INSTRUMENTE PENTRU PROIECTAREA CU CIRCUITE LOGICE PROGRAMABILE

PROIECTARE

Arhitecturi circuite programabile

- celule logice identice
- reţea de conexiuni posibile
- utilizarea acestor arhitecturi implică **translatarea funcţiilor dorite** de proiectant în **conexiuni intra şi intercelulare** pentru a obţine proiectul final

SINTEZĂ

- **Translatare** (compilare sau fitting - potrivire)
  - converteşte funcţiile proiectului în funcţii realizabile
- **Verificare**
  - verifică dacă proiectul translatat este corect

Netlist - listă de componente şi conexiuni

TRANSLATARE

- **Paşii procesului de translatare**
  - se verifică dacă numărul pinilor de intrare / ieşire este mai mare decât numărul pinilor necesari în proiect
  - se verifică dacă există suficiente celule pentru a acoperi numărul de porţi logice din proiect
  - se caută în proiect grupuri de componente al căror număr de intrări şi de ieşiri este egal cu cel al celulei logice de bază
  - se aleg întâi grupurile cele mai mari care pot fi formate
- **Realizarea translatării**
  - set de **reguli** pt. substituţiile de funcţii
    - set de substituţii pt. porţi logice
  - **strategie** pt. **utilizarea optimă** a regulilor de substituţie

IMPLEMENTARE

- **Etape**
  - maparea tehnologică - inclusiv optimizarea
  - plasarea
  - rutarea

MAPARE

- **Optimizare netlistă**
  - etapă esenţială
  - metode clasice
    - teoremele algebrei booleene
    - metode algebrice de minimizare
    - diagrame Karnaugh
- **Netlist**
  - fişier text
    - funcţiile logice din proiect
    - conexiunile de intrare / ieşire
  - aici se fac substituţiile de funcţii
    - există sute de reguli

MAPARE

- Netlist

```
*
NETSTART
*
OUT3 DFF I(OUT3_D, CLOCK) O(OUT3, N_OUT3)
OUT3_D AN2 I(A1, N2O_1) O(OUT3_D)
B2O_1 INV I(A2) O(N2O_1)
OUT2_D OR2 I(N21_2, N21_4) O(OUT2_D)
OUT2 DFF I(OUT2_D, CLOCK) O(OUT2, N_OUT2)
B21_4 AN2 I(N21_3, A4) O(N21_4)
B21_2 AN2 I(N21_1, A2) O(N21_2)
B21_1 INV I(A1) O(N21_1)
B21_3 INV I(A3) O(N21_3)
OUT1 DFF I(OUT1_D, CLOCK) O(OUT1, N_OUT1)
B22_1 AN2 I(A1, A2) O(N22_1)
B22_2 AN2 I(A3, A4) O(N22_2)
OUT1_D OR2 I(N22_1, N22_2) O(OUT1_D)
*
NETEND
*
NETIN A1, A2, A3, A4, CLOCK
NETOUT OUT1, OUT2, OUT3
*
```

PLASARE

- **Plasare**
  - proces de **amplasare fizică** a componentelor logice în celulele logice din circuitele programabile
  - se urmăreşte amplasarea funcţiilor logice adiacente în celule alăturate
  - criteriul critic de plasare - realizarea legăturilor între celule

RUTARE

- **Interconectarea**
  - după plasarea adecvată
  - inspectarea netlistei şi a plasării curente
  - **linii de conectare "consumate"** - la un capăt au un semnal de ieşire şi la celălalt capăt un semnal de intrare
  - **congestie** - nu mai sunt linii de conectare disponibile
  - **re-rutare** - reluarea procesului de plasare şi rutare
  - **rezultat** $\rightarrow$ fişier care descrie proiectul iniţial în termenii celulelor circuitelor programabile
  - fişierul descrie asignarea poziţiilor celulelor şi interconexiunile dintre celule
  - fişierul $\rightarrow$ translatat într-o **hartă de biţi** care va fi transmisă unui programator de circuite programabile $\Rightarrow$ **configurarea**

CĂI CRITICE

- **Probleme de temporizare**
  - apar **căi critice**
    - sporesc numărul de constrângeri adiţionale
    - complică plasarea şi rutarea
  - de obicei se plasează celulele cu căi critice şi apoi celelalte celule din proiect
  - rutarea începe cu celulele de pe căi critice
  - se recomandă un număr minim de căi critice în proiecte, pentru a scurta sesiunile de plasare şi rutare

VERIFICARE

Instrumente de verificare

- programe
  - examinează netlista transformată
  - analizează proprietăţile proiectului iniţial
- verificarea se poate face în timpul translatării sau al simulării
- **exemple de verificări**
  - se izolează ieşirea unei celule şi se numără la intrarea câtor altor celule este transmis acel semnal; fiecare celulă în care intră semnalul contribuie la o sarcină cumulativă a acestuia - programul intervine, distribuind sarcina respectivă la mai multe celule de ieşire identice
  - se caută intrări ale celulelor lăsate neconectate, care creează probleme de zgomote
  - se caută ieşiri care nu sunt three-state şi sunt totuşi legate împreună

SIMULARE

Simulare logică

- instrument de verificare
  - funcţională
  - a performanţelor temporale
- se creează un **model** al reţelei logice → i se aplică un model al intrărilor (numite **stimuli**) → se obţine un model al ieşirilor (numite **răspunsuri**)
- permite observarea răspunsurilor logice interne (inaccesibile la pinii exteriori)

SIMULARE

Tipuri de simulare

- funcţională
- temporală digitală
- a defectelor

SIMULARE

Simulare funcţională

- se modelează celulele logice
- se combină cu un model al intrărilor binare (tensiuni)
- se generează un model relativ de răspunsuri (tensiuni)
- **avantaje**:
  - modelul este simplu
  - generează rapid rezultatele
- **dezavantaje**:
  - furnizează numai relaţii relative între semnale
  - nu oferă informaţii despre temporizare

