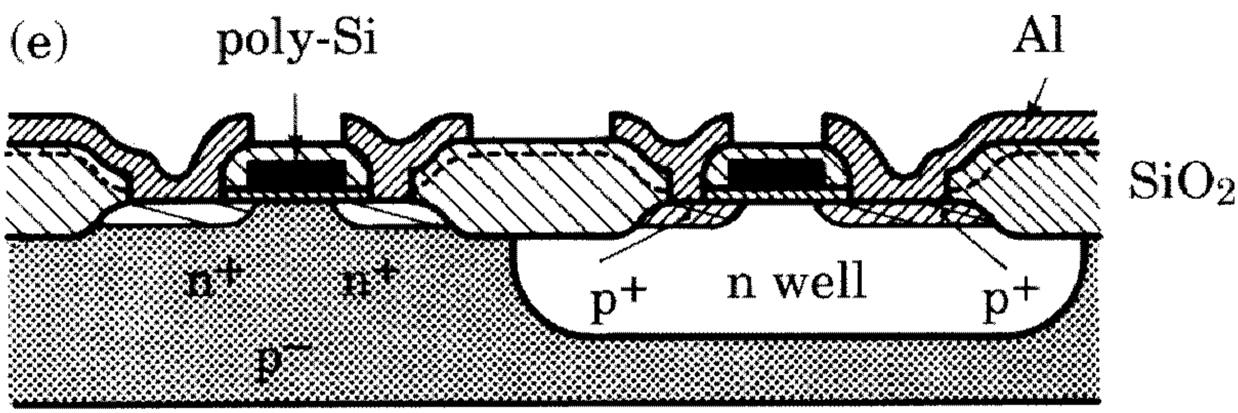
Handout #12

Design Rules/Layout

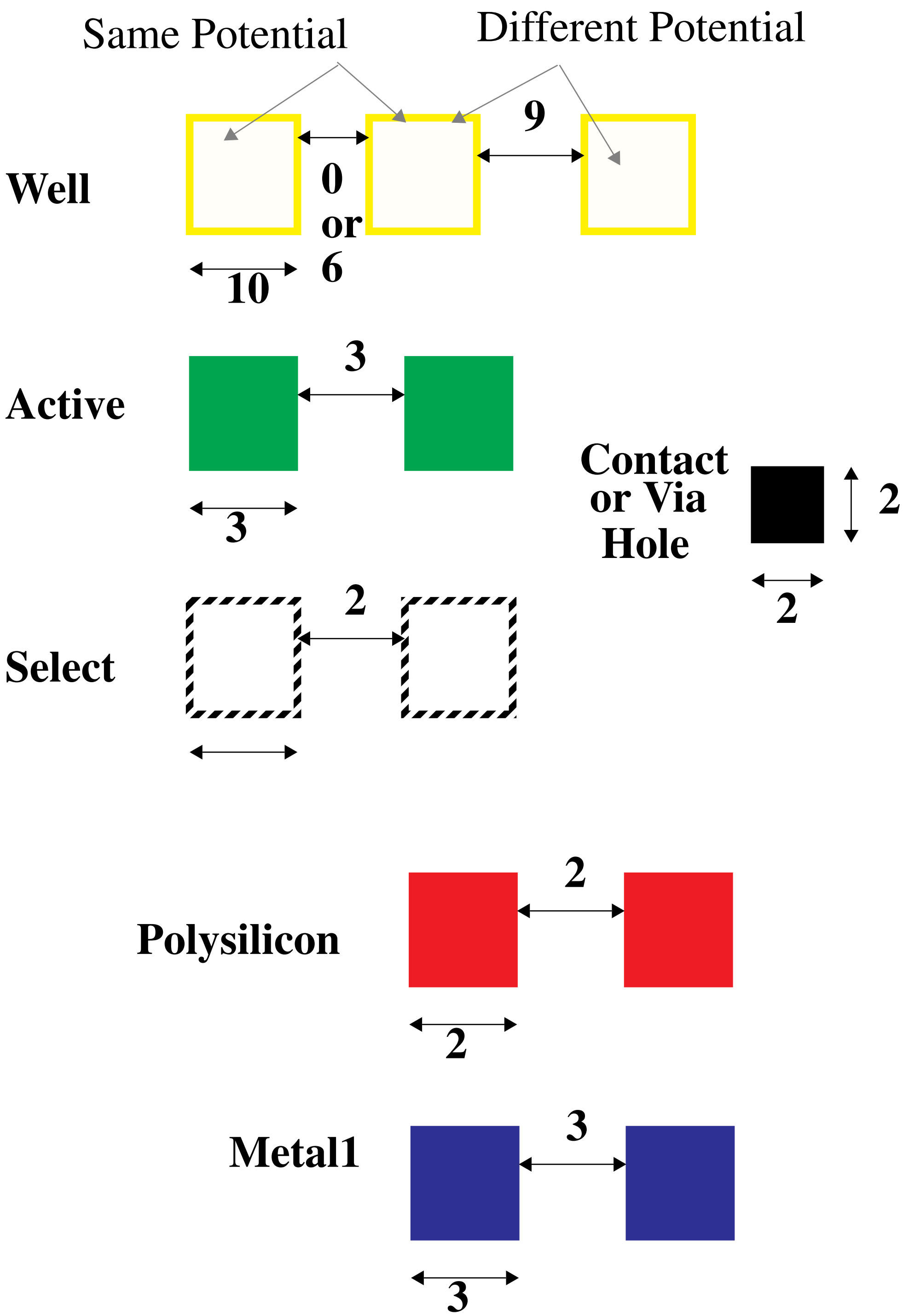
CMOS (NWELL Process)



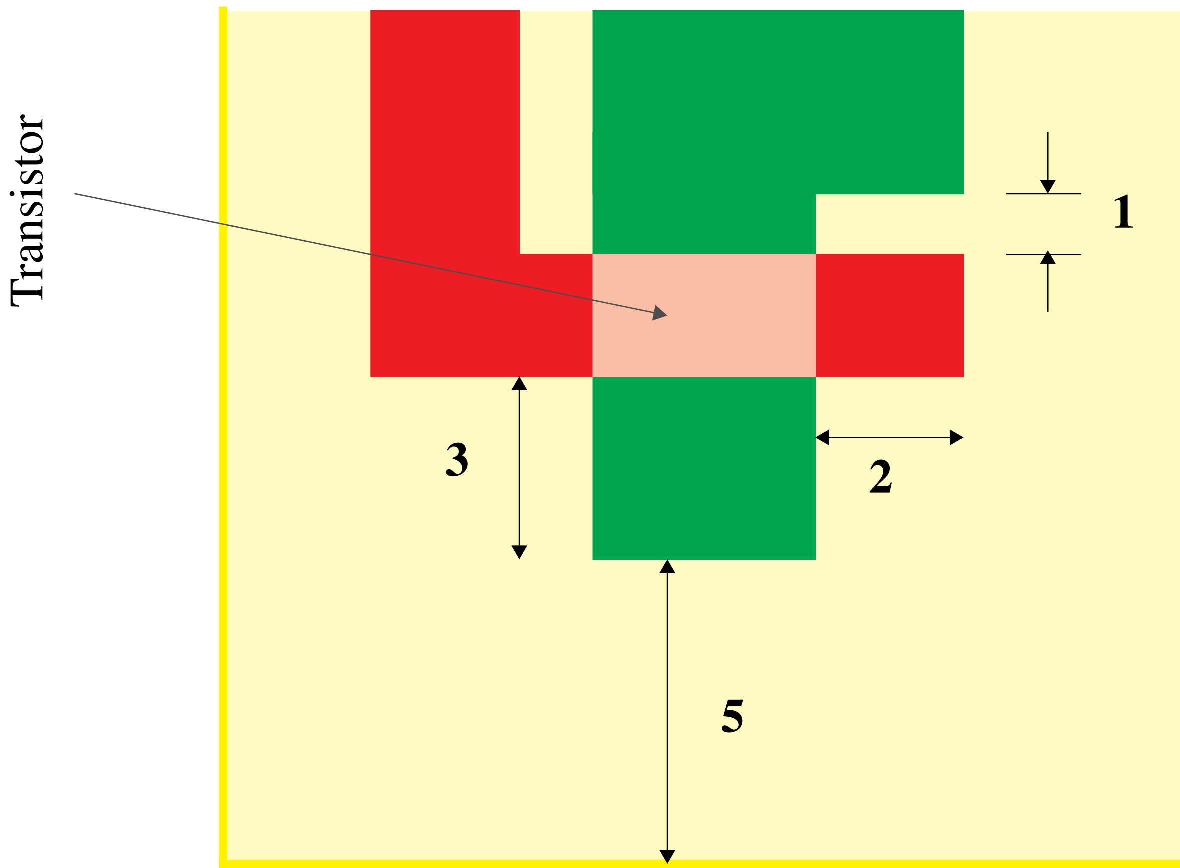
Color Coding of Layers

Layer	Color	Representation
Well (p,n)	Yellow	
Active Area (n+,p+)	Green	
Select (p+,n+)	Green	
Polysilicon	Red	
Metal1	Blue	
Metal2	Magenta	
Contact To Poly	Black	
Contact To Diffusion	Black	
Via	Black	

