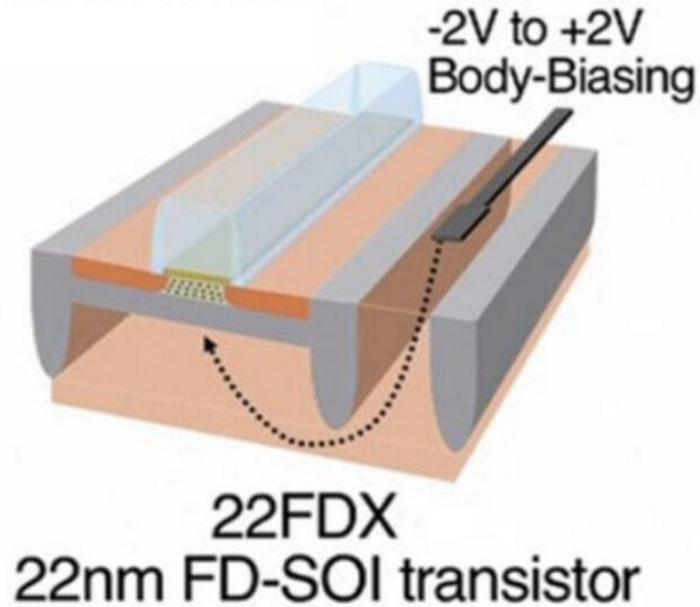




GLOBAL  
FOUNDRIES



## GF22FDX TUTORIAL

Tips & Tricks to Surviving GF22FDX

### Overview

This tutorial will provide guidance for working in the GF22FDX PDK. My hope is that the information provided here will help save you time and let you spend more time working on your design rather than fighting with the tools & PDK.

Michels, Noah  
nmichels@umich.edu

## Table of Contents

Welcome to the PDK .....	2
Metal Stack Version.....	2
EXT Vs. Plus .....	2
PeakView & RF .....	2
Useful Locations .....	3
Standard Cells & Pads .....	4
Digital Flow .....	4
Pads .....	4
PeakView .....	5
How to Import PeakView Devices .....	5
PeakView Vs. Ideal Performance Differences.....	5
How to LVS PeakView Devices.....	5
Layout Tips .....	6
Modify RF FET Pcells .....	6
Layout Suggestions.....	6
Top Level .....	6
chipbreak.....	6
FILLGEN.....	6
Other Tips, Tricks, & General Annoyances.....	7
Running PEX .....	7
Exporting Design .....	7
DRC Licensing Issue .....	7

## Welcome to the PDK

**Note:** Some file locations may not be general enough to be directly applicable to your design, so be sure to keep your working directory in mind. Also know that some file locations or file structures may have changed since this document was first made.

First, the basics. The “FD” portion of “FDX” stands for “fully-depleted.” GF22FDX is a Fully-Depleted Silicon-on-Insulator (SoI) technology. This refers to the ability of the transistor channel to be fully depleted of carriers. Compared to traditional technologies, FD-SoI enables much better transistor characteristics such as lower parasitic capacitance and lower leakage current. FD-SoI also allows for forward body biasing (FBB) and reverse body biasing (RBB) for precise control over transistor threshold voltages and related performance metrics. Refer to the figure below for further detail:

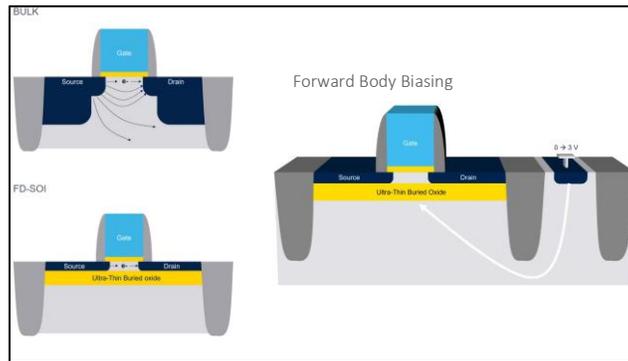


Figure: Bulk Vs. FD-SoI Technologies

Now let’s get into some of the specifics about our setup in this PDK.

