Circuit Simulation: Should it be faster or more insightful (or both)?

By Maxim Ershov

I recently read John Cooley's article on the Troublemakers panel discussion at DAC 2020 between Mentor's Joseph Sawicki and Cadence's Anirudh Devgan about Mentor's AFS-XT SPICE simulator beating Cadence's Spectre-X in speed by a factor of 3x-10x. (reference: http://www.deepchip.com/items/0588-21.html)

A faster SPICE is better than a slower SPICE, no question. If SPICE shows you good circuit behavior and good simulation results, you are done, and doing this faster is always good. However, SPICE speed isn't everything these days.

With the transition in recent years to advanced technology nodes (16/10/7/5nm), the on-chip interconnects and layout parasitics have become the biggest pain for IC designers. In FinFET technologies, the magnitude, impact on circuit characteristics, and number of parasitic elements have grown exponentially. It is these parasitic elements that are causing problems with circuit simulation and verification (SPICE, IR/EM, timing, etc.).

The important point is that parasitics are not only slowing down SPICE simulations, but also causing a circuit to underperform, or to behave in an unexpected, counter-intuitive manner. Common problems include: degraded operating frequency and speed; reduced bandwidth, resolution, matching, linearity, and signal integrity; increased noise, offset, IR drop, EM, ESD and timing violations – and many more issues.

Debugging these parasitic effects is causing a lot of frustration, delayed time-to-market, and degraded PPA (performance, power, area). Engineers who are designing in 7nm for the first time are often baffled by the fact that post-layout simulations are so different from schematic simulations, and are not meeting the specs. Schematic simulations (almost) do not matter at all - ***your layout is your circuit***.

Parasitics are no longer a second or third order effect (as in the past) – they are more important than devices. They are a first order effect, and should be treated that way. But the tools, flows, and methodologies have not changed much since the early days of IC design, and have not adapted to this new reality. These EDA tools are still treating parasitics as a nuisance, as second-class citizens, without providing designers and layout engineers proper tools for understanding, analyzing, debugging, and improving them.

This is a message we received recently from one of our customers:

> "I was on a call earlier with *** (EDA vendor) about their new *** (SPICE tool) simulation engine. They were "selling" us on the new higher speed that it offers. About halfway through and for the remainder of the discussion, (I told them) about how simulation time isn't as critical anymore, and instead trying to understand the results is taking more of our time. It's not enough these days to run simulations faster and faster when it's hard to understand what's fundamentally going on in the post-layout design. In particular, trying to understand how the parasitics affect the simulation is becoming the dominant activity. Which is exactly where your tool comes into play."
>
> - Tom Krawczyk

(VP of Analog Design at Jariet Technologies)

When a designer sees bad simulation results from post-layout netlist – while schematic simulation was working just fine – clearly, this problem is caused by parasitics. But where, what portion of the layout is causing this – which layer, net, and specific polygon? Usually, out of thousands, millions, and billions of parasitics in the post-layout netlist, only very few elements are absolutely critical and need to be fixed, while others can be left alone. But how do you find these critical parasitics, and critical layout shapes?

A faster SPICE will not help you here. It reminded me a famous quote by Henry Ford: "If I had asked people what they wanted, they would have said **faster horses**."

You need something different, something that provides a different viewpoint than SPICE. Comparing sports cars - whether Ferrari or Lamborghini is faster - is irrelevant if you live in a city with heavy traffic congestion. What you need is `insight and actionable results, something to tell you which route to take to avoid traffic, so you can unleash the full power of your sports car (or, maybe, a bike?).

Similarly, designers are dreaming about having a tool that can give them some insight to help find the proverbial needle in a haystack - the few critical parasitic elements responsible for creating a bottleneck, choke point, mismatch or a weak area in their design. Something that will point them out – this net, this layer, these polygons – as shown on the snapshots below. Without such a tool, designers and layout engineers are seeing a black box of parasitics without understanding what's inside. They're shooting in the dark, and searching for a black cat in a dark room.

Such insight and help can be provided by a new methodology and new EDA tools that focus on parasitics as a central, essential part of the design. (Even the term "parasitics" becomes meaningless and misleading when parasitic elements are more important than active elements) A new approach that "understands" parasitics and design intent, and can provide a guidance for layout optimization with the minimum number of required changes, enabling engineers to make changes only where absolutely necessary, and with surgical precision. A tool that shows where the biggest parasitics problems are, and also where parasitics are not critical (where layout can be relaxed and area/power recovered), as they're shown on the screenshots below. And a new solution that can quickly help designers find an optimal tradeoff and balance, in the eternal conflict between resistive and capacitive effects.

With such a tool, engineers see their debugging time for parasitics problems (revealed by post-layout SPICE simulation or anticipated by knowing the design intent) in advanced nodes reduced from days, weeks and months down to minutes. With such a tool, they can verify a lot of things without even doing any SPICE simulations. This is because the engineers know the design intent, they know their circuit, and what's critical – whether it's capacitance or current matching or weighting in precision analog designs, or RC delay or bandwidth in high-speed SERDES, or signal integrity, or resistance on ESD discharge path, and so on. Very importantly, such a new solution also enables analysis at the top-hierarchy level, which is absolutely not possible to do with SPICE (or other similar tools).

Combining the designers' insight into their circuit with powerful analysis, debugging, and visualization capabilities allows designers and layout engineers to see right away where the problem is, so it becomes obvious what layout changes need to be done to correct for parasitics effects.

