

### 【概要】

本アプリケーションノートは、NSL 記述によるサンプルデザインの CRC 生成回路の機能及び動作に関して説明します。

本サンプルデザインは、バイトストリームを入力すると、その巡回冗長検査 (CRC) の値を算出して出力する回路です。crc16 は、CRC-16 Kermit 等と呼ばれているバリエーションの値、crc32 はイーサネットなどで使われている CRC-32 と呼ばれるバリエーションの値を算出します。

### 【ファイルデータ】

#### 1. CRC16 生成機能

CRC16.zip … zip 圧縮ファイル

CRC16/

NSL/ … NSL ソース記述フォルダ

    crc16\_nsl.nsh … crc16 モジュールの入出力の宣言

    crc16\_nsl.nsl … crc16 モジュールの動作の定義

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

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

    crc16\_nsl.v

#### 2. CRC32 生成機能

CRC32.zip … zip 圧縮ファイル

CRC32/

NSL/ … NSL ソース記述フォルダ

    crc32\_nsl.nsh … crc32 モジュールの入出力の宣言

    crc32\_nsl.nsl … crc32 モジュールの動作の定義

    toplevel.nsl … テストパターン

Verilog/ … NSL Core による変換出力フォルダ

    crc32\_nsl.v

## 【デザイン】

### 1. モジュール名

- (1) crc16\_nsl : CRC16 生成機能
- (2) crc32\_nsl : CRC32 生成機能

### 2. インタフェース

#### 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
- data\_in[8] : バイトストリームを入力します。
- shift\_do : data\_inに入力データがある時、アクティブにします。

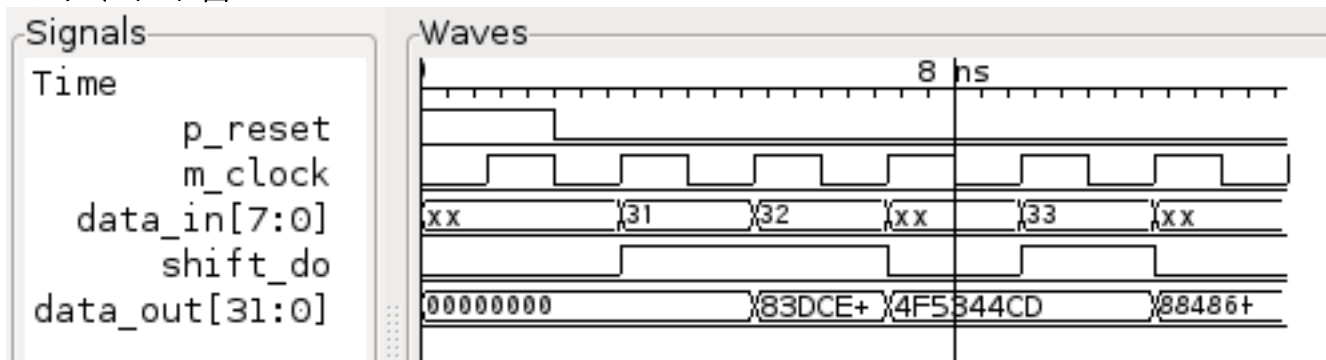
#### 出力信号

- data\_out[16 or 32] : リセット以降、直前のクロックまでに入力されたバイトストリームのCRC 値を出力。常に出力。

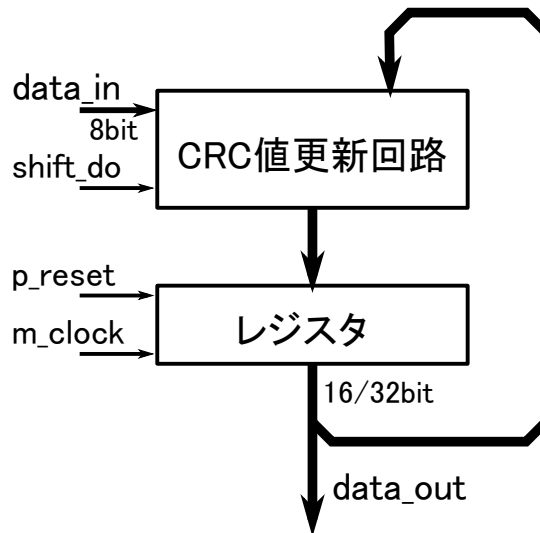
### 3. 基本動作

データは1バイトごとに入力します。data\_inに1バイトの入力データを与え、shift\_doをアクティブにしてください。shift\_doがアクティブであれば、クロック毎に入力を読み込み、出力のCRC 値が更新されます。出力は、直前クロックの入力までのCRC 値を常に出力し続けます。(data\_inのデータは、m\_clockのエッジにおいてshift\_doがアクティブの時、読み込まれてCRC 値が更新されます。) 改めて別のバイトストリームのCRC 値の計算を始める時はリセット信号をアクティブにしてください。

### 4. タイミング図



## 5. ブロック図



### 【シミュレーション】

#### 1. テストパターン

toplevel.nsl 内に動作を理解するため簡単なテストパターンが記述されています。

CRC のテスト用データとしてよく使われている文字列 “123456789” に相当する入力を与え、結果を出力します。

#### 補足

他のソフトウェアによる実装などと結果を比較した際、結果の値のバイトが上位下位逆だと思われるかもしれません。CRC-16 Kermit や CRC-32 は LSB から順に計算する方式のため、一部の実装では出力もそれに合わせ、下位バイトから先に出力するためです。この実装では出力は整数値のワードとして扱っているので、そういった実装と比較した場合、結果のバイト順が逆になっているように見えます。

### 【参照文献(Reference)】

1. 参考情報: CRC のバリエーションは <http://reveng.sourceforge.net/crc-catalogue/> に網羅的なサーベイがあります。

### 【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