**UNIVERSITATEA "ȘTEFAN CEL MARE" DIN SUCEAVA**
**Facultatea de Inginerie Electrică și Știința Calculatoarelor**

# DEZVOLTAREA DE APLICAȚII ÎN CRIPTOGRAFIE, TESTARE ȘI CODURI DETECTOARE ȘI CORECTOARE DE ERORI

- R E Z U M A T -

Conducător științific:
Prof. univ. dr. ing. **Ștefan – Gheorghe PENTIUC**

Doctorand:
As. ing. **SBIERA Mirella Amelia (MIOC)**

**2016**

# 1. Introducere

## 1.1. Generalități

Unul dintre primele modele asemănătoare cu un registru de deplasare a fost o mașină creată pentru spargere de cod care a apărut în anii 40. Acel dispozitiv a fost construit din tuburi de vid si tiratroane. Alte implementări au fost dezvoltate în următorii ani.
În proiectarea de tip hardware un rol deosebit îl are Registrul liniar de deplasare cu feedback, numit și LFSR.
Un șir de celule de memorie care conțin biți produce o deplasare cu o poziție, prin utilizarea unui impuls de ceas (clock). Prin folosirea unui XOR sau XNOR asupra anumitor poziții un nou bit va fi produs cu ajutorul impulsului de ceas.
Pentru fiecare implementare LFSR circuitele sunt corespunzătoare cu coeficienții unui polinom, numit polinom generator. Pentru creșterea gradului de securitate pe care registrul de deplasare îl conferă, polimomul utilizat trebuie să fie ireductibil sau primitiv.
Polinoamele care satisfac anumite condiții speciale matematice determină un LFSR să producă un număr maxim de secvențe în total (2n-1), denumită perioadă și având n numărul de celule sau lungimea registrului de deplasare.
Întotdeauna un registru care are n biti va avea un număr de n + 1 semnale.
Un registru de deplasare liniar având statusul de 0 pentru fiecare putere nu va putea trece în altă stare astfel încât o astfel de posibilă stare trebuie exclusă din posibilitățile unui ciclu complet. În afara situației mai sus menționate fiecare stare internă va fi determintă de transformarea fiecărui bit selectat prin utilizarea câte unui circuit XOR, detrminând astfel un ciclu complet.
Prin utilizarea unui clock sincron, a unui număr de porți XOR și câteva circuite flip-flop se poate realiza un registru de deplasare. În multe aplicații din circuite digitale, un astfel de registru de deplsare se utilizează ca și numărător.
Un registru de deplasare implementat cu porți XOR sau XNOR va produce întotdeauna o întârziere discretă (delay) a semnalului digital. Atunci când avem un registru cu n circuite întârzierea produsă va fi de n ori clockul.
O altă utilitate importantă a utilizării LFSR este crearea de modele de test ( test patterns).
Există anumite concepte matematice pe care se bazează realizarea de LFSR pentru diverse scopuri. Orice registru de deplasare se bazează pe șiftarea conținutului în pozițiile adiacente și transmitere în afară pentru poziția finală.
Orice registru de deplasare poate avea câteva utilizări obișnuite ca:

- Întârzierea unui sir serial de biți;
- Conversia de date din serial în paralel.

Tabelele care conţin secvenţe (taps) pentru realizarea de registre de deplsare se folosesc de asemenea în teoria protocoalelor de comunicaţii.

În logica digitală există o convenţie în notarea fiecărui bit conform căreia să se înceapă numerotarea cu cel mai puţin semnificativ bit din stânga.

Întotdeauna semnalele trebuie adresate şi nu registrele, pentru a rămâne în acord cu protocolul de comunicare. Astfel, pentru un registru proiectat pentru n biţi vor fi n + 1 semnale.

De fiecare dată starea următoare a unui registru de deplasare cu feedback liniar este determinată în mod unic de feedback-ul stării precedente. Secvenţa generată de LFSR conţine diferite stări care încep cu prima, numită seed (sămânţă).

Un registru de deplasare care are o funcţie de feedback conţine următoarele două părţi:

- O parte principală care constă într-un registru de deplasareş
- O parte auxiliară care conţine funcţia propriu-zisă de feedback.

Fiecare implementare a unui LFSR corespunde unui polinom generator numit şi polinom caracteristic, care este un polinom de variabila x şi de grad n.

În această situaţie, întotdeauna un registru de deplasare cu reacţie liniară este un registru de deplasare având secvenţele generate obţinute prin utilizarea unei funcţii liniare a stării iniţiale ( a seed-ului).

Secvenţa produsă va fi întotdeauna complet determinată de starea iniţială.

Secvenţa se va repeta după o perioadă, deoarece toate stările posibile sunt într-un număr limitat. Un obiectiv special în aplicaţiile criptografice este acela de a alege foarte bine fie polinomul generator sau funcţia de feedback pentru creşterea numărului perioadei pentru secvenţa produsă.

Aşa că, după alegerea forte bună a funcţiei de feedback ciclul obţinut va avea o perioadă foarte lungă şi secvenţa produsă va fi aleatorie.

O secvenţă de tap este o listă care conţine toate poziţiile determinate de coeficienţii polinomului generator care va genera viitoarele stări.

Calitatea de a produce stările de biţi pseudoaleatoare conferă un loc special pentru LFSR în a deveni parte a oricăror sisteme digitale şi prin aceasta în a avea un rol special în aplicaţii criptografice, în testare şi în aplicaţii care utilizează detectarea şi corectarea codurilor de eroare precum şi în sistemele de comunicaţie fără fir.

Orice registru de deplasare cu reacţie liniară poate fi implementată în două tipuri:

• configuraţia convenţională (numită Fibonacci)
• configuraţia tip Galois.

Schema de implementare Fibonacci se realizează având conexiuni acolo unde există coeficienţii polinomul generator, situaţii de excepţie fiind prima şi ultima poziţie, care sunt întotdeauna în starea de 1.

Pentru schema Galois conţinutul registrului de deplasare va fi modificat pentru fiecare pas la o valoare binară care va fi trimisă la ieşire.

Din funcţionarea celor două implementări pentru LFSR şi compararea acestora se poate constata că cele două scheme sunt opuse ca şi ordine a ponderilor (weights).

Schema tip Galois pentru implementarea LFSR utilizează porţi XOR în partea internă şi astfel întârzierea (delay-ul) este mai mică decât în cazul Fibonacci, unde toate porţile XOR sunt în partea de Feedback şi deci determinăîntârziere mai mare.

Fiind mai rapidă decât implementarea Fibonacci, implementarea Galois a devenit mult mai utilizată.

O altă posibilă clasificare din punctul de vedere al utilizării este :
- LFSR in-tap și
- LFSR out-tap,

primul cunoscut și ca Multiple Input Shift Register (MISR).
Alt nume utilizat este Simple Shift Generator Generator (SSRG) pentru Fibonacci și Multiple-Return Shift Register Generator (MRSRG) pentru Galois.
În continuare se prezintă o analiză a cercetărilor efectuate comparând funcționarea LFSR și MISR pentru polinoame ireductibile de grad 4, 8 și 16.
Prin utilizarea unui polinom generator de gradul n poate fi creat un câmp Galois de ordin $2^n$.
Acest polinom face parte din totalitatea de polinoame modulo 2 polinoame și este folosit ca un polinom modular in înmulțire.
Acest tip de câmp se notează ca GF ($2^n$) sau GF (n) și în cazul particular unde n = 8 se găsește algoritmul Advanced Encryption Standard (AES).
După spargerea algoritmului Data Encryption Standard de către E.F.F. (Electronic Frontier Foundation) a fost demarat un concurs internațional pentru identificarea unui nou algoritm criptografic international.
Doi criptografi din Belgia au prezentat un nou algoritm numit Rijndael ( numit astfel din anagramarea numelor lor, respectiv Joan Daemen și Vincent Rijmen).
Noul algoritm utilizează un Galois Field GF ($2^8$), cu următorul polinom generator:

$$P(x)=x^8+x^4+x^3+x+1$$

unde coeficienții polinomului de mai sus corespund polinomului '11B' în hexazecimal.Operațiile aritmetice respectă regulile de calcul ale unui Câmp Galois.
Majoritatea criptosistemelor nou apărute s-au bazat pe evoluția metodelor criptografice.
Prin generarea de secvențe pseudoaleatoare într-un câmp finit unele tehnici de criptare au obținut o evoluție remarcabilă.
Pentru proiectarea unor cifruri sigure se pot utiliza unele metode care se bazează pe registre de deplasare cu feedback.
Uneori, un alt LFSR este folosit pentru controlul pașilor (step) frecvenței primului LFSR.
Unele aplicatii sunt bazate pe generatoare tip Geffe, ceea ce înseamnă un design bazat pe funcționarea a trei LFSR cu perioade diferite.
De obicei în aplicațiile reale proiectarea (design-ul) se bazează pe combinația unor metode cunoscute.
Proprietățile unui câmp Galois au fost prezentate de Evariste Galois și au explicitat cele două operații de bază, respectiv adunarea și înmulțirea.
Pentru o sporire a securității obținute este demonstrat că polinomul generator utilizat trebuie să fie Ireductibil sau Primitiv.
În continuare sunt enumerate cele mai frecvente utilizări ale LFSR:

- Pattern Generators ( generatoare de modele) ;
- Optimized Counters ( contoare optimizate);

- Testing ;
- Encryption/Decryption data ( criptare/decriptare de date);
- BIST (Built-in Self-Test) ( autotestare);
- Data Compression ( compresie de date);
- Processing of Digital Signal( prelucrarea semnalelor digitale);
- PN ( Pseudo random Number Generation) ( generator de numere pseudo-aleatoare);
- Direct Sequence Spread Spectrum ( spectru direct de secvențe spread);
- Scrambler/Descrambler;
- Data Integrity ( integritatea datelor);
- Checksums ( sumatoare de control);
- Pseudo-Random Bit Sequences (PRBS);
- Signature Analyzer (analizor de semnale) ;
- Error Correction (corectare de erori);
- Wireless communications ( comunicații fără fir).

## 1.2. Context și Relevanță

În ciuda faptului că fundamentele științifice ale Teoriei Codurilor sunt deja cunoscute de ani de zile, este important să se prezinte cercetările mai noi din acest domeniu.
Fiind foarte ușor de implementat in hardware, LFSR-ul a devenit unul dintre cele mai populare dispozitive utilizate pentru construirea unui generator criptografic.
Conceptul de securitate a fost imbunatatit incepand cu primii algoritmi de Criptografie. Pentru menținerea unui nivel ridicat al serviciilor de securitate este necesară utilizarea unei surse verificate de date aleatoare. Aguirre, Alvarez, Tortosa și Zamora au realizat un generator de numere pseudoaleatoare folosind Packed Matrices (Matrici împachetate).
Generatorul descris în lucrarea menționată se bazează pe puterea unui bloc de matrice superior triunghiulară (BUTM). Eficiența folosirii de așa numite "packed matrices" provine de la efectuarea operațiilor binare între registre ale procesorului.
Acest BUTM poate fi de asemenea utilizat pentru integrarea unui nucleu de securitate cu multe aplicații pentru soluții de preț scăzut/putere scăzută ( low cost / low power).
Anumite tehnici noi pentru a asigura comunicații sigure pentru dispozitive de putere mică au fost experimentate și prezentate în cadrul primei Conferințe având titulatura de First Conference on Adaptive Hardware and Systems NASA/ESA de către Howells, Papoutsis și McDonald-Maier.
Codarea este utilizată pentru criptare în securitate, în compresia codurilor sursă și în detecția și corecția de coduri ăn Channel Coding.
Channel coding utilizează două abordări:

- **Forward error coding (FEC);**

Pentru îmbunătățirea performanței sunt utilizate rectificările anterioare ale codurilor de eroare și măsurarea ratei de eroare de bit (BER bit error rate). BER este de asemenea eficientă pentru evaluarea codurilor convoluționale prin simularea tipurilor de erori.

- **Automatic repeat request (ARQ);**

Solicitarea automată repetată (ARQ) se concentrează pe retransmiterea către destinație a pachetelor nerecepționate.
Cu privire la FEC pot fi utilizate două tipuri de codare:

- Block coding (de obicei, dezvoltat prin codificare ciclică )
- Convolutional coding.

Una dintre cele mai importante prezentări ale informațiilor de bază privind codurile convoluționale se face de către M. Bossert.
O metodă modernă în codare și decodare este utilizarea codului convoluțional.
O soluție specială de obținere a acestui algoritm este Viterbi, cunoscut de asemenea ca un algoritm de maxim likelihood-decoding.
Modelul propus pentru decodor utilizează VHDL. Decodarea tip Viterbi motivează utilizarea VHDL pentru modelare în scopul obținerii unei clasificări a disturbanței calității puterii ( Classification of Power Quality Disturbance).
Codul produs este mai puternic atunci când lungimea utilizată este mai mare.
Algoritmul Viterbi este o soluție optimă pentru decodificarea codurilor convoluționale când este folosită o decodare de tip decizie soft și poate produce performanțe mult mai bune decât codurile de tip BCH ( Bose-Chaudhury-Hocquenghem).
Algoritmul Viterbi este un decodor de tip maximum likelihood astfel încât cuvântul de cod de ieșire este întotdeauna cel cu cea mai mare probabilitate de a fi cuvântul corect transmis de la sursă. Decodarea utilizând algoritmul Viterbi poate produce o reducere semnificativă a puterii prin exploatarea variației în timp real a caracteristicilor sistemului.
Tsui a realizat câteva implementări pentru decodificatoare de tip Viterbi Low Power și a dovedit că ele sunt foarte utile în aplicații de tip high Throughput Wireless Applications.
Reprezentările de bază utilizate pentru a descrie structurile de codare pentru codurile convoluționale sunt:

- Generator Representation
- Tree Diagram Representation
- State Diagram Representation
- Trellis Diagram Representation.

Codurile Trellis sunt frecvent utilizate deoarece acestea oferă posibilități bune de corectare a erorilor. O diagramă Trellis poate fi utilizată pentru calcularea distanței libere a unui cod convoluțional ( free distance). Există posibilitatea de a lucra cu o schemă de modulație codată Trellis.

Pentru decodare există două metode :

- Decizie Hard și
- Decizie Soft.

Deciziile Hard utilizează cuantizarea tip 1-bit pentru valorile recepționate și deciziile Soft pe cea tip multi-bit.
Hernandez arată modul de utilizare a codurilor convoluționale și decodarea de tip Decizie Soft pentru a îmbunătăți ascunderea datelor.

