



XAPP294 (Draft) December 19, 2001

# Digital Component Video Conversion 4:2:2 to 4:4:4

Author: Gregg Hawkes

## Summary

The video standard ITU-R BT.601 was introduced as the need for transporting digital component video between countries and standards increased. The analog component R'G'B' can be sampled in a very regular way and converted from 4:4:4 to the digital 4:2:2 format, essentially cutting in half the number of different components, Cr and Cb.

The digital data is efficiently stored or transmitted to a destination that reverses the process, i.e., converts back to 4:4:4 format, and produces analog YUV or R'G'B' for display. This application note provides technical details surrounding video format conversion and how it is accomplished in the MicroBlaze and Multimedia development board.

## Introduction

When taking live or studio video into a system as 4:2:2 format (reference ITU-R BT.601 and ITU-R BT.656-4)<sup>[1]</sup>, Chroma (Cb and Cr) is provided at 1/2 the data rate as Luma. Since the human eye is less sensitive to the color difference of Chroma signals, the amount of digital video storage and digital video bandwidth can be reduced. Relative to the incoming Luma stream, only the even Chroma values are sampled. To get the missing Chroma values, i.e., convert back to 4:4:4, interpolate the available values. One goal in converting back to an analog format is to calculate the missing values without producing undesirable artifacts.

To adhere to the output frequency response specified in the ITU-R BT.601 standard, pay close attention to the scaling, over-flow, and under-flow of the final R'G'B' values. Failure to do this results in an incorrect data display on the destination equipment. Several filter functions can be used to match the reference. The result is a compromise between complexity, cost, and standard adherence.

## Sampling Schemes

### 4:2:2 and 4:4:4 Sampling Schemes

The video standard ITU-R BT.656 describes how to embed video-timing information in the 4:2:2 bit-parallel sampling scheme of the Y'CrCb color space definition. It also clearly shows the relationship of Luma and Chroma, with Luma values arriving at twice the data rate as Chroma. Refer to application note [XAPP286 "Video Line Field Decode"](#) for this information. The video data words are conveyed as a 27 MHz data stream in the following order:

Cb0, Y'0, Cr0, Y'1, Cb1, Y'2, Cr1, Y'3, Cb2, Y'4, Cr2, Y'5...

Figure 1 and Figure 2 conceptually shows how the 4:2:2 and 4:4:4 data streams are different.

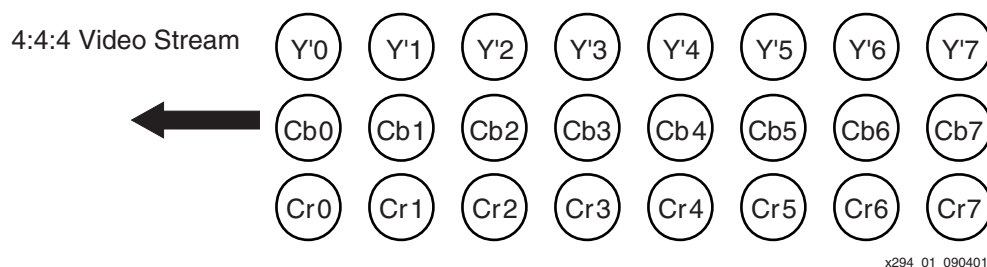


Figure 1: Digital Component 4:4:4 Format

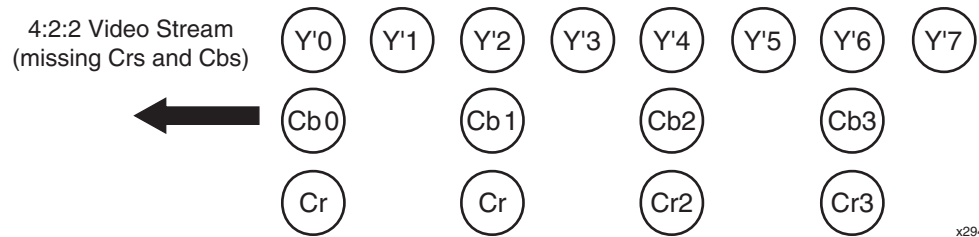


Figure 2: Digital Component 4:2:2 Format

There has been considerable debate over sample frequency. A frequency of 12 MHz was too low, as it approached Nyquist limitations. A higher frequency posed other design issues. A sample frequency of 13.5 MHz supports both NTSC and PAL standards. At this rate, NTSC lines contain 858 samples (720 active) with 525 total lines (485 active lines) and PAL lines contains 864 samples (720 active) with 625 total lines (576 active lines).

