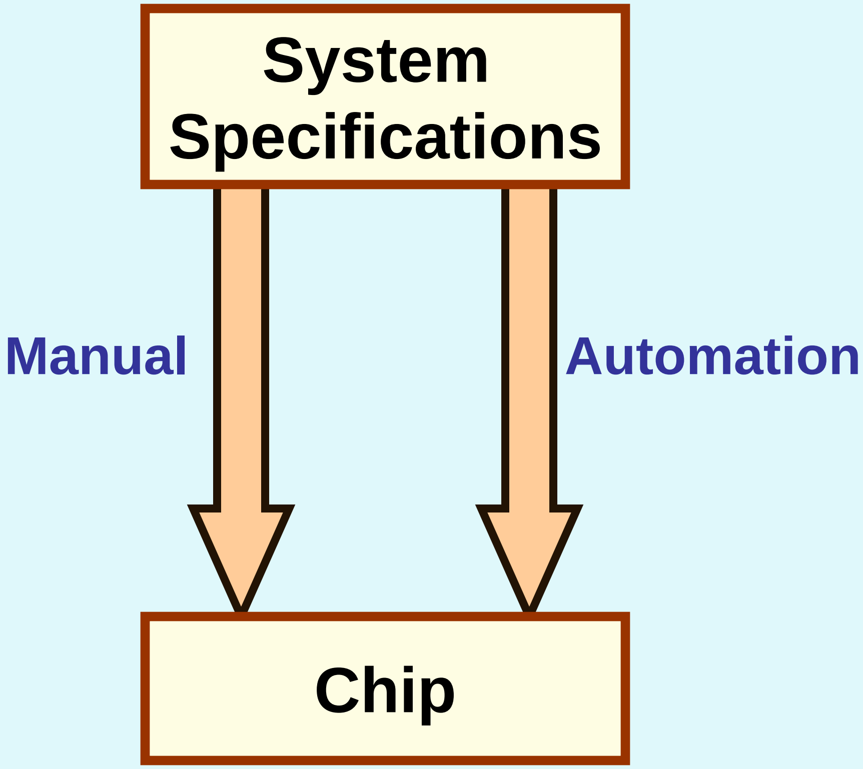
VLSI Design Styles

Basic Concepts in VLSI Physical Design Automation

VLSI Design Cycle

- Large number of devices
- Optimization requirements for high performance
- Time-to-market competition
- Cost



System Specifications

Manual

Automation

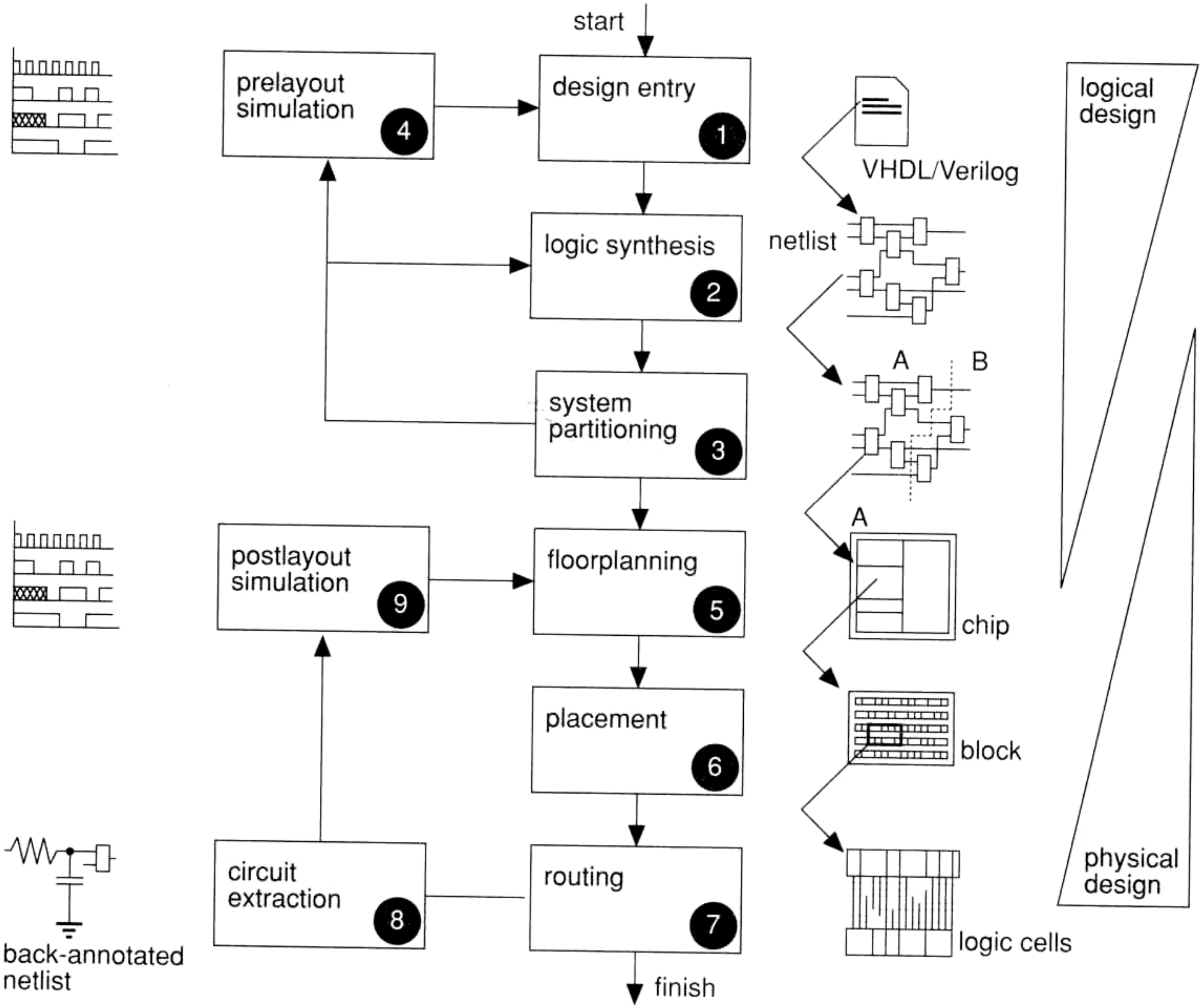
Chip

VLSI Design Cycle (contd.)

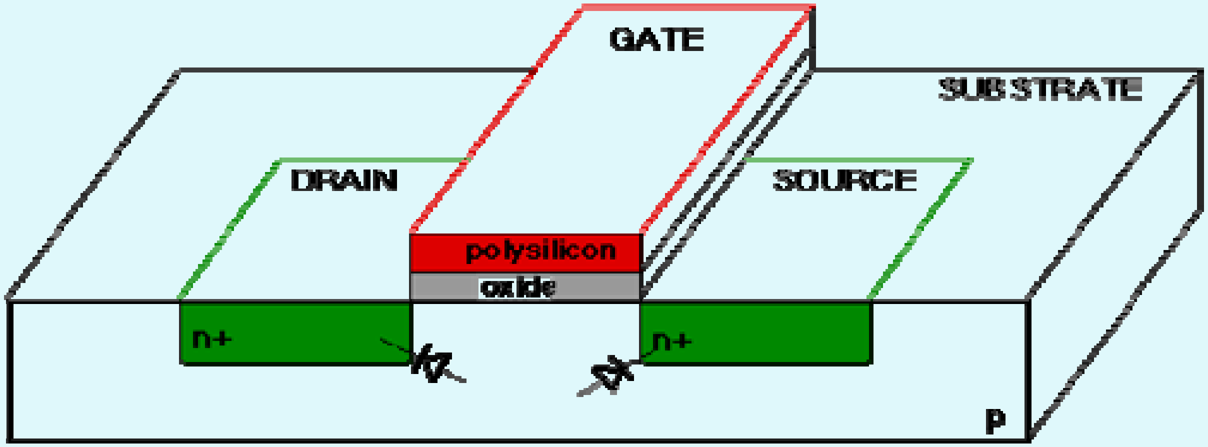
1. System specification
2. Functional design
3. Logic design
4. Circuit design
5. *Physical design*
6. Design verification
7. Fabrication
8. Packaging, testing, and debugging

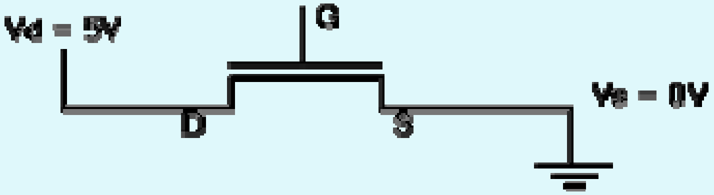
Physical Design

- Converts a circuit description into a geometric description.
  - This description is used for fabrication of the chip.
- Basic steps in the physical design cycle:
  1. Partitioning
  2. Floorplanning and placement
  3. Routing
  4. Compaction

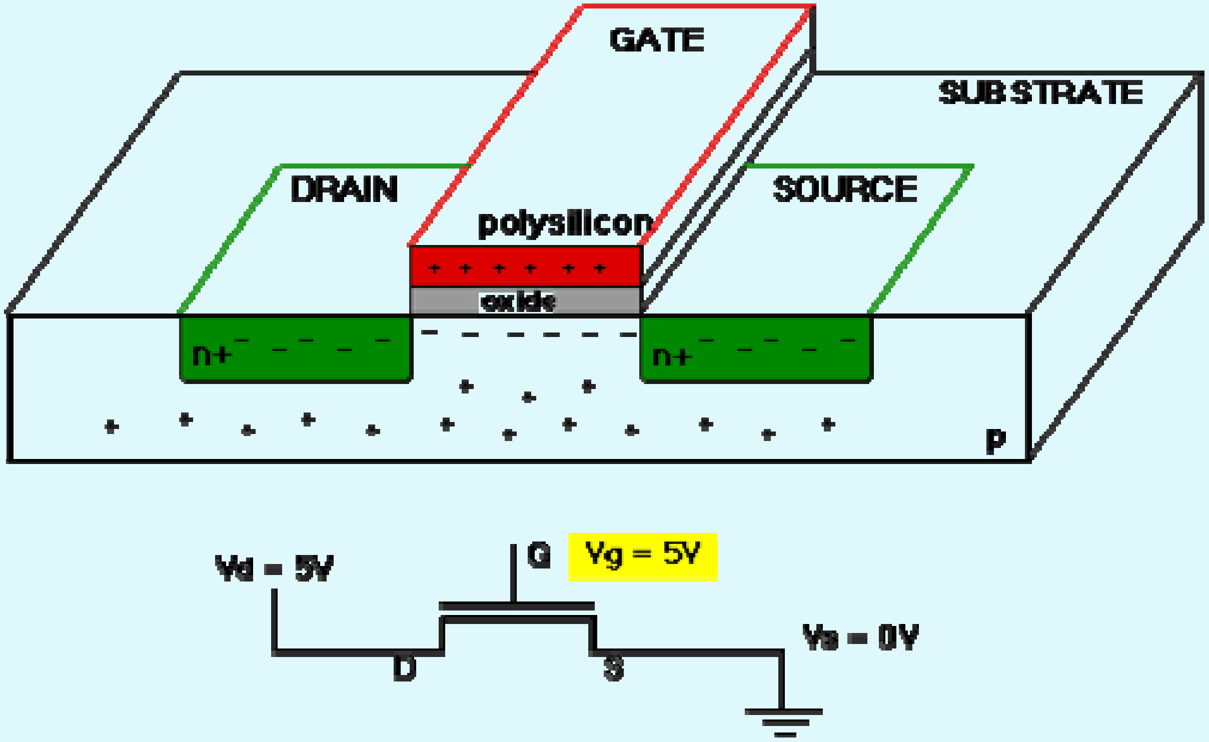

start
design entry
1
logic synthesis
2
system partitioning
3
prelayout simulation
4
floorplanning
5
placement
6
routing
7
circuit extraction
8
postlayout simulation
9
finish
back-annotated netlist
VHDL/Verilog
netlist
A
B
A
chip
block
logic cells
logical design
physical design

