Ambarella

SDK6 AN Custom LCD Driver

Version 1.5

September 17, 2014

Ambarella

Confidentiality Notice:

The contents of this document are proprietary and confidential information of Ambarella Inc.

The material in this document is for information only. Ambarella assumes no responsibility for errors or omissions and reserves the right to change, without notice, product specifications, operating characteristics, packaging, ordering, etc. Ambarella assumes no liability for damage resulting from the use of information contained in this document. All brands, product names and company names are trademarks of their respective owners.

US
3101 Jay Street
Ste. 110
Santa Clara, CA 95054, USA
Phone: +1.408.734.8888
Fax: +1.408.734.0788

Hong Kong
Unit A&B, 18/F, Spectrum Tower
53 Hung To Road
Kwun Tong, Kowloon
Phone: +85.2.2806.8711
Fax: +85.2.2806.8722

Korea
6 Floor, Hanwon-Bldg.
Sunae-Dong, 6-1, Bundang-Gu
SeongNam-City, Kyunggi-Do
Republic of Korea 463-825
Phone: +031.717.2780
Fax: +031.717.2782

China - Shanghai
9th Floor, Park Center
1088 Fangdian Road, Pudong New District
Shanghai 201204, China
Phone: +86.21.6088.0608
Fax: +86.21.6088.0366

Taiwan
Suite C1, No. 1, Li-Hsin Road 1
Science-Based Industrial Park
Hsinchu 30078, Taiwan
Phone: +886.3.666.8828
Fax: +886.3.666.1282

Japan - Yokohama
Shin-Yokohama Business Center Bldg. 5th Floor
3-2-6 Shin-Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Phone: +81.45.548.6150
Fax: +81.45.548.6151

China - Shenzhen
Unit E, 5th Floor
No. 2 Finance Base
8 Ke Fa Road
Shenzhen, 518057, China
Phone: +86.755.3301.0366
Fax: +86.755.3301.0966

I Contents

II Preface

This document provides technical details using a set of consistent typographical conventions to help the user differentiate key concepts at a glance.

Conventions include:

Example	Description
AmbaGuiGen, **DirectUSB** **Save**, **File > Save** **Power**, **Reset**, **Home**	Software names GUI commands and command sequences Computer / Hardware buttons
Flash_IO_control **da**, **status**, **enable**	Register names and register fields. For example, **Flash_IO_control** is the register for global control of Flash I/O, and bit 17 (**da)** is used for DMA acknowledgement.
GPIO81, **CLK_AU**	Hardware external pins
VIL, VIH, VOL, VOH	Hardware pin parameters
INT_O, RXDATA_I	Hardware pin signals
amb_performance_t ***amb_operating_mode_t*** **amb_set_operating_mode()**	API details (e.g., functions, structures, and type definitions)

`yes`
`/usr/local/bin`
`make`
	User entries into software dialogues and GUI windows
	File names and paths
	Command line scripting and Code

Table II-1. Typographical Conventions for Technical Documents.

Additional Ambarella typographical conventions include:

- Acronyms are given in UPPER CASE using the default font (e.g., AHB, ARM11 and DDRIO).
- Names of Ambarella documents and publicly available standards, specifications, and databooks appear in *italic* type.

1 Overview

1.1 Overview: Introduction

This application note (AN) describes the procedure to create a customized LCD panel driver for an Ambarella development platform with an Ambarella chip and software development kit (SDK). Testing a customized LCD panel driver in the development environment provides the basis to implement a private LCD panel device and test features planned for the final product. The document is laid out as follows:

- Chapter 1 “Overview”
- Chapter 2 “Custom LCD Panel Driver API”
- Chapter 3 “Example”
- Appendix 1

1.2 Overview: Scope of Document

The APIs and LCD panel driver example in this document are intended to provide an experienced programmer with the background needed to create and implement a custom panel in a final product. It is assumed that the reader of this document has an LCD panel device for testing purposes and already is familiar with the chip, the development board, and the system software. Further the reader should be familiar with the steps and functions to implement a standard LCD panel, which are covered in the Ambarella document “*SDK6 API System*”.

1.3 Overview: Necessary Resources

For using a customized LCD Panel driver, the user requires the following documents:

- The datasheet of the chosen LCD Panel device.
- The DSP library (AmbaDSP) which provides an interface to generate video frame data as the input of the LCD Panel. Please refer to “*SDK6 DSP Support Package*” for details.
- The System library (AmbaSYS) which provides peripheral drivers that help the user to control the LCD Panel device. Please refer to “*SDK6 API System*” for details.

The customer should confirm that the LCD Panel device is compatible with the Ambarella chip and the platform libraries before attempting to use the camera platform libraries to translate the Image Sensor datasheet to C language.

2 Custom LCD Panel Driver API

2.1 API: Overview

This chapter provides the APIs to create a customized LCD panel driver:

- (Section 2.2) API: Background
- (Section 2.3) API: List of Functions

2.2 API: Background

The APIs in this chapter can help the user to do the following:

- Enable the LCD panel
- Disable the LCD panel
- Get vout configuration for the current LCD display mode
- Configure the LCD display mode
- Control backlight
- Set brightness
- Set contrast
- Set color balance

2.3 API: List of Functions

- AmbaLCD_Enable
- AmbaLCD_Disable
- AmbaLCD_GetInfo
- AmbaLCD_Config
- AmbaLCD_SetBacklight
- AmbaLCD_SetBrightness
- AmbaLCD_SetContrast
- AmbaLCD_SetColorBalance

2.3.1 AmbaLCD_Enable

API Syntax:

int **AmbaLCD_Enable** (UINT8 Chan)

Function Description:

- This function is used to enable the LCD panel.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number

*Table 2-1. Parameters for LCD Panel Driver API **AmbaLCD_Enable()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-2. Returns for LCD Panel Driver API **AmbaLCD_Enable()**.*

Example:

```
AMBA_LCD_CONFIG_sLcdConfig= {
        .Mode = 0,
        .TvSystem = AMBA_DSP_VOUT_SYSTEM_60HZ
};

/* Initializations of LCD channel 0 */
AmbaLCD_Config(0, &LcdConfig);

/* Enable LCD panel of LCD channel 0 */
AmbaLCD_Enable(0);
```

See Also:

AmbaLCD_Disable()

2.3.2 AmbaLCD_Disable

API Syntax:

int **AmbaLCD_Disable** (UINT8 Chan)

Function Description:

- This function is used to disable the LCD panel.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number

*Table 2-3. Parameters for LCD Panel Driver API **AmbaLCD_Disable()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-4. Returns for LCD Panel Driver API **AmbaLCD_Disable()**.*

Example:

```
/* Enable LCD panel of LCD channel 0*/
AmbaLCD_Enable(0);
...
/* Disable LCD */
AmbaLCD_Disable(0);
```

See Also:

AmbaLCD_Enable()

2.3.3 AmbaLCD_GetInfo

API Syntax:

int **AmbaLCD_GetInfo** (UINT8 Chan , AMBA_LCD_INFO_s *pInfo)

Function Description:

- This function is used to get the vout configuration for the current LCD display mode.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number
AMBA_LCD_INFO_s	***pInfo**	Pointer to LCD display mode information. Please refer to Section 2.3.3.1 for more information.

