

TESTAREA COMPACTĂ RAM: INSTRUMENTAR ȘI REZULTATE

G. Bodean

Universitatea Tehnică a Moldovei

INTRODUCERE

Memoria operativă cu acces aleatoriu (RAM) sunt dispozitive digitale simple ce execută reținerea datelor. În același timp, RAM este cel mai complex dispozitiv din punct de vedere al gradului de integritate. Dispozitivele de tip sistemă-pe-cip (system-on-chip, SoC) de asemenea conțin blocuri RAM incorporate (embedded), ponderea cărora va predomina în viitoarele SoC [1]. Prin urmare costul unui cip RAM cât și a unui SoC va fi determinat în mare măsură de costul testării memoriei.

Eficiența testării RAM este determinată de numărul modelelor de defectări analizate și de calitatea testelor elaborate. În ultimii ani zeci de modele de defectări RAM au fost introduse. Pentru detectarea lor, în parte sau în ansamblu, au fost elaborate teste de tip March [2, 3]. Algoritmii March predomină încă în tehnicile testării RAM. Dar, din cauza complexității controlerelor, testele March nu-s acceptabile pentru testarea încorporată [4].

Sunt cunoscute mijloacele de compactare a reacțiilor unității testate. Analizorul de semnături este aplicat cât pentru testarea atât și pentru diagnosticarea RAM. Dar acest instrumentariu nu-i compatibil cu algoritmii March [5, 6]. Una din tehnicile mai adecuate pentru testarea compactă (adică generarea testelor) RAM este așa numita metoda pseudo-inelară, propusă în [7].

În acest articol sunt analizate două scheme de bază ale testării pseudo-inelare (vezi compartimentul 2). În contextul executării și verificării testării pseudo-inelare (sau π -testării) este propus și prezentat (în compartimentul 3) instrumentarul soft de simulare RAM. În partea finală a articolului (vezi compartimentul 4) sunt analizați algoritmii de π -testare a defectărilor statice și dinamice RAM.

1. SCHEME DE π -TESTARE RAM

Conceptul de testare compactă RAM (sau π -testare) este prezentat în [7, ..., 10]. Să remarcăm doar faptul că π -testarea este bazată pe emularea automatului liniar, în particular registrul de deplasare cu legătură de reacție (LFSR), de însăși unitatea testată, adică RAM. Emularea LFSR poate

fi perfectată parțial sau complet. Schemele de π -testare pentru ambele cazuri sunt prezentate în fig. 1. În cazul implementării schemei parțiale de π -testare, numită *ring*, celulele de memorie joacă rolul legăturii de reacție (feedback) a structurii LFSR (fig. 1, a). În modul complet de π -testare, așa numita testare *scan*, rolul celulelor (pozițiilor) registrului de deplasare în structura LFSR îl joacă celulele de memorie (fig. 1, b). De aceea LFSR, sau automatul, se interpretează a fi *virtual*.

În modul de testare scan (v. fig. 1, b) registrul ShReg este folosit numai pentru salvarea temporară a datelor. Registrul suplimentar „dispare” din schema scan, în cazul când numărul de porturi RAM este egal cu cel al pozițiilor LFSR.

Deci, în schema scan unitatea ShReg este copia registrului virtual a automatului, emulat de celulele de memorie. La fiecare impuls de clock (tact) registrul virtual este deplasat relativ de date (v. Fig. 1). Generatorul de adrese GenA determină traiectoria deplasării automatului. Pot fi definite trei tipuri de traiectorii: numărare directă (counting up), numărare inversă (counting down) și pseudoaleatoare (pseudorandom). În cazul unei traiectorii pseudoaleatorii în calitate de generator GenA este folosit așa numitul numărător LFSR (counter-LFSR). În același timp, și în calitate de structură a automatului virtual poate fi selectată ori schema clasică LFS ori schema counter-LFSR (sau maximum-length LFSR [11]). În ambele cazuri

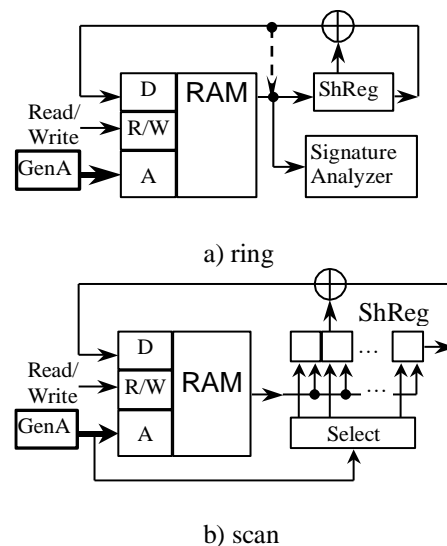


Figura 1. Scheme de π -testare.



Figura 2. Bara principală de instrumente a softului de simulare a π -testării.

pentru implementarea LFSR este selectat din lista propusă un polinom ireductibil $p(x)$.

