MINISTERUL EDUCAŢIEI ŞI CERCETĂRII
UNIVERSITATEA NAŢIONALĂ DE ŞTIINŢĂ ŞI TEHNOLOGIE
POLITEHNICA BUCUREŞTI

Şcoala Doctorală de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

REZUMAT TEZĂ DE DOCTORAT

Ing. Cristian Manolache

TEHNICI DE OPTIMIZARE ȘI ÎNVĂȚARE AUTOMATĂ APLICATE ÎN VERIFICAREA CIRCUITELOR INTEGRATE ANALOGICE

OPTIMIZATION TECHNIQUES AND MACHINE LEARNING APPLIED IN ANALOGUE CIRCUIT VERIFICATION

Prof. Dr. Ing. Corneliu Burileanu
Univ. Politehnica din Bucureşti

Conducător de doctorat

BUCUREŞTI 2025

Cuprins

1 Introducere **1**

1.1 Circuite Analogice 1

1.2 Învățare Automată în Industria Semiconductorilor 1

1.3 Motivație 2

1.4 Obiective 2

1.5 Structura tezei 3

2 Metode Computaționale pentru Eșantionare și Optimizare **4**

2.1 Algoritmi de eșantionare 5

2.1.1 Grid-Based Stratification (GBS) 5

2.1.2 Căutare Aleatorie 5

2.1.3 Latin Hypercube Sampling (LHS) 5

2.1.4 Eșantionare cu Orthogonal Arrays 6

2.2 Algoritmi de optimizare 6

2.2.1 Gradient Descent 6

2.2.2 Algoritmi Evolutivi 6

2.3 Multi-objective Acquisition Ensemble 7

2.4 Procese Gaussienene 8

2.5 Prezentare generală a literaturii privind verificarea circuitelor 8

3 Circuite de test **10**

3.1 Prezentare Generală 10

3.2 Circuite Sintetice 10

3.3 Circuite Reale 10

3.3.1 Stabilizator de Tensiune 11

3.3.2 Amplificator Operațional cu compensație feed-forward (FFOA) 11

3.4 Benchmarking pentru circuite sintetice 11

3.4.1 Introducere 11

3.4.2 Crearea și Dezvoltarea Benchmark-ului Sintetic 11

4 Algoritmul de Verificare Adaptivă pentru Circuite Analogice (AVAC) **12**

4.1 Fixed Planning 12

4.2 Ajustări ale Algoritmului 13
4.3 Adaptive Planning 14
4.4 Selecția candidaților folosind Multi-objective Acquisition Ensemble (MACE) 15
4.5 Process Corners 16

5 Evaluarea Algoritmului și Rezultate Experimentale 18
5.1 Metricile de evaluare 18
5.2 Evaluarea versiunii de bază a algoritmului 18
5.3 Ajustări și studiu aprofundat al Hiper-parametrilor 19
5.4 Analiza Adaptive Planning 19
5.5 Selecția candidaților folosind MACE 19
5.5.1 Comparație între abordările Gradient Descent și MACE pe circuitul SC1-PC 19
5.5.2 Comparație între abordările Gradient Descent și MACE pe circuitele VR 20
5.5.3 Concluzii privind abordările MACE vs Gradient Descent 20
5.6 Validarea pe benchmark sintetic 20
5.6.1 Analiza pe seturile de evaluare de dimensiuni mici 20
5.6.2 Analiza pe seturile de evaluare de complexitate ridicată 20
5.7 Verificarea pe Amplificatorului Operațional cu compensație feed-forward (FFOA) 21
5.7.1 Analiza Inițială 21
5.7.2 Rezultate de verificare pentru 4 condiții de operare 21
5.7.3 Rezultate de verificare pentru 5 condiții de operare 21

6 Metrica de estimare a riscului 22
6.1 Probabilitatea de Eșec a Specificației (SFP) 22
6.2 Configurația experimentală 22
6.3 Rezultate Experimentale 23
6.3.1 Circuitul SC1 cu specificații care nu pot eșua 23
6.3.2 Circuitul SC1 cu specificații care pot eșua 24
6.3.3 Circuitul VR3 cu specificații realiste 24
6.4 Discuție 24

7 Analiza bugetului de simulare 25
7.1 Propunere de Adaptive Planning cu Alocare Inegală a Bugetului 25
7.2 Evoluția versiunilor propuse într-un scenariu simplu 25
7.3 Versiunea implementată a bugetului dinamic 4 26
7.4 Versiunea implementată 4: evaluare și actualizări 26
7.5 Versiunea implementată a bugetului dinamic 5 26

7.6 Concluzii privind alocarea dinamică a bugetului 26

8 Concluzii **27**

8.1 Concluzii Generale 27

8.2 Contribuții originale 27

8.3 Lista publicațiilor originale 28

8.4 Lista contractelor 29

8.5 Perspective pentru dezvoltări ulterioare 30

Bibliografie **31**

Capitolul 1

Introducere

Această teză abordează sarcina verificării circuitelor analogice într-un mediu Pre-Silicon cu ajutorul tehnicilor de învățare automată. Motivația cercetării în acest domeniu constă în complexitatea crescândă și cerințele stricte de fiabilitate ale circuitelor integrate analogice, în special în industria auto.

1.1 Circuite Analogice

Circuitele analogice sunt componente fundamentale în tehnologia modernă, esențiale pentru procesarea semnalelor continue într-o gamă variată de aplicații. Ele sunt integrate în sistemele auto pentru controlul motorului și elementele de siguranță, în echipamentele audio pentru amplificarea sunetului de înaltă calitate, în automatizările industriale pentru controlul de precizie și în sistemele aerospațiale pentru navigație și comunicații. Rolul lor în instrumentație, în special în dispozitivele medicale și industriale, este critic, pentru măsurători și diagnostice precise.

1.2 Învățare Automată în Industria Semiconductorilor

Industria semiconductorilor utilizează din ce în ce mai mult învățarea automată pentru a îmbunătăți procesele de fabricație, a crește randamentele de producție și a asigura calitatea produselor. Învățarea automată este aplicată în domenii critice precum proiectarea optimizată pentru fabricație, unde modelele pot genera automat scheme optime pentru circuitele analogice, înlocuind designul manual [12]. De asemenea, este utilizată pentru predicția randamentului, unde algoritmii prezic randamentele de producție și analizează impactul parametrilor de proces. În plus, tehnicile de învățare automată sunt folosite pentru detectarea defectelor și anomaliilor în producția de plăcuțe de cipuri, permițând identificarea din timp a abaterilor de proces pentru a reduce costurile și a îmbunătăți controlul calității.

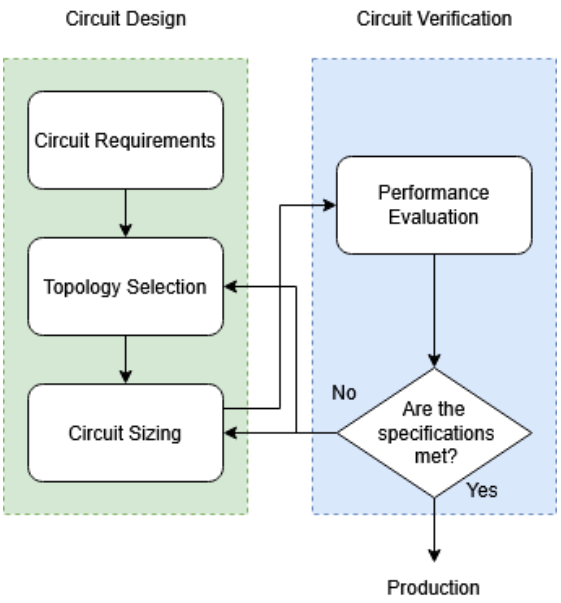


Figura 1.1 Fluxul de producție de bază al unui circuit analogic.

1.3 Motivație

Motivația principală a acestei lucrări provine din provocările din industria auto, unde adăugarea de noi funcționalități duce la o complexitate mai mare a designului și necesită procese riguroase de verificare pentru a asigura siguranța și fiabilitatea. Verificarea circuitelor analogice este un pas important în dezvoltarea circuitelor, asigurând că design-urile funcționează corect conform specificațiilor lor. Figura 1.1 evidențiază cele două etape principale de dezvoltare ale unui circuit analogic. Metodele tradiționale de verificare se bazează pe instrumente CAD și pe intervenția umană, implicând adesea explorări laborioase bazate pe grile, ale spațiului de intrare al condițiilor de operare. Deși inteligența artificială a fost aplicată pentru a ajuta acest proces, lipsește o soluție integrată și automatizată care poate gestiona eficient un număr mare de specificații și condiții de operare. Această teză își propune să adapteze și să aplice metode de inteligență artificială pentru a crea un instrument automatizat care să ajute inginerii de verificare să găsească scenarii nefavorabile mai rapid și mai eficient, un avantaj semnificativ într-o industrie cu timpi de dezvoltare din ce în ce mai mici.

1.4 Obiective

Cercetarea prezentată în această teză are cinci obiective principale. Primul obiectiv este cercetarea, proiectarea și implementarea tehnicilor de învățare automată pentru verificarea circuitelor analogice, culminând cu dezvoltarea algoritmului de Verificare Adaptivă pentru Circuite Analogice (Adaptive Verification for Analog Circuits - AVAC). Acest algoritm este destinat să servească drept alternativă la verificarea manuală, obținând

rezultate comparabile sau mai bune cu mai puține simulări, începând cu o acoperire inițială a spațiului de intrare. După simularea circuitului în aceste puncte inițiale, noile rezultate sunt utilizate pentru a continua căutarea colțurilor nefavorabile. Al doilea obiectiv este de a efectua o analiză aprofundată a algoritmului AVAC, evaluând performanța sa de bază și explorând posibile îmbunătățiri prin ajustarea hiper-parametrilor și alte modificări. Al treilea obiectiv implică evaluarea și validarea amănunțită a algoritmului AVAC prin crearea unui benchmark sintetic ce cuprinde aproximativ 1000 de funcții care emulează răspunsuri reale ale circuitelor de diverse complexități. Al patrulea obiectiv este cercetarea, proiectarea și implementarea unei metrici de estimare a riscului de eșec, numită Probabilitatea de Eșec a Specificației (Specification Failure Probability - SFP), pentru a oferi inginerilor o avertizare din timp a probabilității de eșec a specificației, economisind astfel timp prețios. Al cincilea și ultimul obiectiv este de a crea o strategie de alocare a bugetului de simulare pentru a optimiza eficiența algoritmului AVAC, explorând cum să distribuie cel mai bine un număr limitat de simulări între diferitele răspunsuri ale unui anumit circuit pentru a identifica mai rapid potențialele probleme.

