Well Proximity and STI Stress Effect Parameters Extraction in SmartDRC/LVS

Introduction

From 90nm technology process and deeper, well proximity and shallow trench isolation (STI) stress effects become more significant in alteration of MOS device characteristics. Accurate post-layout SPICE simulation cannot be done without taking these effects into account. New SPICE device models, like BSIM4, have parameters for the simulation of well proximity and STI stress effects.

Silvaco's SmartDRC/LVS tool provides special functions for well proximity and STI stress effects parameter calculation that can be used in post-layout SPICE simulation. There are three functions that can be used in PWRL files for generic devices for this purpose:

- **enclosure_distance()**
- **enclosure_ distance_90d()** – for the perpendicular
- **enclosure_ distance_0d()** – for the parallel case

The function that should be selected depends on the particular needs of the user.

Enclosure Distance

Well proximity and stress effects parameter calculations can be done via enclosure distance. SmartDRC/LVS provides PWRL functions that permit the calculation of the enclosure distance. They have the following syntax:

- **enclosure_distance** (active , %MAX_CHECK_VALUE);
- **enclosure_ distance_90d**(device_layer, target_layer, S_or_D_layer, %MAX_DISTANCE_VALUE);
- **enclosure_ distance_0d**(device_layer, target_layer, S_or_D_layer, %MAX_DISTANCE_VALUE);

The input layers must be traditional polygonal. The layer for enclosure_distance must be either a pin layer or an auxiliary layer. For other functions, all layers must be either device layers, or pin layers, or aux layers.

It is recommended to avoid large values of the maximal check value (e.g. 5u).

The first function (enclosure_distance) calculates the enclosure vector for specified measurement layer <active> over device seed shape. The device seed shape must satisfy the following conditions:

- be rectangular;
- be completely overlapped by measurement layer polygon;
- two opposite sides of the seed shape must coincide with edges on measurement layer.

For enclosure_distance_0d() and enclosure_distance_90d(), functions the device seed shape must satisfy the following conditions:

- be rectangular;
- DRC must be clean.

A call of any of the three functions enclosure_distance(), enclosure_distance_0d() and enclosure_distance_90d() results in a variable in a vector format, which in fact represents two arrays (a and b).

LVS extraction needs median values of those two functions. The median values are obtained with median function of the device formula calculator:

median*(variable_name, function_name)*

that returns the median value for the entire definition domain, e.g.:

s = enclosure_distance_0d(l1, l2, l3, 5.0)
sa = median(s, a);
sb = median(s, b)

Four distances from those functions (up, down, left, right of the median) are used to calculate SCA, SCB, and SCC. The function enclosure_distance90d may be used for calculation of both the parameters SA and SB of stress effect, and parameters of well proximity effect. In the latter case there are more parameters and the calculation is much more complicated.

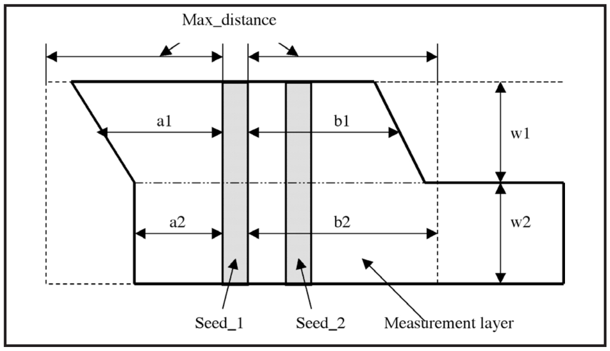

Figure 1. Enclosure distance function for device shape Seed_1. The measurement layer is split into two trapezoids, so enclosure vector has two triplets (a1, b1, w1) and (a2, b2, w2). Value b2 is defined by MAX_DISTANCE_VALUE. The device shape, Seed_2 will have another enclosure vector.

The result may be placed into a variable, e.g. q = enclosure_distance_0d(...), but the variable may further be used only either as a parameter of median(q, ...), or to define another variable like p = q, or in a conditional like s = (condition) ? p : q;

The measurement layer polygon is split into trapezoids with sides that are parallel to the coincident edges of the layer and device seed shape. Each trapezoid provides three values (a, b, and w) for the enclosure distance element (Figure 1). The max_check_value parameter defines the size of the search window and so defines the maximum value for a and b.

