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

Aim:

To understand the basics of KLayout's LVS

Disclaimer:

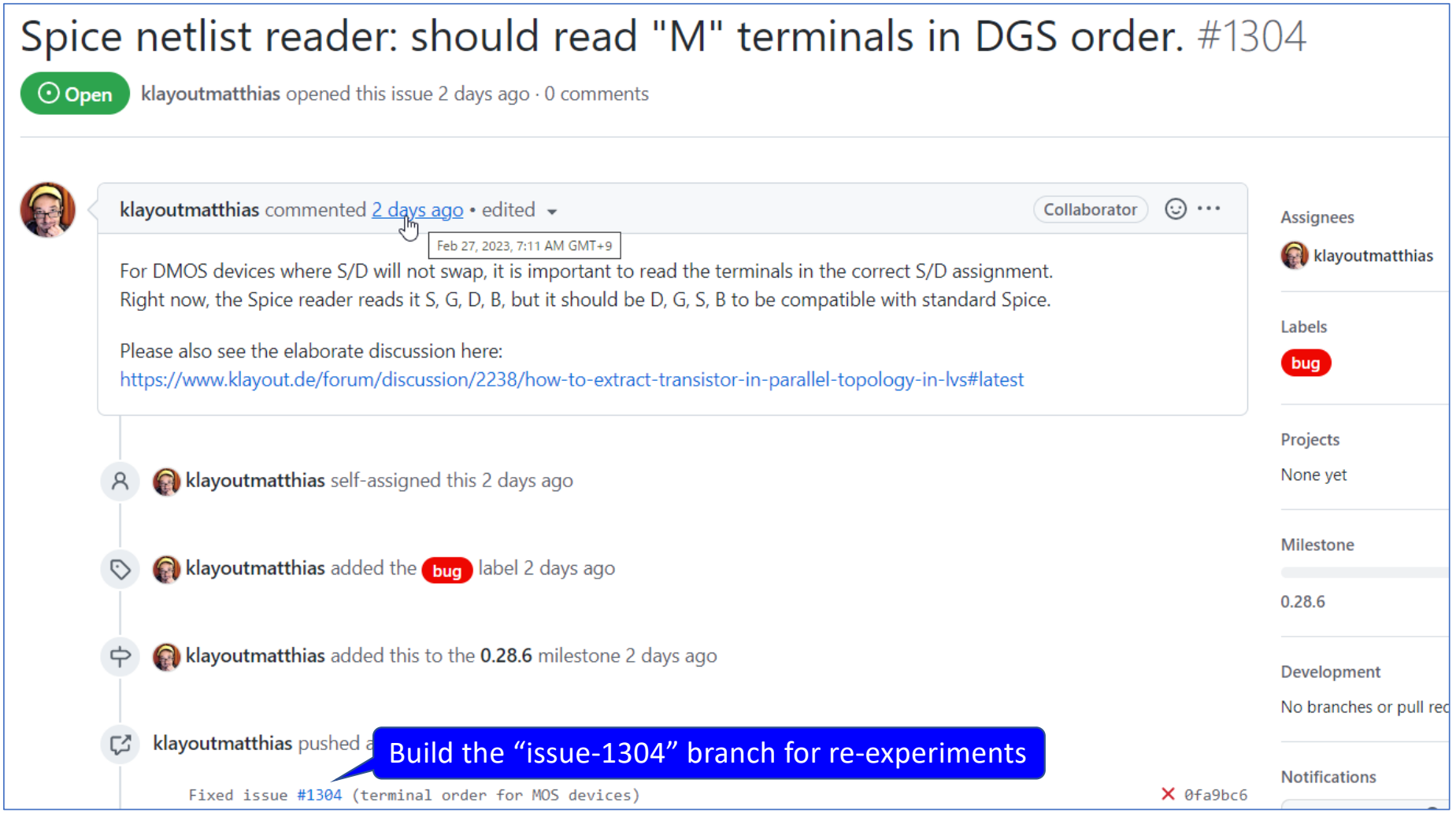
This document is for personal records only. There is NO WARRANTY on technical correctness.

