

INTRODUCTION

Eudes Prado Lopes

Phone: +32 16 28 15 97

Eude.Lopes@imec.be



imec

© IMEC 2011 / CONFIDENTIAL

3

INTRODUCTION (CONT.)

Eudes Prado Lopes

Professional Experience

- ASIC Design Engineer IMEC, Leuven BE – since 2008 (~ last 3 years)
- Design Engineer NXP – San Jose, CA – 2006 - 2008 (~2 years)
- Sr. ASIC Design Engineer Philips Semiconductors – Sunnyvale – San Jose, CA 1999 – 2006 (~7 years)
- Staff Software Engineer - EDA Actel - Sunnyvale CA, 1997-1999 (~ 2 years)

Education

- Ph.D. Computer Science from UPMC (Paris VI) - France ,1996
- M.Sc Electrical Engineering from COPPE –UFRJ – RJ, Brazil, 1992
- Electrical Engineering – from Electronics Dept. UFRJ - RJ, Brazil, 1989

imec

© IMEC 2011 / CONFIDENTIAL

4

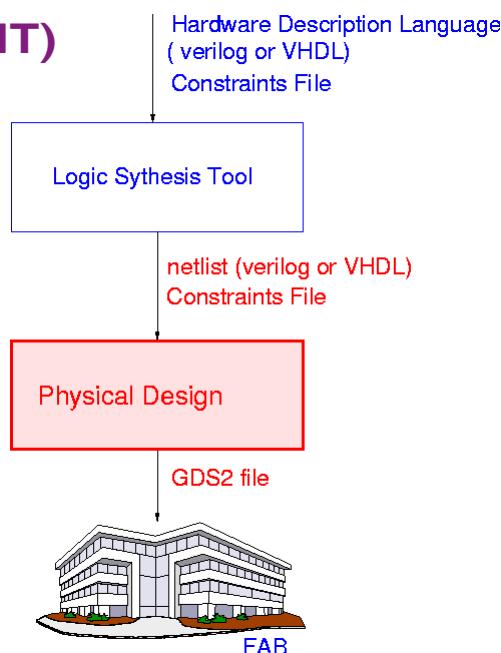
SCOPE

Classical Design Flow for tape-out with UMC 90 nm technology via **Europpractice**

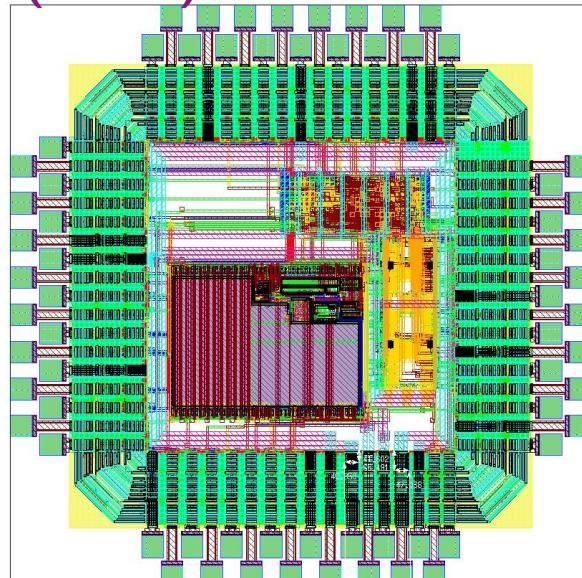
Physical Design

INFN provided Design Case: NanoSOC

SCOPE (CONT)



SCOPE (CONT)



imec

© IMEC 2011 / CONFIDENTIAL

7

COURSE OUTLINE

- Introduction
- Course Outline
- Flow Strategy and Type of Designs
- Data Sources (Gathering/Screening, Creating and Modifying)
- Design Case Facts
- Tools / utilities
- Design Work Flow Overview
 - Physical Design Data Receivables
 - Data Preparation
 - Backend Flow Overview
 - Place and Route
 - Signoff Parasitic Extraction
 - Signoff Static Timing analysis
 - Physical & Logical Verification
- Conclusion

imec

© IMEC 2011 / CONFIDENTIAL

FLOW STRATEGY AND TYPE OF DESIGNS

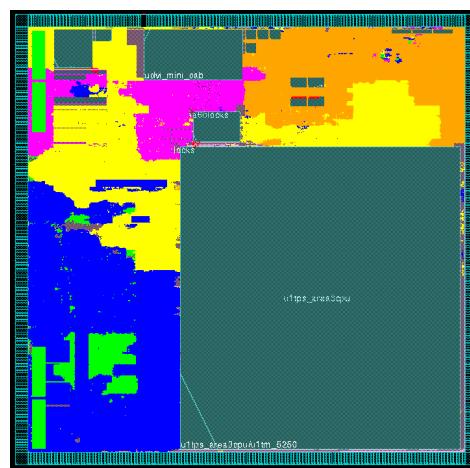
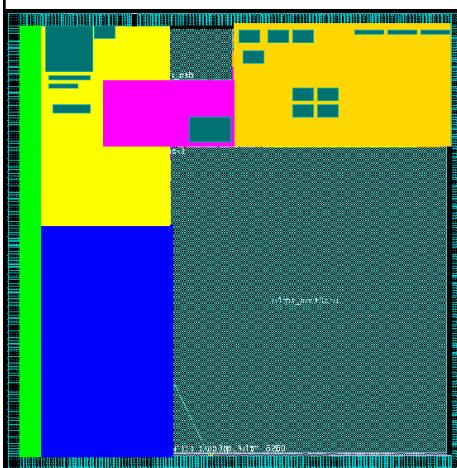
Hierarchical Design X Flat Design

Analog over Digital X Digital over Analog

imec

© IMEC 2011 / CONFIDENTIAL

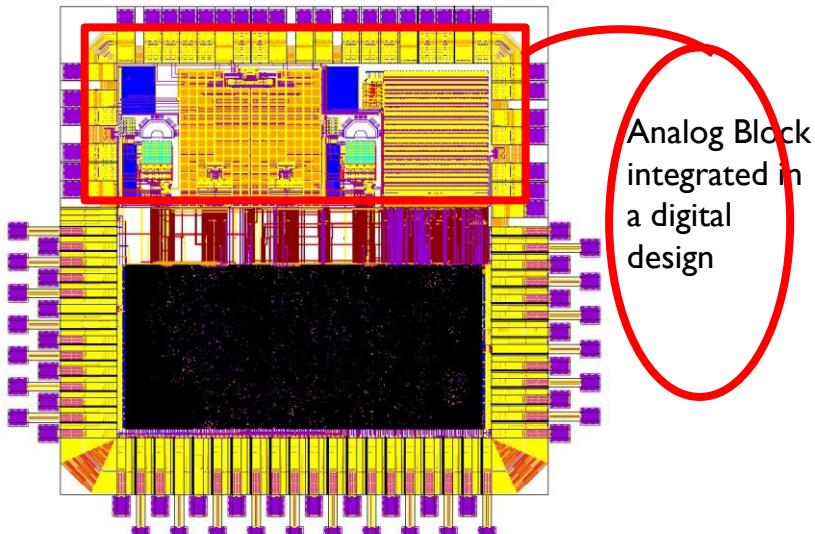
HIERARCHICAL X FLAT



imec

© IMEC 2011 / CONFIDENTIAL

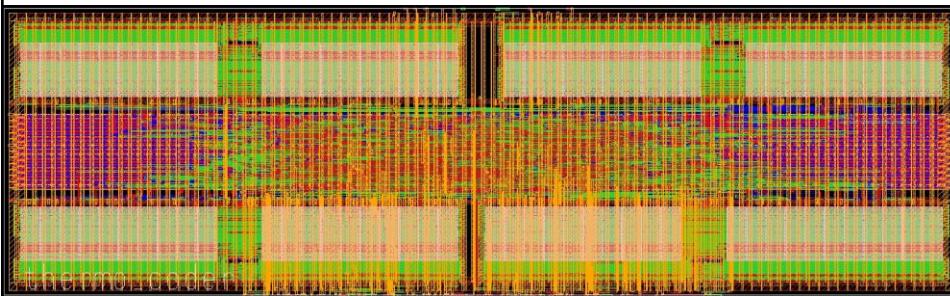
ANALOG OVER DIGITAL VERSUS DIGITAL OVER ANALOG



imec

© IMEC 2011 / CONFIDENTIAL

ANALOG OVER DIGITAL VERSUS DIGITAL OVER ANALOG (CONT.)



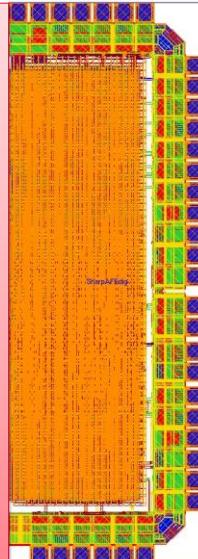
Pure Digital Block to be inserted in a larger chip containing mainly analog blocks and IOs

imec

© IMEC 2011 / CONFIDENTIAL

ANALOG OVER DIGITAL VERSUS DIGITAL OVER ANALOG (CONT)

Analog Design



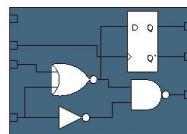
Digital Block with related IOS that is integrated with analog design with its related IOS .
Connections by abutment and/or limited hand editing

imec

© IMEC 2011 / CONFIDENTIAL

DATA SOURCES (GATHERING/SCREENING, CREATING AND MODIFYING)

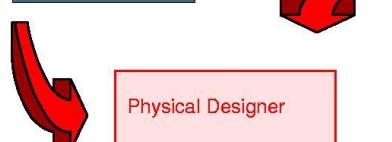
Front End Designer:
Netlist +
Constraints



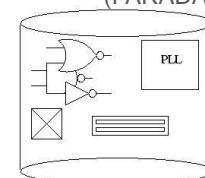
Technology Provider
(UMC)



Physical Designer



IP providers
(FARADAY)



imec

© IMEC 20

DATA FROM FRONTEND

➤ Netlist (verilog or VHDL)

➤ Design Constraints (.sdc)

imec

©IMEC 2011 / CONFIDENTIAL

SCREENING NETLIST AND CONSTRAINTS

(1)

- report path exceptions generates path_exceptions.rpt, This file reports:
 - False Paths
 - Multi Cycle Paths
- checkNetlist generates checkNetlist.rpt, This file reports:
 - @ Top Level Netlist:
 - Floating Ports;
 - Ports connected to multiple Pads;
 - Ports connected to core instances.
 - @ Instance Level:
 - Output pins connected to Power Ground net;
 - Floating Instance terminals;
 - Dont Use Cells Used in Design;
 - Instances with input pins tied together;
 - TieHi/Low term nets not connected to instance's PG terms.

imec

©IMEC 2011 / CONFIDENTIAL

SCREENING NETLIST AND CONSTRAINTS

- reportGateCount

(2)

Generates reportGateCount.rpt. This file reports general statistics of the design such as

- Overall Standard Cells Gate area
- Modules List and
- # of gates cells and area per Module

- get high fanout nets

see NanoSOC_1_fanout_initial.rpt

generates NanoSOC_1_fanout_initial.rpt. This file reports nets fan-out above -fanout_limit parameter

- check unique command checks the uniqueness of the netlist.

Pre-layout netlist can have modules replicated into multiple insts.

Backend initial netlist must have unique module names.

Otherwise netlist must be "uniquified".

Utility "uniquifyNetlist" may be invoked prior to backend flow.

imec

©IMEC 2011 / CONFIDENTIAL

SCREENING NETLIST AND CONSTRAINTS

(3)

* check for unconnected inputs generates in the encounter log file a list of un connected

inputs or:

:INFO: 0 unconnected inputs found

* calculate standard cell area generates in the encounter log file a short report as follows

Calculated area: 1086137.49675

* check clocks generates in the encounter log file a list of sequential instances; i.e.:

####Sequential instances found:####

RAM090_dp8by256: 1 instances

QDFZCRBXI: 462 instances

DFZXI: 37 instances ...

* report floorplan information generates in the encounter log file overall floorplan data

Width height of the design and also average densities for the design. In order to see these results look into "Floorplan information" within the log file

imec

©IMEC 2011 / CONFIDENTIAL

DESIGN SPECIFIC VERSUS LIBRARY SPECIFIC DATA

~/layout/Design_Data/ ~/layout/Library_Data/

NETLIST/

SDC/

MMMC/

FLOORPLAN/

IOFILE/

CTS_CONFIG/

CAPTABLE/

LIB/

CELTIC/

GDS/

LEF/

Firelce/

imec

© IMEC 2011 / CONFIDENTIAL

DATA CONFIGURATION MANAGEMENT (DATA VERSIONING)

- Create ~/received and ~/sent directories to store data;
- Add timestamp and prefix to subdirectories and to files. Quick way to create sort able timestamp:

% date +%y%m%d%H%M%n
- Add README files in data directories with the origin of files;
- Log md5sum tags of relevant files to README files;
- Create ~/received/EMAILS and ~/sent/EMAILS directories with timestamp attached to text files of relevant emails. Moreover the ones holding data files;

imec

© IMEC 2011 / CONFIDENTIAL

DESIGN CASE FACTS

- Design IPs and other Objects
- Design Metrics
- Design Pictures

imec

© IMEC 2011 / CONFIDENTIAL

IP AND OTHER OBJECTS

IP List:

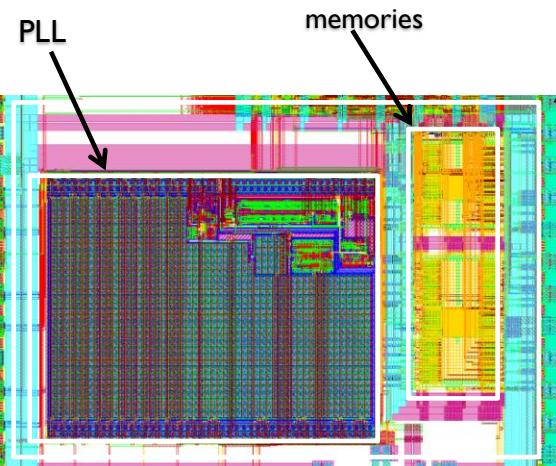
Typical External IP :Analog Blocks, PLL, Memories,
Register Files, Special cells, Logos and other gizmos.