n-channel Transistor

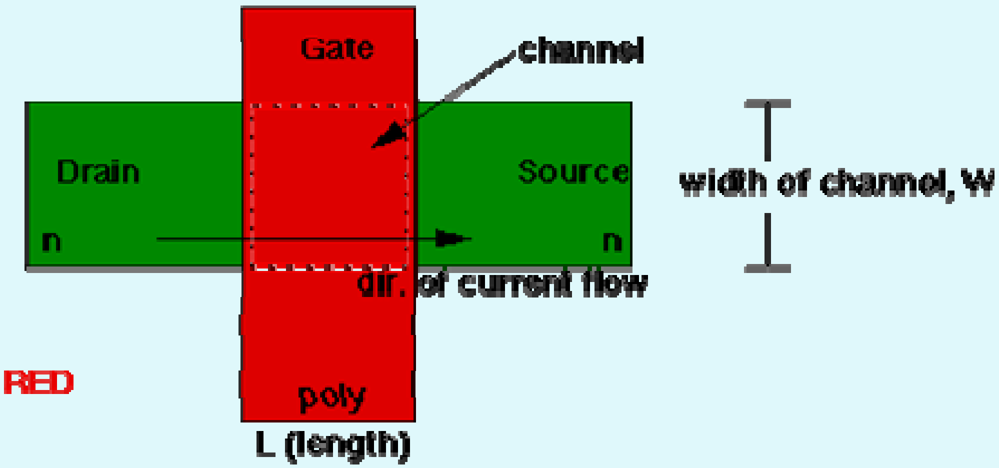




n-channel Transistor Operation



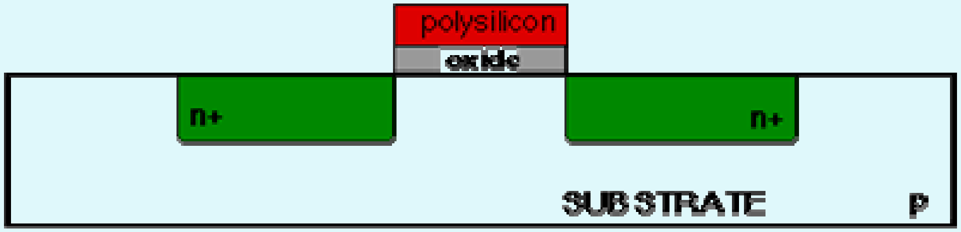
n-channel Transistor Layout



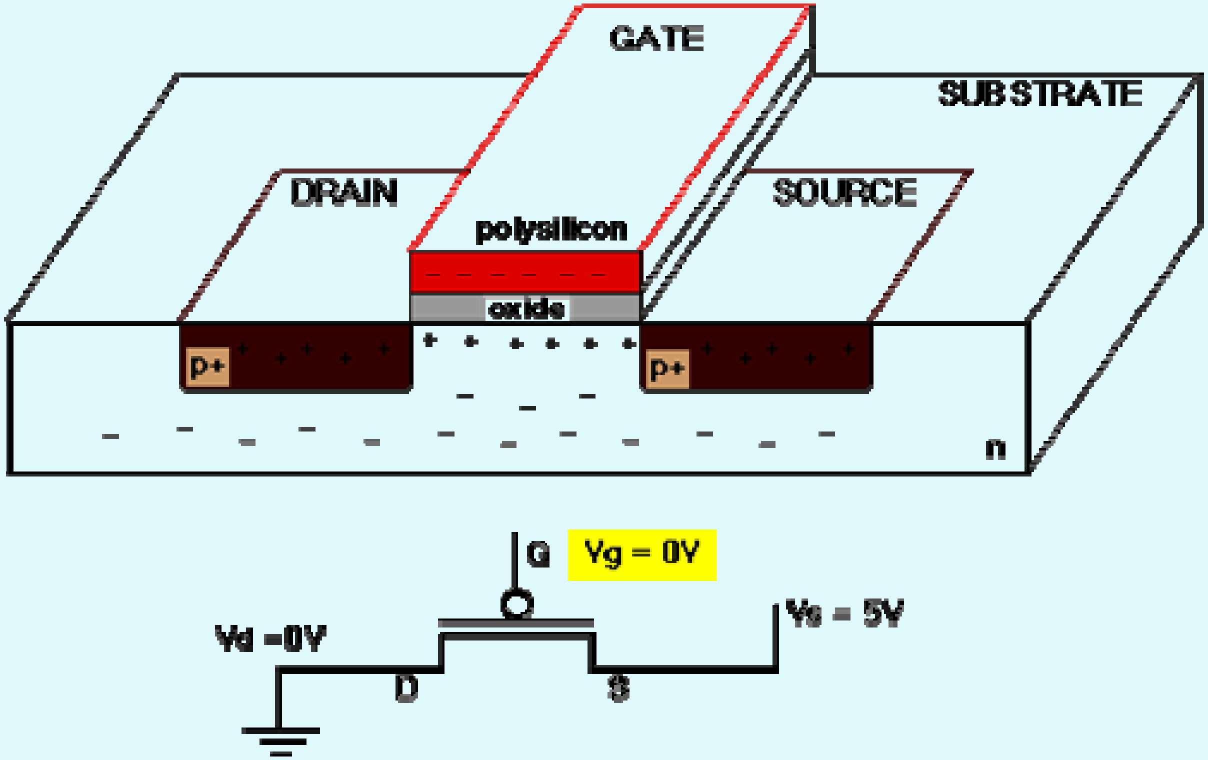
n-diffusion: GREEN

polysilicon ("poly"): RED

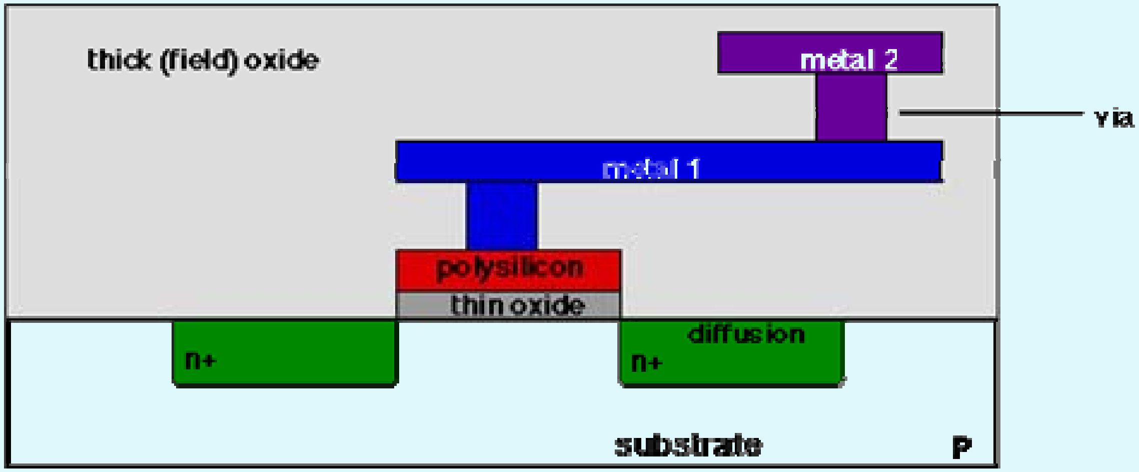
NOTE: Source and drain are physically indistinguishable



p-channel MOS Transistor

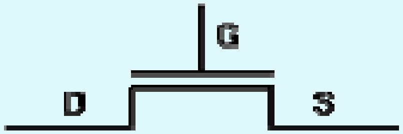


Fabrication Layers



MOS Transistor Behavior

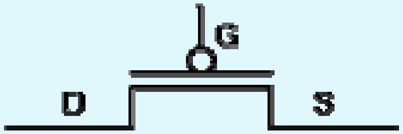
n-channel



Gate HIGH (Vgs = 5V) => CLOSED

Gate LOW (Vgs = 0V) => OPEN

p-channel



Gate HIGH (Vgs = 0V) => OPEN

Gate LOW (Vgs = –5V) => CLOSED

OFF

$V_{gs} < V_t$

$I_{ds} = 0$

$V_{gs} > V_t$

$I_{ds} = 0$

LINEAR

$V_{gs} - V_t > V_{ds}$

$I_{ds} = k'_n (W/L)[(V_{gs} - V_t)V_{ds} - (V_{ds}^2/2)]$

$V_{gs} - V_t < V_{ds}$

$I_{ds} = -k'_p (W/L)[(V_{gs} - V_t)V_{ds} - (V_{ds}^2/2)]$

SATURATION

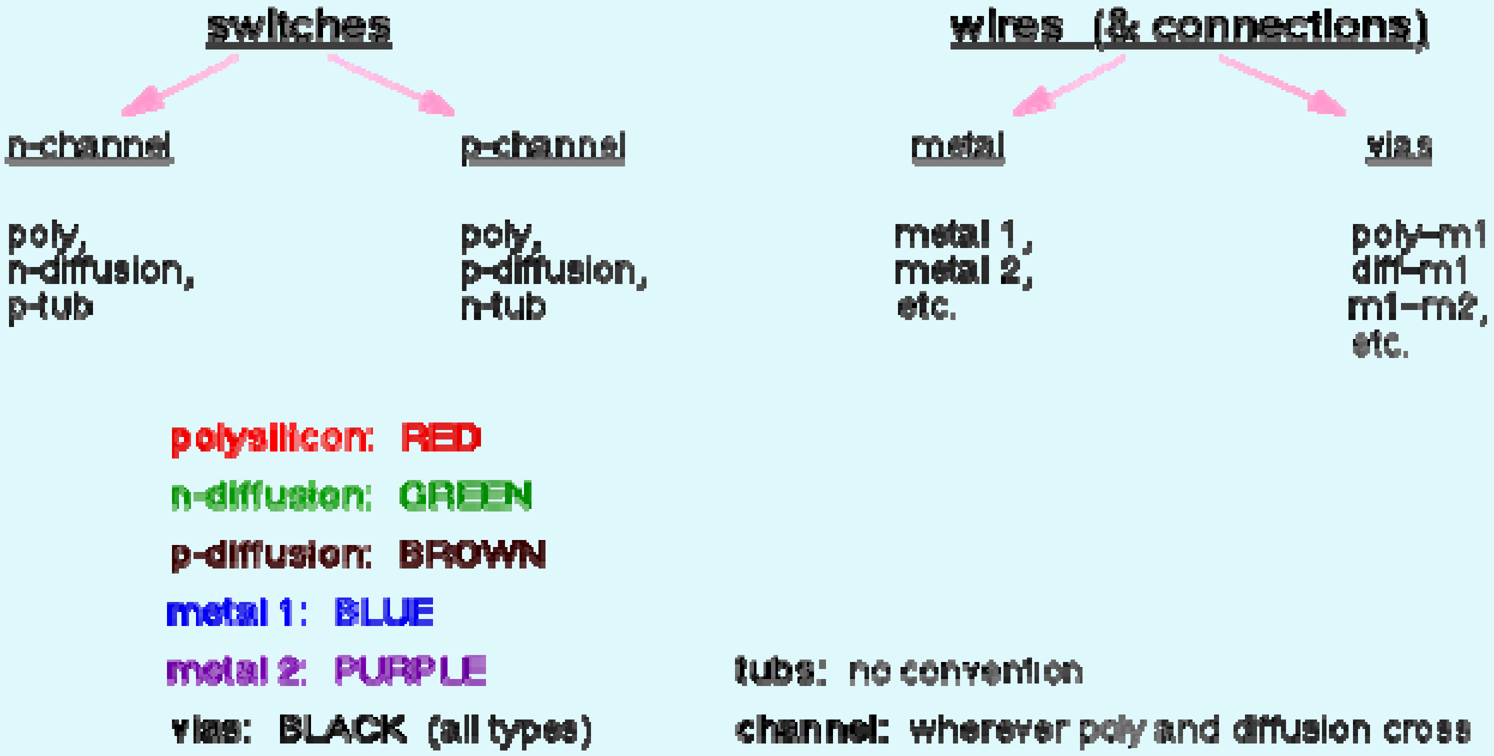
$0 < V_{gs} - V_t \le V_{ds}$

$I_{ds} = (k'_n/2)(W/L)(V_{gs} - V_t)^2$

$0 > V_{gs} - V_t \ge V_{ds}$

$I_{ds} = -(k'_p/2)(W/L)(V_{gs} - V_t)^2$

Summary of VLSI Layers



switches

n-channel

poly,
n-diffusion,
p-tub

p-channel

poly,
p-diffusion,
n-tub

wires (& connections)

metal

metal 1,
metal 2,
etc.

