Crosstalk Aware Static Timing Analysis Environment

B. Franzini, C. Forzan

STMicroelectronics, v. C. Olivetti, 2 20041 Agrate B. (MI), ITALY

bruno.franzini@st.com, cristiano.forzan@st.com

ABSTRACT

Signals crosstalk due to coupling capacitances between adjacent interconnect lines is going to heavily affect the timing behaviour of the 0.18µm (and below) CMOS digital designs. The design timing verification step has been addressed by enhancing the traditional STA approach in order to consider the delay shift (speed-up / slow-down) and the arrival times of aggressors/victim signals. This methodology, now implemented in PrimeTime-SI, is successfully exploited in this work to safely verify high speed digital circuits at STMicroelectronics. In this paper, the main features of PrimeTime-SI and the use experience will be reported, including some considerations about performances and pros / cons of the tool. Moreover, a front-end program (XA) to PrimeTime-SI, including a powerful GUI, is described. XA includes a layout physical viewer and a set of tools to effectively focus on the critical portions of the design. The STA / crosstalk results can be investigated either through a set of graphical widgets, or by running circuit simulations of the critical paths/clusters.
1.0 Introduction

In the today’s high speed digital designs, the crosstalk effects due to the coupling capacitance between interconnection lines has become one of the main performance limiting factors. In fact, as the geometry of transistors and interconnects becomes smaller, the coupling capacitance tends to be more of the 80% of the total wire capacitance [1] [2] [3] [4].

The interaction between signals on adjacent lines may cause both noise injection and signal timing deviation.

In this paper we address the verification of the timing performance of a design using the STA approach while accounting for both delay speed-up and slow-down of the signals due to the crosstalk effects.

2.0 Methodology Description

The crosstalk effects impair the traditional STA approach, because it doesn’t account for a relevant portion of the timing behaviour. Either the speed-up or the slow-down of the signals may cause a timing constraint violation, therefore a circuit failure.

The crosstalk timing effects depend on the dynamic characteristics of the signals, like the relative arrival times (signals overlap), phase and transition time, but the signal arrival times are themselves dependent on the delays in the fan-in cone.

Because of the dynamic nature of the effect, this cannot be addressed by a “coupling capacitance multiplying factor” [5]. Moreover, the main concern of a high performance design flow, is to avoid overlay pessimistic estimations, that may result from very conservative assumptions like the “doubling-the-coupling approach” [6] and/or assuming that every signal may occur at any time.

Nevertheless, the dynamic simulation would not be adequate to guarantee the circuit functionality under any possible input pattern, and would not be applicable for large VLSI designs.

Therefore, a new methodology has been defined in [7] to overcome the limitations of the traditional STA. A similar methodology has been developed and integrated in PrimeTime within a partnership project between STMicroelectronics and SYNOPSYS, and the resulting product-features are called PrimeTime-SI [8].

2.1 Approach

Based on the considerations above, the verification approach is based on the following main 3 tasks:

**TASK 1 Design setup and coupling capacitance filtering.**

Besides the usual setup of the clocks and the timing constraints, the user will possibly specify a set of thresholds that enable to filter the coupling capacitances during the loading of the SPEF files. Moreover, the designer may specify the set of nets that are not switching during...
the typical operation of the circuit (i.e. scan-chain, reset).

**TASK 2** *Full-chip xtalk analysis, assuming infinite arrival windows.*
Assuming that any signal may switch at any time, a fast yet conservative estimation is performed, for both worst-case speed-up and worst-case slow-down. Note that this step of the analysis may conclude the STA process, if no violations are detected.

**TASK 3** *Detailed analysis, accounting for the arrival windows computed at the previous step.*
This step is suitable to give a more accurate (less pessimistic) estimation of the delay values, by accounting for both arrival times and signals correlations. Although this is computationally more expensive, it is applied only to the subset of the nets that are relevant to the determination of the constraint violations, identified on the previous step.
By iteratively re-computing the delays, the min/max arrival windows will progressively shrink and the level of pessimism will reduce.

### 2.2 PrimeTime-SI implementation details

The PrimeTime-SI analysis consists in the following flow, as shown in Fig 1.

- **Filtering**
  During the SPEF reading phase, a coupling capacitance filtering process is performed (see TASK 1 above). When a capacitance is filtered, it is splitted and connected to ground with a multiplier factor of one. The filtering mechanism is based on thresholds that can be modified by the user. Several filtering criteria are available, based on the value of a single coupling capacitance, total net coupling capacitance and so on.
  A further filtering phase is performed when a ‘update_timing‘ command is called. While the previous one is based just on the value of the coupling capacitances, this filtering mechanism takes into account the driver strength. For a given victim net, a potential aggressor net is considered *effective* if the amplitude of the peak injected on the victim net, evaluated with the modified VMS model [9] and normalized to the power supply value, is greater than an user defined threshold.

