techXclusives

Colour Space Conversion
Part 2

By Andy Miller
Staff Engineer - Xilinx UK

The following PowerPoint file contains a slide show that graphically demonstrates the conversion of unity RGB color space into Luma and Colour Difference colour space.

Playing the presentation will enable you build up the conversion step-by-step.

Finally, the diagram of Luma and Colour Difference colour space often viewed in popular video text books is achieved and a matrix equation derived for the conversion (Equation 10):

Download the conversion example
Power Point presentation

Hardware For Equation 10 - RGB2YPBPR

Equation 10 suggests that a colour space converter can be built with nine multipliers and six adders. The circuit below would be the hardware equivalent of Equation 10. In fact, it is the foundation of the Logic Devices Colour Space Convertor/Corrector LF2272 (Previously known as Raytheon TMC2250).

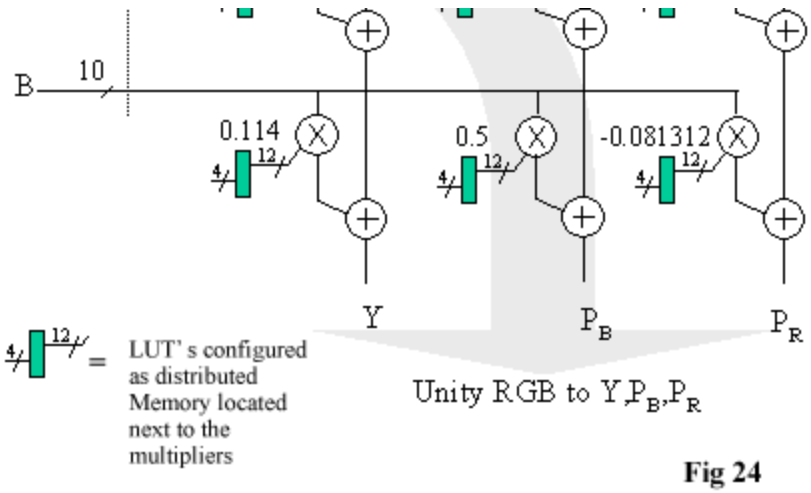
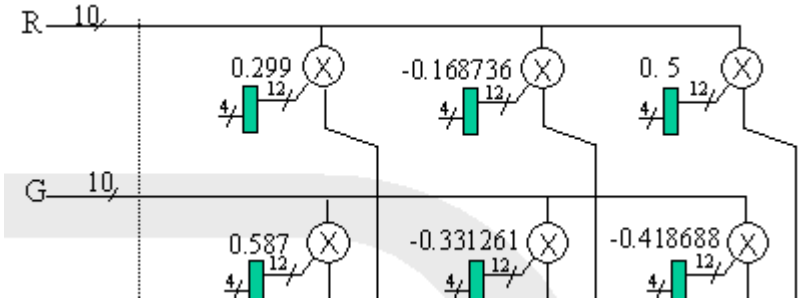


Fig 24



The above structure can realise any 3x3 transform between different colour coordinates; the only difference between an RGB2PBPR and YPBPR2RGB is the 12-bit coefficients. Banks of coefficients can be stored local to each multiplier using Xilinx Look-Up Tables configured as distributed RAM. Up to sixteen 12-bit coefficients could be stored in just six Virtex™ or Spartan™ slices.

A More Optimum RGB2YPBPR Solution

Figure 25 is a more optimum solution than Figure 24 because it requires only five multipliers, two adders, and two subtracters. Three multipliers and two adders are used to construct the Luma (Y), which is then subtracted from the Red and Blue data to construct the scaled colour difference channels PB and PR.

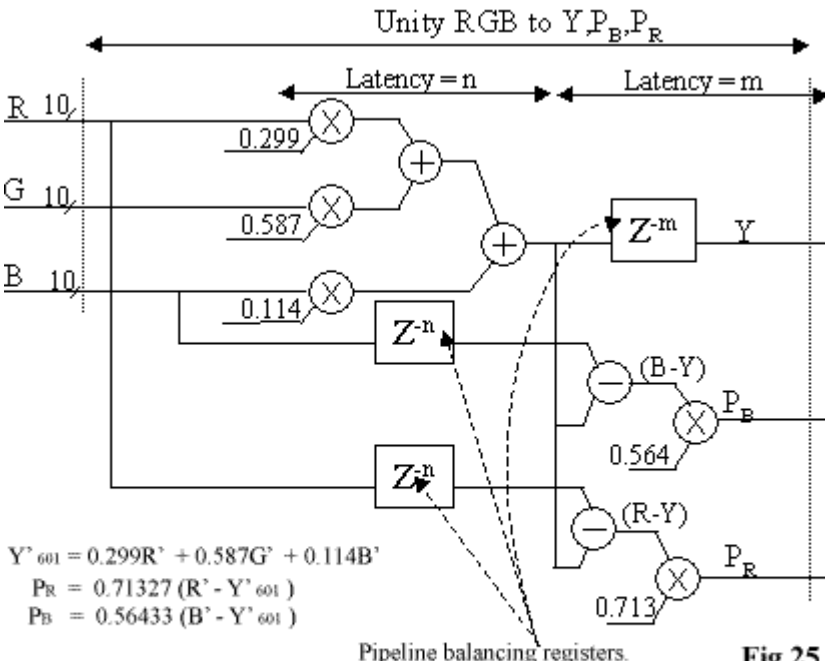
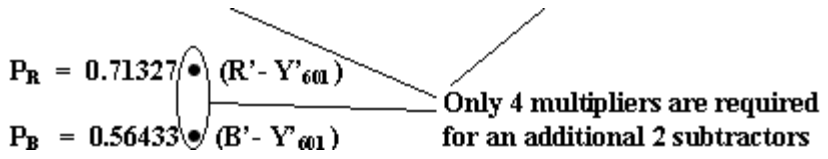