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

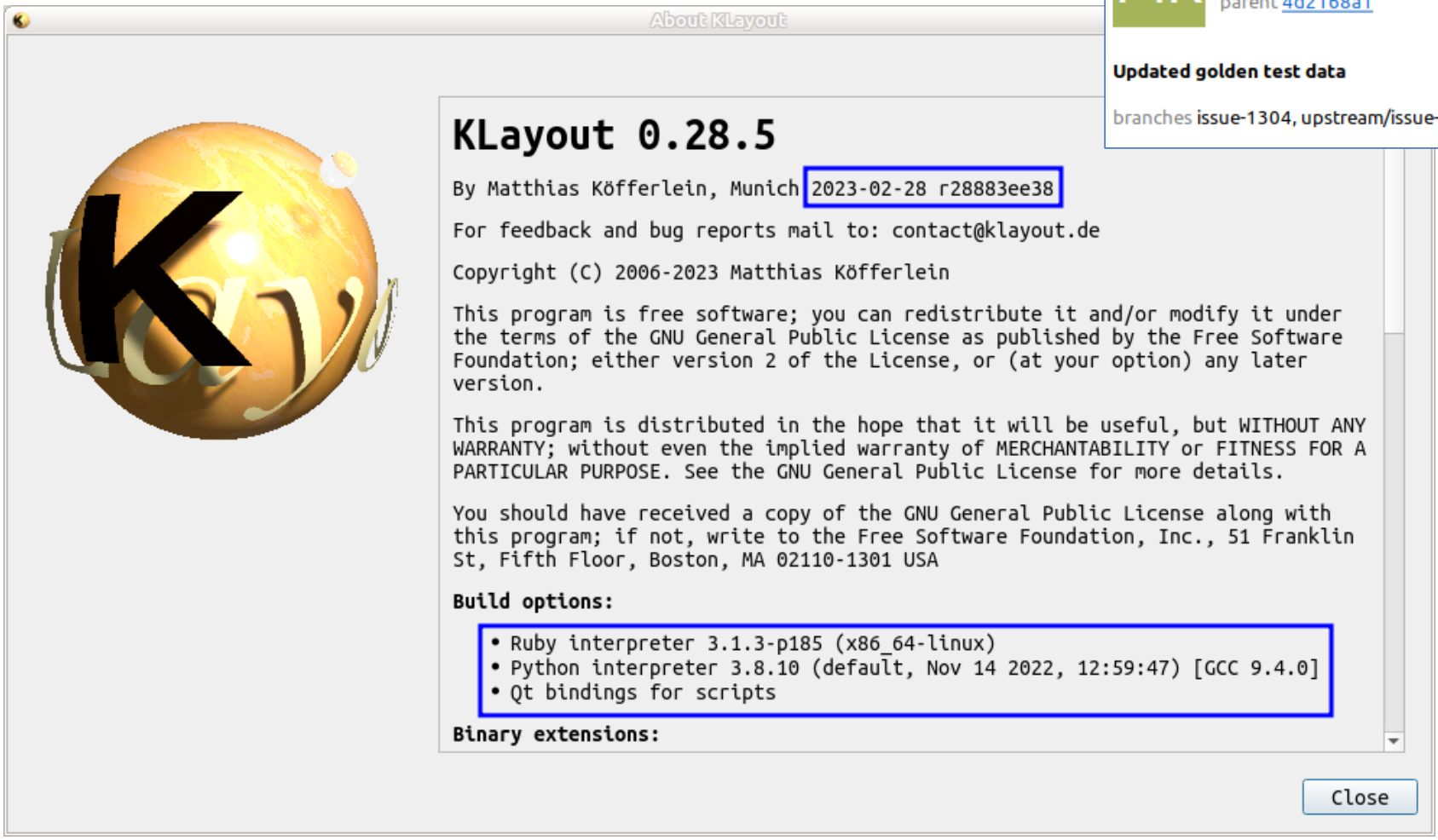
Part-IV: Re-experiments

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

16. Confirmed Issue

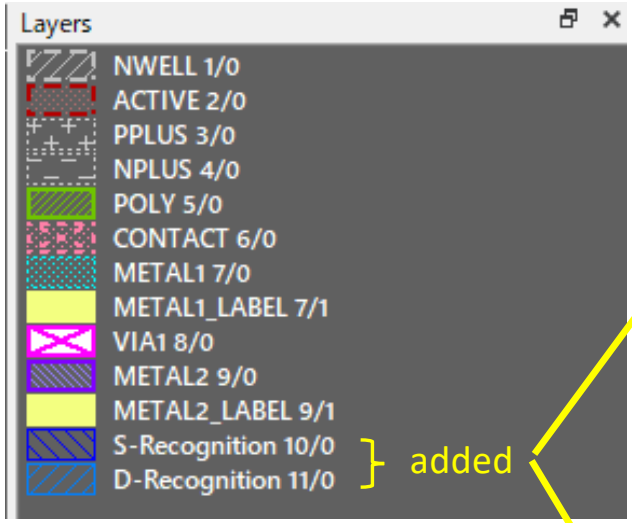


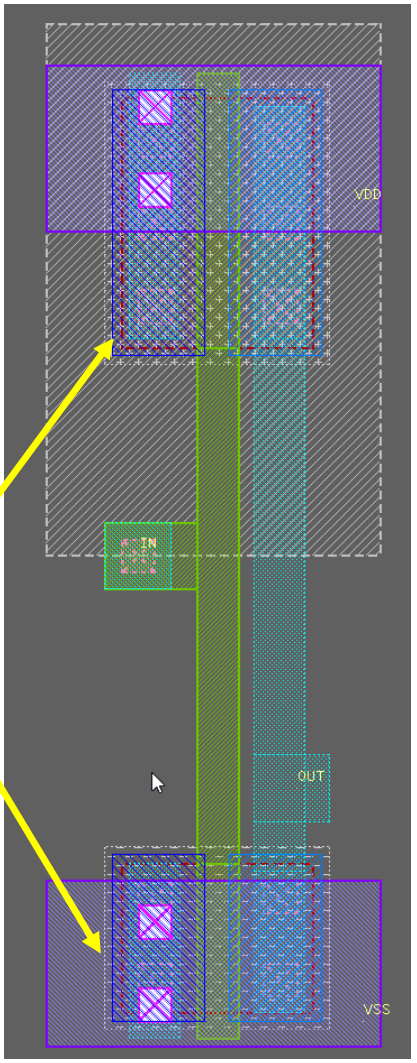
16. Confirmed Issue



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.

inv.cir | invX1.cir | invX2.cir | inv-DMOS4.cir

```
* Simple CMOS inverer circuit : original: (PMOS, NMOS)=(incorrect, correct)

.SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
Mn OUT IN VSS SUBSTRATE NMOS W=0.9U L=0.25U
.ENDS
```

$choice=1

D and S are swapped

inv.cir | invX1.cir | invX2.cir | inv-DMOS4.cir

```
* Simple CMOS inverer circuit: Variant X1: (PMOS, NMOS)=(incorrect, incorrect)

.SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
Mn VSS IN OUT SUBSTRATE NMOS W=0.9U L=0.25U
.ENDS
```

$choice=2

inv.cir | invX1.cir | invX2.cir | inv-DMOS4.cir

```
* Simple CMOS inverer circuit: Variant X2: (PMOS, NMOS)=(  correct, incorrect)

.SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
Mp OUT IN VSS NWELL PMOS W=1.5U L=0.25U
Mn VSS IN OUT SUBSTRATE NMOS W=0.9U L=0.25U
.ENDS
```

$choice=3

inv.cir | invX1.cir | invX2.cir | inv-DMOS4.cir

```
* Simple CMOS inverter circuit
*
* Original: ${klayout_root}/testdata/lvs/inv.cir
*
* Modified: inv-DMOS4.cir (PMOS, NMOS)=(  correct,   correct)
*       by: Kazzz-S
*     date: 2023-02-16
*     aims: 1) correct D-S connections of the PMOS (flipped in the original)

.SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
* Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
Mp OUT IN VDD NWELL PMOS W=1.5U L=0.25U
Mn OUT IN VSS SUBSTRATE NMOS W=0.9U L=0.25U
.ENDS
```

$choice=4

I expect this should match!

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

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

inv-DMOS4A.lvs

```
#-------------------------------------------------------------------------------------------
# Original: ${klayout_root}/testdata/lvs/inv.lvs
#
# Modified: inv-DMOS4A.lvs
#       by: Kazzz-S
#     date: 2023-02-21
#     aims: 1) use the modified "inv-DMOS4.cir" SPICE deck
#           2) use "DMOS4" instead of "MOS4"
#              ==> https://www.klayout.de/forum/discussion/comment/9484/#Comment_9484
#           3) use "helper layers" for source- and drain-recognition
#-------------------------------------------------------------------------------------------

#-----------------------------------------------------
# [0] Utilities
#-----------------------------------------------------
def AboutCurrentDesign(spicedeck)
  designFile    = RBA::CellView::active.filename
  active_layout = RBA::CellView::active.layout
  puts ">>> Current design file = '#{designFile}'"
  puts "          Top cell name = '#{active_layout.top_cell.name}'"
  puts "        SPICE deck file = '#{spicedeck}'"
end

class SpiceDeckSelector
  def initialize
    if $choice == nil # global variable to choose a SPICE deck
      $choice = 10
    end
    @spicedeck = [ $choice, "" ]

    @spiceOpt  = { 0 => "exit", 1 => "original", 2 => "variantX1", 3 => "variantX2", 4 => "correct" }
    @spiceDECK = { "original"  => "inv.cir",       # (PMOS, NMOS)=(incorrect,   correct)
                   "variantX1" => "invX1.cir",     # (PMOS, NMOS)=(incorrect, incorrect)
                   "variantX2" => "invX2.cir",     # (PMOS, NMOS)=(  correct, incorrect)
                   "correct"   => "inv-DMOS4.cir", # (PMOS, NMOS)=(  correct,   correct)
                 }
  end

  def GetSpiceDeck
    return @spicedeck
  end

  def GetOneOption
    case $choice
      when 0
        puts "Bye!  You can ignore the
        exit
      when 1, 2, 3, 4
        @spicedeck = [ $choice, @spice
      else
        puts "### Usage [ 0=>exit, 1=>
        puts "       Set one of above
        puts "       then, execute the
        return
    end
  end
end


#-----------------------------------------------------
# [1] Design
#-----------------------------------------------------
dirName, baseName = File.split( RBA::
Dir.chdir(dirName)

source( "inv-DMOS4.oas", "INVERTER" )

deep

selector = SpiceDeckSelector.new
selector.GetOneOption()
choice, spicedeck = selector.GetSpice
if choice == 10 # 1st run
  return
elsif not [1, 2, 3, 4].include?(choice
  puts "! Your choice '#{choice}' is
  return
end
AboutCurrentDesign(spicedeck)


#-----------------------------------------------------
# [2] Reports
#-----------------------------------------------------
# Store the LVS report to this file
report_lvs( "inv-DMOS4A-choice#{choice}.lvsdb", true )

# Write the extracted netlist to this file
target_netlist( "inv-DMOS4A-choice#{choice}_extracted.cir",
                write spice,
                "Extracted by


#-----------------------------
# [3] Drawing layers
#-----------------------------
nwell       = input(1, 0)
active      = input(2, 0)
pplus       = input(3, 0)
nplus       = input(4, 0)
poly        = input(5, 0)
contact     = input(6, 0)
metal1      = input(7, 0)
metal1_lbl  = labels(7, 1)
via1        = input(8, 0)
metal2      = input(9, 0)
metal2_lbl  = labels(9, 1)
source_rec  = input(10, 0)  #
drain_rec   = input(11, 0)  #

# Bulk layer for terminal pro
bulk = polygon_layer


#-----------------------------
# [4] Computed layers
#
#     In a future "inv-DMOS4B
#     to use the S- and D-rec
#-----------------------------
active_in_nwell       = active
pactive               = active
pgate                 = pactiv
psd                   = pactiv
pss                   = psd &
pdd                   = psd &

active_outside_nwell = active
nactive              = active
ngate                = nactiv
nsd                  = nactive - ngate
nss                  = nsd & source_rec
ndd                  = nsd & drain_rec


#-----------------------------------
# [5] Device extraction
#-----------------------------------
# PMOS transistor device extraction
extract_devices(dmos4("PMOS"), { "S"
                                 "tS"

# NMOS transistor device extraction
extract_devices(dmos4("NMOS"), { "S"
                                 "tS"



#-----------------------------------
# [6] Define connectivity for netlist
#-----------------------------------
# Inter-layer
connect(pss,        contact)
connect(pdd,        contact)
connect(nss,        contact)
connect(ndd,        contact)
connect(poly,       contact)
connect(contact,    metal1)
connect(metal1,     metal1_lbl)   # a
connect(metal1,     via1)
connect(via1,       metal2)
connect(metal2,     metal2_lbl)   # a

# Global
connect_global(bulk,  "SUBSTRATE")
connect_global(nwell, "NWELL")


#-----------------------------------
# [7] Compare section
#-----------------------------------
schematic(spicedeck)

compare

# EOF
```