Memoria testată poate fi bit- sau word-orientată. În cazul RAM bit-orientat automatul este un LFSR clasic ce procesează datele asupra câmpului Galois $\mathbf{GF}(2)$. În cazul RAM word-orientate automatul are o structură generală [12] și procesează datele asupra $\mathbf{GF}^k(2^m)$, unde $m = \deg p(x)$ este binaritatea celulei de memorie și $k = \deg q(x)$ este gradul polinomului ireductibil asupra $\mathbf{GF}^k(2^m)$. De regulă valoarea lui k este multiplă capacității matricei de memorie. În cazul trivial, $k = 2$.

Π -testarea RAM constă în executarea a cel puțin a unei iterații de π -testare care este definită de următorii parametri:

- Starea inițială (seed) **Init**;
- Modul de adresare ori tipul traiectoriei;
- Inversarea datelor de intrare sau intrare-ieșire;
- Numărul de operații citire-înscrisiere executate (este specific pentru unele defectări, de ex. Defectările dinamice).

În rezultatul executării π -iterației, automatul ajunge în starea finală **Fin**, care este comparată cu cea așteptată (în particular, egală cu **Init**).

Remarca 1: inversarea datelor permite extinderea varietății stărilor automatului, ceea ce nu-i specific pentru automatele cunoscute. De exemplu, inversarea dublă a datelor (intrare-ieșire) permite tranzitarea automatului prin stările 11...1 (adică setarea celulelor de memorie în starea logic 1). Prin urmare, π -testarea permite simularea algoritmilor March.

Specificarea parametrilor π -iterațiilor este esența problemei sintetizării algoritmului de π -testare RAM. Scopul π -testării este de detecta lista defectărilor specificate. Încă nu-i elaborat procedeul de automatizare a sintetizării algoritmilor de π -testare. Prin urmare, inginerul de testare trebuie euristic să elaboreze π -testele și să le verifice cu ajutorul instrumentarului soft de simulare RAM.

2. INSTRUMENTARUL SOFT DE SIMULARE A Π -TESTĂRII RAM

Pentru sintetizarea și depanarea testelor de testare pseudo-inelară RAM a fost elaborat instrumentarul soft specializat, numit Development and Simulation Tools Environment (DSTE) [10]. Bara principală de instrumente conține meniul de bază și barele standard, custom și de depanare (fig. 2).

Crearea unui π -test începe cu: asignarea denumirii proiectului, tipul memoriei – uni- sau bi-port, binaritatea celulei și modul de testare – ring sau scan (fig. 3). Apoi continuă cu specificarea parametrilor analizorului de semnături (fig. 4), automatului virtual (fig. 5) și ale iterațiilor de testare (fig. 6). Toate iterațiile specificate vor apărea în fereastra Input-Output (fig. 7).

Proiectul π -testării astfel specificat poate fi salvat în mapa de lucru. În continuare este lansat π -testul RAM fără defectări. Pentru depanarea testului pseudo-inelar inginerul poate utiliza bara respectivă (v. Fig. 8). În rezultat se obțin reacțiile așteptate (etalon), cât ale memoriei atât și ale analizorului de semnături (vezi coloanele Out și SA în fig. 7).

După ce π -testul este sintetizat, trebuie să fie „apreciată” calitatea acestuia, adică performanțele

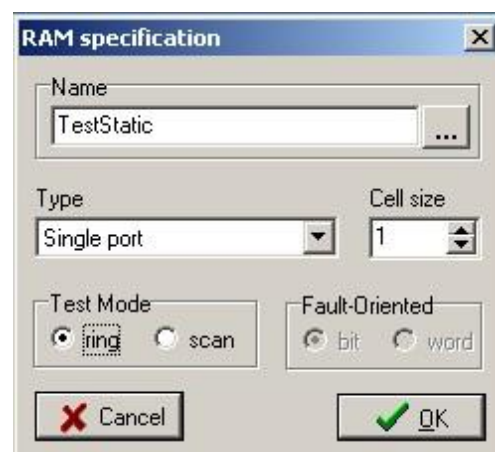


Figura 3. Specificarea RAM.



Figura 4. Specificarea analizorului de semnături.

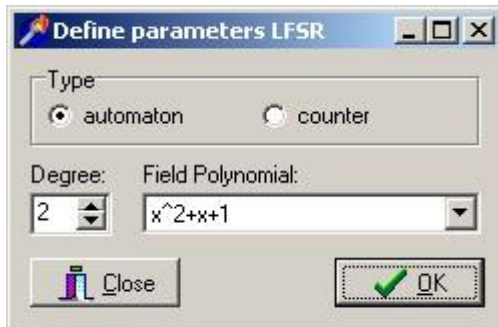


Figura 5. Specificarea automatului virtual.