vias

poly–m1
diff–m1
m1–m2,
etc.

polysilicon: RED

n-diffusion: GREEN

p-diffusion: BROWN

metal 1: BLUE

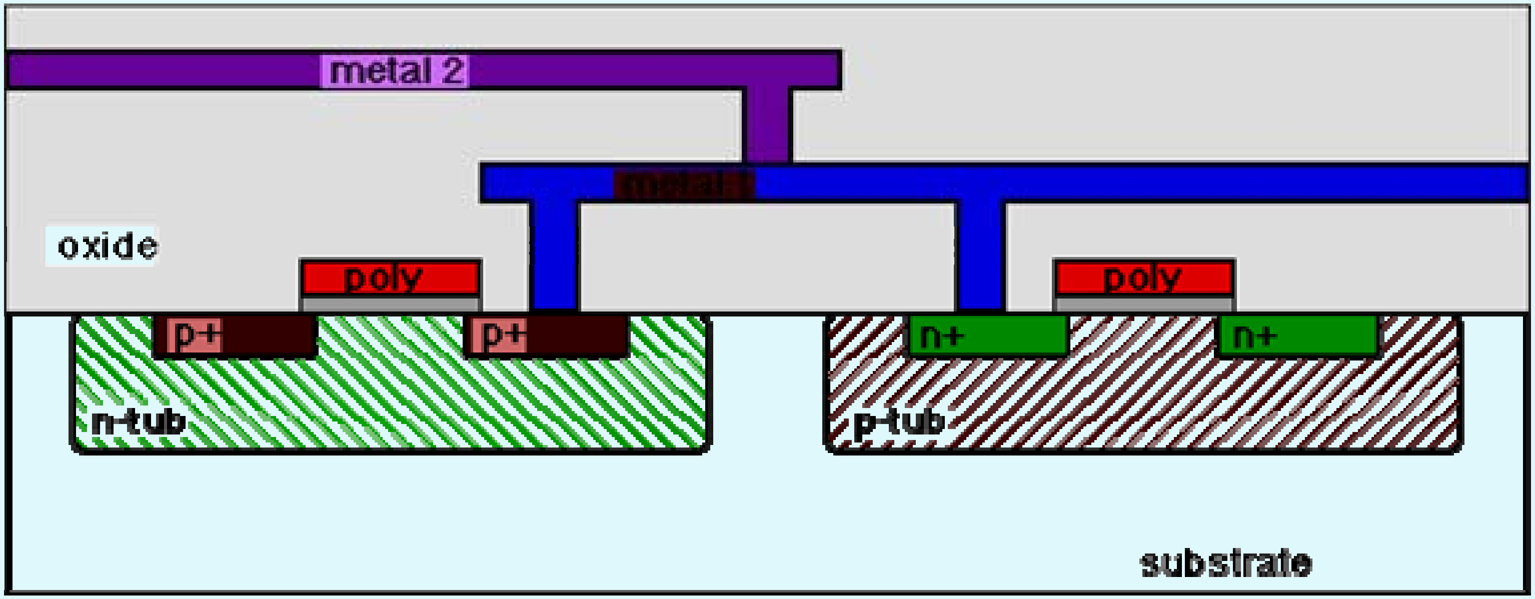
metal 2: PURPLE

vias: BLACK (all types)

