SYNOPSYS®
Predictable Success

IC Compiler 2010.03 Incremental Training

Top-Level Design Closure

Agenda

1. Top-level design runtime, memory, and QoR improvements
2. Top-level design QoR improvement
3. ILM scenario flexibility
4. Retaining parasitics of unrouted nets in ILM
5. Checking and reporting enhancements

Top-Level Design Runtime, Memory, and QoR Improvements

- Runtime improvement
  - Multiple improvements in 2010.03 to reduce runtime
  - 40% out-of-the-box improvement versus 2009.06
- Memory improvement
  - Memory improvement is in line with the flat flow
- QoR is comparable to 2009.06

Agenda

1. Top-level design runtime, memory, and QoR improvements
2. Top-level design QoR improvement
3. ILM scenario flexibility
4. Retaining parasitics of unrouted nets in ILM
5. Checking and reporting enhancements

Top-Level Design QoR Improvement:

High-Fanout Synthesis Enhancement

- Overview
  - Blockage- and driver-aware synthesis during high-fanout synthesis
- UI
  - On by default
- User Benefit
  - The high-fanout synthesis result might have fewer buffers
  - Works for both automatic high-fanout synthesis (AHFS) and `create_buffer_tree`
- Flow recommendation
  - Use the default settings during high-fanout synthesis

```
set_ahfs_options -default
```

Agenda

1. Top-level design runtime, memory, and QoR improvements
2. Top-level design QoR improvement
3. ILM scenario flexibility
4. Retaining parasitics of unrouted nets in ILM
5. Checking and reporting enhancements

ILM Scenario Flexibility:

Introduction

- Overview
  - IC Compiler allows name and temperature mismatches between top-level and ILM scenarios in the multicorner-multimode flow.

- User Benefit
  - IC Compiler now supports:
    - Name mismatch between the top-level design and the ILM
    - Additional modes (does not have additional TLUPlus or temperature) at top level
    - Temperature mismatch between top and ILM
    - Non-multicorner-multimode top-level design and multicorner-multimode ILM

ILM Scenario Flexibility:
UI

- Specify the scenario mapping between the top-level design and the ILM by using the `select_block_scenario` command:

```
select_block_scenario
    [-scenarios top_scenarios]
    [-block_references list_design_names]
     -block_scenario block_scenario_name
```

- Set the `ilm_match_parasitics_temperature` variable when there is a temperature mismatch between the top-level design and ILM:

```
set ilm_match_parasitics_temperature closest
```

  The default value is **`exact`**.

ILM Scenario Flexibility:
UI

- User-specified scenario mapping is reported during the following commands: **`report_ilm`**, **`report_scenario`**, **`extract_rc`** and other analysis and optimization commands

```
Top-level and ILM scenario mapping:
-----------------------------------------------------------------
Top-level scenario        Reference ILM       ILM scenario
-----------------------------------------------------------------
top_scen1                 ilm_ref1             ilm_scen1
-----------------------------------------------------------------
(Only user-specified mappings are reported)
```

ILM Scenario Flexibility:

PVT Consistency Checks Between Top and ILM

- IC Compiler performs the following PVT consistency checks during the **`check_ilm`** command and also during analysis and optimization commands:

Condition	Message ID	Impact
Process (P) of top and ILM does not match	ILM-152 (Warning)	Flow continues.
Temperature (T) of top and ILM does not match. (**`ilm_match_parasitics_temperature`** = **`exact`**)	ILM-155 (Error)	The **`extract_rc`** command reports the ILM-149 or ILM-150 error and exits.
Temperature (T) of top and ILM does not match. (**`ilm_match_parasitics_temperature`** = **`closest`**)	ILM-153 (Warning)	If top has more operating temperatures used for any TLUPlus than the ILMs, the **`extract_rc`** command issues the ILM-149 error and exits. Otherwise, it issues ILM-154 and continues with closest match of top and ILM temperatures.

Agenda

1. Top-level design runtime, memory, and QoR improvements
2. Top-level design QoR improvement
3. ILM scenario flexibility
4. Retaining parasitics of unrouted nets in ILM
5. Checking and reporting enhancements

Retaining Unrouted Net Parasitics in ILM:
Introduction

- Overview
  - The `create_ilm` command stores parasitic information of unrouted nets in the ILM
- Benefit
  - Improved usability by allowing mix of ILMs and top-level design with different route and extraction status.