### Metal Stack Version

There are a lot of available metal stacks that you can use but currently the University Partnership Program (UPP) only allows for two of them to be tapedout: Metal Stack 11 (MS 11) & Metal Stack 19 (MS 19).

- Metal Stack 11: 9M\_2Mx\_5Cx\_1Jx\_1Ox\_LB
- Metal Stack 19: 10M\_2Mx\_5Cx\_1Jx\_2Qx\_LB

Metal Stack 19 has an additional metal layer but seems to lose some of the `cmos22fdsoi_rf` components (specifically the inductors). So, if you plan to use rf components, I recommend using Metal Stack 11.

### EXT Vs. Plus

The next thing to consider when setting up your PDK is the choice between EXT & Plus. Plus is technically the most up to date version of the PDK while EXT has a focus on “automotive” applications. While plus is more up to date, I recommend using the extended (EXT) version of the PDK as this will give you access to PeakView. If you don’t think you’ll use PeakView then it is probably fine to use Plus. Note that my experience is in using EXT and this document will assume that you using EXT for the remainder of the discussion.

### PeakView & RF

As previously mentioned, PeakView is only available in the EXT version of the PDK. GF has technically dropped support for PeakView at the time of writing this document so you will likely not be able to get this for the Plus version of the PDK. PeakView will allow you to simply design inductors and transformers for use in this PDK. More details on the use of this software will be discussed in a PeakView section.

## Useful Locations

While rarely used, the first thing you should get access to is the Global Foundries Portal. For me, this was done by emailing our GF point of contact ([vikas.vijayakumar@gf.com](mailto:vikas.vijayakumar@gf.com)) and requesting access. This site houses a lot of documentation but I think the most relevant documentation is still found on our servers.

There are two general locations you should know about for all of the PDK technology:

- [/afs/eecs.umich.edu/kits/GF/22FDX/2024.09/22FDX-EXT/V1.0\\_4.1/](#)
  - Location of all GF material related to this PDK
- [/afs/eecs.umich.edu/kits/Synopsys/GF22FDX/2023.06/](#)
  - Location of all the Synopsys material related to this PDK

There is a lot of material in each of these locations, so let me narrow down where you should look and explain some things. Let's start with the **GF** material. The main file you'll want to access again and again is the documentation index:

- [/afs/eecs.umich.edu/kits/GF/22FDX/2024.09/22FDX-EXT/V1.0\\_4.1/doc/pdf/22FDX-EXT\\_PDK\\_Document\\_Index.html](#)

This file provides links to all of the critical documentation in this PDK along with descriptions of these files. The main files you'll be interested in are:

- Design Manual
- Library Manual
- Model Reference Guide
- RF Model Reference Guide

You'll likely explore some of the other files too as you need them (such as FILLGEN for Calibre or other parameter and tool documentation). Note that I highly recommend reviewing the metal layer & via resistances and capacitances in the design manual prior to starting any layout to get a better understanding of best layout practices in this PDK.

Within the main GF directory, you'll also find other useful folders but most of these should be self-explanatory when you need them.

Now let's discuss the **Synopsys** directory. When you first open the directory, you'll see a few similar files. They can be broken up as follows:

- **Pads:**
  - **Inline IO:** `dwc_io_gf22fdx_1p8v_gpio_i_ag1_fdk_2.01a`
  - **Staggered IO:** `dwc_io_gf22fdx_1p8v_gpio_s_ag1_fdk_2.01a`
- **STD Cells:**
  - **SLVT:** `gf22nsdllogl20edl116a`
  - **LVT:** `gf22nsdvlogl20edl116a`
  - **RVT:** `gf22nsdslogl20edl116a`

Note that I have imported the Inline IO Pads & the LVT Pads into my Cadence environment. This was a bit finicky and not everything has all of the views (Pads only have Layout & Symbol while STD cells have layout, schematic, & symbol). We will get more into this in the next section

## Standard Cells & Pads

While we have imported some LVT STD cells into Cadence, the STD cells are setup to be much more friendly with a digital flow than with a manual place and route. For example, the standard cells will not pass DRC on their own. They require well tap cells to be placed along with outer ring cells. At time of writing, a digital flow is being setup by Peter and the digital section will be updated upon completion.

