

Appendix B

PLI for Verilog

Introduction

Verilog provides a programming language interface (PLI) that consists of a C language interface mechanism, a set of routines to interact with the simulation environment, and a set of routines to access the Verilog internal data structures. This interface allows you to write C code to interact dynamically with the simulation and data structures. Moreover, you can create your own commands by compiling a C program and linking it into the Verilog executable. The access routines provide the ability to read (and sometimes write, as indicated) the following objects:

- module instances
- module ports
- module paths (R/W)
- inter-module paths (R/W)
- top-level modules
- primitive instances (R/W)
- nets
- registers
- parameters
- specparams
- timing checks (R/W)
- name events
- integer, real, and time variables

PLI Linking

Tasks and functions are dynamically linked to the PLI routines. You must supply a dynamically loadable object with an entry point named `init_usertfs`. This function must call `mti_RegisterUserTF` for each entry in the table of tasks and functions. The name of the file to load is specified in the *quickhdl.ini* file by the Veriuser entry (such as `Veriuser = hello.so`).

Example

Here is a simple example of how to use the PLI:

```
//-----Begin-----hello.c-----
#include "veriusert.h"

static int hello()
{
    io_printf("Hi there\n");
}

static s_tfcell veriuser_tfs[] = {
    {usertask, 0, 0, 0, hello, 0, "$hello", 0},
    {0}
};

void init_usertfs()
{
    p_tfcell usertf;

    for (usertf = veriuser_tfs; usertf; usertf++) {
        if (usertf->type == 0)
            return;
        mti_RegisterUserTF(usertf);
    }
}
//-----End-----hello.c-----
```

Compile `hello.c` then link `hello.o` with `hello.so` (on SunOS4):

```
cc -c -pic hello.c
ld -o hello.so hello.o
```

Supported C calls

The following calls to C routines are supported from the PLI:

```
io_mcdprintf
io_printf
mc_scan_plusargs
reason_endofcompile
reason_reset
reason_endofreset
reason_finish
tf_add_long
tf_[i]asynchoff
tf_[i]asynchon
tf_[i]clearalldelays
tf_compare_long
tf_[i]copypvc_flag
tf_divide_long
tf_dofinish
tf_dostop
tf_error
tf_[i]exprinfo
tf_[i]getcstringp
tf_getinstance
tf_[i]getlongp
tf_getlongtime
tf_getnextlongtime
tf_[i]getp
tf_[i]getpchange
tf_[i]getrealp
tf_[i]getroutine
tf_[i]gettflist
tf_gettime
tf_[i]gettimeprecision
tf_[i]gettimeunit
tf_[i]getworkarea
tf_long_to_real
tf_longtime_tostr
tf_message
tf_[i]mipname
tf_[i]movepvc_flag
tf_multiply_long
tf_[i]nodeinfo
```

```
tf_[i]nump
tf_[i]propagatep
tf_[i]putlongp
tf_[i]putp
tf_[i]putrealp
tf_real_to_long
tf_[i]rosynchronize
tf_scale_longdelay
tf_scale_realdelay
tf_[i]setdelay
tf_[i]setlongdelay
tf_[i]setrealdelay
tf_[i]setroutine
tf_[i]setttflist
tf_[i]setworkarea
tf_[i]sizep
tf_[i]spname
tf_[i]strgetp
tf_strgettime
tf_subtract_long
tf_[i]synchronize
tf_[i]testpvc_flag
tf_text
tf_[i]typep
tf_unscale_longdelay
tf_unscale_realdelay
tf_warning
```

The following calls to C routines are not presently supported from the PLI, but will be provided in a subsequent release:

```
acc_append_delays
acc_append_pulsere
acc_collect
acc_compare_handles
acc_count
acc_fetch_argc
acc_fetch_argv
acc_fetch_attribute
acc_fetch_attribute_int
acc_fetch_attribute_str
acc_fetch_delay_mode
```

acc_fetch_delays
acc_fetch_direction
acc_fetch_edge
acc_fetch_index
acc_fetch_paramtype
acc_fetch_paramval
acc_fetch_polarity
acc_fetch_pulsere
acc_fetch_value
acc_free
acc_handle_condition
acc_handle_conn
acc_handle_datapath
acc_handle_hiconn
acc_handle_loconn
acc_handle_modpath
acc_handle_notifier
acc_handle_path
acc_handle_pathin
acc_handle_pathout
acc_handle_port
acc_handle_simulated_net
acc_handle_tchk
acc_handle_tchkarg1
acc_handle_tchkarg2
acc_handle_terminal
acc_handle_tfarg
acc_handle_itfarg
acc_next
acc_next_bit
acc_next_cell
acc_next_cell_load
acc_next_driver
acc_next_hiconn
acc_next_input
acc_next_load
acc_next_loconn
acc_next_modpath
acc_next_net
acc_next_output
acc_next_parameter
acc_next_port
acc_next_portout

acc_next_primitive
acc_next_specparam
acc_next_tchk
acc_next_terminal
acc_object_in_ttypelist
acc_object_of_type
acc_release_object
acc_replace_delays
acc_replace_pulsere
acc_set_interactive_scope
acc_set_pulsere
acc_set_value
acc_vcl_add
acc_vcl_delete

tf_strdelputp
tf_istrdelputp
tf_strlongdelputp
tf_istrlongdelputp
tf_strrealdelputp
tf_istrrealdelputp

vpi_chk_error
vpi_compare_objects
vpi_free_object
vpi_get
vpi_get_cb_info
vpi_get_delays
vpi_get_str
vpi_get_systf_info
vpi_get_time
vpi_get_value
vpi_get_vlog_info
vpi_handle
vpi_handle_by_index
vpi_handle_by_name
vpi_handle_multi
vpi_iterate
vpi_mcd_close
vpi_mcd_name
vpi_mcd_open
vpi_mcd_printf
vpi_printf

```
vpi_put_delays  
vpi_put_value  
vpi_register_cb  
vpi_register_systf  
vpi_remove_cb  
vpi_scan
```

