NAME

match - a network comparison program

SYNOPSIS

match [options] nominal_network actual_network

PARAMETERS

nominal_network
Name of the nominal network

actual_network
Name of the actual network

OPTIONS

Options are:

-help Print this list.

-release Print the release number of this tool.

-bindings
Print a binding table on the stdout. Only nets and terminals are listed.

-Bindings
Print a full binding table on the stdout. All matched devices are listed as well.

-Printer
Line printer compatible output. The VT100 compatible graphical characters used to print the binding table are replaced by ordinary ascii characters. This option is also set by the -file option.

-List Print a listing of both networks on the stdout.

-Expand
Print a listing of both networks on the stdout after they have been expanded.

-Silent Silent mode. Only error messages are printed.

-Verbose
Verbose mode. Print additional process information on stderr.

-byname
Match terminals, nets and devices by name before starting the actual network comparison. Elements having names starting with an underscore ('_') or a number ('0'-'9') are not considered in this stage. This option can be used to assert equivalences between network elements via their names.

-capacitors
Ignore capacitors in both network descriptions. Capacitors are simply removed.

-caprange *value*
Specify the allowable relative difference between two capacitors. This option is equivalent to *-range cap:v,value*. *v* is the name of the parameter to which the size of a capacitor is assigned in the database.

-edif Produce output in an edif-like format that can be used as input for the program *highlay(1ICD).*

-files Specifies that *nominal_network* and *actual_network* are the names of files that contain complete sls network descriptions. The last network in each file is the one that needs to be processed.

-output *file*
Redirect program output to the specified file name. In addition the option -Printer is set.

-primitive *file*
Specify a primitive element description file. The terminal permutabilities of primitive elements that are specified in this file overrule the default terminal permutability settings. In addition, new

primitive elements can be defined.

-range *dev*:*par*,*value*
: Specify the allowable relative difference between parameter values of two devices. *dev* is the name of the device type, e.g. 'nenh', 'penh', or 'res'. *par* is the name of the parameter, e.g. 'w' for 'nenh' or 'v' for 'res'. 'v' is used as the parameter name for unnamed device values such as resistor values or capacitor values. *value* is the allowable relative difference. If *nom* is the parameter value of one device, *act* the parameter value of another device and *val* the relative difference, then the two devices match with respect to their parameter values if the following (symmetric) condition is satisfied:

nom <= (1 + val) * act AND act <= (1 + val) * nom

-resistors
: Ignore resistors in both network descriptions. Resistors are replaced by a connection.

-resrange *value*
: Specify the allowable relative difference between two resistors. This option is equivalent to *-range res:v,value*. *v* is the name of the parameter to which the size of a resistor is assigned in the database.

-transistors
: Only compare transistors (equivalent to -cap -res).

Options may be abbreviated.

DESCRIPTION

Match is a program that compares two networks. The program is designed to be used as a connectivity verification tool for VLSI circuits.

The program expects two input networks which by default will be read from the database. The networks will be expanded before the matching process is started.

The two networks are:

The **nominal** *network.*
: This (hierarchical) network is used as a reference for the layout connectivity. The nominal network description can be obtained from a schematic editor or can be made by hand. In a top down design process, this description is usually already available as a result of an earlier design step (e.g. simulator input file).

The **actual** *network.*
: This (hierarchical) network is usually directly derived from the circuit layout using an extractor program.

The comparison process is completely independent from the individual labelings of the nets (terminals) and devices in both network descriptions. A graph theoretical partitioning algorithm is used to derive a possible isomorphic equivalence of both networks. If the two networks are found not to be isomorphic, *match* tries to determine which parts of the two networks are equivalent and which parts are not.

The terminal permutabilities of the primitive network elements (that is, the way in which the terminals can be interchanged without altering the topology of the network) can be defined in a user-specified primitive element file. For this purpose, the sls language is extended as follows:

Network definitions preceded by the keyword **primitive** consist of exactly one permutability definition followed by a semicolon. A permutability definition consists of the keyword **perm** followed by a number of classes enclosed by curly brackets. Each class contains a set of terminals which may be mutually interchanged. Terminals that are not part of the same class are not interchangeable.

As an illustration, the default permutability definition of an NMOS enhancement transistor is given.

```
primitive network nenh (terminal g, d, s)
{
```

```
    perm { (g) (d, s) };
}
```

A standard primitive element file defines the default permutabilities of all primitive network elements.

After the comparison, the result is printed on the output. A binding table, which specifies which nets (terminals) and devices from both network descriptions are to be associated is available as well as a list of 'unmatched' nets (terminals) and devices in case the two networks are different.

EXAMPLES

Compare two networks from the database and send binding information to a file *outp*:

```
match -Bindings netw my_netw -out outp
```

Compare two networks that are present in an sls file and ignore capacitors in both networks:

```
match -cap -files handshake.sls flatshake.sls
```

REMARKS

Match considers terminals and nets as different elements. It is thus not possible that an (internal) net of one network matches with a terminal of the other.

Before starting the network comparison, *match* removes all unconnected nets and terminals (including feedthroughs) from both descriptions and issues a warning about them.

Match compares small and medium sized networks so fast that it takes more time to read (and expand) the networks than to compare them.

DIAGNOSTICS

Syntax errors in the sls input specifications and references to undefined networks are reported. Recursive network definitions are detected as well.

All other error messages are self-explanatory.

RETURN STATUS

The program returns the following status:

0 = match succeeded.
1 = error.
2 = match failed.

AUTHOR

T. Vogel, Delft University of Technology.

FILES

`ICDPATH/share/lib/process/`*process*`/match_prim`
(standard primitive element file)

SEE ALSO

T. Vogel, "Connectivity Verification based on Netlist Comparison", Delft University of Technology.
Sls: Switch-Level Simulator User's Manual.