LVS of CMOS Inverter and 2-input NAND

continuation-2

Aim:

To understand the basics of KLayout's LVS

Disclaimer:

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

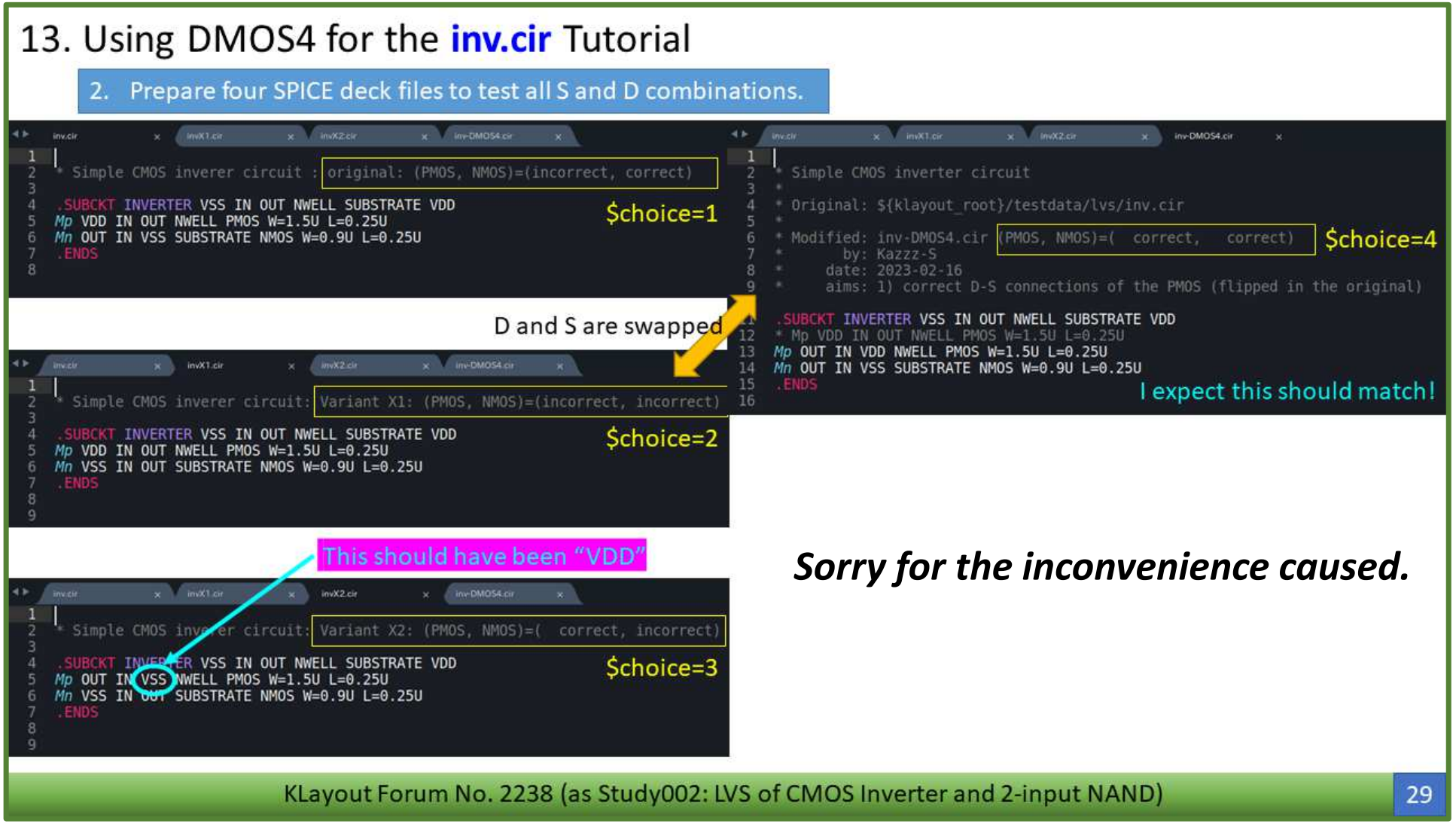
By Kazzz-S (2023-03-08)

con-1 (2023-03-01)

original (2023-02-21)

Part-VI: Errata

In LVS-CMOS-Inverter-NAND.pptx



In LVS-CMOS-Inverter-NAND-Con1.pptx

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

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

inv.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

invX1.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

This should have been "VDD"

invX2.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-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

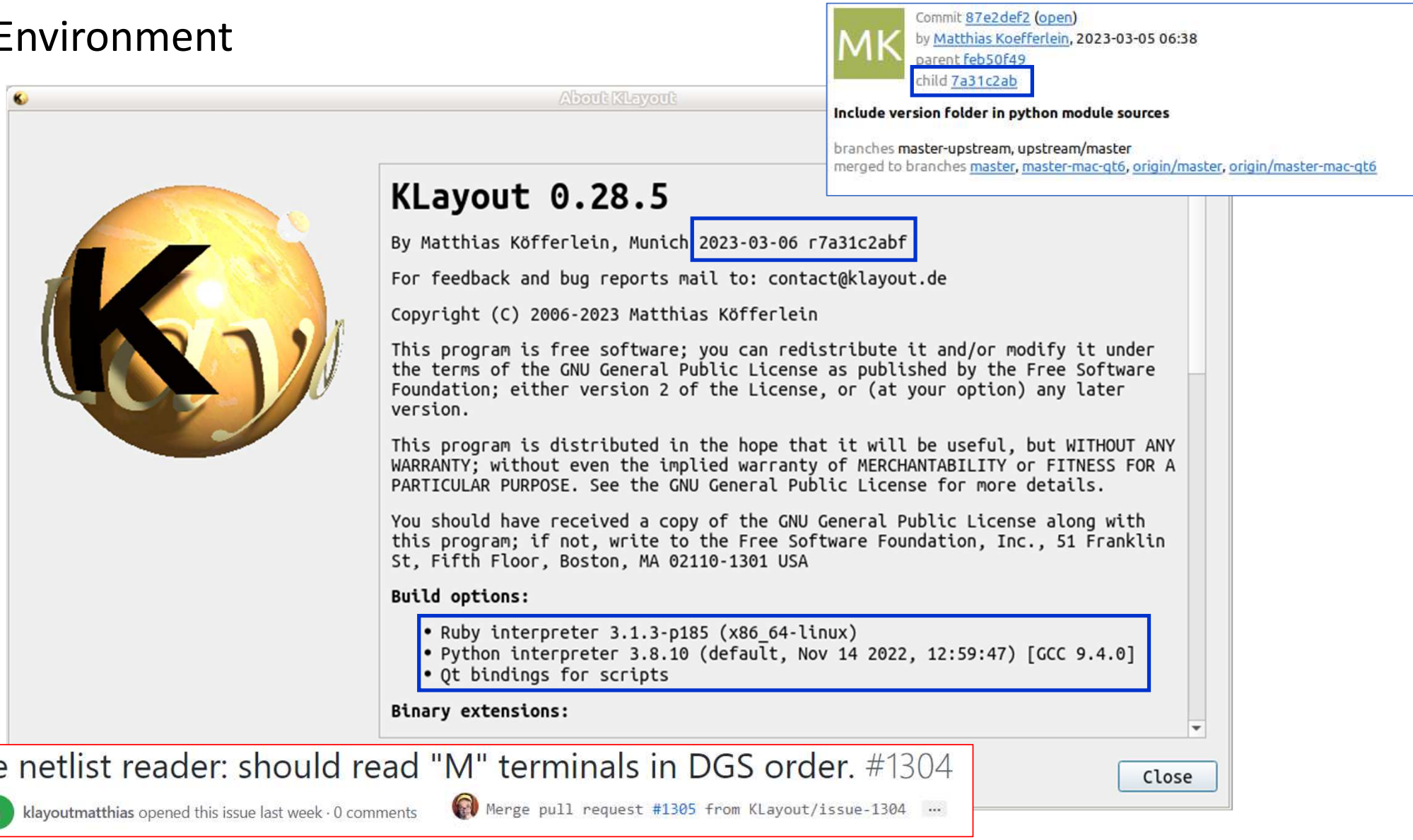
I expect this should match!

Sorry for the inconvenience caused.

KLayout Forum No. 2238 (as Study002: LVS of CMOS Inverter and 2-input NAND) 6

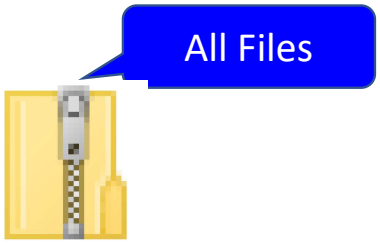
Part-VII: Re-experiments with MOS4

20. Test Environment

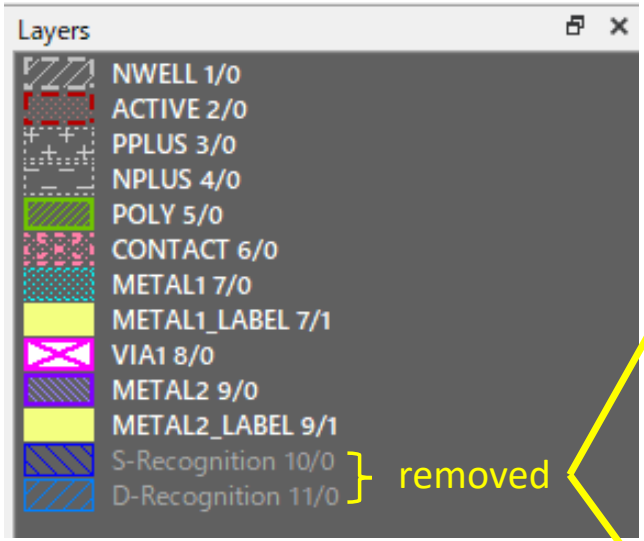


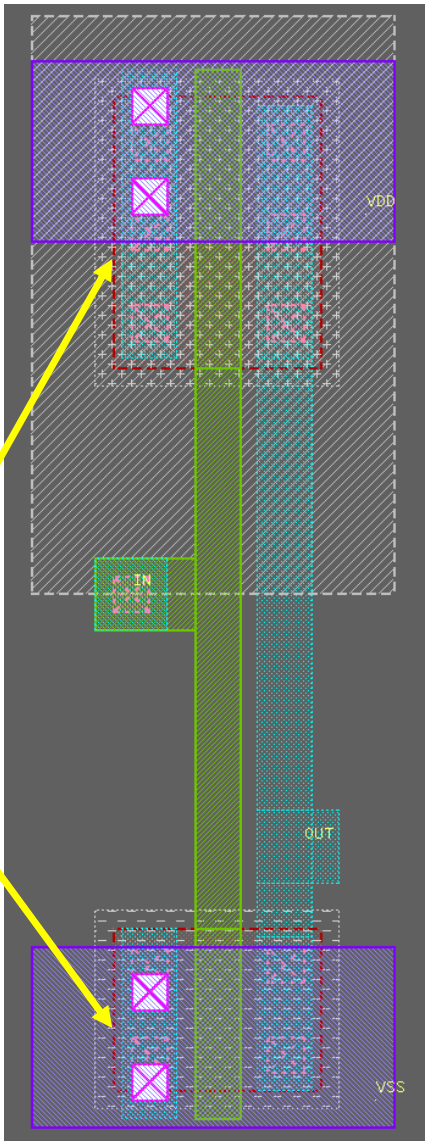
21. Using MOS4 for the **inv.cir** Tutorial

1. Remove the two dummy layers for S- and D-recognition that were added to use the DMOS4 extractor.



Inv.zip





21. Using MOS4 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-correct.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-correct.cir

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

.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-correct.cir

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

.SUBCKT INVERTER VSS IN OUT NWELL SUBSTRATE VDD
Mp OUT IN VDD 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-correct.cir

```
* Simple CMOS inverter circuit
*
* Original: ${klayout_root}/testdata/lvs/inv.cir
*
* Modified: inv-correct.cir (PMOS, NMOS)=(  correct,   correct)
*       by: Kazzz-S
*     date: 2023-03-06
*     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 all four should match by S-D swapping.

21. Using MOS4 for the **inv.cir** Tutorial

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

inv.lvs

