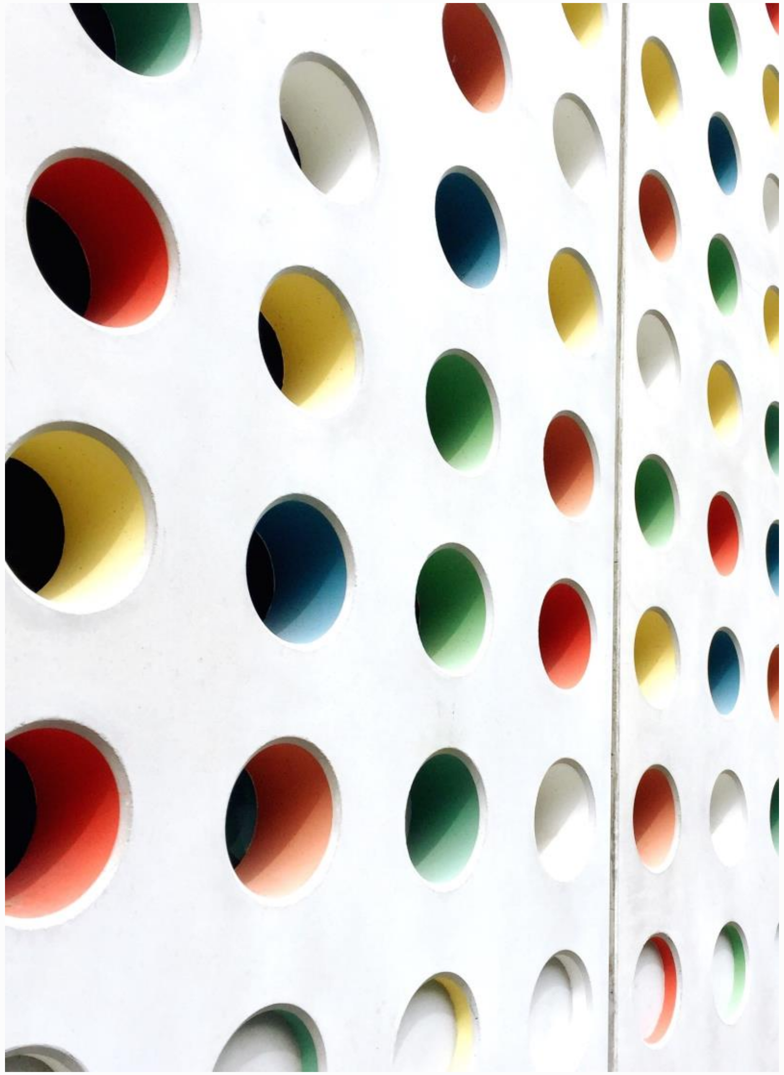


Revolution EDA

MURAT ESKIYERLI, PhD

EMAIL: ESKIYERLI@REVEDA.EU

TEL: +31 6 4200 7540

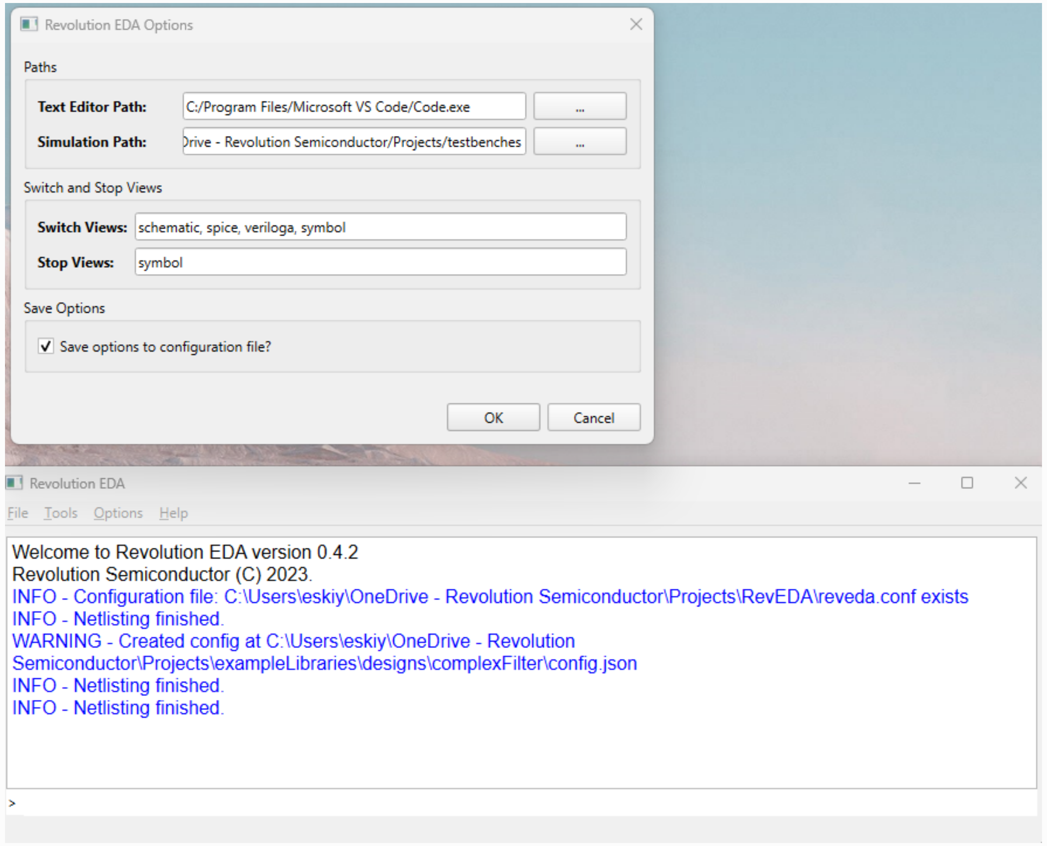
WEBSITE: WWW.REVEDA.EU

Why A Revolution?

- Electronic-design automation is the enabling technology of semiconductor industry.
- Closest analogy is the Software IDEs.
- MS Visual Studio vs Visual Studio Code.
  - Latter's power is in its extendibility.
- Model Revolution EDA is pursuing.

RevEDA Main Window

- Includes a full a Python interpreter.
- Access to configuration options and tools such as *Verilog-A module* import.
- It also highlights errors and warnings.
  - *reveda.log* file has the same info.



Text Editor Path: C:/Program Files/Microsoft VS Code/Code.exe

Simulation Path: Drive - Revolution Semiconductor/Projects/testbenches

Switch Views: schematic, spice, veriloga, symbol

Stop Views: symbol

☑ Save options to configuration file?

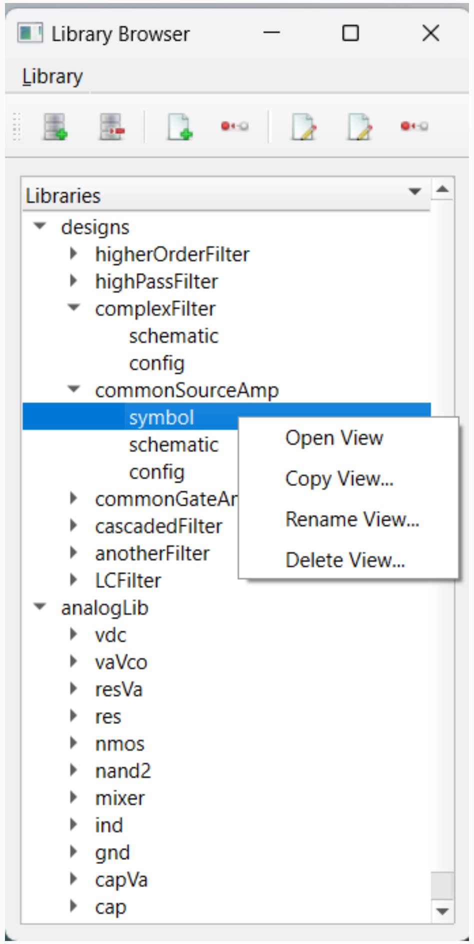
```
Welcome to Revolution EDA version 0.4.2
Revolution Semiconductor (C) 2023.
INFO - Configuration file: C:\Users\eskiy\OneDrive - Revolution Semiconductor\Projects\RevEDA\reveda.conf exists
INFO - Netlisting finished.
WARNING - Created config at C:\Users\eskiy\OneDrive - Revolution
Semiconductor\Projects\exampleLibraries\designs\complexFilter\config.json
INFO - Netlisting finished.
INFO - Netlisting finished.
>
```

Library Browser

- Library Browser should be familiar to most IC designers.

