LVS of CMOS Inverter and 2-input NAND

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 (2023-02-21)

Part-I: Getting Started

Summary of My Confusion

1. Study Materials

Main Index » KLayout User Manual » Layout vs. Schematic (LVS)

Layout vs. Schematic (LVS)

LVS is a verification step which checks whether a layout matches the circuit from the schematic. The LVS feature is described in the following topic chapters:

- Layout vs. Schematic (LVS) Overview
- LVS Introduction
- LVS Devices
- LVS Device Classes
- LVS Device Extractors
- LVS Input/Output
- LVS Connectivity
- LVS Compare
- LVS Netlist Tweaks

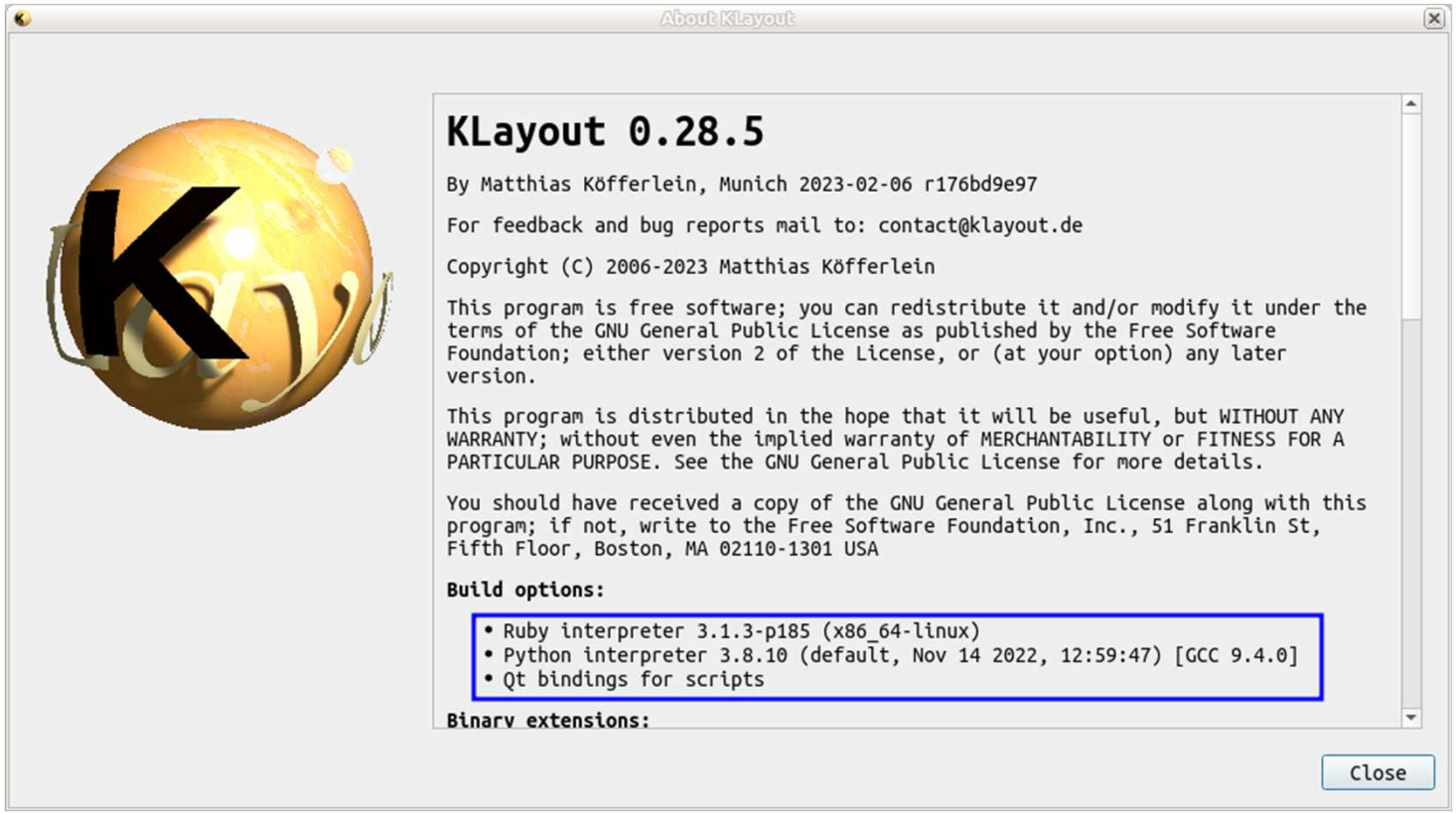
A reference for the functions and objects available for LVS scripts can be found here: LVS Reference.

Forum

[1] https://www.klayout.de/forum/discussion/2234/net-check-on-package-level

[2] https://www.klayout.de/forum/discussion/2238/how-to-extract-transistor-in-parallel-topology-in-lvs

2. Study Environment and Legends in this Document



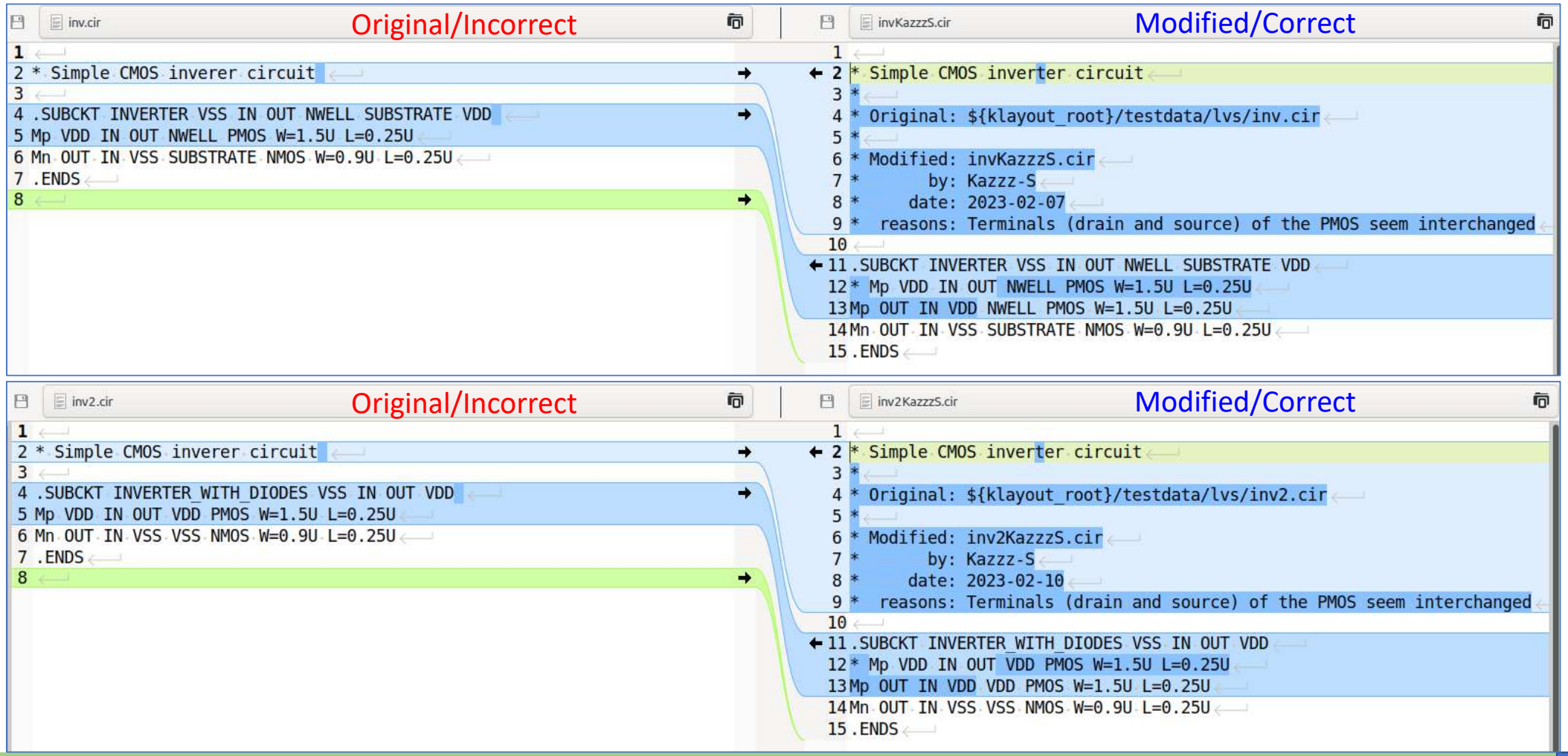
Legends:

(1) my understandings/guesses

<1> my questions

3. Incorrect SPICE Net

<1> The original **inv.cir** and **inv2.cir** seem incorrect; or simple typos?



3. Incorrect SPICE Net

(1) PMOS' swapped drain-source connections in the reference SPICE nets could be **<u>intentional</u>** to test the internal algorithm of LVS because in the Forum [2] …

Matthias
6:53AM on 2023-02-13

@WENSHIH: source and drain are interchangeable in the standard MOS device. So that is not the problem here. KLayout will basically assign S and D randomly and try both ways during compare (and also during device combination).

"combine_devices" will not have an effect as there are no parallel devices as @dick_freebird pointed out.

Could you show us your schematic netlist? After all, this is a simple NAND, so that should actually be very simple.

Matthias

3. Incorrect SPICE Net

<2> Then, why do both **extracted** (resultant) nets coincide but are incorrect (drain and source interchanged)?

inv_extracted.cir

```
1 * Extracted by KLayout
2
3 * cell INVERTER
4 .SUBCKT INVERTER
5 * net 1 IN
6 * net 2 VSS
7 * net 3 VDD
8 * net 4 OUT
9 * net 5 NWELL
10 * net 6 SUBSTRATE
11 * device instance $1 r0 *1 1.025,4.95 PMOS
12 M$1 3 1 4 5 PMOS L=0.25U W=1.5U AS=0.675P AD=0.675P PS=3.9U PD=3.9U
13 * device instance $2 r0 *1 1.025,0.65 NMOS
14 M$2 2 1 4 6 NMOS L=0.25U W=0.9U AS=0.405P AD=0.405P PS=2.7U PD=2.7U
15 .ENDS INVERTER
```

inv_extracted_KazzzS.cir

```
1 * Extracted by KLayout with <invKazzzS.cir>
2
3 * cell INVERTER
4 .SUBCKT INVERTER
5 * net 1 IN
6 * net 2 VSS
7 * net 3 VDD
8 * net 4 OUT
9 * net 5 NWELL
10 * net 6 SUBSTRATE
11 * device instance $1 r0 *1 1.025,4.95 PMOS
12 M$1 3 1 4 5 PMOS L=0.25U W=1.5U AS=0.675P AD=0.675P PS=3.9U PD=3.9U
13 * device instance $2 r0 *1 1.025,0.65 NMOS
14 M$2 2 1 4 6 NMOS L=0.25U W=0.9U AS=0.405P AD=0.405P PS=2.7U PD=2.7U
15 .ENDS INVERTER
```

inv2_extracted.cir

```
1 * Extracted by KLayout
2
3 * cell INVERTER_WITH_DIODES
4 .SUBCKT INVERTER_WITH_DIODES
5 * net 1 IN
6 * net 2 VDD
7 * net 3 OUT
8 * net 4 VSS
9 * device instance $1 r0 *1 1.025,4.95 PMOS
10 M$1 2 1 3 2 PMOS L=0.25U W=1.5U AS=0.675P AD=0.675P PS=3.9U PD=3.9U
11 * device instance $2 r0 *1 1.025,0.65 NMOS
12 M$2 4 1 3 4 NMOS L=0.25U W=0.9U AS=0.405P AD=0.405P PS=2.7U PD=2.7U
13 .ENDS INVERTER_WITH_DIODES
```

inv2_extracted_KazzzS.cir

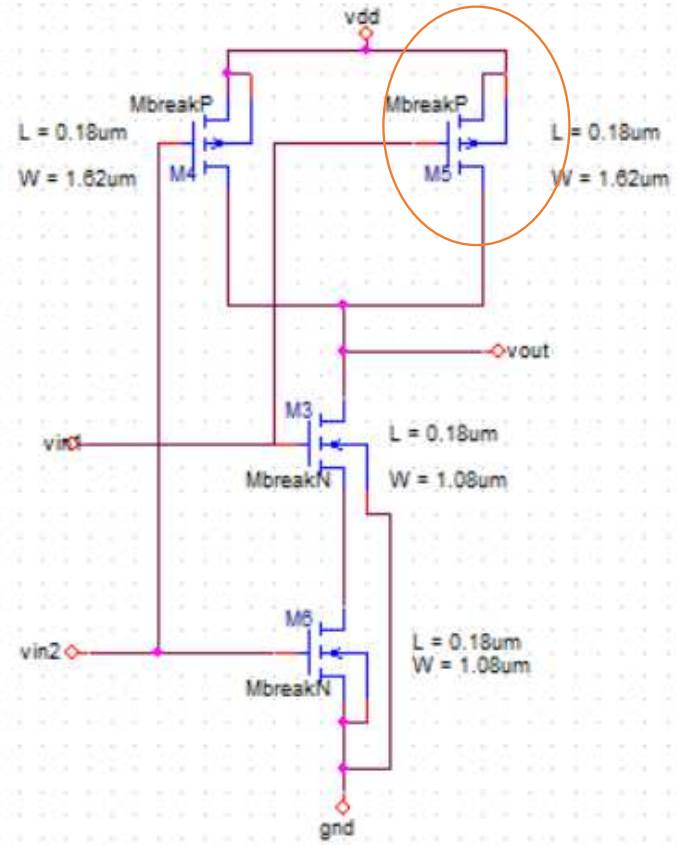
```
1 * Extracted by KLayout with <inv2KazzzS.cir>
2
3 * cell INVERTER_WITH_DIODES
4 .SUBCKT INVERTER_WITH_DIODES
5 * net 1 IN
6 * net 2 VDD
7 * net 3 OUT
8 * net 4 VSS
9 * device instance $1 r0 *1 1.025,4.95 PMOS
10 M$1 2 1 3 2 PMOS L=0.25U W=1.5U AS=0.675P AD=0.675P PS=3.9U PD=3.9U
11 * device instance $2 r0 *1 1.025,0.65 NMOS
12 M$2 4 1 3 4 NMOS L=0.25U W=0.9U AS=0.405P AD=0.405P PS=2.7U PD=2.7U
13 .ENDS INVERTER_WITH_DIODES
```

4. Another Sample of Incorrect SPICE Net

Incorrect. Exported by the schematic capture tool?
Or manually edited for testing? See slide #9 - #11.

