Building the Hierarchy from a Flat Netlist for a Fast and Accurate Post-Layout Simulation with Parasitic Components

Pierluigi Daglio - David Iezzi - Danilo Rimondi - Carlo Roma - Salvatore Santapa‘
STMicroelectronics - Agrate Brianza - Milan - Italy
E-mail: pierluigi.daglio@st.com

Abstract

Main concerns related to post-layout simulation, today, are about the format of the netlist coming out from the parasitic extractor. In fact, such a netlist is usually flat so that readability, whether compared to the pre-layout hierarchical one, is very poor due to device and net names which often change and to the difficulty to compare pre-layout and post-layout output signals. Furthermore, simulating such large flat netlists is frequently time consuming because it is not possible to exploit algorithms like Hierarchical Array Reduction (HAR) and Isomorphic Matching (IM), strength points of state-of-the-art full chip simulators. In this paper, we present a new approach that, starting from a flat netlist with parasitic components and a pre-layout hierarchical one, allows to create a fully hierarchical post-layout netlist containing device parameters and parasitic components directly extracted from the layout. In this way, a fast and accurate post-layout simulation is made possible by the use of look-up table simulators, taking advantages from the HAR and IM algorithms as mentioned before. This methodology has been integrated in a complete design flow to guarantee first silicon success, cut down time-to-design, improve time-to-market and streamline design quality.

Introduction

One of the main challenges designers face with the most advanced technologies is the post-layout simulation of their applications including parasitic components. As a matter of fact, netlists generated by commercial parasitic extractors are usually flat so that transistor level simulation is always very complex and time consuming, sometimes impossible. For that reason, look-up table simulators can not exploit their best algorithms, like Hierarchical Array Reduction and Isomorphic Matching, to speed up circuit simulation, but they are forced to process the whole circuit with a single large Kirchoff matrix, even because elements are tightly coupled by extracted parasitic components so that relaxation algorithms are difficult or dangerous to be used too. Because of all these motivations, a new approach has been put in place in our company with the intention of generating a post-layout fully hierarchical netlist, containing parasitic components, that has been qualified on several test cases to prove its reliability and its benefits with respect to the flat approach. A new tool, called as *icSim*, has been developed to allow designers to suitably load, inside a pre-layout hierarchical netlist, device parameters from the post-layout flat netlist (AS, AD, PS, PD, ...) and parasitic components either from the same flat netlist or from the DSPF (Detailed Standard Parasitic Format) file generated by the parasitic extractor. All those data are combined together to create a fully hierarchical netlist, containing the exact values of device parameters as taken from the layout plus all the extracted parasitic components.

1. Hierarchizing the Flat Netlist

To create a fully hierarchical post-layout netlist with parasitics, a new tool, called as *icSim*, has been developed inside our company. Using *icSim*, you can load a pre-layout hierarchical netlist in CDL format (this format has been chosen because it is the netlist format used for LVS purposes and the parasitic extraction runs properly only whether LVS process is error-free) together with the SPICE.SPI flat netlist generated by the parasitics extractor. The SPICE.SPI file contains all the device parameters and the parasitic components that the designer decided to extract. At this point, icSim merges the information held in both netlists and, finally, generates a hierarchical netlist back-annotating parasitic components internally to sub-circuit blocks instead of at the end of the netlist like in flat mode. Obviously, this tool comprises several algorithms to properly manage controversial situations like, for instance, when at the same sub-circuit master correspond different parasitic values at instance level on the layout. In fact, it may happen that two instances of the same sub-circuit layout master are placed in a different environmental condition on the layout, so that parasitic values can slightly differ. In this kind of situation, the tool is able to choose the most suitable values for the parasitics to fit at the best in the hierarchical netlist or, in case, to double the sub-circuit master generating two different sub-circuits when needed. icSim is also able to load

1530-1591/04 $20.00 (c) 2004 IEEE

directly the DSPF file generated by parasitic extractor, alternatively to the SPICE.SPI netlist, and to create multiple DSPF files, one for any sub-circuit, to exploit at the maximum level the features of certain look-up table simulators. Now, with a simple command, we can generate a fully hierarchical post-layout netlist with parasitics back-annotated at sub-circuit level. If there is any name conflict or any mismatch in the net names, the corresponding parasitic components are back-annotated at the top level, according to algorithms included in the tool.

2. Simulation Output Data Comparison

In the next two paragraphs, we point out two different kinds of comparison. The first is the graphical output data comparison related to the circuit simulation, the second is a data table comparison related to the data collected in the laboratory. Output voltage and efficiency measured on silicon will be compared with those saved during simulation time with dedicated functions of the simulation engine. In figure 1 you can see, starting from the top, all together the pre-layout, the flat post-layout, the DSPF direct inclusion and the hierarchized post-layout output voltages obtained using a look-up table simulator.

