Whom do you trust?

Validating process parameters for open-source tools

Tim Edwards

SVP Analog & Platform

efabless

efabless.com

Open Circuit Design

opencircuitdesign.com

Open-source PDKs are a collaborative effort!

SkyWater sky130 open PDK
skywatertechnology.com

efabless
efabless.com

Google
google.com

OpenROAD
theopenroadproject.org

Github
github.com

. . . and many others!

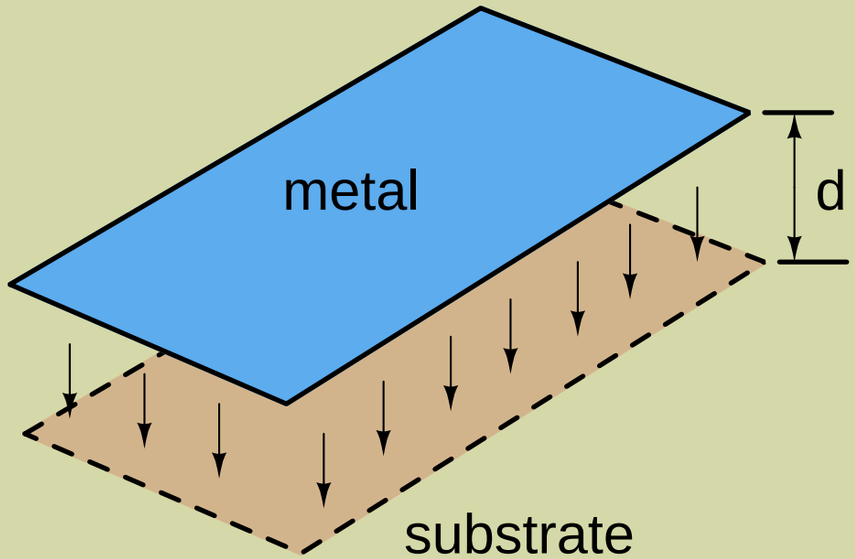
Overview of parasitic capacitance

Parallel plate area capacitance:

Simple physics!

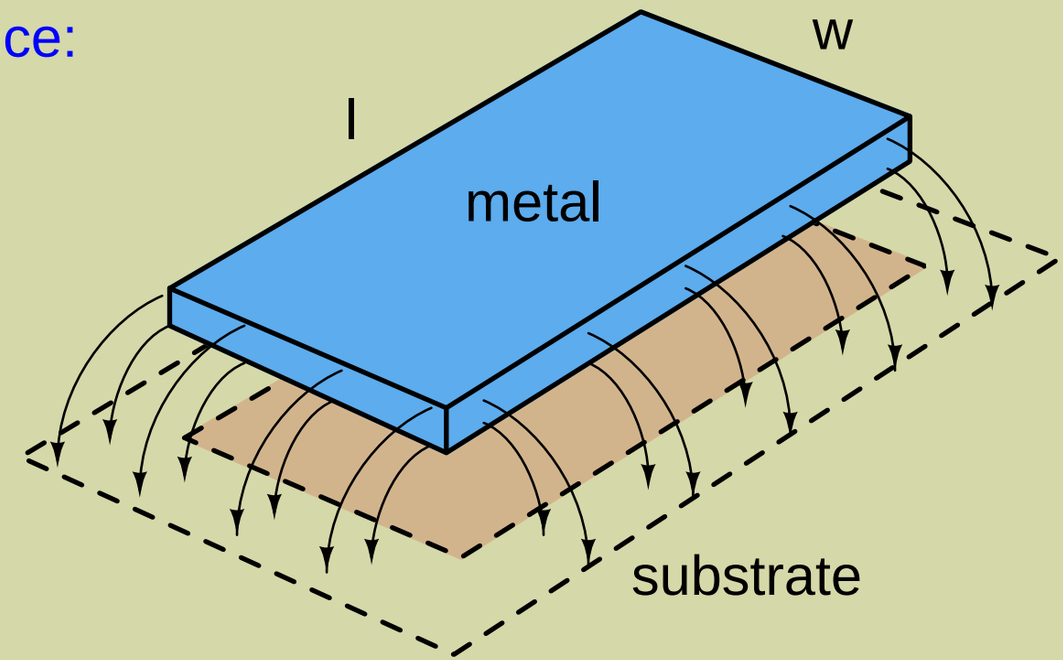
$$C = C_{area} \times area$$

$$C_{area} = \frac{\varepsilon_{Si} \cdot K}{d} \; (F/\mu m^2)$$



Overview of parasitic capacitance

Parallel plate fringe capacitance:



Not simple physics at all!

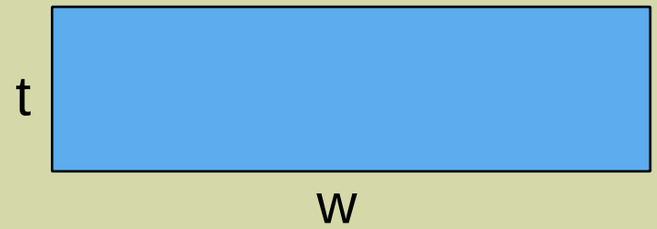
However, *total* fringe can be computed as

$C = C_{perim} \times perim$ $\qquad$ $perim = 2l \times 2w$

C_{perim} = (some ridiculously complicated expression) (F/µm)

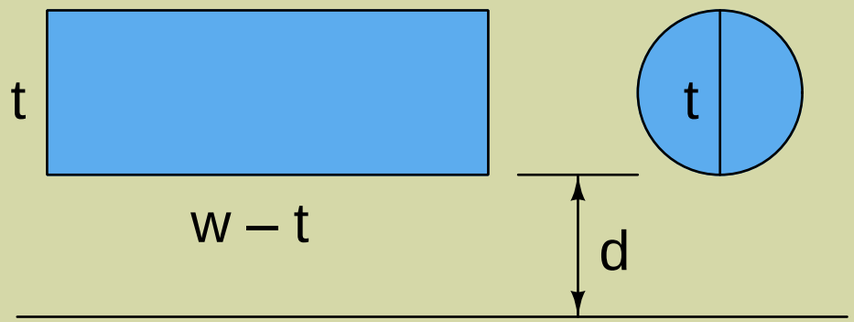
Overview of parasitic capacitance

One (of many) analytic models of fringe capacitance:



wire width w, thickness t, height d above substrate

model as wire width (w – t) plus two half-cylinders diameter t



calculate area capacitance on (shortened) wire and add to result for cylinder:

$$C = K\varepsilon_{Si}(w-t)/d + 2\pi\varepsilon_{Si}/\log(d/t)$$

(C is per unit length, F/μm)

Note that this result is a total capacitance and does not say how it is distributed over distance from a wire. It can be used to estimate the maximum fringe capacitance.

Overview of parasitic capacitance

Analytic fringe parasitic capacitance, Calibre style

cases: overlap, underlap, same net, different net.

```
CAPACITANCE CROSSOVER FRINGE MET1_cond MASK Substrate
 [
     if ((distance() > 0.0) && (same_net() == 0)) {
         C = length() * 0.176651
                 * (1 - exp(-0.418008 * (distance() + 0.358863)))
                 * pow(width() , 0.0106984 * distance() + -0.00163982)
                 * (0.0623825 * thickness() + 0.205178)
                 * m34IN_COEF
                 * (1 - m34RS * exp((-m43ink1 * radius_down()) / (m43ink2 * distance() + m43ink3 * 1.3761)))
                 * (1 - m34RS * exp((-m43ink4 * radius_up()) / (m43ink5 * distance() + m43ink6 * 1.3761)))
     }
     if ((distance() > 0.0) && (same_net() == 1)) {
         C = length() * 0.0365238
                 * (1 - exp(-0.471613 * (distance() + 0.142079)))
                 * m34IN_COEF
                 * (1 - m34RS * exp((-m43ink1 * radius_down()) / (m43ink2 * distance() + m43ink3 * 1.3761)))
                 * (1 - m34RS * exp((-m43ink4 * radius_up()) / (m43ink5 * distance() + m43ink6 * 1.3761)))
     }
     if (distance() <= 0.0) {
         C = length() * 0.157414
                 * pow(width() , 0.0968796)
                 * (0.100672 * thickness() + 0.236357)
                 * m34IN0_COEF
                 * (1 - m34RS * exp((-m43ink1 * radius_down()) / (m43ink2 * 7 + m43ink3 * 1.3761)))
                 * (1 - m34RS * exp((-m43ink4 * radius_up()) / (m43ink5 * 7 + m43ink6 * 1.3761)))
     }
 ]
```

Overview of parasitic capacitance

Sidewall capacitance:

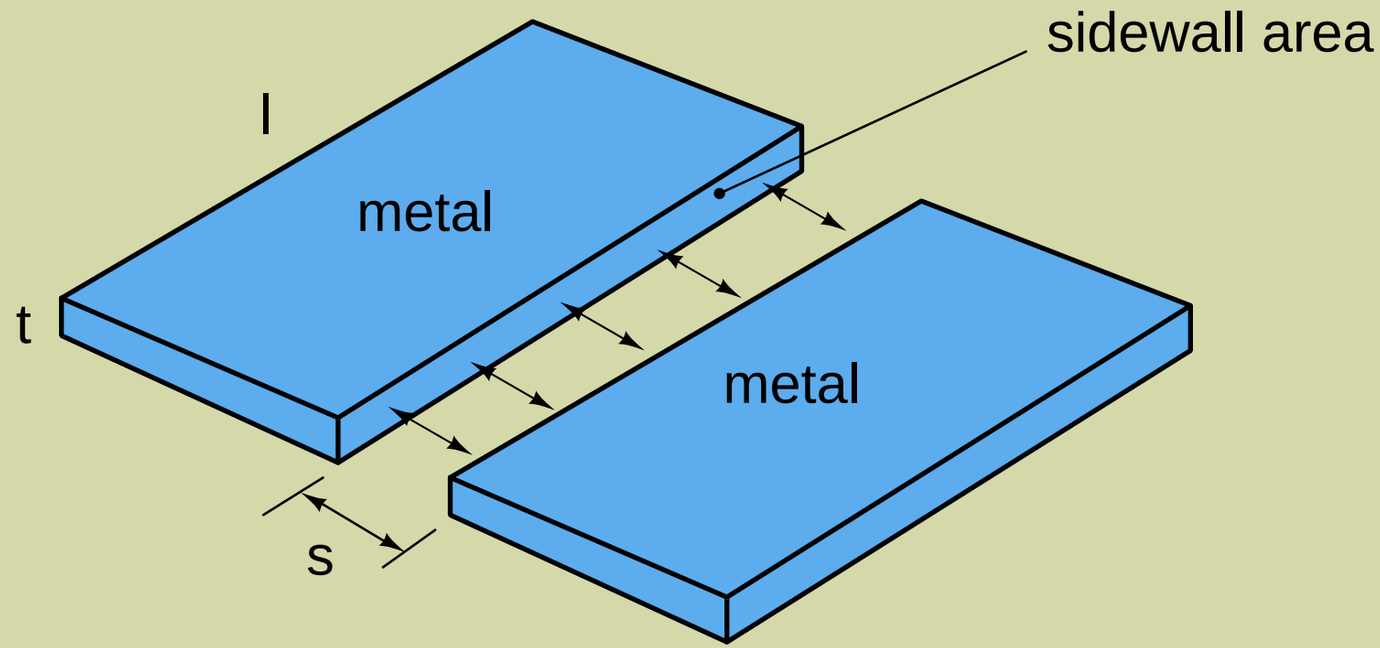
Seems like simple physics. . .

$C = C_{coup} \times \text{sidewall area}$

$= C_{coup} \times l \times t$

$$C_{coup} = \frac{\varepsilon_{Si} \cdot K}{s} \ (F/\mu m^2)$$

. . . but it's not.



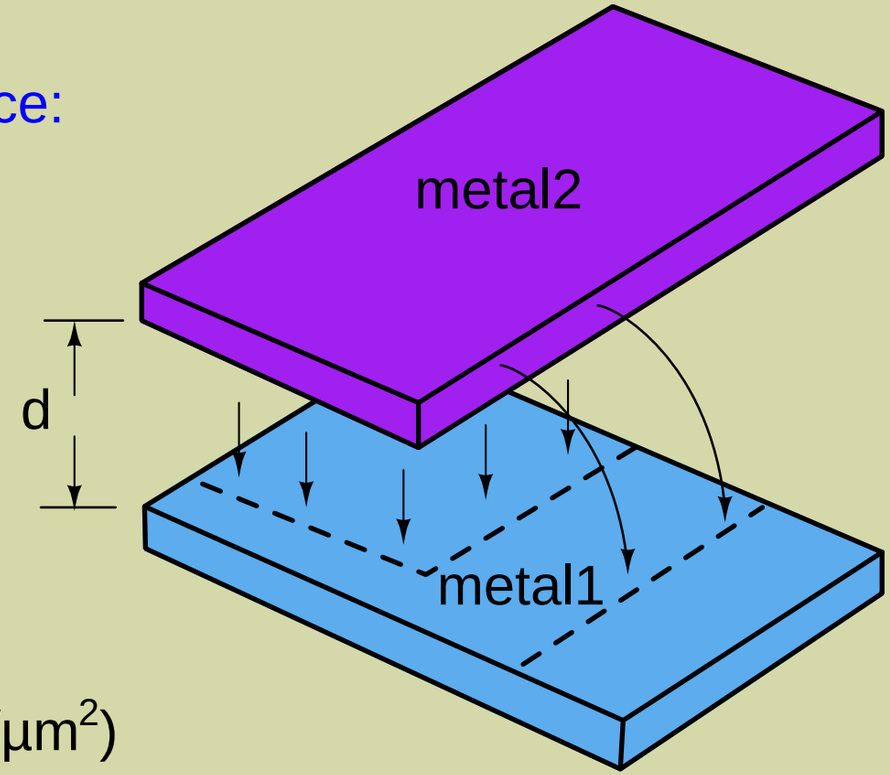
(Normally t is multiplied into the coefficient which then has units F/µm)

(C_{coup} can be given as a constant coefficient if referenced to s = 1)

Overview of parasitic capacitance

Overlap capacitance:

(area and fringe)

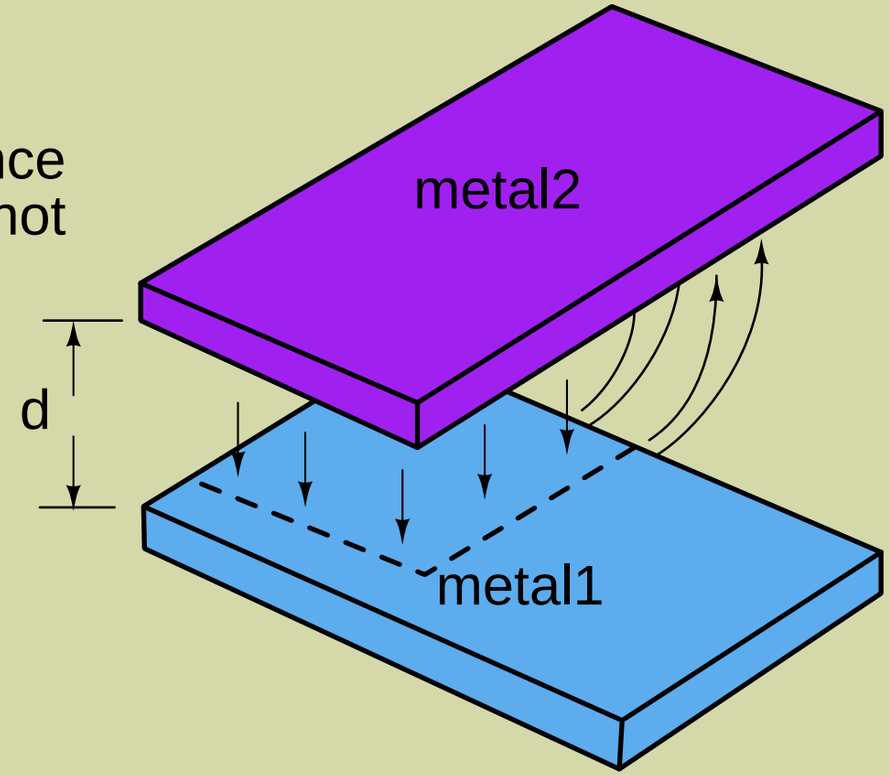


$$C_{area} = \frac{\varepsilon_{Si} \cdot K}{d} \ (F/\mu m^2)$$

C_{perim} = (some ridiculously complicated expression) (F/µm)

Overview of parasitic capacitance

Overlap fringe capacitance goes both ways (and is not symmetric)!



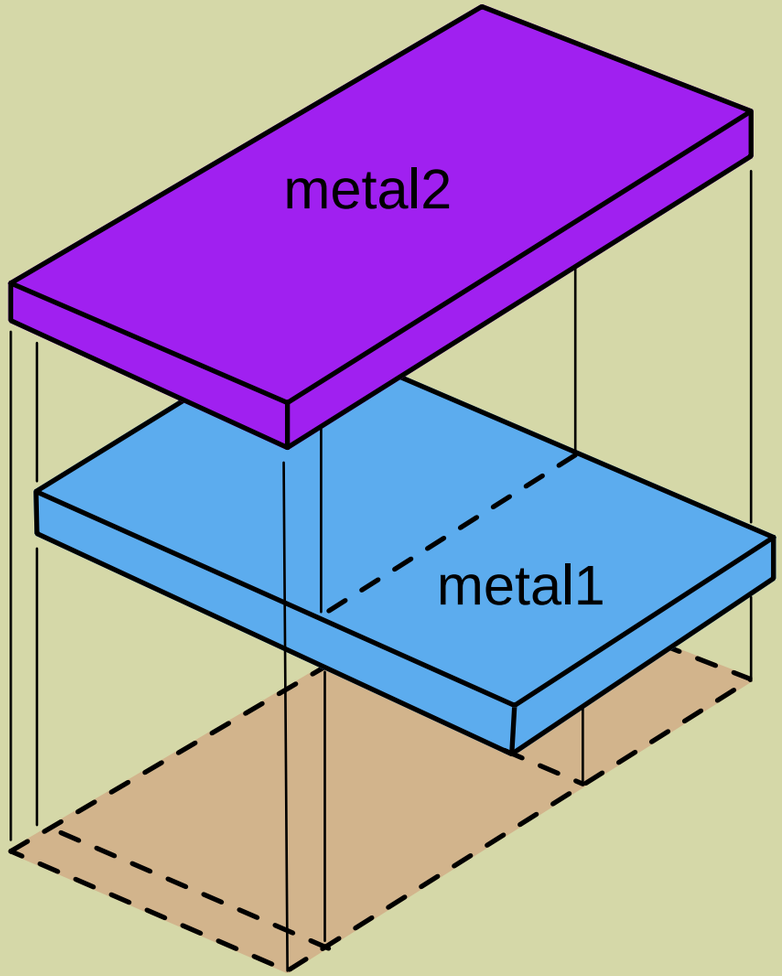
C_{perim} = (some ridiculously complicated expression) (F/µm)

Overview of parasitic capacitance

Major effects:

Vertical shielding

(area and fringe)

