Layout-Dependent Proximity Effects in Deep Nanoscale CMOS

John Faricelli – April 16, 2009

Acknowledgements

This work is the result of the combined effort of many people at AMD and GLOBALFOUNDRIES.

AMD – Alvin Loke, James Pattison, Greg Constant, Kalyana Kumar, Kevin Carrejo, Joe Meier, Yuri Apanovich, Victor Andrade, Bill Gardiol, Steve Hejl

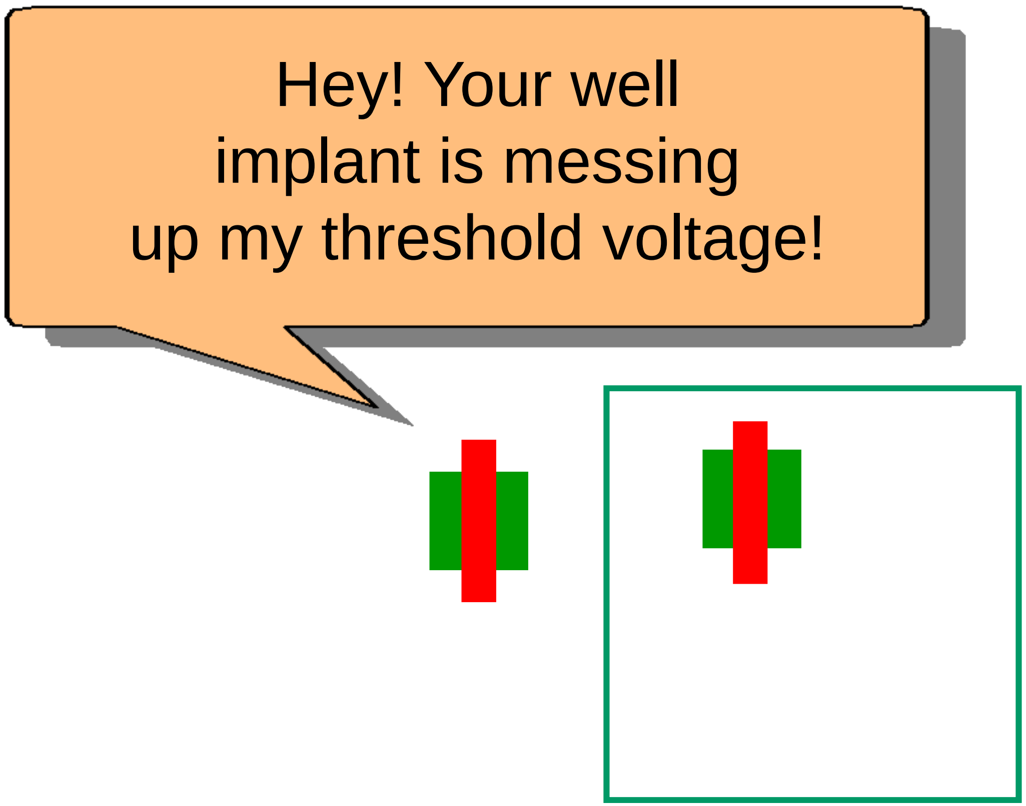
GLOBALFOUNDRIES – Akif Sultan, Sushant Suryagandh, Hans VanMeer, Kaveri Mathur, Rasit Topologlu, Uwe Hahn, Thorsten Knopp, Sean Hannon, Darin Chan, Ali Icel, David Wu

Outline

- Layout-dependent proximity effects
- Modeling philosophy
- CAD tools
- Mitigation of layout-dependent stress effects

Layout-dependent proximity effects

Nanoscaled CMOS devices are so close to each other that they begin to interact.



Why should I care about this?
It's modeled in SPICE…

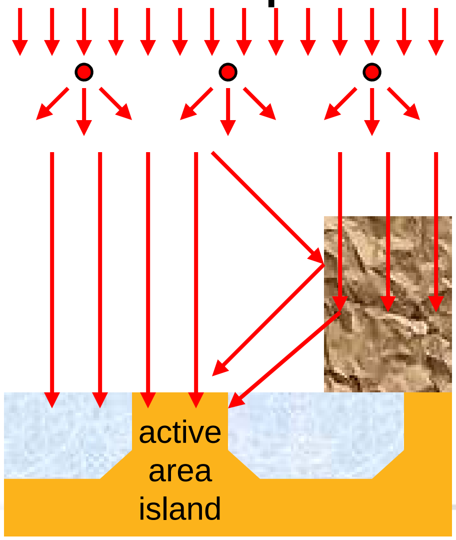
- Proximity effects can de-rate FET current by 10% (or more), or shift threshold by several 10's of mV.
- De-rating factors can only be calculated after layout extraction,
  i.e., *ignored* in schematic-extracted netlists.
- Need to pay attention during layout to minimize proximity effects and discrepancy between layout- & schematic-extracted sims.
- Otherwise…
  more layout rework &
  SCHEDULE IMPACT !!!

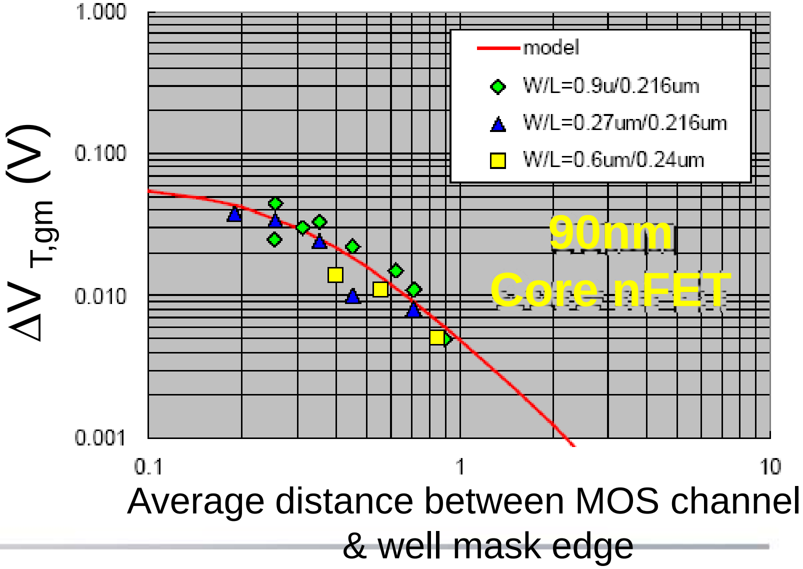
Sources of layout proximity effect

- Well proximity effect
- Unintentional stressors
  - Shallow trench isolation (LOD effect)
- Intentional stressors
  - Dual-stress liners
  - Embedded SiGe

Well proximity effect

- $|V_T|$ ↑ if FET is too close to resist edge due to dopant ions *scattering* off resist sidewall into active area during well implants
- $|\Delta V_T|$ depends on:
  - FET channel distance to well mask edge
  - Implanted ion species/energy
- Other effects: μ ↓, L_{eff} ↑, $R_{extension}$ ↑ → I_{dsat} ↓
- Well mask symmetry now critical for FET matching





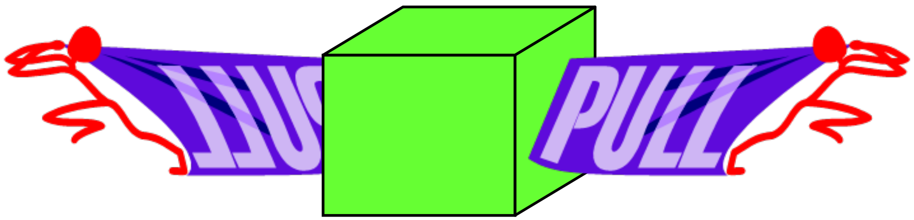
Source: TSMC (CICC 2005).