SIMULARE

Simulare temporală

- se creează un model al întregului proiect
  - se conectează modelele celulelor
  - fiecare model de celulă are la ieşire un **bloc de întârziere**
- **blocul de întârziere**
  - întârzierea introdusă de celulă în sine, fără conexiuni externe
    - întârzierea pentru tranziţia din "1" logic în "0" logic
    - întârzierea pentru tranziţia din ”0" logic în ”1" logic
  - întârzierea asociată capacităţii de rutare a firului de metal care uneşte 2 celule
  - întârzierea dată de suma impedanţelor de intrare ale celulelor comandate
- **planificator de evenimente**

SIMULARE

Simularea defectelor

- se folosesc tehnici speciale pt. a se obţine un scor = **grad de defecte**, pentru proiect şi stimulii aplicaţi
- se testează fiecare aspect al proiectului conform standardelor industriale
- se creează probleme artificiale (defecte) şi se urmăreşte dacă răspunsurile diferă de cele obţinute în cazul normal (corect)
- simulare fundamentală
  - menţinerea ieşirilor unor celule la 0 sau la 1 logic, în timp ce se aplică stimulii
  - procesul nu este efectuat pentru toate nodurile deodată
  - poate fi realizat pentru mai multe noduri simultan, dacă acestea sunt suficient de independente unul faţă de celălalt
  - necesită rulări multiple pentru a calcula un grad de defecte
  - este o mare consumatoare de timp

SIMULARE

Simulare

- depinde de tipurile de circuite programabile
  - FPGA - se foloseşte mai ales simularea temporală
  - CPLD-uri - se preferă simularea funcţională
  - ASIC-uri mari (circuite integrate specifice aplicaţiei) - de regulă se face simularea de defecte

SIMULARE

Evaluare

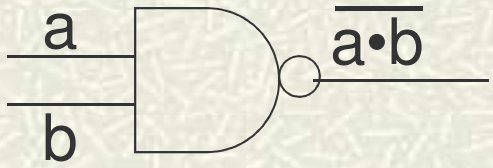
- simulatorul are o **bibliotecă de funcţii**
- fiecare intrare are ataşată o rutină care evaluează un tabel de adevăr
- intrările într-un **tabel de adevăr de simulator** respectă mai realist condiţiile electrice (ieşiri 3-state, intrări necunoscute etc.)
- cu cât numărul de intrări din simulator creşte, cu atât creşte şi acurateţea simulării, dar cu preţul unui consum sporit de timp de rulare

SIMULARE

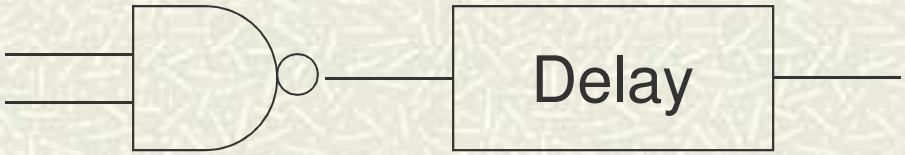
Modelul de simulare a unei porţi ŞI-NU

ŞI-NU	∅	1	*	3	#
∅	1	1	1	1	1
1	1	∅	*	#	#
*	1	*	*	#	#
3	1	#	#	#	#
#	1	#	#	#	#

∅ = ZERO
1 = HIGH
* = NECUNOSCUT
3 = TRI-STATE
# = NEDETERMINAT



Modelul logic



Modelul cu o întârziere adăugată

SIMULARE

Simulare poartă ŞI-NU

- o poartă ŞI-NU cu 2 intrări are în tabelul de adevăr doar patru combinaţii de intrări
- în simulator fiecare intrare poate lua 5 valori diferite
- tabelul de adevăr din simulator are 25 de combinaţii de intrări
- * = necunoscut, poate să apară într-o situaţie reală, dacă intrarea porţii provine de la un bistabil a cărui ieşire Q este necunoscută (de exemplu la punerea sub tensiune)
- 3 = 3-state, poate să apară la o intrare dacă aceasta provine de la o ieşire 3-state de pe un nivel logic anterior
- # = nedeterminat, poate să apară dacă două buffere 3-state comandă acelaşi nod şi ambele sunt în perioada de tranziţie

SIMULARE

Modelare

- simularea performantă include folosirea unor **limbaje de modelare** speciale = BLM (Behavioral Language Models)
- BLM
  - foarte eficient
  - modele de evaluare, care generează răspunsuri de ieşire corecte, la stimuli de intrare daţi
  - alcătuit din proceduri scrise pentru a reacţiona corect la un stimul, dar nu conţine vreun model de poartă anume
  - verificări interne pentru violări de timpi de setup şi de hold în interiorul rutinei care evaluează operaţia logică respectivă
  - evaluează rapid, dar nu arată operaţiile interne - rămâne ca o cutie neagră
  - utilizate la funcţii mari

SIMULARE

Modelare

- exemplu de BLM simplu - bistabil D
- cod scris în VHDL (Limbaj de Descriere Hardware)

```
EDGE_TRIGGERED_D: block (CLK = '1' and not CLK 'STABLE or CLR = '1')
begin
    Q <= guarded '0' when (CLR = '1')
    else D when (CLK = '1' and not CLK 'STABLE)
    else Q;
end block EDGE_TRIGGERED_D;
```

SIMULARE

Modelare

- bistabilul D - poate fi modelat şi în **versiunea cu porţi**
  - versiunea trebuie să aibă ataşată o funcţie externă, pentru a răspunde necesităţilor de verificare a timpilor de setup şi de hold ai modelului
  - face o disecţie completă a operaţiilor modelului, necesitând un timp de simulare mai mare
  - permite o evaluare completă a gradului de defecte

SIMULARE

Biblioteci

- simulatoarele sunt livrate cu biblioteci de modele de simulare
- conţin blocurile constructive necesare gestionării celei mai mari părţi a proiectelor
- includ porţi elementare, bistabile, intrări, ieşiri şi mai multe funcţii adiţionale
- se pot modifica întârzierile nodurilor interne, se pot rula subrutine ale stimulilor etc.

