



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2014 00283

(22) Data de depozit: 10/04/2014

(41) Data publicării cererii:
27/11/2015 BOPI nr. 11/2015

(71) Solicitant:
• IXIA, A CALIFORNIA CORPORATION,
26601 WEST AGOURA ROAD,
CALABASAS, CA, US

(72) Inventatori:
• ȘINCAN SERGIU, ALEEA ISTRU 2C,
BL. A14B, INTRAREA 1, ET. 3, AP. 12,
SECTOR 6, BUCUREȘTI, B, RO;

• MESSMER NEIL, 8344 211 TH STREET,
LANGLEY, CA

(74) Mandatar:
RATZA ȘI RATZA SRL, B-DUL A.I. CUZA,
NR. 52-54, SECTOR 1, BUCUREȘTI

(54) METODĂ ȘI SISTEM PENTRU IMPLEMENTARE HARDWARE
A PACHETELOR DE DATE UNIFORM AMESTECATE

(57) Rezumat:

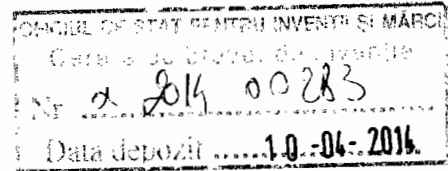
Invenția se referă la o metodă și la un sistem pentru producerea pachetelor de date uniforme, aleator amestecate, pentru transmisia printr-un sistem de generare de pachete de testare. Metoda conform invenției constă în: furnizarea, folosind hardware pentru generarea de secvențe pseudo-aleatoare, a unei multitudini de seturi de numere, în care fiecare set cuprinde o secvență pseudo-aleatoare, în care fiecare set reprezintă un subset al tuturor valorilor posibile ale unei secvențe amestecate la întâmplare, și în care toate seturile combinate reprezintă toate valorile secvenței, și producerea secvenței prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator, selectat ca următoarea valoare în secvență. Sistemul conform invenției cuprinde: o multitudine de module, în care fiecare

modul generează propria secvență pseudo-aleatoare de numere, și în care fiecare modul emite următoarea valoare în secvența sa pseudo-aleatoare, ca răspuns la o cerere de ieșire, și o logică de selecție, pentru a crea o secvență de valori de ieșire prin selectarea repetată a unuia dintre module, conform unei funcții de selectare aleatoare, și prin transmiterea unei cereri de ieșire la modulul selectat, în care secvența de valori creată din ieșirea modulelor selectate aleator cuprinde o secvență uniformă, amestecată la întâmplare.

Revendicări: 22

Figuri: 6





DESCRIERE

METODĂ ȘI SISTEM DE IMPLEMENTARE HARDWARE A PACHETELOR DE DATE UNIFORME, ALEATOR AMESTECATE

DOMENIUL TEHNIC

Prezenta invenție se referă la metode și sisteme pentru producerea pachetelor de date uniforme aleator amestecate pentru transmisia printr-un sistem de generare de pachete de testare. Mai precis, obiectul descris aici se referă la metode și sisteme pentru o implementare hardware a pachetelor de date uniforme, aleator amestecate.

STADIUL ANTERIOR AL TEHNICII

În rețele de pachete de date, nu este garantat că pachetele ajung în aceeași ordine în care au fost trimise, și astfel așa anumitele noduri din rețeaua de pachete de date trebuie să fie capabile să proceseze și să reasambleze pachetele, indiferent de ordinea primirii. Pachetele de date individuale pot fi de asemenea, fragmentate și un nod de rețea de pachete trebuie să poată reasambla corect fragmentele de pachete, indiferent de ordinea în care acestea sunt recepționate. Pentru a testa această capacitate, este de dorit să se furnizeze pachete sau fragmente de pachete de date amestecate aleator de la un nod de testat, pentru a determina cât de bine reasamblează nodul fragmentele.

Problema care trebuie rezolvată este cum să se genereze permutări aleatoare imparțiale a unui număr arbitrar de indici în sistemul hardware, la o viteză de cel puțin un index pentru fiecare cadru, care, presupunând o dimensiune minimă a cadrului de 64 de bytes, 8 bytes preambul și 12 bytes diferență, traduce pentru un index per 16.8 ns la rețele de 40 Gbps și unul la fiecare 6.72 ns la rețele de 100 Gbps. Indicii astfel produși sunt utilizați pentru a construi și a emite fragmente de pachete de date în ordine întâmplătoare. Numărul de fragmente pot varia de la 2 la 144 într-o aplicație tipică.

Acești indici trebuie: (1) să urmeze o distribuție uniformă, și (2), să asigure o acoperire rezonabilă. Pentru că există căile $A_{144}^{144} = 144! \approx 5.55 \cdot 10^{24} \approx 0.775 \cdot 2^{830}$ de reordonarea a 144 fragmente, acoperirea completă ar avea nevoie de un spațiu necesar de 830 de biți. Sistemul hardware având cai de date largi de 830 biți care pot funcționa la frecvența necesară ar fi greoi și scump. Sistemul hardware având o cale de date de 32 de biți este mult mai comun, dar pentru o dimensiune de spațiu necesar de 32 de biți, cel mai mare număr posibil de fragmente pentru care acoperirea este completă este $F = 12$ fragmente, care este mult mai mic decât necesarul de 144.

În consecință, datorită acestor dezavantaje asociat cu abordările convenționale pentru a oferi permutări aleatoare imparțiale la o acoperire completă, există o nevoie de metode de analiză cadre și sisteme de generare hardware de secvențe care sunt statistic similare pentru un pachet de date probă uniform și aleator, a tuturor secvențelor posibile. Mai precis, există o nevoie pentru o implementare hardware a pachetelor de date uniforme aleator amestecate.

EXPUNEREA PE SCURT A INVENȚIEI

Conform unui aspect al invenției, obiectul prezentei invenții include un sistem de implementare hardware a pachetelor de date uniforme, aleator amestecate. Sistemul include generatoare (PRBS) de secvențe multiple pseudo-aleatoare de biți, unde fiecare generator PRBS furnizează o secvență pseudo-aleatoare de numere S, în care fiecare modul emite următoarea valoare în secvența pseudo-aleatoare de biți, ca răspuns la recepționarea unei cereri de ieșire. Sistemul include și logică de selecție pentru a crea o secvență O de valori de ieșire prin selectarea repetată a unuia din multitudinea de module, în conformitate cu o funcție de selecție aleatoare și prin transmiterea unei cereri de ieșire la modulul selectat, în care secvența O de valori creată din ieșirea modulelelor selectate aleator cuprinde o secvență uniformă, amestecată la întâmplare. Probabilitatea ca un generator PRBS să fie selectat este ponderată în funcție de numărul N de valori pseudo-aleatoare care nu au fost încă scoase din L valori posibile în secvență.

Așa cum este folosit în prezenta documentație, termenul "secvență PRBS" se referă la întregul set de valori pseudoaleatorii produs de un anumit generator PRBS.

Așa cum sunt utilizați în prezenta documentație, termenii "secvență aleatoare", "secvență de ieșire", și "valori de ieșire" sunt sinonime care se referă la secvența uniformă, amestecată aleator, a valorilor obținute în mod colectiv de către unul sau mai multe generatoare de secvență pseudo-aleatoare de biți care funcționează împreună, în conformitate cu principiile descrise în prezenta.

Așa cum sunt folosiți aici, termenii "secvență de amestecare completă" și "secvență de ieșire completă" și sunt sinonime care se referă la o ieșire având valori care reprezintă secvențele complete PRBS de la toate generatoarele PRBS active. Atunci când o secvență completă aleatoare a fost finalizată, toate valorile posibile din gama destinată valorilor de ieșire au fost produse în ordine aleatoare. Producerea unei secvențe de amestecare completă poate fi, de asemenea, menționată ca "o rulare".

Conform unui alt aspect, prezenta invenție se referă la o metodă de implementare hardware a pachetelor de date uniforme, amestecate aleator. Metoda include utilizarea unui hardware

025

pentru generarea secvențelor pseudo-aleatoare pentru a asigura o multitudine de seturi de numere, fiecare set fiind o secvență pseudo-aleatoare și care reprezintă un subset de toate valorile posibile ale unei secvențe O amestecate la întâmplare, în care toate seturile combinate reprezintă toate valorile secvenței O. Secvența O este produsă prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator selectat ca și următoarea valoare în secvența O.

