clock to the VDC4 module.
- Supply the clock to the Digital Video Decoder module, if necessary.
- Set I/O port for the external clock (LCD_EXTCLK) or Video image clock (DV_CLK), if necessary.

To check panel clock operation by this function, the wait processing is required. The wait processing is performed by the user-defined function specified by `WaitMsec`. About the implementation of user-defined function, see section 3.3.

Arguments Settings

Type Parameter Name	Input / Output	Description
vdc4_PanelClockSel PanelClockSel	in	Panel clock source select • VDC4_L CDPANEL_CLKSEL_IMG: Video image clock (VIDEO_X1) • VDC4_LCDPANEL_CLKS EL_EXT: External clock (LCDEXT_CLK) • VDC4_LCDPANEL _CLKSEL_PERI: Peripheral clock 1 • VDC4_LCDPANEL_CLKSEL_IMG_DV: Video image clock (DV_CLK)
vdc4_PanelClkDCDR Dcdr	in	Panel clock frequency division ratio • VDC4_L CDPANEL_CLKDIV_1_1: 1/1 • VDC4_L CDPANEL_CLKDIV_1_2: 1/2 • VDC4_L CDPANEL_CLKDIV_1_3: 1/3 • VDC4_L CDPANEL_CLKDIV_1_4: 1/4 • VDC4_L CDPANEL_CLKDIV_1_5: 1/5 • VDC4_L CDPANEL_CLKDIV_1_6: 1/6 • VDC4_L CDPANEL_CLKDIV_1_7: 1/7 • VDC4_L CDPANEL_CLKDIV_1_8: 1/8 • VDC4_L CDPANEL_CLKDIV_1_9: 1/9 • VDC4_L CDPANEL_CLKDIV_1_12: 1/12 • VDC4_L CDPANEL_CLKDIV_1_16: 1/16 • VDC4_L CDPANEL_CLKDIV_1_24: 1/24 • VDC4_L CDPANEL_CLKDIV_1_32: 1/32
_UDWORD SrcClkFreq	in	Source clock frequency [Hz] e.g., if source clock frequency is 40 MHz, set '40000000'.
void (*WaitMsec)(_UDWORD)	in	Pointer to the user-defined wait function Pointer to the wait function implemented by the user. This function will delay until an amount of time specified by the parameter (wait [milliseconds]) has elapsed. '0' should not be specified. Syntax: void Wait_Msec(_UDWORD wait) ; Parameters: • [in]_UDWO RD wait — Wait time [msec] Return Values: • void Description Wait processing implemented by the user is performed.

2.3.22 VDC4_Reset

Syntax	#include "vdc4_api.h" vdc4_ErrorCode VDC4_Reset(void) ;	
Parameters	• [in]void	
Return Values	• vdc4_E rrorCode	Error code
	VDC4_ERR_NONE	Normal end.
	VDC4_ERR_SURFACE_STATUS	Surface error. Illegal state.

Description

In this function, the operations are performed as below.

- Software reset of VDC4 module is performed.
- Software reset of VDC4 module is canceled.

After this function is called, VDC4 module will be the same state immediately after reset.

3. User-defined Functions

Driver calls VDC4_Initialize and VDC4_Terminate allow execution of user-defined functions. And driver call VDC4_CheckClock requires the user-defined wait function. Examples of the user-defined function are shown below.

3.1 Example of User-defined Function within VDC4_Initialize

```
/*****************************************************************************//**
 * Function Name : Init_VDC4_CallBack
 * @brief
 * @param      [in]_UDWORD mode
 * @retval     void
 ******************************************************************************/
void Init_VDC4_CallBack( _UDWORD mode )
{
    /* VDC4 Interrupt Level
           b11:b8  ----0101 -------- ; VI_VSYNC, LO_VSYNC, VSYNCERR
           b7:b4   -------- 0101---- ; VLINE, VFIELD
           b3:b0   -------- ----0101 ; VBUFERR1, 2, 3, 4 */
       INTC.IPR10.WORD &= ~0x0FFFu ;
       INTC.IPR10.WORD |= 0x0555u ;

       /* standby control register 7 (STBCR7)
           b3      ----0--- ; MSTP73 : 0 : VDC4 enable */
       CPG.STBCR7.BYTE &= ~0x0008u ;

       /* port G control register 0
           b14:b12 -010---- -------- ; PG3MD[2:0] : 010 : LCD_DATA3
           b10:b8  -----010 -------- ; PG2MD[2:0] : 010 : LCD_DATA2
           b6:b4   -------- -010---- ; PG1MD[2:0] : 010 : LCD_DATA1
           b2:b0   -------- -----010 ; PG0MD[2:0] : 010 : LCD_DATA0 */
       PORT.PGCR0.WORD = 0x2222u ;
       /* port G control register 1
           b14:b12 -010---- -------- ; PG7MD[2:0] : 010 : LCD_DATA7
           b10:b8  -----010 -------- ; PG6MD[2:0] : 010 : LCD_DATA6
           b6:b4   -------- -010---- ; PG5MD[2:0] : 010 : LCD_DATA5
           b2:b0   -------- -----010 ; PG4MD[2:0] : 010 : LCD_DATA4 */
       PORT.PGCR1.WORD = 0x2222u ;
       /* port G control register 2
           b14:b12 -010---- -------- ; PG11MD[2:0] : 010 : LCD_DATA11
           b10:b8  -----010 -------- ; PG10MD[2:0] : 010 : LCD_DATA10
           b6:b4   -------- -010---- ; PG9MD[2:0]  : 010 : LCD_DATA9
           b2:b0   -------- -----010 ; PG8MD[2:0]  : 010 : LCD_DATA8 */
       PORT.PGCR2.WORD = 0x2222u ;
       /* port G control register 3
           b13:b12 --10---- -------- ; PG15MD[1:0] : 10 : LCD_DATA15
           b9:b8   ------10 -------- ; PG14MD[1:0] : 10 : LCD_DATA14
           b5:b4   -------- --10---- ; PG13MD[1:0] : 10 : LCD_DATA13
           b1:b0   -------- ------10 ; PG12MD[1:0] : 10 : LCD_DATA12 */
       PORT.PGCR3.WORD = 0x2222u ;
       /* port G control register 4
           b14:b12 -010---- -------- ; PG19MD[2:0] : 010 : LCD_DATA19
           b10:b8  -----010 -------- ; PG18MD[2:0] : 010 : LCD_DATA18
           b5:b4   -------- --10---- ; PG17MD[1:0] : 10 : LCD_DATA17
           b1:b0   -------- ------10 ; PG16MD[1:0] : 10 : LCD_DATA16 */
       PORT.PGCR4.WORD = 0x2222u ;
```

```
50        /* port G control register 5
51            b14:b12 -010---- -------- ; PG23MD[2:0] : 010 : LCD_DATA23
52            b10:b8  -----010 -------- ; PG22MD[2:0] : 010 : LCD_DATA22
53            b6:b4   -------- -010---- ; PG21MD[2:0] : 010 : LCD_DATA21
54            b2:b0   -------- -----010 ; PG20MD[2:0] : 010 : LCD_DATA20 */
55        PORT.PGCR5.WORD = 0x2222u ;
56        /* port G control register 6
57            b13:b12 --11---- -------- ; PG27MD[1:0] : 11 : LCD_EXTCLK
58            b9:b8   ------10 -------- ; PG26MD[1:0] : 10 LCD_TCON1
59            b5:b4   -------- --10---- ; PG25MD[1:0] : 10 LCD_TCON0
60            b1:b0   -------- ------10 ; PG24MD[1:0] : 10 LCD_CLK */
61        PORT.PGCR6.WORD = 0x3222u ;
62        /* port J control register 5
63            b14:b12 -011---- -------- ; PJ23MD[2:0] : 011 : LCD_TCON6
64            b10:b8  -----011 -------- ; PJ22MD[2:0] : 011 : LCD_TCON5
65        */
66        PORT.PJCR5.WORD &= ~0x7700u ;
67        PORT.PJCR5.WORD |= 0x3300u ;
68
69        /* port J control register 7
70            b12     ---1---- -------- ; PJ31MD : 1 : DV_CLK
71        */
72        PORT.PJCR7.WORD |= 0x1000u ;
73        /* port E control register 1
74            b5:b4   -------- --11---- ; PE5MD[1:0] : 11 : DV_HSYNC
75            b1:b0   -------- ------11 ; PE4MD[1:0] : 11 : DV_VSYNC */
76        PORT.PECR1.WORD |= 0x0033u ;
77        /* port J control register 0
78            b14:b12 -001---- -------- ; PJ3MD[2:0] : 001 : DV_DATA3
79            b10:b8  -----001 -------- ; PJ2MD[2:0] : 001 : DV_DATA2
80            b6:b4   -------- -001---- ; PJ1MD[2:0] : 001 : DV_DATA1
81            b2:b0   -------- -----001 ; PJ0MD[2:0] : 001 : DV_DATA0
82        */
83        PORT.PJCR0.WORD &= ~0x7777u ;
84        PORT.PJCR0.WORD |= 0x1111u ;
85        /* port J control register 1
86            b14:b12 -001---- -------- ; PJ7MD[2:0] : 001 : DV_DATA7
87            b10:b8  -----001 -------- ; PJ6MD[2:0] : 001 : DV_DATA6
88            b6:b4   -------- -001---- ; PJ5MD[2:0] : 001 : DV_DATA5
89            b2:b0   -------- -----001 ; PJ4MD[2:0] : 001 : DV_DATA4
90        */
91        PORT.PJCR1.WORD &= ~0x7777u ;
92        PORT.PJCR1.WORD |= 0x1111u ;
93    }
```