*Table 2-5. Parameters for LCD Panel Driver API **AmbaLCD_GetInfo()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-6. Returns for LCD Panel Driver API **AmbaLCD_GetInfo()**.*

Example:

```
AMBA_LCD_INFO_s *pInfo;

/* Get vout configuration */
AmbaLCD_GetInfo(0, pInfo);
```

See Also:

None

2.3.3.1 AmbaLCD_GetInfo > AMBA_LCD_INFO_s

Type	Field	Description
UINT16	**Width**	Horizontal display resolution of the LCD panel
UINT16	**Height**	Vertical display resolution of the LCD panel
AMBA_DSP_FRAME_RATE_s	**FrameRate**	Frame rate of the current LCD configuration
AMBA_LCD_ASPECT_RATIO_s	**AspectRatio**	Aspect ratio of the LCD panel display

*Table 2-7. Definition of **AMBA_LCD_INFO_s** for LCD Panel Driver API **AmbaLCD_GetInfo()**.*

2.3.4 AmbaLCD_Config

API Syntax:

int **AmbaLCD_Config** (UINT8 Chan , AMBA_LCD_MODE_ID_u Mode)

Function Description:

- This function is used to configure the default LCD panel display mode.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number
AMBA_LCD_ MODE_ID_u	**Mode**	Configuration of the LCD display mode. Please refer to Section 2.3.4.1 for definition.

*Table 2-8. Parameters for LCD Panel Driver API **AmbaLCD_Config()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-9. Returns for LCD Panel Driver API **AmbaLCD_Config()**.*

Example:

```
AMBA_LCD_MODE_ID_u LcdMode = { .Data = 0 };

/* LCD Initializations  */
AmbaLCD_Config(0,  LcdMode);
```

See Also:

None

2.3.4.1 AmbaLCD_Config > AMBA_LCD_MODE_ID_u

Type	Field	Description
UINT8	**Mode:5**	LCD input mode
UINT8	**FlipHorizontal:1**	Flip the display horizontally.
UINT8	**FlipVertical:1**	Flip the display vertically.
UINT8	**Stereoscopic:1**	3D display

*Table 2-10. Definition of **AMBA_LCD_MODE_ID_u** for LCD Panel Driver API **AmbaLCD_Config()**.*

2.3.5 AmbaLCD_SetBacklight

API Syntax:

int **AmbaLCD_SetBacklight** (UINT8 Chan , INT32 EnableFlag)

Function Description:

- This function is used to turn on/off the LCD display backlight.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number
INT32	**EnableFlag**	0: Backlight Off 1: Backlight On

Table 2-11. Parameters for LCD Panel Driver API ***AmbaLCD_SetBacklight()****.*

Returns:

Return	Description
0	Success
- 1	Failure

Table 2-12. Returns for LCD Panel Driver API ***AmbaLCD_SetBacklight()****.*

Example:

```
/* Enable LCD panel of LCD channel 0 */
AmbaLCD_Enable(0);
...
/* Turn on the back light */
AmbaLCD_SetBacklight(0, 1);
```

See Also:

None

2.3.6 AmbaLCD_SetBrightness

API Syntax:

int **AmbaLCD_SetBrightness** (UINT8 Chan, INT32 Offset)

Function Description:

- This function is used to adjust the brightness of the LCD display.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number
INT32	**Offset**	Offset of the black level

*Table 2-13. Parameters for LCD Panel Driver API **AmbaLCD_SetBrightness()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-14. Returns for LCD Panel Driver API **AmbaLCD_SetBrightness()**.*

Example:

```
/* Set brightness level to normal (i.e. 64) */
AmbaLCD_SetBrightness(0, 64);
```

See Also:

None

2.3.7 AmbaLCD_SetContrast

API Syntax:

int **AmbaLCD_SetContrast** (UINT8 Chan, float Gain)

Function Description:

- This function is used to adjust the contrast of the LCD display.

Parameters:

Type	Parameter	Description
UINT8	**Chan**	LCD channel number
float	**Gain**	Contrast gain value in dB (decibel)

Table 2-15. Parameters for LCD Panel Driver API ***AmbaLCD_SetContrast ()****.*

Returns:

Return	Description
0	Success
- 1	Failure

Table 2-16. Returns for LCD Panel Driver API ***AmbaLCD_SetContrast ()****.*

Example:

```
/* Set 1x contrast gain */
AmbaLCD_SetContrast(0, 1.0);
```

See Also:

None

2.3.8 AmbaLCD_SetColorBalance

API Syntax:

int **AmbaLCD_SetColorBalance** (UINT8 Chan , AMBA_LCD_COLOR_BALANCE_s *pColorBalance)

Function Description:

- This function is used to adjust the color balance of the LCD display.

Parameters:

Type	Parameter	Description
UINT8	Chan	LCD channel number
AMBA_LCD_ COLOR_ BALANCE_s	***pColorBalance**	Gains and Offsets of each color channel. Please refer to Section 2.3.8.1 for more information.

*Table 2-17. Parameters for LCD Panel Driver API **AmbaLCD_SetColorBalance()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 2-18. Returns for LCD Panel Driver API **AmbaLCD_SetColorBalance()**.*

Example:

```
AMBA_LCD_COLOR_BALANCE_s ColorBalance = {
.OffsetRed = 64,
.OffsetGreen = 64,
.OffsetBlue = 64,
.GainRed = 1.0,
.GainGreen = 1.0,
.GainBlue = 1.0
};

/* Set color balance */
AmbaLCD_SetColorBalance(0, &ColorBalance);
```

See Also:

None

2.3.8.1 AmbaLCD_SetColorBalance > AMBA_LCD_COLOR_BALANCE_s

Type	Field	Description
INT16	**OffsetRed**	Offset of color red
INT16	**OffsetGreen**	Offset of color green
INT16	**OffsetBlue**	Offset of color blue
float	**GainRed**	Gain of color red
float	**GainGreen**	Gain of color green
float	**GainBlue**	Gain of color blue

Table 2-19. Definition of ***AMBA_LCD_COLOR_BALANCE_s*** *for LCD Panel Driver API* ***AmbaLCD_SetColorBalance()****.*

3 Example

3.1 Example: Overview

The following example presents a process to apply a custom LCD panel driver. For more details, please refer to "*SDK6 API DSP Support Package"* and "*SDK6 API System"*.