Typical Library/Fab IP: Standard Cells, IO cells, Bond Pads,
Seal Ring;

imec

© IMEC 2011 / CONFIDENTIAL

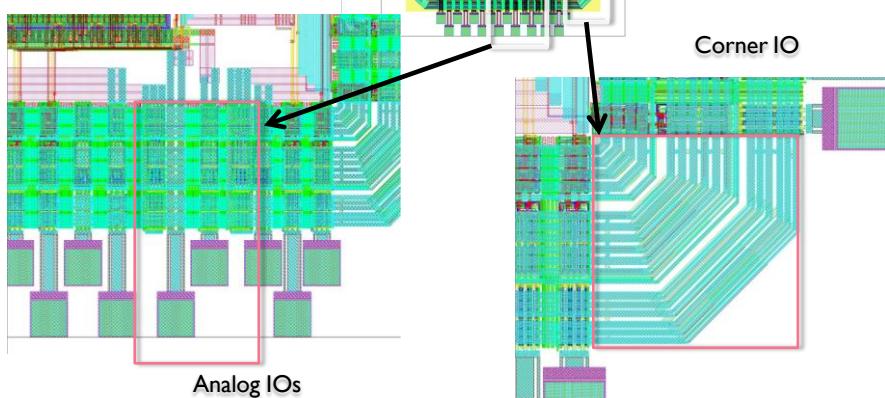
IP PICTURES (1)



imec

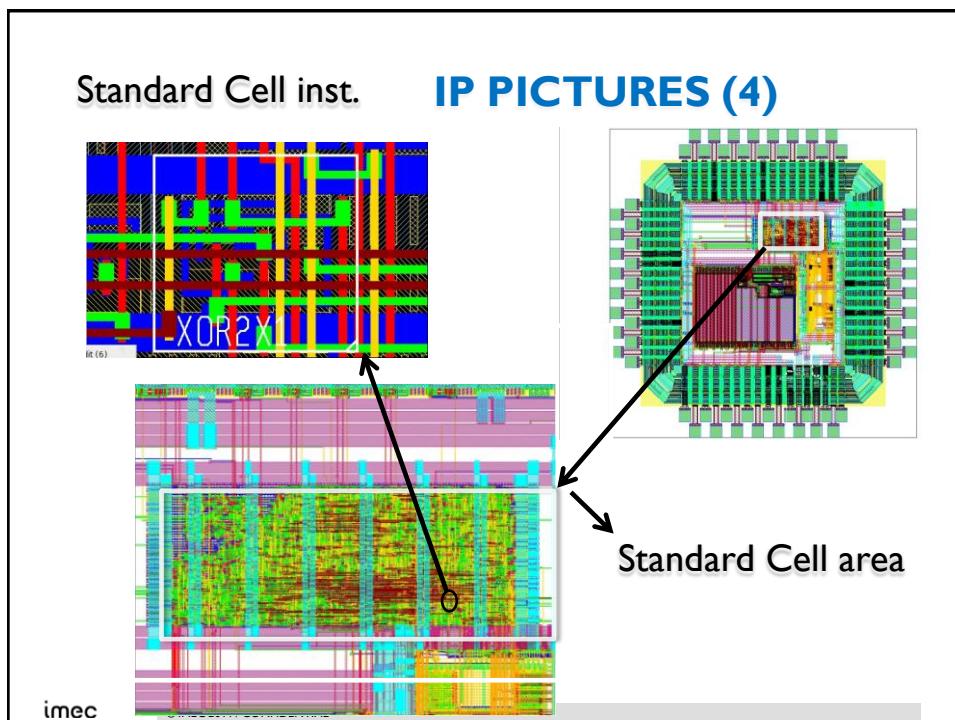
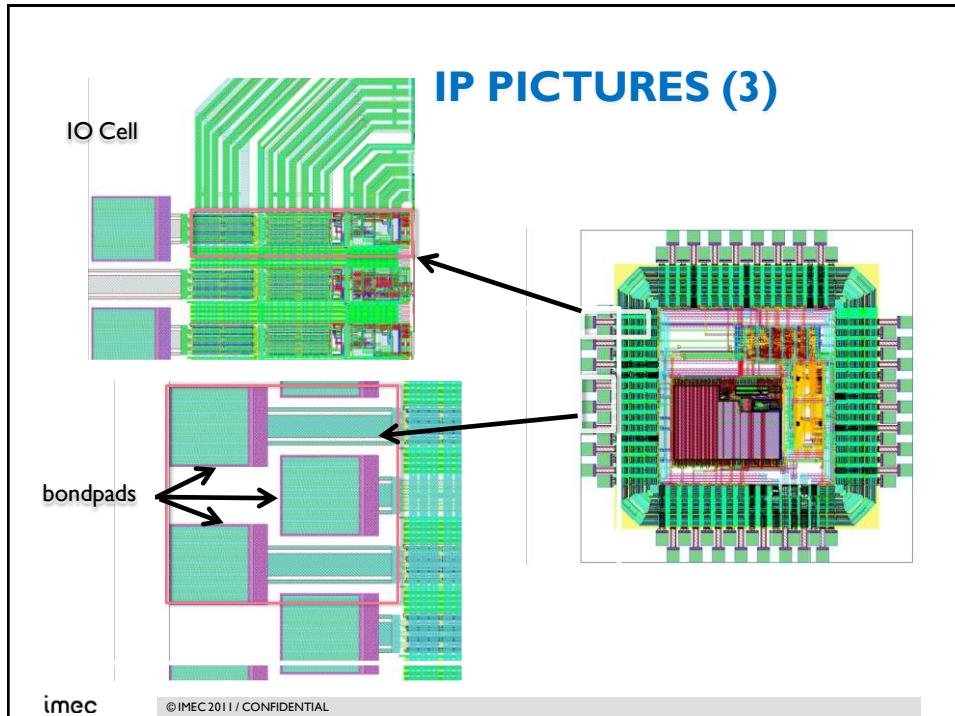
© IMEC 2011 / CONFIDENTIAL

IP PICTURES (2)



imec

© IMEC 2011 / CONFIDENTIAL



DESIGN CASE METRICS

Number of standard Cells: 3011

Area of Standard Cells: 352 Gates 31697 um^2

Area of Placeable Std Cells: $442.9 \times 159.6 = 70376.81 \text{ um}^2$

Utilization : 45% Utilization in area (626484 878395) (1067006 1037552) = 45.06%

Die Area: $912.24 \times 912.24 = 832181.8176 \text{ um}^2$

Chip Area incl.Bond Pads = $1720 \times 1720 = 2958400 \text{ um}^2$

IO+ bond pads = 2126218.1824

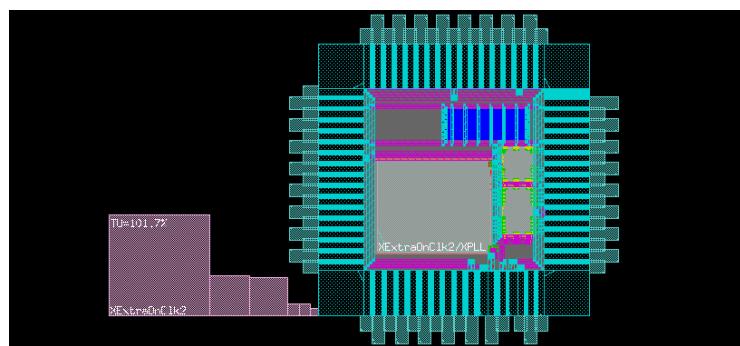
28% total chip area is core, 72% is IO + bond pads

imec

© IMEC 2011 / CONFIDENTIAL

DESIGN CASE PICTURES(I)

Floorplan

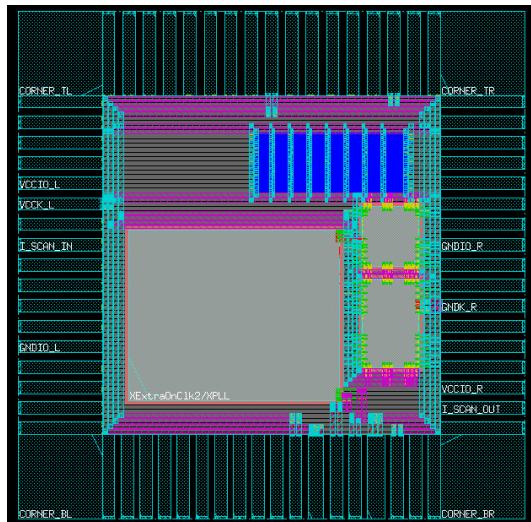


imec

© IMEC 2011 / CONFIDENTIAL

DESIGN CASE PICTURES(2)

Power Plan



imec

© IMEC 2011 / CONFIDENTIAL

DESIGN CASE PICTURES(3)

Design After Placement

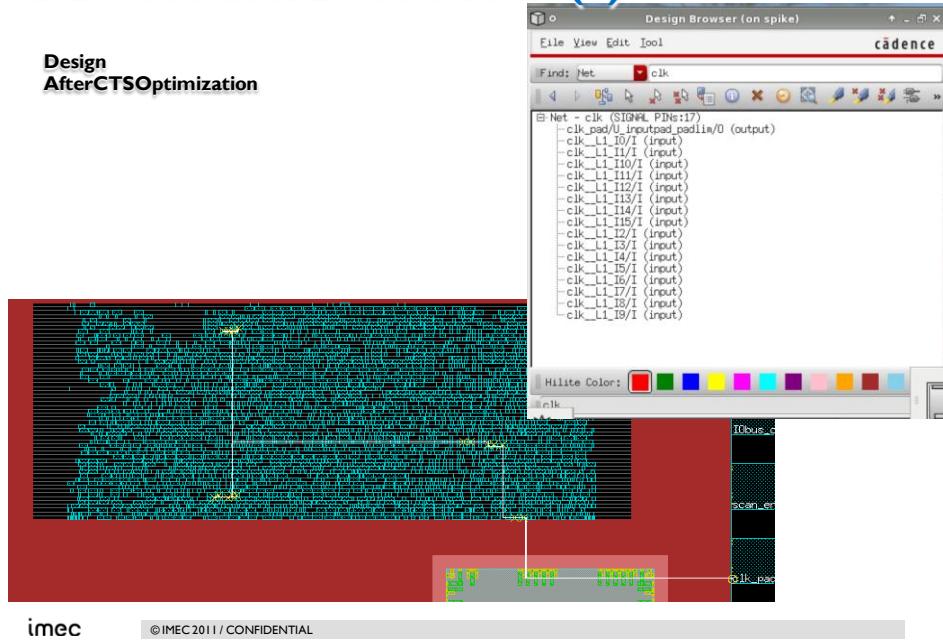


imec

© IMEC 2011 / CONFIDENTIAL

DESIGN CASE PICTURES(4)

**Design
AfterCTSOptimization**



imec

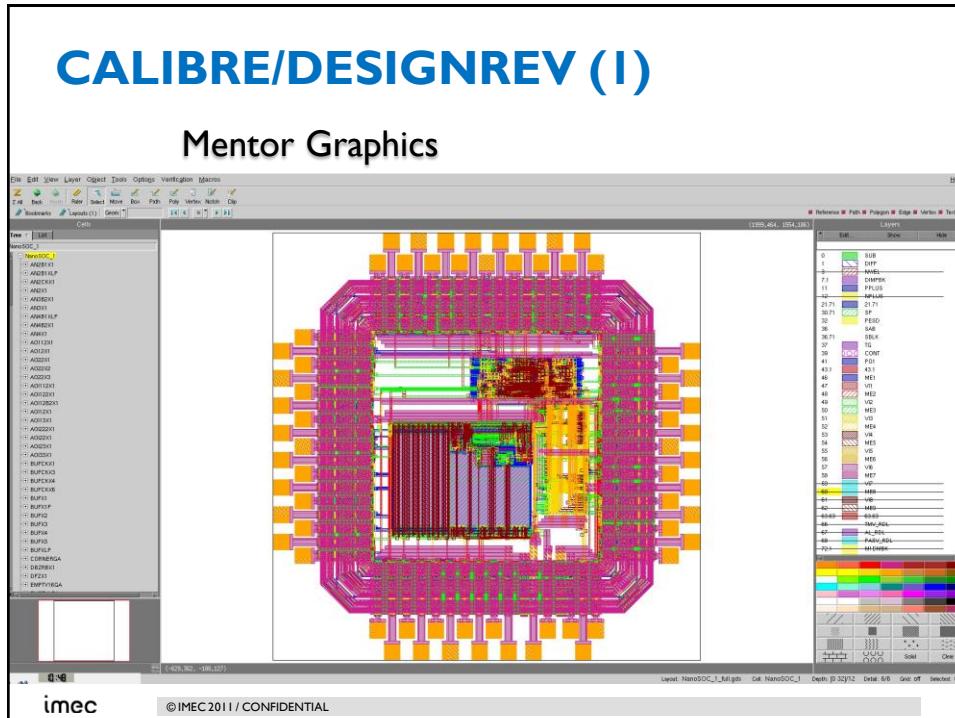
© IMEC 2011 / CONFIDENTIAL

TOOLS AND UTILITIES

- Caliber/DesignRev (Mentor Graphics)
- Encounter (Cadence Design systems)
- generateCapTbl (Cadence Design Systems)
- Fire&Ice (Cadence Design Systems)
- ETS (Cadence Design Systems)
- Caliber DRC/LVS/ANT (Mentor Graphics)
- Conformal (Cadence Design Systems)

imec

© IMEC 2011 / CONFIDENTIAL



CALIBRE/DESIGNREV (2)

Gds2 Viewer

GDS2 Editor: Allow quick modifications in the gds2

Check DRC violations using RVE: Tools -> Start RVE

Setup:

```
source  
/imec/software/mentor/release2007/setup/calibre_v2007.2_34.24.csh
```

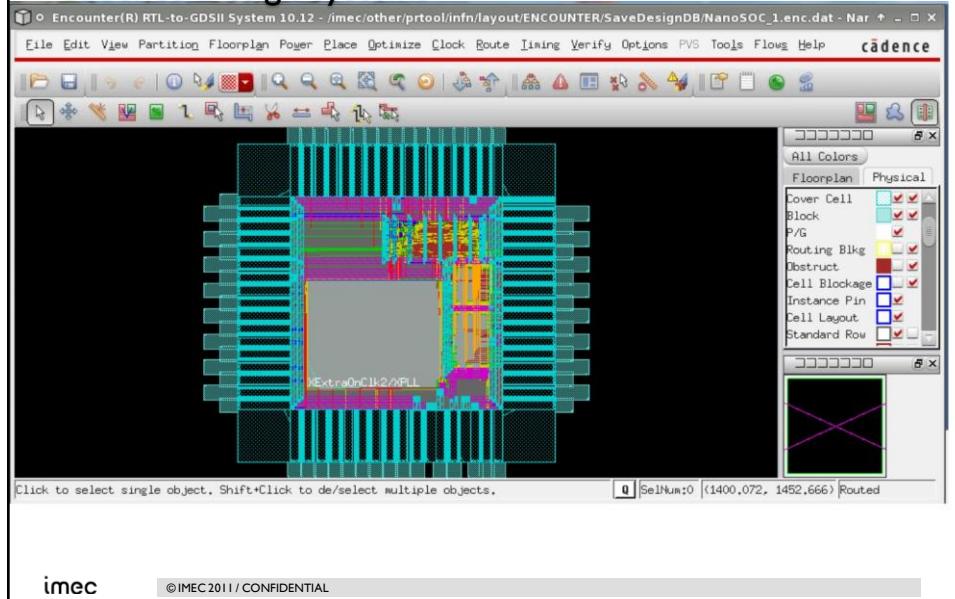
Run:

```
Calibredrv -m <file.gds> -l <layerprops>
```

© IMEC 2011 / CONFIDENTIAL

ENCOUNTER (1)

Cadence Design Systems

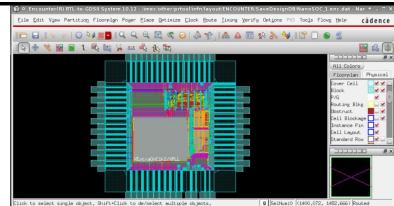


imec

©IMEC 2011 / CONFIDENTIAL

ENCOUNTER (2)

Floorplanning Place and Route tool



Cadence Data Base (LEF DEF and Verilog based).
Export GDS and OASIS

Tcl based commands

Setup:

```
source / source
/imec/software/cadence/release2011/setup/edi_v10.12.000.csh p
```

Run:

```
encounter
```

imec

©IMEC 2011 / CONFIDENTIAL

FIRE & ICE

Cadence standalone Extraction QRC - 64-bit Parasitic Extractor

Setup:

```
source /imec/software/cadence/release2011/setup/ext_v10.11.149.csh c
```

Run:

```
qrc -multi_cpu 6 -cmd qrcExtworst.cmd
```

imec

©IMEC 2011 / CONFIDENTIAL

ETS – ENCOUNTER TIMING SYSTEM

Cadence (used standalone) Timing Engine

.cmd based commands.