3.2 Example of User-defined Function within VDC4_Terminate

```
/*****************************************************************//**
 * Function Name : Quit_VDC4_CallBack
 * @brief
 * @param      [in]_UDWORD mode
 * @retval     void
 *****************************************************************/
void Quit_VDC4_CallBack( _UDWORD mode )
{
    /* VDC4 Interrupt Level
        b11:b8  ----0000 -------- ; VI_VSYNC, LO_VSYNC, VSYNCERR
        b7:b4   -------- 0000---- ; VLINE, VFIELD
        b3:b0   -------- ----0000 ; VBUFERR1, 2, 3, 4 */
    INTC.IPR10.WORD &= ~0x0FFFu ;

    /* standby control register 7 (STBCR7)
        b3              ----1--- ; MSTP73 : VDC4 disable */
    CPG.STBCR7.BYTE |= 0x0008u ;
}
```

3.3 Example of User-defined Function within VDC4_CheckClock

When using VDC4_CheckClock, the user-defined wait function is required. Two sample implementation are shown below.

3.3.1 Compare Match Timer

```
/*****************************************************************//**
 * Function Name : Wait_Msec
 * @brief       Compare Match Timer (1ch) to Wait a msec
 * @n                   Clock select
 * @n                   Peripheral 0 clock 33.33MHz / 512
 * @param       [in]_UDWORD wait    : Wait time in msec
 * @retval      void
 *****************************************************************/
void Wait_Msec( _UDWORD wait )
{
    _UDWORD Compare ;
    _UWORD Reg ;
    _UDWORD PushReg ;
    volatile _UBYTE DummyRead ;

    PushReg = ( _UDWORD )CPG.STBCR7.BYTE & 0x0004ul ;
    /* Standby control register 7 (STBCR7)
        b3      -----0-- ; MSTP72 : 0 : Compare match timer enable */
    CPG.STBCR7.BYTE &= ( _UBYTE )( ~0x0004u ) ;
    /* Dummy read */
    DummyRead = CPG.STBCR7.BYTE ;

    /* Compare Match Timer Control/Status Register
       b7      ---- ---- 0--- ---- CMF: Compare Match Flag; clearing
       b6      ---- ---- -0-- ---- CMIE: Compare Match Interrupt Enable;
disabled
       b1:b0   ---- ---- ---- --11 CKS: Clock Select, P0/512
     */
    CMT.CMCSR1.WORD = ( _UWORD )0x0003u ;

    /* 1000usec / 15.36usec = 65.1 */
    Compare = wait * 66ul ;
    if( Compare >= 0x00010000ul )
    {
        Compare = 0x0000FFFFul ;
    }
    /* Compare Match Constant Register */
    CMT.CMCOR1.WORD = ( _UWORD )Compare ;
    /* Compare Match Counter */
    CMT.CMCNT1.WORD = 0 ;
    /* Compare Match Timer Start Register
       b1      ---- ---- ---- --1- STR1: Count Start 1; started
     */
    CMT.CMSTR.WORD |= ( _UWORD )0x0002u ;

    do
    {
        Reg = ( _UWORD )( CMT.CMCSR1.WORD & 0x0080u ) ;
    }
```

```
49          while( Reg == 0 ) ;
50          /* Compare Match Timer Start Register
51             b1      ---- ---- ---- --0- STR1: Count Start 1; stopped
52           */
53          CMT.CMSTR.WORD &= ( _UWORD )~0x0002u ;
54
55          /* Compare Match Timer Control/Status Register
56             b7      ---- ---- 0--- ---- CMF: Compare Match Flag; clearing
57           */
58          CMT.CMCSR1.WORD &= ( _UWORD )~0x0080u ;
59
60          if( PushReg != 0 )
61          {   /* Standby control register 7 (STBCR7)
62                  b3      -----1-- ; MSTP72 : 1 : Compare match timer disable
*/
63              CPG.STBCR7.BYTE |= ( _UBYTE )0x0004u ;
64          }
65      }
```

In this sample code, 33.33MHz peripheral clock 0 is used as a clock source for the Compare Match Timer module.

3.3.2 Busy-waiting

```
1       /******************************************************************//**
2        * Function Name : Wait_Msec
3       * @brief       Delya loop to Wait a msec
4       * @param       [in]_UDWORD wait    : Wait time in msec
5        * @retval      void
6        *******************************************************************/
7       void Wait_Msec( _UDWORD wait )
8       {
9           _UDWORD i, j ;
10          volatile _UDWORD count ;
11
12          for( i = 0 ; i < wait ; i++ )
13          {
14              count = 0 ;
15              /* The value '53368' is a number to wait a msec.
16                 This value is depends on the environment (i.e., memory
mapping,
17                 memory caching, optimization, and etc.). */
18              for( j = 0 ; j < 53368ul ; j++ )
19              {
20                  count++ ;
21              }
22          }
23      }
```

The number of iterations for busy-waiting depends on the environment (e.g., memory mapping, memory caching, and optimization) and processing (e.g., interrupt service routine and dispatching task/thread). If the OS is available, it is recommended that the wait/delay function provided by the OS should be used.

4. Example Usage

4.1 The Implementation of Double Buffering

This driver program provides two ways of double buffering.

