Using Nondefault Rules

This document presents orientations on how to use nondefault rules (NDRs) that are supported by TritonRoute (TR). Thus, the features of NDRs presented here are the ones that are currently supported in TR, but other features will be implemented latter, and some of the existing will be improved (and also need more testing).

In order to use NDRs you must manually write them in the lef and def files.

NDRs in LEF Files

NDRS are defined in the LEF files. NDRs should be defined after the layers and vias definition. NDR definition take the following form:

[NONDEFAULTRULE *ruleName*
    {LAYER *layerName*
      WIDTH *width* ;
      [SPACING *minSpacing* ;]
    END *layerName*} ...
    [USEVIA *viaName* ;] ...
END *ruleName*]

Explanations of the keywords:

LAYER *layerName* ... END *layerName*

Specifies the constraints of a routing layer. Every routing layer must have a WIDTH keyword and value specified. All other keywords are optional.

SPACING *minSpacing*

Specifies the minimum spacing required by router-created shapes, using this NDR. If the spacing is given, it must be at least as large as the foundry minimum spacing rules defined in the LAYER definitions.

Type: Float, specified in microns

Routing with 3x default spacing seems stable at the moment. Routing with 5x spacing may be too slow, and possibly leaving violations. This is being improved.

This spacing constraint is by default a soft spacing, meaning that there will be no violation if it is not respected (but the router should try to respect it). However, at the moment, TR treats this spacing as hard spacing, meaning that it has obligation to respect the spacing.

USEVIA *viaName*

Specifies a previously defined via from the LEF VIA statement, that should be used by the NDR.

Using large vias needs more testing at this moment.

WIDTH *width*

Specifies the required minimum width for *layerName*.
Type: Float, specified in microns

Using width higher than default width needs more testing at the moment. Also, higher width wires may present violations when accessing the pins. This could be solved by using taper rules. However, since taper rules are not currently supported, using higher width wires require to not define NDRs for Metal1, since it is very likely that they will cause violations.

Example of a NDR called NDR_1W_3S, with wire width equal the default width and spacing equal to 3x default spacing:

```
NONDEFAULTRULE NDR_1W_3S
        LAYER Metal1
                WIDTH 0.06 ;
                SPACING 0.18 ;
        END Metal1
        LAYER Metal2
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal2
        LAYER Metal3
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal3
        LAYER Metal4
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal4
        LAYER Metal5
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal5
        LAYER Metal6
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal6
        LAYER Metal7
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal7
        LAYER Metal8
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal8
        LAYER Metal9
                WIDTH 0.07 ;
                SPACING 0.21 ;
        END Metal9
END NDR_1W_3S
```

NDRs in DEF Files

NDRs can also be defined in def files. The syntax is the following:

NONDEFAULTRULES *numRules* ;
    {- *ruleName*
        {+ LAYER *layerName*
            WIDTH *minWidth*
            [SPACING *minSpacing*]
        } ...
        [+ VIA *viaName*] ...
    ;} ...
END NONDEFAULTRULES

Here you can define multiple NDRs inside the NONDEFAULTRULES statement.

The meanings of the Keywords are the same from LEF. In the case of conflict of information between LEF and DEF, DEF info has priority.

Assigning a NDR to a net

To assign a NDR to a net, it is necessary to add + NONDEFAULTRULE <NDR_name> in the net definition.

Example of a net, named net1, with 2 pins (B from inst1689 and A from inst1989), with the NDR defined in the previous example:

```
- net1
  ( inst1689 B ) ( inst1989 A )
  + NONDEFAULTRULE NDR_1W_3S
 ;
```

For more information on NDRs, please see http://free-online-ebooks.appspot.com/enc/14.17/lefdefref/LEFSyntax.html#NondefaultRule.