1.5 Structura tezei

Această teza este organizată în opt capitole. Capitolul 2 introduce noțiunile teoretice ale diferitelor metode de eșantionare și algoritmi de optimizare relevanți pentru problema verificării circuitelor, alături de o revizuire a literaturii existente. Capitolul 3 prezintă circuitele de test utilizate pentru evaluarea algoritmului AVAC, incluzând atât circuite reale, cât și sintetice, precum și un benchmark sintetic dezvoltat special pentru testarea exhaustivă. Capitolul 4 detaliază dezvoltarea iterativă a algoritmului AVAC, de la versiunea sa inițială la actualizări și îmbunătățiri ulterioare. Capitolul 5 prezintă rezultatele experimentale obținute prin aplicarea diferitelor versiuni ale algoritmului AVAC la circuitele descrise în Capitolul 3. Capitolele 6 și 7 introduc extensii semnificative la algoritmul AVAC: Capitolul 6 prezintă metrica de estimare a riscului SFP, iar Capitolul 7 explorează diferite strategii de alocare dinamică a bugetului de simulare. În cele din urmă, Capitolul 8 prezintă concluziile generale ale tezei, subliniază contribuțiile originale, listează publicațiile legate de teză și discută perspectivele pentru potențiale lucrări viitoare.

Capitolul 2

Metode Computaționale pentru Eșantionare și Optimizare

Verificarea circuitelor vizează să asigure funcționarea corectă a unui circuit conform specificațiilor sale în toate condițiile de intrare posibile, care sunt definite de variabile de variație precum Proces-Tensiune-Temperatură (PVT). Metodele tradiționale de verificare, cum ar fi executarea unui full-factorial al tuturor combinațiilor, sunt amănunțite, dar adesea nefezabile din cauza creșterii exponențiale a simulărilor necesare cu nodurile de proces moderne. Alte abordări, cum ar fi "cea mai bună estimare" a proiectantului, sunt rapide, dar nesigure, în timp ce analiza de sensibilitate și modelele pătratice [21] pot eșua să surprindă comportamentul neliniar al circuitelor. Acest lucru a condus la adoptarea unor modele de regresie mai complexe, cum ar fi Procesele Gaussienene [3], care pot modela mapări arbitrare neliniare de la variabilele PVT la ieșirile circuitului. Fluxul tradițional de proiectare și verificare, prezentat în Figura 2.1, este un proces iterativ și consumator de timp, durând adesea luni chiar și pentru circuite simple. Această teză se concentrează pe accelerarea fazei de verificare, care are loc după ce parametrii de proiectare sunt fixați, prin utilizarea învățării automate pentru a căuta eficient în spațiul condițiilor de operare (intrările circuitului) configurații nefavorabile care ar putea duce la eșecuri ale specificațiilor.

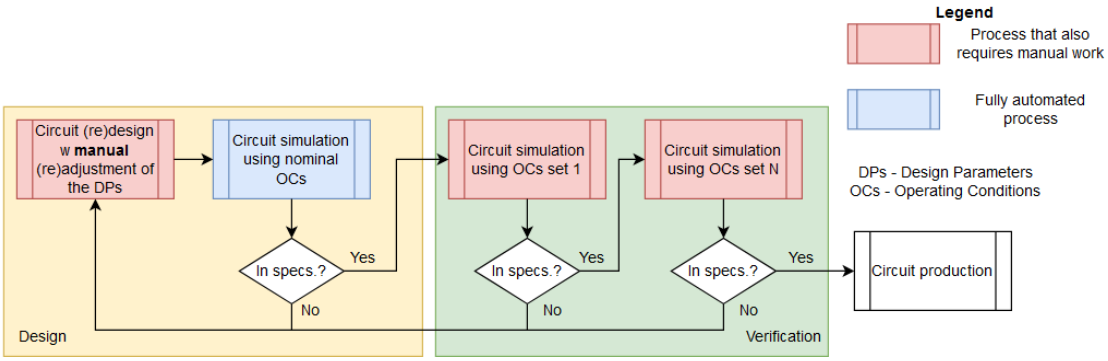


Figura 2.1 Fluxul tradițional de proiectare și verificare pentru circuite analogice/mixte.

2.1 Algoritmi de eșantionare

Pentru a începe procesul de verificare, este necesar un set inițial de eșantioane din spațiul de intrare al condițiilor de operare pentru a obține o perspectivă asupra comportamentului circuitului. Planificarea acestor intrări inițiale este un pas important pentru construirea unui model surogat eficient [24]. Design-uri experimentale ce acoperă spațiul de condiții și care urmăresc să distribuie punctele de eșantionare uniform pe domeniu, sunt adesea preferate atunci când nu se știu multe despre relația intrare-ieșire.

2.1.1 Grid-Based Stratification (GBS)

În GBS, fiecare dimensiune a spațiului de intrare este divizat în părți egale, creând o grilă de sub-cuburi din care se eșantionează un singur punct în mod uniform. Această metodă oferă o reducere semnificativă a variației în comparație cu eșantionarea Monte Carlo standard, în special în cazuri cu dimensiuni reduse. Cu toate acestea, utilitatea sa practică este limitată în spații cu dimensiuni mari, deoarece numărul de eșantioane necesare crește exponențial (2^d, unde d este numărul de dimensiuni ale spațiului din care eșantionăm), și este dificil de extins un set existent de eșantioane GBS [23].

2.1.2 Căutare Aleatorie

Căutarea aleatorie este o variație a GBS, care eșantionează spațiul de căutare aleatoriu în loc să utilizeze o grilă discretă. Deși și această metodă suferă de "curse of dimensionality", este adesea o alegere practică mai bună decât căutarea pe grilă, deoarece explorează valori distincte pentru toți parametrii în fiecare încercare (tentativă cu un set unic de hiper-parametri), fiind mai eficientă în optimizarea funcțiilor cu dimensionalitate efectivă redusă [2]. Simplitatea sa și faptul că poate fi executată în paralel o fac o primă opțiune atractivă, însă nu oferă nici-o garanție de a găsi un minim local fără o eșantionare amănunțită.

2.1.3 Latin Hypercube Sampling (LHS)

LHS este o metodă de stratificare în care fiecare dintre cele d dimensiuni este împărțită în n straturi egale, și n puncte sunt eșantionate astfel încât fiecare strat din fiecare dimensiune să conțină exact un punct. Această metodă este deosebit de eficientă atunci când funcția de modelat este aproape aditivă [23]. LHS poate reduce considerabil varianța în comparație cu eșantionarea Monte Carlo simplă, fără riscul de a o crește semnificativ, fiind un candidat potrivit pentru eșantionarea în spații de dimensiuni mari.

2.1.4 Eșantionare cu Orthogonal Arrays

Eșantionarea cu Orthogonal Arrays este o metodă combinatorie care generalizează LHS prin stratificarea tuturor perechilor (sau t-tuplurilor) de componente. Un Orthogonal Array de putere t asigură că, în orice sub-matrice de t coloane, toate combinațiile posibile de nivele apar de un număr egal de ori. Acest lucru permite analiza interacțiunilor dintre factori, nu doar a efectelor lor individuale. Orthogonal Arrays sunt deosebit de utile pentru detectarea comportamentului în zonele extreme ale domeniului și s-au dovedit a funcționa bine în spații de dimensiuni medii și mari [23].

2.2 Algoritmi de optimizare

După o eșantionare inițială a hiperspațiului condițiilor de operare, este necesară o metodă de optimizare a funcției pentru a ghida căutarea ulterioară a scenariilor nefavorabile pe baza datelor inițiale. Deoarece răspunsurile circuitelor pot fi modelate ca funcții ale condițiilor de operare de intrare, aceste metode urmăresc să găsească minimul sau maximul unei funcții obiectiv.

2.2.1 Gradient Descent

Gradient Descent este un algoritm de optimizare utilizat pe scară largă în învățarea automată pentru a minimiza funcțiile obiectiv prin ajustarea iterativă a parametrilor în direcția celei mai abrupte pante a gradientului funcției. Există trei tipuri principale: Batch, Stochastic și Mini-Batch Gradient Descent, care diferă în ceea ce privește cantitatea de date utilizate pentru a calcula gradientul pentru fiecare actualizare de parametru. Variante precum Momentum, RMSProp și Adam au fost dezvoltate pentru a-i spori eficacitatea. Deși simplu și eficient, Gradient Descent este sensibil la rata de învățare și se poate bloca în minime locale, dar variantele sale moderne au abordat multe dintre aceste limitări, devenind un instrument puternic pentru problema de optimizare.

2.2.2 Algoritmi Evolutivi

Algoritmii evolutivi sunt metode de optimizare inspirate de selecția naturală, ce operează pe o populație de soluții potențiale care sunt îmbunătățite progresiv prin procese precum reproducerea, mutația și selecția. Aceștia sunt deosebit de utili pentru probleme complexe de optimizare. Operatorii comuni includ "simulated binary crossover" și "polynomial mutation". Performanța acestor algoritmi este influențată de mai mulți parametri cheie, cum ar fi "crossover and mutation distribution indices, mutation probability, tournament size, and population size", care controlează echilibrul dintre explorare și exploatare.

Non-Dominated Sorting Genetic Algorithm-II (NSGAII) [4] este un algoritm evolutiv unde fiecărei soluții individuale i se atribuie un rang bazat pe frontul Pareto căruia

îi aparține, cele mai bune soluții având rangul 0 (frontul Pareto). Soluțiile care sunt dominate doar de indivizi de pe frontul Pareto primesc rangul 1, și așa mai departe. Procesul de selecție pentru următoarea generație implică colectarea tuturor candidaților concurenți și clasarea acestora după dominanța Pareto. Dacă numărul soluțiilor nedominante depășește dimensiunea populației, acestea sunt clasate în continuare după distanță și reduse; în caz contrar, locurile disponibile sunt umplute cu soluții dominate pe baza calității lor.