WENSHIH
February 13 edited 12:14AM on 2023-02-14

Here is my schematic and schematic netlist.




```
.SUBCKT nand   VDD  GND  VIN1  VIN2  VOUT
M_M6           GND VIN2 N15943 GND MbreakN   L=0.18um   W=1.08um
M_M3           N15943 VIN1 VOUT GND MbreakN   L=0.18um   W=1.08um
M_M5           VDD VIN1 VOUT VDD MbreakP   L=0.18um   W=1.62um
M_M4           VDD VIN2 VOUT VDD MbreakP   L=0.18um   W=1.62um
.ENDS
```

missing! not required?

Does it mean the LVS of klayout would identify the mos that S and D are opposite in pair?

Like below. M5 is from the schematic netlist S2 is from the layout netlist

correct

MBREA_M5
- S VDD (4)
- D VOUT (3)
- G VIN1 (2)
- B VDD (4)

correct

Mbreal $2
- S vout (4)
- D vdd (5)
- G vin1 (3)
- B vdd (5)

incorrect

Could you please explain more about fully parallel?

Thanks!

WENSHIH

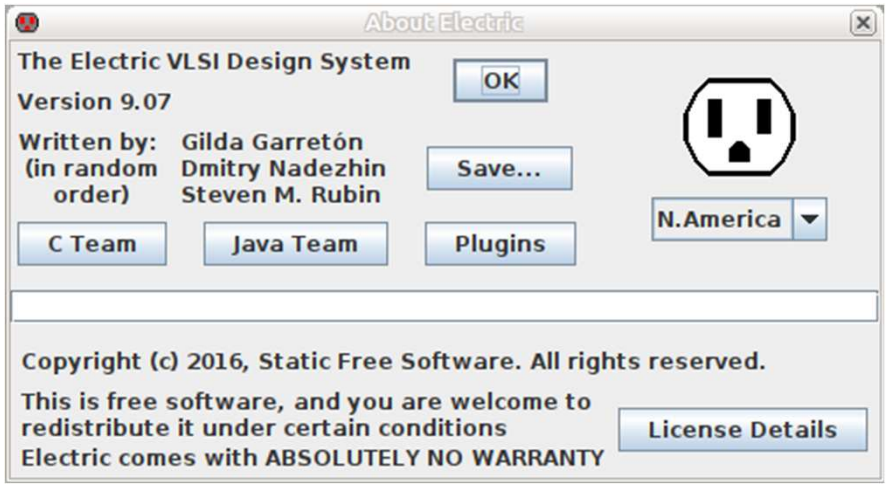
4. Another Sample of Incorrect SPICE Net

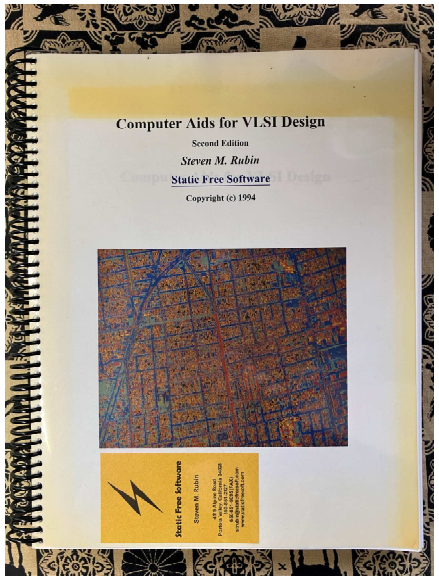
- Correct (expected) SPICE net to be extracted

```
.SUBCKT nand VDD GND VIN1 VIN2 VOUT
M_M6    N15943  VIN2 GND     GND   MbreakN  L=0.18um  W=1.08um
M_M3    VOUT    VIN1 N15943  GND   MbreakN  L=0.18um  W=1.08um
M_M5    VOUT    VIN1 VDD     VDD   MbreakP  L=0.18um  W=1.62um
M_M4    VOUT    VIN2 VDD     VDD   MbreakP  L=0.18um  W=1.62um
.ENDS
```

Extracting a correct SPICE deck is a duty of a schematic editor, not of KLayout!

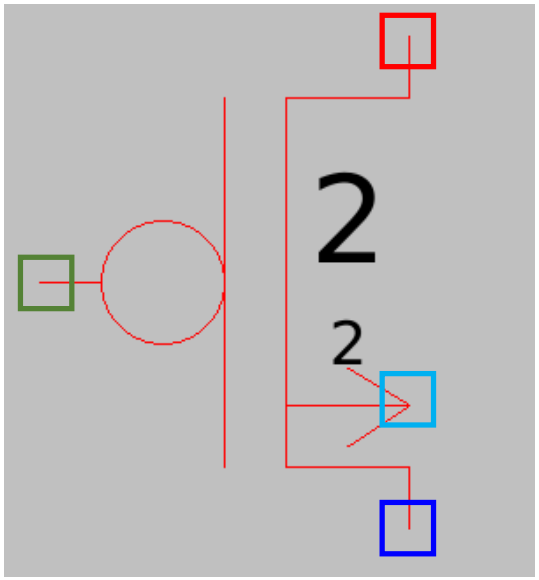
- Let's try the **Electric** CAD (developed by the legendary Steven M. Rubin) after many many many years!

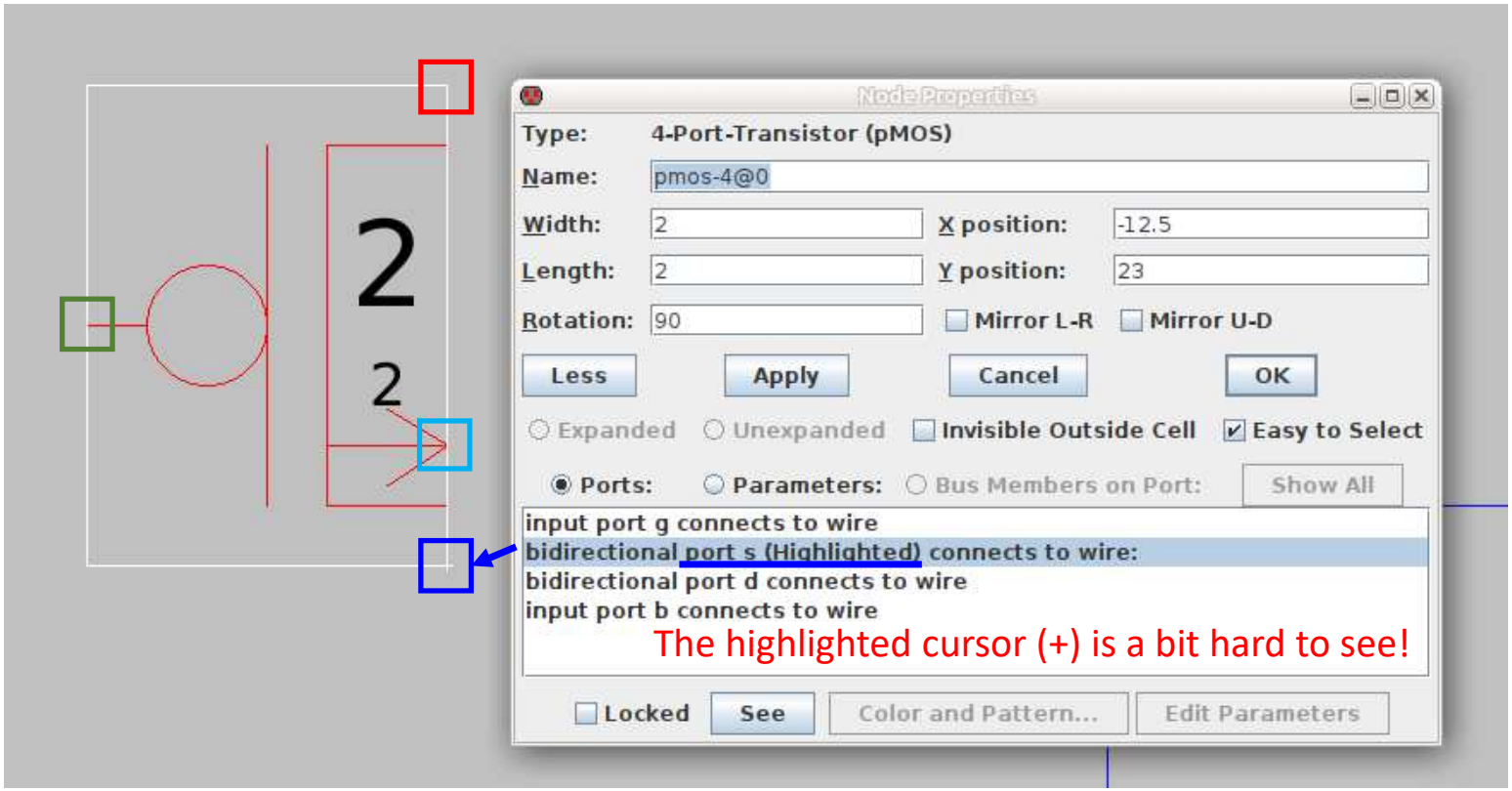




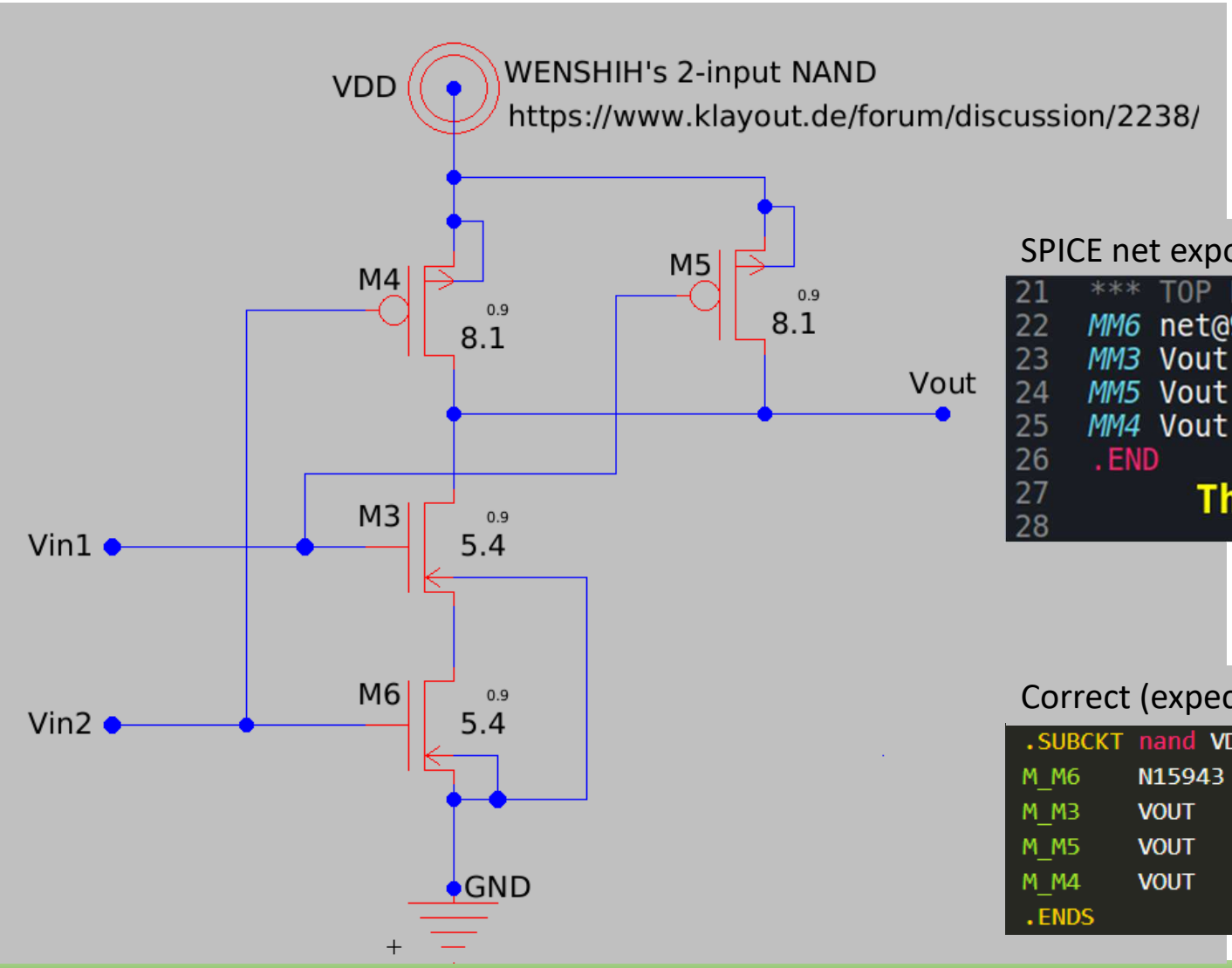
4. Another Sample of Incorrect SPICE Net

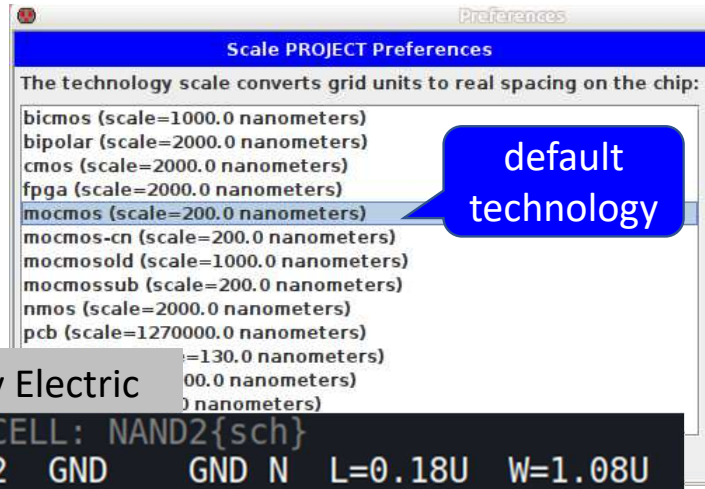
- Schematic symbol of a PMOS and its four terminals (ports) in ***Electric***
- The source is closer to the bulk ; the symbol is asymmetric to avoid misconnection.





