SYNOPSYS®
Predictable Success

Formality D-2010.03 Update Training

Joe Bosia

Formality CAE

March 2010

Agenda

- D-2010.03 has the following Enhancements
  - Language Support
  - Graphical User Interface
  - Reporting
  - UPF Support
  - Verification
  - Debugging of Hard Verifications
  - fm_shell

Goal is to become familiar with these updates

Agenda (cont.)

- Enhanced Language Support
  - Lockstep SystemVerilog Features
    - Arrays of Interfaces
    - Packed Arrays of Enum Types
    - Var Keyword
    - Improved Usability of Interfaces

Agenda (cont.)

- Enhanced Graphical User Interface
  - Extended support for the Copy Name command in the GUI
  - Pruning correlation between cone schematic and the pattern window
  - Hypertext links to GUI man pages

Agenda (cont.)

- Enhanced Reporting
  - New Command report_setup_status to report the status of user setup
  - Enhanced reporting of simulation/synthesis mismatch errors and warnings

Agenda (cont.)

- Enhanced UPF Support
  - .lib/.db support for clock free save/restore retention models
  - Dynamic library update
  - Additional set_retention option
  - Power On Verification mode
  - New UPF interpretation messages

Agenda (cont.)

- Enhanced Verification
  - Distributed processing using ssh
  - Improved Probing Capability
  - Default timeout of 36 hours (with auto saved session)
  - Improved Verification of Large XOR chains
  - New “synthesis” mode for handling undriven signals

Agenda (cont.)

- Enhanced Debugging of Hard Verifications
  - analyze_points now produces Design Compiler “set_verification_priority” recommendation

Agenda (cont.)

- Enhanced fm_shell
  - User specified location for FM_WORK and other temporary files/directories

Enhanced Language Support

Lockstep SystemVerilog Features

Arrays of Interfaces

```
interface try_i;
logic [1:0] send; logic [1:0] receive;
endinterface

module top(input wire [7:0] di1, output wire [7:0] do1);
try_i t_arr[3:0](); //array of interfaces

// Pass in array of interfaces to array of instances
other_sender s2[3:0] (t_arr, di1, do1);
endmodule

module other_sender (try_i t_arr, input wire [1:0] in, output wire [1:0]
out);
endmodule
```

Enhanced Language Support

Lockstep SystemVerilog Features

Packed Arrays of Enum Types

```
enum [1:0] {FALSE, TRUE} [3:0] v;
```

Enhanced Language Support
Lockstep SystemVerilog Features
Var Keyword

```
var logic [1:0] v;
```

- One example is to help distinguish nets from variables

Enhanced Language Support

Lockstep SystemVerilog Features

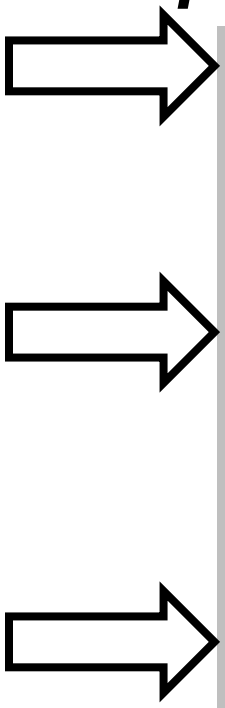
Improved Usability of Interfaces

- Formality now allows interface references before the interface has been read or defined. This lifts the requirement that all files be read at the same time.

Enhanced Language Support

Lockstep SystemVerilog Features

Improved Usability of Interfaces



```
interface try_i;
logic [7:0] send;
logic [7:0] receive;
modport sender(.send(send), .receive(receive));
endinterface

module sender(try_i.sender try,
input logic [7:0] data_in,
output logic [7:0] data_out);
assign data_out= try.receive;
assign try.send= data_in;
endmodule

module top(input wire [7:0] di1, di2, output wire [7:0] do1, do2);
try_it();
sender s (t.sender, di1, do1);
endmodule
```

Enhanced Language Support

Lockstep SystemVerilog Features

Improved Usability of Interfaces

Prior to D-2010.03, all files had to be read at same time:

read_sverilog {interface.sv top_module.sv sender.sv}

Starting with 2010.03, files can be read individually:

read_sverilog{top_module.sv}
read_sverilog{sender.sv}
read_sverilog{interface.sv}