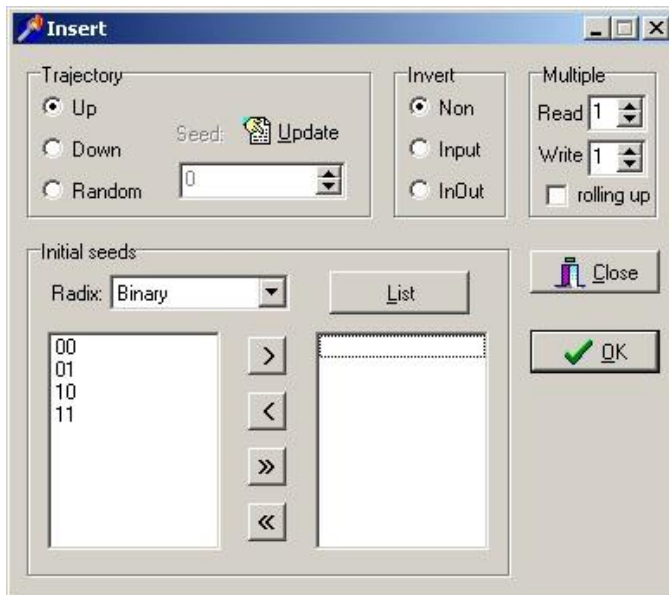


Figura 6. Specificarea iterațiilor de π -testare.

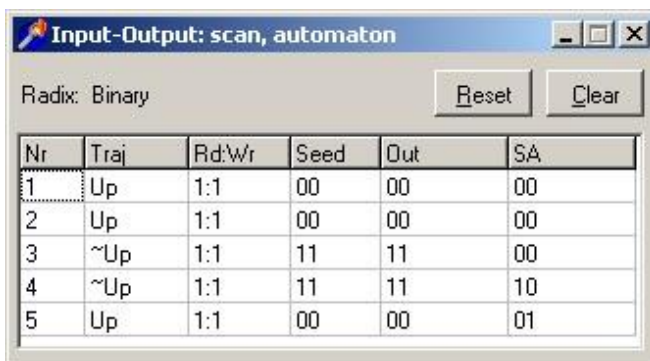


Figura 7. Reacțiile etalon a π -testării pentru defectări statice.

lui de detectare a defectărilor (acoperire, fault coverage). Fiind activizat butonul Fault F...(fig. 8), apare fereastra de specificare a defectării (fig. 9). După selectarea tipului, modelului și a valorii defectării, utilizatorul lansează experimentul de testare. În rezultat, în fereastra Input-Output apar stările **Fin** curente ale automatului virtual și ale analizorului de semnături (v. coloanele #Out și #SA în fig. 7).

„Masa de lucru” a simulării RAM este prezentată în fig. 10. Această fereastră este o interfață pasivă în care sunt reflectate rezultatele curente de depanare a π -testelor.

Rezultatele curente ale π -testării sunt comparate cu cele etalon, iar rezultatul comparării este salvat în fereastra respectivă (fig. 11). În această fereastră acoperirea defectărilor este prezentat printr-o diagramă, iar calitatea (rezoluția) π -testării R este estimată prin formula:

$$R = F_d / F_\Sigma \quad (1)$$

unde F_d este numărul de defectări detectate, F_Σ este numărul total de defectări

Utilizatorul poate aposteriori să selecteze un subset de rezultate, înserând simbolul \checkmark asupra coloanelor respective. În rezultat valoarea rezoluției va fi recalculată.

Simulatorul DSTE conține o listă sofisticată de defectări funcționale RAM. Fereastra cu lista modelelor de defectări este prezentată în fig. 9. Iar în compartimentul următor vor fi prezentate testele și rezultatele π -testării RAM pentru defectările statice și dinamice.

3. REZULATELE Π -TESTĂRII

În această secțiune vor fi prezentate și analizate testele și rezultatele testării pseudo-inelare RAM bit-orientate pentru defectările funcționale statice și dinamice.

3.1. Π -testarea defectărilor unicelulare statice

Clasa defectărilor statice unicelulare (single-cell) conține 14 tipuri modele de defectări funcționale, însumând 34 primitive [13]. Precum este menționat în [13], zece (din 14) tipuri de defectări sunt realiste. Dar, din de vedere teoretic, π -testul a fost sintetizat pentru toate celea 14 tipuri de defectări statice single-cell.

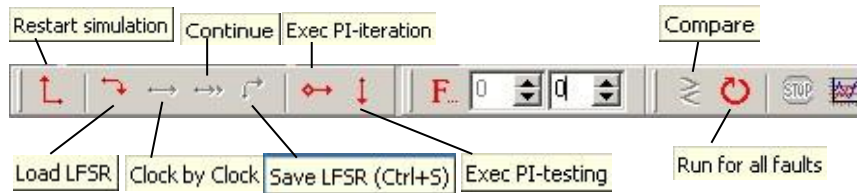


Figura 8. Bara de instrumente de depanare a π -testelor.