### Digital Flow

### Pads

There are a couple of things to note with the pads in this PDK. The first is that you can use either the inline or the staggered. The staggered are more dense but in general you should expect to use the inline as it is the most standard method of doing pads.

The pads in this PDK are provided by Synopsys but have to be used in conjunction with a GF cmos22fdsoi cell called “pad” which the user will have to manually place on top of the Synopsys pad using details provided in the documentation. It is important to note that the GF “Pad” cell is parameterized into a different layout shapes set by the “Pad Type.” By default, it is set to Bump, but you will most likely want to set it to WireBond. This is demonstrated in the figure below:

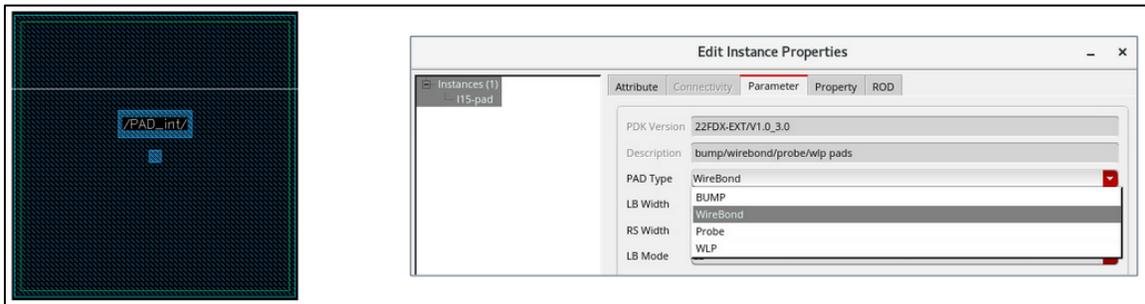


Figure: GF “Pad” Cell Parameterization

The pad cells fit together relatively simply but you will need to add extra filler cells (provided by Synopsys) between the Synopsys IO to make space for the GF pads.

The pads will pass DRC but will require some extra effort to pass LVS in Cadence. This is because we could never determine a way to import the netlists properly. The issue we encountered was with all of the cmos22fdsoi resistor components getting imported as something just called “resistor” and not its actual name. No method was found to fix this.

You will need to manually export your top-level netlist and add a reference to point towards the pad netlists (CDL) files. You will also need to add schematics to the Pads that you use that contain the pins of the pad prior to doing this. This can be done by creating cellviews from symbols for each of the used pads and then deleting duplicated pins. This can be a little tedious on initial setup but shouldn’t take too long.

Similarly, you will need to export and simulate your netlists with pads to see the performance with the pads. I couldn’t find a way to do this within Cadence.

## PeakView

PeakView is very useful in this PDK if you want to generate an inductor or transformer. The provided inductors and transformers are very limited in the range of specs they can be set to. There are few things you should know about when using PeakView in this PDK.

### How to Import PeakView Devices

In general, the simplest way to use PeakView is with the Inductor/Transformer Design Wizard. This will provide you with a number of similar designs all trying to meet the specs you have set. The designs will appear in the Xmfr Wizard window, but probably won't simulate properly here. My process is as follows:

1. Launch Inductor Design Wizard from Create tab
2. Set desired specifications & select OK
3. Stop simulation in Xmfr Wizard
4. Delete any designs that are "out of range for length in line"
5. Copy parent cell to Main window
6. Simulate parent in main
7. Review specs to determine which designs you want
8. Copy one at a time the design you want from xmfr wizard to the main
  - a. Delete unwanted designs from main & delete everything from wizard
9. Resim wanted designs in main (should finish fast)
10. Export wanted design one at a time to cadence from Main

You can then insert these cells into your schematics in Virtuoso using hspiceD or Spectre views.