If the measured layer covers several device seed shapes, the enclosure distance is calculated for each seed shape separately. During the enclosure vector calculation, all other device seed shapes are ignored. If the device seed shape doesn't satisfy the required conditions, the a, b, and w values are set to -1

Well Proximity Enclosure Function

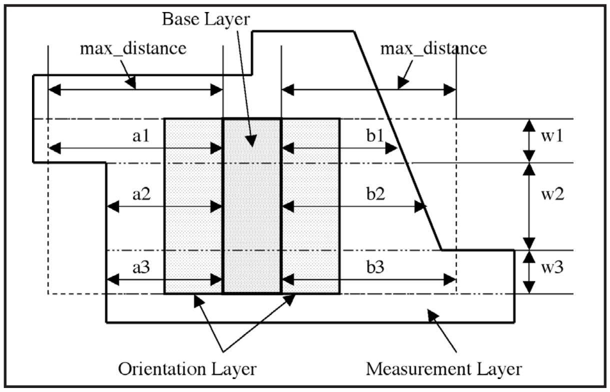

Figure 2. Enclosure_distance_90d function with oriental layer as argument in PERPENDICULAR direction. The overlapped part of the measurement layer is split into three trapezoids, so the output enclosure distance has three triplets (a1, b1, w1), (a2, b2, w2), and (a3, b3, w3). Values a1 and b3 are defined by MAX_DISTANCE_VALUE.

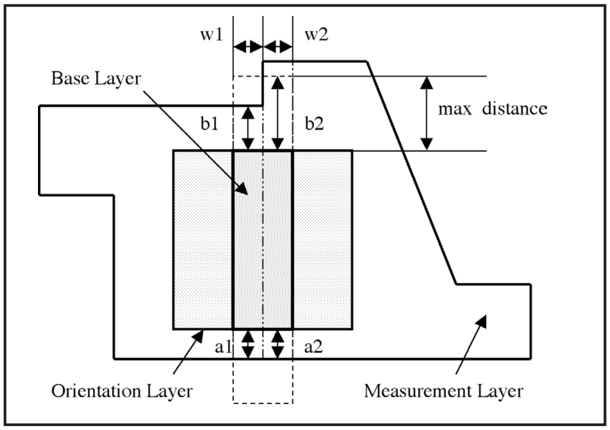

Figure 3. Enclosure_distance_0d function with oriental layer as argument in PARALLEL direction. The overlapped part of measurement layer polygon is split into two trapezoids, so the output enclosure distance has two triplets (a1, b1, w1) and (a2, b2, w2). Value b2 is defined by MAX_DISTANCE_VALUE.

Example

The following example shows how to use the enclosure functions in a user defined PWRL procedure, namely:

- Calculation of stress effect parameters by two variables (an equivalent of ENCLOSURE_VECTOR function) ***s = enclosure_distance (dif , 50);***
- Calculation of stress effect parameters by 4 variables (an equivalent of ENCLOSURE_PERPENDICULAR function) ***P = enclosure_distance_90d (gate, dif, t_sd, 50);***
- Calculation of SC1, SC2, SC3, and SC4 distances for SCA, SCB, and SCC parameters of well proximity effect model.

```
:
$device MOS(mos3) gate gate t_sd t_sd bulk <perim>
<dif> <ASDD> <D1> <NG> <G1> <seed> <well>
$spice_model "name_model"
[
$property W L AS AD PS PD AS1 AD1 SCA SCB
  SCBW1 SCBW2 SCBL1 SCBL2 SCCW1 SCCW2
  SCCL1 SCCL2 SCC SC1 SC2 SC3 SC4 s(u) sc sd ;

s = enclosure_distance (dif , 50); // in microns, 1e-6;
P = enclosure_distance_90d (gate, dif , t_sd, 50);
sc =1e-6 * median(P, a);
sd =1e-6 * median(P, b);


S1 = enclosure_distance_0d (gate, well, t_sd, 10);
Q = enclosure_distance_90d (gate, well, t_sd, 10);
SC2 =1e-6 * median(S1, a);
SC4 =1e-6 * median(S1, b);
SC1 =1e-6 * median(Q, a);
SC3 =1e-6 * median(Q, b);
```

Notes.

- For the function s = enclosure_distance (dif , 50); s(u) is written into device property. As a result, we get sa and sb values in microns.
- 2. To avoid confusion with the same names of the output parameters (sa and sb) when the function enclosure_distance is calculated for both cases of two and four variables, the output parameters of enclosure_distance_90d are named sc and sd. As the test case shows, their values are equal pair-wise: sa=1.720u sb=7.280u sc=7.280u sd=1.720u

Next, based on those results, complicated equations for calculation of SCA, SCB, and SCC parameters are written as follows:

…

SCREF =1e-6*0.95;