Generalized Differential Evolution 3 (GDE3) [16] este o versiune extinsă a evoluției diferențiale de bază, modificând regula sa de selecție și aplicând-o problemelor cu funcții obiectiv și de constrângere. În acest algoritm, noi soluții candidate sunt create prin perturbarea membrilor existenți ai populației, de obicei folosind diferența ponderată dintre doi indivizi selectați aleatoriu, adăugată unui al treilea. Principiul de selecție dictează că un vector de încercare înlocuiește un vector vechi dacă are o valoare obiectiv egală sau mai mică; dacă soluțiile fezabile nu se domină reciproc, atât vectorii descendenți, cât și cei părinți sunt salvați pentru următoarea generație. Acest mecanism ajută la reducerea costurilor computaționale și la îmbunătățirea eficienței, și, deși dimensiunea populației poate crește, ea este apoi redusă la dimensiunea sa originală prin eliminarea celor mai slabi membri pe baza non-dominanței și aglomerației. GDE3 a demonstrat o diversitate îmbunătățită a soluțiilor în comparație cu NSGAII și proprietăți de convergență mai bune în anumite probleme de test.

2.3 Multi-objective Acquisition Ensemble

O metodă avansată de selecție a candidaților a fost preluată de la provocarea de optimizare Black-Box NeurIPS 2020, unde metodele cu cele mai bune performanțe au utilizat abordări noi pentru optimizarea Bayesiană. O metodă deosebit de influentă a fost algoritmul Multi-objective Acquisition Ensemble (MACE) [19]. Algoritmul MACE tratează optimizarea ca o problemă multi-obiectiv, unde trei funcții de achiziție sunt combinate pentru a identifica un front Pareto. Algoritmul începe prin crearea unui model de Proces Gaussian dintr-un set inițial de puncte și apoi construiește cele trei funcții de achiziție: Lower Confidence Bound (LCB), îmbunătățirea așteptată (expected improvement) și probabilitatea de îmbunătățire (probability of improvement). Un algoritm evolutiv (DEMO [25]) este utilizat pentru a găsi frontul Pareto pe baza acestor trei funcții, reprezentând compromisurile optime între explorare și exploatare. Un lot de puncte este apoi selectat din acest front pentru evaluare, iar procesul este iterat. Această abordare a demonstrat eficacitate atât pe benchmark-uri sintetice, cât și pe aplicații din lumea reală, dovedindu-se robustă și eficientă în navigarea spațiilor obiectiv de dimensiuni mari și zgomotoase. Algoritmul MACE este prezentat în Algoritmul 1.

Algorithm 1: Algoritmul Multi-objective Acquisition Ensemble, unde EI: Îmbunătățire Așteptată, PI: Probabilitatea de Îmbunătățire și GP: Proces Gaussian.

Input: Număr de puncte de eșantionare inițiale N_{init}, număr de iterații N_{iter}, dimensiune lot B

Output: cel mai bun $f(x)$ înregistrat în timpul iterațiilor

Se eșantionează aleatoriu N_{init} puncte în spațiul de proiectare
Se creează modelul GP inițial
for $t = 1, 2, \ldots, N_{iter}$ **do**
    Se creează funcțiile LCB, EI și PI
    Se caută frontul Pareto al funcțiilor LCB, EI, PI folosind algoritmul DEMO
    Se eșantionează aleatoriu B puncte $x_1, \ldots, x_B$ din punctele optime Pareto
    Se evaluează $x_1, \ldots, x_B$ pentru a obține $y_1 = f(x_1), \ldots, y_B = f(x_B)$
    Se actualizează modelul GP
end for

2.4 Procese Gaussienene

Pentru a ghida algoritmii de optimizare, este necesar un model surogat care să ofere o estimare a funcțiilor de răspuns ale circuitului. Procesele Gaussienene au fost alese pentru acest rol datorită rădăcinilor lor în inferența Bayesiană, ceea ce le permite să ofere nu numai predicții, ci și o estimare fiabilă a propriei lor incertitudini [15]. Acesta este un beneficiu cheie, deoarece incertitudinea unui Proces Gausian ajustat crește în regiunile îndepărtate de datele de antrenament. Cu toate acestea, Procesele Gaussienene au dezavantaje, inclusiv costul computațional ridicat în spații de dimensiuni mari și dificultatea în gestionarea variabilelor de intrare categorice sau întregi. Abordările naive pentru gestionarea unor astfel de variabile, cum ar fi rotunjirea, pot duce la rezultate suboptimale. Metode mai avansate modifică kernelul Procesului Gaussian pentru a asigura reflecția corectă a naturii discrete a intrărilor de către model [8].

2.5 Prezentare generală a literaturii privind verificarea circuitelor

Sarcina verificării circuitelor este importantă, în special în domenii critice pentru siguranță, și poate consuma până la 70% din timpul total de proiectare [5] [7]. Procesul de verificare a circuitelor este împărțit în 2 sub-etape, anume, etapa de verificare Pre-Silicon și etapa de verificare Post-Silicon, cu accentul acestei teze pe prima. În timp ce verificarea este aplicată atât circuitelor digitale [14] [9], cât și circuitelor analogice, acestea din urmă sunt mai complexe [10] datorită valorilor lor de ieșire continue. Abordările avansate pentru verificarea analogică utilizează adesea tehnici de învățare automată [13] [6] [32] [26]. De exemplu, unele metode folosesc "sequential pattern mining" pentru

a maximiza acoperirea testelor [18], în timp ce altele propun framework-uri Hybrid Formal/Machine-learning Verification (HFMV) [13] care utilizează modele probabilistice precum Procesele Gaussienene pentru a găsi eșecuri mai eficient decât metodele Monte Carlo. Domeniul adiacent al dimensionării circuitelor beneficiază, de asemenea, de tehnici similare, unde algoritmi evolutivi și optimizare Bayesiană sunt utilizați pentru a găsi parametri optimi de proiectare cu un buget limitat de simulări [27] [29].

Capitolul 3

Circuite de test

3.1 Prezentare Generală

În verificarea Pre-Silicon, un circuit este definit de trei caracteristici principale: parametrii de proiectare, condițiile de operare și răspunsurile. Parametrii de proiectare au valori fixe, stabilite în faza de proiectare. Condițiile de operare sunt variabile de intrare (e.g., temperatură, Process Corners) care sunt variate pentru a testa robustețea circuitului. Răspunsurile reprezintă ieșirile ce trebuie să se încadreze în anumite praguri de specificație. Performanța algoritmului este evaluată pe circuite sintetice, cu răspunsuri definite matematic, și pe circuite reale, simulate în instrumente CAD.

3.2 Circuite Sintetice

Circuitele sintetice sunt avantajoase pentru evaluarea algoritmului, deoarece răspunsurile lor sunt funcții cunoscute ale condițiilor de operare de intrare, permițând calcularea precisă a valorilor minime și maxime reale. Acest lucru permite o evaluare exactă a cât de aproape ajunge algoritmul de valorile adevărate minime sau maxime ale răspunsurilor. Un circuit sintetic cheie utilizat în experimente este Circuitul Sintetic 1 (SC1), care are șase condiții de operare și 25 de răspunsuri definite. Aceste răspunsuri sunt combinații liniare de funcții ce depind de o singură condiție de operare. O variantă, SC1-PC, include un Process Corner suplimentar pentru a emula variațiile de fabricație.

3.3 Circuite Reale

Pentru a valida aplicabilitatea algoritmului în scenarii din lumea reală, au fost utilizate două tipuri de circuite reale. Aceste circuite "reale" sunt modele obținute după faza de proiectare și sunt simulate în instrumente CAD. Spre deosebire de circuitele sintetice, valorile minime și maxime reale ale răspunsurilor pentru aceste circuite reale sunt în general necunoscute.

3.3.1 Stabilizator de Tensiune

Primul circuit real este un stabilizator de tensiune (VR), pentru care sunt utilizate trei variante diferite (VR1, VR2, VR3). Fiecare circuit VR are 27 de parametri de proiectare, șapte condiții de operare, un Process Corner și trei răspunsuri cu limite de specificație care trebuie respectate: Gain Margin > 8 dB, Phase Margin > 20 deg. și Power Supply Rejection Ratio (PSRR) < –26 dB.

3.3.2 Amplificator Operațional cu compensație feed-forward (FFOA)

Al doilea circuit real constă într-un amplificator operațional cu compensație feed-forward (FFOA) și care este un design open-source [28]. Acest FFOA are 11 parametri de proiectare ficși, un număr variabil de condiții de operare (între două și cinci), un Process Corner și șase răspunsuri, incluzând Current Consumption (IDD), Unity-gain Bandwidth (f0dB), DC gain (LG0), Phase Margin (PM), Gain Margin (GM), and Slew Rate (SR).

3.4 Benchmarking pentru circuite sintetice

3.4.1 Introducere

Deși se găsesc benchmark-uri sintetice pentru circuite digitale în literatură, acestea sunt rare pentru verificarea circuitelor analogice Pre-Silicon. Pentru a aborda acestă problemă, un benchmark cuprinzător a fost proiectat și implementat în colaborare cu ingineri din industrie pentru a testa algoritmul AVAC.

3.4.2 Crearea și Dezvoltarea Benchmark-ului Sintetic

O abordare bazată pe date a fost utilizată pentru a crea un benchmark de 900 de circuite sintetice, cu 100 de circuite pentru fiecare număr de condiții de operare de intrare, variind între două și zece. Benchmark-ul a fost construit dintr-o listă de bază de 30 de funcții unidimensionale proiectate de inginerii de circuite ce emulează comportamente plauzibile ale circuitelor reale. Complexitatea acestor funcții de bază a fost evaluată printr-un proces de clasificare subiectivă de către un grup de șase persoane. Circuitele sintetice finale (un singur răspuns) au fost create prin adunarea acestor funcții unidimensionale. Pentru a compara dificultatea circuitelor cu numere diferite de intrări, a fost definită o metrică numită Complexitate Multiplicativă (MCpx) ca produsul complexităților funcțiilor componente. Benchmark-ul a fost conceput pentru a se concentra pe circuite cu un MCpx maxim de 40 și până la 10 condiții de operare, reflectând scenarii realiste.

Capitolul 4

Algoritmul de Verificare Adaptivă pentru Circuite Analogice (AVAC)

Acest capitol detaliază algoritmul de Verificare Adaptivă pentru Circuite Analogice (AVAC), al cărui obiectiv principal este verificarea faptului că răspunsurile unui circuit rămân în limitele specificațiilor pentru toate condițiile de operare. Algoritmul tratează circuitele ca pe "black boxes" și urmează un proces în trei etape: (1) Fixed Planning pentru achiziția inițială de date, (2) antrenarea modelelor surogat și (3) aplicarea unui algoritm de propunere a candidaților pentru a găsi potențiale eșecuri ale specificațiilor. Pașii principali ai metodei de verificare sunt, de asemenea, evidențiați în Figura 4.1.

