RZ/A2Mグループ

Video ユーティリティー

要旨

本書はRZ/Aシリーズの RZ/A2M グループのルネサスビデオアプリケーションインタフェース(RVAPI)の機能仕様について説明します。

動作確認デバイス

RZ/A2M

目次

1. 仕様

RVAPI は、RZ/A2M に搭載されているビデオディスプレイコントローラ（VDC）、キャプチャエンジンユニット（CEU）、MIPI、スプライトエンジン（SPEA）およびビデオ入力モジュール（VIN）のドライバを使用して、ディスプレイおよびビデオ入力の制御を実現します。RVAPI は、各ドライバ制御の参考例としても使用できます。

RVAP で使用する周辺機能と用途を表 1-1に示します。

表 1-1 RVAPI で使用する周辺機能と用途

周辺機能	用途
RZ/A2M 内蔵 VDC 制御	表示及び映像入力制御 表示及び映像の画質調整
RZ/A2M 内蔵 CEU 制御	CMOS カメラの映像入力制御
RZ/A2M 内蔵 MIPI VIN 制御	MIPI カメラの映像入力制御

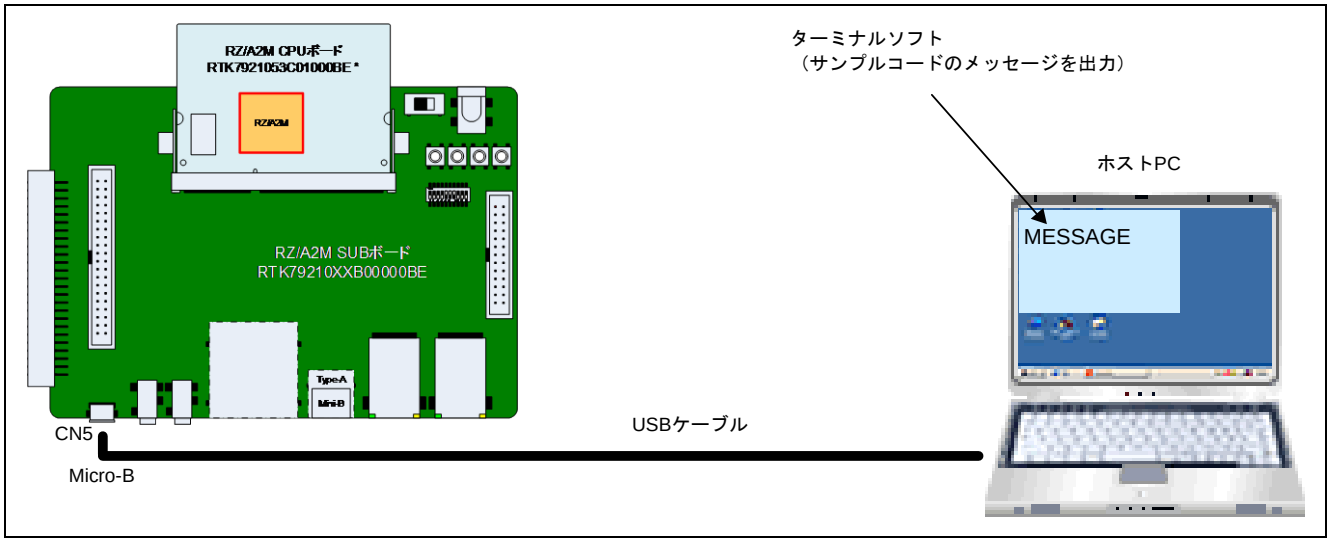

図1.1 動作環境

2. 動作環境

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1　動作確認条件

項目	内容
使用 MCU	RZ/A2M
動作周波数（注）	CPU クロック（Iφ）：528MHz 画像処理クロック（Gφ）：264MHz 内部バスクロック（Bφ）：132MHz 周辺クロック 1（P1φ）：66MHz 周辺クロック 0（P0φ）：33MHz QSPI0_SPCLK：66MHz CKIO：132MHz
動作電圧	電源電圧（I/O）：3.3V 電源電圧（1.8/3.3V 切替 I/O（PVcc_SPI））：3.3V 電源電圧（内部）：1.2V
統合開発環境	e2 studio V7.4.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update
	コンパイラオプション（ディレクトリパスの追加は除く） Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0
動作モード	ブートモード 3 （シリアルフラッシュブート 3.3V 品）
ターミナルソフトの通信設定	• 通信速度：115200bps • データ長：8 ビット • パリティ：なし • ストップビット長：1 ビット • フロー制御：なし
使用ボード	RZ/A2M CPU ボード　RTK7921053C00000BE RZ/A2M SUB ボード　RTK79210XXB00000BE
使用デバイス （ボード上で使用する機能）	• シリアルフラッシュメモリ（SPI マルチ I/O バス空間に接続） メーカ名：Macronix 社、型名：MX25L51245GXD • RL78/G1C（USB 通信とシリアル通信を変換し、ホスト PC との通信に使用） • LED1

【注】 クロックモード 1（EXTAL 端子からの 24MHz のクロック入力）で使用時の動作周波数です。

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/A2M グループ　Capture Engine Unit Sample Driver(R01AN4474)
  CEU ドライバの機能仕様を記載しています。
- RZ/A2M グループ　Video Display Controller and Sprite Engine Sample Driver(R01AN4475)
  VDC および SPEA ドライバの機能仕様を記載しています。
- RZ/A2M グループ　MIPI Driver(R01AN448)
  MIPI ドライバの機能仕様を記載しています。

4. ハードウェア説明

4.1 ハードウェア構成例

ハードウェア構成については RZ/A2M 評価ボードのマニュアルを参照ください。

4.2 使用端子一覧

使用端子と機能を表 4-1に示します。

表 4-1 使用端子と機能(注)

端子名	入出力	内容	RZ/A2M 評価ボード接続
DV0_CLK	入力	外部入力クロック 0	未使用
DV0_VSYNC	入力	外部入力垂直同期 0	未使用
DV0_HSYNC	入力	外部入力水平同期 0	未使用
DV0_DATA23~0	入力	外部入力映像データ 0	未使用
LCD0_CLK	出力	パネルクロック 0	PJ_6
LCD0_DATA23~0	出力	パネル用映像データ 0	PB_5-0, PA_7-0, P8_0, PF_7-0, PH_2
LCD0_TCON6~0	出力	パネル用制御信号 0	PC_3(TCON4), PC_4(TCON3), P7_7(TCON0)
LCD0_EXTCLK	入力	パネルクロックソース 0	PJ_6
TXCLKOUTM/P	出力	LVDS クロック出力端子	P4_7, P4_6
TXOUT2M/P	出力	LVDS データ出力端子	P4_5, P4_4
TXOUT1M/P	出力	LVDS データ出力端子	P4_3, P4_2
TXOUT0M/P	出力	LVDS データ出力端子	P4_1, P4_0
VIO7~VIO0	入力	CEU 用データバス	PE_6-1, PH_1-0
VIO_CLK	入力	CEU 用クロック	P6_1
VIO_VD	入力	CEU 用垂直同期	P6_2
VIO_HD	入力	CEU 用水平同期	P6_3
VIO_FLD	入力	フィールド信号	未接続
CSI_DATA0P	入力	CSI2 データレーン 0 (差動ポジティブ)	専用端子
CSI_DATA0N	入力	CSI2 データレーン 0 (差動ネガティブ)	専用端子
CSI_DATA1P	入力	CSI2 データレーン 1 (差動ポジティブ)	専用端子
CSI_DATA1N	入力	CSI2 データレーン 1 (差動ネガティブ)	専用端子
CSI_CLKP	入力	CSI2 クロックレーン (差動ポジティブ)	専用端子
CSI_CLKN	入力	CSI2 クロックレーン (差動ネガティブ)	専用端子

【注】詳細については、各評価ボードの仕様をご確認ください

5. ソフトウェア説明

5.1 各種関数

表 5-1 に RVAPI 関数の一覧を記載します。『表示のみ』、『映像入力のみ』、『映像入力し表示する』場合に設定が必須な関数についてもあわせて記載しています。

表 5-1 RVAPI 関数一覧(1/2)

表示のみ	映像入力	映像表示	関数名	項番	概要
VDC 映像入力表示関数					
必須	必須	必須	R_RVAPI_InitializeVDC	6.1	VDC 初期化クロック設定
-	-	-	R_RVAPI_TerminateVDC	6.2	VDC 終了設定
必須	-	必須	R_RVAPI_DispControlVDC	6.3	ディスプレイ出力設定
必須	-	-	R_RVAPI_GraphCreateSurfaceVDC	6.4	表示領域の生成
-	-	-	R_RVAPI_GraphChangeSurfaceVDC	6.5	表示バッファアドレスの変更
-	-	-	R_RVAPI_GraphChangeSurfaceConfigVDC	6.6	データ読み出し処理の設定変更
-	-	-	R_RVAPI_GraphDestroySurfaceVDC	6.7	表示領域の破棄
必須	-	必須	R_RVAPI_DispPortSettingVDC	6.8	ディスプレイ出力端子設定
-	必須	必須	R_RVAPI_VideoControlVDC	6.9	映像入力設定
-	必須	必須	R_RVAPI_VideoCreateSurfaceVDC	6.10	映像領域と表示領域の生成
			R_RVAPI_VideoCreateSurfaceIMRL2	6.11	IMR-LS2 用映像表示領域生
-		-	R_RVAPI_VideoDestroySurfaceVDC	6.12	映像領域と表示領域の破棄
-	必須	必須	R_RVAPI_VideoPortSettingVDC	6.13	映像入力端子設定
-	-	-	R_RVAPI_InterruptEnableVDC	6.14	VDC 割り込み許可設定
-	-	-	R_RVAPI_InterruptDisableVDC	6.15	VDC 割り込み禁止設定
-	-	-	R_RVAPI_AlphablendingRectVDC	6.16	矩形アルファブレンド
			R_RVAPI_ChromakeyVDC	6.17	クロマキーを使用した透過
VDC 画質調整関数					
-	-	-	R_RVAPI_DispCalibrationVDC	6.18	画面出力較正処理
-	-	-	R_RVAPI_DispGammaVDC	6.19	ガンマ補正設定
-	-	-	R_RVAPI_VideoCalibrationVDC	6.20	カラーマトリクス設定
-	-	-	R_RVAPI_VideoSharpnessLtiVDC	6.21	画質改善設定処理
-	-	-	R_RVAPI_AlphablendingVDC	6.22	1bit alpha blending setup
CEU 映像入力関数(注 1)					
-	必須	必須	R_RVAPI_InitializeCEU	7.1	CEU 初期化設定
-	-	-	R_RVAPI_TerminateCEU	7.2	CEU 終了設定
-	必須	必須	R_RVAPI_PortSettingCEU	7.3	映像入力端子設定
-	必須	必須	R_RVAPI_OpenCEU	7.4	映像取り込み設定
-	必須	必須	R_RVAPI_CaptureStartCEU	7.5	フレームキャプチャ開始
-	-	-	R_RVAPI_CaptureStopCEU	7.6	キャプチャ停止
-	-	-	R_RVAPI_InterruptEnableCEU	7.7	割り込み許可設定

【注 1】映像入力に CEU を使用する場合、設定が必要になります

表 5-2 RVAPI 関数一覧(1/2)

表示のみ	映像入力	映像表示	関数名	項番	概要
MIPI 映像入力関数(注 2)					
-	必須	必須	R_RVAPI_InitializeMIPI	8.1	MIPI 初期化設定
-	-	-	R_RVAPI_TerminateMIPI	8.2	MIPI 終了設定
-	必須	必須	R_RVAPI_OpenMIPI	8.3	MIPI 映像取り込み設定
-	-	-	R_RVAPI_InterruptEnableMIPI	8.4	割り込み許可設定
-	必須	必須	R_RVAPI_SetupMIPI	8.5	VIN 映像取り込み設定
	必須	必須	R_RVAPI_SetBufferMIPI	8.6	キャプチャバッファ設定
-	必須	必須	R_RVAPI_CaptureStartMIPI	8.7	フレームキャプチャ開始
-	-	-	R_RVAPI_CaptureStopMIPI	8.8	キャプチャ停止
SPEA 表示設定関数					
必須	-	-	R_RVAPI_GraphCreateSurfaceSPEA	9.1	表示領域の生成(SPEA)
-	-	-	R_RVAPI_WindowOffsetSPEA	9.2	SPEA Windowの座標オフセットの設定
必須	-	-	R_RVAPI_SetWindowSPEA	9.3	SPEA Window パラメータの設定
必須	-	-	R_RVAPI_WindowUpdateSPEA	9.4	SPEA Window パラメータの更新
必須			R_RVAPI_GraphCreateSurfaceRLE	9.5	表示領域の生成(RLE)
必須	-	-	R_RVAPI_SetWindowRLE	9.6	RLE パラメータの設定及び更新

【注 2】映像入力に MIPI を使用する場合、設定が必要になります

6. 関数リファレンス(VDC)

6.1 R_RVAPI_InitializeVDC

R_RVAPI_InitializeVDC

概　要　VDC 初期化クロック設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_InitializeVDC(
                const vdc_channel_t ch,
                const clock_config_t * const c_cnf);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	clock_config_t * c_cnf	: クロックコンフィグレーション
リターン値	VDC_OK:		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
	VDC_ERR_PARAM_NULL		: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH		: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED		: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE		: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION		: 不許可条件エラー
	VDC_ERR_RESOURCE_LVDS_CLK		: LVDS クロックリソースエラー

備考

(1) 説明

VDC では、様々な入力クロックをソースクロックとしてパネルクロックを生成することが可能です。本関数では、そのクロック設定を行います。パネルクロックは、表示機器の制御に使用される為、表示機器の仕様に合わせたクロックを設定する必要があります。

本関数内では、以下のドライバを使用しています。

- R_VDC_Initialize ()

(2) パラメータ詳細

(a) clock_config_t

clock_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_panel_clksel_t    panel_clk;
    vdc_panel_clk_dcdr_t   panel_clk_div;
    const vdc_lvds_t    * lvds;
} clock_config_t;
```

型/メンバ名	説明
vdc_panel_clksel_t panel_clk	パネルクロック選択 • VDC_PANEL_ICKSEL_IMG_DV 映像クロック (DV_CLK)の分周クロック • VDC_PANEL_ICKSEL_EXT_0 外部クロック 0 (LCD0_EXTCLK)の分周クロック • VDC_PANEL_ICKSEL_PERI 周辺クロック 1 (P1φ)の分周クロック • VDC_PANEL_ICKSEL_LVDS: LVDS PLL のクロック • VDC_PANEL_ICKSEL_LVDS_DIV7 LVDS PLL の 7 分周クロック
vdc_panel_clk_dcdr_t panel_clk_div	クロック分周比設定 • VDC_PANEL_CLKDIV_1_1: 1/1 • VDC_PANEL_CLKDIV_1_2: 1/2 • VDC_PANEL_CLKDIV_1_3: 1/3 • VDC_PANEL_CLKDIV_1_4: 1/4 • VDC_PANEL_CLKDIV_1_5: 1/5 • VDC_PANEL_CLKDIV_1_6: 1/6 • VDC_PANEL_CLKDIV_1_7: 1/7 • VDC_PANEL_CLKDIV_1_8: 1/8 • VDC_PANEL_CLKDIV_1_9: 1/9 • VDC_PANEL_CLKDIV_1_12: 1/12 • VDC_PANEL_CLKDIV_1_16: 1/16 • VDC_PANEL_CLKDIV_1_24: 1/24 • VDC_PANEL_CLKDIV_1_32: 1/32 このパラメータはパネルクロック選択 (panel_icksel)が LVDS PLL (VDC_PANEL_ICKSEL_LVDS か VDC_PANEL_ICKSEL_LVDS_DIV7)の時は無効です
const vdc_lvds_t * lvds	LVDS 関連パラメータ 不要な場合は NULL を設定してください

(b) vdc_lvds_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_lvds_in_clk_sel_t   lvds_in_clk_sel;
    vdc_lvds_ndiv_t         lvds_idiv_set;     /* Not use */
    uint16_t             lvdspll_tst;      /* Not use */
    vdc_lvds_ndiv_t      lvds_odiv_set;
    vdc_channel_t        lvds_vdc_sel;
    uint16_t             lvdspll_fd;
    uint16_t             lvdspll_rd;
    vdc_lvds_pll_nod_t      lvdspll_od;        /* Not use */
} vdc_lvds_t;
```

型/メンバ名	説明
vdc_lvds_in_clk_sel_t lvds_in_clk_sel	分周器 1 への入力クロック選択 • VDC_LVDS_INCLK_SEL_DV_0: DV0_CLK0 • VDC_LVDS_INCLK_SEL_EXT_0: LCD0_EXTCLK • VDC_LVDS_INCLK_SEL_PERI: P1φ
vdc_lvds_ndiv_t lvds_idiv_set	分周器 1 の分周数 NIDIV 設定(未使用) • VDC_LVDS_NDIV_1: NIDIV = 1 • VDC_LVDS_NDIV_2: NIDIV = 2 • VDC_LVDS_NDIV_4: NIDIV = 4
uint16_t lvdspll_tst	LVDS PLL の内部パラメータ設定(未使用)
vdc_lvds_ndiv_t lvds_odiv_set	分周器 2 の分周数 NODIV 設定 • VDC_LVDS_NDIV_1: NODIV = 1 • VDC_LVDS_NDIV_2: NODIV = 2 • VDC_LVDS_NDIV_4: NODIV = 4
vdc_channel_t lvds_vdc_sel	LVDS から出力する VDC のチャネル選択 • VDC_CHANNEL_0
uint16_t lvdspll_fd	LVDS PLL の帰還分周 NFD 設定 NRD = lvdspll_fd + 1 NFD = lvdspll_fd (22 ~ 62)
uint16_t lvdspll_rd	LVDS PLL の入力分周 NRD 設定 NRD = lvdspll_rd + 1 lvdspll_rd (0 ~ 7)
vdc_lvds_pll_nod_t lvdspll_od	LVDS PLL の出力分周 NOD 設定(未使用) • VDC_LVDS_PLL_NOD_1: NOD = 1 • VDC_LVDS_PLL_NOD_2: NOD = 2 • VDC_LVDS_PLL_NOD_4: NOD = 4 • VDC_LVDS_PLL_NOD_8: NOD = 8

