

UNIVERSITATEA TEHNICĂ A MOLDOVEI

ȘCOALA DOCTORALĂ

Cu titlu de manuscris

CZU 621.3.049.774:004.94:004.4(043.3)

PLETEA IONICA-MARCELA

PROIECTAREA CIRCUITELOR INTEGRATE PE VERTICALĂ FOLOSIND SOFTURILE CURENTE 2D DE PROIECTARE DIGITALĂ

233.01 ” Nano-microelectronică şi optoelectronică”

Teză de doctorat în științe inginerești

Conducător științific

ȘONTEA Victor, doctor în științe fizico-matematice, profesor universitar

CHIŞINĂU, 2026



UNIVERSITATEA TEHNICĂ A MOLDOVEI

ȘCOALA DOCTORALĂ

Cu titlu de manuscris

CZU 621.3.049.774:004.94:004.4(043.3)

PLETEA IONICA-MARCELA

PROIECTAREA CIRCUITELOR INTEGRATE PE VERTICALĂ FOLOSIND SOFTURILE CURENTE 2D DE PROIECTARE DIGITALĂ

233.01 ” Nano-microelectronică şi optoelectronică”

Teză de doctorat în științe inginerești

Autor: PLETEA Ionica-Marcela _____

Conducător științific:
ȘONTEA Victor, doctor în științe fizico-matematice, profesor universitar _____

Membrii comisiei de îndrumare:

1. LUPAN Oleg, dr. hab., prof. univ., UTM _____
2. TROFIM Viorel, dr. hab., prof. univ., UTM _____
3. TIGHINEANU Ion, acad., dr. hab., prof. univ., AȘM _____

CHIŞINĂU, 2026

© Pletea Ionica-Marcela, 2026

CUPRINS

ADNOTARE 7

АННОТАЦИЯ 8

ANNOTATION 9

LISTA ABREVIERILOR 10

INTRODUCERE 12

1. METODE ȘI TEHNOLOGII DE PROIECTARE A CIRCUITELOR INTEGRATE ..19

1.1. Tehnologii de producere a circuitelor integrate VLSI 19

1.2. Tehnologii actuale 3D de implementare a circuitelor integrate 25

1.2.1 Tehnologie de stivuire pe verticală - TSV 25

1.2.2 Tehnologia Monolitic 3D 29

1.3 Dispozitive comerciale bazate pe tehnologia TSV și instrumente 3D EDA 34

1.4 Instrument de simulare pentru tehnologia Monolitic 3D 37

1.5 Concluzii la capitolul 1 38

2. METODOLOGIE DE IMPLEMENTARE A UNUI CIRCUIT INTEGRAT 2D 40

2.1 Sinteza circuitului, STA, implementarea DFT și verificarea formală 44

2.2 Proiectarea suprafeței circuitului integrat, stabilirea floorplanului și a planului de alimentare 46

2.3 Plasarea celulelor logice standard, optimizările necesare plasării optime 49

2.4 Crearea arborelui de ceas 53

2.5 Rutarea conexiunilor din circuit 55

2.6 Concluzii la capitolul 2 56

3. PROIECTARE 3D UTILIZÂND INSTRUMENTE 2D. MODURI DE PARTIȚIONARE A CIRCUITULUI 59

3.1 Rolul partiționării în trecerea de la integrarea 2D la 3D 59

3.2 Metode de proiectare a unui circuit integrat pentru adaptarea la cea de a 3 a dimensiune 62

3.3 Concluzii la capitolul 3 66

4. IMPLEMENTAREA ȘI VERIFICAREA UNUI FLUX DE PROIECTARE 3D UTILIZÂND PROGRAME 2D EXISTENTE 68

4.1 Valorificarea și optimizarea metodologiei 2D pentru implementarea circuitelor integrate 3D 68

4.2. Descrierea detaliată a etapelor premergătoare și a etapelor parcurse pentru plasarea blocurilor de memorie peste celulele logice 84

4.2.1. Etapa 1: Evaluarea circuitului 85

4.2.2. Etapa a 2 a: Plasarea blocurilor de memorii 86

4.2.3. Etapa a 3 a: Plasarea detaliată 87

4.3 Implementarea și analiza arborelui de ceas 88

4.4 Rutarea conexiunilor 92

4.5 Impactul conexiunilor verticale asupra ariei în integrarea 3D 95

4.6 Metodă specifică de implementare a unui circuit integrat 3D folosind instrumente 2D 98

4.7 Concluzii la capitolul 4 101

5. PERSPECTIVE ÎN DEZVOLTAREA STRUCTURILOR DE CALCUL AI ÎN TEHNOLOGIA MONOLITIC 3D 104

5.1 Rețea neuronală cubică cu trei straturi 110

5.2 Implementarea NG 4x4x1 într-o structură FPGA 112

5.3 Implementarea NG 4x4x1 in tehnologie monolitic 3D 114

5.4 Concluzii 120

CONCLUZII GENERALE ȘI RECOMANDĂRI 122

BIBLIOGRAFIE 125

ANEXA 1. Script pentru rularea procesului de lucru 2D 137

ANEXA 2. Fișierul LEF utilizat la generarea memoriei SRAM_128x38slp în formă de serpentină 142

ANEXA 3. Script pentru rularea procesului de lucru 3D 163

ANEXA 4. Descriere cod VHDL perceptron 171

ANEXA 5. Descriere cod VHDL GN_4x4x1 172

ANEXA 6. Reprezentări detaliate ale implementării circuitelor 175

ANEXA 7. Implementarea rezultatelor științifice 190

MULȚUMIRI 201

DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII.....202

CURRICULUM VITAE.....203

PARTICIPARE LA PROIECTE ȘTIINȚIFICE INTERNAȚIONALE.....207

LISTA LUCRĂRILOR ȘTIINȚIFICE ȘI ȘTIINȚIFICO-METODICE-DIDACTICE PUBLICATE LA TEMA TEZEI 211

EXPERIENȚĂ STIINȚIFICĂ.....214

ADNOTARE

la teza cu titlul ”**PROIECTAREA CIRCUITELOR INTEGRATE PE VERTICALĂ FOLOSIND SOFTURILE CURENTE 2D DE PROIECTARE DIGITALĂ**”, înaintată de **PLETEA Ionica-Marcela**, pentru conferirea gradului științific de doctor în științe inginerești, la specialitatea 233.01 ”NANO-MICROELECTRONICĂ ŞI OPTOELECTRONICĂ”.

Structura tezei: Teza este scrisă în limba română și constă din introducere, 5 capitole, concluzii generale și recomandări, bibliografie din 141 titluri, 7 anexe, 113 pagini de text de bază, 61 figuri, 5 tabele. Rezultatele obținute sunt publicate în 10 articole științifice și validate prin două certificate de implementare și 2 cereri de brevet, care atestă aplicarea practică a metodologiei și a soluțiilor dezvoltate în cadrul cercetării doctorale.

Cuvinte-cheie: circuite integrate, nanotehnologii, integrare pe verticală, monolitic 3D, EDA.

Scopul lucrării: constă în dezvoltarea unei metodologii integrate pentru proiectarea și validarea experimentală a circuitelor monolitice 3D, cu accent pe adaptarea fluxului de lucru, a fișierelor tehnologice, respectiv a instrumentelor tradiționale de implementare 2D, la integrarea pe verticală.

Obiectivele cercetării: Dezvoltarea unei metode cheie necesare adaptării uneltelor EDA 2D pentru proiectarea circuitelor integrate în tehnologie de integrare monolitică 3D. În acest scop, a fost elaborată o metodologie bazată în mare parte pe fluxurile existente de proiectare 2D, completată cu etape specifice și necesare integrării 3D monolitice. Cercetarea a urmărit modificarea etapelor de floorplanning, plasare și rutare astfel încât acestea să devină sensibile la cea de-a treia dimensiune. De asemenea, bibliotecile utilizate în cadrul fluxului de proiectare au fost completate cu parametri, atribute și valori specifice integrării monolitice 3D. Un alt obiectiv a constat în definirea unei strategii de partiționare pe două straturi, în care celulele logice sunt plasate pe primul strat, iar blocurile de memorie pe al doilea strat, pentru a permite aplicarea eficientă a metodologiei propuse.

Noutatea şi originalitatea ştiinţifică: constă în demonstrarea implementării unui circuit integrat monolitic 3D utilizând doar instrumente de proiectare 2D. Abordarea propusă in cadrul acestei teze este de a reutiliza cât mai mult posibil instrumentele convenționale existente 2D EDA și concentrarea doar pe câteva elemente critice EDA necesare pentru a permite crearea de metodologii 3D eficiente cu utilizarea instrumentelor actuale.

Problema științifică și de cercetare soluționată: constă în dezvoltarea și validarea unei metodologii care permite implementarea circuitelor integrate în tehnologie 3D monolitică utilizând instrumente EDA destinate proiectării ASIC 2D, prin modelarea memoriilor cu structuri virtuale de tip serpentină și adaptarea fluxului de implementare fizică la cerințele integrării verticale, eliminând astfel lipsa unui flux consacrat de proiectare pentru această tehnologie.

Semnificația teoretică și valoarea aplicativă a lucrării: Metodologia propusă spre cercetare stă la baza creării unui proces complet pentru proiectarea circuitelor integrate folosind tehnologia monolitică 3D. Aplicarea acestei metodologii creează premisele realizării unor circuite integrate cu performanțe superioare și consum redus de energie, valorificând infrastructura existentă și contribuind la reducerea costurilor de dezvoltare..

Implementarea rezultatelor stiințifice: Rezultatele cercetării demonstrează fezabilitatea utilizării instrumentelor de proiectare 2D pentru implementarea circuitelor integrate monolitic 3D și validează aplicabilitatea practică a metodologiei propuse. Aceste rezultate vor fi transferate în mediul educațional prin dezvoltarea și introducerea unui modul de curs dedicat proiectării circuitelor integrate 3D, destinat studenților masteranzi, contribuind la formarea specialiștilor în domeniul microelectronicii avansate.

АННОТАЦИЯ

к диссертации ”**ПРОЕКТИРОВАНИЕ ВЕРТИКАЛЬНЫХ ИНТЕГРАЛЬНЫХ СХЕМ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ПРИЛОЖЕНИЙ 2D ДЛЯ ЦИФРОВОГО ПРОЕКТИРОВАНИЯ**”, представленной соискателем **ПЛЕТЯ Ионика-Марчела**, для присуждения ученой степени доктора технических наук по специальности 233.01 ”НАНО-МИКРОЭЛЕКТРОНИКА И ОПТОЭЛЕКТРОНИКА”.

Структура диссертации: Диссертация написана на румынском языке и состоит из введения, 5 глав, общих выводов и рекомендаций, библиографии, включающей 141 наименование, 5 приложений, 113 страниц основного текста, 61 рисунков и 7 таблиц. Полученные результаты опубликованы в 10 научных статьях и подтверждены 2 сертификатами внедрения и 2 заявками на патент, что свидетельствует о практическом применении методологии и решений, разработанных в рамках докторского исследования.

Ключевые слова: интегральные схемы, нанотехнологии, вертикальная интеграция, монолитная 3D, САПР.

Цель работы: Разработка методологии для проектирования и экспериментальной проверки монолитных 3D схем, с фокусом на применение потока работы, технологических файлов, а также традиционных инструментов разработки 2D, в вертикальной интеграции.

Задачи исследования: Разработка ключевой методики при использовании инструментов САПР 2D в технологии интеграции монолитной 3D. Разработка методики, состоящей из шагов, применяемых в уже существующих методиках проектирования интегральных схем 2D, дополненная шагами для интеграции монолитной 3D. Некоторые этапы проектирования должны быть изменены, чтобы стать «чувствительными к третьему измерению». А также библиотеки должны быть дополнены параметрами, атрибутами и значениями, в зависимости от характеристик монолитной 3D интеграции.

Научная новизна и оригинальность состоит в демонстрации экспериментальной разработки монолитной интегральной схемы 3D с использованием только инструментов проектирования 2D. Предложенный в этой работе подход заключается в повторном использовании как можно большего набора существующих инструментов 2D САПР и фокусирование только на некоторых критичных элементах САПР, необходимых для создания методологии 3D при использовании нынешних инструментов.

Решённая научно-техническая задача Заключается в разработке и валидации методологии, позволяющей реализовывать монолитные 3D интегральные схемы с использованием EDA-инструментов для проектирования 2D ASIC, тем самым устраняя отсутствие устоявшегося маршрута проектирования для данной технологии.

Теоретическая значимость и прикладная ценность работы: Предложенная методика стоит в основе создания полного процесса проектирования монолитных интегральных схем 3D. Проектирование интегральных схем с использованием этой методики проектирования поспособствует созданию более эффективных схем для ежедневного использования, но за ту же цену и потребляющих меньше энергии.

Внедрение научных результатов: Результаты исследования демонстрируют возможность использования инструментов двумерного проектирования для реализации монолитных трёхмерных интегральных схем и подтверждают практическую применимость предложенной методологии. Полученные результаты будут внедрены в образовательный процесс посредством разработки и внедрения учебного модуля по проектированию трёхмерных интегральных схем для студентов магистратуры, способствуя подготовке специалистов в области современной микроэлектроники.

ANNOTATION

to the thesis entitled "DESIGN OF VERTICAL INTEGRATED CIRCUITS USING CURRENT 2D DIGITAL DESIGN SOFTWARE", submitted by PLETEA Ionica-Marcela, for the award of the scientific degree of Doctor of Engineering Sciences, in the specialty 233.01 "NANO-MICROELECTRONICS AND OPTOELECTRONICS"

Thesis structure: The thesis is written in Romanian and consists of an introduction, 5 chapters, general conclusions and recommendations, a bibliography of 141 titles, 7 annexes, 113 pages of basic text, 61 figures, 5 tables. The results obtained are published in 10 scientific articles and validated through 2 implementation certificates and 2 patent applications filed, which attest to the practical applicability of the methodology and solutions developed within the doctoral research

Keywords: integrated circuits, nanotechnologies, vertical integration, monolithic 3D, EDA

The purpose of the work: consists of developing an integrated methodology for the design and experimental validation of 3D monolithic circuits, with a focus on adapting the workflow, technological files, and traditional 2D implementation tools, to vertical integration.

Research objectives: The research objectives included the development of a key method required for adapting 2D EDA tools to the design of integrated circuits using monolithic 3D integration technology. To achieve this goal, a methodology was developed based largely on existing 2D integrated circuit design flows, complemented by specific steps required for monolithic 3D integration. The research focused on modifying the floorplanning, placement, and routing stages to make them aware of the third dimension. In addition, the libraries used throughout the design flow were extended with parameters, attributes, and values specific to monolithic 3D integration. Another objective was the definition of a two-tier partitioning strategy, where logic cells are placed on the first tier and memory blocks on the second tier, enabling the efficient application of the proposed methodology.

Scientific novelty and originality: consists in the demonstration of the implementation of a 3D monolithic integrated circuit using only 2D design tools. The approach proposed in this thesis is to reuse as much as possible existing conventional 2D EDA tools and focus only on a few critical EDA elements necessary to enable the creation of efficient 3D methodologies using current tools.

The scientific and research problem solved: consists of the development and validation of a methodology that enables the implementation of monolithic 3D integrated circuits using EDA tools originally intended for 2D ASIC design, by modelling memory blocks as virtual serpentine structures and adapting the physical implementation flow to the requirements of vertical integration, thereby overcoming the absence of an established design flow for this technology.

Theoretical significance and applied value of the work: The proposed methodology provides the foundation for the development of a complete design process for integrated circuits using monolithic 3D technology. The application of this methodology creates the prerequisites for the realization of integrated circuits with enhanced performance and reduced power consumption while leveraging existing infrastructure and contributing to lower development costs.

Implementation of scientific results: The research results demonstrate the feasibility of using 2D design tools for the implementation of monolithic 3D integrated circuits and validate the practical applicability of the proposed methodology. These results will be transferred to the educational environment through the development and introduction of a course module dedicated to 3D integrated circuit design for master's students, contributing to the training of specialists in advanced microelectronics.

LISTA ABREVIERILOR

2D – integrare planară (bidimensională)

3D – integrare tridimensională

AI – inteligență artificială (Artificial Inteligence)

ASIC – circuit integrat cu aplicație specifică (Aplication Specific Integrated Circuit)

BACKEND – etapă de proiectare de la floorplan pâna la rutare

BUFFER – celulă de întârziere

CAD – proiectare asistată de calculator (Computer Aided Design)

CIS – senzor de imagine CMOS (CMOS Image Sensor)

CMOS – tehnologie cu semiconductor complementar metal-oxid (Complementary Metal Oxide Semiconductor)

CORNER – condiții extreme de funcționare

DB – bază de date (Database)

DEF – fișier cu informații despre circuit (Design Exchange Format)

DFM – circuit pentru fabricare (Design for Manufacturing)

DFT – circuit pentru testare (Design for Test)

DRC – reguli de constrângere a circuitului (Design Rule Constraints)

EDA – automatizarea proiectării electronice (Electronic Design Automation)

EM – electromigrare (ElectroMigration)

FinFET – tranzistor cu efect de câmp cu aripioară (Fin Field-Effect Transistor)

FLOW – flux de proiectare

FPGA – matrice de porți logice programabilă (Field Programmable Gate Array)

FPIC – cipuri de interconectare programabile în câmp (Field Programmable Interconnect Chips)

FRONTEND – etapă de proiectare incluzând sinteză până la floorplan

HBM – memorie cu lățime de bandă mare (High Bandwidth Memory)

HDL – descriere hardware a unui circuit (Hardware Description Language)

HMC – cub de memorie hibrid (Hybrid Memory Cube)

IC – circuit integrat (Integrated Circuits)

I/O – pini de intrare-ieșire

ILD – dielectric între straturi (Inter Layer Dielectric)

IR Drop – cădere de tensiune (Voltage Drop)

KOH – Hidroxid de Potasiu

LEF – fișier tehnologic cu extensia .lef (Library Exchange Format)

LIB – format de librărie (LIBrary format)

LDPC – decodor cu verificare de paritate de densitate redusă (Low Density Parity Check)

LTCVD – depunere chimică din fază de vapori la temperatură joasă (Low-Temperature Chemical Vapor Deposition)

LVS – implementare fizică versus schematic (Layout versus Schematic)

MEMS – Sisteme Micro-Electromecanice (Micro-Electro-Mechanical Systems)

MIV – monolitic via între straturi (Monolithic Inter-tier Vias)

MOSFET – tranzistoare cu efect de câmp metal-oxid-semiconductor (Metal Oxide Semiconductor Field Effect Transistor)

MW – librărie MilkyWay

NAND – poartă logică ȘI NEGAT

SAED – biblioteci de proiectare Synopsys (Synopsys Academic Education Design)

SI – integritatea semnalelor (Signal Integrity)

SOC – sistem într-un circuit integrat (System On Chip)

SSD – unitate de stocare (Solid State Drive)

TSV – via prin Siliciu (Through Silicon Via)

VCD – fișier de înregistrare a variațiilor valorilor (Value Change Dump)

VLSI – integrare la scara larga (Very Large Scale Integration)

WLM – model de fire pentru sinteză (WireLoad Model)

INTRODUCERE

Actualitatea și importanța temei abordate. Prin cercetarea şi dezvoltarea de noi metode de proiectare a circuitelor integrate 3D monolitice, se va deschide o nouă eră pentru Legea lui Moore [1] în industria semiconductorilor şi industria microelectronică, deschizând calea pentru continuarea creşterii densității de integrare [2–5]. Un număr de studii au analizat circuitele integrate 3D monolitice dintr-o perspectivă teoretică. Davis și echipa sa a raportat o reducere de 3 ori a ariei totale de siliciu și o reducere de 12 ori a dimensiunilor pentru un circuit integrat realizat în tehnologie 3D monolitică pe 4 straturi, comparativ cu un circuit 2D [6]. Reducerea lungimii traseelor folosite pentru realizarea conexiunilor în circuit permite scăderea dimensiunii buffer-elor folosite pentru retransmitere și astfel se reduce distanța dintre porțile logice creându-se un "efect de feedback pozitiv" ce duce la reducerea semnificativă a ariei de siliciu [7]. Zhou și echipa sa a raportat o îmbunătățire cu mai mult de un ordin de magnitudine în produsul (putere x arie x întârziere) pentru un decodor LDPC [8].

Un studiu [9] arată că diametrul trecerilor TSV are un impact mare asupra ariei, lungimii traseelor, întârzierii și consumului de putere pentru circuite integrate 3D realizate în tehnologii de 45 nm și 22 nm. În plus, studiul arată că, dacă numărul trecerilor TSV este ales cu grijă, circuitul 3D este superior unui circuit 2D echivalent din punct de vedere al lungimii traseelor și al întârzierilor. În cazul unui circuit de test aria totală ocupată de trecerile TSV este de aproximativ 40%. Datorită acestei dimensiuni crește și lungimea traseelor. Când sunt folosite treceri TSV cu diametru mai mic, dimensiunea ariei ocupate de acestea este doar de 11,2%, pentru același circuit de test. Astfel, se obține și o reducere a lungimii traseelor ce se traduce într-o întârziere a căii critice cu aproximativ 36% mai mică decât în cazul circuitului 2D.

Instrumentele EDA (automatizarea proiectării electronice) consacrate (ex. Synopsys, Cadence) nu oferă încă suport nativ pentru integrarea monolitică 3D pe o singură pastilă de siliciu, însă este de așteptat ca această capabilitate să fie dezvoltată în viitor. Un obiectiv al acestei teze este integrarea metodelor propuse în unelte EDA existente și validarea potențialului practic al fluxului de proiectare propus.

Tehnici inovative de proiectare și automatizare sunt imperativ necesare pentru realizarea integrării 3D cu succes [10–12]. În special, partiționarea are un mare impact asupra performanțelor circuitului la toate nivelurile de abstractizare [13]. Pe de altă parte, optimizările realizate la un nivel înalt de abstractizare pot conduce la avantaje mai mari decât cele realizate la un nivel scăzut. Astfel, partiționarea funcțională la nivel arhitectural prezintă avantaje mai mari decât partiționarea circuitului la nivel fizic, deoarece partiționarea funcțională realizează optimizări la nivelul relației funcționalitate-conectivitate dintre blocurile logice [14]. În plus, problemele ridicate de

partiționarea la nivel arhitectural [15] pot avea o complexitate mai mică decât cele ridicate de partiționarea la nivel fizic (de ex**.** zeci, sute de blocuri logice comparativ cu milioane, miliarde de celule logice).

Problema partiționării funcționale nu reprezintă un subiect de interes recent în domeniul proiectării circuitelor integrate. Din cauza timpilor de proiectare mai mici și a costurilor relativ scăzute, dispozitivele de tip FPGA s-au utilizat ca soluții pentru realizarea mai rapidă de prototipuri ale ASIC-urilor [16–18]. Mai mult, integrarea FPGA-urilor cu circuite programabile FPIC a condus la apariția unor arhitecturi hardware reconfigurabile, devenind o tendință importantă în prototiparea rapidă și în aplicațiile cu cerințe mari de calcul. Una dintre provocările esențiale în implementările care implică mai multe FPGA-uri este procesul de partiționare optimă a circuitului între aceste dispozitive [19].

Integrarea 3D monolitică folosește treceri MIV cu dimensiuni reduse pentru conectarea diferitelor straturi active [20, 21]. Trecerile ILV au dimensiuni asemănătoare cu dimensiunea minimă de realizare a componentelor prin proces fotolitografic și numărul lor într-un circuit integrat nu reprezintă o constrângere severă precum trecerile TSV sau numărul de pini I/O. Astfel, crește flexibilitatea procesului de proiectare a circuitelor integrate, permițând optimizări funcționale, de performanță și de cost. Exploatarea eficientă a acestor avantaje necesită dezvoltarea unor metodologii de proiectare adaptate tehnologiei Monolithic 3D.

Scopul lucrării: constă în dezvoltarea unei metodologii integrate pentru proiectarea și validarea experimentală a circuitelor monolitice 3D, cu accent pe adaptarea procesului de lucru, a fișierelor tehnologice, respectiv a instrumentelor tradiționale de implementare 2D, la integrarea pe verticală.

Obiective de cercetare propuse:

- Investigarea capabilităților instrumentelor EDA destinate proiectării ASIC 2D și identificarea etapelor, fișierelor tehnologice și bibliotecilor care pot fi adaptate pentru implementarea structurilor Monolithic 3D.
- Dezvoltarea unei metode de reprezentare a celui de-al doilea strat activ prin modelarea blocurilor de memorie cu structuri virtuale de tip serpentină și utilizarea atributului CLASS COVER, pentru a permite utilizarea eficientă a instrumentelor EDA 2D în implementările Monolitic 3D.
- Adaptarea etapelor principale ale fluxului de implementare fizică, inclusiv floorplanning-ul, plasarea, rutarea și verificarea fizică, astfel încât acestea să poată gestiona distribuirea funcționalităților pe două straturi active.

- Completarea bibliotecilor și fișierelor tehnologice cu parametri și atribute specifice integrării Monolitic 3D, necesare reprezentării și implementării structurilor tridimensionale în cadrul fluxurilor ASIC 2D existente.
- Dezvoltarea și validarea unui flux complet de proiectare Monolitic 3D bazat pe reutilizarea infrastructurii EDA ASIC 2D, utilizând instrumentele Synopsys și analiza comparativă a implementărilor 2D și 3D din punct de vedere al ariei, interconectărilor, resurselor utilizate și performanței.
- Evaluarea unei strategii de partiționare bazate pe separarea memoriilor și a logicii pe două straturi active și determinarea impactului acesteia asupra densității de integrare și eficienței implementării Monolitic 3D.

Noutatea științifică a rezultatelor cercetării: constă în dezvoltarea și validarea unei metodologii complete de proiectare a circuitelor integrate Monolitic 3D utilizând exclusiv instrumente EDA destinate fluxurilor ASIC 2D. În cadrul cercetării a fost propusă o metodă originală de reprezentare a memoriilor prin intermediul unor structuri virtuale de tip serpentină, memoriile fiind definite în fișierele LEF utilizând atributul **CLASS COVER**, ceea ce permite modelarea celui de-al doilea strat activ în mediile de proiectare 2D și implementarea logicii sub blocurile de memorie. A fost elaborată și validată o strategie de partiționare logică–memorie pe două straturi active, integrată într-un flux unitar de implementare fizică Monolitic 3D. De asemenea, au fost evidențiate efectele integrării verticale asupra ariei, interconectărilor și rețelei de distribuție a semnalului de ceas și a fost demonstrată fezabilitatea implementării unor structuri neuronale hardware compatibile cu dezvoltarea viitoare a arhitecturilor bazate pe tehnologia Monolitic 3D.

Problema stiințifică de cercetare soluționată: constă în dezvoltarea și validarea unei metodologii care permite implementarea circuitelor integrate în tehnologie 3D monolitică utilizând instrumente EDA destinate proiectării ASIC 2D, prin modelarea memoriilor cu structuri virtuale de tip serpentină și adaptarea fluxului de implementare fizică la cerințele integrării verticale, eliminând astfel lipsa unui flux consacrat de proiectare pentru această tehnologie. Utilizând această metodologie de proiectare pot fi testate diferite configurații de circuite monolitice 3D, iar cercetările pot progresa în direcția asta pâna la optimizarea programelor care să permită integrarea monolitic 3D. De asemenea fișierele tehnologice utilizate pentru implementarea acestei metodologii pot constitui un punct de plecare pentru actualizarea bibliotecilor viitoare ce vor permite integrarea monolitic 3D. Această metodologie de proiectare 3D utilizând instrumentele 2D va ajuta producătorii de circuite integrate să fabrice circuite integrate mai performante și mai mici la aceleași costuri cu ajutorul echipamentelor de fabricație existente și deja amortizate.

Semnificația teoretică și valoarea aplicativă a lucrării: Metodologia propusă spre cercetare stă la baza creării unui proces complet pentru proiectarea circuitelor integrate folosind tehnologia monolitică 3D. Proiectarea circuitelor integrate folosind această metodologie de proiectare va duce la dispozitive destinate uzului de zi cu zi mai performante, dar la același preț și care vor consuma mai puțină energie. Folosind aceleași echipamente, fabricile producătoare de circuite integrate vor putea ajuta la lansarea pe piață a următoarei generații de dispozitive. Tipurile de dispozitive care vor putea beneficia de aceste îmbunătățiri vor fi folosite pentru aplicații industriale, științifice, medicale, de cercetare, comerciale, militare și altele. Un exemplu de utilizare al acestor tipuri de circuite integrate ar putea fi echipamentele militare de apărare care vor asigura o performanță sporită, astfel vor salva mai multe vieți. Aceste circuite integrate pot sta la baza dezvoltării unor dispozitive medicale cu performanțe îmbunătățite. Totodată, metoda propusă pentru tehnologia Monolitic 3D prezintă un potențial ridicat de aplicabilitate în multiple domenii.

Tezele științifice înaintate spre susținere:

1. Dezvoltarea unei metodologii de implementare **Monolitic 3D** bazate pe instrumente EDA destinate proiectării ASIC 2D, demonstrând posibilitatea realizării implementării fizice tridimensionale fără modificarea instrumentelor comerciale, prin adaptarea fluxului de proiectare și a fișierelor tehnologice.
2. Dezvoltarea unei metode de abstractizare a memoriilor prin structuri virtuale de tip serpentine și definite în fișierele LEF cu atributul **CLASS COVER**, care permite reprezentarea celui de-al doilea strat activ și implementarea arhitecturilor Monolitic 3D utilizând un flux convențional de proiectare ASIC.
3. Integrarea Monolitic 3D a logicii și memoriilor conduce la reducerea semnificativă a ariei de implementare, a lungimii interconectărilor și a numărului de bufere inserate, îmbunătățind utilizarea suprafeței de siliciu și eficiența implementării fizice comparativ cu implementarea ASIC 2D.
4. Optimizarea rețelei de distribuție a semnalului de ceas într-o arhitectură Monolitic 3D reduce resursele necesare arborelui de ceas, aria ocupată de celulele de ceas și skew-ul, conducând la o sincronizare mai eficientă și la o distribuție mai uniformă a rețelei de ceas.
5. Dezvoltarea și validarea unei arhitecturi neuronale modulare tridimensionale, denumită **Grăunte Neuronal**, implementată pe FPGA și evaluată printr-un flux complet ASIC, demonstrează fezabilitatea utilizării acesteia ca element de bază pentru acceleratoare hardware dedicate aplicațiilor de inteligență artificială bazate pe integrarea Monolithic 3D.

Prezentarea rezultatelor cercetărilor științifice din cadrul tezei de doctorat.

Principalele rezultate ale tezei de doctorat au fost prezentate și discutate la:

- Conferința tehnică "*IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference S3S 2015*", USA, New Jersey, 2015.
- Conferința tehnico-stiințifică "*Conferința Internațională* "*Telecommunicații, Electronică și Informatică"*. Chișinău, Republica Moldova, 2018.
- Conferința tehnico-stiințifică "*Conferința Internațională* "*Telecommunicații, Electronică și Informatică"*. Chișinău, Republica Moldova, 2019.
- *„Conferinţa tehnico-ştiinţifică a studenţilor, masteranzilor şi doctoranzilor"*. Chișinău, Republica Moldova, 2019.
- *„International Conference on Electronics, Communications and Computing",* IC ECCO Chișinău, Republica Moldova, 2024.
- Conferinta *International Symposium on Signals, Circuits, and Systems,* Iași, România, 2025.

De asemenea capitolul 5 din teză, intitulat "Perspective în dezvoltarea structurilor de calcul AI în tehnologia monolitic 3D" a fost conceput și editat în cadrul proiectului de cercetare "**Regions and (E)DIHs alliance for AI-atthe-Edge adoption by European Industry 5.0 Manufacturing SMEs**" , proiect ce face parte din programul European HORIZON-CL4-2022-TWIN-TRANSITION-0, codul proiectului este 101092069, acronim REDGIO 5.0. Adresa web a proiectului este: https://www.airedgio5-0.eu/.

Publicații la tema tezei. Rezultatele principale ale tezei au fost publicate în 10 articole stiințifice și anume 3 articole recenzate SCOPUS, 1 articol în revista JES din Registrul Național al revistelor de profil, 1 articol în revista Scientific Bulletin of UPT, precum și 8 lucrări prezentate și publicate la Conferințe Naționale și Internaționale, 3 dintre articole sunt indexate IEEE-Xplore (Lista publicațiilor este anexată la sfârșitul tezei).

Rezultatele obținute în cadrul prezentei cercetări au fost valorificate atât în mediul industrial, prin implementarea metodologiei dezvoltate în cadrul companiei Silicon Service SRL, cât și în mediul academic, prin utilizarea rezultatelor și a conținutului tezei ca material didactic în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației din cadrul Universității Tehnice „Gheorghe Asachi" din Iași. Totodată, pe baza rezultatelor cercetării au fost depuse la OSIM România 2 cereri de brevet de invenție privind metodologia propusă. Documentele care atestă implementarea industrială și academică, precum și depunerea cererilor de brevet, sunt prezentate în Anexa 7.

Volumul și structura tezei: Teza este constituită din introducere, cinci capitole, concluzii generale și recomandări, bibliografia din 141 titluri și 7 anexe. Conține 113 pagini de text, 61 de figuri și 5 tabele.

Cuvinte-cheie: circuite integrate, nanotehnologii, integrare pe verticală, monolitic 3D, EDA.

Conținutul de bază al lucrării.

În **Introducere** este argumentată actualitatea temei, descrisă situația actuală în domeniu, scopul şi obiectivele cercetării, noutatea științifică a rezultatelor cercetării, problema stiințifică de cercetare soluționată, semnificația teoretică și valoarea aplicativă a lucrării, tezele științifice înaintate spre susținere, prezentarea rezultatelor cercetărilor științifice din cadrul tezei de doctorat, publicațiile la tema tezei şi sumarul compartimentelor tezei.

În ***Capitolul 1*** sunt prezentate tehnologiile actuale de implementare a circuitelor integrate 3D și sunt explicate tehnologiile TSV și monolitic 3D. Se efectuează o sinteză a tehnologiilor folosite la implementarea circuitelor integrate la modul general și a modalităților de implementare a circuitelor 3D. Se face o analiză comparativă între tehnologiile actuale de implementare 3D și sunt trase concluzii despre avantajele respectiv dezavantajele fiecărei tehnologii în parte.

În ***Capitolul 2*** este descris un proces complet de implementare al unui circuit integrat planar 2D, modelat și optimizat cu informații obținute după rularea mai multor proiecte reale (proiecte în cadrul cărora au fost fabricate circuite integrate care sunt acum utilizate pe piață) și în care s-au parcurs toate etapele de producere a circuitelor integrate. Pentru o înțelegere mai clară a întregului proces de lucru s-au folosit atât imagini cu un caracter general descriptiv, cât și imagini generate prin intermediul programului utilizat pentru acea etapă, programe încorporate în instrumentul IC Compiler de la Synopsys.

În ***Capitolul 3*** este abordată tema partiționării circuitelor integrate, o etapă foarte importantă în procesul de trecere de la proiectarea circuitelor integrate în tehnologie 2D la integrarea 3D. În această etapă de partiționare, practic se decide exact ce componente (atât celule logice cât și blocuri de memorii) sunt plasate pe fiecare strat, astfel încât numărul de conexiuni verticale (MIV sau TSV, în funcție de tehnologia utilizată) să fie cât mai mic. Cu cât partiționarea este mai optimă, cu atât scade numărul de conexiuni verticale și astfel este redus consumul de energie, întârzierile pe traseele de semnal sunt micșorate, iar rutarea este mai puțin complexă.

În ***Capitolul 4*** este prezentată o metodologie originală de implementare a unui circuit integrat tridimensional (3D) prin suprapunerea memoriilor deasupra celulelor logice standard dintr-un circuit, care conține 50% memorii și 50% logică, folosind software-urile existente pentru circuite bidimensionale (2D) de la Synopsys. Folosind această metodologie s-au implementat 3

configurații de circuite și s-au prezentat avantajele integrării 3D comparativ cu 2D. Implementarea 3D s-a realizat modificând fișierele tehnologice și adaptând procesul de lucru astfel încât instrumentele 2D să permită suprapunerea memoriilor peste celulele logice standard fără a raporta erori de overlapping. Toate imaginile care conțin implementarea fizică a circuitului (layout-ul) sunt capturi de ecran din programul cu care s-a realizat procesul (partea Backend) de proiectare a circuitului integrat.

In ***Capitolul 5*** s-a inițiat ideea de a modela structuri neuromorfice 3D care să fie capabile să proceseze local algoritmi AI cu un consum minim de energie și să ocupe un spaţiu minim. Un element de noutate este lansarea și prezentarea conceptului de "Neural Grain" (grăunte neuronal) ce reprezintă o structură AI hardware prin care se mimează o strucură a creierului biologic. A fost modelată o structură cubică 3D a unui "Neural Grain" și s-a elaborat un cod VHDL ce este implementat, prin programul de sinteză VIVADO, într-un circuit FPGA din familia ARTIX 7. În paralel, aceeași structură NG 4x4x1 a fost modelată într-o tehnologie ce emulează tehnologia monolithic 3D prin parcurgerea pașilor de sinteză utilizând programul Genus de la Cadence.

Toate capitolele au la sfârșit concluzii asupra cercetărilor și rezultatelor obținute.

În ***Concluzii generale şi recomandări*** se prezintă analiza, sinteza și avantajele rezultatelor experimentale obținute, precum și recomandări privind aplicațiile practice viitoare.

În **Bibliografie** sunt prezentate materialele științifice analizate şi citate în teză.

1. METODE ȘI TEHNOLOGII DE PROIECTARE A CIRCUITELOR INTEGRATE

1.1. Tehnologii de producere a circuitelor integrate VLSI

Densitatea circuitelor integrate a crescut enorm în această eră a SOC-urilor, iar specialiștii au venit cu soluții care sa permită creșterea numărului de componente într-o arie mai mică, dar fără a afecta creșterea densității IC-urilor și lungimea interconexiunilor dintre componente. Tehnologia tridimensionala 3D [22] a apărut ca o soluție promițătoare pentru a crește performanțele circuitelor integrate, într-o arie mai mică și cu lungimea redusă a interconexiunilor [23]. Integrarea 3D reprezintă o tehnologie de ultimă generație, care s-a dezvoltat în ultimii ani și care a venit cu numeroase beneficii ca: performanță îmbunătățită a circuitelor integrate, arie mai mică și consum de energie scăzut. Însă, posibilitatea de a reduce tensiunea de alimentare fără pierderi de performanță, depinde în mare măsură de natura critică a căilor de semnal. Se poate concluziona astfel că impactul pozitiv al tehnologiei 3D variază în funcție de natura circuitului și nu se poate discuta de aceleași beneficii pentru toate tipurile de circuite [24, 25].

Un circuit integrat tridimensional 3D reprezintă o configurație în care mai multe straturi de circuite integrate sunt conectate pe verticală între ele, iar conexiunea se face fie la nivel de substrat (wafer) de Siliciu, fie la nivel de circuit integrat individual [26]. La ora actuală sunt cunoscute câteva tehnici de implementare a circuitelor integrate tridimensionale, iar cele mai importante sunt:

1. Suprapunerea substraturilor de siliciu – Wafer Bonding – în această metodă, substraturile de Siliciu sunt fabricate și procesate individual, iar apoi sunt suprapuse și conectate între ele folosind diferite tehnici.
2. Suprapunerea circuitelor integrate individuale – Die Stacking – în această metodă circuitele integrate sunt decupate de pe substratul de Siliciu și apoi sunt suprapuse și conectate între ele individual.
3. O metodă diferită de cele anterioare este tehnologia “Monolitic 3D” în care fiecare strat este fabricat unul peste altul pe același substrat (wafer) de Siliciu, fără a mai fi necesară suprapunerea și conectarea ulterioară a substraturilor de Siliciu ori a circuitelor integrate individuale.

Integrarea tridimensională promite mai mult decât o alternativă pentru creșterea densității de integrare. Aceasta oferă o altă dimensiune a flexibilității în proiectarea circuitelor integrate. Un aspect bine cunoscut al acestei flexibilitati este capacitatea de a împărți componentele circuitului

pe straturi care pot fi ulterior prelucrate și operate independent, dar să fie totuși strâns interconectate (valabil mai ales pentru tehnologia monolitică 3D).

Înainte de a face o descriere mai detaliată a metodelor de implementare 3D actuale, este necesară o scurtă parcurgere a istoricului circuitelor integrate [27] de la descoperirea dispozitivelor semiconductoare și până la momentul actual în care tranzistoarele sunt conectate pe verticală.

Descoperirea dispozitivelor semiconductoare și obținerea circuitelor integrate, au condus la implementarea unor componente și sisteme electronice de mare complexitate care a avut un impact major asupra dezvoltării tehnicii moderne [28]. Progresul electronicii datorat acestei descoperiri și încercarea continuă de miniaturizare a componentelor electronice, au permis avansarea rapidă a tehnologiei în mai multe ramuri ale științei și tehnicii moderne [29, 30].

Tehnologiile de fabricare a circuitelor integrate [31] au fost permanent dezvoltate și perfecționate, ceea ce a condus la creșterea apreciabilă a producției de astfel de circuite și la aplicarea acestora în cele mai diverse aplicații ale electronicii moderne.

Printr-un circuit integrat se înțelege o capsulă de material semiconductor monocristalin pe care sunt plasate componente atât active cât și pasive, interconectate electric inseparabil, componente care sunt realizate în procesul de fabricare a circuitului integrat. Circuitul integrat de tip monolitic are elemente de circuit formate dintr-o singură structură semiconductoare, iar cele fragmentare se realizează din mai multe structuri monolitice, interconectate între ele. Toate componentele din structura acestor circuite integrate se realizează în același timp, în cadrul operațiilor tehnologice specifice la care sunt supuse plachetele semiconductoare. Tehnologiile folosite la fabricare circuitelor integrate au fost permanent dezvoltate, iar începând cu anii 70 revoluția tehnologică a fost reprezentată de CMOS [32, 33]. CMOS este o tehnologie complementar-metal-oxid-semiconductor, utilizată în mod obișnuit și care atinge un nivel ridicat de performanță, consum redus de energie și fiabilitate în proiectarea circuitelor integrate, prin exploatarea proprietății complementare a tranzistoarelor de tip p și de tip n.

Tehnologia CMOS constă în două tipuri de tranzistoare cu efect de câmp metal-oxid-semiconductor și anume: tranzistor de tip P numit pMOS și tranzistor de tip N numit nMOS. Aceste tranzistoare MOSFET au 3 electrozi: drenă, sursă și poartă. Aplicând o tensiune pe poarta tranzistorului, fluxul de curent între drenă și sursă este controlat. Tranzistorul de tip N conduce curent atunci când tensiunea pe poartă depășește o anumită valoare de prag, permițând curgerea curentului de la drenă la sursă, iar tranzistorul de tip P conduce curent atunci când tensiunea pe poartă este sub o valoare de prag și permite curgerea curentului de la sursă la drenă. Componentele CMOS utilizează atât tranzistoare de tip pMOS cât și de tip nMOS pentru a obține un control flexibil al curentului, frecvență mărită și consum redus de putere.

Pentru a asigura o funcționalitate fiabilă a circuitelor CMOS, tehnologia de interconectare joacă un rol foarte important în proiectarea circuitului, conectarea realizându-se prin fire metalice sau polisiliciu. Trasarea firelor metalice afectează semnificativ performanțele circuitelor, considerând factori importanți precum lungimea, rezistența, volumul și proiectarea stratului de interconectare. Cele mai comune componente în tehnologia CMOS sunt inversoarele, a căror structură poate fi observată în figura 1.1 [34].

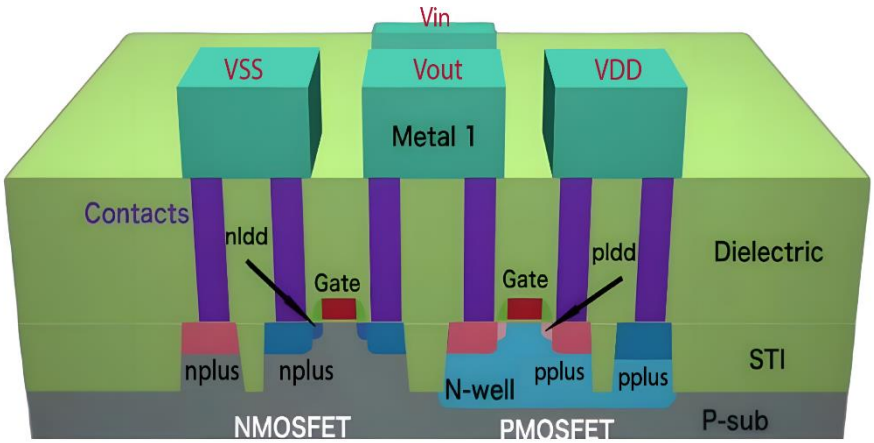


Fig. 1.1. Diagrama unui inversor CMOS [34]

Odată cu dezvoltarea la scară largă a aplicațiilor electronice, o înțelegere profundă și necesitatea optimizării tehnologiei CMOS au devenit cruciale pentru îmbunătățirea performanțelor dispozitivelor electronice. Practic, optimizarea tehnologiei cu CMOS implică reducerea dimensiunii tranzistoarelor [35–37] astfel încât densitatea de tranzistoare pe un substrat de Siliciu să crească, ceea ce va conduce la timpi de comutare rapizi, performanțe mai mari și un consum de energie mai mic. Reducerea dimensiunii tranzistoarelor nu se poate realiza decât până la o anumită limită, astfel apărând necesitatea integrării pe verticală care reprezintă o abordare revoluționară în industria semiconductorilor și implică suprapunerea straturilor de siliciu pe verticală comparativ cu plasarea straturilor unul lângă celălalt ca în integrarea tradițională 2D. Integrarea tridimensională reprezintă o tehnologie importantă pentru depășirea limitelor integrării clasice 2D, deoarece va permite continuarea creșterii performanțelor circuitelor integrate. Dar, deși integrarea tridimensională oferă avantaje promițătoare, ea vine și cu o serie de provocări, în special în ceea ce privește procesul de fabricare și gestionare termică a circuitelor integratelor [38].

Procesul de manufacturare a circuitelor integrate implică mai mulți pași: după descrierea funcționării circuitului în limbaj HDL, codul rezultat împreună cu o anumită tehnologie și un fișier de constrângeri vor parcurge toate etapele necesare implementării CI. În afară de verificările necesare după fiecare etapă din fluxul de producere a CI, legate de arie, viteză și consum de putere, este necesară și testarea fizică a CI care se realizează după producerea circuitului integrat. O

etapizare a proceselor de testarea automată a circuitelor este pusă în evidență în diagrama din figura 1.2.

După cum se poate observa în diagrama din figura 1.2, netlistul circuitului, care reprezintă porțile logice și interconexiunile dintre ele, rezultat după parcurgerea întregului proces de producere a circuitului integrat, împreună cu tehnologia CMOS, sunt implementate pe substratul de Siliciu. După implementarea circuitului, pe substratul de Siliciu se efectuează prima testare fizică automată din procesul de fabricare a circuitului, iar ulterior se efectuează încapsularea fiecărui circuit în parte.

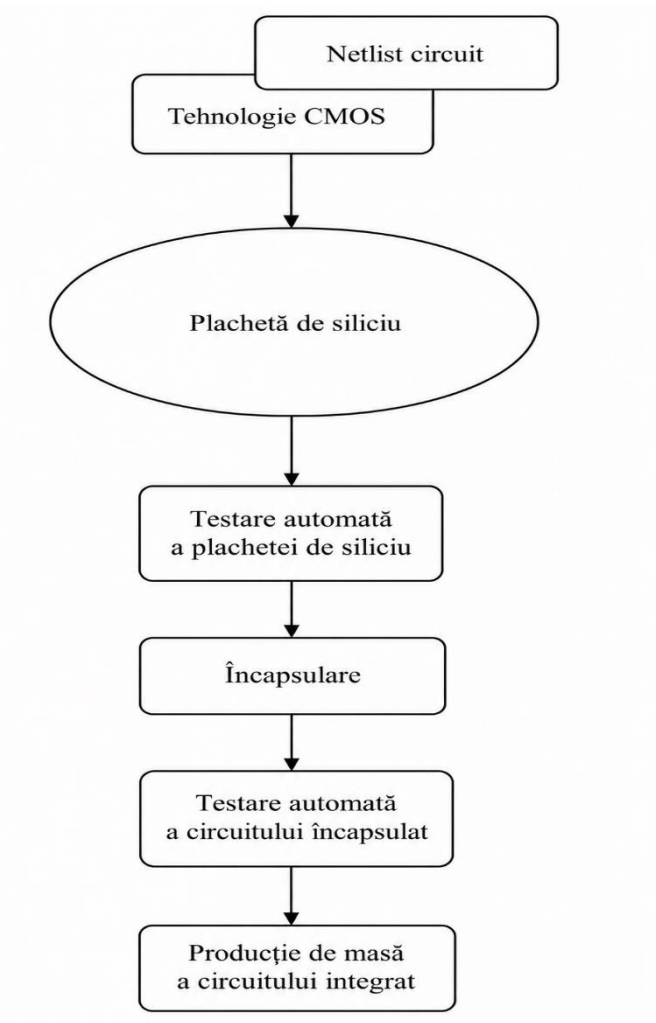


Fig. 1.2. Etape de testare automată a circuitelor integrate

Încapsularea circuitelor integrate se referă la procesul de protejare a cipului de siliciu și a conexiunilor sale într-o carcasă protectoare, astfel încât sa fie protejat de deteriorări fizice, umiditate și alte pericole din mediul înconjurător.

Scopul încapsulării se poate evidenția prin câteva aspecte:

- Protecție: protejează circuitul integrat de deteriorări fizice, umiditate, praf și contaminanți chimici.
- Disipare termică: ajută la gestionarea căldurii generate de circuitul integrat în timpul funcționării.
- Izolare electrică: previne interferențele electrice între circuitul integrat și mediul extern.
- Suport mecanic: oferă integritate structurală și suport pentru cipul de siliciu și conexiunile sale delicate.

Datorită numărului mare de componente dintr-un circuit integrat, testarea acestuia se face automat folosind un tester programabil și se realizează în două etape: în prima etapă se testează circuitul integrat când încă este pe substratul de Siliciu, înainte de tăiere, iar în a doua etapă de testare este după încapsulare când circuitele integrate sunt testate individual.

Pentru a obține netlistul folosit ca fișier de intrare din figura 1.2, echipa de proiectare a circuitului integrat parcurge o serie de etape, de la redactarea specificațiilor și modelarea într-un limbaj HDL și până la tape-outul spre producător care reprezintă fabricarea circuitului implementat software până la acest pas.

Tot procesul de proiectare a circuitelor integrate este unul complex care implică resursă umană, programe (instrumente) de proiectare a circuitelor integrate, cunoașterea etapelor de proiectare, dar și optimizarea și analiza acestora, respectarea regulilor de fabricare și asigurarea producerii unor circuite integrate care să asigure calitatea produsului final.

Toate aceste etape trebuie luate în considerare când se trece de la integrarea 2D la cea 3D, iar adaptarea întregului flux de lucru pentru producerea circuitelor integrate la cea de-a treia dimensiune trebuie luată în calcul la fiecare etapă începând cu modelarea circuitului într-un limbaj HDL, unde se poate deja partiționa circuitul și până la etapa finală de producere fizică a circuitului integrat.

O etapă intermediară între implementarea 2D a circuitelor integrate și integrarea 3D, a constituit-o integrarea 2.5D, care a avut o contribuție importantă în evoluția procesului de integrare pe verticală [39]. Cerința tot mai mare de performanță a circuitelor integrate, care să mențină costurile de producție la niveluri rezonabile, a condus la integrarea 2.5D și ulterior la integrarea pe verticală 3D. Tehnologia 2.5D nu se realizează pe verticală, ci se referă la tehnica de împachetare, în care mai multe cipuri (die-uri) sunt plasate unul lângă altul pe un substrat comun intermediar (siliciu, sticlă sau material organic). Cipurile sunt interconectate între ele, prin micro-bumps (structuri conductive foarte mici) și uneori prin intermediul TSV-urilor [40], în interiorul stratului intermediar, obținând astfel performanțe și densitate mai bune decât în cazul circuitelor

integrate 2D, însă fără complexitatea deplină a suprapunerii 3D [41]. Deși nu este la fel performantă ca integrarea 3D, integrarea 2.5D prezintă o serie de avantaje comparativ cu integrarea planară 2D și anume:

- Interconectări mai scurte, datorate distanțelor reduse dintre componente, care conduc la întârzieri mai mici, prin urmare semnale mai rapide;
- Integrarea 2.5D permite integrarea eterogenă [42, 43]: se pot combina circuite integrate implementate în diferite tehnologii. Pe același strat intermediar se pot plasa CPU alături de memorii sau circuite integrate cu alte tipuri de funcționalități (RF, analog, etc). Astfel se pot obține sisteme performante cu funcționalitate extinsă, fără a fi nevoie de un circuit integrat monolitic de dimensiune mare și costisitor;
- Consum de energie mai mic comparativ cu conectarea 2D tradițională: traseele electrice scurte și interconectările dense realizate cu ajutorul micro-bump-urilor reduc pierderile de semnal precum și capacitatea parazită și inductanțele traseelor. Prin urmare scade consumul global al sistemului, se reduce încălzirea și permite astfel operarea la frecvențe mai mari;
- Flexibilitatea în proiectare: permite reutilizarea circuitelor integrate validate anterior, adică a celor care au fost deja produse și testate. Acest aspect conduce la timpi de dezvoltare mai mici și costuri de producție mai reduse;
- Testarea circuitelor în configurarea 2.5D este mai ușoară decât în cazul 3D [44]. Fiind plasate pe aceeași parte a stratului intermediar, accesul este mai facil la cipurile individuale, putându-se identifica și astfel izola mult mai rapid defectele.

Integrarea 2.5D servește ca o punte pragmatică, oferind multe dintre beneficiile suprapunerii 3D dar care evită în același timp toate obstacolele mai dificile specifice integrării pe verticală [45]. Cu toate aceste avantaje enumerate mai sus limitările tehnologiei 2.5D se reflectă în următoarele aspecte:

- Stratul intermediar poate fi costisitor, mai ales dacă este din Siliciu;
- Alinierea și bonding-ul necesită cerințe tehnice ridicate;
- Densitatea maximă nu este comparabilă cu cea a circuitelor integrate 3D.

Mai jos sunt enumerate câteva aplicații uzuale ale circuitelor integrate 2.5D:

- Procesoare high-end (exemple: AMD Epyc/Intel EMIB);
- GPU-uri (ex: Nvidia și AMD cu memorie HBM);
- Servere și supercomputere;
- Sisteme embedded avansate;

- Dispozitive medicale compacte.

1.2. Tehnologii actuale 3D de implementare a circuitelor integrate

La ora actuală sunt 3 metode de implementare a circuitelor integrate tridimensionale: o metodă este prin suprapunerea substraturilor de Siliciu și care necesită precizie în alinierea lor, o altă metodă se face la nivel de circuite integrate individuale, unde se suprapun die-urile individual după ce sunt tăiate din substratul de Siliciu, dar și în această situație este necesară precizie în alinierea circuitelor integrate, iar cea de-a 3 a metoda, în care alinierea nu mai este necesară, o reprezintă tehnologia *monolitic 3D*, unde straturile sunt direct fabricate unul peste altul pe același substrat de Siliciu [46].

Primele două metode de integrare tridimensională, necesită o precizie ridicată în alinierea substraturilor de Siliciu, respectiv circuitelor integrate individuale, iar suprapunerea și conectarea lor se face după fabricarea circuitelor integrate [47–49]. În cazul primei metode unde conectarea se face la nivel de substrat de Siliciu, toate die-urile sunt procesate simultan, iar a doua metodă permite testarea individuală a fiecărui circuit reducând riscul integrării unor componente defecte. Ambele metode folosesc pentru conectarea straturilor, via-uri numite TSV și care nu sunt altceva decât găuri umplute cu material conductor create prin perforarea siliciului [40]. La nivel de substrat de Siliciu, TSV-urile sunt create pentru a realiza conectivitatea electrică între straturile multiple ale unui substrat de Siliciu, iar la nivel de die (de circuit integrat individual) TSV-urile sunt create pentru a conecta cipurile suprapuse într-o arhitectură tridimensională.

1.2.1 Tehnologie de stivuire pe verticală - TSV

Conectarea pe verticală a substraturilor de Siliciu, ori a circuitelor integrate individuale se poate realiza folosind via-urile TSV, care nu sunt practic altceva decât coloane de metal (Cupru, Tunxten, Aluminiu, Nichel) care străpung stratul de Siliciu [50] și asigură conexiunea electrică la bump-urile sau micro-bump-urile metalice de pe stratul anterior [51]. TSV-urile reprezintă componentele cheie de a realiza interconexiunile pe verticală în această tehnologie și care ajută la formarea circuitelor integrate tridimensionale [52]. Aceste conexiuni suplimentare, realizate în a treia dimensiune, reduc întârzierile pe căile de date, reduc consumul de putere și de asemenea oferă un transfer de date între straturi de o mare densitate [53] . Tehnologia în sine a parcurs un drum anevoios și a ajuns la maturitate doar în ultimii ani pentru a putea fi folosită eficient (cost acceptabil și procent de defecte redus) în producție de serie [54–56].

În figura 1.3 se poate observa cum sunt conectate două circuite individuale (2 die-uri) folosind metoda de conectare cu TSV-uri [50].

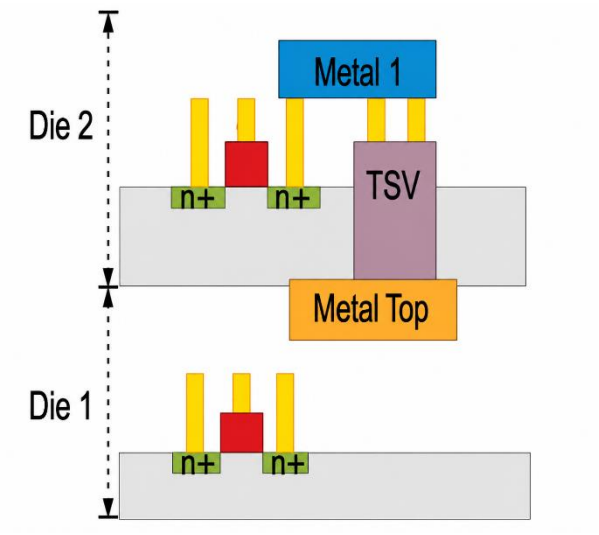


Fig.1.3. Conectarea a două circuite integrate individuale cu TSV [50]

Integrarea tridimensională se referă la o configurație și o tehnică utilizate pentru a integra vertical mai multe cipuri prin lipirea plachetelor și interconectări electrice utilizând găuri de trecere prin siliciu**.** Ideea de a utiliza găuri goale pentru transmiterea semnalului printr-o plachetă a fost propusă pentru prima dată de Shockley în 1958. Ulterior, în 1964, IBM a introdus explicit conceptul de gravare și depunere de metal pentru fabricarea TSV. În 1969, IBM a propus în continuare o metodă de integrare 3D folosind gravarea pe două fețe cu KOH și depunerea de metal pentru a crea TSV-urile, cu lipire metalică utilizată pentru lipirea cipurilor. Cu toate acestea, abia în 1984 a fost realizată stivuirea cipurilor CMOS într-o configurație tridimensională. Introducerea tehnologiei de gravare profundă cu ioni reactivi (DRIE) Bosch, la mijlocul anilor 1990, a făcut posibilă gravarea de găuri verticale și adânci în plachetele de siliciu, permițând fabricarea de TSV-uri cu rapoarte mari de aspect, umplute cu tungsten (W) sau conductori de polisiliciu până la sfârșitul secolului trecut. Începând cu anul 2000, placarea cu cupru (Cu) în găurile adânci a devenit tehnologia principală pentru umplerea TSV-urilor cu rapoarte mari de aspect [50, 57]. Scopul galvanizării este de a obține o tensiune scăzută, fără găuri și goluri în timpul fabricării TSV-urilor [52]. Tehnologia TSV permite o interconectare de înaltă densitate în direcția axei Z, ceea ce reduce semnificativ dimensiunea tridimensională a microsistemului. Lungimea redusă a interconectărilor rezultă, de asemenea, într-un consum mai redus de energie al cipurilor, îmbunătățește viteza de propagare a datelor, reduce întârzierea semnalului și îmbunătățește fiabilitatea interconectărilor electrice ale sistemului [58]. Ca urmare, tehnologia TSV este importantă pentru succesul tehnologiei 3D IC și alături de celelalte tehnologii de integrare 3D are potențialul de a revoluționa industria semiconductorilor. O ilustrare a unui circuit integrat tridimensional 3D IC se poate observa în figura 1.4, unde o tehnologie eterogenă compusă din procesor, memorie, circuit RF,

circuite analogice și senzori, poate fi integrată în același pachet evitând astfel conexiunile lungi care produc întârzieri [57].

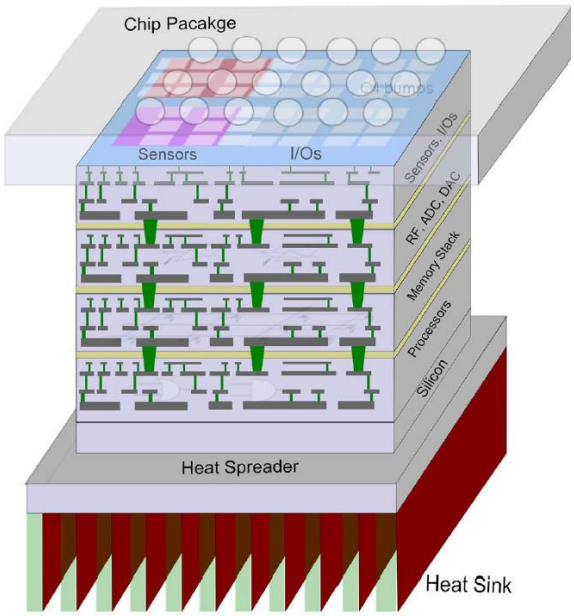


Fig. 1.4. Tehnologie eterogenă într-un circuit integrat [57]

Integrarea 3D prin intermediul via-urilor TSV a fost cunoscută pe scară largă ca o tehnologie promițătoare pentru viitoarele sisteme electronice. Această tehnologie oferea cele mai scurte interconexiuni verticale cu o serie semnificativă de avantaje cum ar fi: consum redus de energie, densitate mare de componente, arie mai mică de integrare comparativ cu integrarea convențională 2D. Interconexiunile verticale sunt formate prin substratul de siliciu (wafer) pentru a permite comunicarea între cipurile stivuite. Aplicațiile sale sunt de asemenea utilizate la o arie largă de dispozitive electronice, care include senzorii MEMS, senzorii de imagine CMOS (CIS), memorii, memorii hibride, circuite de radio-frecvență etc. Pentru majoritatea proceselor de dezvoltare a TSV-urilor, electrodepunerea de cupru este una dintre cele mai importante tehnologii pentru implementarea conexiunilor verticale. Cu toate acestea, deoarece majoritatea TSV-urilor au rapoarte de aspect ridicate iar procesul de umplere este un proces complex și dinamic fizico-chimic, este dificil de a umple TSV-urile fără a rămâne goluri [59]. Chiar și așa, procesul este destul de controlabil astăzi, pentru a permite folosirea lui în producția de masă. Companiile care vin cu produse cu preț mai ridicat sunt dispuse să-și asume costurile pentru a folosit această tehnologie.

În figura 1.5 poate fi observat procesul Bosch utilizat pentru formarea TSV-urilor.

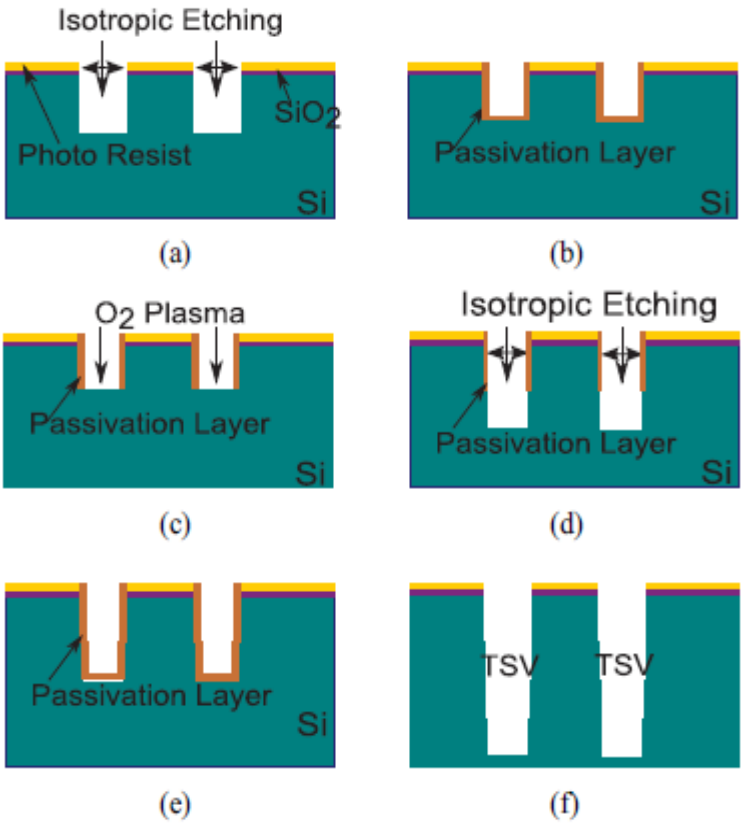


Fig. 1.5. Procesul Bosch utilizat pentru formarea TSV-urilor prin gravură ionică reactivă profundă (DRIE): (a) Oxidare termică, definirea măștii și prima gravură izotropă ; (b) Depunerea stratului de pasivizare (c) Îndepărtarea pasivizării de pe fundul TSV-ului; (d) A doua etapă de gravură izotropă; (e) Repetarea ciclului de gravură/pasivizare; (f) Obținerea secțiunii transversale finale a TSV-ului [57]

Formarea TSV-urilor prin procesul Bosch, bazat pe gravura ionică reactivă profundă, este cea mai utilizată metodă datorită preciziei sale. Practic gravura izotropă cu plasmă alternează cu depunerea unui start de pasivizare pentru a controla profilul pereților laterali. Cu această metodă se pot obține TSV-uri cu diametre până la 5 microni.

În timp ce dispozitivele 3D cu TSV-uri înregistrează progrese semnificative, dimensiunile mari ale TSV-urilor – de ordinul a 3-5 microni în diametru – și grosimea semnificativă a straturilor suprapuse, limitează beneficiile acestei abordări. În ultimii ani, o altă abordare [60] care utilizează un strat subțire – sub 1 micron – pentru stivuire 3D monolitică a straturilor semiconductoare, a devenit fezabilă și promite beneficii mult mai mari decât cele disponibile prin abordarea care folosește TSV-uri. În particular, MIV-urile în procesul de integrare 3D monolitică [61], sunt cu două ordine de mărime mai mici decât TSV-urile și permit o conectivitate verticală mult mai densă, permițând noi arhitecturi de sisteme și randamente (yield) mai mari. În plus, straturile subțiri și

conectivitatea mai densă permit o îndepărtare mai ușoară a căldurii și un preț mai scăzut de fabricație.

1.2.2 Tehnologia Monolitic 3D

O altă tehnologie pentru implementarea circuitelor integrate 3D, asupra căreia și-au concentrat cercetările companii și instituții ca IBM, Standford, CEA-Leti, Monolithic 3D, este tehnologia monolitică 3D.

O ordine cronologică care ilustrează evoluțiile primare în procesul de fabricare în tehnologie monolitic 3D este prezentată în figura 1.6 [62], unde se poate observa că dezvoltarea unui proces de fabricare secvențial pentru a construi mai multe straturi de tranzistor pe un singur substrat, datează încă de la sfârșitul anilor 80.

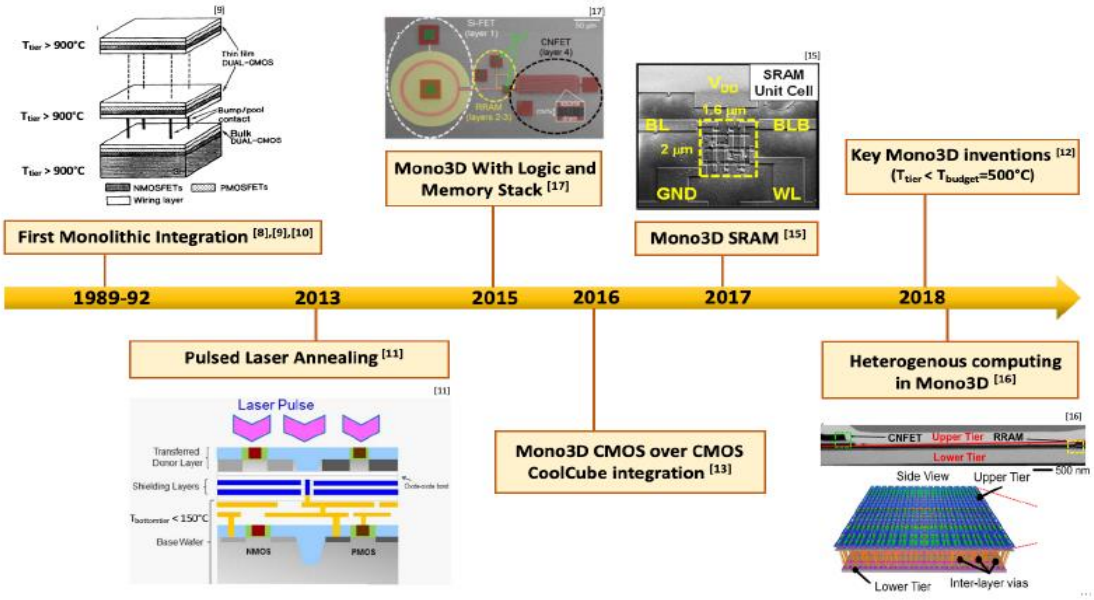


Fig. 1.6 Cronologie a principalelor etape de dezvoltare în procesul de fabricare al circuitelor integrate Monolitic 3D [62]

Aceste studii timpurii s-au bazat pe recristalizarea cu fascicul laser pentru a forma multiple straturi de tip silicon-izolator (SOI) deasupra substratului de tip p. Un singur strat de dispozitive conținea doar un singur tip de tranzistor, reducând astfel numărul de pași de proces aferenți formării zonelor de tip p/n și implantării ionice. În loc să aibă atât tranzistoare de tip n, cât și de tip p în același strat (cum se întâmplă în tehnologiile CMOS tradiționale), fiecare strat avea doar un singur tip de tranzistor. Acest lucru simplifica procesul tehnologic, deoarece nu mai era nevoie să se creeze regiuni distincte (well-uri) pentru fiecare tip de tranzistor și nici să se facă implanturi ionice diferite pentru n și p. Practic, stratul era mai ușor de fabricat, cu mai puține etape complexe.

