The SKY130 passive catalogue

This page lists the resistors and capacitors SKY130 gives you, what each one costs in area, and where each number comes from. It's meant to stay open beside your tools while you design.

We measured every resistance and capacitance below in `hpretl/iic-osic-tools:2026.08` with ngspice 47: we instantiated the device at a stated geometry, simulated it and divided, rather than copying the value from a datasheet. Every parameter is quoted along with the file it lives in, so you can check any line on this page with one `grep`.

Units of W and L

`W` and `L` on a SKY130 device are **plain numbers in microns, with no unit suffix**. `W=1` is one micrometre, and `W=1u` asks for a millionth of that. On a resistor the run doesn't stop: a strip written `W=1u L=10u` comes out at 3193.812 Ω where `W=1 L=10` gives 3550.443 Ω, with no error, and ngspice exits with status 0. The same rule applies to a capacitor's `W` and `L` and to a diode's `area` and `perim`. Getting started shows what happens on a resistor, and AD103′s Reading a SKY130 device model has the exact error text for transistors.

Resistors

Resistor table

The table has thirteen devices in nine rows, and their sheet resistances span five orders of magnitude. `R measured` is the resistance of one square (a strip drawn W = 1 µm, L = 1 µm), measured at 0.1 V:

Device	What it is physically	R_s from the PDK	`R` measured, W=1 L=1
`res_xhigh_po`	poly with the ultra-high-resistance implant	2000 Ω/□	2152.022 Ω
`res_high_po`	poly with the high-resistance implant	317.3885 Ω/□	695.4646 Ω
`res_generic_pd`	p+ diffusion	197 Ω/□	195.4779 Ω
`res_generic_nd`	n+ diffusion	120 Ω/□	117.4914 Ω
`res_generic_po`	ordinary silicided poly, the same material as the gate	48.2 Ω/□	50.92314 Ω
`res_generic_nw`	the n-well itself	1700 Ω/□	1692.556 Ω
`res_iso_pw`	isolated p-well	3816 Ω/□	14969.38 Ω at W=2.65 L=10
`res_generic_l1`	local interconnect	12.2 Ω/□	11.97427 Ω
`res_generic_m1` … `m5`	metal 1–5	0.125 / 0.125 / 0.047 / 0.047 / 0.0285 Ω/□	0.1288327 Ω (m1), 0.04770963 Ω (m3)

The first five devices and `res_iso_pw` are subcircuits, so you instantiate them on `X` lines with `W=` and `L=` in capitals. `res_generic_nw`, `res_generic_l1` and the metal resistors are bare `.model` cards, so they go on `R` lines with lowercase `w=` and `l=`:

```
XR1 a 0 0 sky130_fd_pr__res_high_po W=1 L=10 mult=1
r1  b 0   sky130_fd_pr__res_generic_m1 w=1 l=1
```

If you get that backwards, ngspice says `Error: unknown subckt:` or `warning, can't find model`. The sheet resistances in the table come from these files in the image:

```
/foss/pdks/sky130A/libs.tech/ngspice/r+c/res_typical__cap_typical.spice
        rp1=48.2  rdn=120  rdp=197  rnw=1700  rl1=12.2
        rm1=0.125  rm2=0.125  rm3=0.047  rm4=0.047  rm5=0.0285
        rspwres=3816
/foss/pdks/sky130A/libs.ref/sky130_fd_pr/spice/sky130_fd_pr__res_high_po.model.spice
        rsheet = 317.3885     rhead_ps = 345.8312
/foss/pdks/sky130A/libs.ref/sky130_fd_pr/spice/sky130_fd_pr__res_xhigh_po.model.spice
        rsheet = 2000.0
```

Magic's extraction technology file, `/foss/pdks/sky130A/libs.tech/magic/sky130A.tech`, keeps its own copy of these values in milliohms per square (`xhrpoly 319800`, `uhrpoly 2000000`, `mrp1 48200`), and it doesn't agree with the ngspice file everywhere. Ohms per square explains which one to use when.

