Resistance Calculation Approach in Hipex-NET

Hipex-NET uses two techniques to calculate resistor values. First method of resistance extraction, which is usual for full chip netlist extractors and uses heuristic algorithms to recognize common shapes for which were obtained empirical formulas depending on geometry of resistor body and resistor terminals. Unfortunately, this method doesn't cover the wide range of resistors and can handle only rectangular resistors, L-bends, and T-shaped resistive fragments. It also can be used to calculate resistance of snake and dog-bone shapes. We should also note that comparing to ***Maverick***, ***Hipex-NET*** recognizes some new shapes as routine shapes for which resistance value can be calculated by the well-known formulas [1].

If this method cannot be applied to a specific resistor then the tool uses the general technique.

The general technique is capable of finding the resistance between any set of arbitrarily shaped boundaries through any shape resistive region, see figure 1. This well-known technique solves Laplace's equation,

$$\nabla^2 \varphi = 0,$$

over the resistive region Ω. The potential φ is a function of (x, y). Boundary Γ consists of two kinds of boundaries, Γ_φ and Γ_q. Γ_φ is the forced boundary and potential on this boundary is constant and the current flows across the boundary. This condition can be written as

$$\varphi = \varphi \text{ on } \Gamma_\varphi$$

The other Γ_q is free boundary, , and the current along the perpendicular direction to this boundary is zero. This condition can be written as

$$q = \frac{\partial_\varphi}{\partial_n} = q_0 \text{ on } \Gamma_q$$

Hipex-NET can extract a full equivalent resistance network from the shape of which the terminals are connected to P different electrical nodes. Usually the typical value of P is two, but the algorithm implemented in ***Hipex*** implies no restrictions on P. Resistance between any ports and can be easily found if bias voltages are chosen as follows:

$$V_j = 1 \text{ and } V_j = 0,\ k = 1,\ldots, j - 1, j + 1,\ldots,P.$$

Then,

$$R_{kj} = \frac{\Delta V_{kj}}{I_k} = \frac{1}{I_k}$$

and the current can be evaluated by the line integral

$$I_k = \int_{\Gamma_\varphi} \frac{\partial_\varphi}{\partial_n} d\Gamma.$$

In ***Hipex-NET***, the boundary element method (BEM) is employed to solve the above Laplace's equation. The BEM transfers partial differential equation in a domain into a set of integral equations on the boundary. The discretized integral equations are the only equations solved. This method yields a much smaller linear algebraic equation systems comparing to finite difference method (FDM) and finite element method (FEM). In addition, mesh generation in the BEM is not that expensive. We should note that the similar approach was implemented in the old Silvaco's tool ***Maverick*** but the algorithms used in ***Hipex-NET*** to generate boundary meshes and auxiliary matrices are faster and more accurate. They also reduce the memory space drastically.

References

[1] Kuang-Wei Chiang, "Resistance Extraction and Resistance Calculation in GOALIE2", Proceeding 26-th Design Automation Conference, pp. 682–685, June 1989.

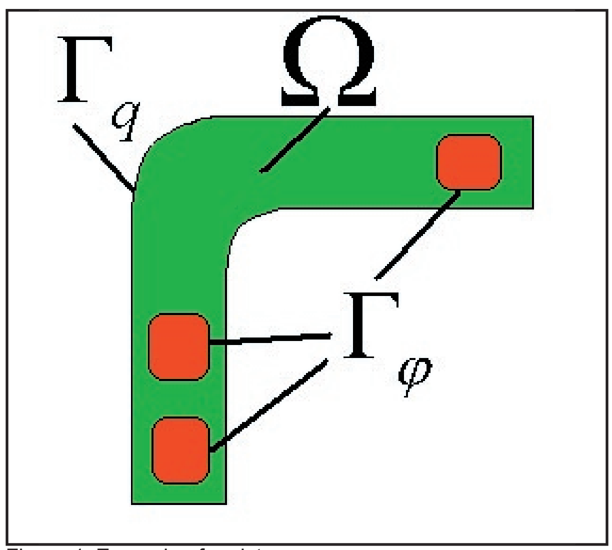

Figure 1. Example of resistor.