Conform unui alt aspect, prezenta invenție include un sistem pentru efectuarea testelor de depreciere a unui dispozitiv de rețea de pachete de date. Sistemul include un modul de amestecare pentru amestecarea aleatoare a unui set de numere pentru a crea o secvență O de amestecare, în care modulul de amestecare include mai multe module care generează fiecare propria sa secvență pseudo-aleatoare de numere și care produce următoarea valoare în secvența sa pseudo-aleatoare ca răspuns la recepționarea unei cereri de ieșire. Modulul de amestecare include de asemenea logică de selecție pentru crearea secvenței O de amestecare prin selectarea repetată la întâmplare a unuia din multiplele module și prin transmiterea unei cereri de ieșire la modulul selectat. Secvența O de amestecare creată de ieșirea modulelor selectate aleator este o secvență uniformă, amestecată la întâmplare. Sistemul include de asemenea un modul de reordonare pachete de date pentru fragmentarea unui prim pachet de rețea în interiorul unui set de fragmente secvențial comandate, pentru amestecarea ordinii fragmentelor conform unei secvențe O de amestecare, și pentru transmiterea fragmentelor amestecate la un modul de ieșire pentru transmisia către un dispozitiv de testat.

Conform unui alt aspect, prezenta invenție include o metodă pentru efectuarea testelor de deprecierea unui dispozitiv de rețea de pachete de date. Metoda include utilizarea unui hardware pentru a genera mai multe seturi de secvențe pseudo-aleatoare și crearea unei secvențe O de ieșire amestecată aleator prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator selectat ca următoarea valoare în secvența O până când toate valorile de secvență O au fost produse; folosirea secvenței O produsă pentru a amesteca fragmente dintr-un prim pachet de rețea pentru a crea un al doilea pachet de rețea care are fragmente amestecate; și transmiterea celui de-al doilea pachet de rețea la un dispozitiv de testat.

Obiectele invenției pot fi implementate în software-ul în combinație cu hardware și/sau firmware. De exemplu, obiectele prezentei invenții pot fi implementate în software-ul executat de către un procesor.

Într-un exemplu de implementare, obiectele prezentei invenții pot fi implementate folosind un suport non-tranzitoriu care poate fi citit de calculator care are stocate instrucțiuni executabile

de calculator care, atunci când sunt executate de către procesorul unui calculator, comandă acestuia să efectueze pașii. Suportul citibil de calculator adecvat pentru punerea în aplicare a prezentei invenții include suportul non-tranzitoriu care poate fi citit de calculator, cum ar fi dispozitive cu memorie disc, dispozitive cu memorie cip, dispozitive logic programabile și circuite integrate specifice aplicației. În plus, un suport care poate fi citit de calculator care implementează prezenta invenție poate fi amplasat pe un singur dispozitiv sau o singură platformă de calcul sau poate fi distribuit pe mai multe dispozitive sau platforme de calcul.

DESCRIEREA PE SCURT A DESENELOR

Exemplele de realizare preferate ale prezentei invenții vor fi explicate în continuare cu referire la desenele însoțitoare, în care numere de referință similare reprezintă elemente similare, în care:

Figura 1 este o schemă bloc care ilustrează un sistem de implementare hardware a pachetelor de date uniforme, amestecate aleator, conform unui exemplu de realizare a prezentei invenții;

Figura 2 este o schemă bloc care ilustrează în detaliu un circuit de selecție aleatoare ponderată, conform unui exemplu de realizare a prezentei invenții;

Figura 3 este o schemă bloc care prezintă în detaliu un modul de generarea unei secvențe pseudo-aleatoare de biți, conform unui exemplu de realizare a prezentei invenții;

Figura 4 este o organigramă care ilustrează un exemplu de procedeu de implementare hardware a pachetelor de date uniforme, amestecate aleator, conform unui exemplu de realizare a prezentei invenții;

Figura 5 este o schemă bloc care prezintă un sistem pentru efectuarea testelor de deprecierea unui dispozitiv de rețea de pachete de date, conform unui exemplu de realizare a prezentei invenții; și

Figura 6 este o organigramă care ilustrează un exemplu de procedeu pentru efectuarea testelor de deprecierea unui dispozitiv de rețea de pachete de date, conform unui exemplu de realizare a prezentei invenții;

DESCRIEREA DETALIATĂ

Conform prezentei invenții, se descriu sisteme, metode, și suport citibil de calculator pentru o implementare hardware a pachetelor de date uniforme, amestecate aleator. Exemplele de realizare ale prezentei invenții se vor descrie în continuare în detaliu, în legătură cu figurile explicative. Ori de câte ori este posibil, vor fi utilizate aceleași numere de referință din figuri, pentru face referire la aceleași repere sau la părți similare.

Pentru a genera secvențe care sunt statistic similare la un eșantion uniform aleator din toate secvențele posibile, metodele și sistemele descrise intercalează, printr-o selecție aleatoare corect influențată, permutări mai mici ale reprezentării binare a numărului de indici. Această putere de două permutări se pretează bine la punerea în aplicare a hardware-ului. Conform unui exemplu de realizare, permutări pseudoaleatoare de 2^i sunt intercalate pentru a obține permutări imparțiale de indici de mărime arbitrară. Într-un exemplu de realizare, permutările astfel produse sunt procesate în continuare pentru a îmbunătăți acoperirea permutărilor celor 2^i fără a fi nevoie de funcții multiple de feedback ale registrului de deplasare.

Figura 1 este o schemă bloc care ilustrează un sistem de implementare hardware a pachetelor de date uniforme, amestecate aleator, conform unui exemplu de realizare a prezentei invenții. În varianta prezentată în Figura 1, sistemul **100** include un modul de selectare aleatoare ponderată **102**, care selectează unul dintr-o multitudine de module **104**, fiecare pentru generarea unui permutări pseudoaleatoare de valorilor 2^i .

Fiecare modul **104** include un generator de secvență pseudo-aleatoare de biți ($PRBS_N$) **106** și un registru mască ($MASK_N$) **108**, care modifică ieșirea unui generator PRBS. În exemplul de realizare ilustrat în Figura 1, de exemplu, ieșirea unui generator PRBS **106** și a unui MASK **108** sunt logic SAU, împreună utilizând o funcție SAU **110**. În varianta prezentată în Figura 1, fiecare modul **104** include o intrare de validare **E**. Activarea **E** provoacă generatorul PRBS **106** să calculeze valoarea următoare din secvența sa, care este SAU cu valoarea stocată în interiorul MASK **108** și ieșirea prin intermediul unui driver sau circuit tampon **112**. În exemplul de realizare ilustrat în Figura 1, numai un modul **104** este activat la un moment dat, prin semnalele **114** de activare și valoarea modulului activat devine următoarea valoare **116** de ieșire produsă de sistemul **100**.

Într-un exemplu de realizare, fiecare modul **104** produce o secvență pseudo-aleatoare de lungimi diferite. De exemplu, dacă sistemul **100** include șapte module **104** ($N = 7$), generatorul $PRBS_7$ poate produce apoi o secvență pseudo-aleatoare, care este de 7 biți lățime și produce, astfel, o secvență repetitivă de 128 de valori, de exemplu, un PRBS de lungime 128. $PRBS_6$ poate produce o secvență pseudo-aleatoare, care este de 6 biți lățime și, astfel, produce o secvență repetitivă de 64 de valori, de exemplu, un PRBS de lungime 64. $PRBS_5$ poate produce o secvență 5 biți lățime având un PRBS de lungime 32, $PRBS_4$ poate produce o secvență de 4 biți având o lățime PRBS de 16, și așa mai departe, cu $PRBS_1$ care are o secvență de 1 bit având un PRBS de lungime 2 și $PRBS_0$ având o secvență de 0 biți și producând doar o singură valoare atunci când este selectat. În acest exemplu de realizare, sistemul **100** poate produce o secvență uniformă de amestecare aleatoare a oricăror lungimi de

la 2 până la 255, prin activarea sau dezactivarea selectivă a modulelor specifice **104**. Acest lucru va fi descris mai în detaliu în continuare.

