(19) OFICIUL DE STAT
PENTRU INVENŢII ŞI MĂRCI
Bucureşti

ROMÂNIA

(11) **RO 126730 A2**
(51) **Int.Cl.**
H03K 19/00 (2006.01);
G06F 7/58 (2006.01)

(12) # CERERE DE BREVET DE INVENŢIE

(21) Nr. cerere: **a 2009 01002**

(22) Data de depozit: **30.11.2009**

(41) Data publicării cererii:
30.09.2011 BOPI nr. **9/2011**

(71) Solicitant:
• **UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**,
STR.CONSTANTIN DAICOVICIU NR.15, CLUJ-NAPOCA, CJ, RO

(72) Inventatori:
• **SUCIU ALIN DUMITRU**,
STR. GRIGORE ALEXANDRESCU NR.8, AP.20, CLUJ NAPOCA, CJ, RO;
• **CREŢ OCTAVIAN AUGUSTIN**,
GRIGORE ALEXANDRESCU NR.42, AP.17, CLUJ NAPOCA, CJ, RO;
• **GYORFI TĂMAS**,
STR. KOLCSEY FERENCZ NR.31, MARGHITA, BH, RO

(74) Mandatar:
CABINET DE PROPRIETATE INDUSTRIALĂ CIUPAN CORNEL,
STR. MESTECENILOR NR. 6, BL. 9E, AP. 2, CLUJ NAPOCA, JUDEŢUL CLUJ

(54) # METODĂ DE IMPLEMENTARE A GENERATOARELOR DE NUMERE REAL-ALEATOARE ÎN DISPOZITIVE FPGA

(57) **Rezumat:**

Invenţia se referă la o metodă de implementare a generatoarelor de numere real-aleatoare în dispozitive FPGA (Field Programmable Gate Array). Metoda propusă exploatează o caracteristică a circuitelor electronice digitale numită "fanout", pentru a obţine un comportament nedeterminist sau, altfel spus, o sursă de entropie. Pentru aplicarea metodei conform invenţiei este folosit un generator alcătuit din următoarele componente: un bloc driver (**1**) care generează un semnal periodic şi comandă circuitele unui bloc sarcină (**2**), circuite care sunt în număr egal cu fanout-ul blocului driver (**1**), şi un bloc post-procesor (**3**) ce corectează posibilele dezechilibre ale fluxului de biţi. Metoda conform invenţiei constă, într-o primă etapă, din generarea, de către blocul driver, a unui semnal periodic a cărui frecvenţă este egală cu frecvenţa semnalului de tact, urmată, într-o a doua etapă, de eşantionarea cu aceeaşi frecvenţă a semnalului de ieşire al blocului driver, eşantionare realizată de circuitele blocului sarcină, rezultatele eşantionării fiind combinate, într-o a treia etapă, prin aplicarea unei operaţii SAU-EXCLUSIV pe un anumit număr de biţi consecutivi, în cadrul blocului post-procesor.

Revendicări: 1
Figuri: 2

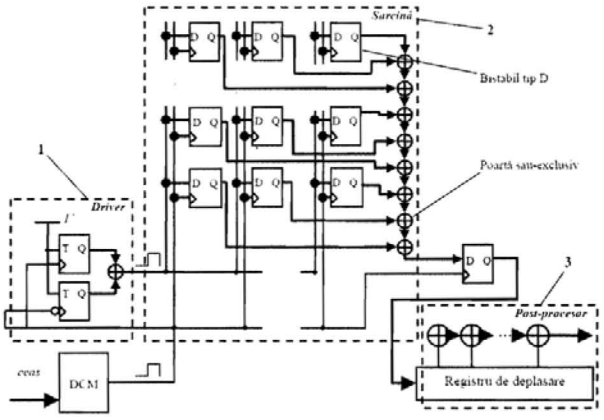

Fig. 2

RO 126730 A2

OFICIUL DE STAT PENTRU INVENŢII ŞI MĂRCI
Cerere de brevet de invenţie
Nr. a 2009 01002
Data depozit 30-11-2009

METODĂ DE IMPLEMENTARE A GENERATOARELOR DE NUMERE REAL-ALEATOARE ÎN DISPOZITIVE FPGA

Invenţia se referă la o metodă de implementare a generatoarelor de numere *real-aleatoare* în dispozitive FPGA (Field Programmable Gate Array).