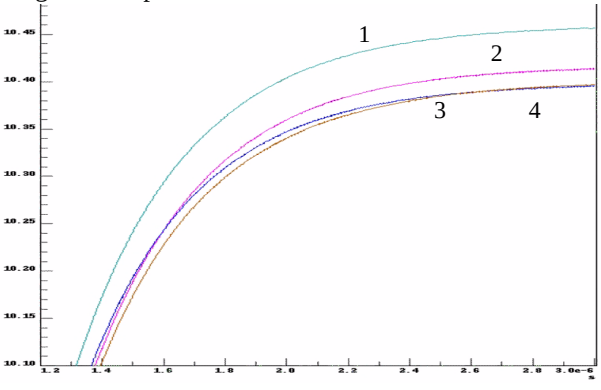

Figure 1. Pre-layout (1), flat post-layout (2), DSPF direct inclusion (3) and hierarchized post-layout simulation (4) output results

The output voltage in the pre-layout simulation is about 60mV greater than in the post-layout simulation due to the effects of parasitic components. You can also notice how the difference between the flat and the hierarchized post-layout simulation is quite small, about 15mV, so that we can say that our benchmark demonstrates that the hierarchized netlist is reliable. Simulation of hierarchized netlist is about twice faster than the flat one, therefore we were able to get the same output results gaining a factor of two in speed. Another thing worth to notice is that output waveforms related to DSPF direct inclusion and to hierarchized netlist are very close, less than 1% of difference.

3. Simulation Data Compared with Data Measured on the Silicon

At this point, we can compare all the data related to simulations directly with the data measured on the silicon. To accomplish this task we ask the simulator, during the simulation phase, to save the efficiency of the charge pump, then we create some tables and graphs to compare all together the different results related respectively to pre-layout, flat post-layout, hierarchized post-layout (generated by icSim) and silicon measurements. In table 1, you can see output voltage (Vout) and efficiency (EFF) of the charge pump versus sunk current for the different kinds of simulation and the measurement on the silicon. The current values have been forced at each of the nine values in the first column of the table with the aim of running nine different simulations to observe the behaviour of output voltage and efficiency with respect to the typical loads of the charge pump. The value of the loads has been kept constant for all the different simulations. Looking at table 1, you can examine very meaningful details to understand and evaluate the difference of various simulations due to the netlists. The most important measurement is the efficiency. As you can see, in pre-layout simulation, efficiency is maximum at 150uA of sunk current reaching almost 70%, output voltage is 9.08V. If you take a look at flat and hierarchized post-layout, you can see that they both get to the maximum efficiency at 300uA of sunk current reaching, respectively, almost 50% and 51% with output voltages at 8.0559V and 8.0552V.

	prelayout		flat		hierachized		silicon	
current	V_{OUT}	EFF	V_{OUT}	EFF	V_{OUT}	EFF	V_{OUT}	EFF
0uA	1,05E+01	0	1,02E+01	0	1,02E+01	0	1,01E+01	0
50uA	1,00E+01	0,5575	9,90E+00	0,2026	9,91E+00	0,2153	9,52E+00	0,1959
100uA	9,56E+00	0,674	9,56E+00	0,3293	9,57E+00	0,3446	9,13E+00	0,3196
150uA	9,08E+00	0,6988	9,21E+00	0,4101	9,24E+00	0,4265	8,70E+00	0,39
200uA	8,58E+00	0,6885	8,85E+00	0,46	8,86E+00	0,4741	8,38E+00	0,4509
250uA	8,06E+00	0,6573	8,47E+00	0,4876	8,47E+00	0,5009	7,98E+00	0,485
300uA	7,47E+00	0,6114	8,06E+00	0,4978	8,06E+00	0,5104	7,54E+00	0,4967
350uA	6,76E+00	0,5448	7,60E+00	0,4929	7,59E+00	0,5031	7,08E+00	0,4975
400uA	5,74E+00	0,441	7,07E+00	0,4727	7,05E+00	0,4793	6,54E+00	0,4809

Table 1. Vout and efficiency versus current for the different simulations and silicon data

Looking at data measured on silicon, maximum efficiency, corresponding with 350uA of sunk current, is 50% with 7.08V output voltage. The difference in current (50uA) and voltage (about 0.7V) is due to the fact that we did not extract, in our benchmark, the parasitic resistors responsible for the voltage drop on Vout and for the higher sunk current.