The xspef program

S. de Graaf

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

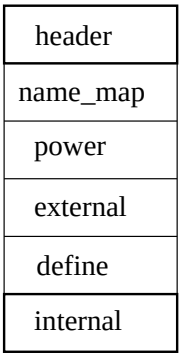
Report EWI-ENS 11-03
April 19, 2011

Copyright © 2011 by the author.

Last revision: April 20, 2011.

1. INTRODUCTION

The *xspef* program extracts a SPEF network description out of the Space circuit database. SPEF stands for "Standard Parasitic Exchange Format" and is an IEEE standard. See IEEE Std 1481-1999 (IEEE Standard for IC DPCS). IC stands for "Integrated Circuit" and DPCS for "Delay and Power Calculation System". SPEF is based on the Standard Parasitic Format (SPF) technology from Cadence Design Systems, but quite different.
In a SPEF file the following definitions can be found:



The <name_map>, <power>, <external> and <define> definitions are optional. Thus, a SPEF file contains at least a <header> and <internal> definition.
The <header> definition contains the following quoted(") string lines: SPEF_version, design_name, date, vendor, program_name, program_version, design_flow. And the following definition lines: hierarchy_divider, pin_delimiter, bus_delimiter, time_unit, cap_unit, res_unit, induc_unit.

The <name_map> definition can be used to define an index for names. An index is a positive integers (>= 0) with a asterix(*) prefix. This section is not used by *xspef*.

In the <power> definition two lines can be defined. One line for the net names which are "power_nets". And one line for net names which are "ground_nets". *Xspef* defines only the used ground net name (default GND).

The port names are specified in the <external> definition section. The *xspef* program does not add an instance name to the port names. It shall use direction 'B' (bidirectional), when no direction attribute is found.

The <define> section can be used to support hierarchical SPEF files. It is used for the instance names, to define which entities it are. An entity can be a sub circuit or a device name (everything not being a parasitic).

The *xspef* program obeys the escape rules using the backspace escape character before special characters found in identifiers (not alphanumerics or underscore and not being a divider or delimiter). Note that it shall also escape identifiers starting with a digit.

2. THE INTERNAL SECTION

The <internal> definition section is the main part of the SPEF file. It defines the nets with the parasitics and the connections. Four different net types can be used: D_NETs, R_NETs, D_PNETs and R_PNETs. But, the *xspef* program uses only D_NETs.

A D_NET contains distributed parasitic information for a logical net. The parasitic information is only available when extracted from the layout. A D_NET can contain 4 sections (CONN, CAP, RES and INDUC), as follows:

*D_NET netname totalcap
*CONN
*CAP
*RES
*INDUC
*END

The net name is an identifier (possibly with a bus postfix). When a D_NET contains a label or terminal (port), then that name is used. The net name is depended of the extraction parameters used. Other net names can also be generated by the *space* extractor. A net name can also be a number, when it is a completely internal net. This happens, when only instance pins are in the D_NET. Note that the *xspef* program does not calculate a totalcap value, but uses the value 0.

The CONN section (optional) can contain connect definitions. The section can contain "*P" lines for ports or "*I" lines for sub circuit instance pins which are in this D_NET. When the RES section is missing, then they are connected to each other without resistances. When there is a RES section, then all resistances between the pins and ports must be specified.

The CAP section (optional) can contain the capacitances of the D_NET. The first capacitance node is connected with a instance pin or port of the D_NET or else it must be connected with the D_NET netname followed by a netindex. The second capacitance node can be connected to ground, in that case it may be omitted. The second node can also be connected to a port or a instance pin of another D_NET or be connected to a netname with netindex of another D_NET.

The RES section (optional) contains all resistances of the D_NET. Both nodes must be in the D_NET or else it must be the power or ground netname. When two ports and/or pins are connected with each other a zero resistance value must be specified.

The INDUC section (optional) contains all inductances of the D_NET.

3. THE CIRCUIT DATABASE

Each circuit cell in the Space circuit database uses 3 files ("term", "mc" and "net") to store the circuit data. They contain the following information:

First, the "term" file, which contains all terminals (ports) of the circuit cell. A terminal can have a dimension (> 0), in which case an array (bus) is specified.