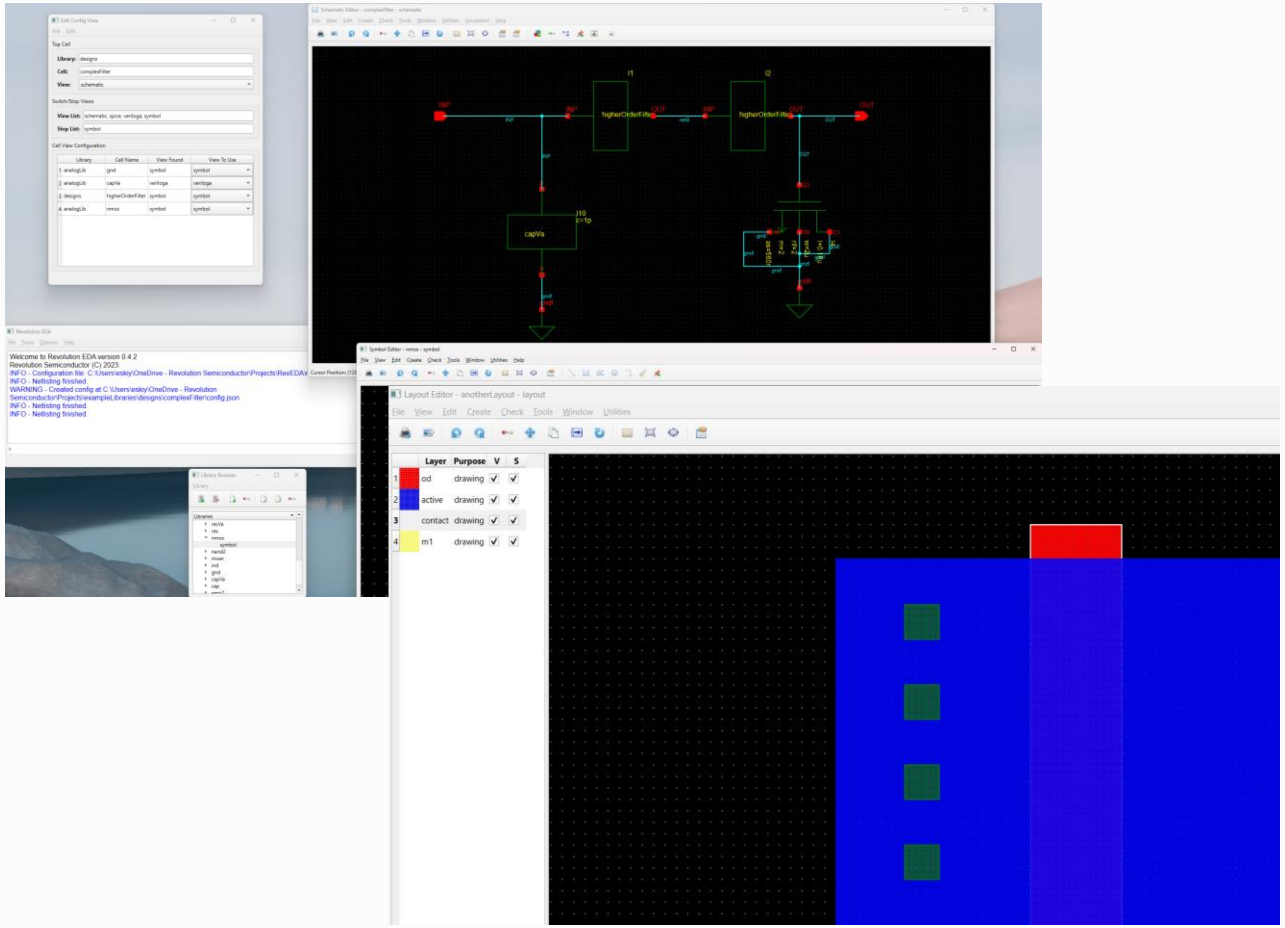
- Library, cell and cell view creation, editing, renaming and deletion done at Library Browser.

- Library paths are also defined in the library path editor.



Design Entry

- Revolution EDA has robust design entry tools:
  - All-around Python based.
  - Built-in Python console.
  - State-of-the-art QT6 toolkit.
  - What is Next?
    - Layout Editor
    - Simulation GUI cockpit.


Edit Config View
File Edit
Schematic Editor - complexFilter - schematic
File View Edit Create Check Tools Window Utilities Simulation Help
I1
I2
higherOrderFilter
higherOrderFilter
capVa
Revolution EDA
File Tools Options Help
Cursor Position:
Symbol Editor - nmos - symbol
File View Edit Create Check Tools Window Utilities Help
Library Browser
Library
Layout Editor - anotherLayout - layout
File View Edit Create Check Tools Window Utilities

Top Cell

Library: designs
Cell: complexFilter
View: schematic

Switch/Stop Views

View List: schematic, spice, veriloga, symbol
Stop List: symbol

Cell View Configuration

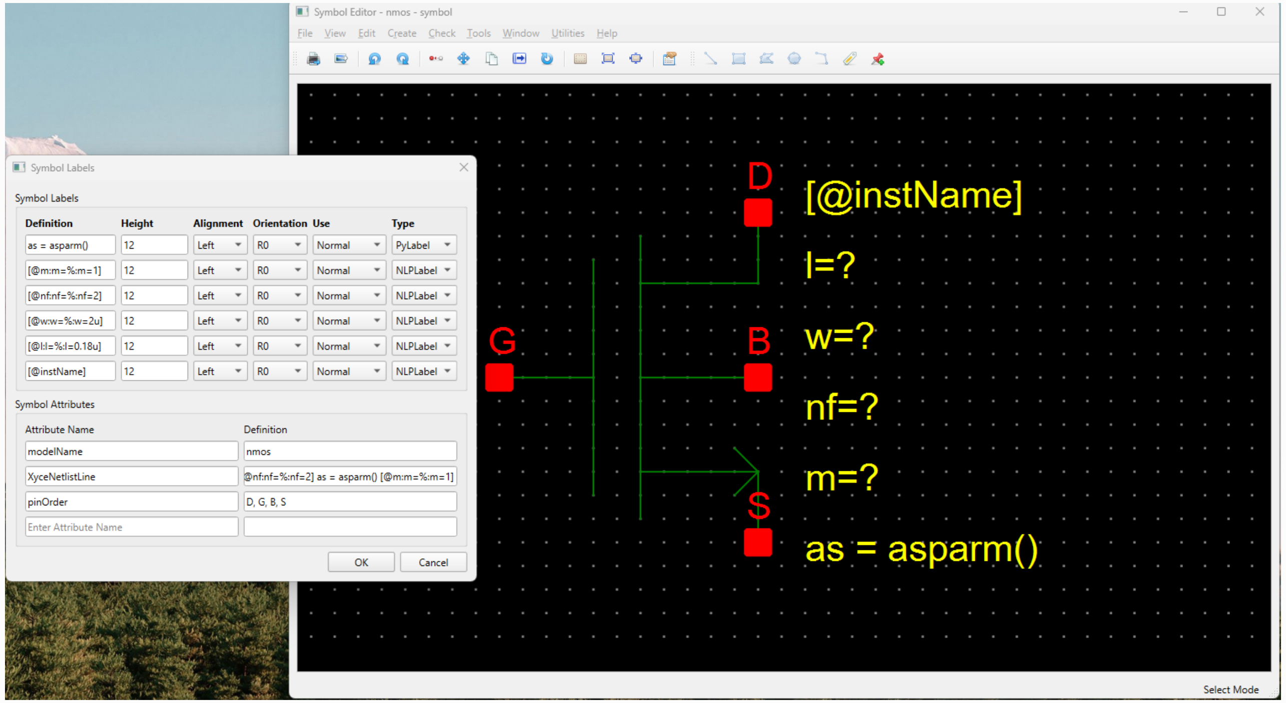
	Library	Cell Name	View Found	View To Use
1	analogLib	gnd	symbol	symbol
2	analogLib	capVa	veriloga	veriloga
3	designs	higherOrderFilter	symbol	symbol
4	analogLib	nmos	symbol	symbol

```
Welcome to Revolution EDA version 0.4.2
Revolution Semiconductor (C) 2023
INFO - Configuration file: C:\Users\eskiy\OneDrive - Revolution Semiconductor\Projects\RevEDA\r
INFO - Netlisting finished.
WARNING - Created config at C:\Users\eskiy\OneDrive - Revolution
Semiconductor\Projects\exampleLibraries\designs\complexFilter\config.json
INFO - Netlisting finished.
INFO - Netlisting finished.
```

