

【概要】

本サンプルデザインの機能及び動作を説明します。本サンプルデザインは、RGB と YUV のカラービデオ信号を相互に変換します。

リニア 8 ビットで入力されたカラービデオの各信号に対し、以下の変換処理を行い出力します。

(1) RGB から YUV への変換機能 (RGB2YUV)

$$Y = 0.25700 * R + 0.50400 * G + 0.09800 * B + 16$$

$$U = -0.14800 * R - 0.29100 * G + 0.43900 * B + 128$$

$$V = 0.43900 * R - 0.36800 * G - 0.07100 * B + 128$$

(2) YUV から RGB への変換機能 (YUV2RGB)

(原理: YUV の定義から逆変換を求めた式です)

$$R = 1.164(Y-16) + 1.596(V-128)$$

$$G = 1.164(Y-16) - 0.391(U-128) - 0.813(V-128)$$

$$B = 1.164(Y-16) + 2.018(U-128)$$

(実装: 原理式を展開し、定数項をまとめて値を求めた式です)

$$R = 1.164 * Y + 0.000 * U + 1.596 * V - 222.912$$

$$G = 1.164 * Y - 0.391 * U - 0.813 * V + 135.488$$

$$B = 1.164 * Y + 2.018 * U + 0.000 * V - 276.928$$

【ファイルデータ】

(1) RGB2YUV.zip ... zip 圧縮ファイル

RGB2YUV/

NSL/ ... NSL ソース記述フォルダ

RGB2YUV_conv.nsh ... 変換器モジュールの入出力の宣言

RGB2YUV_conv.nsl ... 変換器モジュールの動作の定義

RGB2YUV_Ymac.nsh ... 各成分の生成のための乗算と混合のモジュールの入出力の宣言

RGB2YUV_Ymac.nsl ... 各成分の生成のための乗算と混合のモジュールの動作の定義

RGB2YUV_Umac.nsh ... 以下同様

RGB2YUV_Umac.nsl

RGB2YUV_Vmac.nsh

RGB2YUV_Vmac.nsl

toplevel.nsl ... テストパターン(「テストパターン」の節で解説)

SIMMODEL/ ... ModelSim によるシミュレーションフォルダ(参考。toplevel.nsl とは内容が異なります)

Verilog/ ... NSL Core による変換出力フォルダ(変換例を収録)

RGB2YUV_conv.v

RGB2YUV_Ymac.v

RGB2YUV_Umac.v

RGB2YUV_Vmac.v

(2) YUV2RGB.zip ... zip 圧縮ファイル

(注)RGB2YUV を例に YUV2RGB も同様なデータ構成。

【デザイン】

1. モジュール名

(1) RGB2YUV_conv: RGB から YUV への変換機能

(2) YUV2RGB_conv: YUV から RGB への変換機能

2. インタフェース

(注)RGB2YUV を例に YUV2RGB モジュールも同様な構成。

入力信号

- p_reset : NSL 処理系が生成するリセット入力
- m_clock : NSL 処理系が生成するクロック入力
- R_in[8] : R 信号の入力
- G_in[8] : G 信号の入力
- B_in[8] : B 信号の入力
- VSYNCin : 同期用の垂直同期信号の入力
- HSYNCin : 同期用の水平同期信号の入力
- ConvStart(R_in, G_in, B_in) : 新しいデータの変換開始を指示する入力信号。同時に、それぞれの色信号の入力を設定する。

出力信号

- Y_out[8] : Y 信号の出力
- U_out[8] : U 信号の出力
- V_out[8] : V 信号の出力
- VSYNCout : 同期用の垂直同期信号の出力。入力に対し常に 3 クロック遅れる。
- HSYNCout : 同期用の水平同期信号の出力。遅れは同様。
- ConvFinish(Y_out, U_out, V_out) : データの変換終了を示す出力信号。同時に、それぞれの色信号が出力される。ConvStart に対し常に 4 クロック遅れる。

3. 基本動作

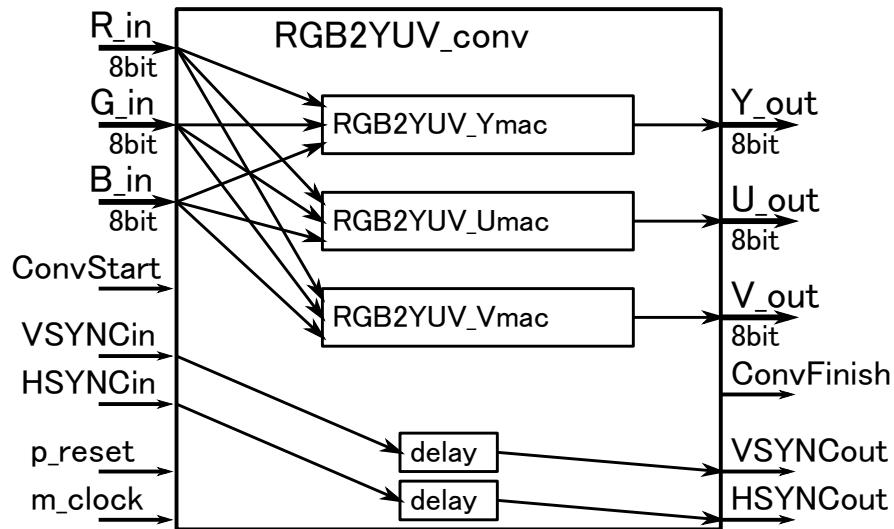
ConvStart を 1 クロック以上の間アクティブにすると同時に、色信号の各入力を設定してください。入力の状態を保持し続けてもかまいませんが、必須ではありません。

