



XAPP931 (v1.0) 2006 年 5 月 9 日

カラー スペース コンバータ : YCrCb から RGB

著者 : Gabor Szedo

はじめに

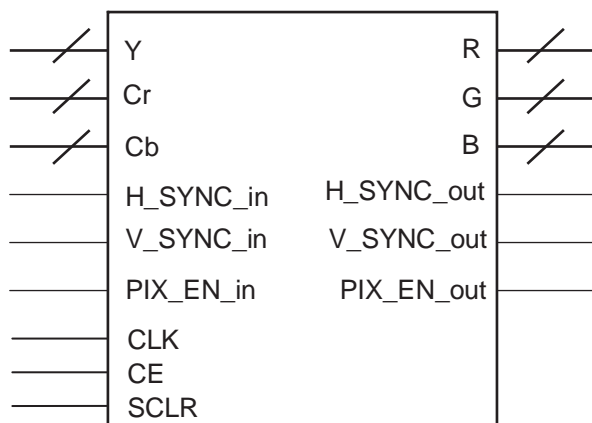
このアプリケーション ノートでは、多数のビデオ デザインで必要となる YCbCr から RGB へのカラー スペース変換回路のインプリメンテーションについて説明します。リファレンス デザイン ファイルには、5 つの乗算器のみを使用して YCrCb から RGB への変換を実行する最適化済みストラクチャを定義する RTL VHDL コードが含まれます。コンパイルするソース ファイル順は次のとおりです。

1. GenXlib_util.vhd
2. GenXlib_arch.vhd
3. color_space_pkg.vhd
4. Xil_YCrCb2RGB.vhd

System Generator を使用する場合は、HDL コードをカプセル化する System Generator トークンが利用可能です。また、System Generator のテストベンチを使用すると、出力結果を GUI で確認できます。コードは、入力/出力精度 (8 ビットまたは 10 ビット)、内部ワード長、および係数精度 (8 ビットから 18 ビットが定義済み) に対応するようパラメータ設定されています。多くの規格に対応する、一般的なスケールリング、オフセット、クリッピング、およびクランピング パラメータが提供されています。

概要

リファレンス デザインの CE、CLK、および SCLR ポートは完全な同期インターフェイスです。ポート Y、Cr、Cb は YCrCb カラー スペース入力、R、G、B は RGB カラー スペース出力です (図 1 参照)。



x931_01_032406

図 1 : YCrCb - RGB コンバータのピン

実際のビデオ システムへすぐに挿入できるリファレンス デザインを活用すると、最大 3 つのストリーム制御信号 (H_SYNC、V_SYNC、PIX_EN) が使用可能で、これらは適切に遅延させることができるため、制御信号と出カストリームの同期化が容易です。リファレンス デザインでは制御信号を使用していないので、これらの信号の接続はオプションとなります。

© 2006 Xilinx, Inc. All rights reserved. すべての Xilinx の商標、登録商標、特許、免責条項は、<http://www.xilinx.co.jp/legal.htm> にリストされています。他のすべての商標および登録商標は、それぞれの所有者が所有しています。すべての仕様は通知なしに変更される可能性があります。

保証否認の通知 : Xilinx ではデザイン、コード、その他の情報を「現状有姿の状態」で提供しています。この特徴、アプリケーションまたは規格の一実施例としてデザイン、コード、その他の情報を提供しておりますが、Xilinx はこの実施例が権利侵害のクレームを全く受けないということを表明するものではありません。お客様がご自分で実装される場合には、必要な権利の許諾を受ける責任があります。Xilinx は、実装の妥当性に関するいかなる保証を行なうものではありません。この保証否認の対象となる保証には、権利侵害のクレームを受けないことの保証または表明、および市場性や特定の目的に対する適合性についての黙示的な保証も含まれます。

パラメータ設定

表 1 に、デザインの入力パラメータを示します。

表 1: デザイン パラメータ

デザイン パラメータ	タイプ	範囲	説明
FAMILY_HAS_MAC	整数	0、1	ターゲット デバイスに MAC ユニットなどの DSP48 が搭載されている場合は 1 ⁽¹⁾
FABRIC_ADDS	整数	0、1	加算器/減算器をファブリックにインプリメントする必要がある場合は 1 ⁽¹⁾
IWIDTH	整数	8、10	入力 (RGB) データ幅
CWIDTH	整数	8 ~ 18	係数データ幅
MWIDTH	整数	IWIDTH+1 ~ 18	エンベデッド乗算器幅 ⁽²⁾
OWIDTH	整数	8、10	出力 (YCrCb) データ幅
RGBMAX	整数	0 ~ 2 ^{OWIDTH-1}	R、G、B 出力のクリッピング値
RGBMIN	整数	0 ~ 2 ^{OWIDTH-1}	R、G、B 出力のクランピング値
COFFSET	整数	0 ~ 2 ^{OWIDTH-1}	色度 (Cr、Cb) 出力のオフセット値
ACOEFF	整数	-2 ^{CWIDTH} ~ 2 ^{CWIDTH-1}	係数 A の値 ⁽³⁾
BCOEFF	整数		係数 B の値 ⁽³⁾
CCOEFF	整数		係数 C の値 ⁽³⁾
DCOEFF	整数		係数 D の値 ⁽³⁾
ROFFSET	整数	0 ~ 2 ^{OWIDTH-1}	R 出力のオフセット値
GOFFSET	整数	0 ~ 2 ^{OWIDTH-1}	G 出力のオフセット値
BOFFSET	整数	0 ~ 2 ^{OWIDTH-1}	B 出力のオフセット値
HAS_CLIP	整数	0、1	出力にクリッピング ロジックがある場合は 1 ⁽⁴⁾
HAS_CLAMP	整数	0、1	出力にクランピング ロジックがある場合は 1 ⁽⁴⁾

メモ :

1. 図 3 を参照してください。
2. 「エラー解析」を参照してください。
3. 「デザイン パラメータ値の割り当て」を参照してください。
4. 「出力クリッピング ノイズ」を参照してください。

詳細説明

カラー スペース

カラー スペースとは、色を規定する数学的表現体系のことであり、次の 3 つの主要モデルがあります。

- RGB (コンピュータ グラフィックで使用)、R'G'B' (ガンマ補正済み RGB)
- YIQ、YUV、YCrCb (ビデオ システムで使用)
- CMYK (カラー プリントで使用)

