Signal Integrity Methodology on 300 MHz SoC design using ALF libraries and tools

Wolfgang Roethig, Ramakrishna Nibhanupudi, Arun Balakrishnan, Gopal Dandu
NEC Electronics

Steven McCormick, Vinay Srinivas, Robert Macys, Dhiraj Sogani, Kevin Walsh
Sequence Design

Abstract

This paper presents a new design methodology to address signal integrity issues in ASIC-style designs, using innovative EDA tools which concurrently analyze and optimize timing, crosstalk, noise, electromigration and hot electron constraints, based on accurately characterized cell libraries in the Advanced Library Format (ALF) [1], [2]. This methodology is demonstrated on a 3.5 million gate, 333 MHz design in 0.13μ technology.

I. Problem statement

A. *Point tool solutions for signal integrity*

Traditionally, ASIC design has been almost exclusively focused on timing closure. Heuristic rules and guard bands were used to protect the design against adverse effects on signal integrity, such as noise, electromigration. However, it is well known to the industry today, that such guard bands are no longer sufficient. They prevented the efficient use of technology to the point that clock frequencies above 200 MHz could only be reached with difficulty on ASIC-style designs.

Therefore the guard bands have been eliminated. Instead, point tools have been introduced into the design flow to check for signal integrity, namely for crosstalk-induced noise, electromigration (EM) and hot electron (HE) effects. In addition, static timing analysis has been enhanced to consider crosstalk-induced delay.

Figure 1 illustrates a design flow, where point tools for signal integrity analysis are applied after place & route. Eventually, the tools find signal integrity violations and generate scripts for buffer insertion/deletion/resizing, for placement changes, or for routing changes.

The drawback is that the place & route tools are not completely controlled by these scripts. For example, "*INSERT BUFFER*" does not specify the exact size and location of the buffer. "*REROUTE NET WITH NONDEFAULT RULE*" does not specify where the new route will be. Therefore, after applying the repair scripts, signal integrity rules have to be checked again.

Another drawback is that the signal integrity checking tools do not know the timing. The directives for signal integrity repair could eventually conflict with meeting timing constraints. Therefore a one-pass analysis and repair can not be guaranteed by this flow.
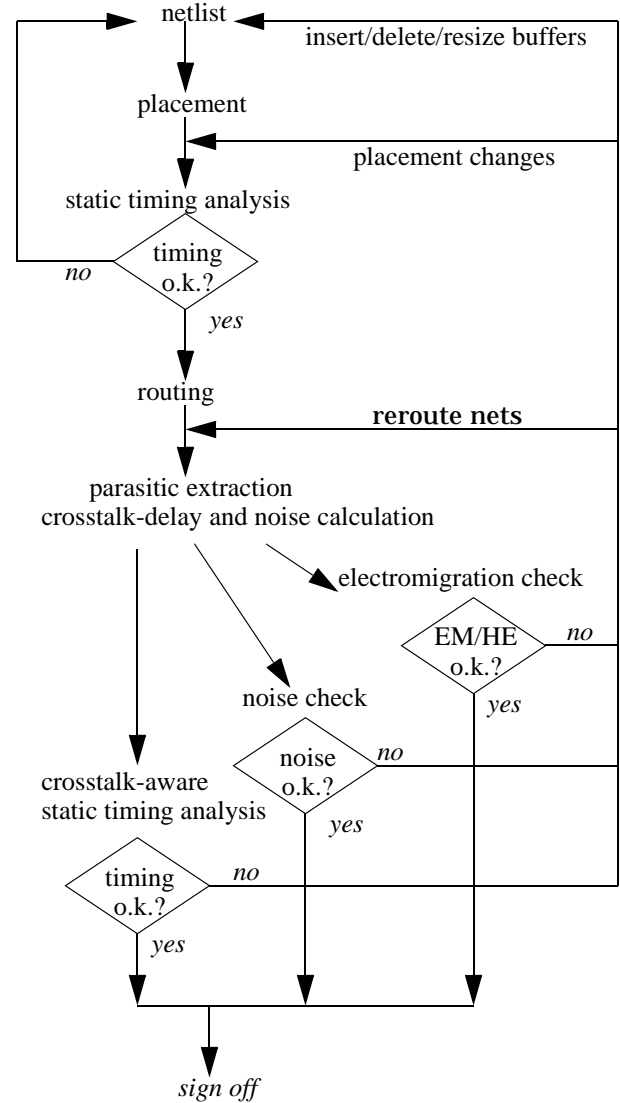

Figure 1: Signal integrity design flow with point tools

B. *Crosstalk-aware static timing analysis*

Crosstalk-aware timing analysis itself is done by combining delay calculation and static timing analysis iteratively.

