



Introduction to the IBIS Macro Model Library

Arpad Muranyi
Signal Integrity Engineering

March 10, 2006





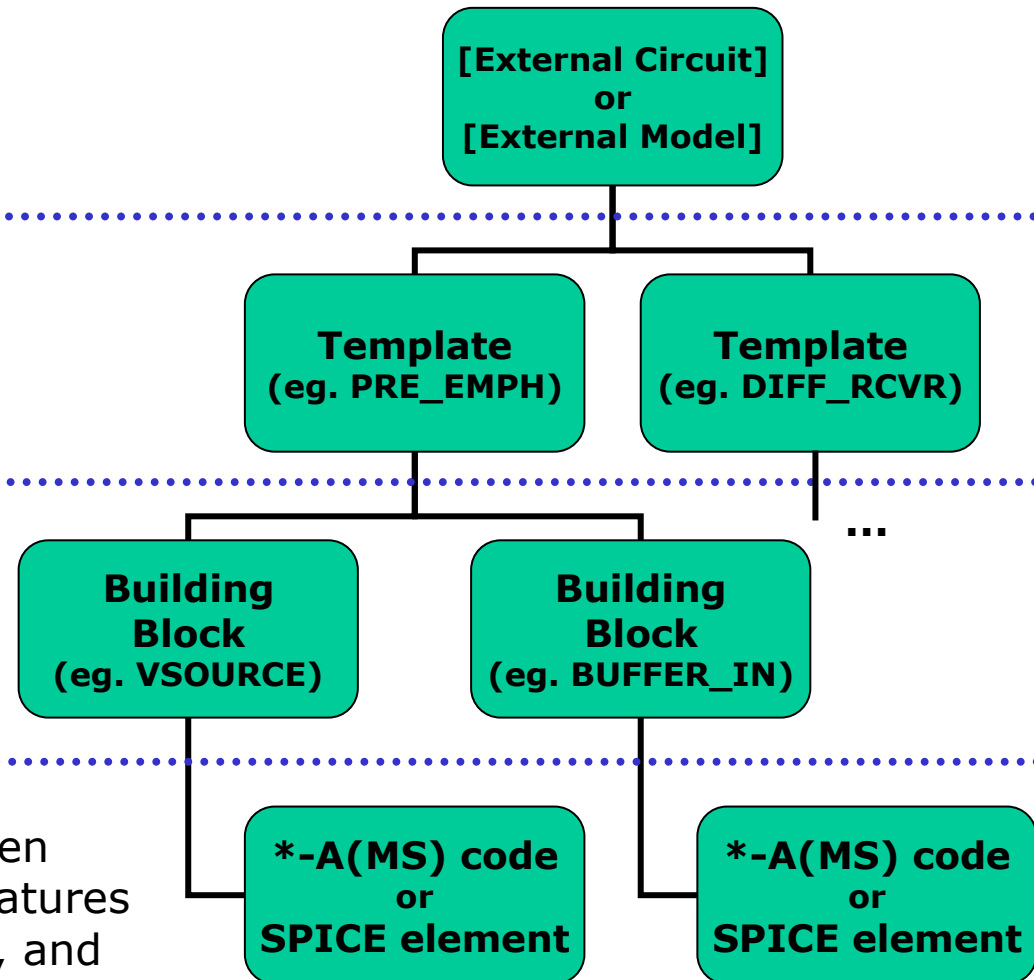
IBIS macro modeling concept

IBIS File

[External Circuit] or
[External Model] call
macro model templates

Macro model templates
call building blocks from
standard library

Building blocks are written
using the analog only features
of the *-AMS languages, and
can be substituted with native
SPICE elements in SPICE tools
if necessary



*Other brands and names are the property of their respective owners





Contents of the library

- **Three resistors**
 - ◆ R, VCR, CCR
- **Three capacitors**
 - ◆ C, VCC, CCC
- **Four inductors**
 - ◆ L, VCL, CCL, K
- **22 voltage and 22 current sources, including**
 - ◆ Delay, Min, Max, Abs, Sum, Mult, Div, PWL
- **An ideal T-line**
- **Eight IBIS buffers**
 - ◆ Input, Output, IO, 3-state, Opensource, IO_opensource, Opensink, IO_opensink



*Other brands and names are the property of their respective owners





Philosophy of the test suite

- Assumption:
 - ◆ HSPICE* **with Verilog-A option** installed, and/or
 - ◆ SMASH* installed (for the VHDL-A(MS) examples)
- All examples are ready to go
 - ◆ No editing required, just simulate and look at the waveforms
 - ◆ The examples are simple, just enough to show the concept
 - ◆ No attempt was made to show all possibilities
 - ◆ The data files contain very short data tables so the waveforms may not all be smooth and rounded
 - ◆ Some lines are commented out in the VHDL-A(MS) building blocks to allow them work in the evaluation version of SMASH (Seduction)
- A PERL script has been developed by Paul Fernando (NCSU) to extract and reformat the data from IBIS files so that the library building blocks can read it



*Other brands and names are the property of their respective owners





Parameter data file format

```

-----
-- C_comp parameters
-----
C_comp
5.00e-12
kC_comp_pc
0.25
kC_comp_pu
0.25
kC_comp_pd
0.25
kC_comp_gc
0.25
-----
-- Vectors of the IV curve tables
-----
Ipc_data
0.08
0.00
0.00
0.00
Vpc_data
-5.00
-1.00
5.00
10.00

Ipu_data
0.10
0.00
-0.10
-0.20
. . .
. . .

```

VHDL-A(MS)

```

`define IO_data \
\
.C_comp(5.0p), \
.kC_comp_pc(0.25), \
.kC_comp_pu(0.25), \
.kC_comp_pd(0.25), \
.kC_comp_gc(0.25), \
\
.IVpc_length(4), \
.Ipc_data({ 0.08, 0.00, 0.00, 0.00}), \
.Vpc_data({-5.00, -1.00, 5.00, 10.00}), \
.IVpu_length(4), \
.Ipu_data({ 0.10, 0.00, -0.10, -0.20}), \
.Vpu_data({-5.00, 0.00, 5.00, 10.00}), \
.IVpd_length(4), \
.Ipd_data({-0.10, 0.00, 0.10, 0.20}), \
.Vpd_data({-5.00, 0.00, 5.00, 10.00}), \
.IVgc_length(4), \
.Igc_data({-0.08, 0.00, 0.00, 0.00}), \
.Vgc_data({-5.00, -1.00, 5.00, 10.00}), \
\
. . .
. . .

```

Verilog-A(MS)





Test suite architecture

- **Top level SPICE files**
 - ◆ Include the simulation control statements
 - ◆ Contain simple stimulus source(s), load(s) and a call to the macro model netlist (i.e. template)
 - ◆ **Equivalent of the SI simulator tool's IBIS environment with an IBIS file using an [External Model] or [External Circuit] statement**
- **“Macro model template”-s**
 - ◆ Contain Verilog-A(MS) or VHDL-A(MS) netlists to instantiate the building blocks from the library
 - ◆ Show how to pass parameter values into the instances
 - ◆ **Equivalent of a “complicated buffer” macro model**
- **Model library file**
 - ◆ Contains the various building blocks of “primitives”



*Other brands and names are the property of their respective owners





File system of the Verilog-A(MS) distribution for HSPICE

- **Installation directory of your choice**
 - ◆ One .sp and .va file per library building block
 - ◆ The .sp file contains the simulation control and stimulus statements and the call to the .va netlist (or the “template” file)
- **Macro_lib subdirectory**
 - ◆ Contains two mandatory files and the library file
 - ◆ constants.vams
 - ◆ disciplines.vams
 - ◆ IBIS_macro_library.va
 - ◆ plus a small collection of parameter data files (*.dat)
 - ◆ these files contain IBIS buffer parameters, such as C_comp, I-V and V-t tables, or
 - ◆ appropriate data tables for the PWL sources