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

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

inv.cir | invX1.cir | invX2.cir | inv-DMOS4.cir

```
1
2  * Simple CMOS inverer 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	$2 / NMOS [L=0.25,	N / NMOS [L=0.25, W=0.9]
	S	VSS (1)	VSS (2)
	D	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	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 | invX1.cir | invX2.cir | inv-DMOS4.cir

```
1
2  * Simple CMOS inverer 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
```

$choice=2

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	$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	S ⇔ D	VSS (1)	VSS (2)
	D ⇔ S	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	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=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 | invX1.cir | invX2.cir | inv-DMOS4.cir

```
1
2  * Simple CMOS inverer 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	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
	Pins		
	Nets		
	Devices		
	NMOS	$2 / NMOS [L=0.25,	N / NMOS [L=0.25, W=0.9]
	D ⇔ S	OUT (2)	OUT (3)
	S ⇔ D	VSS (1)	VSS (3)
	G	IN (2)	IN (3)
	B	SUBSTRATE (1)	SUBSTRATE (2)
	PMOS	$1 / PMOS [L=0.25,	P / PMOS [L=0.25, W=1.5]
	S	VDD (1)	VSS (3)
	D	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	NWELL (1)	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

inv.cir | invX1.cir | invX2.cir | 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: Kazzz-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!