4.1 Fixed Planning

Etapa de Fixed Planning reprezintă faza inițială a algoritmului AVAC, concentrându-se pe generarea unui set inițial de date pentru antrenarea modelelor surogat. În mod tradițional, inginerii de verificare se bazau pe o abordare full-factorial, simulând toate combinațiile de valori minime, maxime și nominale pentru condițiile de operare. Atunci

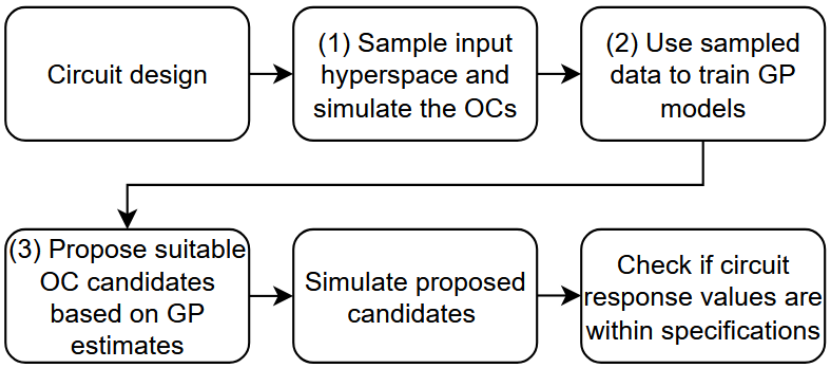


Figura 4.1 Pașii principali ai algoritmului de bază AVAC. Unde OC: Condiție de Operare; GP: Proces Gaussian.

când numărul condițiilor de operare este mare, această metodă nu este fezabilă, deoarece numărul de simulări crește exponențial. Pentru a depăși acest inconvenient, sunt utilizate metode alternative de eșantionare, incluzând Monte Carlo, eșantionarea cu Orthogonal Arrays [20] și Latin Hypercube Sampling (LHS) [22]. Eșantionarea cu Orthogonal Arrays este deosebit de eficientă pentru explorarea colțurilor hiperspațiului condițiilor de operare cu puține simulări, în timp ce LHS oferă o acoperire generală mai bună prin stratificarea fiecărei dimensiuni. O altă abordare considerată, constă în combinarea Orthogonal Arrays și LHS pentru a valorifica punctele forte respective, vizând atât explorarea colțurilor, cât și o acoperire cuprinzătoare a hiperspațiului.

Datele colectate în timpul etapei de Fixed Planning sunt apoi utilizate pentru a antrena modelele surogat de Proces Gaussian pentru fiecare răspuns al circuitului. Procesele Gaussienene sunt preferate datorită capacității lor de a modela funcțiile probabilistic și de a oferi o estimare fiabilă a propriei lor incertitudini, care crește odată cu distanța față de datele de antrenare. Pentru a ghida căutarea potențialelor probleme ale circuitului, este utilizată o funcție de achiziție numită Lower Confidence Bound (LCB). Funcția LCB estimează anvelopa inferioară a incertitudinii Procesului Gaussian. Parametrul beta pentru LCB permite ajustarea echilibrului între exploatarea regiunilor cunoscute cu valori scăzute (beta mai mic) și explorarea zonelor incerte (beta mai mare). Pentru implementarea acestor modele de Proces Gaussian și funcții de achiziție, s-a utilizat Framework-ul BoTorch [1].

După antrenarea modelelor de Proces Gaussian, un algoritm de propunere a candidaților este aplicat pentru a sugera Configurații de Condiții de Operare (OCC) care sunt susceptibile să ducă la eșecuri ale specificațiilor circuitului. Acest proces începe prin generarea unui set mare de evaluare de OCC-uri, de obicei o combinație de full-factorial și eșantioane suplimentare LHS. Procesele Gaussienene antrenate prezic apoi răspunsurile circuitului și incertitudinile asociate pentru aceste OCC-uri. Acești candidați sunt clasificați folosind scorul LCB, iar OCC-ul cu cel mai mic scor LCB (pentru probleme de minimizare) este selectat ca următorul candidat pentru simularea circuitului.

4.2 Ajustări ale Algoritmului

Au fost considerate mai multe ajustări ale algoritmului de bază pentru a-i îmbunătăți performanța. O actualizare cheie a fost aplicarea de Gradient Descent pe setul inițial de candidați pentru evaluare. Această metodă, utilizată de obicei pentru minimizarea funcției, folosește estimările Procesului Gaussian pentru a ghida căutarea către condițiile de operare ce generează valori ale răspunsurilor mai aproape de minimul funcției, rafinând astfel grupul de candidați potențiali. Alte actualizări au inclus un studiu aprofundat asupra hiper-parametrilor critici din algoritmul actualizat de propunere a candidaților. Această investigație a cuprins parametrul beta din funcția de achiziție LCB, rata de învățare, alegerea optimizatorului (Adam sau Stochastic Gradient Descent) și numărul

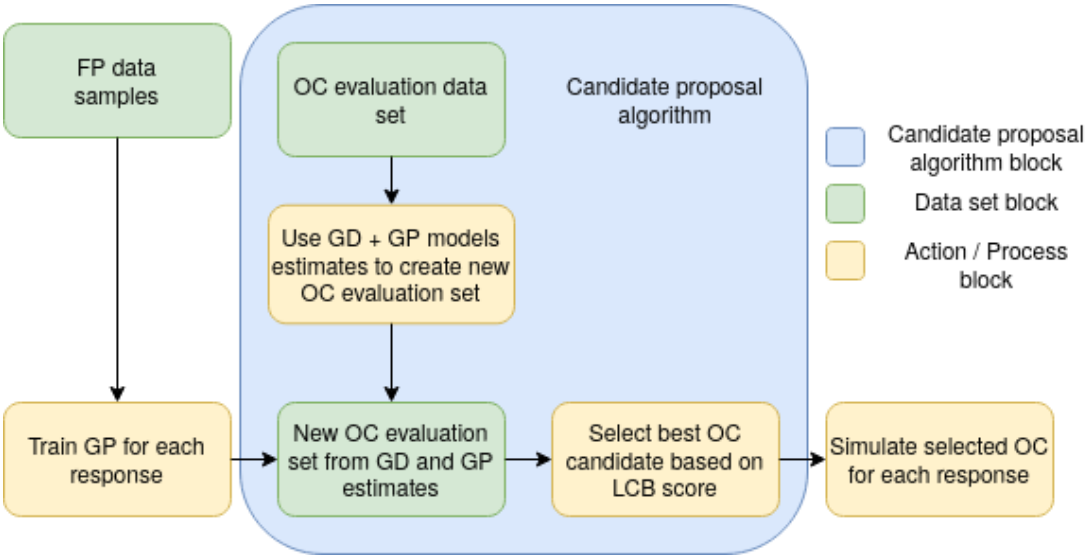


Figura 4.3 Diagrama algoritmului de selecție a candidaților, unde FP: Fixed Planning, OC: Condiție de Operare, GD: Gradient Descent, GP: Proces Gaussian.

de pași pentru Gradient Descent. În timp ce Adam este recunoscut pentru viteza sa computațională, SGD este notabil pentru capacitățile sale superioare de generalizare [11] [31]. În plus, au fost luate în considerare și alte funcții de achiziție pe lângă LCB, pentru clasarea condițiilor de operare, în special probabilitatea de îmbunătățire (PI) și îmbunătățirea așteptată (EI). Probabilitatea de îmbunătățire reprezintă probabilitatea de a obține o valoare mai bună decât cea mai bună valoare observată, în timp ce îmbunătățirea așteptată măsoară magnitudinea așteptată a unei astfel de îmbunătățiri. O prezentare generală a algoritmului complet, cu metoda actualizată de selecție a candidaților, este prezentată în Figura 4.3.

4.3 Adaptive Planning

Această secțiune introduce o nouă etapă, numită Adaptive Planning (Figura 4.4), ca o fază iterativă concepută pentru a depăși o limitare a versiunilor anterioare ale algoritmului, unde modelele de Proces Gaussian efectuau o singură selecție OCC per răspuns. Inițial, modelele GP au fost antrenate folosind date de Fixed Planning, derivate dintr-o combinație de eșantionare cu Orthogonal Array și LHS. Ulterior, Gradient Descent a fost aplicat unui set de evaluare de OCC-uri candidate pentru a le identifica pe cele mai susceptibile de a provoca probleme circuitului.

Mecanismul de bază al Adaptive Planning implică reantrenarea iterativă a Proceselor Gaussienene pe parcursul mai multor cicluri. La fiecare iterație, modelele sunt actualizate prin încorporarea atât a eșantioanelor inițiale din Fixed Planning, cât și a tuturor noilor OCC-uri candidate simulate în iterațiile anterioare. Această rafinare continuă permite Proceselor Gaussienene să-și îmbunătățească progresiv precizia predictivă pentru răspunsurile circuitului, ducând la propunerea de candidați mai buni care pot conduce

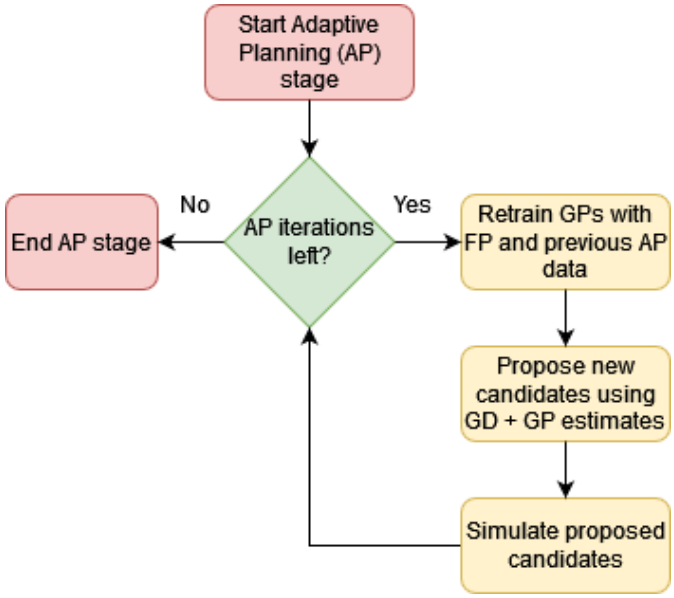


Figura 4.4 Diagrama etapei de Adaptive Planning. Numărul de iterații de Adaptive Planning este ales de expertul uman care conduce procesul de verificare. FP: Fixed Planning, GD: Gradient Descent, GP: Proces Gaussian.

la scenarii nefavorabile. Numărul de iterații de Adaptive Planning este un parametru configurabil, determinat de un expert uman care supraveghează procesul de verificare.