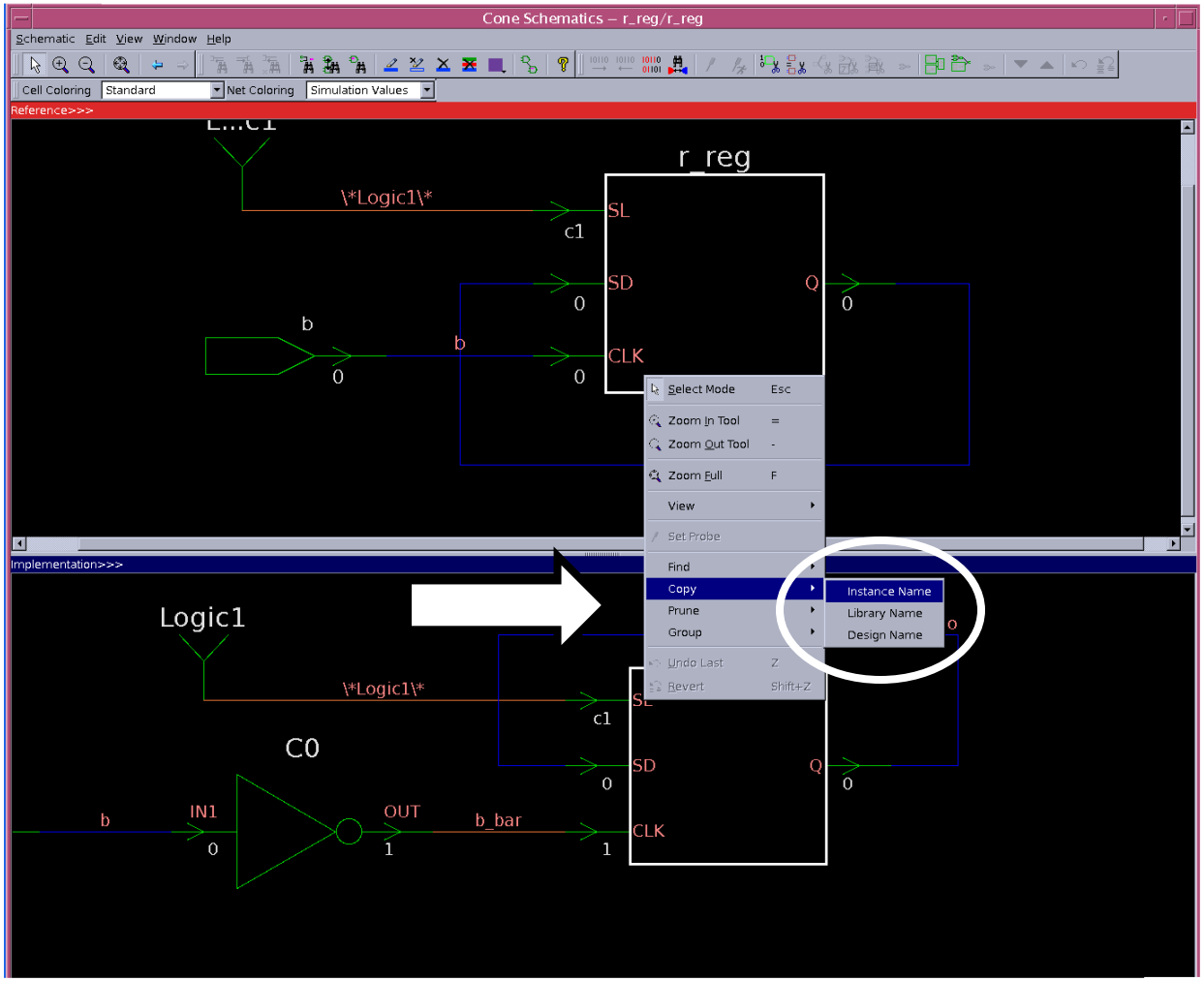
```
interface try_i;
logic [7:0] send;
logic [7:0] receive;
modport sender(.send(send), .receive(receive));
endinterface

module sender(try_i.sender try,
input logic [7:0] data_in,
output logic [7:0] data_out);
assign data_out= try.receive;
assign try.send= data_in;
endmodule

module top(input wire [7:0] di1, di2, output wire [7:0] do1, do2);
try_it();
sender s (t.sender, di1, do1);
endmodule
```

Enhanced Graphical User Interface
Extended support for the Copy Name command

- Support for copying names has been extended in Formality.
  - There are new options to copy design names and library names in the context menu of a selected cell in a schematic.
  - The context menu has been consolidated with sub-menus to make it easier to navigate.