Deoarece aceste straturi erau expuse la temperaturi de aproximativ 900°C în timpul fabricării dispozitivelor, interconectările din polisiliciu dopat au fost utilizate în aceste implementări timpurii, datorită capacității lor de a rezista la temperaturi înalte. Funcționarea corectă a dispozitivelor a fost demonstrată prin măsurarea caracteristicilor I-V în fiecare strat. Totuși, dispozitivele SOI au prezentat o variație a tensiunii de prag de aproximativ 4 ori mai mare [62–64].

Tehnologiile SOI s-au folosit și se folosesc în noduri tehnologice ce nu coboară sub 22 nm. Aceste tehnologii se folosesc pentru produsele unde se dorește un consum de leakage redus (perioada în care circuitul integrat este alimentat, dar semnalul de ceas este oprit sau funcționează la frecvență redusă). Aceste tehnologii sunt disponibile în procesele de fabricație oferite de producători de circuite integrate precum TSMC și GlobalFoundries.

Tehnologia monolitic 3D este o tehnologie avansată, folosită la construirea circuitelor tridimensionale, în care straturile de tranzistoare sunt construite unul peste altul pe același substrat de Siliciu fără a mai fi necesară suprapunerea și alinierea substraturilor de Siliciu ori a circuitelor integrate individuale, ca în cazul tehnologiei TSV, unde aceste procese sunt necesare pentru a crea circuite integrate tridimensionale [65]. Procesul de fabricare a circuitelor integrate cu tehnologie monolitic 3D începe cu stratul standard de silicon, folosit ca substrat, peste care se construiește primul strat de tranzistoare. După ce primul strat de tranzistoare și interconexiunile dintre ele sunt fabricate, se utilizează un strat dielectric ILD care izolează stratul de tranzistoare de stratul inferior. Un strat subțire de siliciu este apoi depus peste ILD folosind o tehnică numită LTCVD, iar acest strat de siliciu va fi substratul următorului strat de tranzistoare. Procesul de fabricare al celui de-al doilea strat de tranzistoare este identic cu cel de fabricare al primului strat, implicând dopare, modelare și gravare. Interconexiunea între straturile de tranzistoare se face utilizând MIV-uri, care sunt asemănătoare ca dimensiune cu via-urile folosite la interconexiunile dintre metale în tehnologia planară 2D.

Procesul de fabricare a tehnologiei monolitic 3D este ilustrat în figura 1.7 [66] unde sunt prezentate toate etapele pornind de la crearea stratului inferior, procedeu asemănător cu tehnologie 2D, peste care este depozitat un strat subțire de siliciu și apoi urmează crearea conexiunilor verticale MIV-urile cu care se realizează conexiunea la stratul superior.

Tehnologia monolitic 3D, spre deosebire de tehnologia TSV, nu folosește pentru conexiunile pe verticală via-uri ca TSV-urile, care sunt de ordinul micronilor, ci folosește via-urile MIV care sunt de ordinul nanometrilor deci sunt mult mai mici decât TSV-urile. MIV-urile sunt implementate în timpul procesului de fabricare a unui singur substrat de Siliciu, realizând conexiunile între straturile fabricate secvențial.

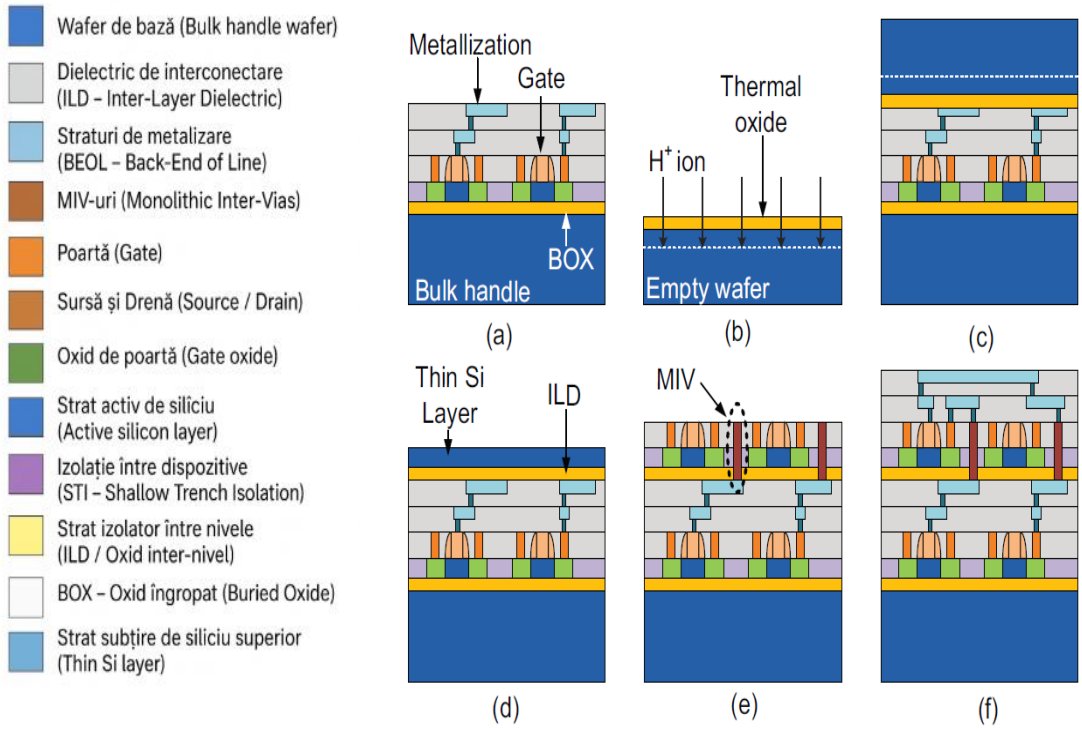


Fig. 1.7 Procesul de fabricare a tehnologiei Monolitic 3D:
(a) stratului inferior asemenea cu 2D, (b), (c), (d) Crearea stratului de siliciu,
(e) Implementarea MIV-urilor,
(f) Conectarea cu stratul superior [66]
Legenda a fost adăugată de autoare

O ilustrare a dimensiunilor MIV-urilor folosite pentru conexiunea pe verticală în tehnologia monolitic 3D poate fi observată în figura 1.8:

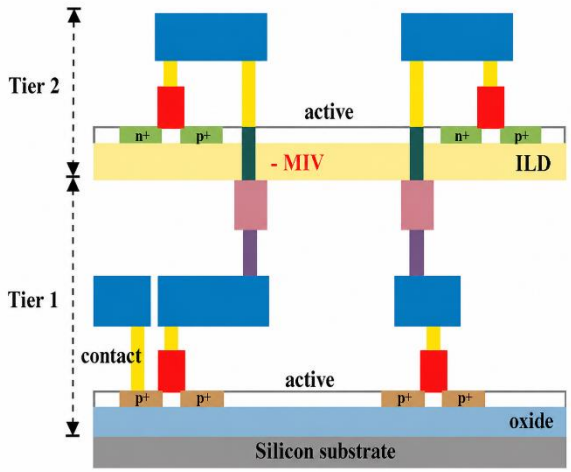


Fig. 1.8. MIV realizat între 2 straturi în tehnologia monolitic 3D [50]

Integrarea monolitică 3D prezintă un număr de avantaje ce sunt enumerate în următoarele paragrafe.

Diferite arii logice active într-un circuit integrat 3D monolitic pot diferi din punct de vedere al dimensiunii minime folosite în procesul litografic, cele mai mici fiind mai rapide dar mai scump de realizat. Într-un circuit integrat 3D fiecare strat poate fi fabricat printr-un proces propriu, de exemplu, folosind propriul set de reguli de proiectare, față de circuitele integrate 2D unde toate tranzistoarele există pe același strat și vor fi realizate prin același proces. Conform procesului de partiționare a circuitelor integrate, fiecare strat poate avea la bază tipuri de circuite ce pot fi integrate împreuna într-un mod eficient. Într-un circuit integrat modern, costul litografiei influențează puternic costul produsului finit [67, 68]. În consecință, poate fi mai eficientă maximizarea numărului de straturi ce utilizează un proces litografic mai ieftin. În contrast, minimizarea numărului de straturi poate impune folosirea celui mai agresiv și scump proces litografic. Un strat ar putea conține circuite de memorie similare unde se folosește un proces litografic bazat pe spațiere ce duce iarăși la scăderea costurilor. Logica lentă a unui dispozitiv ar putea fi partiționată pe straturi ce folosesc un proces litografic mai vechi, dar mai ieftin. Asemănător ar putea fi tratate și modulele I/O, unde pot fi folosite procese litografice mai simple.

Un alt avantaj al integrării 3D este partiționarea memoriilor pe diferite straturi ale unui circuit integrat 3D. De exemplu, straturile pot varia în funcție de memoria pe care o implementează (de ex. memorie volatilă versus memorie nevolatilă, memorie dinamică versus memorie statică).

Integrarea monolitică 3D permite partiționarea circuitului în elemente pe un singur strat, comparativ cu elementele ce se întind pe mai multe straturi adiacente. Similar cu alte metode de partiționare specializate, această caracteristică se poate traduce printr-un "cost" sau poate fi folosită pe post de constrângere în procesul de partiționare. Mai mult, dacă sunt prezente atât obiecte pe un singur strat, cât și obiecte pe mai multe straturi, blocul de partiționare poate selecta varianta cea mai bună de integrare pe baza unor reguli de proiectare globale.

Unul dintre principalele avantaje ale integrării monolitic 3D este integrarea diferitelor materiale pe lângă straturile dispozitivelor din siliciu (Si) de înaltă performanță. Au fost diverse materiale alternative evaluate ca o alternativă la Si ca și canale superioare. Oxizii metalici prezintă mobilitatea mare a electronilor, sunt transparenți și flexibili, făcându-i potriviți pentru TFT în tehnologiile de afișare. Materialele 2D oferă proprietăți unice electronice care promit performanțe îmbunătățite în circuitele monolitic 3D. Compușii Ge și III-V , care au cea mai mare mobilitate a electronilor și a găurilor, comparativ cu Siliciu, pot contribui la viteze mai mari de comutare a tranzistoarelor. Aceste materiale alternative sunt impactate de limitările specifice ale Siliciului, cum ar fi problemele termice și provocările de scalare, de asemenea permițând proiectarea de dispozitive și circuite noi [69].

Deși un efort uriaș este făcut în direcția proiectării circuitelor integrate tridimensionale cu atenție la metodologii de proiectare și testare, algoritmi de optimizare, adaptarea programelor la cea de-a treia dimensiune, modelare și analiză 3D [70], există încă provocări fundamentale și multe probleme de rezolvat pentru adaptarea circuitelor 3D în piața semiconductoarelor. În figura 1.9 [71] poate fi analizată o comparație între cele 2 tehnologii de realizare a interconexiunilor pe verticală, în figura 1.9 (a) se poate observa că diametrul TSV-urilor este de ordinul micrometrilor, în timp ce în figura 1.9 (b) sunt puse in evidență MIV-urile, cu care se realizează conexiunile pe verticală în tehnologia monolitic 3D și care sunt de ordinul nanometrilor.

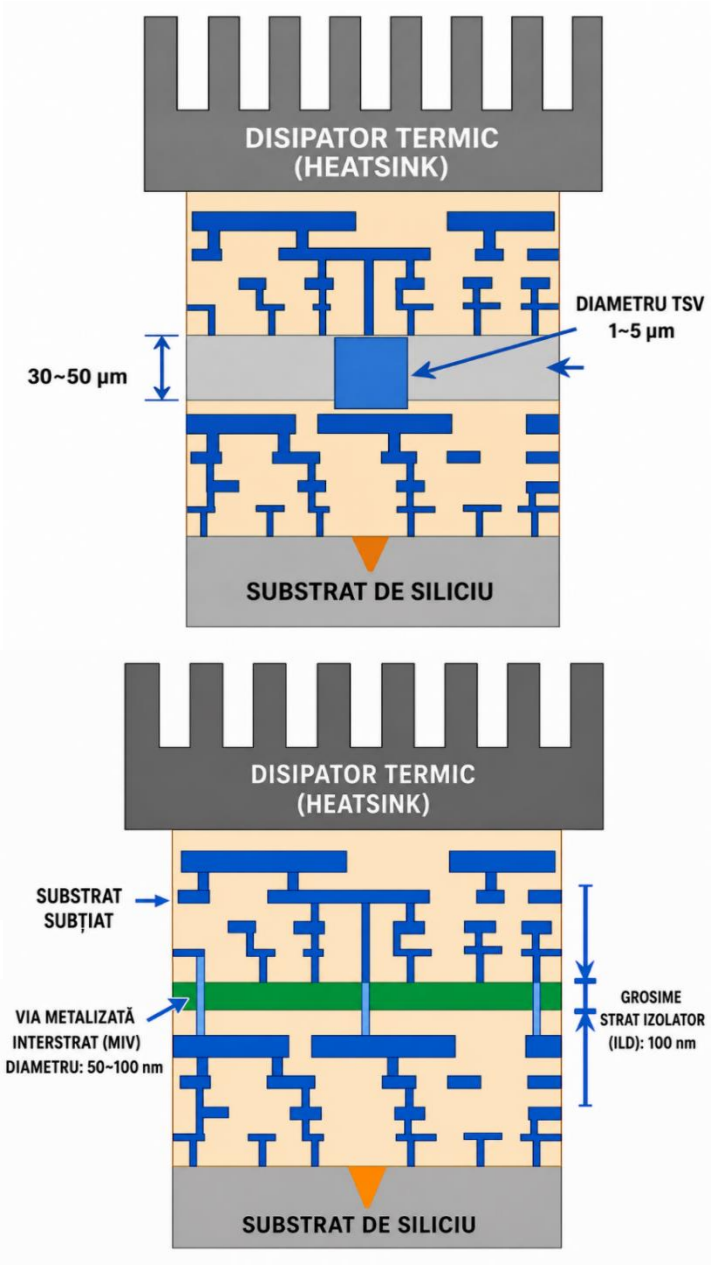


Fig. 1.9. Comparație între interconexiunile pe verticală: (a) Interconexiuni cu TSV, (b) Interconexiuni cu MIV [71]

Tehnologia monolitică de integrare 3D este o abordare demonstrată fizic și viabilă din punct de vedere tehnic, dar rămâne predominantă în domeniul cercetării avansate și al prototipurilor de laborator experimentale [72]. În timp ce mai multe implementări au fost fabricate și testate cu succes, pentru a dovedi conceptul (de ex. stivele CMOS suprapuse peste CMOS sau configurațiile logică suprapusă peste memorii), totuși ele rămân limitate la setările academice și pre-comerciale. Până în prezent integrarea monolitică 3D nu a fost adoptată în produsele electronice comerciale principale, cum ar fi telefoanele smart, procesoarele de înaltă performanță sau SOC-urile de calitate pentru consumatori. Absența unei astfel de adoptări este din cauza mai multor bariere tehnologice și economice nerezolvate printre care sunt cele enumerate mai jos:

1. Constrângeri termice: fabricarea secvențială a straturilor aparținând circuitului pe un singur substrat de Siliciu necesită procesare la temperatură scăzută pentru a evita deteriorarea tranzistoarelor de pe stratul anterior. Dezvoltarea proceselor de fabricare la temperaturi joase care mențin încă yieldul circuitelor de înaltă performanță este o provocare semnificativă.
2. Limitări EDA și a flow-ului de proiectare: Instrumentele EDA actuale folosite în implementarea circuitelor integrate sunt optimizate în primul rând pentru arhitecturi planare [73]. Sprijinirea completă a integrării monolitice 3D – unde sincronizarea semnalului între straturi, problemele termice și distribuția energiei sunt strâns cuplate între ele – necesită progrese semnificative în lanțul de instrumente pentru proiectarea circuitelor integrate.
3. Randament: asigurarea randamentului ridicat în stivele de tranzistoare multi-strat este complexă din cauza sensibilității crescute la defecte.
4. Testabilitate: acces limitat pentru structurile de testare convenționale. Integrarea metodologiilor eficiente ca DFT-ul într-un flux monolitic 3D rămâne un domeniu activ de cercetare.
5. Costul de producție și scalabilitatea: În ciuda avantajelor teoretice în ceea ce privește densitatea și performanța, rentabilitatea și scalabilitatea proceselor monolitice 3D la volume comerciale nu sunt încă competitive cu tehnologiile de stivuire 3D planare sau eterogene consacrate.

1.3 Dispozitive comerciale bazate pe tehnologia TSV și instrumente 3D EDA

În prezent, tehnologia de integrare 3D bazată pe conexiuni TSV este implementată în mai multe dispozitive comerciale, în special în memorii de înaltă performanță și în alte aplicații destinate sistemelor de calcul de înaltă performanță. Cu toate acestea, utilizarea sa rămâne limitată

la anumite categorii de produse, deoarece procesele de fabricație implică o complexitate ridicată și costuri suplimentare comparativ cu tehnologiile convenționale. Din acest motiv, tehnologia TSV nu este adoptată pe scară largă în întreaga industrie a semiconductorilor. În continuare sunt prezentate câteva exemple reprezentative de dispozitive comerciale care utilizează această tehnologie.

1. Memorii flash 3D NAND:
    - SSD-uri : Unul dintre primele exemple ale succesului integrării 3D în industria semiconductorilor îl reprezintă memoriile flash care au experimentat o schimbare de paradigma în ultimii ani. SSD-urile bazate pe tehnologia 3D NAND sunt printre cele mai răspândite și utilizate dispozitive comerciale și de la primul produs comercial în 2014 au progresat rapid crescând numărul de straturi de la 24 la 176 și chiar mai multe și la 4 biți pe celulă de memorare [74, 75]. Această tehnologie permite crearea de memorii cu o densitate foarte mare, prin suprapunerea mai multor straturi de celule de memorie într-un singur cip. Mărci precum Samsung, Western Digital și Micron comercializează SSD-uri care folosesc 3D NAND.
2. Memorii HBM:
    - Memoriile HBM sunt utilizate în plăcile grafice de înaltă performanță, cum ar fi cele comercializate de companiile AMD și NVIDIA Această tehnologie permite o lățime de bandă foarte mare și un consum redus de energie, prin stivuirea verticală a modulelor de memorie și utilizarea TSV-urilor pentru interconectare [76].
3. Procesoare avansate:
    - Pe piață se comercializează deja procesoare implementate în tehnologie 3D. În mod particular, procesoarele folosite în aplicații de inteligență artificială și supercomputere încep să folosească tehnologia 3D pentru a integra mai multe straturi de circuite logice și memorie pe aceeași pastilă de siliciu. Intel produce procesoare 3D cu tehnologie Foveros care permite suprapunerea die-urilor logică peste logică [77], iar AMD folosește tehnologia 3D V-cache care folosește TSV-uri pentru a adăuga memorie cache deasupra procesorului [78].
4. Memorii HMC:
    - Memoriile HMC sunt o altă tehnologie de memorie 3D care suprapune module DRAM pe verticală și le conectează cu TSV-uri, permițând lățimi de bandă foarte mari și un consum energetic optimizat. Aceste tehnologii sunt folosite în servere de înaltă performanță și centre de date. Aceste memorii au fost dezvoltate de compania Micron Technology în parteneriat cu alte companii precum Intel [79].

5. Dispozitive mobile:
    - Telefoanele mobile, tabletele și dispozitivele electronice mobile, folosesc tehnologia “Package on Packge”, unde procesorul și memoria sunt fabricate separat și apoi sunt suprapuse. Această tehnologie este utilizată de mărci precum Apple și Samsung în produsele lor mobile.

În Tabelul 1.1 sunt prezentate principalele instrumente software 3D EDA utilizate pentru proiectarea circuitelor integrate tridimensionale. Toate soluțiile analizate sunt caracterizate prin costuri ridicate de licențiere, conform estimărilor bazate pe informații din industrie, întrucât prețurile oficiale nu sunt disponibile public.

Tabel 1.1 Instrumente 3D EDA

Tehnologie	**Funcționalitate**	**Tool Cadence**	**Tool Synopsys**	**Cost estimativ/an**
TSV	Planificarea conexiunilor verticale (TSV Planning)	Integrity 3D-IC Platform	3DIC Compiler	Foarte ridicat
TSV	Integrarea și conectivitatea între straturi de siliciu	Integrity 3D-IC Platform	3DIC Compiler	Foarte ridicat
TSV	Analiza performanței, consumului și efectelor termice	Integrity 3D-IC Platform	3DIC Compiler + PrimeTime	Foarte ridicat
Monolithic 3D (MIV)	**Platformă dedicată de proiectare 3D**	**Nu există**	**Nu există**	—

Costurile ridicate ale platformelor dedicate proiectării bazate pe TSV și absența unor instrumente comerciale dedicate tehnologiei Monolitic 3D reprezintă o limitare importantă în procesul de proiectare 3D. În acest context, dezvoltarea unei metodologii bazate pe instrumente EDA standard utilizate în industrie constituie o direcție justificată și necesară, reprezentând obiectivul principal al prezentei tezei.

1.4 Instrument de simulare pentru tehnologia Monolitic 3D

MonolithIC 3D Inc. a dezvoltat un instrument CAD open-source [82] folosit pentru simularea circuitelor integrate 2D și 3D, scris în Java și care poate fi rulată pe mai multe sisteme de operare (Windows, Linux, MacOS). Acest instrument include modele pentru circuite integrate realizate în tehnologie 3D monolitică și permite efectuarea simulărilor necesare pentru înțelegerea și evaluarea acestei tehnologii emergente, cu un potențial semnificativ de dezvoltare.

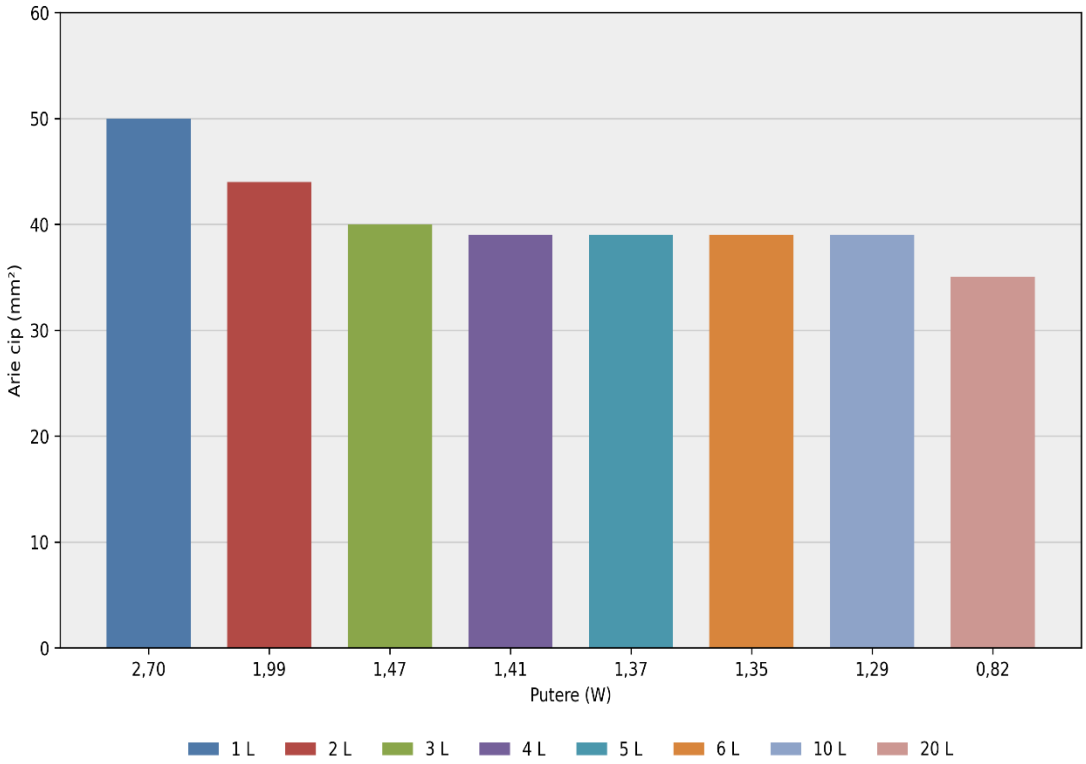


Fig. 1. 10. Aria minimă vs. Putere consumată pentru diferite numere de straturi (32nm, 1 GHz, 196M porți) [82]

Figura 1.10 prezintă variația ariei minime de siliciu în funcție de numărul de straturi pentru un circuit integrat realizat în tehnologie 32 nm, care conține 196 milioane de porți logice și o frecvență de lucru de 1 GHz. Prin creșterea numărului de straturi, rezultatele simulării arată o scădere medie a ariei de siliciu cu 22.6%. Mai mult, prin folosirea unui număr de 5 straturi, nu numai că obținem o reducere a ariei de siliciu cu 20%, dar obținem și o reducere a consumului de putere cu 50%.

Consumul de putere scade cu numărul de straturi, în principal datorită scăderii numărului de circuite repetoare (Figura 1.11) . Rezultatele prezentate sunt pentru un circuit integrat realizat în tehnologie de 22 nm, circuit care conține 196 milioane de porți logice. Se observă că odată cu creșterea frecvenței de lucru, un număr mai mare de straturi duce la o economie mai mare de putere. Pentru o

frecvență de 3 GHz, dacă se folosesc 16 straturi, consumul de putere se reduce de 3,7 ori și pentru o frecvență de 2,5 GHz, dacă se folosesc 8 straturi, puterea consumată este redusă la jumătate. În medie, dacă se folosesc două straturi, se obține o reducere a consumului de putere cu 50% și ajunge până la 60% pentru 16 straturi.

Pe baza acestui tool dezvoltat de MonolithIC 3D Inc se poate face o estimare a creșterii performațelor circuitelor în funcție de tehnologie, numărul de straturi respectiv numărul de componente.

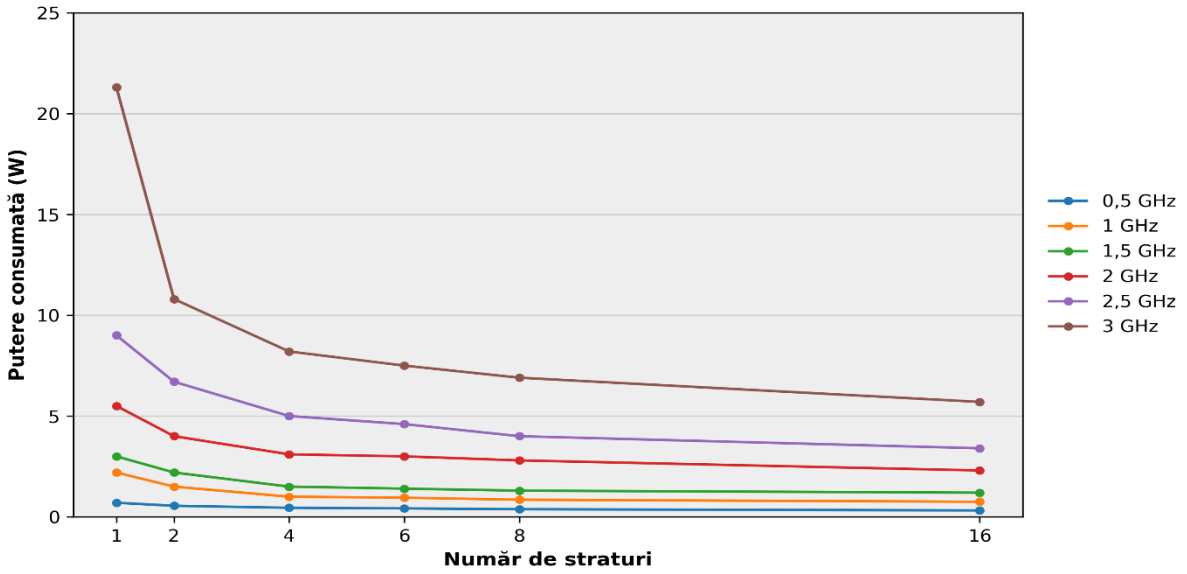


Fig. 1.11 Putere consumată vs. Frecvență de lucru pentru diferite numere de straturi de (22 nm, cu 196 milioane de porți) [82]

1.5 Concluzii la capitolul 1

Integrarea tridimensională a circuitelor electronice vine cu o serie de avantaje față de integrarea clasică 2D și deschide de asemenea o serie de perspective pentru cercetare într-o multitudine de direcții, pornind de la tehnologia de integrare 3D, procesul de fabricare a circuitelor integrate și până la modificarea, respectiv adaptarea programelor folosite în procesul de implementare și manufacturare a unui circuit integrat.

În urma analizării metodelor de integrare 3D expuse mai sus și comparând cu rezultatele obținute folosind implementarea circuitelor integrate în tehnologie 2D, se menționează următoarele avantaje, care se pot obține folosind implementarea circuitelor în tehnologie 3D.

1. Consumul de putere în integrarea 3D este mai redus: lungimea interconexiunilor între componentele circuitului se reduce, prin urmare rezistențele și capacitățile au valori mai mici și astfel necesită mai puțină energie pentru transmiterea datelor. Consumul redus de

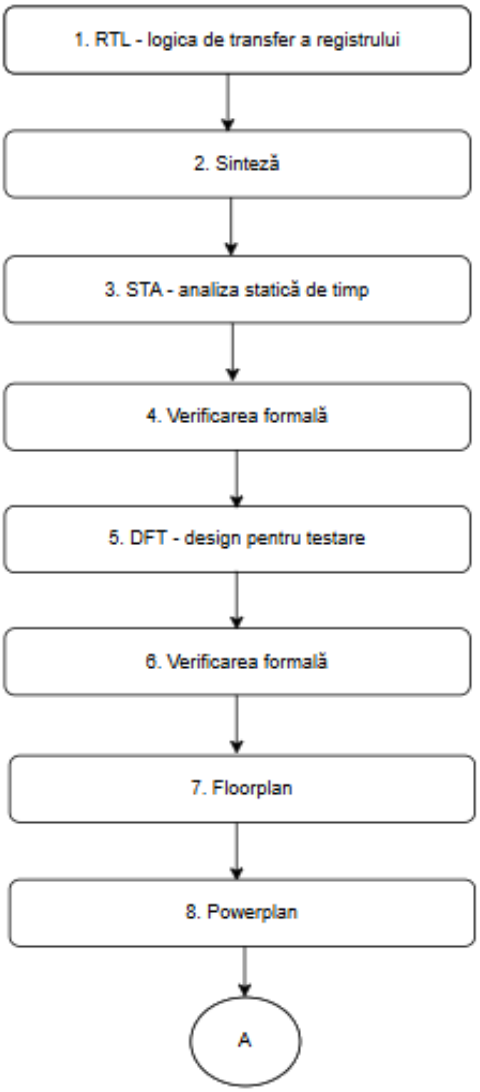
energie, evidențiat în toate analizele comparative ale circuitelor macro [80] sau micro, subliniază importanța evaluării performanțelor și contribuie semnificativ la creșterea duratei de viață a bateriilor în dispozitivele mobile, demonstrând modul în care optimizarea parametrilor poate aduce beneficii practice și măsurabile.

2. Viteza de propagare a datelor crește: interconexiunile verticale între componentele circuitului sunt mai scurte decât în cazul 2D, iar întârzierile pe fire vor scădea, ceea ce va conduce la mărirea frecvenței circuitului integrat. O frecvență mărită a procesării datelor, va ajuta exponențial era inteligenței artificiale spre care sunt îndreptate majoritatea cercetărilor la ora actuală.
3. Dimensiunea circuitului integrat scade: abordarea integrării verticale duce la o arie totală a circuitului integrat mai mică, ceea ce este avantajos pentru aplicațiile portabile cum ar fi telefoanele mobile, tabletele, laptopurile etc.
4. Abordarea tridimensională a implementării circuitelor electronice facilitează integrarea eterogenă (partea logică, senzori și memorii pot fi implementate pe aceeași pastilă de siliciu), ceea ce oferă performanțe ridicate și optimizare la nivel de arie și consum de energie. Integrarea eterogenă poate conține de asemenea partea logică, partea analogică și partea de gestionare a energiei pe același strat, iar astfel de soluții ca cele utilizate la scara macro [81] pot fi adaptate pentru a inspira soluții de reglare a tensiunii și de decuplare adaptate fiecărui strat în circuitele integrate 3D.
5. Tehnologia monolitică 3D permite amestecarea de tehnologii semiconductoare diferite într-un singur dispozitiv la costuri mici. De exemplu, sistemele care conțin senzori, blocuri analogice, diferite tipuri de memorie (DRAM, Flash) și logică pot fi suportate de această tehnologie. Stivele monolitice tridimensionale permit amestecarea straturilor proiectate cu linii de fabricație mult mai vechi, complet amortizate, cu straturi logice rapide și avansate, dar scumpe. Cu alte cuvinte, permite crearea de compromisuri mai eficiente între caracteristicile de proiectare și costurile de proiectare.

2. METODOLOGIE DE IMPLEMENTARE A UNUI CIRCUIT INTEGRAT 2D

În capitolul acesta este descris un proces complet de implementare al unui circuit integrat planar 2D, modelat și optimizat cu informații obținute după rularea mai multor proiecte reale (proiecte în cadrul cărora au fost fabricate circuite integrate care sunt acum utilizate pe piață) și în care s-au parcurs toate etapele de producere a circuitelor integrate. Pentru o înțelegere mai clară a întregului flux de lucru s-au folosit atât imagini cu un caracter general descriptiv, cât și imagini generate prin intermediul software-ului utilizat pentru acea etapă, programe încorporate în instrumentul IC Compiler de la Synopsys. Scriptul utilizat la rularea comenzilor în partea de BACKEND este cel din ANEXA 1.

Etapele ce trebuie parcurse atunci când se implementează un circuit integrat digital pot fi integrate într-un flux de lucru urmând pașii enumerați în diagrama din figura 2.1 de mai jos :



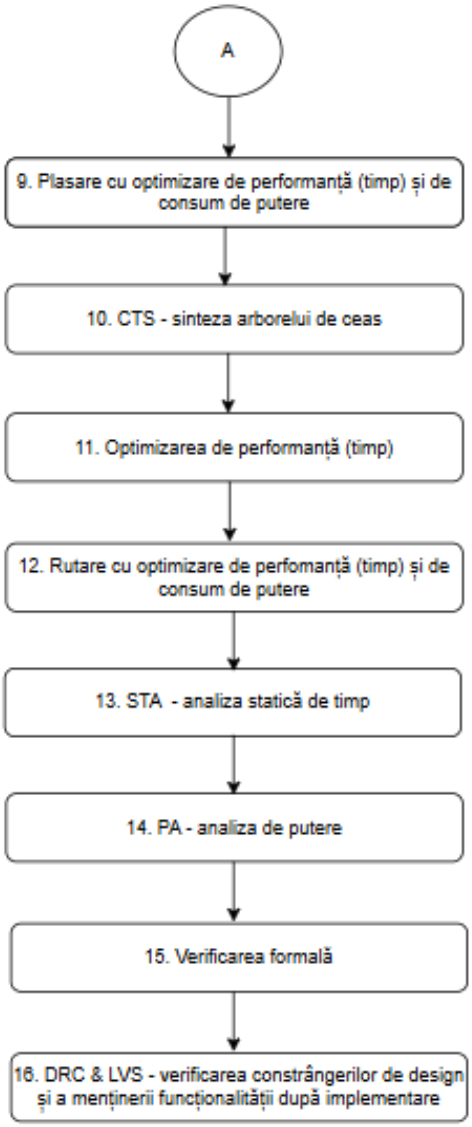


Fig. 2.1. Diagrama etapelor ce trebuie parcurse pentru implementarea unui circuit integrat

Procesul de implementare a unui circuit integrat cuprinde 2 părți importante numite Frontend și Backend și fiecare separat are o importanță deosebită în obținerea unui circuit integrat optim care să îndeplinească toate cerințele de proiectare a circuitului respectiv. În prima etapă numită Frontend sunt incluse primele 6 puncte din diagramă și se axează exclusiv pe informațiile despre funcționarea logică a circuitului din codul descris într-un limbaj de descriere hardware, pe aria bazată pe dimensiunea celulelor din librăria logică, precum și viteza și consumul de putere estimate din librăria logică. Primele 6 puncte se găsesc descrise în diagrama din figura 2.2, unde sunt reprezentați în detaliu pașii ce trebuie parcurși pentru a obține un netlist valid și optim cu care să se poată trece în partea a doua a flow-ului de proiectare, partea de Backend.

În această primă etapă Frontend lipsesc informații despre forma fizică a celulelor din circuit, despre aria în care este plasat circuitul, despre lungimea net-urilor folosite pentru interconectare, precum și întârzierile care apar după plasarea și rutarea circuitului. Toate aceste informații care lipsesc în prima parte, apar în etapa a doua numită Backend și în care sunt incluse etapele de la 7 până la 16.

Înainte de parcurgerea întregului flux de lucru de implementare a unui circuit integrat, o etapă premergătoare este cea în care este verificată și validată librăria tehnologică în care se implementează circuitul. O parcurgere rapidă a fluxului de lucru, prin care un cod HDL sau chiar un netlist împreună cu tehnologia trec prin toate etapele, ajută la înțelegerea problemelor ce pot apărea în funcție de complexitatea circuitului ori a tehnologiei. Se pot lua decizii după această parcurgere rapidă a procesului de lucru, care să ajute la obținerea unui circuit integrat optim, în legătură cu componentele tehnologice și compatibilitatea lor cu cerințele circuitului ce urmează a fi implementat. Acest proces este adesea realizat înaintea unei prime versiuni funcționale a circuitului integrat, tocmai pentru a identifica potențiale probleme care nu apar în simulările logice, dar care pot afecta fabricarea sau performanța finală a cipului.

Tipurile de verificări efectuate pe o librărie nouă sunt cele enumerate mai jos [83]:

1. Verificări fizice (DRC și LVS):
    - Se aplică regulile de design pentru a verifica dacă layout-ul fiecărei celule respectă restricțiile impuse de tehnologia procesului.
    - Se efectuează LVS pentru a confirma că layout-ul fizic corespunde schematicului logic al cicuitului.
2. Validare a caracterizării librăriilor (verificarea fișierelor .lib):
    - Fișierele .lib sunt analizate pentru a verifica corectitudinea informațiilor de timp, consum de energie, și funcționalitate.
    - Se pot rula scripturi automate pentru a identifica valori lipsă sau greșite.
3. Verificări de compatibilitate cu instrumentele utilizate la implementare:
    - Se încarcă librăria în instrumentele de plasare și rutare și în intrumentele care fac analiza de timp STA (Static Timing Analysis) pentru a verifica dacă este complet funcțională.
    - Se generează netlist-uri de test și se execută implementări de tip test pentru a observa posibile erori de procesare.
4. Validare funcțională și semnal de referință:
    - Celulele sunt simulate funcțional în condiții extreme (se verifică setup in worst-case și hold în best-case, glitchurile, etc).

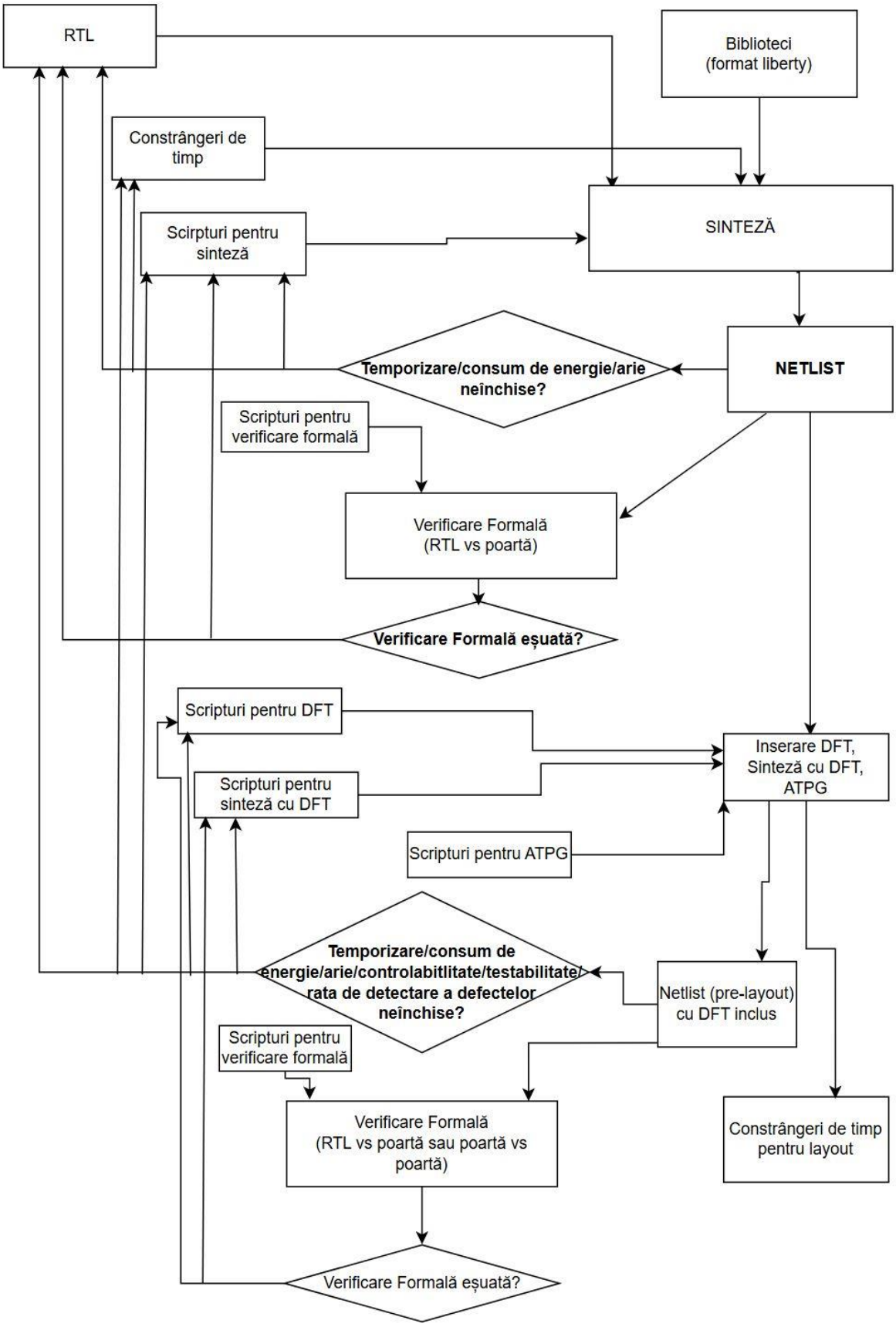


Fig. 2.2. Diagrama care cuprinde etapele detaliate din etapa de Frontend

2.1 Sinteza circuitului, STA, implementarea DFT și verificarea formală

După descrierea funcționalității circuitului într-un limbaj HDL, se face sinteza circuitului, care reprezintă un proces de transformare a circuitului descris HDL într-un netlist la nivel de celulă tehnologică, ținând cont de constrângerile specificate. Sinteza circuitului se face în 3 pași și anume:

- Interpretarea codului: codul RTL este tradus în reprezentare logică independentă de tehnologie. Logica convertită este disponibilă sub formă de ecuație booleană.
- Optimizare: ecuația booleană este optimizată.
- Translarea tehnologică: ecuațiile logice booleene independente de tehnologie sunt mapate pe porțile logice din biblioteca dependentă de tehnologie, pe baza constrângerilor de proiectare și a bibliotecii de porți tehnologice disponibile (există biblioteci cu porți mai complexe ori specifice diferitelor cerințe de proiect: multi-voltage, multi bit, etc).

Ulterior, circuitul generat la nivel de poartă este optimizat logic pentru a îndeplini țintele sau obiectivele stabilite conform constrângerilor utilizatorului. Frecvența ceasului și aria circuitului sunt principalele obiective care trebuie îndeplinite prin operația de sinteză. În figura 2.3 pot fi observați pașii parcurși în cadrul procesului de sinteză:

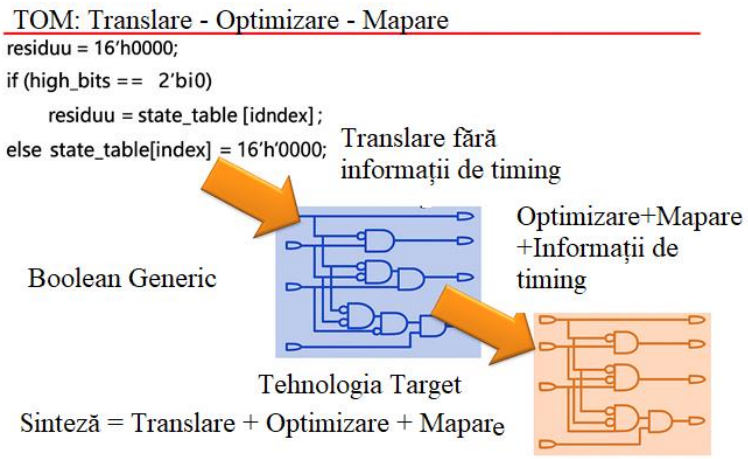


Fig. 2.3. Etapele principale din cadrul procesului de sinteză

Fișierele de intrare utilizate în cadrul procesului de sinteză includ codul RTL, biblioteca tehnologică și fișierul cu constrângeri, în timp ce fișierele de ieșire sunt netlistul și noul fișier de constrângeri. O sinteză cât mai optimă se realizează atunci când se ține cont de informațiile ce provin din partea de backend [84, 85].

După finalizarea procesului de sinteză se face o analiză de timp a circuitului (STA), în care

sunt verificate toate căile de timp, căi ce sunt împărțite în funcție de punctul de plecare și cel de sosire, precum și de tipul semnalului care le parcurge:

- ✓ Căile de date: sunt căile cele mai importante dintr-un circuit pentru că trebuie să respecte constrângerile de setup și hold pentru a evita problemele de sincronizare;
- ✓ Căile de ceas: sunt căile parcurse de semnalul de ceas;
- ✓ Căile de reset: sunt căile parcurse de semnalul de reset;
- ✓ Căile asincrone : sunt căile de timp ce nu depind de frecvența semnalului de ceas.

Există și alte categorii specifice de căi de timp care necesită o atenție specială când se face analiza statică de timp a circuitului, iar aceste sunt:

- ✓ Căile de timp critice;
- ✓ Căile de timp false;
- ✓ Căile care necesită mai multe perioade de ceas;
- ✓ Calea de timp cea mai scurtă și cea mai lungă din circuit.

Analiza statică de timp este o metodă de validare a performanțelor de timp a circuitului prin verificarea tuturor căilor de timp, necesară pentru a asigura frecvența de lucru proiectată pentru circuit și a evita astfel problemele de sincronizare. STA permite identificarea problemelor de sincronizare și anume: pot fi identificate *întârzierile excesive* (Setup Violations) a căror consecință este capturarea unor date greșite de către bistabile, ori *probleme de menținere a datelor* (Hold Violations) unde datele se schimbă prea rapid, înainte ca tranzistoarele să stabilizeze semnalul. În timpul analizei de timp, se face evaluarea întârzierilor din circuit pe baza modelelor de timp din librării, fără a fi necesară simularea logică a circuitului.

După obținerea rezultatelor dorite în procesul de sinteză, pe baza verificările de timp și arie, urmează verificarea funcționalității circuitului, conform descrierii acestuia în limbaj HDL, descris pe baza specificațiilor. Aproximativ 70% din timpul și costurile asociate procesului de proiectare sunt alocate activităților de verificare și validare. Devine foarte important de verificat corectitudinea circuitelor. Circuitul complex modern conține câteva sute de milioane de tranzistoare, iar verificarea unui astfel de sistem complex într-un interval mai scurt de timp devine un factor dominant înainte de a ajunge la nivelul siliciului. Conceptul principal al verificării formale nu este de a simula unii vectori, ci de a dovedi corectitudinea funcțională a unui circuit.

În procesul de verificare formală, circuitul sub testare este comparat cu circuitul inițial descris în limbaj de descriere hardware. Circuitul inițial este considerat circuit de referință și este, de asemenea, numit ca referință de aur. Atât circuitul sub testare cât și referința sunt reduse la ecuații booleene și sunt comparate între ele astfel:

- ✓ RTL versus Netlist;

- ✓ pre routed Netlist versus post routed Netlist;
- ✓ Netlist versus ECO Netlist.

După verificarea formală post sinteză, urmează pregătirea circuitului pentru DFT. Proiectarea scanării interne este una dintre cele mai utilizate tehnici DFT [86–88], iar principiul acestei tehnici este de a modifica elementele secvențiale deja existente în circuit ca în figura 2.4, pentru a susține capacitatea de schimbare, practic să devină accesibile pentru testare. Pe lângă funcționarea normală a celulelor secvențiale este adăugată această nouă funcție care permite legarea lor în lanțuri seriale și astfel sunt obținute registrele de scanare. Aceste registre permit testarea detaliată a stărilor interne ale circuitului, facilitând astfel identificarea defectelor în timpul procesului de testare. Deși înlocuirea bistabilelor fără scan cu bistabile cu scan conduce la mărirea întregii arii a circuitului integrat, această metodă permite testarea mai eficientă a funcționării circuitului, ajutând la detectarea și prevenirea blocărilor în stările logice “1” sau “0”. Astfel se reduc semnificativ timpul și costurile aferente procesului de validare a circuitului integrat.

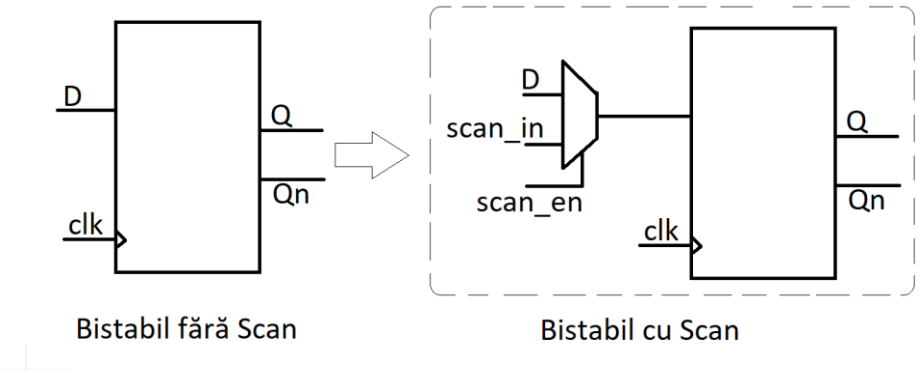
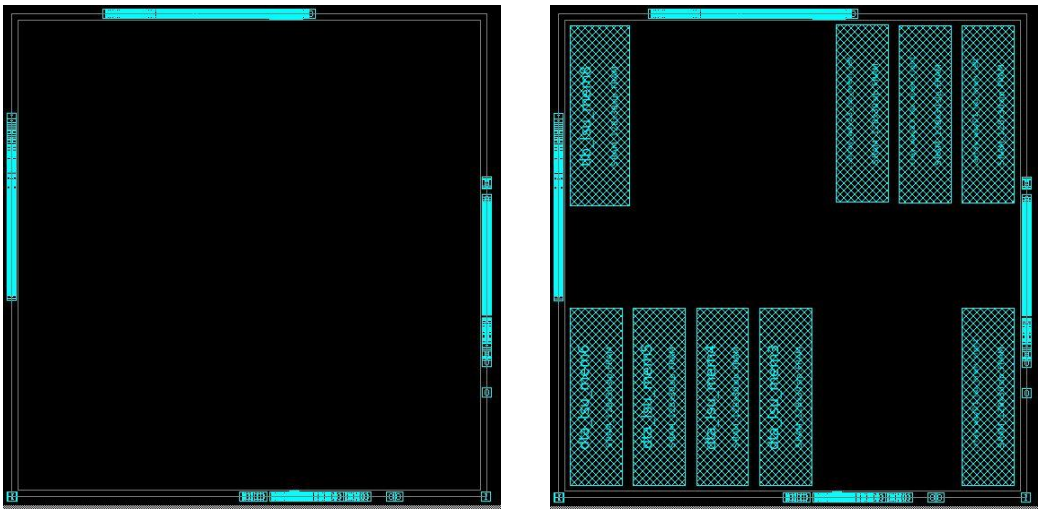


Fig. 2.4. Bistabil de tip D fără intrarea de Scan și bistabil de tip D cu intrare de Scan multiplexată cu intrarea de date

2.2 Proiectarea suprafeței circuitului integrat, stabilirea floorplanului și a planului de alimentare

Floorplan: în această etapă se stabilește forma și aria circuitului în care pot fi plasate toate celulele circuitului și se plasează pinii de intrare-ieșire din circuit pe marginea ariei ca în figura 2.5 (a), conform cu cerințele de proiectare a circuitului integrat. Pinii de intrare-ieșire sunt plasați în general în funcție de conexiunea cu alte blocuri externe. Tot în această etapă sunt plasate și fixate pe poziții celulele macro, după cum se poate observa în figura 2.5 (b), celule care reprezintă fie memorii, fie blocuri analogice și care sunt instanțiate în circuit, iar pozițiile lor pe suprafața

circuitului depinde tot de cerințele de proiect.



(a) **(b)**

Fig. 2.5. Plasarea pinilor și a memoriilor pe cele 4 laturi ale circuitului: (a) Plasare pinilor I/O, (b) Plasarea memoriilor

După plasarea pinilor de intrare-ieșire și a celulelor macro, sunt rulate comenzi care să verifice legitimitatea plasării, iar în figurile 2.6 respectiv 2.7 se pot vedea rapoartele generate de programul IC Compiler cu care este parcursă partea de Backend. Dacă sunt depistate erori de plasare în această etapă, în raport vor fi semnalizate cu 1.

```
****************************************************
Check_legality: Report for Fixed Placement Cells
Information: Use the -verbose option to get details about the legality violations. (PSYN-054)
****************************************************
(fixed placement) Cells Not on Row            : 0
(fixed placement) Cell Overlaps               : 0
(fixed placement) Cells overlapping blockages : 0
(fixed placement) Orientation Violations      : 0
(fixed placement) Site Violations             : 0
(fixed placement) Power Strap Violations      : 0
****************************************************
```

Fig. 2.6. Raport generat de IC Compiler pentru verificarea legitimității plasării celulelor fixate pe poziții

Raportul generat după plasarea și fixarea pinilor intrare-ieșire și a blocurilor macro, confirmă plasarea lor corectă în aria circuitului, toate fiind aliniate corespunzător pe rânduri dedicate, fără a avea erori de suprapunere sau erori de orientare. Se poate observa în raportul din figura 2.7, ca nu au fost identificate probleme care sa reflecte nerespectarea constrângerilor impuse de tehnologia utilizată și astfel este asigurat un floorplan viabil pe care se pot parcurg următoarele etape din procesul de implementare. Același raport este generat și pentru componentele care nu sunt fixate pe poziții, iar rezultatele sunt cele din figura 2.8 din care se poate trage concluzia că nu au fost identificate probleme la crearea floorplanului.

```
****************************************************
Check_legality: Report for Non-fixed Placement Cells
Information: Use the -verbose option to get details about the legality violations. (PSYN-054)
****************************************************
Number of Cells Not on Row           : 0
Number of Cell Overlaps              : 0
Number of Cells overlapping blockages : 0
Number of Orientation Violations     : 0
Number of Site Violations            : 0
Number of Power Strap Violations     : 0
****************************************
```

Fig. 2.7. Raport generat de IC Compiler pentru verificarea legitimității plasării celulelor ne-fixate pe poziții

Înainte de plasarea și fixarea celulelor macro în locațiile stabilite pe suprafața circuitul integrat, dacă nu sunt indicate pozițiile fixe, blocurile sunt plasate în afara ariei asa cum se observă în figura 2.8 și sunt raportate erori de plasare ilegală.

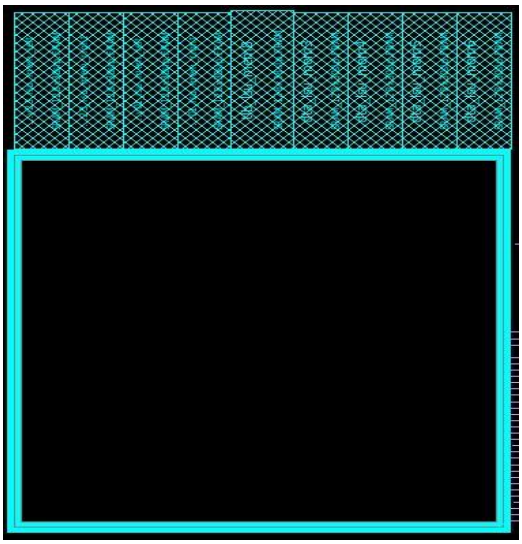


Fig. 2.8. Plasarea celulelor macro în afara ariei circuitului

Poziția, orientarea cât și "HALO" (distanța minimă necesară în jurul blocurilor macro), au o influență importantă asupra lungimii interconexiunilor, cu consecințe directe asupra întarzierilor. O planificare inițială nu tocmai optimă a floorplanului poate avea un impact major asupra proiectării fizice a circuitului cu consecințe asupra întârzierilor pe căile de date și această situație poate necesita eforturi semnificative de reiterare a procesului de implementare a CI-ului, pentru corectarea problemelor de sincronizare [85, 89].

O evaluare a corectitudinii floorplan-ului se face verificând utilizarea circuitului care indică procentul de utilizare a zonei de siliciu de către celulele standard și celulele macro, informații care apar în raportul generat de IC Compiler după rularea comenzii de generare a utilizării ariei ca în figura 2.9.

```
**************************************
Std cell utilization: 47.96%  (460986/(1802640-841461))
(Non-fixed + Fixed)
Std cell utilization: 47.96%  (460986/(1802640-841461))
(Non-fixed only)
Chip area:            1802640  sites, bbox (10.00 10.00 684.88 688.83) um
Std cell area:        460986   sites, (non-fixed:460986 fixed:0)
                      31831    cells, (non-fixed:31831  fixed:0)
Macro cell area:      717724   sites
                      9        cells
Placement blockages:  841461   sites, (excluding fixed std cells)
                      841461   sites, (include fixed std cells & chimney area)
                      0        sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       103
Avg. std cell width:  2.29 um
Site array:           unit     (width: 0.152 um, height: 1.672 um, rows: 406)
Physical DB scale:    1000 db_unit = 1 um


**************************************
```

Fig. 2.9. Raport de utilizare a ariei generat după etapa de Floorplan

Tot după această etapă de floorplan se analizează dacă blocurile macro sunt corect conectate atât la planul de alimentare cât și la semnalele de control și de asemenea sunt evidențiate regiunile unde densitatea conexiunilor este prea mare, ceea ce poate conduce la dificultăți de rutare.

Un floorplan bine organizat și optimizat reduce timpul de implementare a circuitului integrat, îmbunătățește performanțele circuitului și de asemenea minimizează costurile de fabricație.

După floorplan urmează ***powerplan*** unde pot fi folosite, pentru calculul consumului de putere a circuitului, informațiile obținute după procesul de sinteză, unde sunt generate rapoarte referitoare la puterea statică și de asemenea rapoarte referitoare la puterea dinamică ce poate fi calculată utilizând fișierele în care sunt înregistrate schimbările de stare ale semnalelor dintr-un circuit pe parcursul unei simulări (VCD). Un plan de alimentare eficient asigură alimentarea componentelor din întregul circuit, fără a avea probleme de electromigrație (EM) sau pierderi de tensiune (IR Drop). Sunt create inele de alimentare (Power Rings) atât separat pentru celulele macro din circuit cât și pentru întreaga arie activă a circuitului astfel încât să se asigure o alimentare optimă pentru toate componentele circuitului. Lățimea și grosimea traseelor metalice utilizate în planul de alimentare sunt calculate în funcție de consumul de putere și de constrângerile de layout.

2.3 Plasarea celulelor logice standard, optimizările necesare plasării optime

După crearea planului de alimentare urmează plasarea celulelor logice în aria rămasă după plasarea celulelor macro, etapă în care sunt șterse toate modelele de conexiuni fizice (WLM). În timpul plasării se folosesc valorile de RC extrase din rutarea virtuală (VR) pentru a calcula informațiile de timp [90]. Rutarea virtuală reprezintă cea mai scurtă distanță Manhattan dintre doi

pini, care oferă o estimare inițială a lungimii necesare între două componente fără a lua în considerare obstacolele din traseu, cum ar fi celulele macro sau alte conexiuni existente [91]. Plasarea logicii se realizează în 4 etape de optimizare:

1. Optimizare pre-plasare: etapă în care este optimizat netlistul înainte de plasare. În aceasta etapă celulele logice pot fi de asemenea înlocuite cu celule de dimensiuni mai reduse decât dimensiunile inițiale (downsize cells) [92].
2. Optimizare în-timpul-plasării: se re-optimizează logica pe baza rutării virtuale a conexiunilor fizice [93, 94]. În această etapă se face redimensionarea celulelor logice, mutarea celulelor logice, duplicarea de celule logice, inserarea de buffere și recuperarea de arie. Obiectivul principal în aceasta etapă este aria și fixarea *întârzierilor excesive* (Setup Violations).
3. Optimizare post-plasare înainte de CTS: plasarea care se face în această etapă este realizată înainte de crearea arborelui de ceas, luând în considerare rutarea globală a conexiunilor fizice din circuit.
4. Optimizare post-plasare după CTS : etapă în care sunt optimizate căile de timp ținând cont de informațiile de pe arborele de ceas. În această etapă instrumentul de plasare păstrează skew-ul ceasului obținut după crearea arborelui de ceas. Această etapă este necesară pentru a asigura compensarea întârzierilor de pe arborele de ceas, evitând astfel problemele de sincronizare.

În figura 2.10 (a) este evidențiată, prin utilizarea culorii mov, suprafața în care s-au plasat celulele logice standard în aria ramasă disponibilă după plasarea și fixarea blocurilor de memorie. Celulele logice sunt plasate inclusiv în zona dintre memorii în funcție de constrângerile de rutare. În figura 2.10 (b), 2.10 (c) și 2.10 (d) sunt prezentate vederi mărite ale unor secțiuni din circuit. Figura 2.10 (b) prezintă plasarea a 4 memorii dta_lsu_mem3, dta_lsu_mem4, dta_lsu_mem5 și dta_lsu_mem6 și a celulelor logice în jurul lor. Figura 2.10 (c) oferă o perspectivă asupra structurilor porților logice care conțin forma geometrică și poziția pinilor pe suprafața celulelor. Iar în ultima figură 2.10 (d) se poate observa în detaliu forma geometrică a celulei logice AO21X2_RVT cu numele de instanță U28236, care este o poartă logică complexă care încorporează o poartă logică ȘI împreună cu o poartă logică SAU.

Structura internă a componentei logice AO21X2_RVT [95] poate fi observată în figura 2.11 (a), iar tabelul de adevăr corespunzător funcționalității componentei este ilustrat în figura 2.11 (b). De asemenea parametrii electrici caracteristici componentei cu numele de instanță U28236, extrași din librăria cu care s-a realizat implementarea circuitului integrat, pot fi observați în figura 2.11 (c).

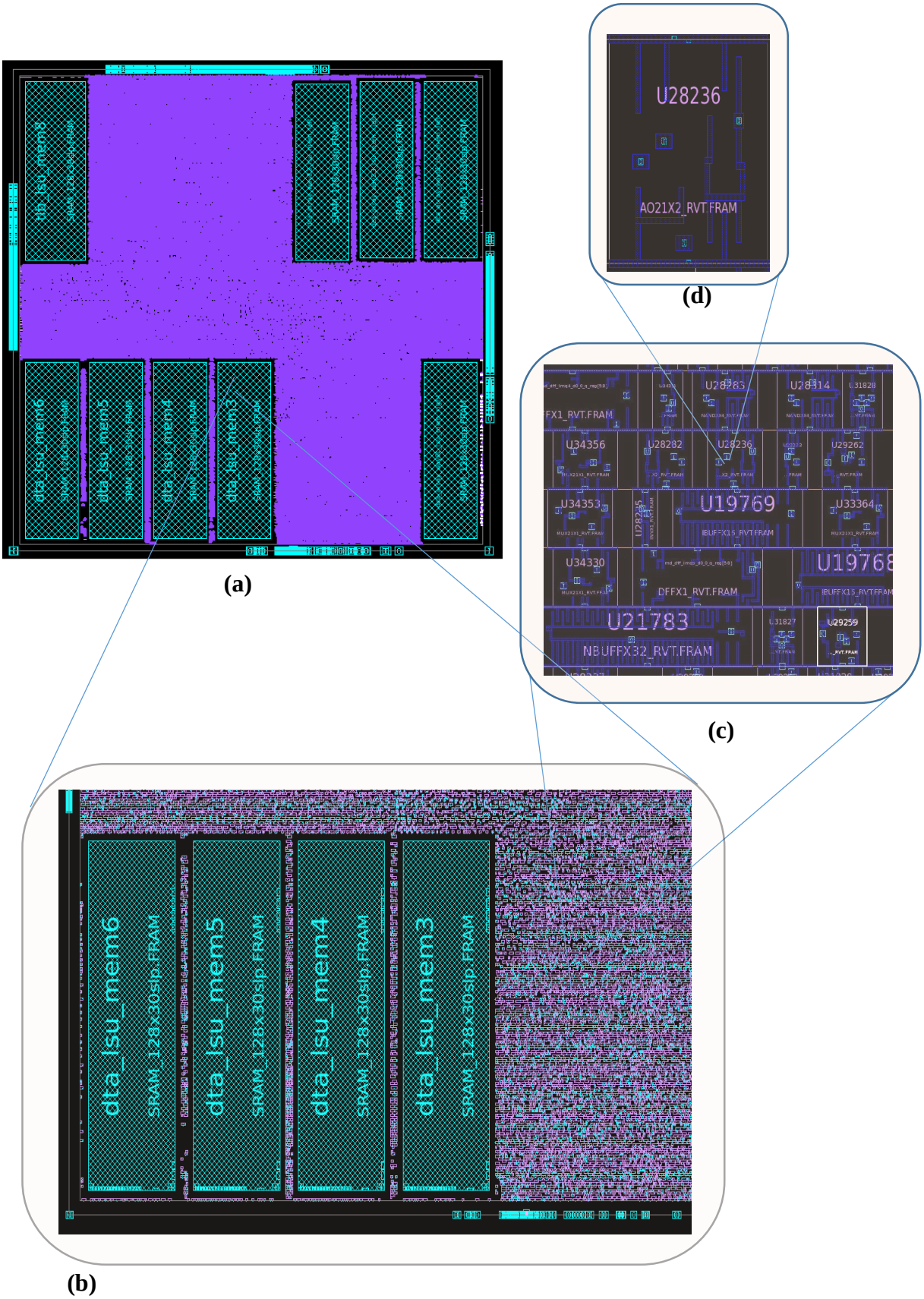
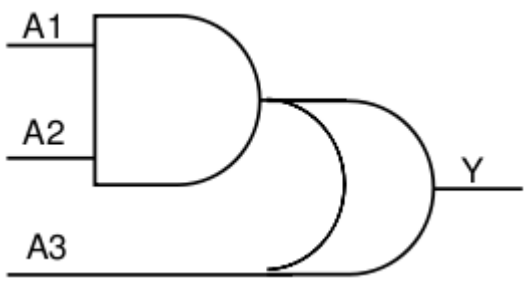


Fig. 2.10. (a). Suprafața circuitului integrat pe care s-au plasat blocurile de memorie și celulele logice standard; (b). Zonă mărită din circuit cu memorii și cu celule logice (c) Zonă mărită cu celule logice; (d) Geometria unei porți logice



(a)

A1	A2	A3	Y
1	1	X	1
X	X	1	1
0	X	0	0
X	0	0	0

(b)

Cell Name	Operating Conditions: VDD=1.05 V DC, Temp=25 Deg.C, Operating Frequency: Freq=500 MHz, Capacitive Standard Load: Csl=4 fF				Area
	Cload	Prop Delay (Avg)	Power		
			Leakage (VDD=1.05 V DC, Temp=25 Dec.C)	Dynamic	
		ps	nW	nW/MHz	(um^2)
AO21X1	1 x Csl	63	277	3	2.54144
AO21X2	2 x Csl	64	524	5	2.795584

(c)

Fig. 2.11. (a) Structura internă a porții logice AO21X2_RVT (b) Tabel de adevăr al porții AO21X2_RVT (c) Tabel cu parametrii electrici a porților AO21X1 și AO21X2_RVT [95]

După plasarea celulelor logice și toate operațiile de optimizare, se verifică legitimitatea plasării și se generează raportul de utilizare a ariei, care față de etapa anterioară de floorplan trebuie sa fie mai mare datorită celulelor logice adăugate la optimizare. În figura 2.12 se pot vedea rezultatele după plasare și optimizare, într-un raport generat de instrumentul de plasare încorporat în IC Compiler.

```
Std cell utilization: 48.53%  (466505/(1802640-841461))
(Non-fixed + Fixed)
Std cell utilization: 48.53%  (466505/(1802640-841461))
(Non-fixed only)
Chip area:            1802640  sites, bbox (10.00 10.00 684.88 688.83) um
Std cell area:        466505   sites, (non-fixed:466505 fixed:0)
                      31883    cells, (non-fixed:31883  fixed:0)
Macro cell area:      717724   sites
                      9        cells
Placement blockages:  841461   sites, (excluding fixed std cells)
                      841461   sites, (include fixed std cells & chimney area)
                      0        sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       104
Avg. std cell width:  2.30 um
Site array:           unit     (width: 0.152 um, height: 1.672 um, rows: 406)
Physical DB scale:    1000 db_unit = 1 um
```

Fig. 2.12. Raport de utilizare a ariei generat după etapa de Plasare

De asemenea pentru un control mai eficient asupra plasării celulelor logice, care să prevină eventualele probleme care ar putea apărea la etapa de rutare, sunt generate rapoarte de utilizare pe toate regiunile din circuit și astfel se pot identifica timpuriu eventualele blocaje de rutare. În figura 2.13 este raportul generat de instrumentul de plasare și se poate observa că toată suprafața circuitului este împărțită în 8 intervale de utilizare a ariei și numărul de celule logice cu utilizarea respectivă.

```
*********************************
      Sub-Region Utilization
*********************************
Number of regions with placement utilization 0 - 0.125 is 68 (1.75%)
Number of regions with placement utilization 0.125 - 0.25 is 76 (1.96%)
Number of regions with placement utilization 0.25 - 0.375 is 523 (13.47%)
Number of regions with placement utilization 0.375 - 0.5 is 1313 (33.81%)
Number of regions with placement utilization 0.5 - 0.625 is 1642 (42.29%)
Number of regions with placement utilization 0.625 - 0.75 is 157 (4.04%)
Number of regions with placement utilization 0.75 - 0.875 is 29 (0.75%)
Number of regions with placement utilization 0.875 - 1 is 75 (1.93%)
```

Fig. 2.13. Raport de utilizare a ariei pe regiuni

2.4 Crearea arborelui de ceas

După plasarea celulelor logice urmează etapa de ***creare a arborelui de ceas*** necesar pentru sincronizarea datelor. Transferul datelor între componentele funcționale din circuit este sincronizat de semnalul de ceas. Crearea arborelui de ceas (CTS) este procesul prin care sunt inserate celule de întârziere (buffere ori inversoare) pe căile semnalului de ceas astfel încât semnalul de ceas să ajungă în același timp la toate bistabilele din circuit. Scopul principal în procesul de creare al arborelui de ceas este de a minimiza skew-ul (ideal ar fi un skew egal cu zero), care reprezintă diferența de timp în care semnalul de ceas ajunge la bistabile diferite. Un arbore de ceas balansat corect contribuie la atingerea constrângerilor de timp din circuit și ajută la funcționarea circuitului la frecvența dorită. Optimizarea arborelui de ceas este esențială pentru performanța finală a ASIC-ului, în special la viteze mari și implementat în tehnologii avansate [96].