```
#----------------------------------------------------------------------------------------------
# Original: ${klayout_root}/testdata/lvs/inv.lvs
#
# Modified: inv.lvs
#       by: Kazzz-S
#     date: 2023-03-06
#     aims: 1) retry the four SPICE deck files, namely ["inv.cir", "invX1.cir", "invX2.cir",
#              "inv-correct.cir"] with the "MOS4" (not "DMOS4") extractor after the issue
#              in https://github.com/KLayout/klayout/issues/1304 has been fixed
#----------------------------------------------------------------------------------------------

#-------------------------------------------------------
# [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-correct.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, 2, 3, 4
        @spicedeck = [ $choi
      else
        puts "### Usage [ 0=
        puts "       Set one
        puts "       then, ex
        return
    end
  end
end


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

source( "inv.oas", "INVERTER

deep

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


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

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


#-------------
# [6] Define cc
#-------------
# Inter-layer
connect(psd,
connect(nsd,
connect(poly,
connect(contact
connect(metal1,
connect(metal1,
connect(via1,
connect(metal2,

# Global
connect_global(
connect_global(


#-------------
# [7] Compare s
#-------------
schematic(spice

compare

# EOF


#-------------------------------------------------------
# [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)

# Bulk layer for terminal provisioning
bulk = polygon_layer


#-------------------------------------------------------
# [4] Computed layers
#-------------------------------------------------------
active_in_nwell      = active & nwell
pactive              = active_in_nwell & pplus
pgate                = pactive & poly
psd                  = pactive - pgate

active_outside_nwell = active - nwell
nactive              = active_outside_nwell & nplus
ngate                = nactive & poly
nsd                  = nactive - ngate


#-------------------------------------------------------
# [5] Device extraction
#-------------------------------------------------------
# PMOS transistor device extraction
extract_devices(mos4("PMOS"), { "SD" => psd, "G" => pgate, "W" => nwell,
                                "tS" => psd, "tD" => psd, "tG" => poly, "tW" => nwell })

# NMOS transistor device extraction
extract_devices(mos4("NMOS"), { "SD" => nsd, "G" => ngate, "W" => bulk,
                                "tS" => nsd, "tD" => nsd, "tG" => poly, "tW" => bulk })
```

21. Using MOS4 for the **inv.cir** Tutorial

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

inv.cir | invX1.cir | invX2.cir | inv-correct.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

This net matched as expected.

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 ⇔ D	VDD (1)	VDD (2)
	D ⇔ S	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-Con2/Inv/inv.oas'
         Top cell name = 'INVERTER'
        SPICE deck file = 'inv.cir'
```

21. Using MOS4 for the **inv.cir** Tutorial

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

inv.cir | invX1.cir | invX2.cir | inv-correct.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
```

$choice=2

This net matched as expected.

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 ⇔ D	VDD (1)	VDD (2)
	D ⇔ S	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-Con2/Inv/inv.oas'
         Top cell name = 'INVERTER'
        SPICE deck file = 'invX1.cir'
```

21. Using MOS4 for the **inv.cir** Tutorial

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

inv.cir | invX1.cir | invX2.cir | inv-correct.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 VDD NWELL PMOS W=1.5U L=0.25U
6  Mn VSS IN OUT SUBSTRATE NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=3

This net matched as expected.

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)

Configure | Probe Net | Lock

```
> $choice=3
3
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv/inv.oas'
         Top cell name = 'INVERTER'
        SPICE deck file = 'invX2.cir'
```

21. Using MOS4 for the **inv.cir** Tutorial

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

inv.cir | invX1.cir | invX2.cir | inv-correct.cir

```
1
2  * Simple CMOS inverter circuit
3  *
4  * Original: ${klayout_root}/testdata/lvs/inv.cir
5  *
6  * Modified: inv-correct.cir (PMOS, NMOS)=(  correct,   correct)
7  *       by: Kazzz-S
8  *     date: 2023-03-06
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

This net matched as expected.

Netlist | Schematic | Cross Reference | Log

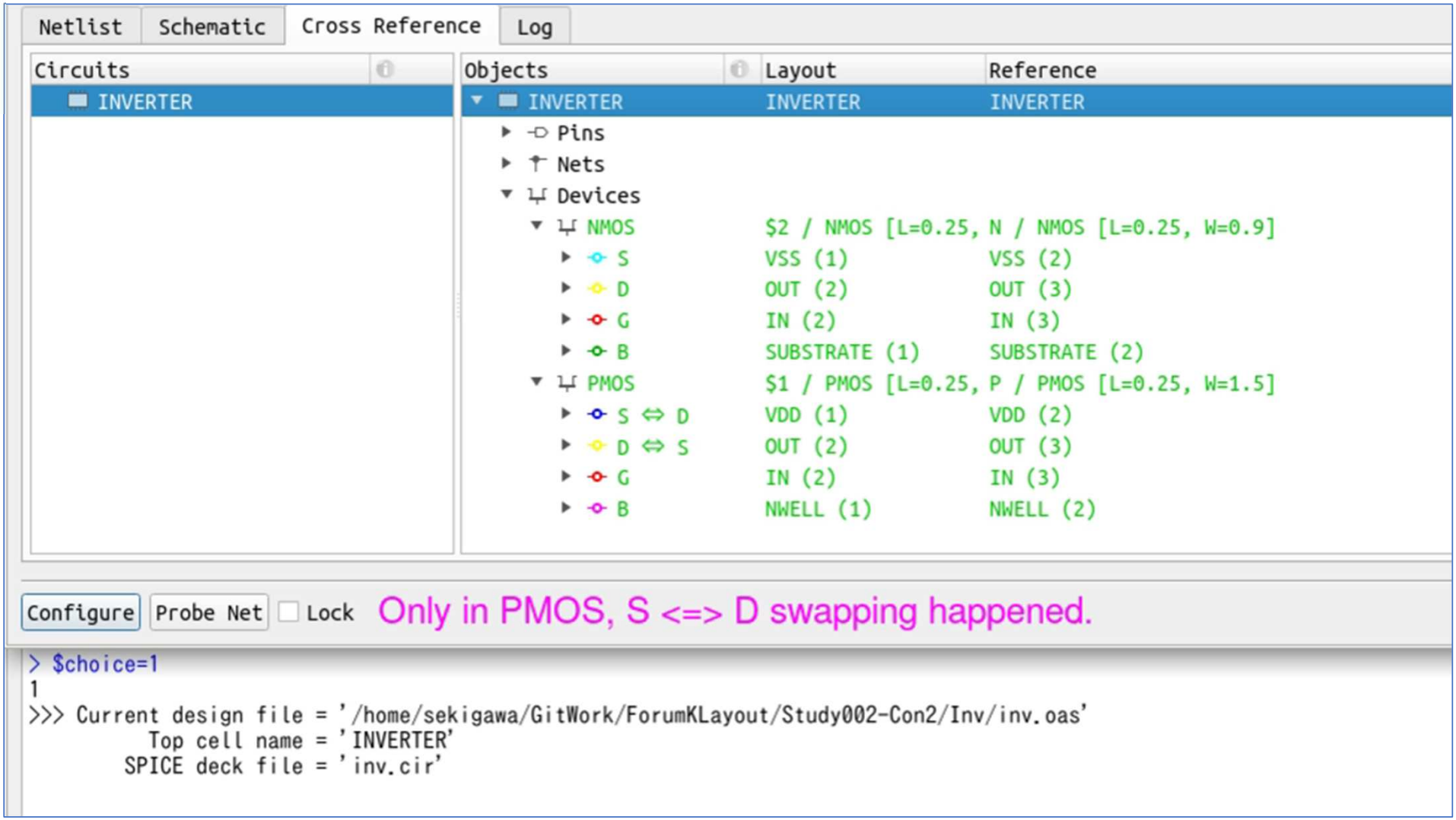
Circuits	Objects	Layout	Reference
INVERTER	INVERTER	INVERTER	INVERTER
	Pins		
	Nets		
	Devices		
	NMOS	$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	S	VSS (1)	VSS (2)
	D	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	SUBSTRATE (1)	SUBSTRATE (2)
	PMOS	$1 / PMOS [L=0.25, P / PMOS [L=0.25, W=1.5]	
	S	VDD (1)	VDD (2)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

Configure | Probe Net | Lock

```
> $choice=4
4
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv/inv.oas'
         Top cell name = 'INVERTER'
        SPICE deck file = 'inv-correct.cir'
```

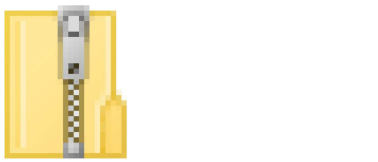
21. Using MOS4 for the **inv.cir** Tutorial

Video



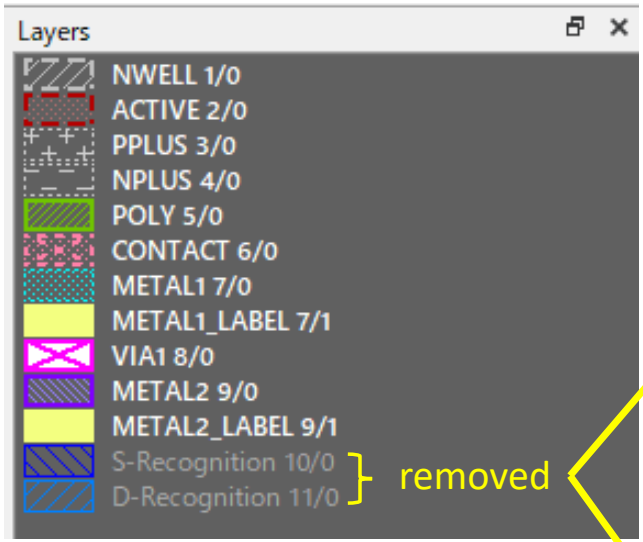
22. Using MOS4 for the **inv2.cir** Tutorial

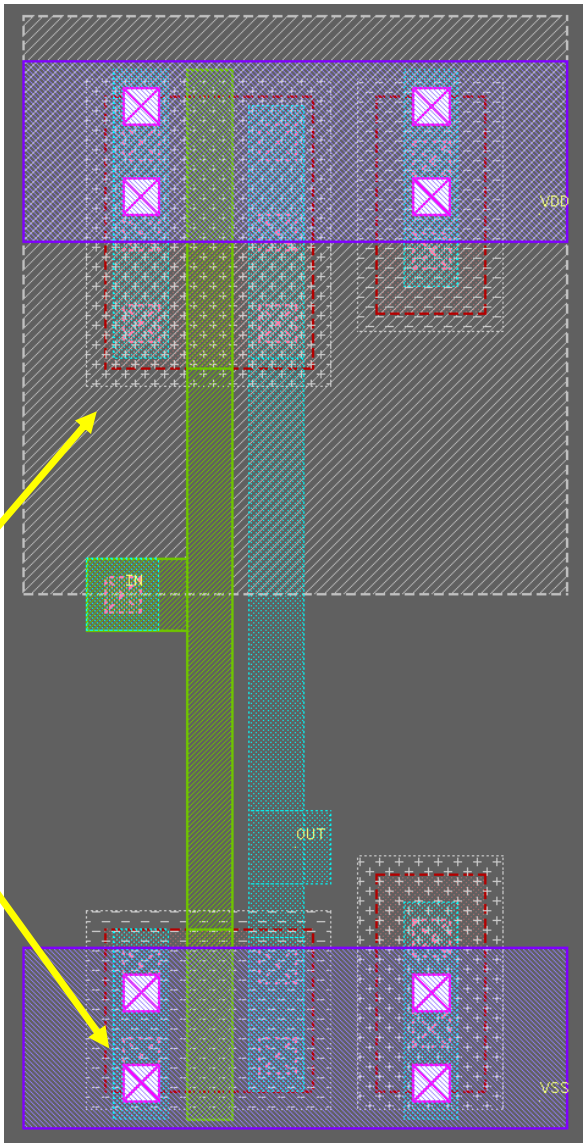
1. Remove the two dummy layers for S- and D-recognition that were added to use the DMOS4 extractor.



All Files

Inv2.zip





22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit : original: (PMOS, NMOS)=(incorrect, correct)
3
4  .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
5  Mp VDD IN OUT VDD PMOS W=1.5U L=0.25U
6  Mn OUT IN VSS VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=1

D and S are swapped

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit: Variant X1: (PMOS, NMOS)=(incorrect, incorrect
3
4  .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
5  Mp VDD IN OUT VDD PMOS W=1.5U L=0.25U
6  Mn VSS IN OUT VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=2

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit: Variant X2: (PMOS, NMOS)=(  correct, incorrect
3
4  .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
5  Mp OUT IN VDD VDD PMOS W=1.5U L=0.25U
6  Mn VSS IN OUT VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=3

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverter circuit
3  *
4  * Original: ${klayout_root}/testdata/lvs/inv2.cir
5  *
6  * Modified: inv2-correct.cir (PMOS, NMOS)=(  correct,   correct)
7  *       by: Kazzz-S
8  *     date: 2023-03-06
9  *     aims: 1) correct D-S connections of the PMOS (flipped in the original)
10
11 .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
12 * Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
13 Mp OUT IN VDD VDD PMOS W=1.5U L=0.25U
14 Mn OUT IN VSS VSS NMOS W=0.9U L=0.25U
15 .ENDS
16
```

$choice=4

I expect all four should match by S-D swapping.

