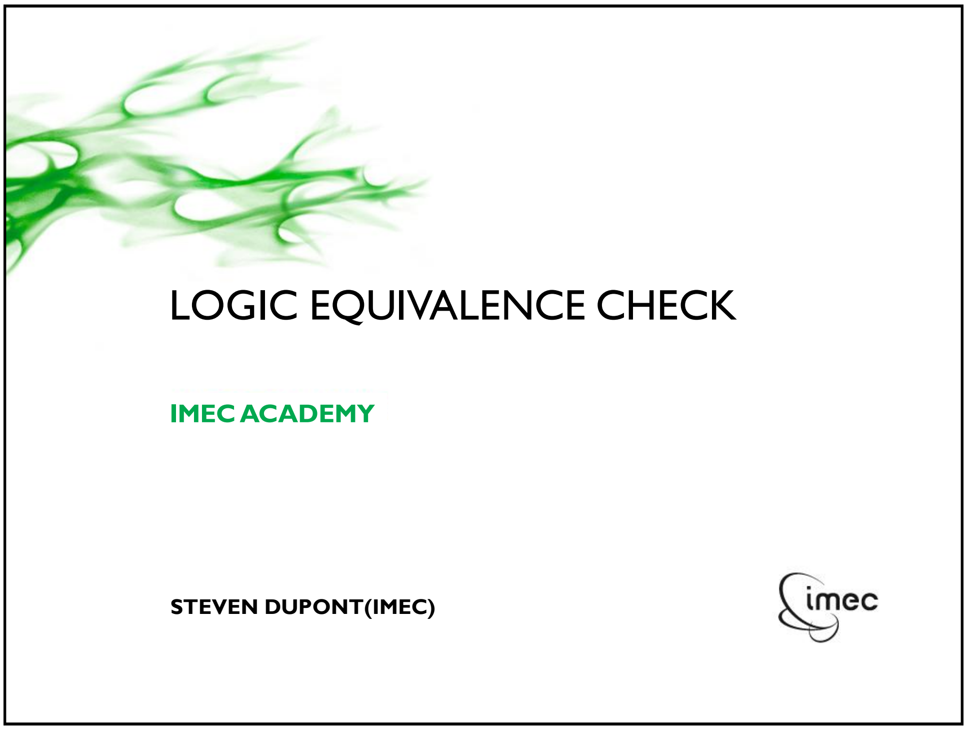
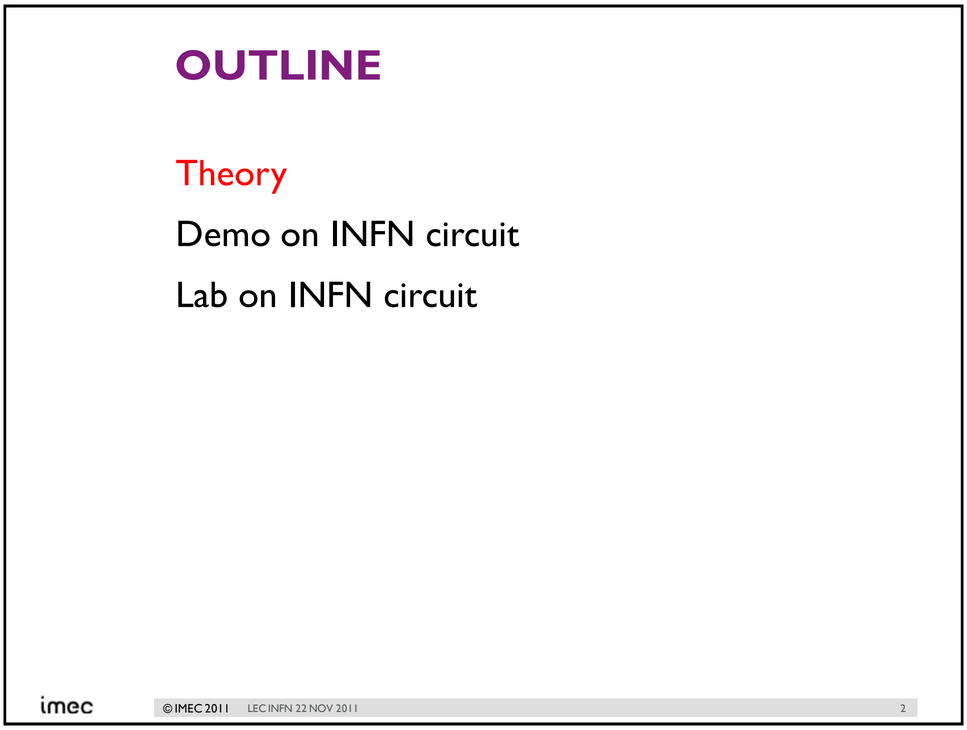


LOGIC EQUIVALENCE CHECK

IMEC ACADEMY

STEVEN DUPONT(IMEC)

imec



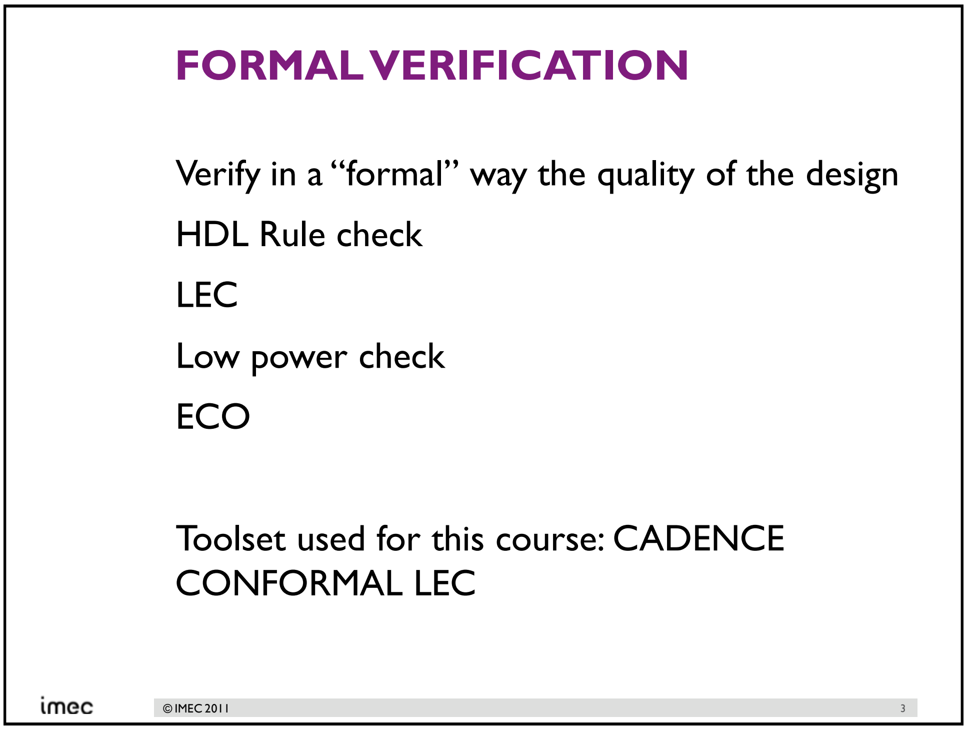
OUTLINE

Theory

Demo on INFN circuit

Lab on INFN circuit

imec © IMEC 2011 LEC INFN 22 NOV 2011 2



FORMAL VERIFICATION

Verify in a "formal" way the quality of the design

HDL Rule check

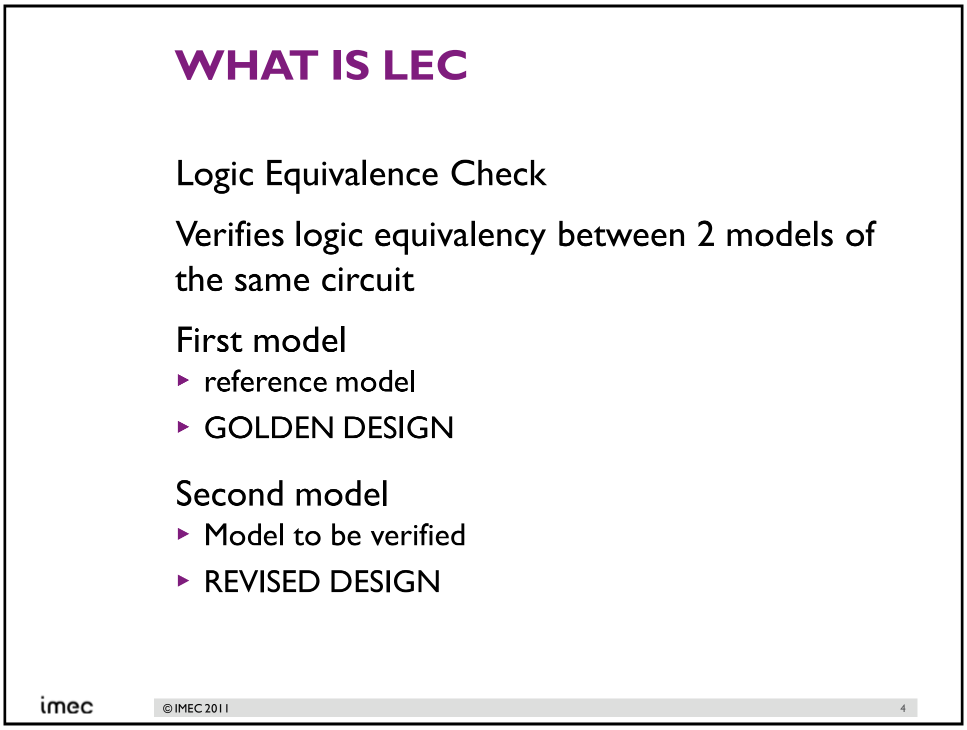
LEC

Low power check

ECO

Toolset used for this course: CADENCE CONFORMAL LEC

imec © IMEC 2011 3



WHAT IS LEC

Logic Equivalence Check

Verifies logic equivalency between 2 models of the same circuit

First model
- reference model
- GOLDEN DESIGN

Second model
- Model to be verified
- REVISED DESIGN

imec © IMEC 2011 4

DESIGN TYPES

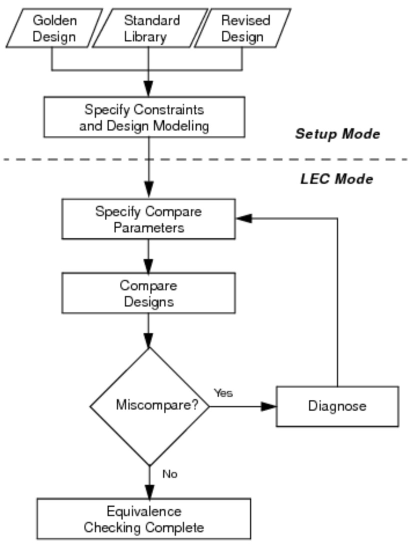
RTL

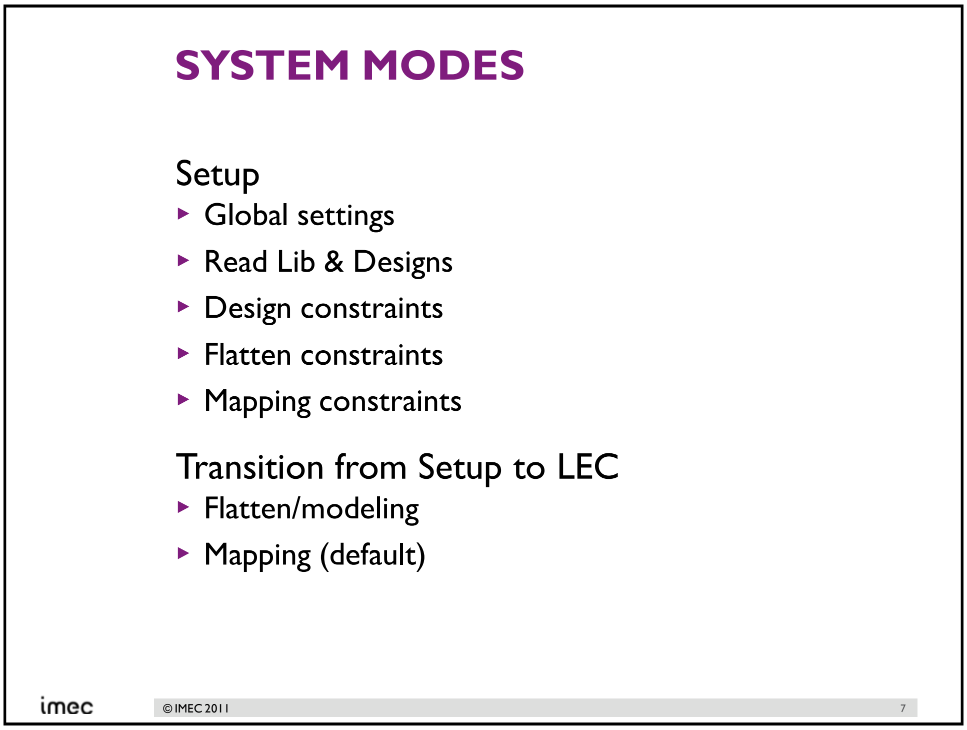
Netlist

GDSII

Frequent use cases:
- RTL vs first mapped Netlist
- First mapped Netlist vs optimised/test inserted Netlist
- Netlist before P&R versus after P&R