Generatorele de numere real-aleatoare se bazează întotdeauna pe un proces fizic nedeterminist, numit şi *sursă de entropie*, ceea ce conduce la obţinerea unor numere complet impredictibile, real-aleatoare.

Din acest motiv, generatoarele de numere real-aleatoare sunt superioare în anumite tipuri de aplicaţii (ex: securitate, criptografie, jocuri) faţă de generatoarele de numere *pseudo-aleatoare* (generatoare deterministe, bazate pe formule matematice şi algoritmi care produc secvenţe de numere complet predictibile).

Generatoarele de numere real-aleatoare sunt utilizate cu precădere în domeniile:

- criptografie digitală
- criptografie cuantică
- protocoale de securitate
- jocuri de noroc, loterie

Generatoarele de numere real-aleatoare sunt în general dispozitive sau ansambluri de dispozitive fizice, de natură mecanică, electrică, electronică, etc. La ora actuală, cel mai des sunt folosite în practică generatoarele de numere real-aleatoare care se bazează pe dispozitive şi circuite electronice, analogice şi/sau digitale.

Arhitectura generală a generatoarelor de numere real-aleatoare bazate pe circuite electronice are următoarele componente:

1. *sursa de entropie* – generează un semnal care este rezultatul unui fenomen fizic nedeterminist;
2. *digitizor* – eşantionează periodic semnalul generat de sursa de entropie, convertind acest semnal într-o secvenţă de biţi aleatori;
3. *post-procesor* – ajustează distribuţia de probabilitate a ieşirii, compensând imperfecţiunile statistice din interiorul sursei de entropie sau din digitizor.

Pentru asigurarea securităţii fizice, este foarte util ca generatorul de numere real-aleatoare să fie compact, implementat într-un singur dispozitiv electronic care să conţină toate cele trei componente esenţiale: sursa de entropie, digitizor şi post-procesor. În plus, ideal este ca sursa de entropie să fie tot un sistem digital şi nu unul mixt analog-digital.

Dispozitivele electronice FPGA (Field Programmable Gate Array), care sunt reţele de porţi logice reconfigurabile, prezintă ambele avantaje menţionate anterior, la care se adaugă avantajul flexibilităţii în proiectare şi a unui *time-to-market* mult redus.

Metoda de proiectare preferată până la ora actuală pentru generatoarele real-aleatoare în dispozitive FPGA, atestată prin lucrări ştiinţifice şi rapoarte tehnice publicate în literatura de specialitate ([FD02],[KG04],[Golić04]), constă în crearea unei oscilaţii cu *jitter*. Această oscilaţie se obţine prin utilizarea unor oscilatoare inelare (*ring oscillators*) constând într-un număr impar de inversoare conectate într-o topologie de tip inel.

De exemplu, design-ul folosit de Kohlbrenner şi Gaj [KG04] utilizează două oscilatoare inelare care se eşantionează reciproc, însă plasarea acestor oscilatoare în CLB-urile dispozitivului FPGA este delicată, deoarece frecvenţele de oscilaţie trebuie să fie foarte apropiate.

Prin urmare, obţinerea unui design funcţional în parametri acceptabili necesită o plasare manuală a oscilatoarelor, care este neportabilă (necesită o replasare manuală în fiecare familie de dispozitive FPGA). Chiar şi prin plasare manuală, calitatea numerelor aleatoare

generate este slabă. Pentru creşterea calităţii este necesară folosirea unui bloc de post-procesare de mari dimensiuni, ceea ce duce la scăderea semnificativă a debitului.