3.2 Example: Custom LCD Display

In this example, the user assumes that the normal brightness/black level is 0x40(64) and that no rotation or contrast enhancement is performed.

Before using the custom LCD panel driver, the user needs to set up the video mode info of the LCD panel according to the pixel format, timing, and special property of the LCD panel.

An example of the information from the WDF9648W panel is given below:

```
/*--------------------------------------------------------------------------------*\
 * WDF9648W video mode info
\*--------------------------------------------------------------------------------*/
static AMBA_LCD_WDF9648W_CONFIG_s  WdF9648w_Config[2]={
  [AMBA_LCD_WDF9648W_960_480_60HZ] = {
     .Width         = 960,
     .Height        = 480,
     .FrameRate     = {
       .Interlace      = 0,
       .TimeScale      = 60000,
       .NumUnitsInTick = 1001,
     },
     .ScreenMode    = AMBA_LCD_WDF9648W_SCREEN_MODE_WIDE,
     /* clock frequency to LCD */
     .DeviceClock   = AMBA_DSP_VOUT_PIXEL_CLOCK_FULL_DCLK,
     /* pixel format */
     .OutputMode    = AMBA_DSP_VOUT_LCD_1COLOR_PER_DOT,
     /*  even line color order */
     .EvenLineColor = AMBA_DSP_VOUT_LCD_COLOR_RGB,
     /* odd   line color order */
     .OddLineColor  = AMBA_DSP_VOUT_LCD_COLOR_RGB,
     .VideoTiming   = {
        .PixelClock       = 36000000,
        .PixelRepetition  = 1,
        .Htotal           = 1144,
        .Vtotal           = 525,
        .HsyncColStart    = 0,
         .HsyncColEnd     = 1,
         .VsyncColStart   = 0,
         .VsyncColEnd     = 0,
         .VsyncRowStart   = 0,
         .VsyncRowEnd     = 1,
         .ActiveColStart  = 32,
         .ActiveColWidth  = 960,
         .ActiveRowStart  = 42,
         .ActiveRowHeight = 480
      }
   },

   [AMBA_LCD_WDF9648W_960_480_50HZ] = {
      .Width         = 960,
      .Height        = 480,
      .FrameRate  =  {
        .Interlace      = 0,
        .TimeScale      = 50,
        .NumUnitsInTick = 1,
      },
      .ScreenMode    = AMBA_LCD_WDF9648W_SCREEN_MODE_WIDE,
      /* clock frequency to LCD */
      .DeviceClock   = AMBA_DSP_VOUT_PIXEL_CLOCK_FULL_DCLK,
      /* pixel format */
      .OutputMode    = AMBA_DSP_VOUT_LCD_1COLOR_PER_DOT,
      /*  even line color order */
      .EvenLineColor = AMBA_DSP_VOUT_LCD_COLOR_RGB,
      /* odd   line color order */
      .OddLineColor  = AMBA_DSP_VOUT_LCD_COLOR_RGB,
      .VideoTiming   = {
         .PixelClock      = 30030000,
         .PixelRepetition = 1,
         .Htotal          = 1144,
         .Vtotal          = 525,
         .HsyncColStart   = 0,
         .HsyncColEnd     = 1,
         .VsyncColStart   = 0,
         .VsyncColEnd     = 0,
         .VsyncRowStart   = 0,
         .VsyncRowEnd     = 1,
         .ActiveColStart  = 32,
         .ActiveColWidth  = 960,
         .ActiveRowStart  = 42,
         .ActiveRowHeight = 480
     }
   }
};
```

Once the user has access to the LCD video mode parameters, the user can use the parameters to make some initializations. Initializations are usually done once after the system is booted. In this case, the user uses mode 0 and 60HZ TV system of the LCD panel WDF9648W, and then uses it to configure LCD channel 0 and set the brightness and contrast.

```
/* Using LCD mode 0 (AMBA_LCD_WDF9648W_960_480_60HZ) and no rotation */
AMBA_LCD_MODE_ID_u LcdMode = { .Data = 0 };

AmbaLCD_Config(0,  LcdMode);        /* Initialize LCD panel */

AmbaLCD_SetBrightness(0, 0x40);     /* normal brightness */
AmbaLCD_SetContrast(0, 0.0);        /* 0.0 dB */
```

After all the programming is done, the user can power-on the LCD device and then turn the LCD backlight on.

```
AmbaLCD_Enable(0);
AmbaLCD_SetBacklight(0, 1);
```

Now, the user can see the video/image/background color displayed on the LCD panel.

Appendix 1 Digital Vout Data Format and Pin Mapping

Supported Digital Output Modes
Digital Output Modes 0 - LCD – Single color per pixel 1 - LCD – 3 colors per pixel, no dummy clock 2 - LCD – 3 colors per pixel, dummy clock 3 - LCD – 5:6:5 mode 4 - 656 output mode 5 - 601 output mode (16-bit, 4:2:2 YUV) 6 - 601 output mode (24-bit, 4:4:4 YUV or RGB – Display 0 only) 7 - 601 output mode (8-bit, CbYCrY order) 8 - Bayer pattern output mode Other values reserved

Table A1-1. Vout Digital Output Mode.

