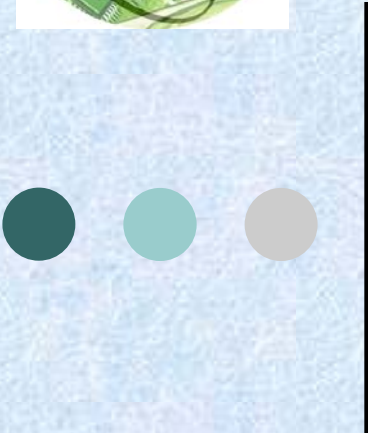


Three colored circles (dark teal, light teal, and grey) are arranged horizontally to the left of a vertical black line.

Proiectarea și Verificarea cu HDL a Circuitelor Digitale

Cap. 2 Platforme hardware pentru implementarea sistemelor digitale



Platforme hardware pentru implementarea sistemelor digitale

○ Cuprins

- 2.1 Metode de proiectare a sistemelor electronice
- 2.2 Procesoare
- 2.3 ASIC (Circuite integrate cu aplicații specifice)
- 2.4 Dispozitive programabile



2.1 Metode de proiectare a sistemelor electronice

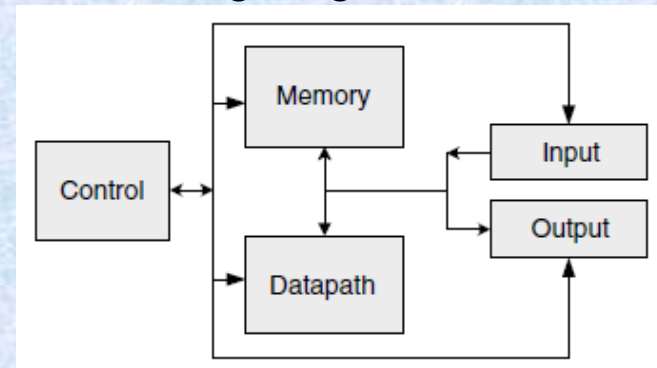
- Considerente pentru implementarea unui proiect:
 - Tehnologia de fabricație
 - Organizarea elementelor și blocurilor
 - Materializarea funcționalității sistemului
- Aceste considerente afectează modul de proiectare al sistemului și stabilește limitele complexității acestuia.
- Majoritatea IC sunt bazate pe tehnologia CMOS. Elementele de comutație CMOS (porți NOT, NAND, NOR, etc) stau la baza implementării funcțiilor booleene.
- Parametrii esențiali în organizarea elementelor de comutație:
 - **Regularitatea** – se referă la posibilitatea reutilizării elementelor din proiect. Structura unui proiect poate limita gradul de utilizare și performanțele acestuia
 - **Granularitatea** – exprimă nivelul de funcționalitate încorporat într-un obiect.
 - Granulație fină: porți logice, bistabile
 - Granulație medie: unități aritmetice logice, etc
 - Granulație mare: blocuri din proiect (procesor, memorie, interfețe)

2.1 Metode de proiectare a sistemelor electronice

2.1.1 Blocuri de bază ale sistemelor integrate

- Bloc intrări-ieșiri
 - Conectează sistemul cu lumea exterioară
- Cale de date
 - Tipic constă dintr-o interacțiune de **porți logice**, **operatori aritmetici** (sumatoare, multiplicatoare, etc) și **registre** pentru memorarea rezultatelor intermediare.
- Memoria
 - Poate fi caracterizată de dimensiune, număr de porturi, latență și lățime de bandă.
- Bloc de control
 - Determină ce acțiuni sunt efectuate la orice moment de calea de date și când și cum comunică memoria, blocurile I/O și calea de date între ele.
- Rețeaua de comunicație
 - Canalele de comunicație pot fi: **canale dedicate** (semnale) și **canale partajate** (bus-uri și rețele)

Componentele unui procesor digital generic.



2.1 Metode de proiectare a sistemelor electronice

2.1.2 Elemente cheie ale proiectelor (design metrics)

○ **Costul.** Este de 2 tipuri:

- **Cost de dezvoltare** (NRE cost-nonrecurring engineering cost)
- **Cost de fabricație** a unei copii a sistemului (cost unitar- UC)

Costul total (TC):

$$TC = NRE + Q \cdot UC$$

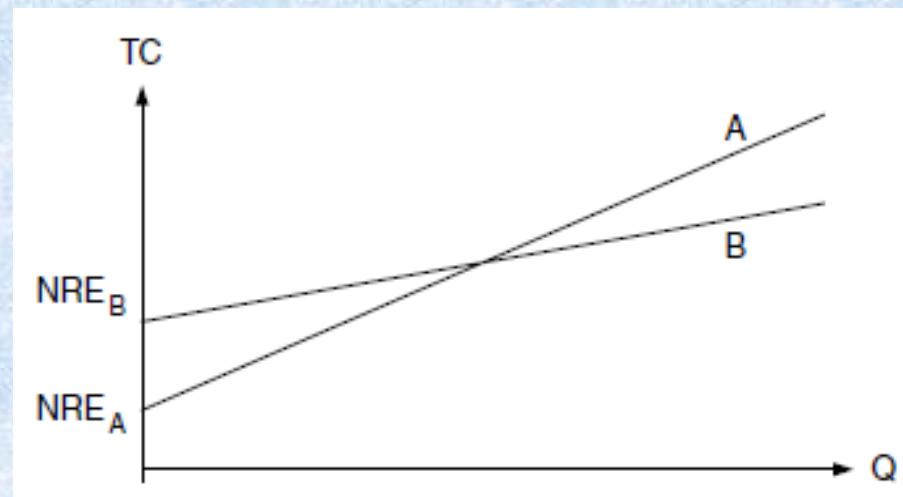
Q = cantitate

Costul final, per produs (FC):

$$FC = CT/Q = NRE/Q + UC$$

NRE cost pentru un ASIC in tehnologie < 100nm este de câteva milioane de dolari!

Evaluarea costului în funcție de cantitate

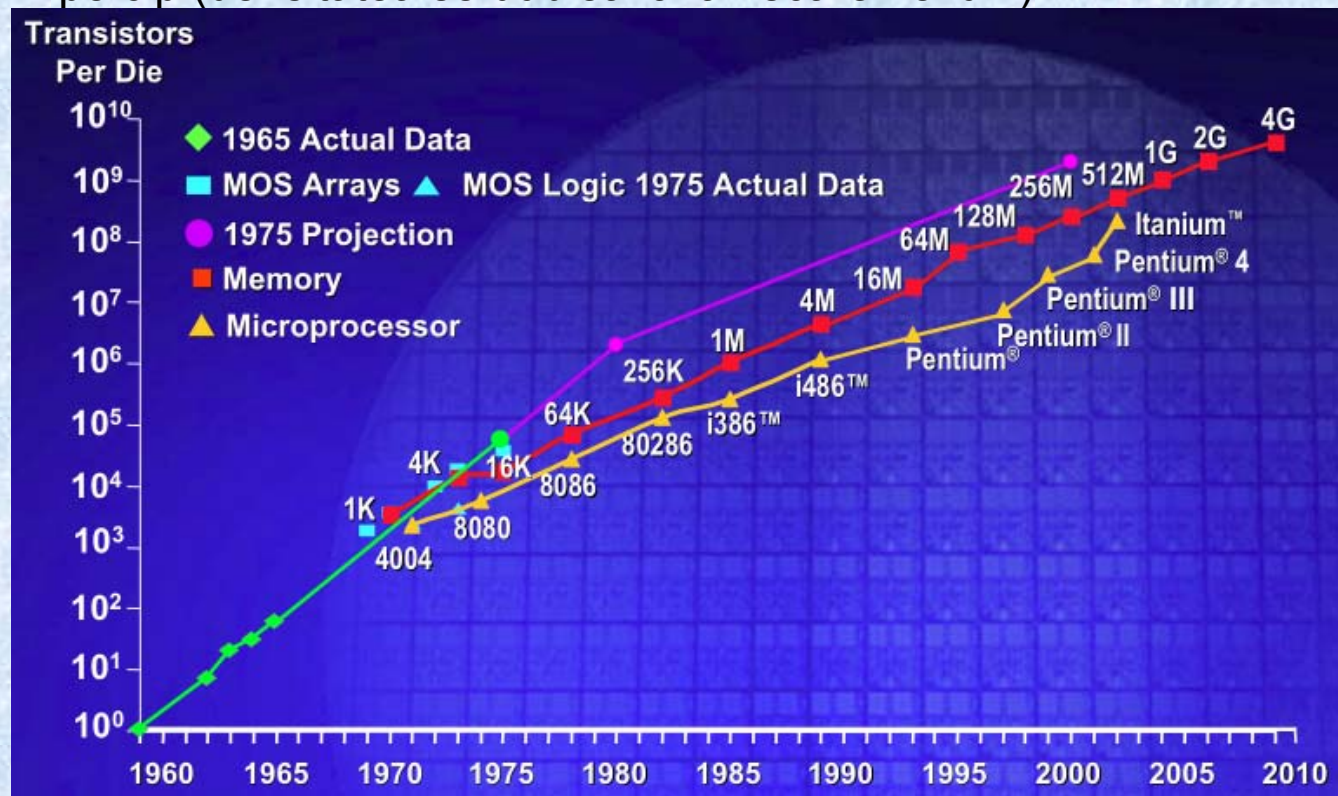


2.1 Metode de proiectare a sistemelor electronice

2.1.2 Elemente cheie ale proiectelor (design metrics)

- **Densitatea.** Legea lui Moore

- În cursul ultimelor decenii: creștere exponențială a nr. de tranzistoare pe cip (densitatea se dublează la fiecare 18 luni)



