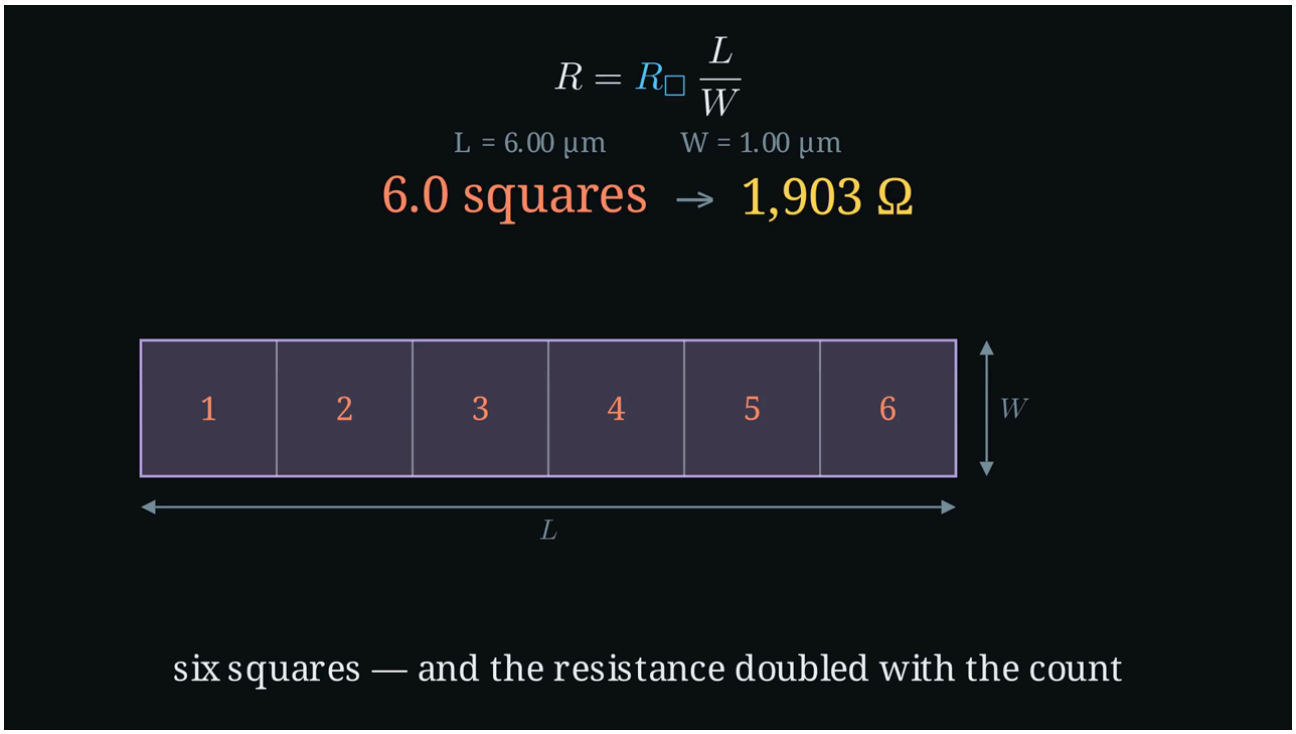
Resistors on a chip

In a first circuits course such as ECE110, someone handed you a resistor. It came out of a drawer with four coloured bands on it, and someone in a factory had already decided it was 10 kΩ, so your job was to put it in a mesh and write KCL. Every equation you learned there still holds on a chip.

What changes is where the 10 kΩ comes from. There is no drawer of parts on a chip. A chip is one crystal of silicon with patterns printed on it, and the only thing you can decide is what shape to print. So "how do I get 10 kΩ?" becomes a question about geometry, and because geometry costs area and area costs money, it's a question about cost as well. That question is the subject of AD102, and once you can answer it, the rest is arithmetic.

Animation: resistor geometry

This clip runs for 192 seconds and has no sound; the captions carry the narration. It covers this page and the next two (the symbol, the strip, ohms per square, and what 100 kΩ costs in silicon), so come back to it after you've read those as well.



Your browser will not play embedded video — download the clip.

Source: `tooling/manim/scenes/ad102_resistor_geometry.py`, the `ResistorGeometry` scene. Every number shown in it is one this course measured itself, and the two footprints in the closing comparison are drawn to the same scale.

Before you press play, make a prediction. A strip of poly 40 µm long and 2 µm wide has some resistance. If you redraw it 40 µm long and 4 µm wide, does the resistance go up, go down or stay the same, and by how much? Write your number down, because the animation counts it out on screen and you can check it there.