LEC: FLOW



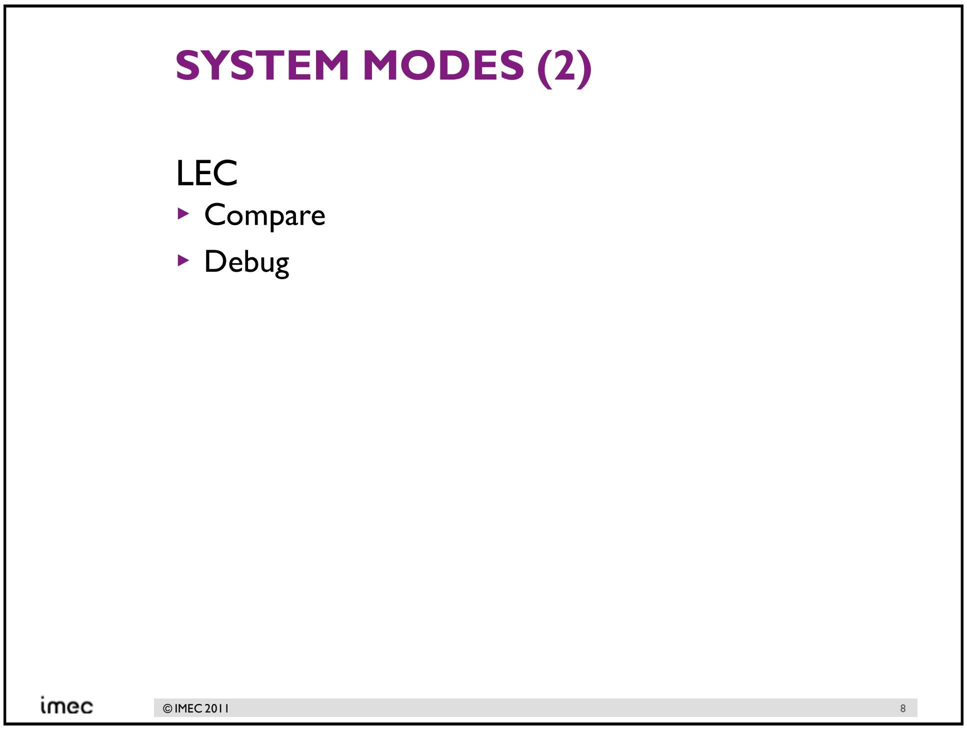


SYSTEM MODES

Setup
- Global settings
- Read Lib & Designs
- Design constraints
- Flatten constraints
- Mapping constraints

Transition from Setup to LEC
- Flatten/modeling
- Mapping (default)

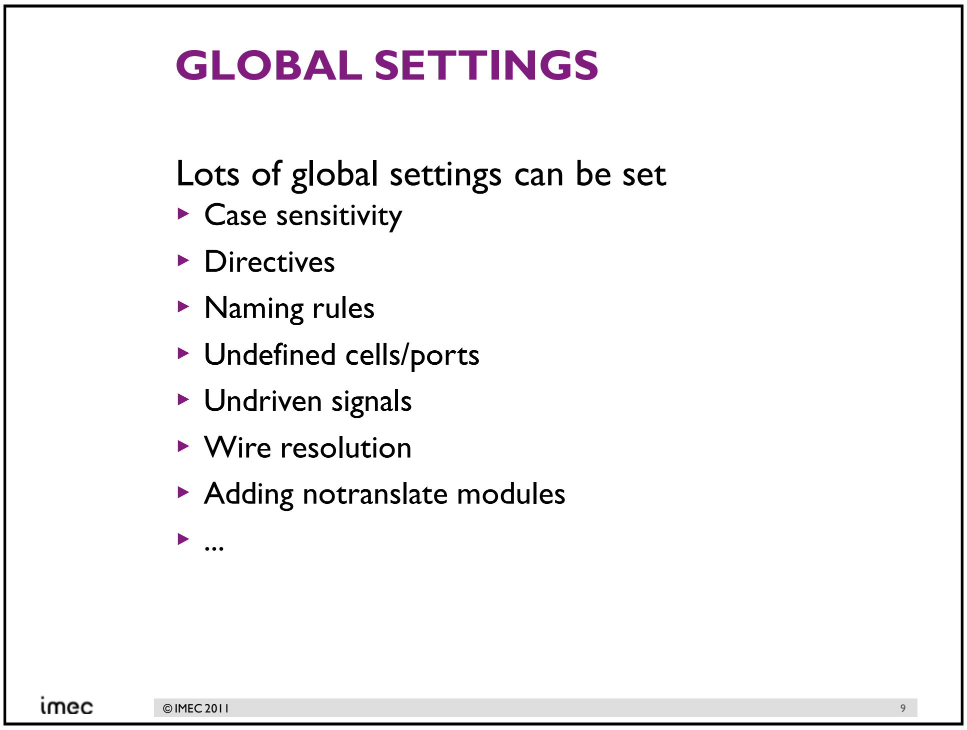
imec © IMEC 2011 7



SYSTEM MODES (2)

LEC
- Compare
- Debug

imec © IMEC 2011 8

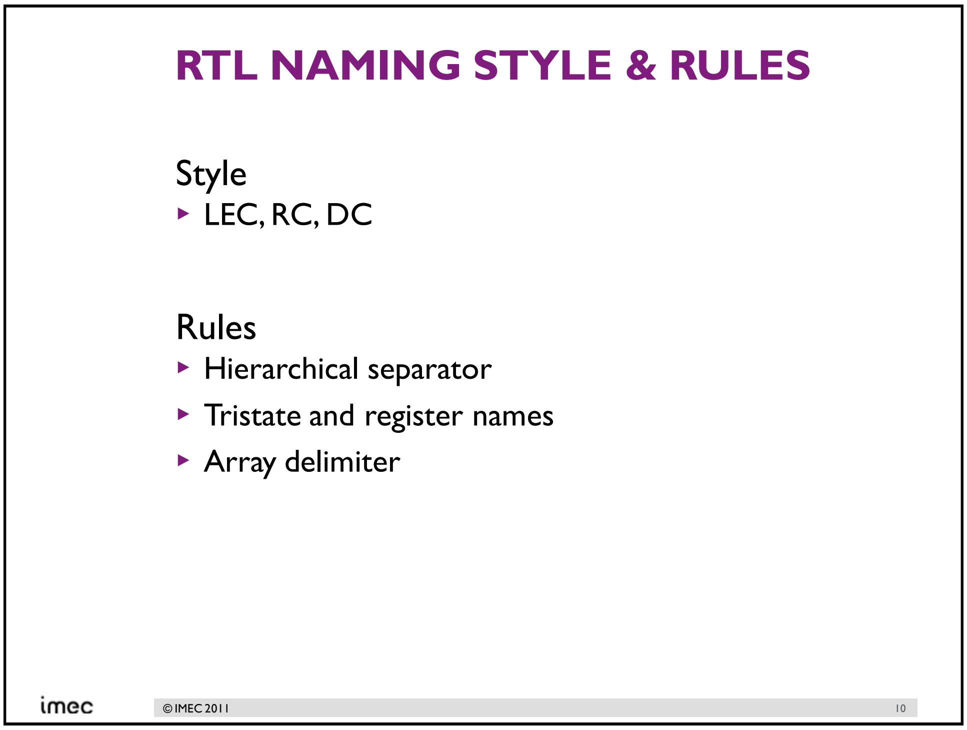


GLOBAL SETTINGS

Lots of global settings can be set

- Case sensitivity
- Directives
- Naming rules
- Undefined cells/ports
- Undriven signals
- Wire resolution
- Adding notranslate modules
- ...

imec © IMEC 2011 9



RTL NAMING STYLE & RULES

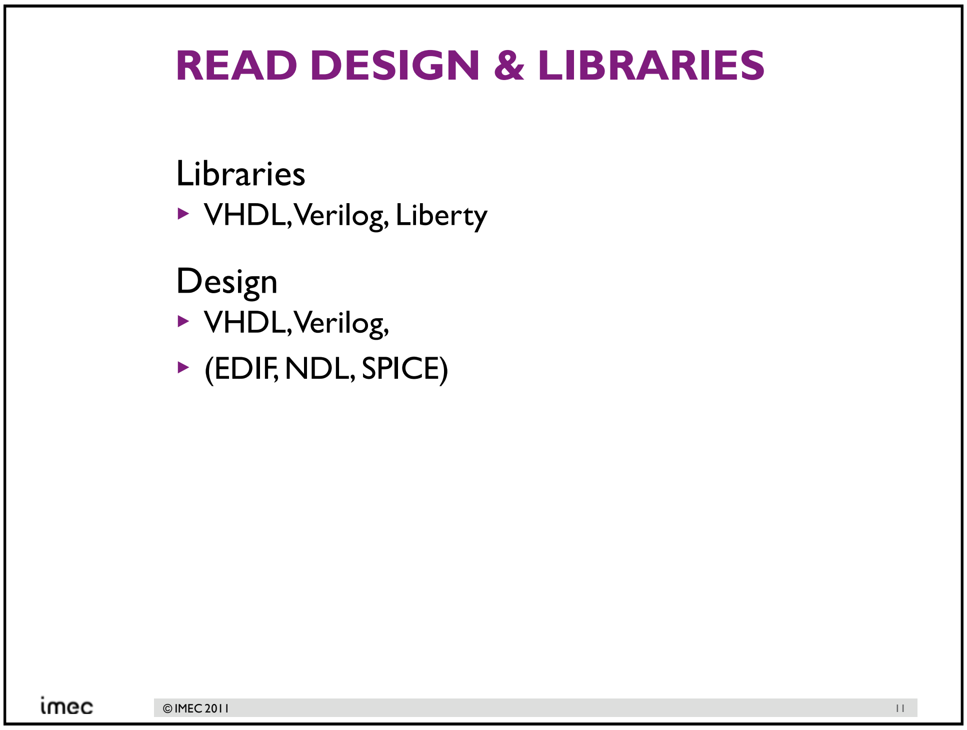
Style

- LEC, RC, DC

Rules

- Hierarchical separator
- Tristate and register names
- Array delimiter

imec © IMEC 2011 10

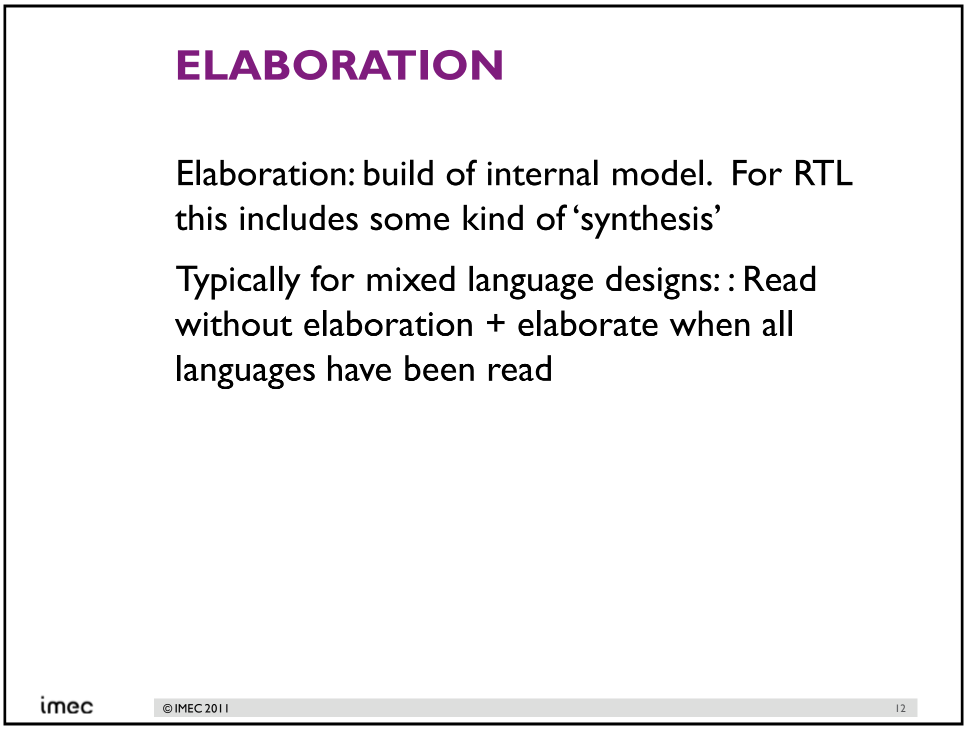


READ DESIGN & LIBRARIES

Libraries
- VHDL, Verilog, Liberty

Design
- VHDL, Verilog,
- (EDIF, NDL, SPICE)

imec © IMEC 2011 11

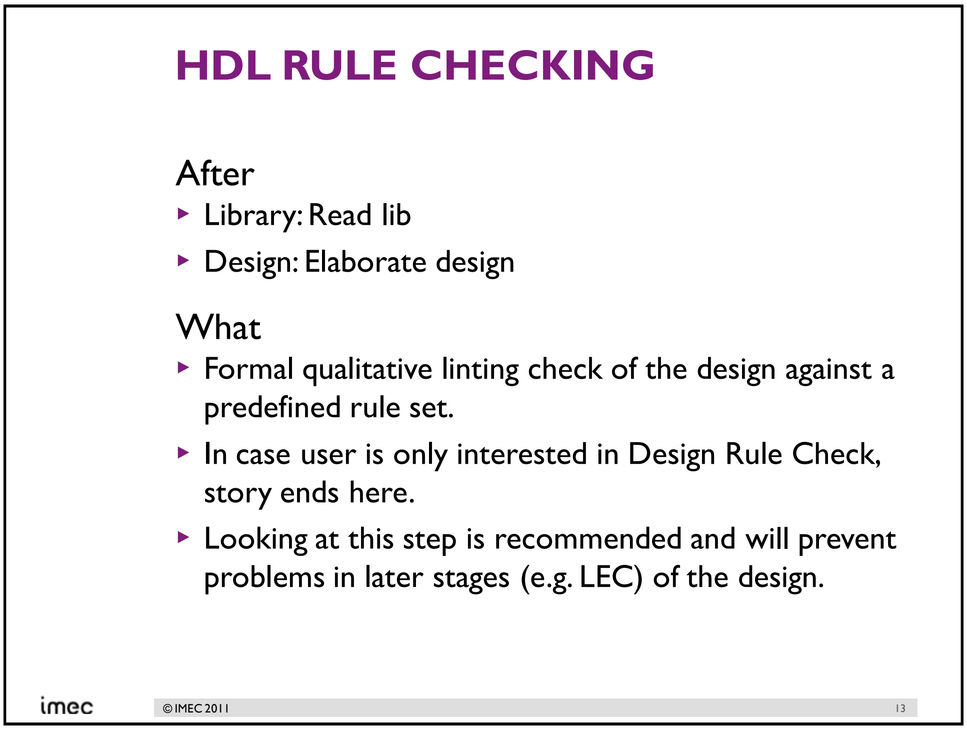


ELABORATION

Elaboration: build of internal model. For RTL this includes some kind of 'synthesis'