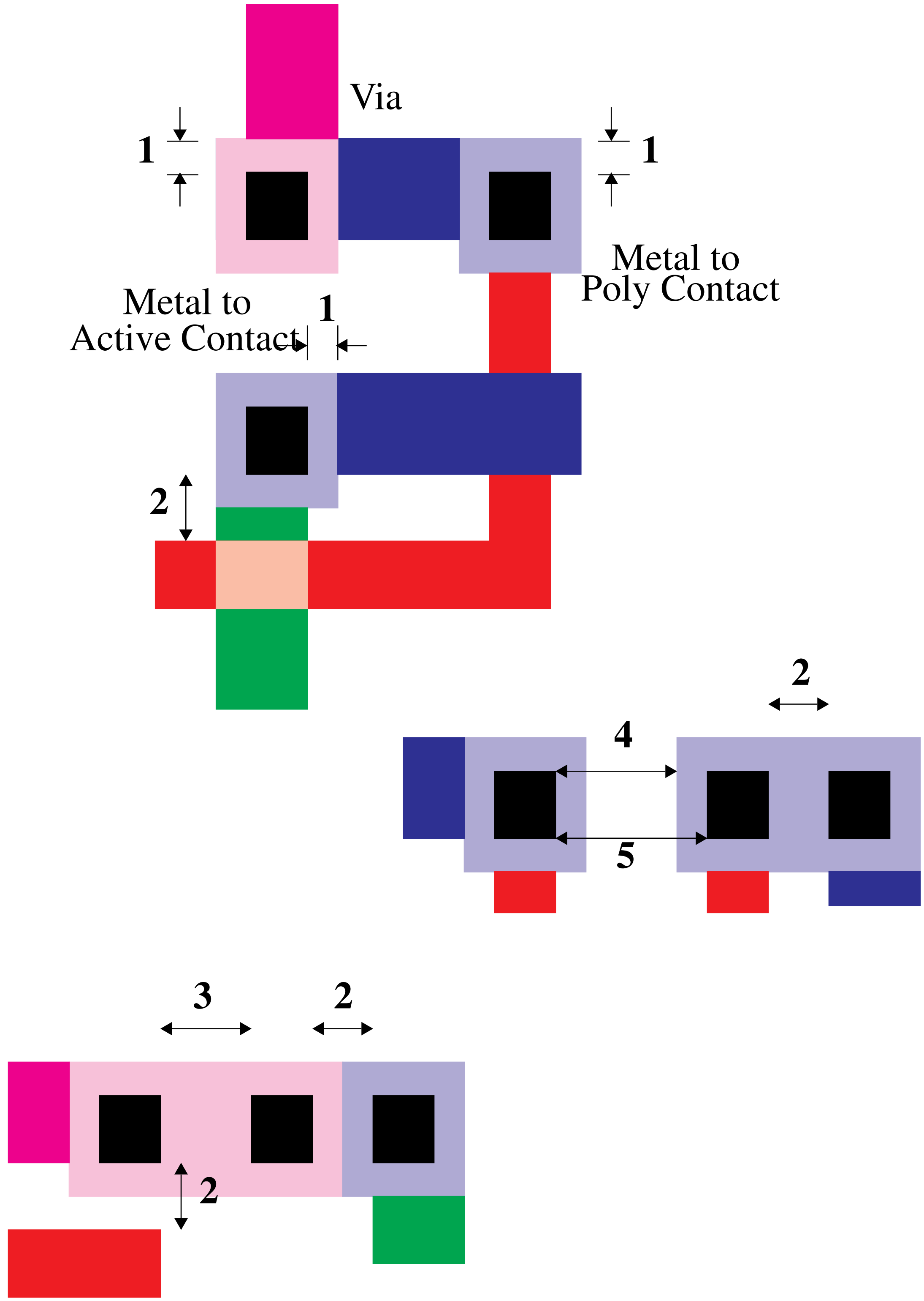
Intra-Layer Design Rules



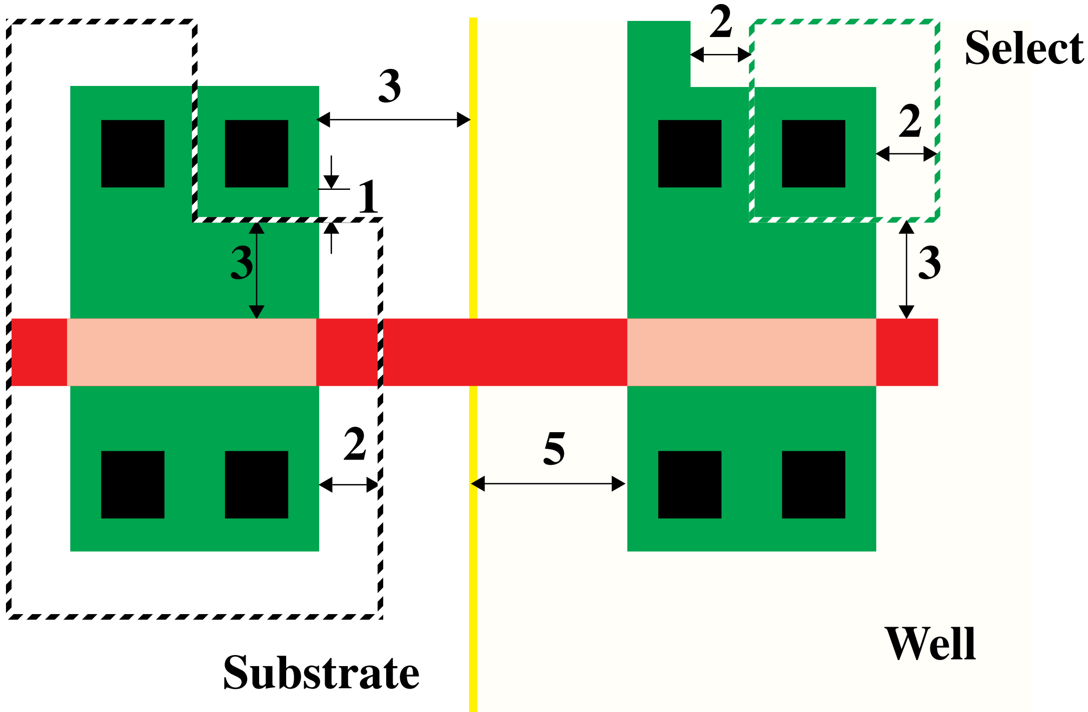
Layout Rules Regarding Transistor Layout