Netlist | Schematic | Cross Reference | Log

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

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

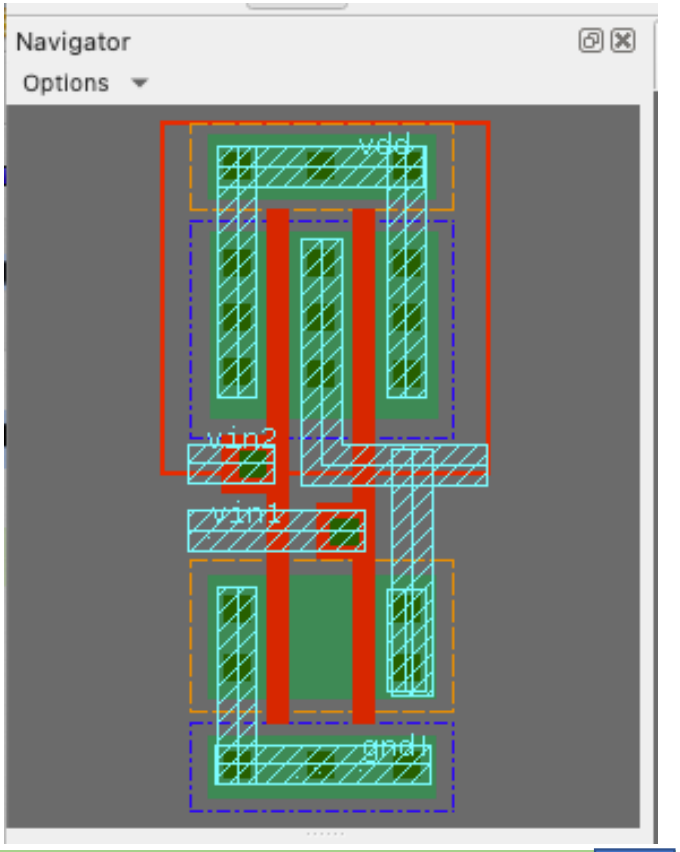
LVS-CMOS-Inverter-NAND.pdf
5.2M

Inv-DMOS4.zip
398.9K

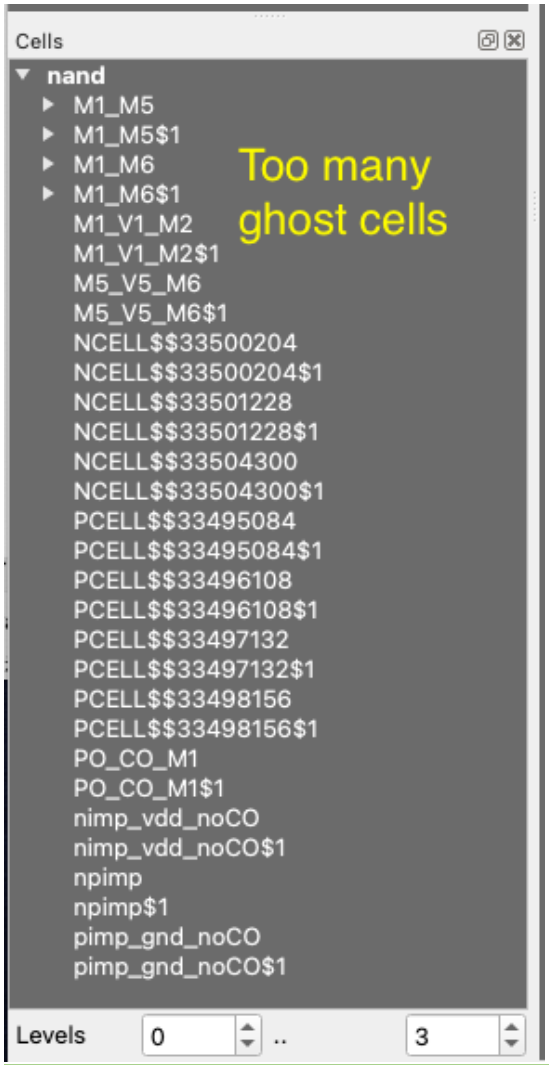
NAND-DMOS4.zip
1.6M

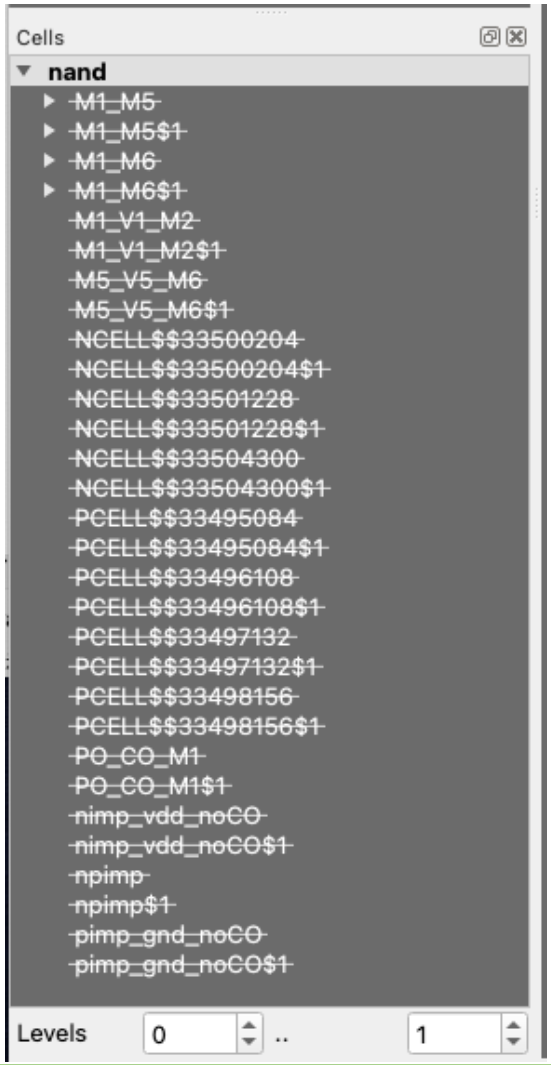
Refer to this ZIP

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

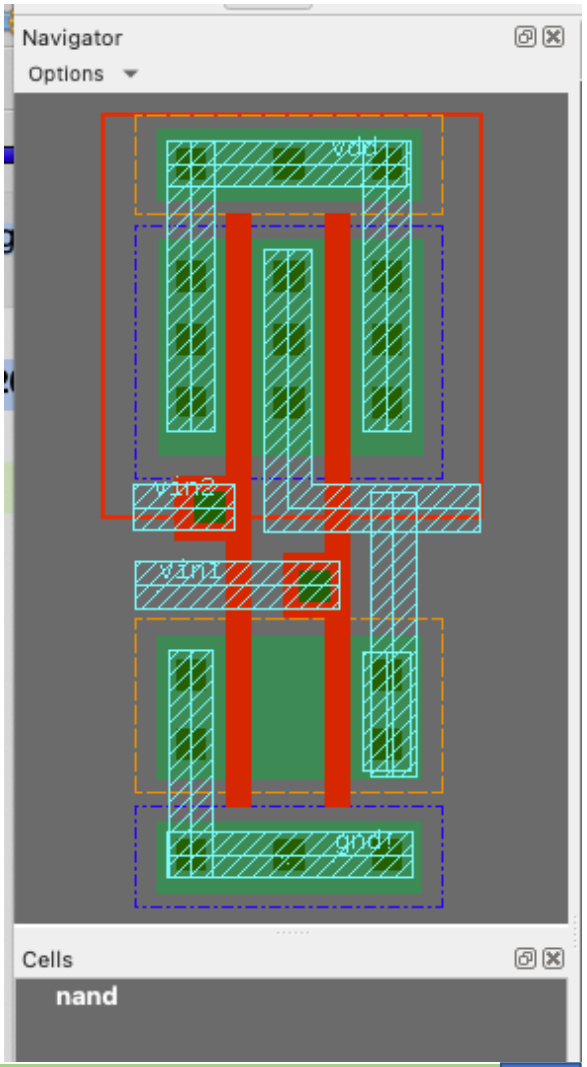


18. Using DMOS4 for a **2-input NAND**



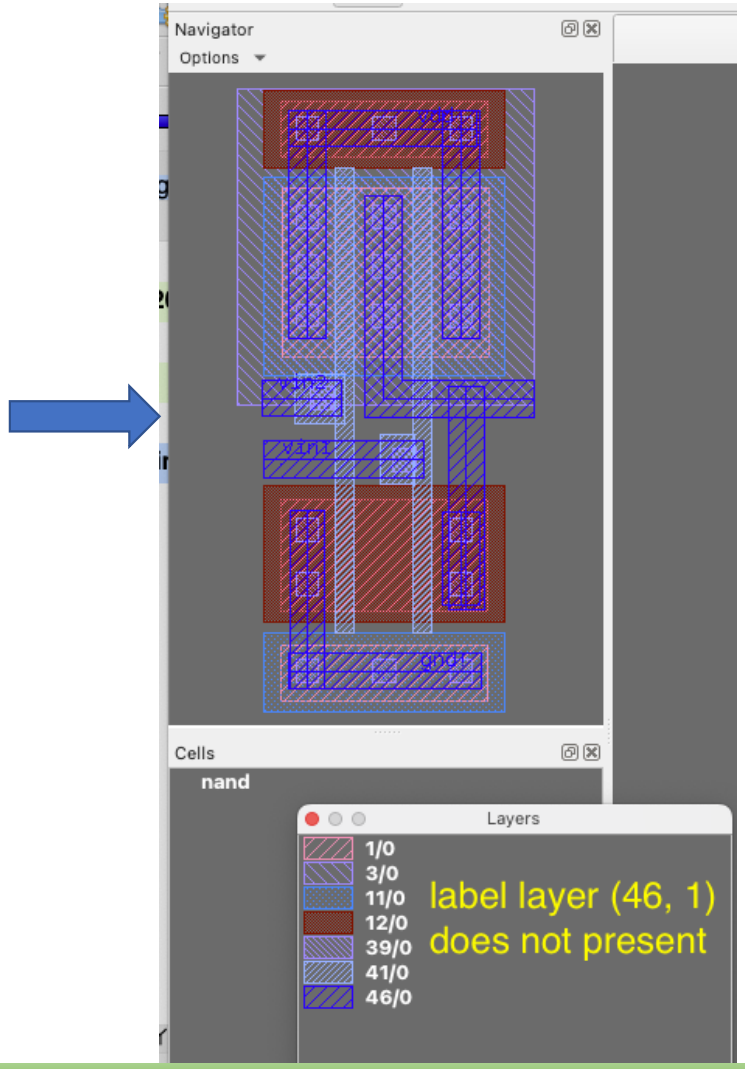


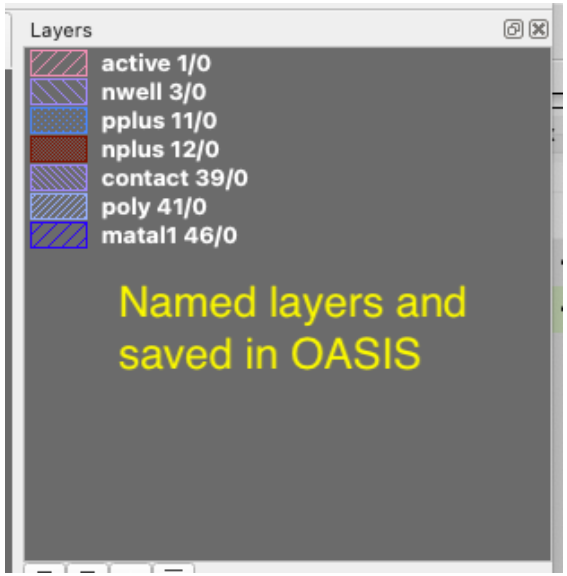
I need only the top cell.



