



XAPP932 (v1.0) 2006 年 5 月 9 日

色差信号リサンプラー

著者 : Clive Walker

はじめに

このアプリケーション ノートでは、多様な色差フォーマット変換に使用する 6 つの標準回路のインプリメンテーションについて説明し、ジェネリック RTL VHDL コードを含むリファレンス デザインを提供します。コードは、すべてのザイリンクス デバイス ファミリに対応できるように記述されており、System Generator ユーザー向けに、HDL コードをカプセル化する System Generator™ トークンもあります。また、System Generator テストベンチも提供しており、出力結果の確認や MATLAB/Simulink 環境で既存モデルを検証できます。

概要

本来の RGB フォーマットは、各 R、G、および B 情報のバンド幅が同一です。このフォーマットで情報の格納および伝送が可能です。ただし、このフォーマットが最も効率的であるとは限りません。実際、人間の目は、輝度 (ルミナンス) には敏感ですが、色差 (クロミナンス) には鈍感です。カラー スペース変換ブロックを使用すると、RGB から YCbCr (Y は輝度信号、Cr と Cb は色差信号) へカラー スペースを変換できます。Y のバンド幅は最大で保持されますが、Cr と Cb はサブサンプル (間引き) されます。この変換は単純ですが、映像圧縮の初段階としては非常に大きな効果があり、ストレージおよび伝送コストを抑えることができます。「サブサンプルされる画像フォーマットおよびサポートされる伝送フォーマット」では、サブサンプルされる色差フォーマットについて説明します。このアプリケーション ノートでは、標準的な色差フォーマットの変換に焦点を当てて説明します。

RGB と YCrCb 間の変換の詳細は、System Generator の Color-Space Converter ブロックの説明を参照してください。また、アプリケーション ノート [XAPP930 : 『カラー スペースの変換 : RGB から YCrCb』](#) および [XAPP931 : 『カラー スペースの変換 : YCrCb から RGB』](#) も参照してください。

シンボル

図 1 で示すシンボルは、図 2、図 4、および図 6 で使用されます。

- = Luma only pixel
- × = Luma only pixel (Cr and Cb)
- ⊗ = Cosited Luma and Chroma pixel

x932_01_032906

図 1 : シンボル

標準的なインターフェイス信号

ブロック内でのデータの入力と出力の動作を説明するには、まずインターフェイス信号の定義が必要です。

表 1 に、CRS ブロックで使用される信号を示します。

表 1：インターフェイス信号

信号名	説明
luma_in	輝度チャンネルと色差チャンネルの遅延を同等にするブロックで使用される。
cr_in/out	Cr 色差信号。4:4:4 インターフェイスでのみ使用。このフォーマットの詳細は、「4:4:4」を参照。
cb_in/out	Cb 色差信号。4:4:4 インターフェイスでのみ使用。このフォーマットの詳細は、「4:4:4」を参照。
chroma_in/out	最大バンド幅で Cr と Cb がインターリーブ (交互に配置) する。4:2:2 または 4:2:0 インターフェイスで使用。慣例では Cr が常に最初。このフォーマットの詳細は、「4:2:2」および「4:2:0 (MPEG2)」を参照。
luma_out	luma_din 信号が遅延した信号。オプション使用。
vs_in	垂直同期入力信号。垂直ブランキング期間中は必ず High になる。この信号の立ち上がりエッジをフィールドベースのリセットとして使用。vs_out を生成中は最短でも 1 H 周期が必要。
hs_in	水平同期入力。水平ラインがアクティブ周期中は必ず High になる (例 : 1080i で 1920 サイクル)。
din_valid	この信号は、フィルタされるラインの hs_in 信号と必ず同じになる。ただし、画像エッジ (輪郭) がフィルタリングによって入力データを複製する場合は、ユーザーが din_valid を拡張して偽輪郭を排除できる。
hs_out	水平同期信号。フィルタのレイテンシと同等のクロック サイクル数を水平方向に遅延してタイミングを計る。
vs_out	垂直同期信号。フィルタのレイテンシと同等のクロック サイクル数および H 周期数を垂直と水平に遅延してタイミングを計る。
dout_valid	din_valid 信号が遅延した信号。luma_out および chroma_out/cr_out/cb_out の有効なデータを示す。
saturate/sat_type	<p>saturate が High の場合、フィルタリング後の加算チェーンでのオーバーフローまたはアンダーフローを示す。飽和状態は、sat_type 出力で示す。</p> <p>Saturate = 1 の場合 sat_type = 0 は、オーバーフロー ($\text{full_precision_result} > 2(\text{full_precision_result_width}) - 1$) sat_type = 1 は、アンダーフロー ($\text{full_precision_result} < 0$)</p> <p>この機能により、必要に応じてユーザーがクリップ/クランプ構造をインプリメントできる。</p>
chroma_dout_valid	4:2:0 指定の dout_valid フラグ。オプション使用。オルタネート (交互に配置する) ライン出力のみ有効。

共通機能

- すべてのブロックは、すべてのデバイスファミリでサポートされる。
- すべてのブロックには、色差がフィルタされるパスで生じる遅延と同等の遅延を輝度パスに与える機能がある。輝度出力を接続しない場合は、その他のブロックは最小化され、スライス使用率が少なく、乗算器ブロックは使用しない。
- すべてのブロックは、符号なし輝度および色差入力データを想定する。
- 垂直フィルタリングを要するすべてのブロックは、dout_valid 信号出力に遅延を与えると同時に、h_sync と v_sync 信号にも遅延を与える。

