

## 【はじめに】

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移動平均フィルタに関して説明します。信号処理において、複数個の入力値を受け取りそれを記憶して、一定の入力を受け取った後、それまでの入力値の合計を元に、入力の平均値を出力するのが移動平均フィルタです。

移動平均フィルタの構成法は、いくつか考えられます。ひとつめは、ストリーム中に一定の大きさの「窓」を想定し、その窓の中にある値の合計を常に維持する方式です。合計レジスタに、窓に入ってきた値を1個加算すると同時に、窓から出てゆく値を1個減算するようにします。実装には、窓の中にある個々の値を後の減算のために記憶する、シフトレジスタかリングバッファが必要でしょう。この方式は高いスループットが期待できますが、フィルタ専用の設計であり、応用の柔軟性がありません。

もうひとつの構成法は、入力を保存する必要な個数のメモリを単に用意し、平均の値が必要になった時に合計を計算する方式です。平均の値を常に必要とするのでなければ、こちらの設計のほうが単純で、他への応用も考えられます。この方式の平均生成回路を元に、フィルタを構成することも可能でしょう。本サンプルデザインは、後者の方式の平均生成回路となっています。

## 【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、平均を算出する対象の値を記憶するメモリを内蔵し、そのメモリに書き込まれた値の平均を算出して出力する回路です。入力には、その値を書き込むべきメモリアドレスを同時に与える必要がありますので、利用する回路側でカウンタを用意してください。平均値を得るには、計算開始を指示するワンショットの信号を与えます。計算開始が指示されると、平均を取る入力の個数に比例したクロック数をかけて合計の計算を行い、合計値を元に平均値を出力します。

入力数が 32 個・64 個・128 個の各サンプルデザインがあります。以下では主に 32 個を例に説明しますが、他のデザインも同様の考えで利用できます。

## 【ファイルデータ】

### 1. 32 入力の平均生成機能

average.zip … zip 圧縮ファイル  
average\_32/ … 32 個の平均を算出するサンプルデザイン  
NSL/ … NSL ソース記述フォルダ  
    average\_32.nsh … average\_32 モジュールの入出力の宣言  
    average\_32.nsl … average\_32 モジュールの動作の定義  
    average\_32.v … NSL Core による変換例 (参考)  
    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モデル (参考。上記テストパターンとは別)  
VLOG/ … Verilog HDL による同機能の実装例 (参考。32 のみ)

### 2. 64 入力の平均生成機能

average\_64/ … 64 個の平均を算出するサンプルデザイン  
(上記構成と同じ)

### 3. 128 入力の平均生成機能

average\_128/ … 128 個の平均を算出するサンプルデザイン  
(上記構成と同じ)

## 【デザイン】

### 1. モジュール名

- (1) average\_32 : 32 入力の平均生成機能
- (2) average\_64 : 64 入力の平均生成機能
- (3) average\_128 : 128 入力の平均生成機能

### 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
- DataIn[16] : 平均を計算する入力値を入力します。
- DataAdrs[5/6/7] : 入力を保存するメモリのアドレスを入力します。入力数により幅が異なります。
- HoldReq( DataAdrs, DataIn ) : 内蔵するメモリのアドレス DataAdrs に、DataIn の値を書き込むことを指示する入力です。
- Calc() : 平均の計算の開始を指示するワンショット入力です。

#### 出力信号

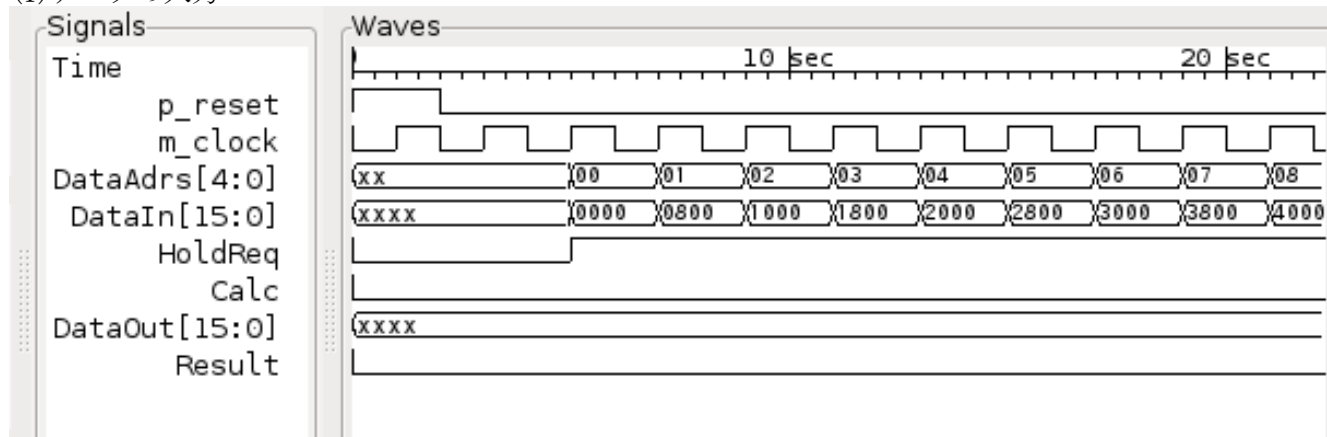
- DataOut[16] : 平均の値の出力です。
- Result( DataOut ) : 平均の計算が終了し、結果を DataOut に出力することを示すワンショットの出力です。