This is due to a chicken-end-egg situation in crosstalk-induced delay. To calculate crosstalk-induced delay, the arrival times of aggressor and victim must be known. To calculate arrival times, delay must be known. Therefore pessimistic arrival time windows are calculated in a first pass. These time windows are then used for crosstalk-induced delay calculation. Then the time windows are re-calculated. This process is repeated until convergence, as shown in figure 2.

SPEF file

↓

delay calculation
without crosstalk

↓

SDF file

↓

pessimistic
time window
calculation

↓

aggressor/victim
time overlap?

yes → crosstalk-induced
extra delay

↓

incremental SDF file

↓

static timing analysis

no → less pessimistic
time window
calculation

timing
o.k.?   *no* →   time windows
accurate?   *no* →

yes   *yes*

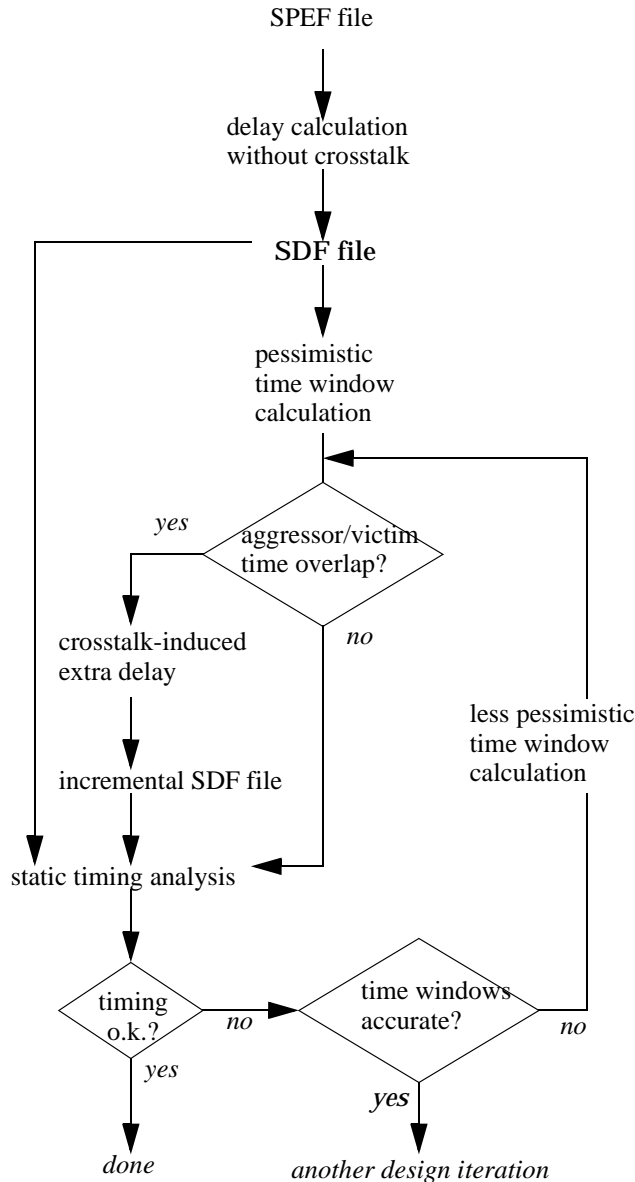↓   ↓

done   *another design iteration*

Figure 2: Details on crosstalk-aware static timing analysis

The inconvenience of this method is obvious. Large SDF files are passed between delay calculation and static timing analysis. Crosstalk-aware timing analysis amplifies the run time and disk space overhead of the SDF interface, because it is now repeated in a loop.

To avoid the SDF interface, STA tools could be used "as is" with their native delay calculator and library. However, some ASIC vendors insist on their proprietary delay calcula-

tion tools and promote the usage the IEEE 1481 standard for delay calculation (DCM). DCM prescribes a binary interface between delay calculator and timing analyzer within a single executable [3]. However, the native delay calculators of commercial EDA tools as well as the IEEE 1481 standard fall short in supporting crosstalk and other signal integrity issues.