		Layer	Purpose	V	S
1	🟥	od	drawing	✓	✓
2	🟦	active	drawing	✓	✓
3		contact	drawing	✓	✓
4	🟨	m1	drawing	✓	✓

Symbol Editor

- RevEDA Symbol Editor can capture:
  - Properties common to all instances (*attributes*)
  - Properties that can change per instance (*labels*)
  - Python Labels (*PyLabel*) allow full use Python functions for instance callbacks. *No more messing with Tcl or SKILL.*


Symbol Editor - nmos - symbol
File View Edit Create Check Tools Window Utilities Help
D
G
B
S
[@instName]
l=?
w=?
nf=?
m=?
as = asparm()
Select Mode

Symbol Labels

Symbol Labels

Definition	Height	Alignment	Orientation	Use	Type
as = asparm()	12	Left	R0	Normal	PyLabel
[@m:m=%:m=1]	12	Left	R0	Normal	NLPLabel
[@nf:nf=%:nf=2]	12	Left	R0	Normal	NLPLabel
[@w:w=%:w=2u]	12	Left	R0	Normal	NLPLabel
[@l:l=%:l=0.18u]	12	Left	R0	Normal	NLPLabel
[@instName]	12	Left	R0	Normal	NLPLabel

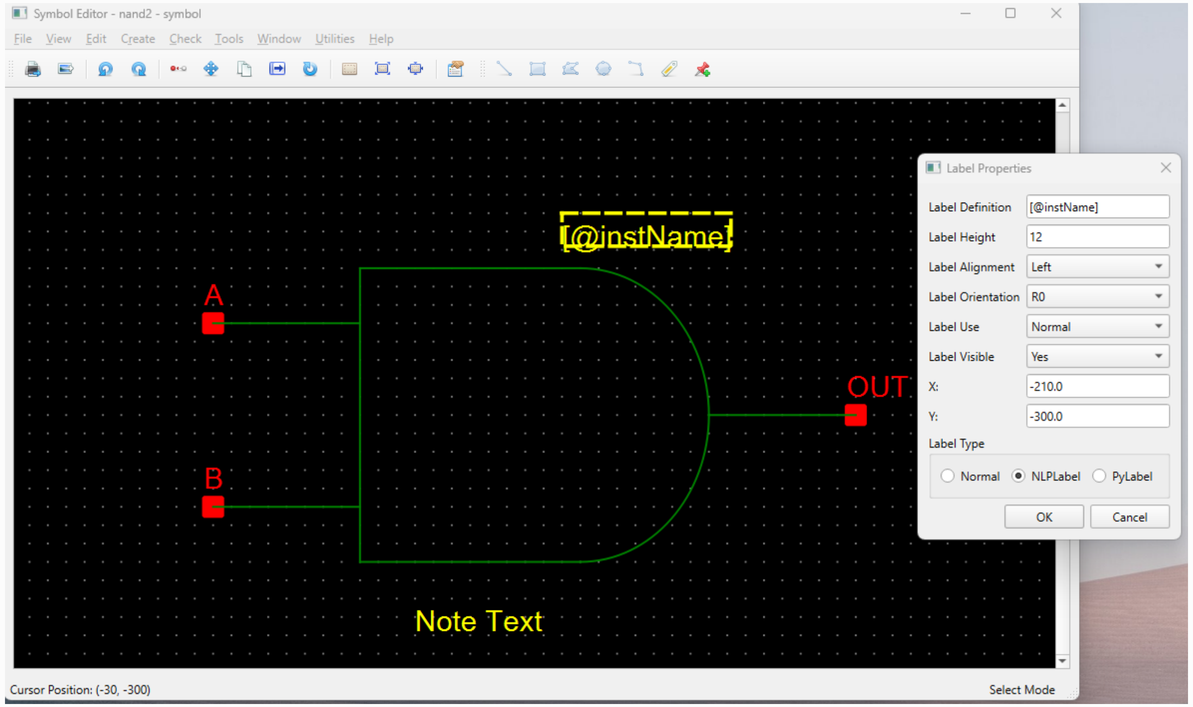
Symbol Attributes

Attribute Name	Definition
modelName	nmos
XyceNetlistLine	@nf:nf=%:nf=2] as = asparm() [@m:m=%:m=1]
pinOrder	D, G, B, S
Enter Attribute Name	

OK Cancel

Symbol Editor

- Lines, Circles, Rectangles and arches can be drawn.
- Pins and labels can be added.
- Labels can be visible or hidden.
- Normal labels can be used at notes on symbol.

