NAME

sim − format of .sim files read by esim, crystal, etc.

DESCRIPTION

The simulation tools *crystal* (1) and *esim* (1) accept a circuit description in **.sim** format. There is a single **.sim** file for the entire circuit, unlike Magic's *ext* (5) format in which there is a **.ext** file for every cell in a hierarchical design.

A **.sim** file consists of a series of lines, each of which begins with a key letter. The key letter beginning a line determines how the remainder of the line is interpreted. The following are the list of key letters understood.

| units: *s* **tech:** *tech* format: *MIT|LBL|SU*

If present, this must be the first line in the **.sim** file. It identifies the technology of this circuit as *tech* and gives a scale factor for units of linear dimension as *s*. All linear dimensions appearing in the **.sim** file are multiplied by *s* to give centimicrons. The format field signifies the sim variant. MIT and SU are compatible and understood by all tools. LBL is understood only by gemini(1).

type g s d l w x y **g=***gattrs* **s=***sattrs* **d=***dattrs*

Defines a transistor of type *type*. Currently, *type* may be **e** or **d** for NMOS, or **p** or **n** for CMOS. The name of the node to which the gate, source, and drain of the transistor are connected are given by *g*, *s*, and *d* respectively. The length and width of the transistor are *l* and *w*. The next two tokens, *x* and *y*, are optional. If present, they give the location of a point inside the gate region of the transistor. The last three tokens are the attribute lists for the transistor gate, source, and drain. If no attributes are present for a particular terminal, the corresponding attribute list may be absent (i.e, there may be no **g=** field at all). The attribute lists *gattrs*, etc. are comma-separated lists of labels. The label names should not include any spaces, although some tools can accept label names with spaces if they are enclosed in double quotes. **In version 6.4.5 and later** the default format produced by ext2sim is SU. In this format the attribute of the gate starting with S_ is the substrate node of the fet. The attributes of the gate, and source and substrate starting with A_, P_ are the area and perimeter (summed for that node only once) of the source and drain respectively. This addition to the format is backwards compatible.

C *n1 n2 cap*

Defines a capacitor between nodes *n1* and *n2*. The value of the capacitor is *cap* femtofarads. **NOTE:** since many analysis tools compute transistor gate capacitance themselves from the transistor's area and perimeter, the capacitance between a node and substrate (GND!) normally does not include the capacitance from transistor gates connected to that node. If the **.sim** file was produced by *ext2sim* (1), check the technology file that was used to produce the original **.ext** files to see whether transistor gate capacitance is included or excluded; see "Magic Maintainer's Manual #2: The Technology File" for details.

R *node res*

Defines the lumped resistance of node *node* to be *res* ohms. This construct is only interpreted by a few programs.

r *node1 node2 res*

Defines an explicit resistor between nodes *node1* and *node2* of resistance *res* ohms. This construct is only interpreted by a few programs.

N *node darea dperim parea pperim marea mperim*

As an alternative to computed capacitances, some tools expect the total perimeter and area of the polysilicon, diffusion, and metal in each node to be reported in the **.sim** file. The **N** construct associates diffusion area *darea* (in square centimicrons) and diffusion perimeter *dperim* (in centimicrons) with node *node*, polysilicon area *parea* and perimeter *pperim*, and metal area *marea* and perimeter *mperim*. *This construct is technology dependent and obsolete.*

A *node attr*

Associates attribute *attr* for node *node*. The string *attr* should contain no blanks.

= *node1 node2*

Each node in a **.sim** file is named implicitly by having it appear in a transistor definition. All node names appearing in a **.sim** file are assumed to be distinct. Some tools, such as *esim* (1), recognize aliases for node names. The **=** construct allows the name *node2* to be defined as an alias for the name *node1*. Aliases defined by means of this construct may not appear anywhere else in the **.sim** file.

SEE ALSO

crystal (1), esim (1), ext2sim (1), sim2spice (1), ext (5)