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
```





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

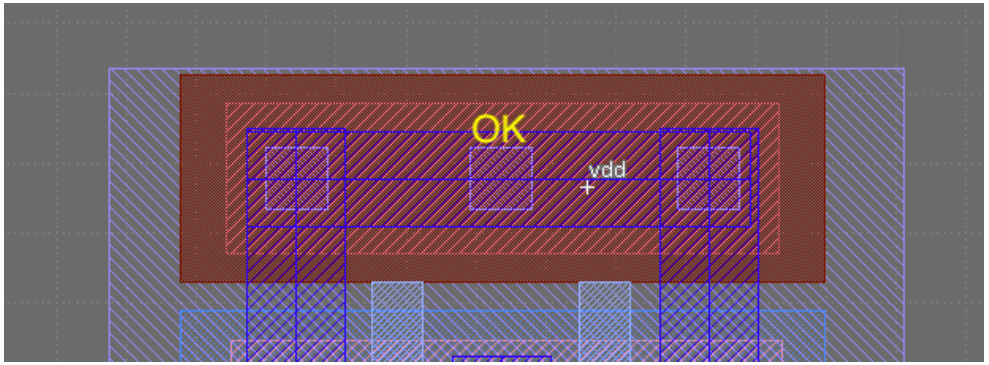
```
1  * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2  *
3  * Modified the original SPICE net
4  *   1) there is no tie-down diodes in the original layout
5  *   2) added terminals [VDD GND IN1 IN2 OUT NWELL SUBSTRATE]
6  *   3) transistor name changed to the default (MbreakN, 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
```

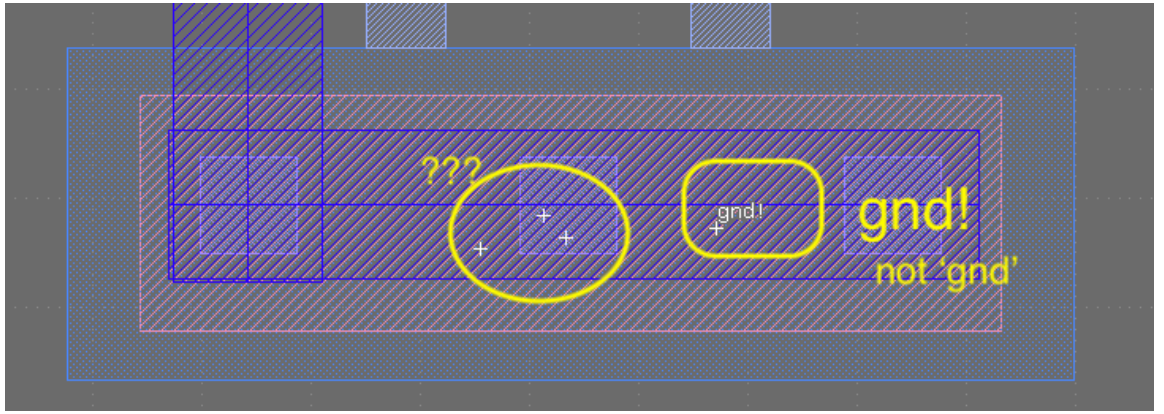
$choice=4 nand.cir

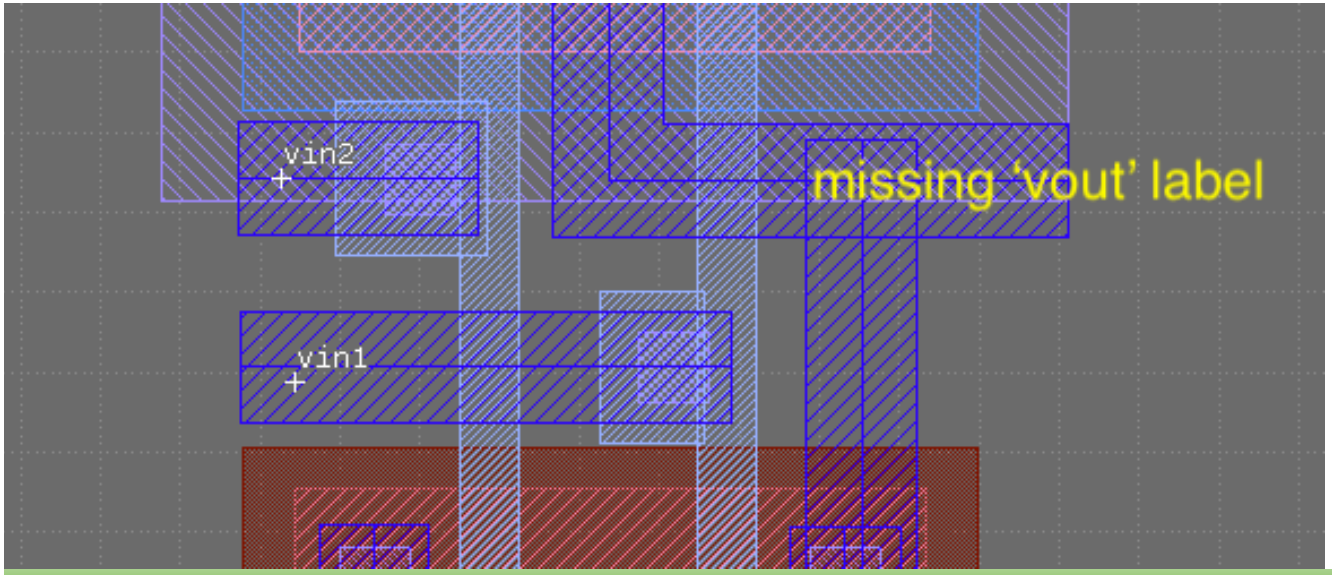
I expect this should match!

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

3. Check the original design in GDS2.





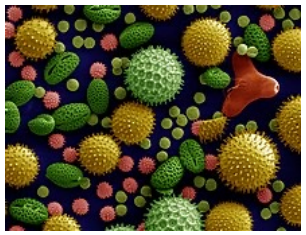


18. Using DMOS4 for a **2-input NAND**

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

Monoecy (/məˈniːsi/; adj. **monoecious** /məˈniːʃəs/)[1] is a sexual system in seed plants where separate male and female cones or flowers are present on the same plant.

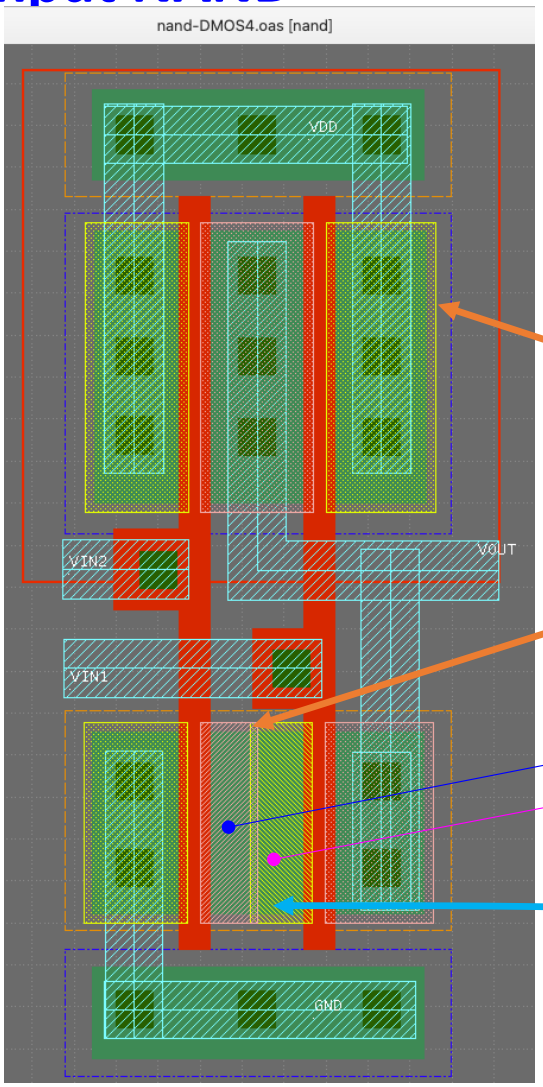
Monoecy - Wikipedia



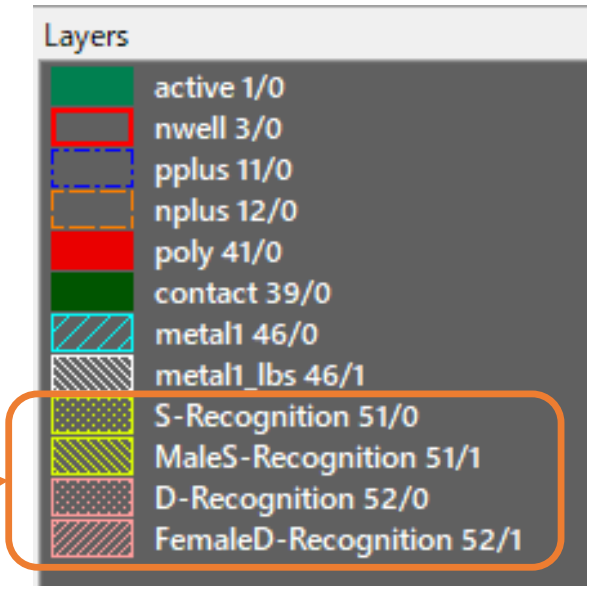
Pollen - Wikipedia

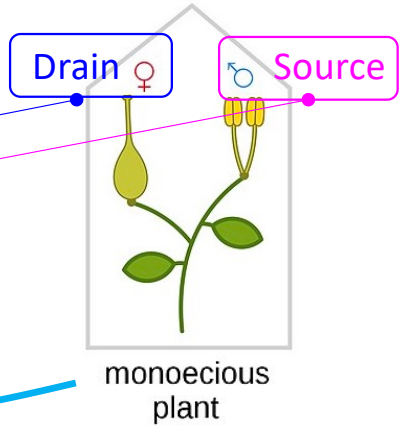
Gate

Pollen can be seen as the majority carrier.



Added some helper layers for experiments





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

5. Modify the LVS script for testing and debugging.

nand-DMOS4A.lvs