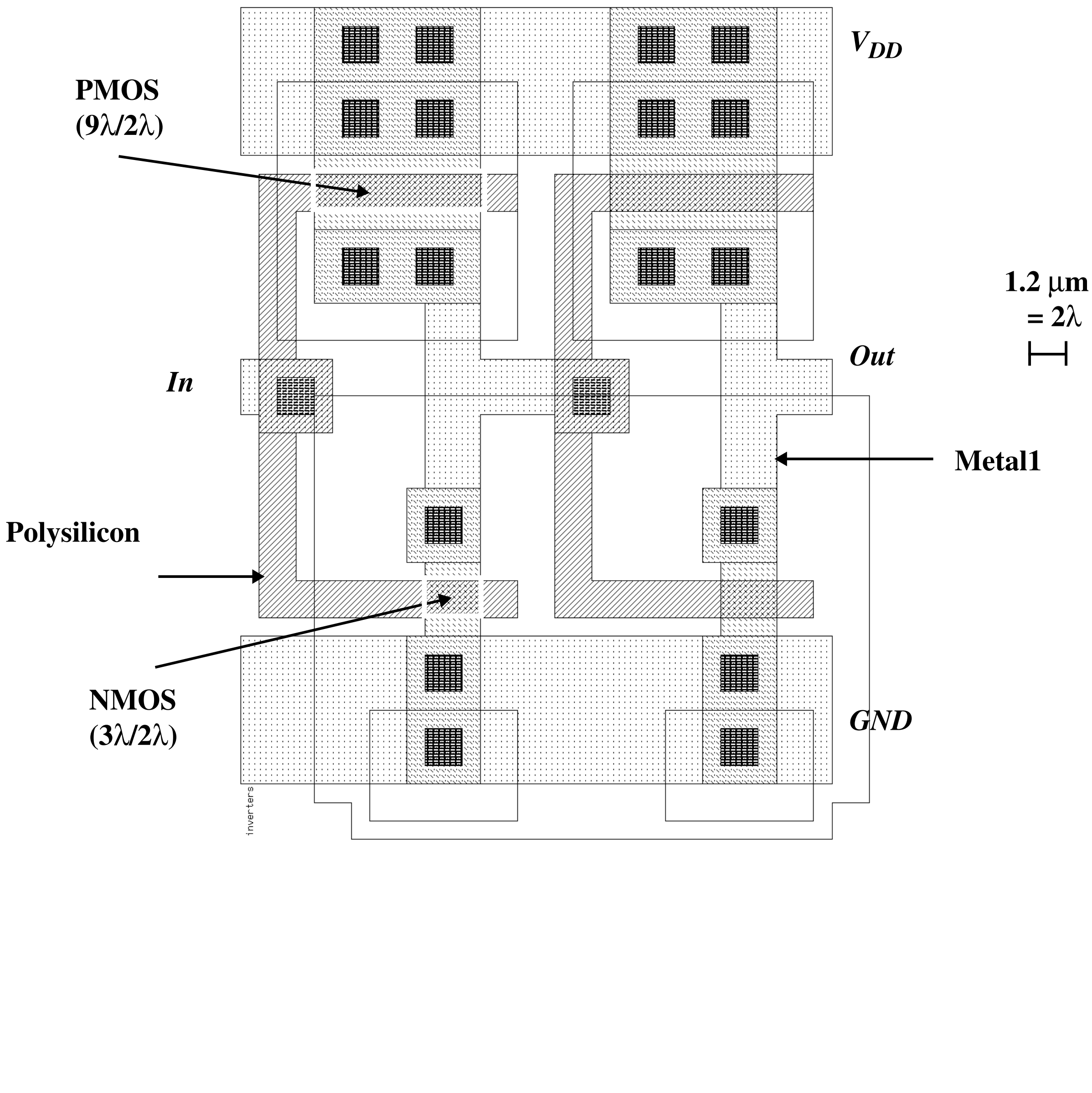
Vias and Contact



Select Layer



Inverter Layout



Areas and Perimeters for NMOS and PMOS

NMOS:

Area (W * L) = $3\lambda * 2\lambda = 6\lambda^2$

Drain Area: $4\lambda * 4\lambda + 3\lambda * 1\lambda = 19\lambda^2$

minimum contact size

Drain Perimeter: $4\lambda * 3 + 1\lambda + 1\lambda + 1\lambda = 15\lambda$

Source Area: $4\lambda * 4\lambda + 3\lambda * 1\lambda = 19\lambda^2$

Source Perimeter: $4\lambda * 3 + 1\lambda + 1\lambda + 1\lambda = 15\lambda$

PMOS:

Area (W * L) = $9\lambda * 2\lambda = 18\lambda^2$

Drain Area: $9\lambda * 4\lambda + 9\lambda * 1\lambda = 45\lambda^2$

Drain Perimeter: $9\lambda + 4\lambda + 1\lambda + + 4\lambda + 1\lambda = 19\lambda$

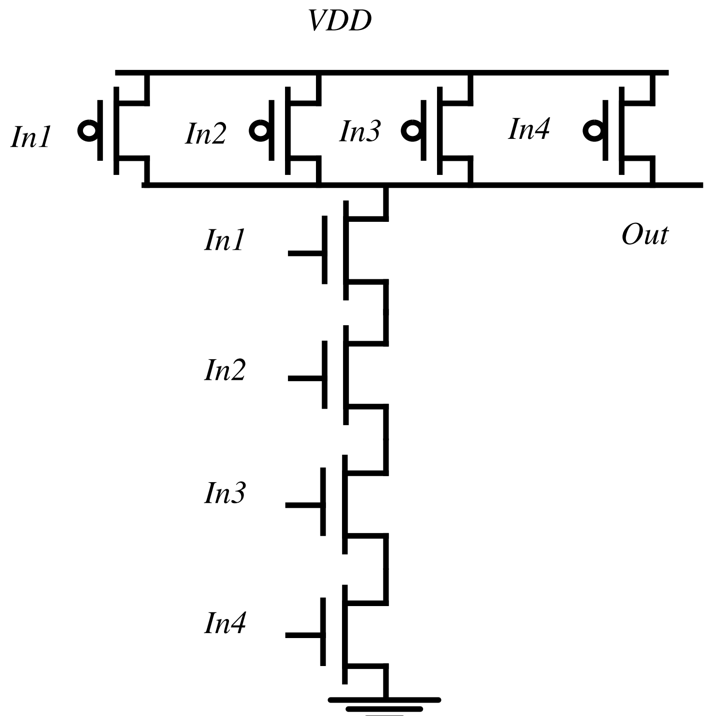
Source Area: $9\lambda * 4\lambda + 9\lambda * 1\lambda = 45\lambda^2$

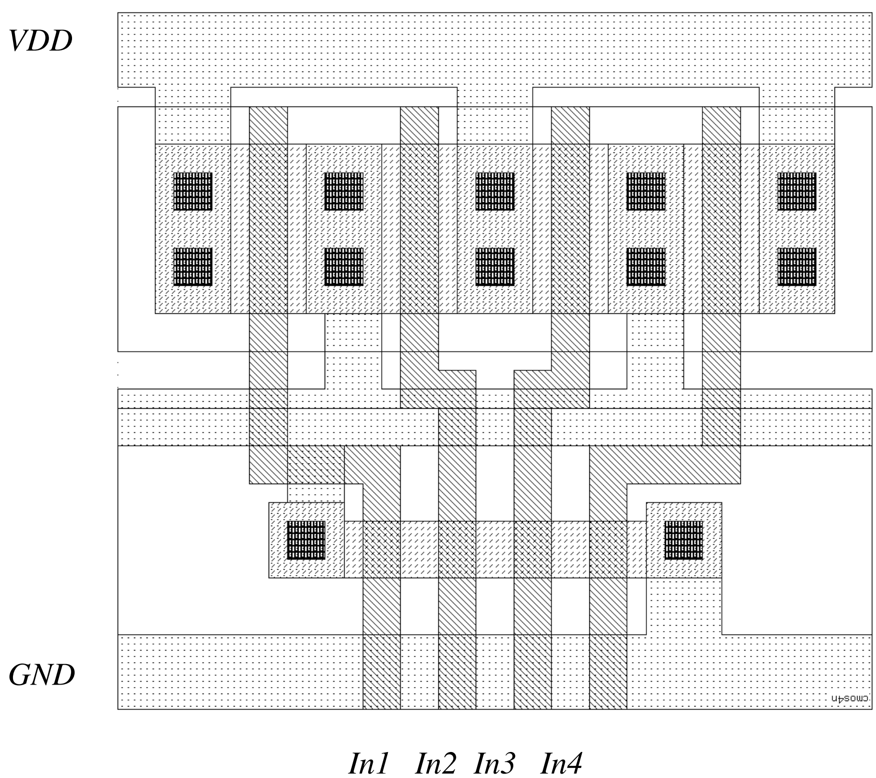
Source Perimeter: $9\lambda + 4\lambda + 1\lambda + 4\lambda + 1\lambda = 19\lambda$

Example (Cont.)