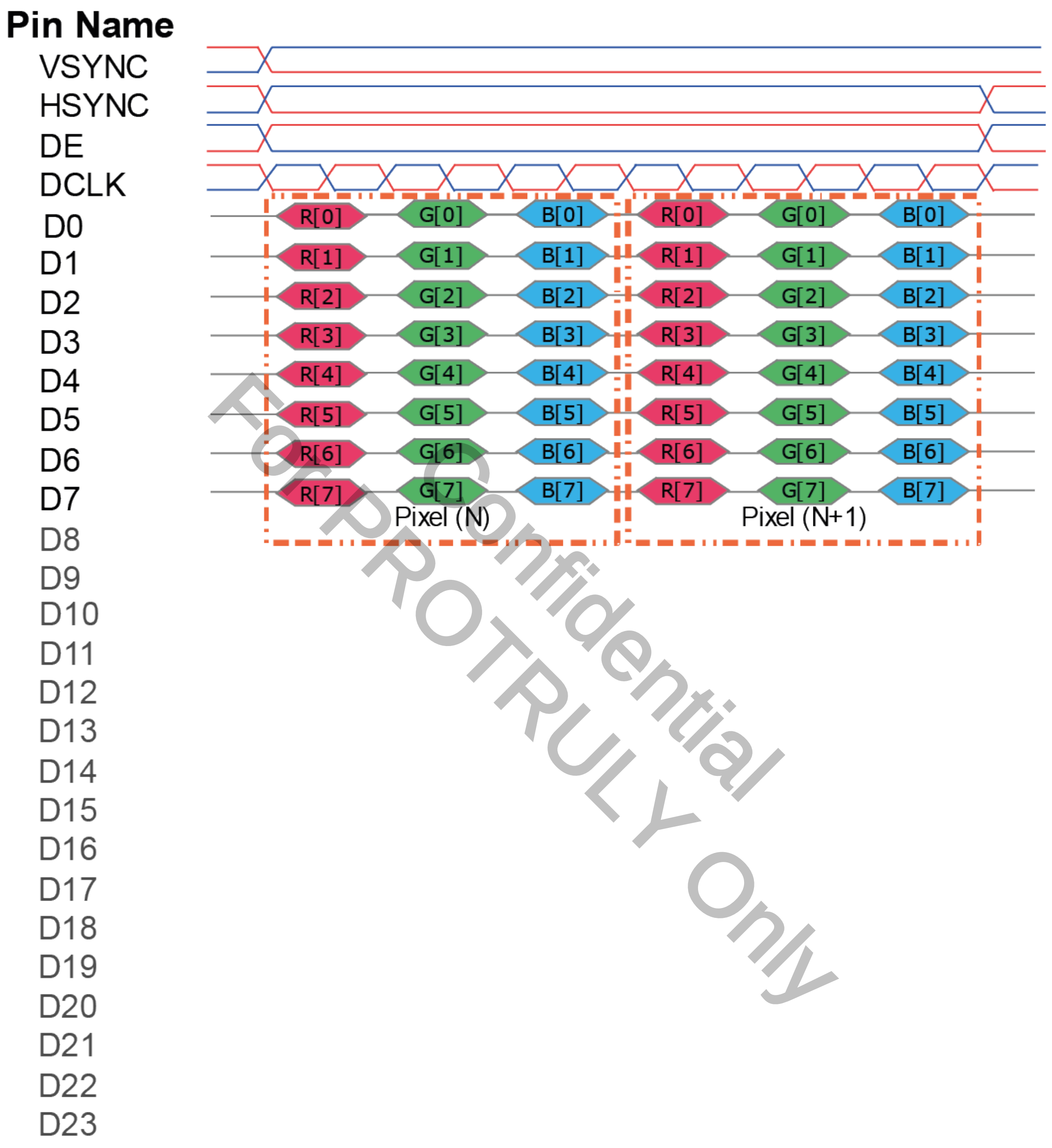
Pin Name
VSYNC
HSYNC
DE
DCLK
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
R[0] G[0] B[0] R[0] G[0] B[0]
R[1] G[1] B[1] R[1] G[1] B[1]
R[2] G[2] B[2] R[2] G[2] B[2]
R[3] G[3] B[3] R[3] G[3] B[3]
R[4] G[4] B[4] R[4] G[4] B[4]
R[5] G[5] B[5] R[5] G[5] B[5]
R[6] G[6] B[6] R[6] G[6] B[6]
R[7] G[7] B[7] R[7] G[7] B[7]
Pixel (N)
Pixel (N+1)

Figure A1-1. Digital Mode 0/1 (RGB data come from different/same pixel).

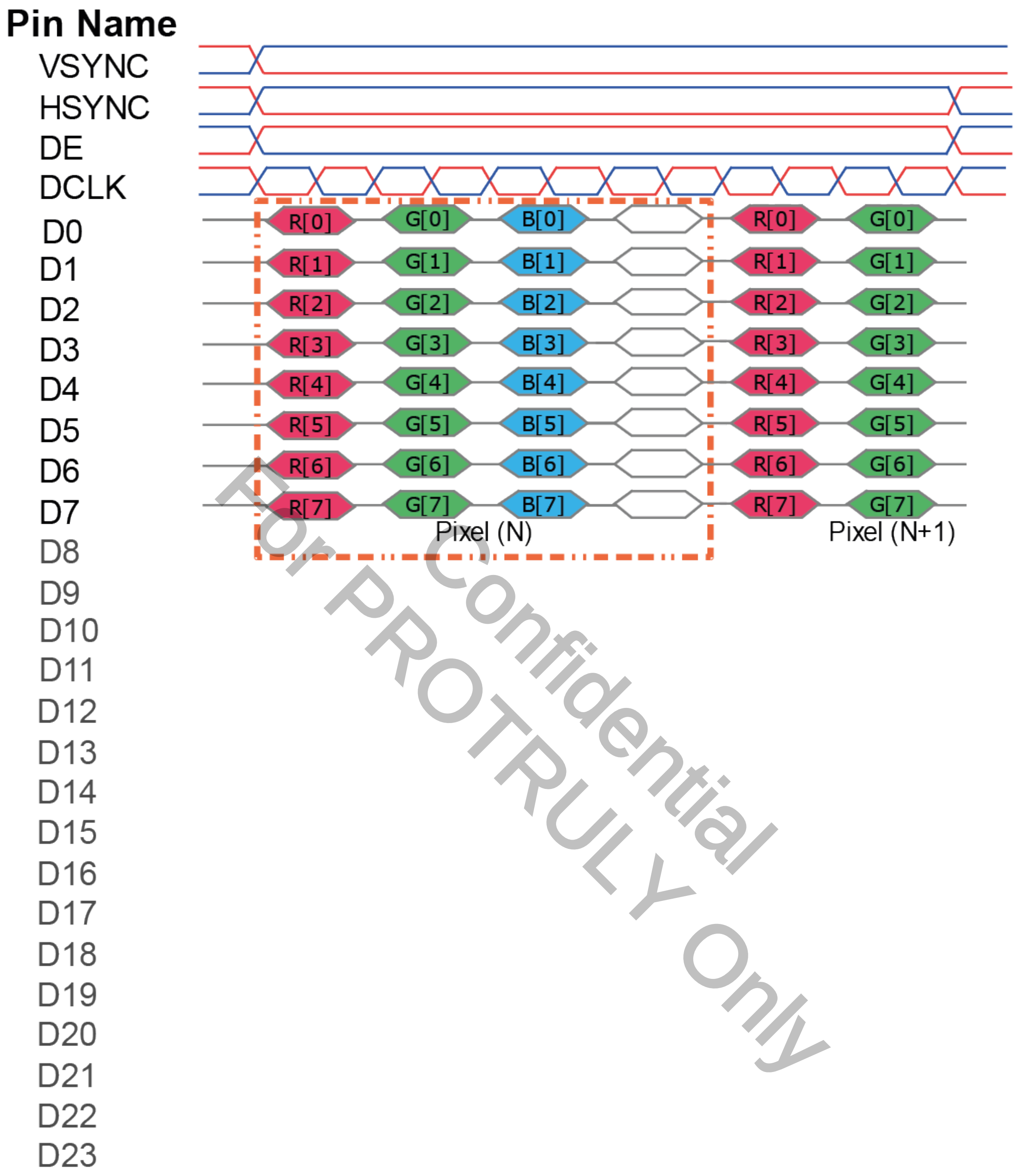

Figure A1-2. Digital Mode 2 (RGB data with one dummy clock).