Therefore the proposed solution in this paper is to introduce tools with support for ALF, which can describe timing, noise, EM/HE models for comprehensive analysis and optimization.

II. Modeling with ALF

This section outlines the pertinent features of ALF for accurate timing and signal integrity modeling.

A. Timing calculation

For technologies of 0.25μ and smaller, the shape of the signal waveform plays a significant role in describing the timing characteristics. Both waveform shape and susceptibility to noise are modeled by a driver resistance Rd. Figure 3 illustrates the dual role of the driver resistance model.

aggressor waveform   aggressor waveform
@ driving point   @ coupling point

Rd (aggressor)

Rd (victim)

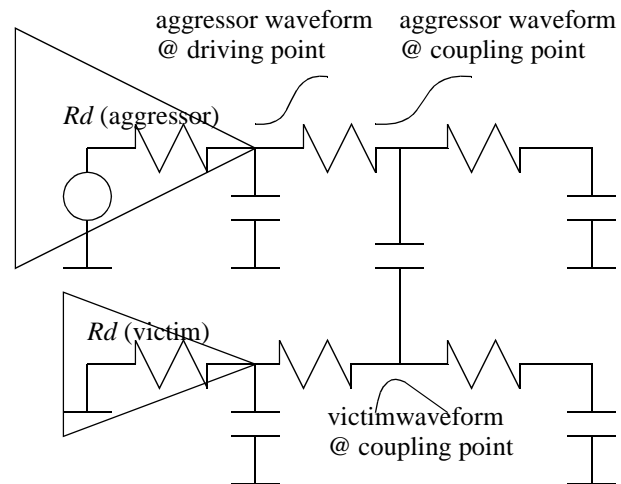victimwaveform
@ coupling point

Figure 3: Driver resistance affects timing and noise waveforms

A switching driver (aggressor) can be modeled as an ideal voltage source producing a ramp, in series with a resistance. The waveform at the driver output depends on the effective capacitance [4] which itself depends on the interconnect RC network. The shape of the signal waveform changes along the interconnect. Eventually, the aggressor signal appears as noise on another net through capacitive coupling. The magnitude of the noise depends on the driver resistance and the interconnect RC network on the victim net. Both aggressor and victim driver resistance are state-dependent and must therefore be characterized in the library.

Eventually, multiple signals on coupled nets switch simultaneously and cause mutual waveform distortion. The use of driver resistance models allows to calculate the resulting waveforms using linear circuit analysis.

The resulting waveforms depend also on the alignment of the original waveforms. If pessimistic time windows are used, the waveform alignment is not known with much certainty.

Therefore, the concept of activity windows is introduced. The idea is to calculate multiple narrow time windows of possible switching activity per clock cycle in order to decide with more certainty, whether aggressor and victim waveforms will overlap or not. This idea is illustrated in figure 4.
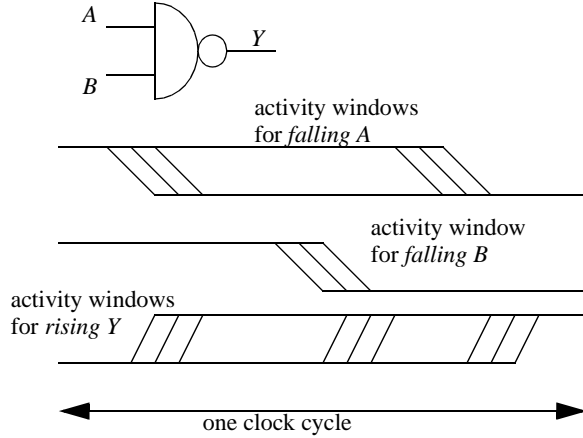

Figure 4: Accurate time window representation

Signal *A* has two activity windows within a clock cycle, one very early, the other very late. Signal *B* has one activity window in the middle of the clock cycle. As a consequence, signal *Y*, the output of a cell with inputs *A* and *B*, has three activity windows within the clock cycle.