```
#-------------------------------------------------------------------------------------------
# Original: "lvs_2_1.lylvs" in the attached ZIP file of
#            https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
#
# Modified: nand-DMOS4A.lvs
#       by: Kazzz-S
#     date: 2023-02-21
#     aims: 1) test WENSHIH's NAND circuit with two SPICE decks: INCORRECT(original) & CORRECT
#           2) use "DMOS4" instead of "MOS4"
#              ==> https://www.klayout.de/forum/discussion/comment/9484/#Comment_9484
#           3) use "helper layers" for source- and drain-recognition
#-------------------------------------------------------------------------------------------

#-------------------------------------------------------
# [0] Utilities
#-------------------------------------------------------
def AboutCurrentDesign(spicedeck)
  designFile    = RBA::CellView::active.filename
  active_layout = RBA::CellView::active.layout
  puts ">>> Current design file = '#{designFile}'"
  puts "          Top cell name = '#{active_layout.top_cell.name}'"
  puts "        SPICE deck file = '#{spicedeck}'"
end

class SpiceDeckSelector
  def initialize
    if $choice == nil # global variable to choose a SPICE deck
      $choice = 10
    end
    @spicedeck = [ $choice, "" ]

    @spiceOpt  = { 0 => "exit", 1 => "original", 4 => "correct" }
    @spiceDECK = { "original"  => "nand-incorrect.cir", # (PMOS, NMOS)=(incorrect, incorrect)
                   "correct"   => "nand.cir"            # (PMOS, NMOS)=(  correct,   correct)
                 }
  end

  def GetSpiceDeck
    return @spicedeck
  end

  def GetOneOption
    case $choice
      when 0
        puts "Bye!  You can ignore the exception if caught."
        exit
      when 1, 4
        @spicedeck
      else
        puts "### I
        puts "
        puts "
        return
    end
  end
end


#-------------------------------------------------------
# [1] Design
#-------------------------------------------------------
dirName, baseName =
Dir.chdir(dirName)

source( "nand-DMOS4

deep

selector = SpiceDe
selector.GetOneOpt
choice, spicedeck =
if choice == 10 # 1
  return
elsif not [1, 4].i
  puts "! Your cho
  return
end
AboutCurrentDesign


#-------------------------------------------------------
# [2] Reports
#-------------------------------------------------------
# Store the LVS report to this file
report_lvs( "nand-DMOS4A-choice#{choice}.lvsdb", true )

# Write the extracted netlist to this file
target_netlist( "nand-DMOS4A-choice#{choice}_extracted.cir",
                write_spice,
                "Extracted by KLayout with <#{spicedeck}>" )
```

```
#-------------------------------------------------------
# [3] Drawing layers
#-------------------------------------------------------
active      = input(1, 0)
nwell       = input(3, 0)
pplus       = input(11, 0)
nplus       = input(12, 0)
poly        = input(41, 0)
contact     = input(39, 0)
metal1      = input(46, 0)
metal1_lbl  = labels(46, 1)
source_rec  = input(51, 0)  # not realis
drain_rec   = input(52, 0)  # not realis

# More helper layers to virtually disti
# of a "monoecious plant" (extracted as
# I call them "male source" and "female
male_S_rec   = input(51, 1) # helper la
female_D_rec = input(52, 1) # helper la

# Bulk layer for terminal provisioning
bulk = polygon_layer


#-------------------------------------------------------
# [4] Computed layers
#
#     In a future "nand-DMOS4B.lvs", we will
#     to use the S- and D-recognition helper
#-------------------------------------------------------
active_in_nwell       = active & nwell
pactive               = active_in_nwell & pp
pgate                 = pactive & poly
psd                   = pactive - pgate
#ntie                  = active_in_nwell & n
pss                   = psd & source_rec
pdd                   = psd & drain_rec

active_outside_nwell  = active - nwell
nactive               = active_outside_nwell
ngate                 = nactive & poly
nsd                   = nactive - ngate
#ptie                  = active_outside_nwell & pplus
```


```
nss
ndd
nsmale
ndfemale


#-------------------------------------------------------
# [5] Device extracti
#-------------------------------------------------------
# PMOS (with ordinary
extract_devices(dmos4


# NMOS (with ordinary
extract_devices(dmos4


# NMOS (with male S)
extract_devices(dmos4


# NMOS (with female D
extract_devices(dmos4


#-------------------------------------------------------
# [6] Define connecti
#-------------------------------------------------------
# Inter-layer
connect(pss,        c
connect(pdd,        c
connect(nss,        c
connect(ndd,        c
connect(poly,       c
connect(contact,    m
connect(metal1,     m
connect(nsmale,     r