Golić a propus în [Golić04] două variante promiţătoare de oscilatoare inelare clasice folosind concepte înrudite cu registrele de deplasare cu reacţie lineară (LFSR – Linear Feedback Shift Register): oscilatoare inelare Fibonacci şi Galois. Aceste circuite asincrone cu buclă de reacţie (*feedback*) combină aleatorismul real (dat de fenomenul de *jitter* din cadrul unor oscilatoare inelare) şi pseudo-aleatorismul (dat de registrul LFSR).

Design-ul a fost implementat experimental în dispozitive FPGA, dar nu s-au oferit detalii precise; s-a spus doar că implementarea satisface testele statistice standard, dacă se aplică o post-procesare foarte puternică (un LFSR pe 64 de biţi). Principalul dezavantaj al acestui tip de generator este acela că nu este un generator real aleator, ci un generator hibrid care combină elemente ale generatoarelor real-aleatoare cu cele ale generatoarelor pseudo-aleatoare.

Parametrii de design ai majorităţii propunerilor publicate până la ora actuală (cum ar fi: frecvenţa de eşantionare, volumul post-procesării, şi astfel debitul de biţi aleatori care rezultă) sunt determinaţi în principal printr-un proces iterativ (*trial-and-error*), până când şirul de biţi produs trece cu succes testele statistice existente la dispoziţia comunităţii ştiinţifice internaţionale: NIST [RSN+01], TestU01 [L'ES07] sau DIEHARD [Marsaglia96].

Problema tehnică pe care o rezolvă invenţia este implementarea unui generator de numere real-aleatoare în dispozitive FPGA având un *debit ridicat* şi o *sursă de entropie solidă* (bazată pe un fenomen fizic de natură electronică, care nu este sensibil la plasarea componentelor în interiorul dispozitivului FPGA). De asemenea, invenţia are ca scop implementarea unui generator *portabil* care să funcţioneze pe cât mai multe familii de dispozitive FPGA, şi deci să aibă o cât mai mare aplicabilitate practică.

Metoda propusă exploatează o caracteristică a circuitelor electronice digitale denumită "*fanout*", pentru a obţine un comportament nedeterminist, sau altfel spus o sursă de entropie.

Fanout-ul unei componente logice este definit ca numărul de dispozitive logice ce pot fi conectate la ieşirea sa fără a depăşi limitele de curent ale componentei logice. Pentru a menţine niveluri logice corecte, ieşirea unei componente logice nu poate comanda un număr infinit de alte dispozitive logice, deoarece dispozitivele logice reale au capacităţi şi rezistenţe de intrare.

Prin urmare, dacă ieşirea unei componente logice comandă multe alte dispozitive logice, această componentă logică va avea de „înfruntat" o sarcină capacitivă ridicată, ceea ce încetineşte tranziţia ieşirii, mărind astfel întârzierea de propagare a semnalului.

În figura 1 se ilustrează timpul de tranziţie al unui semnal digital pentru diferite valori ale sarcinii capacitive care este direct proporţională cu *fanout*-ul. Linia punctată indică momentul eşantionării care în exemplul de faţă generează un rezultat nedeterminist pentru o sarcină capacitivă de 6.5 nF.

Datorită faptului că întârzierea de propagare a semnalului creşte, frecvenţa operaţională maximă a sistemului scade. Prin urmare dacă frecvenţa depăşeşte o anumită valoare limită care este dată de valoarea *fanout-ului*, atunci apare un comportament nedeterminist.

Generatorul, conform invenţiei, are următoarele componente (figura 2):

1. blocul *Driver* – generează un semnal periodic şi comandă circuitele din componenta *Sarcină*;
2. blocul *Sarcină* – circuite care eşantionează semnalul generat de componenta *Driver*;
3. blocul *Post-procesor* – circuit care corectează posibilele dezechilibre ale fluxului de biţi.

Fanout-ul blocului *Driver* este egal cu numărul de circuite din blocul *Sarcină*. Semnalul generat de blocul *Driver* este eşantionat în intervalul de timp în care are loc tranziţia semnalului.