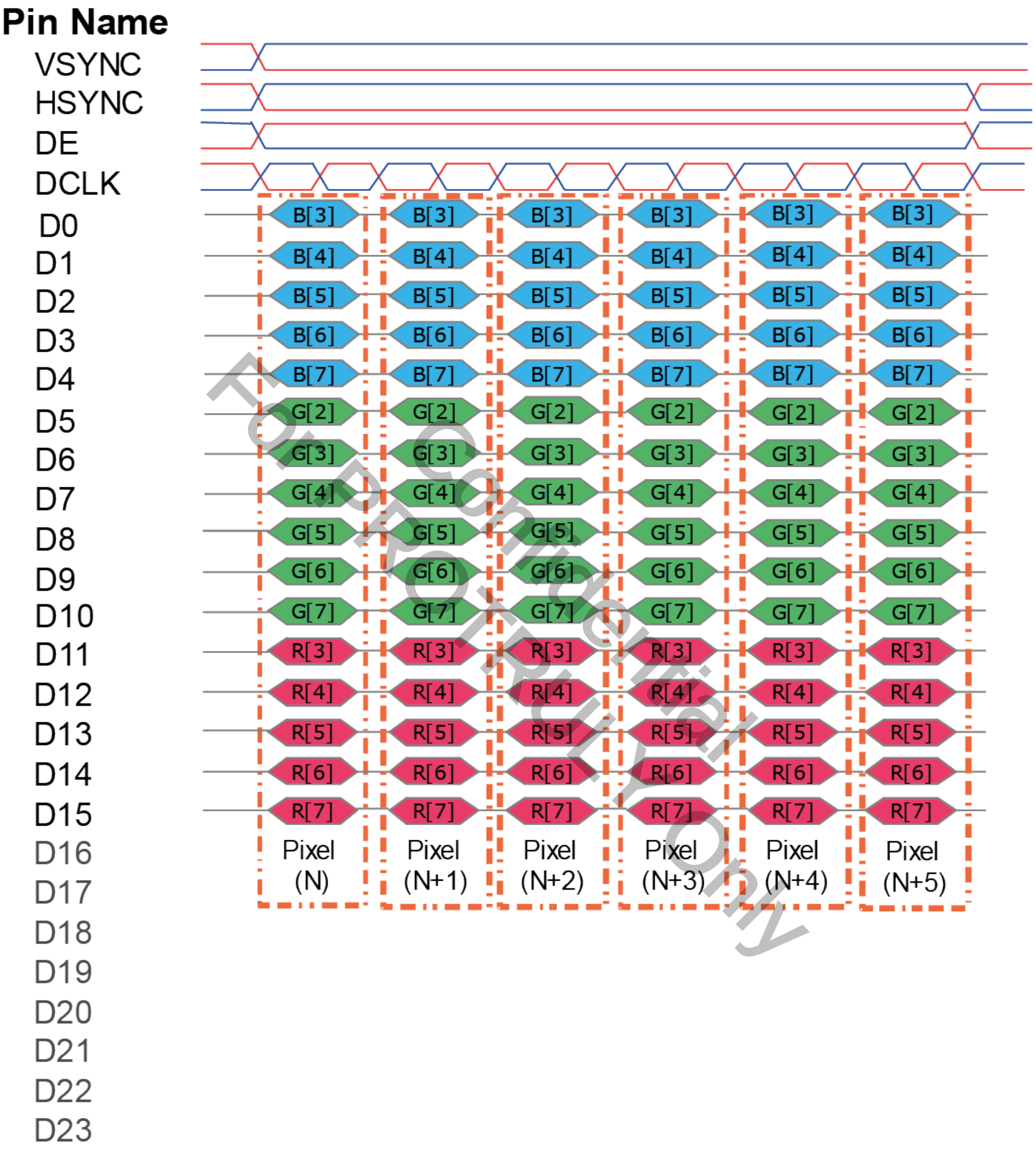

Figure A1-3. Digitnal Mode 3 (RGB565).