tubs: no convention

channel: wherever poly and diffusion cross

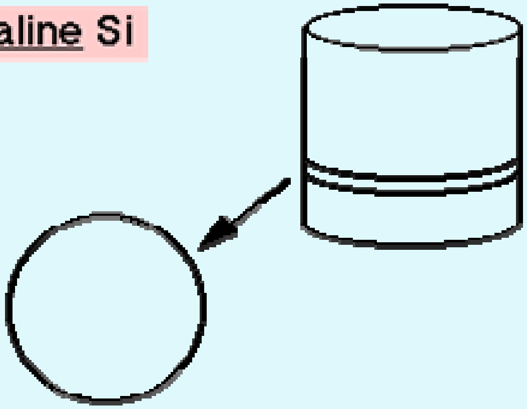
VLSI Fabrication



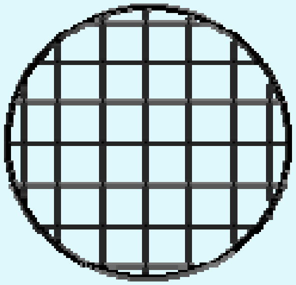
Silicon Wafer

1. Grow in ingot of monocrystaline Si

2. Slice into wafers

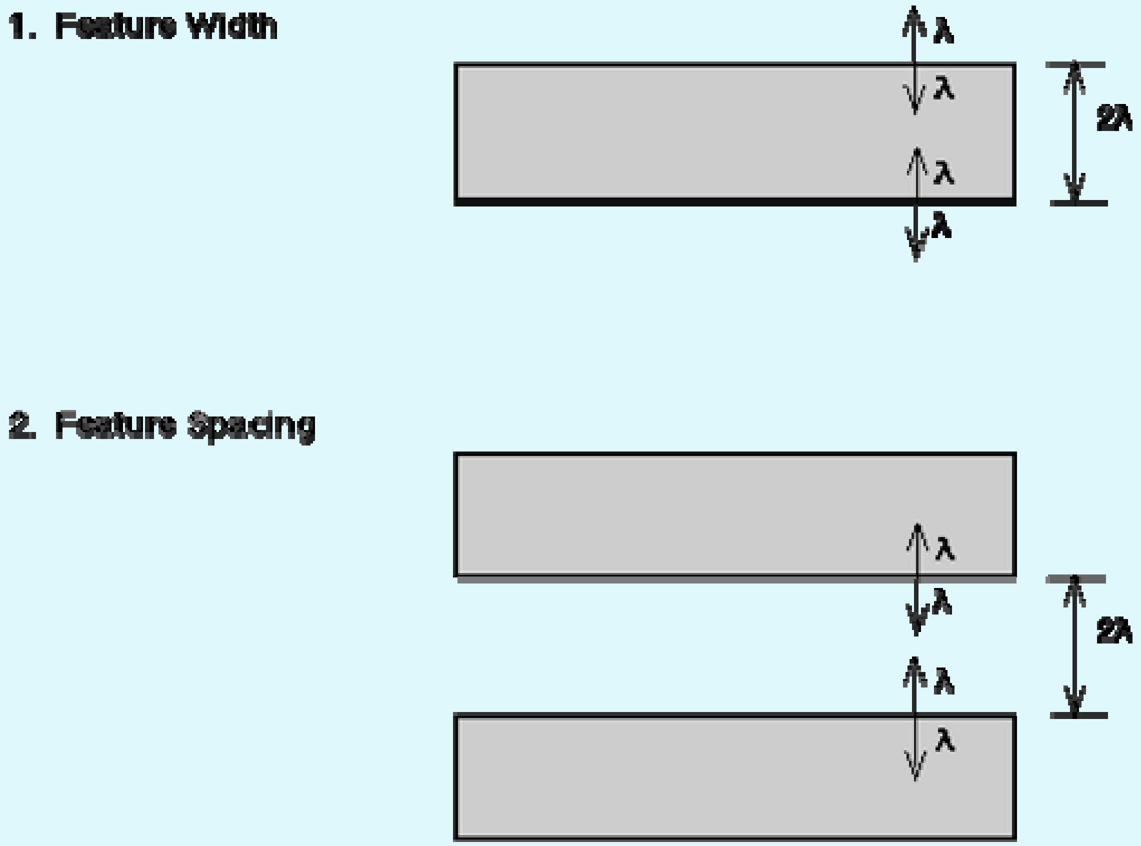


3. Implement repeated copies of the chip



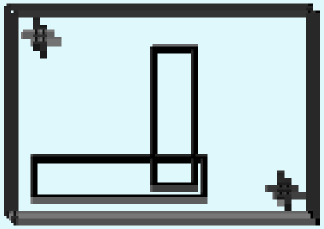
4. Test, cut into die, package

General Design Rules

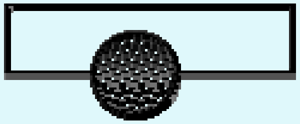


Types of Fabrication Errors

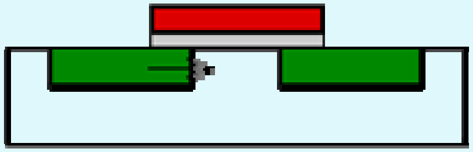
1. Mask Misalignment



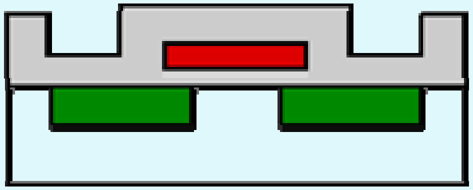
2. Dust



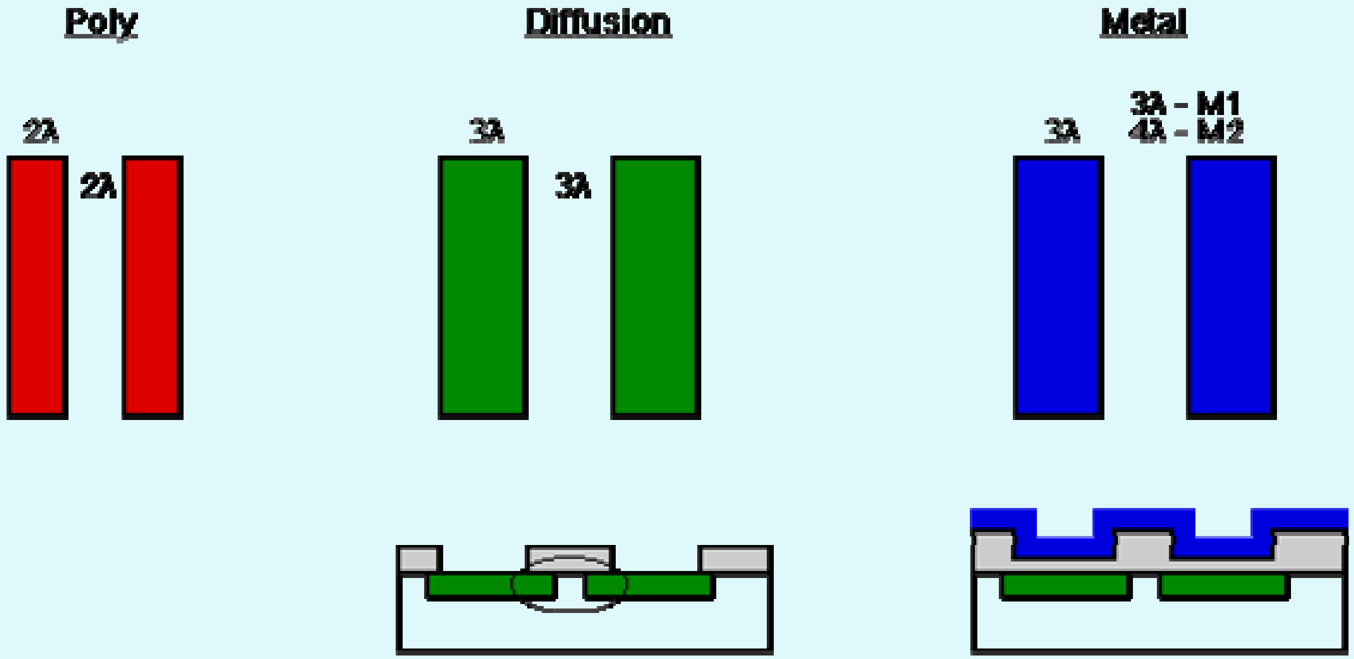
3. Process Parameters



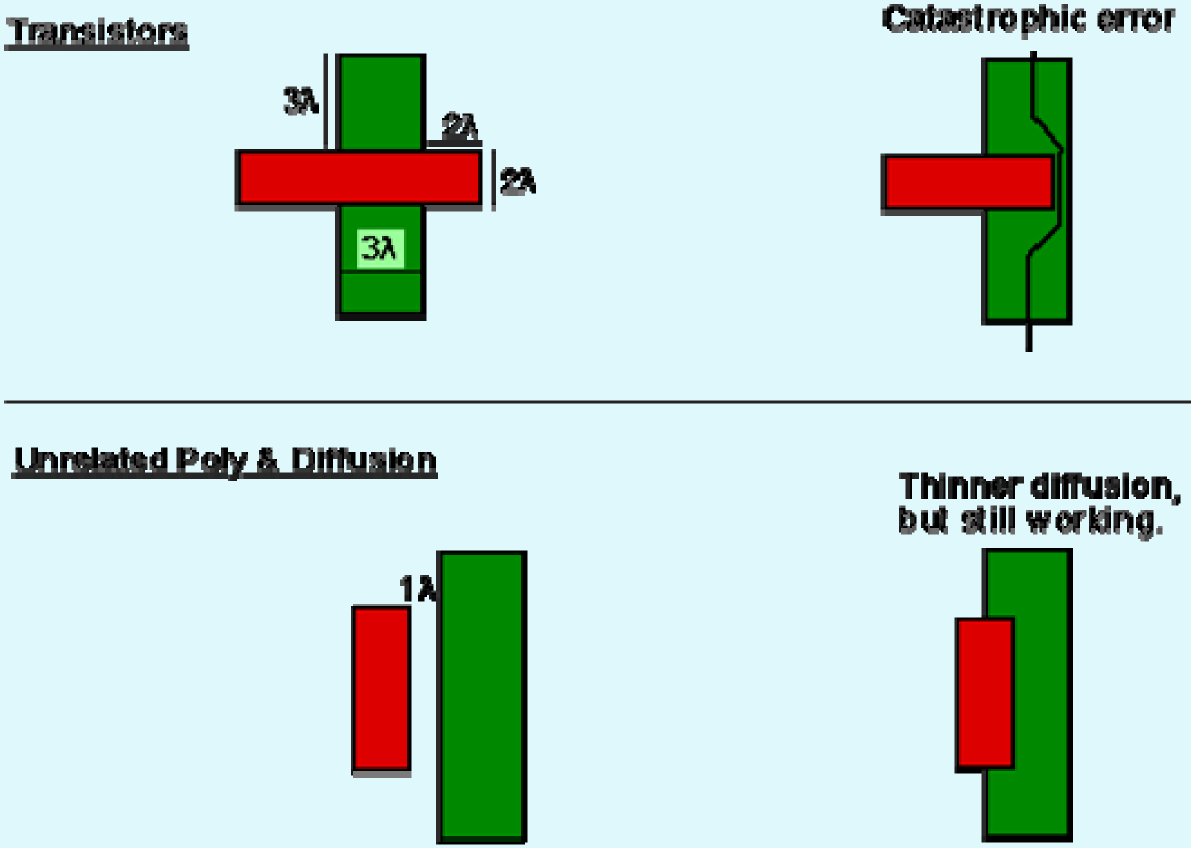
4. Bumpy Surfaces



Width/Spacing Rules (MOSIS)



Poly-Diffusion Interaction

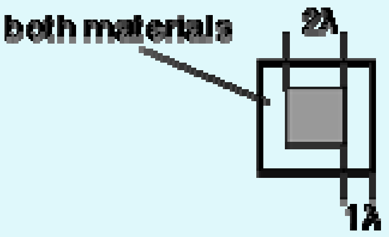


Contacts

M1 contact to p-diffusion
M1 contact to n-diffusion
M1 contact to poly] Contact Mask

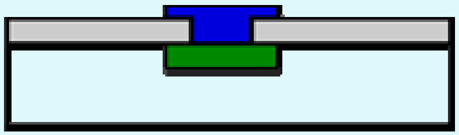
M2 contact to M1 Via Mask

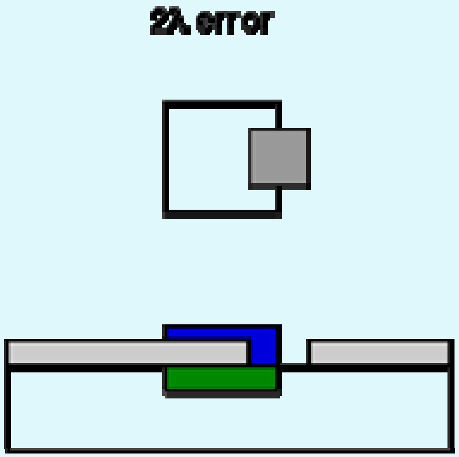
Generally:



Contact hole: 2 x 2

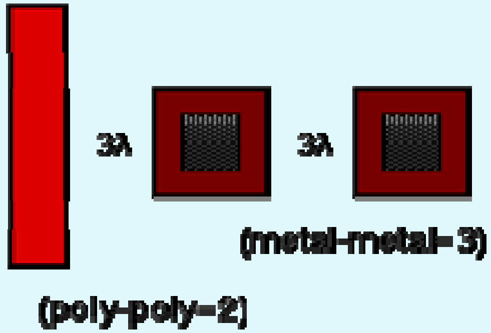
Total size: 4 x 4



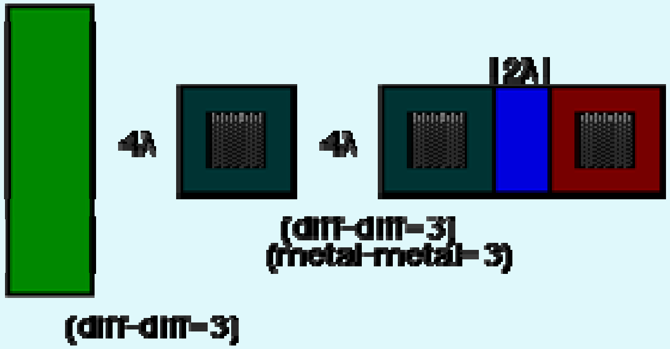


Contact Spacing

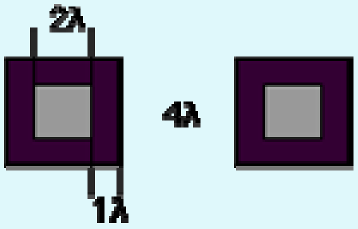
Poly contact



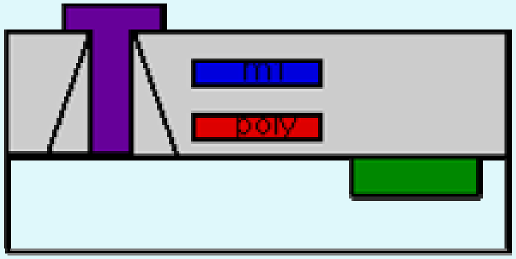
Diffusion contact



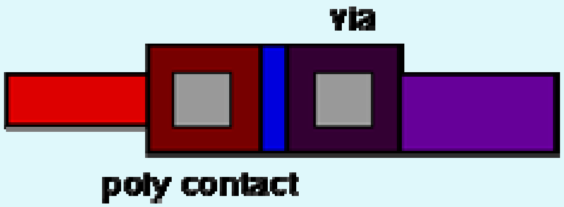
M2 Contact (Via)