Alte implementări în care apar Codurile Convoluționale Parțial Concatenate (Partially Concatenated Convolutional Codes -PTCCC) sunt utilizate pentru decodare.
O altă posibilă clasă în codurile convoluționale este clasa de Coduri Convoluționale de Date Paralele (Parallel Data Convolutional Codes -PDCC) utili într-o evaluare completă a Codurilor Convoluționale (Gadkari și Rose). Există un design deosebit oferit de Codurile Convoluționale Concatenate Parale (Parallel Concatenated Convolutional Codes -PCCC) și unele simulări care compară performanța celor de tip PDCC cu cele de tip PCCC.
Există clase speciale de coduri convoluționale cu o distanță maximă separabilă (Maximum Distance Separable –MDS) sau având un Profil Distanță Maximă (Maximum Distance Profile -MDP). Clasele de tip MDS au proprietatea că distanțele coloanelor ating banda de generalizare Singleton în cel mai scurt posibil pas de timp.
Yamanoyo și Fujiwara au dezvoltat un algoritm pentru funcții de transfer a codurilor convoluționale.
Codurile Turbo sunt o nouă clasă de coduri convoluționale cu performanțe în Bit Error Rate (Rată de biți de eroare) apropiate de limita Shannon.
Levannier și Bailly în Transaction on Communications au stipulat că pentru codurile Turbo nu au fost încă reliazte concatenări cu coduri de corectare a erorilor Reed-Solomon. Ele pot oferi un câștig semnificativ asupra schemelor clasice de codificare de corecție de eroare de tip forward.
O problemă specială dezvoltată în Transactions on Information Theory tratează Decodificarea Turbo pe Binary Erasure Channel. Datorită performanțelor de capacitate și a complexității decodării de tip low, Codurile Turbo au dobândit o importanță considerabilă din 1993, când Berrou le-a introdus.
Un alt caz compară funcționarea Decodării utilizând Coduri Convoluționale și Coduri Turbo Decodarea într-un Mediu Software Definit Radio (Software Defined Radio - SDR).
Un loc aparte în proiectarea codurilor Turbo face parte din codul tip Turbo-likede rată scăzută, adică ceea ce reprezintă Codurile Turbo Super-Ortogonale (Super-Orthogonal Turbo Codes -SOTC), codurile Turbo Hadamard (THC) și altele.
Un nou principiu Turbo de codificare în Codarea de Canal și în Criptografie se bazează pe concatenare sau Decriptarea de tip Soft Input cu feedbackl. Noul cod poate fi numit Cod de recunoaștere a Erorii Exterioare ( Outer Error Recognizing Code).
Automatic repeat request utilizează pentru detectarea de erori FEC.
Pentru codificarea blocului există două posibilități:
• suma oricăror două cuvinte de cod trebuie să fie un cuvânt de cod liniar
• orice șiftare ciclică a unui cuvânt de cod produce un alt cod de tip cuvânt ciclic.
Există unele importante aplicații care utilizează coduri ciclice:
• Codurile Hamming
• Codurile repetitoare
• Codurile de lungime maximă
• Bose - Chaudhuri - Hocquenghem (coduri BCH) i
• Coduri Reed-Solomon (RS) și altele.
Codurile Hamming au fost dezvoltate pe baza distanței Hamming. Numărul de poziții prin care două cuvinte binare de aceeași lungime diferă reprezintă distanța Hamming, după R. W. Hamming, care a realizat prima cercetare sistematică a codurilor de corectare a erorilor.

Unul dintre punctele slabe ale unui cod Hamming este că, în cazul în care apar două erori atunci corecția realizată de algoritm, inversând poziția corespunzătoare bitului din mesaj introduce de fapt, o a treia eroare.
Codurile BCH sunt renumite în teoria codurilor. Codurile BCH au fost inventate în 1959 de Hocquenghem, și independent în 1960 de către Bose și Ray- Chaudhuri. Din punct de vedere tehnic un cod BCH este un cod digital, pe mai multe niveluri ciclice de lungime variabilă de corectie a erorilor folosit pentru a corecta mai multe modele aleatoare de eroari. Un cod BCH este un cod generat de un polinom peste un câmp finit astfel încât polinomul generator să fie ales special. Este de asemenea un cod ciclic.
Codurile bloc care includ codurile Reed-Solomon sunt capabile să corecteze erorile care apar în burst (rafale) și de multe ori sunt folosite în sistemele codificate concatenate.
Codurile Reed-Solomon cuplate cu un sistem codificat de Space-Frequencya și codurile convoluționale măresc eficiența utilizării pentru o multitudine de tehnici.
În mediul de comunicare, există defecte fizice și interferențe de mediu care pot produce erori de bit aleatoare în timpul transmisiei de date.
Codificarea de erori este o metodă de detectare și de corectare a erorilor pentru a ne asigura că informațiile sunt transferate corect de la sursă la destinație.
Cyclic Redundancy Check este un cod de detectare a erorilor utilizat pentru identificarea unor modificări accidentale ale transferului de date de intrare. Există mai multe metode în Teoria Codurilor pentru detectarea și calcularea indicatorului de Cyclic Redundancy (CRC).
Aceste metode alătură simulările realizate în diferite limbaje de programare cu informațiile asupra utilizării de registre liniare de deplasare LFSR.
Toate registrele de tip LFSR utilizate în mod direct sunt nesigure. Prin utilizarea unor registre de deplasare cu mai multe secvențe produse, respectiv având o perioadă mai mare, acestea vor părea mai întâmplătoare și prin combinarea mai multor LFSR se poate realiza o îmbunătățire a securității de criptare obținute. Această caracteristică este utilă, de asemenea, atunci când LFSR-urile sunt utilizate în determinarea CRC-ului.
Sapir și Stein au arătat că este posibil să se implementeze funcția CRC în proiectarea unui sistem încorporat utilizând proprietăți LFSR cu un impact minim asupra performanței sau a memoriei.
Restul obținut CRC-8 a fost calculat și tabelul cu erori a fost obținut pentru sindroamele de CRC-8 . O altă metodă dezvoltată este în Teoria cuantică de corectare a erorilor.
Aplicații ale codurilor ciclice pot utiliza două tipuri de polinoame:
• polinoame cod
• polinoame generatoare.
Aceste polinoame sunt utilizate în Câmpuri Galois.
Algoritmul de cheie publică aleatorie Random Public Key este un nou algoritm de criptare implementat de o exponențializare discretă peste un câmp finit Galois Fields ($GF[2^n]$).
Unele analize cu privire la utilizarea polinoamelor ireductibile sunt dezvoltate în Identity Testing (Testarea Identității).
Fiind la limita dintre matematică și informatică, criptologia este într-o continuă și constantă dezvoltare.
Toate analizele se bazează pe funcționarea LFSR.
Pentru aceasta, există unele funcții:

• LFSR-autocorelație (L, P, K)
• LFSR-conexiune-polinomială (S)
• LFSR-secvență (key, fill, n).

Solomon Golomb a inventat o familie de coduri de compresie a datelor. Utilizarea secvențelor de comenzi produse de un registru de deplasare liniar utilizat ca generator de numere pseudo aleatoare care funcționează ca stream ciphers. Aplicațiile criptografice trebuie să îndeplinească cele trei condiții de a fi aleatorii.

Massey a realizat un registru de deplasare de sinteză și apoi a creat algoritmul Berlekamp -Massey.

Unele abordări legate de această dezvoltare în cercetare sunt utilizate in aplicații de Knowledge Engineering , cât și aplicații din Inteligenta Artificială.

Alte aplicații importante sunt realizate de către Benson și Neame în domeniul sănătății Healthcare Computing.

Byrd, Hillery și Symon au realizat un Encryptor comercial de mare viteză pentru rețele de transfer în mod asincron (Asynchronous Transfer Mode -ATM).

O cercetare specială este realizată cu privire la securitatea în rețele și toate metodele de a o obține și păstra. Pentru maximizarea securității rețelei trebuie să fie utilizate proprietățile specifice de nod.

Au existat cercetări privind securitatea rețelelor în țara noastră realizate de V.V. Patriciu si alții. Gestionarea rețelei pentru conexiuni generale a fost construită prin utilizarea unor relații între nodurile de rețea și prin codarea fiecărui nod (Tracer Ho, Medard M, Koetter R).

Au existat cercetări în domeniul bazelor de date, cel al politicii, al guvernării, al fraudelor financiare și altele. Cercetători cum ar fi Kakkar, Gunter și alții au păstrat unele concepte de securitate împotriva atacurilor din rețele active. De asemenea, au fost propuse aspecte de arhitectură din mai multe domenii de securitate Multi-Security Domain Networks cum ar fi cele pentru sectorul militar și comercial.

Confidențialitatea, integritatea și disponibilitatea sunt principalele obiective pentru serviciile de securitate.

Autenticitatea este, de asemenea, un obiectiv care trebuie atins și pentru aceasta pot fi utilizate implementări software și hardware.