変換には複数クロックの時間がかかります。(同時を 0 クロックとして)入力から 4 クロック遅れて、ConvFinish がアクティブになり、変換された信号が出力されます(ConvFinish は、常に ConvStart の 4 クロック遅れです)。

入出力の間の同期用に、VSYNC と HSYNC の入出力があります。どちらも入力の 3 クロック遅れで出力されます。

4. ブロック図

(注)RGB2YUV を例に YUV2RGB モジュールも同様な構成。



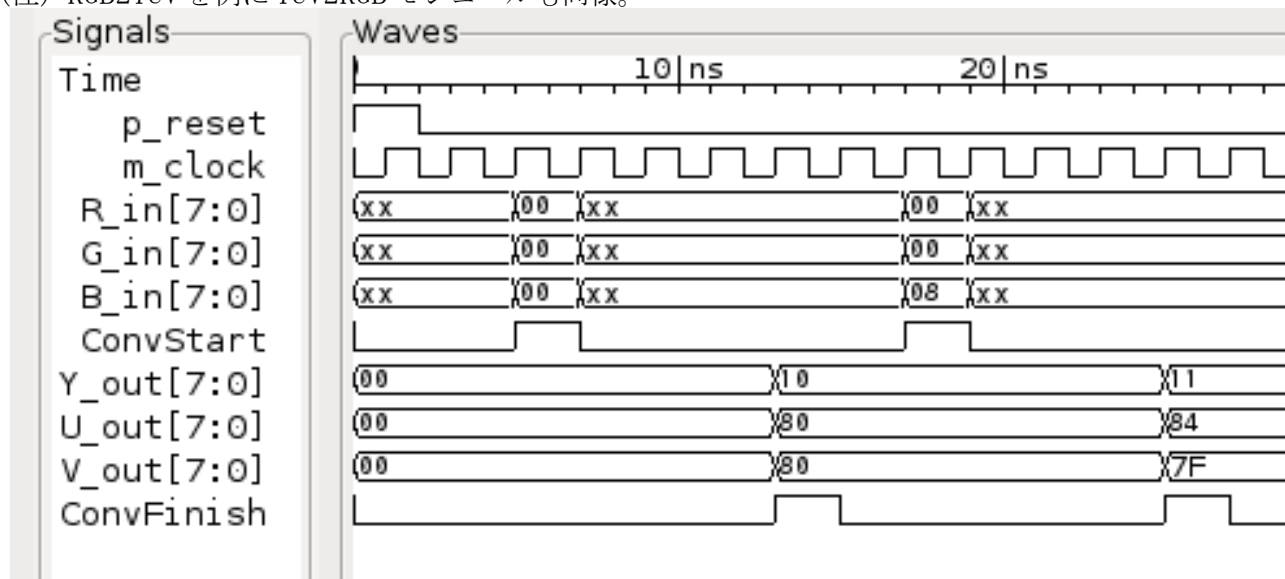
【シミュレーション】

1. テストパターン

toplevel.nsl 内にテストパターンが記述されています。各色信号に毎回異なる値を与えて、変換を実行するサンプルです。添付のタイミング図を参照してください。(SIMMODEL ディレクトリ内のものと基本的には同様ですが、toplevel.nsl ではテスト用モジュールを NSL で記述しています。また、数分以内にシミュレーションが終わるようパターンの組合せ数を調整しています)。

2. タイミング図

(注) RGB2YUV を例に YUV2RGB モジュールも同様。



【注意点】

RGB2YUV_Ymac 他各信号のモジュール（および逆変換の同等のモジュール）を、NSL Core のシミュレーションモード（-O[12] オプションを付けなかった場合のデフォルト）で Verilog に合成してシミュレーションした場合、使用方法によっては register set hazard というウォーニングが各クロックごとに出ます。内部で不定値から不定値が生成されているため、入力値が設定されていて正しく動作中の状態には問題を及ぼしませんので大丈夫です。

【改版履歴(Revision History)】

版数(Version)	日付(Date)	内容(Content)
V.1.0	2013年4月17日	初版リリース