### PeakView Vs. Ideal Performance Differences

Because the exporting process from PeakView to Cadence can be tedious, it is desirable to test designs in schematic using ideal components that attempt to replicate the reported performance of PeakView devices. Unfortunately, the resulting performance using ideal components are often much better than what you will get with PeakView. This is the case even with modeling the exact quality factor, inductance, coupling, and parasitic capacitances. You should expect a significant degradation of performance when using PeakView devices. I think this is most likely due to parasitic capacitances that I haven't been able to model with ideal components to match the PeakView devices perfectly.

### How to LVS PeakView Devices

To LVS PeakView devices, you should follow the PowerPoint documentation uploaded alongside this document. Note that there is documentation that differs for each PDK.

## Layout Tips

This is a collection of my tips for avoiding annoyances with layout (primarily with the RF FETs).

### Modify RF FET Pcells

While the changing of width, length, and fingers should be clear enough, there are some behind the scenes settings you will also want to adjust. This is done by selecting the Display Group at the bottom of the object properties:

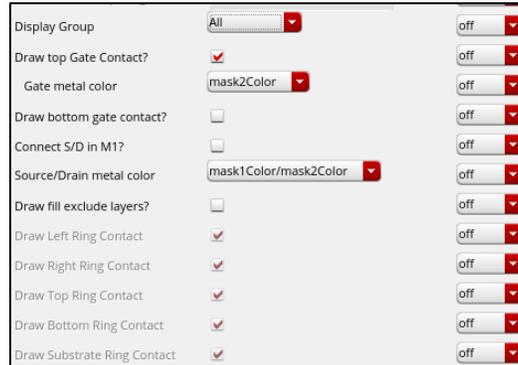


Figure: Edit Object Properties Display Group “All”

The main settings you will be interested in are the drawing of gate contacts on top or bottom (so you can connect to the gate of your pcell) and the connecting of S/D in M1 (so it will pass LVS prior and allow for simpler connections to source and drain).

You can also modify what guard rings are included (depending on which version of cell you are using). It can sometimes be beneficial to remove some rings, particularly if you are not using FBB/RBB. It will allow you to have a much denser design.

**NOTE:** Never manually remove the guard rings of a Pcell (via flattening)!. It will not pass LVS (according to documentation)!

### Layout Suggestions

I also recommend using at least 3 fingers (technically 8 is the minimum that has been modeled). It is also simpler if you use a minimum length of 20n instead of 18n. These are things that may hurt your performance slightly but will make your life a lot easier later on during layout.

## Top Level

### chipbreak

The pad ring setup should be pretty self-explanatory from the documentation. The main thing you should know is that you will have to add a chipbreak pcell on top of the design. This will reduce the total area you have available. So if you have a 2mm\*2mm chip, don't just make a 2mm\*2mm pad ring. First place chipbreak and then design the pad ring within it.

### FILLGEN

There are three FILLGEN tools but we only have necessary tools to run the Calibre flow. You will inevitably encounter “known issues” related to JQ density and CA density. Both of these can be waived. Do not waste your time trying to fix them.

## Other Tips, Tricks, & General Annoyances

Here is a collection of seemingly small issues I ran into that cost me days to fix during a tapeout. Enjoy.

### Running PEX

PEX is not very friendly in this PDK. We don't have the normal PEX method and instead you should use xACT. To use it, you will need to manually export your design from the CIW and use that GDS. You will need to add additional settings such as "LVS DEVICE PUSH" and pretty much modify everything possible to have PEX not run LVS or look at the schematic. You will want to PEX with the layout as the source.

### Exporting Design

Never export from layout. Export from the CIW File->Export->Stream/Oasis. Exporting from the layout causes issues with coloring not exporting properly.

### DRC Licensing Issue

If you run into an issue with not being able to run DRC due to a lack of licenses, it is likely due to something called calmpgold which we only have 10 licenses of. I recommend using the following command:

```
lmstat -a -c $LM_LICENSE_FILE
```

This should bring up a list of all the licenses. Find the calmpgold and see who is using it. Email them and beg them to share.