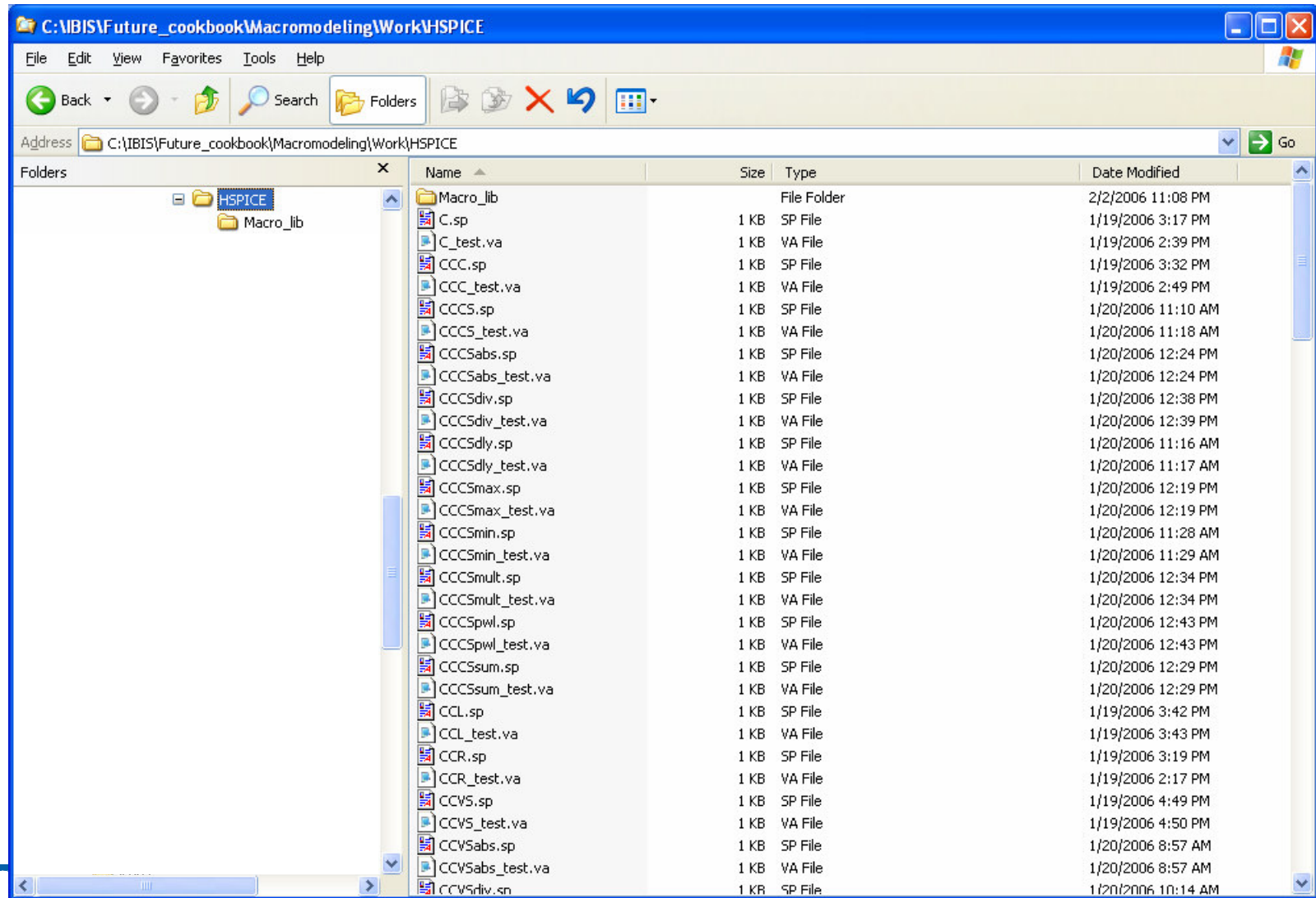


*Other brands and names are the property of their respective owners



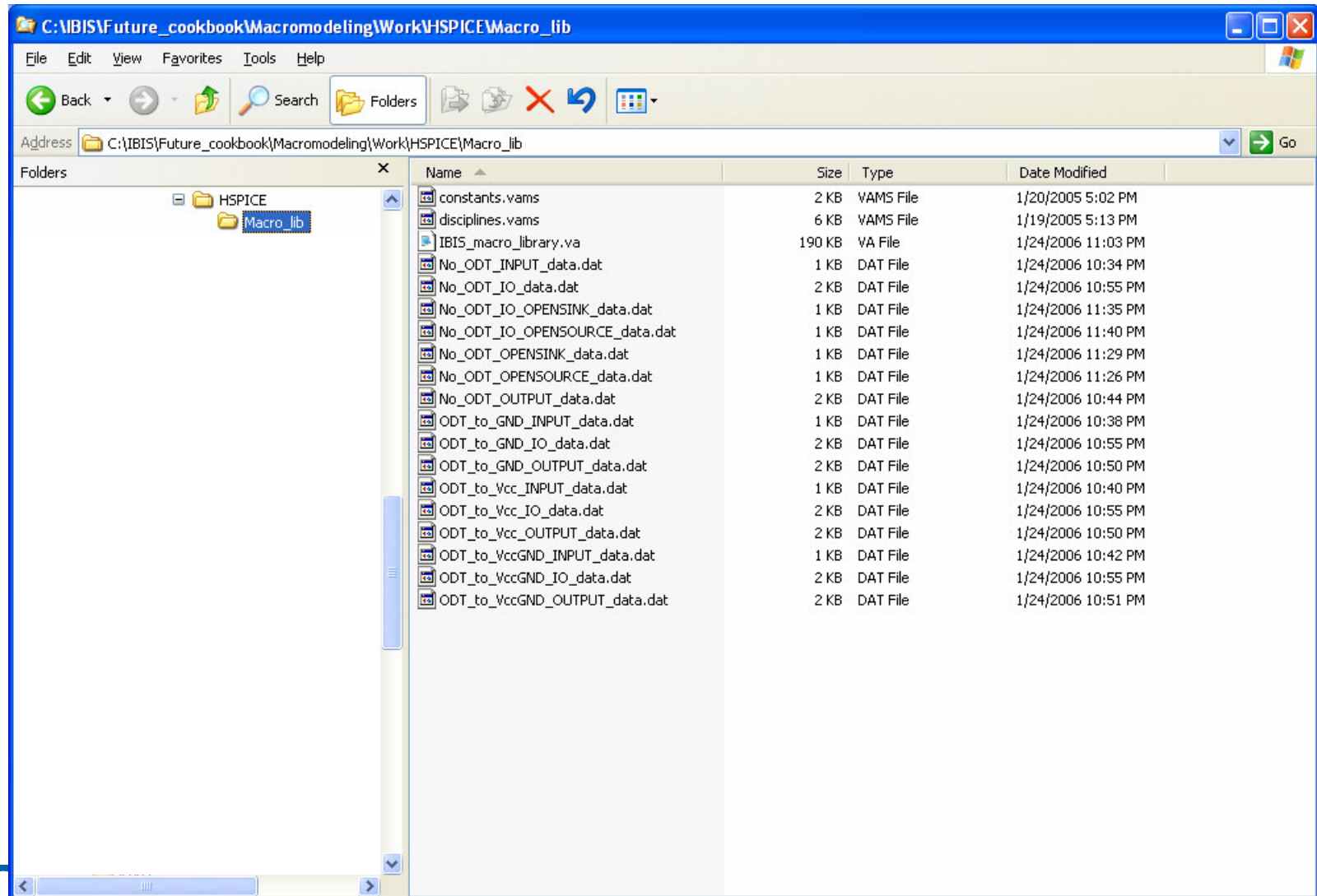


File system of the Verilog-A(MS) distribution for HSPICE





File system of the Verilog-A(MS) distribution for HSPICE





File system of the VHDL-A(MS) distribution for SMASH

- **Installation directory of your choice**
 - ◆ One .nsx and .pat file per library building block
 - ◆ The .pat (pattern) file contains the simulation control statements
 - ◆ The .nsx file contains both the top level SPICE and the VHDL-AMS netlist of the “template”
- **Macro_lib subdirectory**
 - ◆ Contains the library and a function file
 - ◆ IBIS_macro_library.vhd
 - ◆ MacroLib_functions.vhd
 - ◆ plus a collection of parameter data files (*.txt)
 - ◆ these files contain IBIS buffer parameters, such as C_comp, I-V and V-t tables, or
 - ◆ appropriate data tables for the PWL sources

