

# <SOC Encounter> LAB

Aug. 2008 TW Nov. 2010 CS Dec. 2011 CS Jul. 2012 CS & KT



Cell-Based Training LAB

	LAB File List for Front-End De	sign
Folder	Name	Description
MEM	SRAM_SP_ADV.ps	Post script file
	SRAM_SP_ADV.spec	<b>RAM specification</b>
	SRAM_SP_ADV.v	Verilog model
	SRAM_SP_ADV.vclef	Physical geometry
		file
	SRAM_SP_ADV_ff_1.140.0_syn.lib	Timing files for
	SRAM_SP_ADV_ff_1.10_syn.lib	SOC encounter
	SRAM_SP_ADV_ss_0.9_125.0_syn.lib	
	SRAM_SP_ADV_tt_1.0_25.0_syn.lib	
	SRAM_SP_ADV_ff_1.140.0_syn.db	Timing files for
	SRAM_SP_ADV_ff_1.10_syn.db	Design Compiler
	SRAM_SP_ADV_ss_0.9_125.0_syn.db	
	SRAM_SP_ADV_tt_1.0_25.0_syn.db	
RTL	core.v top_lv_bisr.v lv*.v	Verilog netlist
		Test bench
		Signal
		configuration file
		Waveform
		aliasing file
	SRAM_SP_ADV.v	Verilog model
GTL	run.script	Synthesis script
		file
	core.v top_lv_bisr.v lv*.v	Verilog netlist
	.synopsys_dc.setup	Design compiler
		setup file
		Test bench
	core.sdc	Timing constraint
		file
	core.sdf	Gate-level timing
		file
	core.spf	Scan chain
		configuration



	core.vg	Gate-level netlist
GTL/	tsmc090.v	Verilog model of
SIMULATIO		standard cells
Ν	core.vg	Gate-level netlist
	core.sdf	Gate-level timing
		file
	SRAM_SP_ADV.v	Verilog model
		Test bench

	LAB File List for Back-E	nd Design
Folder	Name	Description
SOC/lef	antenna.lef	LEF file for antenna rules
	tpzn90gv3_91m.lef	LEF file of IO PAD
	tsmc090lk_9lm_2thick_tech.lef	LEF file of standard cells
	tsmc090nvt_macros.lef	LEF file for antenna rules
SOC/lib	fast.lib	Fast timing library for
		standard cells
	slow.lib	Slow timing library for
		standard cells
	typical.lib	Common timing library
		for standard cells
	tpzn90gv3bc.lib	Fast timing library for IO
		PAD
	tpzn90gv3tc.lib	Common timing library
		for IO PAD
	tpzn90gv3wc.lib	Slow timing library for
		IO PAD
SOC/	••••	QX libraries
library.cl		
SOC	addIoFiller.cmd	Add IO filler script
	addbonding_v3.6.pl	Add bonding PAD script
	icecaps.tch	FireIce capacitance file
	t90g_rcb.CapTbl	Capacitance table
	t90g_rct.CapTbl	
	t90g_rcw.CapTbl	



Advanced REI	able System Lab	Cell-Based Training LAB
	io.list	IOPAD list file
ADD_LVS	0.18um_Virtuoso4.4.tf	Technology file
	tsmc18_core.gds	Layout of standard cells
	tsmc18_io_final.gds3	Layout of IO PADs
	t18abs2lay.ile	•
	display.drf	Display file for TSMC 18
	T18drc_13a25a.drc	DRC command file
DRC	T18drc_13a25a.drc	DRC command file
LVS	Caliber-lvs-cur_soce	LVS command file
	tsmc18_lvs.spi	SPICE model for
		standard cells
	tsmc18_lvs.v	Verilog model for
		standard cells
POSIM	CHIP.cfg	Nanosim configuration
	CHIP.vec	Po-sim Test pattern
PATTERN	-	-
_GEN		

1			encounter		
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>P</u> laces	<u>H</u> elp		
	GTL		LEF	LIB	
	B		P		
	MEM		RTL	SOC	
🎾 er	ncounter 🔻 🤇	5 items, F	Free space: 4.5 GB		

Fig. LAB Folders

- Preparations before APR:
  - Goto [SOC]



- Copy <core.vg> and <core.sdc> from [GTL] to [SOC]
- Goto [SOC/lef]
- Copy <SRAM\_SP\_ADV.vclef> from [MEM] to [SOC/LEF]
- Goto [SOC/LIB]
- Copy <\*.lib> from [MEM] to [SOC/LIB]
- Edit the file  $\langle \text{CHIP.vg} \rangle \rightarrow$  chip-level netlist
- Edit the file  $\langle \text{CHIP.sdc} \rangle \rightarrow$  chip-level timing constraint file
- Edit the file  $\langle \text{CHIP.ioc} \rangle \rightarrow \text{IO pad assignment}$
- Open <core.vg>, then save as <CHIP.vg>. Then, add the following content in the last of file:

TSMC 90: IO pad module names are as follow input IO: PDIDGZ\_33 output IO: PDO24CDG\_33

#### ⊢ 是英文

For example:

PDIDGZ\_33 PAD\_clk (.PAD(I\_clk), .C(clk)); //"I\_clk" is from external signal to IO PAD //"clk" is from IO PAD to internal signal

PDO24CDG\_33 PAD\_out (.PAD(I\_out), .I(out));

//"I\_out" is from IO PAD to external signal

//"out" is from internal signal to IO PAD

module CPU\_CHIP( I\_clk, I\_rst, I\_control, I\_memaddr, I\_in, O\_out, I\_se, I\_si, I\_scantest, O\_so,



I\_tpclk,

I\_test\_si2,

O\_test\_so2);

## //INPUT 27; OUTPUT 10 (TOTAL:37)

input [3:0] I\_control; input [7:0] I\_memaddr; input [7:0] I\_in; output [7:0] O\_out; input I\_clk, I\_rst, I\_se, I\_si, I\_scantest, I\_tpclk, I\_test\_si2; output O\_so, O\_test\_so2;

```
//INTERCONNECTIONS
wire [3:0] control;
wire [7:0] memaddr;
wire [7:0] in;
wire [7:0] out;
wire clk, rst, se, si, scantest, tpclk, test_si2;
wire so, test_so2;
```

## //CORE: note that the reference module in the TOP MODULE should be //"CALL BY REFERENCE"

cpu cpu( .clk(clk), .rst(rst), .control(control), .memaddr(memaddr), .in(in), .out(out), .se(se), .si(si), .scantest(scantest), .so(so), .tpclk(tpclk), .test\_si2(test\_si2), .test\_so2(test\_so2)

);

