



XAPP497 (v1.0) 2011 年 8 月 15 日

USR_ACCESS を使用するビットストリーム認識

著者 : Arthur Yang

概要

ビットストリームの追跡は、多くのアプリケーションで有用です。たとえば、特定デザインのバージョンを識別したり、デザインを最適化するために複数回実行したインプリメンテーションを追跡する場合などのより複雑な状況ではシリアルナンバーを付けるために使用します。デザインにはあらゆる種類のスタティックコードを組み込むことができるため、バージョンデータのサイズやフォーマットには無限の可能性がありますが、ただし、デザインの一部あるいは全体を再コンパイルする必要があり、その作業は退屈で制限があります。Virtex®-5、Virtex-6、およびすべての 7 シリーズ FPGA に含まれる USR_ACCESS レジスタは、ファブリックアクセス可能な 32 ビットのレジスタで、ビットストリーム生成プロセス中にバージョン情報を組み込むことができるため、設計およびインプリメンテーション時間に与える影響を最小限にして、ユーザーに柔軟性を提供する最適なソリューションです。

JTAG USERCODE ソリューション

Virtex-5、Virtex-6、および 7 シリーズ FPGA は、USR_ACCESS と類似した IEEE Std. 1149.1 JTAG を使用する USERCODE をサポートしています。この USERCODE もユーザー指定値を格納できる 32 ビットのレジスタですが、USECODE の場合は FPGA ロジックを介してアクセスできないという制限があります。USERCODE は、JTAG を介して即座にアクセスすることを意図したレジスタです。このレジスタには Generate Programming File プロセス中でも入力できます。動作中のタイムスタンプ機能が不要、あるいはこのレジスタ値へロジックを介してアクセスする必要がない場合は、USERCODE ソリューションを選択肢として検討できます。

USR_ACCESS プリミティブ

USR_ACCESS レジスタのプリミティブ名は、USR_ACCESS_VIRTEX5、USR_ACCESS_VIRTEX6、および USR_ACCESS2 です。このアプリケーション ノートでは、これらのプリミティブすべてを USR_ACCESS と簡略して示しています。このコンポーネントを使用することによって、FPGA ビットストリームで格納された 32 ビット値へ FPGA ロジックから直接アクセスできます。USR_ACCESS は、外部コンフィギュレーション インターフェイス、JTAG、または内部コンフィギュレーション アクセスポート (ICAP) のいずれかを介してコンフィギュレーションレジスタへ追加書き込みを行うことで変更できますが、このアプリケーション ノートの目的上、ここでは静的な値として考えます。動的書き込みには、CFGCLK ポートおよび DATAVALID ポートが使用されます。静的な値の場合は、32 ビットの DATA バスのみ必要です。ここでは動的書き込み動作については説明していません。このレジスタへの動的書き込み動作の詳細は、各 FPGA ファミリのコンフィギュレーション ユーザー ガイドで、「AXSS レジスタ」を検索して参照してください。

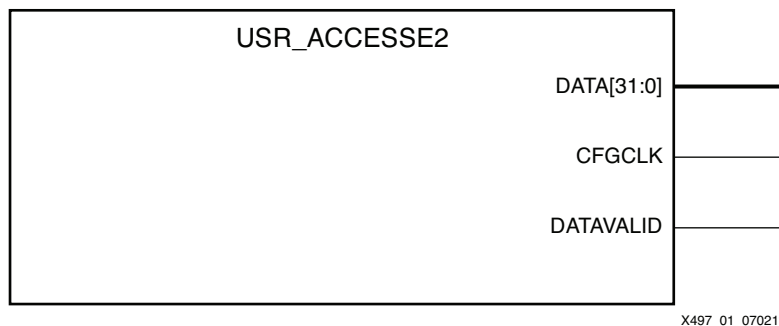


図 1：7 シリーズ FPGA の USR_ACCESS コンポーネント

タイムスタンプ

USR_ACCESS は、32 ビットのユーザー指定値を用いてコンフィギュレーションするか、あるいはタイムスタンプ機能を使用してビットストリーム生成プログラム (BitGen) で自動的にロードできます。リビジョン、デザイン追跡、シリアル ナンバーには、ユーザー指定値が使用できます。インプリメンテーションが複数回実行され、それによってデザイン最適化の値が変更されていても、ソース デザイン自体は変更されていない場合には、タイムスタンプ機能が有益です。タイムスタンプ値はビットストリームファイルのタイムスタンプと比較されて、デバイスのデザインを数あるソースからその 1 つと関連付けます。ソース コードを変更してタイムスタンプ機能をインプリメントするのは簡単ではありません。USR_ACCESS を使用することで、より正確なタイムスタンプ値を得ることができます。