4.1.1 Double Buffering provided by VDC4

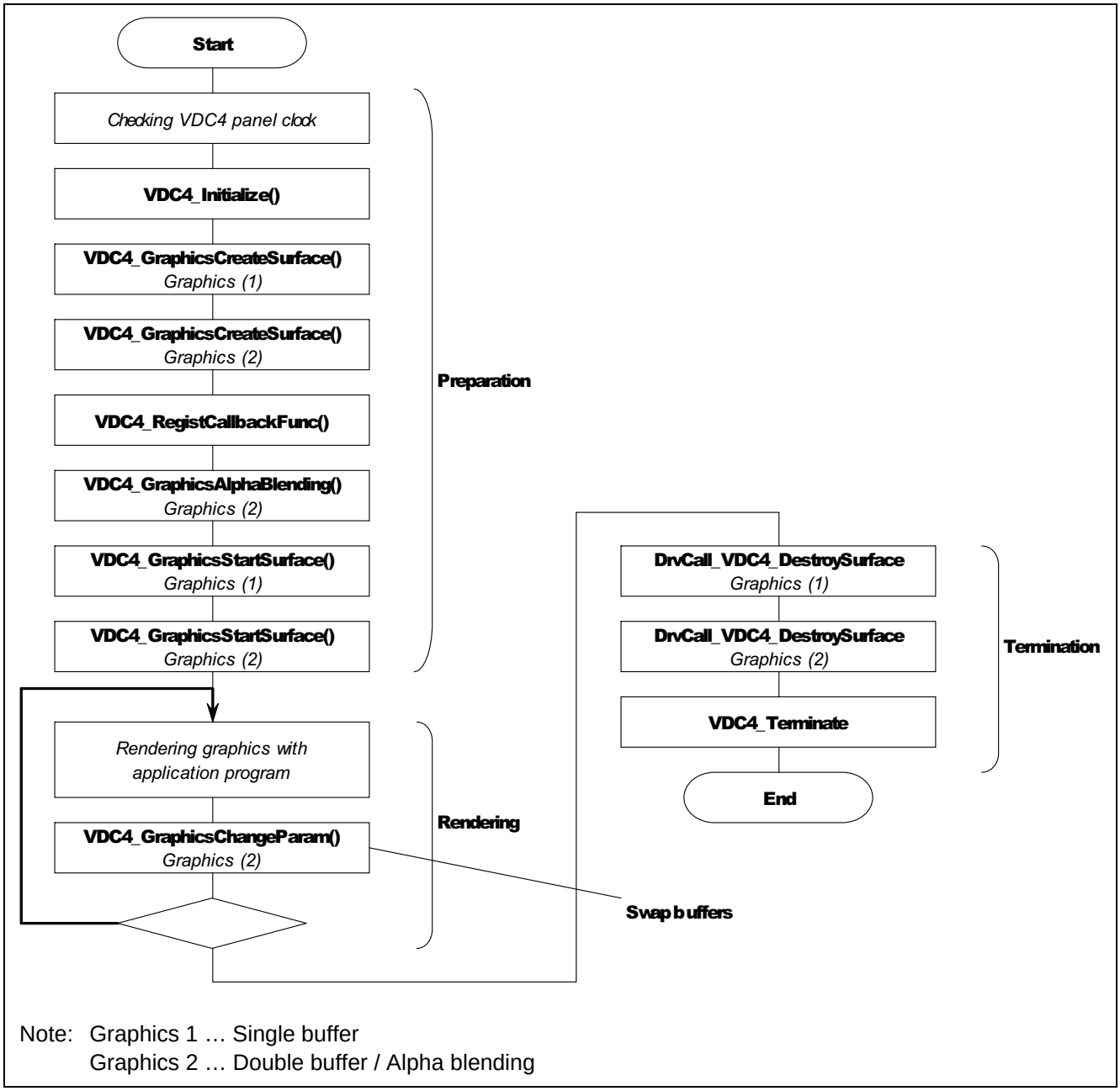
If the second frame buffer is specified, double buffering is performed in VDC4 driver. When drawing is started or frame buffer is swapped, displayed frame is specified by frame number.

4.1.2 Double Buffering in User Applications

In this case, only frame buffer base address is specified and single buffering is performed in VDC4 driver. Frame buffer base address is changed with calling VDC4_GraphicsChangeParam and specified by frame buffer address.

4.2 Example of Processing Flow

4.2.1 Graphics 1 and Graphics 2



Note: Graphics 1 … Single buffer
Graphics 2 … Double buffer / Alpha blending

Figure 14 Example of Processing Flow (Graphics 1 and Graphics 2)

About Checking VDC4 panel clock (the first process in Figure 14), see section 4.2.2.

4.2.2 Detection of the Abnormal Panel Clock and Retrying

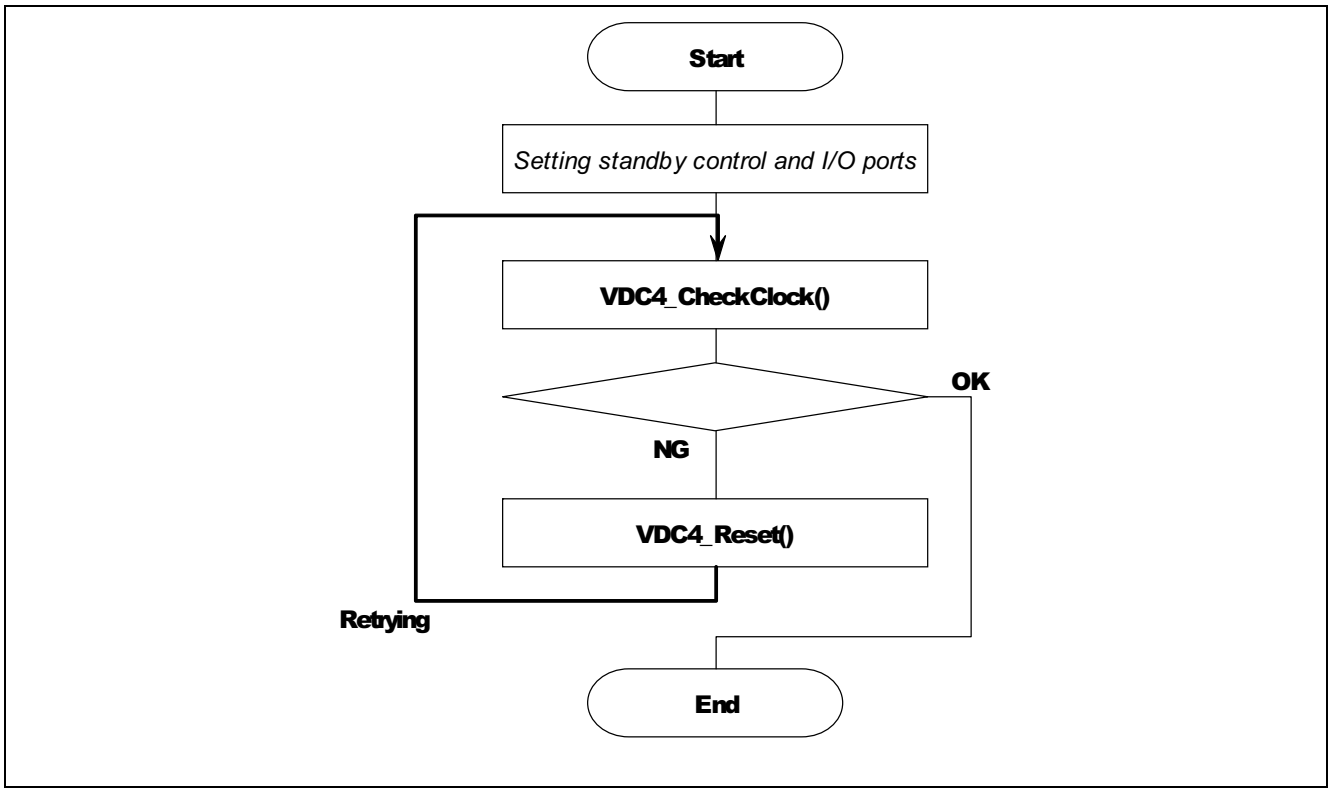


Figure 15 Example of Processing Flow for the Detection of the Abnormal Panel Clock

At first, the supply of VDC4 clock and I/O port settings are performed. Then, the panel clock is checked by calling VDC4_CheckClock. If VDC4_CheckClock returns VDC4_ERR_NONE, it is end because the panel clock operation is normal. If VDC4_CheckClock returns VDC4_ERR_SYSTEM_PNLCLK, the retry processing should be done because the panel clock operation is abnormal. In the retry processing, the panel clock is checked by calling VDC4_CheckClock, after VDC4_Reset is called to reset VDC4.