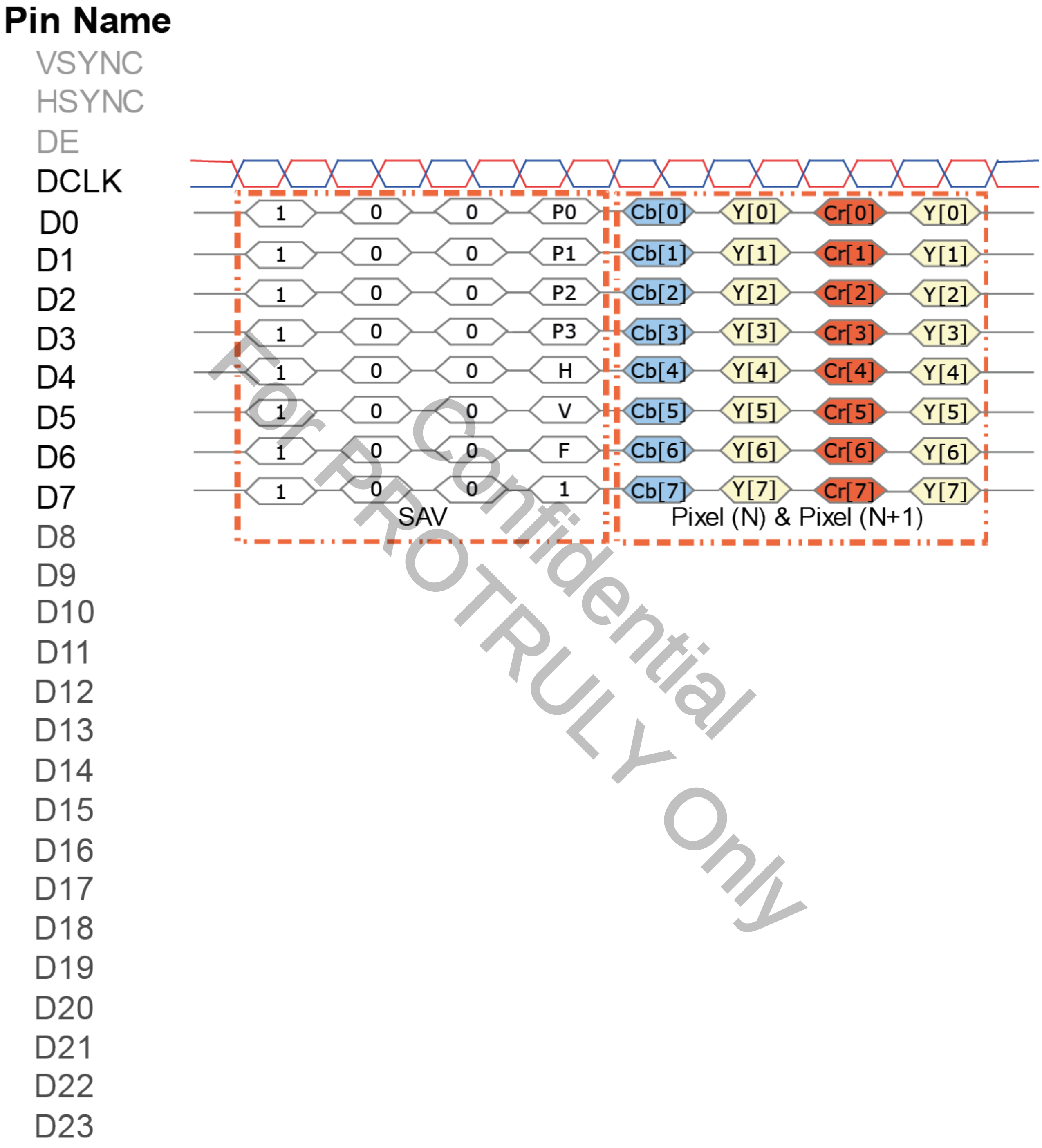

Figure A1-4. Digital Mode 4 (BT656 Output mode).

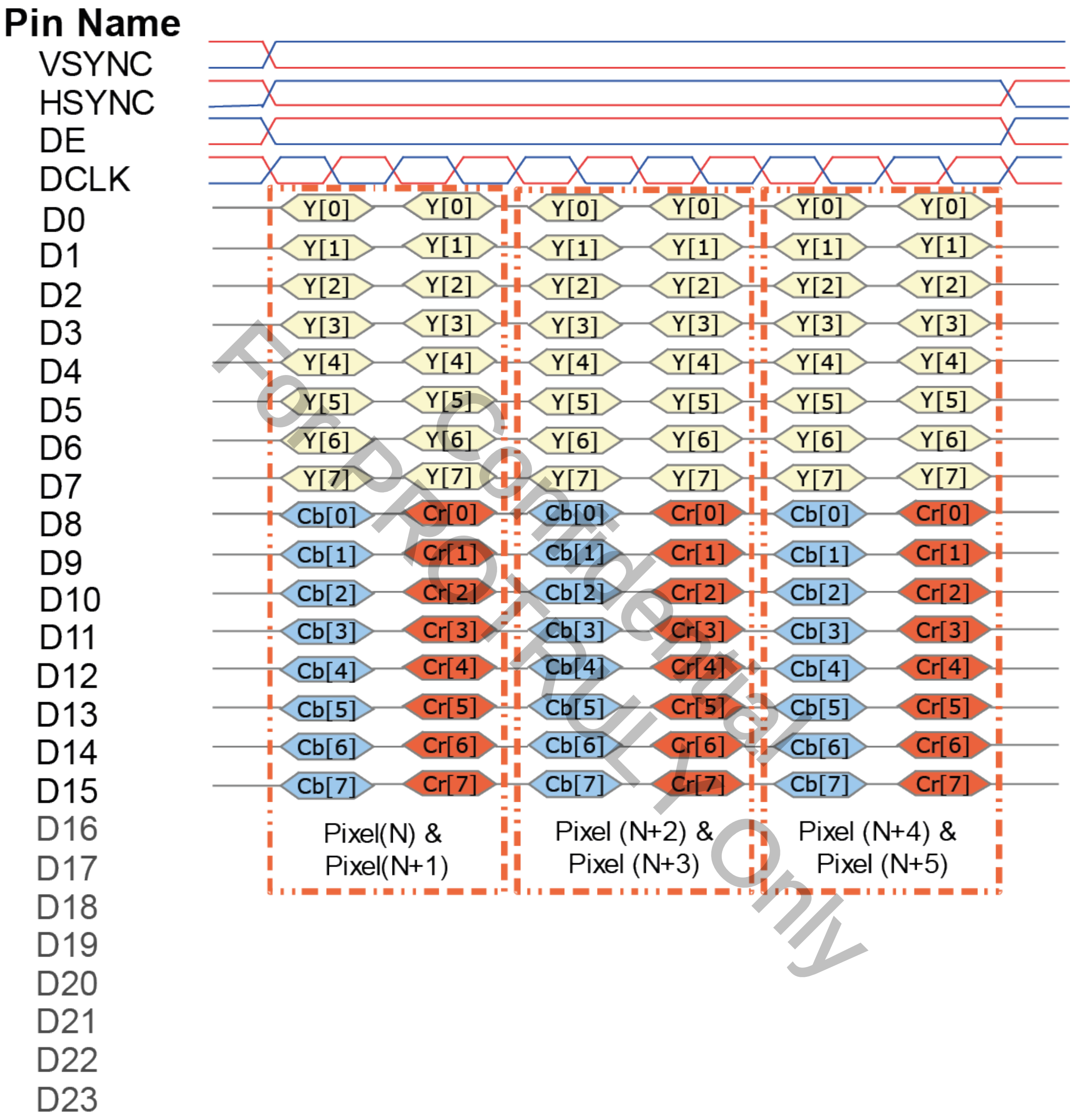

Figure A1-5. Digital Mode 5 (16Bits YUV422).

