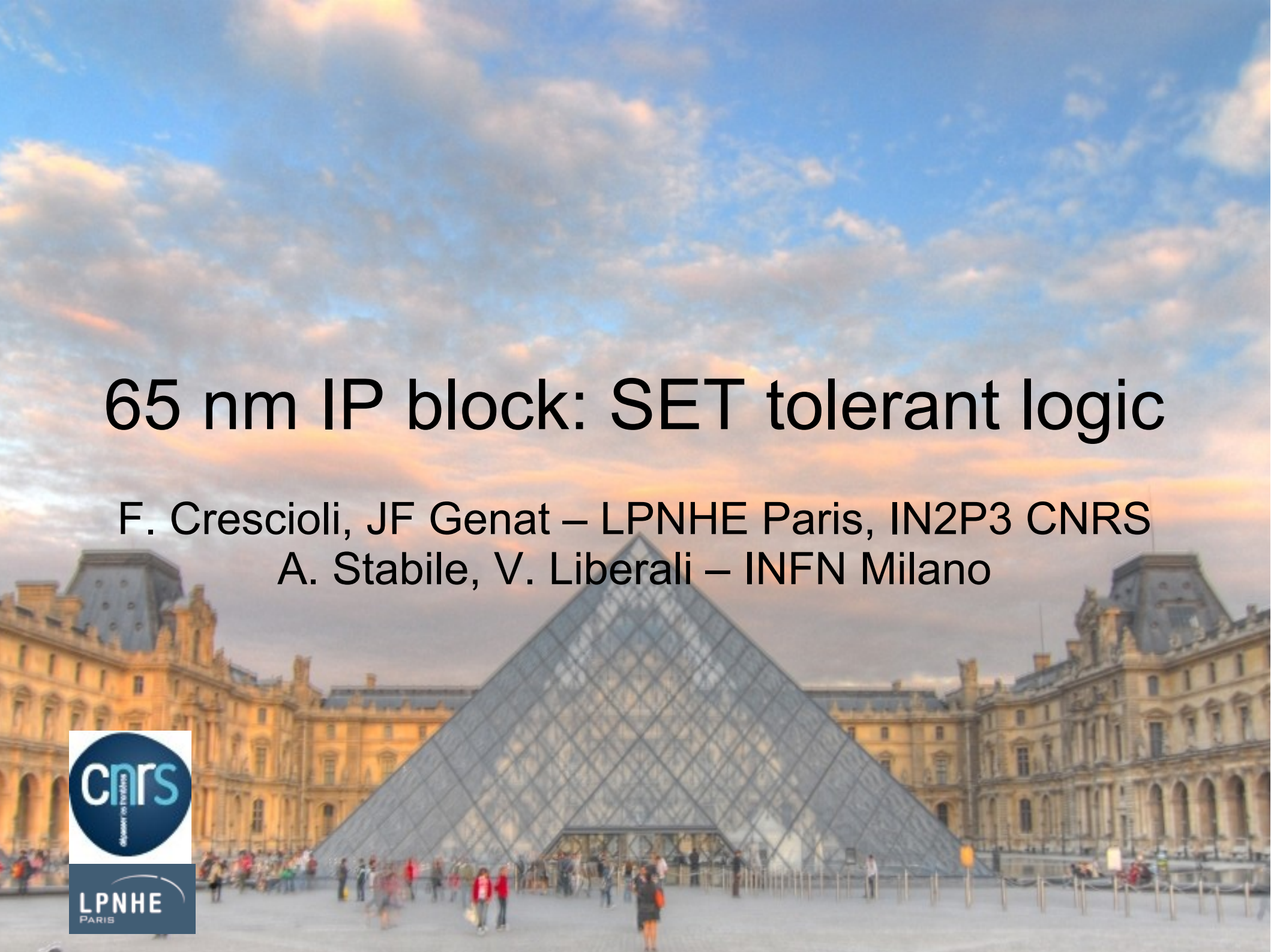
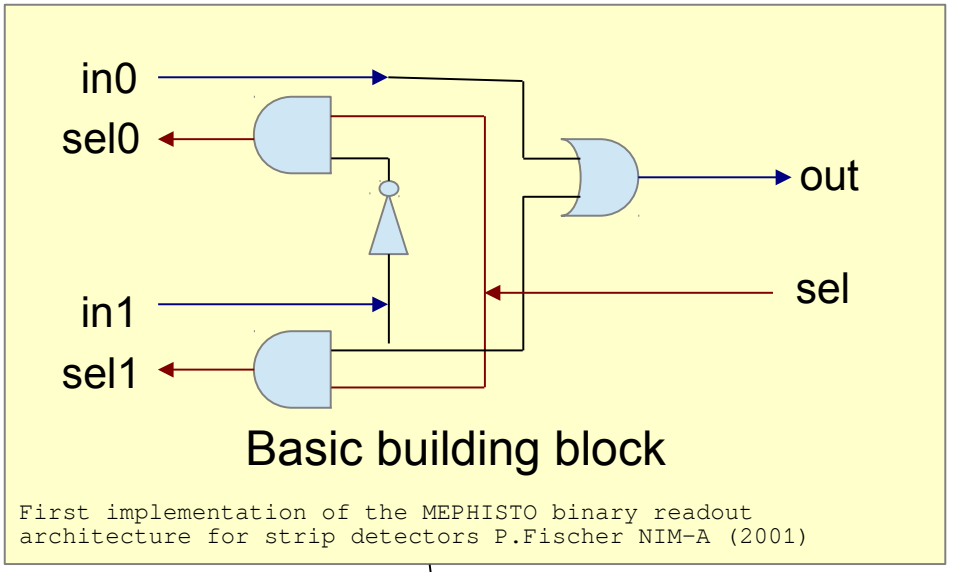