Each activity window is associated with bounds for output arrival time, slewrate and driver resistance. These parameters are calculated from input arrival times and slewrates, using timing models in the ALF library, as shown in figure 5.


```
VECTOR (01 A -> 10 Y) {
  DELAY { FROM { PIN=A; } TO { PIN=Y; }
    HEADER {
      CAPACITANCE { PIN=Y; TABLE { ... } }
      SLEWRATE { PIN=A; TABLE { ... } }
    } TABLE { ... }
  }
  SLEWRATE { PIN=Y;
    HEADER { ... } TABLE { ... }
  }
  RESISTANCE { PIN=Y;
    HEADER { ... } TABLE { ... }
  }
}
```
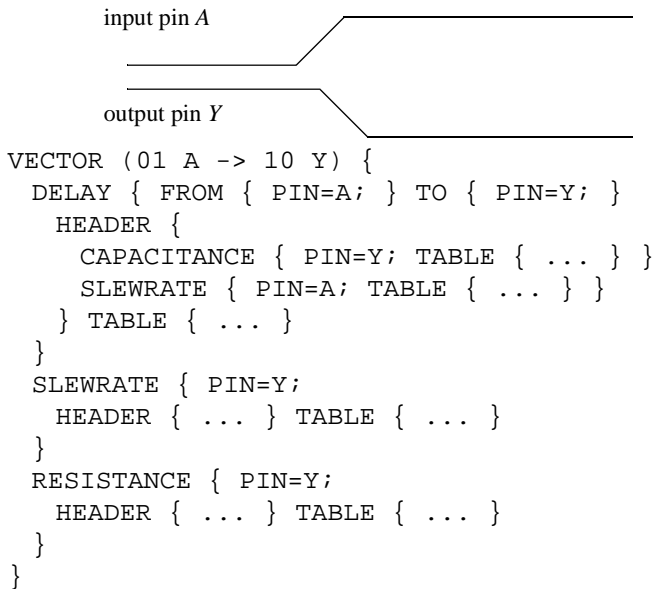
Figure 5: Symbolic waveform and ALF model for timing

The particularity of an ALF model is its association with a symbolic waveform, represented as a VECTOR. This allows a 1-to-1 correspondence between characterization specification and resulting timing model. In the example of figure 5, a rising transition on pin *A* followed by a falling transition on pin *Y* is associated with DELAY, SLEWRATE and transient RESISTANCE measurements. The characterization measurements can be represented as TABLE data or as EQUATION.

B. Noise calculation

The driver resistance for steady state is used for noise calculation on a quiet victim driver. In addition, a noise margin on a victim receiver must be provided in order to decide whether the noise can be tolerated.

A criterion for maximum allowed noise peak at the output defines the noise margin at the input. The noise peak at the output depends not only on the noise peak at the input but also on the input pulsewidth and the effective output load capacitance. Narrow noise pulses get filtered. The larger the load capacitance, the stronger the filter. Therefore a dynamic noise margin, also called noise rejection, can be described as illustrated in figure 6. For a long input pulses, the dynamic noise margin equals the static noise margin.
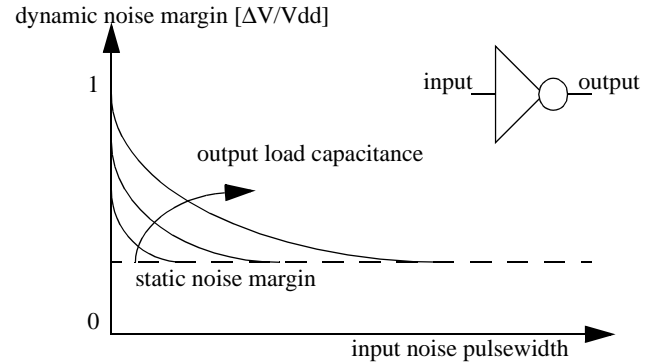

Figure 6: Static and dynamic noise margin

Noise rejection can be described by an ALF VECTOR with an associated NOISE_MARGIN, as shown in figure 7.


```
VECTOR (0* A -> *0 A -> 11 Y) {
  NOISE_MARGIN { PIN=A;
    HEADER {
      PULSEWIDTH { PIN=A; TABLE { ... } }
      CAPACITANCE { PIN=Y; TABLE { ... } }
    } TABLE { ... }
  }
}
```
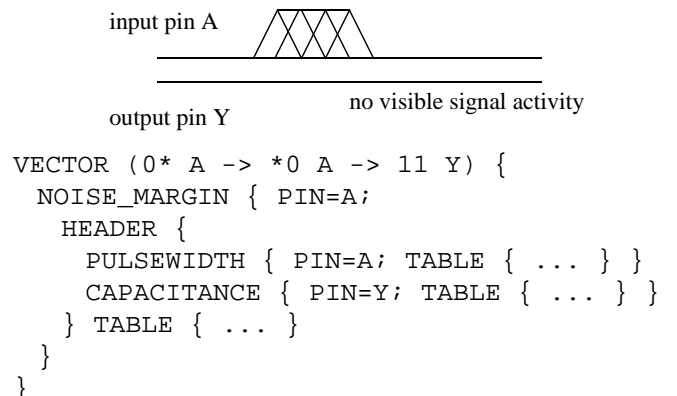
Figure 7: Symbolic waveform, ALF model for noise rejection