- **Delay calculation**
  The first time, called "phase 0" or "infinite window phase", the design is timed by using an Arnoldi based engine [10] as delay calculator for all the design nets and the ECMF model [11] to estimate the crosstalk extra-delay on the coupled nets. The evaluated crosstalk extra-slopes are propagated too.
  The ECMF model requires as an input the Thevenin representation of both victim and aggressors driving cells: they are computed during the *effective capacitance* iteration step, which is performed during the timing analysis.

- **Reselection**
  At this level, a set of nets that may need a further analysis is identified. A set of selection criteria is available to the user. For example, the user may specify that only the nets which lie on
the top critical path (of each timing group), and the nets that are coupled with them, will be selected for the detailed phase.

Furthermore, since the crosstalk analysis is an iterative refinement process, a set of exit criteria are provided to the user, in order to guarantee the iteration scheme convergence. The iterative loop will end either because one of the above mentioned criteria is met or because no more nets are reselected.

The successive delays calculation will be performed by using the Arnoldi based engine both for delay calculation and for crosstalk extra-delay estimation. Moreover, the timing arrival window information will be taken into account, allowing to dramatically reduce the level of pessimism.

<table>
<thead>
<tr>
<th>Filtering</th>
<th>Delay calculation</th>
<th>Reselection</th>
</tr>
</thead>
<tbody>
<tr>
<td>rc_cc_absolute_threshold_in_pf</td>
<td>rc_cc_to_net_gc_ratio_threshold</td>
<td>xtalk_critical_path_reselection</td>
</tr>
<tr>
<td>rc_cc_vms_ratio_threshold</td>
<td>xtalk_max_mode_slack_threshold</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.** Crosstalk analysis flow implemented in PT-SI and variables defining the thresholds for the filtering and the reselection phases.
3.0 Use Experience

The only pre-requisite to run the PrimeTime-SI analysis, is that the SPEF file includes the net-to-net coupling capacitances.

A typical script to run the crosstalk analysis looks like the one shown in Figure 2.

In order to activate the crosstalk analysis features, the variable: `rc_activate_capacitive_crosstalk` has to be set to “true”. The crosstalk analysis will begin when the `update_timing`, `check_timing` or `report_timing` commands are issued.

A new option of the `read_parasitics` command (`-keep_capacitive_coupling`) has been added in order to load the coupling capacitances (by default, the coupling capacitances are splitted and lumped to ground). As mentioned above, the user may have direct control over the number of cross-coupled capacitors to analyze, by filtering from the analysis the capacitors that are small in value, therefore improving run time with negligible loss of accuracy. Moreover, a further filtering, based on the amplitude on the peak injected on the victim net, may be specified throughout the variable: `rc_cc_vms_ratio_threshold`.

The user may control the subset of nets that will be reselected for the next iteration by specifying the value of threshold variables. For example, to reselect the nets belonging to the critical path, the following variable: `xtalk_critical_path_reselection` has to be set to “true”. Other variables may be specified by the user in order to exit from the iteration refinement process.

After the analysis, the results regarding the crosstalk effects on timing are provided directly by using the `report_timing` command with the following new option: `-crosstalk_delta`. Moreover, a new set of attributes related to the crosstalk info has been introduced in PrimeTime-SI. For example, the voltage peak amplitude of the noise bump injected by each effective aggressor on a victim net: `<victim>` can be obtained with the command:

```sh
get_attribute -class net <victim> rc_xtalk_bumps
```

```
set rc_activate_capacitive_crosstalk TRUE
read_db ./ref_test.db
current_design ref_test
link
read_parasitics -keep_capacitive_coupling SPEF.spf
create_clock -period 10.0 clock
report_timing -input -crosstalk_delta
```

Figure 2. Example of a simple script file to activate the crosstalk analysis.
3.1 Performance
The new features introduced in PrimeTime-SI have been tested on two designs, D1 and D2. The first one is a 0.18\(\mu\)CMOS small block, while the second one is a 2.7 million gates design, based on a 0.25\(\mu\)CMOS technology. Regarding D2, the crosstalk analysis has been performed only on the top-level interconnects (about 11,000 nets over the about 1 million total nets).

For both the designs, all the filtering variables have been set to 0, in order to load all the coupling capacitances. Regarding the reselection criteria, just the nets belonging to the critical paths have been reselected for the next phase.