### 3. 基本動作

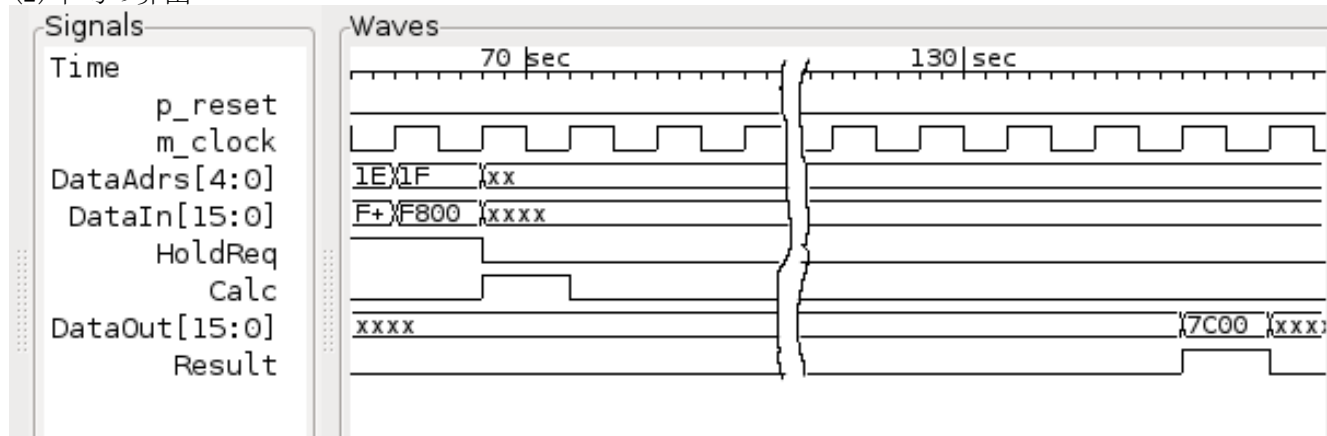
データの入力は16ビットの1ワードごと、DataAdrsに与えるアドレスをインクリメントさせながら、入力DataInに与え、HoldReqをアクティブにします。平均を得るには、Calcをワンショットでアクティブにします。計算時間は平均を取る対象のデータ数によって変わります。計算が終了すると、Resultがワンショットでアクティブになり、DataOutに平均の値が出力されます。平均の算出において除算は単純にシフトしておこなっているため、小数点以下の端数は単に切り捨てられた値になります。値がまだ入力されていない内蔵メモリの値は、リセット直後は0に初期化されています。

### 4. タイミング図

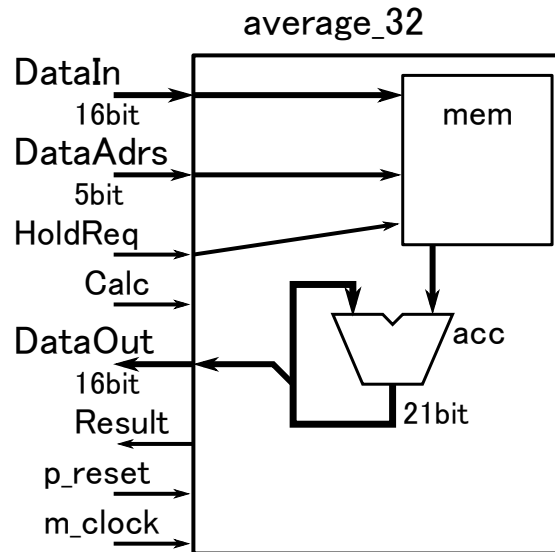
#### (1) データの入力



#### (2) 平均の算出



## 5. ブロック図



### 【シミュレーション】

#### 1. テストパターン

toplevel.nsl 内に動作理解のための簡単なテストパターンが記述されています。

入力に 0x0000~0xf800 の値を与え、平均を算出して出力します。その後、0xffff を追加して与え、新たな平均値を出力します。

### 【参照文献 (Reference)】

1. 三上直樹『改訂新版 C/C++によるデジタル信号処理入門』(CQ 出版) ISBN978-4-7898-3098-0

### 【改版履歴 (Revision History)】

版数 (Version)	日付 (Date)	内容 (Content)
V.1.0	2013 年 4 月 25 日	初版リリース