Fig 25

Even Smaller!

A further optimization to the circuit of Figure 25 is to re-express the Green contribution to the Luma as a none-scaled item (i.e., no multiplier is required to scale the Green data when generating 'Y').



techXclusives

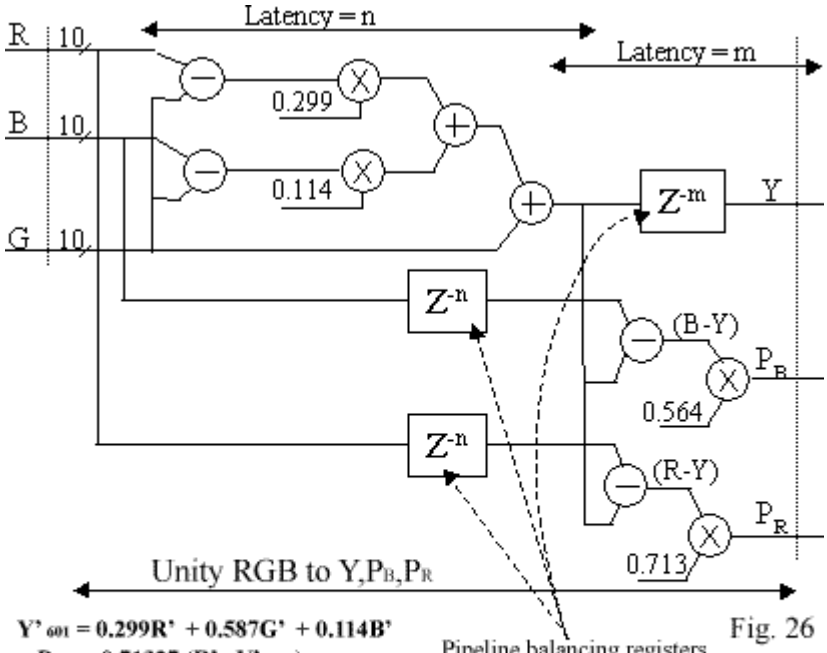
Colour Space Conversion
Part 2

By Andy Miller
Staff Engineer - Xilinx UK

(continued from page 1)

A More Optimum RGB2YPBPR Solution

Figure 26 shows 2 multipliers, 2 adders, and 2 subtracters used to construct the Luma component (Y). This solution will be smaller than that of Figure 25, due to the need for 1 fewer multiplier at the expense of just two subtracters.



$Y'_{601} = 0.299R' + 0.587G' + 0.114B'$
$P_R = 0.71327 (R' - Y'_{601})$
$P_B = 0.56433 (B' - Y'_{601})$

Fig. 26

Pipeline balancing registers
It is necessary to balance the delays through all paths of the design so that the coded Y PB PR values are derived from the correct set of RGB inputs.
Note that the Virtex SRL16's are ideal for such pipeline delays.

Coding PBPR to Give CBCR

So far, you have seen diagrams for RGB to Y PBPR.

PBPR are the ± 0.5V analogue colour-different outputs often found on the

rear panels of broadcast studio equipment. The signals can also be sent as digital video components in both serial and parallel form at Standard Definition (SD) and High Definition (HD), data rates, 27MHz and 74.25MHz respectively.

The quantization requirements for coding Y P_BP_R as Y C_BC_R are defined in Recommendation ITU-R BT.601-4 601, often referred to as *Rec. 601*.

Rec 601 is primarily concerned with broadcast quality imaging and outlines a quantization scheme that provides some “headroom” and “footroom” in the quantized values for Y C_B and C_R. It is worth noting that Rec-601 originally proposed 8-bit values for digital component video, but it is now industry standard to use10-bits. Rec-601 reflects this modification by stating that all 10-bit values should be treated as an 8-bit exponent and 2-bit mantissa. The values used in Rec-601 are given in 8-bit notation, and this has been maintained below so as not to confuse the reader by introducing 10-bit values.

