

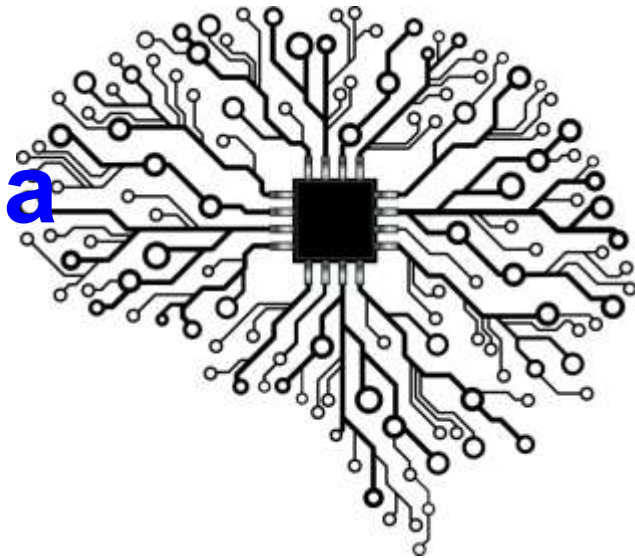
# Tehnici de Proiectare pentru Structuri VLSI

**TPS VLSI**

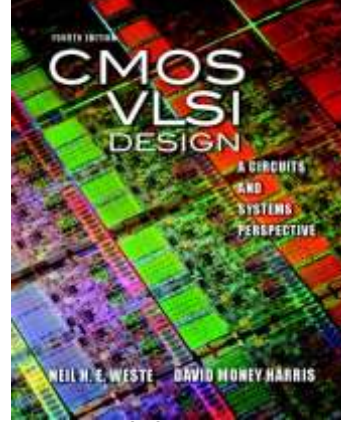
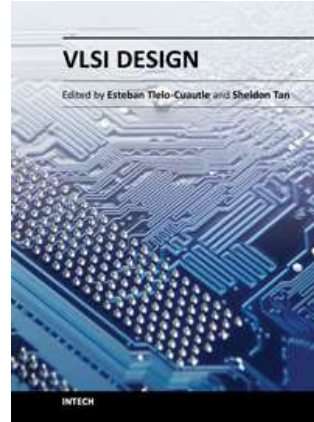
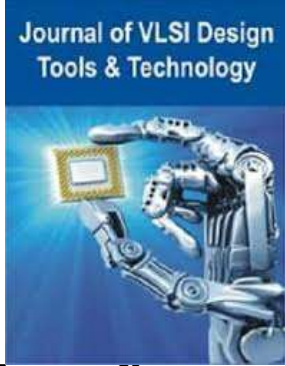
**Consultații**

**Marti 15:00-18:00**

**orice zi disponibila**



Pentium 4 Processor



## Bibliografie

Lupan Oleg, **TPS VLSI**. Note de curs. Chişinău, R.Moldova, 2018, – 150 pag.

J.F. Wakerly „**Circuite digitale - principiile si practicile folosite in proiectare**”, Teora, 2005 – 952 pag.

Baker, R. Jacob (2010). **CMOS: Circuit Design, Layout, and Simulation**, Third Edition. Wiley-IEEE. p. 1174. [ISBN 978-0-470-88132-3](https://doi.org/10.1002/9781118027554). <http://CMOSedu.com/>

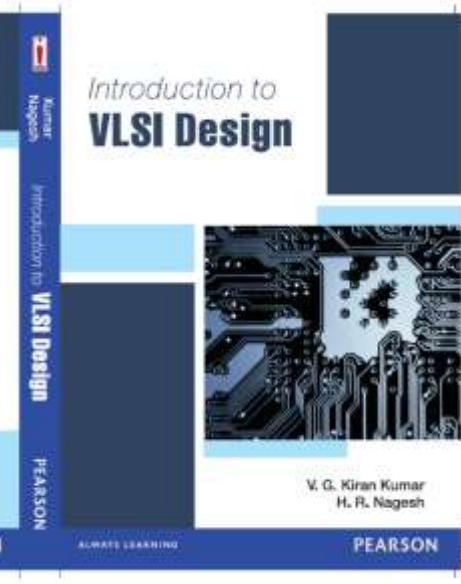
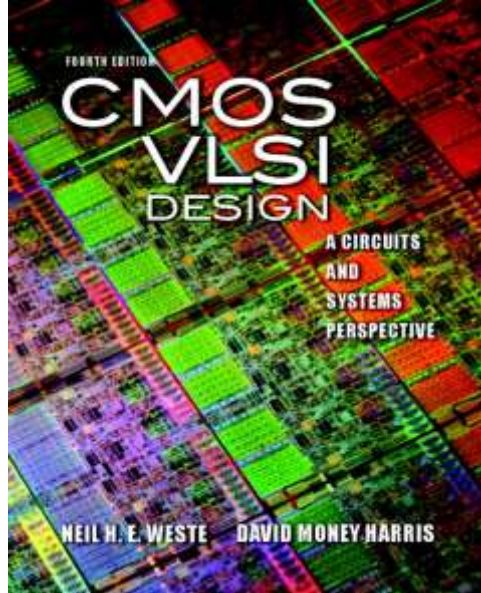
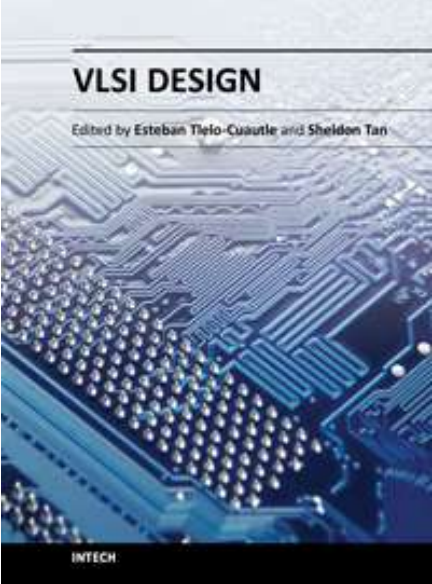
W. Maly, **Atlas of IC Technologies**, Menlo Park, CA: Benjamin/Cummings, 1987.

A. S. Grove, **Physics and Technology of Semiconductor Devices**, New York, NY: John Wiley & Sons, Inc., 1967.

G. E. Anner, Planar Processing Primer, New York, NY: Van Nostrsi Rheinhold, 1990.

T. E. Dillinger, VLSI Engineering, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1988.

S.M. Sze, **VLSI Technology**, New York, NY: McGraw-Hill, 1983.



# Foundation of VLSI Design



# De ce?

## Programul TPS VLSI

urmărește instruirea studenților în proiectarea circuitelor VLSI analogice și digitale.

Aici sunt vizate aspecte de proiectare și de implementare cu un accent pe tehnologiile CMOS, bipolar și pe BiCMOS, layout, testare, etc.

## Oportunități de angajare

Proiectarea circuitelor integrate analogice și digitale (**Microchip**, Silicon Service, **Infineon**, Microchip).

Proiectarea de circuite și sisteme electronice folosind micro-controlere și procesoare de semnal (**Continental** și Siemens).

Verificarea funcțională a sistemelor digitale (NoBug-Iași)

# **1. Introducere în proiectarea sistemelor VLSI**

## **1.1 Perspectiva istorica**

## **1.2 Procesul de proiectare VLSI**

## **1.3 Ierarhia Proiectarii.**

## **1.4 Conceptele de regularitate, modularitate și localitate.**

## **1.5 Stiluri de proiectare VLSI**

### **1.5.1 FPGA (Field Programmable Gate Array)**

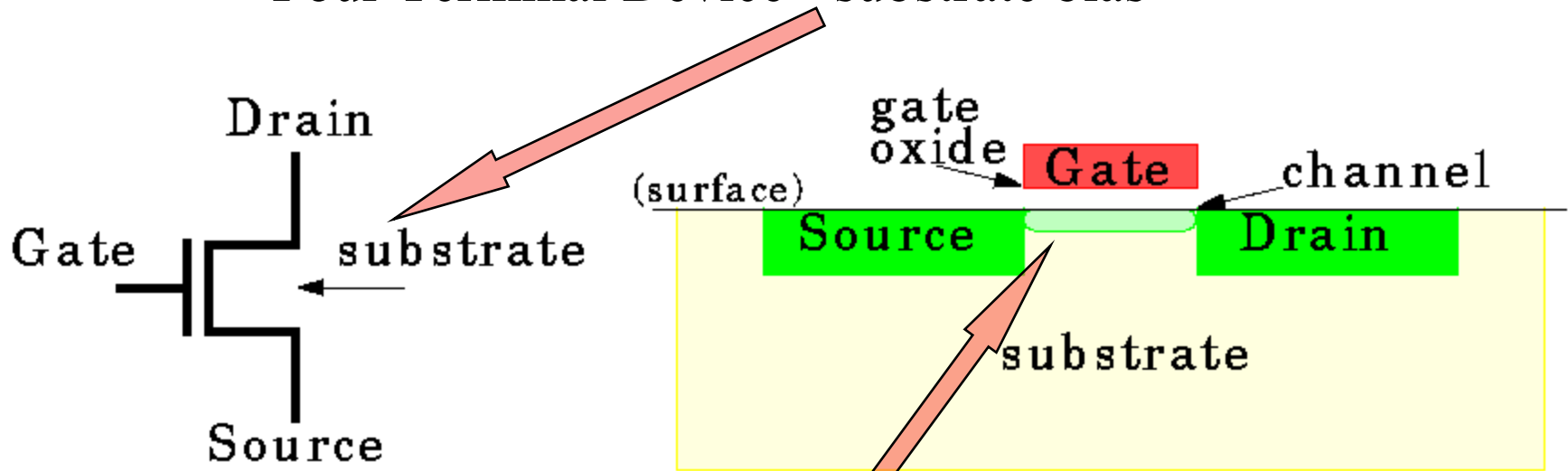
### **1.5.2 Proiectarea matricilor de porți logice**

### **1.5.3 Proiectarea bazată pe celule standard**

### **1.5.4 Proiectarea complet personalizată (full custom design).**

# N-Channel Enhancement mode MOS FET

- Four Terminal Device - substrate bias



–The “self aligned gate” - key to CMOS

# IC Products

- Processors
  - CPU, DSP, Controllers
- Memory chips
  - RAM, ROM, EEPROM
- Analog
  - Mobile communication, audio/video processing
- Programmable
  - PLA, FPGA
- Embedded systems
  - Used in cars, factories
  - Network cards
- System-on-chip (SoC)



# VLSI: Very Large Scale Integration

- Integration: Integrated Circuits
  - multiple devices on one substrate
- How large is Very Large?
  - SSI (small scale integration)
    - 7400 series, 10-100 transistors
  - MSI (medium scale)
    - 74000 series 100-1000
  - LSI 1,000-10,000 transistors
  - **VLSI > 10,000 transistors**
  - ULSI/SLSI (some disagreement)

# Help from Computer Aided Design tools

- **Tools - Instrumente**
  - Editors
  - Simulators
  - Libraries
  - Module Synthesis
  - Place/Route
  - Chip Assemblers
  - Silicon Compilers
- **Experts – Experti**
  - Logic design
  - Electronic/circuit design
  - Device physics
  - Artwork
  - Applications - system design
  - Architectures

În ultimele două decade, industria electronică a înregistrat o dezvoltare fără precedentă, în principal datorită rezultatelor de excepție obținute în

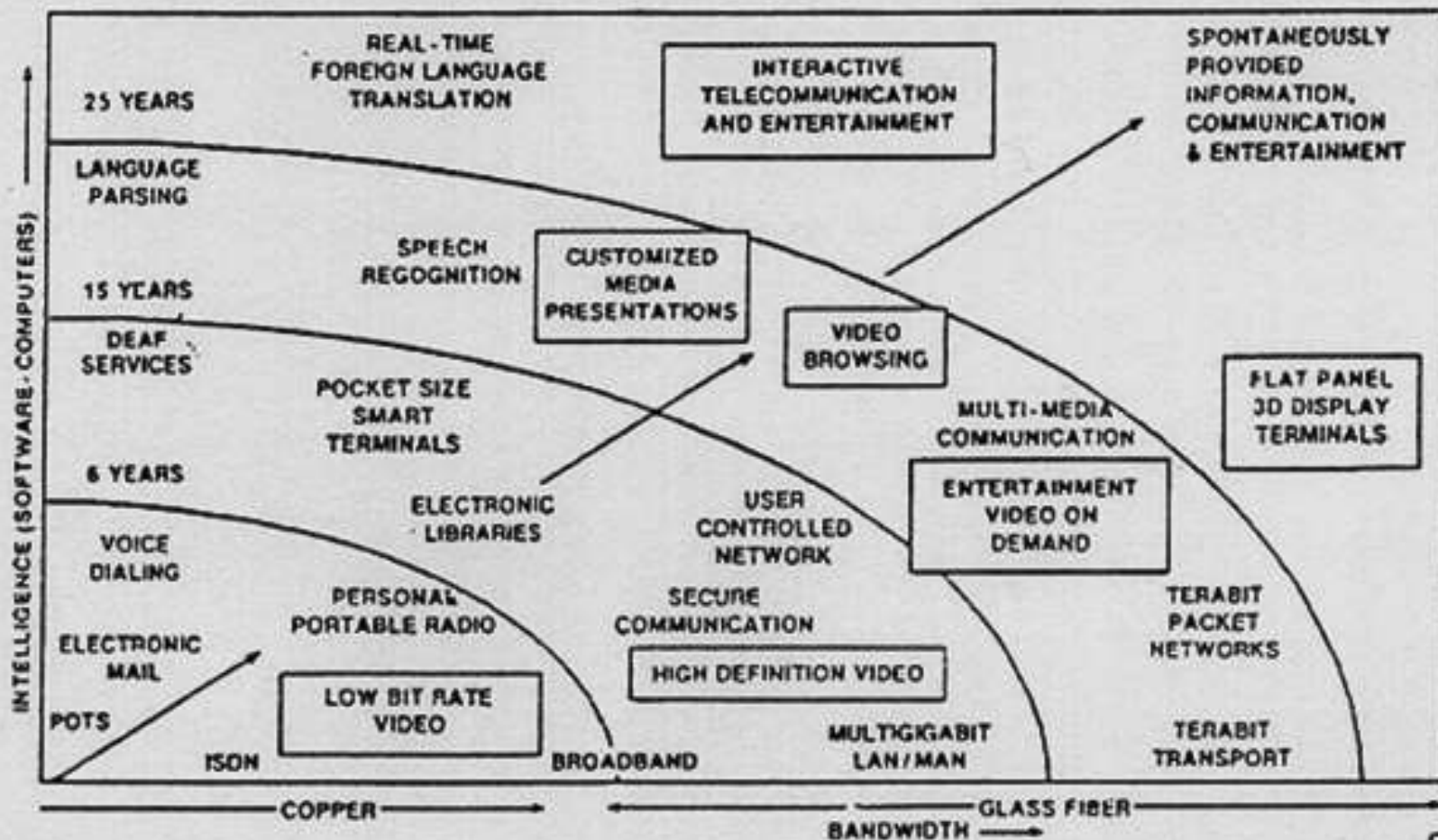
- + tehnologiile circuitelor integrate,
- + proiectarea sistemelor larg integrate – pe scurt - datorită apariției **VLSI**.

Numărul de aplicații ale circuitelor integrate în calculul de înaltă performanță, telecomunicații și aparatură electronică a crescut rapid și substanțial.

A VLSI integrated-circuit [die](#)

O [matriță-pastilă](#) cu circuit integrat VLSI

Figura 1.1 arata **tendintele principale în tehnologia informatiei** de-a lungul următoarelor decade. Tehnologiile de varf din prezent, cum ar fi video la viteza redusa (low-bit rate) de transmisie si comunicatiile fara fir (wireless) deja asigura utilizatorului, atat portabilitate, cat si o cantitate importanta de putere de procesare.



Source: Bellcore

Figura 1.1: Tendintele predominante în tehnologiile serviciilor informationale.

# Example: Intel Processor Sizes

Silicon Process Technology    1.5 $\mu$     1.0 $\mu$     0.8 $\mu$     0.6 $\mu$     0.35 $\mu$     0.25 $\mu$

Intel386™ DX  
Processor

Intel486™ DX  
Processor

Pentium® Processor

Pentium® Pro &  
Pentium® II Processors

Source: <http://www.intel.com/>

# 80386

- 32-bit processor (1985)
  - Modern x86 ISA
- Characteristics
  - 1.5-1  $\mu\text{m}$  process
  - **275k transistors**
  - 16-33 MHz
  - 32-bit word size
  - 100-pin PGA
- 32-bit datapath,  
microcode ROM,  
synthesized control

# Pentium

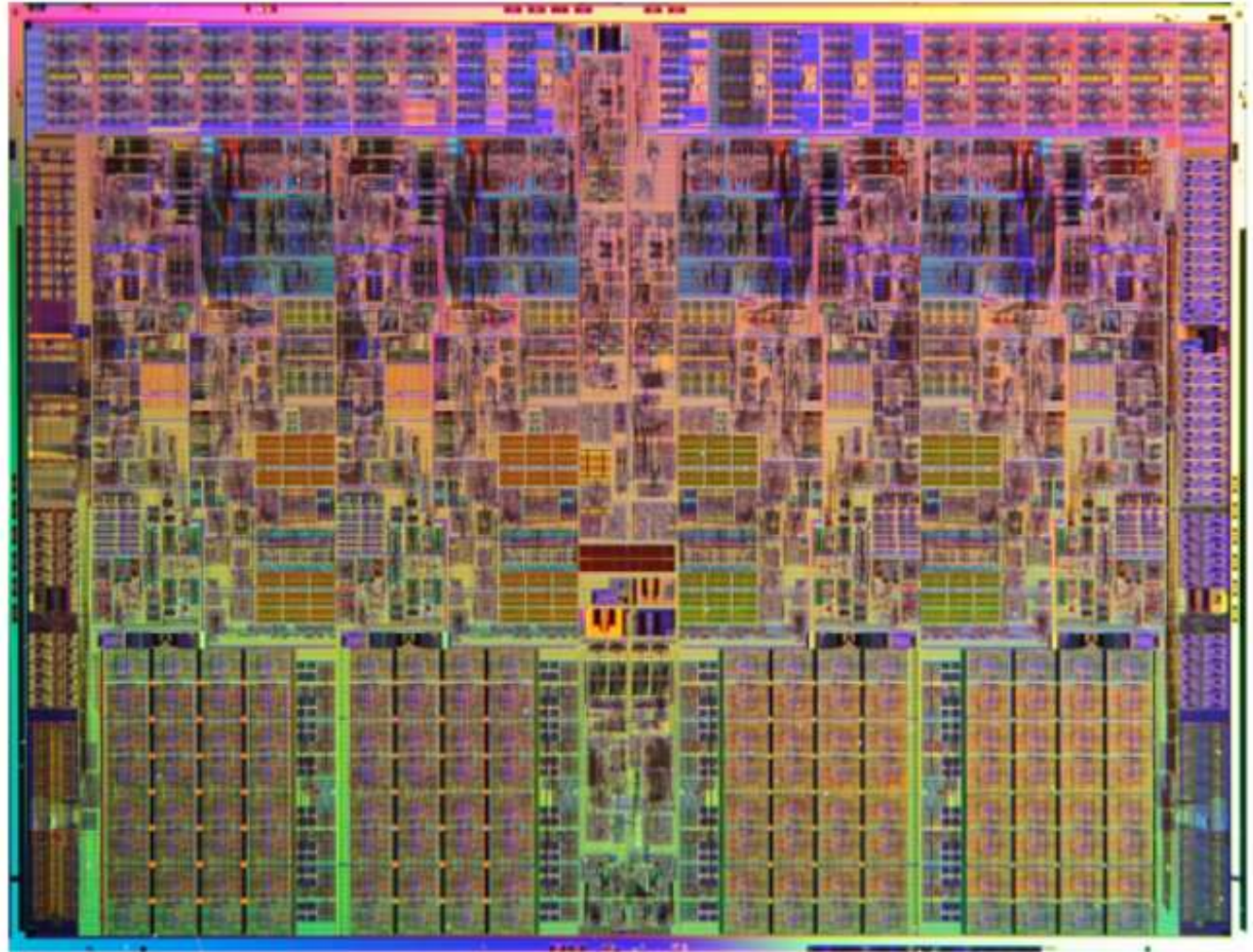
- Superscalar (1993)
  - 2 instructions per cycle
  - Separate 8KB I\$ & D\$
- Characteristics
  - 0.8-0.35  $\mu\text{m}$  process
  - **3.2M transistors**
  - 60-300 MHz
  - 32-bit word size
  - 296-pin PGA
- Caches, datapath, FPU, control

# Pentium 4

- Deep pipeline (2001)
  - Very fast clock
  - 256-1024 KB L2\$
- Characteristics
  - 180 – 90 nm process
  - **42-125M transistors**
  - 1.4-3.4 GHz
  - 32-bit word size
  - 478-pin PGA
- Units start to become invisible on this scale



- ◆ Intel Core i7
- ◆ > ¾ billion trans.
- ◆ Very expensive to design
- ◆ Very expensive to manufacture
- ◆ Not viable unless the market is very large



1,000,000,000 (one [billion](#))

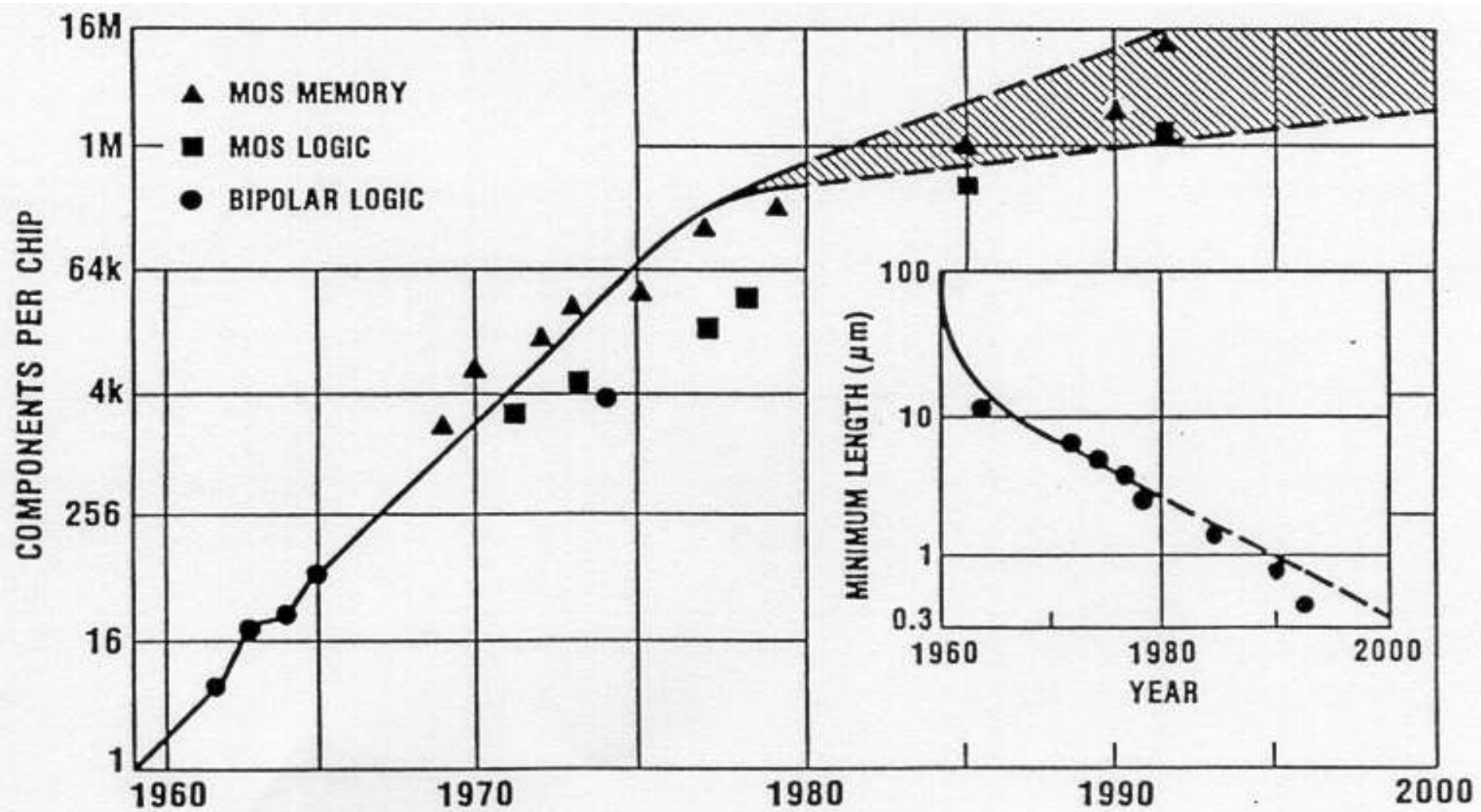
$$1 \times 10^9$$

Era (nr. de blocuri logice pe circuit)	Data	Complexitatea
Tranzistor sigular	1959	mai putin de 1
Unitate logica (poarta)	1960	1
Multi-functii	1962	2 – 4
Functii complexe	1964	5 – 20
Integrarea la scara medie (MSI)	1967	20 – 200
Integrarea la scara larga (LSI)	1972	200 – 2000
Integrarea la scara foarte larga (VLSI)	1978	2000 – 20000
Integrarea la scara ultra larga (ULSI)	1989	20000 – ?

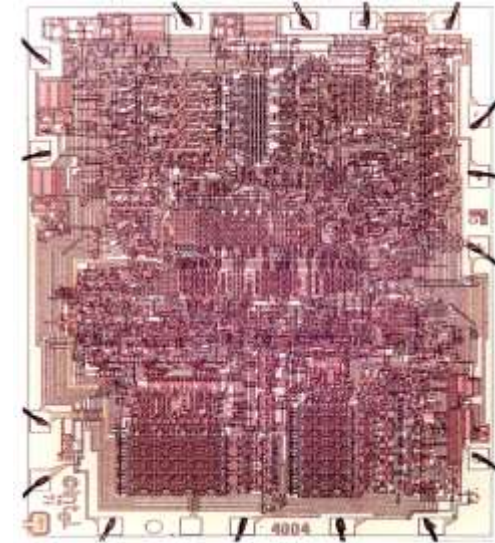
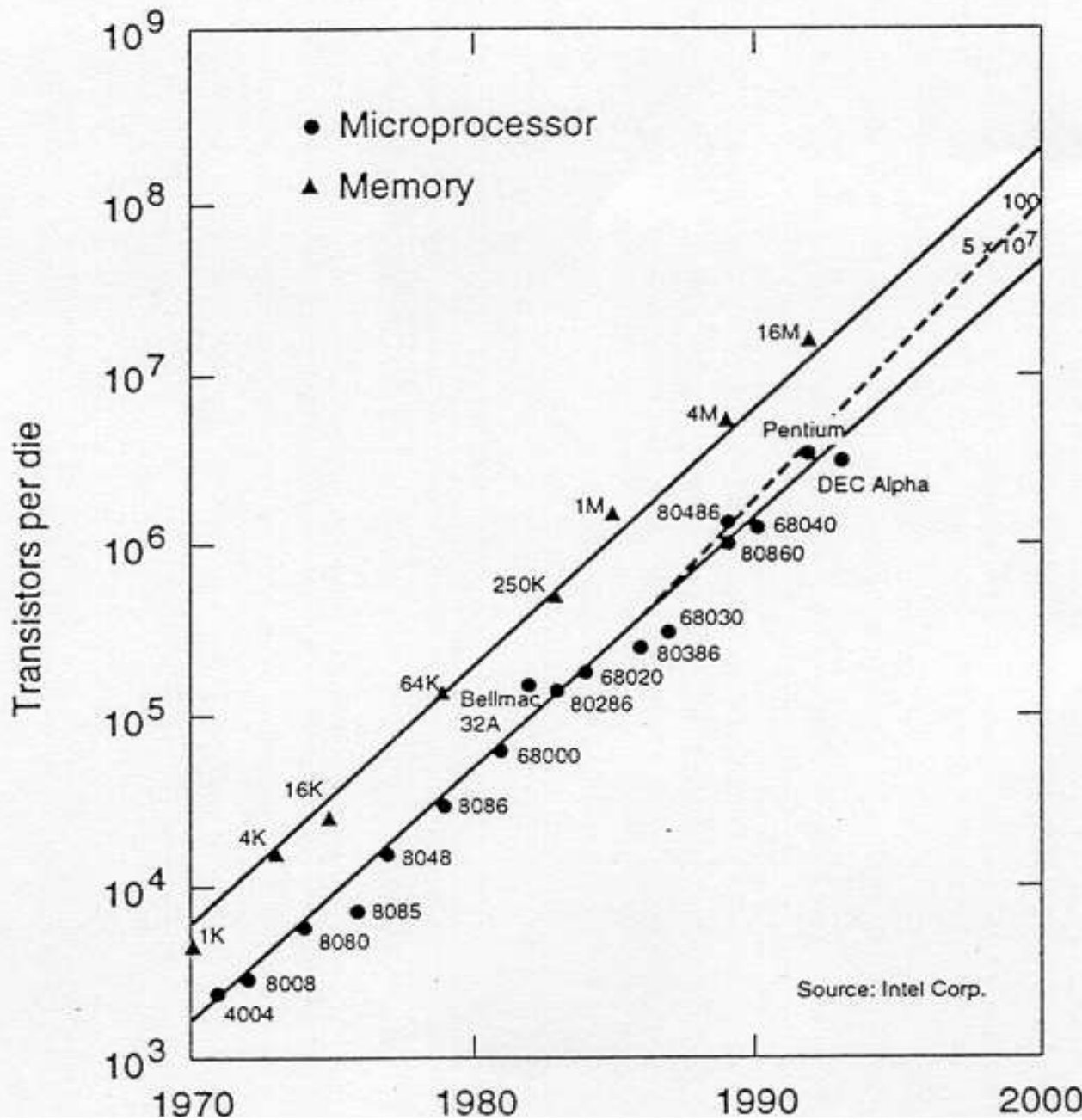
*Tabelul 1.1: Evolutia complexitatii logicii în circuitele integrate*

Cel mai important mesaj transmis de catre tabela de mai sus se refera la cresterea exponentiala a complexitatii logicii la nivel de pastila monolitica. Integrarea monolitica a unui numar mare de functii pe o singura pastila, în mod uzual, asigura:

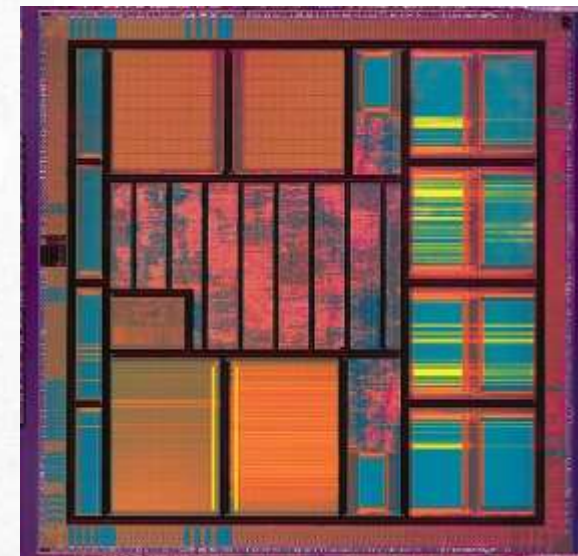
- o mai mica arie/volum; echipamente mai compacte;
- putere consumata mai mica;
- nevoia unor teste mai reduse la nivelul sistemului;
- fiabilitate înalta, în principal datorita perfectionarii interconexiunilor în interiorul circuitului;
- viteza crescuta, in conditiile reducerii semnificante a lungimii interconexiunilor
- scaderea însemnata a costurilor.



*Figura 1.2: Evolutiile densitatii de integrei si a dimensiunii caracteristice minime, asa cum erau vazuea la începutul anilor 1980*



Early - 2300 transistors



Current chips has 1B transistors

Figura 1.3: Nivelul integrării de-a lungul timpului pentru circuitele de memorie și circuitele logice

# Presiuni mai puternice pe piață

- Fereastra de proiectare în scădere  
*Decreasing design window*
- Toleranță mai mică pentru reviziile de proiectare  
*Less tolerance for design revisions*



Exponentially more complex, greater design risk, greater variety, and a smaller design window!

## 1.2 Procesul de proiectare VLSI

Procesul proiectării, la diverse niveluri, are de obicei un caracter evolutiv. Începe cu un **set dat de specificații**.

Proiectul inițial este dezvoltat și testat în conformitate cu specificațiile. Când specificațiile nu sunt îndeplinite, proiectul trebuie îmbunătățit. Dacă aceste îmbunătățiri sunt imposibile sau prea costisitoare, atunci trebuie luate în considerare revizuirea specificațiilor și analiza impactului prin această revizuire.

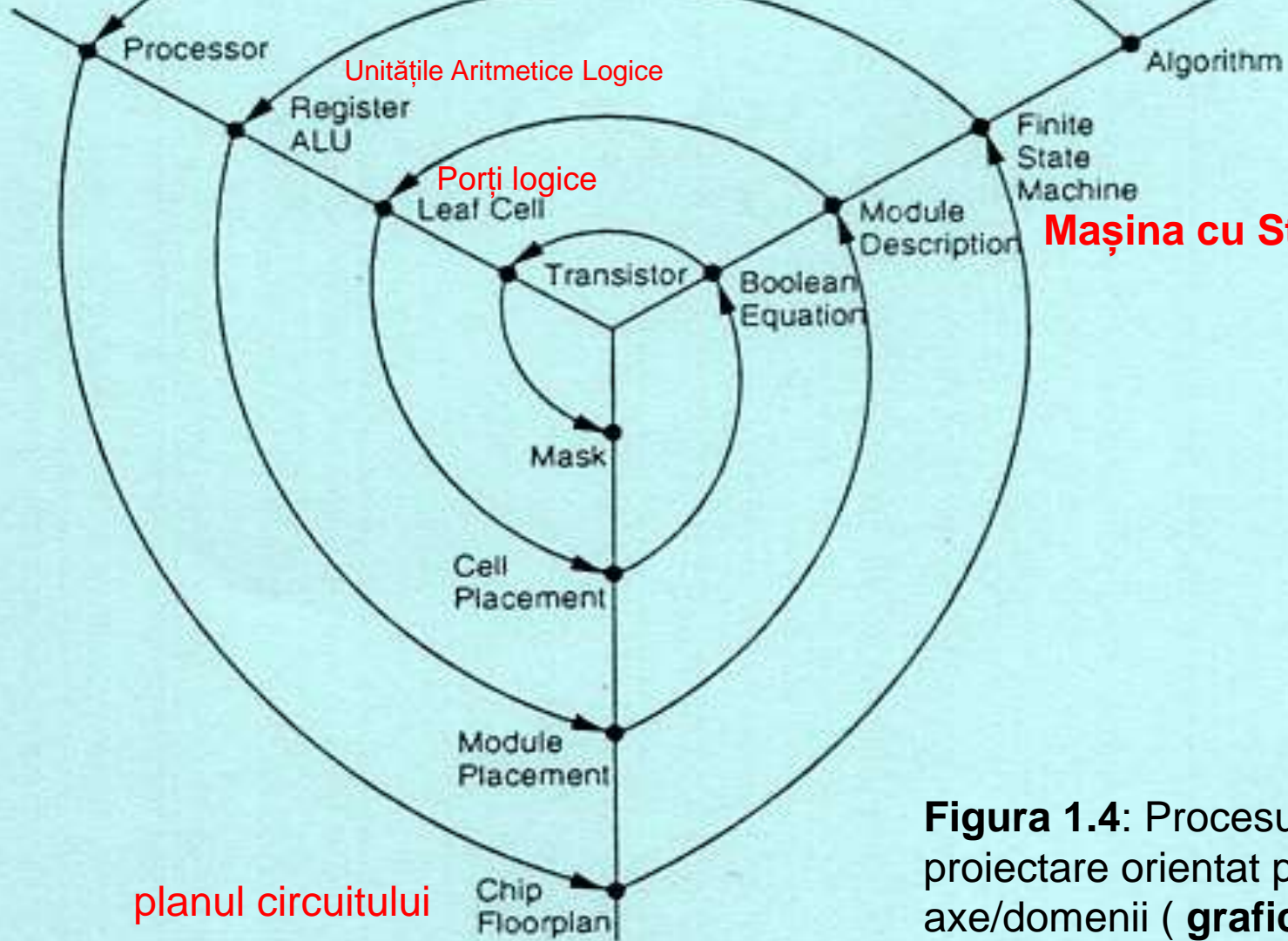
Graficul Y din **figura 1.4**, introdus de către D. Gajski, arată procesul de proiectare pentru cele mai multe dintre circuitele logice, utilizând activități de proiectare orientate pe trei axe /domenii diferite.

domeniul structural

domeniul comportamental

Structural Domain

Behavioral Domain



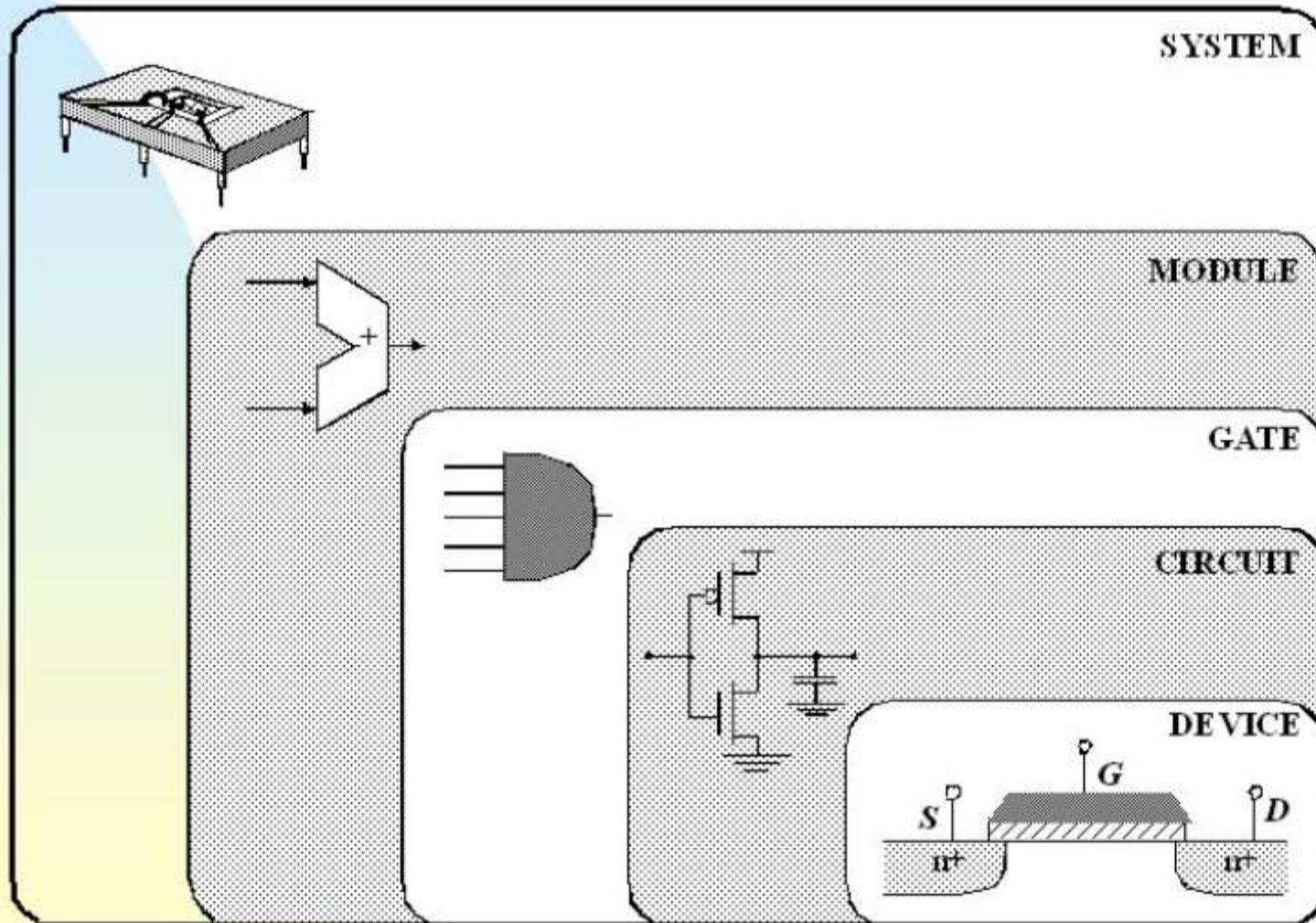
planul circuitului

Geometrical Layout Domain

domeniul machetării geometrice

Figura 1.4: Procesul tipic de proiectare orientat pe trei axe/domenii (graficul Y)

# Design Abstraction Levels

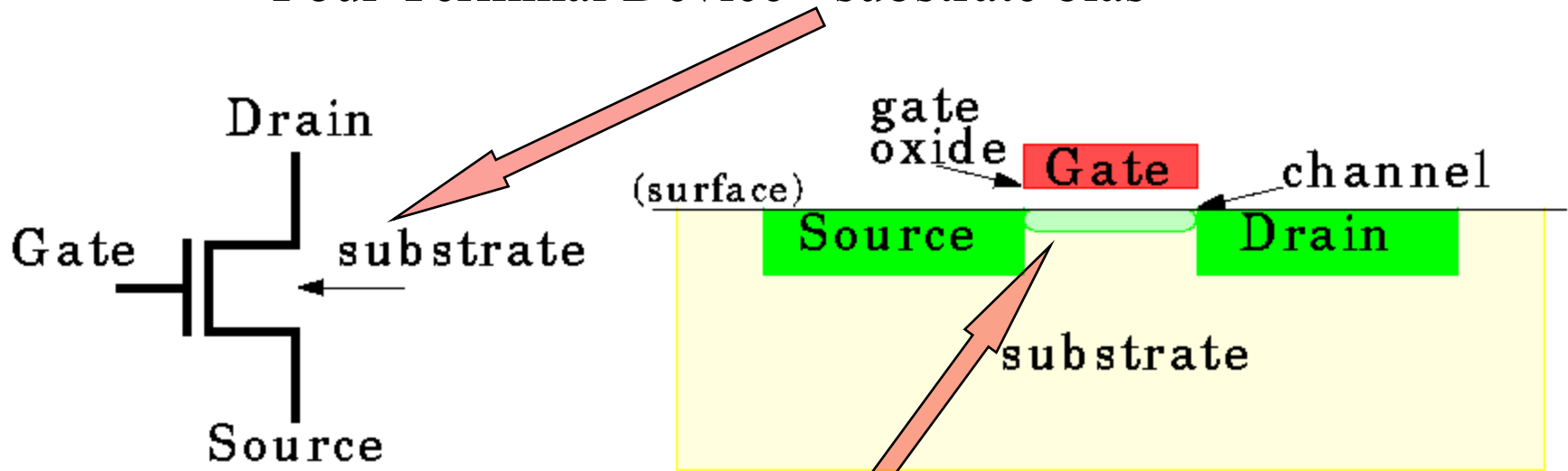


[Adapted from <http://infopad.eecs.berkeley.edu/~icdesign/>. Copyright 1996 UCB]

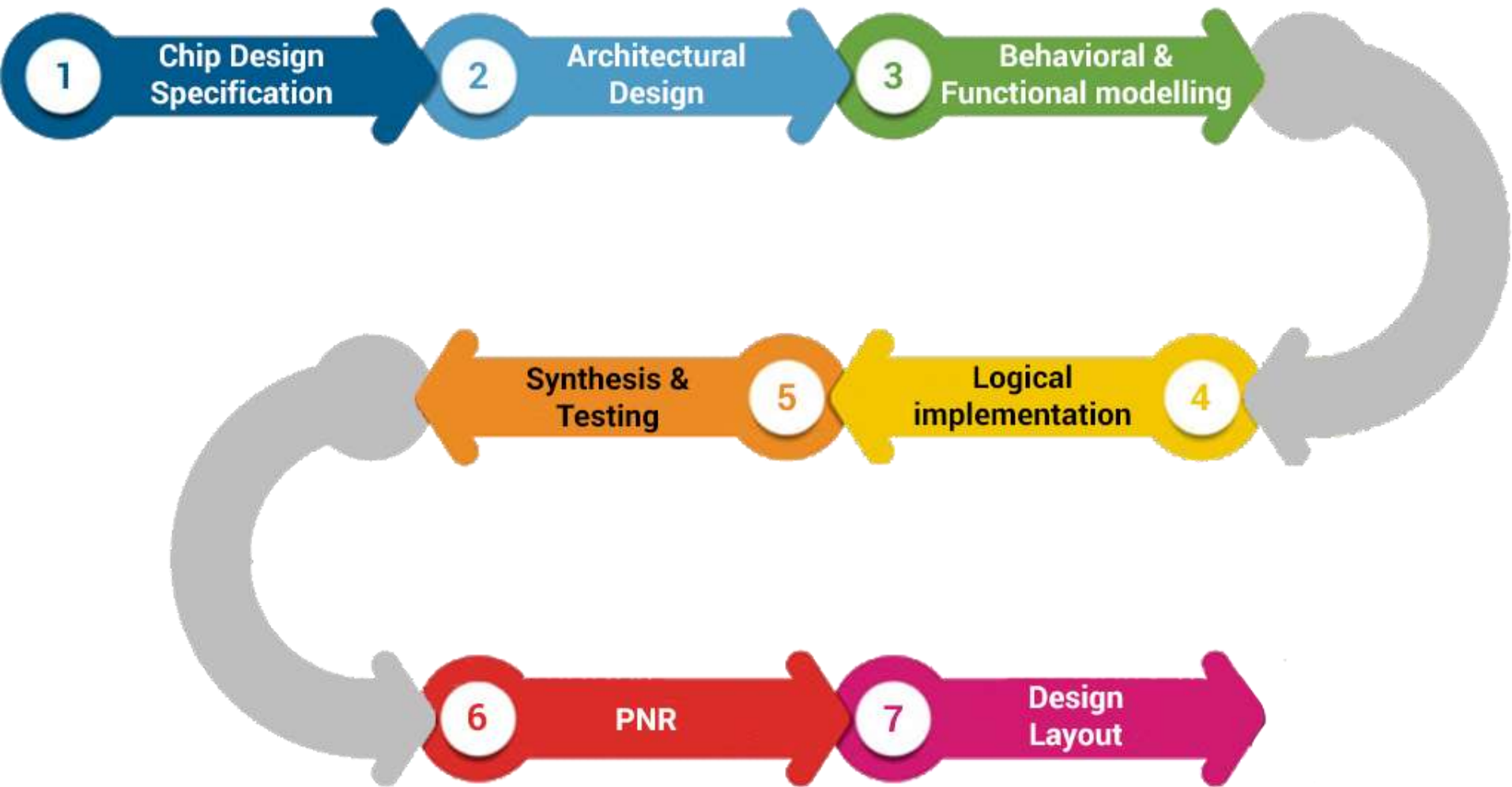


# N-Channel Enhancement mode MOS FET

- Four Terminal Device - substrate bias



-The "self aligned gate" - key to CMOS

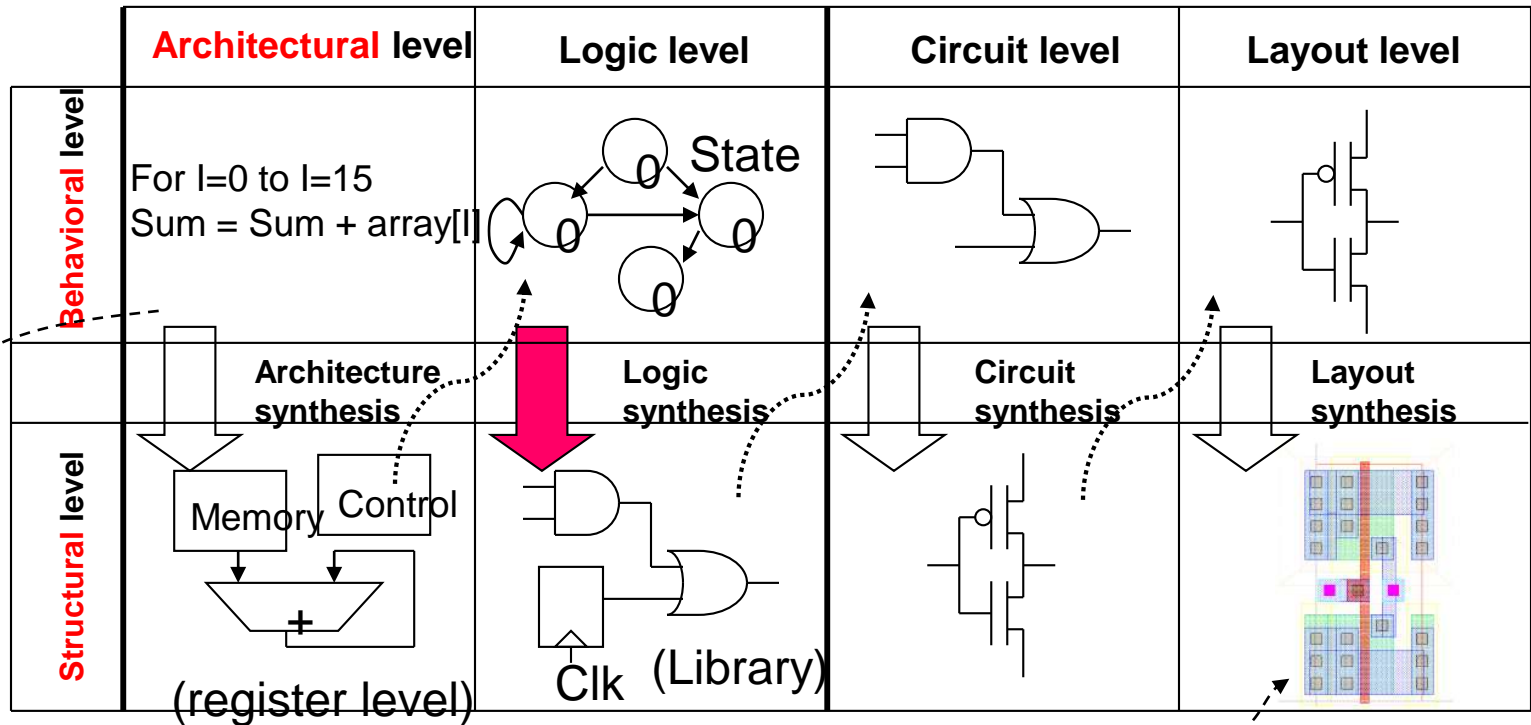


**Graficul Y** consta in trei domenii majore:

- **domeniul comportamental**;
- **domeniul structural**;
- **domeniul machetarii geometrice**;

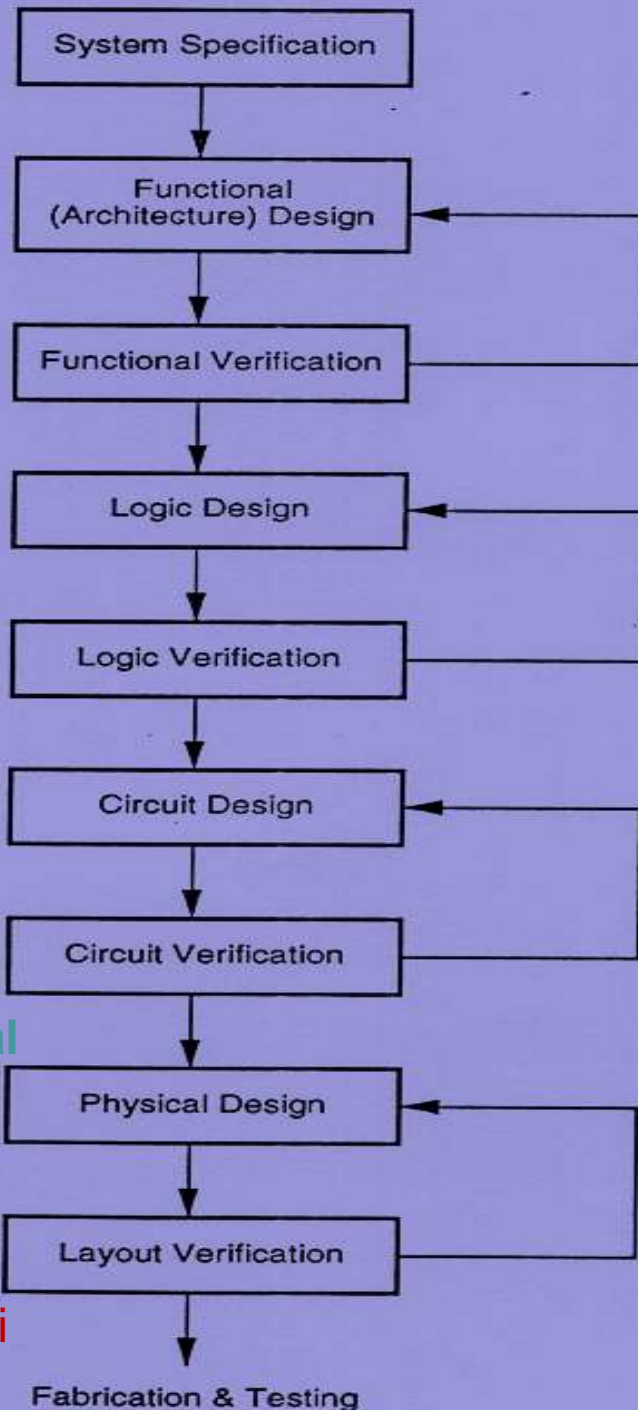
Procesul de proiectare porneste de la algoritmul care descrie comportarea dorita a circuitului. In primul rând este definita arhitectura corespunzatoare pentru procesor. Aceasta arhitectura este mapata pe suprafata circuitului prin stabilirea planului circuitului (floorplanning). Urmatoarea etapă de proiectare, în domeniul **comportamental**, definește **Masina cu Stari Finite (FMS – Finite State Machine)**. Implementarea structurala se realizeaza cu module functionale cum ar fi Registrele si Unitatile Aritmetice Logice (ALU). Aceste module sunt plasate geometric pe suprafata pastilei, folosind unelte CAD pentru amplasarea automata a modulelor, urmata de stabilirea traseelor de legatura intre module (routing), in scopul minimizarii ariei ocupate de catre interconexiuni si minimizarii întârzierilor semnalelor. A treia etapa începe cu descrierea comportamentala a modulului. Module individuale sunt apoi implementate cu porti logice (leaf cells). La aceasta etapa, circuitul este descris in termeni de porti logice, care pot fi plasate si interconectate folosind un program pentru amplasarea celulelor si trasarea conexiunilor dintre celule. Ultima etapa implica o descriere detaliata a portilor logice urmate de o implementare la nivel de tranzistor a portilor logice si de generarea mastii. În proiectul bazat pe celule standard, portile logice sunt deja proiectate si stocate într-o biblioteca.

# Abstraction levels and synthesis



Silicon compilation (not a big success)

# VLSI DESIGN FLOW



domeniul  
comportamental

Behavioral  
Representation

Logic  
(Gate-level)  
Representation

Circuit  
Representation

domeniul structural

Layout  
Representation

domeniul machetării  
geometrice

**Figura 1.5:** O viziune simplificata a procesului de proiectare VLSI

Figura 1.5 prezinta o versiune simplificata a procesului de proiectare VLSI, luând în considerare diversele reprezentari sau abstractizari ale acestui proces: comportamentala, logica, de machetare a circuitelor si a mastii. Trebuie remarcat faptul ca verificarea proiectului joaca un rol foarte important în fiecare etapa a acestui proces. Esecul verificarii complete a proiectului în etapele de început provoaca pierderi însemnate si costisitoare pentru reproiectare în etapele ulterioare, ceea ce duce la cresterea timpului în care produsul ajunge pe piata. Cu toate ca procesul de proiectare a fost descris, pentru simplitate, într-o maniera liniara în realitate exista multe iteratii intr-un sens sau altul, în special între doua etape succesive, si, ocazional, între etape mai distantate.

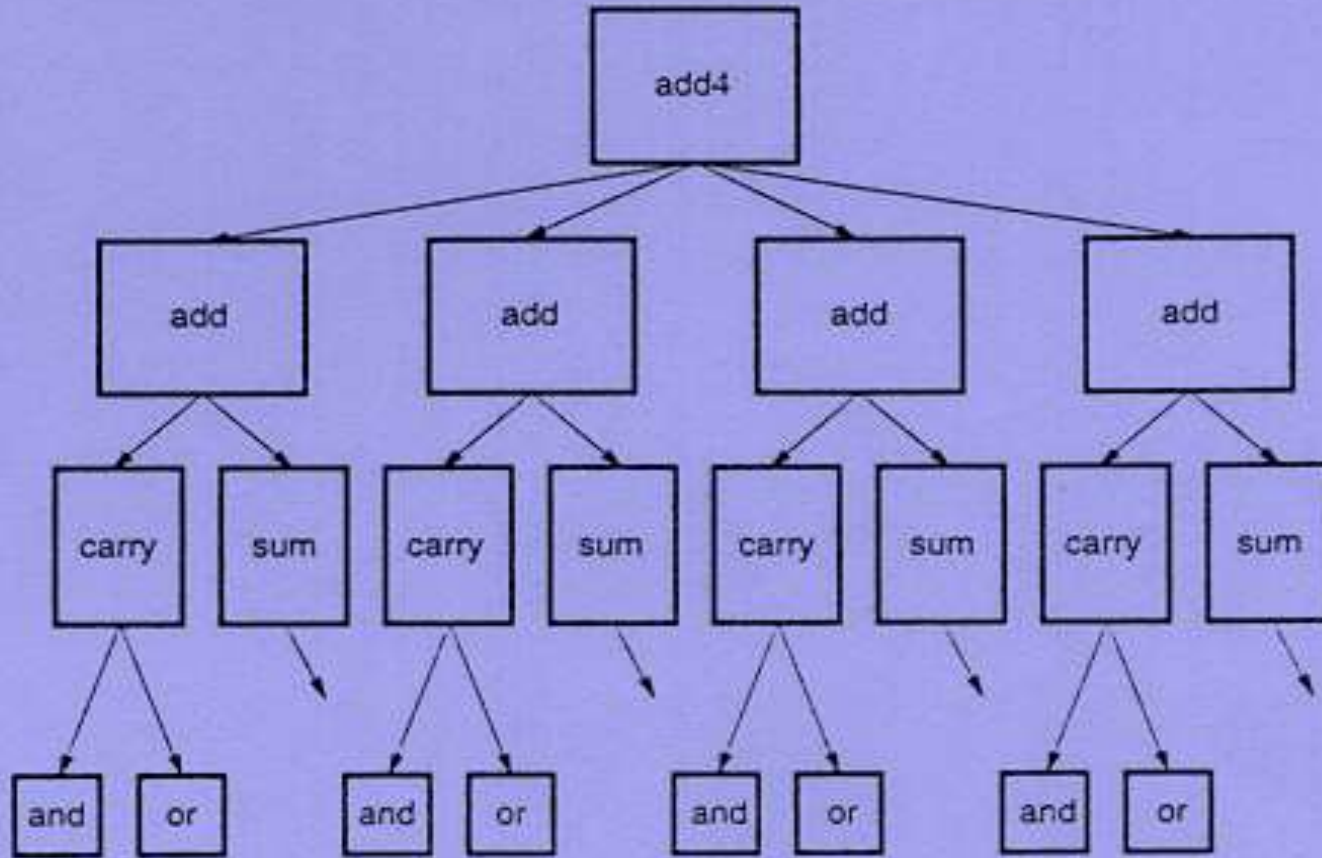
## 1.3 Ierarhia Proiectarii.

*dezbină și cucerește*

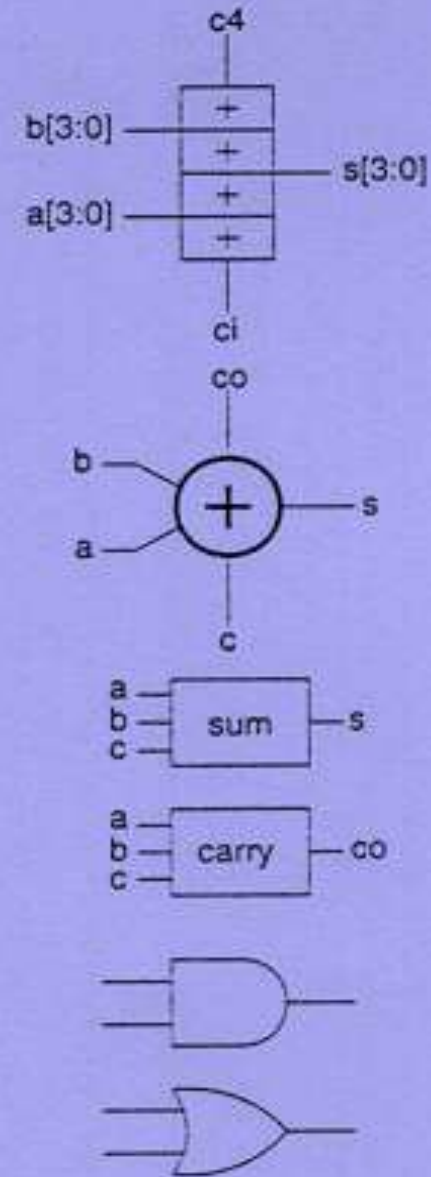
*разделить и победить*

Abordarea ierarhica/tehnica „**divide and conquer**” implica împartirea unui modul în submodule și apoi repetarea acestei operații pentru fiecare submodule până când complexitatea celor mai mici părți devine ușor de rezolvat. Această abordare este foarte asemănătoare cu cea din *software în care, programele mari sunt împartite în secțiuni* din ce în ce mai *mici până la subrutine simple*, cu funcții și interfețe bine definite, care pot fi ușor de implementat.

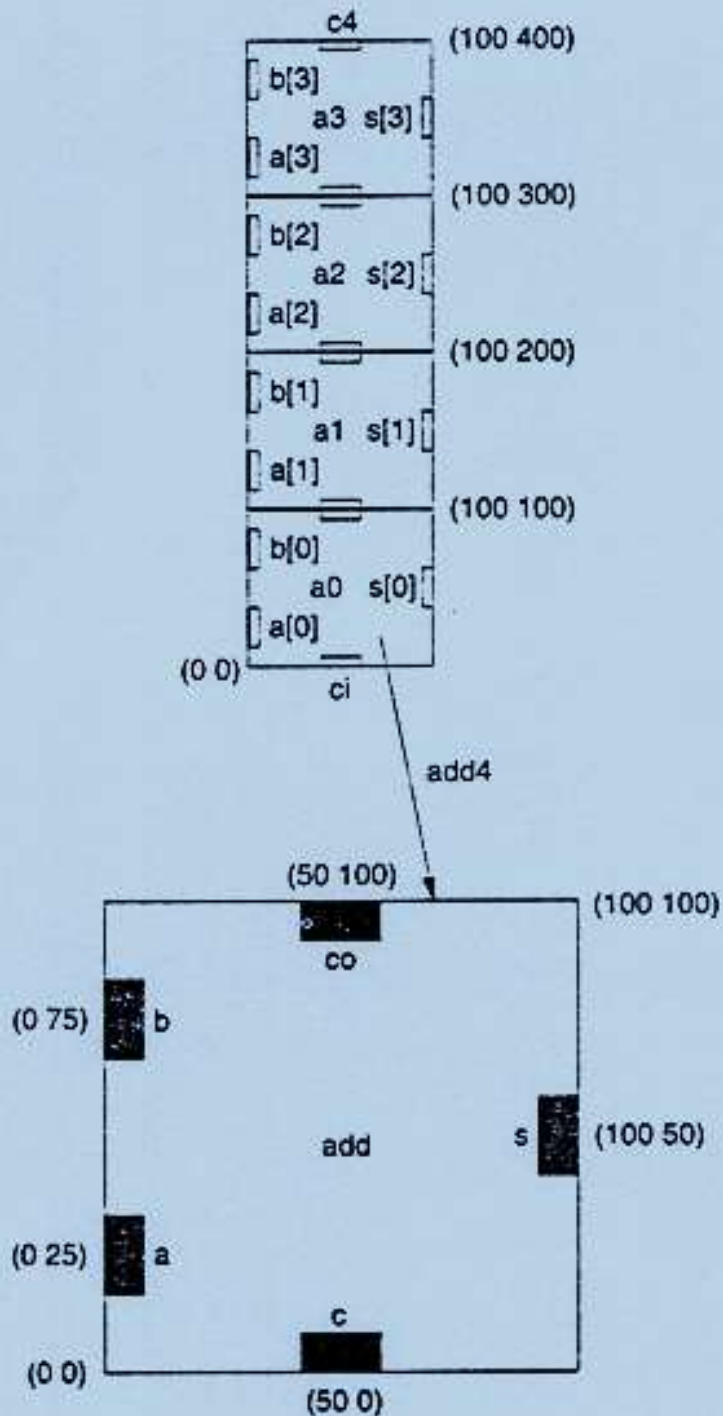
Ca un **exemplu de ierarhie structurala**, figura 1.6 arata descompunerea structurala a unui sumator pe 4 biti.



(a)



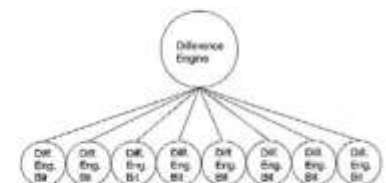
**Figura 1.6:** Descompunerea structurala a unui circuit sumator pe patru biti, cu reprezentarea ierarhiei pana la nivelul portilor logice



**Figura 1.7:** Descompunerea ierarhica a unui circuit sumator pe patru biti în domeniul descrierii fizice (geometrice sau domeniul machetarii geometrice)

### exemplu de ierarhie fizică

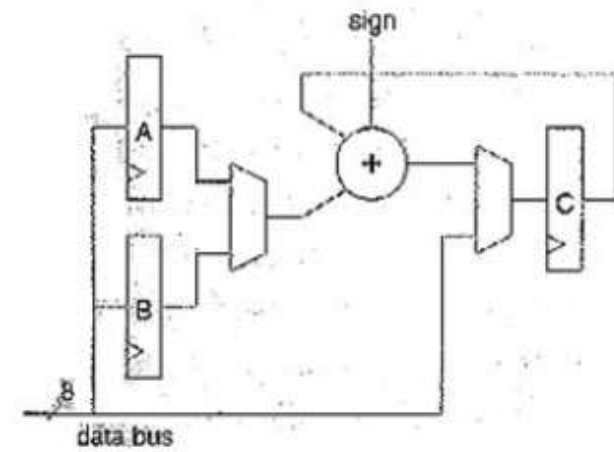
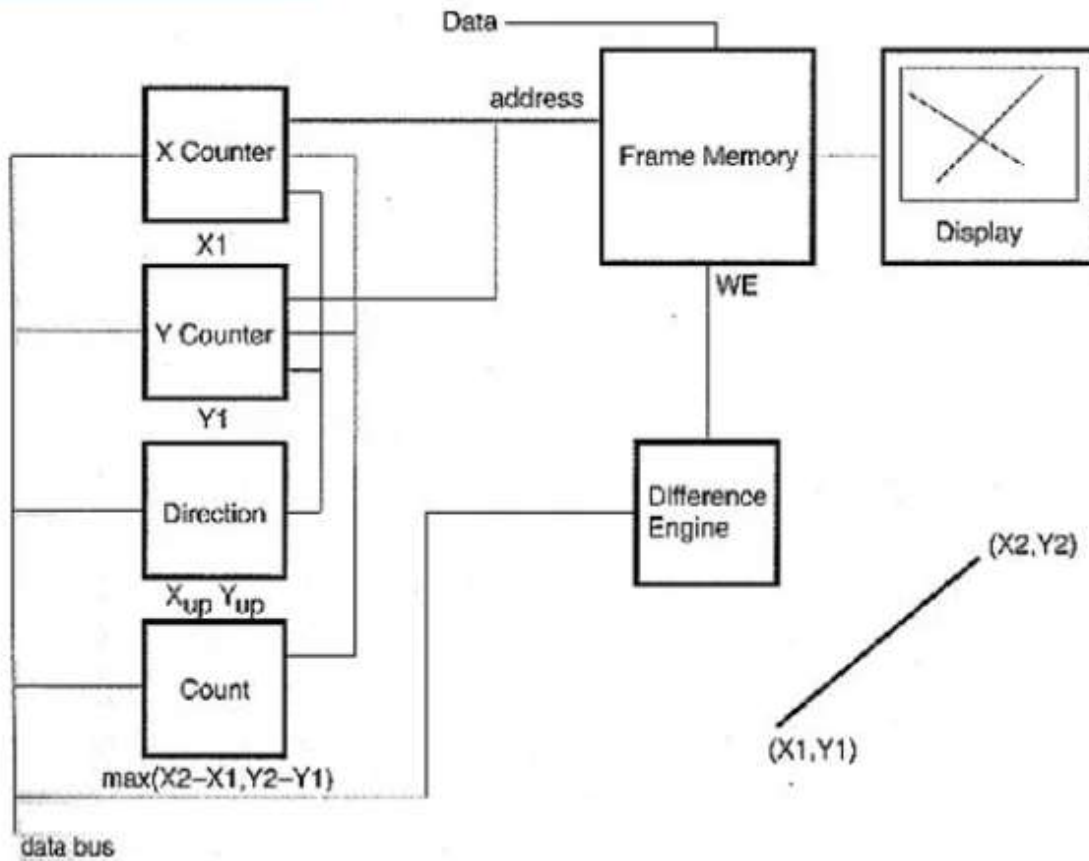
În domeniul fizic, partitionarea unui sistem complex în diverse blocuri functionale va asigura un ghid valoros pentru realizarea acestor blocuri ca circuite. În mod evident, dimensiunea (aria) si forma aproximativa a fiecaruia dintre submodule ar trebui estimate pentru a se putea crea o masca utilizabila.

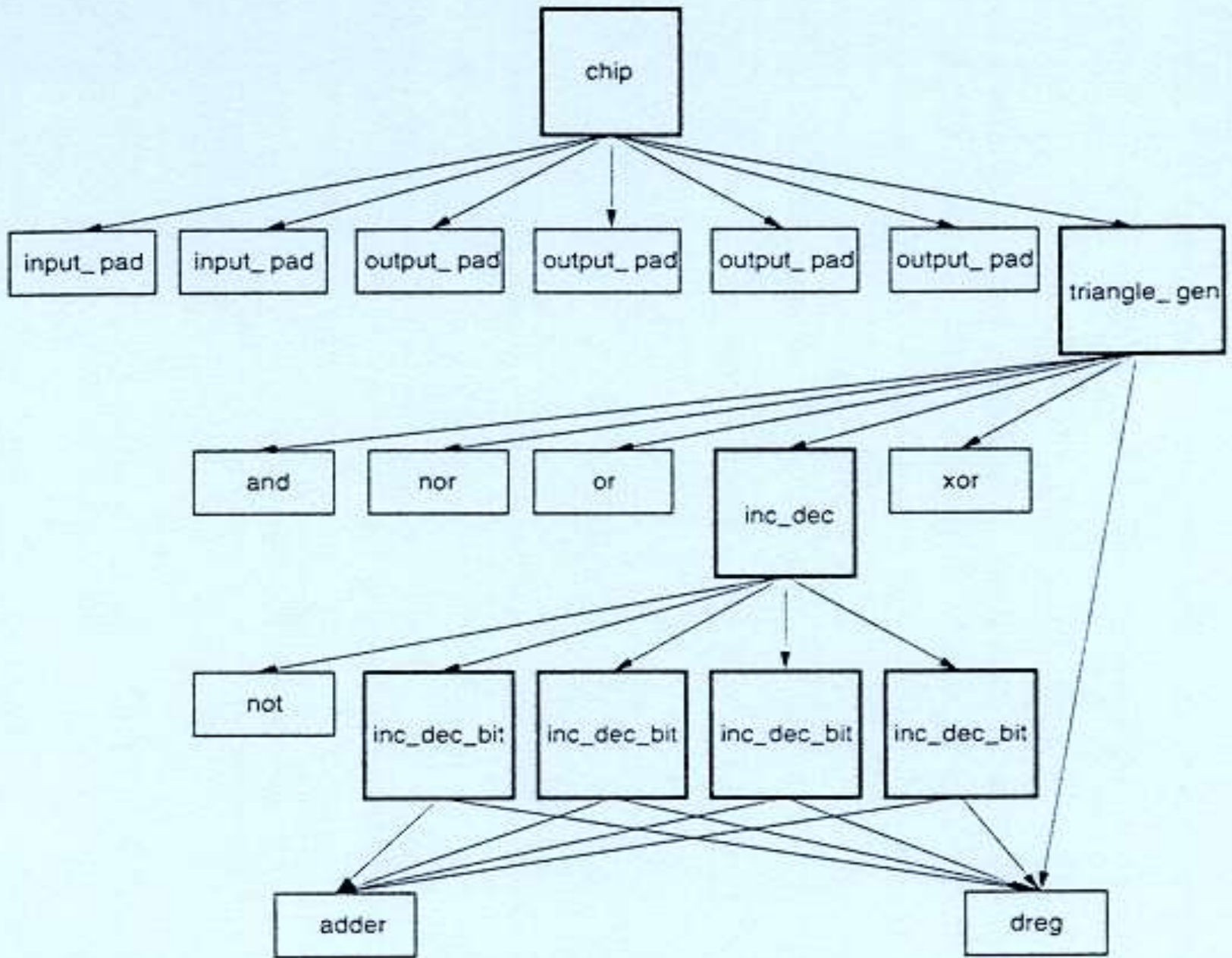




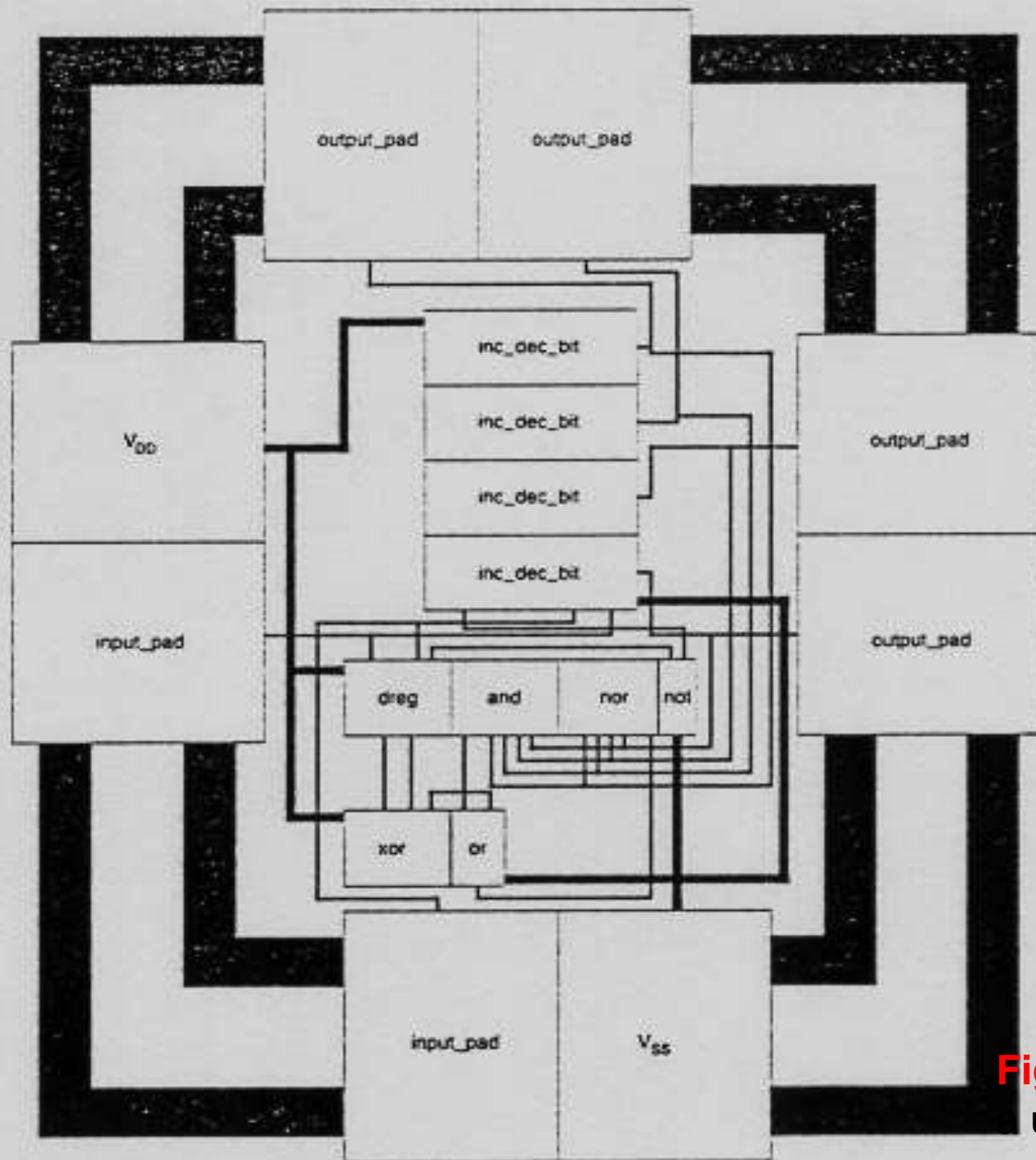
# Figura

## exemplu de ierarhie structurală





**Figura 1.9:** Ierarhia structurala a unui circuit generator de semnal triunghiular.



**Figura-1.10:** Macheta fizica unui circuit generator de semnal triunghiular.

## 1.4 Conceptele de regularitate, modularitate si localitate.

Abordarea ierarhica a proiectului reduce complexitatea acestuia prin descompunerea unui sistem complex în mai multe submodule. In mod curent celelalte concepte si abordari de proiectare sunt, in continuare, necesare pentru simplificarea acestui proces.

**Regularitatea** înseamna ca **descompunerea structurala a unui sistem complex** trebuie sa furnizeze ca rezultat nu numai într-un numar de blocuri simple, dar si, pe cât posibil, blocuri similare.

**Un exemplu bun de regularitate** este proiectul structurilor matriciale constând in **celule identice**, cum ar fi matricile de multiplicare paralela.

**Regularitatea** poate exista la toate nivelurile de abstractizare:

- la nivelul tranzistorilor (**tranzistorii cu dimensiuna similara simplifica proiectul**),
- la nivelul logic (**pot fi folosite structuri de porti logice identice**) etc



Figura 1.11 arata proiectul curent al unui multiplexor 2:1, un registru cu bistabile de tip D cu comanda pe front si al unui sumator pe un bit.

Trebuie remarcat ca toate aceste circuite au fost construite folosind numai inversoare si tampoane cu trei stari.

Daca proiectantul dispune de o mica biblioteca cu blocuri de baza bine definite si bine caracterizate, pot fi construite, utilizând acest principiu, un numar mare de functii diferite.

**Regularitatea,**

de obicei, reduce numarul de module diferite, care trebuie sa fie proiectate si verificate, la toate nivelurile de abstractizare.

## Exemple de regularitate

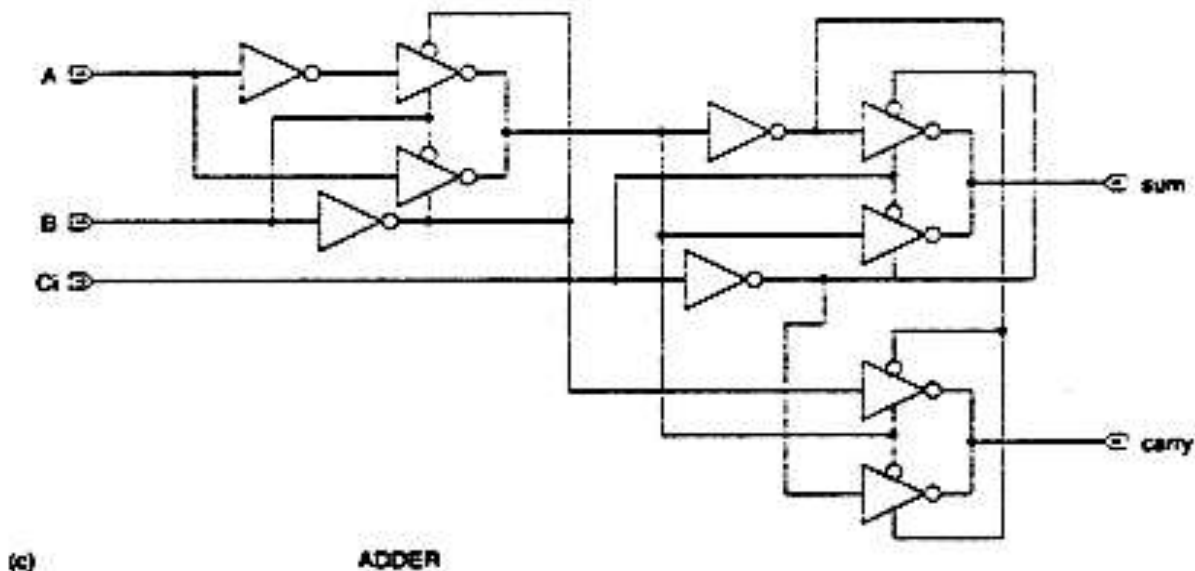
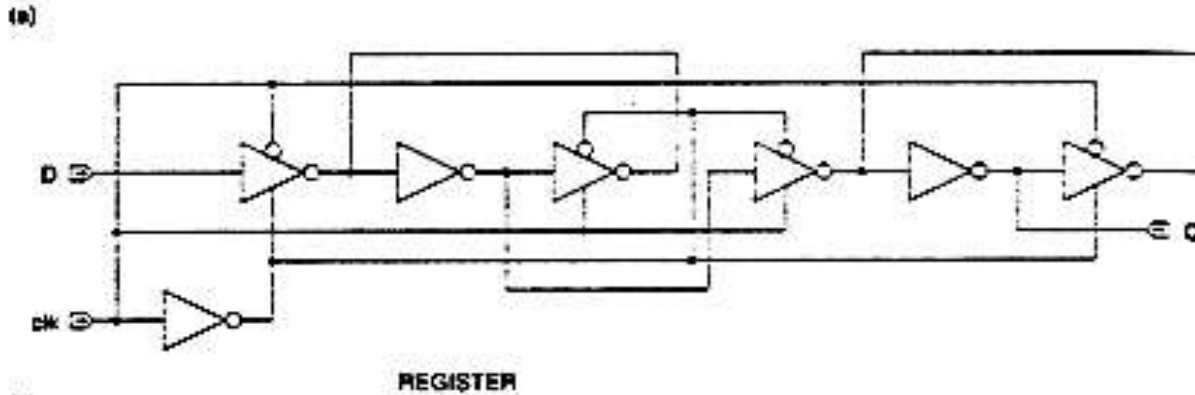
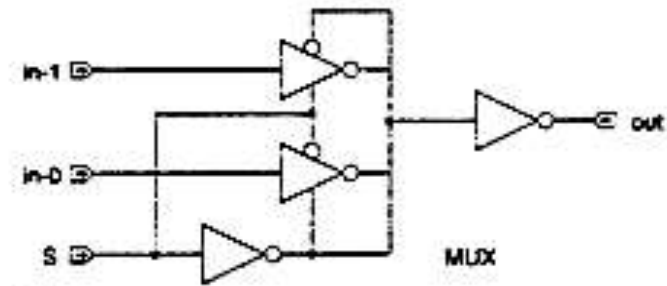


Figura 1.11: Schemele uzuale ale:

unui multiplexor 2:1,

unui registru si

unui sumator,

bazate pe inversoare si tampoane cu trei stari.

**Modularitatea** în proiectare înseamnă ca diversele blocuri functionale, care compun un sistem mare, trebuie să aibă **funcții și interfețe bine definite**.

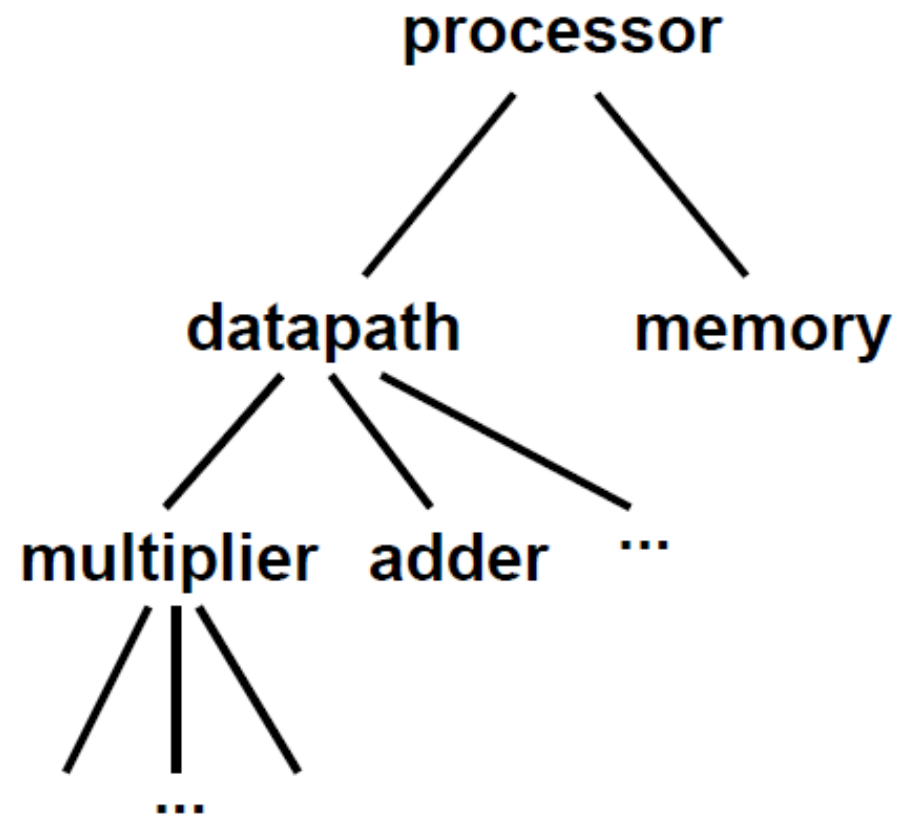
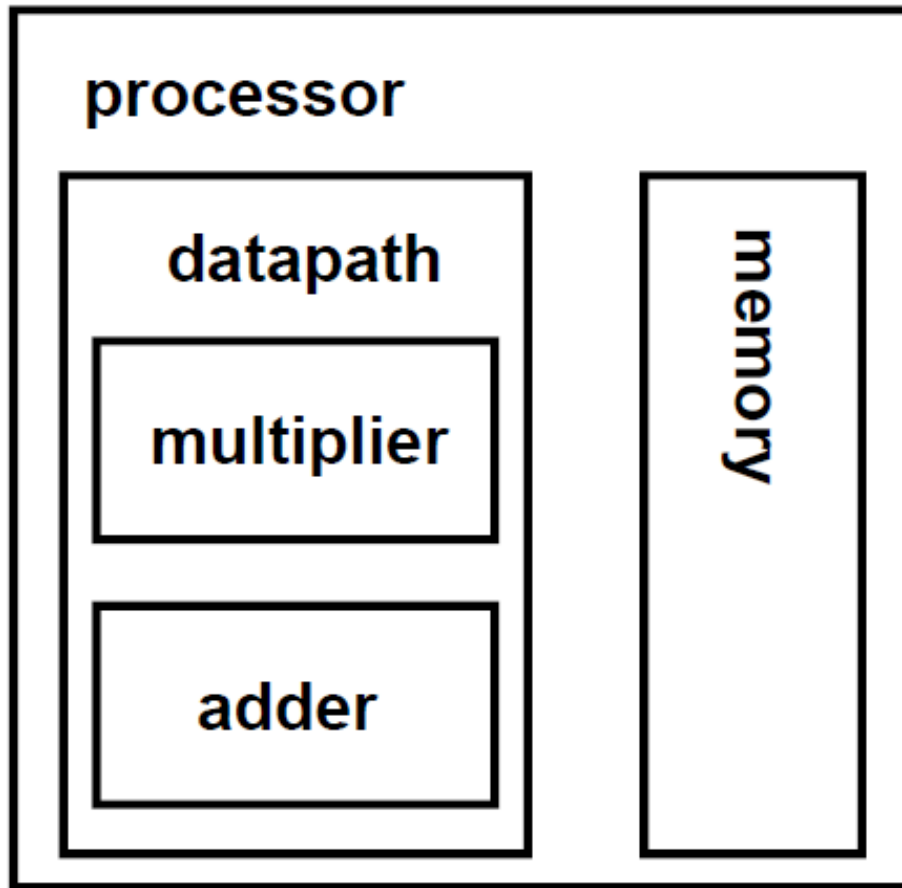
Modularitatea permite ca fiecare bloc sau modul să fie proiectat relativ independent de celelalte, din moment ce nu există nici o ambiguitate legată de vreo funcție sau semnal ale acestor blocuri. Toate aceste blocuri pot fi combinate cu ușurință la sfârșitul procesului de proiectare, pentru a forma sistemul dorit.

**Conceptul de modularitate** face posibil **paralelismul** în procesul de proiectare. Permite, de asemenea, utilizarea modulelor generice în numeroase proiecte; Funcțiile bine definite și interfatarea semnalelor permit proiectarea pe baza conceptului „**plug and play**”.

Definind interfețe bine caracterizate pentru fiecare modul din sistem, se realizează premisa ca mecanismul intern al fiecăruia dintre acestea să devină neimportant pentru „lumea externă” a modului. Detaliile interne rămân la nivelul local.

# Hierarchy (Modularity)

Hierarchy: Something is composed of simpler things

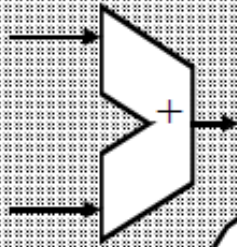




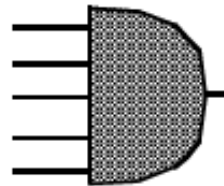
# Design Levels



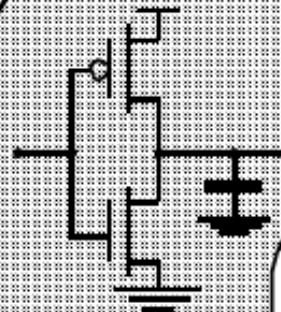
SYSTEM



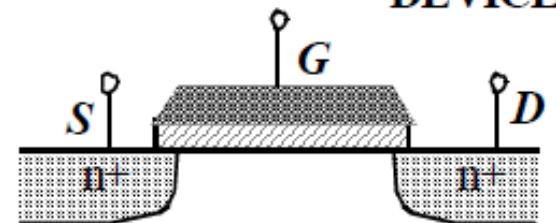
MODULE



GATE

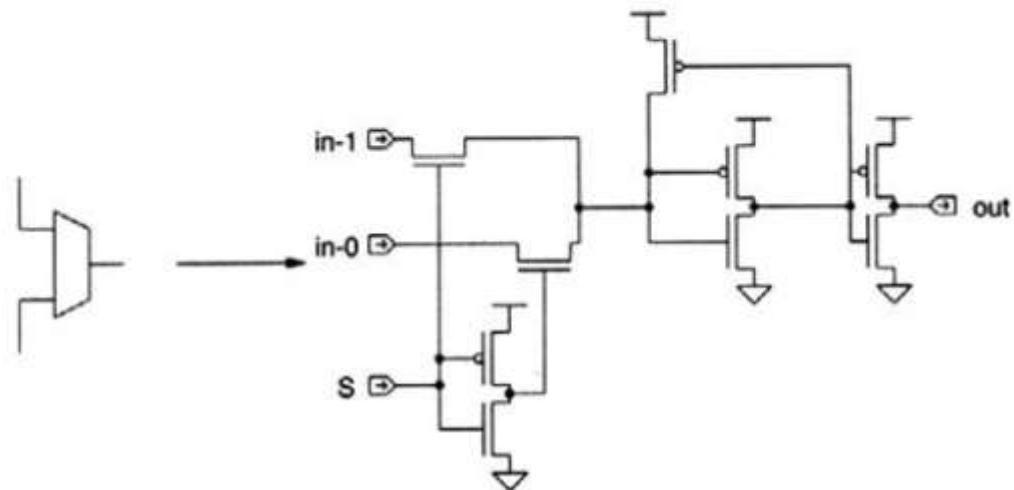


CIRCUIT

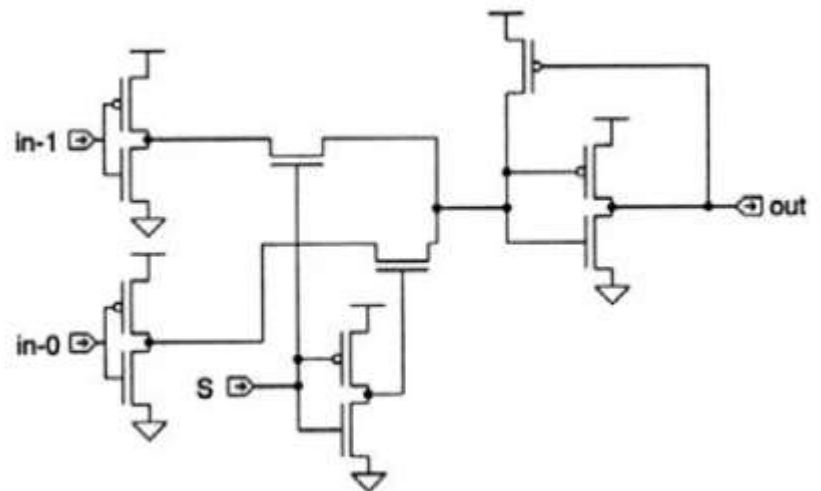


DEVICE

# Example of Poor Modularity



(a)



(b)

An example of poor modularity: (a) mux with transmission-gate inputs; (b) a solution—a buffered mux

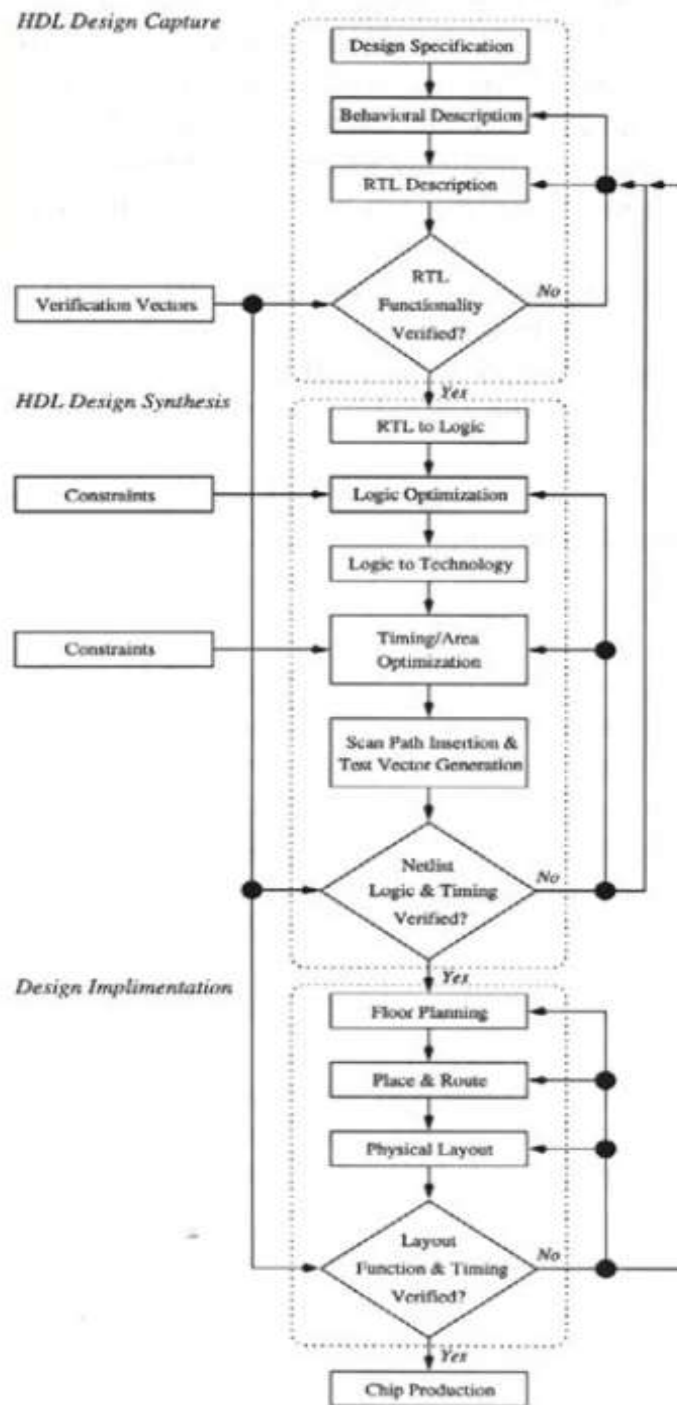
## Conceptul de **localitate**

asigura ca legaturile/conexiunile sa se faca in special între module vecine, **evitând, pe cât este posibil conexiunile la distanta mare.**

Ultima idee este **extrem de importanta pentru evitarea întârzierilor excesive pe interconexiuni.**

Operatiile critice, din punctul de vedere al timpului, ar trebui sa fie executate local, fara a avea nevoie de accesul la unele module sau semnale aflate la distanta. Daca este necesar, replicarea unor portiuni ale logicii poate rezolva aceasta problema în arhitecturile de sisteme mari.

# Typical VLSI Design Flow



## 1.5 Stiluri de proiectare VLSI

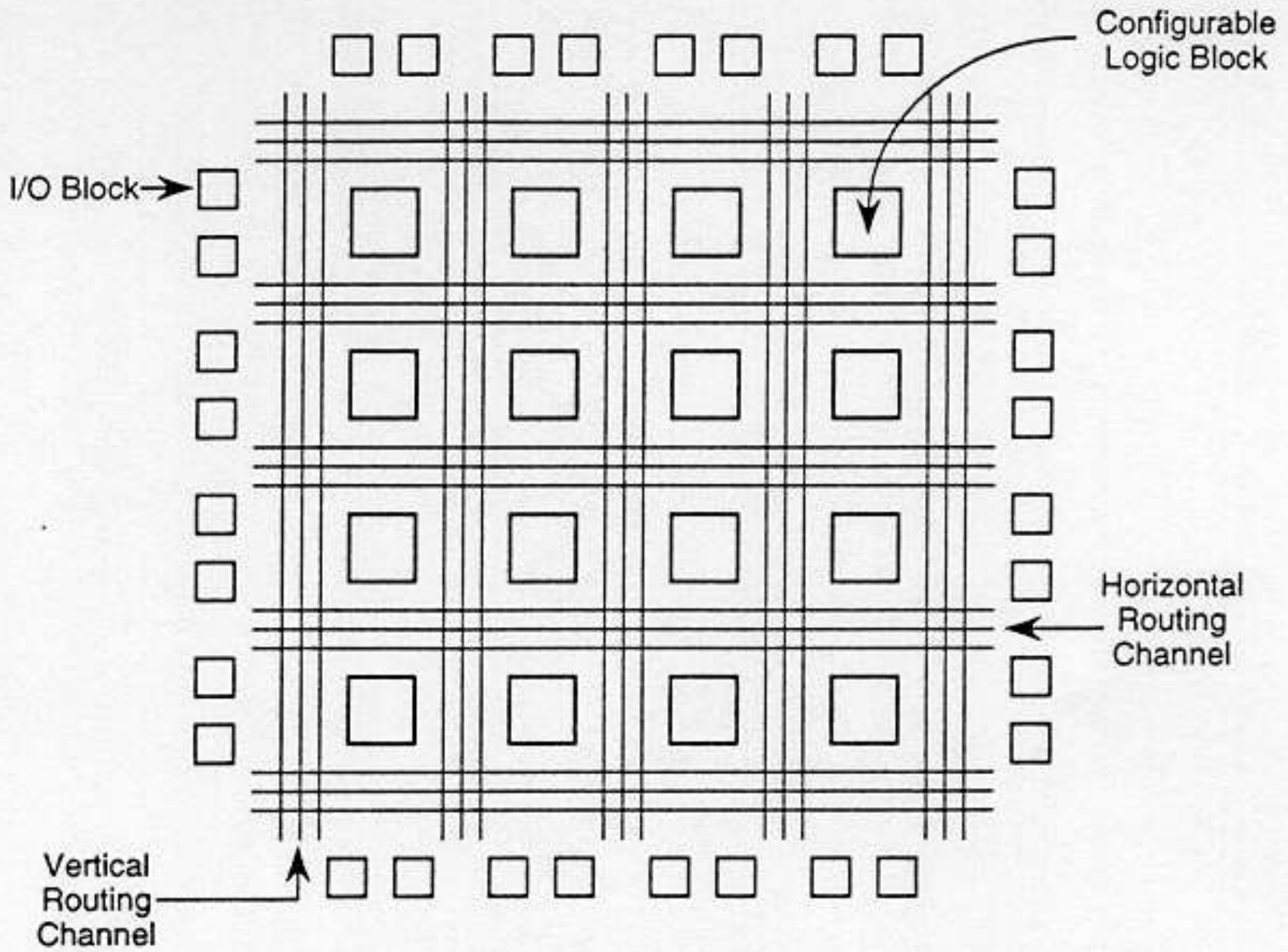
Pentru implementarea unui algoritm specific sau a unei functii logice pot fi considerate mai multe stiluri de proiectare. Fiecare stil are propriile sale merite si deficiente si, de aceea, proiectantul trebuie sa faca o selectie adecvata pentru a asigura functionalitatea dorita si un cost scazut.

# Design styles

- Full custom - Personalizat complet
- **Semi-custom design**
- Standard cell - Celula standard
- Gate-array - Poarta-matrice
- IP-blocks - Macro-cell
- “FPGA”
- Combinations - Combinații

**CAD is the key**

The design time is **critically** dependent on the quality and features of the CAD set available. The drive to reduce design time has now made it possible to design systems in **months** (if not weeks) which only five years ago would have taken **tens of man years**.



**Figura 1.12:** Arhitectura generala a unui FPGA Xilinx

## 1.5.1 FPGA (Field Programmable Gate Array)

Circuitele FPGA, care contin mii si zeci de mii de porti logice, cu interconexiuni programabile, sunt programate de catre utilizator. Stilul de proiectare cu aceste circuite se preteaza la prototipuri, care trebuie implementate în timp scurt, si la circuite realizate in cadrul unui buget redus, folosite în special pentru aplicatiile pe scara redusa. Un circuit FPGA uzual consta in tampoane de intrare/iesire, o matrice de blocuri logice configurabile (CLB – Configurable Logic Blocks) si structuri programabile de interconectare. Programarea interconexiunilor este implementata prin programarea celulelor RAM ale caror terminale de iesire sunt conectate la portile tranzistorilor de tip MOS. O arhitectura generala de FPGA, de la XILINX, este prezentata în **figura 1.12**.



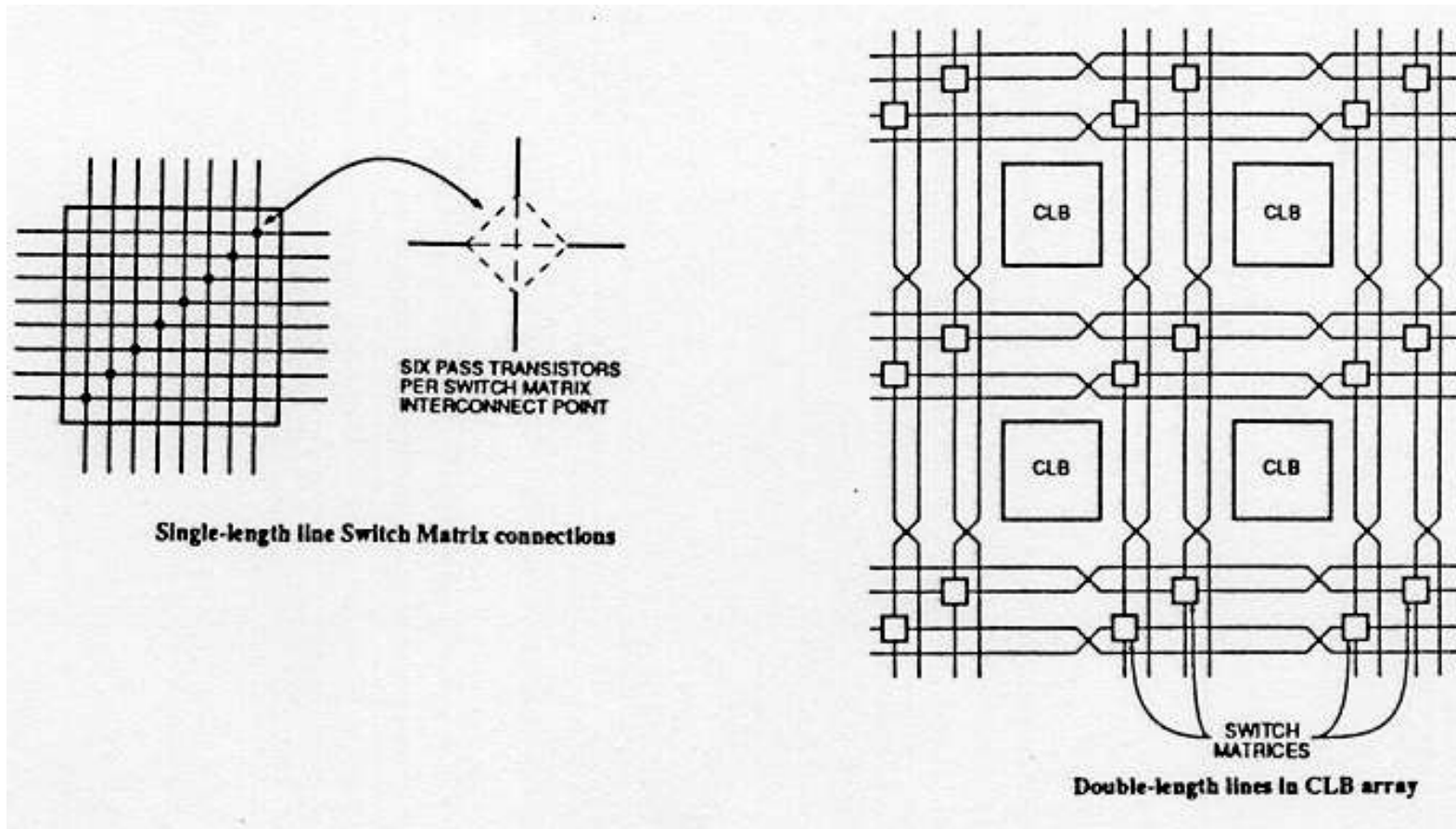
Un CLB simplu (model XC2000 de la XILINX) este aratat în figura 1.14. El consta in patru terminale pentru semnalele de intrare (A, B, C, D), un terminal pentru semnalul de ceas, multiplexoare programabile de catre utilizator, un comutator de tip latch SR, si o tabela de asociere (LUT – Look-Up Table). Tabela de asociere este o memorie digitala care retine tabela de adevar a functiei booleene. Prin urmare, acesta poate genera orice functie de pâna la 4 variabile sau oricare doua functii de trei variabile. Terminalele de control pentru multiplexare nu sunt prezentate explicit în figura 1.14.

**CLB-ul este configurat astfel încât o mare varietate de functii logice poate fi realizata prin programarea matricii.**

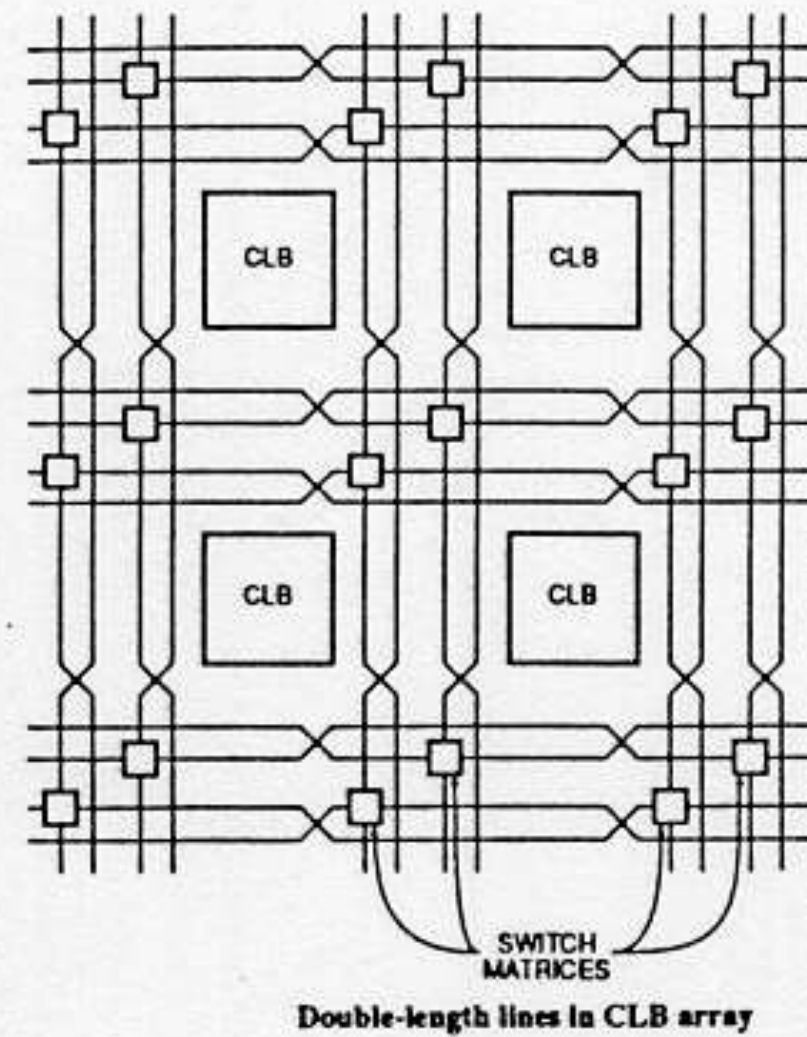
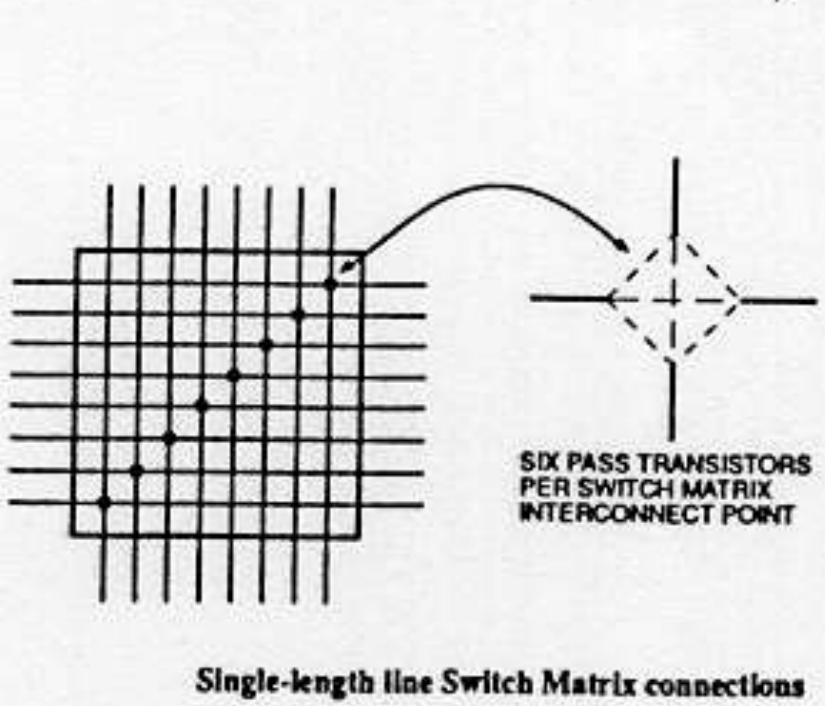
Pentru a mapa functii complexe, au fost introduse circuite mai sofisticate.

Etapele tipice ale procesului de proiectare pentru un circuit FPGA încep cu descrierea lui comportamentala, folosind un limbaj de descriere hardware cum ar fi Verilog sau VHDL. Arhitectura sintetizata este mapata (partitionata) tehnologic în circuite sau celule logice. Urmeaza apoi plasamentul si gasirea traseelor, care conecteaza celule logice individuale la blocurile CLB ale FPGA si determina mastile traseelor de-a lungul celulelor, în concordanta cu lista nodurilor.

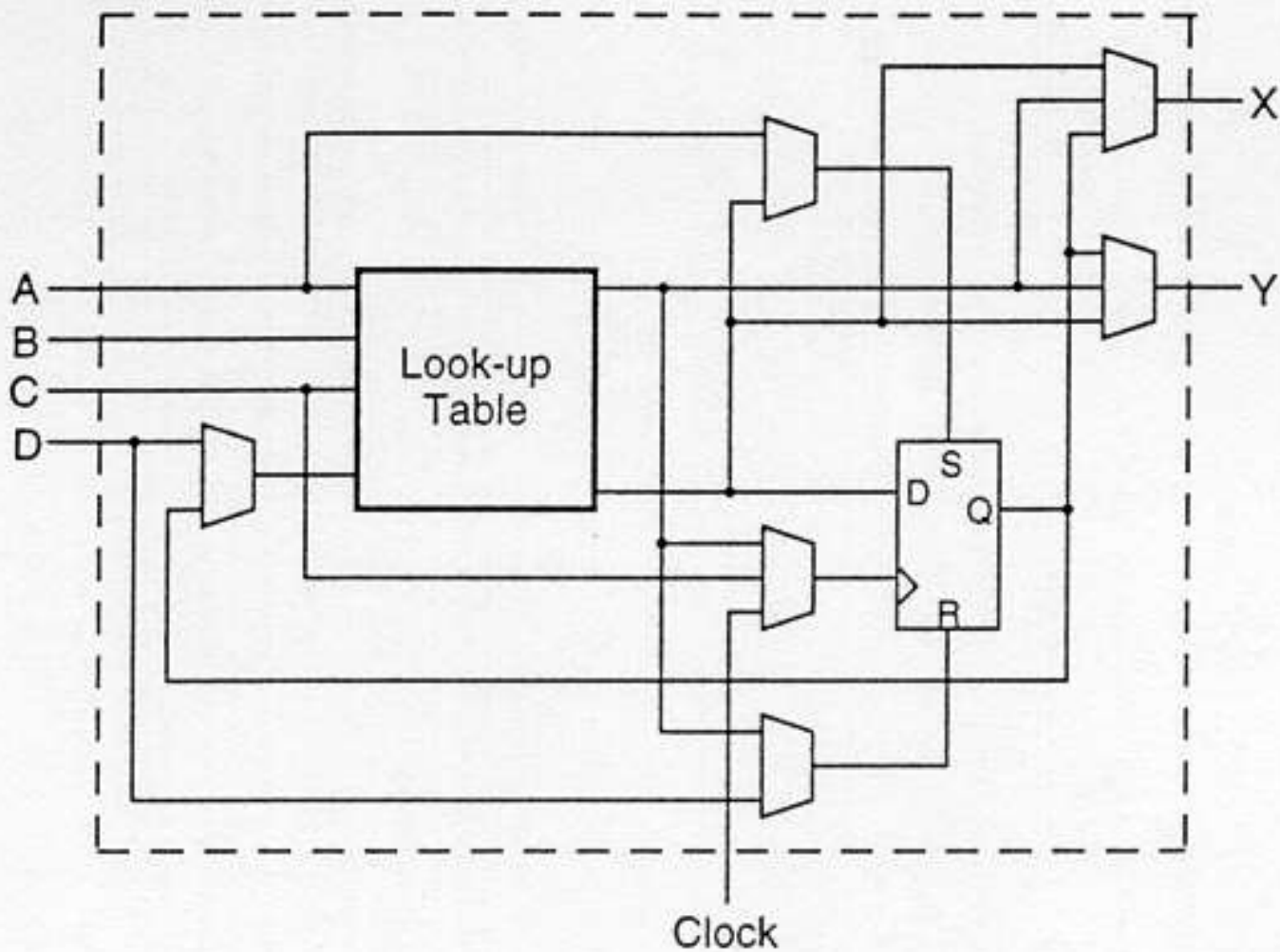
Dupa stabilirea acestor trasee, performanta proiectului poate fi simulata si verificata, înainte de programarea fizica a circuitului FPGA, prin descarcarea fisierului proiectului, in formatul corespunzator, in circuitul fizic FPGA. Programarea circuitului ramâne valida atâta timp cât circuitul este sub tensiune, sau pâna in momentul în care este facuta o noua (re)programare. În majoritatea cazurilor, nu este posibila utilizarea completa a circuitului, multe blocuri de celule pot ramâne neutilizate.



O vedere mai detaliata incluzând si locatiile comutatoarelor matricelor utilizate pentru traseele interconexiunilor este data în **figura 1.13**.



**Figura 1.13:** Vedere detaliata a comutatoarelor matricelor si interconexiunilor dintre CLB-uri



 = User-programmed Multiplexer

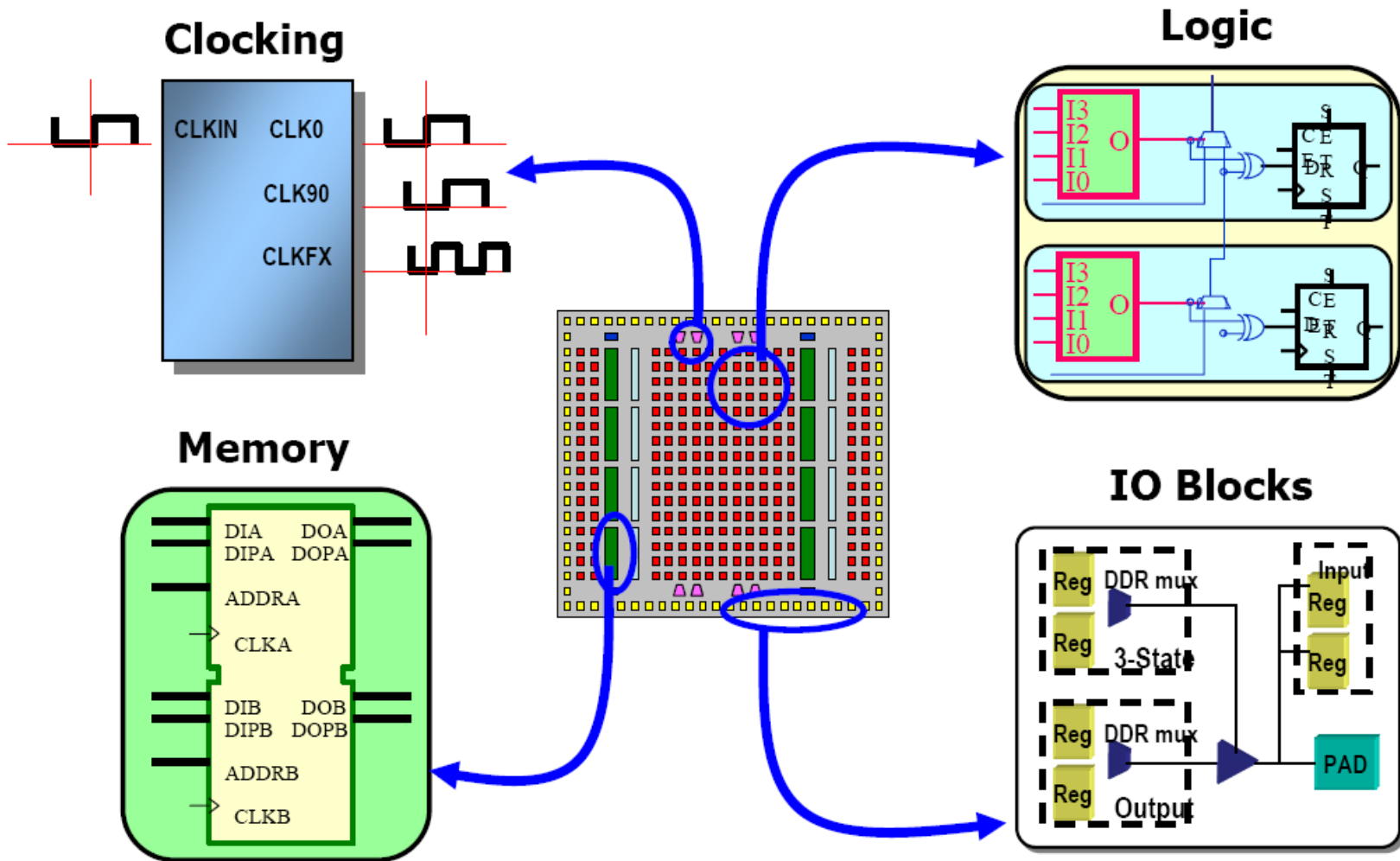
**Figura 1.14:** CLB XC2000 din FPGA produs de Xilinx

Cel mai mare avantaj al proiectării bazate pe FPGA este timpul foarte scurt de proiectare: timpul necesar de la începerea procesului de proiectare până în momentul în care primul circuit este disponibil. Întrucât nu este necesară nici o etapă de fabricație, un prototip functional FPGA poate fi obținut rapid, în momentul în care proiectul este transferat pe circuitul respectiv. Pretul unui circuit FPGA este de obicei mai mare decât pretul altor alternative pentru realizarea aceluiași proiect (cum ar fi matrice de porți logice sau celule standard), dar în cazul unui număr redus de circuite și pentru obținerea rapidă a unui prototip, FPGA oferă o opțiune foarte prețioasă.


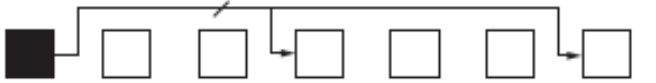



**1,000,000,000** (one [billion](#))

**$1 \times 10^9$**

# Basic FPGA architecture



# A practical example: Xilinx Virtex II Pro family

<p>24 Horizontal Long Lines 24 Vertical Long Lines</p>	
<p>120 Horizontal Hex Lines 120 Vertical Hex Lines</p>	
<p>40 Horizontal Double Lines 40 Vertical Double Lines</p>	
<p>16 Direct Connections (total in all four directions)</p>	
<p>8 Fast Connects</p>	

D5031\_60\_110200

Routing resources

# The Inverted Pyramid

Electronic Systems > **\$1 Trillion**  
(**10<sup>12</sup>** one million million)

Semiconductor > **\$220 B**  
one thousand million, or 10<sup>9</sup>

CAD **\$3 B**



# A free CAD flow for all who care!

- Alliance CAD tools
- "Architecture des Systèmes intégrés et Micro électronique" department of LIP6
- 875 000 Transistor design done
- Limit is your processor power and your design abilities!

Block  
Diagram

Design

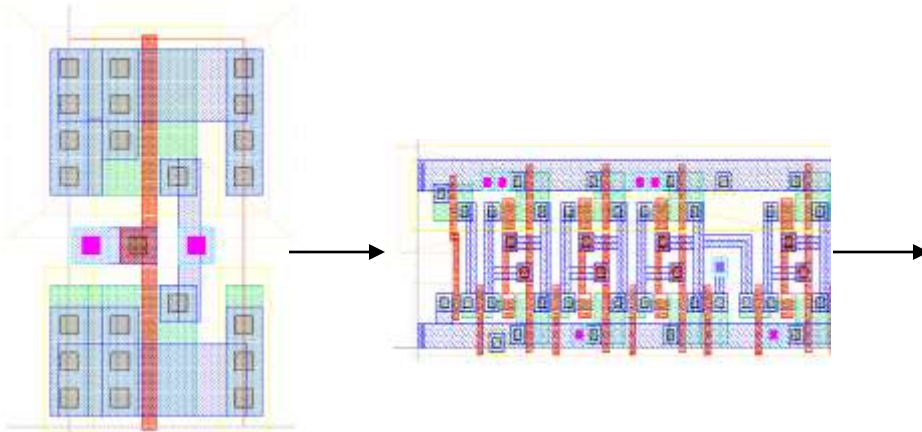
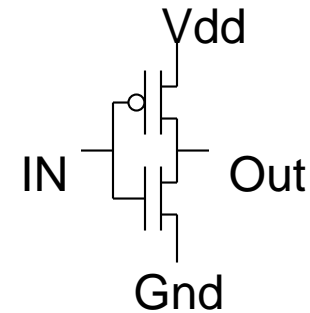
Syntesis

Layout



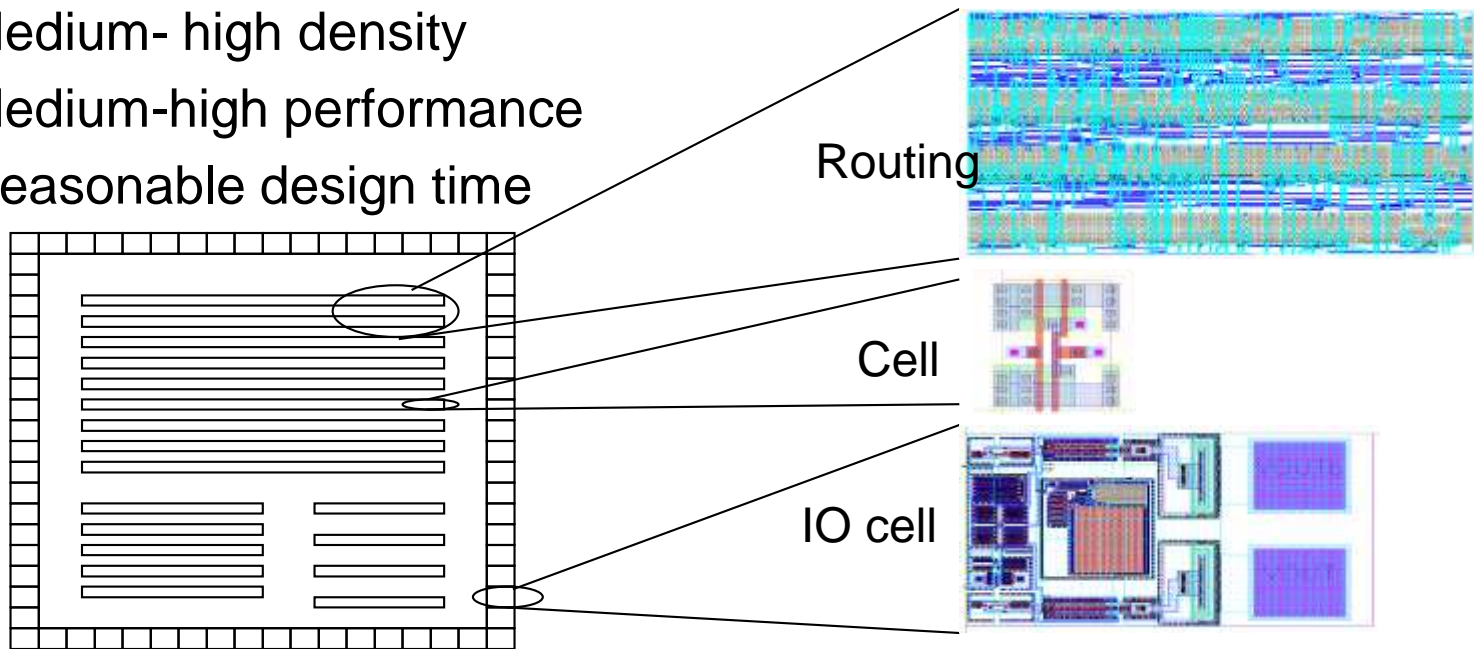
# Full custom

- Hand drawn geometry
- All layers customized
- Digital and analog
- Simulation at transistor level (analog)
- High density
- High performance
- Loong design time



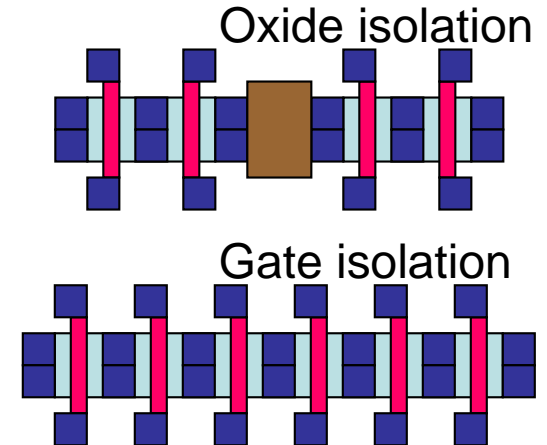
# Standard cells

- Standard cells organized in rows (and, or, flip-flops, etc.)
- Cells made as full custom by vendor (not user).
- All layers customized
- Digital with possibility of special analog cells.
- Simulation at gate level (digital)
- Medium- high density
- Medium-high performance
- Reasonable design time

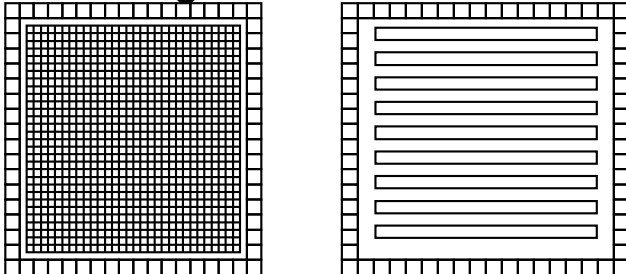


# Gate-array

- Predefined transistors connected via metal
- Two types: Channel based, Sea of gates
- Only metal layers customized
- Fixed array sizes (normally 5-10 different)
- Digital cells in library (and, or, flip-flops, etc.)
- Simulation at gate level (digital)
- Medium density
- Medium performance
- Reasonable design time

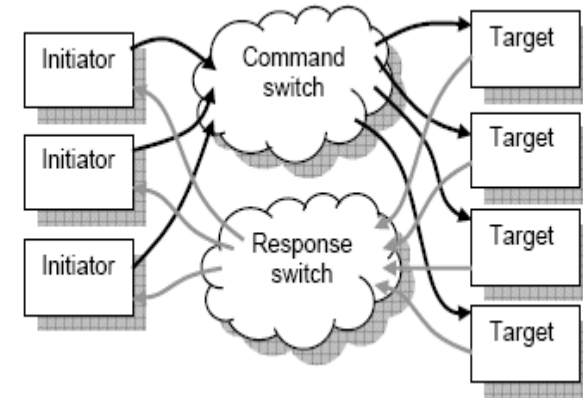
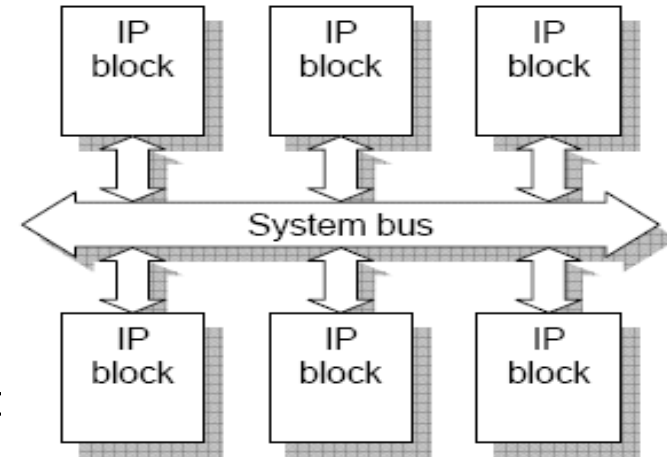


Sea of gates Channel based



# IP blocks

- Functional blocks from specialized companies
  - Rely on external expertise to reduce design time
  - Quite a large selection now available
- Hard blocks
  - Full custom by vendor, **Technology dependent**
  - All layers customized: High density, High performance
  - Digital and analog (ADC)
  - Simulation at behavioral or gate level (digital)
  - Memories, ADC, DAC, PLL, CPU, etc.
- Soft blocks
  - Synthesizable HDL model, **Technology independent**
  - User to synthesize into given technology using available libraries and perform himself timing and design verification
  - Digital blocks: DSP, processor, MPEG, etc.
- Use standard on-chip busses (like on boards)
  - New trend: on-chip networks (like computer networks)
- “System On Chip”: **SOC**



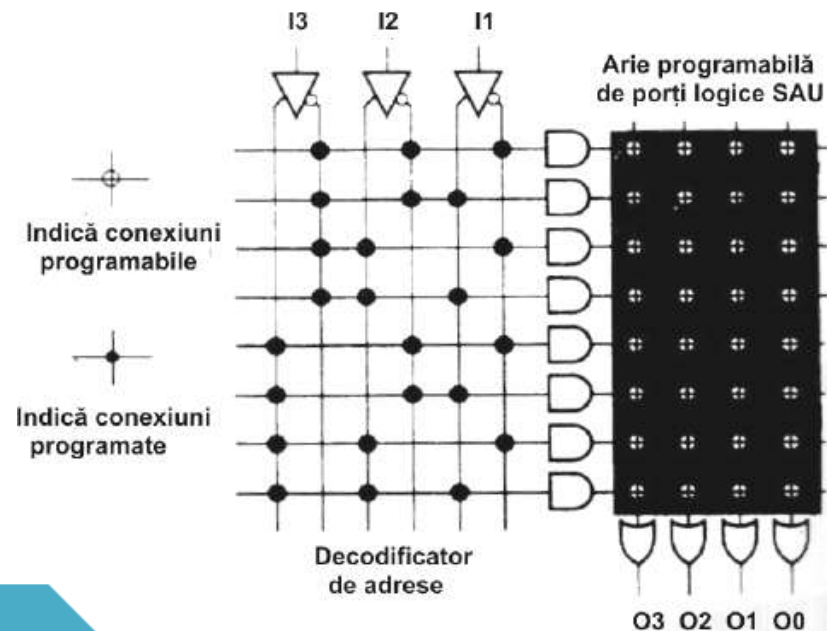
## 1.5.2 Proiectarea matricilor de porti logice

În ceea ce priveste posibilitatea de a obtine rapid un prototip, matricea de porti logice (GA – Gate Array) se plaseaza imediat dupa FPGA.

În timp ce implementarea proiectului circuitelor FPGA este facuta prin programarea de catre utilizator, pentru GA este facuta prin proiectarea si procesarea mastii de metal. Implementarile matricelor de porti logice necesita un proces de fabricatie în doi pasi.

Prima faza, care se bazeaza pe masti standard (generice), are drept rezultat o matrice de tranzistori neconectati pe fiecare structura GA.

Aceste circuite neconectate pot fi stocate pentru o proiectare ulterioara, care este finalizata prin definirea interconexiunilor dintre tranzistorii matricei (figura 1.15).



Block  
Diagram

Design

Synthesis

Layout

Figura 2.2.a Arhitectură PROM

two-step manufacture :

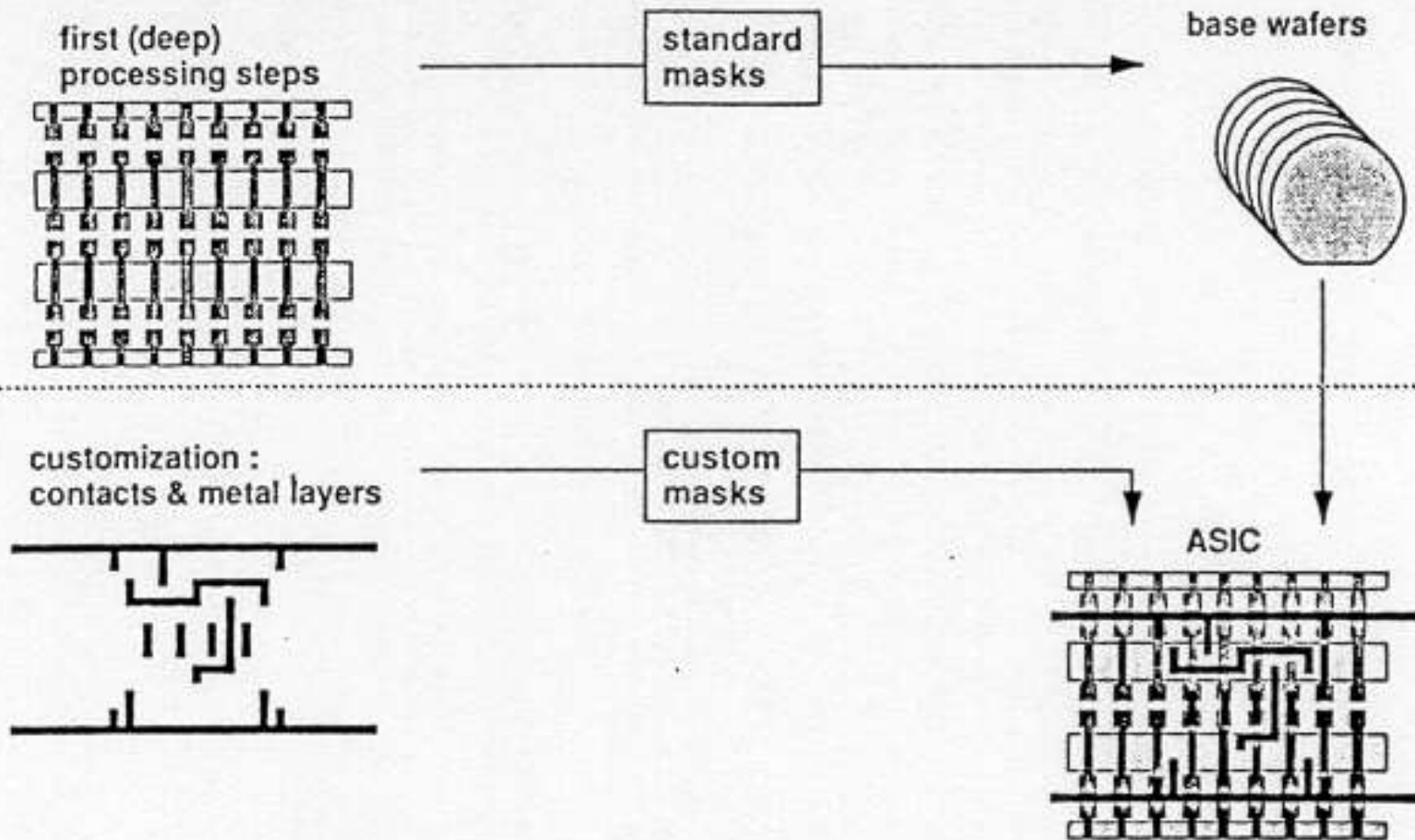


Figura 1.15: Pasii de baza necesari pentru implementarea GA

Din moment ce masca interconexiunilor metalice este facuta la sfarsit, timpul de proiectare poate fi, de asemenea, scurtat de la câteva săptămâni la câteva zile.

Figura 1.16 arata un colt al unui circuit GA, care contine ploturile de interconectare la marginea din stânga - jos, diodele pentru protectie la intrari/iesiri, tranzistorii nMOS si pMOS pentru circuitele de iesire din vecinatatea ploturilor de interconectare, matricile de tranzistori nMOS si pMOS, canalele de fire de conexiune, magistralele de alimentare si masa cu ferestrele de contact.

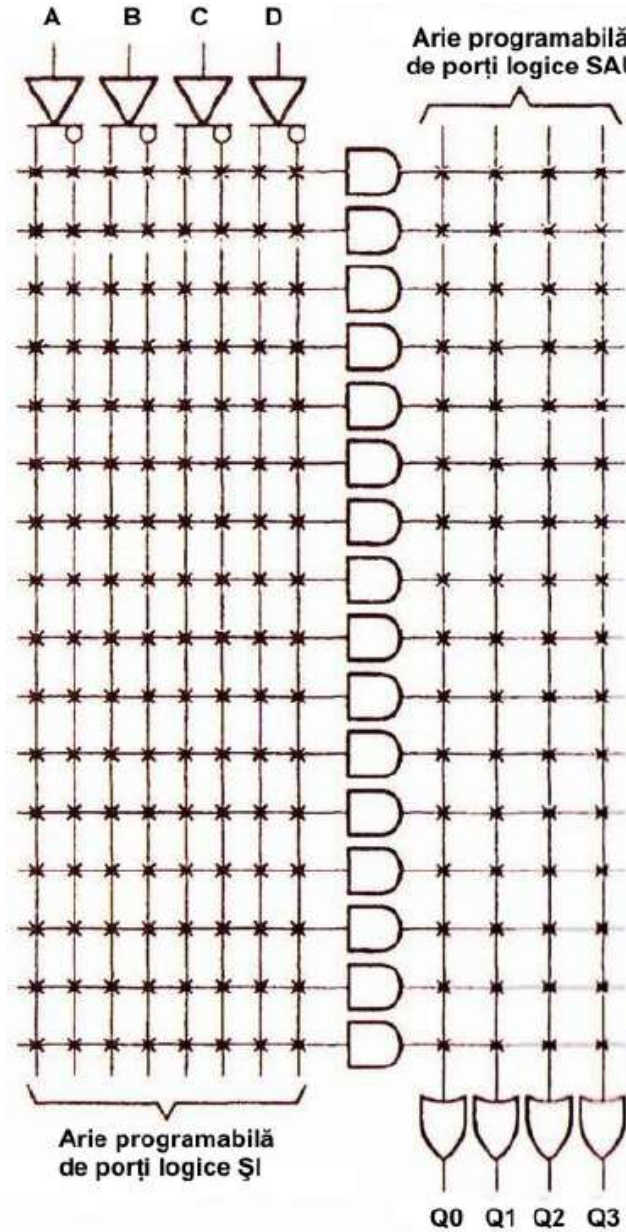


Figura 2.2.b Arhitectură FPLA



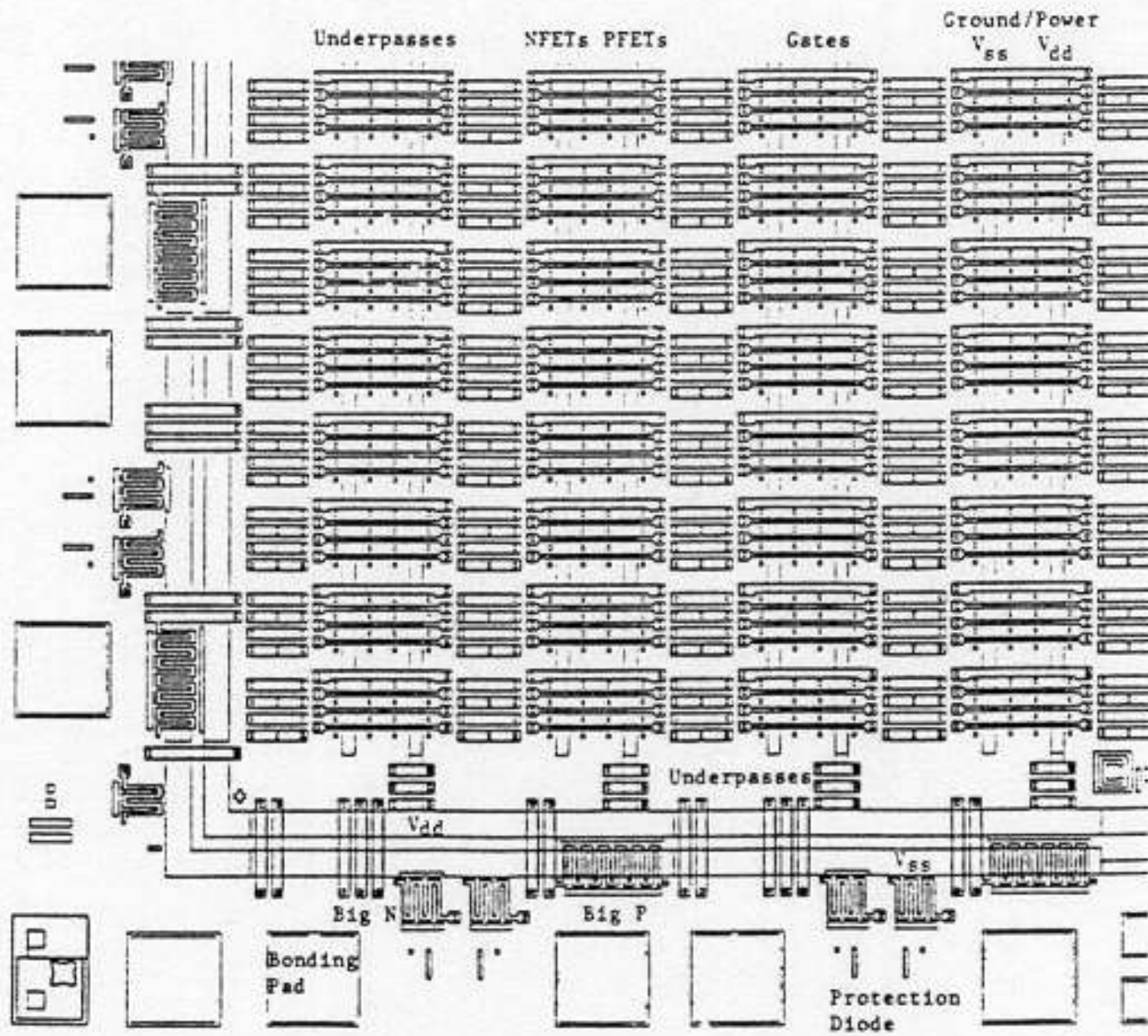
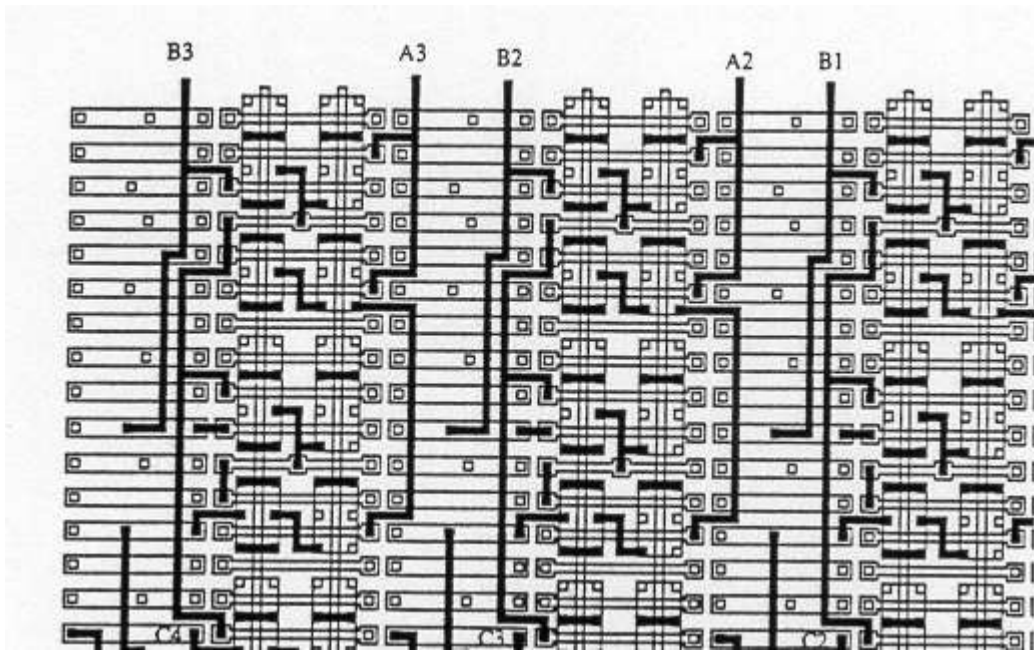


Figura 1.16 Un colt obisnuit al unui circuit GA

Figura 1.17 arata o portiune marita a matricii interne cu proiectul mastii de metal (liniile de metal evidentiata în negru) pentru a realiza o functie logica complexa. Structurile GA obisnuite folosesc portiuni dedicate, numite canale, pentru traseele dintre celule, dintre liniile sau coloanele de tranzistori MOS, asa cum se arata în figurile 1.16 si 1.17. Existenta acestor canale pentru trasee simplifica interconexiunile, chiar folosind numai un singur strat de metal. Sablonele interconexiunilor pentru realizarea unor porti logice de baza pot fi stocate într-o biblioteca, care poate fi utilizata pentru a aranja liniile de tranzistori nelegati în concordanta cu lista nodurilor. În timp ce platformele de tip GA contin numai linii de tranzistori nelegati, separati de traseele de interconectare, alte structuri ofera si matrici de memorie RAM dedicata, pentru a permite o densitate crescuta acolo unde sunt necesare functii de memorare.



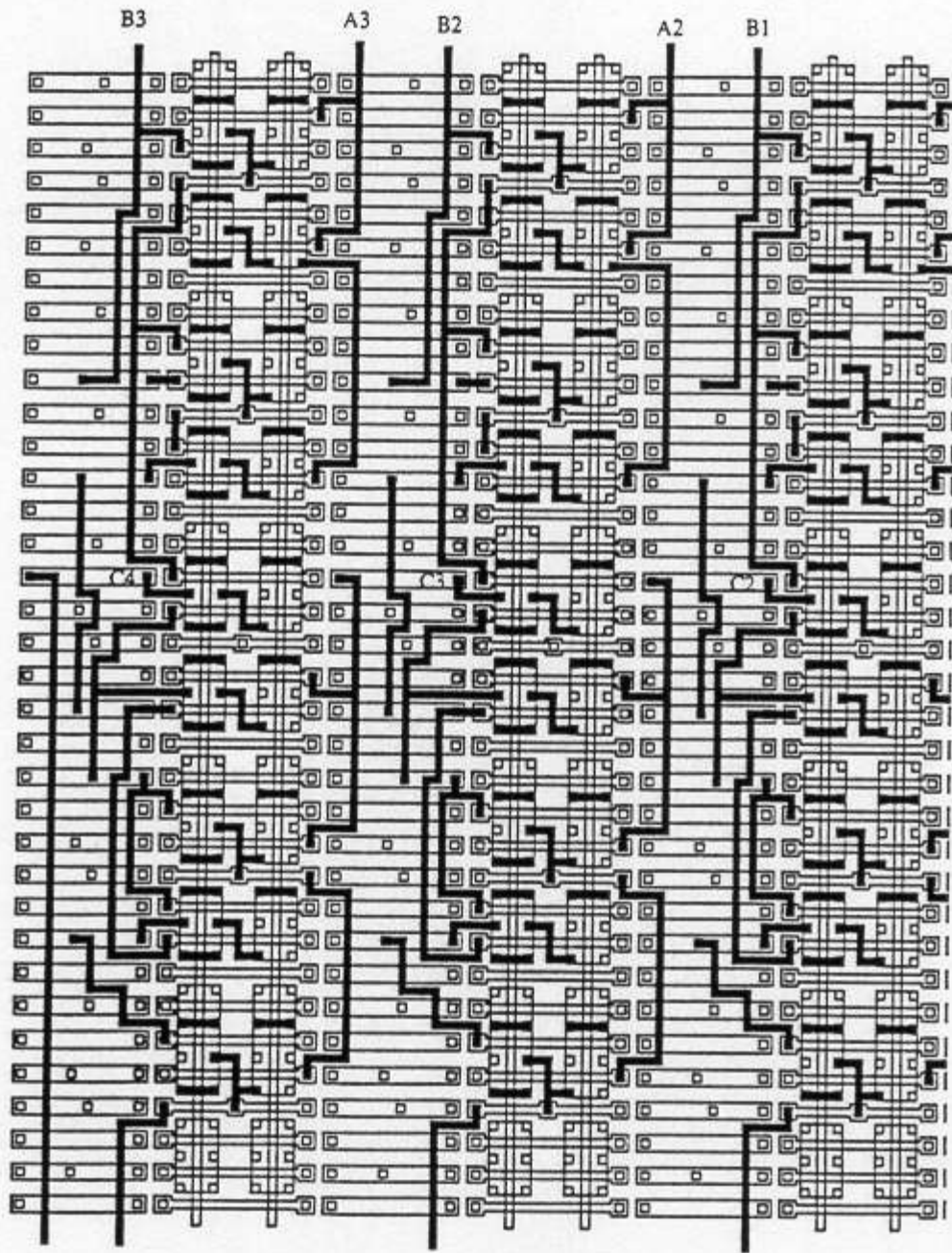


Figura 1.17: Proiectarea mastii de metal pentru realizarea unei functii logice complexe pe o structura GA cu canale de interconectare

Figura 1.18 prezinta o vedere de ansamblu a unui GA conventional si a unei structuri GA cu doua blocuri de memorie dedicata.

Odata cu utilizarea mai multor straturi de interconectare, plasarea traseelor poate fi realizata pe deasupra ariilor celulelor active; prin urmare, se poate renunta la canalele de conectare, ceea ce s-a realizat în circuitele SOG (Sea-of-Gates). În acest caz, întreaga suprafata a circuitului este acoperita cu tranzistori nMOS si pMOS neconectati. Ca si în cazul GA, tranzistorii care se învecineaza pot fi conectati folosind o masca de metal pentru a forma porti logice de baza.

Pentru conectarea celulelor, poate fi sacrificata o parte dintre tranzistorii nelegati. Aceasta abordare are ca rezultat o flexibilitate mai mare pentru interconexiuni si, în general o mai mare densitate. Platforma de baza a unui circuit SOG este aratata în figura 1.19. Figura 1.20 ofera o comparatie sumara între abordarea folosind circuite GA cu canale pentru interconectare si abordarea folosind circuite SOG, fara canale de interconectare.

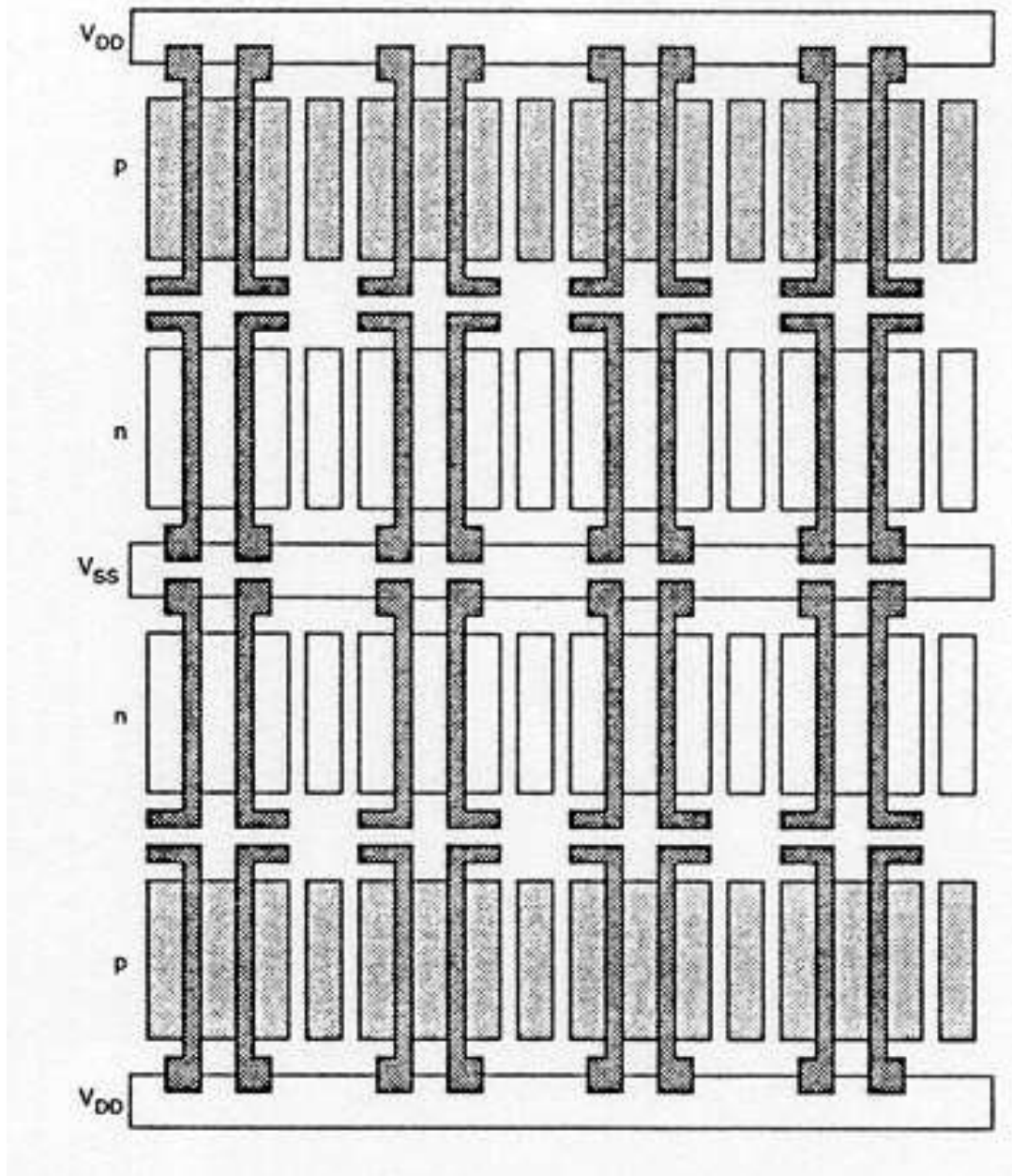


Figura 1.19: Structura unui circuit SOG (Sea-of-Gates)

În general, factorul de utilizare pentru un circuit GA, măsurat ca arie folosită din circuit raportată la aria totală a circuitului, este mai mare decât la FPGA. În același mod și viteza circuitelor păstrează aceeași relație, din moment ce se poate obține un proiect cu densitate mai mare prin folosirea mastii de metal. Structurile GA actuale pot implementa până la sute de mii de porți logice.

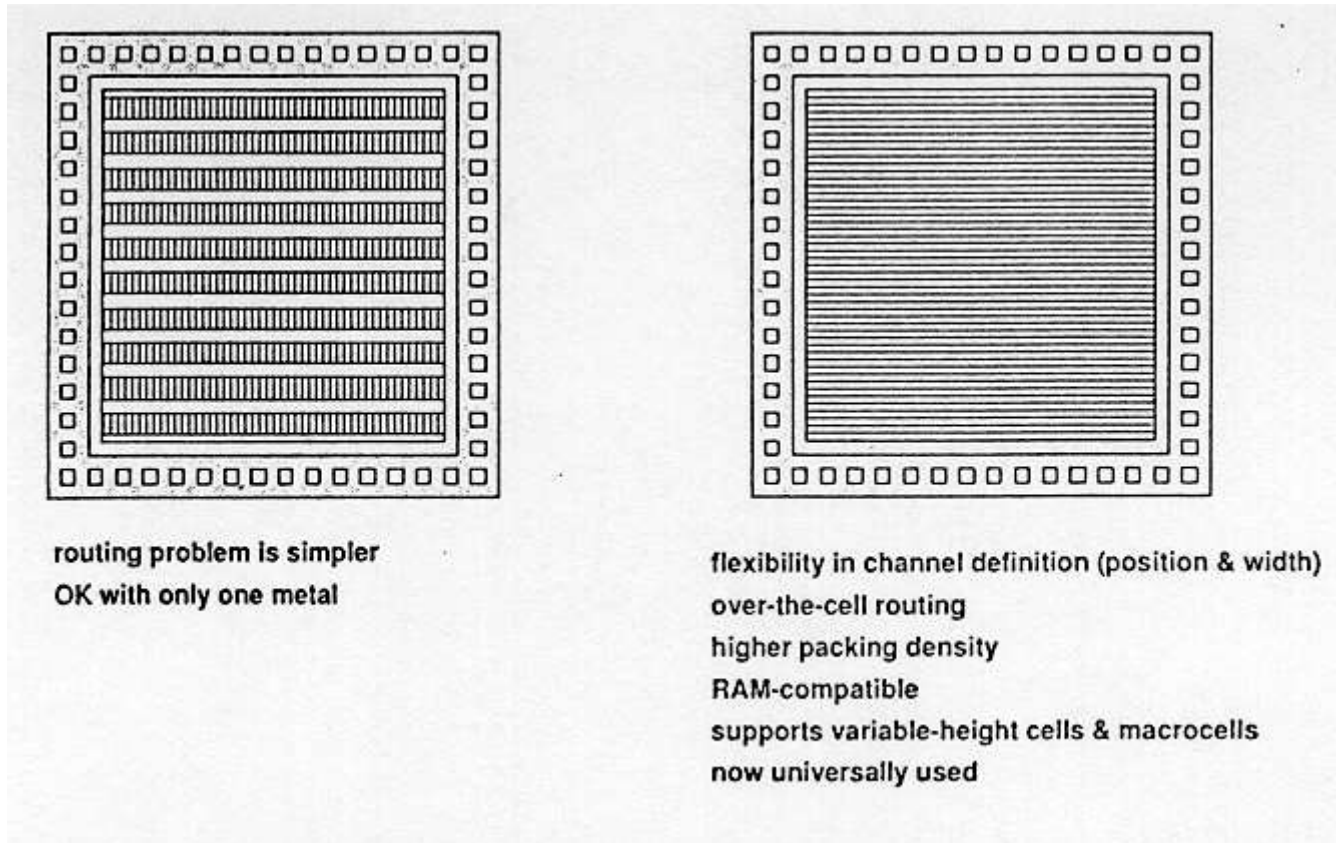


Figura 1.20: Comparatie între abordarea folosind GA cu canale de interconectare și abordarea folosind SOG fără canale de interconectare

### 1.5.3 Proiectarea bazata pe celule standard

Proiectarea bazata pe celule standard reprezinta unul dintre cele mai obisnuite stiluri de proiectare de catre utilizator. Aceasta necesita dezvoltarea de catre utilizator a unei set de masti.

Celulele standard mai sunt numite si policelule.

În acest stil de proiectare, sunt implementate, caracterizate si salvate într-o biblioteca de celule standard, toate celulele logice uzual folosite.

O biblioteca obisnuita poate contine cateva sute de celule incluzând inversoare, porti NAND, porti NAND, porti complexe AOI si OAI, latch-uri de tip D si flip/flop. Fiecare poarta poate avea mai multe implementari, pentru a asigura o capacitate de comanda a semnalului adecvata pentru diferite abordari.

De exemplu, o poarta inversor poate avea tranzistori de dimensiune standard, tranzistori de dimensiune dubla si tranzistori de dimensiune cvadrupla, astfel încât proiectantul poate alege dimensiunea potrivita pentru obtinerea unei viteze superioare a circuitului si a unei densitati marite.



Caracterizarea fiecăreia dintre celule este făcută pentru câteva categorii diferite care constau în:

- timpul de întârziere în raport cu capacitatea de încărcare;
- modelul de simulare a circuitului;
- modelul de simulare a sincronizării (timing);
- modulul de simulare la defecte;
- datele despre celule pentru plasare și interconectare;
- datele despre mască;

Pentru a permite un plasament automat al celulelor si trasarea conexiunilor intre celule, fiecare masca de celula este proiectata cu o înaltime fixata, asa încât mai multe celule pot fi adiacente, fata în fata, pentru a forma linii. Traseele de alimentare si masa sunt de obicei paralele cu limitele de sus si de jos ale celulei. Prin urmare celulele învecinate împart aceleasi magistralele de alimentare si masa. Pini de intrare si iesire sunt asezati pe limitele de sus si de jos ale celulei. Figura 1.21 arata forma obisnuita a unei celule standard. De remarcat ca tranzistorii nMOS sunt plasati aproape de traseul de masa în timp ce tranzistorii pMOS sunt plasati aproape de traseul de alimentare.

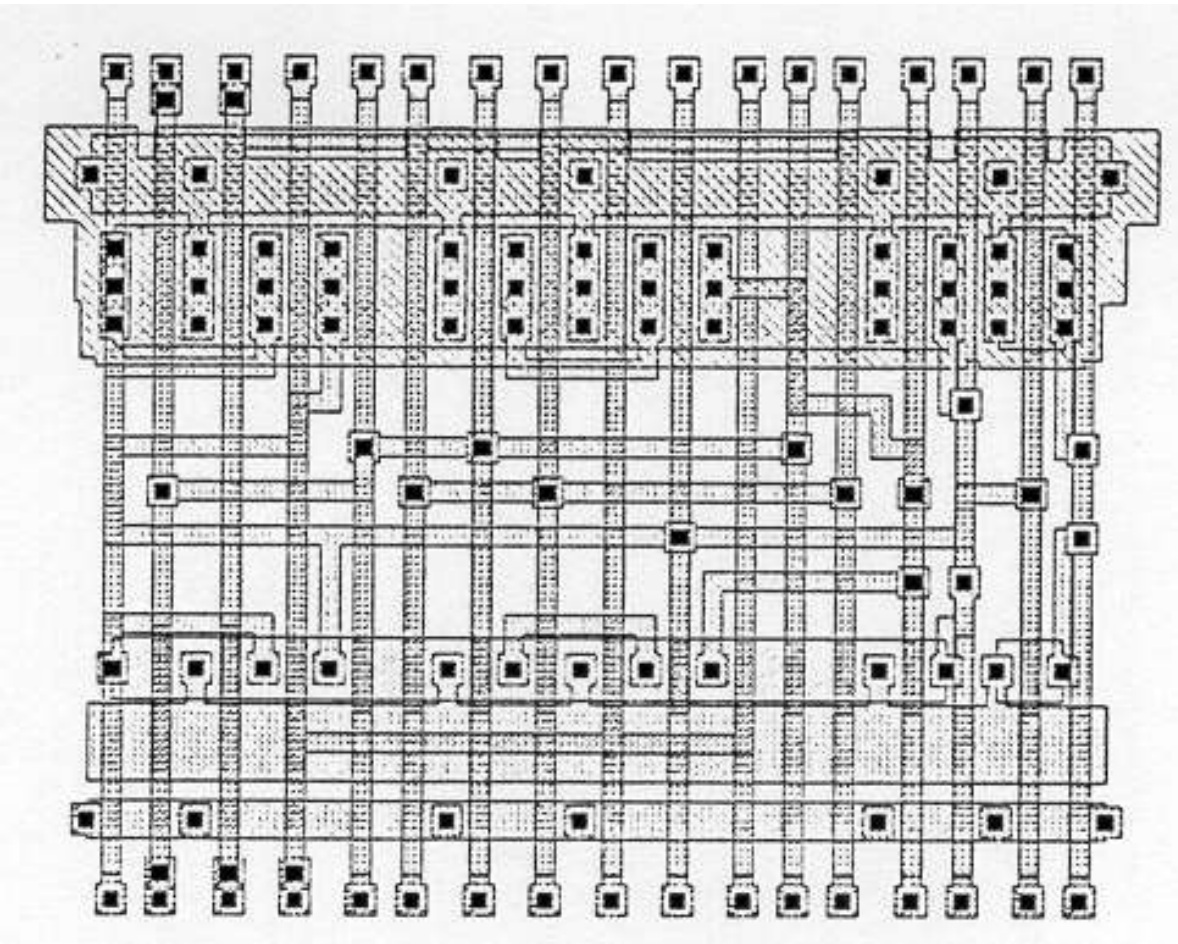


Figura 1.21: Forma obișnuită a unei celule standard

Figura 1.22 prezinta un plan pentru o proiectarea bazata pe celule standard. În interiorul cadrului de intrare/iesire, care este rezervat pentru celulele de intrare/iesire, suprafata circuitului contine linii sau coloane de celule standard. Între liniile de celule se afla canale pentru traseele dedicate.

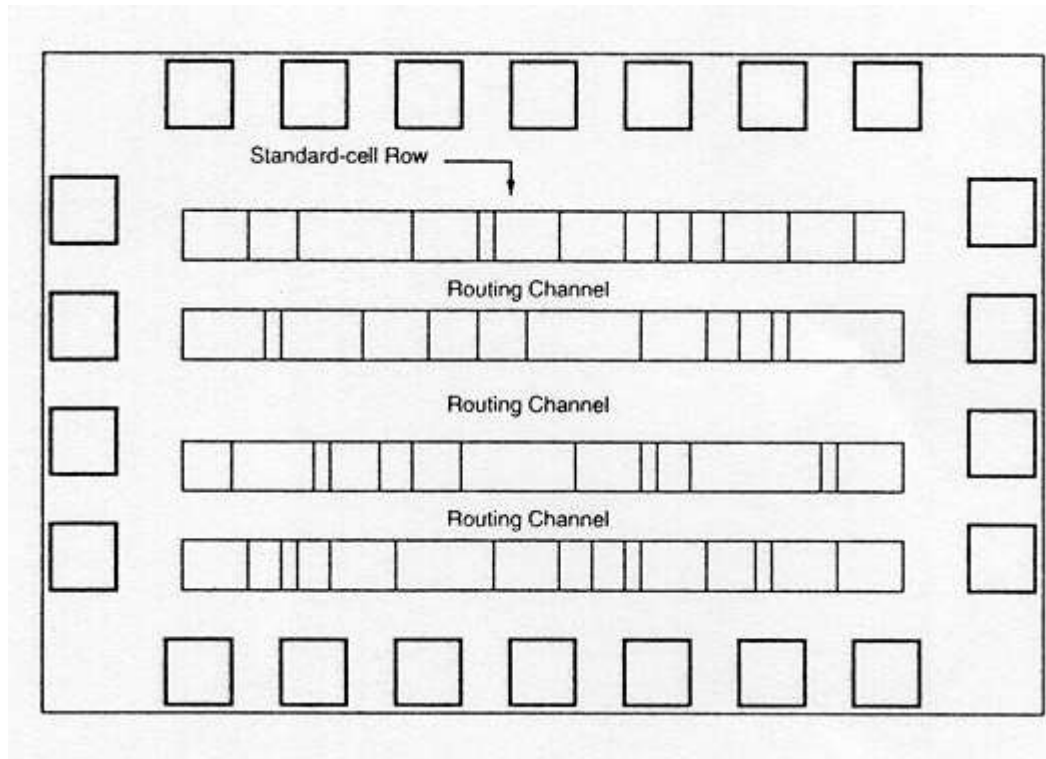


Figura 1.22: Un plan simplificat de proiect bazat pe celule standard

Ca si în cazul SOG, în care interconectarea se face pe deasupra stratului de celule, suprafata folosita de aceste canale poate fi redusa sau chiar îndepartata in conditiile in care liniile de celule ofera suficient spatiu de interconectare. Proiectarea fizica si forma celulelor logice asigura conditiile ca, în momentul în care acestea sunt plasate pe linii, înaltimile sa fie corespunzatoare, iar celulele vecine sa poata fi adiacente, fata în fata, ceea ce va asigura conexiuni normale pentru liniile de alimentare si masa, pe fiecare linie. Întârzierea semnalului, marginile de zgomot si puterea consumata ale fiecărei celule trebuie sa fie optimizate prin dimensionarea potrivita a tranzistorilor, folosind circuite simulate.

În cazul în care mai multe celule trebuie să împartă aceleași semnale de intrare și/sau ieșire, în forma celulei standard se poate încorpora o magistrală comună pentru semnal. Figura 1.23 prezintă o vedere simbolică simplificată a cazului în care o magistrală de semnal a fost introdusă între liniile celulelor standard. De remarcat că, în acest caz, circuitul constă în două blocuri, iar interconectarea traseelor de masă și alimentare trebuie asigurată de ambele părți ale structurii. Proiectarea bazată pe celule standard poate consta în mai multe macroblocuri de acest gen, fiecare corespunzând unei unități a sistemului, cum ar fi UAL (Arithmetic and Logic Unit), logica de control etc.

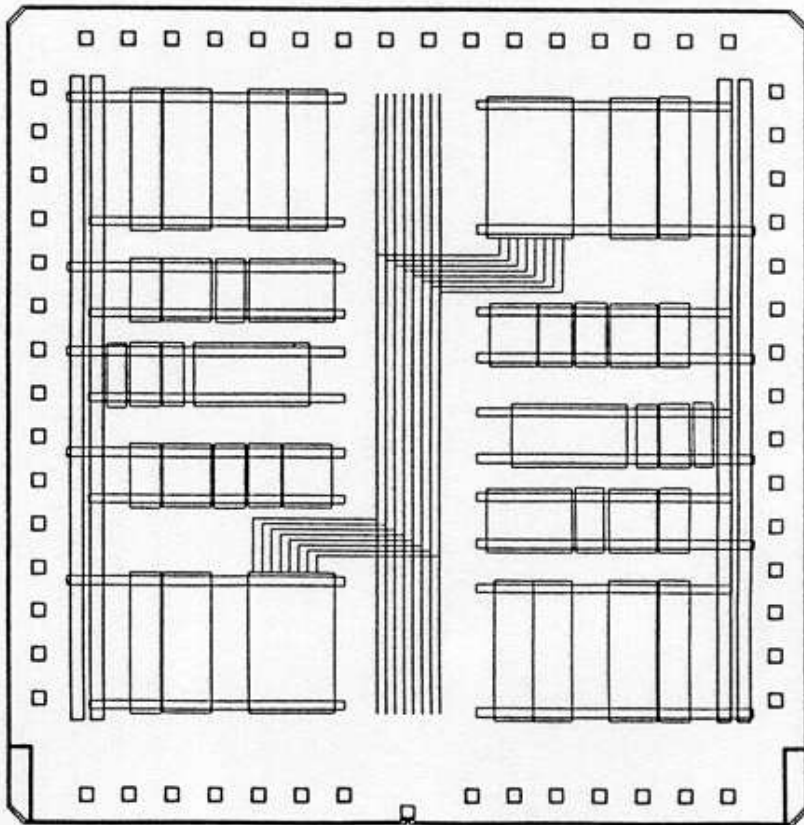


Figura 1.23: Planul simplificat format din două blocuri separate și o magistrală comună pentru semnal.

Dupa terminarea proiectarii logice a circuitului, folosind celulele standard din biblioteca, cea mai interesanta operatii sunt plasarea celulelor individuale pe linii si interconectarea acestora, corespunzator cu necesitatile proiectului în materie de viteza, suprafata ocupata si putere consumata. In vederea atingerii acestor obiective au fost dezvoltate numeroase unelte CAD evaluate. De asemenea, din mastile circuitului, pot fi extrase modelele de circuit care includ si elementele parazite de circuit. Acestea sunt folosite pentru simulari legate de sincronizare si analiza, pentru a identifica traseele critice de sincronizare. Plecând de la traseele critice ca intarziere, dimensionarea corecta a portii este de obicei folosita pentru a satisface specificatiile legate de sincronizare.

În multe circuite VLSI, cum ar fi microprocesoarele și procesoarele digitale de semnal, proiectarea bazată pe celule standard este folosită pentru a implementa module de logică complexă. O parte din circuite poate fi implementată exclusiv cu celule standard. În final, figura 24 prezintă forma detaliată a mării unui circuit bazat pe celule standard cu un bloc neîntrerupt de celule și trei blocuri de memorie plasate pe o parte a circuitului. De remarcat că, în interiorul blocului de celule, separarea dintre liniile vecine depinde de numărul de fire în canalul de interconectare dintre linii. Dacă în canalul de interconectare poate fi atinsă o densitate mare de interconexiuni, liniile de celule standard pot fi plasate mai aproape una de alta, conducând la o suprafață mai mică a circuitului. În cazul în care sunt disponibile blocuri dedicate de memorie, suprafața se poate reduce, întrucât realizarea elementelor de memorie folosind celule standard va ocupa o suprafață mai mare.



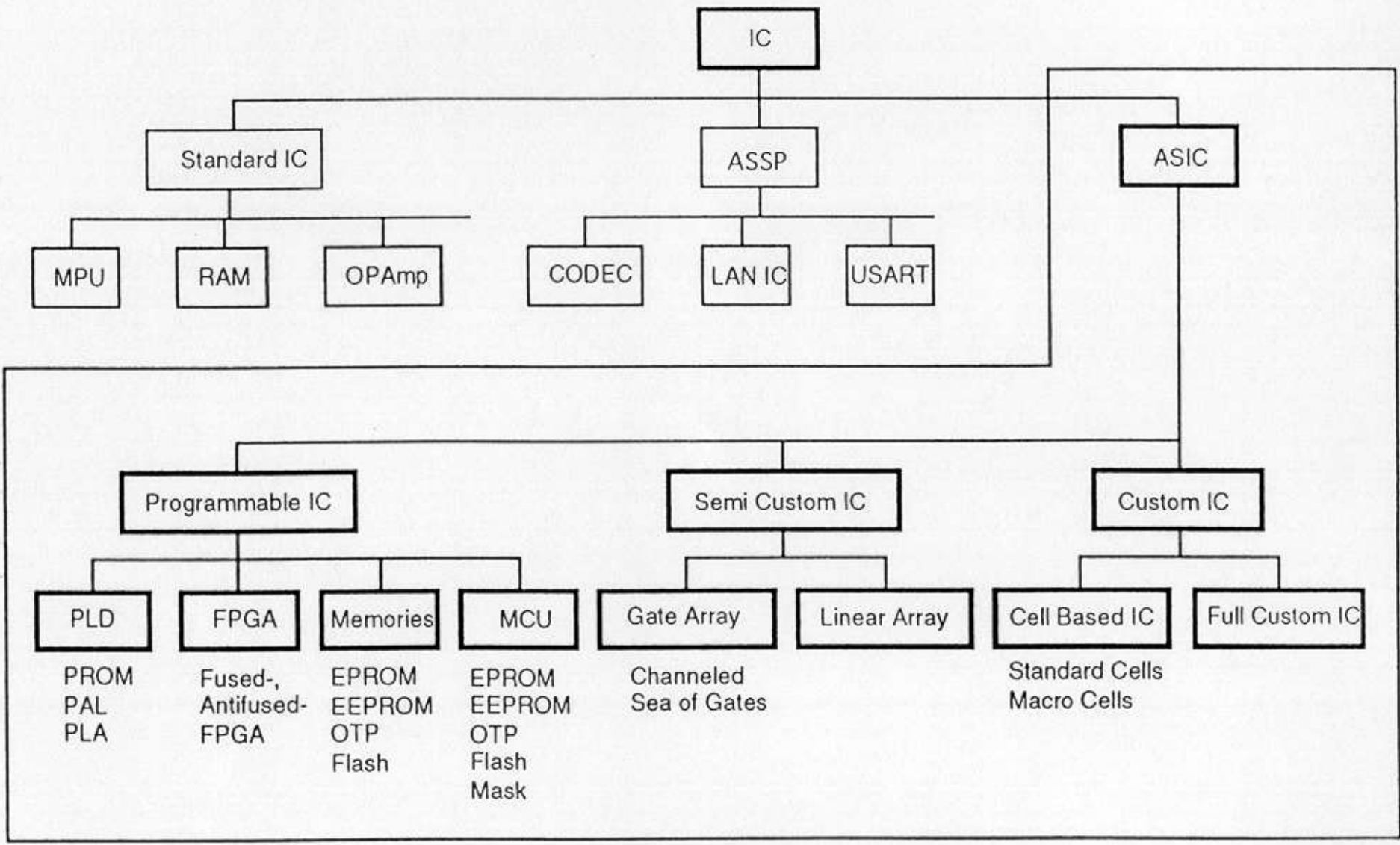


Figura 1.26: privire generala a stilurilor de proiectare VLSI

# Chip Design Flow (Wolf p.2)

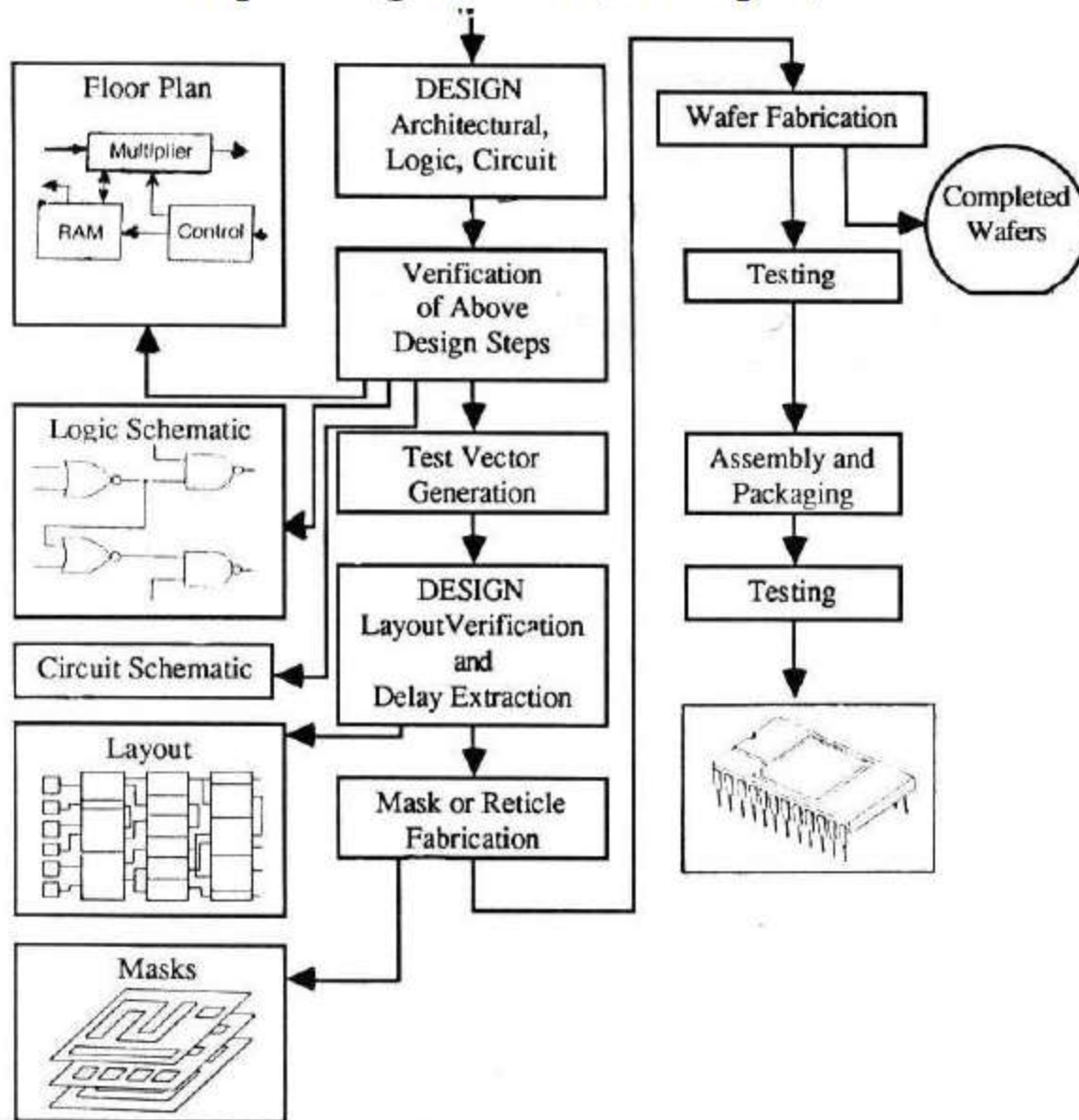


Fig. 1-2 Steps required for the manufacture of very large scale integrated circuits (VLSI).