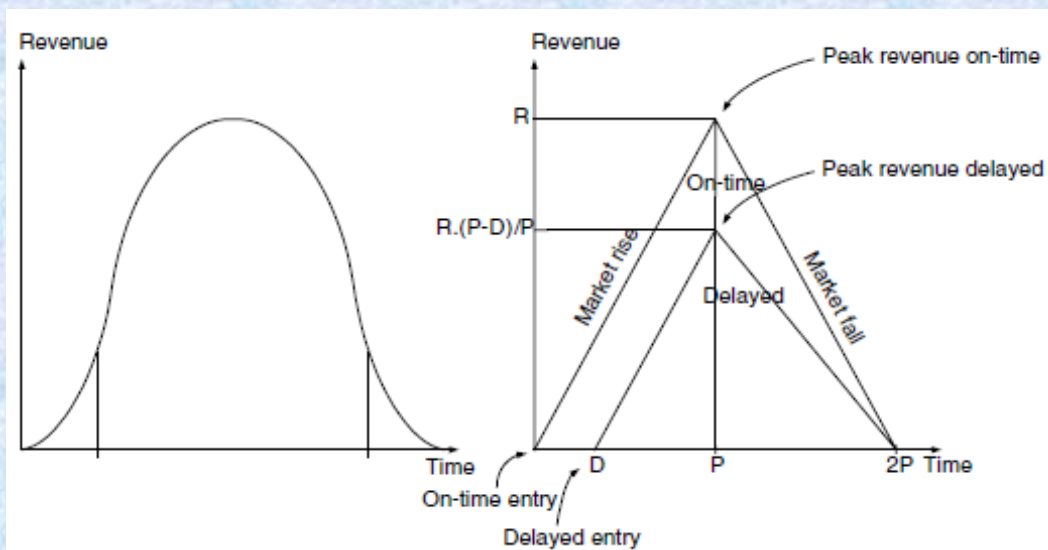
2.1 Metode de proiectare a sistemelor electronice

2.1.2 Elemente cheie ale proiectelor (design metrics)

○ Time-to-Market

- Timpul necesar dezvoltării unui produs până la momentul în care poate fi vândut
- Pentru sistemele electronice actuale: 6-12 luni
- **Market window** (fereastra de piață) – durata în care produsul are cele mai mari vânzări

Model tipic al distribuției veniturilor (fereastra de piață) și modelul liniar simplificat



Întârzierea (D) este un factor care poate produce pagube importante.

De exemplu: o întârziere de 5 săpt. pentru un produs care are cerere pe piață 1 an produce o pierdere de 27%!

Motiv pentru efort de adoptare de noi tehnologii, noi metode și instrumente de proiectare.



2.1 Metode de proiectare a sistemelor electronice

2.1.2 Elemente cheie ale proiectelor (design metrics)

○ Performanțele proiectului

- În general criteriile de performanță se consideră **frecvența de ceas** și **numărul de instrucțiuni pe secundă**.
- Alte criterii:
 - **latența** (durata dintre începutul și sfârșitul unei operații) Ex: durata procesării unei imagini
 - **viteza** (numărul de operații pe secundă). Ex: nr. de imagini procesate într-o secundă.

○ Puterea disipată

- Este foarte importantă în cazul dispozitivelor portabile
- **Putere statică** – este consumată în perioada de inactivitate a sistemului
- **Putere dinamică** – este datorată activităților de comutație (încărcare-descărcare capacități de sarcină)
- În dispozitivele CMOS:

$$P = k \cdot C \cdot V^2 \cdot f$$

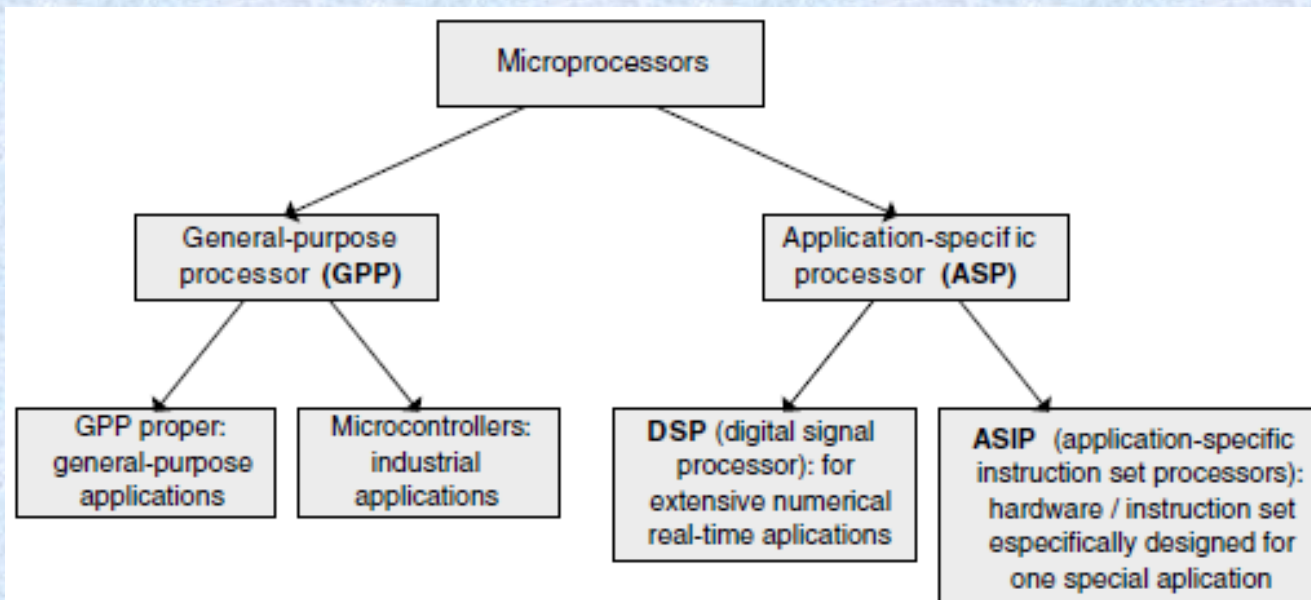
- Se adoptă tehnici de reducere a puterii consumate (oprire procesor în perioada de inactivitate, controlul puterii pe componente, etc)



2.2 Procesoare

- Arhitecturi care execută un set redus de instrucțiuni (*reduced instruction set computer*-RISC): microprocesoare, procesoare digitale de semnal (DSP), procesoare cu set de instrucțiuni specifice aplicației (application specific instruction set processors (ASIPs), etc

Clasificarea microprocesoarelor în funcție de nivelul de specializare



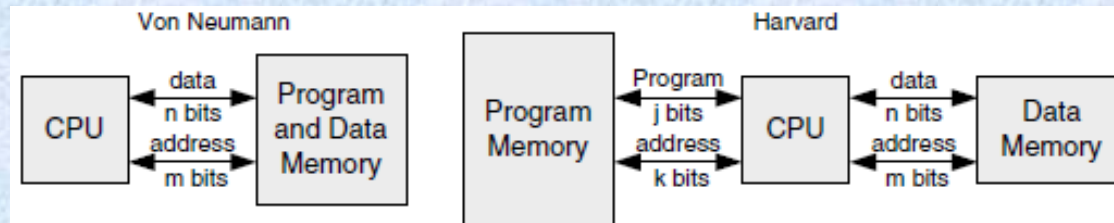
2.2 Procesoare

2.2.1 Microprocesoare

Tipuri și caracteristici ale microprocesoarelor din punct de vedere al structurii hardware

Complex Instruction Set Computer (CISC)	Reduced Instruction Set Computer (RISC)
• Large number of complex addressing modes • Many versions of instructions for different operands • Different execution times for instructions • Few processor registers • Microprogrammed control logic	• One instruction per clock cycle • Memory accesses by dedicated load/store instructions • Few addressing modes • Hard-wired control logic
Very Long Instruction Word (VLIW)	Superscalar Processors
• Statically determined instruction-level parallelism (under compiler control) • Instructions are made of different machine operations whose execution is started in parallel • Many parallel functional units • Large register sets	• Subclasses of RISCs or CISCs • Multiple instruction pipelines for overlapping execution of instructions • Parallelism not necessarily exposed to the compiler

Arhitecturi ale microprocesoarelor din punct de vedere al accesului la memorie



Arhitecturi Von Neumann: procesoare cu scop general; Arhitecturi Harvard: DSP, microcontrolere



2.2 Procesoare

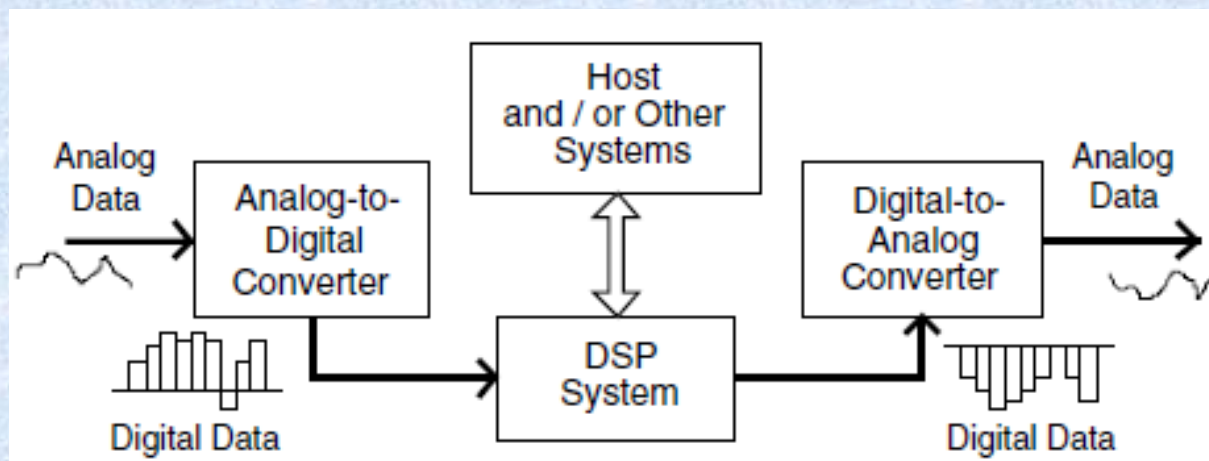
- 2.2.2 Microcontrolere
 - Sunt computere pe un singur cip.
 - Sunt lente și au memorie foarte mică
 - Sunt ieftine
 - Sunt foarte ușor de interfațat cu alte dispozitive
 - Se programează în C sau limbaje de asamblare
 - Se înglobează de regulă în alte sisteme acționând ca unități de procesare sau control.
 - Folosesc de regulă arhitecturi RISC cu un set foarte redus de instrucțiuni
- 2.2.3 Procesoare înglobate
 - Sunt sisteme de calcul înglobate în diverse echipamente electronice (radio, TV, telefoane, echipamente de uz casnic, automobile, etc)
 - Sunt prezente în număr mare în noile automobile
 - Fiecare echipament dintr-un sistem PC conține un procesor înglobat (mouse, hard-disk, CD-ROM/DVD drive, etc)