The full excursion of the Luma is quantized across 220 levels (0->219), and footroom in the signals is provided by offsetting the minimum Luma signal to value 16. Hence, Y_{601} is achieved by multiplying the colour converted Y output by 219 and adding 16 to the result. Headroom in the Y_{601} signal is achieved by the scaling multiplier and offset, which ensure that a maximum excursion of Y gives a Y_{601} value of 235. This leaves values (236->254)** as headroom.

The full excursion of each colour difference channel is quantized across 225 levels (0->224) by multiplying the 2’s compliment P_BP_R signals by 224 and applying an offset of +128. This causes the 2’s compliment C_BC_R outputs to swing across the digital range -112 and +112, leaving values (0->15) and (113 ->127) as footroom and headroom in the Rec-601-compliant digital output.

*** Levels 0 and 255 are used exclusively for synchronization.*

An RGB to YC_BC_R Solution

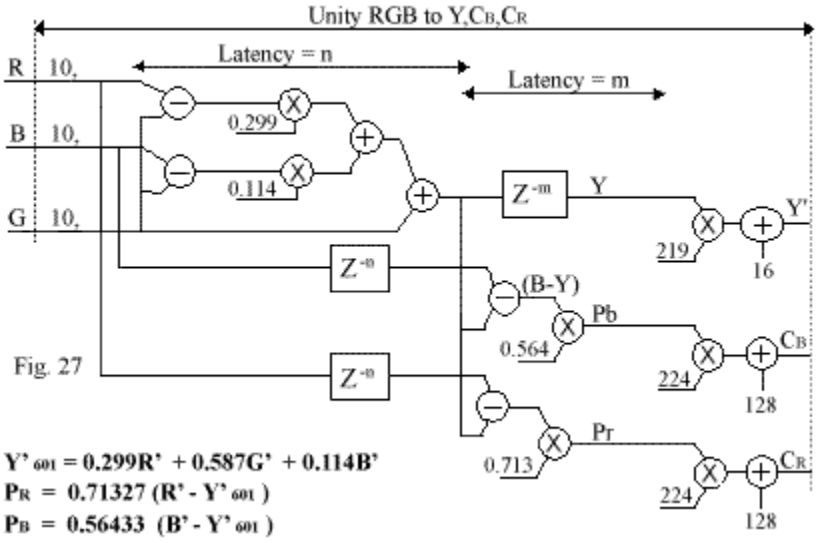


Fig. 27

$Y'_{601} = 0.299R' + 0.587G' + 0.114B'$
$P_R = 0.71327\ (R' - Y'_{601})$
$P_B = 0.56433\ (B' - Y'_{601})$

ARTICLE SUMMARY

This section has provided some background material into the principles of colour and how it is coded and transmitted in different coordinate systems that define "colour space."

Hopefully, the terms (RGB), (Y B-Y R-Y) (Y,P_B, P_R) and (YC_BC_R) should now have some meaning when you hear the term "colour space converters" in the future.

This article has only covered the conversion from RGB to digital component colour space. The reciprocal conversion is similar in its use of multipliers to scale data and adders/subtracters to add and remove offsets given by Rec-601.

FURTHER INFORMATION:

Having derived a block diagram of what needs to be built, the following Power Point presentations move on to the Matlab/Simulink and System Generator domain to explore how the theory can be modeled:

Section A	Section B	Section C
A Step-By-Step Description of the System Generator Flow For a Colour Space Convertor	A Step-By-Step Description of the Synplicity Flow	Passing the EDIF netlists to the Xilinx Place and Route tools. (Xilinx Alliance 3.3i software.)

$$Y'_{601} = 0.299\,\text{Red} + 0.587\,\text{Green} + 0.114\,\text{Blue} \quad \textit{(from Equation 1)}$$

$$Y'_{601} = 0.299\,\text{Red} + (\text{Green} - 0.483\,\text{Green}) + 0.114\,\text{Blue}$$

$$Y'_{601} = 0.299\,\text{Red} + (\text{Green} - 0.114\,\text{Green} - 0.299\,\text{Green}) + 0.114\,\text{Blue}$$

$$Y'_{601} = 0.299\,\text{Red} + \text{Green} - 0.114\,\text{Green} - 0.299\,\text{Green} + 0.114\,\text{Blue}$$

$$Y'_{601} = (0.299\,\text{Red} - 0.299\,\text{Green}) + \text{Green} + (0.114\,\text{Blue} - 0.114\,\text{Green})$$

$$Y'_{601} = 0.299 \odot (\text{Red} - \text{Green}) + \text{Green} + 0.114 \odot (\text{Blue} - \text{Green})$$