コマンドの利用

コマンドを利用する場合は、[Generate Programming File] プロセス (BitGen) で **-g USR_ACCESS** スイッチを使用します。

```
-g USR_ACCESS: NONE | xxxxxxxxxx | TIMESTAMP
```

このオプションに値が入力されていない場合、または「NONE」と入力されている場合は、このレジスタに対して実行される動作はありません (デフォルトはすべて 0)。

NONE - DEFAULT

8 文字の 16 進数値が検出された場合、この値が USR_ACCESS レジスタへ格納されます。

```
xxxxxxxx
```

値としてキーワード、TIMESTAMP と入力された場合は、

```
TIMESTAMP
```

BitGen ツールが、次のフォーマットで、その時点のタイムスタンプを 32 ビットの USR_ACCESS レジスタへ挿入します。

```
dddd_MMMM_yyyyyy_hhhh_mmmmm_sssss
(bit 31) ..... (bit 0)
```

説明：

dddddd = 1 ~ 31 で「日」を示す 5 ビット

MMMM = 1 ~ 12 で「月」を示す 4 ビット

yyyyyy = 0 ~ 63 で「西暦」を示す 6 ビット (2000 年 ~ 2063 年)

hhhhh = 0 ~ 23 で「時」を示す 5 ビット

mmmmm = 0 ~ 59 で「分」を示す 6 ビット

sssss = 0 ~ 59 で「秒」を示す 6 ビット

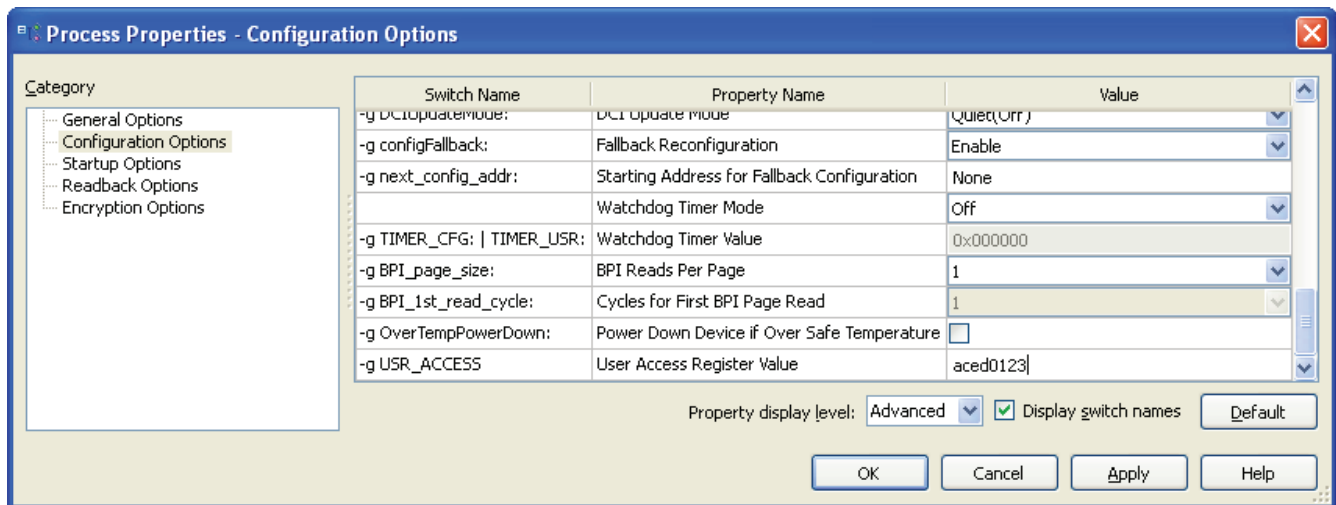
TIMESTAMP 値を使用する場合、分と秒の値はファイルのタイムスタンプと一致しない場合があります。タイムスタンプ値は、ビットストリーム生成プロセスが開始される時に取得されますが、オペレーティングシステムファイルのタイムスタンプはファイル生成プロセスの最後で取得されます。したがって、マシンのスピードや BitGen に要求される動作の複雑さなどによって、これらの値は厳密には一致しない場合があります。同様に、「時」または「日」の終りに近い段階でファイル生成が行われる場合も一致しない可能性があります。

