Flujo de Diseno Analógico basado en Software Libre

Leandro Marsó

Febrero 2011

ClariPhy Argentina S.A.

Outline

- Introducción
- Flujo de diseño detallado
- Herramientas disponibles
- Conclusiones
  - Procesos de Fabricación y Herramientas disponibles
  - EDA and Workflow Setup
  - Propuestas
  - Trabajo futuro

ClariPhy Argentina S.A.

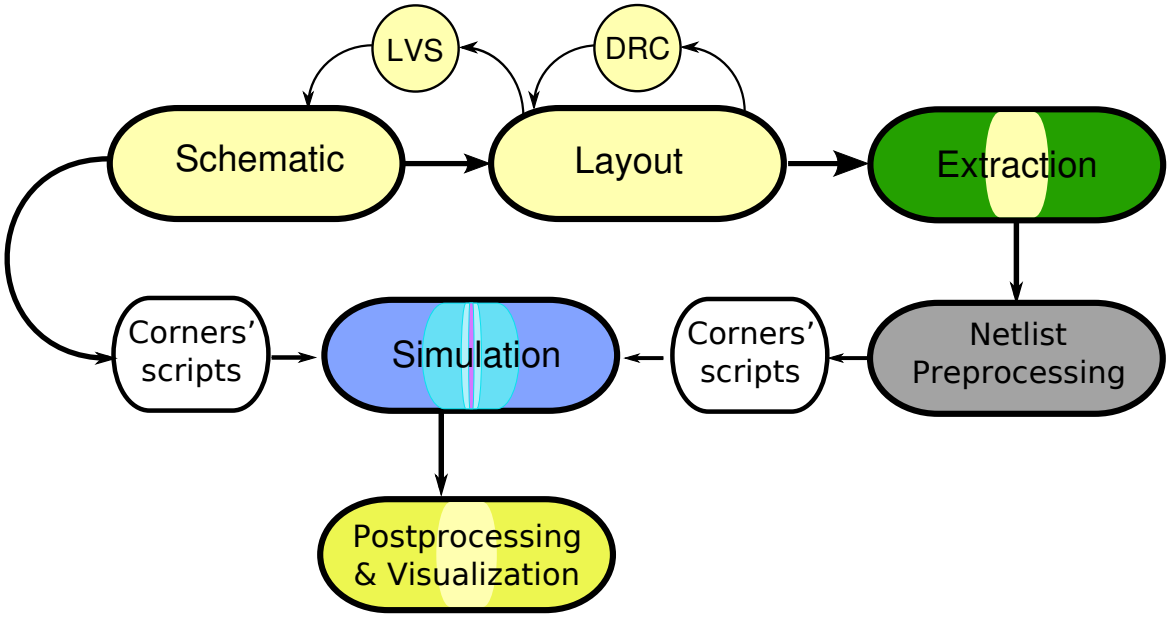
Introducción

Introducción

- Un flujo de diseño está compuesto por un programa específico para cada tarea, e interfaces y formatos comunes para pasar de una etapa a la otra del dise ño.
- Al mismo tiempo, según avanzan los procesos de fabricación, también cambian las herramientas.
- Las herramientas fueron estudiadas por separado teniendo en cuenta la posibilidad de integración con otras, libres o no, necesario cuando se llega al punto de no encontrar herramientas libres.
- Teniendo encuenta estos items, y mirando los testbenches de los diseñadores fué encarado este estudio

Flujo de diseño detallado

Flujo analógico:

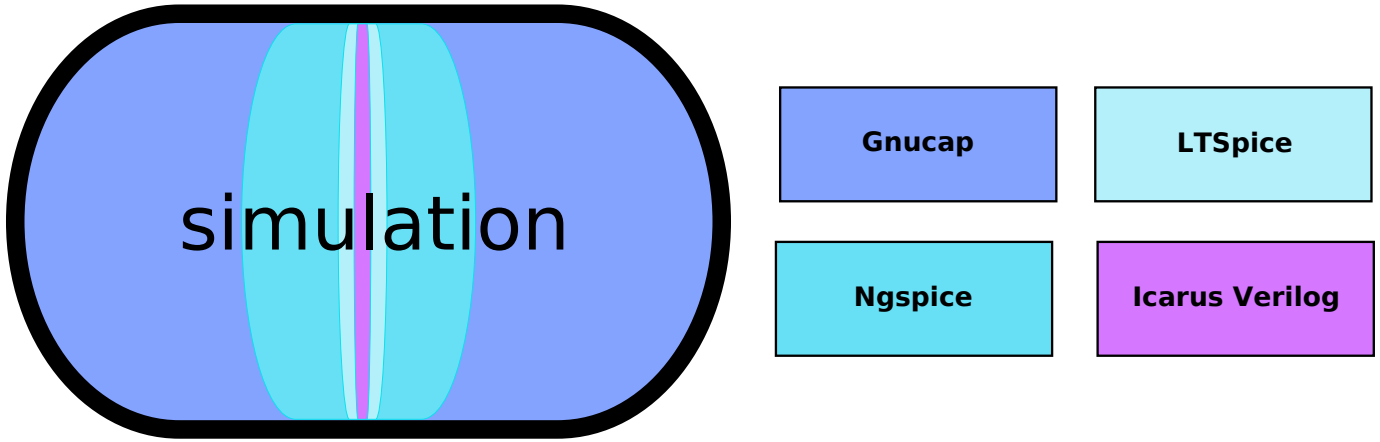


Tools:

Electric VLSI	**Gnucap**	**Icarus Verilog**	**Matplotlib, Numpy Octave/Matlab**	**Pyspice**
Ngspice	**Magic VLSI**:	**LTSpice**	**NYE**	

Flujo de diseño detallado

Simulación:



Herramientas necesarias:

- Analog Simulation Engine
- Digital Simulation Engine
- Mixed Mode Simulation Engine
- Simulation Models: built in on simulation engine or compiled using mot-adms

Analog Simulation Engine

Gnucap:

- **Pros:** Fast simulator. Fully scriptable. Scales better than ngspice and LTSpice. Usefull plug-ins system to use other netlist syntax (such as spectre, spice3f5, verilog-ams) or compile new simulation models without compiling the whole simulator. There is a fork which implements icarus verilog cosimulation, mot-adms model compiler, among other minor enhancements.
- Due to its plug-in system, it inherits some feature improvements done on other free simulators such as ngspice. It also allows easier development of features.
- Digital Verilog cosimulation is possible replacing PWL device for a verilog circuit.
- Icarus Verilog-AMS simulation will be available for cosimulation when implemented. Not there yet.
- **Cons:** Lacks some simulation analisys such as noise (which is being developed now, not yet available for beta testing) or PSS.
- Verilog-AMS not yet available, testbenches are more script oriented handcrafted. Coding skills needed.

Analog Simulation Engine

Ngspice:

- Merges 3 different simulators: Berkeley spice3f5, Cider and Xspice which gives us a mixed-signal/mixed-level simulator.
- **Pros:** Widely used and support several analysis and compact simulation models, mot-adms model compiler can be used for verilog-ams models. Output can be retrieved by the electric waveform viewer, or using its built-in viewer.
- **Cons:** It is slower than gnucap and needs to be recompiled when a new simulation model is added.

LTSpice:

- The same ngspice pros and cons' applies to LTSpice. It is not free software, but its license allows it to be used for almost all industrial scenario.

Schematic and Layout

Electric VLSI:

Schematic, Layout, DRC, LVS, spice netlist

- **Pros:** Actively maintained: fast bugs corrections, periodic releases with new features.
- Netlisting generator targets several simulators
- Waveform viewer accepts several simulator data output
- TSMC PDK avaiable for 180nm, 90nm and 45nm and even can export cells to SKILL
- Can read DRC error output from Cadence and Menotor tools
- **Cons:** Schematic lacks finger/multiplier params, Layout is done in a different way, but easy to learn

Schematic and Layout

Magic VLSI:

We use only **parasitic extraction** and netlist generation feature of this layout tool

- **Pros:** More accurate than Electric, widely used for old processes (0.25um and above)
- **Cons:** Need to write another techfile for this tool
- Not acceptable for nanometer processes.Schematic, Layout, DRC, LVS, spice netlist

Postprocesamiento y visualización

Cadence calculator replacement:
Numpy library can be used as a tool to postprocess simulation results (ascii data in gnucap and spice raw format in ngspice) It is a matlab replacement also.
Matplolib is a library that can be used to view the simulation results. Electric waveform viewer can be used also (not for gnucap results)

- Pros: Very flexible tools, known as a easy replacement for matlab
- Cons: This methodology can be slower for designers that are used to Cadence tools. CAD engineer or the designer should prepare a set of common task before start designing.