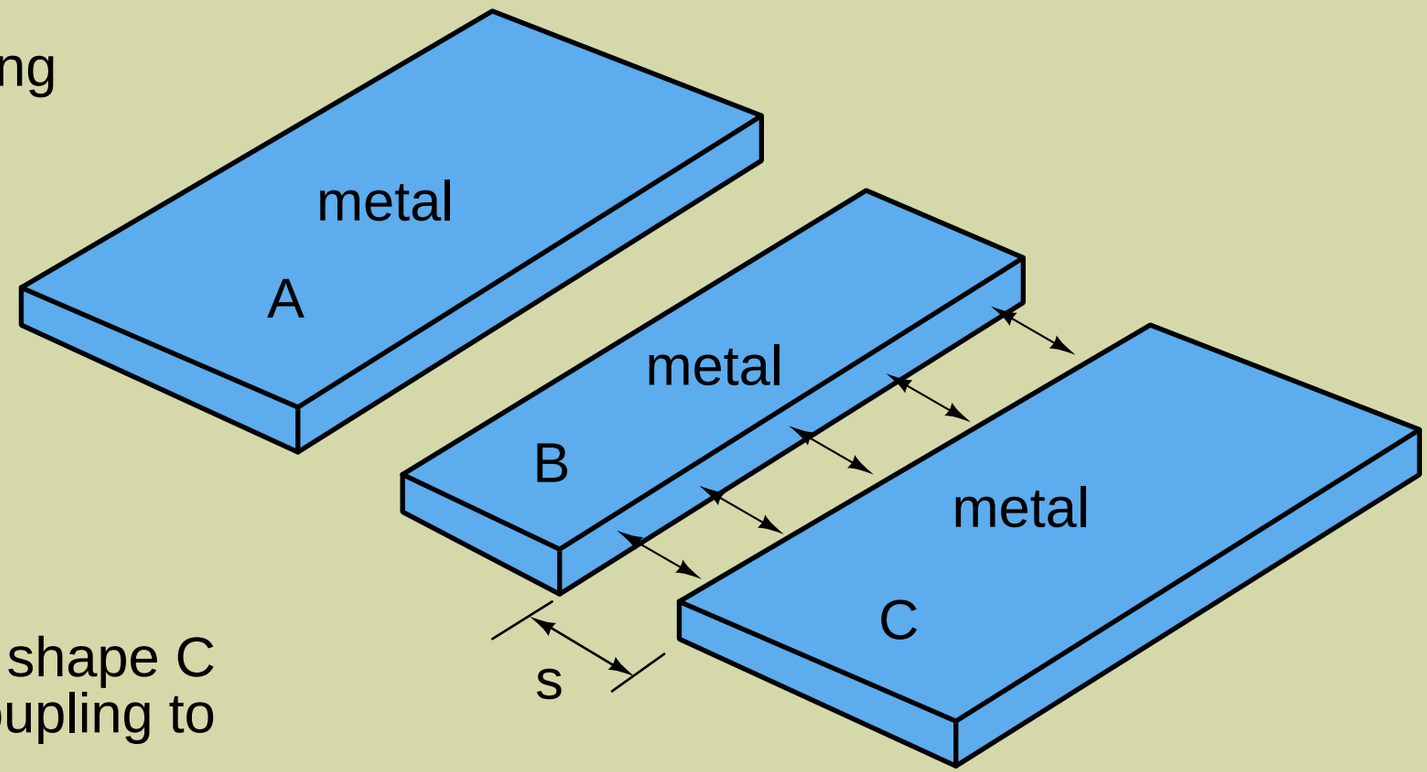


metal1 shields part of metal2 from the substrate

Overview of parasitic capacitance

Major effects:

Lateral sidewall shielding

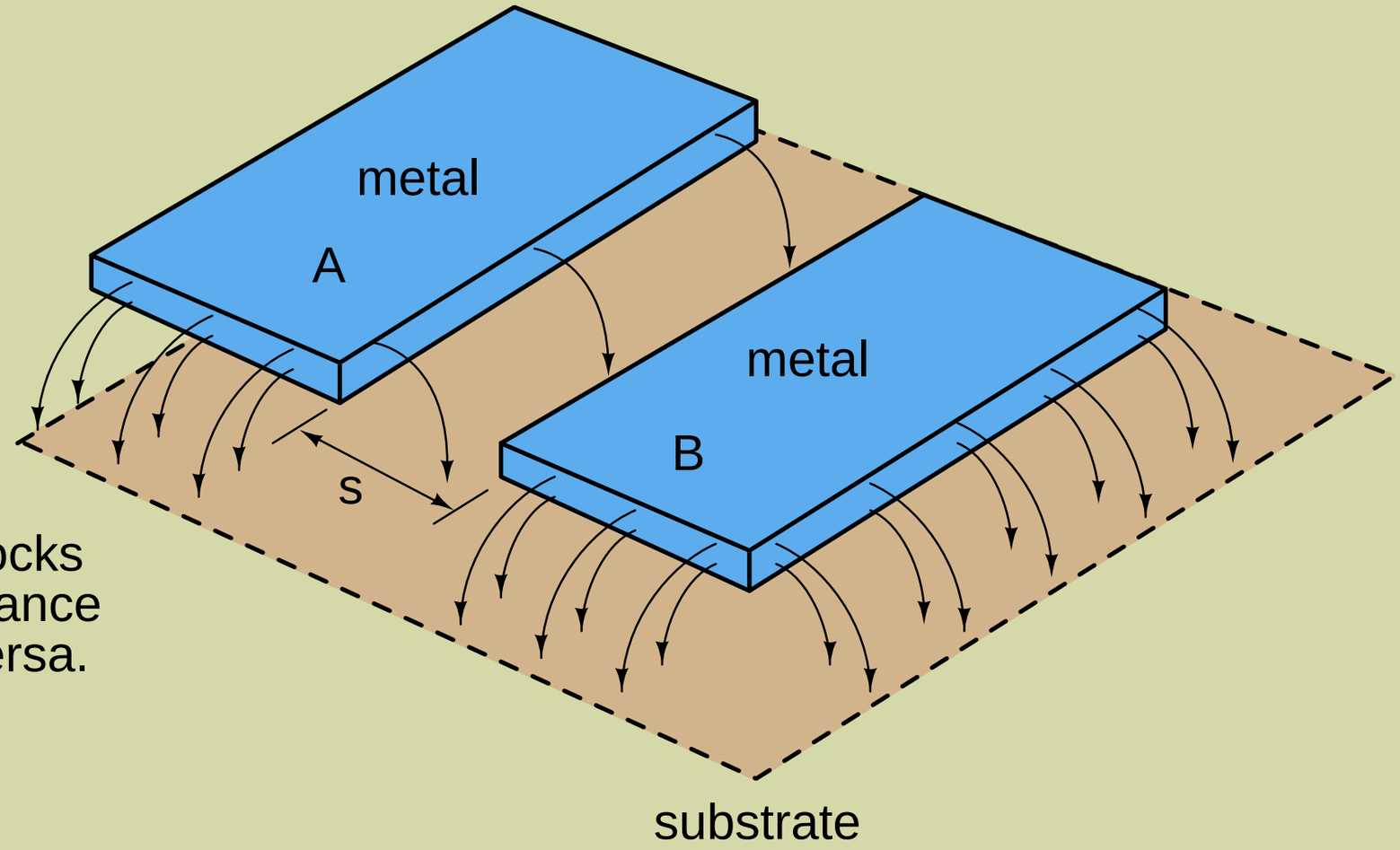


Shape B blocks shape C from sidewall coupling to shape A

Overview of parasitic capacitance

Major effects:

Lateral fringe shielding



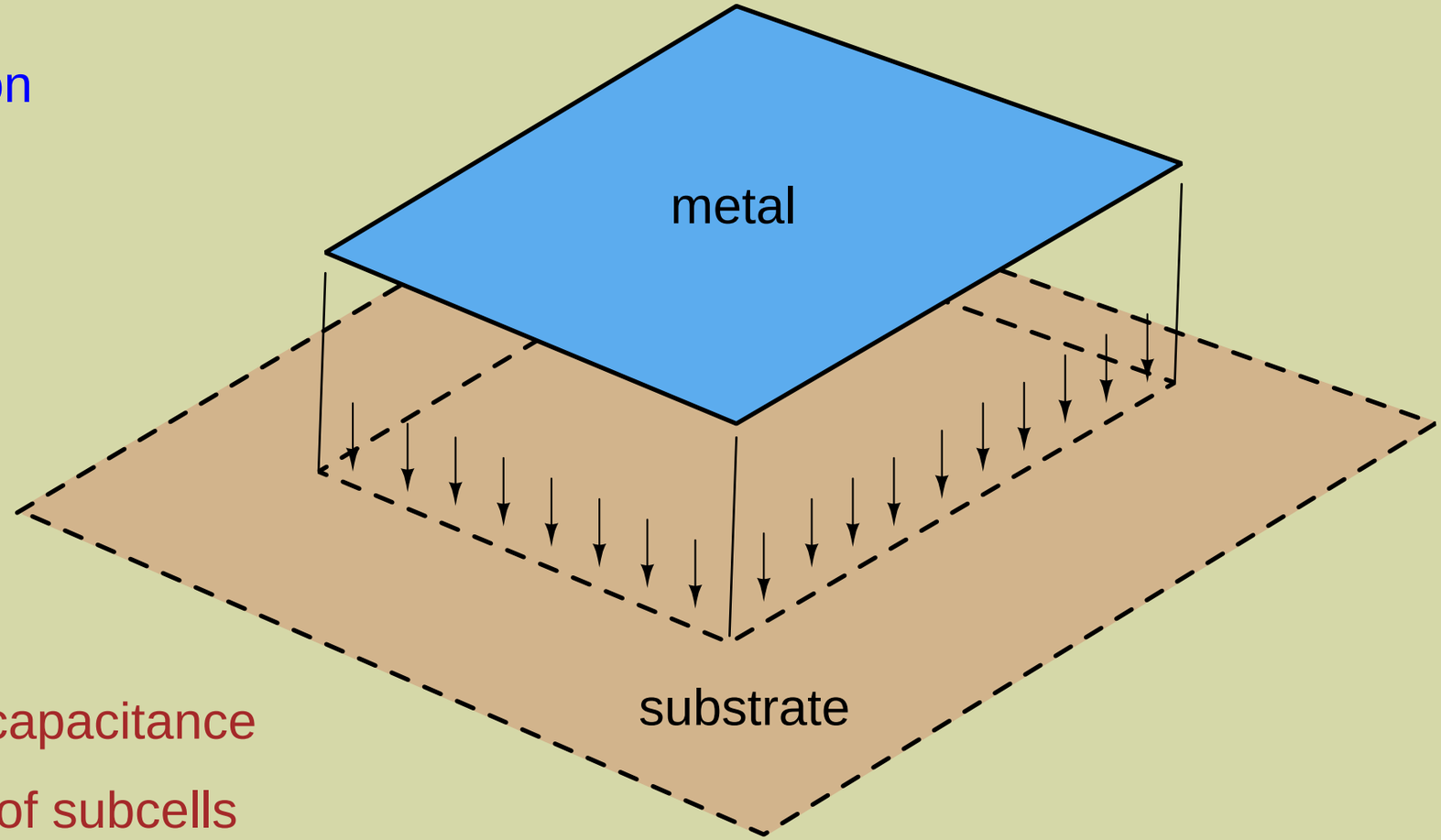
Presence of shape A blocks shape B's fringe capacitance to substrate, and vice versa.