ISE® ツール フロー

このオプションは、[Generate Programming File] プロセスで入力します。オプション設定方法は次のとおりです。

1. [Generate Programming File] を右クリックします。
2. [Process Properties] を選択します。
3. [Property Display Level] で [Advanced] を選択します。
4. [Category] で [Configuration Options] を選択します。
5. [User Access Register Value] へ図 2 のようにコマンドを入力します。
6. タイムスタンプ機能を使用する場合は、「aced0123」を「TIMESTAMP」に変更します。

注記：このオプションに何も入力しない場合はデフォルトの NONE となり、レジスタ値がすべて 0 になります。



X497_02_072211

図 2：コンフィギュレーション オプション

まとめ

USR_ACCESS は、デザインに変更を加えた場合のビットストリームの追跡や、ソースコードは変更せずにインプリメンテーションを実行したり、再インプリメントを実行した場合のビットストリーム追跡に使用する機能です。

補足資料 A : インスタンスエー ション テンプレート

ここでは、7 シリーズ デバイスのインスタンスレーション テンプレートを示します。Virtex-5 および Virtex-6 の場合、ポート名は同じでコンポーネント名のみ異なります。

- Virtex-5 : USR_ACCESS_VIRTEX5
- Virtex-6 : USR_ACCESS_VIRTEX6
- Artix™-7 : USR_ACCESSE2
- Kintex™-7 : USR_ACCESSE2
- Virtex-7 : USR_ACCESSE2

7 シリーズ デバイスの VHDL インスタンスレーション テンプレートは次のとおりです。

```
Library UNISIM;
use UNISIM.vcomponents.all;

USR_ACCESS_7series_inst : USR_ACCESSE2

port map (
  CFGCLK => CFGCLK, -- Not utilized in the static use case in this application
  note
  DATA => DATA, -- 32-bit output Configuration Data output
  DATAVALID => DATAVALID -- Not utilized in the static use case in this
  application note
);
```

7 シリーズ デバイスの Verilog インスタンスレーション テンプレートは次のとおりです。

```
USR_ACCESSE2 USR_ACCESS_7series_inst (
  CFGCLK(CFGCLK), // Not utilized in the static use case in this application
  note
  DATA(DATA), // 32-bit output Configuration Data output
  DATAVALID(DATAVALID) // Not utilized in the static use case in this
  application note
);
```

補足資料 B : ビットストリーム 構成

ビットストリーム内の `USR_ACCESS` レジスタ値を見つけるには、次のコマンドを探します。

```
Type 1, Write command, address 01101, 1 word
```

これらのコマンドの後にある 32 ビット値が `USR_ACCESS` レジスタの値です。コンフィギュレーションポートを介して、これらの値へ読み出し/書き込みを実行する際の構文については、各デバイスのコンフィギュレーション ユーザー ガイドにある「コンフィギュレーションの詳細」を参照してください。

タイムスタンプを含む Kintex-7 デバイスのビットストリームの一部を示します。

```
00110000000000100010000000000001
00000000000000000000000000000000
00110000000000011010000000000001 ← Type 1, write, address 01101, one word
11101001100101110101110101000011 ← USR_ACCESS 値
00110000000000100110000000000001
00000000000000000000000000000000
```

上記の例には、次のタイムスタンプ値が含まれています。

```
11101 0011 001011 10101 110101 000011
```

説明:

「日」を示す 5 ビット: 11101 = 29 日

「月」を示す 4 ビット: 0011 = 3 月

「西暦」を示す 6 ビット: 001011 = 2011 年

「時」を示す 5 ビット: 10101 = 21 時

「分」を示す 6 ビット: 110101 = 53 分

「秒」を示す 6 ビット: 000011 = 3 秒

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
2011 年 8 月 15 日	1.0	初版リリース

Notice of Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was

reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

本資料は英語版 (v1.0) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

この資料に関するフィードバックおよびリンクなどの問題につきましては、

jpn_trans_feedback@xilinx.com までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメールアドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。