UNIVERSITATEA TEHNICĂ TIMIŞOARA

ing. Luminița VASIU

ELEMENTE DE STRUCTURĂ REDUNDANTE DESTINATE IMPLEMENTĂRII EFICIENTE A TOLERANȚEI LA DEFECTARE ÎN SISTEME DE CALCUL

TEZĂ DE DOCTORAT

Conducător ştiinţific:
Prof. dr. ing. Mircea VLĂDUȚIU

1995

CUPRINS

0. Introducere

Pentru câştigul, dar şi pentru menţinerea unui segment de piaţă, un anumit produs tehnic trebuie să posede atributul de calitate, constituind unul dintre cei mai acerbi factori de concurenţă. Calitatea este exprimată prin intermediul unei multitudini de parametri [Sch.-93], dintre care prin importanţă, se detaşează fiabilitatea reprezentând obiectivul pentru a cărui majorare sunt focalizate preocupările din lucrarea de faţă. Din vasta problematică acoperită de domeniul fiabilităţii, investigaţiile din teză sunt concentrate asupra dezideratului de tolerare a stărilor de defectare din sisteme de calcul.În fond, cercetările întreprinse se subordonează fundamentalului principiu lansat de John von Neuman încă din 1956 şi constând din sinteza unor sisteme fiabile din componente mai puţin fiabile, apelând, în acest scop, la introducerea de elemente de structură redundante. Această cale de soluţionare contravine tentativelor de sinteză clasice care, urmărind optimizarea impactului performanţă-cost, vizează eliminarea redundanţei. Prin introspecţii de profunzime, relativ la acel nucleu hardware mai vulnerabil la fenomene de defectare,în teză sunt propuse elemente de structură originale care permit, prin capacitatea de detecţie îmbunătăţită, diminuarea latenţei erorilor provocate de defectări şi, drept consecinţă, eficientizarea implementării toleranţei la defectare în sisteme de calcul. Conturate astfel, rezultatele pot fi considerate un aport la soluţionarea mereu actualei probleme a generalilor bizantini [RaKS-90,MiNi-92,KiAz-94],constituind varianta translatată la stadiul tehnologic actual a anterior amintitei problematici John von Neuman.În conformitate cu aceasta, într-o prezentare sintetică, o redută în curs de cucerire este împresurată de mai multe armate, sub comanda câte unui general.Informarea acestora asupra stadiului de desfăşurare a bătăliei este asigurată prin soli.Bazat pe tradiţia istorică-comportând multiple acte de trădare, probabil mai frecvente decât în alte părţi de pe mapamond-, armatele cuceritoare se consideră a fi bizantine. Astfel se admite că fie unul sau mai mulţi generali trădează cauza, fie că actul este atribuit unuia sau mai multor soli. Problema constă în faptul că, indiferent dacă există sau nu trădare, să existe soluţie pentru cucerirea redutei. Revenind la domeniul calculului, asociem generalilor bizantini subsisteme procesoare şi solilor subsisteme magistrale, iar obiectivelor de cucerire pe cel de furnizare a unor rezultate ca urmare a efectuării unui proces de calcul.Se conturează în acest mod un mediu de multiprocesare la care, datorită defectării, anumite subsisteme trădează şi totuşi rezultatele calculelor se impun a fi corecte.Localizând aportul tezei în contextul jalonat, noile elemente de structură redundante propuse asigură gardarea corectei funcţionări la nivelul magistralelor cu o probabilitate ridicată de detecţie a potenţialelor erori.

În cele ce urmează, organizat pe trei paragrafe, urmărind un parcurs top-down, coborâm treptele unei ierarhii pentru a atinge obiectivele tezei de doctorat.

0.1. Îmbunătăţirea dependabilităţii prin toleranţă la defectare

În urmă cu aproximativ zece ani, corespondenţa în limba engleză pentru termenul de fiabilitate era aproape unanim noţiunea de *reliability*, fapt ce poate fi certificat de o foarte lungă listă

de repere bibliografice dintre care subliniem doar lucrarea de sinteză [Prad-86].În urma activității prestate, la mijlocul deceniului nouă, de grupul de lucru IFIP 10.4 sub conducerea lui A.Avizienis şi J.C.Laprie [AvLa-86], odată cu statuarea unor dependențe clare relativ la triunghiul cădere-eroare-defect, este lansată noțiunea de dependabilitate (dependability).

Termenul *dependability* acoperă conceptele de fiabilitate (reliability), disponibilitate (availability), mentenabilitate (maintenability), securitate (security) şi testabilitate (testability), fiecare din acestea oferind indicatori numerici meniți a permite cuantificarea dependabilității unui sistem. În fond, dependabilitatea reprezintă calitatea serviciului pe care îl asigură un sistem particular.

Întrucât termenul de dependabilitate în literatura tehnică autohtonă are o arie de răspândire restrânsă, nu vom insista asupra defectului de terminologie dependabilitate-fiabilitate.

În ceea ce priveşte fiabilitatea, prin definiție, ea reprezintă capacitatea unui sistem de a exercita funcțiuni specifice într-un interval de timp prestabilit şi în condiții de exploatare determinate.În esență, fiabilitatea R(t) unui sistem reprezintă o funcție de timp, definită drept probabilitatea condiționată că sistemul va funcționa corect pe parcursul intervalului de timp $[t_0,t]$, fiind admis că la momentul t_0 sistemul executa corect funcțiile pentru care a fost creat. În vederea jalonării problematicii propuse, apelăm la conturarea arborescentă a noțiunii de fiabilitate din fig. 0.1 [Sc Go - 92].

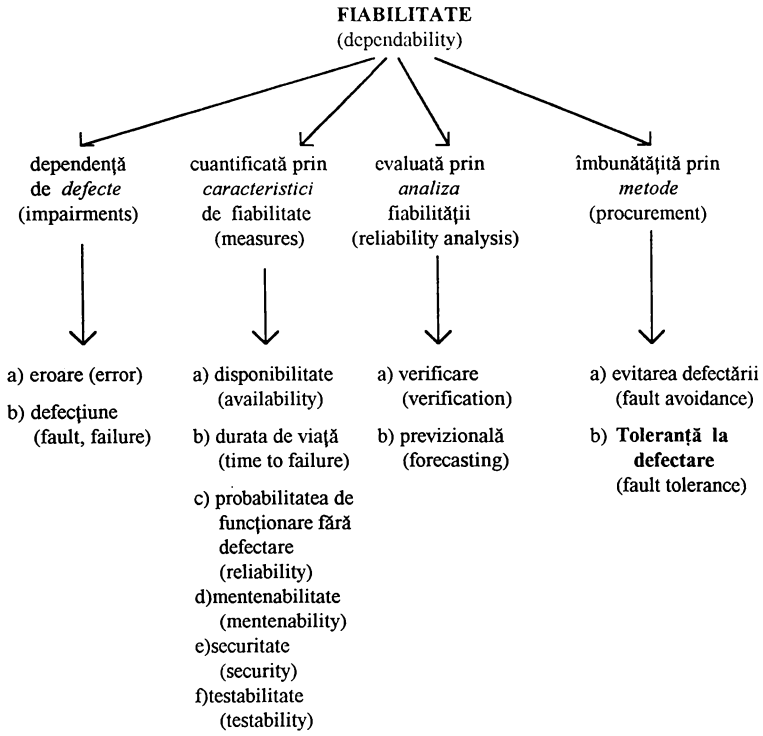


Fig.0.1

Cu toate că, prin perfecționările traversate, industria microelectronică -principala furnizoare de componente pentru sistemele de calcul - se poate afirma că se află la maturitate tehnologică deplină, evenimentul de defectare, fiind unul probabilistic,se poate manifesta determinând erori în procesul de calcul. Cum parcul de calculatoare a suferit o răspândire explozivă, pentru tot mai multe aplicații efectuarea corectă a calculelor devine de importanță crucială. Dacă cerințele de înaltă fiabilitate constituiau, până nu demult, atribute doar ale aplicațiilor militare sau aerospațiale, ele sunt astăzi solicitate de un număr în continuă creştere de aplicații comerciale. Aceste deziderate pot fi satisfăcute prin apelare la tehnici de evitare a defectării (fault avoidance) a căror obiectiv primar constă în prevenirea apariției defectelor. Acestea includ revizii ale proiectelor-prin care sunt surmontate multe din greşelile de specialitate-, aplicarea de măsuri de ecranare-prin care sistemele sunt protejate împotriva perturbațiilor externe captate, cu precădere, prin radiație-şi aplicarea de cât mai numeroase şi mai eficiente metode de control a calității-incluzând sortări de componente cu parametri mai severi decât cei revendicați de componentele ordinare, mod în care rezultă unele prevăzute cu prefixul HR(High Reliability), ofertate de către producători la prețuri corespunzător mai ridicate. La implementări bazate pe astfel de tehnici, numărul malfuncționărilor potențiale se diminuează, dar prețul de cost al sistemului poate creşte abrupt.

O soluție alternativă pentru majorarea indicatorilor de fiabilitate o reprezintă apelarea la metodele de toleranță la defectare care-spre deosebire de anterioarele tehnici a căror adaptare poate avea loc, ca grefe, pe parcursul vieții unui sistem-necesită luarea în considerare într-o fază de concepție cât mai incipientă, constituind un obiectiv primar de proiectare.

0.2 Toleranța la defectare ca obiect de proiectare

Încercând definirea toleranței la defectare *(fault tolerance)* în strictă dependență de anterior definita fiabilitate, vom înțelege capacitatea unui sistem de a executa funcțiunile specificate, adică serviciile dorite, în intervalul de timp prestabilit şi în condițiile de exploatare determinate, chiar şi în condițiile în care un număr limitat dintre subsistemele sale sunt în stare defectă. Colateral, o legătură trebuie făcută între această definiție şi fundamentala lucrare,a lui John von Neumann "Logica probabilistică şi sinteza unor automate fiabile din componente mai puțin fiabile".

Toleranța la defectare îşi găseşte aplicațiile în domenii tehnice variate, care în domeniul calculului prezintă concepte, mecanisme şi strategii specifice. Pentru conturarea noțiunii, apelăm la structura arborescentă din fig.0.2.

Bazat, pe această prezentare, într-un prim capitol vom analiza defectele urmărind relevarea modului specific de manifestare al acestora în domeniul calculului. [Nels - 90]. Vom insista asupra relației cădere-eroare-defect, reliefând aspectele particulare ale latenței defectelor, constituind intervalul de timp dintre momentul apariției condiției fizice de anomalie funcțională reprezentată de defectare şi momentul când aceasta este activată ("sensitized") afectând procesul de calcul.Caracteristicile rezultate influențează strategiile care se impun adoptate în vederea maximizării indicatorilor de fiabilitate în strânsă corelație cu degradările de performanță şi/sau creşterile de investiții care le implică.În contextul lucrării un loc aparte revine marcării defectării constituind

procesul care previne ca efectele dintr-un sistem să nu inducă erori în structura informațională a acestuia.

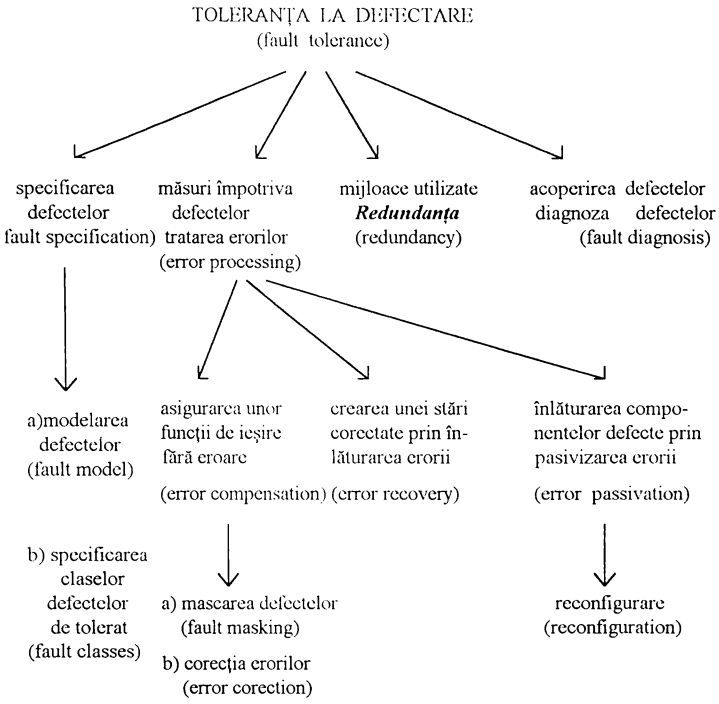


Fig.0.2

Pentru a preciza structura din fig.0.2,prin mascarea defectării se înțelege, spre exemplu, aplicarea votării după majoritate (majority voting), dar şi corectarea unei (sau mai multor) erori la operații de citire/scriere din/în memorie.Soluții alternative de toleranță la defectare sunt oferite de reconfigurare, termen care în lucrare va fi utilizat diferit față de accepțiunea sa.convențională.În conformitate cu aceasta din urmă, reconfigurarea reprezintă procesul de eliminare a unității defecte din sistem şi restaurarea stării, în această nouă condiție, a stării operaționale.Dacă se apelează la tehnica reconfigurării, proiectantul este confruntat cu următoarele procese: (1) Detecția defectului (Fault detection) reprezentând acel proces esențial de recunoaştere că a apărut o anomalie funcțională cauzată de defect; (2) Localizarea defectului (Fault location) constituind procesul de determinare a unității purtătoare a defectului; (3) Limitarea propagării efectului declanşat de defect (Fault containment) reprezentând procesul de izolare a defectului şi de prevenire a contaminării funcționării corecte prin propagarea efectului determinat de defect; (4) Restabilirea din starea de defectare constituind procesul de rămâne în stare operațională sau de a o recâştiga pe aceasta prin reconfigurare pentru un sistem în care a apărut o stare de defectare.

Pentru soluționarea complexei sarcini de proiectare a unui sistem de calcul cu toleranță la defectare de o mai mare eficiență se dovedesc metodele bazate pe introducerea redundanței, diametral opuse metodelor de sinteză convenționale care în căutarea unui optim pentru indicatorul performanță/cost al unui sistem urmăresc eliminarea redundanței.

0.3. Redundanța ca soluție pentru toleranța la defectare

Încercând definirea, în context de fiabilitate, a redundanței, vom înțelege prin aceasta, la modul general, prezența în stare funcțională pregătită a mai multor mijloace tehnice decât sunt necesare unui sistem pentru a executa funcțiunile utile necesare.În acest sens, apelăm la conturarea noțiunii de redundanță prin structura arborescentă din fig.0.3. [Sc.Go - 92]. Trebuie remarcat că față de referințele bibliografice de marcă în domeniu [John-89 sau Gork -89], în fig.0.3 apar elemente, poate nu în totalitate, noi, dar care sunt mai pregnant legate de sisteme de calcul multiprocesor,fapt pentru care vom insista asupra definirii unora dintre termeni.

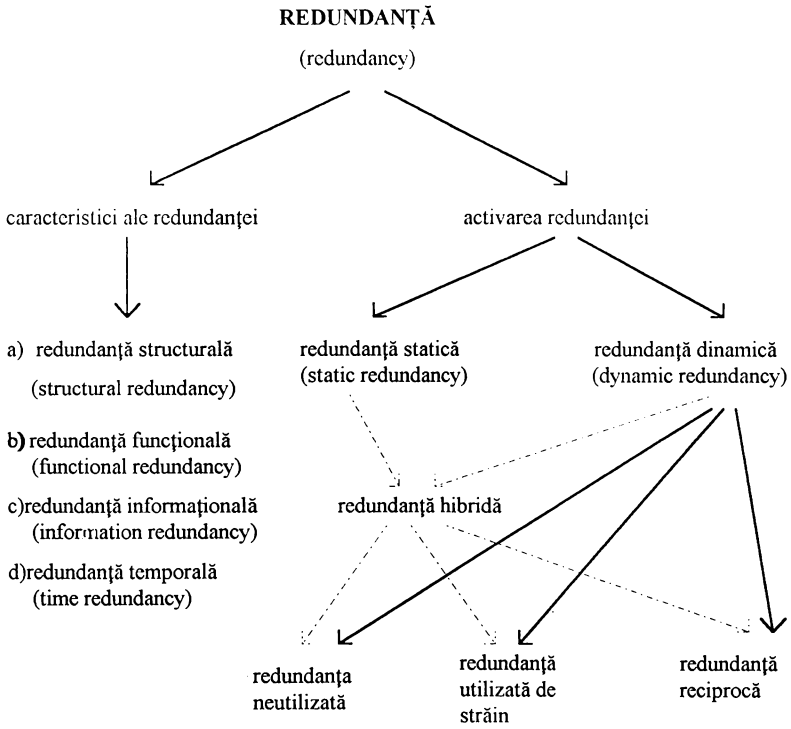


Fig.0.3

Astfel, caracteristica de *redundanță structurală* (structural redundancy) uzual legată în mod strict de componente hardware ale sistemului, este reprezentată de extensii ale sistemului cu componente (atât hardware, cât și software) suplimentare (de același tip sau de tip diferit) raportat

la cele strict necesare execuţiei funcţiunilor utile specificate. Deosebirea dintre componentele "originale" şi cele "suplimentare" nu este totdeauna posibilă, un exemplu în acest sens fiind un sistem multiprocesor TMR, cu redundanţă triplă modulară (Triple Modular Redundancy), la care nu poate fi făcută o distincţie între cele trei calculatoare. Se cuvine însă menţionat că între componentele redundante hardware, respectiv cele software, apare o deosebire esenţială în sensul că primele contribuie în mod mult mai substanţial la creşterea costului unui sistem tolerant la defectare.

Prin *redundanţă funcţională* (functional redundancy) se înţelege extensia unui sistem cu funcţii suplimentare, raportat la unele necesare funcţionării normale. Este interesant că se distinge între funcţii suplimentare în sens strict - funcţii a căror specificaţii (şi implementări) diferă faţă de cele corespunzătoare funcţiilor normale, cum ar fi, spre exemplu, comutarea pe o componentă sau un calculator de rezervă - şi funcţii suplimentare caracterizate prin diversitate (diversity), care uzitează de implementări diferite pentru îndeplinirea uneia şi aceleiaşi funcţii normale. Această din urmă categorie de funcţii, orientată, cu precădere, pentru detecţia sau tolerarea erorilor de concepţie,apare şi în [Renn - 84], dar limitată la variantele de programe elaborate pentru soluţionarea uneia şi aceleiaşi probleme, a căror rezultate se supun unei decizii bazate pe principiul majoritar. Extensia diversităţii la funcţiile suplimentare hardware cărora li se oferă un teren favorabil în condiţiile creşterii densităţii de împachetare la nivelul VLSI, constituie o idee originală.

Caracteristica de *redundanţă informaţională* (informational redundancy) este tratată în [Sc.Go.-92] absolut convenţional, prezentând clasica redundanţă codală exprimată în termeni de distanţă Hamming şi fiind bazată pe utilizarea codurilor detectoare şi corectoare de erori.

Prin *redundaţă temporală* (time redundancy), se înţelege acel interval de timp, suplimentar faţă de unul necesar funcţionării normale care este necesar la un sistem redundant funcţional pentru executarea tuturor funcţiilor. Noţiunea are un sens larg, acoperind atât suplimentul de durată necesar unei funcţii de calcul utile care prevede repetarea unor operaţii - cum ar fi resetarea citirii de la o unitate periferică a unui bloc de informaţie găsit cu eroare CRC, cât şi diferenţa dintre durata de execuţie a funcţiei de calcul normale, utile şi una necesară execuţiei variantei celei mai defavorabile prin prisma timpului revendicat. De asemenea, redundanţa de timp se referă şi la intervalul necesar aşa numitelor *instanţe de toleranţă la defectare* (Fehlertoleranz - Instanzen), constituind acele componente (hardware sau software) suplimentare care execută algoritmi de diagnosticare şi tratare a erorilor. Se impune remarcat faptul că multe lucrări de specialitate consultate nu pun în relief, în mod explicit, caracteristica temporală a redundanţei, chiar dacă o înţeleg.

În ceea ce priveşte divizarea metodelor de activare în unele dinamice, şi altele statice, precum şi în cele hibride, este una convenţională, majoritar acceptată [Gork-89,John-89,SeGo-92].

0.4. Structura tezei de dizertaţie

Parcurgând în sens top-down ierarhia constituită de treptele dependabilitate (fiabilitate)-toleranţă la defectare-redundanţă, aşa cum este sugerată în acest capitol introductiv, deschidem perspectiva preocupărilor din prezenta teză de dizertaţie aparţinând acelui subdomeniu al fiabilităţii constituit de implementarea toleranţei la defectare în sisteme de calcul.Ajungând prin traseul descris

la miezul constituit de redundanță, anticipăm elementul de esență al lucrării reprezentat de structurile ASIM, descrise în capitolul 3, unități hardware redundante menite a asigura, prin capacitate de detecție sporită, o soluționare eficientă pentru cruciala sarcină de detecție a defectelor caracteristică oricărui sistem de calcul tolerant.

Primele două capitole pregătesc atingerea acestor structuri redundante țintă motivând, prin ample incursiuni de literatură, aplicarea lor, iar ultimele trei capitole prezintă versiuni de utilizare precum și implementarea lor practică într-un sistem dual sincron cu toleranță la defectare.

Astfel, problematica primului capitol constă în analiza modului de defectare specifice sistemelor de calcul și modelarea acestora.În urma unor clarificări de terminologie legate de triunghiul cădere-eroare-defect, împănate cu exemple originale, este disecată critica problemă a latenței defectării și, imediat conex cu aceasta, a latenței detecției. Motivația investigațiilor legate de cele două latențe o constituie importantul parametru de proiectare pe care îl reprezintă acesta pentru viitorul sistem tolerant dual sincron, obiectiv esențial descris în capitolul 6. Această parte a lucrării mai conține modelarea defectelor, un accent deosebit fiind pus pe cele de scurtcircuit și pe cele de blocare la întrerupere, acestea din urmă specifice actual foarte răspânditei tehnologii CMOS. În contextul defectelor de scurtcircuit, teza conține o originală analiză corespunzătoare tehnologiei TTL, dar care poate fi extrapolată și la alte tehnologii.Acest prim capitol se încheie prin examinarea delicatelor defecte temporare, precum și a modelelor acestora, efectuată în baza consultării unui număr mare de repere bibliografice.

Justificând probabilitatea de apariție mai mare a defectelor singulare în raport cu cele de multiplicitate superioară, și alegând, pe baza argumentației rezultate în urma analizei din capitolul întâi, în al doilea, se selectează acele metode de implementare a toleranței la defecte menite a pregăti rezolvările tehnice care vor fi adoptate în capitolele următoare. Astfel, bazat pe surse bibliografice de marcă este compilat un paragraf cu măsuri de izolare-decuplare a defectelor în care autoarea își aduce unele contribuții la clasificarea metodelor recomandate.Dintre multiplele metode hardware de implementare a toleranței la defecte sunt sumar prezentate cele menite a contura viitoarea soluție constând din supravegherea funcțională reciprocă prin dublare statică, respectiv prin triplare statică. Paragraful final, al treilea, al acestui capitol se referă la strategiile conceptuale pentru elaborarea produselor program de autotestare.În această parte, autoarea efectuează o originală selecție a structurii sistemelor de calcul prin prisma elaborării programelor de autotestare în baza conceptelor de activare modulară, respectiv de verificare a instrucțiilor. Concluzia care se desprinde, în ambele situații, este aceea că, indiferent de scopul urmărit-detecție sau diagnoză-,împănarea cu frecvente instrucții de comparare determină o capacitate de trecere redusă a programelor de autotestare,ori, cu precădere, în sisteme tolerante dual sincrone, importanța minimizării acestui parametru este crucială atât la stabilirea calculatorului defect, respectiv a celui cu funcționare normală, cât și la diagnosticarea modulului purtător al defectului din cadrul calculatorului la care a fost sesizată anomalia funcțională. Aceasta constituie argumentația naturală a unor structuri redundante care să permită reducerea capacității de trecere a programelor de autotestare, structuri elaborate în proximul capitol.

Prin urmare, în capitolul trei sunt analizate metodele de comprimare paralelă şi structurile BILBO rezultând în baza unor preocupări îndelungate ale autoarei în acest domeniu, noi structuri de comprimare denumite abreviat ASIM (Analizor de Semnături cu Intrări Multiple). Căutând o justificare pentru conexiunile de pe lanțul de reacție al structurii BILBO în regim de analizor de semnături și în regim de generator de secvențe pseudoaleatoare,autoarea găseşte că, prin modificarea legăturilor, noua structură ASIM prezintă o capacitate de detecție superioară structurilor BILBO cunoscute din literatură.În acest context,autoarea prezintă critica sintezelor BILBO, în lucrare fiind inserate un număr mare de repere bibliografice care cuprind sinteze inferioare noilor structuri ASIM.În plus, lucrarea conține două structuri ASIM echivalente prin prisma capacității de detecție a potențialelor defecte cu cea corespondentă structurii BILBO din literatură, dar superioară prin prisma performanței exprimată prin valoarea frecvenței impulsurilor de tact destinate procesului de comprimare. Cele trei structuri menționate sunt fundamentate din punct de vedere formal prin 3 teoreme și 3 leme originale. Materializând preocupări ale autoarei în domeniul codurilor detectoare și corectoare de erori, celor trei structuri ASIM li se asociază, în mod original, matrici de control care permit generarea asistată de calculator a semnăturilor etalon necesare în procesul de testare. În acest sens, autoarea propune un număr de 6 proceduri originale de soluționare a problemei amintite prin intermediul calculatorului.

Capitolul patru urmăreşte fructificarea structurilor propuse anterior pentru facilitarea testării corespunzătoare la trei obiective din structura sistemelor de calcul. Astfel,în primul paragraf, sunt analizate metodele de proiectare pro-testabilitate bazate pe tehnici de scanare în vederea stabilirii unei strategii optime care să permită evaluarea răspunsurilor prin noile structuri ASIM. În acest context, autoarea justifică o combinație originală a metodei LSSD cu structurile ASIM, pe care o propune ca alternativă la standardul IEEE1149.1 în scopul eficientizării testabilității prin probabilitate inferioară-față de soluția convențională-de recunoaştere ca funcțional corecte a unei unități testate defecte. Al doilea paragraf are ca obiectiv schemele PLA în vederea îmbunătățirii testării acestora prin adusuri built-in a căror element de esență, în ceea ce priveşte evaluarea răspunsurilor, să fie reprezentat de aceleași structuri ASIM.Stimularea schemelor PLA se realizează prin vectori binari elaborați pe baza cunoscutului concept de testare independentă de funcție.Prin analize comparative, justificate formal printr-o teoremă originală și adaptarea a 8 leme, se fundamentează intercalarea structurilor ASIM în scheme PLA, cu posibilități de extindere.În fine,cel de-al treilea paragraf încearcă intercalarea structurilor ASIM în scheme UAL (Unități Aritmetice şi Logice) pentru eficientizarea autocontrolului acestora.În acest context, în urma analizelor corespunzătoare detecției erorilor la adunarea binară prin coduri sumă de control și prin coduri combinate, autoarea ajunge să propună un nou cod sumă de control bazat pe generarea semnăturii prin mecanisme specifice structurilor ASIM.Codul propus este superior celui descris în [RaFu-89], prin prisma capacității de detecție a potențialelor erori.În continuare, autoarea analizează detecția erorilor și corecția erorii singulare prin coduri bazate pe paritate la operațiile de adunare/scădere binară și înmulțire binară, finalizând acest capitol cu propuneri de aplicare a structurilor ASIM în circuite integrate dedicate.

Spre deosebire de fructificarea prin propuneri a aplicării structurilor ASIM din capitolul patru, în al cincelea se prezintă o implementare practică a acestora în forma unui modul hardware, component al unei biblioteci destinate unei magistrale standard, în particular MULTIPROM.Modulul este caracterizat prin utilizarea originală a conceptului de reconfigurare, aplicat la sinteza structurii prin utilizarea mai multor polinoame generatoare, în scopul maximizării, pe această nouă cale, a capacității de detecție.

Capitolul şase extinde realizările prin furnizarea documentației de execuție pentru un sistem dual sincron cu toleranță la defectare destinat proceselor industriale, elaborat de un colectiv de cercetare a cărei activitate a fost coordonată de către autoare.Cumulând investigații de natură teoretică întreprinse în anterioarele capitole, structura sistemului include ca element de esență modulul hardware bazat pe structuri ASIM. În mod aparte, sistemul cuprinde un nucleu hardware de fiabilitate sporită care conține un modul generator de tact şi unul comparator, ambele cu toleranță la defectare, precum şi un modul memorie comună cu cod corector al erorii singulare şi detector alnumărului maxim posibil de defecte duble. Configurarea de ansamblu al sistemului dual precum şi documentația de detaliu a principalelor realizări este ataşată în finalul acestui capitol.

Teza de dizertație se încheie cu un capitol de concluzii care sintetizează contribuțiile autoarei la modul general,cele de detaliu fiind anexate, sub forma unui paragraf distinct, la fiecare din cele şase capitole de esență ale lucrării.

Autoarea ține să mulțumească, în primul rând, conducătorului ştiințific prof.dr.ing.Mircea Vlăduțiu pentru îndrumarea competentă şi de înaltă ținută ştiințifică acordată atât în calitate de profesor coordonator al tezei, cât şi în calitate de şef al filialei din Timişoara a Institutului de Cercetare Ştiințifică şi Inginerie Tehnologică pentru Automatizări unde autoarea şi-a realizat majoritatea aplicațiilor practice.

De asemenea, autoarea mulțumeşte tuturor profesorilor şi conducătorilor instituțiilor unde a activat, pentru contribuțiile acestora la formarea sa în calitate de cercetător ştiințific şi de cadru didactic, precum şi colegilor care au încurajat-o pentru finalizarea tezei de dizertație.

1 Analiza modurilor de defectare specifice sistemelor de calcul şi modelarea acestora

1.1. Căderi şi defecte în sistemele de calcul

Pentru eliminarea ambiguităţilor de exprimare, plecând de la faptul că, strict etimologic, delimitarea dintre termenii în limba engleză "failure" şi "fault" nu este, se pare, aceeaşi cu cea dintre "cădere" şi "defect", apelăm la următoarele definiţii, fără însă a avea pretenţia epuizării acestei probleme care în literatura apărută în limba română este destul de confuză.

Astfel, se înţelege, în general, prin *cădere (defecţiune)*, corespondentă a englezescului "failure", pierderea totală sau parţială, de către o componentă electronică a capacităţii sale de funcţionare. Urmărind un câştig în claritate, menţionăm că atunci când ne referim la componentă electronică, fără a face alte precizări, subânţelegem o capsulă de circuit integrat aparţinând nivelului de integrare pe scară foarte largă (VLSI), fapt justificat de dominaţia numerică a acestora în sistemele de calcul actuale [Haye-88].

Pentru a elucida suplimentar noţiunea de defecţiuni, apelăm la dependenţa de timp a unei importante caracteristici de fiabilitate, intensitatea de defectare λ. Această dependenţă cunoscută sub denumirile de "curbă în şea" sau " curbă cadă de baie ", permite punerea în relief a celor trei perioade ilustrate în fig.1.1., specifice duratei de viaţă a unei componentei electronice.

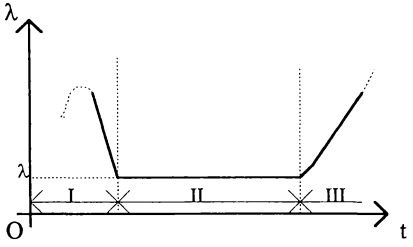


Fig.1.1

Prima perioadă (I), denumită de rodaj, corespunde etapei de "copilărie" a componentei în care pericolele de apariţie a evenimentului de cădere sunt multiple având la bază variate cauze acoperite prin noţiunea generică de imaturitate tehnologică. Acestea se reduc numeric prin perfecţionarea procedeelor de fabricaţie, fapt relevat de panta cu alură, în general, abrupt căzătoare din fig.1.1, ajungând la o stabilizare, caracterizată prin valoarea aproximativ constantă a lui λ care corespunde perioadei a doua (II), denumită de exploatare normală. Căderile din această etapă se disting prin modificarea bruscă, catastrofică, a valorilor parametrilor cu consecinţa pierderii totale a capacităţii de funcţionare a componentei, un exemplu constituindu-l blocarea ieşirii unui circuit integrat la una din valorile logice indiferent de vectorul binar aplicat în calitate de stimul la intrările sale. Cauzele care provoacă aceste căderi sunt multiple, ţinând seama inclusiv de fazele de geneză ale

componentei, fapt pentru care studiul lor este întreprins prin aparatul oferit de teoria probabilităţilor şi statistica matematică. Urmărind alura curbei înspre partea finală a etapei de exploatare normală, se sesizează, în general, o uşoară creştere care devine din ce în ce mai pronunţată, dar cu o pantă domol totuşi crescătoare în perioada a treia (III), denumită de îmbătrânire. În această etapă, căderile sunt cauzate, în esenţă, de deplasări previzibile, în general lente, ale valorilor parametrilor, datorate fenomenului de îmbătrânire fizică, determinând pierderea, mai întâi, parţială şi, în final totală a capacităţii de funcţionare a componentei. Aceste căderări afectează preponderent parametrii statici şi dinamici ai componentei, cum ar fi valoarea curentului I_{CC} absorbit de la sursa de alimentare sau valoarea timpului t_{PD} de propagare corespunzător unui circuit integrat, determinând migrarea valorică a acestora până la părăsirea câmpurilor tolerate de o funcţionare fiabilă. Numărul, în general mare al parametrilor, precum şi cel al factorilor de mediu (în special temperatura, dar şi umiditatea, presiunea atmosferică, şocuri, vibraţii, radiaţii, ş.a.) fac ca această categorie de căderi să fie studiate prin legi mai degrabă probabilistice decât deterministe, oferite de acea parte a teoriei fiabilităţii, cunoscută sub denumirea de fiabilitate parametrică.

Anterioara prezentare ne dă prilejul unei conturări mai clare a noţiunii de cădere, noi legând-o doar de perioada exploatării normale. Motivăm această restricţie, în primul rând, prin faptul că, datorită lărgirii continue a spectrului familiilor logice digitale, la sistemele de calcul intervine, în general, îmbătrânirea morală a acestora, şi deci pierderea lor în interes, mai înainte ca să apară îmbătrânirea fizică a componentelor electronice utilizate la sinteza lor tehnologică. Pe de altă parte, considerăm că un fabricant de calculatoare, dacă nu posedă propria linie tehnologică de fabricaţie a componentelor, supune acestea din urmă unei faze de testare prealabilă împachetării pe plachete cu cablaj imprimat, prin aceasta fiind reduse la minim căderile specifice perioadei de rodaj. Restrângând aceste consideraţii la perioada de exploatare normală , trebuie să mai adăugăm că, în aceasta, căderile parametrice, chiar dacă nu excluse, sunt rare, influenţa factorilor aleatori ai mediului fiind, în general, acoperită prin algoritmii de evaluare a fiabilităţii în raport cu căderile totale ale componentelor.

Referindu-ne, în al doilea rând, la noţiunea de defect, pe care o asociem englezescului "fault", aceasta reprezintă, printr-o definiţie majoritară, o condiţie fizică de anomalie funcţională provocată de variate cauze, pe care,fără pretenţia de a le acoperi în totalitate,le supunem următoarelor comentarii :

a) *Erori de concepţie - proiectare*, care în conformitate cu [SeGo-92] sunt subdivizate în erori de specificare (care apar, în mod uzual, pe traseul client - utilizator al sistemului de calcul - fabricant şi care sunt combătute, cu precădere în ultima perioadă, prin apariţia de standarde sponzorizate de IEEE, un exemplu edificator constituindu-l limbajul de descriere hardware VHDL - standard IEEE 1076), de implementare şi de documentare. Această clasă de erori, au drept consecinţă defecte de concepţie, proiectare, constituind după [Renn - 84] călcâiul lui Achile al toleranţei la defectare şi care, conform aceleiaşi surse bibliografice se împart în defecte ale procesului de fabricaţie la nivel semiconductor (semiconductor processing faults) -cum ar fi, spre exemplu, contaminarea la nivelul substratului, şi care necesită în vederea inspectării standuri de verificare, inclusiv de îmbătrânire accelerată, definite în conformitate cu standarde, uzual, militare şi traduse în

milioane de dolari -, în defecte ale proiectului logic (logic design faults) - considerate cele mai subtile, ele constând din stări de control care apar rar şi care impun măsuri protective speciale, - şi în defecte software (software faults) - considerate ca cele mai numeroase, acoperind inclusiv date de intrare neanticipate şi relaţii temporale (timing relationships) care apar rar.

Concluzionând, conform cu [Nels-90], se poate afirma că defectele determinate de erori concepţie-proiectare sunt dificil de modelat şi, în consecinţă, de combătut datorită apariţiilor şi a efectelor greu predictibile.

b) *Imperfecţiuni ale tehnologiei echipamentelor de calcul.* Cuprindem sub denumirea generică de tehnologie atât în faza de sinteză zehnologică cât şi procesul de fabricaţie al împachetării, asamblării echipamentelor de calcul.

Referindu-ne, mai întâi, la sinteza tehnologică, defecte pot fi provocate de insuficienţa măsurilor care se iau împotriva interferenţelor electromagnetice generatoare de perturbaţii interne sistemului logic.Acestea din urmă,-constituind semnale, în general, neregulate de tensiune şi curent, care nu prezintă informaţie utilă şi care la anumite valori ale amplitudinii şi duratei pot provoca funcţionarea instabilă a schemelor-au o multitudine de cauze, în mod uzual, specifice familiilor logice (spre exemplu, fenomenul de latch-up la familia CMOS), iar recomandările pentru diminuarea lor sunt deseori contradictorii (spre exemplu, un montaj mai relaxat favorabil reducerii diafoniei conduce la creşterea în lungime a traseelor imprimate favorizând apariţia reflexiilor). Problematica defectelor cauzate de perturbaţii interne sistemului logic (uzual, divizate în perturbaţii pe barele de alimentare, datorate diafoniei şi datorate reflexiilor) se acutizează o dată cu apariţia familiilor bazate pe tehnologie hibridă, cum ar fi, spre exemplu, SN 74ABT(Advanced BiCMOS Technology, implicând o structură logică realizată în tehnologie bipolară şi CMOS). Găsirea soluţiilor tehnice de compromis acoperitoare a unui număr cât mai mare de defecte aparţinând acestei categorii, cu consecinţă favorabile legate de creşterea acurateţei proiectului la nivel tehnologic implică investiţii însemnate în echipamente de testare, bazate inclusiv pe investigare optică, a căror cost se traduce corespunzător în preţul de calcul.