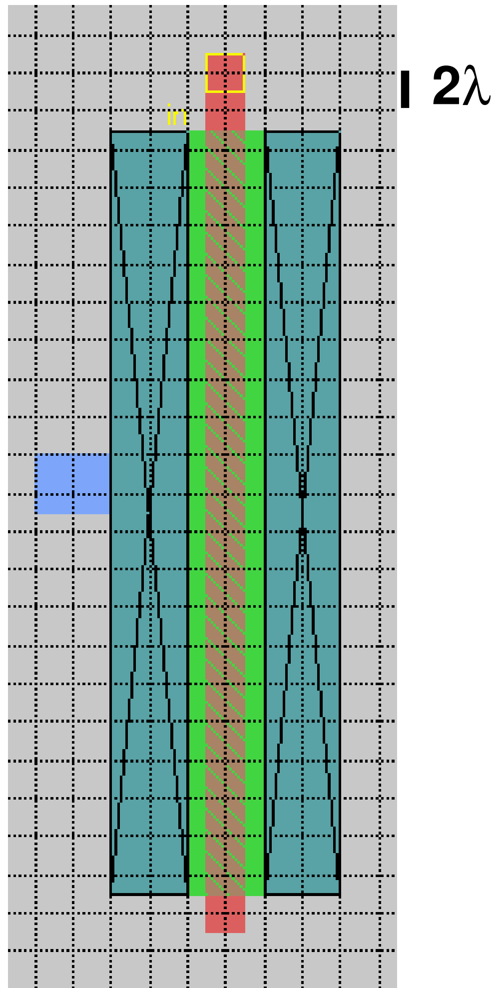
	W/L	**AD (μm^2)**	**PD (μm)**	**AS (μm^2)**	**PS (μm)**
NMOS	1.8/1.2	6.84 (19 λ^2)	9.0 (15λ)	6.8 (19 λ^2)	9.0 (15λ)
PMOS	5.4/1.2	3*5.4 = 16.2(45 λ^2)	3*2 + 5.4 = 11.4 (19λ)	16.2 (45 λ^2)	11.4 (19λ)

4 Input NAND





Layout of Large Transistors

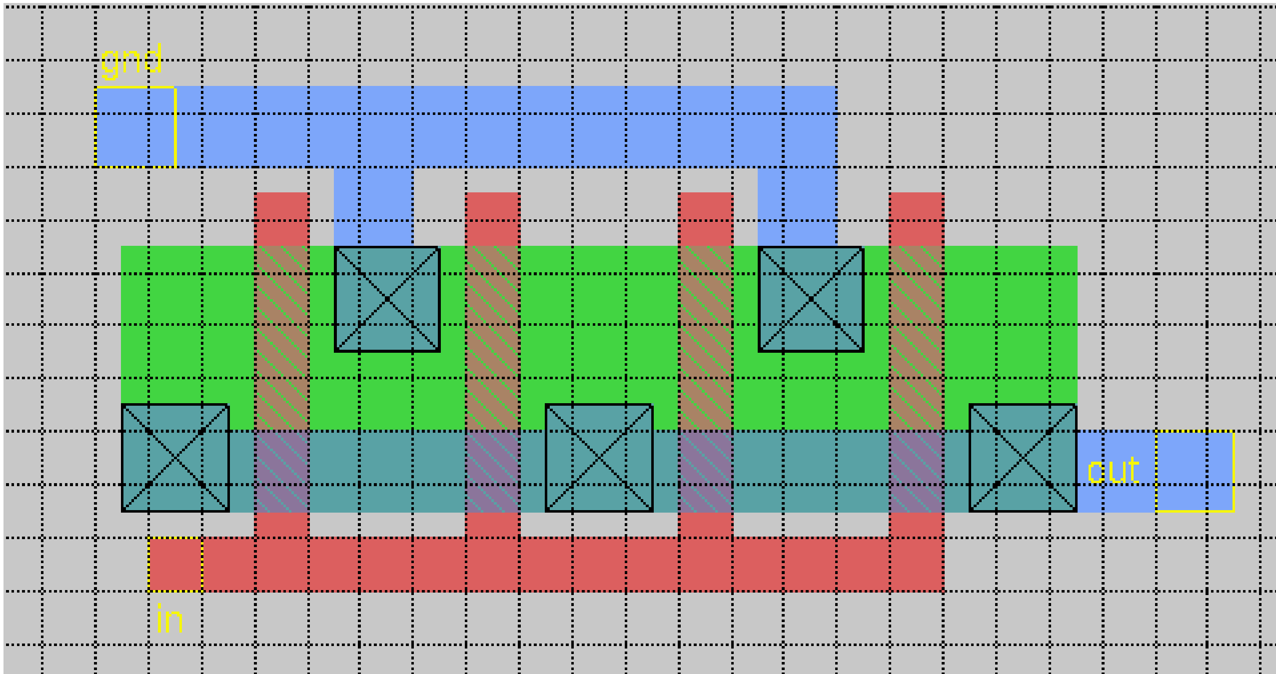


W/L = 40/2

AD =

PD =

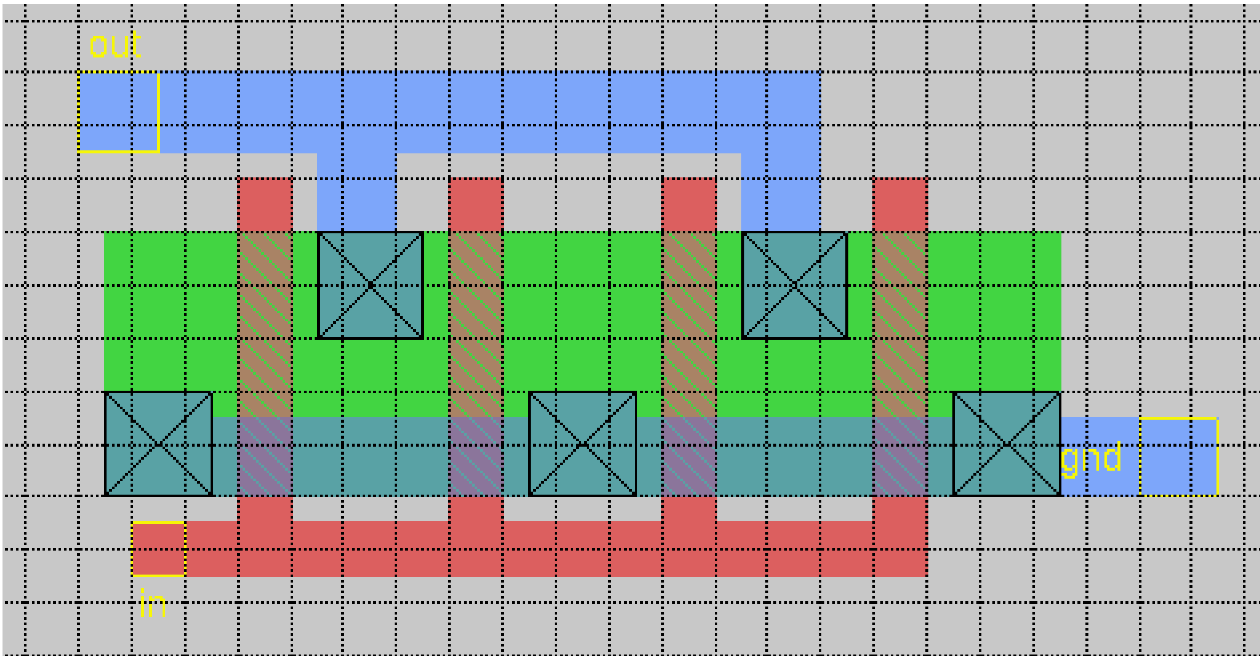
Parallel Transistors ("Finger" Layout)



AD =

PD =

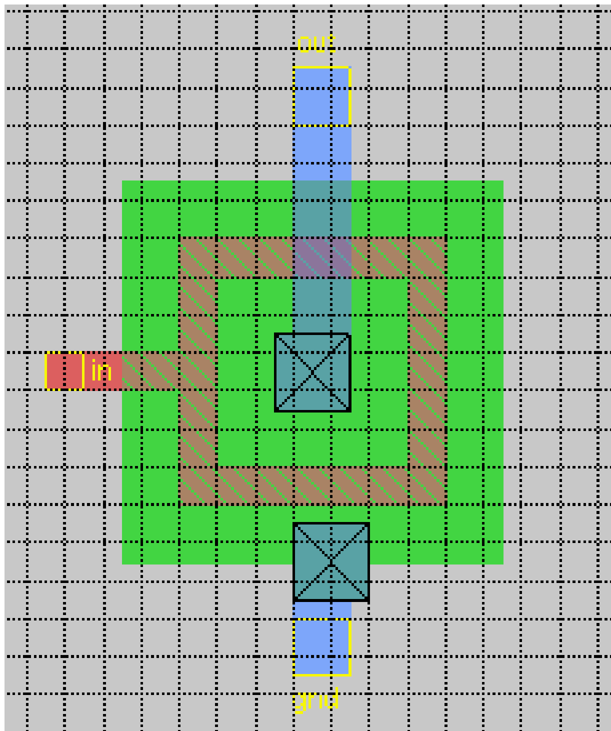
Parallel Transistors ("Finger" layout)



AD =

PD =

Doughnut Transistors



AD =

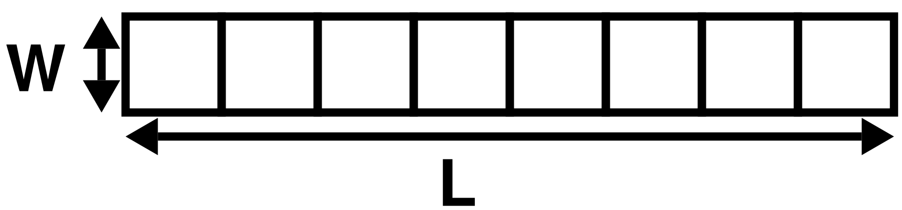
PD =

Resistance of Interconnect

Material	Sheet Resistance ($\Omega/\square$)
n+ diffusion	10 ± 2
p+ diffusion	10 ± 2
n-well (under field)	1150 ± 250
polysilicon (H = 0.33μm)	10 ± 2
metal1, metal2	0.07 ± 0.006

Sheet Resistance Values for a Typical 1.0 μm CMOS process.