65 nm IP block: SET tolerant logic

F. Crescioli, JF Genat – LPNHE Paris, IN2P3 CNRS
A. Stabile, V. Liberali – INFN Milano

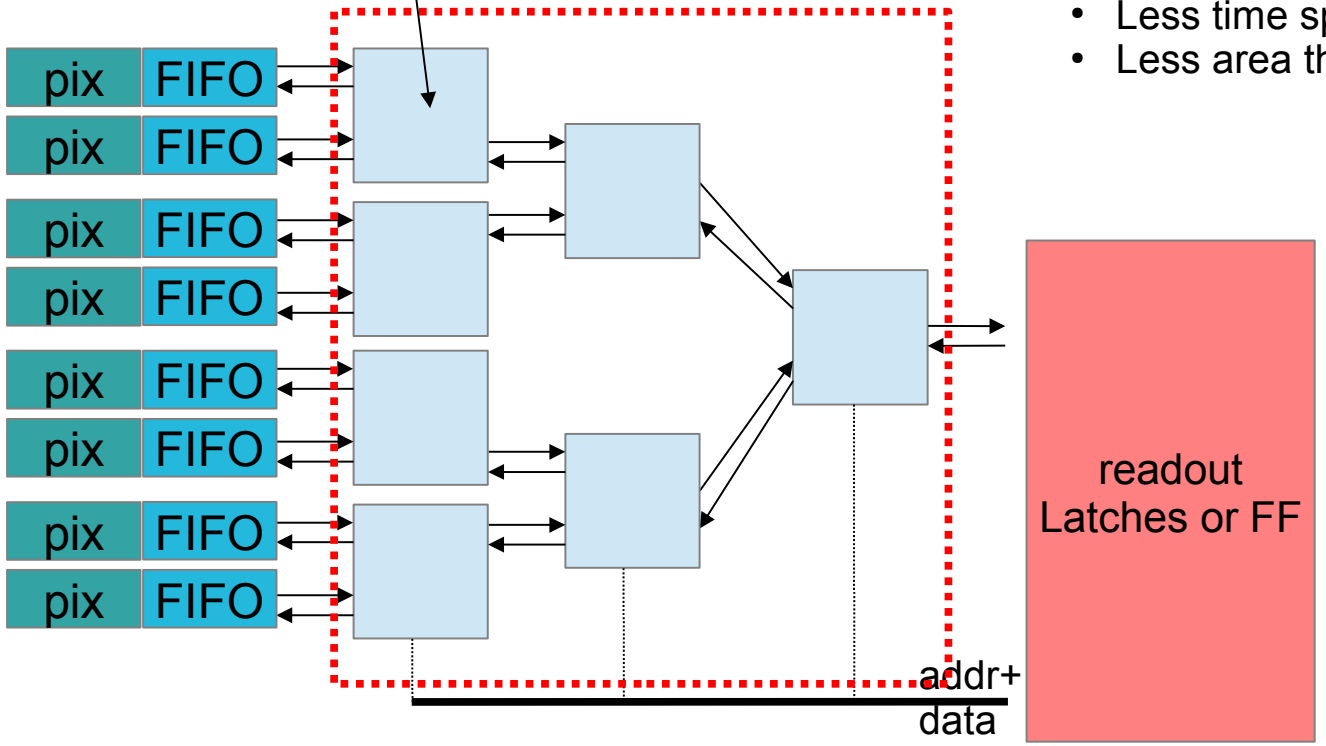


Basic building block

First implementation of the MEPHISTO binary readout architecture for strip detectors P.Fischer NIM-A (2001)

Build a library of logic blocks specialized for **SET-tolerant** combinatorial networks (ie. readout applications)

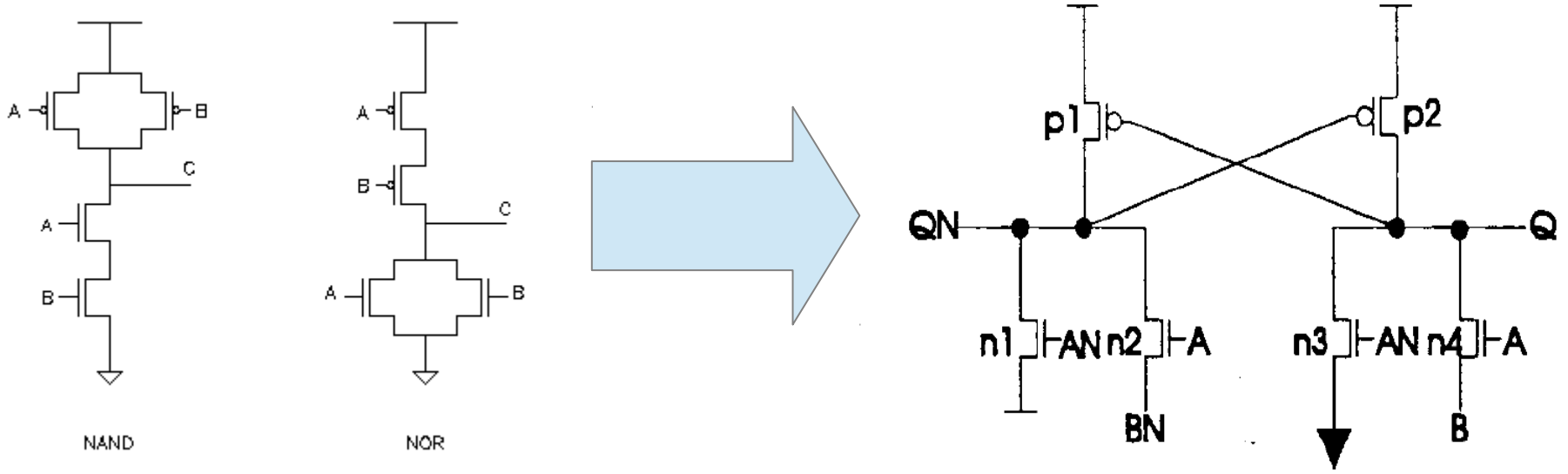
- Std cell kit compatible
  - Same size, same rules, abuttable to other std cells in the kit
  - Usable by automatic place&route programs (ie. Cadence Encounter)
- Optimized
  - Compact and regular designs
  - Less time spent on routing
  - Less area than triplication with standard cells