(3) パネルクロックの設定

VDC のパネルクロック設定例を表 6-1に記載します。また、LVDS の PLL で生成したクロックは、LVDS 液晶出力以外にも使用することが可能な為、任意のクロックが作成可能です。LVDS の PLL を使用した場合の設定例を表 6-2に記載します。

表 6-1 パネルクロック設定例

メンバ名	33.0[MHz]	22.0[MHz]
panel_icksel	VDC_LVDS_INCLK_SEL_PERI 周辺クロック 1 (P1φ) 66.0[MHz]	
panel_dcdr	VDC_PANEL_CLKDIV_1_2	VDC_PANEL_CLKDIV_1_3

【注】周辺クロック 1 (P1φ)は、66.0[MHz]を想定しています

表 6-2 LVDS PLL を使用したパネルクロック設定例

メンバ名	74.25[MHz]	85.25[MHz]
panel_icksel	VDC_PANEL_ICKSEL_LVDS	VDC_PANEL_ICKSEL_LVDS
lvds_in_clk_sel	VDC_LVDS_INCLK_SEL_PERI	VDC_LVDS_INCLK_SEL_PERI
lvds_idiv_set	-	-
lvds_odiv_set	VDC_LVDS_NDIV_4	VDC_LVDS_NDIV_4
lvdspll_fd	(27u-1u)	(31u-1u)
lvdspll_rd	(6u-1u)	(6u-1u)
lvdspll_od	-	-

【注】周辺クロック 1 (P1φ)は、66.0[MHz]を想定しています

6.2 R_RVAPI_TerminateVDC

R_RVAPI_TerminateVDC			
概　要	VDC 終了設定		
ヘッダ	r_rvapi_vdc.h		
宣　言	vdc_error_t R_RVAPI_TerminateVDC(const vdc_channel_t ch);		
引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
リターン値	VDC_OK		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
備考			

(1) 説明

本関数では、VDC ドライバの終了処理を行います。VDC の割り込み及びパネルクロックのディセーブル処理などを行います。

本関数内では、以下のドライバを使用しています。

- R_VDC_Terminate ()

6.3 R_RVAPI_DispControlVDC

R_RVAPI_DispControlVDC

概　要　ディスプレイ出力設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_DispControlVDC(
                const vdc_channel_t ch,
                const vdc_onoff_t res_vs_sel,
                const qe_config_t * const q_cnf);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_onoff_t res_vs_sel	: 出力する垂直同期信号の選択 (自走同期信号) • VDC_OFF(注 1) 映像入力の垂直同期信号を液晶の垂直同期信号に使用します • VDC_ON 内部生成した自走用垂直同期信号
	[IN]	qe_config_t * q_cnf	: ディスプレイ出力コンフィグレーション

リターン値	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_RESOURCE_CLK	: クロックリソースエラー
	VDC_ERR_RESOURCE_INPUT	: 入力信号リソースエラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
	VDC_ERR_RESOURCE_VSYNC	: 垂直同期信号リソースエラー

備考

【注 1】映像入力がない場合、設定不可

(1) 説明

本関数では、ディスプレイ出力に関する設定を行います。設定値については、統合開発環境 e² studio 上で動作するソリューション・ツールキット『RZ/A ディスプレイ対応開発支援ツール QE for Video Display Controller 5』で生成した値をそのまま設定することも可能になります。『RZ/A ディスプレイ対応開発支援ツール QE for Video Display Controller 5』については、ルネサス Web サイトをご確認ください。

本関数内では、以下のドライバを使用しています。

- R_VDC_SyncControl ()
- R_VDC_DisplayOutput ()

(2) パラメータ詳細

(a) qe_config_t

qe_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t           vps;
    uint16_t           vpw;
    uint16_t           vs;
    uint16_t           vdp;
    uint16_t           hps;
    uint16_t           hpw;
    uint16_t           hs;
    uint16_t           hdp;
    uint16_t           vtp;
    uint16_t           htp;
    vdc_lcd_tcon_pin_t  tcon_vsync;
    vdc_lcd_tcon_pin_t  tcon_hsync;
    vdc_lcd_tcon_pin_t  tcon_de;
    vdc_sig_pol_t      tcon_vsync_inv;
    vdc_sig_pol_t      tcon_hsync_inv;
    vdc_sig_pol_t      tcon_de_inv;
    uint16_t           tcon_half;
    uint16_t           tcon_ofset;
    vdc_edge_t         lcd_data_out_edge;
    vdc_lcd_outformat_t lcd_outformat;
} qe_config_t;
```

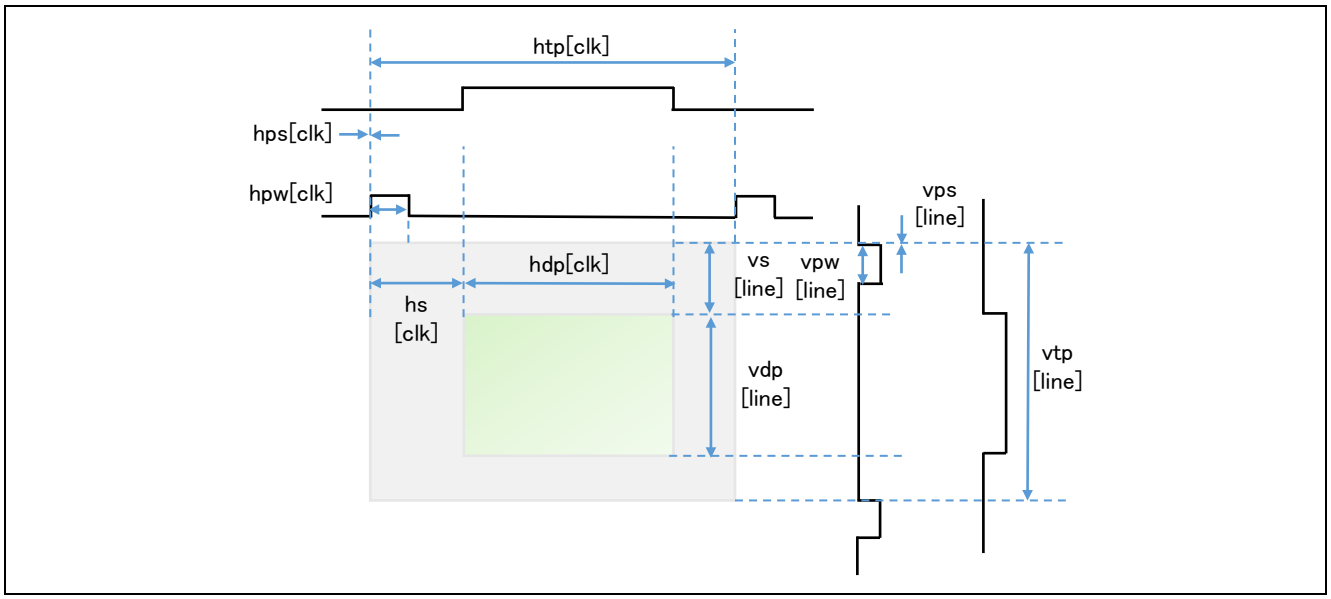

図 6-1 信号設定パラメータ関係図

型 / メンバ名		説明
uint16_t	vps	Vsync パルス開始位置 [line 単位]
uint16_t	vpw	Vsync パルス幅 [line 単位]
uint16_t	vs	表示領域の垂直開始位置 [line 単位]
uint16_t	vdp	垂直表示期間 [line]
uint16_t	hps	Hsync パルス開始位置[clk 単位]
uint16_t	hpw	Hsync パルス幅 [clk 単位]
uint16_t	hs	表示領域の水平開始位置[clk 単位]
uint16_t	hdp	水平表示期間[clk]
uint16_t	vtp	垂直トータル期間 [line 単位]
uint16_t	htp	水平トータル期間 [clk 単位]
vdc_lcd_tcon_pin_t	tcon_vsync	LCD TCON 出力端子選択 • VDC_LCD_TCON_PIN_NON (-1): 出力なし • VDC_LCD_TCON_PIN_0 (0): LCD_TCON0 出力 • VDC_LCD_TCON_PIN_1 (1): LCD_TCON1 出力 • VDC_LCD_TCON_PIN_2 (2): LCD_TCON2 出力 • VDC_LCD_TCON_PIN_3 (3): LCD_TCON3 出力 • VDC_LCD_TCON_PIN_4 (4): LCD_TCON4 出力 • VDC_LCD_TCON_PIN_5 (5): LCD_TCON5 出力 • VDC_LCD_TCON_PIN_6 (6): LCD_TCON6 出力
vdc_lcd_tcon_pin_t	tcon_hsync	
vdc_lcd_tcon_pin_t	tcon_de	
vdc_sig_pol_t	tcon_vsync_inv	水平信号の動作基準選択 • ・ VDC_LCD_TCON_REFSEL_HSYNC (0): 水平同期信号基準 • ・ VDC_LCD_TCON_REFSEL_OFFSET_H (1): オフセット後の水平同期信号基準
vdc_sig_pol_t	tcon_hsync_inv	
vdc_sig_pol_t	tcon_de_inv	
uint16_t	tcon_half	htp を設定してください
uint16_t	tcon_ofset	0 を設定してください
vdc_edge_t	lcd_data_out_edge	LCD_DATA23 ~ 0 端子の出力位相制御 • VDC_EDGE_RISING: LCD_CLK 端子の立ち上がりエッジで出力 • VDC_EDGE_FALLING: LCD_CLK 端子の立ち下がりエッジで出力
vdc_lcd_outformat_t	lcd_outformat	出力フォーマット選択 • VDC_LCD_OUTFORMAT_RGB888 (0): RGB888 • VDC_LCD_OUTFORMAT_RGB666 (1): RGB666 • VDC_LCD_OUTFORMAT_RGB565 (2): RGB565

6.4 R_RVAPI_GraphCreateSurfaceVDC

R_RVAPI_GraphCreateSurfaceVDC

概　要　表示領域の生成

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_GraphCreateSurfaceVDC(
                const vdc_channel_t ch,
                const gr_surface_disp_config_t * const gr_disp_cnf);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	gr_surface_disp_config_t * gr_disp_cnf	: グラフィックス表示領域の設定

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	: レイヤ ID 不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数では、バッファに配置されたメモリを表示する為の設定を行ないます。

本関数内では、以下のドライバを使用しています。

- R_VDC_ReadDataControl ()
- R_VDC_CLUT ()
- R_VDC_StartProcess ()

(2) パラメータ詳細

(a) gr_surface_disp_config_t

gr_surface_disp_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_layer_id_t       layer_id;
    vdc_pd_disp_rect_t    disp_area;
    void               * fb_buff;
    uint32_t             fb_stride;
    vdc_gr_format_t      read_format;
    uint32_t           * clut_table;
    vdc_gr_ycc_swap_t    read_ycc_swap;
    vdc_wr_rd_swa_t      read_swap;
    vdc_gr_disp_sel_t    disp_mode;
} gr_surface_disp_config_t;
```

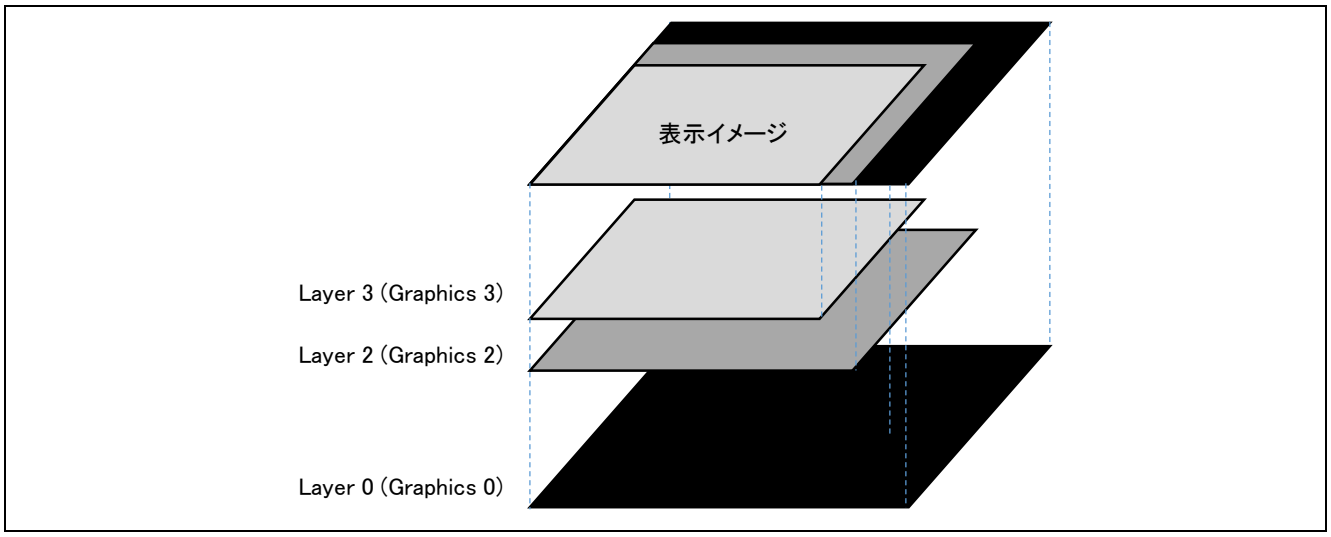

図 6-2 レイヤ構成

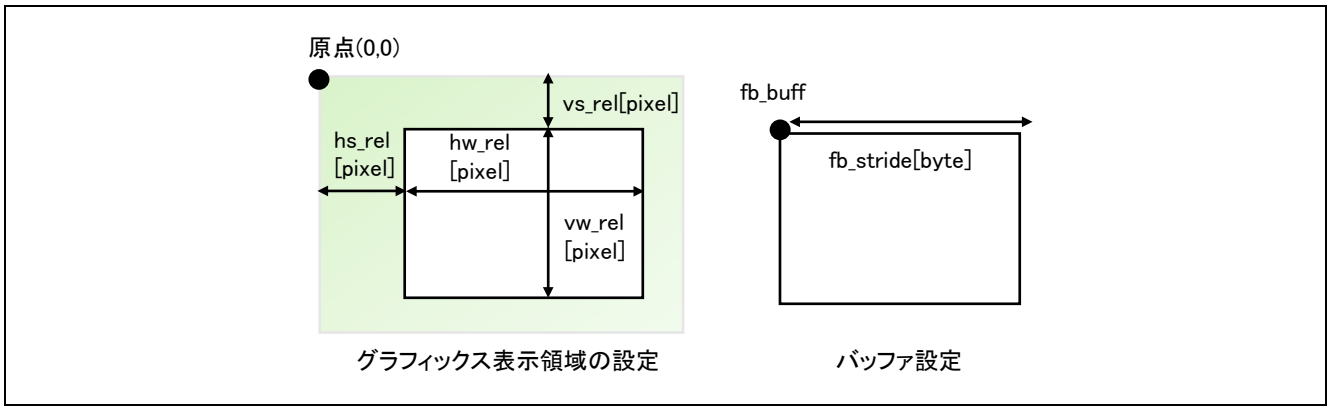

図 6-3 グラフィックスパラメータ関係図

型 / メンバ名	説明
vdc_layer_id_t layer_id	表示レイヤ(図 6-2を参照) • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
vdc_pd_disp_rect_t disp_area	グラフィックス表示領域[pixel 単位] (図 6-3 を参照) • disp_area.vs_rel / vw_rel：垂直表示開始位置 / 垂直表示サイズ • disp_area.hs_rel / hw_rel：水平表示開始位置 / 水平表示サイズ vs_rel = hs_rel = 0 で原点から表示されます
void * fb_buff	フレームバッファのベースアドレス(図 6-3 を参照) NULL は設定しないでください
uint32_t fb_stride	フレームバッファのラインオフセットアドレス [byte 単位] (図 6-3 を参照) 32[byte]の倍数で指定してください
vdc_gr_format_t read_format	フレームバッファ読み出し信号のフォーマット • VDC_GR_FORMAT_RGB565 (0): RGB565 • VDC_GR_FORMAT_ARGB8888 (4): ARGB8888 • VDC_GR_FORMAT_CLUT8 (5): CLUT8 • VDC_GR_FORMAT_CLUT4 (6): CLUT4 • VDC_GR_FORMAT_CLUT1 (7): CLUT1 • VDC_GR_FORMAT_YCBCR422 (8): YCbCr422 (注 1) • VDC_GR_FORMAT_RGBA8888 (11): RGBA8888
uint32_t * clut_table	カラールックアップテーブル このパラメータは read_format に指定された値が VDC_GR_FORMAT_CLUT8/4/1 の場合のみ有効です 色数分の CLUT データ(ARGB8888)の格納アドレスを設定してください この時、NULL を選択した場合、デフォルトの CLUT データが設定されます (デフォルト) CLUT8(256 色)：CLUT 番号 0～255 モノクロ (黒→白) CLUT4(16 色)：CLUT 番号 0～15 黒、赤、緑、水色、青、ピンク、茶、 深緑、枯草、紺、紫、灰色、橙、白、透過色 CLUT1(2 色)：CLUT 番号 0 ～1 黒、白
vdc_gr_ycc_swap_t read_ycc_swap	YCbCr422 フォーマット時バッファ読み出しデータのスワップ制御 このパラメータは read_format に指定された値が VDC_GR_FORMAT_YCBCR422 の場合のみ有効です • VDC_GR_YCCSWAP_CBY0CRY1 (0): CbY0/Cr/Y1 • VDC_GR_YCCSWAP_Y0CBY1CR (1): Y0/Cb/Y1/Cr • VDC_GR_YCCSWAP_CRY0CBY1 (2): Cr/Y0/Cb/Y1 • VDC_GR_YCCSWAP_Y0CRY1CB (3): Y0/Cr/Y1/Cb • VDC_GR_YCCSWAP_Y1CRY0CB (4): Y1/Cr/Y0/Cb • VDC_GR_YCCSWAP_CRY1CBY0 (5): Cr/Y1/Cb/Y0 • VDC_GR_YCCSWAP_Y1CBY0CR (6): Y1/Cb/Y0/Cr • VDC_GR_YCCSWAP_CBY1CRY0 (7): Cb/Y1/Cr/Y0
vdc_wr_rd_swa_t read_swap	8 ビット/16 ビット/32 ビットスワップ設定 • VDC_WR_RD_WRSWA_NON (0): スワップなし 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit スワップ 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2):