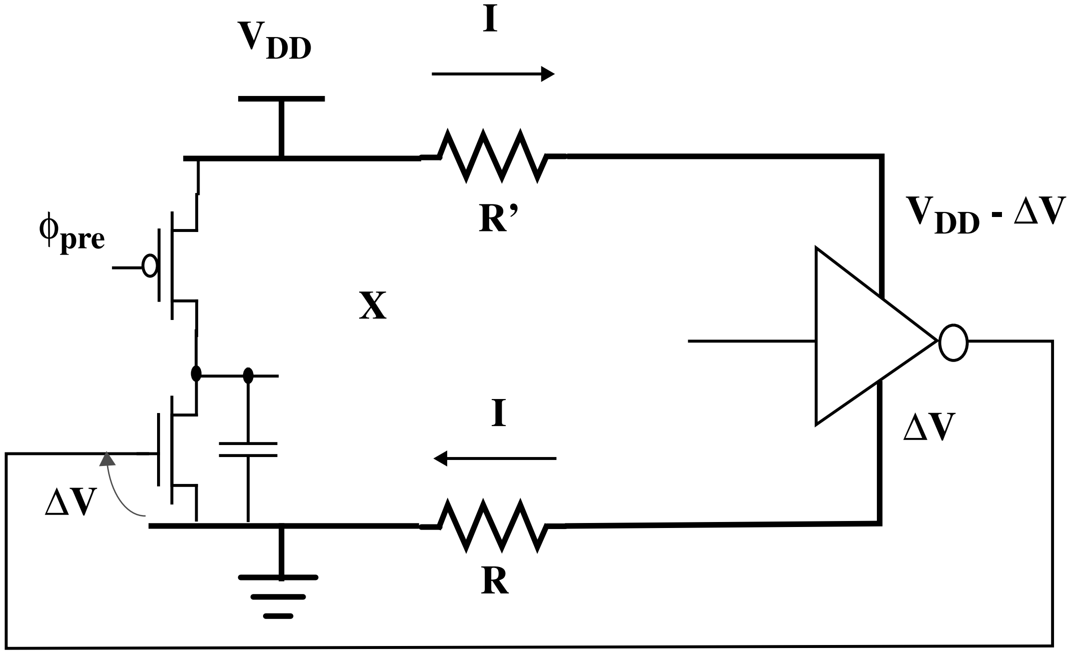
$$R = R_{\square} \frac{L}{W}$$



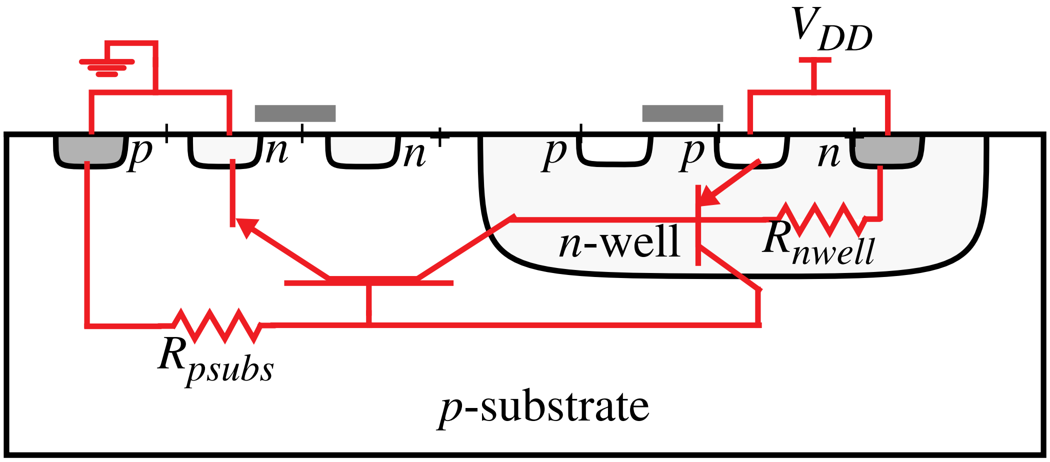
- OK to route using poly for local interconnect (< 50 squares) but avoid using it for global interconnect (why?)

- Minimize routing using diffusion

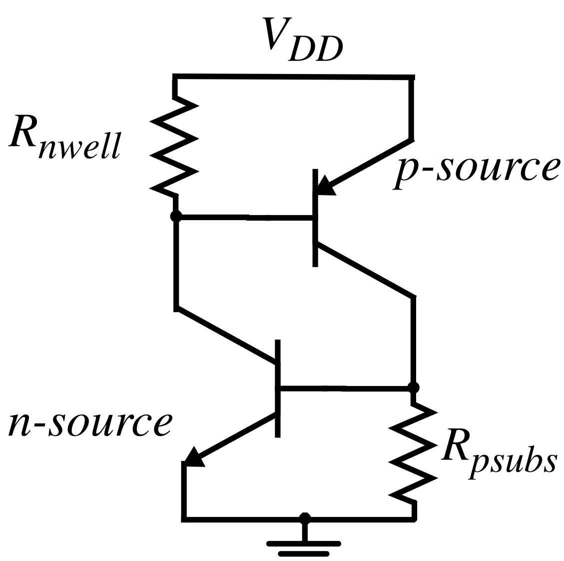
RI Noise



Latch Up



(a) Origin of latchup



(b) Equivalent circuit

To avoid latchup - R_{nwell} and R_{psubs} should be minimized (place a lot of substrate and well contacts)

Standard Cell Approach

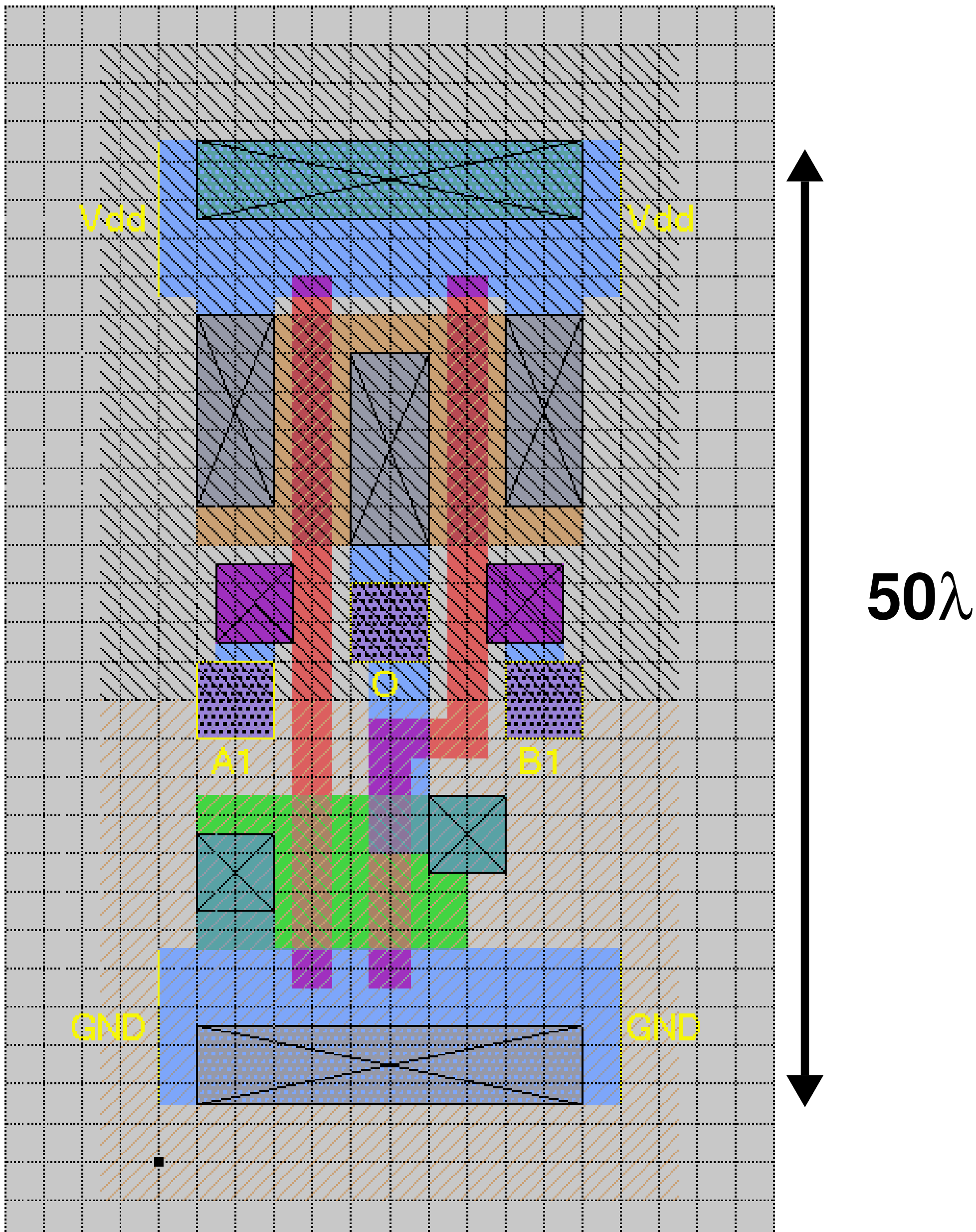
- A library based approach which includes a wide selection of logic gates with a wide range of fan-in and fan-out (AND, NOR, XOR, Adder, Latches, etc.)