Parasitic capacitance extraction in Magic

Magic's parasitic extraction
(up to version 8.3.277)

Computes:

- area capacitance
- fringe capacitance
- sidewall capacitance
- overlap area and fringe capacitance
- hierarchical contribution of subcells

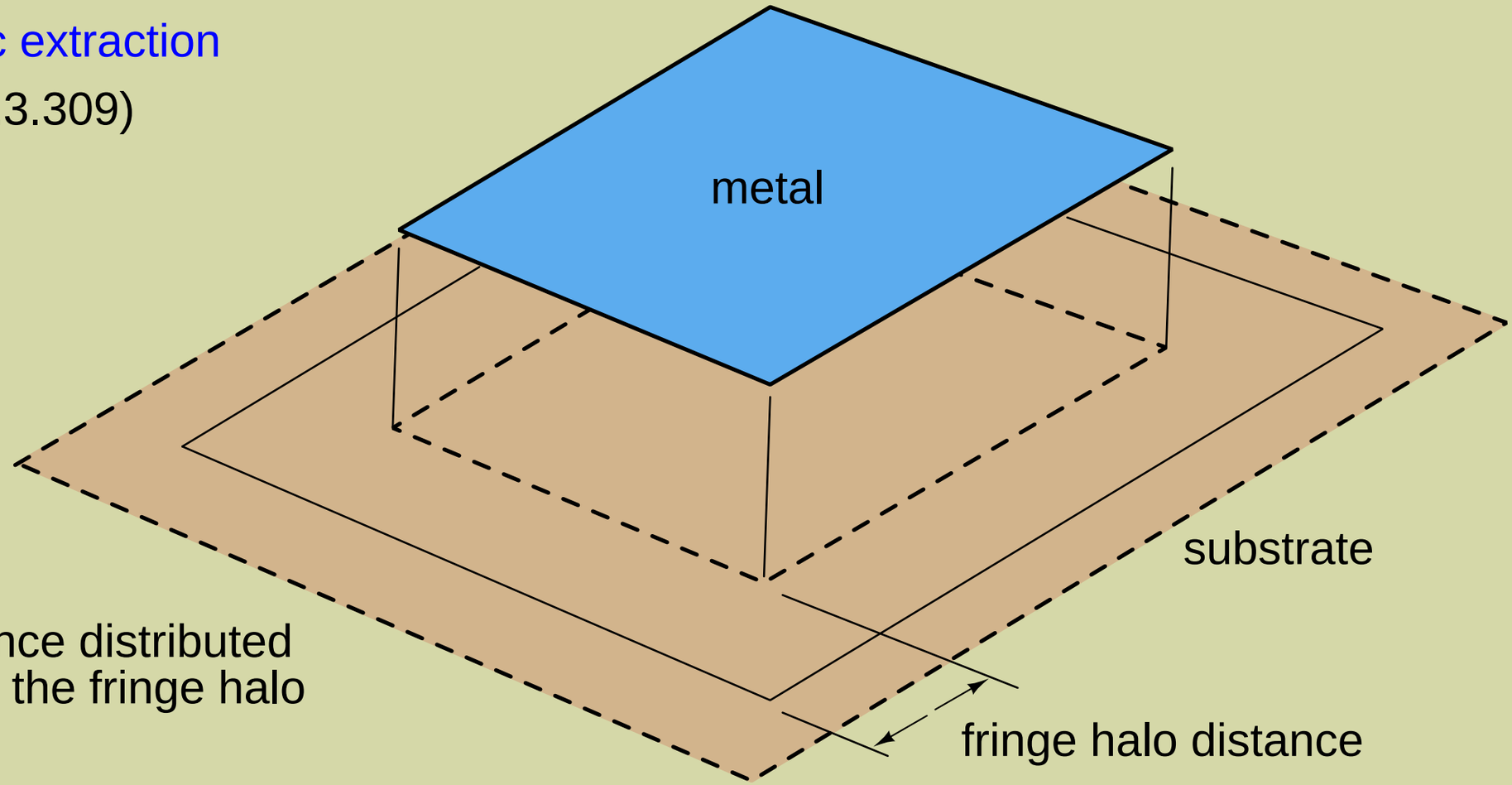


All fringe capacitance calculated as if it is applied on an infinitesmally thin line along the edge.

Parasitic capacitance extraction in Magic

Magic's parasitic extraction

(since version 8.3.309)



Fringe capacitance distributed over area out to the fringe halo distance.

Parasitic capacitance extraction in Magic

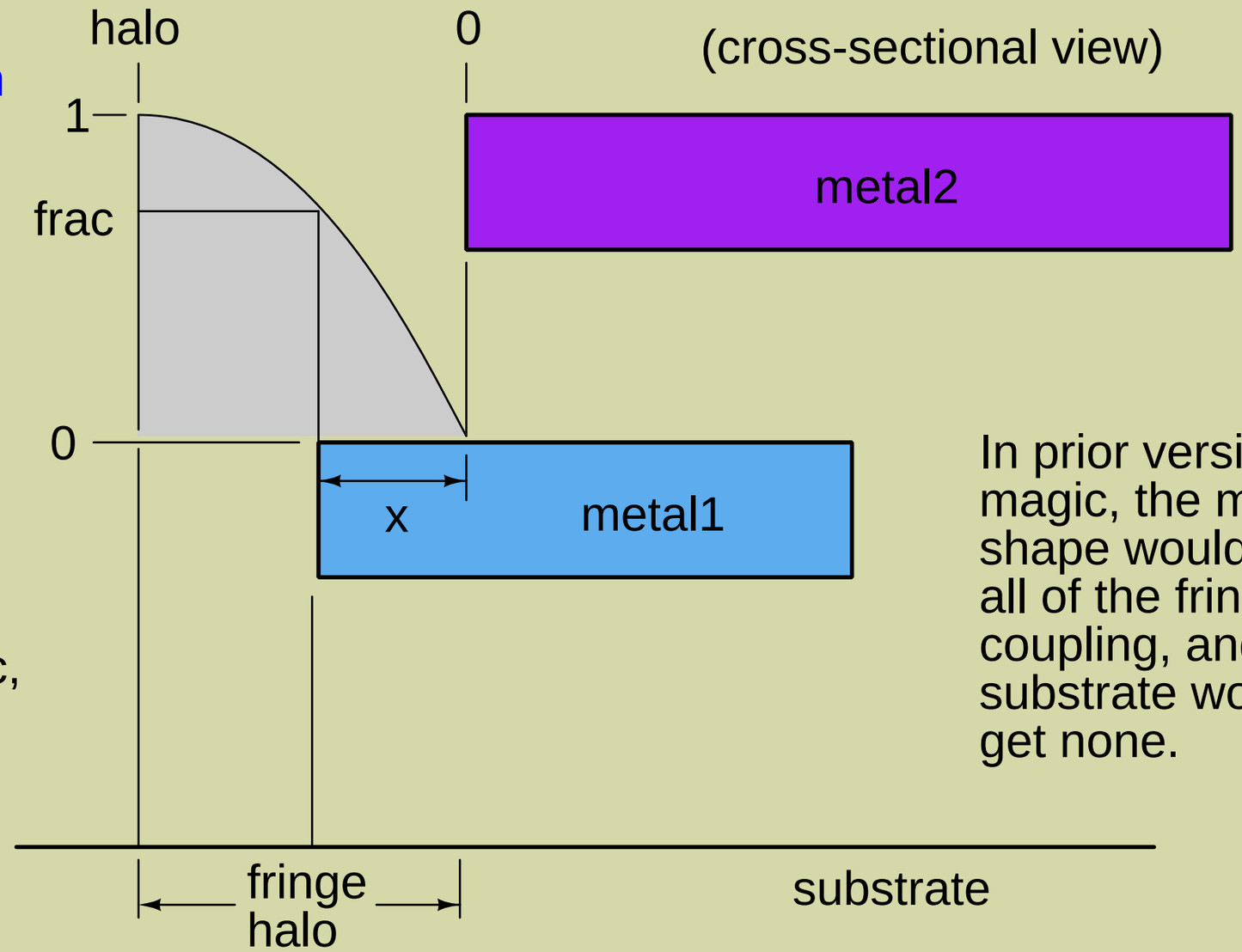
Magic's parasitic extraction

(since version 8.3.309)

Theory of approximate analytic expression for fringing field:

Assume field is bounded by the fringe halo. The amount of fringe field out to distance $x = C_{perim} \times frac$, where

$$frac = \sin((\pi/2)\cdot(x/halo))$$

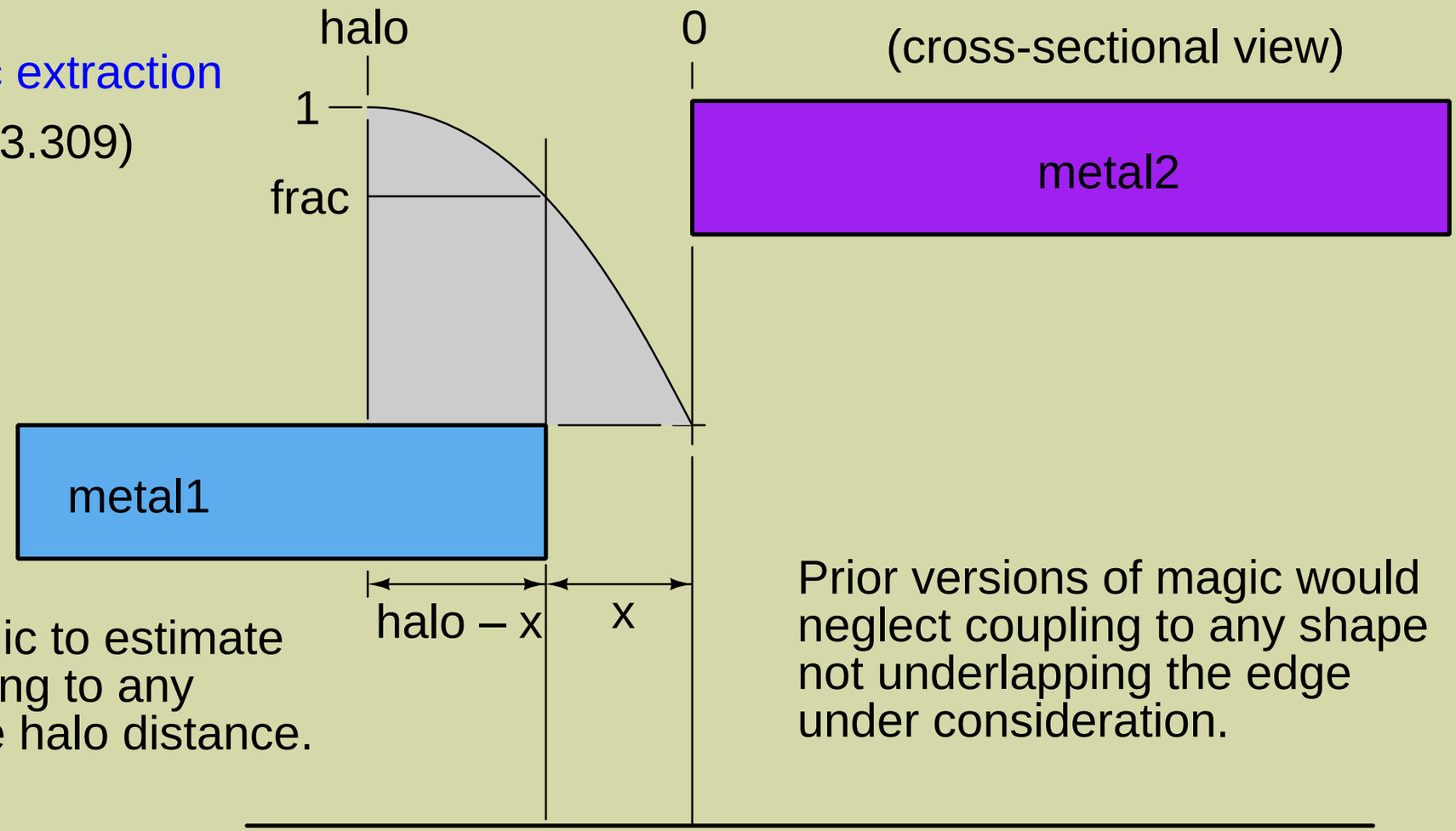


In prior versions of magic, the metal1 shape would get all of the fringe coupling, and the substrate would get none.

Parasitic capacitance extraction in Magic

Magic's parasitic extraction

(since version 8.3.309)



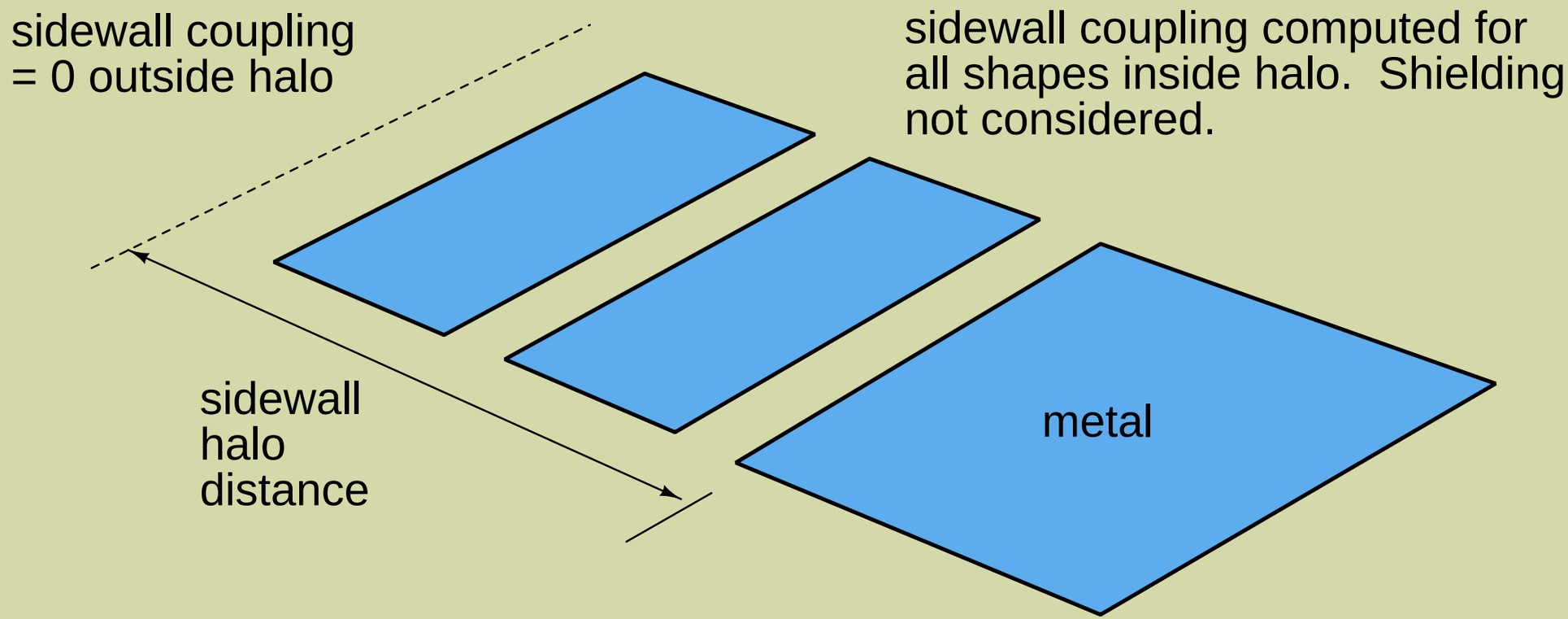
This allows magic to estimate the fringe coupling to any shape within the halo distance.

Prior versions of magic would neglect coupling to any shape not underlapping the edge under consideration.

Parasitic capacitance extraction in Magic

Magic's parasitic extraction

(up to version 8.3.277)

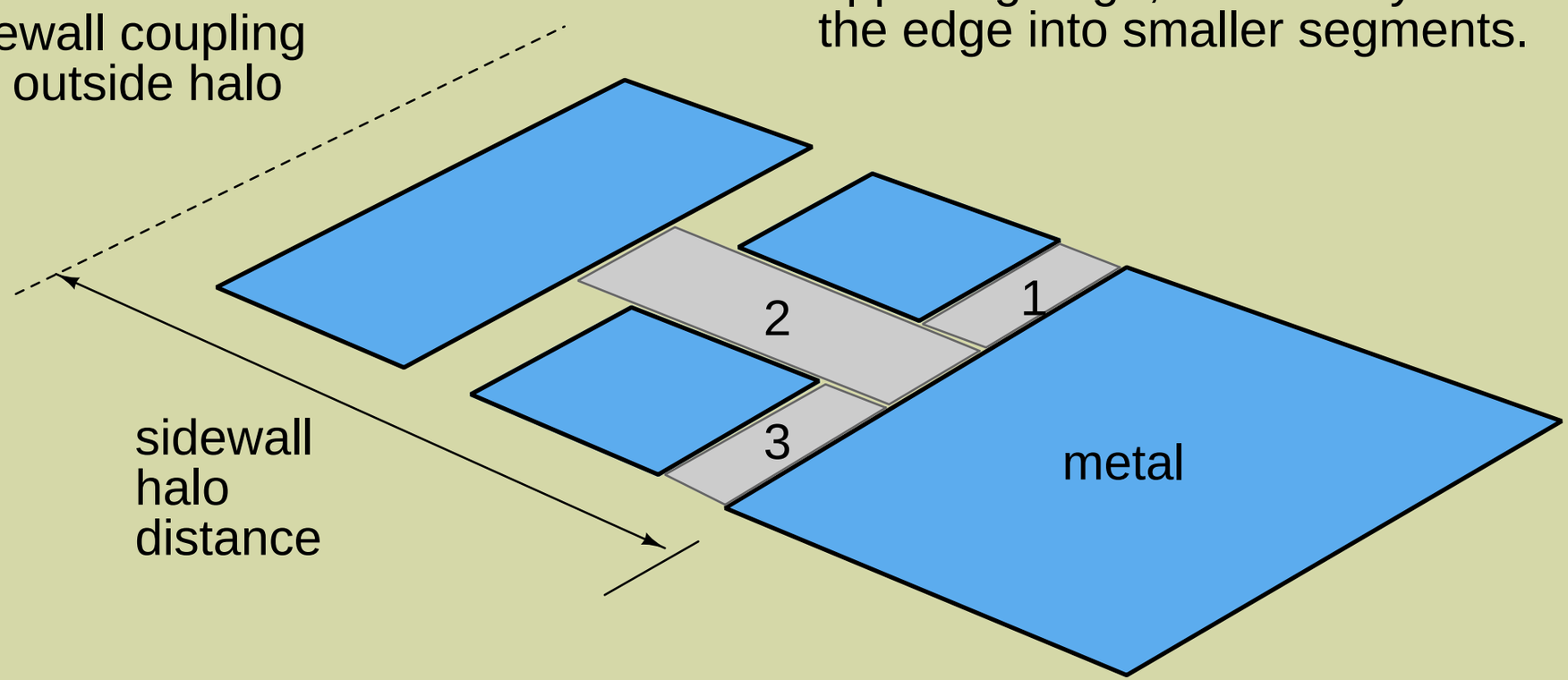


Parasitic capacitance extraction in Magic

Magic's parasitic extraction

(since version 8.3.309)