Pixel address and additional data can be multiplexed using the sel signal

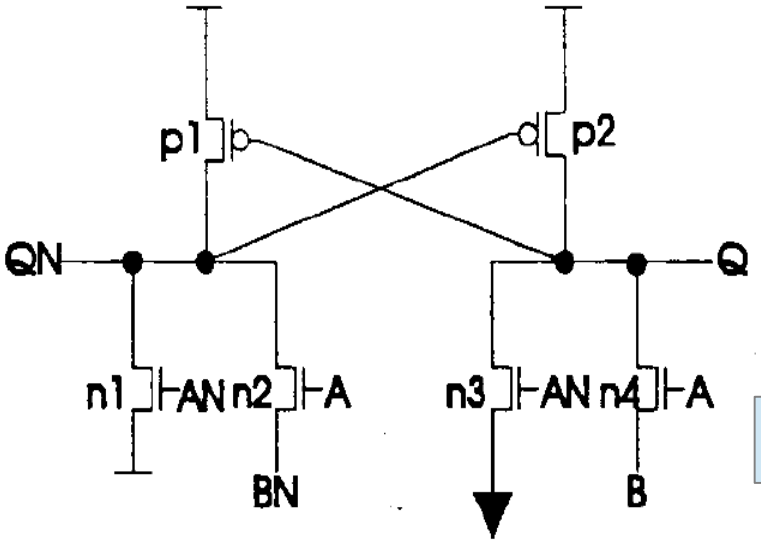
Standard CMOS vs DCVSPG

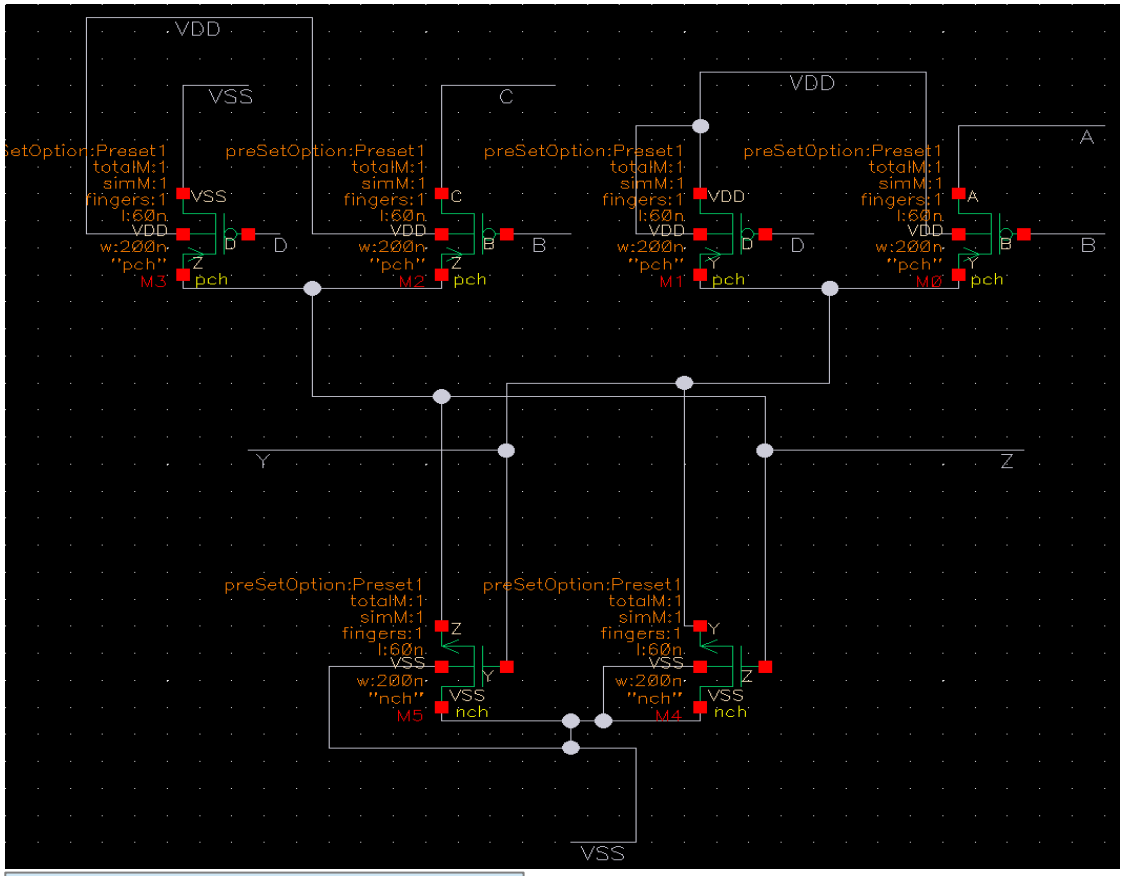
DCVSPG → Differential Cascode Voltage Switch with the Pass Gate logic tree



- Logic function is implemented with two complementary pass-gate trees using nMOS (for complex logic they might also share transistors)
- Circuit load is done by cross-coupled pMOS
- **2-rail encoding**: provides always output and negated-output, needs always both inputs and negated-inputs
- Typically provides better performances wrt standard CMOS logic (**power, delay**)
- Well suited for radhard applications: **SET are always non-logic states** and propagate less in the combinatorial logic (Hatano, H., "Single Event Effects on Static and Clocked Cascade Voltage Switch Logic (CVSL) Circuits," Nuclear Science, IEEE Transactions on , vol.56, no.4, pp.1987,1991, Aug. 2009)
- It's more difficult to handle with commercial synthesis and P&R tools

Standard DCVSPG → pMOS DCVSPG





- Pass Gate logic tree done with pMOS, cross-coupled load done with nMOS
- **In radhard applications pMOS shows less increase of leakage current with dose**
- Cell area is bigger and it's a bit more difficult to route

Guard rings?

