User's Manual

CONFIDENTIAL

RH850/D1x Device Family

Renesas Graphics Library Sprite Engine A (SPEA) Driver

User’s Manual: Software

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev. 2.0 May 2020

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

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

(Rev.4.0-1 November 2017)

Trademark

- Green Hills, the Green Hills logo, INTEGRITY, MULTI, DoubleCheck, EventAnalyzer, Integrate, SuperTrace, ResourceAnalyzer, CodeFactor, INTEGRITY MULTIvisor, GMART, GSTART, G-Cover, PathAnalyzer, GHNet, TimeMachine, μ-velOSity, Padded Cell, TotalDeveloper, and Optimizing Compiler are trademarks or registered trademarks of Green Hills Software in the US and/or internationally.
- This software contains the technology owned by TES Electronic Solutions GmbH. All rights reserved for TES Electronic Solutions GmbH
- Trademarks and trademark symbols (® or ™) are omitted in the text of this manual.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding the functions of SPEA. This manual is written for engineers who use SPEA.

> Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

> The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Please refer to documents of drivers and hardware for a target system implementing SPEA as necessary.

The following documents are related documents. Make sure to refer to the latest versions of these documents.

Document Type	Description	Document Title	Document No.
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description	RH850/D1L/D1M Group User's Manual: Hardware	R01UH0451EJ0220
User's manual for Software	Description of RGL overview	Renesas Graphics Library User's Manual: Software	R01US0181ED0400
	Description of WM	Renesas Graphics Library Window Manager (WM) Driver User's Manual: Software	LLWEB-10035990
	Description of SPEA	Renesas Graphics Library Sprite Engine A (SPEA) Driver User's Manual: Software	LLWEB-10035991 (This manual)
	Description of VDCE	Renesas Graphics Library Video Data Controller E (VDCE) Driver User's Manual: Software	LLWEB-10035992
	Description of VOWE	Renesas Graphics Library Video Output Warping Engine (VOWE) Driver User's Manual: Software	LLWEB-10035993
	Description of JCUA	Renesas Graphics Library JPEG Codec Unit A (JCUA) Driver User's Manual: Software	LLWEB-10035994
	Description of SFMA	Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver User's Manual: Software	LLWEB-10064753
	Description of HYPB	Renesas Graphics Library HyperBus Controller (HYPB) Driver User's Manual: Software	LLWEB-10064754
	Description of OCTA	Renesas Graphics Library OctaBus Controller (OCTA) Driver User's Manual: Software	LLWEB-10064755

	Description of VOCA	Renesas Graphics Library Video Output Checker (VOCA) Driver User’s Manual: Software	LLWEB-10063801
	Description of DISCOM	Renesas Graphics Library Display Output Comparator (DISCOM) Driver User’s Manual: Software	LLWEB-10063802
	Description of DRW2D	Renesas Graphics Library 2D Graphics (DRW2D) Driver User’s Manual: Software	LLWEB-10059472
Porting Layer Guide	Description of porting layer of RGL	Renesas Graphics Library Porting Layer Guide	LLWEB-10035995

2. Notation of Numbers and Symbols

This manual uses the following notation.

Binary	0bXXXXXXXX	(X=0 or 1)
Decimal	XXX	(X=0-9)
Hex	0xXXXXXXXX	(X=0-9,A-F)

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
API	Application Programming Interface.
bpp	bit per pixel
Frame buffer	A region in the memory attached to a window that can be shown on the screen
H/W	Hardware.
Layer	H/W concept of the stackable visual area on the display.
RAM	Random Access Memory.
RLE	Run Length Encoding. TARGA run-length encoded image standard, for easy image compression, supported by the SPEA.
S/W	Software.
SPEA	Sprite Engine A. This is H/W, which controls Sprite data and RLE decoding.
Sprite	A graphical entity which can be moved on the screen independently.
TARGA	Truevision Graphics Adapter
VDCE	Video Data Controller E. This is H/W in RH850/D1x, which controls video output, image synthesis and video input.
WM	Window Manager. This is a driver stack, which enables an abstract access to VDCE driver and SPEA driver.

Table of Contents

1.Overview

1.1 Feature and Scope

The SPEA driver is a driver stack, which enables an abstract access to the device's sprite blit and RLE decoding hardware. The abstraction shall simplify the usage by the application developer and also make it possible to use the same API for different hardware.

1.2 Component Structure

The component structure of SPEA is shown in ***Figure 1-1.***

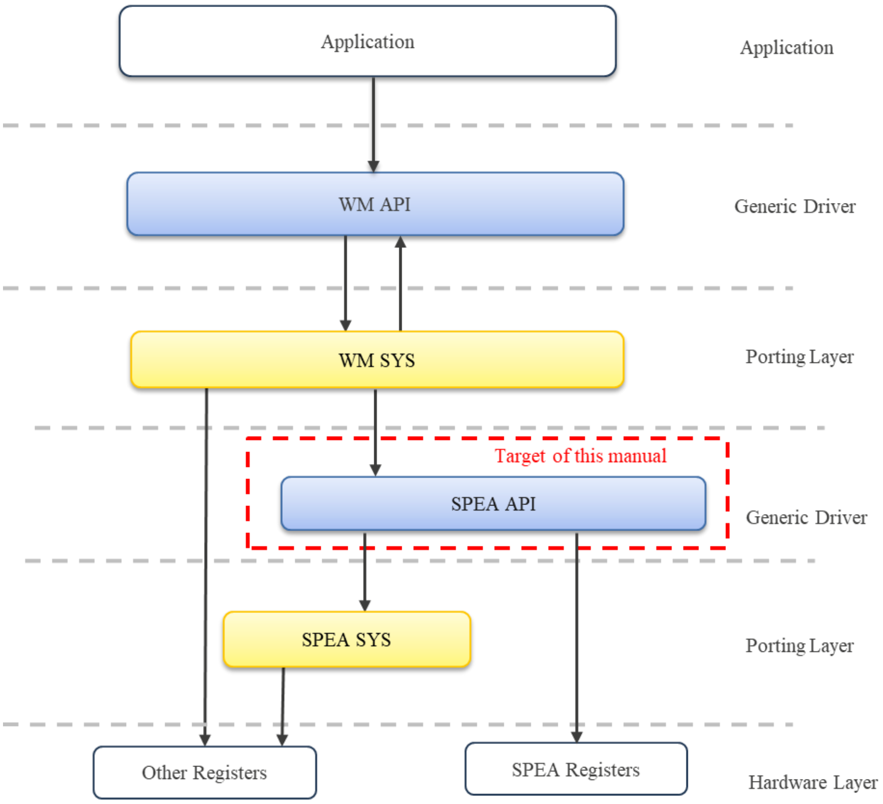

Figure 1-1 Component Structure

For the details of the API, please refer to ***Chapter 4.***

2.Basic Specification

2.1 Summary Specification

The summary of specification is described in ***Table 2-1***

Table 2-1: Summary Specifications

Items	Description
Target LSI	RH850/D1L2(H), RH850/D1M1(H), RH850/D1M1-V2, RH850/D1M1A, RH850/D1M2(H)
Main Feature	• RLE compressed color data in the memory is expanded and delivered to the VDCE as separate pixel color data. • Blit Sprite data, which is up to 16 rectangle areas can be placed anywhere on the layer area.
Semaphore / Mutex	N/A
Interrupts	N/A

2.2 Reserved Word

SPEA Driver uses the following prefixes for avoiding confusion from other software. Prefixes of SPEA is described in *Table 2-2.*

Table 2-2: Prefixes

Prefix	Description
R_SPEA_*	Prefix for SPEA Module.
r_spea_*	

2.3 Interrupt Handler List

None.

2.4 Error Handling

2.4.1 Return code

SPEA driver returns 3 types of error codes.

2.4.1.1 Parameter level

Following errors occur by a cause such as abnormality of parameter. In this case, please set valid parameter again.

- R_SPEA_ERR_RANGE
- R_SPEA_ERR_PARAM
- R_SPEA_ERR_ADDR

2.4.1.2 Timing level

Following errors occur by a cause such as abnormality of execution timing. In this case, please call again after changing to valid state or timing.

- R_SPEA_ERR_TIME
- R_SPEA_ERR_NOT_ACCEPTABLE

2.4.1.3 Hardware level

Following errors occur when unexpected error occurs internally. In this case, please reset the RH850/D1x device.

- R_SPEA_ERR_NG

2.5 State Transition

Table 2-3: State Details

No.	State Name	Description
(1)	DeInit	Specifies that the SPEA Driver is not initialized.
(2)	Idle	Specifies that the operation mode of SPEA Driver is set to a decoding mode.
(3)	Updating	Specifies that SPEA Driver register update is in progress.
(4)	Executing	Specifies that the SPEA Driver is performing an operation such as handling the Sprites for image or RLE engine decompresses the data for image synthesizers.

The image describes the state transition.