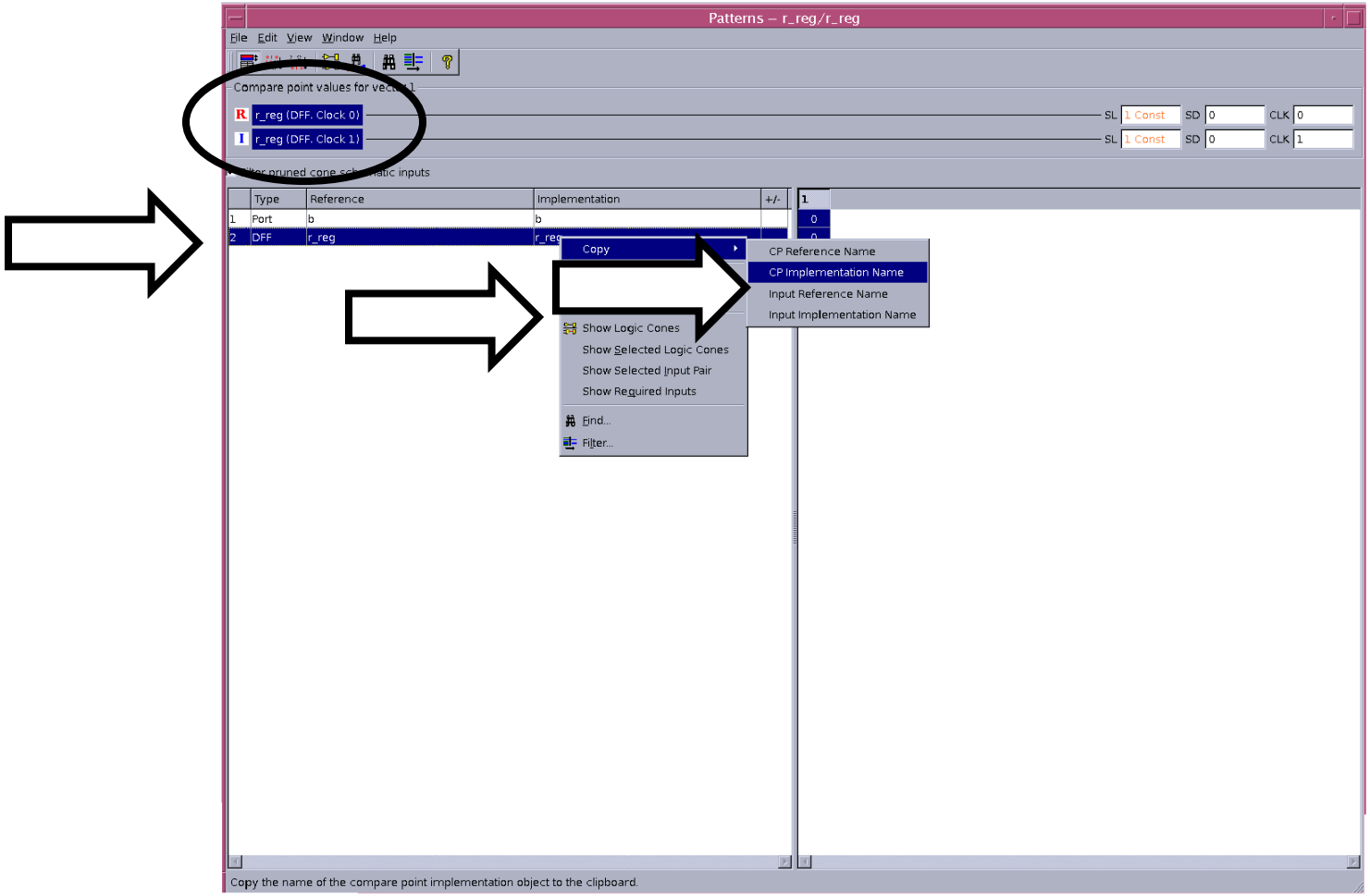
Enhanced Graphical User Interface

Extended support for the Copy Name command



Enhanced Graphical User Interface

Extended support for the Copy Name command



Enhanced Graphical User Interface
Pruning correlation between cone schematic and the pattern window

When a cone schematic is pruned by commands like:

`remove_subcone, isolate_subcone`, etc.)

cone inputs may also be pruned away.