În al doilea rând, referindu-ne la procesul tehnologic de asamblare al calculatorului, se impun noi precizări legate de numărul de împachetare. Astfel, de la microcalculatoare înspre sistemele mari (mainframes) sunt implicate, în general, nivelurile de plachetă-sertar-dulap, dar tot mai consistentul segment de piaţă al echipamentelor terminale (multi-user servers, engineering workstations, high-end desktop PCs, laptop computers) apelează uzual, doar la nivelurile plachetă - cutie (cabinet) [TILV-93].

Chiar dacă excludem din consideraţii, pe baza celor mai sus expuse, defectele cauzate de imperfecţiuni tehnologice la nivelul componentelor, cum ar fi transpunerea eronată a măştilor, catalogându-le drept căderi, operaţiile implicate de împachetare la nivelul plachetei generează noi căderi, respectiv defecte, delimitarea dintre cei doi termeni fiind germenele pentru alte noi ambiguităţi. În încercarea de elucidare a problemei apelăm la noţiunea de unitate tehnologică replasabilă asociată rezoluţiei de diagnosticare din domeniul testării logice, constituind unitatea tehnologică (capsulă de circuit integrat, plachetă sau în ultimă instanţă, calculator) până la care este

condusă activitatea de investigare în vederea localizării malfuncționării, continuarea acesteia în cadrul respectivei unități pierzându-și sensul, aceasta urmând a fi oricum înlocuită în cadrul operației de reparare. Astfel, căderile provocate în procesul productiv al plachetei, problema punându-se în mod similar și în raport cu nivelurile de împachetare superioare, sunt receptate ca atare când unitatea replasabilă este constituită de plachetă, dar ele devin defecte atunci când urmează a fi înlocuită unitatea tehnologică reprezentată de chip. Convenția propusă necesită argumentări suplimentare, ceea ce ne permite să adăugăm noi comentarii asupra malfuncționărilor cauzate de imperfecțiuni ale procesului tehnologic.

Punând sub lupă căderile care se manifestă la nivelul plachetei echipate și excluzând cele corespunzătoare componentelor, a căror apariție nu trebuie neglijată întrucât solicitări termice anormale nu pot fi integral evitate chiar dacă asamblarea este realizată prin mașini productive robotizate pentru tehnologia de montare la suprafață (SMT - Surface Mounting Technology), malfuncționările pot fi cauzate de lipituri reci, contacte imperfecte (spre exemplu, în cuplele extensiilor de pe o plachetă mamă), întreruperi ale traseelor imprimate, scurtcircuite între acestea, dar și de creșterea valorii impedanței unora dintre liniile de cablaj peste cea tolerată sau coborârea valorii impedanței de izolație dintre unele traiecte conductoare sub cea care asigură funcționarea fiabilă ș.a. Ultimele două tipuri vizează perioada de îmbătrânire și, motivat pe cele anterior relevate, nu le luăm în considerare (de altfel, conform cu specificațiile majorității echipamentelor de testare a plachetelor neechipate [Fuji-90]) și, de asemenea, excludem din considerentele următoare lipiturile reci și contactele imperfecte cauzate de imaturități de natură tehnologică. Rezumându-ne la întreruperi și scurtcircuite, trebuie remarcat că sub aspect comportamental, unele dintre aceste căderi de plachetă se suprapun peste căderi de componente. Mai trebuie subliniat că aceste cauze de manifestare apar frecvent la testarea plachetelor din cadrul fluxului de fabricație dar probabilitatea lor de apariție în exploatarea calculatoarelor este, în mod uzual, redusă.

Cu toate că referirile noastre privitoare la tehnologia calculatoarelor au fost exclusiv orientate spre malfuncționări vizând hardware-ul, erori în elaborarea programelor aparținând sistemelor de operare - cum ar fi, spre exemplu, erori la scrierea compilatoarelor sau copierea eronată de programe trebuie , de asemenea, luate în considerare [ScGo-92].

c) *Deficiențe în exploatarea calculatoarelor.* Gama cauzelor de defectare privind exploatarea sistemelor de calcul este diversă și se impune restricționarea ei la evenimente mai probabile. Vom exclude, pentru început, erorile provocate intenționat (acte de sabotaj sau atentate la protecția datelor, de importanță majoră în unele aplicații, cum ar fi cele bancare), precum și acele concentrări întâmplătoare de solicitări de natură fizică care se traduc prin defecte ale componentelor, evenimente guvernate în această fază, așa cum s-a prezentat, de legi probabilistice. Rămân însă, perturbațiile captate din mediul de exploatare constând din câmpuri electrice, magnetice, electromagnetice, dar și din influențe mecanice, termice sau de altă natură. Proveniența lor, preponderent prin conducție și radiație, este, în general, combătută prin măsuri protective adaptate prioritar în faza de sinteză tehnologică (optocuploare, filtre, diferite tipuri de ecranări), dar, plecând

de la paleta largă a aplicațiilor, ele nu pot fi ignorate cu atât mai mult cu cât modelarea și predicția lor este dificilă.

Desigur, un număr apreciabil de defecte pot fi provocate prin erori de operare, precum și prin măsuri inadecvate de întreținerea echipamentelor de calcul.

Încercând unele concluzii la investigarea de terminologie, din incursiunea întreprinsă, care ne-a permis și o anume jalonare a problematicii care urmează a fi supusă dezbaterii, rezultă percepția noțiunii de defect ca o cauză ipotetică sau identificată a unei căderi ca și a unei erori, fapt ce justifică, în intenția unei cât mai avansate acoperiri a evenimentelor de malfuncționare, conectarea noțiunii de toleranță la cea de defect mai degrabă decât la cea de cădere, conducând la sintagma a cărei corespondent în limba engleză este "fault tolerance". Amintim că investigațiile de terminologie sunt, în general, concordante cu sinteza cuprinsă în [AvKe.-84] și cu concluziile grupului de lucrului de lucru IFIP Working Group 10.4 Reliable Computing and Fault Tolerance.Rezumându-le pe acestea din urmă,problema este descrisă în contextul în care un sistem numit resursă (r) trebuie să furnizeze un serviciu (s) pentru un alt sistem sau persoană numite utilizator (u). Se consideră că apare o cădere (failure) atunci când utilizatorul percepe că resursa îl frustează de serviciul dorit. Pe un parcurs "top down" dăm următoarele exemple:

(1) Un agent de rezervare (u) percepe că sistemul de calcul (r) recunoaşte cererea sa, dar nu îi furnizează un răspuns (s).

(2) Ceasul de gardă (watchdog timer) (u) percepe că programul curent (r) nu l-a inițializat (s) înaintea începerii execuției.

(3) Unitatea centrală (CPU) (u) percepe că memoria (r) i-a furnizat un cuvânt (s) cu paritate eronată.

(4) Un tranzistor B (u) percepe că ieşirea tranzistorului A (r) nu se modifică după ce B a aplicat un stimul lui A.

Prin urmare, indiferent de definiția resursei-care poate fi un calculator, un program, o memorie (privită ca subsistem) sau chiar un singur tranzistor -, o cădere constituie pierderea serviciului percepută de către utilizator la interfață cu resursa.

Înainte de a ajunge la noțiunea de defect (fault), este definită cea de eroare (error), care apare atunci când o anumită parte a resursei ajunge într-o stare nedorită,fiind opusă specificațiilor resursei sau necesităţilor utilizatorului.Dăm următoarele exemple de erori:

(1) *Eroarea de paritate*. Toate cuvintele sunt stocate într-o unitate de memorie cu paritate impară, iar la o operație de citire este furnizat un cuvânt cu paritate pară.

(2) *Eroare de comparare.* Două sumatoare identice recepționează aceiași operanzi și furnizează, în mod simultan, suma la un comparator care sesizează că acestea nu coincid în toate pozițiile binare.

(3) *Eroare de timp.* Anterior menționatele sumatoare recepționează simultan aceiași operanzi, dar numai unul furnizează suma la comparator în intervalul de timp prestabilit.

(4) *Eroarea de program.* Ieşirile identice ale ambelor sumatoare indică prezenţa depăşirii care ar fi trebuit prevenită într-o etapă anterioară printr-o corespunzătoare testare a domeniului valoric al operanzilor.

Prin urmare, o eroare reprezintă o stare nedorită a resursei ,existând într-un punct intern al acesteia- datorită unei căderi şi care poate fi latentă până când starea activează un serviciu nedorit la interfaţa cu utilizatorul.

În fine, un defect (fault) este detectat fie când apare o cădere a resursei, fie când este observată o eroare în cadrul acesteia. Diferenţele dintre cădere- eroare şi defect este determinată de modul în care este definită limita de deservire a resursei, fiind posibilă, dependent de aceasta, o suprapunere a noţiunilor. Se poate considera că, o resursă cu toleranţă la defectare detectează defecte prin intermediul unor algoritmi care recunosc erori. Algoritmii de restabilire (recovery algorithms) ai unei resurse corectează erorile şi elimină defectele. Prezenţa defectelor nu trebuie să ajungă să fie percepută drept cădere la interfaţa resursei cu utilizatorul [John-89].

1.1.1. Aspecte specifice domeniului calculului ale evenimentelor de defectare

În acest subparagraf ne propunem aprofundarea modului de manifestare a evenimentului de defectare, introducând noi restricţii care să justifice conceptele de toleranţă la care ne vom referi.

Pentru fixarea ideilor vom admite că unitatea tehnologică replasabilă este reprezentată de capsula VLSI şi că posibilitatea să se defecteze constă din blocarea la o valoare logică, manifestată pe un pin de intrare sau ieşire. În condiţiile acceptării echiprobabilităţii de defectare a tuturor componentelor şi a echiprobabilităţii de blocare la cele două valori logice (0, respectiv 1), ipoteze comentabile din punctul de vedere a unor autori [Wojt-88, Fuji-90], se poate justifica probabilitatea mai mare de apariţie a unei căderi singulare în raport cu cea corespunzătoare apariţiei unor căderi cu grad de multiplicitate superior. Astfel, în baza teoremei lui Bernoulli din teoria probabilităţilor, pentru probabilitatea defectării singulare Q_1 a unui sistem de calcul fără redundanţă (conectat serie din punct de vedere al fiabilităţii), avem:

$$Q_1 = C_n^1 \, p^{n-1} (1 - p) \qquad (1.1)$$

în care p este probabilitatea de funcţionare fără defectare a oricărei din cele n componente- toate admise echifiabile- ale sistemului de calcul, iar C_n^1 reprezintă numărul combinărilor de n elemente luate câte unul, aşa numitul coeficient binomial. În mod similar, pentru probabilitatea Q_2 a defectării de tip blocare (stuck-at) a două elemente, avem :

$$Q_2 = C_n^2 \, p^{n-2} (1 - p)^2 \qquad (1.2)$$

Raportând expresiile celor două probabilităţi, şi anume Q_2 la Q_1 ,şi luând în considerare că p pentru componente electronice tinde către unitate, rezultă în mod evident:

$$R_{21} = \frac{(n-1)(1-p)}{2p} \qquad (1.3)$$

Cum valorile pentru n sunt de ordinul sutelor şi miilor (uneori, la mainframe-uri, chiar de zeci de mii), iar valorile pentru p tind spre 1, se poate uşor remarca, că R_{21} tinde către 1/2, deci, sub aspectul analizat, ipoteza apariţiei mai probabilă,la un moment a unui singur eveniment de defectare poate fi considerată drept acceptabilă.Această afirmaţie este cu atât mai adevărată, se poate dovedi prin raţionamente similare, că ordinul de multiciplitate creşte (defecte triple, cuadruple,etc.).

Spre deosebire de majoritatea căderilor din electronica analogică, care au drept efect pierderea cvasiinstantanee cu momentul defectării unei componente a capacităţii funcţionale de către întreg sistemul electronic (cum ar fi, de exemplu, străpungerea unui condensator având ca efect dereglarea imaginii unui televizor), în electronica numerică majoritatea căderilor implică, în mod uzual, în intervale de timp variabile, dar care pot atinge uneori valori mari (uneori chiar ani, cum ar fi, spre exemplu, căderea unei celule de memorie activată doar în condiţiile unei anumite combinaţii binare) până când ajung să afecteze prin eroziune procesul de calcul. O altă caracteristică distinctivă pentru o mare parte dintre defecţiunile din domeniul calculului raportat la cele din electronica convenţională constă în faptul că este mai rară provocarea de malfuncţionări secundare, în lanţ, a mai multor componente.

Să insistăm, prin urmare, asupra căderilor din domeniul calculului, analizând modul lor de manifestare, de importanţă decisivă în sinteza configuraţiilor tolerante la defectări. Focalizând analiza la nivelul de intimitate constituit de poarta logică (gate level), căderile pot fi divizate dependent de efectul provocat în unele aşa numit logice, care determină o funcţionare eronată din punct de vedere logic, şi în altele aşa numit parametrice, care determină ca anumiţi parametri de circuit să prezinte valori pe care nu s-a contat la proiectare. În mod uzual, căderile parametrice se referă la porţi "leneşe" a căror parametri dinamici (timpi de propagare sau de tranziţie) afectează operarea în timp (timing operation) determinând funcţionarea instabilă a schemelor prin hazarduri. Acestea constau din impulsuri parazite care au ca efect erori tranzitorii (transient errors), dar când sunt aplicate la bistabile şi au energia necesară se pot transforma în stări incorecte cu caracter permanent de manifestare. În ceea ce priveşte căderile logice, acestea constau în abateri vizând funcţionarea logică aşteptată şi ele se caracterizează prin propagarea efectului lor prin schemă.În acest mod, eroarea poate să ajungă să fie detectată după mai multe niveluri logice în raport cu locul acesteia de manifestare primară, implicând ca succesiv detecţiei să fie apelat un mecanism, de cele mai multe ori, complicat pentru diagnosticarea erorii. Este momentul să intercalăm o importantă remarcă pentru consideraţiile viitoare, legată de analiza şi sinteza sistemelor tolerante la defectări, şi anume că majoritatea investigaţiilor surselor de literatură evită problema propagării erorii admiţând, în esenţă, o acoperire perfectă a erorilor prin mecanismele de detecţie. Conform şi cu [ShLi - 88], o astfel de abordare este nerealistă şi, deseori, inacceptabilă pentru sisteme reale, deoarece chiar şi mecanisme de detecţie aproape perfecte (near - perfect detection mechanism) sunt foarte dificil de obţinut fără o creştere excesivă a resurselor şi/sau o degradare avansată a performanţelor. Modelele

de propagare a erorilor sunt admise, în marea lor parte, deterministe în natură, deoarece sunt bazate pe modele defect / eroare restrictive sau presupun că sistemul are o comportare predictibilă. Totuşi, aşa cum se susţine şi în [ShLi - 88], în practică există o informaţie prealabilă, în mod uzual, redusă despre comportarea defectelor şi erorilor, astfel că pot avea loc comunicaţii intercomponente în moduri foarte diferite. Astfel, propagarea erorilor nu poate fi modelată, în general, în mod deterministic, fiind necesare modele stohastice.

Abordate prin prisma duratei la manifestare, este uzuală defalcarea evenimentelor de defectare în tranzitorii, intermitente şi permanente [Nels-90,Gork-91]. În mod convenţional, sunt acceptate drept defecte tranzitorii, cu precădere, unele provocate de perturbaţii externe, care există pe o durată finită şi sunt nerecursive. Atunci când sistemul oscilează între o stare de operare defectă şi una normală, aceasta este în general atribuită unui defect intermitent,adică un defect temporar, recurent care reapare în mod regulat. În situaţii când sistemele au prevăzute intervale de profilaxie a defectelor sau în situaţii de testare (atât on-line, cât şi off-line), strategiile de evidenţiere a defectelor prevăd, în general, supunerea unităţilor suspecte la condiţii de solicitare extremă - atât electrică, cât şi din punct de vedere al parametrilor aleatori ai mediului de exploatare - aşa numite de îmbătrânire accelerată, în care mare parte din defecte devin permanente. Această ultimă categorie, denumită uneori şi de defecte "hard", pentru detecţia cărora există variate mecanisme, sunt reprezentate în mod uzual de condiţii de funcţionare incorectă independente de timp. Se impune însă remarcat că există şi unele defecte permanente a căror mod de manifestare este intermitent, sens în care dăm exemplul unui numărător a cărui linie de iniţializare este întreruptă (defect permanent) şi care la conectare poate ajunge într-o stare oarecare, exercitând funcţia de numărare dependent de aceasta stare iniţială.

1.1.2. Latenţa defectării şi latenţa detecţiei

În [ChIy - 89], latenţa defectării (fault latency) este definită ca intervalul de timp dintre apariţia evenimentului de defectare fizică şi subsecvenţa corupţie a datelor cauzată de eroare. Cu cât un defect este latent mai mult timp, cu atât este mai probabil ca aria de contaminare să fie mai mare, degradând prezumpţia de defect singular care stă la baza majorităţii strategiilor de tolerare a defectelor. Latenţa defectării constituie un parametru de proiectare esenţial la concepţia strategiei de autotestare încorporată (built-in self test) într-un sistem tolerant.

Conform şi cu [LiSh - 90], vom insista asupra acestei problematici considerând un sistem de calcul alcătuit din N module, unde prin modul vom înţelege acea "macrocomponentă" reprezentată de o subunitate bine definită a sistemului, care poate conţine una sau mai multe "microcomponente" defecte şi care poate fi o unitate eminamente hardware, una eminamente software sau o combinaţie hardware / software, având fiecare propriul mecanism de detecţie independent. Un astfel de sistem N modular poate fi reprezentat printr-un digraf D = (V,E), unde V = { v_1, ... ,v_N } denotă setul de noduri, iar E = { e_{ij}, $1 \le i,j \le N$ } denotă setul de arce. Fiecare nod din V reprezintă un modul al sistemului şi un anumit arc e_{ij} reprezintă o legătură de comunicaţie (communication link) de la v_i la v_j. Legăturile de comunicaţie nu trebuie confundate cu legăturile fizice de interconectare (connection

links) dintre module. Metodele tipice de comunicare dintre modulele software sunt constituite de transmisiile de mesaje (message passing) sau memoriile divizate (shared memories), iar modulele hardware pot comunica via semnale de control sau de date. Dacă nu există un link de comunicație de la v_i la v_j, atunci E nu va conține pe e_{ij} sau e_{ij} este un arc nul (null edge).

Toleranța la defectare a unui astfel de sistem multimodular se obține prin fazele secvențiale relevate în fig.1.2. Procesul de defectare (fault process) determină locul şi tipul defectului, precum şi momentul apariției acestuia. Un defect poate induce o eroare (sau erori), care sunt descrise prin procesul de eroare (error process).Pentru a distinge între defect şi eroare, în [LiSh-90] se uzează de definițiile următoare, conforme cu cele statuate de J.C.Laprie şi IFIP Working Group 10.4.

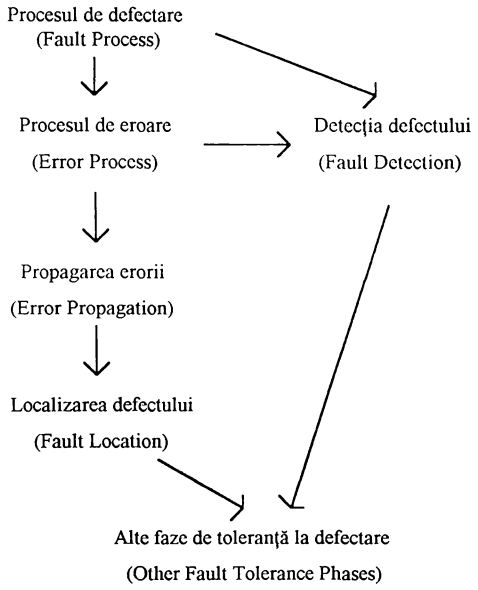


Fig. 1.2.

Un *defect* reprezintă o deteriorare (damage) sau o deviere de la starea normală a unui sistem de calcul pe care se execută sarcini (tasks).

O *eroare* este o deviere de la specificațiile unui anumit task.

Ca exemple enumerăm întreruperea unei legături, o interferență electromagnetică sau o greşeală într-un program în calitate de defecte, în timp ce o eroare poate fi constituită de un semnal de control incorect sau care apare la un moment de timp inoportun.

Cu aceste precizări de terminologie se poate distinge în continuare între detecția de defect (fault detection) şi detecția de eroare (error detection), prima privind punerea în evidență a unui defect prin mijloace diferite de execuția programului, spre exemplu prin mecanisme de testare built-

in, iar cea de-a doua tratând erorile din execuţia programului induse prin defect. După cum se poate observa şi din fig. 1.2, faza de localizare a defectului este apelată numai după detecţia unei erori.

Un modul se spune că este defect (faulty) dacă conţine unul sau mai multe defecte şi se spune că este contaminat (contaminated) dacă conţine o eroare. Fie T_i^F (faulty time) momentul de timp la care apare un defect în v_i şi fie T_i^C (contaminating time) momentul de timp la care apare prima eroare în v_i ca rezultat fie a manifestării unui defect în cadrul lui v_i, fie a propagării unei (sau mai multor) erori dintr-un alt (sau din alte) modul(e).

Dacă v_i este modulul defect, atunci latenţa defectării (fault latency) lui v_i, notată cu L_i, este definită ca intervalul de timp dintre T_i^F şi T_i^C. Viteza de apariţie a defectelor în v_i este caracterizată prin ciclul de defectare (fault cycle) a lui v_i, notat cu Y_i şi care este intervalul de timp dintre două momente consecutive de defectare ale lui v_i. Parametrii anterior definiţi sunt ilustraţi pe diagrama de timp din fig. 1.3.

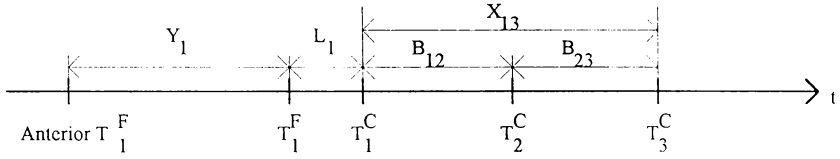


Fig.1.3.

Dacă mecanismul de detecţie al unui modul nu este perfect, defecţiunile din respectivul modul induc erori care apoi se propagă la alte module înainte ca modulele să le detecteze. Pentru a descrie propagarea erorilor într-un sistem, se introduce o cale (path) de propagare de la v_i la v_k, notată prin $(v_i, \ldots, v_k)$, şi definită ca o cale direcţionată în D în care toate nodurile contaminate apar în mod distinct. Se defineşte, de asemenea, timpul de propagare a erorii (error propagation time) de la v_i la v_j, notat cu X_{ij}, drept intervalul dintre momentul de contaminare a lui v_i şi cel de contaminare a lui v_j. Timpii de propagare corespunzători erorilor conţin informaţie completă despre comportamentul (behavior) propagării erorilor şi pot fi măsuraţi experimental. Totuşi, datorită dificultăţilor şi costului de măsurare a timpilor de propagare a erorilor, se defineşte timpul de propagare directă a fiecărui arc e_{ij},nenule, notat B_{ij}, ca timpul pentru propagarea unei erori de la v_i la v_j via e_{ij}. Diferenţa dintre timpul de propagare al erorii şi timpul de propagare directă constă în: 1)ultimul este asociat unui anumit arc iar primul este definit pentru orice pereche ordonată de module, şi 2) ultimul ia în considerare propagarea erorii de-a lungul unui arc particular, în timp ce primul reprezintă timpul de propagare minim luând în considerare toate căile de propagare posibile dintre modulele perechii date. Pentru determinarea relaţiilor dintre mărimile X_{ij} şi B_{ij}, se identifică mai întâi toate căile de propagare a erorii de la v_i la v_j şi se calculează timpul de propagare de-a lungul fiecărei căi prin însumarea timpilor de propagare directă corespunzători tuturor arcelor dintr-o cale. Apoi, se determină X_{ij} drept valoare minimă corespunzătoare tuturor timpilor de propagare a tuturor acestor căi. De exemplu, pentru sistemul având asociat graful din fig. 1.4,pentru, spre exemplu, X_{13}, avem:

$$X_{13}=\min(B_{12}+B_{23},\ B_{14}+B_{45}+B_{53},\ B_{14}+B_{45}+B_{52}+B_{23},\ B_{12}+B_{24}+B_{45}+B_{53}) \qquad (1.4)$$

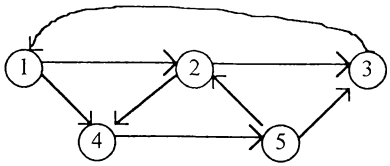


Fig. 1.4.

Toate mecanismele de detecție evidențiază doar erori, astfel că defectele se impune a fi "constrânse" să provoace erori detectabile. Bazat pe metodologia de detecție, mecanismele de detecție pot fi clasificate în trei tipuri: 1) de detecție a defectelor, 2) de detecție a erorilor, și 3) diagnosticări periodice. În calitate de mecanisme de detecție a defectelor (sau detecție la nivel de semnal) enumerăm schemele de autotestare încorporată (built-in self-checking circuits), coduri detectoare de erori și scheme cu duplicare complementară (duplicate complementary circuits). Detecția erorilor (sau detecția la nivel de funcție) poate fi implementată prin teste de acceptanță, controlul la coduri de operații invalide, scheme watchdog-timer. Diagnosticările periodice sunt întreprinse prin programe de testare off-line care sunt periodic rulate și care stimulează modulele sistemului cu intrări imitând pe cele corespunzătoare funcționării normale în vederea activării defectelor și a subsecvenței lor detectări.

Caracteristica distinctivă a mecanismelor de detecție a defectelor constă în faptul că permit punerea în evidență imediat după apariția defectelor, astfel încât nu se ajunge la propagarea erorilor. Defectele care pot fi evidențiate prin mecanismele de detecție din cadrul unui modul sunt definite ca FD-detectabile. Acoperirea detecției defectelor (fault detection coverage) a lui v_i, constituie probabilitatea ca un defect din v_i să fie FD-detectabil.

O diagnosticare periodică poate, în mod uzual, să detecteze mai multe defecte la un cost mai redus decât un mecanism de detecție a defectelor. Deoarece în intervalul dintre două testări de diagnosticare succesive nu pot fi detectate defecte, se evidențiază dezavantajul acestui mecanism constând în faptul că pot fi induse erori care se pot propaga la alte module înainte de a fi localizate. Defectele care nu sunt FD-detectabile, dar care pot fi reliefate prin diagnosticările periodice sunt definite ca PD-detectabile. Constituie o practică răspândită să se execute un scurt program de diagnosticare în timpul operării normale și să se execute un program de diagnosticare complet pe măsură ce apare o astfel de necesitate sau când sistemul este în aşteptare (inactiv). Testele de diagnosticare periodică nu trebuie activate la intervale uniforme de timp, dar intervalul maxim la care trebuie executate este, în mod uzual, fixat.

Defectele care nu sunt nici FD-detectabile, nici PD-detectabile, sunt considerate nedetectabile. Acestea pot fi captate doar în faza de localizare (fault location phase) după ce ele au indus erori care au fost detectate prin mecanismele de detecție a erorilor. Bazat pe locul în care este detectată eroarea dintr-un modul, mecanismele de detecție a erorilor sunt divizate în două grupe: scheme de detecție internă și scheme de detecție a graniță (boundary detection schemes). Pentru un anumit modul, schemele de detecție internă se referă la toate mecanismele de detecție a erorilor care controlează structura internă a acestuia, în timp ce schemele de detecție "boundary" controlează

ieşirile modulului. Se exemplifică mecanismele de detecţie internă prin testele de acceptanţă, iar cele de detecţie "boundary" prin votarea NMR (N modular redundancy) aplicată la ieşirile unui modul. Se defineşte drept latenţă de detecţie internă (internal detection latency) a lui v_i, notată prin I_i, intervalul de timp din momentul contaminării lui v_i până în momentul când a fost detectată o eroare prin scheme de detecţie internă în v_i. Pe de altă parte, este definită latenţa de detecţie "boundary" (boundary detection latency) pe e_{ij}, notată cu J_{ij}, drept intervalul de timp din momentul contaminării lui v_j până în momentul detectării unei erori pe partea lui v_i a lui e_{ij} printr-o schemă de detecţie "boundary". Pentru orice modul v_i se poate defini timpul de detecţie drept momentul de timp la care este detectată prima eroare în v_i fie printr.o schemă de detecţie internă, fie printr-una de detecţie "boundary". Latenţa de detecţie a lui v_i, notată cu K_i, este atunci intervalul din momentul contaminării lui v_i până la momentul său de detecţie. În mod tradiţional, acoperirea detecţiei erorii (error detection coverage) este definită ca probabilitatea de a detecta erorile care există într.un modul sau un sistem, ea fiind utilizată ca o măsură pentru evaluarea eficienţei mecanismelor de detecţie a erorilor.

Latenţa detecţiei poate fi privită ca un rezultat al competiţiei dintre schemele de detecţie internă şi cele "boundary", adică

$$K_i=\min (I_i, B_{ik1}+J_{ik1}, \ldots , B_{ikn}+J_{ikn}) \qquad (1.5)$$

cu $\{e_{ik1}, \ldots , e_{ikn}\}$ reprezentând setul tuturor arcelor care pleacă din v_i.
Diferitele latenţe definite în contextul detecţiei erorilor sunt ilustrate în fig. 1.5.

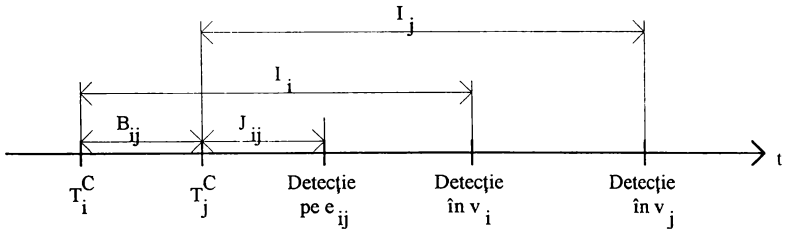


Fig. 1.5.

Este de remarcat că anterioara definiţie a acoperirii detecţiei erorii nu specifică latenţa detecţiei, implicând de asemenea faptul că dacă o eroare nu este detectată într-un interval de timp limitat, aceasta se consideră nedetectabilă pentru totdeauna. Prin urmare este mai adecvat de a defini acoperirea detecţiei erorilor ca o funcţie de timp, pe care o notăm $C_i(t)$, reprezentând funcţia de distribuţie cumulativă a lui K_i şi care implică faptul că o eroare este nedetectabilă doar când latenţa detecţiei asociată acesteia tinde spre infinit.

Admiţând că I_i, B_{ij} şi J_{ij} din (1.5) sunt independente, $C_i(t)$ poate fi obţinută prin

$$1-C_i(t)=(1-F_i^I(t))(1-F_{ik1}^{B*J}(t)) \ldots (1-F_{ikn}^{B*J}(t) \quad (1.6)$$

în care $F_i^I(t)$ reprezintă funcția de distribuție a variabilei aleatoare I_i, iar $F_{ikj}^{B*J}(t)$ reprezintă funcția de distribuție a variabilei aleatoare $B_{ikj} *_{Jikj}$, unde * denotă convoluția, având și

$$f_{ikj}^{B*J}(t) = f_{ikj}^{B}(t) * f_{ikj}^{J}(t) \quad (1.7)$$

în care $f_{ikj}^{B}(t)$ și $f_{ikj}^{J}(t)$ reprezintă funcțiile densitate de probabilitate ale variabilelor aleatoare B_{ikj} respectiv J_{ikj}.

Motivația insistenței asupra referinței [LiSh - 90] constă în introducerea unei terminologii și a unui model la care aderăm. Acesta din urmă admite ipoteze mai realiste decât clasicul model Preparata-Metze-Chien [PrMC- 67] și extensiile acestuia [HaNa-84, SoAA-87, SoAA-89, ș.a.], relaxând restricția referitoare la acoperirea defectelor prin testele de diagnoză și permițând o definire mai intimă, mai adâncă a modulelor sistemului. În [LiSh-90] se fac, de asemenea, referiri pertinente la conceptul de comparare, important prin prisma lucrării pe care o dezvoltăm. Sintetizând problematica legată de comparare, ideea rezidă în detectarea și identificarea unității defecte prin intermediul sarcinilor de calcul utilizator (user tasks). Presupunând că este disponibil un set de unități procesoare identice, un task (sau job) este executat concomitent de către o pereche selectată de procesoare, iar rezultatele celor două unități sunt comparate. Neconcordanța dintre rezultate indică faptul că una dintre cele două unități procesoare este defectă, ea putând fi identificată dacă este efectuat un număr suficient de mare de comparații, care pot fi executate prin module hardware (hardware matchers) sau prin executarea task-ului de către o unitate procesoare, alta decât cele două dintre care una este cea defectă. Avantajele conceptului de comparare constau în: 1) acoperirea unui număr mare dintre problematicele defecte tranzitorii și intermitente, și 2) mijloace hardware/software de anvergură rezonabilă implicate în mecanismele de detecție și programele de diagnosticare. Totuși, utilitatea conceptului de comparare este limitată deoarece 1) sunt acoperite doar defectele din unitățile procesoare, 2) diagnoza unității defecte poate revendica timp îndelungat, și 3) nu sunt luate în considerare comunicațiile dintre task-uri și, prin urmare, este ignorată în mod uzual propagarea erorilor prin intermediul task-urilor.

1.2 Modelarea defectelor

Este de neimaginat o sinteză a unui sistem tehnic în general, în particular și a unuia de calcul, care, în tendința de tolerare a defectelor, să permită acoperirea tuturor malfuncționărilor potențiale. Conectat la această idee apare drept evident faptul că unul din obiectivele preliminare ale concepției unui sistem de calcul tolerant la defecte este tocmai delimitarea, luând în considerare aspecte specifice ale aplicației, acelei clase (sau acelor clase) de defecte care prin probabilitatea lor de apariție superioară reclamă adoptarea de măsuri tehnice menite a asigura deziderатul ca efectele prezenței lor să nu fie resimțite la nivelul aplicației. Dar pentru a stabili cu claritate aceste obiective, plecând și de

la datele de literatură relativ ambigue, inserăm acest paragraf în scopul trecerii în revistă a principalelor modele de defectare, insistând asupra unora, de dată mai recentă, apărute o dată cu progresele tehnologice specifice stadiului de integrare pe scară foarte largă.

Conform cu [Lala-85], efectul unei malfuncționări este reprezentat prin intermediul unui model care pune în relief modificarea semnalelor provocată de către defect. Aceeași sursă bibliografică defalcă modelele defectelor în :

1.Defect de blocare (Stuck-at fault)

2.Defect de scurtcircuit (Bridging fault)

3.Defect de blocare la întrerupere (Stuck-open fault)

1.2.1.Defecte de blocare (Stuck-at Faults)

Cel mai răspândit model pentru acea categorie de defecte cuprinse sub genericul de logice, care determină ca valoare logică dintr-un punct al schemei să devină opusă celei specificate, este "defectul singular de blocare" ("single stuck-at faults"). Acesta presupune ca malfuncționarea să fixeze pe intrarea sau ieşirea unei porți logice fie valoarea logică 0 (stuck-at-0, abreviat s-a-0), fie valoarea logică 1 (stuck-at-1, abreviat s-a-1).

Modelul de blocare, deseori denumit model de defectare clasic, oferă o bună reprezentare pentru majoritatea tipurilor de alterări a bunei funcționări cauzată de scurtcircuite ("shorts") şi întreruperi ("opens") în variate tehnologii. Astfel, pentru ilustrare, să considerăm, pentru început, poarta fundamentală realizată în tehnologie TTL reprezentată în fig.1.6, pe care s-au marcat prin literele *a, b* şi *c* principalele moduri de defectare prin întrerupere, iar prin literele *d* şi *e* principalele moduri de defectare prin scurtcircuit [Lala-85].

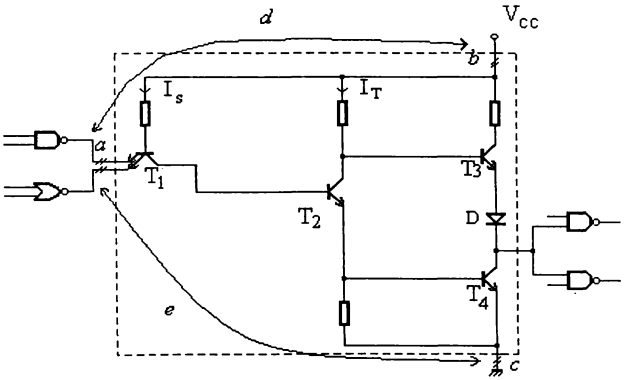


Fig.1.6

a) *Linie de semnal întreruptă (Signal line open)*

Acest defect blochează trecerea curentului I_S prin emiterul tranzistorului de intrare T_1 înspre ieşirea porţii precedente, determinând ca intrarea să apară a fi conectată la un nivel constant corespunzător valorii logice 1, deci defect s-a-1.

b) Întreruperea tensiunii de alimentare (Supply voltage open)

În acest caz, poarta este privată de alimentarea cu tensiune şi astfel este blocată curgerea curenţilor I_S, care ar putea comuta tranzistorul T_1, şi I_T, care ar putea excita tranzistorul T_3. Ambele tranzistoare de ieşire sunt deconectate şi ieşirea porţii apare a fi întreruptă, astfel că defectul poate fi interpretat ca blocare la 1 logic (s-a-1) pe ieşirea porţii.

c) Întreruperea barei de masă (Ground open)

Acest defect determină tranzistoarele T_2 şi T_4 să nu conducă şi astfel curentul I_T va ţine tranzistorul T_3 în permanentă stare de conducţie, astfel încât pe ieşire se va manifesta valoarea normală de 1 logic, adică defectul poate fi interpretat prin blocarea 1 logic (s-a-1) pe ieşirea porţii.

d) Scurtcircuit între un conductor de semnal şi V_{CC} (Signal line and V_{CC} short - circuited)

Un astfel de defect, evident de tipul s-a-1, are însă efectul secundar de supraâncărcare a tranzistorului T_4 din poarta precedentă putând cauza distrugerea acestuia prin putere disipată excesivă.

e) Scurcircuit între un conductor de semnal şi bara de masă (Signal line and ground short - circuited)

Un astfel de defect, evident de tipul s-a-0, are şi un efect secundar,în acest caz nesemnificativ, provocând încărcarea tranzistorului T_3 din poarta precedentă prin curentul de scurtcircuit.