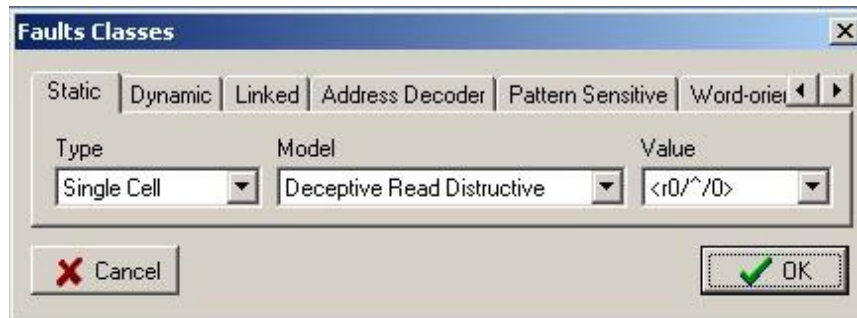


Figura 9. Specificarea defectării.

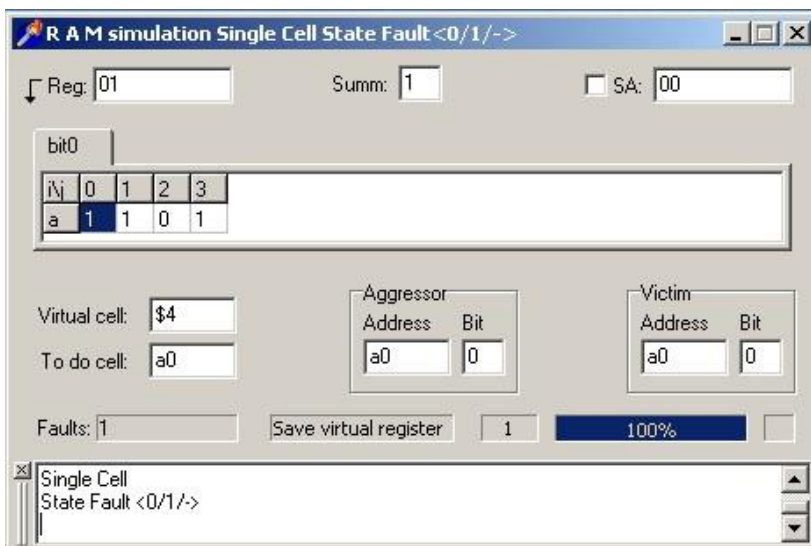


Figura 10. Test-Bench-ul sistemului de simulare RAM.

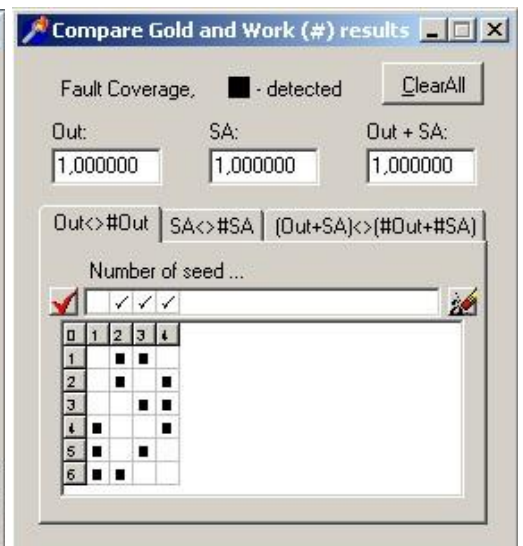


Figura 11. Fereastra rezultatelor π -testării.

Figura 7 conține iterațiile de π -testare cu rezoluția 100% a 7 tipuri de defectări **deterministe** statice: stare (state SF), tranziție (transition TF), înscriere (write WDF) și citire destructivă (read destructive RDF), citire deceptiv-destructivă (deceptive read destructive DRDF), citire incorectă (incorrect read IRF) și blocat în (stuck-At SAF). De asemenea, aceste π -teste detectează toate defectările de citire aleatoriu-destructive (random read destructive faults RRDF).

În π -testele susmenționate automatul linear are structură trivială, adică $p(x) = x^2 + x + 1$. Π -iterațiile nr. 1 și 2 sunt executate cu traiectoria count up și citirea-înscrierea o singură dată (Rd: Wr= 1:1). Totodată, la fiecare deplasare a automatului virtual se execută 2 operații de citire și 1 de înscriere. Deci, complexitatea (lungimea) unei π -iterații este egală cu $3N$, unde N este capacitatea matricei de

memorie. Lungimea π -testului pentru toate celea 7 tipuri de defectări deterministe statice specificate este egală cu $15N$ (fără conectarea analizorului de semnături!).

Restul defectărilor statice – citirea aleatoriu-destructivă (Random Read Destructive RRDF), citirea aleatorie (random read RRF), stare nedeterminată (undefined state USF), citire (undefined read URF) și înscriere nedeterminată (write UWF), nu-i acces (no access NAF) și reținerea datelor (data retention faults DRF), sunt de natură **probabilistică**. Pentru determinarea modelului de statistică și pentru derularea încercărilor, utilizatorul va aplica instrumentul Run Graphical Window (fig. 12).

Însă un deosebit interes prezintă determinarea modelului teoretic al repartițiilor.

