



## Intellectual Property

### OSD Color Palette

Version 1.0

July 2014

Corporate HQ & Design Center  
380 Stevens Ave. Suite 206  
Solana Beach, CA 92075  
<http://www.macnica-na.com>

## License and Terms of Use

This IP Core with its associated source code and support files, are being provided on an "as-is" basis and as an accommodation. Therefore all warranties, representations or guarantees of any kind (whether express, implied or statutory) including, without limitation, warranties of merchantability, non-infringement, or fitness for a particular purpose, are specifically disclaimed.

This source code may only be used in an Altera programmable logic device and may not be distributed without permission from Macnica Americas, Inc. It is provided free of royalties or fees of any kind.

## Document Revision History

Revision	Date	Comments
0.01	July, 2014	Initial Draft

## Contents

License and Terms of Use .....	2
Document Revision History.....	2
List of Figures .....	4
1 Introduction .....	5
2 References .....	5
2.1 Industry Standards and Specifications.....	5
2.2 Related Documents.....	5
3 Functional Specification.....	6
3.1 Symbol.....	6
3.2 Interface Description .....	7
3.2.1 Control .....	7
3.2.2 Video .....	7
3.3 Functional Description .....	8
3.4 Programming .....	9
4 Control and Status Registers.....	9
4.1 Registers.....	9
5 Specifications .....	10
5.1 Performance .....	10
5.2 Resource Utilization .....	10
6 Software Samples .....	10

## List of Figures

No table of figures entries found.

## 1 Introduction

The OSD Color Palette converts Avalon Streaming Video from 8bpp to 24bpp RGB for displays. It is typically used with the Altera Video Image Processing library in which it receives an On Screen Display video stream from a VIP Frame Reader module and outputs RGB color data to a VIP Alpha Blender for blending with other streams, or directly to a Clocked Video Output module to drive an external monitor.

## 2 References

### 2.1 Industry Standards and Specifications

### 2.2 Related Documents

Video and Image Processing User Guide, Altera Corp.

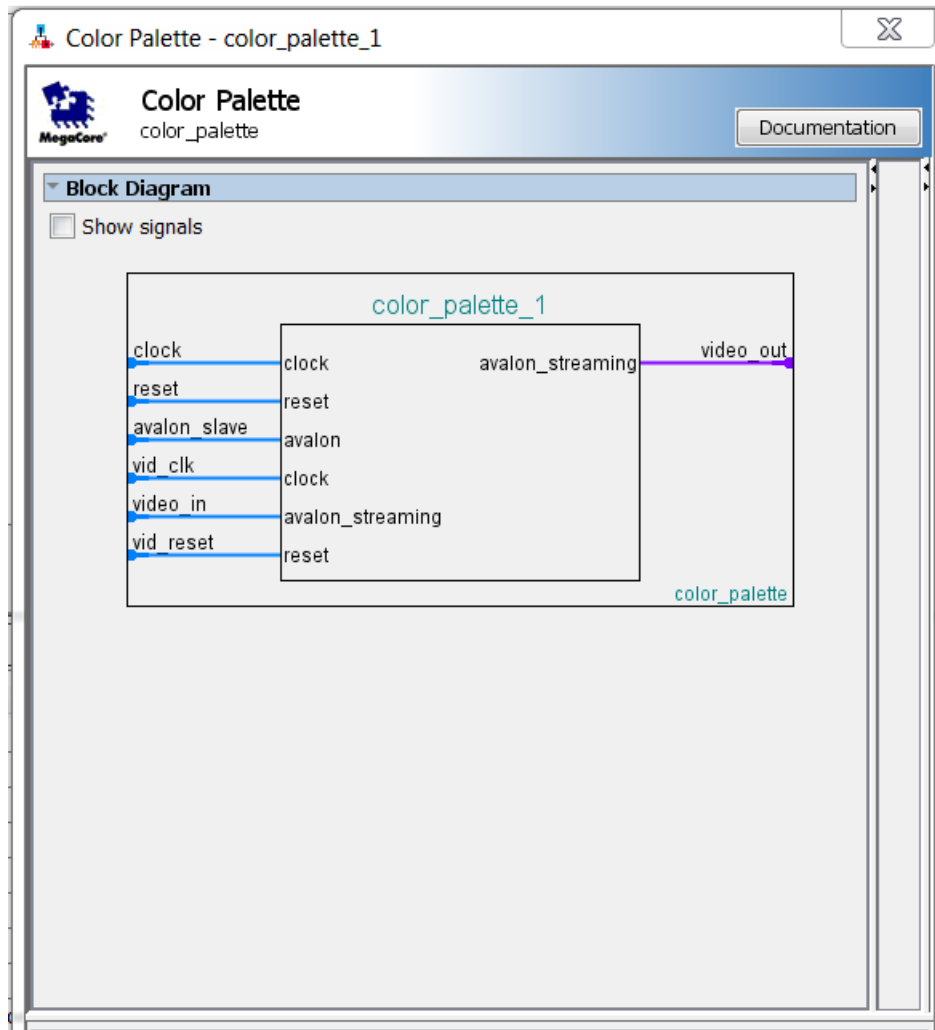
[www.altera.com/literature/ug/ug\\_vip.pdf](http://www.altera.com/literature/ug/ug_vip.pdf)

