Getting started

Before you start Lab 01 you need three things: the workbench, this course's files inside it, and Magic opening with the SKY130 technology loaded. The third is the one that usually goes wrong, and when it does, the message you get doesn't tell you why. This page explains how the pieces fit together. You won't set anything up here: IC101 sets up the workbench, and Lab 01, Part 1 loads this course into it.

Prerequisites

IC101 ⟶ AD101 ⟶ AD102 ⟶ AD103.

Of the four, AD103 matters most. You should have drawn a CMOS inverter in XSchem and swept its transfer curve, and you should be comfortable with the idea that W and L are numbers you choose. AD104 takes that same inverter and asks where the silicon goes.

The workbench

Everything in this course runs in the noVNC desktop from IC101, which is the image `hpretl/iic-osic-tools:2026.08`. You start it the way IC101 showed you, open `http://localhost` in a browser, and type commands in the terminal on that desktop.

The image can work with more than one manufacturing process, and the environment variable `$PDK` chooses which one. This course needs **`sky130A`**, but the image ships with `PDK=ihp-sg13g2`, a real 130 nm process from a different foundry, with its own layer names and its own rules. Magic opens in IHP's process without complaint, so the first sign that anything is wrong is that none of the layers this course names are there.

The course environment script, `/foss/designs/common/.designinit`, switches the whole shell to SKY130, and it also defines `mod`, the command that copies a course's files into your workspace. The change lasts only as long as that shell, so you load the script once in every new terminal you open.

Magic

Magic is older than most of the people taking this course: it started at Berkeley in 1983, and it looks it. It is also the only open-source layout editor with a live design-rule checker, a built-in extractor and a scripting console, which is why the open SKY130 ecosystem is built around it.

Whenever a lab has you open Magic, you do it with `make edit`, run from inside the lab package. From Lab 02 on, that starts Magic in the folder that holds the `.mag` file and passes it the cell's name without the `.mag` extension; in `labs/lab-03-draw-an-inverter/mag`, for example, it runs `magic -d X11 my_inv`. Lab 01′s cell is the foundry's, so there `make edit` hands Magic the full path of the file in the PDK instead.

Two windows appear: **`layout1`**, which shows the drawing, and **`tkcon`**, a Tcl console. Most of your work in this course happens in the console. Every Magic command in the guide pages and labs can be typed there, and the lab scripts drive Magic by feeding it the same commands.

The `-d X11` option

Magic has three graphics drivers in this image, and `X11` is the one that works. The Cairo/OpenGL driver (`-d XR`) can fail with a `BadAlloc` inside the container, and when it does you get a black window rather than no window at all, so it takes a while to work out what went wrong. The team's `.designinit` defines a `magic130` alias that uses `-d X11` for the same reason.

Where Magic gets the technology

When Magic starts, it decides which technology to load from two startup files:

- The image's own `~/.magicrc` loads whichever PDK `$PDK` names. With the image's default `PDK=ihp-sg13g2` you get IHP's layers and none of SKY130′s.

- A file called `.magicrc` in **the directory you start Magic in**. Every lab package has one at its top level, and it loads SKY130 only if nothing has loaded it yet; if `~/.magicrc` already did, it does nothing. The check is there because a `.magicrc` that loads SKY130 a second time makes Magic print `Error parsing ".magicrc": can't rename to "closewrapperonly": command already exists` and `Bad local startup file ".magicrc", continuing without.` in the first two lines of the console, which makes a toolchain that is working fine look broken.

Every lab's Makefile also sets `PDK=sky130A` for the Magic it starts, so `make edit` opens in SKY130 even if `.designinit` failed for you.

If no technology file gets loaded at all, Magic falls back on its own default technology, `scmos`, which has about a dozen layers and none of SKY130's. You'll see either `Failed to load technology`, or a Magic that opens normally but has no `nwell` in its layer list.

What the labs give you

Every lab is a folder with a `Makefile`, and typing `make` on its own runs every check and prints the result:

`make`	run every check and print `PASS` or `FAIL` with a reason
`make edit`	open the cell you are working on in Magic
`make clean`	delete `results/`

Because the Makefiles set `PDK=sky130A` themselves, **the labs work even if `.designinit` did not**. If a setup step in Lab 01 doesn't work for you, run `make` in the lab anyway: it checks everything itself and tells you in plain words what it found.

Getting help

Ask in the team Discord. Layout is a craft, and it's quicker to have someone show you than to read about it. A screenshot of your Magic window, with the DRC count in the toolbar visible, is usually enough for someone to spot the problem.

When a tool reports a problem, paste its exact words. `drc why` names a rule and netgen names a net, and those names are specific enough that someone helping you can search for them, which they can't do with "it doesn't work".

Problems opening Magic

Setup problems with `.designinit` and `mod` are covered at the step where they happen, in Lab 01, Part 1.

What you see	What it means
`Failed to load technology "sky130A"`	No SKY130 startup file was loaded. Open the cell with `make edit` from its lab folder, or copy a lab's `.magicrc` into the folder you start Magic in.
Magic opens, and there is no `nwell` in the layer list	Same cause: the technology did not load.
`Bad local startup file ".magicrc", continuing without.`	A `.magicrc` loaded SKY130 a second time. Use a copy of a lab's `.magicrc`, which checks whether SKY130 is already loaded first.
`Cell ... couldn't be read`	You named a cell that does not exist in this folder, or you are in the wrong folder. Magic then creates an

What you see	What it means
	empty cell with that name, which is why an empty window is a common symptom.
The layout is one empty rectangle with a name in it	That's an unexpanded subcell. Select it and press `x`, or type `expand`.
Everything looks dim and the title says `[NOT BEING EDITED]`	You are looking at a cell that is not the edit cell. Typing `load <name>` makes it the edit cell.