A brief review of stress and strain…

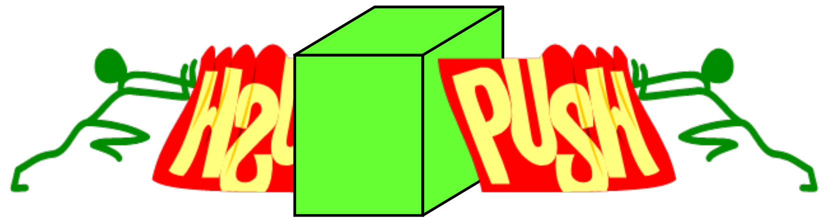
Stress / strain definitions

$$Stress(\sigma) = \frac{Force}{Area} \qquad Strain(\varepsilon) = \frac{\Delta \ell}{\ell_0}$$

Tension (positive stress) vs. Compression (negative stress)

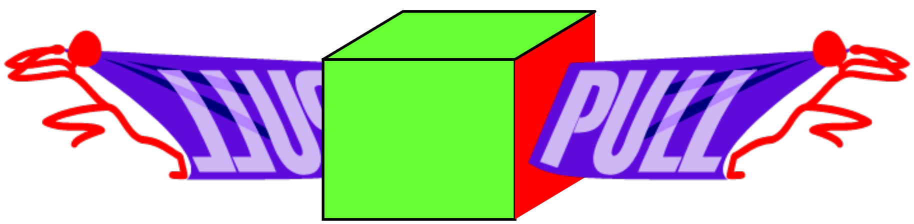


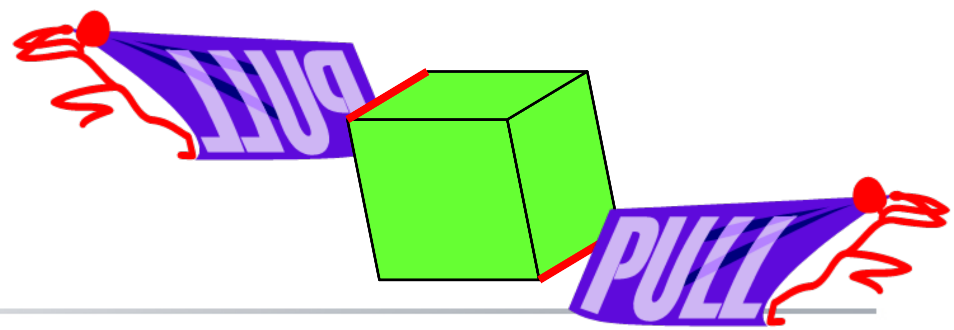
atomic spacing > equilibrium spacing



atomic spacing < equilibrium spacing

Normal Stress (on-axis) vs. Shear Stress (off-axis)



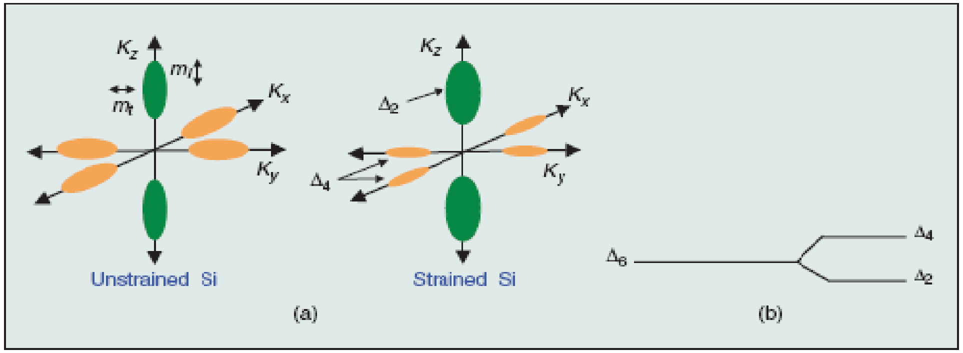


Stress affects carrier mobility

Compression or expansion of silicon lattice causes

- Changes shapes of bands → changes carrier effective mass
- Shifts relative position of band energy → redistributes carriers to different bands

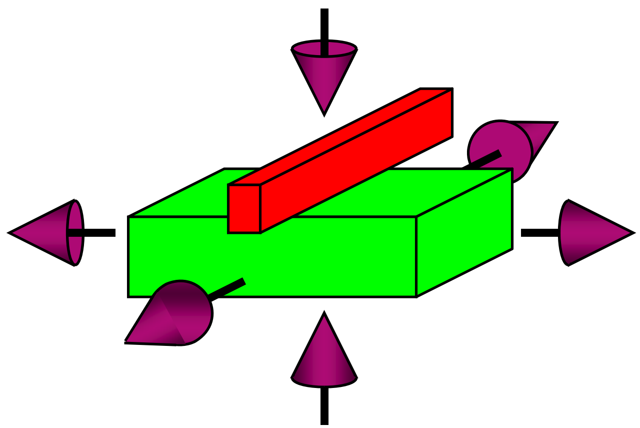
Net effect is change in carrier mobility → current!



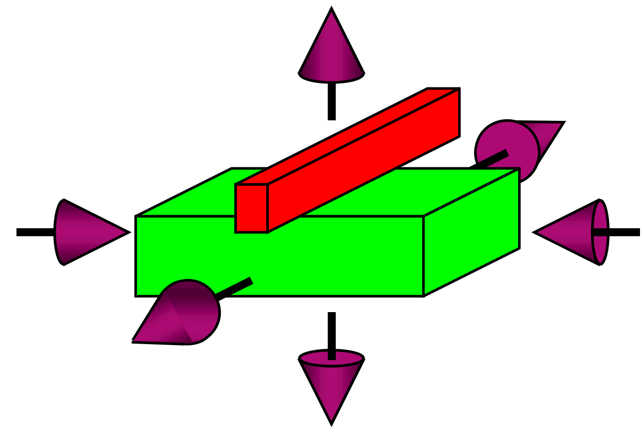
Source: N. Mohta and S. Thompson, IEEE Circuits and Devices, Sep/Oct 2005.

Desired stress orientations

Desired nFET strain



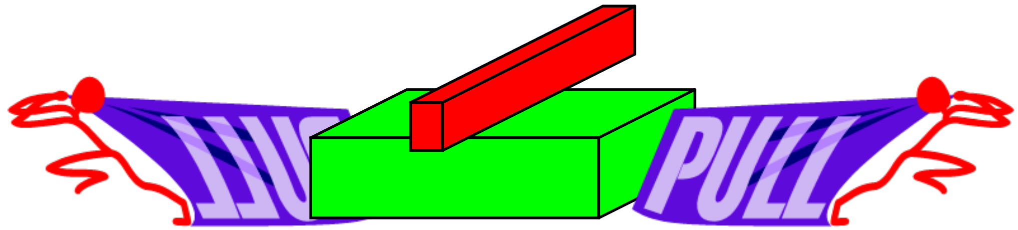
Desired pFET strain



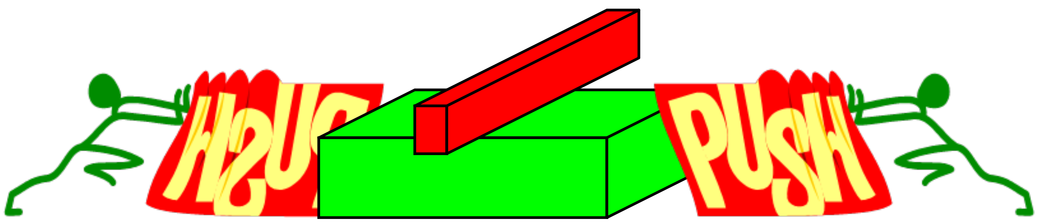
- Net mobility factor (FET performance improvement factor) is a very complicated function of stress *tensor*
- Can apply substrate-induced bi-axial vs. uni-axial strain to improve FET performance of both nFET and pFET

Uni-axial strain

Tension (stretch atoms apart) → faster nFET



Compression (squeeze atoms together) → faster pFET



- Increase I_{ON} for the same I_{OFF} without increasing C_{OX}
- Want 1-4GPa (high-strength steel breaks at 0.8GPa)
- Uni-axial strain along channel length is main effect to consider, but strain along other directions are important too

Source of stress…

- Un-intentional
  - Shallow trench isolation (nFET & pFET)

    → compressive

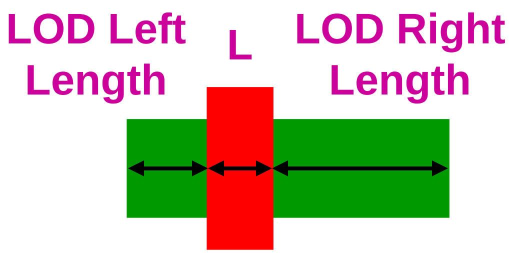
- Intentional
  - Stress memorization (nFET)
  - Dual-stress liners (nFET & pFET)

    → tensile & compressive
  - Embedded SiGe (pFET only)