4.4 Selecția candidaților folosind Multi-objective Acquisition Ensemble (MACE)

Procesul de selecție al candidaților pentru algoritmul de bază, care utiliza Gradient Descent și o singură funcție de achiziție (cum ar fi LCB, probabilitatea de îmbunătățire sau îmbunătățirea așteptată), a prezentat mai multe dezavantaje, inclusiv potențiala convergență la minime locale și o lipsă de robustețe în diferite scenarii. Pentru a rezolva aceste limitări, a fost introdusă o versiune modificată a algoritmului Multi-objective Acquisition Ensemble (MACE) [19] [30]. Această abordare tratează selecția candidaților ca o problemă de optimizare multi-obiectiv. Aceasta vizează combinarea punctelor forte ale mai multor funcții de achiziție pentru a realiza o căutare mai robustă și eficientă a defecțiunilor circuitului.

Varianta MACE a algoritmului începe prin crearea unei populații inițiale de OCC-uri utilizând o combinație de eșantionare tip grilă și LHS. Acești candidați, împreună cu punctele simulate anterior, sunt apoi evaluați utilizând trei funcții de achiziție distincte: LCB, îmbunătățirea așteptată și probabilitatea de îmbunătățire. Algoritmul identifică frontul Pareto din aceste evaluări multi-obiectiv folosind algoritmul GDE3 [17], echilibrând explorarea regiunilor incerte cu exploatarea celor promițătoare. Un grup de puncte este apoi selectat din acest set optim Pareto pentru simularea circuitului, actualizând

iterativ modelele de Proces Gaussian pentru a rafina căutarea. Punctele principale ale variantei MACE alternative utilizate în algoritmul AVAC sunt prezentate în Algoritmul 2, în timp ce o diagramă a algoritmului complet se găsește în Figura 4.5.

Algorithm 2: Varianta MACE alternativă, unde OC: Condiție de Operare și GP: Proces Gaussian

Input: Populație inițială de dimensiune N, populație redusă de dimensiune P, număr de iterații GDE3 I

Output: Cele mai bune B OC candidate

1 Se creează o populație inițială de candidați OC. Se pornește de la o selecție tip grilă de nivel L per dimensiune de intrare, iar punctele rămase sunt completate până la N folosind LHS.
2 Se adaugă puncte simulate din iterațiile anterioare de Adaptive Planning.
3 Se evaluează populația inițială folosind GP și toate cele trei funcții de achiziție (îmbunătățire așteptată, probabilitatea de îmbunătățire, LCB).
4 Se reduce populația de pornire la cele mai bune P soluții folosind trierea non-dominant.
5 Se utilizează GDE3 pentru a evolua cele P soluții pentru I iterații/epoci.
6 Cele I populații din fiecare epocă sunt combinate și filtrate pentru duplicate, rezultând un set final de candidați OC.
7 Sunt selectate cele mai bune B soluții din setul final de candidați OC.

4.5 Process Corners

Algoritmul a fost extins pentru a gestiona Process Corners, care sunt variabile categorice ce reprezintă variații de fabricație ce pot afecta semnificativ performanța circuitului. Pentru encodarea lor, a fost dezvoltată o schemă eficientă ce reprezintă colțurile într-un spațiu întreg multi-dimensional. Această abordare profită de ordinea inerentă dintre valorile colțurilor (e.g., slow, typical, fast) și permite modelului de Proces Gaussian să învețe relațiile dintre ele mai eficient. Au fost, de asemenea, implementate mecanisme pentru a gestiona natura acestei grilei de colțuri valide, cum ar fi atribuirea de scoruri slabe combinațiilor invalide generate în timpul procesului de optimizare evolutiv.

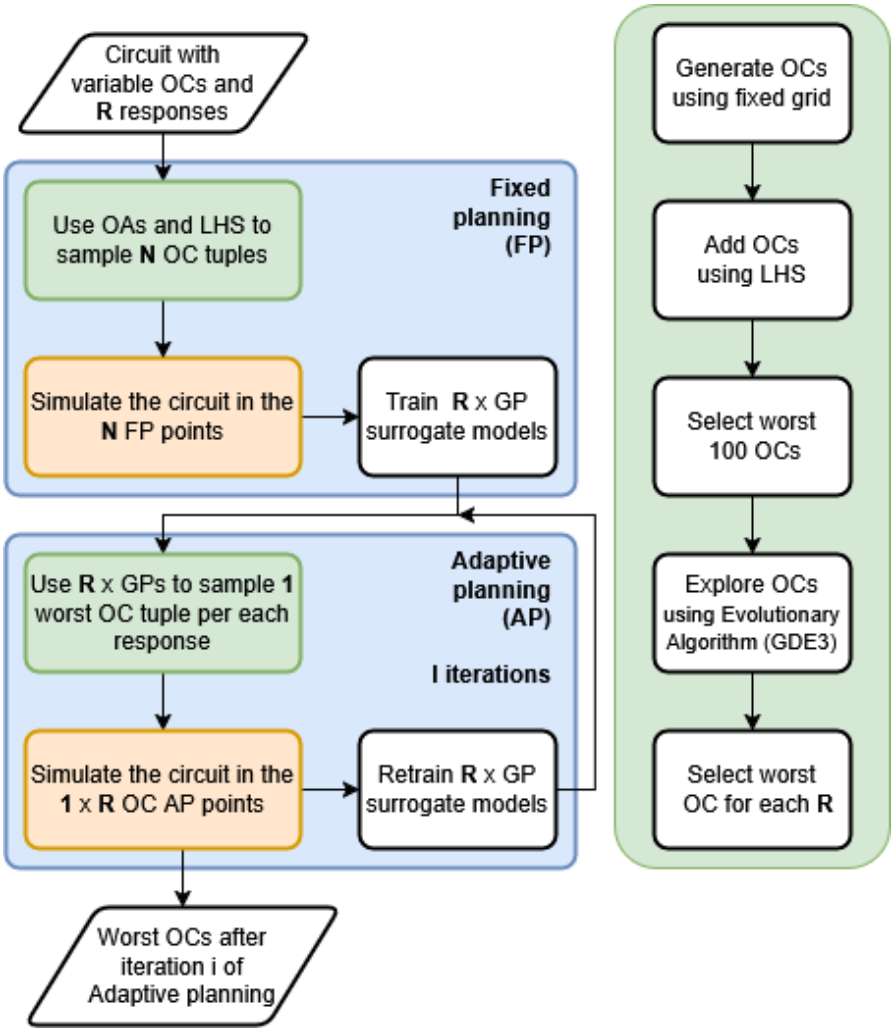


Figura 4.5 Diagrama algoritmului actualizat de selecție a candidaților, cu blocul MACE (verde) în dreapta. FP: Fixed Planning, OA: Orthogonal Array, OC: Condiție de Operare, GP: Proces Gaussian.

Capitolul 5

Evaluarea Algoritmului și Rezultate Experimentale

Acest capitol prezintă rezultatele experimentale și evaluarea algoritmului de Verificare Adaptivă pentru Circuite Analogice (AVAC) pe parcursul diferitelor sale etape de dezvoltare. Este prezentată performanța algoritmului atât pe circuite sintetice, cât și pe circuite reale (descrise în Capitolul 3), începând cu versiunea de bază și progresând prin actualizări care implică variantele Gradient Descent și Multi-objective Acquisition Ensemble (MACE). Capitolul introduce, de asemenea, metricile specifice de evaluare utilizate pentru a verifica eficacitatea algoritmului.

5.1 Metricile de evaluare

Performanța algoritmului a fost evaluată folosind diferite metrici pentru circuite sintetice și reale. Pentru circuitele sintetice, unde sunt cunoscute valorile minime și maxime reale ale răspunsului, metrica principală a fost Eroarea Relativă a Valorii (RVE), care măsoară diferența normalizată între valoarea găsită de algoritm și minimul real. Două metrici extinse, RVE Mediu (ARVE) și RVE Maxim (MRVE), au fost, de asemenea, utilizate pentru a evalua performanța pentru toate răspunsurile unui circuit. ARVE reprezintă RVE-ul mediu pentru toate răspunsurile unui circuit, în timp ce MRVE reprezintă RVE-ul maxim. Pentru circuitele reale, unde extremele reale sunt necunoscute, evaluarea s-a bazat pe valorile reale cele mai mici/mari găsite și pe dacă acestea au eșuat specificațiile impuse.

5.2 Evaluarea versiunii de bază a algoritmului

în experimentele inițiale pe circuitul sintetic SC1 s-au comparat diferite metode de eșantionare pentru etapa de Fixed Planning. S-a constatat că o combinație de eșantionare cu Orthogonal Arrays și LHS a oferit cea mai bună acoperire inițială și a depășit abordă-

rile full-factorial și Monte Carlo, folosind bugete de simulare reduse. Ulterior, antrenarea modelelor de Proces Gaussian pe aceste date și propunerea unui singur candidat nou pentru fiecare răspuns a dus la o îmbunătățire semnificativă, atingând un ARVE sub 1% folosind mult mai puține simulări decât abordarea clasică full-factorial.

5.3 Ajustări și studiu aprofundat al Hiper-parametrilor

Experimentele ulterioare axate pe ajustarea algoritmului au arătat că varierea hiper-parametrului beta pentru funcția de achiziție LCB a avut un impact relativ minor asupra performanței. Cea mai semnificativă îmbunătățire a venit din aplicarea Gradient Descent pe setul de evaluare a Configurațiilor de Condiții de Operare (OCC) pentru a genera candidați mai buni. Un studiu de aprofundat asupra optimizatorilor de Gradient Descent (Adam și SGD) și a diferitelor rate de învățare a arătat că anumite configurații specifice ar putea reduce în continuare ARVE, cele mai bune rezultate atingând 0,13% ARVE și 0,62% MRVE.