22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.lvs

```
#-------------------------------------------------------------------------------------------------
# Original: ${klayout_root}/testdata/lvs/inv2.lvs
#
# Modified: inv2.lvs
#       by: Kazzz-S
#     date: 2023-03-06
#     aims: 1) retry the four SPICE deck files, namely ["inv.cir", "invX1.cir", "invX2.cir",
#               "inv-correct.cir"] with the "MOS4" (not "DMOS4") extractor after the issue
#               in https://github.com/KLayout/klayout/issues/1304 has been fixed
#-------------------------------------------------------------------------------------------------

#-------------------------------------------------------
# [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"  => "inv2.cir",         # (PMOS, NMOS)=(incorrect,   correct)
                   "variantX1" => "inv2X1.cir",       # (PMOS, NMOS)=(incorrect, incorrect)
                   "variantX2" => "inv2X2.cir",       # (PMOS, NMOS)=(  correct, incorrect)
                   "correct"   => "inv2-correct.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, 2, 3, 4
        @spicedeck = [ $choice, @spiceDECK[ @spice
      else
        puts "### Usage [ 0=>exit, 1=>original, 2=
        puts "      Set one of above options in th
        puts "      then, execute the script by cl
        return
    end
  end
end


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

source( "inv2.oas", "INVERTER_WITH_DIODES" )

deep

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


#-------------------------------------------------------
# [2] Reports
#-------------------------------------------------------
# Store the LVS report to this file
report_lvs( "inv2-choice#{choice}.lvsdb", true )

# Write the extracted netlist to this file
target_netlist( "inv2-choice#{choice}_extracted.cir",
                write_spice,
                "Extracted by KLayout with <#{spicedeck}>" )


#-------------------------------------------------------
# [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)

# Bulk layer for terminal provisioning
bulk = polygon_layer


#-------------------------------------------------------
# [4] Computed layers
#-------------------------------------------------------
active_in_nwell      = active & nwell
pactive              = active_in_nwell
pgate                = pactive & poly
psd                  = pactive - pgate
ntie                 = active_in_nwell

active_outside_nwell = active - nwell
nactive              = active_outside_
ngate                = nactive & poly
nsd                  = nactive - ngate
ptie                 = active_outside_


#-------------------------------------------------------
# [5] Device extraction
#-------------------------------------------------------
# PMOS transistor device extraction
extract_devices(mos4("PMOS"), { "SD" => psd, "G" => pgate, "W" => nwell,
                                "tS" => psd, "tD" => psd, "tG" => poly, "tW" => nwell })

# NMOS transistor device extraction
extract_devices(mos4("NMOS"), { "SD" => nsd, "G" => ngate, "W" => bulk,
                                "tS" => nsd, "tD" => nsd, "tG" => poly, "tW" => bulk })


#-------------------------------------------------------
# [6] Define connectivity for intra-layer and inter-layer connections
#-------------------------------------------------------
# Inter-layer
connect(psd,      contact)
connect(nsd,      contact)
connect(poly,     contact)
connect(ntie,     contact)
connect(nwell,    ntie)
connect(ptie,     contact)
connect(contact,  metal1)
connect(metal1,   via1)
connect(via1,     metal2)
connect(metal2,   metal2_lbl)

# Global
connect_global(bulk, "SUB")
connect_global(ptie, "SUB")


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

compare

# EOF
```

22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit : original: (PMOS, NMOS)=(incorrect, correct)
3
4  .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
5  Mp VDD IN OUT VDD PMOS W=1.5U L=0.25U
6  Mn OUT IN VSS VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=1

This net matched as expected.

Netlist | Schematic | Cross Reference | Log

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

Configure | Probe Net | Lock

```
> $choice=1
1
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv2/inv2.oas'
         Top cell name = 'INVERTER_WITH_DIODES'
        SPICE deck file = 'inv2.cir'
```

22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit  Variant X1: (PMOS, NMOS)=(incorrect, incorrect
3
4  .SUBCKT INVERTER WITH DIODES VSS IN OUT VDD
5  Mp VDD IN OUT VDD PMOS W=1.5U L=0.25U
6  Mn VSS IN OUT VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=2

This net matched as expected.

Netlist | Schematic | Cross Reference | Log

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

Configure | Probe Net | Lock

```
> $choice=2
2
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv2/inv2.oas'
         Top cell name = 'INVERTER_WITH_DIODES'
        SPICE deck file = 'inv2X1.cir'
```

22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverer circuit  Variant X2: (PMOS, NMOS)=(  correct, incorrect
3
4  .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
5  Mp OUT IN VDD VDD PMOS W=1.5U L=0.25U
6  Mn VSS IN OUT VSS NMOS W=0.9U L=0.25U
7  .ENDS
8
```

$choice=3

This net matched as expected.

Netlist | Schematic | Cross Reference | Log

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

Configure | Probe Net | Lock

```
> $choice=3
3
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv2/inv2.oas'
         Top cell name = 'INVERTER_WITH_DIODES'
        SPICE deck file = 'inv2X2.cir'
```

22. Using MOS4 for the **inv2.cir** Tutorial

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

inv2.cir | inv2X1.cir | inv2X2.cir | inv2-correct.cir

```
1
2  * Simple CMOS inverter circuit
3  *
4  * Original: ${klayout_root}/testdata/lvs/inv2.cir
5  *
6  * Modified: inv2-correct.cir (PMOS, NMOS)=(  correct,   correct)
7  *       by: Kazzz-S
8  *     date: 2023-03-06
9  *     aims: 1) correct D-S connections of the PMOS (flipped in the original)
10
11 .SUBCKT INVERTER_WITH_DIODES VSS IN OUT VDD
12 * Mp VDD IN OUT NWELL PMOS W=1.5U L=0.25U
13 Mp OUT IN VDD VDD PMOS W=1.5U L=0.25U
14 Mn OUT IN VSS VSS NMOS W=0.9U L=0.25U
15 .ENDS
16
```

$choice=4

This net matched as expected.

Netlist | Schematic | Cross Reference | Log

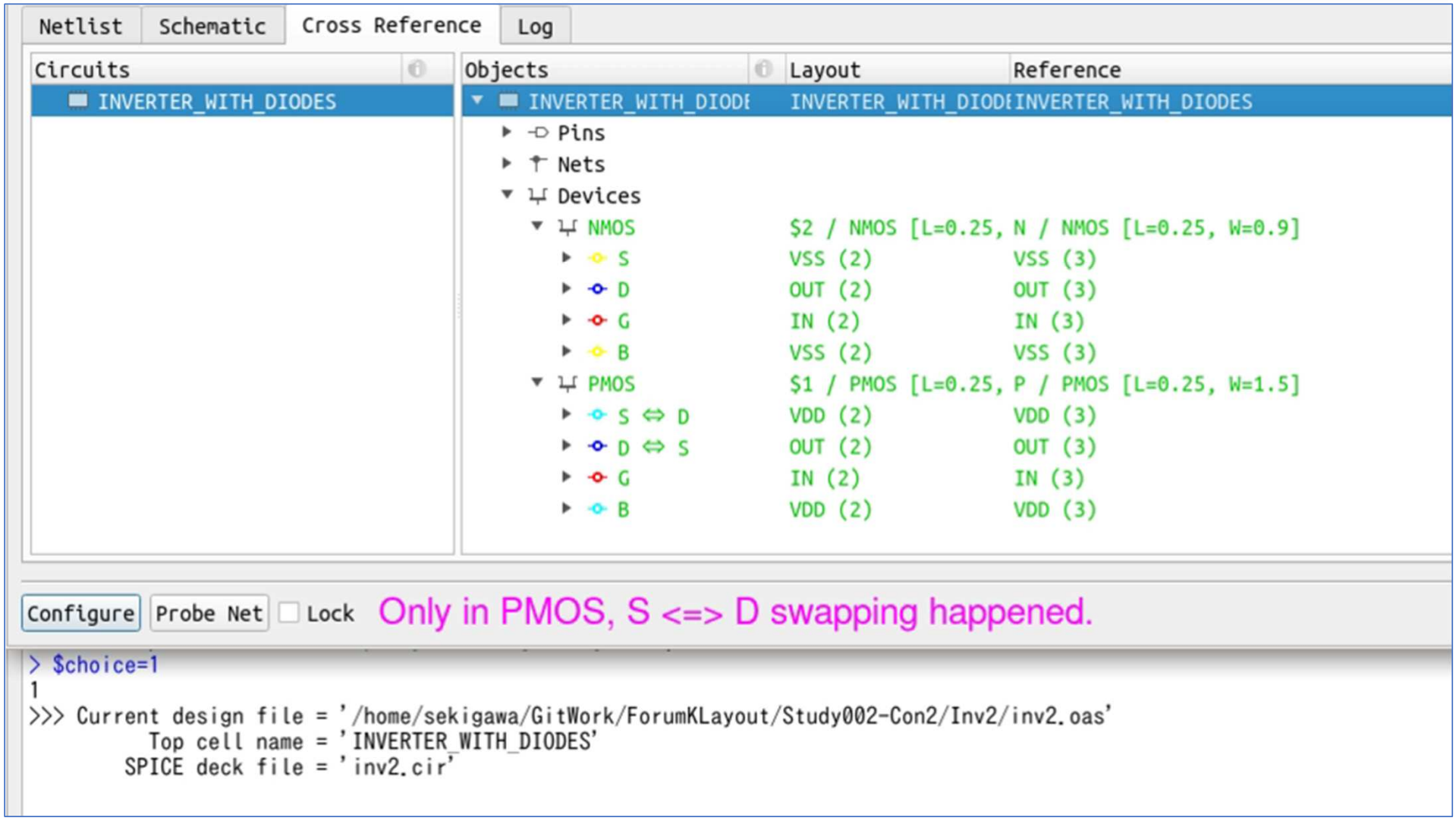
Circuits	Objects	Layout	Reference
INVERTER_WITH_DIODES	INVERTER_WITH_DIODE	INVERTER_WITH_DIODE	INVERTER_WITH_DIODES
	Pins		
	Nets		
	Devices		
	NMOS	$2 / NMOS [L=0.25, N / NMOS [L=0.25, W=0.9]	
	S	VSS (2)	VSS (3)
	D	OUT (2)	OUT (3)
	G	IN (2)	IN (3)
	B	VSS (2)	VSS (3)
	PMOS	$1 / PMOS [L=0.25, P / PMOS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

Configure | Probe Net | Lock

