

**NOTE ABOUT
READY NODE GROUPS
IN SPACE**

S. de Graaf

Circuits and Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
The Netherlands

Report EWI-ENS 04-05
July 6, 2004

Copyright © 2004 by the author.

Last revision: August 5, 2004.

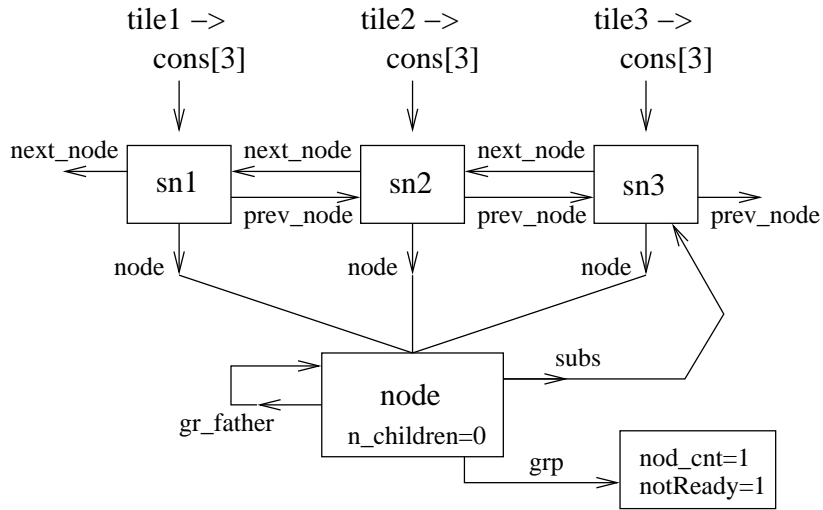
1. INTRODUCTION

For the network reduction heuristics in the *space* extraction program it is important to know what node groups are and when node groups become ready and which reduction heuristic must be done at which moment.

This note is written to investigate the implementation of the substrate resistance maximum parallel resistance reduction in the *space* program.

2. SUBNODES

The smallest layout part is a tile. Each tile can contain any number of mask combinations. Single masks or mask combinations can represent conductors. For each of this conductors the tile points to a subnode. Each subnode points to a node, which is allocated with subnodeNew. Note that the subnode data is contained in the tiles. No new node needs to be allocated, when adjacent tiles contain the same conductor. In that case function subnodeCopy* is used. The following figure shows three tiles and subnodes, which points to the same node. Note that the conductor number [3] is used as index.



As can be seen, the node above points also to a group data struct (with "grp"). When there are more nodes, only the group father node points to group data. In that case, the node child counter "n_children" is greater than 0.

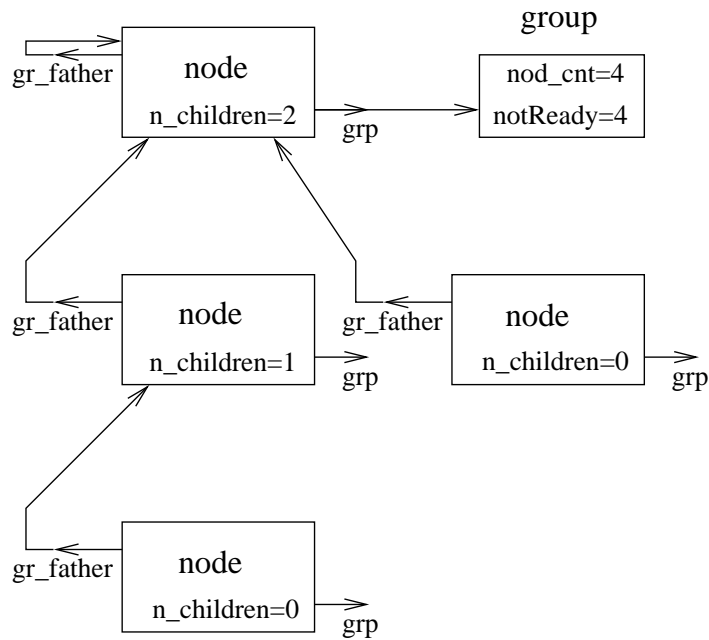
* Note that function subnodeJoin is used, when at later moment is detected that tiles must be joined. Note that by the special options **-K** and **-M** (optNoCore and optReadNetTerm) the subnodeCopy / Join not always take place. When the tiles have different cell instance number, they become / stay also different nodes.

