



Procedural Language Interface

یک مثال از نحوه ی نوشتن کد PLI و اجرای آن در یک سیمولاتور HDL

گردآورنده : نسترن نعمتی

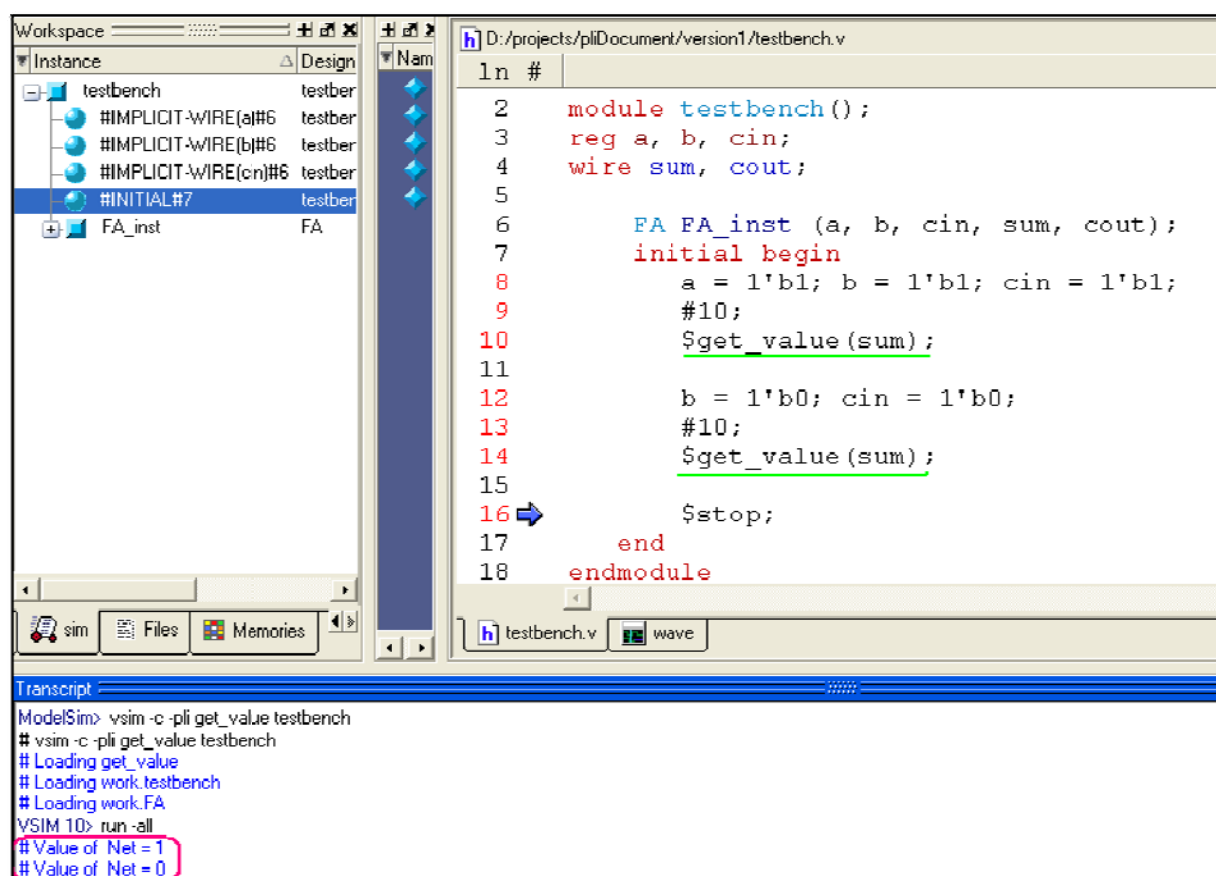
پاییز 1387

یک مثال از نحوه ی نوشتن کد PLI و اجرای آن در یک سیمولاتور HDL

در این document نحوه نوشتن یک کد کوچک PLI و اجرای روتین نوشته شده در یک سیمولاتور HDL شرح داده شده است.