PDIDGZ_33	PAD_clk	(.PAD(I_clk),	.C(clk)); //27 pads
PDIDGZ_33	PAD_rst	(.PAD(I_rst),	.C(rst));
PDIDGZ_33	PAD_control0	(.PAD(I_control[0]),	.C(control[0]));
PDIDGZ_33	PAD_control1	(.PAD(I_control[1]),	.C(control[1]));
PDIDGZ_33	PAD_control2	(.PAD(I_control[2]),	.C(control[2]));
PDIDGZ_33	PAD_control3	(.PAD(I_control[3]),	.C(control[3]));
PDIDGZ_33	PAD_memaddr0	(.PAD(I_memaddr[0]),	.C(memaddr[0]));
PDIDGZ_33	PAD_memaddr1	(.PAD(I_memaddr[1]),	.C(memaddr[1]));
PDIDGZ_33	PAD_memaddr2	(.PAD(I_memaddr[2]),	.C(memaddr[2]));
PDIDGZ_33	PAD_memaddr3	(.PAD(I_memaddr[3]),	.C(memaddr[3]));
PDIDGZ_33	PAD_memaddr4	(.PAD(I_memaddr[4]),	.C(memaddr[4]));
PDIDGZ_33	PAD_memaddr5	(.PAD(I_memaddr[5]),	.C(memaddr[5]));
PDIDGZ_33	PAD_memaddr6	(.PAD(I_memaddr[6]),	.C(memaddr[6]));
PDIDGZ_33	PAD_memaddr7	(.PAD(I_memaddr[7]),	.C(memaddr[7]));
PDIDGZ_33	PAD_in0	(.PAD(I_in[0]),	.C(in[0]));
PDIDGZ_33	PAD_in1	(.PAD(I_in[1]),	.C(in[1]));
PDIDGZ_33	PAD_in2	(.PAD(I_in[2]),	.C(in[2]));
PDIDGZ_33	PAD_in3	(.PAD(I_in[3]),	.C(in[3]));
PDIDGZ_33	PAD_in4	(.PAD(I_in[4]),	.C(in[4]));
PDIDGZ_33	PAD_in5	(.PAD(I_in[5]),	.C(in[5]));
PDIDGZ_33	PAD_in6	(.PAD(I_in[6]),	.C(in[6]));
PDIDGZ_33	PAD_in7	(.PAD(I_in[7]),	.C(in[7]));
PDIDGZ_33	PAD_se	(. <b>PAD</b> ( <b>I</b> _se),	.C(se));
PDIDGZ_33	PAD_si	(. <b>PAD</b> ( <b>I</b> _si),	.C(si));
PDIDGZ_33	PAD_scantest	(.PAD(I_scantest),	.C(scantest));
PDIDGZ_33	PAD_tpclk	(.PAD(I_tpclk),	.C(tpclk));
PDIDGZ_33	PAD_test_si2	(.PAD(I_test_si2),	.C(test_si2));
PDO24CDG_3	33 PAD_out0	(.PAD(O_out[0]),	.I(out[0])); //10 pads
PDO24CDG_3	33 PAD_out1	(.PAD(O_out[1]),	.I(out[1]));
PDO24CDG_3	33 PAD_out2	(.PAD(O_out[2]),	.I(out[2]));
PDO24CDG_3	33 PAD_out3	(.PAD(O_out[3]),	.I(out[3]));
PDO24CDG_3	33 PAD_out4	(.PAD(O_out[4]),	.I(out[4]));
PDO24CDG_3	33 PAD_out5	(.PAD(O_out[5]),	.I(out[5]));
PDO24CDG_3	33 PAD_out6	(.PAD(O_out[6]),	.I(out[6]));
PDO24CDG_3	33 PAD_out7	(.PAD(O_out[7]),	.I(out[7]));



PDO24CDG\_33 PAD\_so PDO24CDG\_33 PAD\_test\_so2 (.PAD(O\_so), (.PAD(O\_test\_so2), .I(so)); .I(test\_so2));

endmodule

• Open <cpu.sdc>, then save as <CHIP.sdc>. Then, modify it as follows:

# Created by Design Compiler write\_sdc on Sun Jul 20 17:09:22 2008

set\_false\_path -from [get\_ports {I\_rst}]

set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_scantest}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_si}] set input delay 0.34 -clock [get clocks {CLK1}] [get ports {I se}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[0]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[1]}] set input delay 0.34 -clock [get clocks {CLK1}] [get ports {I in[2]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[3]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[4]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[5]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[6]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_in[7]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[0]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[1]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[2]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[3]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[4]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[5]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[6]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_memaddr[7]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_control[0]}]



set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_control[1]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_control[2]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_control[3]}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_rst}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_clk}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_tpclk}] set\_input\_delay 0.34 -clock [get\_clocks {CLK1}] [get\_ports {I\_tpclk}]

set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_so}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[0]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[1]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[2]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[3]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[4]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[4]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[5]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[5]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[6]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[7]}] set\_output\_delay 1.6 -clock [get\_clocks {CLK1}] [get\_ports {O\_out[7]}]

set drive 0.288001 [get ports {I clk}] set\_drive 0.288001 [get\_ports {I\_rst}] set\_drive 0.288001 [get\_ports {I\_control[3]}] set drive 0.288001 [get ports {I control[2]}] set\_drive 0.288001 [get\_ports {I\_control[1]}] set\_drive 0.288001 [get\_ports {I\_control[0]}] set\_drive 0.288001 [get\_ports {I\_memaddr[7]}] set\_drive 0.288001 [get\_ports {I\_memaddr[6]}] set\_drive 0.288001 [get\_ports {I\_memaddr[5]}] set drive 0.288001 [get ports {I memaddr[4]}] set\_drive 0.288001 [get\_ports {I\_memaddr[3]}] set\_drive 0.288001 [get\_ports {I\_memaddr[2]}] set\_drive 0.288001 [get\_ports {I\_memaddr[1]}] set\_drive 0.288001 [get\_ports {I\_memaddr[0]}] set\_drive 0.288001 [get\_ports {I\_in[7]}] set\_drive 0.288001 [get\_ports {I\_in[6]}] set\_drive 0.288001 [get\_ports {I\_in[5]}] set\_drive 0.288001 [get\_ports {I\_in[4]}]



set\_drive 0.288001 [get\_ports {I\_in[3]}] set\_drive 0.288001 [get\_ports {I\_in[2]}] set\_drive 0.288001 [get\_ports {I\_in[1]}] set\_drive 0.288001 [get\_ports {I\_in[0]}] set\_drive 0.288001 [get\_ports {I\_se}] set\_drive 0.288001 [get\_ports {I\_si}] set\_drive 0.288001 [get\_ports {I\_scantest}] set\_drive 0.288001 [get\_ports {I\_tpclk}] set\_drive 0.288001 [get\_ports {I\_test\_si2}]

set\_load -pin\_load 0.06553 [get\_ports {O\_out[7]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[6]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[5]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[4]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[3]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[2]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[1]}] set\_load -pin\_load 0.06553 [get\_ports {O\_out[0]}] set\_load -pin\_load 0.06553 [get\_ports {O\_so}] set\_load -pin\_load 0.06553 [get\_ports {O\_so}]

- Why the modification?
  - A clock source CLK1 is declared representing the external tester clock
  - Only the input delay, input drive, output delay, and output load are remained, since other settings (i.e. wire load model) will be ignored in the SOC Encounter.

• Edit <CHIP.ioc> as follows: (Here a 48-pin package is assumed)

Version: 1		
Pad: CORNER0	NW	PCORNERDG
Pad: PAD_clk	Ν	
Pad: PAD_rst	Ν	
Pad: PAD_CoreVDD1	Ν	PVDD1DGZ_33
Pad: PAD_CoreVSS1	Ν	PVSS1DGZ_33
Pad: PAD_control0	Ν	
Pad: PAD_control1	Ν	
Pad: PAD_control2	Ν	
Pad: PAD_control3	Ν	