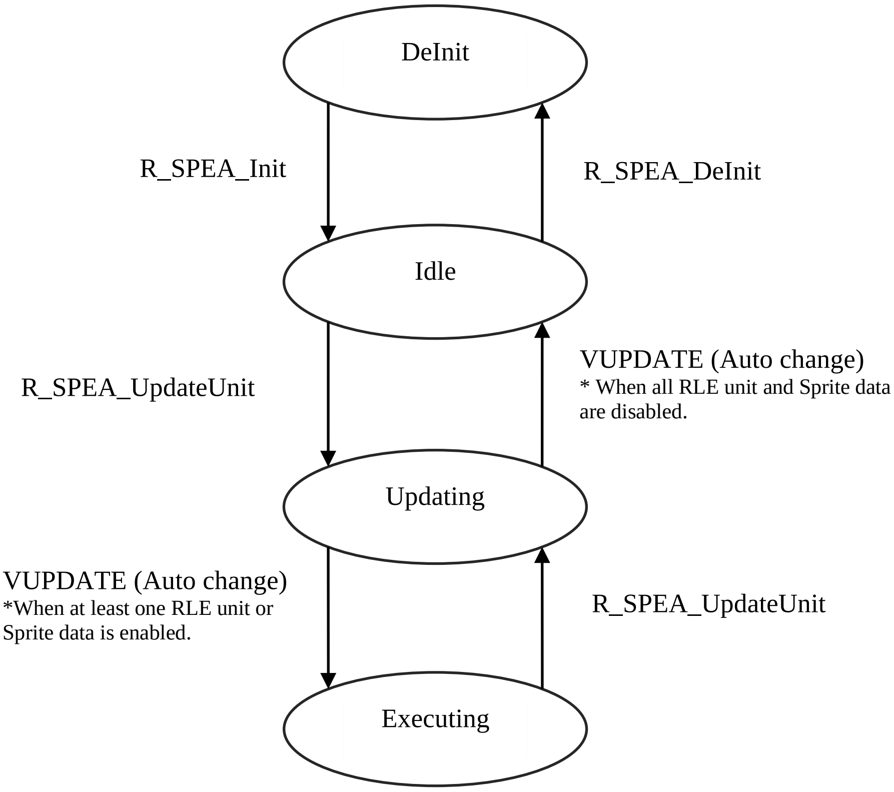

Figure 2-1 State Transition Diagram of SPEA driver

Table 2-4: State Transition Table of SPEA driver

Function Name	State			
	DeInit	Idle	Updating	Executing
R_SPEA_GetVersionStr	OK	OK	OK	OK
R_SPEA_GetMajorVersion	OK	OK	OK	OK
R_SPEA_GetMinorVersion	OK	OK	OK	OK
R_SPEA_Init	OK	NG	NG	NG
R_SPEA_DeInit	OK	OK	NG	NG
R_SPEA_UnitUpdateActive	NG	OK	OK	OK
R_SPEA_UpdateUnit	NG	OK	OK	OK
R_SPEA_UnitCapabilitieseSet	OK	OK	NG	NG
R_SPEA_SetSprite	NG	OK	OK *1	OK
R_SPEA_SpriteEnable	NG	OK	OK *1	OK
R_SPEA_SetSpritePos	NG	OK	OK *1	OK
R_SPEA_SetRle	NG	OK	OK *1	OK
R_SPEA_UnitEnable	NG	OK	OK *1	OK
R_SPEA_SetErrorCallback	NG	OK	OK	OK

*1: These functions wait for transition from Updating state to Idle or Executing state before processing is started.

3.Function Description

3.1 Fundamental Concepts

3.1.1 SPEA unit

RH850/D1x device has the following number of units of the SPEA.

Feature	RH850/D1x Device Name
	D1L2(H), D1M1(H), D1M1A, D1M1-V2, D1M2(H)
Number of SPEA units	1

Almost SPEA API functions have the argument "Unit".
If this "Unit" is described as SPEA unit number, specify the SPEA H/W unit number to be controlled. The range is only 0.

Note: Following SPEA API functions requires to specify the RLE unit number in the argument "Unit".

- R_SPEA_UnitUpdateActive
- R_SPEA_UpdateUnit
- R_SPEA_SetRle
- R_SPEA_UnitEnable

3.1.2 VDCE Image synthesizer

VDCE Image synthesizer acquire data from memory and generates a synthesized image to be displayed. There are four image synthesizers (Scaler0, Scaler1, Image Synthesizer2, Image Synthesizer3).
Following figure shows the VDCE data flow diagram.

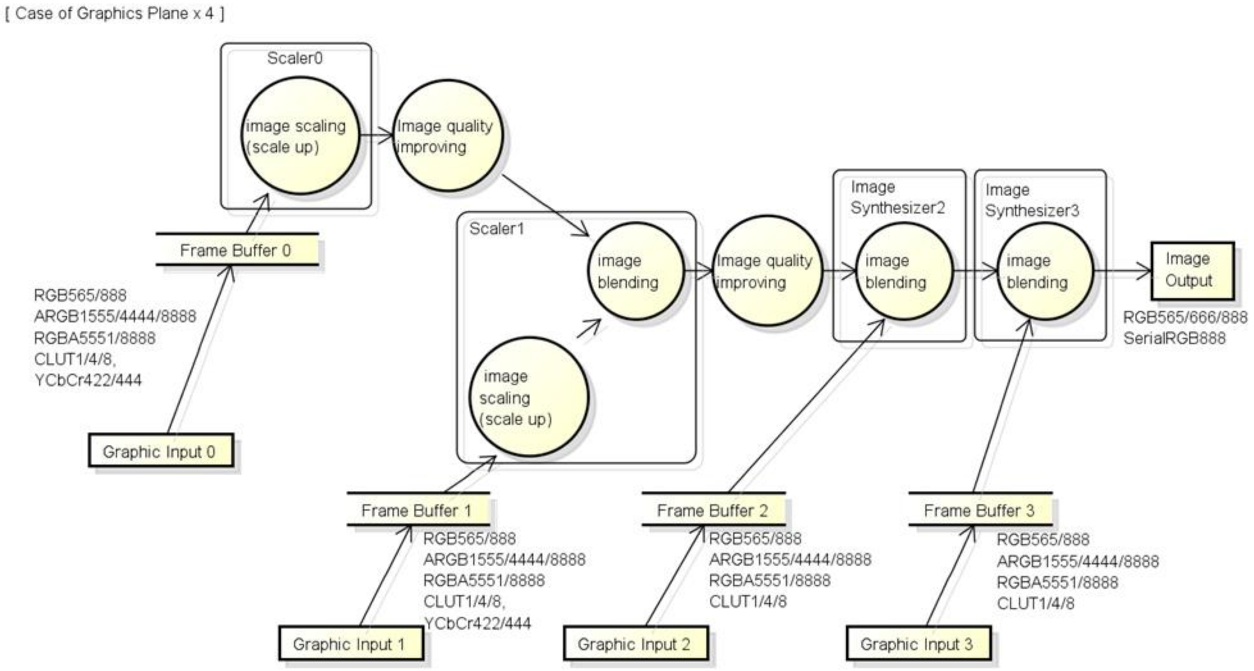

Figure 3-1 VDCE Image synthesizer

If SPEA is used, image synthesizer can select RLE data or Sprite layer instead of normal frame buffer input. SPEA consists of two main blocks, RLE unit and Sprite unit.

Following table shows the connection between REL / Sprite unit number and the image synthesizer.

Table 3-1 Connection between SPEA and VDCE

SPEA		VDCE	
RLE unit	Sprite unit	VDCE0	VDCE1
0 (*1)	3 (*2)	Scaler0	Scaler0
1 (*2)	0 (*1)	Scaler1	Image Synthesizer3
2 (*2)	1 (*1)	Image Synthesizer2	Image Synthesizer2
3 (*2)	2 (*1)	Image Synthesizer3	Scaler1

(*1): Available by default
(*2): Configurable by R_SPEA_UnitCapabilitiesSet depending on the R850/D1x device.

3.1.3 RLE unit

RLE unit is used to decode RLE (run-length encoded) compressed color data in the memory and to deliver that data to the image synthesizer as separate pixel color data. This feature is supposed to be used for drawing like a background images with less update frequency.

SPEA has one or four RLE units depending on the device. There are two RLE engines in one RLE unit. RLE engine0 is used for VDCE0, RLE engine1 is used for VDCE1.
Following figure shows the connection of VDCE and RLE engine.