Setup:

```
source /imec/software/cadence/release2011/setup/ets_v10.12.000.csh c
```

Run:

```
ets -64 -nowin -overwrite \
-init ${STAGE}/cmds/etsTim_func_bccbest.cmd \
-log ${STAGE}/logs/etsTim_func_bccbest.log \
-cmd ${STAGE}/cmds/etsTim_func_bccbest.cmdlog
```

imec

©IMEC 2011 / CONFIDENTIAL

CALIBRE DRC/LVS/ANT MENTOR GRAPHICS

Mentor Graphics (used standalone) physical verification tool set.

.cmd based commands.

Setup:

```
source  
/imec/software/mentor/release2007/setup/calibre_v2007.2_34.24.csh
```

Run: i.e. DRC checks:

```
/imec/other/invosoft/Software/dracula/COM/gen_report.perl  
NanoSOC_1.sum NanoSOC_1_DRC_RES.txt NanoSOC_1.rpt NanoSOC_1.log
```

imec

© IMEC 2011 / CONFIDENTIAL

UTILITIES (I)

gen_report.perl

Pearl script generates a condensed DRC report from Calibre summary

Run after DRC checks i.e.:

```
/imec/other/invosoft/Software/dracula/COM/gen_report.perl  
NanoSOC_1.sum NanoSOC_1_DRC_RES.txt NanoSOC_1.rpt NanoSOC_1.log
```

imec

© IMEC 2011 / CONFIDENTIAL

UTILITIES (2)

generateCapTbl – Cadence Design Systems

Cadence executable that reads in the interconnect technology (ICT) process file and generates a Capacitance table file.

See details in :

/imec/other/prtool/infn/layout/Library_Data/FireIce/README

Setup:

```
cd /imec/other/prtool/infn/layout/Library_Data/FireIce/  
source /imec/software/cadence/release2010/setup/edi_v10.10.csh p
```

Run before starting Encounter P&R i.e.:

```
generateCapTbl \  
-ict ./G-DF-LOGIC90N-1P9M2T1F-LOW_K_POLY1.5K_FIRE_AND_ICE-LPE-T.3-P2.ict \  
-lef ./header9m126_V55.lef -output UMC_1P9M2T1F.CapTbl
```

imec

©IMEC 2011 / CONFIDENTIAL

CONFORMAL

– Cadence Design Systems

Cadence.

See details in :

/imec/other/prtool/infn/..../README

Setup:

Run to compare pre layout X post layout gate level netlist i.e.:

```
cd /imec/other/prtool/infn/layout/..../  
source /imec/software/cadence/release2010/setup/edi_v10.10.csh p
```

Add here conformal ruin

imec

©IMEC 2011 / CONFIDENTIAL

DESIGN WORK FLOW OVERVIEW(1)

- Gathering the Physical Design Data;
- Screen Front-end Data and Create Initial Backend Data;
- Build Floorplan, Place, Route, STA and Stream Out GDS data (Encounter);
- Sign Off Extraction (Fire&Ice);
- Sign Off STA in ETS;
- Final Verification. (Calibre and Conformal)

imec

© IMEC 2011 / CONFIDENTIAL

DESIGN WORK FLOW OVERVIEW(2)

- Gathering the Physical Design Data
 - Create Data (Cap Tables, Lef of some IP, etc)
 - Link Data (Libraries and IP)
 - Copy/Modify Data (Libraries and IP)
- Screen Front End Data Create Initial Backend Data
 - Netlist
 - Constraints
 - IO file
 - Floorplan

imec

© IMEC 2011 / CONFIDENTIAL

DESIGN WORK FLOW OVERVIEW(3)

Build Floorplan, Place, Route and Stream Out GDS (Encounter)

- Set Up Design Basic Variables: `setupDesign.tcl`
- Load useful Scripts: `variables_usefullScripts.tcl` `userEnableGiftScripts.tcl`
`my_procedures.tcl`
- Create netlist of power connections in the database: `define_global_connections.tcl`
- ▶ CheckTiming of the pre-layout netlist (ideal clock, no physical information): `check_timing_nowireloads.tcl`
- Floorplan the design including die dimensions , IO positions, Macros/Gizmos placement, Standard Cell area definition, Power Plan, Bond Pads and labels positioning: `floorplan.tcl`
- Prepare for placement (Create blockages, fences, guides, etc.: `prePlace.tcl`
- Place the Standard Cells and Check timing: `place.tcl`
- Build the Clock trees and Check timing: `CTS.tcl`
- Optimize the design after CTS and Check timing: `CTSOpt.tcl`
- Route the signal Nets and Check timing: `nanoRoute.tcl`
- Check timing for test mode: `postRouteCheckTiming_test_mode.tcl`
- Optimize the design after routing and Check timing: `postRouteOpt_hold_for_test_mode.tcl`
- Add decap cells and Fillers cells in the standard Cell area and Check timing: `addDCAPFillers.tcl`
- Final Trim of the power nets: `trimPower.tcl`
- Native Extraction (extractRC), Sref out, and native STA : `nativeSref_out.tcl` `nativeSignOffSTA.tcl`
- Stream Out final gds2: `streamout_full.tcl`
- Save Final design (including def file for sign Off extraction: `saveDesign ${savedir}/${topCell}.enc`, ...

imec

©IMEC 2011 / CONFIDENTIAL

DESIGN WORK FLOW OVERVIEW(4)

Sign Off Extraction (Fire&Ice)

Run standalone Fire&Ice Tool on def database:

`/imec/other/prtool/infn/FIREANDICE/SCRIPTS/run.cmd`

Sign Off STA using ETS

Run standalone ETS tool on def database and spfs generated by Fire&Ice

`source /imec/other/prtool/infn/STA/SCRIPTS/run.cmd`

imec

©IMEC 2011 / CONFIDENTIAL

DESIGN WORK FLOW OVERVIEW(5)

Final Verification:

➤ Calibre:

DRC
ANT
LVS/ERC

➤ Conformal

Gate versus Gate formal Proof

imec

© IMEC 2011 / CONFIDENTIAL

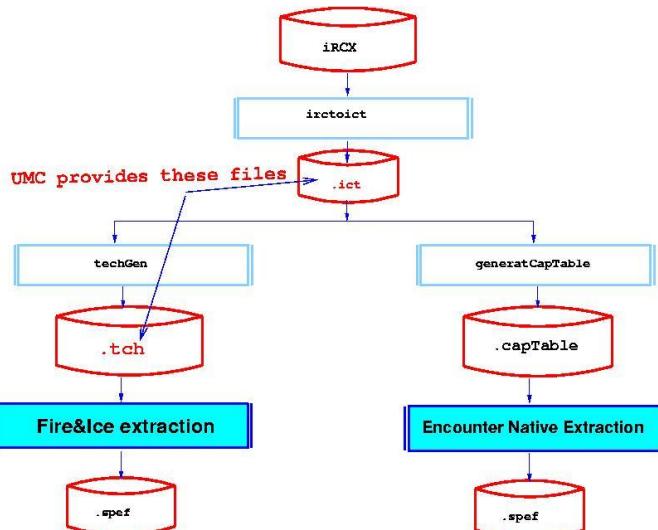
GATHERING PHYSICAL DESIGN DATA - CAP TABLES

Cap Tables are to be used within Encounter in order to allow the tool to do STA with preliminary (not extracted) parasitic delays. Encounter uses the delay information provided by the cap tables until actual parasitic extraction (from routing) is available.

imec

© IMEC 2011 / CONFIDENTIAL

PARASITIC EXTRACTION: NATIVE VERSUS SIGN OFF



imec

© IMEC 2011 / CONFIDENTIAL

GENERATING CAP TABLES- ICT FILES (I)

- Provided by the Library Vendor (i.e. G_DF-LOGIC90N-1P9M2T1F-LOW_K_POLY1.5K_FIRE_AND_ICE-LPE-T.3-P2.ict);
- Accurate Parasitic Extraction requires solving large systems of Maxwell equations (differential equations) ;
- Field Solvers are programs that solve systems of such differential equations using numeric methods;
- Library Vendor provided Files for extraction .ict and .tch files are supposed to be the best data designers can use to perform STA;
- The Library Vendor provided files are in general created using complex and expensive 3D field solver programs in order to allow RC extractors provide reasonably accurate RC data.

imec

© IMEC 2011 / CONFIDENTIAL

GENERATING CAP TABLES- ICT FILES (2)

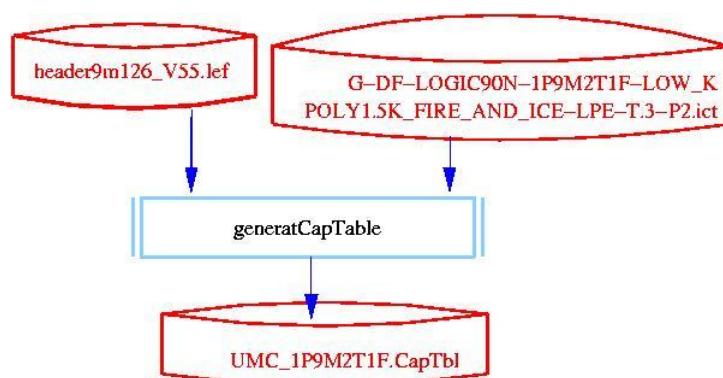
In UMC provided process tech files we note:

- Typical process parameters with WEE (Wire Edge Enlargement) data are used in the ict file ;
- Metal slot and Metal fill are not considered.
- This command file describes POLY thickness =1.5kA without MMC (Metal-Metal capacitor) and only for cell-level extraction.
- The .tch binary file contains the data necessary for Parasitic Extraction (RC delay values extracted from the connectivity within the design)
- .ict files are divided into sections that hold, for each technology layer, information such as:
 - Minimum spacing
 - Minimum Width
 - Height w.r.t. the substrate
 - Thickness
 - Resistivity
 - Inter-layers dielectric constant
 - etc.

imec

© IMEC 2011 / CONFIDENTIAL

CAP TABLE GENERATION FLOW (I)



imec

© IMEC 2011 / CONFIDENTIAL

CAP TABLE GENERATION FLOW (2)

- The CapTbl file holds connectivity delay data in a format (Capacitance and resistances tables) that is more suitable for P&R tools delay calculation engine to work with;
- This is due to the fact that Delay calculation during P&R and optimization phase would be very time consuming if, at each interaction the tool had to do a full RC wiring extraction in order to evaluate wiring delays;
- In order to get a coarse wiring delay annotation and optimization within the optimization loops, the tool uses the CapTables data pre generated for a given technology;
- This part of the flow is called "Native" STA because the STA is being performed with the "default" approach used by the P&R tool;
- Later on the flow, at the final Static Timing Analysis Sign-off phase, the timing engine will be fed with more accurate data (i.e.. full extracted RC results stored in .spf files) so to calculate more precisely the connectivity portion of the delays. These results must corroborate most of the results obtained during "Native" STA. **Nevertheless it is not un-common that some violations that have not been caught before (during "Native" STA) to be highlighted at the end during the Static Timing Analysis Sign-off phase.**

imec

© IMEC 2011 / CONFIDENTIAL

CAP TABLES ADAPTATION

Reading in ict file for 9 layers metal (**G-DF-LOGIC90N-1P9M2T1F-LOW_K_POLY1.5K_FIRE_AND_ICE-LPE-T.3-P2.ict**)

that contains LAYER M1 to LAYER M10 and

reading in lef file for 9 layers metal (**header9m126_V55.lef**) that contains LAYER M1 to LAYER M9
we get

[~/layout/Library_Data/FireIce/UMC_IP9M2T1F.CapTbl](#)

with LAYER M1 to LAYER M10 and VIA CONT,VIA VAI,VIA VA2,...VIA VA9

So we get 10 layers and also 9 vias ! Basically, generateCapTbl utility follows the .ict file.

The ict file contains an extra layer PAD and via VA9 that serves to the extra aluminum layer that is not used to route in encounter. This is called the "redistribution" layer and is typically used when we have pads inside the core. (this is not INFN project design case).

The solution used was to comment out in the generated UMC_IP9M2T1F.CapTbl file

[\(~/layout/Library_Data/FireIce/UMC_IP9M2T1F.CapTbl\)](#)

The following:

➤ LAYER M10 and VIA VA9 definition sections;

➤ M10 Capacitance table sections

This data will NEVER be used in Encounter that will route only to metal 9 layer.

imec

© IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA - LEF FILES (I)

There are two main types of lef files:

- Technology Lef

i.e.: header9m126_V55.lef

- P&R Macro Lef

i.e.: fsd0a_a_generic_core.lef,
RAM090_dp16by256.lef

imec

©IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA - LEF FILES(2)

In the Design case of NanoSOC design changes were necessary to be done in header9m126_V55.lef

Also a new P&R Macro Lef have been manually created for the short and long staggered bond pads:

UMC90_IMEC_PAD_60X60_32KA_F_V1X1_stagge
red_s.lef and
UMC90_IMEC_PAD_60X60_32KA_F_V1X1_stagge
red_s.lef

imec

©IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA - CONFIG FILE

(1)

The config file (NanoSOC.conf) is points to the relevant data used within Encounter
 /imec/other/prtool/infn/layout/ENCOUNTER/CONFIG/NanoSOC.conf

Timing libraries for all IPs (Std Cells, Memories and IOs)

```
set rda_Input(ui_timelib,min)
    ../../Library_Data/LIB/fsd0a_a_generic_core_ff1p1vm40c.lib \
    ../../Library_Data/LIB/fod0a_b25_33vt_generic_io_ff1p1vm40c.lib \
    ../../Library_Data/LIB/RAM090_dp16by256_BC.lib \
    ../../Library_Data/LIB/RAM090_dp8by256_BC.lib \
    ../../Library_Data/LIB/FXPLL110HD0A_BC.lib \
    ../../Library_Data/LIB/fod0a_b33_t33_analogesd_io_ff1p1vm40c.lib"
set rda_Input(ui_timelib,max)

    ../../Library_Data/LIB/fsd0a_a_generic_core_ss0p9v125c.lib \
    ../../Library_Data/LIB/fod0a_b25_33vt_generic_io_ss0p9v125c.lib \
    ../../Library_Data/LIB/RAM090_dp16by256_WC.lib \
    ../../Library_Data/LIB/RAM090_dp8by256_WC.lib \
    ../../Library_Data/LIB/FXPLL110HD0A_WC.lib \
    ../../Library_Data/LIB/fod0a_b33_t33_analogesd_io_ff1p1vm40c.lib"
```

imec

© IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA - CONFIG FILE

(2)

The config file (NanoSOC.conf) is points to the relevant data used within Encounter

LEF files libraries for all IPs (Std Cells, Memories and IOs) and Technology LEF :

```
set rda_Input(ui_lef_file)
    ../../Library_Data/LEF/header9m126_V55.lef \
    ../../Library_Data/LEF/fsd0a_a_generic_core.lef \
    ../../Library_Data/LEF/FSD0A_A_GENERIC_CORE_ANT_V55.9m126.lef \
    ../../Library_Data/LEF/RAM090_dp8by256.lef \
    ../../Library_Data/LEF/RAM090_dp16by256.lef \
    ../../Library_Data/LEF/fod0a_b25_33vt_generic_io.8m026.lef \
    ../../Library_Data/LEF/FXPLL110HD0A.8m026.lef \
    ../../Library_Data/LEF/UMC90_IMEC_PAD_60X60_32KA_F_VIXI_staggered.lef \
    ../../Library_Data/LEF/fod0a_b33_t33_analogesd_io.8m026.lef"
```