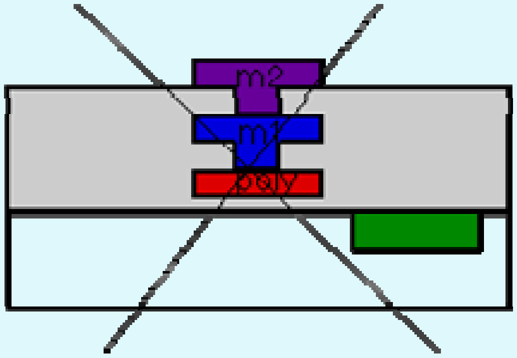


NOTE: M2 can connect only to M1



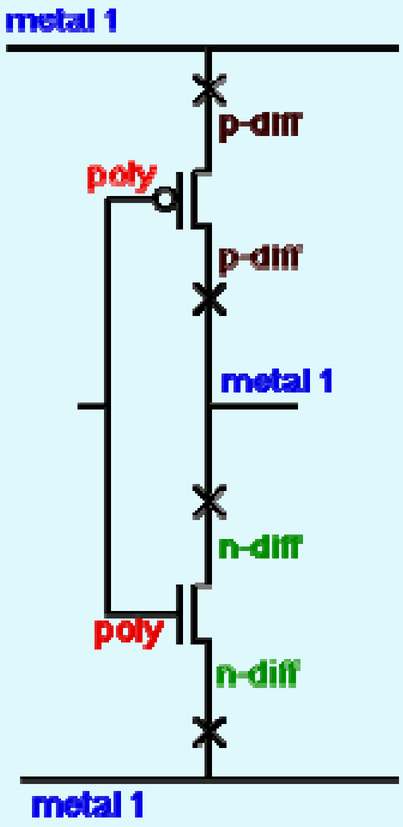
- No via-contact overlap

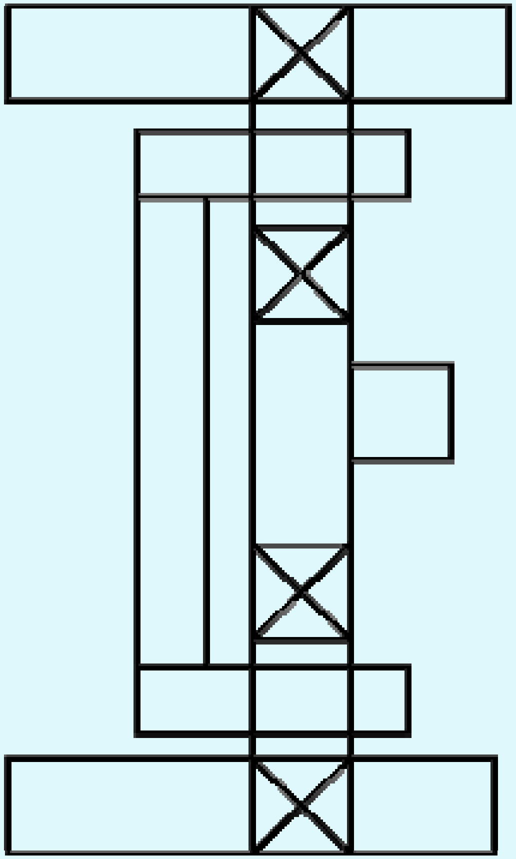




Not allowed in MOSIS rules

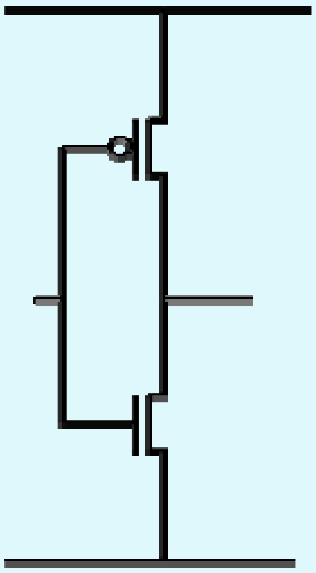
CMOS Layout Example



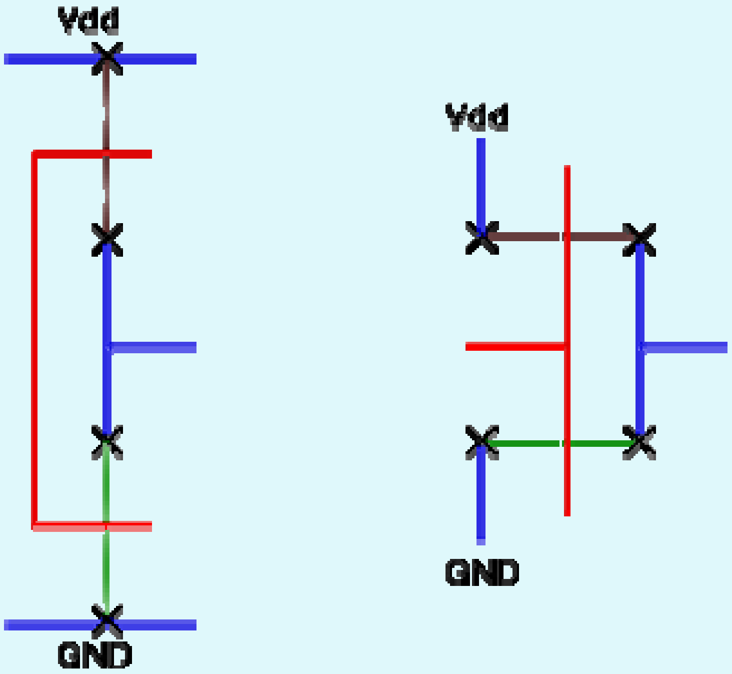


Stick Diagrams

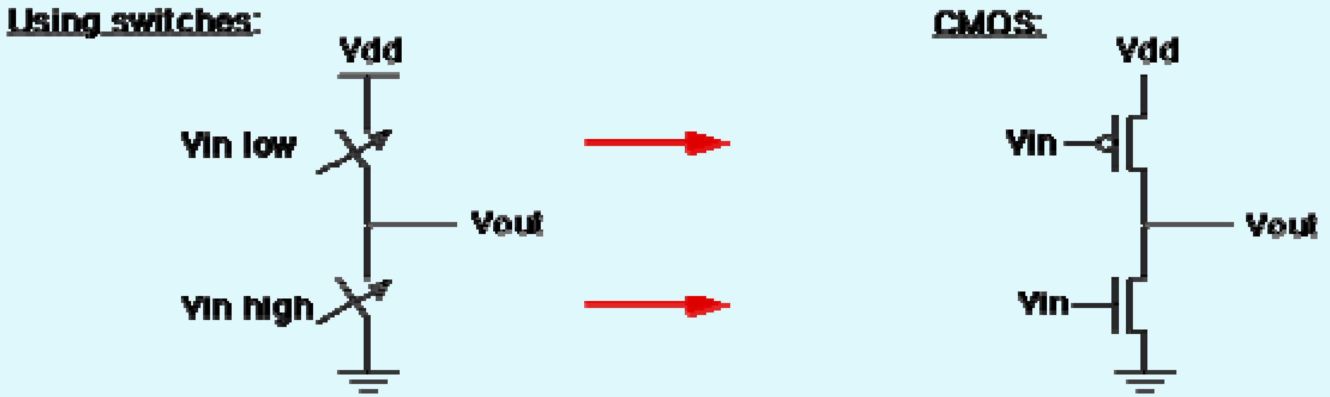
circuit diagram

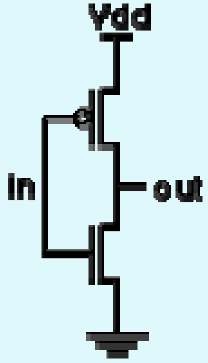


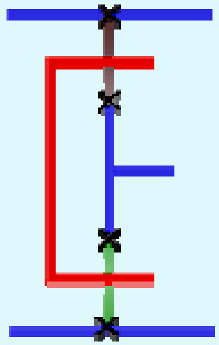
stick diagrams



Static CMOS Inverter







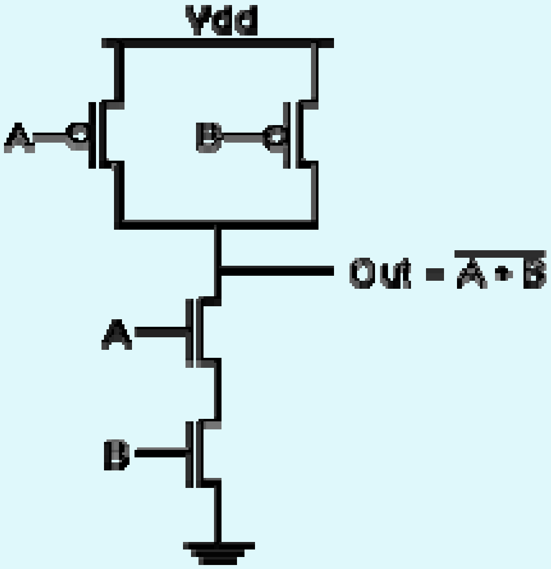
In = 1 (Vdd):

Pull down: ON
Pull up: OFF
Out = 0 (Gnd)

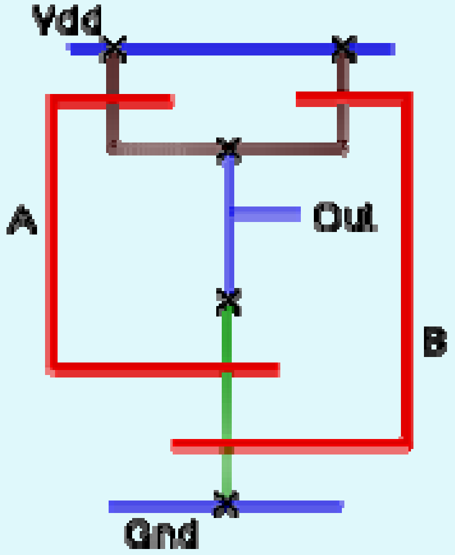
In = 0 (GND):

Pull down: OFF
Pull up: ON
Out = 1 (Vdd)

Static CMOS NAND Gate



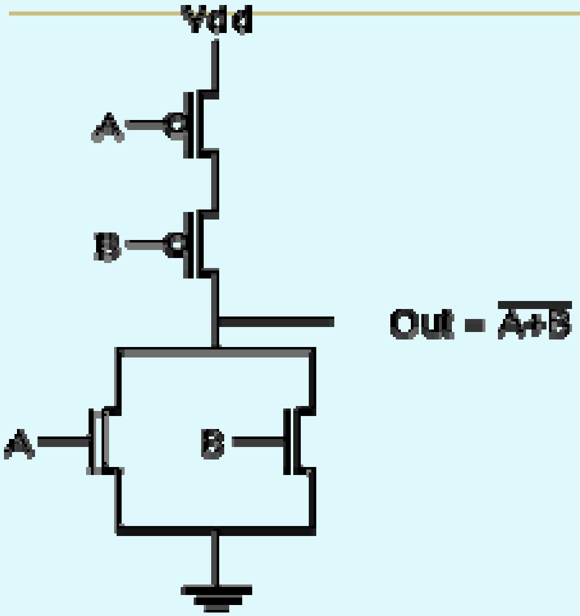
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



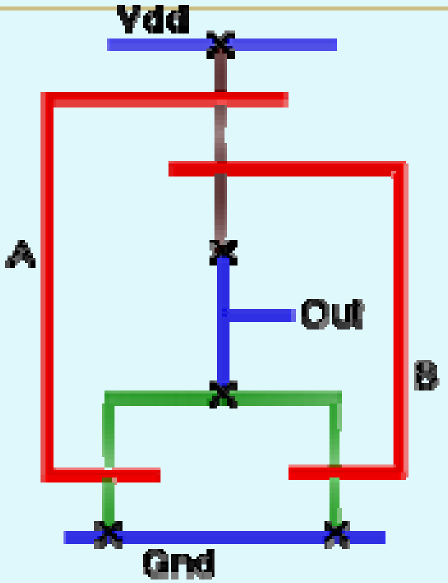
1. Pull-down: Connect to ground if A=1 AND B=1

2. Pull-up: Connect to Vdd if A=0 OR B=0

Static CMOS NOR Gate



A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

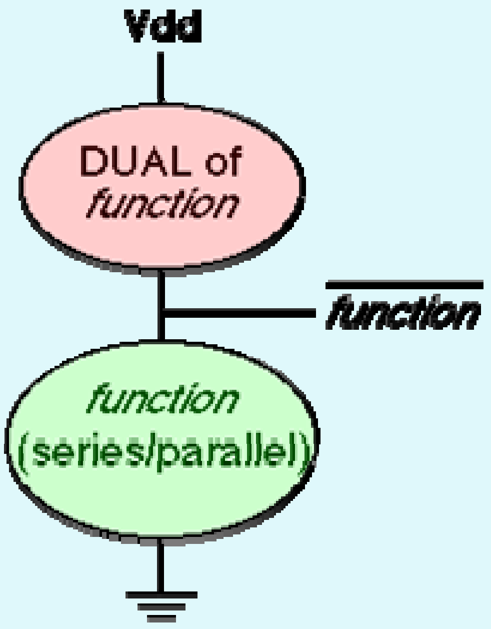


1. Pull-down: Connect to ground if A=1 <u>OR</u> B=1

2. Pull-up: Connect to Vdd if A=0 <u>AND</u> B=0

Static CMOS Design :: General Rule

1. Get into inverted form.
2. Design SERIES/PARALLEL <=> AND/OR circuit for NMOS pull-down
3. Design the Dual circuit for the PMOS pull-up



<u>P-channel side:</u> AND => Parallel
OR => Series

<u>N-channel side:</u> AND => Series
OR => Parallel

Simple Static CMOS Design Example

Example: $f = \overline{a \bullet b + c}$

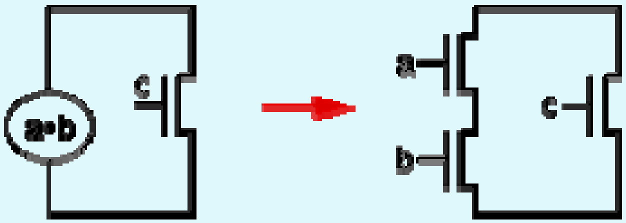
N-channel (pull-down)

Format: $\overline{\text{(Ands,Ors)}}$

$f = \overline{a \bullet b + c}$

(a AND b) OR c

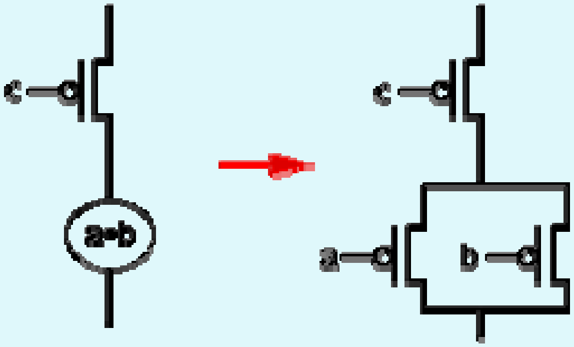
(a SERIES b) PARALLEL c



P-channel (pull-up)

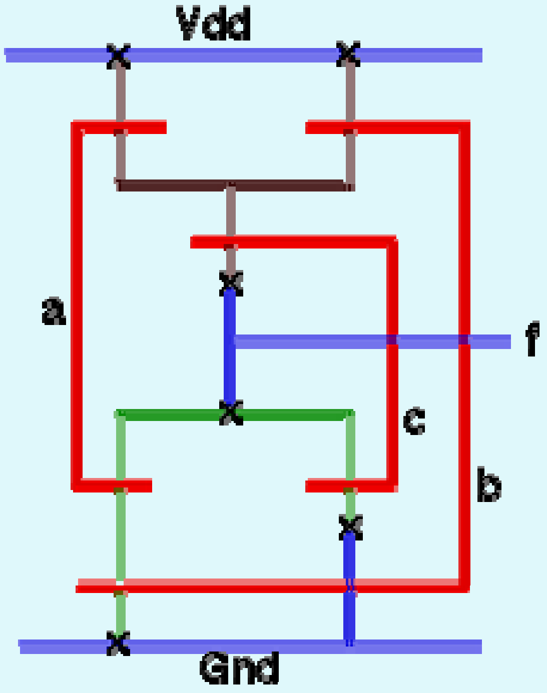
Format: DUAL((a AND b) OR c)

(a PARALLEL b) SERIES c



Static CMOS Design Example Layout

Example: $f = \overline{a \cdot b + c}$



VLSI Design Styles

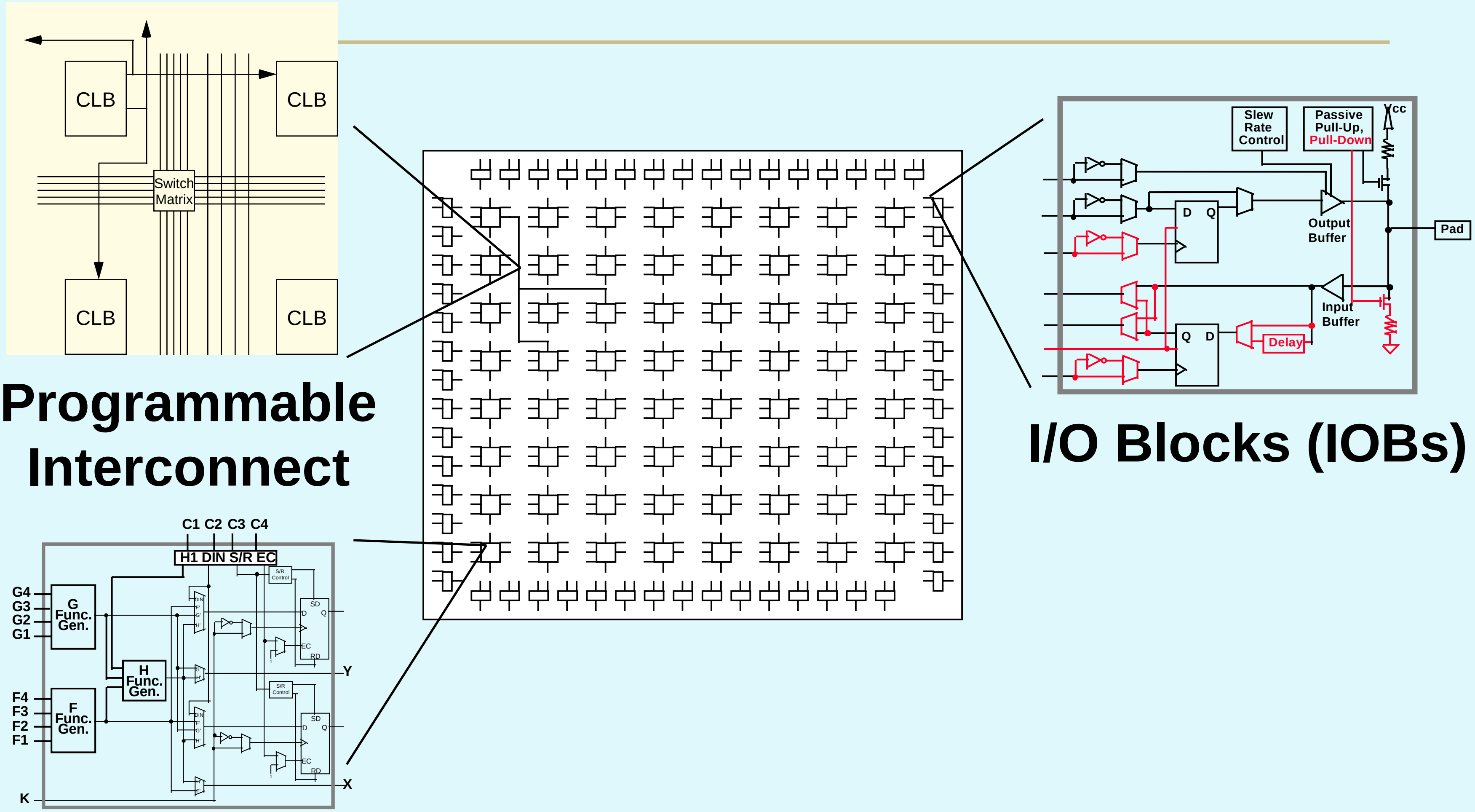
- Programmable Logic Devices
  - Programmable Logic Device (PLD)
  - Field Programmable Gate Array (FPGA)
  - Gate Array
- Standard Cell (Semi-Custom Design)
- Full-Custom Design

Field Programmable Gate Array (FPGA)

Introduction

- User / Field Programmability.
- Array of logic cells connected via routing channels.
- Different types of cells:
  - Special I/O cells.
  - Logic cells.
    - Mainly lookup tables (LUT) with associated registers.
- Interconnection between cells:
  - Using SRAM based switches.
  - Using antifuse elements.

Xilinx XC4000 Architecture

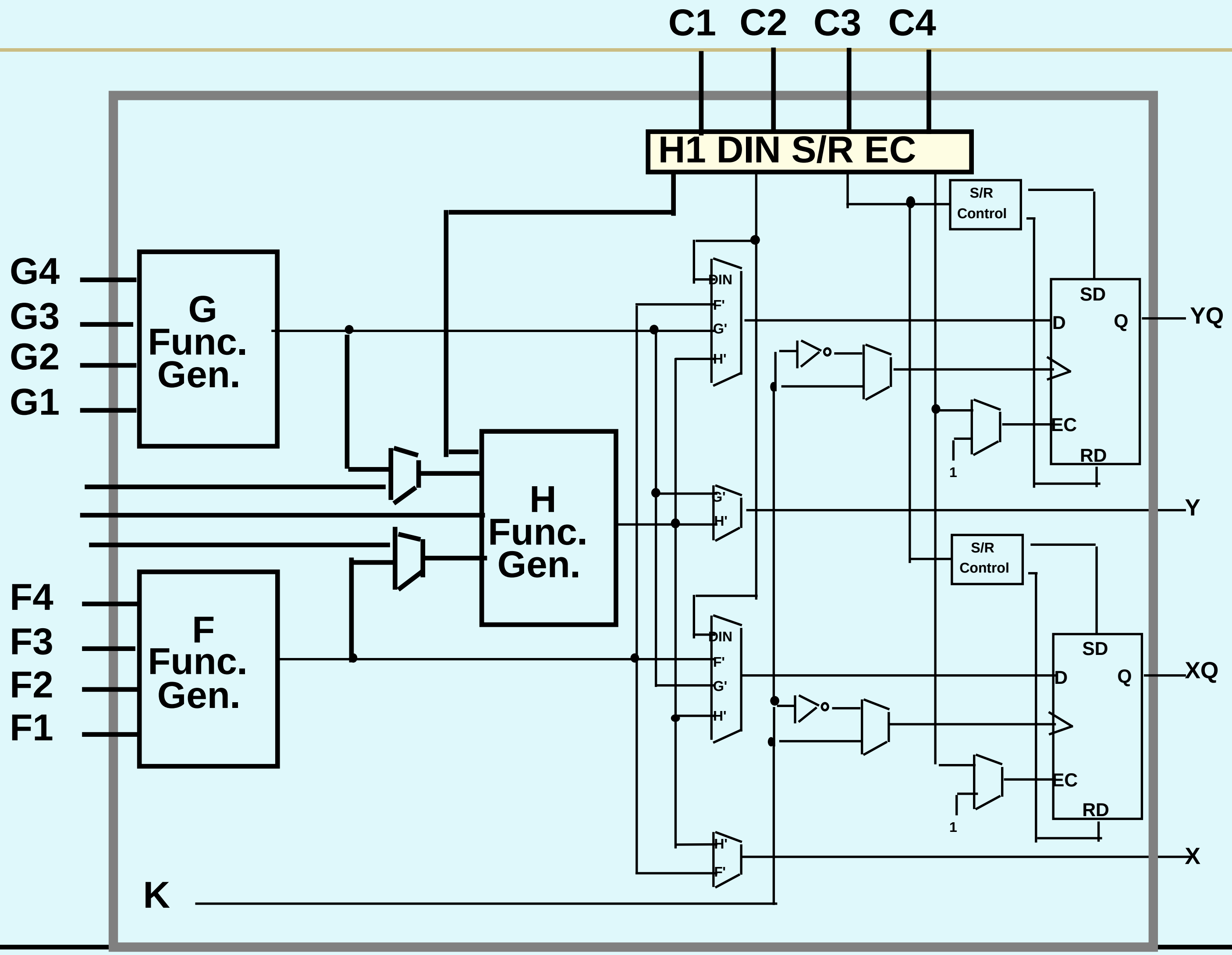


Programmable Interconnect

I/O Blocks (IOBs)

Configurable Logic Blocks (CLBs)

XC4000E Configurable Logic Blocks

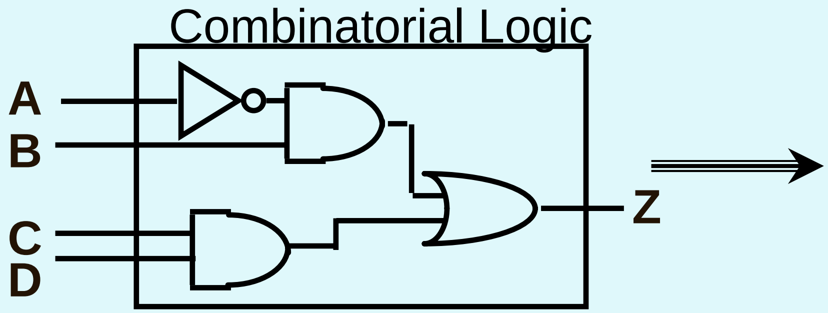


CLB Functionalities

- Two 4-input function generators
  - Implemented using Lookup Tables using 16x1 RAM.
  - Can also implement 16x1 memory.
- Two Registers
  - Each can be configured as flip-flop or latch.
  - Independent clock polarity.
  - Synchronous and asynchronous Set / Reset.

Look Up Tables

- Combinatorial Logic is stored in 16x1 SRAM Look Up Tables (LUTs) in a CLB
- Example:

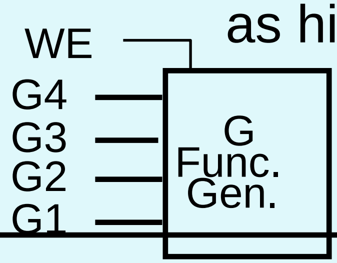


Look Up Table

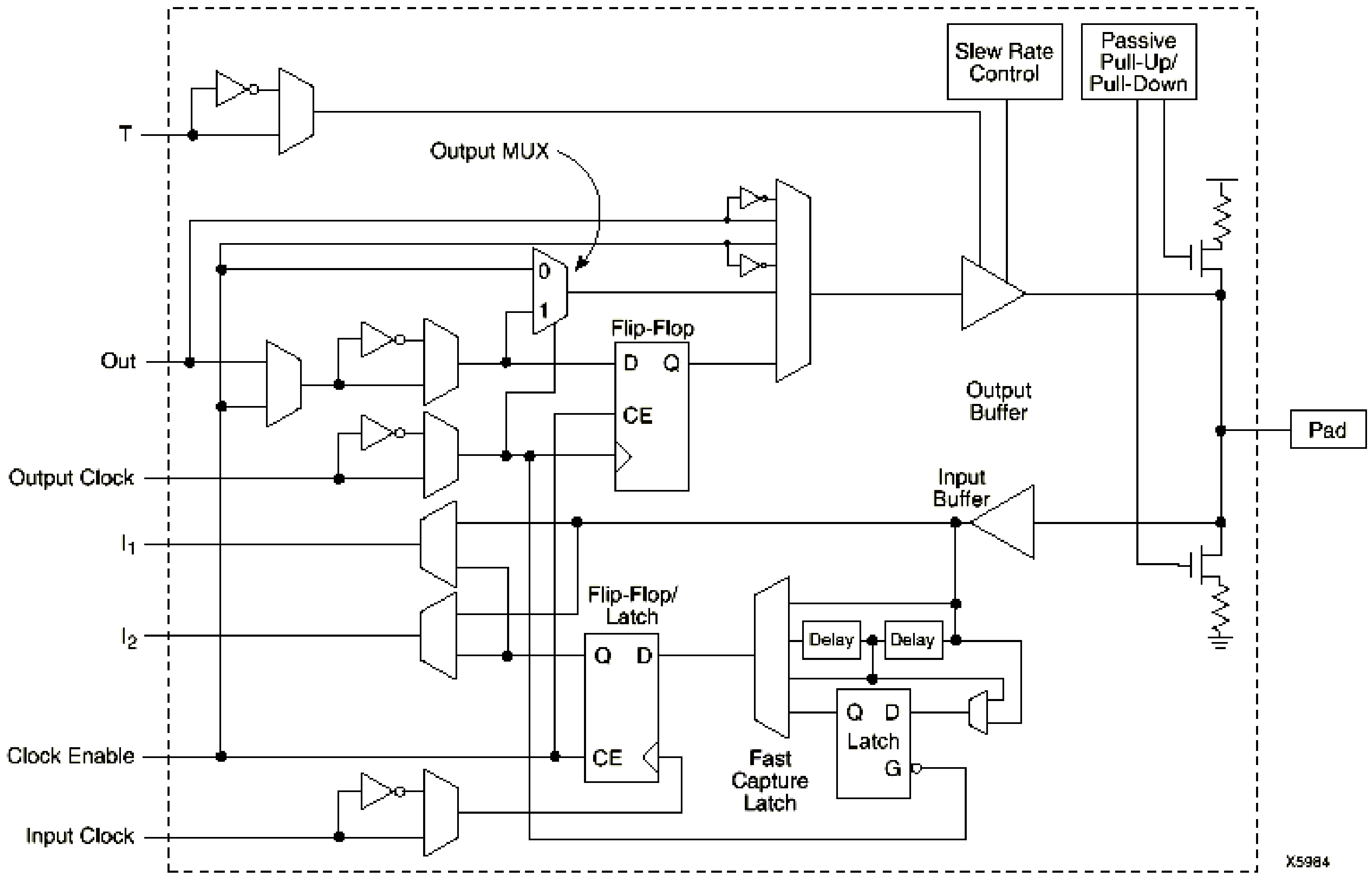
4-bit address

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

- Capacity is limited by number of inputs, not complexity
- Choose to use each function generator as 4 input logic (LUT) or as high speed sync.dual port RAM



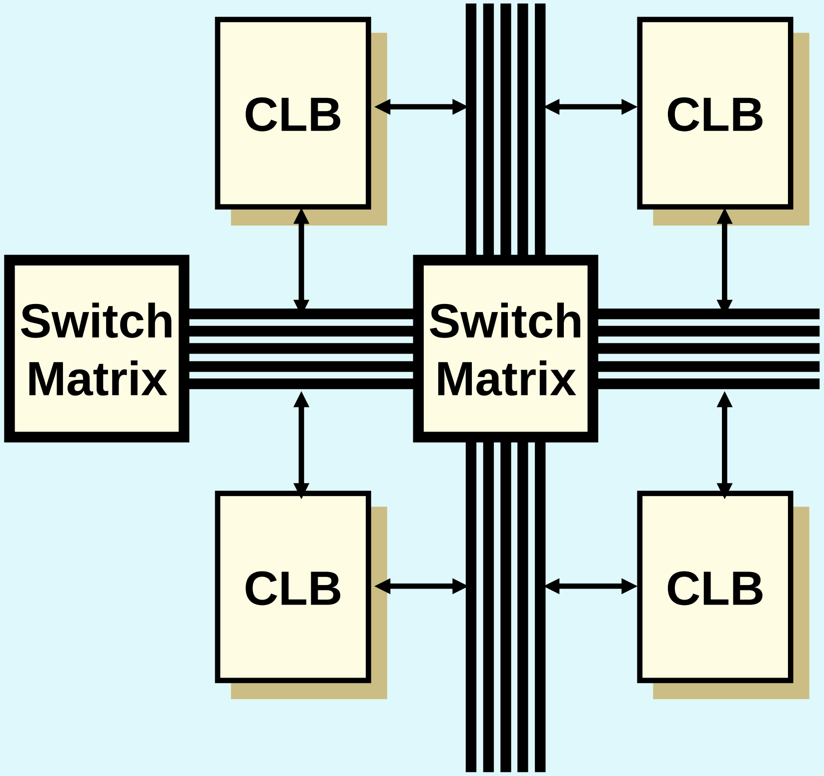
XC4000X I/O Block Diagram



Xilinx FPGA Routing

1) Fast Direct Interconnect - CLB to CLB

2) General Purpose Interconnect - Uses switch matrix



FPGA Design Flow

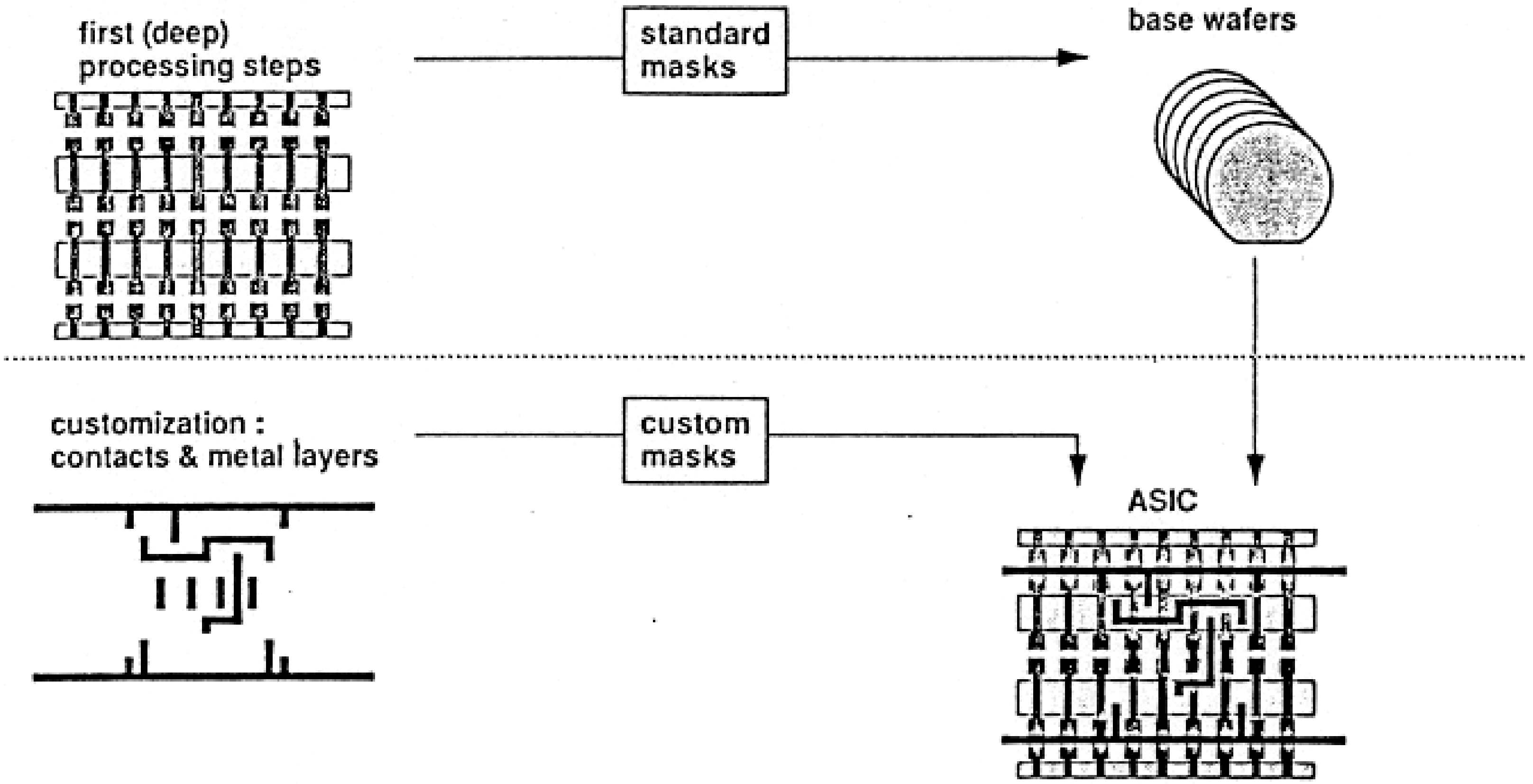
- Design Entry
  - In schematic, VHDL, or Verilog.
- Implementation
  - Placement & Routing
  - Bitstream generation
  - Analyze timing, view layout, simulation, etc.
- Download
  - Directly to Xilinx hardware devices with unlimited reconfigurations.

Gate Array

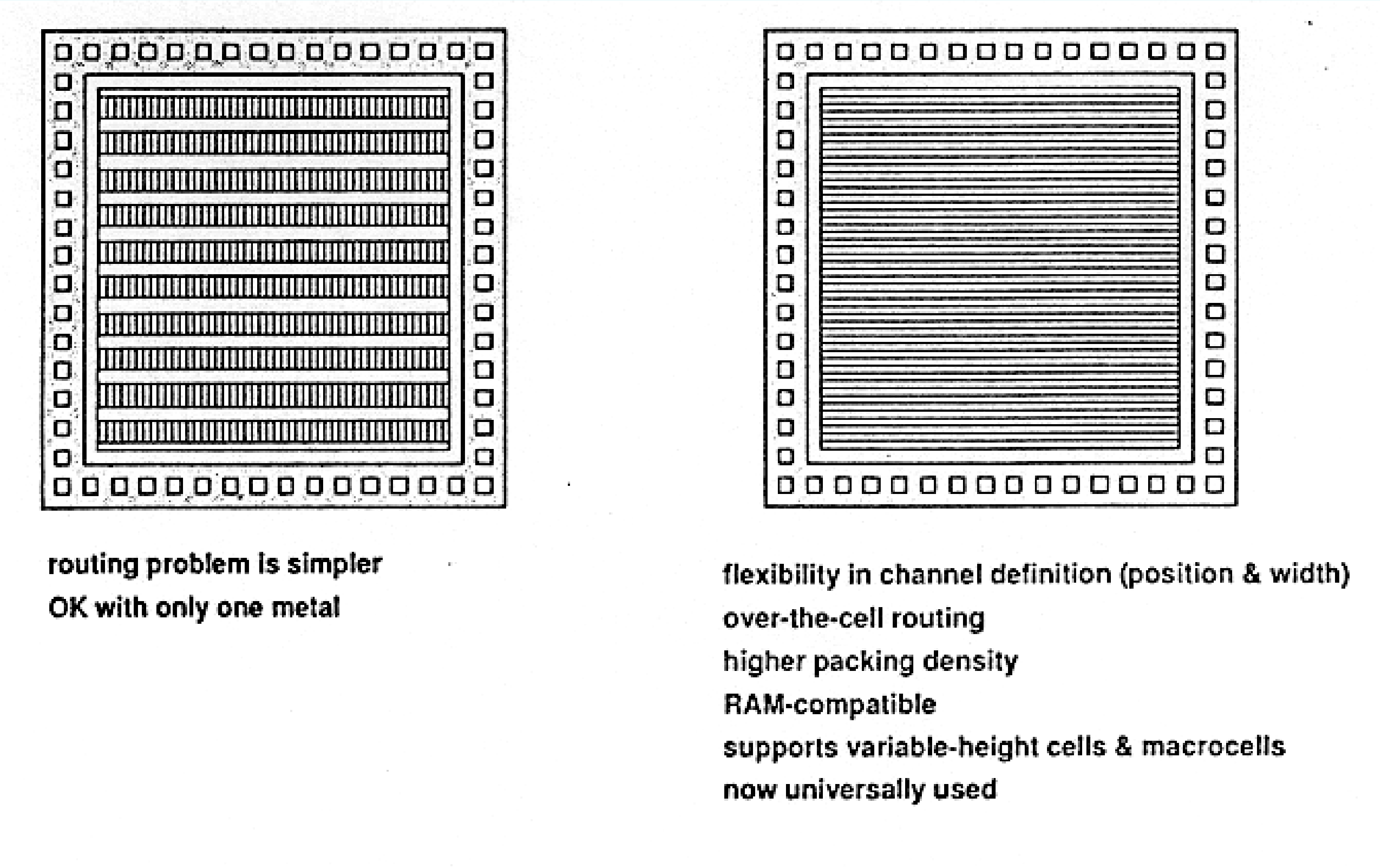
Introduction

- In view of the fast prototyping capability, the gate array (GA) comes after the FPGA.
  - Design implementation of
    - FPGA chip is done with user programming,
    - Gate array is done with metal mask design and processing.
- Gate array implementation requires a two-step manufacturing process:
  a) The first phase, which is based on generic (standard) masks, results in an array of uncommitted transistors on each GA chip.
  b) These uncommitted chips can be customized later, which is completed by defining the metal interconnects between the transistors of the array.

two-step manufacture :



Channeled vs. Channel-less (SoG) Approaches



routing problem is simpler

OK with only one metal

flexibility in channel definition (position & width)

over-the-cell routing

higher packing density

RAM-compatible

supports variable-height cells & macrocells

now universally used

- **The GA chip utilization factor is higher than that of FPGA.**
  - **The used chip area divided by the total chip area.**
- **Chip speed is also higher.**
  - **More customized design can be achieved with metal mask designs.**
- **Current gate array chips can implement as many as hundreds of thousands of logic gates.**