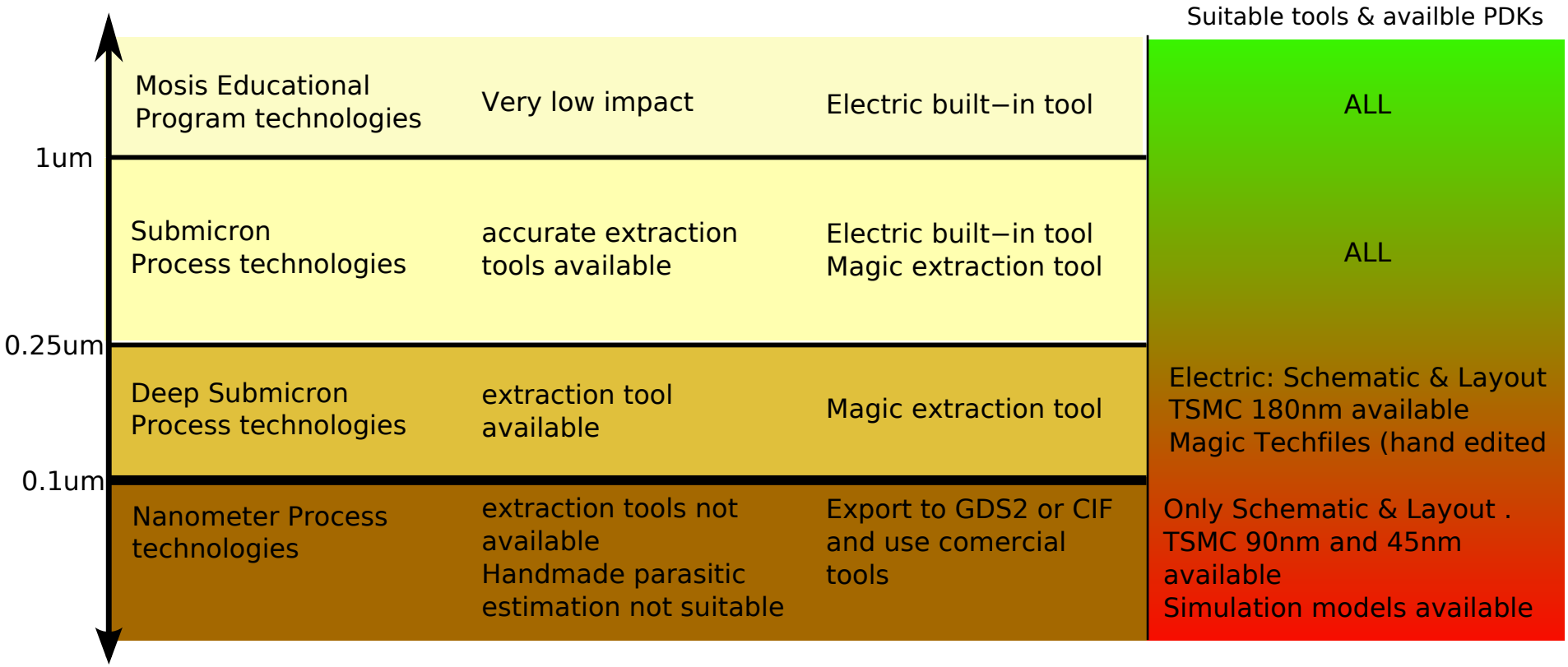
Netlist preprocessing

pyspice: This tool can do some reduction on parallel transistor and capacitors to help low down number of nodes on post extraction netlist, to reduce simulation time.