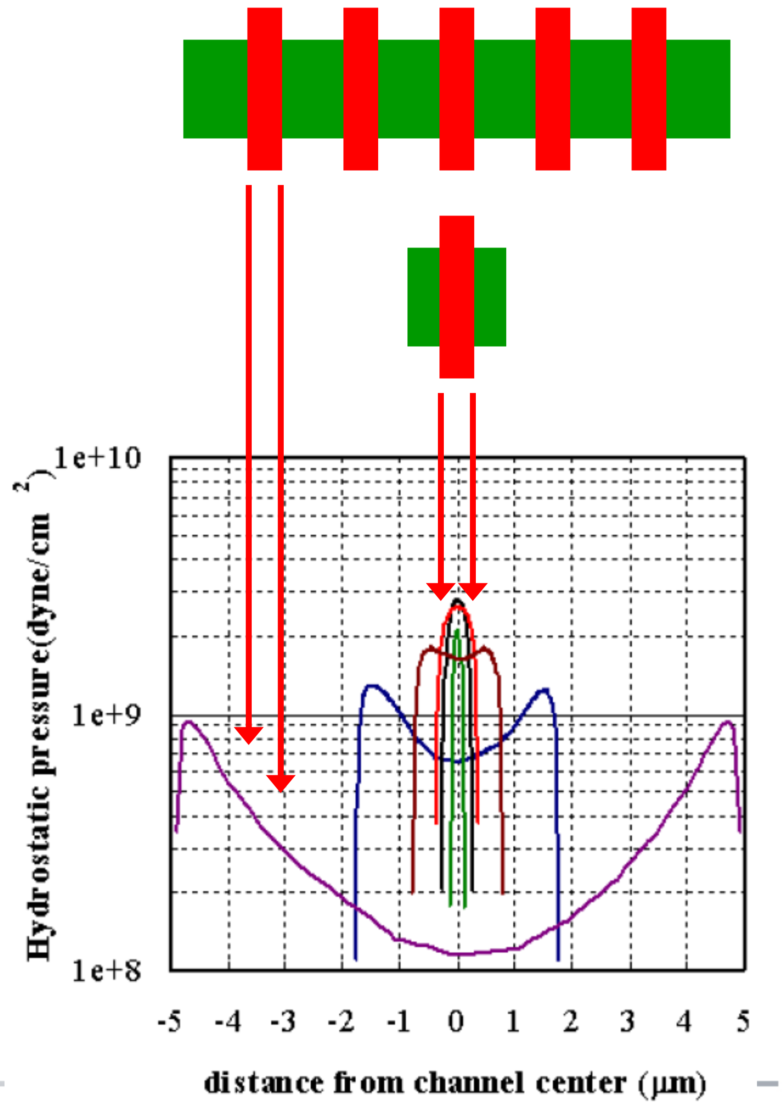
    → compressive

Shallow trench isolation (LOD effect)

- LOD left length, L, & LOD right length specify where channel is located along active area

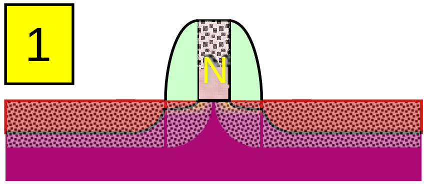


- Compressive stress *degrades* NMOS
- Net strain depends on both left and right extents of LOD

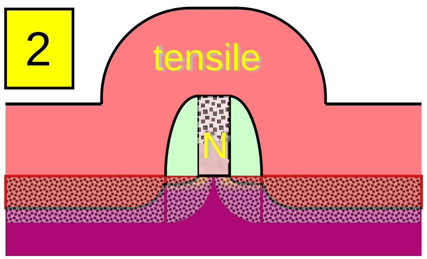


Source: Xi *et al.*, UC Berkeley (2003).

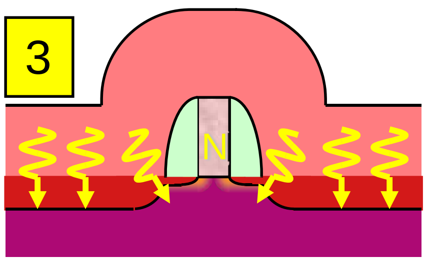
Stress memorization (NMOS)



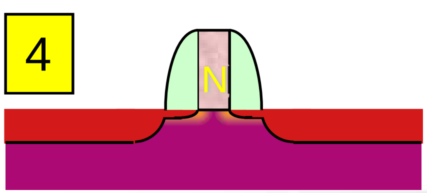
Amorphize poly & diffusion with silicon implant



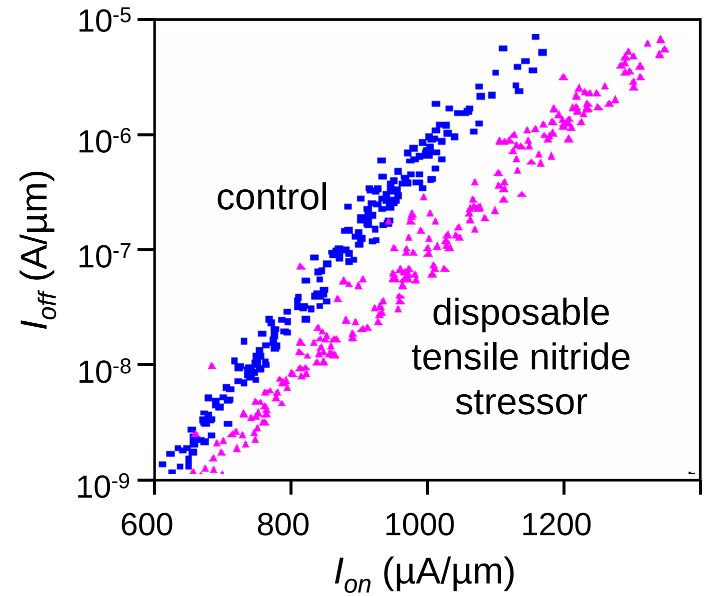
Deposit tensile nitride



Anneal to *make nitride more tensile* and transfer nitride tension to crystallizing amorphous diffusion



Remove nitride stressor (tension now frozen in diffusion)

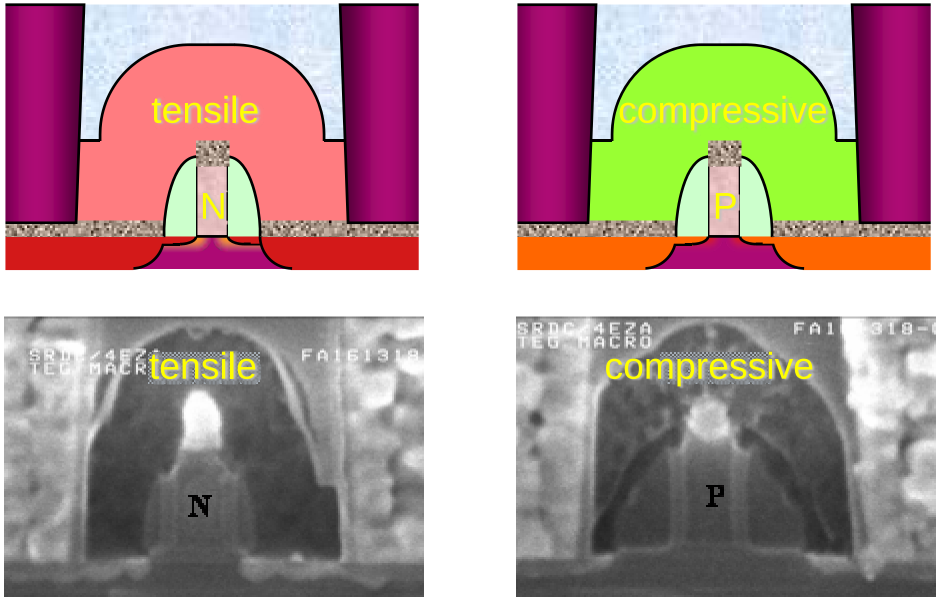


Source: Chan, IBM (CICC 2005).