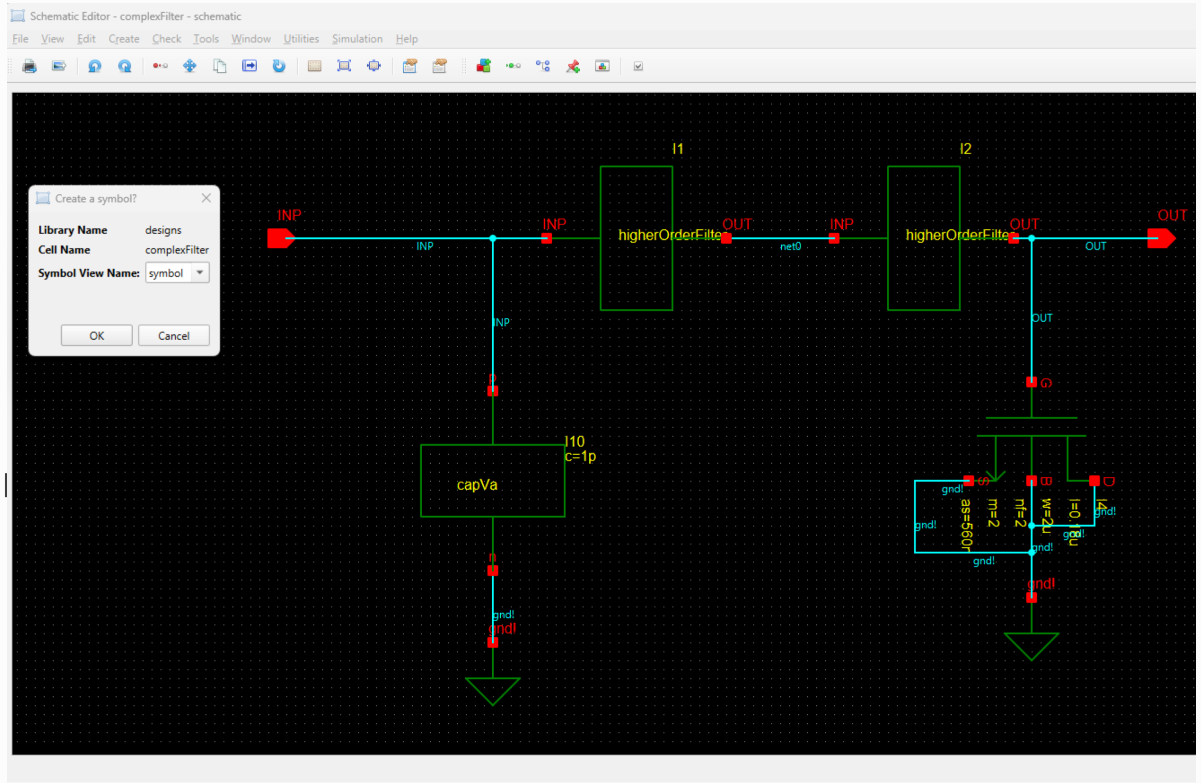

Symbol Editor - nand2 - symbol
File View Edit Create Check Tools Window Utilities Help
[@instName]
A
B
OUT
Note Text
Label Properties
OK
Cancel

Label Definition: [@instName]
Label Height: 12
Label Alignment: Left
Label Orientation: R0
Label Use: Normal
Label Visible: Yes
X: -210.0
Y: -300.0
Label Type: ○ Normal ● NLPLabel ○ PyLabel

Cursor Position: (-30, -300)
Select Mode

Schematic Editor

- Schematic Editor
  - Hierarchical, a symbol can be generated for each schematic.
  - Some instance labels are automatically generated.
  - Other instance labels can be entered pe instance based (*NLPLabels*)
  - *PyLabels* generated using associated callback functions.



Schematic Editor - complexFilter - schematic

Create a symbol?

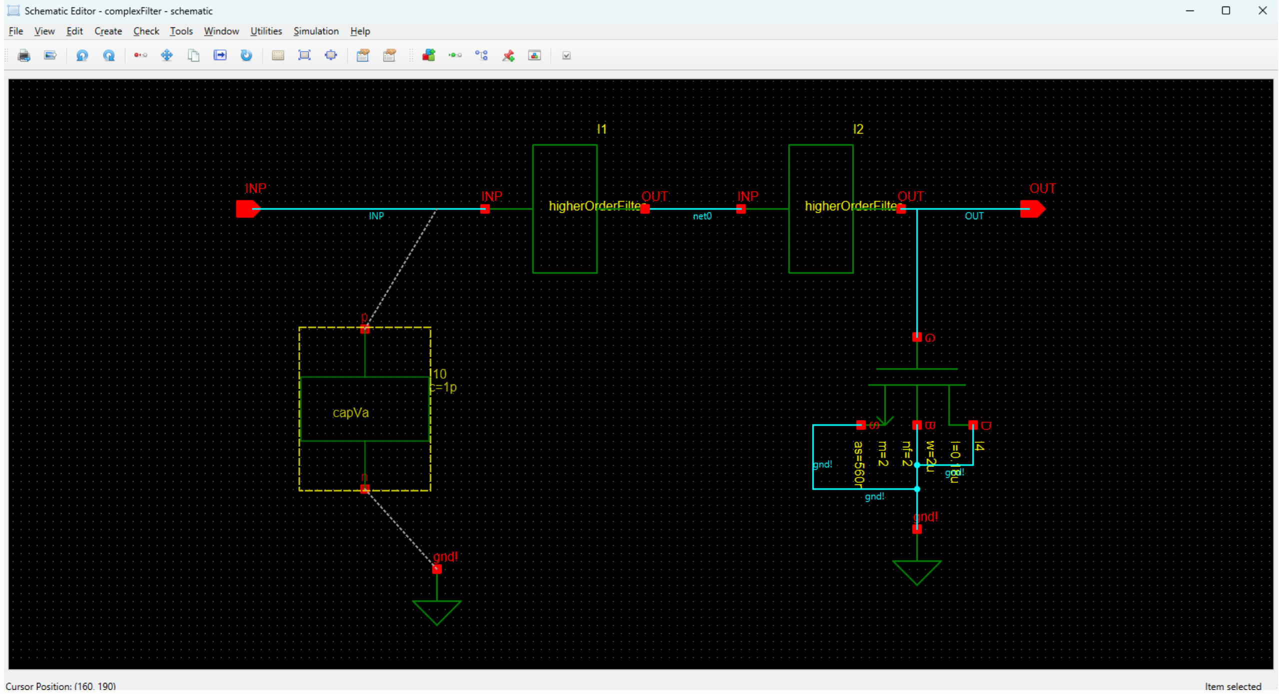
Library Name designs

Cell Name complexFilter

Symbol View Name: symbol

Schematic Editor

- Net editing is constantly improving:
  - Net naming
  - Net merging
  - Snapping to instance pins
  - Snapping to other net end points.
  - Global nets defined using symbols (gnd, vdd, etc).

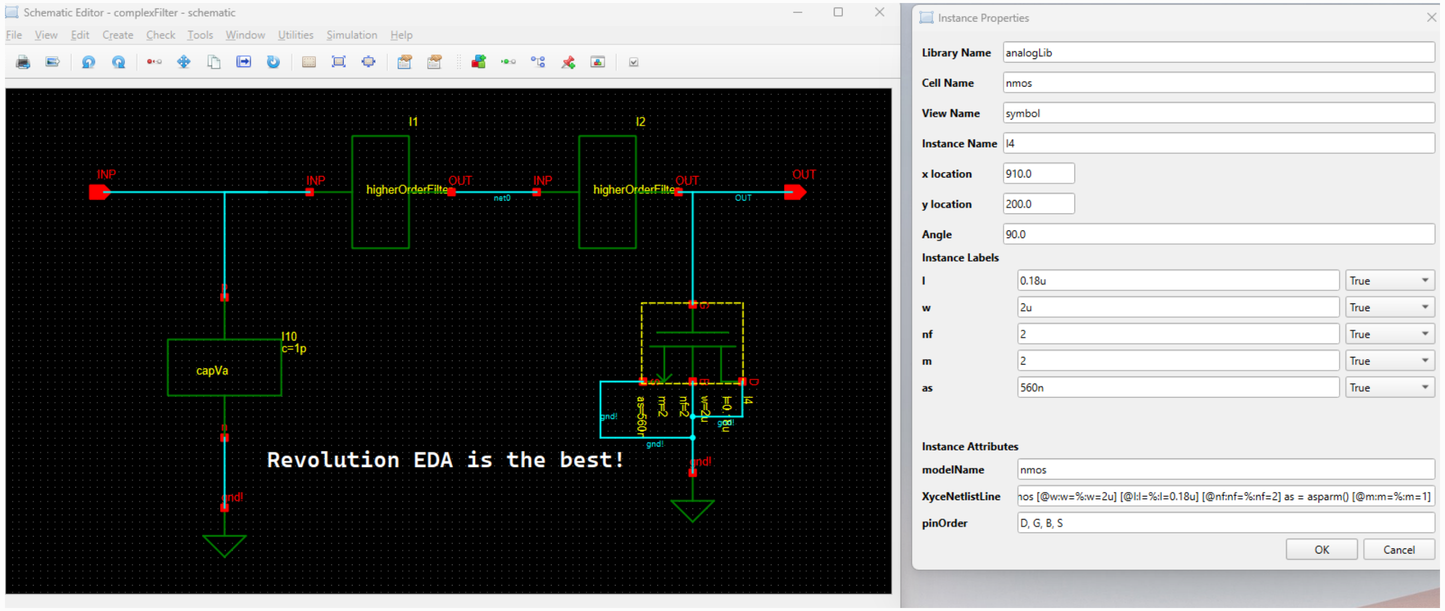


Cursor Position: (160, 190)

Item selected

Schematic Editor

- Instance properties (labels), location, and angle can be changed using dialogues.
- Schematic cells can be traversed both up and down.
- Text notes can be added to schematics.