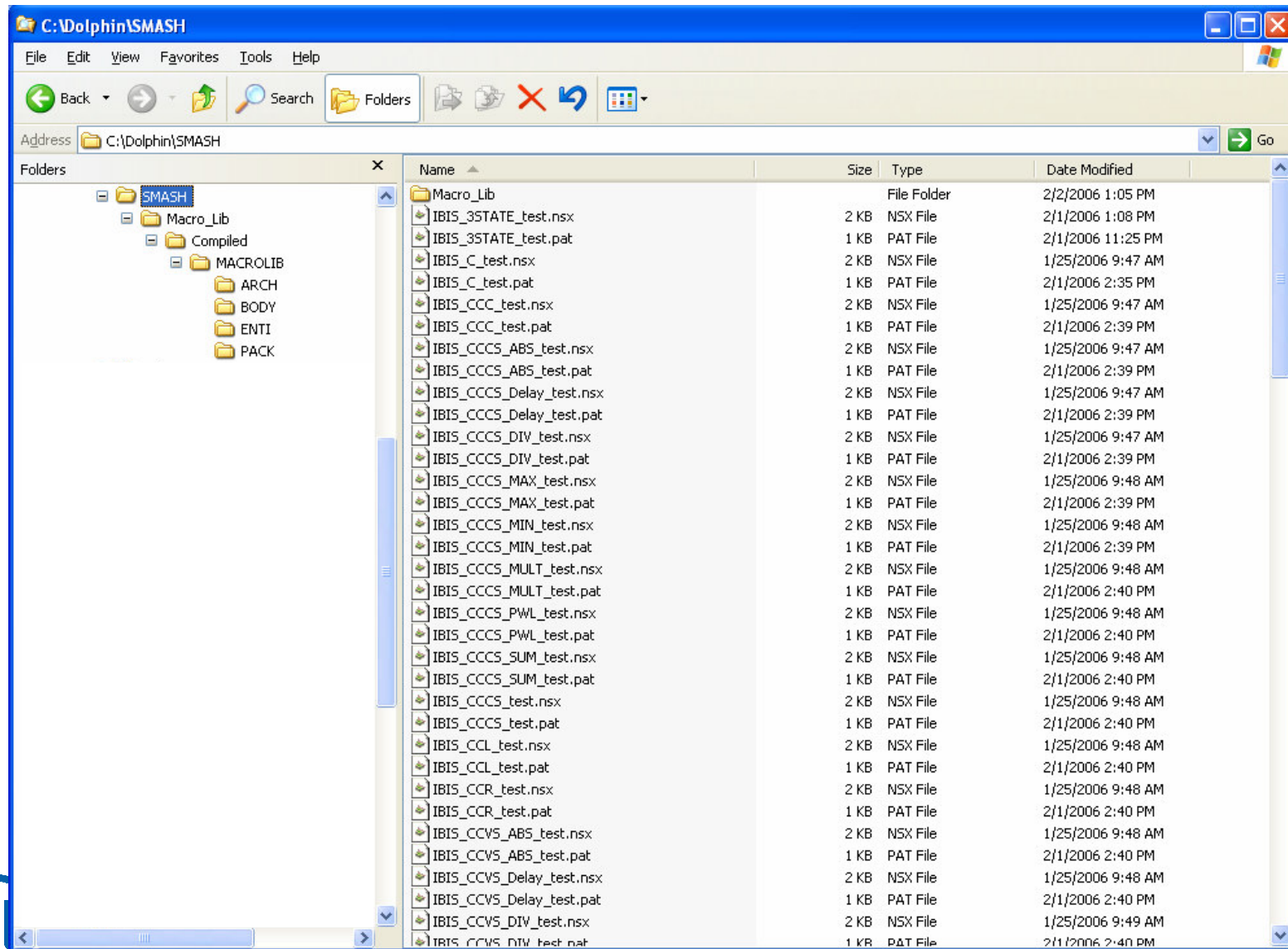


*Other brands and names are the property of their respective owners





File system of the VHDL-A(MS) distribution for SMASH

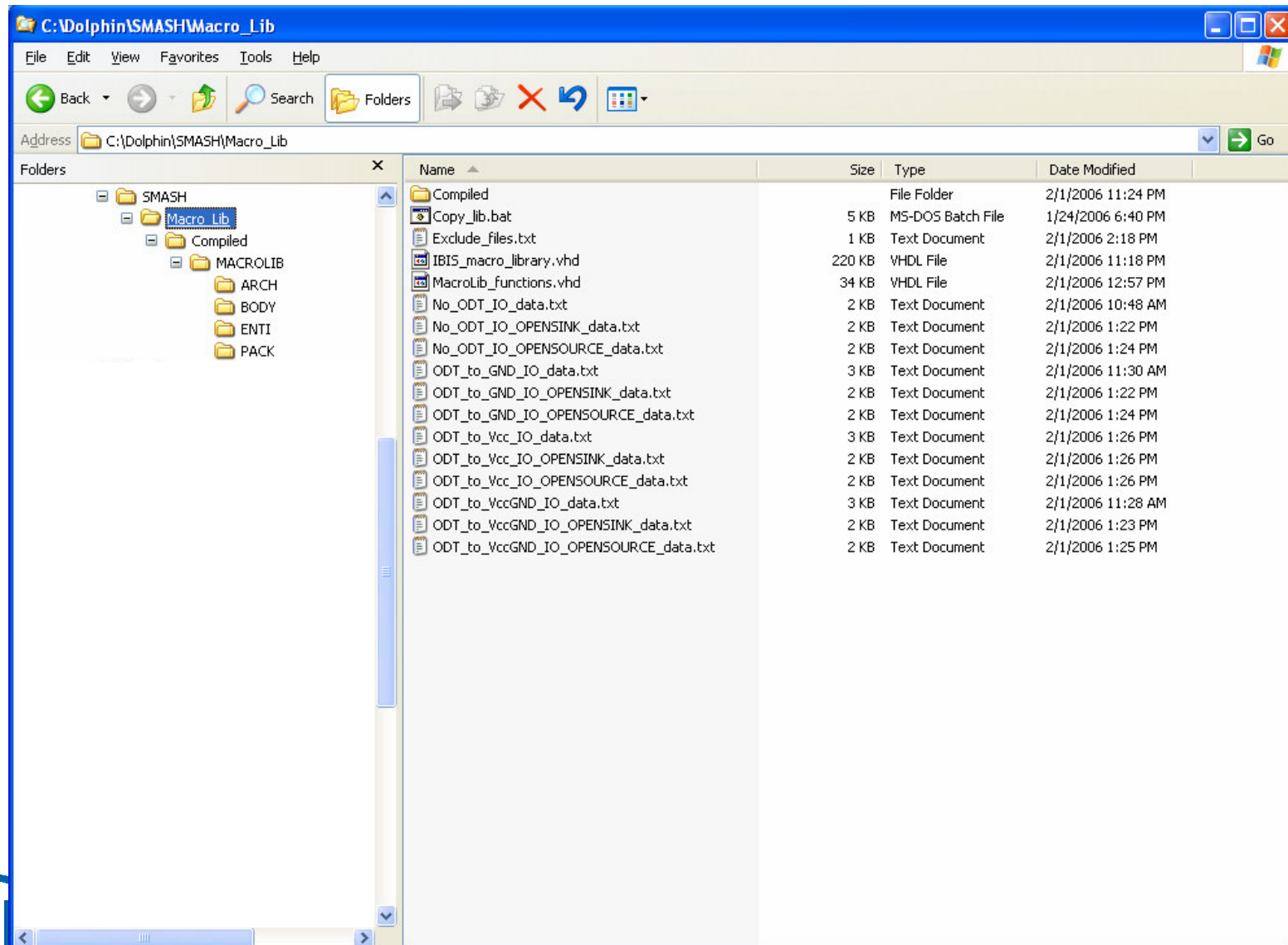


*Other brands and names are the property of their respective owners





File system of the VHDL-A(MS) distribution for SMASH





The “Compiled” subdirectory

- The Macro_lib directory of the VHDL-A(MS) distribution includes a directory called “Compiled”
 - ◆ This contains a compiled version of the library and all of its functions as a convenience to speed up testing
 - ◆ You don’t have to use it, but it can save you time
 - ◆ The “Copy_lib.bat” file copy the compilation into the work area of each library building block example
 - ◆ (Copying is faster than compiling it 63 times)
 - ◆ The “Exclude_files.txt” file will prevent the duplication of files which do not need to be copied
- This is a SMASH specific step, other tools may deal with the WORK and user library hierarchy in a different manner