Cadence CelsIC signal integrity analyzer noise library files: (Std Cells Only)

```
set rda_Input(ui_cdb_file,min)
    ../../Library_Data/CELTIC/fsd0a_a_generic_core_ff1p1vm40c.cdb"
set rda_Input(ui_cdb_file,max)
    ../../Library_Data/CELTIC/fsd0a_a_generic_core_ss0p9v125c.cdb"
set rda_Input(ui_cdb_file)
    ../../Library_Data/CELTIC/fsd0a_a_generic_core_tt1v25c.cdb"
```

imec

© IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA AND OTHER CONFIG ISSUES (1)

1)- The reading order of lef files is defined in the .conf file and its important to read at this order:

- a)- technology lef: i.e. header9m126_V55.lef
- b)- core std cells lef: i.e. fsd0a_a_generic_core.lef
- c)- core Antenna std cells lef: i.e. FSD0A_A_GENERIC_CORE_ANT_V55.9m126.lef
- d)- macro/blocks lef: i.e. RAM090_dp8by256.lef.
- e)- io lef: i.e. fod0a_b25_33vt_generic_io.9m126.lef

i.e. if you read the tech lef after the core lef you get an error

if you read the antenna lef before the core lef you get an warning (ENCLF-44)

2)- LEF of the IOS have been modified in order to change

```
"LAYER metal"      by      "LAYER ME"
"LAYER via ;"     by      "LAYER VII ;"
"LAYER via"       by      "LAYER VI"
```

imec

© IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA AND OTHER CONFIG ISSUES (2)

3)- The top cell name of the design should match the top module in the netlist:

```
top cell name: NanoSOC_I
in /imec/other/prtool/infn/layout/ENCOUNTER/CONFIG/NanoSOC.confNanoSOC_I.conf
set rda_Input(ui_topcell) {NanoSOC_I}
in grep "module NanoSOC_I" /imec/other/prtool/infn/layout/Design_Data/NETLIST/NanoSOC.v
module NanoSOC_I ( pad_prom_clk, pad_prom_din, pad_prom_oe, pad_reset, pad_clk,
4)- When reading in the pre-layout netlist Encounter remove all buffers and replace assign commands by a user defined
buffers. Check the simpler buffer name to replace eventual Assign commands in the netlist
grep "MACRO BUF" layout/Library_Data/LEF/fsd0a_a_generic_core.lef | grep X1
MACRO BUFBX1
MACRO BUFBX12 ....
ADD buffer footprint t to NanoSOC.conf or runscript.tcl
in ./layout/ENCOUNTER/SCRIPTS/runscript.tcl
setDoAssign -buffer BUFBX1
grep BUFBX1 ./layout/ENCOUNTER/SCRIPTS/*
in ./layout/ENCOUNTER/CONFIG/NanoSOC.conf
set rda_Input(ui_buf_footprint) {BUFBX1}
grep foot /imec/other/prtool/infn/layout/ENCOUNTER/CONFIG/NanoSOC.conf
set rda_Input(ui_buf_footprint) {}
```

imec

© IMEC 2011 / CONFIDENTIAL

GATHERING PHYSICAL DESIGN DATA AND OTHER CONFIG ISSUES (3)

5)- Its important to Check ERRORS and WARNINGS within encounter.log* during the flow.

Check ERROR:

```
grep ERROR ./layout/ENCOUNTER/RUNDIR/encounter.log3
```

Check WARN:

```
grep WARN ./layout/ENCOUNTER/RUNDIR/encounter.log3
```

```
grep WARN ./layout/ENCOUNTER/RUNDIR/encounter.log3 | wc
```

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING - FLOW

Source .../SCRIPTS/to_check_timing_no_wireloads.tcl

Here the script will do:

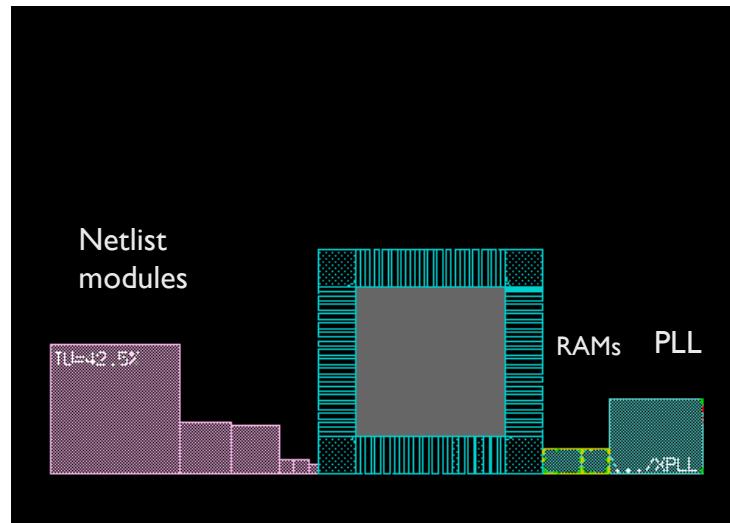
- Setup the initial variable for runscript (**setupDesign.tcl**)
- Set the pointers for the files depicted in **NanoSoc.conf** and actually load them (**loadConfig** and **commitConfig** commands)
- Load many tcl based utilities (procedures) possibly used further in the flow (tcl files : **variables_usefullScripts**, **userEnableGiftScripts**, **my_procedures**) Command **userEnableGiftScripts** (vendor provided scripts)
- Reports and checks: from **report_clocks** to **report_floorplan_information**
- Connect the special nets netlist, mainly analog and digital power net. (**define_global_connections.tcl**)
- Check pre layout timing with **timeDesign -PrePlace** command (**check_timing_no_wireloads**)

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING – FLOORPLANVIEW (I)

Source `../SCRIPTS/to_check_timing_no_wireloads.tcl`

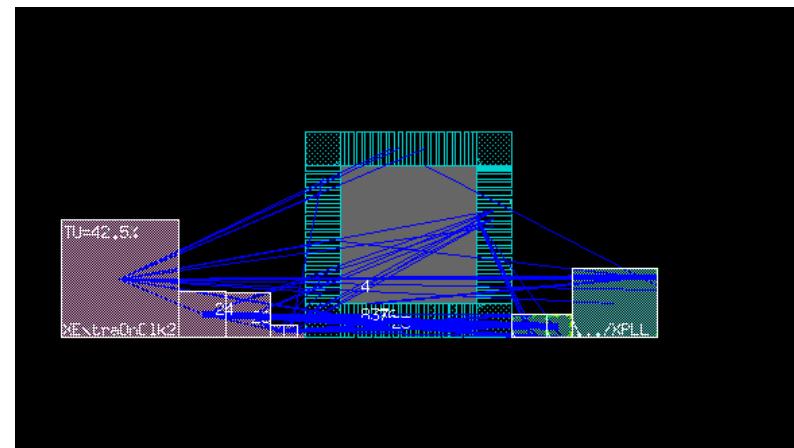


imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING – FLY LINES (2)

Source `../SCRIPTS/to_check_timing_no_wireloads.tcl`

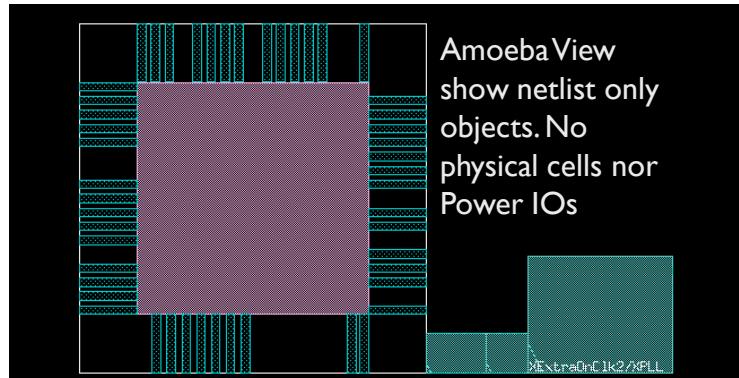


imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING – AMOEBAVIEW (3)

Source/SCRIPTS/to_check_timing_no_wireloads.tcl



imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING – REPORTCLOCKS (4)

Source/SCRIPTS/to_check_timing_no_wireloads.tcl

Checking report_clocks results<CMD> report_clocks

Clock Name	Source	View	Period	Lead	Trail	Generated	Propagated
SYSTEM_CLK	pad_clk	func_wcctyp	10.000	0.000	5.000	n	n
SYSTEM_CLK2	pad_clk2	func_wcctyp	3.000	0.000	1.500	n	n
SYSTEM_CLKPLL	XExtraOnC1k2/XPLL/CLKOUT	func_wcctyp	8.000	0.000	4.000	n	n
VIRTUAL_CLK	-	func_wcctyp	10.000	0.000	5.000	n	n

s

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FROM BEGIN TO PRE LAYOUT TIMING – PRE LAYOUT TIMING REPORT (5)

Source .. /SCRIPTS/to_check_timing_no_wireloads.tcl

Checking pre-layout timing:

```
-----+-----+-----+-----+-----+-----+
|      Setup mode     |    all   | reg2reg | in2reg  | reg2out | in2out  | clkgate |
+-----+-----+-----+-----+-----+-----+
|          WNS (ns):| 2.941  | 2.941  | 6.500  | 3.285  | N/A     | 6.300  |
|          TNS (ns):| 0.000  | 0.000  | 0.000  | 0.000  | N/A     | 0.000  |
| Violating Paths:| 0       | 0       | 0       | 0       | N/A     | 0       |
| All Paths:      | 1652   | 1633   | 30      | 16      | N/A     | 112    |
+-----+-----+-----+-----+-----+-----+
Density: 0.000%
```

Reported timing to dir
 /imec/other/prtool/infn/layout/ENCOUNTER/TIMING_REPORTS/NanoSOC_1_initial
 s

imec

©IMEC 2011 / CONFIDENTIAL

SUMMARY CHECKS (SEE SLIDES “SCREENING NETLIST AND CONSTRAINTS” FOR CHECKS RESULTS AFTER SOURCE .. /SCRIPTS/TO_CHECK_TIMING_NO_WIRELOADS.TCL

- * report_path_exceptions generates path_exceptions.rpt
- * checkNetlist generates checkNetlist.rpt
- * reportGateCount generates reportGateCount.rpt
- * get_high_fanout_nets generates NanoSOC_1_fanout_initial.rpt
- * check_unique command checks the uniqueness of the netlist
- * check_for_unconnected_inputs
- * calculate_standard_cell_area generates in the encounter log file a short report as follows
- * report_floorplan_information generates in the encounter log file floorplan data

imec

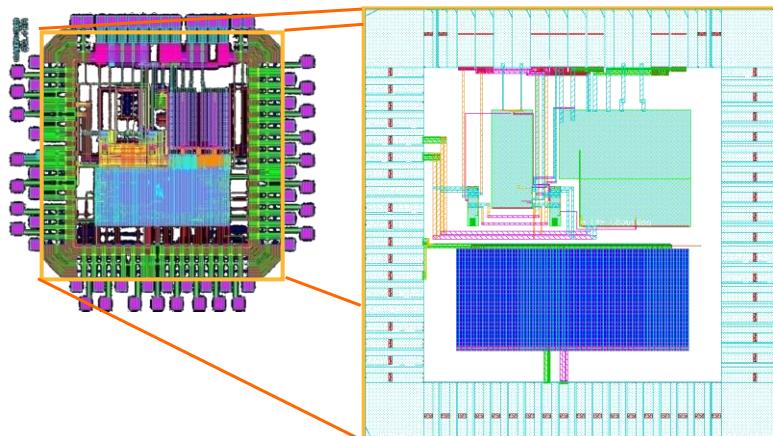
©IMEC 2011 / CONFIDENTIAL

FLOORPLAN - INTRODUCTION

FLOORPLAN VIEWS X LAYOUT VIEWS

Gds views : Full geometric Objects data

Lef or FRAM files: Only boundary and Floorplan, P&R Objects data



imec

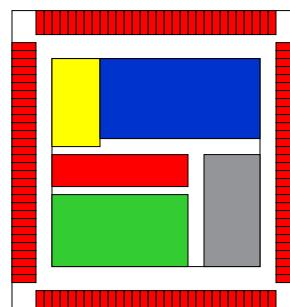
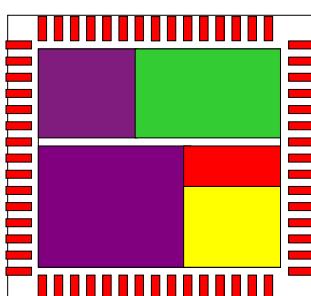
© IMEC 2011 / CONFIDENTIAL

FLOORPLAN - INTRODUCTION

FLOORPLAN – FLOORPLAN STRATEGIES

Core Limited Design

Pad Limited Design

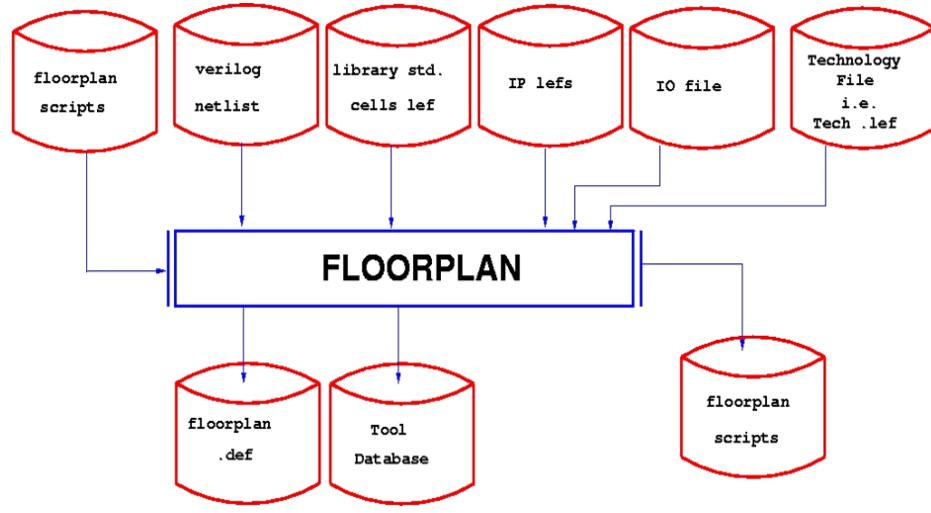


imec

© IMEC 2011 / CONFIDENTIAL

FLOORPLAN - INTRODUCTION

INPUTS & OUTPUTS



imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (I)

Source .../SCRIPTS/floorplan.tcl

Here the script floorplan.tcl will do:

- Create the basic dimensions of the design (**floorPlan** command)
- Load the pre created IO file (**NanoSOC.io**);
- Add the IO fillers (**addIOFillers.tcl**);
- Create no placement areas for standard cells with command **createObstruct**
- Place the IPs (**one PLL and 2 RAMS blocks**) with **placeInst** command;
- Create the power plan with commands: **addRing** and **addStripe** and route special nets with special net router (**sroute** command);
- Save final floorplan with command **saveFPlan**.