	16-bit スワップ 3-4-1-2-7-8-5-6 • VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit スワップ 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit スワップ 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit スワップ 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit スワップ 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 32-bit + 16-bit + 8-bit スワップ 8-7-6-5-4-3-2-1
vdc_gr_disp_sel_t disp_mode	グラフィックス表示設定 • VDC_DISPSEL_BACK：背景色表示 • VDC_DISPSEL_LOWER：下層グラフィックス表示 • VDC_DISPSEL_CURRENT：カレントグラフィックス表示 • VDC_DISPSEL_BLEND： 下層グラフィックスとカレントグラフィックスのブレンド表示

【注 1】Layer 0 で設定可能

6.5 R_RVAPI_GraphChangeSurfaceVDC

R_RVAPI_GraphChangeSurfaceVDC

概　要　表示バッファアドレスの変更

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_GraphChangeSurfaceVDC(
            const vdc_channel_t ch,
            const vdc_layer_id_t layer_id,
            void* const fb_buff);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id	: レイヤ ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	void * framebuff	: フレームバッファのベースアドレス

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	: レイヤ ID 不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数では、データ読み出しバッファアドレスの変更を行います。

本関数内では、以下のドライバを使用しています。

- R_VDC_ChangeReadProcess ()

6.6 R_RVAPI_GraphChangeSurfaceConfigVDC

R_RVAPI_GraphChangeSurfaceConfigVDC

概　要　データ読み出し処理の設定変更

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_GraphChangeSurfaceConfigVDC (
                const vdc_channel_t ch,
                const vdc_layer_id_t layer_id,
                void* const fb_buff,
                vdc_period_rect_t * const gr_grc,
                vdc_width_read_fb_t * const width_read_fb,
                vdc_gr_disp_sel_t * const gr_disp_sel);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id	: レイヤ ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	void * framebuff	: フレームバッファのベースアドレス
	[IN]	vdc_period_rect_t * gr_grc	: グラフィックス表示領域
	[IN]	vdc_width_read_fb_t * width_read_fb	: 読み出しフレームバッファサイズ
	[IN]	vdc_gr_disp_sel_t * r_disp_sel	: グラフィックス表示設定

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	: レイヤ ID 不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数では、データ読み出し処理の設定を変更します。

本関数内では、以下のドライバを使用しています。

- R_VDC_ChangeReadProcess ()

(2) パラメータ詳細説明

(a) vdc_period_rect_t

vdc_period_rect_t は VDC の信号の水平／垂直タイミングを表す構造体です。

```
typedef struct
{
    uint16_t    vs;
    uint16_t    vw;
    uint16_t    hs;
    uint16_t    hw;
} vdc_period_rect_t;
```

型 / メンバ名	説明
uint16_t vs	基準信号からの垂直信号開始位置 (ライン数)
uint16_t vw	垂直信号幅 (ライン数)
uint16_t hs	基準信号からの水平信号開始位置 (クロック数)
uint16_t hw	水平信号幅 (クロック数)

(b) vdc_width_read_fb_t

vdc_width_read_fb_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t              in_vw;
    uint16_t              in_hw;
} vdc_width_read_fb_t;
```

型 メンバ名	説明
uint16_t in_vw	1 フレームのライン数設定 0x0000 ~ 0x07FF
uint16_t in_hw	水平有効期間の幅設定 0x0000 ~ 0x07FF

(c) vdc_gr_disp_sel_t

vdc_gr_disp_sel_t はグラフィックス表示設定のタイプを表す列挙型です。

```
typedef enum
{
    VDC_DISPSEL_IGNORED    = -1,
    VDC_DISPSEL_BACK       = 0,
    VDC_DISPSEL_LOWER      = 1,
    VDC_DISPSEL_CURRENT    = 2,
    VDC_DISPSEL_BLEND      = 3,
    VDC_DISPSEL_NUM        = 4
} vdc_gr_disp_sel_t;
```

列挙定数	値	説明
VDC_DISPSEL_IGNORED	-1	無視、変更なし
VDC_DISPSEL_BACK	0	背景色表示
VDC_DISPSEL_LOWER	1	下層グラフィックス表示
VDC_DISPSEL_CURRENT	2	カレントグラフィックス表示
VDC_DISPSEL_BLEND	3	下層グラフィックスとカレントグラフィックスのブレンド表示
VDC_DISPSEL_NUM	4	グラフィックス表示設定のタイプ数

6.7 R_RVAPI_GraphDestroySurfaceVDC

R_RVAPI_GraphDestroySurfaceVDC

概　要　表示領域の破棄

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_GraphDestroySurfaceVDC(
                const vdc_channel_t ch,
                const vdc_layer_id_t layer_id);
```

引　数	[IN]	vdc_channel_t ch	：VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id	：レイヤ ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD

リターン値	VDC_OK:	：正常終了
	VDC_ERR_PARAM_CHANNEL	：チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	：レイヤ ID 不正エラー
	VDC_ERR_RESOURCE_LAYER	：レイヤリソースエラー

備考

(1) 説明

本関数では、指定されたレイヤの停止処理を行います。フレームバッファからの読み出しを停止し、レイヤのグラフィックス表示設定を初期状態に戻します。

本関数内では、以下のドライバを使用しています。

- R_VDC_StopProcess ()
- R_VDC_ReleaseDataControl ()

6.8 R_RVAPI_DispPortSettingVDC

R_RVAPI_DispPortSettingVDC

概　要　　ディスプレイ出力端子設定

ヘッダ　　r_rvapi_vdc.h

宣　言

```
void R_RVAPI_DispPortSettingVDC(
       const vdc_channel_t ch,
       void (* const port_func)(uint32_t));
```

引　数	[IN]	vdc_channel_t ch	：VDC チャネル • VDC_CHANNEL_0
	[IN]	void (*port_func) (uint32_t)	：表示制御端子の設定を行う関数ポインタ

リターン値　なし:

備考

(1) 説明

本関数に設定するコールバック関数では、ディスプレイ出力に必要な端子設定を行なってください。本関数は、図 6-4に示すように VDC の表示設定を全て行ってから呼び出してください。表示設定を行う前に端子設定を行った場合、予期しない周期の制御信号を出力する可能性があります。

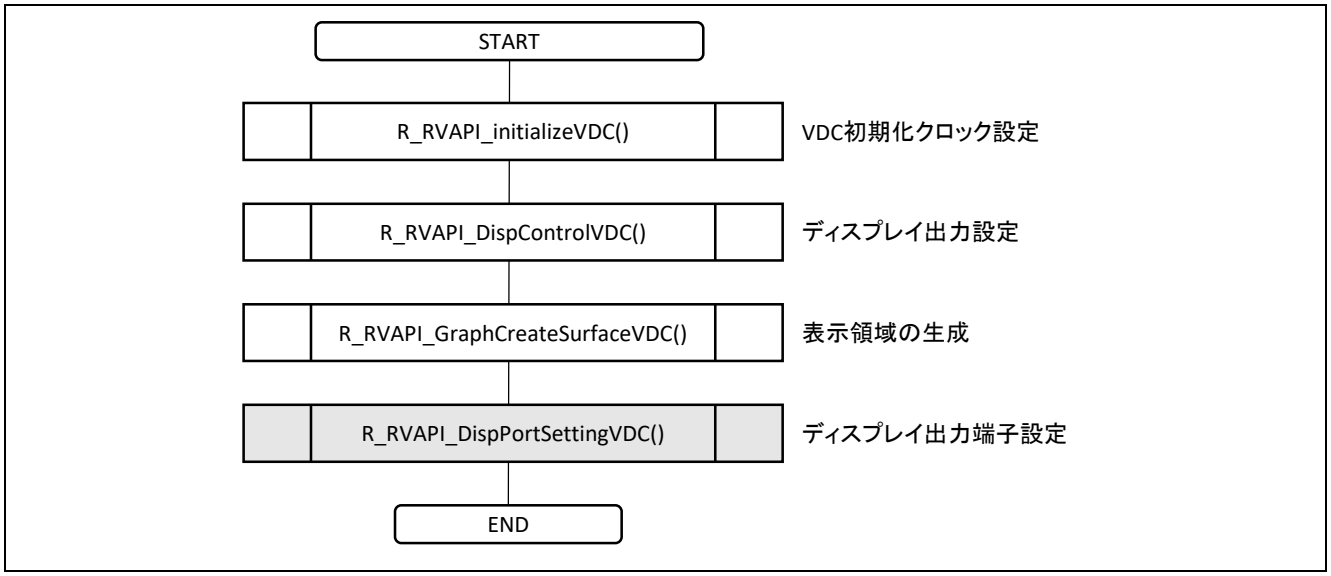

図 6-4 ディスプレイ出力端子設定を行うタイミング

6.9 R_RVAPI_VideoControlVDC

R_RVAPI_VideoControlVDC

概　要　映像入力設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_VideoControlVDC(
                const vdc_channel_t ch,
                const digital_in_t * const digital);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	digital_in_t * digital	: デジタル映像の設定 NULL を設定しないでください

リターン値	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー

備考

(1) 説明

本関数では、映像入力設定を行います。VDC では、CMOS カメラなど、デジタル映像入力の設定を行います。

本関数内では、以下のドライバを使用しています。

- R_VDC_VideoInput ()

(2) パラメータ詳細

(a) digital_in_t

digital_in_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_extin_format_t    inp_format;
    vdc_edge_t           inp_pxd_edge;
    vdc_onoff_t          inp_endian_on;
    vdc_onoff_t          inp_swap_on;
    vdc_sig_pol_t         inp_vs_inv;
    vdc_sig_pol_t         inp_hs_inv;
    vdc_extin_ref_hsync_t  inp_h_edge_sel;
    vdc_extin_input_line_t inp_f525_625;
    vdc_extin_h_pos_t     inp_h_pos;
} digital_in_t;
```

型 / メンバ名	説明
vdc_extin_format_t inp_format	外部入力のフォーマット選択 • VDC_EXTIN_FORMAT_RGB888 (0): RGB888 • VDC_EXTIN_FORMAT_RGB666 (1): RGB666 • VDC_EXTIN_FORMAT_RGB565 (2): RGB565 • VDC_EXTIN_FORMAT_BT656 (3): BT656 • VDC_EXTIN_FORMAT_BT601 (4): BT601 • VDC_EXTIN_FORMAT_YCBCR422 (5): YCbCr422 • VDC_EXTIN_FORMAT_YCBCR444 (6): YCbCr444
vdc_edge_t inp_pxd_edge	外部入力の映像信号 DV_DATA の入力段取り込みクロックのエッジ選択 • VDC_EDGE_RISING: 立ち上がりエッジ • VDC_EDGE_FALLING: 立ち下りエッジ
vdc_onoff_t inp_endian_on	外部入力のビットエンディアン変更 • VDC_OFF • VDC_ON
vdc_onoff_t inp_swap_on	外部入力の B/R 信号入れ替え • VDC_OFF • VDC_ON
vdc_sig_pol_t inp_vs_inv vdc_sig_pol_t inp_hs_inv	外部入力の垂直同期信号 DV_VSYNC / DV_HSYNC の反転制御 • VDC_SIG_POL_NOT_INVERTED: 非反転(正極性) • VDC_SIG_POL_INVERTED: 反転(負極性)
vdc_extin_ref_hsync_t inp_h_edge_sel	外部入力系統の BT656 水平同期信号の基準選択 inp_format に VDC_EXTIN_FORMAT_BT656 が指定された場合のみ有効 • VDC_EXTIN_REF_H_EAV (0): EAV 基準 • VDC_EXTIN_REF_H_SAV (1): SAV 基準
vdc_extin_input_line_t inp_f525_625	外部入力系統の BT656 入力時のライン数設定 inp_format に VDC_EXTIN_FORMAT_BT656 が指定された場合のみ有効 • VDC_EXTIN_LINE_525 (0): 525 ライン • VDC_EXTIN_LINE_625 (1): 625 ライン

vdc_extin_h_pos_t inp_h_pos	水平同期基準に対するデータ列の開始タイミング設定 inp_format に VDC_EXTIN_FORMAT_BT656、VDC_EXTIN_FORMAT_BT601 が指定された場合、以下が設定可能 • VDC_EXTIN_H_POS_CBYCRY (0): Cb/Y/Cr/Y • VDC_EXTIN_H_POS_YCRYCB (1): Y/Cr/Y/Cb • VDC_EXTIN_H_POS_CRYCBY (2): Cr/Y/Cb/Y • VDC_EXTIN_H_POS_YCBYCR (3): Y/Cb/Y/Cr 但し、inp_format に VDC_EXTIN_FORMAT_YCBCR422 が指定された場合、以下が設定可能 • VDC_EXTIN_H_POS_CBYCRY (0): Cb/Y/Cr/Y • VDC_EXTIN_H_POS_YCBYCR (3): Y/Cb/Y/Cr

6.10 R_RVAPI_VideoCreateSurfaceVDC

6.11 R_RVAPI_VideoCreateSurfaceIMRL2

R_RVAPI_VideoCreateSurfaceVDC
R_RVAPI_VideoCreateSurfaceIMRL2

概　要　映像領域と表示領域の生成
IMR-LS2 用映像表示領域生成

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_VideoCreateSurfaceVDC(
              const vdc_channel_t ch,
              const v_surface_config_t * const v_cnf,
              const v_surface_disp_config_t * const v_disp_cnf);
vdc_error_t R_RVAPI_VideoCreateSurfaceIMRL2(
              const vdc_channel_t ch,
              const v_surface_config_t * const v_cnf,
              const v_surface_disp_config_t * const v_disp_cnf);
```

引　数

[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
[IN]	v_surface_config_t * v_cnf l	: 映像入力領域の設定 映像入力を使用しない場合、NULL を設定してください
[IN]	v_surface_disp_config_t * v_g_cnf	: 映像入力領域の表示設定 表示を行なわない場合、NULL を設定してください

リターン値

VDC_OK:	: 正常終了
VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
VDC_ERR_PARAM_NULL	: NULL 指定エラー
VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
VDC_ERR_RESOURCE_LVDS_CLK	: LVDS クロックリソースエラー

備考

(1) 説明

本関数では、映像入力領域の設定として、映像取り込みタイミングやバッファへの書き込みサイズ。入力した映像の表示設定を行います。映像の取り込みのみの場合、映像入力領域の表示設定を行う必要はありません。IMR-LS2 を使用する場合、『R_RVAPI_VideoCreateSurfaceIMRL2 ()』関数を使用してください。パラメータについては、『R_RVAPI_VideoCreateSurfaceVDC ()』関数と同様になります。

本関数内では、以下のドライバを使用しています。

- R_VDC_WriteDataControl ()
- R_VDC_ReadDataControl ()
- R_VDC_StartProcess ()

(2) パラメータ詳細

(a) v_surface_config_t

v_surface_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_layer_id_t      layer_id;
    vdc_period_rect_t   cap_area;
    void              * fb_buff;
    uint32_t            fb_stride;
    uint32_t            fb_offset;
    uint32_t            fb_num;
    vdc_res_md_t        write_format;
    uint16_t            write_fb_vw;
    uint16_t            write_fb_hw;
    vdc_wr_rd_swa_t     write_swap;
    vdc_wr_md_t         write_rot;
    vdc_res_inter_t     res_inter;
} v_surface_config_t;
```

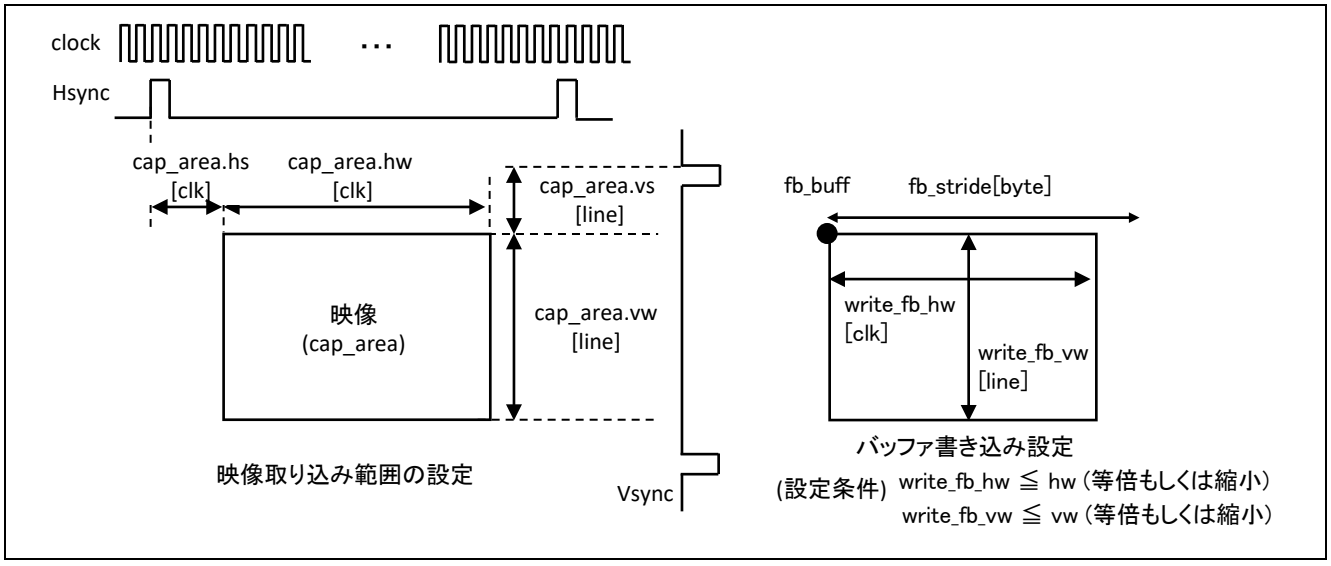

図 6-5映像入力領域パラメータ関係図

型 / メンバ名	説明
vdc_layer_id_t layer_id	レイヤ ID • VDC_LAYER_ID_0_WR
vdc_period_rect_t cap_area	映像取り込み範囲：水平[clock 単位] 垂直[line 単位](図 6-5を参照) cap_area.vs / vw ：垂直取り込み開始位置 / 垂直取り込みサイズ cap_area.hs / hw：水平取り込み開始位置 / 水平取り込みサイズ
void * fb_buff	フレームバッファのベースアドレス(図 6-5を参照) 32[byte]アライメントのアドレスを指定してください
uint32_t fb_stride	フレームバッファのラインオフセットアドレス(図 6-5を参照) 32[line]の倍数で指定してください
uint32_t fb_offset	フレームバッファのフレームオフセットアドレス このパラメータはフレーム数が 1 面(fb_num が'1')の時は無効です。 32 の倍数で指定してください。
uint32_t fb_num	書き込みフレームバッファのフレーム数 1 or 2 を設定してください
vdc_res_md_t write_format	フレームバッファ書き込み映像フォーマット · VDC_RES_MD_YCBCR422 (0): YCbCr422 · VDC_RES_MD_RGB565 (1): RGB565 · VDC_RES_MD_RGB888 (2): RGB888 · VDC_RES_MD_YCBCR444 (3): YCbCr444
uint16_t write_fb_vw	バッファ書き込み垂直サイズ[pixel 単位] 0x0000 ~ 0x07FF 4[line]アライメント且つ cap_area.res.vw 以下を指定してください 設定サイズで等倍もしくは縮小しバッファに書き込みます
uint16_t write_fb_hw	バッファ書き込み水平サイズ[clock 単位] 0x0000 ~ 0x07FF 4[pixel]素アライメント且つ cap_area.hw 以下を指定してください 設定サイズで等倍もしくは縮小しバッファに書き込みます
vdc_wr_rd_swa_t write_swap	8 ビット/16 ビット/32 ビットスワップ設定(注 1) • VDC_WR_RD_WRSWA_NON (0): スワップなし 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit スワップ 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2): 16-bit スワップ 3-4-1-2-7-8-5-6 • VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit スワップ 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit スワップ 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit スワップ 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit スワップ 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 32-bit + 16-bit + 8-bit スワップ 8-7-6-5-4-3-2-1
vdc_wr_md_t write_rot	フレームバッファ書き込み動作モード • VDC_WR_MD_NORMAL (0): 通常書き込み • VDC_WR_MD_MIRROR (1): 水平鏡像書き込み • VDC_WR_MD_ROT_90DEG (2): 90 度回転書き込み • VDC_WR_MD_ROT_180DEG (3): 180 度回転書き込み • VDC_WR_MD_ROT_270DEG (4): 270 度回転書き込み

	90 度、180 度、270 度回転書き込みは、フレームバッファ書き込み映像フォーマット(write_format)が YCbCr422 か RGB565 の時のみ有効です。
vdc_res_inter_t res_inter	フィールド動作モード設定 • VDC_RES_INTER_PROGRESSIVE (0) :プログレッシブ • VDC_RES_INTER_INTERLACE (1): インタレース

【注 1】write_format に YCbCr422、RGB565 を選択した時、必ず(0)[スワップなし]を設定してください

(b) v_surface_disp_config_t

v_surface_disp_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_period_rect_t   disp_area;
    vdc_gr_ycc_swap_t   read_ycc_swap;
    vdc_wr_rd_swa_t     read_swap;
} v_surface_disp_config_t;
```

