

# Validating netlist reduction, circuit extraction

By Mathias Silvant  
President and CEO  
edXact SA

The significant burden of parasitics on the performance of post-layout verification forces design engineers to use additional techniques to match the requirements of next-generation designs. A real gap appears between layout extraction and circuit simulation when adding layout parasitics into the flow. A review of the existing techniques as well as a merciless way to doubtlessly validate netlist reduction and circuit extraction are presented in this technical article.

Moving forward to advanced technology nodes severely endangers circuit design. The International Technology Roadmap for Semiconductors' process technology roadmap together with Moore's Law states that the number of transistors over process nodes raises exponentially, doubling approximately every second year. The number of interconnections between all these transistors and the length and complexity of the supply lines is growing at the same pace. Today, when working on design kits in 40nm and below, more parasitics need to be taken into account in order to model effects. Their impact on circuit function and performance is getting predominant: Delay of a signal on a line is mainly determined by the interconnection and its physical parameters and not by the active devices anymore.

To model interconnections, EDA tools use so-called parasitic elements (resistors, capacitors, coupling capacitors, inductors, mutual inductors and others). We call them parasitic, because they have not been designed, but their behaviour needs to be taken into account. They substantially alter

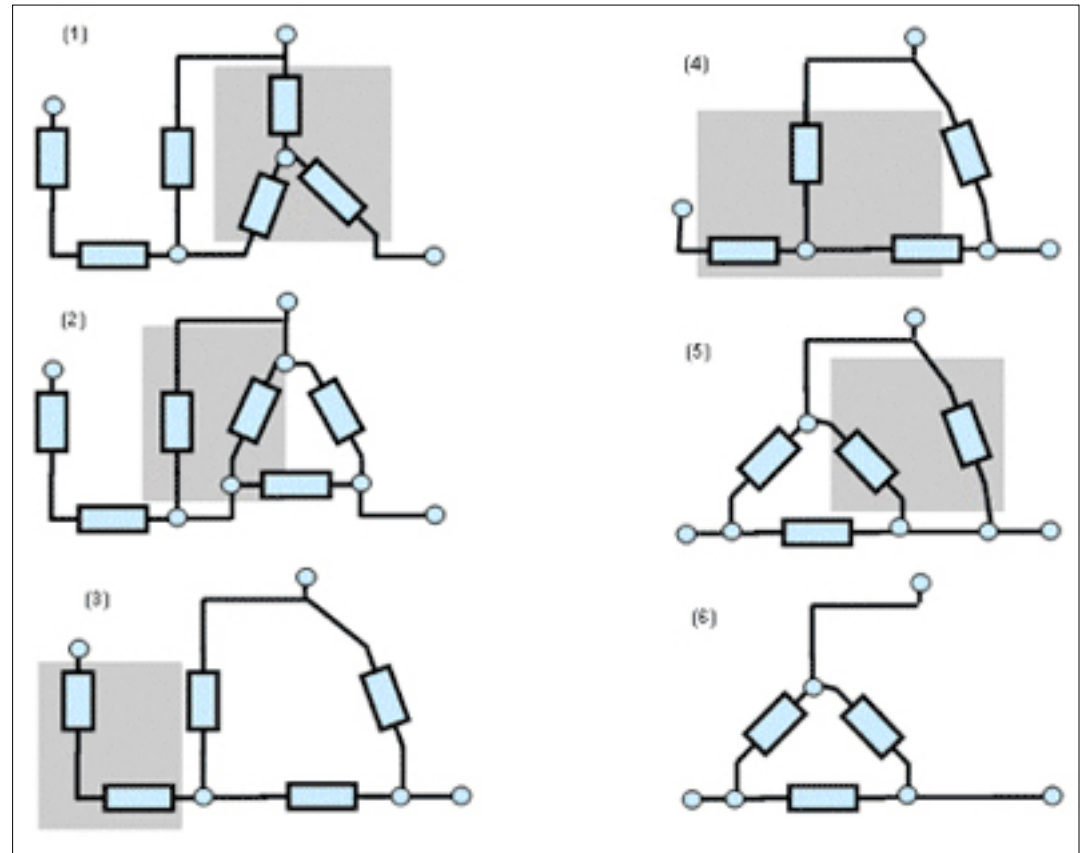


Figure 1: A mesh of six resistors is iteratively reduced to a mesh of three resistors by star-triangle transformations.

the intended circuit behaviour. Moving from one technology node to the next the number of parasitics increases at least at the same pace as the number of devices: A rule of thumb is about 4 parasitics per transistor. A full-chip extraction of an advanced circuit design will therefore contain several hundreds of millions of those parasitic network components.

