



LVS of CMOS Inverter and 2-input NAND *continuation*

Aim:

To understand the basics of
KLayout's LVS

Disclaimer:

This document is for personal records only.
There is NO WARRANTY on technical
correctness.



By Kazzz-S (2023-03-01)
original (2023-02-21)

Part-IV: Re-experiments

<https://github.com/KLayout/klayout/issues/1304>

16. Confirmed Issue

Spice netlist reader: should read "M" terminals in DGS order. #1304

 **Open** klayoutmatthias opened this issue 2 days ago · 0 comments



klayoutmatthias commented [2 days ago](#) • edited ▾


Feb 27, 2023, 7:11 AM GMT+9

For DMOS devices where S/D will not swap, it is important to read the terminals in the correct S/D assignment. Right now, the Spice reader reads it S, G, D, B, but it should be D, G, S, B to be compatible with standard Spice.

Please also see the elaborate discussion here:
<https://www.klayout.de/forum/discussion/2238/how-to-extract-transistor-in-parallel-topology-in-lvs#latest>

  klayoutmatthias self-assigned this 2 days ago

  klayoutmatthias added the **bug** label 2 days ago

  klayoutmatthias added this to the **0.28.6** milestone 2 days ago

 klayoutmatthias pushed a

Fixed issue [#1304](#) (terminal order for MOS devices)

 0fa9bc6

Collaborator

Assignees

 klayoutmatthias

Labels

bug

Projects

None yet

Milestone

0.28.6


Development

No branches or pull req

Notifications

Build the "issue-1304" branch for re-experiments

16. Confirmed Issue



About KLayout

KLayout 0.28.5

By Matthias Köfferlein, Munich [2023-02-28 r28883ee38](#)

For feedback and bug reports mail to: contact@klayout.de

Copyright (C) 2006-2023 Matthias Köfferlein

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.


You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Build options:

- Ruby interpreter 3.1.3-p185 (x86_64-linux)
- Python interpreter 3.8.10 (default, Nov 14 2022, 12:59:47) [GCC 9.4.0]
- Qt bindings for scripts

Binary extensions:

[Close](#)



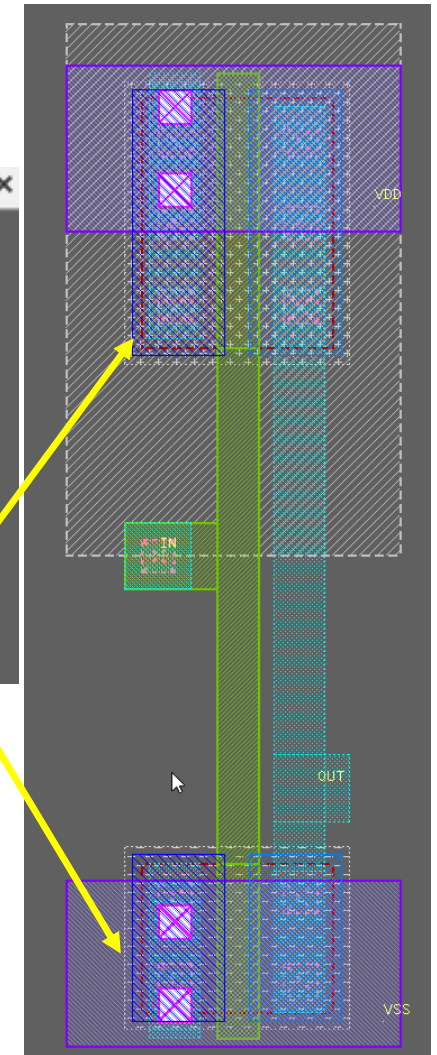
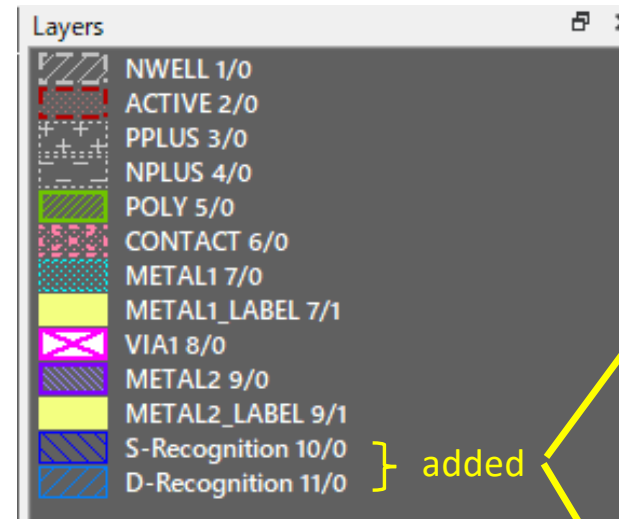
Commit [28883ee3](#) ([open](#))
by [Matthias Koefferlein](#), 2023-02-28 08:22
parent [4d2168a1](#)

Updated golden test data

branches issue-1304, upstream/issue-1304

17. Using DMOS4 for the **inv.cir** Tutorial

1. Add two dummy layers to recognize S and D easily. They are logical instances.



LVS-CMOS-Inverter-NAND.zip
11.1M

LVS-CMOS-Inverter-NAND.pdf
5.2M

Inv-DMOS4.zip
398.9K

Refer to this ZIP

NAND-DMOS4.zip
1.6M

https://www.klayout.de/forum/discussion/comment/9500/#Comment_9500

17. Using DMOS4 for the **inv.cir** Tutorial

2. Prepare four SPICE deck files to test all S and D combinations.

The image displays four SPICE deck files for a CMOS inverter, each with a different configuration of PMOS and NMOS connections. The files are named `inv.cir`, `invX1.cir`, `invX2.cir`, and `inv-DMOS4.cir`.

inv.cir (Choice=1): Original: (PMOS, NMOS)=(incorrect, correct). The PMOS is connected to VDD and the NMOS to VSS.

invX1.cir (Choice=2): Variant X1: (PMOS, NMOS)=(incorrect, incorrect). The PMOS is connected to VDD and the NMOS to VDD. An arrow points from this file to the next one, labeled "D and S are swapped".

invX2.cir (Choice=3): Variant X2: (PMOS, NMOS)=(correct, incorrect). The PMOS is connected to VSS and the NMOS to VSS.

inv-DMOS4.cir (Choice=4): Modified: inv-DMOS4.cir (PMOS, NMOS)=(correct, correct). The PMOS is connected to VDD and the NMOS to VSS. A note says "I expect this should match!".