Valorile MASK **108** modifică ieșirea PRBS **106** pentru a produce o secvență pseudo-aleatoare de biți care ocupă o anumită parte din numărul total de secvențe produse de sistemul **100**. Acest lucru este ilustrat în Tabelul 1, de mai jos, care prezintă un exemplu de realizare pentru producerea unei secvențe uniforme de amestecare aleatoare de lungime 15, care produce valori binare în intervalul 0000 - 1110. În această configurație, sunt necesare numai generatoarele PRBS₃, PRBS₂, PRBS₁ și PRBS₀. Dacă sistemul **100** include PRBS-uri suplimentare, cum ar fi PRBS₄, PRBS₅, și așa mai departe, aceste PRBS-uri nu sunt necesare pentru a produce o lungime de amestecare de 15 și pot fi dezactivate sau altfel, neutilizate. (Acestea vor fi activate și utilizate pentru lungimi de amestecare mai mari)

Tabelul 1

PRBS #	Lățime	Interval de ieșire	Valoare Mask	Valorile de ieșire
3	3	000	0000	0000
		001		0001
		010		0010
		011		0011
		100		0100
		101		0101
		110		0110
		111		0111
2	2	00	1000	1000
		01		1001
		10		1010
		11		1011
1	1	0	1100	1100
		1		1101
0	0	0	1110	1110

Se poate observa în Tabelul 1 că fiecare PRBS **106** produce o parte diferită a valorilor de ieșire. Când este selectat, PRBS₃ produce valori pseudoaleatoare în intervalul de 0-7, PRBS₂ produce valori pseudoaleatoare în intervalul 8-11, PRBS₁ produce valori pseudoaleatoare în intervalul 12 -13, iar PRBS₀, când s-a selectat, scoate valoarea 14. Astfel, valorile MASK **108** definesc porțiunea de valori de ieșire produse de un anumit modul **102**.

Tabelul 2 prezintă un alt exemplu, în care sistemul **100** este configurat pentru a produce o secvență uniformă de amestecare aleatoare de lungime 11, care produce valori binare în intervalul 0000 – 1010, în această configurație PRBS₂ nefiind utilizat.

Tabelul 2

PRBS #	Lățime	Interval de ieșire	Valoare Mask	Valorile de ieșire
3	3	000 . . 111	0000	0000 0000 . . 0111
2	2	00 . . 11		
1	1	0 1	1100	1100 1101
0	0	0	1110	1110

Exemplul de realizare ilustrat în Figura 1 este destinat a fi ilustrativ și nu limitativ. De exemplu, driverele de ieșire **112** pot fi înlocuite cu un circuit de selecție care acceptă ca intrare semnalele **114** de activare, care este folosit pentru a selecta în mod logic ieșirea uneia dintre funcțiile SAU **110** ca următoarea valoare de ieșire **116**.

Figura 2 este o schemă bloc care ilustrează în detaliu un circuit de selecție aleatoare ponderată în conformitate cu un exemplu de realizare a prezentei invenții. Circuitul de selecție aleatoare ponderată va fi descris cu referire la Figurile 1 și 2. În exemplul de realizare prezentat în Figura 2, selectorul aleator ponderat **102** activează o linie de selecție **114** la un moment în conformitate cu un algoritm de amestecare la întâmplare. Într-un exemplu de realizare, selectorul **102** include un registru (LFSR) **200** de deplasare liniară - feedback de K biți, care creează o valoare aleatoare de M - biți, care este stocată în registrul RAND **202**.

Într-un exemplu de realizare, selectorul **102** conține un circuit de ponderare pentru fiecare linie de selecție **114**, de exemplu, un circuit de ponderare per modulul **104**, așa cum s-a prezentat în Figura 1 disponibilă pentru selecție. În exemplul prezentat în Figura 2, selectorul **102** include un registru PONDERE **204** pentru stocarea unui număr care este folosit pentru a calcula o pondere de module **104** care sunt conectate la linia de selecție **114** corespunzătoare.

În timpul funcționării, LFSR **200** generează o valoare aleatoare sau pseudoaleatoare, care este stocată în registrul RAND **202**. Valoarea stocată în RAND **202** este comparată cu valorile stocate în fiecare registru PONDERE **204**, prin circuitele de comparare **206**, iar în cazul în care valoarea în RAND **202** este mai mică decât valoarea stocată în registrul PONDERE **204**, ieșirea circuitului de comparație **206** merge activă, ceea ce determină ca valoarea în registrul PONDERE **204** să fie scăzută cu unu și care este, de asemenea, furnizată ca intrare **208** într-un circuit de prioritate **210**. Circuitul de prioritate **210** activează o linie de selecție **114**, care

corespunde la intrare **208** de la modulul cu cea mai înaltă prioritate, care, în varianta ilustrată în Figura 2 este modulul care are cel mai mare număr N. Modulul selectat **104** furnizează valoarea la IEȘIREA **116**, și procesul se repetă. De fiecare dată când un nou număr aleator este încărcat la registrul RAND **202**, cel puțin unul dintre registre PONDERE **204** vor scădea. În varianta prezentată în Figura 2, în timpul funcționării, LFSR **200** produce o valoare aleatoare între 0 și o valoare mai mică decât valoarea stocată în registrul PONDERE **204** pentru modulul 0 ("PONDERE0"). Astfel, așa cum sunt produse valori aleatoare, devine progresiv mai mic intervalul de valori aleatoare posibile. Valorile stocate în registrele PONDERE **204** operează pentru a împărți intervalul de valori aleatoare posibile în porțiuni distincte sau "felii". Fiecare felie este atribuită unei anumite linii **114** de selecție (și prin extindere modulului special **104** la care linia de selecție **114** este conectată.) În timpul funcționării, valorile stocate în registrele PONDERE **204** sunt manipulate pentru a determina mărimea porțiunii (sau "feliei") alocate pentru un modul special **104** pentru a se reduce de fiecare dată când este selectat modulul special. Pentru a explica modul în care funcționează acest lucru, un exemplu ar fi de ajutor.

Un exemplu de utilizare a registrelor PONDERE **204** va fi descris pentru un sistem cu patru module **104**, de exemplu, care produce o valoare aleatoare între 0 și 14.

Tabelul 3

MOD # (N)	Lățimea	# de Valori	Porțiune alocată	PONDERE
3	3	8	0, 1, 2, 3, 4, 5, 6, 7	8
2	2	4	8, 9, 10, 11	12
1	1	2	12, 13	14
0	0	1	14	15

Tabelul 3 este o listă de module **104** din numărul de module (coloana "MOD #"), arătând lățimea PRBS a fiecărui modul (coloana "Lățime"), numărul de valori în secvența sa pseudoaleatoare (coloana "# de valori"), gama de posibile valori de ieșire pentru care acest modul va fi responsabil de furnizarea, de exemplu, a "feliei" lui de valori de ieșire posibile (coloana "felie alocată"), și valoarea care va fi pusă în registrele PONDERE **204** ale modulelor (coloana "PONDERE"). La selectare, modulul 3 va produce valori de ieșire în domeniul 0 până la 7, modulul 2 va produce valori de ieșire în intervalul între 8 și 11, modulul 1 va produce valori de ieșire de 12 și 13, precum și modul 0 va produce valoare de ieșire 14. Intervalul de valori alocate pentru fiecare modul este controlat de valorile stocate în registrele PONDERE **204** respective.

Folosind Tabelul 3 ca punct de plecare, valoarea inițială a registrului PONDERE0 este "15", astfel încât prima valoare produsă de LFSR **200** și stocată în RAND **202** va fi o valoare între 0 și 14. Tabelul 4 prezintă evoluția valorilor stocate în interiorul selectorului **102** în timpul unui exemplu de funcționare. Coloanele reprezintă cicluri de funcționare.

Tabelul 4

Ciclul	0	1	2	3	4	5	6	7	8	...
PONDERE3	8	6	<u>7</u>	7	<u>6</u>	6	6	<u>5</u>	5	...
PONDERE2	12	<u>11</u>	<u>10</u>	10	9	<u>8</u>	8	<u>7</u>	7	...
PONDERE1	14	<u>13</u>	<u>12</u>	<u>11</u>	<u>10</u>	<u>9</u>	9	<u>8</u>	<u>7</u>	...
PONDERE0	15	<u>14</u>	<u>13</u>	<u>12</u>	<u>11</u>	<u>10</u>	<u>9</u>	<u>8</u>	<u>7</u>	...
RAND	-	9	2	11	5	8	9	3	7	...
SELECTAT	-	2	3	1	3	2	0	3	1	...