Avalon Interface Specifications, Altera Corp.

[www.altera.com/literature/manual/mnl\\_avalon\\_spec.pdf](http://www.altera.com/literature/manual/mnl_avalon_spec.pdf)

### 3 Functional Specification

#### 3.1 Symbol



## 3.2 Interface Description

### 3.2.1 Control

Interface	Description	Notes
Avalon Slave	A standard Avalon Slave interface for controlling the various settings and reading the color palette memory The Avalon Slave interface is on a separate clock domain (clock, reset)	

### 3.2.2 Video

Interface	Description	Notes
Video In (Avalon Streaming sink)	A standard Avalon-ST Video interface, 8 bits per color, one pixel per clock. The video input port uses the video clock domain (vid_clk, vid_reset)	8bpp, parallel
Video Out (Avalon Streaming source)	A standard Avalon-ST Video interface with 8bpp and three sub-pixels in parallel. The video output port uses the video clock domain (vid_clk, vid_reset)	24bpp

### 3.3 Functional Description

Intro...

The advantage is that the NIOS, or the hardened ARM Cortex A9 in the Altera SOC devices, can render the desired on-screen display much more quickly since the required display memory is only one-third of a similar RGB frame buffer. Additionally, it simplifies rendering due to the simple display memory organization.

The Altera VIP Frame Reader expects that only complete pixels are stored in a memory location. For example, with a 64-bit wide external DRAM memory can store only two complete RGB pixels in a memory location – requiring six bytes. That leaves two bytes that are unused in memory and complicates the rendering process. Alternatively, the Frame Reader can be configured to read 32-bit pixels from display memory in RGBA format – where each pixel includes an Alpha value for that pixel. The memory rendered for each frame is now four times the size of the memory required for an 8-bit display memory, thus further reducing the render performance and required processor bandwidth.

The Altera “Chroma Key” function in the VIP library allows for several Alpha settings. A default setting is applied to all pixels that do not match any specified color pattern - which then can be the foreground color with 0% transparency. Other color patterns can be specified, such as black, that can be the background color with 100% transparency. Still other color patterns can be specified for custom alpha blend values.

The OSD Color Palette, combined with many of the standard Altera VIP library elements, can provide for OSD implementations that require less memory, less processor bandwidth, and higher refresh rates for OSD applications.