サブサンプルされる画像フォーマットおよびサポートされる伝送フォーマット

4:4:4

このフォーマットは、本来 RGB で使用されるレートですが、変換ブロックについて説明する目的として、オリジナル RGB と同じサンプル レートを使用して YCbCr を定義します。実際、YCbCr の 4:4:4 フォーマットは、RGB でも 4:4:4 です。このレートに従って映像の取得や表示を行います。Cr および

Cb チャンネルは、輝度成分と同等の最大バンド幅でサンプリングされるため、すべてのピクセルには輝度と色差の両方が混在します。図 2 を参照してください。



図 2 : 4:4:4 画像表示

このフォーマットの伝送は単純です。Y、Cr、および Cb には、それぞれ個別のバスがあります。入力フォーマットの場合、din_valid 信号とデータは同時入力する必要があります。dout_valid 信号は din_valid 信号と同一になるはずですが、数クロック サイクルまたは H 周期分の遅延が生じる可能性があります。この遅延値は、変換の特性およびフィルタ タップ数 (水平および垂直) に依存します。図 3 を参照してください。

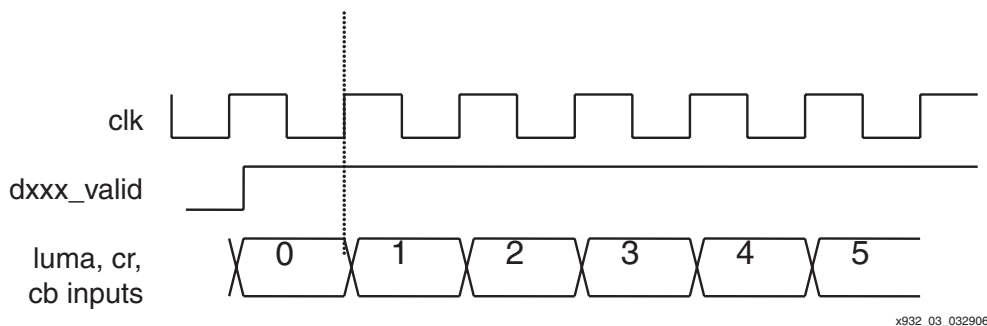


図 3 : 4:4:4 画像表示

4:2:2

このフォーマットの場合、水平ラインで色差信号がサブサンプルされます。輝度が 2 回サンプリングされると色差 (Cr と Cb のペア) が 1 回サンプリングされます。サブサンプルされた色差は、輝度と同じピクセルに混在します。図 4 を参照してください。



図 4 : 4:2:2 画像表示

このフォーマットの場合、1 つのバス上に Cr と Cb がインターリーブ (最初は Cr) され、このバスを最大サンプルレートで実行します。図 5 を参照してください。

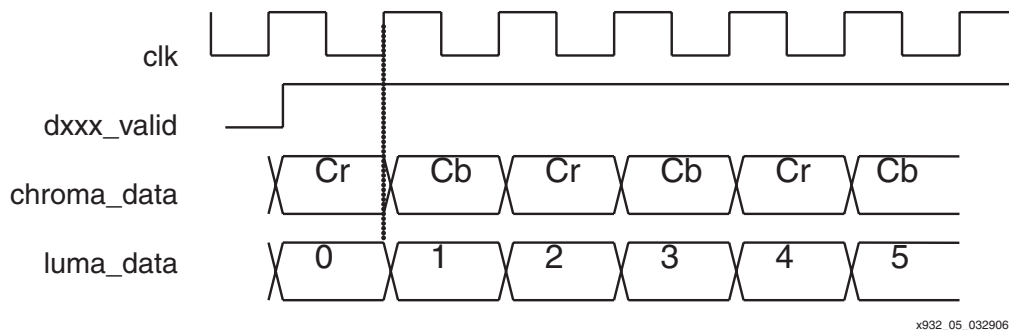


図 5 : 4:2:2 伝送フォーマット

4:2:0 (MPEG2)

このフォーマットは主に PEG2 エンコーディングで使用され、水平ラインと垂直ラインに色差がサブサンプルされます。このフォーマットの特徴は、色差ピクセルと輝度ピクセルが混在しないことです。実際、このフォーマットを形成するには、垂直フィルタリングを使用して、またオルタネートライン (1 ライン置き) に色差成分が配置されます。これらの値はインターポレート (内挿補間) されています。水平方向では、輝度サンプルが 2 つあると、その 1 つに色差サンプルがあります。図 6 を参照してください。

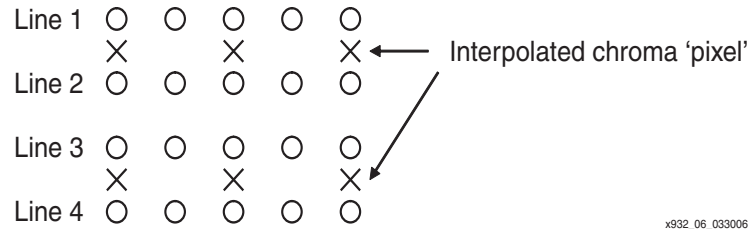


図 6 : 4:2:0 画像表示

この伝送フォーマットは、オルタネートラインのみ有効であることを除いては図 4 に示す 4:2:2 フォーマットと同一です (図 6 参照)。この論理は入力および出力の両方に適用します。出力を検証する信号として Dout_valid 信号があります。輝度ラインの順番が 0、1、2、3、4、5、... (最上ラインは 0) の場合、出力される有効な色差は、0、2、4、6、... となり、4:2:2 フォーマットの場合は Cr と Cb がインターリーブ (交互に配置) します。色差検証信号 (図 7 参照) は、オプション出力ですが実際に高い頻度で使用されています。