imec

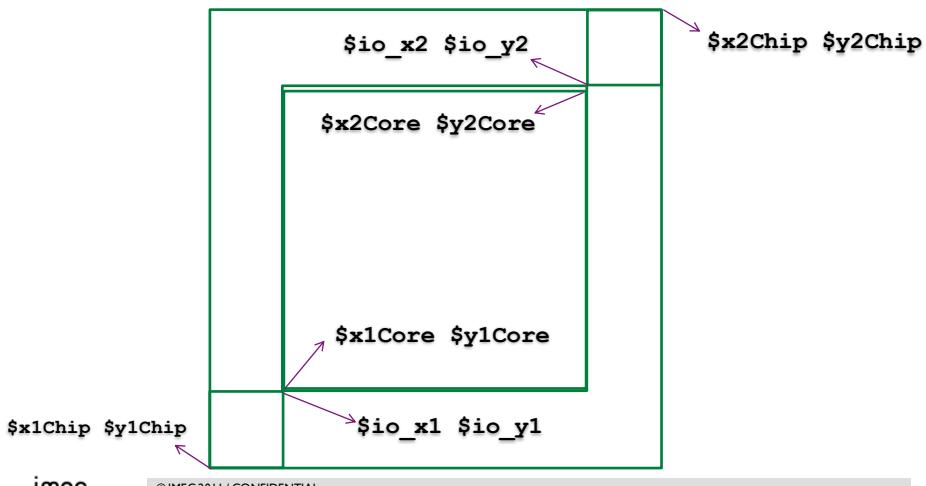
©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (2)

floorPlan command

```
floorPlan -b $x1Chip $y1Chip $x2Chip $y2Chip $io_x1 $io_y1  

$io_x2 $io_y2 $x1Core $y1Core $x2Core $y2Core -fplanOrigin 1
```

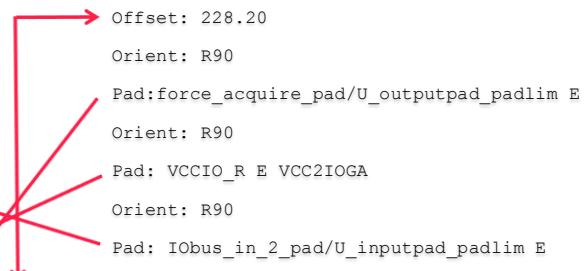
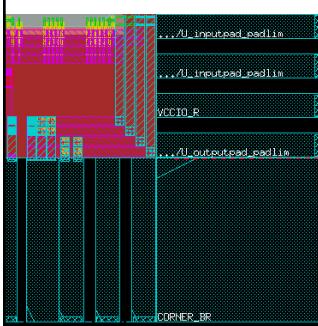


imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (3)

IO file **NanoSOC.io**

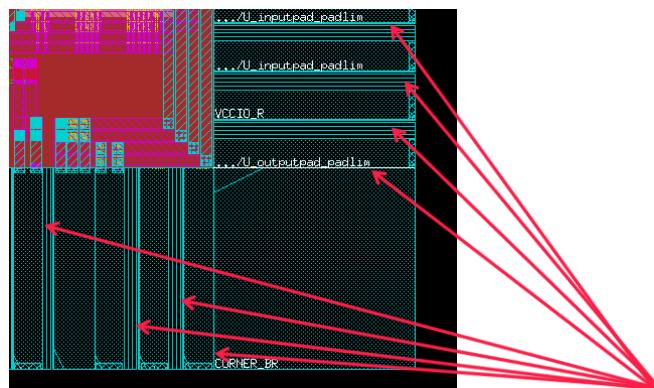


imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT:FLOORPLAN (4)

Add the IO fillers **addIOFillers.tcl**



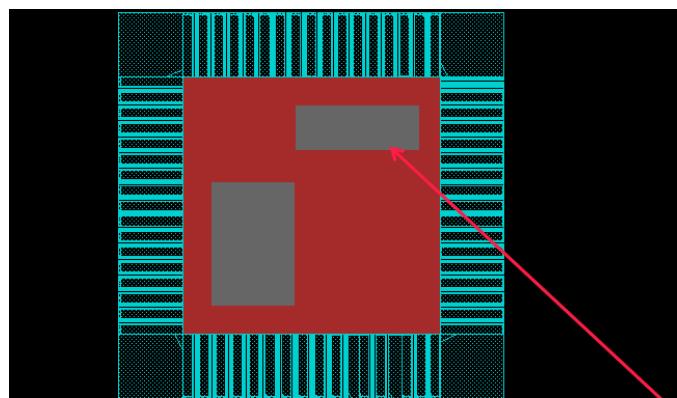
Fillers added here

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT:FLOORPLAN (5)

Create standard cell area with **createObstruct**



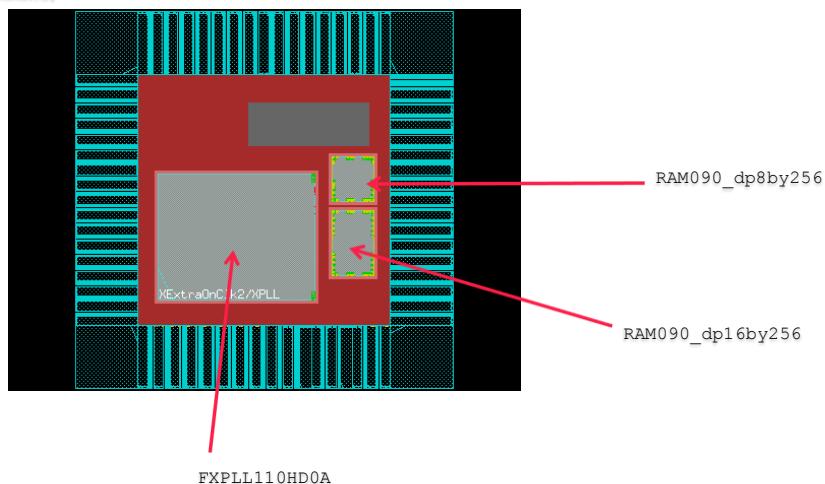
Std. Cell area here

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (6)

Place Macro IP (one PLL and 2 RAMS blocks) with `placeInst` command;



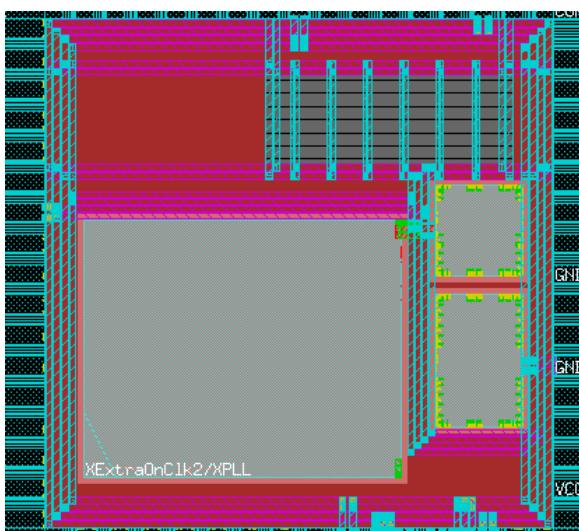
imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (7)

Create the power plan with commands:

`addRing` and `addStripe` and route special nets with special net router (`sroute` command);



```

#- 1 - addRing: Create Ring
close to IOS
#- 2 - sroute: Connect all P&G
IOs to the ring
#- 3 - addRing: Add small core
ring around std cell
#- 4 - addStripe: Connect core
ring to the right of IO ring
#- 5 - addStripe: Connect core
ring to the top side of IO ring
#- 6 - addStripe: Add vertical
metal 8 stripes inside the std
cell core
#- 7 - addRing: Create a ring
around the Rams
#- 8 - sroute: Connect VDDA GND
nets from pads to the ring
#- 9 - addStripe: create the ME2
stripes that sroute command did
not do
#- 10 - addRing: Create a
"ring"
(only top stripe) around the PLL

```

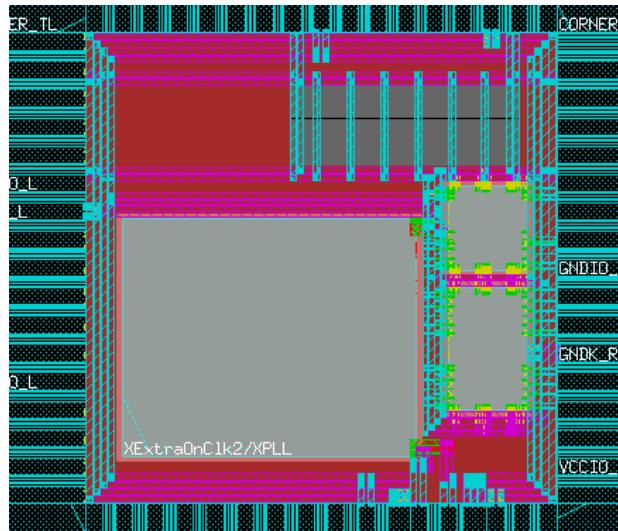
imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (8)

Create the power plan with commands:

`addRing` and `addStripe` and route special nets with special net router (`sroute` command);



```

-- 11 - addRing: Create a
"ring"
(only top stripe) around RAM
16by256
-- 12 - editCutWire: Here
cut the ring
to let pass the analog power
to the PLL
-- 13 - sroute: Connect
U_RAM090_dp8by256
and U_RAM090_dp16by256 to
its ring
-- 14 - addStripe: Create
corner stripes t
on the lower left side of
the PLL left
-- 15 - sroute: First
connect the Analog padss
to the created corner
stripes
-- 16 - sroute: Connect XPLL
to the created corner
stripes
-- 17 - addStripe: create
the ME2
that the sroute command did
not do

```

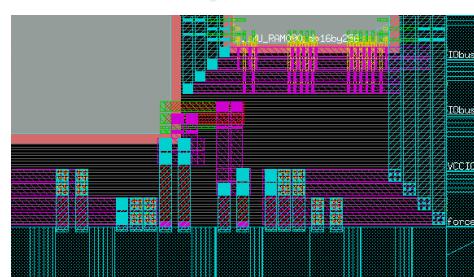
imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: FLOORPLAN (9)

Create the power plan with commands:

Details of the analog IOs connections



Digital ring is cut in
order to not interfere
with analog
connections.
The power scheme is
suggested in the
Faraday App. Note

90_nm_1.0_V_3.3_V_Analog_ESD_IO_Application_Note_v1.0.pdf Application Note

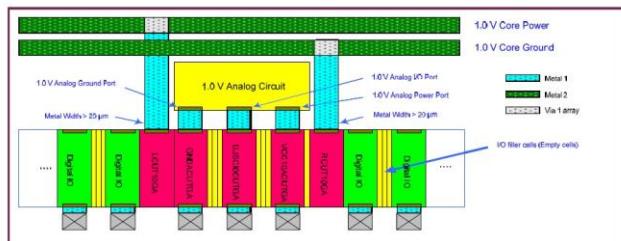


Figure 4-4. Example of Using Faraday 90 nm 1.0 V Analog ESD I/O Set

RUNSCRIPT: PLACEMENT (1)

Here the script will:

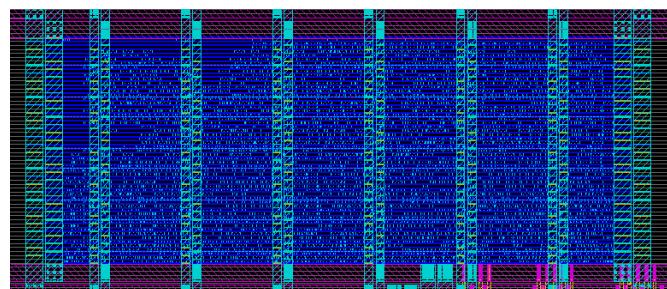
- Source `../SCRIPTS/place.tcl` to place and optimize the design with command: `placeDesign -inPlaceOpt`
- Run native STA with command `(timeDesign);`
- Run sroute to route the Standard Cells rails (`sroute` command);
- Check the placement results with `checkDesign` command;
- Save final placement database with command `saveDesign`.

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: PLACEMENT (2)

Here a snapshot of the standard cell area after placement:



```

encounter 133> Design has 1 cell with cell padding or block halo.
Begin checking placement ... (start mem=592.1M, init mem=592.1M)
*info: Placed = 2961
*info: Unplaced = 0
Placement Density:48.18%(30057/62384)
Finished checkPlace (cpu:total=0:00:00.2, vio checks=0:00:00.0;mem=592.1M)
encounter 133> Design has 1 cell with cell padding or block halo.
Begin checking placement ... (start mem=592.1M, init mem=592.1M)
*info: Placed = 2961
*info: Unplaced = 0
Placement Density:48.18%(30057/62384)
Finished checkPlace (cpu:total=0:00:00.2, vio checks=0:00:00.0;mem=592.1M)

```

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: PLACEMENT (3)

Checking Timing results after placement:

Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate	
<hr/>							
WNS (ns):	1.766	2.577	6.224	1.766	N/A	5.945	
TNS (ns):	0.000	0.000	0.000	0.000	N/A	0.000	
Violating Paths:	0	0	0	0	N/A	0	
All Paths:	1652	1633	30	16	N/A	112	
<hr/>							
func_wccotyp	1.766	2.577	6.224	1.766	N/A	5.945	
	0.000	0.000	0.000	0.000	N/A	0.000	
	0	0	0	0	N/A	0	
	1652	1633	30	16	N/A	112	
<hr/>							

imec

©IMEC 2011 / CONFIDENTIAL

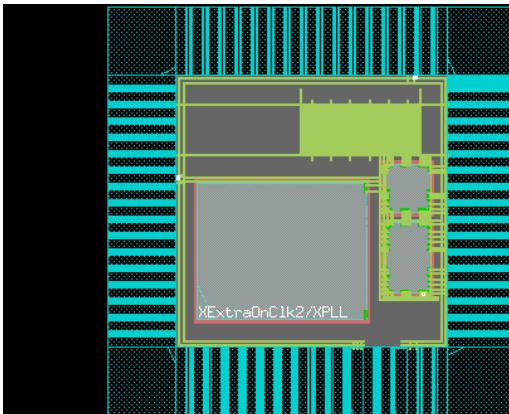
EARLY POWER ANALYSIS

Here a snapshot of the results of voltage storm early power analysis after floorplan and initial placement:

```
source
analyse_early_rail.tcl

source
view_analysis_results
```

```
grep -B1 "Overall data minimum"
layout/ENCOUNTER/RUNDIR/FE2VSEarlyRA/VDD_25C_avg_1/results
* Data filtering results for IR drop:
* Overall data minimum: 1.7983V <-----
```



Early Power Analysis help find un connected nets in the power plan

imec

©IMEC 2011 / CONFIDENTIAL

CTS - INTRODUCTION

- Start from a list of Clock definitions;
- Construct the Clock Trees of the design (clk tree configuration file)
- Describe the list of buffers to be used
- Uses the constraints file (.sdc) and the .lib files
- CTS synthesis can implement the clock tree with Gated Clocks
- CTS synthesis calculates and try to balance the (Rise and Fall) skew values while meeting the constraints

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: CTS AND OPTIMIZING CTS (I)

Here the script will:

- Run Clock Tree Synthesis with script CTS.tcl and command
`clockDesign -specFile NanoSOC.ctstch`
- Save Encounter Database after CTS
- Optimize timing results with `optDesign -postCTS`
- Save Encounter Database after optCTS

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: CTS AND OPTIMIZING CTS (2)