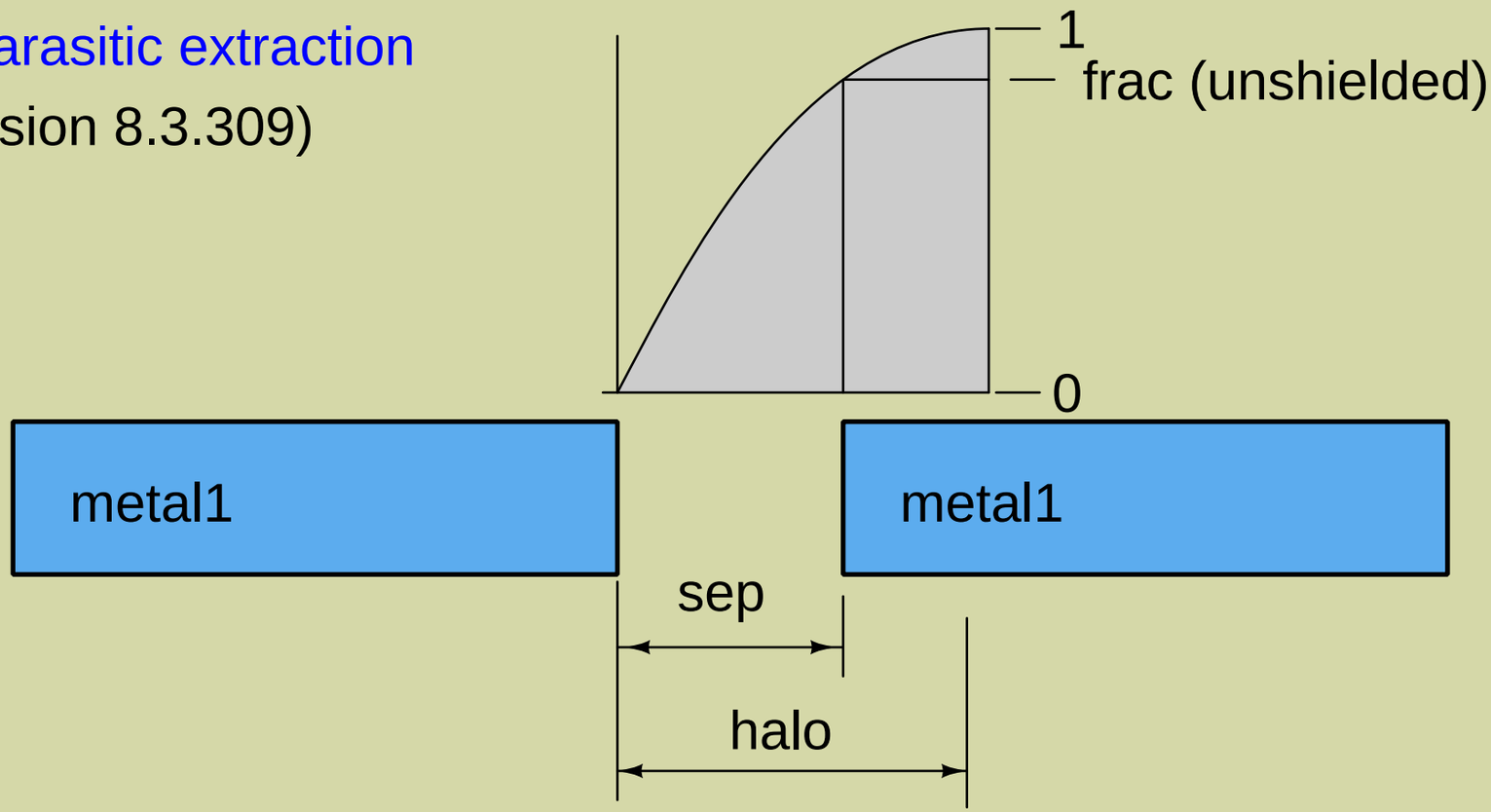
Magic now scans every edge from end to end and finds the nearest neighbor opposing edge, recursively breaking the edge into smaller segments.



Parasitic capacitance extraction in Magic

Magic's parasitic extraction

(since version 8.3.309)



Fringe shielding computed as:
Fraction of fringe field unshielded = $\sin((\pi/2) \cdot \text{sep} / \text{halo}$

Cross-validation

Foundry measured data, in SPICE format:

condition: p1 (poly) over field oxide (f)

line width

line spacing

```
mrxvt
* SKY130 Spice File.
.param  globalk=1
.param  localkswitch=1
.param  capunits = '1.0*1e-6'
.param
+ mcp1f_ca_w_0_150_s_0_210 = 1.06e-04  mcp1f_cc_w_0_150_s_0_210 = 7.62e-11  mcp1f_cf_w_0_150_s_0_210 = 1.03e-11
+ mcp1f_ca_w_0_150_s_0_263 = 1.06e-04  mcp1f_cc_w_0_150_s_0_263 = 6.19e-11  mcp1f_cf_w_0_150_s_0_263 = 1.25e-11
+ mcp1f_ca_w_0_150_s_0_315 = 1.06e-04  mcp1f_cc_w_0_150_s_0_315 = 5.27e-11  mcp1f_cf_w_0_150_s_0_315 = 1.45e-11
+ mcp1f_ca_w_0_150_s_0_420 = 1.06e-04  mcp1f_cc_w_0_150_s_0_420 = 4.04e-11  mcp1f_cf_w_0_150_s_0_420 = 1.85e-11
+ mcp1f_ca_w_0_150_s_0_525 = 1.06e-04  mcp1f_cc_w_0_150_s_0_525 = 3.29e-11  mcp1f_cf_w_0_150_s_0_525 = 2.17e-11
+ mcp1f_ca_w_0_150_s_0_630 = 1.06e-04  mcp1f_cc_w_0_150_s_0_630 = 2.76e-11  mcp1f_cf_w_0_150_s_0_630 = 2.46e-11
+ mcp1f_ca_w_0_150_s_0_840 = 1.06e-04  mcp1f_cc_w_0_150_s_0_840 = 2.03e-11  mcp1f_cf_w_0_150_s_0_840 = 2.94e-11
+ mcp1f_ca_w_0_150_s_1_260 = 1.06e-04  mcp1f_cc_w_0_150_s_1_260 = 1.18e-11  mcp1f_cf_w_0_150_s_1_260 = 3.62e-11
+ mcp1f_ca_w_0_150_s_2_310 = 1.06e-04  mcp1f_cc_w_0_150_s_2_310 = 5.10e-12  mcp1f_cf_w_0_150_s_2_310 = 4.24e-11
+ mcp1f_ca_w_0_150_s_5_250 = 1.06e-04  mcp1f_cc_w_0_150_s_5_250 = 1.20e-12  mcp1f_cf_w_0_150_s_5_250 = 4.63e-11
+ mcp1f_ca_w_1_200_s_0_210 = 1.06e-04  mcp1f_cc_w_1_200_s_0_210 = 9.44e-11  mcp1f_cf_w_1_200_s_0_210 = 1.02e-11
+ mcp1f_ca_w_1_200_s_0_263 = 1.06e-04  mcp1f_cc_w_1_200_s_0_263 = 7.89e-11  mcp1f_cf_w_1_200_s_0_263 = 1.25e-11
+ mcp1f_ca_w_1_200_s_0_315 = 1.06e-04  mcp1f_cc_w_1_200_s_0_315 = 6.86e-11  mcp1f_cf_w_1_200_s_0_315 = 1.46e-11
+ mcp1f_ca_w_1_200_s_0_420 = 1.06e-04  mcp1f_cc_w_1_200_s_0_420 = 5.49e-11  mcp1f_cf_w_1_200_s_0_420 = 1.84e-11
+ mcp1f_ca_w_1_200_s_0_525 = 1.06e-04  mcp1f_cc_w_1_200_s_0_525 = 4.61e-11  mcp1f_cf_w_1_200_s_0_525 = 2.19e-11
+ mcp1f_ca_w_1_200_s_0_630 = 1.06e-04  mcp1f_cc_w_1_200_s_0_630 = 3.98e-11  mcp1f_cf_w_1_200_s_0_630 = 2.49e-11
+ mcp1f_ca_w_1_200_s_0_840 = 1.06e-04  mcp1f_cc_w_1_200_s_0_840 = 3.12e-11  mcp1f_cf_w_1_200_s_0_840 = 2.99e-11
+ mcp1f_ca_w_1_200_s_1_260 = 1.06e-04  mcp1f_cc_w_1_200_s_1_260 = 2.14e-11  mcp1f_cf_w_1_200_s_1_260 = 3.71e-11
+ mcp1f_ca_w_1_200_s_2_310 = 1.06e-04  mcp1f_cc_w_1_200_s_2_310 = 1.08e-11  mcp1f_cf_w_1_200_s_2_310 = 4.64e-11
+ mcp1f_ca_w_1_200_s_5_250 = 1.06e-04  mcp1f_cc_w_1_200_s_5_250 = 3.40e-12  mcp1f_cf_w_1_200_s_5_250 = 5.36e-11
+ mcl1f_ca_w_0_170_s_0_180 = 3.69e-05  mcl1f_cc_w_0_170_s_0_180 = 7.98e-11  mcl1f_cf_w_0_170_s_0_180 = 3.26e-12
+ mcl1f_ca_w_0_170_s_0_225 = 3.69e-05  mcl1f_cc_w_0_170_s_0_225 = 6.83e-11  mcl1f_cf_w_0_170_s_0_225 = 4.04e-12
+ mcl1f_ca_w_0_170_s_0_270 = 3.69e-05  mcl1f_cc_w_0_170_s_0_270 = 6.07e-11  mcl1f_cf_w_0_170_s_0_270 = 4.81e-12
+ mcl1f_ca_w_0_170_s_0_360 = 3.69e-05  mcl1f_cc_w_0_170_s_0_360 = 4.97e-11  mcl1f_cf_w_0_170_s_0_360 = 6.42e-12
```