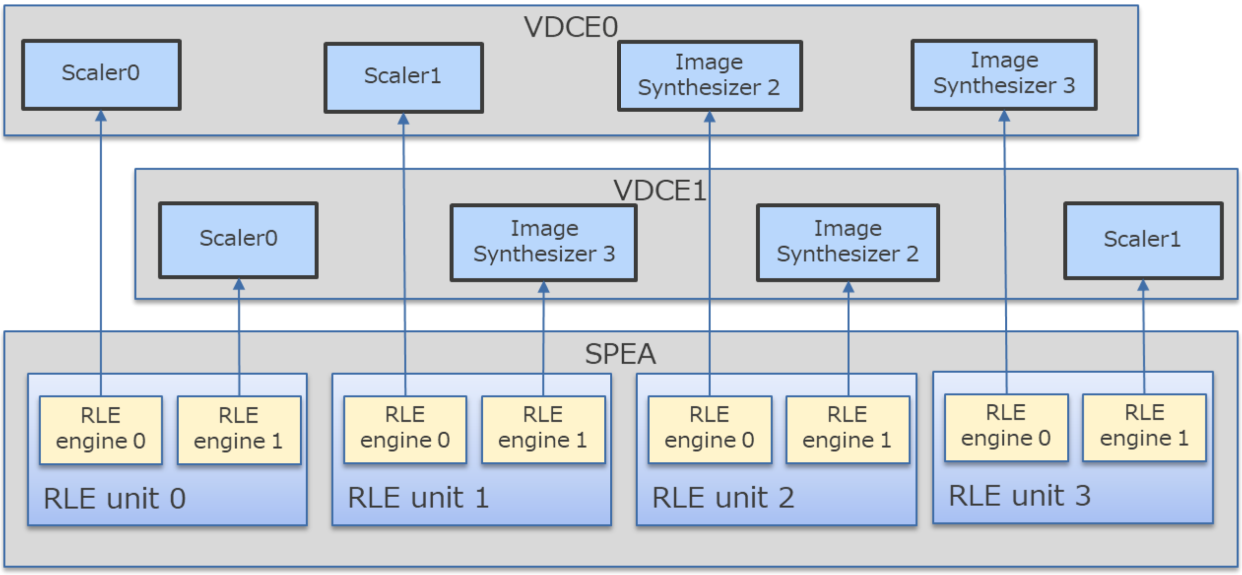

Figure 3-2 Connection of VDCE and RLE engine

An RLE Engine has RLE decoder and virtual frame buffer.
Each image synthesizer decides whether to synthesize virtual frame buffer or normal frame buffer by setting address via VDCE driver.
Following figure shows the block diagram of one RLE unit.

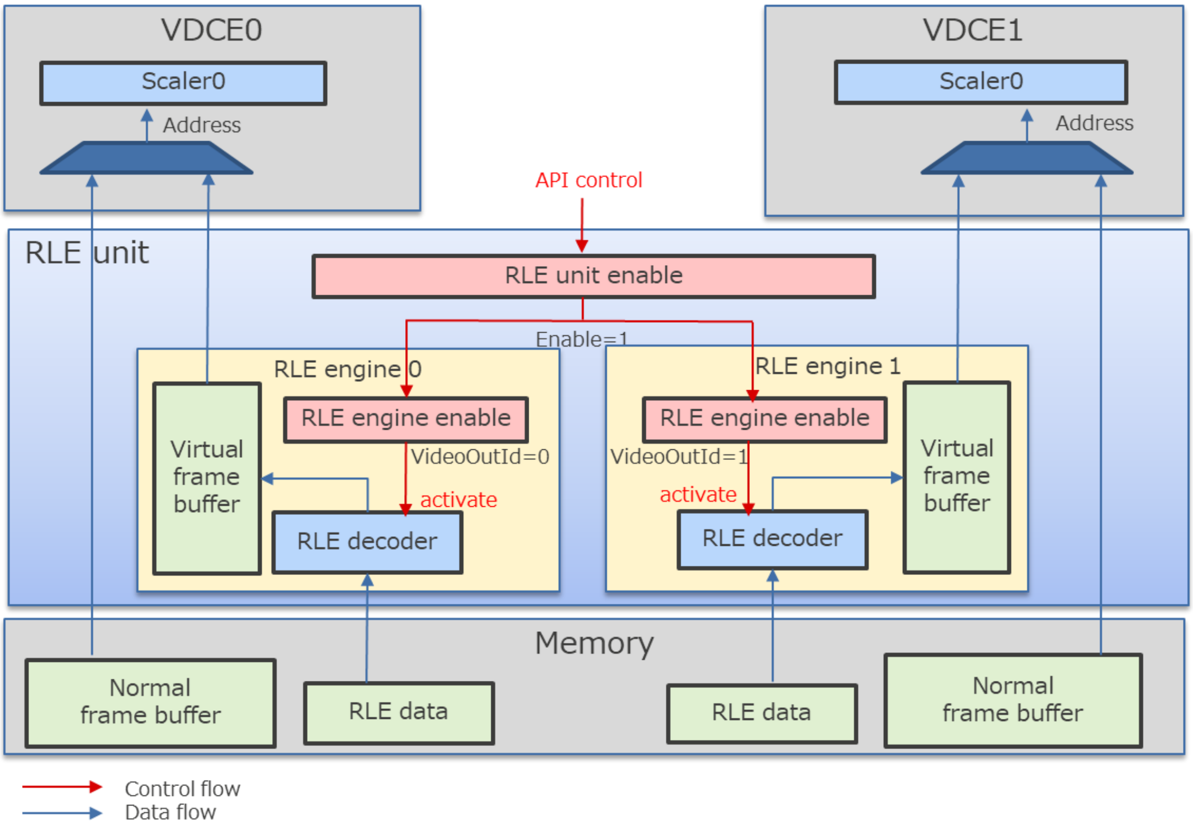

Figure 3-3 RLE unit block diagram

3.1.4 RLE data

When user prepares the RLE compressed data, it needs an additional dummy border. Dummy border requires more than 128 bytes. And, total width of display area and dummy border requires 128 byte-aligned.

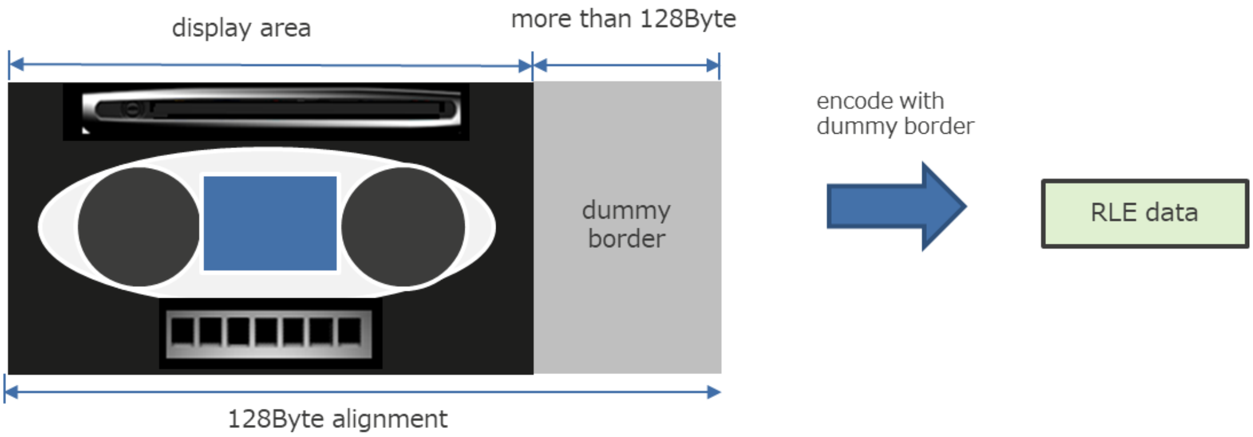

Figure 3-4 Dummy border

For example, width of display area is 720 [pixel] and virtual frame buffer is 32 [bpp] then;
total_width = ((((720 * 4) + 128 + (128 - 1)) / 128) * 128) / 4 = 768 [pixel]
In this case, 48 pixels dummy boarder should be added to image data before RLE data is created.

3.1.5 Sprite unit

Sprite unit blits the Sprite data to virtual frame buffer and delivers that data to the image synthesizer.

SPEA has three or four Sprite units depending on the RH850/D1x device.
Following figure shows the connection of VDCE and Sprite unit.

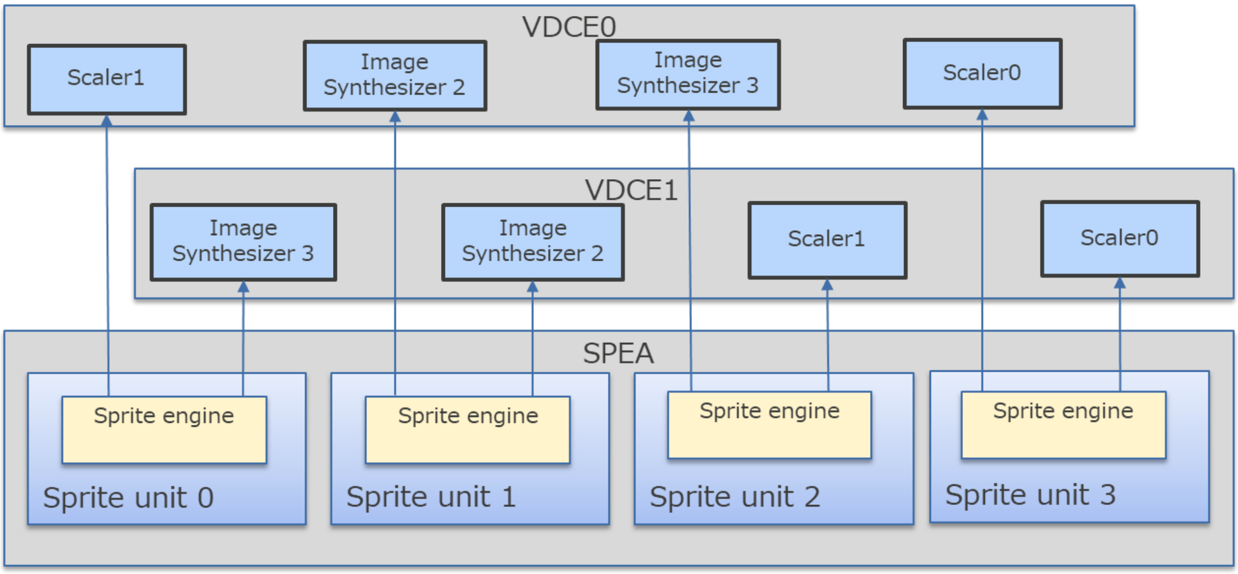

Figure 3-5 VDCE Sprite unit block diagram

A Sprite unit has sprite engine and Virtual frame buffer. Each Sprite Unit is shared by two image synthesizers.