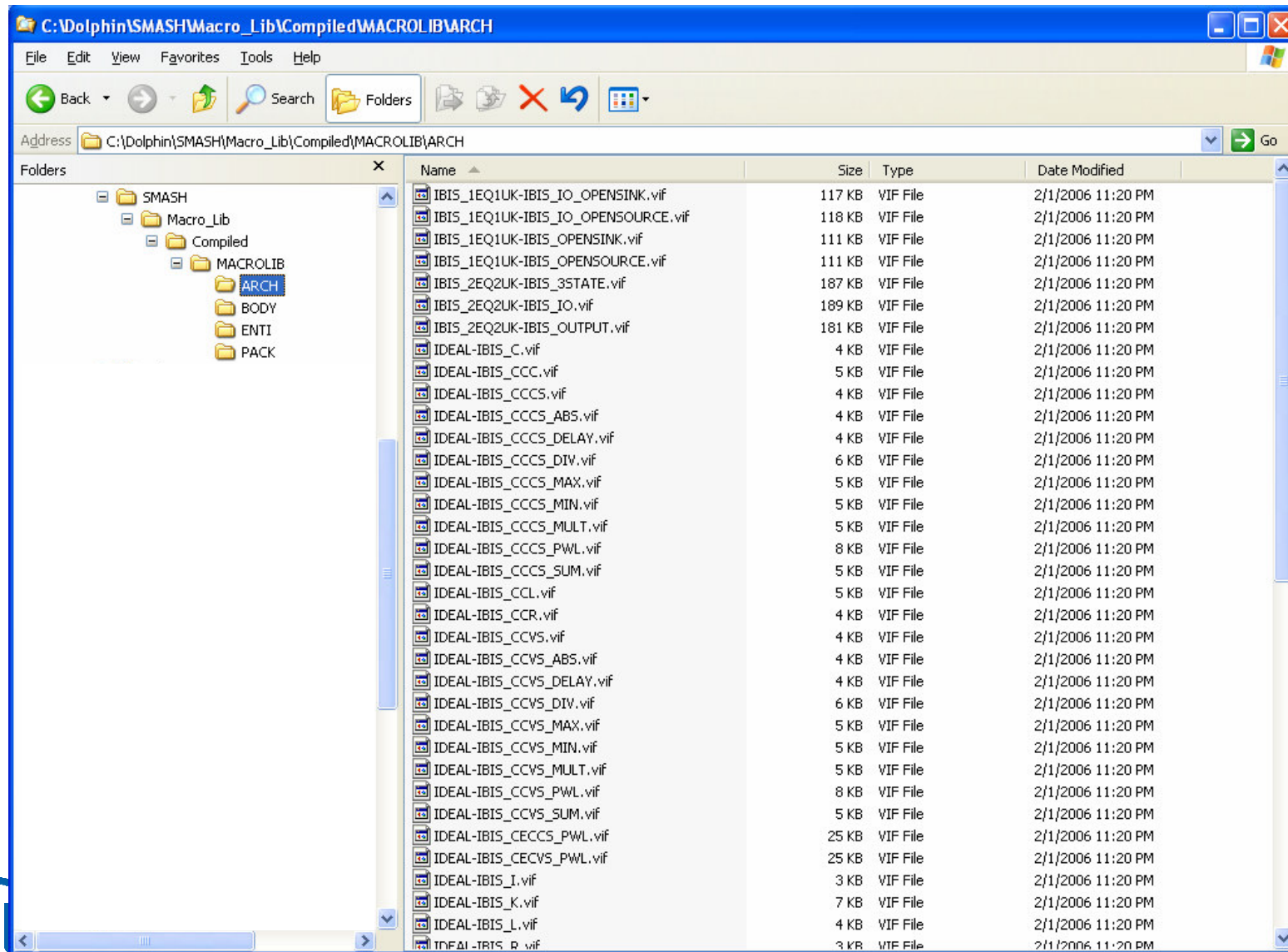


*Other brands and names are the property of their respective owners





File system of the VHDL-A(MS) distribution for SMASH





Resistor example - simulation code

macro model netlist

library call parameters nodes

control statements

```
SMASH 5.6.0
File Edit View Design Simulate Waveforms Tools Windows Help

C:\Dolphin\Macro_Lib\VHDMIBIS_R_test.nsx
1 >>> SPICE
2 *-----
3 Vp1s P1s 0 PULSE ( 0.0 1.0 1.0ns 1.0ns 1.0ns 4.0ns 10.0ns )
4 *
5 R1 P1s VHDL_p R= 100
6 X1 VHDL_p 0 IBIS_R_VHDL(Simple_test)
7 *
8 R2 P1s RR R= 100
9 R3 RR 0 R= 100
10 *
11 *-----
12 >>> VHDL
13 -----
14 library IEEE, MacroLib;
15 use IEEE.ELECTRICAL_SYSTEMS.all;
16
17 entity IBIS_R_VHDL is
18     port (terminal Node_p, Node_n : electrical);
19 end entity IBIS_R_VHDL;
20
21 architecture Simple_test of IBIS_R_VHDL is
22
23 begin
24
25     IBIS_R1 : entity MacroLib.IBIS_R(IDEAL)
26         generic map ( Rval => 50.0,
27                     Scale => 2.0 )
28         port map ( P => Node_p,
29                 N => Node_n );
30
31 end architecture Simple_test;
32 -----
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

C:\Dolphin\Macro_Lib\VHDMIBIS_R_test.pat
1 .VHDL SET KIND=AMS
2 *
3 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\MacroLib_functions.vhd
4 .VHDL ADD LIBRARY=MacroLib REFLIB=MacroLib
5 *
6 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\IBIS_macro_library.vhd
7 .VHDL ADD LIBRARY=WORK REFLIB=MacroLib
8 *
9 .Tran lps 15ns 0s noise=no noisestep=10ns traceBreak=yes
10 .Eps lu 100m 100n
11 .Tolerance DEFAULT_TOLERANCE lu
12 .H lfs lfs 10ps 125m 2
13 .Method TRAP
14 *
15 .Trace Tran V(P1s) V(VHDL_p) V(RR) Min=-1.0000000E+000 Max=3.0000000E+000
16
```





Resistor example - library code

```
-----  
-----  
library IEEE;  
use IEEE.electrical_systems.all;  
-----  
entity IBIS_R is  
  
    generic (Rval  : real := 1.0;  
             Scale : real := 1.0);  
  
    port (terminal p, n : electrical);  
  
end entity IBIS_R;  
-----  
architecture ideal of IBIS_R is  
  
    quantity Vout across Iout through p to n;  
  
begin  
  
    Vout == Scale * Rval * Iout;  
  
end architecture ideal;  
-----  
-----
```

VHDL-A(MS)

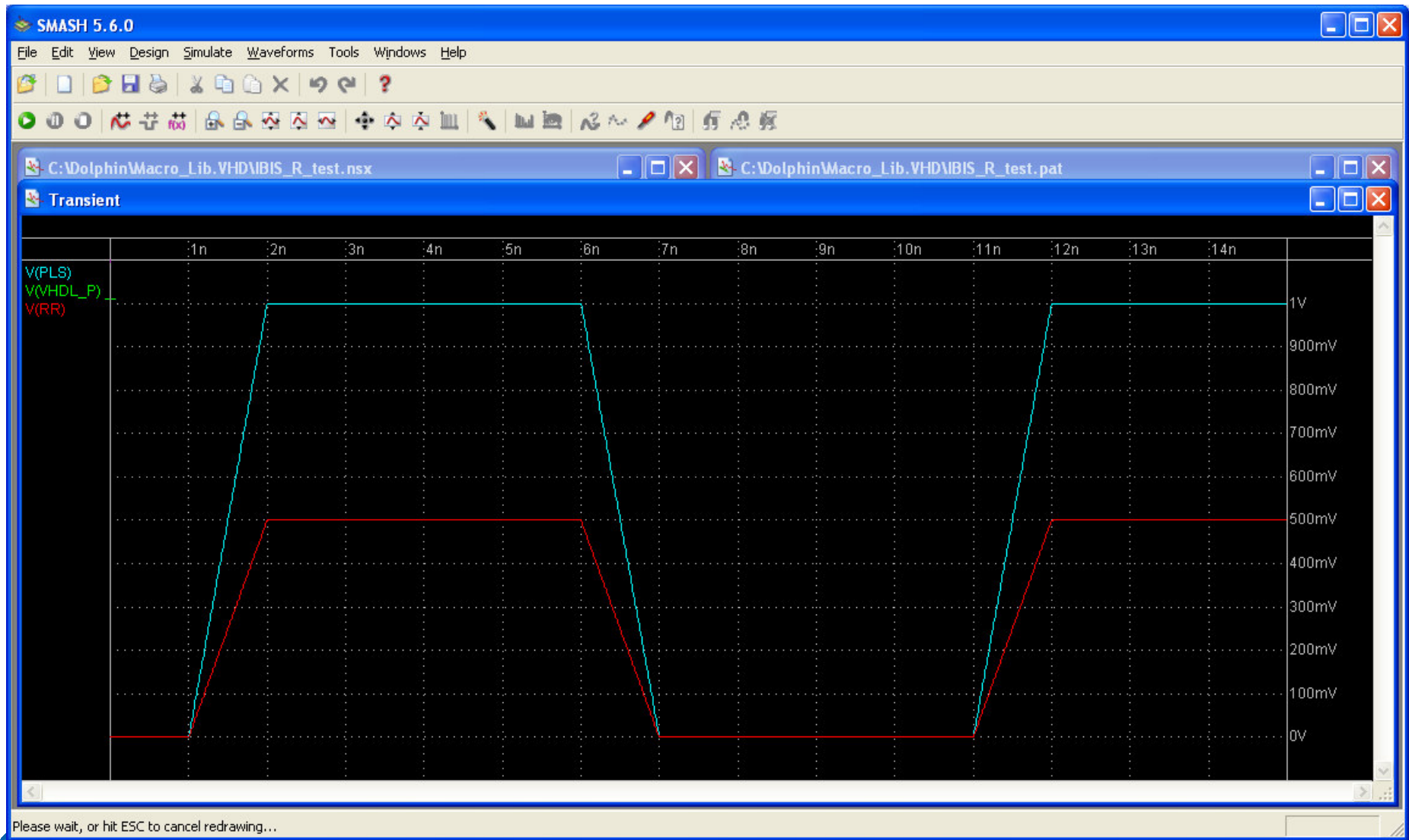
```
//=====  
//=====  
module IBIS_R (p, n);  
    electrical p, n;  
    branch      (p, n) Out;  
    parameter real Rval  = 1.0;  
    parameter real Scale = 1.0;  
  
    analog begin  
        V(Out) <+ Scale * Rval * I(Out);  
    end  
endmodule  
//=====  
//=====
```

Verilog-A(MS)





