The new IIT standard cell library

What's New?

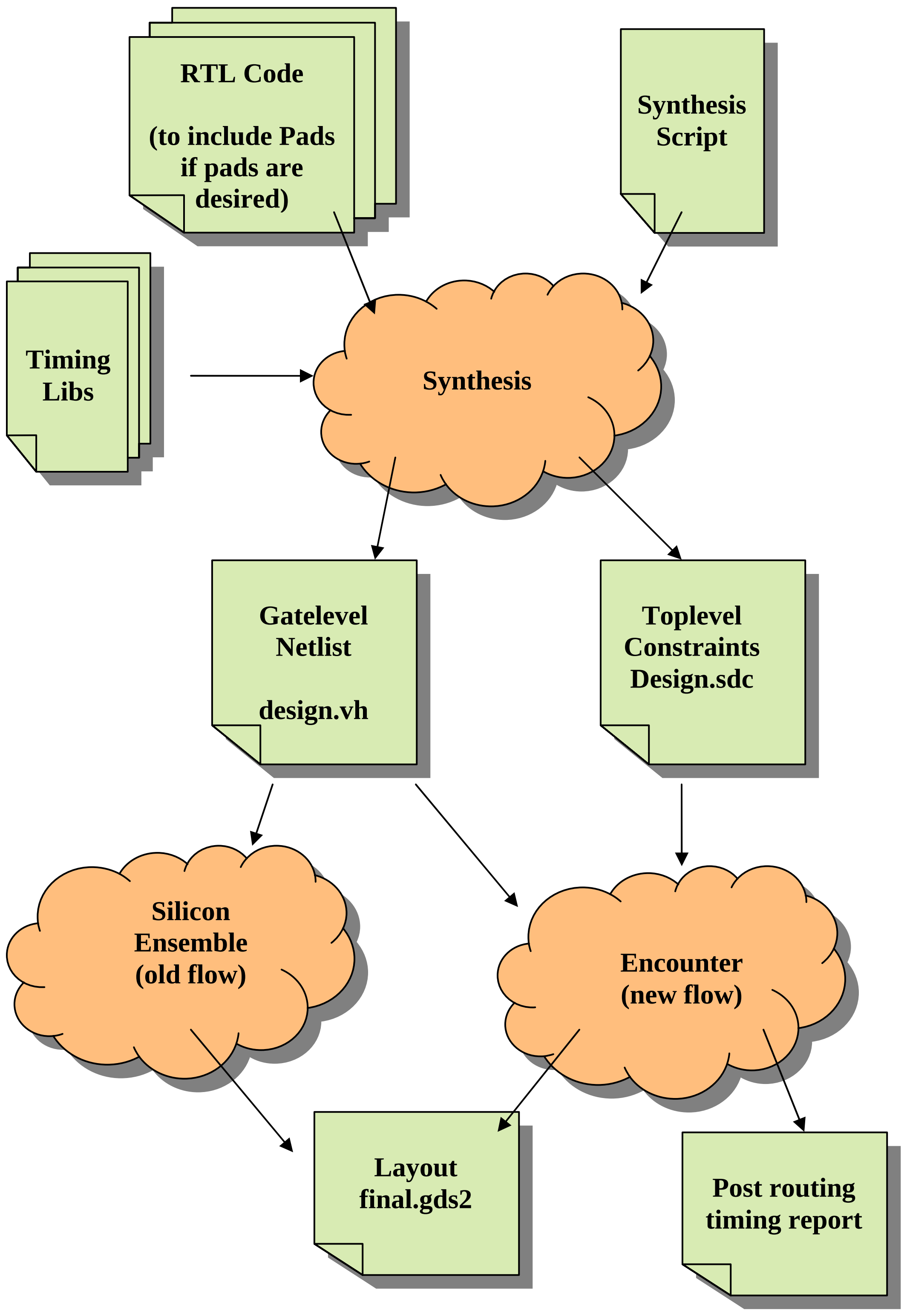
- Support for AMI 0.35um library
- Renamed ami06 to ami05 to be more consistent
- New cells: Asynchronous Set/Reset Flip-Flop, Latch, Clock buffers
- Support for stacked vias (for Virtuoso, not Magic)
- AMI 0.5 has lef files for both stacked vias and non-stacked vias (429 flow)
- LEF and TLF files now support Cadence Encounter
- All libraries now include Cadence Schematics
- Added metal6 to TSMC 0.18um library
- Added layer resistance and capacitance to all libraries
- Added supply and filler pads to all timing libraries
- Fixed AMI 0.5um pad flow to route to VDD/GND pads with full metal width
- New flow scripts for Encounter
- New flow scripts for Cadence BuildGates Synthesis
- Encounter flow offers
  - timing-driven placement and routing
  - static timing analysis (replaces Primetime)
  - clock tree synthesis
  - in-place-optimization and netlist modifications
  - Geometry and Connection Verification (DRC)
- Enhanced Synthesis scripts
  - All modifications in the script header
  - Script body remains constant
  - Automatic generation of timing constraints
  - Output of toplevel constraints for Encounter

Where are the files located?