2.2 Procesoare

2.2.4 Procesoare digitale de semnal (DSP)

- Sunt utilizate pentru analiza, modificarea și extragerea informațiilor din semnale.
- Intern, semnalele sunt reprezentate digital ca o secvență de eșantioane.
- Semnalele digitale sunt obținute din semnalele fizice cu ajutorul traductoarelor (ex: microfon) și a convertoarelor analog-digitale (ADC).
- Semnalele digitale sunt convertite înapoi în semnale analogice cu ajutorul convertoarelor digital-analogice (DAC).

Interacțiunea tipică a unui sistem DSP cu lumea reală





2.2 Procesoare

2.2.4 Procesoare digitale de semnal (DSP)

○ Aplicații

- **Procesare de imagini** (recunoașterea formelor, hărți meteo satelit, roboți)
- **Instrumentație și control** (analiză de spectru, reducere zgomot, compresie date, ghidaj, procesare GPS, etc)
- **Procesare audio-video** (recunoașterea vocii, sinteza de voce, audio digital, egalizare, prelucrare video, etc)
- **Militare** (comunicații securizate, ghidaj rachete, procesare semnal radar, etc)
- **Telecomunicații** (transmisii de date, videoconferință, telefonie mobilă digitală, sisteme wireless, modem-uri, etc)
- **Biomedicale** (monitorizarea pacienților, scanere, analiză ECG, etc)

○ Caracteristici

- DSP implementează algoritmi pentru a efectua filtrare cu răspuns finit (infini) FIR (IIR), transformări frecvență-timp, transformata Fourier rapidă (FFT), operații de convoluție sau corelație, etc
- Cele mai importante operații: calcule numerice repetitive
- Majoritatea operațiilor sunt procese în timp real; este necesară o memorie cu lățime de bandă mare
- Majoritatea operațiilor DSP necesită adunări și înmulțiri; de aceea, DSP conțin sumatoare și multiplicatoare utilizate în paralel într-o singură instrucțiune: unități de multiplicare-acumulare (MAC)



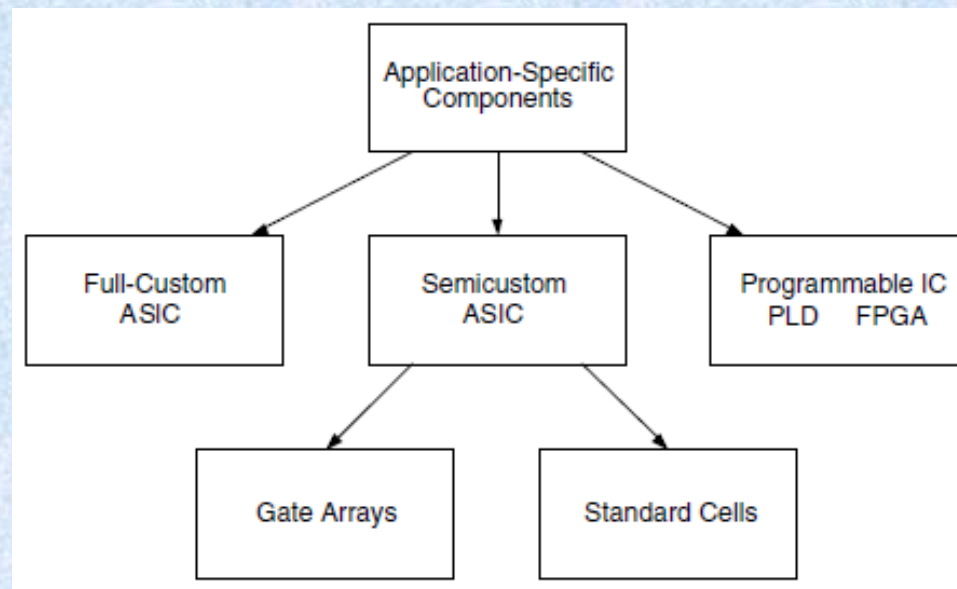
2.3 Circuite integrate cu aplicație specifică (ASIC)

- Sunt circuite integrate realizate special pentru a îndeplini o anumită funcționalitate.
- **Avantaje** față de microprocesoare sau dispozitive logice programabile:
 - Viteză de lucru mare
 - Consum mic de putere
 - Cost mic la volume de producție mari
 - Securitate mai bună
 - Control mai bun al caracteristicilor de intrare/ieșire
 - Structură și împachetare mai compactă
- **Dezavantaje:**
 - Timp mare (câteva săptămâni) de la finalizarea proiectării până la implementarea fizică.
 - Scump pentru producție în cantități mici
 - Cost NRE foarte mare (investiții mari în unelte CAD, stații de lucru, manoperă)
 - Odată implementat nu poate fi modificat



2.3 ASIC

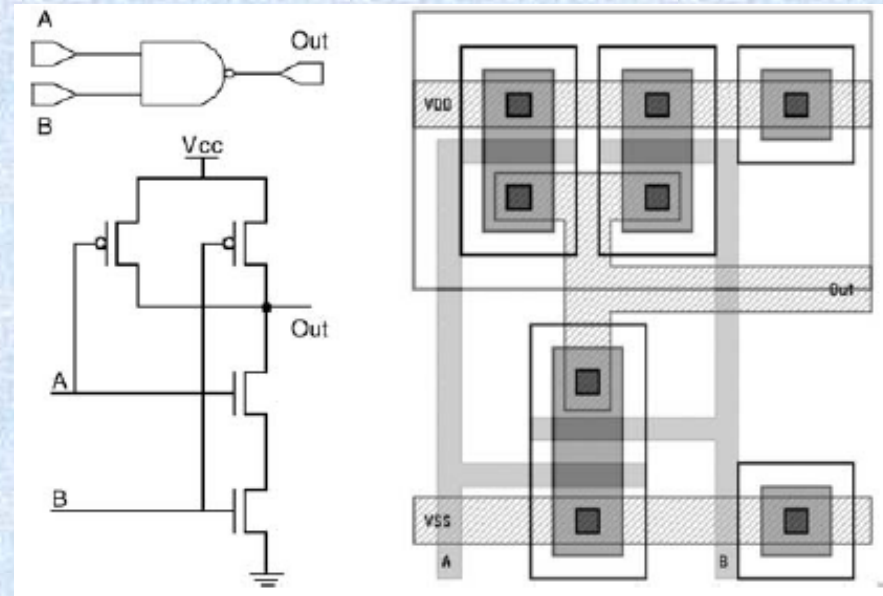
- Componentele cu aplicație specifică se împart în: full-custom ASIC, semicustom ASIC și circuite integrate programabile



2.3 ASIC

2.3.1 Full-custom ASIC

- Toate măștile sunt proiectate complet
- Proiecte full-custom oferă performanțe mai mari și arii ocupate mai mici
- **Dezavantaje:** timp de proiectare mare, complexitate mai mare, risc mare de erori
- Proiectarea full-custom se adoptă când nu se pot utiliza biblioteci sau blocuri IP sau când se doresc performanțe foarte mari.
- Majoritatea activităților full-custom sunt pentru generarea de celule pentru biblioteci sau părți minore dintr-un proiect
- **Exemple:** CI de înaltă tensiune pentru auto/avioane, CI analogice, dispozitive de comunicații digitale/analogice, senzori, traductoare, etc
- În trecut, microprocesoarele și memoriile se proiectau full-custom



Poartă ȘI-NU (NAND).

Simbol, Circuit, Layout

2.3 ASIC

2.3.2 Semi-custom ASIC

- Proiectele semi-custom se realizează la nivel de poartă logică.
- Soluții pentru proiectare semi-custom:
 - **Arii de porți** (gate arrays)
 - **Celule standard** (standard cells)

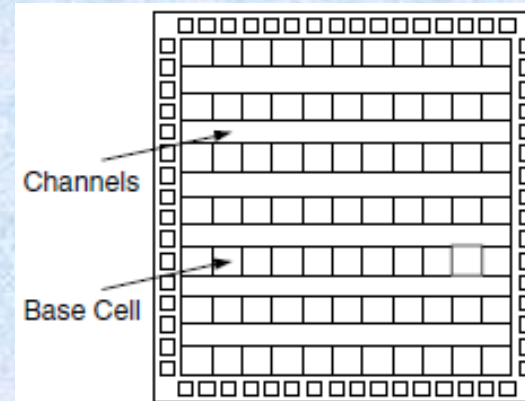
Gate-arrays ASIC

Ariile de porți sunt compuse din arii continue de tranzistoare p și n

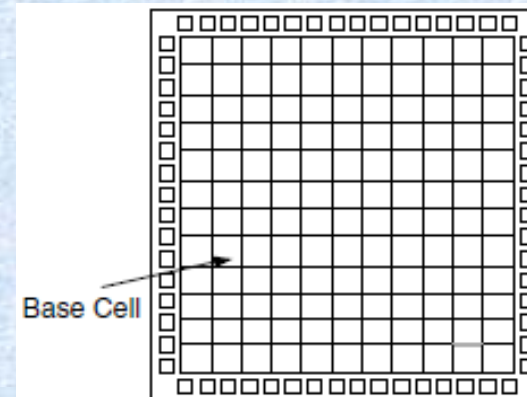
Furnizorii de siliciu pun la dispoziție waferi de bază sau master care apoi sunt personalizate conform informațiilor date de client cu privire la conectarea tranzistoarelor.