	Top level is Virtual Routed	Top level is Global Routed	Top level is Detail Routed
Virtual Routed ILM	✓	✓	✓
Global Routed ILM	✓	✓	✓
Detail Routed ILM	✓	✓	✓

Retaining Parasitics of Unrouted Nets in ILM

- UI
  - To retain the parasitics of unrouted nets in an ILM, create the ILM in IC Compiler version 2010.03.
    - Set the `ilm_retain_parasitics_for_nets` variable to `routed_nets` to retain only the routed nets' parasitics in ILM (as in IC Compiler versions prior to 2010.03)
  - Changes to the `create_ilm` command UI:

	Prior to 2010.03	2010.03
`keep_parasitics`	On-by-default for partially routed and routed blocks	Always on-by-default (ILM-92 is obsolete)
`include_side_load`	Side loads are retained for boundary and routed nets	Side loads are retained for all nets

  - The ILM-32 message is now a warning instead of an error.

Retaining Parasitics of Unrouted Nets in ILM

- Limitation
  - You can use only routed nested ILMs in the top-level design routing stages.

Agenda

1. Top-level design runtime, memory, and QoR improvements
2. Top-level design QoR improvement
3. ILM scenario flexibility
4. Retaining parasitics of unrouted nets in ILM
5. Checking and reporting enhancements

Checking and Reporting Enhancements:
Introduction

- Overview
  - You can now use the **check_ilm** command to perform checks on ILMs from the top level.
  - The functionalities of the **report_ilm** and **check_ilm** commands are enhanced.
  - New attributes to mark ILM objects in the reports generated by the **report_timing** and **report_cell** commands.
- UI

```
check_ilm
   [-stage pre_place_opt | pre_clock_opt |
           pre_route_opt]
```

Checking and Reporting Enhancements:
User Benefits

- You can now use the **`check_ilm`** command to perform checks on ILMs from the top level. Prior to 2010.03, the **`check_ilm`** command was run only as part of the **`check_physical_design`** command.
- The **`check_ilm`** command now performs the following additional checks:

Check (Error / Warning)	Error ID
[Warning] The libraries from which ILM and FRAM/CEL are loaded are different for ILM instance %s. ILM Library: %s; FRAM/CEL Library: %s	ILM-126
[Error] Top-level PG connection is not made to ILM PG pin %s .	ILM-128
[Warning] Port %s in ILM/CEL/FRAM %s does not have matching port in ILM/CEL/FRAM view.	ILM-129

Checking and Reporting Enhancements:
User Benefits

- The **`report_ilm`** command now reports the following additional information:
  - User-specified multicorner-multimode scenario mapping between the top-level design and the ILM
  - TLUPlus settings for both active and inactive scenarios
  - UPF information. See the IC Compiler version 2009.06-SP3 Release Notes for more information.
- You can now use the **`report_ilm`** command with ILMs also (i.e. **`open_mw_cel`** Block.ILM)

Checking and Reporting Enhancements:
User Benefits

- In reports generated by the **`report_timing`** command, ILM objects are marked using the following new attributes:
  - IL - ILM instance
  - il – pin or net inside ILM

```
Attributes:
    d - dont_touch
    u - dont_use
   mo - map_only
   so - size_only
    i - ideal_net or ideal_network
   IL - ILM instance
   il - pin/net inside ILM

  Point                                   Fanout      Incr       Path      Attributes     Voltage
  ---------------------------------------------------------------------------------------------
  clock Clk (rise edge)                               0.00       0.00
  clock network delay (ideal)                         0.00       0.00
  ...
  ...
  ILM_inst/in[1] (ILM_ref)                            0.00       0.27 f    IL             1.08
  ILM_inst/in[1] (net)                                0.00       0.27 f    d il
  ILM_inst/u0/in (ref)                                0.00       0.27 f    il             1.08
  ILM_inst/u0/in (net)                                0.00       0.27 f    d il
  ...
```

Checking and Reporting Enhancements:
User Benefits

- In reports generated by the **`report_cell`** command, ILM objects are marked using the following new attributes:
  - IL - ILM instance
  - il - cell inside ILM

```
report_cell inst_ILM1 -nosplit
...
Cell                    Reference       Library            Area  Attributes
--------------------------------------------------------------------------------
inst_ILM1                 ILM                          115179.927039 IL, d, h
--------------------------------------------------------------------------------
Total 1 cells                                          115179.927039
1

report_cell inst_ILM1/U0 -nosplit
...
Cell                    Reference       Library            Area  Attributes
--------------------------------------------------------------------------------
inst_ILM1/U0             ref1            lib1.slow        19.958401 d, il
--------------------------------------------------------------------------------
Total 1 cells                                          19.958401
1
```

Summary Top Level Design Closure

- Average 40% runtime improvement
- Improved HFS QoR
- Improved usability of ILMs in multicorner-multimode flow
- Unrouted ILMs can now be used in top-level postroute stages
- Improved checking and reporting capability of ILMs

Predictable Success