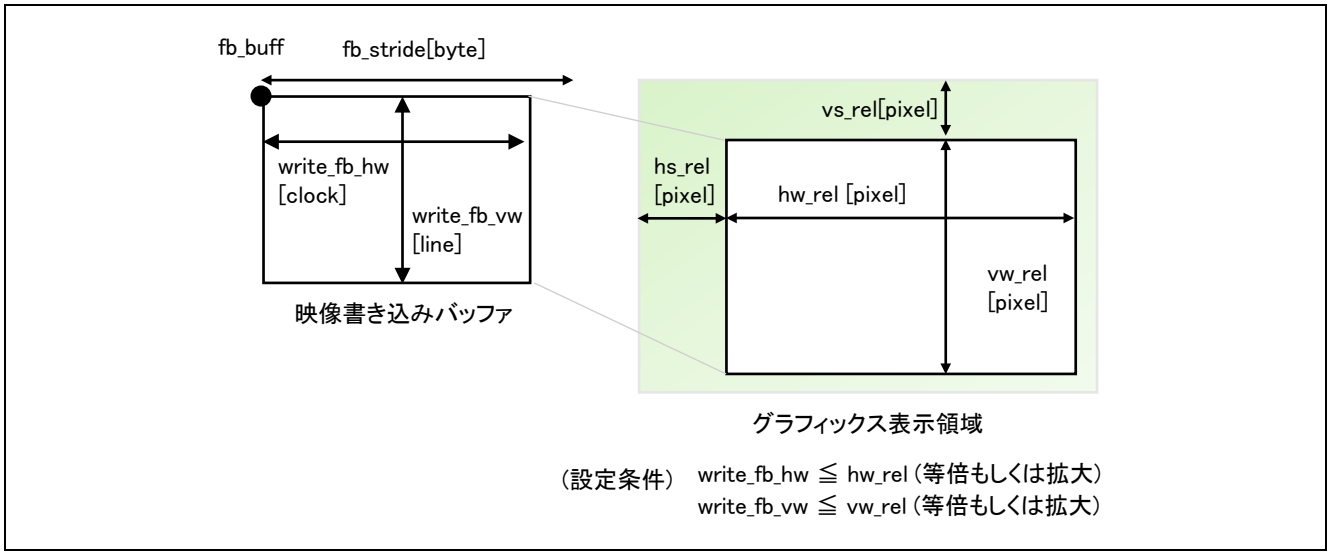

図 6-6 映像入力領域の表示パラメータ関係図

型 / メンバ名	説明
vdc_pd_disp_rect_t disp_area	グラフィックス表示領域[pixel 単位] (図 6-6を参照) • disp_area.vs_rel / vw_rel：垂直表示開始位置 / 垂直表示サイズ • disp_area.hs_rel / hw_rel：水平表示開始位置 / 水平表示サイズ
vdc_gr_ycc_swap_t read_ycc_swap	YCbCr422 フォーマット時バッファ読み出しデータのスワップ制御 このパラメータは read_format に指定された値が VDC_GR_FORMAT_YCBCR422 の場合のみ有効です • VDC_GR_YCCSWAP_CBY0CRY1 (0): CbY0/Cr/Y1 • VDC_GR_YCCSWAP_Y0CBY1CR (1): Y0/Cb/Y1/Cr • VDC_GR_YCCSWAP_CRY0CBY1 (2): Cr/Y0/Cb/Y1 • VDC_GR_YCCSWAP_Y0CRY1CB (3): Y0/Cr/Y1/Cb • VDC_GR_YCCSWAP_Y1CRY0CB (4): Y1/Cr/Y0/Cb • VDC_GR_YCCSWAP_CRY1CBY0 (5): Cr/Y1/Cb/Y0 • VDC_GR_YCCSWAP_Y1CBY0CR (6): Y1/Cb/Y0/Cr • VDC_GR_YCCSWAP_CBY1CRY0 (7): Cb/Y1/Cr/Y0
vdc_wr_rd_swa_t read_swap	8 ビット/16 ビット/32 ビットスワップ設定 • VDC_WR_RD_WRSWA_NON (0): スワップなし 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit スワップ 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2): 16-bit スワップ 3-4-1-2-7-8-5-6 • VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit スワップ 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit スワップ 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit スワップ 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit スワップ 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 32-bit + 16-bit + 8-bit スワップ 8-7-6-5-4-3-2-1

(3) 映像取り込み範囲の設定について

映像取り込み範囲の設定例を表 6-3に記載します。

(デジタル入力例)

・VGA(640 x 480)サイズのプログレッシブ入力

・YCbCr422 フォーマットで、縮小せず VGA(640 x 480)サイズでバッファに書き込み

・表示サイズは、VGA(640 x 480) → SVGA(800 x 600)に拡大

(アナログ入力例)

・NTSC/PAL を入力し表示サイズは、IP 変換して SVGA(800 x 600)で表示

表 6-3 映像取り込み範囲の設定例

構造体名	メンバ名	デジタル入力 24/18/16 bit I/F	デジタル入力 8bit I/F
digital_in_t	inp_format	RGB888/666/565 YCbCr422/444	BT6556 BT601
v_surface _config_t	layer_id	VDC_LAYER_ID_0_WR	
	cap_area.vs	任意	
	cap_area vw	480u	
	cap_area.hs	任意	
	cap_area.hw	640u x 1u 1[pixel] / 1[clock]	640u x 2u(注 1) 1[pixel] / 2[clock]
	fb_buff	内蔵 RAM 領域	
	fb_stride	640u x 2u(YCbCr422 より)	
	fb_num	2 面	
	write_format	YCbCr422	
	write_fb_vw	480u	
	write_fb_hw	640u	640u (注 2)
	res_inter	プログレッシブ	
	fb_offset	バッファオフセット	
v_surface _disp_config_t	disp_area.vs_rel	0u	
	disp_area.vw_rel	800u (等倍なら 640u)	
	disp_area.hs_rel	0u	
	disp_area.hw_rel	600u (等倍なら 480u)	

【注 1】外部入力の I/F によって、取り込み幅のクロックが異なる(1[pixel] / 1[clock] や 1[pixel] / 2[clock])

【注 2】VDC の仕様より BT.656/601 の時、同じ映像データを 2 回取り込む為、水平縮小が必要
取り込み幅のクロック(cap_area.hw=640u x 2u)に対して、バッファ書き込み設定(write_fb_hw)に半分の 640u で設定

(4) IMR を使用する場合の映像取り込み範囲の設定について

IMR-LS2 を使用する場合、『6.11 R_RVAPI_VideoCreateSurfaceIMRL2()』を使用し VDC の設定を行なう。パラメータの項目については、『6.10 R_RVAPI_VideoCreateSurfaceVDC()』と同じであるが、参照しない項目がある為、その内容について表 6-4 に記載する。

表 6-4 IMR を使用する場合のパラメータについて

構造体名	メンバ名	IMR-LS2
v_surface _config_t	layer_id	未使用
	cap_area.vs	映像入力の取り込み位置及び取り込みサイズ
	cap_area.hs	
	cap_area.vw	
	cap_area.hw	
	fb_buff	IMR-LS2 の設定に合わせて設定
	fb_stride	
	fb_offset	
	fb_num	
	write_format	未使用
	write_fb_vw	IMR-LS2 に入力する映像高さと幅
	write_fb_hw	
	res_inter	映像入力に合わせて選択 プログレッシブ/インタレース
v_surface_ disp_config_t	disp_area.vs_rel	表示サイズに合わせて設定
	disp_area.vw_rel	
	disp_area.hs_rel	
	disp_area.hw_rel	

6.12 R_RVAPI_VideoDestroySurfaceVDC

R_RVAPI_VideoDestroySurfaceVDC

概　要　映像領域と表示領域の破棄

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_VideoDestroySurfaceVDC (
                const vdc_channel_t ch,
                const vdc_layer_id_t layer_id);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id	: レイヤ ID • VDC_LAYER_ID_0_WR

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
	VDC_ERR_RESOURCE_LVDS_CLK	: LVDS クロックリソースエラー

備考

(1) 説明

本関数では、指定されたレイヤの停止処理を行います。フレームバッファからの読み出しを停止し、レイヤのグラフィックス表示設定を初期状態に戻します。

本関数内では、以下のドライバを使用しています。

- R_VDC_StopProcess ()
- R_VDC_ReleaseDataControl ()

6.13 R_RVAPI_VideoPortSettingVDC

R_RVAPI_VideoPortSettingVDC

概　要　　映像入力端子設定

ヘッダ　　r_rvapi_vdc.h

宣　言

```
void R_RVAPI_VideoPortSettingVDC(
                const vdc_channel_t ch,
                void (* const port_func)(uint32_t));
```

引　数	[IN]	vdc_channel_t ch	：VDC チャネル • VDC_CHANNEL_0
	[IN]	void (* const port_func) (uint32_t)	：映像入力端子の設定を行う関数ポインタ

リターン値　なし

備考

(1) 説明

本関数に設定するコールバック関数では、映像入力端子の設定を行なってください。本関数は、図 6-7に示すように映像領域の生成を行なう前までに呼び出してください。

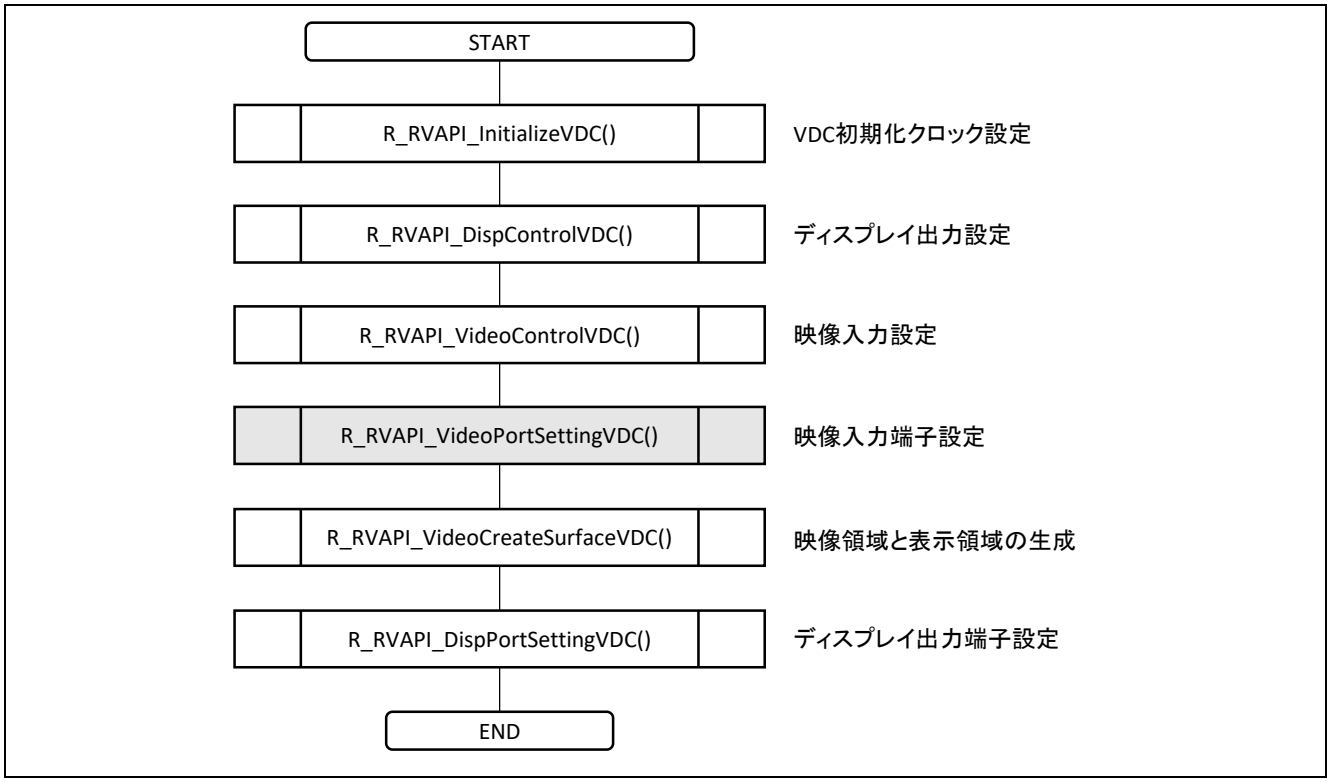

図 6-7　映像入力端子設定を行うタイミング

6.14 R_RVAPI_InterruptEnableVDC

R_RVAPI_InterruptEnableVDC

概　要　VDC 割り込み許可設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_InterruptEnableVDC(
                const vdc_channel_t ch,
                const vdc_int_type_t flag,
                const uint16_t line_num,
                void (* const callback)
                      (vdc_int_type_t int_type));
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_int_type_t flag	: VDC 割り込みタイプ
	[IN]	uint16_t line_num	: ライン割り込み設定 VDC_INT_TYPE_VLINE の時のみ有効
	[IN]	void (*callback) (vdc_int_type_t, void * buff)	: 割り込みコールバック関数ポインタ

リターン値	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_RESOURCE_CLK:	: クロックリソースエラー
	VDC_ERR_RESOURCE_VSYNC	: 垂直同期信号リソースエラー

備考

(1) 説明

本関数では、表 6-5に記載される VDC 割り込みタイプで指定された割り込みを許可し、指定された割り込みコールバック関数の登録を行ないます。

本関数内では、以下のドライバを使用しています。

- R_VDC_CallbackISR ()

(2) パラメータ詳細

VDC 割り込みタイプについては、表 6-5に記載します。

表 6-5 VDC 割り込みタイプ

列挙定数	値	説明
VDC_INT_TYPE_S0_VI_VSYNC	0	スケーリング 0 に入力される垂直同期信号
VDC_INT_TYPE_S0_LO_VSYNC	1	スケーリング 0 から出力される垂直同期信号
VDC_INT_TYPE_S0_VSYNCERR	2	スケーリング 0 の垂直同期信号の欠落信号
VDC_INT_TYPE_VLINE	3	グラフィックス(3)パネル出力の指定ライン信号
VDC_INT_TYPE_S0_VFIELD	4	スケーリング 0 の録画機能のフィールド終了信号
VDC_INT_TYPE_IV1_VBUFERR	5	スケーリング 0 のフレームバッファ書き込みオーバフロー信号
VDC_INT_TYPE_IV3_VBUFERR	6	グラフィックス(0)フレームバッファ読み出しアンダフロー信号
VDC_INT_TYPE_IV5_VBUFERR	7	グラフィックス(2)フレームバッファ読み出しアンダフロー信号
VDC_INT_TYPE_IV6_VBUFERR	8	グラフィックス(3)フレームバッファ読み出しアンダフロー信号

6.15 R_RVAPI_InterruptDisableVDC

R_RVAPI_InterruptDisableVDC

概　要　VDC 割り込み禁止設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_InterruptDisableVDC(
            const vdc_channel_t ch,
            const vdc_int_type_t flag);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_int_type_t flag	: VDC 割り込みタイプ
リターン値	VDC_OK:		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
	VDC_ERR_PARAM_NULL		: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH		: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED		: 未定義パラメータ指定エラー
	VDC_ERR_RESOURCE_CLK		: クロックリソースエラー
	VDC_ERR_RESOURCE_VSYNC		: 垂直同期信号リソースエラー