Pe lângă defectele singulare, modelul de blocare poate fi, de asemenea, utilizat pentru reprezentarea acelei grupări de defecte logice cunoscute sub denumirea de defect multiplu de blocare ("multiple stuck-at fault"),care presupune ca simultan într-o schemă logică să fie ţinute mai multe linii de semnal la 0 sau 1 logic. O variaţie a acestei ultime categorii de defectare o constituie aşa numitele "defecte unidirecţionale" ("unidirectional faults") caracterizate prin faptul că toate defectele constituente ale celui multiplu sunt fie blocări la 0 logic (s-a-0), fie blocări la 1 logic (s-a-1), dar nu ambele simultan.

Modelul de blocare a câştigat o largă acceptanţă în trecut datorită, cu precădere, succesului de aplicare relativ la nivelul de integrare pe scară joasă. El îşi pierde din eficienţă în contextul creşterii densităţii de împachetare, astfel încât corespunzător stadiului de integrare pe scară largă şi foarte largă, dominant realizate în tehnologii MOS, sfera lui de descriere comportamentală se restrânge semnificativ.

Să considerăm, pentru ilustrare, schema logică reprezentată în fig.1.7, care implementează în tehnologie n-MOS funcţia booleană:

$$Z = \overline{(A+B)(C+D)+EF} \tag{1.8}$$

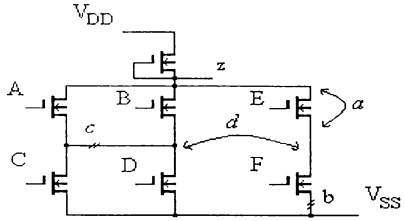


Fig. 1.7

Unele dintre defectele posibile pot fi descrise prin modelul de blocare, altele în schimb nu permit aceasta. Astfel, referindu-ne la malfuncționarea cauzată de scurtcircuitul sursă-drenă a tranzistorului comandat prin variabila E (defect marcat cu litera *a* pe fig. 1.7), aceasta poate fi modelată prin blocarea la 1 logic a respectivei intrări, conducând în consecință la modificarea funcției logice realizată de schemă în:

$$Z_a=\overline{(A+B)(C+D)+F} \tag{1.9}$$

Într-o manieră asemănătoare, defecțiunea de întrerupere marcată cu litera *b* pe fig.1.7, constând din lipsa legăturii echipotențiale dintre electrodul sursă al tranzistorului comandat cu variabila F şi substratul conectat la masă, poate fi modelată cu blocarea la 0 logic (s-a-0) a oricăreia dintre intrările E sau F, sau a ambelor, ceea ce are drept consecință modificarea funcției logice realizată de schemă în:

$$Z_b = \overline{(A+B)(C+D)} \tag{1.10}$$

Pe de altă parte, defectul de întrerupere marcat cu litera c pe fig. 1.7 nu mai poate fi modelat prin blocarea uneia sau mai multor conexiuni ale schemei la una dintre stările logice definite, el având drept efect modificarea funcției realizată de schemă în:

$$Z_c = \overline{AC+BD+EF} \tag{1.11}$$

Un comportament, în linii mari, asemănător este propriu şi defectului marcat cu litera *d* pe fig.1.7, reprezentat de o punte nedorită, a cărui efect nu poate fi descris prin clasicul model de blocare, fiind un exemplu de neaplicare a modelului de blocare. Consecința acestui defect o reprezintă modificarea funcției logice realizată de schemă în:

$$Z_d= \overline{(A+B+E)(C+D+F)} \tag{1.12}$$

În acelaşi context al contraexemplelor de aplicare a modelului de blocare mai inserăm schema din fig.1.8, constând din două porți n-MOS afectate de puntea nedorită marcată cu *x*. În lipsa acesteia, la funcționare normală, cele două ieşiri independente realizează funcțiile logice $Z_1=\overline{AB}$, respectiv $Z_2=\overline{CD}$. În prezența scurtcircuitului, funcțiile logice corespunzătoare celor două ieşiri devin identice:

$$Z_1 = Z_2 = \overline{AB + CD} \qquad (1.13)$$

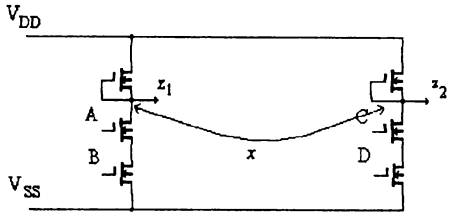


Fig.1.8

Vom insista, într-o oarecare măsură, asupra acestui ultim mod de defectare, urmărind evidențierea unui alt aspect comportamental constând din modificarea spectrului de închidere a curenților la masă. Această fațetă a analizei obține valențe de mare însemnătate în contextul strategiei de testare aflată încă în fază de maturizare constând din verificarea bazată pe termografie,dar şi în contextul testării valorii curentului absorbit de la sursă (i_{DD} testing) aplicată complexelor structuri integrate în tehnologie CMOS.Bazat tocmai pe valorile mai mari ale curenților specifice tehnologiei TTL în raport cu realizările MOS care fac posibilă-prin mijloace optice de captare în infraroşu deocamdată disponibile industrial-punerea în relief a modificărilor de câmp termic provocate de către defect, vom transla analiza, la schema din fig.1.9, sintetizată cu porți fundamentale TTL.

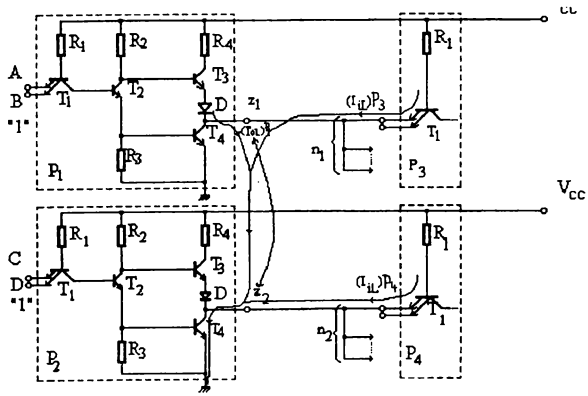


Fig.1.9

Admitem că, la funcționarea normală, poarta P_1 comandă n_1 sarcini TTL, una dintre acestea fiind reprezentată de intrarea porții P_3, și, independent de acest lanț logic, poarta P_2 comandă n_2 sarcini TTL, dintre care una este constituită de intrarea porții P_4. Din punct de vedere logic, puntea nedorită *x* are efectul modificării funcțiilor corespunzătoare celor două subscheme făcându-le identice și date de ecuație booleană (1.13). Considerând, pentru simplitate, că intrările B și D sunt funcțional fixate la 1 logic, tabelele de adevăr-corespunzătoare, pe de o parte, absenței defectului, și, pe de altă parte, prezenței acestuia- reprezentate în fig.1.10, reliefează, prin contrast, deviația comportamentală din punct de vedere logic cauzată de puntea nedorită. Mai apare însă anterior menționatul aspect de modificare a distribuției curenților absorbiți de la sursă, pe care îl detaliem, în cele ce urmează , prin următoarea analiză originală.

	Intrări		Ieşiri normal		Ieşiri defect	
	A	C	Z_1	Z_2	Z_1	Z_2
L_1	0	0	1	1	1	1
L_2	0	1	1	0	**0**	0
L_3	1	0	0	1	0	**0**
L_4	1	1	0	0	0	0

Fig.1.10

Limitându-ne, fără a pierde din generalitate, la situația de stimulare a schemei descrisă de linia L_2 din fig.1.10- după cum se poate observa, doar cazurile corespunzătoare liniilor L_2 și L_3 permit detecția malfuncționării -, consumul de curent $(\sum i_{cc})_{as}$ al întregii scheme din fig.1.9, în absența scurtcircuitului, este dat de relația:

$$(\sum i_{cc})_{as} = (n_1 + 1)i_{ccL} + (n_2 + 1)i_{ccH} \qquad (1.14)$$

în care + reprezintă suma aritmetică, ca de altfel pe întreg parcursul acestei analize, iar $i_{ccL} \cong 3mA$ și $i_{ccH} \cong 1mA$ constituie cunoscuții curenți absorbiți de la sursa de alimentare de către o poartă fundamentală ce se află în stare 0 logic, respectiv 1 logic.

Pe de altă parte, prezența defectului determină modificarea consumului de curent $(\sum i_{cc})_{as}$, acesta devenind:

$$(\Sigma i_{cc})_{ps} \cong (i_0)_{P1} + i_{ccL} + (n_1 + n_2 + 1)\, i_{ccH} \qquad (1.15)$$

în care $(i_0)_{P1}$ (fig.2.9.)constitue curentul de ieşire al porții P_1.

Ţinând cont de notaţiile specifice parametrilor caracteristici etajului final al unei porţi fundamentale şi atribuind acestora indicii P_1, respectiv $P_{2,}$, dependent de poarta căreia îi aparţin, pentru mărimea $(i_0)_{P1}$ rezultă:

$$(i_0)_{P1} = \frac{[V_{cc} - (V_{BET3})_{P1} - (V_D)_{P1} - (V_{CET4})_{P2}]\,[(\beta_{T3})_{P1} + 1]}{(R_2)_{P1}} \qquad (1.16)$$

Uzitând în această relaţie de valori tipice pentru parametri (V_{cc}= 5V, $(V_{BET3}) \cong (V_D)_{P1} \cong$ 0,7V, $(V_{CET4})_{P2} \cong$0,4V, $(\beta_{T3})_{P1} \cong$20 şi $(R_2)_{P1}$=1,6KΩ), rezultă $(i_0)_{P1} \cong$42mA.

Prin puntea nedorită *x*, în cazul cel mai defavorabil prin prisma strategiei de testare bazată pe termografie, când încărcarea celor două porţi de comandă (P_1 şi P_2) este la limita inferioară (n_1=1, n_2=1), o valoare pentru intensitatea curentului cumulat prin punte (fig.1.9) este de $(\Sigma I)_x = (i_0)_{P1} + (i_{iL})_{P2} \cong$42mA + 1,6mA =43,6 mA. Această valoare, superioară cu un ordin de mărime valorilor intensităţilor curenţilor care traversează conexiunile neafectate de malfuncţionare (excepţie fac barele de alimentare), trebuie coroborată cu caracteristicile dimensionale ale punţii, care stabilesc rezistenţa şi, implicit puterea disipată şi căldura emisă de aceasta. Desigur, varietatea dimensională a scurtcircuitelor este considerabilă, dar mai probabil este ca secţiune acestora fie mai mică decât a celorlalte conductoare, aspect favorabil reliefării defectului prin termografie.

Dacă urmărirea prezenţei scurtcircuitului din schemă se realizează la nivelul barei de alimentare (V_{cc}), atunci rezoluţia mijloacelor optice şi de prelucrare a imaginilor termice trebuie să satisfacă relevarea curentului diferenţă ΔI_{cc} a cărui valoare, ţinând cont de (1.14) şi de (1.15), este dată de relaţia:

$$\Delta i_{cc} = (\Sigma i_{cc})_{ps} - (\Sigma i_{cc})_{as} \cong (i_0)_{P1} - n_1(i_{ccL} - i_{ccH}) \qquad (1.17)$$

Pentru a fi relevantă în scopul detecţiei defectului, această variaţie de curent nu trebuie să fie acoperită de variaţiile de curent care apar la funcţionarea normală. Considerând că schema poate fi baleiată exhaustiv prin prisma vectorilor liniari aplicabili la cele două intrări, A şi C, în tabelul din fig.1.11 se prezintă consumurile specifice celor patru stări de excitare.

	Intrări		$(\Sigma I_{cc})_{as}$	$(\Sigma I_{cc})_{ps}$
	A	C		
L_1	0	0	$(n_1+n_2)I_{ccL}+2I_{ccH}$	$(n_1+n_2)I_{ccL}+2I_{ccH}$
L_2	0	1	$(n_1+1)I_{ccL}+(n_2+1)I_{ccH}$	$(I_0)_{P1}+I_{ccL}+(n_1+n_2+1)I_{ccH}$
L_3	1	0	$(n_2+1)I_{ccL}+(n_1+1)I_{ccH}$	$(I_0)_{P1}+I_{ccL}+(n_1+n_2+1)I_{ccH}$
L_4	1	1	$2I_{ccL}+(n_1+n_2)I_{ccH}$	$2I_{ccL}+(n_1+n_2)I_{ccH}$

Fig. 1.11

Analizând datele din tabel, se poate remarca că variațiile de curent sunt dependente de încărcările, n_1 și n_2, ele fiind (cu excepția tranziției de la starea corespunzătoare liniei L_2 la cea corespunzătoare liniei L_3) cu atât mai mari cu cât avem valori mai mari pentru n_1 și n_2. Astfel, la trecerea din starea de excitație corespunzătoare liniei L_1 la cea corespunzătoare liniei L_4, în funcționare normală avem:

$$(\Delta i_{cc})_{as\text{-}L1/L4} = (n_1 + n_2 - 2)(i_{ccL} - i_{ccH}) \qquad (1.18)$$

Apelând la valorile de încărcare maxim admise ($n_1=n_2=10$, exceptând deci circuitele de putere), rezultă $(\Delta i_{cc})_{as\text{-}L1/L4} \cong 36mA$.

Pe de altă parte, la funcționarea în prezența defectului, se poate observa din fig.1.10, că intervin ca esențiale doar două variații și anume $(\Delta i_{cc})_{ps\text{-}L2/L1} \cong (i_0)_{P1} - (n_1+n_2-1)(i_{ccL}-i_{ccH})$ și $(\Delta i_{cc})_{ps\text{-}L2/L4} \cong (i_0)_{P1} - (i_{ccL}- i_{ccH})$. Ori, luând în considerare prima dintre acestea, cea mai defavorabilă, rezultă în condițiile ipotetice de încărcare specificate, $(\Delta i_{cc})_{ps\text{-}L2/L1} \cong 4mA$, valoare mult mai mică decât cea stabilită anterior pentru $(\Delta i_{cc})_{as\text{-}L1/L2}$. Concluzia imediată este că această strategie a defectului considerat este deficitară.

Continuând investigația, rezultă condiția pentru care totuși metoda de urmărire a variațiilor de consum pe bara V_{cc} ar fi aplicabilă, și anume stabilind restricția: $(\Delta i_{cc})_{ps\text{-}L2/L1} > (\Delta i_{cc})_{as\text{-}L1/L4}$, obținem $(n1+n2) < 11$.

Pe de altă parte, excluzând din setul exhaustiv de excitație anumiți vectori binari, cum ar fi spre exemplu, cel corespunzător liniei L_4, situația, prin prisma urmărită, se ameliorează simțitor. Mărimea dată de (1.18), va fi substituită de una dintre $(\Delta i_{cc})_{as\text{-}L1/L2} = (n_2-1)(i_{ccL}-i_{ccH})$ respectiv $(\Delta i_{cc})_{as\text{-}L1/L3} = (n_1-1)(i_{ccL}-i_{ccH})$. Apelând, din nou, la valori maxime de încărcare, conform celor expuse mai sus, rezultă, în ambele cazuri, doar $(\Delta i_{cc})_{as} \cong 18mA$. Uzitând de raționamente similare, deducem pentru această situație condițiile restrictive $(2n_1+n_2)<23$, respectiv $(n_1+2n_2)<23$, care relaxează anterioara restricție $(n_1+n_2)<11$.

Concluziile rezultate, în dependență de o anumită strategie adoptată pentru acoperirea tipului de defect supus analizei, pot fi grefate, în variate moduri, pe uneltele existente de generare asistată de calculator a vectorilor binari de stimulare. Ele pot constitui, de asemenea, elemente de analiză ale testabilității schemei raportat la tipul de defect considerat. În tot cazul, concluzia certă este că modelul de blocare este deficitar relativ la acest tip de defectare a cărui probabilitate de apariție este ridicată, în special, în faza de producție a sistemelor de calcul și care este mai puțin probabil să apară în faza de exploatare a acestora.

În fine, un ultim exemplu de defect care nu poate fi acoperit prin clasicul model de blocare este așa numitul defect de punct de intersecție (crosspoint fault), care este tipic pentru, răspânditele în sinteze, rețele logice programabile (programmable logic arrays, PLA). O astfel de realizare, în tehnologie MOS, constând din cunoscuta implementare pe două niveluri, în acest caz ȘI-SAU, a funcțiilor booleene, este prezentată în fig.1.12a, iar schema logică echivalentă este dată în fig.1.12b [Fuji-85].

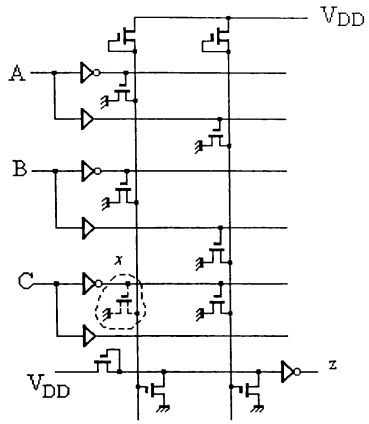


Fig.1.12a

Structura regulată matricială a unui circuit PLA este divizată în două submatrici (una din circuite ŞI pentru implementarea termenilor şi una de circuite SAU pentru implementarea funcţiilor booleene), fiecare dintre acestea având integrat, în toate punctele de intersecţie ale barelor orizontale cu cele verticale, câte un dispozitiv (diodă sau tranzistor), chiar dacă unele dintre acestea nu sunt utilizate. Conexiunea unui astfel de dispozitiv este programată şi, la această operaţie pot să apară defecte în sensul omiterii sau programării în exces a unor legături.

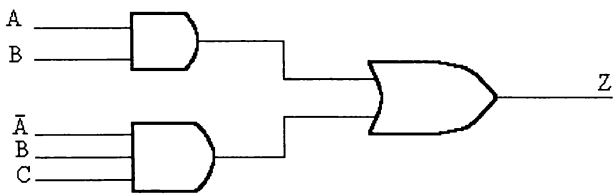


Fig.1.12b

Cu toate că multe dintre aceste defecte de puncte de intersecţie pot fi reprezentate prin modelul de blocare, unele dintre ele, cum ar fi, spre exemplu, conectarea tranzistorului din punctul de intersecţie *x* (fig.1.12a), nu pot fi acoperite prin acest model clasic. Astfel, în absenţa defectului, ieşirea z realizează funcţia logică = $AB+\overline{A}\,\overline{B}C$, iar în prezenţa defectului de punct de intersecţie realizează funcţia logică $Z = ABC + \overline{A}\,\overline{B}C$, care nu poate fi obţinută prin nici un defect de blocare aplicat la schema logică echivalentă din fig.1.12b.

1.2.2. Defecte de scurtcircuit

Scurtcircuitele formează o importantă clasă de defecte permanente care nu sunt acoperite prin modelul de blocare. În interiorul unei capsule de circuit integrat, ele pot să apară prin străpungerea izolației dintre două straturi de metalizare adiacente sau, în cadrul aceluiași strat de metalizare, prin imprecizii la operațiile de transpunere a măștilor. La nivelul plachetei cu cablaj imprimat ele constau din scurtcircuitarea nedorită a unor pini sau trasee, predilect cu poziții adiacente, și pot fi cauzate prin insuficiențe variate în procesele de corodare sau lipire, precum și în cel de realizare a găurilor de trecere între straturi [Lala-85].

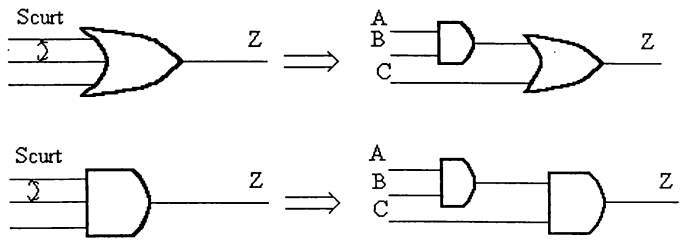
Dacă avem o schemă care cuprinde n conexiuni (sau noduri), atunci aceasta prezintă 2n defecte singulare de blocare și $2^2C_n^2+2^3C_n^3+.....+2^nC_n^n = 3^n - 1 - 2n$ defecte multiple (exceptându-le pe cele singulare) de blocare. Admițând că s noduri ale schemei considerate pot fi în scurtcircuit, doar numărul defectelor singulare de acest tip atinge valoarea enormă $C_n^s!$. Bineânțeles, aserțiunile anterioare sunt pur teoretice, întrucât defectelor practice trebuie să li se asocieze probabilități reale de apariție.

O primă observație legată de scurtcircuite este că ele depind de tipul de logică, pozitivă sau negativă, specifică tehnologiei în care sunt realizate circuitele utilizate la implementarea schemelor, fapt care este sintetizat în fig.1.13.

Precizând în același sens, dacă două ieșiri de circuite logice sunt scurtcircuitate și este dominantă valoarea logică 0, așa cum este specific tehnologiei TTL (vezi fig.1.9), atunci descrierea efectului acestei malfuncționări este dată de substituirea celor două conexiuni printr-o poartă fictivă care realizează funcția logică ȘI, sau, altfel exprimat, avem un efect de ȘI cablat (wired -AND). Dacă însă pentru circuitele utilizate valoarea logică dominantă este 1, cum este cazul celor realizate în tehnologie ECL, avem de-a face cu un efect de SAU cablat (wired-OR) pentru modelarea circuitului [Fuji-85].

LOGICĂ POZITIVĂ

Circuit original Schemă echivalentă a circuitului defect



LOGICĂ NEGATIVĂ

Circuit original Schema echivalentă a circuitului defect

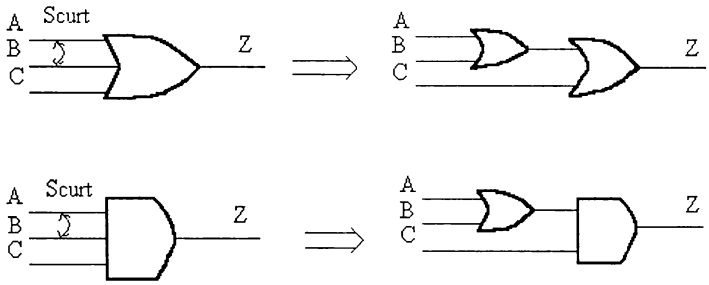


Fig.1.13

Pe de altă parte, este uzuală [Lala-85] clasificarea defectelor de scurtcircuit în următoarele două tipuri:

1.Scurtcircuite de intrare (Input bridging)

2.Scurtcircuite de reacție (Feedback bridging)

Admitem dată schema logică combinațională care implementează funcția $F(x_1,x_2,...x_n)$. Dacă există un scurtcircuit între s linii de intrare ale schemei, atunci avem de-a face cu un scurtcircuit de intrare de multiplicitate s, a cărui model logic este prezentat în fig.1.14. Dacă ieşirea Y a schemei este în scurtcircuit cu s dintre intrările acesteia, atunci avem de-a face cu un scurtcircuit de reacție de multiplicitate s, a cărui model logic este prezentat în fig.1.15.

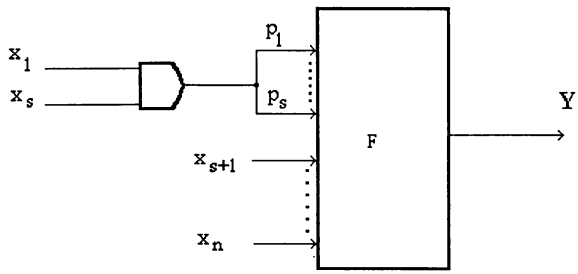


Fig.1.14

Un defect de scurtcircuit de reacție poate determina oscilația unei scheme sau o

Un defect de scurtcircuit de reacție poate determina oscilația unei scheme sau o poate converti dintr-una combinațională într-una secvențială. În prezența scurtcircuitului de reacție (Y, x_1, x_2,.....x_s), o schemă combinațională care implementează funcția $F(x_1,x_2,.....x_n)$ oscilează dacă vectorul binar de intrare $(x_1,x_2,.....x_n)$ satisface următoarea condiție:

$$x_1x_2.....x_s\,F(0,0,..0.x_{s+1},\ ...,x_n)\ \overline{F}(1,1,...,1,x_{s+1},.....,x_n)=1 \qquad (1.19)$$

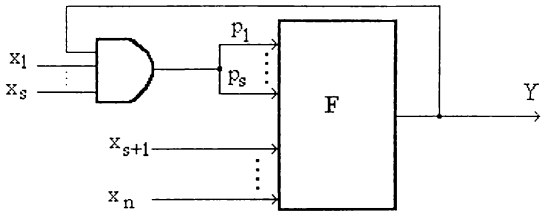


Fig.1.15

Pentru exemplificare să considerăm schema combinațională din figura 1.16 [Lala-85], care implementează cu circuite logice NAND funcția booleană:

$$Y = \overline{x_1 x_2 x_3 x_4} + x_4 x_5 x_6 \qquad (1.20)$$

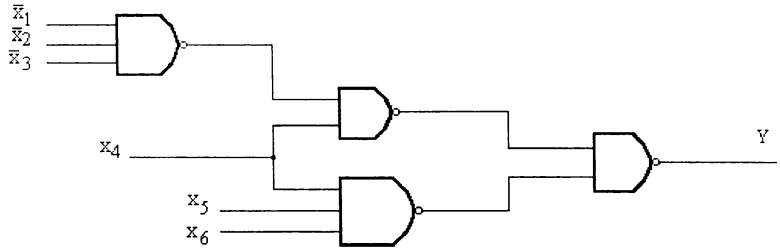


Fig.1.16

Dacă în schemă apare scurtcircuitul ieşirii Y cu intrările x_1 şi x_2, atunci schema se transformă în cea din fig.1.17, şi pentru combinația de intrare $(x_1,x_2,x_3,x_4,x_5,x_6)$ =(1,1,1,1,0,0), ea, îndeplinind condiția (1.19), oscilează.

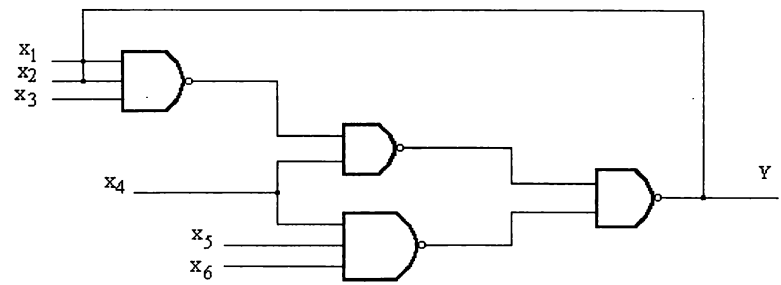


Fig. 1.17

Pe de altă parte, în prezența scurtcircuitului de reacție $(Y,x_1,x_2,....,x_s)$, o schemă combinațională care implementează funcția $F(x_1,x_2,....,x_n)$ devine una secvențială asincronă dacă vectorul binar de intrare $(x_1,x_2,....,x_n)$ satisface următoarea condiție:

$$x_1x_2 \quad x_s \overline{F}\,0,0,....,0,x_{s+1},....,x_n)F(1,1,....,1,x_{s+1},....,x_n)=1 \qquad (1.21)$$

Dacă, în aceeași schemă din fig.1.16 apare scurtcircuitul ieșirii Y cu intrările x_4, x_5 și x_6, atunci schema se transformă în cea din fig.1.18, care, pentru combinația de intrare $x_4 = x_5 = x_6 = 1$ determină $\overline{F}(x_1,x_2,x_3,0,0,0) = 0$ și $F(x_1,x_2,x_3,1,1,1) = 1$, și, prin aceasta, îndeplinirea condiției (1.21). Prin urmare schema din fig.1.18 este secvențială asincronă.

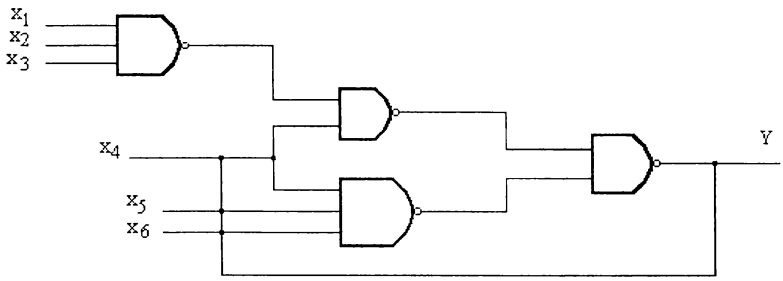


Fig.1.18

1.2.3. Defecte de blocare la întrerupere

O categorie aparte de defecte, specifică circuitelor implementate în tehnologie CMOS- una dintre cele mai uzitate la nivelul de integrare VLSI-, o reprezintă așa numitele defecte de blocare la întrerupere (stuck-open faults, abreviat s-op). Acestea nu sunt echivalente cu clasicele defecte de blocare (s-a-0, s-a-1), diferența majoră dintre ele constând în faptul că cele din urmă, în situația că afectează o schemă combinațională, nu modifică comportamentul combinațional al schemei, pe când defectele s-op dintr-o schemă combinațională o transformă într-una secvențială.

Să considerăm, pentru exemplificare, poarta NAND realizată în tehnologie CMOS cu tranzistoare FET cu canal p și cu canal n (fig.1.19) [Fuji-85], a cărei ieșire *z* prezintă nivel de tensiune corespunzător stării logice 0 dacă și numai dacă ambelor intrări, x_1 și x_2, li se aplică nivel de tensiune ridicat, corespunzător stării logice 1.

În fig.1.19 sunt marcate prin literele *a, b, c* și *d* patru defecte posibile, toate întreruperi (s-op). Astfel, defectul indicat prin litera *a* constă dintr-o întrerupere, sau omitere, de conexiune a intrării x_1 la tranzistorul cu canal p care are rolul de sarcină. În prezența acestui defect, atunci când la x_1 se aplică 0 logic și la x_2 se aplică 1 logic, ieșirea z obține o stare nedorită de înaltă impedanță, nefiind conectată nici la V_{ss}, nici la V_{DD}, ea reținând starea logică corespunzătoare stării anterioare a ieșirii.

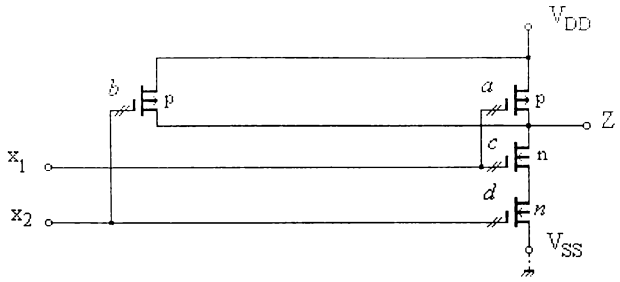


Fig.1.19

Prin aceasta, schema care include poarta afectată de defectul menționat, prezintă un comportament secvențial. Durata stării reținute este determinată de valoarea curentului de scurgere corespunzător nodului din schema logică în care este conectată ieșirea porții. De altfel, în fig.1.20 prezentăm tabelele de adevăr corespunzătoare funcționării normale și, prin contrast, corespunzătoare funcționării afectată de cele 3 tipuri relevante de întreruperi (s-op). În aceste tabele s-a notat prin z_a starea ieșirii care reține valoarea logică corespunzătoare stării anterioare.

Intrări		Ieșirea z			
x_1	x_2	normală	s-op în a	s-op în b	s-op în c sau d
0	0	1	1	1	1
0	1	1	z_a	1	1
1	0	1	1	z_a	1
1	1	0	0	0	z_a

Fig.1.20

Completăm exemplificarea modelării defectelor s-op considerând poarta NOR cu două intrări realizată în tehnologie CMOS (fig.1.21), a cărei funcționare normală și defectă poate fi urmărită în tabelele de adevăr prezentate în fig.1.22. Defectele x_1 s-op, respectiv x_2 s-op, pot fi cauzate prin omiterea conexiunilor de intrare, sursă sau drenă , la tranzistoarele FET T_3, respectiv T_4 (fig.1.21). Cel de-al treilea defect, V_{DD} s-op, poate fi determinat de către o întrerupere oriunde pe calea serială a celor două tranzistoare FET cu canal p la bara de alimentare V_{DD}.

Acest al doilea exemplu confirmă comportamentul secvențial pe care îl conferă schemelor de tip combinațional porțile CMOS afectate prin defecte s-op, ceea ce face neaplicabile strategiile de testare elaborate pe baza modelului clasic de blocare. Cu alte cuvinte, detecția defectelor s-op nu mai poate fi efectuată prin vectori individuali de stimulare, cum este specific defectelor clasice de blocare (s-a-1, s-a-0) în cazul schemelor combinaționale, ci fiecare defect s-op revendică elaborarea unor secvențe de testare în maniera proprie schemelor logice secvențiale.

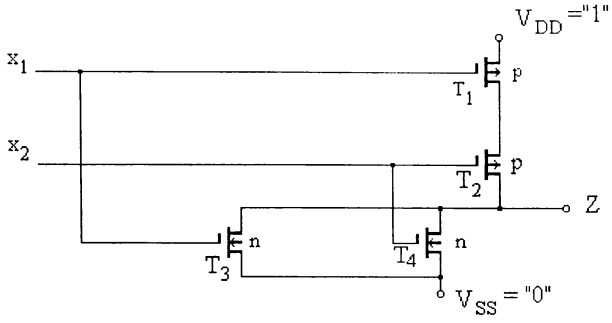


Fig.1.21

Intrări		Ieşire z			
x_1	x_2	normală	x_1 s-op	x_2 s-op	V_{DD} s-op
0	0	1	1	1	z_a
0	1	0	0	z_a	0
1	0	0	z_a	0	0
1	1	0	0	0	0

Fig.1.22

1.3. Defecte temporare

În [Lala-85] se afirmă că o parte însemnată dintre malfuncţionările sistemelor de calcul îşi au cauza în defecte temporare, acestea revendicând, datorită dificultăţilor pe care le ridică la detecţia şi izolarea lor , mai mult de 90% din timpul total de mentenanţă. Cu referire la acest subiect, literatura este, în mare parte, confuză în sensul că defectele temporare sunt denumite fie intermitente (intermittent faults), fie tranzitorii (transient faults), atribuindu-le aceeaşi semnificaţie. Conform cu [ClWa-81] şi [Lala-85], între cele două tipuri de defecte se impune o distingere netă.

Astfel, prin tranzitorii se înţeleg defectele temporare nerecurente, cauzate, în mod uzual, prin radiaţii cu particole α şi fluctuaţii ale sursei de alimentare. Ele nu sunt reparabile întrucât nu sunt cauzate de degradarea fizică a hardware-ului. Defectele tranzitorii constituie sursa majoră a erorilor soft care afectează funcţionarea capsulelor de memorii semiconductoare. De altfel, în [Lala-85] se susţine că funcţionarea circuitelor integrate RAM poate fi alterată prin două tipuri de erori, hard şi soft.

Erorile hard (hard errors) sunt permanente, fiind cauzate de următoarele trei tipuri majore de defecte care se pot produce în capsulele RAM: defecte de metalizare şi conectare la pin (metalization and bounding failures), defecte de oxidare (oxide defects) şi contaminarea cu ioni (ion contamination). Imprecizii la operaţia de metalizare constituie o sursă pentru, cu precădere, întreruperi pe intrările şi ieşirile dispozitivului. Defectele de oxidare pot deteriora întreaga capsulă sau pot afecta un singur bit (single-bit failure), iar contaminarea ionică poate include defecte, cu precădere, ale decodificatoarelor de linie sau coloană.

Pe de altă parte, erorile soft (soft errors) constituie modificări aleatoare nerecurente de stări logice din memorie, în sensul că, spre exemplu, un 1 logic se modifică în 0 logic sau invers. Surse tradiţionale pentru astfel de erori sunt zgomote ale tensiunii de alimentare, perturbaţii sau dispozitive cu valori de parametri la limita câmpurilor tolerate (marginal devices), la care se adaugă, mai recent observatele, particole α emise de impurităţile de uraniu şi toriu din materialele de încapsulare. Defectele datorate emisiei de particole α determină modificarea de stare a unor biţi din capsulele de memorie RAM dinamice, dar ele nu au un efect permanent, respectivele celule putând fi ulterior înscrise şi citite în mod corect. Aceste defecte devin mai probabile cu creşterea densităţii de împachetare a capsulelor RAM dinamice- ceea ce poate fi observat în tabelul din fig.1.23 [Lala-85]-, întrucât scade valoarea capacităţii celulei de memorare şi astfel creşte vulnerabilitatea la erori soft cauzate de radiaţia α.