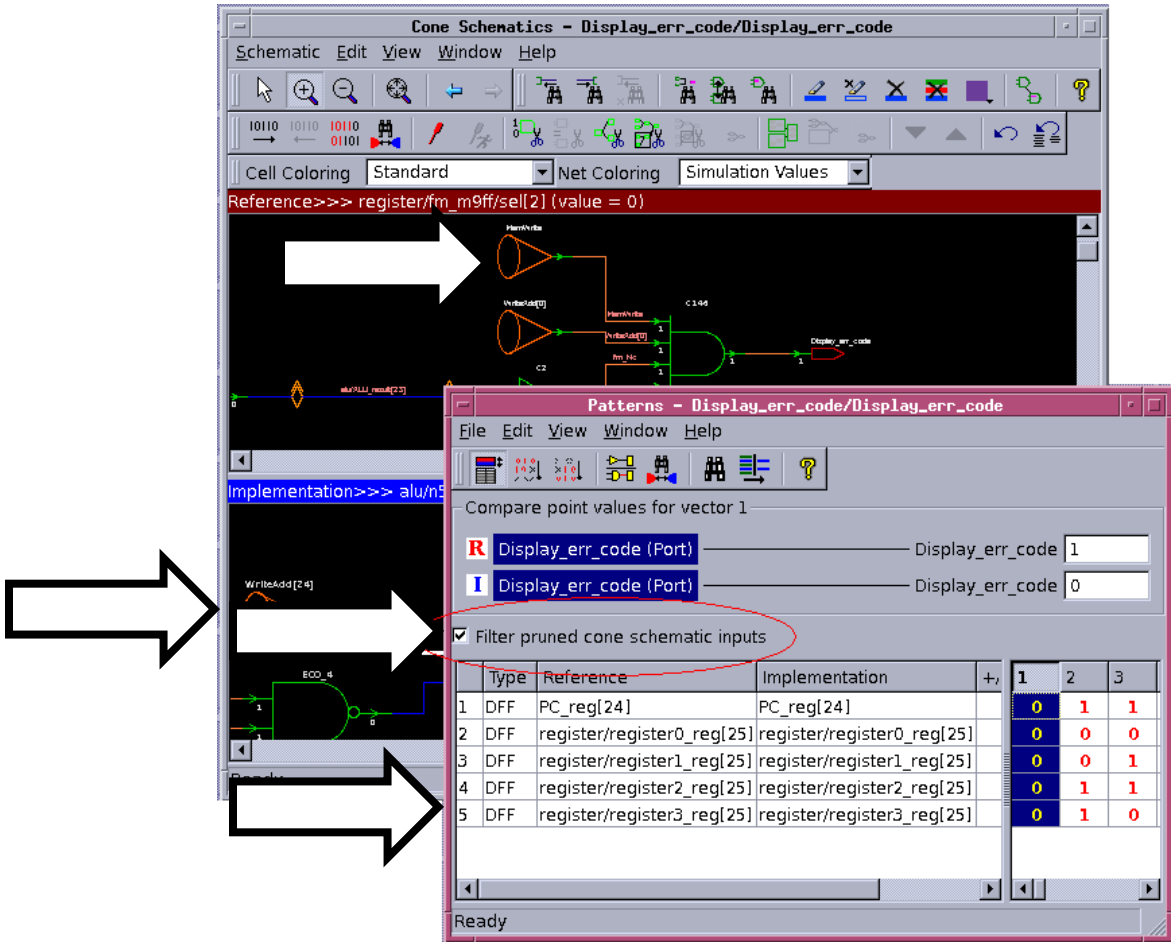
The cone inputs that have been pruned away can now be filtered in the Pattern Window for easier debugging.

Notes:

If a cone input is only removed from one design, it will still appear in the pattern window. When it is removed from both designs then it may be filtered from the pattern window as well.