4.3 Setting Example for LCD

LCD setting processing is performed by the API function VDC4_Initialize. The relations between the specification of the LCD and the setting values of a function VDC4_Initialize are explained below.

4.3.1 Example of Settings for the LCD that requires Hsync and Vsync

An example of settings for the LCD that requires the horizontal sync (Hsync) signal and the vertical sync (Vsync) signal is as shown below.

Table 17 VDC4 Settings for the LCD

Output pins for the signals to drive the LCD panel	Hsync: LCD_TCON1 Vsync: LCD_TCON0
LCD TCON	STVA/VS: STVA signal is used as Vsync. STH/SP/HS: STH signal is used as Hsync.
The source of the panel clock	Peripheral clock 1 (66.67MHz)
Vertical sync signal output select	Internally generated free-running Vsync signal
Bit allocation of LCD signals	R[7:3]-G[7:2]-B[7:3]

Note: In addition to VDC4, a setup of I/O ports is required to use output pins LCD_TCON1 and LCD_TCON0.

Table 18 LCD Specifications

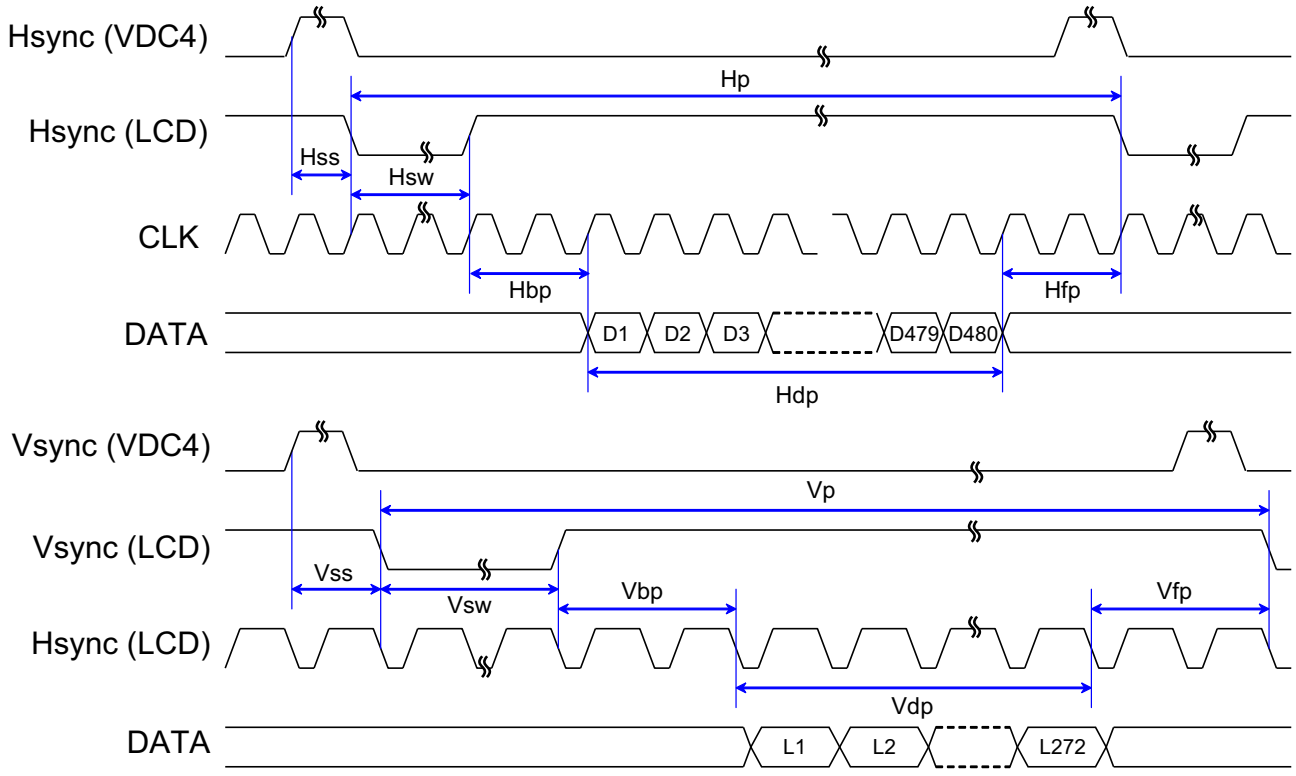
Display size	480 x 272
Output format	RGB565
Data sampling	The falling edge of the panel clock
Sync signal polarity	Hsync: Negative polarity Vsync: Negative polarity
Frequency	8.0 ~ 9.0 MHz

Table 19 Timing Signals for Driving the LCD Panel

Signal Sy	mbol	Value
Horizontal period	Hp	525 [clock]
Horizontal display period	Hdp	480 [clock]
Horizontal sync signal start position	Hss	0 [clock]
Horizontal sync signal pulse width	Hsw	41 [clock]
Horizontal back porch	Hbp	2 [clock]
Horizontal front porch	Hfp	2 [clock]
Vertical period	Vp	286 [line]
Vertical display period	Vdp	272 [line]
Vertical sync signal start position	Vss	0 [line]
Vertical sync signal pulse width	Vsw	10 [line]
Vertical back porch	Vbp	2 [line]
Vertical front porch	Vfp	2 [line]

Note: Except for Hss and Vss, setting values are complied with LCD specification. Setting values of Hss and Vss define the timing of the LCD sync signal from the VDC4 internal sync signal.

The control timing for driving the LCD panel is as shown below (see Figure 16).



Hsync (VDC4): VDC4 internal horizontal sync signal
Vsync (VDC4): VDC4 internal vertical sync signal
Hsync (LCD): LCD horizontal sync signal
Vsync (LCD): LCD vertical sync signal
CLK: Panel clock (pixel clock)
DATA: Data
D1 ~ D480 … data per pixel
L1 ~ L272 … data per line