Schematic Editor - complexFilter - schematic
File View Edit Create Check Tools Window Utilities Simulation Help
I1
I2
INP
INP
higherOrderFilter
OUT
net0
INP
higherOrderFilter
OUT
OUT
OUT
I10
c=1p
capVa
gnd!
gnd!
I4
D
Revolution EDA is the best!
Instance Properties
OK
Cancel

Library Name: analogLib

Cell Name: nmos

View Name: symbol

Instance Name: I4

x location: 910.0

y location: 200.0

Angle: 90.0

Instance Labels

l	0.18u	True
w	2u	True
nf	2	True
m	2	True
as	560n	True

Instance Attributes

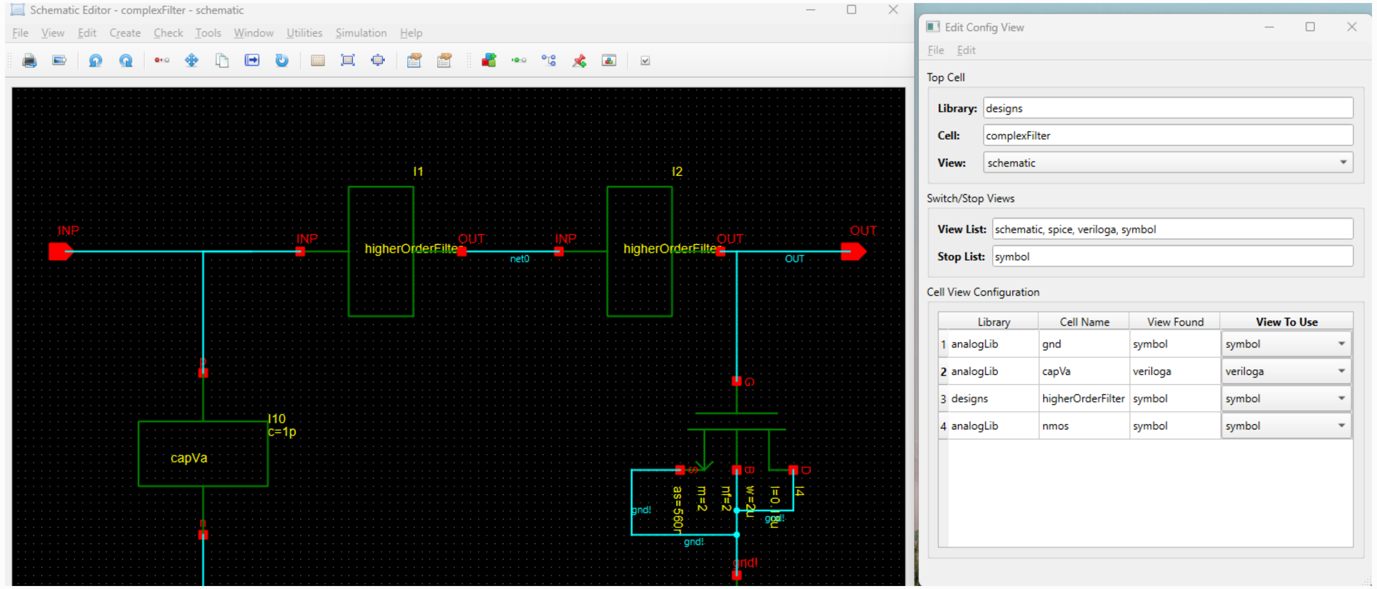
modelName: nmos

XyceNetlistLine: nos [@w:w=%:w=2u] [@l:l=%:l=0.18u] [@nf:nf=%:nf=2] as = asparm() [@m:m=%:m=1]

pinOrder: D, G, B, S

Config Editor

- RevEDA config editor allows per cell based steering of circuit netlisting.
- Familiar interface for experienced users.
- Displayed cells adapt to selected *cellview*.



Top Cell

Library: designs

Cell: complexFilter

View: schematic

Switch/Stop Views

View List: schematic, spice, veriloga, symbol

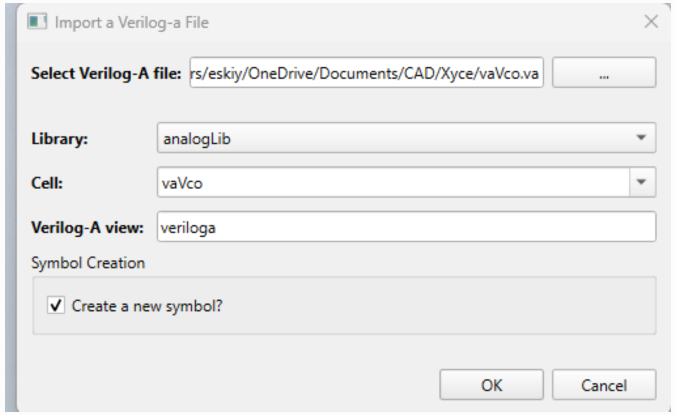
Stop List: symbol

Cell View Configuration

	Library	Cell Name	View Found	View To Use
1	analogLib	gnd	symbol	symbol
2	analogLib	capVa	veriloga	veriloga
3	designs	higherOrderFilter	symbol	symbol
4	analogLib	nmos	symbol	symbol

Verilog-A module Import

- Verilog-A module import is *experimental* and currently only for *Xyce* simulator.
- A symbol can be generated for any module.
  - Including labels and attributes for model and instance parameters.
- ngspice support is planned.



Select Verilog-A file: rs/eskiy/OneDrive/Documents/CAD/Xyce/vaVco.va

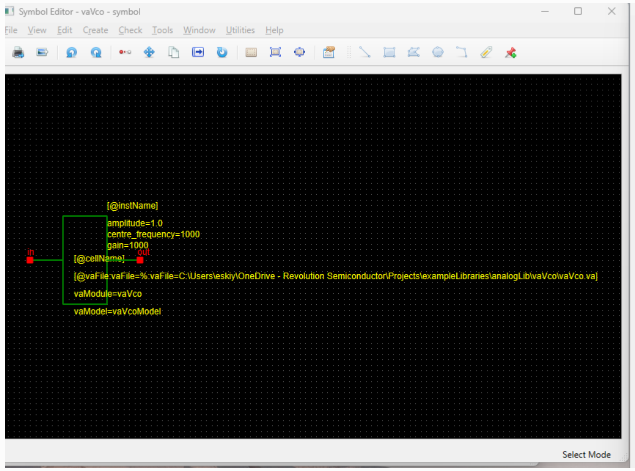
Library: analogLib

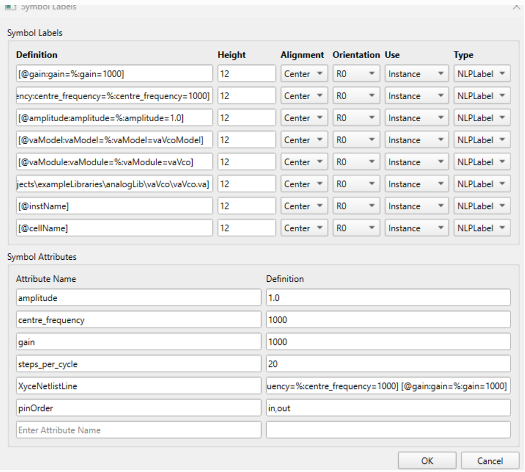
Cell: vaVco

Verilog-A view: veriloga

Symbol Creation

☑ Create a new symbol?





