**Inverter layout with Magic**

*Abstract*—This is a documents illustrate how to do the physical layout of an inverter cell in Magic.

Keywords—physical layout, Magic, scmos, backend VLSI, layout designing

**What is Magic ?**

Magic is an interactive system for creating and modifying VLSI circuit layouts. With Magic, you use a color graphics display and a mouse to design basic cells and to combine them hierarchically into large structures. For more information visit “opencircuitdesign.com”

**How to install and run Magic ?**

Magic can be installed in a number of ways but the easiest would be to just use

sudo apt-get install magic

For debian linux systems and do the same in cygwin for windows systems.

**How to move around Magic ?**

In Magic the box and the cursor are used to select things on the layout display. If you press the left mouse button and then release it, the box will move so that its lower left corner is at the cursor position. If you press and release the right mouse button, the upper right corner of the box will move to the cursor position, but the lower left corner will not change.

There are also different shortcuts to do things. For example, if you want to fill the box area with a layer, you can either type ‘paint <layer>’ or move the cursor to the layer and middle click to select and fill the box with that layer. Also, z is for zoom-in and shift + z is for zoom-out.

# 

**About the design**

This inverter design was done with scmos version 8.2.8 from MOSIS CMOS technology. For an inverter we need a PMOS and NMOS S-D-D-S connection. Here the local interconnect layer is not available so I used metal1 for all source-drain contact. We can make this cell compact if we can use the Li layer.

In this circuit, port and labels are assigned to input ‘x’ , output ‘y’ , Power ‘vdd’ & ground ‘vss’. I have also extracted a spice file from this layout file for simulation purposes. For this we need to use layout command ‘extract all’ to first convert our layout to a magic only file format and than use this format to create spice format with ‘ext2spice’ command.

This means using technology provided by MOSIS we can create out own standard cell library in magic.

**Design flow with qflow**

*Abstract*—This is a documents illustrate how to go from RTL to GDS with qflow. This includes various open source softwares.

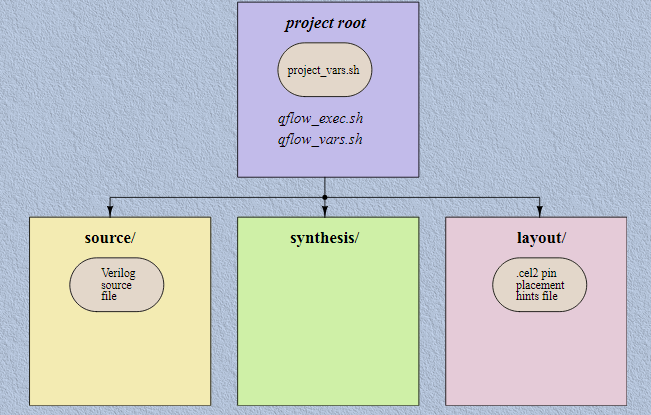
Keywords—physical layout, qflow, yosys, graywolf, qrouter, magic, vesta, open sta, backend VLSI, layout designing

**What is qflow ?**

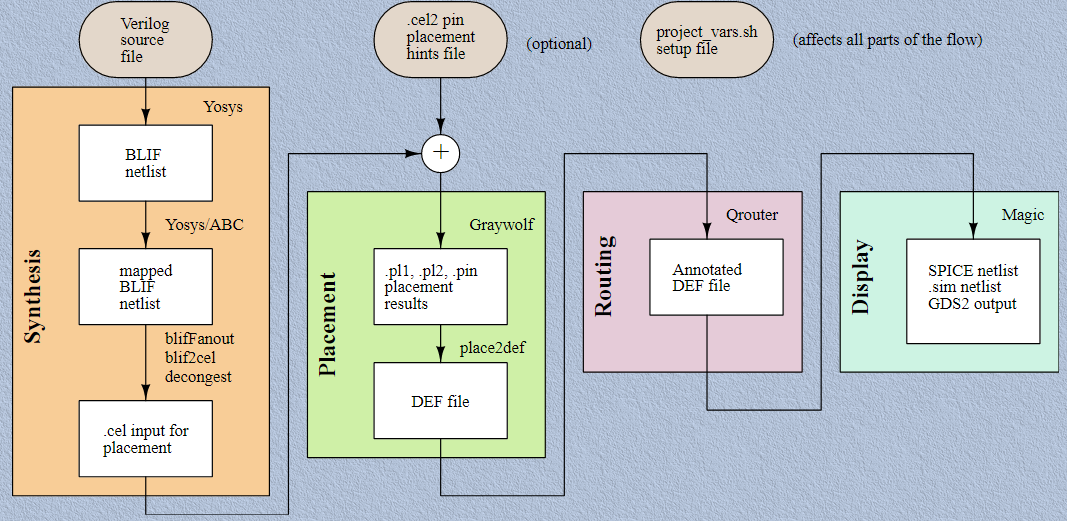
Qflow is a complete tool chain for synthesizing digital circuits starting from verilog source and ending in physical layout for a specific target fabrication process.