The conversion between PAL and NTSC is relatively straightforward. Unfortunately, in a multimedia environment, the data must eventually be transformed from Y'CrCb 4:2:2 to R'G'B' 4:4:4 for further processing.

## Converting from 4:2:2 to 4:4:4

This section suggests a few filter functions with varying size and performance trade-offs. All of the solutions have a pixel clock (27 MHz) used to clock in each Y', Cr, and Cb 10-bit value, in sequence. The design outputs are Y', Cr, and Cb (30 bits) transferred at a rate of 13.5 MHz. Several different filter approaches are suggested. The sample code for method 3 is available on the Xilinx ftp site at: <ftp://ftp.xilinx.com/pub/applications/xapp/xapp294.zip>.

Some helpful reference material on this subject is available from, Charles Poynton at [www.inforamp.net/~poynton](http://www.inforamp.net/~poynton)<sup>[1]</sup>. Although, filters may not completely adhere to the ITU-R BT.601 standard, one of the important features of an FPGA is the ability to experiment and evaluate the results empirically. Three of Poynton's suggestions are further discussed in this application note.

## Method 1

### Simple Cb or Cr Replication

If the design goal is regular VHS-quality video, the missing Crs and Cbs can be replicated from previous ones. Assuming no data sampling inaccuracies on the input, the output should never exceed the frequency limitations. Of course, since there are always inaccuracies, checking the resulting Crs and Cbs for overflow and underflow is a good idea.

### Two Pixel Linear Interpolation

Poynton mentions in his work that a linear interpolation gives better results. Equation 1 is merely an average of two available Crs or Cbs to arrive at a missing one.

$$Cb[i] = \frac{(Cb[i-1] + Cb[i+1])}{2} \quad \text{Equation 1}$$

Figure 3 shows how this Cr or Cb reconstruction using four input components looks pictorially.

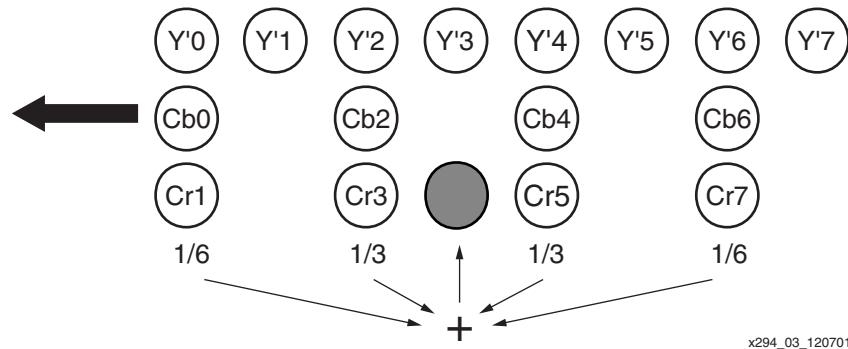


Figure 3: Reconstructing 4:4:4 Format From 4:2:2

## Method 2

### Parallel FIR Filter

Higher quality video requires a suitable FIR filter. If the coefficients do not sum to one, then the result should be scaled appropriately. The second equation has coefficients, suggested by Poynton, that sum to 256. The result is divided by 256. This can be done in hardware by a simple 8-wire shift.

Equation 2:

$$\begin{aligned} \text{Cb}[i] = & (+160(\text{Cb}[i-1] + \text{Cb}[i+1]) - 48(\text{Cb}[i-3] + \text{Cb}[i+3]) + 24(\text{Cb}[i-5] + \text{Cb}[i+5]) \\ & - 12(\text{Cb}[i-7] + \text{Cb}[i+7]) + 6(\text{Cb}[i-9] + \text{Cb}[i+9]) - 42(\text{Cb}[i-11] + \text{Cb}[i+11]))/256 \end{aligned}$$

For this set of equations, Poynton further recommends watching for signs, scaling, overflow, and end conditions. He suggests surrounding the picture data with zero samples. One experiment is a method where a screen has duplicate pixels around the border. Be sure to scale "Cb and Cr" to real B'-Y' and R'-Y'. Further discussion is in application note: [XAPP283: "Y'CrCb to R'G'B' Color Space Converter"](#).

## Method 3

### Parallel FIR Filter

The following Xilinx labs filter meets the specification requirement. Even though it takes advantage of symmetry, where the "like terms" in the equations are summed before the corresponding multiply, there are still 12 multiplies.