Un semnal de ceas distribuit într-un circuit, înainte de crearea arborelui de ceas, arată ca în figura 2.14 [97] , unde semnalul de ceas se conectează la toate elementele secvențiale din circuit fără a ține cont de lungimea traseelor și a întârzierilor aferente. Semnalul de ceas este distribuit direct la toate celulele secvențiale fără a ține cont de plasarea celulelor secvențiale pe suprafața cipului, de optimizările de rutare a conexiunilor și fără a fi inserate buffere. Practic semnalul de ceas ajunge în același timp la toate celulele secvențiale din circuit ignorând distanța fizică dintre sursa semnalului de ceas și pozitia celulelor la care este conectat semnalul de ceas. În realitate însă lungimea interconexiunilor, metalul cu care se face conexiunea, precum și încărcarea capacitivă

afectează momentul în care ajunge semnalul de clock la bistabile.

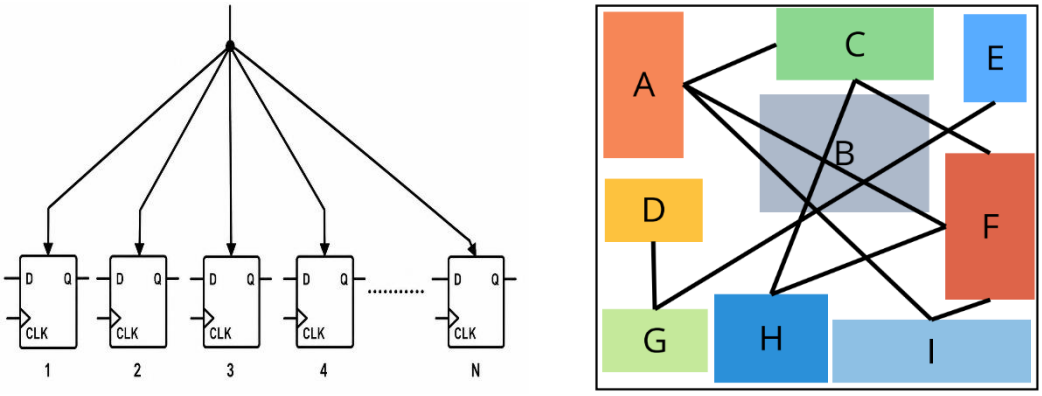


Fig. 2.14. Semnal de ceas distribuit la celulele secvențiale înainte de crearea arborelui de ceas [97]

În schema din figura 2.15 este prezentat un exemplu de arbore de ceas care pornește din sursa semnalului de ceas și este distribuit la 6 bistabile cu diferite întârzieri. Calculul skew-ului în cazul acestui arbore de ceas se face calculând diferența dintre cea mai lungă cale de timp care este de 20ns și cea mai scurtă cale de timp care este de 5ns.

$$Skew = 20ns - 5ns = 15ns$$

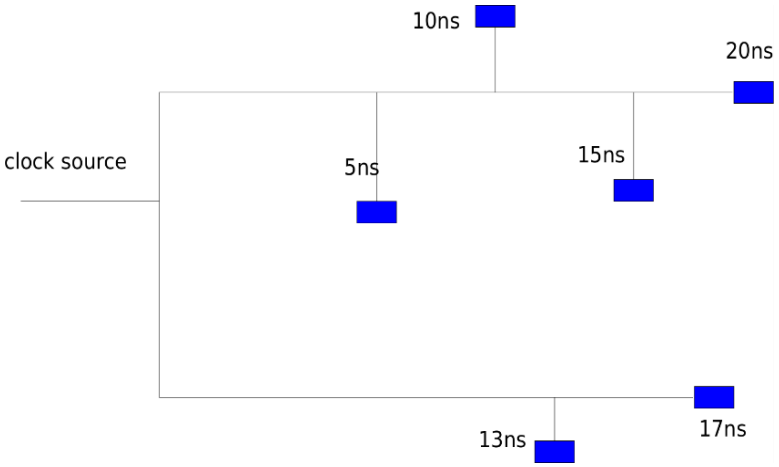


Fig. 2.15. Arbore de ceas distribuit într-un circuit

Un alt aspect important luat în considerare la crearea arborelui de ceas este consumul de putere, cunoscut fiind faptul că semnalul de ceas este cel mai mare consumator de putere din întregul circuit și poate contribui cu 20% până la 45% [98, 99] din consumul total al circuitului digital. Consumul de energie al arborelui de ceas depinde de frecvența semnalului de ceas precum și de numărul respectiv lungimea conexiunilor de pe arborele de ceas. Din perspectiva implementării fizice, arborele de ceas influențează semnificativ suprafața ocupată pe circuit, din

cauza numărului considerabil de buffere respectiv inversoare inserate pe traseele semnalului de ceas. Etapa dinaintea implementării arborelui de ceas, numită și etapă pre-rutare este esențială pentru analiza de timp a circuitului (STA) și necesită o analiză [100] amănunțită a informațiilor de timp din circuit pentru a obține un circuit care să funcționeze corect și la frecvența la care a fost proiectat.

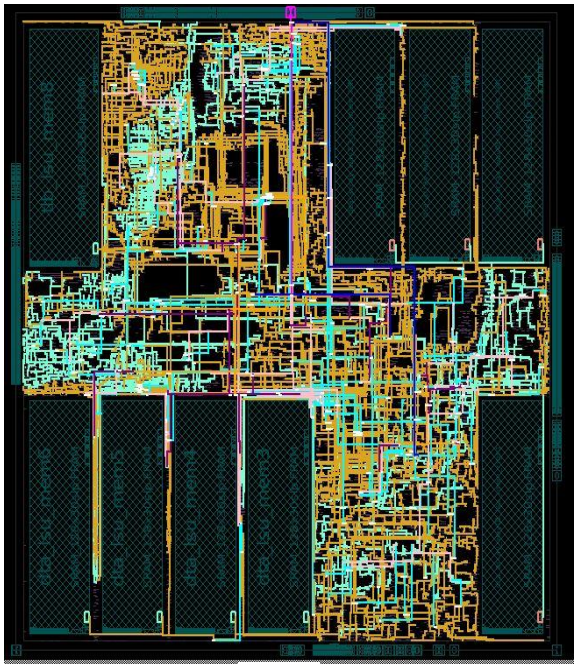
În figura 2.16 (a) se poate observa raportul generat după implementarea arborelui de ceas, a cărui sursa este l2clk, are o perioadă de 2ns, cuprinde 9 nivele și are un număr total de 475 celule de ceas. Skew-ul raportat în funcție de cea mai lungă cale din circuit (0.566ns) și cea mai scurtă cale din circuit (0.466ns) este de 0.100 ns. În figura 2.16 (b) se pot observa toate conexiunile fizice folosite la rutarea semnalului de ceas.

```
============ Global Skew Report =================

Clock Tree Name                : "l2clk"
Scenario                       : "func"
Clock Period                   : 2.00000
Clock Tree root pin            : "l2clk"
Number of Levels               : 9
Number of Sinks                : 7729
Number of CT Buffers           : 295
Number of CTS added gates      : 0
Number of Preexisting Gates    : 180
Number of Preexisting Buf/Inv  : 0
Total Number of Clock Cells    : 475
Total Area of CT Buffers       : 1602.37866
Total Area of CT cells         : 2065.67969
Max Global Skew                : 0.09992
Number of MaxTran Violators    : 0
Number of MaxCap Violators     : 27
Number of MaxFanout Violators  : 0


Operating Condition              worst
Clock global Skew                0.100
Longest path delay               0.566
Shortest path delay              0.466
```

(a)



(b)

Fig. 2.16. (a) Raport generat după construirea arborelui de ceas; (b) Trasarea conexiunilor fizice corespunzătoare arborelui de ceas

2.5 Rutarea conexiunilor din circuit

După crearea arborelui de ceas urmează procesul de trasare a conexiunilor fizice care determină precizia căilor pentru interconectare. Acest proces de rutare include realizarea conexiunilor fizice între pinii celulelor logice, pinii blocurilor macro și pinii de intrare/ieșire ai circuitului. În etapa de rutare se utilizează straturi metalice și via-uri (contacte cu care se trece de pe un metal pe altul) pentru a crea conexiunile electrice necesare în layout, astfel încât să se finalizeze toate conexiunile logice definite în netlist.

Procedura de rutare a conexiunilor fizice se realizează în două etape: prima etapă este rutarea globală, cea în care se definesc traseele generale ale conexiunilor, dar fără a le plasa pe straturi metalice și cea de-a doua etapă, rutarea detaliată, în care sunt plasate conexiunile logice fizic pe straturi respectând regulile tehnologice. Se verifică căile de timp și se face analiza de putere și apoi se efectuează optimizări suplimentare pentru închiderea constrîngerilor de timp și reducerea consumului de putere.

Înainte de generarea fișierului GDS, care împreună cu librăria tehnologică sunt trimise la fabrică pentru realizare, se fac ultimele verificări de constrângeri (DRC) și de menținere a funcționalității circuitului (LVS) cu instrumente de sign-off (dedicate) care să garanteze veridicitatea rezultatelor [101].

Pentru întregul proces de lucru s-au folosit pentru automatizare scripturi descrise în TCL conform ANEXEI 1.

2.6 Concluzii la capitolul 2

În acest capitol, este descris un flux de lucru 2D particularizat, dezvoltat pe baza unei experiențe de 10 ani în proiectarea de circuite integrate. Acest flux de lucru a fost utilizat cu succes în implementarea mai multor circuite integrate fabricate, demonstrând eficiența și aplicabilitatea sa practică. Concluziile principale ale acestui studiu sunt următoarele:

1. Procesul de lucru 2D descris în acest capitol pentru implementarea circuitelor integrate reprezintă rezultatul aprofundării practice în industrie, încapsulând cele mai bune practici și optimizări care sporesc eficiența circuitului. Prin integrarea sistematică a sintezei, a circuitului pentru testabilitate (DFT), a verificării formale (FV) și a considerațiilor de plasare timpurie, această metodologie oferă o abordare structurată, repetabilă și scalabilă a implementării circuitului digital.
2. Metodologia descrisă realizează o legătură între principiile teoretice și cele mai bune practici din industrie prin integrarea constrângerilor din lumea reală, limitările instrumentelor și considerațiile de fabricabilitate. Acest lucru asigură că fluxul nu este doar riguros din punct de vedere academic, ci poate fi implementat în mediile comerciale de proiectare a semiconductorilor, sporind impactul și aplicabilitatea acestuia.
3. Această abordare descrisă în acest capitol încorporează în mod explicit DFT și FV ca și etape primare ale fluxului de front-end, asigurând că testabilitatea și corectitudinea funcțională (care trebuie menținută de-a lungul întregului proces de proiectare) sunt abordate la începutul procesului de proiectare. Această abordare proactivă reduce

semnificativ riscul eșecurilor de fabricație și eforturilor de depanare post-siliciu, ceea ce duce la un randament mai mare și costuri de producție mai mici.

4. Procedura de lucru utilizată facilitează explorarea spațiului de proiectare în stadiu incipient, permițând proiectanților să facă compromisuri informate între putere, performanță și arie. Prin structurarea eficientă a strategiilor de sinteză și de plasare respectiv rutare a conexiunilor, aceasta permite optimizări care asigură eficiența proiectării, respectând în același timp constrângerile de timp, putere și implementare fizică.
5. În timp ce metodologia este dezvoltată având în vedere instrumentele standard din industrie, principiile sale sunt structurate pentru a fi adaptabile în diferite lanțuri de instrumente de automatizare a circuitului electronic (EDA). Această flexibilitate îi crește aplicabilitatea în diverse medii de proiectare, făcându-l un cadru versatil pentru implementarea circuitelor integrate.
6. Utilizarea scripturilor TCL facilitează controlul structurat al iterației, permițând ajustări și optimizări rapide în timpul sintezei, DFT și plasării respectiv rutării. Această abordare accelerează convergența circuitului și îmbunătățește productivitatea permițând reglarea eficientă a parametrilor.
7. Flexibilitatea în adaptarea procesului de lucru la cerințele specifice ale proiectului este un avantaj semnificativ. Abilitatea de a ajusta parametrii și metodele utilizate permite o abordare mai eficientă a provocărilor specifice fiecărui circuit.
8. Capacitatea de a analiza și interpreta corect rezultatele fiecărei etape este un factor critic în evitarea unor probleme costisitoare în stadiile avansate ale proiectului. O abordare metodică în interpretarea rapoartelor de verificare permite luarea unor decizii informate în timp util.
9. Cunoașterea detaliată a flow-ului de proiectare este esențială pentru prevenirea și remedierea rapidă a erorilor. O înțelegere profundă a fiecărei etape permite identificarea timpurie a potențialelor probleme, contribuind astfel la un proces de proiectare mai robust.
10. Unul dintre punctele forte cheie ale acestei metodologii este adaptabilitatea sa la diferite niveluri de complexitate a circuitului, variind de la circuite integrate mici specifice aplicației (ASIC) până la proiecte de sistem pe circuit integrat (SoC) la scară largă. Modularitatea fluxului permite scalabilitate, asigurând că principiile cheie de proiectare rămân valabile în diferite noduri tehnologice și constrângeri de proiectare.
11. Natura structurată și modulară a procesului de lucru descris pune bazele progreselor viitoare în proiectarea circuitelor integrate. Domeniile potențiale pentru cercetări ulterioare includ integrarea tehnicilor de inteligență artificială (AI) și de învățare automată (ML)

pentru a îmbunătăți sinteza și optimizările locului și rutei, îmbunătățiri ale metodologiilor de procesare paralelă și perfecționarea în continuare a strategiilor de verificare pentru a crește automatizarea și acuratețea în validarea circuitului în stadiu incipient.

În concluzie, metodologia prezentată în această lucrare demonstrează că un proces de lucru 2D bine definit, susținut de experiență și optimizări continue, poate conduce la proiectarea și fabricarea cu succes a circuitelor integrate. Validarea acestui flux de lucru prin multiple circuite integrate fabricate confirmă aplicabilitatea și eficiența sa în industrie, reprezentând o contribuție relevantă la domeniul proiectării de circuite integrate.

3. PROIECTARE 3D UTILIZÂND INSTRUMENTE 2D. MODURI DE PARTIȚIONARE A CIRCUITULUI

În acest capitol este abordată tema partiționării circuitelor integrate, o etapă foarte importantă în procesul de trecere de la proiectarea circuitelor integrate în tehnologie 2D la integrarea 3D. În această etapă de partiționare, practic se decide exact ce componente (atât celule logice cât și memorii) sunt plasate pe fiecare strat, astfel încât numărul de conexiuni verticale (MIV sau TSV în funcție de tehnologia utilizată) să fie cât mai mic [102, 103]. Cu cât partiționarea este mai optimă, cu atât scade numărul de conexiuni verticale și astfel este redus consumul de energie, întârzierile pe traseele de semnal sunt micșorate, iar rutarea este mai puțin complexă.

3.1 Rolul partiționării în trecerea de la integrarea 2D la 3D

Una dintre cele mai promițătoare tehnologii revoluționare, care poate obține o densitate de integrare mai mare, o miniaturizare a formelor circuitelor integrate, cu performanțe mai bune și consum de putere mai redus, este integrarea tridimensionala 3D. Atât mediul academic cât și industria acordă o atenție deosebită cercetării în aria 3D, explorând tehnologii de fabricație, metodologii de proiectare și testare, algoritmi de optimizare, precum și fiabilitatea interconectării între straturi.

De cel puțin 40 de ani sunt cunoscute softuri folosite pentru automatizarea proiectării circuitelor electronice integrate [104]. Acest segment al proiectării asistate de calculator a circuitelor integrate, cuprinde un spectru larg al activităților inginerești, pornind de la dezvoltarea teoretică a ideii și trecând prin diferite stadii (automate și manuale), de la modelare și simulare, până la înlocuirea cu componente tehnologice, partiționarea logică și fizică a acestora, plasarea componentelor și rutarea conexiunilor, verificarea regulilor și producerea măștilor. Primele etape din acest proces de proiectare fac parte din domeniul de modelare, înainte de înlocuirea cu componente tehnologice (tranzistoare, celule logice, blocuri de memorare) și poartă denumirea de proiectare logică. După înlocuirea cu componente tehnologice, procesul poartă denumirea de proiectare fizică.

Scăderea rapidă a dimensiunilor tranzistoarelor folosite în procesul de fabricație pe placheta de siliciu, permite o creștere a numărului acestora. Circuitele integrate moderne depășesc frecvent ca dimensiune sute de milioane de tranzistoare și au nevoie de programe de proiectare complexe, care să fie capabile să proceseze circuite integrate la asemenea dimensiuni.

Instrumentele (EDA) utilizate la automatizarea procesului de implementare a circuitelor integrate au evoluat de-a lungul timpului de la asistență semi-automată pentru implementarea fizică, până la instrumentele complexe actuale capabile să facă automat toate etapele de la verificare cod, sinteză, plasare și rutare, creare arbore de ceas și în final verificări automatizate pentru validarea circuitului ce urmează a fi fabricat. Totodată a fost necesar, pe langă automatizarea procesului, ca instrumentele EDA să se adapteze în permanență la noile tehnologii astfel încât să suporte provocările apărute cu apariția fiecarei noi tehnologii [104, 105]. Pe același principiul, la ora actuală este necesară includerea în instrumentele EDA a unor noi algoritmi care să efectueze partiționarea circuitului pe straturi diferite, în diferite etape de-a lungul procesului de implementare a CI-urilor [106, 107], astfel încât integrarea 3D să se efectueze optim.

Momentul integrării algoritmului de partiționare în procesul de implementare a circuitelor integrate este esențial pentru obținerea unor rezultate optime în ceea ce privește performanța, consumul energetic și scalabilitatea tridimensională [108]. În mod tradițional, partiționerul este introdus imediat după sinteza logică [109], într-un moment în care netlistul este deja definit, dar încă nu a fost supus etapelor de plasare și rutare fizică. Această poziționare strategică permite împărțirea eficientă a componentelor circuitului în grupuri funcționale sau clustere, care pot fi ulterior alocate fiecărui strat într-o implementare 3D, sau fiecărui bloc local în contextul unui circuit 2D. Integrarea timpurie a unei etape de partiționare permite optimizarea conexiunilor interne, reducerea congestiei și minimizarea numărului de TSV-uri sau MIV-uri în cazul arhitecturilor 3D, contribuind astfel la îmbunătățirea disipării termice și la scăderea latenței globale [110].

În figura 3.1 este prezentat un proces normal de proiectare fizică a unui circuit integrat 2D. Procesul pornește de la un netlist, alcătuit din componente dependente de tehnologie (componentele fizice), și un set de constrângeri provenite din partea logică a procesului de proiectare. Netlistul împreună cu constrângerile pot fi împărțite într-un mic set de blocuri, de ordinul 1 până la 100, utilizând un program numit “partiționer”, care poate produce un netlist modificat și un fișier de constrângeri modificat. Acestea, la rândul lor, pot constitui intrări pentru un floorplaner care poate aranja aceste blocuri ca un mozaic, ținând cont de constrângerile din netlist, pe o suprafață dreptunghiulară care poate sublinia amprenta fizică a circuitului integrat final și va produce un nou netlist modificat și un nou fișier de constrângeri modificat. Pentru tranziția de la 2D la 3D partiționerul decide ce componente sunt plasate pe fiecare strat în funcție de anumite criterii (astfel influențează plasarea inițială) și face optimizări pentru a obține un număr minim de conexiuni verticale (TSV-uri sau MIV-uri în funcție de tehnologie).

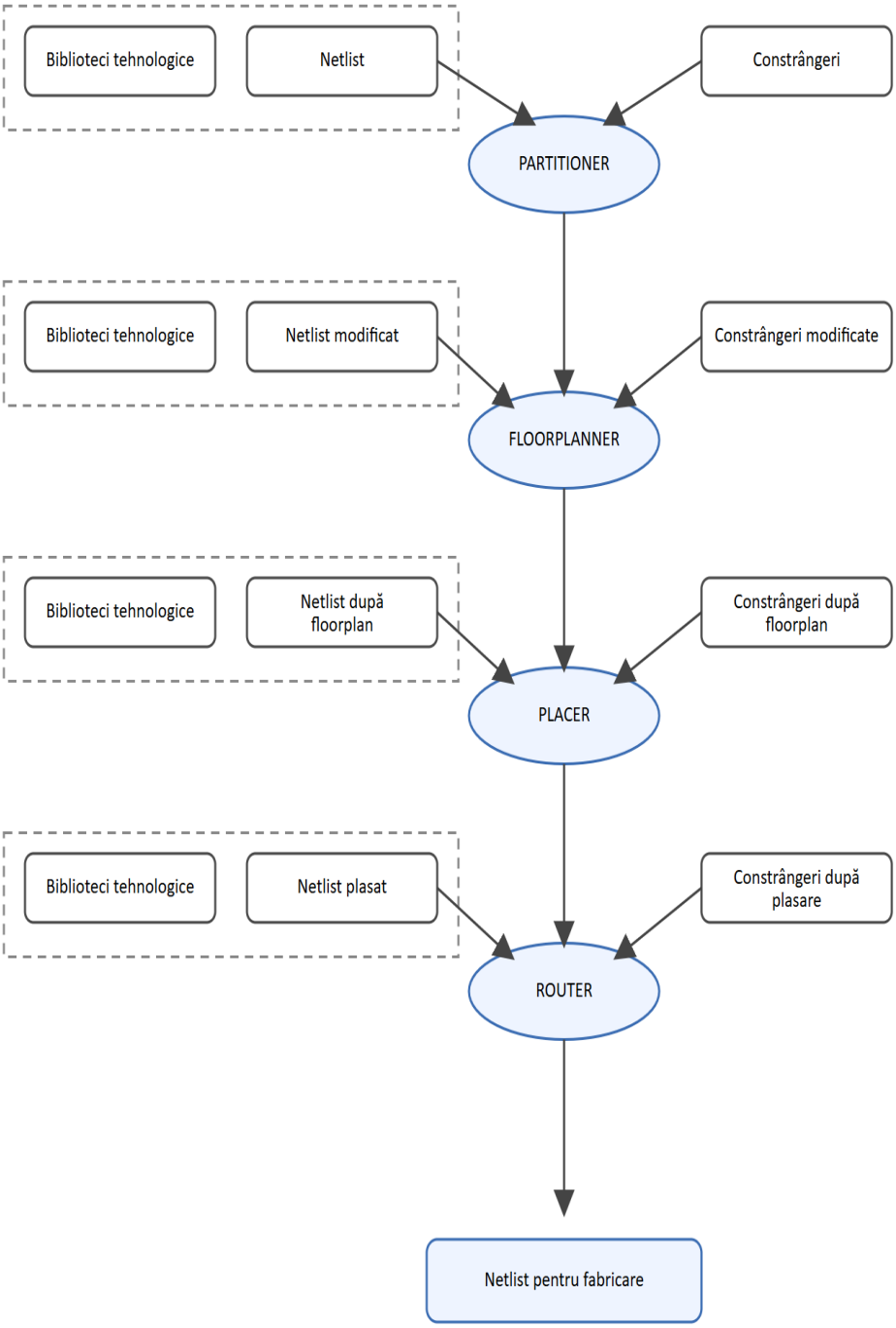


Fig. 3.1. Proces de proiectare 2D cu partiționare inclusă

Instrumentul de plasare poate atribui obiectelor din cadrul fiecărui bloc din noul netlist o locație, care să respecte constrângerile din fișierul de constrângeri. După această etapă, netlistul plasat și fișierul modificat de constrângeri pot fi fișiere de intrare pentru instrumentul de realizare a conexiunilor, care va realiza trasarea conexiunilor și va produce un nou netlist, care poate fi verificat și procesat mai departe, în vederea fabricării circuitului integrat. Pe tot parcursul procesului de proiectare a circuitului integrat, programele de proiectare folosesc, pe lângă netlist

și fișierul de constrângeri, un set de librării tehnologice care descriu componentele instanțiate în netlist și un set de regulile care definesc acțiunile permise asupra componentelor și asupra relațiilor legale dintre ele, dar și dintre componente și reprezentările lor tehnologice abstracte. Fiecare pas din procesul descris în figura 3.1 necesită intervenția inginerului proiectant.

Programele de proiectare de circuite integrate tradiționale țin cont de faptul că tranzistoarele vor fi plasate pe un singur strat. În ultimii ani, unele programe de proiectare au fost îmbunătățite cu o nouă caracteristică ce le permite să înțeleagă plasarea tranzistoarelor pe mai multe straturi. Prin utilizarea acestei caracteristici se urmăresc beneficiile integrării 3D și reducerea utilizării metodei TSV care este scumpă. Adăugarea acestei caracteristici extinde funcționalitatea softurilor de proiectare astfel încât să opereze cu tehnologia monolitic 3D. Așa cum a fost evidențiat în primul capitol, principala diferență dintre tehnologia Monolithic 3D și integrarea pe straturi bazată pe TSV constă în dimensiunea conexiunilor realizate între straturi. TSV-urile sunt foarte mari în comparație cu mărimea litografiei, iar scalarea TSV-urilor nu se raportează la litografie, ci mai degrabă la factorul ridicat de gravare și umplere și la abilitatea de a manevra plachete de siliciu extrem de subțiri. Astăzi, cel mai bun factor de gravare și umplere este de aproximativ 10:1, iar cea mai subțire plachetă de siliciu care poate fi manevrată în mod corespunzător este de aproximativ 20 microni, cu diametrul TSV de aproximativ 5 microni. Prin contrast, dimensiunea conexiunilor între straturi pentru procesul monolitic 3D scade o dată cu scalarea tehnologică (scalarea semiconductoarelor) și a ajuns deja cu 100 nm, iar reducerea conexiunilor va continua atâta timp cât și industria va continua să avanseze in această direcție.

Drept consecință a celor descrise mai sus, optimizarea proceselor softurilor de proiectare a circuitelor în cazul TSV se va face astfel încât numărul de TSV să fie cât mai mic. În schimb, pentru monolitic 3D conectivitatea este mai densă și optimizarea proceselor softurilor de proiectare se va face ținând cont de plasarea optimă a componentelor pe straturi diferite, în funcție de caracteristicile straturilor și de creșterea spațiului dintre componente, comparativ cu 2D.

3.2 Metode de proiectare a unui circuit integrat pentru adaptarea la cea de a 3 a dimensiune

Metode de proiectare a unui circuit integrat 3D folosind instrumente actuale de proiectare a circuitelor integrate 2D și care ar ajuta la optimizarea softurilor deja existente pe piață ar fi:

1. Utilizând un algoritm de plasare 2D încorporat în instrumentele de plasare și rutare utilizate pentru proiectarea circuitelor integrate, se realizează plasarea pe cel puțin 2 straturi. După finalizarea etapei de plasare, se realizează rutarea conexiunilor și se continuă cu etapele finale de proiectare a unui circuit integrat 3D.

2. Înainte de plasare se realizează partiționarea circuitului pe cel puțin 2 straturi după anumite criterii care țin cont de arie, timing, consum de energie, apoi se efectuează plasarea separat pe cele 2 straturi folosind un placer 2D încorporat în instrumentele de plasare și rutare clasice și se continuă cu etapele finale de proiectare a unui circuit integrat 3D.
3. Se partiționează circuitul în două grupuri egale ca arie, și se decide care componente ramân pe primul strat și care merg pe al doilea strat. Se realizează întâi plasarea primului grup de celule pe primul strat, utilizând un instrument de plasare 2D și apoi plasarea celui de-al doilea grup de celule pe al doilea strat. În final se efectuează rutarea și se continuă cu etapele finale de proiectare a circuitului integrat 3D.
4. Pentru circuitele integrate cu module care funcționează la frecvențe diferite, se poate partiționa circuitul în funcție de frecvență astfel încât un strat poate fi implementat cu celulele care funcționează la frecvența mai mare, iar celălalt strat sa conțină celulele care funcționează la frecvența mai redusă. În acest fel se optimizează consumul de energie și de asemenea integritatea semnalelor.
5. Se realizează partiționarea circuitului pe două straturi astfel: primul strat cuprinde celulele logice și cel puțin un decodor de memorie, iar al doilea strat conține celulele (bit celuri) de memorie. Se continuă apoi cu plasarea pe primul strat cu un instrument de plasare 2D urmată de plasarea pe al doilea strat. Plasarea pe al doilea strat se realizează în corelație cu plasarea de pe primul strat, astfel încât să fie asigurată suprapunerea verticală necesară care să permită aranjarea simetrică a conexiunilor celor două părți a blocurilor de memorie. O suprapunere exactă în această situație a memoriilor împărțite pe două straturi, asigură o uniformitate a întârzierilor necesară în cazul unei astfel de conexiuni.
6. Partiționarea circuitului se realizează în etapa de plasare, se împarte circuitul în celule logice care se plaseaza pe primul strat și memorii care se plaseaza pe stratul al doilea. Întâi se realizează plasarea blocurilor de memorie și se fixează pozițiile, iar apoi se realizează plasarea celulelor logice pe primul strat ținând cont de pozițiile memoriilor. Se continuă cu rutarea și ceilalți pași specifici proiectării de circuite integrate 3D.
7. Celulele standard combinaționale și celulele secvențiale se separă pe două straturi și astfel resursele sunt folosite într-un mod optim. Se plasează în prima etapă celulele combinaționale pe primul strat, iar apoi cele secvențiale pe al doilea strat în funcție de pozițiile celulelor combinaționale de pe primul strat. O suprapunere cât mai precisă a celulelor secvențiale asigură reducerea lungimii conexiunilor și întârzieri de semnal

mai mici asociate respectivelor conexiuni. Se continuă cu rutarea și verificările necesare în timpul și după implementarea circuitelor integrate.

8. Se partiționează circuitul în funcție de cele mai importante trasee de semnal, denumite căi de timp critice și care sunt identificate prin analiza datelor de timing. Tranzițiile de semnal de pe aceste tipuri de căi afectează semnificativ perfomanța circuitului și prin urmare trebuie tratate cu prioritate în procesul de partiționare pe straturi. Pentru a repartiza celulele circuitului pe straturi diferite, metoda TA3D [111] utililizează algoritmi de clustering, cum ar fi algoritmul Leiden, care grupează celulele bazându-se pe conexiunile lor și pe informațiile de timing. Acest algoritm care poate fi observat în figura 3.2 ajută la formarea unor grupuri de celule care sunt strâns interconectate între ele și care ar trebui să fie plasate împreună pe același strat pentru a reduce distanța de semnal și întârzierile. Pentru această metodă de partiționare în cazul tehnologiilor planare nu este necesară informația de plasare 2D și astfel procesul este simplificat și este redusă dependența de etapele anterioare ale procesului de proiectare a CI. În cazul tehnologiilor avansate (FinFET) informația de plasare este necesară.

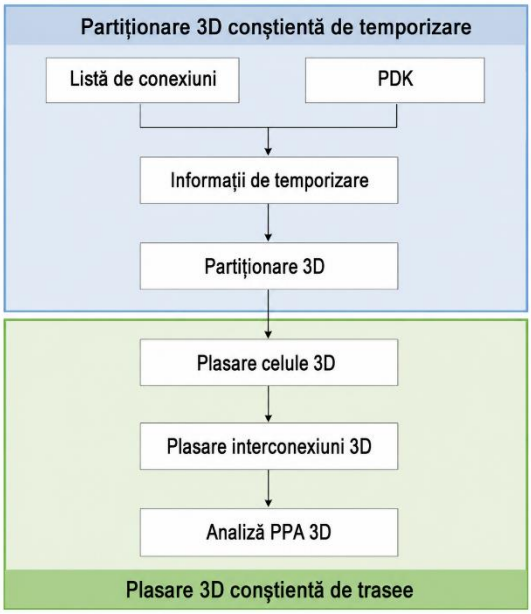


Fig. 3.2. Proces de proiectare a CI utilizând metoda TA3D [111]

Straturile dintr-un circuit integrat monolitic 3D pot diferi în funcție de dimensiunile caracteristice de litografiere, straturile impletementate în procese tehnologice mai mici fiind mai rapide dar mai scump de produs.

În circuitele integrate monolitic 3D, fiecare strat poate fi fabricat în propriul proces tehnologic cu regulile de proiectare aferente procesului, spre deosebire de circuitele integrate 2D unde toate tranzistoarele sunt pe un singur strat și sunt fabricate în același proces tehnologic. Partiționarea circuitului integrat 3D în diferite straturi s-ar putea face în funcție de tipurile de circuite care ar putea fi eficient de procesat împreună. O astfel de decizie poate fi bazată pe un anumit criteriu, de exemplu tipul de cerințe litografice. În circuitele integrate moderne, costul procesului litografic are un impact mare asupra costului final al produsului. Prin urmare ar putea fi eficient să se mărească numărul de straturi care utilizează procese de litografiere mai ieftină și să se micșoreze numărul de straturi care utilizează procese de litografiere mai scumpe, cum ar fi circuitele de viteză înaltă. Un strat diferit care poate fi dedicat intrărilor și ieșirilor din circuit, poate folosi de asemenea un proces de litografiere mai ieftin. Straturile pot diferi în funcție de numărul de metale folosit la rutare, un număr mare de metale oferind o creștere a conectivității și prin urmare o densitate mai mare a ariei utilizate, lucruri care conduc la creșterea costului procesului de fabricație. Straturile pot diferi și în funcție de puterea disipată de tranzistoare și de leakageul tranzistoarelor, de exemplu părțile de circuit cu consum redus de putere pot fi grupate pe un singur strat, împreună cu porțiuni de viteză redusă și/sau consum redus de leakage. Astfel de diferențe pot crea noi dimensiuni de optimizare a structurilor monolitice 3D. Straturile, pot diferi în funcție de regulile procesului de proiectare, pentru a forma dispozitive și circuite pe fiecare strat, mai exact, primul strat poate avea un set de reguli de proiectare mai avansat decât setul de reguli de proiectare pentru al doilea strat.

Din punct de vedere al tipurilor de memorii folosite în circuitele integrate, se poate realiza de asemenea o partiționare a blocurilor de memorii în funcție de tipurile de memorii folosite și anume: memorii volatile pe un strat și memorii nevolatile pe un alt strat, sau memoriile dinamice pe un strat, iar cele statice pe alt strat. În acest caz, afinitatea dintre caracteristicile stratului și tipul de memorie ce va fi implementată pe el, poate fi înțeleasă ca o compatibilitate ridicată între memorie și strat, sau ca un atribuit de cost și poate fi inclus în instrumentul de partiționare. Compatibilitatea ridicată poate fi un atribut ce forțează instrumentul de partiționare să grupeze doar obiectele cu atribute asemănătoare.

Partiționarea se poate implementa și între diverse funcții analogice, incluzând funcțiile de intrare și ieșire, și restul logicii digitale și a blocurilor de memorii din circuit, unde partea analogică poate fi plasată pe unul sau mai multe straturi din circuitul integrat monolitic 3D. Straturile analogice pot varia în procesul lor de litografiere, sau în funcție de materialul semiconductor folosit la implementare și anume Silicon, Germaniu sau material semconductor compozit cum ar fi Galium-Arsen sau Indiu-Fosfor. Asemănător cazurilor memoriilor descris mai sus, afinitatea dintre

tipurile obiectelor analogice și caracteristicle straturilor analogice poate fi tradus într-o compatibilitate ridicată sau o funcție cost și poate fi inclusă în partiționer. Din punct de vedere al consumului și temperaturii disipate, componentele unui circuit pot fi partiționate în straturi cu consum ridicat, unde componentele sunt plasate mai aproape de mecanismele de dispare termică și straturi cu consum redus, unde componentele pot fi plasate mai departe pentru a reduce disipare termică. O afinitate puternică între atribute poate forța partiționerul să grupeze doar obiectele cu atribute compatibile.

După partiționare și plasare pe straturi diferite, se face rutarea și restul verificărilor necesare pentru a valida funcționarea corectă a circuitului în funcție de constrângerile tehnologice.

3.3 Concluzii la capitolul 3

1. Este necesară dezvoltarea unui partiționer, care pe baza unor date de intrare să poată lua decizia să selecteze împărțirea pe straturi a componentelor circuitului, în funcție de imaginea globală a circuitului și de costul total al procesului de fabricare. Asta va ajuta la dezvoltarea software-urilor 3D și va reprezenta un salt uriaș în integrarea 3D.
2. Această etapă numită partiționare pe straturi a componentelor unui circuit integrat 3D este parte importantă din procesul de proiectare a unui circuit integrat pe verticală și necesită o atenție deosebită, astfel încât împărțirea pe straturi a componentelor circuitului să fie optimă de la început și să nu necesite reiterarea procesului de partiționare.
3. Partiționarea unui circuit integrat pentru implementarea 3D depinde de o serie de constrângeri precum: cerințele aplicației proiectată a fi implementată tridimensional, constrângerile tehnologice, prioritizarea performanțelor precum viteză de procesare, consum de putere, latență, cost. Metodele eficiente de partiționare a componentelor circuitului conduc la dezvoltarea unor circuite mai dense și mai performante în tehnologie 3D.
4. O partiționare eficientă realizată la începutul flow-ului de proiectare, într-o fază cât mai incipientă asigură o distribuire optimă a funcționalităților circuitului și astfel sunt reduse problemele legate de consumul de putere și disiparea termică, de interconexiunile dintre celule, obținerea vitezei optime a circuitului. Împărțirea circuitului pe straturi va duce la o organizare a circuitului care va avea impact asupra performanțelor circuitului și asupra resurselor utilizate în procesul de integrare tridimensională.

5. În dezvoltarea software-urilor care să realizeze implementarea circuitelor tridimensionale, o atenție deosebită se va acorda proiectării unui instrument de partiționare eficient, care pe baza unor date de intrare cum ar fi netlistul, fișierul de constrângeri și librăriile tehnologice să genereze soluții de împărțire a circuitului pe straturi.
6. Înainte de a implementa circuitul în scenariu 3D, circuitul se poate trece prin proces de lucru 2D și extrase rezultatele după fiecare etapă. Pe baza acestor informații obținute după fiecare etapă, partiționerul poate decide cum este mai eficient de împărțit circuitul și asta va conduce la un circuit mai performant.

4. IMPLEMENTAREA ȘI VERIFICAREA UNUI FLUX DE PROIECTARE 3D UTILIZÂND PROGRAME 2D EXISTENTE

În acest capitol este prezentată o metodologie originală de implementare a unui circuit integrat tridimensional (3D) prin suprapunerea memoriilor deasupra celulelor logice standard dintr-un circuit, care conține 50% memorii și 50% logică, folosind programele 2D existente pentru circuite bidimensionale (2D) de la Synopsys. Folosind această metodologie s-au implementat 3 configurații de circuite și s-au prezentat avantajele integrării 3D comparativ cu 2D [112–114]. Implementarea 3D s-a realizat modificând fișierele tehnologice astfel încât instrumentele 2D să permită suprapunerea memoriilor peste celulele logice standard fără a raporta erori de suprapunere [115]. Toate imaginile care conțin implementarea fizică a circuitului (layout-ul) sunt capturi de ecran din programul cu care s-a realizat procesul de proiectare a circuitului integrat și sunt reproduse, la o scară mărită, în Anexa 6, pentru a permite o analiză detaliată a acestora.

4.1 Valorificarea și optimizarea metodologiei 2D pentru implementarea circuitelor integrate 3D

În acest subcapitol sunt prezentate etapele și metodologia utilizată pentru implementarea unui circuit integrat tridimensional 3D, având ca punct de plecare un software de implementare a unui circuit integrat bidimensional. De asemenea, sunt discutate tehnologia și circuitul folosite în procesul de implementare a circuitului 3D, detaliind atât procesul de selecție a componentelor necesare, cât și modul în care acestea au fost generate. În plus, subcapitolul include o descriere detaliată a procesului de implementare 2D a aceluiași circuit, pe care se bazează cercetarea din cadrul acestei teze, susținută de o diagrama [116] care ilustrează clar etapele procesului de dezvoltare. Aceasta oferă o înțelegere mai clară a modului în care s-a realizat tranziția de la un circuit 2D la un circuit 3D, evidențiind pașii esențiali și interdependențele dintre aceștia în contextul cercetării.

Cea mai importantă caracteristică pentru valoarea unui circuit integrat (CI) este integrarea mai multor funcții într-un singur dispozitiv. Acest lucru este și va rămâne cel mai important factor ce determină respectarea Legii lui Moore [1], conform căreia numărul de tranzistoare dintr-un circuit integrat se dublează aproximativ la fiecare 2 ani ducând la creșterea performanței și reducerea costurilor pe tranzistor. Astfel, prin integrarea funcțiilor într-un singur CI putem obține beneficii însemnate pentru arie, consum de putere, frecvență de funcționare, costuri și fiabilitate.

Una dintre cele mai importante caracteristici ale integrării monolitice 3D este dimensiunea redusă a căilor de interconectare între straturi. Prin utilizarea unei rețele de interconectare dense

între straturi, anumite etape ale fluxului de proiectare sunt afectate pozitiv. De exemplu, rutarea este mai relaxată și consumul de putere este îmbunătățit.

Un obstacol important la ora actuală pentru punerea în practică a tehnologiei 3D monolitică îl reprezintă lipsa programelor de proiectare digitală. Într-un anumit sens, această problemă este de fapt aceeași care apare în cazul unor noi tehnologii: marii producători nu investesc în programe de proiectare noi până când nu se demonstrează viabilitatea lor comercială.

Prin urmare cercetarea pentru integrarea 3D s-a realizat cu ajutorul programelor de proiectare 2D, prin adaptarea fișierelor de intrare astfel încât să permită suprapunerea memoriilor peste logica funcțională care le însoțește. Unul dintre scopurile realizării integrării 3D este de a reduce complexitatea interconexiunilor și întârzierile asociate în tehnologia standard 2D. Acestea sunt considerate ca fiind principalele obstacole în calea creșterii în continuare a performanțelor pentru generațiile viitoare de circuite integrate. S-a ales pentru sinteza circuitului (translarea codului HDL în porți logice) programul Design Compiler de la Synopsys, iar pentru etapa de plasare și realizarea interconexiunilor, parte numită **place and route,** s-a folosit IC Compiler tot de la Synopsys. Tehnologia folosită la implementare este o tehnologie open source de la Synopsys în 32 de nanometri din pachetul educațional SAED_EDK32/28 Synopsys.

Pentru realizarea procesului de integrare tridimensională, s-a ales un circuit specific (dar metoda poate fi aplicată și la alte arhitecturi) care include două categorii principale de componente, care pot fi distribuite în mod aproximativ egal pe două straturi astfel:

- Pe un strat s-au plasat peste 30.000 de porți logice standard, utilizate pentru implementarea funcțiilor logice ale circuitului.
- Pe un al doilea strat s-au plasat 9 memorii, utilizate pentru stocarea datelor.

Această separare dintre porțile logice standard și memorii s-a făcut astfel încât aria celulelor de pe fiecare strat să fie distribuită uniform, iar fiecărui strat să i se aloce aproximativ 50% din suprafața totală a circuitului 3D. Această metodologie asigură o distribuție echilibrată a resurselor și optimizează utilizarea spațiului în cadrul procesului de integrare 3D.

Circuitul ales pentru implementare este un bloc care aparține procesorului openSPARC T2 cunoscut anterior ca UltraSparc T2, care este o arhitectură open-source dezvoltată de Sun Microsystem și care este un procesor de tip multi-core și multi-thread proiectat pentru performanțe ridicate în aplicații server. Blocurile componente principale din arhitectura procesorului openSPARC T2 sunt prezentate in figura 4.1 [117].

În figura 4.1 este prezentată diagrama bloc a unui nucleu de procesor UltraSPARC T2. Fiecare nucleu de procesor SPARC de pe UltraSPARC T2 include suport hardware complet pentru executarea a opt secvențe de instrucțiuni independente [117].

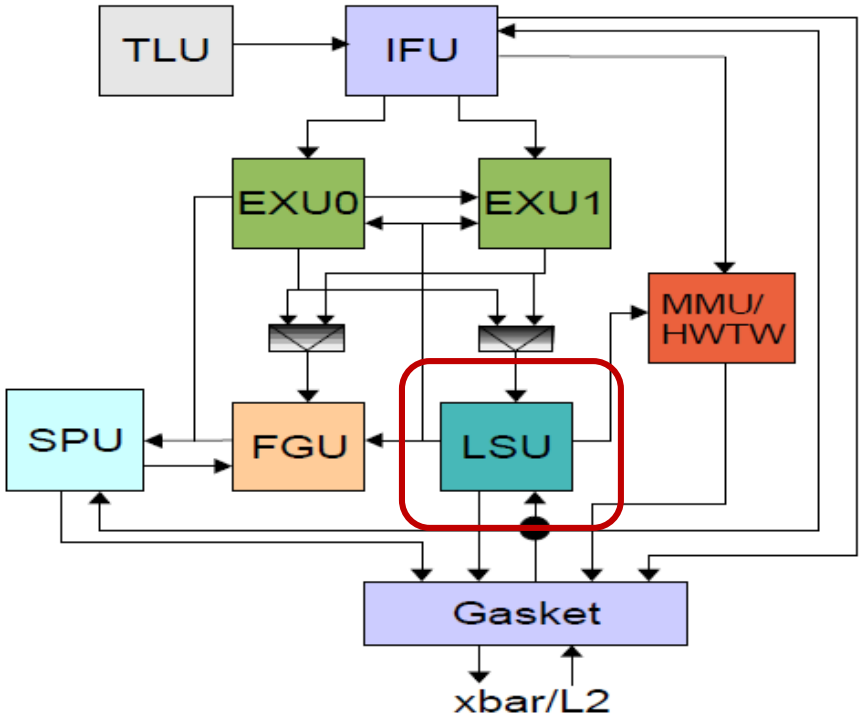


Fig. 4.1. Diagrama bloc a procesorului openSPARC T2 [117]

Nucleul procesorului SPARC, asa cum poate fi observat în figura 4.1 conține următoarele blocuri componente:

- IFU – unitatea de preluare a instrucțiunilor, este responsabilă pentru prelucrarea instrucțiunilor din cache-ul L1 de instrucțiuni și livrarea acestora către pipeline.
- EXU0 și EXU1 – două unități de execuție pentru operații întregi. Aceste două componente funcționează în paralel și sunt optimizate pentru procesarea instrucțiunilor legate de numere întregi.
- FGU – O unitate pentru operații cu virgulă mobilă și grafică, oferă suport în operațiile cu virgulă mobilă și calcul grafic.
- LSU – O unitate de încărcare-stocare, care gestionează accesul la memorie pentru încărcarea și stocarea datelor, utilizând cozi dedicate.
- SPU - O unitate criptografică pentru procesare de fluxuri, este o componentă hardware specializată care optimizează procesarea necesară pentru securitatea datelor.
- TLU – O unitate logică pentru tratarea excepțiilor, este responsabilă pentru gestionarea și tratarea excepțiilor și întreruperilor apărute în timpul execuției programelor.

- MMU – O unitate de gestionare a memoriei, care facilitează translarea adreselor virtuale în adrese fizice.
- Gasket Unit – O unitate intermediară pentru conectare și integrare.

Toate aceste componente lucrează împreună și oferă o arhitectură robustă, optimizată pentru procesare paralelă și multitasking.

Blocul ales pentru implementarea în 3D este LSU și selecția acestuia s-a făcut în urma unei analize riguroase post sinteză efectuată pe toate blocurile componente ale procesorului openSPARC T2. Principalul criteriu de selecție a fost identificarea unui circuit care să aibă o arie a logicii funcționale aproximativ egală cu aria totală a memoriilor folosite la implementarea circuitului 3D. Procesul de sinteză a blocurilor componente a procesorului openSPARC T2 a implicat evaluarea caracteristicilor fiecărui bloc în parte din punct de vedere al ariei, complexității și compatibilității cu integrarea 3D.

Blocul optim care a îndeplinit condițiile de utilizare maximă a spațiului, a fost circuitul LSU care are aria logicii funcționale aproximativ egală cu aria memoriilor.

Blocul LSU este responsabil pentru generarea adresei virtuale a tuturor operațiunilor de încărcare și stocare a datelor și pentru accesarea memoriei cache de date facilitând transferul eficient de informații între procesor și memorie. LSU folosește o coadă de încărcare care stochează și gestionează operațiunile de încărcare aflate în așteptare și o coadă de stocare a datelor care organizează operațiunile de stocare ce urmează a fi procesate. Acest bloc LSU permite emiterea unei singure operațiuni de încărcare sau stocare pe fiecare ciclu de ceas și în acest fel optimizează fluxul instrucțiunilor și utilizarea resurselor procesorului.

Fiind o componentă centrală în procesorul UltraSPARC T2, LSU joacă un rol critic în gestionarea accesului la memorie. Optimizările sale avansate contribuie semnificativ la performanța generală a procesorului, făcându-l capabil să gestioneze eficient aplicații complexe, cu cerințe mari de memorie.

Blocurile componente din structura LSU sunt cele din figura 4.2 (figura 4.2 este o captură a terminalului în care se găsește fișierul verilog cu modulul principal lsu.v, iar celelalte fișiere verilog sunt incluse în blocul top lsu), în care se poate observa lista cu blocurile incluse în LSU. Toate blocurile componente incluse în LSU, inclusiv blocul LSU, sunt descrise în verilog, iar sinteza blocului s-a făcut la nivel de poartă logică cu Design Compiler. S-a evitat sinteza ierarhică pentru a avea un control mai detaliat asupra ariei și un grad mai ridicat de control asupra procesului de optimizare. Prin concentrarea pe implementarea la nivel de poartă logică, a fost posibilă analizarea și ajustarea parametrilor de design cu o rezoluție mai fină, obținând rezultatul de 50%

arie a logicii Alinierea dintre circuitul sintetizat și obiectivele arhitecturale stabilite a fost realizată prin proiectarea circuitului LSU și aplicarea unei metodologii de sinteză la nivel de poartă logică.

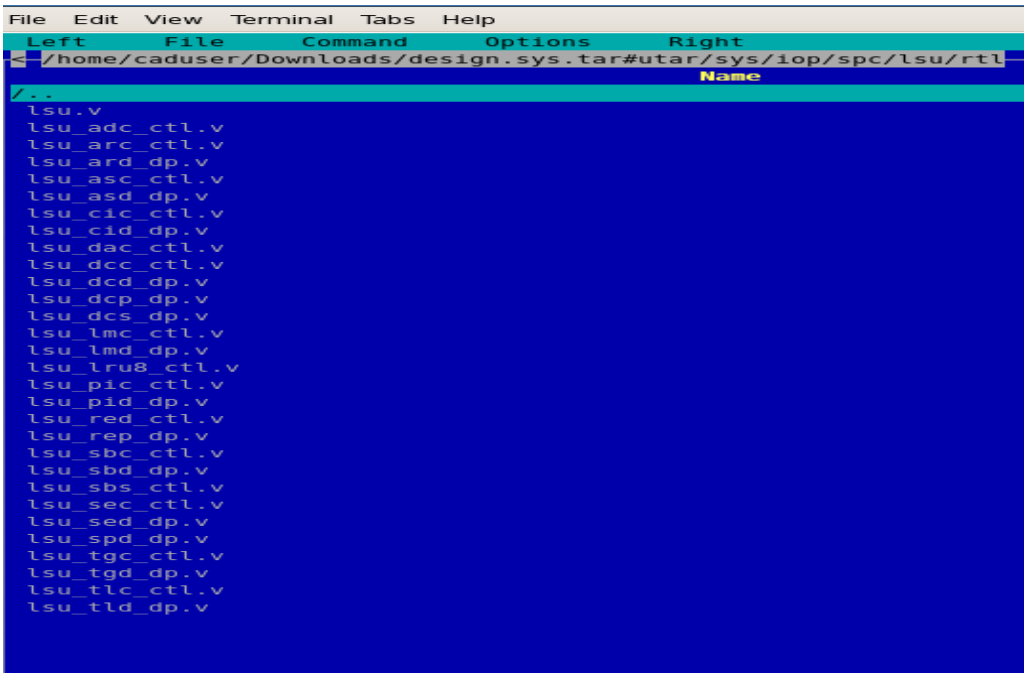


Fig. 4.2. Blocurile componente din structura LSU

Alinierea dintre circuitul sintetizat și obiectivele arhitecturale stabilite a fost realizată prin proiectarea circuitului LSU și prin aplicarea unei metodologii de sinteză la nivel de poartă logică. În cadrul acestui proces, descrierea la nivel RTL a fost transpusă într-o implementare hardware care respectă atât cerințele funcționale, cât și constrângerile de timp impuse. Metodologia de sinteză utilizată a urmărit obținerea unui compromis optim între viteza de funcționare și aria ocupată de circuit, fără a afecta comportamentul funcțional al acestuia. Astfel, implementarea rezultată răspunde obiectivelor arhitecturale stabilite și oferă o bază adecvată pentru etapele ulterioare ale fluxului de proiectare.

Blocurile de memorie utilizate la implementarea circuitului 3D constau în 8 memorii de tip **SRAM_128x30slp** și o memorie de tip **SRAM_128x38slp** și sunt generate cu SAED Memory Compiler. SAED Memory Compiler reprezintă un set de instrumente utilizat pentru a genera blocuri de memorie personalizate într-un proces specific de 90nm, 45nm, 32 nm, bazat pe tehnologia SAED, care este o bibliotecă academică oferită de Synopsys în scop educațional și pentru proiectele de cercetare. Memoriile generate cu acest set de instrumente sunt compatibile cu software--urile de la Synopsys folosite în acest proiect și anume Design Compiler și IC Compiler.

Memoriile **SRAM_128x30slp** sunt generate folosind o tehnologie de 32 nm și sunt de tip „single-port” (un singur port pentru scriere și citire) și sunt „low power” (cu două nivele de reduce

a puterii consumate: deconectarea alimentării bufferelor de ieșire respectiv deconectarea decodoarelor – în ambele cazuri datele din celulele de memorie rămânând nealterate; de asemenea memoriile acceptă deconectarea totală de la alimentare, caz în care conținutul memoriei se pierde), ceea ce asigură un consum redus de energie. Fiecare memorie are un număr de 128 cuvinte și fiecare cuvânt are o lățime de 30 de biți. În procesul de generare a memoriilor s-au luat în considerare o serie de verificări, care includ validarea funcționării și a caracteristicilor de performanță, după cum se poate observa și în figura 4.3.

```
#32nm mems
#
#mem_type = single_32 | dual_32 | singlelp_32 | duallp_32 | dual_90
mem_type=singlelp_32
#word_count = 16 | 32 | 64 | 128 | 256
word_count=128
#word_bits=4 .. 512
word_bits=30
instance_name=SRAM_128x30slp
do_logic=1
do_lib_nldm=1
do_lib_ccs=0
do_lef=1
do_spice=1
do_layout=1
do_gds=1
do_drc=1
do_lvs=1
do_cx=0
do_rcx=0
```

Fig. 4.3. Script cu comenzi pentru generarea memoriei SRAM_128x30slp

Memoria **SRAM_128x30slp** a fost generată considerând următoarele setări și verificări: activarea logicii “do_logic = 1”, utilizarea modelului non-linear “do_lib_nldm=1”, crearea fișierului LEF “do_lef=1”, generare modelului SPICE “do_spice=1”, generarea modelului layout “”do_layout=1”, generarea modelului grafic GDS “do_gds=1”, verificarea constrângerilor tehnologice “do_drc=1” și validarea echivalenței dintre schematic și layout “do_lvs=1”.

Memoria **SRAM_128x38slp** are un număr de 128 cuvinte și fiecare cuvânt având o lățime de 38 de biți și a fost generată având în vedere următoarele setări și verificări după cum se poate observa în figura 4.4: activarea logicii “do_logic = 1”, utilizarea modelului non-linear “do_lib_nldm=1”, crearea fișierului LEF “do_lef=1”, generare modelului SPICE “do_spice=1”, generarea modelului layout “”do_layout=1”, generarea modelului grafic GDS “do_gds=1”, verificarea constrângerilor tehnologice “do_drc=1” și validarea echivalenței dintre schematic și layout “do_lvs=1”.

```
#32nm mems
#
#mem_type = single_32 | dual_32 | singlelp_32 | duallp_32 | dual_90
mem_type=singlelp_32
#word_count = 16 | 32 | 64 | 128 | 256
word_count=128
#word_bits=4 .. 512
word_bits=38
instance_name=SRAM_128x38slp
do_logic=1
do_lib_nldm=1
do_lib_ccs=0
do_lef=1
do_spice=1
do_layout=1
do_gds=1
do_drc=1
do_lvs=1
do_cx=0
do_rcx=0
```

Fig. 4.4. Script cu comenzi pentru generarea memoriei SRAM_128x38slp

După etapa de sinteză, aria totală a blocurilor de memorie este de aproximativ **180000 µm²**, în timp ce aria celulelor logice standard ce s-au plasat pe primul strat este de aproximativ **115000 µm²**. Aceste valori reprezintă estimările preliminare ale suprafeței utilizate în circuit, înainte de parcurgerea etapelor de plasare, crearea arborelui de ceas, optimizări și rutarea conexiunilor.

În cadrul procesului de plasare și rutare, suprafața totală a circuitului poate crește semnificativ din cauza unor factori esențiali:

- Adăugarea de buffer-e și inversoare, necesare pentru optimizarea performanței circuitului, reducerea întârzierilor de propagare și îmbunătățirea robusteții semnalului.
- Construcția arborelui de distribuție a semnalului de ceas (Clock Tree Synthesis - CTS), proces prin care se introduc elemente suplimentare pentru a asigura o distribuție echilibrată a semnalului de ceas, minimizând skew-ul și maximizând sincronizarea între diferitele componente ale circuitului.

Astfel, aria finală a circuitului după finalizarea procesului de plasare și rutare va fi mai mare decât cea obținută după sinteză, datorită acestor elemente suplimentare necesare pentru asigurarea respectării constrângerilor de performanță, temporizare și fiabilitate. În configurația actuală, în care celulele logice standard au fost plasate pe primul strat, iar blocurile de memorie au fost plasate pe al doilea strat, elementele suplimentare utilizate pentru optimizare și construcția arborelui de distribuție a semnalului de ceas au fost adăugate exclusiv pe primul strat.

Aria inițială alocată acestui prim strat a fost deliberat dimensionată astfel încât să fie inferioară celei rezervate pentru memorie, permițând astfel integrarea celulelor suplimentare necesare până la atingerea unei arii echivalente cu cea a blocurilor de memorie. Această strategie a asigurat un echilibru optim între resursele disponibile și cerințele de proiectare, facilitând atât optimizările de performanță, cât și distribuția eficientă a semnalului de ceas.

Circuitul s-a implementat folosind o tehnologie CMOS de 32 nanometri, pusă la dispoziție în scop educațional, sub formă de pachet „open source", de către Synospys. Această tehnologie face parte din biblioteca SAED_EDK32/28nm, care respectă regulile de proiectare aferente procesului SAED32/28nm 1P9M 1.05V/1.8V/2.4V. Caracteristicile tehnologice specifice acestui pachet tehnologic educațional aparținând companiei Synopsys, conține suport pentru 10 straturi metalice, pornind de la metalul 1 (M1) și până la RDL (care reprezintă metalul superior, top metal) și condițiile de operare care includ tensiunile 1.05 V, 1.8V, 2.5V și temperaturile -40 C, +25 C respectiv +125 C.

Implementarea și analiza performanțelor circuitului s-a realizat în condiții tehnologice specifice și anume, la fel ca în cazul clasic de integrare 2D, s-au utilizat cele 2 cornere din tehnologie care conțin condițiile cele mai favorabile, respectiv cele mai dificile. Utilizând cele 2 cornere extreme pentru analiza de timp și putere, funcționarea circuitului este asigurată într-un spectru larg de condiții de funcționare, întregul spectru fiind acoperit de cele 2 cornere [118].

Cele doua cornere, numite „best-case" și „worst-case" sunt caracterizate de următorii parametri în aceasta tehnologie SAED32nm:

Corner-ul „worst-case" se caracterizează prin:

- Tensiunea de alimentare este de 1.06V;
- Temperatura de funcționare este de +125 C, temperatură care asigură condițiile de stres termic ridicat și toleranța circuitului la temperaturi ridicate.
- Proces de fabricație sub-optimal (slow-slow).

Cornerul „best-case" se caracterizează prin:

- Tensiunea de alimentare este de 0.945V;
- Temperatura de funcționare este de -40 C;
- Proces de fabricație optim (fast-fast).

Aceste 2 cornere „worst-case" și „best-case" în care s-a implementat și s-a analizat circuitul au fost utilizate pentru a acoperi întregul spectru de funcționare posibil al circuitului. Această abordare asigură nu doar o funcționalitate optimă a circuitului implementat, ci îndeplinește și cerințele industriei pentru proiectarea modernă și analiza a circuitelor 3D.

Memoriile, de diferite dimensiuni, s-au generat separat tot în tehnologie 32 nm, astfel încât sa se încadreze optim pe stratul al doilea din punct de vedere al ariei. Modificările blocurilor de memorie pentru adaptarea la integrare tridimensională s-au făcut în fișierele text tehnologice .LEF, iar ulterior au fost generate modele binare cu software-ul Milkyway pentru a putea fi folosite de către software-ul ICC Synopsys în cadrul procesului de implementare fizică.

Folosind circuitul și tehnologia descrise anterior, în etapa inițială s-a urmat procedura de lucru clasică pentru implementarea unui circuit pe un singur strat, corespunzătoare integrării planare 2D. Plasarea obișnuită 2D a circuitului care conține logică și memorii pe un singur strat poate fi observată în figura 4.5, unde în prima fază au fost plasate memoriile, iar apoi în funcție de constrângerile de timp și de conexiunile dintre componente s-a plasat logica în jurul memoriilor. Atât logica cât și memoriile sunt plasate pe un singur strat. Circuitul ales pentru implementare conține 50% memorii și 50% porți logice, astfel încât plasarea în aria aleasa să se facă optim și să se pună clar în evidență avantajele integrării tridimensionale.

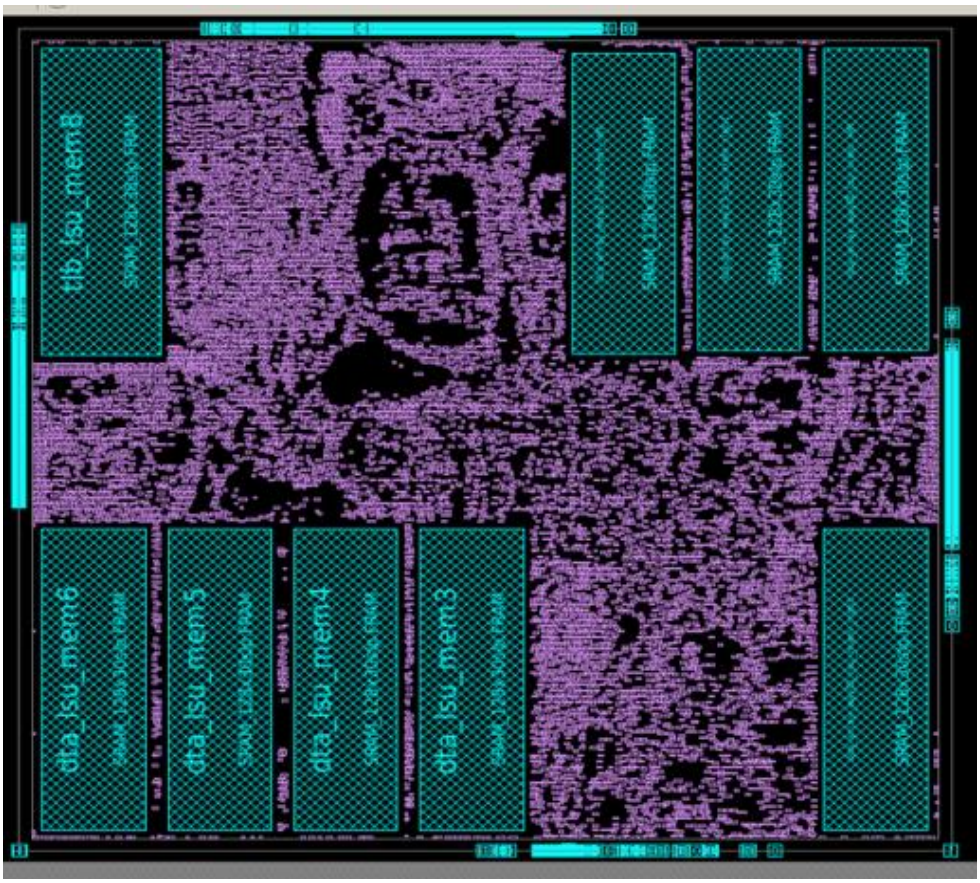


Fig. 4.5. Plasare memorii și porți logice 2D

Etapele parcurse sunt cele clasice din flow-ul 2D de proiectare a circuitelor integrate. Aria în care sunt plasate toate celulele în configurația 2D este de 485.526 μm^2.

După sinteză este generat netlistul care, împreună cu fișierele tehnologice și librăriile logice și fizice, sunt folosite ca fișiere de intrare în partea de backend și cu care este parcurs tot procesul de lucru, până la etapa de fabricare a circuitului integrat.

În diagrama din figura 4.6 sunt prezentați în detaliu toți pașii de după sinteză care trebuie urmați în partea de backend în cazul integrării planare 2D pentru a obține un circuit optim atât din punct de vedere al frecvenței cât și al ariei în care este încadrat [116]. O prezentare generală a procesului de lucru ilustrat în figura 4.6 include următorii pași principali:

- ✓ În prima etapă se realizează sinteza și se definesc constrângerilor esențiale, constrângeri de layout și timing care pot proveni pe de o parte din fișierele tehnologice și pe de altă parte din etapa de Backend – proces ce poarta denumirea de back-annotation (sunt adăugate constrângeri noi ce apar după plasare, crearea arborelui de ceas și care conțin întârzierile de pe conexiunile fizice, informații despre paraziți RC etc).
- ✓ Se generează apoi floorplan-ul și powerplan-ul, având ca intrări netlistul generat după sinteză, fișierele tehnologice precum și fișierele care conțin informații despre funcționarea logică a componentelor, fișierele RC și constrângerile de layout. Se utilizează scripturi dedicate pentru automatizarea întregului proces de lucru.
- ✓ După această etapă, se trece la plasarea componentelor și optimizarea inițială. Dacă apar probleme legate de respectarea constrângerilor de timp, se revine asupra pașilor anteriori pentru ajustări.
- ✓ Se continuă cu etapa de creare a arborelui de ceas (CTS), urmată de o altă etapă de optimizare, ținând cont de interferențele de tip crosstalk și de analiza OCV (On-Chip Variation). Dacă nici acum nu sunt îndeplinite condițiile de timing sau de alimentare, procesul se reia.
- ✓ Următorul pas este rutarea, însoțită de verificări DFM, DRC, analiza IR drop și a riscurilor de electromigrare. Se revine la etapele anterioare daca verificările raporteaza erori.
- ✓ După ce se obține o rutare corectă, se intră în faza de close timing și de analiză a integrității semnalului (SI). Ulterior, se finalizează verificările finale DRC, LVS, ERC și cele legate de efectul antenă.
- ✓ În final, se generează netlist-ul final și fișierul GDSII destinat fabricației. Procesul se încheie cu o serie de verificări fizice și formale, efectuate prin scripturi dedicate. Dacă apar erori în această ultimă etapă, fluxul se reia din punctele necesare pentru corectare.

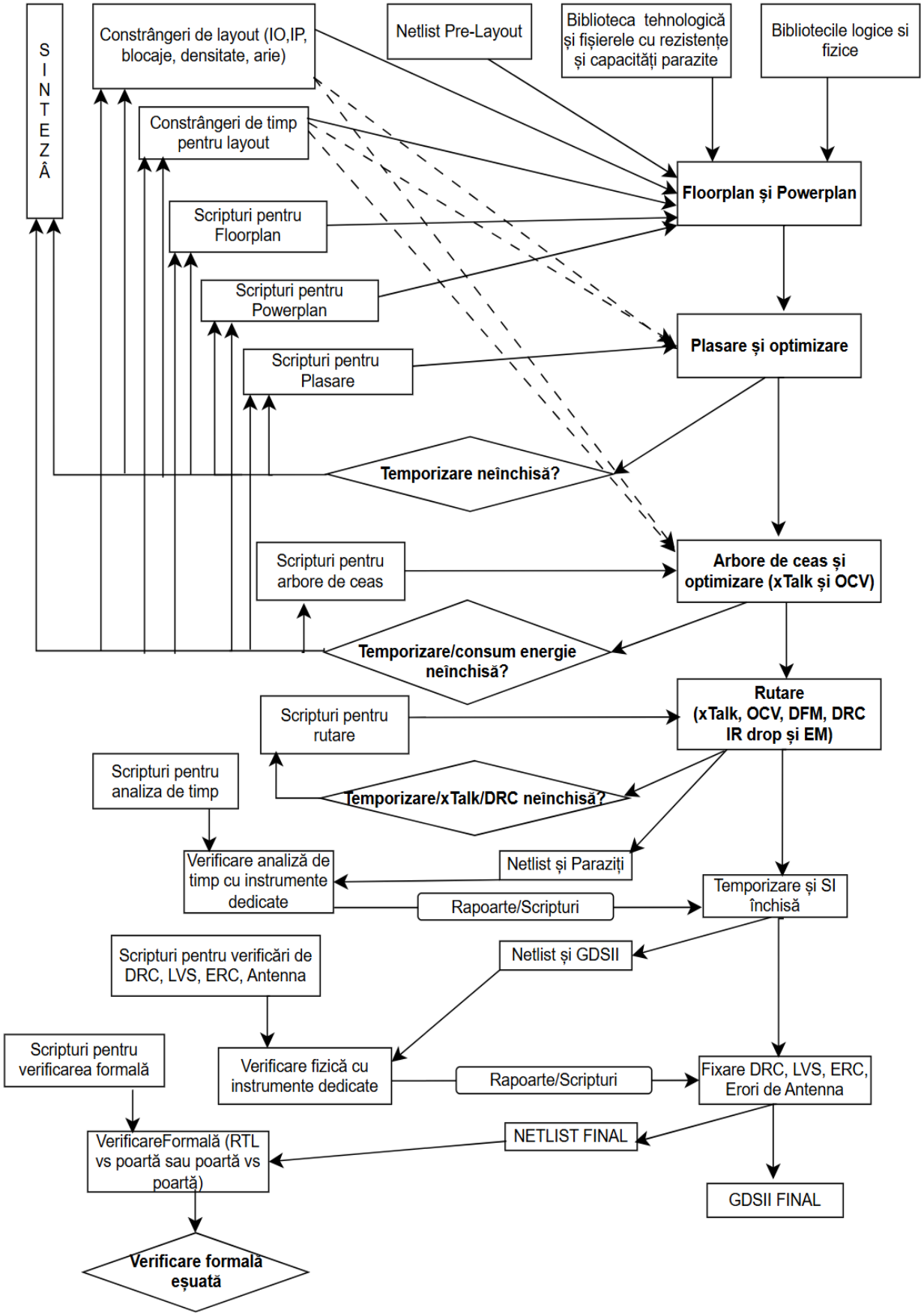


Fig. 4.6. Flow de lucru detaliat pentru etapa de BACKEND [116]

În faza de floorplan s-a conturat forma blocului LSU urmată de plasarea pinilor de intrare-ieșire și de plasarea memoriilor în funcție de conexiuni. După plasarea memoriilor s-a realizat blocarea memoriilor pe poziții fixe, iar în a doua etapă numită plasare s-au plasat celulele logice în aria rămasă. Atât plasarea memoriilor cât și plasarea porților logice, s-a făcut ținând cont de constrângerile tehnologice și de constrângerile de timp.

