Training Course of SOC Encounter

REF:
• CIC Training Manual – Cell-Based IC Physical Design and Verification with SOC Encounter, July, 2006

Speaker: C. –S. Hou
Outline

- Basic Concept of the Placement & Routing
- Auto Place and Route Using SOC Encounter
- Hard Block Abstraction Using Abstract Generator
- LAB
Basic Concept of the Placement & Routing
Cell-Based Design Flow

System Level
- MATLAB/ C/ C++/ System C/ ADS/ Covergen (MaxSim)
- Memory Generator

RTL Level
- Verilog/ VHDL
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- Conformal/ Formality
- Design/ Power Compiler
- DFT Compiler/ TetraMAX
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- SOC Encounter/ Astro
- GDS II
- DRC/ LVS (Calibre)
- PVS: Calibre xRC/ NanoSim (Time/ Power Mill)

Gate Level
- Physical Compiler/ Magma Blast Fusion

Logic Synthesis
- Verilog/ VHDL
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- Design/ Power Compiler
- DFT Compiler/ TetraMAX
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- SOC Encounter/ Astro
- GDS II
- DRC/ LVS (Calibre)
- PVS: Calibre xRC/ NanoSim (Time/ Power Mill)

Design for Test
- Verilog/ VHDL
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- Design/ Power Compiler
- DFT Compiler/ TetraMAX
- NC-Verilog/ ModelSim Debussy (Verdi)/ VCS
- SOC Encounter/ Astro
- GDS II
- DRC/ LVS (Calibre)
- PVS: Calibre xRC/ NanoSim (Time/ Power Mill)

Layout Level
- SOC Encounter/ Astro
- GDS II
- DRC/ LVS (Calibre)
- PVS: Calibre xRC/ NanoSim (Time/ Power Mill)

Post-Layout Verification
- SOC Encounter/ Astro
- GDS II
- DRC/ LVS (Calibre)
- PVS: Calibre xRC/ NanoSim (Time/ Power Mill)

Tape Out

Advanced Reliable Systems (ARES) Lab.
**SOC Encounter P&R Flow**

- **Netlist (Verilog)**
- **Timing Constraints (sdc)**
- **IO Constraints (ioc)**

1. Specify Floorplan
2. Timing Analysis
3. Pre-CTS Optimization
4. Power Planning
5. Power Analysis

1. IO, P/G Placement
2. Clock Tree Synthesis
3. Timing Analysis
4. Post-CTS Optimization
5. Power Route
6. SI Driven Route
7. Timing/SI Analysis

**GDS Netlist Spef DEF**

_Avanced Reliable Systems (ARES) Lab._
IO, P/G Placement

- Determine the positions of the PADs
  - Functional IO PAD
  - Power/Ground PAD
  - Corner PAD
- Just for the connection of PAD power rings

Advanced Reliable Systems (ARES) Lab.
Specify Floorplan

- Determine the aspect ratio of the Core and the gap between the PAD and Core
  - The Core Utilization is determined in this step
  - The final CHIP area is almost determined in this step

Advanced Reliable Systems (ARES) Lab.
Determine the related positions of Hard Blocks
- The performance is highly affected
Amoeba Placement

- Observe the result of cells and Hard Blocks placement
Power Planning

- Plan the power ring & power stripe
- IR-drop consideration
Power Analysis

- IR-drop & electron migration
Power Route

- Connect the power pins of standard cells to the global power lines

Advanced Reliable Systems (ARES) Lab.
Add IO Filler

- Fill the gap between PADs
  - Connect the PAD power rings
Routing

- Construct the final interconnections
Prepare Data

Library
- Physical Library (LEF)
  - Information of technology, standard cells, Hard Blocks, and APR
- Timing Library (LIB)
  - Timing information of the standard cells and Hard Blocks
- Capacitance Table
  - For more accurate RC analysis
- Celtic Library
  - For crosstalk analysis
- FireIce/Voltage Storm Library
  - For RC extraction and power analysis