Typically for mixed language designs: : Read without elaboration + elaborate when all languages have been read

imec © IMEC 2011 12

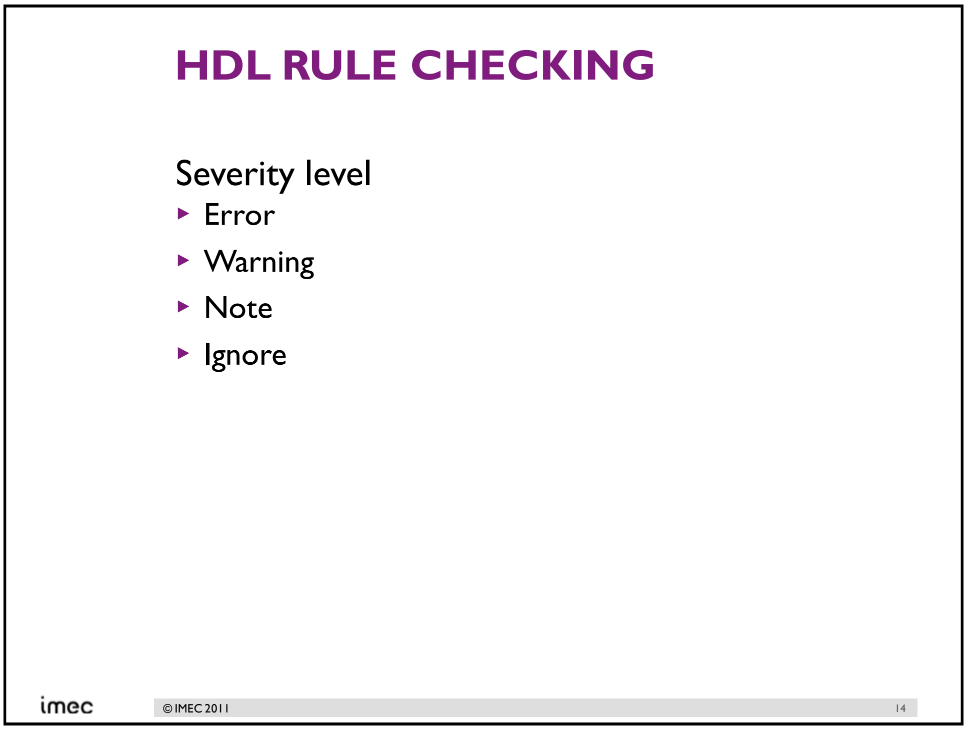


HDL RULE CHECKING

After
- Library: Read lib
- Design: Elaborate design

What
- Formal qualitative linting check of the design against a predefined rule set.
- In case user is only interested in Design Rule Check, story ends here.
- Looking at this step is recommended and will prevent problems in later stages (e.g. LEC) of the design.

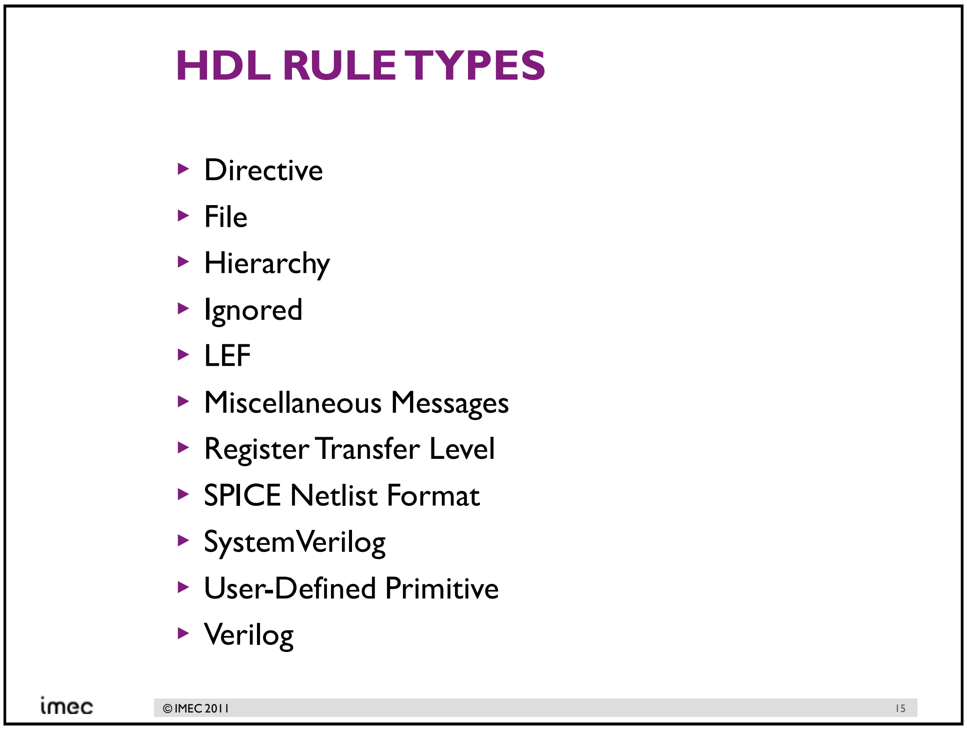
imec © IMEC 2011 13



HDL RULE CHECKING

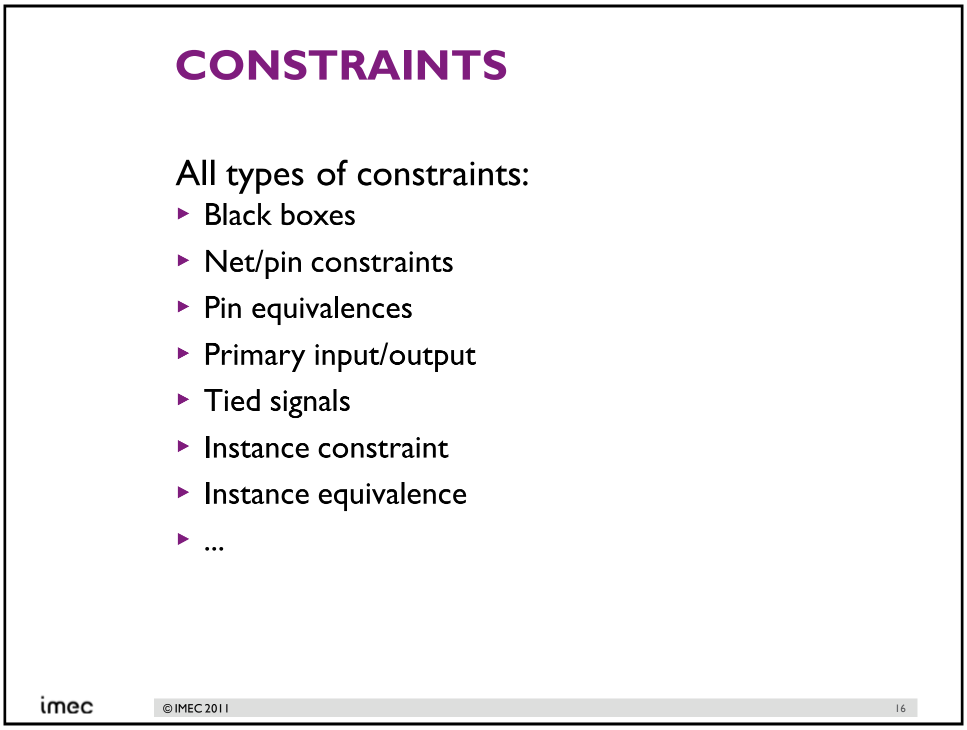
Severity level
- Error
- Warning
- Note
- Ignore

imec © IMEC 2011 14



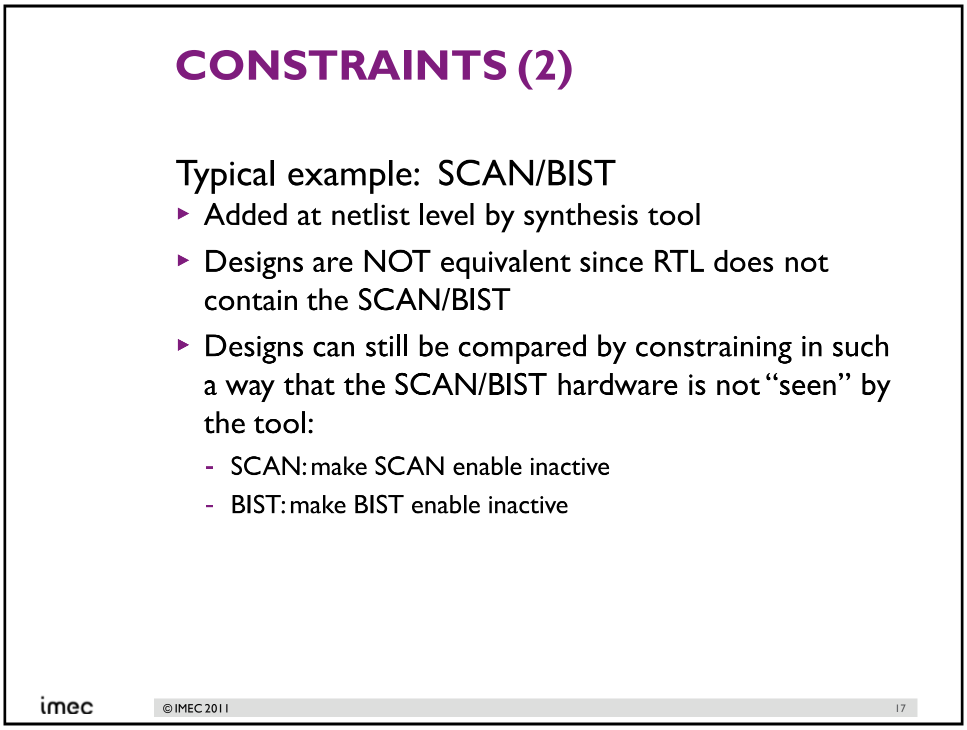
HDL RULE TYPES

- Directive
- File
- Hierarchy
- Ignored
- LEF
- Miscellaneous Messages
- Register Transfer Level
- SPICE Netlist Format
- SystemVerilog
- User-Defined Primitive
- Verilog



CONSTRAINTS

All types of constraints:
- Black boxes
- Net/pin constraints
- Pin equivalences
- Primary input/output
- Tied signals
- Instance constraint
- Instance equivalence
- …