Symbol Labels

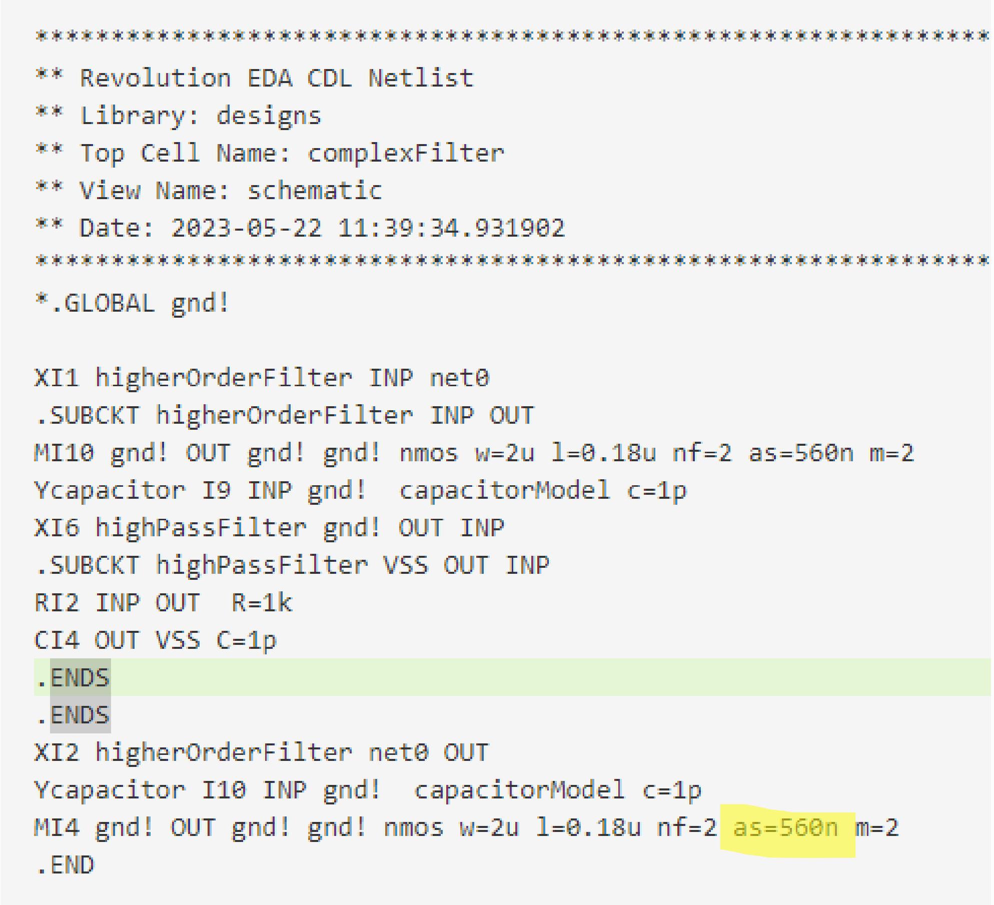
Definition	Height	Alignment	Orientation	Use	Type
[@gain:gain=%:gain=1000]	12	Center	R0	Instance	NLPLabel
ency:centre_frequency=%:centre_frequency=1000]	12	Center	R0	Instance	NLPLabel
[@amplitude:amplitude=%:amplitude=1.0]	12	Center	R0	Instance	NLPLabel
[@vaModel:vaModel=%:vaModel=vaVcoModel]	12	Center	R0	Instance	NLPLabel
[@vaModule:vaModule=%:vaModule=vaVco]	12	Center	R0	Instance	NLPLabel
jects\exampleLibraries\analogLib\vaVco\vaVco.va]	12	Center	R0	Instance	NLPLabel
[@instName]	12	Center	R0	Instance	NLPLabel
[@cellName]	12	Center	R0	Instance	NLPLabel

Symbol Attributes

Attribute Name	Definition
amplitude	1.0
centre_frequency	1000
gain	1000
steps_per_cycle	20
XyceNetlistLine	uency=%:centre_frequency=1000] [@gain:gain=%:gain=1000]
pinOrder	in,out
Enter Attribute Name	

Netlisting

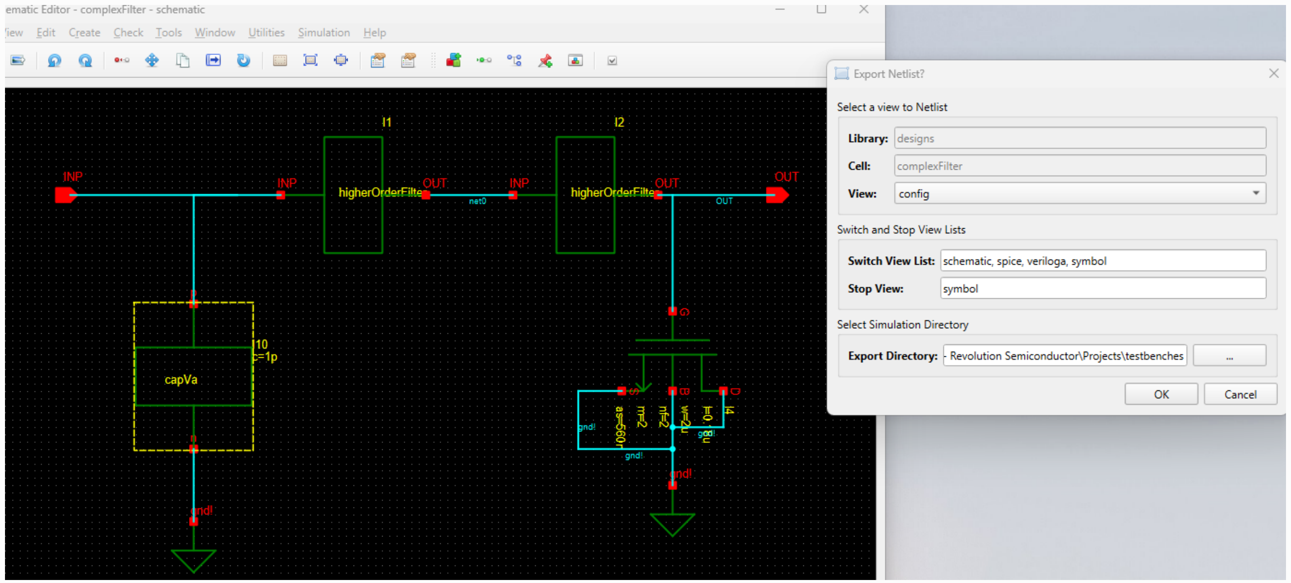
- Hierarchical netlisting is possible:
  - Switch view list based
  - Config view based
- *PyLabel* values will be carried to netlist.
- Full power of Python for all modelling needs.



```
***************************************************************
** Revolution EDA CDL Netlist
** Library: designs
** Top Cell Name: complexFilter
** View Name: schematic
** Date: 2023-05-22 11:39:34.931902
***************************************************************
*.GLOBAL gnd!

XI1 higherOrderFilter INP net0
.SUBCKT higherOrderFilter INP OUT
MI10 gnd! OUT gnd! gnd! nmos w=2u l=0.18u nf=2 as=560n m=2
Ycapacitor I9 INP gnd!  capacitorModel c=1p
XI6 highPassFilter gnd! OUT INP
.SUBCKT highPassFilter VSS OUT INP
RI2 INP OUT  R=1k
CI4 OUT VSS C=1p
.ENDS
.ENDS
XI2 higherOrderFilter net0 OUT
Ycapacitor I10 INP gnd!  capacitorModel c=1p
MI4 gnd! OUT gnd! gnd! nmos w=2u l=0.18u nf=2 as=560n m=2
.END
```

Netlisting

- Still being worked on.
- Hierarchical netlisting steered by switch view list setting or config view.
- Currently targeting Xyce, but ngspice support is planned.