SIMULARE

Back Annotation

- Back Annotation = procesul de **alterare a întârzierilor** nodurilor interne
- modelul de simulare iniţial = netlista
- înainte de plasarea şi rutarea proiectului, întârzierile sunt necunoscute $\rightarrow$ după plasare şi rutare, întârzierile devin precizate
- se calculează întârzierile interne folosind legile circuitului electric, lungimea firelor metalice, constantele dielectrice şi alţi parametri
- cu aceste întârzieri $\Rightarrow$ o **netlistă nouă** cu întârzieri notate nod cu nod
- simularea pe baza noii netliste este mult mai precisă, detectând puncte critice care nu au cum să fie observate din exterior

PROGRAMARE (CONFIGURARE)

Programarea = configurarea circuitelor

- controlează circuitele programabile la nivel de **memorie** şi de **porţi logice**
- se urmăreşte **procesarea paralelă**, simultană a porţilor logice
- **optimizarea** este:
  - în spaţiu - nr. mic de porţi logice
  - în timp = întârzieri → viteză mare de execuţie

CONFIGURARE

Configurarea

- **etape**
  - iniţializare
  - ştergere memorie de configurare
  - încărcare date de configurare
  - start-up
- **control sincron** - Clock de configurare (CCLK)

CONFIGURARE

Configurarea

- **iniţializare**
  - aplicare alimentare
  - validare pin de intrare $\overline{\text{PROGRAM}}$ la 0 logic
  - circuitul setează ieşirea DONE la 0 logic
- **ştergere memorie de configurare**
  - ştergerea în progres e indicată prin $\overline{\text{INIT}}$ (bidirecţional) la 0 logic
  - terminarea ştergerii setează $\overline{\text{INIT}}$ la 1 logic
- **încărcare date de configurare**
  - încărcarea datelor specifice aplicaţiei pentru:
    - a defini operaţiile funcţionale ale blocurilor interne circuitelor programabile
    - a realiza legăturile între blocuri
  - între 54K şi 4,3G biţi de configurare
  - unele dintre variantele FPGA au posibilitatea de criptare / decriptare

CONFIGURARE

Configurarea

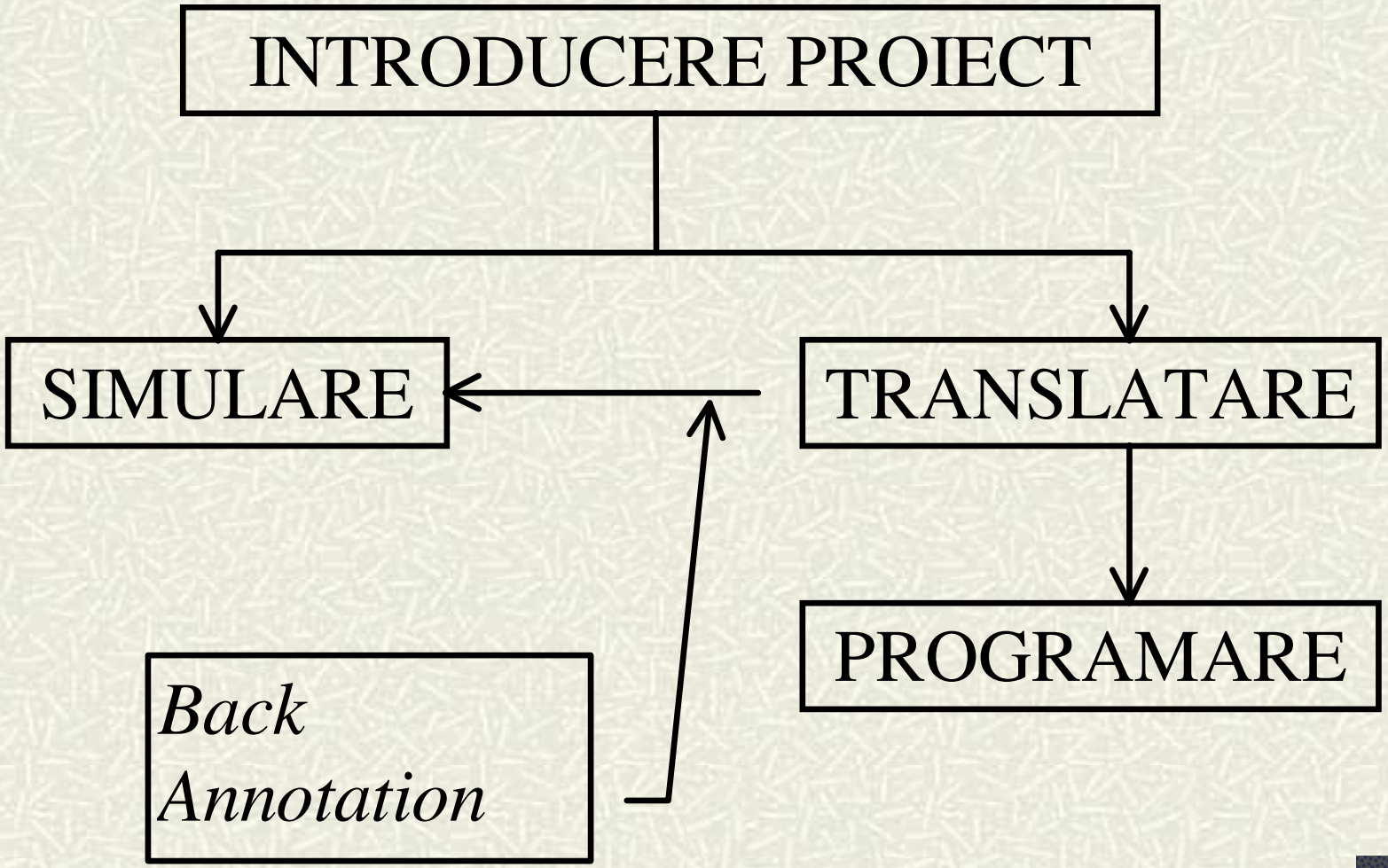
- **încărcare date de configurare**
  - modul de încărcare stabilit cu 3 pini dedicaţi: M_2, M_1, M_0
  - Slave Serial Mode
  - Master Serial Mode
  - Slave SelectMap Mode - paralel
  - Master SelectMap Mode - paralel
  - Boundary Scan (JTAG, IEEE 1532) Mode
    - (IEEE = Institute of Electrical and Electronics Engineers = I-triple-E)