Each image synthesizer decides whether to synthesize virtual frame buffer or normal frame buffer by setting address via VDCE driver.

Following figure shows the block diagram of one Sprite unit.

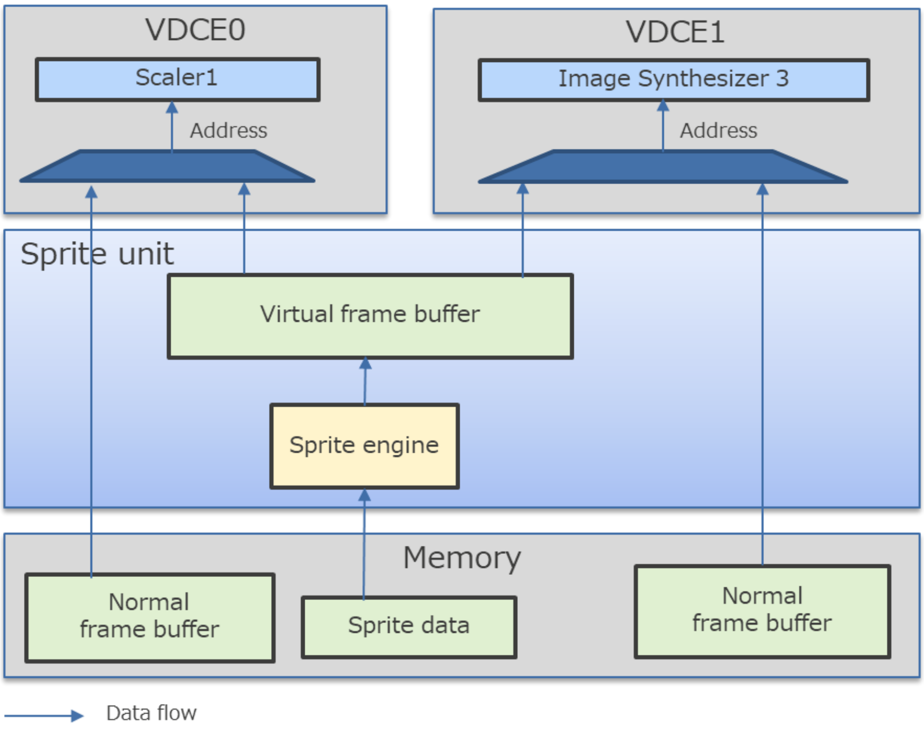

Figure 3-6 Sprite unit block diagram

3.1.6 Sprite data

A Sprite data is specified by rectangle area. Up to 16 Sprite data can be set to one Sprite unit.

When two or more Sprite data areas overlap, the one with the smaller Sprite index number (SpNum) takes precedence. Note that alpha blending between sprite data is not effective within one sprite layer.

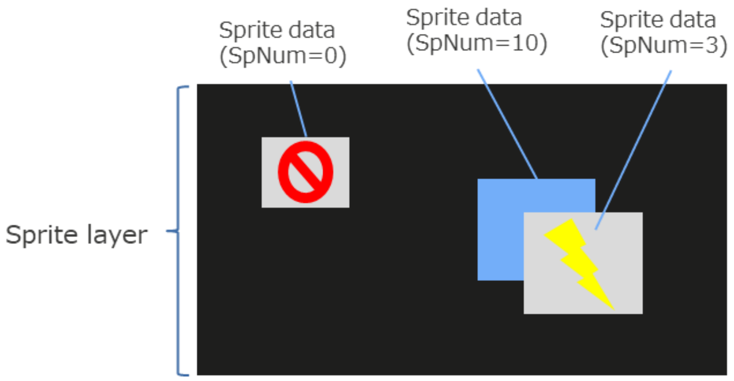

Figure 3-7 Sprite data

3.2 Using the API

3.2.1 Common

3.2.1.1 Initialize / De-initialize

If user wants to change configuration of Sprite unit and RLE unit, R_SPEA_UnitCapabilitieseSet can be called before SPEA API is initialized,
R_SPEA_Init should be called before user starts using Sprite unit or RLE unit.
R_SPEA_DeInit should be called after user finishes using Sprite unit and RLE unit.

3.2.1.2 Register Update

SPEA registers set by following functions need the update control for synchronous updating.

- R_SPEA_SetSprite
- R_SPEA_SpriteEnable
- R_SPEA_SetSpritePos
- R_SPEA_SetRle
- R_SPEA_UnitEnable

The update procedure is as follows.

1. R_SPEA_UpdateUnit is executed.
2. VUPDATE0 or VUPDATE1 signal occurs.

Regarding of VUPDATE0 and VUPDATE1, SPEA driver does not control directly. User (or SPEA Porting layer) should control registers for VUPDATE0 and VUPDATE1. See H/W User's manual section 37.6 for the detail.

3.2.2 RLE unit

3.2.2.1 RLE enable / disable

SPEA APIs can control enable / disable of each RLE unit and each RLE engine. When Both RLE unit and RLE engine are enabled, RLE decoder is activated and outputs decoded data to virtual frame buffer.

Table 3-2 RLE units supported color formats

Control target	API Function	Argument (Set value)
RLE unit	R_SPEA_UnitEnable	Enable (1: Enable, 0: Disable)
RLE engine 0	R_SPEA_SetRle	VideoOutId (0: Enable, 3-15: Disable)
RLE engine 1	R_SPEA_SetRle	VideoOutId (1: Enable, 3-15: Disable)

3.2.2.2 RLE color format

Following table shows the RLE unit supported color formats. Color mode (bpp) of RLE data is set by R_SPEA_SetRle. Color format of virtual frame buffer is set to VDCE driver.

Table 3-3 RLE unit supported color formats

RLE data (before decoding)		Virtual frame buffer (after decoding)	
Color format	SPEA color mode selection	VDCE color format selection	bpp
RGB888	R_SPEA_CM_24BPP	ARGB8888 or RGB888 (A = 0xFF fixed.)	32 bpp
RGB666	R_SPEA_CM_18BPP	ARGB8888 or RGB888 (R, G, B is expanded to 8 bits. A = 0xFF fixed.)	32 bpp
One 8 bpp color index in one byte.	R_SPEA_CM_8BPP	CLUT8	8 bpp
Two 4 bpp color indices in one byte.	R_SPEA_CM_8BPP	CLUT4	4 bpp
Eight 1 bpp color indices in one byte.	R_SPEA_CM_8BPP	CLUT1	1 bpp

See H/W User’s manual section 43.3.2 about detail of color format.

3.2.2.3 RLE data address

RLE data can be stored both RAM and Flash. The stored address is set with MemStart of R_SPEA_SetRle. Start address of RLE data must be 8 byte-aligned.
The start address of virtual frame buffer is set with VirtStart of R_SPEA_SetRle. It should be fixed at 0x30000000.

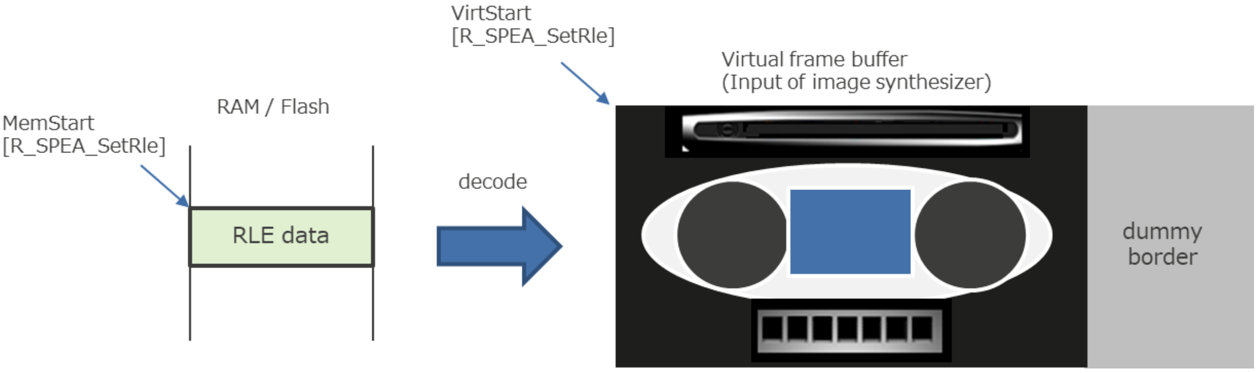

Figure 3-8 SPEA setting for RLE

Following figure shows VDCE setting to synthesize RLE decoding data. The start address should be fixed at 0x30000000.