ただし、カラー スペースは色相、彩度、および明度の知覚的認識に直接関連します。

すべてのカラー スペースは、カメラやスキャナなどの機器からの RGB 情報を元に再現できます。

各カラー スペースには、異なるアプリケーション用に発展してきた経緯があり、それぞれのカラー スペースが選択された背景には特定の理由があります。たとえば、必要なストレージ、バンド幅、アナログまたはデジタルでの演算処理が少ないアプリケーションでは、それ用により適したカラー スペースが選択されました。

それぞれのカラー スペースが選択されてきた経緯にいかなる理由があっても、コンピュータ、インターネット、多様なビデオ機器はすべて異なるカラー体系を使用しており、今日のデジタル設計者は、カラー スペース変換を含めた設計を行う必要があります。つまり、アルゴリズムおよびプロセス実行前にすべての入力を共通のカラー スペースに変換する必要があります。コンバータは、画像処理やビデオ処理を含む広範のマーケットで有用です。このアプリケーション ノートでは、このようなコンバータの 1 つについて説明します。

RGB カラー スペース

RGB (Red, Green, Blue) カラー スペースは、コンピュータ グラフィック分野で広く使用されています。赤、緑、青は加法混色の三原色であり、これらを混ぜ合わせることによって適切な色が作成され、3 次元のデカルト座標系で表されます ([参考文献 2])。

表 2 に、RGB の各値を 100% で表す 100% 飽和色バーを示します。これらは、ビデオ テストで頻繁に使用される信号です [参考文献 1]。RGB は、赤、緑、青を組み合わせて任意の色を作成し、表現できるため、コンピュータ グラフィック用に最も普及しているカラー スペースです。したがって、RGB カラー スペースを選択することによって、システムのアーキテクチャおよびデザインが簡潔化します。また、RGB カラー スペースは長年に渡り広く普及しているため、このカラー スペースを使用するよう設計されたシステムは、非常に多くの既存ソフトウェア アルゴリズムを活用できます。

ただし、RGB は実在する対象の画像処理には必ずしも有効ではありません。RGB カラー キューブ内で任意の色を作成するには、3 つのコンポーネントすべてを同一バンド幅にする必要があります。さらに、RGB カラー スペースでの画像処理は、通常、最も効率的な方法ではありません。たとえば、あるピクセルの強度または色を修正するには、3 つの RGB 値すべての読み出し、修正、およびフレーム バッファへの再書き込みが必要となります。強度および色の形式で保存された画像にアクセスできるシステムでは、このような処理手順が短縮します。

表 2 : 100% RGB カラー バー

	公称範囲	白	黄色	シアン	緑	マゼンタ	赤	青	黒
R	0 ~ 255	255	255	0	0	255	255	0	0
G	0 ~ 255	255	255	255	255	0	0	0	0
B	0 ~ 255	255	0	255	0	255	0	255	0

R'G'B' カラー スペース

RGB がコンピュータ グラフィックに理想的なカラー スペースであるのに対し、8 ビット リニアライト コーディングでは表示画像の品質が低下します [参考文献 2]。高品質で表示させるには、各コンポーネントが 12 または 14 ビットである必要があります。非リニアの光度に対するヒトの視覚反応を模倣した非リニア コーディングを使用することによって、限られたビット数を最大限に活用します。ビデオ JPEG、MPEG、コンピュータ、デジタル静止画像などでは、RGB 信号に非リニア転送機能を使用して、非リニア方式でコード化されたガンマ補正済みコンポーネント (R'G'B' と表示) を使用します。709 [参考文献 4] または sRGB と同様の変換機能を使用する 10 ビットの非リニア コーディングによって、高品質画像が実現できます。

YUV カラー スペース

YUV カラー スペースは、PAL、NTSC、および SECAM カラー ビデオ/TV 規格で使用されます。以前の白黒システムでは輝度 (Y) 情報のみを使用していましたが、これに、通常の白黒映像が白黒レシーバで表示されるよう彩度情報 (U および V) を追加したのが YUV カラー スペースです。

YCrCb (または YCbCr) カラー空間

YCbCr カラー空間は、デジタルコンポーネントとしてのビデオの世界規格を策定中に ITU-R BT.601 [参考文献 3] の一部として開発されました。YCbCr は、YUV カラー空間を部分的に使用するオフセットバージョンです。Y は 16 ~ 235 の公称範囲で定義され、Cb および Cr は 16 ~ 240 の公称範囲で定義されます。4:4:4、4:2:2、4:2:0 など複数の YCbCr サンプルングフォーマットがあります。

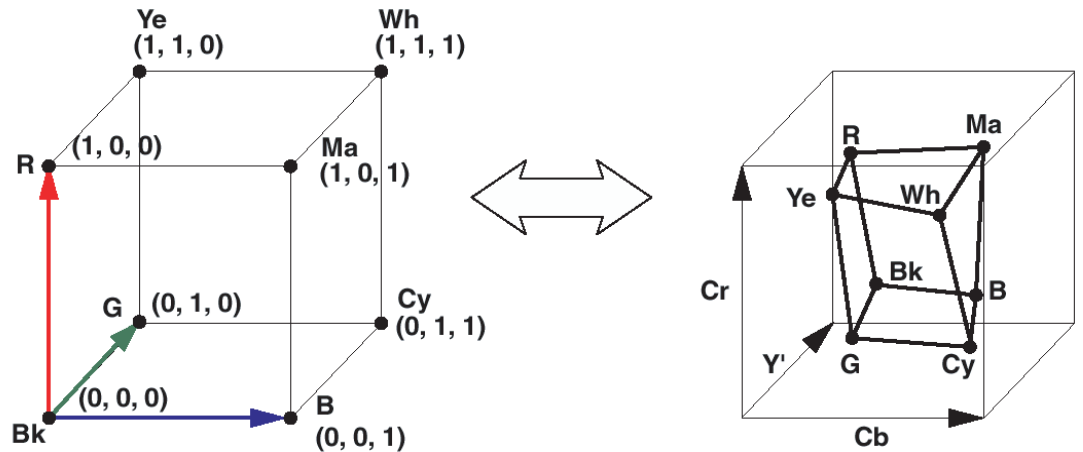


図 2：RGB および YCrCb カラー表示

変換式

微分変換式

輝度 (Y、または灰色の値) コンポーネントを作成するため、ヒトの目が赤色、緑色、青色の強度をどのように感知するかを計測する生体実験が実施されました。この実験結果から、係数 CA および CB の最適値は次の式から求められることがわかりました。