17. Using DMOS4 for the **inv.cir** Tutorial

3. Modify the sample LVS script for testing and debugging.

```
136 #-----
137 # [5] Device extraction
138 #-----
139 # PMOS transistor device extraction
140 extract_devices(dmos4("PMOS"), { "S"
141                                     "ts"
142                                     }
143 # NMOS transistor device extraction
144 extract_devices(dmos4("NMOS"), { "S"
145                                     "ts"
146                                     }
147 #-----
148 # [6] Define connectivity for netlist
149 #-----
150 # Inter-layer
151 connect(pss, contact)
152 connect(pdd, contact)
153 connect(nss, contact)
154 connect(ndd, contact)
155 connect(poly, contact)
156 connect(contact, metall)
157 connect(metall, metall_lbl) # a
158 connect(metall, vial)
159 connect(metall, metall2)
160 connect(vial, metall2)
161 connect(metall2, metall2_lbl) # a
162 # Global
163 connect_global(bulk, "SUBSTRATE")
164 connect_global(nwell, "NWELL")
165 #-----
166 # [7] Compare section
167 #-----
168 schematic(spicedeck)
169 #-----
170 compare
171 # EOF
172
```

```
91 #-----
92 # [3] Drawing layers
93 #-----
94 #-----
95 # [3] Drawing layers
96 #-----
97 nwell = input(1, 0)
98 active = input(2, 0)
99 pplus = input(3, 0)
100 nplus = input(4, 0)
101 poly = input(5, 0)
102 contact = input(6, 0)
103 metall = input(7, 0)
104 metall_lbl = labels(7, 1)
105 vial = input(8, 0)
106 metall2 = input(9, 0)
107 metall2_lbl = labels(9, 1)
108 source_rec = input(10, 0) #
109 drain_rec = input(11, 0) #
110 # Bulk layer for terminal pro
111 bulk = polygon_layer
112 #-----
113 # [4] Computed layers
114 #-----
115 #-----
116 # In a future "inv-DMOS4B
117 # to use the S- and D-rec
118 #-----
119 active_in_nwell = active
120 pactive = active
121 pgate = pactive
122 psd = pactive
123 pss = psd &
124 pdd = psd &
125 #-----
126 active_outside_nwell = active
127 nactive = active
128 ngate = nactive
129 nsd = nactive & ngate
130 nss = nsd & source_rec
131 ndd = nsd & drain_rec
132
```

```
46 puts "Bye! You can ignore the
47 exit
48 when 1, 2, 3, 4
49 @spicedeck = [ $choice, @spicedeck
50 else
51 puts "### Usage [ 0=>exit, 1=>
52 puts " Set one of above (
53 puts " then, execute the
54 return
55 end
56 end
57 end
58
59 #-----
60 # [1] Design
61 #-----
62 dirName, baseName = File.split( RBA::
63 Dir.chdir(dirName)
64
65 source( "inv-DMOS4.oas", "INVERTER" )
66
67 deep
68
69 selector = SpiceDeckSelector.new
70 selector.GetOneOption()
71 choice, spicedeck = selector.GetSpice
72 if choice == 10 # 1st run
73 return
74 elsif not [1, 2, 3, 4].include?(choice)
75 puts "! Your choice '#{choice}' is i
76 return
77 end
78 AboutCurrentDesign(spicedeck)
79
```

inv-DMOS4A.lvs

```
1 #-----
2 # Original: ${klayout_root}/testdata/lvs/inv.lvs
3 #
4 # Modified: inv-DMOS4A.lvs
5 # by: Kazzz-S
6 # date: 2023-02-21
7 # aims: 1) use the modified "inv-DMOS4.cir" SPICE deck
8 # 2) use "DMOS4" instead of "MOS4"
9 # ==> https://www.klayout.de/forum/discussion/comment/9484/#Comment_9484
10 # 3) use "helper layers" for source- and drain-recognition
11 #-----
12 #-----
13 # [0] Utilities
14 #-----
15 #-----
16 def AboutCurrentDesign(spicedeck)
17 designFile = RBA::CellView::active.filename
18 active_layout = RBA::CellView::active.layout
19 puts ">>> Current design file = '#{designFile}'"
20 puts " Top cell name = '#{active_layout.top_cell.name}'"
21 puts " SPICE deck file = '#{spicedeck}'"
22 end
23
24 class SpiceDeckSelector
25 def initialize
26 if $choice == nil # global variable to choose a SPICE deck
27 $choice = 10
28 end
29 @spicedeck = [ $choice, "" ]
30
31 @spiceOpt = { 0 => "exit", 1 => "original", 2 => "variantX1", 3 => "variantX2", 4 => "correct" }
32 @spiceDECK = { "original" => "inv.cir", # (PMOS, NMOS)=(incorrect, correct)
33 "variantX1" => "invX1.cir", # (PMOS, NMOS)=(incorrect, incorrect)
34 "variantX2" => "invX2.cir", # (PMOS, NMOS)=( correct, incorrect)
35 "correct" => "inv-DMOS4.cir", # (PMOS, NMOS)=( correct, correct)
36 }
37 end
38
39 def GetSpiceDeck
40 return @spicedeck
41 end
42
43 def GetOneOption
44 case $choice
45 when 0
```


17. Using DMOS4 for the [inv.cir](#) Tutorial

4. Run the modified LVS script four times: #1/4

```
inv.cir x invX1.cir x invX2.cir x inv-DMOS4.cir x
1
2 * Simple CMOS inverter circuit : original: (PMOS, NMOS)=(incorrect, correct)
3
4 .SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
5 Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
6 Mn OUT IN VSS SUBSTRATE NMOS W=0.9U L=0.25U
7 .ENDS
8
```

\$choice=1

Netlist Schematic Cross Reference Log

Circuits	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
	▶ Pins		
	▶ Nets		
	▼ Devices		
	▼ NMOS		
	▶ S	\$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	▶ D	VSS (1)	VSS (2)
	▶ G	OUT (2)	OUT (3)
	▶ B	IN (2)	IN (3)
	▶ SUBSTRATE (1)	SUBSTRATE (2)	
	▼ PMOS	\$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]	
	▶ S	VDD (1)	OUT (3)
	▶ D	OUT (2)	VDD (2)
	▶ G	IN (2)	IN (3)
	▶ B	NWELL (1)	NWELL (2)

Configure Probe Net Lock

```
> $choice=1
1
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/Inv-DMOS4/inv-DMOS4.oas'
      Top cell name = 'INVERTER'
      SPICE deck file = 'inv.cir'
```


17. Using DMOS4 for the [inv.cir](#) Tutorial

4. Run the modified LVS script four times: #2/4

```
inv.cir x invX1.cir x invX2.cir x inv-DMOS4.cir x
1
2 * Simple CMOS inverter circuit: Variant X1: (PMOS, NMOS)=(incorrect, incorrect)
3
4 .SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
5 Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
6 Mn VSS IN OUT SUBSTRATE NMOS W=0.9U L=0.25U
7 .ENDS
8
9 Unlike the previous experiment, this net did not match.
```

Netlist Schematic Cross Reference Log

Circuits	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
	▶ Pins		
	▶ Nets		
	▼ Devices		
	▼ NMOS		
	▶ S ↔ D	⊖ \$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	▶ D ↔ S	⊖ VSS (1)	VSS (2)
	▶ G	OUT (2)	OUT (3)
	▶ B	⊖ IN (2)	IN (3)
	▼ PMOS	SUBSTRATE (1)	SUBSTRATE (2)
	▶ S	\$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]	
	▶ D	⊖ VDD (1)	OUT (3)
	▶ G	⊖ OUT (2)	VDD (2)
	▶ B	⊖ IN (2)	IN (3)
		NWELL (1)	NWELL (2)

Configure Probe Net Lock

```
> $choice=2
2
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/Inv-DMOS4/inv-DMOS4.oas'
      Top cell name = 'INVERTER'
      SPICE deck file = 'invX1.cir'
```

17. Using DMOS4 for the [inv.cir](#) Tutorial

4. Run the modified LVS script four times: #3/4

```
inv.cir x invX1.cir x invX2.cir x inv-DMOS4.cir x
1
2 * Simple CMOS inverter circuit: Variant X2: (PMOS, NMOS)=( correct, incorrect)
3
4 .SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
5 Mp OUT IN VSS NWELL PMOS W=1.5U L=0.25U
6 Mn VSS IN OUT SUBSTRATE NMOS W=0.9U L=0.25U
7 .ENDS
8
9
```

\$choice=3

Netlist Schematic Cross Reference Log

Circuits

- INVERTER

Objects

- INVERTER
 - Pins
 - Nets
 - Devices
 - NMOS
 - D ↔ S
 - S ↔ D
 - G
 - B
 - PMOS
 - S
 - D
 - G
 - B

Layout

- INVERTER
 - \$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]
 - OUT (2)
 - VSS (1)
 - IN (2)
 - SUBSTRATE (1)
 - \$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]
 - VDD (1)
 - OUT (2)
 - IN (2)
 - NWELL (1)

Reference

- INVERTER
 - OUT (3)
 - VSS (3)
 - IN (3)
 - SUBSTRATE (2)
 - VSS (3)
 - OUT (3)
 - IN (3)
 - NWELL (2)

Configure Probe Net Lock

```
> $choice=3
3
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/Inv-DMOS4/inv-DMOS4.oas'
      Top cell name = 'INVERTER'
      SPICE deck file = 'invX2.cir'
```

17. Using DMOS4 for the [inv.cir](#) Tutorial

4. Run the modified LVS script four times: #4/4

Netlist Schematic Cross Reference Log

Circuits

INVERTER

Objects

INVERTER

Layout

INVERTER

► Pins

► Nets

▼ Devices

▼ NMOS

► S

► D

► G

► B

▼ PMOS

► S

► D

► G

► B

\$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]

VSS (1) VSS (2)

OUT (2) OUT (3)

IN (2) IN (3)

SUBSTRATE (1) SUBSTRATE (2)

\$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]

VDD (1) VDD (2)

OUT (2) OUT (3)

IN (2) IN (3)

NWELL (1) NWELL (2)

Configure Probe Net Lock

> \$choice=4

4

>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/Inv-DMOS4/inv-DMOS4.oas'

Top cell name = 'INVERTER'

SPICE deck file = 'inv-DMOS4.cir'

```
1 |
2 * Simple CMOS inverter circuit
3 *
4 * Original: ${klayout_root}/testdata/lvs/inv.cir
5 *
6 * Modified: inv-DMOS4.cir (PMOS, NMOS)=( correct, correct) $choice=4
7 * by: Kazzzz-S
8 * date: 2023-02-16
9 * aims: 1) correct D-S connections of the PMOS (flipped in the original)
10
11 .SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
12 * Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
13 Mp OUT IN VDD NWELL PMOS W=1.5U L=0.25U
14 Mn OUT IN VSS SUBSTRATE NMOS W=0.9U L=0.25U
15 .ENDS
16
```

This net matched as expected!

18. Using DMOS4 for a **2-input NAND**




WENSHIH

February 17


<https://www.klayout.de/forum/discussion/comment/ 9493>
on 2023-02-17


Thanks for your suggestion and helps! If there is any information I can supply, please tell me.
Here is the test file.


 nand.test.zip
14.7K


← Starting point. Thank you for providing this!

1. Check the provided resource files and modify them if necessary.

 LVS-CMOS-Inverter-NAND.zip
11.1M

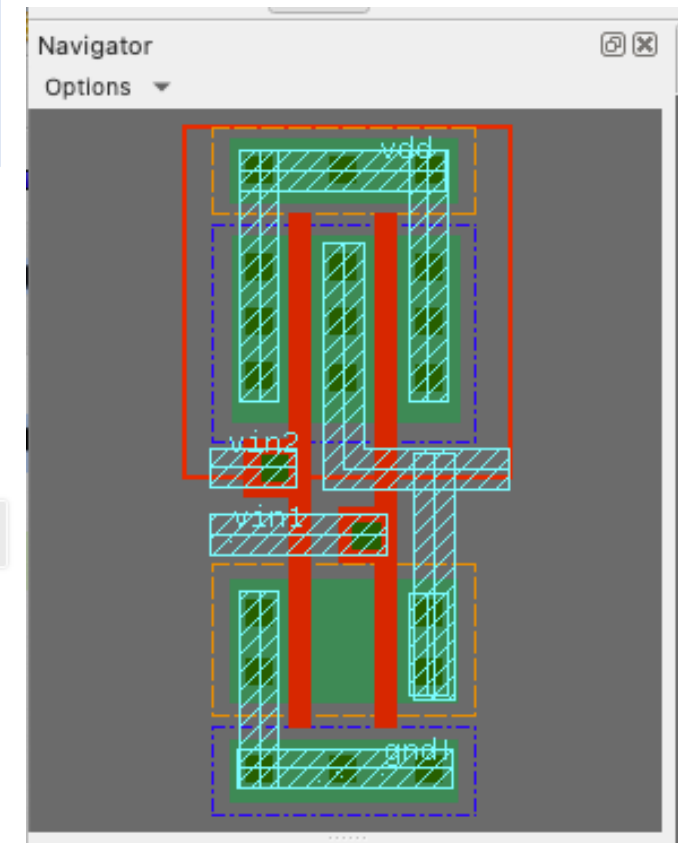
 LVS-CMOS-Inverter-NAND.pdf
5.2M

 Inv-DMOS4.zip
398.9K

 NAND-DMOS4.zip
1.6M

Refer to this ZIP

<https://www.klayout.de/forum/discussion/comment/9500/#Comment 9500>



18. Using DMOS4 for a 2-input NAND

Cells

- ▼ nand
 - ▶ M1_M5
 - ▶ M1_M5\$1
 - ▶ M1_M6
 - ▶ M1_M6\$1
 - M1_V1_M2
 - M1_V1_M2\$1
 - M5_V5_M6
 - M5_V5_M6\$1
 - NCELL\$333500204
 - NCELL\$333500204\$1
 - NCELL\$333501228
 - NCELL\$333501228\$1
 - NCELL\$333504300
 - NCELL\$333504300\$1
 - PCELL\$333495084
 - PCELL\$333495084\$1
 - PCELL\$333496108
 - PCELL\$333496108\$1
 - PCELL\$333497132
 - PCELL\$333497132\$1
 - PCELL\$333498156
 - PCELL\$333498156\$1
 - PO_CO_M1
 - PO_CO_M1\$1
 - nimp_vdd_noCO
 - nimp_vdd_noCO\$1
 - npimp
 - npimp\$1
 - pimp_gnd_noCO
 - pimp_gnd_noCO\$1

Too many ghost cells

Cells

- ▼ nand
 - ▶ -M1_M5
 - ▶ -M1_M5\$1
 - ▶ -M1_M6
 - ▶ -M1_M6\$1
 - M1_V1_M2
 - M1_V1_M2\$1
 - M5_V5_M6
 - M5_V5_M6\$1
 - NCELL\$333500204
 - NCELL\$333500204\$1
 - NCELL\$333501228
 - NCELL\$333501228\$1
 - NCELL\$333504300
 - NCELL\$333504300\$1
 - PCELL\$333495084
 - PCELL\$333495084\$1
 - PCELL\$333496108
 - PCELL\$333496108\$1
 - PCELL\$333497132
 - PCELL\$333497132\$1
 - PCELL\$333498156
 - PCELL\$333498156\$1
 - PO_CO_M1
 - PO_CO_M1\$1
 - nimp_vdd_noCO
 - nimp_vdd_noCO\$1
 - npimp
 - npimp\$1
 - pimp_gnd_noCO
 - pimp_gnd_noCO\$1

I need only the top cell.

Navigator

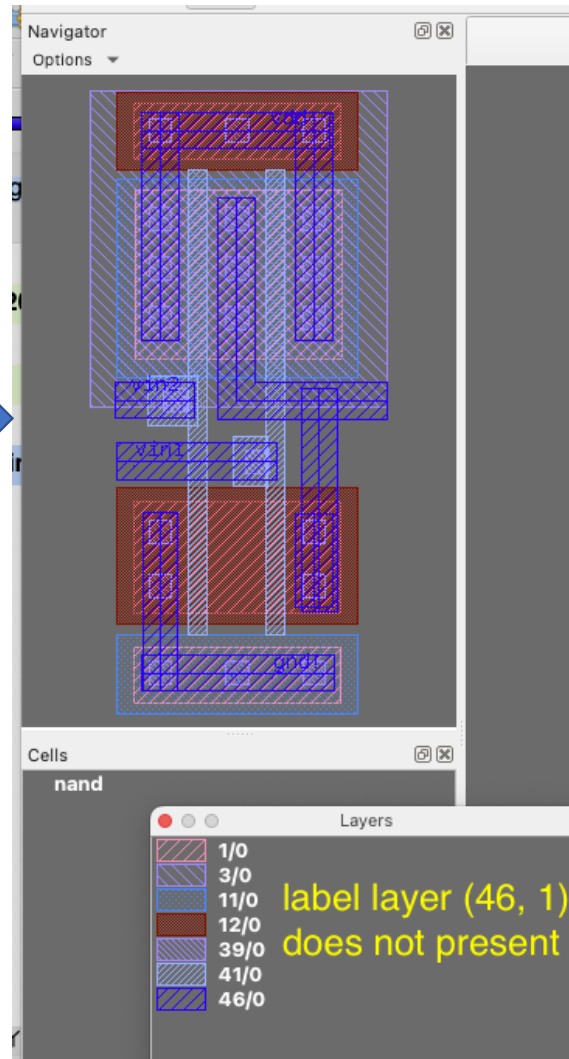
Options

Cells

nand

18. Using DMOS4 for a 2-input NAND

```
11 # Drawing layers:
12
13 poly = input(41, 0)
14 active = input(1, 0)
15 nwell = input(3, 0)
16 gate = poly & active
17 nplus = input(12, 0)
18 pplus = input(11, 0)
19 #nplus_od = nplus & active
20 #pplus_od = pplus & active
21 #nplus_od_nw = nplus & active & nwell
22 #pplus_od_nw = pplus & active & nwell
23 contact = input(39, 0)
24 metal1 = input(46, 0)
25 metal1_lbl = labels(46, 1)
26 via1 = input(47, 0)
27 metal2 = input(48, 0)
28 metal2_lbl = labels(48, 1)
29 via2 = input(49, 0)
30 metal3 = input(50, 0)
31 metal3_lbl = labels(50, 1)
32 # Bulk layer for terminal provisioning
33
```



18. Using DMOS4 for a 2-input NAND

2. Prepare two SPICE deck files.

```
nand.cir
1 .SUBCKT nand
2 M_M6 GND VIN2 N15943 GND MbreakN L=0.18um W=1.08um
3 M_M3 N15943 VIN1 VOUT GND MbreakN L=0.18um W=1.08um
4 M_M5 VDD VIN1 VOUT VDD MbreakP L=0.18um W=1.62um
5 M_M4 VDD VIN2 VOUT VDD MbreakP L=0.18um W=1.62um
6 .ENDS
7
```

terminals are missing

original SPICE deck

```
1 * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2 *
3 * Modified the original SPICE net
4 * 1) there is no tie-down diodes in the original layout
5 * 2) added terminals [VDD GND VIN1 VIN2 VOUT NWELL SUBSTRATE]
6 * 3) transistor name changed to the default (MbreakN, MbreakP) ==> (NMOS, PMOS)
7 * 4) Still INCORRECT! ==> Source-Drain interchanged
8
9 .SUBCKT nand VDD GND VIN1 VIN2 VOUT NWELL SUBSTRATE
10 M_M6 GND VIN2 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
11 M_M3 N15943 VIN1 VOUT SUBSTRATE NMOS L=0.18um W=1.08um
12 M_M5 VDD VIN1 VOUT NWELL PMOS L=0.18um W=1.62um
13 M_M4 VDD VIN2 VOUT NWELL PMOS L=0.18um W=1.62um
14 .ENDS
```