În experimentele de π -testare asupra fiecărei celule RAM se execută una și aceeași secvență de

operații. Fiecare iterație se execută independent. Experimentul de π -testare constă în executarea a 5 iterații de testare. Deci, poate fi ușor determinat că rezultatele încercărilor în testul Single urmează legea binomială de repartiție:

$$P_n(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k=0,1,\dots,n, \quad (2)$$

unde p este probabilitatea de nedetectare a defectării într-o π -iterație; $P_n(k)$ este probabilitatea a k succese (cota parte a defectărilor detectate) în n încercări.

Valorile probabilităților p , rezultate din analiza modelelor matematice ale defectărilor statice sunt prezentate în tabelul 1.

Tabelul 1. Probabilitățile de nedetectare a defectărilor statice aleatorii single-cell.

#	Tipul defectării	p
1	UWF toate	1/2
2	DRF* <0T/?/->, <1T/?/->; URF <r1/?/1>	1/4
3	URF <r0/?/0>	1/8
4	RRF <r1/1/?>; DRF <1T/↓/->	1/16
5	RRF <r0/0/?>; DRF <0T/↑/->; URF <r1/?/?>	1/64
6	URF <r0/?/?>	1/512
7	NAF	1/1024

Remarca 2: este cunoscut faptul că în cazul când $p \leq 0,1$ atunci repartiția (2) poate fi aproximată de legea Poisson:

$$P_n(k) \approx \lambda^k e^{-\lambda} / k!, \quad \text{unde } \lambda = n p. \quad (3)$$

În același timp, dacă $p > 0,1$, atunci poate fi aplicată legea locală Laplace. Deci, repartițiile pentru ultimele 4 defectări pot fi approximate de legea Normală.

Cunoscând legea de repartiție, producătorul poate determina numărul necesar de încercări a π -testării cu scopul de a asigura nivelul acceptabil al riscului.

Din punctul de vedere al calității testării, cel mai bun March-algorithm este SRD+, complexitatea căruia este egală cu 22N [13].

3.2. Π -testarea defectărilor bi-cellulare statice

Clasa defectărilor statice bicelulare (two-cell) conține 15 tipuri modele de defectări funcționale, însumând 86 primitive [13]. Numai șapte (din 15) tipuri de defectări sunt realiste. Dar π -testul a fost sintetizat pentru toate celea 15 tipuri de defectări statice two-cell.

Tabelul 2 conține iterațiile de π -testare cu rezoluția 100% a 9 tipuri de defectări **deterministe** statice a cuplului: stare (state CFst), perturbare (disturb CFds), idempotente (CFid), inversie (CFin), tranziție (CFtr), înscriere (write destructive CFwd) și citire destructivă (read destructive CFrd), citire decetiv-destructivă (deceptive read destructive CFrd), citire incorectă (incorrect read CFir). Totodată, aceste π -teste detectează toate defectările de citire aleatorie-destructivă (read destructive faults RRDF).

În testul, prezentat în tabelul 2, este utilizată structura counter-LFSR cu $p(x) = x^2 + x + 1$. Iterațiile

Tabelul 2. Π -testul **Double** cu schema RING și COUNTER-LFSR.

Nr	Traj	Rd:Wr	Seed	Out	SA
1	~Up	1:1	00	10	00
2	~Up	1:1	00	10	00
3	~Up	1:1	11	01	00
4	~Up	1:1	11	01	10
5	~Up	1:1	00	10	01
6	~Up	1:1	11	01	10
7	~Down	1:1	00	10	01
8	~Down	1:1	00	10	10
9	~Down	1:1	11	01	11
10	~Down	1:1	00	10	11

nr. 1,...,6 sunt executate cu traiectoria count up, iar celelalte 4 iterații – cu traiectoria count down. Simbolul ~ specifică inversarea datelor la intrarea RAM.

Să estimăm performanța testului. Deoarece analizorul de semnături (SA) este conectat la ieșirea RAM, atunci complexitatea iterației de testare este egală cu 4N, iar lungimea π -testului pentru toate celea 9 tipuri de defectări deterministe a cuplului este egală cu 40N.

Restul defectărilor din clasa analizată, și anume: stare nedeterminată (undefined state CFus), perturbare nedeterminată (undefined disturb CFud), citire aleatorie (random read CFrr), citire aleatorie-destructivă (random read destructive CFrrd), înscriere nedeterminată (undefined write CFuw), citire nedeterminată (undefined read CFur) sunt de natură **probabilistică**. Legea normală de repartiție este valabilă pentru aceste defectări. Valoarea medie de detectare a defectării poate fi obținută prin depanarea experimentului statistic de testare (vezi compartimentul 3.4).

Anticipăm că testul Double nu detectează următoarele defectări statice de tipul cuplu: <1w1;...> și <0w0;...> în CFds, CFud, CFwd. Aceste defectări pot fi considerate cazuri particulare a defectărilor dinamice.