$$Y = CA * R + (1 - CA - CB) * G + CB * B \tag{等式 1}$$

CA および CB の実際の値は、規格によってわずかに異なります。

RGB カラー空間から輝度および色度 (カラーコンポーネントの差分) への変換は、次の式で表すことができます。

$$\begin{bmatrix} Y \\ B - Y \\ R - Y \end{bmatrix} = \begin{bmatrix} CA & 1 - CA - CB & CB \\ -CA & CA + CB - 1 & 1 - CB \\ 1 - CA & CA + CB - 1 & -CB \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{等式 2}$$

係数 CA、CB、および 1-CA-CB は 0 から 1 の間で選択され、これによって、Y の範囲は RGB_{min} と RGB_{max} 間に制限されます。

ただし、B-Y の最大値と最小値は次の式で求められる値になります。

$$\min_{B-Y} = RGB_{min} - (CA * RGB_{max} + (1 - CA - CB) * RGB_{max} + CB * RGB_{min}) = -(1 - CB) * (RGB_{max} - RGB_{min})$$

$$\max_{B-Y} = RGB_{max} - (CA * RGB_{min} + (1 - CA - CB) * RGB_{min} + CB * RGB_{max}) = (1 - CB) * (RGB_{max} - RGB_{min})$$

したがって、B-Y の範囲は 2(1-CB) (RGB_{max} - RGB_{min}) となります。

同様に、R-Y の最大値と最小値は次の式で求められる値になります。

$$\min_{R-Y} = RGB_{min} - (CA * RGB_{min} + (1 - CA - CB) * RGB_{max} + CB * RGB_{max}) = -(1 - CA) * (RGB_{max} - RGB_{min})$$

$$\max_{R-Y} = RGB_{max} - (CA * RGB_{max} + (1 - CA - CB) * RGB_{min} + CB * RGB_{min}) = (1 - CA) * (RGB_{max} - RGB_{min})$$

したがって、R-Y の範囲は 2(1-CA) (RGB_{max} - RGB_{min}) となります。

実際のインプリメンテーションでは、輝度および色度コンポーネントの範囲は同一である必要があります。これには、輝度コンポーネント (B-Y および R-Y) の正規化 (圧縮とオフセット補正) が可能か、あるいは色度範囲の上下の値がクリップ可能かの 2 つの方法があります。

クリッピングおよびダイナミック範囲の圧縮と再量子化によって情報が失われますが、その影響には差異があります。入力 (RGB) 範囲の差を活用するには、異なる規格に対するクリッピングと正規化にそれぞれのトレードオフがあります。

YCrCb から RGB への変換アプリケーションは、次の標準公式に当てはまる変換のみをサポートします。

$$\begin{bmatrix} Y \\ C_R \\ C_B \end{bmatrix} = \begin{bmatrix} CA & 1-CA-CB & CB \\ CC(-CA) & CC(CA+CB-1) & CC(1-CB) \\ CD(1-CA) & CD(CA+CB-1) & CD(-CB) \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} O_Y \\ O_C \\ O_C \end{bmatrix} \tag{等式 3}$$

CC および CD の値によって、B-Y および R-Y のダイナミック範囲の縮小が可能であり、等式から求められる C_B および C_R のオフセット圧縮が容易になるよう O_Y および O_C に制限が与えられます。この係数 CC および CD によって、色度コンポーネント (C_B および C_R) のダイナミック範囲を狭めることができます。RGB 値を [0..1]range, $O_C = 0.5$ に制約する場合、次のようになります。

$$CC = \frac{1}{2(1-CB)} \quad CD = \frac{1}{2(1-CA)} \tag{等式 4}$$

これにより、演算でのオーバーフローおよびアンダーフローを回避します。

逆に、等式 3 に示した行列式を変形することによって、YCrCb から RGB カラー スペースへの変換式は、次のように定義できます。

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1/CC & 0 \\ 1 & \frac{-CA}{CC(1-CA-CB)} & \frac{-CB}{CD(1-CA-CB)} \\ 1 & 0 & 1/(CD) \end{bmatrix} \begin{bmatrix} Y - O_Y \\ C_R - O_C \\ C_B - O_C \end{bmatrix} \tag{等式 5}$$

デザイン パラメータ値の割り当て

次に、使用頻度の高い規格に対する特定のパラメータ値を示します。通常、COEF および OFFSET 以外のパラメータには表に記載されている値をそのまま設定できます。VHDL パラメータに対応する整数値の割り当て前に、次の等式を使用して、これらのパラメータの算出、スケール、丸め込みを行う必要があります。

$$ACOEF = \left[2^{CWIDTH-2} \frac{1}{CC} \right]$$

$$BCOEF = \left[2^{CWIDTH-2} \frac{-CA}{CC(1-CA-CB)} \right]$$

$$CCOEF = \left[2^{CWIDTH-2} \frac{-CB}{CD(1-CA-CB)} \right]$$

$$DCOEF = \left[2^{CWIDTH-2} \frac{-1}{CD} \right]$$

[] は、近似整数への丸め込みを示します。

OFFSET パラメータの算出には、次の変数を使用します。

$$\text{ROUNDING_CONST} = 2^{\text{MWIDTH}-\text{OWIDTH}-2} \cdot \text{FAMILY_HAS_MAC}$$

$$\text{SCALE_M} = 2^{\text{IWIDTH}+\text{CWIDTH}-\text{MWIDTH}+\text{FAMILY_HAS_MAC}}$$

$$\text{ROFFSET} = \lceil \text{ROUNDING_CONST} - (\text{ACOEFF} \cdot \text{COFFSET} + \text{YOFFSET}) / \text{SCALE_M} \rceil$$

$$\text{GOFFSET} = \lceil \text{ROUNDING_CONST} - ((\text{BCOEFF} + \text{CCOEFF}) \cdot \text{COFFSET} + \text{YOFFSET}) / \text{SCALE_M} \rceil$$

$$\text{BOFFSET} = \lceil \text{ROUNDING_CONST} - (\text{DCOEFF} \cdot \text{COFFSET} + \text{YOFFSET}) / \text{SCALE_M} \rceil$$