User Data
- Gate-Level Netlist (Verilog)
- SDC Constraint (*.sdc)
- IO Constraint (*.ioc)
## LEF Format – Process Technology

<table>
<thead>
<tr>
<th>Layers</th>
<th>Design Rule</th>
<th>Parasitic</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLY</td>
<td>Net Width</td>
<td>Resistance</td>
</tr>
<tr>
<td>Contact</td>
<td>Net Spacing</td>
<td>Capacitance</td>
</tr>
<tr>
<td>Metal 1</td>
<td>Area</td>
<td></td>
</tr>
<tr>
<td>Via1</td>
<td>Enclosure</td>
<td></td>
</tr>
<tr>
<td>Metal 2</td>
<td>Wide Metal Slot</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Antenna</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Current Density</td>
<td></td>
</tr>
</tbody>
</table>

*Advanced Reliable Systems (ARES) Lab.*
Layer Metal1
  TYPE ROUTING;
  WIDTH 0.28;
  MAXWIDTH 8;
  AREA 0.202;
  SPACING 0.28;
  SPACING 0.6 RANGE 10.0 10000.0;
  PITCH 0.66;
  DIRECTION VERTICAL;
  THICKNESS 0.26;
  ANTENNACUMDIFFAREARATIO 5496;
  RESISTANCE RPERSQ 1.0e-01;
  CAPACITANCE CPERSQDIST 1.11e-04;
  EDGECAPACITANCE 9.1e-05;
END Metal1
LEF Format – APR Technology

☐ Unit
☐ Site
☐ Routing Pitch
☐ Default Direction
☐ Via Rule
The placement site gives the placement grid of a family of macros.
Row Based PR

Advanced Reliable Systems (ARES) Lab.
LEF Format – APR Technology: Routing Pitch, Default Direction

**Metal1 Routing Pitch**

**Metal2 Routing Pitch**

**Metal1 Routing Pitch**

**Via**

<table>
<thead>
<tr>
<th>Horizontal Routing</th>
<th>Vertical Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal1</td>
<td>Metal2</td>
</tr>
<tr>
<td>Metal3</td>
<td>Metal4</td>
</tr>
<tr>
<td>Metal5</td>
<td>Metal6</td>
</tr>
</tbody>
</table>
LEF Format – APR Technology: Via Generation

- To connect the wide metal, a via array is generated to reduce the via resistance
- Formulas for generating via arrays are defined

```
Layer Metal1
  Direction HORIZONTAL
  OVERHANG 0.2
Layer Metal2
  Direction VERTICAL
  OVERHANG 0.2
Layer Via1
  RECT -0.14 -0.14 0.14 0.14
  SPACING 0.56 BY 0.56
```

Advanced Reliable Systems (ARES) Lab.
LEF Format – APR Technology: Same Net Spacing

SPACING
SAMENET Metal1 Metal1 0.23;
SAMENET Metal2 Metal2 0.28 STACK;
SAMENET Metal3 Metal3 0.28;
SAMENET VIA12 VIA12 0.26;
SAMENET VIA23 VIA23 0.26;
SAMENET VIA12 VIA23 0.0 STACK;
END SPACING

VIA12 and VIA23 allow stack

0.23

Same Net Spacing Rule

Advanced Reliable Systems (ARES) Lab.
LEF Format – APR Technology: Physical Macros

- Define physical data for
  - Standard cells
  - I/O pads
  - Memories
  - Other hard macros

- Describe abstract shape
  - Size
  - Class
  - Pins
  - Obstructions
MACRO ADD1
CLASS CORE;
FOREIGN ADD1 0.0 0.0;
ORIGEN 0.0 0.0;
LEQ ADD;
SIZE 19.8 BY 6.4;
SYMMETRY x y;
SITE coresite;
PIN A
  DIRECTION INPUT;
  PORT
  LAYER Metal1;
  RECT 19.2 8.2 19.5 10.3;
  ....
END
END A
....
END ADD1
LIB Format