Forma floorplanului și proiectarea powerplanului sunt obținute ținând cont de constrângerile datorate pozițiilor pinilor de intrare-ieșire din circuit, de plasarea memoriilor, de densitatea ariei și de blocajele de plasare și rutare.

După etapa de plasare a fost implementat arborele de ceas în care s-a urmărit reducerea skew-ului și folosirea unui număr cât mai mic de inversoare și buffere astfel încât aria sa fie cât mai mică. A urmat apoi optimizarea căilor de timp și în final etapa de rutare în care s-a realizat trasarea metalelor pentru crearea conexiunilor dintre componente.

Optimizarea căilor de timp este necesar a fi făcută după fiecare etapă din procesul de lucru, în special după plasare când întârzierile pe căile de timp încep sa țină cont de informații reale date de lungimile traseelor și de toate caracteristicile tehnologiei folosită la implementare. Informații despre căile de timp apar într-un stadiu mai incipient și anume în etapa de sinteză, însă în acea etapă întârzierile pe căile de timp nu sunt reale, deoarece informațiile date de biblioteca fizică lipsesc și sunt doar estimate întârzieri bazate pe modele simulate.

Diagrama prezentată în figura 4.6 trebuie urmată pas cu pas și astfel se evită problemele care pot fi identificate mult prea târziu pentru eficientizarea timpului în procesul de fabricare al circuitului integrat și această metodologie de proiectare este testată în tehnologia planară 2D în producerea circuitelor integrate folosite în producție. Dacă unele probleme nu sunt identificate la timp în fluxul de lucru, timpul de proiectare și fabricare a unui circuit integrat crește direct proporțional cu numărul și respectiv gravitatea problemelor identificate [119, 120]. Prin analiza și rafinarea fluxului de lucru, identificarea blocajelor și a problemelor timpurii dintr-un circuit și metodologia care este urmată pentru fabricarea lui, procesul de optimizare va asigura o traiectorie minimizată de erori, de la sinteză și până la proiectarea finală pregătită pentru fabricare. Luarea în considerare a tuturor detaliilor, în fiecare etapă a fluxului de lucru, va ajuta inginerul de backend să rezolve problema identificată mai rapid și să economisească timp. S-au creat scripturi detaliate pentru automatizarea procesului de lucru în fiecare etapă, inclusiv floorplan, powerplan, plasare de celule standard și blocuri de memorii, crearea arborelui de ceas, rutarea conexiunilor și analizele fizice și logice. Fluxul de lucru pentru producerea circuitului integrat a fost optimizat folosind bucle la toate nivelurile, extrăgând informații importante la toate nivelurile și îmbunătățind procesul de lucru. Mai mult, tehnicile de optimizare contribuie semnificativ la precizie și calitate

în implementarea proiectării unui circuit integrat. Acest flux de lucru descris în figura 4.6 a fost folosit pentru a implementa în 3D circuitul integrat folosit la experimentare cu ajutorul instrumentelor automate 2D de la Synopsys.

Pentru a produce circuite integrate 3D sunt cel puțin două direcții care pot pregăti drumul pentru implementare: o direcție poate fi îndreptată spre producerea de tehnologii care să permită integrarea 3D și cealaltă direcție poate fi îndreptată spre dezvoltarea de instrumente (software) care să ia în calcul cea de-a treia dimensiune. Dezvoltarea de noi tehnologii și noi instrumente va ajuta enorm progresul în direcția dezvoltării unui proces de lucru pentru implementarea circuitelor integrate 3D.

Proiectarea detaliată a unui circuit integrat 2D, care include toate etapele din partea de backend, de la faza inițială de floorplan și până la ultima etapă numită rutare, constituie punctul de plecare pentru dezvoltarea flow-ului de lucru pentru integrarea 3D. Etapele obișnuite din fluxul de lucru pentru proiectarea circuitelor integrate 2D care cuprind: floorplan, powerplan, plasare și optimizare, sinteza arborelui de ceas și rutarea, pot fi adaptate și optimizate pentru implementarea circuitelor integrate 3D. În afara de etapele care trebuie urmate obligatoriu în proiectarea circuitelor integrate 2D, mai sunt câteva opțiuni de optimizare a fluxului de lucru pentru a accelera procesul de fabricare a circuitului integrat și asta implică verificări timpurii pentru a extrage informații de la începutul flowului și re-verificarea tuturor rapoartelor după fiecare etapă din flux de lucru.

Metodologia folosită la plasarea memoriilor deasupra celulelor logicii în cazul integrării 3D, este compusă din pași care utilizează în mare parte elemente ale metodologiilor existente de proiectare a circuitelor integrate 2D, dar care este completată cu pași specifici și necesari pentru integrarea 3D [121].

Utilizând flow-ul 2D de proiectare a circuitelor integrate, în prima etapă s-au plasat memoriile și pinii de intrare și ieșire și s-au fixat pozițiile lor în locațiile respective. Pentru fiecare memorie de pe stratul 2 s-a generat o reprezentare specială care să permită programului de proiectare a circuitelor integrate 2D plasarea de celule logice sub memorii. Pentru a evita suprapunerea memoriilor peste alte memorii separarea pe cele două straturi s-a făcut astfel: memorii pe stratul 2 și logică pe stratul 1. Generarea reprezentării memoriilor a implicat o formă specială de serpentină a cărei pas a fost mai mic decât cea mai mică memorie din circuit, ca în figura 4.7. Serpentina permite programului de proiectare să plaseze logica sub memorii, dar nu va permite să fie plasate memoriile unele peste altele. Înainte de a se ajunge la această reprezentare a memoriilor sub forma de serpentină, au fost testate reprezentări sub forme de litere pentru memorii și anume U, L, T, E, F, H, “MELC” dar s-au dovedit a nu fi la fel de eficiente pentru că în procesul

de plasare automată a memoriilor, software-ul permite, în unele cazuri, suprapunerea memoriilor între ele. Deși cea mai bună formă dintre cele enumerate anterior a fost forma “MELC”, totuși ea nu a fost o metodă care să satisfacă cerințele necesare pentru această cercetare pentru că ocupa mai multă arie decât forma “serpentină”.

Utilizând reprezentarea memoriilor sub formă de serpentină, instrumentul 2D de proiectare a circuitelor integrate, vede o utilizare a ariei mult mai mică (~1%) decât în cazul în care sunt folosite memoriile în forma lor reală, iar aceasta permite acceptarea memoriilor și a logicii în 50% din aria reala (~49% este logica funcțională și ~1% sunt memorii).

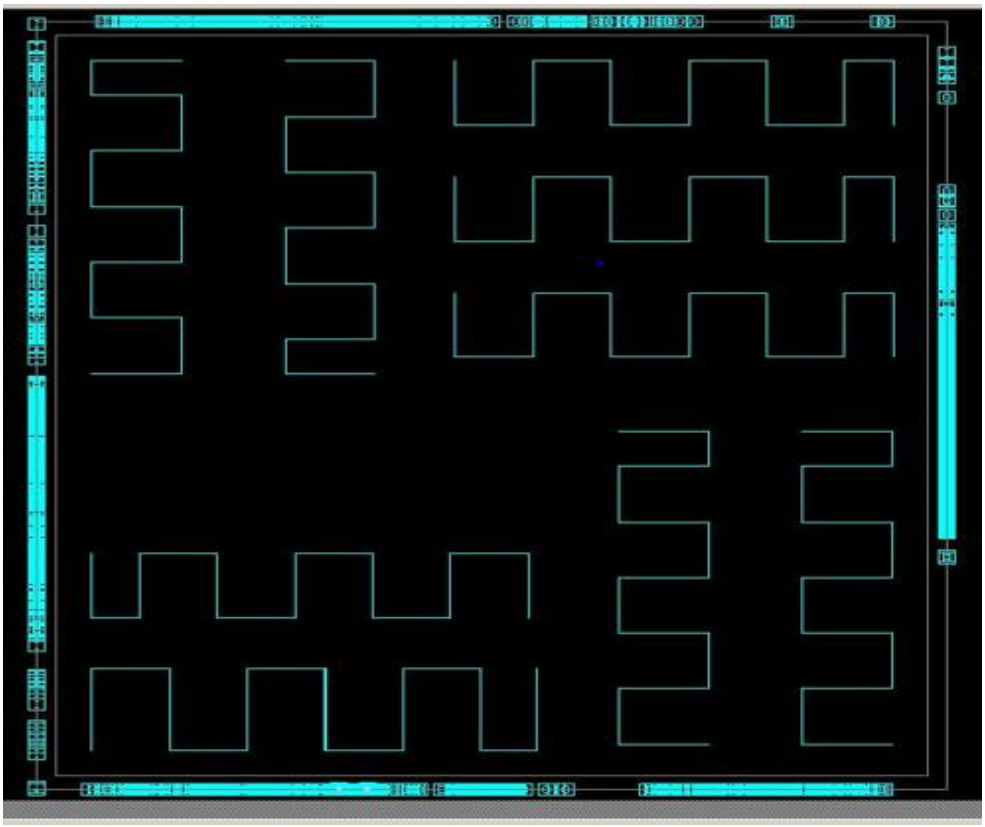


Fig. 4.7. Memorii cu formă de serpentină plasate în etapa de floorplan

Fișierul tehnologic LEF utilizat la obținerea formei de serpentine a memoriei **SRAM_128x38slp** se poate observa în ANEXA 2.

În prima etapă din flowul de backend, etapă numită floorplan, cele 9 blocuri de memorie cu forme speciale de serpentină sunt plasate în aria calculată pentru plasarea celulelor logice și sunt fixate pe poziții după cum se poate observa în figura 4.7. Această plasare a blocurilor de memorii cu forme speciale de serpentine, forțează software-ul 2D să calculeze o arie mai mică necesară plasării memoriilor și va permite plasarea celulelor logice pe primul strat, sub memorii,

fără a raporta erori de suprapunere între cele 2 straturi și fără a a vedea o utilizare mai mare de 100%. Tot în aceasta etapă de floorplan sunt plasați și pinii circuitului pe marginea die-ului.

În etapa a doua s-a realizat plasarea globală a logicii sub memorii în funcție de constrângerile de timp și de conexiunile dintre componente, așa cum se observă în figura 4.8. Utilizarea ariei folosită pentru plasarea logicii, crește în cazul integrării 3D a circuitului, având în vedere micșorarea suprafeței pe care se plasează întreaga logică a blocului LSU. De asemenea, lungimea conexiunilor dintre componente se reduce datorită scurtării distanței dintre acestea – așezarea pe verticală [122]. Aria în care sunt plasate toate celulele în configurația 3D este de 259.995 μm2.

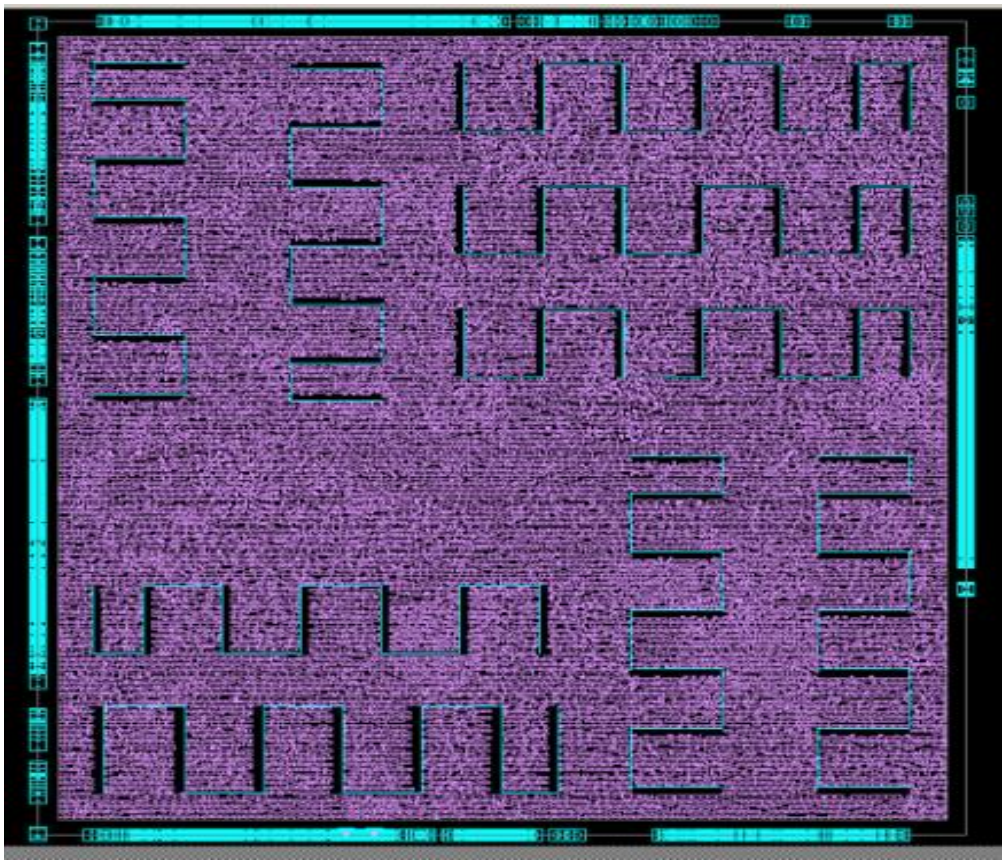


Fig. 4.8. Memorii cu formă de serpentină plasate deasupra celulelor logice [116]

După plasarea globală a logicii, s-a efectuat plasarea detaliată a componentelor logice care a implicat optimizarea componentelor logice și a legăturilor dintre ele, fapt evident prin creșterea ariei celulelor combinaționale și a numărului de buffere respectiv inversoare folosite pentru respectarea constrângerilor de timp din circuit. Această creștere este mai mică în cazul implementării 3D datorită micșorării distanțelor între memorii și logica aferentă.

Parcurgerea etapelor de implementare 3D a circuitului până la această fază, s-a făcut modificând fișierele tehnologice, precum și scripturile folosite în flow-ul de implementare.

Modificările s-au realizat iterativ, adaptând și modificând metoda de implementare în funcție de rezultate. După plasarea memoriilor sub formă de serpentină deasupra celulelor standard din circuit, s-au înlocuit „reprezentările serpentină” cu „reprezentările fantomă” (reprezentări care permit suprapunerea totală a memoriilor cu celulele standard logice) și se poate observa în figura 4.9 modul în care memoriile fantomă înlocuiesc memoriile serpentină. Nu s-au putut folosi reprezentările fantomă de la începutul fazei de plasare, și au trebuit înlocuite cu reprezentările serpentină, pentru că doar primul pas de plasare globală (create_placement) nu ia în calcul atributul de OVERLAP folosit în reprezentările fantomă – din acest motiv software-ul vedea o utilizare spre 200%.

Flow-ul a fost implementat folosind instrumentele Synopsys Design Compiler și Synopsys IC Compiler, iar bibliotecile pentru memorii au fost ajustate manual.

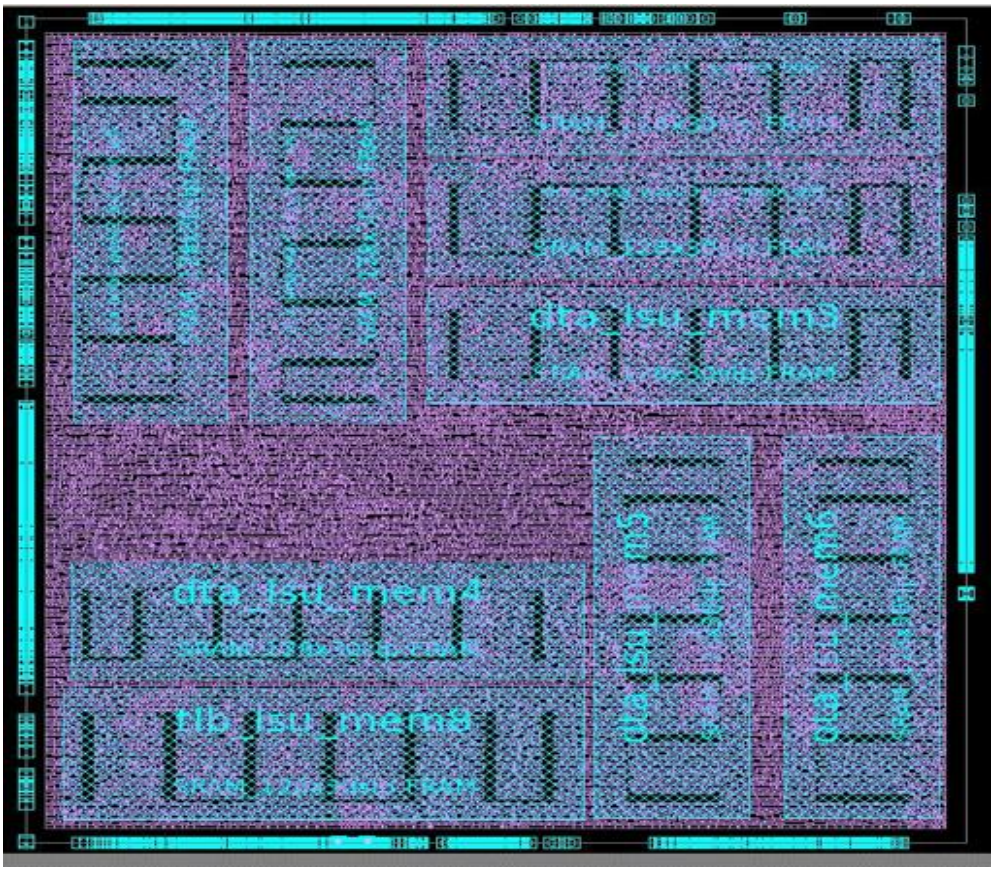


Fig. 4.9. Înlocuirea serpentinelor cu memoriile fantomă din fișierele tehnologice

Înlocuirea formelor memoriilor sub formă de serpentină cu memorii fantomă, se face doar după prima etapă de plasare din flux de lucru. Înlocuirea cu memoriile reale se face doar la final, înainte de exportul de GDS. Trecerea dintr-o etapă în alta a flow-ului se face folosind reprezentările virtuale (serpentină respectiv fantomă) ale memoriilor pentru a evita posibilele erori de DRC (de plasare și de rutare) care ar apărea în cazul în care s-ar folosi memoriile reale.

Înlocuirea memoriilor cu forma de serpentina cu memorii fantomă este necesară pentru a folosi spațiul ocupat anterior de serpentină în vederea rezolvării problemelor de timing și evident folosirea optimă a ariei. S-a făcut o optimizare de plasare care a rearanjat celulele logice ținând cont de constrângerile de timp și arie. În figura 4.10 se poate observa rearanjarea respectiv plasarea

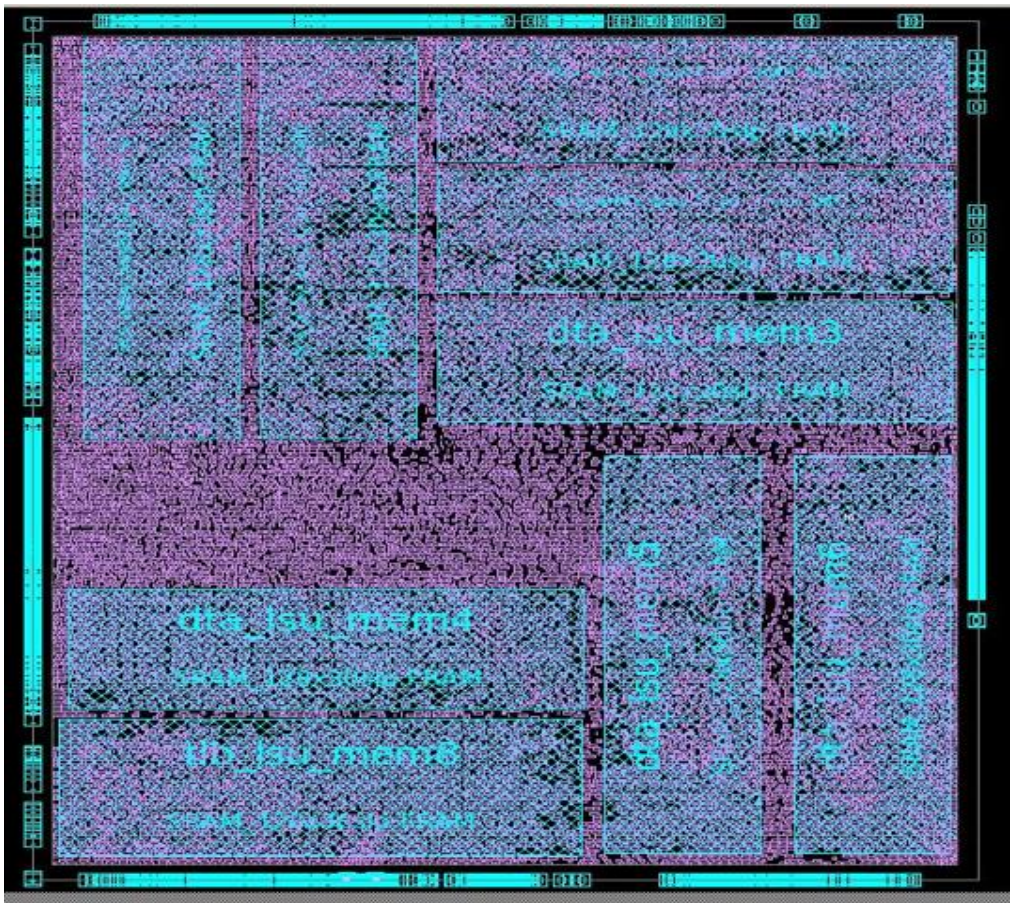


Fig. 4.10. Plasarea finală a memoriilor deasupra celulelor standard

automată a celulelor logice sub memorii după optimizare și reumplerea spatiilor lăsate de plasarea memoriilor în formă de serpentine.

4.2. Descrierea detaliată a etapelor premergătoare și a etapelor parcurse pentru plasarea blocurilor de memorie peste celulele logice

În acest subcapitol, se prezintă în detaliu pașii necesari pentru plasarea blocurilor de memorie deasupra celulelor logice standard, utilizând software-ul 2D IC Compiler. În mod obișnuit, software-ul 2D operează într-o singură dimensiune, vizualizând doar primul strat al circuitului și nu este capabil să perceapă sau să integreze al doilea strat. Astfel, se va detalia procesul de depășire a acestei limitări tehnice și implementarea unei metode care permite plasarea

blocurilor de memorie într-o zonă destinată exclusiv celulelor logice, conform restricțiilor impuse de specificațiile software-ului utilizat.

4.2.1. Etapa 1: Evaluarea circuitului

Înainte de a ajunge la concluzia că cel mai potrivit circuit trebuie să aibă jumătate memorii și jumătate celule logice, au fost testate blocuri funcționale de diferite dimensiuni și în diferite arii. Dar cel mai optim bloc funcțional în care să se pună în evidență avantajele integrării 3D comparativ cu integrarea 2D, folosind metoda descrisă anterior, este cel în care blocurile de memorie ocupă jumătate din arie și pot fi plasate pe stratul 2, iar celulele logice ocupă cealaltă jumătate de arie și pot rămâne pentru a fi plasate pe primul strat. Cele două straturi au utilizări diferite, dar asta depinde de mărimea circuitului, de mărimea blocurilor de memorie, de numărul blocurilor de memorie și de forma circuitului. Forma suprafeței (în care sunt plasate celulele logice) aleasă în situația expusă este dreptunghiulară pentru a încadra cel mai optim memoriile în arie, dar în cazul general al unui circuit integrat care trebuie implementat, forma depinde de constrângerile circuitului. Există mici excepții de la regula 50% memorii și 50% celule logice standard în cazul circuitului implementat și analizate în această lucrarea și ele pot fi concretizate astfel:

Excepția 1: există memorii personalizate care permit alipirea pe 3 laturi – au toți pinii pe o singura latura. Astfel de memorii vor permite în 3D alipirea pe toate cele 4 laturi în faza de plasare și asta poate conduce la o utilizare de 100% pe stratul al 2-lea [123].

Excepția 2: Memoriile mici ca dimensiune pot rămâne pe stratul 1, alături de celulele logice standard [123].

Toate componentele din circuit, celulele logice standard și memoriile, au toate modelele necesare în formatele următoare: .lib (fișier în care sunt modelate întârzierile prin componente și consumul de energie), .db (fișier obținut după compilarea modelului .lib), .LEF (fișier care conține reprezentarea fizică a componentelor și parametrii electrici) respectiv .MW (librăriile milkyway care sunt generate din compilarea librăriilor în format .LEF împreună cu fișierul tehnologic .tf și librăriile compilate în format .db și transformate în model vizibil).

Deoarece pinii memoriilor pe Metal 1 ar crea probleme în timpul plasării din cauza faptului că toate celulele logice standard au pinii pe Metal 1 și astfel pot apărea scurtcircuite, toți pinii memoriile sunt mutați pe Metal 7 (implementarea are loc pe 6 metale) și astfel putem considera Metal 7 ca fiind primul metal de pe stratul al doilea.

Considerând aria memoriilor și utilizarea ariei de pe stratul al doilea, precum și aria celulelor logice standard și utilizarea ariei de pe stratul 1, se poate calcula un factor de reducere pentru dimensiunea memoriilor. Fiecare memorie trebuie redusă cu acest factor pe fiecare dintre

cele 4 laturi, lățimea și înălțimea memoriilor se reduce cu 2 înmulțit cu factorul de reducere. Aceasta se datorează modului în care instrumentul 2D de la Synopsys evaluează aria întregului circuit în etapa de început a plasării blocurilor de memorie. Instrumentul 2D Synopsys de evaluare și verificare a ariei circuitului nu întelege și nu cunoaște cel de-al doilea strat și consideră ca tot circuitul este plasat pe același strat, prin urmare o utilizare a ariei mai mare de 100% nu este acceptabilă . Aria totală a blocurilor de memorie ar trebui să fie echivalentă cu aria celulelor logice standard (de exemplu o utilizare de 90% pe stratul al doilea duce la o reducere a ariei totale a blocurilor de memorii cu aria celulelor logice considerând o utilizare de 100%), minus 10% din aria de pe stratul al doilea. Această micșorare a dimensiunii memoriilor este necesară pentru a permite instrumentului de plasare să poată optimiza plasamentul, altfel în situația în care memoriile ar ocupa 100% instrument de plasare nu ar avea opțiuni de plasare.

4.2.2. Etapa a 2 a: Plasarea blocurilor de memorii

În aceasta etapă, fișierele tehnologice LEF, care conțin informații despre forma fizică a memoriilor și pozițiile pinilor, sunt actualizate cu noile valori ale dimensiunilor blocurilor de memorii, obținute după reducerea dimensiunilor pe fiecare parte cu valoarea calculată în faza inițială. O altă actualizare care apare în fișierele tehnologice LEF, este forma de ”serpentină”, care reprezintă o obstrucție având forma de ”șarpe”, care are o ”constanta de pas a meandrului” mai mică decât dimensiunea (lățimea și lungimea) celei mai mici memorii de pe stratul al doilea. În figura 4.11 au fost puse în evidență “constanta de pas a meandrului” și forma de “șarpe” a unui bloc de memorie. Aceste actualizări din fișierele tehnologice LEF, formă și dimensiune, împreună cu fișierele originale .db sunt folosite pentru a genera noile fișiere folosind Milkyway de la Synopsys, fișiere care conțin informații despre noile forme ale blocurilor de memorii, fișiere numite “serpentine reduse”. Astfel este definit pentru fiecare memorie un “HALO”, care reprezintă spațiul liber (tradus ca și blocaj de plasare pentru orice fel de celulă) lăsat în jurul memoriilor la plasarea automată și a cărui valoare este egală cu 2 înmulțit cu factorul de reducere a memoriilor plus distanța minimă dintre memorii (s-a calculat 5 microni distanța minimă dintre memorii). Cu această lațime a “HALO”-ului se poate trece de etapa de verificare a ariei de către instrumentul 2D de verificare a ariei după plasarea blocurilor de memorie. Factorul de reducere în situația de față este de 15 microni, prin urmare “HALO” are valoarea de 35 microni și valoarea acestuia poate fi observată în ANEXA 3 în scriptul de plasare automată 3D, la etapa de plasare a blocurilor de memorii.

“HALO” este un factor important [124] care are un impact major în evitarea congestiei de rutare în jurul blocurilor macro și care este calculat în funcție de tehnologie și de regulile de circuit.

Un calcul corect al HALO-ului determină un acces facil la pinii blocurilor de memorie și astfel se evită aglomerarea traseelor de rutare și a blocajelor în jurul celulelor macro.



Fig. 4.11. Forma de serpentină și constanta de pas a meandrului

Adițional se poate considera o orientare specifică de plasare a blocurilor de memorii, de exemplu tehnologiile de 40 nm de astăzi nu permit orientări R90 pentru blocurile de memorii. Instrumentul care plasează toate blocurile de memorii, poziționează toate blocurile fără a le suprapune datorită formei de „șarpe” și fără a le plasa prea aproape unul de altul datorită cerinței de “HALO”. Plasarea care rezultă ținând cont de acești noi parametri definiți, este făcută pastrând o anumită distanță între blocurile de memorii, precum și între blocurile de memorii și marginea circuitului, distanță ce are o valoare care va permite ulterior expandarea blocurilor de memorii la forma reală fără a suprapune memoriile între ele în interiorul ariei necesare plasării memoriilor. În această etapă se pot alege de asemenea poziții specifice (fixate) de plasare pentru anumite blocuri de memorii (în funcție de cerințele circuitului implementat), iar restul blocurilor de memorii să fie plasate automat de către instrumentul de plasare. După plasarea automată a blocurilor de memorii de către instrumentul de plasare, pozițiile memoriilor pot fi ajustate și manual în caz de necesitate.

4.2.3. Etapa a 3 a: Plasarea detaliată

În această etapă se folosesc fișiere Milkyway numite “serpentine”, fișiere care sunt generate folosind fișierele .db originale și fișierele tehnologice .LEF actualizate cu forma „șarpe”. Fișierele “serpentină” utilizate în această etapă conțin memorii sub formă de serpentină de dimensiune neredusă, “șarpele” este extins până la marginea memoriei reale. Pentru a transmite informațiile de plasare a blocurilor de memorii de la etapa a 2 a la etapa a 3 a sunt două modalități: prima metodă este copierea bazei de date din etapa a 2 a și legarea acesteia la noua biblioteca Milkyway, iar a doua metoda constă în salvarea plasării blocurilor de memorii în etapa a 2 a și încărcarea acesteia în etapa a 3 a. A doua metodă este mai eficientă și odată realizată această metodă, plasarea blocurilor este blocată. Plasarea obișnuită, similară cu plasarea în flow-ul 2D urmează după această fază.

4.3 Implementarea și analiza arborelui de ceas

În acest subcapitol se continuă flow-ul de proiectare a circuitelor integrate și se construiește arborele de ceas în cele 2 scenarii 2D și 3D. Se prezintă arhitectura arborelui de ceas în cele două scenarii de integrare: 2D (planară) și 3D (tridimensională) și se analizează rapoartele generate după implementare. Scopul acestei analize este de a compara caracteristicile fiecărui scenariu și de a evidenția avantajele pe care integrarea tridimensională le poate oferi față de abordarea tradițională 2D.

După etapa de plasare a componentelor circuitului urmează implementarea arborelui de ceas în vederea obținerii unui skew eficient prin inserarea de buffere și inversoare pe căile semnalului de ceas. Sinteza arborelui de clock reprezintă o etapă importantă în flow-ul de proiectare a circuitelor integrate și joacă un rol important în implementarea unui arbore de clock echilibrat care să ajute la atingerea constrângerilor de timp din circuit. Scopul principal în construirea arborelui de ceas este de a reduce atât latența pe arbore cât și skew-ul dintre componentele secvențiale din circuit, care sunt critice din punct de vedere al timing-ului și în același timp de a menține o arie cât mai mica a circuitului.

S-a implementat în primă fază arborele de ceas pentru integrarea 2D a circuitului și în figura 4.12 sunt puse în evidență prin marcarea cu alb, bufferele și inversoarele care formează

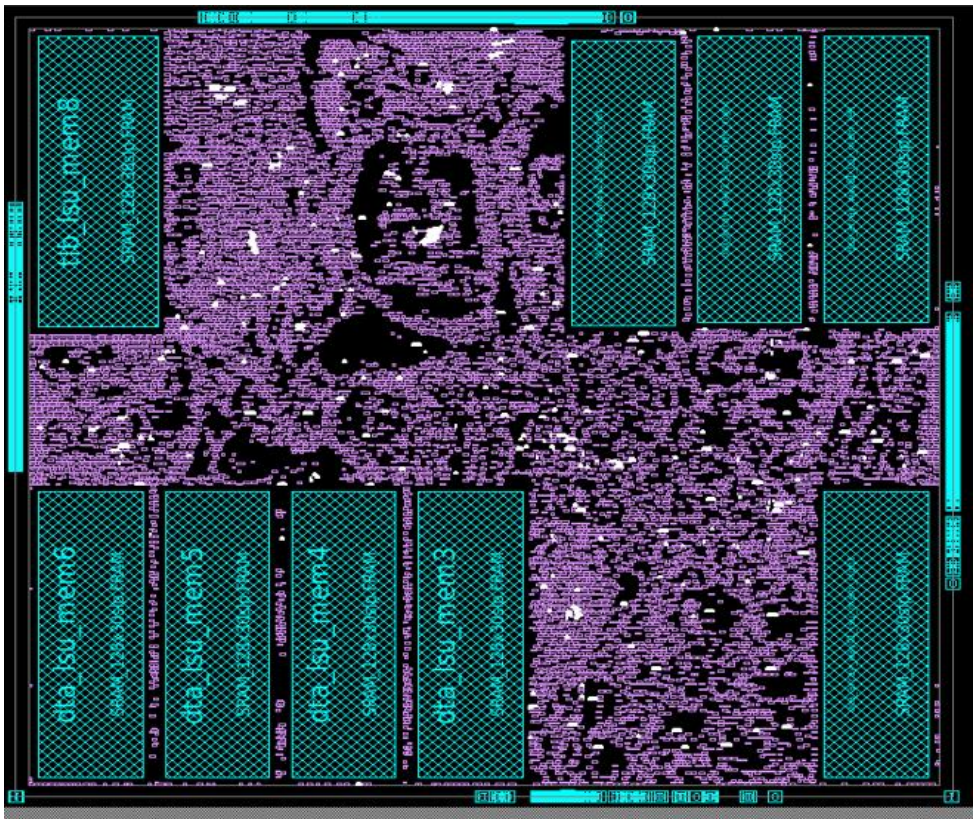


Fig. 4.12. Arborele de ceas implementat în scenariul 2D [114]

arborele de ceas implementat pentru o perioada de 3ns. Structura arborelui de ceas în scenariul 2D cuprinde 8 nivele cu 274 de buffere, respectiv inversoare care ajung la cele 7729 de celule secvențiale din circuit.

După plasarea logicii pe stratul 1 și a blocurilor de memorii sub formă de serpentină pe stratul 2, s-a implementat arborele de ceas pe stratul 1 în dimensiunea 3D folosind instrumentul de construire al arborelui de ceas încorporat în ICC Synopsys. Biblioteca utilizată la implementare conține buffere și inversoare de diferite dimensiuni pornind de la 0X și până la 32X, unde acest factor X indică o scalare proporțională a dimensiunii tranzistoarelor interne ale celulei față de o referință. Cu cât factorul X este mai mare, cu atât capacitatea celulei logice este mai mare și astfel celula respectivă poate conduce sarcini capacitive mai mari reducând astfel întârzierea semnalului pe trasee lungi și reducând de asemenea și consumul de energie [125]. În structura arborelui de ceas s-au utilizat buffere și inversoare din tehnologia disponibilă, având mărimi cuprinse între 2X și 32X, pentru a asigura un arbore echilibrat și cu o arie minimizată. Selecția dimensiunilor componentelor a fost realizată astfel încât să optimizeze atât latența semnalului de ceas, cât și consumul de energie, evitând în același timp problemele legate de skew. Spre deosebire de distribuția arborelui de ceas 2D unde bufferele și inversoarele utilizate în structură sunt plasate în jurul memoriilor, în cazul plasării 3D arborele de ceas este împrăștiat pe toată suprafața alocată plasării celulelor logice. În lipsa constrângerilor evidente de arie din scenariul 2D, unde toate componentele trebuie plasate pe un singur strat, distribuția arborelui de ceas în configurația 3D devine mult mai uniformă. Această uniformizare se datorează posibilității de a dispersa mai eficient nodurile arborelui pe mai multe nivele, reducând concentrarea excesivă a resurselor într-o singură zonă. O consecință directă a acestei distribuții optimizate este scăderea semnificativă a necesarului de buffere și inversoare. Într-un arbore de ceas 2D, propagarea semnalului trebuie să acopere distanțe mai mari pe un singur plan, ceea ce implică un număr crescut de buffere pentru compensarea întârzierilor și menținerea sincronizării. În schimb, în arhitectura 3D, distanțele sunt reduse datorită stratificării componentelor, permițând o propagare mai eficientă a semnalului de ceas și, implicit, reducând necesitatea inserării unui număr mare de elemente de regenerare a semnalului. Comparativ cu tehnologia TSV, unde conexiunile pe verticală sunt de dimensiuni mari și în număr restrâns, în situația monolitic 3D conexiunile pe verticală (MIV-urile) sunt dense și cu capacitate redusă) și astfel arborele de ceas este mai avantajos, unele studii menționând chiar un skew egal cu 0 [126]. În figura 4.13 este pus în evidență cu alb arborele de ceas în scenariul 3D și se pot observa buffere și inversoarele plasate sub blocurile de memorii în formă de serpentine.

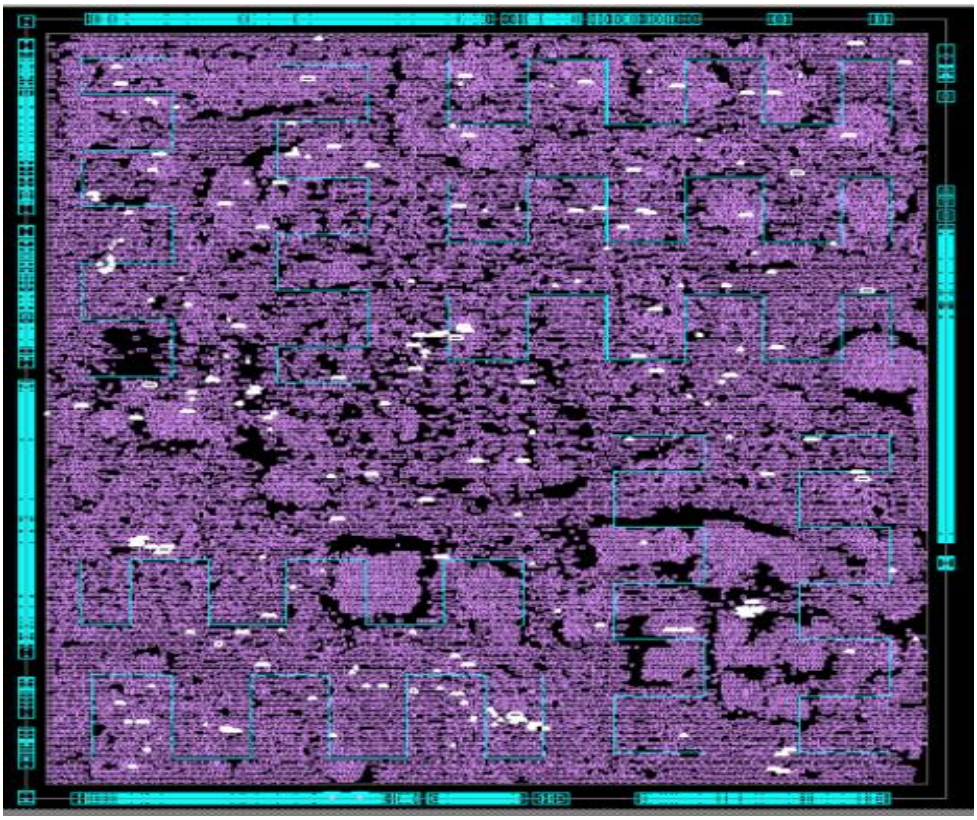


Fig. 4.13. Arborele de ceas implementat în scenariul 3D [114]

În raportul detaliat din tabelul 4.1 [114] sunt prezentate caracteristicile arborelui de ceas "l2clk" în varianta 2D și anume: scenariul "funcțional" în care a fost implementat arborele de ceas folosind o perioadă de 3 ns, pe un număr de 8 nivele, la un circuit care conține un număr de 7729 de bistabile, și la implementarea căruia s-au folosit un număr de 274 buffere cu arie totală de 1565.27 um și un număr total de celule de clock 464 cu o arie totală de 2057.03 um. Skew-ul arborelui de ceas in scenariul 2D este de 0.056 ns. În tabelul 4.2 sunt rezultatele arborelui de ceas generat în scenariul 3D pentru a realiza o comparație cu rezultatele obținute după implementarea arborelui de ceas în varianta 2D.

În varianta 3D raportul arborelui de ceas "l2clk" prezintă următoarele caracteristici și anume: scenariul "funcțional" în care a fost implementat arborele de ceas folosind o perioadă de 3 ns, pe un număr de 9 nivele, la un circuit care conține un număr de 7729 de bistabile, și la implementarea căruia s-au folosit un număr de 204 buffere cu arie totală de 1383.56 um și un număr total de celule de clock 393 cu o arie totală de 1879.39 um. Skew-ul arborelui de ceas în scenariul 3D este de 0.046 ns.

Tabelul 4.1 Raport arbore de clock 2D vs 3D

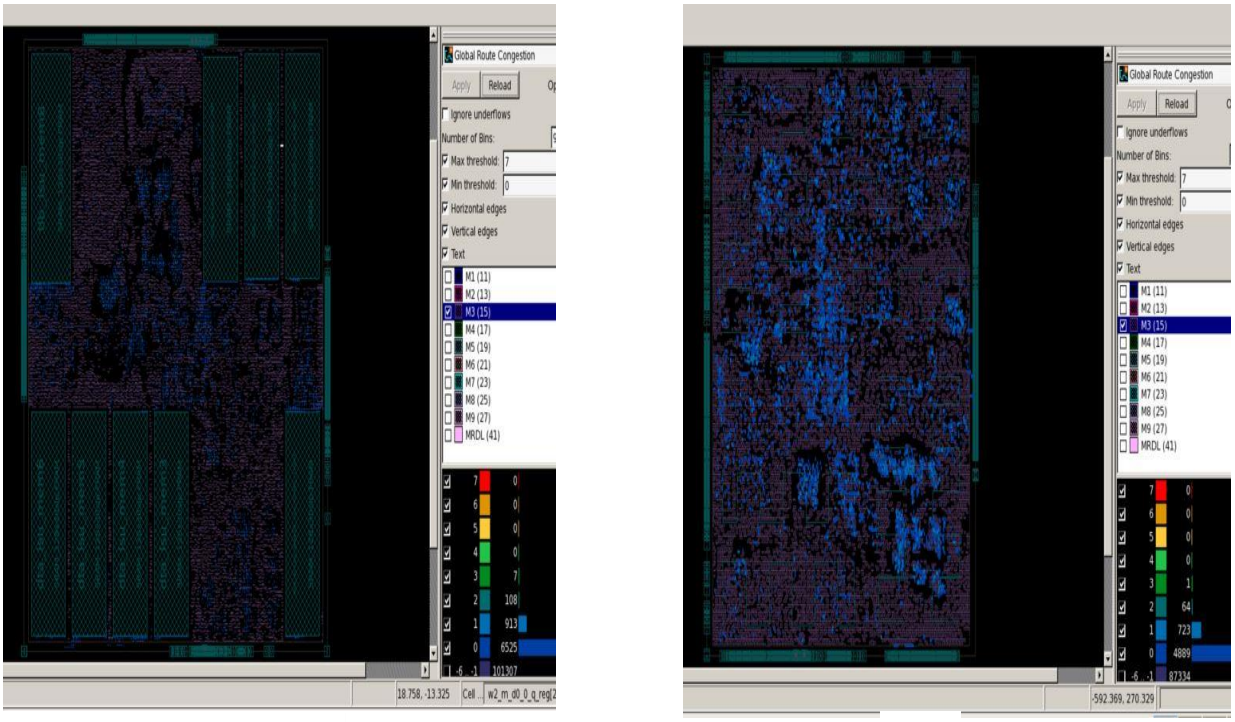
Parametri	**CTS 2D**	**CTS 3D**	**Variație obținută prin CTS 3D (%)**
Clock Tree Name	: "l2clk"	: "l2clk"	-
Scenario:	"func"	"func"	-
Clock Period:	3.00000	3.00000	-
Clock Tree root pin	: "l2clk"	: "l2clk"	-
Number of Levels	: 8	**: 9**	**−12,50%**
Number of Sinks	: 7729	: 7729	**0,00%**
Number of CT Buffers	: 274	: **204**	**25,55%**
Number of CTS added gates	: 0	: 0	**0,00%**
Number of Preexisting Gates	: 180	: 180	**0,00%**
Number of Preexisting Buf/Inv	: 10	:9	**10,00%**
Total Number of Clock Cells	: 464	: **393**	**15,30%**
Total Area of CT Buffers	:1565.27307 µm	: **1383.56042 µm**	**11,61%**
Total Area of CT cells	:2057.03809 µm	: **1879.39172 µm**	**8,64%**
Max Global Skew	: 0.05655 ns	**: 0.04670 ns**	**17.4 %**

Scăderea numărului de buffere/inversoare folosite la CTS cu 25.5% în varianta 3D comparativ cu 2D și a numărului total de celule de clock cu 15.3 % arată beneficiile integrării tridimensionale comparativ cu integrarea clasică 2D. Optimizarea tridimensională a circuitului, reduce lungimile conexiunilor atât cele medii cât și cele maxime, necesare pentru a conecta componentele circuitului, reducând disiparea puterii și crescând performanța în același timp. Reducerea totală a ariei circuitului datorată suprapunerii blocurilor de memorii peste celulele standard se reflectă la nivelul arborelui de ceas în următoarele aspecte:

- ✓ Numărul de inversoare și buffere folosite la crearea arborelui de ceas scade cu 25.5%;
- ✓ Numărul total de celule de ceas s-a redus cu 15.3%;
- ✓ Aria totală a inversoarelor și buffere-lor a scăzut cu 11.6% ;
- ✓ Aria totală a celulelor de ceas s-a redus 8.6% ;
- ✓ Numărul de niveluri ale arboreluid e ceas a crescut cu 12.5%;
- ✓ Valoarea skew-ului este redusă cu 17.4%.

4.4 Rutarea conexiunilor

După implementarea arborelui de clock, s-a realizat rutarea circuitului, etapă în care conexiunile logice între componentele circuitului sunt implementate și fizic cu metalele din tehnologie. Tehnologia folosită pentru implementarea acestui circuit conține 10 metale de la M1 la RDL (ReDistribution Layer), RDL fiind metalul TOP, în general din aluminiu. S-au folosit pentru rutarea conexiunilor de semnal doar 6 metale din 10 pentru primul strat, ultimele fiind folosite pentru rutarea pe stratul al doilea unde M7 devine primul metal de rutare – pe M7 sunt pinii memoriilor de pe stratul al doilea, iar alimentarea lor se face de pe metalele M8 și M9. Un factor important în etapa de rutare îl reprezintă congestia pe fiecare metal. Congestia de rutare reprezintă raportul dintre numărul de track-uri de rutare disponibile și numărul de track-uri de rutare necesare pentru rutarea conexiunilor. În figura 4.14 (a) și 4.14 (b) sunt prezentate capturi de ecran realizate în cadrul mediului de proiectare ICC Synopsys și reprezintă congestiile de rutare pe metal 3 în cele 2 scenarii de proiectare a circuitului integrat: 2D și respectiv 3D.



(a) **(b)**

Fig. 4.14. Congestie pe M3 după rutare: (a) Rutare 2D, (b) Rutare 3D

Congestia la nivelul stratului Metal 3 în cele 2 scenarii 2D și 3D este pusă în evidență de culoarea albastru, care după cum se poate observa este mai accentuată și mai compactă în scenariul 3D datorită micșorării suprafeței pe care este implementat circuitul integrat. Această creștere a congestiei pe metal 3 este o consecință directă a reducerii ariei totale disponibile pentru rutare,

rezultat al suprapunerii blocurilor de memorie peste celulele logice standard. Datorită integrării pe verticală pe cele două straturi, aria totală a circuitului este minimizată iar densitatea metalelor la rutare este concentrată într-un spațiu mai compact, astfel evitându-se irosirea suprafeței de Siliciu neutilizată.

Rutarea interconexiunilor într-un circuit are două etape care sunt executate una dupa alta: rutarea globală, în care se elaborează un plan de rutare astfel încat fiecare segment dintr-o conexiune să fie alocat unei zone particulare de rutare și rutarea detaliată**,** care se aplică fiecărei zone particulare de rutare și unde fiecărui segment i se alocă o pistă particulară de rutare.

O congestie ridicată pe fiecare metal în parte, după cum se observă în figura 4.14 (b), scoate în evidență o folosire optimă a ariei circuitului și o rutare a interconexiunilor mai eficientă decât în scenariul 2D. În cele două rapoarte 2D respectiv 3D din tabelul 4.2 sunt extrase informațiile despre numărul total al conexiunilor din circuit, despre lungimea metalelor folosite la rutarea interconexiunilor de semnal și numărul contactelor folosite între metale, rapoarte generate după etapa de rutare detaliată. Lungimea conexiunilor rutate este raportată pentru fiecare metal în parte de la M1 până la M6. Lungimea totală a conexiunilor rutabile în scenariul 3D scade cu aproximativ 16% față de scenariul 2D.

Tabel 4.2 Raport conexiuni 2D vs conexiuni 3D

Parametri	**Rutare 2D**	**Rutare 3D**	**Reducere obținută prin rutarea 3D (%)**
Total Wire Length	1413384 **µm**	**1185951 µm**	**16.09%**
Total Number of Contacts	296263	**282697**	**4.58%**
Total Number of Wires	296115	**280907**	**5.14%**
Total Number of PtConns	43717	**42127**	**3.64%**
Total Number of Routable Wires	296115	**280907**	**5.14%**
Total Routable Wire Length	1413384 **µm**	**1185951 µm**	**16.09%**
Layer M1 :	132506 **µm**	**91947 µm**	**30.61%**
Layer M2 :	459813 **µm**	**379519 µm**	**17.46%**
Layer M3 :	367047 **µm**	**345682 µm**	**5.82%**
Layer M4 :	206906 **µm**	**157387 µm**	**23.93%**
Layer M5 :	141079 **µm**	**127957 µm**	**9.30%**
Layer M6 :	106032 **µm**	**83458 µm**	**21.29%**

Se poate observa scăderea lungimii totale a interconexiunilor rutate în cazul plasării și rutării 3D cu 16% comparativ cu 2D, precum și scăderea lungimii pe fiecare metal în parte

comparativ cu plasarea și rutarea 2D. De asemenea numărul de contacte folosite pentru trecerea de pe un metal pe altul, este mai mic în cazul plasării și rutării 3D.

Tabelul 4.3 prezintă raportul de arie generat de instrumentul IC Compiler de la Synopsys după finalizarea procesului de rutare a circuitului, pentru configurațiile 2D și 3D, precum și reducerea procentuală a parametrilor obținută prin implementarea arhitecturii tridimensionale raportat la configurația bidimensională. Aceste rapoarte includ informații detaliate referitoare la structura circuitului, cum ar fi numărul total de celule combinaționale și secvențiale utilizate în circuit, numărul de porturi de intrare și ieșire, precum și numărul total de conexiuni fizice între celule. De asemenea, este prezentată suprafața totală ocupată de celulele și conexiunile circuitului, atât înainte, cât și după aplicarea proceselor de optimizare și rutare, oferind o perspectivă comparativă asupra eficienței utilizării resurselor în cele două scenarii de integrare.

Tabel 4.3 Raport de arie după rutare 2D vs 3D

Parametri	**Rutare 2D**	**Rutare 3D**	**Reducere obținută prin rutarea 3D (%)**
Number of ports:	1672	1672	0,00%
Number of nets:	34757	**33645**	**3,20%**
Number of cells:	31710	**30603**	**3,49%**
Number of combinational cells:	23981	**22869**	**4,64%**
Number of sequential cells:	7729	7729	0,00%
Number of macros:	9	9	0,00%
Number of buf/inv:	7644	**6599**	**13,67%**
Number of references:	104	104	0,00%
Combinational area:	61909.987169 **µm**	**61754.959188 µm**	**0,25%**
Buf/Inv area:	17534.665541 **µm**	**17409.372403 µm**	**0,71%**
Noncombinational area:	1126091.000566 **µm**	**1126260.768732 µm**	−0,02%
Net Interconnect area:	**73964.251076 µm**	**73894.455072 µm**	**0,09%**

Comparativ cu implementarea 2D, integrarea 3D permite o distribuție mai eficientă a celulelor pe straturi multiple, ceea ce duce la o reducere a suprafeței totale ocupate de circuit. Această optimizare poate contribui la creșterea densității circuitului și la economisirea spațiului pe suprafața circuitului integrat. În implementarea 3D, separarea pe cele două straturi dintre celule

logice și memorii reduce lungimea traseelor dintre acestea și îmbunătățește performanța globală a circuitului. Numărul de buffere și inversoare utilizate la optimizare scade cu 13.67 % în scenariul 3D, iar acest lucru indică o îmbunătățire a conectivității și o reducere a necesității de amplificare a semnalelor, ceea duce la un consum mai redus de energie. De asemenea aria conexiunilor fizice este mai mică în circuitul 3D, iar acest lucru sugerează o reducere a lungimii totale a conexiunilor, ceea ce poate îmbunătăți atât viteza circuitului, cât și consumul energetic.

4.5 Impactul conexiunilor verticale asupra ariei în integrarea 3D

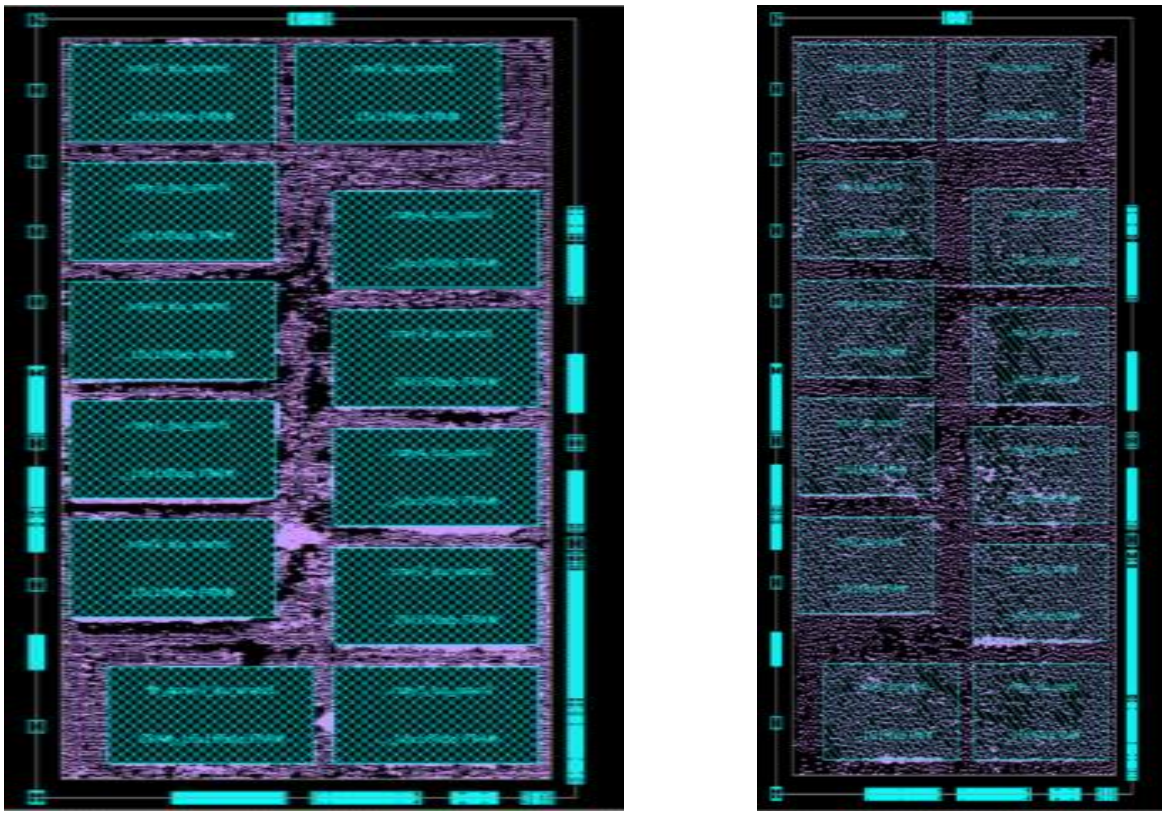
În acest subcapitol este implementată și analizată o variantă diferită de circuit integrat, care conține același bloc LSU, dar diferă blocurile de memorie de pe stratul 2. În acest nou scenariu pe stratul 2 s-au plasat 12 memorii, mai mici decât cele plasate în circuitul anterior, pastrând însă aceiași proporție de 50% din aria totală a circuitului [113]. Prin implementarea și testarea acestui circuit cu o configurație diferită de memorii pe stratul al 2 lea, s-a urmărit demonstrarea avantajelor integrării 3D în diferite scenarii.

Performanțele tot mai ridicate ale dispozitivelor mobile, împreună cu dimensiunile mici și consumul de putere redus, fac ca acest segment sa fie unul dintre cele mai importante pentru produsele semiconductoare. În configurațiile System on Chip clasice, unde integrarea se face bidimensional, funcțiile sunt toate implementate pe același strat de siliciu folosind același proces tehnologic. Practic, logica cuprinsă în circuit și care reprezintă aproximativ 25% din aria circuitului, împreună cu memoriile care reprezintă 50% din arie și restul componentelor cum ar fi logica analogică și pinii de intrare-ieșire sunt toate implementate în aceeași tehnologie [127]. Această abordare însă este limitată, din cauza faptului că toate componentele trebuie să respecte aceleași constrângeri tehnologice.

Pentru a răspunde cerințelor dispozitivelor mobile, este necesar ca cele 3 categorii de componente dintr-un System-on-Chip: logică, memorii și parte analogică să fie implementate în procesele tehnologice diferite, în funcție de necesitățile fiecărei categorii de componente. Această integrare eterogenă este posibilă folosind tehnologia monolitic 3D, unde fiecare strat poate fi implementat într-o tehnologie specifică și astfel optimizarea se poate face pentru fiecare categorie de componente, separat, în funcție de cerințe [127, 128]. Aceasta înseamnă ca partea logică, memoriile și partea analogică pot fi implementate și optimizate în procese tehnologice diferite, iar în final integrate într-o arhitectură tridimensională. Această abordare nu doar că reduce semnificativ costurile de producție, dar permite și îmbunătățirea performanței prin reducerea latențelor de comunicare între straturi și a consumului de energie.

Știind că blocurile de memorie ocupă cel mai mare procent din aria unui System-on-Chip, atenția în procesul de proiectare a unui circuit integrat tridimensional se concentrează pe partiționarea inteligentă a memoriilor și a celorlalte componente pe straturi diferite. Această separare a memoriilor și logicii pe straturi diferite, permite optimizarea fiecărui strat separat obținându-se astfel o arie mai mică, viteză de transmitere a datelor mai mare și un consum de putere mai redus ceea ce înseamnă un circuit mai performant din punct de vedere al ariei, vitezei și consumului de putere. Viteza de transmitere a datelor mai mare și consumul redus de putere este datorat și reducerii semnificative a lungimii interconexiunilor rezultată în urma integrării pe verticală.

În figura 4.15 (a) este prezentat scenariul de plasare 2D a celor 12 blocuri de memorie și a celulelor logice în jurul memoriilor, unde atât memoriile cât și celulele logice sunt plasate pe un singur strat. În figura 4.15 (b) [113] se poate observa arhitectura de plasare tridimensională unde memoriile sunt pe stratul 2, iar celulele logice sunt plasate pe primul strat. Pentru ambele scenarii s-a utilizat aceeași arie de plasare a circuitului integrat și se poate observa diferența de congestie a ariei în care sunt plasate celBulele logice în cele 2 scenarii.



(a) **(b)**

Fig. 4.15. Plasare memorii și celule logice: (a) Plasare 2D, (b) Plasare 3D

În scenariul 2D, datorită suprafeței de plasare redusă rămasă după plasarea și fixarea blocurilor de memorii, celulele logice sunt plasate într-o arie care pune probleme de rutare și de respectare a constrângerilor de timing. Optimizarea celulelor logice pentru atingerea obiectivelor de timing este imposibilă într-un astfel de scenariu în care celulele sunt plasate una lângă alta (se pot observa în figura 4.15 locurile marcate cu mov intens unde celulele logice sunt plasate, în funcție de constrângerile logice, una lângă alta.) În schimb, în situația plasării tridimensionale, suprafața disponibilă pentru plasarea logicii pe primul strat este suficientă pentru a nu pune probleme de rutare, iar constrângerile de timp pot fi îndeplinite datorită ariei disponibile după plasarea memoriilor pe stratul al doilea.

Diferența de arie între cele 2 scenarii de plasare 2D și plasare 3D se poate observa în tabelul 4.4 care cuprinde rapoartele de arie generate cu IC Compiler după plasarea celulelor logice și a blocurilor de memorii. Aria necesară plasării celulelor logice în scenariul 2D este mai mică cu aproximativ 50% în scenariul 2D față de 3D din cauza plasării memoriilor pe același strat și poate fi găsită în tabel ca raport al ariei blocate după plasarea memoriilor.

Tabel 4.4 Diferență de arie între implementare 2D și 3D [113]

Parametri	**Arie 2D**	**Arie 3D**	**3D/2D**
Număr celule combinaționale	26616	**25415**	0.954877
Arie celule combinaționale	69620	**65883**	0.946323
Număr celule secvențiale	8298	8298	1
Arie celule secvențiale	54730	54742	1.000219
Număr buffere	4283	**3465**	0.809012
Arie buffere	12108	**9753**	0.8055
Număr inversoare	4481	**3976**	0.887302
Arie inversoare	8654	**7220**	0.834296
Număr buffere și inversoare în arborele de clock	282	**244**	0.865248
Număr de conexiuni	37876	**36647**	0.967552
Lungimea totală a conexiunilor	1523443	**1218919**	0.800108
Arie blocată	**462400**	**230050**	**0.497513**

Această abordare a plasării tridimensionale exploatează avantajele integrării pe mai multe straturi, permițând o utilizare mai eficientă a suprafeței disponibile, reducând numărul de conexiuni precum și lungimea totală a conexiunilor, îmbunătățind performanța circuitului prin scăderea latențelor de propagare a semnalelor și a consumului de putere asociat transmisiei între componente. Cu cât nivelul de interconectivitate între straturi este mai mare, cu atât lungimea

totală a conexiunilor este mai mică, iar asta se traduce prin rezistențe și capacități mai mici, care conduc la reducerea întârzierilor de propagare și a consumului dinamic de energie.

Prin urmare, integrarea monolitică 3D reprezintă o soluție avansată pentru a răspunde cerințelor tot mai mari ale pieței dispozitivelor mobile și oferă un echilibru optim între performanță, arie, consum de putere și cost.

4.6 Metodă specifică de implementare a unui circuit integrat 3D folosind instrumente 2D

Acest subcapitol prezintă o metodă de integrare tridimensională utilizând un instrument specific destinat structurilor bidimensionale, având în vedere o strategie de partiționare a memoriei (a fiecărei memorii) pe două straturi distincte. Conceptul fundamental al acestei metode constă în separarea funcțională a componentelor memoriei: unitatea logică, reprezentată de decodor și de multiplexor, este păstrată pe primul strat, alături de celulele logice necesare procesării, în timp ce celulele de memorie propriu-zise sunt amplasate pe cel de-al doilea strat. Această abordare permite optimizarea spațiului utilizat, îmbunătățirea performanței circuitului și reducerea consumului energetic, aspecte esențiale în proiectarea sistemelor integrate moderne.

În circuitele moderne, o mare parte din aria logică este dedicată blocurilor de memorie încorporate, SRAM. De multe ori, într-un SOC obișnuit aria ocupată de blocurile de memorii este mai mare de 50% din aria totală a sistemului. Aceste tipuri de memorii de capacități mari, deși nu ajung la dimensiunile cerute de DRAM sau NVRAM, ar putea beneficia de o tehnologie CMOS optimizată. În general, memoriile încorporate nu necesită 10 până la 12 metale pentru conectare, sau tranzistoare rapide care au un consum mare, dar care sunt necesare în logica cu performanțe ridicate.

Tehnologia monolitica 3D a cărei densitate de conectare pe verticală se apropie de cea pe orizontală, deschide noi oportunități în ceea ce privește optimizarea proiectării. Proiectarea logică poate fi împărțită în straturi 3D specializate, fiecare strat având proprii parametri de proiectare, prin urmare propria dimensiune și respectiv propriul cost de fabricare. De exemplu, partea (core area) logică din circuit poate fi împărțită într-un strat de 12 metale (o tehnologie de vârf), un strat de 5,6 metale (o tehnologie mai puțin performantă și cost reduc) și un strat pentru parte necritică din circuit folosind o tehnologie mai ieftină de generație n-1 sau n-2. Datorită procentului mare de arie ocupată în circuit de memoriile încorporate, în mod natural atenția va fi direcționată asupra partiționării pe straturi distincte a logicii și a memoriilor.

Blocurile de memorii sunt alcătuite dintr-o arie de celule de memorare înconjurată de un decodor pentru selecția celulelor, a cărui suprafață poate fi în formă de tip L, T, H, E, U etc. Se pot distinge două moduri de partiționare 3D a memoriilor: varianta A, unde întreaga memorie se

plasează pe un strat diferit de cel al logicii și varianta B unde aria celulelor la nivel de bit se plasează pe stratul de memorie, în timp ce partea de selecție, decodorul și multiplexorul rămân plasate pe stratul de plasare al logicii. Decizia cu privire la varianta ce poate fi utilizată, depinde de tehnologia folosită pentru decodor. Dacă decodorul folosește tranzistoare rapide, este indicată varianta B de partiționare, dacă nu, varianta A reprezintă o alegere mai bună. În plus,,varianta B prezintă un avantaj semnificativ: plasarea celulelor de memorie pe un strat separat simplifică proiectarea, îmbunătățește yieldul și micșorează costul. În anumite cazuri, în cadrul variantei B se poate folosi o strategie mixtă, unde decizia pentru separarea sau menținerea decodoarelor și a celulelor de memorie pe același strat, poate fi luată pentru fiecare memorie în parte. O astfel de strategie oferă o flexibilitate mărită pentru controlul suprafețelor alocate memoriilor și restul logicii funcționale.

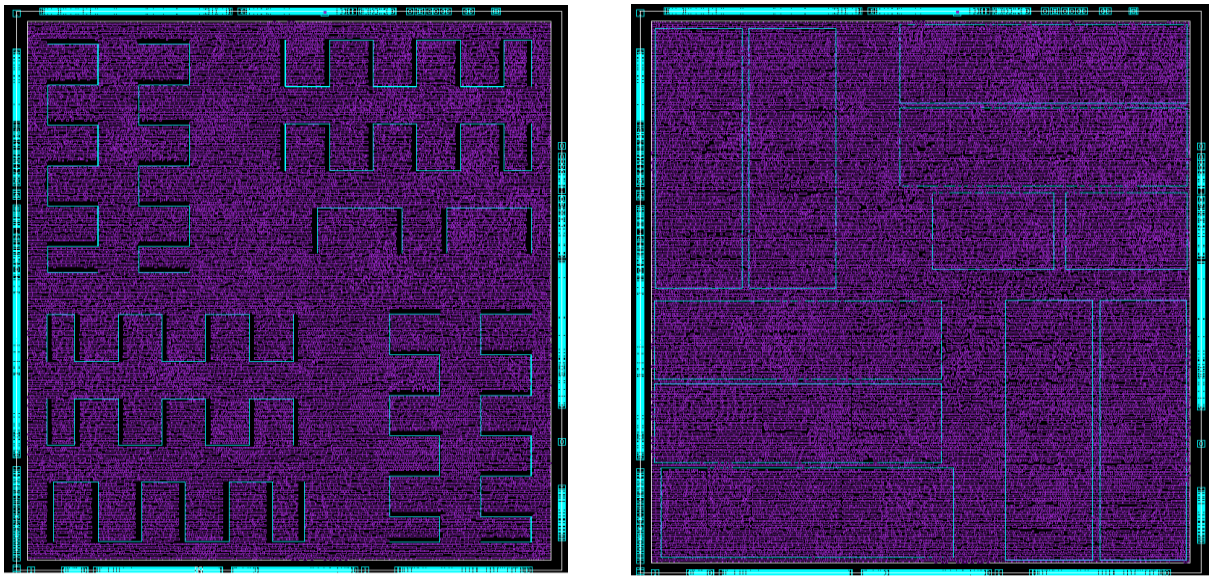
O provocare importantă în cazul strategiei de plasare a memoriilor clasice este dată de faptul că, deși ocupă o arie mică pe stratul 1 (conectarea la pinii memoriei în varianta A, decodoarele și pinii în varianta B), memoriile nu ar trebui să se suprapună între straturi. De exemplu, pentru un decodor în forma de L al unei anumite memorii, plasat pe stratul de logică, pe suprafața dreptunghiulară care include L-ul nu trebuie suprapusă o altă memorie. Însă în varianta B, decodoarele nu mai trebuie să păstreze formele clasice de L, T, H, E, U etc, ele pot ocupa întreg spațiul de sub celulele de memorie optimizând astfel timpii de acces la aceasta.

Primul pas îl reprezintă planificarea: luând în considerare alegerea variantei A sau B, se ia o decizie care dintre memoriile mici ar trebui mutate pe stratul 2 pentru a echilibra cele doua straturi. Procedura de lucru utilizând IC Compiler de la Synopsys este următoarea:

a. Pentru fiecare memorie care va fi implementată pe stratul 2 se generează o reprezentare specială a memoriei. Reprezentarea va include pinii memoriei pentru varianta A sau pinii zonei bit-celurilor pentru varianta B. În plus se va crea o obstrucție pe cel mai înalt metal din tehnologia utilizată, obstrucție care trebuie sa fie în formă de serpentină cu un pas de meandru mai mic decât cea mai mică memorie din circuit. Serpentina va permite instrumentului de plasare 2D să plaseze celule logice în spațiile goale, dar nu va permite să fie plasate memorii.
b. Instrumentul de plasare ICC de la Synopsys plasează logica și memoriile mici în mod obișnuit. Tot în această etapă sunt plasate și blocurile de memorii ca în figura 4.16. (a)
c. Folosind locațiile blocurilor de memorii de la pasul b, se face plasarea detaliată ca în figura 4.16. (b).
d. După plasarea detaliată, blocurile de memorii se afla plasate pe stratul 2 conform pasului b. Legătura dintre straturi este implementată în funcție de alegerea variantei A

sau B. Trebuie observat că, din punct de vedere al legăturii dintre straturi, varianta B este mai solicitantă. Ea va necesita via între straturi pentru fiecare linie și coloană. Această metodă este practică doar în situația în care stratul 2 este sub stratul 1, caz în care rutarea nu este blocată.

După implementarea celor 4 pași descriși anterior, fiecare strat parcurge în mod independent toți pașii obișnuiți din fluxul de lucru al implementării circuitelor integrate 2D: implementarea arborelui de clock, rutarea conexiunilor, optimizările necesare la fiecare pas, iar în final verificările necesare conform constrângerilor.



(a) **(b)**

Fig. 4.16. Plasarea blocurilor de memorie: (a) Obstrucție de serpentină a memoriilor, (b) Obstrucție perferică a memoriilor

Această metodă specifică descrisă mai sus funcționează optim în cazul în care circuitul are un raport 1:1 între logică și memorii. Nu toate circuitele reale se încadrează într-o astfel de categorie. În această categorie de 50-50 diferența de arie dintre variantele A și B se potrivește. Decodorul poate ocupa de la 15% la 40% din aria blocurilor de memorie, în funcție de configurația memoriei. Dacă raportul memorie/logică este mai mare decât 1, înlocuirea variantei A cu varianta B va ajuta raportul memorie/logica să fie mai aproape de 1. Dacă această metodă se dovedește insuficientă, memoriile mai critice pot fi plasate pe stratul 1 pentru a echilibra raportul memorie/logică și mai mult. În cazul contrar, când raportul memorie/logică este mai mic decât 1, se pot alege căile de timp mai rapide (estimare de timp făcută pe plasare 2D) și se mută pe stratul 2 care este mai puțin performant decât stratul 1.

Fluxul de lucru descris în acest subcapitol a fost implementat folosind Synopsys Design Compiler, Synopsys IC Compiler și biblioteci ajustate manual. Rezultatele obținute dovedesc viabilitatea acestui concept de partiționare în logică și memorii așa cum este prezentat în figura 4.16. (a) respectiv figura 4.16. (b).

4.7 Concluzii la capitolul 4

1. Trecerea de la metodologia de proiectare a circuitelor integrate în tehnologie 2D la proiectare în tehnologie 3D reprezintă o provocare majoră în microelectronică, dar și o oportunitate pentru inovare. Utilizarea unui flux de lucru experimental de proiectare 3D care folosește instrumente 2D poate constitui un punct de plecare pentru adaptarea și dezvoltarea de noi tehnologii care să faciliteze progresul în dezvoltarea circuitelor integrate tridimensionale.

2. Întregul flux de lucru de Backend al implementării circuitului integrat tridimensional a fost executat folosind instrumente standard 2D EDA cu modificări adecvate ale fișierelor tehnologice. Acest lucru dovedește ca proiectarea 3D a circuitelor integrate se poate face folosind instrumentele standard actuale 2D la care să se adauge anumite caracterisitici care să modeleze și a treia dimensiune.

3. Capacitatea de a implementa un circuit integrat 3D utilizând instrumentele tradiționale 2D sugerează că o tranziție treptată de la metodologiile de proiectare 2D la metodologiile de proiectare 3D este fezabilă. Această abordare reduce bariera de intrare în domeniu pentru companiile și instituțiile de cercetare care doresc să exploreze integrarea 3D, cu investiții minime în lanțuri de instrumente noi și costisitoare. Costul instrumentelor 3D EDA de ultimă generație poate varia de la sute de mii la milioane de dolari anual, iar utilizarea instrumentelor 2D reduce semnificativ aceste cheltuieli, permițând cercetarea și dezvoltarea metodologiilor pentru implementarea 3D.

4. Cercetătorii și inginerii familiarizați **cu flux-urile de implementare** tradițională 2D pot folosi această metodologie descrisă în acest capitol pentru a încorpora treptat principiile 3D fluxurilor de lucru, fără a fi nevoie să învețe o suită complet nouă de instrumente într-un timp scurt.

5. Capacitatea de a adapta metodologiile existente pentru proiectarea circuitelor integrate 3D permite reutilizarea infrastructurii EDA deja disponibile, reducând riscurile asociate adoptării unor noi instrumente și facilitând tranziția către tehnologiile 3D pentru companiile care ezită să investească în fluxuri de proiectare complet noi.

6. Succesul acestei abordări demonstrează că soluțiile de instrumente EDA hibride ar putea reduce decalajul dintre proiectarea circuitelor integrate 2D și 3D, încurajând potențialii furnizori

EDA să abordeze dezvoltarea de capabilități incrementale în cadrul instrumentelor 2D existente, mai degrabă decât să creeze lanțuri de instrumente 3D complet separate. Acest lucru ar putea duce la dezvoltarea de instrumente EDA hibride 2D/3D rentabile, facând proiectarea 3D IC mai accesibilă în diferite sectoare, de la mediul academic la proiectarea semiconductorilor comerciali.

7. Metodologia descrisă demonstrează că circuitele integrate 3D monolitic pot fi proiectate folosind instrumente convenționale 2D EDA, facilitând companiilor crearea de prototipuri și tranziția către integrarea monolitică fără a investi în software costisitor specific 3D. Această abordare permite validarea rapidă a conceptelor 3D monolitice, asigurându-se testarea diferitelor strategii de stivuire înainte de a trece la procesul de fabricare.

8. Integrarea 3D monolitică a logicii și a memoriilor, realizată prin adaptarea fluxului 2D, a condus la reducerea ariei totale cu 50%. Lungimea totală a conexiunilor rutate a scăzut cu aproximativ 20% în scenariul 3D comparativ cu 2D. Această reducere a lungimii interconectărilor a determinat o scădere cu 13.5% a numărului total de buffere**,** datorită necesității mai reduse de compensare a întârzierilor de propagare pe trasee mai scurte. Aria blocată pentru plasarea celulelor din circuit scade cu 50% în scenariu 3D.

9. Abordarea plasării blocurilor de memorie peste celulele logice standard este potrivită pentru circuitele 3D monolitice, deoarece optimizează timpul de acces la date, densitatea de interconectare și planificarea suprafeței în care este plasat întregul circuit. Spre deosebire de circuitele 3D bazate pe tehnologia TSV, stivuirea monolitică evită dimensiunile mari, permițând o integrare mai fină a memoriilor și a straturilor logice.

10. Proiectarea circuitului utilizând tehnologia monolitic 3D duce la creșterea performanțelor circuitului integrat final. Distribuția uniformă a rețelei de ceas în scenariul 3D pe întreaga arie a circuitului (datorită eliberării suprafeței) comparativ cu scenariul 2D a condus la:

- ✓ Numărul de inversoare și buffere folosite la crearea arborelui de ceas a scăzut cu 25.5%;
- ✓ Numărul total de celule de ceas s-a redus cu 15.3%;
- ✓ Aria totală a inversoarelor și buffere-lor a scăzut cu 11.6% ;
- ✓ Aria totală a celulelor de ceas s-a redus 9% ;
- ✓ Numărul de niveluri ale arborelui de ceas a crescut cu 12.5% contribuind la o distribuție mai uniforma a semnalului de ceas;
- ✓ Valoarea skew-ului este redusă cu 17.4%.

11. Integrarea 3D monolitic conduce la reducerea costurilor circuitului integrat final. Asta se datorează diferitelor scenarii de fabricare, în care tehnologii (linii) vechi de fabricație (e.g. 28nm) se pot folosi împreună cu noduri avansate (de ex. 7nm). De asemenea utilizarea

tehnologiilor mixte: tehnologii analogice (BiCMOS), împreună cu tehnologii digitale (standard CMOS) și cu tehnologii specifice memoriilor (memorii FLASH), constituie un scenariu viabil.

12. Această nouă abordare a optimizării costurilor în implementarea circuitelor care conțin multe memorii, în tehnologie 3D, oferă o mare flexibilitate în realizarea circuitelor cu gamă largă de modificare a raportului memorie/logică. Pentru implementarea fluxului de lucru s-au folosit instrumentele 2D care există pe piață la ora actuală și s-a demonstrat astfel suficientă flexibilitate pentru a îndeplini constrângerile legate de implementarea 3D.

5. PERSPECTIVE ÎN DEZVOLTAREA STRUCTURILOR DE CALCUL AI ÎN TEHNOLOGIA MONOLITIC 3D

Modelarea unor structuri monolitice 3D ce încorporează pe același substrat de Siliciu, prin dispunere multistrat, nuclee de procesare și unități cubice de memorie duce la creșterea capacității de procesare și la consum energetic redus. Această tehnologie este potrivită pentru conceperea "AI edge computing processors" cu aplicabilitate în analiza de imagini cu algoritmi AI, dezvoltarea de senzori inteligenți (self-decision intelligent sensors) sau a unor dispozitive ce încorporează PUF (Physiscal Unclonable Function). În figura 5.1 poate fi observată schema bloc a unei arhitecturi 3D generice de calcul la nivel de margine.

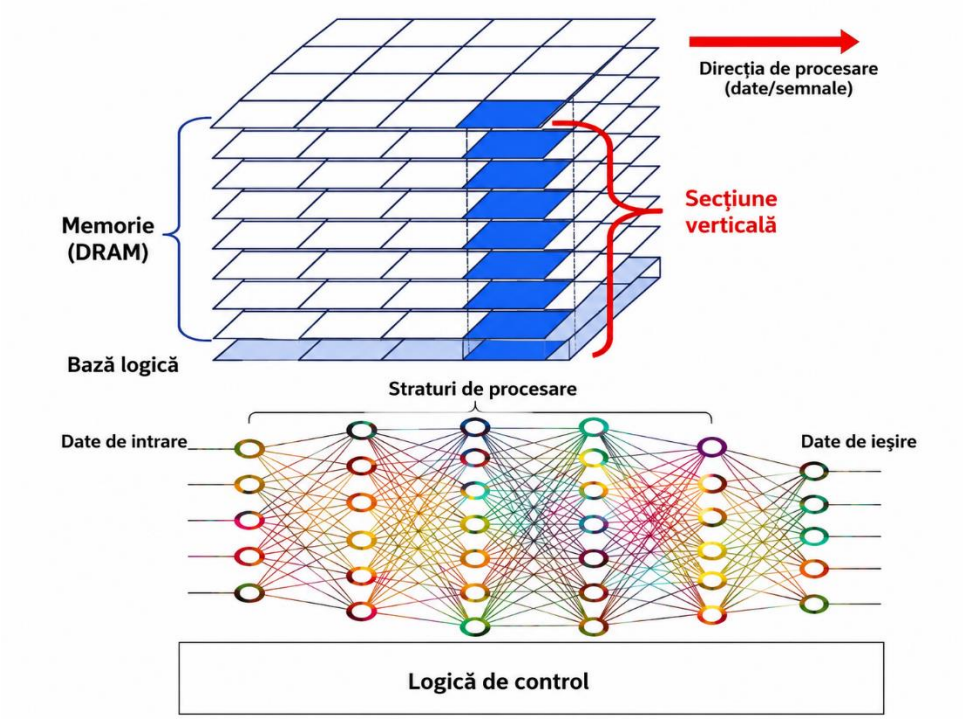


Fig. 5.1. Schema bloc a unei arhitecturi 3D generice de calcul la nivel de edge

O rețea neuronală 3D monolitică (3D-NN) pentru aplicațiile de calcul de margine ar trebui să fie compactă, eficientă din punct de vedere energetic și capabilă de inferențe de mare viteză în timp ce este implementată pe dispozitive cu resurse limitate. În loc de circuitele integrate 2D tradiționale, acest sistem va folosi integrarea 3D monolitică, în care diferite straturi ale rețelei neuronale sunt stivuite vertical într-un singur circuit integrat. Aceasta oferă o serie de avantaje, printre care:

a) Interconexiuni mai scurte → Comunicare mai rapidă între straturi;
b) Consum redus de energie → Critic pentru aplicațiile de margine;
c) Densitate de calcul mai mare → Mai multă putere de procesare pe unitate de suprafață.

Integrarea 3D va fi implementată folosind matrice de bare transversale bazate pe ReRAM 3D pentru calcul în memorie și Logica 3D CMOS pentru procesare digitală. Pentru a obţine interconexiuni cu latenţă scăzută se va utiliza tehnologia Through-Silicon Vias (TSV) sau Monolithic Inter-tier Vias (MIV).

Brevetul CN110414676A dezvăluie un procesor de rețea neuronală care integrează o matrice de stocare longitudinală tridimensională (3D) cu un circuit de procesare neuronală [129]. În acest circuit, matricea de stocare 3D, care memorează greutăți sinaptice, este stivuită vertical deasupra circuitului de procesare neuronală. Această integrare verticală îmbunătățește puterea de calcul și capacitatea de stocare pe unitate de suprafață în comparație cu dispozitivele tradiționale bidimensionale, în care unitățile de memorie și de procesare sunt plasate una lângă alta. Stivuirea 3D reduce amprenta procesorului și minimizează distanța dintre memorie și unitățile de procesare, ceea ce duce la o performanță și eficiență îmbunătățite în calculele rețelei neuronale. În figura 5.2 este reprezentat circuitul 730 care reprezintă o componentă funcțională cheie a circuitului de calcul neuronal încorporat într-o arhitectură de procesor neuromorfic. Integrarea circuitului 730 sub straturile de memorie suprapuse, permite procesorului să efectueze operații de acumulare multiplă direct sub stocarea ponderilor, reducând semnificativ mișcarea datelor și consumul de energie.

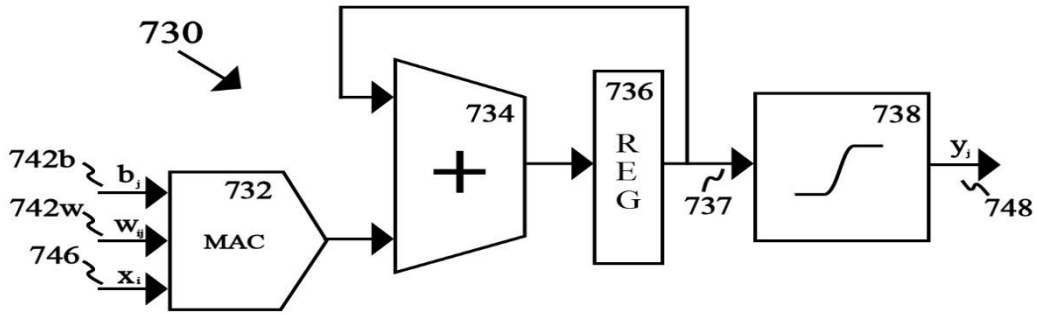


Fig. 5.2 Circuit de procesare neuronală [128]

Circuitul 730 se referă la circuitul de procesare neuronală integrat cu matricea de stocare longitudinală tridimensională (3D) [129]. Acest circuit este responsabil pentru executarea calculelor rețelei neuronale și este poziționat sub matricea de stocare 3D stivuită vertical. Circuitul minimizează distanța dintre memorie și unitățile de procesare, sporind eficiența de calcul și reducând amprenta totală a procesorului. Circuitul de procesare neuronală interacționează direct cu matricea de stocare, facilitând accesul rapid la date și procesarea esențială pentru operațiunile rețelei neuronale.

În figura 5.3 de mai jos este reprezentată o componentă de calcul ce cuprinde un multiplicator 152, un sumator 154, un registru 156 şi un circuit aferent funcţiei de activare 190. Multiplicatorul 152 multiplică datele de intrare xi cu ponderile wij. Apoi, sumatorul 154 și registrul 156 efectuează împreună adunări acumulative pentru rezultatele înmulțirii (adică wij*xi). Suma acumulată rezultată 158 este trecută prin circuitul cu funcţia de activare 190 pentru a genera datele de ieșire yj [129].

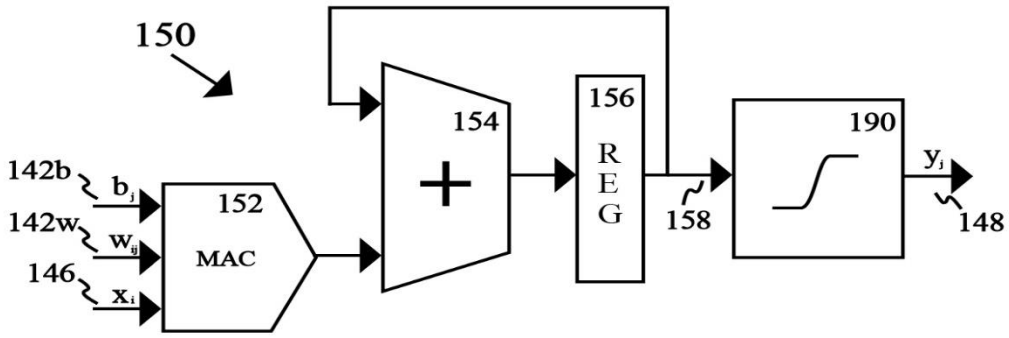


Fig. 5.3 Componentă de calcul neuronal [129]

Un circuit de rețea neuronală poate fi atât secvențial, cât și combinațional, în funcție de arhitectura și funcționalitatea pentru care este proiectat.

Circuit combinațional: în cazul în care rețeaua neuronală realizează exclusiv operații de tip feedforward, iar ieșirea depinde doar de valorile curente ale intrărilor, aceasta poate fi modelată ca un circuit combinațional. Nu este implicată nicio memorie sau buclă de feedback.

Circuit secvențial: în cazul în care rețeaua neuronală include conexiuni recurente sau mecanisme de memorare a stărilor anterioare, precum în rețelele neuronale recurente (RNN) sau în rețelele Long Short-Term Memory (LSTM), comportamentul acesteia este echivalent cu cel al unui circuit secvențial. Acest lucru se datorează faptului că ieșirea depinde nu doar de intrarea curentă, ci și de stările sau datele anterioare.

În rețelele neuronale, Weight Stationary (WS) [130] se referă la o strategie de flux de date utilizată în acceleratoarele hardware (cum ar fi cele pentru “deep learning”) în care ponderile rămân fixe (staționare) în elementele de procesare (PE), în timp ce activările de intrare și sumele parțiale sunt transmise prin hardware.

Lucrarea intitulată „A New Dataflow Implementation to Improve Energy Efficiency of Monolithic 3D Systolic Arrays” introduce WS-MONO3D, o implementare inovatoare a fluxului de date cu greutate staționară (WS) concepută pentru rețele sistolice monolitice 3D (MONO3D). Această abordare folosește mai multe straturi RAM rezistive (RRAM) și SRAM, interconectate

prin interconexiuni verticale de înaltă densitate, pentru a îmbunătăți eficiența mișcării datelor în timpul inferenței rețelei neuronale profunde (DNN) și poate fi observată în figura 5.4 [131]. Prin eliminarea ciclurilor de intrare și de transmitere a greutății, WS-MONO3D realizează o îmbunătățire cu până la 40% a produsului de întârziere a energiei (EDP) în comparație cu implementările tradiționale 2D WS. În plus, datorită structurii sale cu mai multe niveluri, oferă o creștere de zece ori a inferenței pe secundă pe watt pe amprentă. Studiul subliniază, de asemenea, importanța considerațiilor termice, menționând că variațiile de temperatură pot afecta beneficiile eficienței energetice ale WS-MONO3D.

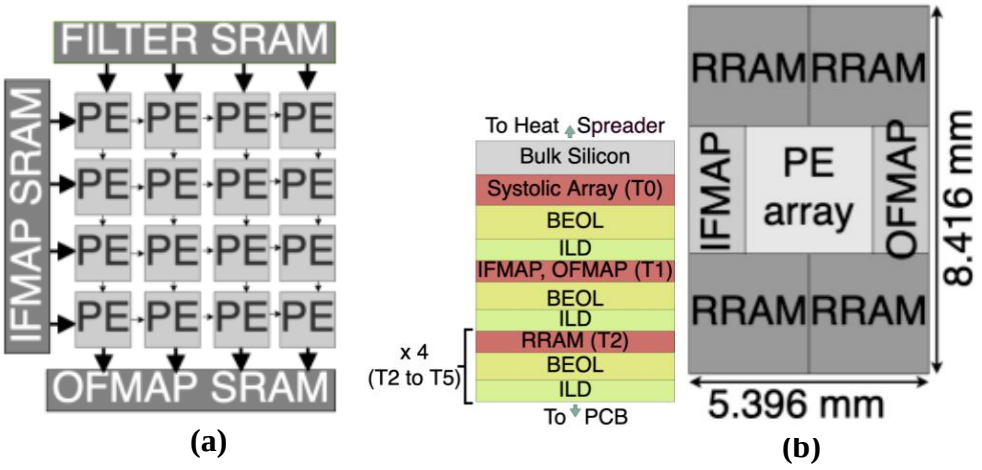


Fig. 5.4 Matrice sistolica 4x4 (a); stivă de cipuri Mono3D cu 6 niveluri flip-chip cu 4 niveluri RRAM pentru stocarea greutăților (b); Vedere de sus a omologului 2D [131]

În acest capitol au fost parcurși pașii de sinteză pentru o structură neuronală 3D ce este descrisă în limbaj VHDL. Pentru început s-au parcurs pașii de sinteză specifici unui circuit FPGA prin utilizarea programului VIVADO de la AMD. În cele ce urmează este exemplificat un cod sursă VHDL pentru un neuron simplu cu două intrări:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use ieee.Numeric_std.all;

entity Perceptron_OR is
   Port (
      x1, x2 : in STD_LOGIC;  -- Two input neurons
      clk    : in STD_LOGIC;  -- Clock signal
      y      : out STD_LOGIC  -- Output neuron
   );
end Perceptron_OR;

architecture Behavioral of Perceptron_OR is
```

```
    signal weight1 : integer := 1;  -- Initial weight for x1
    signal weight2 : integer := 1;  -- Initial weight for x2
    signal bias    : integer := -1; -- Bias term
    signal sum     : integer := 0;  -- Weighted sum

    signal x1b, x2b : std_logic_vector (1 downto 0);

begin

    x1b <= '0'&x1;
    x2b <= '0'&x2;

    process (clk)
    begin
      if rising_edge(clk) then
        -- calculul sumei ponderilor
        sum <= (weight1 * to_integer(unsigned(x1b))) + (weight2 *
to_integer(unsigned(x2b))) + bias;
        --sum <= to_integer(unsigned(a)) + to_integer(unsigned(b));
        -- functia de activare (de tip treapta)
        if sum >= 0 then
          y <= '1';
        else
          y <= '0';
        end if;
      end if;
    end process;

end Behavioral;
```

După încărcarea codului descris în VHDL în instrumentul VIVADO, s-a generat schematicul din figura 5.5.

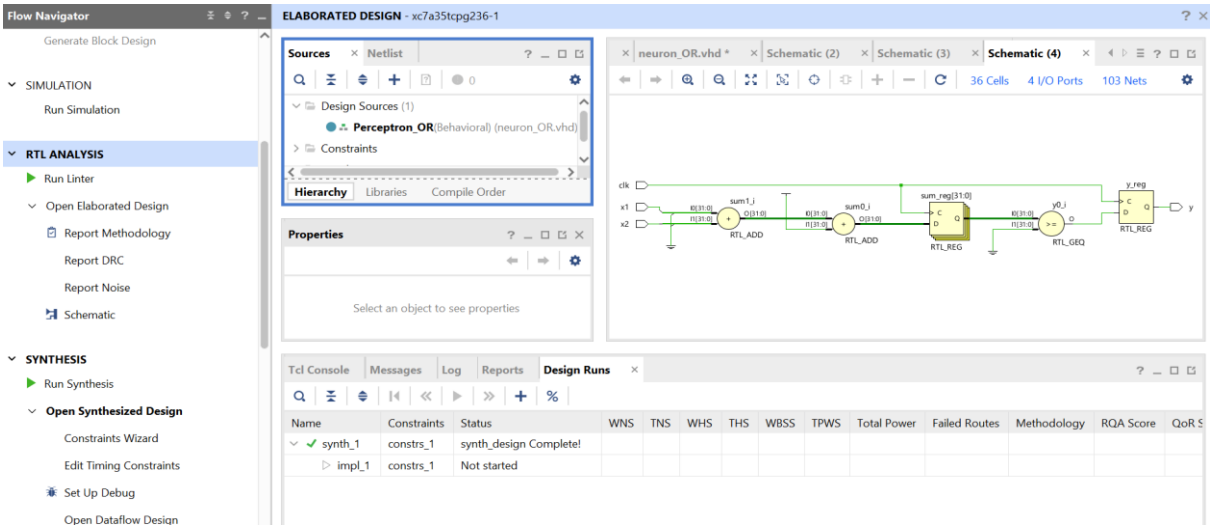


Fig. 5.5 Schema generată în urma etapei de analiză RTL cu instrumentul VIVADO

Un neuron într-o formă și mai simplificată este prezentant în figura 5.6. Semnalele de intrare și ponderile sunt aplicate unei porți AND. Rezultatul obținut este combinat cu semnalul de bias printr-o poartă SAU, a cărei ieșire reprezintă ieșirea neuronului. Această structură simulează comportamentul unui perceptron de bază, unde neuronul se activează dacă intrarea ponderată este mare sau dacă semnalul bias activează neuronul.

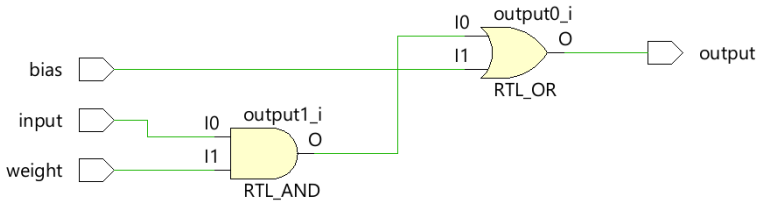


Fig. 5.6 Forma simplificată a unui neuron

Descrierea în VHDL a funcționalității unui neuron de bază este cea din figura 5.7 , cod ce ulterior a fost încărcat în VIVADO pentru a genera un schematic la nivel de poartă logică cu ajutorul instrumentului de sinteză încorporat în instrumentul de programare a FPGA-ului.

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.STD_LOGIC_ARITH.ALL;
4   use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6   entity Neuron is
7       Port (
8            input     : in  STD_LOGIC;
9            weight    : in  STD_LOGIC;
10           bias      : in  STD_LOGIC;
11           output    : out STD_LOGIC
12      );
13  end Neuron;
14
15  architecture Behavioral of Neuron is
16      signal weighted_sum : STD_LOGIC;
17  begin
18      process(input, weight, bias)
19      begin
20          if (input = '1' and weight = '1') or bias = '1' then
21              weighted_sum <= '1';
22          else
23              weighted_sum <= '0';
24          end if;
25      end process;
26      output <= weighted_sum;
27  end Behavioral;
```

Fig. 5.7 Codul VHDL pentru neuronul în structură simplificată

O formă generalizată a unui perceptron este prezentată în Anexa 4 sub forma unui cod sursă VHDL prin care se pot seta dimensiunile parametrilor de intrare. Schema generată cu instrumentul de sinteză VIVADO pentru perceptonul prezentat în Anexa 4 este cea din figura 5.8.

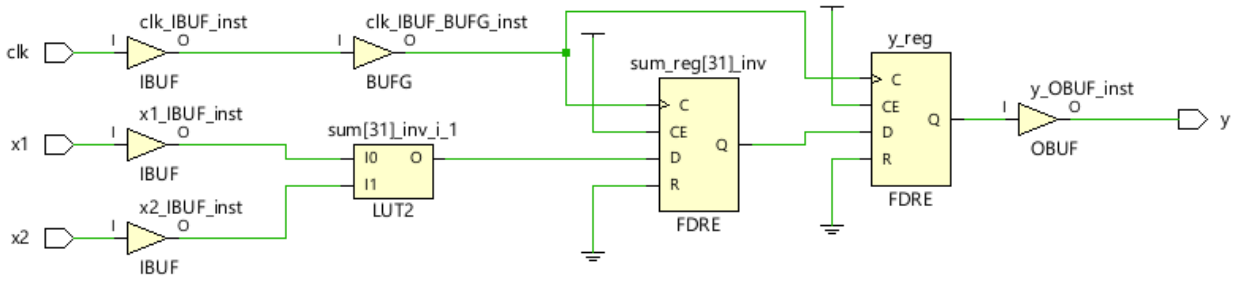


Fig. 5.8 Schemă generată de instrumentul VIVADO pentru perceptronul prezentat în Anexa 4

5.1 Rețea neuronală cubică cu trei straturi

O reţea neurală cubică este reprezentată în figura 5.9 fiind alcătuită din trei straturi, un strat de intrare, un strat ascuns și un strat de ieșire. S-au parcurs pașii de sinteză atât în tehnologia FPGA cât și în tehnologia ASIC 2D (pentru o structură monolitică 3D) și s-a realizat o comparație estimativă de performanțe.

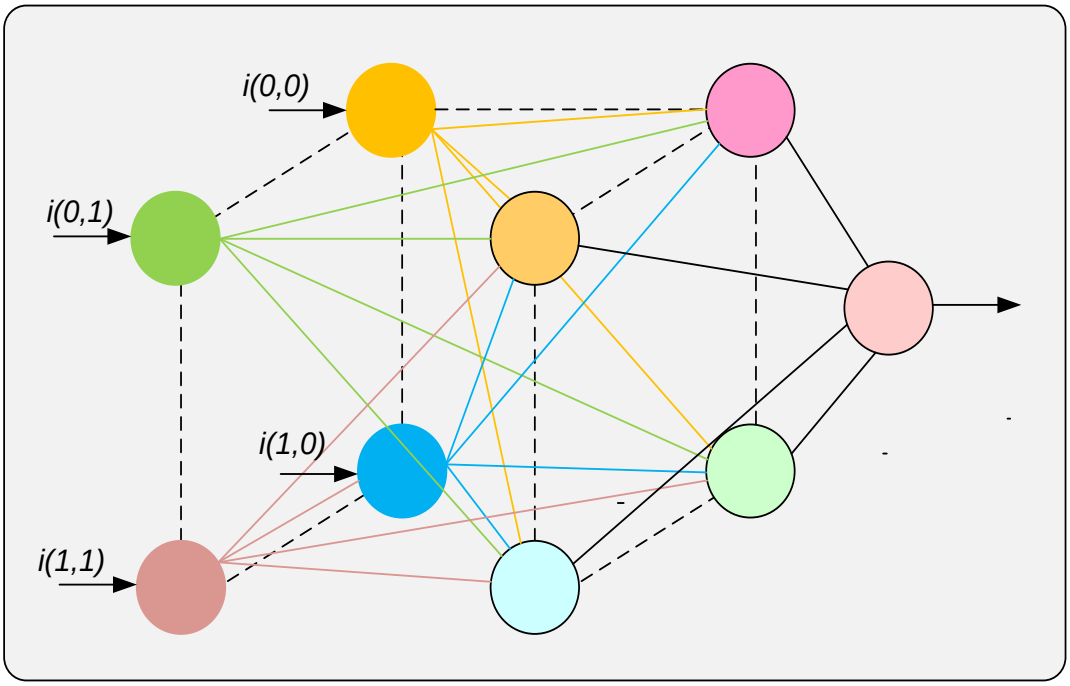


Fig. 5.9 Reprezentarea grafică a rețelei neuronale cu trei straturi

Structura reţelei este următoarea: stratul de intrare – 4 neuroni; stratul ascuns – 4 neuroni; stratul de ieșire: un singur neuron. Această arhitectură NN 4x4x1 o definim ca “neural grain” (NG) pentru care matricea ponderilor este:

$$W1 = \begin{bmatrix} w_{00} & w_{01} & w_{02} & w_{03} \\ w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \\ w_{30} & w_{31} & w_{32} & w_{33} \end{bmatrix}$$

Un “grăunte neuronal” poate fi văzut ca un bloc neuronal 3D modular autonom independent care are propriul său strat de intrare, unul sau mai multe straturi ascunse și un strat de ieșire, astfel încât mai multe NG-uri pot fi interconectate atât în paralel cât și în cascadă, această celulă neuronală făcând parte dintr-o reţea tridimensională mai vastă, destinată unor procesări locale prin altgoritmi AI dedicaţi extragerilor de caracteristici ori clasificărilor.

Grăunţii neuronali (NGs) pot fi combinaţi în arhitecturi mai complexe prin stivuire în straturi, ceea ce duce la procesări mai profunde (deeper processing), prin aranjare în paralel pentru algoritmi de luare de decizii sau prin conectarea în cascadă unde rezultatul unui nivel devine intrare pentru urmatorul având aplicabilitate în învățarea în mai multe etape (“multistage learning”).

Sunt întâlnite în literatură mai multe modele de arhitecturi NN independente cum ar fi “Modular Neural Network” [132].

Arhitecturile neuronale modulare (“Modular Neural Network”) au fost concepute ca și structuri neuronale independente, intermodulare, dedicate unor task-uri specifice, fiind inspirate din structurile biologice ale creierului [132]. Acestea sunt uneori numite „module neuronale”, „capsule” (în Rețelele Capsule) sau „elemente de procesare neuronală” (NPE). (“These are sometimes called "neural modules", "capsules" (in Capsule Networks), or "neural processing elements" (NPEs).”)

Arhitecturile neuromorfice (Neuromorfic Architectures) sunt structuri de calcul inspirate din modelele biologice și utilizează principiile funcţiilor neuronale [133]. Structurile neuromorfice integrează memoria și procesarea cu ajutorul neuronilor și sinapselor care imită impulsuri și astfel rezultă arhitecturi de procesare cu consum redus de energie [134, 135]. Din punct de vedere tehnologic aceste structuri pot fi combinaţii între elemente anorganice și organice sau analogice și digitale și se clasifică în funcţie de tipul de sinapse care pot să fie “spiking synapses” sau “memristive synapses”. În figura 5.10 se poate observa o comparație între structurile clasice de calcul (a,b,c) și o arhitectură neuromorfică.

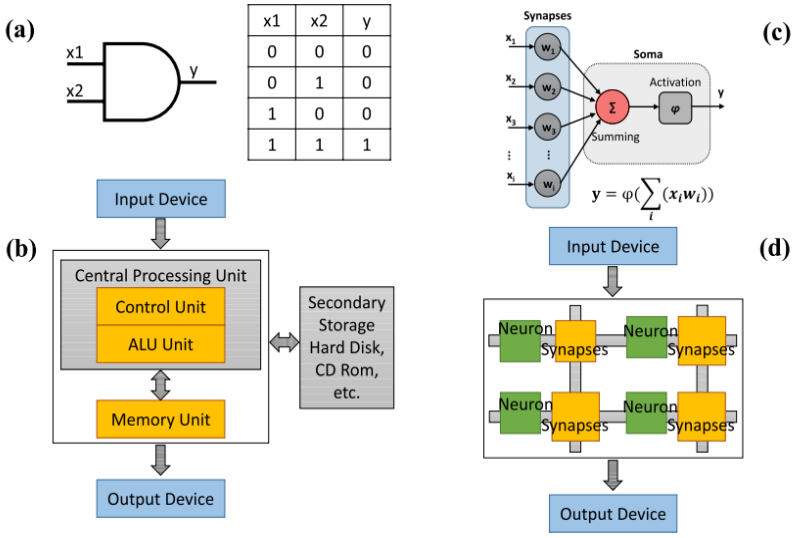


Fig. 5.10 Comparaţie între structuri "clasice" de calcul (a, b,c) și o arhitectură neuromorfică (d)

Grile de procesare / Blocuri de procesare în FPGAs: În proiectele de rețele neuronale bazate pe FPGA, este obișnuit să existe plăci de procesare sau PE (Elemente de procesare) care calculează ieșirile neuronale. Acestea pot fi împărțite în plăci pe dimensiuni, dar de obicei nu sunt organizate ca o unitate granulară 3D cu propria topologie stratificată. Un exemplu ce poate fi amintit este Versal AI Edge Series Gen 2 de la AMD-XILINX ce încorporează până la 144 de AI Engine-ML tiles.

Arhitecturi 3D personalizate (cercetare emergentă): Unele grupuri de cercetare lucrează la rețele neuronale 3D sau structuri de memorie 3D (DRAM stivuite cu nuclee neuronale). Există interes pentru CNN-uri 3D pentru video, imagistică medicală sau date spațiotemporale - dar acestea sunt despre date de intrare 3D, nu despre structura rețelei 3D în sine.

5.2 Implementarea NG 4x4x1 într-o structură FPGA

Conceptul de 3D FPGA este mai vechi, Xilinx a anunţat încă din 2012 disponibilitatea comercială pentru circuitul Virtex-7 H580T ce este prezentat ca "The world's first 3D heterogeneous all programmable product" – "Primul produs 3D heterogen complet programabil din lume" [136]. Configurarea 3D presupune realizarea unor straturi de siliciu suprapuse pe verticală ce sunt conectate prin via-uri de siliciu (TSV sau MIV), fiecare strat fiind optimizat independent, spre exemplu un strat fiind dedicat memoriilor, iar altul pentru logica de calcul sau de management. O structură monolitică 3D FPGA ca cea din figura 5.11 este prezentată de către compania MonolithIC 3D™ Inc. [137].

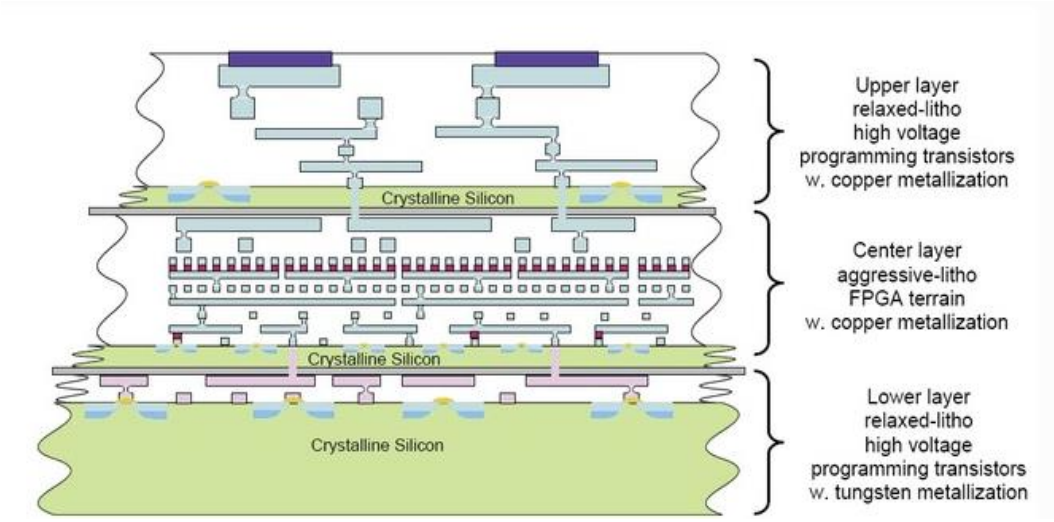


Fig. 5.11 FPGA 3D de înaltă densitate bazat pe antifuzibil [136]

Un grăunte neural în topologie NG 4x4x1 a fost descris în limbajul VHDL așa cum se poate observa în Anexa 5 și a fost implementat într-o structură FPGA a unui circuit ARTIX 7 de la AMD-XILINX. S-a optat inițial pentru implementarea pe FPGA, deoarece aceasta permite testarea și validarea funcționalității și performanței prototipului înainte de trecerea la implementarea finală în tehnologie ASIC [138]. Elementul de bază este un perceptron simplu ce preia un semnal de intrare pe opt biţi și un semnal de activare (bias) iar la ieșire generează un semnal binar. Acest comportament este echivalent cu o funcție de activare binară de tip treaptă, în care neuronul se „declanșează” (ieșirea este 1) dacă intrarea nu depășește pragul (bias) și rămâne dezactivat (0) în caz contrar. Logica ar putea fi inversată în funcție de intenția de proiectare, dar acest cod se „activează” în prezent când xi <= bias.

Yo = (xi <= bias) ? '1' : '0' ; (5.1)

A fost adoptat un model simplu de perceptron în care nu sunt specificate ponderile, matricea ponderilor fiind distribuită la nivel de arhitectură. Această abordare reflectă îndeaproape modul în care s-ar putea produce calculul în sistemele biologice sau în hardware-ul neuromorfic, unde „granulele neuronale” sau elementele de procesare dețin fiecare o parte din matricea de ponderi și contribuie local la calculul general.

Sistemele de numere reziduale, așa cum sunt analizate în [139], oferă avantaje semnificative în realizarea calculelor de mare viteză. Aceste principii sunt relevante în contextul arhitecturilor monolitice 3D care integrează unități de procesare a semnalelor, ce necesită reprezentări numerice eficiente.

Pornind de la perceptronul prezentat anterior, utilizat ca element de bază, a fost modelată prin descriere structurală arhitectura unui Neural Grain în topologia NG4x4x1. Aceasta dispune

de patru intrări corespunzătoare primului strat, alcătuit din patru neuroni, și de un strat ascuns format, la rândul său, din patru neuroni, rezultând o structură cubică. Ieșirile neuronilor din stratul ascuns constituie intrările neuronului din stratul de ieșire [140]. Ponderile sunt definite ca și constante pe 8 biti, pentru a simplifica arhitectura.

Acest cod (din ANEXA 5) prezintă o arhitectură clară de rețea neuronală orientată spre hardware de tip FPGA, urmând o filozofie a arhitecturii de tip grăunte neuronal: fiecare neuron funcționează independent cu ponderi locale și prag. Ponderile sunt constante, accentuând calculul feed-forward iar ieșirile sunt binare, imitând activarea (cum ar fi funcțiile prag).

Acest modul VHDL, numit neuron_NG, modelează un singur neuron artificial (perceptron) folosind logica digitală. Preia două intrări — o valoare xi pe 8 biți și o polarizare pe 8 biți — și emite un semnal binar Yo bazat pe o comparație între cele două intrări. Schema generată după implementarea codului VHDL pe placa FPGA este cea din figura 5.12 în care se pot observa cele 3 straturi de neuroni pe verticală, dintre care primele două straturi sunt formate din 4 neuroni, iar ultimul strat conține doar un singur neuron. Fiecare neuron din fiecare strat este alcătuit dintr-un multiplicator și o celulă de memorare așa cum se poate observa în figura 5.12.

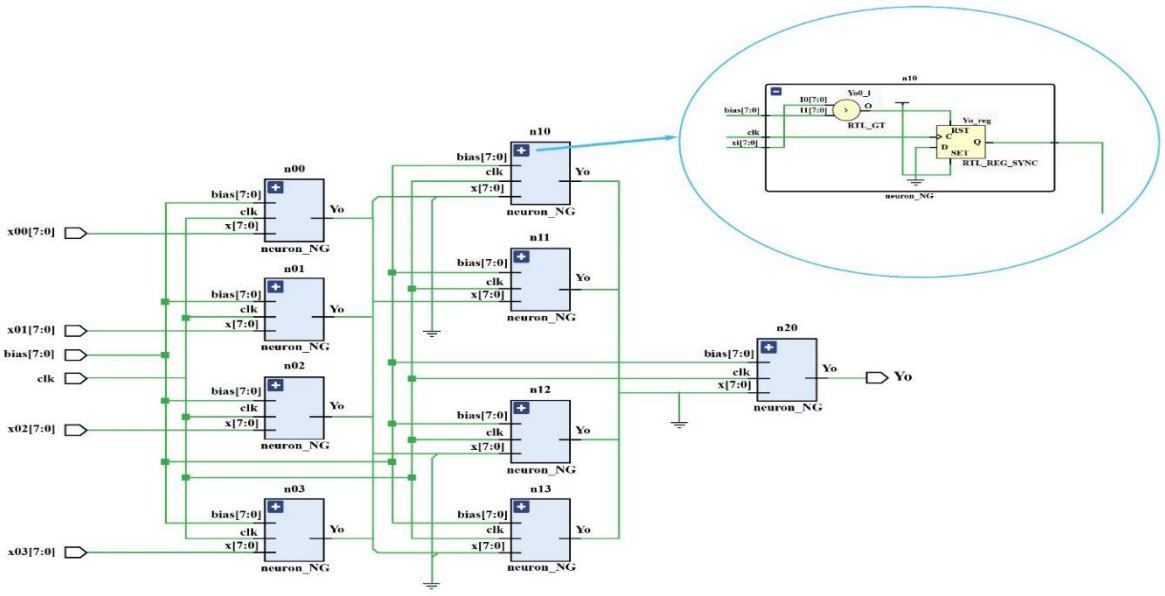


Fig. 5.12 Schema generată cu instrumentul VIVADO pentru rețeaua Grain Neuron

5.3 Implementarea NG 4x4x1 în tehnologie monolitic 3D

După parcurgerea flow-ului integrat în instrumentul VIVADO pentru implementarea pe placa FPGA, s-a parcurs și flow-ul de implementare a unui ASIC și anume sinteză, plasare și rutare

precum și generarea raporturilor de arie. Pentru parcurgerea procesului de lucru necesar ASIC-ului s-a folosit codul VHDL din ANEXA 5, care a fost utilizat de asemenenea la parcurgerea flow-ului în scenariul FPGA. Tehnologia utilizată pentru parcurgerea procesului de lucru corespunzător implementării ASIC-ului este cea utilizată pe tot parcursul acestei teze, CMOS de 32 nanometri pusă la dispoziție de către Synopsys și care face parte din biblioteca SAED_EDK32/28 nanometri.

Sinteza circuitului Grain Neuron s-a realizat ierarhic cu instrumentul Genus de la Cadence iar ulterior plasarea și rutarea s-a efectuat cu IC compiler de la Synopsys. Soluția aleasă pentru sinteză, instrumentul Genus este un instrument de sinteză RTL-to-gate de înaltă performanță conceput pentru a oferi rezultate de sinteză rapide, precise și eficiente atât din punct de vedere energetic cât și al ariei utilizate pentru generarea netlistului [141]. Este orientat și utilizat în proiectele digitale pentru dezvoltarea ASIC-urilor si SOC-urilor și în special oferă soluții optime în cazul circuitelor complexe, cum este cel al rețelelor neuronale. Motorul de sinteză și algoritmii pe care se bazează instrumentul Genus acceptă optimizări avansate, care includ rulări iterative pentru optimizarea timpului (retiming), partajarea resurselor și implementarea mecanismelor de clock gating. De asemenea, aceștia utilizează informații fizice relevante pentru etapele ulterioare de plasare și rutare a circuitului. Este compatibil cu instrumentele folosite după generarea netlistului, cele pentru plasarea și rutarea fizică a circuitului, în situația de față este compatibil cu IC Compiler de la Synospys și astfel ne putem baza pe o calitate previzibilă a rezultatelor. Un atu important al instrumentului Genus este capacitatea de sinteză ierahică, care în cazuri speciale, cum ar fi rețelele neuronale reprezintă o caracteristică utilă dat fiind structura cu caracter modular a acestor rețele. În acest caz al modulului Grain Neuron, Genus a permis sinteza fiecărui modul neuronal separat, iar această sinteză independentă fără a fi nevoie de a resintetiza tot circuitul este un avantaj major în cazul necesității unor modificări la nivel de neuron. În timpul sintezei, Genus realizează maparea circuitului pe o anumită librărie tehnologică și realizează optimizări de arie, viteză, consum energetic și generează rapoarte de analiză a timpului, utilizare a ariei și estimare de consum de power. Pentru rețelele neuronale acest aspect este esențial pentru atingerea obiectivelor de performanță care cuprind minimizarea ariei și creșterea vitezei de procesare.

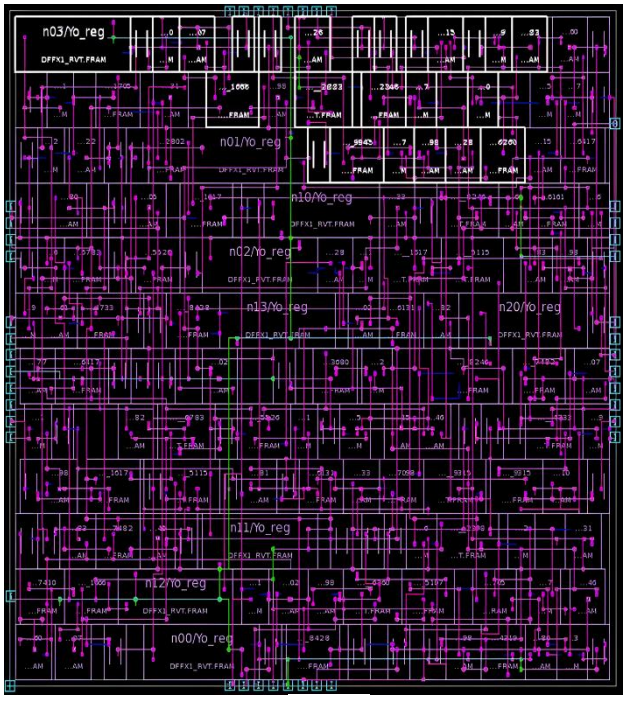
Implementarea circuitului Grain Neuron care cuprinde atât sinteza care are ca rezultat netlistul la nivel de poartă logică cât și plasarea respectiv rutarea circuitului în aria necesară, poate fi observată în figura 5.13.

Grupurile de porturi x01, x02, x03 și x04 corespunzătoare celor 4 neuroni instațiați în modulul Grain Neuron, au fost plasate strategic pe cele 4 laturi ale suprafeței circuitului. Această plasare s-a făcut astfel încât implementarea fizică a fiecărui neuron să se realizeze în proximitatea grupului său asociat de porturi de intrare, optimizând astfel organizarea circuitului în aria alocată

și o rutare mai eficientă decât în situația plasări flatenate. În figura 5.13.b grupul de celule logice marcate cu alb aparțin modulului neuronului care poarta numele de neuron_NG și al cărui netlist se poate observa în figura 5.13.a. Acest modul neuronal este instațiat în modulul top Grain Neuron cu numele de n03 și identificarea grupului de celule care aparțin acestui neuron se poate face prin asocierea logică și fizică cu grupul de porturi de intrare x03[*] ale acestui modul, care sunt plasate pe latura superioară a ariei circuitului. Aceste porturi de intrare pot fi interpretate ca și interfață primară prin care neuronii instanțiați în modulul Grain Neuron primesc semnale de la nivelurile ierarhice superioare. Corelarea spațială dintre celulele logice marcate cu alb și plasarea porturilor de intrare pe latura top a circuitului x03[*] furnizează un reper de încredere pentru localizarea fizică a instanței neuron_NG în layoutul prezentat în figura 5.13.b.

```
module neuron_NG(xi, bias, clk, Yo);
  input [7:0] xi, bias;
  input clk;
  output Yo;
  wire [7:0] xi, bias;
  wire clk;
  wire Yo;
  wire UNCONNECTED, n_0, n_1, n_2, n_3, n_4, n_5, n_6;
  wire n_7, n_8, n_9, n_10, n_11, n_12, n_13, n_14;
  wire n_15, n_16, n_17, n_18, n_19, n_20, n_21, n_22;
  wire n_23;
  DFFX1_RVT Yo_reg(.CLK (clk), .D (n_23), .Q (Yo), .QN (UNCONNECTED));
  NAND4X0_RVT g403__2398(.A1 (n_22), .A2 (n_20), .A3 (n_18), .A4
       (n_11), .Y (n_23));
  AO221X1_RVT g404__5107(.A1 (n_0), .A2 (xi[3]), .A3 (n_16), .A4
       (n_17), .A5 (n_21), .Y (n_22));
  OA222X1_RVT g405__6260(.A1 (xi[3]), .A2 (n_0), .A3 (n_8), .A4 (n_19),
       .A5 (xi[2]), .A6 (n_5), .Y (n_21));
  OA21X1_RVT g406__4319(.A1 (xi[4]), .A2 (n_16), .A3 (n_15), .Y (n_20));
  AOI22X1_RVT g407__8428(.A1 (n_14), .A2 (bias[1]), .A3 (n_9), .A4
       (n_4), .Y (n_19));
  NAND3X0_RVT g408__5526(.A1 (n_10), .A2 (n_1), .A3 (bias[6]), .Y
       (n_18));
  NAND3X0_RVT g409__6783(.A1 (n_13), .A2 (n_12), .A3 (n_3), .Y (n_17));
  NAND3X0_RVT g410__3680(.A1 (n_13), .A2 (n_2), .A3 (bias[5]), .Y
       (n_15));
  NAND3X0_RVT g411__1617(.A1 (n_13), .A2 (n_12), .A3 (bias[4]), .Y
       (n_16));
  OR2X1_RVT g412__2802(.A1 (n_4), .A2 (n_9), .Y (n_14));
  OA21X1_RVT g413__1705(.A1 (bias[6]), .A2 (n_1), .A3 (n_10), .Y
       (n_13));
  NAND2X0_RVT g414__5122(.A1 (n_6), .A2 (bias[7]), .Y (n_11));
  OR2X1_RVT g415__8246(.A1 (bias[5]), .A2 (n_2), .Y (n_12));
  AND2X1_RVT g416__7098(.A1 (n_5), .A2 (xi[2]), .Y (n_8));
  OR2X1_RVT g417__6131(.A1 (bias[7]), .A2 (n_6), .Y (n_10));
  NAND2X0_RVT g418__1881(.A1 (n_7), .A2 (xi[0]), .Y (n_9));
  INVX1_RVT g419(.A (bias[0]), .Y (n_7));
  INVX1_RVT g420(.A (xi[7]), .Y (n_6));
  INVX1_RVT g421(.A (bias[2]), .Y (n_5));
  INVX1_RVT g422(.A (xi[1]), .Y (n_4));
  INVX1_RVT g423(.A (xi[4]), .Y (n_3));
  INVX1_RVT g424(.A (xi[5]), .Y (n_2));
  INVX1_RVT g425(.A (xi[6]), .Y (n_1));
  INVX1_RVT g426(.A (bias[3]), .Y (n_0));
endmodule
```

(a)



(b)

Fig. 5.13. Proiectarea și implementarea circuitului Grain Neuron: (a) Netlistul generat de Genus pentru un neuron; (b) Plasarea și rutarea circuitului Grain Neuron[140]

Cele trei straturi neuronale sunt amplasate strategic în cadrul circuitului pentru a optimiza atât topologia conexiunilor, cât și eficiența comunicației interne. Primul strat neuronal, care servește drept strat de interfață cu exteriorul, conține neuronii n01, n02, n03 și n04. Aceștia sunt distribuiți de-a lungul celor patru laturi ale circuitului, în apropierea porturilor de intrare x01, x02, x03 și x04, astfel încât lungimea conexiunilor de la intrări către neuroni să fie minimizată. Această plasare are ca scop reducerea întârzierilor de propagare și a consumului energetic asociat transmisiei de date. Cel de-al doilea strat neuronal, compus din trei neuroni suplimentari, este

amplasat central, spre interiorul circuitului. Această poziționare facilitează colectarea semnalelor provenite de la primul strat și permite o distribuție echilibrată a resurselor de calcul și comunicație între straturi. În final, al treilea strat neuronal, care reprezintă stratul de ieșire și este responsabil de generarea rezultatului final al rețelei (ieșirea y), este localizat de-a lungul laturii drepte a circuitului, în imediata apropiere a portului de ieșire. Această amplasare permite o transmitere rapidă a semnalului de ieșire către exteriorul circuitului, asigurând un traseu optim din punct de vedere al distanței și latenței.

În figura 5.14 se poate observa marcat cu alb portul x03[7] din grupul de porturi de intrare x03[*] de intrare care aparțin modulului n03. De asemenea se pot observa în fereastra Properties, atributele asociate portului x03[7] și anume: direcția portului de intrare “in”, numele portului, metalul pe care este construit, precum și forma geometrică a portului reprezentată de coordonatele geometrice pe axele X și Y a circuitului.

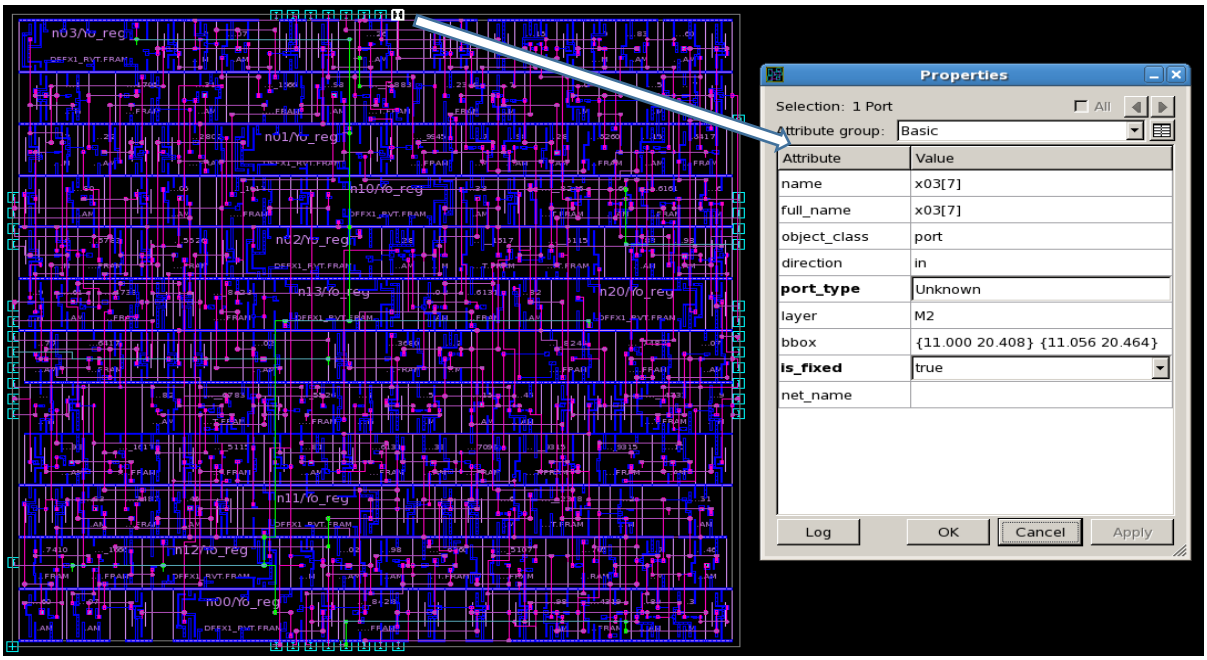


Fig. 5.14 Proprietățile portului de intrare x03[7] care aparține neuronului n03 [140]

Atât figura 5.13, cât și figura 5.14 oferă suport vizual și logic concludent pentru demonstrarea faptului că schematicul generat în mediul VIVADO este derivat din același cod sursă VHDL utilizat și în fluxul de proiectare ASIC. Acest cod a fost folosit pentru generarea netlisturilor aferente unui singur neuron, respectiv structurii neuronale Grain Neuron, asigurând astfel coerența între etapele de sinteză digitală pentru FPGA și pentru ASIC. Un exemplu concret al menținerii coerenței între cele 2 implementări îl reprezintă porturile de intrare și ieșire, care păstrează aceeași denumire atât în implementare pe FGPA cât și în implementarea ASIC.

Raportul de arie atât pentru un singur neuron cât și pentru întregul circuit Grain Neuron s-a generat cu instrumentul folosit la plasare și rutare IC Compiler și poate fi observat în figura 5.15. În figura 5.15.a este reprezentat raportul de arie pentru un singur neuron și cuprinde numărul de porturi, numărul de conexiuni, numărul de celule combinaționale și secvențiale precum și aria asociată celulelor și aria întregului netlist generat la nivel de poartă (flatenat). În contrast figura 5.15.b cuprinde informații despre numărul de porturi și a modulelor instanțiate în intregul circuit Grain Neuron al cărui netlist a fost generat ierarhic. Modulul Grain Neuron fiind sintetizat ierhic cuprinde 9 celule macro, o celulă macro pentru fiecare neuron în parte și anume: 4 neuroni instanțiați n01, n02, n03 și n04 care fac parte din primul strat de neurori, 4 neuroni care sunt în stratul din mijloc și nu pot fi accesați din exterior, un neuron la ieșirea rețelei neurale și componenentele adiționale responsabile cu generarea datelor de intrare necesare neuronilor.

O analiză critică comparativă dintre arhitectura NG și arhitecturile neuronale cunoscute în literatură este facută în tabelul 5.1, unde au fost luate în considerare mai multe caracteristici ale celor două tipuri de rețele neuronale. Arhitectura NG permite o mapare mult mai flexibilă a neuronilor și a operațiilor dintre aceștia pe resursele hardware, comparativ cu rețelele tradiționale care necesită un flux de date structurat. Acest avantaj al NG este evident în scenariile marginale, unde resursele hardware sunt restrânse. De asemenea modularitatea NG asigură o flexibilitate a împărțirii modulelor și a interconexiunilor dintre ele, care se poate face cu ușurință asigurând astfel și susținând implementarea scalabilă.

```
Number of ports:                         20
Number of nets:                          42
Number of cells:                         25
Number of combinational cells:           24
Number of sequential cells:               1
Number of macros:                         0
Number of buf/inv:                        8
Number of references:                    11

Combinational area:         44.983489
Buf/Inv area:               10.165760
Noncombinational area:       6.607744
Net Interconnect area:       5.320928

Total cell area:            51.591233
Total area:                 56.912161
```

(a)

```
Number of ports:                         44
Number of nets:                          72
Number of cells:                          9
Number of combinational cells:            0
Number of sequential cells:               0
Number of macros:                         0
Number of buf/inv:                        0
Number of references:                     9

Combinational area:        301.923076
Buf/Inv area:               66.077440
Noncombinational area:      59.469698
Net Interconnect area:      55.590310

Total cell area:           361.392774
Total area:                416.983083
```

(b)

Fig. 5.15 Raportul ariei după plasare și rutare; (a) Raportul ariei pentru un neuron; (b) Raportul ariei pentru Grain Neuron

Tabel 5.1 Comparație între rețeaua Grain Neuron și Arhitecturi neuronale tradiționale.

Caracteristică	**Grăunte Neuronal (GN)**	**Arhitecturi neuronale tradiționale**
Structură	Unitate compactă 3D: de exemplu, 4×4×1 straturi într-o configurație în formă de cub.	De obicei rețele 2D (stratificate); topologii complet conectate, CNN sau RNN.
Modularitate	Proiectat ca un bloc autonom (un "grăunte").	În general, nu este modularizat la nivel hardware; Organizare mai globală strat cu strat.
Scalabilitate	Extinderea arhitecturii se realizează prin interconectarea mai multor GN-uri.	Straturile pot fi scalate în software, dar replicarea hardware necesită adesea reproiectare sau duplicarea resurselor.
Implementare hardware	Proiectat nativ pentru implementare eficientă pe FPGA și ASIC.	Necesită optimizări suplimentare pentru implementări hardware complexe.
Împărțirea greutății / Localizare	Greutăți și biases locale - izolate în NG.	Greutăți globale sau la nivel de strat; poate necesita memorie partajată sau comunicare cu magistrală.
Strategia de antrenare	Antrenare offline: greutăți fixe sau actualizări locale limitate, cu accent pe eficiența hardware.	Antrenare intensivă realizată predominant în software.
Viteza de inferență	Inferență de mare viteză din unități compacte, paralele.	Poate suferi de latență din cauza rețelelor mai profunde, a recuperărilor de memorie.
Flexibilitatea topologiei	Structură locală definită, fix 4×4×1, dar flexibil ca unitate în modele mai mari.	Foarte flexibil în modelele abstracte, dar mai puțin structurat în aspectul fizic.
Amprentă de memorie	Mic — poate fi implementat în BRAM sau LUT-uri.	Mare – greutăți adesea stocate în memoria externă sau DDR.
Noutatea conceptuală	Unic: **Tratează un grup de neuroni 3D ca o unitate funcțională („grăunte de inteligență”).**	Convențional: Abstract, în mare parte matematică; Nu este proiectat având în vedere modularitatea fizică.
Caz de utilizare	AI încorporată, inferență constrânsă de hardware, extensii neuromorfice.	Inteligență artificială de uz general, modele bazate pe cloud, procesare de imagine/audio, calcul de mare putere.

În noua structură prezentată în figura 5.16, mai multe straturi de neuroni sunt suprapuse fizic, formând o arhitectură de rețea tridimensională.

Deasupra acestor straturi neuronale, poate fi integrat un circuit integrat compus din celule standard logice și memorii plasate deasupra celulelor logice, circuit care să activeze rutarea conexiunilor dintre straturi, controlul semnalelor și calculul local. Organizarea straturilor are ca rezultat un circuit 3D compact și scalabil, unde fiecare strat poate funcționa semi-independent sau în cooperare cu celelalte straturi. Implementarea actuală lasă spațiu pentru adăugarea de straturi neuronale suplimentare, ajutând astfel la extinderea viitoare. Această structură combină integrarea de înaltă densitate cu modularitatea, sporind astfel atât performanța cât și flexibilitatea arhitecturală. O astfel de abordare, deschide calea pentru sisteme neuromorfe de mare capacitate inspirate din punct de vedere biologic.

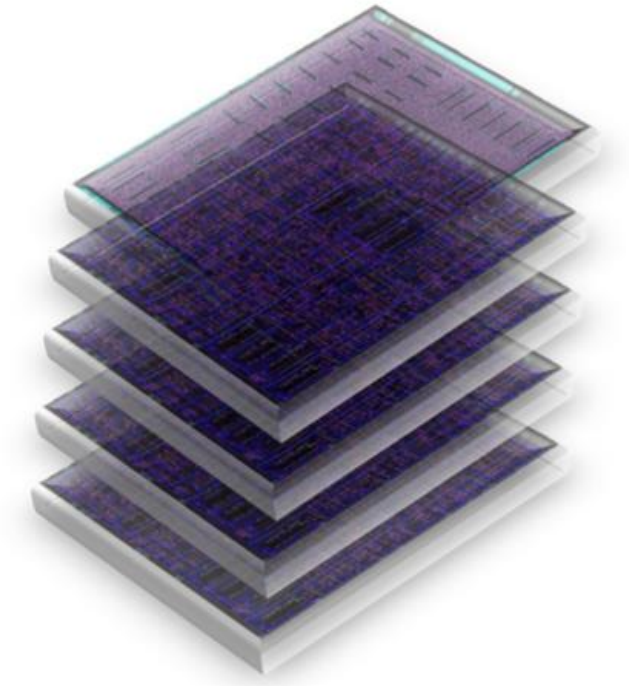


Fig. 5.16 Structură cu mai multe straturi de neuroni, iar deasupra lor circuitul compus din celule standard pe primul strat, cu blocuri de memorie poziționate deasupra celulelor

5.4 Concluzii

Rețeaua neuronală Neural Grain este introdusă ca o nouă arhitectura neuronală modulară 3D și definită ca un perceptron multistrat compact de dimensiune 4x4x1 care poate fi utilizată ca o unitate atomică de calcul în cadrul unor sisteme neuronale mai mari și scalabile. Spre deosebire de arhitecturile tradiționale de rețele neuronale, care adoptă de obicei topologii 2D centrate pe straturi și cu ponderi distribuite la nivel global, Neural Grain pune accent pe modularitatea fizică, calculul local și circuitul optimizat hardware. NG permite replicarea și aranjarea ușoară în

platforme bazate pe FPGA sau neuroforme fără a reproiecta logica de rutare sau alocare a resurselor. În timp ce arhitecturile neuronale tradiționale, precum rețelele complet conectate (FCN), rețelele neuronale convoluționale (CNN) și rețelele neuronale recurente (RNN), oferă flexibilitate topologică și performanțe ridicate în medii software sau cloud, implementarea acestora în hardware este adesea limitată de probleme de scalabilitate, constrângeri de timp real și cerințe ridicate privind consumul energetic. În schimb, Neural Grain excelează în sisteme încorporate și contextele edge-AI oferind inferență cu latență scăzută, comportament determinist și amprentă redusă de memorie. Compromisul constă în structura sa mai rigidă și lipsa suportului inerent pentru operațiunile spațiale (de exemplu convoluția), facându-l cel mai potrivit pentru utilizare ca element de bază în arhitecturi hibride mai complexe. Conceptul Neural Grain face astfel o punte între calculul neuronal abstract și implementările hardware practice, modulare.

CONCLUZII GENERALE ȘI RECOMANDĂRI

Tema propusă spre cercetare a avut ca scop dezvoltarea unei metodologii de implementare a unui circuit tridimensional folosind instrumentele tradiționale de implementare 2D, prin adaptarea fișierelor tehnologice astfel încât sa fie permisă suprapunerea straturilor. Această metodologie oferă suport tehnologiei monolitic 3D pentru fabricare înainte de dezvoltarea instrumentelor de proiectare care să suporte conceptul de multistrat. În baza cercetărilor și rezultatelor obținute se pot formula următoarele concluzii generale:

1. Abordarea tridimensională a implementării circuitelor electronice facilitează integrarea eterogenă (partea digitală, senzori și memorii pot fi implementate pe aceeași pastilă de siliciu) si utilizarea de tehnologii mixte (tehnologii vechi – 28nm - alături de tehnologii noi -7nm -, tehnologii analogice – BiCMOS - alături de tehnologii digitale standard – CMOS - și tehnologii specifice memoriilor – FLASH) [128].

2. Metodologia descrisă realizează o legătură între principiile teoretice și cele mai bune practici din industrie prin integrarea constrângerilor din lumea reală, limitările instrumentelor și considerațiile de fabricabilitate. Acest lucru asigură că fluxul nu este doar riguros din punct de vedere academic, ci și poate fi implementat în mediile comerciale de proiectare a semiconductorilor, sporind impactul și aplicabilitatea acestuia [116].

3. Procedura de lucru utilizată facilitează explorarea spațiului de proiectare în stadiu incipient, permițând proiectanților să facă compromisuri informate între putere, performanță și arie. Prin structurarea eficientă a strategiilor de sinteză și de plasare respectiv rutare a conexiunilor, aceasta permite optimizări care asigură eficiența proiectării, respectând în același timp constrângerile de timp, putere și implementare fizică [115-116].

4. Utilizarea scripturilor TCL facilitează controlul structurat al iterațiilor, permițând ajustări și optimizări rapide în timpul sintezei, DFT și plasării respectiv rutării. Această abordare accelerează convergența circuitului și îmbunătățește productivitatea permițând reglarea eficientă a parametrilor [116].

5. Integrarea 3D monolitică a logicii și a memoriilor, realizată prin adaptarea fluxului 2D, a condus la reducerea ariei totale cu 50%. Lungimea totală a conexiunilor rutate a scăzut cu aproximativ 20% în scenariul 3D comparativ cu 2D. Această reducere a lungimii interconectărilor a determinat o scădere cu 13.5% a numărului total de buffere**,** datorită necesității mai reduse de compensare a întârzierilor de propagare pe trasee mai scurte [113].

