A Brief Excerpt on Mixed Signal Circuits for Xschem

Synthesize Digital Components

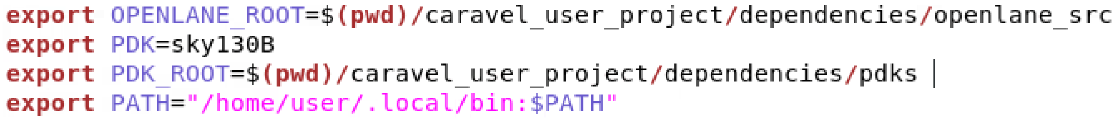
Update toolchain and PDKs

The first step is to ensure the environment dependencies are updated. At the time of writing, it can be confirmed this mixed-signal method works with Toolchain version 1.0.4. In terms of the PDK, setup detailed in this document was used: Precheck Guide.

Create digital components with OpenLane

Next, harden your digital macro with OpenLane. In this environment, the easiest way would be to follow the typical digital flow for the Carvel User Harness. To do so, you can clone and setup the following Caravel User Harness git repo: https://github.com/efabless/caravel.

Assuming the repository and PDKs have been setup properly you'll want to ensure the following environment variables have been set:



```
export OPENLANE_ROOT=$(pwd)/caravel_user_project/dependencies/openlane_src
export PDK=sky130B
export PDK_ROOT=$(pwd)/caravel_user_project/dependencies/pdks
export PATH="/home/user/.local/bin:$PATH"
```

You should then be able to follow the digital workflow as stated in the OpenLane documentation here: https://openlane.readthedocs.io/en/latest/usage/index.html
And this document from sddec23-06 detailing the digital flow using the Caravel User Harness: https://sddec23-06.sd.ece.iastate.edu/reports/492/UserGuide.pdf

Verilog to Combinational Logic Circuitry

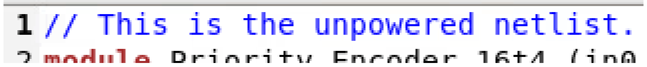
This process is relatively straight-forward and a guide by Stefan Schippers, the author of Xschem, can be found in this Slack discussion here: https://open-source-silicon.slack.com/archives/C017P3RAD42/p1715774213341429?thread_ts=1715769291.518399&cid=C017P3RAD42

The following is a brief guide based on that video.

1) Once you've run through the digital design process, OpenLane should have spat out a gate-level Verilog netlist. This can generally be found in the following directory:

```
/caravel_user_project/openlane/MODULE/runs/MODULE/results/final
/verilog/gl/
```

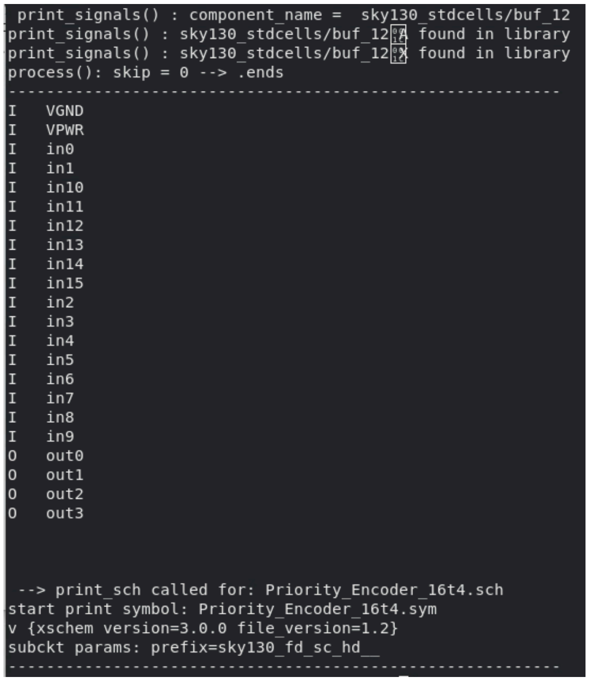
2) From here, there may be two files.
   `MODULE.nl.v` and `MODULE.v`
   You'll want to take the powered netlist, and this is usually `MODULE.v`. You can verify this by comparing the two files. The unpowered netlist should have the following comment.



```
// This is the unpowered netlist.
module Priority_Encoder_16t4 (in0
```

3) Copy this netlist file into your Xschem directory (or wherever your .xschemrc)

4) Run awk script with the Verilog netlist with the following command
   `./make_sky130_sch_from_verilog.awk MODULE.v`
   This will generate the schematic and subsequent symbol for your module.
   You should see something similar to the following should the operation be successful.