End resistance of `res_high_po`

A real resistor has to be contacted, so the device is a strip of resistive poly with a wide head at each end that holds a row of contacts, and those heads add resistance of their own. If you increase the length by 10 µm, the difference tells you what the *body* alone costs:

Device	R at L = 1	R at L = 11	slope (Ω per µm of L)
`res_high_po`, W=1	695.4646 Ω	3867.662 Ω	**317.2198**
`res_xhigh_po`, W=1	2152.022 Ω	—	—

That's 317.2198 Ω per micron of length at W = 1 µm, against `rsheet = 317.3885` on the model card. The two are 0.05 % apart, so the body really is the sheet resistance the card promises. The remaining $695.4646 - 317.2198 = 378.24\ \Omega$ is end resistance: the contact head (299.8915 Ω, see Ohms per square) plus a short body extension that the model adds. The end resistance stays the same when you shorten the resistor, so in a one-square `res_high_po` the ends contribute more than the body does.

The design rule that follows is **never draw a short precision resistor.** The ends are 378 Ω whatever you do, so they make up 54 % of the value at one square, 19 % at five squares and 11 % at ten squares. Below about five squares, a large fraction of your resistor is contact resistance, and contact resistance matches far worse than sheet resistance does.

The fixed-width precision family

`res_high_po` and `res_xhigh_po` also ship in five pre-characterised widths each. These are the versions a layout tool will give you, because their contact head geometry is fixed:

Width (µm)	`res_high_po_*` Ω/µm of length	`res_xhigh_po_*` Ω/µm of length
0.35	1085.483	5713.715
0.69	472.111	2898.261
1.41	219.662	1418.298
2.85	105.534	701.684
5.73	56.460	349.005

Each figure is `(R at L=11 − R at L=1) / 10` from a measured pair of resistors, so the contact heads have already been subtracted out.

If you multiply each figure in the `xhigh` column by its width, you get 1999.80 Ω/□ for all five ($0.35 \times 5713.715 = 1999.80$, $5.73 \times 349.005 = 1999.80$), against `rsheet = 2000.0` in the model file. All five widths are the same material described by the same sheet resistance.

The same calculation on the `high` column gives 379.9, 325.8, 309.7, 300.8 and 323.5 Ω/□, which don't agree with each other. The narrow-width flavours are separately fitted devices, and `res_high_po_0p35` behaves like a 380 Ω/□ material rather than a 317 Ω/□ one. So if you move a design from one width to another, re-simulate it instead of scaling the old value by hand.

10 kΩ four ways

Here is the same 10 kΩ target in four flavours, with each length tuned until the simulator agreed:

Device	W (µm)	L (µm)	R measured	**area**
`res_xhigh_po_0p35`	0.35	1.647	9998.007 Ω	**0.576 µm²**
`res_high_po_0p35`	0.35	8.351	10000.90 Ω	**2.923 µm²**
`res_generic_nd`	1	85.1	9998.521 Ω	**85.1 µm²**
`res_generic_po`	1	196.4	10001.30 Ω	**196.4 µm²**

That's a factor of 341 in area for the same 10 kΩ, and it's the reason a process bothers to have a high-resistance implant at all: without one, a modest bias resistor is bigger than a logic gate.

For scale, here is the smallest inverter's size, from `/foss/pdks/sky130A/libs.ref/sky130_fd_sc_hd/lef/sky130_fd_sc_hd.lef`:

```
MACRO sky130_fd_sc_hd__inv_1
  SIZE 1.380 BY 2.720 ;
```

That's 3.7536 µm² for the smallest inverter in the standard-cell library, so the 10 kΩ `res_xhigh_po_0p35` is smaller than one inverter, while the `res_generic_po` version takes the area of fifty-two of them. Scale that up to 100 kΩ in ordinary poly (1964 µm², about 520 inverters) and the resistor is bigger than the circuit it biases, so choosing the resistor flavour is a design decision in its own right.