Pentru creșterea securității mai multe metode au fost elaborate, acestea fiind:

Utilizarea lucrului în câmpuri Galois Domenii folosite de Daemen și Rijmen în algoritmul Rijndael (AES) sau de implementare a multiplicării modulare.

Există metode pentru a studia punctele forte și punctele slabe ale unui sistem.

Utilizarea registrelor de deplasare liniare cu feedback este bine cunoscută în Criptografie și Testare.

Registrele de deplasare tip LFSR pot produce secvențe de numere aleatoare prin care se obține securitatea în transmiterea de informații.

O mare problemă este găsirea unui registru de acest tip cât mai bun în acest sens.

Schneier arată că un registru LFSR bun trebuie să aibă la bază un polinom ireductibil.

Udar și Kagaris au propus o schemă de LFSR folosind polinoame ireductibile non-primitive.

Rezultatele lor experimentale au demonstat posibilitatea de a utiliza mecanismul obținut într-un mediu de test-per-scan.

Cercetarea realizată de ani de zile de către Philip Koopman este materializată în mai multe lucrări ştiinţifice despre alegerea celui mai bun polinomul ireductibil pentru calculul CRC-ului.

## 1.3. Obiectivele tezei

Registrele de deplasare pot fi folosite în diverse moduri şi fac parte din nucleul oricărui sistem digital. Aplicaţiile acestora au apărut în criptografie şi testare precum şi în detectarea şi corectarea erorilor şi în sistemele de wireless (comunicaţii fără fir).
Principalul obiectiv al cercetării propuse a fost cel de a analiza funcţionarea atât aregistrelor de tip LFSR (Linear Feedback Shift Register) cât şi a celor de tip MISR (Multiple Input-output Shift Register). Obiectivul principal a fost cercetarea funcţionării tuturor polinoamelor ireductibile de grad 4, 8 şi 16 în scopul găsirii celor mai potrivite alegeri pentru diferite utilizări posibile ale LFSR şi MISR.
Găsirea celor trei scheme posibile pentru implementările registrelor LFSR a constituit un alt obiectiv prin căutarea unor corelaţii între aceste rezultate ale implementărilor aceluiaşi polinom ireductibil.
Un alt obiectiv important a fost de a analiza utilizarea LFSR în Criptografie. În acest scop, două aspecte au fost analizate: funcţionarea algoritmului AES şi compararea celor trei algoritmi acreditaţi ISO / IEC.
Scopul principal pentru efectuarea studiului a fost implementarea codului original C ++, care a implicat, de asemenea, utilizarea operării în Câmpuri Galois Fields şi cercetarea implicaţiilor utilizării unui polinom ireductibil.
În acelaşi timp, registrele de deplasare au o utilizare deosebită în Testare, fapt materializat prin cele două tipuri cunoscute de implementări :Built In Test (BIT) sau Built In Self Test (BIST).
Autotestarea de tip BIST este absolut necesară în maşini complexe de toate tipurile, maşini nesupravegheate de toate tipurile, circuite integrate în armament, avionică, dispozitive medicale şi electronice auto.
Un alt obiectiv propus a fost găsirea unei metode alternative de realizarea a unui BIST.
Această metodă trebuie să evite utilizarea de algoritmi de compresie avansaţi, din cauza cărora se fac cheltuieli mari pentru implementările Hardware. Pentru toate experimentele a fost necesară crearea şi utilizarea unui software special.
Cele mai multe dintre programe au fost concepute in limbajul C ++.
Modelul propus pentru implementarea BIST-ului a fost modelat în VHDL şi sintetizat cu Xilinx Synthesis Tool (XST) pentru Xilinx Virtex-4 FPGA (Field Programmable Gate Array).
Pentru verificarea rezultatelor a fost folosit Matlab (Anexa B4).
Un alt obiectiv propus cel de a găsi o metodă rapidă de calcul a Cyclic Redundancy Code , care este utilizat în Detectarea şi Corectarea Codurilor de Eroare.
Pentru asigurarea corectitudinii datelor recepţionate şi stocate , CRC este utilizat pentru toate tipurile de comunicaţii.

# 2. Codare, Testare și Fundamente matematice

Teoria Codării reprezintă una dintre cele mai importante domenii ale utilizării câmpurilor finite.
Această teorie își are originea într-o teoremă celebră a lui Shannon. Această teoremă a răspuns la întrebarea despre cum se poate realiza protecția la perturbații în transmiterea sau stocarea informației.
C.E. Shannon în lucrarea ”A mathematical theory of communication” (1948) a publicat și prezentat ceea ce a devenit cunoscut ca teorema a II-a a lui Shannon sau teorema codării canalelor cu perturbații. Astfel, atunci când informația este transmisă sau stocată prin canale cu perturbații sau medii de stocare semnalele recepționate pot fi alterate din cauza zgomotului și așa se impune luarea de măsuri pentru reducerea acestora. Prin utilizarea tot mai intensă a rețelelor mari naționale și internaționale pentru transmiterea datelor de mare viteză s-a intensificat problema protecției la erori. În acest sens se garantează existența unor coduri care pot transmite informații la viteze apropiate de capacitate cu probabilitate mică de eroare. Scopul principal în detectarea și corectarea erorii este de utilizarea și dezvoltarea de astfel de coduri.
In ultimii ani mai multe instrumente și mai abstracte au fost folosite, cum ar fi teoria câmpurilor finite și teoria polinoamelor peste corpuri finite.
Ambele, Codarea și Testarea cu circuite IC digitale (Integrated Circuits -circuite integrate) se bazează pe utilizarea registrelor de deplasare liniare tip LFSR. Dezvoltarea Teoriei Codurilor a condus la crearea de aplicatii de succes în Criptografie, Error Detecting and/or Correcting and in Wireless Communication Systems.
În urma efectuării unei analize laborioase s-a reușit o sistematizare a informațiilor, care s-a constituit în schemele cu privire la taxonomia de codificare și testare IC digitale , precum și la taxonomia testării.

# 3. Analiza funcționării LFSR și MISR

În acest capitol a fost prezentată o analiză a utilizării implementărilor posibile pentru realizarea unor registre de deplasare de tip liniar și multiplu, respectiv LFSR și MISR.
După o scurtă prezentare a LFSR au fost descrise formele cunoscute ca Fibonacci si Galois pentru implementari.
În același timp, unele informații generale au fost precizate cu privire la seed, la numărul de secvențe pseudo aleatoare care se pot obține precum și cele două aspecte importante:
• Utilizarea Polinoamelor Ireductibile;
• Lucrul într-un Câmp Galois GF ($2^n$)..
La început am efectuat un studiu al tuturor polinoamelor ireductibile și primitive de gradul 4. Am verificat relațiile prin care se obține fiecare pondere în parte.
Pe lângă implementările cunoscute a fost analizată și o a treia variantă de implementare a aceluiaș polinom. Au fost verificate legăturile între rezultatele obținute prin cele trei implementări posibile diferite. Acest tip de verificare a condus la obținerea unei relații de corelare între rezultatele furnizate de cele trei scheme.