În exemplul ilustrat în Tabelul 4, registrele PONDERE pentru modulele 3, 2, 1 și 0, sunt 8, 12, 14, și respectiv 15, așa cum se arată la momentul zero. La ciclul 1, este generat un număr aleator între 0 și 15 (valoarea stocată în PONDERE0). În acest exemplu, numărul generat a fost un "9". Valoarea 9 nu este mai mică decât valoarea registrului PONDERE3, deci nu se schimbă. Cu toate acestea, 9 este mai mic decât valorile stocate în registrele PONDERE2, PONDERE1, și PONDERE0, astfel încât acestea sunt scăzute și noile lor valori sunt prezentate în coloana pentru ciclul 1. În acest tabel, valorile scăzute sunt evidențiate prin subliniere. În ciclul 1, intrările **208** corespunzătoare MOD2, MOD1 și MOD0 sunt toate active, dar numai modulul cu cea mai înaltă prioritate, MOD2, va fi selectat de către circuitul de prioritate **210**, așa cum se arată în rândul de jos din Tabelul 4. În acest tabel, modulele selectate, de exemplu, care furnizează următoarea valoare în secvența de ieșire, sunt prezentate în culoare inversă, adică, text alb pe fond negru.

În timpul ciclului 2, va fi generat un număr aleator între 0 și 13. În acest exemplu, numărul următor generat a fost un "2". Deoarece valoarea 2 este mai mică decât valorile registrelor PONDERE3, PONDERE2, PONDERE1 și PONDERE0, valorile tuturor acestor registre vor scădea, intrările **208** corespunzătoare MOD3, MOD2, MOD1 și MOD0 vor fi active, iar circuitul de prioritate **210** va activa numai linia de selecție pentru modulul care are cea mai mare prioritate, de exemplu, MOD3.

În timpul ciclului 3, este generat un număr între 0 și 12, de exemplu, "11". Din moment ce

valoarea 11 este mai mică decât valorile stocate în registrele PONDERE1 și PONDERE0, doar valorile registrelor PONDERE1 și PONDERE0 vor scădea și numai acele intrări **208** ale căror registru PONDERE1 are cea mai mare prioritate vor fi active. Astfel, va fi activat modulul MOD1. Acest proces continuă până când valorile tuturor registrelor PONDERE ajung la zero.

Se observă că în ciclul 6, valorile registrelor PONDERE1 și PONDERE0 sunt aceleași. Acest lucru înseamnă că, din acest moment și până când valorile PONDERE sunt reinițializate, de exemplu, pentru a începe o altă secvență aleatoare, modulul MOD0 nu va fi niciodată selectat deoarece următorul modul cu prioritatea cea mai mare, MOD1, va fi de asemenea selectat în același timp și MOD1 va avea întotdeauna prioritate față de MOD0. Este de așteptat, totuși, ca modulul MOD0 să trebuiască selectat o singură dată, și anume, la valoarea de ieșire "14" (a se vedea Tabelul 3). Se mai observă că în ciclul 8, valorile PONDERE2, PONDERE1 și PONDERE0 sunt aceleași. Acest lucru înseamnă că, din acest moment, modulele MOD0 și MOD1 nu vor fi niciodată selectate deoarece următorul modul de cea mai mare prioritate, MOD2, va fi de asemenea selectat în același timp și MOD2 va avea totdeauna prioritate față MOD1 și MOD0. Acest lucru este, de asemenea, de așteptat, deoarece modulul MOD1 trebuie selectat doar de două ori, și anume, la valorile de ieșire "12" și "13" (a se vedea Tabelul 3). În acest mod, odată ce un anumit modul are la ieșire toate posibilele sale valori, rezultă că modulul va nu fi selectat din nou, până când sistemul este reinițializat la ieșire cu o altă secvență completă de amestecare.

Astfel, fiecare modul **104** are atribuită inițial o "felie" de valori care are o lățime care corespunde numărului de valori pe care le poate furniza modulul special. Se poate observa că, fiecare modul **104** este selectat pentru a furniza o ieșire **116**, lățimea feliei alocate fiind redusă în raport cu lățimile modulelor neselectate **104**. Astfel, probabilitatea ca modulul selectat **104** să fie selectat prin următorul număr aleator stocat în RAND **202** este redusă în mod corespunzător. Lățimile relative ale feliilor sunt controlate de valorile încărcate în registrele PONDERE **204**. În exemplul de realizare prezentat în Figura 2, lățimea modulului MOD N este $(PONDERE_{N-1} - PONDERE_N)$. Dacă $PONDERE_{N-1} = PONDERE_N$, atunci lățimea MOD $N = 0$, iar MOD N este dezactivat în mod eficient.

Tabelul 5 prezintă modulele **104**, care vor fi activate/dezactivate pentru a genera o secvență amestecată aleator de lungime L . Un X indică un modul activat, în timp ce o celulă goală indică un modul dezactivat.

Tabelul 5

	L														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MOD3								X	X	X	X	X	X	X	X
MOD2				X	X	X	X					X	X	X	X
MOD1		X	X			X	X			X	X			X	X
MOD0	X		X		X		X		X		X		X		X

După cum se poate vedea din Tabelul 5, pentru generarea secvenței uniforme, aleator amestecate care are 10 valori, modulele MOD3 și MOD1 ar fi activate și celelalte module dezactivate: MOD3 ar furniza 8 din cele 10 valori, în timp ce MOD1 ar furniza 2 valori care rămân din 10 valori. Prin activarea sau dezactivarea anumitor module, poate fi generată o secvență uniformă amestecată aleator de lungimi specifice. În acest exemplu de realizare, aceeași piesă hardware poate fi reconfigurată destul de flexibil. Adăugarea de alte module, de exemplu, MOD 4, MOD 5, etc, pot genera secvențe de lungimi mai mari.

Logica de selecție aleatoare ponderată prezentată în Figura 2 este ilustrativă și nu intenționează să fie limitativă. Alte tehnici pentru generarea de selecție aleatoare ponderată de ieșirile 116 sunt avute în vedere și pot fi utilizate în cadrul sistemului 100.

Figura 3 este o schemă bloc care prezintă în detaliu un modul pentru generarea unei secvențe pseudo-aleatoare de biți în conformitate cu un exemplu de realizare a prezentei invenții. Modulul pentru generarea unei secvențe pseudoaleatoare de biți va fi acum descris cu referire la Figurile 1 și 3. În exemplul de realizare prezentat în Figura 3, modulul 104 include un generator (PRSG) de secvență pseudo-aleatoare 300. Într-un exemplu de realizare, PRSG 300 include un registru de deplasare liniară-feedback. În timpul funcționării, un registru VALUE 302 are valoarea următoarea în secvența care trebuie furnizată de către modulul 104 la activarea unei intrări 304. În exemplul de realizare ilustrat în Figura 1, de exemplu, liniile de selecție 114 sunt conectate la intrările activate pentru modulele 104. Registrul VALUE 302 devine valoarea lui următoare prin intermediul unui selector 306, care selectează în mod normal ieșirea generatorului PRSG 300 ca și sursă de date pentru a fi stocate în registrul VALUE 302. La fel, PRSG 300 devine ca intrarea sa următoare printr-un al doilea selector 308, care în mod normal selectează ieșirea registrului VALUE 302 ca sursă de date care trebuie furnizate la PRSG 300. Astfel, pentru majoritatea funcționării modulului 104, PRSG 300 produce următorul registru VALUE 302, care este următoarea intrare pentru PRSG 300, care produce următorul registru VALUE 302, și așa mai departe.

Totuși, un dezavantaj la utilizarea unui LFSR pentru a genera o secvență pseudo-aleatoare

este că un LFSR de lățime M produce $2^M - 1$ valori, nu 2^M valori. Valoarea neprodusă de un LFSR este menționată ca valoare de "blocare", pentru că aspectul său ar determina LFSR să continue să genereze aceeași valoare din acel moment, provocând LFSR să apară pentru a fi închis. Pentru LFSR-urile de tip SAU EXCLUSIV, de exemplu, valoarea de blocare este formată numai din zerouri.

Din acest motiv, fiecare modul **104** include un registru LOCKUP **310** pentru stocarea valorii de blocare. Un registru DISCONTINUTY **312** stochează o valoare din lista de valori produse de LFSR **300**. În cazul în care valoarea stocată în registrul VALUE **302** corespunde cu valoarea în registrul DISCONTINUTY **312**, așa cum este determinat de circuitul de comparare **314**, de exemplu, selectorul **306** alege ieșirea registrului LOCKUP **310** ca și a registrului VALUE **302**. În ciclul următor, un al doilea circuit de comparare **316** detectează că registrul VALUE **302** conține valoarea de blocare, și instruește selectorul **308** pentru a trimite valoarea DISCONTINUTY ca următoarea intrare în PRSG **300** mai mult decât valoarea de blocare prezentă în mod curent în registrul VALUE **302**, așa că PRSG **300** nu se blochează. Valoarea de blocare prezentă în mod curent în registrul VALUE **302** determină circuitul de comparare **314** să aducă selectorul **306** înapoi la starea sa normală. Numărul următor produs de PRSG **300** va fi valoarea în secvență *după* valoarea de discontinuitate. În acest mod, modulul **104** are capacitatea de a introduce valoarea de blocare în orice loc în secvența PRSG, unde registrul DISCONTINUTY **312** definește punctul de inserare. Adăugarea registrelor LOCKUP **310** și DISCONTINUTY **312** creează un modul care produce un set complet de 2^M de valori pseudoaleatoare.

Într-un exemplu de realizare, secvența PRBS creată de PRSG **300** poate fi amestecată în continuare pentru a oferi amestecarea aleatoare suplimentară. În exemplul prezentat în Figura 3, de exemplu, ieșirea registrului VALUE **302** poate fi modificată înainte de a fi transmisă de către modulul **104** prin executarea unei operații SAU EXCLUSIV folosind registru de identitate secvență SEQID **320** și un circuit exclusiv SAU **322** și prin stocarea rezultatului operației SAU EXCLUSIV într-un registru de valori de ieșire **324**. Operația SAU EXCLUSIV schimbă ordinea secvenței PRBS.

Valorile stocate în registrele DISCONTINUTY **312** și/sau SEQID **320** pot fi schimbate, de exemplu, după fiecare secvență completă de ieșire, mai mult la întâmplare, de rezultate, astfel încât valorile de secvență de amestecare să se schimbe de la o rulare la alta. Într-un exemplu de realizare, registrele DISCONTINUTY **312** și/sau SEQID **320** pot fi separate de LFSR-uri sau felii dintr-un LFSR mai mare. De asemenea, fiecare modul **104** poate avea propriul LFSR în acest scop sau poate împărți toate feliile ale unui LFSR chiar mai mare, în acest scop. Alte

exemple de realizare sunt de asemenea avute în vedere, care includ valori predeterminate pentru registrele DISCONTINUTY și SEQID, inclusiv registre programabile de utilizator pentru aceste valori, care pot crește sau descrește automat la începutul fiecărei rulări, și așa mai departe.

Figura 4 este o organigramă care ilustrează o metodă de implementare hardware a pachetelor de date uniforme, amestecate aleator, conform unui exemplu de realizare a prezentei invenții. Etapa **400** include furnizarea de mai multe seturi de numere folosind hardware pentru generarea de secvențe pseudo-aleatoare, fiecare set fiind o secvență pseudo-aleatoare care reprezintă un subset de toate valorile posibile ale unei secvențe *O* amestecată la întâmplare, în care toate seturile combinate reprezintă toate valorile secvenței *O*. În exemplul de realizare prezentat în Figura 1, de exemplu, sistemul **100** include mai multe module **104**, fiecare modul **104** având hardware (PRBS **106**, MASK **108**, etc.) pentru producerea unei secvențe pseudo-aleatoare.

Etapa **402** include producerea secvenței *O* prin selectarea repetată a valorii următoare a unui set selectat aleator ca și valoare următoare în secvența *O*. În exemplul prezentat în Figura 1, de exemplu, sistemul **100** include un selector **102** pentru selectarea în mod aleator a unuia dintre modulele **104** pentru a furniza valoarea următoare în secvența *O* de ieșire. Selectorul **102** și modulele **104** funcționează împreună pentru a produce secvențe *O* de ieșire amestecate aleator.

Figura 5 este o schemă bloc care prezintă un sistem pentru efectuarea testelor de depreciere a unui dispozitiv de rețea de pachete de date, în conformitate cu un exemplu de realizare a prezentei invenții. În varianta prezentată în Figura 5, sistemul **500** include modulul de generare secvență **502** pentru generarea unui set aleator de numere utilizate pentru a defini o secvență *O* de amestecare. Într-un exemplu de realizare, modulul de generare de secvență **502** include un set de unul sau mai multe module (PRSG) **504** de generare de secvență pseudoaleatoare. Fiecare modul PRSG **504** generează propria sa secvență pseudo-aleatoare de numere și emite următoarea valoare în secvența sa pseudo-aleatoare, ca răspuns la o cerere de ieșire. Modulul de generare secvență **502** include logica de selecție **506** pentru crearea secvenței *O* de amestecare selectând repetat unul dintre modulele PSRG **504** la întâmplare și transmițând o cerere de ieșire la modulul selectat. Secvența *O* de amestecare creată de ieșirea modulelor selectate aleator în acest mod, este o secvență uniformă, amestecată la întâmplare. În varianta prezentată în Figura 5, sistemul **500** include un modul **508** de reordonare pachete de date pentru fragmentarea unui prim pachet de rețea **510** într-un set de fragmente secvențial ordonate $\{1, 2, 3, \dots, N\}$ și pentru amestecarea ordinii fragmentelor conform secvenței *O* de

amestecare. Fragmentele **512** amestecate pot fi trimise în mod individual la un modul de ieșire **514** pentru transmiterea printr-o conexiune de rețea **516** la un dispozitiv de testat (DUT) **518**, sau pot fi reasamblate într-un al doilea pachet de rețea înainte de a fi transmise prin modulul de ieșire **514**.

De exemplu, într-un exemplu de realizare, primul pachet de rețea neamestecat **510** poate fi stocat într-o locație în memorie și a doua rețea amestecată **512** poate fi stocată într-o altă locație din memorie, de la care modulul de ieșire **514** preia fragmentele de transmis la DUT **518**. Într-un exemplu de realizare alternativ, pachetul de date de rețea neamestecat **510** este stocat într-o locație în memorie și modulul de ieșire **514** utilizează secvența O de amestecare ca un index în acea memorie, recuperează fragmentul care este stocat acolo și transmite acel fragment la DUT **518**. Acest exemplu de realizare are avantajul că nu este nevoie de stocarea unei copii a pachetului amestecat în memoria **512** - în schimb, este creat în mod dinamic, rapid, în timpul etapei de ieșire.

Într-un exemplu de realizare, modulul de generare secvență **502** poate fi un sistem **100** de implementare hardware a pachetelor de date uniforme, amestecate aleator, așa cum este ilustrat în Figurile 1, 2 și 3.

Figura 6 este o organigramă care ilustrează o metodă pentru efectuarea testelor de deprecierea unui dispozitiv de rețea de pachete de date, în conformitate cu un exemplu de realizare a prezentei invenții. În exemplul prezentat în Figura 6, procesul include, la etapa **600**, folosirea hardware pentru a genera mai multe seturi de secvențe pseudo-aleatoare și crearea unui secvențe O de ieșire amestecată aleator prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator selectat ca următoarea valoare în secvența O până când au fost produse toate valorile secvenței O. Etapa **602** include folosirea secvenței O produsă pentru a amesteca fragmente dintr-un prim pachet de rețea pentru a crea o a doua rețea de pachete de date având fragmente amestecate. Etapa **604** include transmiterea celui de-al doilea pachet de rețea la un dispozitiv de testat.

Se înțelege că diferite detalii ale prezentei invenții pot fi modificate fără îndepărtarea de la scopul invenției descrise în prezenta documentație. Mai mult decât atât, descrierea de mai sus este în scop numai de ilustrare și nu în scopul limitării.