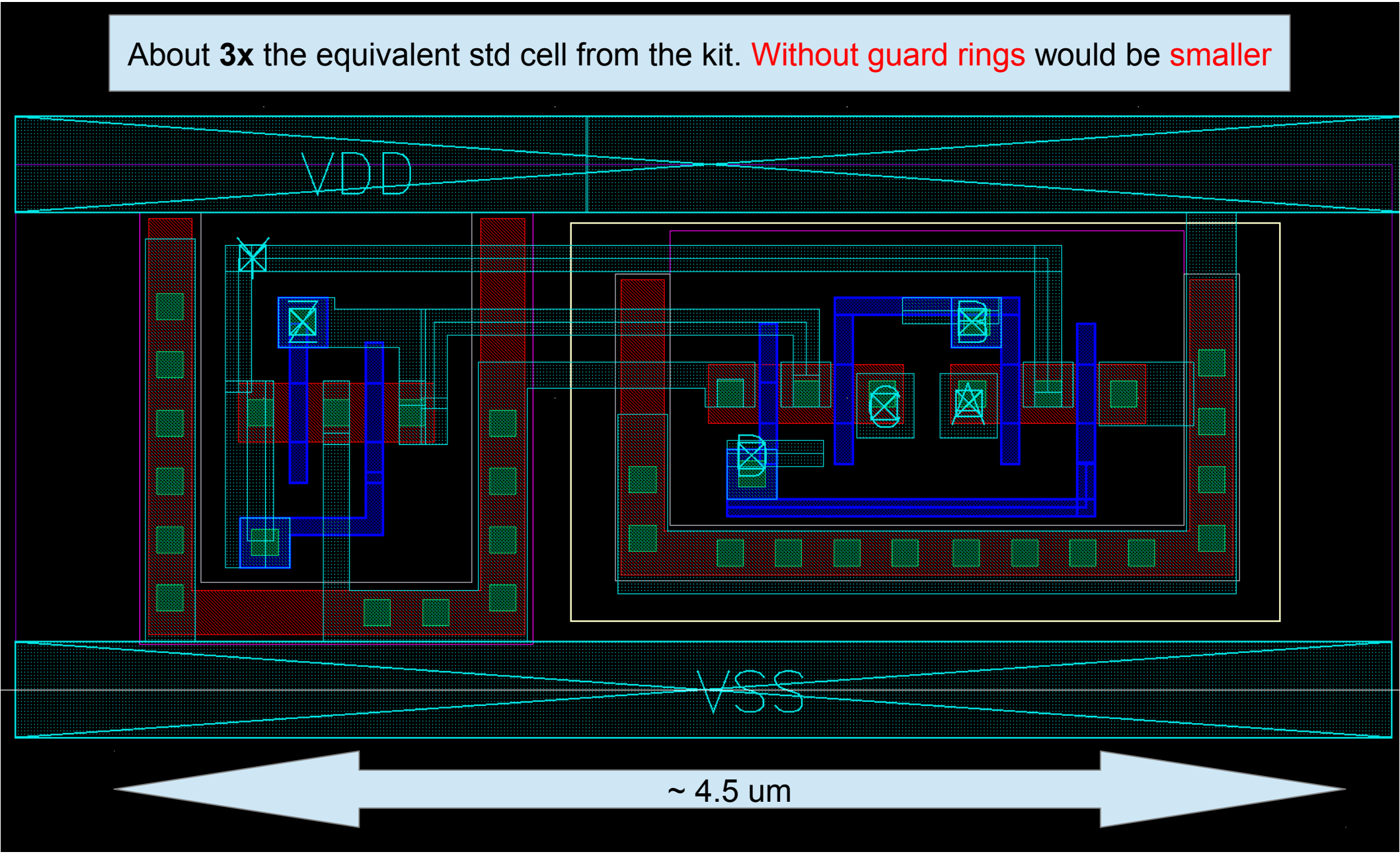
Protect from latch-up, But not really useful at 1.2-1V. Useful at higher Voltage.

ELT for nMOS?