# Global
#connect_global(bulk,
#connect_global(ptie,
connect_global(bulk,
connect_global(nwell, "NWELL")
```

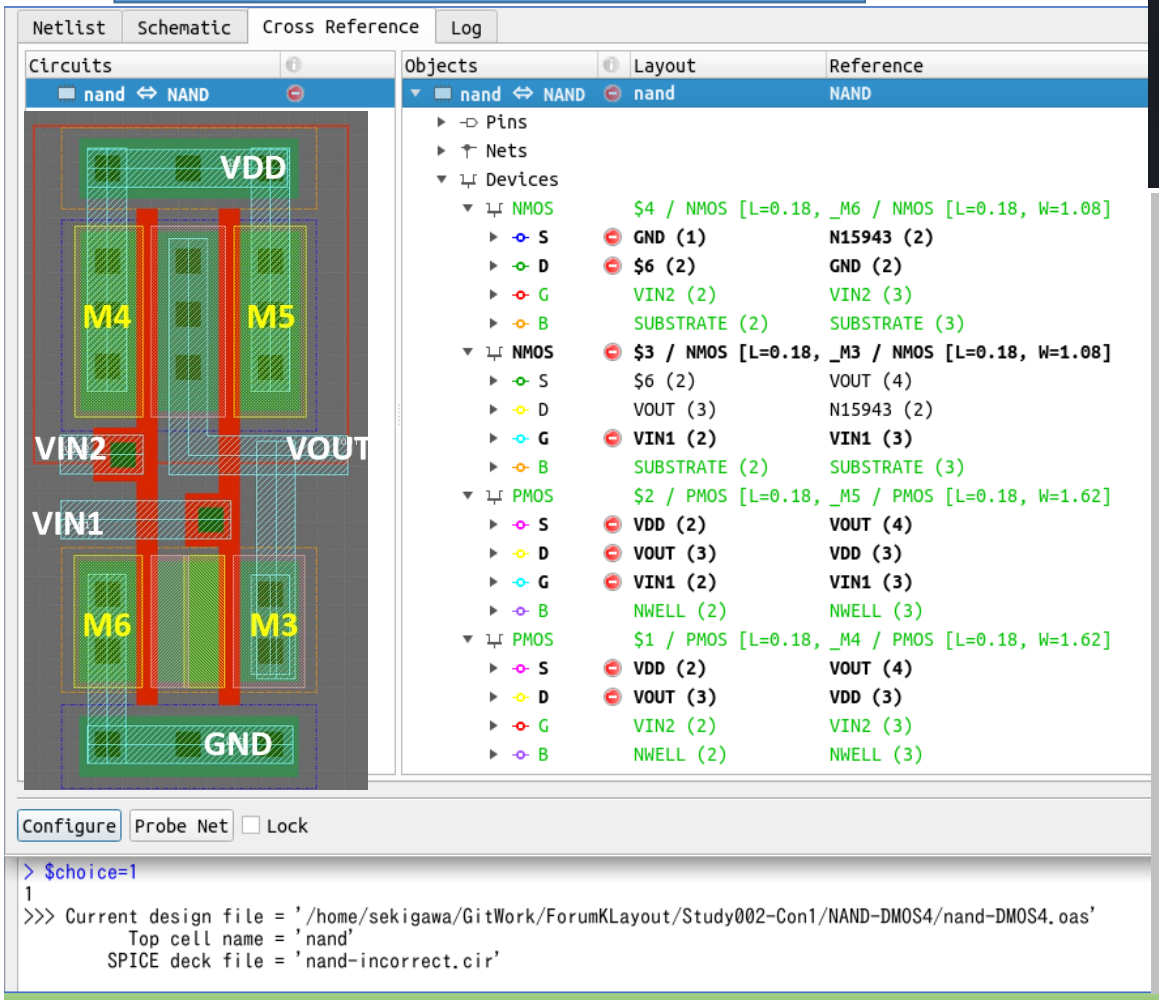
```
#-------------------------------------------------------
# [7] Com
#-------------------------------------------------------
schematic

compare

# EOF
```

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

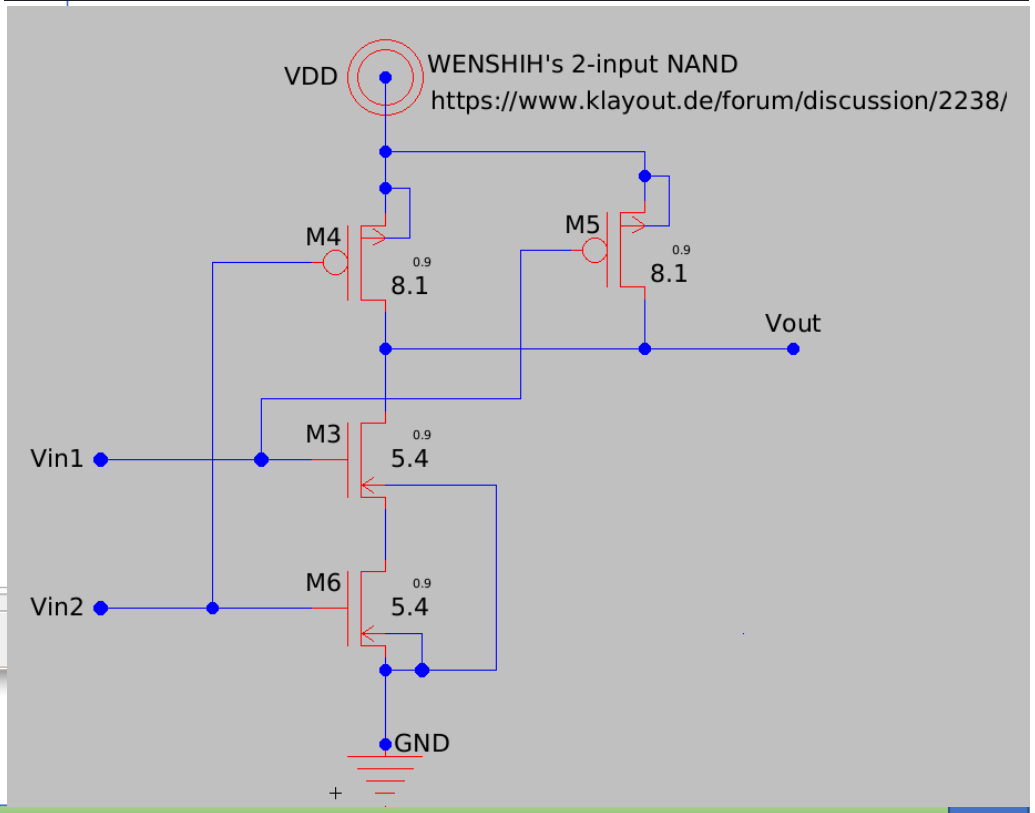
6. Run the modified LVS script twice: #1/2

```
1  * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2  *
3  * Modified the original SPICE net                                    $choice=1
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.




```
> $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'
```