area capacitance | sidewall capacitance | fringe capacitance

Cross-validation

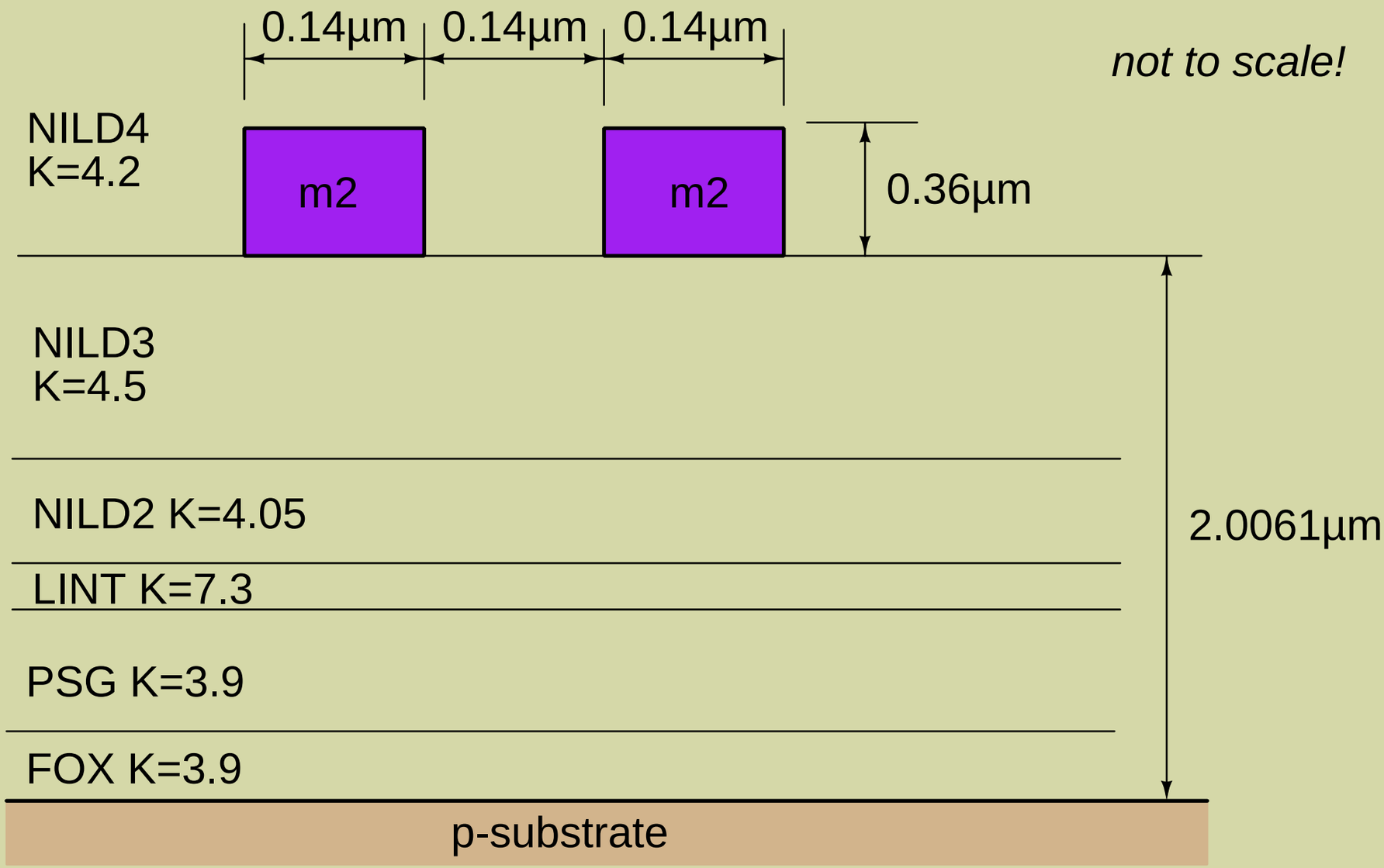
Field equation solver: FasterCap (2D)

```
* 2D - metal2 to field substrate
* mcm2f_cc_w_0_140_s_0_140 follows SPICE parameter of the same name.
* Indicates a metal of width 0.14um with adjacent metal2 wire at 0.14um distance.
* For this configuration, wire centers are separated by 0.28um.
*
* According to cap tables, the sidewall capacitance is 1.05e-10 pF/um.
*
* Dielectric stack pulled from diagram at:
* https://skywater-pdk.readthedocs.io/en/main/rules/assumptions.html\
* #process-stack-diagram
*
*
* TOPNIT to air
D k_boundary   1.0  7.5  0.0  5.7488   0.0  5.90
* NILD6 to TOPNIT
D k_boundary   7.5  4.0  0.0  5.3711   0.0  5.50
* NILD5 to NILD6
D k_boundary   4.0  4.1  0.0  4.0211   0.0  4.20
* NILD4 to NILD5
D k_boundary   4.1  4.2  0.0  2.7861   0.0  3.00
* NILD3 to NILD4
D k_boundary   4.2  4.5  0.0  2.0061   0.0  2.20
* NILD4_C
* Metal 2 right shield endcaps
D k_endcap_rgt 3.5  4.2 -1.26 2.0061  -1.26 2.70
D k_endcap_rbt 3.5  4.5 -1.26 2.0061  -1.26 2.70
D k_endcap_lft 3.5  4.2 -1.26 2.0061  -1.26 2.70
D k_endcap_lbt 3.5  4.5 -1.26 2.0061  -1.26 2.70
* Metal 2 left shield endcaps
D k_endcap_rgt 3.5  4.2  1.26 2.0061   1.26 2.70
D k_endcap_rbt 3.5  4.5  1.26 2.0061   1.26 2.70
D k_endcap_lft 3.5  4.2  1.26 2.0061   1.26 2.70
D k_endcap_lbt 3.5  4.5  1.26 2.0061   1.26 2.70
* Metal 2 left
C metal2_top   4.2      -0.14 2.0061 +
C metal2_bot   4.5      -0.14 2.0061
* Metal 2 right
C metal2_top   4.2       0.14 2.0061 +
C metal2_bot   4.5       0.14 2.0061
* NILD2 to NILD3
D k_boundary   4.5  4.05 0.0  1.3761   0.0  1.50
* LINT to NILD2
D k_boundary   4.05 7.3  0.0  1.0111   0.0  1.20
* FOX/PST to LINT
D k_boundary   7.3  3.9  0.0  0.9361   0.0  0.95
* substrate (ground plane)
C gnd_plane    3.9       0.0  0.0
End
```

```
File gnd_plane
0 ground plane
*
S ground -10.0 0.0 10.0 0
End

File k_boundary
0 dielectric boundary
*
S plane -10.0 0.0 10.0 0
End

File k_endcap_rbt
0 dielectric boundary
*
S plane 0.07 0.0 0.10 0.0
End

File k_endcap_rgt
0 dielectric boundary
*
S plane 0.07 0.36 0.10 0.36
S plane 0.10 0.0 0.10 0.36
End

File k_endcap_lbt
0 dielectric boundary
*
S plane -0.07 0.0 -0.10 0.0
End

File k_endcap_lft
0 dielectric boundary
*
S plane -0.07 0.36 -0.10 0.36
S plane -0.10 0.0 -0.10 0.36
End

File metal2_top
0 metal2 0.14um wide top
*
S metal2  -0.07  0.0  -0.07  0.36  0.07  0.36  0.07  0.0
End

File metal2_bot
0 metal2 0.14um wide bottom
*
S metal2  -0.07  0.0  0.07  0.0
End
```

Cross-validation

FasterCap 2D geometry for mcm2f_cc_w_0_140_s_0_140 (SkyWater sky130A stackup)



Cross-validation

FasterCap 2D results

```
mrxvt
****************************************
Computing the links..
Number of panels after refinement: 284
Number of links to be computed: 9354
Done computing links
****************************************
Precond Type(s) (-p): Jacobi
GMRES Iteration: 0 1 2 3 4 5 6 7 8 9 10 11 12
GMRES Iteration: 0 1 2 3 4 5 6 7 8 9 10 11 12
Capacitance matrix is:
Dimension 2 x 2
g1_metal2  1.19671e-10 -8.41409e-11
g2_metal2  -8.39095e-11 1.20159e-10

Weighted Frobenius norm of the difference between capacitance (auto option): 0.000907436

Solve statistics:
Number of input panels: 25 of which 5 conductors and 20 dielectric
Number of input panels to solver engine: 25
Number of panels after refinement: 284
Number of potential estimates: 6872
Number of links: 9638 (uncompressed 80656, compression ratio is 88.1%)
Max recursion level: 11
Max Mesh relative refinement value: 0.192763
Iteration time: 0.079057s (0 days, 0 hours, 0 mins, 0 s)
Iteration allocated memory: 16666 kilobytes

Total allocated memory: 16666 kilobytes
Total time: 0.027655s (0 days, 0 hours, 0 mins, 0 s)


Tim@borodin(fastercap_on_sky130)>
```