5.4 Analiza Adaptive Planning

Introducerea etapei complete de Adaptive Planning, cu multiple iterații de reantrenare, a fost evaluată pe circuitul sintetic SC1-PC și pe circuitul real VR2. Pe circuitul sintetic, execuția timp de 10 iterații a redus ARVE de la 0,75% după o iterație, la 0,35%. Pe circuitul VR2, procesul multi-iterație a fost crucial pentru găsirea unui eșec de specificație pentru răspunsul Gain Margin, un eșec care a fost ratat de versiunea de bază. Acest lucru a demonstrat că rafinarea iterativă a modelelor surogat cu date noi poate duce la descoperirea eșecurilor subtile în circuite complexe.

5.5 Selecția candidaților folosind MACE

Această secțiune prezintă rezultate experimentale care compară versiunea de bază (Gradient Descent) și cea actualizată (MACE) ale algoritmului AVAC, atât pe circuitul sintetic SC1-PC, cât și pe toate cele trei variante ale circuitelor reale VR. Toate experimentele au utilizat aceeași configurație, s-a variat doar abordarea de selecție a candidaților și numărul de iterații.

5.5.1 Comparație între abordările Gradient Descent și MACE pe circuitul SC1-PC

Pe circuitul sintetic SC1-PC, abordarea MACE a demonstrat o performanță puțin mai bună. După o iterație de Adaptive Planning, MACE a obținut un ARVE mai bun (0,38%

față de 0,71%) decât varianta Gradient Descent. Pe parcursul a 10 iterații, MACE a atins cel mai bun rezultat de 0,12% ARVE (față de 0,31%) mai rapid (la iterația 7) și a găsit minimul real pentru mai multe răspunsuri ale circuitului (18 din 25) în comparație cu abordarea Gradient Descent (12 din 25).

5.5.2 Comparație între abordările Gradient Descent și MACE pe circuitele VR

Pe cele trei variante ale circuitului real VR, rezultatele au fost mai nuanțate, ambele metode având rezultate similare pe mai multe răspunsuri. Cu toate acestea, o observație importantă a reieșit din testele pe circuitul VR1: după 10 iterații, abordarea MACE a identificat cu succes un OCC care a dus la un eșec de specificație pentru răspunsul Gain Margin, în timp ce abordarea Gradient Descent nu a găsit acest eșec critic.

5.5.3 Concluzii privind abordările MACE vs Gradient Descent

Deși algoritmul MACE este mai complex, natura sa multi-obiectiv s-a dovedit a fi ușor mai robustă, în special în capacitatea sa de a identifica eșecuri critice ale specificațiilor care au fost ratate de metoda Gradient Descent cu un singur obiectiv. Este posibil ca MACE să funcționeze mai bine în verificarea circuitelor mai complexe, deoarece circuitele testate au fost relativ simple, iar abordarea Gradient Descent a fost "suficient de bună".

5.6 Validarea pe benchmark sintetic

Această secțiune validează performanța algoritmului pe toate cele nouă seturi de evaluare ale benchmark-ului sintetic, care includ 900 de circuite sintetice cu complexități variate și 2 până la 10 condiții de operare de intrare. Toate experimentele au utilizat cea mai recentă versiune MACE a algoritmului de verificare, fiecare circuit fiind supus la 100 de simulări de Fixed Planning și 100 de Adaptive Planning.

5.6.1 Analiza pe seturile de evaluare de dimensiuni mici

Testele inițiale pe circuite cu două și trei condiții de operare au arătat că algoritmul a funcționat bine, numărul de iterații necesare pentru a găsi o valoare aproape de minimul real (RVE < 2%) corelând cu complexitatea subiectivă a funcțiilor.

5.6.2 Analiza pe seturile de evaluare de complexitate ridicată

În continuare, algoritmul a fost evaluat pe seturi mai complexe de circuite (4 până la 10 condiții de operare). Pentru majoritatea circuitelor (peste 70% pentru seturi de până la

șapte condiții de operare), a găsit cu succes o valoare aproape de minimul real în mai puțin de 10 iterații de Adaptive Planning. Pe măsură ce numărul condițiilor de operare și complexitatea circuitelor au crescut, au fost necesare mai multe iterații, ceea ce este de așteptat datorită dimensiunii mai mari a spațiului de căutare. Validarea efectuată pe acest benchmark divers a confirmat capacitatea algoritmului pentru sarcinile de verificare Pre-Silicon.

5.7 Verificarea pe Amplificatorului Operațional cu compensație feed-forward (FFOA)

Această secțiune detaliază procesul de verificare pe circuitul real FFOA, utilizând cea mai recentă versiune MACE a algoritmului pentru a găsi rezultate comparabile sau mai bune decât o abordare clasică (full-factorial), dar cu mai puține simulări. Scopul este de a identifica rezultate similare sau îmbunătățite în comparație cu verificarea clasică, utilizând un număr redus de simulări, variînd configurațiile pentru ambele abordări.

5.7.1 Analiza Inițială

O analiză inițială cu două condiții de operare și cinci Process Corners nu a găsit eșecuri de specificație, deși răspunsul Phase Margin a fost foarte aproape de limita sa. O căutare mai detaliată de tip grilă a confirmat că valoarea cea mai nefavorabilă era într-adevăr un colț și nu a eșuat specificația.

5.7.2 Rezultate de verificare pentru 4 condiții de operare

Verificarea a fost extinsă pentru a include patru condiții de operare, tratând doi parametri anterior fixați ca variabile. În acest scenariu, algoritmul AVAC a identificat cu succes un eșec de specificație pentru răspunsul Phase Margin. Algoritmul a găsit acest eșec utilizând 160 de simulări, mai puțin de 35% din cele 486 de simulări necesare pentru o abordare clasică full-factorial. De asemenea, a identificat un eșec la un OCC care nu era un colț, un punct pe care o verificare standard ce căuta în colțuri l-ar fi ratat.

5.7.3 Rezultate de verificare pentru 5 condiții de operare

În cele din urmă, a fost adăugată o a cincea condiție de operare, crescând și mai mult complexitatea problemei. Algoritmul a obținut rezultate similare cu căutarea exhaustivă full-factorial, găsind de această dată eșecuri atât pentru răspunsurile Phase Margin, cât și pentru Slew Rate. Acest lucru a fost realizat cu 160 de simulări, aproximativ 10% din cele 1458 de simulări necesare pentru baza de referință full-factorial, evidențiind eficiența algoritmului.

Capitolul 6

Metrica de estimare a riscului

Acest capitol introduce o nouă metrică, Probabilitatea de Eșec a Specificației (SFP), care indică probabilitatea unei defecțiuni a circuitului în timpul procesului de verificare Pre-Silicon. Metrica este concepută pentru a fi utilizată cu orice algoritm de verificare care utilizează Procese Gaussienene, valorificând capacitatea acestora de a oferi estimări ale incertitudinii alături de predicțiile lor.

6.1 Probabilitatea de Eșec a Specificației (SFP)

Metrica SFP este definită ca probabilitatea ca răspunsul unui circuit să depășească pragul său de specificație. Aceasta este calculată pentru o anumită Configurație de Condiții de Operare (OCC) utilizând media și varianța prezise de modelul de Proces Gaussian pentru a determina probabilitatea ca valoarea răspunsului să intre depășească specificația. Pentru a oferi o singură estimare a riscului pentru un întreg răspuns, se calculează un SFP total. Acest lucru se face prin eșantionarea unui set mare de OCC-uri, grupându-le pentru a evita influența de la punctele corelate, identificând apoi SFP-ului cel mai nefavorabil în fiecare grup și apoi combinând aceste probabilități individuale pentru a obține un SFP total pentru răspuns. O prezentare generală a modului în care este calculat SFP-ul Total poate fi observată în Algoritmul 3.

6.2 Configurația experimentală

Validarea experimentală a metricii SFP s-a concentrat pe două tipuri distincte de circuite: sintetice și reale, alese pentru a oferi contexte de evaluare atât controlate, cât și realiste. Circuitul Sintetic 1 (SC1) a fost utilizat mai întâi datorită timpului său rapid de evaluare a funcțiilor și a valorilor reale minime/maxime cunoscute. Această caracteristică a permis o evaluare clară și fiabilă a acurateței metricii SFP. Complementar, a fost utilizat și un model de stabilizator de tensiune (VR3), reprezentând un circuit real și care are trei răspunsuri: Gain Margin, Phase Margin, and Power Supply Rejection Ratio (PSRR).

Algorithm 3: Probabilitatea de Eșec a Specificației

Input: Un set de N puncte de date OCC compuse din 3^L + *candidați OCC anteriori de Adaptive Planning*, unde L este numărul de condiții de operare de intrare

Output: Valoarea SFP totală pentru un răspuns

1 Se calculează valoarea SFP pentru fiecare punct de date ca probabilitatea estimării de a depășii specificația.
2 Se creează 10 grupuri folosind K-means, pe baza poziției punctelor de date în hiperspațiu.
3 Se creează vectorul SFP de lungime 10 pe baza grupurilor K-means.
4 **for** $i = 1$ **to** *10* **do**
5 | Se adaugă punctul de date cu cea mai bună valoare SFP la vectorul SFP.
6 **end for**
7 Se calculează Probabilitatea de Non-Eșec a Specificației (SNFP) ca $SNFP = 1 - SFP$
8 Se calculează valoarea SFP totală pentru răspuns ca: $Total\ SFP = 1 - \prod(SNFP)$

Aceste răspunsuri sunt supuse unor praguri operaționale specifice, cum ar fi Gain Margin > 8 dB, Phase Margin > 20 deg., and PSRR < –26 dB. În ambele scenarii, circuitele au fost tratate ca "black boxes", cu intrările lor limitate în intervale definite. Fiabilitatea SFP a fost evaluată prin compararea valorilor sale prezise cu limitele valorilor răspunsurilor cunoscute sau estimate, inclusiv cele obținute în timpul Fixed Planning și a iterațiilor de Adaptive Planning, precum și valorile minime sau maxime reale. Pe măsură ce valorile răspunsului s-au apropiat de pragurile de specificație, ar trebui să se obțină un SFP mai mare.