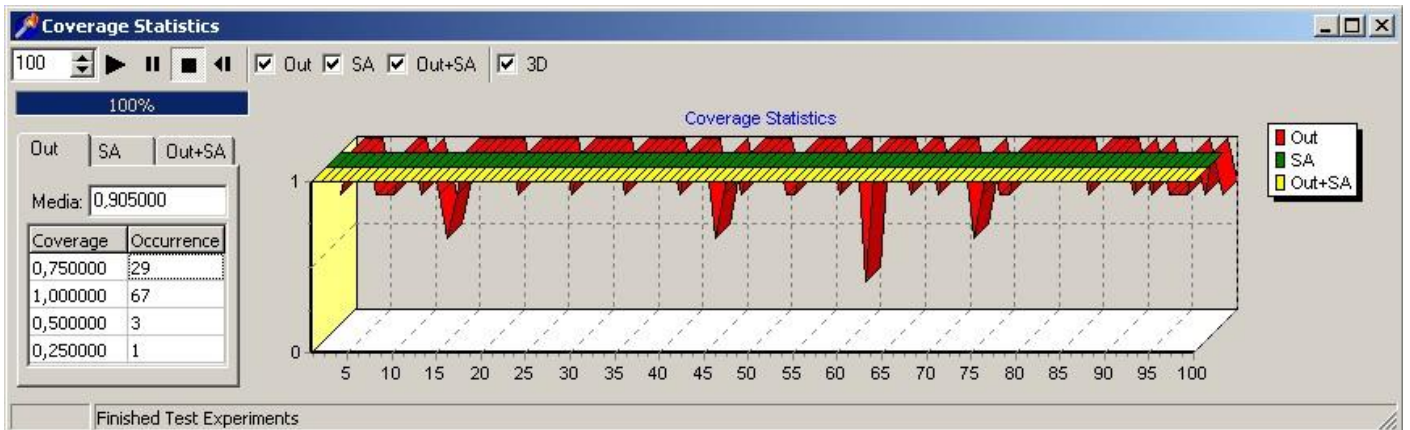


Figura 12. Fereastra pentru analiza statistică a π -testării.

3.3. Π -testarea defectărilor uni-cellulare dinamice

La început va fi prezentat π -testul pentru defectările statice a cuplului, și anume: CFds $\langle 1w1; 0/\uparrow/- \rangle$ și $\langle 0w0; 1/\downarrow/- \rangle$, CFud $\langle 1w1; 0/?/- \rangle$ și $\langle 0w0; 1/?/- \rangle$, CFwd $\langle 0; 1w10/\downarrow/- \rangle$ și $\langle 1; 0w0/\uparrow/- \rangle$, Cfuw $\langle 0; 1w1/?/- \rangle$ și $\langle 1; 0w0/?/- \rangle$. Π -testul acestor defectări este prezentat în tabelul 3.

Tabelul 3. Π -testul Special cu schema RING și COUNTER-LFSR.

Nr	Traj	Rd:Wr	Seed	Out	SA
1	Up	1:1	00	00	00
2	"Up	1:2	00	00	11
3	Up	1:2	00	00	01
4	"Down	1:2	00	00	00
5	Down	1:2	00	00	11

În testul Special este aplicată structura counter-LFSR cu $p(x) = x^2 + x + 1$ și SA conectat. Iterațiile nr. 1,2,3 sunt executate cu traiectoria count up, iar iterațiile nr. 4 și 5 – cu traiectoria count down. Totodată în iterațiile nr. 2 și 4 are locul inversarea dublă (la intrarea și ieșirea RAM) a datelor (vezi simbolul ") și iterațiile nr. 2,...,5 conțin o instrucțiune de citire (Rd= 1) și două instrucțiuni de înscriere (Wr= 2). Astfel, lungimea iterației nr. 1 este egală cu 4N, iar celelalte iterații ale testului Special au lungimea egală cu 5N. Deci, în total, complexitatea testului Special este egală cu 24N.

Testul Special este o introducere în π -testul defectărilor dinamice. Clasa defectărilor unicelulare (single-cell) dinamice conține 5 tipuri de modele a defectărilor funcționale, însumând 30 primitive [14]. Este necesar de executat secvențial mai mult de o operație citire sau înscriere (2 sau 3 în cazul analizat) pentru a detecta o defectare dinamică. Iată anume de ce schema ring cu autodeplasare (self-rolling și boxa rooling up este selectată, v. Fig. 6)

este preferabilă pentru testarea defectărilor dinamice. În acest regim de funcționare automatul operează în mod autonom, executând operația de citire a memoriei numai după dezvoltarea operațiilor de înscriere specificate.

Remarca 3: pentru executarea citirii secvențiale se impune folosirea schemei scan de π -testare.

Prin urmare, π -testarea defectărilor dinamice necesită utilizarea atât a schemei ring, cât și a schemei scan. Prin extindere vom aplica π -testul deja elaborat, și anume testul Special, pentru analiza performanțelor acestuia referitor la defectările dinamice. Simularea RAM a arătat că testul Special detectează defectările dinamice dynamic RDF (dRDF) și dIRF, și parțial defectările dynamic DRDF, TF și WDF.