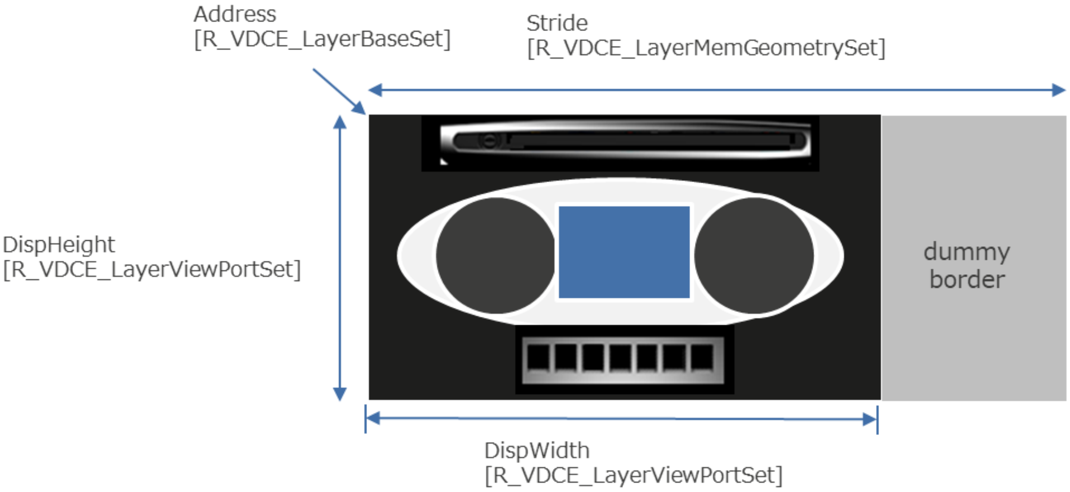
Address
[R_VDCE_LayerBaseSet]
Stride
[R_VDCE_LayerMemGeometrySet]
DispHeight
[R_VDCE_LayerViewPortSet]
dummy
border
DispWidth
[R_VDCE_LayerViewPortSet]

Figure 3-9 VDCE setting for RLE

3.2.3 Sprite unit

3.2.3.1 Sprite enable / disable

R_SPEA_SetSprite can control enable / disable of the sprite data with Sprite data setting.
R_SPEA_SpriteEnable can control enable / disable of each sprite data.

3.2.3.2 Sprite color format

SPEA API only requires the information of bpp of color format. It is set with DispBpp of R_SPEA_SetSprite and R_SPEA_SetSpritePos.

Detailed color format is specified for image synthesizer via VDCE driver. Sprite data should consist of the same color format specified for the image synthesizer.

Image synthesizer read data with value 0 if areas of the sprite layer without enabled sprite data. So, these data are processed as fully transparent color to the Image synthesizer if color format is RGB format with alpha value (ARGB8888 / ARGB4444 / ARGB1555 / RGBA5551 / RGBA8888).

Note that Scaler0 does not have alpha blending feature. Thus, areas of the sprite layer without enabled sprite data looks black if Scaler0 is used as Sprite and RGB format is selected.

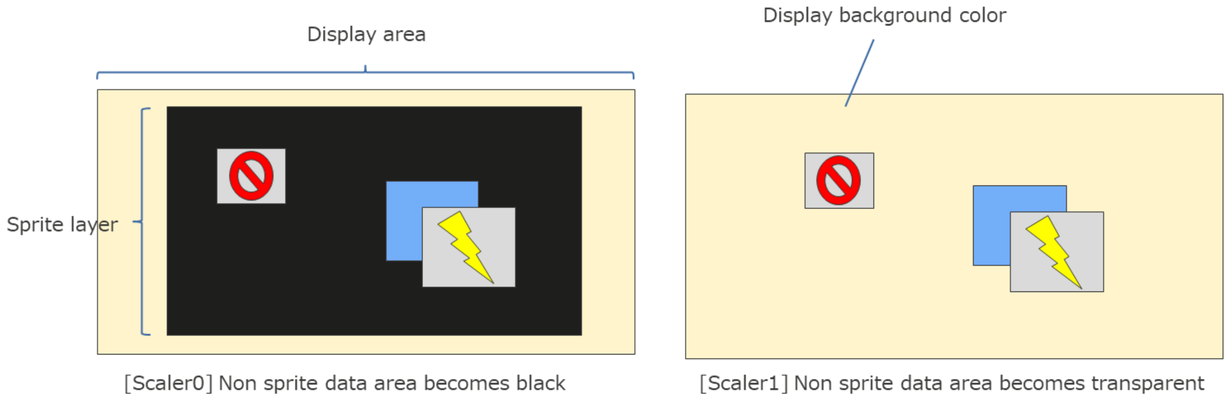
Display area
Display background color
Sprite layer
[Scaler0] Non sprite data area becomes black
[Scaler1] Non sprite data area becomes transparent

Figure 3-10 No sprite data area

3.2.3.3 Sprite data blit position

Sprite data can be stored both RAM and Flash. The stored address is set with SrcAddr of R_SPEA_SetSprite. Start address of Sprite data must be 8 byte-aligned.

The start address of virtual frame buffer is fixed as 0x30000000.

Sprite data size is set with Width and Height of R_SPEA_SetSprite.

Blit position of sprite data is set with PosX and PosY of R_SPEA_SetSprite. PosX and PosY are also change by R_SPEA_SetSpritePos. X/Y position indicates top-left position on the virtual frame buffer.

Following figure shows SPEA setting to blit sprite data.

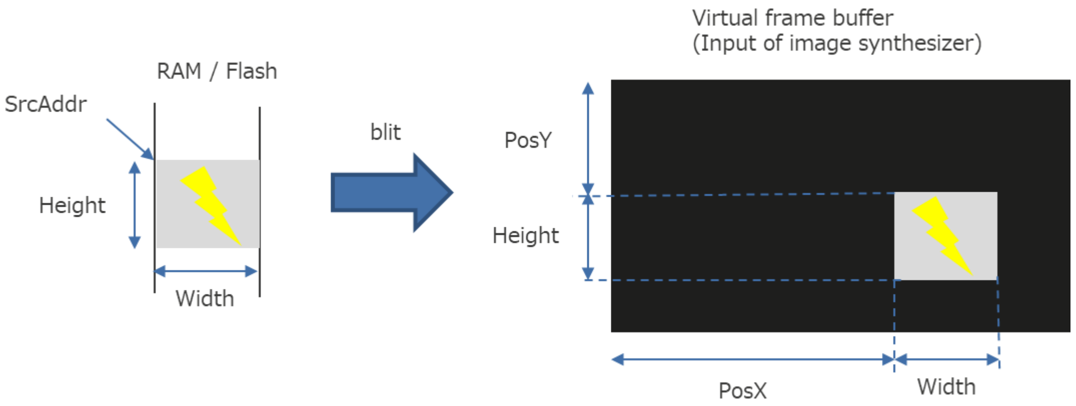

Figure 3-11 SPEA setting for Sprite

PosX and Width are set in pixel units and should be multiple of 8 bytes (i.e. multiple of 2 pixels in case of 32 bpp color format). The sum of PosX and Width should not exceed 8192 bytes. (i.e. the maximum value is 2048 pixels in case of 32 bpp color format).
Regarding PosX and Width, if the alignment restriction is not satisfied, SPEA functions are rounded the values.
PosY and Height have no multiple restriction. The sum of PosY and Height should not exceed 8192 pixels.

Table 3-4 Range and alignment

Parameter	Setting Unit	Range		Alignment
		Min	Max	
SrcAdr	Byte	-	-	8 Byte
PosX	Pixel	0	(8184 * 8) / DispBpp	(8 * 8) / DispBpp
Width	Pixel	0	(8184 * 8) / DispBpp	(8 * 8) / DispBpp
(PosX + Width)	Pixel	0	(8192 * 8) / DispBpp	(8 * 8) / DispBpp
PosY	Pixel	0	8191	-
Height	Pixel	0	2047	-
(PosX + Hight)	Pixel	0	8192	-

Following figure shows VDCE setting to synthesize Sprite layer. Stride should be the value converted from 8192 bytes in bpp.

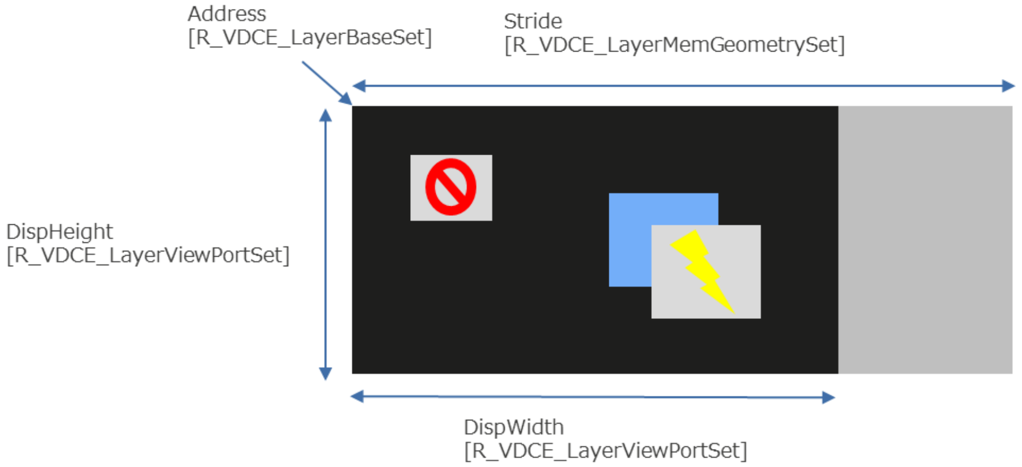
Address
[R_VDCE_LayerBaseSet]
Stride
[R_VDCE_LayerMemGeometrySet]
DispHeight
[R_VDCE_LayerViewPortSet]
DispWidth
[R_VDCE_LayerViewPortSet]

Figure 3-12 VDCE setting for Sprite

3.3 Device difference

The following table shows the number of units depending on the device.