Export Netlist?

Select a view to Netlist

Library: designs

Cell: complexFilter

View: config

Switch and Stop View Lists

Switch View List: schematic, spice, veriloga, symbol

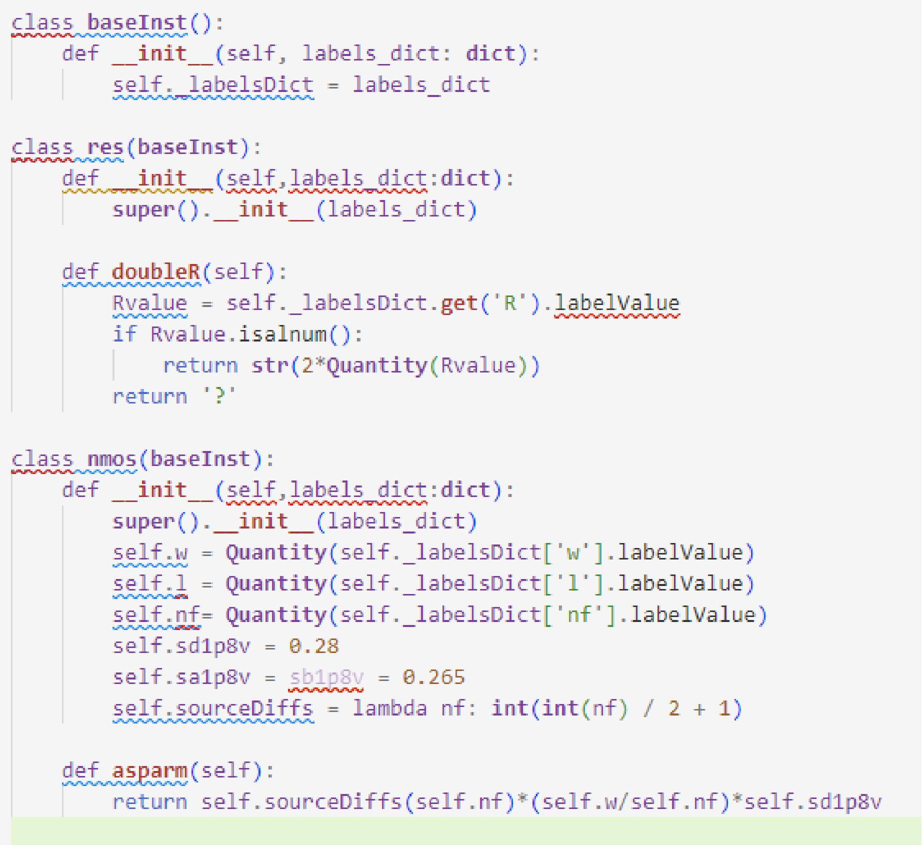
Stop View: symbol

Select Simulation Directory

Export Directory: - Revolution Semiconductor\Projects\testbenches

PDK Support

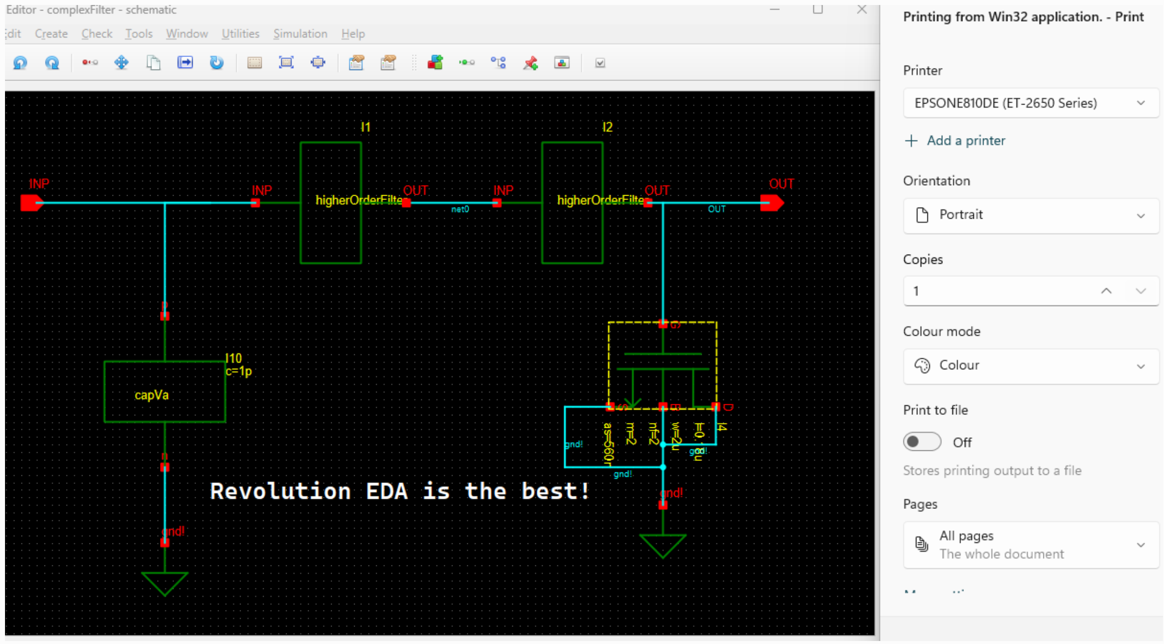
- PDK location can be defined using *REVEDASIM_PATH* environment variable or using .env file.
- PDK directory should be named *pdk* and instance callbacks should be defined in *callbacks.py* file.
- Each cell will have a separate class defined.
- Python label values can depend on other label values.



```python
class baseInst():
    def __init__(self, labels_dict: dict):
        self._labelsDict = labels_dict

class res(baseInst):
    def __init__(self,labels_dict:dict):
        super().__init__(labels_dict)

    def doubleR(self):
        Rvalue = self._labelsDict.get('R').labelValue
        if Rvalue.isalnum():
            return str(2*Quantity(Rvalue))
        return '?'

class nmos(baseInst):
    def __init__(self,labels_dict:dict):
        super().__init__(labels_dict)
        self.w = Quantity(self._labelsDict['w'].labelValue)
        self.l = Quantity(self._labelsDict['l'].labelValue)
        self.nf= Quantity(self._labelsDict['nf'].labelValue)
        self.sd1p8v = 0.28
        self.sa1p8v = sb1p8v = 0.265
        self.sourceDiffs = lambda nf: int(int(nf) / 2 + 1)

    def asparm(self):
        return self.sourceDiffs(self.nf)*(self.w/self.nf)*self.sd1p8v
```

Print and Image Export Support

- ○ Schematics and symbols can be printed using system printers.
- ○ Schematics and symbol drawings can be exported to common graphics formats.