VHDL パラメータ値の算出は負荷であると思われるかもしれませんが、これによって、4つの乗算器を使用するソリューションを構成するための値ではなく、通常は COEFF および OFFSET 値を指定する任意の規格からの値を置換できます (図 3)。次の表は、[参考文献 1]、[参考文献 2]、[参考文献 3]、および [参考文献 4] に記載のデータに基づいています。

ITU 601 (SD) および 709 - 1125/60 (NTSC)

表 3 : 601 および NTSC 709 規格のパラメータ値

係数/ パラメータ	範囲		
	16-240	16-235	0-255
CA	0.299		0.2568
CB	0.114		0.0979
CC	0.564	0.5772	
CD	0.713	0.7295	0.5910
YOFFSET	$2^{\text{OWIDTH}-4}$		
COFFSET	$2^{\text{OWIDTH}-1}$		
HAS_CLIP	1		0
HAS_CLAMP	1		0
YMAX	$240 * 2^{\text{OWIDTH}-8}$	$235 * 2^{\text{OWIDTH}-8}$	$255 * 2^{\text{OWIDTH}-8}$
CMAX	$240 * 2^{\text{OWIDTH}-8}$	$235 * 2^{\text{OWIDTH}-8}$	$255 * 2^{\text{OWIDTH}-8}$
YMIN	$16 * 2^{\text{OWIDTH}-8}$		0
CMIN	$16 * 2^{\text{OWIDTH}-8}$		0

ITU 709 (HD) 1250/50 (PAL) 規格

表 4 : PAL 709 規格のパラメータ値

係数/ パラメータ	入力範囲		
	16-240	16-235	0-255
CA	0.2126		0.1819
CB	0.0722		0.0618
CC	0.5389	0.5512	
CD	0.6350	0.6495	0.6495
YOFFSET	$2^{\text{OWIDTH}-4}$		
COFFSET	$2^{\text{OWIDTH}-1}$		

表 4 : PAL 709 規格のパラメータ値 (続き)

係数/ パラメータ	入力範囲		
	16-240	16-235	0-255
HAS_CLIP	1		0
HAS_CLAMP	1		0
YMAX	240*2 ^{OWIDTH-8}	235*2 ^{OWIDTH-8}	255*2 ^{OWIDTH-8}
CMAX	240*2 ^{OWIDTH-8}	235*2 ^{OWIDTH-8}	255*2 ^{OWIDTH-8}
YMIN	16*2 ^{OWIDTH-8}		0
CMIN	16*2 ^{OWIDTH-8}		0

YUV 規格

表 5 : YUV 規格のパラメータ値

係数/ パラメータ	値
CA	0.299
CB	0.114
CC	0.492111
CD	0.877283
YOFFSET	2 ^{OWIDTH-4}
COFFSET	2 ^{OWIDTH-1}
HAS_CLIP	1
HAS_CLAMP	1
YMAX	240*2 ^{OWIDTH-8}
CMAX	240*2 ^{OWIDTH-8}
YMIN	16*2 ^{OWIDTH-8}
CMIN	16*2 ^{OWIDTH-8}

カラー スペース変換式

$$R = ACOEF * Cr + Y + ROFFSET$$

$$G = BCOEF * Cb + CCOEF * Cr + Y + GOFFSET$$

$$B = DCOEF * Cb + Y + BOFFSET$$

これらの変換式は、[図 3](#) に示す回路図に直接マップできます。

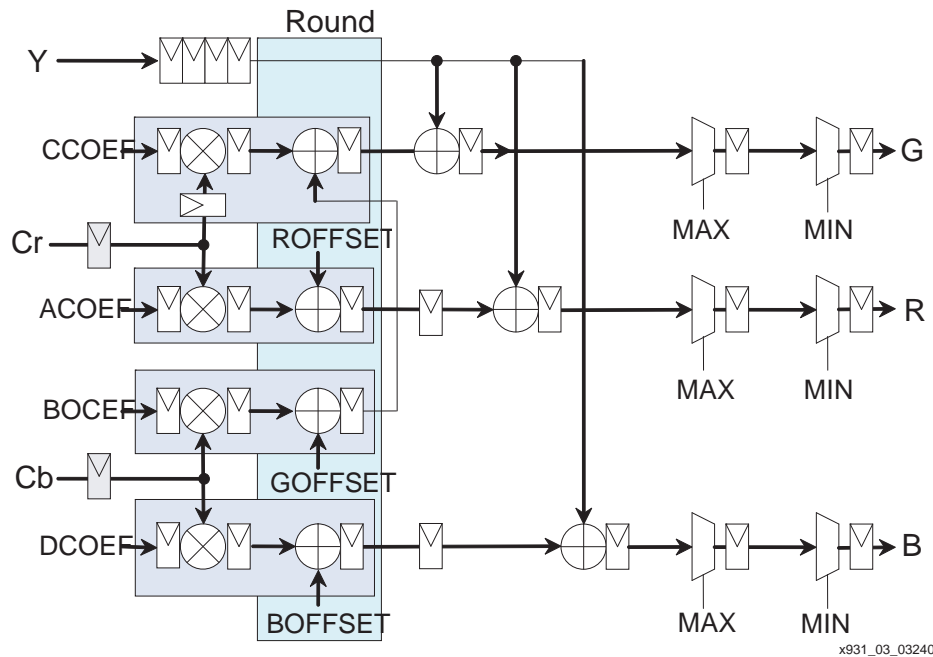


図 3：アプリケーション回路図

エラー解析

灰色のボックスはロジックブロックを示し、ターゲットデバイスで DSP ブロックが使用可能な場合は、常に DSP ブロックを使用してインプリメントされます。Virtex-4 デバイスをターゲットとしている場合は FAMILY_HAS_MAC を 1 に設定します。FABRIC_ADDDS = 1 と設定すると、デザイン内のほか演算コンポーネントも DSP ブロックにマップされます。

