techXclusives

Colour Space Conversion
Part 1

By Andy Miller
Staff Engineer - Xilinx UK

Introduction

Parts 1 and 2 of these presentations are intended to walk you through the design and implementation of a Colour Space Convertor, using MathWorks Simulink and the Xilinx System Generator tool.

Colour Space Conversion is a typical DSP application found in broadcast-quality video systems.

Most DSP applications that appear complicated in theory, can often be reduced to a collection of basic functions that are well-suited for implementation in Xilinx FPGAs, (i.e., adders, subtracters, Multipliers, and delays). The Xilinx System Generator delivers a library of such functions that can be used to evaluate bit true/cycle true algorithms with the MathWorks Simulink Tool. System Generator can export the Simulink design as a hierarchy of VHDL design files and, where possible, invoke Xilinx CORE Generator™ to automatically build the bit-true cores inferred in the Simulink block diagram.

I hope this article delivers a practical introduction to the Simulink/System Generator design flow whilst also providing some useful background information on colour and the process of mapping it between different colour coordinate systems. The coding schemes discussed are YP_BP_R, YC_BC_R and RGB.

What is Colour?

In the physical world, colour does not exist outside of the human head. The experience of colour is a result of how we interpret the power levels of radiation found across a range of frequencies in the visible spectrum.

The human eye has three types of receptors that are sensitive to the power distribution of these energies. The receptors create stimulus signals for the brain, which in turn interprets them as colour. The relationship between perceived colour and Spectral Power Density (SPD) is the domain of colour scientist. However, from an engineering perspective, it is important to understand the mechanics of colour in order to create an electronic system that can capture, transmit, and reproduce it.

In 1931, an organization called the **Commission Internationale de L'Elairage** (CIE), carried out investigations to show that human colour vision is inherently *trichromatic* and requires three components (Red, Green, Blue), to mix in an additive manner. The CIE mapped the SPD of the

visible spectrum (400nm -> 700nm), to triple number coordinate systems that mathematically defines colour space.

There are different coordinate systems for mapping colour space like there are different coordinate systems for mapping 2D and 3D space. Which one is used depends on how the feature being mapped is to be expressed *(i.e., Cartesian or polar coordinates for geographical land mass)*.

Available coordinate systems mapped by the CIE are CIE XYZ, CIE L*u*v*, and CIE L*a*b*

The following diagram is taken from **"Technical Introduction to Digital Video"** by Charles Poynton. It illustrates four groups of coordinate systems used to represent colour: Tristimulus, Chromticity, Perceptibly Uniform, and Hue-Oriented.

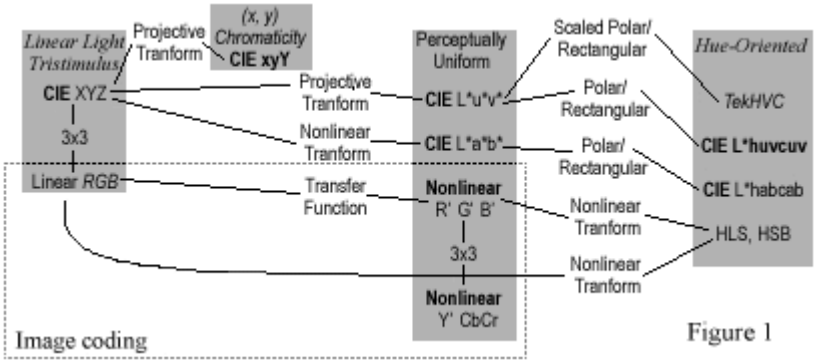


Figure 1

CIE Colour Systems Classified into 4 Groups
Source: Technical Introduction to Digital Video
Author: Charles Poynton.

The work carried out by the CIE is useful in all areas where colour reproduction is used. Video, film, and photography all benefit from the various coordinate systems, but colour science only establishes the basis for a numerical description of colour. Further transforms are required to map CIE colour space to coordinates that lend themselves to practical image coding systems.

Describing a colour in terms of three linear light components has been adopted as the basis for coding images for digital video and computer systems.

Tri-stimulus colours must be chosen and then coded into a perceptually uniform system. *(A "perceptually uniform system" is one where moving from the low end to the high end of a range in incremental steps causes equal perceptible changes in the system.)*

How are CIE XYZ tri-stimulus components derived from a color image?
Derivation of the CIE XYZ tristimulus values are complicated to understand, as they contain spectral power density figures (SPD's), which are multiplied by a matrix of colour matching functions (CMF's) that are derived from experiment! As with a lot of things in video work, "rules" are defined by experiment, and the quality of results are subjective.