Resistor example - waveforms





VCC example - simulation code

macro model netlist

library call parameters

nodes

control statements

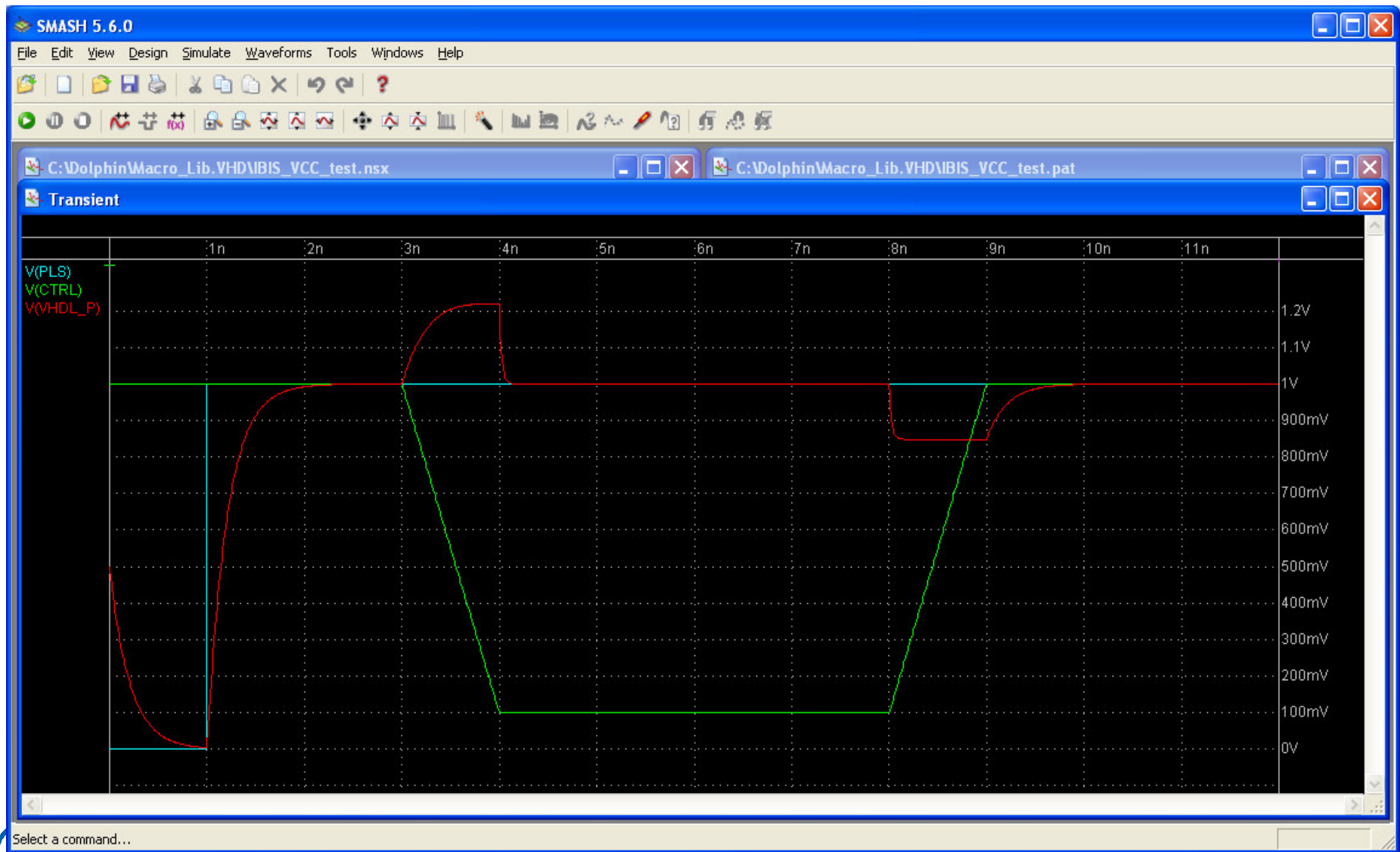
```
SMASH 5.6.0
File Edit View Design Simulate Waveforms Tools Windows Help

C:\Dolphin\Macro_Lib\VHDMIBIS_VCC_test.nsx
1 >>> SPICE
2 *-----
3 Vp1s P1s 0 PULSE ( 0.0 1.0 1.0ns 1.0ps 1.0ps 50.0ns 100.0ns
4 Vctrl Ctrl 0 PULSE ( 1.0 0.1 3.0ns 1.0ns 1.0ns 4.0ns 10.0ns
5 *
6 R1 P1s VHDL_p R= 100
7 X1 VHDL_p 0 Ctrl 0 IBIS_VCC_VHDL(Simple_test)
8 *
9 *-----
10 >>> VHDL
11 -----
12 library IEEE, MacroLib;
13 use IEEE.ELECTRICAL_SYSTEMS.all;
14
15 entity IBIS_VCC_VHDL is
16     port (terminal Node_p, Node_n,
17           Ctrl_p, Ctrl_n : electrical);
18 end entity IBIS_VCC_VHDL;
19
20 architecture Simple_test of IBIS_VCC_VHDL is
21
22 begin
23
24     IBIS_C1 : entity MacroLib.IBIS_VCC(IDEAL)
25         generic map ( V0 => 0.5,
26                       Scale => 2.0e-12 )
27         port map ( P => Node_p,
28                   N => Node_n,
29                   PS => Ctrl_p,
30                   NS => Ctrl_n );
31
32 end architecture Simple_test;
33 -----

C:\Dolphin\Macro_Lib\VHDMIBIS_VCC_test.pat
1 .VHDL SET KIND=AMS
2 *
3 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\MacroLib_functions.vhd
4 .VHDL ADD LIBRARY=MacroLib REFLIB=MacroLib
5 *
6 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\IBIS_macro_library.vhd
7 .VHDL ADD LIBRARY=WORK REFLIB=MacroLib
8 *
9 .Tran lps 12ns 0s noise=no noisestep=10ns traceBreak=yes
10 .Eps lu 100m 100n
11 .Tolerance DEFAULT_TOLERANCE lu
12 .H lfs lfs 10ps 125m 2
13 .Method TRAP
14 .Trace Tran V(P1s) V(Ctrl) V(VHDL_p) Min=-1.0000000E+000 Max=3.0000000E
15 *
16
```



VCC example - waveforms





IBIS_IO buffer example - code

```
C:\DolphinMacro_Lib.VHDMIBIS_IO_test.nsx
1 >>> SPICE
2 *-----
3 Vvcc Vvcc 0 DC= 5.0
4 Vvtt Vvtt 0 DC= 2.5
5 Vpls Pls 0 PULSE ( 0.0 1.0 1.0ns 0.1ns 0.1ns 4.9ns 10.0ns )
6 *
7 X1 Pls Out Vvcc 0 Vvcc IBIS_IO_VHDL(Simple_test)
8 * in out pow gnd en
9 *
10 R1 Out Vvtt R= 50
11 C1 Out Vvtt C= 5.0pF $ Substitutes C_comp in the VHDL-AMS model
12 *
13 *-----
14 >>> VHDL
15 -----
16 library IEEE, MacroLib;
17 use IEEE.ELECTRICAL_SYSTEMS.all;
18
19 entity IBIS_IO_VHDL is
20     port (terminal Input, Output,
21           Power, Ground,
22           Enable      : electrical);
23 end entity IBIS_IO_VHDL;
24
25 architecture Simple_test of IBIS_IO_VHDL is
26
27 begin
28
29     IBIS_I01 : entity MacroLib.IBIS_IO(IBIS_2EQ2UK)
30
31     -- Use ONLY ONE of the following four lines
32
33     generic map ( DataFile => ".\Macro_lib\No_ODT_IO_data.txt" )
34     -- generic map ( DataFile => ".\Macro_lib\ODT_to_GND_IO_data.txt" )
35     -- generic map ( DataFile => ".\Macro_lib\ODT_to_Vcc_IO_data.txt" )
36     -- generic map ( DataFile => ".\Macro_lib\ODT_to_VccGND_IO_data.txt" )
37
38     port map ( PU_ref => Power,
39               PD_ref => Ground,
40               Pad    => Output,
41               In_D   => Input,
42               EN_D   => Enable,
43               -- Rcv_D => RcvOut,
44               PC_ref => Power,
45               GC_ref => Ground );
46
47 end architecture Simple_test;
48 -----
49
50 C:\DolphinMacro_Lib.VHDMIBIS_IO_test.pat
1 .VHDL SET KIND=AMS
2 *
3 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\MacroLib_functions.vhd
4 .VHDL ADD LIBRARY=MacroLib REFLIB=MacroLib
5 *
6 .VHDL COMPILE LIBRARY=MacroLib SOURCE=.\Macro_lib\IBIS_macro_library.vhd
7 .VHDL ADD LIBRARY=WORK REFLIB=MacroLib
8 *
9 .Tran lps 15ns 0s noise=no noisestep=10ns traceBreak=yes
10 .Eps lu 100m 100n
11 .Tolerance DEFAULT_TOLERANCE lu
12 .H lfs lfs 10ps 125m 2
13 .Method TRAP
14 *
15 .Trace Tran V(PLS) V(OUT) Min=-3.7500000E-001 Max=4.1250000E+000
16 .Trace Tran X1.IBIS_I01.KPU X1.IBIS_I01.KPD Min=-5.8092432E-001 Max=1.4
17 .VhdlTrace Tran X1.IBIS_I01.EVENTSTATE
18 .VhdlTrace Tran X1.IBIS_I01.PU_ON
19 .VhdlTrace Tran X1.IBIS_I01.PD_OFF
20 .VhdlTrace Tran X1.IBIS_I01.PD_ON
21 .VhdlTrace Tran X1.IBIS_I01.PU_OFF
22
```