The results of the crosstalk aware timing analysis are summarized in Table 1.

<table>
<thead>
<tr>
<th>Design</th>
<th>Nets under analysis</th>
<th>SPEF size</th>
<th>Number of Cc</th>
<th>Reading parasitics phase</th>
<th>Update timing phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>7000 (over 7000)</td>
<td>80 M bytes</td>
<td>1.2 M</td>
<td>10’</td>
<td>20’</td>
</tr>
<tr>
<td>D2</td>
<td>11,100 (over &gt; 1,000,000)</td>
<td>2.1 G bytes</td>
<td>21 M</td>
<td>3 h 30’</td>
<td>3h 38’</td>
</tr>
</tbody>
</table>

By performing the same analysis without accounting for the crosstalk effects, the update timing phase would take 8’ and 44’ for the design D1 and D2 respectively.

Note that the performances are heavily dependent on the filtering thresholds (not exploited in these experiments). The setting of the filtering variables is tightly related to the particular technology and even to the design style. A preliminary analysis of the impact of the filtering step is strongly recommended in order to identify the optimal trade-off between performance and accuracy.

4.0 XA Front-End
The Xtalk Analyzer tool (XA) is a system that allows to better analyze the crosstalk results obtained by using PrimeTime-SI. It can be regarded as a front-end to the STA environment that includes a GUI enabling both net-by-net and full-chip analysis. Main features are:

- to identify the nets that are critical for crosstalk effects
- to browse a particular net/path in order to determine the source causes (strength of the drivers of the victim and aggressor nets, effective capacitance seen by every gate, arrival windows for
each signal, etc)

- to run circuit simulation on the critical nets
- to highlight the selected nets, and their aggressors, on the design layout. This enables powerful and useful cross-probings between the timing analysis environment and the physical design.

### 4.1 Architecture

The XA architecture is represented in Fig 3.

XA communicates with several PrimeTime-SI shells through a TCP-IP socket connection. The user may interact either through the PrimeTime-SI command line interface or by using the XA GUI.

XA has the capability to parse the SPEF file and run the circuit simulations of a particular portion of the design. Moreover, it handles layout information retrieved by the Design Exchange Format (DEF) file.

![Figure 3. XA architecture](image-url)
4.2 GUI

Several graphical widgets are available in XA. Firstly, the user will identify the critical nets/paths by the ‘Path Viewer’ widget, shown in Fig 4.

![Figure 4. The Path Viewer widget](image)

The Path Viewer shows a barchart, every bar representing a ‘stage’ which is composed by the *driving cell* and the *driven net* in the path. Separate contributes are highlighted with different color for gate delay, net delay and crosstalk delay. The barchart allows to easily identify the more critical stages from a timing (crosstalk delay) point of view. In the example, the most critical stage is the fourth (the interested net is SCLK_MUX/CLKCONTROLI/n477). By clicking on the related bar, a new window appears (Fig 5): it shows all the effective aggressors (and their contribute in terms of amplitude of the injected noise peak) of the selected net.
Figure 5. List of all the aggressors of the most critical net of the path shown in the previous figure.

A more detailed analysis can be done on the selected nets by using the ‘Net Viewer’ widget. It summarizes, for a given victim net, all the information regarding the drivers characteristics, the interconnects parasitics and the timing data, like the arrival windows, slew times and the delay times. Fig 6 shows a window of the ‘Net Viewer’ widget.

A further widget available in XA is the ‘Wiring Viewer’ widget, showing a graphical representation of the physical layout of the design. Every net is represented by a set of segments, disregarding the layer information. The segments may be colored based on the level of the delay deviation, allowing to identify crosstalk hot spots in the design.

Moreover, a net, a path or a victim-aggressors cluster can be identified in the physical layout, simply by clicking on the net name/bar in the previous widgets. The Fig 7 shows the top-level interconnects of the design D2.
Figure 6. The Net Viewer widget.

Figure 7. The Wiring Viewer widget.
5.0 Conclusions

The new features in PrimeTime-SI enable the STA of the circuits in presence of timing deviations due to the crosstalk effects.

The methodology and the main commands/attributes have been presented and the tool has been applied to two 0.25µ / 0.18µ high speed digital designs.

Moreover, a program called XA, acting as a front-end to the PrimeTime-SI shells, has been illustrated, summarizing the “use model” and the main GUI widgets.

Although some areas of enhancement, like performance and accuracy, will have to be addressed, PrimeTime-SI can be regarded as a valuable tool to handle a problem that is expected to become dominant in the next future technologies.

6.0 References