CONFIGURARE

Configurarea

- **start-up**
  - iniţiat de o **verificare CRC** (Cyclic Redundancy Check) corectă (încărcare corectă a datelor de configurare)
  - **4 operaţii** care durează **8 cicluri de CCLK**
    - DONE se setează pe 1 logic
    - GTS (Global Three State) activează toate I/O
    - GST (Global Set/Reset) permite bistabililor să-şi schimbe starea
    - GWE (Global Write Enable) permite ca memoriile şi bistabilii să-şi schimbe starea

Procesul de proiectare



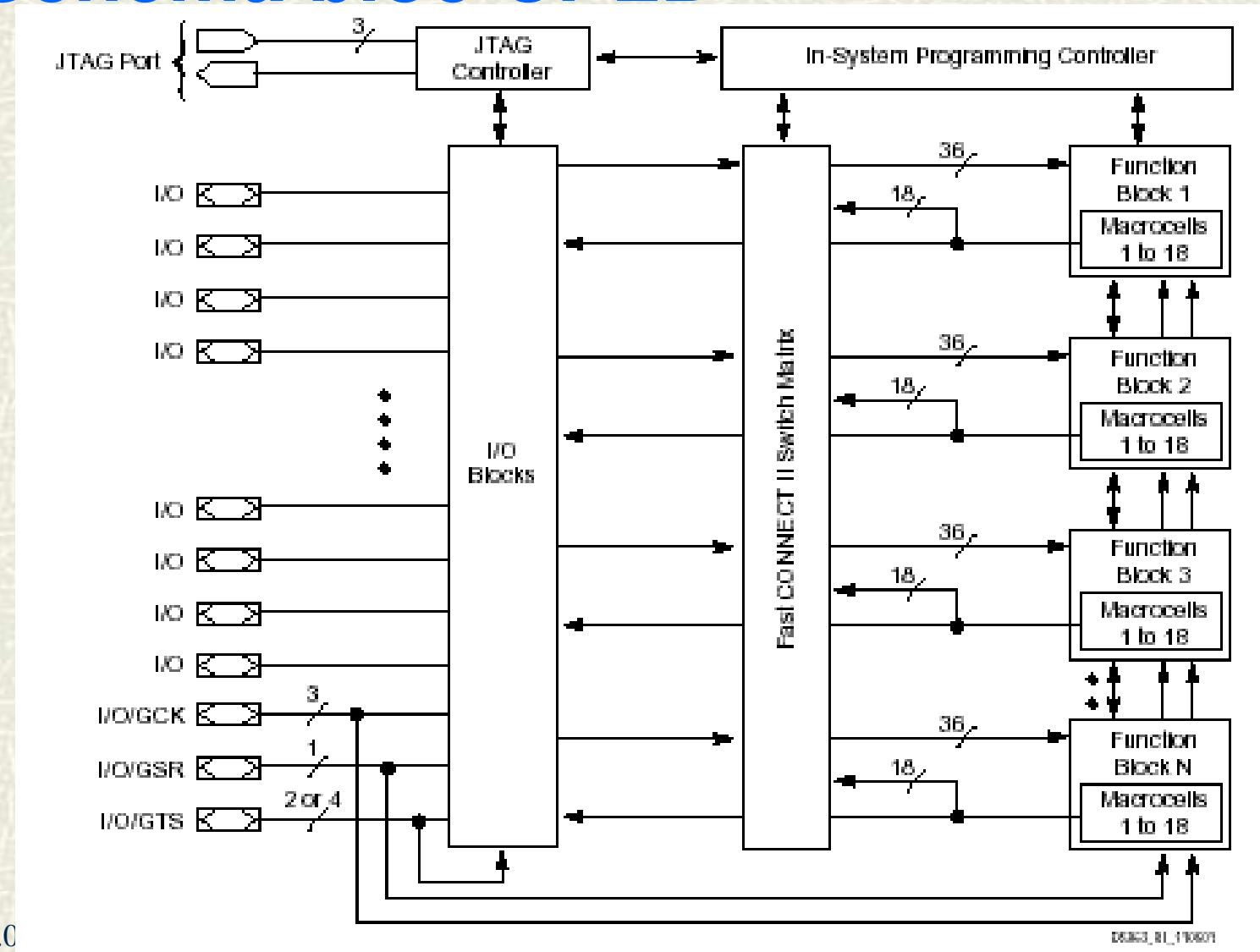
CIRCUITE LOGICE PROGRAMABILE

CPLD = Complex PLD

- **CPLD** = colecţie de **PLD**-uri individuale plasate într-o **structură de interconectare programabilă** - arhitectură foldback
- **PLD**
  - un nivel de porţi ŞI
  - blocuri logice (macrocelule)

CPLD - seria 9500

- Schema bloc CPLD



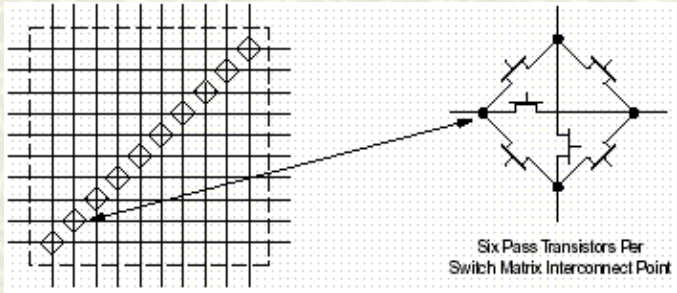
FPGA - seria 4000

- **CLB** - **Configurable Logic Block** - celula logică programabilă din LCA
  - **3 generatoare (F, G, H) de funcţii logice combinaţionale** - realizate cu memorie SRAM
  - **2 bistabili D**, care comută pe front (la unele componente din familie pot fi configuraţi ca latch-uri)
  - mai multe căi de multiplexare şi **interconectare locală** programabile
  - **13 intrări** şi **4 ieşiri**

FPGA - seria 4000

Arhitectura de interconectare

- fire metalice cu **puncte de conexiuni programabile**
- matrice de conectare



- memorie de interconectare SRAM - fiecare bit este dedicat controlului unui punct de conectare intern (**PIP** – Programmable Interconnection Point)
- blocurile de I/O au un inel de rutare (**VersaRing**) în jurul matricei CLB-urilor