4. Another Sample of Incorrect SPICE Net





SPICE net exported by Electric

```
21  *** TOP LEVEL CELL: NAND2{sch}
22  MM6 net@94 Vin2  GND    GND N  L=0.18U  W=1.08U
23  MM3 Vout   Vin1  net@94 GND N  L=0.18U  W=1.08U
24  MM5 Vout   Vin1  VDD    VDD P  L=0.18U  W=1.62U
25  MM4 Vout   Vin2  VDD    VDD P  L=0.18U  W=1.62U
26  .END
27        The row order has been changed.
28
```

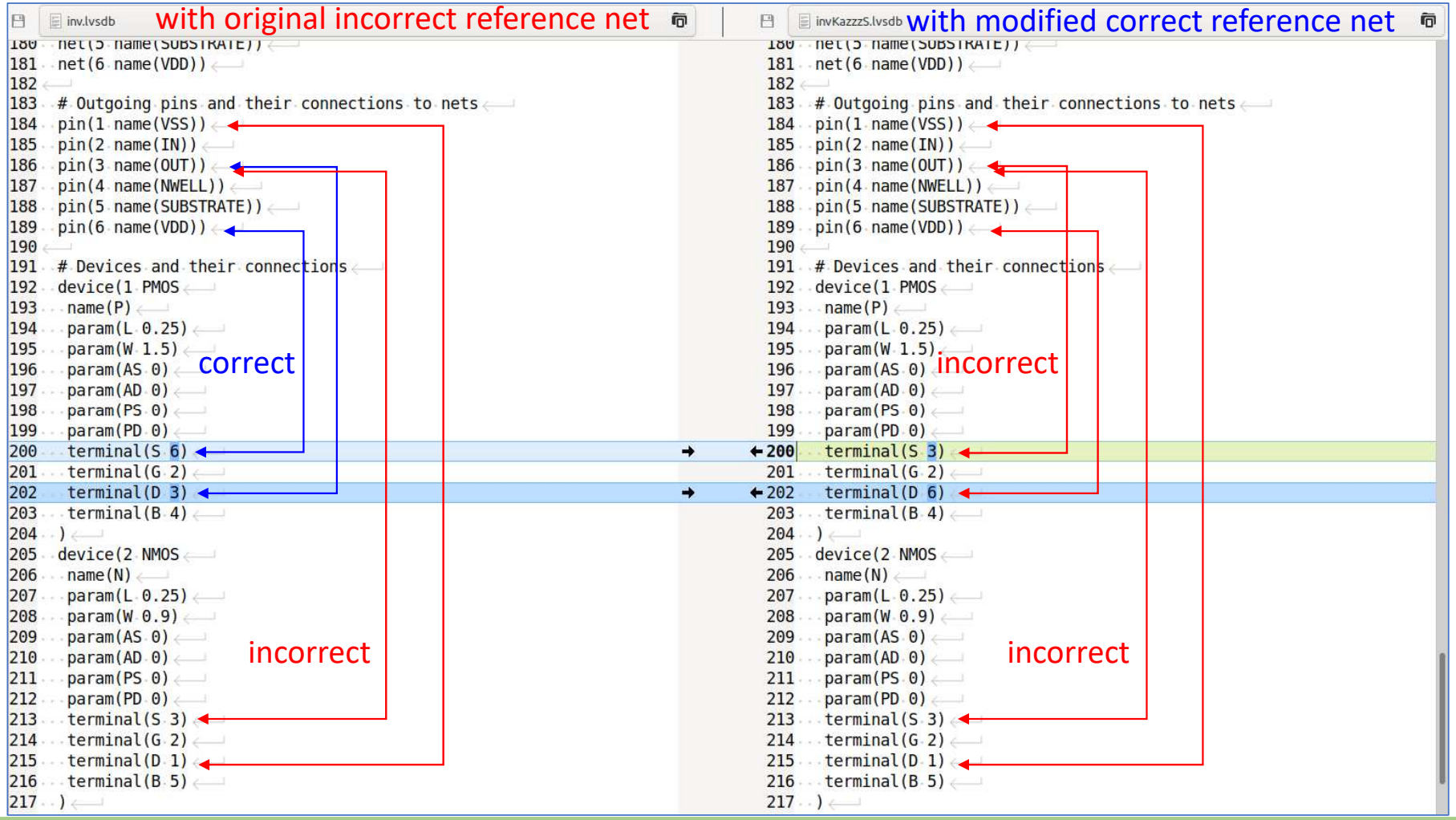
equivalent

Correct (expected) SPICE net

```
.SUBCKT nand VDD GND VIN1 VIN2 VOUT
M_M6    N15943  VIN2 GND     GND   MbreakN  L=0.18um  W=1.08um
M_M3    VOUT    VIN1 N15943  GND   MbreakN  L=0.18um  W=1.08um
M_M5    VOUT    VIN1 VDD     VDD   MbreakP  L=0.18um  W=1.62um
M_M4    VOUT    VIN2 VDD     VDD   MbreakP  L=0.18um  W=1.62um
.ENDS
```

5. Incorrect Device Terminal Connections

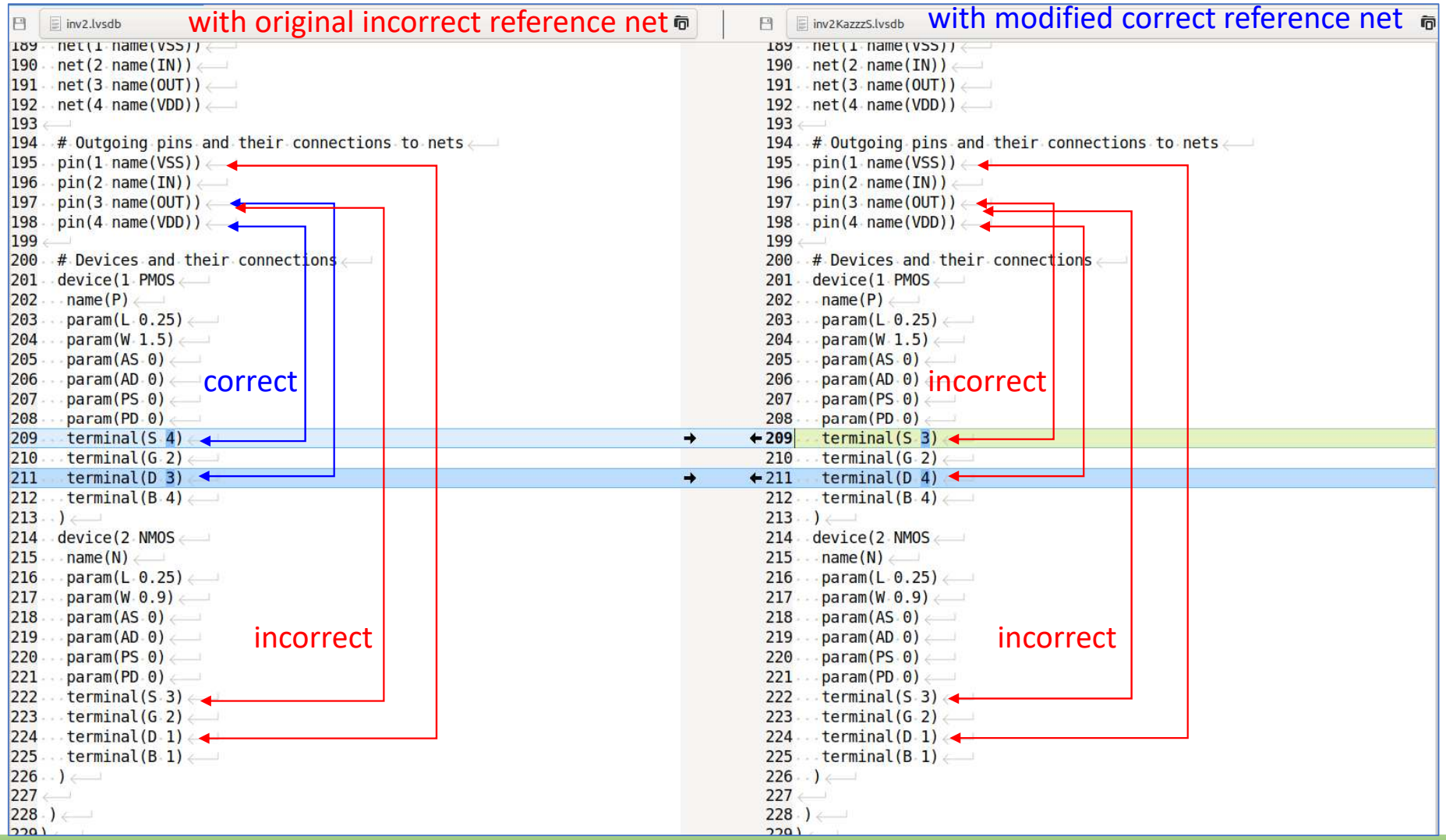
<3> The device connections are more confusing because…



to be cont.

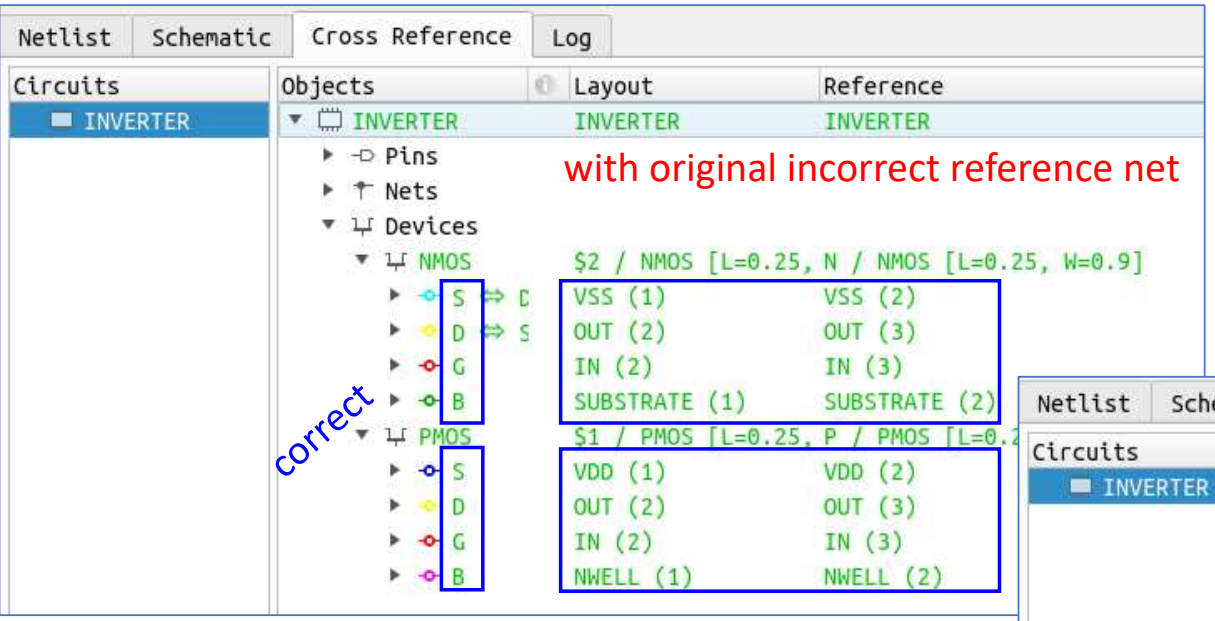
5. Incorrect Device Terminal Connections

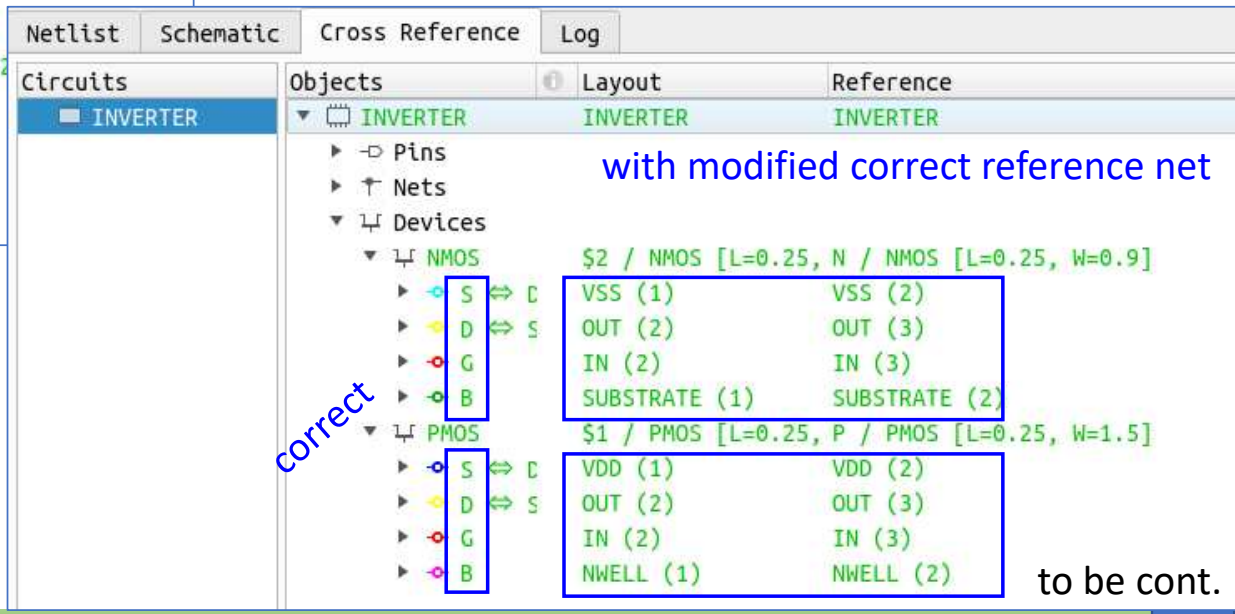
cont.



