

# SILOS-XとHarmonyにおけるPLIを使用した ユーザ定義システムタスクのインプリメンテーション

## はじめに

IEEE 1364 Verilog 標準には多数の標準システムタスクとシステム関数が用意されていますが、カスタマイズされたシステムタスクやシステム関数をベンダから提供されて使用したり、PLI (Programming Language Interface) ルーチンを使用して設計者が独自に作成したりしなければならないこともあります。IEEE 1364 Verilog 標準で提供されている PLI ルーチンには、TF ルーチン、ACC ルーチン、および VPI ルーチンがあります。本稿は、これら PLI ルーチンの使用法の詳細については触れませんが、ユーザ定義のシステムタスクやシステム関数を SILOS-X と Harmony で作成し使用するのに役立つ、PLI ライブラリの作成フローを解説します。

次の例は、ユーザ定義システムタスクの命名から、C ルーチンとしてのインプリメント、コンパイル、PLI ライブラリへのリンク、Verilog コードからの呼び出しの方法までの流れを分かりやすく示しています。図 1 に、SILOS-X と Harmony においてユーザ定義のシステムタスク / システム関数を作成し呼び出すまでのフローの概要を示します。

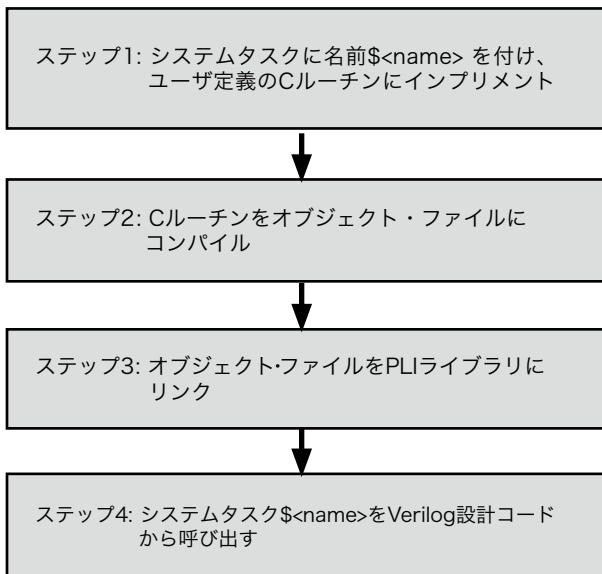


図 1: SILOS-X と Harmony における PLI システムタスクの作成から呼び出しまでのフロー

## PLI ライブラリの作成方法と Linux でのシミュレーションの例

**Step 1: ユーザ定義 C ルーチンの命名とインプリメント**  
設計の目的に応じて、設計者が独自に作成したユーザ定義システムタスクを命名し、PLI ルーチンを使って C ソース・コードにインプリメントすることができます。SILOS-X と Harmony は、ユーザ定義のコードとシミュレータ間のインタフェースとして、TF ルーチンおよび ACC ルーチンをサポートしています。これらの PLI ルーチンについての詳細は、IEEE から入手可能な『IEEE 1364 Verilog HDL 仕様』を参照してください。

tf\_putp() ルーチンを使用したユーザ定義システム関数の C ソース・コードを以下に示します。

```

/*//////////////////////////////////////
//
// title: C code for tf_putp() test to
// write a single simple value
//
// This PLI C code uses tf_putp() to
// write a simple value to a Verilog
// system task/function argument (param-
// eters).
//
////////////////////////////////////*/

#include <stdio.h>
#include "veriusertfs.h"

extern int putp_01_calltf();
char *veriusertfs_version_str = "";
int (*endofcompile_routines[])() =
{
    /*** my_eoc_routine, ***/
    0 /*** final entry must be 0 ***/
};

bool err_intercept(level, facility, code)
int level; char *facility; char *code;
{ return(true); }

s_tfcertfs veriusertfs[] = {
    {usertask,      0, 0, 0, putp_01_
    calltf,

```

```

0, "$putp_test", 1},
{0}
};

/* Use tf_putp() with a single task/function
argument */
int putp_01_calltf()
{
    int val = 1;
    io_printf("PLI Code:  tf_putp(1, val)
is writing a value of %x (hex)\n",val);
    tf_putp(1, val);
    return(0);
}

```

上記の C ソース・コードにおいて、\$putp\_test は、Verilog コードから呼び出し可能なユーザ定義システムタスクです。putp\_01\_calltf() は、tf\_putp() ルーチン (PLI ルーチン) をインプリメントするための C ルーチンです。\$putp\_test と putp\_01\_calltf() は、veriusertfs テーブルを介して実行時にリンクされます。

次の C ソース・コード例の veriusertfs テーブルは、s\_tfcell 構造内にエントリを持ちます。

```

s_tfcell veriusertfs[] = {
    {usertask,      0, 0, 0, putp_01_
calltf, 0, "$putp_test", 1},
    {0}
};

```

veriusertfs テーブルの各フィールドについての簡単な説明が、veriusertfs.h ファイル内にあります。

```

/* VERILOG user tasks and functions C
header file */
typedef struct t_tfcell
{
    short type;          /* either usertask
or userfunction */
    short data;         /* parameter for
the following routines */
    int (*checktf)();  /* routine for
checking parameters */
    int (*sizetf)();   /* for providing
size of function return value */
    int (*calltf)();   /* routine called
during simulation */
    int (*misctf)();   /* miscellaneous
routine (see below) */
    char *tfname;      /* the name of
the system task or function */
    int forwref;       /* indicates spe-
cial parameters allowed */
    char *tfveritool;  /* Which Veritool
owns the task */
    char *tferrmessage; /* An optional
special case error message
which will be
printed if the task is skipped */