Figure 16 Signal Changing Timing of the LCD driven by Hsync and Vsync

The LCD settings are set to the structure `vdc4_InitAttr` as follows when VDC4_Initialize is called.

```
1     vdc4_InitAttr InitAttr ;
2     vdc4_LcdTconTim LcdTconTim_STVA_VS ;
3     vdc4_LcdTconTim LcdTconTim_STH_SP_HS ;
4     vdc4_LcdOutput LcdOutput ;
5
6     /* Vsync */
7     LcdTconTim_STVA_VS.tcon_hsvs = ( _UWORD )( 0 * 2 ) ;
8     LcdTconTim_STVA_VS.tcon_hwvw = ( _UWORD )( 10 * 2 ) ;
9     LcdTconTim_STVA_VS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
10    LcdTconTim_STVA_VS.tcon_hs_sel = ( _UWORD )0 ;
11    LcdTconTim_STVA_VS.tcon_inv = ( _UWORD )1 ;
12    LcdTconTim_STVA_VS.tcon_pin = VDC4_LCD_TCON_PIN_0 ;
13    LcdTconTim_STVA_VS.outcnt_edge = VDC4_EDGE_RISING ;
14    /* Hsync */
15    LcdTconTim_STH_SP_HS.tcon_hsvs = ( _UWORD )0 ;
16    LcdTconTim_STH_SP_HS.tcon_hwvw = ( _UWORD )41 ;
17    LcdTconTim_STH_SP_HS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
18    LcdTconTim_STH_SP_HS.tcon_hs_sel = ( _UWORD )0 ;
19    LcdTconTim_STH_SP_HS.tcon_inv = ( _UWORD )1 ;
20    LcdTconTim_STH_SP_HS.tcon_pin = VDC4_LCD_TCON_PIN_1 ;
21    LcdTconTim_STH_SP_HS.outcnt_edge = VDC4_EDGE_RISING ;
22    /* LCD Output Control */
23    LcdOutput.res_f.hs = ( _UWORD )43 ;
24    LcdOutput.res_f.hw = ( _UWORD )480 ;
25    LcdOutput.res_f.vs = ( _UWORD )12 ;
26    LcdOutput.res_f.vw = ( _UWORD )272 ;
27    LcdOutput.out_endian_on = VDC4_OFF ;
28    LcdOutput.out_swap_on = VDC4_OFF ;
29    LcdOutput.out_format = VDC4_LCD_OUTFORMAT_RGB565 ;
30    LcdOutput.out_frq_sel = VDC4_LCD_PARALLEL_CLKFRQ_1 ;
31    LcdOutput.out_dir_sel = ( _UWORD )0 ;
32    LcdOutput.out_phase = VDC4_LCD_SERIAL_CLKPHASE_0 ;
33
34    /* Initialization parameter */
35    InitAttr.Vsel.FreeRunVsync = VDC4_ON ;
36    InitAttr.Vsel.res_fv = ( _UWORD )( 286 - 1 ) ;
37    InitAttr.Vsel.res_fh = ( _UWORD )( 525 - 1 ) ;
38    InitAttr.sync = NULL ;
39    InitAttr.panel_icksel = VDC4_LCDPANEL_CLKSEL_PERI ;
40    InitAttr.panel_dcdr = VDC4_LCDPANEL_CLKDIV_1_8 ;
41    InitAttr.tcon_half = ( _UWORD )262 ;
42    InitAttr.tcon_offset = ( _UWORD )0 ;
43    InitAttr.outcnt_lcd_edge = VDC4_EDGE_RISING ;
44    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVA_VS ] = &LcdTconTim_STVA_VS ;
45    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVB_VE ] = NULL ;
46    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STH_SP_HS ] = &LcdTconTim_STH_SP_HS ;
47    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STB_LP_HE ] = NULL ;
48    InitAttr.outctrl[ VDC4_LCD_TCONSIG_CPV_GCK ] = NULL ;
49    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLA ] = NULL ;
50    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLB ] = NULL ;
51    InitAttr.outctrl[ VDC4_LCD_TCONSIG_DE ] = NULL ;
52    InitAttr.settings = &LcdOutput ;
53    InitAttr.IntEnable_1 = VDC4_INT_BIT_VLINE ;
54    InitAttr.IntEnable_2 = VDC4_INT_BIT_NONE ;
```

The above sample code is explained as follows.

1. Setting Vsync
   STVA/VS signal generated by LCD TCON is used for the Vsync signal. The timing parameters of the Vsync signal are set in the structure `vdc4_LcdTconTim` (line 7 - 13). And a pointer to this structure `vdc4_LcdTconTim` is set in the array member `outctrl` (referred to by an index `VDC4_LCD_TCONSIG_STVA_VS`) in the structure `vdc4_InitAttr` as STVA/VS signal settings (line 44).
   — Setting Structure `vdc4_LcdTconTim`
   - `tcon_hsvs`: The vertical sync signal start position (Vss in Table 19) is specified in 1/2 cycles (line 7).
   - `tcon_hwvw`: The vertical sync signal pulse width (Vsw in Table 19) is specified in 1/2 cycles (line 8).
   - `tcon_md`: In STVA/VS signal settings, this parameter is ignored.
   - `tcon_hs_sel`: In STVA/VS signal settings, this parameter is ignored.
   - `tcon_inv`: The vertical sync signal is negative polarity (in Table 18), therefore this parameter is set to '1' (line 11).
   - `tcon_pin`: The vertical sync signal output pin is LCD_TCON0 (in Table 17), therefore this parameter is set to `VDC4_LCD_TCON_PIN_0` (line 12).
   - `outcnt_edge`: Rising edge `VDC4_EDGE_RISING` is specified to synchronize with the LCD data signal (line 13).
2. Setting Hsync
   STH/SP/HS signal generated by LCD TCON is used for the Hsync signal. The timing parameters of the Hsync signal are set in the structure `vdc4_LcdTconTim` (line 15 - 21). And a pointer to this structure `vdc4_LcdTconTim` is set in the array member `outctrl` (referred to by an index `VDC4_LCD_TCONSIG_STH_SP_HS`) in the structure `vdc4_InitAttr` as STH/SP/HS signal settings (line 46).
   — Setting Structure `vdc4_LcdTconTim`
   - `tcon_hsvs`: The horizontal sync signal start position (Hss in Table 19) is specified (line 15).
   - `tcon_hwvw`: The horizontal sync signal pulse width (Hsw in Table 19) is specified (line 16).
   - `tcon_md`: In STH/SP/HS signal settings, this parameter is ignored.
   - `tcon_hs_sel`: The horizontal reference offset control is not used, therefore this parameter is set to '0' (line 18).
   - `tcon_inv`: The horizontal sync signal is negative polarity (in Table 18), therefore this parameter is set to '1' (line 19).
   - `tcon_pin`: The horizontal sync signal output pin is LCD_TCON1 (in Table 17), therefore this parameter is set to `VDC4_LCD_TCON_PIN_1` (line 20).
   - `outcnt_edge`: Rising edge `VDC4_EDGE_RISING` is specified to synchronize with the LCD data signal (line 21).
3. Setting LCD Output
   The LCD output parameters are set in the structure `vdc4_LcdOutput` (line 23 - 32). And a pointer to this structure `vdc4_LcdOutput` is set in the member `settings` in the structure `vdc4_InitAttr` (line 52).
   — Setting Structure `vdc4_LcdOutput`
   - `res_f.hs`: The horizontal enable signal start position (Hss + Hsw + Hbp in Table 19) is specified (line 23).
   - `res_f.hw`: The horizontal enable signal width (Hdp in Table 19) is specified (line 24).
   - `res_f.vs`: The vertical enable signal start position (Vss + Vsw + Vbp in Table 19) is specified (line 25).
   - `res_f.vw`: The vertical enable signal width (Vdp in Table 19) is specified (line 26).
   - `out_endian_on`: The LCD data signal is output MSB first (in Table 17), therefore this parameter is set to `VDC4_OFF` (line 27).
   - `out_swap_on`: The LCD data signal is output in order RGB (in Table 17), therefore this parameter is set to `VDC4_OFF` (line 28).
   - `out_format`: The LCD output format is RGB565 (in Table 18), therefore this parameter is set to `VDC4_LCD_OUTFORMAT_RGB565` (line 29).
   - `out_frq_sel`: When the LCD output format is not serial RGB, this parameter is ignored.
   - `out_dir_sel`: When the LCD output format is not serial RGB, this parameter is ignored.
   - `out_phase`: When the LCD output format is not serial RGB, this parameter is ignored.
4. Setting Others
   Other settings for the LCD are set in the structure `vdc4_InitAttr` (line 35 - 43).