6. Correct Cross-References

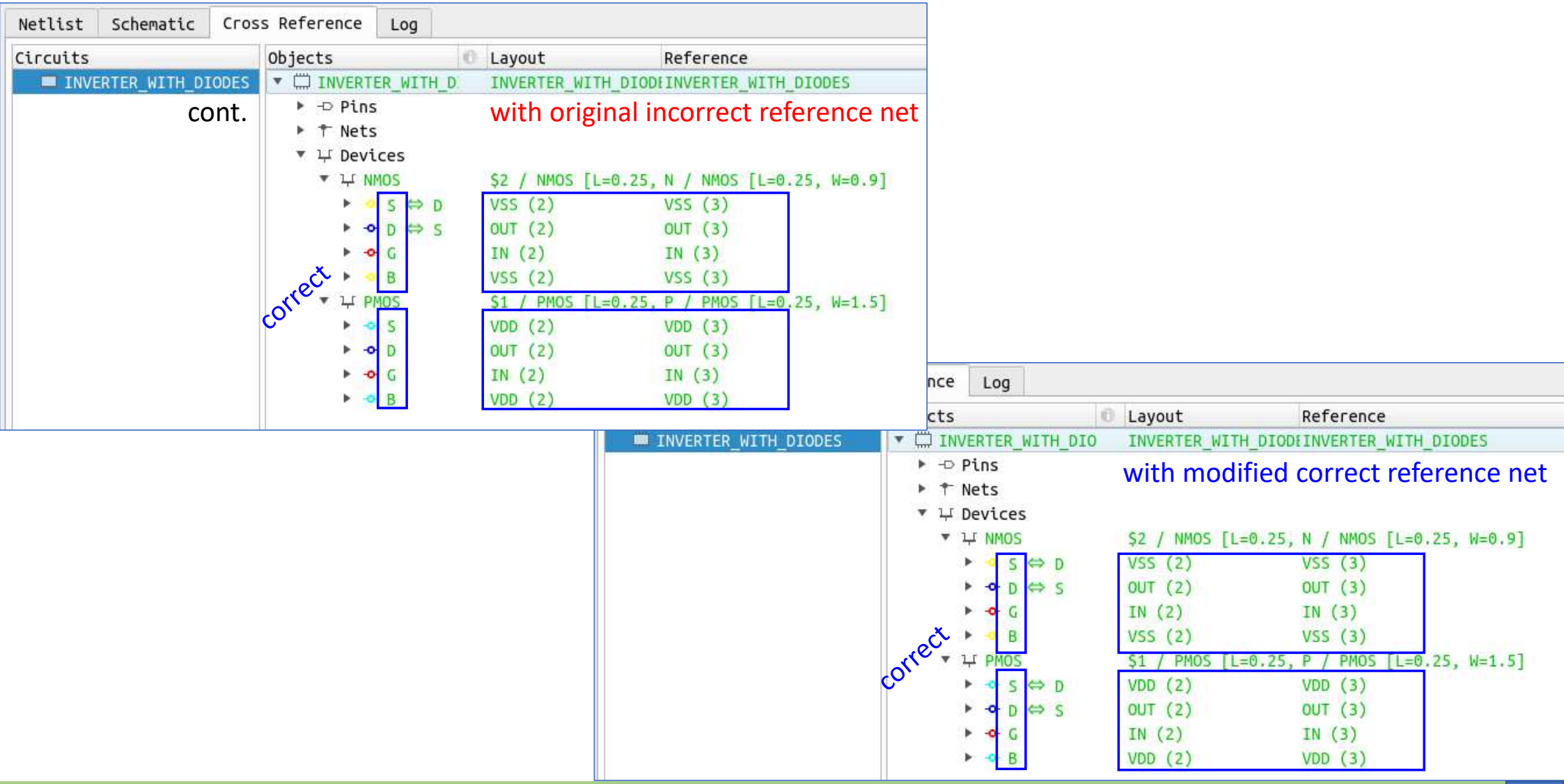
<4> The cross-references are much more confusing because all seem **correct** **eventually** but are inconsistent with the other (intermediate) results. **Where are my misunderstandings?**





to be cont.

6. Correct Cross-References



7. Netlist Database Browser

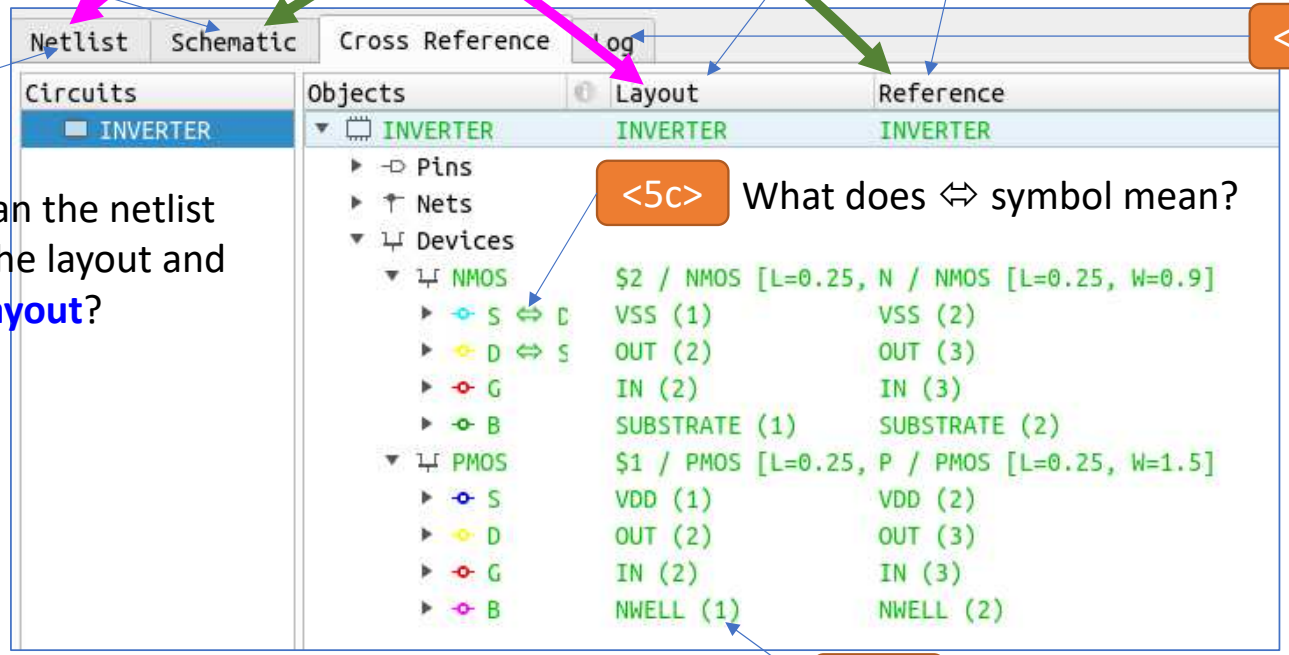
<5> Is there any detailed document interpreting each item that could appear in the netlist database browser? I've tried to find it but couldn't so far. Some items are very intuitive. However, others are not.

<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**

<5e>



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

<5c> What does ⇔ symbol mean?

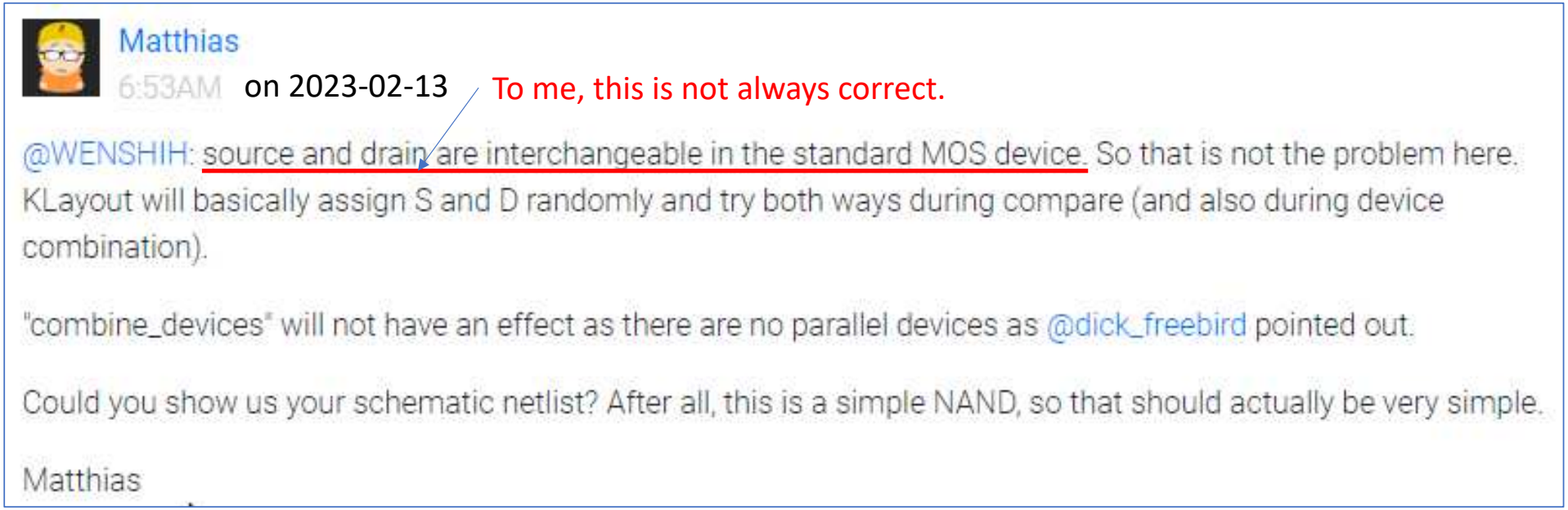
Log is empty in this case. However, there would be something to point out such as the incorrect reference SPICE net.

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

Part-II: Digesting the Discussions

8. Interchangeability of Source and Drain in the Standard MOS Device

- I believe part of my confusion stems from my understanding of **source and drain interchangeability**.



Matthias
6:53AM on 2023-02-13

To me, this is not always correct.

@WENSHIH: source and drain are interchangeable in the standard MOS device. So that is not the problem here. KLayout will basically assign S and D randomly and try both ways during compare (and also during device combination).

"combine_devices" will not have an effect as there are no parallel devices as @dick_freebird pointed out.

Could you show us your schematic netlist? After all, this is a simple NAND, so that should actually be very simple.

Matthias

8. Interchangeability of Source and Drain in the Standard MOS Device

Matthias
2:17AM on 2023-02-20

@sekigawa Thanks for these elaborate slides!

I should comment about the "correctness" here: my viewpoint is that "S" and "D" for a symmetric device should both be read as "S or D" as there is no differentiation.

Sure the schematic is not correct if you tie D to vdd, but S and D are only logical concepts for symmetric devices. You could say that for a PMOS, D is the terminal with the lower and S is the terminal with the higher potential. For NMOS it is the other way round.

I guess that for simulation, the models do not necessarily reflect that symmetry, so for that practical reason it is probably better to assign D and S correctly in the schematic. At least if you want to use this netlist for simulation. For the topology however, the assignment is not binding for symmetric devices.

Lacking any differentiating marker, the layout extractor on the other hand will arbitrarily assign S and D for symmetric devices (like said, both terminals should be regarded "S or D"). It does not take any efforts correcting this "mistake". Instead, the netlist *compare* step regards "S" and "D" as interchangeable, hence the netlists match and all cases.

For DMOS3/4 class devices, the extractor can tell apart S and D and the netlist comparer will no longer regard S and D as interchangeable.

Matthias

This is a very important tool design concept, and I respect it. However, I could not read it from the inverter tutorials.

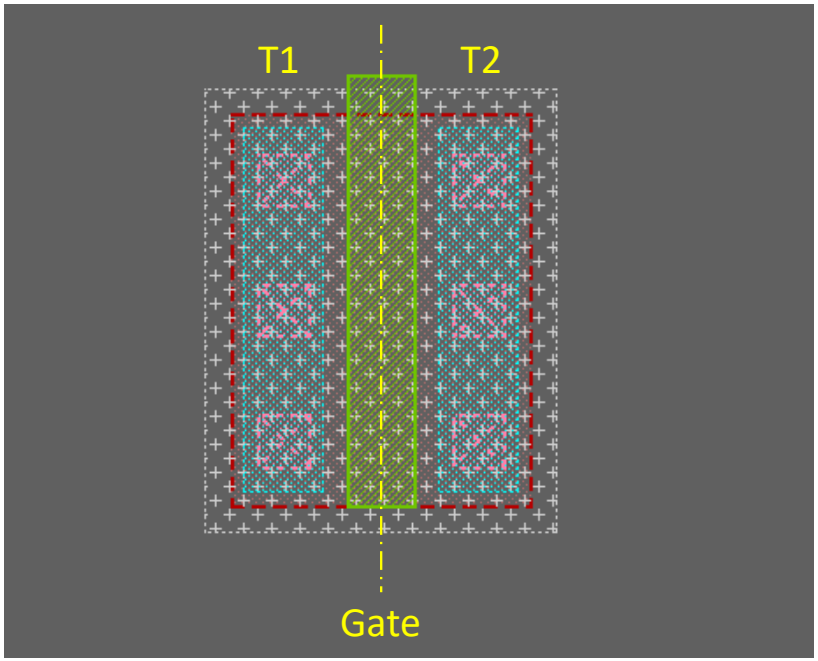
Moreover, influenced by my previous experiences, I have a different view (#20 - #24) on “S and D,” which created a fundamental gap that caused my confusion.

8. Interchangeability of Source and Drain in the Standard MOS Device

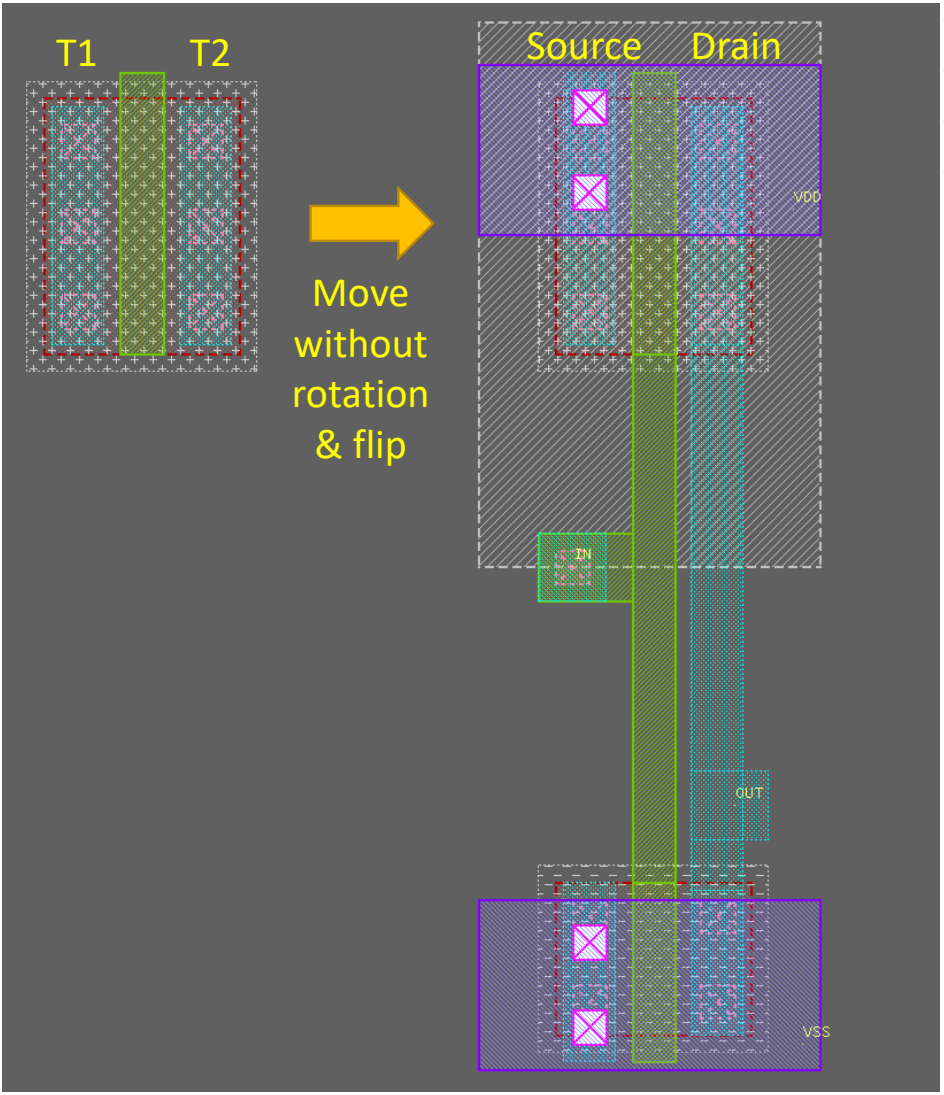
- Below is a layout of a floating PMOS that is geometrically symmetry with respect to the gate poly.
- The terminals T1 and T2 can become the source and drain, or the drain and source, respectively.
- Therefore, we say ***"the source and drain can be interchangeable."***
  - No objection in this stage.
- In other words, however, we cannot identify which is the source and which is the drain.