備考

(1) 説明

本関数では、表 6-5に記載される VDC 割り込みタイプで指定された割り込みを禁止します。

本関数内では、以下のドライバを使用しています。

- R_VDC_CallbackISR ()

6.16 R_RVAPI_AlphablendingRectVDC

R_RVAPI_AlphablendingRectVDC

概　要　矩形アルファブレンド

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_AlphablendingRectVDC(
            const vdc_channel_t ch,
            const vdc_layer_id_t layer_id,
            const vdc_onoff_t alpha_onoff,
            const vdc_pd_disp_rect_t * const alpha_area,
            const uint8_t alpha_value);
```

引　数	[IN]	vdc_channel_t ch	：VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id,	：レイヤ ID • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	vdc_onoff_t alpha_onoff	：矩形アルファブレンドの ON/OFF 設定 • VDC_ON • VDC_OFF
	[IN]	vdc_pd_disp_rect_t * alpha_area	：矩形アルファブレンド領域[pixel 単位]
	[IN]	uint8_t alpha_value	：アルファ値（0 ~ 255）0：完全透過
リターン値	VDC_OK:		：正常終了
	VDC_ERR_PARAM_CHANNEL		：チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID		：レイヤ ID 不正エラー
	VDC_ERR_PARAM_BIT_WIDTH		：ビット幅エラー
	VDC_ERR_PARAM_EXCEED_RANGE		：設定範囲外エラー
	VDC_ERR_IF_CONDITION		：インタフェース条件エラー
	VDC_ERR_RESOURCE_LAYER		：レイヤリソースエラー

備考

(1)　説明

本関数では、矩形領域アルファブレンディングの ON/OFF、矩形領域設定、アルファ値設定を行ないます。本関数内では、以下のドライバを使用しています。

- R_VDC_AlphaBlendingRect ()

6.17 R_RVAPI_ChromakeyVDC

R_RVAPI_ChromakeyVDC

概　要　クロマキーを使用した透過

ヘッダ　r_vapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_ChromakeyVDC(
            const vdc_channel_t ch,
            const vdc_layer_id_t layer_id,
            const vdc_onoff_t gr_ck_on,
            const uint32_t ck_color,
            const uint8_t rep_alpha);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id,	: レイヤ ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	vdc_onoff_t gr_ck_on	: クロマキーON/OFF 設定 • VDC_ON • VDC_OFF
	[IN]	uint32_t ck_color	: クロマキー対象の色信号 ターゲットとなるレイヤで使用されているカラーフォーマットで指定して下さい(LSB 詰め)
	[IN]	uint8_t rep_alpha	: クロマキー置換後アルファ値 (0 ~ 255)
リターン値	VDC_OK:		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID		: レイヤ ID 不正エラー
	VDC_ERR_PARAM_BIT_WIDTH		: ビット幅エラー
	VDC_ERR_IF_CONDITION		: インタフェース条件エラー
	VDC_ERR_RESOURCE_LAYER		: レイヤリソースエラー

備考

(1) 説明

本関数では、クロマキーの ON/OFF 設定、クロマキー対象の色信号、及び置換後のアルファ値設定を行ないます。本関数内では、以下のドライバを使用しています。

- R_VDC_Chromakey ()

6.18 R_RVAPI_DispCalibrationVDC

R_RVAPI_DispCalibrationVDC

概　要　画面出力較正処理

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_DispCalibrationVDC(
            const vdc_channel_t ch,
            const vdc_calibr_route_t route,
            const vdc_calibr_bright_t * const bright,
            const vdc_calibr_contrast_t * const contrast,
            const vdc_calibr_dither_t * const panel_dither);
```

引　数

[IN]	vdc_channel_t ch		: VDC チャネル • VDC_CHANNEL_0
[IN]	vdc_calibr_route_t route		: 補正回路の順番の制御 • VDC_CALIBR_ROUTE_BCG ブライト ⇒ コントラスト⇒ ガンマ補正 • VDC_CALIBR_ROUTE_GBC ガンマ補正⇒ ブライト⇒ コントラスト
[IN]	vdc_calibr_bright_t	* bright	: ブライト(DC)調整パラメータ 変更する必要が無い場合は NULL を指定してください
[IN]	vdc_calibr_contrast_t	* contrast	: コントラスト(ゲイン)調整パラメータ 変更する必要が無い場合は NULL を指定してください
[IN]	vdc_calibr_dither_t	* panel_dither	: パネルディザパラメータ 変更する必要が無い場合は NULL を指定してください

リターン値

VDC_OK:	: 正常終了
VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
VDC_ERR_PARAM_NULL	: NULL 指定エラー
VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
VDC_ERR_RESOURCE_OUTPUT	: 出力リソースエラー

備考

(1) 説明

本関数では、パネルブライト設定、コントラスト調整設定、パネルディザ設定、パネル出力補正回路の制御設定を行ないます。本関数による設定は、ハードウェアのリセットか、本関数による別の設定で上書きされるまでは有効です。

本関数内では、以下のドライバを使用しています。

- R_VDC_DisplayCalibration ()

(2) パラメータ詳細

(a) vdc_calibr_bright_t

vdc_calibr_bright_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t    pbrt_g;
    uint16_t    pbrt_b;
    uint16_t    pbrt_r;
} vdc_calibr_bright_t;
```

型 / メンバ名	初期値	説明
uint16_t pbrt_g	512	G 信号のブライト(DC)調整 0x0000 (-512) ~ 0x03FF (+511)
uint16_t pbrt_b	512	B 信号のブライト(DC)調整 0x0000 (-512) ~ 0x03FF (+511)
uint16_t pbrt_r	512	R 信号のブライト(DC)調整 0x0000 (-512) ~ 0x03FF (+511)

(b) vdc_calibr_contrast_t

vdc_calibr_contrast_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint8_t     cont_g;
    uint8_t     cont_b;
    uint8_t     cont_r;
} vdc_calibr_contrast_t;
```

型 / メンバ名	初期値	説明
uint8_t cont_g	128	G 信号のコントラスト(ゲイン)調整 0x0000 (0/128[倍]) ~ 0x00FF (255/128[倍])
uint8_t cont_b	128	B 信号のコントラスト(ゲイン)調整 0x0000 (0/128[倍]) ~ 0x00FF (255/128[倍])
uint8_t cont_r	128	R 信号のコントラスト(ゲイン)調整 0x0000 (0/128[倍]) ~ 0x00FF (255/128[倍])

(c) vdc_calibr_dither_t

vdc_calibr_dither_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_panel_dither_md_t  pdth_sel;
    uint8_t                pdth_pa;
    uint8_t                pdth_pb;
    uint8_t                pdth_pc;
    uint8_t                pdth_pd;
} vdc_calibr_dither_t;
```

型 / メンバ名	初期値	説明
vdc_panel_dither_md_t pdth_sel	0	パネルディザ動作モード • VDC_PDTH_MD_TRU (0): 切り捨て • VDC_PDTH_MD_RDOF (1): 四捨五入 • VDC_PDTH_MD_2X2 (2): 2x2 パターンディザ • VDC_PDTH_MD_RAND (3): ランダムパターンディザ
uint8_t pdth_pa	3	2x2 パターンディザのパターン値(A) 0 ~ 3 pdth_sel に VDC_PDTH_MD_2X2 が指定された場合のみ参照されます
uint8_t pdth_pb	0	2x2 パターンディザのパターン値(B) 0 ~ 3 pdth_sel に VDC_PDTH_MD_2X2 が指定された場合のみ参照されます
uint8_t pdth_pc	2	2x2 パターンディザのパターン値(C) 0 ~ 3 pdth_sel に VDC_PDTH_MD_2X2 が指定された場合のみ参照されます
uint8_t pdth_pd	1	2x2 パターンディザのパターン値(D) 0 ~ 3 pdth_sel に VDC_PDTH_MD_2X2 が指定された場合のみ参照されます

6.19 R_RVAPI_DispGammaVDC

R_RVAPI_DispGammaVDC

概　要　ガンマ補正設定

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_DispGammaVDC(
                const vdc_channel_t ch,
                const vdc_onoff_t gam_on,
                const uint16_t * const gam_r_gain,
                const uint8_t * const gam_r_th,
                const uint16_t * const gam_g_gain,
                const uint8_t * const gam_g_th,
                const uint16_t * const gam_b_gain,
                const uint8_t * const gam_b_th);
```

引　数				
	[IN]	vdc_channel_t	ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_onoff_t	gam_on	: ガンマ補正 ON/OFF 設定 • VDC_ON • VDC_OFF
	[IN]	uint16_t	* gam_r_gain,	: R 信号の領域 0 ~ 31 のゲイン調整 符号無し (0 ~ 2047[LSB], 1024[LSB] = 1.0[倍])
	[IN]	uint8_t	* gam_r_th	: R 信号の領域 1 ~ 31 の開始閾値 符号無し (0 ~ 255[LSB])
	[IN]	uint16_t	* gam_g_gain	: G 信号の領域 0 ~ 31 のゲイン調整 符号無し (0 ~ 2047[LSB], 1024[LSB] = 1.0[倍])
	[IN]	uint8_t	* gam_g_th	: G 信号の領域 1 ~ 31 の開始閾値 符号無し (0 ~ 255[LSB])
	[IN]	uint16_t	* gam_b_gain	: B 信号の領域 0 ~ 31 のゲイン調整 符号無し (0 ~ 2047[LSB], 1024[LSB] = 1.0[倍])
	[IN]	uint8_t	* gam_b_th	: B 信号の領域 1 ~ 31 の開始閾値 符号無し (0 ~ 255[LSB])

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_RESOURCE_OUTPUT	: 出力リソースエラー

備考

(1) 説明

本関数では、ガンマ補正の ON/OFF 設定、G/B/R 信号のガンマ補正ゲイン調整値設定、G/B/R 信号のガンマ補正開始閾値設定を行なっています。ガンマ補正処理は、ガンマ補正のパラメータ設定と ON/OFF の制御を別個に設定できます。一度設定されたガンマ補正のパラメータは、ハードウェアのリセットか、別の設定で上書きされるまでは有効です。

本関数内では、以下のドライバを使用しています。

- R_VDC_GammaCorrection ()

6.20 R_RVAPI_VideoCalibrationVDC

R_RVAPI_VideoCalibrationVDC

概　要	カラーマトリクス設定		
ヘッダ	r_rvapi_vdc.h		
宣　言	vdc_error_t R_RVAPI_VideoCalibrationVDC(const vdc_channel_t ch, const vdc_color_matrix_t * const color_matrix);		
引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_color_matrix_t * color_matrix	: カラーマトリクス設定パラメータ
リターン値	VDC_OK:		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
	VDC_ERR_PARAM_NULL		: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH		: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED		: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_CONDITION		: 不許可条件エラー
	VDC_ERR_RESOURCE_LAYER		: レイヤリソースエラー

備考

(1) 説明

本関数では、指定されたカラーマトリクスの設定を行います。このカラーマトリクスを使用し映像入力のコントラスト及びブライトネスの調整を行ないます。

本関数内では、以下のドライバを使用しています。

- R_VDC_ImageColorMatrix ()

(2) パラメータ詳細

(a) vdc_color_matrix_t

vdc_color_matrix_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_colormtx_module_t  module;
    vdc_colormtx_mode_t    mtx_mode;
    uint16_t              offset[VDC_COLORMTX_OFFST_NUM];
    uint16_t              gain[VDC_COLORMTX_GAIN_NUM];
} vdc_color_matrix_t;
```

型 / メンバ名	説明
vdc_colormtx_module_t module	カラーマトリクス設定対象モジュール選択 • VDC_COLORMTX_IMGCNT (0): 入力制御部 • VDC_COLORMTX_ADJ_0 (1): 画質改善部 0
vdc_colormtx_mode_t mtx_mode	カラーマトリクス動作モード • VDC_COLORMTX_GBR_GBR: GBR ⇒ GBR • VDC_COLORMTX_GBR_YCBCR: GBR ⇒ YCbCr (注 1) • VDC_COLORMTX_YCBCR_GBR: YCbCr ⇒ GBR • VDC_COLORMTX_YCBCR_YCBCR: YCbCr ⇒ YCbCr (注 1)
uint16_t offset[VDC_COLORMTX_OFFST_NUM]	Y/G、B、R 信号のオフセット(DC)調整 0x0000 (-128) ~ 0x0080 (0) ~ 0x00FF (+127)
uint16_t gain[VDC_COLORMTX_GAIN_NUM]	GG、GB、GR、BG、BB、BR、RG、RB、RR のゲイン調整 符号付 (2 の補数) -1024 ~ +1023[LSB], 256[LSB] = 1.0[倍]

【注 1】YCbCr へ変換する動作モードは、module に入力制御部 (VDC_COLORMTX_IMGCNT)を指定した場合のみ使用可能

6.21 R_RVAPI_VideoSharpnessLtiVDC

R_RVAPI_VideoSharpnessLtiVDC

概　要　画質改善設定処理

ヘッダ　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_VideoSharpnessLtiVDC(
            const vdc_channel_t ch,
            const vdc_imgimprv_id_t imgimprv_id,
            const vdc_onoff_t shp_h_on,
            const vdc_enhance_sharp_t * const sharp_param,
            const vdc_onoff_t lti_h_on,
            const vdc_enhance_lti_t * const lti_param,
            const vdc_period_rect_t * const enh_area);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_imgimprv_id_t imgimprv_id	: 画質改善部 ID • VDC_IMG_IMPRV_0 : 画質改善部 0
	[IN]	vdc_onoff_t shp_h_on	: シャープネス ON/OFF 設定
	[IN]	vdc_enhance_sharp_t * sharp_param	: シャープネス設定パラメータ
	[IN]	vdc_onoff_t lti_h_on	: LTI ON/OFF 設定
	[IN]	vdc_enhance_lti_t * lti_param	: LTI 設定パラメータ
	[IN]	vdc_period_rect_t * enh_area	: 画質改善領域設定パラメータ

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_IF_CONDITION	: インタフェース条件エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数では、シャープネスの ON/OFF 設定、シャープネスのパラメータ設定、LTI の ON/OFF 設定、LTI のパラメータ設定、シャープネスと LTI の適用される矩形領域設定を行ないます。

本関数内では、以下のドライバを使用しています。

- R_VDC_ImageEnhancement ()

(2) パラメータ詳細

(a) vdc_enhance_sharp_t

vdc_enhance_sharp_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_onoff_t           shp_h2_lpf_sel;
    vdc_sharpness_ctrl_t   hrz_sharp[VDC_IMGENH_SHARP_NUM];
} vdc_enhance_sharp_t;
```

型 / メンバ名	初期値	説明
vdc_onoff_t shp_h2_lpf_sel	VDC_OFF (0)	H2 エッジ検出前の折り返し除去用 LPF 選択 • VDC_OFF: LPF なし • VDC_ON: LPF あり
vdc_sharpness_ctrl_t hrz_sharp [VDC_IMGENH_SHARP_NUM]	-	シャープネス制御パラメータ 水平シャープネス (H1、H2、H3)

(b) vdc_sharpness_ctrl_t

vdc_sharpness_ctrl_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint8_t    shp_clip_o;
    uint8_t    shp_clip_u;
    uint8_t    shp_gain_o;
    uint8_t    shp_gain_u;
    uint8_t    shp_core;
} vdc_sharpness_ctrl_t;
```

型 / メンバ名	初期値	説明
uint8_t shp_clip_o	0	シャープネスの補正値クリップ (オーバーシュート側) 0x0000 ~ 0x00FF
uint8_t shp_clip_u	0	シャープネスの補正値クリップ (アンダーシュート側) 0x0000 ~ 0x00FF
uint8_t shp_gain_o	0	シャープネスのエッジ振幅値に対するゲイン設定 (オーバーシュート側) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
uint8_t shp_gain_u	0	シャープネスのエッジ振幅値に対するゲイン設定 (アンダーシュート側) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
uint8_t shp_core	0	シャープネスの能動範囲の指定 0x0000 ~ 0x007F

(c) vdc_enhance_lti_t

vdc_enhance_lti_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    vdc_onoff_t           lti_h2_lpf_sel;
    vdc_lti_mdfil_sel_t    lti_h4_median_tap_sel;
    vdc_lti_ctrl_t        lti[VDC_IMGENH_LTI_NUM];
} vdc_enhance_lti_t;
```

型 / メンバ名	初期値	説明
vdc_onoff_t lti_h2_lpf_sel	VDC_OFF(0)	H2 エッジ検出前の折り返し除去用 LPF 選択 • VDC_OFF: LPF なし • VDC_ON: LPF あり
vdc_lti_mdfil_sel_t lti_h4_median_tap_sel	0	メディアンフィルタの参照画素選択 • VDC_LTI_MDFIL_SEL_ADJ2 (0): 隣接 2 画素目参照 • VDC_LTI_MDFIL_SEL_ADJ1 (1): 隣接 1 画素目参照
vdc_lti_ctrl_t lti[VDC_IMGENH_LTI_NUM]	-	LTI 制御パラメータ 水平 LTI (H2、H4)

(d) vdc_lti_ctrl_t

vdc_lti_ctrl_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint8_t    lti_inc_zero;
    uint8_t    lti_gain;
    uint8_t    lti_core;
} vdc_lti_ctrl_t;
```

型 / メンバ名	初期値	説明
uint8_t lti_inc_zero	10	メディアンフィルタの LTI 補正スレッシュ設定 0x0000 ~ 0x00FF
uint8_t lti_gain	0	LTI のエッジ振幅値に対するゲイン設定 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
uint8_t lti_core	0	LTI のコアリング 0x0000 ~ 0x00FF

(e) vdc_period_rect_t

vdc_period_rect_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t    vs;
    uint16_t    vw;
    uint16_t    hs;
    uint16_t    hw;
} vdc_period_rect_t;
```

型 / メンバ名		初期値	説明
uint16_t	vs	0	エンハンサ有効領域の垂直有効画像領域の開始位置設定 [line 単位] 2[line]ライン以上の設定にしてください。
uint16_t	vw	0	エンハンサ有効領域の垂直有効画像領域の幅設定 [line 単位]
uint16_t	hs	0	エンハンサ有効領域の水平有効画像領域の開始位置設定 [clk 単位] 4 クロック以上の設定にしてください。
uint16_t	hw	0	エンハンサ有効領域の水平有効画像領域の幅設定 [clk 単位]

