NAME

simfile − .sim file file format

DESCRIPTION

The **.sim** files are ASCII files used by various programs to describe MOS transistor networks and their associated parameters. *Mextra* and *netlist* output **.sim** files; *sim2spice*, *presim* and others read the files.

Each line of a **.sim** file is a separate "record" whose type is determined by the first character of the line. The possible record types are described in this section.

| ...

Lines beginning with vertical bar (|) are treated as comments and ignored by programs that read **.sim** files.

A only exception is if this line contains the information units:, tech: and format:. Units specifies the conversion factor to centimicrons, tech declares the technology (e.g. nmos) that was used in the design and format declares the **.sim** file format used for the various other records.

The format specifier is required as there are currently two slightly different formats used. One is used by tools from Berkeley and the other by tools from MIT. If a format isn't declared in one of these intrepreted comments the original file format used by the tool is assumed.

The only program that uses a **.sim** file for input that doesn't support both formats is the *erc*. This is because it is heavily dependent on the circuit extractor *mextra*.

Other than the record formats the main cause for concern are the units used in transistor sizes and capacitive loads. MIT generally uses a lambda value for transistor size and picofarad loads, UCB uses centimicron transistor sizes and femtofarad loads.

@ *filename*

Redirects input from the named file. When the end-of-file is reached, input reverts to the current file at the following line. Indirect files can be nested; usually there is some system dependent limit on the number of simultaneously open files which limits the depth of nesting.

The following record is format dependent;

MIT format:

e *gate source drain length width xpos ypos [rpa] area*

i *gate source drain length width xpos ypos [rpa] area*

d *gate source drain length width xpos ypos [rpa] area*

p *gate source drain length width xpos ypos [rpa] area*

l *gate source drain length width xpos ypos [rpa] area*

UCB format:

e *gate source drain length width xpos ypos [g=att s=att d=att]*

n *gate source drain length width xpos ypos [g=att s=att d=att]*

p *gate source drain length width xpos ypos [g=att s=att d=att]*

d *gate source drain length width xpos ypos [g=att s=att d=att]*

u *gate source drain length width xpos ypos [g=att s=att d=att]*

Above are the possible transistor records. Note that the **n** type record is output only by the *magic*'s extractor. The first character tells the type of the transistor:

MIT format:

e n-channel enhancement

i n-channel zero threshold (intrinsic) enhancement

p p-channel enhancement

d depletion

l low-power depletion

UCB format:

e n-channel enhancement

n n-channel enhancement (from ‘magic’ output)

p p-channel enhancement

d depletion

u unusual implant

The next three parameters are the names of the *gate, source,* and *drain* nodes.

The *length* and *width* can be either integers or floating-point giving the dimensions of the active transistor area.

xpos and *ypos* report the coordinates of the upper left hand corner of the transistor (NETLIST always specifies 0,0).

The next parameter (MIT only) is a letter specifying the shape of the transistor:

r rectangular

p rectangular, monotonically increasing width

a other shapes...

Netlist always puts an "r" in this field.

area (MIT only) tells the true active area; may be different from width∗length if the network extractor used an approximation (in output from *netlist*, area always equals width∗length).

G=att, s=att and d=att (UCB only) are optional transistor attributes used by *crystal* a timing verifier.

The following record is format dependent;

MIT format:

N *node xpos ypos MA PA DA DP*

UCB format:

N *node DA DP PA PP MA MP*

L *node M2A M2P MA MP PA PP DA DP PDA PDP*

Node record from the network extractor. Tells the node name, coordinates of the upper left corner of the node and various geometrical info described below.

MIT format only;

M *node xpos ypos M2A M2P MA MP PA PP DA DP PDA PDP*

This record is similar to (N) in a format (MIT only) that includes more information.

xpos and *ypos* give the coordinates of the upper left corner of the node and following provide geometrical information:

M2A area of 2nd-level metal

M2P perimeter of 2nd-level metal

MA area of 1st-level metal

MP perimeter of 1st-level metal

PA area of polysilicon

PP perimeter of polysilicon

DA area of n-diffusion

DP perimeter of n-diffusion

PDA area of p-diffusion

PDP perimeter of p-diffusion

Presim and *sim2spice* can use these numbers to compute interconnect capacitances from layout information; *netlist* never produces this type of record. Note that not all programs use all of these values for computing the lumped capacitor value. The perimeter measures are the actual total perimeters.

The following record is format dependent.

MIT format:

c *node cap*

UCB format:

C *node1 node2 cap*

Used to specify *node* capacitance in pf. *cap* can be either an integer or floating-point number. This is the type of record used by *netlist* to described user-specified capacitors. Node2 is assumed GND in the MIT format.

= *node1 node2 ...*

Used to establish a name equivalence. Indicates that two or more names refer to the same electrical node. Parameters for the specified nodes are merged; the user can subsequently use the names interchangeably.

The following are MIT records and have no UCB equivalent.

D *node low-to-high high-to-low*

Describes user-specified delays for the given node. Delays are integral numbers of RNL time units (1/10th nanosecond).

t *node low high*

User specified node thresold. Describes user-specified voltage thresholds in normalized voltage units (i.e., low and high are floating-point numbers in the range 0.0 to 1.0).

The following is a UCB record and has no MIT equivalent.

A *node attribute*

This is the node equivalent of the transistor attribute mentioned earlier and is also used by the timing verifier *crystal*.

The following record for the resistance value is output only by the *magic*'s circuit extractor and hence does not have an MIT equivalent.

R *node1 node2 resistance*

SEE ALSO

mextra(CAD1)
presim(1.VLSI)
netlist(1.VLSI)
sim2spice(CAD1)
pext(1.VLSI)
nlc(1.VLSI)