- Operating condition
  - Slow, fast, typical

- Pin type
  - Input/output/inout
  - Function
  - Data/clock
  - Capacitance

- Path delay

- Timing constraint
  - Setup, hold, mpwh, mpwl, recovery
Gate-Level Netlist

- If designing a chip, IO PADs, power PADs, and Corner PADs should be added before the netlist is imported.
- Make sure that there is no “assign” statement and no “*cell*” cell name in the netlist.
SDC Constraint

- Clock constraints
- Input delay/ Input drive
- Output delay/ Output load
- False path
- Multi-cycle path
**IO Constraint**

- **Version:** 1
- **Pad:** CORNER0 NW PCORNERDGZ
- **Pad:** PAD_CLK N
- **Pad:** PAD_HALT N
- **Pad:** CORNER1 NE PCORNERDGZ
- **Pad:** PAD_X1 W
- **Pad:** PAD_X2 W
- **Pad:** CORNER2 SW PCORNERDGZ
- **Pad:** PAD_IOVDD1 S PVDD2DGZ
- **Pad:** PAD_IOVSS1 S PVSS2DGZ
- **Pad:** CORNER3 SE PCORNERDGZ
- **Pad:** PAD_VDD1 E PVDD1DGZ
- **Pad:** PAD_VSS1 E PVSS1DGZ

(*.ioc File)

*Advanced Reliable Systems (ARES) Lab.*
How To Decide the NO. of Power/Ground PADs

- The following factors are considered:
  - SSO: Simultaneously Switch Outputs
  - SSN: The noise produced by SSO buffers
  - DI: Maximum NO. of copies for one specific kind of IO PAD switching from high to low simultaneously without making ground voltage level higher than 0.8 volt for one ground PAD
  - DF: Driving Factor, \( DF = \frac{1}{DI} \)
  - SDF: Sum of Driving Factor

- Suggestion in SSO case:
  - Required NO. of ground PADs = SDF
  - Required NO. of power PADs = SDF/1.1

Advanced Reliable Systems (ARES) Lab.
If a design has 20 PDB02DGZ (2mA) and 10 PDD16DGZ (16mA). Then,

SDF = 20 \times 0.02 + 10 \times 0.3 = 3.4

In SSO case,
- NO. of VSS PAD = 3.4 \rightarrow 4
- NO. of VDD PAD = 3.4/1.1 = 3.09 \rightarrow 4
Tips to Reduce the Power/Ground Bounce

- Don’t use stronger output buffers than what is necessary
- Use slew-rate controlled outputs
- Place power pad near the middle of the output buffer
- Place noise sensitive I/O pads away from SSO I/Os
- Place VDD and VSS pads next to clock input buffer
Auto Place and Route Using SOC Encounter
CHIP-Level Netlist

If your gate-level netlist is generated by "CORE-level synthesis", you should all the "CHIP-level module" in it.
CHIP-Level Netlist (Cont’)

- If your design has a “Hard Block”, you should add an “empty module” for it
  - The module name should be the same as the “cell name” of the Hard Block

Ex:

```verilog
module memory_0407 (O, clock, cen_in, oen_in, wen_in, A, D);
  input   clock;
  input [7:0] A;
  input [7:0] D;
  input   cen_in;
  input   oen_in;
  input   wen_in;
  output [7:0] O;
endmodule
```

(Module Declaration)

```verilog
wire [7:0] D;
wire [7:0] DI_T;
wire [7:0] A;
wire [7:0] ADDR_T;
wire [7:0] Q2;
wire [7:0] Q1;
RA1SHD256x8 RA1SHD256x8 (.Q(Q1), .CLK(clk), .CEN(CEN1), .CEN(n188), .WEN( WEN), .A(A), .D(D));
memory_0407 memory_0407(.Q(Q2), .clock(clk), .cen_in(CEN2), .oen_in(n188),
wen_in(WEN), .A(A), .D(D));
```