Table 3-5 Number of Units

Number of units	RH850/D1x Device Name	
	D1M1A, D1M1-V2	D1L2(H), D1M1(H), D1M2(H)
Sprite units	4	3
RLE units	4	1

In case of RH850/D1M1A and RH850/D1M1-V2

- Selectable RLE unit number is 0 to 3 (R_SPEA_RLE0 / R_SPEA_RLE1 / R_SPEA_RLE2 / R_SPEA_RLE3).
- Selectable Sprite unit number is 0 to 3 (R_SPEA_SU0 / R_SPEA_SU1 / R_SPEA_SU2 /R_SPEA_SU3).

In case of RH850/D1L2(H), RH850/D1M1(H) and RH850/D1M2(H)

- Selectable RLE unit number is only 0 (R_SPEA_RLE0).
- Selectable Sprite unit number is 0 to 2 (R_SPEA_SU0 / R_SPEA_SU1 / R_SPEA_SU2).

The following table shows the function differences depending on the device.

Table 3-6 APIs supported by SPEA driver

Function	RH850/D1x Device Name	
	D1M1A, D1M1-V2	D1L2(H), D1M1(H), D1M2(H)
Selection of Sprite and RLE feature. [Function] R_SPEA_UnitCapabilitiesSet	OK	NG *1

*1: The function will not fail, but the feature will not work.

3.4 Header File List

Table 3-7 Header File List

No.	Header File Name	Description
1	r_spea_api.h	Header file of the SPEA driver.
2	r_typedefs.h	Predefined data types.

4.Functions

4.1 Function List

This section describes about the SPEA API functions which are in *Table 4-1* and executable state of each function is described in the specification of each function.

Table 4-1 Function List

Function Name	Purpose
R_SPEA_GetVersionStr	Get the driver version number in human readable form (string).
R_SPEA_GetMajorVersion	Get the driver major version number.
R_SPEA_GetMinorVersion	Get the driver minor version number.
R_SPEA_Init	Driver initialization function.
R_SPEA_DeInit	Driver de-initialization function.
R_SPEA_UnitUpdateActive	Get the information if a register update is ongoing or finished for a given unit.
R_SPEA_UpdateUnit	Set the register update request for the given unit.
R_SPEA_UnitCapabilitiesSet	Configure the video output layers to decode RLE compressed data or to show Sprite.
R_SPEA_SetSprite	Set configuration for the given Sprite.
R_SPEA_SpriteEnable	Enable or disable a given Sprite within one unit.
R_SPEA_SetSpritePos	Set or change just the position (on the virtual layer) of the given Sprite.
R_SPEA_SetRle	Set the configuration for the given RLE.
R_SPEA_UnitEnable	Enable or disable a given RLE or Sprite unit.
R_SPEA_SetErrorCallback	Set a user function as error handler.

4.2 SPEA API Functions

This chapter describes the application interface functions, which are required for general use of the driver, but which are related to a specific functionality of the macro itself.

4.2.1 Common Functions (Sprite and RLE)

4.2.1.1 R_SPEA_GetVersionStr

Function Prototypes

```
const int8_t *R_SPEA_GetVersionStr(void)
```

Parameter

None.

Return Codes

Receive String - Pointer to receive string destination.

Description

This function gets the driver version number in human readable form.

Reentrancy

Reentrant.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

None.

4.2.1.2 R_SPEA_GetMajorVersion

Function Prototypes

```
const uint16_t R_SPEA_GetMajorVersion(void)
```

Parameter

None

Return Codes

Major version number.

Description

This function gets the driver major version number.

Reentrancy

Reentrant.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

None.

4.2.1.3 R_SPEA_GetMinorVersion

Function Prototypes

```
const uint16_t R_SPEA_GetMinorVersion(void)
```

Parameter

None.

Return Codes

Minor version number.

Description

This function gets the driver minor version number.

Reentrancy

Reentrant.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

None.

4.2.1.4 R_SPEA_Init

Function Prototypes

```
r_spea_Error_t R_SPEA_Init(const uint32_t Unit)
```

Input Parameter

Table 4-2 Input parameter of R_SPEA_Init

Parameter	Description
Unit	Specifies the SPEA unit number.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_NG	- Some error occurs in the porting layer.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.
R_SPEA_ERR_RANGE	- The unit-number was outside the range.

Description

This function initializes the SPEA driver.
This function calls R_SPEA_SYS_HardwareInit to initialized other than SPEA H/W.

If the function successfully executes, the return code will be R_SPEA_ERR_OK and the state of the SPEA unit will be in the Idle state.

Reentrancy

Non-reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t

4.2.1.5 R_SPEA_DeInit

Function Prototypes

```
r_spea_Error_t R_SPEA_DeInit(const uint32_t Unit)
```

Input Parameter

Table 4-3 Input parameter of R_SPEA_DeInit

Parameter	Description
Unit	Specifies the SPEA unit number.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_NG	- Some error occurs in the porting layer.
R_SPEA_ERR_RANGE	- The unit-number was outside the range.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function de-initializes the SPEA driver.
This function calls R_SPEA_SYS_HardwareDeInit to de-initialize other than SPEA H/W.

If the function successfully executes, the return code will be R_SPEA_ERR_OK and the state of the SPEA unit will be in the DeInit state.

Reentrancy

Non- Reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Precondition

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t**.**

4.2.1.6 R_SPEA_UnitUpdateActive

Function Prototypes

```
uint32_t R_SPEA_UnitUpdateActive(const uint32_t      Unit,
                                 const r_spea_Unit_t SpUnit)
```

Input Parameter

Table 4-4 Input parameter of R_SPEA_UnitUpdateActive

Parameter	Description
Unit	Specifies the RLE unit number in case of RLE unit control. Specifies the SPEA unit number in case of Sprite unit control.
SpUnit	Specifies RLE / Sprite unit number from followings. R_SPEA_RLE0 R_SPEA_RLE1 R_SPEA_RLE2 R_SPEA_RLE3 R_SPEA_SU0 R_SPEA_SU1 R_SPEA_SU2 R_SPEA_SU3

Input -Output Parameter

None

Output parameter

None

Return Codes

==0	- Update finished or update is not started.
!= 0	- Register update in progress.

Description

This function gets the information if a register update is ongoing or finished for the specified RLE / Sprite unit. Selectable RLE / Sprite unit number is depending on the RH850/D1x device. See ***Table 3-5***.

If error occurs in this function, the return code is 0 and following error code is notified by error callback.

R_SPEA_ERR_RANGE	- Parameter is invalid.
R_SPEA_ERR_NOT_ACCEPTABLE	- SEPA unit status is invalid.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Unit_t

4.2.1.7 R_SPEA_UpdateUnit

Function Prototypes

```
r_spea_Error_t R_SPEA_UpdateUnit (const uint32_t       Unit,
                                  const r_spea_Unit_t  SpUnit,
                                  const uint8_t        VSync)
```

Input Parameter

Table 4-5 Input parameter of R_SPEA_UpdateUnit

Parameter	Description
Unit	Specifies the RLE unit number in case of RLE unit control. Specifies the SPEA unit number in case of Sprite unit control.
SpUnit	Specifies RLE / Sprite unit number from the following. R_SPEA_RLE0 R_SPEA_RLE1 R_SPEA_RLE2 R_SPEA_RLE3 R_SPEA_SU0 R_SPEA_SU1 R_SPEA_SU2 R_SPEA_SU3
VSync	Specifies VUPDATE0 or VUPDATE1 to be used for update. 0: VUPDATE0 1: VUPDATE1

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No Error has occurred.
R_SPEA_ERR_RANGE	- A parameter value was outside the range.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.
R_SPEA_ERR_PARAM	- A parameter provided to a function was incorrect.

Description

This function sets the register update request for the specified RLE or Sprite unit.
If the function successfully executes, the return code will be R_SPEA_ERR_OK and the state of the SPEA unit will be in the Updating state.
The actual register update will be executed after VUPDATE0 or VUPDATE1 signal occurs.

Selectable RLE / Sprite unit number is depending on the RH850/D1x device. See ***Table 3-5***.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Asynchronous

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status condition.