În tabelul 4 este prezentat π -testul, numit Dynamic I (fără SA), destinat pentru detectarea defectărilor DRDF $\langle 0w1r1/0/1 \rangle$ și $\langle 1w0r0/1/0 \rangle$, WDF de tipul $\langle 0w0w0/1/- \rangle$, $\langle 1w1w1/0/- \rangle$, $\langle 0w1w1/0/- \rangle$ și $\langle 1w0w0/1/- \rangle$. Complexitatea testului Dynamic I este egală cu 30N.

Tabelul 4. Π -testul Dynamic I cu schemele SCAN și AUTOMATON.

Nr	Traj	Rd:Wr	Seed	Out
1	Up	1:3	00	00
2	"Up	1:3	00	00
3	Up	1:2	00	00
4	"Up	1:2	00	00
5	Up	1:1	00	00
6	"Up	1:1	00	00

Tabelul 5 prezintă π -testul cu rolling-up (simbolul @) pentru detectarea defectărilor dinamice de tranziție (dTf), nedectate de testele Special și Dynamic I. Complexitatea testului Dynamic II este egală cu 16N.

Tabelul 5. Π -testul Dynamic II cu schemele RING și AUTOMATON.

Nr	Traj	Rd:Wr	Seed	Out
1	~Up	@1:3	10	11
2	Up	@1:3	01	01
3	~Up	@1:3	01	00
4	Up	@1:3	10	10

Din analiza tabelelor 3, 4 și 5 rezultă că lungimea π -testelor pentru toate defectările unicelulare (single-cell) dinamice este egală cu 60N. Testul March DS1 [15], destinat pentru detectarea defectărilor unicelulare dinamice, are complexitatea egală cu 43N.

3.4. Π -testarea defectărilor statice și dinamice: sumar

Una din regulile „de aur” a testării RAM prescrie ca de fiecare dată când un test nou este elaborat, calitatea acestuia pentru *alte* defectări trebuie de verificat. În cele ce urmează vor fi analizate rezoluțiile testelor elaborate – Single, Double, Special, Dynamic I și Dynamic II, referitor la defectările non-scop, pentru care aceste teste au fost proiectate.

Primordial analizăm performanțele de acoperire pentru subclasa defectărilor realiste deterministe. Lista acestor defectări este următoarea:

- statice single-cell: SF(2), TF(2), WDF(2), RDF(2), DRDF(2), IRF(2) și SAF(2);
- statice two-cell: CFst(2), CFds(12) \subset {Cfid(4) \cup Cfin(4)}, CFtr(4), CFwd(4), CF(2), IRF(2) și SAF(2);
- dinamice single-cell: dRDF(6), dDRDF(6), dIRF(6), dTF(6) și dWDF(6).

În paranteze este specificat numărul de defectări. În tabelul 6 sunt prezentate rezultatele „acoperirii” defectărilor analizate. Testele cu celule evidențiate sunt complementare referitor la defectările corespunzătoare. De asemenea, din tabelul 6 rezultă că testele Special și Dynamic II detectează toate (în afară de dWDF) defectările deterministe ale listei specificate. Complexitatea

acestor teste este egală cu 40N. Pentru detectarea defectărilor dWDF este necesar de aplicat adițional primele 4 iterații ale testului Dynamic I (fără SA !). Deci, complexitatea testelor Special+Dynamic II+(4 iterații) Dynamic I este egală cu 54N.

În tabelul 7 sunt prezentate rezultatele simulării π -testelor pentru defectările probabilistice.

Tabelul 7. Valorile medii ale probabilităților de detectare a defectărilor aleatorii realiste.

π -testul	Static Single-Cell				Static Two-Cell		
	RRF	UWF	URF	DRF	CFrr	CFuw	CFur
Single	0,962	0,5	0,94	0,824	0,762	0,25	0,63
Double	0,998	0,757	0,986	0,95	0,91	0,47	0,89
Special	0,997	0,75	0,983	0,927	0,95	0,494	0,895
Dynamic I	1	0,83	0,988	0,963	0,92	0,402	0,83
Dynamic II	0,991	0,82	0,871	0,93	0,84	0,485	0,68

Remarca 4: în unele experimente de test rezoluția π -testării (fără SA) a fost mai mare decât rezoluția analizorului de semnături, când acesta era conectat, și viceversa pentru alte tipuri de defectări. Dar spre deosebire de SA, parametrii automatului virtual sunt controlabili.

CONCLUZII

În lucrare sunt prezentate schemele și mijloacele de testare compactă, numită π -testare, a dispozitivelor de memorie RAM. Schemele de testare – *ring* și *scan*, pot fi aplicate separat sau mixt. Elementele π -testării pot fi incorporate în circuitul RAM sau implementate exterior referitor la unitatea testată. Trebuie de remarcat că, în cazul incorporării, mijloacele π -testării devin compatibile cu tehnicile cunoscute de testare cu scanare periferică (vezi standardul IEEE 1149).

Aplicarea π -testării pentru autotestarea RAM este particularitatea esențială a tehnicii propuse. Autotestarea este asigurată prin emularea automatului liniar prin intermediul componentelor RAM. Ca urmare, sintetizarea π -testelor se reduce la programarea comportamentului automatului. Scopul

Tabelul 6. Rezoluția testelor deterministe.