A digital synthesis flow is a set of tools and methods used to turn a circuit design written in a high-level behavioral language like verilog or VHDL into a physical circuit, which can either be configuration code for an FPGA target like a Xilinx or Altera chip, or a layout in a specific fabrication process technology, that would become part of a fabricated circuit chip.

Following image illustrates the directory hierarchy with qflow



Qflow has a GUI interface to simplify the whole process of running synthesis, place, and route, and all of the other related processes. Following image illustrates the flow of design in qflow

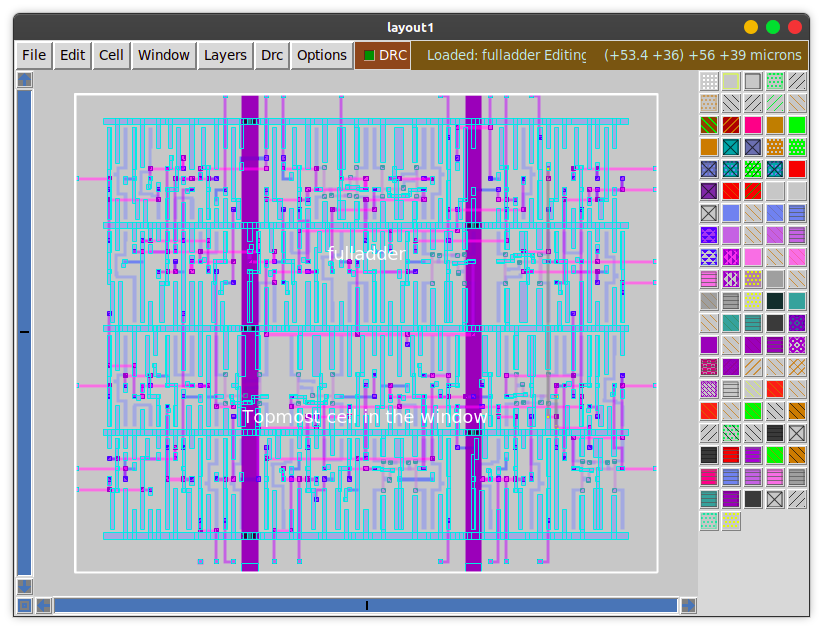


There are different stages of qflow as we can see from the images this can also be seen in qflow\_exec.sh as follows,

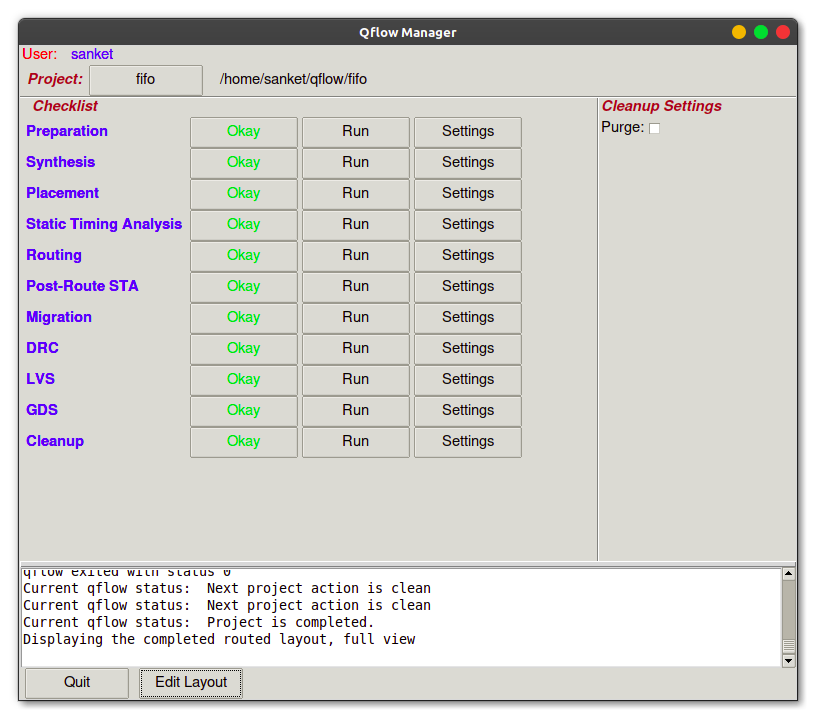
* Synthesize.sh
* Vesta.sh
* Placement.sh
* Router.sh
* Vesta.sh -d
* Migrate.sh
* Drc.sh
* Lvs.sh
* Gdsii.sh