Soliditatea sursei de entropie este asigurată prin faptul că numărul de circuite din blocul *Sarcină* este de ordinul miilor. Astfel, intervalul în care semnalul generat de blocul *Driver* are o valoare nedeterministă este lărgit. În plus, acest semnal va fi eşantionat de fiecare circuit din blocul *Sarcină*. Rezultatele eşantionării sunt combinate prin aplicarea operatorului SAU-EXCLUSIV (XOR) în cadrul blocului *Post-procesor*.

În figura 2 este ilustrată implementarea generatorului de numere real-aleatoare propus. Blocul *Driver* generează un semnal periodic, a cărui frecvenţă este egală cu frecvenţa semnalului de tact. Circuitele din blocul *Sarcină* sunt bistabile de tip D, care vor eşantiona cu aceeaşi frecvenţă semnalul de ieşire al blocului *Driver*.

Implementarea propusă are mai mulţi parametrii, cei principali fiind frecvenţa semnalului de tact şi numărul de bistabile din componenta *Sarcină*. În exemplul de realizare a generatorului, semnalul de tact este generat de un bloc DCM (Digital Clock Manager), având frecvenţa de 300 MHz.

Numărul de bistabile din blocul *Sarcină* este 1536 pentru acest exemplu de realizare. Majoritatea bistabilelor vor eşantiona semnalul generat de blocul *Driver* în momentul tranziţiei sale. Rezultatele eşantionării sunt combinate prin aplicarea operaţiei SAU-EXCLUSIV. Blocul *Post-procesor* aplică operaţia SAU-EXCLUSIV pe cinci biţi consecutivi pentru creşterea entropiei şi pentru eliminarea posibilelor dezechilibre ale fluxului de biţi extraşi din sursa de entropie.

Debitul generatorului astfel obţinut este de 60 Mbps, iar şirul de biţi produs trece cu succes testele statistice NIST, TestU01 şi DIEHARD. Această configuraţie a fost realizată practic, testată exhaustiv şi este funcţională pe următoarele familii de dispozitive FPGA: Xilinx Virtex II-Pro, Virtex 4, Virtex 5 şi Spartan-3E.

Prin aplicarea invenţiei, se obţine un generator de numere real-aleatoare care prezintă următoarele avantaje:

1. debit ridicat – se constată creşterea semnificativă a debitului, comparativ cu alte medode bazate pe dispozitive FPGA ([FD02],[KG04],[Golić04]);
2. calitate ridicată – generatorul furnizează secvenţe de numere aleatoare de înaltă calitate, care trec cu succes testele statistice existente: NIST [RSN+01], TestU01 [L'ES07] sau DIEHARD [Marsaglia96];
3. portabilitate – aceeaşi configuraţie a generatorului funcţionează pe mai multe familii de dispozitive FPGA;
4. simplitate – generatorul este simplu şi uşor de implementat/configurat;
5. insensibilitate la plasare – generatorul nu este sensibil la plasarea componentelor în interiorul dispozitivului FPGA.

Bibliografie:

[BL05] M. Bucci and R. Luzzi. Design of Testable Random Bit Generators. In *Criptografic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, Scotland, August 29 - September 1, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer, 2005, pp. 147-156.

[Brouwer06] A. E. Brouwer. Server for bounds on the minimum distance of q-ary linear codes. [Online]. Available: http://www.win.tue.nl/~aeb/.

[BST03] B. Barak, R. Shaltiel, and E. Tromer. True Random Number Generators Secure in a Changing Environment. In *Criptografic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003,*

a-2009-01002--
30 -11- 2009

Revendicare

1. Metodă pentru implementarea a generatoarelor de numere real-aleatoare într-un singur dispozitiv FPGA, **caracterizată prin aceea că** sursa de entropie o constituie eşantionarea unui semnal digital periodic generat de un bloc *Driver* (1) având un *fanout* ridicat, ceea ce determină un timp de tranziţie al semnalului mărit, starea logică a semnalului fiind nedeterministă pe perioada tranziţiei, toate circuitele din blocul *Sarcină* (2) contribuind la determinarea valorii parametrului *fanout* şi eşantionând semnalul în acelaşi timp, rezultatele eşantionării fiind combinate prin aplicarea operaţiei SAU-EXCLUSIV într-un bloc *Post-procesor* (3).