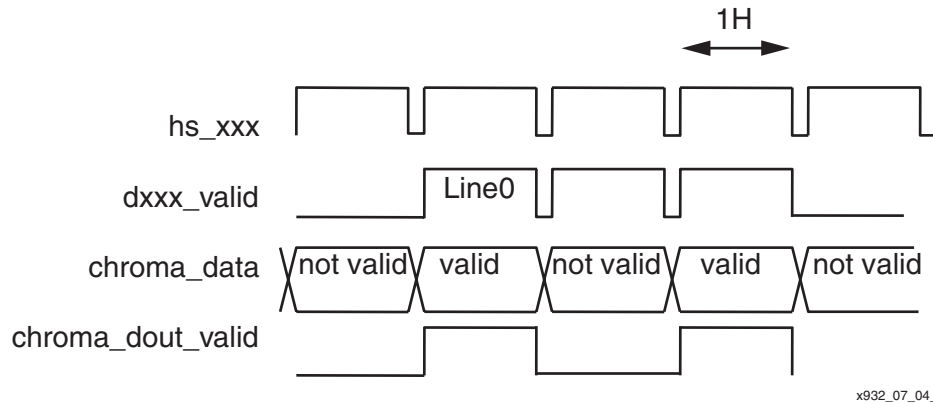


図 7 : 4:2:0 の伝送フォーマット

同様に、4:2:0 入力の場合、din_valid が最初に High になると、色差入力のデータが有効になり (vs_in は非アクティブ)、ライン 0 として認識されます。ただし、4:2:0 入力では、入力で 3 番目の輝度ラインが有効になるまで、2 番目の垂直サンプリングは有効になりません。

変換ブロック

現在、変換回路ブロックは 6 種類あります。これらのブロックでは、FIR フィルタを使用して変換します。水平方向のみフィルタリングするブロック、垂直方向のみフィルタリングするブロック、また水平および垂直の両方向をフィルタリングするブロックがあります。表 2 に、デフォルトのフィルタ内容などの詳細を示します。

表 2: フィルタのまとめ

変換	フィルタの特性	デフォルト FIR サイズ	メモ
444 から 422 へ	水平ダウン変換	3 水平方向タップ	
444 から 420 へ	水平および垂直ダウン変換	2 垂直方向タップ X 3 水平方向タップ	
422 から 444 へ	水平アップ変換	2 水平方向タップ	2 位相構造 - 位相 1 が入力を複製
422 から 420 へ	垂直ダウン変換	4 V タップ	
420 から 444 へ	水平および垂直アップ変換	2 水平方向タップ X 2 垂直方向タップ	2 位相構造
420 から 422 へ	垂直アップ変換	2 水平方向タップ X 2 垂直方向 タップ	2 位相構造 - 位相 1 は位相 2 の反転

パラメータ

パラメータ値は各ブロックで異なります。すべてのブロックは RTL VHDL でコード記述されており、パラメータ化されています。ユーザーの基準に応じてパラメータ値を設定できます。System Generator を伝送機能として使用する場合は、MATLAB で自動的に算出されるパラメータがあります。自動設定されないパラメータは直接指定する必要があります。

複製 (Replicate)

このオプションは、アップ変換で使用できます。

- 422 から 444 へ変換
- 420 から 444 へ変換
- 420 から 422 へ変換

Replicate オプションは、必要に応じて垂直ドメインおよび水平ドメインの両方に対して適用できます。使用した場合は、フィルタを使用しないアップ変換になります。その代わりに、前の入力サンプルが複製されます。結果として、ブロック合成でシステムのフィルタ ハードウェア数を最小限に抑えることができます。

Pixel_drop

このオプションは、ダウン変換に使用できます。

- 444 から 422 へ変換 (水平フィルタのみ)

pixel_drop オプションを使用した場合、フィルタを使用しないダウン変換になります。いくつかのサンプルのみ直接出力へ伝送され、その他のサンプルはすべて落とします。これはピクセル単位でのみ行われます。結果として、ブロック合成でシステムのフィルタ ハードウェア数を最小限に抑えることができます。

Line_drop

このオプションは、ダウン変換に使用できます。

- 422 から 420 へ変換 (垂直フィルタのみ)

line_drop オプションを使用した場合、フィルタを使用しないダウン変換になります。いくつかのサンプルのみ直接出力へ伝送され、その他のサンプルはすべて落とします。これはライン単位でのみ行われます。結果として、ブロック合成でシステムのフィルタ ハードウェア数を最小限に抑えることができます。

Sample_drop

このオプションは、ダウン変換に使用できます。

- 444 から 420 へ変換 (水平および垂直フィルタ)

sample_drop オプションを使用した場合、フィルタを使用しないダウン変換になります。いくつかのサンプルのみ直接出力へ伝送され、その他のサンプルはすべて落とします。これはライン単位およびピクセル単位の両方で行われます。結果として、ブロック合成でシステムのフィルタ ハードウェア数を最小限に抑えることができます。