**Fulladder and FIFO design with qflow**

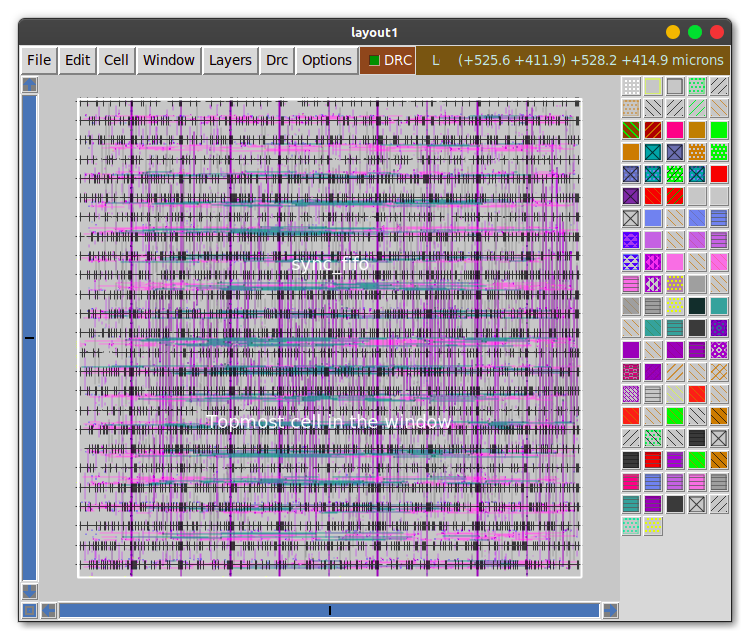
Following is the layout of a full adder circuit designed with qflow tools. You can find RTL of this design at <https://www.edaplayground.com/x/G2qf>.



Following image shows a completed qflow of fifo project in local dir. We have to input a verified design file to qflow and ideally it will generate a clean gds file. You can check out my UVM verified design at <https://www.edaplayground.com/x/9nje> . Although I did change system verilog to verilog file as yosys does not support SV right now.



During this design I have encountered more than 15 drc errors which had to be resolved one by one manually. Here familiarity with magic tool became very useful. We can also see output of this design in magic as seen with following image



In all of these designs osu018 technology was used. Alternatives to this are osu035 and osu050 available at this time.

Here a significant amount of time is spent at the placement stage as it takes a while to achieve set density in the qflow placement settings. We can reduce this density to achieve better run time if we are still not finalizing the layout.

**References:**

[1]"Open Circuit Design", *Opencircuitdesign.com*, 2021. [Online]. Available: http://opencircuitdesign.com/. [Accessed: 12- Jul- 2021]

[2]"Graywolf placement tool", *github.com/rubund/graywolf*, 2021. [Online]. Available: https://github.com/rubund/graywolf. [Accessed: 12- Jul- 2021]

[3]"Qrouter routing tool", *Opencircuitdesign.com*, 2021. [Online]. Available: opencircuitdesign.com/qrouter/. [Accessed: 12- Jul- 2021]

[4]"Yosys", *clifford.at/yosys/*, 2021. [Online]. Available: http://www.clifford.at/yosys/. [Accessed: 12- Jul- 2021]

[5]"Yosys git repo", *github.com/YosysHQ/yosys*, 2021. [Online]. Available: https://github.com/YosysHQ/yosys. [Accessed: 12- Jul- 2021]

[6]"Magic git repo", *github.com/RTimothyEdwards/magic*, 2021. [Online]. Available: https://github.com/RTimothyEdwards/magic. [Accessed: 12- Jul- 2021]

[7]"Magic Tutorial #1: Getting Started", *John Ousterhout*, 2001. [Online]. Available: http://opencircuitdesign.com/magic/. [Accessed: 12- Jul- 2021]

[8]"The magic home pae", *bwrcs.eecs.berkeley.edu*, 2021. [Online]. Available: http://bwrcs.eecs.berkeley.edu/Classes/IcBook/magic/. [Accessed: 12- Jul- 2021]