(Module Reference)

Connected Wire Name in Verilog

Pin Name in SPICE

Advanced Reliable Systems (ARES) Lab.
CHIP-Level Timing Constraint

Ex:

```plaintext
# Created by Design Compiler write_sdc on Sun Jul 29 06:04:11 2007

******************************************************************************
set_sdc_version 1.4

create_clock -period 0.34 -waveform {0 4.5} [get_ports {clk}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[0]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[1]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[2]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[3]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[4]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[5]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[6]}]
set_input_delay 0.34 -clock "clk" [get_ports {ADDR_S[7]}]
set_input_delay 0.34 -clock "clk" [get_ports {DI_S[0]}]
set_input_delay 0.34 -clock "clk" [get_ports {DI_S[1]}]
set_input_delay 0.34 -clock "clk" [get_ports {DI_S[2]}]
set_input_delay 0.34 -clock "clk" [get_ports {DI_S[3]}]
```

CHIP-Level Clock Declaration

Set False Path to Your Test Pins

Set Parameters to the PAD IO

---

Advanced Reliable Systems (ARES) Lab.
Getting Started

- *linux %> ssh -l “user name” cae18.ee.ncu.edu.tw*  
  - Connect to Unix

- *unix %> source /APP/cad/cadence/SOC/CIC/soc.csh*

- *unix %> encounter*

(Do not run in the background mode !!)

*Advanced Reliable Systems (ARES) Lab.*
Import Design <Design>

- **Design/Design Import**
- Verilog Files: your gate-level netlist
- Tot Cell
- LEF Files (*.lef): including all the LEF files of cell libraries & hard blocks
- LIB Files (*.lib):
  - Max Timing Libraries
  - Min Timing Libraries
  - Common Model Libraries
- IO Assignment File: *ioc

Advanced Reliable Systems (ARES) Lab.
Import Design <Timing>

- Capacitance Table File
- Timing Constraint File: *.sdc
Import Design <Power> <IPO/CTS>

- Power Nets
- Ground Nets
- Footprints for In-Place Layout Optimization (IPO) and Clock Tree Synthesis (CTS)
Import Design <Misc.>

- QX Tech File
- QX Library Directory

(Floorplan View)
Global Net Connection

- Floorplan/Global Net Connections

![Global Net Connection Window]

Advanced Reliable Systems (ARES) Lab.
Specify Floorplan

Floorplan/Specify Floorplan

Advanced Reliable Systems (ARES) Lab.
Specify Scan Chain

- `encounter %> specifyScanChain ScanChainName`
  - `start {ftname | instPinName}`
  - `start {ftname | instPinName}`

- `encounter %> scantrace`

Ex:

```
encounter 2> specifyScanChain scan1 -start PAU_si/C -stop PAU_so/I
encounter 3> specifyScanChain scan2 -start PAD_test_si2/C -stop PAD_test_so2/I
encounter 4> scantrace
```

```
Tracing scan chain: scan1
Successfully traced scan group scan1 (99 elements: 97 scan bits).
Tracing scan chain: scan2
Successfully traced scan group scan2 (57 elements: 56 scan bits).
*** Scan Trace Summary:
Successfully traced scan group scan1 (99 elements: 97 scan bits).
Successfully traced scan group scan2 (57 elements: 56 scan bits).
Successfully traced 2 scan groups (total 156 elements: 153 scan bits).
INFO: Performed sanity check on scan group scan1 (+1 scan edge marked as fixed).
INFO: Passed sanity check on scan group scan2.
*** Scan Sanity Check Summary:
*** 1 scan group passed sanity check.
*** 1 scan group corrected sanity check (total +1 fixed scan edge).
```