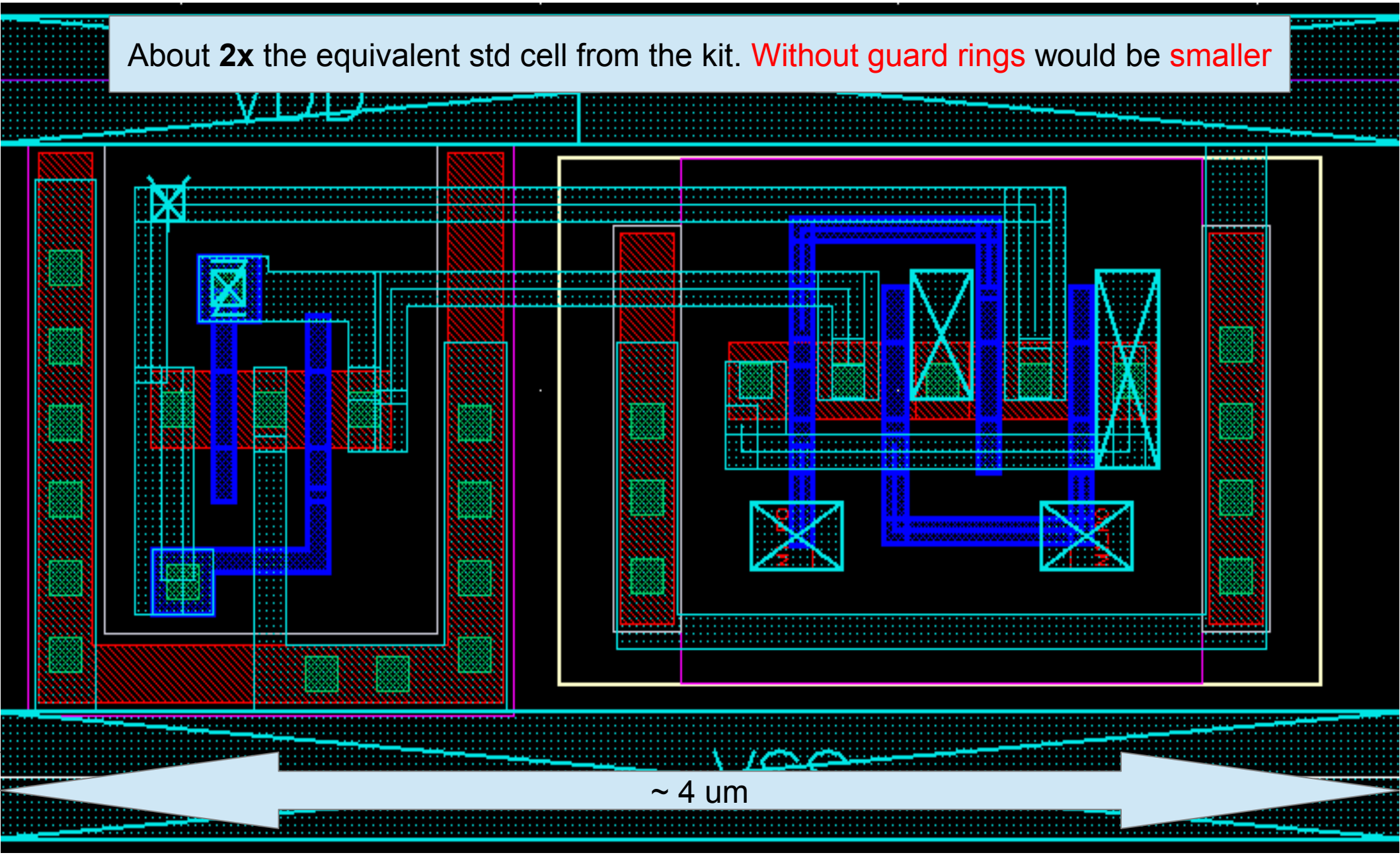
Not possible with current 65 nm rules

2 INPUT AND-like logic layout

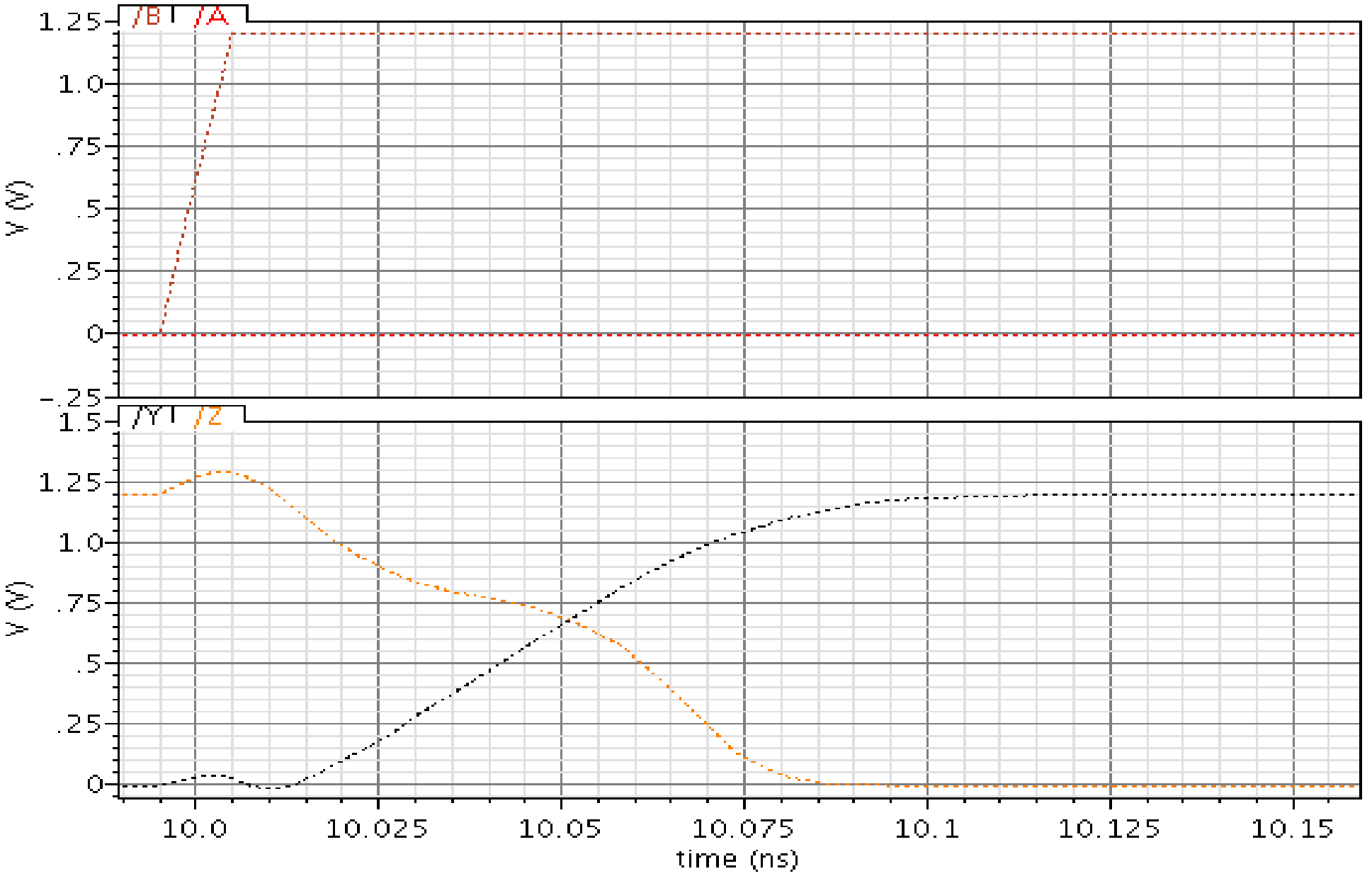
About **3x** the equivalent std cell from the kit. Without guard rings would be smaller



2 INPUT XOR-like logic layout



2 INPUT AND-like timing



Synthesis and P&R issues

- Software tools are not very good (yet) with differential/2-rail logic
- A possible flow to overcome software limitations is
  - Develop a 'fake' non-differential library with AND/NAND/OR/NOR/BUF/INV/LATCH/FF/etc
  - Convert with a script the netlist to the 'real' library with 2-rail inputs/outputs connected
    - Write a custom script is not difficult, however many programs to do that exists in the async logic community

Readout tree synthesis test

Priority encoder + address generator + mux tree (8-bit): 1394 AND-like cells

Synthesis done with RTL Compiler + awk/sed/python scripts to add differential wires