Densitate (biţi/capsulă)	Viteza de eroare tipică (% la 1000 ore)	
	Erori hard	Erori soft
1K	0,0001	0,001
4K	0,002	0,02
16K	0,011	0,10
64K	0,016	0,13

Fig.1.23

Pe de altă parte, prin intermitenţe se înţeleg acele defecte temporare recurente care reapar în mod regulat. Astfel de defecte pot apare datorită unor contacte imperfecte, a unor componente electronice parţial defecte sau a unor inexactităţi de proiectare. Acele defecte intermitente cauzate de deteriorarea parţială sau îmbătrânirea componentelor electronice pot deveni permanente. Unele dintre aceste defecte pot fi provocate de parametri ai mediului de exploatare cum ar fi temperatura, umiditatea, vibraţii, ş.a. Probabilitatea de manifestare a intermitenţelor depinde de gradul de protecţie - prin ecranare, filtrare, răcire, ş.a. al sistemului în raport cu factorii perturbatori ai mediului. Un defect intermitent dintr-o schemă cauzează malfuncţionarea acestuia numai dacă el este activat, iar

schema se spune că este în stare de defect activ (fenet active state, abreviat FA) dacă defectul în schemă este activat. În mod similar, schema se spune că este în stare de defect neactiv (fault not-active state, abreviat FN) dacă defectul existent în schemă rămâne inactiv.

Întrucât defectele intermitente sunt aleatoare, ele nu pot fi modelate decât prin utilizarea metodelor probabilistice. În literatură au fost prezentate mai multe modele probabilistice pentru descrierea comportamentului defectelor intermitente. Astfel, Breuer a propus un model Markov de ordinul întâi cu două stări (fig.1.24) pentru o clasă specifică de defecte intermitente care sunt "cu comportare bună" ("well behaved") şi "independente de semnal" ("signal independent")[Lala-85]. Un defect intermitent se spune că este cu comportare bună dacă, pe durata aplicării unui vector binar de stimulare, schema testată se comportă fie ca una cu funcţionare normală, fie ca una în care există un defect permanent. În al doilea rând, un defect intermitent se spune că este "independent de semnal" dacă, pe durata cât este activat, nu depinde de intrările aplicate schemei sau de starea curentă a acesteia. Modelul lui Breuer admite că defectul oscilează între starea FA şi starea FN. Probabilităţile de tranziţie indicate pe fig.1.24 depind de pasul de timp selectat şi ele trebuie modificate dacă se schimbă pasul de timp. Lala şi Hopkins au utilizat o adaptare a modelului introdus de Breuer care caracterizează tranziţia din starea FA în satrea FN prin doi parametri α şi δ, denumiţi frecvenţe de tranziţie. Raportul α/δ este denumit factor de latenţă (latency factor) şi are valoarea cu atât mai mare cu cât este mai mică probabilitatea ca defectul să fie activat.

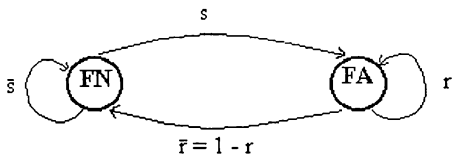


Fig.1.24

Kamal şi Page au introdus pentru defectele intermitente un model Markov de ordinul zero şi au sugerat o procedură de detecţie a defectelor intermitente singulare, cu comportare bună şi independente de semnal din scheme logice combinaţionale fără redundanţă. Modelul admite estimări prealabile ale probabilităţii ca schema să posede un defect intermitent şi ale probabilităţii condiţionate ca defectul să devină activ în situaţia când el este prezent în schemă. Procedura de detecţie a defectelor intermitente propusă foloseşte aplicarea, în mod repetat, a testelor care au fost generate pentru a pune în evidenţă defecte permanente. După aplicarea fiecărui vector de testare, uzitând de regula lui Bayes, este calculată probabilitatea de detecţie a unui defect intermitent prestabilit. Această probabilitate devine 1 dacă testul este repetat de un număr infinit de ori. Totuşi, poate fi găsit un număr finit de repetiţii a testului prin utilizarea uneia dintre următoarele două reguli de decizie. Prima regulă constă în terminarea repetării atunci când probabilitatea posterioară (probabilitatea ca un defect intermitent prestabilit să existe în schemă după aplicarea testului) devine mai mică decât o anumită valoare. Cea de a doua regulă constă în oprirea aplicării testului atunci când "raportul de probabilităţi" (care este o funcţie de probabilităţi posterioare) devine mai mic decât

un număr prag. În mod uzual, numărul de repetări necesare rămâne foarte mare. Modelul Markov de ordin zero a mai fost utilizat de Savir, precum şi de Koren şi Kohavi pentru a descrie comportamentul defectelor intermitente [Lala-85].

Pe de altă parte, pentru defectele intermitente, Su a propus un model Markov cu parametru continuu, care este o generalizare a modelului cu parametru discret introdus de Breuer. În acest model, prezentat în fig.1.25, probabilităţile de tranziţie depind în mod liniar de pasul de timp Δt. De exemplu, dacă schema se află la momentul de timp t în starea FN, probabilitatea ca ea să treacă la momentul de timp$(t+\Delta t)$ în starea FA este proporţională cu Δt. Dacă este admisă o constantă de proporţionalitate λ, atunci probabilitatea este dată de $\lambda\Delta t$.

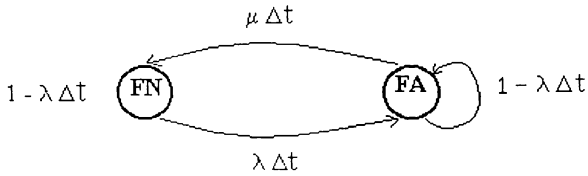


Fig.1.25

În mod similar, proporţionaliatatea de trecere din starea FA la momentul de timp t în starea FN la momentul de timp $(t+\Delta t)$ este $\mu\Delta t$. Perioada de timp pe durata căreia schema rămâne în starea FA(FN) este distribuită exponenţial cu valoarea medie $1/\mu(1/\lambda)$. Dacă pasul de timp Δt este foarte mare, modelul Markov continuu se reduce la un model Markov discret de ordin zero în care caz probabilitatea ca defectul să devină activ este dată de $\lambda/\lambda+\mu$.

Problema majoră a modelelor de defecte intermitente amintite constă în faptul că este foarte dificilă obţinerea datelor statistice necesare verificării validităţii lor. Pentru evitarea acesteia, Stifler propune un model mai complicat constând din cinci stări (fig.1.26) [Lala-85].

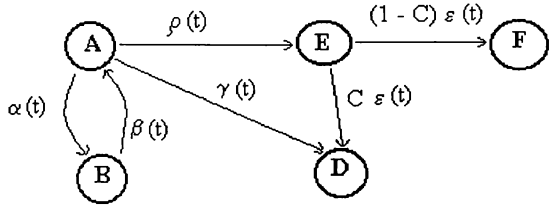


Fig.1.26

Starea A este cea în care defectul este activat, iar B este o stare benignă. Dacă apare defectul, se intră în starea E, iar D este starea în care defectul este detectat. F este starea defectă care rezultă prin propagarea unei erori nedetectate. $\alpha(t)$ reprezintă probabilitatea de tranziţie din starea în care este activat defectul în starea benignă, iar $\beta(t)$ reprezintă viteza de apariţie a tranziţiei din starea benignă în starea în care este activat defectul. $\rho(t)$, $\delta(t)$ şi $\varepsilon(t)$ reprezintă vitezele de apariţie a generării erorii, detecţiei defectului şi propagării erorii. Fiecare dintre aceste probabilităţi de tranziţie

sunt funcţii doar de timpul t consumat în starea sursă. Parametrul C reprezintă probabilitatea de acoperire ("coverage"), care reprezintă probabilitatea de a detecta o eroare înainte ca aceasta să producă o avarie.[Lala-85].

În final, se impune remarcat faptul că există o literatură bogată care acoperă problematica defectelor permanente din hardware-ul sistemelor de calcul, dar rămâne încă deschisă problema defectelor temporare. Pentru a preveni defectarea sistemelor de calcul datorată defectelor temporare se utilizează , pe de o parte, tehnici de mascare a defectelor (fault masking)- care tolerează prezenţa defectelor şi asigură operarea continuă a sistemelor -, şi pe de altă parte, tehnici de detecţie concurentă a defectelor (concurrent fault detection) - care folosesc scheme total autotestabile (" totally self-checking circuits") pentru a semnala prezenţa defectelor, dar care nu maschează defectele.

1.4.Concluzii

Pregătind aplicarea unor metode care să asigure implementarea eficientă a toleranţei la defectare în sisteme de calcul, în acest prim capitol de consistenţă al lucrării au fost supuse analizei aspectele de defectare specifice domeniului calculului care să constituie obiective de combătut prin metodele care vor fi dezvoltate în următoarele capitole.Rezumând, în această parte, lucrarea cuprinde următoarele contribuţii:

a) Un aport în sistematizarea aspectelor de defectare specifice domeniului calculului, precum şi precizări referitoare la folosirea terminologiei specifice.

b) Analiza modului de manifestare al defectului de scurtcircuit, întreprinsă în contextul unui mediu TTL, dar care poate fi extrapolată la actuala problematică a testării prin urmărirea curentului i_{DD} absorbit de la sursa de alimentare specifică realizărilor CMOS.

c) Conturarea unor definiţii clare pentru latenţele de defectare, respectiv de detecţie.

2. Implementarea toleranței la defectare în sisteme de calcul

Odată investigată problematica defecte / erori, se impune conturarea metodelor de implementare a tolerării acestora. Pentru aceasta, în sistemele de calcul, sunt oferite alternative apelând la mijloace hardware, respectiv software [Gork - 89]. Deosebirile dintre aceste metode sunt neesențiale pentru toleranța la defecte, cu atât mai mult cu cât una şi aceeaşi metodă poate să-şi găsească soluția de implementare în ambele variante. Mult mai importantă decât deosebirile de implementare este optimizarea soluției adoptate pentru toleranța la defecte, astfel încât la o investiție dată să fie epuizate disponibilitățile existente, sens în care, de regulă, mai atractivă se prezintă combinarea celor două alternative de implementare.

Înainte de a intra în detaliile principalelor metode hardware de implementare a toleranței la defecte, se impune relevarea anumitor condiții de delimitare (în sensul izolării-decuplării propagării efectelor unei malfuncționări), care necesită o relativ timpurie considerare în faza de concepție a sistemului. Scăparea din vedere a unor asemenea efecte propagate face dificilă, dacă nu imposibilă, eliminarea lor ulterioară. Chiar dacă, în cele ce urmează, vom insista asupra unor caracteristici hardware ale decuplării defectelor, domeniu în care sunt evidente, ele nu sunt totusi mai puțin importante în domeniul software.

2.1. Măsuri de izolare - decuplare a defectelor

Implementarea toleranței la defectare solicită o serie intreagă de măsuri care se impun respectate pe tot parcursul, de la concepție la exploatare, pentru a conferi unui sistem cerințele de fiabilitate care să-l facă competitiv pe piață. Cu precădere sistemele de calcul dedicate unor aplicații de înaltă siguranță trebuie să se supună unor astfel de măsuri, sintetizate în norme. Un exemplu în acest sens îl constituie documentația [VDI - 85], valabilă în Germania, pentru calculatoare destinate conducerii de procese industriale.

Pentru a ne forma o imagine asupra cerințelor statuate prin aceste măsuri vom orienta prezentarea lor în corespondență cu diferite faze ale ciclului unui produs de fiabilitate sporită.

A. *Faza de concepție - proiectare*

Măsurile din faza de concepție vizează preponderent scopul evitării defectelor (fault avoidance), respectiv prevederea de mijloace pentru recunoaşterea stării de defect şi pentru izolarea subansamblului defect, astfel încât efectul malfuncționării să nu contamineze alte subansamble şi în acest fel, să fie asigurată o aşa numită comportare fail - safe a sistemului.

a) Se detaşează prin importanță problematica alimentării, care la rândul ei cuprinde mai multe aspecte

a_1) Se recomandă alimentarea separată a acelor unități tehnologice interconectate prin magistrale. În acest context, pentru fixarea ideilor, admitem că sistemul este structurat tehnologic din plachete montate în sertare prevăzute cu plachete fund de sertar având cablate linii ale unei magistrale de sistem standard. Chiar dacă pe plachetele individuale sunt luate, din punct de vedere

logic, măsuri care să evite propagarea erorilor, nu pot fi excluse defecte ale etajelor de ieşire pe magistrală, astfel încât aceasta constituie calea cea mai vulnerabilă de contaminare, impunând capacitatea de izolare electrică a plachetei suspecte de la magistrala de sistem. În acest sens, fiecare astfel de unitate tehnologică se prevede cu propriul său regulator de tensiune independent, precum şi cu posibilitatea de deconectare a alimentării, astfel încât respectiva plachetă să poată fi izolată electric de magistrala de sistem. Această izolare a modulelor hardware poate fi întreprinsă prin scheme cu tranzistoare cu efect de câmp, care permit decuplarea căilor de semnal critice precum şi a legăturilor de alimentare.

Detaliind acest aspect, ne vom referi,în calitate de exemplu, la minisistemul Parallel / XR [McMu - 86] al firmei Parallel Systems, oferit în variate configuraţii redundante, care asigură o ridicată integritate a datelor precum şi o exploatare neântreruptă. Structura sistemului, ilustrată la nivel de schemă bloc în fig.2.1, constă din perechi de unităţi procesoare paralele (UPP) bazate pe microprocesoare MC 68010, respectiv MC 68020, care execută în manieră sincronă acelaşi flux de instrucţii.

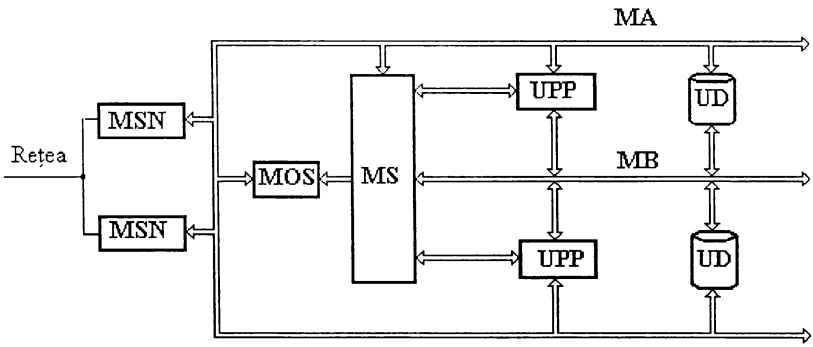


Fig.2.1

Descrisă sumar, funcţionarea este supravegheată de modulul de sincronizare (MS), aflat sub controlul monitorului sistemului (MOS). La defectarea uneia dintre unităţile procesoare, aceasta este substituită printr-o unitate de rezervă, fără a determina întreruperea exploatării. Prin extensii ale sistemului de operare UNIX au fost adăugate rutine de diagnoză, care permit detecţia de defecte, vizând inclusiv modulul de supraveghere, la a cărui cădere este prevăzută exploatarea neredundantă asigurată de către o singură unitate procesoare.Astfel,operaţiile de intrare - ieşire sunt executate de un singur modul de calcul, urmând ca în urma remedierii defectului să aibă loc resincronizarea modulului de calcul rămas inactiv. Unităţile de disc de masă (UD) sunt, de asemenea, dublate pentru a permite funcţionarea în oglindă, toleranţa la căderi. De interes deosebit, pe lângă aplicarea în manieră particulară a principiului redundant al dublării, sunt prevederile (reliefate într-o anumită măsură de schema din fig.2.1) legate de alimentarea cu tensiune şi de facilitarea întreţinerii sistemului de către utilizator. Pe lânga magistralele locale, care asigură traficul informaţional la memorii

operative de pe plachetele cu unitățile procesoare paralele, are loc un flux informațional bidirecțional pe magistrala Multibus (MB). Liniile acesteia sunt protejate la defecte în sensul că toate semnalele unui modul defect pot fi decuplate prin comutatoare suplimentare realizate cu tranzistoare. În acest mod, respectiva unitate tehnologică poate fi izolată din punct de vedere logic de restul sistemului şi poate fi îndepărtată din acesta fără a-l afecta în vreun fel. În plus, este prevăzută o magistrală suplimentară de stare şi alimentare (MA), care permite alimentarea redundantă separată cu tensiune prin modulele sursă, cu funcționare neântreruptă (MSN), precum şi supravegherea continuă a componentelor sistemului relativ la factori ai mediului de exploatare cum ar fi temperatura sau funcționarea subsistemului de aerisire. [AbBr-86].

a_2) Pentru situațiile de cădere a rețelei, se recomandă prevederea unei alimentări redundante, bazată pe baterii, şi deconectarea în trepte a sistemului. Cu toate că, în general, constituie un eveniment rar, căderea rețelei de alimentare poate avea consecințe uneori dezastruoase (constând din pierderea unor date esențiale), în special pentru sisteme care se doresc cu toleranță la defecte, a căror redundantă, destinată acoperirii altor evenimente de defectare, o poate face total ineficientă. Trebuie remarcat că şi sistemele convenționale au, în general, prevăzută o protecție în acest sens, dispunând de scheme de sesizare a scăderii tensiunii continue sub un prag prestabilit, când declanşează (pe durata câtorva milisecunde) rutine de întrerupere corespunzătoare salvării informațiilor de stare şi a datelor de proces pe un mediu de memorare nevolatil, iar la revenirea tensiunii asigură reactualizarea stării sistemului, astfel încât acesta să-şi poată continua funcționarea din momentul întreruperii. Pentru a acoperi situațiile trecătoare de cădere a rețelei, de scurtă durată (spre exemplu de până la un minut), pentru sistemele tolerante sunt prevăzute baterii de alimentare, urmând ca numai în cazul nerevenirii tensiunii de rețea după scurgerea unui interval determinat, tolerat, să se declanşeze decuplarea totală a sistemului de la rețea. Aceasta este uzual prevăzută a se petrece în trepte, în care sens astfel de sisteme sunt prevăzute cu o magistrală adițională, de indicare a stării,cum ar fi MA din fig.2.1, menită a supraveghea corecta funcționare a alimentării, dar şi a altor dispozitive anexe destinate indicării unor anomalii funcționale cum ar fi stări necorespunzătoare de încărcare a bateriilor sau de funcționare a ventilației, precum şi depăşirea valorii limită tolerate a temperaturii mediului de exploatare. Această magistrală redundantă este interogată periodic, la intervale de durată redusă, prin intermediul sistemului de operare, care face posibilă în cazul detecției unei unități tehnologice purtătoare de defect, mai întâi o redistribuire a sarcinii şi,de abia, apoi, în cazul persistenței anomaliei, debranşează total alimentarea cu tensiune.

a_3) O a treia categorie de măsuri, care nu sunt exclusive alimentării, dar o vizează în mod special, constau din separările galvanice (cu precădere a conductorului de masă), precum şi ecranarea împotriva câmpurilor perturbatoare de natură electromagnetică. Separarea galvanică se referă la excluderea unui cuplaj electric, fie el de natură inductivă, capacitivă, electromagnetică sau optică (prin filtre optice), şi intermedierea legăturii prin componente specializate, denumite optocuploare, care asigură evitarea transmiterii şi deci a propagării potențialelor perturbații. Privitor la ecranare, fără a insista asupra importanței acestei măsuri protective, fie amintită doar sursa de câmpuri perturbatoare de natură magnetică pe care o reprezintă transformatoarele de rețea, al căror câmp de

dispersie induce în părţile metalice de impedanţă scăzută ale echipamentelor curenţi de joasă frecvenţă (50 Hz) de valoare importantă. Pentru evitarea acestui fenomen se recomandă învelirea înfăşurării transformatorului într-o foaie de cupru sau aluminiu, reprezentând un ecran electrostatic care formează o spiră în scurtcircuit cu rolul de a diminua fluxul de scăpări.

b) Măsurile de izolare - decuplare vizează, desigur, nu numai căile de alimentare, cu deosebire critice, dar şi cele destinate transmiterii informaţiei utile, aşa cum s-a amintit la prezentarea anterioară a sistemului Parallel / XR. De interes deosebit sunt măsurile referitoare la sesizarea apariţiei malfuncţionării, a semnalizării acesteia, precum şi a creării de bariere pentru a evita, din punct de vedere logic, propagarea efectelor, care ar determina contaminarea prin funcţionare eronată a subunităţilor corecte. Pe de o parte, detecţia şi semnalizarea defectelor poate fi intreprinsă prin transmiterea multicanal a semnalelor şi apelare la formarea de antivalenţe, precum şi o supraveghere funcţională dinamică prin frecvente schimburi de semnale, ceea ce permite, de asemenea, reducerea latenţei defectului. Pe de altă parte, în vederea izolării, cu precădere a elementelor considerate critice, se recomandă apelarea la scheme logice de separare, asigurând căi de impedanţă înaltă sau izolări galvanice pentru reacţii, astfel încât o perturbaţie care se manifestă pe ieşirea unui astfel de element (componentă electronică sau schemă) să nu fie resimţită la intrările sale. Aceste măsuri ridică bariere pentru propagarea erorilor şi, de asemenea, favorizează localizarea defectelor, precum şi remedierea acestora [YaMa-86].

Mai trebuie relevat un aspect, cel legat de faptul că evitarea defectelor se poate materializa doar în parte, astfel că la apariţia unei malfuncţionări se impune asigurarea trecerii sistemului într-o stare sigură sau, altfel spus, se urmăreşte asigurarea unui comportament fail-safe al sistemului. Acest comportament poate fi obţinut,în mod nemijlocit, când însuşi defectul cauzează tranziţia în starea funcţională sigură (măsură ilustrată, în maniera cea mai simplă, prin arderea siguranţei electrice la apariţia unei suprasarcini netolerate, care determină deconectarea alimentării, ceea ce presupune ca această stare să fie una sigură) sau în mod intermediat, când apariţia defectului determină un proces de deconectare sau substituire a subunităţii afectate înainte de a fi atinsă starea sigură a sistemului [IyDH-86].

c) Măsurile de izolare - decuplare s-au referit la nivelul cel mai intim, cel al hardware-ului, unui sistem, dar ele trebuie să vizeze, în mod firesc, şi nivelurile superioare. Ele se bazează pe construcţia modulară a componentelor de proces individuale şi acordă o atenţie specială comunicaţiei dintre procese, context în care a devenit cunoscut principiul de fail - silent. Acesta corespunde situaţiei de anomalie când în cazul defectării unui proces, se invalidează orice comunicaţie a acestuia, el comportându-se ca "adormit" sau "mort". Modelarea unui asemenea comportament se bazează pe două situaţii de defectare frecvent întâlnite pe nivelurile înalte ale sistemului, şi anume cea de oprire (halt), respectiv cea de parcurgere a unei bucle infinite, ambele constituind bariere în propagarea efectelor de malfuncţionare. Mult mai delicate sunt cazurile când defectele constau în derularea unor porţiuni de program necontrolate, cu efecte mult mai imprevizibile decât cele specifice nivelului hardware, care conduc la ieşiri nedorite şi pot, de asemenea, determina supraâncărcarea sistemelor de comunicaţie [IyKo-86,LiCF-89].

d) Un alt grup de măsuri se referă la evitarea defectelor care pot fi declanşate prin exploatare eronată, context în care se tinde spre minimizarea intervenţiei în timpul funcţionării normale. De asemenea, se amintesc declanşarea unei comenzi operatorii la apăsarea a două butoane sau prevederea de comutatoare cu cheie, precum şi validarea unei anumite funcţii operator de abia după verificarea prin simulare sau testare a efectelor execuţiei acesteia. Sunt de amintit, în acest context, analize complexe precursoare stabilirii funcţiilor de operare, care apelează la arbori de defectare şi simulări a cazurilor celor mai defavorabile, urmărind maximizarea fiabilităţii sistemului prin prisma exploatării acestuia [ShLe-86,Gork-91].

B. *Faza de producţie*

În esenţă, măsurile specifice acestei faze se referă la variatele aspecte de testare, începând cu controlul intrare al componentelor electronice (prevăzut, în cele mai multe cazuri, cu condiţii de îmbătrânire accelerată) şi traversând etapele fluxului de fabricaţie cu prevederea de operaţii de verificare intermediare şi finale. În acest context, se impun amintite acele măsuri care se recomandă a fi luate la nivelul schemelor, dar şi la cel al componentelor individuale, menite a creşte testabilitatea, deci a facilita sarcinile de verificare. Acestea îşi dovedesc utilitatea atât la testarea off-line, efectuată în faza de producţie, la punerea în funcţiune, dar şi ulterior, în exploatare, când este necesară înlăturarea prin reparare a defectelor apărute, cât şi la testarea on-line, efectuată în cursul exploatării, prin rutine de verificare declanşate la momente de timp prestabilite sau în anumite situaţii semnalizate de către sistem. Facilităţile de testare includ de la accesibilitatea unor puncte de test până la logica adiţională în calitate de dispozitive destinate localizării defectelor. Legat de testarea efectuată pe parcursul fluxului de producţie, se menţionează importanţa elaborării unei documentaţii corespunzătoare, aspect deseori neglijat, dar care în cazul sistemelor tolerante poate juca un rol decisiv.

Ar mai fi de amintit că anumite situaţii vizând particularităţi constructive, omise în faza de concepţie - proiectare, cum ar fi lungimea unor cabluri, necesită prevederea unor scheme de autotestare suplimentare. Este cazul supravegherii cablurilor lungi la întreruperi şi scurtcircuite (la masă sau la conductoare cu tensiune străină), dar şi a căderilor surselor de energie auxiliară, cum ar fi bateriile [ArKL-901.

C. *Faza de exploatare*

Măsurile specifice fazei de exploatare vizează preponderent izolarea spaţială pentru unităţile sistemului astfel încât să fie evitată perturbarea reciprocă a acestora. Astfel, dispozitive sensibile la factori ai mediului de exploatare, cum ar fi de exemplu unităţile de disc la vibraţii, se recomandă a fi dispuse, în măsura posibilităţilor, în încăperi care să permită protejarea lor prin prisma vulnerabilităţii pe care o prezintă. Un alt exemplu îl constituie dispunerea ecranelor în medii unde sunt dominante câmpuri perturbatoare de natură electromagnetică. În acest sens, confecţionate din materiale cu conductivitate electrică mare (aluminiu sau cupru), se recomandă a fi aşezate într-un plan transversal pe direcţia câmpului magnetic perturbator, mod în care în ecran se induc curenţi electrici care generează câmpul de reacţie protector, opus ca sens celui perturbator. Pentru ca aceşti curenţi să nu determine apariţia unor tensiuni parazite periculoase, se impune ca impedanţa ecranului să prezinte

valori cât mai scăzute, fapt ce implică o corectă conectare la masa de referinţă a sistemului [BaRa-90].

2.2.1. Supravegherea funcţională reciprocă prin dublare statică

Metodele de supraveghere funcţională reciprocă revendică scheme logice adiţionale care implementează toleranţa la defecte prin funcţii de comparare sau / şi decizie majoritară.

În conformitate cu principiul dublării statice, cunoscută şi sub denumirea de activă sau "fierbinte", unei anumite unităţi funcţionale, admitem A, dintr-un sistem convenţional, pentru transformare într-unul tolerant bazat pe dublare i se adaugă o a doua unitate funcţională, admitem B, excitată prin aceleaşi intrări. Ieşirile sunt validate pentru a comanda alte unităţi numai în măsura în care coincid logic, fapt supravegheat de blocul comparator C, conform cu configuraţia la nivel bloc din fig.2.2. În cazul neconcordanţei logice la nivelul ieşirilor unităţilor A şi B se generează şi se semnalizează -punctat în fig.2.2- eroarea, care determină întreruperea funcţionării, făcând acest principiu adecvat pentru acele aplicaţii la care este preferabilă întreruperea funcţionării (eventual deconectarea sistemului) unei continuări riscante, poate chiar periculoase a acesteia.

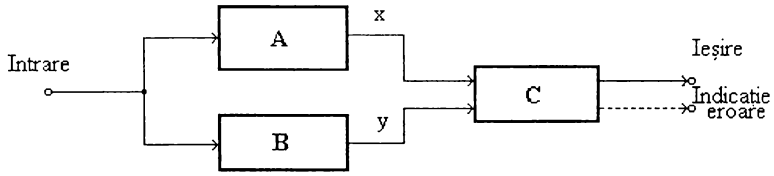


Fig 2.2

Căderea unuia dintre canale este pusă în evidenţă în mod nemijlocit prin funcţia comparatorului, reducând la minim latenţa defectului şi făcând posibilă declanşarea imediată a procedurilor de tratare a erorii. Această din urmă sarcină reprezintă călcâiul lui Achile al principiului dublării, fiind dificilă datorită faptului că prin comparare se semnalizează doar neconcordanţa funcţională a celor două unităţi fără a se da nici o indicaţie asupra faptului care din cele două este defectă. De interes aparte este implementarea funcţiei de comparare, context în care vom considera, pentru simplitate, că cele două unităţi, A şi B, prezintă câte o singură ieşire x, respectiv y (fig.2.2). Făcând uz de circuite logice uzuale, în semantica lor de reprezentare cea mai răspândită, o posibilă realizare a comparatorului C este prezentată în fig. 2.3. În mod evident, ieşirea c a comparatorului, furnizată de poarta three-state marcată cu *, este coincidentă cu x (poate fi, în mod echivalent, şi y) numai când la intrarea de control a respectivei porţi se aplică $v = 1$, unde v constituie funcţia logică de coincidenţă corespunzătoare variabilelor de intrare x si y, adică:

$$v = \overline{x \oplus y} \tag{2.1}$$

în care $\oplus$ semnifică operaţia logică de SAU EXCLUSIV.

În aceeaşi manieră, când $v = 1$, avem pentru ieşirea de eroare $e = 0$, unde e este dată de:

$$e = x \oplus y \qquad (2.2)$$

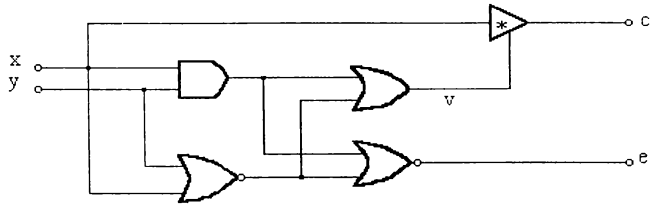


Fig 2.3.

Caracterizată prin simplitatea implementării, metoda este vulnerabilă la defectarea oricăruia dintre circuitele elementare din fig.2.3, un astfel de eveniment fiind netolerabil, compromiţând întreaga configuraţie.

În calitate de exemplu pentru aplicarea supravegherii prin comparare a 2 canale, relevant pentru domeniul microprocesoarelor, să considerăm structura sistemului iAPX 432 al firmei Intel [Gork-91].

Într-o descriere sumară, schema de principiu pe care este fundamentată funcţionarea este prezentată în fig.2.4.,din care reiese că fiecare circuit logic elementar este substituit prin două identice, grupate în unitatea funcţională (UF) şi unitatea destinată verificării (UV).În plus, pe fiecare dintre cele două canale, a şi b, se află conectate circuite de control a ieşirii în magistrală şi circuite de comparaţie C. Funcţional, UF controlează ieşirea prin faptul că circuitele tree-state ataşate sunt deblocate, iar cele corespunzătoare UV sunt blocate.Schema de comparare C asociată UF compară, de fapt, semnale provenite de la aceleaşi ieşiri, fiind practic inactivă.În schimb. cea ataşată UV compară ieşirile celor două unităţi, fiind deci activă şi permiţând semnalizarea oricărei neconcordanţe logice la nivelul unei singure ieşiri prin semnalul f_b=1, care blochează ambele seturi de circuite three-state, izolând deci circuitul sau, altfel exprimat, trecând deci circuitul în stare inactivă (fail-silent).

Conceptul nu este restrâns la implementarea de funcţii logice, ci este aplicat la configurarea întregului sistem microprocesat iAPX 432. Acesta dispune, ca elemente de structură, esenţiale , de circuite integrate procesoare GDP (general data procesor), circuite integrate de interfaţare la magistrală BIU (bus interface unit), precum şi de circuite integrate de control a unităţilor de memorie MCU (memory control unit), care pot fi interconectate în moduri diverse. În fig.2.5 se arată modalitatea în care unităţile procesoare GDP şi unitatea de memorie sunt legate la magistrala de sistem prin intermediul interfeţelor BIU şi a circuitelor de control ale memoriei MCU. La reprezentare s-au utilizat săgeţi duble pentru a marca unitatea funcţională activă şi săgeţi simple pentru a indica unitatea de verificare. Linia verticală unduită marchează interfaţa la circuitele de supraveghere prin comparare. Este de remarcat că la unitatea de memorie nu trebuie prevăzute perechi de celule de stocare a informaţiei, ci, în scopul unei reduceri a investiţiei, controlul este asigurat prin intermediul implementării unui cod Hamming (39, 32, 4).

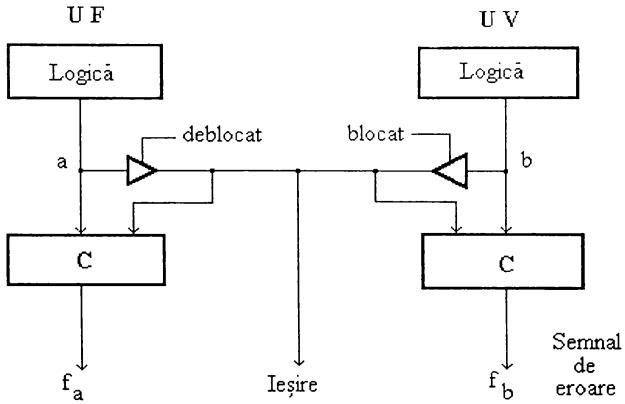


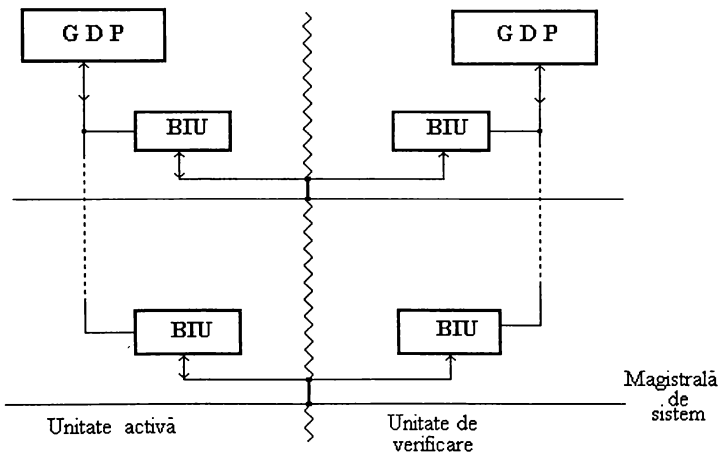
Fig. 2.4

În rest însă, se poate remarca dublarea elementelor de structură cu compararea de rigoare a ieşirilor acestora

Să trecem în revistă avantajele metodei expuse:

- În sistemele configurate cu circuite aparținând familiei iAPX 432 este posibilă introducerea redundanței doar pentru acele elemente de structură cu probabilitate critică la defectare.
- Fiabilitatea sistemului poate fi îmbunătățită fără ca acesta să fie reconceput integral.
- Acoperirea defectelor potențiale este ridicată.
- Nu se majorează numărul circuitelor integrate pe scară largă.

În vederea implementării, hardware-ul sistemului este divizat în blocuri logice, fiecare dintre acestea fiind duplicat şi prevăzut, în cadrul componentei integrate pe scară largă, cu logica de comparare aferentă.



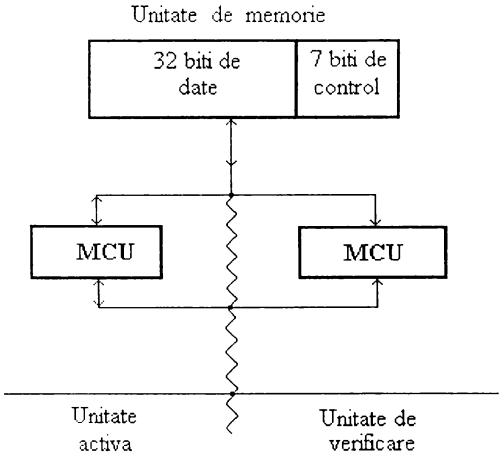


Fig. 2.5

Cu toate că în modul relatat, cu privire la organizarea sistemului şi la disponibilităţile tehnologice de integrare pe scară largă, a fost propus un concept elegant, se impun reliefate şi scăderile acestuia:

- În esenţă, rezultă o investiţie dublată, în timp ce, este posibilă doar detecţia unui defect, fără a se crea nici o facilitate de localizare a acestuia.

- La apariţia unui defect nu mai este posibilă continuarea fiabilă a funcţionării.

- Căderea unui comparator sau a unuia dintre circuitele three-state de control nu este tolerată, putând duce la compromiterea întregului sistem prin prisma fiabilităţii.

2.2.2. Supravegherea funcţională reciprocă prin triplare statică

Prin extensie, dacă la unitatea funcţională A se adaugă încă două, admitem B şi C, cu funcţionare în paralel, atunci se ajunge la triplarea statică (activă sau "fierbinte") cu configuraţia principială din fig.2.6. Considerând din nou pentru simplitate, că cele trei unităţi funcţionale prezintă câte o singură ieşire c (x,y,respectiv z în fig.2.6), în urma traversării schemei de comparare logică - într-o posibilă implementare dată în fig.2.7-, are loc atunci când când v'=1, unde v' este dată de expresia logică:

$$v'=xyz+\overline{xyz} \qquad (2.3)$$

Pe de altă parte, eroarea e' este generată de funcţia logică care urmăreşte punerea în relief, prin $e' = 1$, a oricărei neconcordanţe între cele 3 variabile de intrare, fiind dată de expresia:

$$e' = x \oplus y + y \oplus z + z \oplus x \qquad (2.4)$$

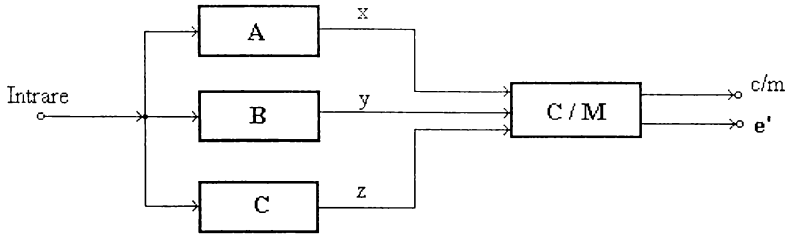


Fig. 2.6.