Equation 3:

$$\begin{aligned} \text{Cb}[i] = & (-4(\text{Cb}[i-23] + \text{Cb}[i+23]) + 6(\text{Cb}[i-21] + \text{Cb}[i+21]) - 12(\text{Cb}[i-19] + \text{Cb}[i+19]) \\ & + 20(\text{Cb}[i-17] + \text{Cb}[i+17]) - 32(\text{Cb}[i-15] + \text{Cb}[i+15]) + 48(\text{Cb}[i-13] + \text{Cb}[i+13]) \\ & - 70(\text{Cb}[i-11] + \text{Cb}[i+11]) + 104(\text{Cb}[i-9] + \text{Cb}[i+9]) - 152(\text{Cb}[i-7] + \text{Cb}[i+7]) \\ & + 236(\text{Cb}[i-5] + \text{Cb}[i+5]) - 420(\text{Cb}[i-3] + \text{Cb}[i+3]) + 1300(\text{Cb}[i-1] + \text{Cb}[i+1]))/2048; \end{aligned}$$

This equation is implemented as shown in [Figure 4](#).

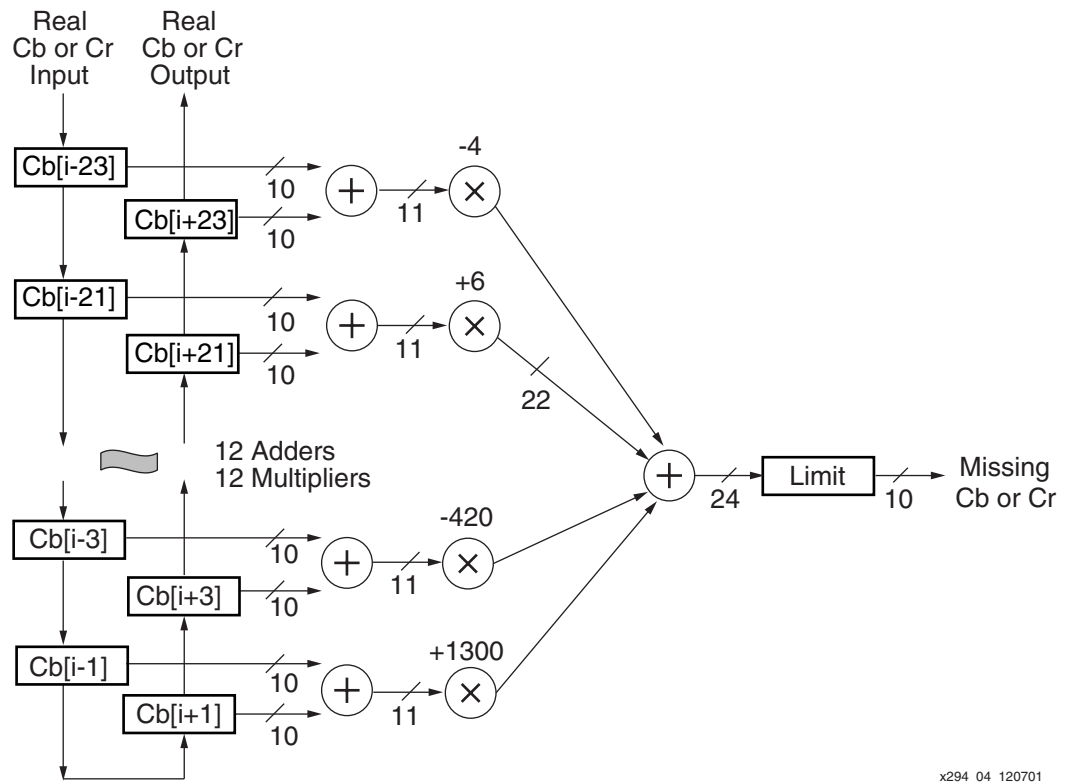


Figure 4: 24 TAP, Parallel FIR Implementation of 4:2:2 to 4:4:4

## Efficient Video Math for SDTV and HDTV

There are a number of arithmetic techniques available in Virtex and Spartan-II devices. First, the multiplication and other math requirements should be analyzed. Remember, for video pipeline applications, there are only two pixel sample rates of concern: 13.5 MHz for Standard-Definition Television (SDTV) and 72.5 MHz for High-Definition Television (HDTV). Big multipliers are also not necessary since consumer equipment has 8-bit color components and studio quality equipment has 10-bit representations.

The analysis and detailed information for some video applications is provided in a separate application note: [XAPP249 "Efficient Mathematics Implementations for Video in Virtex FPGAs"](#).