The definition for noise margin implies, that the noise activity at the output due to the noise at the input is negligible

For combinatorial cells, noise activity at the output can be tolerated, as long as it does not corrupt the data of a memory element, a flip-flop or a latch.

Therefore, noise propagation instead of noise rejection can be described in the library, as illustrated in figure 8.
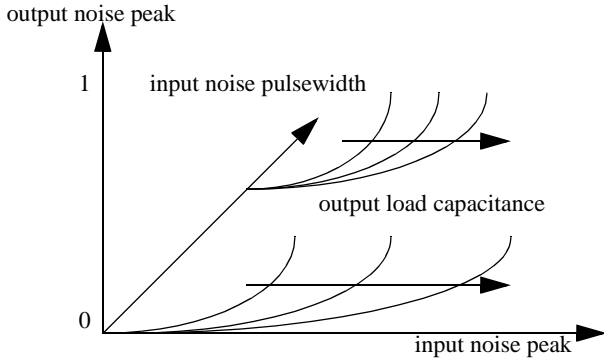

Figure 8: Input-to-output noise propagation

Figure 9 shows the corresponding ALF model, again using the isomorphism between a VECTOR and a symbolic timing diagram. The noise peak, pulsewidth and delay at the output depend on load capacitance as well as on noise peak and pulsewidth at the input.


```
VECTOR (0* A -> *0 A <&> 1* Y -> *1 Y) {
  NOISE { PIN=Y;
    HEADER {
      NOISE { PIN=A; TABLE { ... } }
      PULSEWIDTH { PIN=A; TABLE { ... } }
      CAPACITANCE { PIN=Y; TABLE { ... } }
    } TABLE { ... }
  }
  PULSEWIDTH { PIN=Y;
    HEADER { ... } TABLE { ... }
  }
  DELAY { FROM { PIN=A; } TO { PIN=Y; }
    HEADER { ... } TABLE { ... }
  }
}
```
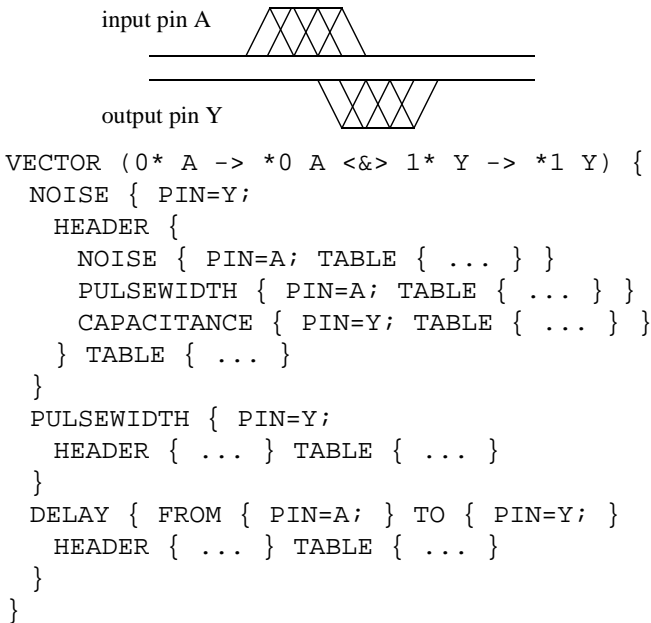
Figure 9: Symbolic waveform, ALF model for noise propagation

This section showed that noise analysis is a natural extension to timing analysis. This extension requires additional characterization data that can be well-described in ALF. Noise waveforms are a derivative of signal waveforms. The latter are described by SLEWRATE, the former are described by PULSEWIDTH and NOISE.

