Circuits Multi-Projets®

MPW Services Center for ICs, Photonics & MEMS Prototyping & Low Volume Production

mycmp.fr

Grenoble - France
From layout to chips

Design Kit Support Center Access

**https://cmp3.mycmp.fr/support/**

=> Sent an email to:
cmp-support@mycmp.fr
Typical support requests

- Downloading problems
- Installation problems
- Error messages when starting the CAD tool
- Difficulties to find specific information on the Design Kit documentation

https://cmp3.mycmp.fr/support/
2017 Support Statistics

- 193 tickets opened
- 181 tickets closed
- 3 pending dk provider support
- 7 in progress
- 2 waiting

Important to submit your technical support request through CMP help desk structure
## Available Tutorials

Tutorials are developed on a regular basis in order to help users at several stages of the design flow.

### Available Tutorials Table

<table>
<thead>
<tr>
<th>Schematics &amp; layouts</th>
<th>Digital design</th>
<th>Verifications</th>
<th>Electrical simulations</th>
<th>Parasitic extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import a GDS file from Encounter to Virtuoso</td>
<td>How to export a design at GDSII format ?</td>
<td>CDSOA layout magnification</td>
<td>RTL to GDS complete digital flow</td>
<td>Run LVS bbox or final LVS</td>
</tr>
<tr>
<td>CMOS028FD500</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CMOS065</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiCMOS9-MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiCMOS55</td>
<td>X (+ CMP web site : Services &gt; FAQ)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCMOS9GP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCMOS9A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSOI FEDO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Available Tutorials

- **CMOS028FD500**
- **CMOS065**
- **BiCMOS9-MW**
- **BiCMOS55**
- **HCMOS9GP**
- **HCMOS9A**
- **HSOI FEDO**
Diagram of the Digital Design-flow used in this tutorial
✓ A basic version of this tutorial was developed in 2015 (v1.4).
✓ A second version with several updates and additional functionalities was released on summer 2016 (v2.3).
✓ On autumn 2017 a third version has been released: it is closely related to the PDK and DP updates to version 2.9. Besides migration from 10 to 8 metal layers, the main changes with reference to the second version concern enhancements of several scripts and compatibility with the latest versions of the digital design flow.

Verilog RTL
```verilog
always @ (negedge clk) begin
  if (load == 1) eq <= 0;
  else if (eq1 == 0) eq <= 0;
  else eq <= 1;
end
```

GDSII layout
This plug and play tutorial presents a step by step introduction to digital design.

Each step from RTL to GDSII is detailed,

- based on standard methodologies and CAD tools,
- all scripts and testbenches are provided,
- illustrated from a simple digital circuit example,
- implemented on an advanced technology,
- integrates body biasing functionalities.
CAD tools used in this tutorial and their versions

- Cadence Incisive (NCVerilog, NCElab, NCSim), version 15.20.010 or later
- Synopsys Design Compiler, version K-2015.06-sp5 or later
- Cadence Genus, version 15.23.000 or later
- Cadence Innovus, version 15.23.000 or later
- Cadence Virtuoso, version 06.17.706 or later

Technology

- CMOS28FDSOI from STMicroelectronics
- PDK version 2.9.
Main updates between this tutorial release and the previous one (v. 2.3)

- upgrade of the whole tutorial to latest PDK version 2.9 and its corresponding design platform
- generation of SDC file during Design Compiler synthesis (it was only generated for Genus synthesis)
- update of the P&R script to be compatible with one of latest Cadence Innovus version: 16.12
- update of the P&R script to be compatible with 8ML stack metal PDK
- review of the Clock Tree Synthesis strategy.
Basic synchronous and sequential circuit, called “TOP_FIR”

Composed of 24 parallel Finite Impulse Response (FIR) filters

Final circuit:
- 73’000 standard cells
- 0.7µm² area

RTL simulation results:

<table>
<thead>
<tr>
<th>Reset</th>
<th>Coefficients loading</th>
<th>Outputs calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>reset</td>
<td>load</td>
</tr>
<tr>
<td>in[15:0]</td>
<td>0</td>
<td>6375</td>
</tr>
<tr>
<td>eq</td>
<td>in[15:0]</td>
<td>0</td>
</tr>
<tr>
<td>out[15:0]</td>
<td>x</td>
<td>0</td>
</tr>
</tbody>
</table>
Verilog RTL

generate
for (i=0; i<=N-1; i=i+1)
begin
    FIR
    FIR FIR1(
        .clk(clk),
        .reset(reset),
        .load(load),
        .input_sample(in),
        .output_sample(out1[i]) );
end
generate