Capacitors

Capacitor table

Each capacitor was driven by a 1 V AC source at 1 kHz, and its value computed as $C = |I|/(2\pi f V)$ with $V = 1$ V:

Device	Geometry	C measured	density
`cap_mim_m3_1`	2 × 2 µm	9.302250 fF	2.326 fF/µm²
`cap_mim_m3_1`	10 × 10 µm	206.5822 fF	2.066 fF/µm²
`cap_mim_m3_1`	22.2 × 22.2 µm	1000.314 fF	2.030 fF/µm²
`cap_mim_m3_2`	10 × 10 µm	206.5822 fF	2.066 fF/µm²
`cap_vpp_11p5x11p7_l1m1m2m3m4_shieldm5`	11.5 × 14.7 µm (fixed)	147.3400 fF	1.095 fF/µm²

Device	Geometry	C measured	density
`nfet_01v8` as a MOS cap, gate at 1.8 V	10 × 10 µm gate	787.6883 fF	7.877 fF/µm²
the **same** device, gate at 0 V	10 × 10 µm gate	220.7182 fF	2.207 fF/µm²
`cap_var_lvt` at 1.8 V	10 × 10 µm	817.8665 fF	8.179 fF/µm²
the **same** varactor at 0 V	10 × 10 µm	99.51349 fF	0.995 fF/µm²

A few things in the table are worth pointing out:

1. `m3_1` and `m3_2` are the same capacitor built on different metals. `cap_mim_m3_1` sits between met3 and the `capm` plate, and `cap_mim_m3_2` sits between met4 and `cap2m`. Their capacitance is identical to seven digits, so you choose between them by what you need to route underneath.
2. MIM density rises as the plate shrinks, because the perimeter term is fixed per micron of edge and a smaller plate has more edge for its area: 2.326 fF/µm² at 2 × 2 µm against 2.030 at 22.2 × 22.2 µm.
3. A MOS capacitor is 3.8× denser than a MIM, but its value depends on the voltage across it: 787.6883 fF at 1.8 V and 220.7182 fF at 0 V, **a factor of 3.57 apart on the same physical device.** Use it for supply decoupling, where one plate is always at the rail, and never in a filter whose corner frequency you care about.
4. The varactor uses that nonlinearity on purpose. It goes from 99.5 fF to 817.9 fF, a factor of 8.22, over the supply range, which makes it the tuning element in an oscillator.

MIM capacitance from the model parameters

The MIM model comes down to three lines of `/foss/pdks/sky130A/libs.ref/sky130_fd_pr/spice/sky130_fd_pr__cap_mim_m3_1.model.spice`:

```
.param wc = 'w+m3_dw*1e6+tol_m3*1e6'
.param carea  = 'camimc*(wc)*(lc)'
.param cperim = 'cpmimc*((wc)+(lc))*2'
```

(`lc` is defined the same way as `wc`, on the line immediately after it.)

and three constants, which live in `/foss/pdks/sky130A/libs.tech/ngspice/r+c/res_typical__cap_typical__lin.spice` and `.../sky130_fd_pr__model__r+c.model.spice`:

```
+ camimc=  2.00e-15  ; Units: farad/micrometer^2
+ cpmimc = 0.19e-15 ; Units: farad/micrometer
+ m3_dw = -0.025u
```

So when you draw a plate 10 µm square, the metal etches back by 0.025 µm and the real plate is 9.975 µm on a side:

$$2.00 \times 9.975^2 + 0.19 \times 2 \times (9.975 + 9.975) = 199.00125 + 7.581 = \boxed{206.582\ \text{fF}}$$

ngspice measured `2.065822e-13`, which matches to six significant figures using only three constants and a subtraction. When a PDK number and your own arithmetic agree that closely, you can be confident you understand how the device is modelled.

What 1 pF costs

Device	Area for 1 pF	Caveat
MOS cap at 1.8 V	127 µm²	only if one plate stays at a rail
MIM	**493 µm²**: 22.2 × 22.2 µm, measured at 1000.314 fF	linear; the usual choice for a signal path
VPP fringe	913 µm²	it uses met3 and met4, the MIM's own layers, so it can't share area with a MIM

493 µm² is about the area of 130 inverters. It's the number to bring to any "just put a 1 nF cap on it" conversation, because a nanofarad of MIM is half a square millimetre, the area of about 130 000 inverters. What a picofarad costs goes through this at length.

