

Color Space Conversion

with RGB to YUV and YUV to RGB

Copyright Notice

PEC Technologies Copyrights this specification. No part of this specification may be reproduced in any form or means, without the prior written consent of PEC Technologies.

Disclaimer

This specification is preliminary and is subject to change at any time without Notice. PEC Technologies assumes no responsibility for any errors contained herein.

Questions and/or comments may be directed to

PEC Technologies

Engineering R&D Team

Email: info@pectechnology.com

Revision History

| Rev. | Date | Author | Description |
|------|----------------|-------------------------------|-------------------------------------------------------------|
| 0.1 | Jun. 01 / 2008 | Antonio Park, PEC R&D Team | First Draft Release |
| 0.2 | Jun. 20 / 2009 | Antonio Park, PEC R&D Team | RGB, YUV Conversion Verification Several Formula applied |
| | | | |

1

Introduction

Many image processing applications require separation of luminance (brightness) and chrominance (color) components of image data. For instance, such basic image manipulations as brightness, contrast, saturation, hue, and gamma might be easily performed in YUV(Y/Cr/Cb) domain, but it is more difficult in RGB color space.

Complex video processing algorithms is also involving various manipulations and analysis of luminance and chrominance components separately. The reasons are for example:

- 1. In Video compression**, the luminance and chrominance components separation is necessary since their different MTF (Modulation Transfer Function). As a result, chroma components require less data to be kept after compression than luma components; i.e. chroma components might be compressed strongly with no salient distortions addition.
- 2. In Video improvement systems**, it's needed since most sharpness and noise reduction algorithms, that is based on temporal (frame-to-frame) and spatial (into frame) adaptive filtration technique, are strongly different in respect of luma and chroma image components. Applying such video improvement algorithms to RGB video data brings about low effectiveness and visible color hue distortion.

In many cases, some applications are getting RGB video components to their video processors mainly from digitized analog RGB, DVI or HDMI interfaces, CCD or CMOS cameras. Thus, the Image Processors necessarily require a format conversion from **RGB to YUV color space conversion stage**.

In real-time HDTV video processing modules for PDP/LCD display, the controllers with ASIC, FPGA are starting Video improvement, MPEG2 compression, pre-processing system with this Color Space Conversion Cores..

This Color Space Conversion Core is used for a start of scaling processing from RGB of Image Sensor Processor(CMOS).

All Video/Graphic Device displays in RGB Color Space, So Eventually after YUV Image processing, the all data should convert to **from YUV to RGB Color Space conversion stage**

FEATURES

Support

- Input / Output precision: 8 bit/channel and 10 Bit I/O /Channel
- Multiplication factors precision: 8 bit
- YUV4:4:4, YUV4:2:2, YUV4:2:0 Format
- YUV to RGB Conversion
- RGB to YUV Conversion
- Video format: 1080i/30, 1080i/25, 720p/60, 720p/50, 576p, 480p, 576i, 480i
- Video data stream clock: 27MHz...74.25MHz
- Destination platform: Xilinx FPGA Virtex, Spartan 3, Altera Cyclone, etc.

2

I/O Ports

2.1 Core Register Map

| Parameter | Type | Default | Description |
|-----------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User_Set | Bit | 8'b0 | <p>RGB to YUV / YUV to RGB .</p> <p>This registers are configured by external jumpers.</p> <p>I Byte System Register: Bit 7 6 5 4 3 2 1 0</p> <p>Bit0-1 : YUV Format</p> <p>00 : YUV 444</p> <p>01 : YUV 422</p> <p>10 : YUV 420</p> <p>11 : Reserved</p> <p>Bit2 : RGB to YUV Bit Precision</p> <p>0 : 8bit</p> <p>1 : 10 bit</p> <p>Bit3 : YUV to RGB Bit Precision</p> <p>0 : 8bit</p> <p>1 : 10 bit</p> <p>Bit4-5:</p> <p>00 : Formula 1 (in RGB to YUV)</p> <p>01 : Formula 2 (in RGB to YUV)</p> <p>10 : Formula 3 (in RGB to YUV)</p> <p>Bit 6-7:</p> <p>00 : Formula 1 (in YUV to RGB)</p> <p>01 : Formula 2 (in YUV to RGB)</p> <p>10 : Formula 3 (in YUV to RGB)</p> |

2.2 RST_B

The asynchronous reset level can be set to either active high (1'b1) or active low (1'b0).