\$choice=1 nand-incorrect.cir

D and S are swapped

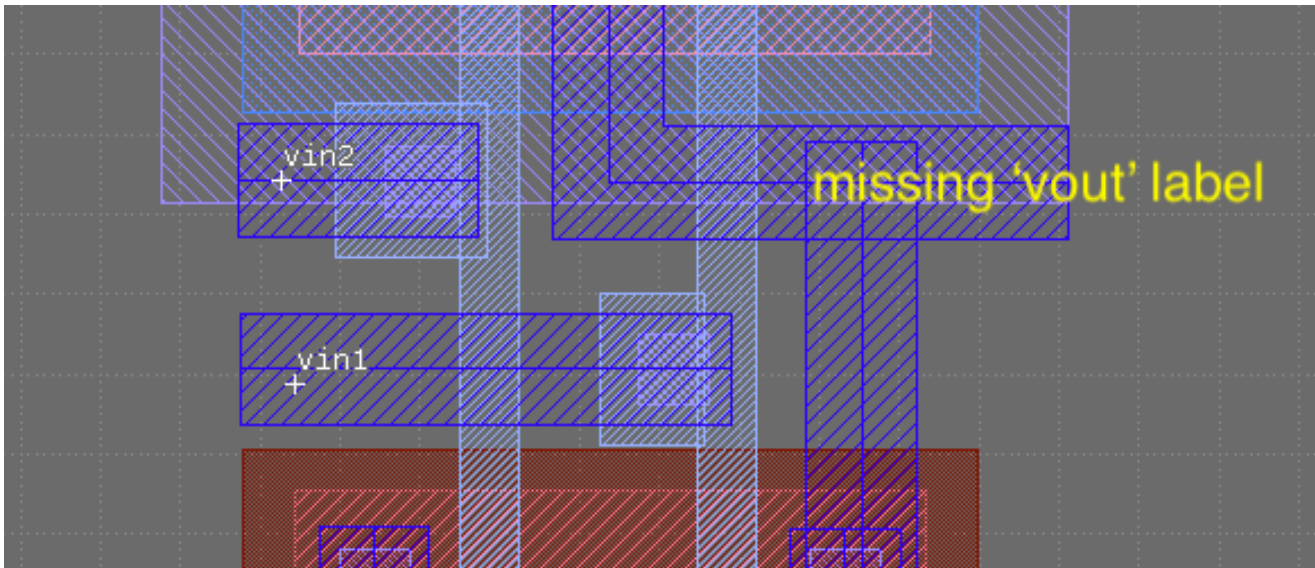
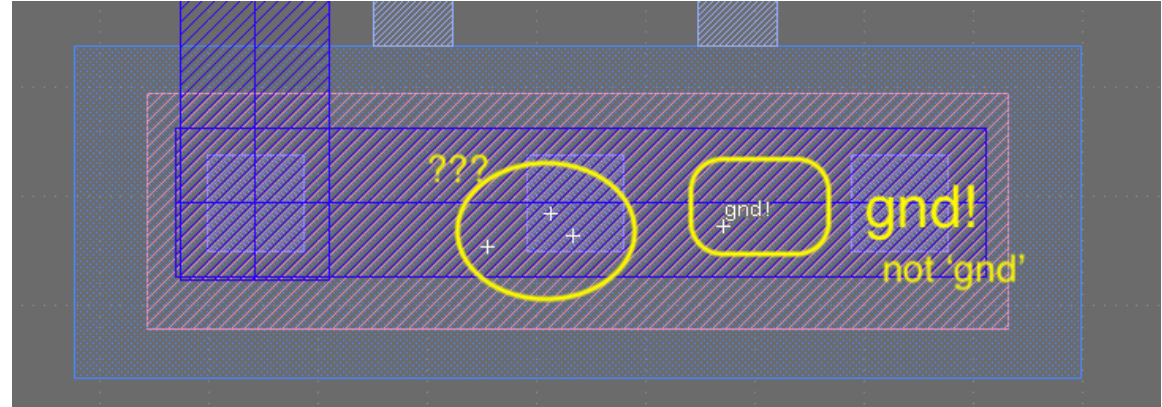
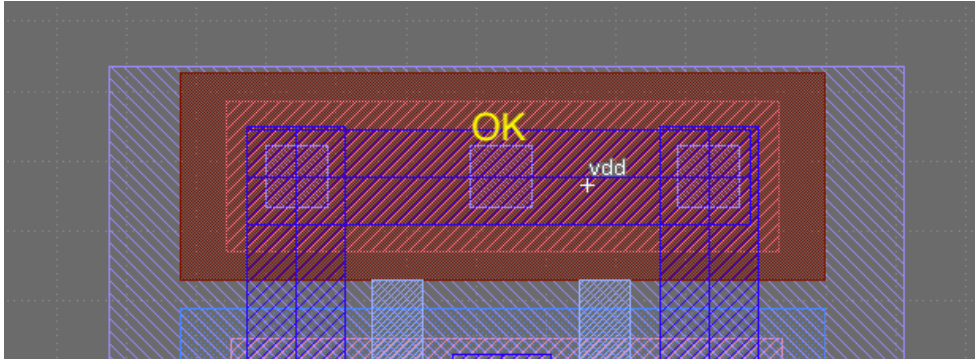
```
1 * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2 *
3 * Modified the original SPICE net
4 * 1) there is no tie-down diodes in the original layout
5 * 2) added terminals [VDD GND IN1 IN2 OUT NWELL SUBSTRATE]
6 * 3) transistor name changed to the default (MbreakN, MbreakP) ==> (NMOS, PMOS)
7 * 4) S-D connections are !CORRECT!
8
9 .SUBCKT nand VDD GND IN1 IN2 OUT NWELL SUBSTRATE
10 M_M6 N15943 VIN2 GND SUBSTRATE NMOS L=0.18um W=1.08um
11 M_M3 OUT VIN1 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
12 M_M5 OUT VIN1 VDD NWELL PMOS L=0.18um W=1.62um
13 M_M4 OUT VIN2 VDD NWELL PMOS L=0.18um W=1.62um
14 .ENDS
```

\$choice=4 nand.cir

I expect this should match!

18. Using DMOS4 for a **2-input NAND**

3. Check the original design in GDS2.

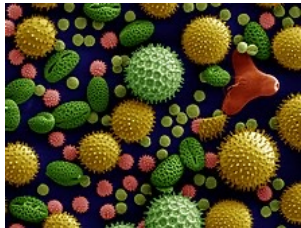


18. Using DMOS4 for a 2-input NAND

4. Prepare my design in OASIS:
"nand-DMOS4.oas"

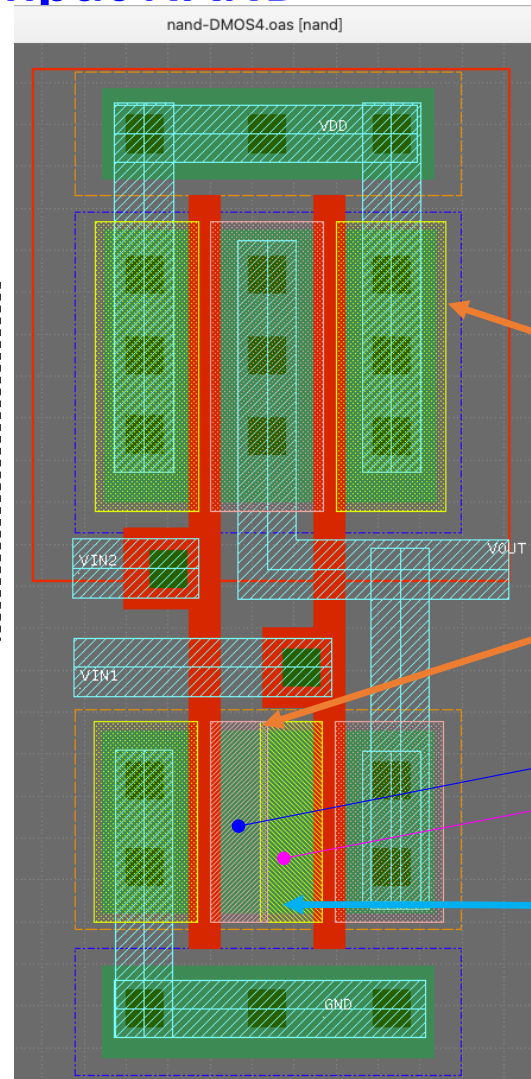
Monoecy (/mə'ni:si/;
adj. **monoecious** /mə'ni:fəs/)^[1] is
a sexual system in seed
plants where
separate male and female cones
or flowers are present on the same
plant.

[Monoecy - Wikipedia](#)



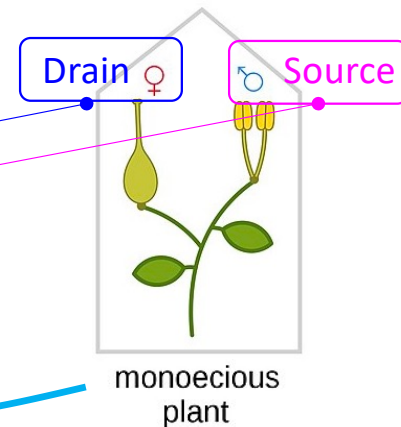
Pollen can be
seen as the
majority carrier.