係数

すべてのブロックに、正または負の整数パラメータとして、係数を与える必要があります。デフォルトでは、ザイリンクスが判断した適正値が設定されています。

Num_h_taps/Num_v_taps

すべてのブロックにフィルタ サイズを設定する必要があります。System Generator の場合、これらのパラメータは自動的に計算されます。MATLAB では、係数パラメータに入力した係数の合計値が計算されます。

Data_width

色差および輝度ストリームの入力/出力の幅を示すパラメータです。

メモ: フィルタリングで生じる出力データ ビット幅の拡張はありません。出力は、切り捨ての前に高精度ドメインで出力 LSB の半数値を加算してラウンド (丸め込み) されます。

System Generator の場合、入力データ幅は Xilinx Input Gateway で入力したビット幅から自動的に算出されます。

係数幅

最適な結果を得るためには、係数の合計値を正確に $2^{(\text{coef_width} - 1)}$ とすることを推奨します。合計値が $2^{(\text{coef_width} - 1)}$ より少ない場合は、ダイナミック レンジの損失となります。また、単一係数は、 $2^{(\text{coef_width} - 1)} - 1$ を超えないようにしてください。

アクティブ入力ラインのサンプル

このオプションは、垂直フィルタを要するすべての変換で使用できます。

- 420 から 444 へ変換
- 420 から 422 へ変換
- 422 から 420 へ変換
- 444 から 420 へ変換

インプリメンテーション時にライン バッファのサイズを設定する必要があります。

パディング/ボーダー値

この値は、有効なデータが `din_valid` に現れる前後に FIR ストラクチャへ与えられます。この値を使用したデザインは、画像エッジ (輪郭) で入力サンプルを複製しません。したがって、画像エッジでは有効データの前後でこのパディング値がフィルタに与えられます。ザイリンクスでは、`din_valid` の拡張に従って、コア外部で入力サンプルを反復させることを推奨します。これを行わない場合は、固定パディング値により画像エッジにノイズが生じます。これは垂直および水平ラインの両方で生じる現象です。

各ブロックの詳細

422 から 444 へ変換するブロック

このブロックは (図 8)、2:1 水平アップ変換を示します。

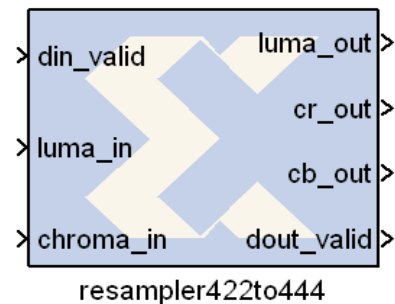


図 8 : 422 から 444 へ変換するブロック

2つの位相構造を使用して変換され、2つの出力ピクセルに対して、1つの入力サンプルが対応しています。この入力サンプルを複製するだけで、最も理想的な高精細出力を得ることができます。この方法は、ハードウェアの使用率を抑えることができ、機能的には1つのタップ係数を1、その他すべての係数を0にした場合の位相と同等です。2番目の位相のデフォルト係数は、**[0.5 0.5]**です。これらの係数は2ビットの整数で表すと**[1 1]**になります。

2番目の位相の係数は、ユーザーが指定する必要があります。次のようにブロックに属する係数を与えます。

```
num_taps : integer: = 4;
coefs    : INTEGER_ARRAY: = (a, b, c, d, ...)
```

a は、フィルタ アパーチャの最右端の入力サンプルに与えられた係数です (ラスタ スキャン順では最新の係数。図 9 参照)。num_taps より多いタップ数が与えられた場合は無視されます。

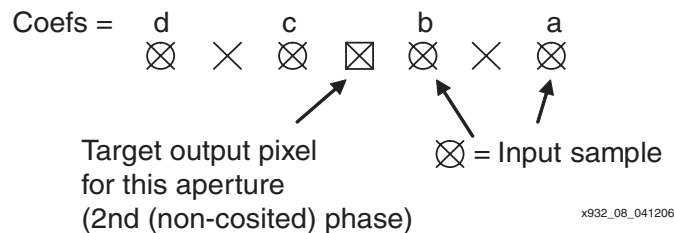


図 9 : 4:2:2 から 4:4:4 の係数コンフィギュレーション

このデフォルト フィルタで生じるレイテンシは、8 クロック サイクルです。デフォルト以外のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{レイテンシ} = (2 * \text{num_taps}) + 4$$

replicate オプションを使用する場合、レイテンシは 8 クロック サイクルになります。

444 から 422 へ変換するブロック

このブロック (図 10) は、2:1 水平ダウン変換を示します。デフォルトは、[0.25 0.5 0.25] (3 ビット形式の整数では [1 2 1]) に相当する係数を持つ 3 タップ構造を使用します。

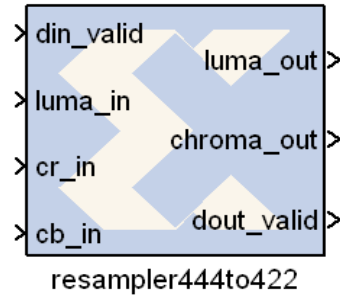


図 10 : 444 から 422 へ変換するブロック

この係数は、ユーザーが入力する必要があります。次のようにブロックに属する係数を与えます。

```
num_taps : integer := 5;
coefs    : INTEGER_ARRAY := (a, b, c, d, e . . . )
```

a は、フィルタ アパーチャの最右端の入力サンプルに与えられる係数です (ラスタ スキャン順では最新の係数。係数。図 11 参照)。num_taps より多いタップ数が与えられた場合は無視されます。