What the resistor is made of

Without its packaging, a resistor is a piece of material that current has to squeeze through. A long, thin piece has a high resistance, and a short, wide piece has a low one. The through-hole resistor you soldered works the same way: inside it is a spiral of resistive carbon film cut into a ceramic rod, and the spiral is cut longer to give more ohms.

On a chip, the resistive material is a **strip of doped silicon or polysilicon**, which you draw as a rectangle on one mask layer. Current goes in at one end and out at the other, and three things set the value:

1. What the strip is made of, which comes down to how heavily it was doped. You don't choose this; the foundry's recipe fixes it.
2. How long you draw it.
3. How wide you draw it.

You control the length and the width directly, but for the material you can only choose from a menu that the foundry provides.

The SKY130 resistor menu

Here is that menu, taken from the header comment of the Magic technology file, `/foss/pdks/sky130A/libs.tech/magic/sky130A.tech`:

```
# sky130_fd_pr__res_generic_nd   rdn      n+ diff resistor
# sky130_fd_pr__res_generic_pd   rdp      p+ diff resistor
# sky130_fd_pr__res_generic_l1   rli      local interconnect resistor
# sky130_fd_pr__res_generic_po   npres    n+ poly resistor
# sky130_fd_pr__res_high_po_*    ppres    p+ poly resistor (300 Ohms/sq)
# sky130_fd_pr__res_xhigh_po_*   xres     p+ poly resistor (2k Ohms/sq)
# sky130_fd_pr__res_iso_pw       rpw      pwell resistor (in deep nwell)
```

Those seven entries are seven materials rather than seven values. The two `p+ poly` entries, `res_high_po` and `res_xhigh_po`, are on the same layer with the same drawing rules, and they differ by a factor of about 6.7 only because they get a different implant dose. That factor is the only choice of value the foundry gives you; everything else you get by drawing.

`res_high_po` and `res_xhigh_po` are also the **precision** resistors. SkyWater characterised them carefully because analog designers build references out of them, and they're the two this course uses.

The same strip in four materials

Suppose you draw one strip, 314.046460 µm long and 1 µm wide, and change nothing but the resistor model it uses. Section C of `spice/r_head_and_body.spice` does exactly that:

```
--- C. one 314 um x 1 um strip, four implants (ohms) ---
r_high = 1.000000e+05
r_xhigh = 6.653872e+05
r_pdiff = 6.138915e+04
r_ndiff = 3.689775e+04
```

model	value of that one strip
`sky130_fd_pr__res_high_po`	**100.0000 kΩ**
`sky130_fd_pr__res_xhigh_po`	665.3872 kΩ
`sky130_fd_pr__res_generic_pd` (p+ diffusion)	61.38915 kΩ
`sky130_fd_pr__res_generic_nd` (n+ diffusion)	36.89775 kΩ

Every row has the same drawing and the same silicon area, and yet the values spread over a ratio of 18:1. Choosing the material is the first design decision you make, and it's the only one that doesn't cost area. You'll run this deck yourself in Lab 01, Part 6.

When you do, the run first prints a block of model warnings (see Getting started). One of them, the `rbody_model` warning, shows `rsh` as `3.173885e+02`. That number is the sheet resistance of `res_high_po` in ohms per square, which is the most important parameter of the device, and it sits in a warning you would normally scroll past.

Writing W and L

As Getting started explained, `W` and `L` on a SKY130 device are plain numbers in microns, with no unit suffix:

```
Xa a 0 0 sky130_fd_pr__res_high_po W=1 L=10       <- one micron by ten microns
Xa a 0 0 sky130_fd_pr__res_high_po W=1u L=10u     <- a millionth of that, in both
```

Every deck in this course uses the first form. The second multiplies both numbers by 10^{-6}, which puts the device outside every bin its model was fitted over. On a MOSFET that stops the run with `could not find a valid modelname`, but on a resistor the run carries on, which is what makes the mistake so hard to catch: the strip that reads 3550.443 Ω when it's written `W=1 L=10` reads 3193.812 Ω when it's written `W=1u L=10u`, an ordinary-looking resistance 10 % off, and ngspice still exits with status 0. Apart from Lab 01's `u_trap.spice`, which exists to show you the mistake, no deck in this course has a `u` in it, and none of yours should either.

Next: Ohms per square, the unit that turns a rectangle into ohms.