FPGA - seria 4000

- **Alimentarea FPGA**
  - distribuită ca o **reţea**
    - scădere a zgomotelor
    - izolare a I/O de partea de logică
  - inele separate de Vcc şi GND pentru driverele de I/O
  - decuplarea fiecărui pin de Vcc la planul de masă (GND) cu condensatoare de 1µF

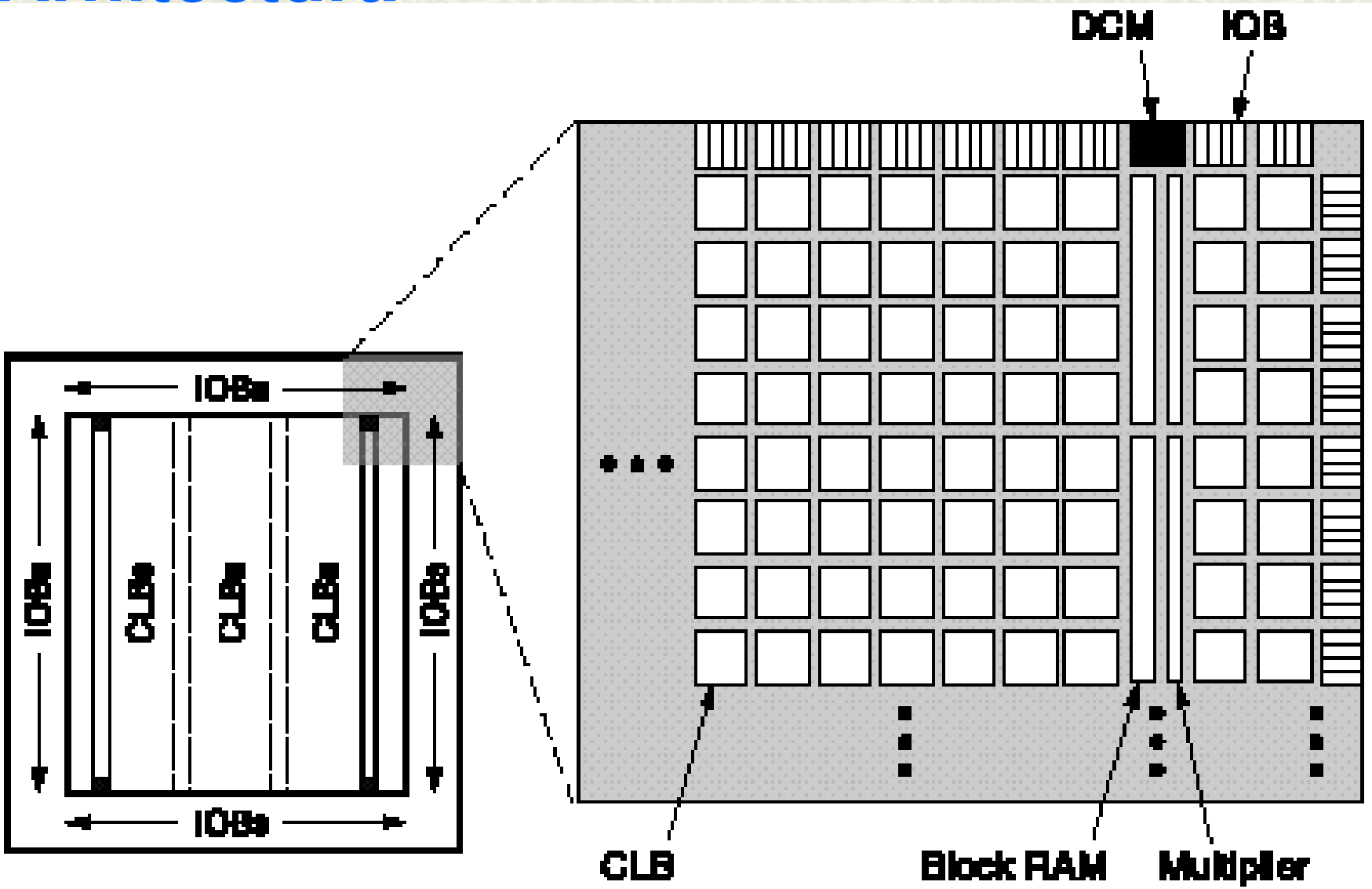
FPGA - Spartan

Caracteristici generale

- în evoluţie au fost create mai multe variante: Spartan: Spartan-II 2,5V, Spartan-IIE 1,8V, Spartan-3, Spartan-3E, Spartan-3A, Spartan-3AN, Spartan-6
- număr de porţi logice: de la 40.000 la 1.400.000
- frecvenţă: de la 80 MHz la 450 MHz
- tehnologie de la 0,18 microni la 45nm
- compatibilitate PCI
- versiuni la diverse tensiuni de alimentare
- **LUT** (Look-Up Table) - memorie RAM - pt. funcţiile combinaţionale din CLB
- blocuri separate de memorie
- DCM - control pentru Clock
- opţiuni 3-state pt. bus intern

FPGA - Spartan 3

Arhitectura



FPGA - Virtex

Familia Virtex

- mai multe variante:
  - Virtex 2,5V
  - Virtex E 1,8V
  - Virtex II
  - Virtex IIPro şi IIProX
  - Virtex 4
  - Virtex 5
  - Virtex 6
  - Virtex 7