REVENDICĂRI

1. Metodă pentru implementarea hardware a pachetelor de date uniforme, amestecate aleator, care constă în:
furnizarea, folosind hardware pentru generarea de secvențe pseudo-aleatoare, a unei multitudini de seturi de numere, în care fiecare set cuprinde o secvență pseudo-aleatoare, în care fiecare set reprezintă un subset de toate valorile posibile ale unei secvențe O amestecate la întâmplare, și în care toate seturile combinate reprezintă toate valorile secvenței O ; și
producerea secvenței O prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator selectat ca următoarea valoare în secvența O .
2. Metodă, conform revendicării 1, **caracterizată prin aceea că** mai constă în continuarea selectării repetate a următoarei valori până ce au fost utilizate toate valorile de la toate seturile.
3. Metodă, conform revendicării 1, **caracterizată prin aceea că** seturile sunt selectate aleator utilizând o funcție ponderată de selecție aleatoare și **prin aceea că** fiecare set are atribuită propria pondere.
4. Metodă, conform revendicării 3, **caracterizată prin aceea că** ponderea fiecărui set se bazează pe numărul de valori în setul care nu au fost încă selectat pentru a fi utilizat în secvența O .
5. Metodă, conform revendicării 4, **caracterizată prin aceea că** pentru fiecare set de numere, secvența pseudo-aleatoare pornește de la o valoare inițială V care este selectată prin programare.
6. Metodă, conform revendicării 5, **caracterizată prin aceea că** secvența pseudo-aleatoare se oprește atunci când valoarea următoare din secvență este din nou V .
7. Metodă, conform revendicării 5, **caracterizată prin aceea că** pentru fiecare set de numere, a fost produsă valoarea V pentru care setul se modifică după toate valorile în secvența O .

8. Metodă, conform revendicării 7, **caracterizată prin aceea că** valoarea V se modifică în conformitate cu o a doua secvență pseudoaleatoare.
9. Metodă, conform revendicării 1, **caracterizată prin aceea că** pentru fiecare set de numere, valoarea care este ieșirea din acel set este modificată de o funcție M de modificare pentru care setul este inclus înainte în secvența de ieșire O.
10. Metodă, conform revendicării 1, **caracterizată prin aceea că** pentru fiecare set de numere, funcția M de modificare se schimbă după ce au fost produse toate valorile din secvența O.
11. Metodă pentru efectuarea testelor de deprecierea unui dispozitiv de rețea de pachete de date care constă în:
folosirea hardware pentru a genera o multitudine de seturi de secvențe pseudoaleatoare și crearea unei secvențe de ieșire O amestecate aleator prin selectarea repetată a următoarei valori a unui set aleator sau pseudo-aleator selectat ca următoarea valoare în secvența O până când au fost produse toate valorile secvenței O; folosirea secvenței O produsă pentru a amesteca fragmente dintr-un prim pachet de rețea pentru a crea o a doua rețea de pachete de date având fragmente amestecate, care sunt transmise la un dispozitiv de testat.
12. Sistem de implementare hardware a pachetelor de date uniforme aleator amestecate, care cuprinde:
o multitudine de module, în care fiecare modul generează propria secvență pseudoaleatoare de numere și în care fiecare modul emite următoarea valoare în secvența sa pseudo-aleatoare, ca răspuns la o cerere de ieșire; și
logică de selecție pentru a crea o secvență de valori de ieșire O prin selectarea repetată a unuia din multitudinea de module în conformitate cu o funcție de selectare aleatoare și prin transmiterea unei cereri de ieșire la modulul selectat, în care secvența de valori O creată de ieșirea modulelor selectate aleator cuprinde o secvență uniformă, amestecate la întâmplare.
13. Sistem, conform revendicării 12, **caracterizat prin aceea că** funcția de selectare aleatoare este o funcție ponderată de selectare aleatoare și **prin aceea că** fiecare modul

are atribuită propria sa pondere.

14. Sistem, conform revendicării 13, **caracterizat prin aceea că** ponderea fiecărui modul se calculează în funcție de numărul N de valori pseudoaleatoare care nu au ieșit încă din valorile L posibile în secvența S produsă de acel modul.
15. Sistem, conform revendicării 12, **caracterizat prin aceea că** pentru fiecare modul, secvența pseudo-aleatoare pornește de la o valoare inițială care V este selectată prin programare.
16. Sistem, conform revendicării 15, **caracterizat prin aceea că** pentru fiecare modul, se modifică valoarea V, după ce toate valorile în secvența S au fost ieșit prin acel modul.
17. Sistem, conform revendicării 16, **caracterizat prin aceea că** valoarea V se modifică în conformitate cu o a doua secvență pseudoaleatoare.
18. Sistem, conform revendicării 12, **caracterizat prin aceea că** pentru fiecare modul, valoarea care este ieșirea din acea secvență pseudo-aleatoare este modificată de o funcție M de modificare pentru care setul este inclus înainte în acel modul.
19. Sistem, conform revendicării 18, **caracterizat prin aceea că** funcția de modificare exclusiv SAU iese din secvența pseudo-aleatoare cu o valoare M de modificare pentru a produce valoarea care este transmisă de către acel modul.
20. Sistem conform, revendicării 19, **caracterizat prin aceea că** pentru fiecare modul, valoarea M se modifică după ce toate valorile din secvența S au fost generate de către acel modul.
21. Sistem pentru efectuarea testelor de depreciere a unui dispozitiv de rețea de pachete de date care cuprinde:
un modul de amestecare pentru amestecarea aleatoare a unui set de numere pentru a crea o secvență O de amestecare, în care modulul de amestecare include mai multe module care generează fiecare propria sa secvență pseudo-aleatoare de numere și care produce următoarea valoare în secvența sa pseudo-aleatoare ca răspuns la

recepționarea unei cereri de ieșire, în care fiecare modulul de amestecare include logică de selecție pentru crearea secvenței O de amestecare prin selectarea repetată la întâmplare a unuia din multiplele module și prin transmiterea unei cereri de ieșire la modulul selectat, în care secvența O de amestecare creată de ieșirea modulelor selectate aleator conține o secvență uniformă, amestecată la întâmplare;

un modul de reordonare pachete de date pentru fragmentarea unui prim pachet de rețea în interiorul unei multitudini de fragmente secvențial comandate, care amestecă ordinea fragmentelor conform cu secvența O de amestecare, și care assemblează fragmentele amestecate în interiorul unui al doilea pachet de rețea; și

un modul de ieșire pentru transmiterea celui de-al doilea pachet de rețea către un dispozitiv de testat.

22. Suport non-tranzitoriu care poate fi citit de calculator care are stocate instrucțiuni executabile de calculator care atunci când sunt executate de către procesorul unui calculator comandă acestuia să efectueze pașii care constau în:
- furnizarea, folosind hardware pentru generarea de secvențe pseudo-aleatoare, a unei multitudini de seturi de numere, în care fiecare set cuprinde o secvență pseudo-aleatoare, în care fiecare set reprezintă un subset de toate valorile posibile ale unei secvențe O amestecată la întâmplare, și în care toate seturile combinate reprezintă toate valorile secvenței O; și
- producerea secvenței O prin selectarea repetată a următoarei valori a unui set aleator selectat ca următoarea valoare în secvența O, până când au fost utilizate toate valorile de la toate seturile.