DSP fundamentals [参考文献 5] に基づく次の解析は、IWIDTH ビット RGB 入力データ、OWIDTH ビット幅 YCrCb 出力データ、および CWIDTH ビットを係数精度に使用することを前提した場合の YCrCb から RGB 変換の MSE (平均二乗誤差) の算出を示します。[参考文献 5] の場合も固定係数値および入力と出力は、同様の結果となります。

丸め込み/量子化を考慮すると、図 3 に示すストラクチャは次の等式をインプリメントします。

$$R = \lceil Cr \cdot ACOEF + ROFFSET \rceil_{OWIDTH} + Y \tag{等式 6}$$

$$G = \lceil Cb \cdot BCOEF + Cr \cdot CCOEF + ROFFSET \rceil_{OWIDTH} + Y \tag{等式 7}$$

$$B = \lceil Cb \cdot DCOEF + BOFFSET \rceil_{OWIDTH} + Y \tag{等式 8}$$

$\lceil \cdot \rceil_k$ は、k ビットへの丸め込みを示します。DSP48 ブロックを内蔵しないデバイスをターゲットとして、デザインをインプリメントする場合、乗算器の出力は、加算される前に MWIDTH の値に切り捨てられ、ファブリックベースの加算器のサイズおよびキャリーチェーンの長さが縮小します。この切り捨てに起因するノイズは、ワード長がさらに OWIDTH まで短縮される段階で挿入される量子化ノイズ以下に抑制できます。6.02 MWIDTH [dB] で SQNR を概算する場合、MWIDTH を増加させることによって、丸め込みによるノイズを削減できることがわかります。ただし、MWIDTH は、デザインにおけるリソース使用率および最大キャリーチェーン長、つまりは最大スピードに影響を与えます。

MWIDTH > 18 とすると、デザインで使用する専用乗算器数が大幅に増加します。したがって、IWIDTH+4 ~ 18 の範囲で最適な MWIDTH 値では、リソース数は大幅に増加しませんが、挿入された量子化ノイズが無視できるほど小さくなります (最大でも入力ノイズより 20dB 低い)。

デザインの出力では、量子化ノイズおよびクリッピング ノイズが発生する可能性があります。データが出力で **OWIDTH** ビットに丸められると、量子化ノイズが挿入されます。**R**、**G**、**B** 値が出力フォーマット [0..1] 許容範囲より大きくなる (オーバーフロー) または小さくなる (アンダーフロー) 可能性のある **COEF** および **OFFSET** 値が選択されると、クリッピング ノイズが発生する場合があります。

これらのノイズ ソースの解析前に、入力の **SQNR** (信号 - 量子化ノイズ比) を確認します。量子化エラーが均等に分散していることを前提とすると、**Y** 入力チャネルの **SQNR** は等式 9 になります。

$$\text{SQNR}_{\text{YCrCb}} = 10 \log \frac{P_x}{P_N} = 10 \log \frac{\int_{\text{YMIN}}^{\text{YMAX}} (x - m(x))^2 dx}{\frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} x^2 dx} \quad \text{等式 9}$$

$\text{LSB} = 2^{-\text{INBITS}}$ (**INBITS** は入力 (**RGB**) 精度) に置換すると、 SQNR_{RGB} は入力ダイナミック範囲の関数となります。この関数を簡潔に示すため、**YCrCb** in the $x = [0,1]$ 範囲での **YCrCb** を使用する代わりに **INBITS** = 8 として、次の範囲の $\text{SQNR}_{\text{YCrCb}}$ を算出すると、次のようになります。

YCrCb 値が **(0, 255)** 範囲の場合 :

$$\text{SQNR}_{\text{YCrCb}} = 10 \log \frac{\frac{1}{255} \int_0^{255} x^2 dx}{\int_{-1/2}^{1/2} x^2 dx} = 10 \log \frac{\frac{1}{3 \cdot 255} [255^3]}{\frac{1}{12}} = 54.15 \text{dB} \quad \text{等式 10}$$

YCrCb 値が **(16, 240)** 範囲の場合 :

$$\text{SQNR}_{\text{YCrCb}} = 10 \log \frac{\frac{1}{224} \int_{16}^{240} x^2 dx}{\int_{-1/2}^{1/2} x^2 dx} = 53.92 \text{dB} \quad \text{等式 11}$$

RGB 値が **(16, 235)** 範囲の場合 :

$$\text{SQNR}_{\text{YCrCb}} = 10 \log \frac{\frac{1}{219} \int_{16}^{235} x^2 dx}{\int_{-1/2}^{1/2} x^2 dx} = 53.47 \text{dB} \quad \text{等式 12}$$

出力量子化ノイズ

量子化回路を適切な AWGN ソースに置き換えることによって、図 3 に基づく信号フローの量的ノイズ解析のグラフが得られますが、クリッピング ノイズも考慮する最終的なノイズを求める式の微分は非常に複雑であり、このアプリケーション ノートでは取り扱いません。

表 6 に、標準的なパラメータの組み合わせによるノイズ計測結果を示します [参考文献 3]。

表 6 : ITU-REC 601 (SD) での入力および出力 SNR 計測結果 [dB]

SNR	IWIDTH = OWIDTH = 8 ビット	IWIDTH = OWIDTH = 10 ビット	入力範囲
SNR _{YCrCb} (入力)	53.5	65.6	[0..255] (8 ビット) または [0..1023] (10 ビット)
SNR _R	47.7	59.7	
SNR _G	42.4	54.4	
SNR _B	45.7	57.9	
SNR _{YCrCb} (入力)	53.4	65.3	[16..240] (8 ビット) または [64..960] (10 ビット)
SNR _R	47.5	59.4	
SNR _G	42.1	54.2	
SNR _B	45.5	57.5	
SNR _{YCrCb} (入力)	53.3	65.2	[16..235] (8 ビット) または [64..920] (10 ビット)
SNR _R	47.2	59.3	
SNR _G	42.0	54.1	
SNR _B	45.3	57.3	

出力クリッピング ノイズ

入力 (R, G, B) およびパラメータ (COEF と OFFSET) の組み合わせによっては、出力可能な範囲 [0..1) を越える値が出力される場合があります。出力値が最大値より大きくなる (オーバーフロー) または最小値より小さくなる (アンダーフロー) 可能性があり、この結果、出力値が不適切となります。クリッピング ロジックのないデザイン (HAS_CLIPPING=0) でオーバーフローまたはアンダーフローが発生した場合、バイナリ値がラップアラウンドし、出力に多くのノイズが発生します。HAS_CLIPPING=1 の場合は、出力値がサチュレートし、ノイズ発生は抑制されます (図 4)。