```
> $choice=4
4
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/Inv2/inv2.oas'
         Top cell name = 'INVERTER_WITH_DIODES'
        SPICE deck file = 'inv2-correct.cir'
```

22. Using MOS4 for the **inv2.cir** Tutorial

Video



23. Using MOS4 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.

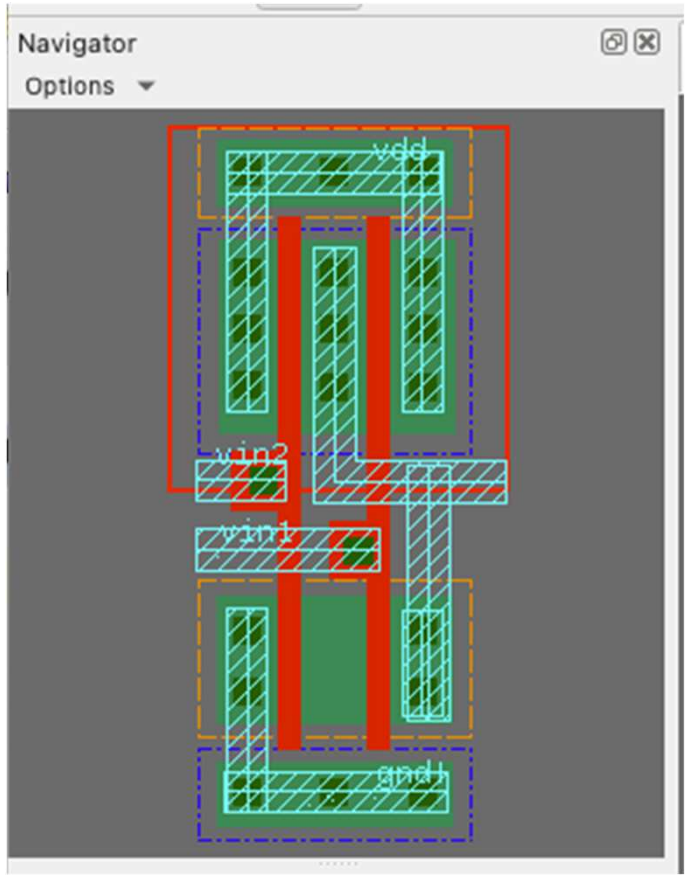
nand.test.zip
14.7K

⟵ Starting point. Thank you for providing this!

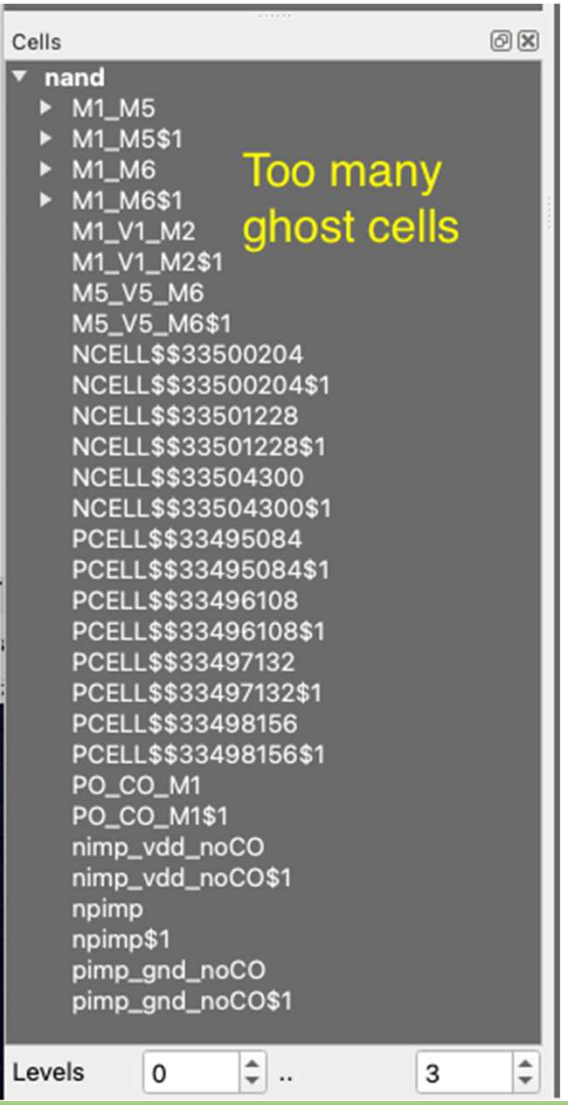
1. Check the provided resource files and modify them if necessary.

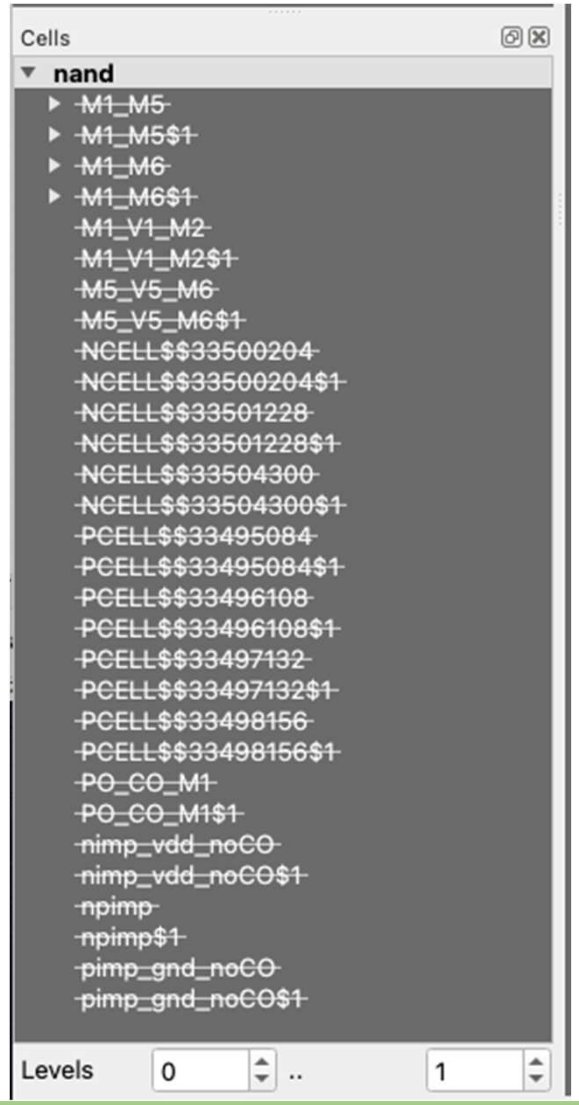
All Files

NAND.zip

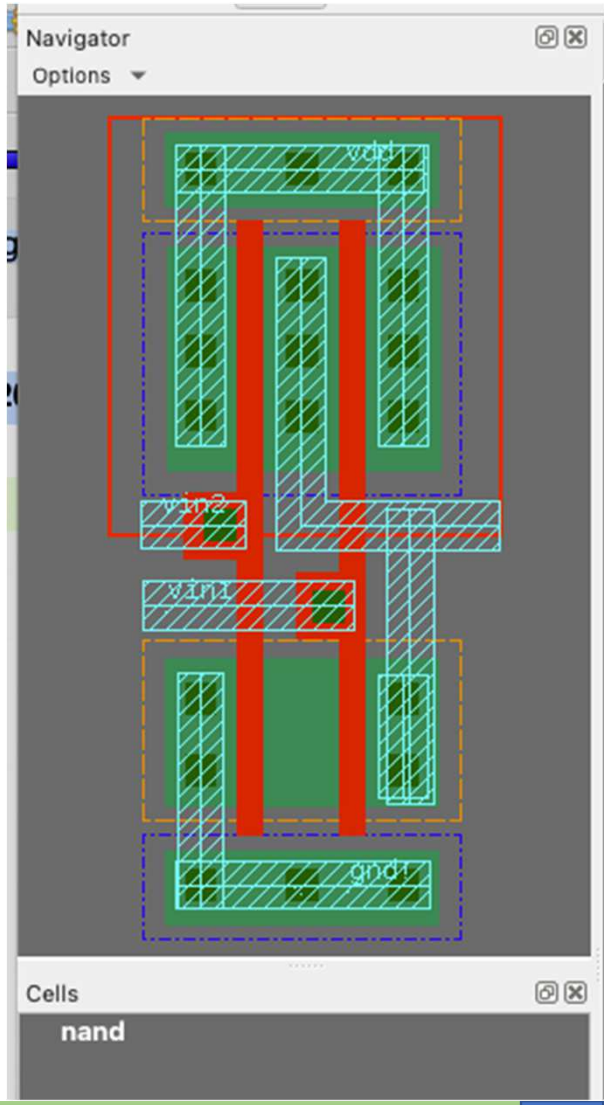


23. Using MOS4 for a **2-input NAND**



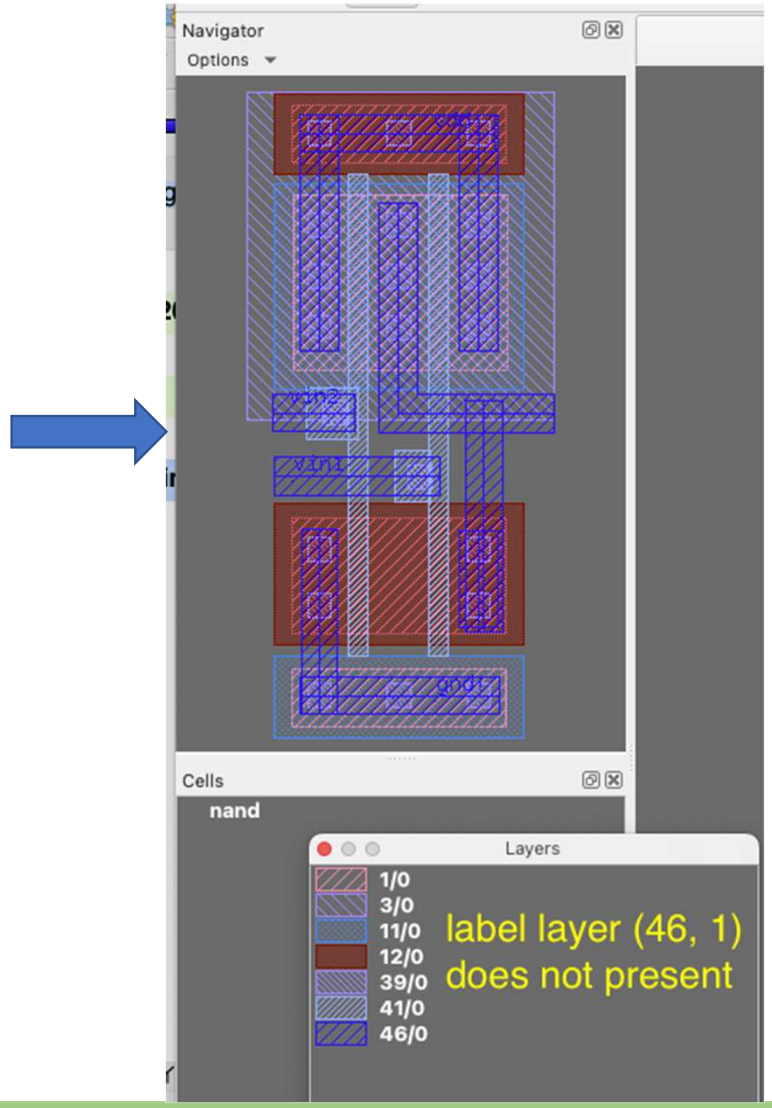


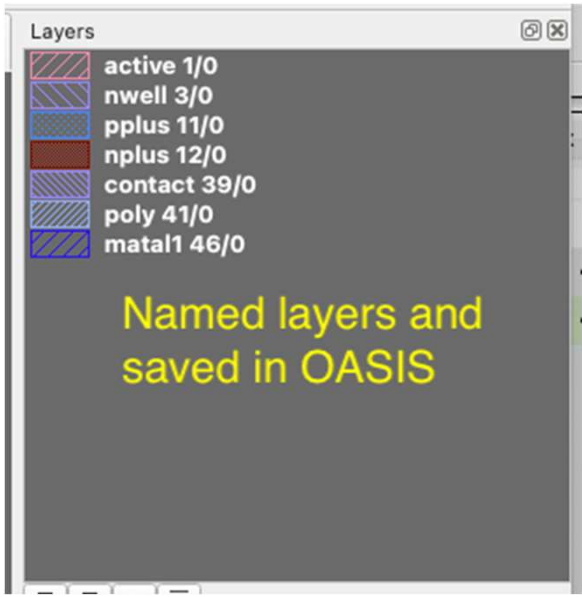
I need only the top cell.



23. Using MOS4 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
```