NanoSOC.ctstch file

```

AutoCTSRootPin XExtraOnClk2/XPLL/CKOUT          AutoCTSRootPin pad_clk           AutoCTSRootPin pad_clk2
Period     8ns                                     Period    10ns                      Period    50ns
MaxDelay   0.01ns # sdc driven default          MaxDelay  0.01ns # sdc driven default      MaxDelay  0.01ns # sdc driven default
MinDelay   0ns  # sdc driven default             MinDelay  0ns  # sdc driven default        MinDelay  0ns  # sdc driven default
MaxSkew    80ps # sdc driven default            MaxSkew   300ps # set_clock_uncertainty      MaxSkew   120ps # sdc driven default
SinkMaxTran 780ps # set_clock_transition       SinkMaxTran 780ps # set_clock_transition      SinkMaxTran 780ps # set_clock_transition
BufMaxTran 780ps # set_clock_transition         BufMaxTran 780ps # set_clock_transition      BufMaxTran 780ps # set_clock_transition
Buffer    BUFCKX1 BUFCKX2 BUFCKX3               Buffer    BUFCKX1 BUFCKX2 BUFCKX3           Buffer    BUFCKX1 BUFCKX2 BUFCKX3
BUFCKX4 BUFCKX6
NoGating   NO                                     NoGating  NO                         NoGating  NO
DetailReport YES                                    DetailReport YES                     DetailReport YES
#SetDPinAsSync NO                                #SetDPinAsSync NO                   #SetDPinAsSync NO
#SetIoPinAsSync NO                               #SetIoPinAsSync NO                  #SetIoPinAsSync NO
#SetASyncSRPinAsSync NO                          #SetASyncSRPinAsSync NO            #SetASyncSRPinAsSync NO
#SetTriStEnPinAsSync NO                         #SetTriStEnPinAsSync NO            #SetTriStEnPinAsSync NO
#SetBBoxPinAsSync NO                            #SetBBoxPinAsSync NO              #SetBBoxPinAsSync NO
RouteClkNet YES                                  RouteClkNet YES                    RouteClkNet YES
PostOpt    YES                                    PostOpt   YES                     PostOpt   YES
OptAddBuffer YES                                 OptAddBuffer YES                  OptAddBuffer YES
#RouteType specialRoute                         #RouteType specialRoute           #RouteType specialRoute
#LeafRouteType regularRoute                     #LeafRouteType regularRoute       #LeafRouteType regularRoute
END
END
END

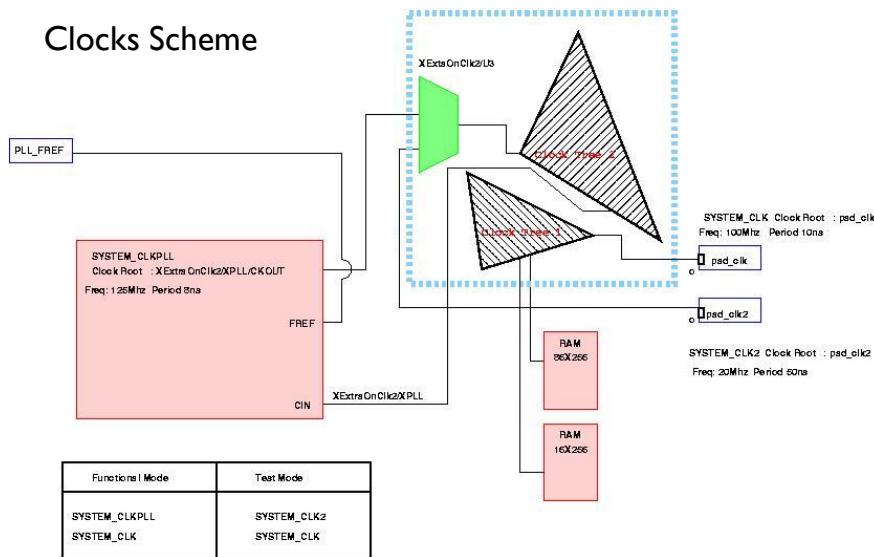
```

imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: CTS AND OPTIMIZING CTS (3)

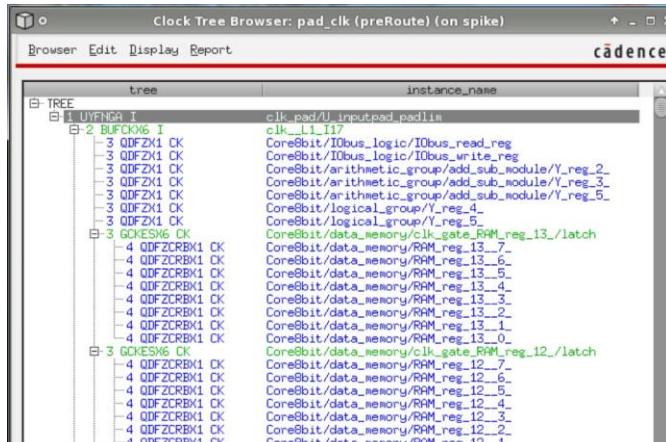
Clocks Scheme



imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: CTS AND OPTIMIZING CTS (4)



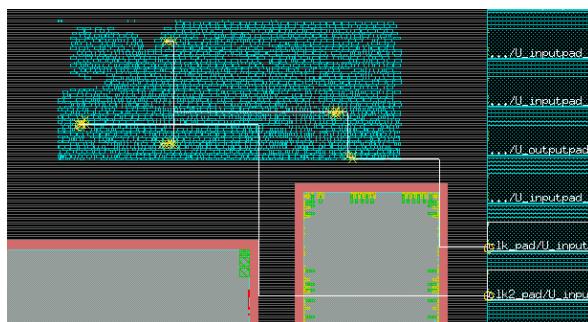
imec

© IMEC 2011 / CONFIDENTIAL

89

RUNSCRIPT: CTS AND OPTIMIZING CTS (5)

Selecting Clock nets in the GUI



Select first level of clock tree..

```
selectInst clk_pad/U_inputpad_padlim
selectObject Pin clk_pad/U_inputpad_padlim/O
selectInst clk2_pad/U_inputpad_padlim
selectObject Pin clk2_pad/U_inputpad_padlim/O
```

imec

© IMEC 2011 / CONFIDENTIAL

90

RUNSCRIPT: CTS BEFORE OPTIMIZATION (6)

reg2out setup violation

```
see ~infn/layout/ENCOUNTER/TIMING_REPORTS/NanoSOC_1_CTS/NanoSOC_1_postCTS.summary
+-----+-----+-----+-----+-----+-----+
| Setup mode | all | reg2reg | in2reg | reg2out | in2out | clkgate |
+-----+-----+-----+-----+-----+-----+
| WNS (ns):| -0.156 | 2.698 | 8.205 | -0.156 | N/A | 5.963 |
| TNS (ns):| -0.442 | 0.000 | 0.000 | -0.442 | N/A | 0.000 |
| Violating Paths:| 4 | 0 | 0 | 4 | N/A | 0 |
| All Paths:| 1652 | 1633 | 30 | 16 | N/A | 112 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| | Real | Total |
| DRVs +-----+-----+
| | Nr nets(terms) | Worst Vio | Nr nets(terms) |
+-----+-----+
| max_cap | 0 (0) | 0.000 | 0 (0) |
| max_tran | 0 (0) | 0.000 | 30 (30) |
| max_fanout | 0 (0) | 0 | 0 (0) |
+-----+-----+
```

imec

©IMEC 2011 / CONFIDENTIAL

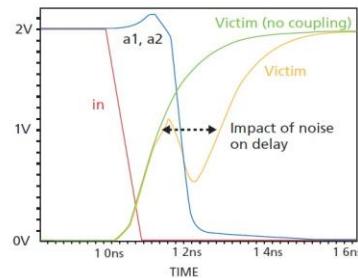
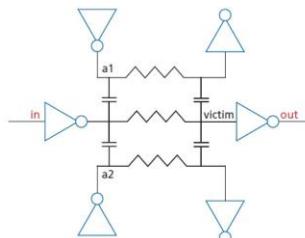
RUNSCRIPT: CTS AND OPTIMIZING CTS (7)

```
+-----+-----+-----+-----+-----+-----+
| Setup mode | all | reg2reg | in2reg | reg2out | in2out | clkgate |
+-----+-----+-----+-----+-----+-----+
| WNS (ns):| 0.123 | 2.697 | 7.383 | 0.123 | N/A | 5.961 |
| TNS (ns):| 0.000 | 0.000 | 0.000 | 0.000 | N/A | 0.000 |
| Violating Paths:| 0 | 0 | 0 | 0 | N/A | 0 |
| All Paths:| 1652 | 1633 | 30 | 16 | N/A | 112 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| | Real | Total |
| DRVs +-----+-----+
| | Nr nets(terms) | Worst Vio | Nr nets(terms) |
+-----+-----+
| max_cap | 0 (0) | 0.000 | 0 (0) |
| max_tran | 0 (0) | 0.000 | 30 (30) |
| max_fanout | 0 (0) | 0 | 0 (0) |
+-----+-----+
Density: 50.706%
Routing Overflow: 0.00% H and 0.04% V
-----+
| Hold mode | all | reg2reg | in2reg | reg2out | in2out | clkgate |
+-----+-----+-----+-----+-----+-----+
| WNS (ns):| 0.081 | 0.081 | 1.129 | 2.283 | N/A | 0.101 |
| TNS (ns):| 0.000 | 0.000 | 0.000 | 0.000 | N/A | 0.000 |
| Violating Paths:| 0 | 0 | 0 | 0 | N/A | 0 |
| All Paths:| 1652 | 1633 | 30 | 16 | N/A | 112 |
+-----+-----+-----+-----+
Density: 50.706%
Routing Overflow: 0.00% H and 0.04% V
```

imec

©IMEC 2011 / CONFIDENTIAL

CELTIC (SI)



imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: ROUTING (I)

Here the script will:

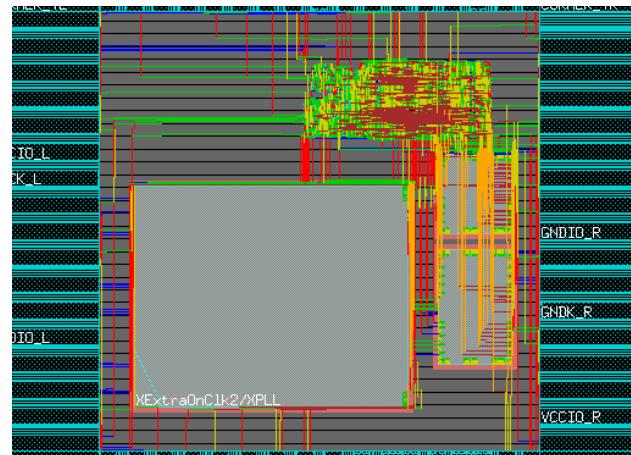
- Setup the router parameters with command `setNanoRouteMode`
- Setup the native extraction parameters with command `setExtractRCMode`
- Setup the SI (signal Integrity)parameters with command `setSiMode`
- Route the design with command `routeDesign -globalDetail`
- Optimize the vias with command `routeDesign -viaOpt`
- Analyze timing with command `timeDesign -postRoute` with and without taking into account SI phenomena (see `-si` option)
- Generate STA results with `-si` (always worst than without `-si`)
- Save Encounter Database after nanoRoute (nanoRoute.enc)

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: ROUTING (2)

Signal Nets after routing



imec

©IMEC 2011 / CONFIDENTIAL

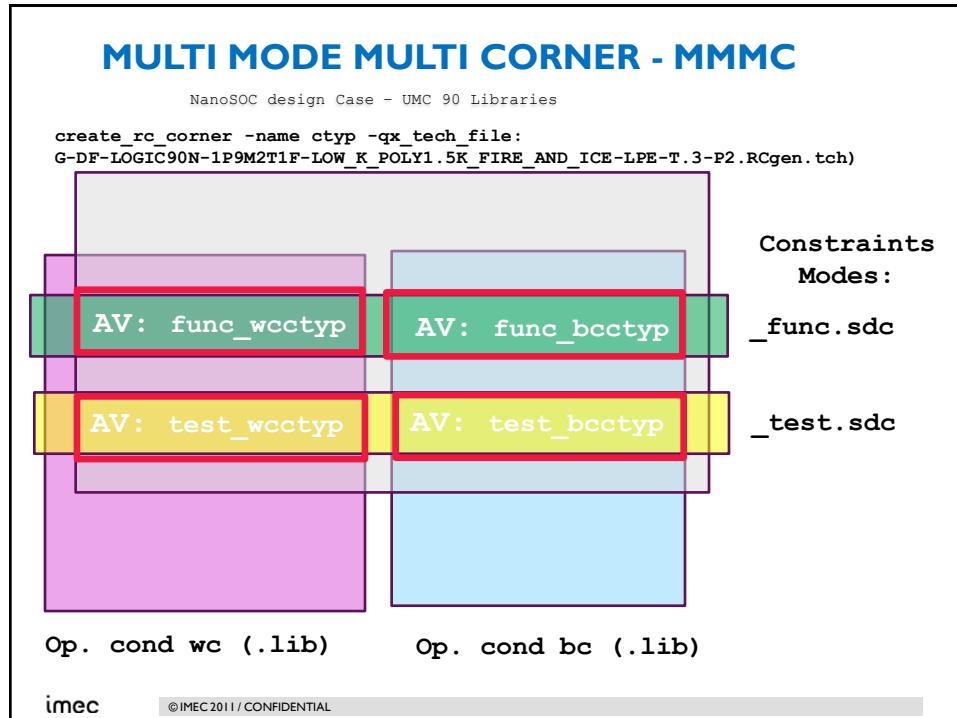
RUNSCRIPT: ROUTING (3)

Timing results after routing

	Setup mode	all	reg2reg	in2reg	reg2out	in2out	clkgate
	WNS (ns):	0.049	2.627	8.236	0.049	N/A	5.846
	TNS (ns):	0.000	0.000	0.000	0.000	N/A	0.000
	Violating Paths:	0	0	0	0	N/A	0
	All Paths:	1652	1633	30	16	N/A	112
<hr/>							
DRVs		Real		Total			
		Nr nets(terms)	Worst Vio	Nr nets(terms)			
	max_cap	0 (0)	0.000	0 (0)			
	max_tran	0 (0)	0.000	30 (30)			
	max_fanout	0 (0)	0	0 (0)			
<hr/>							
Hold mode	all	reg2reg	in2reg	reg2out	in2out	clkgate	
	WNS (ns):	0.085	0.085	1.360	2.279	N/A	0.104
	TNS (ns):	0.000	0.000	0.000	0.000	N/A	0.000
	Violating Paths:	0	0	0	0	N/A	0
	All Paths:	1652	1633	30	16	N/A	112

imec

©IMEC 2011 / CONFIDENTIAL



ANALYSIS VIEWS DEFINITION SCRIPT

socViewDefinition.tcl

```
#####
# Library set definition #####
# nvLib nom_vol nom_temp op_cond rcCorner UMC90 only 1 .ict& .tch#
#
#core_ff stdcell 1.100 -40 ff=BC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#core_ss stdcell 0.900 125 ss=WC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
#io_ff io 1.100 -40 ff=BC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# io_ss io 0.900 125 ss=WC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# RAM_8by256_BC RAM 1.100 -40.0 ff=BC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# RAM_8by256_WC RAM 0.900 125 ss=WC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# RAM_16by256_BC RAM 1.100 -40.0 ff=BC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# RAM_16by256_WC RAM 0.900 125 ss=WC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# PLL_BC.lib PLL 1.100 -40.0 ff=BC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#
# PLL_WC.lib PLL 0.900 125 ss=WC UMC_1P9M.tch UMC_1P9M.CapTbl #
#
#####
# end Library set definition #####