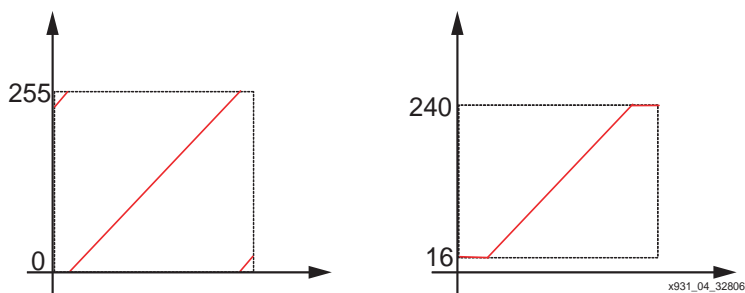


図 4 : ラップアラウンドとサチュレート

同様に、HAS_CLAMPING=1 とすると、デザインにクランピング ロジックが含まれます。クリッピングおよびクランピングを使用することにより、デザインで使用するスライス数が約 6OWIDTH スライス分増加します。

ターゲットとした規格で、出力値が ITU-R BT.601-5 [参考文献 3] などのバイナリ値ではなく、定義済みの範囲に制限されている際には、規格仕様に基づいて YMAX および YMIN 値 (輝度に対する制約) と CMAX および CMIN 値 (色度に対する制約) を設定することによって、クリッピングおよびクランピング ロジックを使用すると、制約のある出力値があらかじめ定義された範囲内で使用できます。

パフォーマンス、 レイテンシ、 リソース概算

オプションをデフォルト設定とする ISE 8.1 ツールを使用してテストを実施し、評価データを取得しました。XST では、最適化ゴールををエリアに設定しました。また、MAP の最適化およびタイミングドリブン パッキング オプションはオンとしています。

Virtex-4 テストには、-10 スピード グレード、FF668 パッケージの XC4VSX35 デバイスを使用しました (表 7)。

表 7 : Virtex-4 デバイスのパフォーマンスおよびリソース概算

最大動作周波数	288.76MHz
スライス フリップフロップ数	141
4 入力 LUT 数	122
使用スライス数	97
DSP48 数	4

Spartan™-3 テストには、-4 スピード グレード、FG320 パッケージの XC3S1000 デバイスを使用しました (表 8)。

表 8 : Spartan-3 デバイスのパフォーマンスおよびリソース概算

最大動作周波数	158.17MHz
スライス フリップフロップ数	206
4 入力 LUT 数	122
使用スライス数	131
MULT18x18 数	4

モジュールの正確なレイテンシを計算するには、次のファンクションが使用可能です。

```
YCrCb2RGB_LATENCY(FAMILY_HAS_MAC, FABRIC_ADDS, HAS_CLIP, HAS_CLAMP:
integer)
return integer
```

このファンクションは、color_space_pkg.vhd に含まれ、YCrCb2RGB モジュールのレイテンシを返します。

System Generator の トークン

System Generator を使用して開発された複雑なシステムへの YCrCb - RGB 変換デザインの統合を容易にするため、VHDL コードをカプセル化するトークンが提供されています。

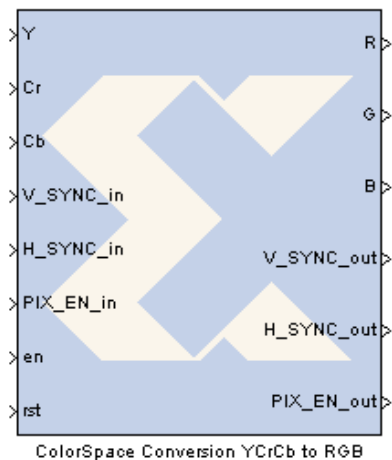


図 5 : System Generator のトークン

トークンをダブルクリックすると、アクティブな GUI 上で System Generator インスタンスのパラメータが設定できます。YCrCb から RGB の変換で「デザイン パラメータ値の割り当て」に記載の汎用規格を使用する場合に必要な設定は、GUI の [Basic] タブ (図 6) で規格および必要な入力/出力精度を選択するのみです。

System Generator トークンを適切に実行させるには、次のファイルを MATLAB の作業ディレクトリにコピーする必要があります。

- Xil_YCrCb2RGB_config.m
- ycrb2rgb_action.m
- ycrb2rgb_enablement.m
- Xil_YCrCb2RGB_GUI.xml

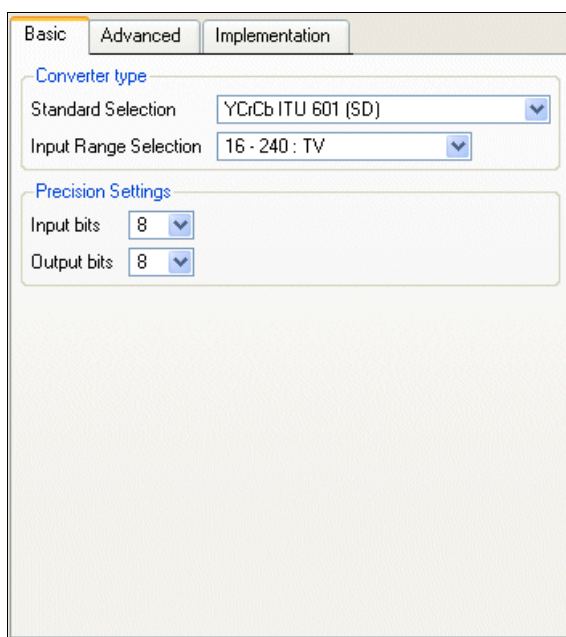


図 6 : System Generator GUI の [Basic] タブ

ドロップダウン リストに表示されない、カスタム設計のコンバータまたは規格を使用する場合、[Standard Selection] で [Custom] を選択すると [Advanced] タブ (図 7) を使用した設定が可能となります。[Conversion matrix] では、「変換式」に記載の CA、CB、CC、および CD のパラメータを浮動小数点形式 (範囲 [0..1]) で直接入力できます。[Advanced] タブは、RGB から YCrCb への変換用リファレンス デザインのものと非常に類似するよう設計されています。これらの GUI に同一のパラメータを入力することで、演算ノイズ以外が一致する変換ペアが得られ、これはオリジナルのピクセルデータを返します。