Tipuri de arii de porți: cu canal și fără canal pentru interconexiuni

Arii de porți cu canal pentru interconexiuni



Arii de porți fără canal pentru interconexiuni

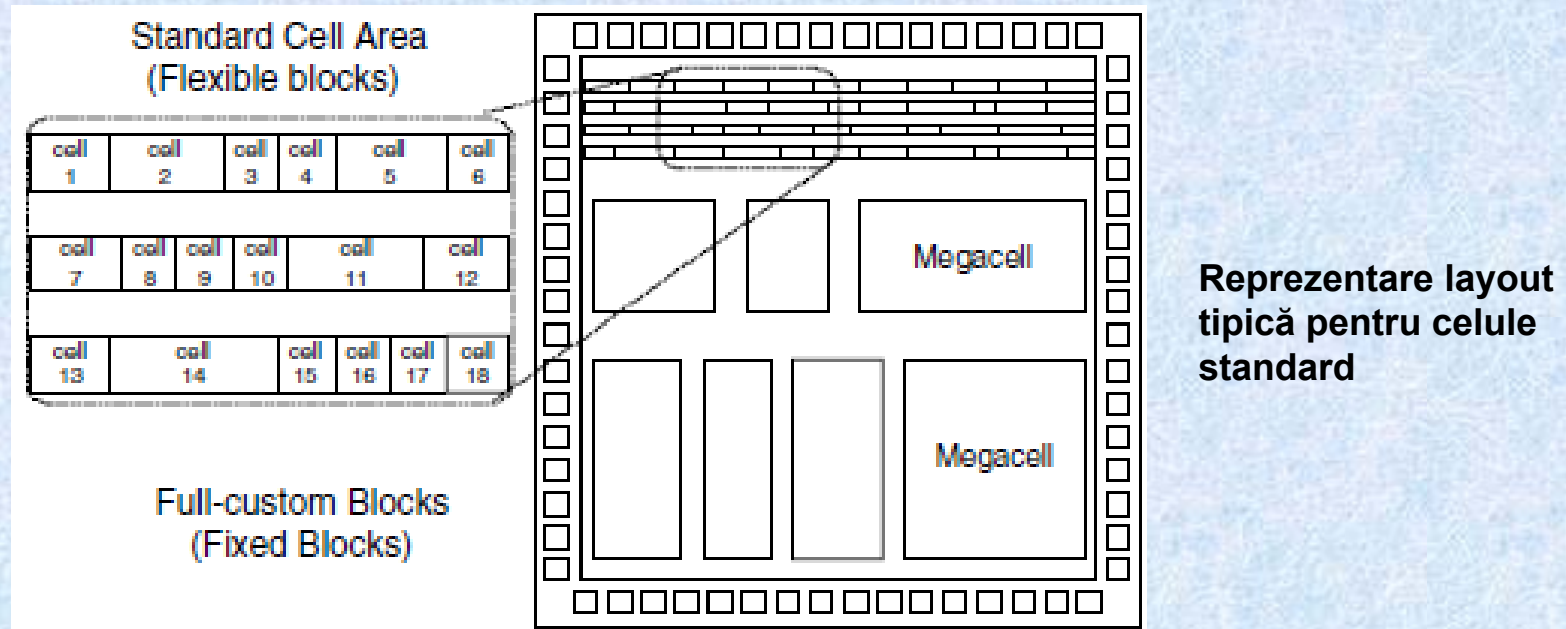


2.3 ASIC

2.3.2 Semi-custom ASIC

ASIC bazat pe celule standard

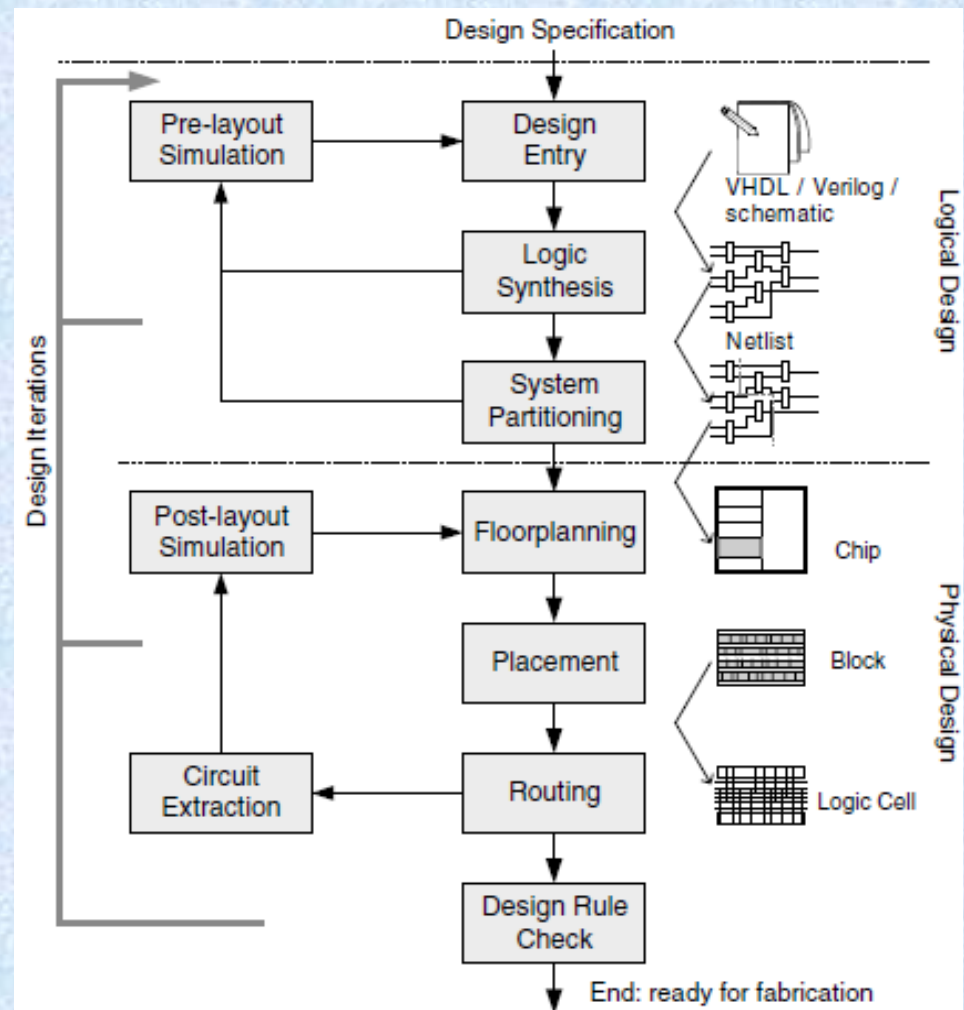
- Celulele standard sunt componente logice (porți, bistabile, etc) proiectate anterior și stocate într-o bibliotecă.
- Un proiect se creează folosind aceste celule ca intrări într-un sistem CAD: schemă logică sau descriere în limbaj de descriere hardware. Apoi, cu un alt program se convertește proiectul în layout.
- Celulele standard sunt pe cip aranjate pe rânduri de aceeași înălțime



2.3 ASIC

2.3.3 Etapele proiectării ASIC

- **Design Entry:** descriere proiect în HDL sau schematic
- **Sinteza logică:** Din HDL sau schematic se obține o descriere tip netlist constând din celule logice și interconectarea lor.
- **Partiționarea sistemului:** se împarte sistemul mare în blocuri
- **Simulare pre-layout:** verifică dacă circuitul sintetizat funcționează
- **Floorplanning:** se aranjează diferitele blocuri ale circuitului pe cip.
- **Plasare:** se stabilesc locațiile celulelor într-un bloc
- **Rutare:** se creează conexiunile dintre celule și blocuri
- **Extragere:** se extrag capacitățile și rezistențele interconexiunilor
- **Simulare post-layout:** simulare care ia în considerație întârzierile
- **Verificare reguli proiectare (DRC):** verifică dacă layoutul respectă regulile impuse de tehnologie