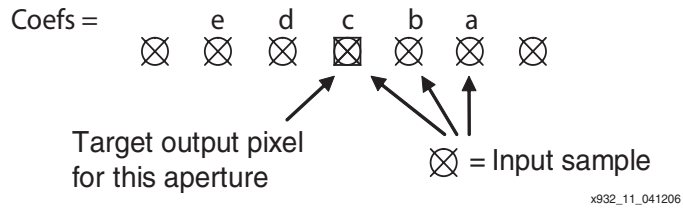


図 11 : 4:4:4 から 4:2:2 の係数コンフィギュレーション

このデフォルト フィルタで生じるレイテンシは、8 クロック サイクルです。デフォルト以外のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{レイテンシ} = (\text{num_taps} + 5)$$

pixel_drop オプションを使用する場合、レイテンシは 4 クロック サイクルになります。

420 から 422 へ変換するブロック

このブロック (図 12) は、2:1 垂直アップ変換を示します。デフォルトは、[0.25 0.75] (3 ビットの整数では [1 3]) に相当する係数を持つ 2 タップと 2 位相構造を使用します。2 番目の位相には、これらの係数の反転を適用します。

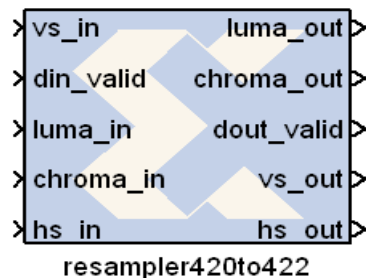


図 12 : 420 から 422 へ変換するブロック

2 位相構造の場合、ユーザーは 1 位相に対してのみ係数を与えます。次のようにブロックに属する係数を与えます。

```
num_v_taps : integer := 4;
coefs      : INTEGER_ARRAY := (a, b, c, d, . . .)
```

1 つ目の位相 (位相 0) では、**a** はフィルタ アパーチャで最も低い位置にある入力サンプルに与えられる係数です (ラスタ スキャン順では最も新しい係数。図 13 参照)。num_v_taps より多いタップ数が与えられた場合は無視されます。

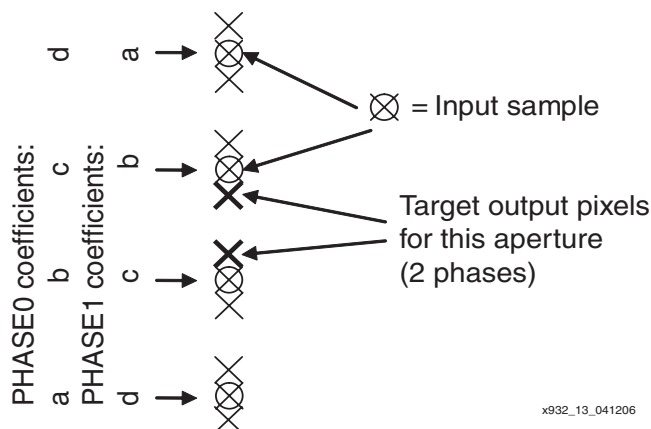


図 13 : 4:4:4 から 4:2:2 の係数コンフィギュレーション

このデフォルト フィルタで生じるレイテンシは、(1 ライン + 9 クロック サイクル) です。デフォルト以外のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{Vertical_Latency} = \text{num_v_taps} - 1$$

$$\text{Horizontal_Latency} = 9 \text{ サイクル (定数)}$$

422 から 420 へ変換するブロック

このブロック (図 14) は、2:1 垂直ダウン変換を示します。デフォルトは、**[0.125 0.375 0.375 0.125]** (4 ビットの整数では **[1 3 3 1]**) に相当する係数を持つ 4 タップ構造です。

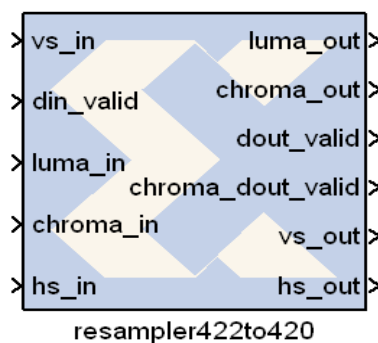


図 14 : 422 から 420 へ変換するブロック

次のようにブロックに属する係数を与えます。

```
num_v_taps: integer := 4;
coefs      : INTEGER_ARRAY := (a, b, c, d, . . .)
```

a は、フィルタ アパーチャで最も低い位置にある入力サンプルに与えられた係数です (ラスト スキャン 順では最も新しい係数。図 15 参照)。num_v_taps より多いタップ数が与えられた場合は無視されます。