- Simple layout rules followed for the cell design (e.g., fixed height, where input/output terminal are located, etc.)

- Each cell is characterized for different process corners (V_{DD}, temp., loading, etc.) - just like components in a printed circuit board

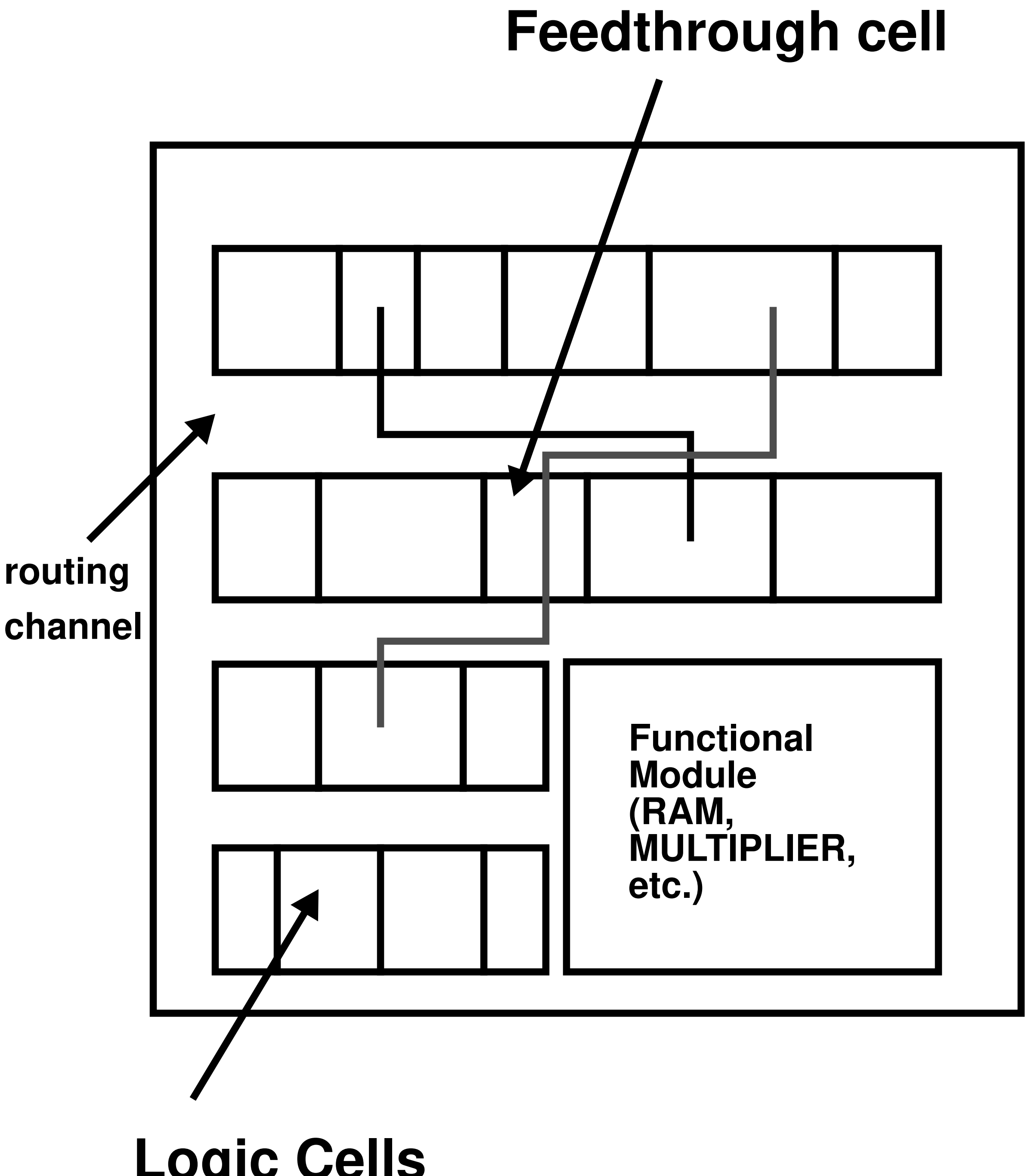
- Automatic placement and routing tools exist

Example Standard Cell



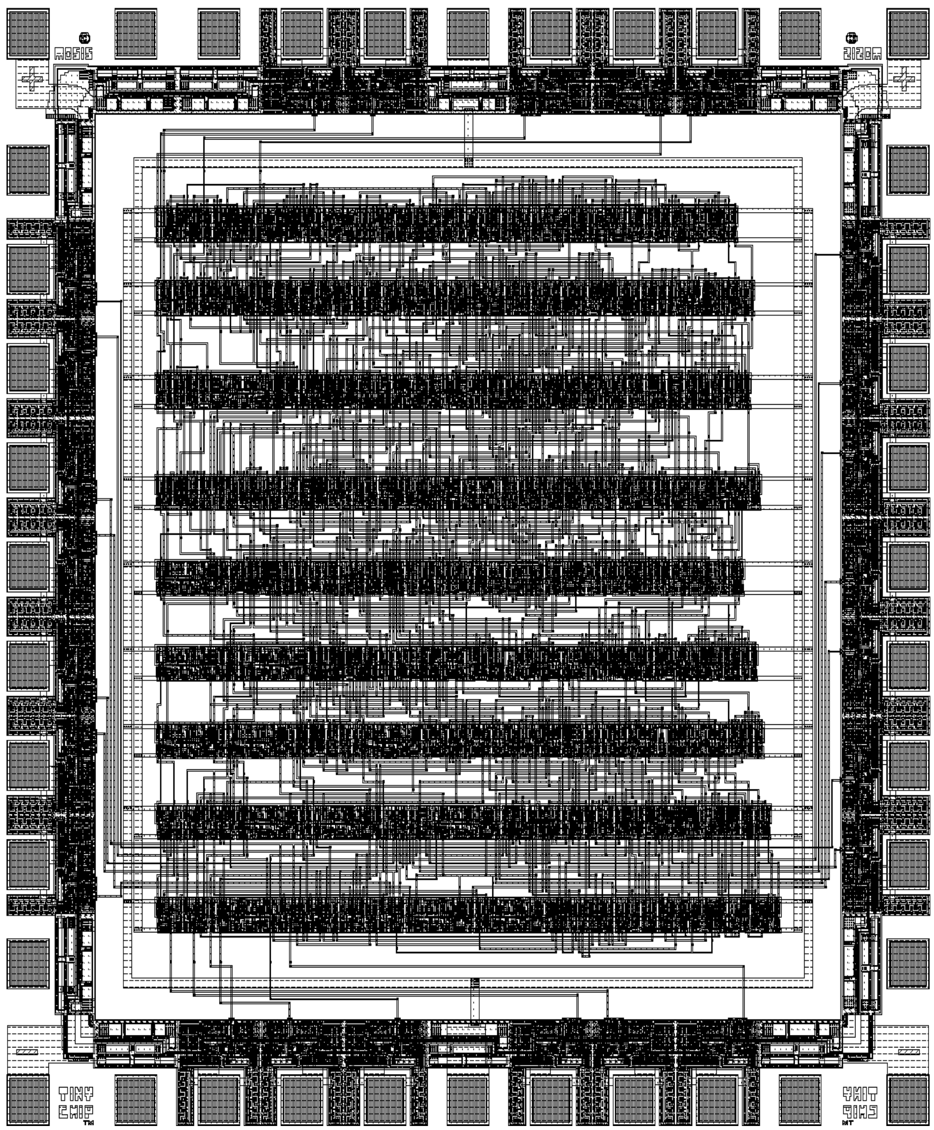
- Height of *all* cells are fixed and width of cells allowed to vary to accomodate variation in complexity
- Terminals connections made in Metal 2