Advanced Reliable Systems (ARES) Lab.
Hard Block Placement

- Move/Resize/Reshape floorplan object
Edit Block Halo

- Floorplan/Edit Block Halo
- Reserve space without standard cell placement
Standard Cell Placement

Place/Place

Advanced Reliable Systems (ARES) Lab.
Power Planning – Add Rings

- Floorplan/Custom Power Planning/Add Rings

Advanced Reliable Systems (ARES) Lab.
Power Planning – Add Block Rings

- Floorplan/Custom Power Planning/Add Rings

Advanced Reliable Systems (ARES) Lab.
Example for Power Rings

Advanced Reliable Systems (ARES) Lab.
PAD Pins

Route/SRoute

<table>
<thead>
<tr>
<th>Route</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Block pins</td>
<td>✅</td>
</tr>
<tr>
<td>Pad pins</td>
<td>✅</td>
</tr>
<tr>
<td>Pad rings</td>
<td>✅</td>
</tr>
<tr>
<td>Standard cell pins</td>
<td>✅</td>
</tr>
<tr>
<td>Stripes (unconnected)</td>
<td></td>
</tr>
</tbody>
</table>

Advanced Reliable Systems (ARES) Lab.
Power Planning – Add Stripes

Floorplan/Custom Power Planning/Add Stripes

Advanced Reliable Systems (ARES) Lab.
Ex:

Advanced Reliable Systems (ARES) Lab.
Fix Un-Connected Stripes

**Route/SRoute**

<table>
<thead>
<tr>
<th>Route</th>
<th>Block pins</th>
<th>Pad pins</th>
<th>Pad rings</th>
<th>Standard cell pins</th>
<th>Stripes (unconnected)</th>
</tr>
</thead>
</table>

![Diagram of un-connected stripes before and after fixing](image)

*Advanced Reliable Systems (ARES) Lab.*
Flow Clock Tree Synthesize

Create Clock Tree Spec

Specify Clock Tree

Modify

Synthesis Clock Tree

Display Clock Tree

Netlist
Synthesis report
Clock nets
Routing guide

clock spec
Create/Specify/Synthesis Clock Tree Spec.

- **Clock/Create Clock Tree Spec**
  - Create Clock Tree Specification
  - Buffer Footprint: clkbuf
  - Inverter Footprint: clkin
  - Buffer List:
    - Ignore Don’t Use
  - Save Spec To: CHIP.ctstch

- **Clock/Specify Clock Tree**

- **Clock/Synthesis Clock Tree**
  (Clock Spec.)

Advanced Reliable Systems (ARES) Lab.
Example for CTS Report

<table>
<thead>
<tr>
<th>Nr. of Subtrees</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr. of Sinks</td>
<td>99</td>
</tr>
<tr>
<td>Nr. of Buffer</td>
<td>5</td>
</tr>
<tr>
<td>Nr. of Level (including gates)</td>
<td>1</td>
</tr>
</tbody>
</table>

Max trig. edge delay at sink(L): bist_group/bist/m0/shift_reg_reg_10_CBCN 230.9(ps)
Min trig. edge delay at sink(R): bist_group/memory_0407/clock 176.8(ps)