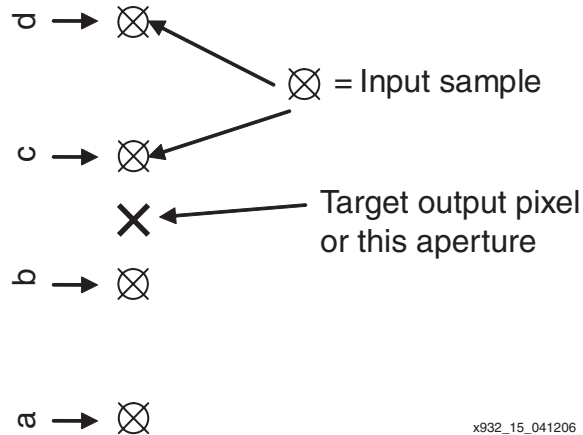


図 15 : 4:2:2 から 4:2:0 の係数コンフィギュレーション

このデフォルト フィルタで生じるレイテンシは、(2 ライン + 8 クロック サイクル) です。デフォルト以外 のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{Vertical_Latency} = \text{num_v_taps}/2$$

$$\text{Horizontal_Latency} = \text{num_v_taps}/2 + 6$$

420 から 444 へ変換するブロック

このブロック (図 16) は、二次元構造の動作です。この動作は、2 つの水平位相と 2 つの垂直位相を使用し、デフォルト コンフィギュレーションの 4 タップ構造 (2 水平タップ X 2 垂直タップ) を使用して実行します。

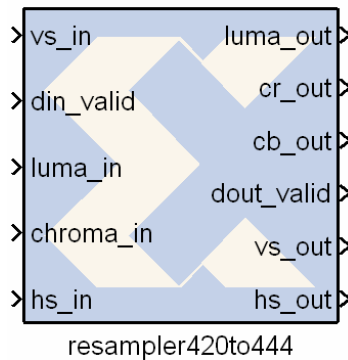


図 16 : 420 から 444 へ変換するブロック

- 1 つ目の水平位相 : 水平ラインで 2 つの出力ピクセルが 2 つの入力サンプルとアラインしているため、アラインしていないサンプルの水平ライン係数は 0 です。残りの 2 つのサンプルの係数は、420 から 422 への変換と同等です (反対の垂直位相は、[0.25 0.75] の反転)。
- 2 つ目の水平位相 : 水平ラインで 2 つの出力ピクセルは 2 つの入力ピクセルから均等距離に配置されていますが、垂直オフセットは同一です。したがって、4 つすべての入力サンプルには係数が必要です。1 つ目の垂直位相の係数は次のようになります。

$$[0.375 \ 0.375]$$

$$[0.125 \ 0.125]$$

ただし、これは 2 つ目の垂直位相では、垂直反転します。

次のようにブロックに属する係数を 2 つの水平位相に与えます。

```

num_v_taps : integer := 4;
num_h_taps : integer := 4;
coefs_hphase0 : INTEGER_ARRAY := (a, b, c, d, . . . );
coefs_hphase1 : INTEGER_ARRAY := (e, f, g, h, . . . )

```

各ジェネリックの長さは、 $\text{num_h_taps} * \text{num_v_taps}$ になります。垂直位相 0 の最下位ライン上の入力サンプルおよび垂直位相 1 の最上ラインにある入力サンプルに、各水平位相ジェネリックの最初の num_h_taps 係数が使用されます。

垂直位相 0 の場合、**a** は、水平位相 0 のフィルタ アパーチャの最下位の最右端に適用される係数です (ラスタ スキャン順では最新の係数。図 17 参照)。垂直位相 1 の場合、順序は水平反転します。

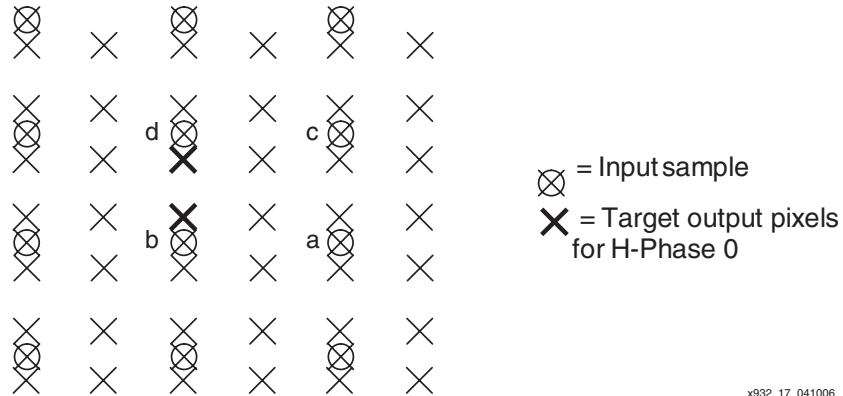


図 17 : 4:2:0 から 4:4:4 の係数コンフィギュレーション (水平位相 0)

垂直位相 0 の場合、**e** は、水平位相 1 のフィルタ アパーチャの最下位の最右端に適用される係数です (ラスタ スキャン順では最新の係数。図 18 参照)。垂直位相 1 の場合、順序は反転します。

