**Project-2 (end-of-line rule)** (10%)

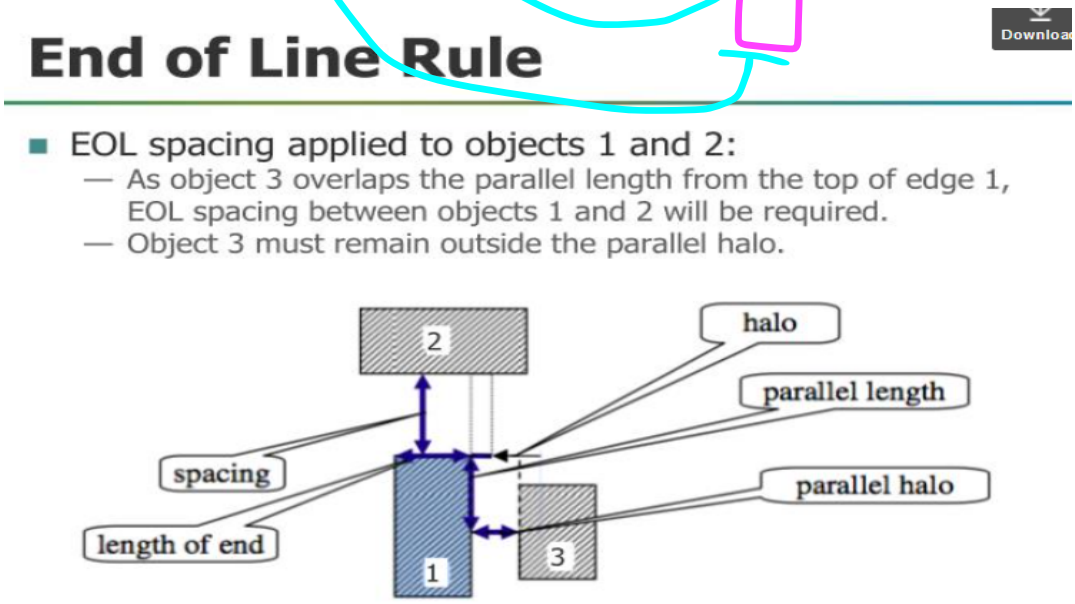
Please see a picture below, which describes this rule. Let's say

EOL spacing between objects 1 and 2 is at least 1 (um)

Length of end is 0.6 (um) (i.e., if the width of object-1 is smaller & equal than 0.6, consider.)

Parallel length from the top of edge 1 is 0.7 (um).

Parallel halo away from object-1 is 0.4 (um)




Another simpler picture for EOL is below:

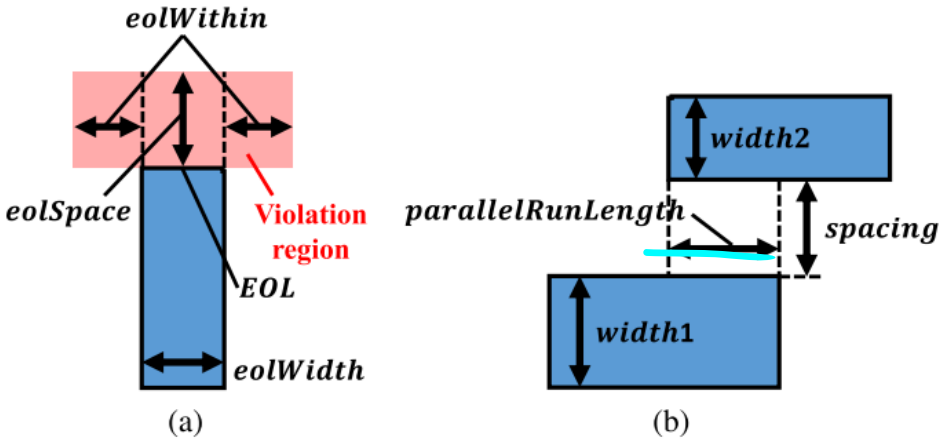
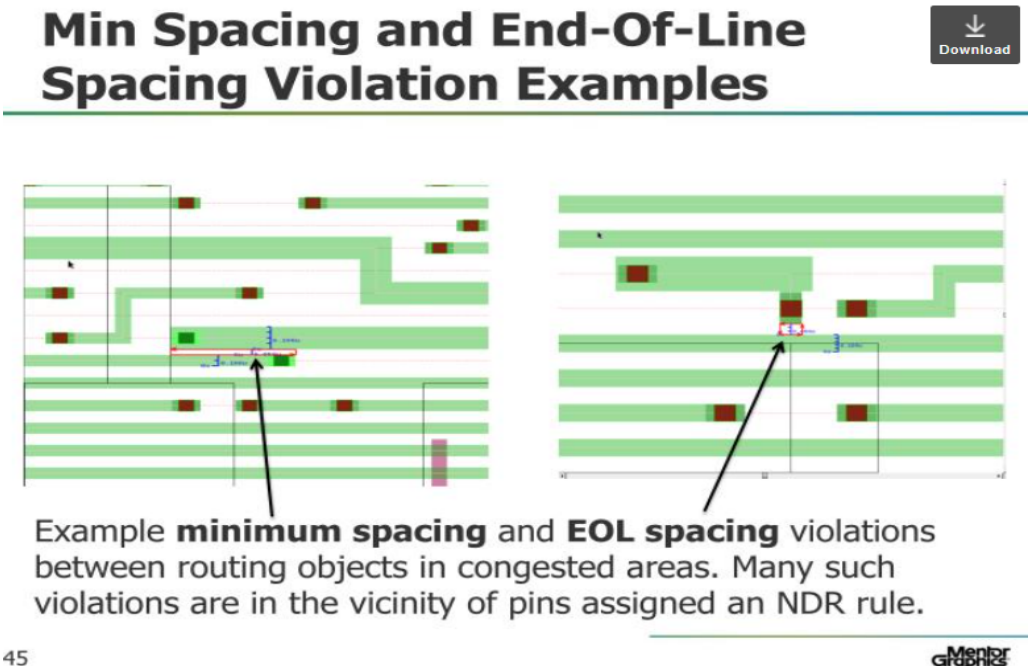



Fig. 2. Example of (a) EOL spacing and (b) parallel-run spacing.

EOL Spacing: A metal end is an EOL if its width is shorter than eolWidth. EOL is required to preserve a spacing greater than or equal to eolSpace beyond the EOL anywhere less than the eolWithin distance, as Fig. 2(a) shows

Below is a layout sample which has an end-of-line error.




In this project, please write down your algorithms for this checking. Then, implement it to read the following layout patterns (using the format defined in project-1) to see if there are EOL errors. In this exercise, it is totally up to you if you want to use Boost APIs.

Layout patterns are described below.

number_of_layers 1

layer 1 M1

end_of_layer

number_of_rectangles 6

1 0 4.0, 7.8 0.6

1 0 1.0, 6.1 1.6

1 2.0 2.0, 4.0 2.6

1 7.0 1.0, 14 1.6

1 10.4 0.8, 11 1.8

1 8.6 2.2, 14 2.8

end_of_rectangle

Please read in this file to build your data structure, then, check if there is/are end-of-line DRC errors.

The layout looks like below: