# Digital IC & Sytems Design

Iain McNally

≈ 10 lectures

Koushik Maharatna

≈ 12 lectures

Basel Halak

≈ 12 lectures





- Assessment

| | | |
|---|---|---|
| 10% | Coursework | L-Edit Gate Design (BIM) |
| 90% | Examination | |

- Books

**Integrated Circuit Design**

a.k.a. Principles of CMOS VLSI Design - A Circuits and Systems Perspective

Neil Weste & David Harris

Pearson, 2011

**Digital System Design with SystemVerilog**

Mark Zwolinski

Pearson Prentice-Hall, 2010





Iain McNally

## Integrated Circuit Design

- Content
  - Introduction
  - Overview of Technologies
  - Layout
  - CMOS Processing
  - Design Rules and Abstraction
  - Cell Design and Euler Paths
  - System Design using Standard Cells
  - Wider View
- Notes & Resources

`http://users.ecs.soton.ac.uk/bim/notes/icd`



## History

**1947** First Transistor

John Bardeen, Walter Brattain, and William Shockley (Bell Labs)

**1952** Integrated Circuits Proposed

Geoffrey Dummer (Royal Radar Establishment) - *prototype failed...*

**1958** First Integrated Circuit

Jack Kilby (Texas Instruments) - *Co-inventor*

**1959** First Planar Integrated Circuit

Robert Noyce (Fairchild) - *Co-inventor*

**1961** First Commercial ICs

Simple logic functions from TI and Fairchild

**1965** Moore's Law

Gordon Moore (Fairchild) observes the trends in integration.

## Moore's Law

Predicts exponential growth in the number of components per chip.

### **1965 - 1975** Doubling Every Year

In 1965 Gordon Moore observed that the number of components per chip had doubled every year since 1959 and predicted that the trend would continue through to 1975.

Moore describes his initial growth predictions as "ridiculously precise".

### **1975 - 201?** Doubling Every Two Years

In 1975 Moore revised growth predictions to doubling every two years.

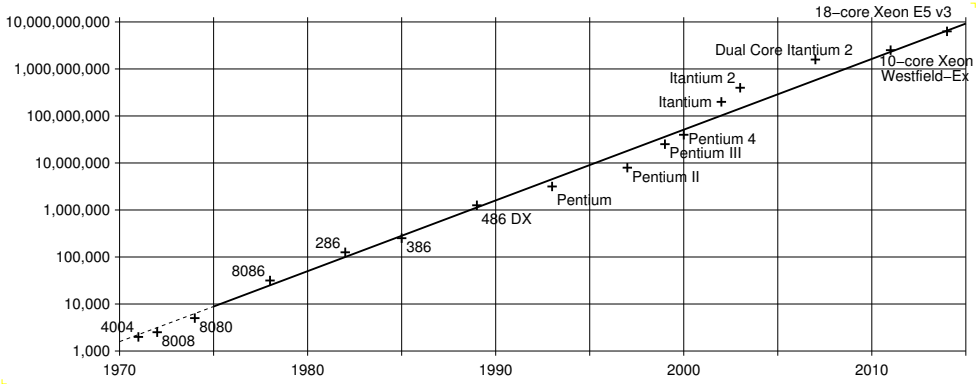
Growth would now depend only on process improvements rather than on more efficient packing of components.

In 2000 he predicted that the growth would continue at the same rate for another 10-15 years before slowing due to physical limits.





## Moore's Law at Intel[1]




[1]Intel was founded by Gordon Moore and Robert Noyce from Fairchild





## Moore's Law; a Self-fulfilling Prophesy

The whole industry uses the Moore's Law curve to plan new fabrication facilities.

**Slower** - wasted investment

Must keep up with the Joneses[2].

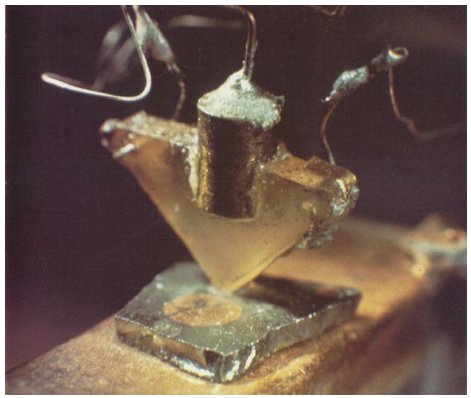
**Faster** - too costly

Cost of capital equipment to build ICs doubles approximately every 4 years.

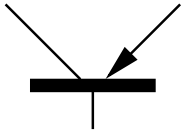
*Moore's law is not yet dead, although there are worries that below 20nm, clever processing required for smaller transistors means that cost per transistor will go up rather than down.*

[2]or the Intels

1947 Point Contact transistor

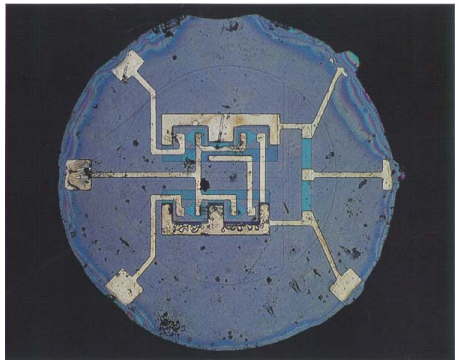


Source: Bell Labs

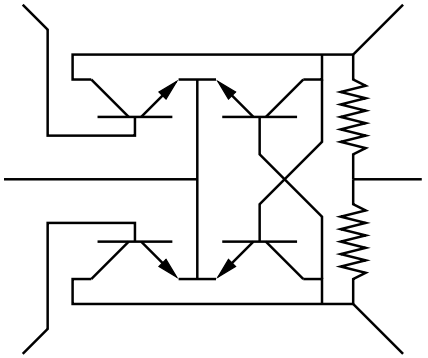


1961 Fairchild Bipolar RTL RS Flip–Flop

(4 Transistors)

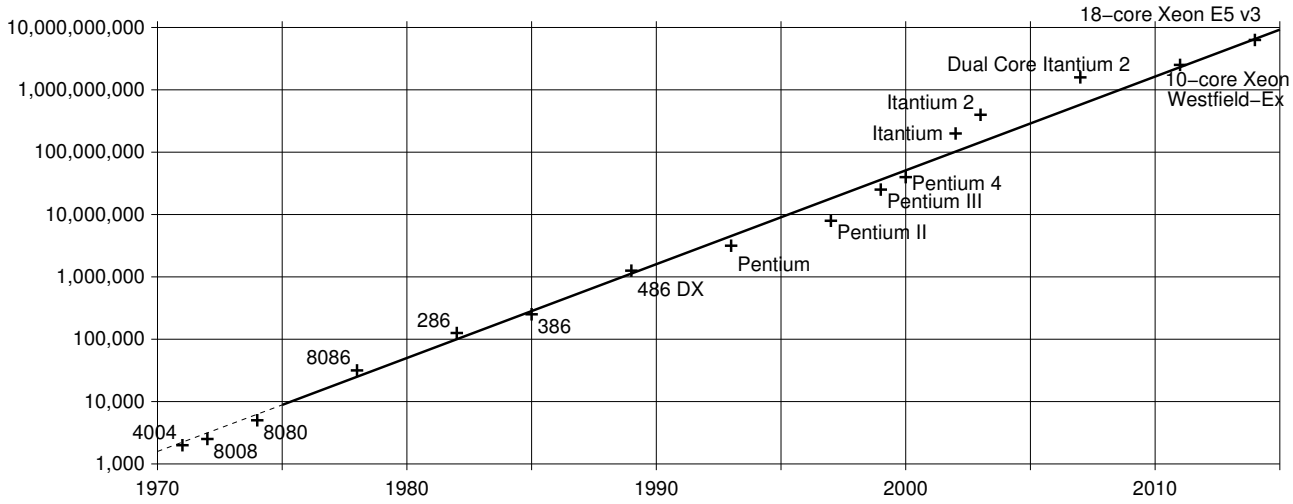


Source: Fairchild



Moore's Law (1965) Number of transistor has doubled every year and will continue to do so until 1975

Moore's Law (1975) Number of transistors will double every two years

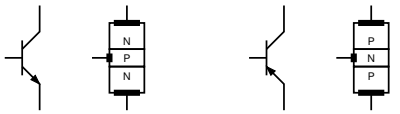



Self–fulfilling Prophecy

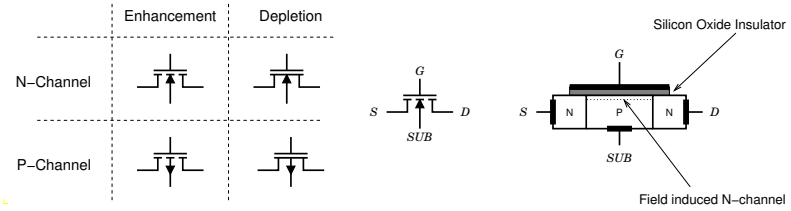
## Components for Logic

Diode


Bipolar Transistors



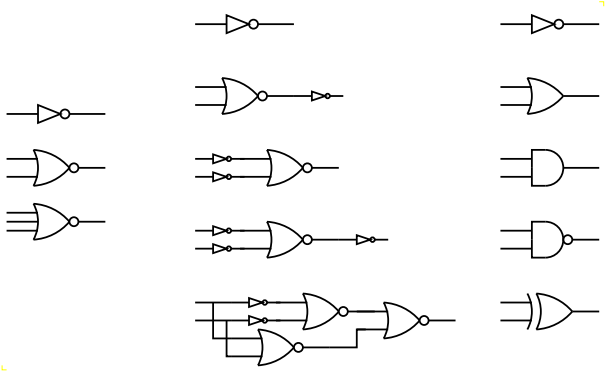

MOS Transistors








All functions can be realized using only the NOR gates[1] available in the RTL logic family.[2]

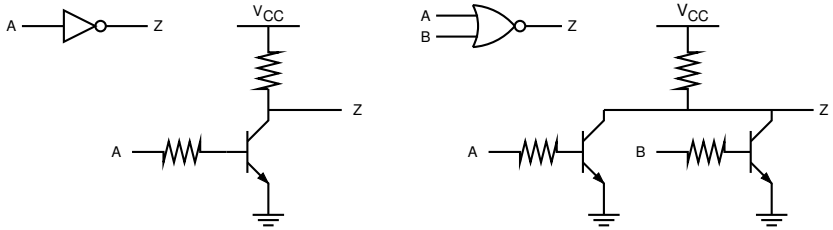


[1]Note that an inverter is a special case of a NOR gate with only one input.
[2]NAND gates could be used instead for logic families which support only NAND gates.





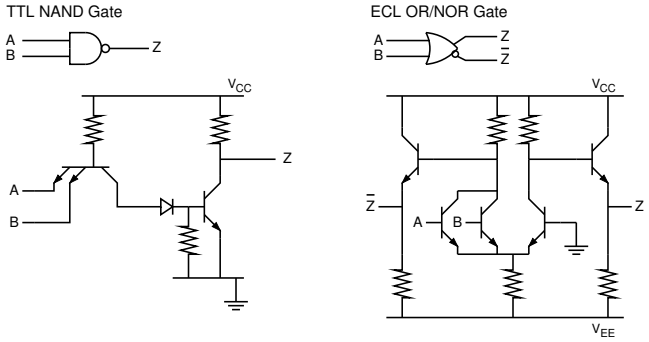
RTL Inverter and NOR gate








## Other Bipolar Technologies

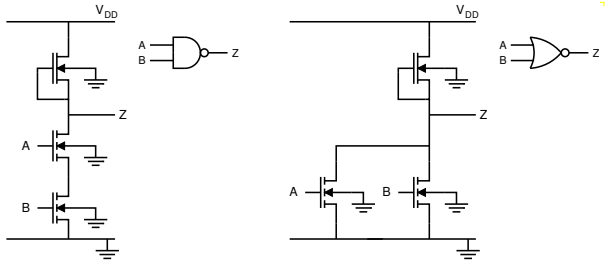



- TTL gives faster switching than RTL at the expense of greater complexity[3]. The characteristic multi-emitter transistor reduces the overall component count.
- ECL is a very high speed, high power, non-saturating technology.

[3]Most TTL families are more complex than the basic version shown here

## Overview of Technologies

NMOS - a VLSI technology.



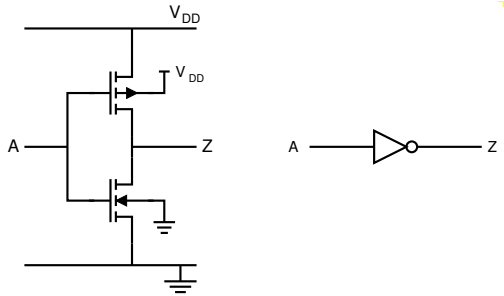

- Circuit function determined by series/parallel combination of devices.
- Depletion transistor acts as non-linear load resistor.
  Resistance increases as the enhancement device turns on, thus reducing power consumption.
- The low output voltage is determined by the size ratio of the devices.



## Overview of Technologies

### CMOS logic

CMOS - *state of the art* VLSI.

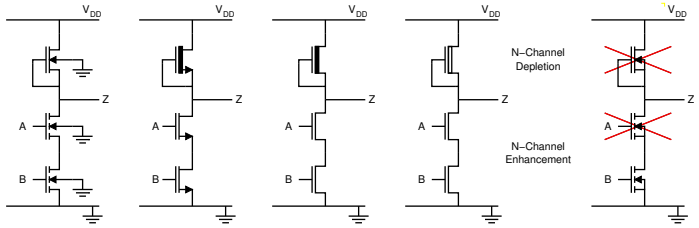



- An active PMOS device complements the NMOS device giving:
  - rail to rail output swing.
  - negligible static power consumption.



## Overview of Technologies

### Alternative transistors representations for NMOS circuits




Various shorthands are used for simplifying NMOS circuit diagrams.

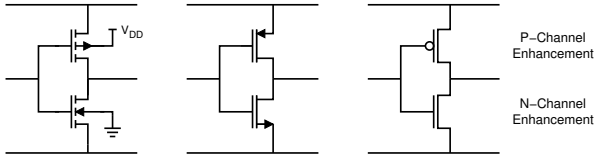
- Substrate connections need not be drawn since all must connect to Gnd.
- Since source and drain are indistinguishable in the layout, there is no need to show the source on the circuit diagram.

*Note that schematic tools not designed for IC design will usually include inappropriate 3-terminal symbols.*



## Digital CMOS Circuits

### Alternative transistor representations for CMOS circuits

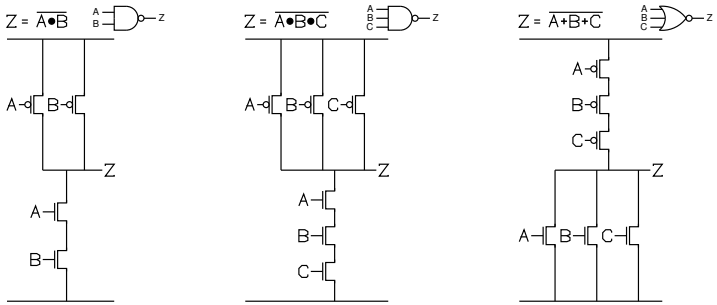



Digital CMOS circuits[4] tend to use simplified symbols like their NMOS counterparts.

- In general substrate connections are not drawn where they connect to Vdd (PMOS) and Gnd (NMOS).
- All CMOS devices are enhancement mode.
- Transistors act as simple digitally controlled switches.

[4] in analog CMOS circuits we may have wells not connected to Vdd/GND

Static CMOS complementary gates

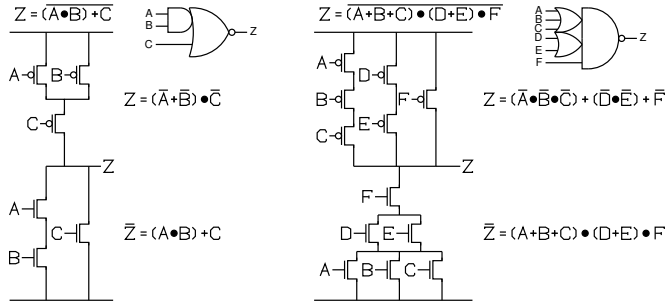



- For any set of inputs there will exist either a path to Vdd or a path to Gnd.





## Compound Gates




- All compound gates are inverting.
- Realisable functions are arbitrary AND/OR expressions with inverted output.

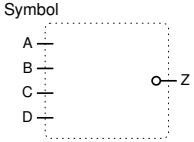




## Compound Gate Example

Z = $\overline{(A \bullet B) + (C \bullet D)}$

Symbol




Pull Up Network

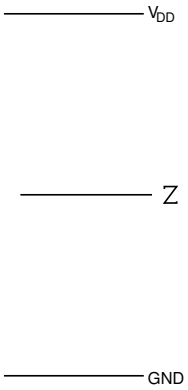
$Z = f(\bar{A}, \bar{B}, \bar{C}, \bar{D})$

Z = .....

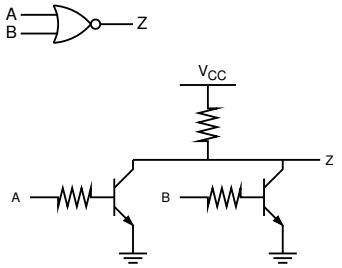
Pull Down Network

$\bar{Z} = f(A, B, C, D)$

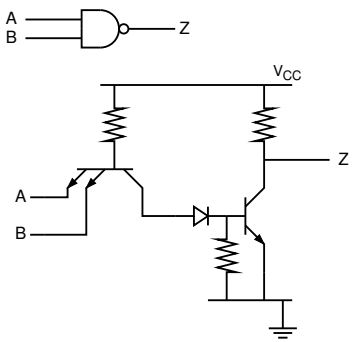
$\bar{Z} = (A \bullet B) + (C \bullet D)$



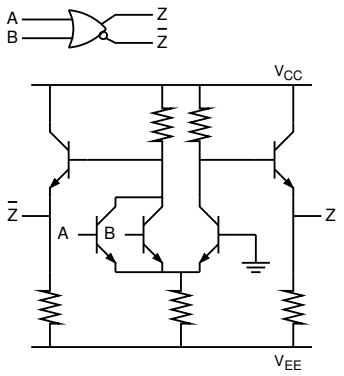
RTL NOR Gate



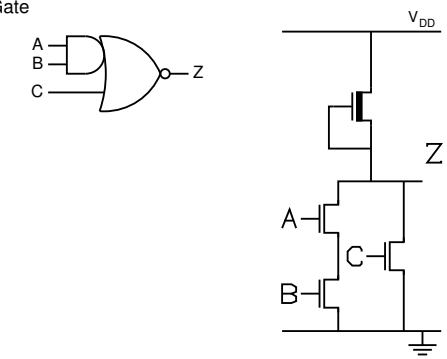

TTL NAND Gate



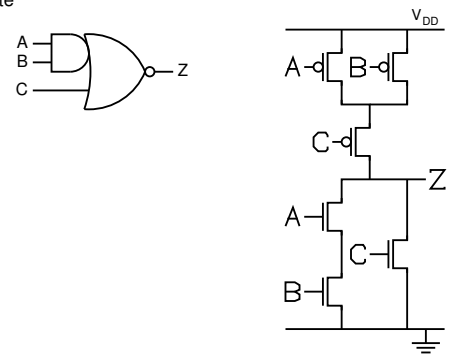

ECL OR/NOR Gate




NMOS Compund Gate




CMOS Compund Gate




- Bipolar Transitors with Resistors - MSI/LSI

  RTL - NOR  TTL - NAND  ECS - OR/NOR

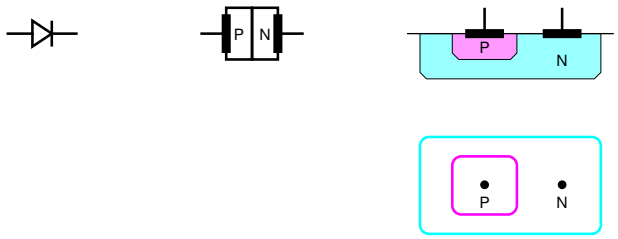
- MOS Transistors (no resistors) - VLSI

  NMOS  CMOS - No static power!

*Both allow construction of NOR, NAND & Compound gate (always inverting)*

## Diodes and Bipolar Transistors

Diode



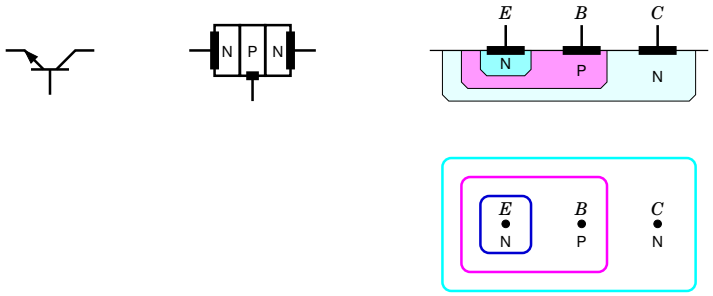

- Ideal structure - 1D
- Real structure - 3D
- Depth controlled implants.





## Diodes and Bipolar Transistors

NPN Transistor



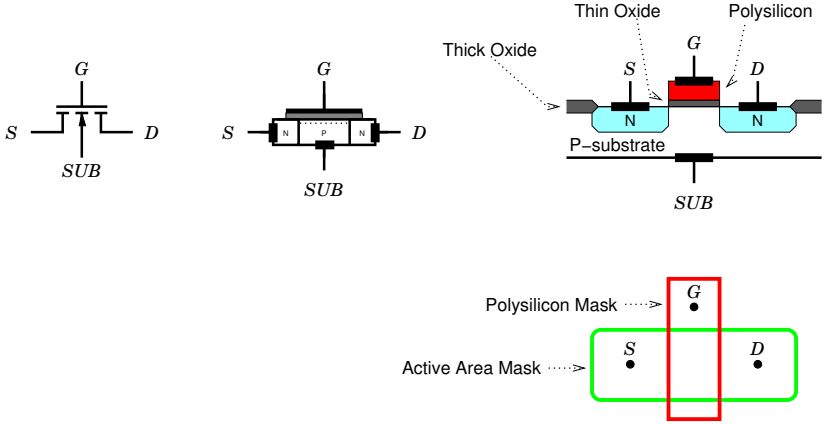

- Two n-type implants.





## MOS Transistors

Simple NMOS Transistor



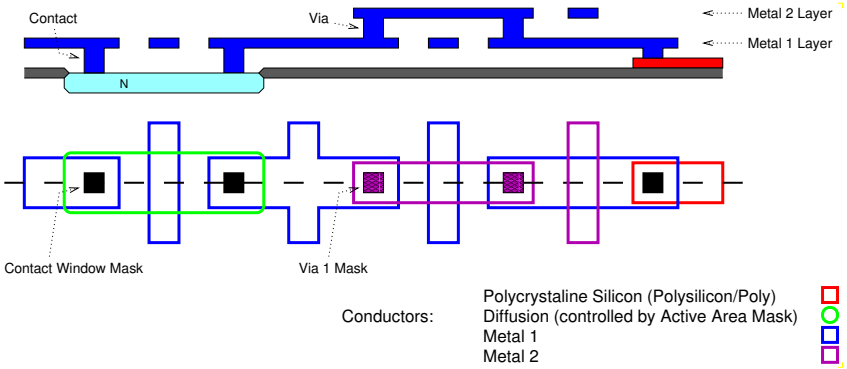





## Simple NMOS Transistor

- Active Area mask defines extent of *Thick Oxide*.
- Polysilicon mask also controls extent of *Thin Oxide* (alias *Gate Oxide*).
- N-type implant has no extra mask.
  – It is blocked by thick oxide and by polysilicon.
  – The implant is *Self Aligned*.
- Substrate connection is to bottom of wafer.
  – All substrates to ground.
- Gate connection not above transistor area.
  – Design Rule.

# Interconnect




- Crossing conductors on different masks do not interact[1].
  - Explicit contact/via is required for connection.
- Crossing conductors on the same mask are always connected.

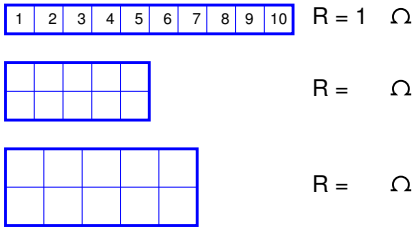
[1] the exception to this rule is that polysilicon crossing diffusion gives us a transistor



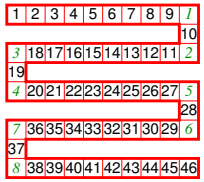
# Components for IC Design

## Resistors

Examples for Metal
assuming Rs = 0.1 ohms per square




Example for Polysilicon
assuming Rs = 200 ohms per square



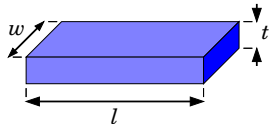

- for larger resistances we need minimum width poly (often combined with a *serpantine* shape) to save on area
- corner squares count as half[2] squares
- for predicatability and matching we may need wider tracks without corners

[2] effective resistance $\approx 0.56 R_s$



# Interconnect

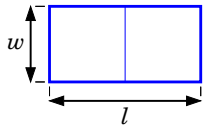
## Resistance




$$R = \left(\frac{\rho}{t}\right)\left(\frac{l}{w}\right)$$

where $\rho$ is the resistivity constant
$3.2 \times 10^{-8} \Omega m$ for aluminium
$1.7 \times 10^{-8} \Omega m$ for copper

Since $t$ and $\rho$ are fixed for a paricular mask layer, the value that is normally used is the sheet resistance: $R_s = \left(\frac{\rho}{t}\right)$.




$$R = R_s\left(\frac{l}{w}\right)$$

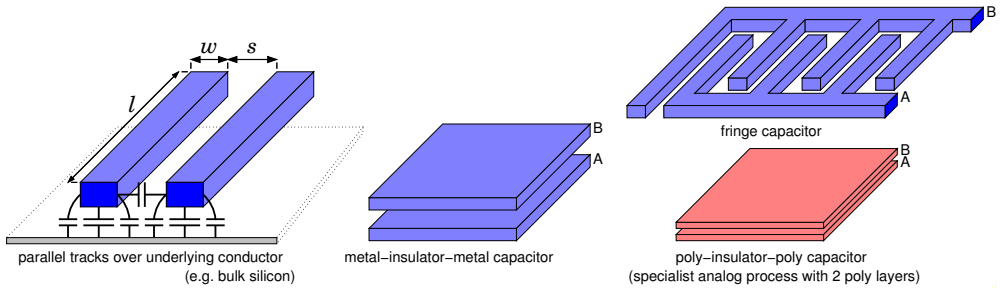
where $R_s$ is sheet resistance
$0.1\Omega/\square$ for $170nm$ thick copper

*$R_s$ = resistance of a square (i.e. $w = l$) so the units for $R_s$ are $\Omega/\square$ (ohms per square).*



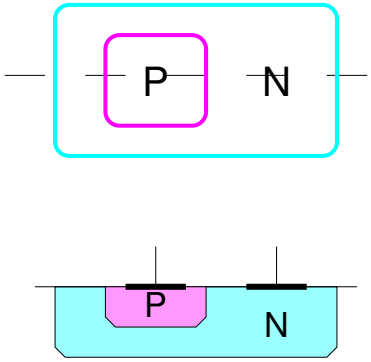
# Components for IC Design

## Capacitors

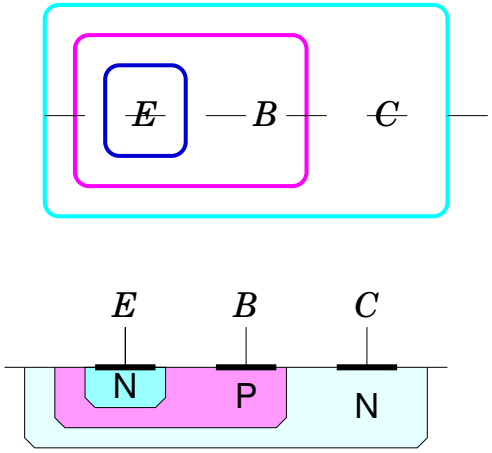



- Capacitance to underlying conductor $C = C_a\ w\ l + 2\ C_f\ l$
- Coupling capacitance to adjacent track $C = C_c\ l/s$
  where $C_a$, $C_f$, $C_c$ are constants for a given layer and process
  *in digital designs our only aim is to minimise **parasitic** capacitance*

## Diode

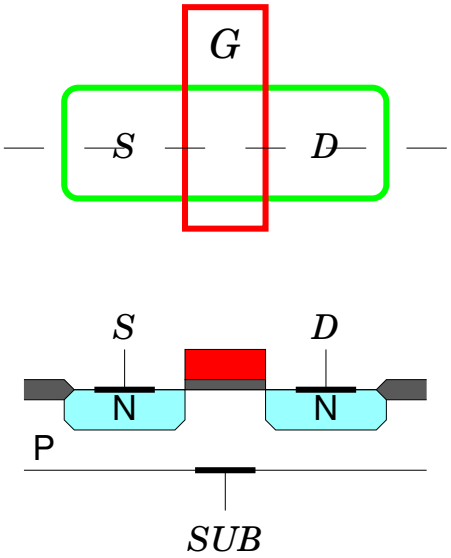



## NPN Transistor

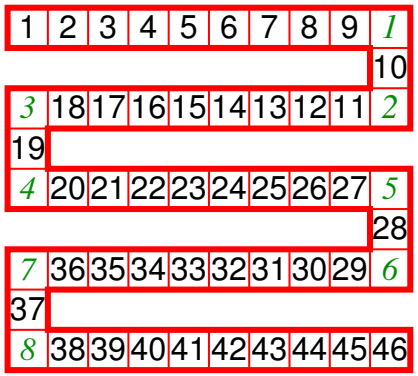



## NMOS Enhancement transistor

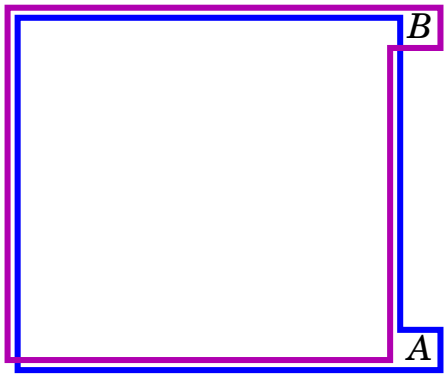
NMOS Process

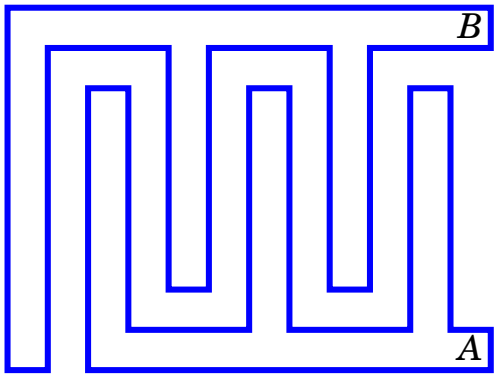



## Resistor

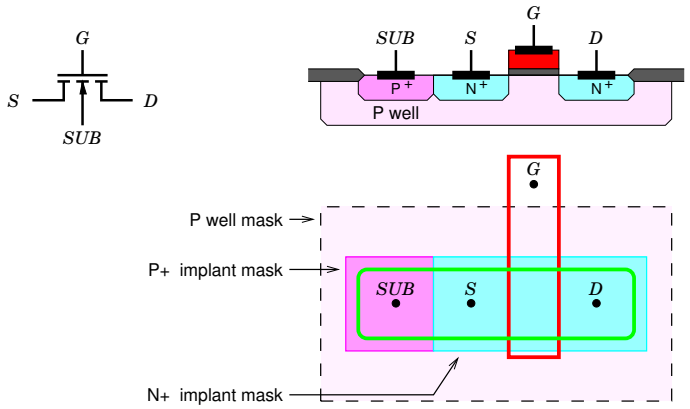



## Capacitors



## NMOS Transistor – with top substrate connection








## NMOS Transistor – with top substrate connection

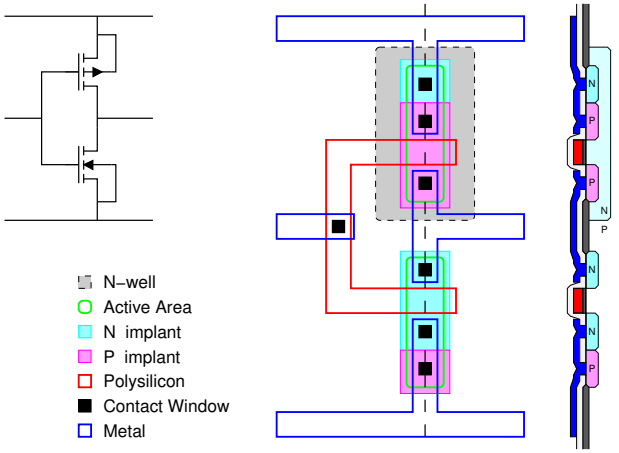
Where it is not suitable for substrate connections to be shared, a more complex process is used.

- Five masks must be used to define the transistor:
  - P Well
  - Active Area
  - Polysilicon
  - N+ implant
  - P+ implant
- P Well, for isolation.
- Top *substrate* connection.
- P+/N+ implants produce good *ohmic* contacts.





## CMOS Inverter



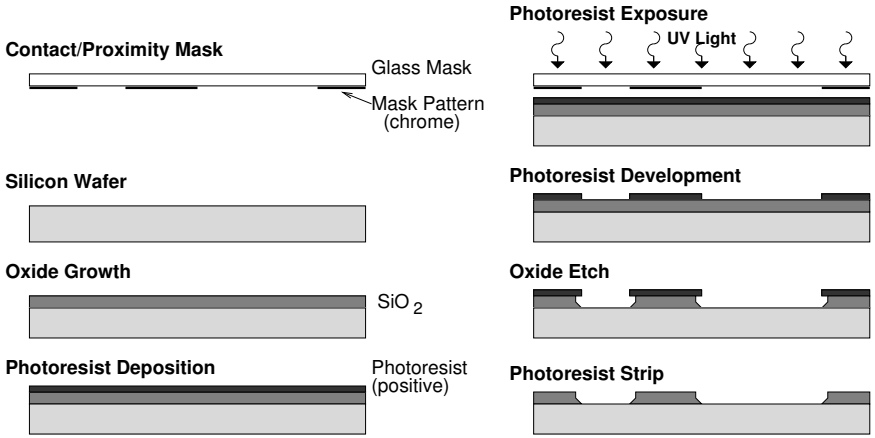





## CMOS Inverter

- The process described here is an *N Well process* since it has only an N Well.
  P Well and Twin Tub processes also exist.
- Note that the P-N junction between chip substrate and N Well will remain reverse biased.
  Thus the transistors remain isolated.
- N implant defines NMOS source/drain and PMOS substrate contact.
- P implant defines PMOS source/drain and NMOS substrate contact.

# Processing – Photolithography

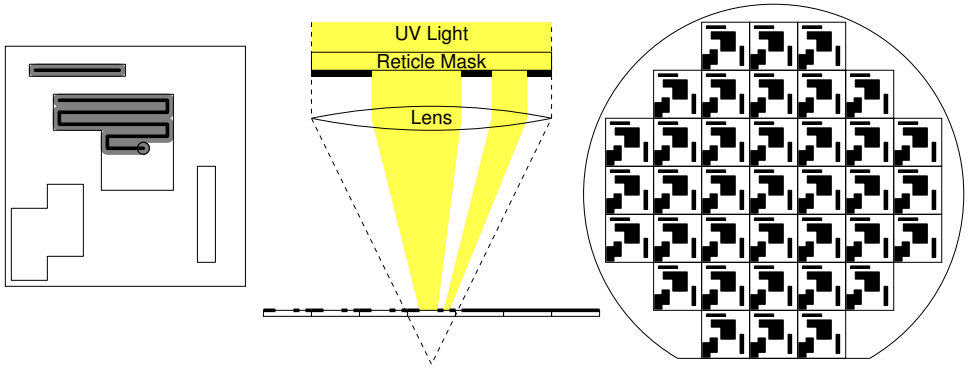



4005

# Processing – Mask Making

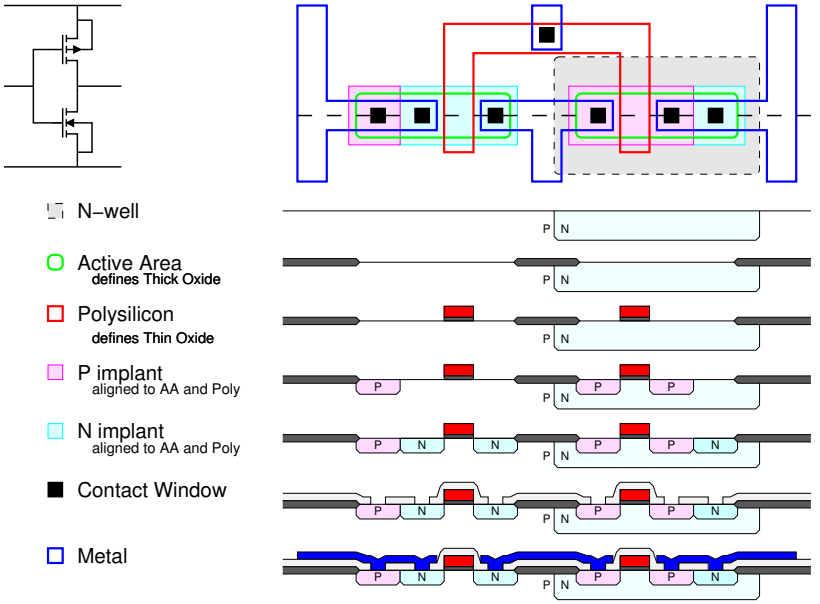
**Reticle written by scanning electron beam**

**Pattern reproduced on wafer (or contact/proximity mask) by step and repeat with optical reduction**




- Optical reduction allows narrower line widths.

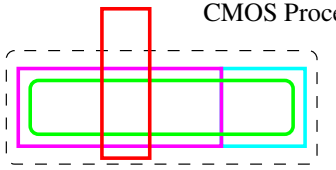
4006



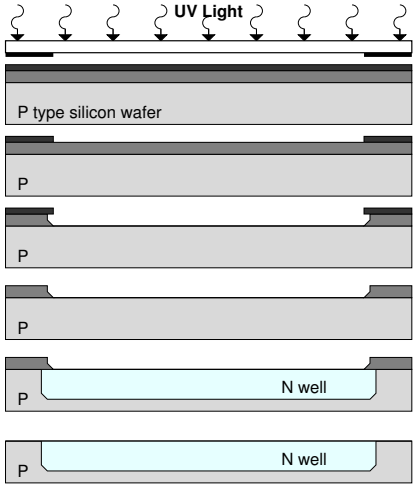

4007

PMOS Enhancement transistor

CMOS Process

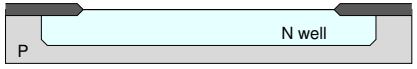


N Well

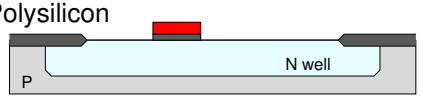



many steps for a single mask

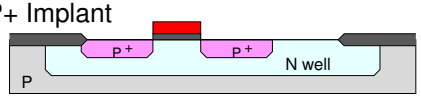
Active Area



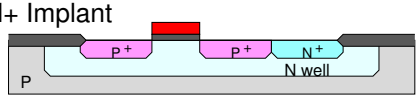

Polysilicon




P+ Implant

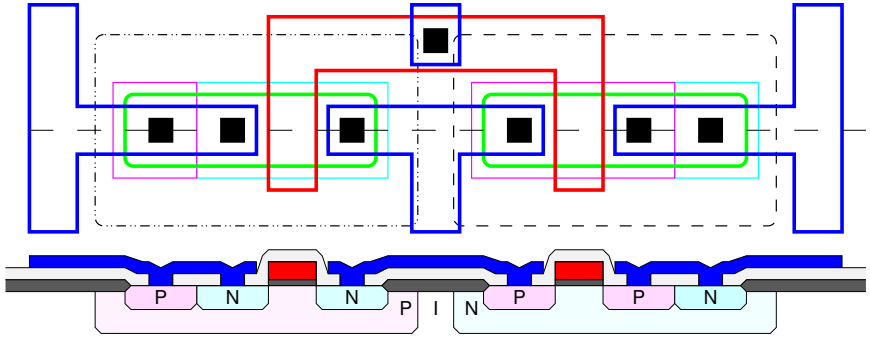



N+ Implant



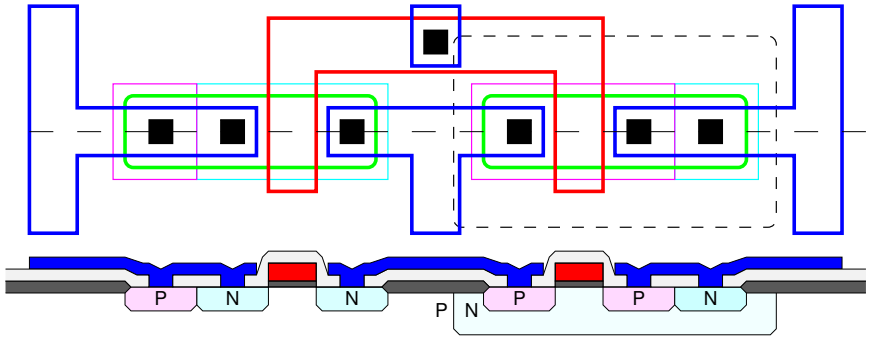

CMOS Inverter

Twin Tub CMOS Process



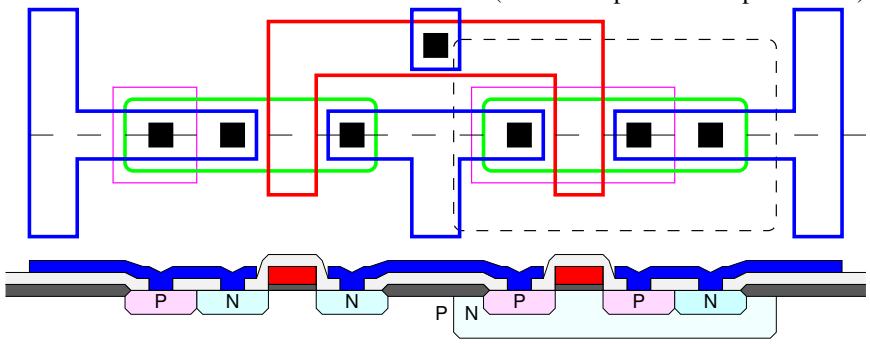

CMOS Inverter

N–Well CMOS Processs (with explicit N+ implant mask)




CMOS Inverter

N–Well CMOS Processs (without explicit N+ implant mask)



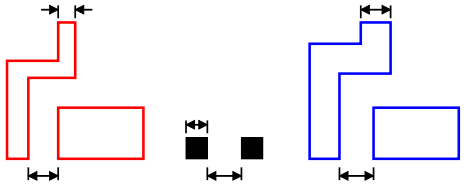

Features may be determined by a number of masks

e.g. NMOS source drain: ActiveArea AND NOT(NWell OR Poly OR PImplant)

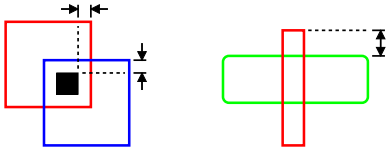
# Design Rules

To prevent chip failure, designs must conform to design rules:

- Single layer rules

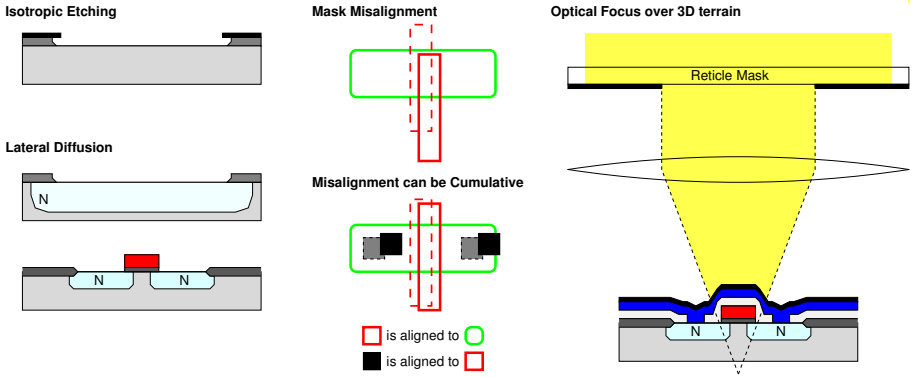


- Multi-layer rules





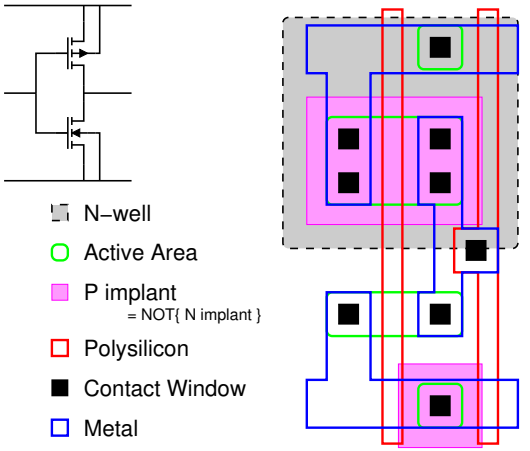
# Derivation of Design Rules






# Design Rules

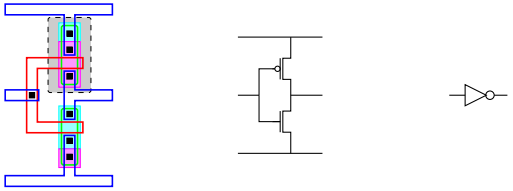
## 0.5 $\mu m$ CMOS inverter





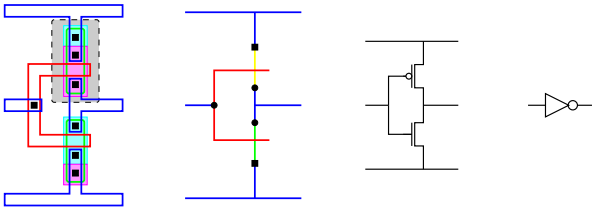

# Abstraction

## Levels of Abstraction



- Mask Level Design
  - Laborious Technology/Process dependent.
  - Design rules may change during a design!
- Transistor Level Design
  - Process independent, Technology dependent.
- Gate Level Design
  - Process/Technology independent.

## Abstraction - Stick Diagrams



Stick diagrams give us many of the benefits of abstraction:

- Much easier/faster than full mask specification.
- Process independent (valid for any CMOS process).
- Easy to change.

while avoiding some of the problems:

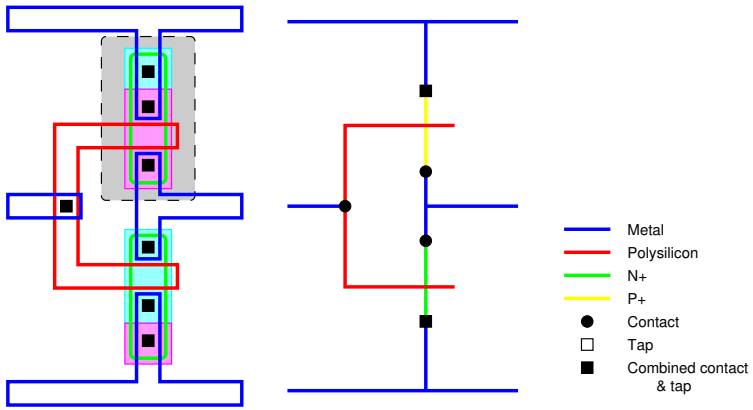
- Optimized layout may be generated much more easily from a stick diagram than from transistor or gate level designs.[1]

[1] note that all IC designs must end at the mask level.



## Digital CMOS Design

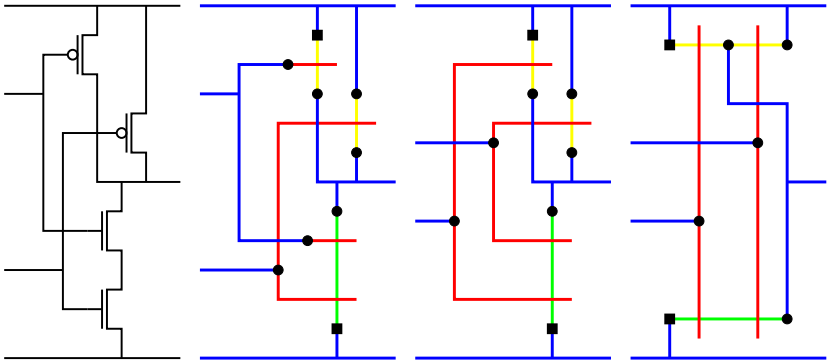
### Stick Diagrams






## Digital CMOS Design

### Stick Diagrams



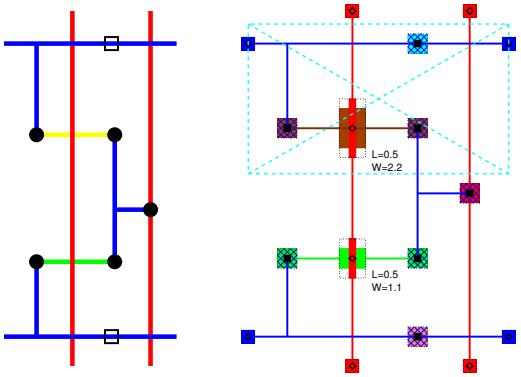


## Digital CMOS Design

### Stick Diagrams

- *Explore your Design Space.*
  - Implications of crossovers.
  - Number of contacts.
  - Arrangement of devices and connections.
- Process independent layout.
- Easy to expand to a full layout for a particular process.

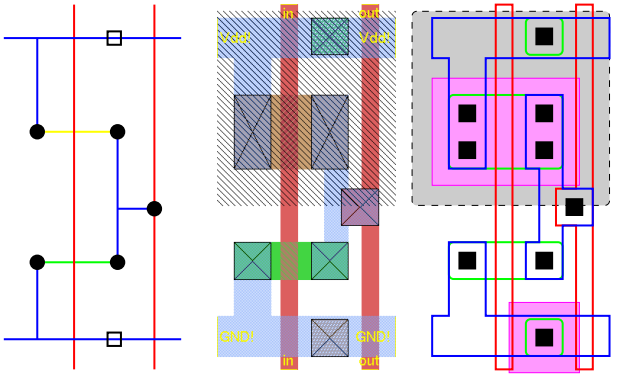
# Sticks and CAD - Symbolic Capture




- Transistors are placed and explicitly sized.
  - components are joined with zero width wires.
  - contacts are automatically selected as required.
- A semi-automatic compaction process will create DRC correct layout.

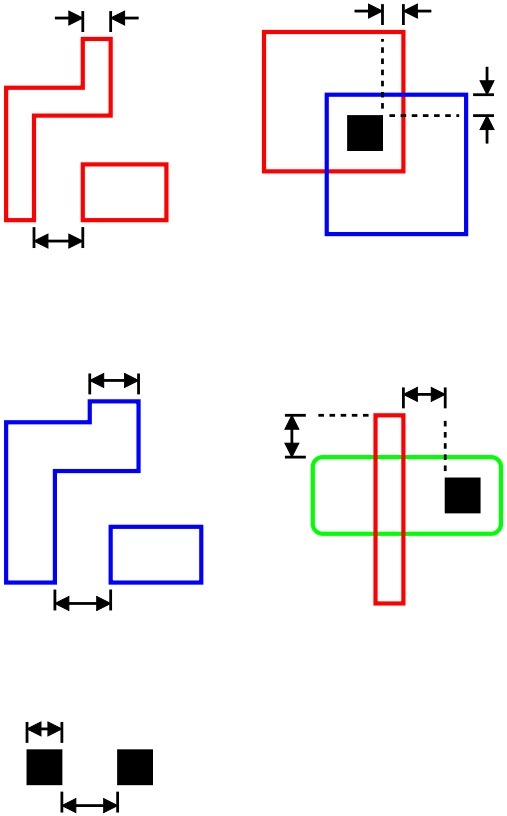


# Sticks and CAD - Magic

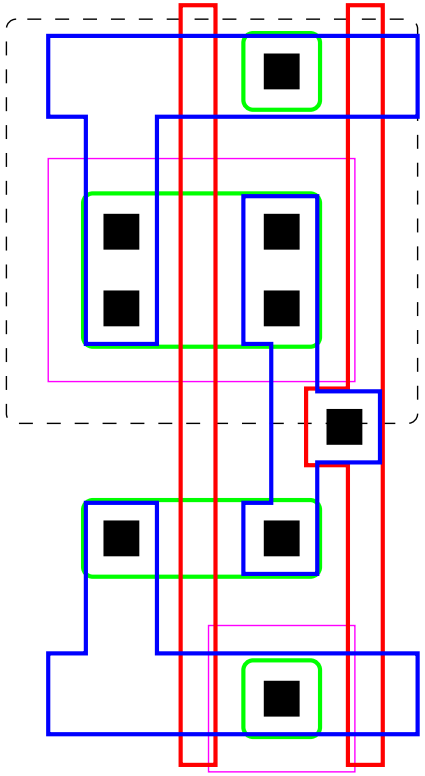



- Log style design (sticks with width) - DRC errors are flagged immediately.
  - again contacts are automatically selected as required.
- On-line DRC leads to rapid generation of correct designs.
  - symbolic capture style compaction is available if desired.

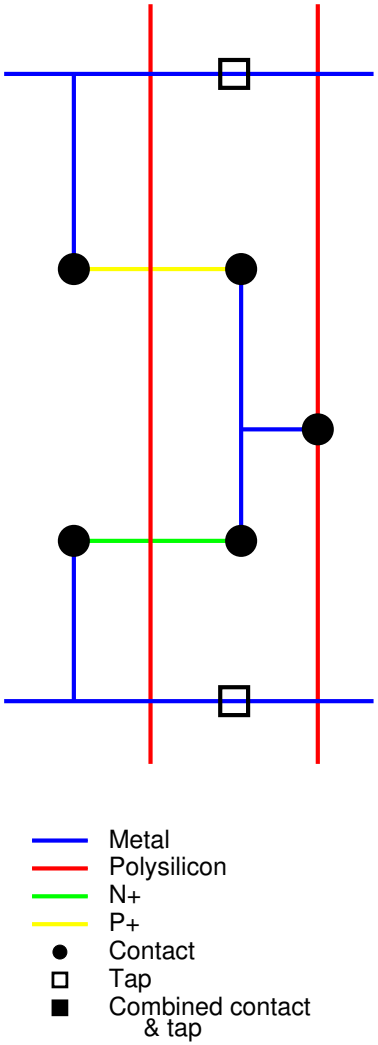
Design Rules – width, separation, overlap



Optimised Mask Layout

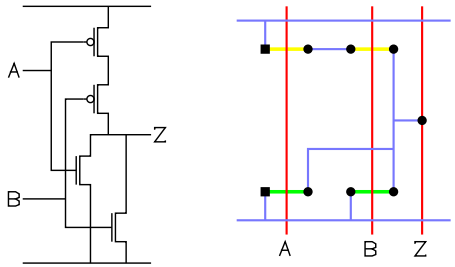


Equivalent Stick Diagram



## A logical approach to gate layout.

- *All complementary gates may be designed using a single row of n-transistors above or below a single row of p-transistors, aligned at common gate connections.*

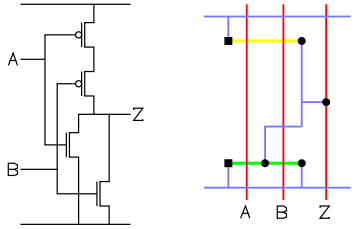







## Euler Path

- For the majority of these gates we can find an arrangement of transistors such that we can butt adjoining transistors.
  - Careful selection of transistor ordering.
  - Careful orientation of transistor source and drain.
- Referred to as *line of diffusion*.








## Finding an Euler Path

Computer Algorithms

- It is relatively easy for a computer to consider all possible arrangements of transistors in search of a suitable Euler path.
  This is not so easy for the human designer.

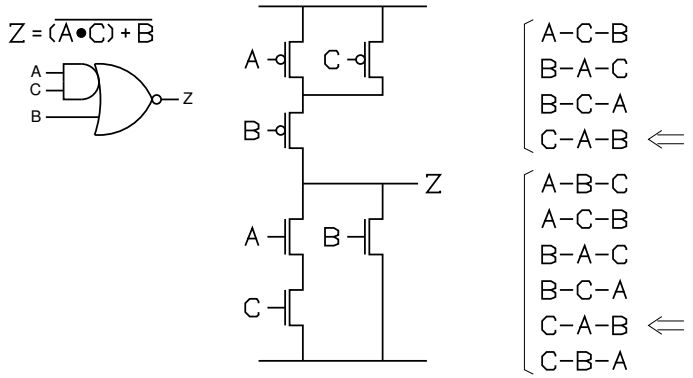
One Human Algorithm

- Find a path which passes through all n-transistors exactly once.
- Express the path in terms of the gate connections.
- Is it possible to follow a similarly labelled path through the p-transistors?
  - Yes – you've succeeded.
  - No – try again (you may like to try a p path first this time).



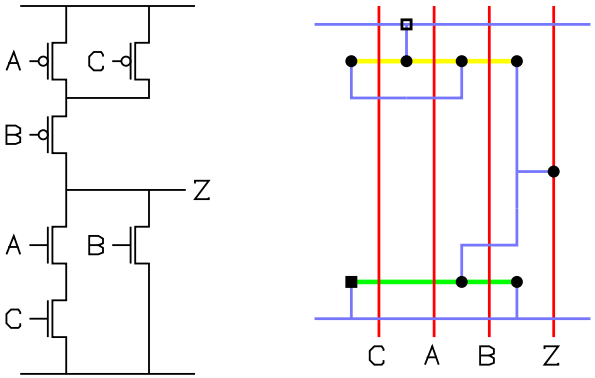


## Finding an Euler Path




Here there are four possible Euler paths.

## Finding an Euler Path

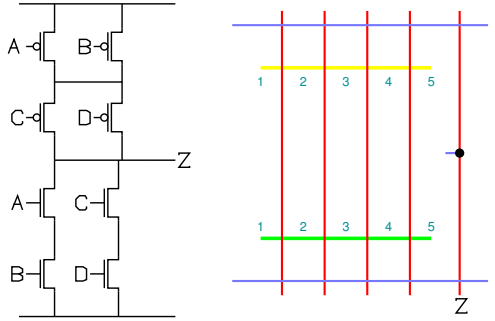







## Euler Path Example

$Z = \overline{(A \bullet B) + (C \bullet D)}$




1. Find Euler path
2. Label poly columns
3. Route power nodes
4. Route output node
5. Route remaining nodes
6. Add taps[1] for PMOS and NMOS

*A combined contact and tap, ■, may be used only where a power contact exists at the end of a line of diffusion. Where this is not the case a simple tap, ⊟, should be used.*

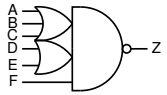
[1] 1 tap is good for about 6 transistors – insufficient taps may leave a chip vulnerable to latch-up

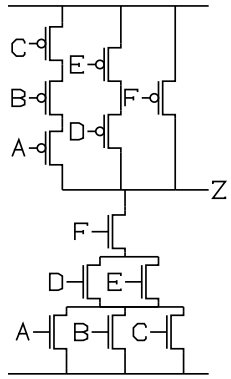




## Finding an Euler Path

$Z = \overline{(A+B+C) \bullet (D+E) \bullet F}$



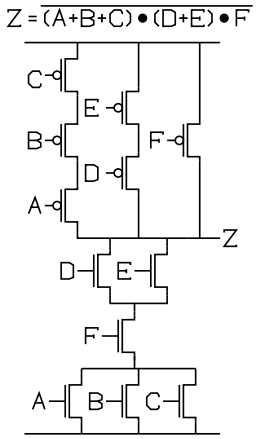

No possible path through n-transistors!

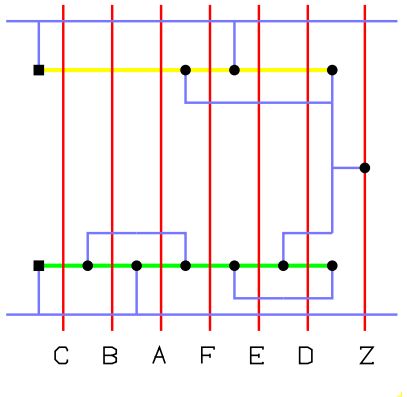




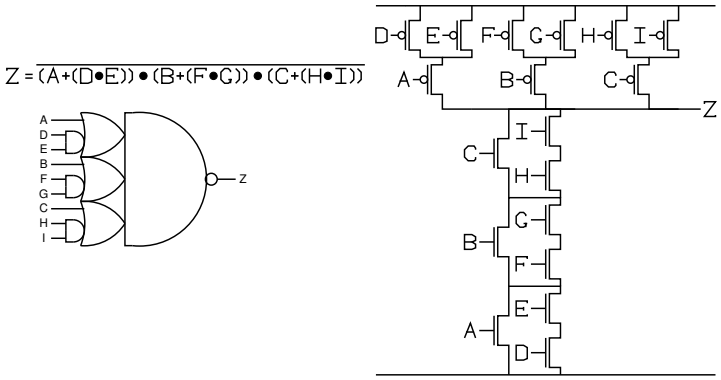
## Finding an Euler Path

$Z = \overline{(A+B+C) \bullet (D+E) \bullet F}$



## Finding an Euler Path

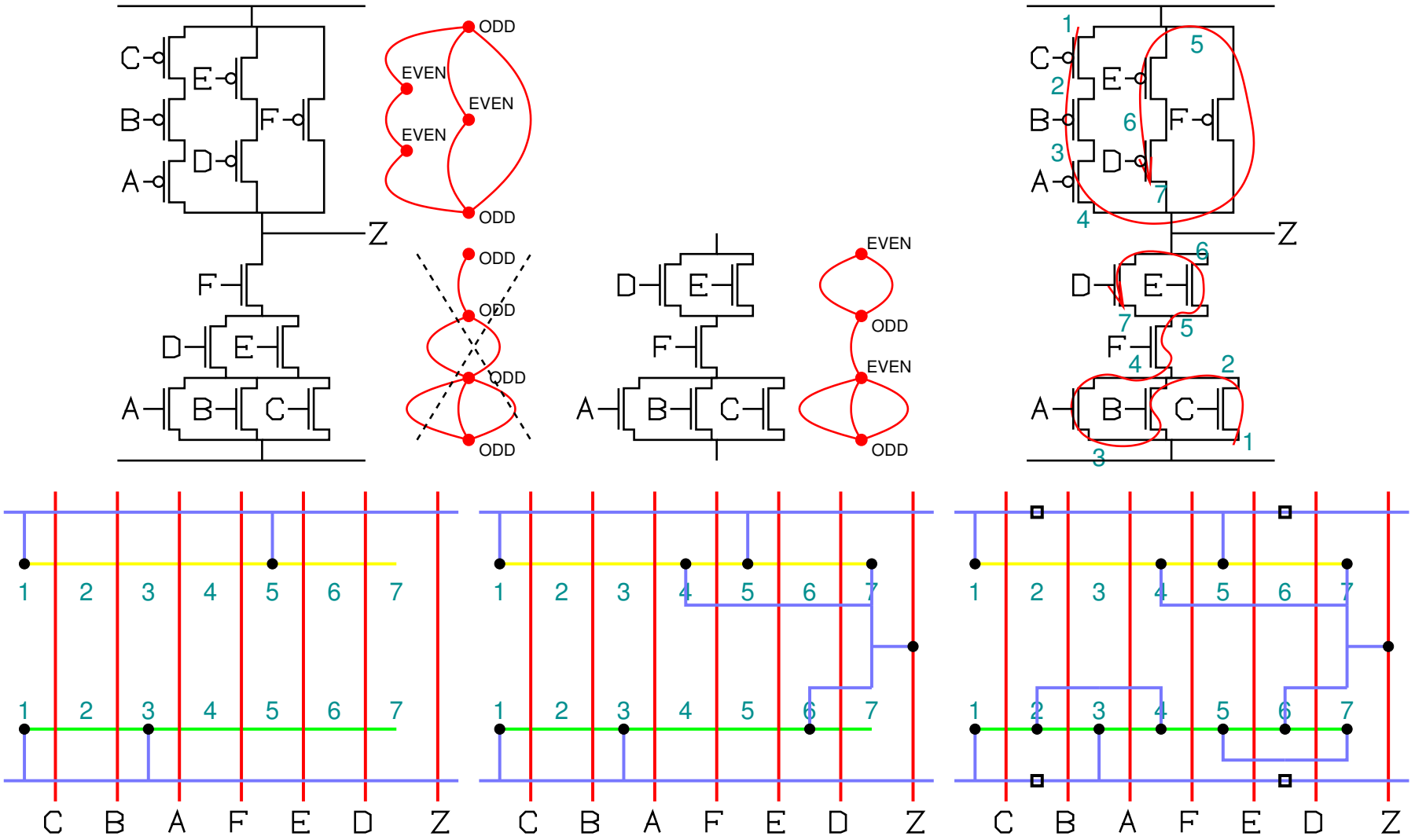






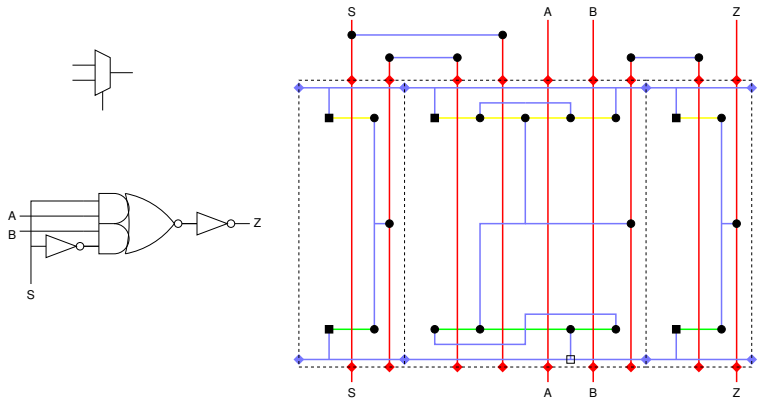
No possible path through p-transistors.
No re-arrangement will create a solution!

# Investigation of Euler paths leads to more efficient layout*




*not all gates will support a common Euler path for both PMOS and NMOS

## Multiple gates







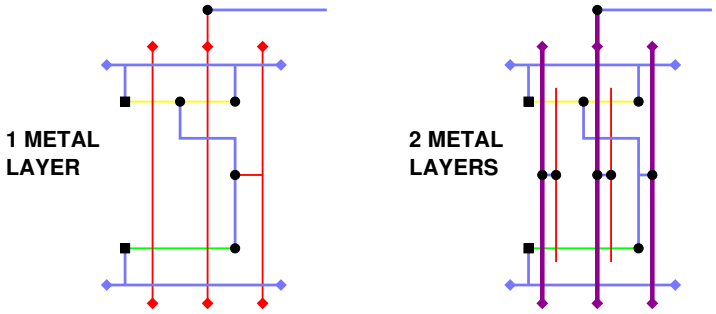

## Multiple gates

- Gates should all be of same height.
  - Power and ground rails will then line up when butted.
- All gate inputs and outputs are available at top and bottom.
  - All routing is external to cells.
  - Preserves the benefits of hierarchy.
- Interconnect is via *two conductor routing*.
  - In this case Polysilicon vertically and Metal horizontally.





## Two-layer Metal




Most modern VLSI processes support two or more metal layers.
The norm is to use only metal for inter-cell routing.

usually Metal1 / Metal2 for horizontal / vertical inter-cell routing (and for power rails) (and for cell inputs and outputs).





Many ICs are designed using the standard cell method.

- Cell Library Creation

  A cell library, containing commonly used logic gates, is created for a process. This is often carried out by or on behalf of the foundry.

- ASIC[1] Design

  The ASIC designer must design a circuit using the logic gates available in the library.

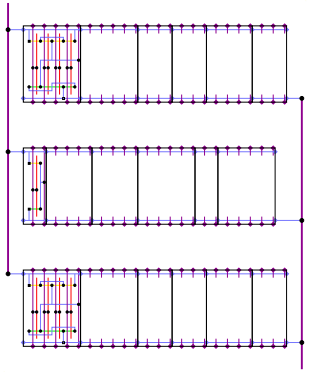
  The ASIC designer usually has no access to the full layout of the standard cells and doesn't create any new cells for the library.

  Layout work performed by the ASIC designer is divided into two stages:

  - Placement
  - Routing

[1] Application Specific Integrated Circuit

## Placement

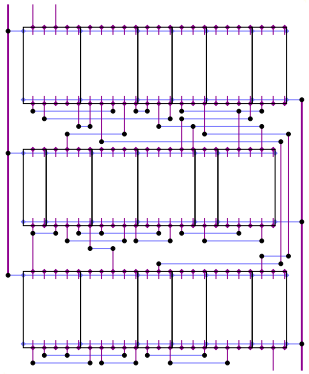


Cells are placed in one or several equal length lines with inter-digitated power and ground rails.





## Routing



In the routing channels between the cells we route metal1 horizontally and metal2 vertically.





## Two conductor routing

- Conductor A / Conductor B for horizontal / vertical inter-cell routing [2]
- This logical approach means that we should never have to worry about signals crossing.

  This makes life considerably easier for a computer (or even a human) to complete the routing.
- We must only ensure that two signals will not meet in the same horizontal or vertical channel.
- Computer algorithms can be used to ensure placement of cells such that wires are short.[3]
- Further computer algorithms can be used to optimize the routing itself.

[2] In the two-metal example Conductor A is Metal1 and Conductor B is Metal2
[3] In VLSI circuits we often find that inter-cell wiring occupies more area than the cells themselves.





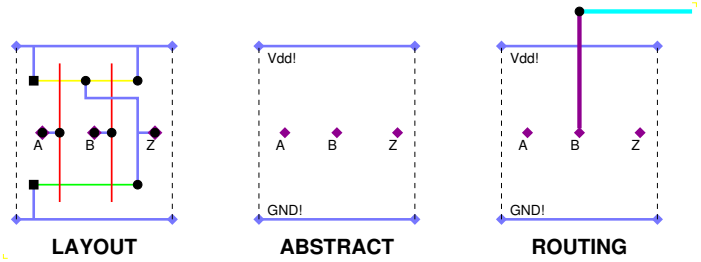
## More Metal Layers

With three or more metal layers it is possible to take a different approach.
The simplest example uses three metal layers.

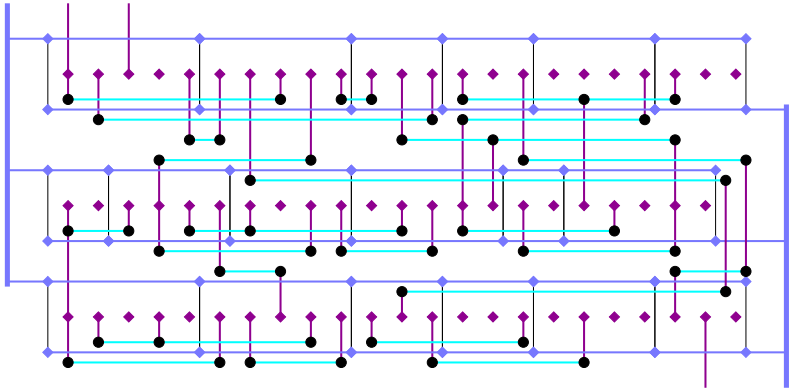
- Standard Cells

  Use only metal1 except for I/O which is in metal2
- Two Conductor Routing

  Uses metal2 and metal3



## More Metal Layers

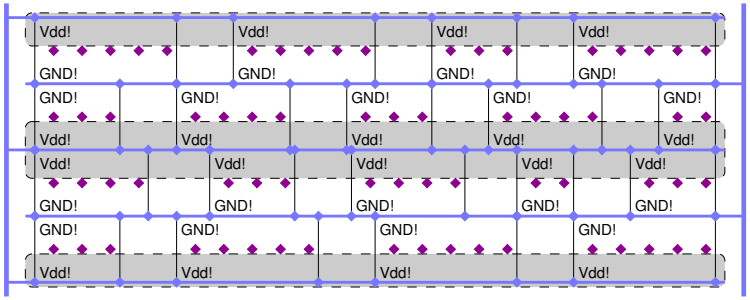


With this approach we can route safely over the cell to the specified pins leading to much smaller gaps between cell rows.





## Alternative Placement Style




By flipping every second row it may be possible to eliminate gaps between rows.
N-wells are merged and power or ground rails are shared.
This approach is normally associated with sparse rows and non channel based routing algorithms.

LAYOUT

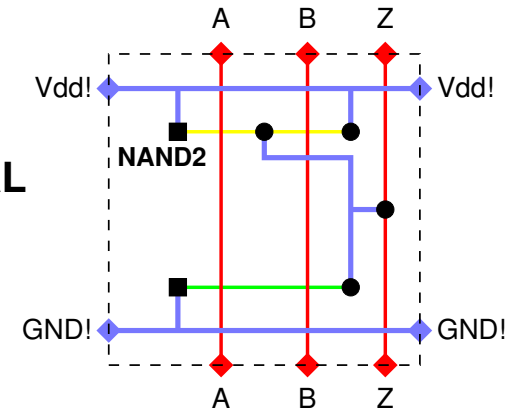
ABSTRACT

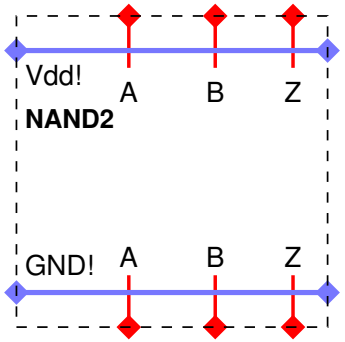
ROUTING

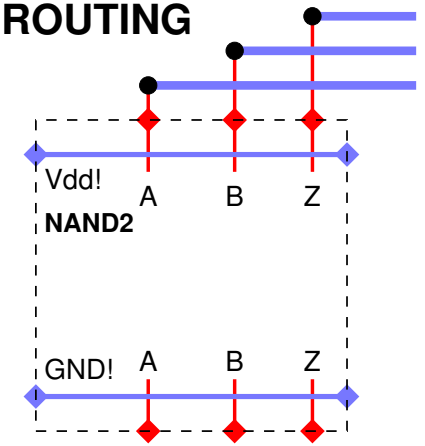
1 METAL LAYER

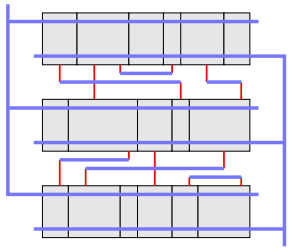
2 METAL LAYERS

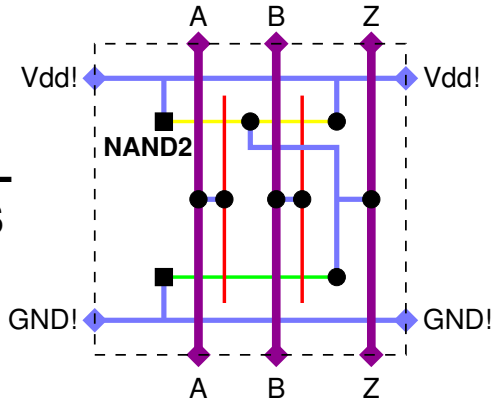
3 METAL LAYERS

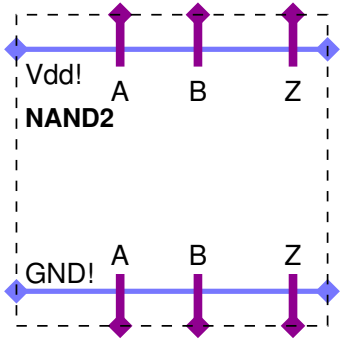


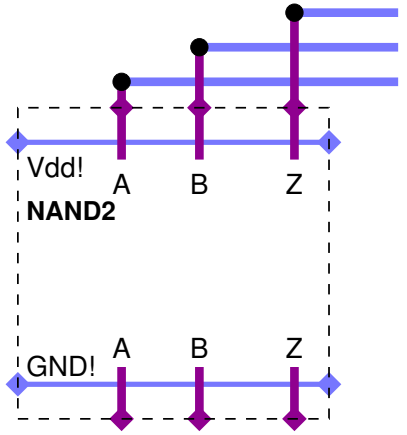



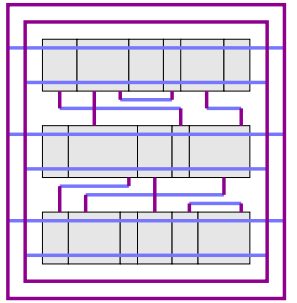



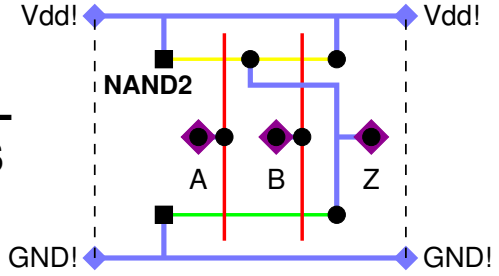



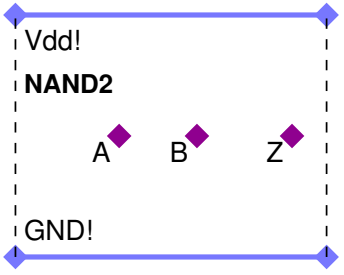


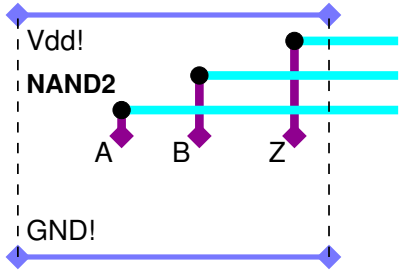



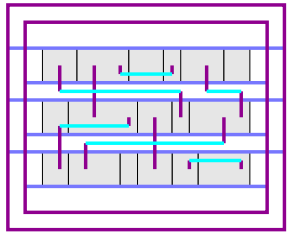



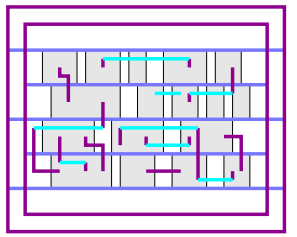







# System Design Choices

- Programmable Logic
  - PLD
    e.g. Lattice ispGAL22V10, Atmel ATF1502 CPLD
  - Field Programmable Gate Array (FPGA)
    e.g. Altera Cyclone III, Xilinx Artix-7/Zync-7000
- Semi-Custom Design
  - Mask Programmable Gate Array
    e.g. ECS CMOS Gate Array
    Altera HardCopy II structured ASICs
  - Standard Cell Design
    e.g. Alcatel Mietec MTC45000 0.35$\mu m$ cell library
- Full Custom Design



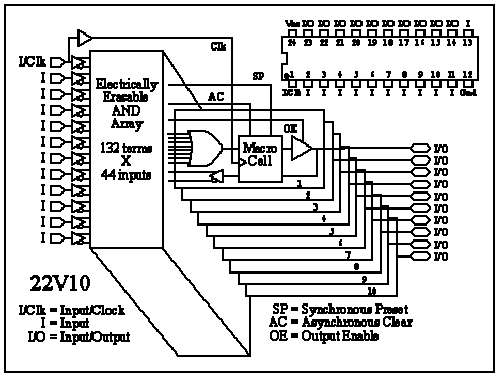
# System Design Choices

- Programmable Logic START HERE
  - Best possible design turnaround time
  - Cheapest for prototyping
  - Best time to market
  - Minimum skill required
- Semi-Custom Design
- Full Custom Design
  - Cheapest for mass production
  - Fastest
  - Lowest Power
  - Highest Density[1]
  - Most skill required

[1]optimization limited by speed/power/area trade off



# Programmable Logic

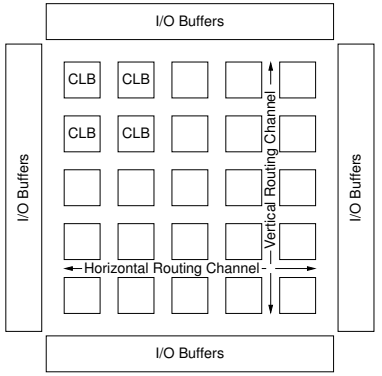


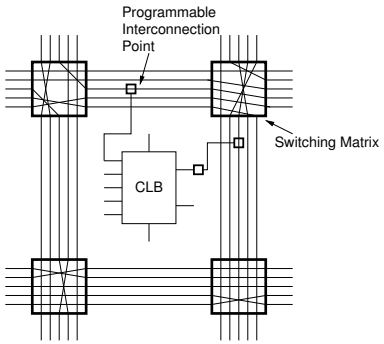

ICT PEEL22CV10 Source: ICT

- One time use - Fuse programmable.
- Reprogrammable - UV/Electrically Erasable.



# Field Programmable Gate Array – Xilinx XC4000

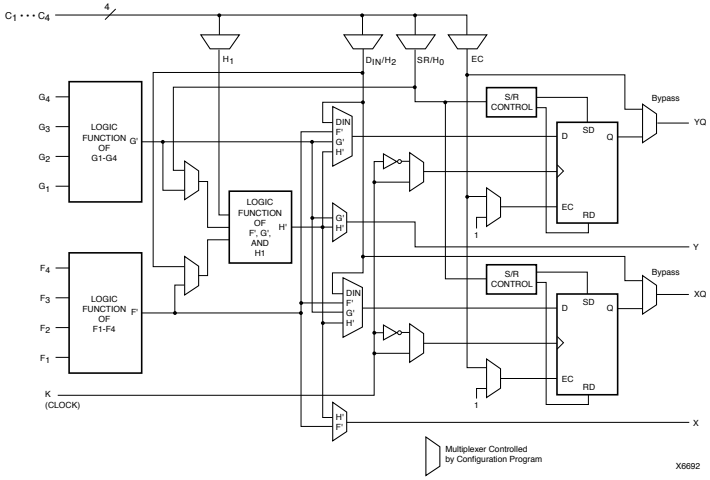



- Configurable Logic Blocks (CLBs) & I/O Blocks[2]
- Programmable Interconnect

[2]Xilinx XC4013 has 576 (24 × 24) CLBs and up to 192 (4 × 48) user I/O pins.

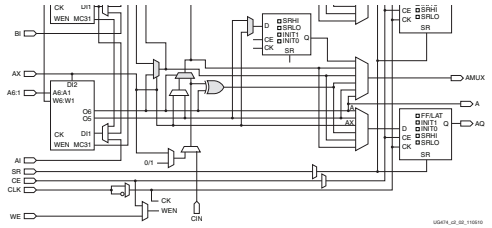
# Field Programmable Gate Array – Xilinx XC4000 CLB




Source: Xilinx



# Xilinx Artix-7 – SLICEM CLB[3]




Source: Xilinx

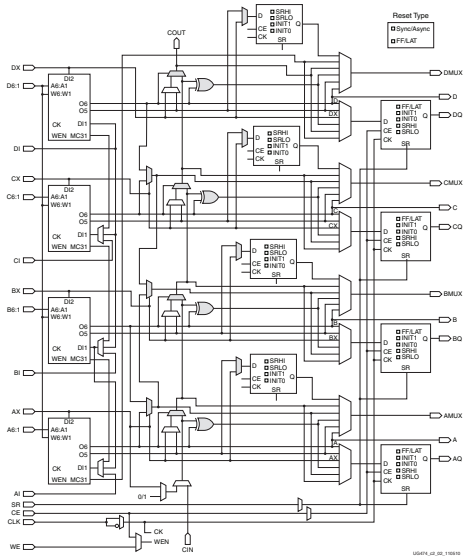
- 4x 6-input Look-Up Tables (LUTs) for combinational logic
- Carry chain supporting fast carry lookahead
- 8x storage elements

LUTs can be alternatively configured as

- 256 bits RAM
- 32-bit shift register

[3]Xilinx XC7A200T has 16,825 CLBs (each containing 2 slices) and up to 500 user I/O pins.






Artix-7 – SLICEM CLB

Source: Xilinx



# FPGA - System On Chip

Modern FPGAs are big enough for:

- One or more soft-core processors
- Program memory
- Data memory

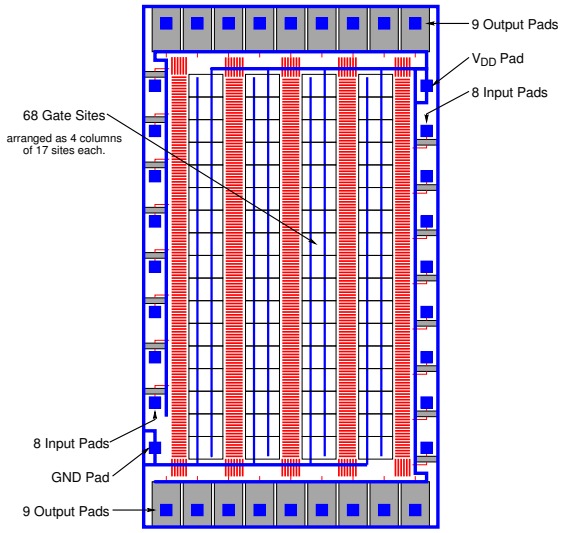
  \+ specialist hardware

The new trend is for FPGAs with hard processors built in:

- Xilinx Zync-7000 includes dual-core ARM A9
- Altera Arria V includes dual-core ARM A9
- Cypress PSoC 4 includes ARM Cortex-M0 and programmable digital[4] and analog blocks

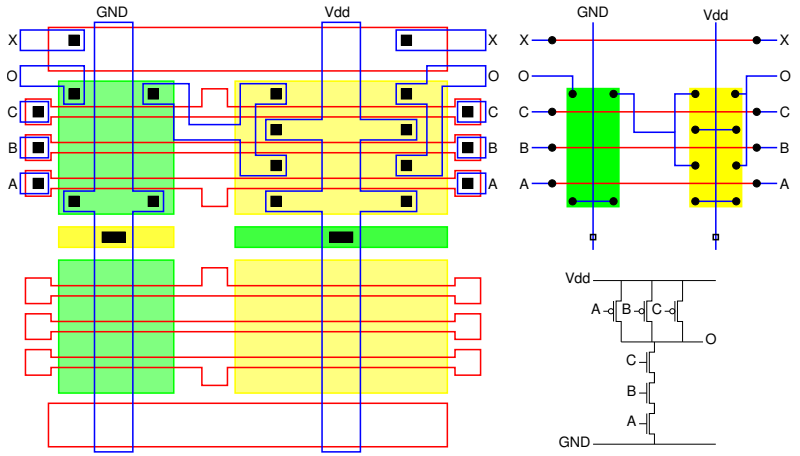
[4]here the digital block is PLD rather than FPGA

# Mask Programmable Gate Array






# Mask Programmable Gate Array

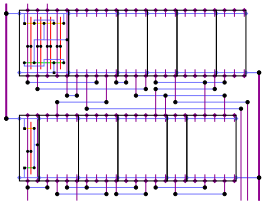



- Customize Metal and Contact Window masks only.



# Standard Cell Design

- Logic Functions



- Auto Generated Macro Blocks
  - PLA
  - ROM
  - RAM
- System Level Blocks
  - Microprocessor core[5]

[5]Will support System On Chip applications.



# Full Custom

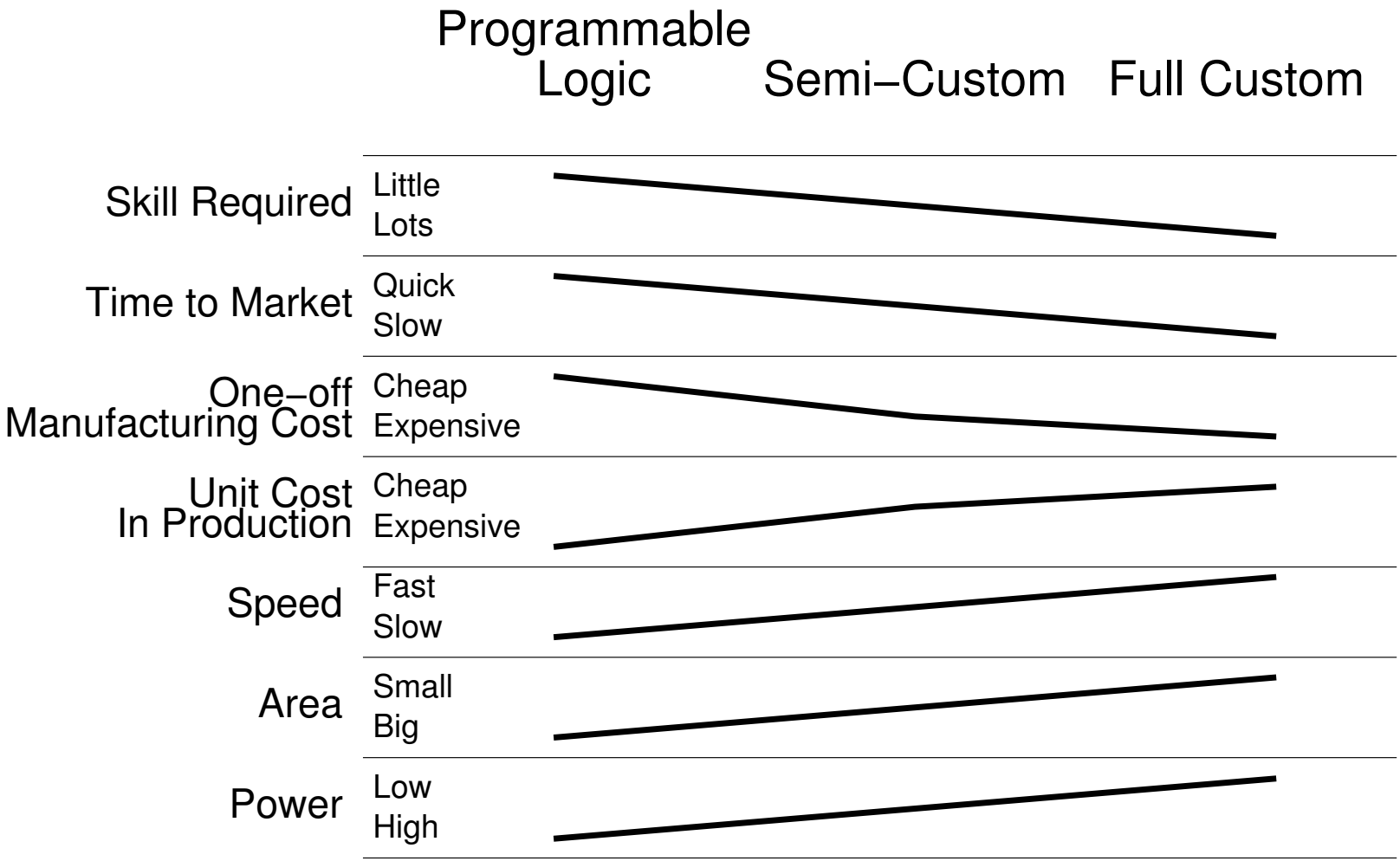
All design styles need full custom designers

- to design the base programmable logic chips
- to design building blocks for semi-custom

Where large ASICs use full custom techniques they are likely to be used alongside semi-custom techniques.

e.g. Hand-held computer game chip

- Full custom bitslice datapath
  hand crafted for optimum area efficiency and low power consumption
- Standard cell controller
- Macro block RAM, ROM




All design styles need full custom designers

A large ASIC (especially SoC) may mix Semi–Custom and Full Custom