Enhanced Graphical User Interface

Pruning correlation between cone schematic and the pattern window



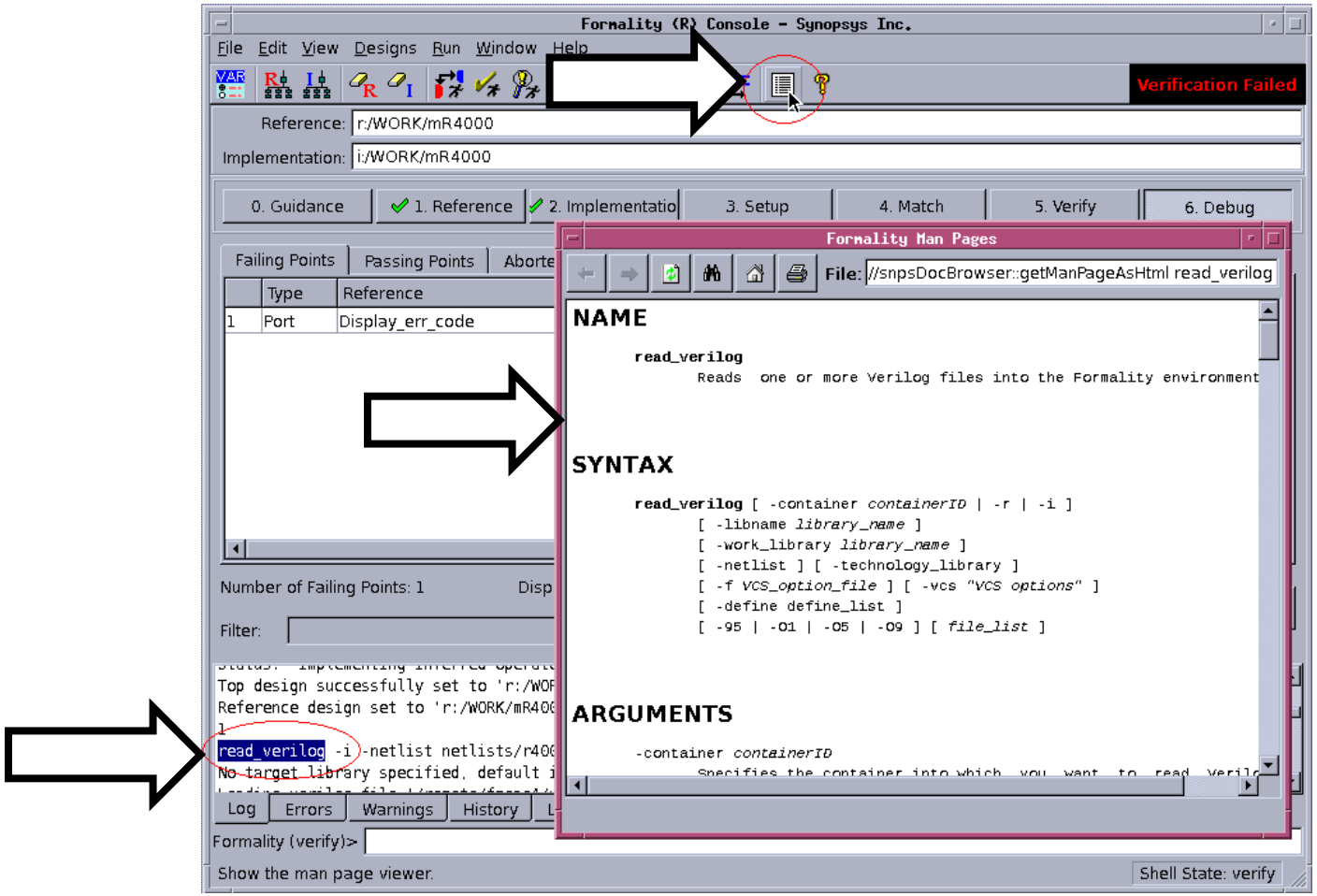
Enhanced Graphical User Interface

Hypertext links to GUI man pages

- Man pages are now linked for text that is selected in the GUI.
- If a command, variable or message ID appears in the transcript window, selecting it and clicking the Man Page Viewer will bring up that specific man page.