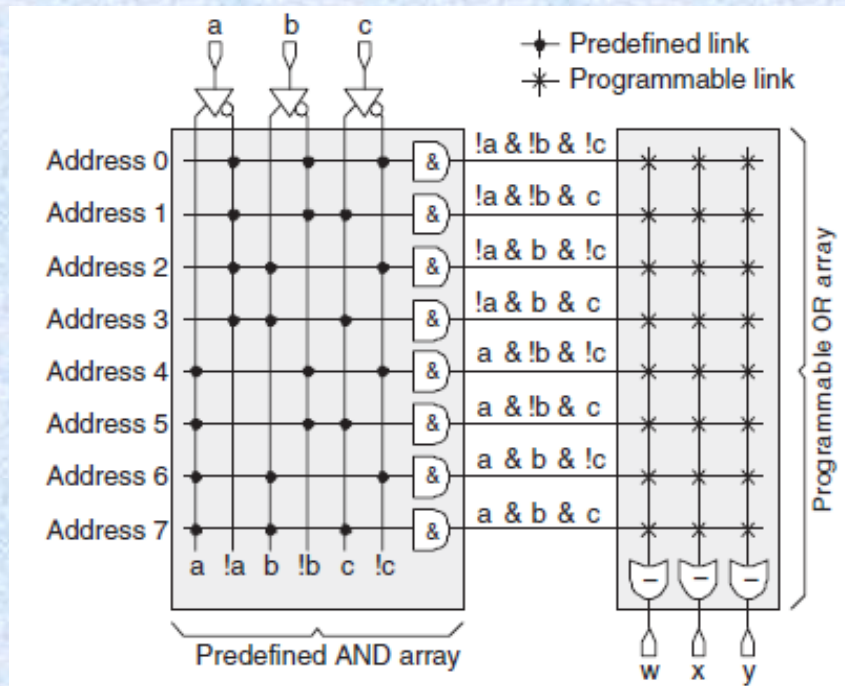
2.4 Dispozitive logice programabile

- Dispozitivele logice se pot împărți în două mari categorii:
 - **Fixe (ex: ASIC)**
 - **Programabile.**
- Dispozitivele logice programabile (PLD) sunt entități care pot fi modificate oricând pentru a efectua anumite funcții. Sunt bazate pe tehnologia memoriilor reinscriptibile, adică pentru a implementa un alt circuit dispozitivul se reprogramează.
- Există diverse tipuri de dispozitive logice programabile.
- În prezent, sunt două tipuri importante:
 - Dispozitive logice programabile complexe (CPLD)
 - Dispozitive programabile în câmp – Field Programmable Gate Arrays (FPGA)
- În continuare: evoluția structurilor programabile de la memorii ROM la CPLD și FPGA

2.4.1 PROM - Programmable ROM

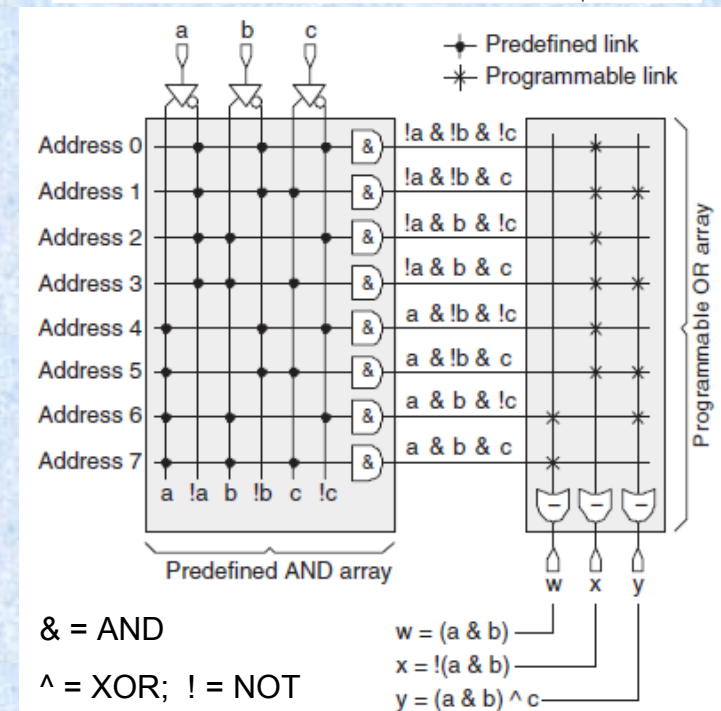
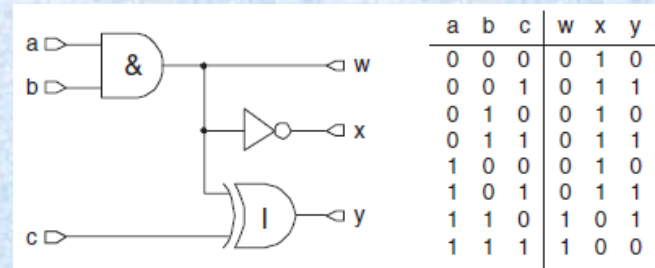
- Conține un plan fix de funcții AND și un plan de funcții OR programabile

Limitare: Suportă număr relativ mic de intrări deoarece toate sunt decodate în planul AND



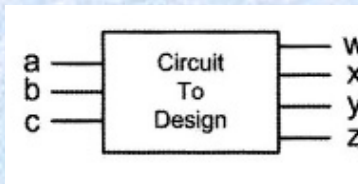
PROM cu 3 intrări și 3 ieșiri neprogramat

Exemplu implementare circuit cu PROM

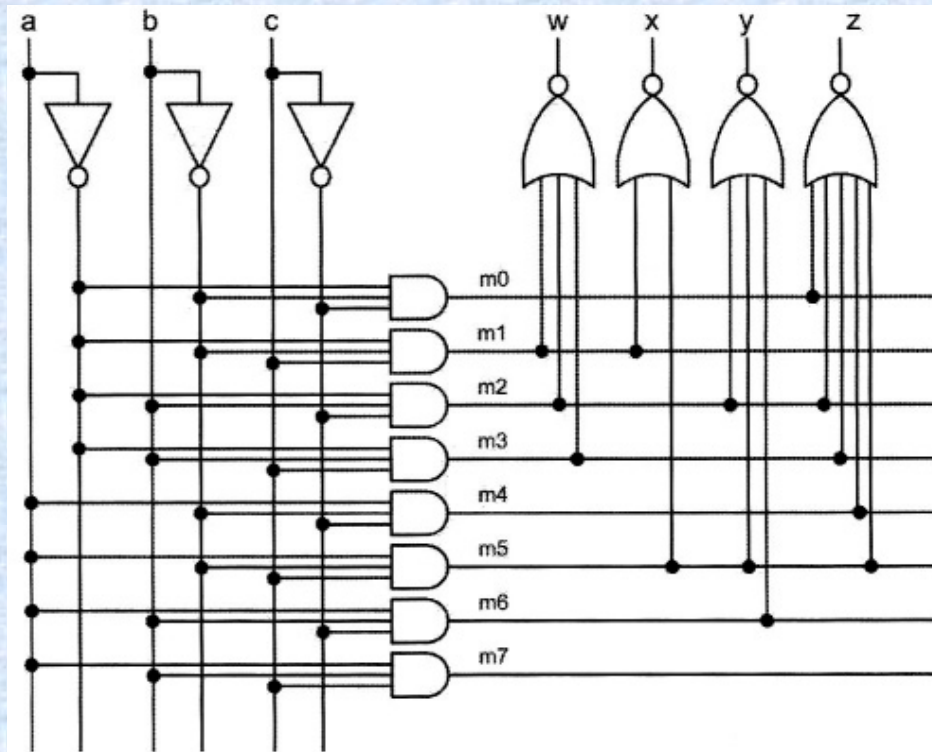


2.4.1 PROM - Programmable ROM

- Alt exemplu implementare circuit combinațional cu PROM

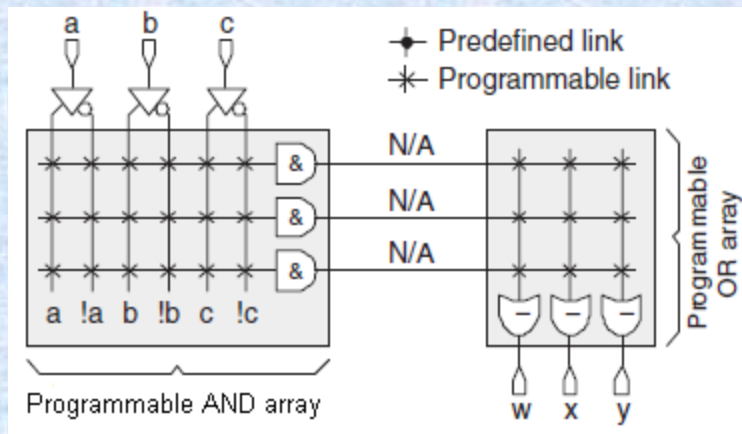


m:	a	b	c	w	x	y	z
0:	0	0	0	0	0	0	1
1:	0	0	1	1	1	0	0
2:	0	1	0	1	0	1	1
3:	0	1	1	1	0	0	1
4:	1	0	0	0	0	0	1
5:	1	0	1	0	1	1	1
6:	1	1	0	0	0	1	0
7:	1	1	1	0	0	0	0



2.4.2 PLA - Programmable Logic Arrays

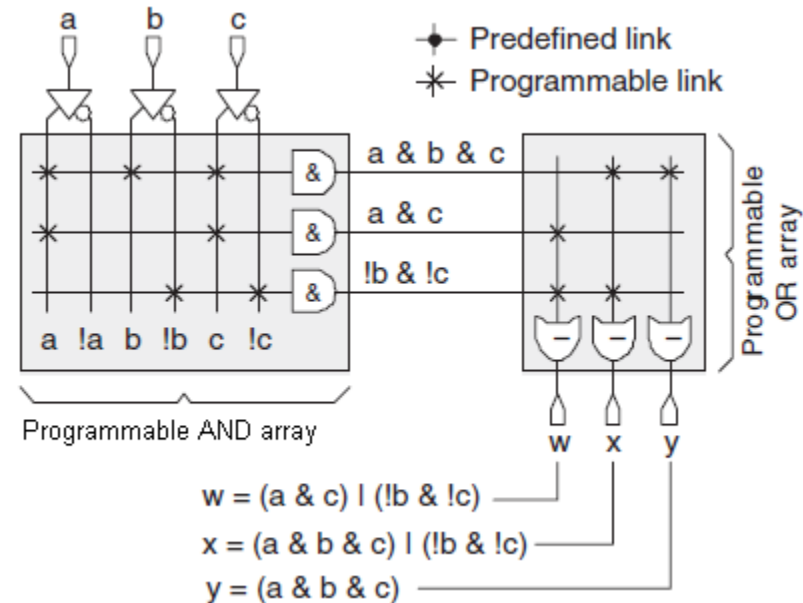
- Atât planul AND cât și planul OR sunt programabile