Pad: PAD_memaddr0	Ν
Pad: PAD_IOVDD1	N PVDD2DGZ_33
Pad: PAD_memaddr1	Ν
Pad: PAD_memaddr2	Ν
Pad: CORNER1	NE PCORNERDG
Pad: PAD_memaddr3	E
Pad: PAD_memaddr4	E
Pad: PAD_CoreVDD2	E PVDD1DGZ_33
Pad: PAD_CoreVSS2	E PVSS1DGZ_33
Pad: PAD_memaddr5	E
Pad: PAD_memaddr6	E
Pad: PAD_memaddr7	E
Pad: PAD_in0	E
Pad: PAD_in1	E
Pad: PAD_IOVSS1	E PVSS2DGZ_33
Pad: PAD_in2	E
Pad: PAD_in3	Ε
Pad: CORNER2	SE PCORNERDG
Pad: CORNER2 Pad: PAD_in4	SE PCORNERDG S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5	SE PCORNERDG S S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3	SE PCORNERDG S S S PVDD1DGZ_33
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3	SE PCORNERDG S S S PVDD1DGZ_33 S PVSS1DGZ_33
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6	SE PCORNERDG S S S PVDD1DGZ_33 S PVSS1DGZ_33 S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7	SE PCORNERDG S S S PVDD1DGZ_33 S PVSS1DGZ_33 S S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se	SE PCORNERDG S S S PVDD1DGZ_33 S PVSS1DGZ_33 S S S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_si	SE PCORNERDG S S S PVDD1DGZ_33 S PVSS1DGZ_33 S S S S
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_si Pad: PAD_si	SEPCORNERDGSSSPVDD1DGZ_33SSSSSSSSSSS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_scantest Pad: PAD_IOVDD2	SEPCORNERDGSSSPVDD1DGZ_33SPVSS1DGZ_33SS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_si Pad: PAD_scantest Pad: PAD_lOVDD2 Pad: PAD_tpclk	SEPCORNERDGSSSPVDD1DGZ_33SPVSS1DGZ_33SSSSSSSSSSSSSSS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_scantest Pad: PAD_tpclk Pad: PAD_tpclk	SEPCORNERDGSSSPVDD1DGZ_33SPVSS1DGZ_33SSSSSSSSSSSSSS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_scantest Pad: PAD_loVDD2 Pad: PAD_tpclk Pad: PAD_test_si2	SEPCORNERDGSSSPVDD1DGZ_33SPVSS1DGZ_33SS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_lout0	SEPCORNERDGSSPVDD1DGZ_33SPVSS1DGZ_33SSS
Pad: CORNER2 Pad: PAD_in4 Pad: PAD_in5 Pad: PAD_CoreVDD3 Pad: PAD_CoreVSS3 Pad: PAD_in6 Pad: PAD_in7 Pad: PAD_se Pad: PAD_se Pad: PAD_si Pad: PAD_si Pad: PAD_si Pad: PAD_tpclk Pad: PAD_tpclk Pad: PAD_tpclk Pad: PAD_test_si2 Pad: CORNER3 Pad: PAD_out0 Pad: PAD_out1	SEPCORNERDGSSPVDD1DGZ_33SPVSS1DGZ_33SSS



Pad: PAD_out2	W	
Pad: PAD_out3	W	
Pad: PAD_out4	W	
Pad: PAD_out5	W	
Pad: PAD_out6	W	
Pad: PAD_out7	W	
Pad: PAD_IOVSS2	W	PVSS2DGZ_33
Pad: PAD_so	W	
Pad: PAD_test_so2	W	

#### Placement and Routing Using <SOC Encounter>

- File preparations:
  - CPU\_CHIP.vg gate-level netlist (i.e. from Design Compiler)
  - CHIP.sdc definitions of I/O driving, delays, and loadings, ...etc (i.e. from Design Compiler, needed to be modified)
  - CHIP.ioc rearrangement of I/O positions (i.e. user edit)
  - Folder "lib" library files of the memory, standard cells, and PADs, which define the electronic parameters of the cells (i.e. from Artisan and CIC Design Kit)
  - Folder "lef" LEF files of the memory, standard cells, which define the technology design rules for P&R (i.e. from Artisan and CIC Design Kit. If you have full-custom design, its lef file is generated by Abstractor)
  - Folder "library.cl" QX library (i.e. from CIC Design Kit)
  - Others: tsmc018.capTbl (capacitance values for timing evaluation), icecaps\_5lm.tch (QX technology file), addIoFiller.cmd (IO filler execution file), addbonding.pl (bonding PAD execution file), ioPad.list (i.e. from CIC Design Kit), streamOut.map (i.e. user edit)



**Cell-Based Training LAB** 

/Solution_LAB08/SOC/lef	/usr3/zwtseng/Solution_	LAB08/SOC/lib
- <u>-</u> (go up)	(go up)	RAM_64B_fast@-40C_syn.lib
RAM_64B.vclef	RAM_64B_fast@0C_syn.lib	RAM_64B_slow_syn.lib
antenna_6_cic.lef	RAM_64B_typical_syn.lib	fast.lib
tpz973g_51m_cic.lef	slow.lib	tpz973gbc.lib
tsmc18_6lm_antenna_cic.lef	tpz973gtc.lib	tpz973gwc.lib
tsmc18_6lm_cic.lef	typical.lib	

Fig. LIB and LEF files

- encounter **-** (In the "SOC/" folder)
- Design  $\rightarrow$  Import Design
  - <Basic> Tag
  - Files: CPU\_CHIP.vg
  - Top Cell: CPU\_CHIP (By User)
  - LEF Files:
    - ◆ select the following files and press "Add" button (In " SOC/lef/ ")
      - tsmc090lk\_9lm\_2thick\_tech.lef (should be selected first)
        - tsmc090lk\_9lm\_2thick\_tech.lef
        - tsmc090nvt\_macros.lef
      - tpzn90gv3\_91m.lef
        - tpzn90gv3\_9lm.lef
        - antenna.lef
      - tpbn90gv\_9lm.lef
    - ◆ Then, revise the "\*.lef" into "all files" in the blank of Filter ◀┛
    - Then, select the following files in the same way:
      - SRAM\_SP\_ADV.vclef
  - IO Assignment File: CHIP.ioc

#### <u>< Advanced/Power> Tag</u>

• Power Nets: VDD



- ♦ Ground Nets: VSS
- Save... → CHIP.conf  $\blacktriangleleft$
- Exit from soc encounter GUI interface

0. vi CHIP.conf

>>>Please find the following line and modify<<<

- 1. <u>set rda\_Input(ui\_timingcon\_file,full)</u> "" Not this line set rda\_Input(ui\_timingcon\_file) ""
- Timing Constraint File: CHIP.sdc
- set rda\_Input(ui\_timingcon\_file) <u>"CHIP.sdc"</u>

#### 2. set rda\_Input(ui\_timelib,max) ""

\*Please check the library names of the timing libraries of SRAM in tt ss ff corners. If the library names are the same please rename these timing libraries.

- Max Timing Libraries:
  - (In " SOC/LIB/ ")
    - LEF/slow.lib
    - LEF/tpzn90gv3wc.lib
    - LEF/SRAM\_SP\_AVD\_ss\_0.9\_125.0\_syn.lib
- set rda\_Input(ui\_timelib,max) "LEF/slow.lib LEF/tpzn90gv3wc.lib LEF/SRAM\_SP\_AVD\_ss\_0.9\_125.0\_syn.lib "
- 3. set rda\_Input(ui\_timelib,min) ""
- Min Timing Libraries:
  - (In " SOC/LIB/ ")
    - LEF/fast.lib
    - LEF/tpzn90gv3bc.lib
    - LEF/SRAM\_SP\_AVD\_ff\_1.1\_-40.0\_syn.lib
    - LEF/SRAM\_SP\_AVD\_ff\_1.1\_.0\_syn.lib
- set rda\_Input(ui\_timelib,min) "LEF/fast.lib LEF/tpzn90gv3bc.lib LEF/SRAM\_SP\_AVD\_ss\_1.1\_-40.0\_syn.lib LEF/SRAM\_SP\_AVD\_ss\_1.1\_.0\_syn.lib"
- 4. set rda\_Input(ui\_timelib) ""
  - Common Timing Libraries:
    - ◆ (In " SOC/LIB/ ")