All files are still in /import/cad2/iit_stdcells
The old library is moved to /import/cad2/iit_stdcells_preDFFSR

Library	Path	Comments
AMI 0.5um	./ami06/main	- Default LEF file to exclude stacked vias - Replace with iit05_stdcells_pads.stacked.lef to get stacked vias - Pad synthesis no longer supported (insert padframe in RTL code)
AMI 0.35um	./ami035/main	- As of now, library based on TSMC 0.35um - Uses stacked vias, Magic not supported
TSMC 0.25um	./tsmc025/main	- Uses stacked vias, Magic not supported
TSMC 0.18um	./tsmc018/main	- Now uses 6 metal layers - Uses stacked vias, Magic not supported

What is the new Design Flow


RTL Code
(to include Pads if pads are desired)
Synthesis Script
Timing Libs
Synthesis
Gatelevel Netlist
design.vh
Toplevel Constraints Design.sdc
Silicon Ensemble (old flow)
Encounter (new flow)
Layout final.gds2
Post routing timing report

How to setup a new design?

1. Simulate and test all Verilog RTL code
2. Make a run directory, e.g. “seultra” or “encounter”
3. Copy templates into that directory, e.g.
   - cp /import/cad2/iit_stdcells/tsmc018/main/* ./encounter
4. If the RTL code contains pads, include iitXXX_stdcells.v in the verilog command line

How to run Synthesis with Synopsys Design Compiler

1. Customize compile_dc.scr
   a. Specify the names of all Verilog files
   b. Specify the toplevel names
   c. Specify the target clock frequency (in MHz)
   d. Specify the name of the clock pin
   e. Specify the input and output delays (or accept the default value of 1ns)
2. Run “dc_shell –f compile_dc.scr
3. The results
   a. The gate-level netlist (.vh)
   b. The toplevlel timing constraints (.sdc)
   c. Timing report: timing.rep
   d. Area report: cell.rep
   e. Power report: power.rep

How to run Synthesis with Cadence BuildGates

4. Customize compile_bgx.scr
   a. Specify the names of all Verilog files
   b. Specify the toplevel names
   c. Specify the target clock frequency (in MHz)
   d. Specify the name of the clock pin
   e. Specify the input and output delays (or accept the default value of 1ns)
5. Run “bgx_shell –f compile_bgx.scr
6. The results
   a. The gate-level netlist (.vh)
   b. The toplevlel timing constraints (.sdc)
   c. Timing report: timing.rep
   d. Area report: cell.rep
   e. Power report: power.rep
   f.

How to place&route with Silicon Ensemble (old flow)

1. Customize seultra.scr
2. Run “se_shell –f seultra.scr”
3. The results
   a. Layout file “final.gds2”

How to place&route with Encounter (new flow)

4. Customize encounter.conf
    a. Enter the toplevel name at the top
    b. Uncomment the “encounter.io” line if there are pads
5. Run “encounter –init encounter.tcl”
6. Either enter “win” to open the GUI with the chip or “exit” to quit
7. The results
    a. Layout file “final.gds2”
    b. Final netlist “.v” (Encounter modifies the netlist to add buffers)
    c. Final timing report: “timing.rep.5.final”
    d. Other timing reports from various place&route stages: “timing.rep*”

How to create a Magic Layout (only AMI 0.5um)

1. If used “seultra”: run “iitcells_se2magic”
2. If used “encounter: run “iitcells_enc2magic”

How to create a Cadence Layout

1. If used “seultra”: run “iitcells_se2icfb”
2. If used “encounter: run “iitcells_enc2icfb”

How to run a design with pads

1. Insert pads in the RTL code of your design (see the example designs)
2. Simulate the RTL code to make sure it still works
3. Run Synthesis as usual
4. For Silicon Ensemble: no change necessary
5. For Encounter
    a. Uncomment the “encounter.io” line in “encounter.conf”
    b. Make sure the name in “encounter.io” match the pads in the RTL code
    c. You can modify the “encounter.io” file to specify pads locations
    d. Now run Encounter as usual

What if I am using AMI 0.5um but I want stacked vias

1. Stacked vias make routing much easier because
    a. The router can change layers without “stair cases”
    b. The router can connect to the cell anywhere, not just on the pin
    c. Everything except the pin used to be an obstruction, even though it was still part of the net and valid for connection
2. There is a LEF file “iit05_stdcells_pads.stacks.lef
3. Edit “seultra.scr” or “encounter.conf” to use that LEF file instead
4. Run as usual
5. Final layout not valid for Magic