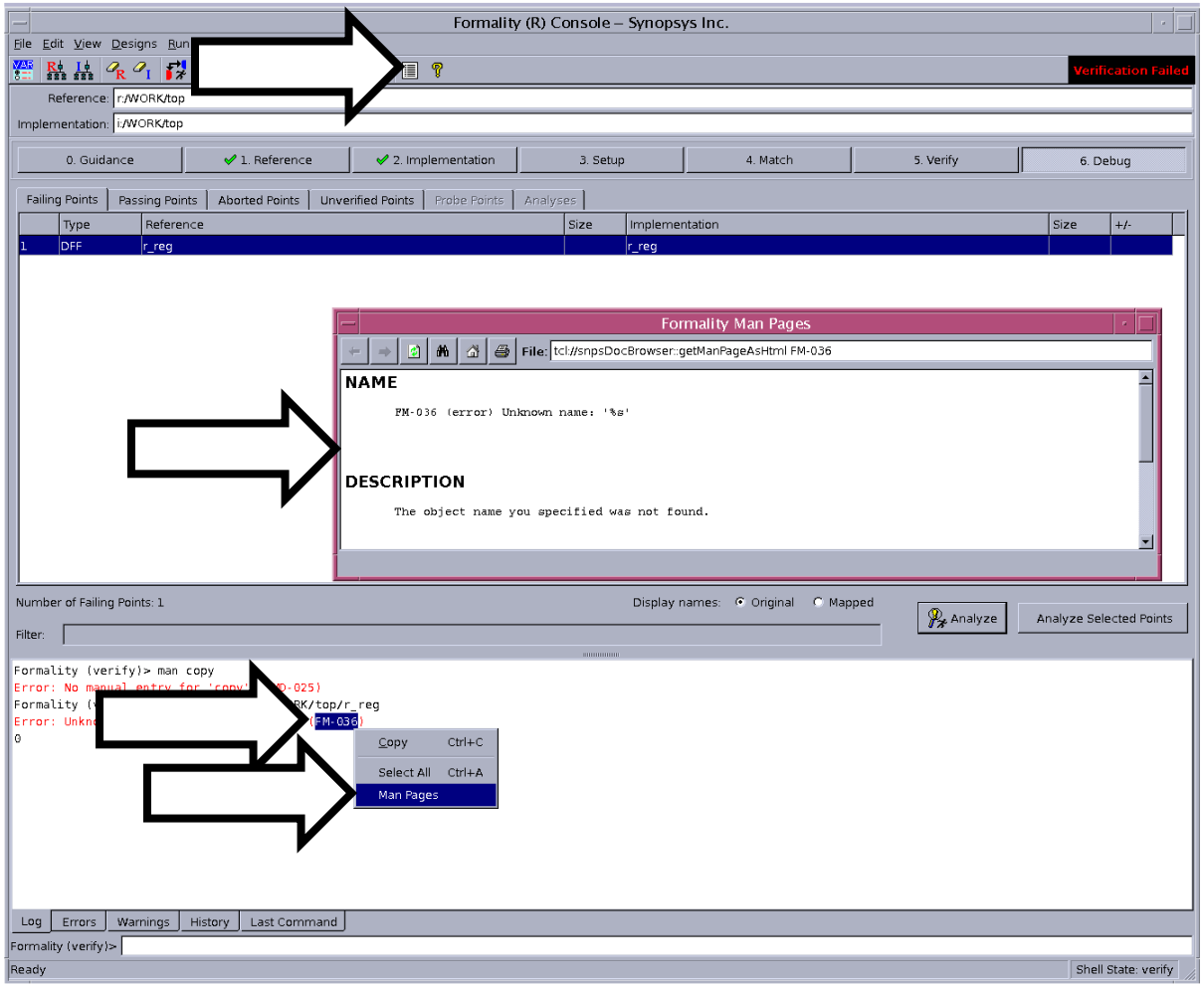
Enhanced Graphical User Interface

Hypertext links to GUI man pages



Enhanced Graphical User Interface

Hypertext links to GUI man pages



Enhanced Reporting

New Command report_setup_status

- Allows users to check critical design setup before running the commands match or verify
  - Check and complete any missing design setup before proceeding with the more time consuming commands of verification.
- The command can be run anytime after reading and linking both reference and implementation containers

Enhanced Reporting

New Command report_setup_status

- Produces a report in summary format
  - Design statistics
  - Design read warning messages
  - User specified setup.
- The command options will be used to display individual report sections. If no option is specified, a consolidated output will be displayed which contains all the sections.

Command syntax:

report_setup_status [-design_info] [-hdl_read_messages] [-commands]

-design_info (reports the set_* commands which define the objects)

-hdl_read_messages (warning and informational messages encountered during design read and link)

-commands (user specified setup)

Enhanced Reporting

New Command report_setup_status

```
fm_shell (setup)> report_setup_status

#### Design Information ####
  # Design Settings #
    set_top reference design:  r:/WORK/top
    set_top implementation design:  i:/WORK/top
    set_reference_design:  r:/WORK/reference
    set _implementation_design:  i:/WORK/impl

  # Design Statistics Ref (Imp) #
    Ports: 1200 (1203)
    Registers: 2420 (2471)
    Black boxes: 8 (8)
      - Unresolved modules: 1 (1)
      - User specified: 2 (2)
    Undriven nets: 41 (41)
    Multiply-driven nets: 5 (5)
```

```
#### HDL Read Message Summary ####
Message ID:       Occurrences:  Ref(Imp)

FMR_ELAB-034      1(0)
FMR_ELAB-058      2(0)


#### User Specified Setup ####
Command Name:          Result:  Ref(Imp)

set_black_box          1 (1)
set_clock              2 (2)
set_compare_rule       3 (3)
set_constant           4 (4)
set_constraint         5 (5)
set_cutpoint           6 (6)
set_dont_verify_point  8 (8)
set_equivalence        9 (9)
set_factor_point       10 (10)
set_inv_push           11 (11)
set_user_match         13 (13)
```

Enhanced Reporting

New Command report_setup_status

- This report will display only the user specified settings
  - Any changes performed by using "**synopsys_auto_setup**" or through SVF processing will not be reflected in the report.

Enhanced Reporting
simulation/synthesis messages

- Formality will now list all of the error-IDs of any unsuppressed RTL simulation/synthesis mismatches using only one error message.

```
Error: Unsuppressed RTL interpretation message(s)
"FMR_ELAB-117 FMR_ELAB-149 FMR_ELAB-11" were
produced during link. (FM-262)
```

Enhanced UPF Support

.lib/.db support for retention register modeling

- Library Compiler 2010.03 allows new syntax to more accurately model retention register behavior
- The supported retention clock-free save/restore (and other) retention register models can now be modeled in .lib
- Formality can read this new syntax
- You can now use a .db model to create a model for synthesis and verification
- To use a Verilog model for verification see: https://solvnet.synopsys.com/retrieve/024106.html

Enhanced UPF Support

Dynamic library update

- Formality can modify .db libraries to add power aware functionality for dynamic library update/modification.
- DC/ICC provides the required mapping files for Formality to load.
- To allow Formality to find the correct mapping file to apply to the appropriate .db library use the variable:

```
library_pg_file_pattern
```

Enhanced UPF Support
Dynamic library update

- In current dir:
  - `set library_pg_file_pattern "script.pg.tcl"`
    `read_db vendor_library_125c.db`
- At a specific location
  - `set library_pg_file_pattern`
    `"/home/bhatt/onthefly/script.pg.tcl"`
    `read_db vendor_library_125c.db`
- In same dir with .db file:
  - `set library_pg_file_pattern "__DIR__/script.pg.tcl"`
    `read_db vendor_library_125c.db`
- With the .db filename in the script name
  - `set library_pg_file_pattern`
    `"__DIR__/sub1/__FILE__.pg.tcl"`
    `read_db vendor_library_125c.db`

Enhanced UPF Support
Additional set_retention option

- **set_retention –noretention**
  - Allows the user to specify specific instances that should not be retained.
  - Can be used when a design has instantiated clock gating (or other sequential cells) that have not been retained in the implementation design.
  - Should be specified in the upf which is supplied to simulation and synthesis so that the entire flow is aware of which cells should not be retained.
- Alternatively, to tell Formality to not retain instantiated clock gate cells without modifying the upf and when using .db libraries, use:

```
set upf_use_additional_db_attributes true
```