Standard Cell Routing Strategy



- logic cells are designed to abut

Example Standard Cell

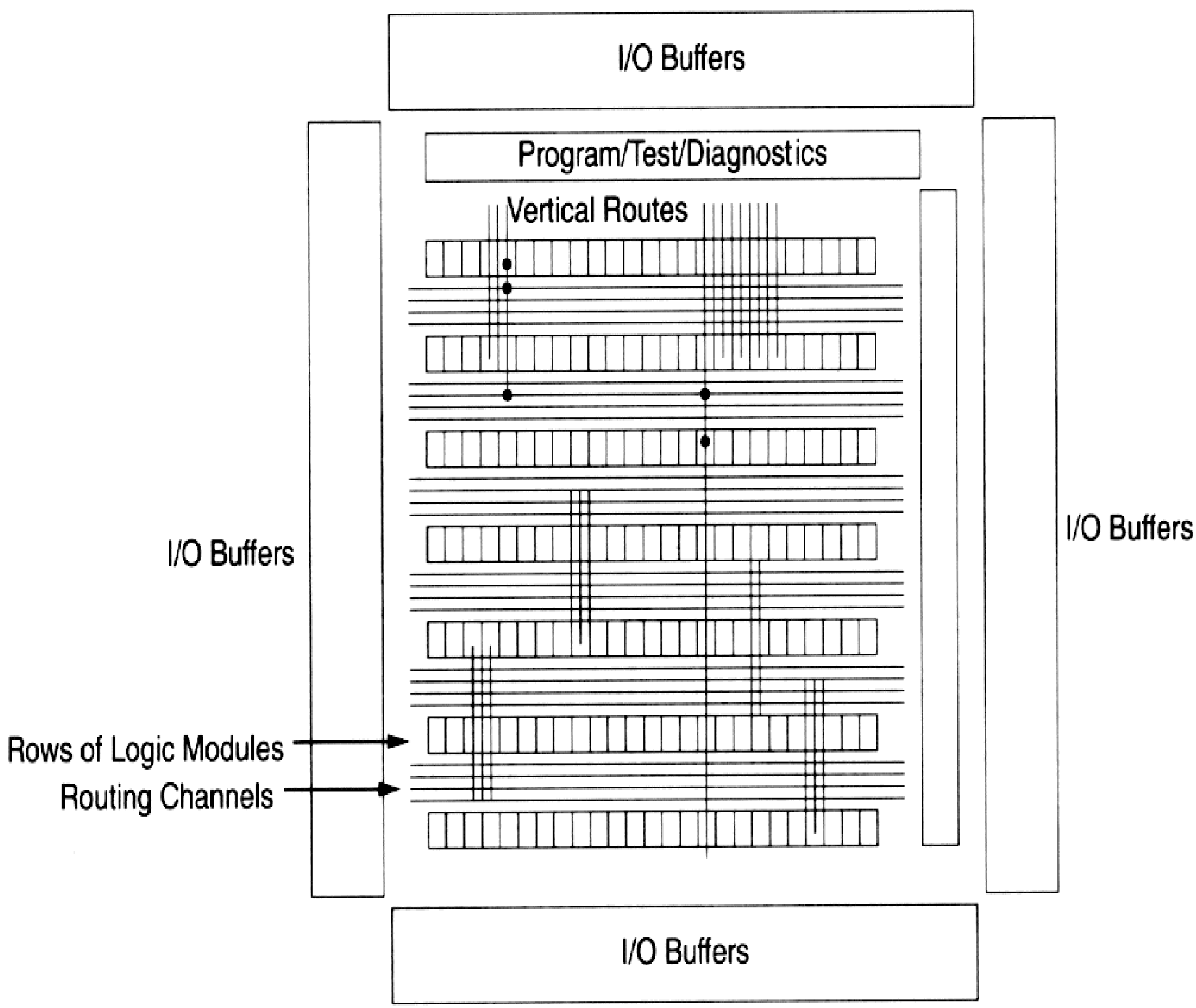


Design with Standard Cells

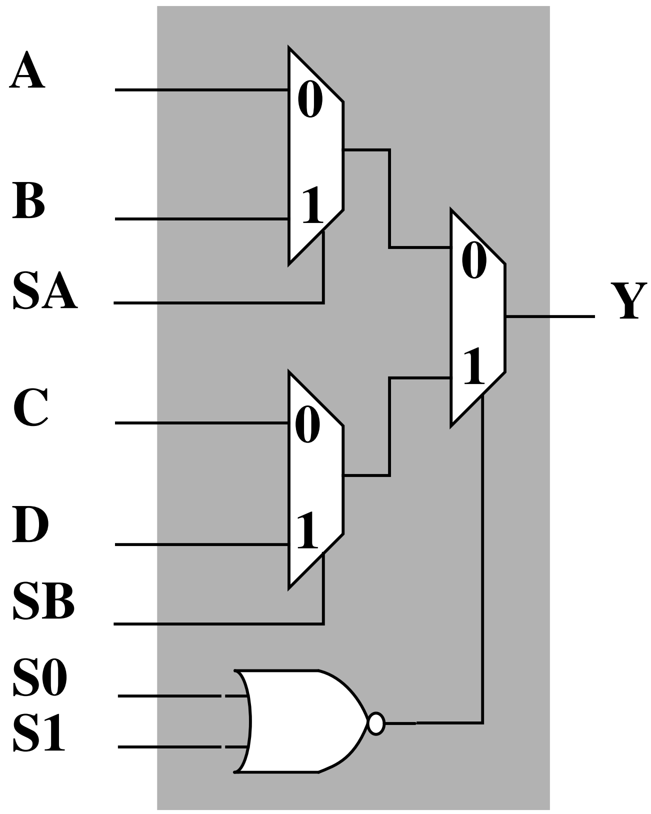
- Wiring capacitance (can be large) - placement often uses simulated annealing (various cost metrics used including wire length, speed, area...)
- Short circuit power can be an issue (but can be addressed by inserting proper buffering) - Why?
- Hard vs. Soft Macros
- Tools exist to synthesize standard cell logic cells
- For more examples, look in:

http://WWW.ERC.MsState.Edu/mpl/libraries/stdcells (many standard cell schematics and layouts for the mosis rules!) - Isn't web technology great?

Field Programmable Gate Array (modeled after the standard cell approach)



Logic Cell used in the Actel fuse based FPGA

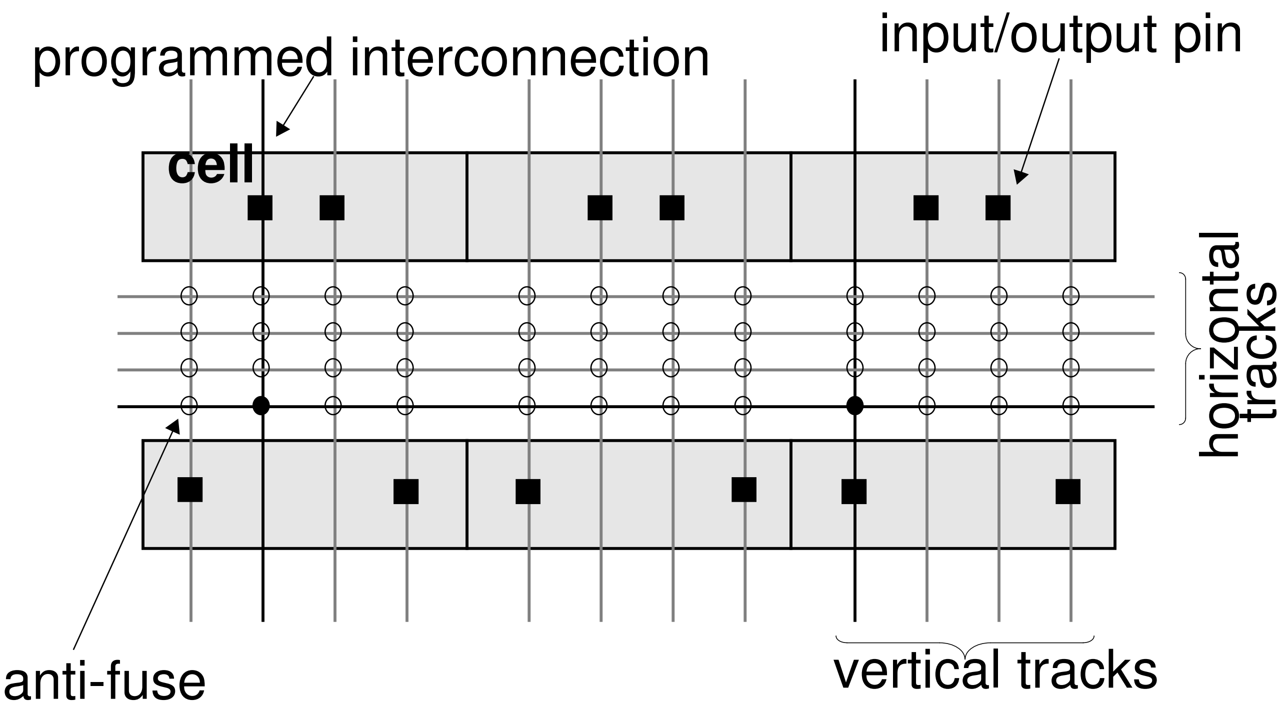


This cell can be used to program any two input or three input logic function, and some four-input functions and a latch. e.g.,

with A=1, B=0, C=0, D=1, SA=SB=In1 and S0=S1=In2 then this module performs the XOR function