23. Using MOS4 for a **2-input NAND**

2. Using nandSpice16.py, prepare 16 distinct SPICE deck files that cover all D-S permutations/combinations.

nand-choice01.cir

```
* KLayout Forum: https://www.klayout.de/forum/discussion/2238/
*
* SPICE deck file for WENSHIH's 2-input NAND
*   generator = <nandSpice16.py> by Kazzz-S
*      choice = 1
*      status = [original] all D-S pairs are connected correctly
*
.SUBCKT nand VDD GND
M_M6 GND     VIN2 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
M_M3 N15943 VIN1 OUT     SUBSTRATE NMOS L=0.18um W=1.08um
M_M5 VDD     VIN1 OUT     NWELL     PMOS L=0.18um W=1.62um
M_M4 VDD     VIN2 OUT     NWELL     PMOS L=0.18um W=1.62um
.ENDS
```

nand-choice08.cir

```
* KLayout Forum: https://www.klayout.de/forum/discussion/2238/
*
* SPICE deck file for WENSHIH's 2-input NAND
*   generator = <nandSpice16.py> by Kazzz-S
*      choice = 8
*      status = [incorrect] at least one D-S pair is swapped
*
.SUBCKT nand VDD GND IN1 IN2 OUT NWELL SUBSTRATE
M_M6 N15943 VIN2 GND     SUBSTRATE NMOS L=0.18um W=1.08um
M_M3 OUT     VIN1 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
M_M5 OUT     VIN1 VDD     NWELL     PMOS L=0.18um W=1.62um
M_M4 OUT     VIN2 VDD     NWELL     PMOS L=0.18um W=1.62um
.ENDS
```

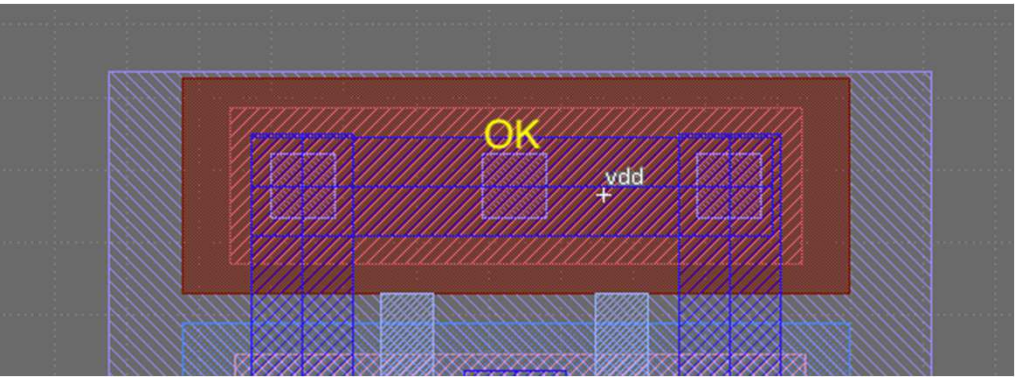
nand-choice16.cir

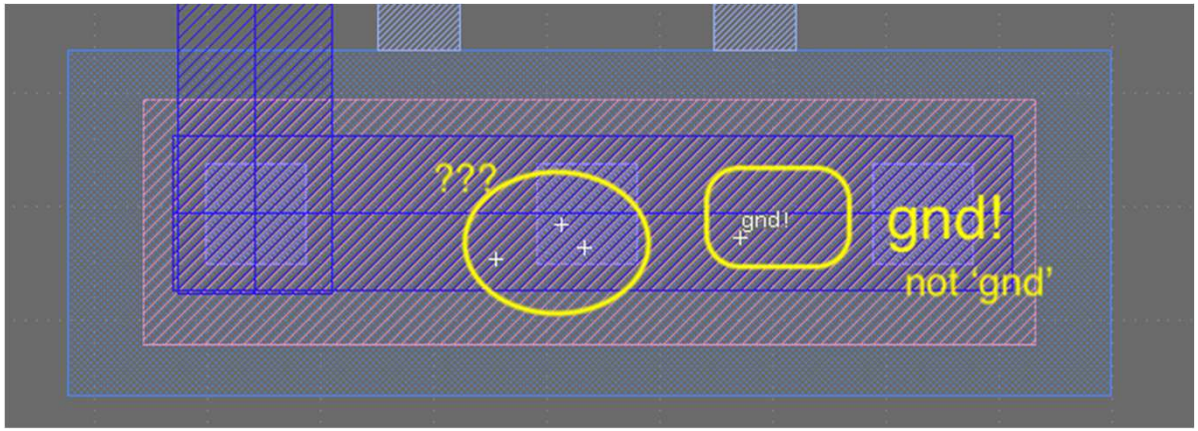
```
* KLayout Forum: https://www.klayout.de/forum/discussion/2238/
*
* SPICE deck file for WENSHIH's 2-input NAND
*   generator = <nandSpice16.py> by Kazzz-S
*      choice = 16
*      status = [correct] all D-S pairs are connected correctly
*
.SUBCKT nand VDD GND IN1 IN2 OUT NWELL SUBSTRATE
M_M6 N15943 VIN2 GND     SUBSTRATE NMOS L=0.18um W=1.08um
M_M3 OUT     VIN1 N15943 SUBSTRATE NMOS L=0.18um W=1.08um
M_M5 OUT     VIN1 VDD     NWELL     PMOS L=0.18um W=1.62um
M_M4 OUT     VIN2 VDD     NWELL     PMOS L=0.18um W=1.62um
.ENDS
```

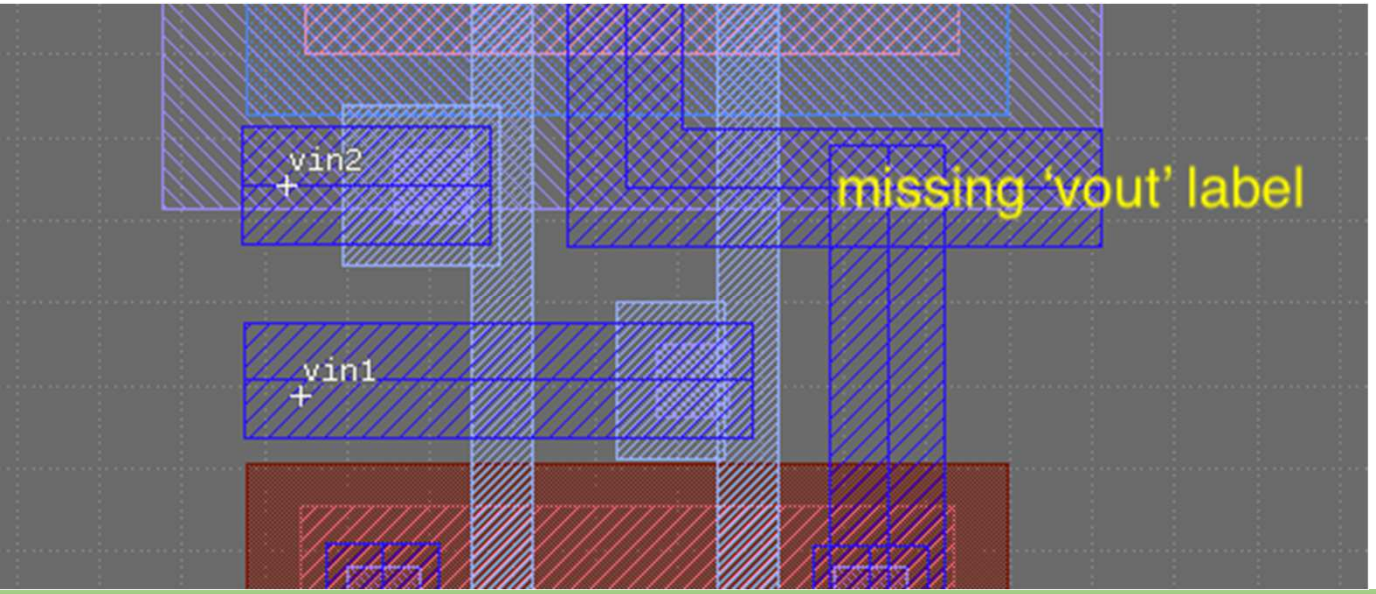
I expect ALL 16 SPICE deck files should match by S-D swapping.

23. Using MOS4 for a **2-input NAND**

3. Check the original design in GDS2.





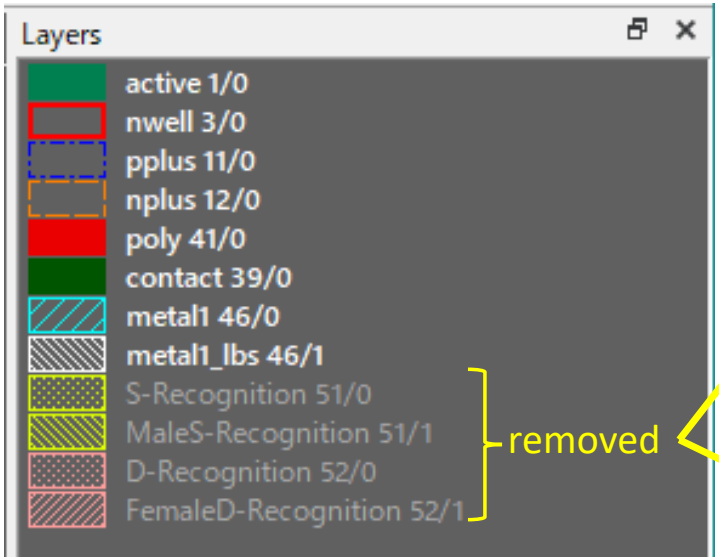


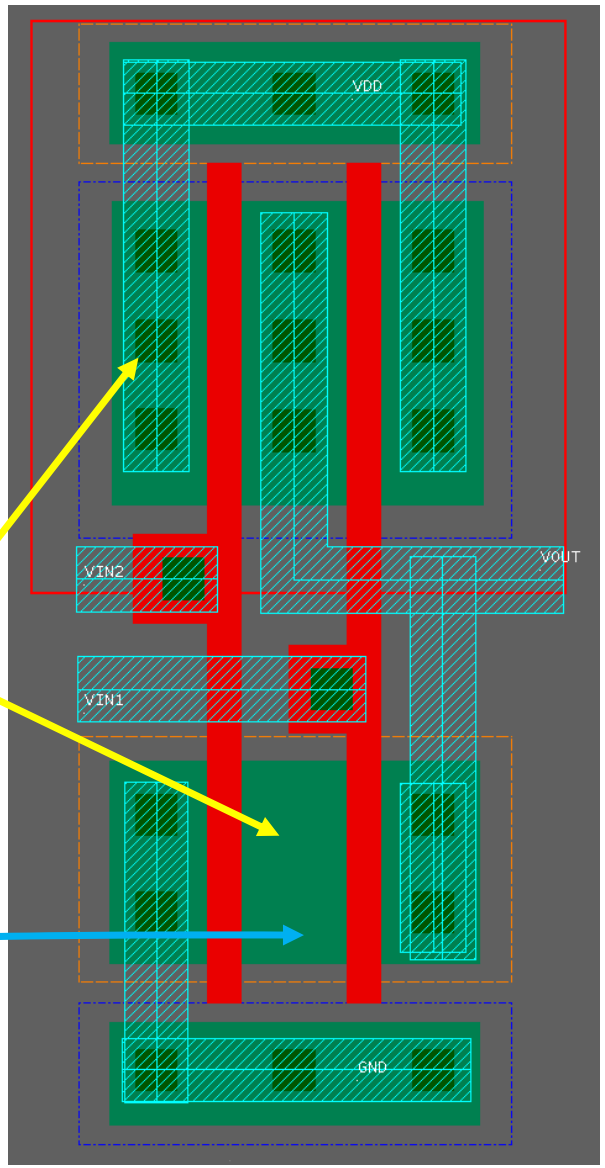
23. Using MOS4 for a **2-input NAND**

4. Remove the four dummy layers for S- and D-recognition that were added to use the DMOS4 extractor.

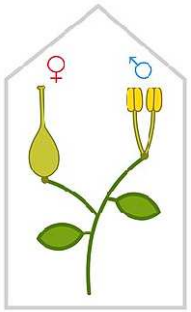
Monoecy (/məˈniːsi/; adj. **monoecious** /məˈniːʃəs/)[1] is a sexual system in seed plants where separate male and female cones or flowers are present on the same plant.