[Pollen - Wikipedia](#)



Added some
helper layers
for
experiments

Layers	
active	1/0
nwell	3/0
pplus	11/0
nplus	12/0
poly	41/0
contact	39/0
metal1	46/0
metal1_lbs	46/1
S-Recognition	51/0
MaleS-Recognition	51/1
D-Recognition	52/0
FemaleD-Recognition	52/1



18. Using DMOS4 for a 2-input NAND


5. Modify the LVS script for testing and debugging.

```
136 nss
137 ndd
138 nsmale
139 ndfemale
140
141 #-----
142 # [5] Device extracti
143 #-----
144 # PMOS (with ordinary
145 extract_devices(dmos4
146 # not realis
147
148 # NMOS (with ordinary
149 extract_devices(dmos4
150 # I call them "male source" and "female
151 male_S_rec = input(51, 1) # helper lay
152 female_D_rec = input(52, 1) # helper lay
153
154 # NMOS (with female
155 extract_devices(dmos4
156
157 # NMOS (with female
158 extract_devices(dmos4
159
160 #-----
161 # [6] Define connecti
162 #-----
163 # Inter-layer
164 connect(pss,
165 connect(pdd,
166 connect(nss,
167 connect(ndd,
168 connect(poly,
169 connect(contact,
170 connect(metal1,
171 connect(nsmale,
172
173 # Global
174 # [7] Com
175 #connect_global(bulk,
176 #connect_global(ptie,
177 connect_global(bulk,
178 connect_global(nwell, "NWell")
179
180 # EOF
181
182
183
184
185
186
187
188
189
```

```
91
92
93 #-----
94 # [3] Drawing layers
95 #-----
96 active = input(1, 0)
97 nwell = input(3, 0)
98 pplus = input(11, 0)
99 nplus = input(12, 0)
100 poly = input(41, 0)
101 contact = input(39, 0)
102 metall = input(46, 0)
103 metall_lbl = labels(46, 1)
104 source_rec = input(51, 0) # not realis
105 drain_rec = input(52, 0) # not realis
106
107 # More helper layers to virtually disti
108 # of a "monoecious plant" (extracted as
109 # I call them "male source" and "female
110 male_S_rec = input(51, 1) # helper lay
111 female_D_rec = input(52, 1) # helper lay
112
113 # Bulk layer for terminal provisioning
114 bulk = polygon_layer
115
116 #-----
117 # [4] Computed layers
118 #-----
119 #
120 # In a future "nand-DMOS4B.lvs", we will
121 # to use the S- and D-recognition helper
122 #-----
123 active_in_nwell = active & nwell
124 pactive = active_in_nwell & pp
125 pgate = pactive & poly
126 psd = pactive - pgate
127 #ntie = active_in_nwell & n
128 pss = psd & source_rec
129 pdd = psd & drain_rec
130
131 active_outside_nwell = active - nwell
132 nactive = active_outside_nwell
133 ngate = nactive & poly
134 nsd = nactive - ngate
135 #ptie = active_outside_nwell & pplus
```

```
46 exit
47 when 1, 4
48 @spicedeck
49 else
50 puts "### (
51 puts "
52 puts "
53 return
54 end
55 end
56 end
57 end
58
59 #-----
60 # [1] Design
61 #-----
62 dirName, baseName =
63 Dir.chdir(dirName)
64
65 source( "nand-DMOS4
66
67 deep
68
69 selector = SpiceDeck
70 selector.GetOneOpt:
71 choice, spicedeck =
72 if choice == 10 #
73 return
74 elsif not [1, 4].i
75 puts "! Your cho
76 return
77 end
78 AboutCurrentDesign
79
80 #-----
81 # [2] Reports
82 #-----
83 # Store the LVS report to this file
84 report_lvs( "nand-DMOS4A-choice#{choice}.lvldb", true )
85
86 # Write the extracted netlist to this file
87 target_netlist( "nand-DMOS4A-choice#{choice}_extracted.cir",
88 write_spice,
89 "Extracted by KLayout with <#{spicedeck}>" )
90
```

```
1 #-----
2 # Original: "lvs 2 1.lylvs" in the attached ZIP file of
3 # https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
4 #
5 # Modified: nand-DMOS4A.lvs
6 # by: Kazzz-S
7 # date: 2023-02-21
8 # aims: 1) test WENSHIH's NAND circuit with two SPICE decks: INCORRECT(original) & CORRECT
9 # 2) use "DMOS4" instead of "MOS4"
10 # ==> https://www.klayout.de/forum/discussion/comment/9484/#Comment_9484
11 # 3) use "helper layers" for source- and drain-recognition
12 #-----
13
14 #-----
15 # [0] Utilities
16 #-----
17 def AboutCurrentDesign(spicedeck)
18 designFile = RBA::CellView::active.filename
19 active layout = RBA::CellView::active.layout
20 puts ">>> Current design file = '#{designFile}'"
21 puts "Top cell name = '#{active.layout.top_cell.name}'"
22 puts "SPICE deck file = '#{spicedeck}'"
23 end
24
25 class SpiceDeckSelector
26 def initialize
27 if $choice == nil # global variable to choose a SPICE deck
28 $choice = 10
29 end
30 @spicedeck = [ $choice, "" ]
31
32 @spiceOpt = { 0 => "exit", 1 => "original", 4 => "correct" }
33 @spiceDECK = { "original" => "nand-incorrect.cir", # (PMOS, NMOS)=(incorrect, incorrect)
34 "correct" => "nand.cir" # (PMOS, NMOS)=( correct, correct)
35 }
36 end
37
38 def GetSpiceDeck
39 return @spicedeck
40 end
41
42 def GetOneOption
43 case $choice
44 when 0
45 puts "Bye! You can ignore the exception if caught."
46 end
47 end
48
```



monoecious plant

18. Using DMOS4 for a 2-input NAND

6. Run the modified LVS script twice: #1/2

The screenshot shows the KLayout LVS tool interface. The left pane displays a circuit layout with transistors M4, M5, M6, and M3, and nodes VDD, GND, VIN1, VIN2, and VOUT. The right pane shows the netlist for the 'nand' cell, listing NMOS and PMOS transistors with their parameters and connections.