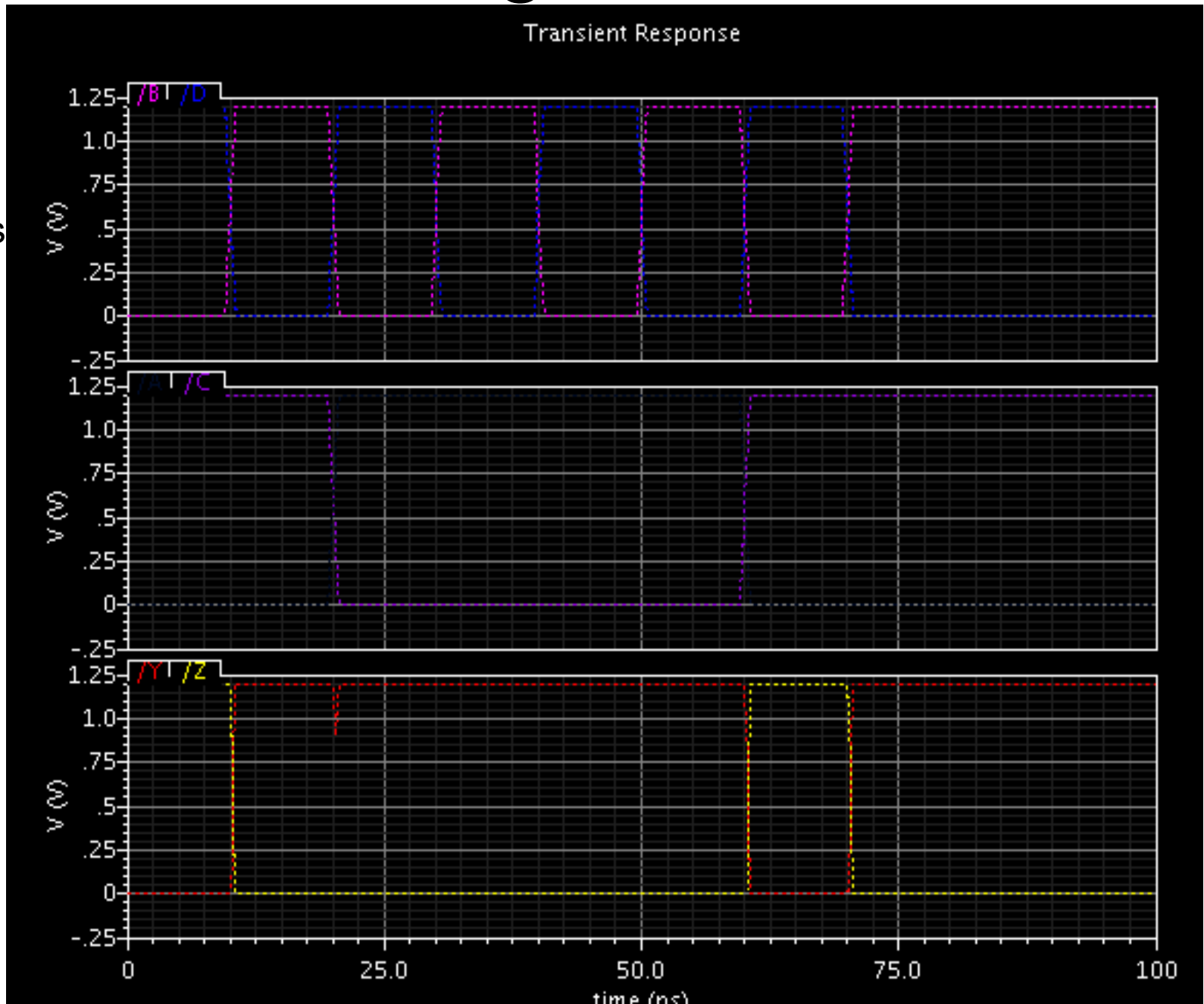
Status & conclusions

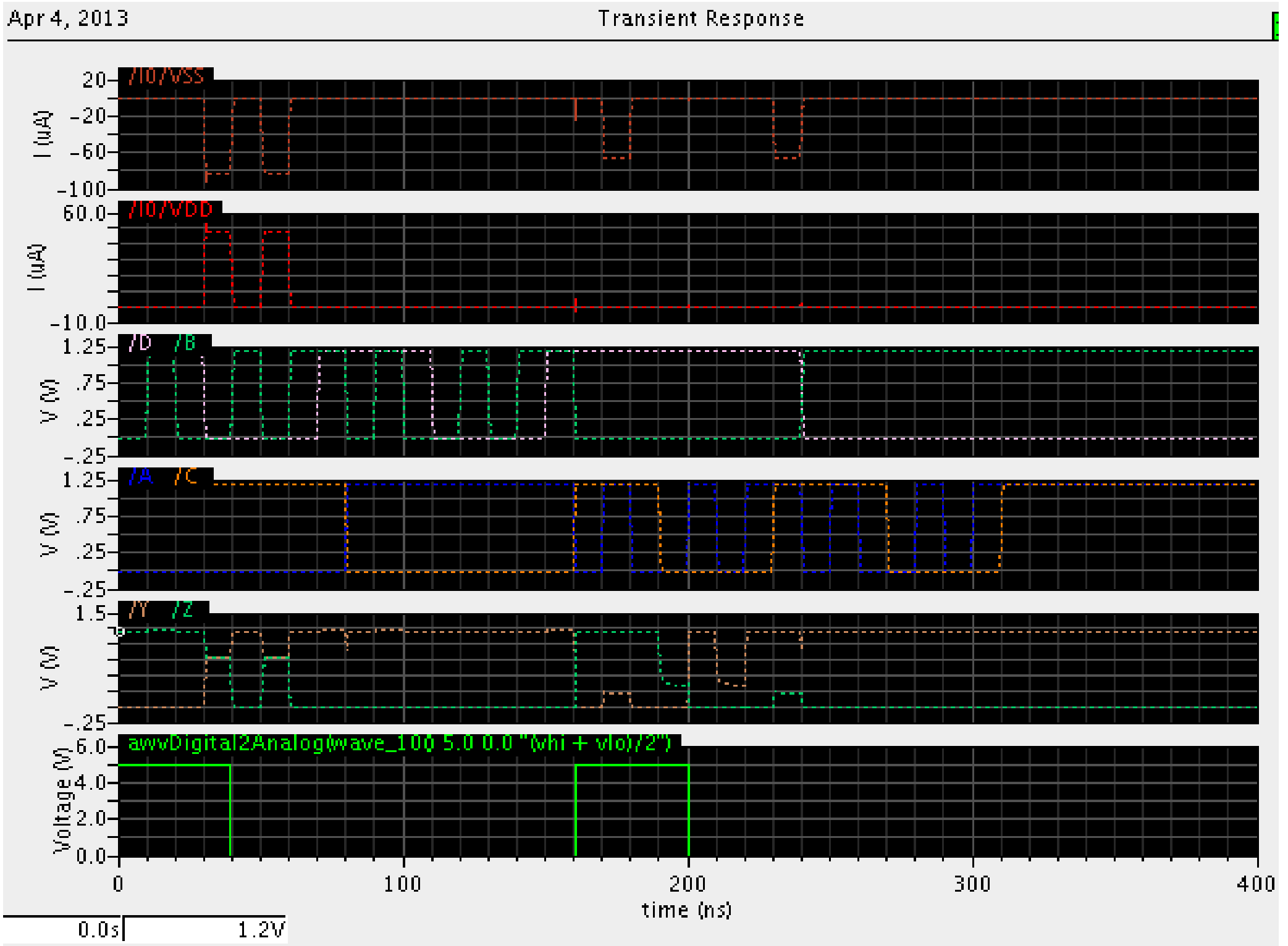
- Dual cascode switch logic with pass gate is an interesting candidate to build SET-tolerant logic
  - Few cells make a complete library → easier to maintain
- Schematics for AND-like, XOR-like, latch and master/slave ff DONE
- Layout with guard rings for AND-like and XOR-like DONE
  - TODO: layouts for latch and ff
  - TODO: layouts without guard rings
  - TODO: explore other layout techniques for dose resistance (are ELT really not possible?)
- Initial cell characterization in progress. The plan is to learn how to use Cadence automated tools for cell characterization in order to speedup comparison of different layouts
- A flow with custom scripts is possible and partially done.
  - TODO: explore tools for 2-rail async netlist conversion (similar to our case)
- TODO: this kind of logic opens interesting possibilities for low power. The cells might be used for NULL convention logic (async)