Monoecy - Wikipedia





No geometrical hint implying ***monoecy*** is given.



monoecious plant

23. Using MOS4 for a **2-input NAND**

5. Modify the LVS script for testing and debugging.

nand-MOS4.lvs

```
#-------------------------------------------------------------------------------------------
# Original: "lvs_2_1.lylvs" in the attached ZIP file of
#            https://www.klayout.de/forum/discussion/comment/9493/#Comment_9493
#
# Modified: nand-MOS4.lvs
#       by: Kazzz-S
#     date: 2023-03-08
#     aims: 1) test WENSHIH's NAND circuit with 16 SPICE decks covering all D-S permutations
#           2) use the "MOS4" extractor after the issue
#              in https://github.com/KLayout/klayout/issues/1304 has been fixed
#-------------------------------------------------------------------------------------------

#-------------------------------------------------------
# [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 = 20
    end
    @spicedeck = [ $choice, "" ]

    @spiceOpt = {}
    @spiceOpt[0] = "exit"
    for choice in Range.new(1, 16) do
      @spiceOpt[choice] = "choice%#02d" % choice
    end

    @spiceDECK = {}
    for choice in Range.new(1, 16) do
      opt = @spiceOpt[choice]
      @spiceDECK[opt] = "nand-#{opt}.cir"
    end
  end

  def GetSpiceDeck
    return @spicedeck
  end

  def GetOneOption
    case $choice
      when 0
        puts "Bye!  You can ignore th
        exit
      when 1..16
        @spicedeck = [ $choice, @spic
      else
        puts "### Usage [ 0=>exit, k=
        puts "      Set one of above
        puts "      then, execute the
        return
    end
  end
end


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

source( "nand.oas", "nand" )

deep

selector = SpiceDeckSelector.new
selector.GetOneOption()
choice, spicedeck = selector.GetSpice
if choice == 20 # 1st run
  return
elsif not Range.new(1, 16).include?(c
  puts "! Your choice '#{choice}' is
  return
end
AboutCurrentDesign(spicedeck)


#-------------------------------------------------------
# [2] Reports
#-------------------------------------------------------
# Store the LVS report to this file
report_lvs( "nand-choice%#02d.lvsdb

# Write the extracted netlist to th
target_netlist( "nand-choice%#02d_e
                write_spice,
                "Extracted by KLayo


#-------------------------------------
# [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)

# Bulk layer for terminal provision
bulk = polygon_layer


#-------------------------------------
# [4] Computed layers
#-------------------------------------
active_in_nwell       = active & nw
pactive               = active_in_n
pgate                 = pactive & p
psd                   = pactive - p
#ntie                  = active_in_


active_outside_nwell  = active - nw
nactive               = active_outs
ngate                 = nactive & p
nsd                   = nactive - n
#ptie                  = active_out


#-------------------------------------
# [5] Device extraction
#-------------------------------------------------------
# PMOS transistor device extraction
extract_devices(mos4("PMOS"), { "SD" => psd, "G" => pgate, "W" => nwell,

# NMOS tran
extract_dev


#----------
# [6] Defin
#----------
# Inter-lay
connect(psd
connect(nsd
connect(pol
connect(con
connect(met

# Global
#connect_gl
#connect_gl
connect_glo
connect_glo


#----------
# [7] Compa
#----------
schematic(s

compare

# EOF
```

6. Run the modified LVS script 16 times.

23. Using MOS4 for a **2-input NAND**

Video choice=[1..16]

Still images are in

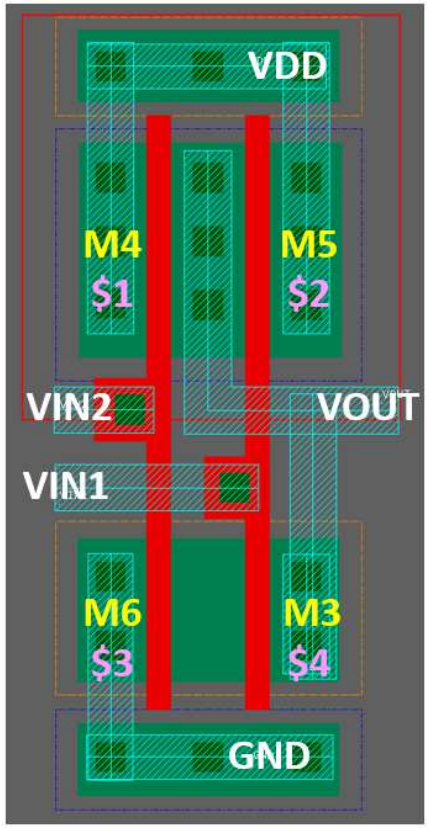
NAND.zip

Netlist | Schematic | Cross Reference | Log

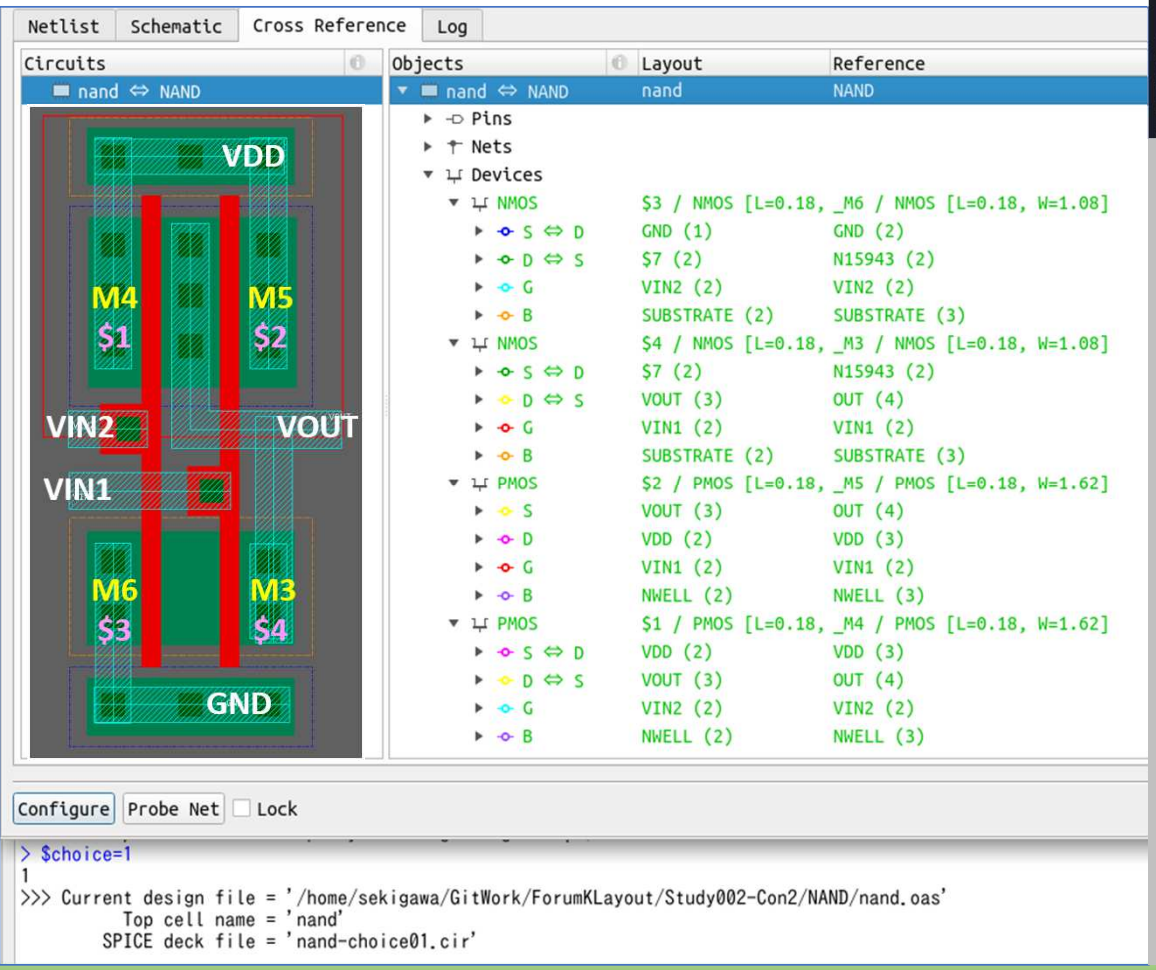
Circuits	Objects	Layout	Reference
nand ⇔ NAND	nand ⇔ NAND	nand	NAND
	Pins		
	Nets		
	Devices		
	NMOS	$3 / NMOS [L=0.18,	_M6 / NMOS [L=0.18, W=1.08]
	S ⇔ D	GND (1)	GND (2)
	D ⇔ S	$7 (2)	N15943 (2)
	G	VIN2 (2)	VIN2 (2)
	B	SUBSTRATE (2)	SUBSTRATE (3)
	NMOS	$4 / NMOS [L=0.18,	_M3 / NMOS [L=0.18, W=1.08]
	S ⇔ D	$7 (2)	N15943 (2)
	D ⇔ S	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	VOUT (3)	OUT (4)
	D	VDD (2)	VDD (3)
	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 ⇔ D	VDD (2)	VDD (3)
	D ⇔ S	VOUT (3)	OUT (4)
	G	VIN2 (2)	VIN2 (2)
	B	NWELL (2)	NWELL (3)

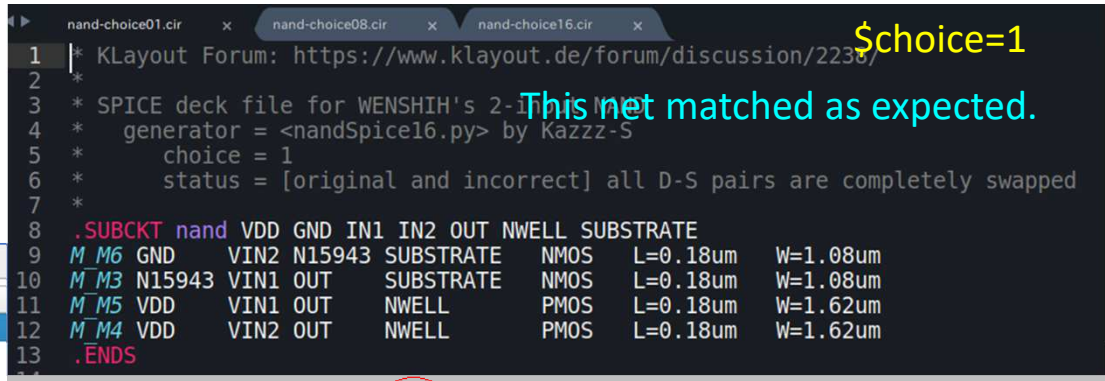
Configure | Probe Net | Lock Only in PMOS _M5, S <=> D swapping did not happen.

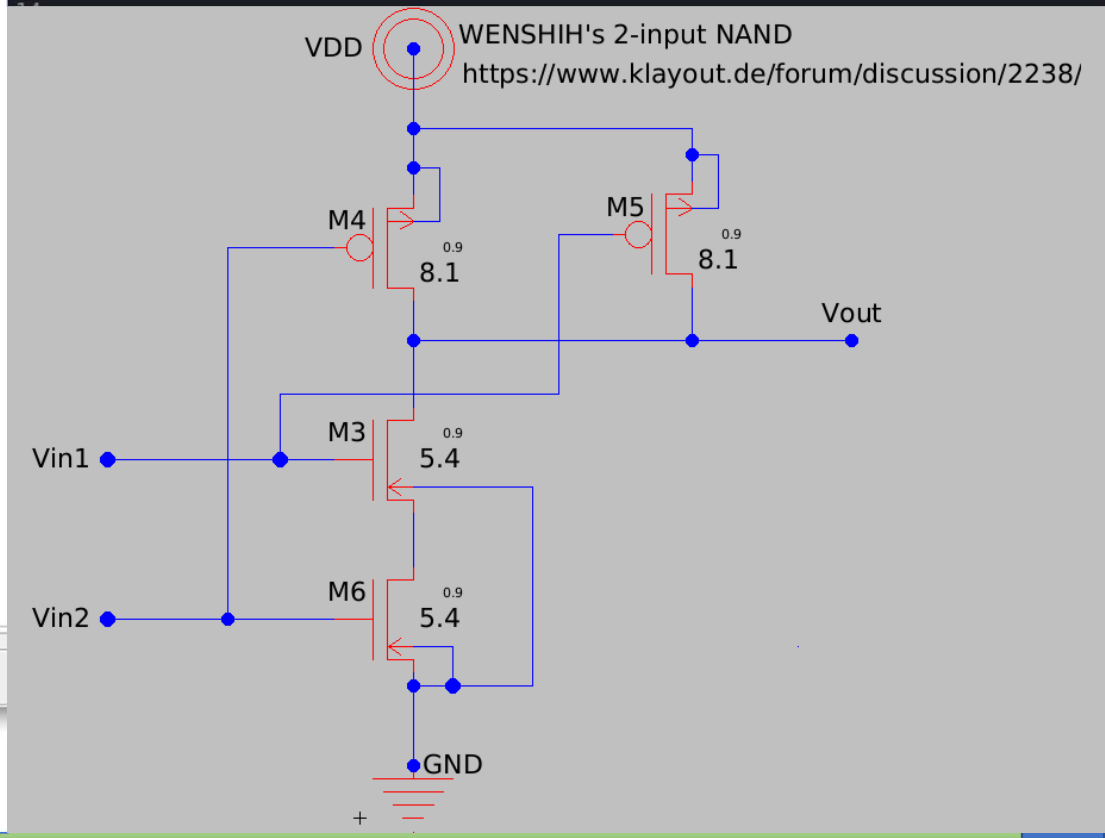
```
> $choice=1
1
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/NAND/nand.oas'
          Top cell name = 'nand'
        SPICE deck file = 'nand-choice01.cir'
```