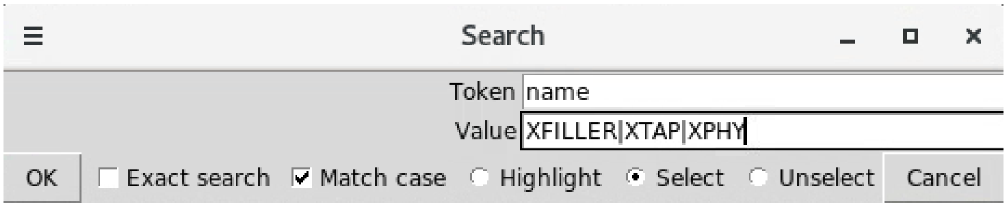


```
 print_signals() : component_name =  sky130_stdcells/buf_12
print_signals() : sky130_stdcells/buf_12␜A found in library
print_signals() : sky130_stdcells/buf_12␜X found in library
process(): skip = 0 --> .ends
---------------------------------------------------------
I   VGND
I   VPWR
I   in0
I   in1
I   in10
I   in11
I   in12
I   in13
I   in14
I   in15
I   in2
I   in3
I   in4
I   in5
I   in6
I   in7
I   in8
I   in9
O   out0
O   out1
O   out2
O   out3



 --> print_sch called for: Priority_Encoder_16t4.sch
start print symbol: Priority_Encoder_16t4.sym
v {xschem version=3.0.0 file_version=1.2}
subckt params: prefix=sky130_fd_sc_hd__
---------------------------------------------------------
```

5) Open your schematic in Xschem and remove non-functional cells like fillers and taps as such:
   - Select all non-functional cells by "CTRL+F" and filling out the following form

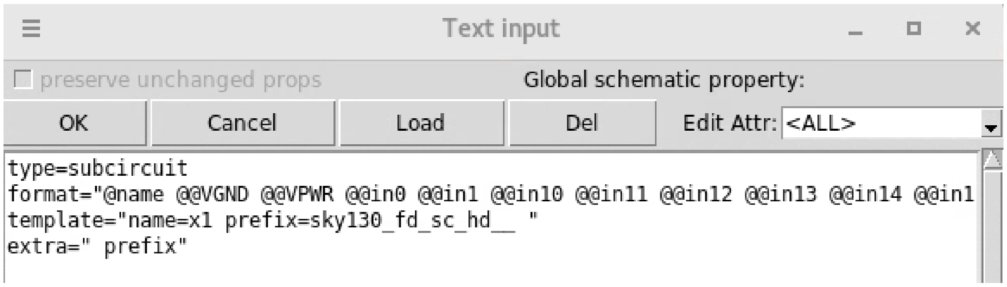


Token: name
Value: XFILLER|XTAP|XPHY

□ Exact search ☑ Match case ○ Highlight ● Select ○ Unselect

   - Once selected, you can delete these cells

6) Open your symbol and change your symbol type from “primitive” to “subcircuit”
    - You can edit this attribute by pressing “Q” in the .sym folder to pull up the following dialog



☐ preserve unchanged props

Edit Attr: <ALL>

```
type=subcircuit
format="@name @@VGND @@VPWR @@in0 @@in1 @@in10 @@in11 @@in12 @@in13 @@in14 @@in1
template="name=x1 prefix=sky130_fd_sc_hd__ "
extra=" prefix"
```

7) Finally, when you place the attributed symbol in a testbench, ensure that you place a `code.sym` with the following input:

```
name=STDCELLS
only_toplevel=true
format="tcleval( @value )"
value=" ** opencircuitdesign pdks install
.include
$::SKYWATER_STDCELLS/sky130_fd_sc_hd.spice
"
spice_ignore=false
```