- LEF/typical.lib
- LEF/tpzn90gv3tc.lib
- LEF/SRAM\_SP\_AVD\_t t\_1.0\_25.0\_syn.lib
- set rda\_Input(ui\_timelib) "LEF/typical.lib LEF/tpzn90gv3tc.lib LEF/SRAM\_SP\_AVD\_tt\_1.0\_25.0\_syn.lib"
- 5. set rda\_Input(ui\_captbl\_cap) ""
- capacitor libraries:
  - (In "SOC")
    - t90g\_rct.CapTbl
    - t90g\_rcb.CapTbl
    - t90g\_rcw.CapTbl
- set rda\_Input(ui\_captbl\_cap) " -typical t90g\_rct.CapTbl -best t90g\_rcb.CapTbl -worst t90g\_rcw.CapTbl"
- 6. set rda\_Input(ui\_qxtech\_file) ""
- QX Technique file
  - ♦ (In "SOC")
    - icecaps.tch
- set rda\_Input(ui\_qxtech\_file) "icecaps.tch"
- 7. set rda\_Input(ui\_qxlib\_file) ""
  - QX Technique Direction
- set rda\_Input(ui\_qxlib\_file) "\$cwd/library"
- OK ◄ ┛
- After a period of parsing time, you can see the initial window as following Fig. .
- Open <encounter.log>, search the key word "skipped". Make sure that all the skipped counts of time constraints are 0.



Encounter(R) RTL-to-GDSII System 10.1.	3 - /home/phd/cshou/Documents/Res	earch_Document/v_code/bi	sr_low_v_sram/encour 🗕 🗖 🗙
<u>File Edit View Partition</u> Floorpl <u>an</u> Po <u>w</u> er <u>P</u> lace	<u>O</u> ptimize <u>C</u> lock <u>R</u> oute <u>T</u> iming <u>V</u> erify Opt <u>i</u>	ons PVS Too <u>i</u> s Flow <u>s H</u> elp	cādence
I⊨ =    > <   0 №    <b>2</b>    <b>2 2</b>	् २. 🔄 ९ २ २ । 🕭 की ाक्ष A २. ९. २. ९. ४. ⇔ ५. १०	💷 💀 농 🐳 🛯 🗂	e 2 الا لا ال
-70.1%         PAD_so           PAD_so         PAD_coreVSS10           PAD_coreVSS10         PAD_coreVSS10           PAD_coreVSS10         PAD_coreVSS10           PAD_coreVSS10         PAD_coreVSS10           PAD_coreVSS10         PAD_coreVSS10           PAD_screptor         PAD_screptor           PAD_ses         PAD_ses           XIE         CORNER3	CORNER1 PAD_10V5 PAD_10V5 PAD_10V5 PAD_183ie PAD_Core PAD_CSI PAD_10V5 PAD_10V5 PAD_10V5 PAD_10V5 PAD_10V5 PAD_10V5	xcore/xstan0	Layer Control       5 ×         All Colors       Fioorplan         Fioorplan Layers       ✓         Module       ✓         Fence       ✓         Guide       ✓         Obstruct       ✓         Region       ✓         Area Density       ✓         Instance       ✓         Std. Cell       ✓         Block       ✓         IO Cell       ✓         Area IO Cell       ✓         Area IO Cell       ✓         Word View       ✓
Click to select single object. Shift+Click to de/select multip	le objects.	Q  SelNum:0 (1	645.717, 623.222) In Memory

#### Fig. Initial wondow

• Power  $\rightarrow$  Connect Global Nets



- Select "Pin" and type "VDD" in "Pin Name(s)", then type "VDD" in "To Global Net" and press "Add to List"
- Select "Pin" and type "VSS" in "Pin Name(s)", then type "VSS" in "To Global Net" and press "Add to List"



- Select "**Net Basename**" and type in "VDD", then type "VDD" in "To Global Net" and press "Add to List"
- Select "**Net Basename**" and type in "VSS", then type "VSS" in "To Global Net" and press "Add to List"
- $\blacksquare \quad \text{Apply} \rightarrow \text{Check} \rightarrow \text{Close} (X)$
- Since the Tie High and Tie Low cells will be handled later, the following warning messages are shown.

encounter l> Warning: term EMA[2] of inst xcore/xsram0 is not connect to global special net. Warning: term EMA[1] of inst xcore/xsram0 is not connect to global special net. Warning: term EMA[0] of inst xcore/xsram0 is not connect to global special net.

Fig. Warning messages

#### • In the "consol"

- encounter> specifyScanChain scan1 -start PAD\_si/C -stop PAD\_so/I
- encounter> specifyScanChain scan2 -start PAD\_test\_si2/C -stop PAD\_test\_so2/I

(i.e. There are 2 scan chain in my design)

■ encounter> scantrace ◄ (i.e. scan chain tracing)

```
encounter 3> scantrace
Tracing scan chain: scan1
Successfully traced scan group scan1 (25 elements; 24 scan bits).
Tracing scan chain: scan2
Successfully traced scan group scan2 (29 elements; 28 scan bits).
*** Scan Trace Summary:
Successfully traced scan group scan1 (25 elements; 24 scan bits).
Successfully traced scan group scan2 (29 elements; 24 scan bits).
Successfully traced scan group scan2 (29 elements; 28 scan bits).
Successfully traced scan group scan2 (29 elements; 28 scan bits).
Successfully traced 2 scan groups (total 54 elements; 52 scan bits).
INFO: Passed sanity check on scan group scan1.
INFO: Passed sanity check on scan group scan2.
*** Scan Sanity Check Summary:
*** 2 scan groups passed sanity check.
```

Fig. Scan chain tracing

- Floorplan  $\rightarrow$  Specify Floorplan
  - Ratio (H/W): 1
  - Core Utilization: 0.65
  - Core to Left: 80
  - Core to Top: 80
  - Core to Right: 80
  - Core to Bottom: 80
  - OK

Design Dimensions		
Specify By: 🔶 Size 😞 Die/IO/Core Coo	rdinates	
Core Size by:      Aspect Ratio:	Ratio (H/W)	1
	+ Core Utilization:	0.65
	🗢 Cell Utilization:	0.047383
Dimension:	Width:	479.38
	Height	478.8
Die Size by:	Width:	950.0
	Height	950.0
Core Margins by: 🔶 Core to IO Bounda	ry	
Core to Die Bound	ary	
Core to Left: 8	0 Core to Top.	80
Core to Right 8	Core to Bottom	80
Die Size Calculation Use: 🗸 Max I	O Height 🔶 Min IO I	reight
Floorplan Origin at: 🔶 Lower	Left Corner 🕹 Cente	r
		Unit: Micr





- The IOs of SRAM are along the bottom of SRAM. Now, we want to change the direction of SRAM IOs.
  - Press and draw the hard-block "xsram0" to the right-up corner
    - i.e. In another way, you can type "setObjFPlanBox Instance xcore/xsram0 324 327 587.66 588.285." in the consol
  - Select the memory "xsram0" and press the hot key "R".
    - Or using: Floorplan  $\rightarrow$  Edit Floorplan  $\rightarrow$  Flip/Rotate Instances...
  - Select R270  $\rightarrow$  Apply  $\rightarrow$  Cancel



Fig. RAM placement

- Press  $\square$  and select the SRAM. Then, Floorplan  $\rightarrow$  Edit Floorplan  $\rightarrow$  Set
  - Instance Placement Status...
  - $\blacksquare \quad \text{Selected} \rightarrow \text{Apply} \rightarrow \text{OK}$
  - i.e. this step keeps the RAM fixed in the selected position avoiding any



changes.