1- یک مثال کوچک از کد PLI

هدف از این مثال نوشتن یک روتین PLI برای خواندن مقدار موجود در یکی از Net های موجود در مدار می باشد. نمونه ی کد testbench به زبان verilog که روتین مورد نظر را فراخوانی می کند در شکل 1 قابل مشاهده است.



شکل 1- نحوه ی فراخوانی روتین PLI و نمایش نتیجه ی اجرای آن در سیمولاتور

همچنین نتیجه ای که از اجرای این کد در modelsim به دست می آید نیز در پنجره ی Transcript از سیمولاتور modelsim قابل مشاهده است.

a. توضیحاتی بر روتین نوشته شده

کد PLI نوشته شده تحت زبان C در شکل 2 قابل مشاهده است. همانطور که در شکل اشاره شده است لازم است مسیر مربوط به header file های include شده را بر اساس مسیر نصب سیمولاتور خود تغییر دهید.

```
#include "C:\\Modeltech_eval_6.1f\\include\\veriusertfs.h"
#include "C:\\Modeltech_eval_6.1f\\include\\acc_user.h"
static PLI_INT32 get_value()
{
    handle arg1; /* handle for the argument of task */
    s_setval_value value_of_arg1;

    acc_initialize();

    arg1 = acc_handle_tfarg(1);

    //put the value of arg1(net value) into value_of_arg1
    value_of_arg1.format = accScalarVal; // specify the format first

    acc_fetch_value (arg1, "%", &value_of_arg1);

    //print 0, 1, 2 or 3 integers for 0,1, X or Z
    io_printf("Value of Net = %d \n", value_of_arg1.value.scalar);
    acc_close();
    return 0;
}

s_tfcell veriusertfs[] = {
    {usertask, 0, 0, 0, get_value, 0, "$get_value"},
    {0} /* last entry must be 0 */
};
```

replace this path with your modelSim installation path

registration part

شکل 2- کد PLI برای روتین get_value

در هنگام نوشتن روتین مورد نظر ابتدا باید آرگومان های ورودی را برای آن معرفی کنیم.

برای اینکار باید بدانیم که به ازای هر **object** ی در مدار که بخواهیم به آن دسترسی داشته باشیم لازم است یک متغیر از نوع **handle** برای آن تعریف کنیم. **Handle** در حقیقت یک **struct** است که اطلاعاتی از قبیل نوع **object** مورد بررسی یا نحوه دسترسی به آن را در اختیار تابعی از کتابخانه ی **acc** که قصد دسترسی به آن **object** را دارد، قرار می دهد.

پس در اینجا متغیر **arg1** را از نوع **handle** تعریف می کنیم و با استفاده از تابع **acc_handle_tfarg** آن را به آرگومان اول تابع مرتبط می کنیم.

همچنین هرگاه بخواهیم با مقدار یک **object** سروکار داشته باشیم – یعنی بخواهیم مقدار آن را بخوانیم یا مقداری را در آن بنویسیم – لازم است متغیری از نوع **struct – s_setval_value** ی که اطلاعاتی از قبیل مقدار، فرمت، نوع و... در مورد **object** را نگهداری می کند – برای آن تعریف کنیم.

برای استخراج مقدار آرگومان مورد نظر از **handle** آن و قرار دادن این مقدار در متغیر **value_of_arg1** باید از تابع **acc_fetch_value** استفاده کنیم. **Option** "%/" موجود در آرگومان های این تابع، نشان می دهد که تابع باید مقداری را که از **handle** استخراج می کند در متغیر **value_of_arg1** قرار دهد. انتخاب دیگر اینست که از تابع بخواهیم مقدار را به عنوان مقدار بازگشتی برگرداند. در اینحالت می توان قراخوانی تابع را مثلاً به شکل **acc_fetc_value(arg1, "%b", null)** تغییر داده و مقدار استخراج شده را به صورت باینری در مقدار بازگشتی تابع بدست آورد.