PLA-urile sunt mai lente decât PROM-urile deoarece ambele planuri AND și OR sunt programabile

Exemplu implementare circuit cu PLA

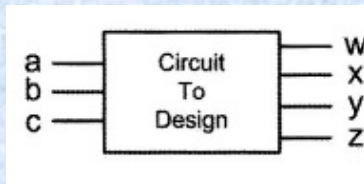
$$\begin{aligned} w &= (a \& c) \mid (!b \& !c) \\ x &= (a \& b \& c) \mid (!b \& !c) \\ y &= (a \& b \& c) \end{aligned}$$



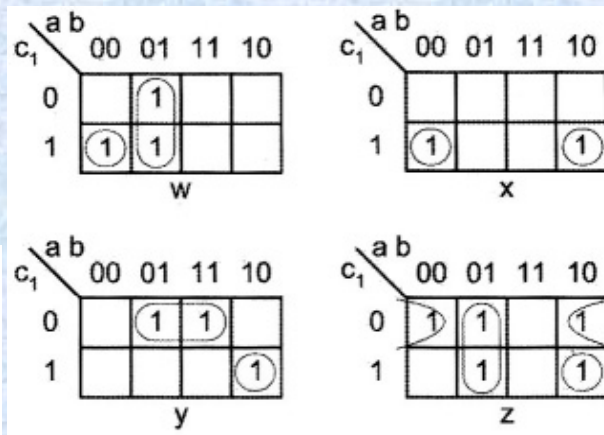
2.4.2 PLA - Programmable Logic Arrays

Alt exemplu: Implementare același circuit de la PROM

Minimizare



m:	a	b	c	w	x	y	z
0:	0	0	0	0	0	0	1
1:	0	0	1	1	1	0	0
2:	0	1	0	1	0	1	1
3:	0	1	1	1	0	0	1
4:	1	0	0	0	0	0	1
5:	1	0	1	0	1	1	1
6:	1	1	0	0	0	1	0
7:	1	1	1	0	0	0	0



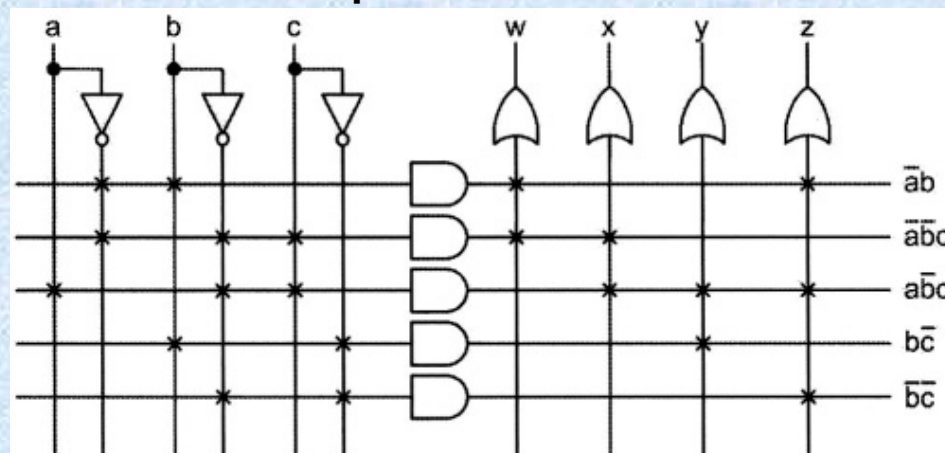
$$w = \bar{a}b + \bar{a}\bar{b}c$$

$$x = \bar{a}\bar{b}c + a\bar{b}c$$

$$y = a\bar{b}c + b\bar{c}$$

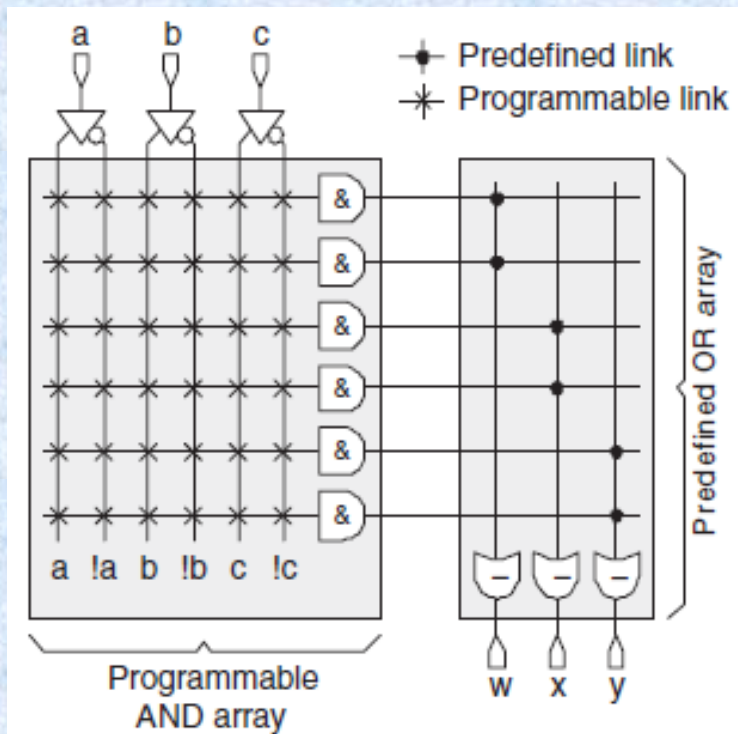
$$z = \bar{a}b + a\bar{b}c + \bar{b}\bar{c}$$

Implementare



2.4.3 PAL - Programmable Array Logic

- Este în opoziție față de structura PROM:
- Planul AND – programabil
- Planul OR – fix (predefinit)

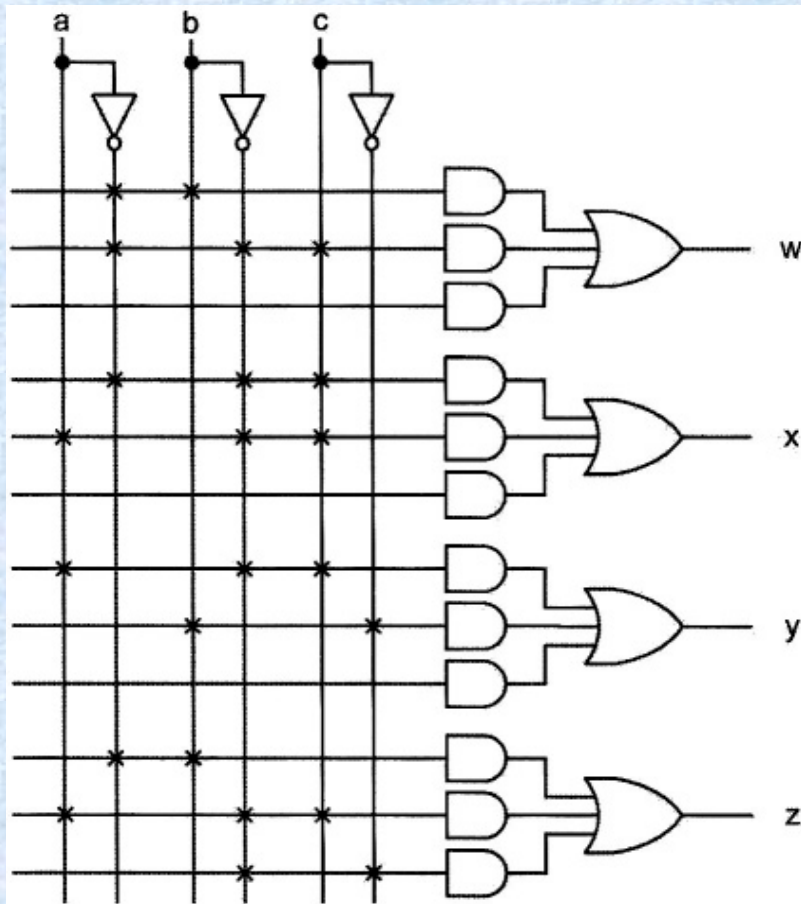


Rațiunea utilizării PAL este că, în general, o funcție logică mare utilizează un număr limitat de termeni produs și astfel nu este necesar să se utilizeze toți termenii produs pentru toate ieșirile.

Fixând numărul de termeni produs pentru ieșirile circuitului se îmbunătățește semnificativ viteza.

2.4.3 PAL - Programmable Array Logic

Exemplu: Implementare același circuit de la PROM și PLA



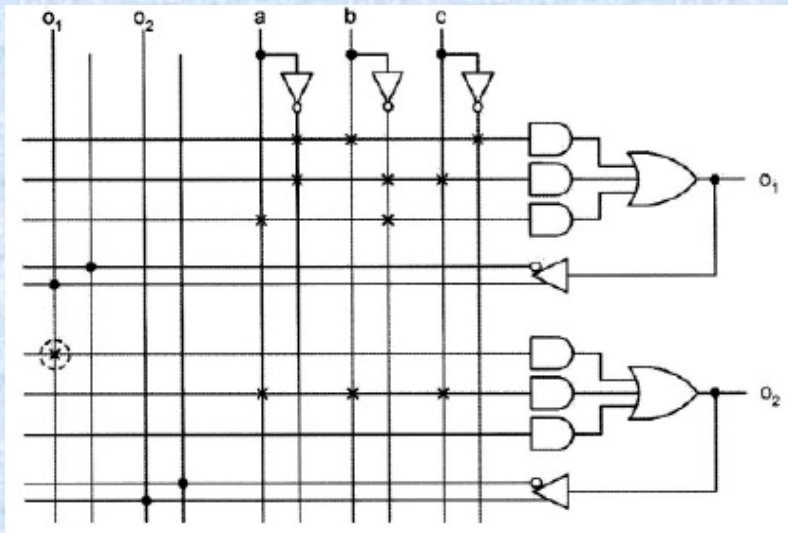
- În planul AND sunt formați câte trei termeni produs ce reprezintă intrări pentru planul OR
- Această structură permite un maxim de trei termeni produs pentru fiecare ieșire
- În exemplu, ieșirea z utilizează toți termenii produs disponibili în timp ce celelalte ieșiri utilizează numai câte doi termeni produs.