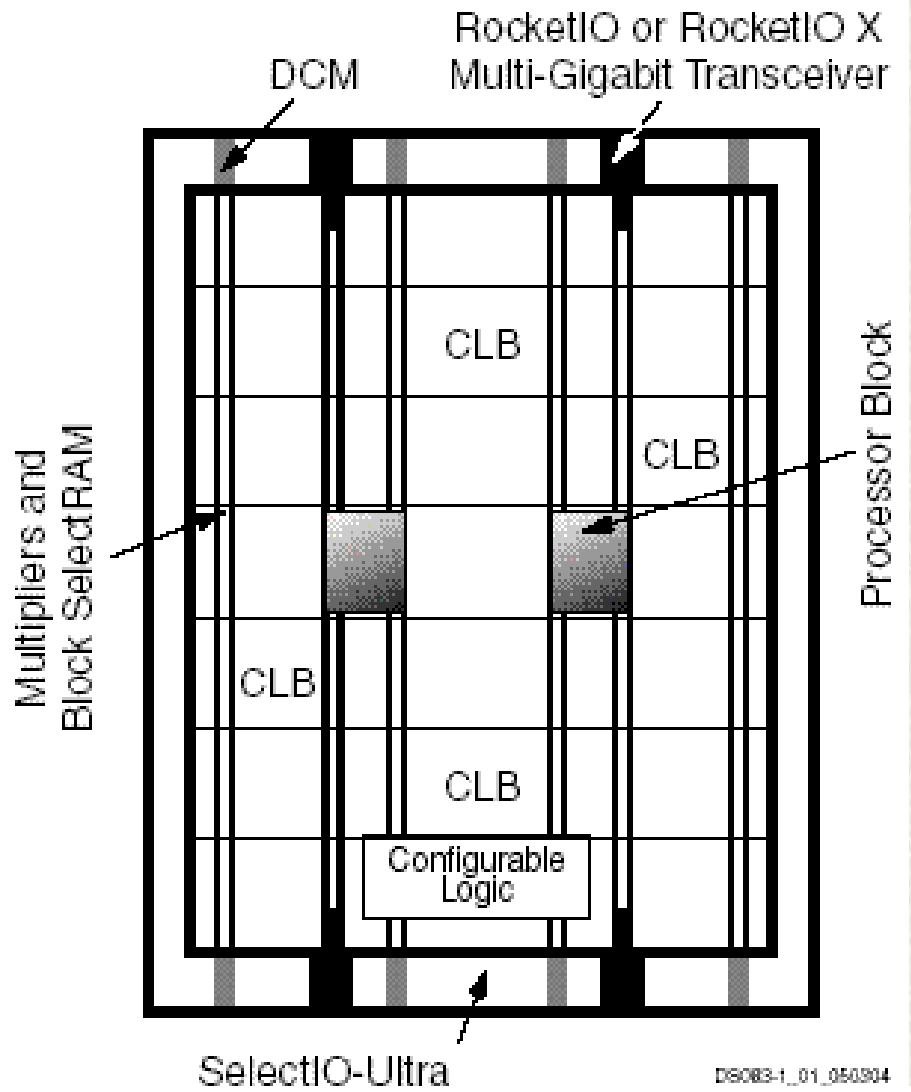
FPGA - Virtex IIPro

Caracteristici - îmbunătăţiri

- prima familie de FPGA-uri care încorporează:
  - Multi Gigabit Transceiver (MGT) – RocketIO (3,125 Gb/s) sau RocketIO X (6,25 Gb/s)
  - procesor - IBM PowerPC 405 RISC CPU (405PPC)
    - poate executa instrucţiuni cu frecvenţa de 1 instrucţiune / ciclu
    - implementează User Instruction Set Architecture (UISA) ale PowerPC şi extensii pt. aplicaţii
    - 32 registre generale de 32 biţi (GPR)
    - predicţie statică a ramificaţiilor
    - majoritatea instrucţiunilor - 5 stagii pipeline; execuţie într-un singur ciclu
    - multiplicatoare / împărţitoare hardware pt. aritmetică binară (4 cicluri înmulţirea, 35 cicluri împărţirea)
- tehnologie: 13 µm, 9 nivele, tranzistori de mare viteză de 90 nm
- Vcc_{IN} = 1,5V; Vcc_{AUX} = 2,5V