Can CIE XYZ be used to drive a monitor?
Driving a real monitor with CIE XYZ signals would work, but would result in poor colour reproduction.

Constructing the Spectral Power Distribution of any colour using the **addition** of three primary SPD's mapped into a unity colour space demands that coordinates [0,0,1] [0,1,0] and [1,0,0] be achieved. Since any

colour being mapped contributes to X **and** Y **and** Z, the only way to achieve zero values in an **additive** system is to add components with negative value. This would require the use of primary colours with negative power, (which is not possible), or scale with colour-matching functions that have negative weighting, (which is also not practical, as the CIE CMF's are all positive).

(The problem is a little more complex than this, but the essence is that the CIE XYZ system of representing colour is not suitable if full-colour reproduction is required.)

To overcome this problem, the CIE XYZ component signals from the camera are transformed using a 3x3 matrix to create a set of Linear Red, Green, and Blue primaries that can define a unity 3-D colour space:

$$\begin{bmatrix} R_{600nm} \\ G_{550nm} \\ B_{470nm} \end{bmatrix} = \begin{bmatrix} 2.179151 & 0.252417 & 0.113803 \\ -1.382685 & 2.327499 & 0.045336 \\ 0.007989 & -0.015138 & 1.333346 \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

techXclusives

Colour Space Conversion
Part 1

By Andy Miller
Staff Engineer - Xilinx UK

(continued from page 1)

Can Linear RGB be used to drive the monitor?
No, because the characteristics of real monitors (Phosphor CRT's and LED's), are nonlinear. *(That is, the intensity of light reproduced at the screen of a CRT monitor is a nonlinear function of its voltage input.)*

If we applied linearly-coded colour components to a display, equal increments of each component would NOT result in perceived equal increments of colour.

Linear RGB values are transformed with another 3x3 matrix multiply to produce a set of primaries that are nonlinear in nature, but which closely match and compensate for the non-linearity of the target monitor. This is known as "gamma correction", and the gamma-corrected RGB primaries are denoted with a prime symbol (i.e., R'G'B').

In video systems, gamma correction is applied at the camera; it is usual for these signals to be available as the analogue outputs R'G'B'.

Can R'G'B' be used to drive the monitor?
Yes. These signals are industry-accepted analogue components and are often found on the interface of most video equipment.

Component Analogue RGB Signals

Gamma corrected R'G'B' components are available as analogue signals that require high-stability interfaces with respect to amplitude, level, and timing between the signals.

Ideally, all signals carry a sync pulse that assists timing and amplitude monitoring, but the gamma-corrected R'G'B' format is also implemented with sync on Green only, or on a fourth channel dedicated to sync pulses.

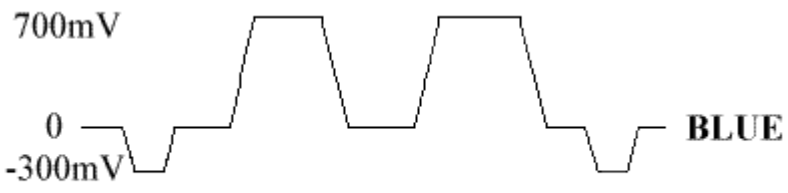
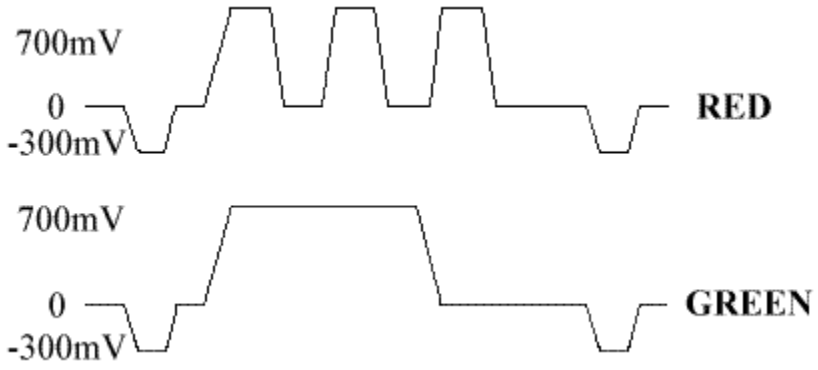


Figure 2



Unity Colour Space

The elements that now define our colour space are three signals with a range of 0->700mV, as shown below. These axes define a colour space cube often referred to as “unity colour space,” as the axes are normalised to a magnitude of 1, and any space within the cube can be located as a vector in the form (Red, Green, Blue).