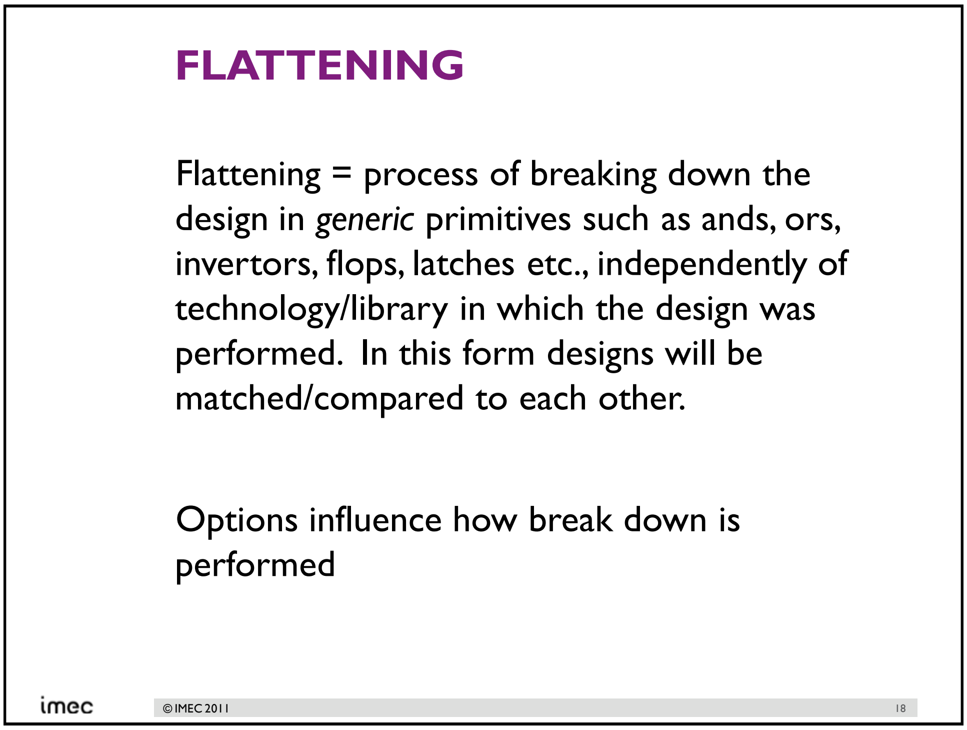


CONSTRAINTS (2)

Typical example: SCAN/BIST

- Added at netlist level by synthesis tool
- Designs are NOT equivalent since RTL does not contain the SCAN/BIST
- Designs can still be compared by constraining in such a way that the SCAN/BIST hardware is not "seen" by the tool:
  - SCAN: make SCAN enable inactive
  - BIST: make BIST enable inactive

imec © IMEC 2011 17

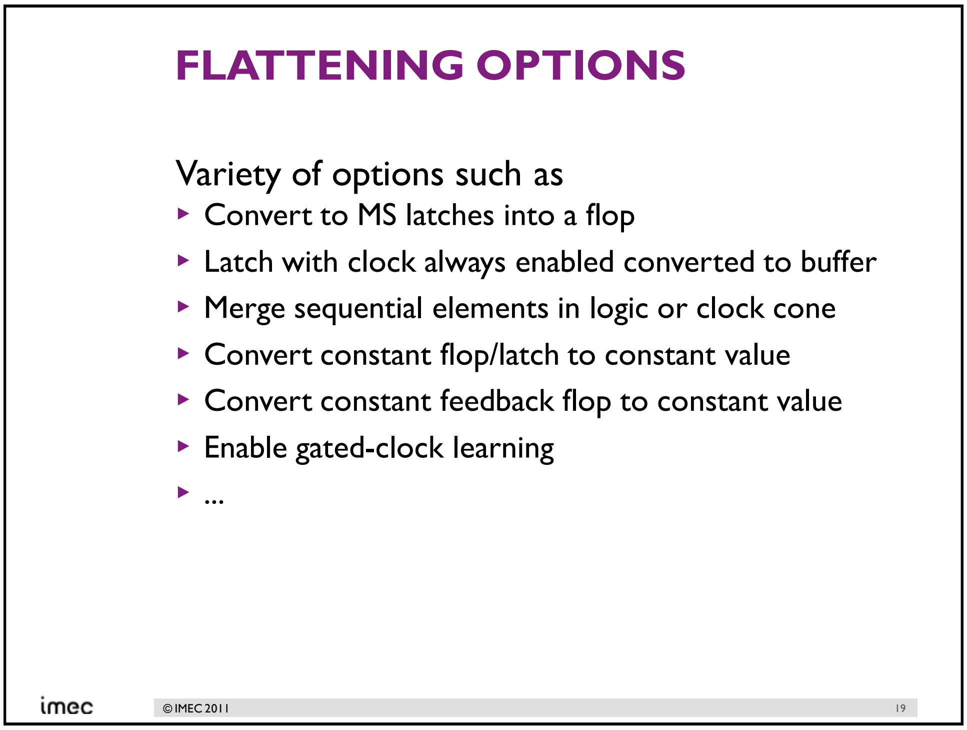


FLATTENING

Flattening = process of breaking down the design in *generic* primitives such as ands, ors, invertors, flops, latches etc., independently of technology/library in which the design was performed. In this form designs will be matched/compared to each other.

Options influence how break down is performed

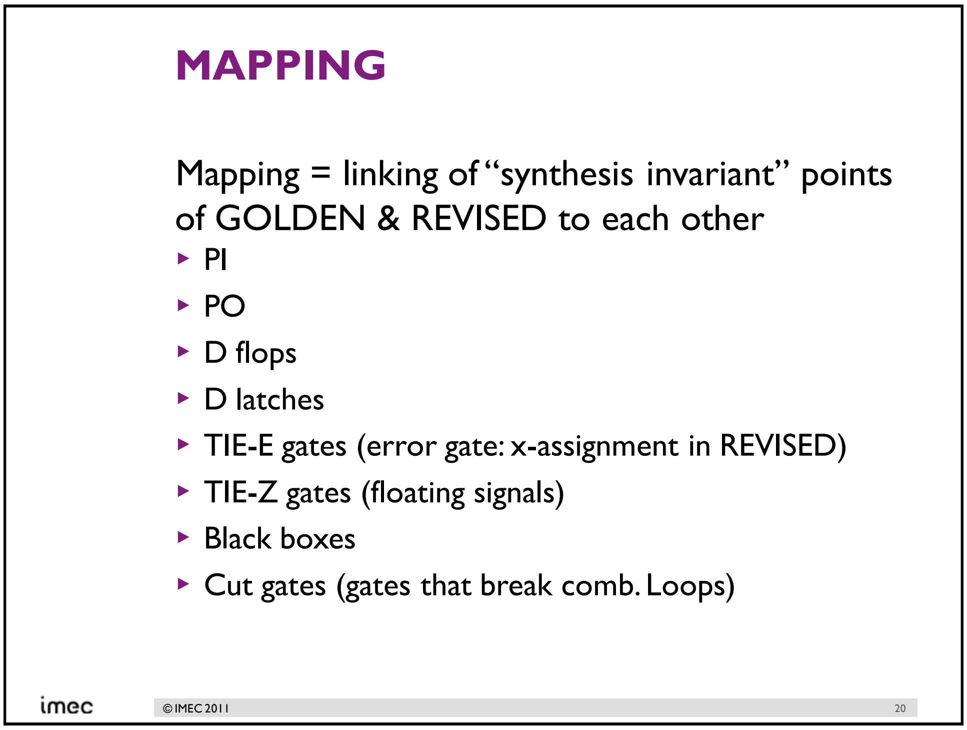
imec © IMEC 2011 18



FLATTENING OPTIONS

Variety of options such as
- Convert to MS latches into a flop
- Latch with clock always enabled converted to buffer
- Merge sequential elements in logic or clock cone
- Convert constant flop/latch to constant value
- Convert constant feedback flop to constant value
- Enable gated-clock learning
- …

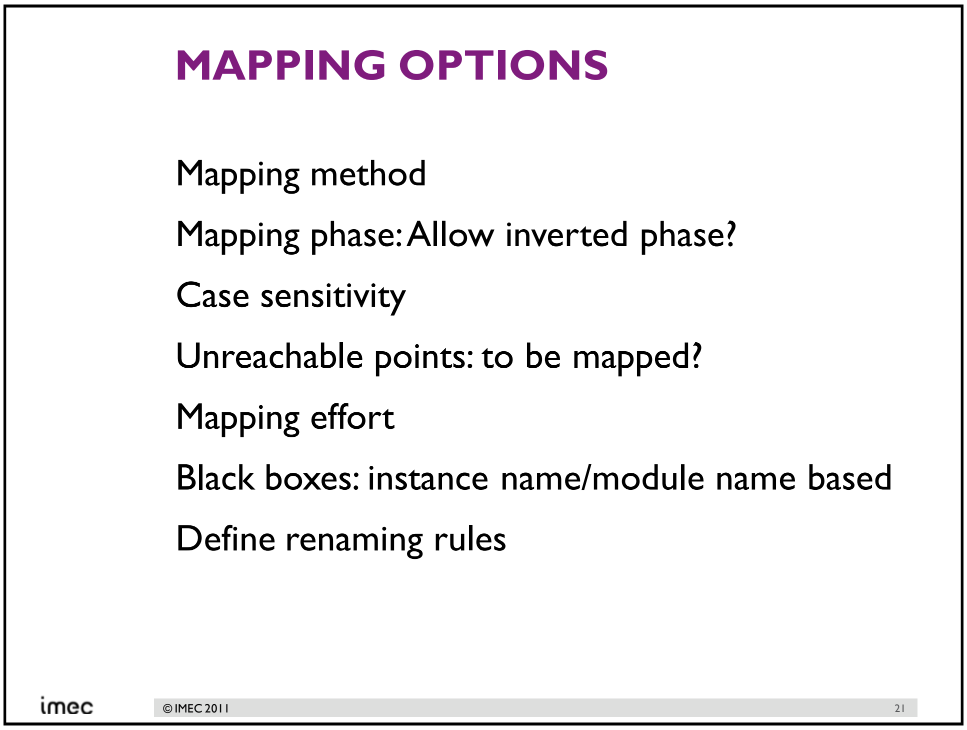
imec © IMEC 2011 19



MAPPING

Mapping = linking of "synthesis invariant" points of GOLDEN & REVISED to each other
- PI
- PO
- D flops
- D latches
- TIE-E gates (error gate: x-assignment in REVISED)
- TIE-Z gates (floating signals)
- Black boxes
- Cut gates (gates that break comb. Loops)

imec © IMEC 2011 20



MAPPING OPTIONS

Mapping method

Mapping phase: Allow inverted phase?

Case sensitivity

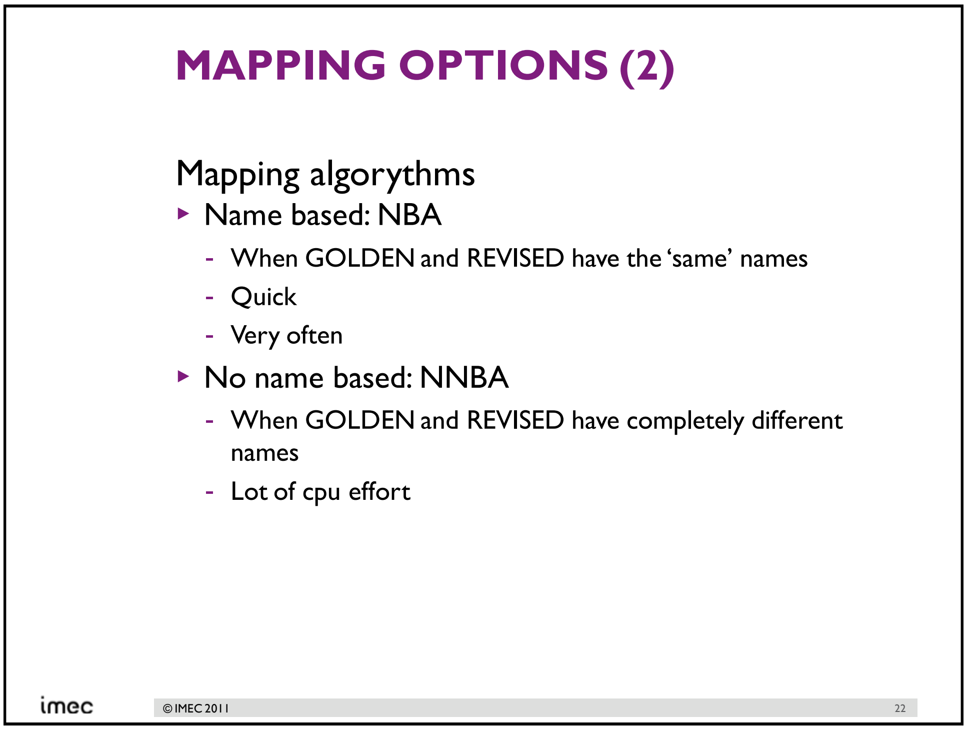
Unreachable points: to be mapped?