Editor - complexFilter - schematic

Printing from Win32 application. - Print

Printer: EPSONE810DE (ET-2650 Series)

+ Add a printer

Orientation: Portrait

Copies: 1

Colour mode: Colour

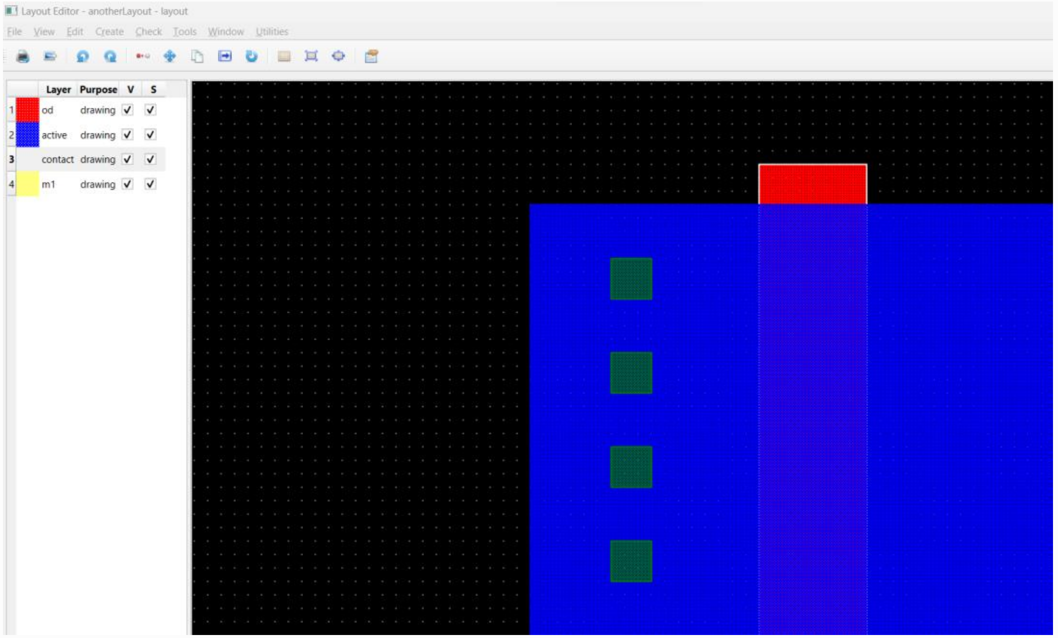
Print to file: Off

Stores printing output to a file

Pages: All pages
The whole document

Layout Editor

- Layout Editor is the new focus.
  - Python-based parametric cells.
  - Gdstk library is used for GDS export.
  - Schematic-driven-Layout flow would be possible when integrated with Schematic Editor.

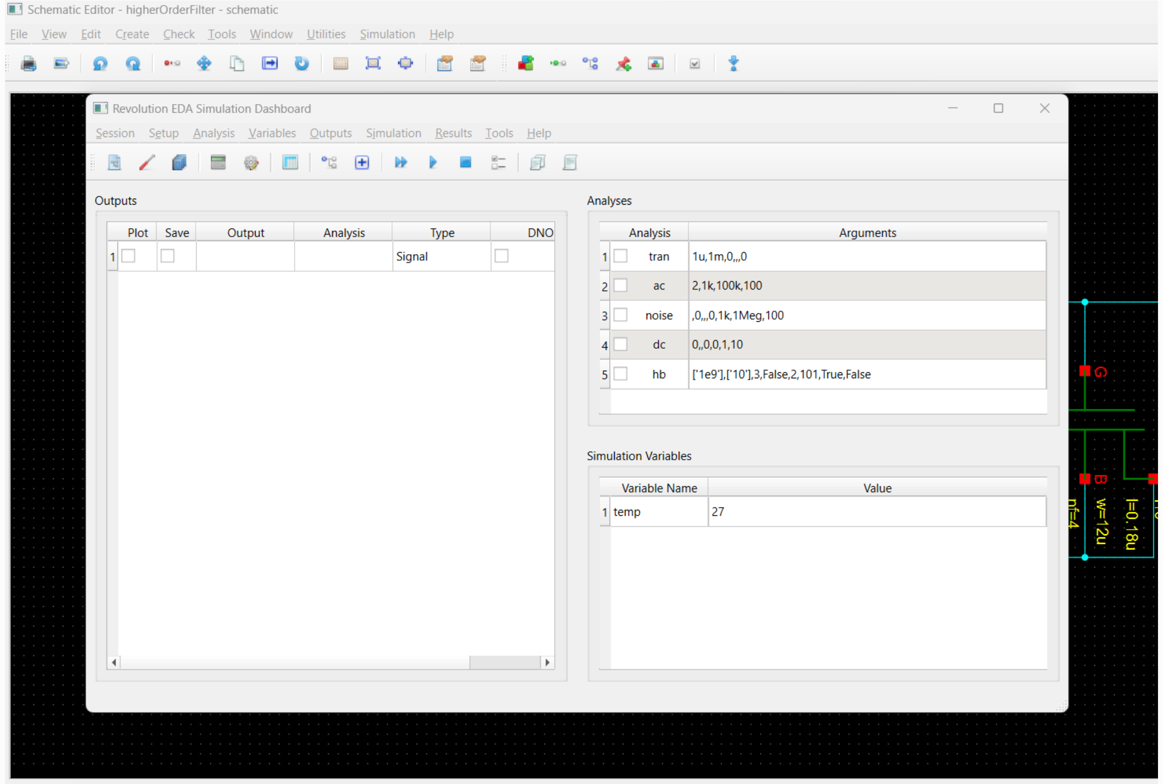


Layout Editor - anotherLayout - layout
File View Edit Create Check Tools Window Utilities

		Layer	Purpose	V	S
1	🟥	od	drawing	☑	☑
2	🟦	active	drawing	☑	☑
3		contact	drawing	☑	☑
4	🟨	m1	drawing	☑	☑

Simulation GUI

- A fully-fledged simulaton GUI targeting Xyce based on Glade was done. Porting to Revolution EDA has started.



Schematic Editor - higherOrderFilter - schematic
File View Edit Create Check Tools Window Utilities Simulation Help
Revolution EDA Simulation Dashboard
Session Setup Analysis Variables Outputs Simulation Results Tools Help
G
B
nf=4
w=12u
l=0.18u

Outputs

	Plot	Save	Output	Analysis	Type	DNO
1	□	□			Signal	□

Analyses

		Analysis	Arguments
1	□	tran	1u,1m,0,,,0
2	□	ac	2,1k,100k,100
3	□	noise	,0,,,0,1k,1Meg,100
4	□	dc	0,,0,0,1,10
5	□	hb	['1e9'],['10'],3,False,2,101,True,False

Simulation Variables

	Variable Name	Value
1	temp	27

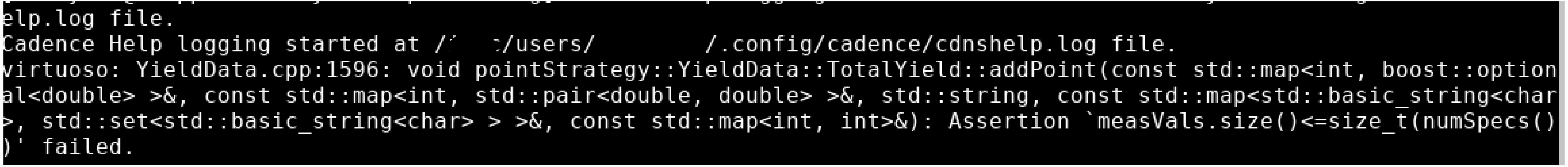
What does RevEDA need?

- A full-fledged PDK
  - Symbols
  - Instance callbacks
  - Parametric cells
  - Device Models

- Testing, testing and more testing.
  - Software unit tests.
  - User tests.
  - Test tape-outs.

One last thing…

- It is buggy, but who isn’t?



```
elp.log file.
Cadence Help logging started at /[redacted]:/users/[redacted]/.config/cadence/cdnshelp.log file.
virtuoso: YieldData.cpp:1596: void pointStrategy::YieldData::TotalYield::addPoint(const std::map<int, boost::option
al<double> >&, const std::map<int, std::pair<double, double> >&, std::string, const std::map<std::basic_string<char
>, std::set<std::basic_string<char> > >&, const std::map<int, int>&): Assertion `measVals.size()<=size_t(numSpecs()
)' failed.
```

- Any ideas for future development, add-ons, etc.?