C. Electromigration and hot electron calculation

Electromigration occurs inside cells as well as on interconnect structures. Electromigration is due to high current density which eventually causes wires and contacts to break. The structures inside the cell tend to break first, especially the contacts at the driver output.

Another damage inside cells occurs on NMOS transistors, due to the hot electron effect. This effect manifests itself by accumulation of trapped carriers in the gate oxide, leading to threshold changes and performance degradation.

Both effects can be evaluated by transistor-level transient current and voltage simulations. However, this is not feasible for large circuits. Therefore a cell-level abstraction model is needed.

Figure 10 illustrates paths for electrical currents within a complex cell. Eventually, transistors or contacts get damaged, if they are exposed to the current for too long.
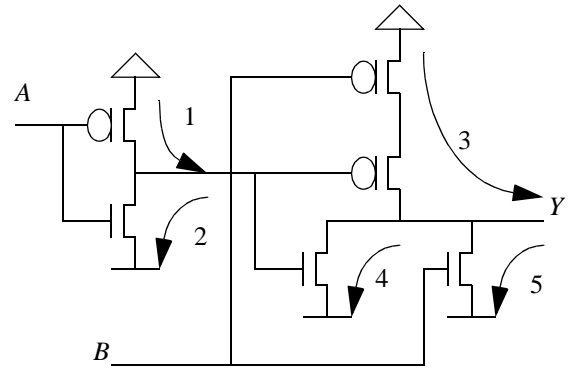

```
VECTOR (10 A) // covers path 1
VECTOR (01 A) // covers path 2
VECTOR (01 Y) // covers path 3
VECTOR (10 A -> 10 Y) // covers path 4
VECTOR (01 B -> 10 Y) // covers path 5

VECTOR (10 A -> 10 Y) {
  LIMIT { FREQUENCY { MAX {
    HEADER {
      CAPACITANCE { PIN=Y; TABLE { ... } }
      SLEWRATE { PIN=A; TABLE { ... } }
    } TABLE { ... }
  } } }
}
```

Figure 10: Electrical current paths, ALF model for EM / HE

The abstraction consists of an ALF VECTOR defining the activation stimulus for a path. Associated with the vector is an upper limit for tolerable activation frequency of the vector. This frequency limit is an abstraction of the tolerable electromigration or hot electron damage, which depends on either input slewrate and output load or both.

Since ALF vectors can contain temporal and logical dependencies, it is possible to represent electromigration and

hot electron constraints affecting internal structures of a complex cell by vector frequency limits describing events and states observable at the boundary of the cell.

III. The new Design Methodology

A. Concurrent analysis-driven optimization

In the previously mentioned point tool approach, noise, electromigration and hot electron effects are checked individually and eventually fixed through manually or automatically generated scripts prescribing incremental layout changes. The point tools are mutually unaware of other effects, therefore the fixing of an electromigration violation may cause a timing violation, for instance.

In contrast, our new methodology applies a layout optimization tool which is conscious of all signal integrity effects and implements only design changes with minimal disturbance. As a result, the number of design iterations is greatly reduced. The tool applied in this methodology performs crosstalk-aware static timing analysis considering multiple activity windows within a clock cycle. No intermediate SDF files are necessary. This tool can immediately take advantage of the ALF library which contains models for timing, noise, electromigration and hot electron constraints.

Design transformations for optimization are driven by timing constraints and available time slack. Before implementing a particular design transformation, the tool checks whether this design transformation would violate timing, noise or electromigration constraints.