6.3 Rezultate Experimentale

6.3.1 Circuitul SC1 cu specificații care nu pot eșua

Evaluarea experimentală a metricii SFP a început cu circuitul sintetic SC1, configurat astfel încât specificațiile celor 25 de răspunsuri ale sale să fie setate în mod deliberat la 2% (din intervalul respectiv al răspunsului) sub minimele lor reale, fiind imposibil de eșuat. Rezultatele inițiale au arătat că mai multe răspunsuri au prezentat valori SFP care depășeau 10% după etapa de Fixed Planning și în primele iterații de Adaptive Planning. Până la iterația 4, toate răspunsurile au atins valori SFP sub 1,51%, indicând o probabilitate minimă de eșec a specificației, în concordanță cu designul circuitului. Acest lucru arată utilitatea SFP în semnalarea momentului în care eforturile ulterioare de verificare ar fi fost inutile.

6.3.2 Circuitul SC1 cu specificații care pot eșua

Într-un experiment ulterior, circuitul SC1 a fost configurat cu specificații setate la minimul real plus 1% din intervalul de răspuns, fiind posibile eșecurile. Pe măsură ce algoritmul se apropia de aceste specificații, ar trebui să se obțină valori SFP din ce în mai mari. Datorită simplității relative a unor răspunsuri, algoritmul de verificare a identificat eșecuri pentru majoritatea răspunsurilor în timpul etapei de Fixed Planning. Pentru răspunsurile rămase, unde nu au fost găsite eșecuri în Fixed Planning, metrica SFP a prezentat valori ridicate, justificând continuarea simulării. Algoritmul a identificat cu succes OCC-uri care au condus la eșecuri de specificație pentru toate aceste răspunsuri până la iterația 1 de Adaptive Planning. S-a observat că pentru răspunsuri mai complexe, valorile SFP inițial mai mici indicau necesitatea unor iterații suplimentare de Adaptive Planning pentru a reantrena modelele Procesului Gaussian în vederea luării unor decizii mai bune.

6.3.3 Circuitul VR3 cu specificații realiste

În cele din urmă, metrica SFP a fost aplicată unui circuit real, stabilizatorul de tensiune VR3, utilizând specificații realiste pentru cele trei răspunsuri ale sale: Gain Margin, Phase Margin, și PSRR. Pentru răspunsul Gain Margin, valorile SFP ridicate, începând de la 97,5% în timpul Fixed Planning și rămânând apoi ridicate, au prezis cu exactitate o probabilitate mare de eșec. Acest lucru a fost validat la iterația 48 de Adaptive Planning, când a fost descoperit un OCC care a dus la o valoare a Gain Margin de 7,92 dB, încălcând specificația > 8 dB. În schimb, SFP-ul răspunsului Phase Margin, inițial la 64,7%, a scăzut constant pe parcursul a 50 de iterații, indicând corect că nu au fost găsite eșecuri de specificație (28,6 deg față de specificația > 20 deg). Similar, SFP-ul răspunsului PSRR a rămas scăzut, ajungând la 0% până la iterația 20, ceea ce este în concordanță cu valoarea răspunsului de -34,1 dB, ce se încadra în specificația < -26 dB. Aceste rezultate arată colectiv că SFP poate ajuta inginerii de verificare să evalueze eficient probabilitatea unei defecțiuni a circuitului.

6.4 Discuție

Rezultatele experimentale au arătat că metrica SFP este un instrument fiabil și obiectiv care poate oferi o perspectivă ajutătoare inginerului de verificare. Aceasta prezice cu succes rezultatul în diverse scenarii, emițând avertismente din timp cu privire la potențialele defecțiuni sau indicând când un circuit este suficient de robust pentru a opri procesul de verificare mai devreme. Deși valoarea absolută a SFP depinde de calitatea modelului surogat, aceasta poate fi un bun indicator pentru compararea probabilității de eșec și pentru luarea deciziilor informate în timpul procesului de verificare.

Capitolul 7

Analiza bugetului de simulare

7.1 Propunere de Adaptive Planning cu Alocare Inegală a Bugetului

Acest capitol abordează problema alocării optime a unui buget limitat de simulări în timpul etapei de Adaptive Planning a algoritmului de Verificare Adaptivă pentru Circuite Analogice (AVAC). Abordarea standard alocă o simulare fiecărui răspuns la fiecare iterație de Adaptive Planning. Acest lucru poate fi ineficient, în special dacă unele răspunsuri au deja specificația lor eșuată sau au o probabilitate foarte scăzută de eșec. Ideea centrală propusă în acest capitol este de a realoca dinamic simulările de la răspunsurile care au deja specificația eșuată sau sunt considerate cu risc scăzut, la cele care au o Probabilitate de Eșec a Specificației (SFP) ridicată, cu scopul de a găsi potențiale eșecuri mai rapid.

7.2 Evoluția versiunilor propuse într-un scenariu simplu

Au fost considerate și analizate mai multe strategii de alocare dinamică a bugetului. Versiunile propuse au evoluat de la o abordare simplă de eliminare a răspunsurilor eșuate din bugetul de simulare (reducerea timpului total sau permiterea unor iterații suplimentare) la o strategie mai complexă de realocare activă a sloturilor de simulare libere către răspunsurile valide rămase (specificația respectivă nu a eșuat). Cele mai avansate dintre aceste propuneri inițiale au implicat utilizarea valorii SFP a fiecărui răspuns valid pentru a ghida realocarea, acordând mai multe simulări răspunsurilor cu un risc estimat de eșec mai mare.

7.3 Versiunea implementată a bugetului dinamic 4

Pe baza analizei versiunilor propuse, s-a dezvoltat o strategie mai complexă de alocare dinamică a bugetului, denumită versiunea 4. Această versiune are mai multe etape. În primul rând, orice răspuns care a eșuat specificația respectivă primește zero simulări pentru toate iterațiile viitoare. Răspunsurile valide rămase primesc fiecare câte o simulare de bază. Simulările suplimentare rămase de la răspunsurile eșuate sunt apoi distribuite între răspunsurile valide pe baza valorilor SFP din iterația anterioară, răspunsurile cu un SFP mai mare primind mai multe simulări. Orice simulări rămase după acest pas sunt alocate aleatoriu.

7.4 Versiunea implementată 4: evaluare și actualizări

Evaluarea experimentală a versiunii 4 a demonstrat eficacitatea sa. În comparație cu alocarea standard a bugetului, unde fiecare răspuns primește o simulare per iterație, strategia de alocare inegală a găsit toate eșecurile de specificație în doar patru iterații, în timp ce metoda standard a necesitat șapte. Acest lucru a confirmat că alocarea strategică a bugetului pe baza SFP poate accelera semnificativ descoperirea eșecurilor. O actualizare ulterioară a fost implementată pentru a permite utilizarea alocării bazate pe SFP chiar de la prima iterație, ceea ce a îmbunătățit performanța și mai mult, găsind toate eșecurile în doar trei iterații.

7.5 Versiunea implementată a bugetului dinamic 5

Una din problemele versiunii 4 constă în faptul că aloca întotdeauna cel puțin o simulare fiecărui răspuns valid, ceea ce poate fi ineficient pentru răspunsurile cu o probabilitate aproape zero de eșec. Pentru a rezolva acest lucru, a fost dezvoltată versiunea 5. Această versiune rafinează logica pentru scenariile în care numărul de simulări paralele disponibile este mai mic decât numărul de răspunsuri valide. În astfel de cazuri, versiunea 5 alocă simulări numai răspunsurilor cu cele mai mari valori SFP, omițându-le complet pe cele cu SFP scăzut și evitând crearea de loturi suplimentare de simulare.

7.6 Concluzii privind alocarea dinamică a bugetului

Analiza privind alocarea dinamică a bugetului demonstrează că versiunile 4 și 5, oferă alternative robuste care valorifică metrica de risc SFP pentru a prioritiza inteligent simulările. Prin concentrarea bugetului de simulare pe răspunsurile cu risc ridicat, aceste strategii pot reduce considerabil timpul și numărul de simulări necesare pentru a identifica potențiale eșecuri de specificație, făcând procesul general de verificare mai eficient.

Capitolul 8

Concluzii

8.1 Concluzii Generale

Obiectivul principal al acestei teze constă în dezvoltarea unui algoritm bazat pe învățare automată pentru verificarea Pre-Silicon a circuitelor integrate analogice. Cercetarea a fost motivată de complexitatea crescândă și cerințele stricte de fiabilitate ale industriei auto. Algoritmul dezvoltat de Verificare Adaptivă pentru Circuite Analogice (Adaptive Verification for Analog Circuits - AVAC) a fost construit printr-un proces iterativ, pornind de la o versiune de bază și evoluând în timp cu mai multe îmbunătățiri. Punctele cheie au inclus evaluarea diverselor strategii de eșantionare pentru colectarea inițială de date, cu o combinație de Orthogonal Arrays și Latin Hypercube Sampling dovedindu-se a fi cea mai eficientă. Algoritmul a fost apoi îmbunătățit prin introducerea unei etape de Adaptive Planning, care permite reantrenarea iterativă a modelelor surogat. Apoi, o metodă avansată de selecție a candidaților bazată pe abordarea Multi-Objective Acquisition Ensemble (MACE), care s-a dovedit a fi puțin mai bună decât metoda Gradient Descent cu un singur obiectiv. Performanța algoritmului a fost validată atât pe modele de circuite reale, cât și pe un benchmark sintetic ce cuprinde 900 de funcții, demonstrând robustețea și eficiența sa. Pentru a îmbunătăți în continuare procesul de verificare, au fost dezvoltate două extensii importante: metrica Probabilitatea de Eșec a Specificației (Specification Failure Probability - SFP), care oferă o estimare a riscului de eșec, și o strategie dinamică de alocare a bugetului de simulare care utilizează SFP pentru a prioritiza simulările pentru răspunsurile cu risc crescut și a găsi eșecurile mai rapid.

8.2 Contribuții originale

Contribuțiile personale ale autorului acestei teze pot fi rezumate astfel:

1. Dezvoltarea unei proceduri de verificare adaptivă bazată pe Învățare Automată pentru circuite analogice.

(a) Implementarea și evaluarea diverselor strategii pentru acoperirea hiperspațiului condițiilor de operare în timpul etapei de Fixed Planning a algoritmului de Verificare Adaptivă pentru Circuite Analogice, cum ar fi Orthogonal Arrays, Latin Hypercube Sampling și Monte Carlo.

(b) Implementarea și evaluarea versiunii de bază a algoritmului de Verificare Adaptivă pentru Circuite Analogice bazat pe Gradient Descent.