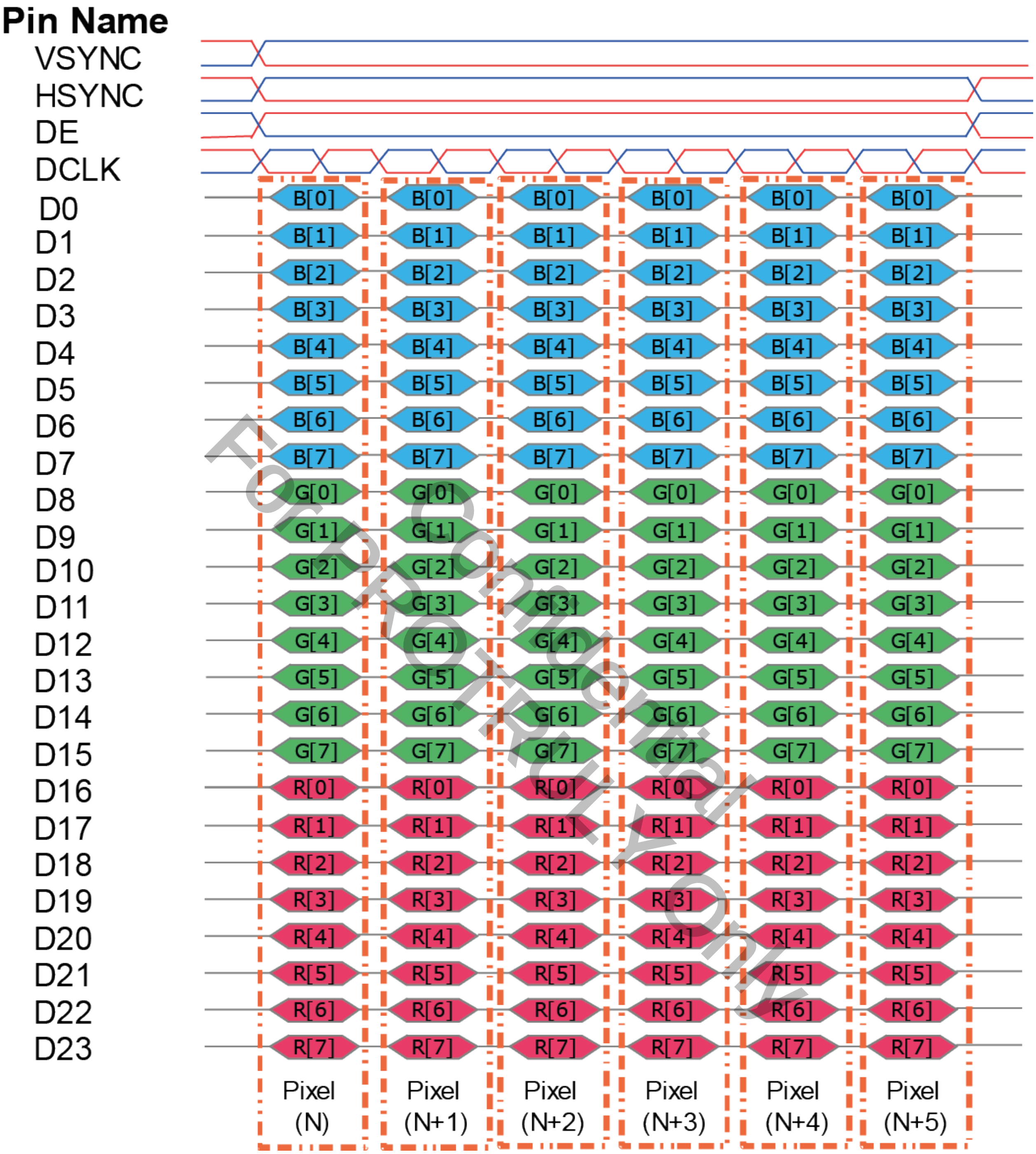

Figure A1-6. Digital Mode 6 (24Bits RGB).

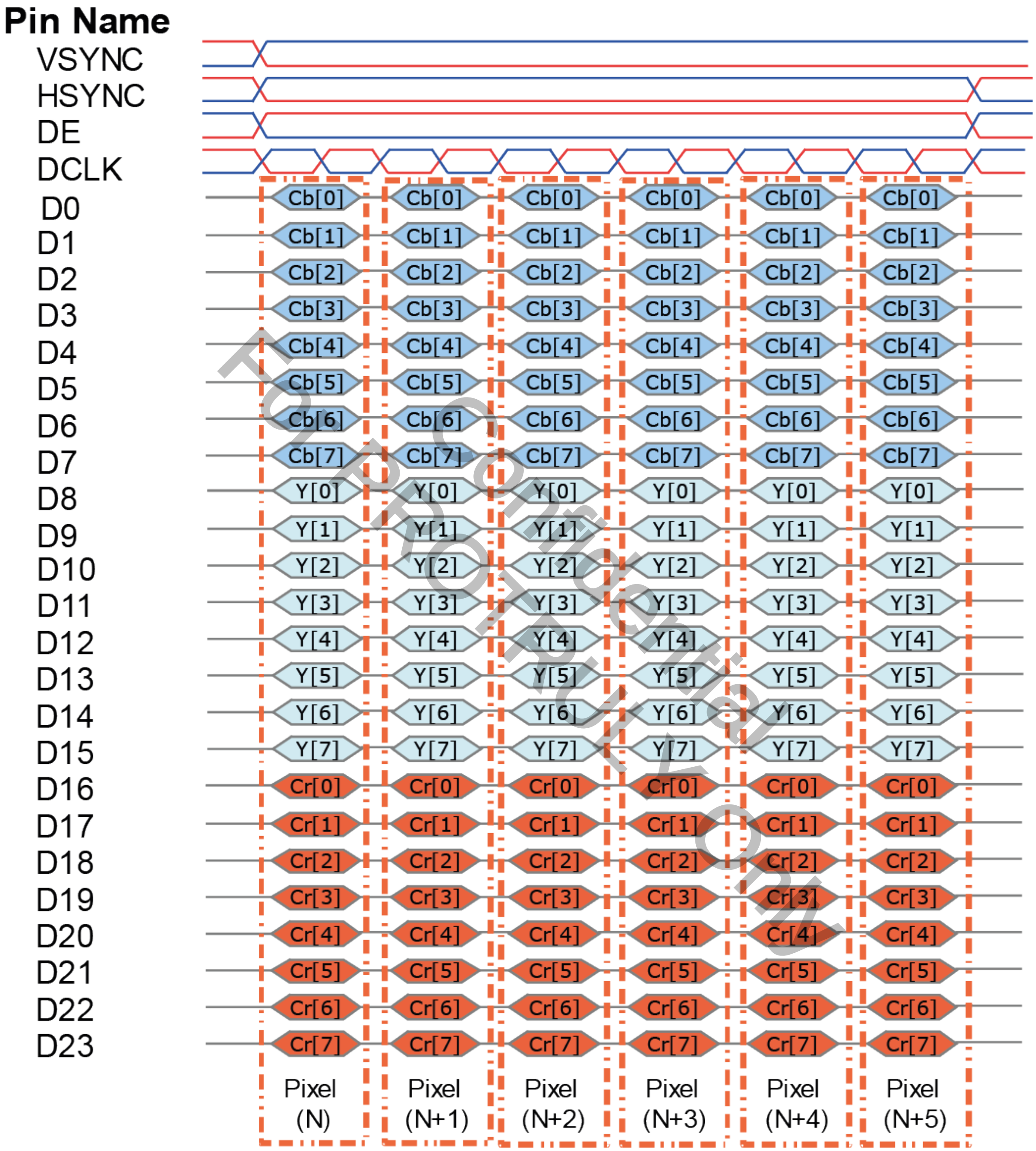

Figure A1-7. Digital Mode 6 (24Bits YUV).