CAD engineers and EDA vendors seek to improve current methodologies and flows to get a hand on the complexity issue in post-layout simulation. They can take advantage of different techniques, which can all be combined: hierarchical accurate spice simulators, multi-threaded and multi-core simulation tools, selective extraction approaches, and selective back-annotating of parasitics to pre-layout simulation, improved netlist reduction.

## Reduction technique

Among the techniques available to improve the speed and capacity of post-layout simulation taken parasitic components into account, netlist reduction is among the most promising techniques.

Netlist reduction is the keyword applied to describe the process of transforming a netlist that contains parasitic elements into a second netlist of smaller size and less parasitic elements. In principle, we can claim that the smaller the netlist, the faster the simulation tool. Ideally, the reduced netlist should have exactly the same electrical behaviour as the original netlist. However, nothing is for free: There is a trade-off to be made between the accuracy of the applied algorithms and the degree of reduction.

On top of the complexity challenges, the most important chal-

lenges of state-of-the-art netlist reduction tools are loopholes in the definition of exchange file formats and the lack of necessary mathematical conditions in the extraction tools. A typical example for a quasi-standard exchange file format is the DSPF (Detailed standard parasitic file) format: At the time of definition, capacitive coupling, inductance and magnetic coupling was not anticipated, and we find ourselves today with at least four different ways of defining coupling capacitors in DSPF files. These loopholes could easily be closed by using the Standard Parasitic Exchange File (SPEF) format. However, main industry applications run on DSPF. While these issues are only related to the format and can be solved by developing several distinct interfaces, mathematical conditions must be respected. To name one:

Passivity is a necessary condition. Inductors, capacitors, resistors even combined cannot generate energy; this is a basic physical law. Most of mathematical Model Order Reduction (MOR) algorithms start off with this condition, which might not be respected by extraction tools due to numerical inaccuracies. The direct application of the many proposed MOR algorithms are therefore generally inapplicable for industrial applications.

Before reviewing today's available techniques, we can already state that there is no universal technique usable for netlist reduction. The good news, though, is that EDA tools can successfully apply a combination of different techniques.

We can distinguish four categories of approaches used in industrial tools: filtering, recursive star-triangle transformations, delay-oriented local operations and mathematical solutions called MOR.

### Filter

The usual industrial approach to netlist reduction is the determination of threshold values. If a component has a value below or above one of the thresholds, it will be eliminated from the netlist. An example of this approach is the postulation of a  $g_{min}$  value, which determines a minimum admittance. Admittance values below this threshold are dropped. This simple, but effective, technique can be applied to any of the linear parasitic components.

For obvious reasons, those techniques are called "filtering techniques." They are very efficient when the values of the parasitics vary a lot. When all values are in the same range, however, it is hard to determine where to set the threshold and important errors might be introduced in the circuit. They also need to be implemented with care to ensure the completeness of the electrical circuit: A filtered element must not leave a hole in the netlist. With respect to accuracy it can be stated that filtering methods

without any error control are relatively dangerous in the sense that filtered parasitics might have been important in their context and thus the simulation result will alter. Filtering without error control might also introduce instabilities into the circuit.

### Recursive star-triangle transformations

Recursive star-triangle transformations are as old as electrical engineering is taught in school and universities. A three-port resistive mesh in a star-shaped constellation can be transferred, without loss, into a three-port mesh in a triangle-shaped constellation. Apart from the advantage of losing an internal node (which is good because it reduces the dimension of the matrices), it prepares the resistors of one three-port to be merged with resistors of an adjacent three-port. In the example given in **Figure 1**, a mesh of six resistors is iteratively reduced to a mesh of three resistors by star-triangle transformations.

Iterative star-triangle transformations can be applied to resistive meshes (e.g. resulting from an extraction of a power rail) without any loss in accuracy, stability or passivity. When targeting a single frequency point, this can also be executed for complex impedances in the same way. Compared with filtering mechanisms, this technique is much more expensive to implement and more time-consuming in execution. This technique is not used in practice.