2×2 matrix, symmetric within margin of error

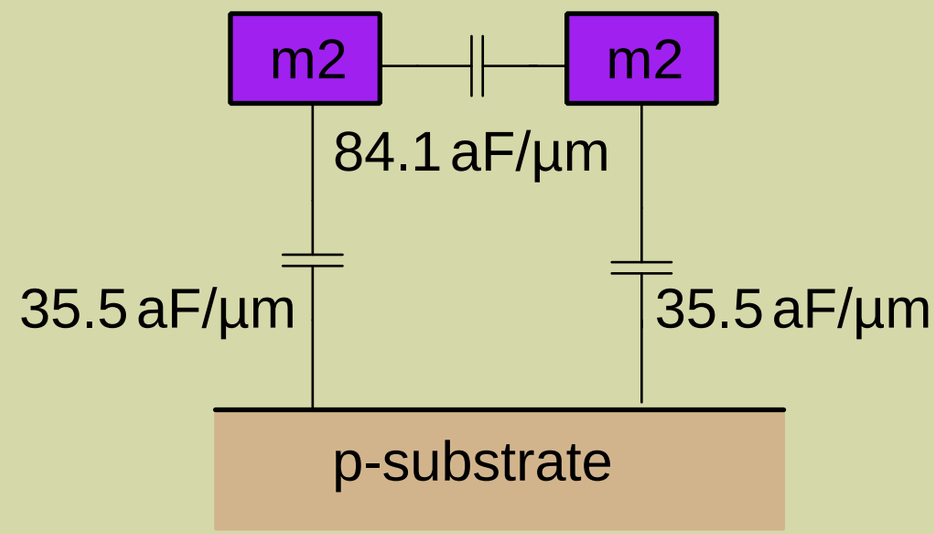
Column 1 = wire on left

Column 2 = wire on right

M_{11} = total capacitance

M_{12} = capacitance between wires

$(M_{11} - M_{12})$ = capacitance to substrate



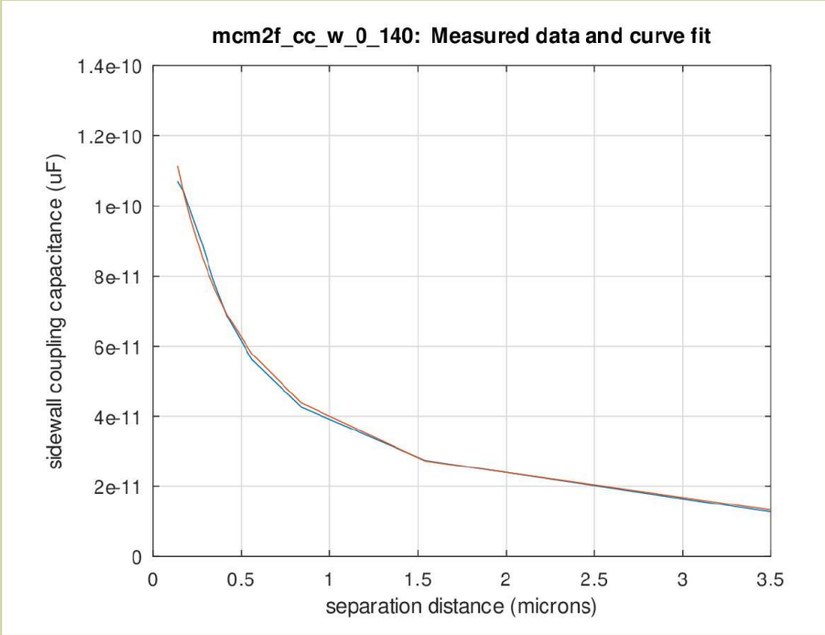
Measured data says sidewall cap is 107 aF/µm, though. . .

Cross-validation

Generate coefficient data for Magic from curve-fitting vendor data

Using Octave least-mean-squares function in the *optim* package:

Sidewall capacitance:

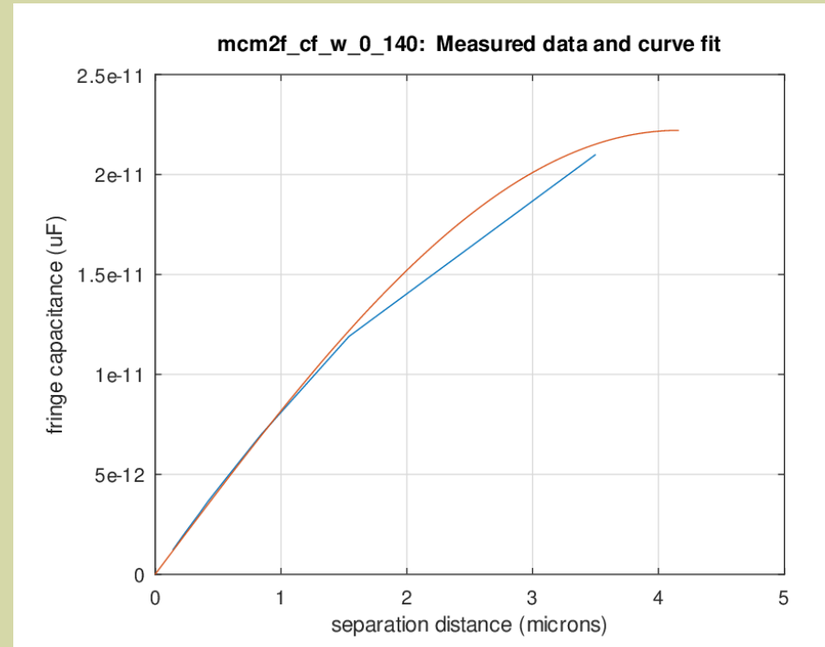


Vendor data fit to
$C = C_{coup}\ (1.0 / (sep + offset))$

$C_{coup} = 50.5\ aF/\mu m$ offset = 0.313 µm

?

Fringe shielding:

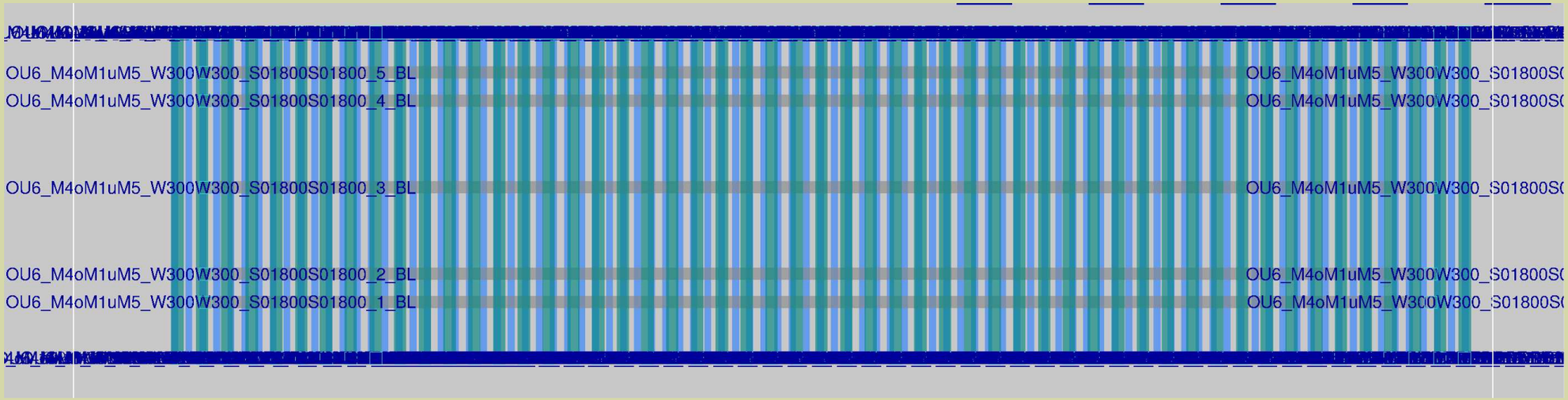


Vendor data fit to
$C = C_{fringe} \sin((\pi/2)(sep / halo))$

$C_{fringe} = 22.2\ aF/\mu m$ halo = 4.17 µm

Applications

OpenROAD OpenRCX calibration



Example of pattern file generated by OpenRCX for calibration

Magic extracted result can be converted to SPEF because structures are simple and follow strict naming conventions.

Applications

OpenROAD OpenRCX calibration

OpenRCX calibration file generated from SPEF data derived from extraction by Magic

```
mrxvt
8 0 8.13525e-05 9.15404e-06
9.6 0 8.23466e-05 9.15404e-06
12.8 0 8.23466e-05 9.15404e-06
16 0 8.23466e-05 9.15404e-06
19.2 0 8.23466e-05 9.15404e-06
END DIST

Metal 6 OVER 4
DIST count 12 width 1.6
1.6 0.000120076 4.73189e-05 9.15404e-06
3.2 6.00379e-05 6.62194e-05 9.15404e-06
4 4.80303e-05 7.46076e-05 9.15404e-06
4.8 4.00253e-05 8.20666e-05 9.15404e-06
5.6 3.43073e-05 8.8449e-05 9.15404e-06
6.4 3.00189e-05 9.36324e-05 9.15404e-06
7.2 2.66835e-05 9.75161e-05 9.15404e-06
8 0 0.000100023 9.15404e-06
9.6 0 0.000101149 9.15404e-06
12.8 0 0.000101149 9.15404e-06
16 0 0.000101149 9.15404e-06
19.2 0 0.000101149 9.15404e-06
END DIST

Metal 6 OVER 5
DIST count 12 width 1.6
1.6 0.000120076 0.000100201 9.15404e-06
3.2 6.00379e-05 0.000126465 9.15404e-06
4 4.80303e-05 0.000138122 9.15404e-06
4.8 4.00253e-05 0.000148487 9.15404e-06
5.6 3.43073e-05 0.000157358 9.15404e-06
```

Then use OpenSTA to calculate delay data on a digital design and compare to results obtained by other methods (commercial tools, technology LEF data).

Conclusions and Future Work

Whom do I trust?

I don't trust anyone completely!

Challenges:

- Data in proprietary or non-human-readable formats
- Human-readable formats following arbitrary standards, with no documentation
- Incomplete and/or erroneous vendor-provided data

Future work:

- Get measured data from silicon
- Automate curve-fitting for sky130 data
- More investigation of models used for parasitics in Magic
- Implementation of more n-th order effects in Magic