Standard Cell Based Design

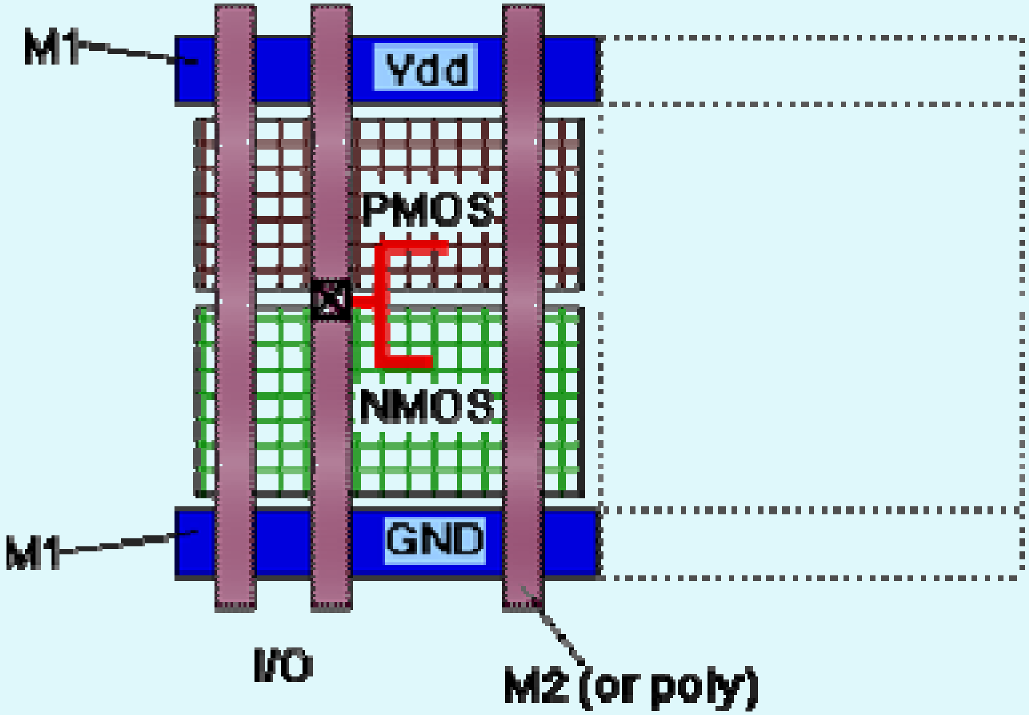
Introduction

- One of the most prevalent custom design styles.
  - Also called semi-custom design style.
  - Requires developing full custom mask set.
- Basic idea:
  - All of the commonly used logic cells are developed, characterized, and stored in a standard cell library.
  - A typical library may contain a few hundred cells.
    - Inverters, NAND gates, NOR gates, complex AOI, OAI gates, D-latches, and flip-flops.

Characteristic of the Cells

- Each cell is designed with a fixed height.
  - To enable automated placement of the cells, and
  - Routing of inter-cell connections.
  - A number of cells can be abutted side-by-side to form rows.
- The power and ground rails typically run parallel to upper and lower boundaries of cell.
  - Neighboring cells share a common power and ground bus.
  - nMOS transistors are located closer to the ground rail while the pMOS transistors are placed closer to the power rail.
- The input and output pins are located on the upper and lower boundaries of the cell.

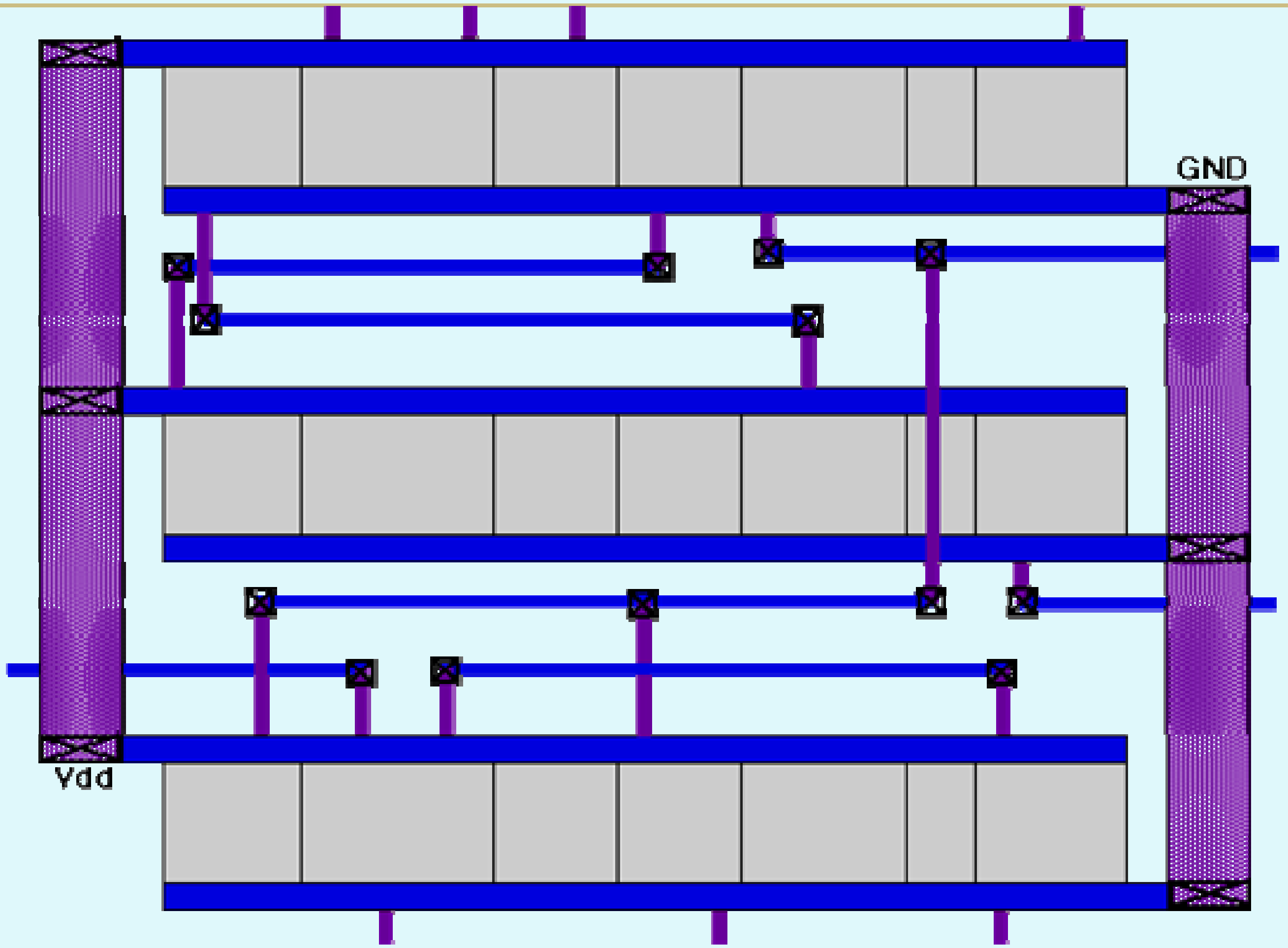
Standard Cells



Made to stack side-by-side

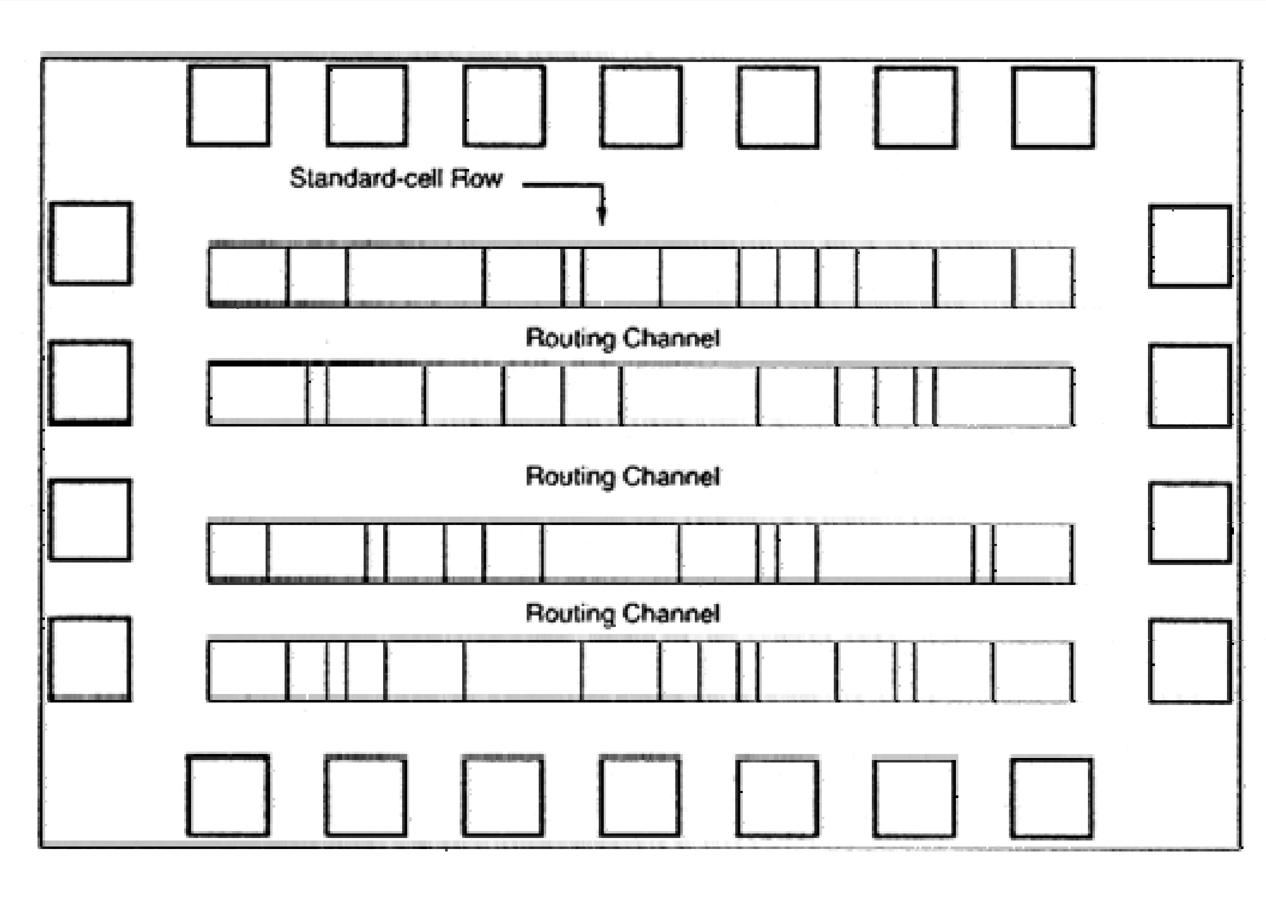
- Fixed height
- Width can vary
- Can abut at Vdd & GND
(without any design rule errors)

Standard Cell Layout



Floorplan for Standard Cell Design

- Inside the I/O frame which is reserved for I/O cells, the chip area contains rows or columns of standard cells.
  - Between cell rows are channels for dedicated inter-cell routing.
  - Over-the-cell routing is also possible.
- The physical design and layout of logic cells ensure that
  - When placed into rows, their heights match.
  - Neighboring cells can be abutted side-by-side, which provides natural connections for power and ground lines in each row.


Standard-cell Row
Routing Channel
Routing Channel
Routing Channel

Full Custom Design

Introduction

- The standard-cells based design is often called semi custom design.
  - The cells are pre-designed for general use and the same cells are utilized in many different chip designs.
- In the full custom design, the entire mask design is done anew without use of any library.
  - The development cost of such a design style is prohibitively high.
  - The concept of design reuse is becoming popular to reduce design cycle time and cost.

Contd.

- The most rigorous full custom design can be the design of a memory cell.
  - Static or dynamic.
  - Since the same layout design is replicated, there would not be any alternative to high density memory chip design.
- For logic chip design, a good compromise can be achieved by using a combination of different design styles on the same chip.
  - Standard cells, data-path cells and PLAs.

- In real full-custom layout in which the geometry, orientation and placement of every transistor is done individually by the designer,
  - Design productivity is usually very low.
    - Typically 10 to 20 transistors per day, per designer.
- In digital CMOS VLSI, full-custom design is rarely used due to the high labor cost.
  - Exceptions to this include the design of high-volume products such as memory chips, high-performance microprocessors and FPGA masters.

Comparison Among Various Design Styles

	Design Style			
	FPGA	Gate array	Standard cell	Full custom
Cell size	Fixed	Fixed	Fixed height	Variable
Cell type	Programmable	Fixed	Variable	Variable
Cell placement	Fixed	Fixed	In row	Variable
Interconnect	Programmable	Variable	Variable	Variable
Design time	Very fast	Fast	Medium	Slow