- Until now, the floorplan and hard blocks placement are done.
- Select the RAM and then, Floorplan  $\rightarrow$  Edit Floorplan  $\rightarrow$  Edit Halo...
  - Selected Block/Pad
  - Placement Halo  $\rightarrow$  Add/Update Block Halo
  - Type 20 in the blanks Top, Bottom, Left, and Right. Apply  $\rightarrow$  OK
  - i.e. this step keeps the RAM surrounded by the gap in which no other cells can be place in.



Fig. Hard block halos

- Place  $\rightarrow$  Place Standard Cells...
  - Select "Run Full Placement"
  - De-select "Include Pre-Place Optimization"
  - Select "Include In-Place Optimization"
  - OK

Place _ 🗆 🗙
🖲 Run Full Placemen 🔾 Run Incremental Placement 🔾 Run Placement In Floorplan Mode
Optimization Options
Include Pre-Place Optimization — De-select
✓ Include In-Place Optimization Select
Number of Local CPU(s): 1 Set Multiple CPU
<u>Q</u> K <u>Apply M</u> ode <u>D</u> efaults <u>Cancel H</u> elp

Fig. The standard cell placement setting



• After a period later (i.e. could be very long for a huge design), the placement of

standard cells is done. Press (physical view), you can see all the cells are placed.





- Until now, the hard blocks and standard cells placement are done.
- Timing  $\rightarrow$  Report Timing...
  - Select "Pre-CTS"
  - Select "setup" (i.e. setup time evaluation. "hold" is selected for the hold time evaluation)
  - OK
  - Does there any violation paths exist?

..... timeDesign Summary Setup mode | all | reg2reg | in2reg | reg2out | in2out | clkgate | Т WNS (ns): | 4.167 | 4.167 | 6.281 | N/A | N/A | N/A | TNS (ns):| 0.000 | 0.000 | 0.000 | N/A | N/A | N/A | Violating Paths: 0 0 0 0 N/A N/A N/A N/A All Paths: 279 154 235 N/A N/A N/A N/A | N/A ------| Real | Total - I +-----DRVs | Nr nets(terms) | Worst Vio | Nr nets(terms) | 

 max\_cap
 0
 0
 0.000
 1
 1
 1

 max\_tran
 1
 1
 1
 -0.090
 2
 2
 2

 max\_fanout
 1
 1
 1
 -36
 2
 2
 2

 Т T Т ----+-----



Fig. Timing analysis results

- If the timing is failed, the in-place optimization of the timing should be performed. Otherwise, you can ignore this step.
  - Optimize  $\rightarrow$  Optimize Design
    - ♦ Select "Pre-CTS"
    - ♦ Select "setup"

	timeDesign Summary
9 DRRT	Worst Slack: -0.237ns TWS: -0.237ns Violating Paths: 1 Pathgroup Slacks reg2reg: 3.594ns in2reg: 2.703ns reg2out: -0.237ns in2out: 8.586ns vensity: 14.415% touting Overflow: 0.00% H and 0.00% V veal DRV (fanout, cap, tran): (2, 2, 0) otal DRV (fanout, cap, tran): (3, 2, 0)
R T T	eported timing to dir timingReports otal CPU time: 1.68 sec otal Real time: 2.0 sec otal Memory Usage: 247.594208 Mbytes
_	optDesign Final Summary
9 DRRT	etup mode Worst Slack: 0.612ns TNS: 0.000ns Violating Paths: 0 Pathgroup Slacks reg2reg: 3.593ns in2reg: 2.895ns in2out: 0.612ns in2out: 9.166ns ensity: 14.546% outing overflow: 0.00% H and 0.00% V eal DRV (fanout, cap, tran): (2, 0, 0) otal DRV (fanout, cap, tran): (3, 0, 0)
* *	*optDesign cpu = 0:0:8, real = 0:0:9, mem = 247.6M ** ** Finished optDesign ***

Fig. Timing analysis before/after IPO



Fig. Trail routing

• Since the timing analysis performs a simple routing (i.e. trial route)



operation, they should be removed avoiding the hazard in the next step *"power ring generation"* 

■ Place  $\rightarrow$  Refine Placement



Fig. Refine placement

- File  $\rightarrow$  Save Design
  - **Type in "PLACE.enc"**  $\rightarrow$  Save
- Power  $\rightarrow$  Power Planning  $\rightarrow$  Add Rings...
  - $\blacksquare \quad \underline{\langle Basic \rangle Tag}$ 
    - Select Power nets: VDD, VSS

Net(s):	VDD VSS	]
	5	

- Change the layer of "Top" and "Bottom" to "METAL9 H"; "Left" and "Right" to "METAL8 V"
- Set all "Width" to 3 and p "Spacing" to 0.805 Press "-----"



Fig. Ring setting

- Advanced>Tag
  - Select "Use wire group"
  - Select "Interleaving" and type in "9" to "Number of bits"
- OK



~Wire Group	
🗹 Use wire group	
🗹 Interleaving	
Number of bits:	9





Fig. Power rings

- Route  $\rightarrow$  Special Route...
  - Only "Pad pins" is selected, then press OK

SRoute	
📃 Block Pins 🗹 Pad Pins 🔲 Pad Rings	Follow Pins Secondary Power Pins
Bouting Control	

Fig. Select Pad pins only





Fig. Power PAD connections

- Design  $\rightarrow$  Save Design As  $\rightarrow$  SoCE
  - Type in "PRE\_STRIPE.enc"  $\rightarrow$  Save
- Power →Power Planning → Add Stripes...
   (In METAL 9)
  - $\blacksquare$  <Basic>Tag
  - Layer: METAL9
  - Width: 1.5
  - Spacing 0.805
  - Set-to-set distance: 60
  - Start from: bottom
  - X from left: 20
  - X from right: 20
  - $\blacksquare \quad \underline{< Advanced > Tag}$
  - Select "Omit stripes inside block rings"
  - Select "Omit stripes over selected blocks/domains"
  - Pad/Core ring connection  $\rightarrow$  Select "Allow jogging"
  - Block ring connection  $\rightarrow$  Select "Allow jogging"
  - De-select "Merge with block rings if spacing less than: 0.56"
  - Select "Use wire group"
  - Select "Interleaving" and set "Number of bits" to "5"
  - <u><Via Generation> Tag</u>
  - Select "Use exact overlap area on partially intersecting wires"
  - Select "Split vias while encountering Obs and different Wires/Pins"
  - De-select "Connect to orthogonal targets only"



- Select "Generate same-sized stack vias while encountering macro Pins/Obs"
- OK

	contracted reducts (Friday 1)	
asic Advanced Via Ge	neration	
Set Configuration		
Net(s)		
Layer M9 .	$\longrightarrow$ META	10
Direction: 🔍 Vertical 🔳	Horizontal IVIL/1/	
Width: 1.5		
Spacing: 0.805	Update	
Set Pattern		
Set-to-set distance:	60	
Number of sets:		
O Bumps & Driff	Detween	
Over P/G pins Fin lay	er Januariavet Mas pro with 0	
🛎 Marter none	G Selected blocks G All blocks	
Stripe Boundary		
Core ring		
C Patt ring C Inner	# Outer	
🔾 Design boundary 🔗	Crisate pina	
<ul> <li>Each selected block/dom</li> </ul>	andence	
C Searchy rankamentar area		
<ul> <li>Specify rectlinear area</li> </ul>		
Destil and Steine		
Stad from A hollow	( ) how (	
Balatua from core or sale	arted area	
V from Intel 20	V from bottom 20	
1 mm mb. 20	A second metalogical second	

