Program: Static of December 19, 1982

Maintainer: Thacker.pa

Location: [Indigo]<chipmonk>static>Static.bcd --code
[Indigo]<chipmonk>static>Static.bravo, .press --this memo

Purpose: This program does some static validity checks on an IC layout produced using Chipmonk.

Input: Text file "name.sim", produced by the Chipmonk circuit extractor, plus some parameters from the user.

Output: Text file "name.erclog". This file is in Chipmonk error format, allowing it to be read using CTRL-TAB-E and TAB-E. As each line is read, the screen is scrolled to the coordinates given at the beginning of each error line (note that some errors do not have coordinates, and therefore do not scroll the screen).

Description: When started, the program requests the following information from the user:

u1) The name of the .sim file (without extension).

u2) The low limit for gate pullup/pulldown ratios. For NOR gates, this ratio is (Ldep/Wdep)/(Lenh/Wenh). When NAND structures are present, the enhancement term is the sum of the impedances (L/W) of the series-connected enhancement transistors. When a structure with ratio less than the low limit is encountered, an error is output containing the name and location of the pullup, and the name and location of all the pulldowns on subsequent lines.

u3) The high limit for gate pullup/pullup ratios. When a structure with a ratio greater than this limit is encountered, a message similar to that described in (u2) is output.

u4) The L/W multiplier for input transistors. Static assumes that nodes that include "lightning arresters" (enhancement transistors with their gate and source grounded) are input nodes of the chip. During ratio checking, the impedance of all transistors driven by an input node is multiplied by this factor. The factor is shown in parentheses in the error file. The usual value is 2.0.

u5) The L/W multiplier for transistors driven by indirectly pulled up nodes. This factor is applied to transistors whose gates are driven by passgates, as with (d). The usual value is 2.0.

When the information above has been supplied, the program does a number of checks, each of which produces the output messages shown below:

c1) During the reading of the input file, depletion mode transistors that do not have their source or drain connected to vdd are either ignored or converted to something else. The cases are:

> c1a) Depletion capacitor: the transistor has source connected to drain, and gate connected to something else. The transistor is counted as a FunnyDTrans, but is otherwise ignored. The message is:
>
> `x y depletion capacitor - ignored: d GateNode DrainNode SourceNode`
>
> c1b) Depletion resistor: the transistor has gate connected to source or drain. It is converted to an enhancement transistor with gate connected to vdd, and counted as a FunnyDTrans. The message is:
>
> `x y depletion resistor: d GateNode DrainNode SourceNode`
>
> c1c) Cases other than (c1a) and (c1b) are "yellow transistors", and are also converted to enhancement transistors with gate connected to vdd, and counted as FunnyDTrans's. The message is:

x y yellow transistor: d *GateNode DrainNode SourceNode*

c2) If both the source and drain of a depletion transistor are connected to vdd, the message is:

x y depletion error: d *GateNode DrainNode SourceNode*

c3) If an enhancement transistor is found with gate = vdd, the message is:

x y Gate is VDD: e *GateNode DrainNode SourceNode*

c4) If an enhancement transistor is found with gate = gnd and neither drain nor source = gnd, the message is:

x y Gate is GND: e *GateNode DrainNode SourceNode*

c5) If an enhancement transistor is found with gate = source or gate = drain, the message is:

x y gate and source or drain equality: e *GateNode DrainNode SourceNode*

When the entire input file has been read a summary of the number of nodes, enhancement devices (ETrans), normal depletion devices (DTrans), and abnormal depletion devices (FunnyDTrans) is output (without coordinates). The read phase of the program also puts nodes into "equivalence classes", and associates all enhancement transistors with exactly one class. An equivalence class is a set of nodes that would be connected together if all gates of the transistors associated with the class were turned on. Subsequent checks use this data structure.

c6) The program checks that each node that is pulled up by a depletion transistor is pulled up by no more than one such transistor. If this is not the case, the message is (note that no coordinates are output):

Node pulled up more than once: *NodeName*

c7) The program identifies normal pullups (gate = drain, source = vdd), inverting superbuffers, and noninverting superbuffers. If a depletion transistor cannot be identified as part of one of these structures, the message is:

x y depletion transitor not a pullup or a superbuffer: d *GateNode DrainNode SourceNode*

After this phase, the program outputs a summary of the pullups, inverting and noninverting superbuffers, and pullups it found.

c8) The program identifies lightning-arrested nodes as inputs, and tells you about each of them (the coordinates are those of the lightning arrester transistor):

x y Assuming lightning arrested node is an Input: d *GateNode DrainNode SourceNode*

c9) The program then checks that all nodes can be pulled up or pulled down. Possible error messages are:

Node name only occurs once: *NodeName*
Node can never be given a value: *NodeName*
Node can never be set to 1: *NodeName*
Node can never be set to 0: *NodeName*

C10) The program also checks that there are no situations in which a passgate drives another passgate. This is a fairly restrictive test (it finds bootstraps, for instance), so it is not necessarily an error. The message is:

x y Pass transistor driven by pass transistor: *GateNode DrainNode SourceNode*

The indicated transistor is the driven one, not the driver.

Finally, the program checks that all gates have ratios between the user-supplied limits. When errors are found, the message is:

```
x y Pullup/pulldown ratio = r: Node NodeName pulled up thru Length by Width
x y pulled down by node NodeName thru Length by Width (x Factor) to node NodeName
x y pulled down by node NodeName thru Length by Width (x Factor) to node NodeName
```

When the structure is a NOR, there will be only one "pulled down . . " line. When it is a NAND, there will be one line for each series pulldown. Factor is the number given in U4 or U5, if appropriate, else 1.0.