6. Viteza de propagare a datelor crește: proiectarea circuitului utilizând tehnologia monolitic 3D duce la creșterea performanțelor circuitului integrat final. Distribuția uniformă a

rețelei de ceas în scenariul 3D pe întreaga arie a circuitului (datorită eliberării suprafeței) comparativ cu scenariul 2D a condus la [114]:

- ✓ Numărul de inversoare și buffere folosite la crearea arborelui de ceas a scăzut cu 25.5%;
- ✓ Numărul total de celule de ceas s-a redus cu 15.3%;
- ✓ Aria totală a inversoarelor și buffere-lor a scăzut cu 11.6% ;
- ✓ Aria totală a celulelor de ceas s-a redus 9% ;
- ✓ Numărul de niveluri ale arborelui de ceas a crescut cu 12.5% contribuind la o distribuție mai uniforma a semnalului de ceas;
- ✓ Valoarea skew-ului este redusă cu 17.8%.

7. În dezvoltarea software-urilor care să realizeze implementarea circuitelor tridimensionale, o atenție deosebită se va acorda proiectării unui instrument de partiționare eficient, care pe baza unor date de intrare cum ar fi netlistul, fișierul de constrângeri și librăriile tehnologice să genereze soluții de împărțire a circuitului pe straturi[123].

8. Natura structurată și modulară a procesului de lucru descris pune bazele progreselor viitoare în proiectarea circuitelor integrate. Domeniile potențiale pentru cercetări ulterioare includ integrarea tehnicilor de inteligență artificială (AI) și de învățare automată (ML) pentru a îmbunătăți sinteza și optimizările locului și rutei, îmbunătățiri ale metodologiilor de procesare paralelă și perfecționarea în continuare a strategiilor de verificare pentru a crește automatizarea și acuratețea în validarea circuitului în stadiu incipient [116,140].

9. Trecerea de la metodologia de proiectare a circuitelor integrate în tehnologie 2D la proiectare în tehnologie 3D reprezintă o provocare majoră în microelectronică, dar și o oportunitate pentru inovare. Utilizarea unui flux de lucru experimental de proiectare 3D care folosește instrumente 2D poate constitui un punct de plecare pentru adaptarea și dezvoltarea de noi tehnologii care să faciliteze progresul în dezvoltarea circuitelor integrate tridimensionale. Întregul flux de lucru de Backend al implementării circuitului integrat tridimensional a fost executat folosind instrumente standard 2D EDA cu modificări adecvate ale fișierelor tehnologice [115-116].

În baza analizei rezultatelor obținute în lucrare se pot formula următoarele **recomandări:**

1. Se recomandă folosirea lanțurilor de instrumente acreditate de fabrica a căror proces este folosit (de ex. TSMC folosește instrumente de la Synopsys și de la Cadence) pentru dezvoltarea și implementarea flow-ului de lucru și/sau a circuitului integrat tridimensional.

2. Se recomandă înțelegerea în aprofunzime a fișierelor tehnologice (LEF, DEF, LIB) ce urmează a fi adaptate pentru integrarea tridimensională. Informațiile conținute în aceste fișiere pot fi modificate astfel încât să facilititeze acceptarea de către instrumentele de plasare și rutare a suprapunerii celulelor între ele fără a raporta erori.

3. Înțelegerea în detaliu a modului de funcționare a instrumentelor folosite pentru implementarea flow-ului – cu referire la algoritmii interni, a etapelor în care sunt folosiți și a succesiunii acestor etape. Aceasta se recomandă pentru a înțelege modul în care instrumentele interpretează datele de intrare în fiecare etapă și modul de evaluare a parametrilor specifici fiecărei etape.

4. Se recomandă folosirea circuitelor care se pliază pe abordarea tridimensională - prezintă avantaje la folosirea de tehnologii diferite:

a. necesită componente analogice (de ex. convertoare, PLL);

b. folosesc memorii FLASH;

c. au în componența lor un necesar de memorii SRAM a căror arie este comparabilă cu aria celulelor logice standard;

d. conțin blocuri funcționale cu necesar de frecvență implementabilă atât în tehnologii noi (de ex. CPU) cât și în tehnologii mai vechi (de ex. interfață de comunicare).

5. Înainte de a investi în instrumente de plasare și rutare 3D, se recomandă să se facă cercetare și analiză pe circuit și librărie cu ajutorul instrumentelor actuale 2D. Astfel se pot valida avantajele implementării tridimensionale în termeni de arie, conexiuni, viteza, consum de putere și se poate dezvolta flow-ul care să fie folosit cu instrumentele 3D.

BIBLIOGRAFIE

1. MOORE, G.E. Cramming More Components onto Integrated Circuits. In: *IEEE Solid-State Circuits Society Newsletter*. 2009. Vol. 11, no. 3, p. 33–35. ISSN 1098-4232.

2. DEBENEDICTIS, E.P., BADAROGLU, M., CHEN, A., CONTE, T.M., and GARGINI, P. Sustaining Moore's law with 3D chips. In: *Computer*. 2017. Vol. 50, no. 8, p. 69–73. ISSN 0018-9162.

3. SHALF, J. The future of computing beyond Moore's Law. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2020. Vol. 378, no. 2166, p. 20190061. ISSN 1364-503X, 1471-2962.

4. ZHANG, N. Moore's Law is dead, long live Moore's Law！. arXiv:2205.15011v1.2022

5. POLLIE, R. Nanosheet Chips Poised to Rescue Moore's Law. In: *Engineering*. 2021. Vol. 7, no. 12, p. 1655–1656. ISSN 20958099.

6. DAVIS, J.A., VENKATESAN, R., KALOYEROS, A., BEYLANSKY, M., SOURI, S.J., BANERJEE, K., SARASWAT, K.C., RAHMAN, A., REIF, R., and MEINDL, J.D. Interconnect limits on gigascale integration (GSI) in the 21st century. In: *Proceedings of the IEEE*. 2001. Vol. 89, no. 3, p. 305–324. ISSN 00189219.

7. ALPERT, C., CHU, C., GANDHAM, G., HRKIC, M., HU, J., KASHYAP, C., and QUAY, S. Simultaneous Driver Sizing and Buffer Insertion Using a Delay Penalty Estimation Technique. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 2004. Vol. 23, no. 1, p. 136-141. ISSN: 0278-0070, 1937-4151.

8. ZHOU, L., WAKAYAMA, C., PANDA, R., JANGKRAJARNG, N., HU, B., and SHI, C.-J.R. *Implementing a 2-Gbs 1024-bit ½-rate low-density parity-check code decoder in three-dimensional integrated circuitss*. In: *2007 25th International Conference on Computer Design*. 2007. p. 194–201. ISBN 978-1-4244-1257-0.

9. KIM, D.H., KIM, S., and LIM, S.K. A Study on the Impact of Nano-Scale TSVs on 3D IC Designs. In: *2011 IEEE/ACM International Workshop on System Level Interconnect Prediction (SLIP)*. 2011. p. 7-12. ISBN 978-1-4577-1240-1, 978-1-4577-1241-8.

10. LIU, Z., LI, Y., HU, J., YU, X., SHIAU, S., AI, X., ZENG, Z., and ZHANG, Z. DeepOHeat: Operator Learning-based Ultra-fast Thermal Simulation in 3D-IC Design. In: *2023 60th ACM/IEEE Design Automation Conference (DAC)*. 2023. p. 1–6. ISBN 979-8-3503-2348-1.

11. CAO, K., ZHOU, J., WEI, T., CHEN, M., HU, S., and LI, K. A survey of optimization techniques for thermal-aware 3D processors. In: *Journal of Systems Architecture*. 2019. Vol. 97, p. 397–415. ISSN 13837621.

12. SAMAL, S.K., PANTH, S., SAMADI, K., SAEIDI, M., DU, Y., and LIM, S.K. Adaptive Regression-Based Thermal Modeling and Optimization for Monolithic 3-D ICs. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2016. Vol. 35, no. 10, p. 1707–1720. ISSN 0278-0070, 1937-4151.

13. DELHAYE, Q., BEYNE, E., GOOSSENS, J., VAN DER PLAS, G., and MILOJEVIC, D. Impact of gate-level clustering on automated system partitioning of 3D-ICs. In: *Microelectronics Journal*. 2023. Vol. 139, p. 105896. ISSN 18792391.

14. SAIT, S.M., OUGHALI, F.C., and AL-ASLI, M. Design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing. In: *Journal of Applied Research and Technology*. 2016. Vol. 14, no. 1, p. 67–76. ISSN 2448-6736.

15. DAS, S., KUMARI, B., SATYENDRA SAHOO, S., CHEN, Y., MYERS, J., MILOJEVIC, D., BISWAS, D., and RYCKAERT, J. Toward Fine-Grained Partitioning of Low-Level SRAM Caches for Emerging 3D-IC Designs. In: *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*. 2024. Vol. 10, p. 67–74. ISSN 2329-9231.

16. LEI, X. Review on the FPGA design optimization for ASIC. In: *Applied and Computational Engineering*. 2023. Vol. 4, no. 1, p. 524–531. ISSN 2755-2721, 2755-273X.

17. NORTH CAROLINA STATE UNIVERSITY, and LUO, Y. FPGA Implementation for Rapid Prototyping of High Performance Voltage Source Inverters. In: *CPSS Transactions on Power Electronics and Applications*. 2021. Vol. 6, no. 4, p. 320–331. ISSN 2475742X.

18. KANG, B.J., LEE, H.I., YOON, S.K., KIM, Y.C., JEONG, S.B., O, S.J., and KIM, H. A survey of FPGA and ASIC designs for transformer inference acceleration and optimization. In: *Journal of Systems Architecture*. 2024. Vol. 155, p. 103247. ISSN 13837621.

19. FAROOQ, U., and ALZAHRANI, B.A. Exploring and optimizing partitioning of large designs for multi-FPGA based prototyping platforms. In: *Computing*. 2020. Vol. 102, no. 11, p. 2361–2383. ISSN 0010-485X, 1436-5057.

20. CHAUDHURI, A., BANERJEE, S., PARK, H., KIM, J., MURALI, G., LEE, E., KIM, D., LIM, S.K., MUKHOPADHYAY, S., and CHAKRABARTY, K. Advances in Design and Test of Monolithic 3-D ICs. In: *IEEE Design & Test*. 2020. Vol. 37, no. 4, p. 92–100. ISSN 2168-2356, 2168-2364.

21. DEWAN, M.I., LIN, S.-E.D., and KIM, D.H. Construction of All Multilayer Monolithic RSMTs and Its Application to Monolithic 3D IC Routing. In: *ACM Transactions on Design Automation of Electronic Systems*. 2024. Vol. 29, no. 1, p. 1–28. ISSN 1084-4309, 1557-7309.

22. COLINGE, J.P., and DEMOULIN, E. A high density CMOS inverter with stacked transistors. In: *IEEE Electron Device Letters*. 1981. Vol. 2, no. 10, p. 250–251. ISSN 0741-3106.

23. SABBAVARAPU, S., ACHARYYA, A., BALASUBRAMANIAN, P., and REDDY, C.R. Fast 3D Integrated Circuit Placement Methodology using Merging Technique. In: *Defence Science Journal*. 2019. Vol. 69, no. 3, p. 217–222. ISSN 0011-748X.

24. KALARGARIS, H., and PAVLIDIS, V.F. Voltage scaling for 3-D ICs: When, how, and how much? In: *Microelectronics Journal*. 2017. Vol. 69, p. 35–44. ISSN 18792391.

25. EMMA, P.G., and KURSUN, E. Is 3D chip technology the next growth engine for performance improvement? In: *IBM Journal of Research and Development*. 2008. Vol. 52, no. 6., p. 541-552, ISSN 0018-8646.

26. MATTHIAS, T., WIMPLINGER, M., PARGFRIEDER, S., and LINDNER, P. 3D Process Integration – Wafer-to-Wafer and Chip-to-Wafer Bonding. In: *MRS Proceedings*. 2006. Vol. 970, p. 0970-Y04- 08. ISSN 0272-9172, 1946-4274.

27. JIN, H. The History, Current Applications and Future of Integrated Circuit. In: *Highlights in Science, Engineering and Technology*. 2023. Vol. 31, p. 232–238. ISSN 2791-0210.

28. LEI, W., and YAO, S. A basic introduction to integrated circuit technology. In: *Applied and Computational Engineering*. 2024. Vol. 78, no. 1, p. 54–59. ISSN 2755-2721, 2755-273X.

29. WIDLAR, R. Some Circuit Design Techniques for Linear Integrated Circuits. In : *IEEE Transactions on Circuit Theory*. 1965. Vol. 12, no. 4, p. 586–590. ISSN 0018-9324.

30. BORKAR, S., and CHIEN, A.A. The future of microprocessors. In: *Communications of the ACM*. 2011. Vol. 54, no. 5, p. 67–77. ISSN 0001-0782, 1557-7317.

31. MEIERAN, E.S., FLINN, P., and CARRUTHERS, J.R. Analysis Technology for VLSl Fabrication. In: *IEEE*. 1987. Vol. 75, no. 7, p. 908–955. ISSN 1558-2256.

32. DEPARTMENT OF ELECTRONIC ENGINEERING, BS WEST VIRGINIA UNIVERSITY CHARLOTTESVILLE., ALMUTAIRI*, A.H., ALSUMAIT, E., and DEPARTMENT OF MEDICAL ELECTRONIC IN COLLEGE OF TECHNOLOGY, PAAET ADAILIYAH, KUWAIT. CMOS Transistor and Applications. In: *International Journal of Innovative Technology and Exploring Engineering*. 2019. Vol. 9, no. 2, p. 2552–2558. ISSN 22783075.

33. DENNARD, R.H., GAENSSLEN, F.H., YU, H.-N., RIDEOUT, V.L., BASSOUS, E., and LEBLANC, A.R. Design of ion-implanted MOSFET's with very small physical dimensions. In: *IEEE Journal of Solid-State Circuits.* 1974. Vol. 9, no. 5, p. 256-268. ISSN: 0018-9200, 1558-173X.

34. CHEN, J. Integrated circuit design based on CMOS technology principle and its application in GPU. In: *Theoretical and Natural Science*. 2023. Vol. 12, no. 1, p. 141–146. ISSN 2753-8818, 2753-8826.

35. WONG, H. Abridging CMOS Technology. In: *Nanomaterials*. 2022. Vol. 12, no. 23, p. 4245. ISSN 2079-4991.

36. RADAMSON, H.H., MIAO, Y., ZHOU, Z., WU, Z., KONG, Z., GAO, J., YANG, H., REN, Y., ZHANG, Y., SHI, J., XIANG, J., CUI, H., LU, B., LI, J., LIU, J., LIN, H., XU, H., LI, M., CAO, J., HE, C., DUAN, X., ZHAO, X., SU, J., DU, Y., YU, J., WU, Y., JIANG, M., LIANG, D., LI, B., DONG, Y., and WANG, G. CMOS Scaling for the 5 nm Node and Beyond: Device, Process and Technology. In: *Nanomaterials*. 2024. Vol. 14, no. 10, p. 837. ISSN 2079-4991.

37. RADAMSON, H.H., HE, X., ZHANG, Q., LIU, J., CUI, H., XIANG, J., KONG, Z., XIONG, W., LI, J., GAO, J., YANG, H., GU, S., ZHAO, X., DU, Y., YU, J., and WANG, G. Miniaturization of CMOS. In: *Micromachines*. 2019. Vol. 10, no. 5, p. 293. ISSN 2072-666X.

38. PATIL, C.V., and SUMA, M.S. Compact modeling of through silicon vias for thermal analysis in 3-D IC structures. In: *Sādhanā*. 2021. Vol. 46, no. 1, p. 35. ISSN 0256-2499, 0973-7677.

39. WANG, L., WOYCHIK, C.G., GAO, G., VILLAVICENCIO, G., MCGRATH, S., SHEN, H., and ARKALGUD, S. Challenges of Scalable 2.5D IC Assembly. In: *Journal of Microelectronics and Electronic Packaging*. 2015. Vol. 12, no. 3, p. 123–128. ISSN 1551-4897.

40. SHEN, W.-W., and CHEN, K.-N. Three-Dimensional Integrated Circuit (3D IC) Key Technology: Through-Silicon Via (TSV). In: *Nanoscale Research Letters*. 2017. Vol. 12, no. 1, p. 56. ISSN 1931-7573, 1556-276X.

41. ARDOUIN, B., JOHANSEN, T.K., CHAUVET, A., HERSENT, R., NODJIADJIM, V., KONCZYKOWSKA, A., DAVY, N., RIET, M., and MISMER, C. Challenges for 2.5D and 3D Integration of InP HBT Technology. In: *Key enabling technologies for future wireless, wired, optical and satcom applications*. 1. 2024. p. 81–89. ISBN 978-87-7004-666-4.

42. ZHANG, P., WANG, D.-W., ZHAO, W.-S., SU, J., YOU, B., and LIU, J. Multiphysics Analysis and Optimal Design of Compressible Micro-Interconnect for 2.5D/3D Heterogeneous Integration. In: *Electronics*. 2021. Vol. 10, no. 18, p. 2240. ISSN 2079-9292.

43. TAO, Z., XU, N., WU, Y., ZHONG, Y., and YU, D. Heterogeneous Integration of Diamond-on-Chip-on-Interposer for High-Performance Thermal Management in Supercomputing 2.5D Chiplets Packaging. In: *2024 25th International Conference on Electronic Packaging Technology (ICEPT)*. 2024. p. 1–4. ISBN 979-8-3503-5380-8.

44. HUNG, S.-C., BHOUMIK, P., CHAUDHURI, A., BANERJEE, S., and CHAKRABARTY, K. Design-for-Test Solutions for 3-D Integrated Circuits. In: *Integrated Circuits and Systems*. 2024. Vol. 1, no. 1, p. 3–17. ISSN 2995-1968, 2995-1976.

45. LIU, D., and PARK, S. Three-Dimensional and 2.5 Dimensional Interconnection Technology: State of the Art. In: *Journal of Electronic Packaging*. 2014. Vol. 136, no. 1, p. 014001. ISSN 1043-7398, 1528-9044.

46. ZHAO, G., SONG, C., and WU, B. 3D Integrated Circuit (3D IC) Technology and Its Applications. In: *Journal of Industrial Engineering & Applied Science*. 2024. Vol. 2, no. 4, p. 60-65. ISSN 3005-6071.

47. JI, L., CHE, F.X., JI, H.M., LI, H.Y., and KAWANO, M. Wafer-to-Wafer Hybrid Bonding Development by Advanced Finite Element Modeling for 3-D IC Packages. In: *IEEE Transactions on Components, Packaging and Manufacturing Technology*. 2020. Vol. 10, no. 12, p. 2106–2117. ISSN 2156-3950, 2156-3985.

48. XU, F., CHANG, S., ZHAO, Y., LI, W., and XIA, H. High-precision moiré alignment method using centrosymmetric grating for 3D interlayer interconnect. In: *Optics and Lasers in Engineering*. 2023. Vol. 171, p. 107815. ISSN 01438166.

49. WALKER, E.M., EMANUEL, A., and HALLEN, H.D. Self-alignment of whole wafers using patterning for capillary forces. In: *Journal of Vacuum Science & Technology B*. 2023. Vol. 41, no. 3, p. 032203. ISSN 2166-2746, 2166-2754.

50. DAE HYUN KIM and SUNG KYU LIM. Physical Design and CAD Tools for 3-D Integrated Circuits: Challenges and Opportunities. In: *IEEE Design & Test*. 2015. Vol. 32, no. 4, p. 8–22. ISSN 2168-2356, 2168-2364.

51. LIM, S.K. Research Needs for TSV-Based 3D IC Architectural Floorplanning. In: *Journal of information and communication convergence engineering*. 2014. Vol. 12, no. 1, p. 46–52. ISSN 2234-8255.

52. YI, Y., and ZHOU, Y. Differential Through-Silicon-Vias modeling and design optimization to benefit 3D IC performance. In: *2013 IEEE 22nd Conference on Electrical Performance of Electronic Packaging and Systems*. 2013. p. 195–198. ISBN 978-1-4799-0707-6.

53. LIU, C.C., GANUSOV, I., BURTSCHER, M., and TIWARI, S. Bridging the Processor-Memory Performance Gapwith 3D IC Technology. In: *IEEE Design and Test of Computers*. 2005. Vol. 22, no. 6, p. 556–564. ISSN 0740-7475.

54. WANG, J., DUAN, F., LV, Z., CHEN, S., YANG, X., CHEN, H., and LIU, J. A Short Review of Through-Silicon via (TSV) Interconnects: Metrology and Analysis. In: *Applied Sciences*. 2023. Vol. 13, no. 14, p. 8301. ISSN 2076-3417.

55. CHO, D.H., SEO, S.M., KIM, J.B., RAJENDRAN, S.H., and JUNG, J.P. A Review on the Fabrication and Reliability of Three-Dimensional Integration Technologies for Microelectronic Packaging: Through-Si-via and Solder Bumping Process. In: *Metals*. 2021. Vol. 11, no. 10, p. 1664. ISSN 2075-4701.

56. ABDULLAH, M.F., and LEE, H.W. Technology review of CNTs TSV in 3D IC and 2.5D packaging: Progress and challenges from an electrical viewpoint. In: *Microelectronic Engineering*. 2024. Vol. 290, p. 112189. ISSN 01679317.

57. LU, T., SERAFY, C., YANG, Z., SAMAL, S.K., LIM, S.K., and SRIVASTAVA, A. TSV-Based 3-D ICs: Design Methods and Tools. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2017. Vol. 36, no. 10, p. 1593–1619. ISSN 0278-0070, 1937-4151.

58. KUMAR, V., and NAEEMI, A. An overview of 3D integrated circuits. In: *2017 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO)*. 2017. p. 311–313. ISBN 978-1-5090-4837-3.

59. XIAO, H., HE, H., REN, X., ZENG, P., and WANG, F. Numerical modeling and experimental verification of copper electrodeposition for through silicon via (TSV) with additives. In: *Microelectronic Engineering*. 2017. Vol. 170, p. 54–58. ISSN 01679317.

60. KIM, D.H., and LIM, S.K. Impact of through-silicon-via scaling on the wirelength distribution of current and future 3D ICs. In: *2011 IEEE International Interconnect Technology Conference*. 2011. p. 1–3. ISBN 978-1-4577-0503-8.

61. VEMURI, M.S., and TIDA, U.R. Modeling and Design of Dual-Purpose MIV in Monolithic 3D IC. In: *IEEE Access*. 2024. Vol. 12, p. 23066–23080. ISSN 2169-3536.

62. DHANANJAY, K., SHUKLA, P., PAVLIDIS, V.F., COSKUN, A., and SALMAN, E. Monolithic 3D Integrated Circuits: Recent Trends and Future Prospects. In: *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2021. Vol. 68, no. 3, p. 837–843. ISSN 1549-7747, 1558-3791.

63. KUNIO, T., OYAMA, K., HAYASHI, Y., and MORIMOTO, M. Three dimensional ICs, having four stacked active device layers. In: *International Technical Digest on Electron Devices Meeting*. 1989. p. 837–840.

64. HAYASHI, Y., OYAMA, K., TAKAHASHI, S., WADA, S., KAJIYANA, K., KOH, R., and KUNIO, T. A new three dimensional IC fabrication technology, stacking thin film DUAL-CMOS layers. In: *International Electron Devices Meeting 1991 [Technical Digest]*. 1991. p. 657–660. ISBN 978-0-7803-0243-3.

65. PANTH, S., SAMAL, S.K., SAMADI, K., DU, Y., and LIM, S.K. Tier Degradation of Monolithic 3-D ICs: A Power Performance Study at Different Technology Nodes. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2017. Vol. 36, no. 8, p. 1265–1273. ISSN 0278-0070, 1937-4151.

66. PANTH, S., SAMADI, K., DU, Y., and LIM, S.K. Power-Performance Study of Block-Level Monolithic 3D-ICs Considering Inter-Tier Performance Variations. In: *Proceedings of the 51st Annual Design Automation Conference*. 2014. p. 1–6. ISBN 978-1-4503-2730-5.

67. YU, J.-C., and YU, P. Impacts of cost functions on inverse lithography patterning. In: *Optics Express*. 2010. Vol. 18, no. 22, p. 23331. ISSN 1094-4087.

68. PENNISI, S. The Integrated Circuit Industry at a Crossroads: Threats and Opportunities. In: *Chips*. 2022. Vol. 1, no. 3, p. 150–171. ISSN 2674-0729.

69. NGUYEN, A.H.-T., NGUYEN, M.-C., NGUYEN, A.-D., JEON, S.J., PARK, N.-H., LEE, J.-H., and CHOI, R. Formation techniques for upper active channel in monolithic 3D integration: an overview. In: *Nano Convergence*. 2024. Vol. 11, no. 1, p. 5. ISSN 2196-5404.

70. PARK, H., KU, B.W., CHANG, K., SHIM, D.E., and LIM, S.K. Pseudo-3D Physical Design Flow for Monolithic 3D ICs: Comparisons and Enhancements. In: *ACM Transactions on Design Automation of Electronic Systems*. 2021. Vol. 26, no. 5, p. 1–25. ISSN 1084-4309, 1557-7309.

71. CHENG, Y., GUO, X., and PAVLIDIS, V.F. Emerging monolithic 3D integration: Opportunities and challenges from the computer system perspective. In: *Integration*. 2022. Vol. 85, p. 97–107. ISSN 01679260.

72. FAN, Y., AN, R., TANG, J., LI, Y., LIU, T., GAO, B., QIAN, H., and WU, H. Monolithic 3D integration as a pathway to energy-efficient computing and beyond: From materials and devices to architectures and chips. In: *Current Opinion in Solid State and Materials Science*. 2024. Vol. 33, p. 101199. ISSN 13590286.

73. SUN, G. Three-dimensional Integrated Circuits: Design, EDA, and Architecture. In: *Foundations and Trends® in Electronic Design Automation*. 2011. Vol. 5, no. 1–2, p. 1–151. ISSN 1551-3939, 1551-3947.

74. BAGATIN, M., GERARDIN, S., PACCAGNELLA, A., COSTANTINO, A., FERLET-CAVROIS, V., PESCE, A., and BELTRAMI, S. Single Event Effects in 3-D NAND Flash Memory Cells With Replacement Gate Technology. In: *IEEE Transactions on Nuclear Science*. 2023. Vol. 70, no. 4, p. 308–313. ISSN 0018-9499, 1558-1578.

75. KIM, H., AHN, S.J., SHIN, Y.G., LEE, K., and JUNG, E. Evolution of NAND Flash Memory: From 2D to 3D as a Storage Market Leader. In: *2017 IEEE International Memory Workshop (IMW)*. 2017. p. 1–4. ISBN 978-1-5090-3274-7.

76. JUN, H., CHO, J., LEE, K., SON, H.-Y., KIM, K., JIN, H., and KIM, K. HBM (High Bandwidth Memory) DRAM Technology and Architecture. In: *2017 IEEE International Memory Workshop (IMW)*. 2017. p. 1–4. ISBN 978-1-5090-3274-7.

77. INGERLY, D.B., ENAMUL, K., GOMES, W., JONES, D., KOLLURU, K.C., KANDAS, A., KIM, G.-S., MA, H., PANTUSO, D., PETERSBURG, C.F., PHEN-GIVONI, M., AMIN, S., PILLAI, A.M., SAIRAM, A., SHEKHAR, P., SINHA, P., STOVER, P., TELANG, A., ZELL, Z., ARYASOMAYAJULA, L., BALANKUTTY, A., BORST, D., CHANDRA, A., CHEEMALAPATI, K., COOK, C.S., and CRISS, R. Foveros: 3D Integration and the use of Face-to-Face Chip Stacking for Logic Devices. In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019. p. 19.6.1-19.6.4. ISBN 978-1-7281-4032-2.

78. WUU, J., AGARWAL, R., CIRAULA, M., DIETZ, C., JOHNSON, B., JOHNSON, D., SCHREIBER, R., SWAMINATHAN, R., WALKER, W., and NAFFZIGER, S. 3D V-Cache: the Implementation of a Hybrid-Bonded 64MB Stacked Cache for a 7nm x86-64 CPU. In: *2022 IEEE International Solid- State Circuits Conference (ISSCC)*. 2022. p. 428–429. ISBN 978-1-6654-2800-2.

79. PAWLOWSKI, J.T. Hybrid memory cube (HMC). In: *2011 IEEE Hot Chips 23 Symposium (HCS)*. 2011. p. 1–24. ISBN 978-1-4673-8877-1.

80. CHIPER, R., ALEXA, D., **PLETEA, I.M.,** Petrichei A., A Comparative Study between Classical Three Phase Rectifier Configuration with Passive Filters and Converter Configurations with Small Harmonic Contents Having Capacitors on the AC Side. In*: Buletinul Institutului Politehnic din Timişoara, Tom 53(67), Fascicola 1. 2008.* p. 19–22. ISSN 1583-3380.

81. CHIPER, R., ALEXA, D., GORAS, T.C., PLETEA, I.V., **PLETEA, I.M.,** and ALEXANDRESCU, A. An Analysis of Asymmetrical RNSIC Converter with Capacitors on the AC Side for High and Medium Power. In: *2007 International Symposium on Signals, Circuits and Systems*. 2007. p. 1–4. ISBN 978-1-4244-0968-6.

82. SEKAR, D.C., AZAD NAEEMI, SARVARI, R., DAVIS, J.A., and MEINDL, J.D. Intsim: a CAD tool for optimization of multilevel interconnect networks. In: *2007 IEEE/ACM International Conference on Computer-Aided Design*. 2007. p. 560–567. ISBN 978-1-4244-1381-2. ISSN 1092-3152.

83. DE CARVALHO, M., NUNES, C., CANAL, B., PURICELLI, L., REINICKE, L., WEBBER, G., NEUTZLING, A., ALTIERI, M., CONTO, E., NAGEL, T., BUTZEN, P., RIBAS, R.P., and FABRIS, E. Standard Cell Library Validation Methodology. In: *Proc. 6th Workshop Circuits and Systems (WCAS).* 2017.

84. E. GERALLA, L., JOEY DE GUZMAN, M., and A. HORA, J. Optimization of Physically-Aware Synthesis for Digital Implementation Flow. In: *International Journal of Engineering & Technology*. 2018. Vol. 7, no. 2.11, p. 31. ISSN 2227-524X.

85. GARG, S., and SHUKLA, N.Kr. A Study of Floorplanning Challenges and Analysis of macro placement approaches in Physical Aware Synthesis. In: *International Journal of Hybrid Information Technology*. 2016. Vol. 9, no. 1, p. 279–290. ISSN 17389968, 17389968.

86. LYLINA, N., WANG, C.-H., and WUNDERLICH, H.-J. A Complete Design-for-Test Scheme for Reconfigurable Scan Networks. In: *Journal of Electronic Testing*. 2022. Vol. 38, no. 6, p. 603–621. ISSN 0923-8174, 1573-0727.

87. SHANTAGIRI, P.V., and KAPUR, R. Aggressive Exclusion of Scan Flip-Flops from Compression Architecture for Better Coverage and Reduced TDV: A Hybrid Approach. In: *Journal of Low Power Electronics and Applications*. 2019. Vol. 9, no. 2, p. 18. ISSN 2079-9268.

88. SHINY, M.I., and NIRMALA DEVI, M. Trustworthy Scan Design and Testability Using Obfuscation and Logic Locking Scheme for Wireless Network Application. In: *Mobile Networks and Applications*. 2022. Vol. 27, no. 3, p. 1000–1018. ISSN 1383-469X, 1572-8153.

89. S, D.C., and MS, S. Multi-objective Floorplanning optimization engaging dynamic programming for system on chip. In: *Microelectronics Journal*. 2023. Vol. 140, p. 105942. ISSN 18792391.

90. HE, X., FU, Z., WANG, Y., LIU, C., and GUO, Y. Accurate timing prediction at placement stage with look-ahead RC network. In: *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 2022. p. 1213–1218. ISBN 978-1-4503-9142-9.

91. LOTFY, S.M.A. Ant Colony Optimization for Power Efficient Routing in VLSI Design. In: *2009 IEEE Swarm Intelligence Symposium.* 2009. ISBN: 978-1-4244-2762-8.

92. XIE, Z., LIANG, R., XU, X., HU, J., DUAN, Y., and CHEN, Y. *Net2: A Graph Attention Network Method Customized for Pre-Placement Net Length Estimation*. In: *Proceedings of the 26th Asia and South Pacific Design Automation Conference (ASP-DAC).* 2021. p. 671-677. DOI: 10.1145/3394885.3431562.

93. HUANG, Z., HUANG, H., SHI, R., LI, X., ZHANG, X., CHEN, W., WANG, J., and ZHU, Z. Detailed Placement and Global Routing Co-Optimization with Complex Constraints. In: *Electronics*. 2021. Vol. 11, no. 1, p. 51. ISSN 2079-9292.

94. HAO, R., CAI, Y., and ZHOU, Q. Intelligent and kernelized placement: A survey. In: *Integration*. 2022. Vol. 86, p. 44–50. ISSN 01679260.

95. DATABOOK: SAED32.28nm_Digital_Standard_Cell_Library_b100_01312012.pdf;

Revision 1.0.0; @2012 Synopsys ARMENIA Educational Department.

96. RAMALINGA REDDY KOTAPATI. Advanced Clock Tree Synthesis Optimization: A Multi-Source Approach to Minimizing Skew and Power in Sub-7nm ASIC Designs. In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2024. Vol. 10, no. 6, p. 2275–2283. ISSN 2456-3307.

97. KWON, N., and PARK, D. Shallow Clock Tree Pre-Estimation for Designing Clock Tree Synthesizable Verilog RTLs. In: *Electronics*. 2023. Vol. 12, no. 20, p. 4340. ISSN 2079-9292.

98. SITIK, C., LIU, W., TASKIN, B., and SALMAN, E. Design Methodology for Voltage-Scaled Clock Distribution Networks. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2016. Vol. 24, no. 10, p. 3080–3093. ISSN 1063-8210, 1557-9999.

99. LUET NG, L., HO YEAP, K., WAN CHING GOH, M., and DAKULAGI, V. Power Consumption in CMOS Circuits. In: SONG, Hai-Zhi, HO YEAP, Kim and WAN CHING GOH, Magdalene (eds.), *Electromagnetic Field in Advancing Science and Technology*. 2023. ISBN 978-1-80356-677-1.

100. CHANG, K., and KIM, T. Pre-route timing prediction and optimization with graph neural network models. In: *Integration*. 2024. Vol. 99, p. 102262. ISSN 01679260.

101. TIGADI, A., and KEDAR, P. Survey on Design Automation and Verification for VLSI Circuits. In: *International Journal of Research in Analytical Reviews (IJRAR)*. 2023. Vol. 10, no. 2, p. 31-35. ISSN 2349-5138, 2348-1269.

102. ROY, S., and BANERJEE, S. An efficient 3D IC partitioning approach using satin bowerbird optimization for reduced TSV count and improved heat dissipation. In: *Engineering Research Express*. 2023. Vol. 5, no. 4, p. 045053. ISSN 2631-8695.

103. ROY, S., and BANERJEE, S. Genetic Algorithm Based 3D IC Partitioning Approach for TSV Minimization and Efficient Layer Assignment. In: *IETE Journal of Research*. 2024. Vol. 70, no. 7, p. 6401–6411. ISSN 0377-2063, 0974-780X.

104. DA LUZ REIS, R.A. EDA: Overview and Some Trends. In: *Journal of Integrated Circuits and Systems*. 2022. Vol. 17, no. 3, p. 1–10. ISSN 1872-0234, 1807-1953.

105. HUANG, G., HU, J., HE, Y., LIU, J., MA, M., SHEN, Z., WU, J., XU, Y., ZHANG, H., ZHONG, K., NING, X., MA, Y., YANG, H., YU, B., YANG, H., and WANG, Y. Machine Learning for Electronic Design Automation: A Survey. In: *ACM Transactions on Design Automation of Electronic Systems*. 2021. Vol. 26, no. 5, p. 1–46. ISSN 1084-4309, 1557-7309.

106. CHENG, R., YIN, L.-Z., JIANG, Z.-H., and XU, X.-M. Gate-Level Circuit Partitioning Algorithm Based on Clustering and an Improved Genetic Algorithm. In: *Entropy*. 2023. Vol. 25, no. 4, p. 597. ISSN 1099-4300.

107. TAMIR, A., SALEM, M., LIN, J., ALASAD, Q., and YUAN, J. Multi-Tier 3D IC Physical Design with Analytical Quadratic Partitioning Algorithm Using 2D P&R Tool. In: *Electronics*. 2021. Vol. 10, no. 16, p. 1930. ISSN 2079-9292.

108. KIM, K.M., SINHA, S., CLINE, B., YERIC, G., and LIM, S.K. Four-tier Monolithic 3D ICs: Tier Partitioning Methodology and Power Benefit Study. In: *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. 2016. p. 70–75. ISBN 978-1-4503-4185-1.

109. YIN, L., HU, H., and LI, C. Research on Circuit Partitioning Algorithm Based on Partition Connectivity Clustering and Tabu Search. In: *Technologies*. 2025. Vol. 13, no. 2, p. 81. ISSN 2227-7080.

110. CHANG, K., SINHA, S., CLINE, B., SOUTHERLAND, R., DOHERTY, M., YERIC, G., and LIM, S.K. Cascade2D: A design-aware partitioning approach to monolithic 3D IC with 2D

commercial tools. In: *Proceedings of the 35th International Conference on Computer-Aided Design*. 2016. p. 1–8. ISBN 978-1-4503-4466-1.

111. KIM, D., KIM, M., HUR, J., LEE, J., CHO, J., and KANG, S. TA3D: Timing-Aware 3D IC Partitioning and Placement by Optimizing the Critical Path. In: *Proceedings of the 2024 ACM/IEEE International Symposium on Machine Learning for CAD*. 2024. p. 1–7. ISBN 979-8-4007-0699-8.

112. **PLETEA, I.-M.**, SONTEA, V., and ALECSANDRESCU, I. CTS OPTIMIZATION ON 3D INTEGRATION. In: *Conferința Internațională "Telecommunicații, Electronică și Informatică.* 2019. Chișinău, Republica Moldova. p. 62. ISBN 978-9975-108-84-3.

113. **PLETEA, I.-M.**, ŞONTEA, V., and COJOCARU, V. Impactul conexiunilor verticale asupra ariei in integrarea 3D. In: *Conferința Internațională "Telecommunicații, Electronică și Informatică* . 2018. Chișinău, Republica Moldova. pp. 375-377. ISBN 978-9975-45-540-4.

114. **PLETEA, I.-M**., Sontea, V. IMPACTUL INTEGRARII 3D ASUPRA CONSTRUIRII ARBORELUI DE CLOCK. In: *Conferința tehnico-ştiinţifică a studenţilor, masteranzilor şi doctoranzilor*. 2019. Chișinău, Republica Moldova. Vol. 1, pp. 397-399. ISBN 978-9975-45-588-6.

115. **PLETEA, I.**, WURMAN, Z., OR-BACH, Z., and SONTEA, V. Monolithic 3D layout using 2D EDA for embedded memory-rich designs. In: *2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. 2015. p. 1–2. ISBN 978-1-5090-0259-7.

116. **PLETEA, I.-M**. METHODOLOGY AND BACKEND FLOW OPTIMIZATION FOR 3D. In: *JOURNAL OF ENGINEERING SCIENCE*. 2024. Vol. 31, no. 2, p. 39–47. ISSN 2587-3482, 2587-3474.

117. SHAH, M., BARREH, J., BROOKS, J., GOLLA, R., GROHOSKI, G., GURA, N., HETHERINGTON, R., JORDAN, P., LUTTRELL, M., OLSON, C., SAHA, B., SHEAHAN, D., SPRACKLEN, L., WYNN, A., and MICROSYSTEMS, S. UltraSPARC T2: A Highly-Threaded, Power- Efficient, SPARC SOC. In: *2007 IEEE Asian Solid-State Circuits Conference*. 2007. ISBN:978-1-4244-1359-1, 978-1-4244-1360-7.

118. ZHAO, Z., ZHANG, S., LIU, G., FENG, C., YANG, T., HAN, A., and WANG, L. Machine-Learning-Based Multi-Corner Timing Prediction for Faster Timing Closure. In: *Electronics*. 2022. Vol. 11, no. 10, p. 1571. ISSN 2079-9292.

119. ISLAM, R. Feasibility Prediction for Rapid IC Design Space Exploration. In: *Electronics*. 2022. Vol. 11, no. 7, p. 1161. ISSN 2079-9292.

120. GUPTA, P., and KAHNG, A.B. Manufacturing-aware physical design. In: *ICCAD-2003. International Conference on Computer Aided Design (IEEE Cat. No.03CH37486)*. 2003. p. 681–687. ISBN 978-1-58113-762-0.

121. BOBBA, S., CHAKRABORTY, A., THOMAS, O., BATUDE, P., ERNST, T., FAYNOT, O., PAN, D.Z., and DE MICHELI, G. CELONCEL: Effective design technique for 3-D monolithic integration targeting high performance integrated circuits. In: *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*. 2011. p. 336–343. ISBN 978-1-4244-7515-5.

122. PANTH, S., SAMADI, K., DU, Y., and LIM, S.K. Shrunk-2-D: A Physical Design Methodology to Build Commercial-Quality Monolithic 3-D ICs. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2017. Vol. 36, no. 10, p. 1716–1724. ISSN 0278-0070, 1937-4151.

123. **Pletea, I.M.**, Șontea Victor. Design and Partitioning Across 2 Strata în 3D IC architectures. In: *International Conference on Electronics, Communications and Computing, IC ECCO 2024*. Chișinău, Republica Moldova p. 107-108. ISBN 978-9975-64-480-8.

124. KU, B.W., CHANG, K., and LIM, S.K. Compact-2D: A Physical Design Methodology to Build Two-Tier Gate-Level 3-D ICs. In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2020. Vol. 39, no. 6, p. 1151–1164. ISSN 0278-0070, 1937-4151.

125. LOK, M., HE, K., MANI, M., CARAMANIS, C., and ORSHANSKY, M. Design of power-optimal buffers tunable to process variability. In : *2010 IEEE Dallas Circuits and Systems Workshop*. 2010. p. 1–4. ISBN 978-1-4244-9535-1.

126. WANG, W., PAVLIDIS, V.F., and CHENG, Y. Zero-skew Clock Network Synthesis for Monolithic 3D ICs with Minimum Wirelength. In : *Proceedings of the 2020 on Great Lakes Symposium on VLSI*. 2020. p. 399–404. ISBN 978-1-4503-7944-1.

127. DAS, S., RIEDEL, S., BERTULETTI, M., BENINI, L., BRUNION, M., RYCKAERT, J., MYERS, J., BISWAS, D., and MILOJEVIC, D. 3D Partitioning with Pipeline Optimization for Low-Latency Memory Access in Many-Core SoCs. In : *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2024. p. 1–5. ISBN 979-8-3503-3099-1.

128. **Pletea, I.M.**, Șontea Victor. Heterogenous integrations on 3d integrated circuits. In: *Materials Science and Condensed Matter Physics, dedicated to the 60th anniversary from the foundation of the Institute of Applied Physics*. Ediția 10. 2024. Chișinău, Republica Moldova. p. 146. ISBN 978-9975-62-763-4.

129. Guobiao Zhang, Neural network processor containing three-dimensional longitudinal storage array, cerere de brevet, CN110414676A, 2019.

130. ZHAO, Y., MA, S., LIU, H., HUANG, L., and DAI, Y. SAC: An Ultra-Efficient Spin-based Architecture for Compressed DNNs. In : *ACM Transactions on Architecture and Code Optimization*. 2024. Vol. 21, no. 1, p. 1–26. ISSN 1544-3566, 1544-3973.

131. SHUKLA, P., PAVLIDIS, V.F., SALMAN, E., and COSKUN, A.K. *A New Dataflow Implementation to Improve Energy Efficiency of Monolithic 3D Systolic Arrays*. 2024. arXiv:2401.03585.

132. HAPPEL, B.L.M., and MURRE, J.M.J. Design and evolution of modular neural network architectures. In: *Neural Networks*. 1994. Vol. 7, no. 6–7, p. 985–1004. ISSN 08936080.

133. NAWROCKI, R.A., VOYLES, R.M., and SHAHEEN, S.E. A Mini Review of Neuromorphic Architectures and Implementations. In: *IEEE Transactions on Electron Devices*. 2016. Vol. 63, no. 10, p. 3819–3829. ISSN 0018-9383, 1557-9646.

134. WANG, Z., and YAO, X. Self-aware Hierarchical Neuromorphic Architecture. In: *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024. p. 1–10. ISBN 979-8-3503-5931-2.

135. AL ABDUL WAHID, S., ASAD, A., and MOHAMMADI, F. A Survey on Neuromorphic Architectures for Running Artificial Intelligence Algorithms. In: *Electronics*. 2024. Vol. 13, no. 15, p. 2963. ISSN 2079-9292.

136. MADDEN, L., WU, E., KIM, N., BANIJAMALI, B., ABUGHARBIEH, K., RAMALINGAM, S., and WU, X. Advancing high performance heterogeneous integration through die stacking. In: *2012 Proceedings of the ESSCIRC (ESSCIRC)*. 2012. p. 18–24. ISBN 978-1-4673-2213-3.

137. MonolithIC 3D. [citat 12.10.2016]. Disponibil: https://www.monolithic3d.com/fpga.html.

138. FAROOQ, U., BAIG, I., BHATTI, M.K., MEHREZ, H., KUMAR, A., and GUPTA, M. Prototyping using multi-FPGA platform: A novel and complete flow. In: *Microprocessors and Microsystems*. 2023. Vol. 96, p. 104751. ISSN 01419331.

139. LEE, Y.S., KIM, K.M., LEE, J.H., GONG, Y.-H., KIM, S.W., and CHUNG, S.W. Monolithic 3D stacked multiply-accumulate units. In: *Integration*. 2021. Vol. 76, p. 183–189. ISSN 01679260.

140. **PLETEA, I.,** ALECSANDRESCU, I., LUCANU, N., AGHION, C., BARABASA, C., ZBANCIOC, M., HĂGAN, M., and SONTEA, V. Neural Grain implemented in monolithic 3D technology. In : *IEEE 2025 International Symposium on Signals, Circuits and Systems*. ISSN 2995-0236.

141. DESHPANDE, N. A Review on ASIC Synthesis Flow Employing Two Industry Standard Tools. In: *International Journal of Engineering Research*. 2020. Vol. 8, no. 17.

ANEXA 1. Script pentru rularea procesului de lucru 2D

```
source -verbose ./scripts/setup_icc.tcl

set VER 2D_01_25Apr2025

#set VLG_FILE "/home/ionica/work/synth/OpenSPARCT2.1.3-saed32/design_spc_rvt/sys/iop/spc/lsu/synopsys/gate/lsu.flat.high_efort.vSyn"

################################################################################################
################################## Design evaluation ###########################################
################################################################################################
	if { [file exists ${MW}/${DESIGN}_lib] } { file delete -force ${MW}/${DESIGN}_lib }
	create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
		-mw_reference_library "$SC_MW $MEM_MW" ${MW}/${DESIGN}_lib
	open_mw_lib ${MW}/${DESIGN}_lib

	read_verilog $VLG_FILE
	#reset_upf
	#load_upf $UPF_FILE
	current_design $DESIGN
	link -force

	create_scenario func
	set_tlu_plus_files -max_tluplus $MAX_TLUPLUS -min_tluplus $MIN_TLUPLUS
	set_operating_conditions -library saed32rvt_ss0p95v125c ss0p95v125c
	#read_sdc $SDC_FILE
	source $SDC_FILE

	derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;# -create_ports all
	set_app_var physopt_delete_unloaded_cells false
	set_dont_touch [get_flat_cells UPD_*]

	report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.preFlp.tm.rpt

	#MW
	#create_floorplan -control_type width_and_height -core_width 680 -core_height 680 \
	#		-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
	#		-start_first_row -flip_first_row
	#MW1
	#create_floorplan -control_type width_and_height -core_width 823 -core_height 823 \
	#		-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
	#		-start_first_row -flip_first_row
	#MW2
	create_floorplan -control_type width_and_height -core_width 20.5 -core_height 20.5 \
			-left_io2core 0.2 -bottom_io2core 0.2 -right_io2core 0.2 -top_io2core 0.2 \
			-start_first_row -flip_first_row
```

```
        foreach layer {M7 M8 M9 MRDL} { create_route_guide -name ${layer}_blk -
no_signal_layers $layer -coordinate "[get_attribute [get_die_area] bbox]" }
        ##I.P routing only layers from M1 to M7
        #set_ignored_layers -min_routing_layer M1 -max_routing_layer M7
        #set_net_routing_layer_constraints  {*}  -min_layer_name M1  -max_layer_name M7

        #For create_fp_placement & legalize_fp_placement
        # 30um mem spacing based on area reduction + 5um hallo
        set_fp_placement_strategy -min_distance_between_macros 5.000 \
                        -macros_min_distance_applies_to [get_cells -hier -filter
is_hard_macro==true]
        set_fp_placement_strategy -macro_orientation all S
        set_keepout_margin -all_macros -type hard -outer {5.000 5.000 5.000 5.000}

        set_fp_placement_strategy -flip_chip on
        set_fp_macro_options [get_cells *mem*] -legal_orientations {S}

        #set_fp_macro_options dta/lsu_mem5 -legal_orientations {S}
        #set_fp_macro_options lru/lsu_mem7 -legal_orientations {S}
        #set_fp_macro_options dva/lsu_mem7 -legal_orientations {S}
        #set_fp_macro_options dta/lsu_mem6 -legal_orientations {S}

        #set_undoable_attribute [get_cells -all dta/lsu_mem5] orientation {S}
        #set_undoable_attribute [get_cells -all lru/lsu_mem7] orientation {S}
        #set_undoable_attribute [get_cells -all dva/lsu_mem7] orientation {S}
        #set_undoable_attribute [get_cells -all dta/lsu_mem6] orientation {S}

        # -virtual_IPO true
        # -congestion_effort medium
        # -macros_min_distance_applies_to [get_flat_cells -filter is_hard_macro==true]
        # -macros_on_edge on
        # -pin_routing_aware true
        # set_fp_macro_options $macros -legal_orientations "N W S E FN FW FS FE" -
side_channel {left right top bottom}
        # set_keepout_margin -type hard -outer {56.000 56.000 56.000 56.00} [get_cells -filter
is_hard_macro==true]
        # set place_max_cell_density_threshold 0.7
        # create_fp_placement  # Virtual flat placement
        create_fp_placement -optimize_pins -congestion_driven -timing_driven
        ##-incremental all
        set_attribute [get_ports ] is_fixed true -quiet

        read_def GN_4x4.modified.v1.def

        save_mw_cel ${DESIGN}
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.blkPlace.tm.rpt

        close_mw_cel
        close_mw_lib
```

```
#############################################################################################
################################### Design placement ########################################
#############################################################################################
        if { [file exists ${MW}/${DESIGN}_lib${VER}_place] } { file delete -force
${MW}/${DESIGN}_lib${VER}_place }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW"
${MW}/${DESIGN}_lib${VER}_place
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib${VER}_place
        open_mw_lib ${MW}/${DESIGN}_lib${VER}_place
        open_mw_cel ${DESIGN}

        current_design $DESIGN
        link -force
        derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;#  -
create_ports all
        set_app_var physopt_delete_unloaded_cells false
        set_dont_touch [get_flat_cells UPD_*]
        set_attribute [get_flat_cells -filter is_hard_macro==true] is_fixed true
        # set_congestion_options
        # create_placement     # Create rough placement
        create_placement -congestion -congestion_effort medium -timing_driven ;# -
continue_on_missing_scandef
        # legalize_placement   # Legalize placement

        # place_fp_pins        # Auto Pin Assignment
        # place_opt           # Run place_opt
        place_opt -skip_initial_placement -congestion

        save_mw_cel ${DESIGN}
        report_placement_utilization      > ${REPORTS}/${DESIGN}.${VER}.place.util.rpt
        report_qor                        > ${REPORTS}/${DESIGN}.${VER}.place.qor.rpt
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.place.tm.rpt

        close_mw_cel
        close_mw_lib

#############################################################################################
######################################### CTS ###############################################
#############################################################################################
        if { [file exists ${MW}/${DESIGN}_lib${VER}_cts] } { file delete -force
${MW}/${DESIGN}_lib${VER}_cts }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW"
${MW}/${DESIGN}_lib${VER}_cts
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib${VER}_place -from ${DESIGN}
\
                -to_library ${MW}/${DESIGN}_lib${VER}_cts
        open_mw_lib ${MW}/${DESIGN}_lib${VER}_cts
```

```
        open_mw_cel ${DESIGN}

        current_design $DESIGN
        link -force
        derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;# -
create_ports all
        set_app_var cts_instance_name_prefix CLK_
        #set_clock_tree_references -references {saed32rvt_ss0p95v125c/NBUFFX8_RVT
saed32rvt_ss0p95v125c/NBUFFX4_RVT saed32rvt_ss0p95v125c/NBUFFX32_RVT
saed32rvt_ss0p95v125c/NBUFFX16_RVT saed32rvt_ss0p95v125c/NBUFFX2_RVT
saed32rvt_ss0p95v125c/INVX8_RVT saed32rvt_ss0p95v125c/INVX4_RVT
saed32rvt_ss0p95v125c/INVX32_RVT saed32rvt_ss0p95v125c/INVX2_RVT
saed32rvt_ss0p95v125c/INVX1_RVT saed32rvt_ss0p95v125c/INVX16_RVT
saed32rvt_ss0p95v125c/INVX0_RVT }
        #set_clock_tree_options -clock_trees l2clk -target_skew 0.0100 -max_transition 0.2
        clock_opt
        save_mw_cel ${DESIGN}
        report_placement_utilization > ${REPORTS}/${DESIGN}.${VER}.CTS.util.rpt
        report_qor > ${REPORTS}/${DESIGN}.${VER}.CTS.qor.rpt
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.CTS.tm.rpt
        close_mw_cel
        close_mw_lib
##########################################################################################
########################################## Route ##########################################
##########################################################################################
        if { [file exists ${MW}/${DESIGN}_lib${VER}_route] } { file delete -force
${MW}/${DESIGN}_lib${VER}_route }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW"
${MW}/${DESIGN}_lib${VER}_route
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib${VER}_cts -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib${VER}_route
        open_mw_lib ${MW}/${DESIGN}_lib${VER}_route
        open_mw_cel ${DESIGN}

        current_design $DESIGN
        link -force
        derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;# -
create_ports all
        # set_route_zrt_common_options
        # set_route_zrt_detail_options

        route_opt -initial_route_only
        save_mw_cel
        route_opt -skip_initial_route ;# -effort low

        save_mw_cel ${DESIGN}
        verify_zrt_route
        verify_lvs -check_open_locator -check_short_locator

        report_placement_utilization > ${REPORTS}/${DESIGN}.${VER}.Route.util.rpt
```

```
report_qor                    > ${REPORTS}/${DESIGN}.${VER}.Route.qor.rpt
report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.Route.tm.rpt
#close_mw_cel
#close_mw_lib
```

ANEXA 2. Fișierul LEF utilizat la generarea memoriei SRAM_128x38slp în formă de serpentină

```
#I Remove polygon from OBS keep only RECT on M1-M6
#I Change DIRECTION from INOUT to correct one, based on liberty file
#I Comment out VDDM, VDDL, VDDB
#I Rename VDDH into VDD
#I Change CE pin's name to CLK according to liberty file
#I Add CLASS COVER
#I Add pins LS, DS, SD on M1, following direction from liberty file
#I Move all signal (non-supply) pins to M7
#C Comment out all layers in OBS and add OverlapCheck layer


VERSION 5.7 ;
BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;

MACRO SRAM_128x38slp
  CLASS COVER ;
  ORIGIN 3.534 45.561 ;
  FOREIGN SRAM_128x38slp -3.534 -45.561 ;
  SIZE 88.223 BY 260.186 ;

   PIN LS
     DIRECTION INPUT ;
     USE SIGNAL ;
     PORT
        LAYER M7 ;
         RECT 56.282 -42.321 56.366 -42.256 ;
     END
  END LS
  PIN DS
     DIRECTION INPUT ;
     USE SIGNAL ;
     PORT
        LAYER M7 ;
         RECT 56.482 -42.321 56.566 -42.256 ;
     END
  END DS
  PIN SD
     DIRECTION INPUT ;
     USE SIGNAL ;
     PORT
        LAYER M7 ;
         RECT 56.682 -42.321 56.766 -42.256 ;
     END
  END SD
```

```
PIN WEB
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 84.376 -34.758 84.689 -34.698 ;
  END
END WEB
PIN O[33]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 46.41 -45.561 46.494 -45.496 ;
  END
END O[33]
PIN O[1]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 2.634 -45.561 2.718 -45.496 ;
  END
END O[1]
PIN O[14]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 20.418 -45.561 20.502 -45.496 ;
  END
END O[14]
PIN I[19]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 26.911 -45.561 26.971 -45.511 ;
  END
END I[19]
PIN A[5]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 84.577 152.082 84.689 152.182 ;
  END
END A[5]
PIN O[2]
  DIRECTION OUTPUT ;
```

```
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 4.002 -45.561 4.086 -45.496 ;
  END
END O[2]
PIN O[12]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 17.682 -45.561 17.766 -45.496 ;
  END
END O[12]
PIN I[23]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 32.383 -45.561 32.443 -45.511 ;
  END
END I[23]
PIN O[23]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 32.73 -45.561 32.814 -45.496 ;
  END
END O[23]
PIN I[7]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 10.495 -45.561 10.555 -45.511 ;
  END
END I[7]
PIN O[28]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 39.57 -45.561 39.654 -45.496 ;
  END
END O[28]
PIN O[15]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
```

```
      LAYER M7 ;
        RECT 21.786 -45.561 21.87 -45.496 ;
    END
  END O[15]
  PIN I[11]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 15.967 -45.561 16.027 -45.511 ;
    END
  END I[11]
#  PIN CE
    PIN CLK
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 84.55 -27.061 84.689 -26.955 ;
    END
  END CLK
#  END CE
  PIN O[30]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 42.306 -45.561 42.39 -45.496 ;
    END
  END O[30]
  PIN O[5]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 8.106 -45.561 8.19 -45.496 ;
    END
  END O[5]
  PIN O[6]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 9.474 -45.561 9.558 -45.496 ;
    END
  END O[6]
  PIN O[26]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
```

```
    LAYER M7 ;
      RECT 36.834 -45.561 36.918 -45.496 ;
    END
  END O[26]
  PIN I[17]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 24.175 -45.561 24.235 -45.511 ;
    END
  END I[17]
  PIN O[18]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 25.89 -45.561 25.974 -45.496 ;
    END
  END O[18]
  PIN I[22]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 31.015 -45.561 31.075 -45.511 ;
    END
  END I[22]
  PIN O[16]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 23.154 -45.561 23.238 -45.496 ;
    END
  END O[16]
  PIN O[19]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 27.258 -45.561 27.342 -45.496 ;
    END
  END O[19]
  PIN CSB
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 84.55 -27.769 84.689 -27.663 ;
```

```
  END
END CSB
PIN O[31]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 43.674 -45.561 43.758 -45.496 ;
  END
END O[31]
PIN I[36]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 50.167 -45.561 50.227 -45.511 ;
  END
END I[36]
PIN I[35]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 48.799 -45.561 48.859 -45.511 ;
  END
END I[35]
PIN O[8]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 12.21 -45.561 12.294 -45.496 ;
  END
END O[8]
PIN OEB
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 84.629 -43.352 84.689 -43.292 ;
  END
END OEB
PIN I[3]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 5.023 -45.561 5.083 -45.511 ;
  END
END I[3]
```

```
  PIN I[2]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 3.655 -45.561 3.715 -45.511 ;
    END
  END I[2]
  PIN O[29]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 40.938 -45.561 41.022 -45.496 ;
    END
  END O[29]
  PIN I[12]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 17.335 -45.561 17.395 -45.511 ;
    END
  END I[12]
  PIN O[11]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 16.314 -45.561 16.398 -45.496 ;
    END
  END O[11]
#  PIN VDDB
#    DIRECTION INPUT ;
#    USE POWER ;
#    PORT
#      LAYER M5 ;
#        RECT 79.387 -43.561 79.687 -43.261 ;
#        RECT 78.937 -43.561 79.237 -43.261 ;
#        RECT 78.487 -43.561 78.787 -43.261 ;
#        RECT 78.037 -43.561 78.337 -43.261 ;
#        RECT 77.587 -43.561 77.887 -43.261 ;
#        RECT 77.137 -43.561 77.437 -43.261 ;
#        RECT 76.687 -43.561 76.987 -43.261 ;
#        RECT 76.237 -43.561 76.537 -43.261 ;
#        RECT 75.787 -43.561 76.087 -43.261 ;
#        RECT 75.337 -43.561 75.637 -43.261 ;
#        RECT 73.087 -43.561 73.387 -43.261 ;
#        RECT 72.187 -43.561 72.487 -43.261 ;
#        RECT 71.737 -43.561 72.037 -43.261 ;
```

```
#       RECT 71.287 -43.561 71.587 -43.261 ;
#       RECT 70.837 -43.561 71.137 -43.261 ;
#       RECT 70.387 -43.561 70.687 -43.261 ;
#       RECT 69.937 -43.561 70.237 -43.261 ;
#       RECT 69.487 -43.561 69.787 -43.261 ;
#       RECT 69.037 -43.561 69.337 -43.261 ;
#       RECT 68.587 -43.561 68.887 -43.261 ;
#       RECT 68.137 -43.561 68.437 -43.261 ;
#       RECT 67.687 -43.561 67.987 -43.261 ;
#       RECT 67.237 -43.561 67.537 -43.261 ;
#       RECT 66.787 -43.561 67.087 -43.261 ;
#       RECT 65.887 -43.561 66.187 -43.261 ;
#       RECT 65.437 -43.561 65.737 -43.261 ;
#       RECT 64.987 -43.561 65.287 -43.261 ;
#       RECT 64.537 -43.561 64.837 -43.261 ;
#       RECT 64.087 -43.561 64.387 -43.261 ;
#       RECT 63.637 -43.561 63.937 -43.261 ;
#       RECT 62.737 -43.561 63.037 -43.261 ;
#       RECT 62.287 -43.561 62.587 -43.261 ;
#       RECT 61.837 -43.561 62.137 -43.261 ;
#       RECT 61.387 -43.561 61.687 -43.261 ;
#       RECT 60.937 -43.561 61.237 -43.261 ;
#       RECT 60.487 -43.561 60.787 -43.261 ;
#       RECT 60.037 -43.561 60.337 -43.261 ;
#       RECT 59.137 -43.561 59.437 -43.261 ;
#       RECT 58.237 -43.561 58.537 -43.261 ;
#       RECT 57.337 -43.561 57.637 -43.261 ;
#       RECT 53.737 -43.561 54.037 -43.261 ;
#       RECT 3.787 -43.561 4.087 -43.261 ;
#       RECT 3.337 -43.561 3.637 -43.261 ;
#       RECT 2.887 -43.561 3.187 -43.261 ;
#       RECT 2.437 -43.561 2.737 -43.261 ;
#       RECT 1.987 -43.561 2.287 -43.261 ;
#       RECT 1.537 -43.561 1.837 -43.261 ;
#       RECT 1.087 -43.561 1.387 -43.261 ;
#     END
#   END VDDB
    PIN O[4]
      DIRECTION OUTPUT ;
      USE SIGNAL ;
      PORT
        LAYER M7 ;
          RECT 6.738 -45.561 6.822 -45.496 ;
      END
    END O[4]
    PIN I[37]
      DIRECTION INPUT ;
      USE SIGNAL ;
      PORT
        LAYER M7 ;
```

```
      RECT 51.535 -45.561 51.595 -45.511 ;
    END
  END I[37]
  PIN O[25]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 35.466 -45.561 35.55 -45.496 ;
    END
  END O[25]
  PIN I[6]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 9.127 -45.561 9.187 -45.511 ;
    END
  END I[6]
  PIN A[0]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 84.577 177.187 84.689 177.287 ;
    END
  END A[0]
  PIN O[32]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 45.042 -45.561 45.126 -45.496 ;
    END
  END O[32]
  PIN I[20]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 28.279 -45.561 28.339 -45.511 ;
    END
  END I[20]
  PIN O[20]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 28.626 -45.561 28.71 -45.496 ;
    END
```

```
END O[20]
PIN VSS
  DIRECTION INPUT ;
  USE GROUND ;
  PORT
    LAYER M5 ;
      RECT 80.737 -43.561 81.037 -43.261 ;
      RECT 80.287 -43.561 80.587 -43.261 ;
      RECT 79.837 -43.561 80.137 -43.261 ;
      RECT 66.337 -43.561 66.637 -43.261 ;
      RECT 63.187 -43.561 63.487 -43.261 ;
      RECT 59.587 -43.561 59.887 -43.261 ;
      RECT 58.687 -43.561 58.987 -43.261 ;
      RECT 57.787 -43.561 58.087 -43.261 ;
      RECT 56.887 -43.561 57.187 -43.261 ;
      RECT 56.437 -43.561 56.737 -43.261 ;
      RECT 55.987 -43.561 56.287 -43.261 ;
      RECT 55.537 -43.561 55.837 -43.261 ;
      RECT 55.087 -43.561 55.387 -43.261 ;
      RECT 54.637 -43.561 54.937 -43.261 ;
      RECT 54.187 -43.561 54.487 -43.261 ;
      RECT 53.287 -43.561 53.587 -43.261 ;
      RECT 52.837 -43.561 53.137 -43.261 ;
      RECT 52.387 -43.561 52.687 -43.261 ;
      RECT 51.937 -43.561 52.237 -43.261 ;
      RECT 51.487 -43.561 51.787 -43.261 ;
      RECT 51.037 -43.561 51.337 -43.261 ;
      RECT 50.587 -43.561 50.887 -43.261 ;
      RECT 50.137 -43.561 50.437 -43.261 ;
      RECT 49.687 -43.561 49.987 -43.261 ;
      RECT 49.237 -43.561 49.537 -43.261 ;
      RECT 48.787 -43.561 49.087 -43.261 ;
      RECT 48.337 -43.561 48.637 -43.261 ;
      RECT 47.887 -43.561 48.187 -43.261 ;
      RECT 47.437 -43.561 47.737 -43.261 ;
      RECT 46.987 -43.561 47.287 -43.261 ;
      RECT 46.537 -43.561 46.837 -43.261 ;
      RECT 46.087 -43.561 46.387 -43.261 ;
      RECT 45.637 -43.561 45.937 -43.261 ;
      RECT 45.187 -43.561 45.487 -43.261 ;
      RECT 44.737 -43.561 45.037 -43.261 ;
      RECT 44.287 -43.561 44.587 -43.261 ;
      RECT 43.837 -43.561 44.137 -43.261 ;
      RECT 43.387 -43.561 43.687 -43.261 ;
      RECT 42.937 -43.561 43.237 -43.261 ;
      RECT 42.487 -43.561 42.787 -43.261 ;
      RECT 42.037 -43.561 42.337 -43.261 ;
      RECT 41.587 -43.561 41.887 -43.261 ;
      RECT 41.137 -43.561 41.437 -43.261 ;
      RECT 40.687 -43.561 40.987 -43.261 ;
```

RECT 40.237 -43.561 40.537 -43.261 ;
RECT 39.787 -43.561 40.087 -43.261 ;
RECT 39.337 -43.561 39.637 -43.261 ;
RECT 38.887 -43.561 39.187 -43.261 ;
RECT 38.437 -43.561 38.737 -43.261 ;
RECT 37.987 -43.561 38.287 -43.261 ;
RECT 37.537 -43.561 37.837 -43.261 ;
RECT 37.087 -43.561 37.387 -43.261 ;
RECT 36.637 -43.561 36.937 -43.261 ;
RECT 36.187 -43.561 36.487 -43.261 ;
RECT 35.737 -43.561 36.037 -43.261 ;
RECT 35.287 -43.561 35.587 -43.261 ;
RECT 34.837 -43.561 35.137 -43.261 ;
RECT 34.387 -43.561 34.687 -43.261 ;
RECT 33.937 -43.561 34.237 -43.261 ;
RECT 33.487 -43.561 33.787 -43.261 ;
RECT 33.037 -43.561 33.337 -43.261 ;
RECT 32.587 -43.561 32.887 -43.261 ;
RECT 32.137 -43.561 32.437 -43.261 ;
RECT 31.687 -43.561 31.987 -43.261 ;
RECT 31.237 -43.561 31.537 -43.261 ;
RECT 30.787 -43.561 31.087 -43.261 ;
RECT 30.337 -43.561 30.637 -43.261 ;
RECT 29.887 -43.561 30.187 -43.261 ;
RECT 29.437 -43.561 29.737 -43.261 ;
RECT 28.987 -43.561 29.287 -43.261 ;
RECT 28.537 -43.561 28.837 -43.261 ;
RECT 28.087 -43.561 28.387 -43.261 ;
RECT 27.637 -43.561 27.937 -43.261 ;
RECT 27.187 -43.561 27.487 -43.261 ;
RECT 26.737 -43.561 27.037 -43.261 ;
RECT 26.287 -43.561 26.587 -43.261 ;
RECT 25.837 -43.561 26.137 -43.261 ;
RECT 25.387 -43.561 25.687 -43.261 ;
RECT 24.937 -43.561 25.237 -43.261 ;
RECT 24.487 -43.561 24.787 -43.261 ;
RECT 24.037 -43.561 24.337 -43.261 ;
RECT 23.587 -43.561 23.887 -43.261 ;
RECT 23.137 -43.561 23.437 -43.261 ;
RECT 22.687 -43.561 22.987 -43.261 ;
RECT 22.237 -43.561 22.537 -43.261 ;
RECT 21.787 -43.561 22.087 -43.261 ;
RECT 21.337 -43.561 21.637 -43.261 ;
RECT 20.887 -43.561 21.187 -43.261 ;
RECT 20.437 -43.561 20.737 -43.261 ;
RECT 19.987 -43.561 20.287 -43.261 ;
RECT 19.537 -43.561 19.837 -43.261 ;
RECT 19.087 -43.561 19.387 -43.261 ;
RECT 18.637 -43.561 18.937 -43.261 ;
RECT 18.187 -43.561 18.487 -43.261 ;

```
      RECT 17.737 -43.561 18.037 -43.261 ;
      RECT 17.287 -43.561 17.587 -43.261 ;
      RECT 16.837 -43.561 17.137 -43.261 ;
      RECT 16.387 -43.561 16.687 -43.261 ;
      RECT 15.937 -43.561 16.237 -43.261 ;
      RECT 15.487 -43.561 15.787 -43.261 ;
      RECT 15.037 -43.561 15.337 -43.261 ;
      RECT 14.587 -43.561 14.887 -43.261 ;
      RECT 14.137 -43.561 14.437 -43.261 ;
      RECT 13.687 -43.561 13.987 -43.261 ;
      RECT 13.237 -43.561 13.537 -43.261 ;
      RECT 12.787 -43.561 13.087 -43.261 ;
      RECT 12.337 -43.561 12.637 -43.261 ;
      RECT 11.887 -43.561 12.187 -43.261 ;
      RECT 11.437 -43.561 11.737 -43.261 ;
      RECT 10.987 -43.561 11.287 -43.261 ;
      RECT 4.237 -43.561 4.537 -43.261 ;
      RECT 0.637 -43.561 0.937 -43.261 ;
      RECT -0.263 -43.561 0.037 -43.261 ;
      RECT -0.713 -43.561 -0.413 -43.261 ;
      RECT -1.163 -43.561 -0.863 -43.261 ;
      RECT -1.613 -43.561 -1.313 -43.261 ;
      RECT -2.963 -43.561 -2.663 -43.261 ;
  END
END VSS
PIN I[14]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 20.071 -45.561 20.131 -45.511 ;
  END
END I[14]
PIN A[6]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 84.577 150.542 84.689 150.642 ;
  END
END A[6]
PIN I[26]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 36.487 -45.561 36.547 -45.511 ;
  END
END I[26]
PIN O[24]
```

```
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 34.098 -45.561 34.182 -45.496 ;
  END
END O[24]
PIN O[35]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 49.146 -45.561 49.23 -45.496 ;
  END
END O[35]
PIN I[16]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 22.807 -45.561 22.867 -45.511 ;
  END
END I[16]
PIN I[31]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 43.327 -45.561 43.387 -45.511 ;
  END
END I[31]
PIN I[32]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 44.695 -45.561 44.755 -45.511 ;
  END
END I[32]
PIN I[1]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 2.287 -45.561 2.347 -45.511 ;
  END
END I[1]
PIN O[9]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
```

```
  PORT
    LAYER M7 ;
      RECT 13.578 -45.561 13.662 -45.496 ;
  END
END O[9]
PIN I[21]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 29.647 -45.561 29.707 -45.511 ;
  END
END I[21]
PIN O[13]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 19.05 -45.561 19.134 -45.496 ;
  END
END O[13]
PIN O[22]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 31.362 -45.561 31.446 -45.496 ;
  END
END O[22]
PIN A[2]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 84.577 168.182 84.689 168.282 ;
  END
END A[2]
PIN O[37]
  DIRECTION OUTPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
      RECT 51.882 -45.561 51.966 -45.496 ;
  END
END O[37]
PIN I[10]
  DIRECTION INPUT ;
  USE SIGNAL ;
  PORT
    LAYER M7 ;
```

```
        RECT 14.599 -45.561 14.659 -45.511 ;
      END
    END I[10]
    PIN I[24]
      DIRECTION INPUT ;
      USE SIGNAL ;
      PORT
        LAYER M7 ;
          RECT 33.751 -45.561 33.811 -45.511 ;
      END
    END I[24]
    PIN I[8]
      DIRECTION INPUT ;
      USE SIGNAL ;
      PORT
        LAYER M7 ;
          RECT 11.863 -45.561 11.923 -45.511 ;
      END
    END I[8]
    PIN O[7]
      DIRECTION OUTPUT ;
      USE SIGNAL ;
      PORT
        LAYER M7 ;
          RECT 10.842 -45.561 10.926 -45.496 ;
      END
    END O[7]
  #  PIN VDDH
      PIN VDD
      DIRECTION INPUT ;
      USE POWER ;
      PORT
        LAYER M5 ;
          RECT 82.087 -43.561 82.387 -43.261 ;
          RECT 81.637 -43.561 81.937 -43.261 ;
          RECT 81.187 -43.561 81.487 -43.261 ;
          RECT 74.887 -43.561 75.187 -43.261 ;
          RECT 73.987 -43.561 74.287 -43.261 ;
          RECT 73.537 -43.561 73.837 -43.261 ;
          RECT 6.037 -43.561 6.337 -43.261 ;
          RECT 5.587 -43.561 5.887 -43.261 ;
          RECT 5.137 -43.561 5.437 -43.261 ;
          RECT -2.063 -43.561 -1.763 -43.261 ;
      END
    END VDD
  #  END VDDH
    PIN O[27]
      DIRECTION OUTPUT ;
      USE SIGNAL ;
      PORT
```

```
      LAYER M7 ;
        RECT 38.202 -45.561 38.286 -45.496 ;
    END
  END O[27]
  PIN O[34]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 47.778 -45.561 47.862 -45.496 ;
    END
  END O[34]
  PIN I[0]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 0.919 -45.561 0.979 -45.511 ;
    END
  END I[0]
  PIN I[30]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 41.959 -45.561 42.019 -45.511 ;
    END
  END I[30]
  PIN I[18]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 25.543 -45.561 25.603 -45.511 ;
    END
  END I[18]
  PIN I[5]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 7.759 -45.561 7.819 -45.511 ;
    END
  END I[5]
#  PIN VDDL
#    DIRECTION INPUT ;
#    USE POWER ;
#    PORT
#      LAYER M5 ;
#        RECT 74.437 -43.561 74.737 -43.261 ;
```

```
#       RECT 72.637 -43.561 72.937 -43.261 ;
#       RECT 4.687 -43.561 4.987 -43.261 ;
#       RECT -2.513 -43.561 -2.213 -43.261 ;
#    END
#  END VDDL
  PIN O[3]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 5.37 -45.561 5.454 -45.496 ;
    END
  END O[3]
  PIN I[34]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 47.431 -45.561 47.491 -45.511 ;
    END
  END I[34]
  PIN I[15]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 21.439 -45.561 21.499 -45.511 ;
    END
  END I[15]
  PIN A[1]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 84.577 169.722 84.689 169.822 ;
    END
  END A[1]
  PIN A[4]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 84.577 159.362 84.689 159.462 ;
    END
  END A[4]
  PIN I[9]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
```

```
      RECT 13.231 -45.561 13.291 -45.511 ;
    END
  END I[9]
  PIN O[0]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 1.266 -45.561 1.35 -45.496 ;
    END
  END O[0]
  PIN I[28]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 39.223 -45.561 39.283 -45.511 ;
    END
  END I[28]
#  PIN VDDM
#    DIRECTION INPUT ;
#    USE POWER ;
#    PORT
#      LAYER M5 ;
#        RECT 10.537 -43.561 10.837 -43.261 ;
#        RECT 10.087 -43.561 10.387 -43.261 ;
#        RECT 9.637 -43.561 9.937 -43.261 ;
#        RECT 9.187 -43.561 9.487 -43.261 ;
#        RECT 8.737 -43.561 9.037 -43.261 ;
#        RECT 8.287 -43.561 8.587 -43.261 ;
#        RECT 7.837 -43.561 8.137 -43.261 ;
#        RECT 7.387 -43.561 7.687 -43.261 ;
#        RECT 6.937 -43.561 7.237 -43.261 ;
#        RECT 6.487 -43.561 6.787 -43.261 ;
#        RECT 0.187 -43.561 0.487 -43.261 ;
#        RECT -3.413 -43.561 -3.113 -43.261 ;
#    END
#  END VDDM
  PIN O[17]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 24.522 -45.561 24.606 -45.496 ;
    END
  END O[17]
  PIN O[10]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
```

```
    LAYER M7 ;
      RECT 14.946 -45.561 15.03 -45.496 ;
   END
 END O[10]
 PIN I[25]
   DIRECTION INPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 35.119 -45.561 35.179 -45.511 ;
   END
 END I[25]
 PIN I[27]
   DIRECTION INPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 37.855 -45.561 37.915 -45.511 ;
   END
 END I[27]
 PIN O[21]
   DIRECTION OUTPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 29.994 -45.561 30.078 -45.496 ;
   END
 END O[21]
 PIN I[13]
   DIRECTION INPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 18.703 -45.561 18.763 -45.511 ;
   END
 END I[13]
 PIN I[33]
   DIRECTION INPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 46.063 -45.561 46.123 -45.511 ;
   END
 END I[33]
 PIN A[3]
   DIRECTION INPUT ;
   USE SIGNAL ;
   PORT
     LAYER M7 ;
       RECT 84.577 160.902 84.689 161.002 ;
```

```
    END
  END A[3]
  PIN O[36]
    DIRECTION OUTPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 50.514 -45.561 50.598 -45.496 ;
    END
  END O[36]
  PIN I[29]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 40.591 -45.561 40.651 -45.511 ;
    END
  END I[29]
  PIN I[4]
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER M7 ;
        RECT 6.391 -45.561 6.451 -45.511 ;
    END
  END I[4]
  OBS
    LAYER OverlapCheck ;
#C1n      POLYGON -3.534 -45.561 84.689 -45.561 84.689 214.625 -3.534 214.625 -3.534 -45.56
-3.533 -45.56 -3.533 214.624 84.688 214.624 84.688 -45.56 -3.534 -45.56 -3.534 -45.561 ;
     # POLYGON -3.534 -45.561 84.689 -45.561 84.689 214.625 -3.534 214.625 -3.534 -45.551 -
3.524 -45.551 -3.524 214.615 84.679 214.615 84.679 -45.551 -3.534 -45.551 -3.534 -45.561 ;

      ##POLYGON 84.689 -45.561 -3.534 -45.561 -3.534 15.561 84.679 15.561 84.679 80.561 -
3.534 80.561 -3.534 145.561 84.679 145.561 84.679 214.615 -3.534 214.615 -3.534 214.625
84.689 214.625 84.689 145.551 -3.544 145.551 -3.544 80.571 84.689 80.571 84.689 15.551 -
3.544 15.551 -3.544 -45.571 84.689 -45.571 84.689 -45.561 ;
     POLYGON 71.689 -30.561 12.534 -30.561 12.534 10.561 71.679 10.561 71.679 50.561
12.534 50.561 12.534 90.561 71.679 90.561 71.679 130.561 12.534 130.561 12.534 170.561
71.679 170.561 71.679 199.605 12.534 199.605 12.534 199.615 71.689 199.615 71.689 170.551
12.544 170.551 12.544 130.571 71.689 130.571 71.689 90.551 12.544 90.551 12.544 50.571
71.689 50.571 71.689 10.551 12.544 10.551 12.544 -30.551 71.689 -30.551 71.689 -30.561 ;


#C    LAYER M1 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
#C    LAYER M2 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
#C    LAYER M3 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
```

```
#C    LAYER M4 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
#C    LAYER M5 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
#C    LAYER M6 ;
#C      RECT -3.534 -45.561 84.689 214.625 ;
  END
END SRAM_128x38slp

END LIBRARY
```

ANEXA 3. Script pentru rularea procesului de lucru 3D

```
source -verbose ./scripts/setup_icc.tcl

set VER 3D_01_flat_26Ian2025

#set VLG_FILE "/home/ionica/work/synth/OpenSPARCT2.1.3-saed32/design_spc_rvt/sys/iop/spc/lsu/synopsys/gate/lsu.flat.high_efort.vSyn"
	#set VLG_FILE   "/home/caduser/proiect/work/lsu.v"

##########################################################################################
############################### Design evaluation ########################################
##########################################################################################
	if { 0 } {
	if { [file exists ${MW}/${DESIGN}_lib_eval] } { file delete -force ${MW}/${DESIGN}_lib_eval }
	create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
		-mw_reference_library "$SC_MW $MEM_MW" ${MW}/${DESIGN}_lib_eval
	open_mw_lib ${MW}/${DESIGN}_lib_eval

	read_verilog $VLG_FILE
	#reset_upf
	#load_upf $UPF_FILE
	current_design $DESIGN
	link -force

	create_scenario func
	set_tlu_plus_files -max_tluplus $MAX_TLUPLUS -min_tluplus $MIN_TLUPLUS
	set_operating_conditions -library saed32rvt_ss0p95v125c ss0p95v125c
	#read_sdc $SDC_FILE
	source  $SDC_FILE

	derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;#  -create_ports all
	set_app_var physopt_delete_unloaded_cells false
	set_dont_touch [get_flat_cells UPD_*]

	################
	set areaSC 0.000
	set areaM  0.000
	foreach_in_col c [get_flat_cells -all] {
	  if { [get_attribute $c mask_layout_type]=="std" } { set areaSC [expr $areaSC+[get_attribute $c area]]
	  } else {                                set areaM  [expr $areaM +[get_attribute $c area]] }
	}
	puts "SC  area: $areaSC" ;# SC  area: 105388.688064
	puts "MEM area: $areaM"  ;# MEM area: 280394.868326
	set memRed [expr [format %.2f [expr ($areaM-$areaSC)/$areaM]]-0.02]

	#set s [expr sqrt ($p*($p-$L)*($p-$l)*($p-$c))*2.000]
```

```
        foreach_in_col c [get_flat_cells -all] {
          if { [get_attribute $c mask_layout_type]!="std" } {
            puts "[get_object_name $c] - [get_attribute $c area] vs. [expr $memRed*[get_attribute
$c area]]"
            puts "                    [get_attribute $c width] x [get_attribute $c height]"
          }
        }
        #tlb_lsu_mem8 - 22954.389478 13772.6336868
        set l0  88.223
        set L0 260.186
        #dva_lsu_mem7 - 8131.289472 4878.7736832
        #lru_lsu_mem7 - 8131.289472 4878.7736832
        set l1  75.456
        set L1 107.762
        ##dta_lsu_mem3 - 19773.068614 11863.8411684
        #dta_lsu_mem5 - 19773.068614 11863.8411684
        #dta_lsu_mem4 - 19773.068614 11863.8411684
        #dta_lsu_mem6 - 19773.068614 11863.8411684
        #dca_array_way23_lsu_mem_left - 19773.068614 11863.8411684
        #dca_array_way23_lsu_mem_right - 19773.068614 11863.8411684
        #dca_array_way01_lsu_mem_left - 19773.068614 11863.8411684
        #dca_array_way01_lsu_mem_right - 19773.068614 11863.8411684
        set l2  77.279
        set L2 255.866
        #stb_ram_lsu_mem2 - 26727.461530 16036.476918
        set l3 144.029
        set L3 185.570
        #stb_cam_lsu_mem9 - 14917.273270 8950.363962
        set l4  90.677
        set L4 164.510
        #cpq_tlu_mem1 - 41348.616192 24809.1697152
        set l5 239.616
        set L5 172.562

        set S [expr 1*($l0-$sp)*($L0-$sp) + \
                2*($l1-$sp)*($L1-$sp) + \
                8*($l2-$sp)*($L2-$sp) + \
                1*($l3-$sp)*($L3-$sp) + \
                1*($l4-$sp)*($L4-$sp) + \
                1*($l5-$sp)*($L5-$sp)]

        } ;# end of if { 0 }
        #end eval
        #close_mw_cel
        #close_mw_lib

##########################################################################################
################################ Blocks placement ########################################
##########################################################################################
        if { [file exists ${MW}/${DESIGN}_lib_rsph] } { file delete -force
${MW}/${DESIGN}_lib_rsph }
```

```
	create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
		-mw_reference_library "$SC_MW $MEM_MW_RSPH" ${MW}/${DESIGN}_lib_rsph
	open_mw_lib ${MW}/${DESIGN}_lib_rsph

	read_verilog $VLG_FILE
	#reset_upf
	#load_upf $UPF_FILE
	current_design $DESIGN
	link -force

	create_scenario func
	set_tlu_plus_files -max_tluplus $MAX_TLUPLUS -min_tluplus $MIN_TLUPLUS
	set_operating_conditions -library saed32rvt_ss0p95v125c ss0p95v125c
	#read_sdc $SDC_FILE
	source $SDC_FILE


	#derive_pg_connection -create_nets
	derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;# -create_ports all
	set_app_var physopt_delete_unloaded_cells false
	set_dont_touch [get_flat_cells UPD_*]

	#report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.preFlp.tm.rpt
	### Memory utilization on 2nd layer will be <100% according to mems' no and mems' shape !!!
	### floorplan must be secified by width & heigh - mems are just some phantoms
	#create_floorplan -control_type aspect_ratio -core_aspect_ratio 1.0 -core_utilization 0.25 \
	#		-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
	#		-start_first_row -flip_first_row
	create_floorplan -control_type width_and_height -core_width 450 -core_height 535 \
			-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
			-start_first_row -flip_first_row
	#create_floorplan -control_type width_and_height -core_width 500 -core_height 500 \
			-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
			-start_first_row -flip_first_row
	# create_floorplan -control_type width_and_height -core_width 375 -core_height 535 \
	#		-left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 \
	#		-start_first_row -flip_first_row

	foreach layer {M7 M8 M9 MRDL} { create_route_guide -name ${layer}_blk -no_signal_layers $layer -coordinate "[get_attribute [get_die_area] bbox]" }
	#For create_fp_placement & legalize_fp_placement
	# HALO - 30um mem spacing based on area reduction + 5um = 35um
	set_fp_placement_strategy -min_distance_between_macros 35.000 \
			-macros_min_distance_applies_to [get_cells -hier -filter is_hard_macro==true]
	set_fp_placement_strategy -macro_orientation all -pin_routing_aware true
```

```
        # -virtual_IPO true
        # -congestion_effort medium
        # -macros_min_distance_applies_to [get_flat_cells -filter is_hard_macro==true]
        # -macros_on_edge on
        # -pin_routing_aware true
        # set_fp_macro_options $macros -legal_orientations "N W S E FN FW FS FE" -
side_channel {left right top bottom}
        # set_keepout_margin -type hard -outer {56.000 56.000 56.000 56.00} [get_cells -filter
is_hard_macro==true]
        # set place_max_cell_density_threshold 0.7
        # create_fp_placement  # Virtual flat placement
        create_fp_placement -optimize_pins -congestion_driven -timing_driven
        # -effort high
        # -timing_driven
        # -congestion_driven
        # -incremental all
        # -optimize_pins
        # legalize_fp_placement # legalize virtual flat placement
        set_attribute [get_flat_cells -filter is_hard_macro==true] is_fixed true
        set_attribute [get_ports ] is_fixed true -quiet
        ### remove_placement -object_type standard_cell
        create_fp_placement -congestion_driven -timing_driven -incremental all


        save_mw_cel ${DESIGN}
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.blkPlace.tm.rpt
        close_mw_cel
        close_mw_lib
##########################################################################################
################################### Design placement ###################################
##########################################################################################
        set incBlkPlc 0
        if { $incBlkPlc } {
          if { [file exists ${MW}/${DESIGN}_lib_sph] } { file delete -force
${MW}/${DESIGN}_lib_sph }
          create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW_SPH"
${MW}/${DESIGN}_lib_sph
          copy_mw_cel -from_library ${MW}/${DESIGN}_lib_rsph -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib_sph
          #set_mw_lib_reference -mw_reference_library "$SC_MW $MEM_MW_SPH"
${MW}/${DESIGN}_lib_sph
          #copy_mw_lib -from ${MW}/${DESIGN}_lib_sph -to ${MW}/${DESIGN}_lib_

          open_mw_lib ${MW}/${DESIGN}_lib_sph
          open_mw_cel ${DESIGN}
          current_design $DESIGN
          link -force
          #derive_pg_connection -create_nets
          derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;#  -
create_ports all
```

```
        set_app_var physopt_delete_unloaded_cells false
        set_dont_touch [get_flat_cells UPD_*]

        set_fp_placement_strategy -min_distance_between_macros 5.000 \
                        -macros_min_distance_applies_to [get_cells -filter
is_hard_macro==true]
        set_fp_placement_strategy -macro_orientation all
        create_fp_placement -congestion_driven -timing_driven -incremental all
        save_mw_cel
        close_mw_lib
        copy_mw_lib -from ${MW}/${DESIGN}_lib_sph -to
${MW}/${DESIGN}_lib${VER}_place
      } else {
        if { [file exists ${MW}/${DESIGN}_lib_rph] } { file delete -force
${MW}/${DESIGN}_lib_rph }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW_PH"
${MW}/${DESIGN}_lib_rph
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib_rsph -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib_rph
        #set_mw_lib_reference -mw_reference_library "$SC_MW $MEM_MW_PH"
${MW}/${DESIGN}_lib_rph
        if { [file exists ${MW}/${DESIGN}_lib${VER}_place] } { file delete -force
${MW}/${DESIGN}_lib${VER}_place }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW_PH"
${MW}/${DESIGN}_lib${VER}_place
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib_rph -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib${VER}_place
      }
      open_mw_lib ${MW}/${DESIGN}_lib${VER}_place
      open_mw_cel ${DESIGN}


      current_design $DESIGN
      link -force
      #derive_pg_connection -create_nets
      derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;#  -
create_ports all
      set_app_var physopt_delete_unloaded_cells false
      set_dont_touch [get_flat_cells UPD_*]
      #set_attribute [get_flat_cells -filter is_hard_macro==true] is_fixed true
      set_attribute [get_flat_cells -filter is_hard_macro==true] is_fixed false
      #set_undoable_attribute [get_cells dta_lsu_mem3] orientation {FN}
      #change_selection [get_cells dta_lsu_mem3]
      #move_objects -delta {61.661 0.000} [get_selection]
      #set_undoable_attribute [get_cells dta_lsu_mem5] orientation {S}
      #change_selection [get_cells dta_lsu_mem5]
      #move_objects -delta {61.661 0.000} [get_selection]
      #set_undoable_attribute [get_cells dta_lsu_mem6] orientation {N}
```

```
        #change_selection [get_cells dta_lsu_mem6]
        #move_objects -delta {-61.661 0.000} [get_selection]
        #set_undoable_attribute [get_cells dva_lsu_mem7] orientation {S}
        #change_selection [get_cells dva_lsu_mem7]
        #move_objects -delta {70.661 0.000} [get_selection]
        #set_undoable_attribute  [get_cells dca_array_way23_lsu_mem_right] orientation {FW}
        #change_selection [get_cells dca_array_way23_lsu_mem_right]
        #move_objects -delta {0.0 -70.0} [get_selection]
        #set_undoable_attribute [get_cells dca_array_way01_lsu_mem_right] orientation {E}
        #change_selection [get_cells dca_array_way01_lsu_mem_right]
        #move_objects -delta {0.0 70.0} [get_selection]

        change_selection [get_cells dca_array_way23_lsu_mem_right]
        move_objects -delta {0.0 4.484} [get_selection]
        change_selection [get_cells tlb_lsu_mem8]
        move_objects -delta {-5.700 0.0} [get_selection]
        set_attribute [get_flat_cells -filter is_hard_macro==true] is_fixed true
        # set_congestion_options
        # create_placement     # Create rough placement
        create_placement -congestion -congestion_effort medium -timing_driven ;# -
continue_on_missing_scandef
        # legalize_placement   # Legalize placement
        # place_fp_pins        # Auto Pin Assignment
        # place_opt            # Run place_opt
        place_opt -skip_initial_placement -congestion ;# -continue_on_missing_scandef ;# -
area_recovery ;# -optimize_dft
        # refine_placement     # optimize congestion
        # legalize_placement ;# -incremental ;# -timing
        save_mw_cel ${DESIGN}
        report_placement_utilization      > ${REPORTS}/${DESIGN}.${VER}.place.util.rpt
        report_qor                        > ${REPORTS}/${DESIGN}.${VER}.place.qor.rpt
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.place.tm.rpt

        close_mw_cel
        close_mw_lib
##########################################################################################
########################################## CTS ###########################################
##########################################################################################


        ##in order to create clock tree, the library shoulb be RSPH, otherwise the nets can not be
routed
        ##replace $MEM_MW_PH with $MEM_MW_RSPH
        if { [file exists ${MW}/${DESIGN}_lib${VER}_cts] } { file delete -force
${MW}/${DESIGN}_lib${VER}_cts }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW_RSPH"
${MW}/${DESIGN}_lib${VER}_cts
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib${VER}_place -from ${DESIGN}
\
                -to_library ${MW}/${DESIGN}_lib${VER}_cts
```

```
        open_mw_lib ${MW}/${DESIGN}_lib${VER}_cts
        open_mw_cel ${DESIGN}
        current_design $DESIGN
        link -force
        derive_pg_connection -ground_net VSS -power_net VDD -create_ports top ;#  -create_ports all
        #set_fix_hold
        clock_opt ;#-congestion ;#-area_recovery ;#-fix_hold_all_clocks -;#no_clock_route; #-optimize_dft ;#-continue_on_missing_scandef
        save_mw_cel
        report_placement_utilization     > ${REPORTS}/${DESIGN}.${VER}.CTS.util.rpt
        report_qor                       > ${REPORTS}/${DESIGN}.${VER}.CTS.qor.rpt
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.CTS.tm.rpt
        close_mw_cel
        close_mw_lib
################################################################################################
########################################### Route ##############################################
################################################################################################
        ##in order to route the design, the library shoulb be RSPH, otherwise the nets can not be routed
        ##replace $MEM_MW_PH with $MEM_MW_RSPH

        if { [file exists ${MW}/${DESIGN}_lib${VER}_route] } { file delete -force ${MW}/${DESIGN}_lib${VER}_route }
        create_mw_lib -case_sensitive -bus_naming_style {[%d]} -technology $TECH_FILE \
                -mw_reference_library "$SC_MW $MEM_MW_RSPH" ${MW}/${DESIGN}_lib${VER}_route
        copy_mw_cel -from_library ${MW}/${DESIGN}_lib${VER}_cts -from ${DESIGN} \
                -to_library ${MW}/${DESIGN}_lib${VER}_route
        open_mw_lib ${MW}/${DESIGN}_lib${VER}_route
        open_mw_cel ${DESIGN}
        # set_route_zrt_common_options
        # set_route_zrt_detail_options
        route_opt -initial_route_only
        save_mw_cel
        route_opt -skip_initial_route ;# -effort low
        save_mw_cel
        verify_zrt_route
        verify_lvs -check_open_locator -check_short_locator
        report_placement_utilization     > ${REPORTS}/${DESIGN}.${VER}.Route.util.rpt
        report_qor                       > ${REPORTS}/${DESIGN}.${VER}.Route.qor.rpt
        report_constraints -all_violators > ${REPORTS}/${DESIGN}.${VER}.Route.tm.rpt
        close_mw_cel
        close_mw_lib
        ################################################################################################
        if { 0 } {
        set areaSC 105389.000
        set areaM  280395.000
        #tlb_lsu_mem8 - 22954.389478 13772.6336868
        set l0  88.223
```

```
set L0 260.186
#dva_lsu_mem7 - 8131.289472 4878.7736832
#lru_lsu_mem7 - 8131.289472 4878.7736832
set l1  75.456
set L1 107.762
##dta_lsu_mem3 - 19773.068614 11863.8411684
#dta_lsu_mem5 - 19773.068614 11863.8411684
#dta_lsu_mem4 - 19773.068614 11863.8411684
#dta_lsu_mem6 - 19773.068614 11863.8411684
#dca_array_way23_lsu_mem_left - 19773.068614 11863.8411684
#dca_array_way23_lsu_mem_right - 19773.068614 11863.8411684
#dca_array_way01_lsu_mem_left - 19773.068614 11863.8411684
#dca_array_way01_lsu_mem_right - 19773.068614 11863.8411684
set l2  77.279
set L2 255.866
#stb_ram_lsu_mem2 - 26727.461530 16036.476918
set l3 144.029
set L3 185.570
#stb_cam_lsu_mem9 - 14917.273270 8950.363962
set l4  90.677
set L4 164.510
#cpq_tlu_mem1 - 41348.616192 24809.1697152
set l5 239.616
set L5 172.562

set S [expr 1*($l0-$sp)*($L0-$sp) + \
        2*($l1-$sp)*($L1-$sp) + \
        8*($l2-$sp)*($L2-$sp) + \
        1*($l3-$sp)*($L3-$sp) + \
        1*($l4-$sp)*($L4-$sp) + \
        1*($l5-$sp)*($L5-$sp)]
} ;# end of if { 0 }

#exit
```

ANEXA 4. Descriere cod VHDL perceptron

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Perceptron is
   Generic (
      N : integer := 4 -- Number of inputs
   );
   Port (
      clk     : in  STD_LOGIC;
      reset   : in  STD_LOGIC;
      inputs  : in  STD_LOGIC_VECTOR(N-1 downto 0);
      weights : in  STD_LOGIC_VECTOR(N-1 downto 0);
      bias    : in  STD_LOGIC_VECTOR(7 downto 0);
      output  : out STD_LOGIC
   );
end Perceptron;

architecture Behavioral of Perceptron is

   signal sum : STD_LOGIC_VECTOR(7 downto 0);

begin

   process(clk, reset)
      variable tmp_sum : INTEGER := 0;
   begin
      if reset = '1' then
         sum <= (others => '0');
      elsif rising_edge(clk) then
         tmp_sum := 0;
         for i in 0 to N-1 loop
            if inputs(i) = '1' then
               tmp_sum := tmp_sum + CONV_INTEGER(weights(i));
            end if;
         end loop;
         tmp_sum := tmp_sum + CONV_INTEGER(bias);
         sum <= CONV_STD_LOGIC_VECTOR(tmp_sum, 8);
      end if;
   end process;

   -- Activation function: Step function
   output <= '1' when sum(7) = '0' else '0';

end Behavioral;
```

ANEXA 5. Descriere cod VHDL GN_4x4x1

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity GN_4x4x1_struct is
    Port (
        clk      : in  std_logic;
        x00, x01, x02, x03  : in  std_logic_VECTOR(7 downto 0);
        bias     : in  std_logic_VECTOR(7 downto 0);
        Yo   : out std_logic
    );
end GN_4x4x1_struct;

architecture Structural of GN_4x4x1_struct is

 type weight_val is array (0 to 15) of std_logic_vector (7 downto 0);

 signal w00 : std_logic_vector(7 downto 0) := "01010101";
 constant w01 : std_logic_vector(7 downto 0) := "01010101";
 constant w02 : std_logic_vector(7 downto 0) := "01010101";
 constant w03 : std_logic_vector(7 downto 0) := "01010101";

 constant w10 : std_logic_vector(7 downto 0) := "01010101";
 constant w11 : std_logic_vector(7 downto 0) := "01010101";
 constant w12 : std_logic_vector(7 downto 0) := "01010101";
 constant w13 : std_logic_vector(7 downto 0) := "01010101";

 constant w20 : std_logic_vector(7 downto 0) := "01010101";
 constant w21 : std_logic_vector(7 downto 0) := "01010101";
 constant w22 : std_logic_vector(7 downto 0) := "01010101";
 constant w23 : std_logic_vector(7 downto 0) := "01010101";

 constant w30 : std_logic_vector(7 downto 0) := "01010101";
 constant w31 : std_logic_vector(7 downto 0) := "01010101";
 constant w32 : std_logic_vector(7 downto 0) := "01010101";
 constant w33 : std_logic_vector(7 downto 0) := "01010101";

 constant w04 : std_logic_vector(7 downto 0) := "01010101";
 constant w14 : std_logic_vector(7 downto 0) := "01010101";
 constant w24 : std_logic_vector(7 downto 0) := "01010101";
 constant w34 : std_logic_vector(7 downto 0) := "01010101";

--signal weight_x :  weight_val;
--signal w00, w01, w02, w03 : std_logic_vector(7 downto 0);
--signal w10, w11, w12, w13 : std_logic_vector(7 downto 0);
```

```
        --signal w20, w21, w22, w23 : std_logic_vector(7 downto 0);
        --signal w30, w31, w32, w33 : std_logic_vector(7 downto 0);

        --signal w04, w14, w24, w34 : std_logic_vector(7 downto 0);

        signal x00_s, x01_s, x02_s, x03_s : std_logic_vector(18 downto 0);
        signal x00_ss, x01_ss, x02_ss, x03_ss : std_logic_vector(7 downto 0);
        signal x10_s, x11_s, x12_s, x13_s : std_logic_vector(18 downto 0);
        signal x10_ss, x11_ss, x12_ss, x13_ss : std_logic_vector(7 downto 0);
        signal x20_s : std_logic_vector(18 downto 0);
        signal x20_ss : std_logic_vector(7 downto 0);
        signal Y00_s, Y01_s, Y02_s, Y03_s, Yo_s : std_logic;
        signal Y10_s, Y11_s, Y12_s, Y13_s : std_logic;


        signal c: std_logic_vector (15 downto 0);
        signal a, b : std_logic_vector(7 downto 0);

        component neuron_NG port
        (xi, bias : in std_logic_vector (7 downto 0);
        clk : in std_logic;
        Yo : out std_logic);
        end component;

        begin
        x00_ss <= x00_s(18 downto 11);
        --x01_ss <= x01_s(18 downto 11);
        --x02_ss <= x02_s(18 downto 11);
        --x03_ss <= x03_s(18 downto 11);

        x10_ss <= x10_s(7 downto 0);
        x11_ss <= x11_s(7 downto 0);
        x12_ss <= x12_s(7 downto 0);
        x13_ss <= x13_s(7 downto 0);

        x20_ss <= x20_s(7 downto 0);

        --c <= a+b;
        --x10_s <= w00*"0000000"&Y0_s
        x10_s <= w00*"0000000"&Y00_s + w01*"0000000"&Y01_s + w02*"0000000"&Y02_s
+ w03*"0000000"&Y03_s;
        x11_s <= w10*"0000000"&Y00_s + w11*"0000000"&Y01_s + w12*"0000000"&Y02_s
+ w13*"0000000"&Y03_s;
        x12_s <= w20*"0000000"&Y00_s + w21*"0000000"&Y01_s + w22*"0000000"&Y02_s
+ w23*"0000000"&Y03_s;
        x13_s <= w30*"0000000"&Y00_s + w31*"0000000"&Y01_s + w32*"0000000"&Y02_s
+ w33*"0000000"&Y03_s;

        x20_s <= w04*"0000000"&Y10_s + w14*"0000000"&Y11_s + w24*"0000000"&Y12_s
+ w34*"0000000"&Y13_s;
```

```
n00 : neuron_NG port map (xi => x00, bias => bias, clk => clk, Yo => Y00_s);
n01 : neuron_NG port map (xi => x01, bias => bias, clk => clk, Yo => Y01_s);
n02 : neuron_NG port map (xi => x02, bias => bias, clk => clk, Yo => Y02_s);
n03 : neuron_NG port map (xi => x03, bias => bias, clk => clk, Yo => Y03_s);

n10 : neuron_NG port map (xi => x10_ss, bias => bias, clk => clk, Yo => Y10_s);
n11 : neuron_NG port map (xi => x11_ss, bias => bias, clk => clk, Yo => Y11_s);
n12 : neuron_NG port map (xi => x12_ss, bias => bias, clk => clk, Yo => Y12_s);
n13 : neuron_NG port map (xi => x13_ss, bias => bias, clk => clk, Yo => Y13_s);

n20 : neuron_NG port map (xi => x20_ss, bias => bias, clk => clk, Yo => Yo);

--w00 <= weight_x (0); w01 <= weight_x (1); w10 <= weight_x (2); w11 <= weight_x (3);

end Structural;
```

ANEXA 6. Reprezentări detaliate ale implementării circuitelor

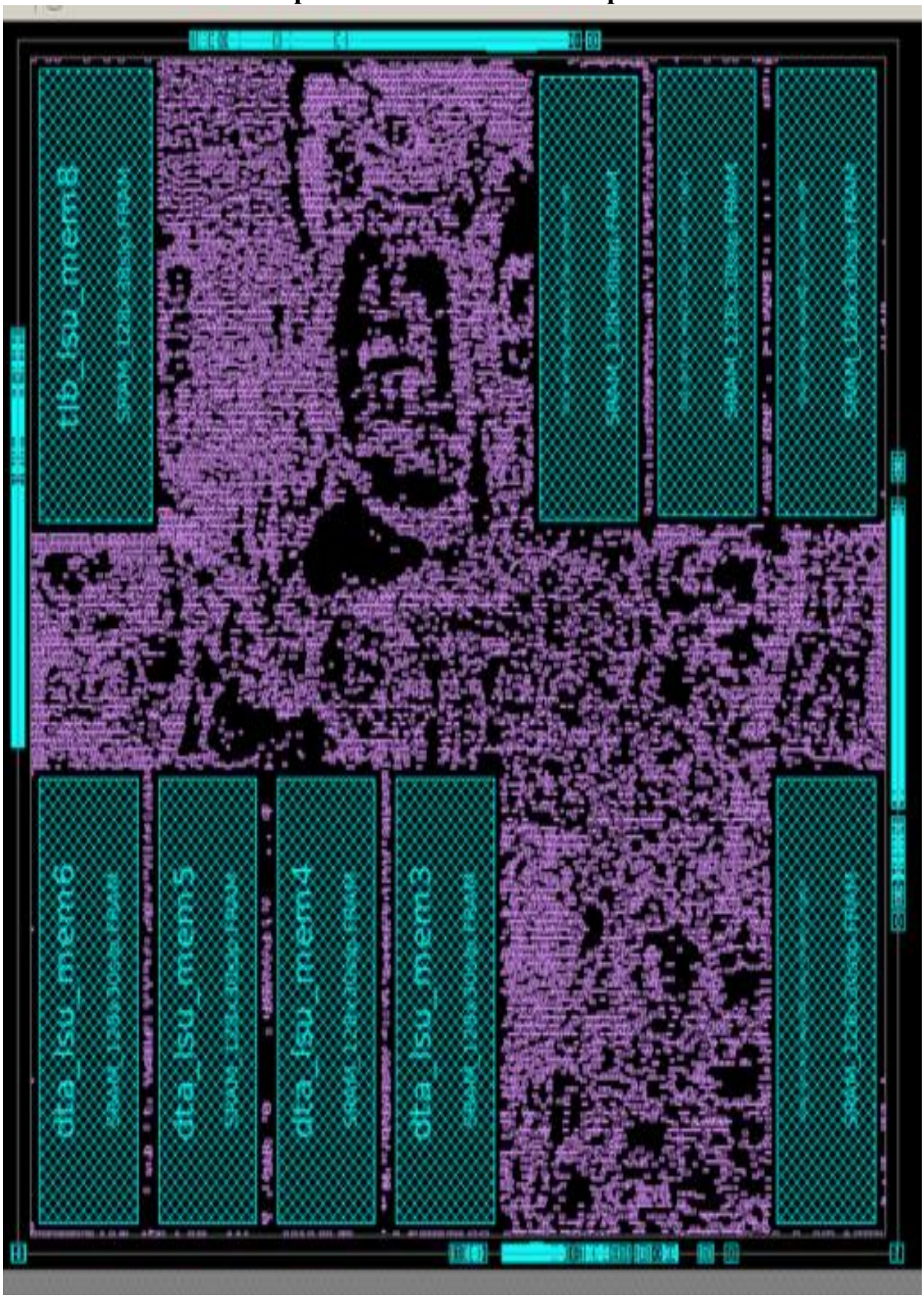


Fig. A6.1 Plasare memorii și porți logice 2D (detaliu Fig. 4.5.)

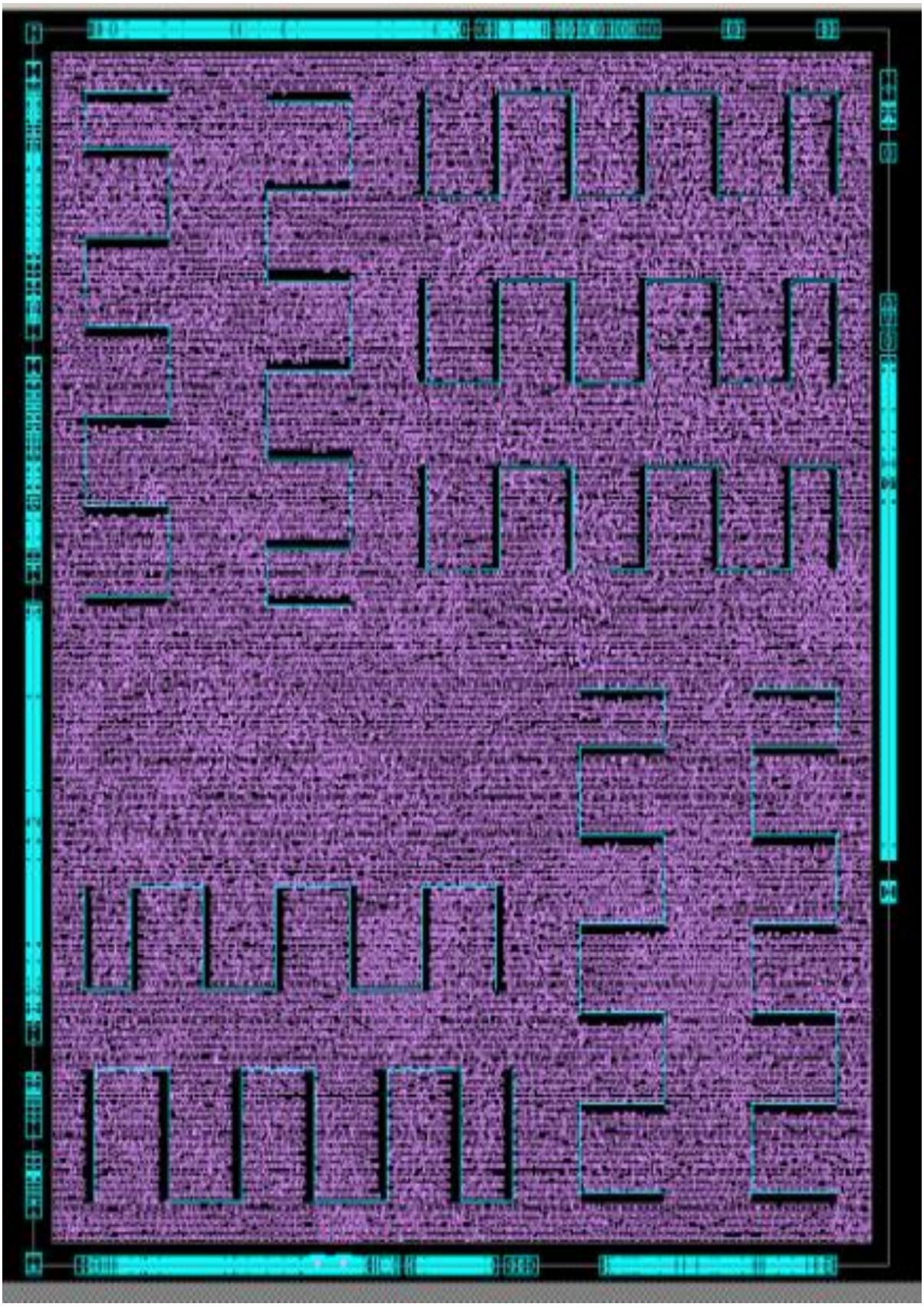


Fig. A6.2 Memorii cu formă de serpentină plasate deasupra celulelor logice [116] (detaliu Fig. 4.8.)

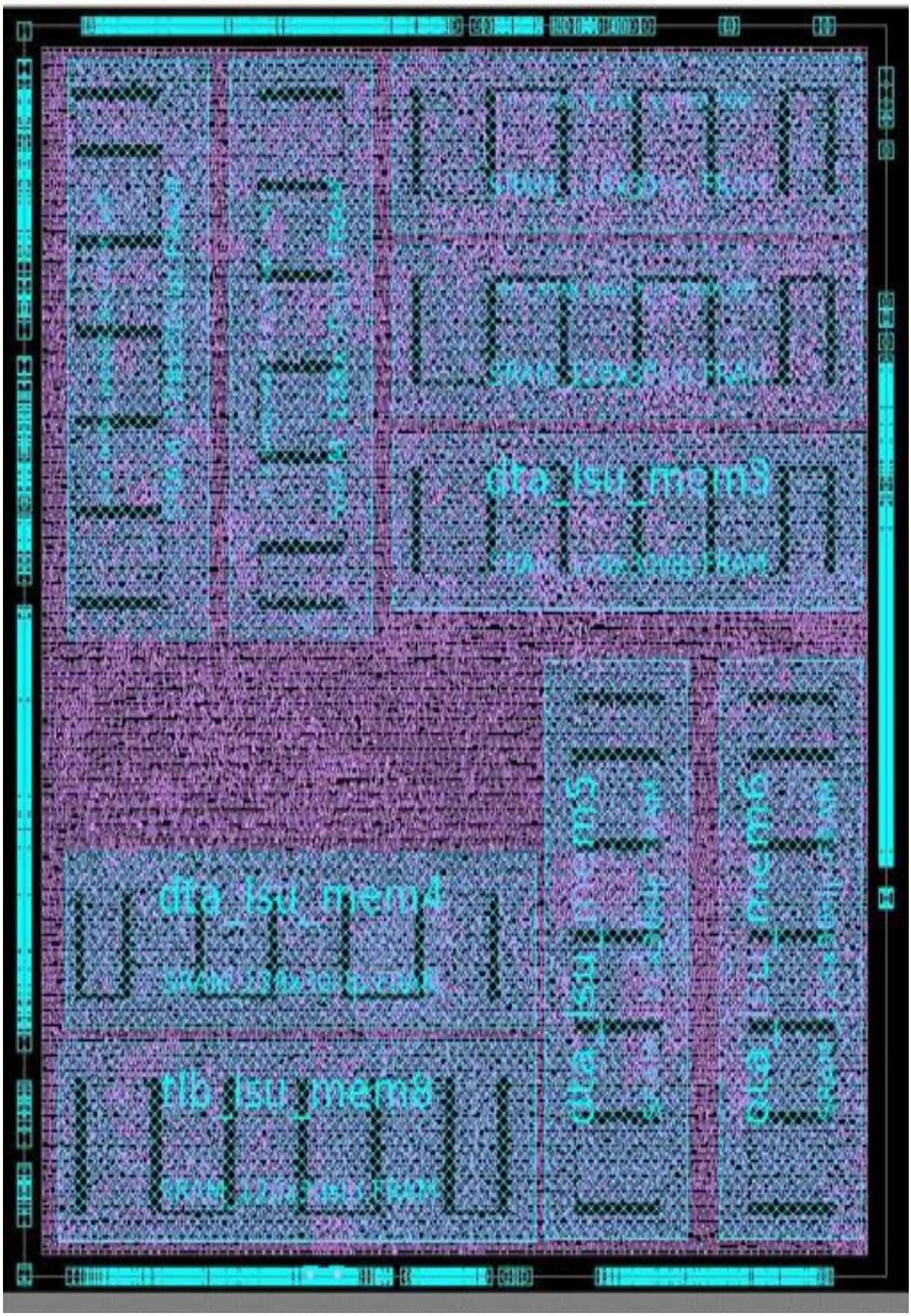


Fig. A6.3 Înlocuirea serpentinelor cu memoriile fantomă din fișierele tehnologice (detaliu Fig. 4.9.)

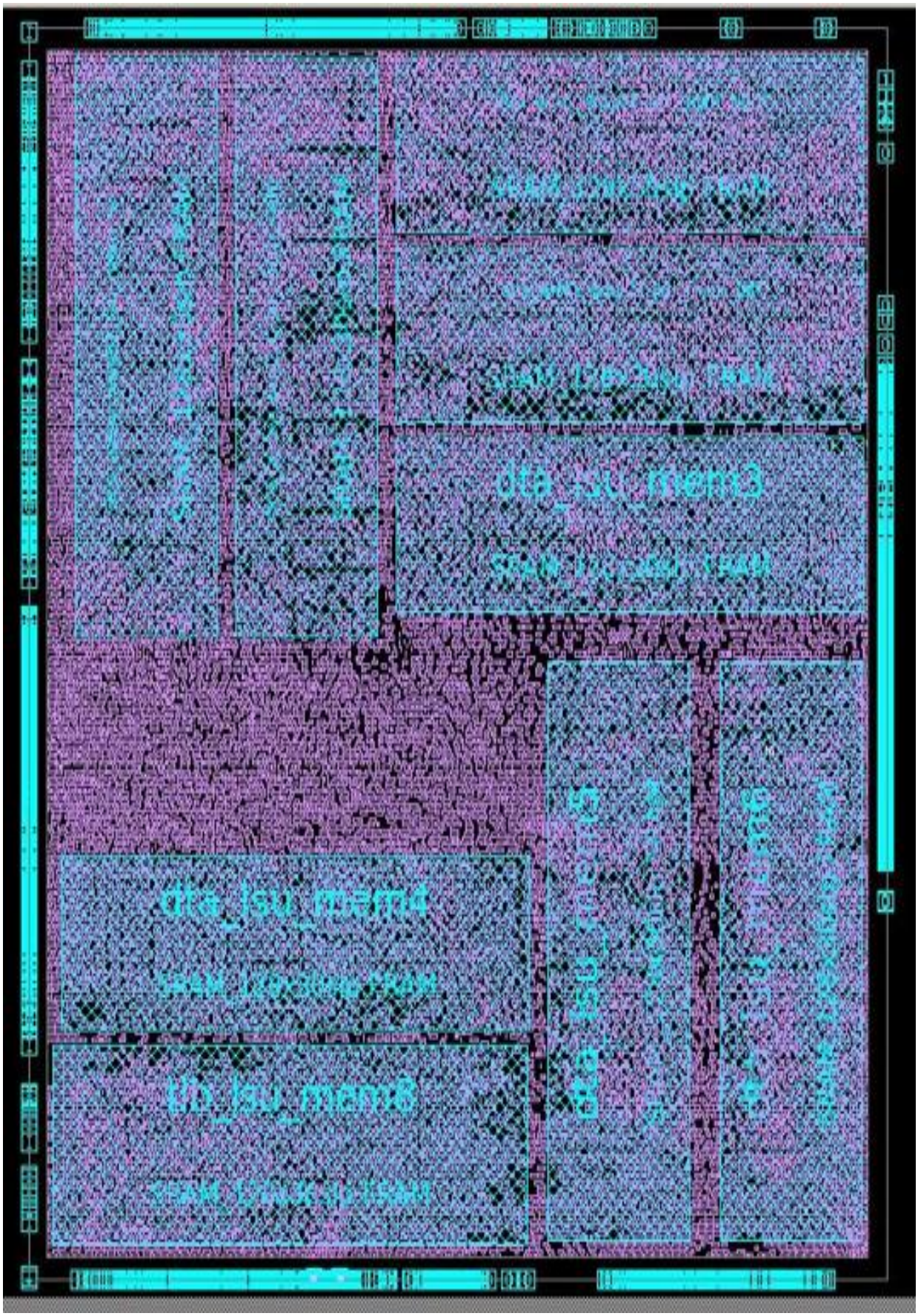


Fig. A6.4 Plasarea finală a memoriilor deasupra celulelor standard (detaliu Fig. 4.10.)

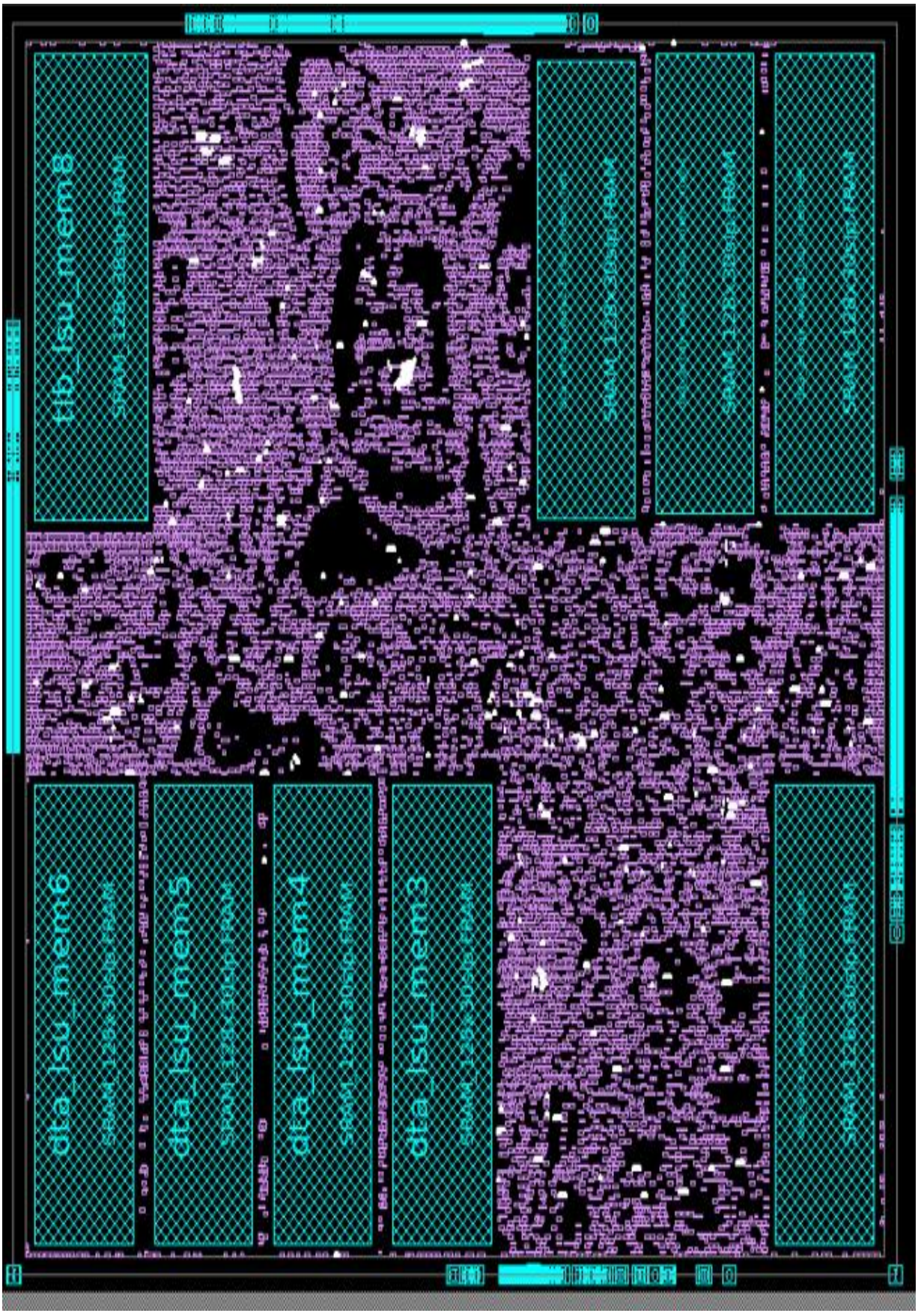


Fig. A6.5. Arborele de ceas implementat în scenariul 2D [114] (detaliu Fig. 4.12)

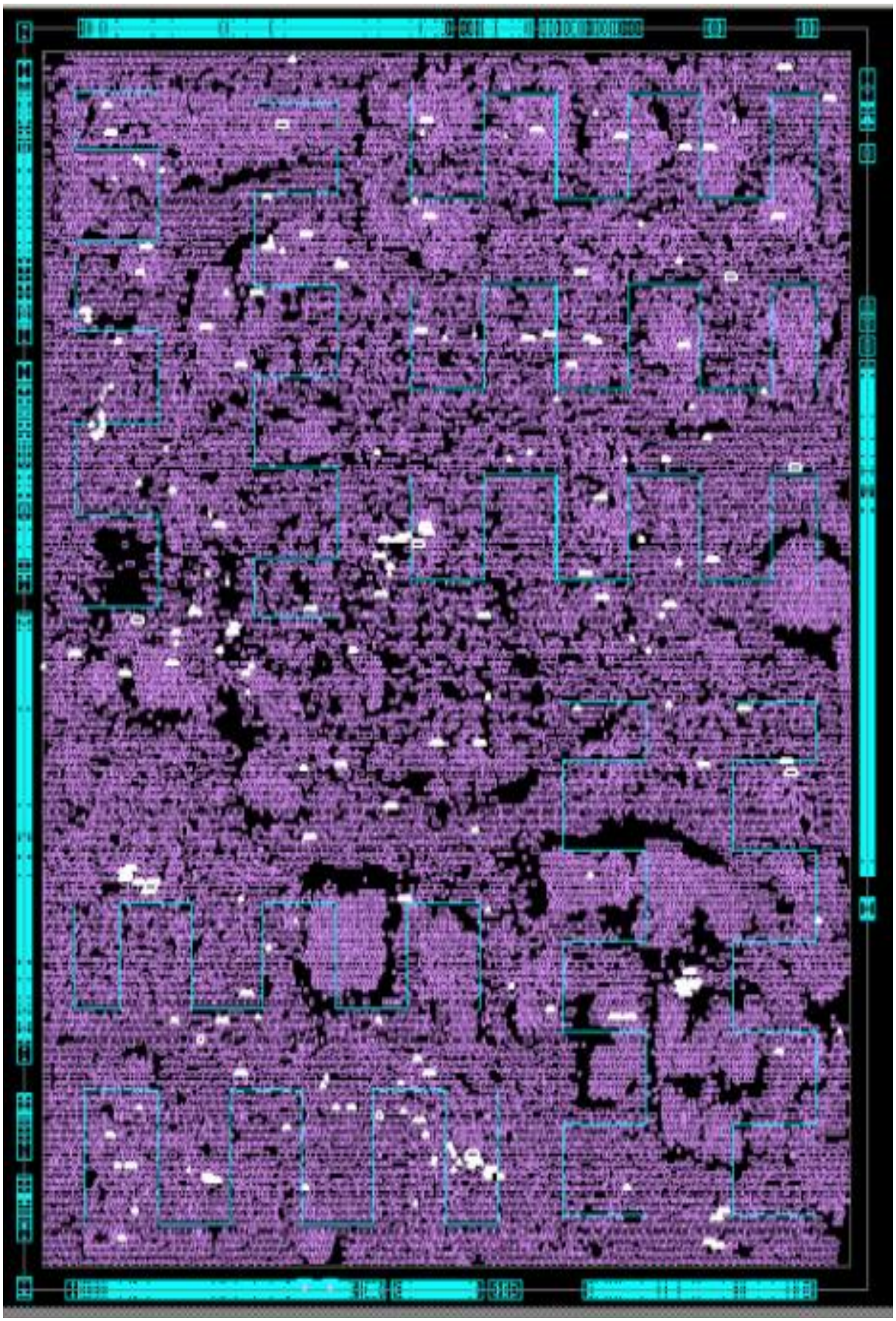


Fig. A6.6 Arborele de ceas implementat în scenariul 3D [114] (detaliu Fig. 4.13.)

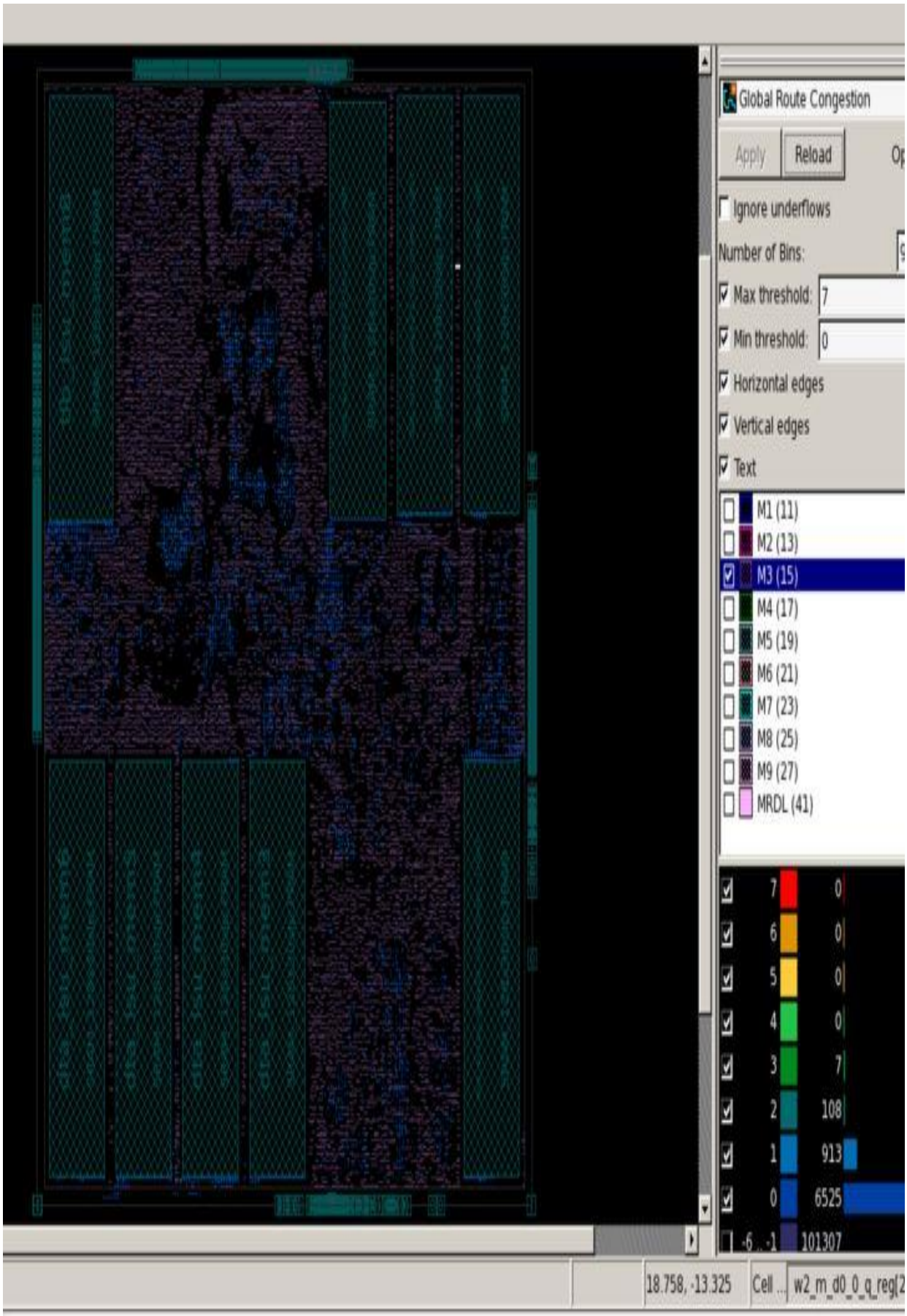


Fig. A6.7 Congestie pe M3 după rutare 2D (detaliu Fig. 4.14. (a))

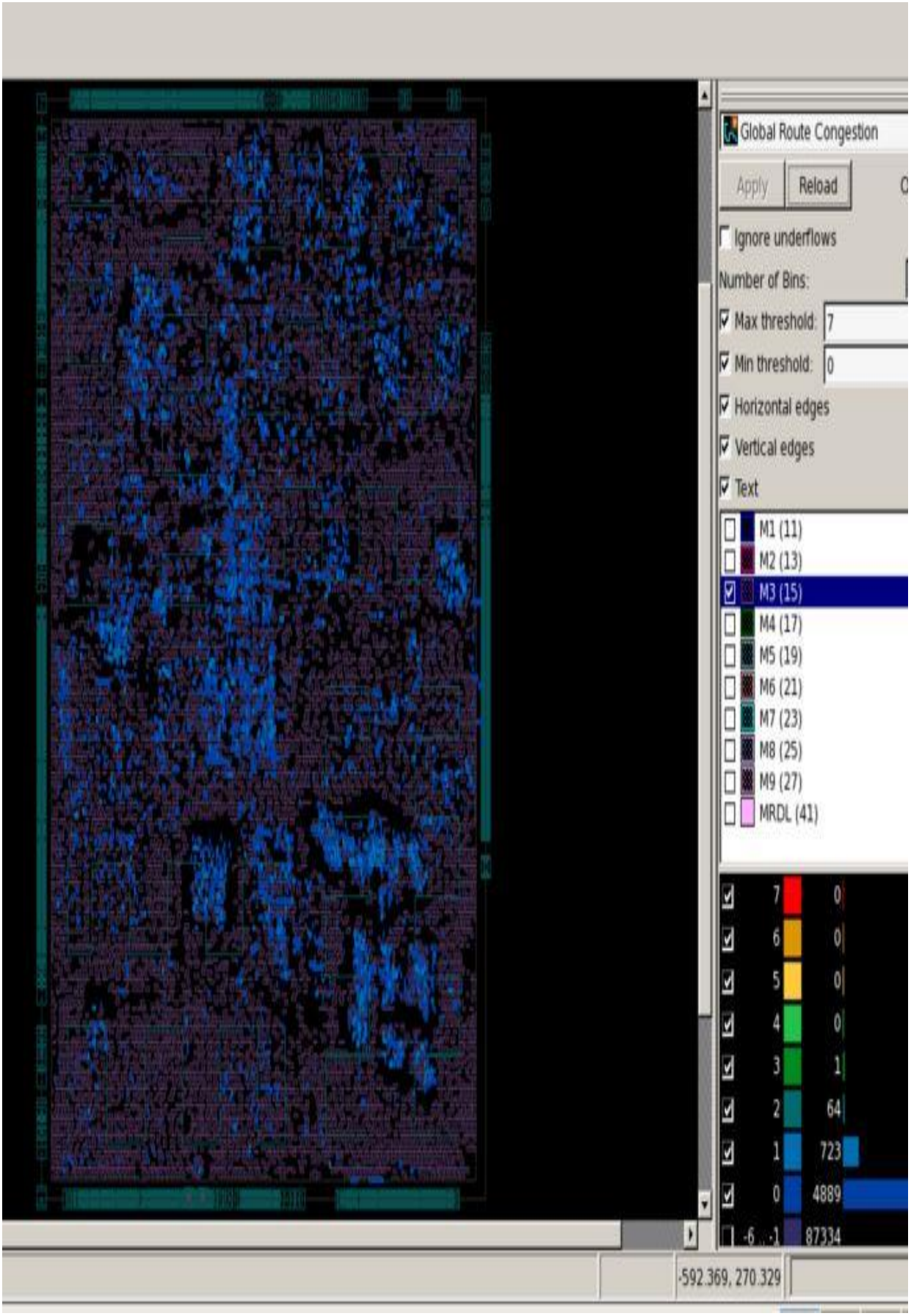


Fig. A6.8 Congestie pe M3 după rutare 3D (detaliu Fig. 4.14. (b))

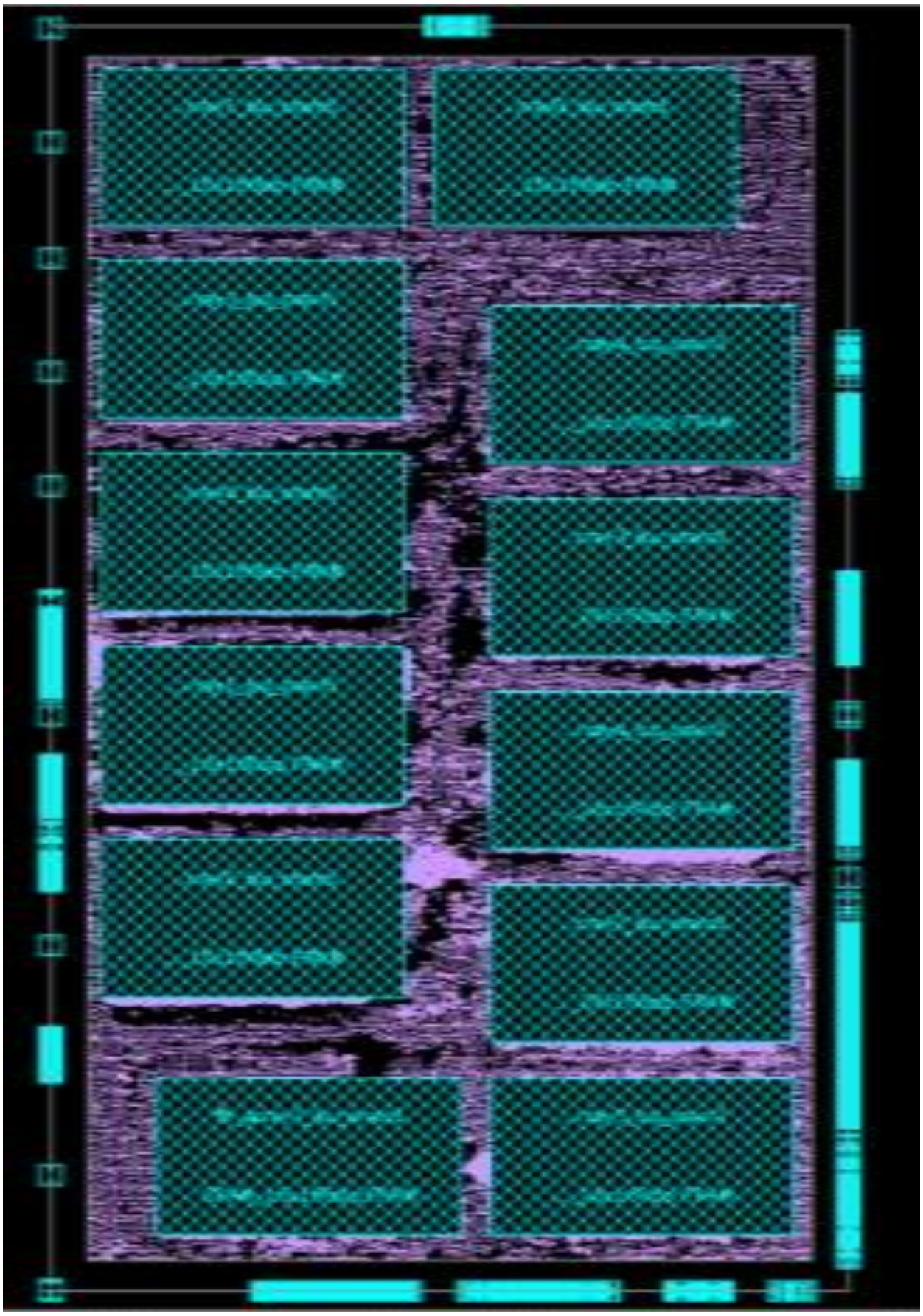


Fig. A6.9 Plasare memorii și celule logice 2D (detaliu Fig. 4.15. (a))

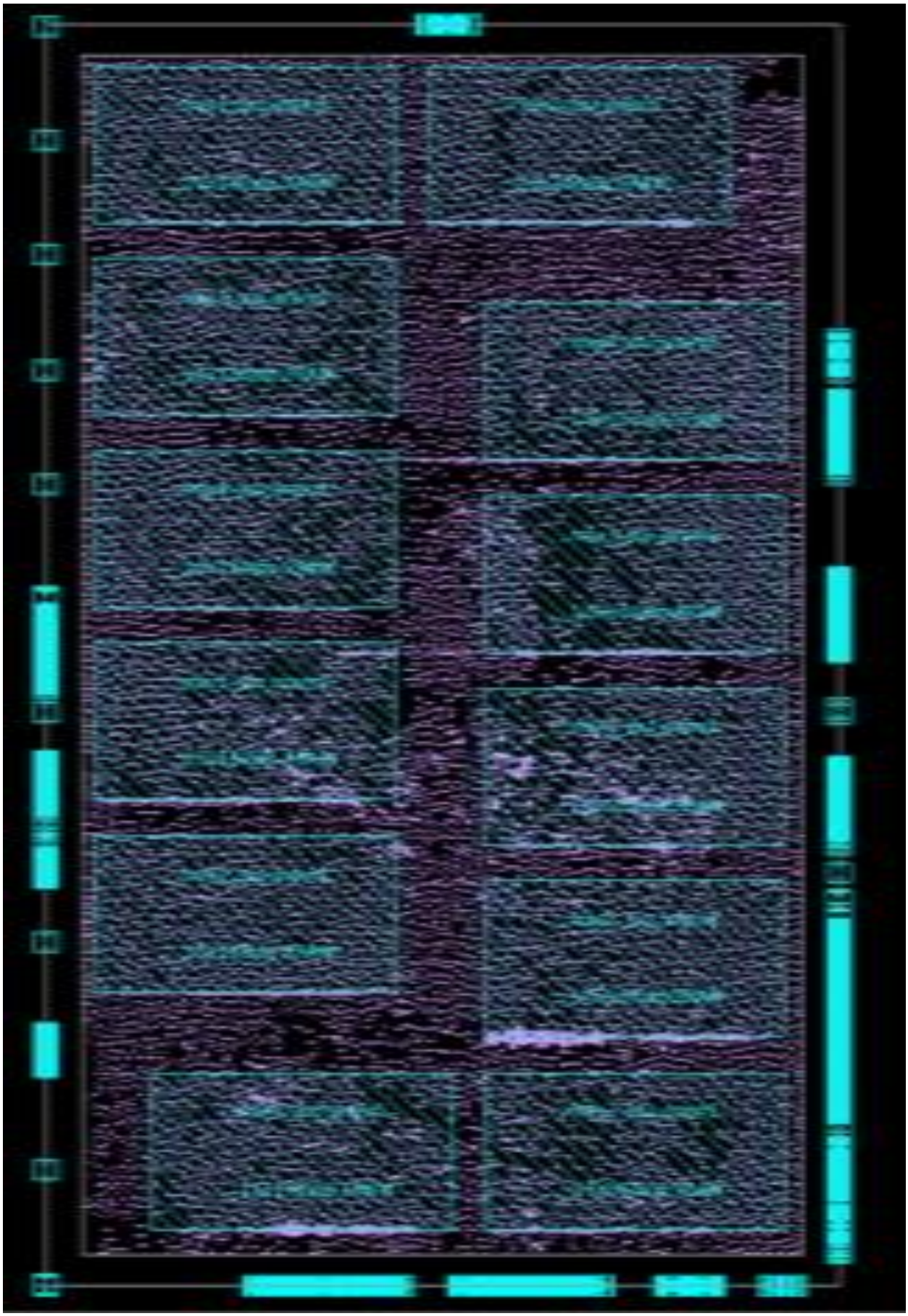


Fig. A6.10 Plasare memorii și celule logice 3D (detaliu Fig. 4.15. (b))