2.4.3 PAL - Programmable Array Logic

Îmbunătățiri ale structurii PAL

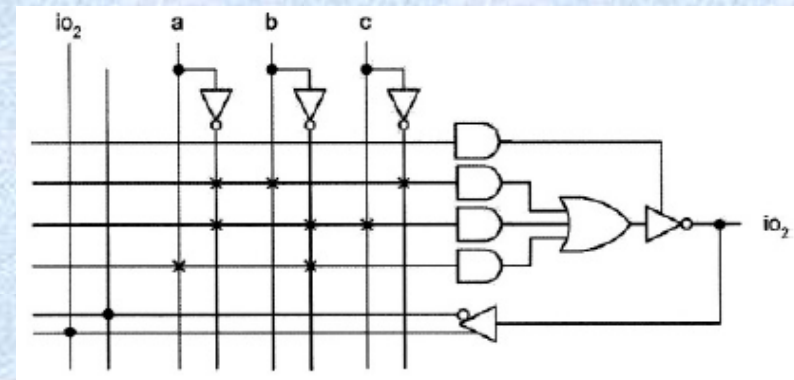
Adaugare reacție: Limitarea numărului de termeni produs per ieșire în PAL poate fi depășită prin adăugarea de reacții de la ieșiri la planul AND

$$w = \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + a \cdot \bar{b} + a \cdot b \cdot c$$

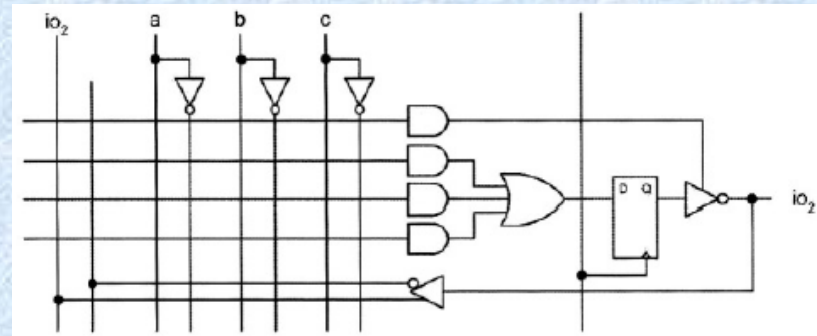


Funcția w este obținută la $o2$. Ieșirea $o1$ conține primii trei termeni. Prin reacție $o2 = o1 + abc$

Ieșiri cu trei stări (tri-state)

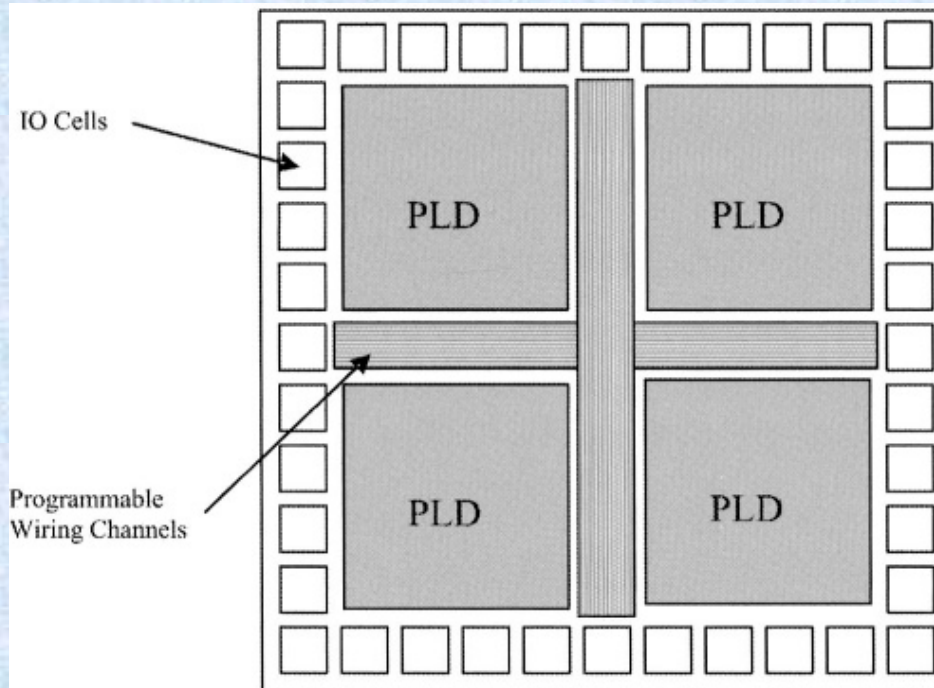


Posibilitate de adăugare regiștri la ieșiri (Avantaj față de PROM și PLA):



2.4.4 Dispozitive logice programabile complexe - CPLD

- Pentru a implementa circuite logice mai mari și complexe extinderea PLD-urilor prin arii mai mari ale planurilor AND și OR nu este o soluție.
- Soluția: utilizarea CPLD – dispozitiv care constă din mai multe PLD-uri între care există canale (matrice) de interconexiuni programabile



Structura CPLD

CPLD comerciale

Altera:

MAX5000, MAX7000, MAX9000

AMD

Familia Mach 1...5

Xilinx

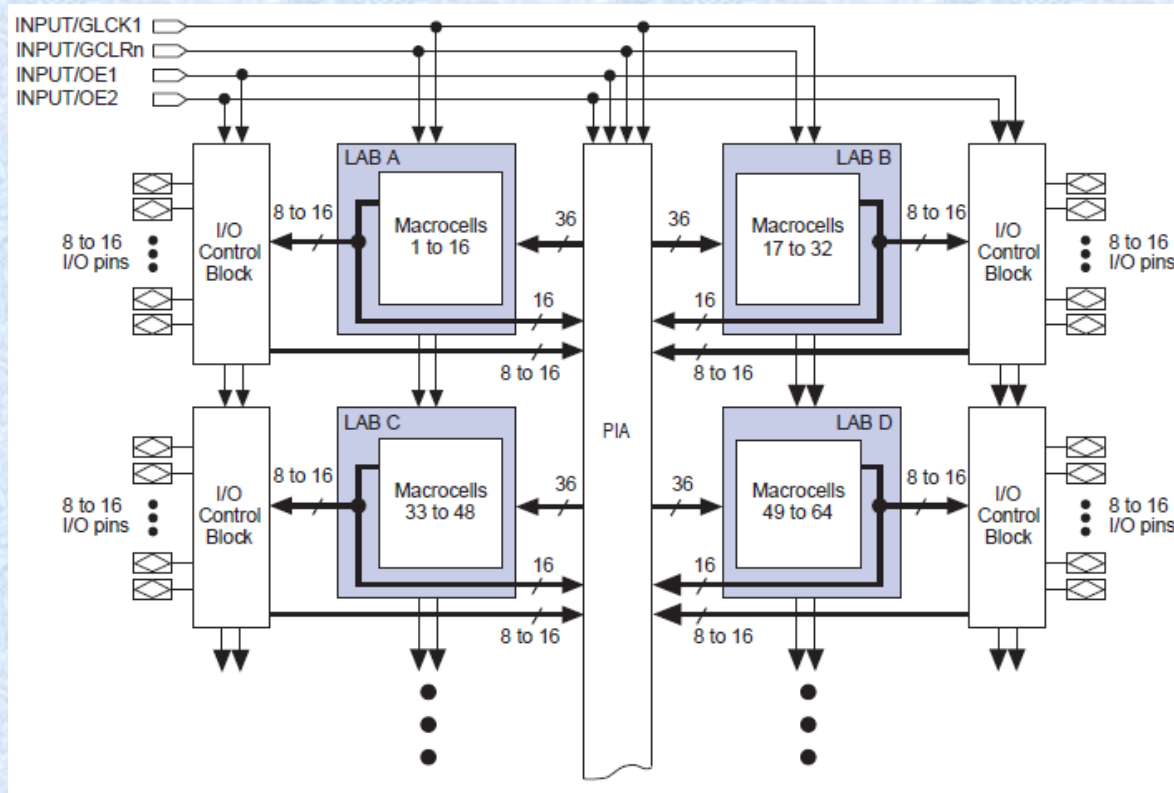
XC7000, XC9500

Lattice

Seriile pLSI și ispLSI

Exemplu: Structura CPLD MAX7000 Altera

Arhitectura generală a seriei MAX7000

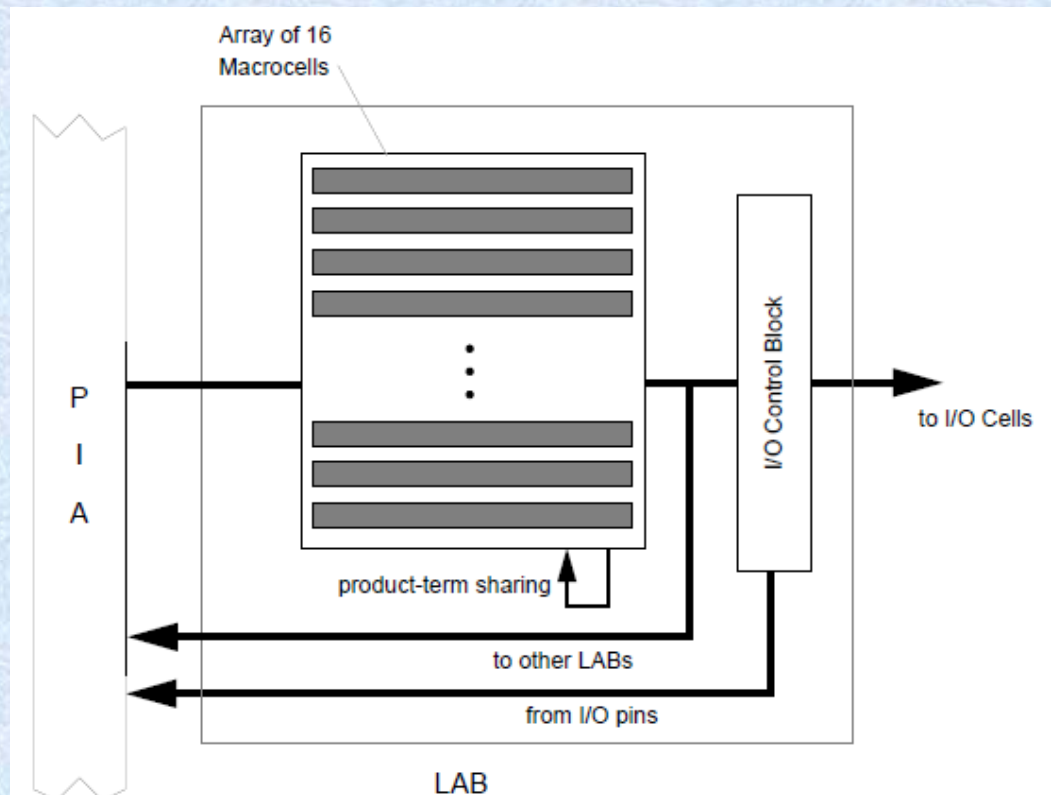


Conține

- o arie de 8 blocuri logice numite **LAB** (Logic Array Blocks)
 - Un bloc de interconexiuni programabile **PIA** (Programmable Interconnect Array)
 - Blocuri de intrări și ieșiri conectate direct la LAB-uri sau PIA.
- Un bloc LAB poate fi gândit ca o structură de tip PLD complexă. Astfel, întregul CPLD poate fi privit ca o arie de PLD-uri.

Exemplu: Structura CPLD MAX7000 Altera

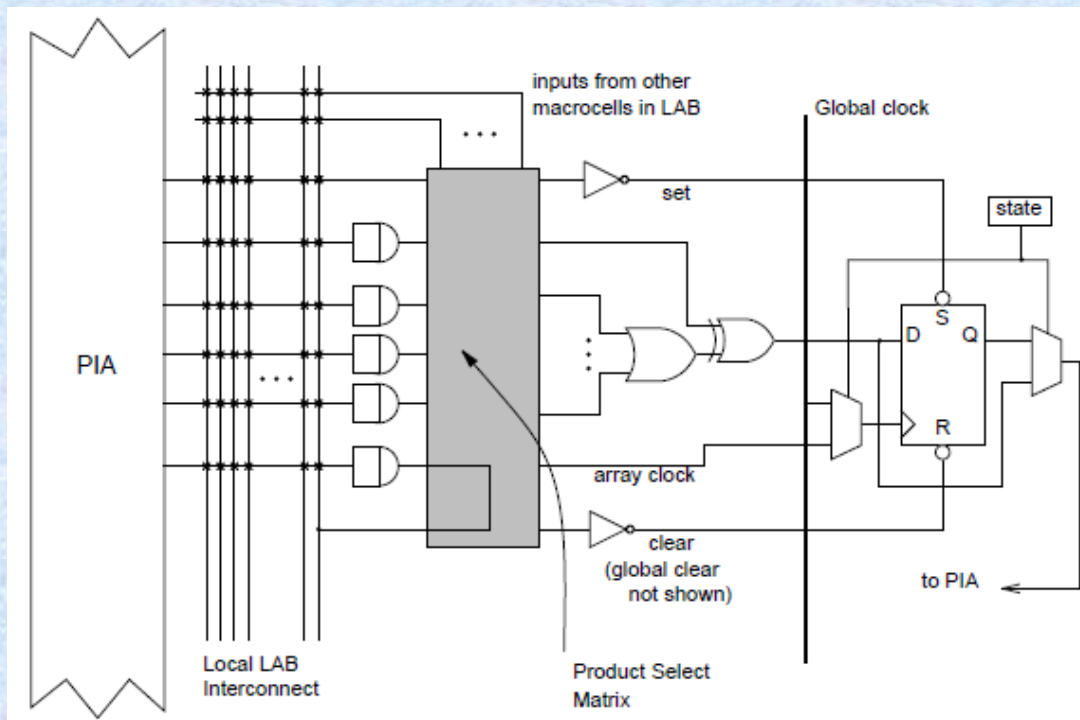
Altera MAX 7000 Logic Array Block (LAB).



- Un LAB conține 16 macrocelule
- Toate blocurile LAB sunt interconectate între ele prin intermediul ariei de interconexiuni programabile (PIA) – un bus global la care sunt conectate toate intrările dedicate, pinii de I/O și macrocelulele.
- Fiecare LAB este comandat de următoarele semnale:
 - 36 semnale de la PIA utilizate ca intrări logice generale
 - Intrări de control globale
 - Căi de intrare directe de la pinii I/O la regiștri.

Exemplu: Structura CPLD MAX7000 Altera

Macrocelula din CPLD MAX7000



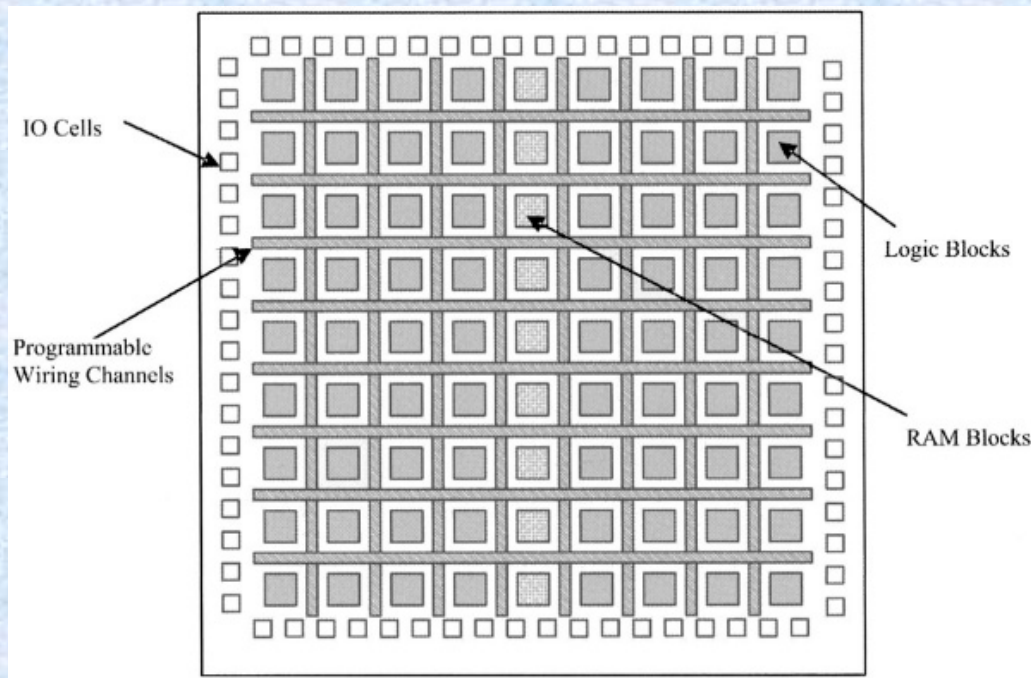
O macrocelulă conține un set de termeni produs programabili (o parte a unui plan AND) urmat de o poartă OR și un bistabil.

- Bistabilul poate fi configurat ca tip D, JK, T, SR sau transparent.
- Numărul de intrări pentru poarta OR este variabil: 1...5 termeni produs din macrocelulă + alte maxim 15 termeni de la alte macrocelule din același LAB
- Această flexibilitate face ca un LAB să fie foarte eficient dpdv al ariei ocupate

2.4.5 Field Programmable Gate Array – FPGA

Structura generală

Structura generală a unui dispozitiv FPGA

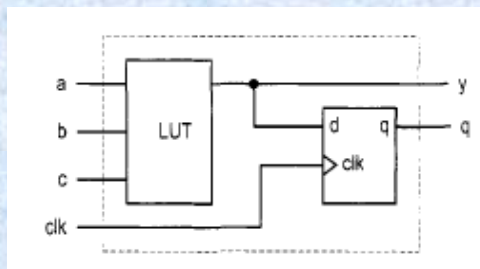


- Un dispozitiv FPGA conține la bază o arie de **blocuri logice configurabile** ce pot fi unite folosind o rețea de **canale** orizontale și verticale de **interconexiuni programabile**.
- De asemenea, structura FPGA poate conține **blocuri de memorie RAM** și alte **macro-blocuri** cu funcții definite (de ex: multiplicatoare, clock manager).
- Un bloc logic configurabil are complexitatea echivalentă a 10-20 de porți logice, conține un bistabil D și poate fi configurat pentru a efectua o anumită funcție.
- Blocurile RAM pot fi utilizate pentru implementare logică sau configurate pentru a forma memorii de diverse capacități.

2.4.5 Field Programmable Gate Array – FPGA

Blocul logic configurabil

Structura conceptuala a unui bloc logic din FPGA



Exemplu tabel adevar pentru LUT

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Un bloc logic conține un mic circuit combinațional configurabil și un bistabil D.
- Circuitul combinațional configurabil se implementează ca un tablou de valori - look-up table (LUT).
- Un LUT cu n intrări poate fi considerat ca o memorie cu 2^n locații pe un bit. Scriind corespunzător valorile din memorie, un LUT poate fi utilizat pentru a implementa orice funcție logică.
- Ieșirea de la LUT poate fi utilizată direct sau poate fi stocată în bistabil pentru a implementa circuite secvențiale.