i.e. pure red = (1,0,0)
pure green = (0,1,0)
pure blue = (0,0,1)

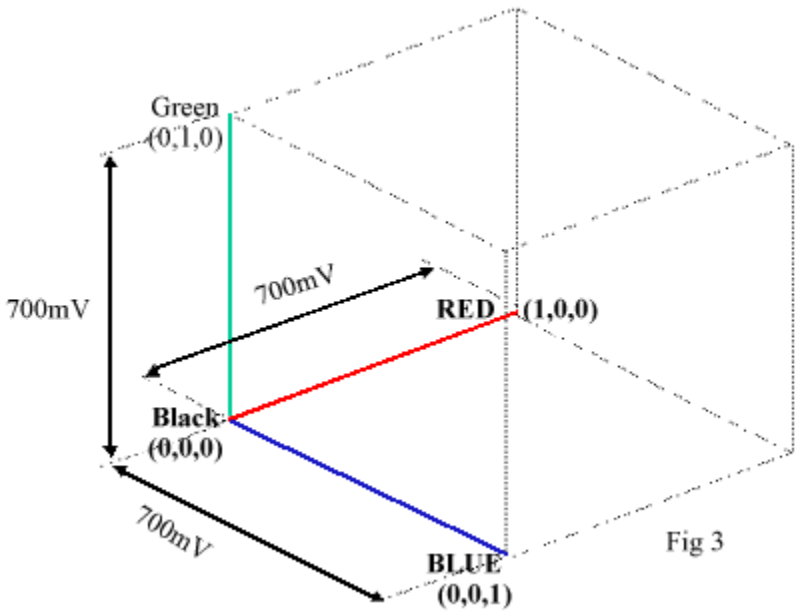


Fig 3

Coding “Yellow”

Yellow is defined as the Maximum excursion of Red + Maximum excursion of Green + NO excursion of Blue (i.e., coordinate [1,1,0]).

The term **excursion** is used because the reference is metric-independent. In the analogue domain, **max excursion** is 700mv. In the colour space domain, a **max excursion** is “1” (and anything less is a fraction). In the digital domain, max excursion depends on the coding system.

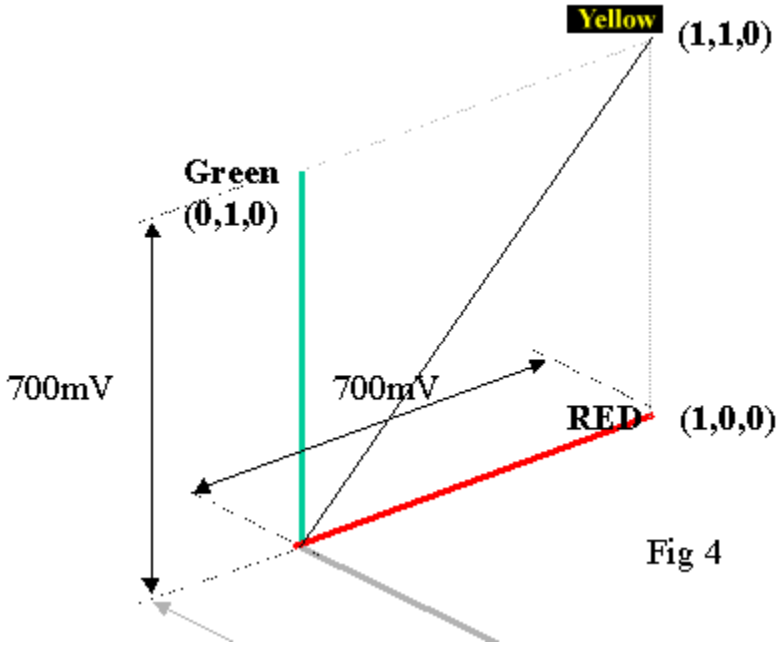


Fig 4

It is useful to be aware of something called “ITU Recommendation 601” (often abbreviated to *Rec-601*), because this defines how the colour coordinates of the “Luma and Colour Difference channel” colour space is digitally coded for 8- and 10-bit words.