De asemenea a fost realizată o corecție în formulele de calcul pentru ponderi la polinoame de grad mai mare ca 4.
Toate formulele corectate au fost testate pentru tote cele 30 de polinoame ireductibile de grad 8 și concluzia este ca relațiile obținute sunt corecte.
Pentru a păstra o bună securitate care să confirme observațiile făcute de Bruce Schneier, toate polinoamele utilizate sunt fie primitive, fie ireductibile.
Orice registru de deplasare are anumite utilizări practice precum conversia între date paralele și date seriale și întârzierea unui flux continuu de biți în serie.
Numărul total de secvențe aleatoare generate depinde de polinomul utilizat.
Generatoarele de secvențe Pseudo aleatoare realizate utilizând registre de deplasare de tip LFSR reprezintă elemente de bază pentru diferite aplicații în criptografie și, de asemenea, în canalele de comunicare pentru proiectarea de dispozitive de codificare și de decodificare.
Acest tip de analiză poate fi făcută din punct de vedere hardware sau software. Unele dintre lucrările de cercetare realizate de mine prezintă experimente făcute prin utilizarea unor programe de simulare .Această preocupare cu privire la variante de implementare pentru polinoame de grad 8, 16 și 32 am regasit-o și la Panda A.K.,Rajput P.,Shukla care au realizat în 2012 o cercetare în acest sens utilizând FPGA pentru simulări.
Este nevoie de un efort mare pentru a alege polinoamele ireductibile care dau rezultate mai bune în aplicații criptografice, astfel încât există multiple preocupări în acest sens în special pentru gradul 8 și 16, pentru 32 devenind imposibila o astfel de încercare.
Probleme similare celor prezentate mai sus sunt conținute și analizate în lucrarea mea prezentată în cadrul Conferinței Internaționale Comunicare Europment, Signal Processing and Computers din Interlaken, Elveția, în 2014 și în 2015 la Conferința INASE de la Viena, precum și în versiunile lor extinse din reviste.
Analizele comparative cu privire la gradul 16 s-au realizat utilizând polinoame primitive alese dintre cele publicate in Peterson Tables și unele comune pentru a putea face diferențierea comportamentului lor. Există un mare interes în alegerea unor polinoame "bune", dar vorbind de gradul al 16-lea, aceasta a devenit un obiectiv foarte greu de atins prin numărul foarte mare al polinoamelor existente. Din compararea timpilor de rulare obținuți și analiza graficelor realizate se pot trage anumite concluzii.
Din toate graficele prezentate se poate concluziona că pentru date de intrare având lungimi de 1000 biți cel mai bun comportament în observăm la polinomul P1 și cel mai de durată la polinomul P9.
Toată această analiză a fost precedată de o alta, similară realizată cu toate polinoamele ireductibile de grad 8 ( determinate de mine printr-un program și fiind în număr de 30).
Concluziile și, de asemenea, contribuțiile au fost:
• LFSR și MISR îndeplinesc aceeași funcționalitate;
• Începând cu gradul 8 toate polinoamele ireductibile prezintă expresiile de calcul pentru ponderi schimbate față de cele pentru gradul 4 (apare o corecție dată de legăturile existente anterior);
• Există o corelație între rezultatele obținute folosind cele trei scheme de implementare posibile;
• S-au realizat programe de simulare pentru registre de tip LFSR și MISR bazate pe polinoame de grad 4, 8 și 16;

• Pentru verificarea calculelor matematice au fost concepute programe de calcul ( de exemplu pentru verificarea restului împărțirii);
• Pentru așa numita alegere a polinoamelor s-au făcut rulări cu lungimi diferite ale datelor de intrare în cazul polinoamelor de grad 8 și 16.

# 4. Aplicații Criptografice

În acest capitol a fost prezentată o scurtă incursiune în tipurile existente de cifruri iar apoi s-a realizat o analiză comparativă a celor trei algoritmi de criptare AES, Camellia și SEED.
Un obiectiv important este acela de a cunoaște exact ce algoritm este mai eficient, în funcție de mărimea fișierului de criptare.
Am prezentat utilizarea registrului de tip LFSR pentru generarea unei secvențe pseudo-aleatoare necesară în creșterea dificultății de criptare.
Un rol important în această comparație efectuată a fost cel de a analiza fiecare dintre cei trei algoritmi comparați în parte. O parte principală a fost analiza algoritmului AES (Advanced algoritm de criptare) și pentru acest lucru a fost creat și folosit un program în C ++. Algoritmul AES (Rijndael) are posibilitatea de a fi utilizat cu trei lungimi de cheie posibile (128 biți, 192 biți și 256 biți), asigură o securitate foarte mare și software și poate fi foarte rapid implemantat atât hard cât și soft.
De-a lungul timpului s-au făcut mai multe tipuri de comparații între algoritmi. Aceste comparații se concentrează pe mai multe criterii , cum ar fi:

- Securitate;
- Performanțele hardware și software;
- Rezistența din mai multe puncte de vedere;
- Potrivire în utilizare pentru spații limitate;
- Găsirea și utilizarea unei metodologii de evaluare a costurilor de calcul și complexitatea diferitelor blocuri de cifru pentru a fi independente de platformă. Această metodologie reușește să reducă decalajul dintre implementarea algoritmilor și a studiilor matematice.

Ideea principală a fost să se ia în considerare numai valoarea operațiunilor necesare, reducând toate transformările la byte wise-AND și byte wise-OR și șiftări.
Pentru fiecare dintre algoritmii analizați a fost calculat costul de implementare.
Implementarea software a algoritmilor de criptare care utilizează același procesor a fost un alt tip de analiză și un alt tip de comparație
Standardele ISO pentru Block Ciphers au fost comparate ținând cont de performanța lor din punct de vedere ASIC (Application Specific Integrated Circuit).
Pentru această comparație ideea de bază a fost de a cerceta eficiența tuturor algoritmilor cunoscuți avân Standardul ISO în funcșie de implementarea pentru S-Box.
O altă comparație pentru Block Ciphers a fost realizată în funcție de performanța hardware. După o descriere generală hardware pentru fiecare dintre algoritmii luați în discuție pentru comparare s-au propus și realizat realizat algoritmi compacți și arhitectură hardware de mare viteză.

Toţi algoritmii au obţinut performanţe similare în implementări compacte. De asemenea, s-a dovedit că invertorul $GF(((2)^2)^2)^2$ este mai mic decât $GF((2^4)^2)$ cu 26%.
O cercetare similară a fost realizată de o echipă din India, în cadrul unui proiect în cadrul Departamentului de Inginerie Informatică şi Tehnologia Informaţiei, Colegiul de Inginerie din Pune, India.
Rezultatele obţinute în urma acestei cercetări au fost prezentate într-un document publicat in International Journal of Network Security & Aplicaţii sale (IJNSA), în luna iulie a anului trecut.
Diverse caracteristici ale fişierelor, cum ar fi: densitatea de date, tipuri de date, dimensiunea cheilor şi dimensiunea datelor au fost analizate folosind diferiţi algoritmi de chei simetrice. Rezultatele obţinute au ajuns la concluzia că mărimea datelor şi timpul de criptare sunt proporţională între ele.
În acelaşi timp criptarea depinde numai de dimensiunea fişierului, nu de tipul de date sau de densitatea acestora.
În acest capitol am adus unele contribuţii originale:
- Am implementat algoritmul AES prin crearea unui program original în C ++.
- Am făcut o comparaţie între cele trei cifruri specificate ISO / IEC 18033-3-2010, 128-bit bloc: AES, Camellia şi SEED.

După cercetări au fost prezentate măsurători ale unor aspecte practice cum ar fi timpii, lungimile datelor de intrare, lungimile cheilor utilizate.

# 5. Proiectarea unui BIST utilizând LFSR

Toate testele de fiabilitate moderne pot fi reprezentate prin auto-testele încorporate.
Aplicaţiile lor pornesc de la criptografie şi măsurători ale BER până la sisteme de comunicaţii wireless care utilizează tehnici de acces multiplu de tip spread sau de divizare de cod. Constrângerile de timp limitează complexitatea testelor astfel încât există mai multe metode de compresie, prin intermediul unui LFSR paralel (Linear Feedback Shift Register) utilizat ca analizor de semnătură.
Scopul acestui capitol a fost de a prezenta aspectele întâlnite în cadrul problemelor legate de autotestare şi de a propune un mediu comun pentru identificarea de pattern resistant logic precum şi asigurarea unei toleranţe mai stabile şi sigure prin utilizarea unei memorii ROM (Read Only Memory) pe post de tabelă de tip Lookup (LUT).
În acest capitol s-a demonstrat că metoda alternativă propusă pentru realizarea BIST-ului utilizează foarte puţin hardware şi are un cost mic , având de asemenea dimensiuni reduse. Un aspect important este utilizarea tabelei LUT (Look-Up Table), care se poate realiza în diferite implementări.
Schema de îmbunătăţire propusă conţine pentru funcţia de tabel LUT o memorie ROM pe 64 de biţi. Este posibilă utilizarea în locul acesteia a unei memorii RAM sau EEPROM.