Figure 11 shows the design flow with the integrated analysis and optimization tool driven by the ALF library.
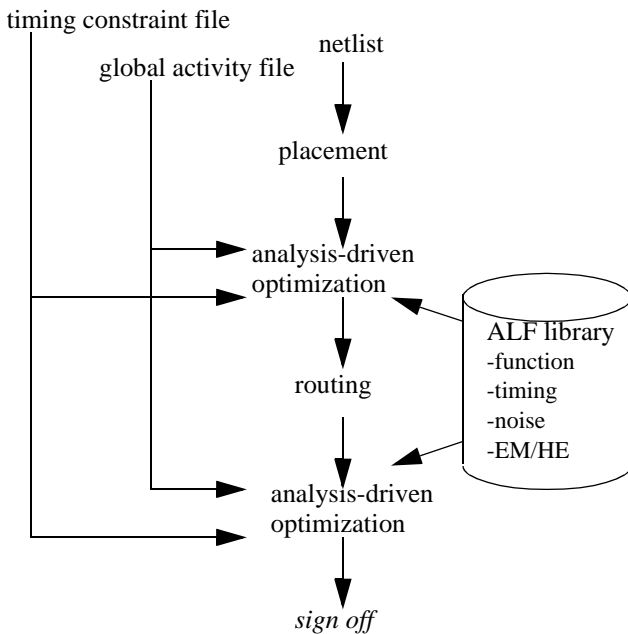
As with any design flow, external system-level timing constraints must be provided. Noise and EM/HE constraints are provided as noise margins and vector-frequency limits, respectively, in the library. In addition, a global activity file is provided for EM/HE analysis. This file contains estimated or simulated vector-frequencies for each cell instance within the design, which have to be checked against the vector frequency limits in the library. For a given vector frequency, the slew- and load-dependent frequency limit translates into a slew-dependent load limit. Since the design transformations done by the optimization tool are restricted to insertion, removal, or substitution of local buffers and cells, the global activity file for the initial netlist can basically be used throughout the flow.

B. Accuracy of the ALF library models

The success of the design flow relies on sign-off accurate libraries. A suite of benchmarks with normative SPICE results has been applied to qualify the ALF library within the context of its usage by the tool. For example, the ALF timing library alone, by virtue of including more precise data, yields significantly better accuracy than a conventional timing library.

Figure 12 and 13 show scatter plots and error plots of delay calculation versus SPICE using the conventional timing library and the ALF library, respectively. It must be noted that the data in the DELAY and SLEWRATE tables in both libraries are exactly the same. The accuracy improvements are due to the inclusion of precharacterized driver resistance data, delay and slewrate measurement reference points, which are used by the tool for better waveform modeling.

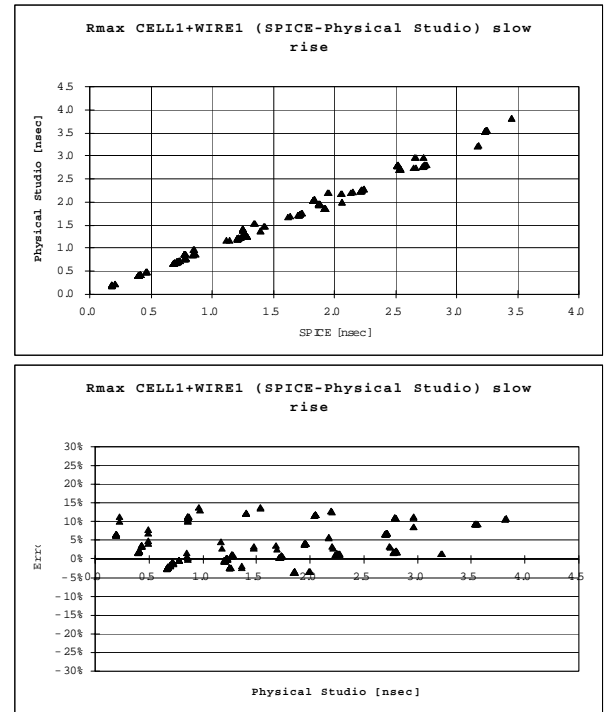Figure 11: Signal integrity design flow with integrated analysis and optimization

Figure 12: Delay calculation vs. SPICE with conventional library
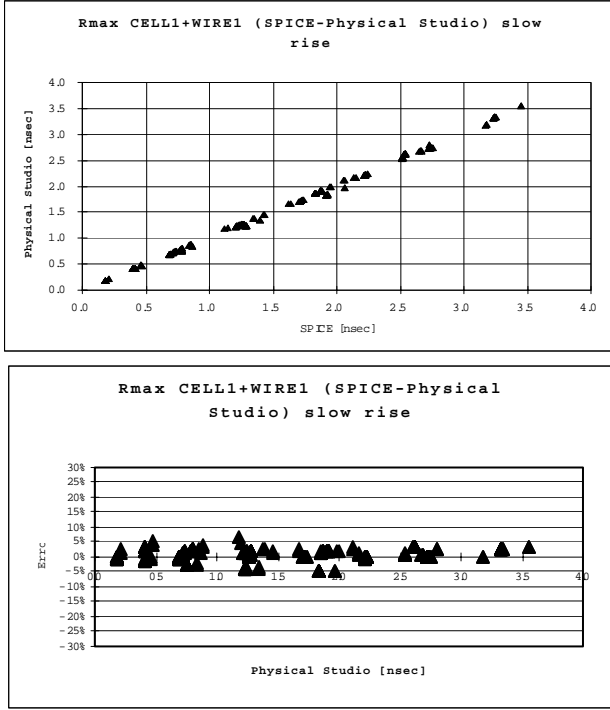
Figure 13: Delay calculation vs. SPICE with ALF library

Table I summarizes the comparison. The average error with ALF is 0.5%, compared to 3.9% without ALF. The standard deviation is +/- 2% compared to +/- 5%. The difference between min and max error is 10.9% compared to 17.4%.