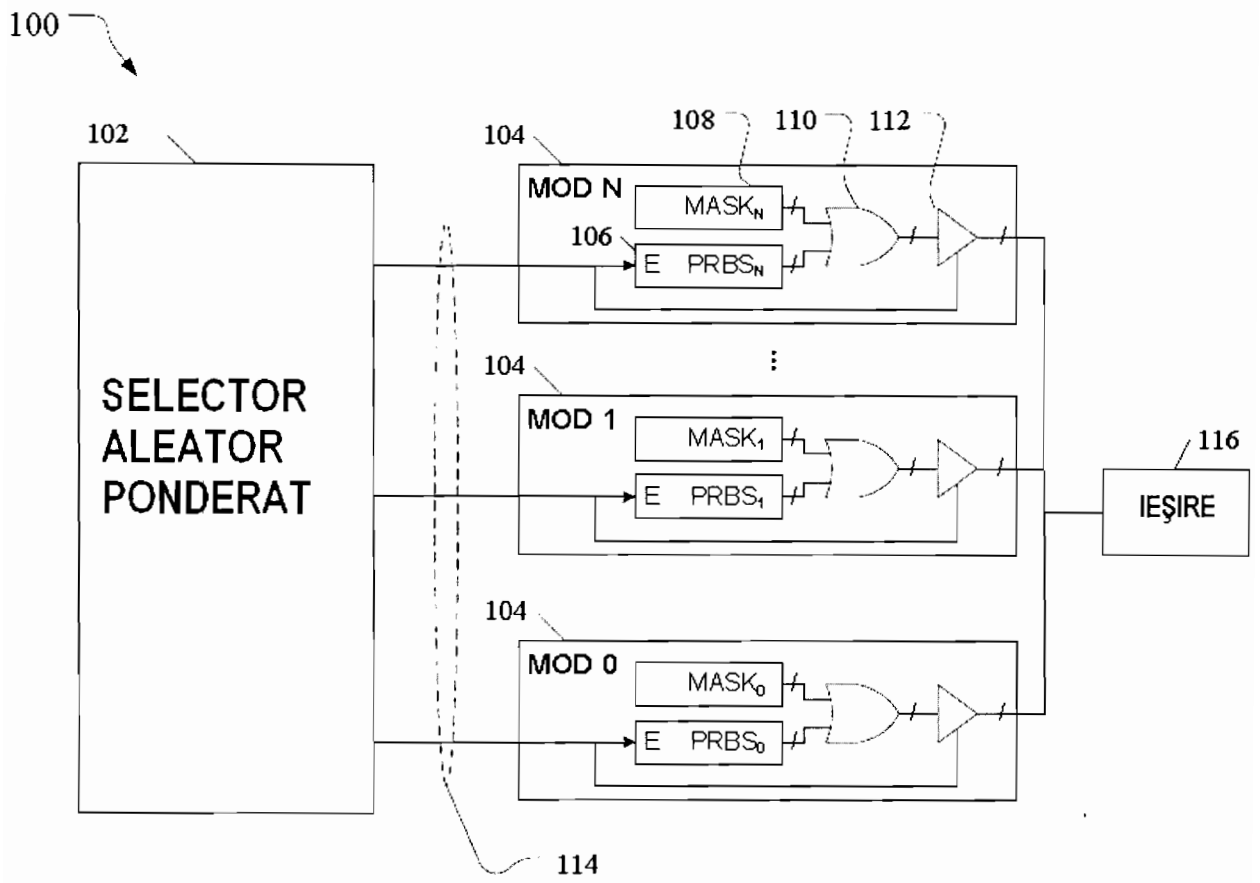


FIG. 1

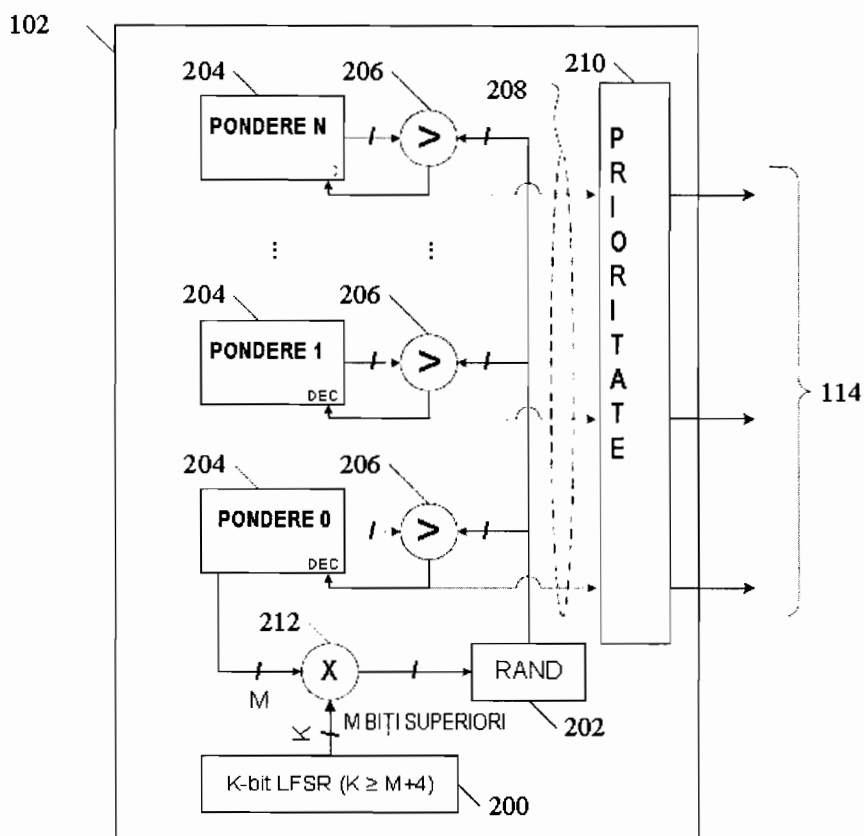


FIG. 2

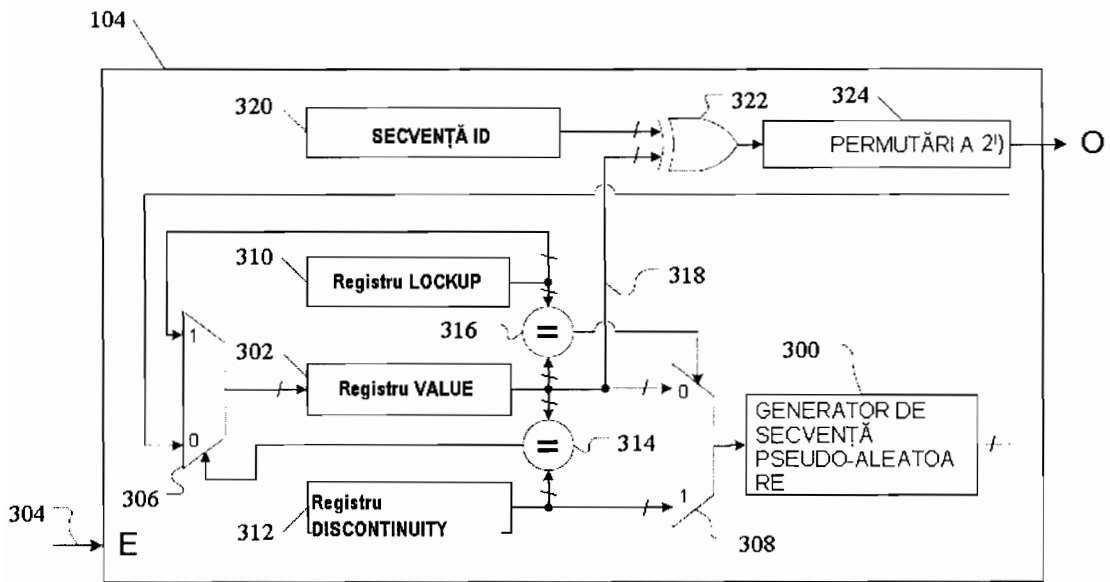
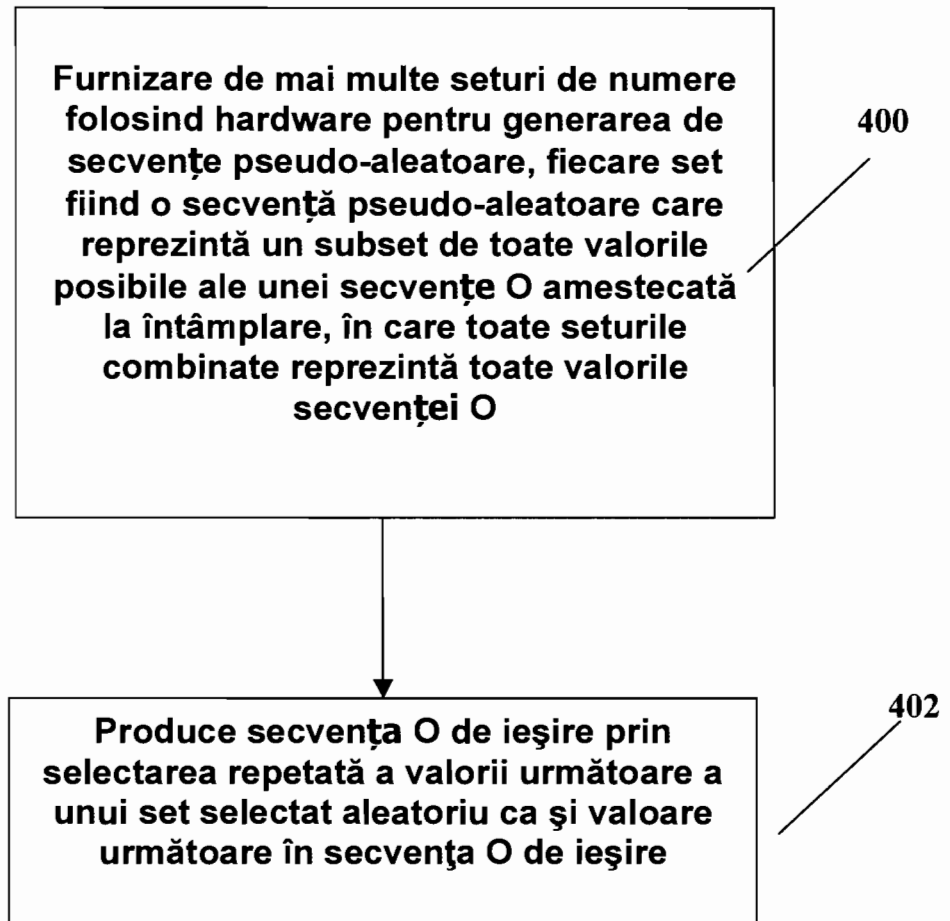