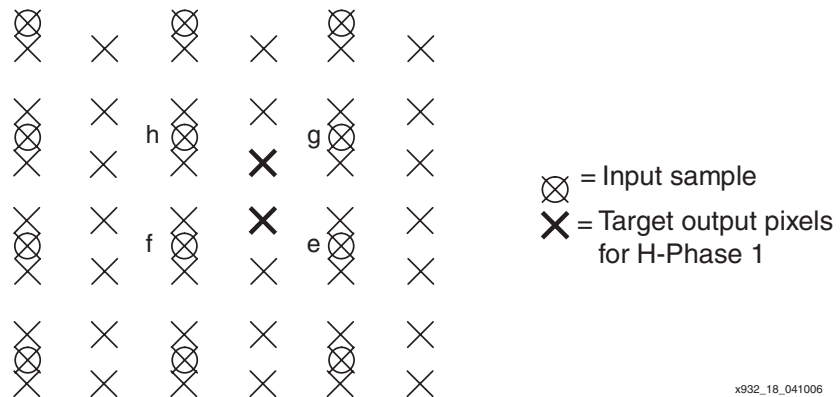


図 18 : 4:4:4 から 4:2:0 の係数コンフィギュレーション (水平位相 1)

このデフォルト フィルタで生じるレイテンシは、(1 ライン + 14 クロック サイクル) です。デフォルト以外のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{Vertical_Latency} = \text{num_v_taps}/2$$

$$\text{Horizontal_Latency} = 2 * \text{num_h_taps} + 10$$

444 から 420 へ変換するブロック

このブロック (図 19) は、垂直および水平方向で 2 倍のダウン変換を行う二次元構造の動作です。デフォルトは、合計 6 タップ (3 水平タップ X 2 垂直タップ) です。このフィルタは、理想的な水平ラインと垂

直ラインが対称的であり、デフォルト係数は **[0.0625 0.375 0.0625]** です。5 ビット形式の整数では、**[1 6 1]** になり、2 つの垂直ラインに繰り返されます。なお、ユーザーが垂直と水平にそれぞれ係数を設定することも可能です。

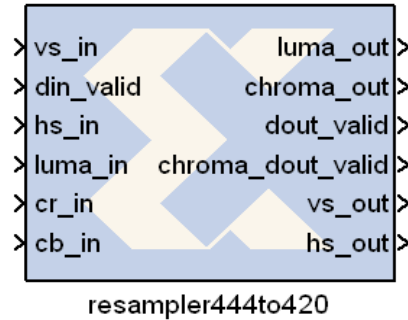


図 19 : 444 から 420 へ変換するブロック

次のようにブロックに属する係数を 2 つの位相に与えます。

```
num_v_taps : integer := 3;
num_h_taps : integer := 2;
coefs      : INTEGER_ARRAY := (a, b, c, d, e, f, ...);
```

a は、フィルタ アパーチャで最も低い位置にある入力サンプルに与えられる係数です (ラスタ スキャン順では最も新しい係数。図 20 参照)。

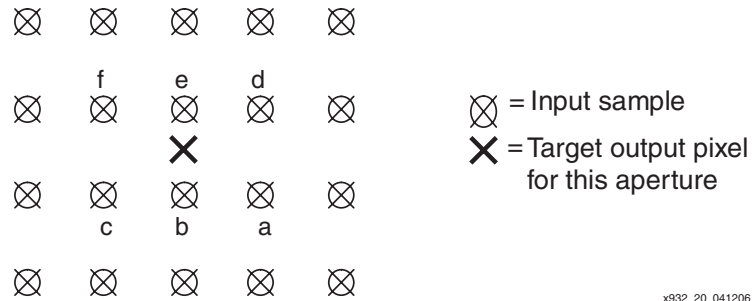


図 20 : 4:4:4 から 4:2:0 の係数コンフィギュレーション

このデフォルト フィルタで生じるレイテンシは、(1 ライン + 11 クロック サイクル) です。デフォルト以外のフィルタの場合、次の式でレイテンシを算出できます。

$$\text{Vertical_Latency} = \text{num_v_taps}/2$$

$$\text{Horizontal_Latency} = \text{num_h_taps} + \text{num_v_taps} + 4$$

System Generator テストベンチ

RGB から YCrCb へのサブシステムの試作品作成、テスト、および 検証を容易にするため、リファレンス デザインと共に system generator のテストベンチを提供しています。テストベンチ ファイルは、chroma_resampler/testbench ディレクトリにあります。

これらのファイルでテストする場合は、次のソフトウェアをインストールする必要があります。

- MATLAB R14 SP3
- ModelSim® SE 6.1a
- System Generator 8.1

テストベンチは各変換用に 1 つずつあり、合計 6 つあります。各テストベンチには関連する MATLAB モデルのファイル (ファイル名 : `convertxxxtoyyy_model.mdl`) があります。これらのテストベンチを使用してテストする場合は、次に手順に従ってください。

1. MATLAB 環境で、`/chroma_resampler/testbench` ディレクトリに変更 (CD) します。
2. MATLAB コマンド プロンプトで `pre_proc.m` を実行します (`pre_proc` を入力して `<ENTER>` キーを押す)。これにより、次が実行されます。
 - a. `reference.mdl` を実行してソース ファイル (`.png`) から画像を取得する。
 - b. ソース データを RGB から 4:4:4 の YCrCb へ変換する。
 - c. Simulink モデルを使用して 4:4:4 から 4:2:2 および 4:2:0 (MPEG2) へ変換する。
 - d. Simulink モデルを使用して 4:2:0 から 4:2:2 (MPEG2) および 4:4:4 へ変換する。
 - e. Simulink モデルを使用して 4:2:2 (MPEG2) から 4:2:0 および 4:4:4 へ変換する。
 - f. すべての変換用に、二次元スティミュラス データから一次元ベクタ データを作成する。
 - g. すべての変換用に、デフォルトのフィルタ コンフィギュレーションを設定する。