See also

r_spea_Error_t
r_spea_Unit_t

4.2.1.8 R_SPEA_UnitCapabilitiesSet

Function Prototypes

```
r_spea_Error_t R_SPEA_UnitCapabilitiesSet(const r_spea_UnitCapbs_t  Unit0,
                                          const r_spea_UnitCapbs_t  Unit1,
                                          const r_spea_UnitCapbs_t  Unit2,
                                          const r_spea_UnitCapbs_t  Unit3)
```

Input Parameter

Table 4-6 Input parameter of R_SPEA_UnitCapablitiesSet

Parameter	Description
Unit0	Selects RLE unit 0 or Sprite unit 3. R_SPEA_SPEACAPBS_RLE : RLE unit 0 (default) R_SPEA_SPEACAPBS_SPRITE: Sprite unit 3 Selected unit is connected to VDCE0 Scaler0 and VDCE1 Scaler0.
Unit1	Selects RLE unit 1 or Sprite unit 0 R_SPEA_SPEACAPBS_RLE : RLE unit 1 R_SPEA_SPEACAPBS_SPRITE: Sprite unit 0 (default) Selected unit is connected to VDCE0 Scaler1 and VDCE1 Image Synthesizer3.
Unit2	Selects RLE unit 2 or Sprite unit 1 R_SPEA_SPEACAPBS_RLE : RLE unit 2 R_SPEA_SPEACAPBS_SPRITE: Sprite unit 1 (default) Selected unit is connected to VDCE0 Image Synthesizer2 and VDCE1 Image Synthesizer2.
Unit3	Selects RLE unit 3 or Sprite unit 2 R_SPEA_SPEACAPBS_RLE : RLE unit 3 R_SPEA_SPEACAPBS_SPRITE: Sprite unit 2 (default) Selected unit is connected to VDCE0 Image Synthesizer3 and VDCE1 Scaler1.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_PARAM	- An error has occurred when invalid parameter is given.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function selects the valid Sprite unit number and RLE unit number.
The validity of this function is depending on RH850/D1x device. See *Table 3-6*.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status condition.

See also

r_spea_Error_t
r_spea_UnitCapabs_t

4.2.1.9 R_SPEA_SetErrorCallback

Function Prototypes

```
void R_SPEA_SetErrorCallback(void (*ErrorCallback) (uint32_t Unit, uint32_t Error))
```

Input Parameter

Table 4-7 Input parameter of R_SPEA_SetErrorCallback

Parameter	Description
ErrorCallback	Specifies the function that is called when an error occurred (can be set R_NULL).

Table 4-8 Input parameter of ErrorCallback

Parameter	Description
Unit	SPEA unit number where the error occurred.
Error	Error code

Input -Output Parameter

None

Output parameter

None

Return Codes

None

Description

This function installs the Error callback function.
When some error occurs in SPEA driver, SPEA driver invokes the callback function.
The error callback is notified during SPEA unit is not DeInit state.
The installed error callback can be uninstalled by R_SPEA_Init or R_NULL setting in this function.

Since there is no asynchronous processing in SPEA driver, it is possible to detect an error by checking the return code of each API function.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t

4.2.2 Sprite functions

4.2.2.1 R_SPEA_SetSprite

Function Prototypes

```
r_spea_Error_t R_SPEA_SetSprite(const uint32_t               Unit,
                                const uint8_t                SpEng,
                                const uint8_t                SpNum,
                                const r_spea_SpriteConfig_t  *SpConf,
                                const uint8_t                DispBpp)
```

Input Parameter

Table 4-9 Input parameter of R_SPEA_SetSprite

Parameter	Description
Unit	Specifies the SPEA unit number.
SpEng	Specifies Sprite unit number.
SpNum	Specifies index number of Sprite data. Supported range is 0-15.
SpConf	Specifies pointer to configuration of Sprite data.
DispBpp	Specifies bits per pixel for the color format. Supported range is 1-32. It should be a power of 2.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_NG	- Some error occurs in the porting layer.
R_SPEA_ERR_RANGE	- The unit-number was outside the range.
R_SPEA_ERR_PARAM	- A parameter provided to a function was incorrect.
R_SPEA_ERR_ADDR	- Start address of Sprite data is not 8 byte-aligned.
R_SPEA_ERR_TIME	- Timeout has happened while waiting for update completion.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function sets configuration for the specified Sprite data.
Selectable Sprite unit number is depending on the RH850/D1x device. See ***Table 3-5***.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t
r_spea_SpriteConfig_t

4.2.2.2 R_SPEA_SpriteEnable

Function Prototypes

```
r_spea_Error_t R_SPEA_SpriteEnable (const uint32_t   Unit,
                                     const uint8_t   SpEng,
                                     const uint8_t   SpNum,
                                     const uint8_t   Enable)
```

Input Parameter

Table 4-10 Input parameter of R_SPEA_SpriteEnable

Parameter	Description
Unit	Specifies the SPEA unit number.
SpEng	Specifies Sprite unit number.
SpNum	Specifies index number of Sprite data. Supported range is 0-15.
Enable	Specifies whether enable or disable the Sprite data. 0: disable 1: enable

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_NG	- Some error occurs in the porting layer.
R_SPEA_ERR_RANGE	- An error has occurred, a parameter value outside the range.
R_SPEA_ERR_PARAM	- A parameter provided to a function was incorrect.
R_SPEA_ERR_TIME	- Timeout has happened while waiting for update completion.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function enables or disables the blit of Sprite data.
Selectable Sprite unit number is depending on the RH850/D1x device. See ***Table 3-5***.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.
The function R_SPEA_SetSprite shall be called before executing this function.

See also

r_spea_Error_t

4.2.2.3 R_SPEA_SetSpritePos

Function Prototypes

```
r_spea_Error_t R_SPEA_SetSpritePos(const uint32_t      Unit,
                                   const uint8_t       SpEng,
                                   const uint8_t       SpNum,
                                   const uint16_t      PosX,
                                   const uint16_t      PosY,
                                   const uint8_t       DispBpp)
```

Input Parameter

Table 4-11 Input parameter of R_SPEA_SetSpritePos

Parameter	Description
Unit	Specifies the SPEA unit number.
SpEng	Specifies Sprite unit number.
SpNum	Specifies Sprite index. Supported range is 0-15.
PosX	Specifies horizontal position of virtual frame buffer in pixels.
PosY	Specifies vertical position of virtual frame buffer in pixels. .
DispBpp	Specifies bits per pixel for the color format. Supported range is 1-32. It should be a power of 2.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_RANGE	- An error has occurred: a parameter value outside the range.
R_SPEA_ERR_PARAM	- An error has occurred: an invalid parameter is given.
R_SPEA_ERR_TIME	- Timeout has happened while waiting for update completion.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function updates the blit position of the specified Sprite data.
Selectable Sprite unit number is depending on the RH850/D1x device. See ***Table 3-5***.
For the detail of range and alignment information of PosX and PosY, see ***Table 3-4***.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.
The function R_SPEA_SetSprite shall be called before executing this function.

See also

r_spea_Error_t

4.2.3 RLE functions

4.2.3.1 R_SPEA_SetRle

Function Prototypes

```
r_spea_Error_t R_SPEA_SetRle(const uint32_t            Unit,
                             const uint8_t             RleNum,
                             const r_spea_RleConfig_t  *RleConf)
```

Input Parameter

Table 4-12 Input parameter of R_SPEA_SetRle

Parameter	Description
Unit	Specifies the RLE unit number.
RleNum	Specifies the RLE engine number. Supported range is 0-1.
RleConf	Specifies the configuration for RLE engine.

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_PARAM	- An error has occurred: an invalid parameter is given.
R_SPEA_ERR_RANGE	- An error has occurred: a parameter is outside the range.
R_SPEA_ERR_ADDR	- Start address is invalid.
R_SPEA_ERR_TIME	- Timeout has happened while waiting for update completion.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function sets configuration of the specified RLE engine.
Selectable RLE unit number is depending on the RH850/D1x device. See ***Table 3-5***.

VideoOutID is the member of RleConf which always make sure to assign different VideoOutID for two RLE engines.
This function checks two VideoOutID and returns with an error if they are the same.
Note that default VideoOutID value of all RLE engines is 0.

Reentrancy

Non- reentrant.
User should control not to re-enter.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t
r_spea_RleConfig_t

4.2.3.2 R_SPEA_UnitEnable

Function Prototypes

```
r_spea_Error_t R_SPEA_UnitEnable(const uint32_t        Unit,
                                  const r_spea_Unit_t   SpUnit,
                                  const uint8_t         Enable)
```

Input Parameter

Table 4-13 Input parameter of R_SPEA_UnitEnable

Parameter	Description
Unit	Specifies the RLE unit number.
SpUnit	Specifies RLE unit number from the following. R_SPEA_RLE0 R_SPEA_RLE1 R_SPEA_RLE2 R_SPEA_RLE3
Enable	Specifies whether enable or disable the RLE unit. 0: disable 1: enable

Input -Output Parameter

None

Output parameter

None

Return Codes

R_SPEA_ERR_OK	- No error has occurred.
R_SPEA_ERR_NG	- Some error occurs in the porting layer.
R_SPEA_ERR_RANGE	- An error has occurred, a parameter value outside the range.
R_SPEA_ERR_PARAM	- A parameter provided to a function was incorrect.
R_SPEA_ERR_TIME	- Timeout has happened while waiting for update completion.
R_SPEA_ERR_NOT_ACCEPTABLE	- This function is called in an incorrect state.

Description

This function enables or disables the specified RLE unit.
Selectable RLE unit number is depending on the RH850/D1x device. See ***Table 3-5***.

Reentrancy

Non- Reentrant.
User should control not to re-enter.

Sync/Async

Synchronous.

Call from Interrupt

Prohibited.

Preconditions

See ***Table 2-4*** about status conditions.

See also

r_spea_Error_t
r_spea_Unit_t

5.Types

5.1 Basic Type

This section shows the basic type used in this library.