De asemenea, ar trebui să fie făcute comparații cu memorii de tip SRAM și DRAM . Toate aceste teste trebuie să fie făcute având în vedere performanțele obținute în funcție de timpul de acces de pe LUT.
Metodele alternative propuse pentru punerea în aplicare a BIST (construit în Auto Test) evită utilizarea de algoritmi de compresie avansați și necesită un hard având costuri mici și dimensiuni reduse. Toată această cercetare este prezentată într-o lucrare prezentată la Conference on Computers din Rhodos în 2009.
Cercetări similare au prezentat posibilitatea de a crește eficiența codificărilor prin utilizarea așa numitelor Reseedings bazate pe teste de compresie.
De asemenea pentru optimizarea funcționării unui BIST s-au folosit teste de compresie de tip înalt ( High Test Compression).
Metoda de Reseeding poate fi dezvoltată pentru BIST în cazul circuitelor folosind registre LFSR de tip Multiple - Polynomial.

# 6. Coduri Detectoare și Corectoare de Erori

Un rol special în lupta împotriva corupției de date este ocupat de Cyclic Redundancy Code (CRC).
În detectarea unui proces de corupție de date este necesară calcularea și utilizarea acestui cod ciclic de redundanță CRC.
Din punct de vedere matematic, calculele necesare obținerii unui CRC sunt cele de împărțire a polinoamelor și cele din aritmetica câmpurilor de numere întregi MOD2.
Acest capitol a explorat diferite opțiuni de posibile implementări pentru calcularea unuia dintre codurile CRC cele mai utilizate pe scară largă. O analiză completă a tuturor miliardelor de posibile polinoame având 32 de biți este o sarcină grea chiar și pentru enumerarea lor și chiar folosind toate tehnicile de filtrare si de calcul existente.
Aproape toate polinoame CRC utilizate in mod obișnuit asigură in mod semnificativ o mai mică capacitate de detectare a erorilor decât ar putea. De asemenea, analiza arată că aceste polinoame reprezintă doar niște alegeri bune pentru anumite lungimi de mesaje.
Pentru un CRC, calculul distanței Hamming (HD) depinde de polinoamele generator de utilizat, de lungimea cuvântului de date și de lungimea Frame Check Sequence (FCS).
Există posibiltatea alegerii de variante de implementare atât Software , cât și Hardware , precum și posibilități de a fi utilizate cu diferite tipuri de procesoare.
Punctul de vedere este propus este acela de a alege cea mai rapidă metodă de calcul CRC, care în urma analizei efectuate s-a dovedit a fi metoda Bit-at-a-time.

# 7. Concluzii

## 7.1. Sumar

Această teză are șase capitole principale, fiecare dintre ele corespunzând unei cercetări specifice.
Primul capitol conține o prezentare generală a principalelor informații necesare pentru a fi cunoscute pentru subiectele prezentate în următoarele trei capitole.
Un aspect important prezentat în capitolul doi constă în menționarea multiplelor utilizări ale unui registru de deplasare liniar LFSR (Linear Feedback Shift Register -LFSR), precum și cele mai relevante operații pentru a calcula într-un câmp Galois. Deja în acest capitol există mai multe aspecte privind subiectul experimentelor realizate și prezentate în capitolele următoare.
Capitolul trei cuprinde toate cele trei analize de bază diferite care se axează pe utilizarea de registre de deplasare de tip LFSR și MISR pentru polinoame ireductibile de grad 4, 8 și 16.
Principalele contribuții sunt prezentate ca și concluzii la toate experimentele.
Aspectele folosirii unui LFSR sau MISR pentru un polinom ireductibil de grad 4 au fost prezentate într-o lucrare publicată în cadrul Conferinței ISPRA în anul 2009, la Cambridge.
Partea din capitolul trei cu privire la analiza regsitrelor de deplsare LFSR și MISR pentru polinoame ireductibile de gradul 8 este subiectul unei alte lucrărisusținutela Conferințade Simulare și Modelare SMO din Santander, în 2008.
O analiză completă, care pornind de la analizele mai sus menționate a dezvoltat cercetarea prin adăugarea unei analize suplimentare cu privire la polinoame ireductibile de grad 16, a fost publicată în journal-ul Transaction on Computers.
Capitolul patru conține informații generale legate de criptare iar apoi se focusează pe trei algoritmi de criptare care corespund standardelor ISO/IEC actuale.
Astfel s-a realizat o prezentare generală a celor trei algoritmi AES, Camellia și SEED.
Un accent deosebit s-a pus pe analiza algoritmului AES ( Advanced Encryption Standard), care a fost implementat printr-un program în C++ și a fost testat pentru diferite secvențe de intrare pseudo-aleatoare.
Principiile efectuării de calcule într-un câmp Galois au fost studiate în algoritmul (Rijndael) AES.
A fost realizată o comparație între cele trei ISO / IEC cifruri bazate pe blocuri standard.
Pentru realizarea acestui deziderat au fost utilizate un număr de 20 de fișiere conținând secvențe diferite de intrare pseudo-aleatoare.
Scopul acestei cercetări a fost alegerea cât mai potrivită a algoritmului utilizat în funcție de dimensiunea de fișierelor care urmează să fie criptate.
În continuare, în capitolul cinci, este cuprinsă o altă parte a cercetării, care conține aspecte ale proiectării BIST folosind LFSR.
Schema propusă de mine a fost analizată din mai multe puncte de vedere.
Schema de îmbunătățire propusă a fost descrisă și evaluată cu prezentarea unor detalii privind implementarea hardware-ului.

Toate aceste cercetări au fost prezentate într-o lucrare ştiinţifică susţinută laInternational Conference on Computers din Rodos.
Lucrarea extinsă a fost publicată în journal-ul Transactions on Electronics.

## 7.2. Concluzii

Această secţiune prezintă contribuţiile acestei teze obţinute prin analiza de cercetare avînd ca scop funcţionarea registrelor de deplasare LFSR pentru polinoame ireductibile şi primitive.
Obiectivele tezei au fost precizate în secţiunea 1.2. din capitolul introductiv.
În ceea ce priveşte analiza LFSR şi MISR folosind polinoame ireductibile contribuţia principală este noua formula de calcul a ponderilor pentru grade mai mari decât 4.
Analiza registrelor de deplasare pentru polinoame ireductibile de grad 4, 8 şi 16 s-a bazat pe faptul că în permanenţă rezultatele obţinute au fost verificate prin câte trei modalităţi diferite.
În cazul folosirii de registre de deplasare LFSR, viteza creşte foarte mult şi este bine cunoscut faptul că atunci când se utilizează registre în criptografie un obiectiv important este creşterea vitezei de calcul.
Este demonstrat matematic că pentru creşterea securităţii polinoamele utilizate trebuie să fie polinoame ireductibile sau primitive.
Formula despre care s-a precizat mai sus a fost testată pentru toate situaţiile , respectiv pentru toate polinoamele ireductibile de grad 8 şi 16, iar concluzia finală a fost că relaţiile matematice sunt corecte.
Această formulă conţine nu numai puterea corespunzătoare ponderii calculate pentru x, dar, de asemenea, nişte coeficienţi suplimentari, care pot fi explicaţi ca nişte corecţii produse de conexiunile precedente celei la care ne referim.
De exemplu, în capitolul al treilea formulele pentru toate ponderile au fost:

$$S_0=1*P(x)$$
$$S_1=x*P(x)$$
$$S_2=x^2*P(x)$$
$$S_3=(x^3+x)*P(x)$$
$$S_4=(x^4+x^2+x)*P(x)$$
$$S_5=(x^5+x^3+x^2)\ *P(x)$$
$$S_6=(x^6+x^4+x^3+x)*P(x)$$
$$S_7=(x^7+x^5+x^4+x^2)*P(x)$$

Această formulă a fost obţinută empiric, prin compararea rezultatelor furnizate de cele trei programe de simulare.
O altă analiză s-a axat pe un studiu comparativ al diferitelor tipuri de implementări pentru un registru de deplasare LFSR. La baza acestor cercetări se află cele trei scheme posibile pentru implementarea unui LFSR.
Analiza a fost făcută având ca şi caz de pornire funcţionarea LFSR pentru un polinom ireductibil de gradul 4 şi anume $x^4+x+1$.