このようにして、すべての変換で使用できるテストおよび検証用のスティミュラスとゴールデン ベクタ データが生成されます。

3. Simulink の play ボタン (▶) を押して、適切な `convertxxxtoyyy_model.mdl` ファイルを実行します。ModelSim が起動します。

これにより、モデルからいくつかの ID ベクタ ファイルが出力から生成されます。例を次に示します。`luma_out_444to422.mat`。

4. 適切なプロセス後のファイルを実行します (`post_proc_xxxtoyyy` と入力)。これにより、次が実行されます。
 - a. 上記の `.mat` ファイルが読み込まれる。このファイルの中にある一次元データが二次元データに変換される。
 - b. この二次元データは、すでに作成されたゴールデン ベクタと比較される。このときの差分は、Cr および Cb に関してデルタ値として表示される。

このリファレンス デザインでは、リファレンス変換として使用される Simulink モデルは、ハードウェアで使用されるラウンディング (四捨五入) 方法を使用しないため、通常この値は 0 以外になります。また、係数が異なる変換もあります。4:4:4 から 4:2:2 変換、4:4:4 から 4:2:0 変換、および 4:2:2 から 4:2:0 変換の場合、`reference.mdl` ファイルで各変換に適切な Simulink パラメータを選択して `pre_proc` を再実行すると、リファレンス モデルの係数を変更できます。

4:4:4 への変換を実行する場合、`show444.mdl` ファイルを開いて実行すると結果を表示できます。

4:2:2 への変換を実行する場合、`reconstruct_422to444.mdl` ファイルを開いて実行すると結果を表示できます。

4:2:0 への変換を実行する場合、`reconstruct_420to444.mdl` ファイルを開いて実行すると結果を表示できます。

回帰テストを実行する場合は、MATLAB コマンド プロンプトから適切な `rtest_xxxtoyyy.m` ファイルを実行してください。異なる係数と条件を設定すると、上記のプロセスが実行されます。また、各テストに必要な出力ファイルのコピーを作成します。各テスト用に作成されたファイルの種類を確認するには、`rtest_xxxtoyyy.m` ファイルを確認してください。

リファレンス デザイン ファイル

次のザイリンクス ウェブ サイトから、リファレンス デザインとテストベンチ ファイルを入手できます。
<http://www.xilinx.co.jp/bvdocs/appnotes/xapp932.zip>

最上位デザイン ファイルは、convertxxxtoyyy.vhd です。コンパイルは、/XLIBディレクトリ内の共通ファイルによって異なります。imagexlib_utils.vhd および imagexlib_arch.vhd ファイルは、ローカルの作業ディレクトリでコンパイルする必要があります。

6つのすべてのデザインにシンプルな FIR フィルタが使用されています。係数は、ユーザーが設定できます。つまり、各変換の定義範囲内であれば、いかなる組み合わせでもパラメータとして設定できます。すべてのフィルタは、ユーザー フレキシビリティを保持するため常に非対称です。コードはすべての入力バンド幅に対してパラメータ化できます。出力バンド幅は常に入力バンド幅と同一です。

FIR フィルタ構造では、最終段階まで高精度が使用されます。すべてのデザインでは、ユーザー アプリケーションが許容する限り、FIR フィルタを使用するのではなく、サンプルの複製オプションまたはドロップ オプションを使用できます。この包括的なオプションにより、映像出力品質を犠牲にしてハードウェア リソース使用率を低減することができます。

完全に拡張可能なデザインは、使用するパラメータによって大きく変化し、それによってリソース使用率が変化します。

リソース使用率

表 3 は、ISE 8.1 および Synplify Pro バージョン 8.1 を使用した場合の結果を示します。これらの値は、ガイドラインとしてのみ使用でき、アーキテクチャおよびデザインによって異なります。

メモ：この表は、デフォルトのパラメータ設定を使用して作成されました。

表 3：クロマ リサンプリング ブロックのパフォーマンスおよびリソース使用率

ターゲット デバイス	XC4VFX12-10				3S200-4			
	リソース	スライス	ブロック RAM	DSP48	Fmax (MHz)	スライス	ブロック RAM	Mults
Synplify Pro 合成ツール								
422 から 444 ⁽¹⁾	49	0	0	400	49	0	0	200
444 から 422 ⁽¹⁾	58	0	0	400	58	0	0	200
420 から 422 ⁽²⁾	224	3	0	200	151	3	0	150
422 から 420 ⁽²⁾	288	5	0	200	180	5	0	125
420 から 444 ⁽²⁾	308	3	8	200	300	3	4	125
444 から 420 ⁽²⁾	277	3	0	200	220	3	0	125
XST 合成ツール								
422 から 444 ⁽¹⁾	57	0	0	300	57	0	0	200
444 から 422 ⁽¹⁾	60	0	0	200	60	0	0	150
420 から 422 ⁽²⁾	183	3	0	200	166	3	0	150
422 から 420 ⁽²⁾	218	5	0	200	201	5	0	125
420 から 444 ⁽²⁾	233	3	8	150	304	3	8	125
444 から 420 ⁽²⁾	252	3	0	250	232	3	0	125

メモ：

1. データ幅 = 8
2. データ幅 = 8、ライン長 = 1920

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	変更内容
2006/05/09	1.0	初版リリース