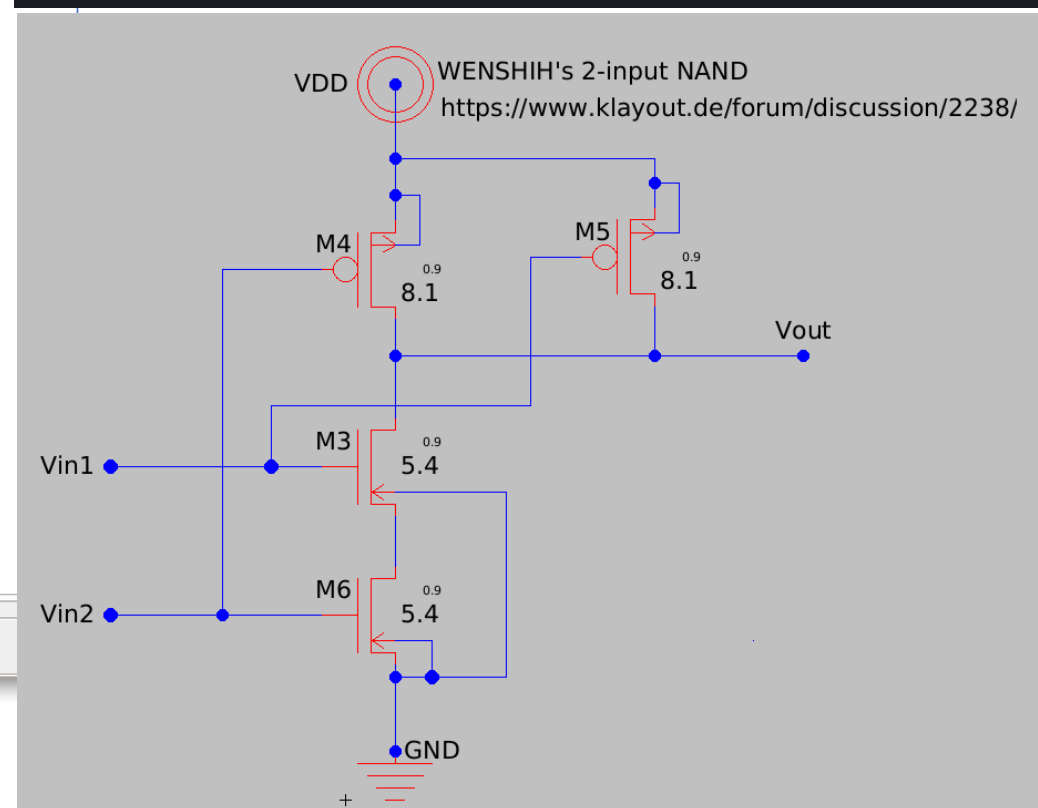
Object	Layout	Reference
Pin		
Net		
Device		
NMOS	\$4 / NMOS [L=0.18, _M6 / NMOS [L=0.18, W=1.08]	
S	GND (1)	N15943 (2)
D	\$6 (2)	GND (2)
G	VIN2 (2)	VIN2 (3)
B	SUBSTRATE (2)	SUBSTRATE (3)
NMOS	\$3 / NMOS [L=0.18, _M3 / NMOS [L=0.18, W=1.08]	
S	\$6 (2)	VOUT (4)
D	VOUT (3)	N15943 (2)
G	VIN1 (2)	VIN1 (3)
B	SUBSTRATE (2)	SUBSTRATE (3)
PMOS	\$2 / PMOS [L=0.18, _M5 / PMOS [L=0.18, W=1.62]	
S	VDD (2)	VOUT (4)
D	VOUT (3)	VDD (3)
G	VIN1 (2)	VIN1 (3)
B	NWELL (2)	NWELL (3)
PMOS	\$1 / PMOS [L=0.18, _M4 / PMOS [L=0.18, W=1.62]	
S	VDD (2)	VOUT (4)
D	VOUT (3)	VDD (3)
G	VIN2 (2)	VIN2 (3)
B	NWELL (2)	NWELL (3)

Configure Probe Net Lock

```
> $choice=1
1
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/NAND-DMOS4/nand-DMOS4.oas'
Top cell name = 'nand'
SPICE deck file = 'nand-incorrect.cir'
```

```
1 * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2 *
3 * Modified the original SPICE net
4 * 1) there is no tie-down diodes in the original layout
5 * 2) added terminals [VDD GND VIN1 VIN2 VOUT NWELL SUBSTRATE]
6 * 3) transistor name changed to the default (MbreakN, MbreakP) ==> (NMOS, PMOS)
7 * 4) Still INCORRECT! ==> Source-Drain interchanged
8
9 .SUBCKT nand VDD GND VIN1 VIN2 VOUT NWELL SUBSTRATE
10 M M6 GND VIN2 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
11 M M3 N15943 VIN1 VOUT SUBSTRATE NMOS L=0.18um W=1.08um
12 M M5 VDD VIN1 VOUT NWELL PMOS L=0.18um W=1.62um
13 M M4 VDD VIN2 VOUT NWELL PMOS L=0.18um W=1.62um
```

Unlike the previous experiment, this net did not match.



18. Using DMOS4 for a 2-input NAND

6. Run the modified LVS script twice: #2/2

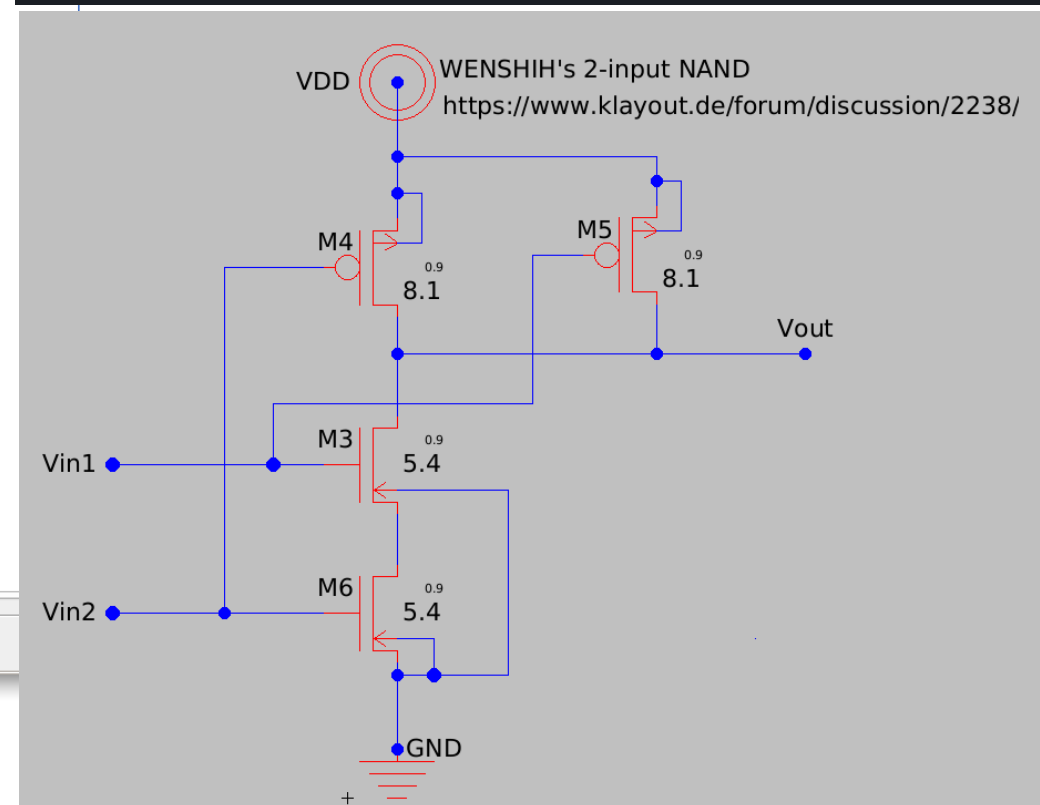
The screenshot displays the KLayout LVS tool interface. The top tabs are 'Netlist', 'Schematic', 'Cross Reference', and 'Log'. The 'Circuits' panel on the left shows a hierarchy starting with 'nand' and 'NAND'. The 'Objects' panel in the center lists various components and their properties, including NMOS and PMOS transistors with their respective parameters like length (L=0.18um) and width (W=1.08um or W=1.62um). The 'Reference' panel on the right shows the corresponding reference components. The bottom status bar indicates the current design file and top cell name.