π -testul	Static Single-Cell							Static Two-Cell							Dynamic Single-Cell				
	SF	TF	WDF	RDF	DRDF	IRF	SAF	CFst	CFds	CFtr	CFwd	CFrd	CFrd	CFir	dRDF	dDRDF	dIRF	dTF	dWDF
Single	1	1	1	1	1	1	1	3/4	7/12	1/4	1/2	7/8	5/8	7/8	1	1	1	1/3	1/3
Double	1	1	1	1	1	1	1	1	5/6	1	1/2	1	1	1	1	2/3	1	1/3	1/3
Special	1	1	1	1	1	1	1	1	23/24	1	1	1	1	1	1	2/3	1	1/3	1/3
Dynamic II	1	1	1	1	1	1	1	1	19/24	7/8	1	7/8	1/2	7/8	1	1/3	1	2/3	1/3
Dynamic I	1	1	1	1	1	1	1	3/4	2/3	1/2	1/2	1	5/8	1	1	2/3	1	1/3	2/3

descrierii comportamentului este de a detecta defectările din lista prescrisă (predeclarată) de defectări. Comportamentul automatului fiind determinist și controlabil există premisele pentru automatizarea procedurii de sintetizare a algoritmilor de π -testare.

Tehnica de π -testare compactă poate fi aplicată cât asupra RAM bit-orientată atât și asupra cea word-orientată. În lucrare sunt prezentate π -testele defectărilor unicelulare (single-cell) statice și dinamice, și pentru toate defectările bicelulare (two-cell) statice. Testele elaborate au fost verificate cu ajutorul softului de simulare RAM. Complexitatea π -testelor elaborate este comparabilă cu cea a March-testelor corespunzătoare. Iar pentru defectările probabilistice au fost determinate legile de repartiție, cunoașterea cărora permite controlul calității producției de circuite RAM.

Bibliografie

1. Semiconductor Industry Association (SIA), "International Technology Roadmap for Semiconductors (ITRS)", 2003.
2. A. van de Goor. *Testing Semiconductor Memories*, ComTex Publishing, Gouda, The Netherlands, 1998.
3. Hamdioui S, *Testing Static Random Access Memories: Defects, Fault Models and test Patterns*, Kluwer Academic Publisher, 2004.
4. Huang C.-T., Huang J.-R., Wu C.-F., Chang T.-Y. "A Programmable BIST core for embedded DRAM," *IEEE Design & Test of Computers*, vol. 16, No.1, 1999, pp. 59-70.
5. Hellebrand S., Wunderlich H.-J., Ivaniuk A. A., Klimets Y. V., Yarmolik V.N. "Efficient Online and Offline Testing of Embedded DRAMs," *IEEE Transactions on Computers*, vol. 51, Issue 7, July 2002, pp. 801- 809.
6. Rajski J., Tyszer J. "Test Data Compression and Compaction for Embedded Test of Nanometer Technology Designs," *Proc. of the 21st International Conference on Computer Design*, Oct 2003, pp.331-336.
7. Bodean Gh. "PRT: Pseudo-Ring Testing – A Method for Self-Testing RAM," *IEEE-TTTC Int. Conf. On Automation, Quality and Testing, Robotics: AQTR 2002 (THETA 13)*, Tome 1, Cluj-Napoca, Romania, May 2002, pp.295-300.
8. Bodean Gh., Bodean D., Labunetz A. "New Schemes for Self-Testing RAM," *Design, Automation and Test in Europe (DATE 2005)*, Munich, Germany, March 2005.
9. Bodean Gh. *Aspecte teoretice ale testării pseudoinelare a memoriei operative Meridian Ingineresc*, Nr. 3, 2005, pp. 11-18.
10. Bodyan G. "Pseudo-Ring Test RAM: Concept, Test Algorithms, and Tools," *Proc. IEEE East-West Design & Test Workshop (EWDTW'07)*, Yerevan, Armenia, Sept 2007, pp. 335-339.
11. McCluskey E. J. *Logic Design Principles with Emphasis on Testable Semicustom Circuits*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1986.
12. Pradhan D. K. and Chatarjee M. "GLFSR – A New Test Pattern Generator for Built-in-Self-Test," *IEEE Trans. on CAD*, vol.18, no. 2, Feb 1999, pp.238-247.
13. Hamdioui S. *Testing Multi-Port Memories. Theory and Practice*, PhD thesis, Delft Univ. of Technology, Delft, the Netherlands, 2001, <http://ce.et.tudelft.nl/~hamdioui/>
14. Hamdioui S., Gaydadjiev G. N., van de Goor Ad J. "A Fault Primitive Based Analysis of Dynamic Memory Faults," *Proc. PRORISC 2003*, Veldhoven, Nov. 2003, pp. 84-89.
15. Hamdioui S., Al-Ars Z., Gaydadjiev G., Reyes J. D. "Investigation of Single-Cell Dynamic Faults in Deep-Submicron Memory Technologies," *IEEE Proc. European Test Symposium Digest of Papers*, May 2006.