Enhanced UPF Support
Power On Verification mode

- Verify only the power state where all UPF supplies are on

```
set verification_force_upf_supplies_on true
```

- This is considered a Partial Verification
  - Only one of the supply combinations is verified
  - No checks to see if it is a legal power state
- Faster
  - Checks to make sure intermediate netlists from Design Complier have no logic errors when power is on.
- Final netlist must be verified with this variable value of false to get a complete verification

Enhanced UPF Support

New UPF interpretation messages

- The **load_upf** command now issues several additional informational messages

- ```
  Info:  load_upf: loading UPF, implementing all UPF constructs.
  Info:  load_upf: created supply constraints based on defined
  PSTs.
  …
  Info:  load_upf: implemented 9538 retention registers and 316
  isolation cells.
  ```

- ```
  Info:  load_upf: loading UPF Generated by Design
  Compiler(C-2009.06-SP4) on Mon Feb  1 13:46:00 2010,
  implementing supply network and connecting retention and
  isolation supplies.
  Info:  load_upf: created supply constraints based on
  defined PSTs.
  ```

Enhanced Verification

General Improvements

- Enhanced verification of Dividers
- Support of new Multiplier Architectures
- Support of Instantiated DW FP blocks
- Improved UPF Low Power verification performance
- Improved support for DC register merging
- Improved loop breaking

Enhanced Verification

Distributed processing using ssh

- Distributed verification now allows the user to specify what remote shell command to use to start the slave processes.
- Many secure networks only allow use of ssh to create remote shells.
- This new feature allows customers on these networks to successfully run distributed verification.

Enhanced Verification

Distributed processing using ssh

set_host_options

Sets distributed processing options.

SYNTAX

set_host_options

-submit_command [rsh | ssh | remsh]

ARGUMENTS

-submit_command [rsh | ssh | remsh]

Specifies which remote shell to use to launch remote servers.

rsh is the default.

Enhanced Verification

Distributed processing using ssh

fm_shell> **set_host_options -submit_command ssh**

Changes the remote shell command to be used for distributed processing from the default rsh to ssh.

Enhanced Verification
improved Probing Capabilities

- Formality introduced probing functionality in C-2009.06
  - Helps in debugging hard or failing verifications.
  - The probing feature allows you to select a reference logic cone net and an implementation logic cone net, and then determine if the logic is equivalent up to those probe points.

Enhanced Verification
improved Probing Capabilities

New to Probing in D-2010.03

- New option "**-inverted**" for command “**set_probe_points**”
- New option “**-status**” for command “**report_probe_status**”
- New support of wild card “*” for command “**remove_probe_points**”
- New support of 1-to-N matching
- Improved error and warning messaging to include ID tags

Enhanced Verification
improved Probing Capabilities

- New option “**-inverted**" for command “**set_probe_points**”
- This option allows you to specify that a pair of probe points has an inverted relationship.

Example:

```
fm_shell (verify)> set_probe_points –inverted
   r:/WORK/tvb/i_core/fm_N95e[1]
   i:/WORK/tvb/apb_reg_1_/next_state
Set user probe between 'r:/WORK/tvb/i_core/fm_N95e[1]' and
   'i:/WORK/tvb/apb_reg_1_/next_state'
```

Enhanced Verification
improved Probing Capabilities

- New option "`-status`" for command "`report_probe_status`"
- "`-status`" has filters "`pass | fail | abort | notrun`"
- The filter options are used to filter the probe report based on probe verification status.

Example:

```
fm_shell (verify)> report_probe_status -status pass
        Ref  PASS      r:/WORK/tvb/A[7]
        Impl PASS      i:/WORK/tvb/ina[7]
```

Enhanced Verification
improved Probing Capabilities

- New support of wild card “*” for command “**remove_probe_points**”
- This enhancement allows you to delete probe nets with similarity in names, such as nets comprising a bus.

Example:

fm_shell (verify)> **remove_probe_points** r:/WORK/tvb/i_core/fm_N94e[*]
Removed probe for 'r:/WORK/tvb/i_core/fm_N94e[*]'

fm_shell (verify)> **remove_probe_points** r:/WORK/tvb/i_core/fm_abc*
Removed probe for 'r:/WORK/tvb/i_core/fm_abc*'

Enhanced Verification
improved Probing Capabilities

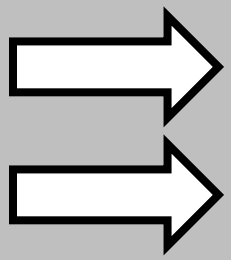
- New support of 1-to-N matching
- This enhancement allows you to specify probe points that include one reference net matched with several implementation nets.