We agree so far!

I should comment about the "correctness" here: my viewpoint is that "S" and "D" for a symmetric device should both be read as "S or D" as there is no differentiation.



8. Interchangeability of Source and Drain in the Standard MOS Device



- Once the PMOS is used/connected in an Inverter, **ambiguity vanishes**.
- That is, **T1** is forcibly given the generic name of **Source**; **T2**, **Drain**
- Then, Source and Drain are no longer interchangeable in the Inverter.*
- And the SPICE net must be as follows; no other terminal connection can be possible as an Inverter.

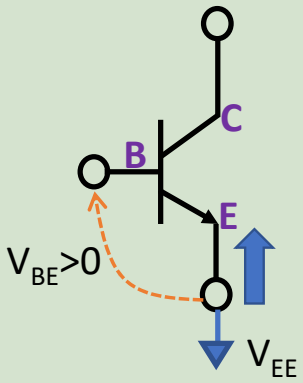
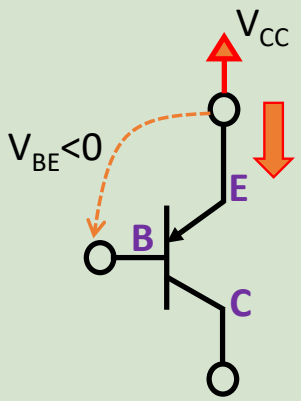
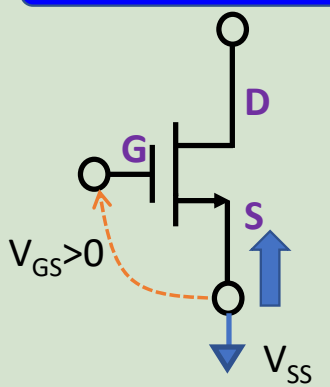
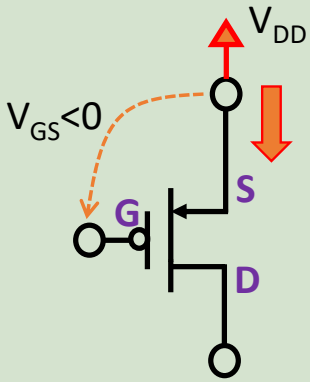
```
1
2   * Simple CMOS inverter circuit
3   *
4   * Original: ${klayout_root}/testdata/lvs/inv.cir
5   *
6   * Modified: invKazzzS.cir
7   *       by: Kazzz-S
8   *     date: 2023-02-07
9   *  reasons: Terminals (drain and source) of the PMOS seem interchanged
10
11  .SUBCKT INVERTER VSS IN OUT NWELL 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
```

Drain Source

SPICE is for simulating the behavior of circuitry.

(*) Source and Drain are the names of geometrical regions in the mask layout, of course. At the same time, more importantly, they imply the physical roles or behaviors in circuitry. Refer to the next slide.

9. My Understandings on Semiconductor Devices

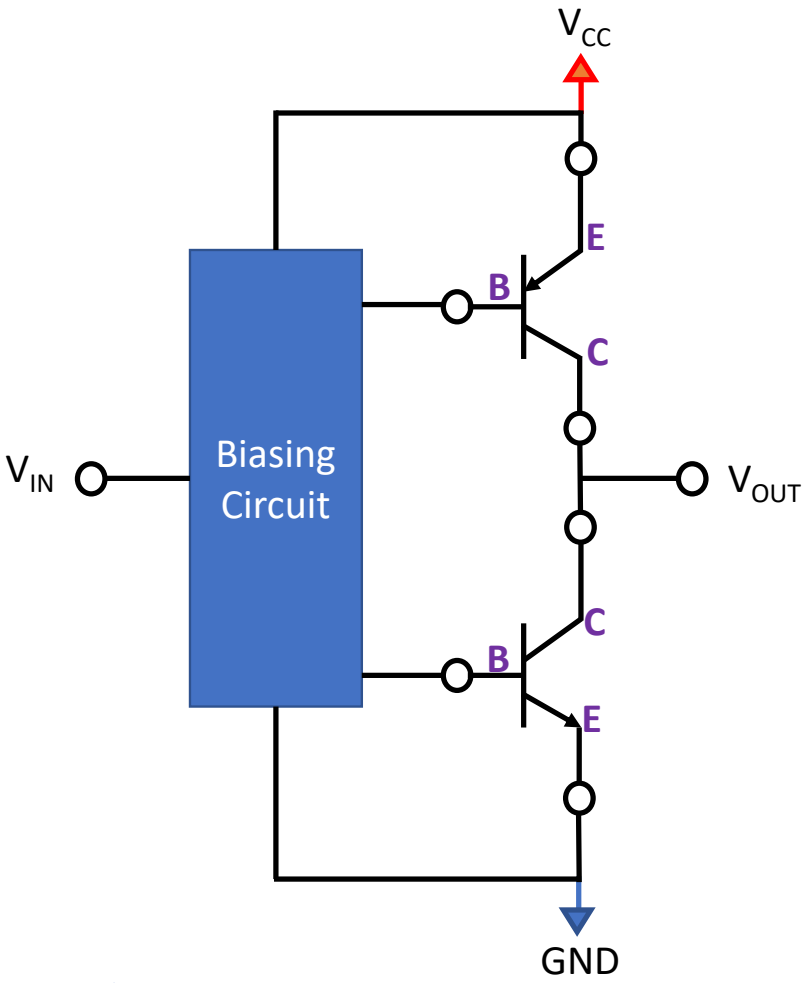
Physics \ Device	NPN BJT	PNP BJT	N-ch MOS Tr	P-ch MOS Tr
Majority Carrier	Electron →	Hole →	Electron →	Hole →
* Majority Carrier Injector (Pitcher)	Emitter	Emitter	Source	Source
* Majority Carrier Controller	Base-Emitter Voltage (V_{BE}>0)	Base-Emitter Voltage (V_{BE}<0)	Gate-Source Voltage (V_{GS}>0)	Gate-Source Voltage (V_{GS}<0)
* Majority Carrier Receiver (Catcher)	Collector	Collector	Drain	Drain
Circuitry				

S and D are not logical but physical concepts

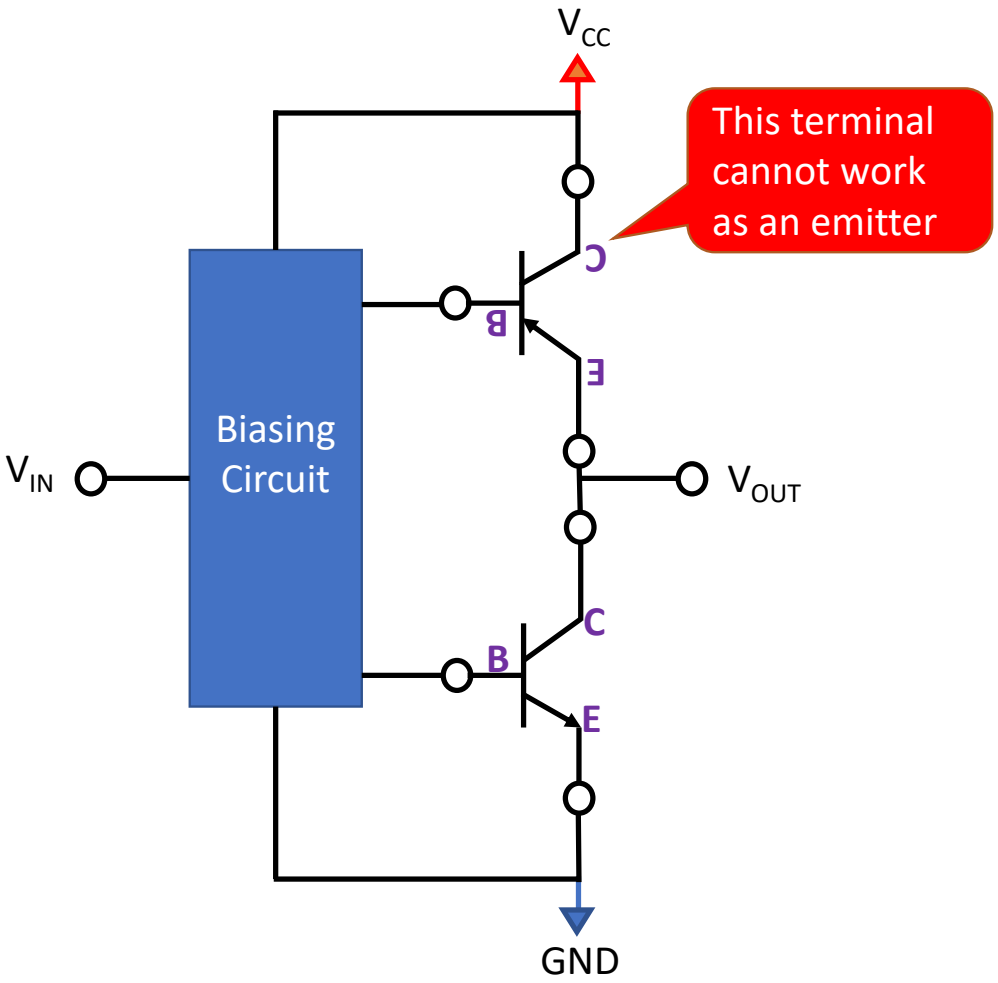
* My terminology

Old aphorism among BJT chip designers: *Emitter current decides everything.*

10. Interchangeability of Emitter and Collector in the BJT Device? No Way!



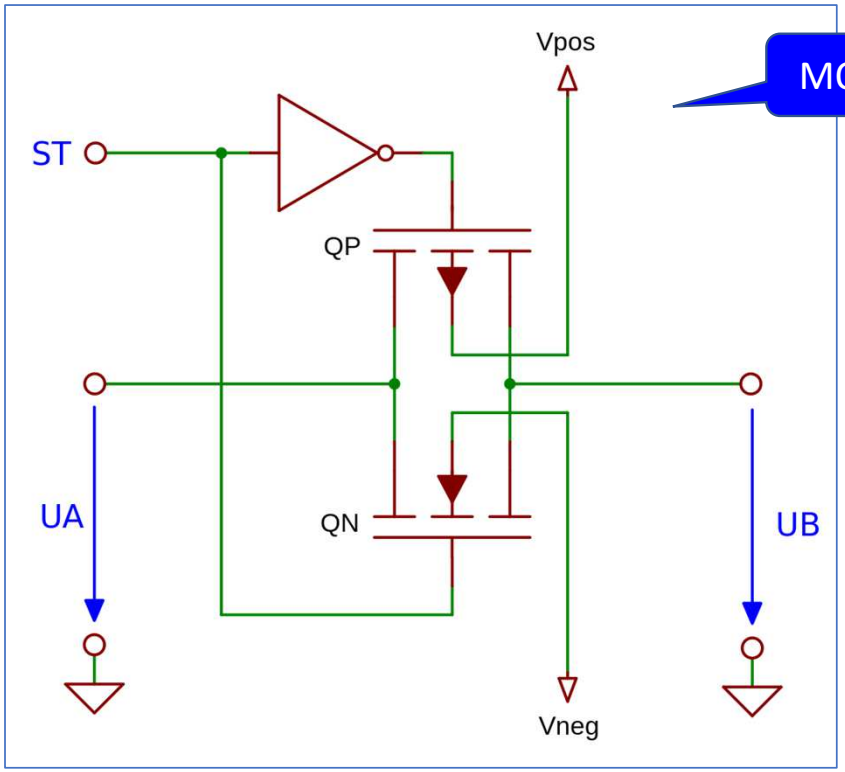
✓ Will work with an appropriate biasing circuit



X Won't work at all
Schematic itself is incorrect

11. Interchangeability of Source and Drain in the Standard MOS Device (again)

<6> In the Transmission Gate (analog switch) configuration, the Source and Drain names (and functions) are swappable. Is present KLayout's LVS designed to support this case?



MOS3 and MOS4 seem for this

Part-III: Experiments

- The original tutorials using **inv.cir** and **inv2.cir** seem not suitable as a “Hello, World!” program for LVS.
- They look simple at first glance but need a deep understanding of the LVS tool design concept.
- That is the pitfall I felt in.
- I prefer to pay an extra price for the overall simplicity and clarity of start something new.

12. Using DMOS4

Matthias
6:30AM on 2023-02-16

@dick_freebird I'm aware of this 🙂

There is another model, called "DMOS3" (no bulk pin) and "DMOS4" (bulk pin) which treats source and drain separately (see https://www.klayout.de/doc-qt5/manual/lvs_device_extractors.html#h2-192). The price to pay is that you need separate recognition layers for source and drain.