### 3.4 Programming

## 4 Control and Status Registers

### 4.1 Registers

Register	Description	R/W	Notes
Control (address 0)	The control register. Bit [0] is the “Go” bit. Writing a ‘1’ to this bit enables the OSD Color Palette	R/W	
Index (address 2)	Bits [7:0] are the palette index – which is the address to the color palette. It must be written manually for each entry into the table.	R/W	
Palette Data (address 3)	<p>The palette ram data. Writing to this address will store a 32-bit value in the palette ram at the address corresponding to the palette index. Reading this address will return the 32-bit value.</p> <p>bits [23:0] – the RGB value bits [31:24] – reserved for future use (e.g. alpha)</p> <p>The palette ram is implemented as a dual-port memory, so may be accessed during operation. Port ‘A’ is dedicated to the host, and port ‘B’ is used for the color look-up, to convert incoming 8bpp values to 24bpp.</p>	R/W	

## 5 Specifications

### 5.1 Performance

The OSD Color Palette easily meets fMAX required for HD video designs of ~150 MHz.

### 5.2 Resource Utilization

Target Device Family	Memory	ALM
Cyclone V	2 M10K	101

## 6 Software Samples

Sample palette.h snippets (source included with the deliverables)

Defining the colors:

```
#define black          0          //000000      (0,0,0)
#define almost_black  1          //010101      (1,1,1)
#define alice_blue     2          //f0f8ff      (240,248,255)
#define antique_white  3          //faebd7      (250,235,215)
#define aqua           4          //00ffff      (0,255,255)
#define aqua_marine    5          //7fffd4      (127,255,212)
#define azure          6          //f0ffff      (240,255,255)
#define beige          7          //f5f5dc      (245,245,220)
#define bisque         8          //ffe4c4      (255,228,196)
#define near_black     9          //101010      (0,0,0)
#define blanched_almond 10         //ffebcd      (255,235,205)
#define blue           11         //0000ff      (0,0,255)
#define blue_violet    12         //8a2be2      (138,43,226)
#define brown          13         //a52a2a      (165,42,42)
#define burly_wood     14         //deb887      (222,184,135)
#define cadet_blue     15         //5f9ea0      (95,158,160)
#define chartreuse     16         //7fff00      (127,255,0)
#define chocolate      17         //d2691e      (210,105,30)
#define coral          18         //ff7f50      (255,127,80)
#define corn_flower_blue 19        //6495ed      (100,149,237)
#define corn_silk      20         //fff8dc      (255,248,220)
#define crimson        21         //dc143c      (220,20,60)
#define cyan           22         //00ffff      (0,255,255)
#define dark_blue      23         //00008b      (0,0,139)
#define dark_cyan      24         //008b8b      (0,139,139)
#define dark_golden_rod 25         //b8860b      (184,134,11)
#define dark_green     26         //006400      (0,100,0)
#define dark_grey      27         //a9a9a9      (169,169,169)
#define dark_khaki     28         //bdb76b      (189,183,107)
#define dark_magenta   29         //8b008b      (139,0,139)
```

Array for loading the palette values:

```
unsigned int color_values[] = {
0x000000    ,//    black
```

```
0x010101 ,// almost_black
0xf0f8ff ,// alice_blue
0xfaebd7 ,// antique_white
0x00ffff ,// aqua
0x7fffd4 ,// aqua_marine
0xf0ffff ,// azure
0xf5f5dc ,// beige
0xffe4c4 ,// bisque
0x000000 ,// black
0xffebcd ,// blanched_almond
0x0000ff ,// blue
0x8a2be2 ,// blue_violet
0xa52a2a ,// brown
0xdeb887 ,// burly_wood
0x5f9ea0 ,// cadet_blue
0x7ffff0 ,// chartreuse
0xd2691e ,// chocolate
0xff7f50 ,// coral
0x6495ed ,// corn_flower_blue
0xffff8dc ,// corn_silk
0xdc143c ,// crimson
0x00ffff ,// cyan
0x00008b ,// dark_blue
0x008b8b ,// dark_cyan
0xb8860b ,// dark_golden_rod
0x006400 ,// dark_green
0xa9a9a9 ,// dark_grey
0xbdb76b ,// dark_khaki
0x8b008b // dark_magenta
}
;
```