<table>
<thead>
<tr>
<th>(Actual)</th>
<th>(Required)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rise Phase Delay</td>
<td>176.8~177.5(ps)</td>
</tr>
<tr>
<td>Fall Phase Delay</td>
<td>169.2~230.9(ps)</td>
</tr>
<tr>
<td>Trig. Edge Skew</td>
<td>54.1(ps)</td>
</tr>
<tr>
<td>Rise Skew</td>
<td>40.7(ps)</td>
</tr>
<tr>
<td>Fall Skew</td>
<td>41.7(ps)</td>
</tr>
<tr>
<td>Max. Rise Buffer Tran</td>
<td>50.4(ps)</td>
</tr>
<tr>
<td>Max. Fall Buffer Tran</td>
<td>58.4(ps)</td>
</tr>
<tr>
<td>Max. Rise Sink Tran</td>
<td>177.9(ps)</td>
</tr>
<tr>
<td>Max. Fall Sink Tran</td>
<td>165.4(ps)</td>
</tr>
<tr>
<td>Min. Rise Buffer Tran</td>
<td>23.2(ps)</td>
</tr>
<tr>
<td>Min. Fall Buffer Tran</td>
<td>23.2(ps)</td>
</tr>
<tr>
<td>Min. Rise Sink Tran</td>
<td>97.9(ps)</td>
</tr>
<tr>
<td>Min. Fall Sink Tran</td>
<td>92.4(ps)</td>
</tr>
</tbody>
</table>

****** NO Max Transition Time Violation ******

****** NO Min Transition Time Violation ******

****** NO Max Fanout Violation ******

Advanced Reliable Systems (ARES) Lab.
Display Clock Tree

- Clock/Display/Display Clock Tree

Ex:

Advanced Reliable Systems (ARES) Lab.
Power Analysis

- **Power/Edit Pad Location**
  - ![Image of Power/Edit Pad Location](image1)

- **Power/Edit Net Toggle Probability**
  - ![Image of Power/Edit Net Toggle Probability](image2)

- **Power/Power Analysis/Statistical**
  - ![Image of Power Analysis/Statistical](image3)

**Ex:**

```
# The Power Analysis Report for VDD net
average power(default): 1.0470e+01 mw
average switching power(default): 2.3212e+00 mw
average internal power(default): 8.1442e+00 mw
average leakage power(default): 4.7800e-03 mw
average user specified power(default): 0.0000e+00 mw

average power by clock domain category:
clock domain1: 1.028e+01 mw
  clock tree power : 4.1149e+00 mw
  non clock tree power : 6.167e+00 mw
unclock domain(0.2): 1.8807e-01 mw

average power by cell category:
core: 7.3761e+00 mw
block: 3.1034e+00 mw
10: 9.7020e-00 mw

average power(considered in rail analysis): 1.0470e+01 mw

worst Ilin drop average analysis: 1.9771e-05 v

number of nodes in rail network: 10645 nodes

worst EM:
  "M1" 5.0000e-02 ma/u
  "M2" 0.0000e+00 ma/u
  "M3" 3.8142e-01 ma/u
  "M4" 3.8142e-01 ma/u
  "M5" 2.647e-01 ma/u
  "V12" 1.4521e-02 ma/cut
  "V23" 1.4521e-02 ma/cut
  "V34" 9.7855e-02 ma/cut
  "V45" 2.8260e-02 ma/cut

biggest toggled net: CLK__L1__NO
do. of terminal: 93

total cap: 7.6044e+02 fF
```

Advanced Reliable Systems (ARES) Lab.
Example for Rail Analysis of IR-Drop & EM

(IR-Drop)  (EM)

Advanced Reliable Systems (ARES) Lab.
Power Route

Route/SRoute

- Block pins
- Pad pins
- Pad rings
- Standard cell pins
- Stripes (unconnected)

Advanced Reliable Systems (ARES) Lab.
IO Filler

- encounter %> source addIoFiller.cmd
Nano Route

Route/NanoRoute

Advanced Reliable Systems (ARES) Lab.
Example for Nano Route
Cell Filler

Place/Filler/Add Filler

Ex:

Advanced Reliable Systems (ARES) Lab.
Save Design

- **Design/Save/Netlist** → *.v
- **Timing/Calculate Delay** → *.sdf
- **Design/Save/DEF** → *.def
  - SELECT “Save Scan”
Bounding PAD

- unix %> chmod 755 addbonding.pl
- unix %> /usr/bin/perl addbonding.pl CHIP.def
- encounter %> source bondPads.cmd

Ex:
Save GDSII

- Design/Save/GDS → *.gds