Irine Brooking	
Omit stripes inside block rings	
Omit stripes over selected blocks/domains	
Switch layer over obstructions	
. Specify area blockage	
	(MipHCReb)
anist Connection Context	
Pad/Core ring connection	
🛃 Allow jagging	
Nock ring connection	
🖌 Allow jagging	
Marge with block rings if spacing less than:	0.42
Maximum length of same layer jog: 0.84	
aver Control for Target Connections	
Pad/Core mgs	
Top limit M9 .	
Bottom limit M8 +	
Bock rings/Over obstructions	
Top limit M9 .	
Dottom limit M0 .	
les Cours	
Use wire group	
😪 Interleaving	
Number of bits: 5	

Specify connection layer	range				
Top stack via layer.	MS .				
Bottom stack via layer	n (MI )	ki			
🖌 Use exact overlap area	on partial	y intersection	g wires		
Split vias while encour	dering Obs	and differen	t net Wires/Pin	6	
Generate same-sized	stack vias v	while encour	tering macro P	Ini/Obs	
Connect to orthogonal	targets only	ý			
Split vias longer than	n	into i	mailer vias		
with center-to-center	step of				
Same layer target exists	, connect t only	0.			
If same layer target exists Same layer target Targets on all lay Make via connections to:	, connect 1 only ers Maxin	num via size Crossov	(%) er 1	Farget	
If same layer target exists Same layer target Targets on all lay Make via connections to:	, connect 1 only ers Maxin	num via size Crossov Widzh	(%) er 1 Height	farget Penetration	Span
If same layer target exists Same layer target Targets on all lay Make via connections to: Pad ring/pm	, connect 1 only ers Maxis	num via size Crossov Width 100	(%) er 1 Heigtt 100	Target Penetration	Span
If same layer target exists Same layer target Targets on all lay Make via connections to: Pad ning/pin Core ring	, connect 1 only ers Maxis 100 100	o: num via size Crossov Width 100 100	(%) er 100 100	Penetration 100 100	Span
If same layer target exists Same layer target Targets on all lay Make via connections to Pad ring/pin Core ring Shipe	Connect 1 only ers 100 100 100	num via size Crossov Width 100 100 100	(%) er Height 100 100	Penetration 100 100 100	Span
If came layer beget oxid. Same layer target Targets on all lay Make via connections to: Pad ring/pin Core ring Shipe Block ring	Connect 1 only ers 100 100 100 100	tim via size Crossov Width 100 100 100 100	(%) er Height 100 100 100 100	arget Penetration 100 100 100 100	Span
If same layer target exist Sime layer target Targets on all ky Make via connections to: Pad ring/pm Core ring Stope Stope Block ring Block pin	Connect 1 only ers Maxis 100 100 100 100 100	time via size Crossov Width 100 100 100 100	(%) er Height 100 100 100 100 100	arget Penebation 100 100 100 100 100	Span
If same layer target exist Sime Layer target Targets on all lay Make via connections to: Plad ring/pin Core ring Stype Millock ring Biock pin Core macro pin	, connect 1 only ers Maxe 100 100 100 100 100 100	rum via size Crossov Width 100 100 100 100 100 100	(%) er Height 100 100 100 100 100	Target Penebration 100 100 100 100 100 100 100 100	Span
If same layer target exist Sime Layer target Targets on all lay Make via connections to: Pad ring/pin Core ring Shipe Block ring Block pin Core fing Block pin Core ring Block pin Core ring Core ring Block pin Core ring Core ring Core ring Shipe Core ring Shipe Core ring Core ring Shipe Core ring Core ring Shipe Core ring Core ring	, connect 1 only ens 100 100 100 100 100 100 100 100 100	rum via size Crossov width 100 100 100 100 100 100	(%) er 1 Height 100 100 100 100 100	Target Penetration 100 100 100 100 100 100 100	Span
r same layer boget oxist Same layer baget Targets on all lay Make via connections to: ✓ Pad rings/pm ✓ Core ring ✓ Shipe ✓ Block ting ✓ Block pin ✓ Cover macro pin → Nahae	, connect 1 only ens 100 100 100 100 100 100 100 100 100 10	a www.via.size Crossev width 100 100 100 100 100 100 100 10	(%) er = 1 100 100 100 100 100 100 100	Target Penetration 100 100 100 100 100 100 100	Span

Basic

Advanced Fig. Power stripes setting

Via Generation

- Power →Power Planning → Add Stripes...
   (In METAL 8)
  - $\blacksquare \quad \underline{\langle Basic \rangle Tag}$
  - Layer: METAL8
  - Start from: bottom
  - Y from top: 20
  - Y from bottom: 20
  - OK



Net(s): Layer: Direction Width: Spacing	VSS VDD METALS	ontal	METAL 8
Set-to     Numbe     Dumps     Dover P     Over P     Over n     Core n     Pad nr     Design     Each s     All dos     Specifi     Specifi	set distance: 60 r of set: 1 VG para Prinsyer: 12 the class and r of the set	torent a pro layer - J Simical to tare pint a	Mar par some <b>p</b>
First/Las Start fro Reladiv Y from Absolu	At Stripe m bottom to to top: [20 y te locations A	np ea from bottom: 20	_

Fig. Power stripe generation

• Power stripes are used to reduce the IR-drop effect. However, some stripes do not connect correctly along the hard blocks.



- Due to GUI remove stripes (unconnected), we need to key in the following command
  - sroute -noBlockPins -noPadRings -noCorePins -noPadPins
     -jogControl { preferWithChanges differentLayer }
     (上面的指令要好好地注意正確性,在一行裡面輸入完成)
- However, some violations occur. We should figure out what's the problem.



Fig. Open violations

- Verify  $\rightarrow$  Violation Browser
  - There're 5 open nets

■ Other (5)				
Connectivity (5)				
🗖 Open (5) Net				
M2	(335.955,	249.355)	(336.005,	249.40
М2	(338.515,	249.355)	(338.565,	249.40
М2	(341.075,	249.355)	(341.125,	249.40
M2	(343.635,	249.355)	(343.685,	249.40
M2	(346.195,	249.355)	(346.245,	249.40

• We can see that the violations are existed on M2, but some M4 wires cover the M2 violations. Therefore, we first block the display of M4. Follow the steps as Fig. below:







• Select each open wire, and press "DEL"





- Tool  $\rightarrow$  Violation browser  $\rightarrow$  Clear Violation  $\rightarrow$  Yes
- Verify  $\rightarrow$  Verify Geometry... $\rightarrow$  OK



• Wire-short violations occur.



Fig. Short violations

- Tools  $\rightarrow$  Violation Browser
  - There're 2 short nets

■ Verify (2)		
■ Short (2)		
🗏 Short (2) Layer	Geom	Net/Cell
M5	NET	VDD
M5	NET	VDD

- Select each short wire, and press "DEL"
- Verify  $\rightarrow$  Verify Geometry... $\rightarrow$  OK
  - Finally, no violations exist

```
Begin Summary ...
 Cells
              : 0
              : 0
  SameNet
 Wiring
              : 0
              : 0
 Antenna
              : 0
 Short
 Overlap
              : 0
End Summary
 Verification Complete : O Viols. O Wrngs.
**********End: VERIFY GEOMETRY*********
*** verify geometry (CPU: 0:00:01.7 MEM: 0.0M)
```

• File  $\rightarrow$  Save Design...