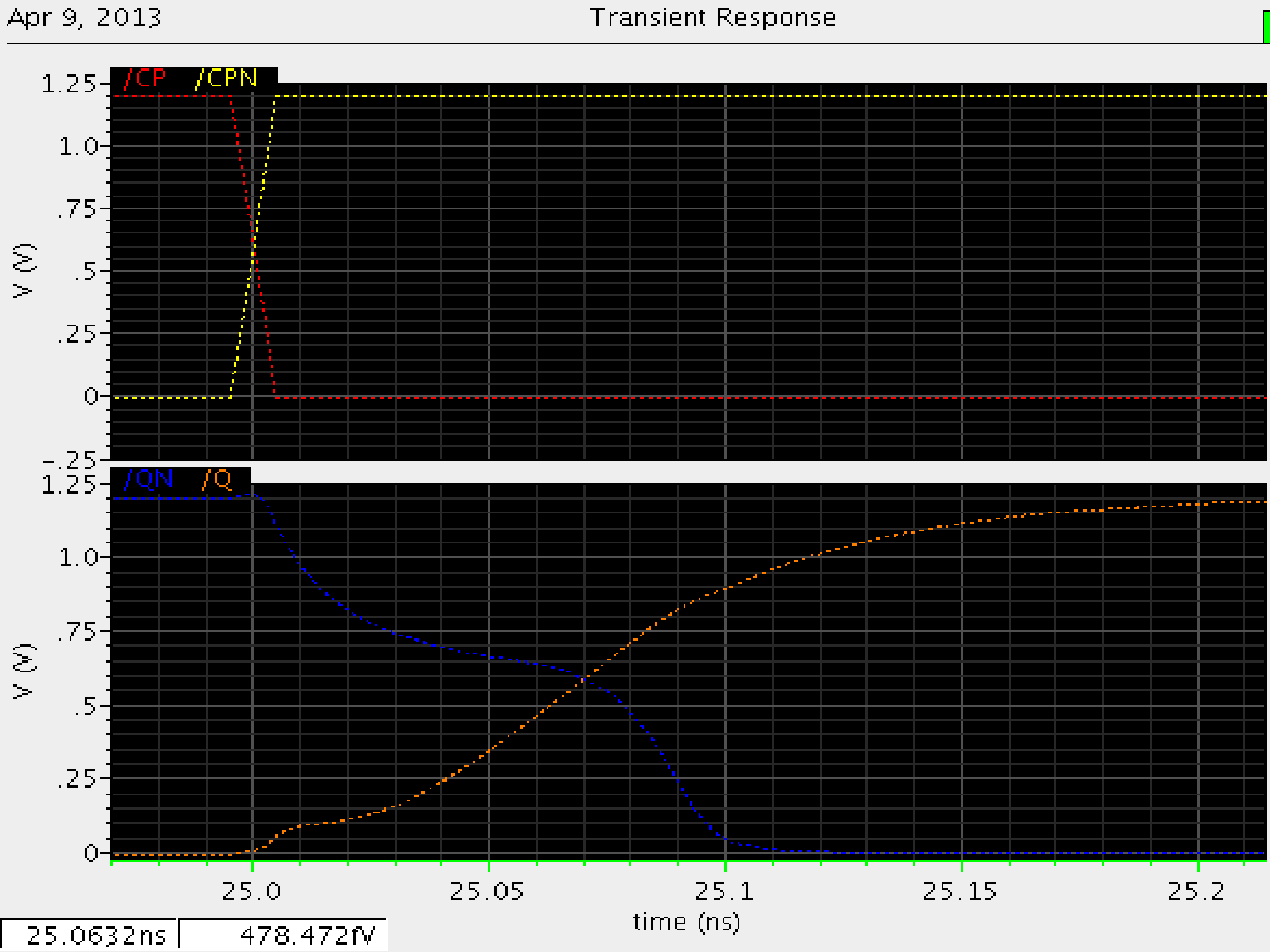
backup

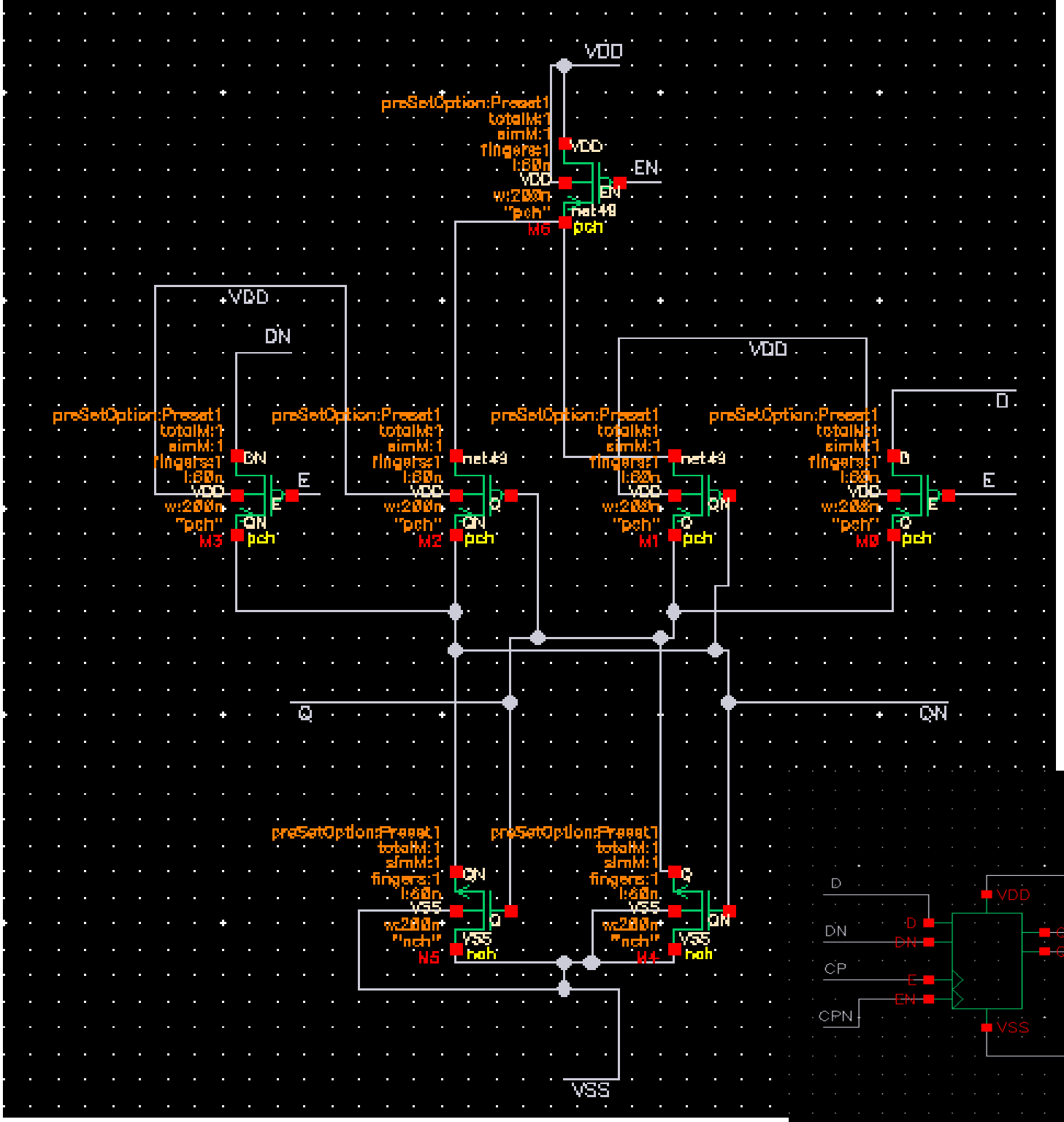
2 INPUT AND-like logic ~ simulation

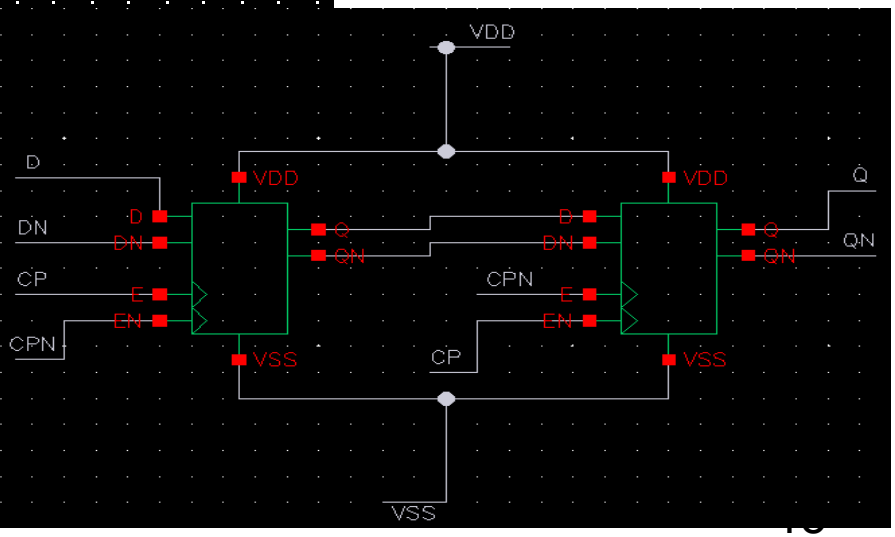
- A-C / B-D differential inputs
- Z-Y differential output
- Several logic functions with one layout:
  - Z = A or B
  - Y = A nor B
  - Z = C and D
  - Y = C nand D
  - ...




Apr 4, 2013
Transient Response
/I0/VSS
I (uA)
20
-20
-60
-100
/I0/VDD
I (uA)
60.0
-10.0
/D
/B
V (V)
1.25
.75
.25
-.25
/A
/C
V (V)
1.25
.75
.25
-.25
/Y
/Z
V (V)
1.5
-.25
awvDigital2Analog(wave_100 5.0 0.0 "(vhi + vlo)/2")
Voltage (V)
6.0
4.0
2.0
0.0
0
100
200
300
400
time (ns)
0.0s
1.2V


Apr 9, 2013
Transient Response
/CP
/CPN
1.25
1.0
.75
.5
.25
0
-.25
V (V)
/QN
/Q
1.25
1.0
.75
.5
.25
0
V (V)
25.0
25.05
25.1
25.15
25.2
time (ns)
25.0632ns
478.472fV


VDD
preSetOption:Preset1
totalM:1
simM:1
fingers:1
l:60n
w:200n
"pch"
M6
pch
EN
net49
DN
D
E
Q
QN
M3
M2
M1
M0
preSetOption:Preset2
totalM:1
simM:1
fingers:1
l:60n
w:200n
"nch"
M5
M4
nch
VSS


VDD
D
DN
CP
CPN
VSS
Q
QN