オフセット圧縮、クリッピング、およびクランピング設定は、「パラメータ設定」に記載の VHDL パラメータと同様です。

[Advanced] タブに表示される値は、[Custom] を選択する前の規格に対応する値に初期化されていますが、これらのオプション設定後は、定義済み規格とカスタム規格間で選択変更を繰り返しても、GUI にはユーザーが定義した値が表示されます。

The screenshot shows the [Advanced] tab of a configuration window. It is divided into three sections: 'Conversion matrix', 'Offset Compensation', and 'Clipping and Clamping'.
 - 'Conversion matrix':
 $Y = R * 0.299 + G * 0.587 + B * 0.114$
 $Cr = (R - Y) * 0.713$
 $Cb = (B - Y) * 0.564$
 - 'Offset Compensation':
 Luma Offset value: 16
 Chroma Offset value: 128
 - 'Clipping and Clamping':
 Outputs clipped
 RGB Maximum value: 240
 Outputs clamped
 RGB Minimum value: 16

図 7 : [Advanced] タブ

3 つめのタブは [Implementation] タブであり (図 8)、デザインのレイアウトを決定するオプションが設定できます。ここで設定可能な係数ビットおよび乗算器入力ビットは、共にデザインのリソース使用率とノイズに影響を与えます。定義済みの値を使用すると、デザインでの DSP48 使用数を増加させることなく、最大の SNR を実現できます。また、乗算器の入力ビット幅を小さくすると、使用スライス数が削減されます。

DSP48 を内蔵するデバイスをターゲットとしてデザインをインプリメントする場合、その直後に DSP48 を使用する 4 つの乗算器と 3 つの加算器、およびパイプラインレジスタがインプリメントされます。DSP48 が使用可能である場合に、DSP48 にインプリメントされるロジックは、図 3 で背景色が灰色となっています。

FABRIC_ADDS VHDL パラメータに対応する [Use fabric for adders] チェックボックスは、その他の加算器を (スライスベースのロジックを使用して) ロジック ファブリックにインプリメントするか、DSP48 にインプリメントするかを制御します。

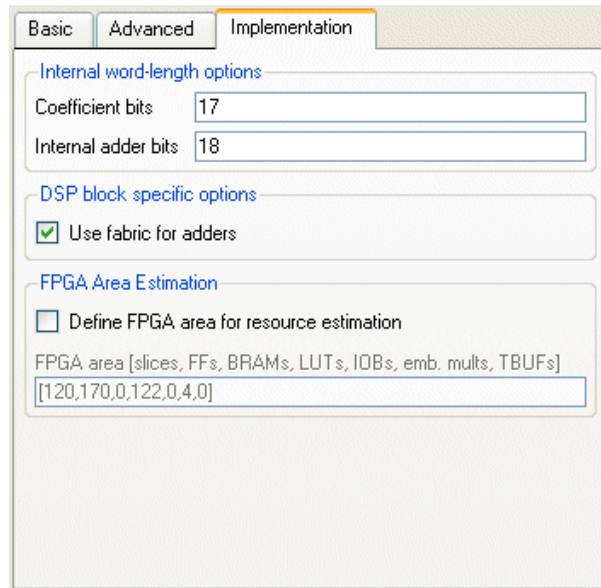


図 8 : [Implementation] タブ

System Generator の テストベンチ

YCrCb から RGB 変換のサブシステムのプロトタイプリング、テスト、および検証を実施するために、リファレンス デザインには System Generator のテストベンチが含まれます。テストベンチ ファイルは、/sysgen/testbench ディレクトリ以下にあります。テストベンチを開くには、MATLAB ディレクトリを /sysgen/testbench に変更し、Xil_YCrCb2RGB_tb.mdl を取り込んでください (図 9)。

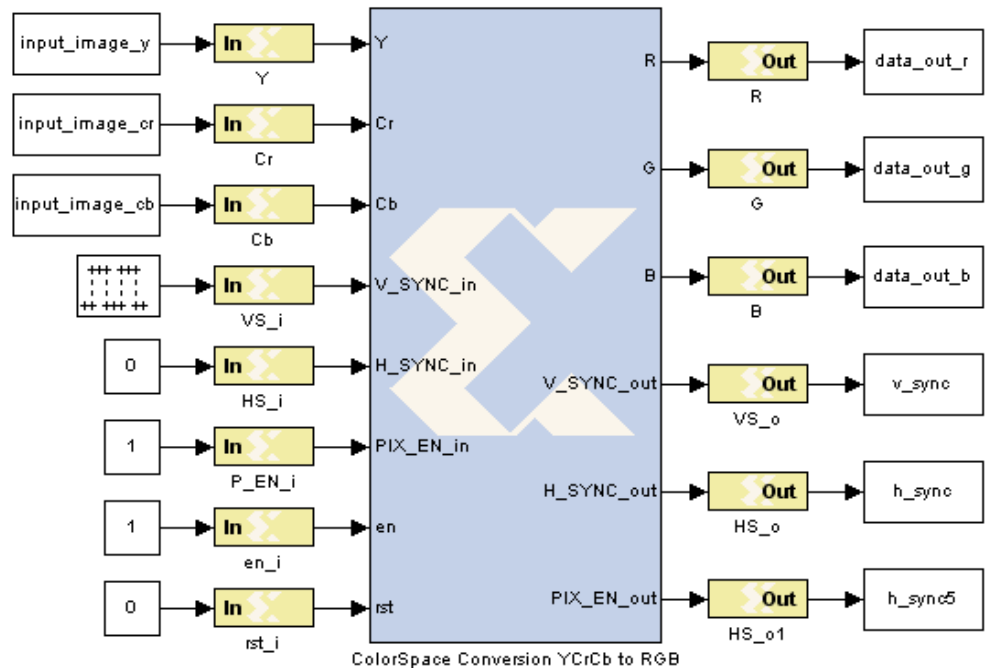


図 9 : System Generator のテストベンチ

モデルの初期化中、Xil_YCrCb2RGB_preload_mdl.m が実行され、カラー バーで使用されるステイミュラス ワークプレイス変数、input_image、input_image_r、input_image_g、input_image_b が初期化されます (図 10)。また、既にロードされたファンクションは、ファンクション double_ycrCb2rgb を使用して出力 matlab_y、matlab_cr、matlab_cb のリファレンスとなります。