SCA = 1 / (W * L) * (SCREF * SCREF) * (W * (1 / SC1 - 1 / (SC1 + L)) + L * (1 / SC2 – 1 / (SC2 + W)) + W * (1 / (SC3 + L)) + L * (1 / SC4 – 1 / (SC4 + W)));

SCBW1 = 1 / (W * SCREF) * (SCREF / 10 * SC2* exp(-10 * SC2 / SCREF) + SCREF *SCREF / 100 * exp(-10 * SC2 / SCREF) - SCREF / 10 * (SC2 + W) * exp(-10 * (SC2 + W) / SCREF) - SCREF * SCREF / 100 * exp(-10 * (SC2 + W) / SCREF));

SCBW2 = 1 / (W * SCREF) * (SCREF / 10 * SC4 * exp(-10 * SC4 / SCREF) + SCREF * SCREF / 100 * exp(-10 * SC4 / SCREF) - SCREF / 10 * (SC4 + W) * exp(-10 * (SC4 + W) / SCREF) - SCREF * SCREF / 100 * exp(-10 * (SC4 + W) / SCREF));

SCBL1 = 1 / (L * SCREF) * (SCREF/10*SC1* exp(-10 * SC1 / SCREF) + SCREF * SCREF / 100 * exp(-10 * SC1 / SCREF) - SCREF / 10 * (SC1 + L) * exp(-10 * (SC1 + L) / SCREF) - SCREF * SCREF / 100 * exp(-10 * (SC1 + L) / SCREF));

SCBL2 = 1 / (L * SCREF) * (SCREF / 10 * SC3 * exp(-10 * SC3 / SCREF) + SCREF * SCREF / 100 * exp(-10 * SC3 / SCREF) - SCREF / 10 * (SC3 + L) *exp(-10*(SC3+L)/ SCREF) - SCREF * SCREF / 100 * exp(-10 * (SC3 + L) / SCREF));

SCB = SCBW1 + SCBW2 + SCBL1 + SCBL2;

SCCW1 = 1 / (W * SCREF) * (SCREF / 20 * SC2 * exp(-20 * SC2 / SCREF) + SCREF * SCREF / 400 * exp(-20 * SC2 / SCREF) - SCREF / 20 * (SC2 + W) * exp(-20 * (SC2 + W) / SCREF) - SCREF * SCREF / 400 * exp(-20 * (SC2 + W) / SCREF));

SCCW2 = 1 / (W * SCREF) * (SCREF / 20 * SC4 * exp(-20 * SC4 / SCREF) + SCREF * SCREF / 400 * exp(-20 * SC4 / SCREF) - SCREF / 20 * (SC4+W) * exp(-20 * (SC4 + W) / SCREF) - SCREF * SCREF / 400 * exp(-20 * (SC4 + W) / SCREF));

SCCL1 = 1 / (L * SCREF) * (SCREF / 20 * SC1 * exp(-20 * SC1 / SCREF) + SCREF * SCREF / 400 * exp(-20 * SC1 / SCREF) - SCREF / 20 * (SC1 + L) * exp(-20 * (SC1 + L) / SCREF) - SCREF * SCREF / 400 * exp(-20 *(SC1 + L) / SCREF));

SCCL2 = 1 / (L * SCREF) * (SCREF / 20 * SC3 * exp(-20 * SC3 / SCREF) + SCREF * SCREF / 400 * exp(-20 * SC3 / SCREF) - SCREF /20 * (SC3 + L) * exp(-20 * (SC3 + L) / SCREF) - SCREF * SCREF / 400 * exp(-20 * (SC3 + L) / SCREF));

SCC = SCCW1 + SCCW2 + SCCL1 + SCCL2;

An example of the output netlist (.spo) for parameters of one of transistors:

```
m_mos3_10 T2 T1 T3 lvs_6 mos3 L=1.000u
W=3.000u AD=5.160p AD1=5.160 AS=21.840p
+ AS1=21.840 PD=9.440u PS=20.560u SC1=8.280u
SC2=1.351u SC3=2.720u SC4=2.000u
+ SCA=0.345 SCB=32.260n SCBL1=0.000a
SCBL2=0.103p SCBW1=32.210n SCBW2=50.237p
+ SCC=10.410f SCCL1=0.0 SCCL2=0.000a
SCCW1=10.410f SCCW2=0.018a sa=1.720u
+ sb=7.280u sc=7.280u sd=1.720u
+ X=28.72 Y=-13.00
```

Conclusion

This application note describes the new functions for well proximity and STI stress effects parameter calculations in SmartDRC/LVS. These features permit SmartDRC/LVS to be successfully used for device layout parameter extraction for the middle (90-65nm) CMOS technology processes and advanced (28nm and below) process nodes.