FOSDEM 2018 – Automated system partitioning based on hypergraphs for 3D stacked integrated circuits

Quentin Delhaye, Dragomir Milojevic
Université libre de Bruxelles, Brussels, Belgium

With the evolution of integrated circuits (ICs), 3D integration is becoming a short and middle term viable solution to increase the system performance, functional density per unit area and reduce system power, as an alternative to costly sub 7nm CMOS technologies. However, as trendy as it may turn, there is no viable Electronic Design Automation (EDA) toolchain that allows to design ICs in 3D. Our work aims at developing tools that could be used in standard 2D EDA chains to deal with 3D ICs. Such tools will automatically transform any 2D design into an optimized 3D architecture by extracting a graph from the DEF file coming out from 2D synthesis and place&route flow. Afterward, this graph can be partitioned automatically using state-of-the art graph partitioning tools.

In order to create the toolchain, we developed a custom DEF parser to answer specific and limited needs that existing solutions such as Verilog-perl or PyVerilog did not allow. The software scans the DEF to find all the gates, pins and nets in the given design. When analyzing nets, it also parses the routing information to determine the total length of the net, which will be needed in the partitioning phase.

In parallel with the DEF parsing (desgin input), the LEF technology file (technology input) is also parsed to fetch the characteristics of the gates that will be used later on in the clustering and partitioning steps.

Once the design has been parsed, it can be clusterized to reduce the problem complexity and the output is formatted to extract the underlying hypergraph. The extraction makes use of standard Python structures such as dictionaries in order to quickly find links in the design and establish references between the vertices (i.e. clusters or gates) and edges (i.e. nets). When building the hypergraph, some metadata are linked to each vertex/edge (e.g. area, power, total length, total connections) to later set their weights that will be needed in the partitioning phase.

As soon as the design has been converted into a hypergraph, it is formatted in such a way so that it can be fed to the hMETIS package. Using the weights sets extracted during DEF/LEF parsing phase, hMETIS partitions the graph. The output is processed into TCL directives that are then interpreted by a proprietary tool handling the 3D P&R, or they can be used to automatically partition the post-synthesis gate-level netlist and thus be used with any EDA flow, including the open-source ones.

Links

- hMETIS package developed by Karypis Lab: http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview
- Hypergraph extraction: https://github.com/parastuffs/metis_unicorn/blob/master/script/phoney.py
- DEF parser: https://github.com/parastuffs/def_parser