Example:

```
fm_shell (verify)> set_probe_points r:/WORK/tvb/i_core/fm_sig
    i:/WORK/tvb/apb_reg_1/next_state
Set user probe between 'r:/WORK/tvb/i_core/fm_sig' and
    'i:/WORK/tvb/apb_reg_1/next_state'

fm_shell (verify)> set_probe_points r:/WORK/tvb/i_core/fm_sig
    i:/WORK/tvb/apb_reg_2/next_state
Set user probe between 'r:/WORK/tvb/i_core/fm_sig' and
    'i:/WORK/tvb/apb_reg_2/next_state'

fm_shell (verify)> set_probe_points r:/WORK/tvb/i_core/fm_sig
    i:/WORK/tvb/apb_reg_3/next_state
Set user probe between 'r:/WORK/tvb/i_core/fm_sig' and
    'i:/WORK/tvb/apb_reg_3/next_state'
```

Enhanced Verification
improved Probing Capabilities

- Improved error and warning messaging to include ID tags
- These improvements will allow you to look up error IDs in the man pages for help in resolving setup issues.



Example:

```
fm_shell (verify)> set_probe_points r:/WORK/tv80s/i_tv80_core/fm_N95e[1]
      i:/WORK/tv80s/Ap_reg_1_/next_state
Error: No probe set. Please remove existing probes on these objects. (FM-526)
0
fm_shell (verify)> man FM-526
messages                    N:  Messages                Command Reference

NAME
      FM-526  (error)  No  probe  set. Please remove existing probes on these
      objects.

DESCRIPTION
      No probe is set by 'set_probe_points' command on the specified net pair
      because  another  probe  definition already exists on them. Please also
      note that Formality's matching allows only 1-to-N pairing of probes.

WHAT NEXT
      Specify a fresh pair of nets as probes.
```

Enhanced Verification

Default timeout of 36 hours (with auto saved session)

- New variable `verification_timeout_auto_session` has been introduced to support this capability.
  - Default is true
  - Set to false if you do not wish to have auto saving of session files.
  - Name of file

    `formality_timeout_session.fss`
- Default of `verification_timeout_limit` is `36:0:0`

Enhanced Verification

Improved Verification of large XOR chains

- Designs that contain large XOR chains, such as CRCs, ECCs, and parity trees are more easily verified.
- Use `set_verification_priority` in Design Compiler on the block containing the XOR logic and DC will preserve the hierarchy.
- If the logic is in a function DC will issue guidance (`guide_group_function`) as necessary.
  - Formality uses this guidance to create the appropriate hierarchy, which helps the verification complete.

Enhanced Verification

New "synthesis" mode for handling undriven signals

- Results in fewer failing verifications due to incorrect values on undriven signals
- When you `set synopsys_auto_setup true` it will also
  `set verification_set_undriven_signals synthesis`
- Treats undriven signals:
  - In the reference the same as Design Compiler.
  - In the implementation as binary.
- Will still detect undriven signals in the implementation that cause a difference

Enhanced Debugging of Hard Verifications

analyze_points now produces Design Compiler "set_verification_priority" recommendation

NOTE: This change was implemented in Formality 2009.06-SP3. It is mentioned here because it was never formally part of an update training.

Enhanced Debugging of Hard Verifications

analyze_points now produces Design Compiler "set_verification_priority" recommendation

Example:

```
Formality (verify)> analyze_points {r:/WORK/test/data_bl/o_reg[11]}

********************************** Analysis Results **********************************
Found 1 Rejected Datapath Guidance Module
------------------------------
These modules contain cells that may be related to
rejected datapath guidance.
------------------------------
r:/WORK/data_bl_0 in file /remote/user/test/data_bl.v
    Module with rejected datapath guidance on cell(s):
        r:/WORK/data_bl_0/add_376
        r:/WORK/data_bl_0/add_377
        r:/WORK/data_bl_0/add_378
        r:/WORK/data_bl_0/mult_404
     Affecting 1 compare point(s):
        r:/WORK/test/data_bl/o_reg[11]
    Use 'report_svf_operation { 1027 }' for more information.
    Try adding the following command(s) to your Design Compiler script:
        set_verification_priority -high [ get_designs { data_bl_0 } ]
```

Enhanced fm_shell

User specified location for FM_WORK and other temporary directories

- Invoke Formality with the new switch **`-work_path`** providing either an existing or currently nonexistent but valid Unix path as an argument.
- Allows users to improve traceability and help manage disk space more effectively

Enhanced fm_shell

User specified location for FM_WORK and other temporary directories

- Example Formality Invocation:

% **fm_shell -work_path /user/specified/path -f run.fms**

formality.log, fm_shell_commands.log, etc. get put under that directory path.

Summary

- D-2010.03 has the following Enhancements
  - Language Support
  - Graphical User Interface
  - Reporting
  - UPF Support
  - Verification
  - Debugging of Hard Verifications
  - fm_shell

Thank You

Predictable Success