În cazul triplării statice este însă posibilă implementarea supravegherii reciproce prin formarea funcției logice majoritare, comparatorul C fiind substituit prin schema logică M (fig.2.6), care furnizează ieşirea *m*, dar poate furniza şi pe *e'*, când avem de-a face practic cu combinarea conceptelor de comparare şi decizie după majoritate.

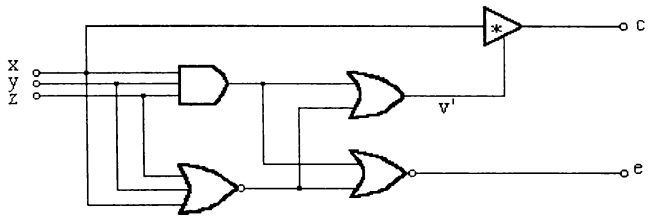


Fig. 2.7.

O posibilă implementare a schemei majoritare M, prevăzută cu indicarea unui canal defect, este dată în fig.2.8, în care funcția logică *m* prezintă expresia:

$$m = xy + yz + zx \qquad (2.5)$$

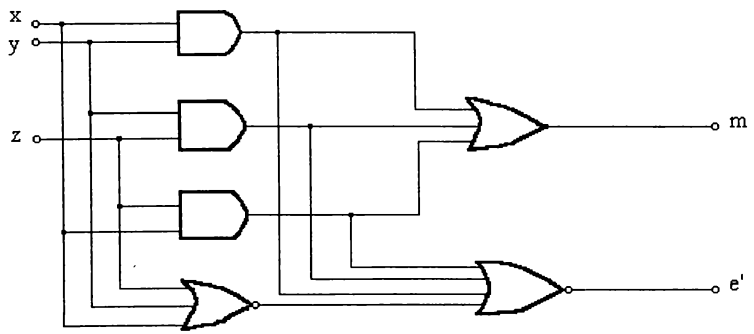


Fig. 2.6.

Schema logică din figura 2.8 realizează votul după principiul majorității, în sensul că valoarea logică la ieşirea *m* este coincidentă cu cea a majorității - în acest caz, 2 din 3 sau 3 din 3 - intrărilor sale. Cu alte cuvinte, este tolerată defectarea unuia, dar numai a unuia, dintre cele trei canale sau, altfel exprimat, este mascată defectarea unui canal.

Principiul a fost împrumutat din mecanică, unde, spre exemplu, s-a încercat prin 3 sau 4 motoare de zbor să fie tolerată căderea unuia dintre ele, dar raportat la acest caz, când în situația funcționării corecte a tuturor motoarelor are loc o distribuire a sarcinii totale, în domeniul calculului nu este uzuală o astfel de fructificare a redundanței.

Chiar dacă sistemul îşi continuă funcționarea corectă în cazul existenței unui canal defect, este recomandată identificarea acestuia şi apoi înlocuirea lui. Prin aceasta este evitată posibilitatea ca o cădere a unui al doilea canal să compromită în mod decisiv ieşirea. Prin urmare, schema majoritară este de dorit să fie înzestrată cu capacitatea de diagnoză a canalului defect, care poate fi implementată în mod facil în sensul că, atunci când prin $e'=1$ este semnalată prezența unui defect, ieşirea fiecărui canal să fie comparată cu ieşirea *m* a schemei majoritare, ceea ce se poate realiza printr-o simplă extensie a schemei din fig. 2.8.

Ca şi în cazul dublării, nu este tolerată defectarea nici unuia dintre circuitele utilizate la sinteza schemei majoritare, sub care aspect aceasta constituie un element vulnerabil al metodei. Triplând însă şi schema de votare cu multiplexarea intrărilor ca în fig. 2.9 se ajunge la o configurație redundantă care poate tolera inclusiv defectarea uneia dintre cele 3 scheme majoritare.

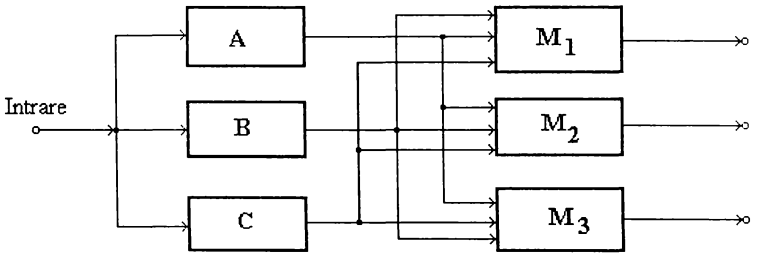


Fig.2.9.

Extinzând discuția, dacă probabilitatea de defectare quasiconcomitentă a 2 canale este semnificativă, atunci configurația 2 din 3 este compromisă şi trebuie apelat, prin corespunzătoarea creştere a investiției, la o configurație 3 din 5 şi, prin generalizare, la una *n* din *m* atunci când este posibilă defectarea concomitentă a $(m - n)$ canale. În acest context, se relevă că poate fi demonstrat fără efort [Gork - 89] că cea mai favorabilă configurație prin prisma costului este cea care prezintă $n = \lceil (m+1) / 2 \rceil$, unde barele $\lceil\ \rceil$ semnifică cel mai mic întreg mai mare decât rezultatul expresiei dintre bare. Menționăm, de asemenea, că principiul poate fi aplicat prin conectarea în cascadă a schemelor majoritare şi că, în general, alegerea configurației redundante se impune precedată de minuțioase analize, în care trebuie urmărită soluționarea optimă a impactului câştig de fiabilitate şi disponibilitate

/ creştere investiţie. Astfel, în multe situaţii, cerinţele de siguranţă în funcţionare şi disponibilitate sunt satisfăcute în mod mulţumitor prin dublarea 2 din 2. Dacă însă aplicaţia revendică indicatori de fiabilitate superiori celor oferiţi de dublare, o configuraţie recomandabilă este 2 din 3, cu decizie majoritară şi comparare, la defectarea unuia dintre canale fiind posibilă comutarea de la configuraţia triplată 2 din 3 la cea bazată pe dublarea 2 din 2 [Anaw-89,TaTr-90,Parh-91].

În continuare, vom extinde aplicarea triplării statice de la un canal util la cuvinte formate din câte n canale, într-o primă instanţă, independente, fiecare dintre acestea fiind substituit prin 3 canale identice. Cum pentru fiecare grupare triplată este acceptat un canal defect, sunt tolerate în total n canale defecte. Situaţia este ilustrată în fig.2.10a, când cuvântul corect este afectat de 2 erori în poziţii diferite (a 5-a, respectiv a 6-a), iar drept efect al votării pe principiul majorităţii pentru fiecare dintre cele, în cazul adoptat, 8 canale, rezultatul apare corectat. Nu acelaşi lucru se întâmplă dacă cele 2 erori afectează unul şi acelaşi canal (al 5-lea în fig.2.10b), când în urma votării rezultatul este compromis, diferind de configuraţia binară corectă.

```
10111000  corect
10110000  eroarea bitului din poziţia a 5-a
10111100  eroarea bitului din poziţia a 6-a
--------------------------------------------------
10111000  rezultat coincident cu cuvântul corect
```

Fig. 2.10a

```
10111000  corect
10110000  eroare a bitului din poziţia a 5-a
10110000  eroare a bitului din poziţia a 5-a
--------------------------------------------------
10110000  rezultat eronat
```

Fig. 2.10b

Dacă însă datele reprezintă numere sau coduri de semne, dependente deci de poziţia binară a canalului pe care se manifestă eroarea, aplicarea triplării 2 din 3 trebuie abordată cu precauţie. Să admitem că în urma efectuării unei operaţii aritmetice rezultatele oferite de cele 3 canale diferă valoric cu ± 1, ca urmare a aplicării a variate procedee de rotunjire. Admiţând, spre exemplu, numărul binar întreg şi fără semn 23, reprezentat pe 8 biţi, cu deviaţiile de ± 1 (adică 24, respectiv 22), în fig.2.11a se poate observa că, în urma activării corespunzător fiecărei poziţii binare a schemei majoritare 2 din 3, rezultă numărul 22. Tot astfel, dacă numărul 24 este cel corect şi deviaţiile valorice de ± 1 sunt respectiv numerele 25 si 23, prin votare rezultă numărul 25 (fig.2.11b). Ambele situaţii descrise, care acoperă posibilităţile în domeniul tipului de numere analizate, trebuie tolerate,

chiar dacă rezultatul nu coincide cu cel corect, ci cu una dintre abaterile -1 pentru cazul din fig. 3 8a, respectiv +1 pentru cazul din fig.2.11b.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	0	0	1	0	1	1	1	= 23 (corect)
0	0	0	1	1	0	0	0	= 24 (+1)
0	0	0	1	0	1	1	0	= 22 (-1)
0	0	0	1	0	1	1	0	= 22 (-1)

Fig.2.11a

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	0	0	1	1	0	0	0	= 24 (corect)
0	0	0	1	1	0	0	1	= 25 (+1)
0	0	0	1	0	1	1	1	= 23 (-1)
0	0	0	1	1	0	0	1	= 25 (+1)

Fig. 2.11b

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	0	0	1	1	0	0	0	= 24 (corect)
0	0	0	1	0	0	0	1	= 17 (eroare +1)
0	0	0	1	0	1	1	1	= 23 (-1)
0	0	0	1	0	0	0	1	= 17 (eroare)

Fig. 2.11c

Dacă însă peste această abatere cauzată de procedură, şi în consecinţă tolerată, se suprapune chiar şi numai o singură eroare, admitem în poziţia binară cu ponderea 2^3 pentru numărul 25 (deci deviaţia cu +1), el devenind numărul 17, în fig.2.11c se observă că schemele majoritare de pe cele 8 canale furnizează rezultatul eronat 17, constituind o situaţie intolerabilă. Dacă însă această eroare singulară ar afecta poziţia binară cu ponderea 2^4, transformând tolerata deviere constituită de numărul 25 în numărul 9, rezultatul obţinut după votare ar duce la acelaşi număr 25 ca şi în fig.2.11b, deci la o abatere tolerată. Sumarele analize întreprinse relevă evidenţa faptului că, în cazul datelor binare ponderate, implementarea hardware a modulului de formare a majorităţii prezentate devine dificilă, fiind preferabilă implementarea votului prin metode software [Gork - 89].

2.3. Strategii conceptuale pentru programele de autotestare

2.3.1. Considerente asupra verificării prin programe

În vederea debutului unui proces de verificare prin programe este necesar ca o parte dintre unitățile tehnologice ale unui sistem de calcul să fie în stare de funcționare. Aceste subsisteme, incluzând sursele de alimentare, generatoarele de clock şi unele dintre funcțiile unității centrale de procesare (Central Processing Unit-CPU), se atribuie unui nucleu hardware (hardware kernel), a cărui funcționalitte este indispensabilă la momentul inițial. Se impune o primă precizare legată de funcțiile CPU care se atribuie nucleului hardware inițial, sub care aspect, pentru claritate, dar fără a pierde din generalitate, vom limita considerațiile la configurații a căror componente de structură sunt relevate în fig.2.12.[PaHe-89].

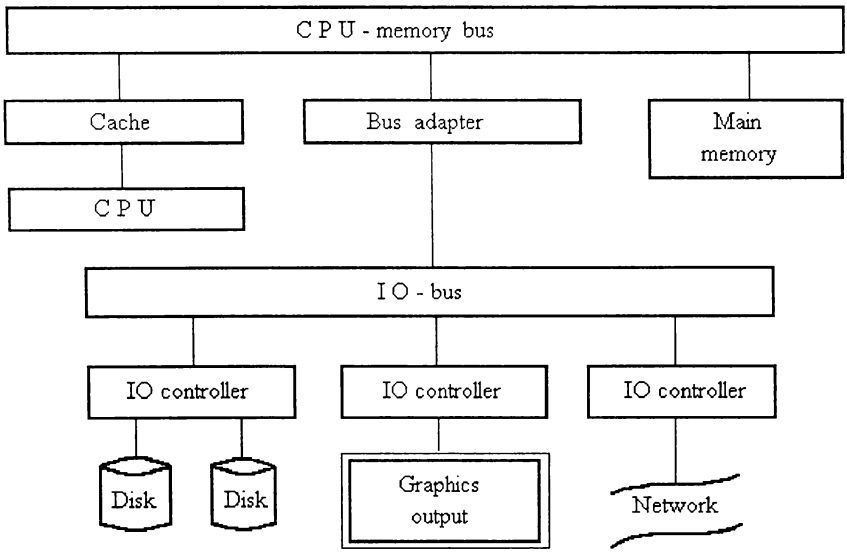


Fig.2.12

După cum rezultă din ilustrație, avem de a face cu un sistem prevăzut cu magistrală dual㠁, una destinată traficului informațional CPU-memorie (CPU-memory bus), admisă drept standard (de tip asincron cum ar fi standard IEEE 1014 şi Future Bus- standard IEEE 896.1 sau, de tip sincron cum ar fi Multibus II - satndard ANSI/IEEE 1296), şi cealaltă destinată traficului de intrare-ieşire (IO-bus), care este de lungime mai mare (25m sau 50m față de 0,5m) şi cu acomodare la dispozitive periferice de viteze foarte diferite, care poate fi şi ea standardizată (de tip asincron cum ar fi IPI sau atât sincronă cât şi asincronă cum ar fi SCSI).

Adaptarea la nivel de semnale între cele două magistrale este asigurată de către blocul Bus adapter, iar conectarea diferitelor periferice, dintre care figura pune în relief unități de disc magnetic (Disk) şi ieşire grafică (Graphics output), la care se adaugă capacitatea de conectare la rețea (Network), este realizată prin procesoare de intrare-ieşire (IOP controller) dedicate. În ceea ce priveşte conectarea adaptorului de magistrală (Bus adapter), în cazul când sistemul este prevăzut cu facilități hardware-software de memorie virtuală (cum ar fi o memorie cache suplimentară ataşată CPU-ului cu rol de buffer "lateral" destinat translatării adreselor logice în unele fizice - translation lookaside buffer-TLB), se dovedeşte mai favorabilă legarea lui Bus adapter direct la memoria cache, aceasta în vederea eficientizării regăsirii în baza principiului de localizare a referinței (locality of reference), mai frecvente a adresei în memoria cache. Evident, caracteristici de structură de acest gen nu prezintă un interes deosebit interes în contextul actualei prezentări, cu toate că ele pot contribui la o eficientizare a adresării în general, deci şi la îmbunătățirea capacității de trecere a programelor de autotestare în particular [PaHe-89,Haye-88]. De asemenea, în contextul prezentei lucrări nu se insistă asupra facilităților de acces direct la memorie, (Direct Memory Access -DMA) sau întrerupere (Interrupts), care pot şi ele contribui în mod consistent la creşterea capacității de trecere a programelor, dar care în actuala expunere sunt referite doar în sensul în care sistemul posedă capacitatea de invalidare a acestora, importantă după cum va rezulta într-o anumită etapă a procesului de de verificare. Fig. 2.12 mai cuprinde o memorie principală (Main memory), care este cu acces aleator (random access memory -RAM) fiind, în ceea ce priveşte alterabilitatea, în esență de citire-scriere (read - write), dar putând acoperi şi o parte destinată doar citirii (read only memory-ROM), şi, pe de altă parte, o memorie cache, care este admisă de uz general (general purpose).

Referindu-ne la blocul CPU din fig.2.12 considerăm adecvată, prin prisma ulterioarelor referiri, detalierea într-o oarecare măsură a structurii acestuia, fapt pentru care aderăm la cele ilustrate în fig.2.13 [Haye-88]. Caracteristicile de structură ale organizării unei CPU tipice le subliniem în următoarea enumerare:

1. Singularul registru acumulator din clasica configurație propusă de von Neumann pentru maşina IAS [Haye-88], calculator de referință pentru structuri CPU orientate pe acumulator, este replasat printr-o organizare pe registre generale (general register organisation), denumite şi pilă de registre (register file) sau memorie scratchpad, reprezentând un set de registre adresabile, destinate stocării operanzilor şi adreselor, şi care pot fi regăsite atât în calculatoare de capacitate mare şi medie (mainframes), cum ar fi IBM S/860-370, dar şi în structurile interne ale capsulelor microprocesoare cum ar fi Motorola 68020.

2. Unitatea aritmetică şi logică cuprinde circuistica necesară implementării cablate - pe lângă cea corespunzătoare convenționalelor funcții logice de negare, ŞI (AND), SAU (OR) şi SAU EXCLUSIV (EX-OR), efectuate bit cu bit -şi a mai complexelor operații aritmetice de adunare, scădere, dar şi înmulțire şi împărțire, efectuate cu numere reprezentate în formate de virgulă fixă.

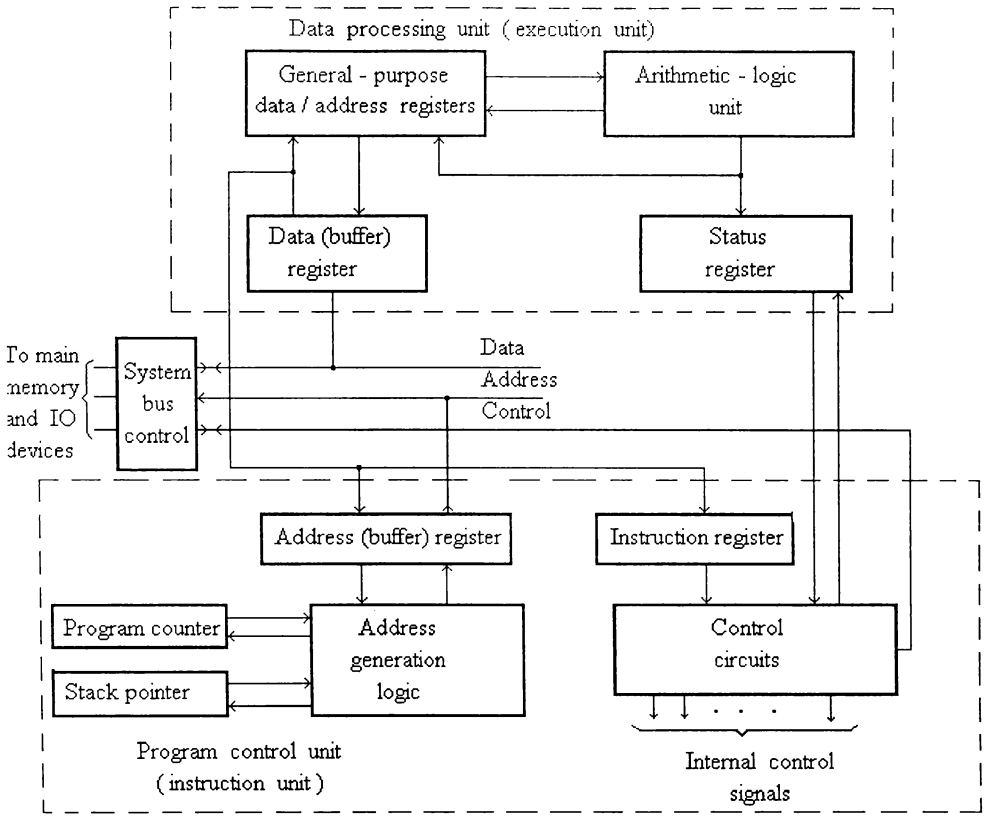


Fig.2.13

Acestea sunt implementate în capsule microprocesoare, cum ar fi Motorola 68020, dacă avem de-a face cu realizări microprocesate. În ceea ce priveşte mai complicatele operaţii de adunare, scădere, înmulţire şi împărţire efectuate în virgulă flotantă cu operanzi de lungime variabilă, precum şi a operaţiilor de exponenţiere, logaritmare, funcţii trigonometrice, ş.a, ele sunt de asemenea cablate în unităţile aritmetice convenţionale ale calculatoarelor de capacitate mare şi medie (mainframes), iar relativ la variantele microprocesate s-a impus implementarea prin procesoare aritmetice fie după mai puţin răspânditul concept al procesorului periferic (peripheral processor), cum ar fi, spre exemplu, AMD 9511/12, fie, după larg acceptatul concept de coprocesor (coprocessor), acesta din urmă reprezentând o extensie logică la CPU, cum ar fi spre exemplu, Motorola 68881 pentru seriile Motorola 68000.

3. Unitatea de procesare a datelor (Data processing unit), numită şi unitate de execuţie (execution unit) mai cuprinde un registru de stare (status register), denumit de asemenea, cod de condiţie (condition code) sau registru de fanioane (flag register), care este destinat facilitării transferului controlului dintre instrucţii în cadrul unui program. Spre exemplu, el este utilizat pentru

indicarea unor condiţii rezultate la execuţia instrucţiei anterioare, cum ar fi semnul (pozitiv sau negativ) al unui rezultat numeric, apariţia unui rezultat constând din 0-uri, sau a unei terminări anormale cauzate de o depăşire (overflow) numerică. Registrul de stare poate fi testat de către o instrucţie de salt condiţionat pentru a altera secvenţa de execuţie a instrucţiilor bazat pe un rezultat al anterioarei instrucţii.

4. Transferul controlului între diferite subprograme-subrutine (subrutines) sau proceduri (procedures) - determinată de întreruperi sau cereri şi reveniri de subrutine este facilitată de alte registre speciale. Spre exemplu, calculatoarele S/360-370 utilizează registrul PSW (program status word), a cărui conţinut include numărătorul de program (program counter) şi fanioanele (flags) pentru a înregistra starea de execuţie a unui program. Controlul este transferat prin salvarea PSW-ului curent într-o locaţie prestabilită din memoria principală (main memory) şi prin încărcarea unui nou PSW în CPU. Controlul este returnat primului program prin regăsirea anterior salvatului PSWdin memorie şi readucerea lui în CPU. Majoritatea calculatoarelor utilizează în prezent o mai flexibilă schemă de control a transferului dintre programe, care se bazează pe folosirea unei părţi a memoriei principale ca o stivă (pushdown stack). Stiva este utilizată pentru a stoca informaţia de stare a programului şi are o disciplină de acces LIFO (last in - first out, adică ultimul- intrat primul - la ieşire). Ea este considerată de către un registru de adresare al memoriei din cadrul CPU, denumit indicator de stivă (stack pointer SP),care indică în permanenţă "vârful" ("top") curent sau adresa de intrare în stivă. Un cuvânt este transferat la sau din stivă prin efectuarea unui acces de memorie utilizând SP-ul ca registru de adresare, iar conţinutul lui SP este apoi ajustat în mod automat pentru a face referire la noul vârf al stivei. Un avantaj important al utilizării stivei constă în abilitatea de tratare eficientă a transferurilor repetate (nested) dintre programe.

5. În ceea ce priveşte blocul circuitelor de control (control circuits) se impune o importantă distincţie, prin prisma inclusiv a strategiilor de implementare a autotestării între varianta în logică cablată (hardwired), sugerată în fig.2.14, şi cea microprogramată (microprogrammed), sugerată în fig.2.15, de realizare.

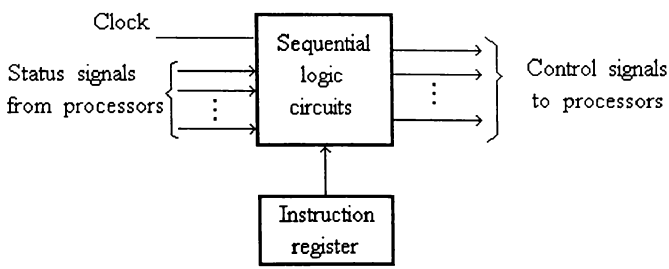


Fig.2.14

O unitate de control în logică cablată (hardwired control unit), aşa cum rezultă din fig.2.14, constituie, în mod tipic, un automat secvenţial, a cărui sinteză poate fi întreprinsă printr-una din cele trei metode [Haye-88]:

a)metoda tabelei de stări (state-table method)

b)metoda elementelor de întârziere (delay-element method)

c)metoda numărătorului de secvență (sequence-counter method)

Orice instrucție din CPU este implementată printr-o secvență de unua sau mai multe așa numite microoperații (microoperations), fiecare microoperație având asociat un set specific de linii de control care, atunci când sunt activate determină ca microoperația să se execute. Indiferent de metoda uzitată la sinteză, întrucât numărul instrucțiilor și al liniilor de control este deseori de domeniul sutelor, o unitate de control în logică cablată, care selectează și înlănțuie semnale de control, poate fi excesiv de complicată. Drept consecință, ea este dificil și costisitor de proiectat, și , în plus, ea este, în mod evident, inflexibilă la modificări (cum ar fi corectarea de erori de proiectare sau modificarea setului de instrucții), acestea necesitând reproiectarea întregii unități de control.

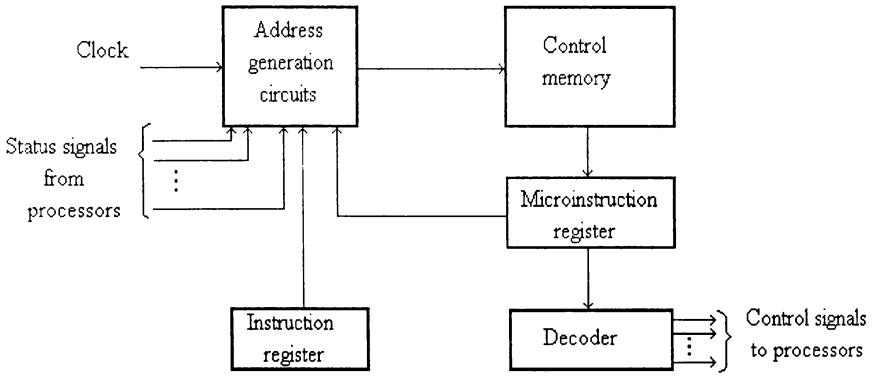


Fig.2.15

O alternativă mai atrăgătoare și mai flexibilă la logica cablată o reprezintă tehnica sistematică a microprogramării propusă de M.V.Wilkes, care conduce la o structură generală a unității de control cu componentele ilustrate în fig.2.15. Fiecare instrucție a procesorului determină execuția unei secvențe de microinstrucții, denumită microprogram (microprogram), care este adusă (fetched) dintr-o memorie ROM sau RAM, denumită memorie de control (control memory). Microinstrucțiile specifică secvența de microoperații sau operații de transfer dintre registre necesare execuției interpretării și execuției instrucției principale. Astfel, fiecare instrucție adusă din memoria principală inițiază o secvență de microinstrucții aduse din memoria de control. Setul de instrucții al unei mașini microprogramate poate fi modificat prin replasarea conținutului memoriei de control, conferind microprogramării atributul de flexibilitate. În plus,aceasta face posibil ca un calculator microprogramat să execute în mod direct programe scrise în limbajul de mașină a unui alt calculator, proces denumit emulare (emulation). Unitățile de control microprogramate tind să fie mai costisitoare și mai lente decât unitățile în logică cablată, dar aceste scăderi sunt, în general, balansate de către flexibilitatea asigurată de microprogramare, precum și de progresele realizate în tehnologia

circuitelor integrate. Datorită interacțiunii strânse dintre hardware și software din sistemele microprogramate, microprogramele sunt uneori denumite firmware.

6. O structură actuală de CPU mai cuprinde, ceea ce în fig.2.14 nu este evidențiat în mod distinct, facilități de procesare simultană a două sau mai multe instrucții. Ne referim în acest context la metode de accelerare de tip pipeline, care pot viza atât execuția suprapusă (overllapped) a instrucțiilor -cum ar fi de exemplu, suprapunerea ciclului de aducere (fetch cycle) a instrucției următoare cu ciclul de execuție (execution cycle) a instrucției curente-, dar și execuția suprapusă a operațiilor aritmetice, cum ar fi, spre exemplu, o înmulțire cu salvarea transportului (carry-save multiplication).

Cu aceste precizări,de construcție și limbaj necesare în contextul dezvoltării problematicii de testare prin programe, să fixăm care anume dintre caracteristicile de structură anterior expuse sunt atribuite unei capsule de microprocesor, adică unei unități tehnologice la care,odată localizată o anumită malfuncționare nu-și au rostul ulterioare investigații. În acest sens,ne alegem drept reper stadiul tehnologic reprezentat de capsule microprocesoare one-chip Motorola 68020, a cărei structură internă la nivelul controlului, corespunde unui calculator de uz general (general purpose computer) cu un sistem de operare multiuser și este ilustrată în fig.2.16.

Din punct de vedere fizic, Motorola 68020 constă dintr-un circuit integrat realizat în tehnologie CMOS de înaltă densitate, compus din aproximativ 200.000 de tranzistoare și închis într-o capsulă PGA (pin-grid array) cu 114 pini. Organizarea funcțională a lui 68020 corespunde unui calculator mainframe de generația a treia. Unitatea de procesare a datelor sau de execuție conține 16 registre de uz general pe 32 de biți pentru date (D0 la D7) și pentru adrese (A0laA7), precum și circuitele aritmetice necesare execuției unui set complet de instrucții în virgulă fixă (nu însă și în virgulă mobilă). Interpretarea instrucțiilor și alte funcții de control sunt implementate printr-o unitate de control microprogramată. Așa cum indică lungimea registrelor de date, 68020 este prevăzut pentru a manipula cuvinte pe 32 de biți (denumite cuvinte "lungi" în literatura seriilor Motorola 68000) în mod eficient, dar instrucțiile sunt, de asemenea, prevăzute a manipula operanzi pe 1,8,16 și 64 biți. Adresele de memorie au lungimea de 32 biți, permițând a fi utilizat un total de 2^{32} adrese diferite. Ca și la alte calculatoare moderne, cum ar fi System/360, memoria principală este organizată ca o matrice de bytes adresabili în mod individual. Astfel, capacitatea maximă de adresare a unui calculator bazat pe seriile Motorola68000 este de 2^{32} bytes, adică de 4 gigabytes (4 Gbytes). Ca și precedesorul lor pe 8 biți, Motorola 6800, procesoarele din seriile 68000 împart spațiul de adresare disponibil între dispozitive de memorie principală (ROM-uri și RAM-uri) și dispozitive IO (de intrare-ieșire). În consecință, aceleași instrucții de transfer a datelor (MOVE) utilizate pentru transferuri CPU-memorie sunt, de asemenea folosite pentru operații IO. Această tehnică de proiectare care elimină necesitatea unei clase speciale de instrucții IO este denumită memory-mapped IO și ea contrastează cu mai răspândita tehnică IO-mapped IO întâlnită la mașini ca S/360-370 și Intel 8085.

Microprocesorul 68020 are în jur de 70 de tipuri de instrucţii [Haye-88], extinzând repertoriul originalului circuit al seriei 68000, care sunt prevăzute cu variate moduri de adresare conferind programatorului flexibilitate considerabilă pentru accesarea datelor.

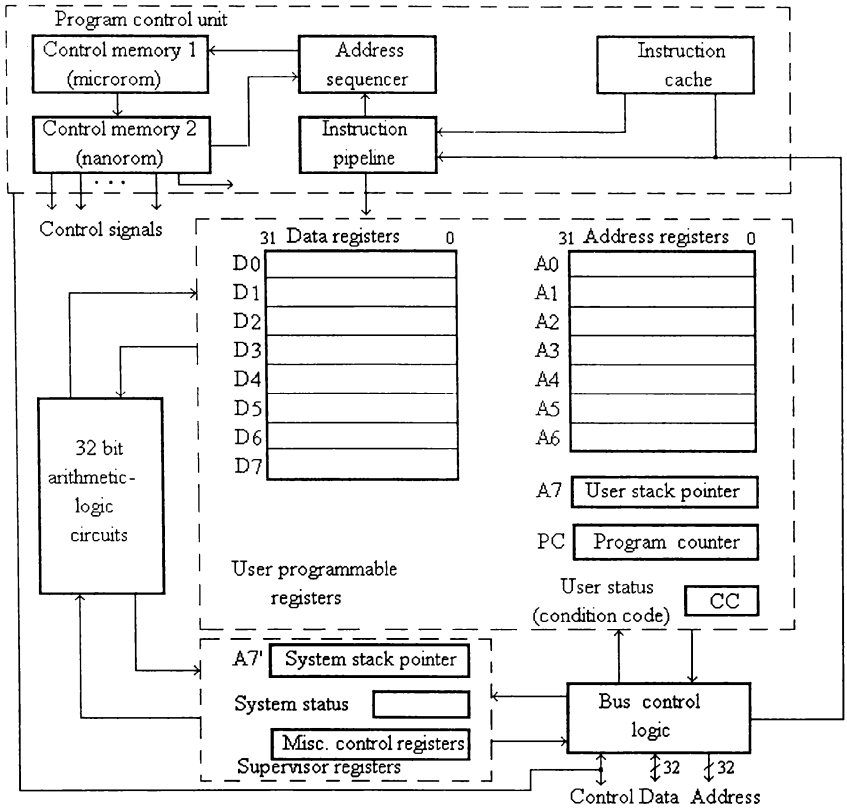


Fig.2.16.

Pe lângă convenţionalele moduri de adresare directă, indirectă şi imediată, microprocesorul 68020 este prevăzut cu adresarea autoindexată, mod prin care se elimină necesitatea unor registre index separate, asigurându-se incrementarea şi determinarea în mod automat a adresei. Autoindexarea, care facilitează în mod consistent adresarea stivei (stack addressing), se poate realiza cu predecrementare (predecrementing) -când câmpul de adresă specificat prin - (A3) într-o instrucţie de limbaj de asamblare semnifică faptul că în mod automat, conţinutul registrului de adresare prevăzut, A3, trebuie decrementat înainte ca instrucţia să fie executată - sau cu postincrementare (postincrementing) - fapt specificat în forma (A3) -, semnificând că, în mod automat, conţinutul lui A3, trebuie incrementat după ce instrucţia curentă a fost executată. În fiecare caz, cantitatea incrementului sau decrementului adresei este dat de lungimea în bytes a operanzilor adresaţi. Admiţând, în calitate de exemplu, că registrul de adrese A3 a fost destinat de către programator ca indicator de stivă şi că stiva creşte înspre adresele mai puţin semnificative ale memoriei, pentru a

introduce (push) în stivă conținutul unui registru de date, să spunem D6,este necesară instrucția MOVE.L D6,-(A3). Codul operației MOVE al acestei instrucții de transfer este prevăzut cu modificatorul . L indicând că datele reprezintă un cuvânt "long" (4-byte). Operandul de intrare este reprezentat de conținutul lui D6, care în anterior specificata instrucție, este adresat în mod direct, în timp ce operandul de ieşire, care este noul conținut al vârfului stivei, este desemnat prin adresare indirectă cu predecrementare. Descrise formal,operațiile, cu care este echivalentă instrucția de push MOVE.L D6,-(A3), se prezintă astfel: A3 ←A3 - 4; M(A3) ← D6; unde M(A3) reprezintă adresa de memorie principală M specificată prin conținutul registrului de adresare A3. Corespondența extragere din stivă se realizează print-o instrucție de pop care face uz de postincrementare, de tipul MOVE.L (A3)+,D6, fiind echivalentă cu operațiile : D6 ←M(A3), A3←A3+4.

Detaliind într-o măsură oarecare controlul stivei, CPU-urile din seria Motorola 68000, incluzând şi microprocesorul 68020, au având instrucții convenționale de apelare (call) şi revenire (return), având numele mnemonice JMS (jump to subroutin şi RTS (return to subroutin). Când sunt executate de către programe ordinare de utilizator, JMS şi RTS fac uz de indicatorul de stivă predestinat A7, care controlează o stivă din memoria principală. Dacă CPU-ul este în stare supervizor, stare revendicată de către sistemul de operare pentru a asigura controlul separat de către acesta al stivei față de programele utilizator, aceasta se indică prin starea distinctă a unui bit fanion (flag bit) din registrul de stare SR şi prin utilizarea unui al doilea indicator de stivă A7' (fig.2.16). Semnalele de întrerupere către CPU, indicând o condiție excepțională care necesită a fi luată în considerare de către sistemul de operare, sunt tratate prin comutarea în stare supervizor şi prin salvarea conținuturilor originale ale numărătorului de program PC şi ale registrului de stare SR în stiva controlată prin A7'. Apoi, CPU-ul crează un nou cuvânt de stare indicând că este în stare supervizor şi încarcă PC-ul cu o adresă X, constituind adresa de început a programului de tratare a întreruperii, care poate fi furnizată fie de către sursa cererii de întrerupere (interrupt request), fie poate fi predeterminată (cablată în CPU) pentru o întrerupere particulară. Returnarea controlului la programul original întrerupt poate fi executată de către programul de deservire a întreruperii prin instrucția de revenire specială RTE (return from exception), care extrage din stivă (pops), prin intermediul lui A7', anterior salvatele cuvânt de stare - pe care îl plasează în SR - şi adresă de revenire pe care o plasează în PC. Astfel, RTE este , din punct de vedere funcțional, echivalentă cu secvența de două instrucții:

```
MOVE  (A7) +, SR                                    ( 2.6)
RTS
```

Stivele sunt foarte convenabile atât pentru stocarea parametrilor ce urmează a fi transferați între programe, cât şi în calitate de zone de memorie de lucru (working storage) pentru variabile locale pe durata unui calcul. În acest sens, este favorabil pentru utilizator de a avea posibilitatea alocării şi dizlocării de blocuri de memorie în şi din vârful stivei când apare această necesitate. Familia 68000 are prevăzute în acest scop două instrucții de control a stivei, şi anume LINK (link) şi

UNLK (unlink). Aceste instrucţii comandă un registru de adresare (unul dintre registrele A0 la A6 - fig.2.16), denumit indicator de cadru (frame pointer) FP, care, împreună cu indicatorul de stivă SP = A7 sau A7', delimitează aria de lucru, aşa numită cadru (frame), în stivă. În limbaj de asamblare, instrucţia de legătură (link) are formatul:

$$\text{LINK } A_i, \#\text{ - k} \qquad (2.7)$$

în care registrul A_i specificat constituie indicatorul de cadru FP aliniat. Efectul instrucţiei este ilustrat în fig. 2.17, imediat înainte (fig.2.17a) şi imediat după (fig.2.17b) execuţia acesteia. Se remarcă faptul că execuţia instrucţiei cauzează, mai întâi, introducerea în stivă a vechiului conţinut de 4 cuvinte (FP_1) a indicatorului de cadre FP, şi, apoi, transferarea în FP a noului conţinut (SP_1 - 4) a indicatorului de stivă SP. În acest mod, FP permite referirea la partea inferioară (bottom) a ariei de memorie alocate cadrului. Din conţinutul lui SP este scăzut numărul k(specificat prin prefixul # semnificând, în convenţia Motorola, modul de adresare imediată), astfel încât vârful stivei este avansat (înspre valorile mai puţin semnificative de adresă) cu k bytes în raport cu locaţia adresată prin FP. Cadrul de k bytes delimitat prin FP şi SP este utilizat, în mod tipic, ca o arie de lucru pentru o subrutină. Variabilele locale ale subrutinei pot fi adresate relativ la FP, în timp ce vârful stivei poate fi deplasat în sus şi în jos independent de FP. Spaţiul alocat prin (2.7) poate fi apoi dizlocat prin execuţia instrucţiei:

$$\text{UNLK } A_i \qquad (2.8)$$

care mută conţinutul din FP = A_i în indicatorul de stivă SP şi apoi descarcă stiva în A_i restaurând astfel starea originală din fig.2.17a. Utilizarea instrucţiilor LINK şi UNLK în maniera descrisă permite, în mod simplu apelarea repetată a unei stive comune de către aceeaşi subrutină sau de către subrutine diferite, fără să apară interferenţe de cadre asociate cu diferitele apeluri.