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

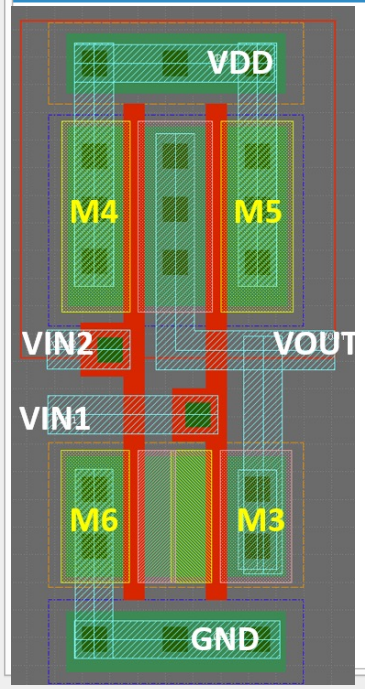
6. Run the modified LVS script twice: #2/2

```
1  * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2  *
3  * Modified the original SPICE net                                    $choice=4
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!

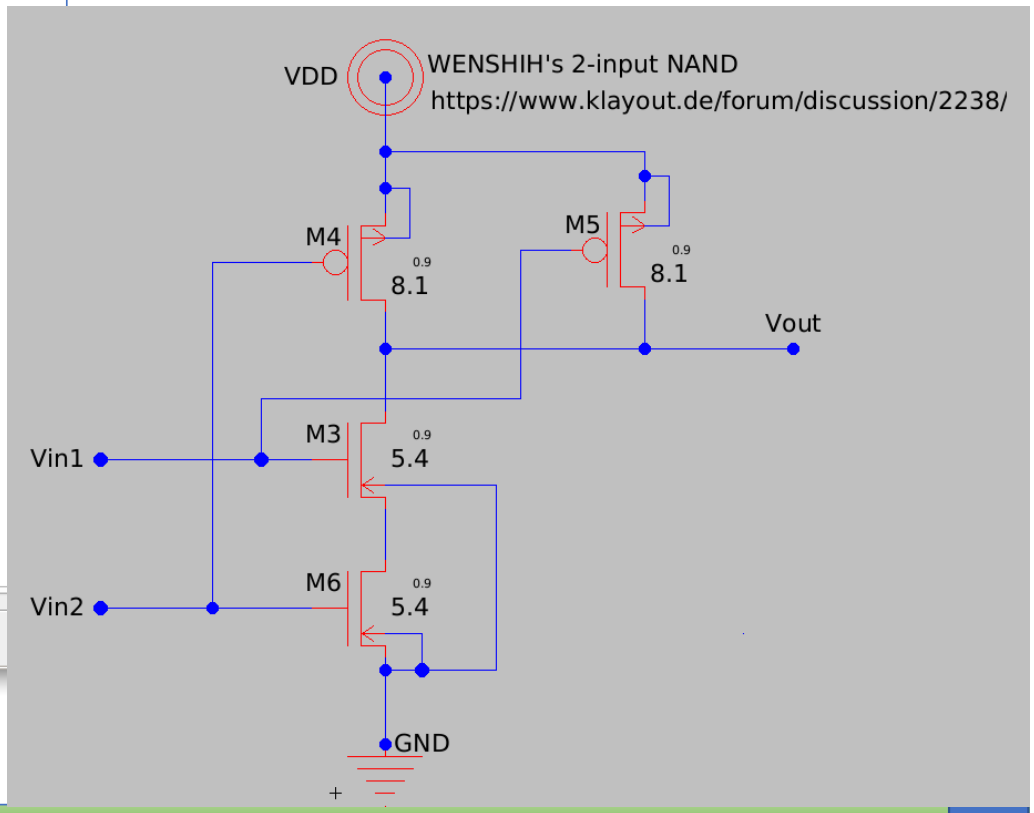
Netlist | Schematic | Cross Reference | Log

Circuits	Objects	Layout	Reference
nand ⇔ NAND	nand ⇔ NAND	nand	NAND
	Pins		
	Nets		
	Devices		
	NMOS	$4 / NMOS [L=0.18,	_M6 / NMOS [L=0.18, W=1.08]
	S	GND (1)	GND (2)
	D	$6 (2)	N15943 (2)
	G	VIN2 (2)	VIN2 (2)
	B	SUBSTRATE (2)	SUBSTRATE (3)
	NMOS	$3 / NMOS [L=0.18,	_M3 / NMOS [L=0.18, W=1.08]
	S	$6 (2)	N15943 (2)
	D	VOUT (3)	OUT (4)
	G	VIN1 (2)	VIN1 (2)
	B	SUBSTRATE (2)	SUBSTRATE (3)
	PMOS	$2 / PMOS [L=0.18,	_M5 / PMOS [L=0.18, W=1.62]
	S	VDD (2)	VDD (3)
	D	VOUT (3)	OUT (4)
	G	VIN1 (2)	VIN1 (2)
	B	NWELL (2)	NWELL (3)
	PMOS	$1 / PMOS [L=0.18,	_M4 / PMOS [L=0.18, W=1.62]
	S	VDD (2)	VDD (3)
	D	VOUT (3)	OUT (4)
	G	VIN2 (2)	VIN2 (2)
	B	NWELL (2)	NWELL (3)



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'
```



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.

MK

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?

Netlist | Schematic | Cross Reference | Log

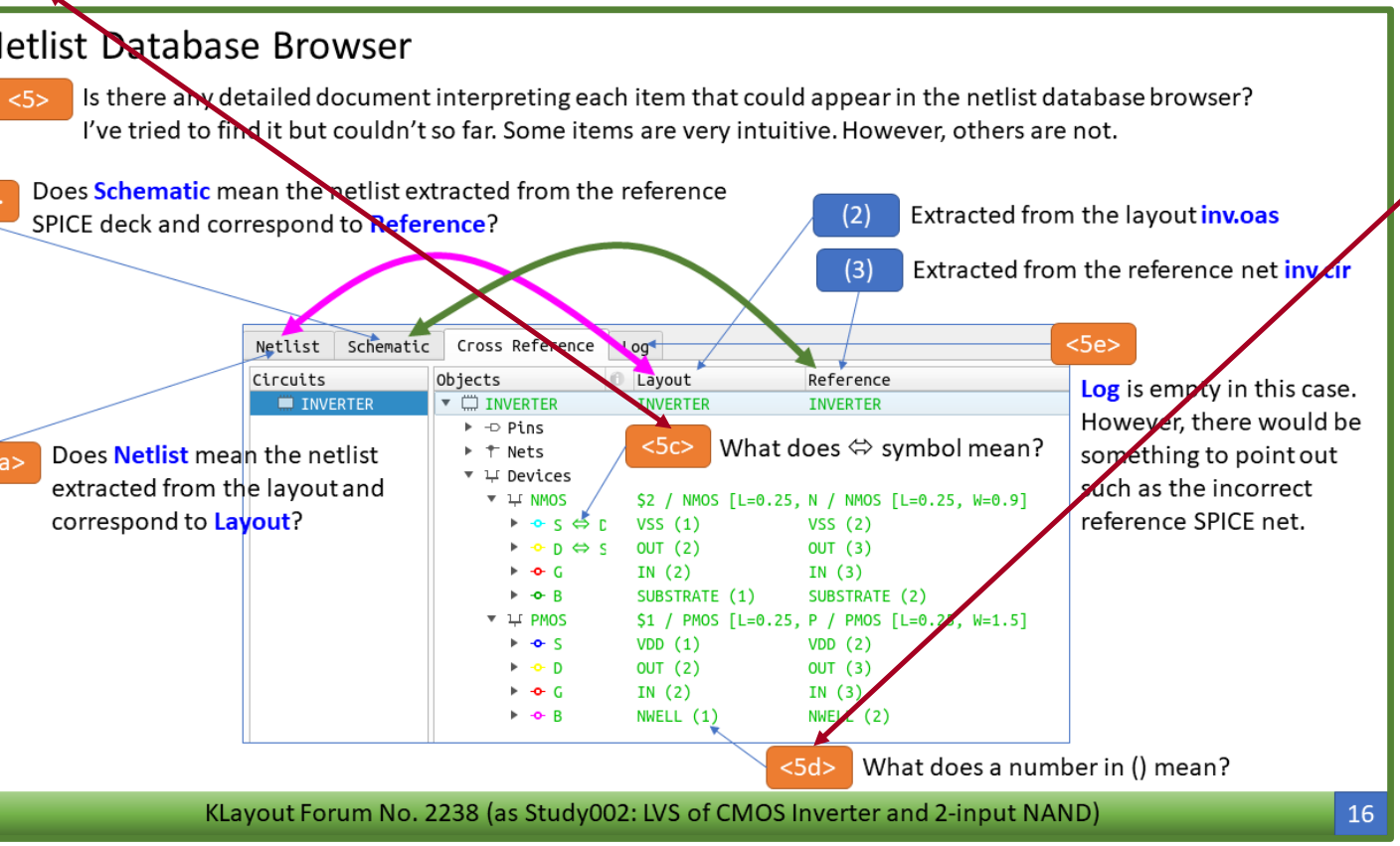
Circuits	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
	Pins		
	Nets		
	Devices		
	NMOS	$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	S ⇔ D	VSS (1)	VSS (2)
	D ⇔ S	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	SUBSTRATE (1)	SUBSTRATE (2)
	PMOS	$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]	
	S	VDD (1)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

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.

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.



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.