(c) Implementarea și evaluarea versiunii MACE a algoritmului.

2. Efectuarea unei analize detaliate și optimizarea algoritmului de Verificare Adaptivă pentru Circuite Analogice.

   (a) Efectuarea unui studiu aprofundat al ajustării hiper-parametrilor și al modificărilor algoritmului. Acestea includ studiul diferitelor funcții de achiziție și optimizatori pentru versiunea Gradient Descent a algoritmului.

   (b) Proiectarea și implementarea metricii de evaluare, Eroarea Relativă a Valorii, utilizată pentru evaluarea performanței algoritmului pe circuite sintetice.

3. Contribuție la proiectarea benchmark-ului sintetic, utilizat pentru validarea algoritmului de Verificare Adaptivă pentru Circuite Analogice.

4. Implementarea metricii de risc, Probabilitatea de Eșec a Specificației, ca un indicator al posibilității de eșec a specificației pentru răspunsul unui circuit.

5. Implementarea alocării bugetului Inegal (Dinamic), care atribuie un număr diferit de simulări pentru fiecare dintre răspunsurile circuitului în raport cu metrica Probabilitatea de Eșec a Specificației.

8.3 Lista publicațiilor originale

1. Manolache Cristian, Alexandru Caranica, Marius Stănescu, Horia Cucu, Andi Buzo, Cristian Diaconu, and Georg Pelz. "Advanced operating conditions search applied in analog circuit verification." In 2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), pp. 1-4. IEEE, 2022, DOI: 10.1109/SMACD55068.2022.9816209.

2. Manolache Cristian, Alexandru Caranica, Horia Cucu, Andi Buzo, Cristian Diaconu, and Georg Pelz. "Enhanced candidate selection algorithm for analog circuit verification." In 2022 International Semiconductor Conference (CAS), pp. 137-140. IEEE, 2022, DOI: 10.1109/CAS56377.2022.9934364.

3. Manolache Cristian, Cristina Andronache, Alexandru Caranica, Horia Cucu, Andi Buzo, Cristian Diaconu, and Georg Pelz. "Adaptive Planning Search Algorithm for Analog Circuit Verification." In 2023 International Semiconductor Conference (CAS), pp. 81-84. IEEE, 2023, DOI: 10.1109/CAS59036.2023.10303674.

4. Manolache Cristian, Cristina Maria Andronache, Alexandru Caranica, Horia Cucu, Andi Buzo, Cristian Vasile Diaconu, and Georg Pelz. "Applying Multi-objective Acquisition Function Ensemble for a candidate proposal algorithm." In 2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), pp. 116-121. IEEE, 2023, DOI: 10.1109/SpeD59241.2023.10314884.

5. Manolache Cristian, Cristina Andronache, Alexandru Guzu, Alexandru Caranica, Horia Cucu, Andi Buzo, and Georg Pelz. "Synthetic Benchmark for Data-Driven Pre-Si Analogue Circuit Verification." Electronics 13, no. 13 (2024): 2600. ISI IF 2.6 (Q2), DOI: 10.3390/electronics13132600, WOS:001269706000001.

6. Manolache Cristian, Alexandru Guzu, Cristina Andronache, Alexandru Caranica, Horia Cucu, Gazmend Alia, Andi Buzo, and Georg Pelz. "SFP: A Metric for Assessing Circuit Failure Probability During Pre-Si Verification". UPB Scientific Buletin, Sec. C Electr. Eng. Comput. ISI IF 0.2 (Q4), WOS:001380943300014.

8.4 Lista contractelor

- Asistent de cercetare (iunie 2021 – mai 2022) la Universitatea Politehnica din București, în proiectul „Efficiency and Effectiveness of Circuit Verification using Machine Learning Methods v2”, proiect de cercetare-dezvoltare finanțat de Infineon Technologies România & Co. SCS.

- Asistent de cercetare (iunie 2022) la Universitatea Politehnica din București, în proiectul „Efficiency and Effectiveness of Circuit Verification using Machine Learning Methods v3”, proiect de cercetare-dezvoltare finanțat de Infineon Technologies România & Co. SCS.

- Bursă privată de la Infineon Technologies România & Co. SCS (iulie 2022 - decembrie 2024).

- Asistent de cercetare (ianuarie 2025 – mai 2025) la POLITEHNICA București, în proiectul „Generative AI for Requirement Engineering (GenAI4RE)”, proiect de cercetare-dezvoltare finanțat de Infineon Technologies România & Co. SCS, contract nr. 4520242912.

8.5 Perspective pentru dezvoltări ulterioare

Cercetarea prezentată în această teză a dus la dezvoltarea unui algoritm funcțional disponibil în prezent pentru utilizare în cadrul unei companii industriale. Deoarece algoritmul a fost testat în principal pe circuite sintetice și un număr limitat de modele de circuite reale, feedback-ul din aplicarea sa pe o gamă mai largă de circuite reale ar fi binevenit pentru validări și îmbunătățiri ulterioare. O zonă cheie pentru lucrări viitoare este validarea suplimentară a metricii SFP. Deși rezultatele inițiale sunt promițătoare, sunt necesare teste mai extinse, în special pe benchmark-ul sintetic în diverse scenarii cu și fără posibilitatea de eșec, pentru a-i stabili pe deplin fiabilitatea. În cele din urmă, deși această teză se concentrează pe verificarea Pre-Silicon, algoritmul dezvoltat ar putea fi aplicat și pe verificări Post-Silicon. Ambele etape implică simulări în diferite OCC-uri, sugerând că algoritmul AVAC ar putea fi adaptat pentru a fi utilizat în această fază ulterioară a verificării circuitelor.

Bibliografie

[1] Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538.

[2] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

[3] Cressie, N. (1989). Geostatistics. *The American Statistician*, 43(4):197–202.

[4] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

[5] Devarajegowda, K. and Ecker, W. (2018). Meta-model based automation of properties for pre-silicon verification. In *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 231–236. IEEE.

[6] Dobler, M., Harrant, M., Rafaila, M., Pelz, G., Rosenstiel, W., and Bogdan, M. (2015). Bordersearch: An adaptive identification of failure regions. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1036–1041. IEEE.

[7] Farooq, U. and Mehrez, H. (2021). Pre-silicon verification using multi-fpga platforms: A review. *Journal of Electronic Testing*, 37(1):7–24.

[8] Garrido-Merchán, E. C. and Hernández-Lobato, D. (2020). Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35.

[9] Gaur, P., Rout, S. S., and Deb, S. (2019). Efficient hardware verification using machine learning approach. In *2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*, pages 168–171. IEEE.

[10] Gielen, G., Xama, N., Ganesan, K., and Mitra, S. (2019). Review of methodologies for pre-and post-silicon analog verification in mixed-signal socs. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1006–1009. IEEE.

[11] Gupta, A., Ramanath, R., Shi, J., and Keerthi, S. S. (2021). Adam vs. sgd: Closing the generalization gap on image classification. In *OPT2021: 13th Annual Workshop on Optimization for Machine Learning*.

[12] He, C., Hu, H., and Li, P. (2021). Applications for machine learning in semiconductor manufacturing and test. In *2021 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*, pages 1–3. IEEE.

[13] Hu, H., Zheng, Q., Wang, Y., and Li, P. (2018). Hfmv: Hybridizing formal methods and machine learning for verification of analog and mixed-signal circuits. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6.

[14] Khan, W., Azam, B., Shahid, N., Moeed Khan, A., and Shaheen, A. (2019). Formal verification of digital circuits using simulator with mathematical foundation. *Applied mechanics and materials*, 892:134–142.

[15] Knagg, O. (2019). An intuitive guide to gaussian processes. https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d. Accessed: 2024-08-02.

[16] Kukkonen, S. and Lampinen, J. (2005a). Gde3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation*, volume 1, pages 443–450. IEEE.

[17] Kukkonen, S. and Lampinen, J. (2005b). GDE3: the third evolution step of generalized differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 443–450 Vol.1. ieeexplore.ieee.org.

[18] Lim, H. G., Jang, J., Ju, B. K., Ko, J. W., and Kim, C. O. (2023). A test vector selection method based on machine learning for efficient presilicon verification. *Expert Systems with Applications*, 224:120056.

[19] Lyu, W., Yang, F., Yan, C., Zhou, D., and Zeng, X. (2018). Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*, pages 3306–3314. PMLR.

[20] McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

[21] Montgomery, D. (2004). *Design and Analysis of Experiments (pp. 71–198)*. John wiley & sons.

[22] Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, pages 439–452.

[23] Owen, A. B. (2013). *Monte Carlo theory, methods and examples*. Stanford.

[24] Pronzato, L. and Müller, W. G. (2012). Design of computer experiments: space filling and beyond. *Stat. Comput.*, 22(3):681–701.

[25] Robič, T. and Filipič, B. (2005). Differential evolution for multiobjective optimization. In *International conference on evolutionary multi-criterion optimization*, pages 520–533. Springer.

[26] Şandru, E.-D., David, E., and Pelz, G. (2019). Pre-silicon yield estimation using machine learning regression. In *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 103–104. IEEE.

[27] Stănescu, M., Vişan, C., Sandu, G., Cucu, H., Diaconu, C., Buzo, A., and Pelz, G. (2021). Multi-objective optimization algorithms for automated circuit sizing of analog/mixed-signal circuits. In *2021 International Semiconductor Conference (CAS)*, pages 117–120. IEEE.

[28] Visan, C. (2024). *Artificial Intelligence Techniques For Integrated Circuit Design Automation*. PhD thesis, National University of Science and Technology Politehnica of Bucharest, Bucharest, Romania.

[29] Vişan, C., Pascu, O., Stănescu, M., Cucu, H., Diaconu, C., Buzo, A., and Pelz, G. (2021). Versatility and population diversity of evolutionary algorithms in automated circuit sizing applications. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 68–73. IEEE.

[30] Zhang, S., Yang, F., Yan, C., Zhou, D., and Zeng, X. (2021). An efficient batch-constrained bayesian optimization approach for analog circuit synthesis via multi-objective acquisition ensemble. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(1):1–14.

[31] Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H., et al. (2020). Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33:21285–21296.

[32] Zhuo, C., Yu, B., and Gao, D. (2017). Accelerating chip design with machine learning: From pre-silicon to post-silicon. In *2017 30th IEEE International System-on-Chip Conference (SOCC)*, pages 227–232. IEEE.