	Rank	Layer	Res [Ohm]	Res_perc [%]	Res_cumul [%]
1	1	n_gpoly_lvt	371.96	39.544	39.544
2	2	n_fpoly_lvt	153.16	16.283	55.826
3	3	VG	96.789	10.29	66.116
4	4	M4	58.525	6.2219	72.338
5	5	VIA2	49.929	5.3079	77.646
6	6	n_odtap	46.811	4.9766	82.622
7	7	VD	33.766	3.5897	86.212
8	8	M2	31.02	3.2977	89.51
9	9	VIA1	29.842	3.1725	92.682
10	10	VIA0	18.655	1.9832	94.665
11	11	VIA3	17.974	1.9109	96.576
12	12	M1	15.205	1.6165	98.193
13	13	M3	11.658	1.2394	99.432
14	14	M0	3.2761	0.34828	99.78
15	15	MD_OD_N	2.065	0.21953	100
16	16	[illegible]	4.9427e-14	5.2547e-15	100
17	17	[illegible]	2.9149e-14	3.0989e-15	100
18	18	[illegible]	2.3092e-14	2.455e-15	100
19	19	[illegible]	6.7853e-15	7.2135e-16	100
20	20	[illegible]	1.5695e-15	1.6686e-16	100
21	21	[illegible]	6.0589e-16	6.4412e-17	100
22	22	[illegible]	2.2509e-17	2.3929e-18	100
23	23	[illegible]	1.33e-17	1.4139e-18	100
24	24	[illegible]	1.8865e-19	2.0055e-20	100
25	25	[illegible]	3.7748e-23	4.013e-24	100
		Total:	940.64	100	100

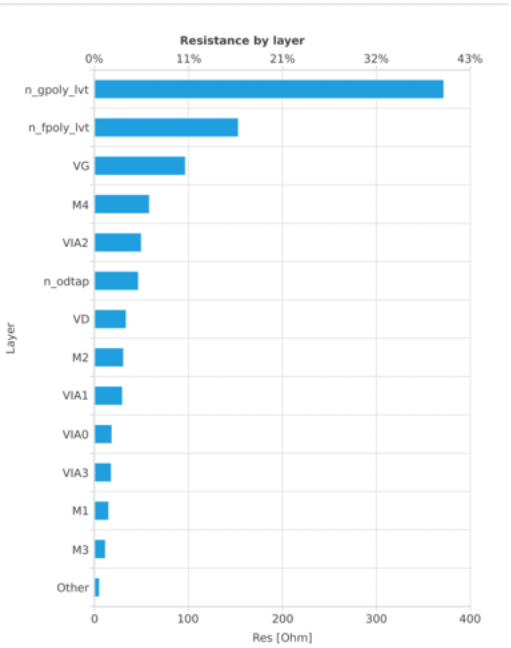

Fig. 1. Ranked contributions of different parasitic elements to the design characteristic, by layer.

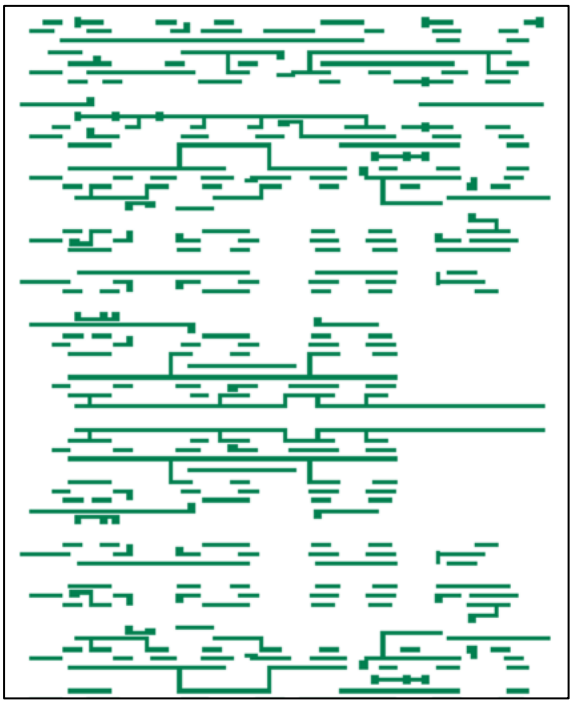

Fig. 2(a). Layout view – can you find polygons of a critical net, that are causing electrical problems for your circuit?

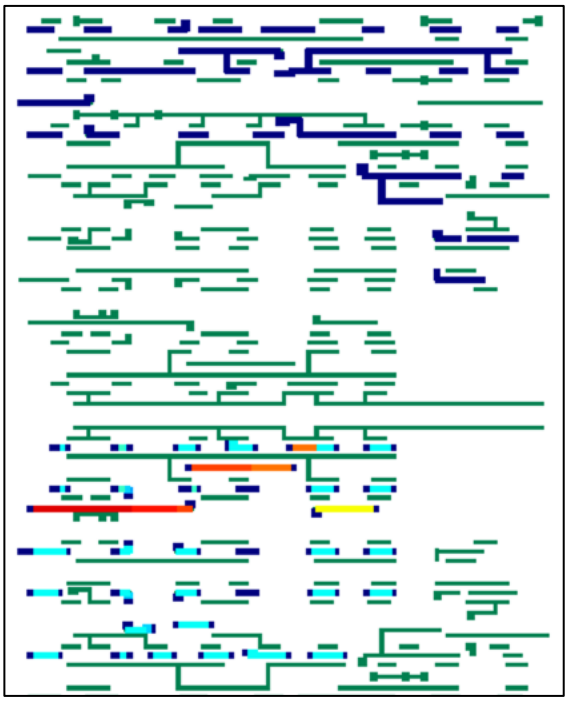

Fig. 2(b). ParagonX view – polygons of critical net are highlighted by color reflecting their impact on IC characteristics. Red color indicates important polygons, blue color – unimportant polygons.