The way I implement swapping is that for "MOS3" and "MOS4" I treat the schematic netlist as given (even if drain is on VDD or GND) and try both ways of the extracted device until a match is found. For "DMOS3" and "DMOS4" the latter step is skipped.

answer to
<1>
<6>

@WENSHIH Just sharing screenshots does not make sense. This is a simple case and no device combination is needed - that only applies to fingered devices which are in parallel (B, S, G and D share the same net). I don't see how the mismatch happens (LVS report), but I also don't see the script, I can't check the layers and derviations and I cannot measure the device dimensions as I do not have the layouts. I can't really help in that case.

Matthias

12. Using DMOS4

dick_freebird on 2023-02-20
5:15AM

Yes, they should be but evidently are not, and I am wanting to get to the bottom of "why?".

I figure maybe removing known "not-right" from input netlists might help expose the unknown.

Which thus far, remains unknown.

I agree with you!

Let me ask another question which I have not seen specifically asked: "Which of the MOS extractors are in fact being used?".

It appears to me that this layout is done on a "found PDK" and perhaps somebody set it up with non-swapping extractors by default? If you poke into the devices do you find evidence that D, S are being force-assigned (the recognition layers mentioned)?

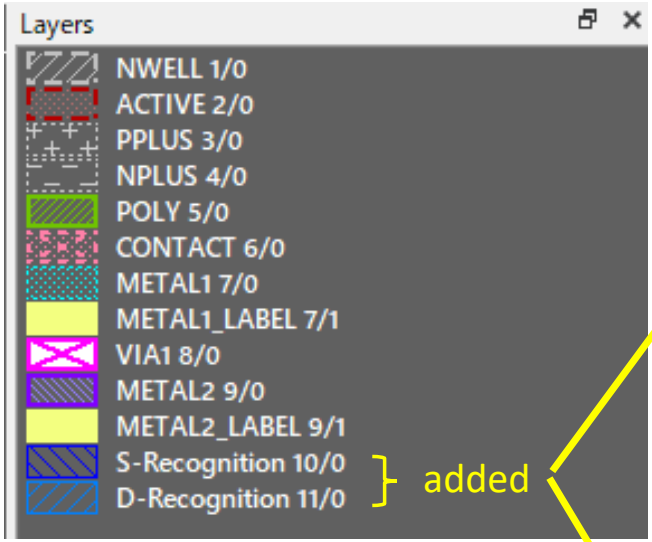
Corrolary activity: If MOS3 / MOS4 -are- being used, how about switching to DMOS3/4 and making the S/D symbol connections "correct"? Does that fix anything? Then if so, why?

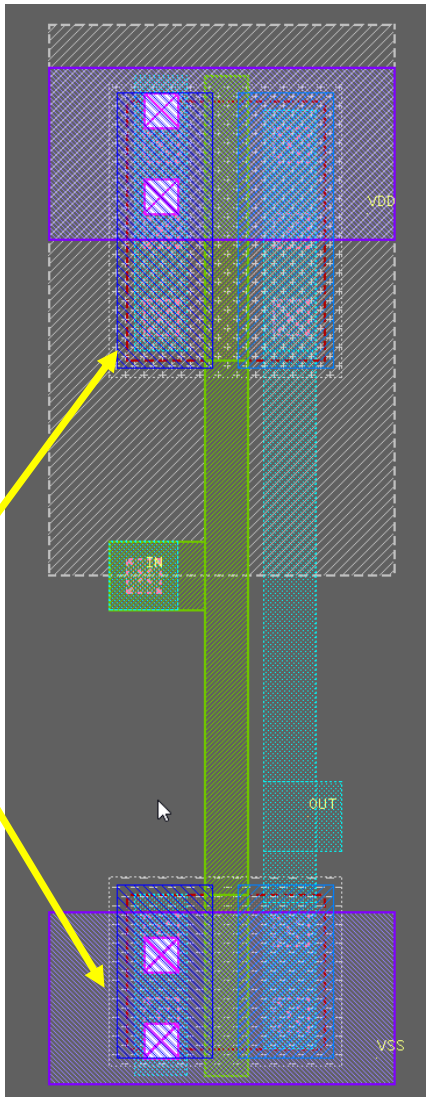
I have already attempted this activity.
Please go through the succeeding slides.

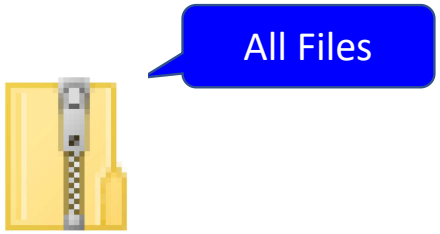
I think that showing the extract deck and the layer objects of the NMOS and PMOS elements might offer clues. If there is any sort of report file generated during layout extraction, maybe that too?

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

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







Inv-DMOS4.zip

13. 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

```
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

D and S are swapped

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

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

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)
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
```

$choice=4

I expect this should match!

13. 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 c
        puts "       then, execute the
        return
    end
  end
end


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

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

deep

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

13. 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	
INVERTER	

Objects		Layout	Reference
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)

Configure | Probe Net | Lock

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

13. 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

Only this matched!

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)	VDD (2)
	D	OUT (2)	OUT (3)
	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/Inv-DMOS4/inv-DMOS4.oas'
          Top cell name = 'INVERTER'
        SPICE deck file = 'invX1.cir'
```

13. 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	
INVERTER	

Objects	Layout	Reference
INVERTER	INVERTER	INVERTER
Pins		
Nets		
Devices		
NMOS	$2 / NMOS [L=0.25,	N / NMOS [L=0.25, W=0.9]
D ⇔ S	OUT (2)	VSS (3)
S ⇔ D	VSS (1)	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)	VSS (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/Inv-DMOS4/inv-DMOS4.oas'
          Top cell name = 'INVERTER'
        SPICE deck file = 'invX2.cir'
```

13. 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)
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
```

$choice=4

I expected this should match. But failed!

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 ⇔ 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=4
4
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002/Inv-DMOS4/inv-DMOS4.oas'
          Top cell name = 'INVERTER'
        SPICE deck file = 'inv-DMOS4.cir'
```

14. 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.

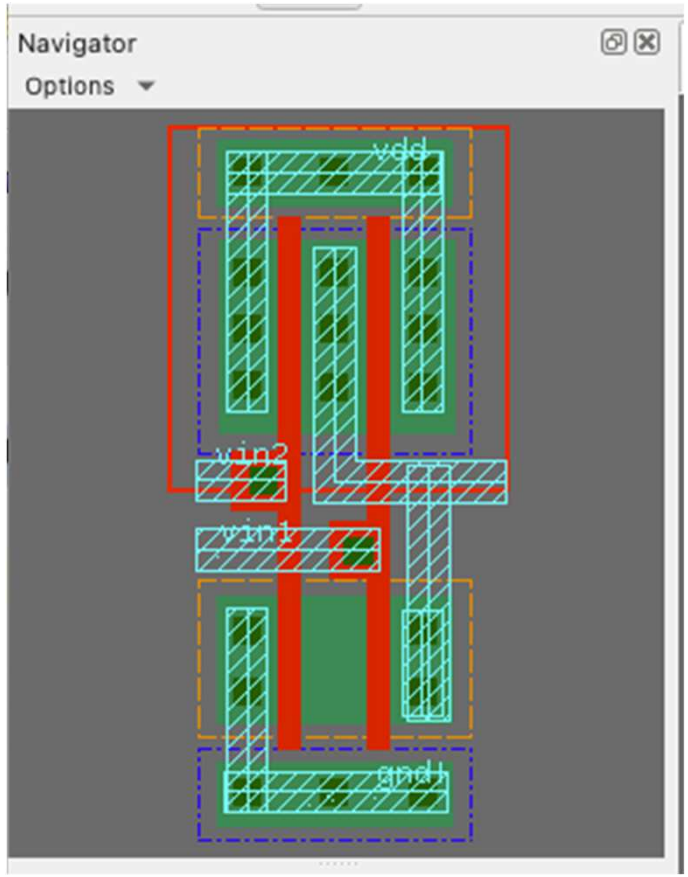
nand.test.zip
14.7K

⬅ Starting point. Thank you for providing this!

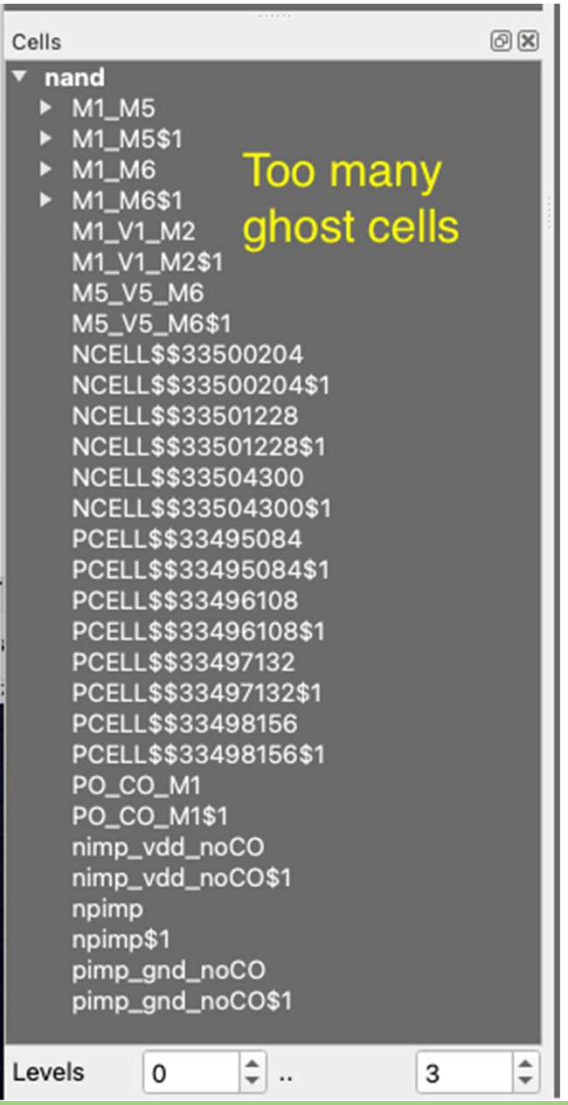
1. Check the provided resource files and modify them if necessary.