Configure Probe Net Lock


```
> $choice=4
4
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con1/NAND-DMOS4/nand-DMOS4.oas'
Top cell name = 'nand'
SPICE deck file = 'nand.cir'
```

```
1 | https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2 | *
3 | * Modified the original SPICE net
4 | * 1) there is no tie-down diodes in the original layout
5 | * 2) added terminals [VDD GND IN1 IN2 OUT NWELL SUBSTRATE]
6 | * 3) transistor name changed to the default (MbreakN, MbreakP) ==> (NMOS, PMOS)
7 | * 4) S-D connections are !CORRECT!
8 |
9 | .SUBCKT nand VDD GND IN1 IN2 OUT NWELL SUBSTRATE
10 | M M6 N15943 VIN2 GND SUBSTRATE NMOS L=0.18um W=1.08um
11 | M M3 OUT VIN1 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
12 | M M5 OUT VIN1 VDD NWELL PMOS L=0.18um W=1.62um
13 | M M4 OUT VIN2 VDD NWELL PMOS L=0.18um W=1.62um
```

This net matched as expected!



19. Summary and Conclusions




klayoutmatthias commented 2 days ago • edited ▾

Collaborator 😊 ⋮

For DMOS devices where S/D will not swap, it is important to read the terminals in the correct S/D assignment. Right now, the Spice reader reads it S, G, D, B, but it should be D, G, S, B to be compatible with standard Spice.

Please also see the elaborate discussion here:
<https://www.klayout.de/forum/discussion/2238/how-to-extract-transistor-in-parallel-topology-in-lvs#latest>

◆ As of the commit below, the Inverter LVS and NAND LVS using DMOS4 work properly.



Commit [28883ee3](#) (open)

by [Matthias Koefferlein](#), 2023-02-28 08:22

parent [4d2168a1](#)

Updated golden test data

branches [issue-1304](#), [upstream/issue-1304](#)

Part-V: Appendix

Q&A on Netlist Database Browser

https://www.klayout.de/forum/discussion/comment/9504/#Comment_9504



Detailed documentation: honestly there is no full documentation about the result browser. Maybe the information here is helpful: https://www.klayout.de/doc-qt5/manual/lvs_compare.html#h2-348, specifically "how the compare algorithm works".

7. Netlist Database Browser

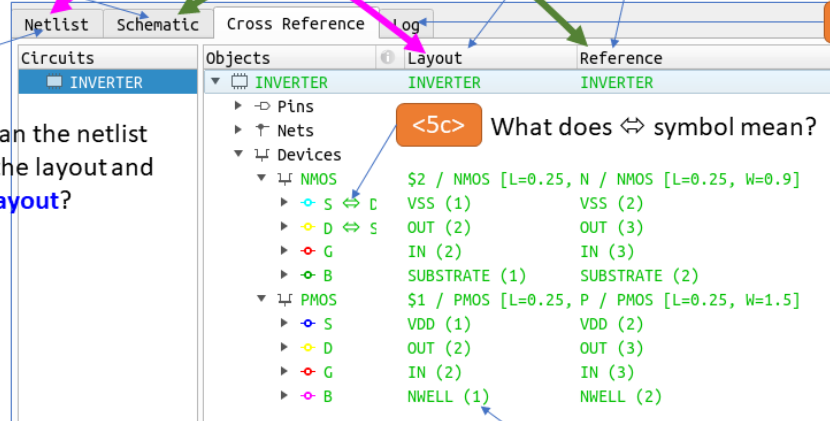
<5> Is there any detailed document interpreting each item that could appear in the netlist database browser?
I've tried to find it but couldn't so far. Some items are very intuitive. However, others are not.

<5b> Does **Schematic** mean the netlist extracted from the reference SPICE deck and correspond to **Reference**?

(2) Extracted from the layout **inv.oas**

(3) Extracted from the reference net **inv.cir**

<5a> Does **Netlist** mean the netlist extracted from the layout and correspond to **Layout**?



<5c> What does ⇔ symbol mean?

<5e> **Log** is empty in this case. However, there would be something to point out such as the incorrect reference SPICE net.

<5d> What does a number in () mean?

KLayout Forum No. 2238 (as Study002: LVS of CMOS Inverter and 2-input NAND)

16



"Netlist" is the layout-extracted netlist,
"Schematic" is the netlist read from the reference schematic



The Log will contain messages in case of mismatches of warnings which hopefully are helpful to understand the root cause. The compare algorithm is a backtracking implementation and without some hints it is often difficult to understand the results. If there is a match, the Log will be empty.

KLayout Forum No. 2238 (as Study002: LVS of CMOS Inverter and 2-input NAND)

23

This symbol indicates that terminal swapping has happened to match the devices from schematic and layout (where layout does not necessarily assign S and D in the correct way).
The first label is the layout terminal name, the second label (after the arrow) the schematic terminal name.



7. Netlist Database Browser

<5> Is there any detailed document interpreting each item that could appear in the netlist database browser?
I've tried to find it but couldn't so far. Some items are very intuitive. However, others are not.

<5b> Does **Schematic** mean the netlist extracted from the reference SPICE deck and correspond to **Reference**?

(2) Extracted from the layout **inv.oas**

(3) Extracted from the reference net **inv.cir**

<5a> Does **Netlist** mean the netlist extracted from the layout and correspond to **Layout**?

Netlist	Schematic	Cross Reference	Log
Circuits	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
<ul style="list-style-type: none"> Pins Nets Devices <ul style="list-style-type: none"> NMOS <ul style="list-style-type: none"> S ↔ D D ↔ S G B PMOS <ul style="list-style-type: none"> S D G B 	<ul style="list-style-type: none"> INVERTER <ul style="list-style-type: none"> Pins Nets Devices <ul style="list-style-type: none"> NMOS <ul style="list-style-type: none"> \$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9] VSS (1) OUT (2) IN (2) SUBSTRATE (1) PMOS <ul style="list-style-type: none"> \$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5] VDD (1) OUT (2) IN (2) NWELL (1) 	<ul style="list-style-type: none"> INVERTER <ul style="list-style-type: none"> Pins Nets Devices <ul style="list-style-type: none"> NMOS <ul style="list-style-type: none"> VSS (2) OUT (3) IN (3) SUBSTRATE (2) PMOS <ul style="list-style-type: none"> VDD (2) OUT (3) IN (3) NWELL (2) 	

<5c> What does ↔ symbol mean?

<5e> **Log** is empty in this case. However, there would be something to point out such as the incorrect reference SPICE net.

<5d> What does a number in () mean?



The "1" here means the number of objects attached to the net when you expand the tree node.

Pins are not generated for top level circuits, so the layout netlist has no pins where the schematic has one. That is why the object count is smaller by one in the layout netlist.