```

imec ©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: POST ROUTE CHECK&FIX STA VIOL. IN TEST MODE (1)

Here the script will:

- source **\$Script_dir/timingSettings_test.tcl** to set the Analysis view to test mode
- Analyze timing in test mode with command **timeDesign -postRoute** with and without taking into account SI phenomena (see **-si** option)
- Generate STA results with **-si** (always worst than without **-si**)
- Save Encounter Database after **nanoRoute** (**nanoRoute.enc**)
- After an optimization phase fix the hold violation with command **optDesign -postRoute -hold**

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: POST ROUTE CHECK&FIX STA VIOL. IN TEST MODE (2)

Results after running post Route Optimization and checking timing with SI in test mode:

```
See report in:  

/imec/other/prtool/infn/layout/ENCOUNTER/TIMING_REPORTS/NanoSOC_1_postRouteOptSI_hold
+-----+-----+-----+-----+-----+-----+
| Hold mode | all | reg2reg | in2reg | reg2out | in2out | clkgate |
+-----+-----+-----+-----+-----+-----+
| WNS (ns): | 0.056 | 0.056 | 5.411 | 24.284 | N/A | 0.104 |
| TNS (ns): | 0.000 | 0.000 | 0.000 | 0.000 | N/A | 0.000 |
| Violating Paths: | 0 | 0 | 0 | 0 | N/A | 0 |
| All Paths: | 3668 | 2536 | 1138 | 16 | N/A | 224 |
+-----+-----+-----+-----+-----+-----+
|test_bccTyp | 0.056 | 0.056 | 5.411 | 24.284 | N/A | 0.104 |
| | 0.000 | 0.000 | 0.000 | 0.000 | N/A | 0.000 |
| | 0 | 0 | 0 | 0 | N/A | 0 |
| | 3668 | 2536 | 1138 | 16 | N/A | 224 |
+-----+-----+-----+-----+-----+-----+
Density: 50.706%
Total number of glitch violations: 0
```

imec

©IMEC 2011 / CONFIDENTIAL

100

RUNSCRIPT: POST ROUTE CHECK&FIX STA VIOL. IN TEST MODE (3)

Results after analyzing time in test mode:

```
See report in:
~infn/layout/ENCOUNTER/TIMING_REPORTS/NanoSOC_1_Route_hold_post_route_test_mode
+-----+-----+-----+-----+-----+-----+
| Hold mode | all | reg2reg | in2reg | reg2ut | in2out | clkgate |
+-----+-----+-----+-----+-----+-----+
| WNS (ns): | -0.118 | -0.118 | 5.683 | 23.789 | N/A | 0.053 |
| TNS (ns): | -0.222 | -0.222 | 0.000 | 0.000 | N/A | 0.000 |
| Violating Paths: | 2 | 2 | 0 | 0 | N/A | 0 |
| All Paths: | 3668 | 2536 | 1138 | 16 | N/A | 224 |
+-----+-----+-----+-----+-----+
```

imec

©IMEC 2011 / CONFIDENTIAL

101

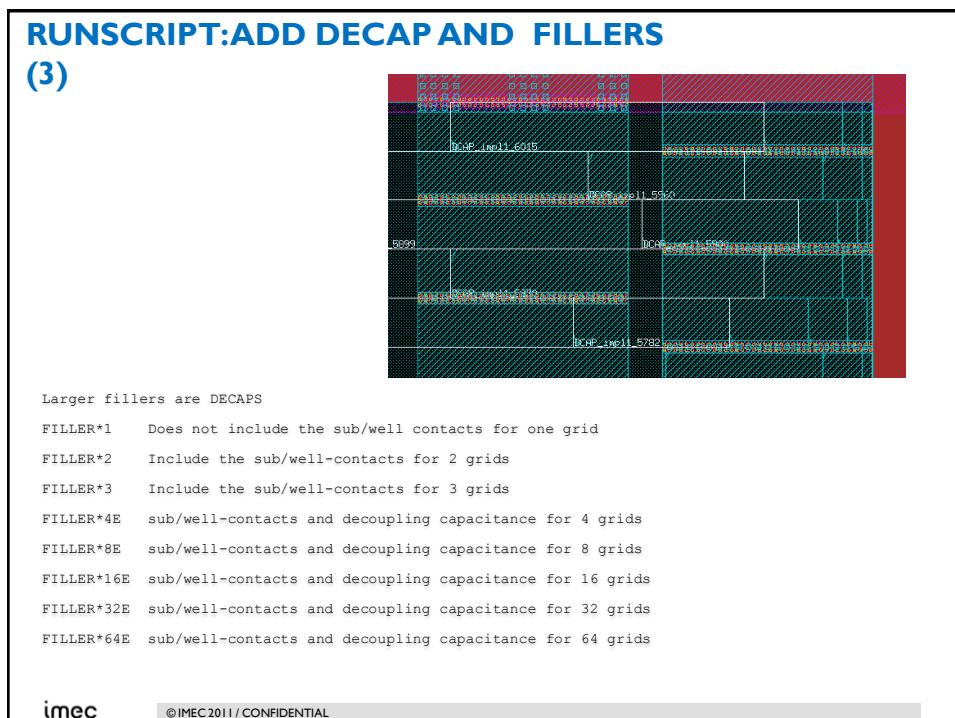
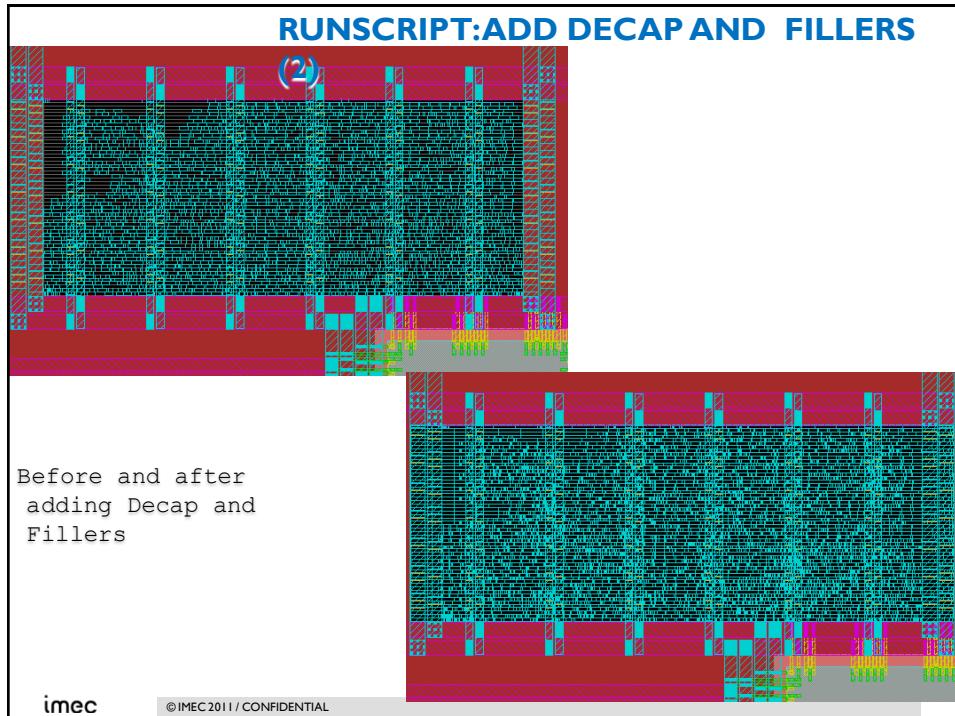
RUNSCRIPT:ADD DECAP AND FILLERS (I)

Here the script will:

- Add Decap and Fillers to standard cells area;
- **ecoRoute** the design to fix eventually created DRC violations due to the addition of the new cells
- Analyze time again using timeDesign -si
- Save database with command **saveDesign**.

imec

©IMEC 2011 / CONFIDENTIAL



RUNSCRIPT:TRIMMING AND FILLING NOTCHES(1)

Here the script will:

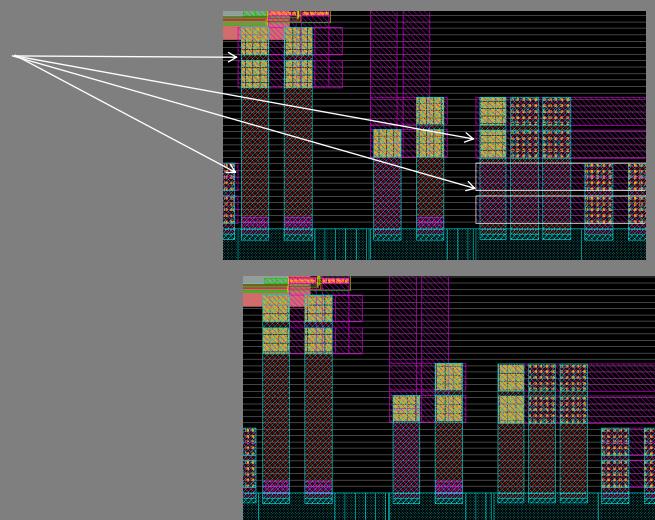
- Trim dangling pieces of metal in power nets with command **editTrim**;
- Fill eventual notches in the design with the command **fillNotch**

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT:TRIMMING AND FILLING NOTCHES(2)

Before and after **editTrim**:



imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: SAVING FINAL DESIGN (DEF) FOR FINAL SIGNOFF STA AND RUNNING FINAL NATIVE STA

Here the script will:

- Save the def file that will be used in Sign Off Extraction/SRA with Fire and Ice and ETS;

```
defOut -floorplan -placement -netlist -routing
${savedir}/${topCell}_signOff.def.gz
```

- Save the final database on which a final NATIVE STA will be performed with the script **nativeSignOffSTA.tcl** with **reportTiming** command and write out preliminary .sdf files.
- Save the .spf resultant from NATIVE extraction (within encounter) using the command **extractRC** and **rcOut**

All the results above are said to be NATIVE because the parasitic extraction is done within Encounter tool and hence with levels of accuracy that start very low in the beginning of the flow but increase as we approach to the final routed data base.

NEVERTHELESS these timing results are known to be less accurate than the ones obtained with the sign off "standalone" flow.

imec

©IMEC 2011 / CONFIDENTIAL

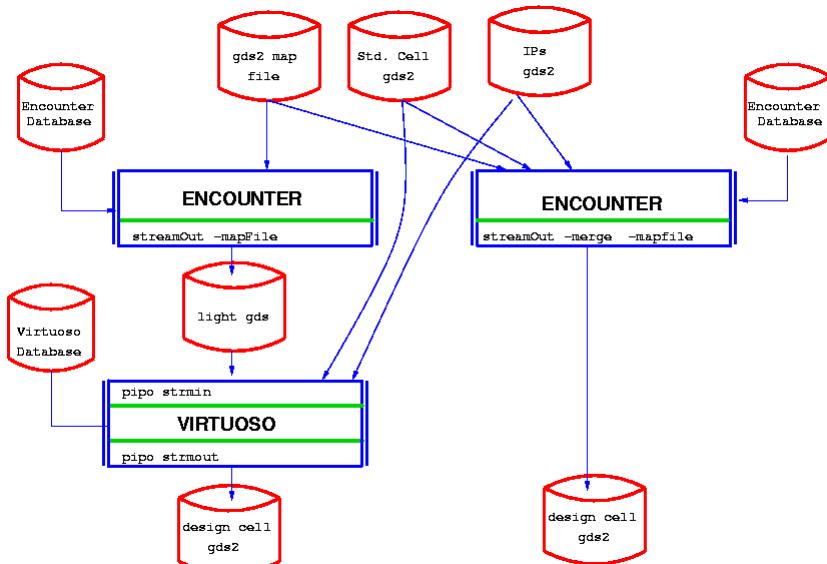
RUNSCRIPT: CHECKING THE RESULTS OF FINAL NATIVE STA

```
see results in/imec/other/prttool/infn/layout/ENCOUNTER/
see results in/imec/other/prttool/infn/layout/ENCOUNTER/
TIMING_REPORTS/AS_ETS/MMC_overview_final_on_16_Nov_2011.rpt
#####
# SETUP CHECKS #####
| Path | Pin | Cause | Slack |
| No. | | | |
+-----+-----+-----+
| 1 | pad_IObus_out[3] ^ | MET Late External Delay Assertion | 0.047 |
+-----+
#####
# RECOVERY CHECKS #####
| Path | Pin | Cause |
| No. | | |
+-----+
| 1 | XExtraOnClk2/Cntr_reg_19_/RB ^ | MET Recovery Check with Pin XExt |
| | | 19_/_CK |
+-----+
#####
# HOLD CHECKS #####
| Path | Pin | Cause |
| No. | | |
+-----+
| 1 | Inst_histo_builder/addr_reg8/clk_gate_q_reg/latch/ | MET Clock Ga |
| | | E ^ | ilder/addr_r |
+-----+
#####
# REMOVAL CHECKS #####
| Path | Pin | Cause |
| No. | | |
+-----+
| 1 | Inst_RST_synchronizer/s_reset_reg/RB ^ | MET Removal Check with P |
| | | _reset_reg/CK |
+-----+
```

imec

©IMEC 2011 / CONFIDENTIAL

STREAMING OUT THE DESIGN – INTRODUCTION



imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: STREAMING OUT THE DESIGN (I)

Here scripts will:

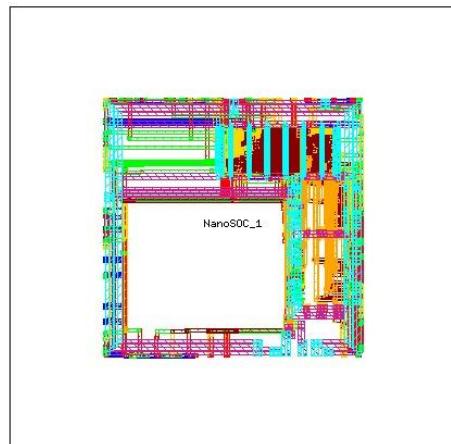
- Add the bond pads and labels with script **addBondpads_and_labels.tcl**;
- Stream out the gds2 of the design with the command **streamOut**;
- One can stream out the gds2 with the IP for which the gds2 is available by using the option **-merge** within the **streamOut** command (see script **streamOut_full.tcl**)

imec

©IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: STREAMING OUT THE DESIGN (2)

Here the results of **streamOut.tcl**:

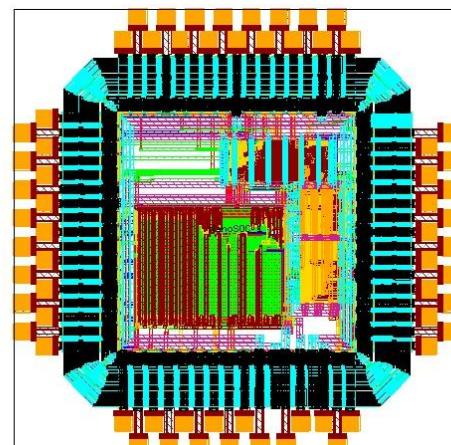


imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: STREAMING OUT THE DESIGN (3)

Here the results of **streamOut_full.tcl**:



imec

© IMEC 2011 / CONFIDENTIAL

RUNSCRIPT: EXPORTING FINAL DESIGN DATA BASE AND NETLISTS (FOR LVS AND CONFORMAL)