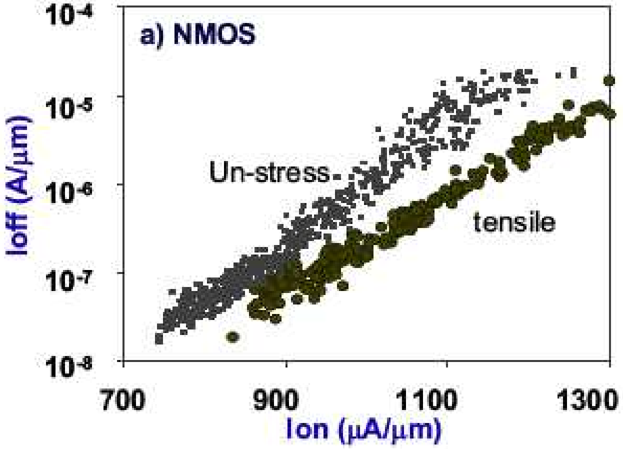
Dual-stress liners

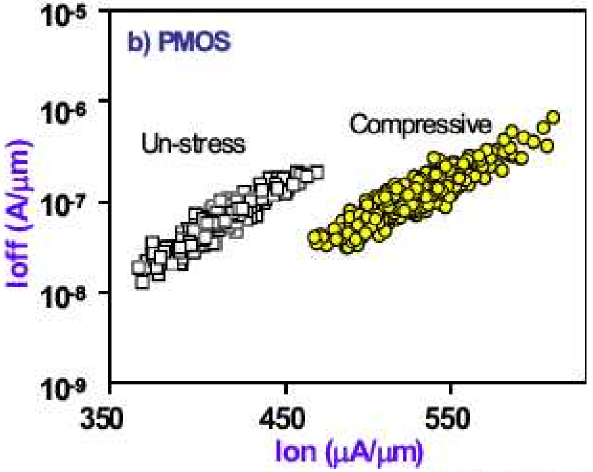
- Deposit tensile/compressive PECVD silicon nitride liners over device
- Liner stress state is function of gas flows & ratios during liner deposition
- PEN = plasma-enhanced nitride

TPEN for nFET *CPEN for pFET*



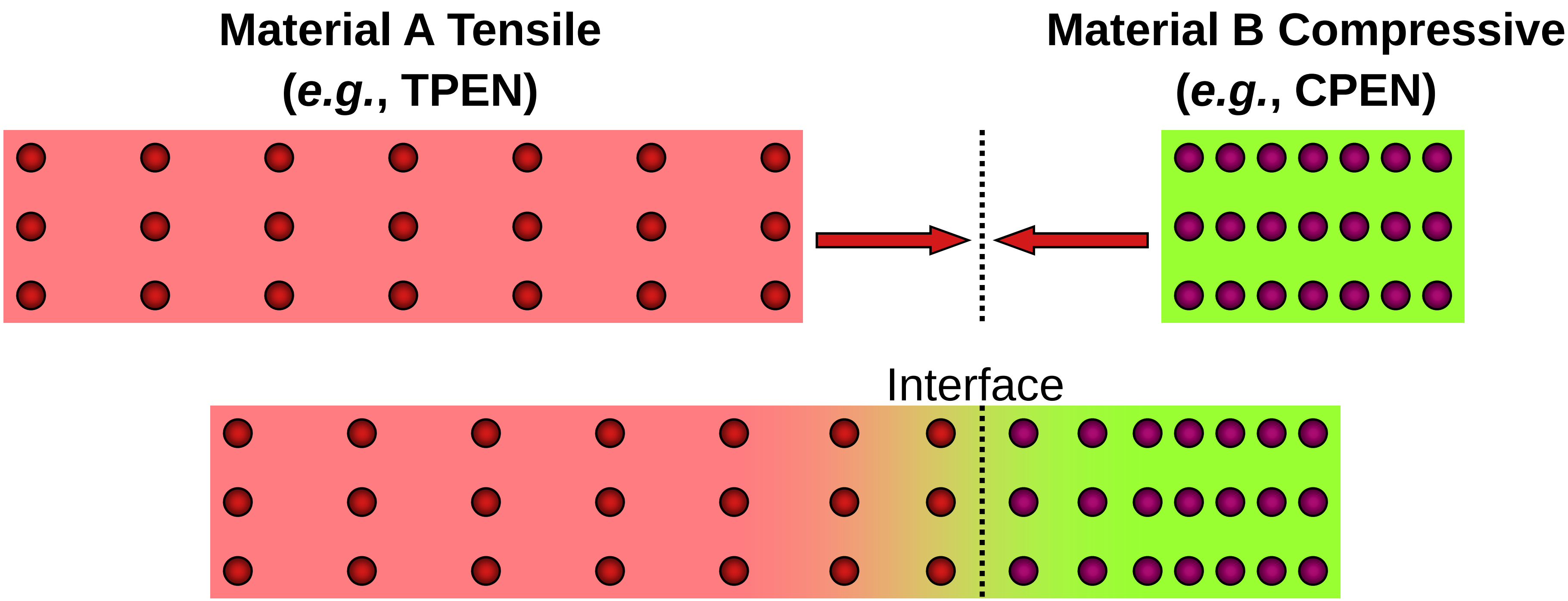
Source: Yang (IEDM 2004).





Stress variation due to stress liners

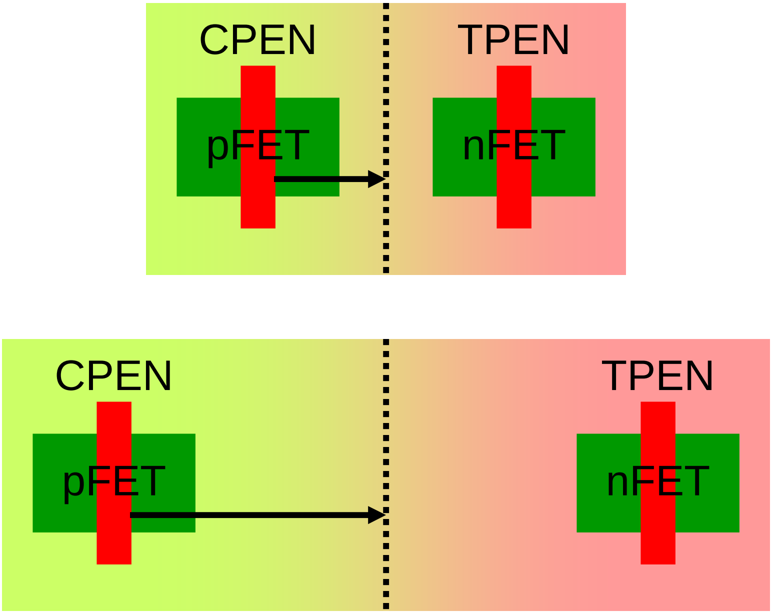
When materials of different strain come together…

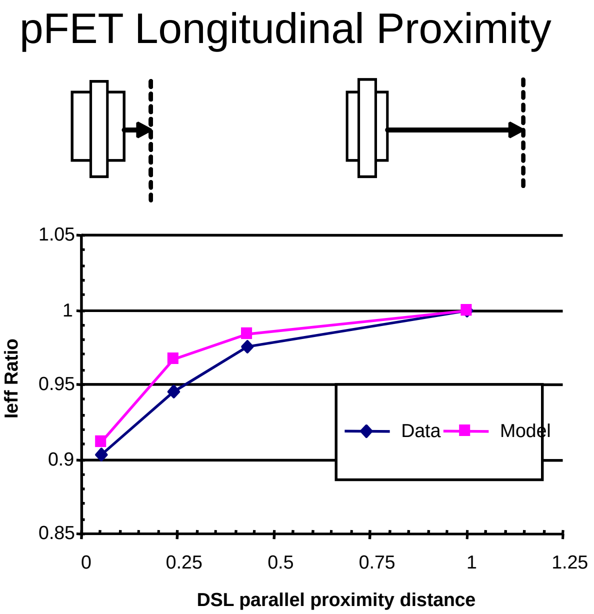


- Both materials will relax at the interface
- Extent of relaxation is gradual & depends on distance from interface
- There is no relaxation far away from the interface

Longitudinal proximity

- Having opposite device nearby in longitudinal direction reduces impact of stress liner, hence *mutually slow each other down*
- Opposite PEN liner absorbs/relieves stress introduced by PEN liner