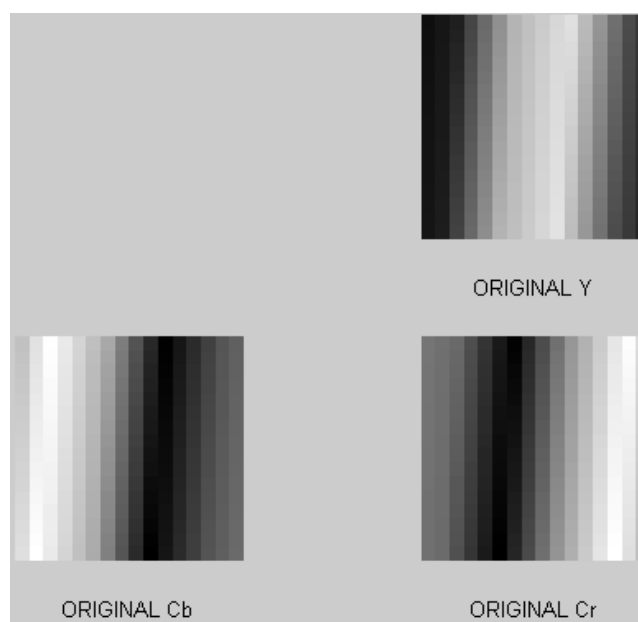


図 10 : R、G、B スティミュラス

テストベンチの実行

メモ: テストベンチは、デフォルト パラメータ設定のトークンを実行するよう設定されています。パラメータ **IWIDTH** または **OWIDTH** を変更すると、**R**、**G**、および **B** の **Gateway In** モジュールでビット数とバイナリ ポイント設定を変更する必要があります。

テストベンチは、**ISE** シミュレータ **ISIM**、外部シミュレータ **ModelSim**、あるいはハードウェア コシミュレーションを使用して実行可能です。ハードウェア コシミュレーションの詳細は、**System Generator** のマニュアルを参照してください。デフォルトでは、テストベンチは **ModelSim** を使用し、シミュレーション完了後も **ModelSim** のシミュレーション ウィンドウが開かれたままとなるオプションが使用されます。

1. シミュレータを変更するには、**YCrCb2RGB** トークンで右クリックし、表示メニューから **[Mask]** → **[Look]** を選択してください。
2. **Colorspace** トークンでダブルクリックし、表示されるブロックの **[Properties]** ダイアログボックスでシミュレーション モードを選択します。**ModelSim** のみに対するオプションは、**ModelSim** トークンをダブルクリックして設定できます。重要な機能として、シミュレーション開始前にマクロ ファイルを読み込むことができ、この機能によって、シミュレーション実行中に追加 (内部) **VHDL** 信号の表示が可能となります。**ModelSim** は、ブラック ボックス デザインのデバッグ機能に優れたツールです。
3. マクロ ファイル名を指定するには、**ModelSim** ブロックの **[Properties]** ダイアログボックスで **[Advanced]** タブを選択し、**[Script to run after vsim]** に **.do** ファイル名を入力します。デフォルトでは、**wave_add_rgb.do** が取り込まれ、重要な信号のいくつかは波形ウィンドウに表示されます。
4. シミュレーション開始アイコン (-) をクリックして、シミュレーションを実行します。シミュレーション完了後、ファンクション **Xil_YCrCb2RGB_post_proc** が呼び出され、検証用に **VHDL** 出力が **GUI** に表示されます (図 11)。

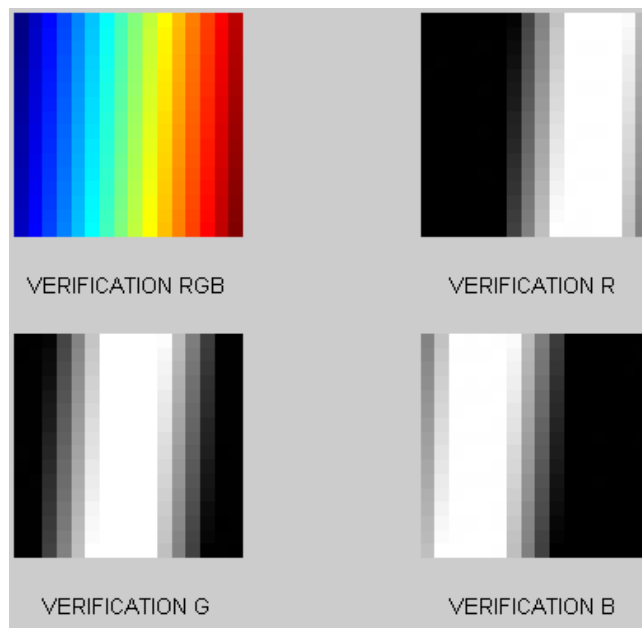


図 11 : Y, Cb, Cr Output

リファレンス デザインファイル

プロセス後のファンクションには、VHDL 出力と倍精度 MATLAB モデル間の主なエラー統計算出用のテンプレートが含まれます。また、VHDL 結果の bit-true 検証を容易にする固定小数点 MATLAB モデル (Xil_YCrCb2RGB_fi_model.m) がバンドルされています。

リファレンス デザイン ファイルは、次のダイリンクスのウェブ サイトからダウンロードできます。
<http://www.xilinx.co.jp/bvdocs/appnotes/xapp931.zip>

参考資料

1. Keith Jack, Video Demystified, 4th Edition, ISBN 0-7506-7822-4, pp 15-19.
2. Charles Poynton, Digital Video and HDTV, ISBN 1-55860-792-7, pp 302-321.
3. Recommendation ITU-R BT.601-5 standard definition: <http://www.itu.int>.
4. Recommendation ITU-R BT.709-5 standard definition: <http://www.itu.int>.
5. Gary Sullivan, "Approximate theoretical analysis of RGB to YCbCr to RGB conversion error," in Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6.)

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2006/05/09	1.0	初版リリース