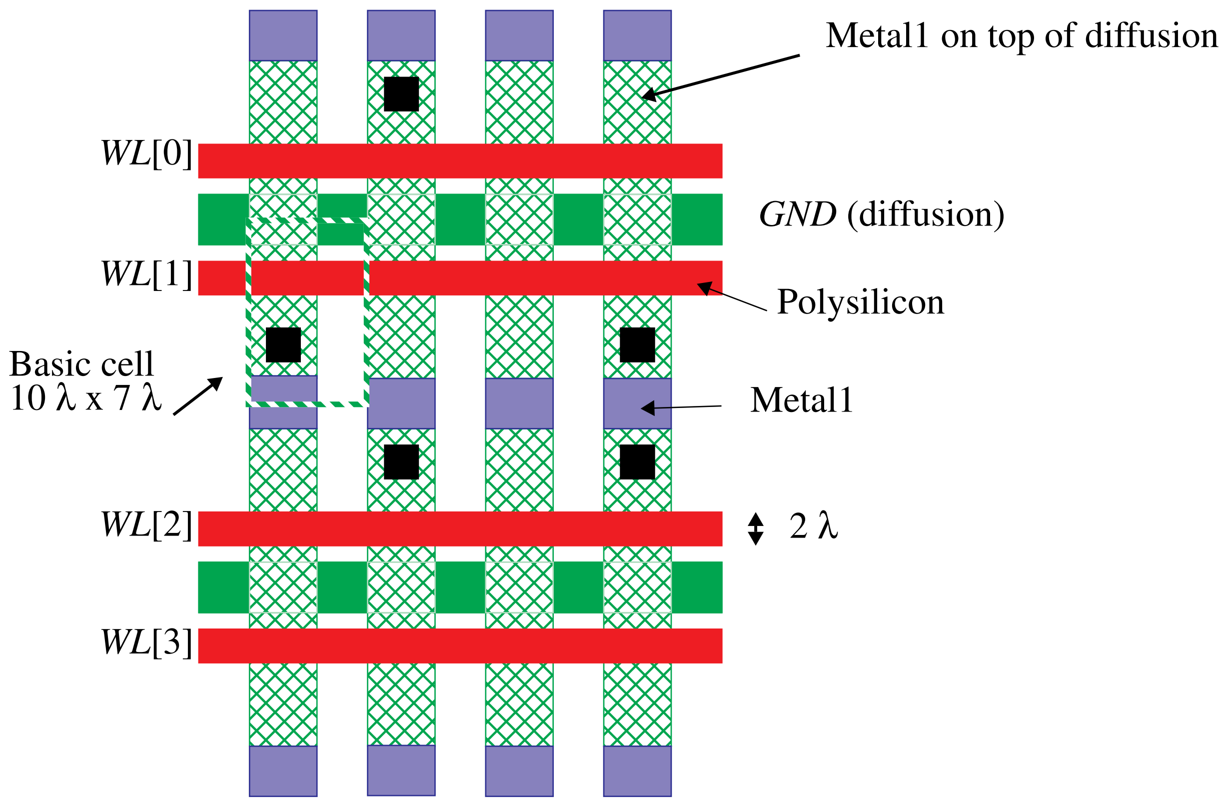
Exercise: determine the programming required for the 2-input XNOR

Programming Interconnect Using Antifuses



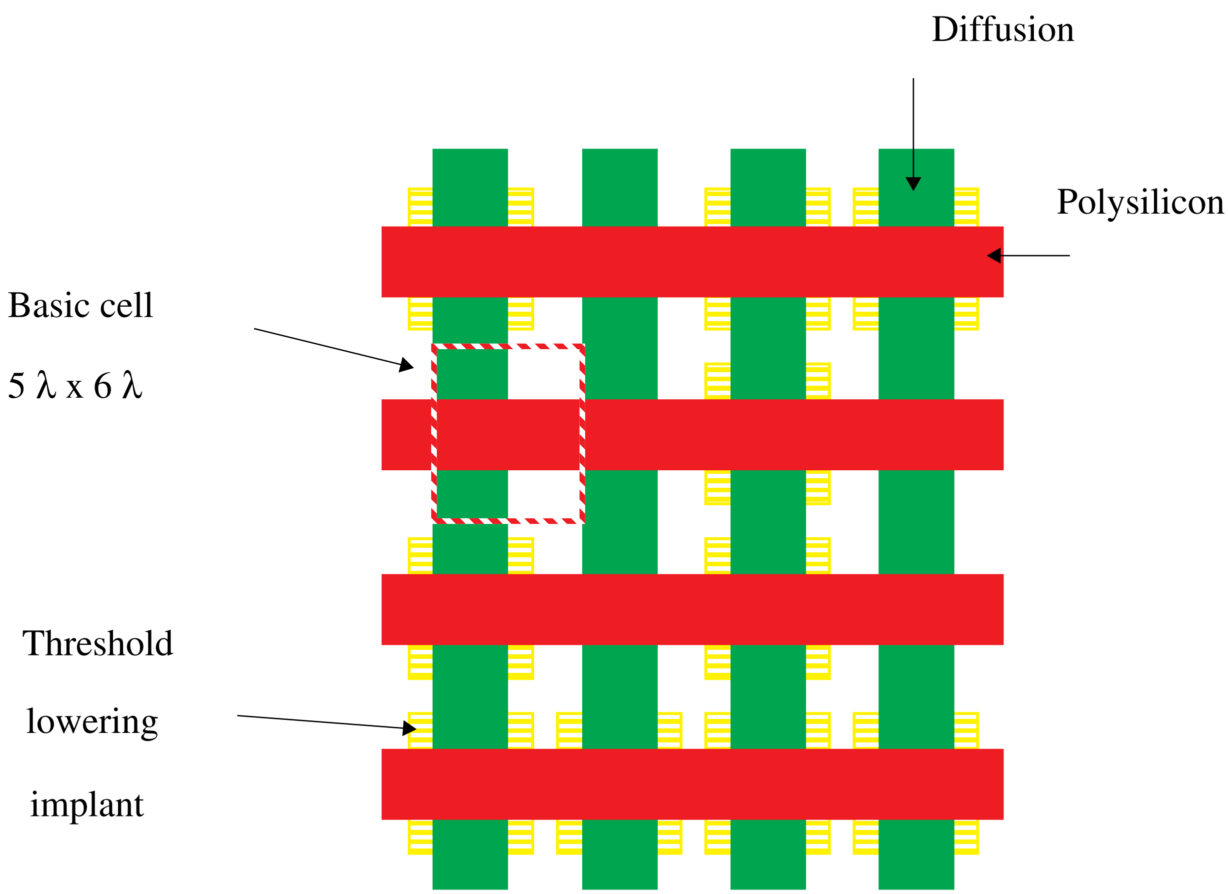
Anti-fuse: This component, which represents an open circuit (or a high resistance > 100 MΩ) in the unprogrammed mode, can be turned into a short-circuit (or small resistance < 500 Ω) during the programming phase. One mechanism to realize an anti-fuse is to sandwich a dielectric (such as Oxide-Nitride-Oxide) between two conducting layers and to cause breakdown in the dielectric during programming by applying a high voltage. This operation is, obviously, a one time event and cannot be undone.

Breaking the rules (memory) - NOR ROM



- routing wordlines in poly for density

NAND ROM Layout



No contact to VDD or GND necessary;
Reduces cell size significantly