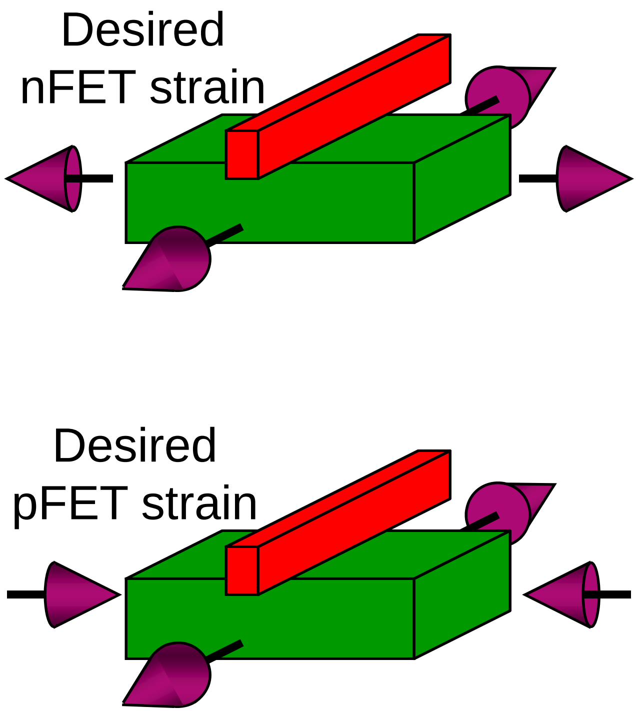


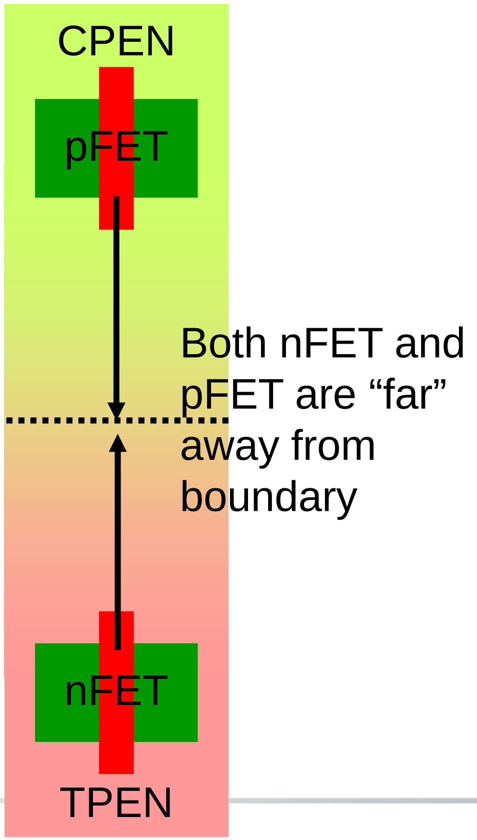
Source: Sultan ISQED (2009).

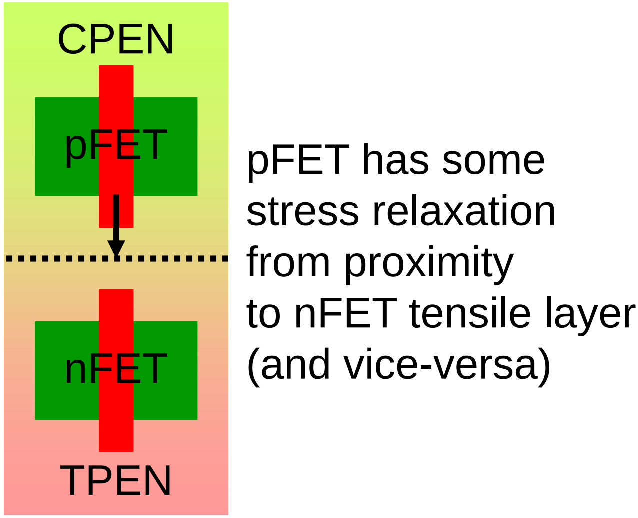
The future is fusion

Transverse proximity

- Both nFET & pFET like tension in transverse direction, unlike longitudinal direction (nFET wants tension, pFET wants compression)
- Recall TPEN & CPEN film stress is isotropic
- nFET near pFET in width direction *helps pFET but hurts nFET*

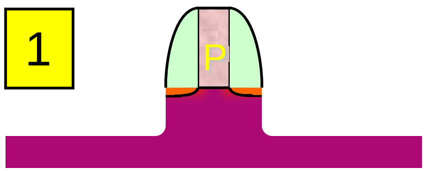




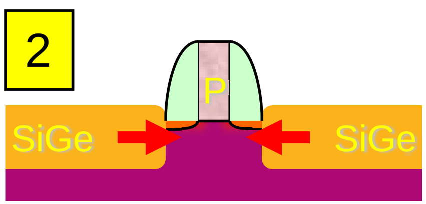


Embedded SiGe

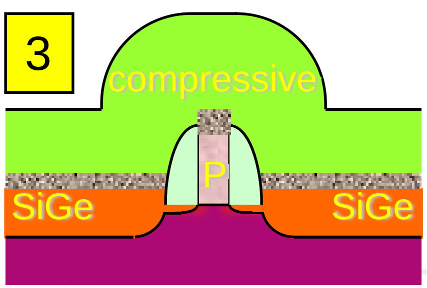
S/D laterally compresses channel since SiGe has higher lattice constant than Si (SiGe constrained to Si lattice will be in compression)



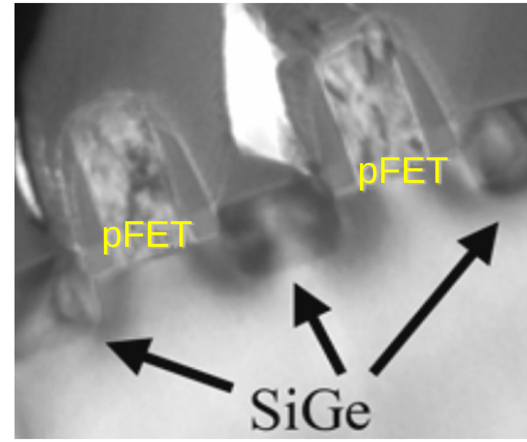
Etch source/drain recess



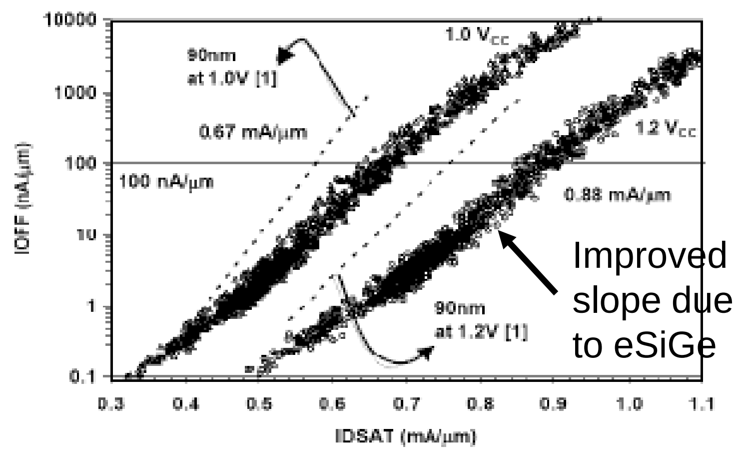
Grow SiGe epitaxially in recessed regions



Build source/drain regions & deposit CPEN



Source: Ouyang (VLSI Symp 2005).

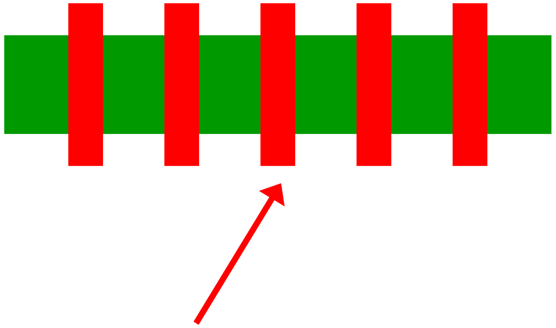


Source: Bai (IEDM 2004).