6.22 R_RVAPI_AlphablendingVDC

R_RVAPI_AlphablendingVDC

概　要　　1 ビットアルファブレンド設定

ヘッダ　　r_rvapi_vdc.h

宣　言

```
vdc_error_t R_RVAPI_AlphablendingVDC(
            const vdc_channel_t ch,
            const vdc_layer_id_t layer_id,
             uint8_t alpha_value0,
            uint8_t alpha_value1);
```

引数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	vdc_layer_id_t layer_id	: レイヤ ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	uint8_t alpha_value0	: ARGB1555/RGBA5551 フォーマットのα信号 0 ~ 255
	[IN]	uint8_t alpha_value1	: ARGB1555/RGBA5551 フォーマットのα信号 0 ~ 255

リターン値	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	: レイヤ ID 不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数内では、以下のドライバを使用しています。

- R_VDC_AlphaBlending ()

7. 関数リファレンス(CEU)

7.1 R_RVAPI_InitializeCEU

R_RVAPI_InitializeCEU	
概　要	CEU 初期化設定
ヘッダ	r_rvapi_ceu.h
宣　言	void R_RVAPI_InitializeCEU(void);
引　数	[IN]　なし　　:
リターン値	なし
備考	

(1) 説明

本関数では､CEU のスタンバイ解除及び割り込みの許可設定､割り込みハンドラの設定を行なっています。

本関数内では、以下のドライバを使用しています。

- R_CEU_Initialize ()

7.2 R_RVAPI_TerminateCEU

R_RVAPI_TerminateCEU	
概　要	CEU 終了設定
ヘッダ	r_rvapi_ceu.h
宣　言	void R_RVAPI_TerminateCEU(void);
引　数	[IN]　なし　　:
リターン値	なし
備考	

(1) 説明

本関数では､CEU のスタンバイ設定及び割り込みの禁止設定､割り込みハンドラの解除を行なっています。

本関数内では、以下のドライバを使用しています。

- R_CEU_InterruptDisable ()
- R_CEU_Terminate ()

7.3 R_RVAPI_PortSettingCEU

R_RVAPI_PortSettingCEU	
概　要	映像入力端子設定
ヘッダ	r_rvapi_ceu.h
宣　言	void R_RVAPI_PortSettingCEU(void (* const port_func)(uint32_t));
引　数	[IN] void (* const port_func) (uint32_t) ：映像入力端子の設定を行う関数ポインタ
リターン値	なし
備考	

(1) 説明

本関数に設定するコールバック関数では、CEU で映像取り込みを行う為に必要な端子設定を行なってください。本関数は、図 7-1に示すように CEU の取り込み開始を行なう前までに呼び出してください。

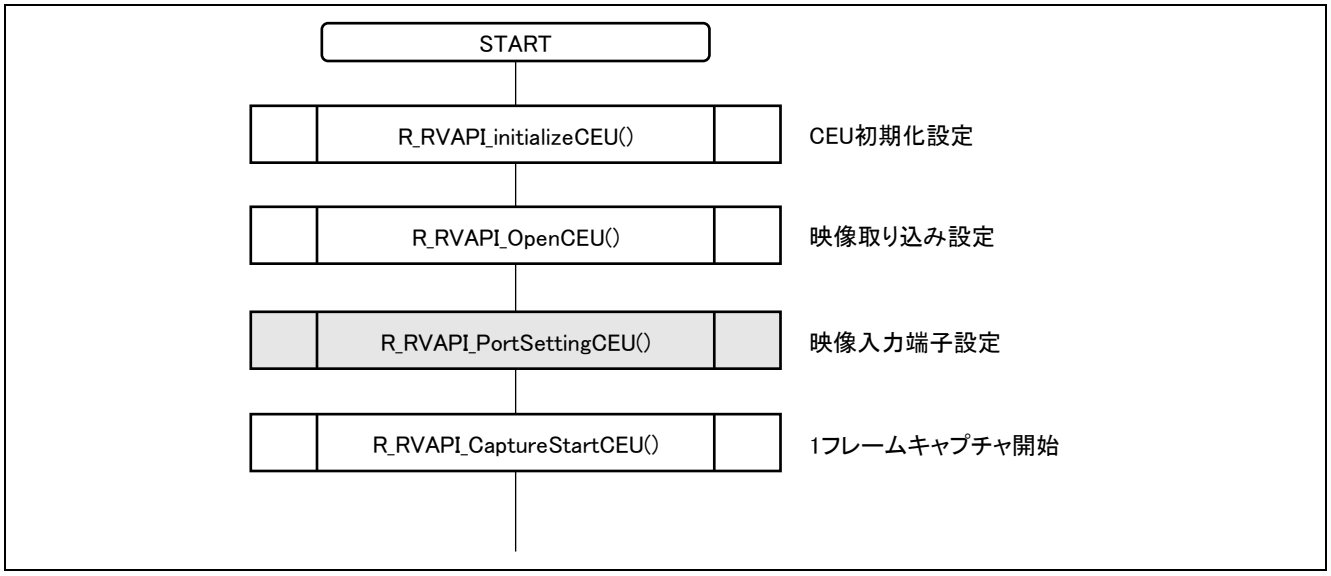

図 7-1 CEU の映像入力端子設定を行うタイミング

7.4 R_RVAPI_OpenCEU

R_RVAPI_OpenCEU			
概　要	映像取り込み設定		
ヘッダ	r_rvapi_ceu.h		
宣　言	ceu_error_t R_RVAPI_OpenCEU(const ceu_config_t * const config);		
引　数	[IN]	ceu_config_t * config	: コンフィグレーション NULL は設定しないでください
リターン値	CEU_OK		: 正常終了
	CEU_ERR_PARAM		: config 、cap の設定が NULL、cap、clp の設定が範囲外
備考			

(1) 説明

CEU の取り込みモードの選択や取り込みサイズの設定、外部モジュールとのインタフェース設定を行います。取り込みモードの選択によって、設定不要なパラメータがあります。表 7-1 に設定が不要なパラメータを記載します。

表 7-1　取り込みモードの選択によって設定が不要なパラメータ

取り込みモードの選択 ceu_jpg_t jpg	画像 取り込みモード	データ同期 取り込みモード	データイネーブル 取り込みモード
ceu_dtif_t dtif	○	○	○
ceu_sig_pol_t vdpol	○	○	設定不要(正極性固定)
ceu_sig_pol_t hdpol	○	○	設定不要(正極性固定)
ceu_dtary_t dtary	○	○(注 1)	○(注 1)
ceu_edge_t dsel	○	○	○
ceu_edge_t fldsel	○	○	○
ceu_edge_t hdsel	○	○	○
ceu_edge_t vdsel	○	○	○
ceu_cap_rect_t * cap	○	○	設定不要
ceu_clp_t * clp	○	設定不要(注 2)	設定不要
ceu_onoff_t cols/ cows/ cobs	○	○	○

【注 1】ドライバで CEU_CB0_Y0_CR0_Y1 を設定してください
【注 2】8 ビットインタフェースでは、vfclp = vwdth、hfclp = hwdth/2、
16 ビットインタフェースでは、vfclp = vwdth、hfclp = hwdth をドライバで設定します

本関数内では、以下のドライバを使用しています。

- R_CEU_Open ()

(2) パラメータ詳細

(a) ceu_config_t

ceu_config_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    ceu_jpg_t          jpg;
    ceu_dtif_t         dtif;
    ceu_sig_pol_t      vdpol;
    ceu_sig_pol_t      hdpol;
    ceu_dtary_t        dtary;
    ceu_edge_t         dsel;
    ceu_edge_t         fldsel;
    ceu_edge_t         hdsel;
    ceu_edge_t         vdsel;
    ceu_cap_rect_t    * cap;
    ceu_clp_t        * clp;
    ceu_onoff_t        cols;
    ceu_onoff_t        cows;
    ceu_onoff_t        cobs;
} ceu_config_t;
```

型 / メンバ名	説明
ceu_jpg_t jpg	取り込みモードの選択 • CEU_IMAGE_CAPTURE_MODE 画像取り込みモード • CEU_DATA_SYNC_MODE データ同期取り込みモード • CEU_DATA_ENABLE_MODE データイネーブル取り込みモード
ceu_dtif_t dtif	キャプチャ対象となるデジタル画像入力端子を設定 • CEU_8BIT_DATA_PINS 8 ビットインタフェース • CEU_16BIT_DATA_PINS 16 ビットインタフェース
ceu_sig_pol_t vdpol	外部モジュールからの垂直同期信号検出の極性設定 • CEU_HIGH_ACTIVE 外部モジュールからの垂直同期信号（VD）を正極性として検出 • CEU_LOW_ACTIVE 外部モジュールからの垂直同期信号（VD）を負極性として検出
ceu_sig_pol_t hdpol	外部モジュールからの水平同期信号検出の極性設定 • CEU_HIGH_ACTIVE 外部モジュールからの水平同期信号（HD）を正極性として検出 • CEU_LOW_ACTIVE 外部モジュールからの水平同期信号（HD）を負極性として検出
ceu_dtary_t dtary	輝度成分と色差成分の入力順序設定 データ同期取り込みモード、データイネーブル取り込みモードの時、 CEU_CB0_Y0_CR0_Y1 を設定してください (8 ビットインタフェースの時) • CEU_CB0_Y0_CR0_Y1 画像入力データを Cb0、Y0、Cr0、Y1 の順序で取り込み

		• CEU_CR0_Y0_CB0_Y1 画像入力データを Cr0、Y0、Cb0、Y1 の順序で取り込み • CEU_Y0_CB0_Y1_CR0 画像入力データを Y0、Cb0、Y1、Cr0 の順序で取り込み • CEU_Y0_CR0_Y1_CB0 画像入力データを Y0、Cr0、Y1、Cb0 の順序で取り込み (16 ビットインタフェースの時) • CEU_CB0_Y0_CR0_Y1 画像入力データを{Cb0、Y0}、{Cr0、Y1}の順序で取り込み • CEU_CR0_Y0_CB0_Y1 画像入力データを{Cr0、Y0}、{Cb0、Y1}の順序で取り込み • CEU_Y0_CB0_Y1_CR0 画像入力データを{Y0、Cb0}、{Y1、Cr0}の順序で取り込み • CEU_Y0_CR0_Y1_CB0 画像入力データを{Y0、Cr0}、{Y1、Cb0}順序で取り込み
ceu_edge_t	dsel	外部モジュールから画像データの取り込みエッジ設定 • CEU_EDGE_RISING カメラクロックの立ち上がりエッジで取り込み • CEU_EDGE_FALLING カメラクロックの立ち下がりエッジで取り込み
ceu_edge_t	fldsel	外部モジュールからフィールド識別信号の取り込みエッジ設定 • CEU_EDGE_RISING カメラクロックの立ち上がりエッジで取り込み • CEU_EDGE_FALLING カメラクロックの立ち下がりエッジで取り込み
ceu_edge_t	hdsel	外部モジュールから水平同期信号の取り込みエッジ設定 • CEU_EDGE_RISING カメラクロックの立ち上がりエッジで取り込み • CEU_EDGE_FALLING カメラクロックの立ち下がりエッジで取り込み
ceu_edge_t	vdsel	外部モジュールから垂直同期信号の取り込みエッジ設定 • CEU_EDGE_RISING カメラクロックの立ち上がりエッジで取り込み • CEU_EDGE_FALLING カメラクロックの立ち下がりエッジで取り込み
ceu_cap_rect_t * cap		キャプチャ取り込みサイズ設定 画像取り込みモード、データ同期取り込みモードの時、設定が必要 設定が不要の場合、NULL を設定してください
ceu_clp_t * clp		フィルタサイズクリップ設定 画像取り込みモードの時、設定が必要になります 設定が不要の場合、NULL を設定してください
ceu_onoff_t	cols	32 ビットスワップ
ceu_onoff_t	cows	16 ビットスワップ
ceu_onoff_t	cobs	8 ビットスワップ

(b) ceu_cap_rect_t

ceu_cap_rect_t 構造体のメンバは以下の通りです。画像取り込みモード、データ同期取り込みモードの時、設定が必要になります。

```
typedef struct
{
    uint32_t    vofst;
    uint32_t    vwdth;
    uint32_t    hofst;
    uint32_t    hwdth;
} ceu_cap_rect_t;
```

型	メンバ名	説明
uint32_t	vofst	キャプチャ位置を垂直同期信号からの HD 数指定 [1HD 単位] 4095 以下で設定してください
uint32_t	vwdth	垂直方向のキャプチャ期間指定 [4HD 単位] 1920 以内で設定してください
uint32_t	hofst	キャプチャ位置を水平同期信号からのサイクル数指定[1 サイクル単位] 8191 以下で設定してください
uint32_t	hwdth	水平方向のキャプチャ期間指定 (8 ビットインタフェースの時) 画像取り込みモードの時 [8 サイクル単位]: 5,120 サイクル以下 データ同期取り込みモードの時 [4 サイクル単位] : 2,560 サイクル以下 (16 ビットインタフェースの時) 画像取り込みモードの時 [4 サイクル単位]: 2,560 サイクル以下 データ同期取り込みモードの時 [2 サイクル単位] : 1,280 サイクル以下

(c) ceu_clp_t

ceu_clp_t 構造体のメンバは以下の通りです。

画像取り込みモードの時、設定が必要になります。

```
typedef struct
{
    uint32_t    vfclp;
    uint32_t    hfclp;
} ceu_clp_t;
```

型	メンバ名	説明
uint32_t	vfclp	垂直方向のフィルタ出力サイズのクリップ値 [4pixel 単位]
uint32_t	hfclp	水平方向のフィルタ出力サイズのクリップ値 [4pixel 単位]

(3) キャプチャ取り込みサイズの設定について

YCbCr422 形式の映像を出力する CMOS カメラを接続した場合のキャプチャ取り込みサイズ設定(cap)について記載します。

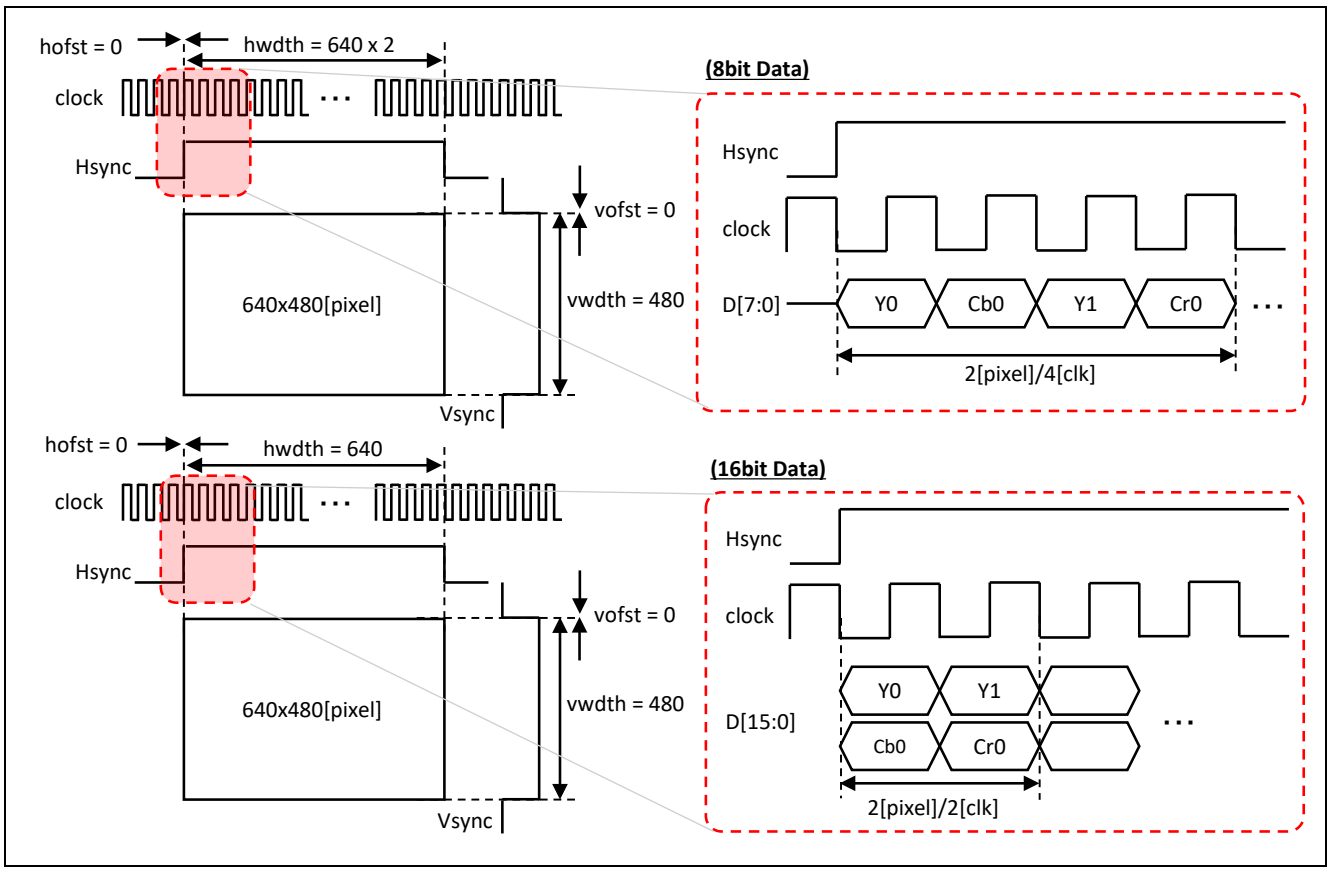

図 7-2 カメラから出力される信号タイミング

カメラから出力される信号タイミングを図 3 1 に記載します。この図よりカメラから映像が出力されるタイミングは、水平同期信号(Hsync)/垂直同期信号(Vsync)の立ち上がりと同時になる為、映像取り込み位置を示す hofst/vofst は、"0"となります。

垂直の映像取り込み期間を表す vwdth については、映像の高さと同じ 480 となりますが、水平の映像取り込み期間を表す hwdth については、1[pixel]の取り込みに必要なクロックで設定が異なります。

8 ビットインタフェースで接続した場合、2[pixel]の取り込みに必要なクロックは、2 倍の 4[clk]となり hwdth は、640 x 2「clk」となります。

16 ビットインタフェースで接続した場合、2[pixel]の取り込みに必要なクロックは、同じ 2[clk]となる為、640[clk]となります。

図 7-3に 8 ビットインタフェースで接続した場合の設定例を記載します。