FIG. 3

**FIG. 4**

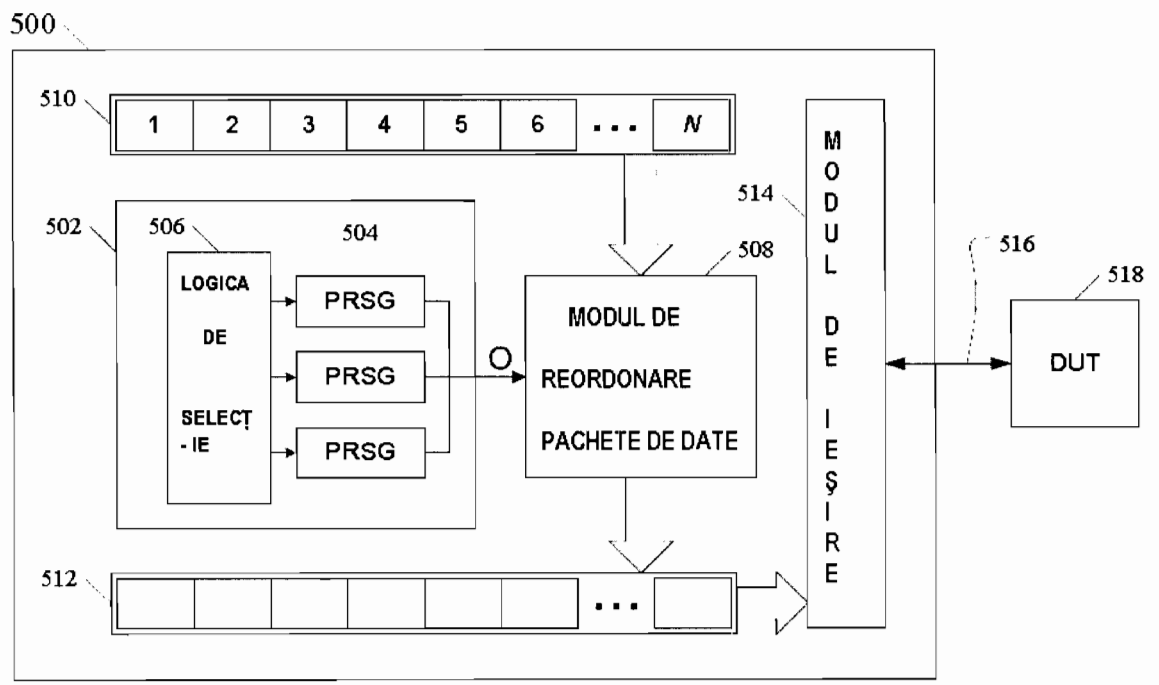


FIG. 5

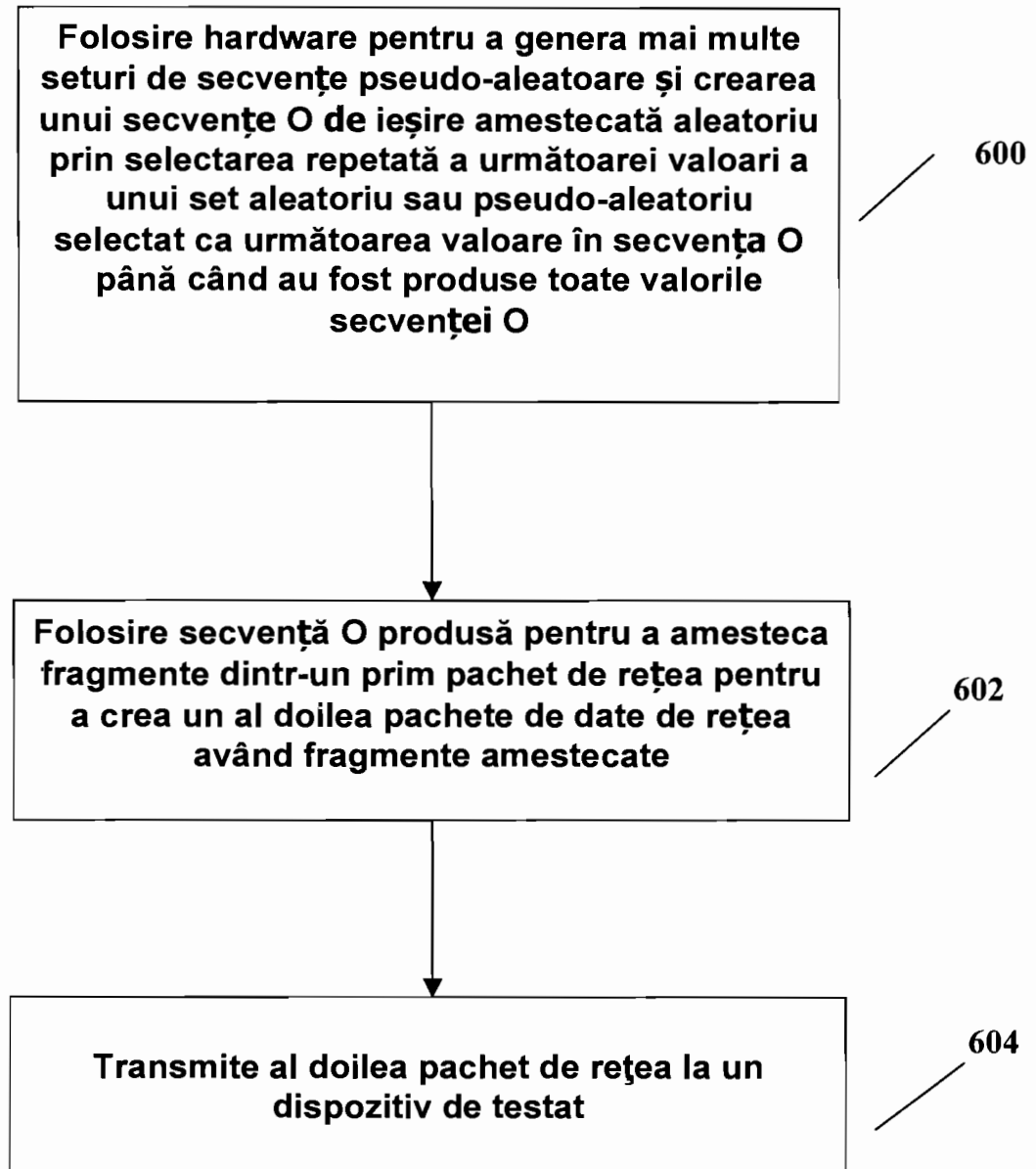


FIG. 6