Gate level netlist

FIR FIR_0_PIR1 (.clk (clk), .reset (reset), .load (load),
    .input_sample (in), .output_sample (out));
FIR_1 FIR_1_PIR1 (.clk (clk), .reset (reset), .load (load),
    .input_sample (in), .output_sample ([out1][15],
    [14], [out1][13], [out1][12], [out1][11],
    [10], [out1][9], [out1][8], [out1][7], [out1]
    [5], [out1][4], [out1][3], [out1][2]
    [1], [out1][0]));
FIR_2 FIR_2_PIR1 (.clk (clk), .reset (reset), .load (load),
    .input_sample (in), .output_sample ([out1][15],
    [14], [out1][13], [out1][12], [out1][11],
    [10], [out1][9], [out1][8], [out1][7], [out1]
    [5], [out1][4], [out1][3], [out1][2]
    [1], [out1][0]));

Gate level simulation results:

<table>
<thead>
<tr>
<th>time</th>
<th>clk</th>
<th>reset</th>
<th>load</th>
<th>in</th>
<th>out</th>
<th>eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6375</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

// Loading coefficients... // Coefficients loaded, start processing...
Gate level netlist  
+ top cell “PAD_TOP_FIR” (including IOs)

```verilog
module PAD_TOP_FIR ( clk, reset, load, in, out, eq );
    input [15:0] in;
    output [15:0] out;
    input clk, reset, load;
    output eq;
    wire w_clk, w_reset, w_load, w_eq;
    wire [15:0] w_out, w_in;
    wire [16:0] node_fir;
    wire netTis1, netTis0;
endmodule
```

**Back-annotated simulation results:**

```plaintext
Annotating SDF timing data:
Compiled SDF file:      PAD_TOP_FIR routed_genus.sdf
Log file:               sdffile_rc.log
Backannotation scope:   top.U
Configuration file:
Annotating completed successfully...

// Loading coefficients...

<table>
<thead>
<tr>
<th>time</th>
<th>clk</th>
<th>reset</th>
<th>load</th>
<th>in</th>
<th>out</th>
<th>eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6375</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[...]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

// Coefficients loaded, start processing...

<table>
<thead>
<tr>
<th>time</th>
<th>clk</th>
<th>reset</th>
<th>load</th>
<th>in</th>
<th>out</th>
<th>eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>192</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>180</td>
<td>180</td>
<td>0</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>33418</td>
<td>0</td>
</tr>
<tr>
<td>216</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>60376</td>
<td>0</td>
</tr>
<tr>
<td>228</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9375</td>
<td>55099</td>
<td>0</td>
</tr>
<tr>
<td>[...]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Verilog netlist**

```verilog
module PAD_TOP_FIR ( clk, reset, load, in, out, eq );
    input clk;
    input reset;
    input load;
    input [15:0] in;
    output [15:0] out;
    output eq;
endmodule
```

**Place & route tool**

Innovus (Cadence)

**Place & route script updated for Innovus 16.12.000**

**GDSII layout**
IOs placement:

- Addition of **2 specific pads** dedicated to supply VDDS and GNDS body biasing voltages (-1.8V to +1.8V):
- Addition of an **IO filler cell** to tie high or low the compensation signals:
- **2 supply pairs** VDDE and GNDE:
Body biasing in layout view:
- Powerful and flexible concept in FDSOI
- Reduces power consumption at a given perf. requirement
- Can be used to compensate Process, Temperature and Aging variations

Body biasing on LVT (flip-well) transistors:

FBB or RBB: speed or leakage optimization

FBB: Forward Body Biasing  RBB: Reverse Body Biasing
From layout to chips

28nm FDSOI features during P&R

Floorplan and power plan generation:

- 4 power rings (VDD, GND, VDDS, GNDS) drawn with thickest metal layers.
- 2 pairs of power stripes to feed the circuit with VDD and GND
- 1 pair of VDDS/GNDS stripes every 50µm spacing to feed body biasing voltages to standard cells
Core cells placement and routing:

- Fillers cells on **top and bottom core rows** (to meet particular DRC rules)
- Implementation of filler tap cells with **separated power and ground rails** (dedicated connections to bodies of standard cells)
- Restriction of the tool to use the 6 first metal layers to route signals, and the 2 top layers for power.
Final verifications

GDSII and netlist can be imported under Cadence Virtuoso 6.1.7:

- LVS and DRC verifications can be performed
118 Institutions received in 2016 version 2.6

Version 3.1 will be delivered in Q1 2018

LVS and DRC steps with Mentor/Calibre or Cadence/PVS will be added as an update to version 3.1 of this tutorial in Q1 2018

Positive feedback from designers!
Thank you!