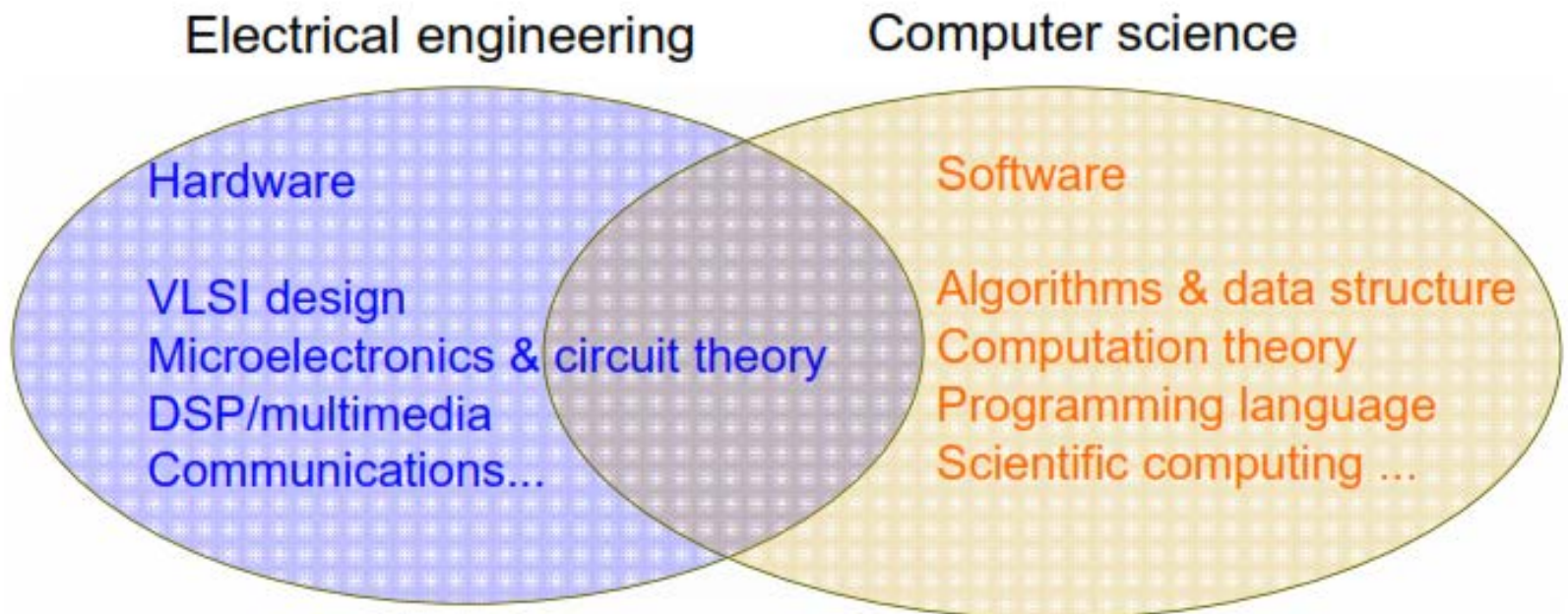


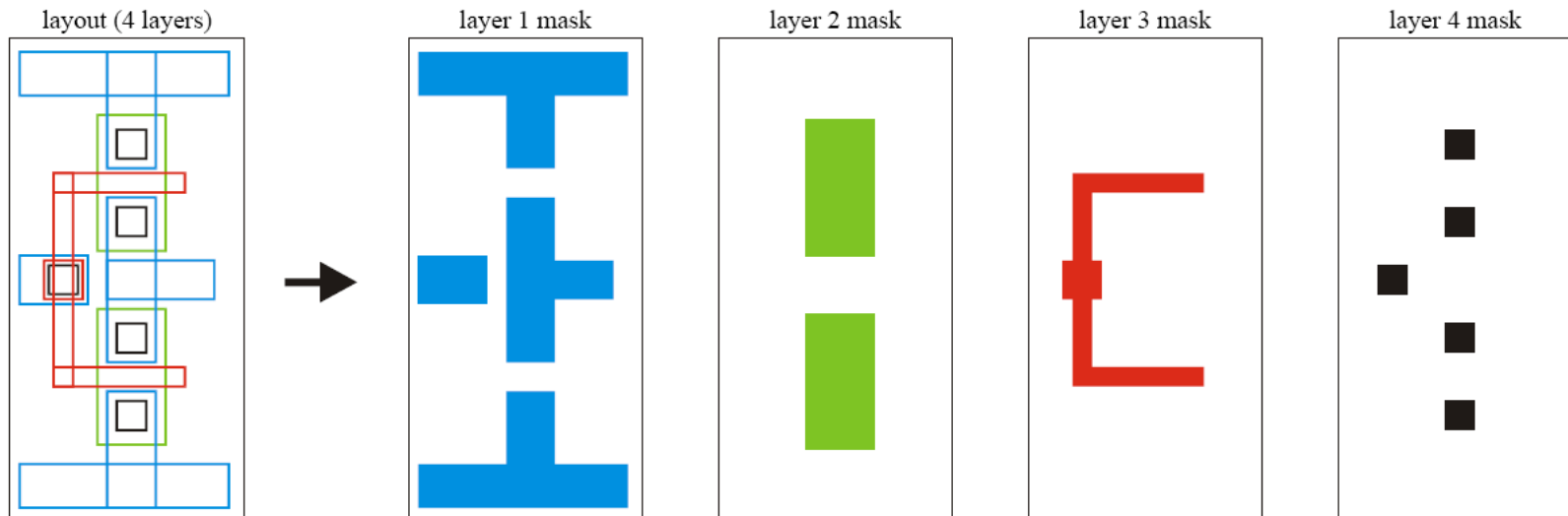
# La progettazione elettronica e la nascita di una nuova industria: The EDA industry (Electronic Design Automation)

EDA : where hardware and software meet each other

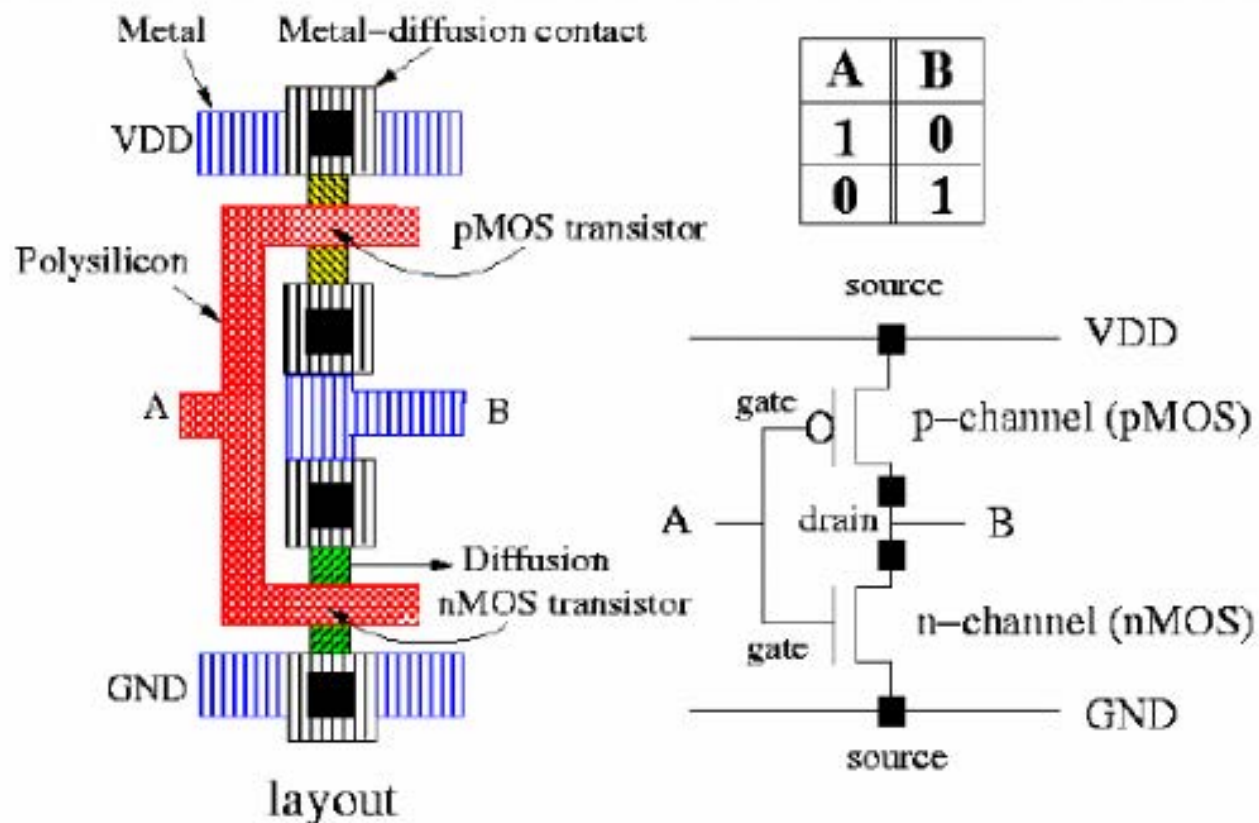


## Example Simple Mask Set

- Shown below is a highly simplified layout for a two transistor digital gate, and the masks which would be required based on its layout (see MOSFET)
- Not in notes, just shown as an example of how masks are derived from a user-generated layout

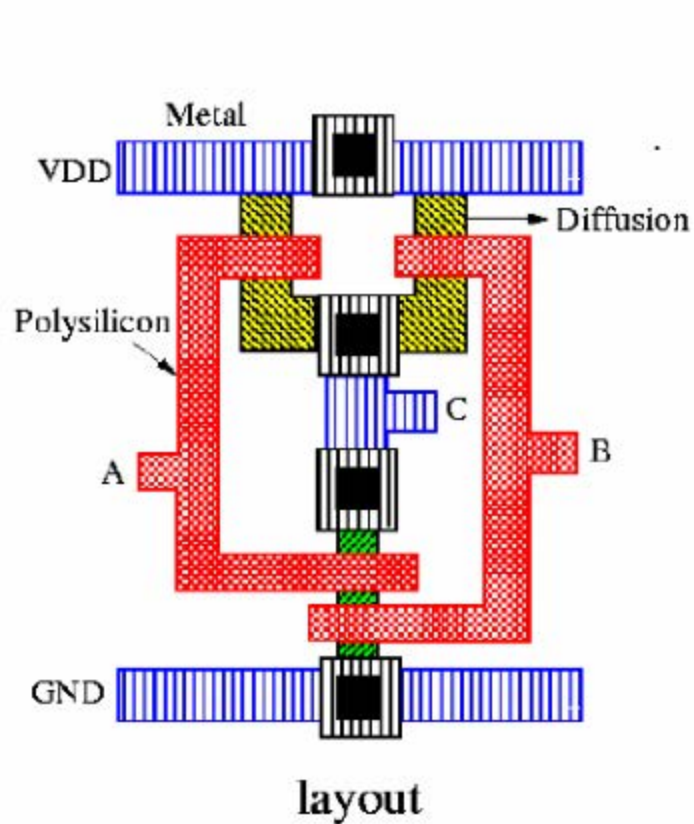


# inverter circuit

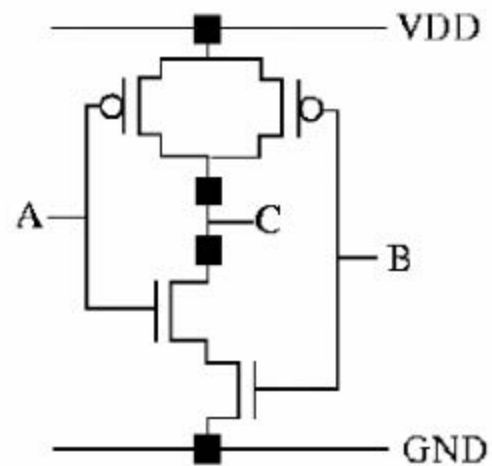


- metal 1: blue
- polysilicon: red
- p-diffusion: yellow (p-well: light yellow)
- metal 2: brown
- contact/via: black
- n-diffusion: green (n-well: light green)

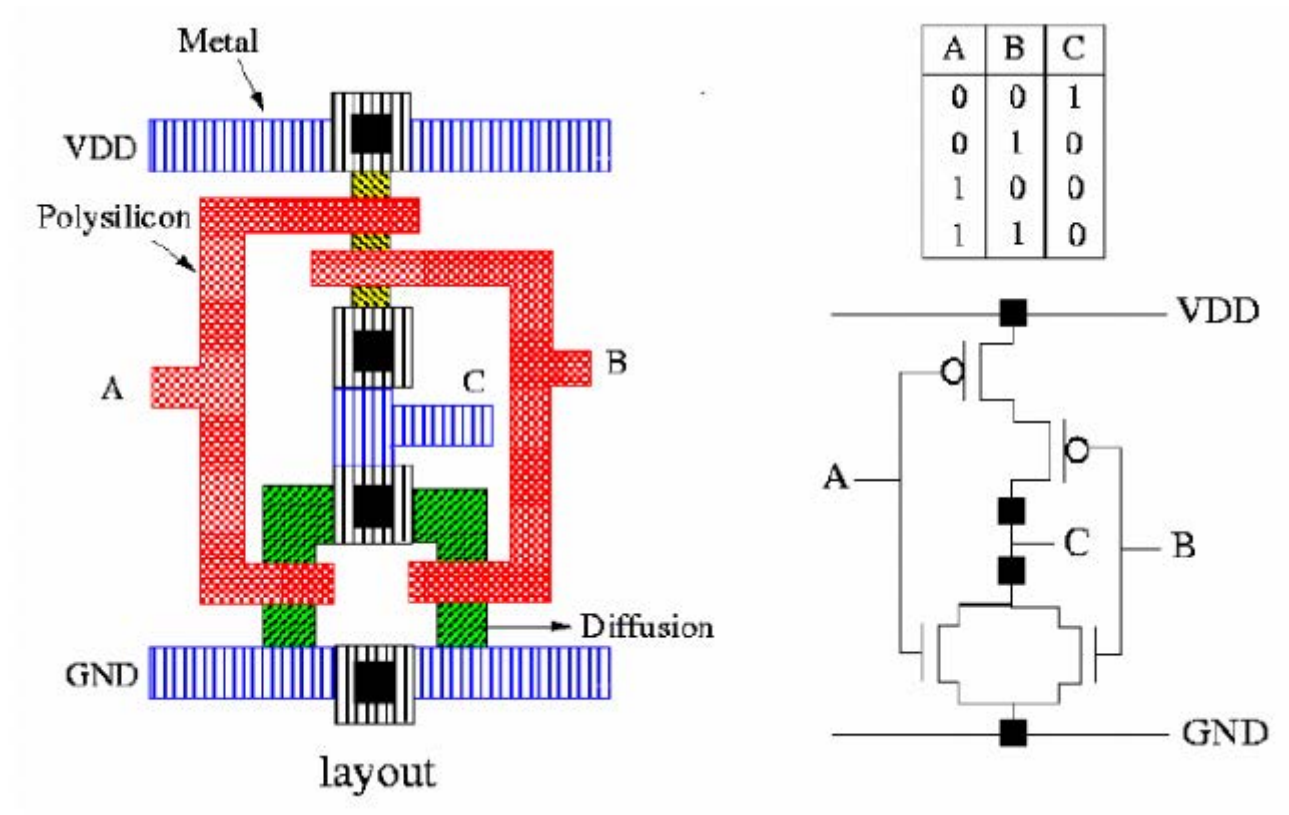
# NAND CMOS



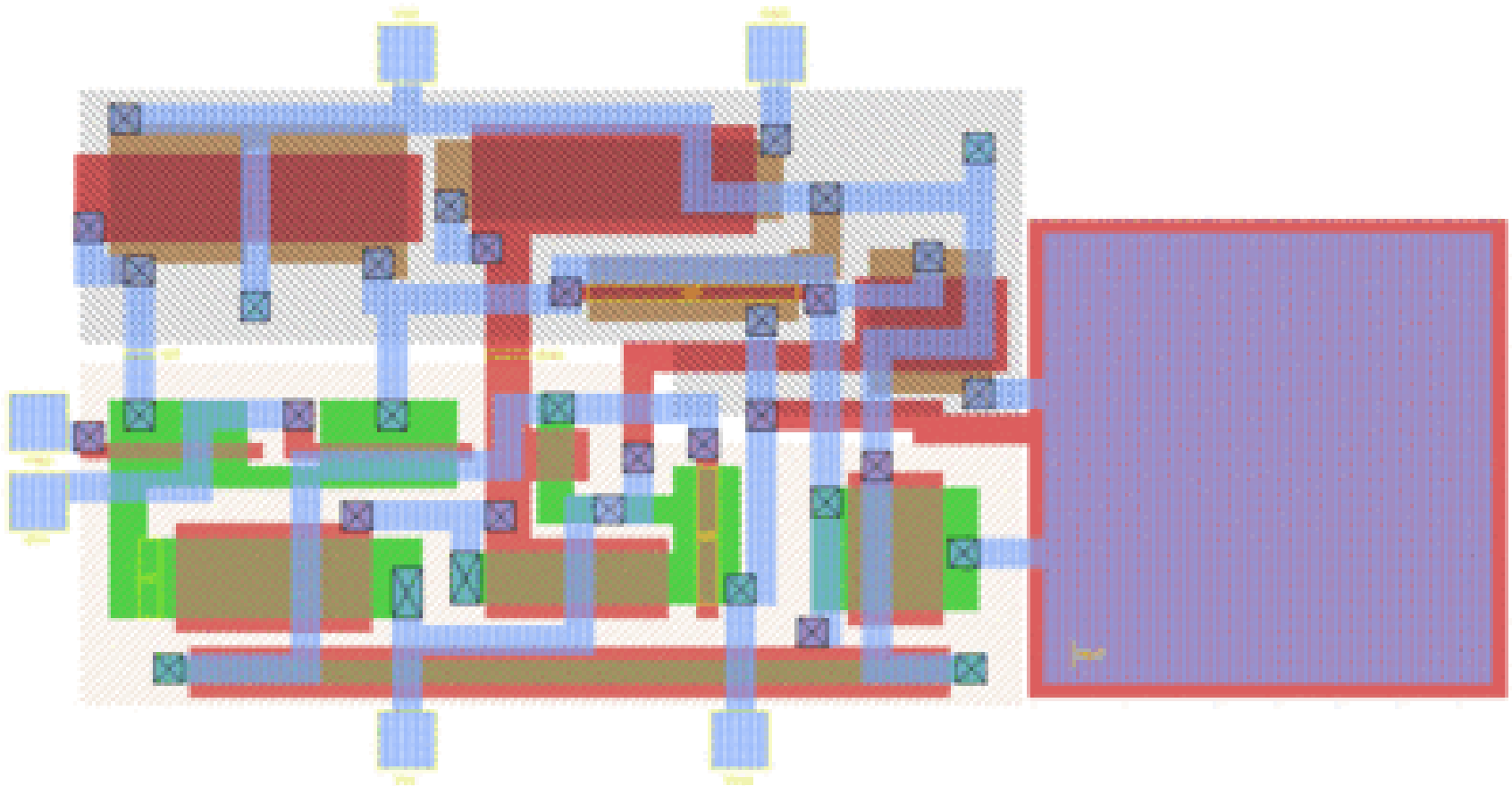
in1	in2	out
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0



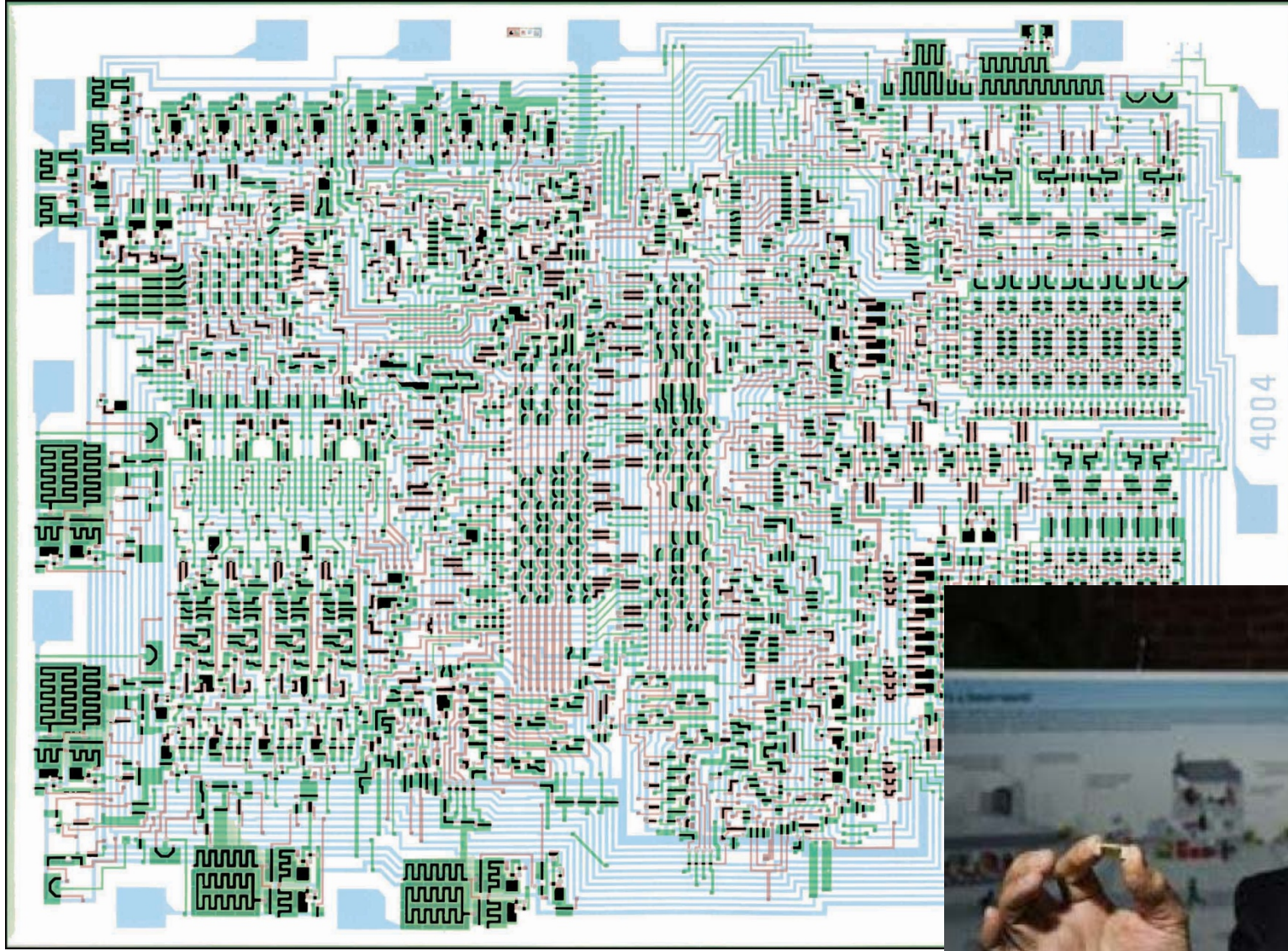
# NOR CMOS



# «Layout» di un circuito integrato









Carver Mead e Lynn Conway

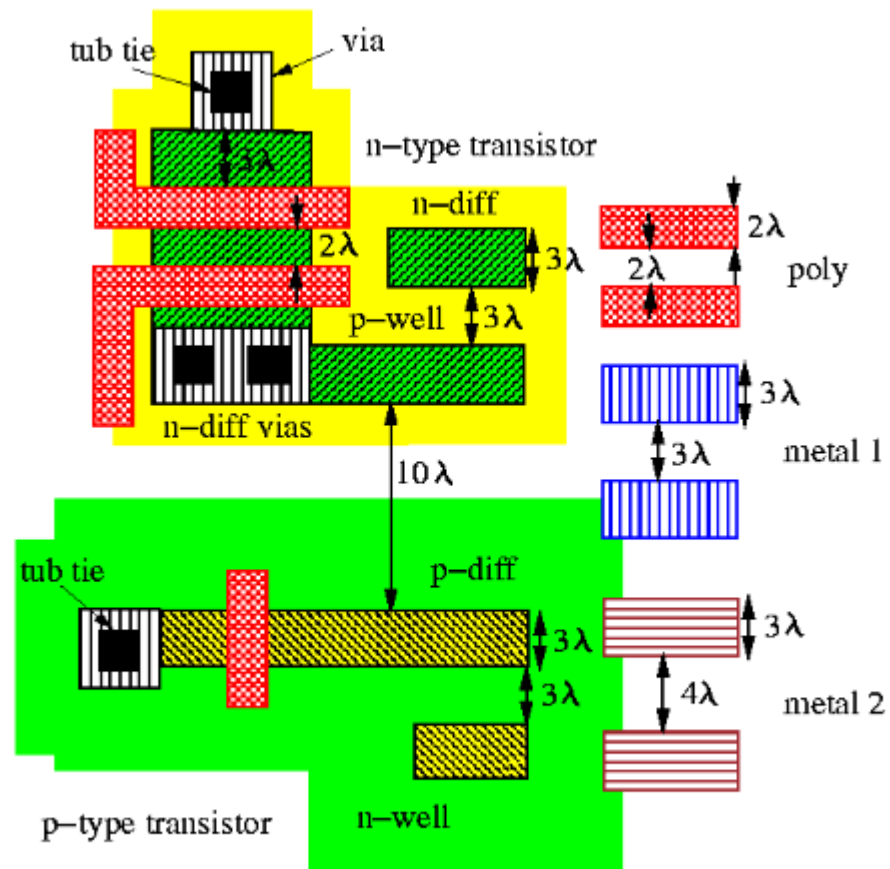
Gli inventori del moderno  
CAD microelettronico (1981)

«regole di layout»  
«multiproject chip»

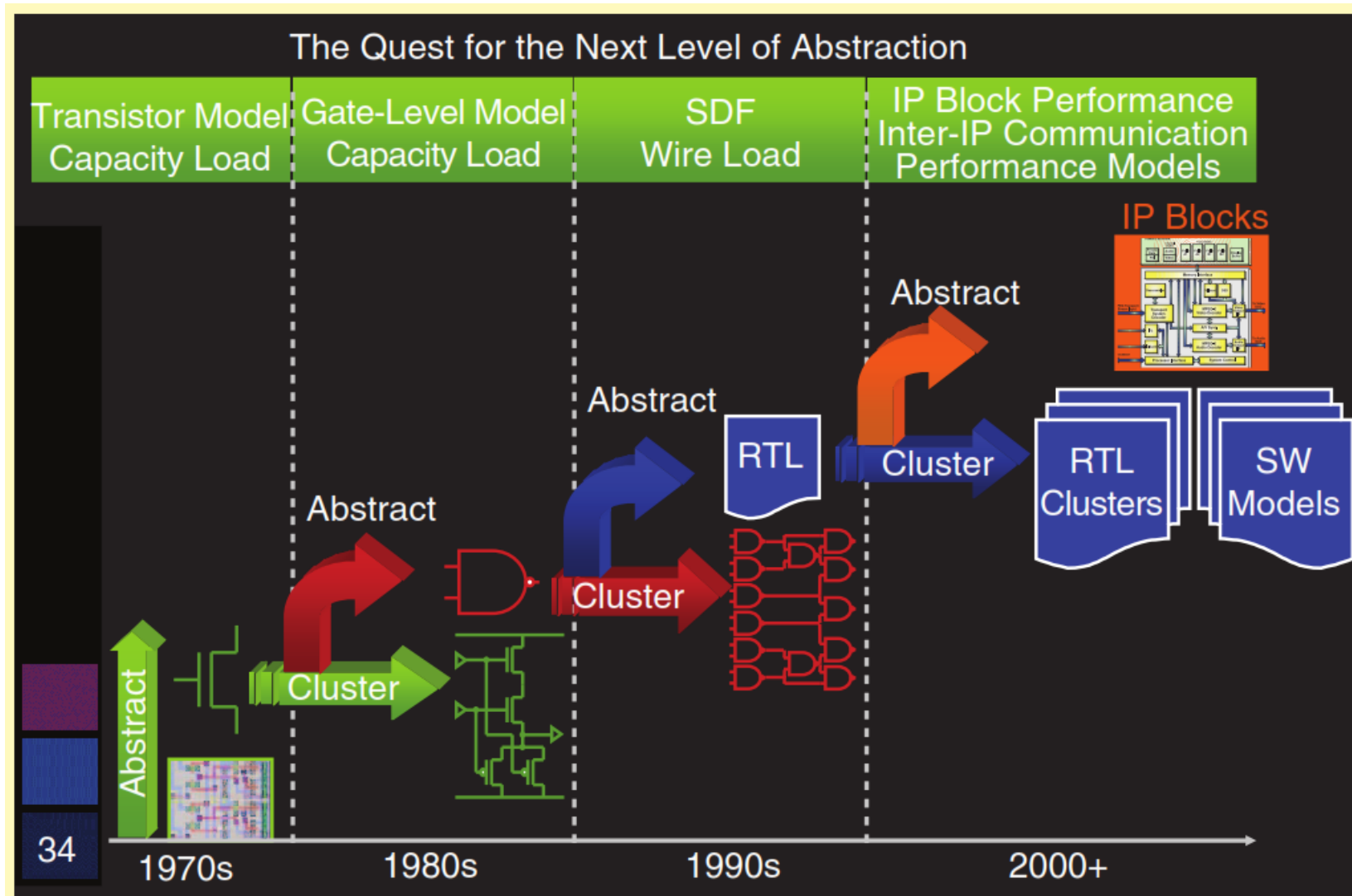
E' stato Carver Mead a  
inventare il termine «legge  
di Moore»



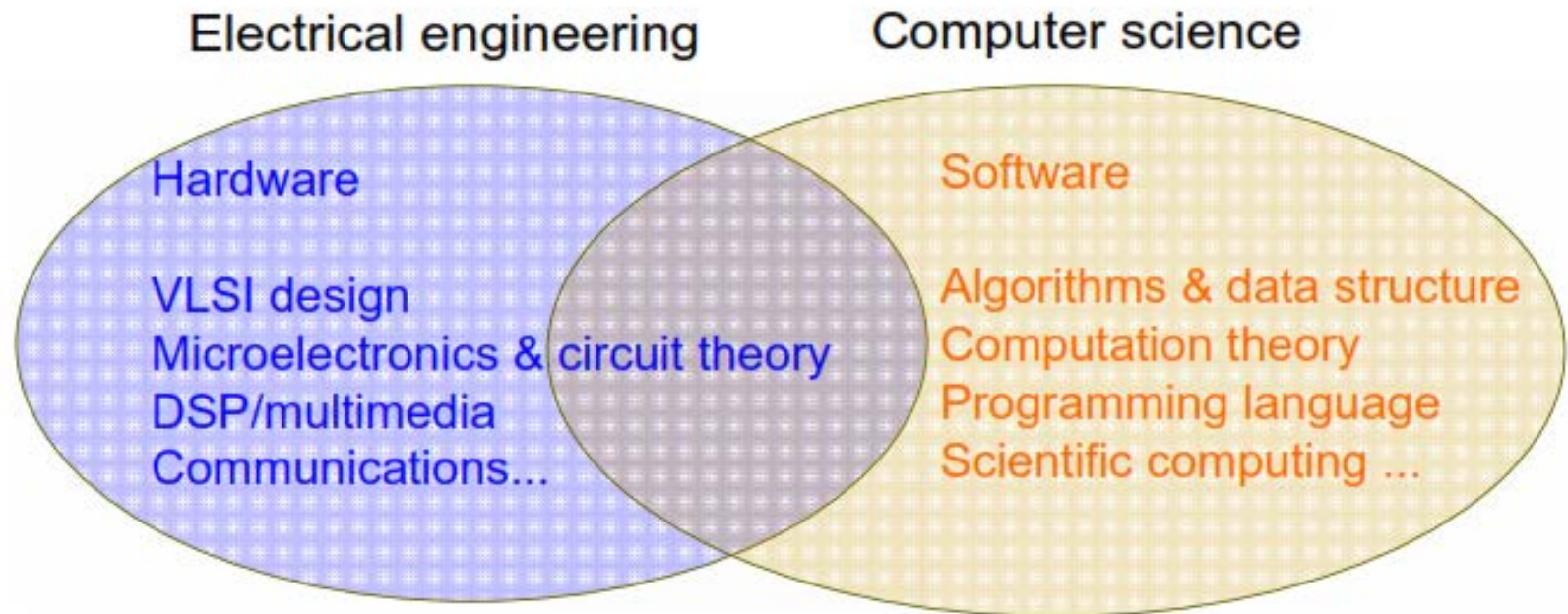
# Regole di layout



# Electronic Design Automation (EDA)



□ EDA, where HW and SW meet each other



Electronic Design Automation is concerned about hardware and software design in terms of

- correctness
- productivity
- optimality
- scalability

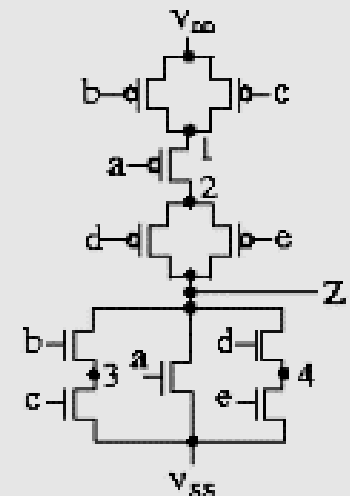
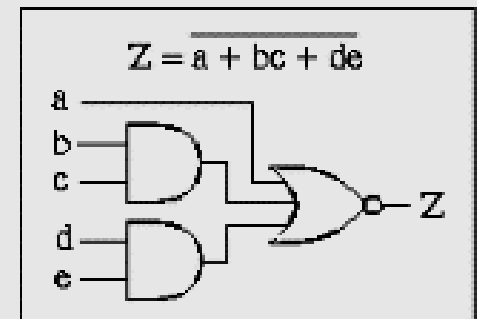
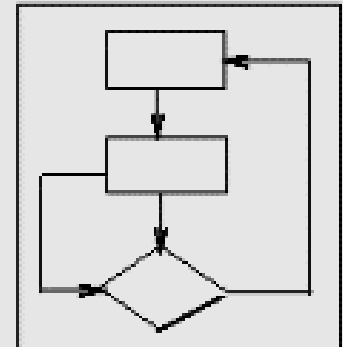
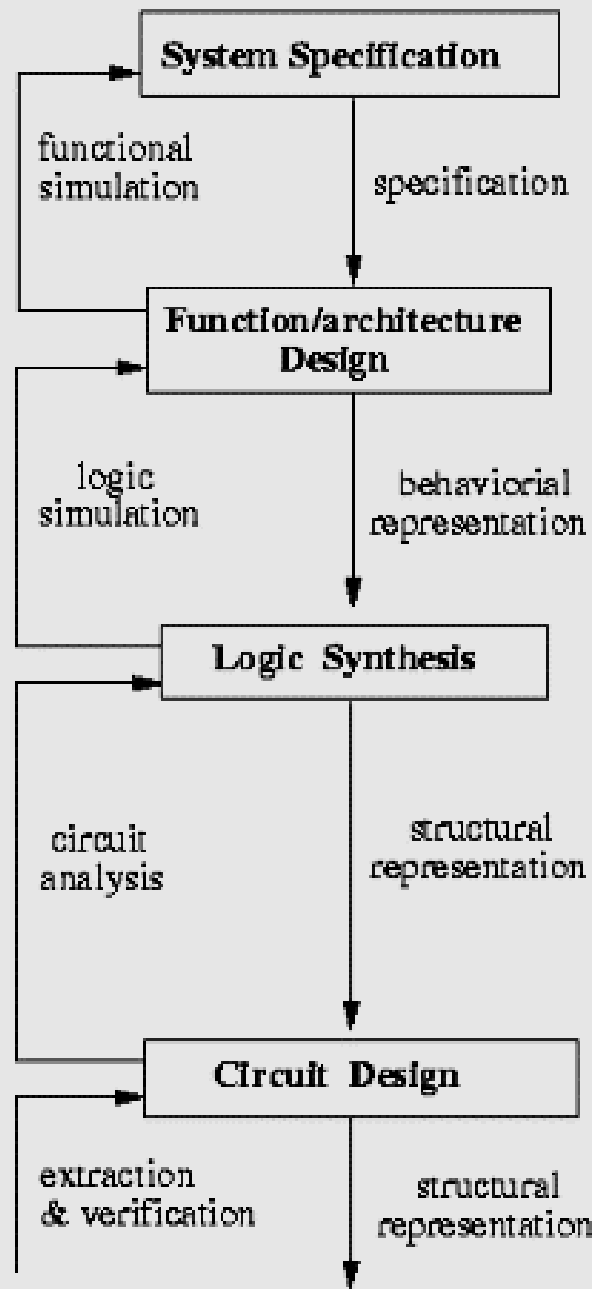
It also derives common methodologies which allow significant savings for semiconductor industry



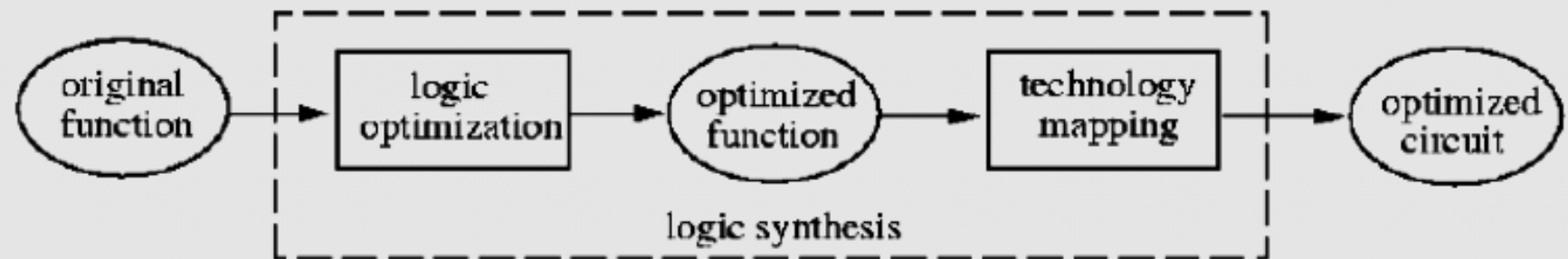
# VLSI Design

1. System specification
  2. Functional design
  3. Logic synthesis
  4. Circuit design
  5. Physical design and verification
  6. Fabrication
  7. Packaging
- ❑ Other tasks involved: testing, simulation, etc.
  - ❑ Design metrics: area, speed, power dissipation, noise, design time, testability, etc.
  - ❑ Design revolution: interconnect (not gate) delay dominates circuit performance in deep submicron era.
    - Interconnects are determined in physical design.
    - Shall consider interconnections in early design stages.

# VLSI Design



# Logic Design/Synthesis



- ❑ **Logic synthesis** programs transform Boolean expressions into logic gate networks in a particular library.
- ❑ Optimization goals: minimize area, delay, power, etc
- ❑ **Technology-independent** optimization: logic optimization
  - Optimizes Boolean expression equivalent.
- ❑ **Technology-dependent** optimization: **technology mapping/library binding**
  - Maps Boolean expressions into a particular cell library.

# Logic Optimization Examples

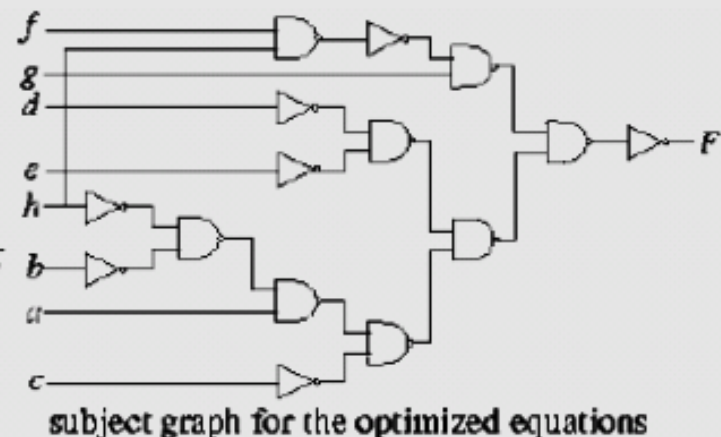
- **Two-level:** minimize the # of product terms.

- $F = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 \Rightarrow F = \bar{x}_2 + x_1\bar{x}_3.$

- **Multi-level:** minimize the #'s of literals, variables.

- E.g., equations are optimized using a smaller number of literals.

$t1 = a + b \ c;$	logic optimization →	$t1 = d + e;$
$t2 = d + e;$		$t2 = b + h;$
$t3 = a \ b + d;$		$t3 = a \ t2 + c;$
$t4 = t1 \ t2 + f \ g;$		$t4 = t1 \ t3 + f \ g \ h;$
$t5 = t4 \ h + t2 \ t3;$		
$F = t5';$		



- Methods/CAD tools: Quine-McCluskey method (exponential-time exact algorithm), Espresso (heuristics for two-level logic), SIS (heuristics for multi-level logic), ABC, etc.



## Physical Design

**Partitioning**

↓

**Floorplanning & Placement**

↓

**Routing**

↓

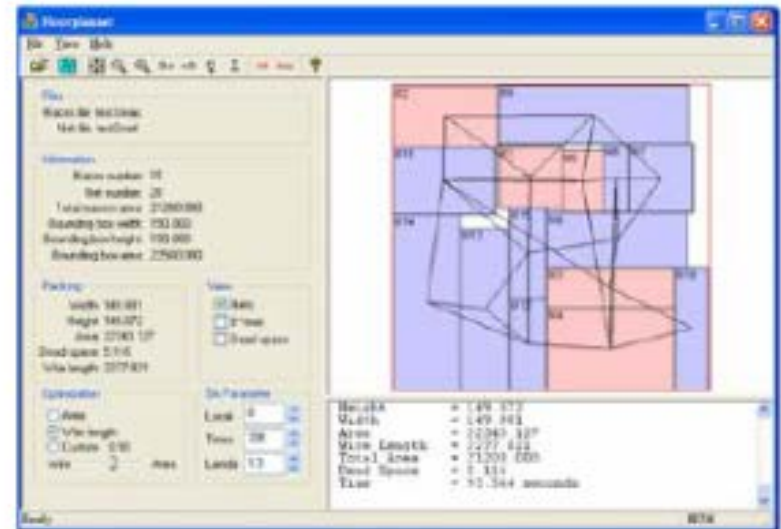
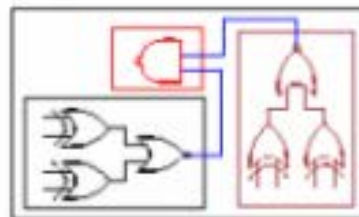
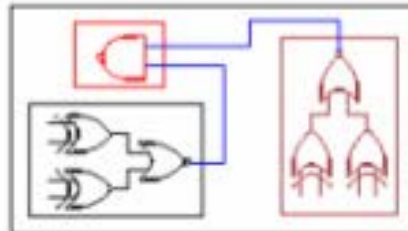
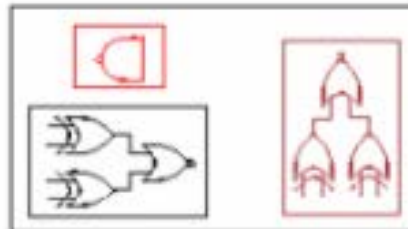
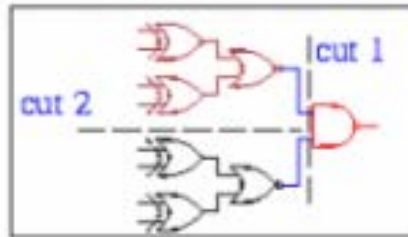
**Compaction**

↓

**Extraction & Verification**

The diagram illustrates the Physical Design process, showing the flow from Partitioning to Extraction & Verification, with corresponding circuit diagrams for each stage.

- Partitioning:** The circuit is divided into two partitions by a dashed line labeled "cut 2". A vertical line labeled "cut 1" separates a sub-circuit from the rest.
- Floorplanning & Placement:** The circuit components are placed into rectangular blocks.
- Routing:** Blue lines represent the routing paths connecting the blocks.
- Compaction:** The blocks are compacted together.
- Extraction & Verification:** The final compacted design is shown.



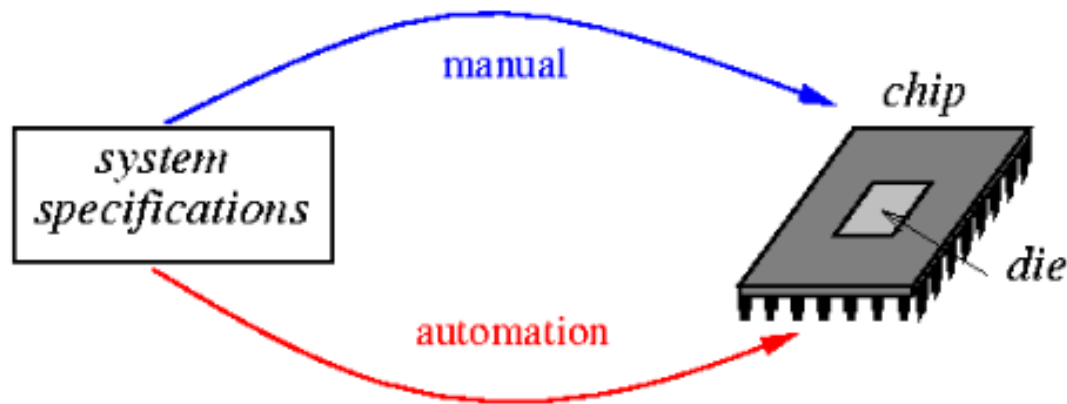
## B\*-tree based floorplanning system



## A routing system

# IC Design Considerations

---

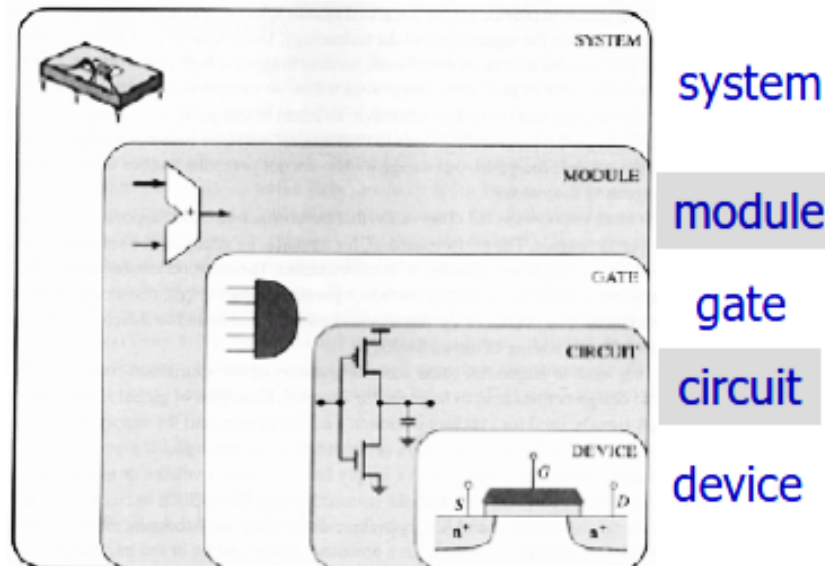


- ❑ Several conflicting considerations:
  - **Design complexity:** large number of devices/transistors
  - **Performance:** optimization requirements for high performance
  - **Time-to-market:** about a 15% gain for early birds
  - **Cost:** die **area**, packaging, testing, etc.
  - Others: power, signal integrity (noise, etc), testability, reliability, manufacturability, etc.

# Cope with Complexity

## □ Abstraction

- Trim away unnecessarily detailed info at proper abstract levels
- Design domains:
  - *Behavioral*: black box view
  - *Structural*: interconnection of subblocks
  - *Physical*: layout properties
  - Each design domain has its own hierarchy



# Gajski's Y-Chart

---

## BEHAVIORAL DOMAIN

Systems ●  
Algorithms ●  
Register transfers ●  
Logic ●  
Transfer functions ●

## STRUCTURAL DOMAIN

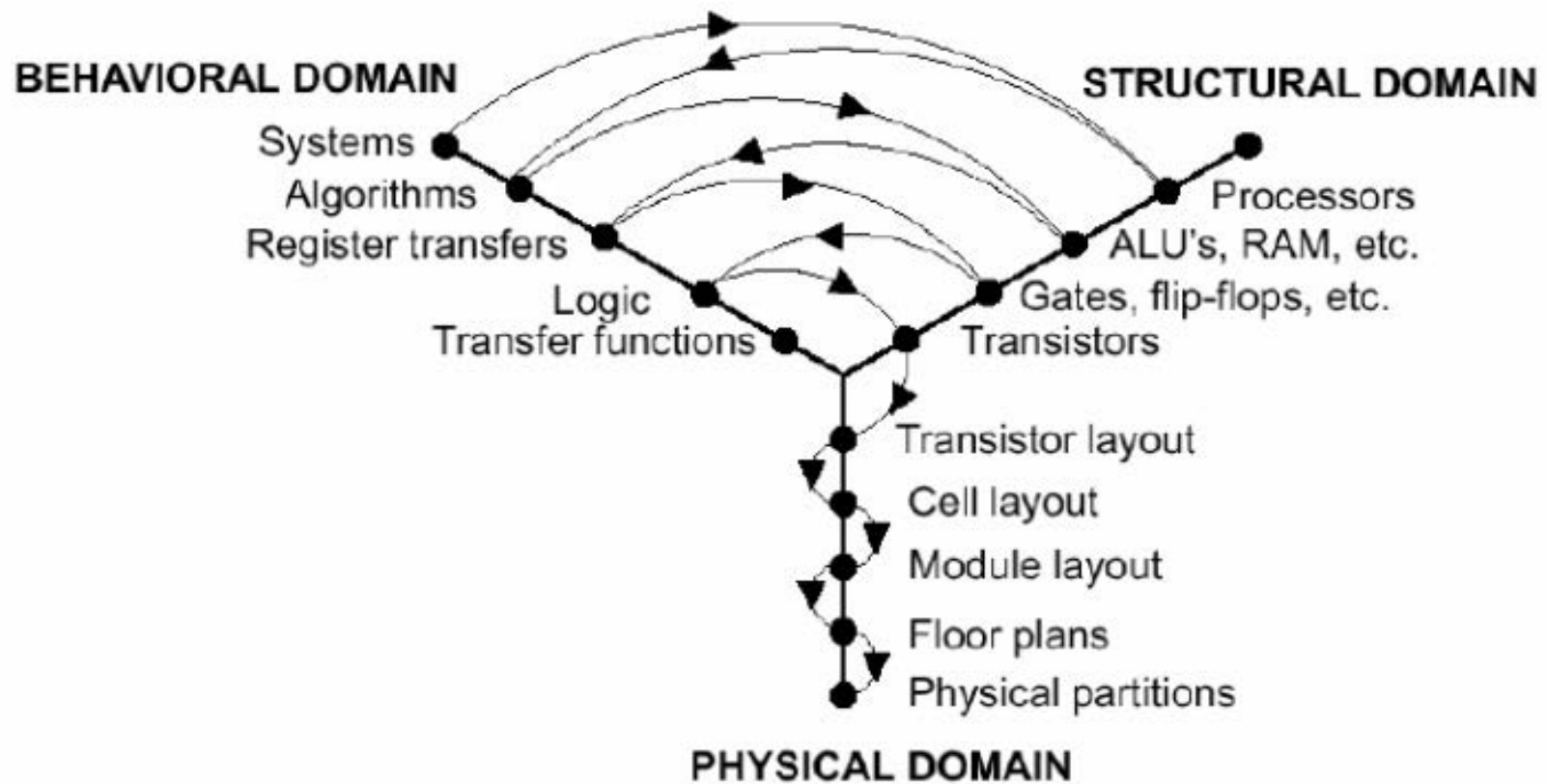
Processors ●  
ALU's, RAM, etc. ●  
Gates, flip-flops, etc. ●  
Transistors ●

● Transistor layout  
● Cell layout  
● Module layout  
● Floor plans  
● Physical partitions

## PHYSICAL DOMAIN



# Top-Down Structural Design



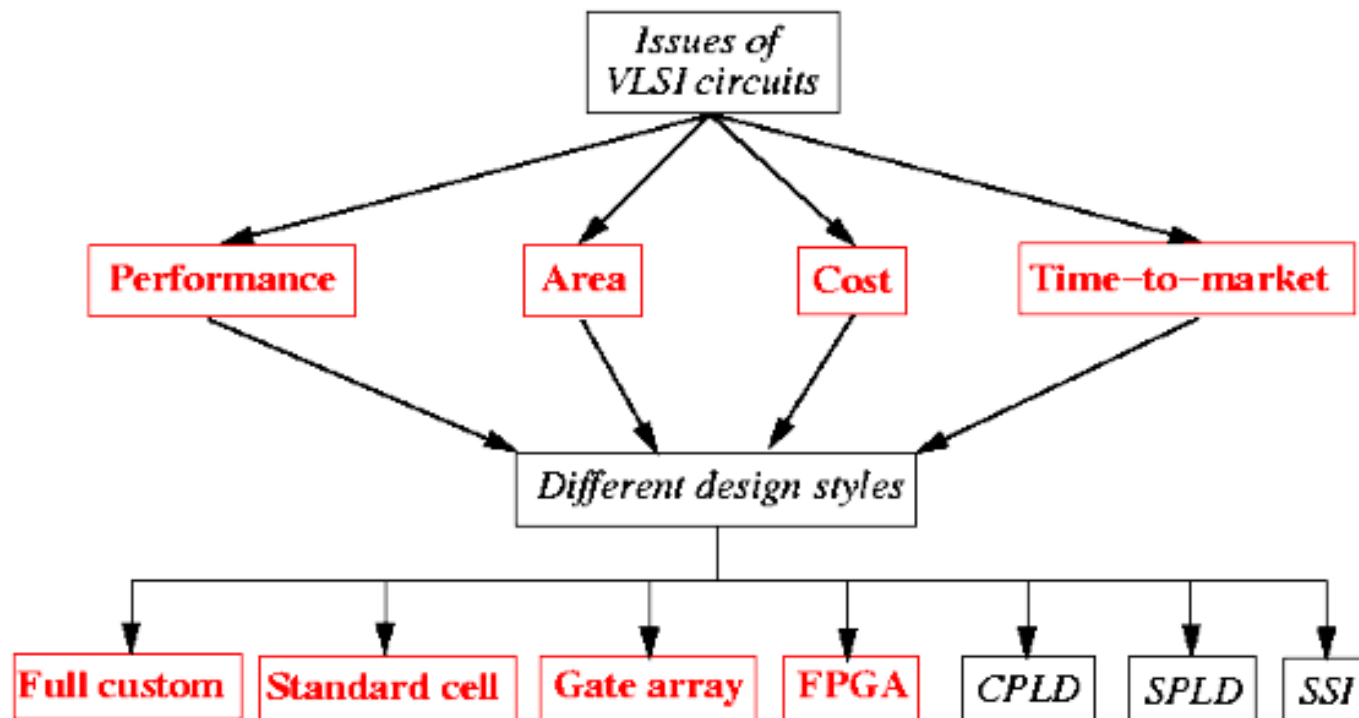
# Design Styles

---

- There are various design styles:
  - Full custom, standard cell, sea of gates, FPGA, etc.
- Why having different design styles?

# Design Styles

- Specific design styles shall require specific CAD tools



*Performance, Area efficiency, Cost, Flexibility*



# Very Large Scale Integration (VLSI)

- microprocessori
  - generici: processori per personal computer
  - specializzati: microcontrollori, DSP
- memorie
  - volatili: DRAM, SRAM
  - non volatili: EEPROM, FLASH
- sistemi audio/video, e altra elettronica di consumo
- telecomunicazioni



# Circuiti e sistemi VLSI

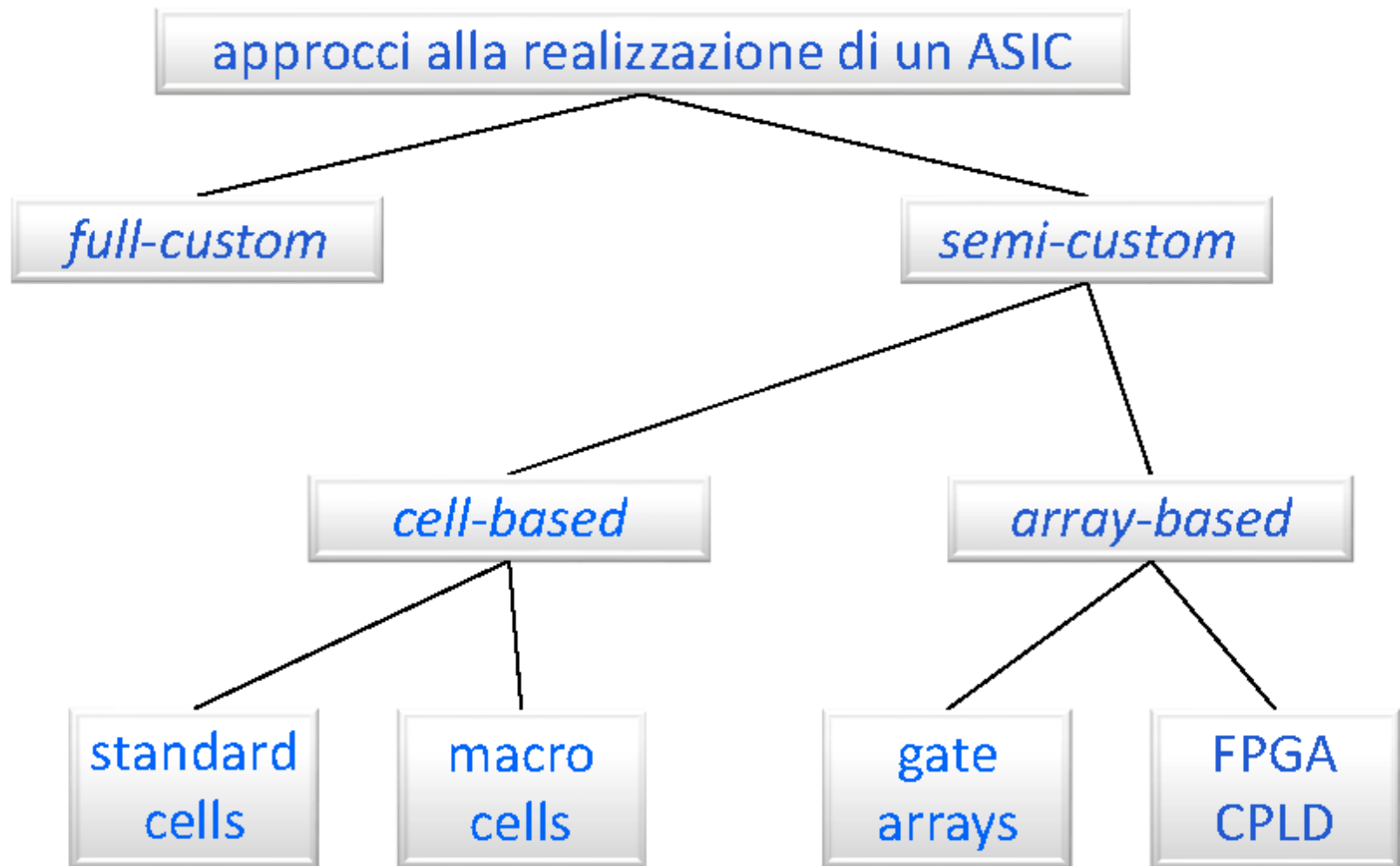
- L'integrazione migliora le prestazioni:
  - minori componenti parassiti = velocità più alta
  - riduzione del consumo di potenza (a parità di funzione)
  - riduzione delle dimensioni (o più funzioni a parità di dimensioni)
- L'integrazione riduce i costi di fabbricazione:
  - fabbricazione e assemblaggio (quasi) completamente automatizzati
  - forti economie di scala
- L'integrazione riduce gli effetti legati ai disturbi elettromagnetici
  - maggiore insensibilità alle interferenze
  - minore generazione di disturbi

## COMPONENTI A CATALOGO

- il sistema è realizzato con componenti a catalogo, disponibili a tutti (o quasi) quelli che vogliono acquistarli  
⇒ microprocessori, memorie, logica sparsa
- i costi di sviluppo sono suddivisi tra più clienti
- i componenti hanno caratteristiche più generiche di quelle richieste dal sistema
- la genericità si paga con prestazioni non ottimizzate

## COMPONENTI (SEMI)CUSTOM

- circuiti integrati progettati specificamente per il sistema in cui vengono utilizzati  
⇒ ASIC: Application Specific Integrated Circuits
- contengono solo l'hardware necessario alle funzioni che devono effettivamente svolgere
- hanno prestazioni ottimizzate rispetto alle specifiche del sistema



# approcci alla realizzazione di un ASIC

*full-custom*

il circuito viene interamente progettato fino a livello del singolo transistor

*semi-custom*

*cell-based*

standard  
cells

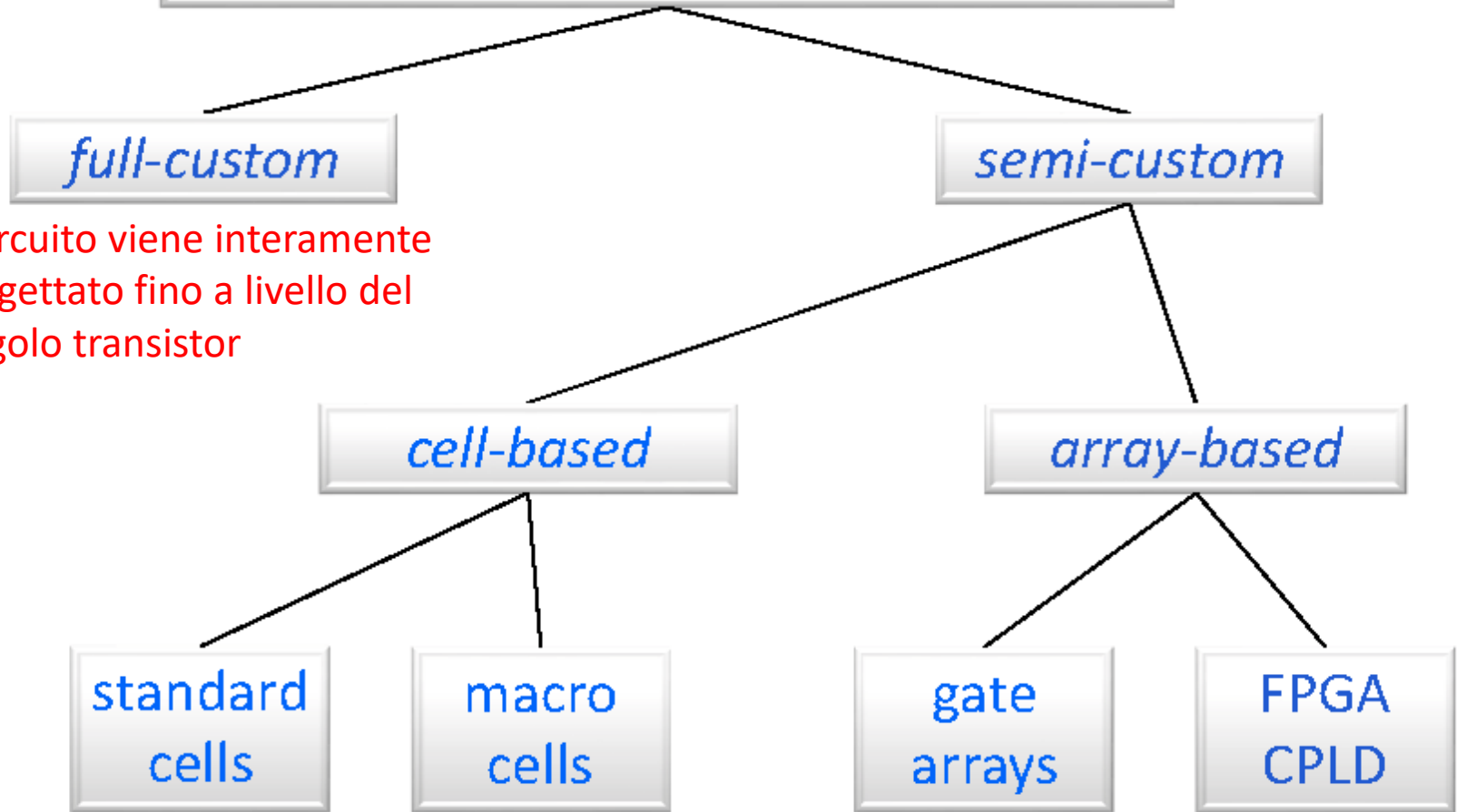
macro  
cells

il progetto è basato sull'utilizzo di singole celle standard (=come dimensioni e prestazioni) a livello elementare (NOR, NAND...) o di macroblocchi (sommatori, ...)

*array-based*

gate  
arrays

FPGA  
CPLD



# approcci alla realizzazione di un ASIC

```
graph TD; A[approcci alla realizzazione di un ASIC] --> B[full-custom]; A --> C[semi-custom]; B --> B1[il circuito viene interamente progettato fino a livello del singolo transistor]; C --> D[cell-based]; C --> E[array-based]; D --> F[standard cells]; D --> G[macro cells]; E --> H[gate arrays]; E --> I[FPGA CPLD]; E --> E1[il circuito integrato è già realizzato come insieme di gate digitali standard. Il progettista specifica solo le interconnessioni tra le gate];
```

*full-custom*

il circuito viene interamente progettato fino a livello del singolo transistor

*semi-custom*

*cell-based*

standard  
cells

macro  
cells

*array-based*

gate  
arrays

FPGA  
CPLD

il circuito integrato è già realizzato come insieme di gate digitali standard  
Il progettista specifica solo le interconnessioni tra le gate



# approcci alla realizzazione di un ASIC

*full-custom*

il circuito viene interamente progettato fino a livello del singolo transistor

*semi-custom*

*cell-based*

standard  
cells

macro  
cells

*array-based*

gate  
arrays

FPGA  
CPLD

Il progettista specifica solo le interconnessioni tra le gate; nel caso dei **gate arrays** le interconnessioni vengono specificate attraverso la maschera di un livello di metallizzazione

# approcci alla realizzazione di un ASIC

*full-custom*

il circuito viene interamente progettato fino a livello del singolo transistor

*semi-custom*

*cell-based*

standard  
cells

macro  
cells

*array-based*

gate  
arrays

FPGA  
CPLD

Nel caso dei **Field Programmable Gate Array** le interconnessioni sono realizzate tramite switch programmabili e il circuito viene specificato a livello software. La descrizione delle interconnessioni viene caricata ad ogni accensione

# approcci alla realizzazione di un ASIC

*full-custom*

il circuito viene interamente progettato fino a livello del singolo transistor

*semi-custom*

*cell-based*

standard  
cells

macro  
cells

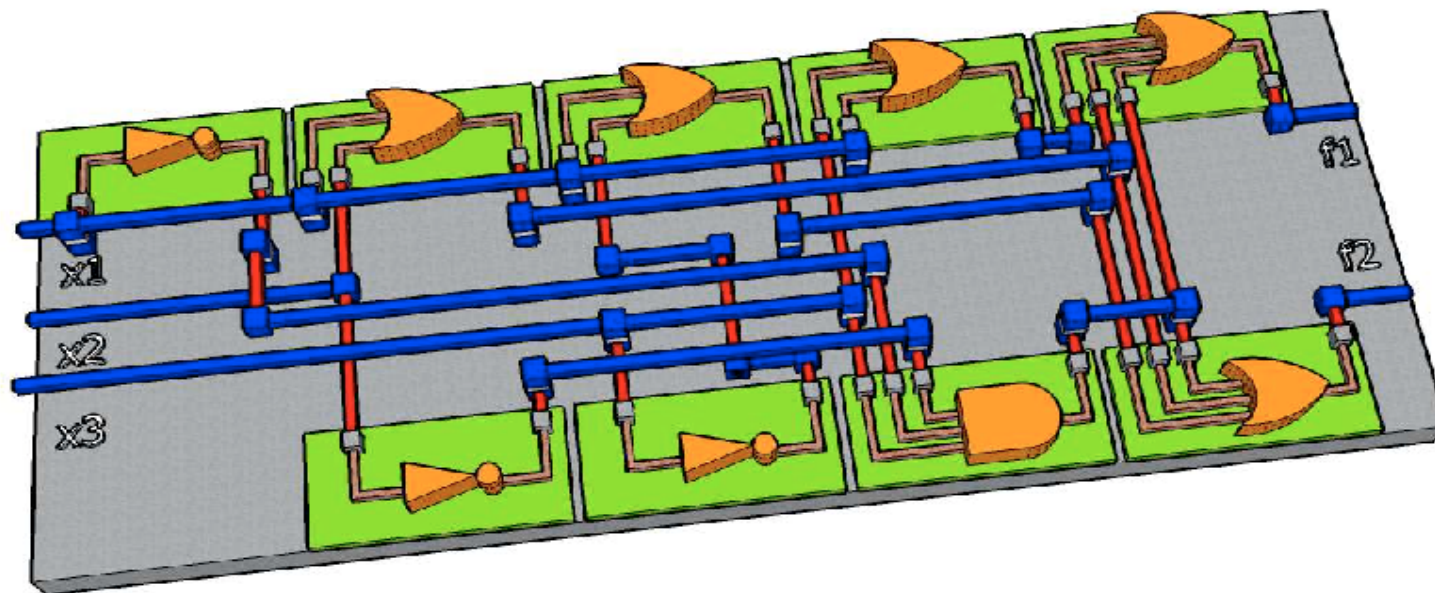
*array-based*

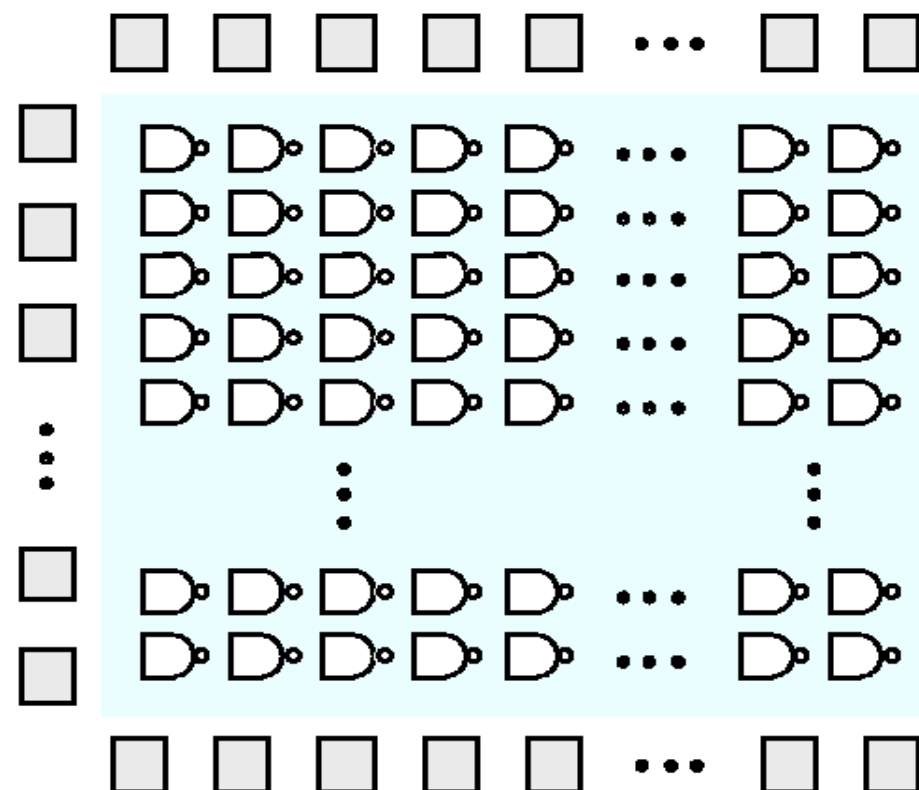
gate  
arrays

FPGA  
CPLD

Nel caso dei **Complex Programmable Logic Device** il circuito viene specificato a livello software. La descrizione delle interconnessioni è immagazzinata in una memoria non volatile

- libreria di celle **pre-progettate** (completamente, fino al layout)
- le celle vengono piazzate su più righe
- nei canali tra le righe di celle vengono realizzate le interconnessioni
- il layout completo viene inviato in fonderia per la fabbricazione





- matrice di celle **prefabbricate** senza interconnessioni
- le celle vengono interconnesse in modo da realizzare il circuito desiderato
- il layout delle interconnessioni viene inviato in fonderia per la fabbricazione
- vengono realizzate solo le metallizzazioni, perché il resto del circuito è prefabbricato



# Circuiti logici programmabili : PAL (piano AND programmabile)

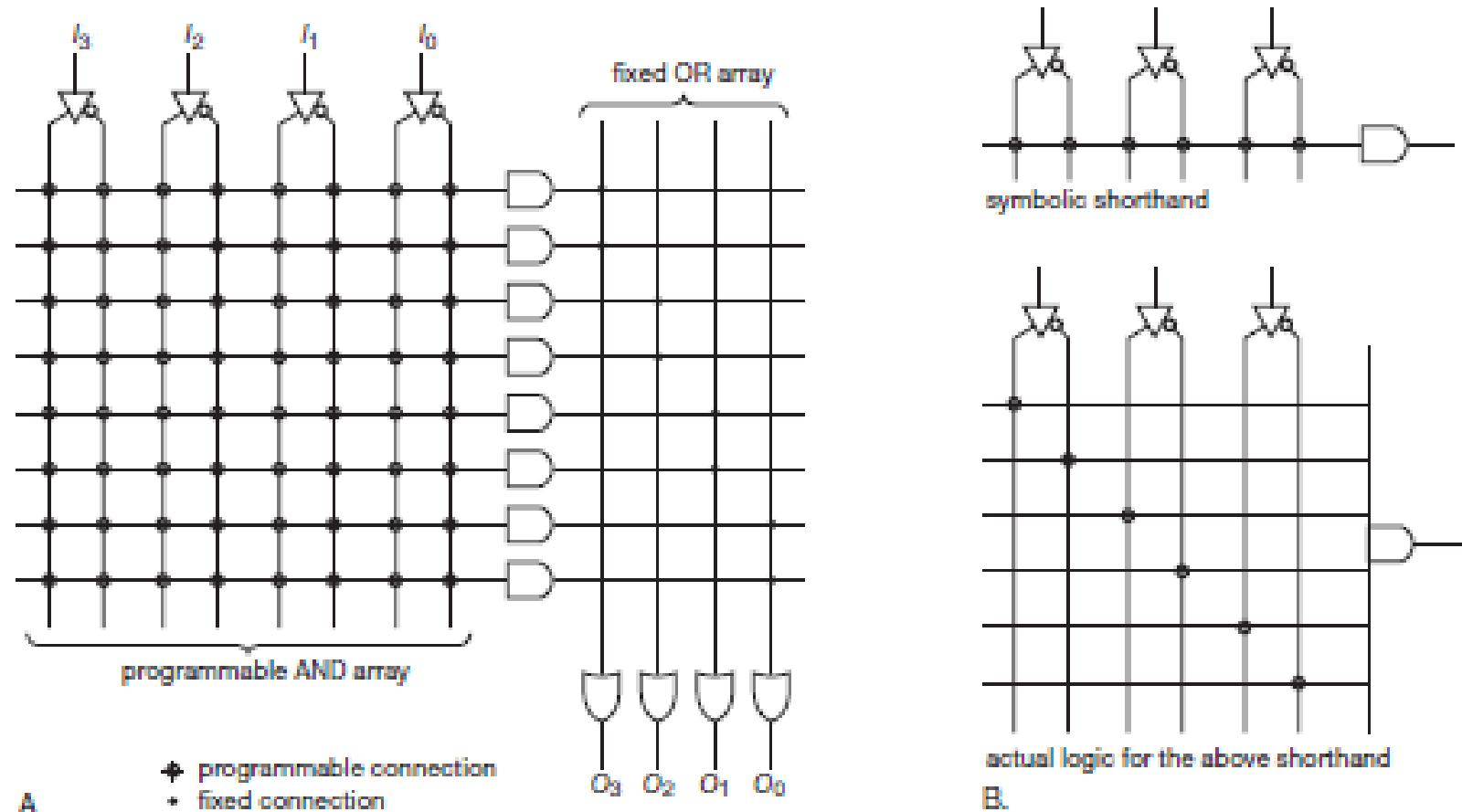
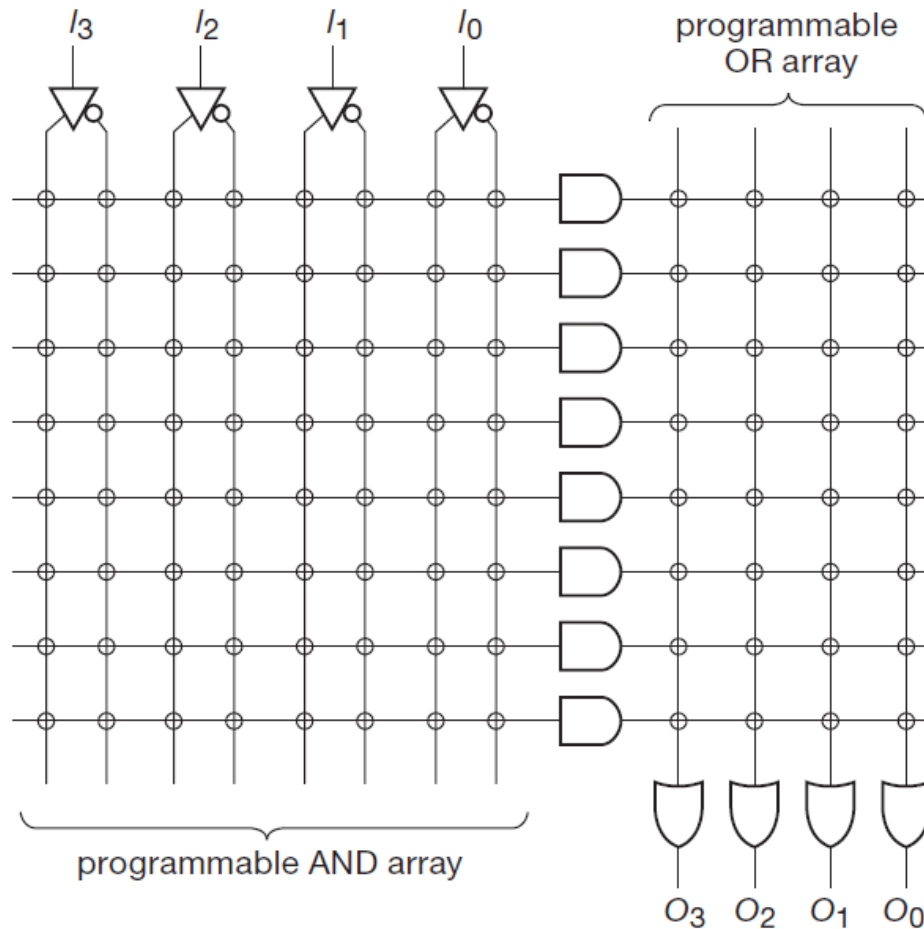


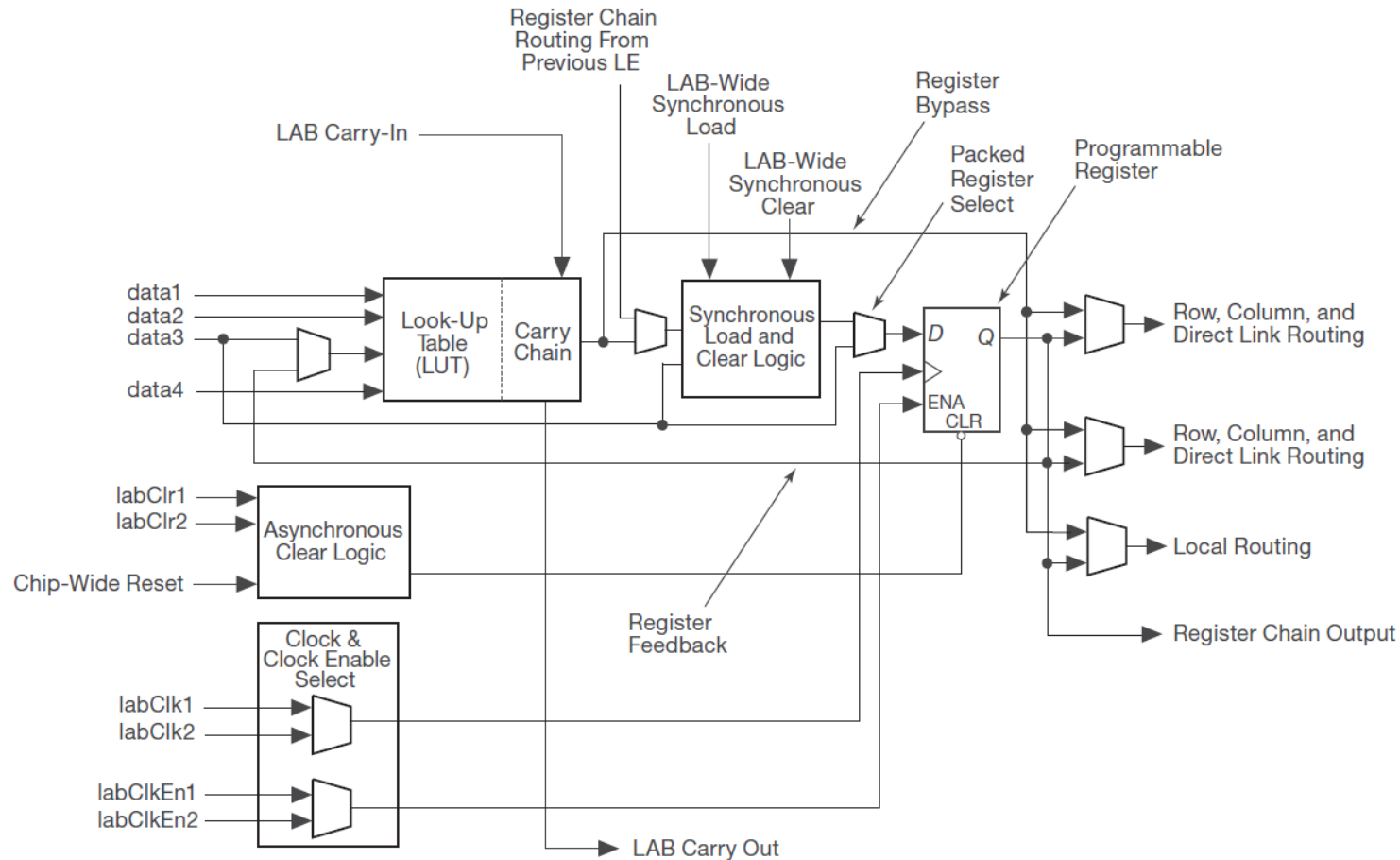
Figure 11.2. A. In a "PAL" (Programmable Array Logic) every available logic signal (or its complement) can be connected to many-input ANDs; the outputs of several such ANDs are OR-ed to form the logical outputs, each then passing through an output macrocell on the way to an output pin. B. Each AND gate has many inputs, shown in conventional shorthand and in fully expanded form.

# Circuiti logici programmabili : PLA (sia il piano AND che il piano OR programmabili)



**Figure 11.5.** A PLA (Programmed Logic Array) is more flexible than the PAL (Figure 11.2), by allowing programming of both the AND and OR arrays.

Field-programmable gate-array (FPGA) : ogni elemento contiene sia logica combinatoria, sia registri (flip-flop)



**Figure 11.6.** An FPGA logic element (LE), one of some tens of thousands in a typical FPGA, from the Altera “Cyclone II” family. The clocked flip-flop and lookup table (LUT) are decorated with extra logic and control signals from the larger logic array block (LAB), for flexible expansion. (Adapted from Altera Cyclone II Device Family datasheet, ©Altera Corporation, 2007.)

## Linguaggi di descrizione dell'hardware (HDL)

Permettono di descrivere la struttura circuitale del sistema digitale che si intende realizzare. Non sono programmi, non contengono istruzioni da eseguire, ma descrizioni di funzioni combinatorie, sequenziali e delle interconnessioni e relazioni tra queste

Verilog

VHDL

System-C

# Circuiti logici programmabili (CPLD e FPGA)

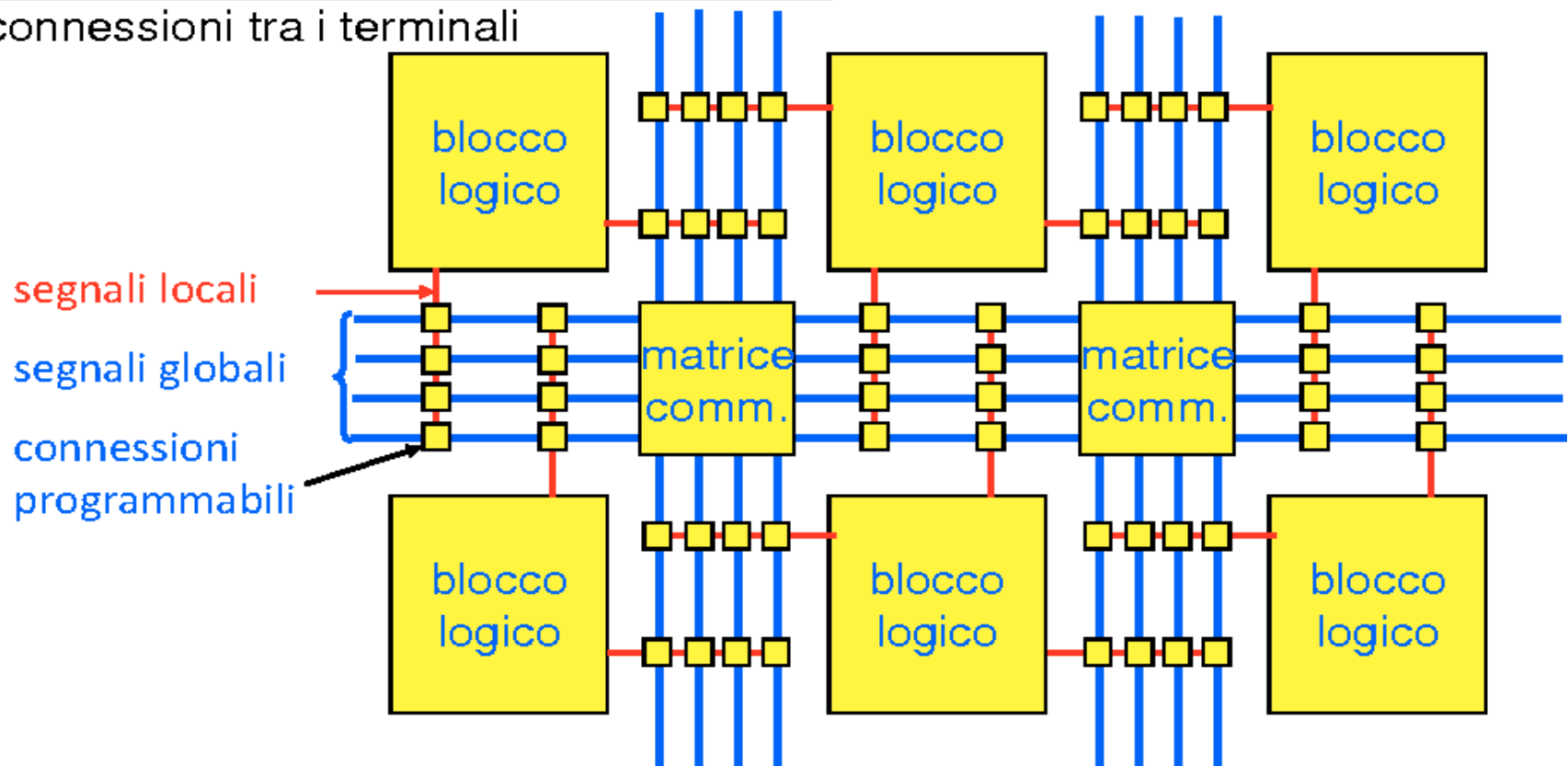
- Concetto base:
  - insieme di celle e interconnessioni prefabbricate ricongfigurabili via software
  - funzione logica svolta da ciascuna cella e collegamenti tra i terminali delle celle determinati commutando interruttori programmabili
- Interruttori programmabili (Programming Technology):
  - a fusibili (programmabili una volta sola)
  - MOSFET con memorie non volatili (EPROM, EEPROM, FLASH)
  - MOSFET con memorie SRAM



# Architettura di un Field Programmable Gate Array

**blocco logico (CLB, LE):** modulo di logica combinatoria (con uno o più registri) la cui funzione è programmabile

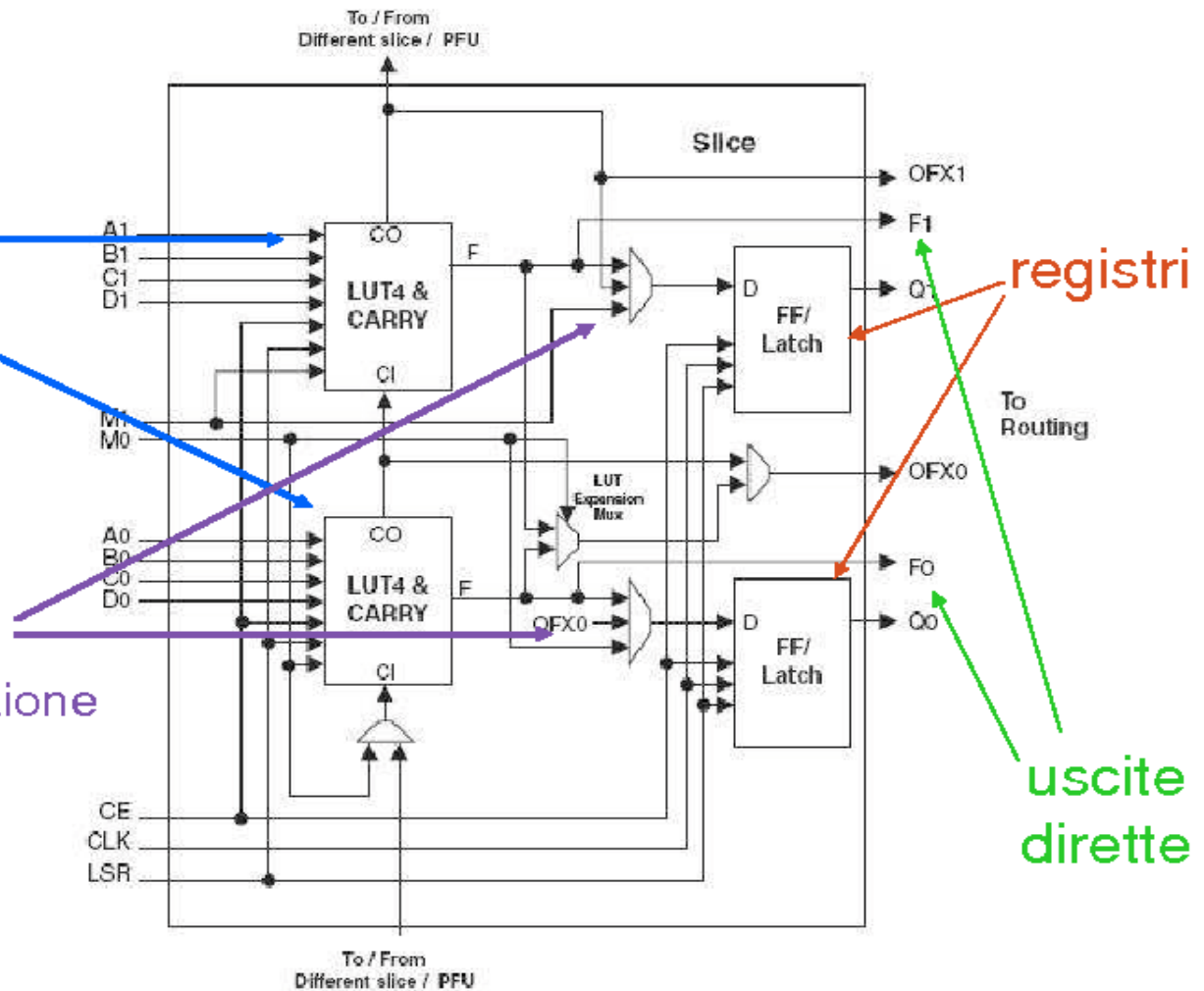
**matrice di commutazione (*switchbox*):** permette di decidere le connessioni tra i terminali



# Blocco logico elementare di una FPGA

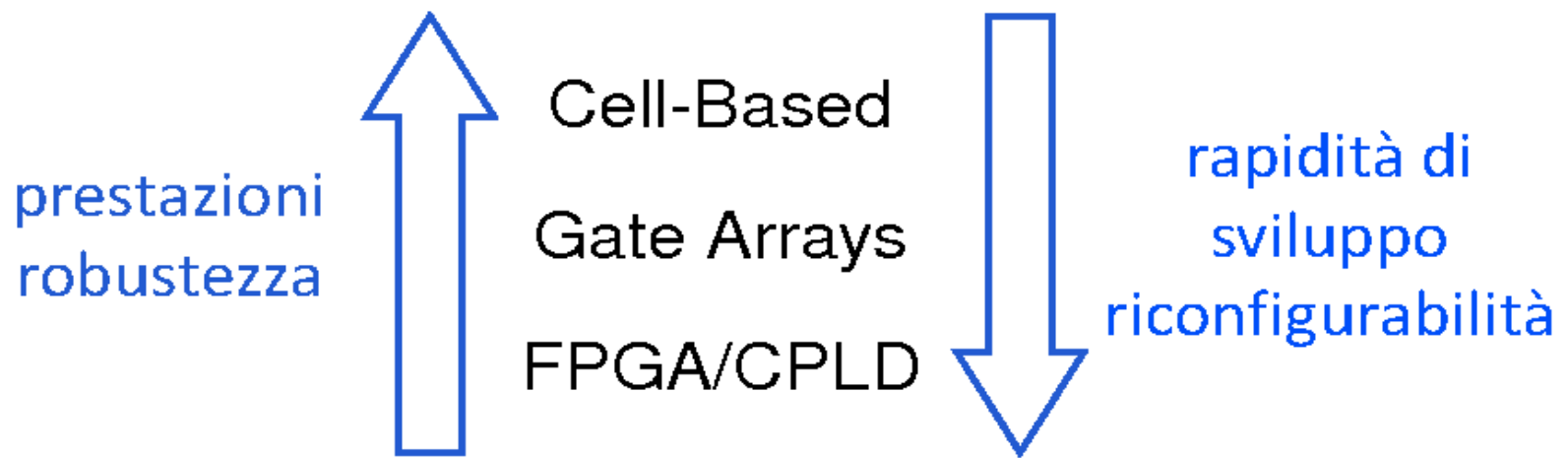
funzioni arbitrarie  
fino a 4 variabili

logica dedicata alla  
generazione/propagazione  
del riporto



## «Programmazione» (=«configurazione») di una CPLD/FPGA

- i CPLD/FPGA sono completamente prefabbricati: per essere trasformati in un circuito VLSI specifico devono solo essere programmati
- la programmazione consiste nel caricare nella memoria di configurazione del componente la sequenza di 0 e 1 che fissa la funzione svolta dai blocchi logici e l'effettivo percorso delle interconnessioni
- è concettualmente diversa dalla programmazione di un processore
  - microprocessore: le istruzioni risiedono in una memoria esterna, vengono caricate in successione, non modificano la configurazione dei circuiti
  - CPLD/FPGA: la sequenza di programmazione viene caricata prima dell'utilizzo del componente in una memoria interna, e determina l'effettiva topologia del circuito



COSTI: sono in funzione del volume di produzione richiesto

- bassi volumi: FPGA/CPLD
- alti volumi: Cell-Based

# Economia degli Application Specific Integrated Circuits (ASIC)

$$(\text{costo totale}) = (\text{costi fissi}) + (\text{costi variabili}) \times (\# \text{ di parti})$$

	Costi Fissi	Costi Variabili
FPGA/CPLD	5000	5
Gate Arrays	10000	1.5
Std Cells	20000	1

# Modello semplificato dei profitti e time-to-market