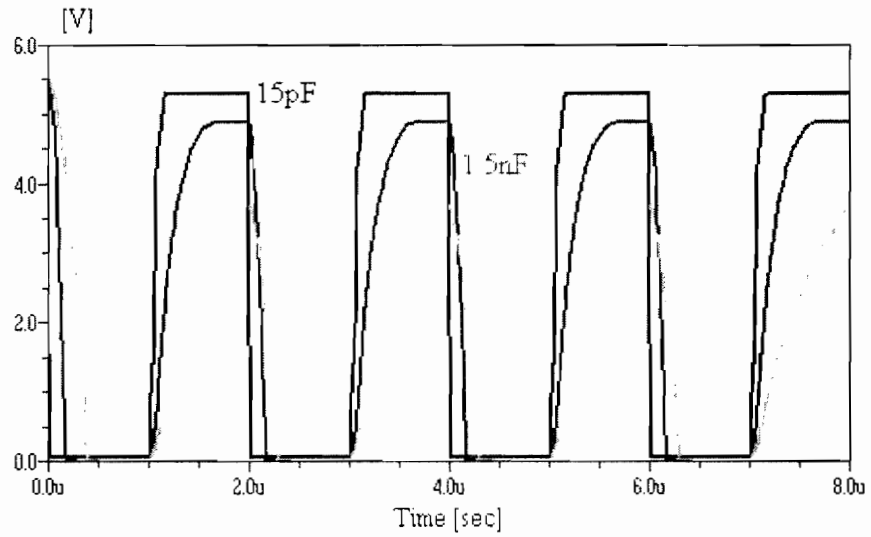

Figura 1

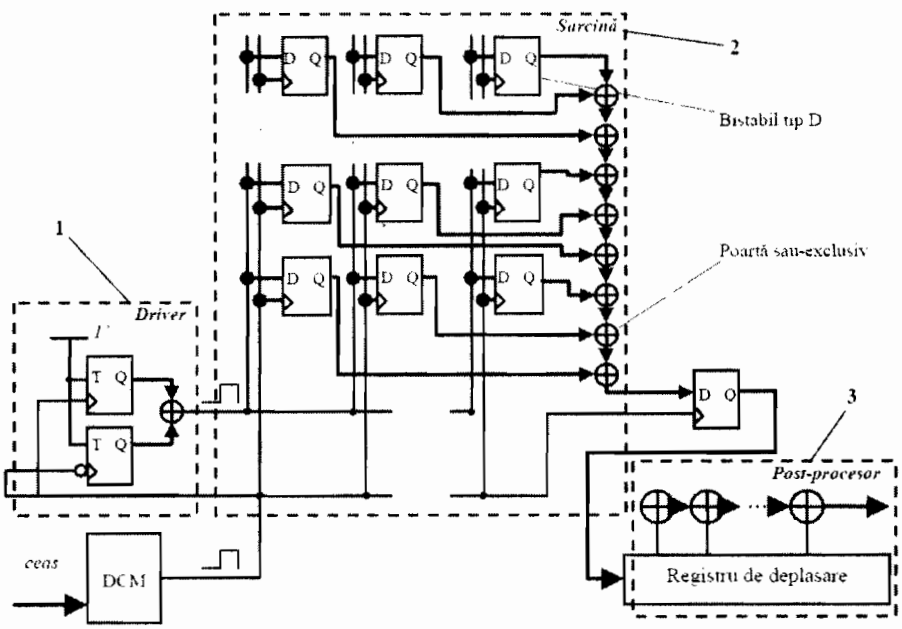

Figura 2