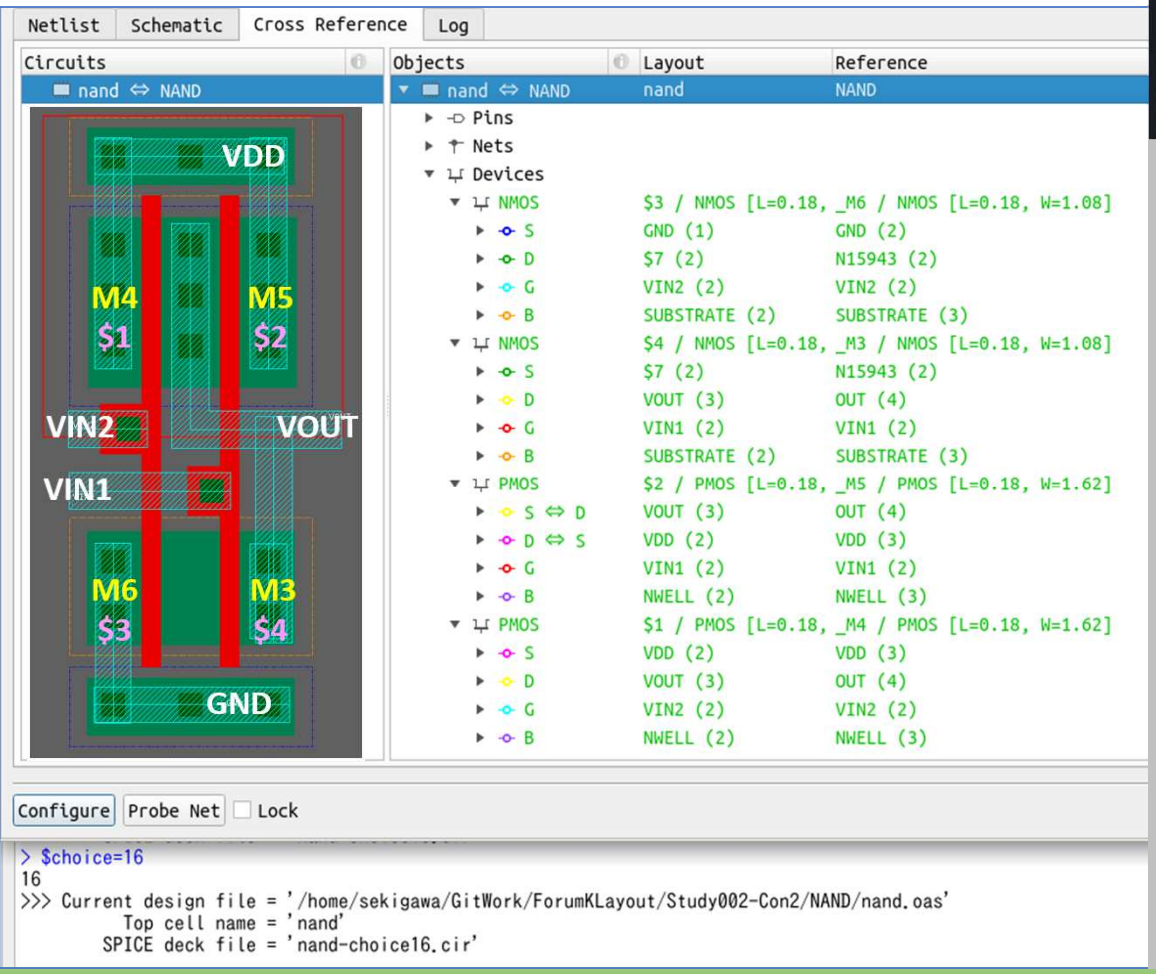
23. Using MOS4 for a **2-input NAND**

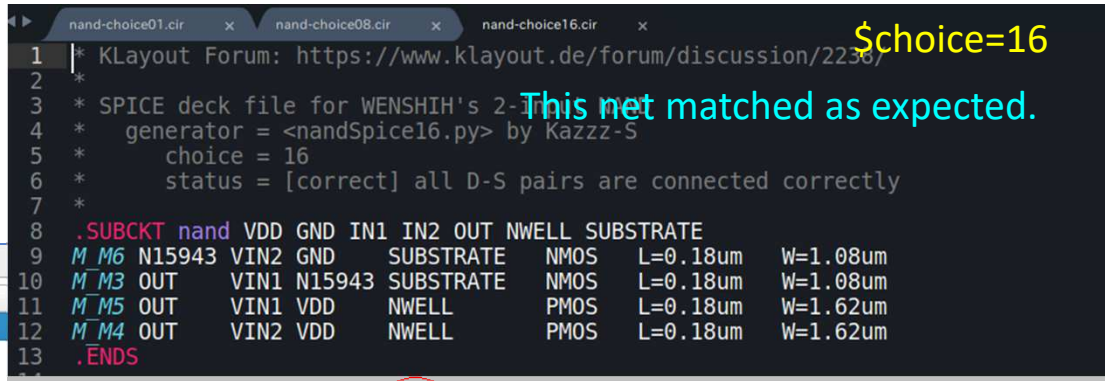


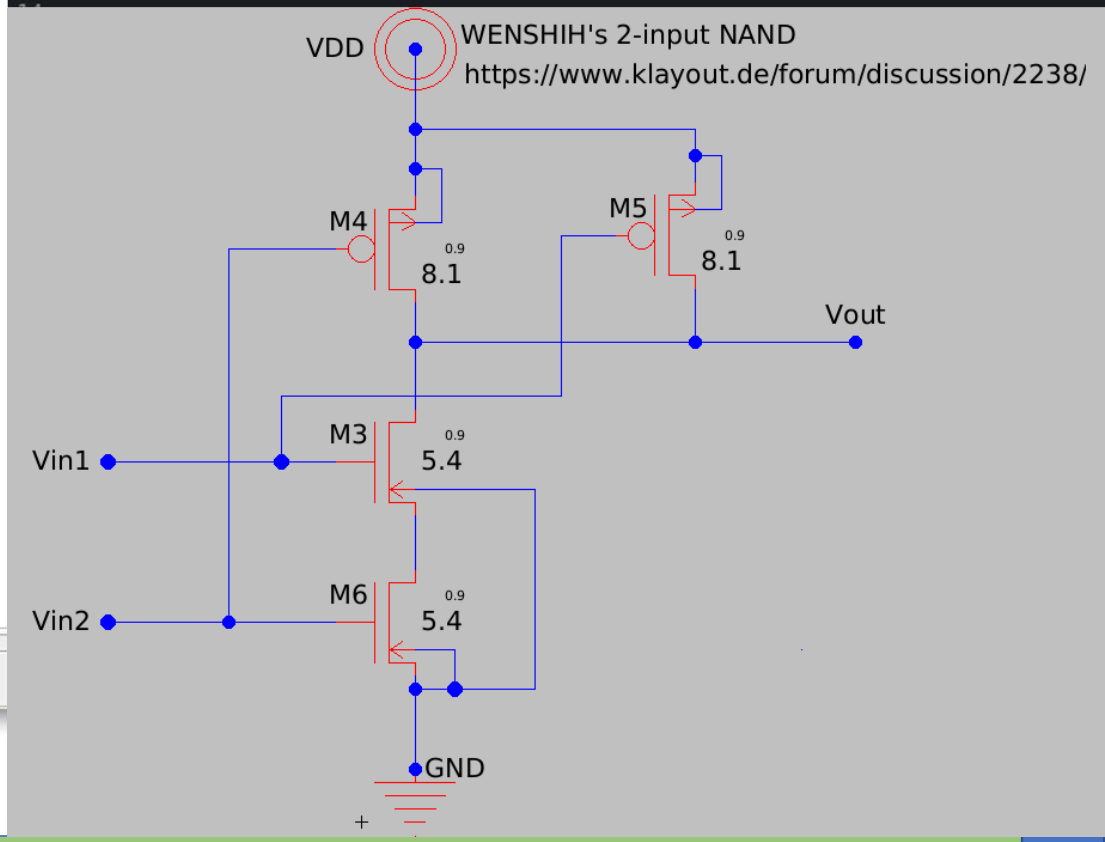




23. Using MOS4 for a **2-input NAND**







23. Using MOS4 for a **2-input NAND**

Video choice=[1, 16]

Still images are in

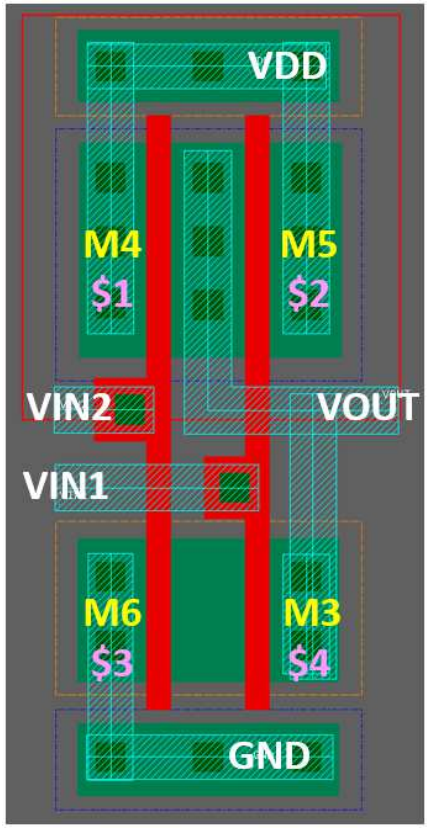
NAND.zip

Netlist | Schematic | Cross Reference | Log

Circuits	Objects	Layout	Reference
nand ⇔ NAND	nand ⇔ NAND	nand	NAND
	Pins		
	Nets		
	Devices		
	NMOS	$3 / NMOS [L=0.18,	_M6 / NMOS [L=0.18, W=1.08]
	S ⇔ D	GND (1)	GND (2)
	D ⇔ S	$7 (2)	N15943 (2)
	G	VIN2 (2)	VIN2 (2)
	B	SUBSTRATE (2)	SUBSTRATE (3)
	NMOS	$4 / NMOS [L=0.18,	_M3 / NMOS [L=0.18, W=1.08]
	S ⇔ D	$7 (2)	N15943 (2)
	D ⇔ S	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	VOUT (3)	OUT (4)
	D	VDD (2)	VDD (3)
	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 ⇔ D	VDD (2)	VDD (3)
	D ⇔ S	VOUT (3)	OUT (4)
	G	VIN2 (2)	VIN2 (2)
	B	NWELL (2)	NWELL (3)

Configure | Probe Net | Lock Only in PMOS _M5, S <=> D swapping did not happen.