More details on the MULT\_AND, LUTs, and carry logic used to form general, efficient multipliers are provided in the application note: [XAPP215 "Design Tips for HDL Implementation of Arithmetic Functions"](#). Briefly, the MULT\_AND forms all of the two-bit partial products in a multiplication. For a 10 x 10, there are 100 such partial products all formed without consuming a single LUT. Next, the efficient and extremely fast ADD and CARRY LOGIC sums the partial products for the final result.

Based on these results, the MicroBlaze and Multimedia development board designs can use two mathematical approaches. Synthesized multipliers from a multiplication equation expressed in HDL using the inferred MULT\_ANDs are preferred when targeting HDTV video rates. The constant coefficient version of these multiplies, on average, equate to very few LUTs. The multiplies are able to run at HDTV rates even in slower Virtex devices.

If the design needs to push the limits of small size and targets mainly SDTV rates, then the Xilinx Core Generator should be used. At SDTV rates and for cases where many multiplications contribute to a single result (as in de-interlace, parallel FIR filters, FFTs, etc.), the Core Generator will give the most efficient result.

Many of the DSP and Math cores are designed to take advantage of separating the summation of many multiplications, i.e., a polynomial, into distributed partial products to be collected at the

correct scaling and summed as a final step. The work of designing a distributed arithmetic solution is handled by the software, including easy input of data and coefficient widths, coefficient data, number of pipeline stages, etc. Some of the solutions do require multi-sample rate clocks, but this tool is the most efficient. The standard definition clock rate is 13.5 MHz. Even using 10-bit studio quality samples, a bit rate clock is only 135 MHz. The Virtex-II Digital Clock Manager (DCM) easily generates the high-speed bit rate clock from the pixel clock. For Virtex and Spartan-II devices, an external 135 MHz crystal can be used.

After installing the tool, download the latest libraries from the Xilinx web and look through the GUIs folder arrangement for possible solutions. The FIR filters are under the DSP folder. The online data sheets provide detailed implementation descriptions, as well as expected size, shape, and speed in targeted devices. An RLOCed version of most cores is available to guide the Xilinx map, place, and route software.

## Reference Design Results

**Table 1** outlines the performance results of the reference design for **Method 3**. The file [xapp294.zip](#) includes the synthesis and implementation results. Each input video stream in the Xilinx MicroBlaze and Multimedia development board uses this code to convert Y'CrCb, 4:2:2 values to Y'CrCb, 4:4:4 values prior to color space conversion. All results were obtained using the Verilog versions of the designs with Xilinx ISE version 4.1i using XST as the synthesis tool. Results using the VHDL files are not shown, but are essentially identical. Virtex-II device results are for a -5 speed grade device. Spartan-II device results are for a -6 speed grade device.

**Table 1: Reference Design Results**

Design Name	Size LUTs/FFs	Speed Virtex-II Device	Speed Spartan-II Device	Ports	Power Consumption

## Conclusions

This application note presents three methods of generating the missing chroma to convert from the 4:2:2 format to the 4:4:4 format. As in most video applications, the quality of the results is a traded-off with the FPGA resources (silicon cost). Method 1, taking the average of the two nearest neighbor chroma values, is the simplest. Only requiring an adder to sum the two nearest chroma values and a wire shift to divide the results of the addition by two. Method 2 is a slightly more complex parallel FIR example where four samples are combined using a "weighted" average to form the missing chroma value. Method 3, used on the Microblaze and Multimedia Demonstration Board, has a 24 tap, symmetric, parallel, FIR filter with coefficients determined from MATLAB to meet the criteria in the ITU-R BT.601 specification.

## References

1. Charles Poynton, tel: +1 416 413 1377, fax: +1 416 413 1378, poynton@poynton.com [www.inforamp.net/~poynton](http://www.inforamp.net/~poynton)
2. The video standards beginning with ITU come from the International Telecommunication Union. ITU-R BT.656 and by ITU-R BT.601 standards are available on the International Telecommunication Union's web site, <http://www.itu.int/itudoc/itu-r/rec/bt/>, for a small fee. The SMPTE or Society of Motion Picture and Television Engineers standards can be found on <http://www.smpite.org> and will also require membership or a fee.
3. Video Demystified, by Keith Jack, published by Harris, ISBN 1-878707-23-X, is a good beginners guide to video techniques. It can be read or purchased on line at the following URL: <http://www.video-demystified.com>

4. Video Demystified - Third Edition, Author: Keith Jack, LLH Technology Publishing,  
[www.LLH-Publishing.com](http://www.LLH-Publishing.com)

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/19/01	Draft	Initial Xilinx release.