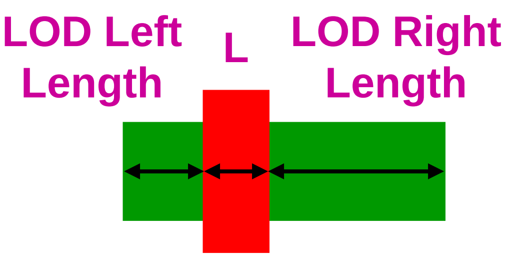
Stress variation to amount of eSiGe

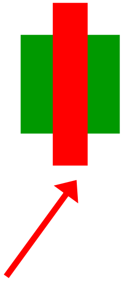
- Volume of eSiGe affects the amount of stress that each device sees

- Size of active area controls volume



This device finger is in a region of higher eSiGe volume → higher current





This device has less eSiGe volume →lower current

Modeling philosophy

Two scenarios:

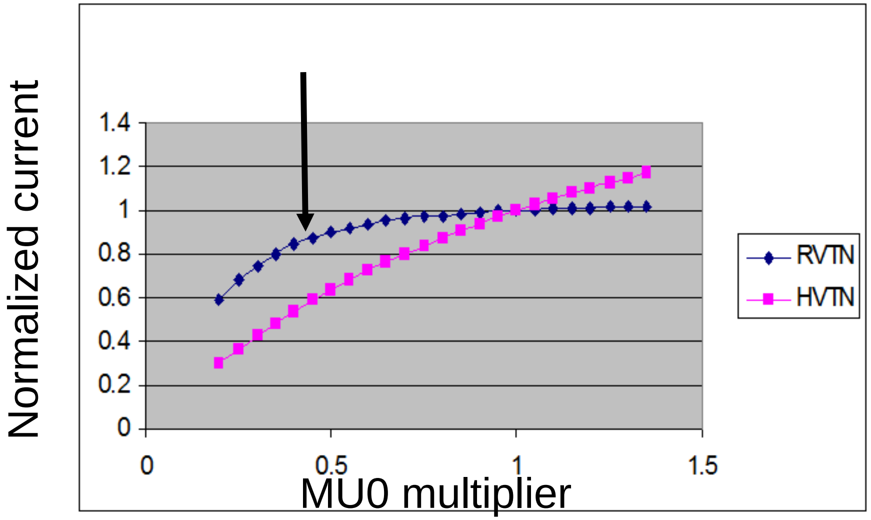
- PhD thesis approach – model everything possible
- “Good enough” approach – model the most important effects and try to get those “right”

Scenario one: “PhD thesis” approach

- Model every possible layout dependency
  - Example: 30 or more measurements per FET finger
  - Need test structures for all of these measurements
  - Need to measure and characterize test structures
  - Model requires modifying several BSIM model parameters on a per-finger basis
- Resultant model is complicated, specific to particular MOS model, hard to fit, costly to measure in LVS, and not very transparent
- Likely to have unexpected interactions

Example of unexpected interaction

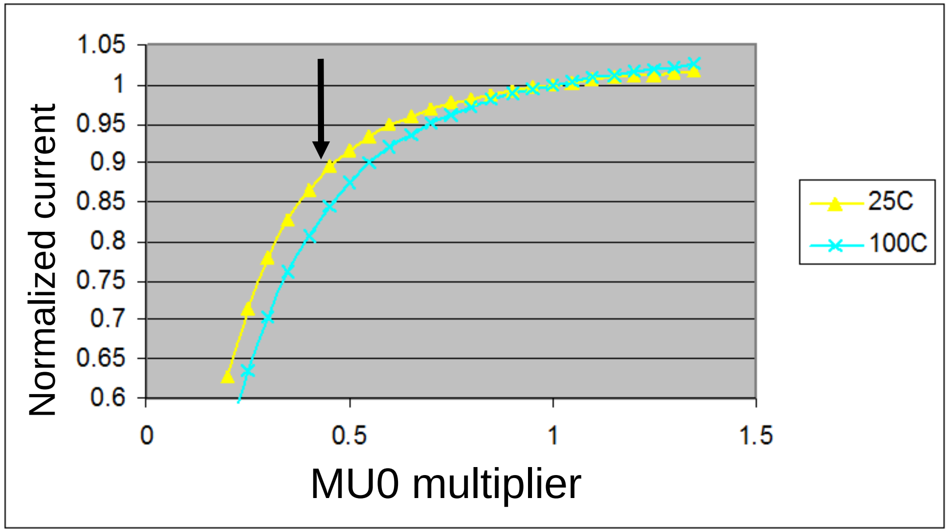
- First implementation of AMD stress model modified BSIM mobility parameter “MU0”
- Choice of BSIM model parameters resulted in a very non-linear relationship of drain current and mobility
- Had to greatly reduce mobility to get any effect on drain current



To get 10% degradation, have to reduce MU0 by 0.45 !!

Unexpected interactions (2)

- Small value of MU0 multiplier caused other problems
  - Non-physical temperature dependence
  - Non-physical dependence on channel length
  - …



At MU0 multiplier of 0.45, current degrades an additional 5% at 100°C

Scenario two: "Good enough" approach

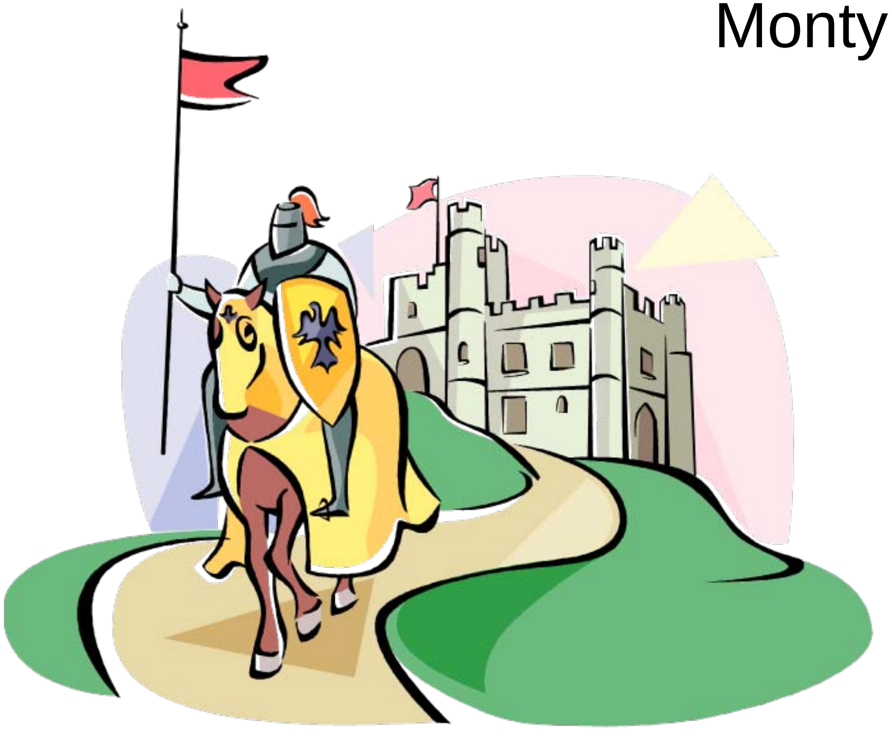
- Model only most important effects
- Use phenomenological approach – we measure changes in *drain current* and *threshold* on test structures
- Use hooks in circuit simulator to adjust drain current and threshold directly on per-instance basis
- Transparent - designer sees exactly what is happening to device
- Easy to debug, no interaction with choice of transistor model parameters
- Downside - not every physical effect can be modeled (maybe a good thing?)

CAD

Implementation

"It's only a model"

Monty Python and the Holy Grail



Multiple tools for evaluating proximity effects

- RC extraction/HSPICE/timing flow
- “Short flow” – Evaluate proximity effects during initial layout
- Stress rule checker – Calibre rule deck to point out “low hanging fruit”

RC extract flow

Inputs: gds and schematic netlist

↓

Calibre LVS extracts layout-dependent model distances for each FET finger

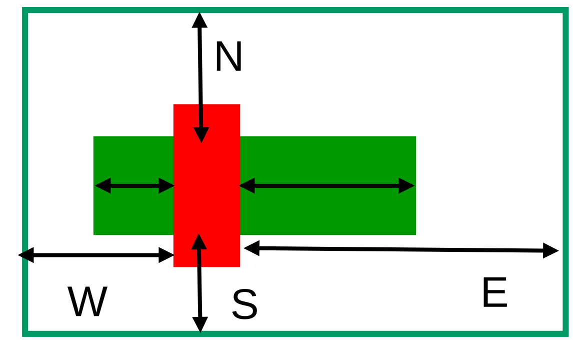
↓

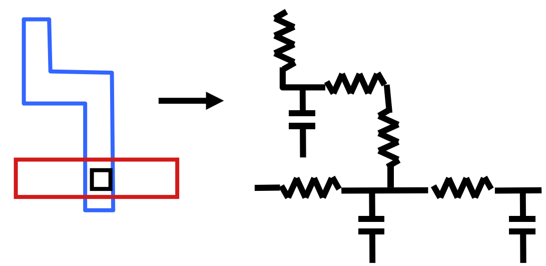
RC extract tool (QRC, StarRCXT, …)

↓

Stress model (in our case, a Perl module) → Extracted netlist is post-processed and stress model is evaluated →

DSL boundary





Each transistor finger has degradation/ enhancement factor MULID0*

M1 D G S B nFET … MULID0 = 0.95

* HSPICE 2009.03 MOS Model Guide

Stress short flow

Disadvantages of RC extract flow:

- Time consuming – may take many hours to run
- Layout should be LVS clean

A short turn-around flow was desired by the analysis and layout teams...