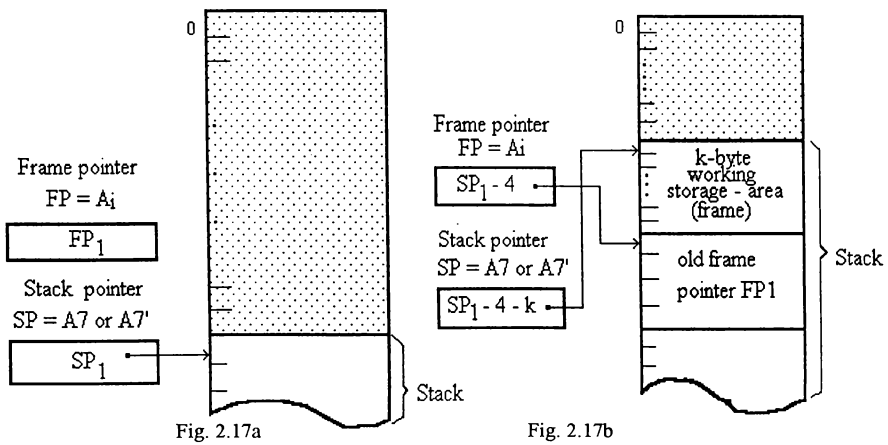


Fig.2.17.

Repertoriul de instrucţii asociat microprocesorului 68020 se caracterizează printr-o bună ortogonalitate, deziderat vizând abilitatea utilizării tuturor codurilor de operaţii ale unui set de intrucţii sau limbaj de programare într-un mod uniform şi consistent cu toate modurile de adresare relevante şi care are, drept consecinţă, simplificarea programării atât prin reducerea numărului de coduri de operaţie distincte, cât şi prin simplificarea regulilor de specifcare a adreselor de operand. Prin cele două stări de control diferite - starea supervizor, prevăzută pentru sistemul de operare, şi starea utilizator, prevăzută pentru programe de aplicaţie -, rezultând o sseparaţie clară între cele două categorii de programe - în care scop sunt prevăzute facilităţi hardware cum ar fi anterior amintite indicatoare de stivă distincte - se asigură o îmbunătăţire a securităţii sistemului. De asemenea, calculatoarele bazate pe 68020 sunt astfel proiectate pentru a permite o implementare facilă a conceptului de memorie virtuală. Multe dintre funcţiile de translare a adresei şi de control a cererilor de pagini solicitate în acest sens sunt implementate de către unitatea de management a memoriei (memory management unit, MMU) 68851, o altă capsulă coprocesor. În timp ce un sistem tipic bazat pe 68020 are o memorie fizică mult mai mică decât cei 2^{32} bytes capabili de a fi fizic adresaţi, prin apelarea la facilitatea de memorie virtuală oferită de coprocesorul 68851 întreaga memorie de 4G-byte poate fi făcut să apară disponibilă pentru fiecare utilizator într-un mediu de programare multiuser. Adăugând facilităţilor oferite de către coprocesorul 68851, cele pe care le asigură coprocesorul aritmetic 68881, prinde contur potenţa de calcul a sistemelor comandate prin 68020.

Considerăm însă oportun, în contextul ulterioarelor dezvoltări, să relevăm încă unele caracteristici de structură cu care este înzestrat microprocesorul 68020, între care amintim mecanismele de control speciale menite a accelera execuţia instrucţiilor. Astfel, până la trei instrucţii pot fi executate simultan printr-un pipeline de instrucţii, care, în medie, introduce un grad de paralelism în execuţia programelor. O altă facilitate de accelerare o constituie memoria cache pentru instrucţii (instruction cache - fig.2.16) de 256 bytes. Ea permite lui 68020 aducerea în devans de instrucţii (prefetch instructions) din memoria principală în intervale de timp când magistrala de sistem ar fi altfel liberă şi încărcarea lor în cache-ul de instrucţii. Dacă, în continuare, aceste instrucţii sunt solicitate pentru execuţie de către CPU, ele pot fi obţinute mult mai rapid din cache-ul integrat în capsula procesoare (on-chip) decât din memoria principală, externă chip-ului procesor (off-chip).O altă caracteristică de structură la 68020, care este una neuzuală, constă în utilizarea unei microprogramări pe două niveluri prin care se creşte flexibilitatea proiectării cu concomitenta reducere a ariei de substrat necesare integrării în raport cu mult convenţionalul control microprogramat pe un singur nivel. Apelând la argumentaţia din literatura Motorola, vom insista asupra acestui aspect, sens în care arătăm că există maşini microprogramate la care microinstrucţiile nu determină, în mod nemijlocit, generarea semnalelor care să comande hardware-ul, ci ele sunt utilizate pentru a accesa o a doua memorie de control, denumită memorie de nanocontrol (nanocontrol memory, nCM), în care sunt stocate aşa numitele nanoinstrucţii (nanoinstructions), care aduse în registrul de nanoinstrucţii (nanoinstruction register, nIR) vor comanda, în mod direct sau cu o decodificare intermediară -dependent de orizontalitatea nanoprogramării -hardware-ul. Cele expuse

sunt ilustrate în fig.2.18, pe care se poate distinge nivelul superior al memoriei microcontrol (microcontrol memory, μCM), a cărui conținut îl reprezintă microinstrucțiile care sunt accesate printr-un prim ciclu de aducere (fatch) fiind încărcate în registrul de microinstrucții (microinstruction register, μIR) și constituind adrese pentru un al doilea acces, efectuat în nCM,în vederea adusului în nIR a nanoinstrucțiilor.

Dezavantajul acestui concept pe două niveluri constând din reducerea de viteză prin accesul suplimentar la nCM, precum și din suplimentarea costului revendicată de organizarea mai complexă a unității de control este contrabalansat, pe de o parte, de flexibilitatea mai mare de proiectare, consecință a "reloxării" instrucțiilor în raport cu hardware-ul comandat prin cele două niveluri de control intermediare în raport cu unul singur la versiunea convențională de microprogramare.

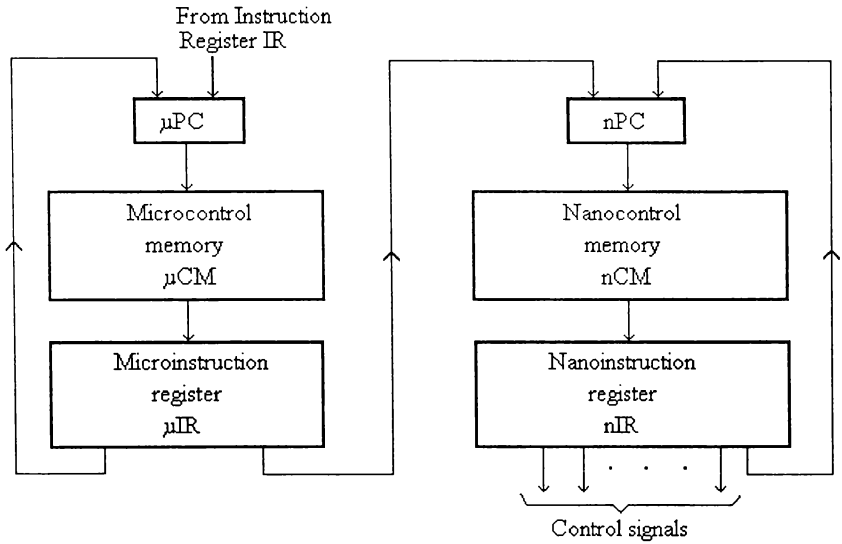


Fig. 2.18

În continuarea aceleiași idei, nanoprogramarea oferă, de asemenea, eficientizarea capacității de emulare a seturilor de instrucții corespunzătoare unui număr lărgit de calculatoare diferite. În al doilea rând, nanoprogramarea permite conservarea spațiului de substrat disponibil integrării, un parametru cu deosebire critic pentru nivelul tehnologic al integrării pe scară foarte largă Pentru a ne apropia această idee, să plecăm de la cele ilustrate în fig.2.19,cu varianta convențională sugerată în fig.2.19a și, prin contrast, varianta nanoprogramată în fig.2.19b. Pe parcurs vom accepta unele premise menite a simplifica analiza comparativă a celor două soluții tehnice de control, dar care nu vor altera concluzia finală.

Astfel, vom admite că memoria de control convențională CM (fig.2.19a), cu un singur nivel, stochează H_m microinstrucții orizontale,fiecare dintre acestea având un format simplu constând din N biți de control și din $\lceil \log_2 H_m \rceil$ biți de adresă, unde barele $\lceil \ \rceil$ semnifică cel mai mic întreg mai

mare decât numărul cuprins între bare. Spațiul S_1 revendicat de integrarea memoriei CM este, prin urmare, dat de relația:

$$S_1 = H_m (N + \lceil \log_2 H_m \rceil) \tag{2.9}$$

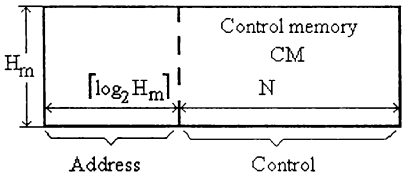


Fig. 2.19a

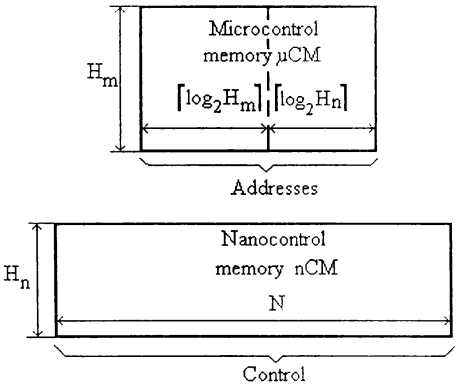


Fig. 2.19b

Fig. 2.19

La o organizare pe două niveluri (fig.2.19b), memoria de microcontrol μCM stochează, din nou, Hm microinstrucții, dar câmpurile de control de N biți sunt transferate în nCM. În locul acestora din urmă, fiecare microintrucție din μCM conține o adresă de $\lceil \log_2 H_n \rceil$ biți pentru a specifica orice locație de nanoinstrucție din nCM. Mai presupunem că, nu sunt prevăzute nanoinstrucții de salt (sau că numărul acestora este neglijabil),astfel încât în modelul de nCM nu sunt incluși biți expliciți pentru adrese. Cu aceste precizări, spațiul S_2 revendicat de integrarea celor două niveluri de memorii este, prin urmare, dat de relația :

$$S_2 = H_m (\lceil \log_2 H_m \rceil + \lceil \log_2 H_n \rceil) + NH_n \tag{2.10}$$

Mai admitem că toate modelele de biți de control (control-bit patterns) din nCM suntdiferite, astfel încât fiecare dintre acestea reprezintă o stare de control unică asociatăsetului de instrucții dat. Raportând numărul H_n al tuturor acestor stări de control unice la numărul total Hm al stărilor de control necesare implementării tuturor instrucțiilor şi folosind pentru acest raport notația $\gamma = H_n/H_m$, în urma substituirii lui H_n cu γH_m în (2.10), se obține:

$$\begin{aligned} S_2 &= H_m (\lceil \log_2 H_m \rceil + \lceil \log_2 H_m \rceil + \gamma N) = \\ &= H_m (2 \lceil \log_2 H_m \rceil + \lceil \log_2 \gamma \rceil + \gamma N) \end{aligned} \tag{2.11}$$

Dacă se face uz de valorile de parametri citați de literatura Motorola și anume Hm= =650, N = 70 și γ =0,4, rezultă H_n = 260 și, prin substituire în (2.9) și (2.11) se obține S_1 = 52000 și S_2 = 30550. În consecință, prin intermediul nanoprogramării, rezultă cantitatea $\Delta S = S_1 - S_2$ = 21450 de biți de memorie de control economisiți, constituind cca 41% din valoarea lui S_1. În același context, plecând de la condiția generală $S_2 < S_1$ și luând în considerare (2.9) și (2.11), se poate determina inegalitatea care trebuie satisfăcută pentru ca aplicarea nanoprogramării să fie justificată prin prisma economiei de arie de substrat pe care o determină. Procedând astfel, rezultă:

$$N > \frac{\lceil \log_2 Hm \rceil + \lceil \log_2 \gamma \rceil}{1-\gamma} \qquad (2.12)$$

cu N, evident, număr întreg.

S-a făcut această incursiune în caracteristicile de structură ale unui sistem de calcul pentru că, pe de o parte, concepția unui produs program de autotestare are drept premisă primară cunoașterea de detaliu, corespunzătoare unui anumit nivel, a construcției în vederea exploatării cît mai eficiente a tuturor facilităților. Pe de altă parte, în mod clar, disecția prezentării ar fi putut continua, insistând asupra detaliilor constructive la nivel de registru,sau, mai intim, la nivel de poartă logică ale facilităților de structură prezentate. În acest context, se conturează problema fundamentală constând din nivelul constructiv la care trebuie perceput defectul, armonizat cu nivelul la care urmează să fie întreprinsă reparația. Precizând în acest sens, fără a insista asupra problematicii de terminologie, dezvoltată în [Vasi-93], și restrângând considerațiile la defectele singulare de componente hardware (component defects [John-89]), din punct de vedere al defectului se poate elabora ierarhia din fig.2.28 [Prad-86].

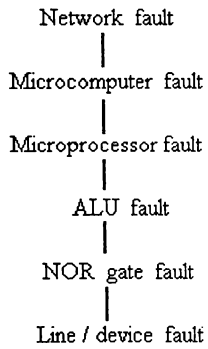


Fig. 2.20.

Unul și același defect implică mijloace de testare specifice corespunzătoare diferitelor niveluri de căutare a acestuia. O primă distincție se impune făcută dependent de scopul urmărit, care poate fi detecție sau diagnoză. Considerând în scop de exemplificare, fără a pierde din generalitate, nivelul ALU, în fig.2.21 se prezintă, principial, varianta fizică , modelul pentru detecția prin

programe a defectelor (fig.2.21b)şi modelul pentru diagnoza prin programe a defectelor (fig.2.21c). Procesul de testare în scop de detecţie este sugerat prin aplicarea în secvenţă a vectorilor binari de stimulare prin intermediul registrelor A şi B (admise, la verificarea ALU, cu funcţionare corectă) la două unităţi ALU, cea reală, testată şi ALUn admisă cu funcţionare normală, rezultatele urmând să fie, de fiecare dată, comparate. În realitate blocul ALUn constă din banca de răspunsuri corecte, iar blocul comparator din secvenţa de instrucţii de comparare.

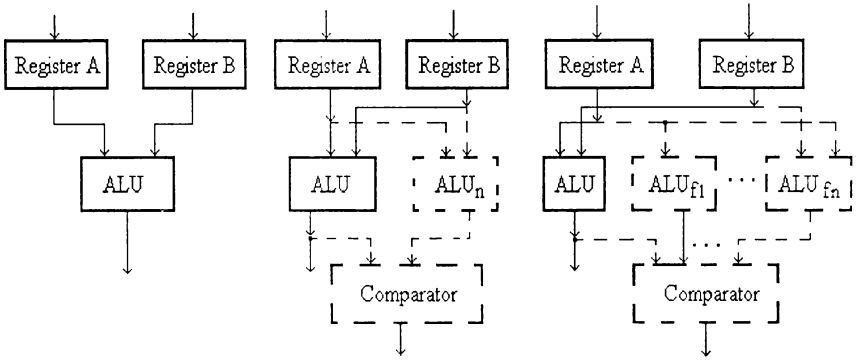


Fig. 2.21a Fig. 2.21b Fig. 2.21c

Fig. 2.21.

2.2. Metode hardware de implementare a toleranţei la defectare

Cu toate măsurile supuse discuţiei în anteriorul paragraf, prin care se evită o serie întreagă de defecte, din punct de vedere probabilistic rămân ,ca potenţiale, defecte, a căror combatere implică apelarea la redundanţă, adică la prezenţa unor mijloace funcţionale suplimentare. Aplicată la nivelul hardware, redundanţa obţine atributul de structurală,semnificând extensia sistemului cu structuri adiţionale, care pot consta în întregi echipamente de calcul identice într-un sistem multiprocesor cu redundanţă la nivelul global, dar şi în aplicaţii distribuite ale redundanţei la nivelul unităţilor funcţionale - cum ar fi, spre exemplu, unitatea de memorie - sau a componentelor sistemelor de operare. Deoarece obiectivul acestui prezentului paragraf constă în descrierea acelor metode hardware de implementare a toleranţei,importante în contextul prezentei lucrări, relevanţa tipului de redundanţă- globală, respectiv distribuită - fiind de importanţă secundară, referirile noastre vor fi făcute, fără a pierde din generalitate, la unităţi funcţionale sau canale. Vom preconiza, de asemenea, o prezentare gradată a conceptelor, începând cu unele care necesită redundanţă redusă şi crescând apoi investiţiile odată cu pretenţiile unui câştig de fiabilitate sporit.

Procesul de testare în scop de diagnoză decurge în mod similar cu excepţia faptului că blocul ALUn este substituit prin blocurile $ALUf_1$,.....,$ALUf_n$ constituind versiuni în care se transformă ALU real în prezenţa defectelor f_1,...,f_n, în realitate reprezentând băncile de răspunsuri eronate pe care le furnizează în prezenţa defectelor f_1, f_2, ... , f_n, blocul ALU şi care permit, prin comparare, identificarea defectelor.

Revenind la fig.2.20, odată detectat, să admitem, un microcalculator defect dintr-o rețea, dacă este necesară repararea lui, atunci se impune mai întâi detecția defectului la nivelul componentelor de structură ale acestuia, cum ar fi microprocesorul, după care se impune localizarea defectelor ș.a.m.d. Se conturează ideea de migrare de sus în jos în ierarhia defectelor până se atinge nivelul unității tehnologice replasabile purtătoare a malfuncționării, care poate fi o capsulă de circuit integrat, dar și un modul hardware care împachetează mai multe capsule. În ceea ce privește microprocesorul, testarea acestuia se poate întreprinde, în primul rând, prin programele de aplicație ale configurației care îl include. Evident, relevanța fenomenelor de malfuncționare corespunzătoare unei astfel de verificări este limitată. În al doilea rând, programe de testare pentru microprocesoare pot fi elaborate prin complexe procese de simulare care, plecând de la modelul de defectare prin blocare [Vasi-93], generează un set foarte amplu de secvențe binare care urmăresc, în ultimă instanță, comutarea fiecărui tranzistor a schemei integrate. Desigur, astfel de programe acoperă orice defect, fiind de maximă utilitate, producătorului unor astfel de capsule. Datorită volumului lor mare, pe lângă spațiul de memorare pe care îl revendică, ele reprezintă și o redusă capacitate de trecere, solicitând timpi îndelungați de testare. Pentru utilizatorul de capsule microprocesoare se dovedește mai eficient un al treilea concept care are la bază testarea instrucțiilor și a modurilor de adresare. Această metodă prezintă variante de implementare dependent de maniera în care se ține cont de structura internă în care capsula microprocesoare care are drept consecință o anume înlănțuire a instrucțiilor în procesul de verificare. Cu toate că în practică își găsesc răspândire versiuni combinate, se conturează ca variante extreme la un pol un concept cu orientare preponderent hardware, așa numita activare modulară, iar la celălalt pol un concept cu orientare software, bazat pe verificarea instrucțiilor.

2.3.2. Conceptul activării modulare

Activarea modulară implică disecția structurii interne integrată în capsula microprocesoare în module și ordonarea acestora într-o ierarhie. Nivelului inferior al acestei ierarhii i se asociază componentele de structură atribuite nucleului hardware inițial. La următoarea treaptă a ierarhiei se asociază acel modul hardware a cărui verificare poate fi întreprinsă apelând doar la instrucții care activează module ce aparțin nucleului hardware inițial. Dacă acest modul acoperă mai multe funcții a căror implementare revendică instrucții diferite, atunci apare ca necesară elaborarea unei noi ierarhii,de data aceasta a instrucțiilor.La treapta inferioară a acestei noi ierarhii se atribuie instrucția implicând numărul cel mai mic de microoperații, adică cea care duce la generarea numărului cel mai mic de semnale de control, deci activează, în ultimă instanță, cele mai puține circuite. Respectând o ordine de creștere în complexitate a instrucțiilor pe seama activării unui număr tot mai mare de microoperații, se alcătuiește ierarhia instrucțiilor asociată modulului. În mod uzual, nucleului hardware inițial îi succede modulul numărător de program (PC), dar nu cu funcția de încărcare asociată nucleului hardware inițial, ci cu funcția de incrementare care permite verificarea capacității de baleiere a întregului câmp de adresare. Cu toate că această funcție poate fi activată prin mai multe instrucții, practic toate având prevăzută această funcție, este preferată cea mai simplă în sensul

menționat, uzual, constituită de instrucția NOP a cărei unică menire constă în incrementarea PC-ului. Această etapă de verificare poartă, în general, denumirea "free-run" şi este momentul să relevăm că, în scopul implementării acesteia, în locul inserării unui lanț de NOP-uri care solicită spațiu de memorare excesiv şi revendică prin ciclurile de fetch implicate, timp de verificare îndelungat se prezintă favorabilă prevederea unei facilități hardware. Această soluție pro-testabilitate implică anticiparea dintr-o fază incipientă de concepție a moului de funcționare "free-run" prin posibilitatea deschiderii- spre exemplu, prin intermediul unui comutator sau utilizarea de punți-, a magistralei de date şi prevederea "forțării" pe aceasta a codului instrucției NOP. Răspunsurile la o astfel de stimulare se urmăresc la nivelul magistralei de adresă, scop în care pot fi utilizate, evident, răspunsuri memorate, dar şi aparatură de testare, cum ar fi analizorul de semnături sau analizorul de date. Dacă verificarea astfel descrisă se soldează cu succes, modulul verificat pe această treaptă este ataşat la nucleul hardware inițial, determinând extensia acestuia şi putând contribui la verificarea modulelor atribuite la niveluri superioare.

La următoarea treaptă a ierarhiei se asociază modulul a cărui stimulare implică instrucții care, în modul gradat descris bazat pe ierarhizarea instrucțiilor, activează, pe lângă funcții ale modulului în cauză, doar circuistica atribuită nucleului hardware extins. Observarea răspunsurilor în vederea evaluării se realizează, în maniera descrisă, la nivelul magistralei de adresă. În mod uzual, urmează în ierarhia de module aşa numită pilă de registre, cuprinzând registrele de date (spre exemplu, D0 la D7-fig.2.16) şi de adresare (spre exemplu, A0 la A6 -fig.2.16).Corespunzător acestora se verifică, într-o ordine oarecare, funcțiile de încărcare, respectiv descărcare cu date constând din 0 peste tot, 1 peste tot, dar şi alternanțe de 0 şi 1, respectiv 1 şi 0, precum şi "plimbarea" unui 0 şi apoi a unui 1 în câmpuri de 1-uri, respectiv 0-uri. Sunt urmărite în acest mod defecte de tipul unor influențe parazite la nivelul substratului, aşa cum se manifestă ele în memorii RAM [ItLe-92,RaFu-89], cu probabilitate mai mare de apariție între celule adiacente pozițional. Pe proxima treaptă ierarhică succede, în mod uzual, indicatorul de stivă (sau a indicatoarelor de stivă, spre exemplu, la 68020, A7şi A7'-fig.2.16) .Corespunzător acestora se verifică pe lângă funcțiile de încărcare şi descărcare, cele de incrementare şi decrementare de care uzitează anumite moduri de adresare (autoindexare cu predecrementare, respectiv cu postincrementare la 68020).Este momentul să amintim că anterior amintitele instrucții LINK şi UNLK pot fi înserate în ierarhia instrucțiilor ataşată indicatorului de stivă doar acum, în urma verificării funcțiilor elementare corespunzătoare modulului indicator de stivă. În mod normal, în ierarhia modulelor succede ALU cu verificarea graduală a corectei funcționări a instrucțiilor de adunare/scădere şi, apoi, a mai complexelor instrucții de înmulțire şi împărțire, în cazul în care acestea sunt cablate (la 68020, sunt cablate operațiile în virgulă fixă), şi, în mod similar, a instrucțiilor de operare logică. Evident, dacă în structura internă a microprocesorului unul dintre registre este predestinat ca acumulator cu funcții specifice de inițializare, încărcare, memorare, deplasare la stânga şi dreapta, rotire la stânga şi dreapta, incrementare şi decrementare, verificarea acumulatorului trebuie să preceadă pe cea a modulului ALU. De asemenea, la verificarea acestuia din urmă, secvențarea corectă a instrucțiilor în ierarhia corespunzătoare acestora trebuie să se bazeze pe detaliile constructive la nivel de poartă logică. Uzual, în urma verificării ALU se trece la un modul

care acoperă circuistica de procesare a excepţiilor şi a semnalelor de control externe de tip întrerupere. Şi în acest caz, corecta secvenţare a testării necesită cunoaşterea detaliilor de implementare a mecanismelor de procesare a excepţiilor. Insistând într-o anumită măsură asupra caracteristicilor funcţionale ale procesării excepţiilor la familia 68000, menţionăm că aceasta este tratată pe bază de priorităţi. Nivelul de prioritate al unei excepţii determină dacă aceasta poate fi întreruptă de către o altă excepţie. În general, procesoarele familiei 68000 vor recunoaşte o cerere de deservire a unei excepţii dacă nu este în derulare o altă excepţie sau dacă funcţia solicitată este de un nivel mai înalt decât excepţia activă. Astfel, în tabelul din fig.2.22 se prezintă gruparea pe niveluri de prioritate a excepţiilor, menţionând că, exceptând cele din grupa 2 care nu au alocate priorităţi individuale, celălalte au asigurate priorităţi în cadrul grupelor. Dacă excepţiile din grupa 1 aşteaptă execuţia completă a instrucţiei în curs de derulare, cele din grupa 0 sunt declanşate între două cicluri de magistrală.

Referindu-ne la bus error, operarea cu magistrala asincronă poate conduce la situaţii când, datorită faptului că procesorul nu recepţionează semnalul de recunoaştere (data acknowledge-$\overline{\text{DTACK}}$), execuţia instrucţiei nu poate fi terminată, microprocesorul rămânând "agăţat" pe instrucţia respectivă. Soluţia o constituie procesarea unei excepţii de bus error, care asigură terminarea ciclului de magistrală iniţiat de către procesor. Condiţia de bus error nu este detectată în mod automat de către procesor, ea trebuind semnalată procesorului prin logică externă care activează intrarea $\overline{\text{BERR}}$ a procesorului plasând 0 pe această linie. Un exemplu de schemă logică care să determine dacă un anumit ciclu de magistrală s-a terminat sau nu este aşa numitul watchdog timer, care este declanşat la debutul unui ciclu de magistrală şi care include un monostabil armat corespunzător unui interval de timp tolerat ca maxim pentru teminarea ciclului de magistrală. Dacă semnalul de control al magistralei care indică terminarea ciclului acesteia nu soseşte în acest interval prestabilit de timp, watchdog timer-ul setează $\overline{\text{BERR}}$ pe 0 şi procesorul abandonează ciclul de magistrală curent iniţiând o comutare în mediul de programare şi trecând la execuţia rutinei de deservire a excepţiei. Aceasta prevede încercarea de corectare a condiţiei de bus error prin reluarea ciclului de magistrală şi, în caz de nereuşită, este semnalată apariţia erorii de magistrală prin afişarea sau imprimarea adresei la care a apărut eroarea, precum şi a tipului ciclului de magistrală.

Excepţia de reset, care asigură iniţializarea sistemului, apare, în mod tipic, la conectarea sau la restabilirea dintr-o condişie de malfuncţionare, cum ar fi , spre exemplu, condiţia de bus error. Linia $\overline{\text{RESET}}$ este bidirecţională, asigurând iniţializarea procesorului când pe această linie este aplicat un semnal de către hardware-ul extern şi iniţializarea resurselor sistemului când procesorul trimite semnalul spre hardware-ul extern. O excepţie de reset a procesorului este iniţiată la comutarea pe 0 a intrării $\overline{\text{RESET}}$, nivel logic care trebuie să se menţină un interval minim de 100 msec. Această funcţie excepţie este de cea mai înaltă prioritate, secvenţa de procesare a excepţiei fiind declanşată întotdeauna. Printre microoperaţiile specifice cu care debutează această excepţie este prevăzută setarea unei măşti de întrerupere la valoarea 7 care reprezintă nivelul de prioritate cel mai înalt, în felul acesta asigurându-se mascarea tuturor întreruperilor externe cu nivel diferit de 7, prevenindu-se deservirea acestora. Rutina de deservire a excepţiei de reset prevede funcţii cum ar fi ştergerea

registrelor interne de date şi adrese, încărcarea indicatorului de stivă utilizator (USP) şi modificarea conţinutului byte-ului de sistem din registrul de stare (SR) pentru a autoriza întreruperile. Pe de altă parte, funcţia de ieşire $\overline{\text{RESET}}$, este declanşată software prin instrucţia RESET, a cărei execuţie nu afectează registrele interne ale procesorului. În schimb, linia $\overline{\text{RESET}}$ este setată să acţioneze ca o ieşire şi este generat un impuls de nivel logic 0 cu durată de 124 perioade de clock, care impuls este aplicat la intrările de reset, clear sau preset ale dispozitivelor externe, cum ar fi bistabile sau circuite integrate LSI periferice, pentru a iniţializa funcţionarea acestora.

Group	Exception	Processing
0	Reset Bus error Address error	Exception processing begins within two cycles
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV CHK Zero Divide	Exception processing is started by normal instruction execution

Fig. 2.22

Inclusă în aceiaşi grupă 0 împreună cu excepţiile reset sau bus error, excepţia address error se deosebeşte de anterioarele prin faptul că nu este declanşată extern, fiind o funcţie excepţie internă (built-in) în procesor. Întotdeauna când se încearcă citirea sau scrierea unei date de lăţimea unui cuvânt de la o adresă impară, procesorul recunoaşte, în mod automat, un astfel de acces ca o eroare de adresare. După detecţie, este iniţiată o secvenţă de microoperaţii, în principiu asemănătoare cu cele de la excepţia bus error, care transferă controlul la rutina de deservire a excepţiei address error. Aceasta încearcă să corecteze condiţia de eroare şi, dacă nu este posibilă corecţia, apariţia erorii de adresare este semnalată prin afişarea adresei şi tipului de acces.

Trecând la excepţiile din grupa 1, tot cu generare internă ca şi address error, se prezintă excepţia trace. Procesorul este prevăzut cu o opţiune de trace care permite un mod de operare pas cu pas şi care poate fi autorizată sub control software prin comanda bitului T din registrul de stare (SR). Când este autorizat acest mod de operare, procesorul iniţiază excepţia trace la terminarea execuţiei oricărei instrucţii. Această rutină excepţie transferă controlul la un monitor care permite examinarea conţinuturilor registrelor interne ale procesorului sau ale memoriei externe, informaţii utile la verificarea execuţiei instrucţiilor şi, prin urmare, la depanarea software.

După cum s-a amintit anterior, procesoarele din familia 68000, au abilitatea de a asigura medii de programe diferite utilizator-supervizor, starea de operare putând fi selectată sub control software. Importanța acestei facilități rezidă în faptul că unele resurse de sistem pot fi accesate doar de către sistemul de operare, mod în care au nivel de securitate sporit al sistemului. O altă excepție internă a procesorului permite identificarea situației când utilizatorul a încercat să acceseze o resursă supervizor. Aceste accese ilegale sunt denumite violări de stare privilegiată. Starea procesorului este determinată de valoarea bitului S din registrul de stare (SR), care atunci când este 0, procesorul este în stare utilizator şi, atunci când este 1, procesorul este în stare supervizor.În această din urmă stare, procesorul poate executa toate instrucțiile din repertoriul de instrucții. Când, însă este în stare utilizator, unele instrucții considerate privilegiate - cum ar fi instrucțiile de AND, OR şi EX-OR a unui cuvânt operand adresat în modul imediat cu conținutul lui SR-nu pot fi executate.Orice tentativă de execuție a unei instrucții privilegiate în stare utilizator conduce la o excepție privilage violation, a cărei rutină de deservire poate semnala apariția unei violări de stare privilegiată şi să asigure mijloace de restabilire a acesteia.

Excepția illegal/unimplemented instructions permite detecția, în mod automat, a apartenenței codului de operații al instrucției curente la setul corespunzător repertoriului de instrucții, cazul eşecului fiind coincident cu identificarea unui cod ilegal care determină blocarea tentativei de execuție a respectivei instrucții şi concomitenta declanşare a procesării unei excepții menită semnalării unei condiții de eroare. Conceptul de instrucție neimplementată (unimplemented) reprezintă o extensie a mecanismului de detecție a unei instrucții ilegale prin care poate fi extins repertoriul de instrucții. Există două domenii pentru codurile de operații neutilizate care permit definirea de noi instrucții, şi anume cele de forma $FXXX_H$ şi $AXXX_H$,unde X poate fi oricare cifră hexazecimală. Totodată când este detectat un astfel de cod, controlul este transferat la o rutină de procesare a excepției care este o rutină de emulare pentru noua instrucție, scrisă şi depanată în limbaj de asamblare şi stocată în memoria principală. Pentru a utiliza noua instrucție-care poate fi , spre exemplu, una de aritmetică în virgulă flotantă sau de operare în dublă precizie- într-un program se inserează ca enunț de instrucție codul $FXXX_H$.

Grupul 2 de excepții include un număr de 5 instrucții (trap TRAP#n, trap on overflow TRAP V, check registers against bounds CHK EA,Dn, signed divide DIVS EA,Dn, unsigned divide DIVU EA,Dn), la care se adaugă anterior menționata instrucție de revenire RTE, care diferă de excepțiile inițiate hardware prin faptul că ele sunt declanşate ca rezultat al execuției unei instrucții. Fără a insista asupra lor, arătăm că instrucția de TRAP poate fi considerată o instrucție de întrerupere software, care permite programatorului să execute o cerere vectorizată a unei rutine de deservire a unei devieri, utilizată, în mod tipic, pentru apeluri de supervizor (supervisory calls).

La familia de procesoare 68000, întreruperile sunt acoperite prin termenul mai larg de excepții (exceptions), care include , în conformitate cu cele expuse, erori induse de către hardware şi devieri generate prin program, dar şi întreruperi IO externe. Uzitând la nivel de principiu de mecanismul întreruperilor vectorizate, fiecare excepție are asociat un număr de vector N pe 8 biți, care face referire, conform cu cele prezentate în fig,2.23 [Ha La-86], la o locație de memorie M (4N)

care stochează adresa (vectorul excepție) a programului de deservire pentru respectiva excepție Locațiile de memorie 0 : 1023 formează un tabel de vectori excepție care stochează 256 adrese de memorie pe 32 de biți utilizate pentru procesarea excepțiilor. Cea mai mare parte a tabelului de vectori este alocată pentru cei până la 192 de vectori de întrerupere furnizați de utilizator, iar restul de locații sunt preasignate de firma Motorola la tipuri specifice de excepții. De exemplu, dacă se întâlnește o instrucție de împărțire la zero, CPU-ul aparținând familiei 68000 execută o secvență de deviere care transferă controlul la programul a cărui adresă de start este memorată în locațiile M(20:23) cărora le corespunde numărul de vector de excepție N=5.

Sunt acceptate două tipuri de întreruperi vectorizate: un mod general (general mode), în care dispozitivul ce solicită întreruperea furnizează un număr de vector pe 8 biți făcând referire la un element din tabelul vectorilor de excepție, şi un mai simplu, mod autovector (autovector mode), care permițând ca un dispozitiv IO să solicite oricare din şapte vectori de excepție predeterminați a căror adrese (numere de vectori) sunt generate intern de către CPU.

Într-o descriere sumară, procesarea întreruperilor implică executarea unei secvențe specifice de microoperații. Astfel, la sfârşitul oricărui ciclu de instrucție, CPU-ul verifică dacă este în derulare o cerere de întrerupere şi testează prioritatea acesteia conform cu cele ce vor fi descrise în continuare. Dacă CPU-ul acceptă cererea, el suspendă procesarea normală de instrucții şi intră într-o secvență de răspuns la întrerupere. Mai întâi, CPU-ul salvează conținutul vechi al registrului de stare SR într-un registru de memorare temporară şi setează starea sistemului în modul supervizor.Apoi, dependent de semnalele de control generate către surse de întrerupere, fie citeşte un număr de vector N furnizat de către sursa de întrerupere (modul general de întrerupere), fie generează pe N intern (modul autovector). CPU-ul procedează apoi la salvarea conținutului (adresa de revenire) numărătorului de program PC, a vechiului conținut al SR şi altor informații interne prin introducerea acestora în stiva supervizor, una din cele două stive menținute în memoria principală M a unui sistem configurat cu procesor aparținând seriei 68000.În continuare, utilizând valoarea 4N în calitate de adresă, CPU-ul execută un ciclu de citire din memorie (memory read cycle), determinând aducerea din M a vectorului excepție M(4N) corespunzător lui N şi M(4N) este încărcat în PC şi este reluată procesarea normală a instrucțiilor.