- Type "STRIPE.enc"  $\rightarrow$  Save
- Options  $\rightarrow$  Set Mode  $\rightarrow$  Mode Setup...
  - In the list, select "TieHiLo"
  - Press "Select" and select cells "TIEHI TIELO"
  - Specify Maximum Fanout: 10
  - Specify Maximum Distance: 100
  - OK



Fig. TIEHI & TIELO cells configurations

- Place  $\rightarrow$  Tie HI/LO Cell  $\rightarrow$  Add  $\rightarrow$  OK
  - We can see that 1 TIELO cell and 1 TIEHI cell are placed

```
encounter 5> Options: Max Distance = 100.000 microns, Max Fan-out = 10.
INFO: Total Number of Tie Cells (TIELO) placed: 1
INFO: Total Number of Tie Cells (TIEHI) placed: 1
```

- Clock  $\rightarrow$  Synthesize Clock Tree
  - Press "Gen Spec..."  $\rightarrow$  OK
- Open the file "Clock.ctstch" and revise the "MaxDelay" to"1", since we do not expect that the delay is large as its default value.
- Return to "Synthesize Clock Tree Form", and press "OK"
- Does there any violation paths exist?
- If the timing is failed, the in-place optimization of the timing should be performed. Otherwise, you can ignore this step.
  - Optimize  $\rightarrow$  Optimize design
    - ♦ Select "Post-CTS"
    - ♦ Select "setup"
    - ♦ OK
- Clock→Display→Display Clock Tree...
  - Select "Display Clock Tree" and "All Level"  $\rightarrow$  OK
  - You can see the clock tree as follows:







• The info. of the clock tree is reported in "../clock\_report/clock.report"

-	Text Editor – clock.report
<u>F</u> ile <u>E</u> dit Fo <u>r</u> mat <u>O</u> pt	ions
# CLOCK: PAD_clk/C # # Mode: preRoute # ##################################	****
Nr. of Subtrees Nr. of Sinks Nr. of Level (including Root Rise Input Tran Root Fall Input Tran Max trig. edge delay at Min trig. edge delay at	: 1 : 25 : 0 gates) : 0 : 0.1(ps) : 0.1(ps) sink(R): cpu/RAM_64B/CLK 40.4(ps) sink(R): cpu/regc_reg_5_/CK 4.5(ps)
Rise Phase Delay Fall Phase Delay Trig. Edge Skew Rise Skew Fall Skew Max. Rise Buffer Tran Max. Fall Buffer Tran Max. Fall Sink Tran Min. Rise Buffer Tran Min. Fall Buffer Tran Min. Rise Sink Tran Min. Rise Sink Tran Min. Fall Sink Tran	(Actual)       (Required)         : 4.5~40.4(ps)       0~1000(ps)         : 4.5~40.2(ps)       0~1000(ps)         : 35.9(ps)       300(ps)         : 35.9(ps)       300(ps)         : 35.7(ps)       400(ps)         : 0(ps)       400(ps)         : 221(ps)       400(ps)         : 205.6(ps)       400(ps)         : 0(ps)       0(ps)         : 205.6(ps)       0(ps)         : 0(ps)       0(ps)         : 203(ps)       0(ps)
***** NO Max Transition	Time Violation *****
***** NO Min Transition	Time Violation *****
***** NO Max_Fanout Vio	ation *****

Fig.: clock.report

- Does there any violations exist?
- Design  $\rightarrow$  Save Design
  - Type "CTS.enc"  $\rightarrow$  Save



- Timing  $\rightarrow$  Analyze Timing...
  - Select "Post-CTS"
  - Select "setup" (i.e. setup time evaluation. "hold" is selected for the hold time evaluation)
  - OK
- -----(Power Analysis)-----
- The power analysis is not a mandatory step for the designer.
- Power  $\rightarrow$  Analysis  $\rightarrow$  Edit Pad Location
  - Type "VDD" in "Net" and press " Auto Fetch"
  - Type "VSS" in "Net" and press "Auto Fetch"
  - Save...  $\rightarrow$  CPU\_CHIP.pp  $\rightarrow$  Save $\rightarrow$ Cancle
- Power  $\rightarrow$  Analysis $\rightarrow$ Edit Net Toggle Probability
  - Get Clock
  - Select "CLK1" and press "Edit"
  - Type "0.9" in the blank and press "Add/Replace" (i.e. For strengthen the IR-drop evaluation)
  - Save...  $\rightarrow$  CPU\_CHIP.tg  $\rightarrow$  Save $\rightarrow$  Cancle
- Power  $\rightarrow$  Analysis  $\rightarrow$  Power Analysis  $\rightarrow$  Statistical...
  - Net Names: VDD
  - Select "post-CTS clock"
  - Net Toggle Probability File: CPU\_CHIP.tg
  - Pad Location Files: CPU\_CHIP.pp
  - Instance Voltage File: instance.voltage
  - Press "Apply"
- Power analysis results:
  - What is the average power?
  - What is the worst IR-drop?



power supply: 1.62 v
average power(default): 3.0797e+00 mw
average switching power(default): 3.2080e-01 mw
average internal power(default): 2.7476e+00 mw
average leakage power(default): 1.1298e-02 mw
average user specified nower(default): 0.0000e+00 mw
average power by clock domain category:
clock domain(CLK1, 0.9) · 3.0771e+00 mw
clock tree power : 1.6775e+00 mw
non clock tree power : 1 3996e+00 mw
combinational instance nower : 8 2722e-01 mw
sequential instance nower : 5 7235e-01 mw
unclock domain(0,2) · 2 6510e-03 mw
average nower by cell category:
core: 1 2367e+00 mw
block: 1.8430e+00 mw
io: 0.0000e+00 mw
average power(considered in rail analysis): 3.0796e+00 mw
worst IR drop average analysis: 4.0125e-04 v
number of nodes in rail network: 11450
worst EM:
"M1" 4.2340e-02 mA/u
"M2" 1.3492e-03 mA/u
"M3" 1.9139e-02 mA/u
"M4" 3.9171e-02 mA/u
"M5" 5.7648e-02 mA/u
"V12" 6.7296e-03 mA/cut
"V23" 6 7296e-03 mA/cut
"V34" 8.3367e-03 mA/cut
"V45" 8.3367e-03 mA/cut
biggest toggled net: clk
no of terminal: 26
total cap: 4 0055e+02 ff
*** Power analysis (cpu=0:00:01.3 mem=411.4M) ***
Eig. norven englysig
Fig. power analysis

- Power  $\rightarrow$  Analysis $\rightarrow$ Display  $\rightarrow$  Display Rail Analysis Results
  - Net Name: VDD
  - Select "IRD (V)"
  - IRD Threshold: 0.003
  - Press "Update filter range"
  - OK
  - The IR-drop evaluation will be shown.
  - You can block the "Net" and "Instance" in the right switch bar

	Net				
Filter		_	888		
IRD Threshold	: 0.003				
Update	e filter range				
V: Min:		Max:			
0	- 0.00042857				
0.00042857	- 0.00085714				
0.00085714	- 0.00128571	1			
0.00128571	- 0.00171429				
0.00171429	- 0.00214286				
0.00214286	- 0.00257143	-			
0.00257143	- 0.003				
0.003	- 1.62		882		
			196363		

Instance 🛛 🔛 🗖



#### Fig. IR-drop

- Power  $\rightarrow$  Analysis $\rightarrow$ Display  $\rightarrow$  Display Rail Analysis Results
  - Net Name: VDD
  - Select "EM (J/Jmax)"
  - Press "Update EM limit"
  - OK
- Design  $\rightarrow$  Save Design As  $\rightarrow$  SoCE
  - Type "POWER\_ANA.enc"  $\rightarrow$  Save
- Route  $\rightarrow$  Special Route...
  - Only "Follow Pin" is selected, then press OK
- We can see that all the standard cells are connected with horizontal power lines



Fig. Power lines connection for standard cells

- Place  $\rightarrow$  Refine Placement..  $\rightarrow$  OK
- Verify  $\rightarrow$  Verify Geometry... $\rightarrow$  OK
  - Does there any violation occur?
- Verify→Verify Connectivity...
  - Select "Special Only"  $\rightarrow$  OK
  - Does there any violation occur?
    - 5 antenna violations:

■ Verify (5)		
■ Connectivity (5)		
🗏 Antenna (5) Net		
M4	VSS	
м4	VSS	

• Select the wire with violation, and press "Shift+T" to kill the violation





Fig. Antenna violations

- Re-do the verification procedure. Check whether any violation occurs.
- encounter> source addIoFiller.cmd
  - i.e. this step place the IO fillers between the IOs. But this design is "PAD-limit", the space for place the IO filler is limited.