Stress short flow

Inputs: gds and schematic netlist

↓

Calibre LVS
extracts layout-dependent model distances for each FET finger

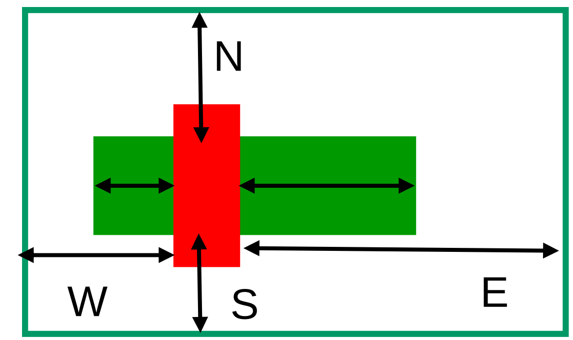
↓

Stress model (in our case, a Perl module) → **Stress model is evaluated** using Calibre measurements for each transistor finger

↓

- Histograms of distribution of MULID0
- Calibre RVE file for browsing results

DSL boundary



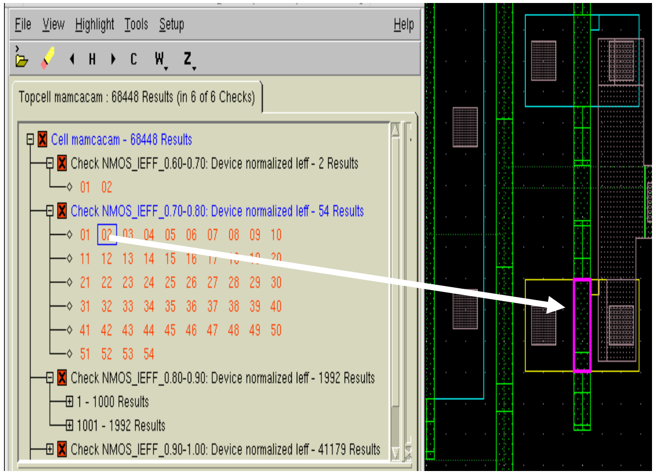
Stress short flow output

Histograms for quick overview

```
Ieff < 0.100 = 0
Ieff >= 0.100 and < 0.200 = 0
Ieff >= 0.200 and < 0.300 = 0
Ieff >= 0.300 and < 0.400 = 0
Ieff >= 0.400 and < 0.500 = 0
Ieff >= 0.500 and < 0.600 = 0
Ieff >= 0.600 and < 0.700 = 2
Ieff >= 0.700 and < 0.800 = 54
Ieff >= 0.800 and < 0.900 = 1992
Ieff >= 0.900 and < 1.000 = 41179
Ieff >= 1.000 and < 1.100 = 0
Ieff >= 1.100 and < 1.200 = 0
Ieff >= 1.200 = 0
```

- Short flow runs quickly, on the order of an LVS run (minutes)
- Can be run in –dirty mode, before LVS clean

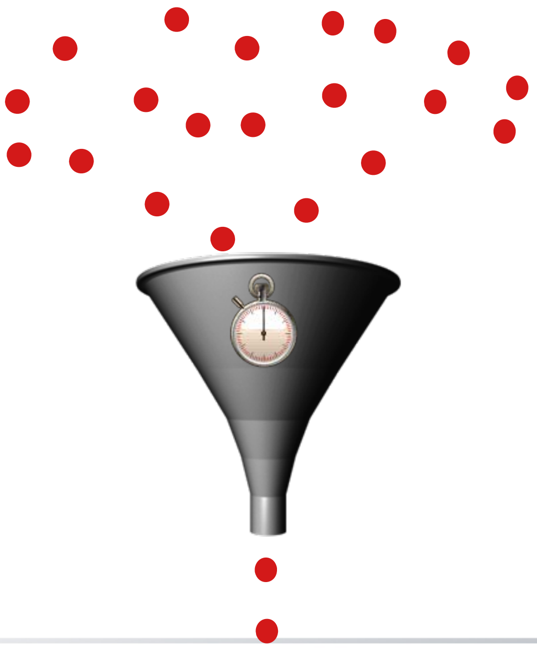
Calibre RVE file for browsing layout



Provides immediate feedback to layout designer on layout-dependent variation

Critical path filtering

- Short flow output can be further filtered using timing reports, which identify which devices are in the critical path
- This allows designer to focus re-layout effort on devices that matter

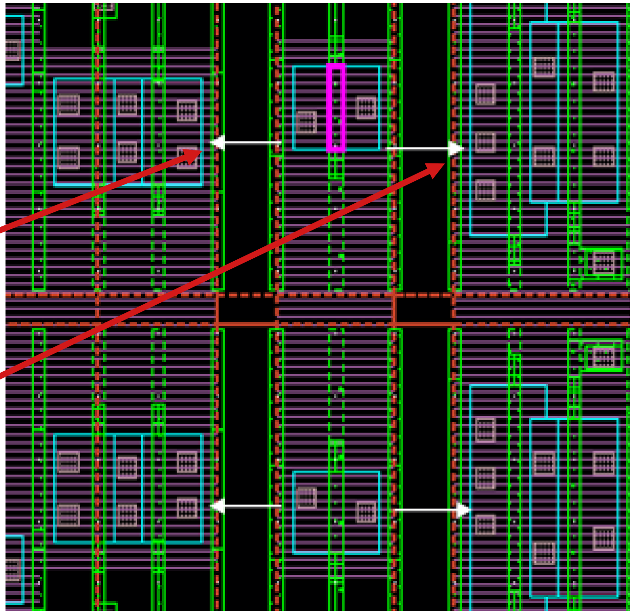


Note: timing filtering can only be done late in design

Stress rule checker

Stress rule checker is a Calibre-based tool to identify layouts that can be easily changed to reduce variation due to layout proximity effects

In this example, the stress rule checker identifies regions of n-well that should be joined in the horizontal direction



Guidelines for mitigation of layout-dependent effects

Mitigation guidelines come in two flavors:

- Minimize variation from base SPICE model
- Minimize variation between devices that need to be matched

We'll focus here on device matching…

Device matching guidelines

- Generic guidelines
  - Use similar active area (OD) shape, size, and orientation
  - Maintain similar distance from device gates to well implant edges
  - Add dummy devices and/or dummy poly over STI so that fingers at edge of shared OD area “look similar” to inner fingers

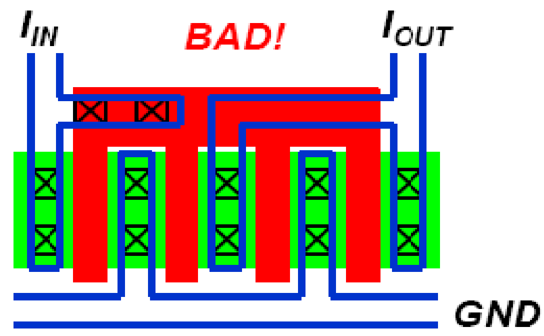
Matching guidelines (cont)

- Process-specific guidelines
  - Maintain similar distance from device gate to dual-stress liner interface
    - Enforce minimum distance so that device does not stray too far from nominal device
  - Keep NMOS and PMOS together in the same row
    - Avoid alternating NMOS and PMOS (DSL relaxation effect)
  - Minimum keep-away distance from well implant edge (well proximity effect)