- Final def file (containing all gizmos including bond pads)

```
defOut -floorplan -placement -netlist -routing ${savedir}/${topCell}.def.gz
```

- Final Netlist (i.e. for formal proof, so without physical nor power cells)

```
saveNetlist ${savedir}/${topCell}.v.gz
```

- Final netlist with DECAP cells (active devices i.e. capacitors) and all other physical cells;

```
saveNetlist -phys ${savedir}/${topCell}_inclDCAP.v
```

- Final netlist with DECAP cells and no library cells for which spice netlists are available (option `-excludeLeafCell`). This netlist is to be used in v2lvs flow where a spice netlist is created from a verilog netlist)

```
saveNetlist -includePowerGround -excludeLeafCell \
-phys ${savedir}/${topCell}_inclDCAP_pg.v
```

- Save final Encounter database with bond pads

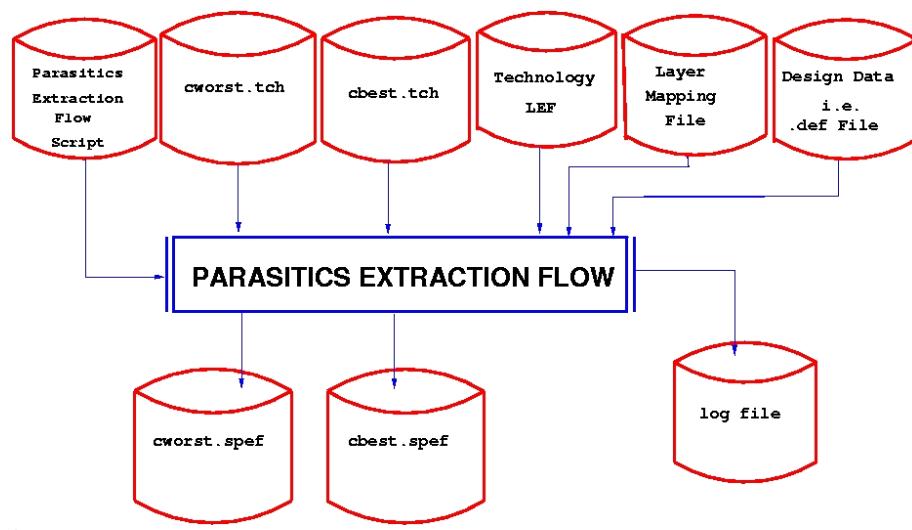
```
saveDesign ${savedir}/${topCell}.enc
```

imec

©IMEC 2011 / CONFIDENTIAL

SIGN OFF EXTRACTION WITH FIRE&ICE -INTRODUCTION

Generic Extraction Flow



imec

©IMEC 2011 / CONFIDENTIAL

SIGN OFF EXTRACTION WITH FIRE&ICE – STANDALONE FLOW

Input data:

1. .tch file : (The Library Vendor provided tch files are in general created using complex and expensive 3D field solver programs in order to allow RC extractors provide reasonably accurate RC data)
2. Technology Library lef
3. Lef files of all IP
4. Def file of the design;
5. Layer Mapping

Output data:

1. Spef files for sign off STA
2. Log files

imec

© IMEC 2011 / CONFIDENTIAL

SIGN OFF EXTRACTION WITH FIRE&ICE – STANDALONE FLOW

➤ Setup the tool:

```
source  
/imec/software/cadence/release2011/setup/ext_v10.11.149.csh c
```

➤ Run directory : ~infn/FIREANDICE

➤ Run command: source ~infn/FIREANDICE/SCRIPTS/run.cmd

➤ Log files:

```
~infn/FIREANDICE/FireAndIce_signOff/qrcExt_signOff_*.log
```

❑ Output SPEF files generated in:

```
~infn/FIREANDICE/FireAndIce_signOff/qrcExt_signOff_cworst.spef.gz  
~infn/FIREANDICE/FireAndIce_signOff/qrcExt_signOff_cbest.spef.gz
```

imec

© IMEC 2011 / CONFIDENTIAL

SIGN OFF STA WITH ETS - INTRODUCTION

STA THROUGHOUT THE FLOW

The Accuracy of the Analysis depends on the Quality of the delay calculations:

pre synthesis timing (not automated)
Architect/RTL designer hints

pre-layout timing

Wire-load Based

pre-route timing

Trial-Route Based

post-layout timing

Native Parasitic extraction based analysis
Sign-off Parasitic extraction based analysis

post-tape-out timing (silicon)

Run-time increases



SIGN OFF STA WITH ETS - STANDALONE FLOW

➤ Setup the tool:

```
source /imec/software/cadence/release2011/setup/ets_v10.12.000.csh c
```

➤ Run directory : ~infn/STA

➤ Run command: source ~infn/STA/SCRIPTS/run.cmd

➤ Checking timing results:

```
~cd ~infn/STA/signOff/rpts/
```

➤ Check reports in:

tim_func_bccbest/	tim_test_bccbest
tim_func_bccworst/	tim_test_bccworst
tim_func_wccbest/	tim_test_wccbest
tim_func_wccworst/	tim_test_wccworst

FINAL VERIFICATION

The final Verification Activities before tape out include:

- Physical and Electrical Verification
- Formal proof
- Simulation Data Delivered to Frontend

imec

© IMEC 2011 / CONFIDENTIAL

PHYSICAL AND ELECTRICAL VERIFICATION – DRC - INTRODUCTION

- DRC - Design Rules Checks are checks performed in the geometric description of the circuit;
- DRC goal is to verify if the fabrication rules are respected while the design is created;
- Typically DRC is performed on the full GDS2 file of the circuit. DRC can be also done in parts of the design (using LEF files of certain IP blocks), provided the supplier of the IP presents its DRC results;
- IMEC uses both methods but, for signoff, DRC of the full design in its GDS form is done. DEF based DRC is only used during chip development (i.e. inside Encounter)
- Preliminary DRC of IPs is recommended to be done.

imec

© IMEC 2011 / CONFIDENTIAL

PHYSICAL AND ELECTRICAL VERIFICATION – DRC (2)

➤ Setup the tool:

```
source /imec/software/mentor/release2010/setup/calibre_2010.3_37.26.csh
```

➤ Run directory : ~infn/layout/CHECKS/NanoSOC_1/DRC/

➤ Run command:

```
calibre -drc -hier -64 -turbo 4 G-DF-LOGIC_MIXED_MODE90N-  
1P9M2T1F-32.5KA-Calibre-drc-1.9-p2 | tee NanoSOC_1.log
```

➤ Run script to filter the reports:

```
/imec/other/invosoft/Software/dracula/COM/gen_report.perl  
NanoSOC_1.sum NanoSOC_1_DRC_RES.txt NanoSOC_1.rpt  
NanoSOC_1.log
```

➤ Uses calibredrv GUI to check the results:

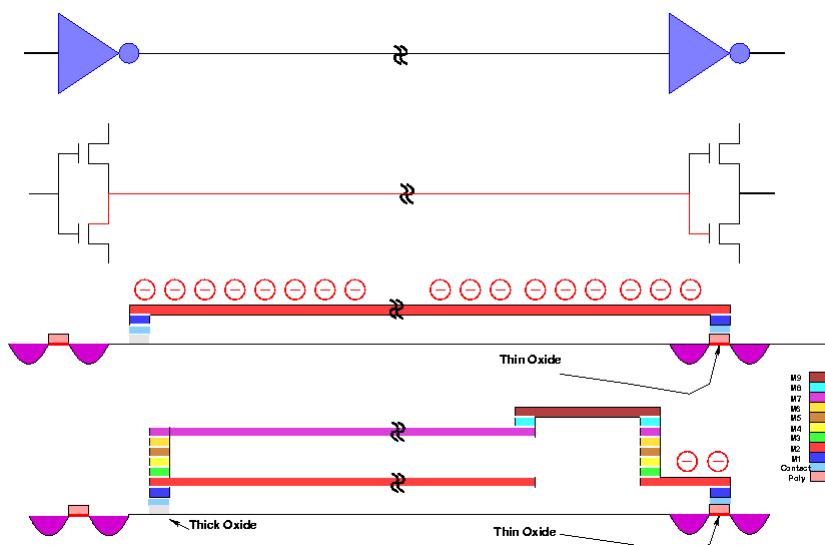
➤ Verification -> Start RVE -> here open file:

➤ NanoSOC_1_DRC_RES.txt

imec

©IMEC 2011 / CONFIDENTIAL

ANTENNAE - INTRODUCTION



imec

©IMEC 2011 / CONFIDENTIAL

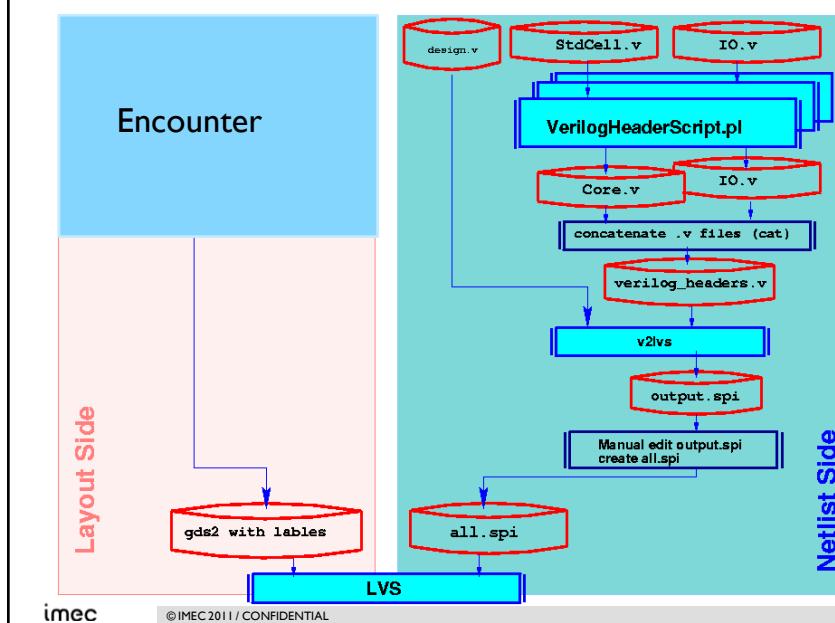
PHYSICAL AND ELECTRICAL VERIFICATION – LVS – INTRODUCTION (I)

- Key check in Physical Verification since it compares the design at transistor level;
- Compares a generated LAYOUT against its intended 'SCHEMATICS' or netlist description;
- Typically the layout is in the GDS2 format and the netlist is in the spice format (.cdl);
- LVS for real designs involves some data preparation steps. Moreover in the "schematics side" of the flow in order to create the spice netlist;
- In the "layout side", labels need to be attached to the GDS2 file in order to help the tool during the checks. .

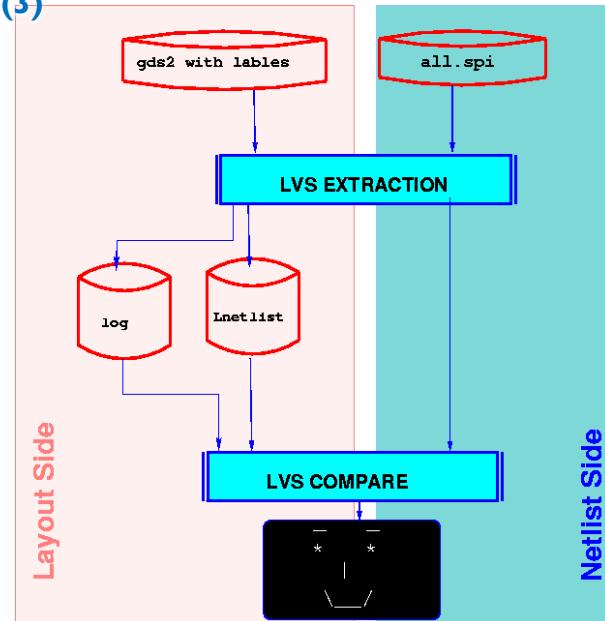
imec

© IMEC 2011 / CONFIDENTIAL

PHYSICAL AND ELECTRICAL VERIFICATION – LVS – INTRODUCTION (2)



PHYSICAL AND ELECTRICAL VERIFICATION – LVS – INTRODUCTION (3)



imec

©IMEC 2011 / CONFIDENTIAL

SIMULATION DATA DELIVERED BACK TO FRONTEND

Simulation Data delivered to front End consists on :

- Post layout netlist
- SDF files

▪ Preliminary SDF:

In the Development phase of the Backend work is not un common the frontend designers to request a preliminary sdc and post-layout netlist.

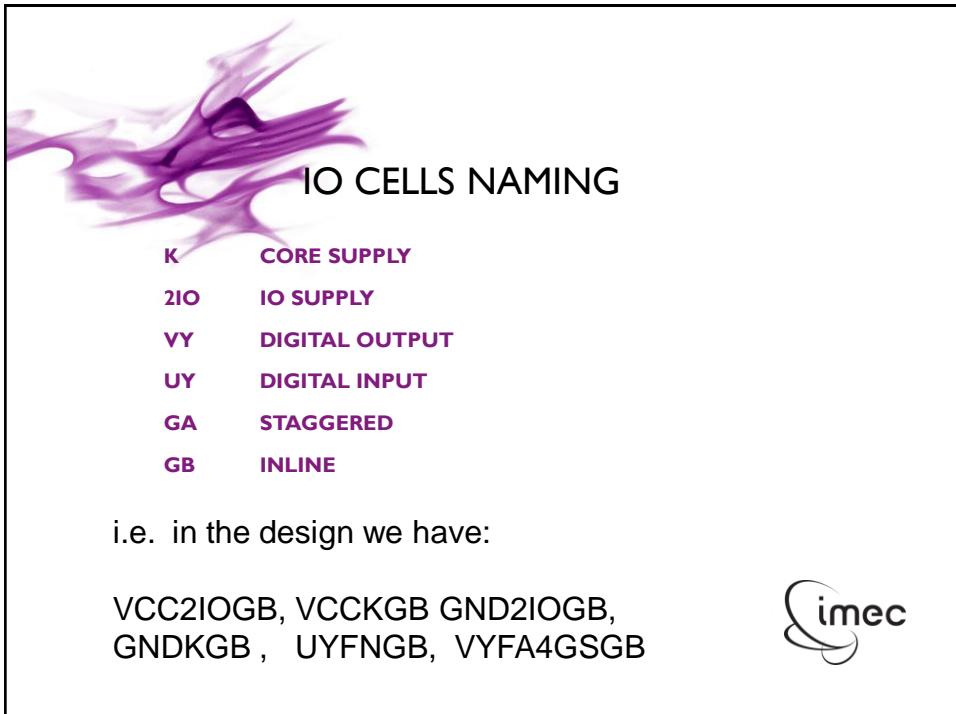
The STA native flow within encounter supports (with commands `extractRC` and `RCout`) described earlier, the generation of such a deliverables in early stage of the P&R flow.

✓ Sign Off SDF

The Sign Off STA flow generates the signoff SDF files .

imec

©IMEC 2011 / CONFIDENTIAL



IO CELLS NAMING

K	CORE SUPPLY
2IO	IO SUPPLY
VY	DIGITAL OUTPUT
UY	DIGITAL INPUT
GA	STAGGERED
GB	INLINE

i.e. in the design we have:

VCC2IOGB, VCCKGB GND2IOGB,
GNDKGB , UYFNGB, VYFA4GSGB



**ASPIRE
INVENT
ACHIEVE**