TABLE I: Delay calculation error versus SPICE

error criterion	without ALF	with ALF
average	+3.9%	+0.5%
standard deviation	+/-5%	+/-2.2%
min	-3.4%	-4.5%
max	+14%	+6.4%

Similar statistics have been established for the noise and electromigration data in the ALF library. However, there was no conventional library to improve upon.

C. Results

The timing and signal integrity optimization tool was applied on a large-scale design in 0.13μ technology. The main clock frequencies were 333 MHz and 167 MHz. More design information is summarized in table II below.

TABLE II: Design statistics

# instances	448K (before optimization) 467K (after optimization)
# macroblocks (RAM, core, IO, analog)	300
equivalent gate count	3.5M

This design was implemented on a 8.5mm*8.5mm die. The utilization was about 50%. The floorplan is shown in figure 14.

Figure 14: Floorplan of the design

The size of the design suggested a hierarchical implementation, however, flat layout was still feasible. Enough space around the macroblocks had to be provided to avoid routing violations. Hierarchical design would impose further restrictions for routing over blocks, spacing and routing channels between blocks, timing budgets for each block. Also, guard bands would be necessary in order to maintain accuracy in extraction, timing and signal integrity analysis. Therefore we attempted a flat implementation.

The design was routed on 5 metal layers, as shown in figure 15. The black rectangles are macroblocks which utilized all 5 layers or contained analog circuitry. Routing over these blocks was not allowed.
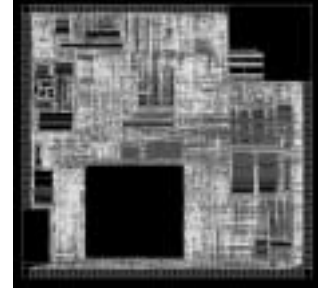

Figure 15: The placed & routed design

Table III below lists the tools used in the flow and their runtimes on SUN 450MHz workstations with 32 bit OS

TABLE III: Tools used in the flow

design step	tool	vendor	runtime
Floorplan	IC Wizard	Monterey Design Systems	2h
Initial placement	N.N.	N.N.	4h
Optimization	PhysicalStudio	Sequence Design	10h+12h
Routing	N.N.	N.N.	8h+8h
Extraction	Columbus Turbo	Sequence Design	8h

Table IV shows some details on the optimization results.

TABLE IV: Optimization results

Timing before optimization	-10.7 ns slack
# of timing violations fixed	8600
# of noise violations fixed	
# of max. load / slew violations fixed	24500
Timing after optimization	+0.2 ns slack

The results show that the optimization is feasible and efficient on a large-scale design.

IV. Conclusion

This paper explains the methods and issues with signal integrity in an ASIC-style design flow. Therefore a new design flow with a greater level of integration of tool functionality and library is proposed. Analysis-based optimization for timing, noise, electromigration and hot electron is accomplished in a single tool, using sign-off worthy ALF models. The principles for creating a comprehensive ALF technology library for timing, noise, electromigration and hot electron constraints are explained. Benchmark results show that the electrical performance predicted by ALF models is indeed accurate. A successful exercise of the flow on a 3.5M gate design testcase provides evidence for the readiness of the new methodology.

REFERENCES

[1]   "IEEE P1603 Standard for an Advanced Library Format describing Integrated Circuit Technology, Cells and Blocks", http://www.eda.org

[2]   W. Roethig "Coherent Functional, Electrical and Physical Modeling of IP blocks using ALF", *CICC,* 2001

[3]   "IEEE 1481-1999 Standard for Delay and Power Calculation System", http://www.eda.org

[4]   R. Macys, S. McCormick "A New Algorithm for Computing the Effective Capacitance in Deep-Submicron circuits", *CICC,* 1998