```
画像取り込みモード
ceu_config_t    config;
ceu_cap_rect_t cap;
ceu_clp_t       clp;

config.jpg     = CEU_IMAGE_CAPTURE_MODE;

cap.hofst      = 0u;
cap.vofst      = 0u;
cap.hwdth      = 640u* 2u;
cap.vwdth      = 480u;
config.cap     = &cap;

clp.hfclp      = 640u;
clp.vfclp      = 480u;
config.clp     = &clp;
```

```
データ同期取り込みモード
ceu_config_t    config;
ceu_cap_rect_t cap;

config.jpg     = CEU_DATA_SYNC_MODE;

cap.hofst      = 0u;
cap.vofst      = 0u;
cap.hwdth      = 640u* 2u;
cap.vwdth      = 480u;
config.cap     = &cap;

config.clp     = NULL;
```

```
データイネーブル取り込みモード
ceu_config_t    config;

config.jpg     = CEU_DATA_ENABLE_MODE;

config.cap     = NULL;

config.clp     = NULL;
```

図 7-3 パラメータ設定例 (8 ビットインタフェース)

図 7-4に 16 ビットインタフェースで接続した場合の設定例を記載します。

```
画像取り込みモード
ceu_config_t    config;
ceu_cap_rect_t cap;
ceu_clp_t       clp;

config.jpg     = CEU_IMAGE_CAPTURE_MODE;

cap.hofst      = 0u;
cap.vofst      = 0u;
cap.hwdth      = 640u;
cap.vwdth      = 480u;
config.cap     = &cap;

clp.hfclp      = 640u;
clp.vfclp      = 480u;
config.clp     = &clp;
```

```
データ同期取り込みモード
ceu_config_t    config;
ceu_cap_rect_t cap;

config.jpg     = CEU_DATA_SYNC_MODE;

cap.hofst      = 0u;
cap.vofst      = 0u;
cap.hwdth      = 640u;
cap.vwdth      = 480u;
config.cap     = &cap;

config.clp     = NULL;
```

```
データイネーブル取り込みモード
ceu_config_t    config;

config.jpg     = CEU_DATA_ENABLE_MODE;

config.cap     = NULL;

config.clp     = NULL;
```

図 7-4 パラメータ設定例 (16 ビットインタフェース)

7.5 R_RVAPI_CaptureStartCEU

R_RVAPI_CaptureStartCEU

概　要　フレームキャプチャ開始

ヘッダ　r_rvapi_ceu.h

宣　言

```
ceu_error_t R_RVAPI_CaptureStartCEU(
            const void * cayr,
            const void * cacr,
            uint32_t chdw);
```

引　数

[IN] void * cayr ：データ格納先アドレスの指定 1
NULL は設定しないでください
- 画像取り込みモードの時
  キャプチャデータ輝度成分データ格納先アドレス[4byte 単位]
- データ同期取り込みモードの時
  データ格納先アドレス[4byte 単位]
- データタイネーブル取り込みモードの時
  データ格納先アドレス[32byte 単位]

[IN] void * cacr ：データ格納先アドレスの指定 2
- 画像取り込みモードの時、設定が必要になります
  キャプチャデータ色差成分データ格納先アドレス[4byte 単位]

[IN] uint32_t chdw ：データ格納バッファのストライド[byte]
- 画像取り込みモードの時
  キャプチャデータ格納バッファのストライド[4byte 単位]
- データ同期取り込みモードの時
  (8 ビットインタフェースの場合)
  水平方向のキャプチャ期間(hwdth)を設定してください
  (16 ビットインタフェースの場合)
  水平方向のキャプチャ期間(hwdth) x 2 を設定してください

リターン値

CEU_OK	：正常終了
CEU_ERR_PARAM	：cayr/ cacr の設定が NULL
	：cayr/ cacr の設定が範囲外
	：chdw の設定が範囲外
	：キャプチャ中に再度関数が呼び出された

備考

(1) 説明

本関数で、1 フレームキャプチャ開始します。本関数は、非同期の為、1 フレームキャプチャの完了は、『7.7 R_RVAPI_InterruptEnableCEU()』を使用し行ってください。

本関数内では、以下のドライバを使用しています。

- R_CEU_Execute ()

7.6 R_RVAPI_CaptureStopCEU

R_RVAPI_CaptureStopCEU	
概　要	キャプチャ停止
ヘッダ	r_rvapi_ceu.h
宣　言	ceu_error_t R_RVAPI_CaptureStopCEU(void);
引　数	なし
リターン値	CEU_OK ：正常終了
備考	

(1) 説明

本関数内で、キャプチャを停止します。

本関数内では、以下のドライバを使用しています。

- R_CEU_Stop ()

7.7 R_RVAPI_InterruptEnableCEU

R_RVAPI_InterruptEnableCEU			
概　要	割り込み許可設定		
ヘッダ	r_rvapi_ceu.h		
宣　言	ceu_error_t R_RVAPI_InterruptEnableCEU(const ceu_int_type_t int_type, void (* const callback)(ceu_int_type_t));		
引　数	[in]	ceu_int_type_t int_type	: CEU 割り込みの選択
	[in]	callback void (*callback)(ceu_int_type_t)	: コールバック関数の登録 必要がない場合、NULL を設定してください
リターン値	CEU_OK		: 正常終了
	CEU_ERR_PARAM		: コールバック関数が NULL
備考			

(1) 説明

本関数では、以下の処理を行います。複数の割り込みを使用する場合、ceu_int_type_t 型の定義を OR して設定してください。コールバック関数の引数で発生した割り込みが判定可能になります。尚、コールバック関数を複数登録した場合、最後に登録されたものが有効になります。

- 引数で指定された CEU 割り込みの許可設定
- 引数で登録されたコールバック関数の登録

本関数内では、以下のドライバを使用しています。

- R_CEU_InterruptEnable ()

8. 関数リファレンス(MIPI)

8.1 R_RVAPI_InitializeMIPI

R_RVAPI_InitializeMIPI	
概　要	MIPI 初期化設定
ヘッダ	r_rvapi_mipi.h
宣　言	void R_RVAPI_InitializeMIPI(void);
引　数	[IN]　なし　　:
リターン値	なし
備考	

(1) 説明

本関数では、MIPI および VIN のスタンバイ解除及び割り込みの許可設定、割り込みハンドラの設定を行ないます。

本関数内では、以下のドライバを使用しています。

- R_MIPI_Initialize ()

8.2 R_RVAPI_TerminateMIPI

R_RVAPI_TerminateMIPI	
概　要	MIPI 終了設定
ヘッダ	r_rvapi_mipi.h
宣　言	void R_RVAPI_TerminateMIPI(void);
引　数	[IN]　なし　　:
リターン値	なし
備考	

(1) 説明

本関数では、MIPI および VIN のスタンバイ設定及び割り込みの禁止設定、割り込みハンドラの解除を行ないます。

本関数内では、以下のドライバを使用しています。

- R_MIPI_InterruptDisable ()
- R_MIPI_Close ()

8.3 R_RVAPI_OpenMIPI

R_RVAPI_OpenMIPI			
概　要	MIPI 映像取り込み設定		
ヘッダ	r_rvapi_mipi.h		
宣　言	e_mipi_error_t R_RVAPI_OpenMIPI(const st_mipi_param_t * const config);		
引　数	[IN]	const st_mipi_param_t * const config	: コンフィグレーションデータ NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_PARAM_ERR		: 引数が NULL
備考			

(1) 説明

本関数では、キャプチャレーンや画像取り込み方式、PHY の設定などの MIPI の映像取り込み設定行ないます。

本関数内では、以下のドライバを使用しています。

- R_MIPI_Open ()

(2) パラメータ詳細

st_mipi_param_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint8_t  mipi_lanenum;         /*!< Mipi Lane Num */
    uint8_t  mipi_vc;             /*!< Mipi Virtual Channel */
    uint8_t  mipi_interlace;       /*!< Interlace or Progressive */
    uint8_t  mipi_laneswap;        /*!< Mipi Lane Swap Setting */
    uint16_t mipi_frametop;        /*!< (for Interlace)Top Field Packet ID */
    uint16_t mipi_outputrate;      /*!< Mipi Data Send Speed(Mbit per sec) */
} st_mipi_param_t;
```

型 / メンバ	説明
uint8_t mipi_lanenum	転送レーン数 1：1 レーン動作 2：2 レーン並列動作
uint8_t mipi_vc	仮想チャネル 0～3 有効な仮想チャネル番号
uint8_t mipi_interlace	入力方式 MIPI_PROGRESSIVE：プログレッシブ MIPI_INTERLACE：インタレース
uint8_t mipi_laneswap	レーンスワップ 0：レーンスワップなし 1：レーンスワップあり
uint16_t mipi_frametop	偶数フィールド番号 0x0000～0xFFFF インタレース入力画像のトップフィールドを検出するための値 先頭ライン同期パケットの ID を設定
uint16_t mipi_outputrate	MIPI 転送レート(MHz) 80～1000 MIPI の転送レートを設定します

偶数フィールド番号(mipi_frametop)は入力方式(mipi_interlace)が MIPI_INTERLACE の場合のみ有効です。

仮想チャネル(mipi_vc)とは、カメラからデータが流れるチャネルを意味します。

8.4 R_RVAPI_InterruptEnableMIPI

R_RVAPI_InterruptEnableMIPI			
概　要	割り込み許可設定		
ヘッダ	r_rvapi_mipi.h		
宣　言	e_mipi_error_t R_RVAPI_InterruptEnableMIPI(const st_mipi_int_t * const param);		
引　数	[IN]	const st_mipi_int_t * const param	: 割り込み設定 NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_PARAM_ERR		: 引数が NULL
備考			

(1) 説明

本関数では、以下の処理を行います。複数の割り込みを使用する場合、e_mipi_interrupt_type_t 型の定義を OR して設定してください。コールバック関数の引数で発生した割り込みが判定可能になります。

- 引数で指定された MIPI 割り込みの許可設定
- 引数で登録されたコールバック関数の登録

本関数内では、以下のドライバを使用しています。

- R_MIPI_InterruptEnable ()

(2) パラメータ説明

(a) st_mipi_int_t

st_mipi_int_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    e_mipi_interrupt_type_t type;
    void (* p_mipiCallback) (e_mipi_interrupt_type_t interrupt_flag);
    void (* p_vinCallback)  (e_mipi_interrupt_type_t interrupt_flag);
    uint32_t line_num;
} st_mipi_int_t;
```

型 / メンバ	説明
e_mipi_interrupt_type_t type	MIPI および VIN の割り込み要因 MIPI および VIN の割り込み要因のうち許可する要因を設定します
void (* p_mipiCallback) (e_mipi_interrupt_type_t interrupt_flag)	MIPI 割り込みコールバック関数 MIPI 割り込みが発生した際に呼び出される関数です NULL は設定しないでください
void (* p_vinCallback) (e_mipi_interrupt_type_t interrupt_flag)	VIN 割り込みコールバック関数 MIPI 割り込みが発生した際に呼び出される関数です NULL は設定しないでください
uint32_t line_num	スキャンライン割り込みのライン指定 0x0000～0x07FF VIN_INT_SCANLINE が type で指定された際の、割り込み発生ラインを指定します

(b) e_mipi_interrupt_type_t

e_mipi_interrupt_type_t は MIPI および VIN の割り込み要因を表す列挙型です。

```
typedef enum
{
    MIPI_INT_LESS_THAN_WC    = 0x00000001,
    MIPI_INT_AFIFO_OF        = 0x00000002,
    MIPI_INT_VD_START        = 0x00000004,
    MIPI_INT_VD_END          = 0x00000008,
    MIPI_INT_SHP_STB         = 0x00000010,
    MIPI_INT_FSFE            = 0x00000020,
    MIPI_INT_LNP_STB         = 0x00000040,
    MIPI_INT_CRC_ERR         = 0x00000080,
    MIPI_INT_HD_WC_ZERO      = 0x00000100,
    MIPI_INT_FRM_SEQ_ERR1    = 0x00000200,
    MIPI_INT_FRM_SEQ_ERR0    = 0x00000400,
    MIPI_INT_ECC_ERR         = 0x00000800,
    MIPI_INT_ECC_CRCT_ERR    = 0x00001000,
    MIPI_INT_ULPS_START      = 0x00002000,
    MIPI_INT_ULPS_END        = 0x00004000,
    MIPI_INT_ERRSOTHS        = 0x00008000,
    MIPI_INT_ERRSOTSYNCHS    = 0x00010000,
    MIPI_INT_ERRESC          = 0x00020000,
    MIPI_INT_ERRCONTROL      = 0x00040000,
    VIN_INT_FIELD2           = 0x00100000,
    VIN_INT_VSYNC_FALL       = 0x00200000,
    VIN_INT_VSYNC_RISE       = 0x00400000,
    VIN_INT_FIELD            = 0x00800000,
    VIN_INT_SCANLINE         = 0x01000000,
    VIN_INT_FRAME            = 0x02000000,
    VIN_INT_FIFO_OF          = 0x04000000
} e_mipi_interrupt_type_t;
```

列挙定数	値	説明
MIPI_INT_LESS_THAN_WC	00000001H	ロングパケットのペイロードデータ長が、WC 値よりも小さいときのエラー割り込み
MIPI_INT_AFIFO_OF	00000002H	PHY からの HS データが格納される非同期 FIFO のオーバフロー割り込み
MIPI_INT_VD_START	00000004H	VD 信号出力の開始割り込み
MIPI_INT_VD_END	00000008H	VD 信号出力の終了割り込み
MIPI_INT_SHP_STB	00000010H	ショートパケット受信割り込み
MIPI_INT_FSFE	00000020H	フレームパケット受信割り込み
MIPI_INT_LNP_STB	00000040H	ロングパケット受信割り込み
MIPI_INT_CRC_ERR	00000080H	CRC エラー割り込み
MIPI_INT_HD_WC_ZERO	00000100H	WC ゼロ割り込み
MIPI_INT_FRM_SEQ_ERR1	00000200H	フレームシーケンスエラー1 割り込み (不正なフレームエンドパケット受信)
MIPI_INT_FRM_SEQ_ERR0	00000400H	フレームシーケンスエラー0 割り込み (不正なフレームスタートパケット受信)
MIPI_INT_ECC_ERR	00000800H	ECC エラー割り込み
MIPI_INT_ECC_CRCT_ERR	00001000H	ECC 1 ビット訂正割り込み
MIPI_INT_ULPS_START	00002000H	ウルトラローパワー転送開始割り込み
MIPI_INT_ULPS_END	00004000H	ウルトラローパワー転送終了割り込み
MIPI_INT_ERRSOTHS	00008000H	同期化 SOT(転送開始)エラー割り込み
MIPI_INT_ERRSOTSYNCHS	00010000H	非同期 SOT(転送開始)エラー割り込み
MIPI_INT_ERRESC	00020000H	エスケープモードエントリエラー割り込み
MIPI_INT_ERRCONTROL	00040000H	PHY 制御エラー割り込み
VIN_INT_FIELD2	00100000H	フィールド割り込み
VIN_INT_VSYNC_FALL	00200000H	VSYNC 立ち下りエッジ検出割り込み
VIN_INT_VSYNC_RISE	00400000H	VSYNC 立ち上がりエッジ検出割り込み
VIN_INT_FIELD	00800000H	フィールドスイッチング割り込み
VIN_INT_SCANLINE	01000000H	スキャンライン割り込み
VIN_INT_FRAME	02000000H	フレーム終了割り込み
VIN_INT_FIFO_OF	04000000H	FIFO オーバフロー割り込み

8.5 R_RVAPI_SetupMIPI

R_RVAPI_SetupMIPI			
概　要	VIN 映像取り込み設定		
ヘッダ	r_rvapi_mipi.h		
宣　言	e_mipi_error_t R_RVAPI_SetupMIPI(const st_vin_setup_t * const setup);		
引　数	[IN]	const st_vin_setup_t * const setup	: コンフィグレーションデータ NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_PARAM_ERR		: mipi_data の設定が不正はたは範囲外
備考			

(1) 説明

本関数では、キャプチャ画像のクリッピングエリアなどの取り込みサイズの設定を行ないます。

本関数内では、以下のドライバを使用しています。

- R_MIPI_Setup ()

(2) パラメータ詳細

(a) st_vin_setup_t

st_vin_setup_t 構造体のメンバは以下の通りです。