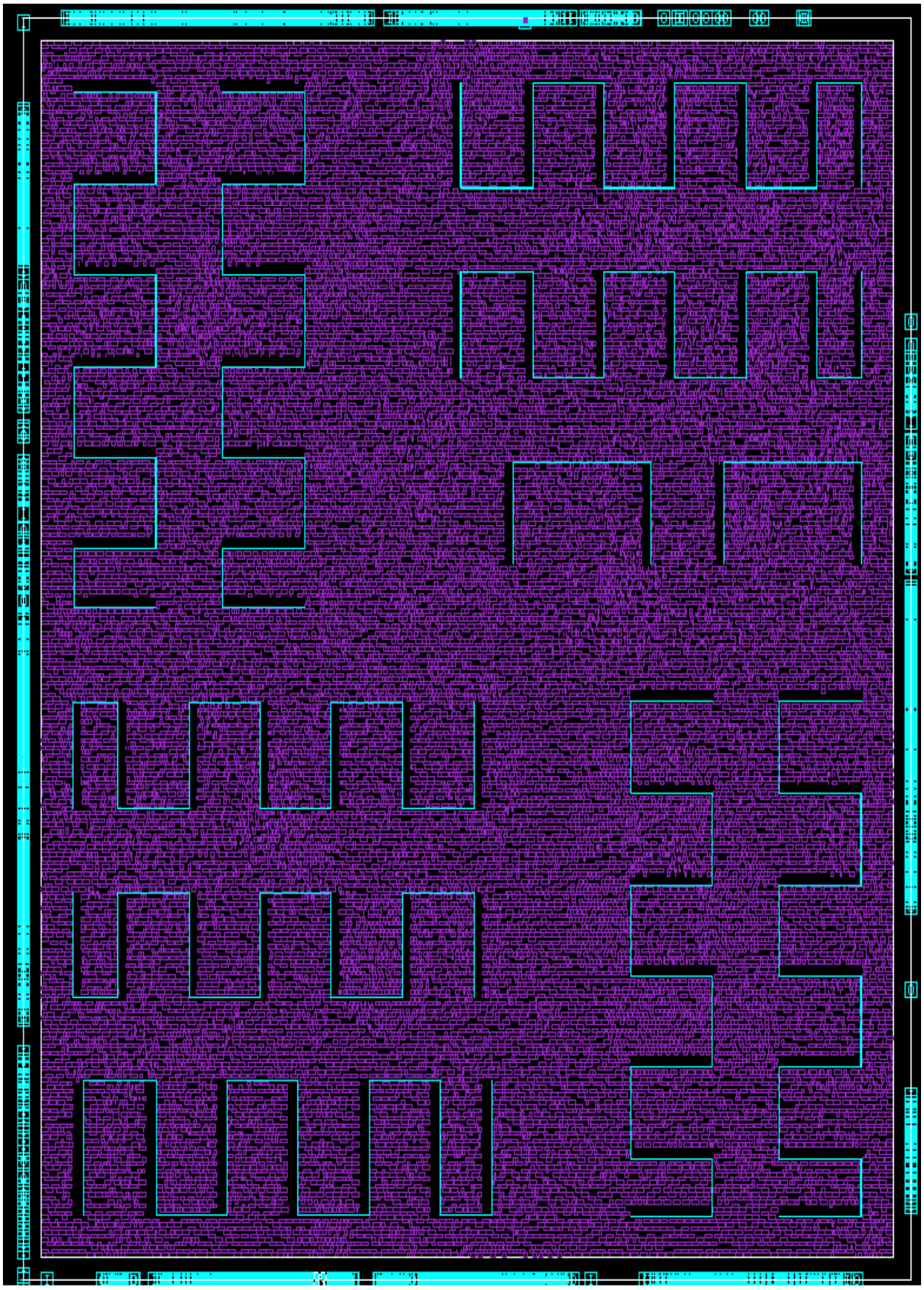


Fig. A6.11 Plasarea blocurilor cu obstrucție serpentină a memorie [115]

(detaliu Fig. 4.16) (a)

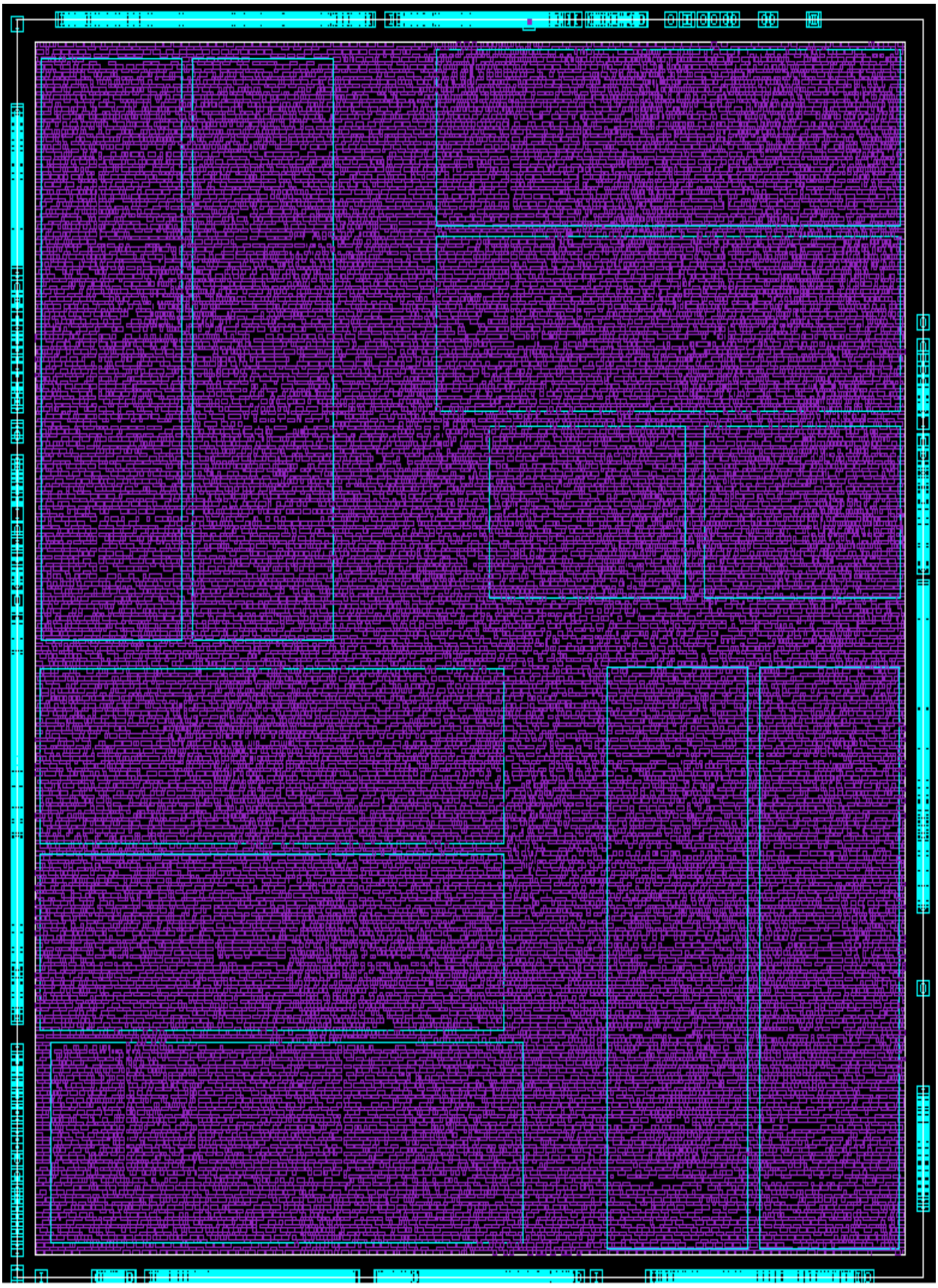


Fig. A6.12 Plasarea detaliată finală cu obstrucție periferică a memoriei [115] (detaliu Fig. 4.16. (b))

```
module neuron_NG(xi, bias, clk, Yo);
  input [7:0] xi, bias;
  input clk;
  output Yo;
  wire [7:0] xi, bias;
  wire clk;
  wire Yo;
  wire UNCONNECTED, n_0, n_1, n_2, n_3, n_4, n_5, n_6;
  wire n_7, n_8, n_9, n_10, n_11, n_12, n_13, n_14;
  wire n_15, n_16, n_17, n_18, n_19, n_20, n_21, n_22;
  wire n_23;
  DFFX1_RVT Yo_reg(.CLK (clk), .D (n_23), .Q (Yo), .QN (UNCONNECTED));
  NAND4X0_RVT g403__2398(.A1 (n_22), .A2 (n_20), .A3 (n_18), .A4
       (n_11), .Y (n_23));
  AO221X1_RVT g404__5107(.A1 (n_0), .A2 (xi[3]), .A3 (n_16), .A4
       (n_17), .A5 (n_21), .Y (n_22));
  OA222X1_RVT g405__6260(.A1 (xi[3]), .A2 (n_0), .A3 (n_8), .A4 (n_19),
       .A5 (xi[2]), .A6 (n_5), .Y (n_21));
  OA21X1_RVT g406__4319(.A1 (xi[4]), .A2 (n_16), .A3 (n_15), .Y (n_20));
  AOI22X1_RVT g407__8428(.A1 (n_14), .A2 (bias[1]), .A3 (n_9), .A4
       (n_4), .Y (n_19));
  NAND3X0_RVT g408__5526(.A1 (n_10), .A2 (n_1), .A3 (bias[6]), .Y
       (n_18));
  NAND3X0_RVT g409__6783(.A1 (n_13), .A2 (n_12), .A3 (n_3), .Y (n_17));
  NAND3X0_RVT g410__3680(.A1 (n_13), .A2 (n_2), .A3 (bias[5]), .Y
       (n_15));
  NAND3X0_RVT g411__1617(.A1 (n_13), .A2 (n_12), .A3 (bias[4]), .Y
       (n_16));
  OR2X1_RVT g412__2802(.A1 (n_4), .A2 (n_9), .Y (n_14));
  OA21X1_RVT g413__1705(.A1 (bias[6]), .A2 (n_1), .A3 (n_10), .Y
       (n_13));
  NAND2X0_RVT g414__5122(.A1 (n_6), .A2 (bias[7]), .Y (n_11));
  OR2X1_RVT g415__8246(.A1 (bias[5]), .A2 (n_2), .Y (n_12));
  AND2X1_RVT g416__7098(.A1 (n_5), .A2 (xi[2]), .Y (n_8));
  OR2X1_RVT g417__6131(.A1 (bias[7]), .A2 (n_6), .Y (n_10));
  NAND2X0_RVT g418__1881(.A1 (n_7), .A2 (xi[0]), .Y (n_9));
  INVX1_RVT g419(.A (bias[0]), .Y (n_7));
  INVX1_RVT g420(.A (xi[7]), .Y (n_6));
  INVX1_RVT g421(.A (bias[2]), .Y (n_5));
  INVX1_RVT g422(.A (xi[1]), .Y (n_4));
  INVX1_RVT g423(.A (xi[4]), .Y (n_3));
  INVX1_RVT g424(.A (xi[5]), .Y (n_2));
  INVX1_RVT g425(.A (xi[6]), .Y (n_1));
  INVX1_RVT g426(.A (bias[3]), .Y (n_0));
endmodule
```

Fig. A6.13 Proiectarea și implementarea circuitului Grain Neuron: (a) Netlistul generat de Genus pentru un neuron (detaliu Fig. 5.13. (a))

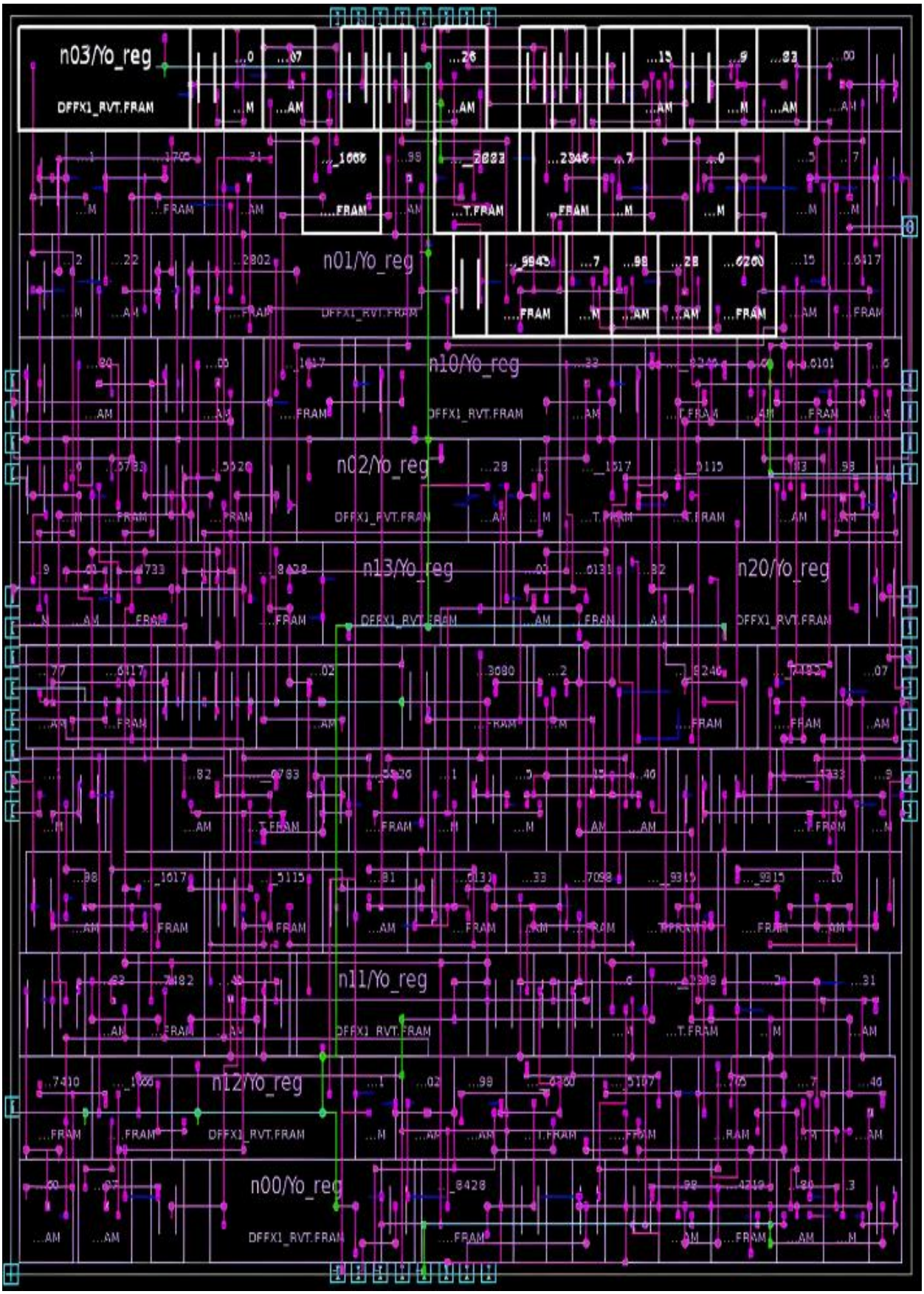


Fig. A6.14 Proiectarea și implementarea circuitului Grain Neuron: (b) Plasarea și rutarea circuitului Grain Neuron[140] (detaliu Fig. 5.13.(b))

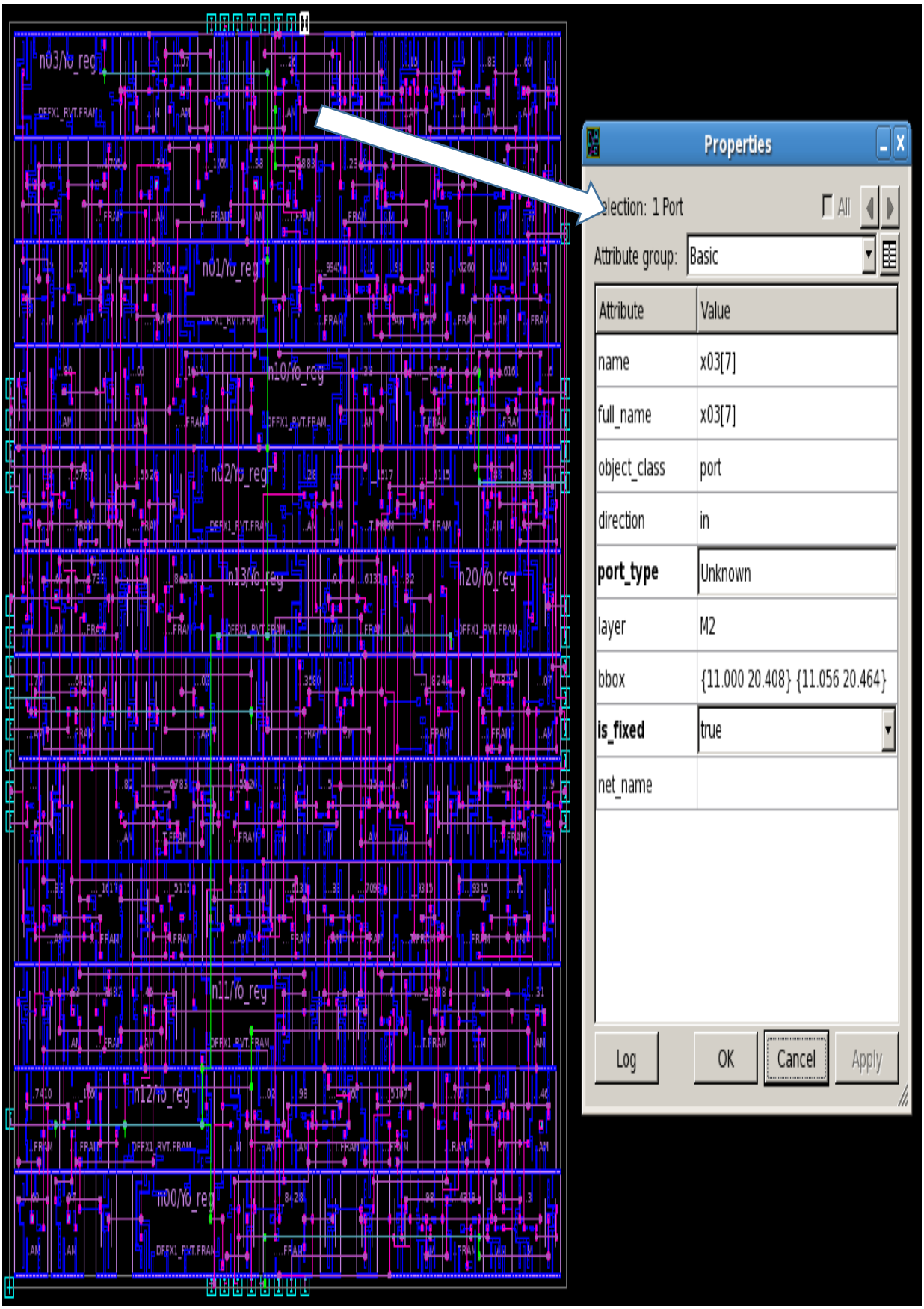


Fig. A6.15 Proprietățile portului de intrare x03[7] care aparține neuronului n03 [140] (detaliu Fig. 5.14)

ANEXA 7. Implementarea rezultatelor științifice

SILICON SERVICE SRL
CUI: RO 12552530
Nr. Reg.: J22/957/1999
Calea Chisinaului nr. 29,
Corp C40,, et.2
700290 Iasi, Romania
Tel: +40 745 064 202
office@silicon-service.com
www.silicon-service.com
Nr. de inregistrare 601 / 27.10.2025

CERTIFICAT DE IMPLEMENTARE

Prin prezenta,

Compania **Silicon Service S.R.L.**, cu sediul în Iași, România, certifică faptul că **doamna Pletea Ionica-Marcela**, angajată a companiei din anul 2009 și până în prezent, a desfășurat activități de **cercetare și dezvoltare** în cadrul companiei, în perioada **2015 – 2018**, în domeniul **circuitelor integrate tridimensionale (3D)**.

În această perioadă, doamna Pletea Ionica-Marcela a **dezvoltat o soluție de implementare a unui circuit 3D**, bazată pe **tehnologia monolitic 3D** și realizată cu **instrumentele clasice de proiectare 2D**. Soluția este **susținută științific prin articolul IEEE Xplore** (https://ieeexplore.ieee.org/document/7333518), realizat în colaborare cu deținătorul tehnologiei.

Activitatea desfășurată a contribuit la **validarea practică a conceptelor 3D** și la **consolidarea competențelor companiei** în domeniul tehnologiilor avansate de integrare.

Prezentul certificat a fost emis pentru a atesta implicarea și realizările doamnei Pletea Ionica-Marcela în cadrul activităților de cercetare și dezvoltare desfășurate la Silicon Service S.R.L.

SILICON SERVICE SRL
Prin Administrator
Aviv Consulting & Management SRL
AVIV ILAN

Întocmit,
Director Resurse Umane
Roxana Andreea Biicu

SILICON SERVICE SRL
CUI: RO 12552530
Nr. Reg.: J22/957/1999
Calea Chisinaului nr. 29,
Corp C40,, et.2
700290 Iasi, Romania
Tel: +40 745 064 202
office@silicon-service.com
www.silicon-service.com
Nr. de inregistrare 601 / 27.10.2025

CERTIFICATE OF IMPLEMENTATION

Hereby,

The company **Silicon Service S.R.L.**, headquartered in Iași, Romania, certifies that **Ms. Ionica-Marcela Pletea**, an employee of the company since 2009 to the present, has carried out research and development activities within the company during the period 2015 – 2018, in the field of **three-dimensional (3D) integrated circuits**.

During this period, Ms. Ionica-Marcela Pletea **developed a 3D circuit implementation solution based on monolithic 3D technology** and executed using **conventional 2D design tools**. This solution is scientifically supported by the **IEEE Xplore** article (https://ieeexplore.ieee.org/document/7333518), developed in collaboration with the holder of the monolithic 3D technology.

Her work contributed to the practical **validation of 3D design concepts and to the strengthening of the company's expertise** in advanced integration technologies.

This certificate is issued to attest the involvement and achievements of Ms. Ionica-Marcela Pletea in the research and development activities carried out at Silicon Service S.R.L.

SILICON SERVICE SRL
By Administrator
Aviv Consulting & Management SRL
AVIV ILAN

Prepared by
HR Manager
Roxana Andreea Biicu

SILICON SERVICE SRL
CUI: RO 12552530
Nr. Reg.: J22/957/1999
Calea Chisinaului nr. 29,
Corp C40,, et.2
700290 Iasi, Romania
Tel: +40 745 064 202
office@silicon-service.com
www.silicon-service.com
Nr. de inregistrare 319 / 18.05.2026

CERTIFICAT DE IMPLEMENTARE

Prin prezenta,

Compania **Silicon Service S.R.L.**, cu sediul în Iaşi, România, certifică faptul că doamna **Pletea Ionica-Marcela,** angajată a companiei din anul 2009 şi până în prezent, a desfăşurat activităţi de cercetare şi dezvoltare în perioada **2023–2025,** în domeniul circuitelor integrate tridimensionale (3D), al optimizării fluxurilor BACKEND şi al automatizării proceselor de implementare fizică utilizând instrumente EDA.

În această perioadă, doamna Pletea Ionica-Marcela a contribuit la dezvoltarea şi optimizarea unei metodologii de implementare pentru circuite integrate 3D, bazată pe adaptarea fluxurilor 2D existente şi automatizarea etapelor de proiectare fizică, incluzând floorplan, power planning, placement, clock tree synthesis, routing şi verificări fizice şi logice.
Activitatea de cercetare a inclus dezvoltarea de scripturi şi metode de automatizare pentru optimizarea fluxului BACKEND destinat implementării 3D, cu accent pe reducerea timpului de proiectare, identificarea timpurie a erorilor şi îmbunătăţirea performanţelor de integrare.

Rezultatele activităţii sunt susţinute ştiinţific prin articolul:

"Methodology and Backend Flow Optimization for 3D", publicat în *Journal of Engineering Science*, Vol. XXXI, no. 2, 2024, DOI: 10.52326/jes.utm.2024.31(2).04 (https://repository.utm.md/bitstream/handle/5014/28854/JES-2024-N2-p39-47.pdf).

Activitatea desfăşurată a contribuit la consolidarea competenţelor companiei în domeniul tehnologiilor avansate de integrare 3D şi al optimizării fluxurilor de proiectare pentru circuite integrate.

Prezentul certificat a fost emis pentru a atesta implicarea şi contribuţia doamnei Pletea Ionica-Marcela în cadrul activităţilor de cercetare şi dezvoltare desfăşurate la Silicon Service S.R.L în perioada 2023–2025.

SILICON SERVICE SRL
Prin Administrator
Aviv Consulting & Management SRL
AVIV ILAN

Întocmit,
Director Resurse Umane
Roxana Andreea Biicu

SILICON SERVICE SRL
CUI: RO 12552530
Nr. Reg.: J22/957/1999
Calea Chisinaului nr. 29,
Corp C40,, et.2
700290 Iasi, Romania
Tel: +40 745 064 202
office@silicon-service.com
www.silicon-service.com
Nr. de inregistrare 319 / 18.05.2026

CERTIFICATE OF IMPLEMENTATION

Hereby,

Silicon Service S.R.L., headquartered in Iași, Romania, certifies that Ms. **Pletea Ionica-Marcela,** an employee of the company since 2009 to the present, carried out research and development activities during the period **2023–2025** in the field of three-dimensional integrated circuits (3D ICs), BACKEND flow optimization, and physical implementation process automation using EDA tools.

During this period, Ms. Pletea Ionica-Marcela contributed to the development and optimization of an implementation methodology for 3D integrated circuits, based on adapting existing 2D flows and automating physical design stages, including floorplanning, power planning, placement, clock tree synthesis, routing, as well as physical and logical verification.
The research activity included the development of scripts and automation methods for optimizing the BACKEND flow dedicated to 3D implementation, with a focus on reducing design time, early error detection, and improving integration performance.

The results of this activity are scientifically supported by the article:

"**Methodology and Backend Flow Optimization for 3D**", published in the *Journal of Engineering Science*, Vol. XXXI, no. 2, 2024, DOI: 10.52326/jes.utm.2024.31(2).04 (https://repository.utm.md/bitstream/handle/5014/28854/JES-2024-N2-p39-47.pdf).

The conducted activity contributed to strengthening the company's competencies in the field of advanced 3D integration technologies and design flow optimization for integrated circuits.

This certificate is issued to attest the involvement and achievements of Ms. Ionica-Marcela Pletea in the research and development activities carried out at Silicon Service S.R.L during the period 2023–2025.

SILICON SERVICE SRL
By Administrator
Aviv Consulting & Management SRL
AVIV ILAN

Prepared by
HR Manager
Roxana Andreea Biicu

UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI" DIN IAȘI
REGISTRATURA
-15-
Nr. 18732 / 27.05.2026

ADEVERINȚĂ

Se adevereşte prin prezenta că rezultatele cercetării realizate în cadrul tezei de doctorat intitulată **„Proiectarea 3D a circuitelor integrate utilizând fluxuri 2D de proiectare digitală"**, elaborată de doctoranda **Pletea Ionica-Marcela** în cadrul **Universității Tehnice a Moldovei**, specialitatea **233.01 – Nano-microelectronică și optoelectronică**, vor fi valorificate și implementate în activitatea didactică desfășurată în cadrul programului de master **„Information Technologies for Telecommunications"** al **Universității Tehnice „Gheorghe Asachi"** din Iași – **Facultatea de Electronică, Telecomunicații și Tehnologia Informației**, prin elaborarea și utilizarea unor materiale didactice, aplicații practice și lucrări de laborator aferente disciplinei **„Smart Sensors and Sensor Networks"**.

Contribuțiile științifice și tehnice dezvoltate în cadrul tezei permit introducerea în procesul educațional a unor concepte moderne privind proiectarea circuitelor integrate pe verticală (3D IC), utilizând fluxuri și instrumente actuale de proiectare digitală 2D existente în industrie.

Materialul didactic rezultat va avea ca obiectiv:

- familiarizarea studenților cu etapele unui flux modern de proiectare digitală;
- înțelegerea conceptelor de integrare verticală și interconectare 3D;
- prezentarea adaptării programelor convenționale 2D pentru dezvoltarea unor structuri integrate complexe;
- dezvoltarea competențelor practice privind proiectarea, verificarea și validarea circuitelor integrate digitale;
- apropierea activităților de laborator de cerințele și tehnologiile utilizate în mediul industrial actual.

În cadrul activităților practice și de laborator, studenții vor avea posibilitatea de a analiza și utiliza exemple și metodologii inspirate direct din cercetarea realizată în teză, facilitând astfel transferul rezultatelor științifice către procesul educațional și formarea unor competențe avansate în domeniul proiectării circuitelor integrate moderne.

Implementarea acestor materiale didactice contribuie la modernizarea conținutului disciplinei **„Smart Sensors and Sensors Networks"**, prin introducerea unor teme actuale și relevante pentru domeniul microelectronicii și al proiectării sistemelor integrate avansate.

De asemenea capitolul 5 din teză, intitulat **"Perspective în dezvoltarea structurilor de calcul AI în tehnologia monolitic 3D"** a fost conceput și editat în cadrul proiectului de cercetare "Regions and (E)DIHs alliance for AI-atthe-Edge adoption by European Industry 5.0 Manufacturing SMEs" , proiect finanțat prin programul European **HORIZON-CL4-2022-TWIN-TRANSITION-0**, codul proiectului este **101092069**, acronim **REDGIO 5.0**. Adresa web a proiectului este: https://www.airedgio5-0.eu/. Rezultatele și direcțiile de cercetare dezvoltate în cadrul acestui capitol contribuie la extinderea conținutului educațional aferent disciplinei **„Smart Sensors and Sensors Networks"**, prin introducerea unor concepte moderne privind structurile hardware dedicate inteligenței artificiale și tehnologiile avansate de integrare monolitic 3D.

Data: 27-05-2026

Director Departament
Conf. univ. dr. ing. Cristian Achion

Decan
Prof. univ. dr. ing. Daniela Tărniceriu

ROMÂNIA * UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI" DIN IAȘI * FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI * MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

Coordonator disciplină
Conf. univ. dr. ing. Marius Hagan

Bulevardul Carol I nr. 11A, 700506, Iași
Tel: 40 232 270 041 | Fax: 40 232 217 720 | www.etti.tuiasi.ro | decanat@etti.tuiasi.ro

INGENIUM European University

CERERE DE BREVET DE INVENTIE

Nr. referinta solicitant/mandatar	Registratura OSIM (numarul si data primirii):
	A/00521 12 NOV 2025

Se completeaza de catre OSIM	
Numarul cererii de brevet de invenţie	
Data primirii la Registratura Generala a OSIM	
Data de depozit	
Data primirii părţii lipsa la Registratura Generala a OSIM	
Data de depozit după primirea părţii lipsă la Registratura Generală a OSIM	
Data primirii cererii de retragere a părţii lipsa la Registratura Generala a OSIM	
Data de depozit atribuita cererii de brevet	

1.Solicitanţi (nume şi prenume/denumire,adresă de domiciliu/sediu,telefon, fax, e- mail)

Pletea Ionica Marcela,

Str. Nicolae Oblu Nr 16 Bl B4 Ap 18, loc. Iasi, jud. Iasi

Email : i_pletea@yahoo.com

☐ continuare pe pag.3

2. Solicitam în baza Legii nr. 64/1991 privind brevetele de invenţie, republicată, modificată prin Legea nr.83/2014 privind invenţiile de serviciu acordarea unui brevet de invenţie cu titlul:

Arhitectură 3D monolitică hibridă

2.1. Solicitantul este îndreptăţit la depunerea cererii de brevet de invenţie în baza :

X Legii nr. 64/1991 privind brevetele de invenţie, republicată;

☐ Legii nr.83/2014 privind invenţiile de serviciu.

☐ unui contract de cercetare

2.2. Referinta la o cerere depusă anterior (numar, data de depozit, tara/oficiul):

3. **Declarăm că inventatorii sunt cei desemnaţi în formularul „ Declaraţie conţinând desemnarea inventatorilor"** X anexat ☐ care va fi transmis ulterior

4. Rezumatul invenţiei se publică împreună cu figura numarul: 1

5. Revendicăm prioritatea convenţională (stat, numar, data depozit):

6. Revendicăm prioritatea internă (numar cerere de brevet, data depozit):

FORM. B 01 - citiţi Ghidul de completare

7. Cererea de brevet este:

- ☐ divizionara din cererea de brevet (numar, data depozit):
- ☐ transformată din cererea de brevet european (nr., data de depozit)
- ☐ rezultată din conversia unei cereri de înregistrare a unui model de utilitate (nr. cerere înreg, dată depozit)

8. La data depunerii cererii solicităm următoarele proceduri:

8.1. Publicarea de urgenţă a cererii de brevet de invenţie	☐
8.2. Întocmirea unui raport de documentare	☐
8.3. Întocmirea unui raport de documentare cu opinie scrisă privind brevetabilitatea	☐
8.4. Examinarea cererii cerută la data de depozit	☐

9. Mandatar autorizat (denumire, sediu) :

prin ☐ procura ; sau ☐ procura generală (nr, dată):

10. Solicitantul/reprezentantul desemnat de solicitant (nume, prenume / denumire, adresă/ sediu) pentru corespondenţa cu OSIM:

11. Semnătură solicitanţi/mandatar autorizat:

Semnătură

L.S. Data 10.11.2025

12. Documente depuse la OSIM de solicitant/mandatar			**14.** Documente primite la OSIM	
12.1. Formular de cerere	în ... exemplare, a ... file	☐	în 3 exemplare, a 3 file	☒
12.2. Descriere	în ... exemplare,a ...file	☐	în 3 exemplare, a 5 file	☒
12.3. Revendicari	în ... exemplare, a .. file	☐	în 3 exemplare, a 1 file	☒
12.4. Desene	în ... exemplare, a .... file	☐	în 3 exemplare, a 2 file	☒
12.5. Rezumat	în ... exemplare, a...file	☐	în 3 exemplare, a 1 file	☒
12.6. Lista de secvente de nucleotide si/sau aminoacizi, parte a descrierii		☐		☐
12.6.1. pe suport hârtie	în ....exemplare, a .... file	☐	în ....exemplare, a .... file	☐
12.6.2. pe suport electronic	tip ..... , în .... exemplare	☐	tip..... , în ....exemplare	☐
12.7. Actul din care rezultă dreptul la acordarea brevetului		☐		☐
12.8. Declaraţia conţinând desemnarea inventatorilor	a...	☐	3 ex a 1 file	☒
12.9. Procura/copie procura generala	a ... file	☐	a .... file	☐
12.10. Document referitor la plata taxelor	a ... file	☐	a .... file	☐
12.11. Act privind solicitarea reducerii taxelor	a .... file	☐	a .... file	☐
12.12. Autorizatia privind transmiterea dreptului de prioritate	a .... file	☐	a .... file	☐
12.13. Act de prioritate	a .... file	☐	a .... file	☐
12.14. Act referitor la depozitul microorganismului/materialului biologic		☐	a .... file	☐
12.15. Document privind o divulgare a inventiei	a ...file	☐	a .... file	☐
12.16. Copie/traducere a cererii anterioare de la rubrica 2.2.	a....file	☐	a .... file	☐
12.17. Alte documente	a ....file	☐	GDPR a 1 file	☒

13. Persoana care a depus cererea, alta decât solicitantul, mandatarul (nume, prenume, act identitate) : Ceobanu Alexandru, CI seria MZ nr. 897176, eliberat de SPCLEP Harlau la data de 07.08.2020

Confirmare OSIM (nume, prenume şi dată)

L.S.



FORM. B 01 - citiţi Ghidul de completare

Declaraţie

conţinand desemnarea inventatorilor invenţiei cu titlul:

Arhitectură 3D monolitică hibridă

……care face obiectul cererii de brevet cu nr…… şi data de depozit ……

Aceasta declarţie este făcută si depusă la OSIM până la data luării unei hotărâri privind cererea de brevet de inventie

Nume si prenume: Pletea Ionica Marcela

Adresa de domiciliu___ Str. Nicolae Oblu Nr 16 Bl B4 Ap 18, loc. Iasi, jud. Iasi

Locul de munca la data crearii inventiei: Silicon Service SRL, Iasi

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Nume si prenume: _____

Adresa de domiciliu_____

Locul de munca la data crearii inventiei: _____

Alti inventatori sunt înscrisi într-o pagină următoare pe un formular identic cu acesta

☐

Semnatura solicitantului sau a mandatarului autorizat *(numele si prenumele precum si calitatea persoanei cu capacitate de reprezentare a solicitantului sau a mandatarului autorizat):*

_____

_____

_____

*Semnatura:*_____ *Data:__10.11.2025__*

L.S. _____

FORM. B 01 - cititi Ghidul de completare

CERERE DE BREVET DE INVENTIE

OSIM

Nr. referinta solicitant/mandatar	Registratura OSIM (numarul si data primirii):
5566/03.06.2026	A/00395 17 JUN 2026

Se completeaza de catre OSIM	
Numarul cererii de brevet de inventie	
Data primirii la Registratura Generala a OSIM	
Data de depozit	
Data primirii partii lipsa la Registratura Generala a OSIM	
Data de depozit după primirea părţii lipsă la Registratura Generală a OSIM	
Data primirii cererii de retragere a partii lipsa la Registratura Generala a OSIM	
Data de depozit atribuita cererii de brevet	

1.Solicitanţi (nume şi prenume/denumire,adresă de domiciliu/sediu,telefon, fax, e- mail)

UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" din IAŞI, Judeţul Iaşi, cod 700050, Str.Prof.dr.doc. Dimitrie Mangeron, nr.67, telefon 0232-278683, fax 0232-211667, e-mail:rectorat@staff.tuiasi.ro

CF 4701606

☐ continuare pe pag.3

2. Solicităm în baza Legii nr. 64/1991 privind brevetele de invenţie, republicată, modificată prin Legea nr.83/2014 privind invenţiile de serviciu acordarea unui brevet de invenţie cu titlul:

METODĂ DE REALIZARE A DISPOZITIVELOR MEMRISTIVE ÎN STRUCTURI SEMICONDUCTOR MONOLITICE 3D

2.1. Solicitantul este îndreptăţit la depunerea cererii de brevet de invenţie în baza :

- ☐ Legii nr. 64/1991 privind brevetele de invenţie, republicată;
- ☒ Legii nr.83/2014 privind invenţiile de serviciu;
- ☐ unui contract de cercetare.

2.2. Referinta la o cerere depusă anterior (numar, data de depozit, tara/oficiul):

3. Declarăm că inventatorii sunt cei desemnaţi în formularul „ Declaraţie conţinând desemnarea inventatorilor" ☒ anexat ☐ care va fi transmis ulterior

4. Rezumatul invenţiei se publică împreună cu figura numarul: Figura 1

5. Revendicăm prioritatea convenţională (stat, numar, data depozit):

6. Revendicăm prioritatea internă (numar cerere de brevet, data depozit):

7. Cererea de brevet este:

- ❑ divizionara din cererea de brevet (numar, data depozit):
- ❑ transformată din cererea de brevet european (nr., data de depozit)
- ❑ rezultată din conversia unei cereri de înregistrare a unui model de utilitate (nr. cerere înreg, dată depozit)

FORM. B 01 - citiţi Ghidul de completare

8. La data depunerii cererii solicităm următoarele proceduri:

8.1. Publicarea de urgenţă a cererii de brevet de invenţie ☐

8.2. Întocmirea unui raport de documentare ☐

8.3. Întocmirea unui raport de documentare cu opinie scrisă privind brevetabilitatea ☐

8.4. Examinarea cererii cerută la data de depozit ☐

9. Mandatar autorizat (denumire, sediu) :

prin ☐ procura ; sau ☐ procura generală (nr, dată):

10. Solicitantul/reprezentantul desemnat de solicitant (nume, prenume / denumire, adresă/ sediu) pentru corespondenţa cu OSIM: **UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" din IAŞI**, Judeţul Iaşi, cod 700050, Str.Prof.dr.doc. Dimitrie Mangeron, nr.67, telefon 0232-278683, fax 0232-211667, e-mail:rectorat@staff.tuiasi.ro

În atenţia doamnei ing.Gavril Georgeta, georgeta.gavril@staff.tuiasi.ro, 0723827612

11. Semnătură solicitanţi/mandatar autorizat:

RECTOR,
Prof.univ.dr.ing. Dan Caşcaval

Semnătură
L.S.

Data 03 IUNIE 2026

12. Documente depuse la OSIM de solicitant/mandatar			**14.** Documente primite la OSIM	
12.1. Formular de cerere	în **3** exemplare, a **1** filă	☒	în 3 exemplare, a 1 file	☒
12.2. Descriere	în **3** exemplare, a **5** file	☒	în 3 exemplare, a 5 file	☒
12.3. Revendicări **nr.4**	în **3** exemplare, a **1** filă	☒	în 3 exemplare, a 1 file	☒
12.4. Desene	în **3** exemplare, a **2** file	☒	în 3 exemplare, a 2 file	☒
12.5. Rezumat	în **3** exemplare, a **1** file	☒	în 3 exemplare, a 1 file	☒
12.6. Lista de secvente de nucleotide si/sau aminoacizi, parte a descrierii		☐		☐
12.6.1. pe suport hârtie	în ....exemplare, a .... file	☐	în ....exemplare, a .... file	☐
12.6.2. pe suport electronic	tip .... , în .... exemplare	☐	tip..... , în ....exemplare	☐
12.7. Actul din care rezultă dreptul la acordarea brevetului		☐		☐
12.8. Declaraţia conţinând desemnarea inventatorilor 3 exemplare a **1** file		☒	3 ex. a 1 file	☒
12.9. Procura/copie procura generala	a ... file	☐	a .... file	☐
12.10. Document referitor la plata taxelor	a ... file	☐	a .... file	☐
12.11. Act privind solicitarea reducerii taxelor **Act ad.**	a **1** file	☒	a 1 file	☒
12.12. Autorizatia privind transmiterea dreptului de prioritate	a .... file	☐	a .... file	☐
12.13. Act de prioritate	a .... file	☐	a .... file	☐
12.14. Act referitor la depozitul microorganismului/materialului biologic		☐	a .... file	☐
12.15. Document privind o divulgare a inventiei	a ...file	☐	a .... file	☐
12.16. Copie/traducere a cererii anterioare de la rubrica 2.2.	a....file	☐	a .... file	☐
12.17. Alte documente Adresa	a **1** file	☒	Adresă a 1 file	☒

13. Persoana care a depus cererea, alta decât solicitantul, mandatarul (nume, prenume, act identitate) :

Confirmare OSIM (nume, prenume şi dată)
L.S.

FORM. B 01 - citiţi Ghidul de completare

Declaraţie

conţinand desemnarea inventatorilor invenţiei cu titlul:

METODĂ DE REALIZARE A DISPOZITIVELOR MEMRISTIVE ÎN STRUCTURI SEMICONDUCTOR MONOLITICE 3D

.....care face obiectul cererii de brevet cu nr..... şi data de depozit .....
Aceasta declaraţie este făcută si depusă la OSIM până la data luării unei hotărâri privind cererea de brevet de inventie

Nume si prenume: ***PLETEA IONICA-MARCELA***
Adresa de domiciliu Str.Prof.Nicolae Oblu, nr.16, bl.B4, et.2, ap.18, Judeţul Iaşi, Municipiul Iaşi
Locul de munca la data crearii inventiei: UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" din IAŞI
Nume si prenume:
Adresa de domiciliu
Locul de munca la data crearii inventiei:
Nume si prenume:
Adresa de domiciliu
Locul de munca la data crearii inventiei:
Nume si prenume:
Adresa de domiciliu
Locul de munca la data crearii inventiei:
Alti inventatori sunt înscrisi într-o pagină următoare pe un formular identic cu acesta ☐
Semnatura solicitantului sau a mandatarului autorizat *(numele si prenumele precum si calitatea persoanei cu capacitate de reprezentare a solicitantului sau a mandatarului autorizat):*

UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" din IAŞI

RECTOR,
prof.univ.dr.ing. Dan Caşcaval

*Semnatura:*_____

L.S. _____ *Data: 03 IUNIE 2026*

MULȚUMIRI

Exprim profunda mea recunoștință conducătorului de doctorat, **dr. prof. univ. Șontea Victor**, pentru îndrumarea competentă, susținerea permanentă și profesionalismul deosebit de care am beneficiat în cadrul realizării acestei teze.

Păstrez o amintire luminoasă și un respect profund pentru dedicarea și generozitatea sa academică. Trecerea sa prematură în neființă reprezintă o pierdere imensă pentru comunitatea științifică și pentru toți cei care au avut privilegiul de a-l cunoaște și de a lucra cu dumnealui.

Aduc sincere mulțumiri membrilor comisiei de îndrumare:
dr. hab., prof. univ. Lupan Oleg,
dr. hab., prof. univ. Trofim Viorel,
dr. hab., prof. univ. Tighineanu Ion
pentru expertiza științifică valoroasă, observațiile pertinente și sprijinul acordat pe parcursul cercetărilor. Rigoarea și profesionalismul dumnealor au avut un rol esențial în finalizarea acestei teze.

Adresez mulțumiri **Directorului Școlii Doctorale UTM, dr., conf. univ. Siminiuc Rodica**, pentru suportul instituțional și pentru cadrul academic care a făcut posibilă desfășurarea studiilor doctorale.

Mulțumesc colegilor și colaboratorilor pentru discuțiile utile, pentru sprijinul tehnic și pentru cooperarea științifică oferită pe întreg parcursul acestor ani.

Cu profund respect,
Pletea Ionica-Marcela

Chișinău, 2026

DECLARAȚIA PRIVIND ASUMAREA RĂSPUNDERII

Subsemnata, declar pe răspundere personală că materialele prezentate în teza de doctorat sunt rezultatul propriilor cercetări și realizări științifice. Conștientizez că, în caz contrar, urmează să suport consecințele în conformitate cu legislația în vigoare.

Pletea Ionica-Marcela

Semnătura:

Data:

CURRICULUM VITAE

Date personale:

Nume: Pletea

Prenume: Ionica Marcela

Naționalitatea: Română

Data nașterii: 27/10/1976

Studii:

- 2016 – 2020: Studii doctorat, Școala doctorală "Știința Calculatoarelor, Electronică și Energetică", UTM, Chișinău, specialitatea 233.01 " Nano-microelectronică și optoelectronică"
- 2002 – 2003: Master – Convertoare de putere, Facultatea de Electronică, Telecomunicații și Tehnologia Informației din cadrul Universității Tehnice „Gheorghe Asachi" din Iași
- 1997 – 2002: Licență – Facultatea de Electronică, Telecomunicații și Tehnologia Informației din cadrul Universității Tehnice „Gheorghe Asachi" din Iași

Experiență profesională:

- Octombrie 2019 – prezent: Asistent universitar, Universitatea Tehnică „Gheorghe Asachi" din Iași

  Predare laborator: „Electronică digitală" și „Aparate electronice de măsurare și control"
- Noiembrie 2018 – prezent: Director Departament Training, S.C. Silicon Service S.R.L., Iași
- Iunie – Octombrie 2013 – Inginer proiectare digitală, Subcontractor IMEC Leuven Belgia
- Iulie 2009 – prezent – Inginer proiectare digitală, S.C. Silicon Service S.R.L., Iași
- Octombrie 2004 – Mai 2009 – Inginer proiectare digitală, S.C. Asic Art S.R.L., Iași

Cursuri predate:

- Circuite Digitale – S.C. Silicon Service S.R.L., Iași

Domeniile de activitate științifică:

- analiza convertoarelor asincrone RNSIC cu capacitori pe partea de curent alternativ, adresând probleme legate de armonici, eficiență și topologii pentru aplicații de medie și înaltă putere.
- proiectarea, optimizarea și integrarea circuitelor integrate tridimensionale (3D IC), cu accent pe arhitecturi monolitice și metodologii moderne de backend digital, precum sinteza și optimizarea arborelui de clock (CTS).
- integrarea heterogenă în circuite 3D, vizând combinarea straturilor funcționale diferite și optimizarea interconexiunilor verticale pentru reducerea ariei și a consumului de energie.

Proiecte industriale implementate:

1. Bloc Digital – plasare și rutare - cu UMC 130nm technology, 6 metal layers, ~200k gates, 6 power domains, (building CPF file, floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extraction).
2. Bloc Digital – plasare și rutare for 3 versions - cu MagnaChip 0.35 um technology, 4 metal layers, ~ 2700 gates (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting).
3. Bloc Digital – plasare și rutare for 2 versions - cu MagnaChip 0.35 um technology, 4 metal layers, ~ 9000 gates (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting).
4. Bloc Digital incluzând IO cells – plasare și rutare pentru 3 versiuni - cu MagnaChip 0.35 um technology, 4 metal layers, ~ 16000 gates (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting).
5. Bloc Digital incluzând IO cells – plasare și rutare pentru 4 versiuni - cu MagnaChip 0.35 um technology, 4 metal layers, ~ 45000 gates (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting.
6. Bloc Digital – plasare și rutare - cu Atmel 0.35 um technology, 3/4 metal layers (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting).

7. Bloc Digital – plasare și rutare - cu Atmel 0.8 um technology, 3/4 metal layers (floorplan, power plan, CTS, place and route, timing analysis, power analysis, SDF files extracting), having complicated shapes.

Participări la manifestări științifice (naționale și internaționale):

1. Participare la Sesiunea științifică organizată de Universitatea „Politehnica" din Timișoara, Facultatea de Electronică și Telecomunicații, 2007, cu prezentarea lucrărilor:
    - Two-Quadrant Converter with RNSIC Analysis Having Capacitors on the AC Side;
    - An Analysis of Asymmetrical RNSIC Converter with Capacitors on the AC Side for High and Medium Power;
2. Participare la Sesiunea științifică organizată de Universitatea „Politehnica" din Timișoara, Facultatea de Electronică și Telecomunicații, 2008, cu prezentarea lucrării A Comparative Study between Classical Three-Phase Rectifier Configuration with Passive Filters and Converter Configurations with Small Harmonic Contents Having Capacitors on the AC Side.
3. Participare la Conferința tehnico-științifică a studenților/ masteranzilor/ doctoranzilor, Chișinău, 2019, cu prezentarea lucrării Impactul integrării 3D asupra Construirii Arborelui de Clock;
4. Participare la Conferința Electronics, Communications and Computing, ediția a 10-a, Chișinău, 2019, cu prezentarea lucrării CTS optimization on 3D integration;
5. Participare la Conferința Telecommunications, Electronics and Informatics, ediția a 6-a, Chișinău, 2018, cu prezentarea lucrării Impactul conexiunilor verticale asupra ariei în integrarea 3D;
6. Participare la Conferința Materials Science and Condensed Matter Physics, ediția a 10-a, Chișinău, 2024, cu prezentarea lucrării Heterogenous integrations on 3D integrated circuits;
7. Participare la Conferința Electronics, Communications and Computing (ECCO), ediția a 13-a, Chișinău, 2024, cu prezentarea lucrării Design and Partitioning Across 2 Strata in 3D IC architectures;

8. Participare la cea de-a XVII-a ediție a Simpozionului Internațional privind Semnale, Circuite și Sisteme, Iași, România, 17–18 iulie 2025, cu prezentarea lucrării Neural Grain implemented in monolithic 3D technology.

Cunoașterea limbilor:

- Limba maternă: Română
- Limba engleză: nivel intermediar superior (CEFR) - B2 (citire, scriere, conversație)

Date de contact de serviciu:

Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Bulevardul Carol I nr. 11A, Universitatea Tehnică „Gheorghe Asachi” din Iași, E-mail: **ipletea@etti.tuiasi.ro**, tel. +0742 147 191 .

PARTICIPARE LA PROIECTE ȘTIINȚIFICE INTERNAȚIONALE

Tehnologia monolitic 3D în proiecte de cercetare:

Abordarea activitătii de cercetare pe perioada activitătii doctorale a fost axată în principal pe studiul tehnologiei monolitice 3D și exemplificarea unor implementări de circuite concrete deschizând însă și o direcție de a continua prin conceperea și implementarea de structuri monolitice 3D ce vor îngloba rețele neuronale de tip MLP (Multilayer Perceptrion).

Capitolul 5 din teză, intitulat "Perspective în dezvoltarea structurilor de calcul AI în tehnologia monolitic 3D" a fost conceput și editat în cadrul proiectului de cercetare "**Regions and (E)DIHs alliance for AI-atthe-Edge adoption by European Industry 5.0 Manufacturing SMEs**" , proiect ce face parte din programul European HORIZON-CL4-2022-TWIN-TRANSITION-0, codul proiectului este 101092069, acronim REDGIO 5.0. Adresa web a proiectului este: https://www.airedgio5-0.eu/

Proiectul REDGIO 5.0 este coordonat de către Universitatea Politehnică din Milano și cuprinde sub aceeași umbrelă 44 de parteneri din 18 ţări europene. Unitersitatea Tehnică "Gheorghe Asachi" din Iași este partener în acest proiect și are ca și menire conceperea și implementarea (până la nivelul TRL4) a unui echipament de detecţie a defectelor în haine, acest echipament numindu-se QUAD-AI@E. Doctoranda Ionica Pletea face parte din echipa proiectului ca și asistent de cercetare. Cerința esențială a proiectului este de a utiliza unităti de procesare locală în care să ruleze algoritmi de inteligenţă artificială (în engleză: AI edge computing). Unitatea de procesare aleasă este Jetson Nano care se regăsește sub forma unei plăci de aplicaţii dezvoltată de NVIDIA [ref_JN]. Medoda de detectie a defectelor in haine se bazează pe procesări de imagini astfel că solicitările de resurse sunt mari, atât in ceea ce priveste puterea de calcul cât si din punct de vedere al consumului energetic iar aceste aspecte devin critice atunci când echipamentul trebuie să folosească, spre exemplu, ca și sure de alimentare baterii sau acumulatori. Acestea sunt premisele de la care s-a pornit atunci când s-a inițiat ideea de a modela structuri neuromorfice 3D care să fie capabile să proceseze local algoritmi AI cu un consum minim de energie și să ocupe un spaţiu minim. Un element de noutate este lansarea și prezentarea conceptului de "Neural Grain" (grăunte neuronal) ce reprezintă o structutura AI hardware prin care se mimează o strucură a creierului biologic. Acest concept este expus în lucrarea "Neural Grain implemented in monolithic 3D technology" [ref_SCS_2025]. A fost modelată o structură cubică 3D a unui "Neural Grain" și s-a elaborat un cod VHDL ce este implementat, prin tool-ul de sinteză VIVADO, într-un circuit FPGA din familia ARTIX 7. În paralel, aceeași structură NG 4x4x1 a fost modelată intr-o

tehnologie ce emulează tehnologia monolithic 3D prin parcurgerea pașilor de sinteză utilizând tool-ul Genus de la Cadence. Câteva concluzii ar fi următoarele: topologiile neuromorfice implementate in tehnologie 3D modeleaza arhitecturi de calcul ce mimează structuri ale creierului biologic rezultând circuite integrate de înaltă performanţă cu puterea ridicată de calcul, consum redus de energie și volum restrâns. Au fost identificate preocupări stiintifice de modelare a unor arhitecturi neuromorfice, aceste preocupari sunt evidentiate si in câteva referinte amintite in teză dar menţionăm și câteva programe de cercetare ample ce sunt focusate pe dezvoltare de circuite monolitice 3D: NeuRAM3 (Horizon 2020 RIA, 2016–2019) [ref_NeuRAM], NimbleAI (Horizon Europe, €10 M, started Nov 2022) [ref_NimbleAI] sau NEUROPULS (Horizon Europe) [ref_NEUROPULS]

Pornind de la rezultatele cercetarilor efectuate pe perioada studiilor doctorale se vor urmări oportunități de cooperare în proiecte de cercetare între instituții din Republica Moldova și parteneri de top din Uniunea Europeană.

Referinţe:

[ref_JN] https://developer.nvidia.com/embedded/jetson-nano, accesat la data de 11 iunie 2025

[ref_SCS_2025] Pleatea, I.M., et al, Neural Grain implemented in monolithic 3D technology, SCS, 2025, accesat la data de 11 iunie 2025

[ref_NeuRAM]https://www.imm.cnr.it/projects/neuram3-neural-computing-architectures-advanced-monolithic-3d-vlsi-nano-technologies?utm_source=chatgpt.com, accesat la data de 11 iunie 2025

[ref_NimbleAI] https://www.eetimes.eu/eu-funded-nimbleai-to-deliver-3d-neuromorphic-chip/?utm_source=chatgpt.com, accesat la data de 11 iunie 2025

[ref_NEUROPULS]https://www.iis.fraunhofer.de/en/ff/sse/ic-design/neuromorphic-computing.html?utm_source=chatgpt.com, accesat la data de 11 iunie 2025

CONTRACT INDIVIDUAL DE MUNCĂ

încheiat şi înregistrat sub **nr. 26136 / 14.03.2025** în registrul general de evidenţă a salariaţilor

A. PĂRŢILE CONTRACTULUI

Angajator – Universitatea Tehnică "Gheorghe Asachi" din Iaşi (TUIASI), cu sediul în Iaşi, Bld. Profesor Dimitrie Mangeron, nr.67, cod postal 700050, cod fiscal 4701606, telefon 0232/701111, reprezentată legal prin prof.univ.dr.ing. Dan Caşcaval, în calitate de Rector,
şi

salariata – asist.cercet. **Pletea Ionica-Marcela**, domiciliată în Mun.Iaşi, str.Prof. Nicolae Oblu nr.16, bl.B4, et. 2, ap.18, posesoare a cărţii de identitate seria MZ, nr. 841839, eliberată de SPCLEP Iaşi, la data de 15.11.2019, CNP: 2761027392925, având în vedere Decizia nr. 667/14.03.2025, am încheiat prezentul contract individual de muncă în următoarele condiţii asupra cărora am convenit:

B. OBIECTUL CONTRACTULUI:

Obiectul contractului îl constituie prestarea de către angajat a muncii corespunzătoare încadrării sale, îndeplinirea atribuţiilor şi efectuarea activităţilor din cadrul proiectului nr. 101092069/30.11.2022, depus în cadrul programului Horizon Innovation Action, finanţat prin European Health and Digital Executive Agency, cu titlul**"Regions and (E)DIHs alliance for AI-atthe – Edge adoption by European Industry 5.0 Manufacturing SMEs"**, acronim **AI REDGIO 5.0,** responsabil proiect – şef lucr.dr.ing. Hăgan Marius-Gheorghe.

C. DURATA CONTRACTULUI

Durata contractului este **determinată.**

Durata contractului este pentru perioada cuprinsă între data de **17.03.2025** şi data de **31.12.2025**.

D. FELUL MUNCII

Functia: ACS */ **COR: 215123.***
Activitatea se desfăşoară la sediul Universităţii Tehnice "Gheorghe Asachi" din Iaşi.
Activitatea prestată se desfăşoară in condiţii normale de muncă, potrivit Legii nr.263/2010 privind sistemul unitar de pensii publice.

E. ATRIBUŢIILE POSTULUI

Atribuţiile postului sunt prevăzute în fişa postului, anexă la contractul individual de muncă.

F. DURATA MUNCII

a) O fracţiune de normă: **nr. ore/lună** :***20,***

b) Repartizarea timpului de muncă se face **inegal,** în funcţie de activităţile prestate pentru realizarea obiectivelor proiectului, inclusiv în zilele de sâmbătă şi duminică, cu respectarea prevederilor legale.

G. SALARIZAREA

1. Salariul brut lunar este **de 2.400 *lei,*** calculat în raport cu tariful orar de ***120 lei/oră***;
2. Salariul brut lunar se calculează funcţie de numărul de ore/lună admise pentru activitatea desfăşurată şi tariful orar stabilit pe categorii de activităţi, în conformitate cu plafoanele stabilite prin acte normative sau alte reglementări legale în vigoare;
3. Salariul brut lunar de plată se acordă proporţional cu activitatea desfăşurată în luna respectivă, funcţie de numărul de ore efectiv lucrate, conform fişei de pontaj întocmite, avizată şi semnată de către directorul de proiect;
4. Plata efectivă a cheltuielilor salariale pentru activitatea depusă (contribuţii salariale, salariul net) se face în baza *Referatului privind înregistrarea timpului utilizat pentru activităţile desfăşurate şi plata drepturilor salariale pentru aceste activităţi*, în limita fondului disponibil existent aferent contractului de finanţare;

H. DREPTURI ŞI OBLIGAŢII ALE PĂRŢILOR PRIVIND SECURITATEA ŞI SĂNĂTATEA ÎN MUNCĂ

Să respecte măsurile de securitate şi sănătate a muncii în unitate, precum şi normele generale de prevenire şi stingere a incendiilor;

I. DREPTURI ŞI OBLIGAŢII GENERALE ALE PĂRŢILOR

1. Salariatul are, în principal, următoarele drepturi:
   a.dreptul la salarizare pentru munca depusă;
   b.dreptul la repaus zilnic şi săptămânal;
   c.dreptul la securitate şi sănătate în muncă.
2. Salariatului îi revin, în principal, următoarele obligaţii:
   a.obligaţiile de a îndeplini atribuţiile ce îi revin conform fişei postului;
   b.obligaţiile de a respecta disciplina muncii;
   c.obligaţiile de a respecta secretul de serviciu.
3. Angajatorul are, în principal, următoarele drepturi;
   a.să dea dispoziţii cu caracter obligatoriu pentru salariat, sub rezerva legalităţii lor;
   b.să exercite controlul asupra modului de îndeplinire a sarcinilor de serviciu;
   c.să constate săvârşirea abaterilor disciplinare şi să aplice sancţiunile corespunzătoare, potrivit legii şi regulamentului intern.
4. Angajatorului îi revin, în principal următoarele obligaţii.
   a.să înmâneze salariatului un exemplar din contractul de muncă;
   b.să elibereze, la cerere, un document care să ateste calitatea de salariat a solicitantului, respectiv activitatea desfăşurată de aceasta, durata activităţii, salariul, vechimea în muncă, în meserie şi specialitate;
   c.să asigure confidenţialitatea datelor cu caracter personal ale salariatului.

J. DISPOZIŢII FINALE

1. Prezentul contract intra în vigoare la data semnării de către ambele părţi.
2. Prezentul contract poate înceta, înainte de termen, în condiţiile în care din motive obiective nu se mai poate asigura disponibilul/salarizarea din contractul de cercetare.
3. Orice modificare privind clauzele contractuale în timpul executării contractului individual de muncă impune încheierea unui act adiţional la contract, conform dispoziţiilor legale, cu excepţia situaţiilor în care o asemenea modificare este prevăzută în mod expres de lege.
4. Prevederile prezentului contract individual de muncă se completează cu dispoziţiile Legii nr. 53/ 2003 - Codul muncii cu modificările şi completările ulterioare, cu alte prevederi legislative în vigoare.
5. Prezentul contract individual de muncă s-a încheiat în două exemplare, câte unul pentru fiecare parte.

ANGAJATOR
Universitatea Tehnică "Gheorghe Asachi" din Iaşi
Reprezentant legal
RECTOR,
Prof.univ.dr.ing. Dan CAŞCAVAL

SALARIAT
Asist. cercet.
Ionica-Marcela Pletea

Responsabil proiect,
Şef lucr.dr.ing.
Marius-Gheorghe Hăgan

S.E.R.U.,
Întocmit,
Adm.patrim. C.Tincu

Prezentul contract încetează pe data de ....., în temeiul art. ..... din Legea nr.53/2003 republicată, în urma îndeplinirii procedurii legale.

Angajator,
.....

LISTA LUCRĂRILOR ȘTIINȚIFICE ȘI ȘTIINȚIFICO-METODICE-DIDACTICE PUBLICATE LA TEMA TEZEI

a doamnei Pletea Ionica-Marcela, doctorandă la forma de învățământ fără frecvență specialitatea 233.01 ” NANO-MICROELECTRONICĂ ŞI OPTOELECTRONICĂ”

LISTA LUCRĂRILOR PUBLICATE LA TEMA TEZEI

Articole în lucrările manifestărilor ştiinţifice incluse în bazele de date Web of Science și SCOPUS:

1. **Pletea, I.M.**, Wurman Zeev, Or-Bach Zvi, Şontea Victor. Monolithic 3D layout using 2D EDA for embedded memory-rich designs. În: *IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference S3S 2015*. 2015. New Jersey. p. 1-2. ISBN 978-150900259-7.
2. CHIPER, R., ALEXA, D., GORAS, T.C., PLETEA, I.V., **PLETEA, I.M.,** and ALEXANDRESCU, A. An Analysis of Asymmetrical RNSIC Converter with Capacitors on the AC Side for High and Medium Power. In : *2007 International Symposium on Signals, Circuits and Systems*. 2007. p. 1–4. ISBN 978-1-4244-0968-6.
3. **Pletea, I.M.**, Iolanda Alecsandrescu, Nicolae Lucanu, Cristian Aghion, Constantin Barabasa, Marius Zbancioc, Marius Hăgan, Victor Sontea. Neural Grain implemented in monolithic 3D technology. În: *In Proceedings of the 17th International Symposium on Signals, Circuits, and Systems (July 17–18, 2025)*, Faculty of Electronics, Telecommunications and Information Technology, Technical University of Iași, Romania. ISBN:979-8-3315-5300-5.

Articole în lucrările manifestărilor ştiinţifice incluse în alte baze de date acceptate de ANACEC:

4. **Pletea, I.M.**, Şontea Victor, Cojocaru Victor. Impactul conexiunilor verticale asupra ariei in integrarea 3D. În: *Conferința Internațională* “*Telecommunicații, Electronică și Informatică”*. Ed. 6. 2018. Chișinău, Republica Moldova. pp. 375-377. ISBN 978-9975-45-540-4.
5. **Pletea, I.M.**, Şontea Victor, Alecsandrescu Iolanda. CTS optimization on 3D integration. În: *Conferința Internațională* “*Telecommunicații, Electronică și*

Informatică”. Editia 10. 2019. Chișinău, Republica Moldova. p. 62. ISBN 978-9975-108-84-3.

6. **Pletea, I.M.**, Şontea Victor. Design and Partitioning Across 2 Strata în 3D IC architectures. În: *International Conference on Electronics, Communications and Computing*, IC ECCO 2024. Editia 13. 2024. Chișinău. Pp. 107-108. ISBN (pdf) 978-9975-64-480-8 (PDF).

Articole în lucrările manifestațiilor ştiinţifice incluse în Registrul materialelor publicate în baza manifestărilor științifice organizate din Republica Moldova:

7. **Pletea, I.M**., Impactul integrării 3D asupra Construirii Arborelui de Clock. În: *Conferinţa tehnico-ştiinţifică a studenţilor, masteranzilor şi doctoranzilor*. Vol.1. 2019. Chișinău, Republica Moldova. Vol. 1, pp. 397-399. ISBN 978-9975-45-588-6.
8. **Pletea, I.M.**, Şontea Victor. Heterogenous integrations on 3d integrated circuits. În: *Materials Science and Condensed Matter Physics, dedicated to the 60th anniversary from the foundation of the Institute of Applied Physics*. Ediția 10. 2024. Chișinău, Republica Moldova. p. 146. ISBN 978-9975-62-763-4

Articole în reviste din Registrul Naţional al revistelor de profil, cu indicarea categoriei:

9. **Pletea, I.M**., Methodology and backend flow optimization for 3D. În: *Journal of Engineering Science*, categoria B+. 2024, Vol. 31 Nr. 2 (iulie 2024), pp. 39–47. ISSN 2587-3474.

Articole în reviste științifice din România:

10. R. Chiper, D. Alexa, **I.M. Pletea**, A. Petrichei. A Comparative Study between Classical Three-Phase Rectifier Configuration with Passive Filters and Converter Configurations with Small Harmonic Contents Having Capacitors on the AC Side. În: *Buletinul Științific al Universității „Politehnica” din Timișoara, Seria Electronică și Telecomunicații, Tom* 53(67), vol. 53(67) Fascic ola 1, no. 1, pp. 19-22, 2008, ISSN 1583-3380.

Certificate de implementare eliberate de firma Silicon Service din România:

11. **Pletea I. M.,** Certificat de Implementare, cu număr de înregistrare 601/27.10.2025, a soluției de implementare a unui circuit 3D utilizând instrumentele clasice de proiectare 2D eliberat de firma Silicon Service SRL. ***https://silicon-service.com***
12. **Pletea I. M.,** Certificat de Implementare, cu număr de înregistrare 319/18.05.2026, a soluției de optimizare al fluxului BACKEND și al automatizării proceselor de implementare unui circuit 3D utilizând instrumentele clasice de proiectare 2D, eliberat de firma Silicon Service SRL. ***https://silicon-service.com.***

Cereri de brevete de invenție depuse la Registratura OSIM – Oficiul de Stat pentru Invenții și Mărci, România:

13. **Pletea I.M**., Arhitectura 3D monolitică hibridă. Cerere de brevet de invenție Numărul A2025/00521 din 12.11.2025.
14. **Pletea I., Metoda de realizare a dispozitivelor mmemristive în structuri Monolitice 3D.** Cerere de brevet de invenție Numărul A/00395 din 17.06.2026.

EXPERIENȚĂ STIINȚIFICĂ

Director de grant, finanțat de guvernul României, printr-un proiect național în 2008, pentru proiectul intitulat **"**CONTRIBUTII LA DEZVOLTAREA UNOR CONVERTOARE ELECTRONICE CU PERFORMANȚE RIDICATE DE CALITATE A ENERGIEI ELECTRICE**"**, cu indicativ **TD-489**.