And two, the "mc" file, which contains all model calls of the circuit cell. A model can be a sub circuit cell, capacitor, resistor, transistor or other device. Each model has a model (cell) name and instance name and some attributes. For example, a capacitor and resistor has the 'v' attribute, which has a value (in Farad and Ohm). Note that a sub circuit instance can have a dimension (> 0), in which case an array of instances is used.

And three, the "net" file contains all nets (connectivity information) of the circuit cell. Each net has a net name or number and specifies one node of the circuit. A net has normally one of more sub nets, which specify what is connected with each other. A sub net can be a terminal or label (of the circuit itself) or a terminal (pin) of a model instance. Note that a terminal and/or instance can have a dimension. When a net does not have a sub net, then it is not connected. Such nets can be used to declare a bus terminal or label.

Each net written by the *space* program (the layout to circuit extractor) has a net attribute 'cd'. This attribute 'cd' stands for "conductor" and has a number > 0. The nets, which have the same "conductor" number, belong to the same node group. These nets are connected with each other by resistors. To get node groups, a resistance extraction must be performed.

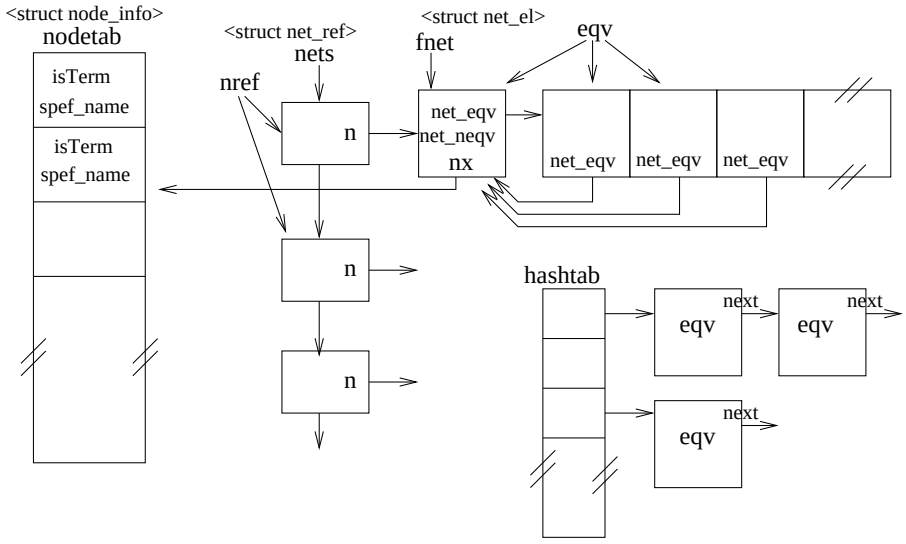
Note that the substrate net (with default the net name SUBSTR) has net attribute "cd=0". This net is as last or forelast net written to the "net" file. The last net written by *space* is the ground net (with default the net name GND). However, the ground net is only written when capacitances are extracted and are connected to the ground net.

Note that it is not wise to change defaults, but with parameter "name_ground" and "name_substrate" the names of the ground and substrate nets can be given another name. When one of the names become equal to another (sub) net name, then you can create accidental connections while net listing. The circuit database, however, does not specify these connections. Note that these connections can also be forced for the ground node with *xspef* options **-x**, **-y**, **-z** and **-O**. When the ground node is renamed, the *xspef* program does not know anymore what the name of the ground node is. In that case you can use option **-O** to identify the ground node.

Other connection problems can exist, when a label name is equal to a terminal name or when two or more terminals have equal names. The *xspef* program uses a net name find function which only can find one net for a name, and shall maybe find the wrong net. The same problem happens with sub circuit terminal pins which have equal names.

4. THE DATA STRUCTURES

The following data structures are used by the *xspef* program:



The "nodetab" table contains the "spef_name" which must be used as net name. Function findNet is used to find a net eqv in the hashtab. The first net eqv (fnet) is returned.