```
typedef struct
{
   st_vin_preclip_t      vin_preclip;
   uint8_t               vin_inputformat;
   uint8_t               vin_outputformat;
   uint8_t               vin_outputendian;
   uint8_t               vin_interlace;
   uint16_t              vin_stride;
   uint16_t              vin_ycoffset;
    e_vin_input_align_t  vin_input_align;
    e_vin_output_swap_t  vin_output_swap;
} st_vin_setup_t;
```

型 / メンバ	説明
st_vin_preclip_t vin_preclip	プリクリップエリア設定 キャプチャ画像に対するクリップエリアを設定します 詳細は「**エラー! 参照元が見つかりません。**」を参照ください
uint8_t vin_inputformat	入力フォーマット VIN_INPUT_YCBCR422_8：YUY(=YCbCr422 8bit) VIN_INPUT_YCBCR422_8I：UYVY VIN_INPUT_RAW8：RAW 8bit
uint8_t vin_outputformat	出力フォーマット VIN_OUTPUT_YCBCR422_8：YUY(=YCbCr422 8bit) VIN_OUTPUT_Y8_CbCr：YC 分離, YCbCr422(Y 8bit, Cb/Cr 8bit) VIN_OUTPUT_Y8：YC 分離, Y data(8bit) VIN_OUTPUT_RAW8：RAW 8bit
uint8_t vin_outputendian	エンディアンタイプ VIN_OUUPUT_EN_LITTLE：リトルエンディアン VIN_OUTPUT_EN_BIG：ビッグエンディアン
uint8_t vin_interlace	インタレースモード VIN_INTERLACE_ODD：奇数フィールドキャプチャモード VIN_INTERLACE_EVEN：偶数フィールドキャプチャモード VIN_INTERLACE_BOTH：奇数/偶数フィールドキャプチャモード VIN_PROGRESSIVE：プログレッシブキャプチャモード
uint16_t vin_stride	イメージストライド 32 以上(32 の倍数) 出力画像のストライドサイズを設定します
uint16_t vin_ycoffset	UV データアドレスオフセット 0～128 の倍数 出力フォーマットで YC 分離出力を指定した際の、UV の転送オフセットアドレスを指定します
e_vin_input_align_t vin_input_align	YCbCr422 入力データアライメント VIN_Y_UPPER：上位ビット(Y) 下位ビット(CbCr) VIN_CB_UPPER：上位ビット(CbCr) 下位ビット(Y)
e_vin_output_swap_t vin_output_swap	出力データバイトスワップモード VIN_SWAP_OFF：スワップしない VIN_SWAP_ON：スワップする

エンディアンタイプ(vin_outputendian)は外部メモリに出力する際のエンディアンタイプを指定します。

イメージストイド(vin_stride)は、**エラー! 参照元が見つかりません。**構造体で指定した水平プリクリッピングサイズ(vin_preclip_endx - vin_preclip_startx)以上の値を設定する必要があります。したがって、以下の式を満たすようにイメージストライドを設定してください。

vin_stride >= vin_afterclip_size_x

また、イメージストライドは出力フォーマット(vin_outputformat)により、以下のようにパラメータを設定する必要があります。

出力フォーマット	設定単位(Pixel)
VIN_OUTPUT_YCBCR422_8	64
VIN_OUTPUT_Y8_CbCr8	128
VIN_OUTPUT_Y8	128
VIN_OUTPUT_RAW8	64

イメージストライドで設定した値は、MIPI ドライバが VnIS レジスタへ設定値を書き込みます。H/W 仕様により､出力フォーマットが VIN_OUTPUT_RAW8 の場合は vin_stride を 2 で割った値を MIPI ドライバが VnIS レジスタへ書き込みます。

(b) st_vin_preclip_t

st_vin_preclip_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t vin_preclip_starty;    /*!< Pre Area Clip Start Line */
    uint16_t vin_preclip_endy;      /*!< Pre Area Clip End Line */
    uint16_t vin_preclip_startx;    /*!< Pre Area Clip Start Column */
    uint16_t vin_preclip_endx;      /*!< Pre Area Clip End Column */
} st_vin_preclip_t;
```

型 / メンバ	説明
uint16_t vin_preclip_starty	スタートライン(垂直方向) 0～2046 (スケーリング使用時は 0～2044) 値 0 は最初の有効な行を意味します
uint16_t vin_preclip_endy	エンドライン(垂直方向) 1～2047 (スケーリング使用時は 3～2047)
uint16_t vin_preclip_startx	スタートピクセル(水平方向) 0～2042 までの偶数 値 0 は最初の有効ピクセルが指定されます
uint16_t vin_preclip_endx	エンドピクセル(水平方向) 5～2047 までの奇数

垂直方向のライン数は、プリクリッピング後のライン数が 2 以上となる必要があるため、

(vin_preclip_endy - vin_preclip_starty) >= 1

となるように設定してください。また、垂直または水平スケーリングを使用する場合は、

(vin_preclip_endy - vin_preclip_starty) >=3

となるように設定してください。

水平方向のピクセル数は、プリクリッピング後のピクセル数が 6 よりも大きな偶数となる必要があるため、

(vin_preclip_endx - vin_preclip_startx) >=5

を満たし、かつ奇数となるように設定してください。

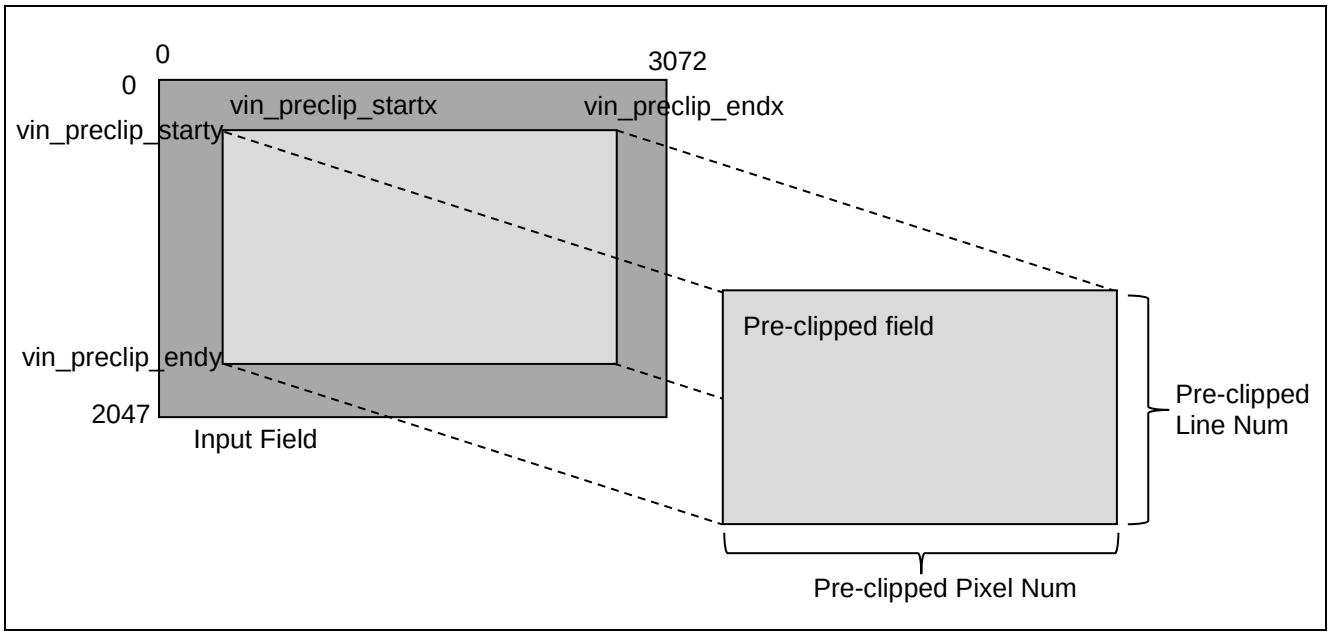

図 8-1 プリクリッピングエリアのイメージ

8.6 R_RVAPI_SetBufferMIPI

R_RVAPI_SetBufferMIPI

概　要	キャプチャバッファ設定		
ヘッダ	r_rvapi_mipi.h		
宣　言	e_mipi_error_t R_RVAPI_SetBufferMIPI(const uint8_t buffer_no, const uint8_t * const buffer);		
引　数	[IN]	const uint8_t buffer_no	: MB レジスタ番号 0 : MB1、1 : MB2、2 : MB3
		const uint8_t * const buffer	: キャプチャバッファアドレス
リターン値	MIPI_OK		: 正常終了
	MIPI_PARAM_ERR		: 引数が NULL
	MIPI_STATUS_ERR		: MIPI ドライバ内部状態不正

備考

(1) 説明

本関数内は、キャプチャバッファのアドレスを、第一引数に従って VIN の MB1～MB3 へ設定します。

本関数内では、以下のドライバを使用しています。

- R_MIPI_SetBufferAdr ()

8.7 R_RVAPI_CaptureStartMIPI

R_RVAPI_CaptureStartMIPI

概　要	フレームキャプチャ開始		
ヘッダ	r_rvapi_mipi.h		
宣　言	e_mipi_error_t R_RVAPI_CaptureStartMIPI(void);		
引　数	[IN]	なし	
リターン値	MIPI_OK		: 正常終了
	MIPI_STATUS_ERR		: MIPI ドライバ内部状態不正

備考

(1) 説明

本関数内で、連続フレームキャプチャを開始します。本関数は、非同期の為、フレームキャプチャの完了は、『8.4 R_RVAPI_InterruptEnableMIPI ()』で登録するコールバックを使用し行ってください。

本関数内では、以下のドライバを使用しています。

- R_MIPI_CaptureStart()

8.8 R_RVAPI_CaptureStopMIPI

R_RVAPI_CaptureStopMIPI		
概　要	キャプチャ停止	
ヘッダ	r_rvapi_mipi.h	
宣　言	e_mipi_error_t R_RVAPI_CaptureStopMIPI(void);	
引　数	なし	
リターン値	MIPI_OK	: 正常終了
	MIPI_STATUS_ERR	: MIPI ドライバ内部状態不正
備考		

(1) 説明

本関数内で、キャプチャを停止します。

本関数内では、以下のドライバを使用しています。

- R_MIPI_CaptureStop ()

9. 関数リファレンス（SPEA）

9.1 R_RVAPI_GraphCreateSurfaceSPEA

R_RVAPI_GraphCreateSurfaceSPEA

概　要　表示領域の生成(SPEA)

ヘッダ　r_rvapi_spea.h

宣　言

```
vdc_error_t R_RVAPI_GraphCreateSurfaceSPEA(
                const vdc_channel_t ch,
                const gr_surface_disp_config_t * const gr_disp_cnf);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	gr_surface_disp_config_t * gr_disp_cnf	: グラフィックス表示領域の設定

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_LAYER_ID	: レイヤ ID 不正エラー VDC_LAYER_ID_0_RD は設定不可
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数ではバッファに配置されたメモリを表示する為の設定を行います。

本関数内では、以下のドライバを使用しています。

- R_VDC_ReadDataControl ()
- R_VDC_StartProcess()

9.2 R_RVAPI_WindowOffsetSPEA

R_RVAPI_WindowOffsetSPEA			
概　要	SPEA Window の座標オフセットの設定		
ヘッダ	r_rvapi_spea.h		
宣　言	spea_error_t R_RVAPI_WindowOffsetSPEA(const vdc_layer_id_t layer_id, const uint16_t offset_x, const uint16_t offset_y);		
引　数	[IN]	vdc_layer_id_t layer_id	：レイヤ ID • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD • VDC_LAYER_ID_0_RD は、設定しないでください。
	[IN]	uint16_t offset_x	: offset_x は、2[pixel]単位で 0 以上 2047 以下を設定してください。
	[IN]	uint16_t offset_y	：offset_y は、0 以上 8191 以下を設定してください。
リターン値	SPEA_OK:		：正常終了
	SPEA_ERR_PARAM_LAYER_ID		：レイヤ ID 不正エラー
	SPEA_ERR_PARAM		：不許可条件エラー
備考			

(1) 説明

本関数では、データ読み出し制御に関する以下の処理を行います。

SPEA の仮想フレームに対する、VDC(レイヤ 2 及び 3)の表示領域の配置を設定する。

本関数内では、以下のドライバを使用しています。

- R_SPEA_WindowOffset()

9.3 R_RVAPI_SetWindowSPEA

R_RVAPI_SetWindowSPEA

概　要　SPEA Window パラメータの設定

ヘッダ　r_rvapi_spea.h

宣　言

```
spea_error_t R_RVAPI_SetWindowSPEA(
                    const vdc_layer_id_t layer_id,
                    const spea_window_id_t  window_id,
                    const spea_onoff_t sken,
                    const spea_sklym_t * size,
                    const spea_skpsm_t * pos,
                    const void * buffer);
```

引　数			
	[IN]	vdc_layer_id_t layer_id	：レイヤ ID ● VDC_LAYER_ID_2_RD ● VDC_LAYER_ID_3_RD ● VDC_LAYER_ID_0_RD は、設定しないでください。
	[IN]	spea_window_id_t window_id	：SPEA ID WINDOW_00 ~ WINDOW15:Window ID
	[IN]	spea_onoff_t sken	：SPEA Window ON/OFF ● SPEA_ON ● SPEA_OFF
	[IN]	spea_sklym_t * size	：Window サイズ size.x は、2[pixel]単位で 0 以上 2047 以下を設定してください。 size.y は、0 以上 8191 以下を設定してください。
	[IN]	spea_skpsm_t * pos	：Window 開始座標 pos.x は 2[pixel]単位で設定してください。また、R_RVAPI_WindowOffsetSPEA で設定した offset_x を pos.x に加算した結果が 0 以上 2047 以下でない場合エラーとなります。 pos.y は R_RVAPI_WindowOffsetSPEA で設定した offset_y を pos.y に加算した結果が 0 以上 8191 以下でない場合エラーとなります。
	[IN]	void * buffer	：Window の読出しバッファアドレス 8 バイトアライメントのアドレスを指定してください。

リターン値		
	SPEA_OK:	：正常終了
	SPEA_ERR_PARAM_LAYER_ID	：レイヤ ID 不正エラー
	SPEA_ERR_PARAM	：不許可条件エラー

備考

(1)　説明

本関数では、データ読出し制御に関する以下の処理を行います。

SPEA の Window の表示/非表示

SPEA の Window 開始座標、サイズ、読み出しバッファの設定

VDC のフレームバッファバースト転送モードの設定(SPEA_ON:128 バイト SPEA_OFF:32 バイト転送)

本関数内では、以下のドライバを使用しています。

R_SPEA_SetWindow()

9.4 R_RVAPI_WindowUpdateSPEA

R_RVAPI_WindowUpdateSPEA			
概　要	SPEA Window パラメータの更新要求		
ヘッダ	r_rvapi_spea.h		
宣　言	spea_error_t R_RVAPI_WindowUpdateSPEA(const vdc_layer_id_t layer_id);		
引　数	[IN]	vdc_layer_id_t layer_id	: レイヤ ID • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD • VDC_LAYER_ID_0_RD は、設定しないでください。
リターン値	SPEA_OK:		: 正常終了
	SPEA_ERR_PARAM_LAYER_ID		: レイヤ ID 不正エラー
備考			

(1) 説明

本関数では、データ読出し制御に関する以下の処理を行います。

SPEA の Window パラメータの更新要求

本関数内では、以下のドライバを使用しています。

- R_SPEA_WindowUpdate()

9.5 R_RVAPI_GraphCreateSurfaceRLE

R_RVAPI_GraphCreateSurfaceRLE

概　要　表示領域の生成(RLE)

ヘッダ　r_rvapi_spea.h

宣　言

```
vdc_error_t R_RVAPI_GraphCreateSurfaceRLE(
                const vdc_channel_t ch,
                const gr_surface_disp_config_t * const gr_disp_cnf);
```

引　数			
	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	gr_surface_disp_config_t * gr_disp_cnf	: グラフィックス表示領域の設定

リターン値		
	VDC_OK:	: 正常終了
	VDC_ERR_PARAM_CHANNEL	: チャネル不正エラー
	VDC_ERR_PARAM_NULL	: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH	: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED	: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE	: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION	: 不許可条件エラー
	VDC_ERR_RESOURCE_LAYER	: レイヤリソースエラー

備考

(1) 説明

本関数ではバッファに配置されたメモリを表示する為の設定を行います。

本関数内では、以下のドライバを使用しています。

- R_VDC_ReadDataControl ()
- R_VDC_StartProcess()

9.6 R_RVAPI_SetWindowRLE

R_RVAPI_SetWindowRLE

概　要　RLE パラメータの設定及び更新

ヘッダ　r_rvapi_spea.h

宣　言

```
vdc_error_t R_RVAPI_SetWindowRLE(
                const vdc_channel_t ch,
                const rle_onoff_t_sken,
                const rle_cfg_t * rle_cfg,
                const void * buffer,
                const uint8_t * g_rle_image,
                const uint32_t size_of_image);
```

引　数	[IN]	vdc_channel_t ch	: VDC チャネル • VDC_CHANNEL_0
	[IN]	rle_onoff_t sken	: RLE ON/OFF • RLE_ON • RLE_OFF
	[IN]	rle_cfg_t * rle_cfg	: NULL を設定してください。（TBD）
	[IN]	void * buffer	: Window の読出しバッファアドレス 8 バイトアライメントのアドレスを指定してください。
	[IN]	uint8_t * g_rle_image	: Targa 形式の画像データ
	[IN]	uint32_t size_of_image	: Targa 形式の画像ファイルサイズ
リターン値	VDC_OK:		: 正常終了
	VDC_ERR_PARAM_CHANNEL		: チャネル不正エラー
	VDC_ERR_PARAM_NULL		: NULL 指定エラー
	VDC_ERR_PARAM_BIT_WIDTH		: ビット幅エラー
	VDC_ERR_PARAM_UNDEFINED		: 未定義パラメータ指定エラー
	VDC_ERR_PARAM_EXCEED_RANGE		: 設定範囲外エラー
	VDC_ERR_PARAM_CONDITION		: 不許可条件エラー
	VDC_ERR_RESOURCE_LAYER		: レイヤリソースエラー

備考

(1) 説明

本関数では、データ読出し制御、更新に関する以下の処理を行います。

SPEA の RLE の有効/無効

SPEA の RLE パラメータの設定

SPEA の RLE パラメータの更新要求

本関数内では、以下のドライバを使用しています。

- R_RLE_SetWindow()
- R_RLE_WindowUpdate()
- R_VDC_ChangeReadProcess()

10. ドライバのインポート方法

10.1 e² studio

Smart Configurator ツールを使用して e2 studio のプロジェクトにドライバをインポートする方法の詳細については、RZ/A2M Smart Configurator ユーザーガイド：e2 studio R20AN0583JJ を参照してください。

10.2 e² studio 以外で作成されたプロジェクトの場合

このセクションでは、ドライバをプロジェクトにインポートする方法について説明します。

一般的に、どの IDE にも 2 つのステップがあります。

1) プロジェクトに必要なソースツリー内の場所にドライバをコピーします。

2) ドライバをコピーした場所へのリンクをコンパイラに追加します。

他に必要なドライバがある場合（例えば r_cbuffer など）、同様にインポートする必要があります。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Sep.14.18	—	初版
1.01	Dec.28.18	74	関数”R_RVAPI_SetupMIPI()” のパラメータ追加
		81~86	“9.関数リファレンス (SPEA)” 追加
1.02	Apr.15.19	59	“R_CEU_InterruptEnable()” は未使用のため削除
		59	表 7-1 に以下のパラメータを追記 ・ceu_edge_t vdsel ・ceu_edge_t hdsel ・ceu_edge_t fldsel ・ceu_edge_t dsel
		60,61	以下のパラメータを ceu_config_t 構造体メンバに追加し、各パラメータの説明を追記 ・ceu_edge_t vdsel ・ceu_edge_t hdsel ・ceu_edge_t fldsel ・ceu_edge_t dsel
1.10	May.17.19	5	表10.1 動作確認条件 コンパイラオプション"-mthumb-interwork"を削除
		87	「10.ドライバのインポート方法」追加

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。