Mapping effort

Black boxes: instance name/module name based

Define renaming rules

imec © IMEC 2011 21

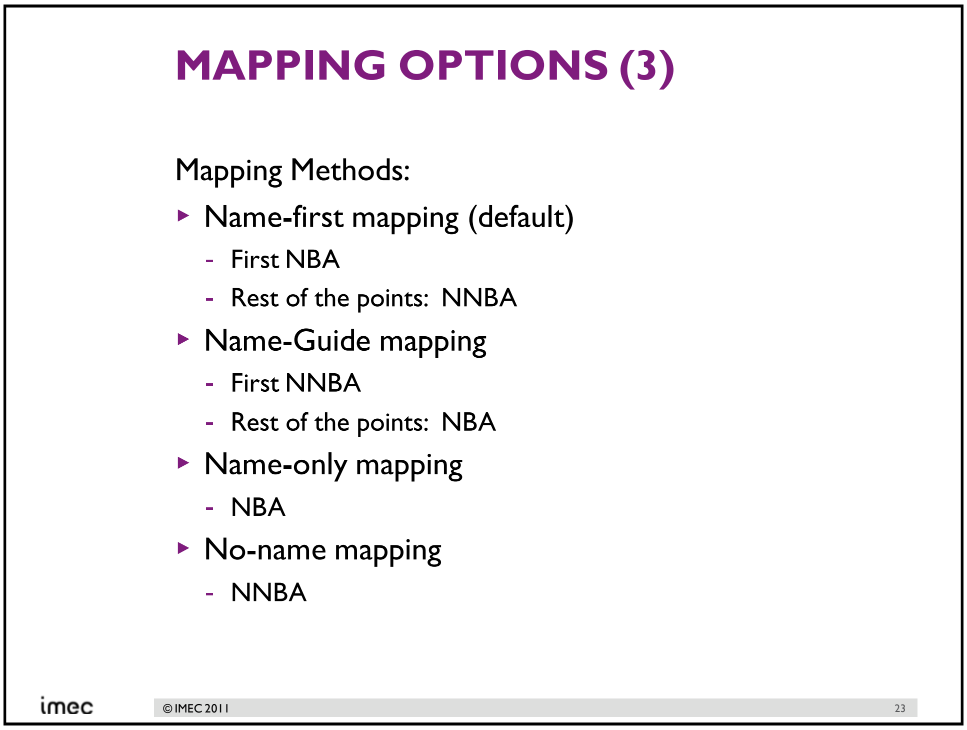


MAPPING OPTIONS (2)

Mapping algorythms

- Name based: NBA
  - When GOLDEN and REVISED have the 'same' names
  - Quick
  - Very often
- No name based: NNBA
  - When GOLDEN and REVISED have completely different names
  - Lot of cpu effort

imec © IMEC 2011 22

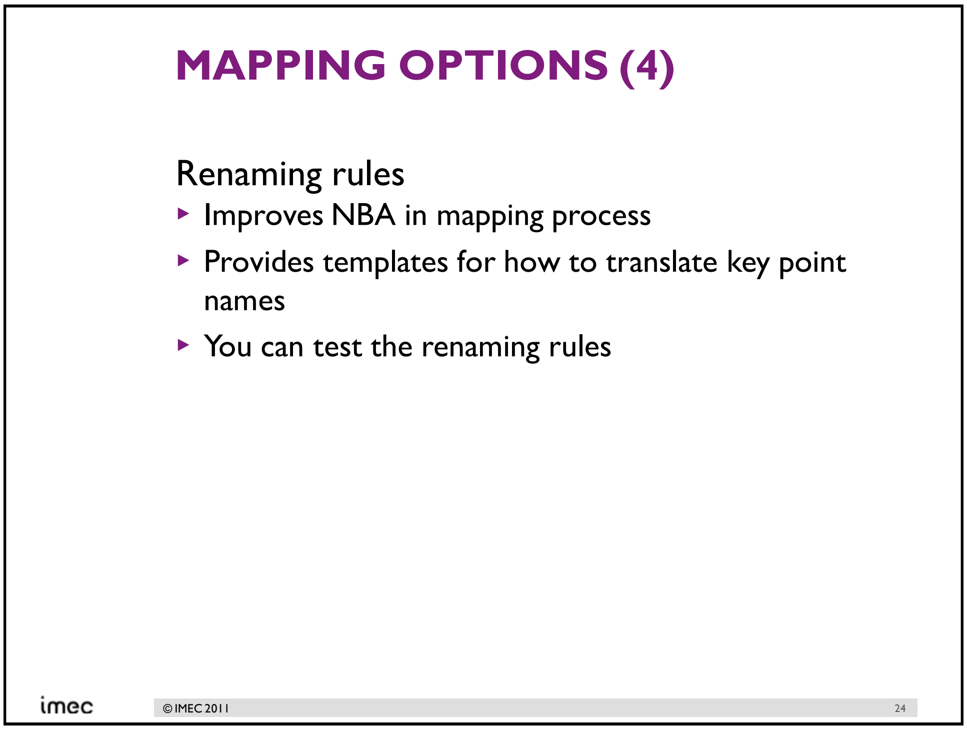


MAPPING OPTIONS (3)

Mapping Methods:

- Name-first mapping (default)
  - First NBA
  - Rest of the points: NNBA
- Name-Guide mapping
  - First NNBA
  - Rest of the points: NBA
- Name-only mapping
  - NBA
- No-name mapping
  - NNBA

imec © IMEC 2011 23

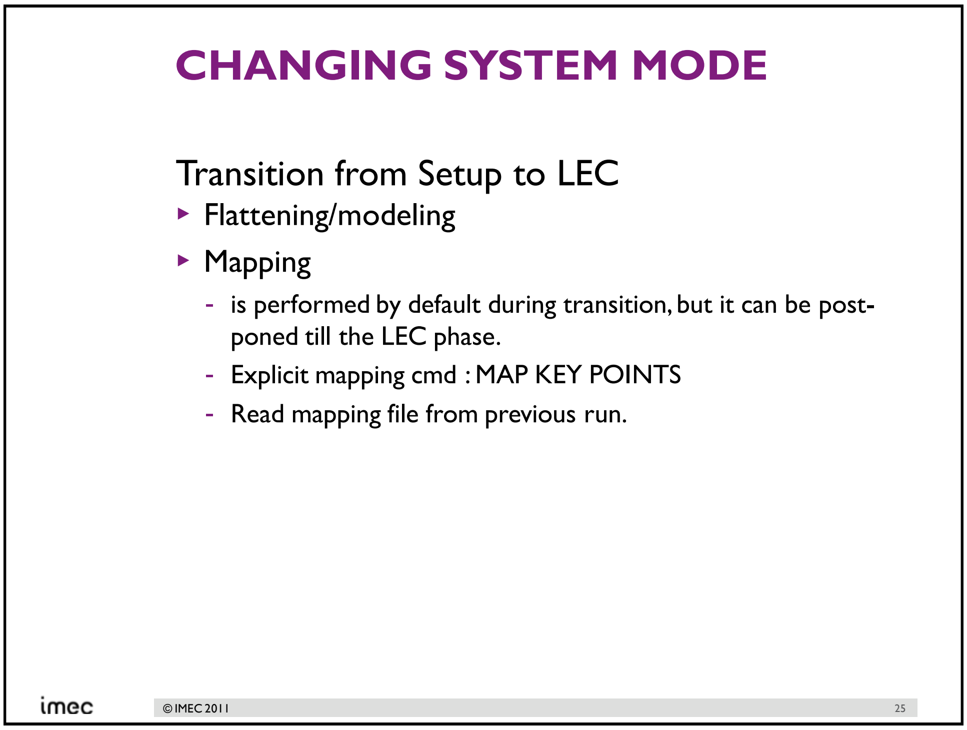


MAPPING OPTIONS (4)

Renaming rules

- Improves NBA in mapping process
- Provides templates for how to translate key point names
- You can test the renaming rules

imec © IMEC 2011 24

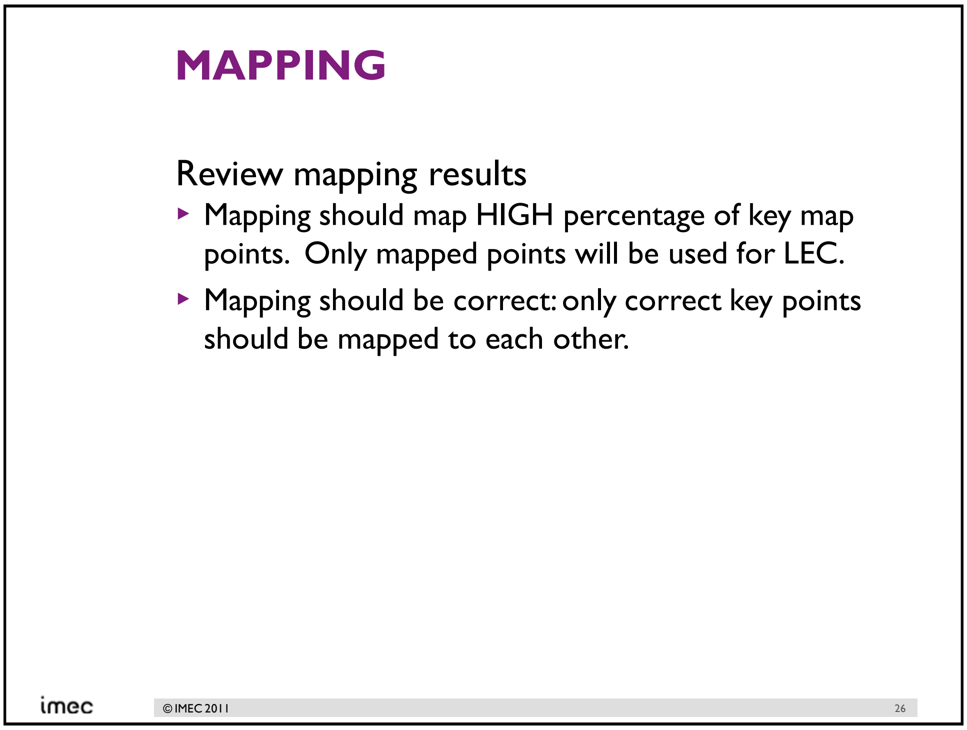


CHANGING SYSTEM MODE

Transition from Setup to LEC

- Flattening/modeling
- Mapping
  - is performed by default during transition, but it can be post-poned till the LEC phase.
  - Explicit mapping cmd : MAP KEY POINTS
  - Read mapping file from previous run.

imec © IMEC 2011 25

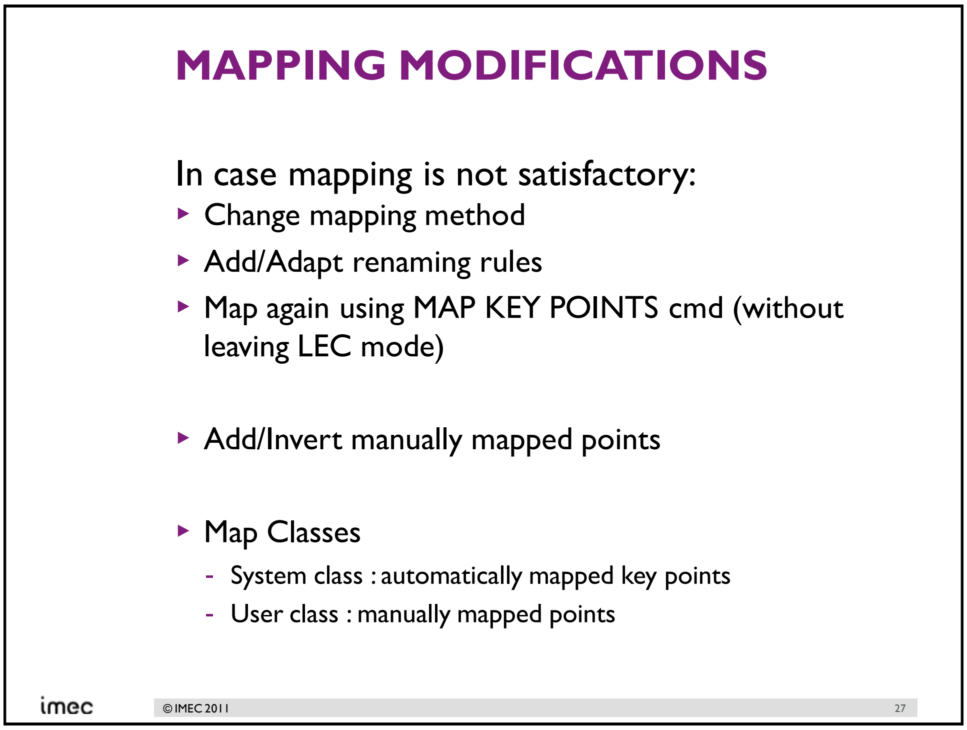


MAPPING

Review mapping results

- Mapping should map HIGH percentage of key map points. Only mapped points will be used for LEC.
- Mapping should be correct: only correct key points should be mapped to each other.

imec © IMEC 2011 26



MAPPING MODIFICATIONS

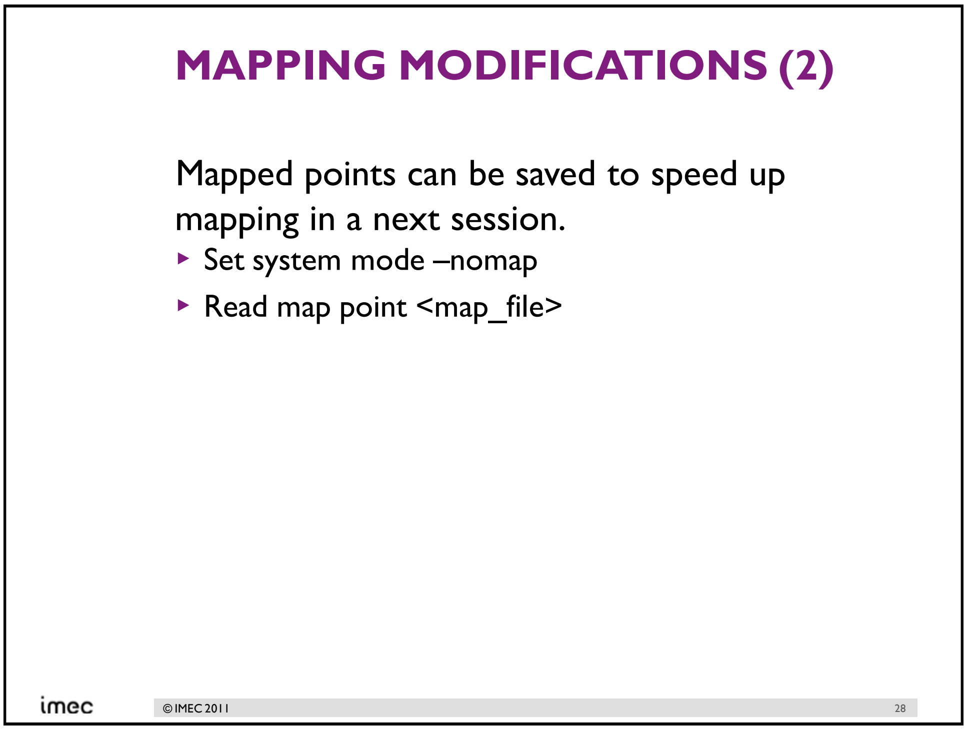
In case mapping is not satisfactory:

- Change mapping method
- Add/Adapt renaming rules
- Map again using MAP KEY POINTS cmd (without leaving LEC mode)

- Add/Invert manually mapped points

- Map Classes
  - System class : automatically mapped key points
  - User class : manually mapped points

imec © IMEC 2011 27

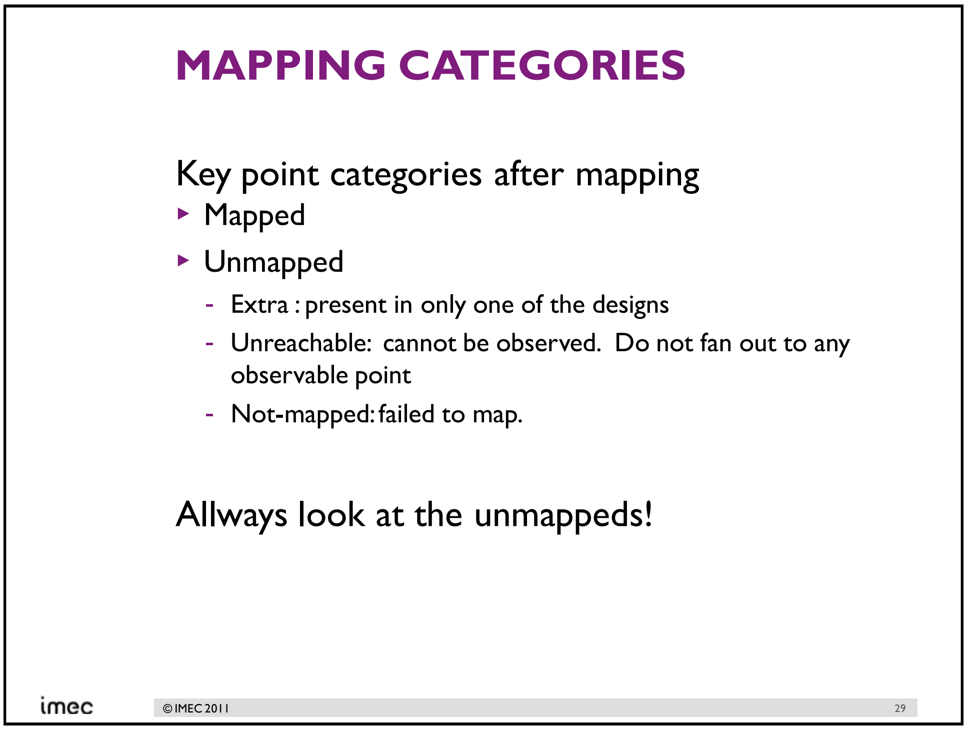


MAPPING MODIFICATIONS (2)

Mapped points can be saved to speed up mapping in a next session.

- Set system mode –nomap
- Read map point <map_file>

imec © IMEC 2011 28

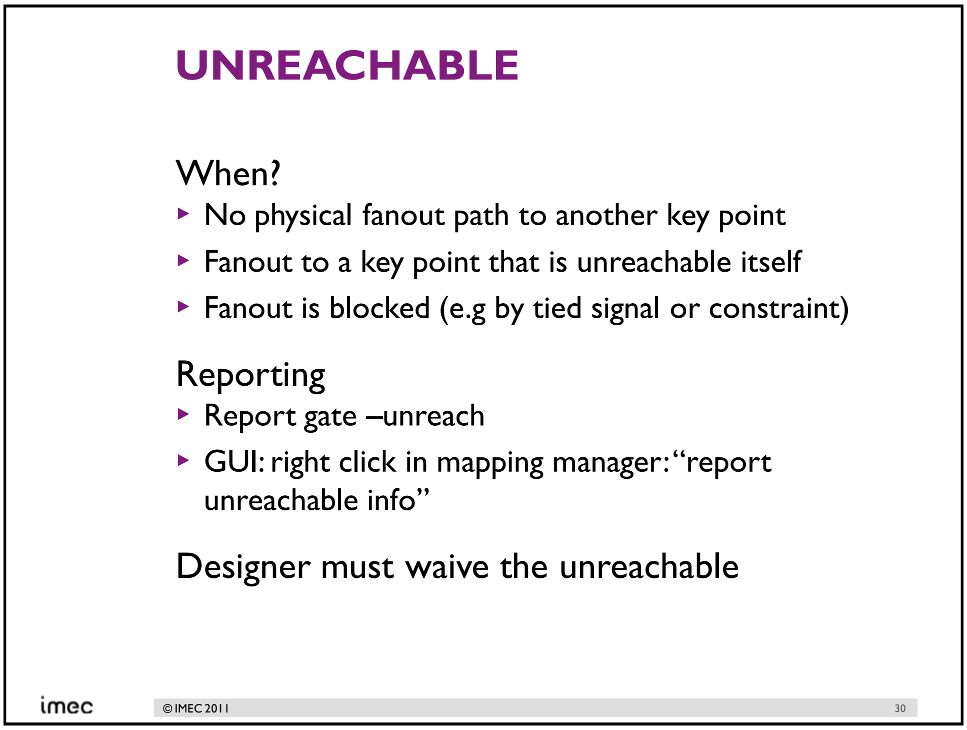


MAPPING CATEGORIES

Key point categories after mapping
- Mapped
- Unmapped
  - Extra : present in only one of the designs
  - Unreachable: cannot be observed. Do not fan out to any observable point
  - Not-mapped: failed to map.

Allways look at the unmappeds!

imec © IMEC 2011 29



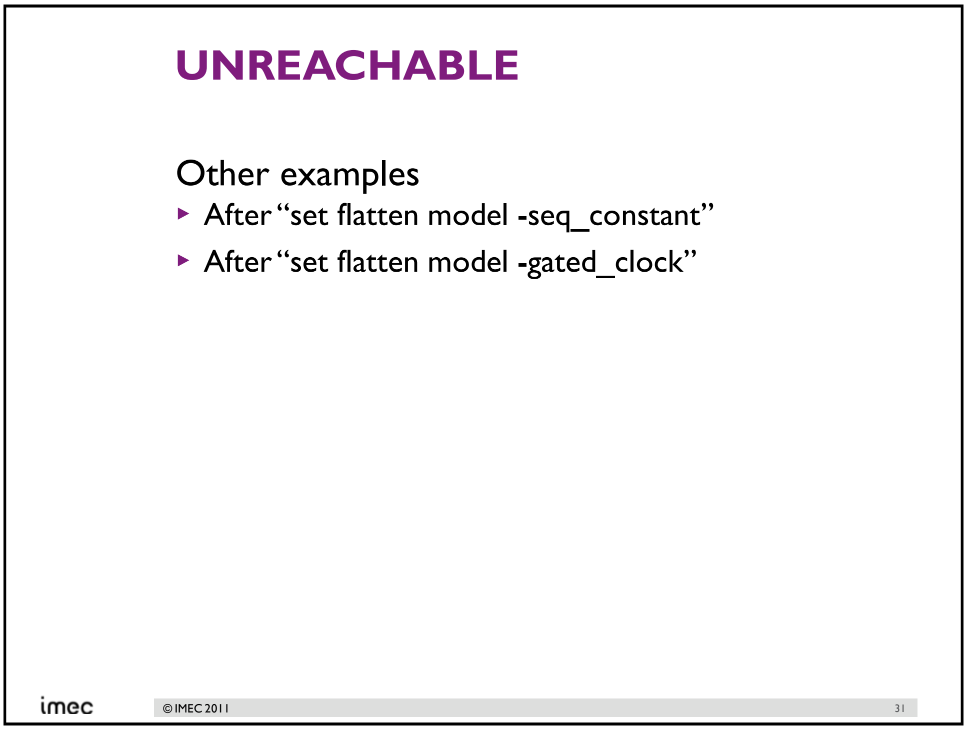
UNREACHABLE

When?
- No physical fanout path to another key point
- Fanout to a key point that is unreachable itself
- Fanout is blocked (e.g by tied signal or constraint)

Reporting
- Report gate –unreach
- GUI: right click in mapping manager: "report unreachable info"

Designer must waive the unreachable

imec © IMEC 2011 30

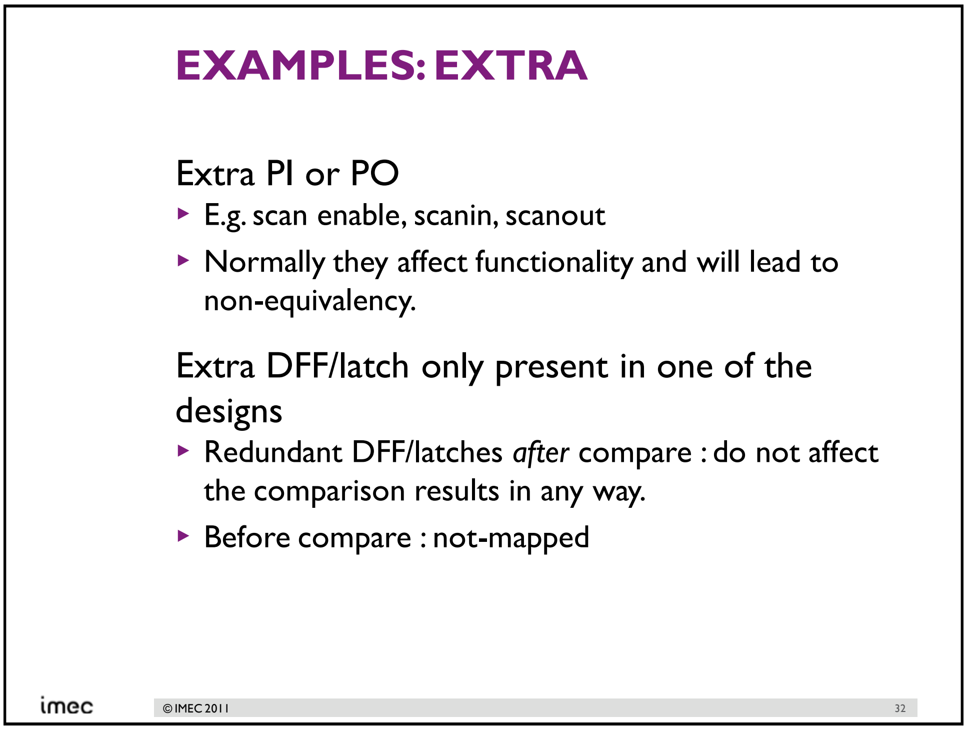


UNREACHABLE

Other examples

- After "set flatten model -seq_constant"
- After "set flatten model -gated_clock"

imec © IMEC 2011 31



EXAMPLES: EXTRA

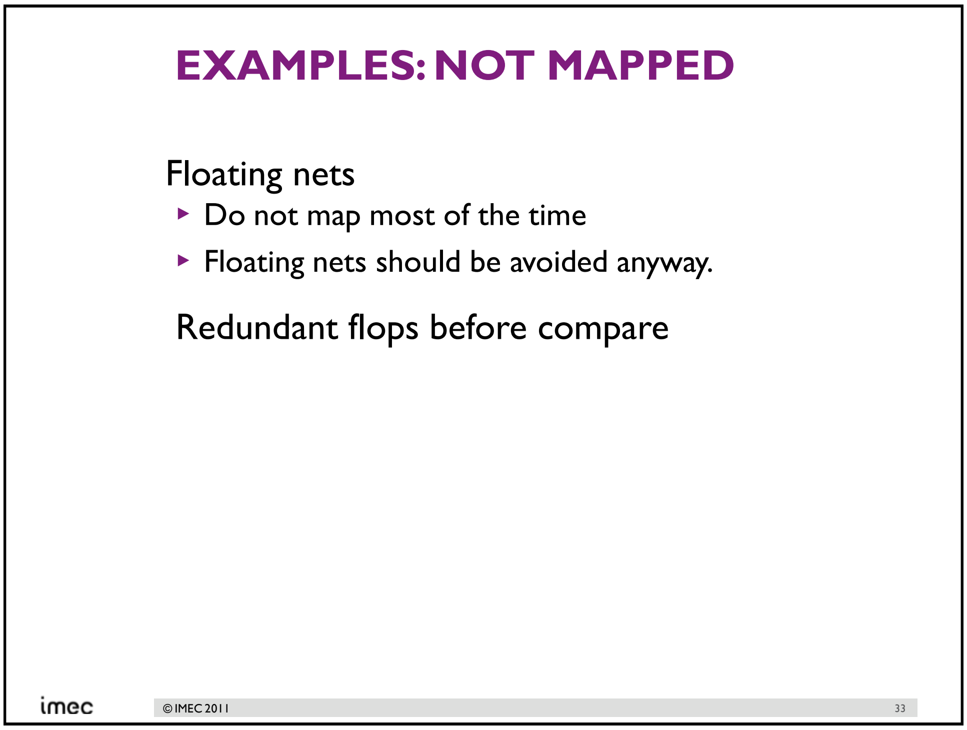
Extra PI or PO

- E.g. scan enable, scanin, scanout
- Normally they affect functionality and will lead to non-equivalency.