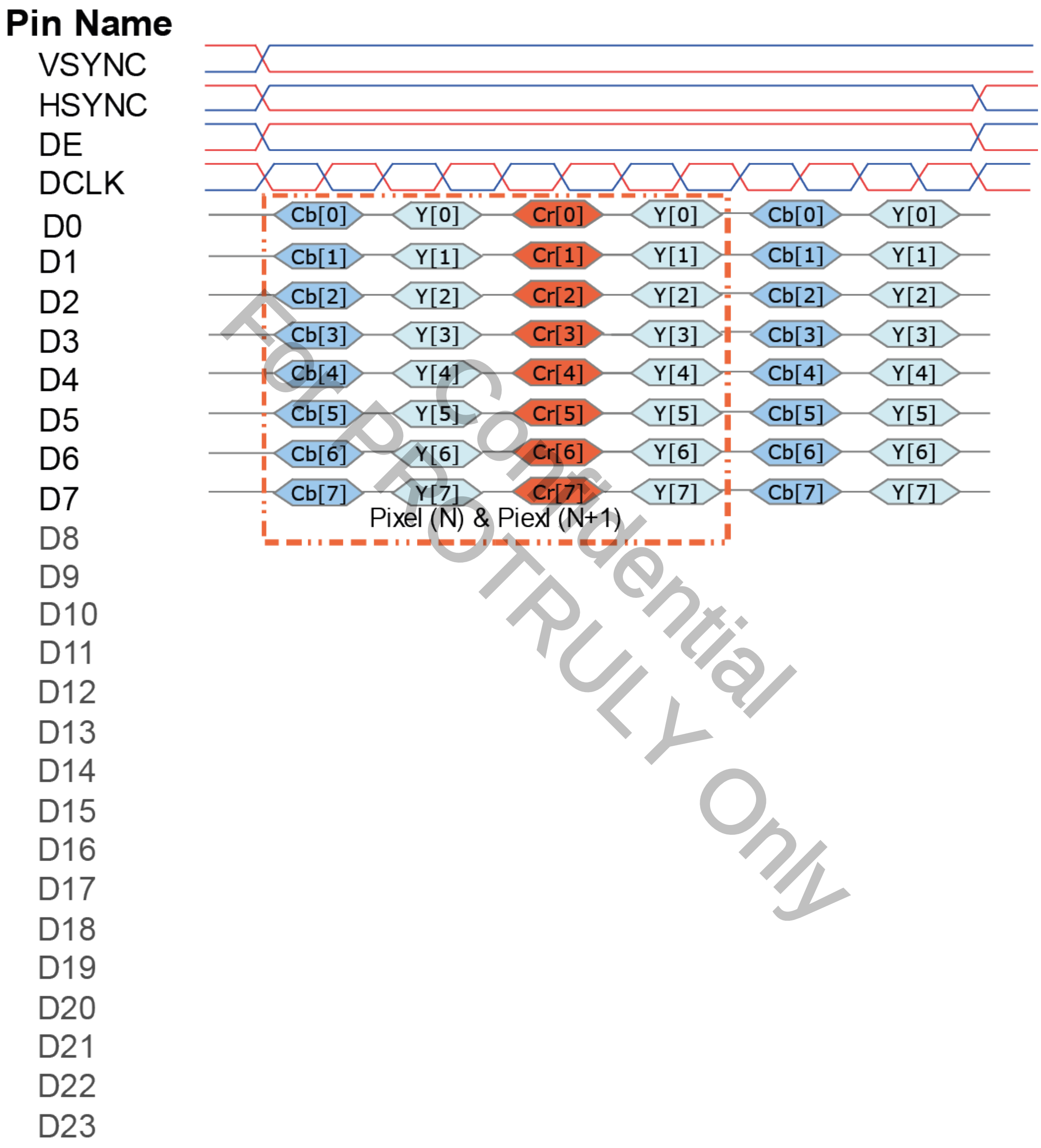
Pin Name
VSYNC
HSYNC
DE
DCLK
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
Cb[0] Y[0] Cr[0] Y[0] Cb[0] Y[0]
Cb[1] Y[1] Cr[1] Y[1] Cb[1] Y[1]
Cb[2] Y[2] Cr[2] Y[2] Cb[2] Y[2]
Cb[3] Y[3] Cr[3] Y[3] Cb[3] Y[3]
Cb[4] Y[4] Cr[4] Y[4] Cb[4] Y[4]
Cb[5] Y[5] Cr[5] Y[5] Cb[5] Y[5]
Cb[6] Y[6] Cr[6] Y[6] Cb[6] Y[6]
Cb[7] Y[7] Cr[7] Y[7] Cb[7] Y[7]
Pixel (N) & Piexl (N+1)

Figure A1-8. Digital Mode 7 (8Bits CbYCrY).

Appendix 2 Additional Resources

Related resources include:

- *SDK6 API DSP Support Package*
- *SDK6 AmbaKAL*
- *SDK6 API AmbaFS*
- *SDK6 API System*

Please contact an Ambarella representative for a full list of related resources.

Appendix 3 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, “materials”) are provided on an “*as is*” basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

All products are sold subject to Ambarella’s terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

Ambarella assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Ambarella components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer’s risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 4 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
	29 March 2013	Formatting
	1 April 2013	Add Chapter 3 (Example)
0.1	12 April 2013	Refine Chapter 3 (Example)
1.0	15 April 2013	Formatting for release
1.1	15 November 2013	Update in Chapter 2, 3, and add Digital Vout Data Format and Pin Mapping in Appendix 1
1.2	21 November 2013	udpate Chapter 3 Example
1.3	17 December 2013	Update in Chapter 2, Chapter 3, and Appendix 1
1.4	28 January 2014	Minor formatting
1.5	17 September 2014	Formatted to SDK6

Table A4-1. Revision History.