CORNERA								CORNER1
PAD so					<b>.</b>			PAD IOVSS5
								101212121212121212121212121212121212121
PAD scantest								PAD IOVDD5
(8)3)108312588W885888								
PAD CoreVSS10							- HARARA	PAD BSIen
01218925360/685388							нинини	(@);;;)]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
PAD CoreVDD10								PAD CoreVSS4
[8]]]][8]]][8]]][8]]][8]]][8]]][8]]][8								(8) 21 8 8 3 3 81 3 82 4 888
PAD IOVSS9	анананан							PAD CoreVDD4
	HUHUHUHU							(0)#INEE===================================
PAD IOVDD9	ыюнонон	U						PAD BSI
(8)3168=586W382888	HUHUHUHU	AF VE	ur pras	alan no		<b>***</b>	<b>האמאמאמא</b>	
PAD st			_	_	_	_	หมายคมที่	PAD IOVSS3
(0) # ( R 4 = 5 36 W 66 88 8								()))))))))))))))))))))))))))))))))))))
PAD se								PAD IOVDD3
(8);;;;[4];;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		-	<b>*</b>		÷.	-		
<u> </u>								
CORNER3		<b>89 8 8</b>						CORNER2

- File  $\rightarrow$  Save Design
  - Type "POWER\_ROUTE.enc"  $\rightarrow$  Save
- Route  $\rightarrow$  NanoRoute  $\rightarrow$  Route...
  - Select "Timing Driven"
    - Select "SI Driven"
    - Press "Attribute"
      - Select "Net Type(s)"

5	NanoRoute/Attrib	outes						×
	Net Attributes	Clock Nets	External Nets	Critical N	ets i Selec	ted Nets		
	<ul> <li>Net Name(s):</li> </ul>	,		-				
	Skip Antenna	💠 TRUE 💠 FAI	.se 🔶 asis	Top Layer	ASIS	Bottom Layer	ASIS	
	Skip Routing	💠 TRUE 💠 FAI	.se 🔶 asis	Weight	10	Spacing	1	
	Avoid Detour	🔶 TRUE 🔶 FAI	.se 💠 Asis	Shield Net(s)	ASIS			
	SI Prevention	$\diamondsuit$ true $\diamondsuit$ fai	.se 🔶 asis	Nondefault Rule	ASIS			
	SI Post Route Fix	$\diamondsuit$ TRUE $\diamondsuit$ FAI	.se 🔶 asis	Pattern	ASIS			
	<u>0</u> K	Appl	у	Select	Cance	1	Help	



- Select "Clock Nets"
- Weight: 10
- ♦ Spacing: 1
- Avoid Detour: TRUE
- ♦ OK
- OK
- Does there any violation occur?
- Timing  $\rightarrow$  Analyze Timing...
  - Select "Post-Route"
  - Select "setup" (i.e. setup time evaluation. "hold" is selected for the hold time evaluation)
  - OK
- Design  $\rightarrow$  Save Design As  $\rightarrow$  SoCE
  - Type "ROUTE.enc"  $\rightarrow$  Save
- Place  $\rightarrow$  Physical Cells  $\rightarrow$  Add Filler...
  - $\bullet \quad \text{Cell Name(s)} \rightarrow \text{Select}$
  - Add all fillers to the left
  - Close
  - OK



Fig. Add Filler

· ) ·	
R 157	FILLER 159
R 147	FILLER 149
R 187	FILLER 139
R 127	FILLER 129
R 117	FILLER_119
BR 107	FILLER 109
DECTRUCT	en de Ress
R SS	



#### Fig. Core fillers

■ How many core fillers are added?

\*INF0: Iteration 0-#1, Found 3068 DRC violations (real: 0:00:01.0). For 1395 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#2, Found 1866 DRC violations (real: 0:00:01.0). For 903 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#3, Found 834 DRC violations (real: 0:00:00.0). For 422 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#4, Found 229 DRC violations (real: 0:00:01.0). For 123 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#4, Found 229 DRC violations (real: 0:00:00.0). For 123 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#5, Found 42 DRC violations (real: 0:00:00.0). For 27 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#6, Found 25 DRC violations (real: 0:00:00.0). For 15 new insts, \*\*\* Applied 2 GNC rules (cpu = 0:00:00.0) \*INF0: Iteration 0-#7, Found 0 DRC violation (real: 0:00:01.0). \*INF0: Adding fillers to top-module. \*INF0: Added 0 filler inst of any cell-type. For 0 new insts, \*\*\* Applied 0 GNC rules. \*INF0: End DRC Checks. (real: 0:00:04.0). \*INF0: Replaced 1330 fillers which had DRC vio's, with 2885 new fillers.

#### Fig. Filler info.

- Verify  $\rightarrow$  Verify Geometry... $\rightarrow$  OK
  - Does there any violation occur?

■ Ottomlon (132)			
■ Overiap (152)			
🗖 Overlap (132)	Layer	Cell	Master
	MO	IOFILLER_N_11	
	MO	IOFILLER_N_15	
	MO	IOFILLER_N_19	
	MO	IOFILLER_N_23	
	MO	IOFILLER_N_27	
	MO	IOFILLER_N_3	
	MO	IOFILLER_N_31	
	MO	IOFILLER_N_35	
	MO	IOFILLER_N_39	
	MO	IOFILLER_N_43	
	MO	IOFILLER_N_7	
	MO	IOFILLER_S_11	
	MO	IOFILLER_S_15	
	MO	IOFILLER_S_19	
	MO	IOFILLER_S_23	

- However, the overlap violations on IO filler can be ignored.
- Tools→Clear Violation→ Yes
- Verify  $\rightarrow$  Verify Connectivity...
  - Select "All"→OK
- Does there any violation occur?
- File  $\rightarrow$  Save Design
  - Type "CORE\_FILLER.enc"  $\rightarrow$  Save
- File  $\rightarrow$  Save  $\rightarrow$  Netlist...  $\rightarrow$  CHIP\_FINAL.v
- Timing  $\rightarrow$  Calculate Delay...  $\rightarrow$  CHIP\_FINAL.sdf
  - If it is the first time to calculate delay, you need to extract RC parameters first.
  - Timing → Extract RC... → OK.
- File  $\rightarrow$  Save  $\rightarrow$  DEF...



- Select "Save Scan"
- CHIP\_FINAL.def
- OK
- unix> chmod 755 addbonding\_v3.6.pl ◀┛
- unix>/usr/bin/perl addbonding\_v3.6.pl CHIP\_FINAL.def
- encounter> source addbond.cmd
- File  $\rightarrow$  Save Design
  - Type "FINISH.enc"  $\rightarrow$  Save
- Options  $\rightarrow$  Set Mode  $\rightarrow$  Mode Setup...
  - Select "StreamOut" tag
  - Un-select the "Virtual Connection $\rightarrow$ OK"
- File → Save → GDS... → CPU\_CHIP.gds → OK (i.e. Map File: streamOut.map)
- Design  $\rightarrow$  Exit  $\rightarrow$  Yes



Fig. Bonding PADs