2.3 I/O signals of YUV to RGB

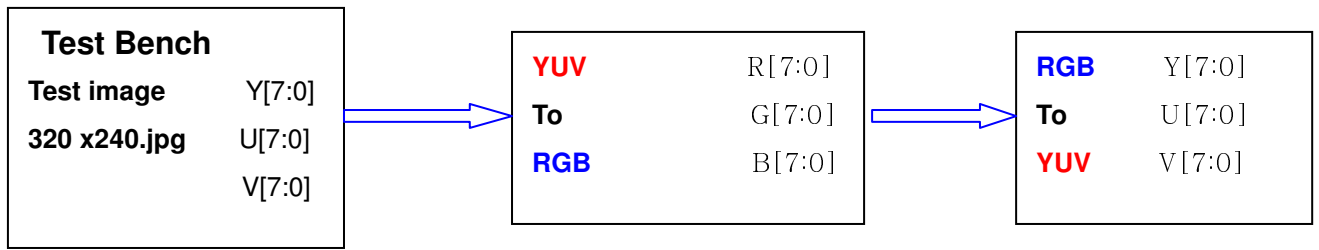
| Port | Width | Direction | Description |
|---------|-------|-----------|------------------------------------------------------------------------------|
| Clk | 1 | Input | Input Source Clock |
| Rst_b | 1 | Input | Asynchronous Reset |
| Y | 8/10 | Input | Luminance |
| U | 8/10 | Input | Chrominance Cb/U |
| V | 8/10 | Input | Chrominance Cr/V |
| R | 8/10 | Output | RED |
| G | 8/10 | Output | GREEN |
| B | 8/10 | Output | BLUE |
| ModeSet | 8 | Input | System Mode Set Register(External) Details are defined as register maps |

2.4 I/O signals of RGB to YUV

| Port | Width | Direction | Description |
|---------|---------|-----------|------------------------------------------------------------------------------|
| Clk | 1 | Input | Input Source Clock |
| Rst_b | 1 | Input | Asynchronous Reset |
| Y | 8 or 10 | Output | Luminance |
| U | 8 or 10 | Output | Chrominance Cb/U |
| V | 8 or 10 | Output | Chrominance Cr/V |
| R | 8 or 10 | Input | RED |
| G | 8 or 10 | Input | GREEN |
| B | 8 or 10 | Input | BLUE |
| ModeSet | 8 | Input | System Mode Set Register(External) Details are defined as register maps |

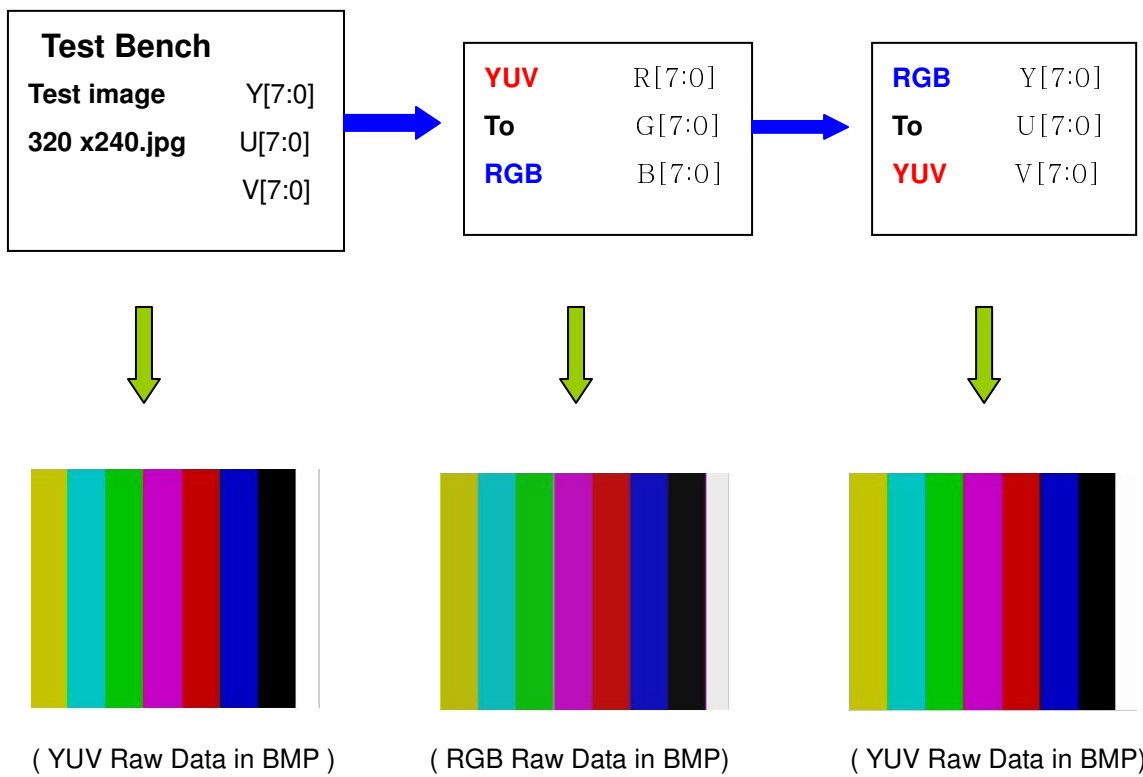
3

Architecture – System Block Diagram (8 Bit process)



4

Operations



5

Image Quality verification

- YUV to RGB Programs in Matlab Programming
- RGB to YUV Programs in Matlab Programming
- Test Bench in Verilog and image writing in RAW Data.