FPGA - Virtex IIPro

Arhitectura



FPGA - Virtex 7

Caracteristici

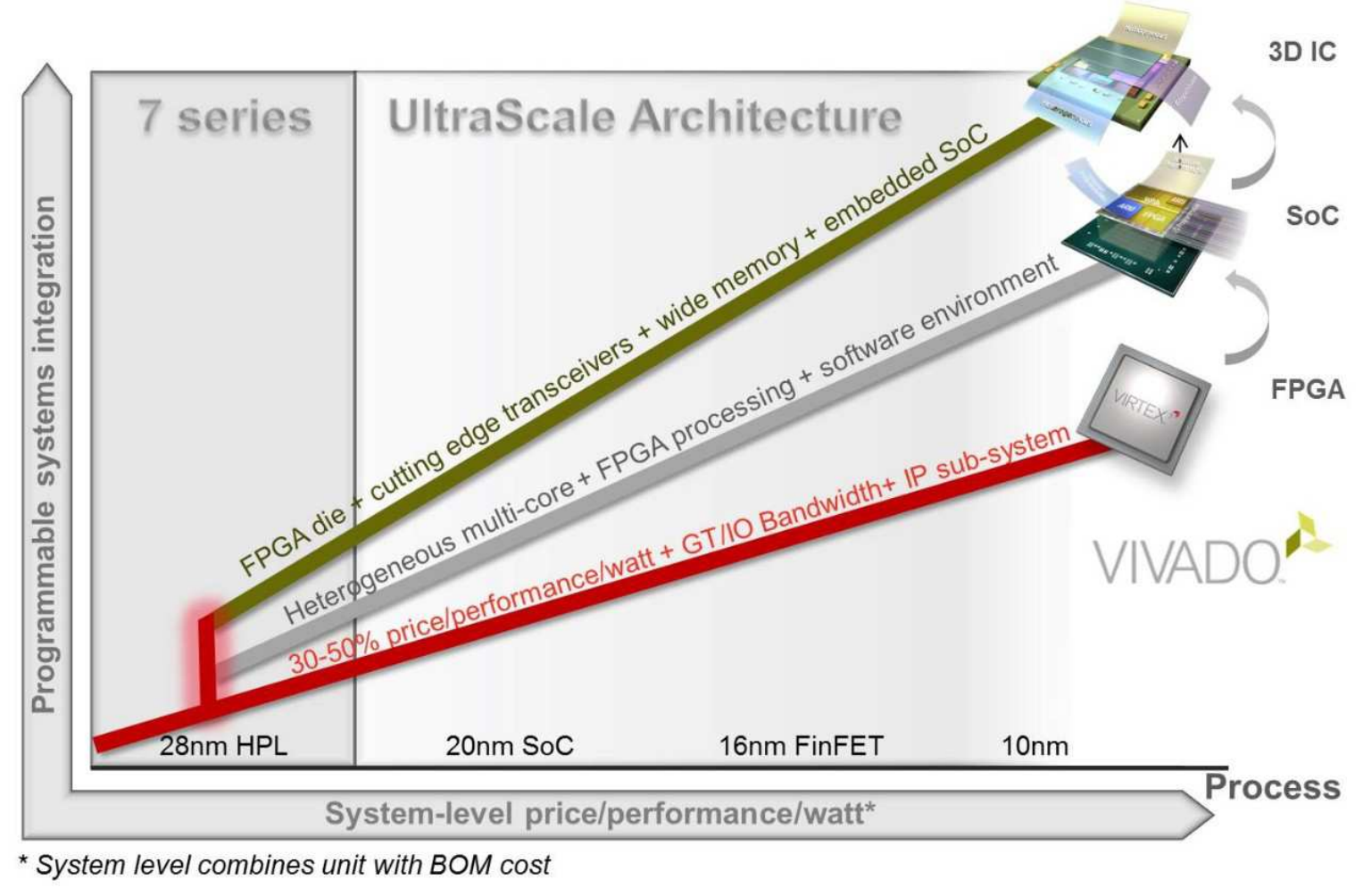
- tehnologie: 28 nm CMOS, 1Vcc (opţiune 0,9Vcc)
- foloseşte arhitectura ASMBL (Advanced Silicon Modular Block) - **aranjare pe coloane**
- slice
  - 4 LUT cu 6 intrări
    - opţiune dual LUT cu 5 intrări
    - folosite ca RAM de 64 biţi
    - folosite ca registre de deplasare stânga-dreapta pe 32 biţi
  - 8 bistabile
  - multiplicator 25 x 18
  - sumator
  - acumulator de 48 biţi
  - slice-uri speciale pentru DSP (digital signal processing)
- blocuri de memorie 36 Kbit RAM/FIFO
- viteză tranceiver – până la 28 Gb/s
- 2 convertoare analog – digitale de 12 biţi
- conectări dedicate pentru cascadare; conectare pe diagonale
- monitorizare circuit pt. alimentare şi temperatură