تنها نکته ای که باقی مانده اینست که پیش از استفاده از تابع **acc_fetch_value** لازم است فیلد فرمت موجود در **struct** ی که از نوع **setval_value** تعریف کرده ایم را برای **object** مورد نظر تنظیم کنیم. در اینجا به این دلیل که آرگومان ورودی ما یک **net** اسکالر است فرمت را به صورت **accScalarVal** تنظیم می کنیم. گزینه های دیگر برای پرکردن این فیلد را در فایل **LRM** می توان یافت.

b. نحوه ی رجیستر کردن روتین نوشته شده برای سیمولاتور HDL

هر روتینی که تحت PLI نوشته می شود برای اینکه قابل فراخوانی توسط سیمولاتور باشد باید خود را در سیمولاتور رجیستر کند. برای اینکار آرایه ای به نام `veriusertfs` وجود دارد که هر اندیس آن به یک `struct` به نام `s_tfcell` مرتبط می شود. لازم است مشابه آنچه در انتهای کد شکل 2 مشاهده می شود یک `entry` به ازای هر روتین تعریف شده در این آرایه قرار دهیم. `Entry` آخر این آرایه همیشه باید صفر باشد. فیلد اول نوع روتین تعریف شده را نشان می دهد که برای کاربردهای معمول عموماً آنرا به عنوان `USERTASK` تنظیم می کنیم. فیلد آخر، نامی که روتین با آن فراخوانی خواهد شد را نشان می دهد. این نام همواره باید با کاراکتر `$` شروع شود. فیلد پنجم نشان می دهد که در هنگام فراخوانی روتین – در اینجا `$get_value` – کدام تابع تعریف شده در کد `C` – در اینجا `get_value` – باید اجرا شود. در کاربردهای معمول بقیه فیلدها را برابر با `null` قرار می دهیم.

2- نحوه کامپایل کردن، لینک کردن و ساختن فایل `dll` مورد نظر

پس از نوشتن کد PLI و انجام عمل `registration`، نوبت به کامپایل و لینک کردن کد نوشته شده می رسد. برای کامپایل کردن در سیستم عامل ویندوز می توان از کامپایلر `Visual Studio 6` و برای `Unix` و `Linux` می توان از کامپایلرهای `gcc` و `cc` استفاده کرد. پس از اینکه کامپایل کردن با موفقیت انجام شد و فایل `object` مربوط به کد را بدست آوردیم باید آنرا با یکی از کتابخانه های `modelsim` به نام `mtipli.lib` لینک کنیم. برای اینکار می توان دستور زیر را در `command line` یا در یک `batch file` نوشته و اجرا کرد:

```
"linker-Path\link.exe" -dll -export:"veriusertfs" "object-file-  
path\get_value.obj" "related-path\mtipli.lib" /out:"desired-output-path  
\get_value.dll"  
pause
```

اگر اینکار با موفقیت و بدون خطا انجام گیرد به عنوان خروجی عمل لینک یک فایل dll در اختیار ما قرار می دهد که در مراحل بعدی باید از آن استفاده کنیم.

3- روش لینک کردن فایل dll در سیمولاتور و شبیه سازی و اجرای پروژه HDL

در این مرحله لازم است فایل dll بدست آمده را در دایرکتوری work پروژه modelSim کپی کنیم و توسط command line آن، عمل شبیه سازی در حضور این فایل را انجام دهیم. برای اینکار از دستور زیر می توان استفاده کرد(همچنین پیشنهاد می شود که این دستور را در یک do file قرار دهید تا اجرای مجدد آن آسانتر شود).

```
Vsim -c -pli your_dll_filename your_toplevel-design_name
```

این دستور برای شبیه سازی روتین get_value به شکل زیر در می آید:

```
Vsim -c -pli get_value testbench
```

در این دستور گزینه ی -c نشان می دهد که simulation در مد command line انجام خواهد شد و گزینه ی -pli نشانگر وجود dll های PLI است.

پس از اینکه این شبیه سازی با موفقیت انجام شد طبق معمول می توان testbench را اجرا کرد و نتیجه را مانند شکل 1 مشاهده کرد.