Toate rezultatele obținute prin acest experiment dovedesc formula următoare:

$$x^4 * S_A = S_B * G(x) + S_C$$

Această formulă de corelare a fost verificată, de asemenea, pentru toate celelalte șase polinoame ireductibile de grad 4.
De menționat că în formula de mai sus G reprezintă polinomul generator, iar secvențele corespund fiecare câte unei implementări posibile , respectiv pentru cele trei scheme numite A,B și C. Puterea lui x din formulă este egală cu gradul polinomului ireductibil numit G (x).
Toate cele trei scheme sunt implementări ale aceluiaș polinom ireductibil.
Secvențele obținute folosind schema A, Schema B sau Schema C pentru acelaș polinom ireductibil au aceeași lungime, dar sunt diferite.
Având în vedere necesitatea de a implementa hardware una dintre aceste trei scheme și cunoscând formula de corelare între cele trei posibilități se va putea alege cea mai convenabilă scopului propus.
O analiză a tuturor polinoamelor ireductibile de grad 8 a fost materilizată într-o lucrare, care a fost prezentată la Conferința Internațională Europment de la Interlaken, Elveția, 2014 și apoi a fost întocmită o lucrare extinsă, care a fost publicată in International Journal of Computers, volumul 8, 2014.
Un alt aspect al cercetărilor ține cont de dimensiunea fișierului de intrare și este reflectat într-o altă lucrare prezentată la Conferința Internațională INASE de la Viena din martie 2015.
În capitolul patru au fost prezentate câteva informații generale din criptografie. Apoi s-a prezentat o analiză a algoritmului Rijndael ( AES) efectuată cu ajutorul unui program in C ++ realizat special pentru acest scop. Principiile de calcul care funcționează într-un câmp Galois au fost analizate și s-a constatat creșterea securității în această situație.
Capitolul cinci a avut ca scop proiectarea un BIST folosind LFSR. Pentru a valida punerea în aplicare propusă pentru BIST a fost realizat un program de simulare în VHDL (Anexa B).
În zilele noastre, tendința este de a crea mai mulți algoritmi care să utilizeze metode performante de compresie.
Alternativa propusă pentru implementarea BIST evită utilizarea unor astfel de algoritmi de compresie avansați.
Sunt cunoscute dezavantajele algoritmilor de compresie ca fiind:
• Investiții mari în hardware;
• Analizoare de semnal paralel costisitoare și puternice;
• Dimensiunei mari;
• Cost ridicat.

Prima idee a fost aceea de a utiliza un tabel de look-up convențional pentru a injecta cazuri de testare pentru pattern resistant logic iar apoi de a folosi generatorul de numere pseudo-aleatoare pentru a continua activitatea. Cât de mare este impactul asupra timpului procesului de autotestare BIST nu este ușor de decis, cu toate acestea, în ceea ce

privește costul și suprafața folosită se poate spune că rezultatul obținut este echitabil inverstiției făcute, pentru că algoritmii de compresie la nivel înalt necesită investiții de hardware masive și duc la o mărire considerabilă a suprafeței.
În cazul BIST-ului hibrid folosirea tehnologiilor mai sus menționate începe să dea roade. Această variantă de abordare propusă pentru LUT poate garanta o creștere a toleranței la erori în cazul în care timpul și dimensiunea suprafeței de cip sunt determinante.
Metoda propusă reduce complicațiile pentru implementare și costul de producție și sporește reutilizabilitatea.
În acest mod se obțin avantaje evidente și o prelungire a duratei de viață a chip-uri VLSI

## 7.3. Contribuții

Această teză aduce următoarele contribuții principale enumerate mai jos:

1. Am efectuat o analiză și a făcut o prezentare grafică originală pentru domeniile de codare, testare, și detectarea și corectarea codurilor de eroare.

2. Am demonstrat o formulă de corelație între rezultatele obținute din cele trei tipuri de scheme de punere în aplicare pentru LFSR pentru polinoame ireductibile de grad 4.

$$\mathbf{x^4 * S_A = S_B * G(x) + S_C}$$

unde $S_A$, $S_B$, $S_C$ reprezintă rezultatele obținute pentru cele trei tipuri de scheme utilizate ca implemantări și G(x) este polinomul ireductibil folosit, adică
$x^4 + x + 1$.
De specificat că formula se păstrează și a fost verificată pentru toate polinoamele ireductibile .
3. Am realizat mai multe programe de simulare in C ++ pentru analizarea registrelor de tip LFSR și MISR.
Cu ajutorul programelor de simulare, s-au realizat mai multe experimente care au dovedit că registrele LFSR și MISR au aceleaşi rezultate.

4. Am verificat functionarea LFSR pentru toate polinoame ireductibile de grad 8. Experimentul s-a efectuat pornind de la utilizarea polinomului ireductibil folosit în AES:

$$\mathbf{G(x)=x^8+x^4+x^3+x+1}$$

O formulă a fost obținută pentru fiecare pondere, formulă care diferă de cea pentru gradul 4 prin apariția unor așa-numite corecții.
Aceste expresii determinate empiric pentru polinomul utilizat de AES (Rijndael) sunt:

$$S_0=1 *P(x)$$
$$S_1=x *P(x)$$
$$S_2=x^2 *P(x)$$
$$S_3=x^3 *P(x)$$
$$S_4=x^4 *P(x)$$
$$S_5=(x^5+x)* P(x)$$
$$S_6=(x^6+x^2+x)* P(x)$$
$$S_7=(x^7+x^3+x^2)* P(x)$$

5. Am comparat comportamentul tuturor polinoamelor ireductibile de grad 8 (în număr de 30) în funcție de numărul de biți de intrare și de timp iar rezultatele au stabilit care polinoame sunt preferabil să fie alese atunci când se dorește obținerea unui timp mai scurt.

6. Am realizat o selectare a polinoamelor cel mai rapid primitive de gradul al 16-lea pentru diferitele dimensiuni ale fișierelor de intrare , experiment ce a fost prezentat într-o lucrare în cadrul Conferinței Internaționale INASE de la Viena, anul trecut, lucrare calificată "best papers".

7. Am realizat un program în C ++ pentru implementarea algoritmului AES și am realizat experimente cu utilizarea a diferite secvențe pseudoaleatoare de date de intrare și cu schimbarea polinomului generator.

8. Am făcut o comparație între cele mai importante trei ISO / IEC Standard Cifruri Block Standard: AES, Camellia și SEED. Această comparație a celor trei algoritmi este originală și în contextul general al cercetării în domeniu aduce informații foarte utile privind posibila alegere a algoritmului de utilizat în funcție de mărimea fișierului de criptare.

9. Am propus o variantă de realizare pentru BIST folosind un minim de hardware prin utilizarea unei memorii ROM. Pentru aceasta s-a utilizat o simulare VHDL pentru varianta de BIST propusă. Toate aspectele acestei cercetări sunt prezentate într-o lucrare susținută la International Conference on Computers, Rhodes, Grecia.