Extra DFF/latch only present in one of the designs

- Redundant DFF/latches *after* compare : do not affect the comparison results in any way.
- Before compare : not-mapped

imec © IMEC 2011 32

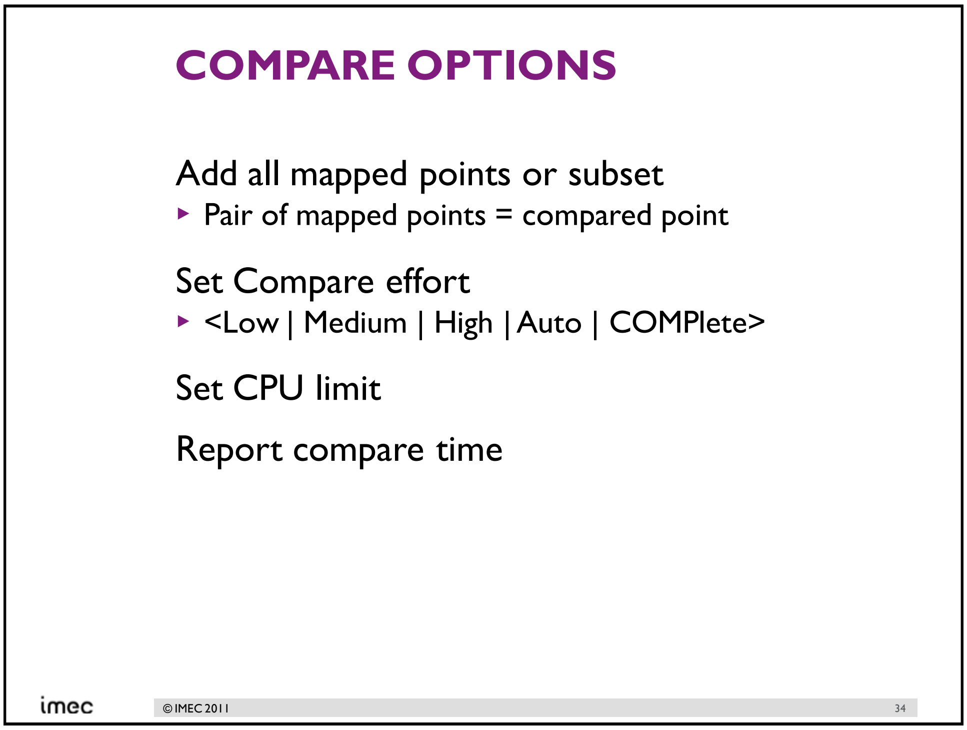


EXAMPLES: NOT MAPPED

Floating nets
- Do not map most of the time
- Floating nets should be avoided anyway.

Redundant flops before compare

© IMEC 2011 33



COMPARE OPTIONS

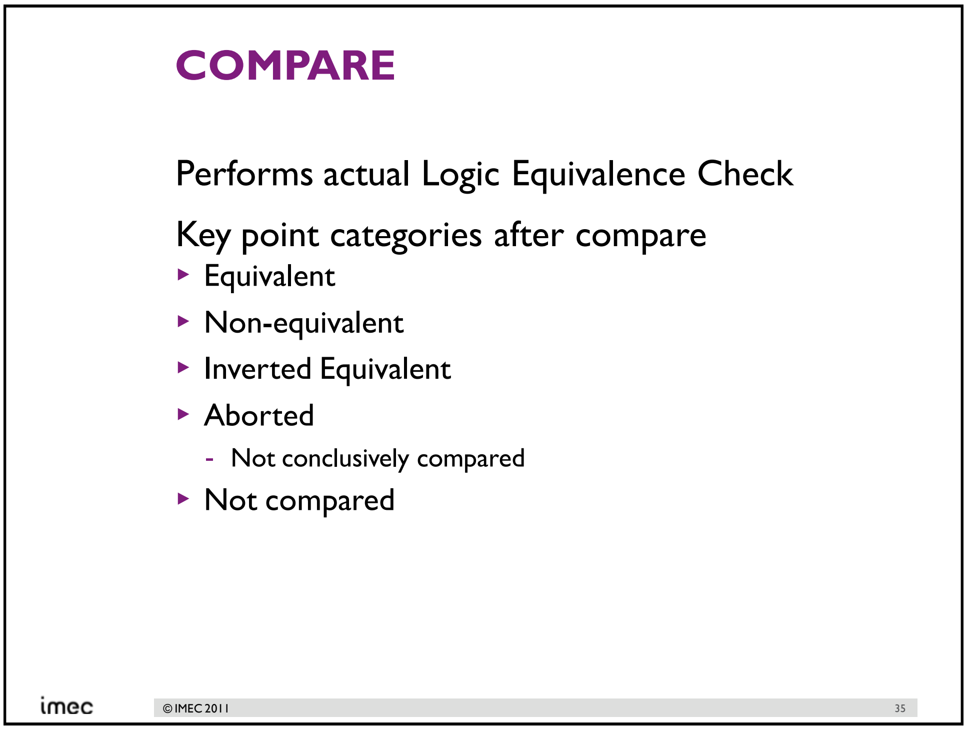
Add all mapped points or subset
- Pair of mapped points = compared point

Set Compare effort
- <Low | Medium | High | Auto | COMPlete>

Set CPU limit

Report compare time

© IMEC 2011 34

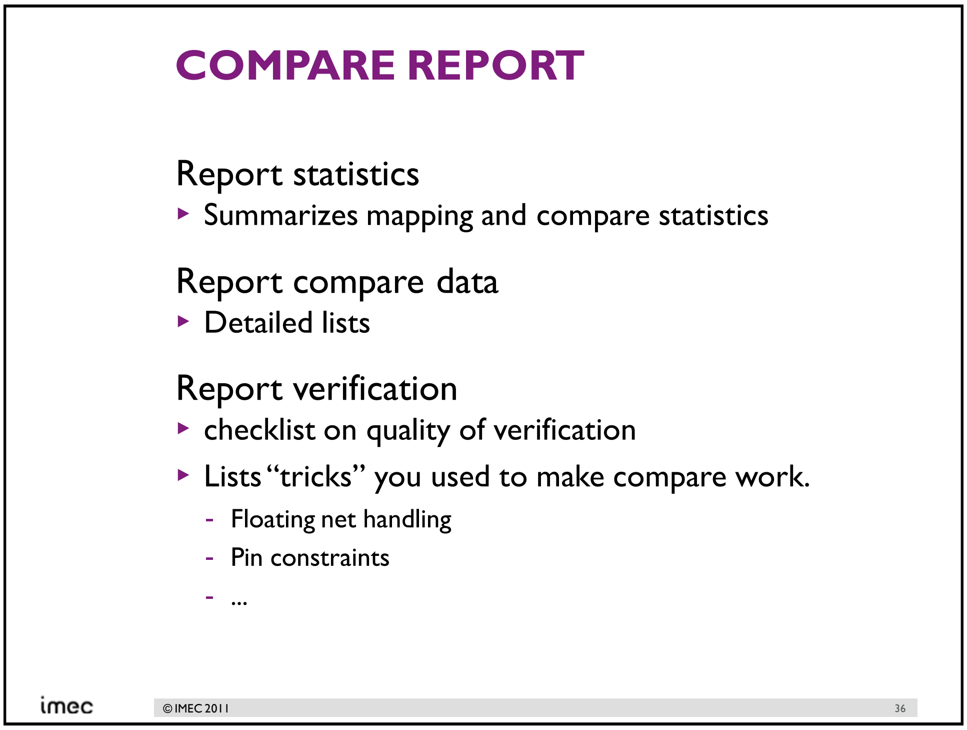


COMPARE

Performs actual Logic Equivalence Check

Key point categories after compare

- Equivalent
- Non-equivalent
- Inverted Equivalent
- Aborted
  - Not conclusively compared
- Not compared



COMPARE REPORT

Report statistics

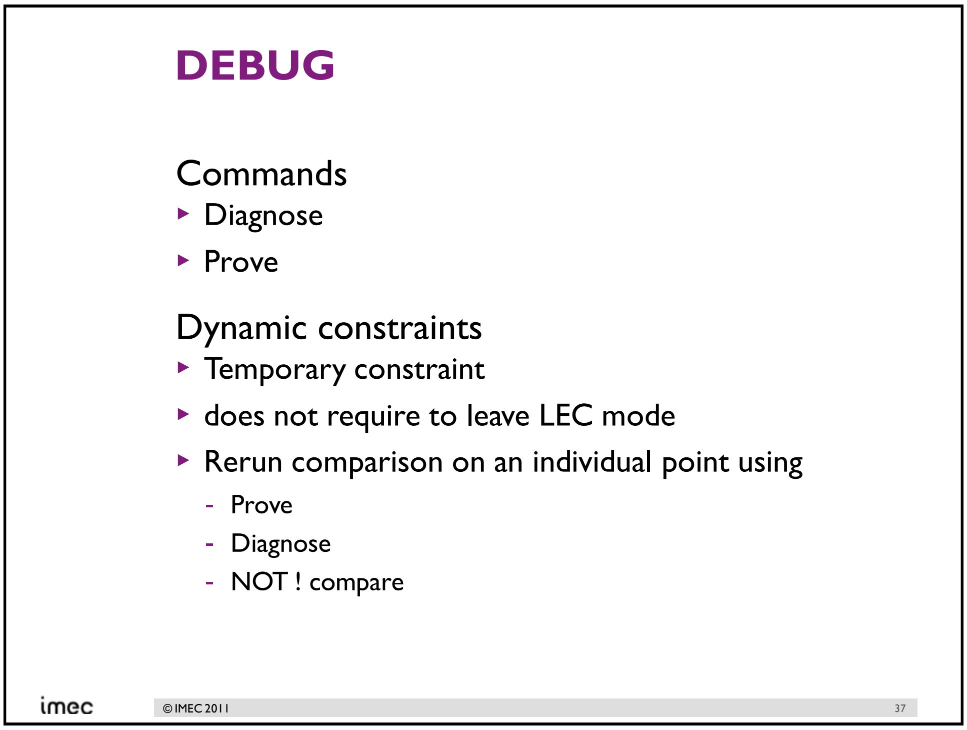
- Summarizes mapping and compare statistics

Report compare data

- Detailed lists

Report verification

- checklist on quality of verification
- Lists "tricks" you used to make compare work.
  - Floating net handling
  - Pin constraints
  - ...

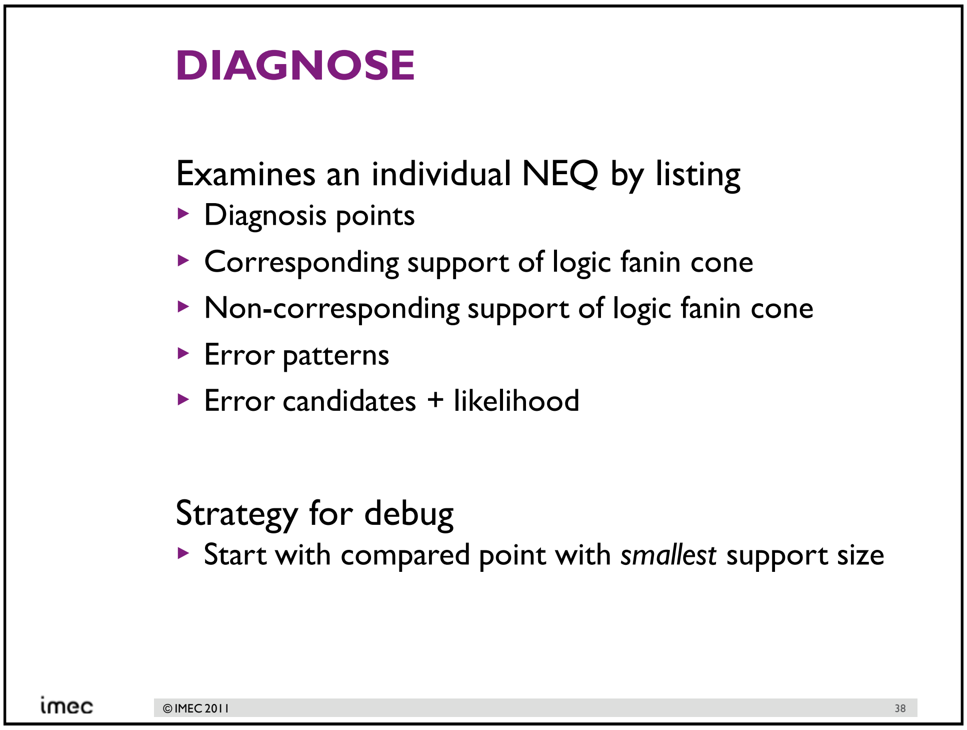


DEBUG

Commands
- Diagnose
- Prove

Dynamic constraints
- Temporary constraint
- does not require to leave LEC mode
- Rerun comparison on an individual point using
  - Prove
  - Diagnose
  - NOT ! compare