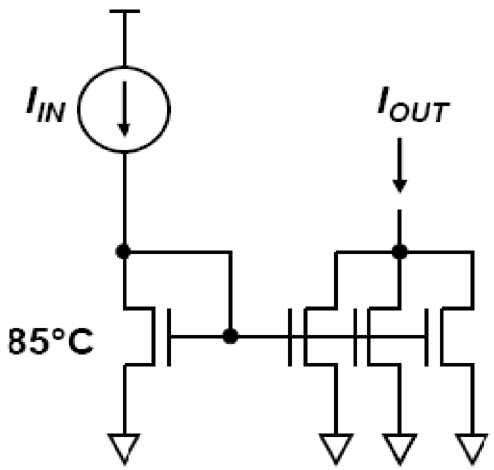
Device matching example

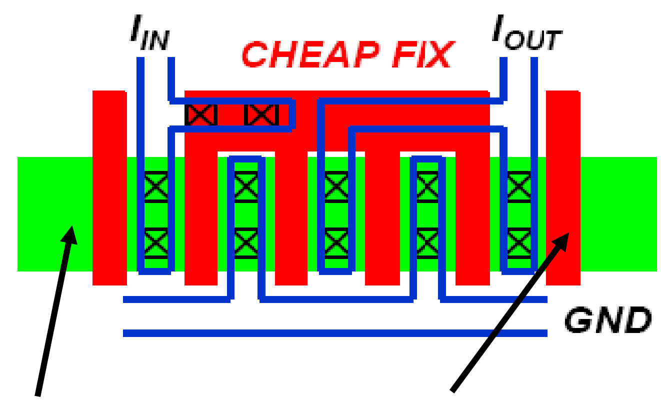
Layout guidelines for optimal matching

- Same L&W
- Same active area size, shape, & orientation
- Same environment (*e.g.*, well mask)



Example: Current Mirror



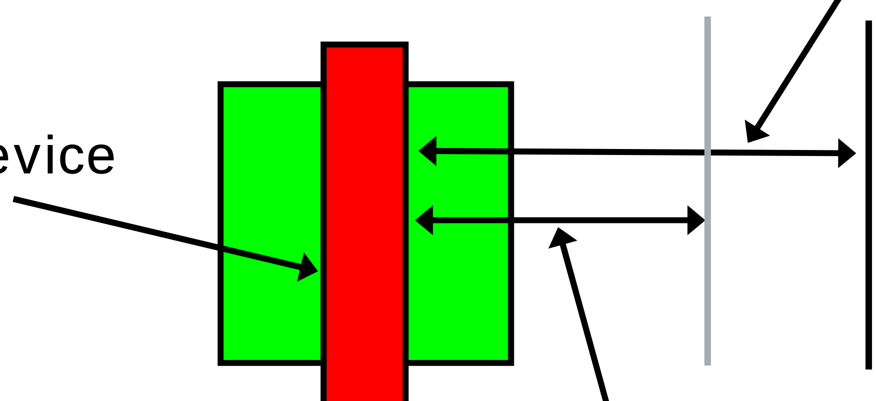


Extended OD and added dummy poly gates

Source: ST Microelectronics (2004)

Enforcement of layout guidelines

- Tag devices that are deemed “layout-critical” in schematic
- During layout implementation, these devices are subject to additional DRC rules that minimize variation due to layout
- Advantage: correct by construction
- Drawback: sacrifice layout density



Layout-dependent models and standard cell characterization

- Layout-dependent MOSFET models depend the presence of other objects in their neighborhoods
- For re-usable layout IP, like standard cell libraries, the environment will not be known until placement
- Standard cell methodology implicitly assumes that cells can be characterized *before* placement.

How do we get ourselves out of this paradox?

Boundary checks

- Enforce boundary DRC rules that minimize interaction with neighbors
- OK for large blocks, not practical for small cells like standard cell library

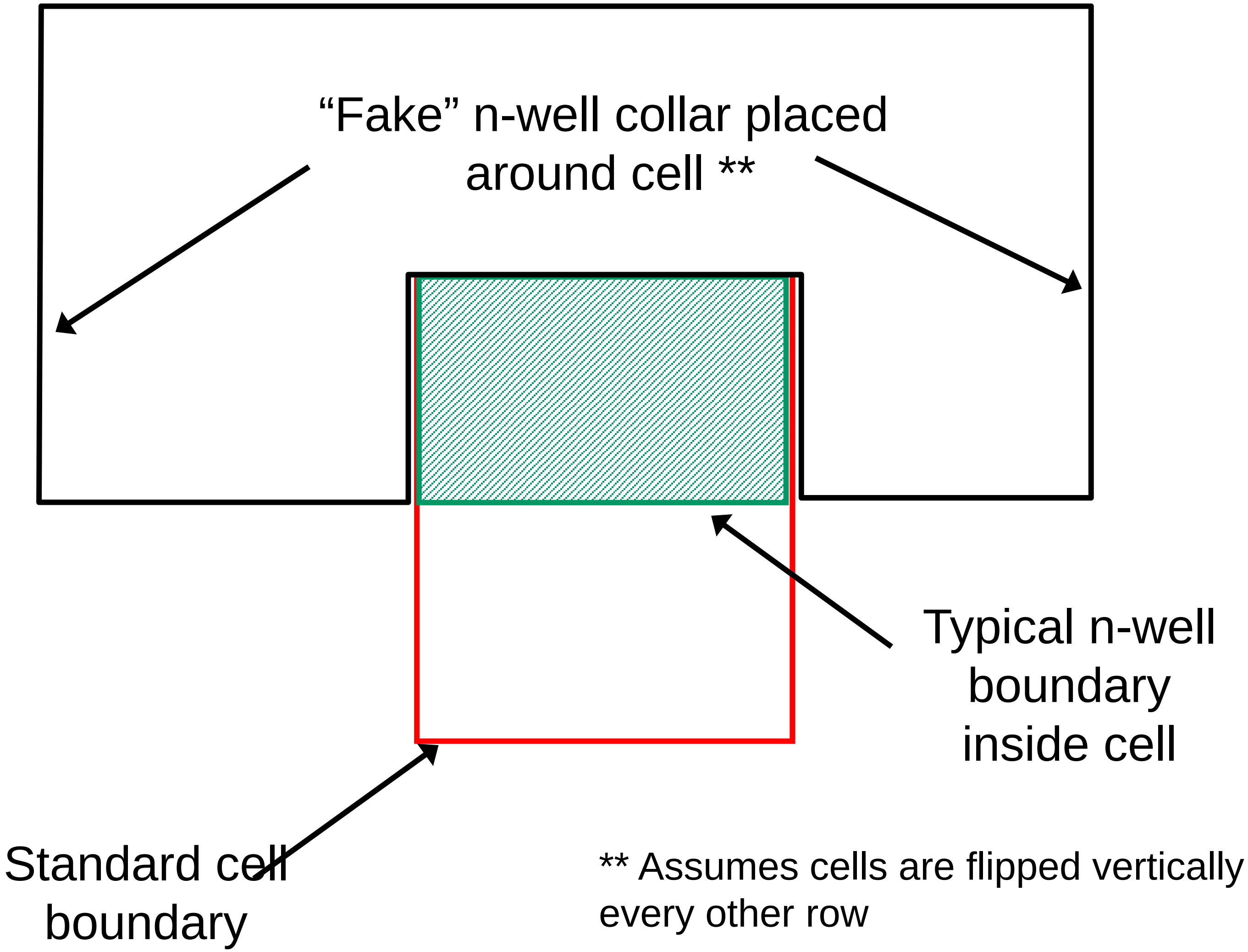
“Fake” environments

• One strategy to break the paradox is to enclose the standard cell in a “fake” environment

• Typically, standard cells are placed in rows

• You may not know exactly what is on left/right/top/bottom, but you can make an educated guess

Example of "fake" environment



** Assumes cells are flipped vertically every other row

Exhaustive simulation

- CAD vendors provide tools that extract the cell with all possible neighbor cells to quantify variation (example: Cadence LEA tool)
- Cell variation information is useful feedback for stdcell design team, but is it useful for design flow?

A posteriori checks

- Run stress_short_flow after placement to look for outlier devices
- The flow is efficient – cost is on the order of an LVS run
- But this is very late in the flow to find these issues

Summary

- Described sources of device variation due to layout
- Modeling methodology
  - Keep things simple
  - A model that can be evaluated outside of a circuit simulator is *really* handy
- CAD tool implementation
  - Provide quick feedback tools for the layout team
  - Interface to detailed analysis tools (*e.g.*, circuit simulation)
- Layout guidelines for critical devices

Trademark Attribution