techXclusives

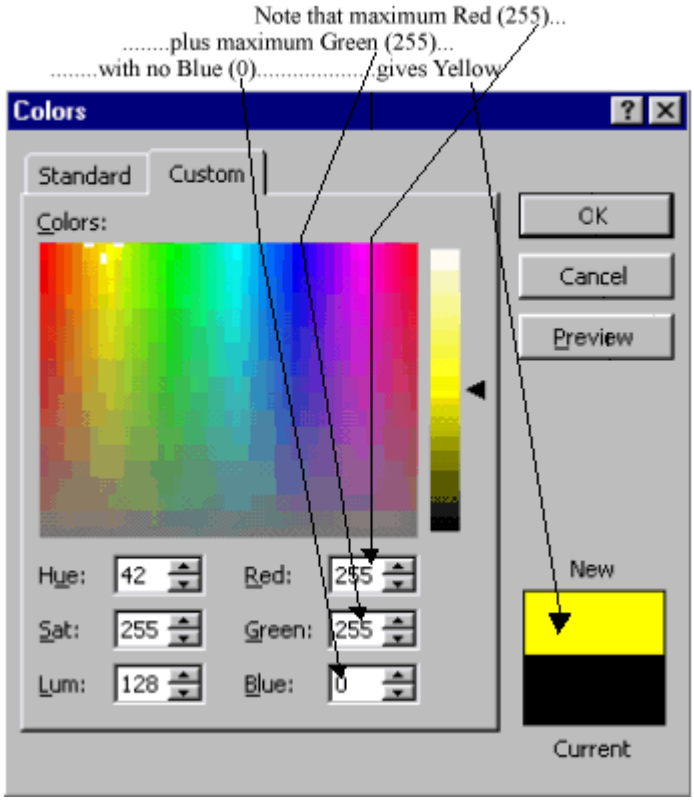
Colour Space Conversion
Part 1

By Andy Miller
Staff Engineer - Xilinx UK

(continued from page 2)

Computer-Coded “Yellow”

For computer colour coding, 8-bit unsigned binary is used so that a maximum “excursion” is “255”



At the extremes of the cube there exist a number of colours constructed from maximum excursions of the three primaries. Those identified below define the boundaries of the RGB colour space cube.

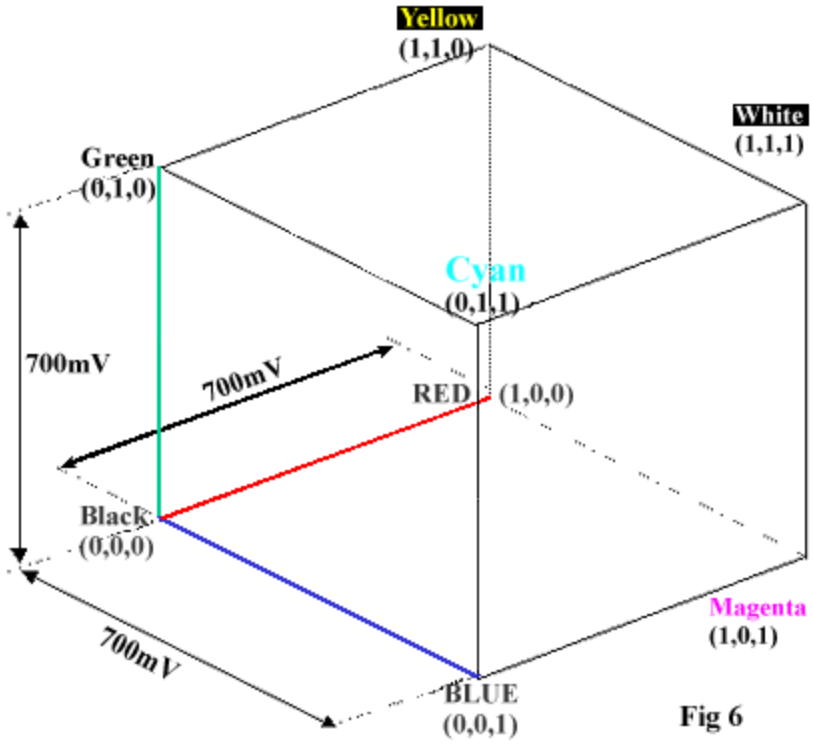


Fig 6

Colour	Colour Map Coordinate (R,G,B)
Red	(1,0,0)
Green	(0,1,0)
Blue	(0,0,1)
Black	(0,0,0)
White	(1,1,1)
Yellow	(1,1,0)
Magenta	(1,0,1)
Cyan	(0,1,1)

Luminance & Colour difference Coding (Y,B-Y,R-Y)

Whether the final display medium is paper (printing), polymer displays (portable PC's), or CRT's (TV), the idea of a quantised piece of 2-D space (the pixel), still holds true.