Table 5-1 Basic Type

Types	Definition	Basic types
char_t	typedef char char_t	signed char
int8_t	typedef signed char int8_t	signed char
int16_t	typedef signed short int16_t	signed short
int32_t	typedef signed int int32_t	signed int
int64_t	typedef signed long long int64_t	signed long long
uint8_t	typedef unsigned char uint8_t	unsigned char
uint16_t	typedef unsigned short uint16_t	unsigned short
uint32_t	typedef unsigned int uint32_t	unsigned int
uint64_t	typedef unsigned long long uint64_t	unsigned long long
float32_t	typedef float float32_t	float
float64_t	typedef double float64_t	double

5.2 Definition

This section shows the definition value used in SPEA API.

Table 5-2 Definition of SPEA API

Name	Description
R_SPEA_VERSION_HI	MSB byte of the version information. It is major version information. This value is changed with release version.
R_SPEA_VERSION_LO	LSB byte of the version information. It is minor version information. This value is changed with release version.

5.3 Enumerated type

This section shows the enumerated value used in SPEA API Function.

5.3.1 r_spea_Error_t

Description

SPEA error code. If an error occurs, this enumerator gives information about the reason.

Definition

```
typedef enum {
    R_SPEA_ERR_OK                 = 0x00u,
    R_SPEA_ERR_NG                 = 0x01u,
    R_SPEA_ERR_RANGE              = 0x02u,
    R_SPEA_ERR_PARAM              = 0x03u,
    R_SPEA_ERR_TIME               = 0x04u,
    R_SPEA_ERR_ADDR               = 0x05u,
    R_SPEA_ERR_NOT_ACCEPTABLE     = 0x06u,
} r_spea_Error_t
```

Table 5-3 Enumerator of r_SPEA_Error_t

Name	Description
R_SPEA_ERR_OK	No error occurred.
R_SPEA_ERR_NG	An error has occurred, but no specific error code is defined for it.
R_SPEA_ERR_RANGE	Parameter is outside the range.
R_SPEA_ERR_PARAM	Parameter provided to a function is incorrect.
R_SPEA_ERR_TIME	Timeout occurred.
R_SPEA_ERR_ADDR	Wrong address (not aligned correctly).
R_SPEA_ERR_ NOT_ACCEPTABLE	The function is called in an incorrect status.

See also

None

5.3.2 r_spea_Color_t

Description

Configuration data for RLE color modes.

Definition

```
typedef enum {
  R_SPEA_CM_8BPP  = 0x00u,
  R_SPEA_CM_18BPP = 0x01u,
  R_SPEA_CM_24BPP = 0x02u,
  R_SPEA_CM_LIMIT = 0x03u,
} r_spea_Color_t
```

Table 5-4 Enumerator of r_spea_Color_t

Name	Description
R_SPEA_CM_8BPP	8 bits per pixel.
R_SPEA_CM_18BPP	18 bits per pixel.
R_SPEA_CM_24BPP	24 bits per pixel.
R_SPEA_CM_LIMIT	delimiter.

See also

r_spea_RleConfig_t

5.3.3 r_spea_Unit_t

Description

List of units in one Sprite engine macro.

Definition

```
typedef enum {
    R_SPEA_RLE0       = 0x00u,
    R_SPEA_RLE1       = 0x01u,
    R_SPEA_RLE2       = 0x02u,
    R_SPEA_RLE3       = 0x03u,
    R_SPEA_SU_OFFSET  = 0x04u,
    R_SPEA_SU0        = 0x05u,
    R_SPEA_SU1        = 0x06u,
    R_SPEA_SU2        = 0x07u,
    R_SPEA_SU3        = 0x08u,
    R_SPEA_SU_LAST    = 0x09u,
} r_spea_Unit_t
```

Table 5-5 Enumerator of r_spea_Unit_t

Name	Description
R_SPEA_RLE0	RLE unit 0.
R_SPEA_RLE1	RLE unit 1.
R_SPEA_RLE2	RLE unit 2.
R_SPEA_RLE3	REL unit 3.
R_SPEA_SU_OFFSET	Offset for correct index calculation
R_SPEA_SU0	Sprite unit 0.
R_SPEA_SU1	Sprite unit 1.
R_SPEA_SU2	Sprite unit 2.
R_SPEA_SU3	Sprite unit 3.
R_SPEA_SU_LAST	delimiter.

See also

R_SPEA_UnitUpdateActive
R_SPEA_UpdateUnit
R_SPEA_UnitEnable

5.3.4 r_spea_UnitCapabs_t

Description

Describes Capability.

Definition

```
typedef enum {
    R_SPEA_SPEACAPBS_RLE        = 0x00u,
    R_SPEA_SPEACAPBS_SPRITE     = 0x01u,
    R_SPEA_SPEACAPBS_LAST       = 0X02u,
} r_spea_UnitCapabs_t
```

Table 5-6 Enumerator of r_spea_UnitCapabs_t

Name	Description
R_SPEA_SPEACAPBS_RLE	Select RLE unit.
R_SPEA_SPEACAPBS_SPRITE	Select Sprite unit.
R_SPEA_SPEACAPBS_LAST	delimiter.

See also

R_SPEA_UnitCapabilitiesSet

5.4 Structure

This section shows the structure used in SPEA API function.

5.4.1 r_spea_SpriteConfig_t

Description

Configuration for a Sprite data.
For the detail of range and alignment information, see ***Table 3-4***.

Definition

```
typedef struct {
   uint8_t   Enable;
   uint8_t   VSync;
   uint32_t  SrcAddr;
   uint16_t  Height;
   uint16_t  Width;
   uint16_t  PosX;
   unit16_t  PosY;
} r_spea_SpriteConfig_t
```

Table 5-7 Member of r_spea_SpriteConfig_t structure

Member	Description
Enable	Specifies whether enable or disable the blit of Sprite data. 0: disable 1: enable
Vsync	Specifies the signal to be used for register update. 0: VUPDATE0 1: VUPDATE1
SrcAddr	Specifies start address of the Sprite data. It should be 8 byte-aligned.
Height	Specifies height of the Sprite data in pixels.
Width	Specifies Sprite width in pixels. It should be multiple of 8 Bytes.
PosX	Specifies horizontal position of virtual frame buffer in pixels. It should be multiple of 8 Bytes.
PosY	Specifies vertical position of virtual frame buffer in pixels.

See also

R_SPEA_SetSprite

5.4.2 r_spea_RleConfig_t

Description

Configuration for an RLE engine.

Definition

```
typedef struct {
   uint32_t       VirtStart;
   uint32_t       MemStart;
   r_spea_Color_t ColorMode;
   uint8_t        VideoOutID;
} r_spea_RleConfig_t
```

Table 5-8 Member of r_spea_RleConfig_t structure

Member	Description
VirtStart	Specifies start address on the virtual frame buffer. It should be fixed to 0x30000000.
MemStart	Specifies start address assigned for RLE compressed data. It should be 8 byte-aligned.
ColorMode	Specifies bits per pixel of RLE data from the following. R_SPEA_CM_8BPP R_SPEA_CM_18BPP R_SPEA_CM_24BPP
VideoOutID	Specifies the enable / disable of RLE engine. Only lower 4 bits are valid. For RLE engine 0 0 : enable 3-15: disable 1-2 : setting prohibit For RLE engine 1 1 : enable 3-15: disable 0, 2 : setting prohibit

See also

r_spea_Color_t
R_SPEA_SetRle

Revision History	Renesas Graphics Library Sprite Engine A (SPEA) Driver User's Manual: Software

Rev.	Date	Description	
		Page	Summary
0.1	Nov 29, 2018	-	First edition.
0.2	Dec 20, 2018	5, 25, 26, 29, 31, 35, 37, 39, 41, 43, 46	Added the return code R_SPEA_ERR_NOT_ACCEPTABLE.
		7, 21, 35	Deleted the internal function R_SPEA_ErrorHandler.
		41	Added the description for VideoOutID.
0.3	Mar 28, 2019	7	Changed the status transition table. R_SPEA_DeInit R_SPEA_SetErrorCallback
		18	Fixed the range of PosX, PosY, Width and Height.
		25, 26, 27, 29, 35, 37, 43	Fixed the return code.
		25, 26, 27, 29, 31, 35, 37, 39, 41, 43,	Added the "const" to argument. R_SPEA_Init R_SPEA_DeInit R_SPEA_UnitUpdateActive R_SPEA_UpdateUnit R_SPEA_UnitCapabilitiesSet R_SPEA_SetSprite R_SPEA_SpriteEnable R_SPEA_SetSpritePos R_SPEA_SetRle R_SPEA_UnitEnable
		33	Changed the error callback specification.
		22	Fixed typo.
1.0	June 12, 2019	5	Improved the description of Error Handling.
		10, 13	Added the figure of VDCE and SPEA connection.
		18, 43, 53	Fixed RLE data address alignment.
		29	Added the description of error callback.
2.0	May 10, 2020	27, 28, 29, 31, 33, 35, 37, 39, 41, 43, 45	Added the description for non-reentrant.

Renesas Graphics Library
Sprite Engine A (SPEA) Driver
User’s Manual: Software
Publication Date: Rev.0.1 Nov 29, 2018
Rev.2.0 May 10, 2020

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc. Milpitas Campus
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus
6024 Silver Creek Valley Road, San Jose, CA 95138, USA
Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：https://www.renesas.com/contact/

Colophon 3.1

Renesas Graphics Library
Sprite Engine A (SPEA) Driver

Renesas Electronics Corporation