macro
model
netlist

control statements

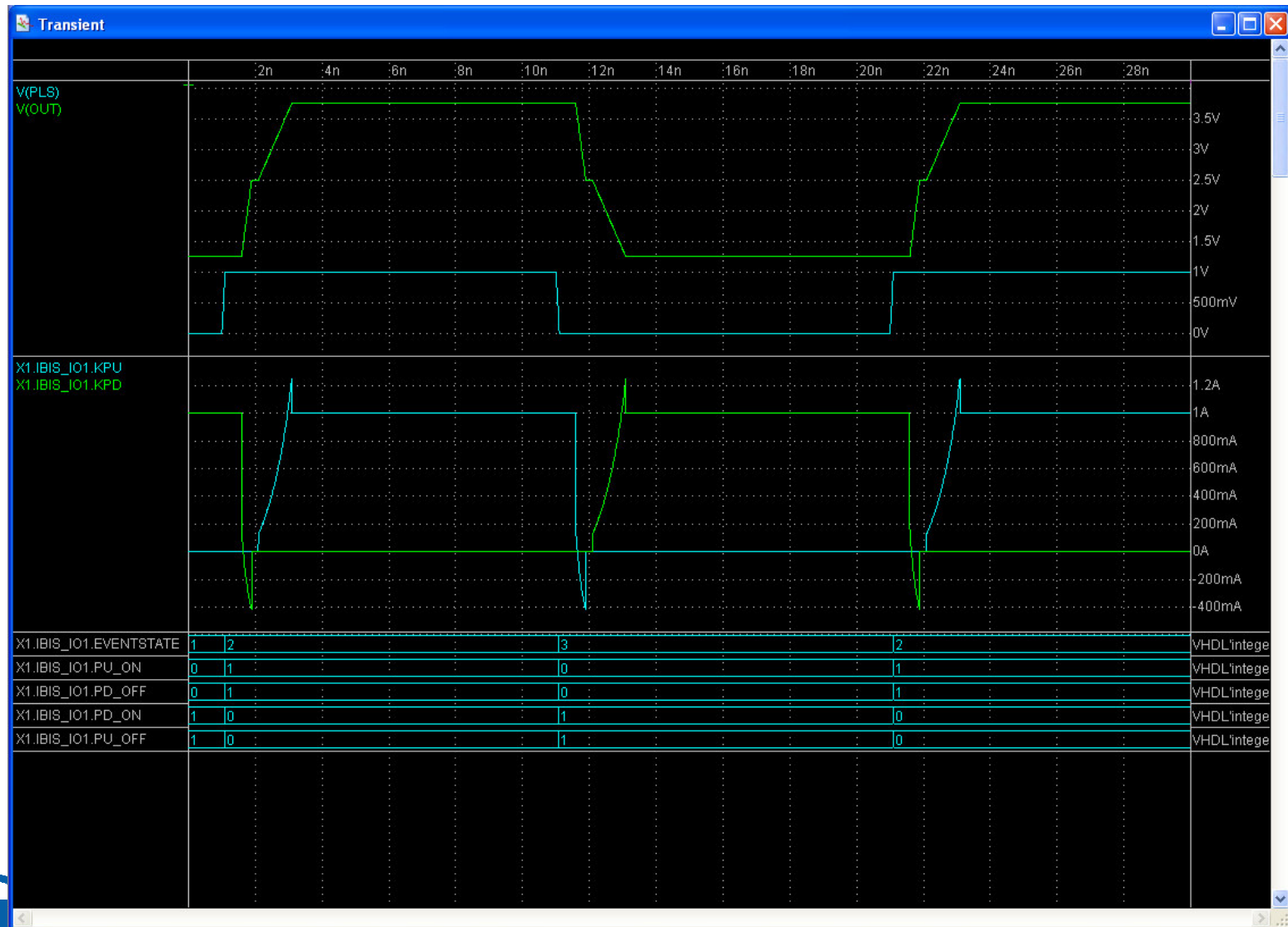
library call

parameters

nodes

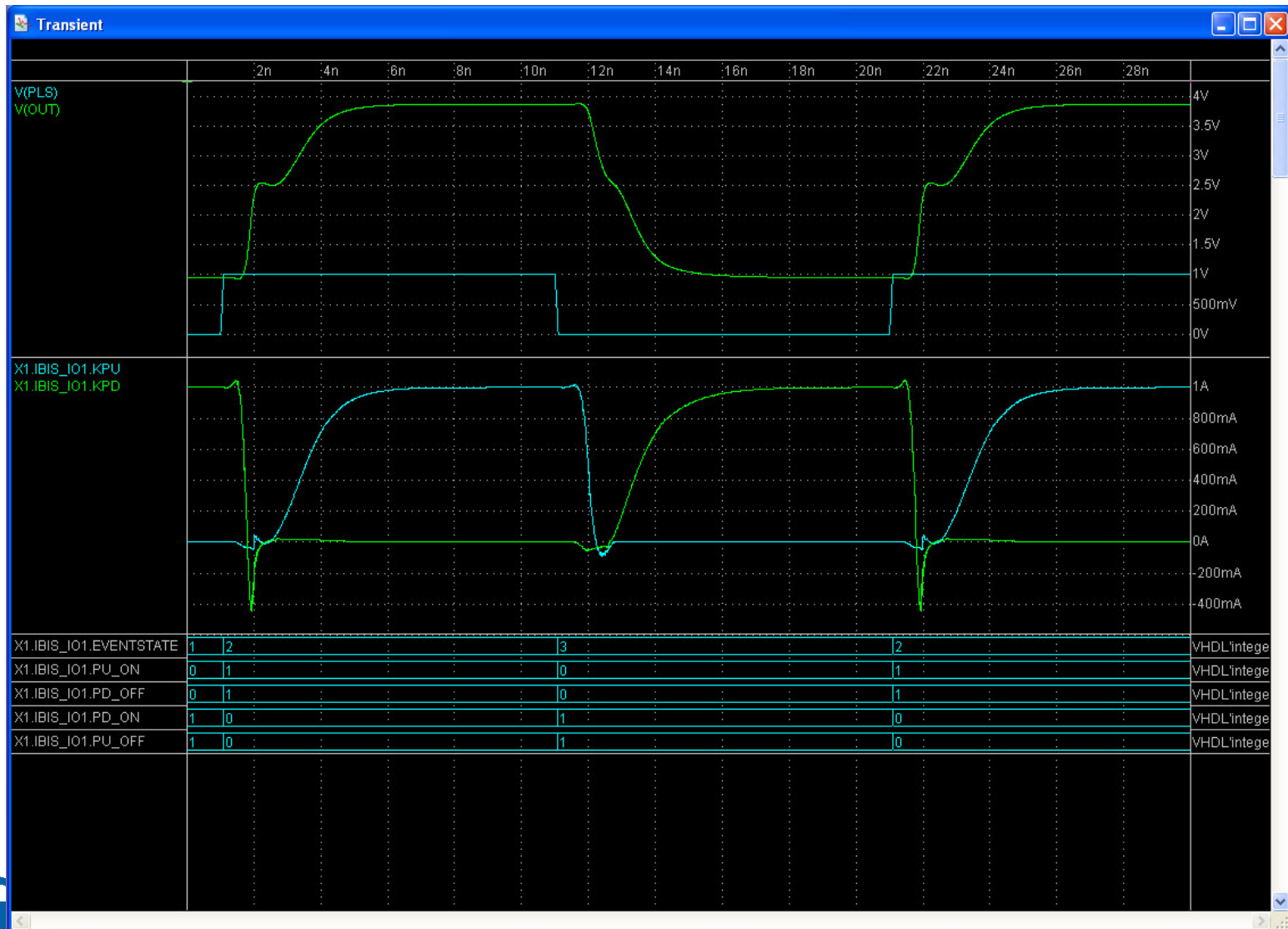


IBIS_IO buffer example - waveforms



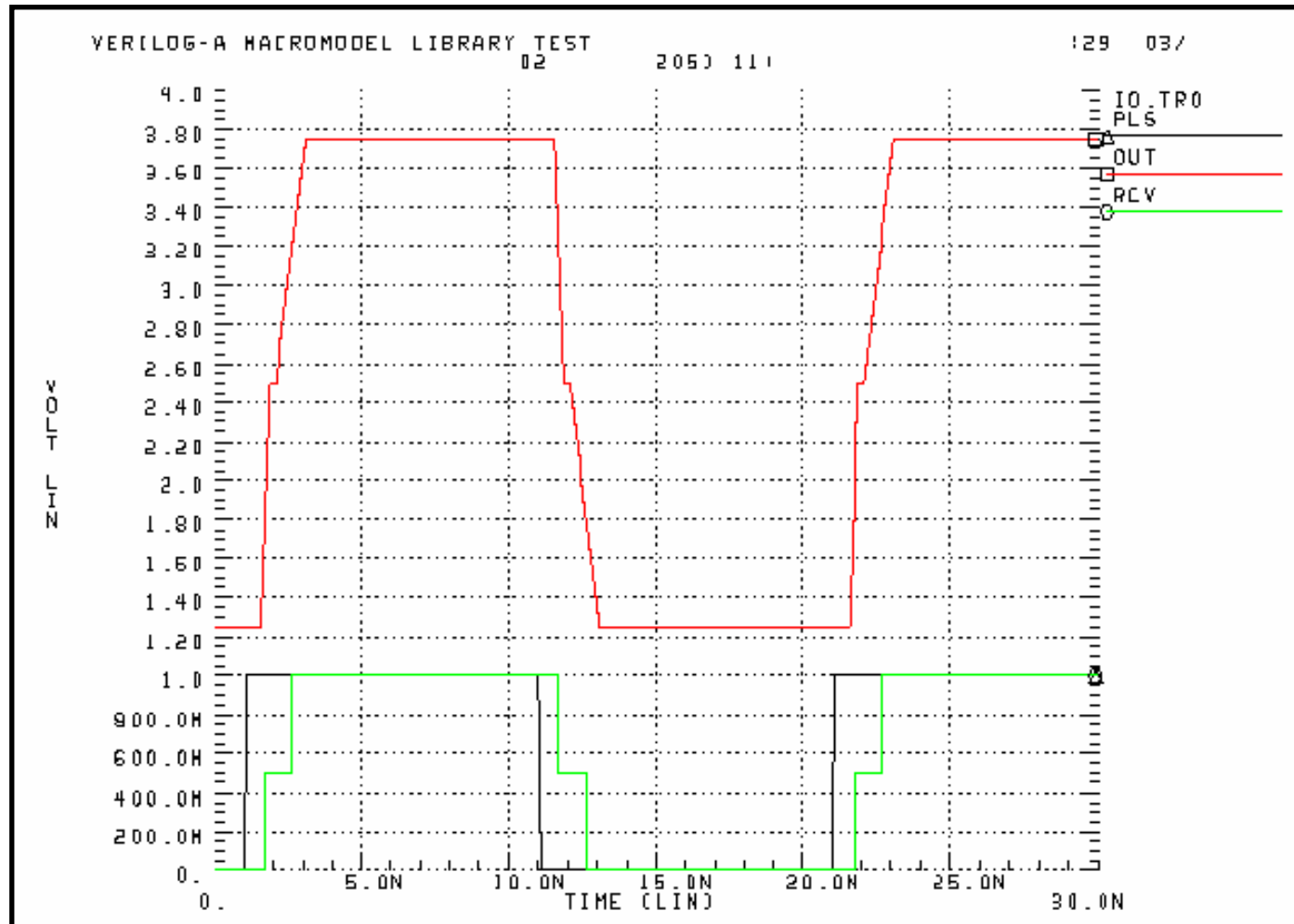


IBIS_IO using real IBIS data