— Setting Structure `vdc4_InitAttr`

- `Vsel.FreeRunVsync`: The internally generated free-running vertical sync signal is used in the VDC4 (in Table 17), therefore this parameter is set to `VDC4_ON` (line 35).
- `Vsel.res_fv`: The free-running vertical sync period should be set to the same value with the LCD vertical sync period, so the vertical period (Vp - 1 in Table 19) is specified (line 36).
- `Vsel.res_fh`: The free-running horizontal sync period should be set to the same value with the LCD horizontal sync period, so the horizontal period (Hp - 1 in Table 19) is specified (line 37).
- `panel_icksel`: The peripheral bus clock 1 is used as the source of the panel clock (in Table 17), therefore this parameter is set to `VDC4_LCDPANEL_CLKSEL_PERI` (line 39).
- `panel_dcdr`: The frequency of the peripheral bus clock 1 used as the source of the panel clock is 66.67MHz (in Table 17). To generate the LCD panel clock frequency (8.0 ~ 9.0MHz in Table 18), the clock frequency division ration is set to `VDC4_LCDPANEL_CLKDIV_1_8` (66.67 / 8 = 8.33MHz, line 40).
- `tcon_half`: Half of the horizontal period (Hp / 2 in Table 19) is specified (line 41).
- `tcon_offset`: The horizontal reference offset control is not used, therefore this parameter is set to '0' (line 42).
- `outcnt_lcd_edge`: The LCD data is latched at the falling edge of the panel clock (in Table 18), so the LCD data should be output at the rising edge. This parameter is set to `VDC4_EDGE_RISING` (line 43).

4.3.2 Example of Settings for the LCD that requires Hsync and DE

An example of settings for the LCD that requires the horizontal sync (Hsync) signal and the data enable (DE) signal is as shown below.

Table 20 VDC4 Settings for the LCD

Output pins for the sig nals to drive the LCD panel	Hsync: LCD_TCON1 DE: LCD_TCON5
LCD TCON	STH/SP/HS: STH signal is used as Hsync. DE: DE signal is used as DE. STVB/VE: STVB signal is used as VE. STB/LP/HE: STB signal is used as HE.
The source of the panel clock	Peripheral clock 1 (66.67MHz)
Vertical sync signal output select	Internally generated free-running Vsync signal
Bit allocation of LCD signals	R[7:2]-G[7:2]-B[7:2]

Note: In addition to VDC4, a setup of I/O ports is required to use output pins LCD_TCON1 and LCD_TCON5.

Table 21 LCD Specifications

Display size	240 x 320
Output format	RGB666
Data sampling	The falling edge of the panel clock
Sync signal polarity	Hsync: negative polarity DE: positive polarity
Frequency	5.0 ~ 6.0 MHz

Table 22 Timing Signals for Driving the LCD Panel

Signal Sy	mbol	Value
Horizontal period	Hp	273 [clock]
Horizontal display period	Hdp	240 [clock]
Horizontal sync signal start position	Hss	0 [clock]
Horizontal sync signal pulse width	Hsw	5 [clock]
Data enable signal start position	DEs	22 [clock]
Vertical period	Vp	327 [line]
Vertical display period	Vdp	320 [line]

Note: Except for Hss, setting values are complied with LCD specification. Setting value of Hss defines the timing of the LCD sync signal from the VDC4 internal sync signal.

The DE signal is generated by synthesizing (AND) the horizontal enable (HE) signal and the vertical enable (VE) signal. The timing of the HE signal and the VE signal to generate the DE signal is shown in Table 23.

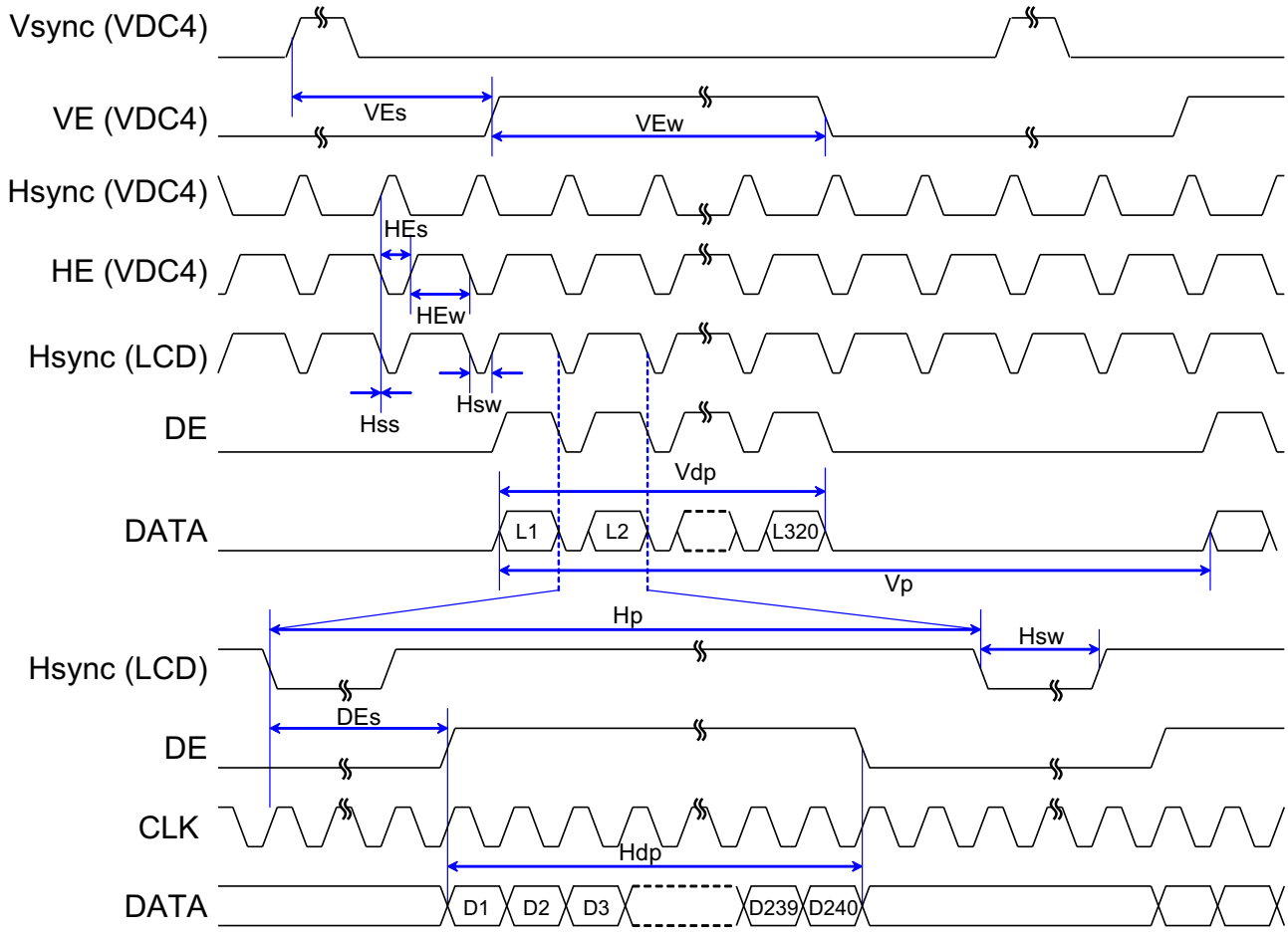
The setting value for VEs in Table 23 is also referred to as the vertical enable signal start position. The vertical enable signal start position must be set to 4 lines or greater because of the limitation of the VDC4. Therefore VEs should be between 4 lines and the vertical blanking period.

