

ANALIZA COMPARATIVĂ A METODELOR DE TESTARE A DISPOZITIVELOR DE MEMORIE DIGITALĂ

Aurelia BALMUȘ, Serghei GRIȚCOV

Universitatea Tehnică a Moldovei

Abstract: În lucrarea dată sunt prezentate metodele de testare a dispozitivelor de memorie digitală. Este reprezentată analiza comparativă a metodelor clasice de testare, testelor march și testelor pseudo-inelare. Sunt evidențiate parametrii principali ale testelor: complexitatea algoritmică și hardware.

Cuvinte cheie: defecte, metoda exhaustivă, metoda pseudoexhaustivă, testele March, testarea pseudo-inelară.

1. Introducere

În orice dispozitiv de memorie pot apărea perturbări fizice în timpul producției a memoriei sau pe parcursul funcționării acestuia. Aceste perturbări fizice sunt numite defecte. Defectele duc la comportament incorect al dispozitivului de memorie. Pentru descrierea comportamentului incorect a fost introdusă definiția defectarea.

În scopul de a detecta defectările se folosesc diferite metode de verificare, numite – testare. Testarea, cu alte cuvinte – o modalitate de a verifica corectitudinea funcționării a memoriei aparatului. Metodele actuale de testare poate nu numai identifica prezența unei defectări, dar, de asemenea, pot localiza o defectare, iar în unele cazuri elimina o defectare. Vom lua în considerare metodele cunoscute de testare a dispozitivelor de memorie.

2. Metodele clasice de testare

Atunci când au apărut primele cipuri de memorie și au fost puse în aplicare a devenit necesitatea de a testa aceste chip-uri. Cea mai simplă metodă de testare constă în înregistrarea în memoria tuturor combinațiilor posibile ale statelor logice "0/1". Combinația care se înscrie în memorie în procesul de testare a fost numită **testul**. Apoi, datele celulelor de memorie sunt citite și analizate. Dacă valorile citite corespund celor înscrise, ajungem la concluzia că nu există nici o defectare. Această metodă de a încerca toate combinațiile posibile a fost numită metoda **exhaustivă** de testare [1].

Un parametru important al oricărei metode de testare este complexitatea algoritmică a testului. Se măsoară în numărul de operațiuni care urmează să fie efectuate în timpul testării a memoriei:

$$O = \alpha * n, \quad (1)$$

unde α – un număr întreg care reflectă numărul de operațiuni efectuate pe fiecare celulă de memorie (numărul de înscriere/citire), n - numărul de celule de memorie, O - complexitatea algoritmică.

Teste cunoscute în general, au trei tipuri de complexitate algoritmică: $O(n)$, $O(n^{3/2})$ și $O(>n^2)$. $O(n)$ - complexitatea algoritmică liniară. Două alte tipuri sunt neliniare [2]. La testarea exhaustivă numărul combinațiilor de înscriere sau citire este egal cu 2^n . Astfel, complexitatea algoritmică de testare completă este $2 * 2^n$. Testarea completă are o capacitate mare de detectare, dar complexitatea algoritmică este prea mare.

Pentru a reduce complexitatea algoritmică a fost oferită metoda **pseudoexhaustivă** [3]. Această metodă implică o căutare exhaustivă a tuturor combinațiilor posibile ale oricăror celule vecine de memorie k ale numărului total n de celule. Testul este reprezentat ca B (k, n). În tabelul 1 sunt prezentate câteva exemple de teste pseudoexhaustive.

Tabelul 1. Niște exemple de teste pseudoexhaustive

Test	$B(3,2)$	$B(4,2)$	$B(5,2)$
Valorile testului	0 0 0 0 1 1 1 0 1 1 1 0	0 0 0 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
Adresele celulelor de memorie	0 1 2	0 1 2 3	0 1 2 3 4

Metodele de testare clasice pot include, de asemenea, teste, cum ar fi GALPAT cu complexitatea algoritmică de n^2 , WALKING I/O cu complexitatea algoritmică $n^{3/2}$ descrise detaliat în [4]. Neajunsul acestor metode constă în complexitatea algoritmică înaltă. Ca o cale pentru rezolvarea acestei probleme sunt testele march.

3. Testele March

În 1982, M. Marinescu în Philadelphia la conferința internațională a prezentat metodele de testare care au fost foarte simple și complexitatea algoritmică a fost liniară [5]. Aceste metode sunt numite testele March.

Vom lua în considerare un simplu test march: MATS. Acest test este reprezentat după cum urmează: $\{\Downarrow(w0); \Downarrow(r0, w1); \Downarrow(r1)\}$. În testul, w și r denotă operație de înscriere și citire de date. Semnul înainte de paranteze indică direcția de operare: \Uparrow – cu adresarea ascendentă, \Downarrow – cu adresarea descendentă și \Downarrow – în ordine aleatorie. Testul MATS poate fi împărțit în trei iterații (fig. 1).

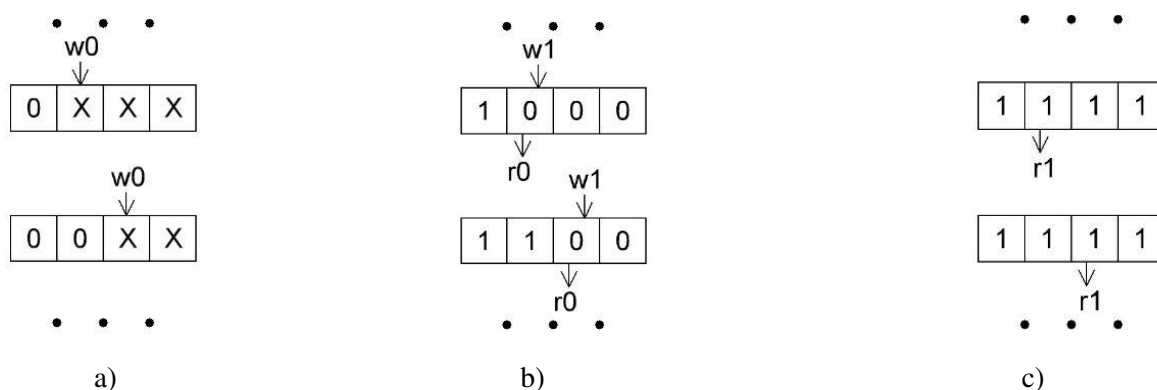


Figura 1 Iterații a testului MATS

În prima iterație (fig. 2, a) va fi înregistrat în toate celulele de memorie '0'. Pe cea de a doua iterație (fig. 2, b) se efectuează citirea "0" și se efectuează înscrierea "1". La a treia iterație (fig. 2, c) se efectuează citirea "1". Simplitatea testului este că, dacă valoarea citită nu este egală cu cea înscrisă (de exemplu, în a treia iterație este citit "0"), atunci această celulă este defectată. Astfel, testul march poate nu numai să detecteze o defectare, dar, de asemenea, poate localiza această defectare. Complexitatea algoritmică de un astfel test este $4n$.

După prezentarea ideii generale de testare march a fost elaborată o mulțime de teste march [6]. Defectările detectate de testele march cele mai cunoscute sunt reprezentate în tab. 2 [7].

Tabelul 2. Cele mai frecvente defectări detectate de testele march

Teste	Defectările					
	AF	SAF	TF	CFin	CFid	LF
MATS+	+	+				
MATS++	+	+	+			
March X	+	+	+	+		
March C-	+	+	+	+	+	
March A	+	+	+	+	+	+

O altă caracteristică a testelor march este elaborarea testelor non-distructive [2]. Este important de a restabili datele după testare, care au fost păstrate până la procesul testării. Acest lucru este posibil în două moduri: de copiat datele în memoria tampon pe timpul testării, sau de aplicat testarea non-distructivă. Prima metodă nu provoacă dificultăți în elaborarea testului, este necesar de a adăuga doar copierea datelor într-un tampon înainte de testare, și de a adăuga copierea datelor din tampon după testare. Cu toate acestea, prima metodă necesită un dispozitiv de memorie suplimentar. Nikolaidis a propus să transforme testele march clasice în cele non-distructive [8]. Aceste transformări duc la o creștere în complexitate algoritmică doar aproximativ 1,5 ori, permițând recuperarea datelor după procesul de testare, și să nu fie aplicat un cip de memorie suplimentară. **Neajunsul** acestor metode constă în complexitatea hardware înaltă. Ca o cale pentru rezolvarea acestei probleme sunt testele pseudo-inelare.

4. Testarea pseudo-inelară

Testarea pseudo-inelară sau π -testarea se bazează pe emularea registrului de deplasare cu legături inverse (LFSR) a memoriei. Memoria testată este reprezentată ca un masiv unidimensional, care conține m -celulele poziționale. Celulele se indexează cu valori ale adreselor de la 0 până la $n-1$, unde n – volumul matricei memoriei. Ideea testării π se bazează pe utilizarea celulelor k ale memoriei în calitate de descărcări (a celulelor) ale registrului de deplasare și migrarea (virtuală) registrului prin toate celulele matricei memoriei. În cazul testării π se realizează migrarea LFSR virtual în raport cu datele, dar nicidecum mișcarea datelor în celulele registrului.

Procedura testării π urmărește îndeplinirea așa-numitor π -iterații. Iterația testării π pentru MSP430 constă din următoarele: *inițializarea* LFSR virtual, *migrarea* LFSR în spațiul memoriei microcontrolerului, *calculul* stării finale **Fin** a registrului virtual LFSR și *analiza* rezultatului obținut. Calitatea testării π se evaluează pe calea comparării stării finale **Fin** cu cea de referință [9]. Pentru evaluarea stării de referință folosim ecuația recurentă cunoscută Kolmogorov-Chapmen:

$$S_n = S_0 * A^n, \quad (1)$$

unde S_0 – vectorul unidimensional al stării LFSR cu valoarea k în momentul inițial de timp; A – matricea de însoțire bidimensională a formatului:

$$A = \begin{bmatrix} g_1 & 1 & 0 & \dots & 0 \\ g_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \\ g_{k-1} & 0 & 0 & \dots & 1 \\ g_k & 0 & 0 & \dots & 0 \end{bmatrix} \quad (2)$$

unde g – coeficientul polinomului $g(z)$ de gradul k pe $\mathbf{GF}(2^m)$, prezentat în format standart, i.e. al cărui coeficient liber g_0 este egal cu zero.

În cazul când numărul operațiilor de deplasare este proporțional perioadei T a polinomului $g(z)$, atunci ecuația va arăta astfel **Init** \approx **Fin**. Dacă starea finală **Fin** va coincide cu cea inițială **Init**, putem concluziona că nu au fost găsite defecte la tipul specificat [10].

Ca o **concluzie**, se poate observa că testele pseudo-inelare elaborate sunt nu mai rele decât cele march la complexitatea algoritmică la testarea memoriei cu celule non-binare (4, 8-bit ...), detectează pe deplin defectările cele mai frecvente: constante, cuplate și de tip coduri sensibile, solicitând în același timp în mod semnificativ mai puține resurse hardware pentru realizare decât testele march.

Concluzie

În rezultatul analizei metodelor de testare a dispozitivelor de memorie digitală vedem, că cele mai importante parametri a testelor este complexitatea algoritmică și hardware. Metodele clasice au complexitatea algoritmică neliniară, ce duce la un timp de testare prea mare, dar permit o calitate înaltă de detectare a defectărilor. Metodele march și pseudo-inelare au complexitatea algoritmică liniară și detectează toate defectările tipice (constante, cuplate). Testarea pseudo-inelară are complexitatea hardware mai scăzută decât testele march, dar complexitatea algoritmică a ei este mai înaltă.

Bibliografie

1. Tang D.T., Woo Lin S. Exhaustive Test Pattern Generation with Constant Weight Vectors. IEEE Transactions on Computers, 1983, vol. C-32, p. 1145-1150.
2. Ярмолик С., Занкович А., Иванюк А. Маршевые тесты для самотестирования ОЗУ. Минск: БГУ, 2009, 270 с.
3. Karpovsky M.G., Van de Goor A.J., Yarmolik V.N. Pseudo-exhaustive word-oriented DRAM testing. ED&TC 1995, Paris, p. 126-132.
4. Schat J. Fault Clustering in deep-submicron CMOS Processes. Design, Automation and Test in Europe, 2008, p. 511-514.
5. Marinescu M. Simple and Efficient Algorithms for Functional RAM Testing. Proceedings International Test Conference, Philadelphia, 1982, p. 236-239.
6. Van de Goor A., Gaydadjiev G., Hamdioui S. Memory testing with a RISC microcontroller. Design, Automation & Test in Europe Conference & Exhibition, 2010, p. 214-219.
7. Amit Dutta, Srinivasulu Alampally, Arun Kumar, Rubin A. Parekhji. A BIST Implementation Framework for Supporting Field Testability and Configurability in an Automotive SOC. Edinburgh, DSN-2007.
8. Nicolaidis M. Theory of transparent BIST for RAMs. IEEE Transactions on Computers, 1996, vol. 45, p. 1141-1156.
9. Gh. Bodean, D. Bodean, A. Labunetz, New Schemes for Self-Testing RAM, Technical University of Moldova, 2005.
10. S. Grițcov, A. Ghincul, Gh. Bodean "AUTOTESTAREA PSEUDOINELARĂ A MICROCONTROLLERELOR NANOSATELITULUI SATUM," ICTEI-2012, vol. 2, 2012, pp. 260-267.