```

```

/* these components are for system
usage only */
int hash;
struct t_tfcell *left_p;
struct t_tfcell *right_p;
char *namecell_p;
int warning_printed; /* Flag is set
when skipping warning is printed */
} s_tfcell, *p_tfcell;

```

## ステップ 2: C ソース・コードのコンパイル、オブジェクト・ファイル (\*.o) の作成

この C ソース・コードをコンパイルするには、ユーザ定義の C ソース・コード・ファイルに加えて、3 つのヘッダ・ファイル (acc\_user.h、ext\_user.h、および veriusertfs.h) が必要です。これらのファイルを入手するにはシルバコ・ジャパン (E-mail: jsupport@silvaco.com) へお問い合わせください。

C ソース・コードをコンパイルし、オブジェクト・ファイル (\*.o) を作成するには、次のコマンドを使用します。

```
gcc -fPIC -c myfile.c # step to compile C source
```

場合によっては、その他のコンパイラ・フラグを追加する必要があります。ソース・コードには、そのソース・コードに含まれるユーザ定義システム関数に関するエントリを持つ、veriusertfs テーブルが 1 つ含まれている必要があります。

## ステップ 3: PLI ライブラリへのリンク

次のコマンドを使用すると、mylib.so という名前の PLI ライブラリが作成されます。

```
"ld -o mylib.so -dy -G myfile.o" # step to load
object file
```

## ステップ 4: Verilog コードからのシステムタスク \$putp\_test() の呼び出しとシミュレーション

.so ファイルを作成した後、!pliload コマンドを使用して SILOS-X/Harmony に読み込ませる .so ファイルを指定します。!pliload コマンドは、モジュールの境界内にはない限り、どの Verilog インプット・ファイルの中にも置くことができます。ユーザ定義システムタスク \$tf\_putp() を使用した Verilog ファイルの例を以下に示します。

```

////////////////////////////////////
//
// title: testbench for tf_putp() test to
write a simple value
//
// This test writes a value to a scalar
reg data type
//

```

```
////////////////////////////////////  
`timescale 1ns/1ns  
  
!pliload /full_path/mylib.so  
  
module test;  
  
    reg r1;  
  
    initial  
        begin  
            #1  
            $display("\nTest Bench: executing  
\"$putp_test(r1);\" ");  
            $display("Test Bench: expect tfarg 1  
to receive 1 (hex)");  
            $putp_test(r1);  
            #1  
            $display("Test Bench: tfarg 1 re-  
ceived %h (hex)", r1);  
  
            #1 $display("\n\n-----End of Test  
Bench-----");  
            $finish;  
        end  
    endmodule
```

## シミュレーション結果

Linux マシンにおいて SILOS-X をバッチ・モードで実行した際のシミュレーション結果を以下に示します。

```
[yunc@silos pli01_linux]$ gcc -fPIC -c  
myfile.c  
[yunc@silos pli01_linux]$ ld -o mylib.so  
-dy -G myfile.o  
[yunc@silos pli01_linux]$ silosx -b  
Xlib: extension "XFree86-Misc" missing  
on display "shangrila:0.0".
```

S I L O S - X Version 4.10.5

Copyright (c) Copyright Simucad Design Automation 2007 All rights reserved.

No part of this program may be reproduced, transmitted, transcribed, or stored in a retrieval system, in any form or by any means without the prior written consent of

Simucad Design Automation, 4701 Patrick Henry Drive  
Santa Clara, California, 95054, U.S.A.  
(408)-567-1000 Fax: (408)-496-6080

```
Web Site: "www.simucad.com"  
Ready: in pli01.v  
Reading "/home/yunc/tmp/Silos/pli_example/pli01_linux/pli01.v"  
!pliload /home/yunc/tmp/Silos/pli_example/pli01_linux/pli01.so  
Ready: sim  
Highest level modules (that  
have been auto-instantiated):  
    test  
    2 total devices.  
Linking ...  
  
1 nets total: 0 saved and 0  
monitored.  
0 registers total: 0 saved.  
Done.  
  
Test Bench: executing "$putp_test(r1);"  
Test Bench: expect tfarg 1 to receive 1  
(hex)  
PLI Code: tf_putp(1, val) is writing a  
value of 1 (hex)  
Test Bench: tfarg 1 received 1 (hex)  
  
-----End of Test Bench-----  
$finish in file "/home/yunc/tmp/Silos/pli_example/pli01_linux/pli01.v" at line 26  
  
0 State changes on observable nets.  
  
Simulation stopped at the  
end of time 0.003us.  
Ready: exit
```

## まとめ

IEEE-1364 Verilog 標準には Verilog 言語を拡張する幅広い PLI ルーチンが提供されているため、設計者は設計の目的に応じてオリジナルのシステムタスクやシステム関数を作成できます。本稿では、SILOS-X と Harmony において PLI ライブラリを作成し、ユーザ定義のシステムタスクを標準の Verilog コードから呼び出すために使用したフローを解説しました。

## 参考文献

[[1] VerilogR HDL: A Guide to Digital Design and Synthesis, Second Edition.

[[2] IEEE Standard VerilogR Hardware Description Language, IEEE Std 1364-2001.