Table 23 Timing Signals to Generate DE Signal

Signal Sy	mbol	Value
Vertical enable signal start position	VEs	7 [line]* (= Vp - Vdp)
Vertical enable signal pulse width	VEw	320 [line] (= Vdp)
Horizontal enable signal start position	HEs	22 [clock] (= Hss + Des)
Horizontal enable signal pulse width	HEw	240 [clock] (= Hdp)

Note: * Setting value of VEs (Vp - Vdp) means a blanking period.

The control timing for driving the LCD panel is as shown below (see Figure 17).



Vsync (VDC4): VDC4 internal vertical sync signal
Hsync (VDC4): VDC4 internal horizontal sync signal
VE (VDC4): VDC4 internal vertical enable signal
HE (VDC4): VDC4 internal horizontal enable signal
Hsync (LCD): LCD horizontal sync signal
DE: LCD data enable signal
CLK: Panel clock (pixel clock)
DATA: Data
D1 ~ D240 … data per pixel
L1 ~ L320 … data per line

Figure 17 Signal Changing Timing of the LCD driven by Hsync and DE

The LCD settings are set to the structure `vdc4_InitAttr` as follows when VDC4_Initialize is called.

```
vdc4_InitAttr InitAttr ;
vdc4_LcdTconTim LcdTconTim_STH_SP_HS ;
vdc4_LcdTconTim LcdTconTim_STVB_VE ;
vdc4_LcdTconTim LcdTconTim_STB_LP_HE ;
vdc4_LcdTconTim LcdTconTim_DE ;
vdc4_LcdOutput LcdOutput ;

/* VE */
LcdTconTim_STVB_VE.tcon_hsvs = ( _UWORD )( 2 * 2 ) ;
   LcdTconTim_STVB_VE.tcon_hwvw = ( _UWORD )( 320 * 2 ) ;
   LcdTconTim_STVB_VE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
   LcdTconTim_STVB_VE.tcon_hs_sel = ( _UWORD )0 ;
   LcdTconTim_STVB_VE.tcon_inv = ( _UWORD )0 ;
   LcdTconTim_STVB_VE.tcon_pin = VDC4_LCD_TCON_PIN_NON ;
   LcdTconTim_STVB_VE.outcnt_edge = VDC4_EDGE_RISING ;
   /* HE */
   LcdTconTim_STB_LP_HE.tcon_hsvs = ( _UWORD )22 ;
   LcdTconTim_STB_LP_HE.tcon_hwvw = ( _UWORD )240 ;
   LcdTconTim_STB_LP_HE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
   LcdTconTim_STB_LP_HE.tcon_hs_sel = ( _UWORD )0 ;
   LcdTconTim_STB_LP_HE.tcon_inv = ( _UWORD )0 ;
   LcdTconTim_STB_LP_HE.tcon_pin = VDC4_LCD_TCON_PIN_NON ;
   LcdTconTim_STB_LP_HE.outcnt_edge = VDC4_EDGE_RISING ;
   /* DE */
   LcdTconTim_DE.tcon_hsvs = ( _UWORD )0 ;
   LcdTconTim_DE.tcon_hwvw = ( _UWORD )0 ;
   LcdTconTim_DE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
   LcdTconTim_DE.tcon_hs_sel = ( _UWORD )0 ;
   LcdTconTim_DE.tcon_inv = ( _UWORD )0 ;
   LcdTconTim_DE.tcon_pin = VDC4_LCD_TCON_PIN_5 ;
   LcdTconTim_DE.outcnt_edge = VDC4_EDGE_RISING ;
   /* Hsync */
   LcdTconTim_STH_SP_HS.tcon_hsvs = ( _UWORD )0 ;
   LcdTconTim_STH_SP_HS.tcon_hwvw = ( _UWORD )5 ;
   LcdTconTim_STH_SP_HS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
   LcdTconTim_STH_SP_HS.tcon_hs_sel = ( _UWORD )0 ;
   LcdTconTim_STH_SP_HS.tcon_inv = ( _UWORD )1 ;
   LcdTconTim_STH_SP_HS.tcon_pin = VDC4_LCD_TCON_PIN_1 ;
   LcdTconTim_STH_SP_HS.outcnt_edge = VDC4_EDGE_RISING ;
   /* LCD Output Control */
   LcdOutput.res_f.hs = ( _UWORD )22 ;
   LcdOutput.res_f.hw = ( _UWORD )240 ;
   LcdOutput.res_f.vs = ( _UWORD )2 ;
   LcdOutput.res_f.vw = ( _UWORD )320 ;
   LcdOutput.out_endian_on = VDC4_OFF ;
   LcdOutput.out_swap_on = VDC4_OFF ;
   LcdOutput.out_format = VDC4_LCD_OUTFORMAT_RGB666 ;
   LcdOutput.out_frq_sel = VDC4_LCD_PARALLEL_CLKFRQ_1 ;
   LcdOutput.out_dir_sel = ( _UWORD )0 ;
   LcdOutput.out_phase = VDC4_LCD_SERIAL_CLKPHASE_0 ;

   /* Initialization parameter */
   InitAttr.Vsel.FreeRunVsync = VDC4_ON ;
   InitAttr.Vsel.res_fv = ( _UWORD )( 327 - 1 ) ;
   InitAttr.Vsel.res_fh = ( _UWORD )( 273 - 1 ) ;
```

```
56    InitAttr.sync = NULL ;
57    InitAttr.panel_icksel = VDC4_LCDPANEL_CLKSEL_PERI ;
58    InitAttr.panel_dcdr = VDC4_LCDPANEL_CLKDIV_1_12 ;
59    InitAttr.tcon_half = ( _UWORD )136 ;
60    InitAttr.tcon_offset = ( _UWORD )0 ;
61    InitAttr.outcnt_lcd_edge = VDC4_EDGE_RISING ;
62    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVA_VS ] = NULL ;
63    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVB_VE ] = &LcdTconTim_STVB_VE ;
64    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STH_SP_HS ] = &LcdTconTim_STH_SP_HS ;
65    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STB_LP_HE ] = &LcdTconTim_STB_LP_HE ;
66    InitAttr.outctrl[ VDC4_LCD_TCONSIG_CPV_GCK ] = NULL ;
67    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLA ] = NULL ;
68    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLB ] = NULL ;
69    InitAttr.outctrl[ VDC4_LCD_TCONSIG_DE ] = &LcdTconTim_DE ;
70    InitAttr.settings = &LcdOutput ;
71    InitAttr.IntEnable_1 = VDC4_INT_BIT_VLINE ;
72    InitAttr.IntEnable_2 = VDC4_INT_BIT_NONE ;
```

The above sample code is explained as follows.