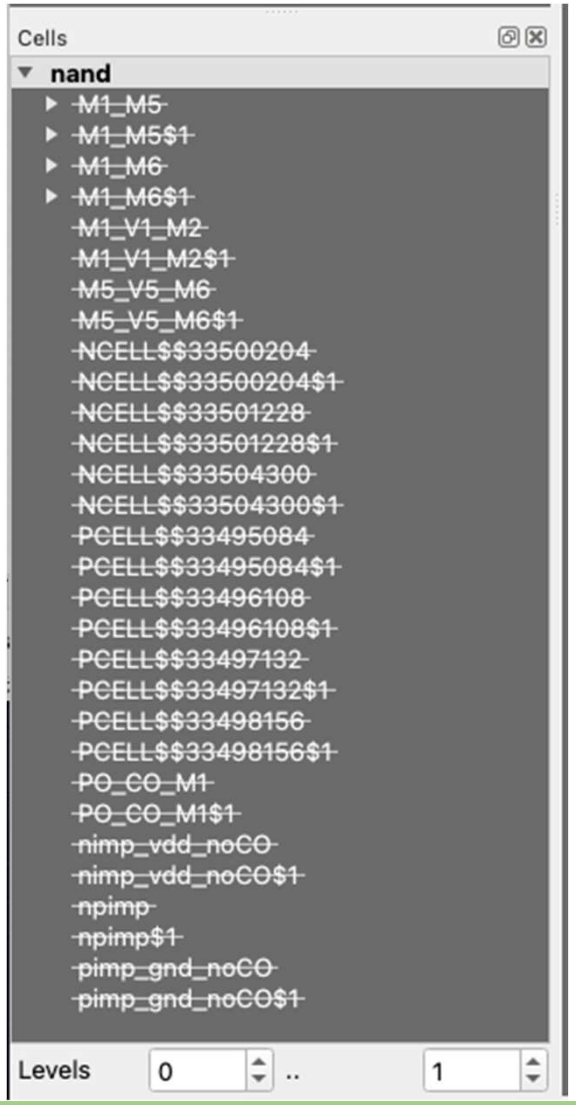
All Files

NAND-DMOS4.zip

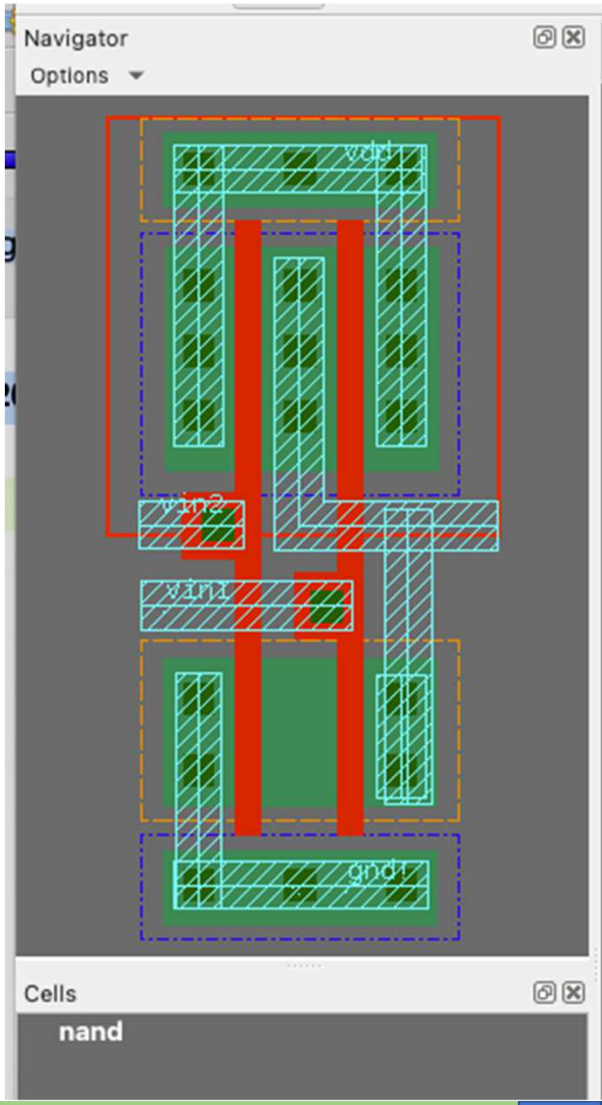


14. Using DMOS4 for a **2-input NAND**



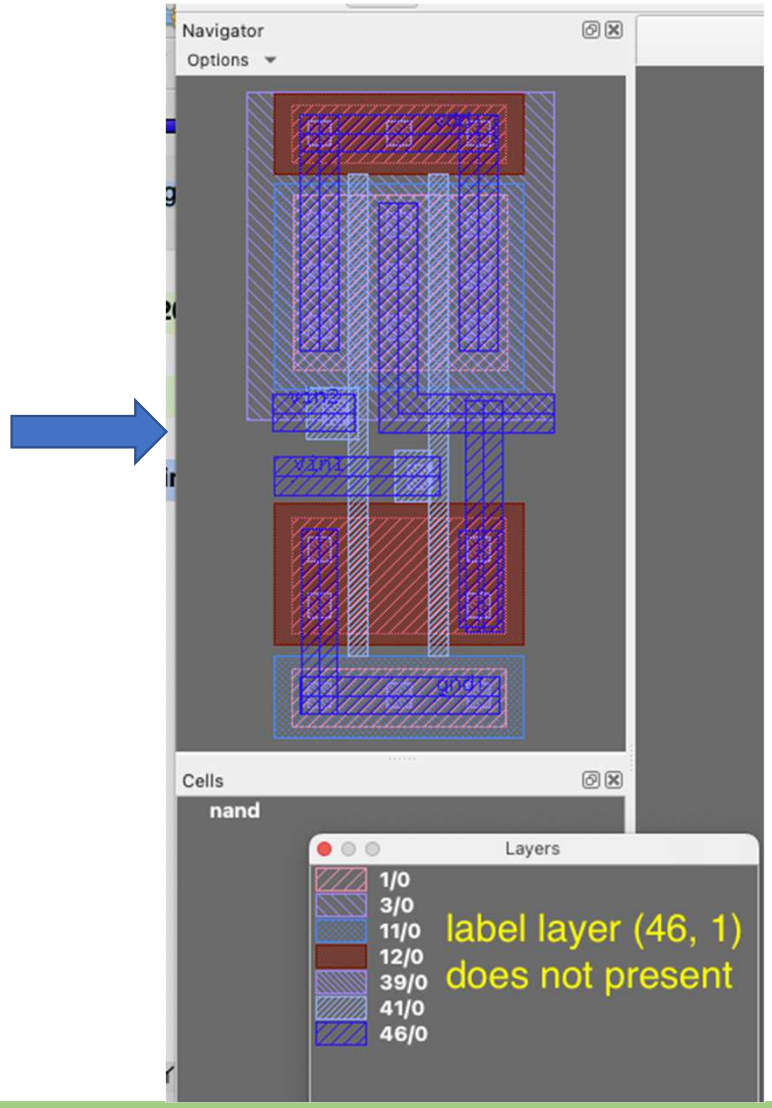


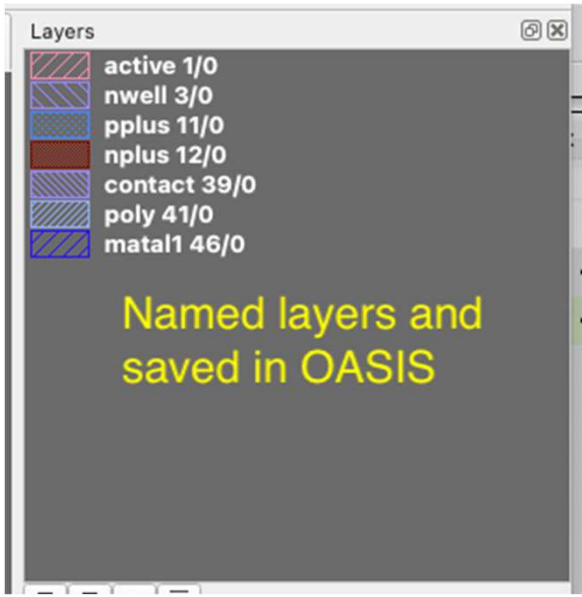
I need only the top cell.



14. 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
```





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

2. Prepare two SPICE deck files.

nand.cir

terminals are missing

```
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
```

original SPICE deck

$choice=1 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
14 .ENDS
```

D and S are swapped

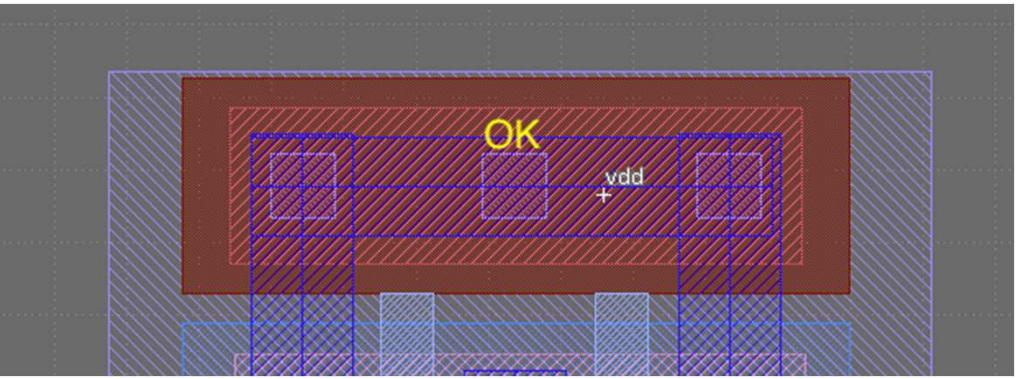
$choice=4 nand.cir

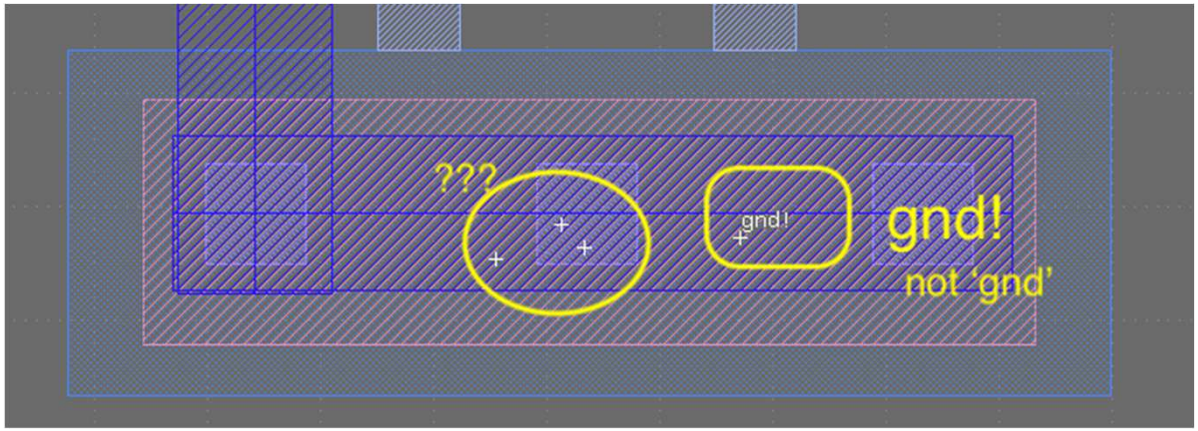
```
1  * https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
2  *
3  * Modified the original SPICE net
4  *   1) there is no tie-down diodes in the original layout
5  *   2) added terminals [VDD GND IN1 IN2 OUT NWELL SUBSTRATE]
6  *   3) transistor name changed to the default (MbreakN,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

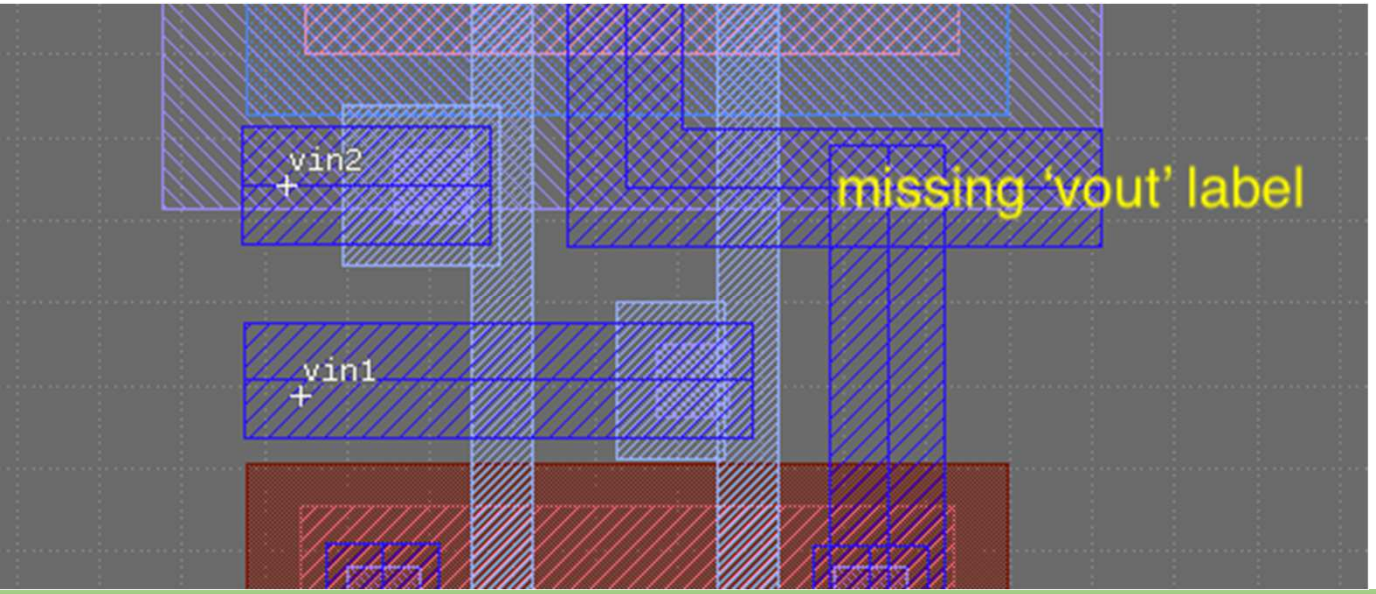
I expect this should match!

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

3. Check the original design in GDS2.





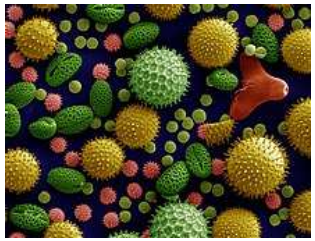


14. Using DMOS4 for a **2-input NAND**

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.

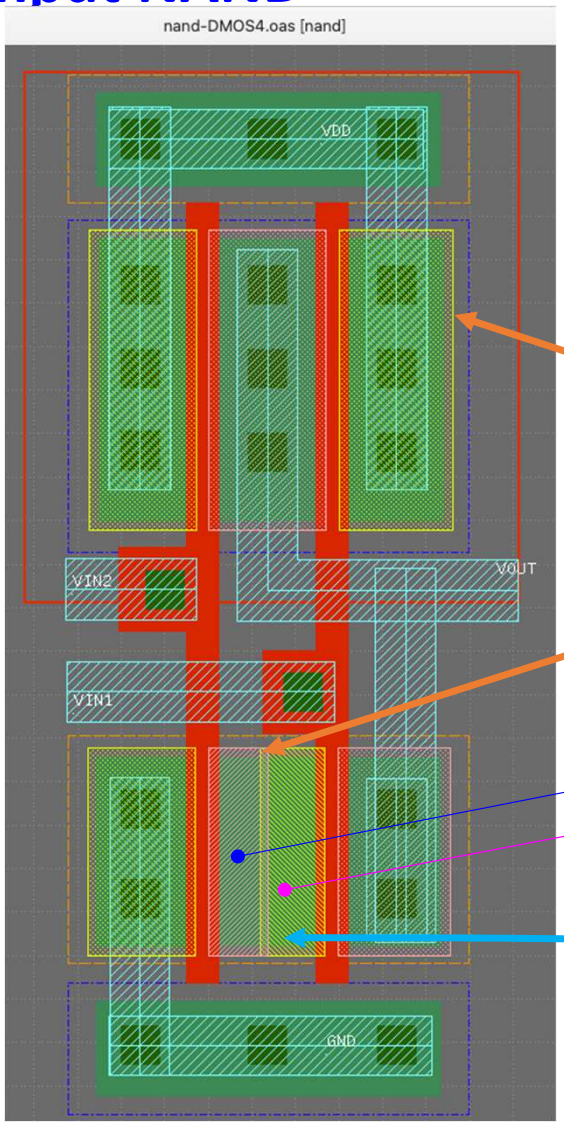
Monoecy - Wikipedia



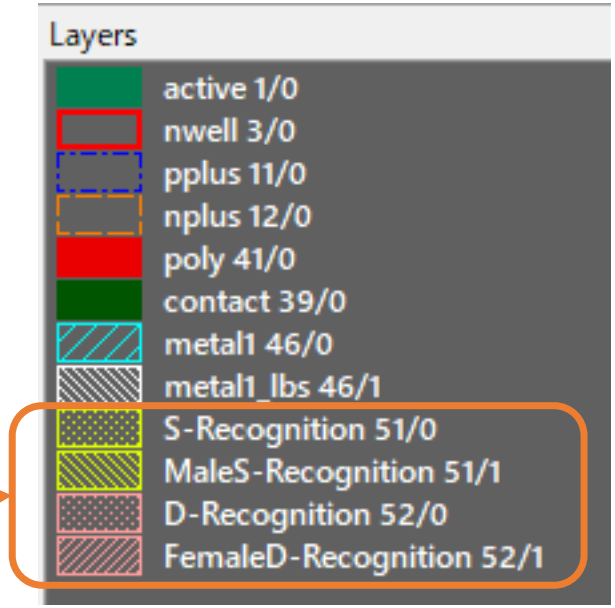
Pollen - Wikipedia

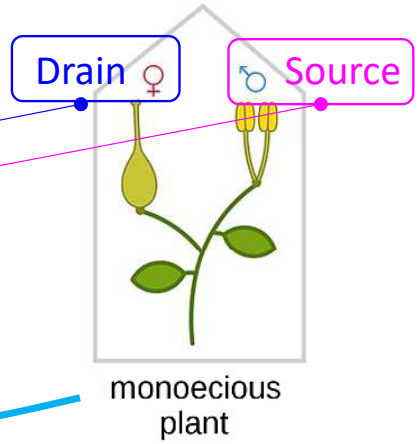
Gate

Pollen can be seen as the majority carrier.



Added some helper layers for experiments





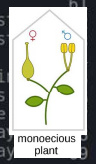
14. Using DMOS4 for a **2-input NAND**

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 "### l
        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].in
  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 lay

# 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
```