UltraSCALE Architecture

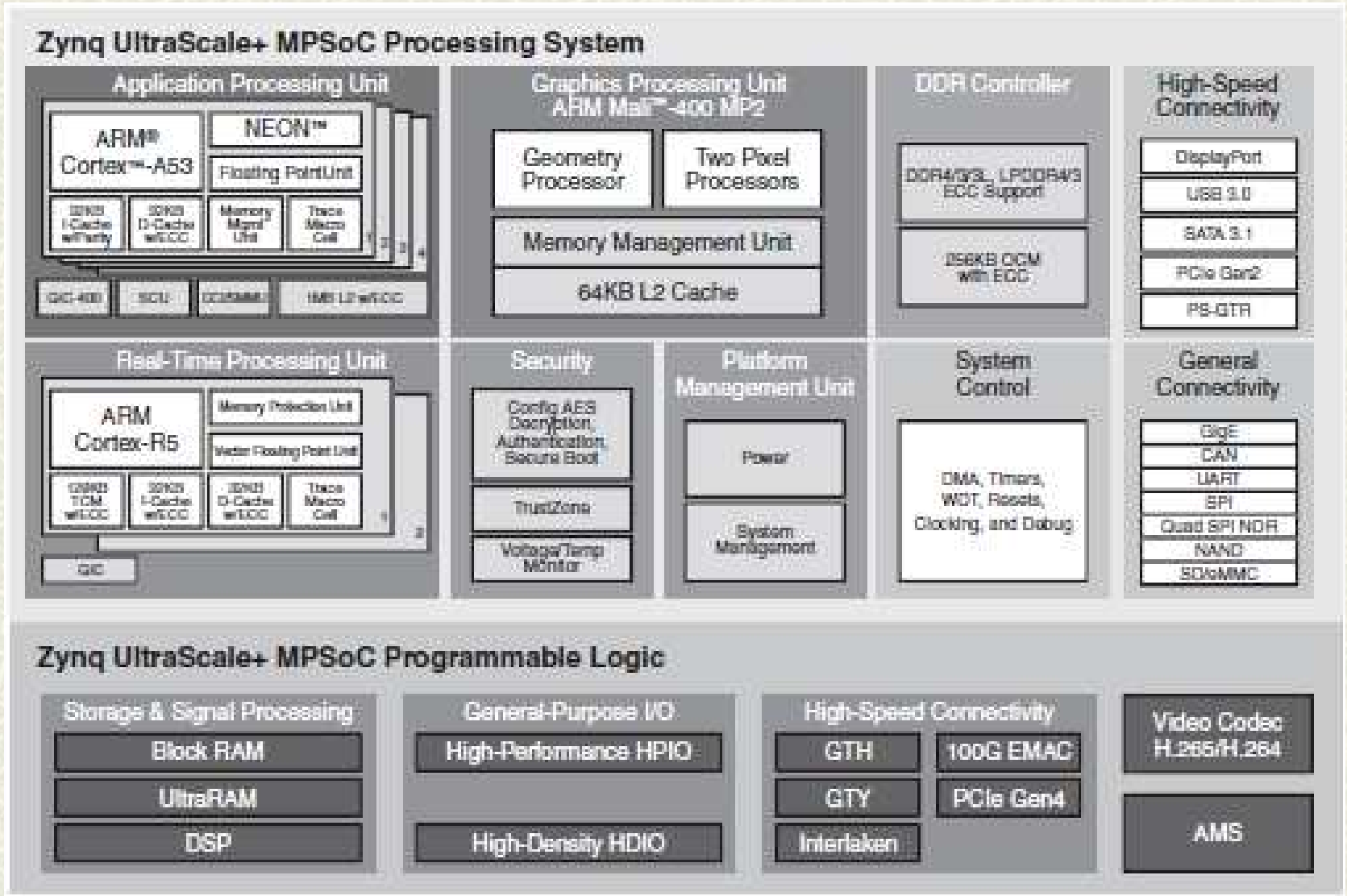
Comparaţie familii FPGA

	Spartan-6	Artix-7	Kintex-7	Virtex-7	Virtex UltraScale
Logic Cells	147,443	215,360	477, 760	1,954,560	4,407,480
BlockRAM	4.8Mb	13Mb	34Mb	68Mb	115Mb
DSP Slices	180	740	1,920	3,600	2,880
DSP Performance (symmetric FIR)	140GMACs	930GMACs	2,845GMACs	5,335GMACs	4,268 GMACs
Transceiver Count	8	16	32	96	104
Transceiver Speed	3.2 Gb/s	6.6 Gb/s	12.5 Gb/s	28.05 Gb/s	32.75 Gb/s
Total Transceiver Bandwidth (full duplex)	50 Gb/s	211 Gb/s	800 Gb/s	2,784 Gb/s	5,101 Gb/s
Memory Interface (DDR3)	800	1,066	1,866	1,866	2,400
PCI Express® Interface	x1 Gen1	x4 Gen2	x8 Gen2	x8 Gen3	x8 Gen3
Analog Mixed Signal (AMS)/XADC	-	XADC	XADC	XADC	System Monitor
Configuration AES	Yes	Yes	Yes	Yes	Yes
I/O Pins	576	500	500	1,200	1,456
I/O Voltage	1.2V - 3.3V	1.2V - 3.3V	1.2V - 3.3V	1.2V - 3.3V	1.0 – 3.3V

Sisteme de procesare eterogene

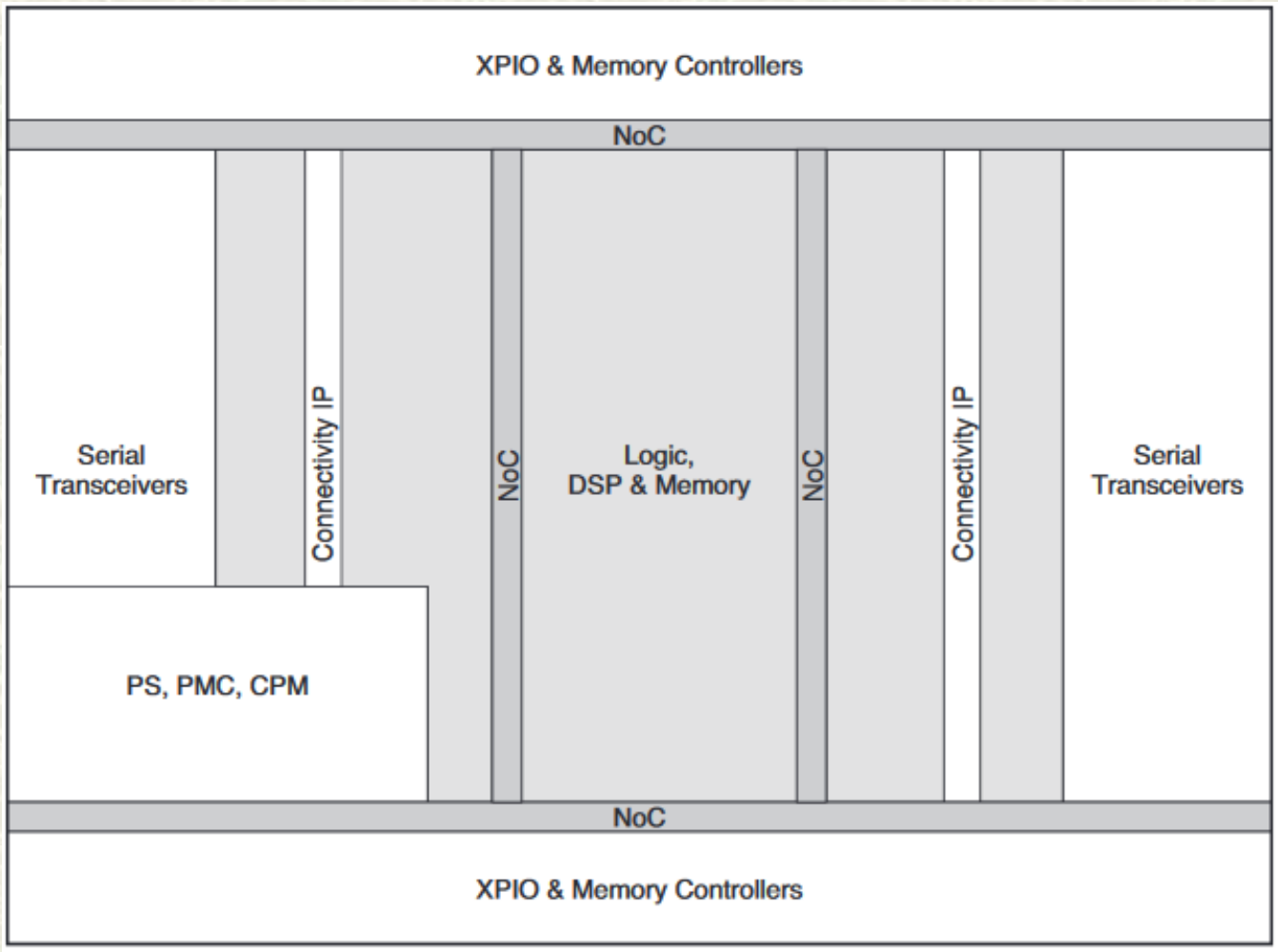


UltraScale MPSoC Architecture



Versal Architecture

Adapting Compute Acceleration Platform



CIRCUITE PROGRAMABILE

Avantaje

- reduceri substanţiale la:
  - dimensiuni
  - consum de putere
- creşteri substanţiale pentru:
  - viteza de operare
  - viteza de reconfigurare

Provocări

- dezvoltarea principiilor, uneltelor şi unificarea tehnologiilor actuale
- strategii viitoare pentru proiectarea arhitecturilor HW şi SW