O interfață hardware reprezentativă pentru întreruperi IO la sisteme configurate cu procesoare din familia 68000 este prezentată în fig.2.24. Cu funcție dublă - atât pentru a executa o cerere de întrerupere, cât şi pentru a indica nivelul de prioritate al acesteia - sunt prevăzute trei linii de control $\overline{IPL}$ (interrupt priority level). $\overline{IPL}$ = 0 semnifică faptul că nu există cerere de întrerupere, în timp ce $\overline{IPL}$ =i, unde i este cuprins între 1 şi 7, semnifică faptul că a fost revendicată o întrerupere de nivel i. La receptarea unei cereri de întrerupere ($\overline{IPL} \neq 0$), CPU-ul compară numărul corespunzător la $\overline{IPL}$ cu un câmp I de trei biți din registrul de stare SR în care este stocată masca de întrerupere. Dacă $\overline{IPL} < I$, cererea de întrerupere este ignorată, iar dacă $\overline{IPL} \geq I$, CPU-ul răspunde la cererea de întrerupere la sfârşitul ciclului său de instrucție curent. Întrucât conținutul registrului SR poate fi alterat prin anumite instrucții (privilegiate), faptul că CPU răspunde sau nu la o cerere de întrerupere se află sub control software

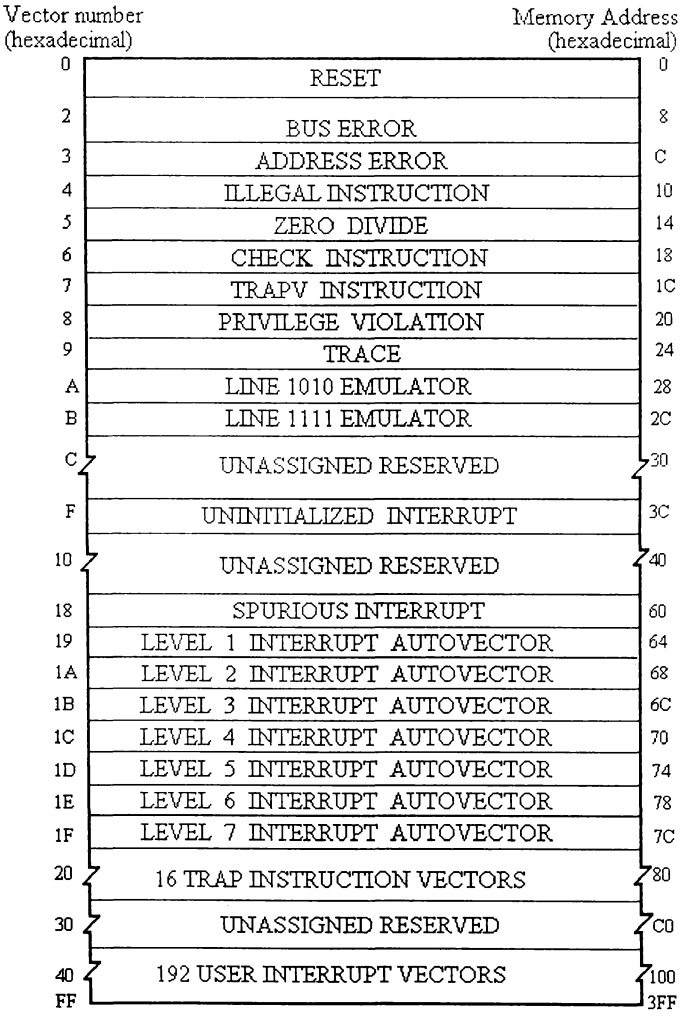


Fig. 2.23

Prin setarea măştii de întrerupere I pe 0 devin autorizate toate cererile de întrerupere, iar dacă I este setat la valoarea 7 atunci sunt rejectate toate întreruperile cu excepţia celor de cea mai înaltă prioritate ($\overline{IPL}$ =7), care sunt nemascabile. Sursele de întrerupere pot astfel utiliza până la 192 vectori de transfer, fiecăruia dintre aceştia putând fi asignaţi la oricare din cele şapte niveluri de prioritate.

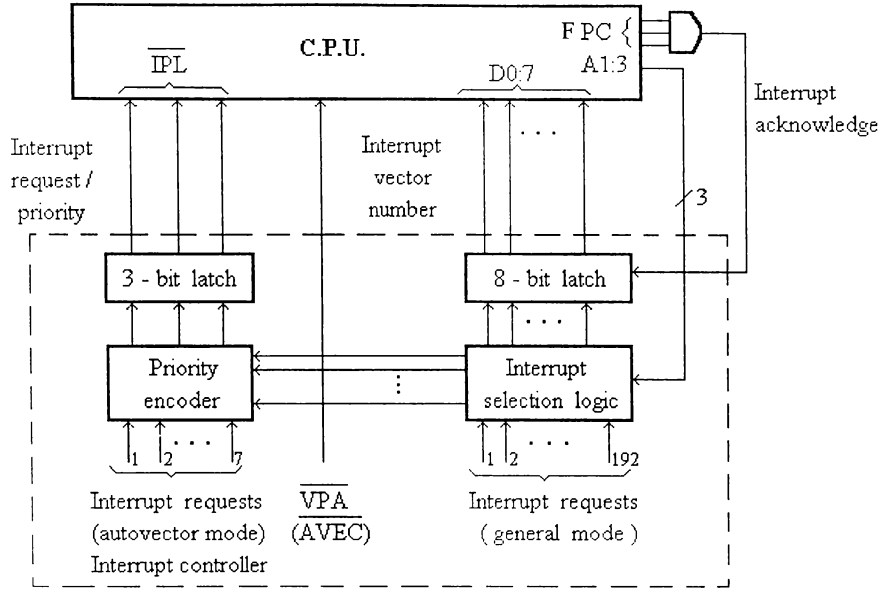


Fig. 2.24

Recunoaşterea de către CPU a unei cereri de întrerupere se realizează prin setarea pe 1 a liniilor de ieşire FC (function code) determinând generarea semnalului de acceptare a întreruperii. CPU-ul plasează, de asemenea, pe liniile de adresă A1:3 nivelul de prioritate al întreruperii recunoscute. În modul de întrerupere general (general mode) controller-ul de întrerupere răspunde prin plasarea pe liniile de date D0:7 a numărului de vector de întrerupere N.La soluția sugerată în fig.2.24 pentru un CPU implementat cu 68000, semnalele FC, trecute printr-o poartă ŞI, sunt utilizate, în mod direct, pentru a stroba numărul de vector de întrerupere N pe magistrala de date. Pentru a indica modul autovector (autovector mode), controller-ul de întrerupere răspunde, la recunoaşterea de către CPU a unei cereri de întrerupere, prin activarea unei linii de control speciale ($\overline{\text{VPA}}$ la un CPU cu 68000 şi $\overline{\text{AVEC}}$ la un CPU cu 68020), determinând ca CPU să genereze pe N intern utilizând formula N = 24 + IPL.

Revenind la problematica verificării corespunzătoare nivelului funcțiilor generate de semnale de control externe, menționăm înlănțuirea acestora plecând de la unele mai simple, implicând un număr mai mic de microoperații, şi continuând cu unele din ce în ce mai complexe, efectuând atribuirea lor la nucleul hardware în curs de extensie pe măsură ce controlul se soldează cu succes. În fine, la ultimul nivel al ierarhiei de module sunt atribuite acele funcții, activate prin instrucții de mare complexitate, care nu au intervenit încă în procesul de verificare, mod în care sunt supuse testării toate ieşirile decodificatorului de instrucție şi generarea tuturor semnalelor de control.

Concluziv, conceptul activării modulare conduce la un produs program de testare caracterizat printr-o secvențare a instrucțiilor și datelor, care urmărește o verificare graduală a componentelor de structură integrate în capsula microprocesoare, ceea ce implică, în calitate de premisă, cunoștiințe avansate cu referire la detaliile constructive interne ale circuitului testat. Înainte de a trece la prezentarea conceptului de testare a instrucțiilor, care, prin contrast, nu implică cunoașterea structurii interne a microprocesorului, ținem să menționăm o altă variantă de implementare a conceptului de activare modulară prin extensia nucleului hardware, și anume în strategia de verificare, de această dată, a structurii unui microsistem bazată pe analiza de semnături. Descrisă sumar, această metodă implică, mai întâi, disecarea din structură-care, fără a pierde din generalitate, admitem că, pe lângă microprocesor, cuprinde o bancă de memorii ROM, o bancă de memorii RAM și porturi IO- a unui nucleu hardware inițial alcătuit din capsula microprocesor la care se adaugă generatorul trenurilor de tact și sursele de alimentare. Într-o primă instanță, în regim de free run, implicând inhibarea semnalelor de control externe de tipul întreruperilor, se verifică capacitatea de baleiere a întregului câmp de adresare prin incrementarea PC-ului. În acest sens, reamintim importanța facilităților de natură tehnologică care să permită, pe de o parte, deschiderea buclelor de reacție constituite de liniile de date, și , pe de altă parte, posibilitatea de "forțare" pe acestea a codului de operație pentru acea instrucție simplă (spre exemplu, NOP) a cărei consecință unică să fie reprezentată de incrementarea PC-ului. Urmărirea răspunsurilor în vederea evaluării se efectuează prin culegerea fluxurilor informaționale prin sonda de date a analizorului de semnături la nivelul liniilor de magistrală de adresă, precum și a ieșirilor din logica de decodificare a adresei și de generare a semnalelor de chip select pentru celelalte componente de structură. Într-o proximă instanță, se revine din regimul free-run prin închiderea magistralei de date și, printr-o selecție adecvată a punctelor de ancorare a sondelor analizorului de semnături, se procedează la verificarea conținuturilor memoriilor ROM, urmărind, de această dată, fluxul informațional al răspunsurilor la nivelul liniilor de magistrală de date. În urma soldării cu succes a testării memoriilor ROM, inclusiv a sumelor de control corespunzătoare conținuturilor acestora, verificarea continuă cu memoriile RAM în care scop, în baza principiului de extensie a nucleului hardware, se face uz de un program de testare stocat într-una din capsulele ROM. Acesta din urmă are menirea încărcării memoriilor RAM cu configurații binare-uzual corespunzătoare testelor ad-hoc (MARCH, WAKPAT, GALPAT, GALTCOL, ș.a) - destinate relevării defectelor mai probabile (selecție concomitentă a mai multor adrese, cuplaje parazite la nivelul substratului între celule cu poziție adiacentă, funcționare necorespunzătoare a amplificatoarelor de citire ș.a.). În urma stimulării în această manieră a memoriilor RAM , din nou printr-o selecție adecvată a pinilor pentru sondele analizorului de semnături, astfel încât să se excludă conflicte de date, se efectuează citirea în secvență a adreselor de memorie urmărind fluxurile de informație, în ordine, pe fiecare dintre liniile de date ale magistralei. În urma verificării cu succes a memoriilor RAM se efectuează testarea porturilor IO prin prealabila încărcare a porturilor ieșire cu configurații binare constând din 0 peste tot, 1 peste tot, alternanțe de 0 și 1, precum și "plimbări" de 0 în câmp de 1-uri, respectiv de 1 în câmp de 0-uri. În scopul verificării la acest nivel este reclamată o altă facilitate reclamată de testare, și anume capacitatea

tehnologică de deconectare a perifericelor şi asigurare, prin corespunzătoare cabluri auxiliare utilizate în regim de testare, a închiderii de bucle între porturi IO de ieşire şi porturi IO de intrare. Această premisă îndeplinită, se pot urmări prin corespunzătoare aplicări ale sondelor analizorului de semnături configuraţiile încărcate în porturi ieşire la nivelul magistralei de date, în urma trecerii lor prin porturile intrare.

Mai menţionăm că o concepţie asemănătoare activării modulare prin extensia nucleului hardware stă, de asemenea, la baza strategiei de testare prin emulare in-circuit. În esenţă, aceasta prevede capacitatea tehnologică de înlăturare a capsulei microprocesoare şi, eventual, a altor circuite logice VLSI programabile prin scoaterea acestora din soclurile aferente şi asigurarea execuţiei funcţiilor lor prin circuistica echipamentului de testare. Se verifică astfel, excitând mai întâi funcţiile mai simple şi crescând gradual complexitatea acestora, logica auxiliară a acestor circuite integrate complexe, iar în cazul soldării cu succes a testării dar a nefuncţionalităţii sistemului, aceasta din urmă este atribuită capsulelor scoase din soclu. Încheiem prin reliefarea încă odată a importanţei asigurării facilităţilor- începând cu baze incipiente de concepţie şi până la detalii de factură tehnologică- menite a simplifica complicatele sarcini de testare.

2.3.3. Conceptul verificării instrucţiilor

Conceptul de elaborare a programelor de autotestare pentru microprocesoare se bazează pe modelarea defectării la nivel de instrucţii şi de transferuri între registre, care permite determinarea secventării instrucţiilor şi a modelelor de biţi necesare verificării [Prad-86]. Un microprocesor poate fi modelat prin intermediul unui graf ale cărui noduri sunt reprezentate de registre sau seturi de registre şi ale cărui arce sunt reprezentate de transferuri de informaţie. Pentru procesoare ale familiei 68000 graful corespunzător este ilustrat în fig.2.25.

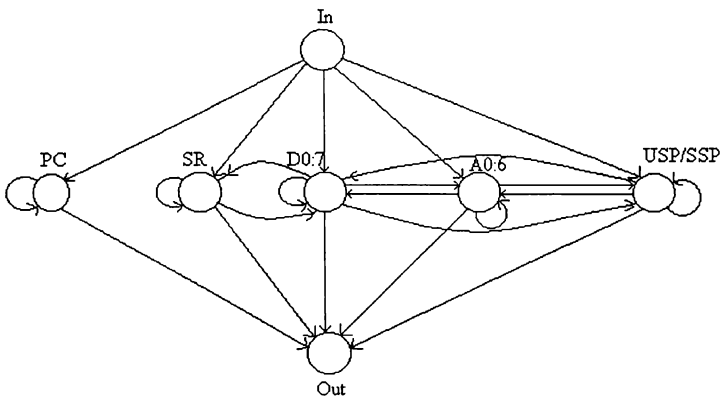


Fig. 2.25

Cele opt registre de date D0:7 şi cele şapte registre de adresare A0:6 formează câte un set de registre echivalente. Se remarcă faptul că indicatoarele de stivă A7(user stack pointer,USP) şi A7'

(supervisor stack pointer,SSP) nu aparţin clasei de echivalenţă a registrelor de adresare întrucât există instrucţii, cum ar fi, spre exemplu, MOVE USP, care se referă la A7 în mod implicit şi care nu poate face referire la clasa de registre A0:6. Mediul extern este reprezentat de către nodurile IN, respectiv OUT, prin care trece informaţia înspre şi dinspre microprocesor. Pe de altă parte, instrucţiile constând din secvenţe de microinstrucţii, iar acestea, la rândul lor, constând din secvenţe de microoperaţii (dar conceptul nu este limitat la procesoare microprogramate)- sunt divizate, dependent de complexitatea acestora, în trei tipuri. Astfel, sunt admise ca aparţinând tipului 0 acele instrucţii care operează asupra unui singur registru, tipului 1 acele instrucţii care implică un transfer de date dintr-un registru în altul sau operaţii logice şi, în fine, tipului 2 i se asociază instrucţiile aritmetice.

Acceptând, în calitate de exemplu, selectiv instrucţii din seturile corespunzătoare procesoarelor familiei 68000, în tabelul din fig.2.26 se prezintă câteva instrucţii pentru fiecare din cele trei tipuri. În ceea ce priveşte modelele de defectare considerate pentru variatele funcţii ale microprocesorului, acestea sunt grupate în următoarele categorii [Prad-86]:

Type 0	SWAP, CLR, NOT, NEG, BTST, BSET, BCLR, BCHG, SHIFTLEFT, SHIFTRIGHT
Type 1	MOVE, AND, OR, EXOR, EXG
Type 2	ADD, SUB, MULT, DIV

Fig. 2.26

1.Funcţii de decodificare defecte, când, spre exemplu, funcţia de decodificare vizând un anumit registru Ri nu se execută sau duce la selecţia unui alt registru Rj($j \neq i$) sau duce la selecţia concomitentă, pe lângă a registrului dorit Ri şi a unui alt registru Rj($j \neq i$); 2.Funcţii defecte de transfer de date, când, spre exemplu, un număr de linii sunt blocate la 0 (s-a-0) sau la 1 (s-a-1) sau când două linii, i şi j, sunt afectate de cuplaje parazite, astfel încât semnalul transmis pe linia i este influenţat de către cel transmis pe linia j (de exemplu, prin diafonie); 3.Funcţii defecte de manipulare a datelor vizând unitatea aritmetică şi logică, a căror verificare se efectusază printr-o secvenţă de instrucţii implicând transferuri de operanzi în registrele sursă, apoi execuţia operaţiei de testat şi, în final, citirea rezultatului din registrul destinaţie în memorie;4.Funcţii de secvenţare şi generare a semnalelor microoperaţii, care, în condiţii de defectare, pot conduce la situaţii când una sau mai multe microoperaţii rămân inactive, astfel încât instrucţia se execută incomplet, sau microoperaţii, în mod normal inactive, devin active sau este activat un set de microinstrucţii în plus sau în locul celor normale.

Plecând de la descrierea modalităților de defectare admise, se impun totuşi unele restricții constând, în primul rând, din faptul că prin defect nu creşte numărul de operații de scriere în memorie. În al doilea rând defectarea nu determină depăşirea unui număr K de microinstrucții corespunzătoare unei anumite instrucții, unde K depăşeşte cu o unitate numărul de microinstrucții corespunzător acelei instrucții din setul dat constituită din secvența cu numărul cel mai mare de microinstrucții, şi, de asemenea, nu determină depăşirea unui număr M de microoperații corespunzătoare unei anumite microinstrucții, unde M depăşeşte cu o unitate numărul de microoperații corespunzătoare acelei microinstrucții având numărul cel mai mare de microoperații generate în secvență. Limitările legate de valorile K şi M, care pot fi determinate cu uşurință pentru fiecare microprocesor, se impun pentru a conferi eficiență, deci productivivate, procesului de testare.

Defectarea la nivelul unei microinstrucții este detectată prin memorarea unui set adecvat de date în registrele microprocesorului, astfel încât anumite date să fie distruse în prezența defectului. Pentru simplitate, considerând microprocesorul 68000, în tabelul din fig.2.27 se sugerează cuvinte de cod care să fie încărcate în registrele microprocesorului. Aparținând unui cod Berger, orice defect constând dintr-o microoperație adițională, care operează cu două cuvinte sursă, va produce un cod invalid la destinație, a cărui detecție este imediată prin modificarea numărului de 0-uri, respectiv a celui de 1-uri care o determină.

Procedura de înlănțuire a instrucțiilor în scopul verificării microprocesorului implică traversarea următorilor paşi:

Pas 1. Definirea şi verificarea unui set nucleu (core set) de instrucții care permite implementarea unei aşa numite funcții de citire Read (Ri) a unui anumit registru Ri. În esență, acest set nucleu, fără a cărui funcționalitate nu poate începe procesul de verificare, implică o instrucție de încărcare a unei date de la o locație din memorie fără a modifica starea microprocesorului, apoi o instrucție de comparare a conținutului registrului încărcat cu o dată etalon şi, în fine, cel puțin o instrucție de salt condiționat în cazul că la comparare s-a obținut inegalitate.

Materializată această definire pentru seturile de instrucții corespunzătoare procesoarelor seriei 68000, se ajunge la nucleul format din instrucțiile MOVE, CMP,BEQ, BRA, care cuprinde, pe lângă instrucția de salt în caz de egalitate, şi instrucția de salt necondiționat, aceasta din urmă fiind, uneori, preferată pentru indicarea stării de eroare. Verificarea acestui set nucleu se întreprinde cu date constând din 0 peste tot, 1 peste tot, alternanțe de 0 şi 1, sau configurații tip "plimbare" a lui 0 în câmp de 1-uri, respectiv a lui 1 în câmp de 0-uri, date care sunt defavorabile prin prisma unor potențiale defecte, dar care pot fi şi aleatoare. Secvența de verificare a setului nucleu pentru procesoarele seriei 68000 este ilustrată în figura 2.28, unde d_1, d_2 şi d_3 reprezintă datele anterior menționate.

Register	Code pattern
D0	11111**0**11111111111111111111...1
D1	111111**0**11111111111111111111...1
D2	1111111**0**11111111111111111111...1
D3	11111111**0**11111111111111111111...1
D4	111111111**0**11111111111111111111...1
D5	1111111111**0**11111111111111111111...1
D6	11111111111**0**11111111111111111111...1
D7	111111111111**0**11111111111111111111...1
A0	1111111111111**0**11111111111111111111...1
A1	11111111111111**0**11111111111111111111...1
A2	111111111111111**0**11111111111111111111...1
A3	1111111111111111**0**11111111111111111111...1
A4	11111111111111111**0**11111111111111111111...1
A5	111111111111111111**0**11111111111111111111...1
A6	1111111111111111111**0**11111111111111111111...1
USP	11111111111111111111**0**11111111111111111111...1
SSP	111111111111111111111**0**11111111111111111111...1

Fig. 2.27

După cum poate fi remarcat, respectiva secvență implică execuția, la un moment dat, a instrucției de comparare de (K*M+1)ori, această execuție în buclă, bazat pe anterioarele definiții ale lui K şi M, asigurând, în cazul cel mai defavorabil, detecția în situația când un anumit defect tinde să mascheze eroarea provocată de alt defect.

Pas 2. Verificarea funcției de citire Read (Ri) a conținutului unui anumit registru Ri, constând , în fond, din testarea funcției de descărcare a conținutului registrului într-o locație de memorie fără a modifica starea microprocesorului. Funcția Read (Ri) implică testarea prin instrucțiile anterior verificate ale setului nucleu a corectei execuții a operației de descărcare/memorare, fără a perturba conținuturile altor registre.Verificarea funcției Read (Ri) este executată pe tipurile de instrucții anterior relevate (fig.2.26), începând cu cele de tip 0, corespunzător cărora registrul Ri este atât sursă cât şi destinație, continuând apoi cu cele de tip 1 care vizează o pereche de registre distincte Ri şi Rj între care există o operație utilizând pe Ri ca sursă şi pe Rj ca destinație şi terminând cu cele de tip 2 implicând trei registre Ri, Rj şi Rk (Ri şi Rj sunt distincte) pentru care există o operație utilizând pe Ri şi Rj ca surse şi pe Rk în calitate de destinație.

```
        MOVE # d1, R1
        MOVE # d1, R2
        CMP  R1, R2
        BEQ  a
        BRA  error
          ⋮
    a:  MOVE # d2, R1
        MOVE # d3, R2
        CMP  R1, R2
        BEQ  error
    b:  MOVE # d1, R1
        MOVE # d1, R2
        {do K*M+1 times
        CMP  R1, R2 end do}
        BEQ  c
        BRA  error
          ⋮
    c:  MOVE # d2, R1
        MOVE # d3, R2
          ⋮
    d:  CMP  R1, R2
        BEQ  error
        BRA  succes
error:  write error !
succes: NOP
```

Fig. 2.28

În fig.2.29 este sugerată secvența de testare pentru instrucțiile de tip 2 - implicând câte trei registre aparținând setului S_2 -, în care se apelează, din nou, la execuția multiplă, în buclă, menită a releva, în cazul cel mai defavorabil, mascarea efectului unei microinstrucții defecte prin eroarea provocată de o altă microinstrucție defectă.

```
for all Ri ∈ R and
for all (Rj, Rk, R1) ∈ S2 do {
1. Read (Ri)
2. Read (Rj)
3. { Read (Rk) K*M+1 times;
4. { Read (R1) K*M+1 times;
5. { Read (Rk); Read (R1) M*K+1 times}}}
6. Read (Ri)
7. Read (Rj) end do}
```

Fig. 2.29

Pas 3. Verificarea funcției de încărcare pentru fiecare din registrele Ri ∈R pentru toate modurile de adresare prevăzute. Această testare implică, de asemenea, ca datele încărcate să fie apoi descărcate și verificate, dar, la acest moment, se cunoaște că funcțiile menționate pe urmă sunt corecte datorită traversării pașilor anteriori. Astfel, în fig.2.30 se prezintă secvența de testare a încărcării registrelor, la care trebuie adăugat că ea se impune repetată pentru toate modurile de adresare.

```
   for all R_i ∈ R do {
   for all R_j ∈ R do {
1. Read ( R_i )
2. MOVE # d_1, R_i
3. Read ( R_j )
   end do }}
```

Fig.2.30

Pas 4. Verificarea corectei operări a setului de instrucții. În această etapă testarea instrucțiilor se efectuează doar pentru un singur mod de adresare, acesta pentru a eficientiza procesul de verificare întrucât testarea tuturor instrucțiilor cu toate modurile de adresare ar reclama un timp îndelungat. Simplificarea menționată apare rezonabilă, cu precădere, pentru acele microprocesoare la care defectele din circuistica secvențării instrucțiilor pot fi considerate ca independente de cele din circuistica de implementare a modurilor de adresare. În fig.2.31 se prezintă secvența de testare a instrucțiilor aparținând setului I al instrucțiilor încă neverificate, în care este prevăzută citirea registrelor interne după execuția fiecărei instrucții și verificarea dacă aceasta nu a determinat distrugeri de date.

```
   for all I_i ∈ I do {
1. Load the registers with the code words
2. Execute I_i
3. Read all the internal registers
   end do }
```

Fig.2.31

Cu toate că pașii descriși până acum asigură testarea tuturor instrucțiilor, pentru verificarea funcțiilor de memorare și transfer de date, precum și a celor de manipulare a datelor, în mod uzual, se apelează la încă doi pași.

Pas 5. Verificarea funcțiilor de stocare și transfer.Corespunzător acestei etape, banca registrelor de date și cea a registrelor de adresare, constituind memoria rapidă internă a microprocesorului, se supune unei verificări bazate pe clasicele teste utilizate la memoriile RAM semiconductoare. Astfel, se prevede încărcarea respectivelor registre cu configurații binare

corespunzătoare testelor ad-hoc (MARCH, GALPAT sau altele) şi se citesc respectivele confuguraţii cu concomitenta verificare executată prin instrucţii oricum verificate în anteriorii paşi.

Pas 6. Verificarea funcţiilor de manipulare a datelor. Această etapă este destinată verificării unităţii aritmetice şi logice, cu precădere, a părţii de aritmetică. În mod uzual se apelează la o testare exhaustivă prin baleierea tuturor combinaţiilor binare posibile pentru operanzi, dar, alternativ, se poate apela şi la o testare bazată pe vectori binari pseudoaleatori completaţi prin unii determinişti pentru condiţii excepţii cum ar fi, spre exemplu, împărţirea la zero.

Făcând aprecieri asupra parametrilor de performanţă corespunzători programelor de autotestare bazate pe conceptul verificării instrucţiilor, în [Prad-86] se arată că lungimea secvenţei de testare pentru detecţia defectelor prin funcţia de citire Read (Ri) este proporţională cu numărul n_R al registrelor interne din structura microprocesorului. Lungimea procedurii pentru detectarea defectelor la execuţia restului de instrucţii este proporţional cu ($n_i \neq n_R$), unde n_i reprezintă numărul instrucţiilor setului corespunzător microprocesorului. Se poate astfel aprecia că, prin prisma lungimii, testarea bazată pe acest concept este rezonabilă, chiar şi pentru microprocesoare cu structuri complexe. În ceea ce priveşte calitatea setului de teste elaborat prin această metodă, o bună măsură o constituie acoperirea defectelor (fault coverage), care necesită cunoaşterea detaliilor constructive interne ale microprocesorului şi dispunerea de un simulator de defecte (fault simulator). În aceiaşi sursă bibliografică [Prad-86] se arată că pentru un microprocesor pe 8 biţi la care au fost admise 2200 de defecte singulare de blocare (single stuck faults), 90% dintre acestea au fost detectate prin execuţia fiecărei instrucţii, la care s-au mai adăugat 6% evidenţiate prin teste de secvenţare a instrucţiilor pentru acele defecte care au determinat execuţia unor funcţii în exces sau lipsă faţă de cele dorite. Restul de 4% din numărul total de defecte admis au fost găsite ca redundante sau nedetectabile.

Părăsind conceptul de verificare a instrucţiilor, vom face o scurtă referire la testarea prin vectori binari pseudoaleatori a microprocesoarelor, apreciată, în general, ca ineficientă, datorită complexităţii inerente a structurilor integrate în capsule microprocesoare. Este totuşi notabilă încercarea întreprinsă în [THEV-83] de aplicare a conceptului de testare aleatoare exploatând informaţii despre arhitectura supusă verificării, în conformitate cu care problema este tratată în mod separat pentru partea de date, respectiv pentru partea de control. În ambele cazuri, modelul de defectare este dezvoltat la nivelul transferului între registre (register-transfer level), luând în considerare defecte de blocare în registre, defecte de decodificare a instrucţiilor şi defecte la circuitele de generare a funcţiilor de control. Bazat pe analiza defectelor, se determină valoarea L a numărului de vectori pseudoaleatori necesar detecţiei acelor defecte considerate ca mai dificil de pus în evidenţă. Pentru un microprocesor Motorola 6800 s-a obţinut că L poate rezulta de ordinul a 6 milioane de vectori, ceea ce ar corespunde la un timp de testare de ordinul a câtorva secunde cu un nivel de confidenţă de 0,999. În ceea ce priveşte structuri mai complexe de microprocesoare, cum ar fi cele aparţinând seriei 68000, se apreciază că L şi, implicit, productivitatea procesului de testare pot rezulta de valori inacceptabil de mari.

Indiferent de metoda în baza căreia au fost generate, programele de testare sunt stocate într-o memorie ROM sau sunt încărcate de pe o unitate de disc în memoria RAM pentru ca microprocesorul să înceapă procedura de autotestare. Pe parcursul programului este recomandabil ca fiecare parte de program corespunzătoare unei sarcini de testare să fie succedată de o sumă de control (checksum), modalitate prin care se asigure că microprocesorul parcurge întreaga secvență de testare, fiind evitate situațiile când, datorită unui anumit defect, microprocesorul verificat efectuează un salt la sfârșitul programului indicând starea sa de funcționalitate corectă, fără însă să parcurgă întreaga secvență de testare. În acest context, considerăm utilă, prezentarea modalităților de generare a sumelor de control. Astfel, în conformitate cu [John-89], sunt utilizate patru tipuri de sume de control: (1) de simplă precizie (single-precision), (2) de dublă precizie (double precision), (3) Honeywell şi (4) reziduală (residue). Contextul favorabil de prezentare a diferitelor tipuri de sume de control îl constituie transferul de blocuri de date, la care aderăm conform [John-89], dar menționăm că el poate fi adaptat cu uşurință la verificarea memoriilor cu precădere a celor ROM. Astfel, conceptul fundamental al sumei de control este ilustrat în fig,2.32, în care se poate remarca generarea corespunzător datelor originale a unei informații suplimentare constituită de suma de control. Aceasta din urmă este transmisă, împreună cu datele originale la recepție unde este comparată cu o informație generată în mod similar, iar neconcordanța chiar şi la nivelul unui singur bit indicând faptul că transmisia datelor a fost afectată de erori.

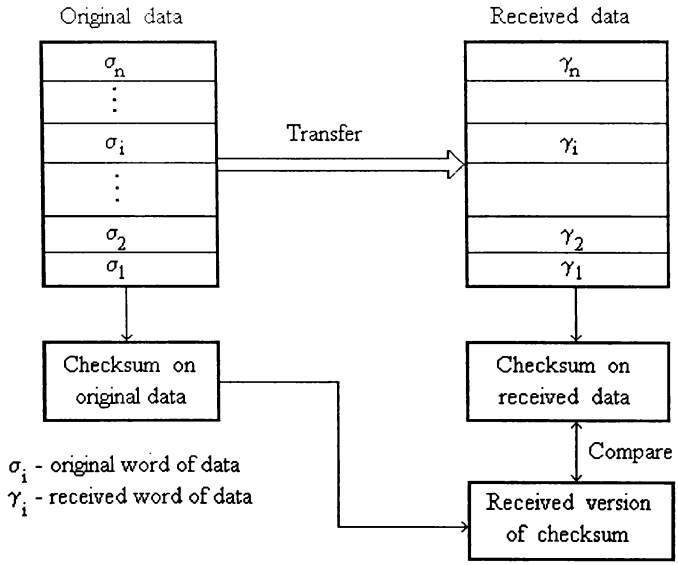


Fig.2.32

Tipurile de sume de control diferă dependent de caracteristicile lor de formare, cea mai simplă metodă constând din simpla adunare binară cu ignorarea depăşirii, a cuvintelor de date, pe

de o parte, originale şi, pe de altă parte, recepţionate. Suma de control generată astfel poartă denumirea de simplă precizie şi constă din acelaşi număr de biţi, admitem n, care este utilizat pentru reprezentarea datelor. Deficienţa acestei metode de verificare constă în capacitatea limitată de detecţie a erorilor cauzată de ignorarea depăşirii (overflow). Un exemplu ilustrativ în acest sens îl constituie transmisia reprezentată în fig.2.33, care, după cum se observă, este afectată prin defectul de blocare la 1 a liniei de date d_3. Pentru cazul particular considerat, se observă că la recepţie toate cele 4 cuvinte de date au bitul d_3 eronat,dar sumele de control pe de o parte, recepţionată şi, pe de altă parte, generată pentru datele recepţionate - rezultă egale, indicând , în mod fals, că transmisia s-a efectuat fără eroare. Ieşirea din impasul relevat o oferă formarea secvenţei de control de dublă precizie, caracterizată prin faptul că seturilor de date, cu fiecare cuvânt pe n biţi, li se ataşează sume binare ale cuvintelor de date generate, în acest caz, pe 2n biţi.

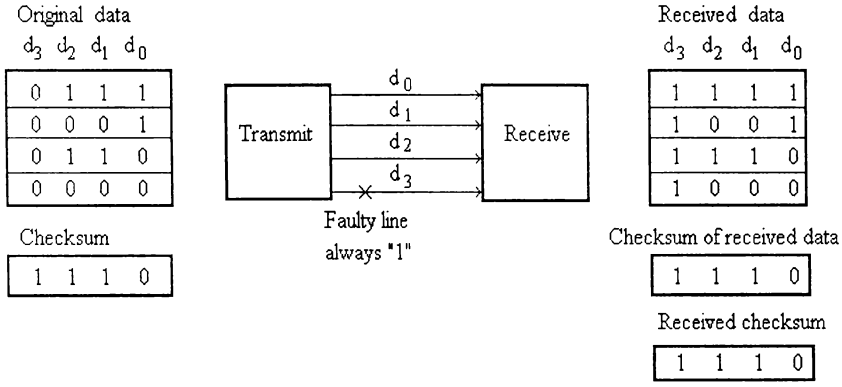


Fig.2.33

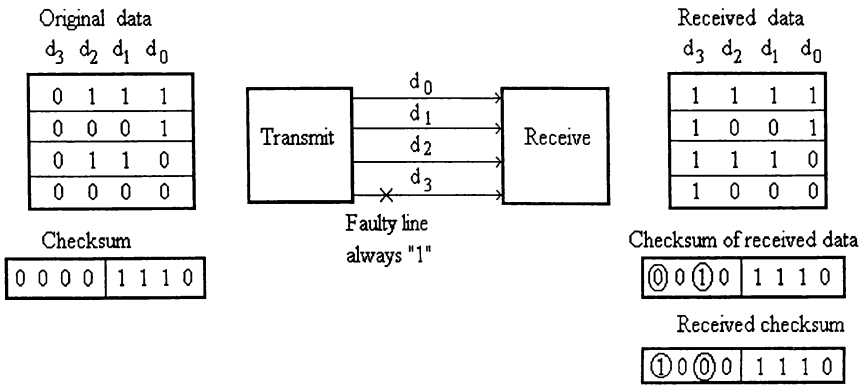


Fig. 2.34

Desigur, şi în acest caz avem de-a face cu ignorarea depăşirii, dar, de acestă dată, la aritmetica modulo 2^{2n}, astfel capacitatea de detecţie a erorilor este substanţial majorată. Luând în considerare acelaşi defect de blocare la 1 a liniei de date d_3 (fig.2.33), în fig.2.34 se prezintă sumele de control de dublă precizie şi punerea în evidenţă a transmisiei eronate prin diferenţa las nivelul a 2 biţi între sumele de control comparate. O a treia modalitate de formare a sumei de control se bazează pe concatenarea cuvintelor succesive pentru a forma o colecţie de cuvinte de lungime dublă - de exemplu, din k cuvinte de date pe n biţi, se formează cu un set de k/2 cuvinte de date de 2n biţi- şi, corespunzător fiecărei jumătăţi de cuvânt a noii structuri de date, se formează câte o sumă pe n biţi prin adunare binară şi ignorarea de overflow.

Concatenarea celor două sume astfel obţinute (fig.2.35) poartă denumirea de sumă de control Honeywell şi ea prezintă avantajul că o eroare de bit care afectează aceiaşi poziţie binară a tuturor cuvintelor va afecta cel puţin două poziţii binare ale sumei de control. Modul în care este reliefată detecţia defectului a aceleiaşi transmisii eronate din fig.2.33 este ilustrat în fig,2.34, în care se observă că cele două cuvinte de control Honeywell supuse comparării diferă prin 3 biţi.