1. Setting VE
   STVB/VE signal generated by LCD TCON is used for the VE signal. The timing parameters of the VE signal are set in the structure `vdc4_LcdTconTim` (line 9 - 15). And a pointer to this structure `vdc4_LcdTconTim` is set in the array member `outctrl` (referred to by an index `VDC4_LCD_TCONSIG_STVB_VE`) in the structure `vdc4_InitAttr` as STVB/VE signal settings (line 63).
   — Setting Structure `vdc4_LcdTconTim`
   - `tcon_hsvs`: The vertical enable signal start position (VEs in Table 23) is specified in 1/2 cycles (line 9).
   - `tcon_hwvw`: The vertical enable signal pulse width (VEw in Table 23) is specified in 1/2 cycles (line 10).
   - `tcon_md`: In STVB/VE signal settings, this parameter is ignored.
   - `tcon_hs_sel`: In STVB/VE signal settings, this parameter is ignored.
   - `tcon_inv`: The vertical enable signal is not output, so this parameter does not affect anything.
   - `tcon_pin`: The vertical enable signal is not output, therefore this parameter is set to `VDC4_LCD_TCON_PIN_NON` (line 14).
   - `outcnt_edge`: The vertical enable signal is not output, so this parameter is ignored.
2. Setting HE
   STB/LP/HE signal generated by LCD TCON is used for the HE signal. The timing parameters of the HE signal are set in the structure `vdc4_LcdTconTim` (line 17 - 23). And a pointer to this structure `vdc4_LcdTconTim` is set in the array member `outctrl` (referred to by an index `VDC4_LCD_TCONSIG_STB_LP_HE`) in the structure `vdc4_InitAttr` as STVB/VE signal settings (line 65).
   — Setting Structure `vdc4_LcdTconTim`
   - `tcon_hsvs`: The horizontal enable signal start position (HEs in Table 23) is specified (line 17).
   - `tcon_hwvw`: The horizontal enable signal pulse width (HEw in Table 23) is specified (line 18).
   - `tcon_md`: In STB/LP/HE signal settings, this parameter is ignored.
   - `tcon_hs_sel`: The horizontal reference offset control is not used, therefore this parameter is set to '0' (line 20).
   - `tcon_inv`: The horizontal enable signal is not output, so this parameter does not affect anything.
   - `tcon_pin`: The horizontal enable signal is not output, therefore this parameter is set to `VDC4_LCD_TCON_PIN_NON` (line 22).
   - `outcnt_edge`: The horizontal enable signal is not output, so this parameter is ignored.
3. Setting DE
   DE signal generated by LCD TCON is used for the DE signal. The timing parameters of the DE signal are set in the structure `vdc4_LcdTconTim` (line 25 - 31). And a pointer to this structure `vdc4_LcdTconTim` is set in the array member `outctrl` (referred to by an index `VDC4_LCD_TCONSIG_DE`) in the structure `vdc4_InitAttr` as STVB/VE signal settings (line 69).
   — Setting Structure `vdc4_LcdTconTim`

- tcon_hsvs: In DE signal settings, this parameter is ignored.
- tcon_hwvw: In DE signal settings, this parameter is ignored.
- tcon_md: In DE signal settings, this parameter is ignored.
- tcon_hs_sel: In DE signal settings, this parameter is ignored.
- tcon_inv: The data enable signal is positive polarity (in Table 21), therefore this parameter is set to '0' (line 29).
- tcon_pin: The data enable signal output pin is LCD_TCON5 (in Table 20), therefore this parameter is set to VDC4_LCD_TCON_PIN_5 (line 30).
- outcnt_edge: Rising edge VDC4_EDGE_RISING is specified to synchronize with the LCD data signal (line 31).

4. Setting Hsync
   STH/SP/HS signal generated by LCD TCON is used for Hsync signal. The timing parameters of the Hsync signal are set in the structure vdc4_LcdTconTim (line 33 - 39). And a pointer to this structure vdc4_LcdTconTim is set in the array member outctrl (referred to by an index VDC4_LCD_TCONSIG_STH_SP_HS) in the structure vdc4_InitAttr as STH/SP/HS signal settings (line 64).
   — Setting Structure vdc4_LcdTconTim
   - tcon_hsvs: The horizontal sync signal start position (Hss in Table 22) is specified (line 33).
   - tcon_hwvw: The horizontal sync signal pulse width (Hsw in Table 22) is specified (line 34).
   - tcon_md: In STH/SP/HS signal settings, this parameter is ignored.
   - tcon_hs_sel: The horizontal reference offset control is not used, therefore this parameter is set to '0' (line 36).
   - tcon_inv: The horizontal sync signal is negative polarity (in Table 21), therefore this parameter is set to '1' (line 37).
   - tcon_pin: The horizontal sync signal output pin is LCD_TCON1 (in Table 20), therefore this parameter is set to VDC4_LCD_TCON_PIN_1 (line 38).
   - outcnt_edge: Rising edge VDC4_EDGE_RISING is specified to synchronize with the LCD data signal (line 39).
5. Setting LCD Output
   The LCD output parameters are set in the structure vdc4_LcdOutput (line 41 - 50). And a pointer to this structure vdc4_LcdOutput is set in the member settings in the structure vdc4_InitAttr (line 70).
   — Setting Structure vdc4_LcdOutput
   - res_f.hs: The horizontal enable signal start position (HEs in Table 23) is specified (line 41).
   - res_f.hw: The horizontal enable signal width (Hdp in Table 22) is specified (line 42).
   - res_f.vs: The vertical enable signal start position (VEs in Table 23) is specified (line 43).
   - res_f.vw: The vertical enable signal width (Vdp in Table 22) is specified (line 44).
   - out_endian_on: The LCD data signal is output MSB first (in Table 20), therefore this parameter is set to VDC4_OFF (line 45).
   - out_swap_on: The LCD data signal is output in order RGB (in Table 20), therefore this parameter is set to VDC4_OFF (line 46).
   - out_format: The LCD output format is RGB666 (in Table 21), therefore this parameter is set to VDC4_LCD_OUTFORMAT_RGB666 (line 47).
   - out_frq_sel: When the LCD output format is not serial RGB, this parameter is ignored.
   - out_dir_sel: When the LCD output format is not serial RGB, this parameter is ignored.
   - out_phase: When the LCD output format is not serial RGB, this parameter is ignored.
6. Setting Others
   Other settings for the LCD are set in the structure vdc4_InitAttr (line 53 - 61).
   — Setting Structure vdc4_InitAttr
   - Vsel.FreeRunVsync: The internally generated free-running vertical sync signal is used in the VDC4 (in Table 20), therefore this parameter is set to VDC4_ON (line 53).
   - Vsel.res_fv: The free-running vertical sync period should be set to the same value with the LCD vertical sync period, so the vertical period (Vp - 1 in Table 22) is specified (line 54).
   - Vsel.res_fh: The free-running horizontal sync period should be set to the same value with the LCD horizontal sync period, so the horizontal period (Hp - 1 in Table 22) is specified (line 55).
   - panel_icksel: The peripheral bus clock 1 is used as the source of the panel clock (in Table 20), therefore this parameter is set to VDC4_LCDPANEL_CLKSEL_PERI (line 57).

- `panel_dcdr`: The frequency of the peripheral bus clock 1 used as the source of the panel clock is 66.67MHz (in Table 20). To generate the LCD panel clock frequency (5.0 ~ 6.0MHz in Table 21), the clock frequency division ration is set to `VDC4_LCDPANEL_CLKDIV_1_12` (66.67 / 12 = 5.56MHz, line 58).
- `tcon_half`: Half of the horizontal period (Hp / 2 in Table 22) is specified (line 59).
- `tcon_offset`: the horizontal reference offset control is not used, therefore this parameter is set to '0' (line 60).
- `outcnt_lcd_edge`: The LCD data is latched at the falling edge of the panel clock (in Table 21), so the LCD data should be output at the rising edge. This parameter is set to `VDC4_EDGE_RISING` (line 61).

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

Rev.	Date	Description Page	Summary
1.00	Apr.18.2013	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clo ck Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2