imec © IMEC 2011 37

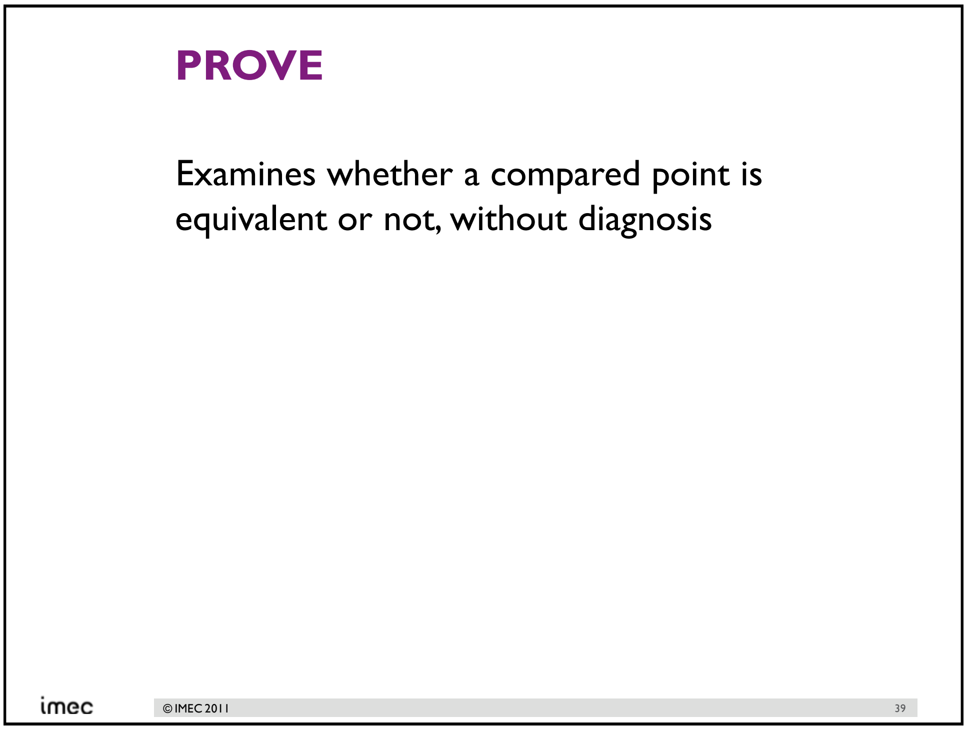


DIAGNOSE

Examines an individual NEQ by listing
- Diagnosis points
- Corresponding support of logic fanin cone
- Non-corresponding support of logic fanin cone
- Error patterns
- Error candidates + likelihood

Strategy for debug
- Start with compared point with *smallest* support size

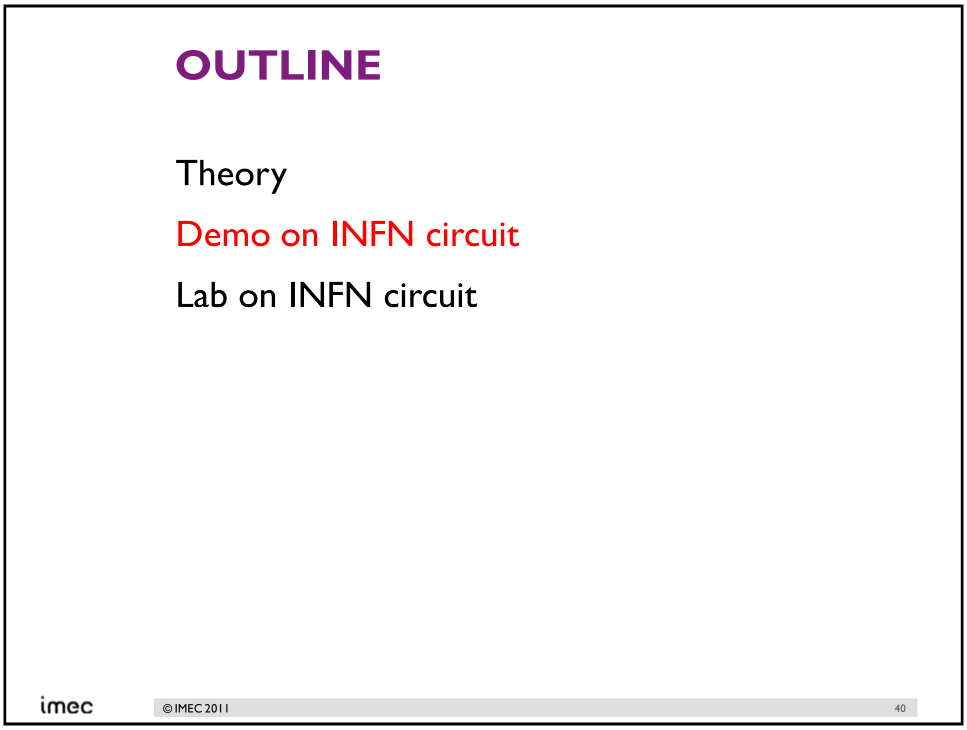
imec © IMEC 2011 38



PROVE

Examines whether a compared point is equivalent or not, without diagnosis

imec ©IMEC 2011 39



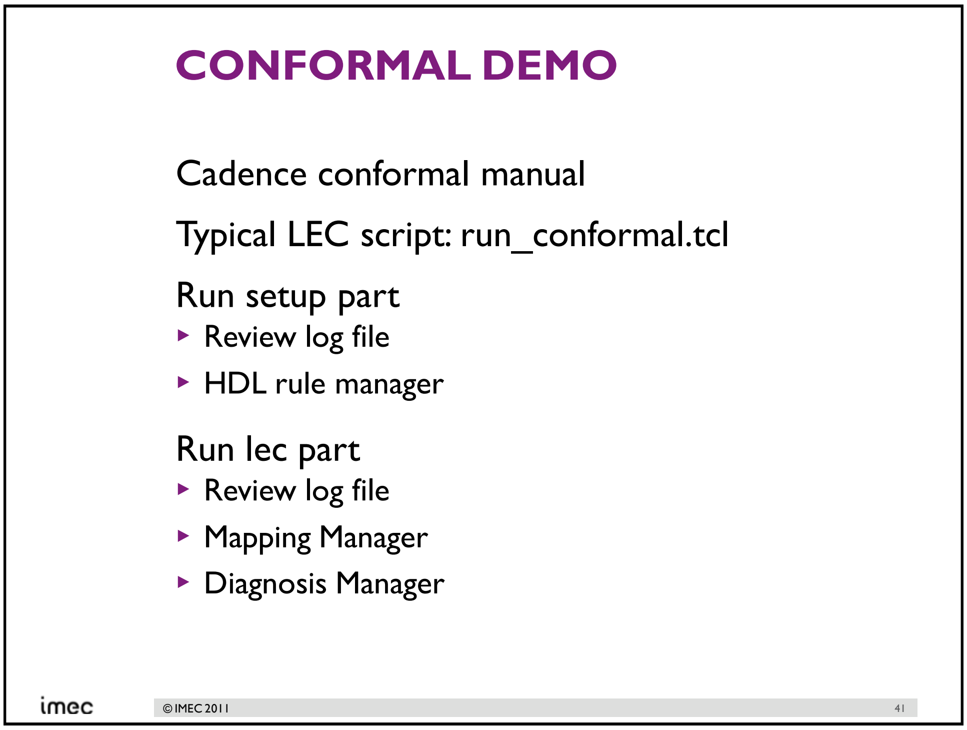
OUTLINE

Theory

Demo on INFN circuit

Lab on INFN circuit

imec ©IMEC 2011 40



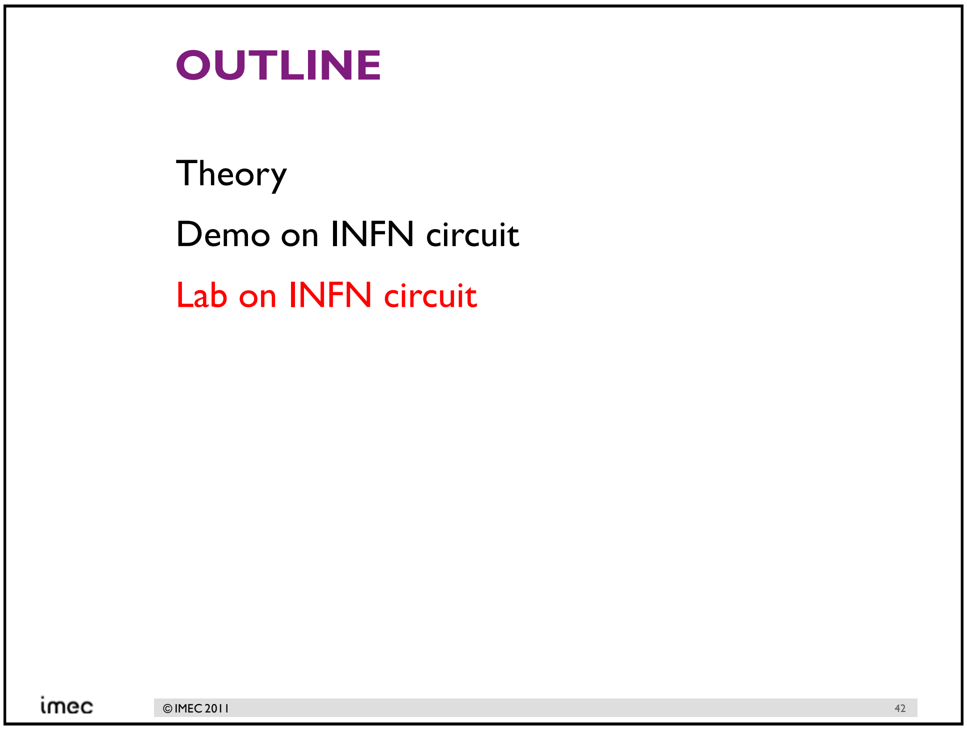
CONFORMAL DEMO

Cadence conformal manual

Typical LEC script: run_conformal.tcl

Run setup part
- Review log file
- HDL rule manager

Run lec part
- Review log file
- Mapping Manager
- Diagnosis Manager

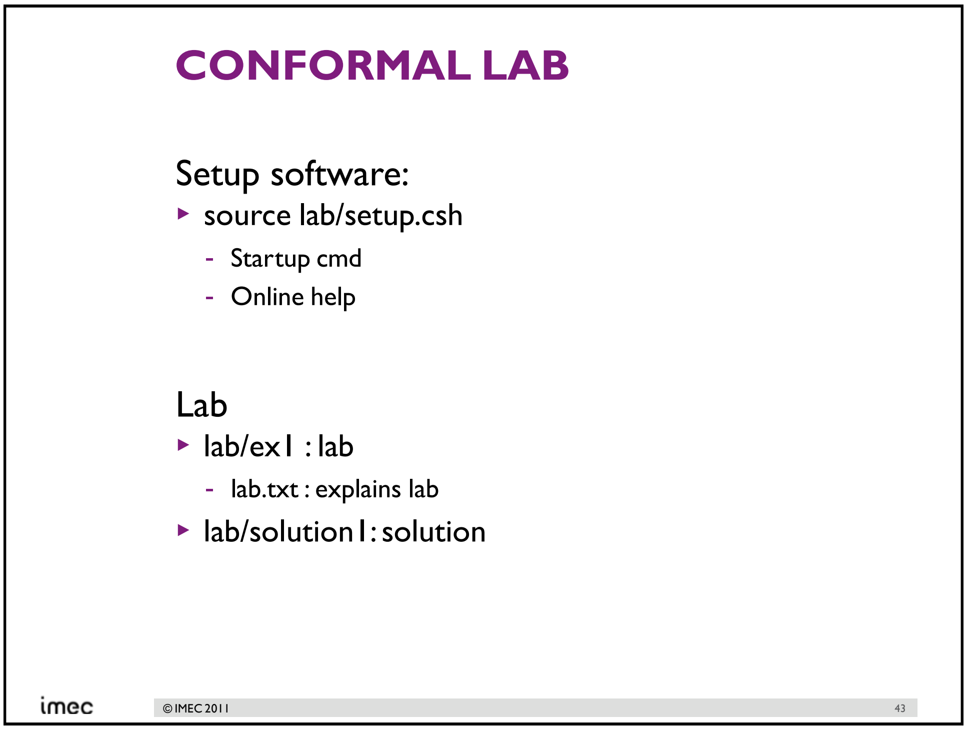


OUTLINE

Theory

Demo on INFN circuit

Lab on INFN circuit



CONFORMAL LAB

Setup software:

- source lab/setup.csh
  - Startup cmd
  - Online help

Lab

- lab/ex1 : lab
  - lab.txt : explains lab
- lab/solution1: solution

imec

© IMEC 2011 43