10. Am propus o metodă de punere în aplicare pentru BIST, fără a utiliza algoritmi de compresie avansați.
Această metodă a fost inițial simulată într-un program VHDL. Au fost semnalate avantaje prin scăderea costului implementării.

11. Am realizat trei programe pentru trei posibile metode de calcul pentru CRC ( Cyclic Redundancy Check) și în urma cercetării efectuate am ales cea mai rapidă metodă.

Rezultatele obținute din această teză au fost prezentate în unele conferințe și publicate în mai multe reviste.

**Proceedings of International Conferences (ISI)**

1. M. A. Mioc - Simulation study of the functioning of LFSR for grade 4 irreducible polynomials, Proceedings 8th WSEAS Int. Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '09); Proceedings of the 8th WSEAS Int. Conference on SOFTWARE ENGINEERING, PARALLEL and DISTRIBUTED SYSTEMS Cambridge, UK, February 21-23, 2009, pp.27-32 (ISI Proceedings WSEAS); ISBN:978-960-474-052-9, ISSN:1790-5117
2. M. A. Mioc - Study of using Shift Registers in Cryptosystems for Grade 8 Irreducible Polynomials, Proceedings 8th WSEAS International Conference on Simulation, modelling and optimization(SMO 08); Santander, Spain, September 23-25, 2008, pp. 148-152 (ISI Proceedings, WSEAS); ISBN: 978-960-474-007-9, ISSN: 1790-5119
3. M. A. Mioc - Some aspects regarding pattern resistant logic, Proceedings 13th WSEAS International Conference on Computers; Rhodes Island, Greece, July 23-25,2009, pp. 514-519 (ISI Proceedings WSEAS); ISBN: 978-960-474-099-4, ISSN: 1790-5109
4. M. A. Mioc - Forms, solutions and effects regarding pattern resistant logic, Transaction on Electronics; WSEAS TRANSACTIONS on ELECTRONICS Issue 1, Volume 6, January 2009; ISSN: 1109-9445
5. M. A. Mioc, M. Stratulat - Software simulation of a random navigation through web graph; Proceedings of SMART Conference 2014, Timisoara, Sept.18-21, 2014; ISSN 1584-4048
6. M. A. Mioc, S. G. Pentiuc - Study of a Random Navigation on the Web Using Software Simulation; BRAIN 2015 - ISSUES 1 & 2 (SEPT. 2015) VOL 6, NO 1-2 (2015); ISSN 2067-3957
7. M. A. Mioc, M. Stratulat - Some aspects of using Shift Registers based on 8th degree irreducible polynomials; Proceedings of the International Conference Communication, Signal Processing and Computers, Eurpoment Conferences, Interlaken, Switzerland, February 22-24, 2014; ISBN: 978-1-61804-219-4; ISI Conference to be indexed.
8. M. A. Mioc - Simulation study of using shift registers based on 16-th Degree Primitive Polynomials; Proceedings of INASE Conference 2015, Wien, March 15-17, New Developments in Pure and Applied Mathematics Pp 363-369; ISBN: 978-1-61804-287-3; ISI Conference to be indexed

**Proceedings of International Conferences (Google Scholar Indexed)**

**1.** M. A. Mioc – Advances in Data Networks, Communications, Computers and Materials Analyze of common CRCs functioning using a software implementation,Sliema,2012, ISBN: 978-1-61804-118-0

**Journals (CNCSIS B/B+)**

1. M. A. Mioc - Synoptic of software implementation for shift registers based on 16th Degree Primitive Polynomials; International Journal of Computers and Communications; ID: 2012-150
2. M. A. Mioc, S. G. Pentiuc - Comparison Between AES, Camellia and SEED; JMEST (Journal of Multidisciplinary Engineering Science and Technology) Vol.2 – Issue 12 (December– 2015); ISSN 2458-9403
3. M. A. Mioc, S. G. Pentiuc - Comparison of Basic, Bit–at-a-time, and Lookup CRC-32; JSAER (Journal of Scientific and Engineering Research) Volume 3 Issue 1 2016; ISSN 2394-2630
4. M. A. Mioc - An analyze of functioning for a linear feed-back shift register and a multiple input-output shift register; Buletinul Ştiinţific al Universităţii „Politehnica" din Timişoara, Seria ELECTRONICĂ şi TELECOMUNICAŢII, Transactions on electronics and communications; Fascicola 2 Tom 50(64)
5. M. A. Mioc - A complete analyze of using Shift Registers in Cryptosystems for Grade 4, 8 and 16 Irreducible Polynomials; WSEAS Transactions on Computers, Volume 7, Issue 10, October, 2008; ISSN:1109-2750
6. M. A. Mioc - Using of Shift Registers in cryptosystems; Proceedings 13th WSEAS International Conference on Computers, Rhodes Island, Greece, July 23-25, 2009; ISSN:1790-5109
7. M. A. Mioc, M. Stratulat - Study of Software implementation for Linear Feedback Shift Register based of 8 degree Irreducible Polynomials; Naun Journal, International Journal of Computers, volume 8, 2014; ISSN 1998-4308

## 7.4. Cercetări Viitoare

Studiind utilizarea LFSR în Câmpuri Galois GF ($2^n$) s-a demonstrat că pentru o mai bună securitate este bine să se utilizeze operații clasa modulo n cu polinoame ireductibile. Acest tip de situație se regăsește atât la codurile Reed Solomon cât și la algoritmul AES (Rijndael).
Pentru Rijndael, fiecare coeficient este un bit, fiecare element este un cuvânt de 4 octeți (32 biți).
Este posibil să se utilizeze codurile convoluționale, care sunt diferite de codurile folosind blocuri, pentru că nu au o lungime constantă.
Există tipuri speciale de modele pentru codificatoare. S-a demonstrat că codificarea convoluțională și decodarea folosind algoritmul Viterbi constituie metode puternice în corectarea erorilor.

Fiind nucleul oricărui sistem digital, registrul de deplasare LFSR (Linear Feedback Shift Register) reprezintă un subiect de bază în multe analize de cercetare.
De obicei, funcționarea unui registru de deplasare LFSR se realizează într-un câmp Galois GF($2^n$), ceea ce înseamnă că toate operațiile sunt efectuate în aritmetică modulo n , unde n reprezintă gradul polinomului ireductibil ales.

Multe studii existente actual au demonstrat că lucrând în acest mod se sporește securitatea.
Utilizarea registrelor de deplasare LFSR este bine cunoscută, de asemenea, în codurile convoluționale și după 1993 în codurile Turbo și aplicațiile lor în Detectarea și / sau Corectarea Codurilor de Eroriprecum și în Sistemul de comunicație fără fir. În ultimii ani a cescut numărul de aplicații din domeniile menționate.
O altă posibilă cercetare viitoare este comparația între AES și criptografia utilizând Elliptic Curve.
Cu ajutorul codurilor de redundanță ciclice (CRCs) pentru detectarea erorilor în domeniul sistemelor integrate se face un compromis între viteză, consumul de memorie, precum și eficiența de detectare a erorilor.
Este important să se înțeleagă opțiunile disponibile și compromisurile în găsirea unor modalități de a atinge o mai bună detectare a erorilor la un cost de calcul redus.
Există mai multe posibilități de a stimula diferite proiecte de cercetare folosind tehnologii avansate tip Field Programmable Gate Array (FPGA) , precum și tehnologii utilizând tooluri tip Electronic Design Automatic (EDA) .
Pentru stimularea unui sistem și observarea rezultatelor este posibil să se creeze test benches în VHDL (Very High Speed Integrated Circuit Hardware Description Language) sau Verilog.
Unele lucrări viitoare pot utiliza tehnici hibride care combină principalele trei deja existente:
• Enumerarea software;
• Enumerarea hardware;
• Programare Integer liniară (ILP).

CUVINTE CHEIE: Teoria codurilor, codare, testare, Built-In-Self-Test, Criptografie, Coduri detectoare și corectoare de erori