### Delay-oriented local operations

Introduced a decade ago, the Time-Constant Equilibration Reduction algorithm is a node-elimination algorithm that can be described as follows: for each resistor of a node in a net, a characteristic time constant is calculated, by multiplying the value of the resistance by the capacitances connected to the node. If the time constant is smaller than a threshold, the resistor can be eliminated or better merged with an adjacent resistor. This operation is iteratively

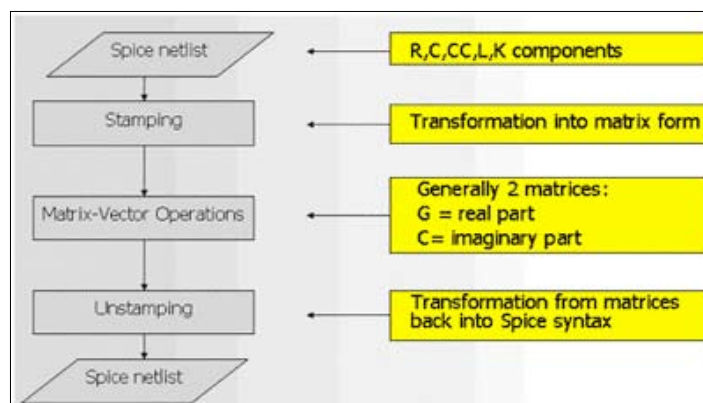


Figure 2: Shown is the general approach of MOR for netlists in EDA.

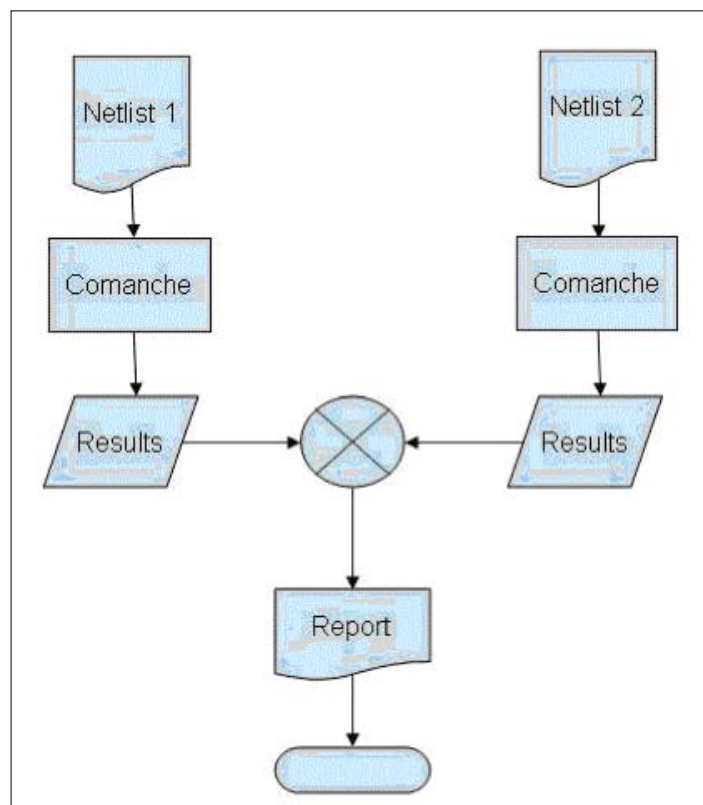


Figure 3: Shown is a netlist reduction validation flow.

done for all nodes in a netlist. The algorithm has been for a long time the best algorithm for industrial applications. Its main advantages are a very easy way to implement, because it doesn't require sparse-matrix or Eigen solvers and a high efficiency. The algorithm also produces an output that takes the form of an RC network close to the original network topology, and it is capable of dealing with networks with many ports. The algorithm works best on independent nets. It is getting in trouble when nets are heavily coupled with others; it cannot well be applied to meshed networks. Its initial application was the preparation for delay calcula-

tion and for a long time it was the ideal algorithm, since signal nets usually are of simple structure.

### Mathematical MOR

MOR exists in different domains of engineering. With respect to netlist reduction for parasitics, MOR is applying sophisticated mathematical approaches on a large, but linear problem. The general approach of MOR for netlists in EDA is depicted in **Figure 2**.

The starting point is a netlist most commonly available in SPICE format (or a derivative like DSPF). It contains all the circuit or parts of the circuit, as well as the parasitics of different types: R (resistors), C

(capacitors from net to ground), CC (coupling capacitors between different nets), L (self inductors of one net), K (mutual inductors between different nets). First, parasitic components are isolated from the rest of the circuit. Then, the linear circuit components are transformed into a matrix representation. This step is called “stamping.”