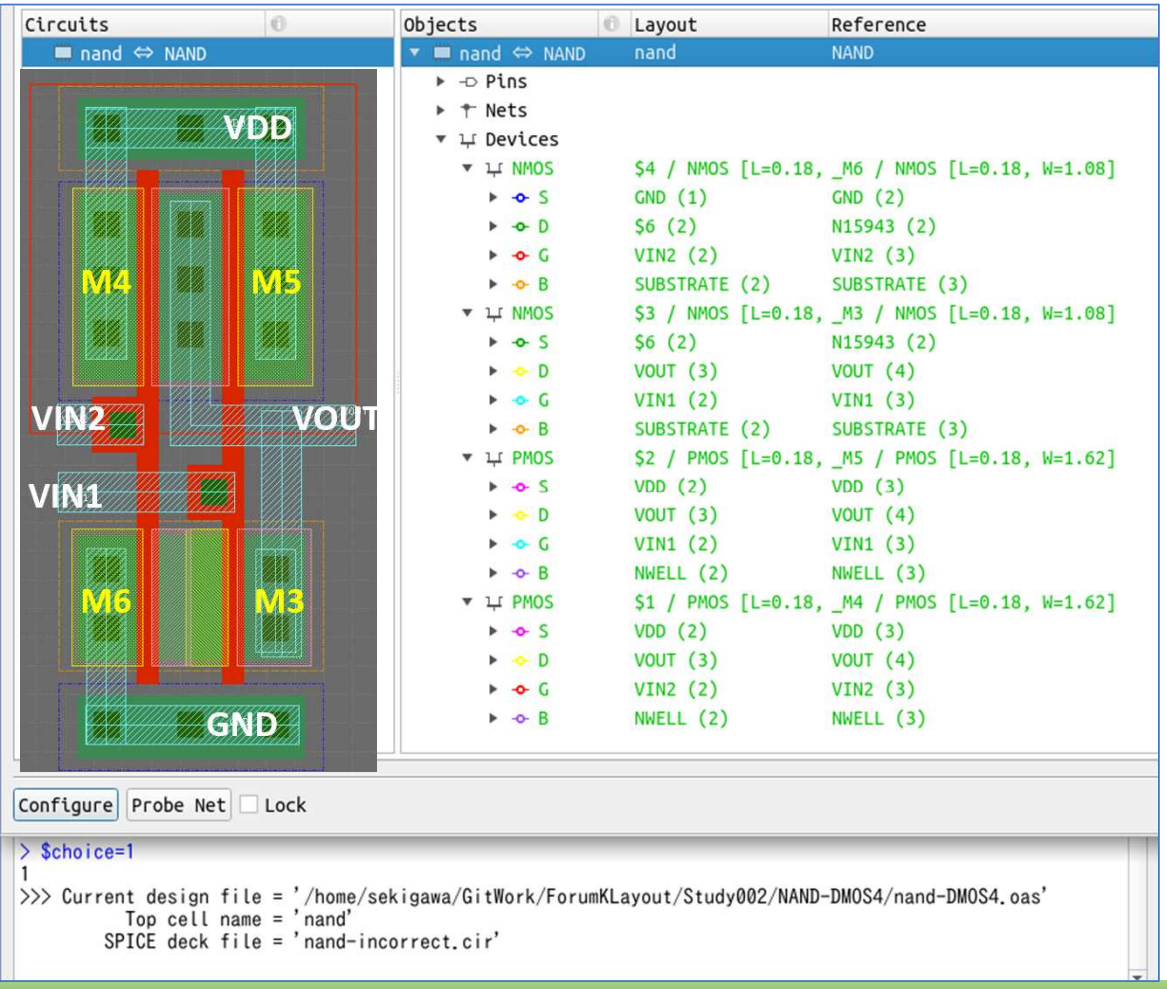

```
nss
ndd
nsmale
ndfemale


#-------------------
# [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,     n

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

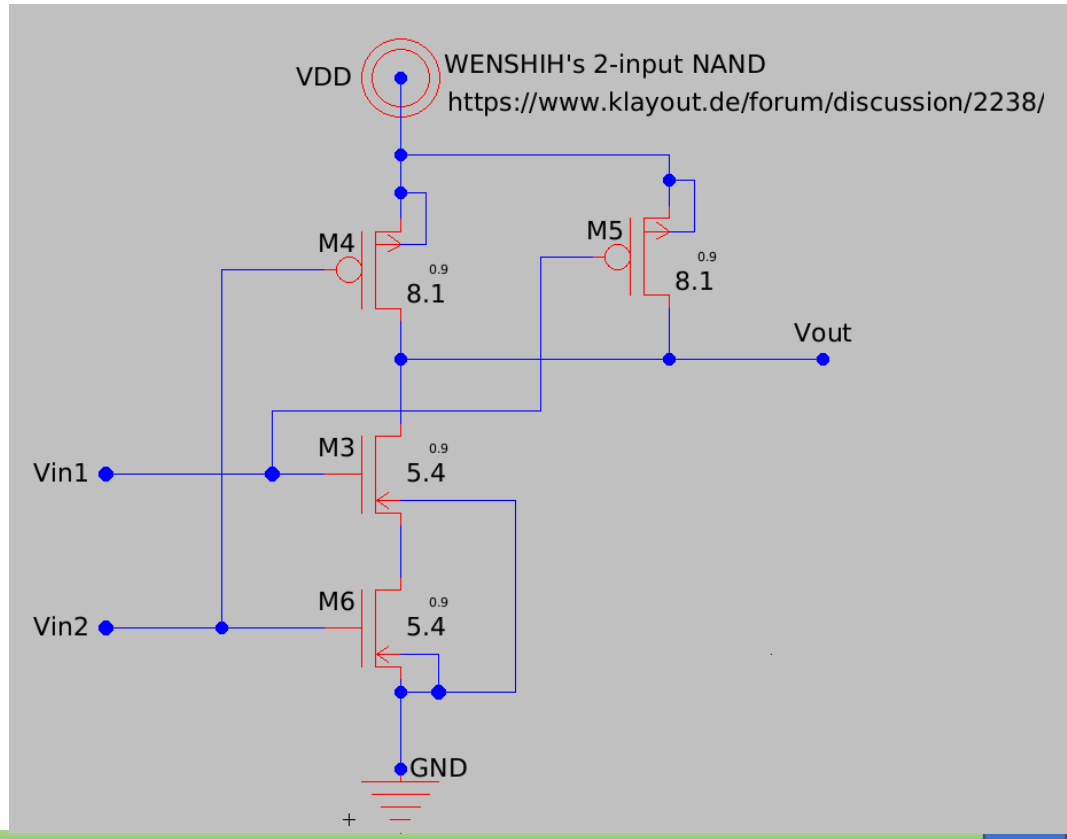
```
#--------
# [7] Com
#--------
schematic


compare


# EOF
```

14. 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                 This matched!
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
```

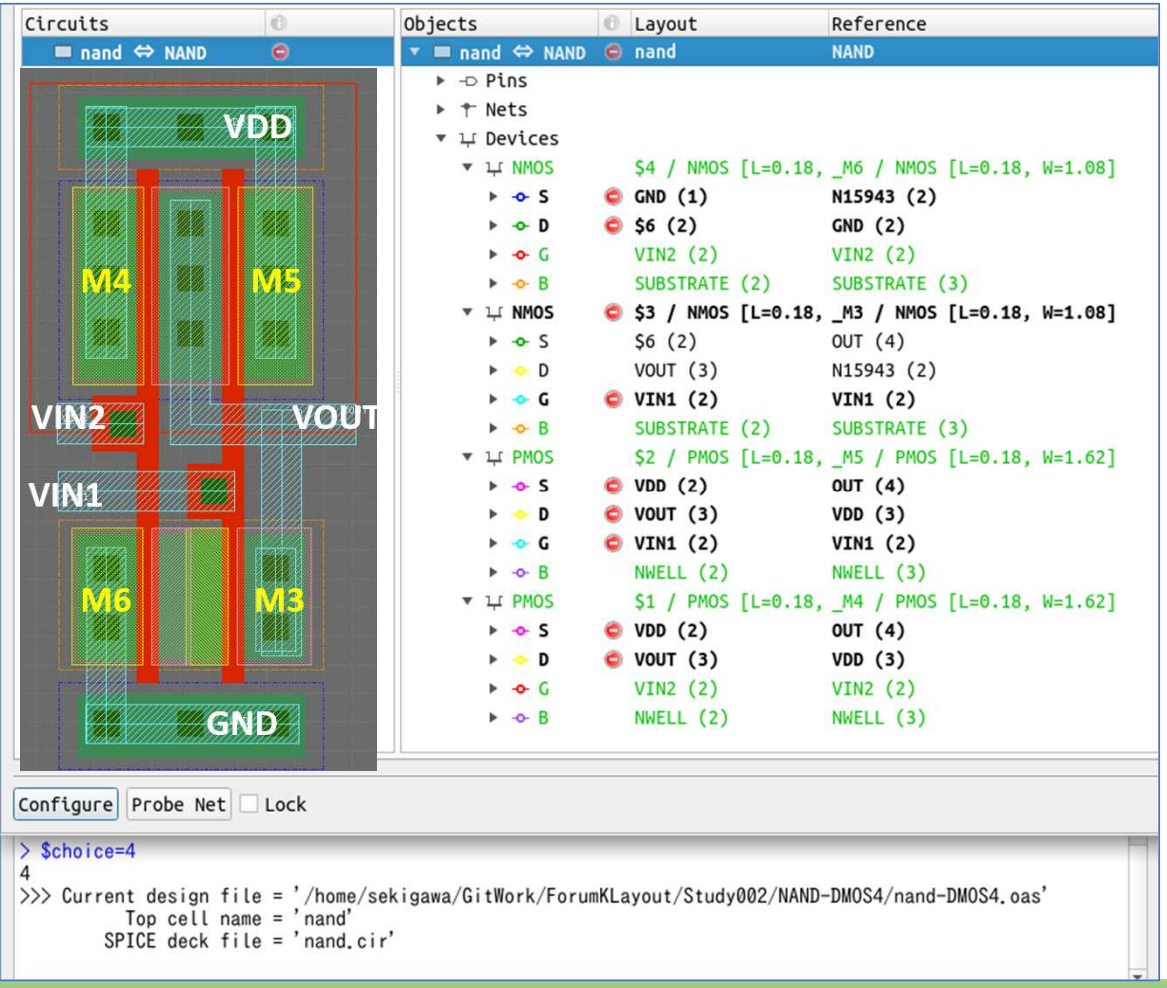




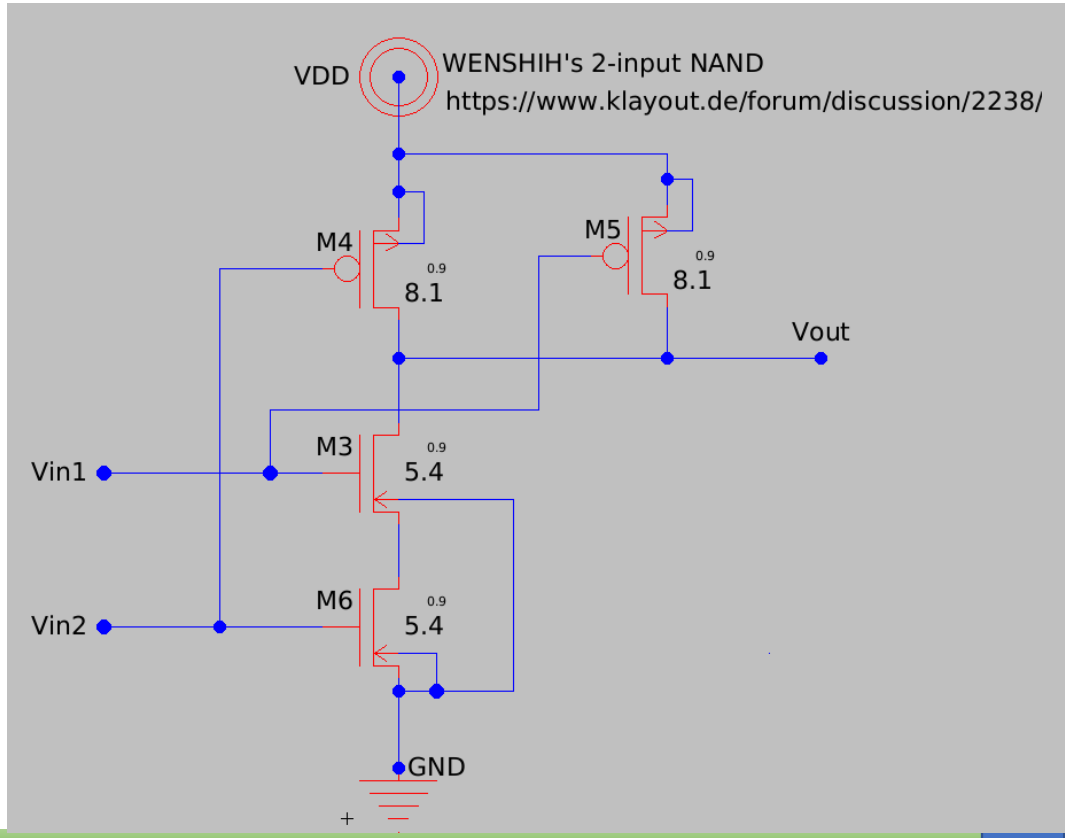
14. Using DMOS4 for a **2-input NAND**

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                                  I expected this should match. But failed!
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
4
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002/NAND-DMOS4/nand-DMOS4.oas'
         Top cell name = 'nand'
        SPICE deck file = 'nand.cir'
```



15. Summary and Intermediate Conclusions as of 2023-02-20

- To use **DMOS4** in LVS, I employed the notion of an extraction helper layer.
  - I'm not sure if inv-DMOS4A.lvs and nand-DMOS4A.lvs are appropriate or not.
  - Any suggestion will be highly appreciated.

- The net extracted from a layout is correct (in the two cases).

- However, a reference net (SPICE deck) seems to be misinterpreted…
  - as if S and D are **always (forcibly) swapped first** even if the intention is…

> The way I implement swapping is that for "MOS3" and "MOS4" I treat the schematic netlist as given (even if drain is on VDD or GND) and try both ways of the extracted device until a match is found. For "DMOS3" and "DMOS4" the latter step is skipped.

Local Parts Storeroom