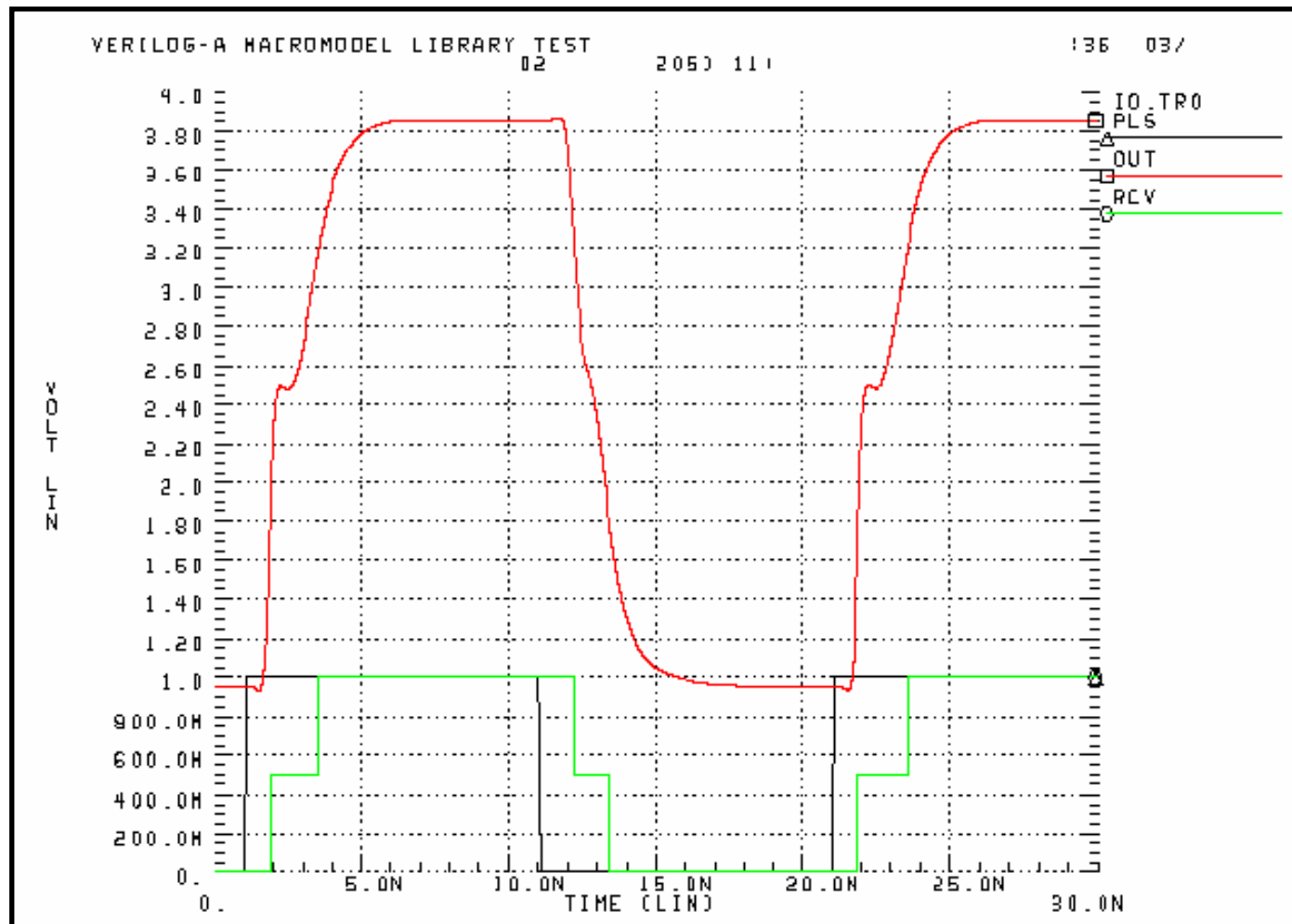


Verilog-A(MS) waveforms of IBIS_IO





IBIS_IO using real IBIS data

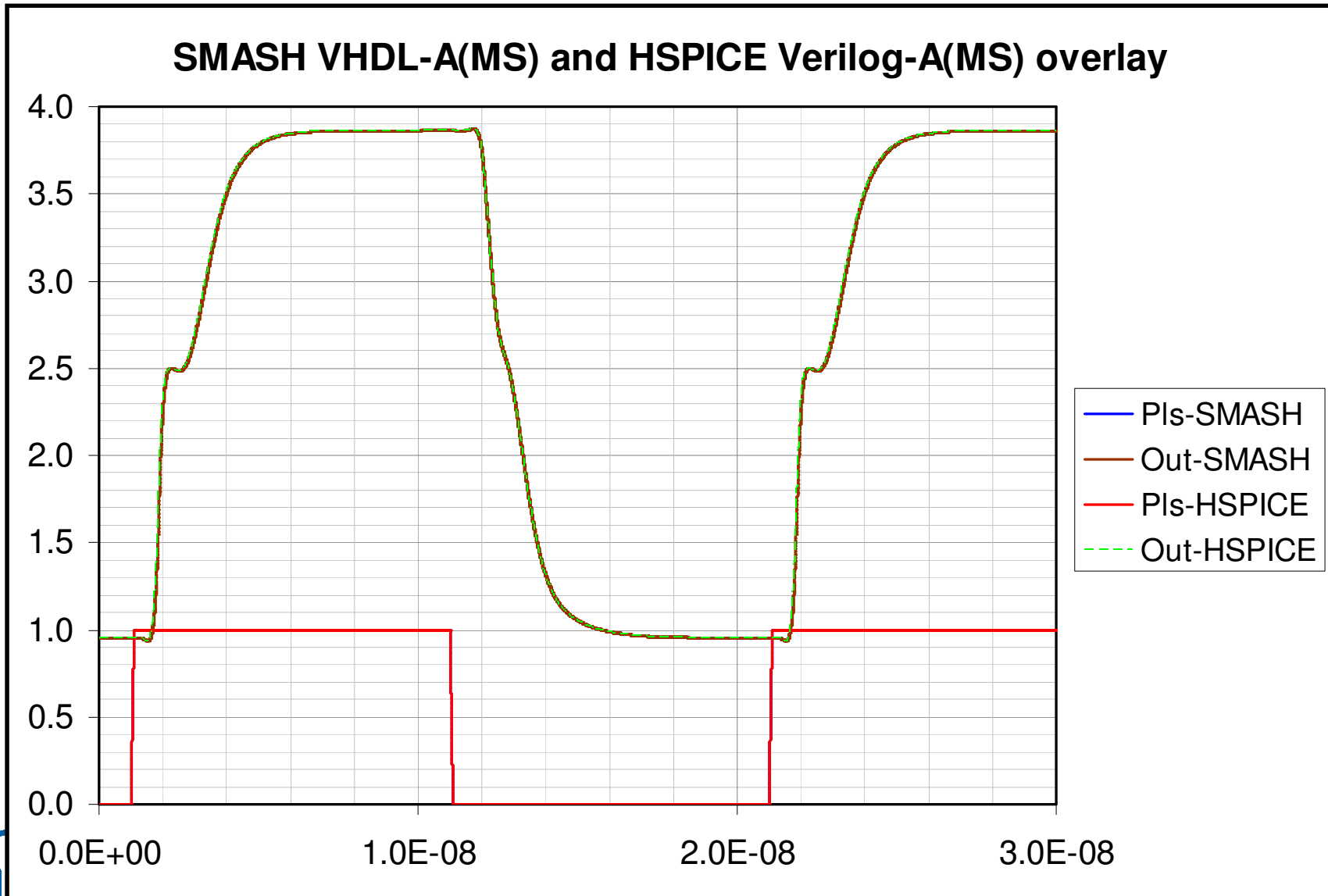


*Other brands and names are the property of their respective owners





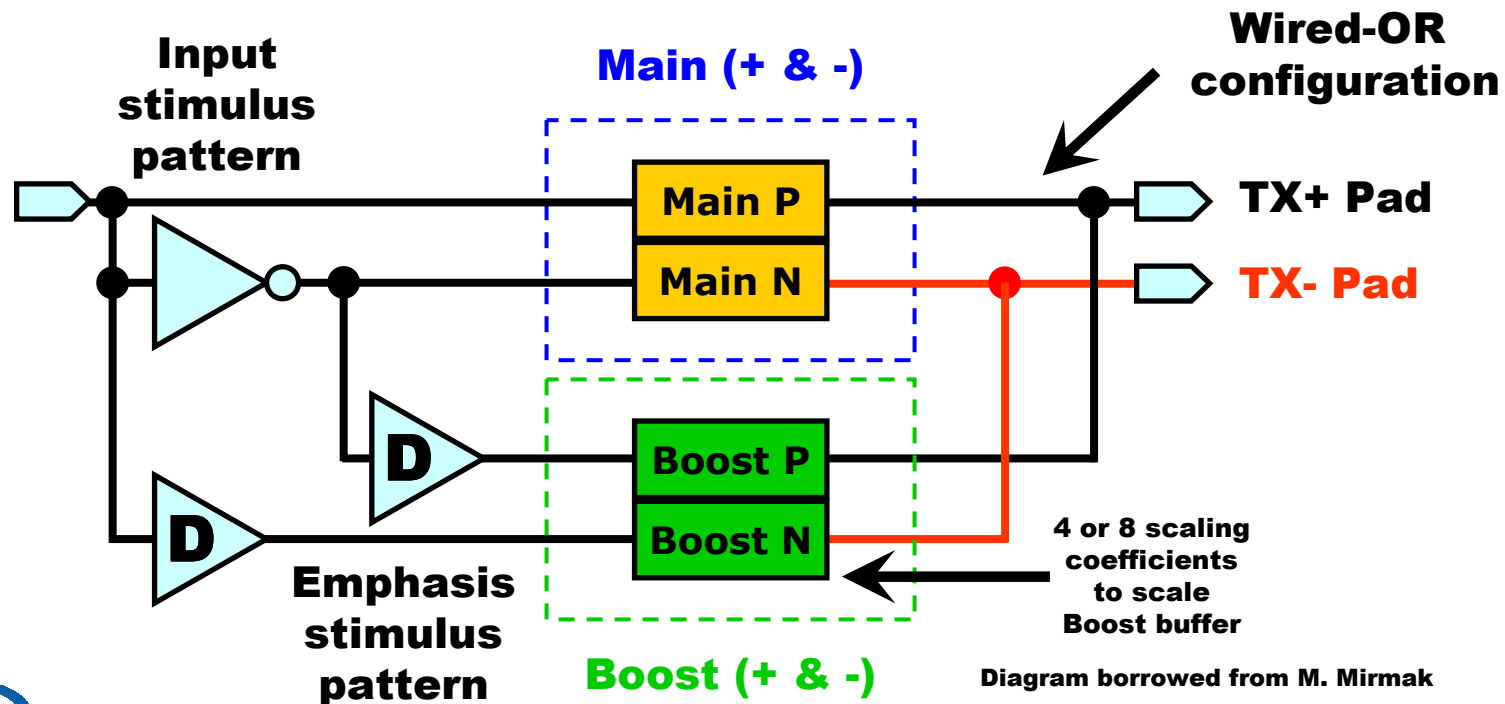
The results are identical!