It generally ends up with two sets of matrices: G for the real parts and C for the imaginary parts. At this point, a mathematical model of the electric circuit exists and can be transferred into a different model with similar behaviour but lower order. To achieve a reduction of the order of the mathematical model, different matrix-vector operations are applied. The algorithms that have been published can have a high complexity and therefore suffer from natural limitations with respect to the size of the problem they can handle.

Once the reduced order model is generated, it is transferred back into a netlist format. This step is called “unstamping” and is one of the most complicated tasks in the netlist reduction approach via MOR. Since only a very limited number of electrical components exist to represent the generated mathematical model and since users of netlist reduction software want to see component values with physical meaning, the choice of algorithms is reduced to extremely few possibilities. If the algorithms are not chosen with the physically meaningful unstamping in mind, the unstamping process can increase the order of the model up to a point that the overall reduction is annihilated.

Pure MOR algorithms are generally computationally very intensive, complicated to implement and need special care to

prepare the incoming data to ensure the mathematical requirements. Some of the algorithms cannot guarantee stability and passivity. The problem to generate physically meaningful netlists in the unstamping process has already been mentioned. The main advantage of those algorithms is that they can treat netlists of any structure, especially those that are impractical for the algorithms based on local operations.

### Accuracy is key

While the increase of performance of the post-layout verification flow is the main interest, attention must be paid to ensure a sufficient level of accuracy. Nobody can be interested to trade better performance against loss in the quality of the simulation results. The trend moving from one technology node to the next is in favour of more accurate tools, to capture effects that were negligible down to a certain feature size, but which play a major role in technology nodes with smaller feature sizes and lower supply voltages.

A common way of measuring the error of different layout extraction and netlist reduction techniques is to run a set of typical simulations for different settings and to compare the simulation results. The measure for the error can be anything, depending on the application. For example, a memory designer will carry out setup- and hold-time simulations. The designer will compare the ramp-up signals and calculate an error from the difference in time. An RF engineer might design a voltage controlled oscillator (VCO).

Since the VCO varies the frequency of an oscillating signal following a potential difference at the entry, the RF engineer will

measure the error in frequency domain. Another engineer might design a power amplifier (PA). The designer will be interested to know about the SNR of the PA and measure the error in that sense.

Other criteria are delay, number and frequency of harmonics, phase of a signal, maximum level and shape of a signal, value of phase noise etc. The problem with this common approach is the difficulty to separate the various impacts of the various tools: extraction with embedded netlist reduction and simulation with another embedded netlist reduction. Even if embedded reductions can be temporarily disabled, the capacity of simulation tools without any compression or acceleration is prohibitive to be used on other than small circuits. Unless the netlist reduction is completely independent from any other tool and has simple and accessible interfaces (like Jivaro), it is difficult to measure the error on large-scale designs. If a design flow allows for several extraction tools, the characterisation of the error even needs to be done for each tool separately, since each extraction tool applies its own, usually unpublished algorithms.

### Quantifying the error

A neutral way to judge on the accuracy of netlist reduction technology, whether integrated or independent, is to use a tool like Comanche. One of the functionalities of this tool is to accurately calculate the effective resistance between external connections (ports) of interconnections. It can be done for one net, for several nets or even for a full netlist. It can be run on unreduced netlists or reduced ones whose input format is DSPF and SPEF.

Even for a very high number n

of ports on a net (the number of combinations N is  $N=n*(n-1)/2$ ), Comanche can quickly calculate all these resistances with accuracy up to the sixth digit after the decimal point. Verifying a large power net by regular simulation takes days, while the Comanche tool can calculate this in a couple of minutes. The typical flow is to consecutively run two netlists through Comanche. The reports can be compared in an automated way, and the difference between the two netlists is flagged. While this approach is useful to judge on the performance and accuracy of netlist reduction, it can also be useful to compare different extraction tools for parasitics. Accuracy on resistance is already giving a lot of insight; the investigations can easily be extended to capacitors. If the accuracy of the verified netlist reduction is on an acceptable level, the CAD engineer can guarantee that the simulations with the netlists that have been compared will generate the same result. As said before, the inverse is not necessarily true: A similar simulation result does not guarantee good reduction accuracy.

### Conclusion

Netlist reduction is a vital part of post layout verification flows for all kinds of simulation and analysis. We have very briefly reviewed the main categories of applicable algorithms for netlist reduction. It turns out that not one netlist reduction algorithm exists. That serves all possible netlist. A combination of different techniques, associated with accurate error control can help solve the dilemma. A neutral and fast way of verifying the efficiency of netlist reduction was outlined, which can also be used in order to compare circuit extraction tools.