3. NODE GROUPS

Nodes are placed in the same group (by mergeGrps), when there are conductance elements between different nodes (see elemAdd/elemNew). These conductance elements must be of type 'G'. Note that only the substrate conductance elements are of type 'S'. Thus, substrate nodes are not placed in the same group. Only by interconnect to substrate contacts they can be merged and become in the same group.

Thus, there are only node groups with two or more nodes, when interconnect resistances are extracted.

The following figure shows a node group:

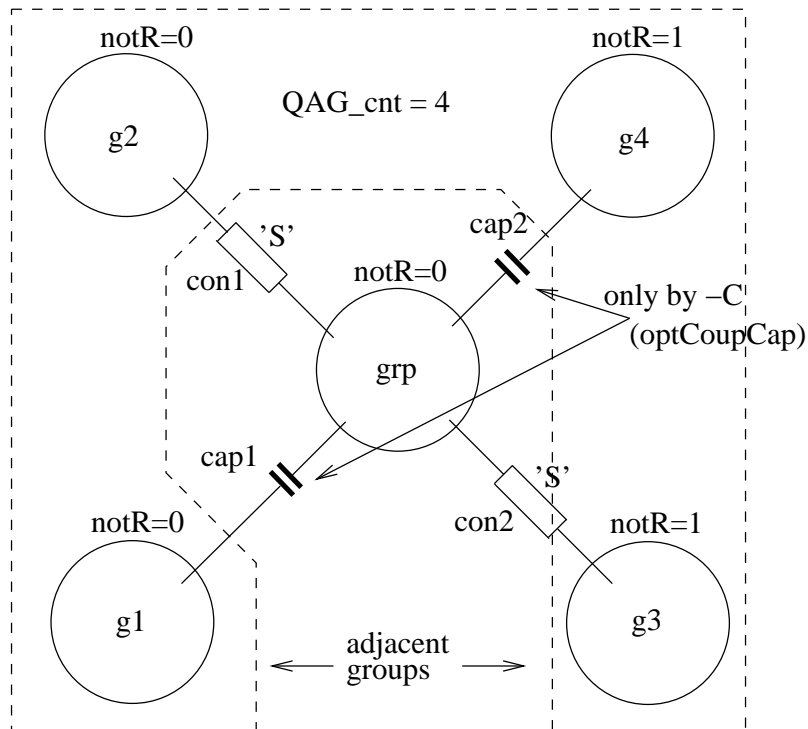


Macro Grp(node) finds the group data for a node in a node group. The macro calls function findRoot(node), when child node \rightarrow gr_father \rightarrow grp is null. This function compress the path for the child node to the root node.

4. ADJACENT GROUPS

Function `readyNode` calls function `readyGroup`, when flag `notReady` becomes zero. And function `readyNode` is called, when the last subnode of the node is ready. Function `readyGroup` shall reduce the ready group (calls `reducGroupI`). After that, `readyGroup` shall try to output (calling `outGroup`) the ready group. But that is only possible, when all adjacent groups are ready. An adjacent group has a capacitive or conductive element connection with the ready group. When all adjacent groups are ready, the ready group may be outputted. After that, the adjacent groups are checked and can be outputted, when their adjacents are all ready. Thus, it is possible, that a number of adjacent groups is outputted, but the current ready group not, because one (or more) adjacents are not ready. Note that nodes (which are outputted) write equivalence information to the by element connected nodes (which are not yet outputted).

For an example of adjacent groups, see the following figure:



The first step, `readyGroup` does, is to find nodes and the adjacent groups. One node of each adjacent group is stored in the QAG array. Variable `QAG_cnt` is the adjacent group counter. There are only adjacent groups, when substrate conductances or/and couple capacitances are extracted.

5. NETWORK REDUCTION

Function `readyGroup` shall reduce the ready group (calls `reducGroupI`), but the group is possible not yet outputted (calling `outGroup`). Note that function `outGroup` can also reduce the ready group (calls `reducGroupII` for `optCoupCap`).

Reductions inside a ready group are possible, but reductions between adjacent groups is another case. The question is, can we walk via connected substrate conductors to the other groups. When we look to the source code, we see that a group becomes only ready, when the substrate contact neighbor count becomes zero. As that is the case, function `subnodeDel` is called for the last subnode and `readyNode` is done. And besides that, function `clearTile` is possibly only done after some bandwidth in the scanning x direction.

6. APPENDICES

APPENDIX A -- Overview of source file changes

...