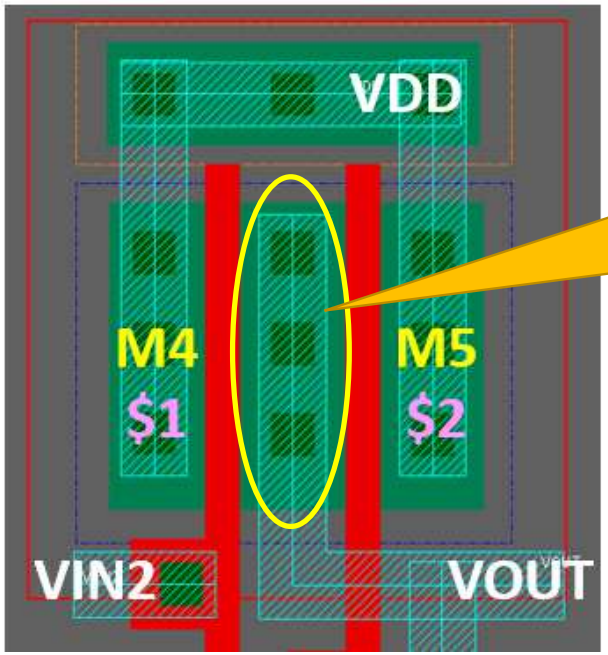
```
> $choice=1
1
>>> Current design file = '/home/sekigawa/GitWork/ForumKLayout/Study002-Con2/NAND/nand.oas'
          Top cell name = 'nand'
        SPICE deck file = 'nand-choice01.cir'
```



24. Summary and Further Questions

- In the case of the Inv and Inv2 LVS, all four reference netlists matched as expected with S⇔D swapping.

- In the case of the 2-input NAND, why does only PMOS M5 behave differently regarding the S⇔D swapping?

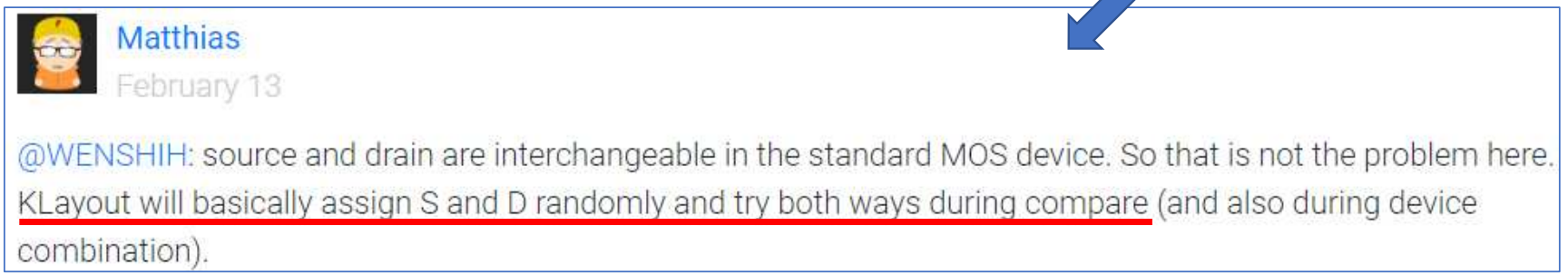


This region appears to be identified as
- the Drain of M4 ($1)

as well as
- the Source of M5 ($2)

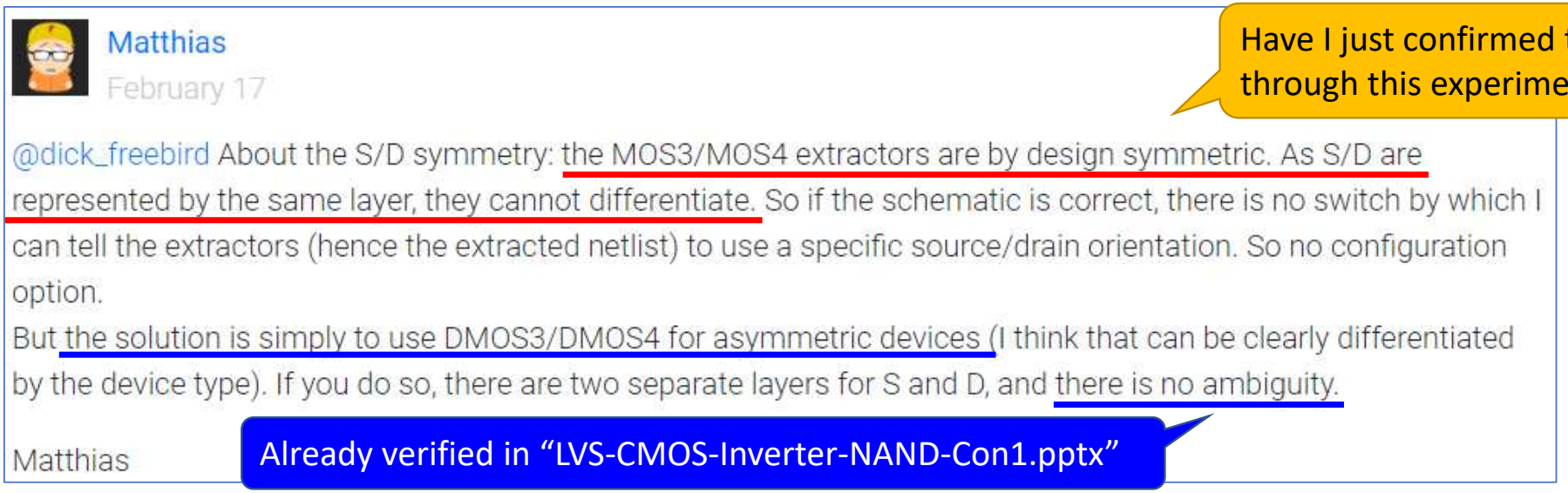
24. Summary and Further Questions

◆ Once a certain combination is matched *by chance* (algorithm/input dependent), are no further possibilities (even if any) checked?



Matthias
February 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).



Matthias
February 17

@dick_freebird About the S/D symmetry: the MOS3/MOS4 extractors are by design symmetric. As S/D are represented by the same layer, they cannot differentiate. So if the schematic is correct, there is no switch by which I can tell the extractors (hence the extracted netlist) to use a specific source/drain orientation. So no configuration option.
But the solution is simply to use DMOS3/DMOS4 for asymmetric devices (I think that can be clearly differentiated by the device type). If you do so, there are two separate layers for S and D, and there is no ambiguity.

Matthias

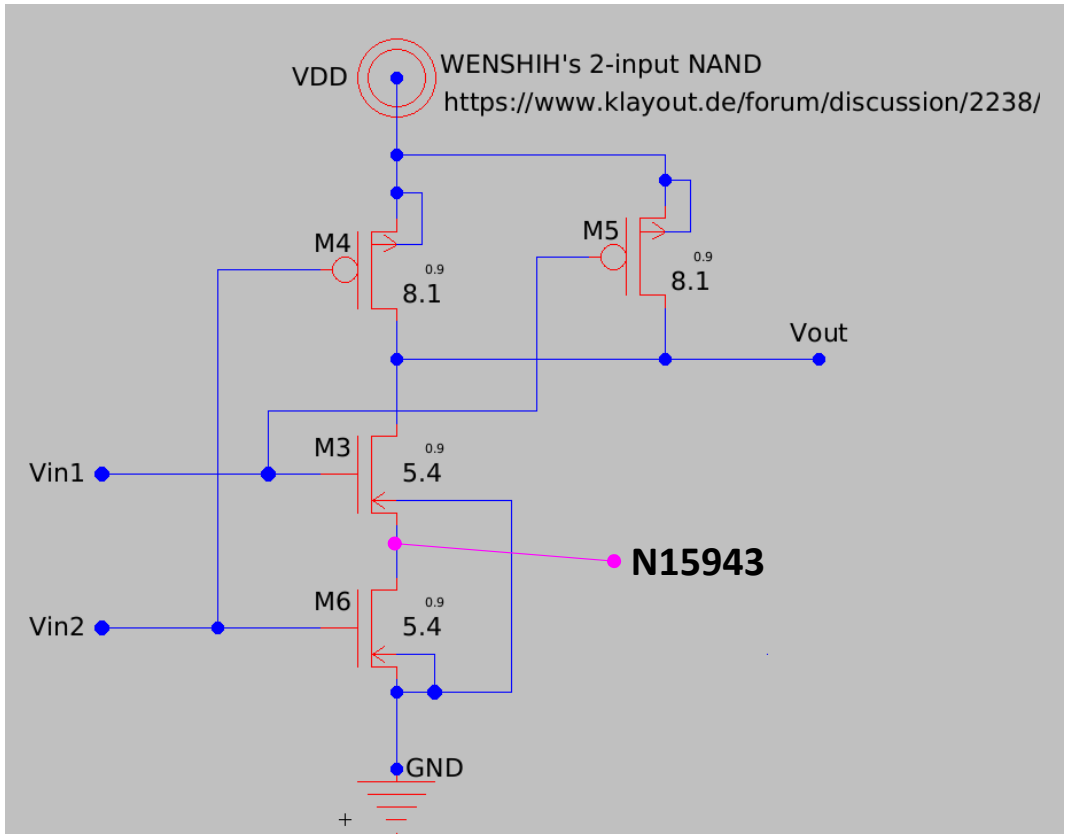
Have I just confirmed this ***specification*** through this experiment?

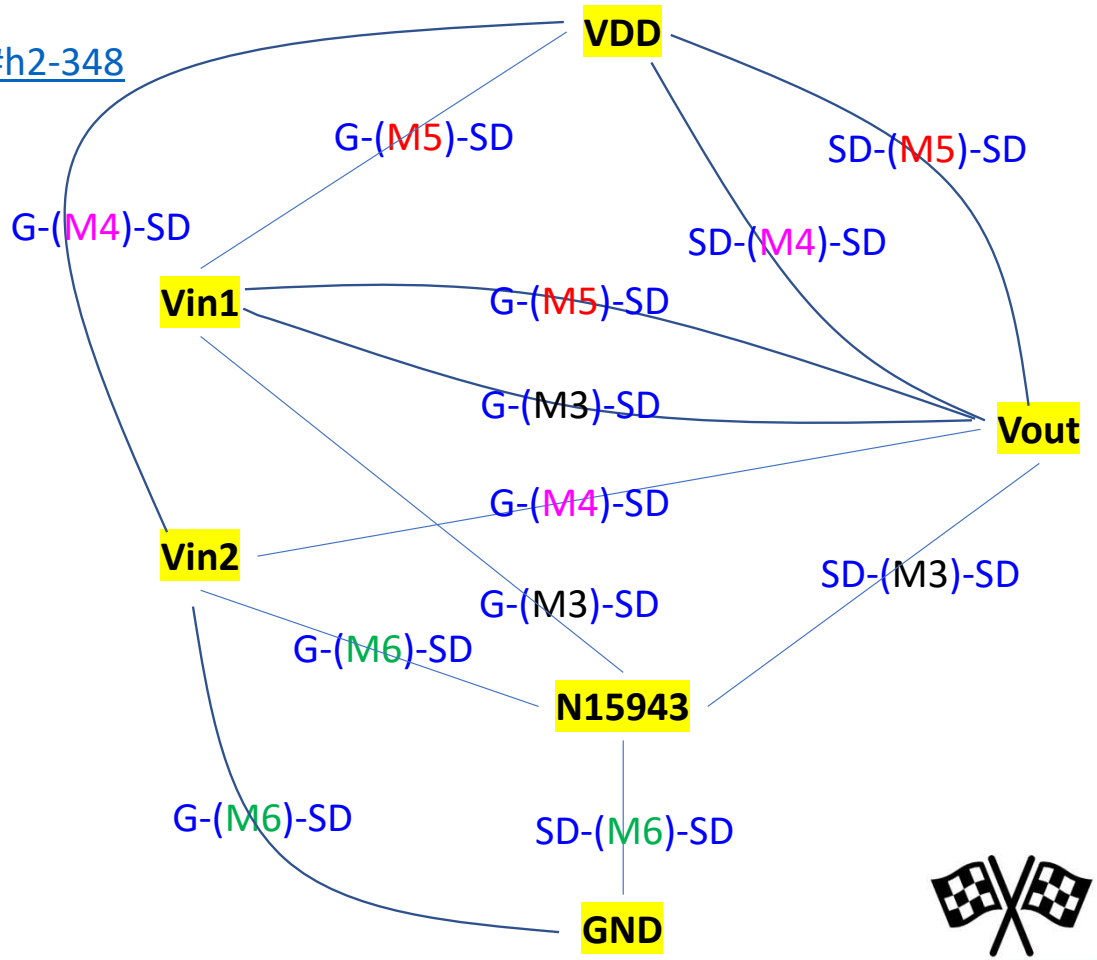
Already verified in "LVS-CMOS-Inverter-NAND-Con1.pptx"

24. Summary and Further Questions

◆ I have drawn the net neighborhood graph for the 2-input NAND for my study.

Ref. https://www.klayout.de/doc-qt5/manual/lvs_compare.html#h2-348





Local Parts Storeroom