Each pixel can be expressed with 3 full bandwidth analogue components (R'G'B'); however, transmitting or storing these color components requires a lot of bandwidth.

Reducing this bandwidth is the objective of luminance and colour difference encoding.

CIE tests noted that human vision has less spatial acuity for colour information than it does for brightness information. If R'G'B' component video is re-coded as a channel of brightness (i.e., luminance -- denoted as

Y) and two colour difference channels (R-Y and BY), the colour difference data can be spatially reduced without the eye detecting that a change in colour has taken place.

When the brightness content of a colour scene is analysed in terms of its RGB components, Green is shown to carry 60% -> 70% of the brightness information. Because the human eye is more receptive to brightness than to colour, we can throw away some of the Red and Blue data by coding the difference between Red and Luma and Blue and Luma, instead of Red and Blue themselves. So the colour difference channels are created by subtracting luminance (Y) from the Red and Blue channels, as this has less impact on the brightness, and Green can be reconstructed from the colour difference channels.

(This basically means that you can throw away some of the Red and Blue data when transmitting the brightness and the colour difference channels!)

The unit R'G'B' colour space cube is now transformed into a new set of coordinates where the brightness (luminance Y') is mapped in the vertical axis, and the colour difference channels BY and R-Y can be regarded as colour coordinates.

So how are R'G'B' coordinates transformed into Y, BY, R-Y coordinates? How do we map gamma-corrected unity colour space into the coordinates that describe a "brightness and colour difference" colour space?

Mapping (R'G'B') to (Y, BY R-Y)...1

In 1990, the radio-communications group of the International Telecommunications Union made a recommendation that defines how to compute nonlinear video luma from nonlinear Red, Green, Blue primaries. This is now an international standard (ITU-R BT.601- 4), and is often abbreviated to *Rec. 601*.

601-compliant luma is defined as:

$Y'_{601} = 0.299R' + 0.587G' + 0.114B'$ ….. = 1 (Equation 1)

This equation defines how to map R'G'B' colour space to the luminance on a vertical axis. Y' is normalised to 1, and the colour difference channels are computed by subtracting this luminance from the Red and Blue channels. *(Just accept it, because that's the way it is!)*

At full excursion, the vector coordinate for Blue is [0,0,1], and the vector coordinate for Luma Y'_{601} is shown above. Subtracting Luma from Blue to obtain ($B'-Y'_{601}$) requires a vector subtraction.

$$B'-Y'_{601} = [0,0,1] - [0.299 \quad +0.587 \quad 0.114]$$
$$= [(0 - 0.299) \quad (0 - 0.587) \quad (1 - 0.114)]$$
$$= [-0.299 \quad -0.587 \quad 0.886] \qquad \text{(Equation 2)}$$

So, the colour difference channel 'Blue minus Luma, ($B'-Y'_{601}$) is computed from the unity space coordinates (R'G'B') using the following vector multiplication:

$$B'-Y'_{601} = [-0.299 \quad -0.587 \quad 0.886] \bullet \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \qquad \text{(Equation 3)}$$

This is repeated for the Red minus Luma colour difference channel.

Mapping (R'G'B') to (Y, BY R-Y)…2

At full excursion, the coordinate for Red is [1,0,0]. Now, perform the vector subtraction of Luma from the red channel:

$$\begin{aligned} R'-Y' &= [1,0,0] - [0.299 + 0.587 + 0.114] \\ &= [(1 - 0.299) \quad (0 - 0.587) \quad (0 - 0.114)] \\ &= [\,0.701 \quad -0.587 \quad -0.114\,] \end{aligned} \qquad \text{(Equation 4)}$$

So, the R'-Y' channel transform is computed as follows:

$$R'-Y' = [0.701 \quad -0.587 \quad -0.114] \bullet \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \qquad \text{(Equation 5)}$$

The complete transform of Gamma-corrected RGB (unity colour space) to Rec 601 [Y' R'-Y', B'-Y'] (Luma & colour difference colour space) is given as the following matrix multiplication:

$$\begin{bmatrix} Y'_{601} \\ B'-Y'_{601} \\ R'-Y'_{601} \end{bmatrix} = - \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \bullet \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

(Equation 6)

If we take the coordinates that define the boundaries of unity colour space and apply them to the above equation, we will map unity colour space into the "Luma & colour difference" colour space that expresses colour more efficiently for transmission.

Part 2 of Andy Miller's article will lead you through a step-by-step example of this theory, offer more information, and present related examples that utilize the MathWorks Simulink and Xilinx System Generator tools. Check back on March 26 for PART 2.