Conclusions

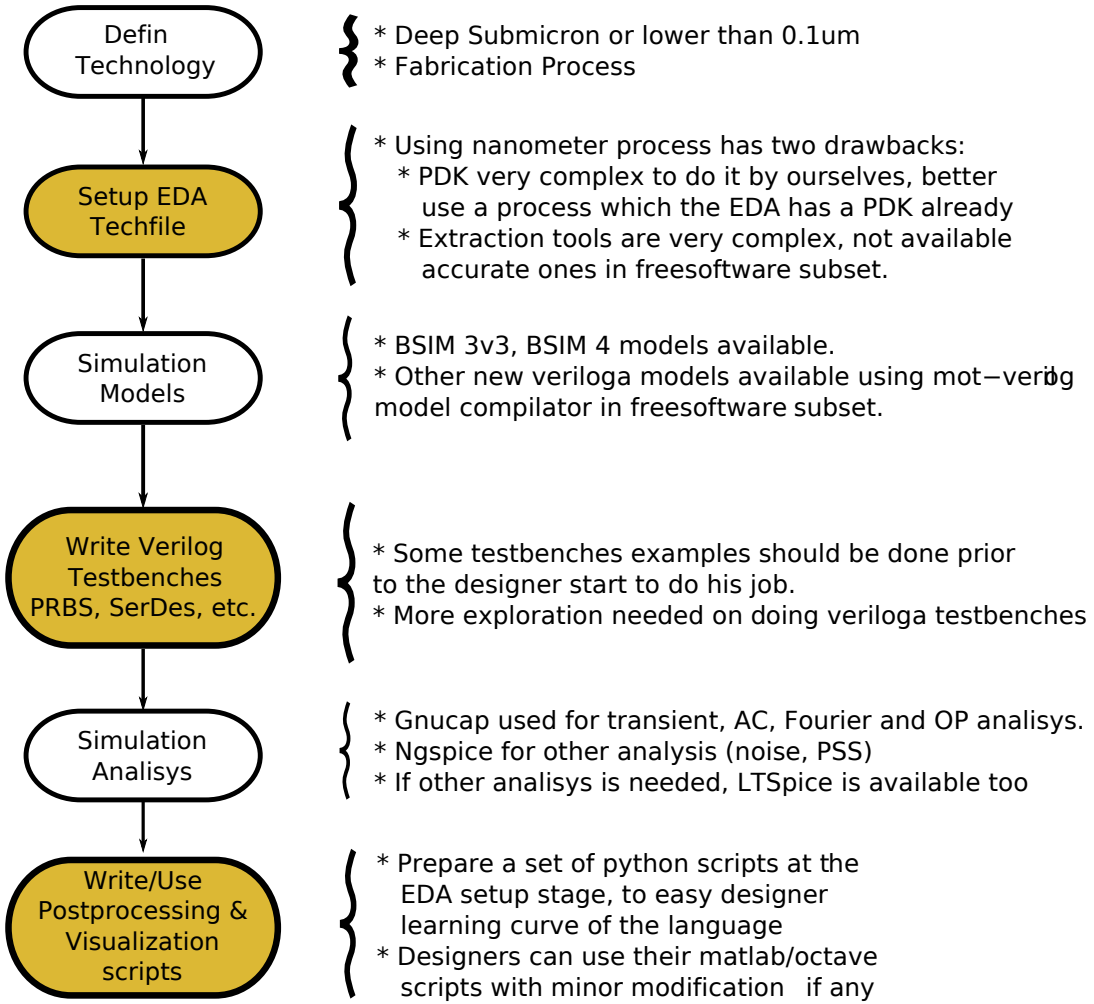
Summary: Technology process versus available tools

Parasitic Extraction tools for Interconection Noise & Crosstalk Impact



Conclusions

EDA and workflow setup



Conclusions

- Before start choosing a tool, process technology must be defined.
- Tools can be replaced one by one, not necesary to forge an entirely custom workflow.
- Tasks more likely to being replaced Cadence workflow is schematic and layout design, depending on the technology process.
- Most of these tools are actively developed, so it is recommended (if not compulsory) compiling and coding skills.
- To help ease the change, well documented examples for every task should be available (expecially for numpy, matplotlib and handcrafted testbenches scripts)
- Previous survey on asitic turns to be very usefull now.

Conclusions

Future work

- Test simulator with large nets to be sure that we don’t hit a tool limit when doing production everday work. We should be able to use extracted netlist from QRC Cadence of a real block designed in CASA.
- Try to export data to allow visibility of CASA blocks for INC database.