Parasitic capacitance to substrate

Every layer has capacitance to the substrate, whether you asked for it or not. These values come from the `defaultareacap` lines in Magic's `/foss/pdks/sky130A/libs.tech/magic/sky130A.tech`, which gives them in aF/µm²:

Layer to substrate	aF/µm²	as fF/µm²
poly	106.13	0.106
local interconnect (li)	36.99	0.037
met1	25.78	0.026
met2	17.5	0.018
met3	12.37	0.012
met4	8.42	0.008
met5	6.32	0.006
n-well / deep n-well	120	0.120

So a 100 µm × 1 µm met1 wire carries 2.6 fF to ground, which is more than a quarter of a 10 fF capacitor you might draw on purpose elsewhere. Parasitic capacitance covers this in more detail, including the sidewall terms that this table leaves out.

Inductors

SKY130 has no good inductor. It ships three spiral models (`sky130_fd_pr__ind_03_90`, `ind_05_125` and `ind_05_220`). `ind_05_220` is 9.92 nH, occupies tens of thousands of µm², and has **Q = 0.0000150516 at 1 kHz**. The best Q any of the three ever reaches is **21.1, at 6.38 GHz**, and above 3.52777 GHz the reactance of `ind_05_220` changes sign. Inductors on a chip and What to build instead of an inductor cover why, and what replaces it.

Corners

`tt` is the typical corner, for transistors and passives alike. Passives also have corners of their own, selected by the same word at the end of the `.lib` line: `ll` (low resistance, low capacitance) and `hh` (high resistance, high capacitance), plus `hl` and `lh`.

We measured these spreads rather than reading them off a parameter file, by simulating the same four devices three times and changing only the corner name:

Device	ll	tt	hh	spread about tt
res_high_po W=1 L=10	3106.637 Ω	3550.443 Ω	3994.248 Ω	**±12.5 %**
res_xhigh_po W=1 L=10	18037.95 Ω	21221.12 Ω	24404.29 Ω	**±15.0 %**
res_generic_po W=1 L=10	451.1426 Ω	509.2314 Ω	583.1667 Ω	−11.4 % / +14.5 %
cap_mim_m3_1 10 × 10	179.7322 fF	206.5822 fF	233.8667 fF	−13.0 % / +13.2 %

The parameters behind those runs are set in /foss/pdks/sky130A/libs.tech/ngspice/r+c/res_low__cap_low.spice, res_typical__cap_typical.spice, res_high__cap_high.spice and their __lin siblings:

Parameter	ll	tt	hh
rp1 (ordinary poly)	44	48.2	53.52
rdn (n+ diffusion)	111.6	120	128.4
rdp (p+ diffusion)	175.3	197	218.7
rl1 (local interconnect)	10.31	12.2	14.02
rm1 (metal 1)	0.111	0.125	0.139
rnw (n-well)	1378	1700	2022
sky130_fd_pr__res_high_po__var	−0.125	0.0	+0.125
camimc (MIM)	1.778e-15	2.00e-15	2.231e-15

A device doesn't always move by exactly its parameter's percentage. rp1 shifts by −8.7 % and +11.0 %, but a real res_generic_po moves by −11.4 % and +14.5 %, because the corner also changes the etch bias that sets the strip's effective width, so **simulate the corner rather than scaling the typical answer by hand.**

Every device measured above moves by more than ±10 % between corners. A *ratio* of two of the same device stays put, which is why analog design is built on ratios; see Matching and Pelgrom's law.

Re-running these numbers

The numbers come from two sets of decks in labs/passives-decks:

```
cd labs/passives-decks
make           # the guide pages' decks: head/body, corners, MIM, spirals, mismatch
make catalogue  # this page's extra devices: the seven materials at one square,
               # both fixed-width families, 10 kΩ four ways, and the MIM/varactor rows
```

make compares its results against 45 stored values and ends with PASS or FAIL. make catalogue prints the numbers from spice/catalogue.spice for you to read against the tables above, and it takes about a minute.

You don't need any environment setup, because every deck names the model library by its absolute path, so a bare container works. The related labs are Lab 01: A resistor you designed, Lab 02: How big is a picofarad? and Lab 04: Two resistors that disagree.