Pre-emphasis buffer - block diagram

- A differential pre/de-emphasis buffer
 - a circuit netlist serves as the macro model, instantiating
 - four Verilog-A or VHDL-AMS IBIS I/O buffer models,
 - an inverter,
 - two ideal delays, and
 - eight current sources to scale the Boost buffer's current





Pre-emphasis buffer - test bench

```
Test Verilog-A "Macro Model Template" in HSPICE
*****
.TRAN 5.0ps 150.0ns
.OPTIONS POST=1 POST_VERSION=9007 PROBE
.hdl ".\PreDeMacro.va"
*****
.PROBE TRAN
+ Pls = V(Pls)
+ OutP = V(Out_p)
+ OutN = V(Out_n)
*****
Vvcc    Vcc    0  DC= 5.0
* This source represents a 111000111000 pattern
Vpls    Pls    0  PULSE (1.0  0.0  1.0ns  1.0ps  1.0ps  30.0ns  60.0ns)
*****
X1      Pls    Out_p  Out_n  Vcc  Vcc  0    0    Vcc    PreDeMacro
*      In     Out_p  Out_n  PC   PU   PD   GC   En
*
Rld1    Out_p  Vtt    R= 50.0
Rld2    Out_n  Vtt    R= 50.0
*****
.END
*****
```





Pre-emphasis buffer - macro model

```
`include "constants.vams"
`include "disciplines.vams"
`include ".\AMS_files\IBIS_macro_library.va"

module PreDeMacro (InD, IOp, IOn, PCref, PUref, PDref, GCref, EnD);
    input      InD, EnD;
    electrical InD, EnD;

    inout      IOp, IOn, PCref, PUref, PDref, GCref;
    electrical IOp, IOn, PCref, PUref, PDref, GCref;

    electrical InNM,      InPB,      InNB,      Dref;
    electrical PUrefPB,   PDrefPB,   PCrefPB,   GCrefPB;
    electrical PUrefNB,   PDrefNB,   PCrefNB,   GCrefNB;
    electrical RcvPM,     RcvNM,     RcvPB,     RcvNB;

    parameter real BitDelay  = 10.0e-9;
    parameter real ScaleBoost = 0.5;
    //-----
    // These lines show how to pass IV and Vt tables into the IBIS_IO model
    //-----
    `include "lab_1_io50v_t.dat"

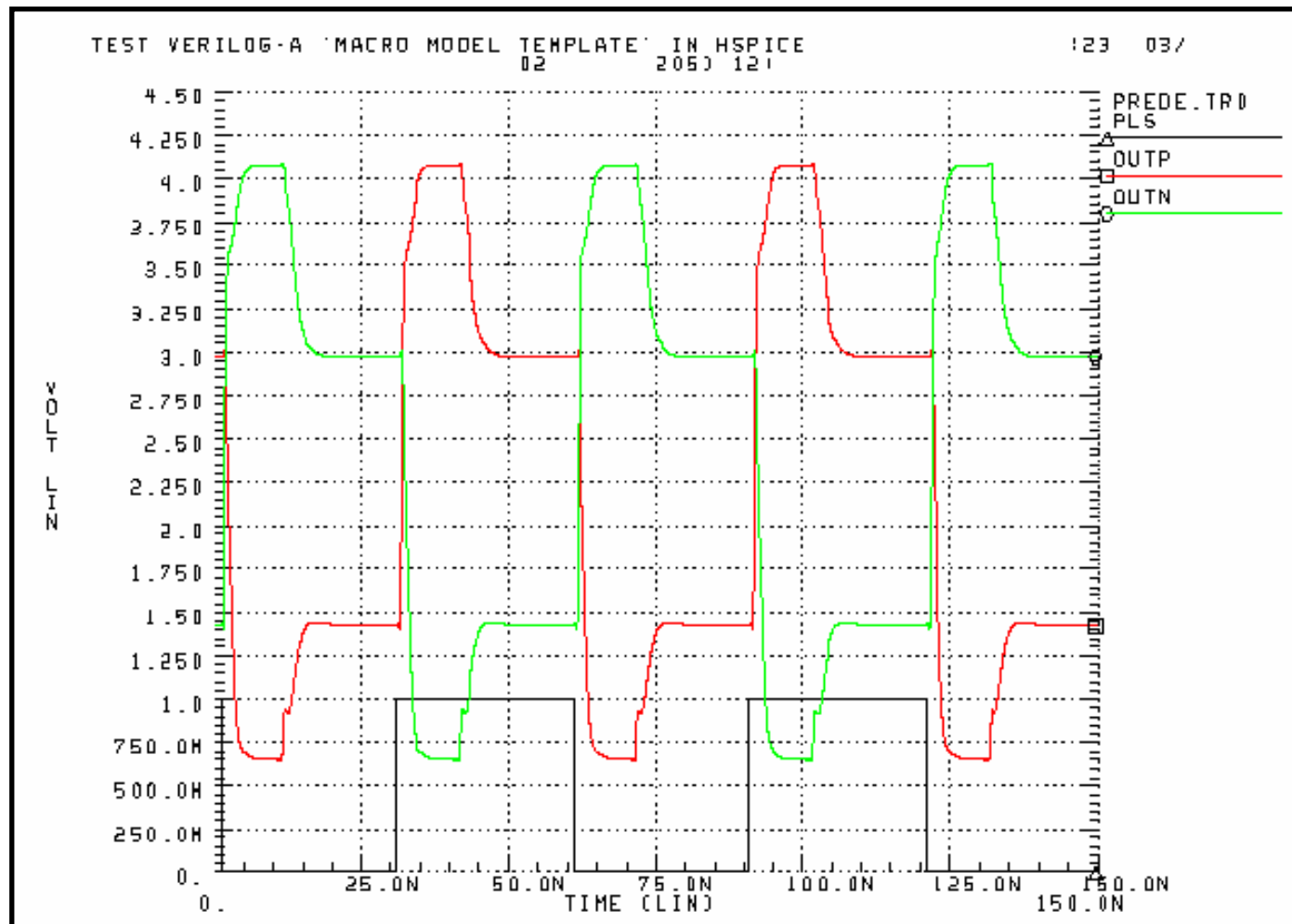
    IBIS_IO #(`IO_data) PosM  (PUref, PDref, IOp, InD,  EnD, RcvPM, PCref, GCref);
    IBIS_IO #(`IO_data) NegM  (PUref, PDref, IOn, InNM, EnD, RcvNM, PCref, GCref);
    IBIS_IO #(`IO_data, .kI_pc(ScaleBoost), .kI_pu(ScaleBoost), \
                      .kI_pd(ScaleBoost), .kI_gc(ScaleBoost)) \
                      PosB  (PUref, PDref, IOp, InPB, EnD, RcvPB, PCref, GCref);
    IBIS_IO #(`IO_data, .kI_pc(ScaleBoost), .kI_pu(ScaleBoost), \
                      .kI_pd(ScaleBoost), .kI_gc(ScaleBoost)) \
                      NegB  (PUref, PDref, IOn, InNB, EnD, RcvNB, PCref, GCref);
    //-----
    IBIS_V      #(.Vdc(1.0))      Dig1  (Dref, PDref);
    IBIS_VCVS    #(.Vdc(1.0))      Inv1  (Dref, InNM, InD,  PDref);
    IBIS_VCVS_DELAY #(.TD(BitDelay)) Dly1  (InNB, PDref, InD,  PDref);
    IBIS_VCVS_DELAY #(.TD(BitDelay)) Dly2  (InPB, PDref, InNM, PDref);

endmodule
```





Pre-emphasis buffer - waveforms





Wrap up

- Links to the two test suites including the most current versions of the macro model library

http://www.eda.org/pub/ibis/macromodel_wip/element_lib/Verilog-A_element_library_HSPICE_test.zip

http://www.eda.org/pub/ibis/macromodel_wip/element_lib/VHDL-AMS_element_library_SMASH_test.zip

- Please try it out and provide feedback, that is the only way this effort can be made useful!
- Lots of capabilities and features could still be added to the library, but we need to know what is needed, and what is practical, etc...
- In case you need help to find SMASH...

http://www.dolphin-integration.com/medal/smash/smash_download.html



*Other brands and names are the property of their respective owners