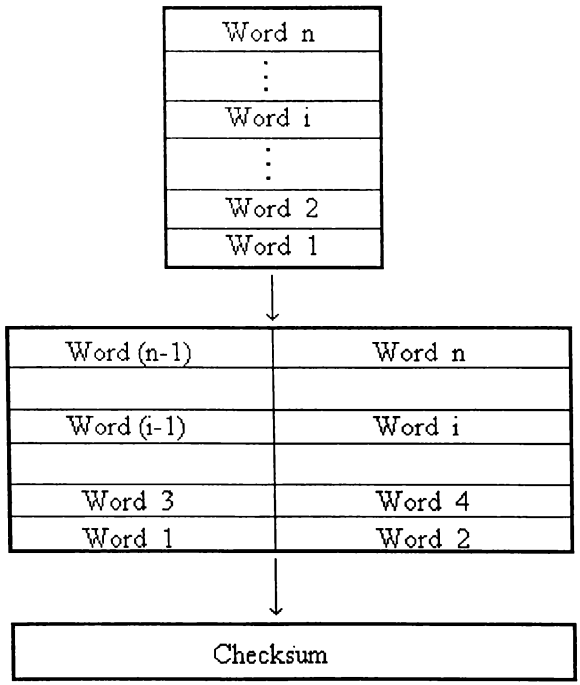


Fig.2.35

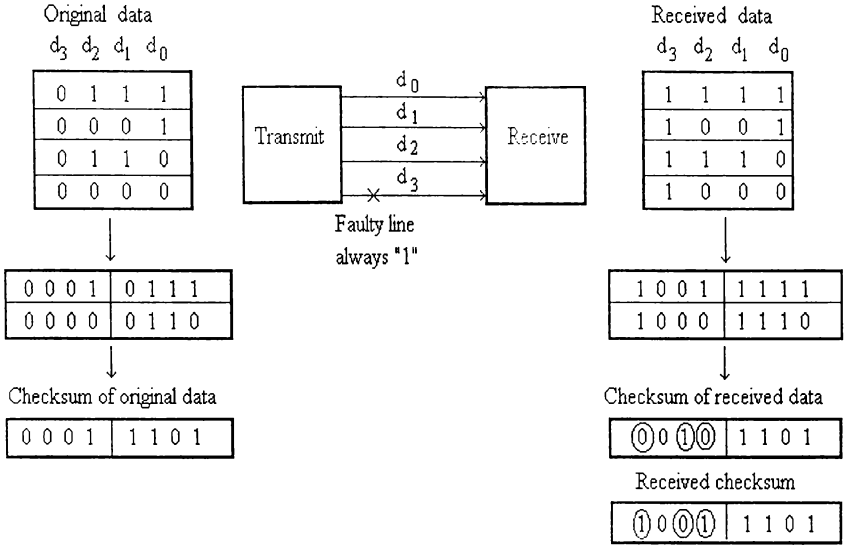


Fig. 2.36

A patra formă de generare a sumei de control se bazează pe principiul formării sumei de control de simplă precizie exceptând faptul că transportul (carry) din poziția binară cea mai semnificativă (depășirea) nu este ignorat, ci el este redundant în manieră end-around carry [Haye-88],conform cu cele ilustrate în fig.2.37..

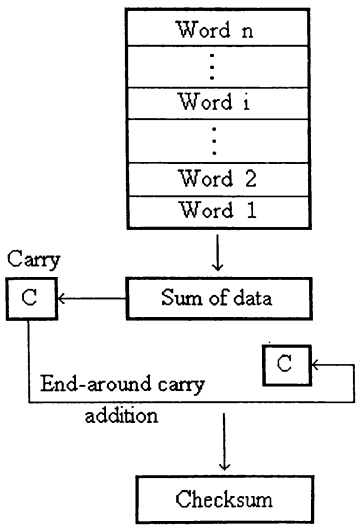


Fig.2.37

Această sumă de control poartă denumirea de reziduală şi modul în care permite reliefarea transmisiei eronate din fig.2.33 este arătat în fig.2.38, unde se observă diferenţe la toţi biţii celor două sume de control cu toate că acestea sunt formate doar pe 4 biţi.

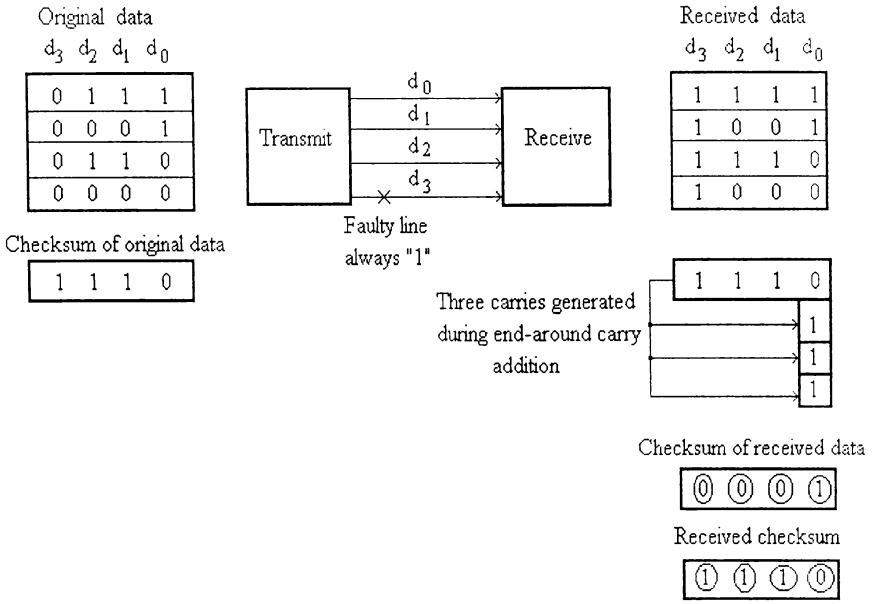


Fig.2.38

2.4.Concluzii

Plecând de la analiza modurilor de defectare specifice sistemelor de calcul, întreprinsă în capitolul anterior, în perspectiva conturării unei aplicaţii de implementare eficientă a toleranţei la defecte într-un sistem dual sincron destinat conducerii proceselor industriale, prezentul capitol îşi propune drept obiectiv trecerea în revistă a acelor metode de implementare importante în contextul prezentei lucrări.În această parte, teza cuprinde următoarele contribuţii:

a)Sistematizarea acelor măsuri de izolare-decuplare a defectelor, în special aplicabile la sisteme duale, cu aport în ceea ce priveşte clasificarea acestor metode.

b)Caracterizarea din punct de vedere al fiabilităţii a metodelor hardware de implementare a toleranţei la defectare bazate pe supravegherea funcţională reciprocă la dublarea statică şi triplarea statică.

c)O originală incursiune în structura sistemelor de calcul în vederea dezvoltării strategiilor de elaborare a produselor program de autotestare bazate pe conceptele de activare modulară, respectiv de verificare a instrucţiilor. Aceste programe vor constitui stimuli pentru procesul de verificare a

cărui evaluare de răspunsuri urmează a fi efectuată prin facilitarea hardware reprezentată de structurile ASIM, care constituie elementul de originalitate esențial al tezei de doctorat.

Se poate conchide că în perspectiva prezentării sistemului dual sincron cu toleranță la defectare din capitolul şase, prezentul material pregăteşte prin primele două paragrafe soluția de redundanță adoptată, iar prin ultimul reprezentat de strategiile conceptuale pentru programele de autotestare, pregăteşte sarcina de stabilire a echipamentului de calcul defect din cadrul sistemului.

3 Structuri ASIM destinate implementării eficiente a toleranței la defectare în sisteme de calcul

3.1. Asupra comprimării paralele multiple a fluxurilor informaționale

O concluzie indubitabilă, care se desprinde din cele prezentate pentru programele de autotestare, independent de conceptul după care au fost structurate şi generate, constă în numărul mare de instrucții de comparare, şi, în consecință, de salt pe care le includ, precum şi în faptul că prin maximizarea acestui număr atât detecția potențialelor malfuncționări, cât şi diagnosticarea defectelor, pot fi facilitate, în sensul creşterii capacității de relevare a anomaliilor funcționale, într-o etapă mai timpurie de parcurgere a programelor. Intercalarea frecvențelor instrucții amintite are însă şi un revers prin creşterea în dimensiune a programelor cu consecințe corespunzătoare în ceea ce priveşte reducerea capacității de trecere a acestora. În scopul îmbunătățirii acestui parametru, deziderat important, cu precădere, pentru sisteme de timp real prevăzute cu capacitate de autotestare pe durata funcționării, se conturează ideea minimizării numărului instrucțiilor de comparare şi, în consecință, şi a companioanelor acestora reprezentate de salturi condiționate sau nu, dar aceasta să fie astfel realizată încât parametrii de testare, constând din capacitatea de detecție şi cea de diagnoză să nu fie alterați. În acest context, expunem în formă sintetică, strategia built-in testing cunoscută sub denumirea Built-In Logic Block Observation, abreviat BILBO [Lala-85, Davi-86, Prad-87, WaCC-87, Gros-89, John-89, ALWI-90]. În conformitate cu aceasta, se propune substituirea în parte, sau chiar în totalitate a registrelor calculatorului -element de structură destinat stocării temporare a informației şi înzestrat, în mod uzual, cu funcții de încărcare şi descărcare- cu structuri BILBO, în fond registre prevăzute cu un număr sporit de funcții, şi anume de deplasare, de generare a unor secvențe binare pseudoaleatoare şi de comprimare a fluxurilor informaționale în baza principiului de analiză de semnături paralelă. Fără a pierde din generalitate, în fig.3.1 prezentăm sinteza unui astfel de registru BILBO pe 8 ranguri binare. Aşa cum rezultă şi din tabelul rezumativ din fig.3.2 al funcțiilor structurii BILBO, atunci când semnalele de control c_1, c_2 şi c_3 sunt la 1, registrul permite încărcarea paralelă convențională, la aplicarea tactului T, a informației de pe liniile D_0 la D_7, aceasta devenind disponibilă la ieşirile Q ale rangurilor R_0 la R_7. Prin modificarea vectorului binar aplicat intrărilor de control (c_1, c_2, c_3) în (0,0,1) (fig.3.2), este inhibată (prin c_1=0) calea de încărcare paralelă şi este deschisă (prin c_2=0) calea de propagare serială între ranguri şi, în fine, este validată (prin c_3=1) încărcarea serială pe la intrarea Scan-In (SI) a unui fluxbinar dorit, cu concomitenta inhibare (prin acelaşi c_3 =1) a căii de reacție a structurii. Condiționarea expusă conferă structurii funcția de registru de deplasare permițând, în primul rând, încărcarea pe la SI, cu un tren de 8 impulsuri de tact, a unui flux informațional de stimulare menit a pune în relief defecte la funcționarea registrului (când configurațiile binare pot consta din deja amintitele şiruri de 0-uri şi 1-uri, alternanțe de
0-uri sau 1-uri care permit "plimbarea" unui 0 în câmp de 1-uri sau a unui 1 în câmp de 0-uri).

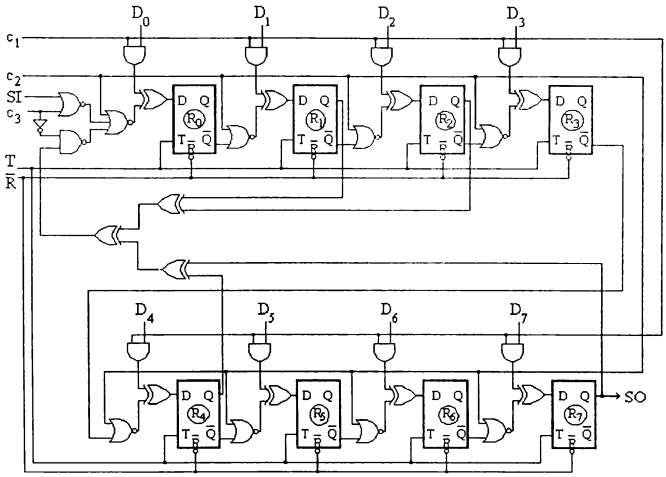


Fig.3.1

În acest caz efectul stimulării este urmărit, prin activarea unui proxim tren de 8 tacte, la ieşirea serială Scan-Out (SO) a structurii.Registrul de deplasare poate fi de asemenea încărcat cu un flux binar care să se constituie în vectori binari, determinişti sau aleatori,care să fie aplicaţi la logica comandată prin ieşirile Q ale registrelor R_0 la R_7 şi al căror efect poate fi captat în paralel, prin primul impuls de tact succesiv celor 8 de încărcare, într-o structură BILBO, similară cu cea prezentată şi conectată la ieşirile logicii combinaţionale stimulate.În al doilea rând, funcţia de registru de deplasare este utilizată pentru evaluarea răspunsurilor când vectorul binar încărcat fie paralel,de pe unul din canalele D0 la D_7, fie serial prin intrarea controlabilă SI, poate fi consultat prin translatarea lui serială cu 8 tacte la punctul de observabilitate SO.

Semnale de control			Funcţia structurii BILBO
c_1	c_2	c_3	
1	1	1	registru conventional
0	0	1	registru de deplasare
0	0	0	generator de secvente pseudoaleatoare
1	0	0	analizor de semnaturi paralel

Fig. 3.2

Celelalte două funcţii ale structurii BILBO necesită comentarii mai ample întrucât literatura de specialitate prezintă, sub acest aspect, de la erori grave la doar, le apreciem, ambiguităţi, din cele consultate nerezultând o recomandare clară a modului în care trebuie condusă sinteza. Astfel, pentru acoperirea funcţiilor de generare a unor secvenţe pseudoaleatoare, respectiv de analiză de semnături paralelă, cu certitudine structura trebuie configurată într-o schemă secvenţială liniară prin conectarea

pe lanțul de reacție a unui arbore de circuite logice SAU EXCLUSIV (EX-OR tree). Problema constă în alegerea acelor puncte din structură care să reprezinte intrări pentru EX-OR tree și, apoi, în modalitatea de sinteză a acestui arbore astfel încât, pe de o parte, să fie asigurati parametri, de testare superiori, și, pe de altă parte, performanța funcțiilor convenționale să fie cât mai puțin afectată prin adausul de circuistică revendicat de noile funcții de built -in testing. Dacă analizăm, sub acest aspect, referința [John-89, pp.544-552] remarcăm, pe de o parte, erorile grave constând din lipsa referirii într-o formă la semnalul de control c_3 și din conectarea ieșirilor Q ale rangurilor R_0 la R_7 la circuitele SAU-NU [John-89, pp.546-Fig.7.57, pp.550-Fig.7.60] și nu a ieșirilor $\overline{Q}$ așa cum se prezintă în fig.3.1. Această ultimă remarcă are drept consecință negarea absolut nejustificată a fluxurilor binare atunci când acestea sunt translate serial la SO, implicând o prelucrare suplimentară la elaborarea dicționarelor răspuns. În al doilea rând, sinteza EX-OR tree [John-89, pp.546-Fig.7.57, pp. 548-Fig. 5.48, pp. 550-Fig. 7.60] este neoptimizată în sensul că circuitele SAU EXCLUSIV sunt înlănțuite serial (și nu arborescent ca în fig.3.11), prin aceasta adăugându-se, la același număr de intrări și circuite logice ca în fig.3.1, un nivel logic suplimentar, care prin întârzierea pe care o provoacă la propagarea semnalelor, determină expandarea perioadei trenului de tact și corespunzător reduce frecvența de funcționare maxim admisă. Această observație câștigă în pondere cu creșterea numărului de ranguri, spre exemplu la 16, 32 sau 64, când pentru păstrarea unor indicatori de performanță a testării cu valori ridicate, numărul circuitelor logice crește și cu atât mai mult pierde din importanță afirmația din [John-89, pp.549,pp.552] cum că viteza de operare a structurii ar fi afectată de nivelurile logice SAU-NU și SAU EXCLUSIV dintre ranguri. În al treilea rând, nu se face nici o precizare referitoare la punctele din schemă care să constituie intrări pentru EX-OR tree. Principial, aceleași observații sunt valabile pentru [Wojt-88, pp.151-Bild 6.42, pp.152-Bild 6.43, pp.153-Bild.6.46], [RKLC-25,pp.113-Fig.8.7], [Lala-85, pp.228-Fig.6.39, pp.231-Fig.6.41], care, în majoritate, constituie cursuri universitare, dar ambiguitatea apare și în articole din reviste cu cenzură științifică severă, cum ar fi , spre exemplu [HoLC-90, pp.1275-Fig.3]. Există surse de literatură la care, prin faptul că exemplifică strategia BILBO pe doar 4 ranguri, aspectele critice subliniate nu sunt relevante. Menționăm în acest sens [Gork-91, pp.5.4/7-Beispiel 5.14], în care schema pe 4 ranguri nu este prevăzută cu funcția de generare de secvențe pseudoaleatoare (ceea ce permite economisirea unui semnal de control, rolul lui c_3 din fig.3.1 fiind preluat de c_1), dar se face referire, în mod incorect, la sinteza schemei secvențiale liniare având la bază o expresie primitivă de polinom generator dar luând intrările pentru, în acest caz, unicul circuit SAU EXCLUSIV de pe lanțul de reacție de la ieșiri de bistabile, cum este prezentat în fig.3.1. Ar mai fi de menționat [Prad-86, pp.150, Fig.2.6.15] cu aceleași observații ca și la [Gork-91], cu considerarea intrărilor la unicul circuit SAU EXCLUSIV din lanțul de reacție tot de la ieșirea bistabilelor, dar nu se specifică, în mod explicit, utilizarea la sinteza schemei secvențiale a unei expresii primitive de polinom generator cu toate că problematica analizei de semnături paralele este prezentată în prelungirea celei seriale unde este subliniată utilizarea unui polinom primitiv. Se poate conchide că articolul mamă B.Konemann,

J.Mucha, G.Zwiehoff: "Built-In Logic Block Observation Technique" Dig.1979, IEEE Test Conf, Cherry Hill, pp.37-41, Oct.23-25,1979, la care se face, direct sau indirect, referire deschide calea spre interpretări ambigue prin precizarea insuficientă a problematicii.

3.2.Stabilirea parametrilor de sinteză pentru structuri ASIM

Plecând de la structura din fig.3.1, vom încerca să aducem precizările necesare legate de sinteza schemei. Astfel, trecând la funcţia de generare de secvenţe pseudoaleatoare prin condiţionarea semnalelor, de control (c_1, c_2, c_3) cu vectorul binar (0,0,0), se inhibă căile de încărcare paralelă (cu c_1=0) şi serială (cu c_3=0), rămânând deschisă (cu c_2=0) calea de deplasare şi reacţie, schema devenind secvenţial liniară. Admiţând că în rangurile registrului de deplasare este stocată iniţial o combinaţie binară diferită de 0 (peste tot), este cunoscut [RaFu-89] că secvenţele binare, culese în paralel sau serial la fiecare rang în parte, prezintă propietatea verificabilă prin criterii statistice -de a fi aleatoare şi, întrucât se repetă după un număr determinat de tacte, poartă denumirea de pseudoaleatoare. Dacă registrul de deplasare pe care îl include schema secvenţială liniară are, în caz general, n ranguri binare şi dacă la baza sintezei acesteia se adoptă o expresie primitivă, implicit ireductibilă, de polinom generator, atunci schema furnizează şirul de lungimea maximă de (2^n - 1) cuvinte de n biţi, care se repetă în mod ciclic cu periodicitatea de (2^n - 1) impulsuri de tact. Pe de altă parte, este, de asemenea, cunoscut [Prad-86,Wojt-88, Gork-89] că testarea aleatoare prezintă un grad de acoperire al defectelor cu atât mai avansat cu cât secvenţa de stimulare este mai lungă. Corelând cele expuse, rezultă faptul că la sinteza schemei secvenţiale liniare trebuie utilizată o expresie primitivă, implicit ireductibilă, de polinom generator. Imediat legat de această primă concluzie, se pune problema alegerii acestei expresii fiind cunoscut faptul că, şi atunci când numărul de ranguri n al registrului de deplasare prezintă valori moderate, familia de polinoame care corespund gradului n şi prezintă proprietatea de a fi primitive este excesiv de mare. Astfel, apelând la metode de determinare sau, mai degrabă, consultând biblioteci matematice de calculator sau anexe ale unor surse bibliografice consacrate de teoria codurilor [PeWa-72] cuprinzând tabele cu expresiile polinoamelor generatoare primitive în dependenţă de numărul n al rangurilor registrului de deplasare, se poate constata că pentru n=16 rezultă 2048 de polinoame primitive. Introducând în acest moment al analizei parametrul de performanţă constituit de frecvenţa trenului de tact care asigură generarea secvenţelor binare pseudoaleatoare, în vederea maximizării acestui parametru se impune ca întârzierea pe circuitele logice amplasate pe lanţul de reacţie al schemei secvenţiale liniare să fie minimă. Evident, la întârzierea menţionată se adaugă şi cea de pe liniile de cablaj, dar considerând implementarea tehnologică caracterizată prin avansată densitate de împachetare implicând lungimi de cablaje reduse, acest parametru dinamic de proiectare poate fi, într-o primă analiză, neglijat în raport cu timpul de propagare corespunzător porţilor logice chiar şi atunci când acestea sunt realizate într-o tehnologie bipolară rapidă de tipul TTL-Schottky, LS sau ECL. Urmărind acest parametru dominant şi apelând la o sinteză arborescentă caracterizată, pentru un număr de m intrări de ale EX-OR tree, prin , în cazul cel mai defavorabil, numărul ($\lceil \log_2 m \rceil$) de niveluri de porţi logice SAU EXCLUSIV- unde barele $\lceil\ \rceil$ semnifică cel mai mic număr întreg mai mare decât valoarea rezultată

pentru logaritm-, rezultă că din subfamiliile de polinoame primitive prezintă interes, sub aspectul menționat, doar acelea caracterizate printr-un număr, admitem m, minim de termeni. Procedând în consecință pentru anterior menționatul n=16, numărul celor 2048 polinoame- se reduce-corespunzător valorii minime a numărului de termeni care, în acest caz, rezultă m=5-drastic la 13. Opțiunea proiectantului pentru o anumită expresie din acest set mult diminuat poate fi una arbitrară dar dacă luăm în considerare avertizarea obținută pe baza de simulare logică [WiNG-90], atunci alegerea trebuie în continuare condusă cu precauție. Problema supusă analizei în [AhNG-90] este una fundamentală pentru toate tehnicile de testare bazate pe comprimarea răspunsurilor unității testate în semnături şi anume de mascare a unui număr din potențialele defecte, aşa numită aliasing, nu de către caracteristici ale unității testate, ci de specificul metodei de comprimare în sensul că unitatea testată cu funcționare normală şi cea defectă prezintă semnături identice. Prin faptul că şi în [PrGK-90], într-o abordare formală, se avertizează, în contextul probabilității de aliasing specifică analizoarelor de semnături cu intrări multiple pentru testarea acelor scheme ramificate a căror logică poate comanda mai multe ieşiri (sharing of logic between outputs), vom insista cu descrierea rezultatelor expuse în [AhNG-90]. Experimentul de simulare este prezentat într-un context de testare, nu al unei structuri BILBO, urmărind studiul comportării raportat la sensibila problemă de aliasing a schemelor secvențiale liniare sintetizate, în scopul generării semnăturilor, atât cu polinoame primitive, cât şi neprimitive. Modelul de testare este prezentat în fig.3.3 şi se observă utilizarea unei analizor de semnături serial prevăzut cu o bancă de circuite EX-OR ale cărei intrări c_1, c_2,.....,c_m se modifică pe parcursul procesului de simulare.În calitate de circuite testate au fost simulate, utilizând descrieri la nivel de poartă, schemele corespunzătoare unor structuri logice încapsulate în circuite integrate pe scară medie (74LS139, 74LS145, SN7482, SN74H87, 74LS138 şi 74LS352) a căror complexitate internă este cuprinsă între 9 şi 21 de porți. Procedura algoritmică utilizată pentru generarea numărului de aliases corespunzător unei anumite scheme testate este prezentată în fig.3.4.

De asemenea, a fost considerat cazul în care simularea schemei verificate s-a efectuat prin baleierea exhaustivă (incluzând vectorul binar constând din 0 la toate intrările) a combinațiilor binare generate de către un numărător de n biți, situație în care a fost generat numărul de aliases pentru fiecare polinom de ordin n utilizat la evaluarea semnăturilor. Rezultatele obținute pentru circuitul 74LS145 (BCD-to-decimal decoder, 4 inputs, 10 outputs, 18 gates) în care au fost inserate 180 defecte sunt sintetizate în tabelul din fig.3.5, dar în [Ah.NG-90] se afirmă că rezultate similare au fost obținute şi pentru celelalte circuite.

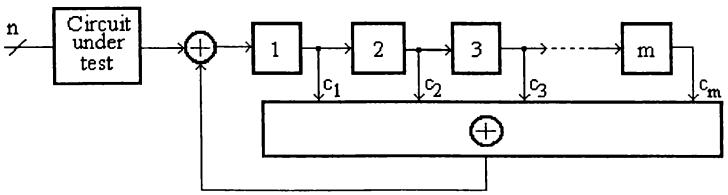


Fig.3.3

Numerele minime de aliases din fiecare coloană au fost însemnate cu un singur asterisc, iar numerele maxime au fost marcate prin două asteriscuri.

```
For an n-input circuit under test
    Choose an n-order primitive polynomial T_i for test-pattern generation
     {i=1 to NPP; where NPP is the total number of primitive polynomials
                                                    of order n};
    Set polynomial seed as 0,0,...,1;
    Generate the sequence of length 2^n-1;
    Choose an n-order polynomial S_j for signature analysis
     {j=1 to NP; where NP is the total number of polynomials of order n};
    Choose circuit response R_k
     {k=0 to 2NL; where NL is the total number of lines in the circuit; R_0 is
      fault-free, R_1 has fault 1 inserted, R_2 has fault 2 inserted, etc.};
        Compute signature r_m
        Check aliasing and, if necessary, update aliasing count;
        End;
    Print aliasing count for (T_i , S_j )
    End;
End
```

Fig.3.4

Polynomials used in signature analysis	Test - Pattern Generator		
	Linear maximal sequence generator polynomials		Ordinary binary counter (including all - 0s input)
	$1+x^3+x^4$	$1+x+x^4$	
$1+x^4$	19	20	30
$1+x^3+x^4$	21	74**	8*
$1+x^2+x^4$	65	18	19
$1+x^2+x^3+x^4$	25	19	13
$1+x+x^4$	74**	22	8*
$1+x+x^3+x^4$	62	32	35**
$1+x+x^2+x^4$	16	11*	23
$1+x+x^2+x^3+x^4$	11*	11*	11

Fig. 3.5

Pe baza rezultatelor obținute prin simulare logică, autorii conchid: (1) Pentru utilizarea în calitate de stimulator a unui generator de secvențe pseudoaleatoare de lungime maximă, comprimarea prin analiză de semnături bazată pe polinoame primitive nu conduce întotdeauna la un număr mai mic de erori aliasing în raport cu comprimarea bazată pe polinoame neprimitive. (2) Unele polinoame primitive de analiză de semnături pot determina numere maxime de aliases. (3) Numărul maxim de aliases apare atunci când polinoamele utilizate la generarea de modele de testare şi la analiza de semnături sunt reciproce. (4) Rezultatele obținute prin inserția de defecte de blocare singulare pot fi extinse şi în raport cu defectele de blocare multiple. (5) Se pare că tehnica de testare cea mai bună constă în utilizarea unui numărător binar pentru generarea de modele de testare şi a unui analizor de semnături bazat pe un polinom primitiv pentru evaluarea răspunsurilor. Pronunțându-ne critic asupra celor cuprinse în [AhNG-90] relevăm inconsistența experimentului de

simulare limitat la scheme de complexitate totuşi redusă-cu privire la numărul de porţi, cât şi celui de intrări (maxim 7)-, apoi la inserţia doar a defectelor de blocare şi, în ultimă instanţă la utilizarea schemei seriale de analiză de semnături. Ignorarea altor modele de defectare decât cel de blocare, cum ar fi scurtcircuitele, stuck-at open [Vasi-93],şi defectele de funcţionare dinamică, de tip bus skew [PrHe-90,Prad-86], precum şi comprimarea paralelă a răspunsurilor unor scheme cu ramificaţii multiple ar putea influenţa concluziile de mai sus. Revenind la problematica generării de secvenţe pseudoaleatoare pentru o structură BILBO, reţinem dintre concluziile de mai sus cu precădere recomandările (3) şi (2), atunci când vom face conexiunea şi cu cea de-a patra funcţie, anume cea de analiză de semnături paralelă. Este de remarcat că la sinteza din fig.3.1 s-a utilizat polinomul primitiv de grad 8 cu expresia următorul polinom primitiv:

$$G(x) = x^8 + x^6 + x^5 + x^3 + 1 \qquad (3.1)$$

Trecând la funcţia de analiză de semnături paralelă a structurii din fig.3.1, vectorul binar (1,0,0) corespunzător semnalelor de control (c_1, c_2, c_3) deschide canalele D_0 la D_7 (prin c_1=1) de captare a fluxurilor informaţionale, inhibând calea de încărcare serială (prin c_3=0) şi lăsând deschisă calea de deplasare internă (prin c_2=0), astfel încât semnăturile să poată fi formate de către schema secvenţială liniară având circuitele logice SAU EXCLUSIV conectate în exteriorul registrului de deplasare.

Funcţia de analiză de semnături cu intrări multiple determină - prin gradul, şi expresia polinomului generator-, în mod decisiv, sinteza unei structuri BILBO de tipul celei din fig. 3.1. Sub acest aspect, făcând abstracţie de caracteristicile schemei testate, alegerea unui polinom primitiv cu număr minim de termeni se prezintă ca o soluţie favorabilă prin prisma triplului impact constituit de cost (exprimat de număr de circuite elementare, respectiv arie de substrat de siliciu)/performanţă (exprimată prin frecvenţa procesului de comprimare)/capacitate de detecţie (exprimat prin, anticipări, probabilitatea de recunoaştere ca funcţional corectă a unei unităţi testate defecte). De fapt, acest ultim parametru al impactului determină, aşa cum va rezulta din următorul paragraf, modificări esenţiale de legături în structura BILBO, ceea ce ne-a determinat să adoptăm abrevierea ASIM (Analizor de Semnături cu Intrări Multiple) pentru structurile analizate.

3.3. Capacitatea de detecţie la comprimarea prin structuri ASIM

Din teoria analizei de semnături seriale [Vlăd-82, Fuji-85, Prad-86, GrJK-90, Sosn-91, UpRa-94], se cunoaşte că, prin prisma capacităţii de detecţie, sunt echivalente unele scheme secvenţiale liniare care vor fi analizate, în noul context al captării paralele a lanţurilor binare răspuns, în cele ce urmează. În calitate de măsură pentru capacitatea de detecţie apelăm la probabilitatea statistică de recunoaştere ca funcţional corectă a unei unităţi testate defecte dată de raportul dintre numărul total al erorilor nedetectate N_{TN} şi numărul total al erorilor posibile N_T, în fond probabilitatea de aliasing.

3.3.1. Structură ASIM cu circuite SAU EXCLUSIV intercalate suplimentar între rangurile registrului de deplasare

Pentru a caracteriza o structură ASIM cu circuite SAU EXCLUSIV intercalate între rangurile registrului de deplasare prin prisma capacității de detecție a potențialelor defecte, introducem următoarea teoremă originală:

Teorema 3.1: Fiind dată o expresie de polinom generator G(x), primitiv sau neprimitiv de grad n, cu mai mult de un termen, şi fiind asociate polinoamele Pp(x), p=0, 1, 2,..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM, sintetizată pe baza polinomului G(x) cu circuite logice SAU EXCLUSIV conectate suplimentar în interiorul registrului de deplasare, generează semnături identice cu cele furnizate de către un analizor de semnături serial de acelaşi tip şi sintetizat pe baza aceluiaşi polinom generator G(x), dar încărcat cu fluxul binar având asociat polinomul:

$$\mathbf{P}(x)= \sum_{p=0}^{n-1}{}^{\oplus} x^{p}P(x), \text{ unde } \sum^{\oplus} \text{ semnifică suma modulo 2.}$$

Demonstrație: Fără a pierde din generalitate, vom admite pentru polinomul generator G(x) următoarea expresie:

$$G(x)=x^{n}+x^{k}+x^{j}+x^{i}+1 \qquad (3.2.)$$

Structura ASIM având la baza sintezei expresia (3.2) şi circuitele SAU EXCLUSIV conectate între rangurile registrului de deplasare este prezentată în fig. 3.6. Se impune o precizare legată de denumire în sensul că o structură ASIM are, independent de conexiunile reacției la circuitele SAU EXCLUSIV, între rangurile registrului de deplasare cel puțin un astfel de circuit SAU EXCLUSIV. Se poate remarca însă în reprezentarea schematică din fig. 3.6 că pentru termenii expresiei (3.2)- excepție făcând, în mod evident, termenul cu gradul cel mai mare- sunt prevăzute între rangurile corespondente ale registrului de deplasare scheme SAU EXCLUSIV cu 3 intrări a căror sinteză revendică două niveluri de circuite. Între restul rangurilor, corespunzătoare absenței termenilor, este prevăzut un singur circuit SAU EXCLUSIV. Denumirea adoptată pentru structura din fig. 3.6 face referire la circuitele SAU EXCLUSIV intercalate suplimentar pentru unul din nivelurile schemelor corespunzătoare termenilor specificați mai sus din expresia polinomului generator G(x).

În cadrul demonstrației ne vom uzita de proprietatea de liniaritate specifică unei stucturi de tipul celei prezentate în fig.3.6 şi vom apela la principiul superpoziției.

Poate fi remarcat faptul că, făcînd Pp(x)=0, pentru p=1, 2,..., i,..., j,..., k, n-2, n-1, schema din fig. 3.6 se transformă în convenționalul analizor de semnături serial cu circuite SAU EXCLUSIV interconectate între rangurile RD_p ale registrului de deplasare.

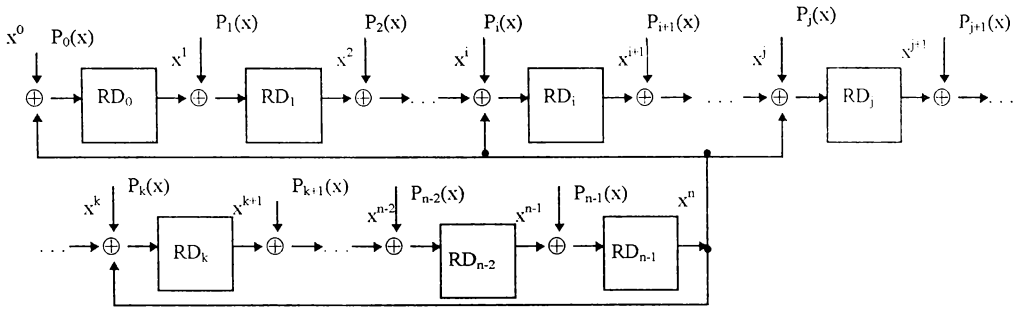


Fig. 3.6

În acest caz, circuitele SAU EXCLUSIV dintre rangurile registrului de deplasare dispar, excepție făcând, desigur, cele care preced rangurile RD_0 , RD_i , RD_j și Rd_k. Pentru o astfel de schemă este cunoscută [Vlăd-89] corelația dintre polinomul $P_0(x)$, asociat lanțului binar și semnătura generată, care admitem că are atașat polinomul $S_0(x)$, și anume:

$$P_0(x)=Q_0(x)G(x)+S_0(x), \tag{3.3}$$

în care $Q_0(x)$ reprezintă polinomul asociat lanțului binar care părăsește schema, atunci când $r > n$, la ieșirea rangului RD_{n-1} și în care operațiile prevăzute sunt executate în câmpul Galois(Galois Field-GF) cu două elemente GF(2).

Facem acum $P_p(x)=0$, pentru p=0, 2, 3, ..., i,...j, ..., k, ...,n-1 și încărcăm în schema secvențială liniară fluxul binar care are asociat polinomul $P_1(x)$ prin circuitul SAU EXCLUSIV conectat la ieșirea rangului RD_0 (fig. 3.6). Din punct de vedere temporal, această încărcare echivalează cu cea efectuată în circuitul SAU EXCLUSIV conectat pe intrarea rangului RD_0 - în fond, intrarea analizorului de semnături serială, dar care se efectuează cu un impuls de tact în avans. Din punct de vedere formal, echivalența menționată se traduce prin captarea în analizorul de semnături serial a secvenței binare având asociat polinomul $xP_1(x)$. Dacă în același analizor am capta secvența binară cu P(x), admitem că am obține polinoamele $Q_1(x)$, respectiv $S_1(x)$, pentru lanțul binar care părăsește schema la ieșirea rangului RD_{n-1} , respectiv pentru semnătură, adică informația care rămâne în registrul de deplasare. Prin analogie cu (3.3), luînd în considerare operația de împărțire efectuată în GF(2) de către structura din fig.3.6 atunci când p=0, 2, 3, ..., i, ..., j, ..., k, ..., n-1, avem:

$$xP_1(x)=xQ_1(x)+xS_1(x) \tag{3.4}$$

Raționamentul expus pentru polinomul $P_1(x)$ poate fi aplicat, în manieră asemănătoare, cu corespunzătoare luare în considerare a decalajului de impulsuri de tact cu care se efectuează încărcarea, ceea ce, din punct de vedere formal, se adaptează prin înmulțirea identității împărțirii

efectuate în GF(2) pentru analizorul de semnături serial cu x ridicat la puterea egală cu numărul tactelor de decalaj. Astfel, trecând la ultimul rang şi corelând notaţiile, avem:

$$x^{n-1} P_{n-1}(x)=x^{n-1} Q_{n-1}(x)G(x)+ S_{n-1}(x) \qquad (3.5)$$

Cumulând (3.3) la (3.5) şi aplicând principiul superpoziţiei, pentru schema din fig. 3.6, rezultă:

$$\begin{aligned}
&x^0 P_0(x)=x^0 Q_0(x)G(x)+x^0 S_0(x)\\
&x^1 P_1(x)=x^1 Q_1(x)G(x)+x^1 S_1(x)\\
&\text{----------}\\
&x^i P_i(x)=x^i Q_i(x)G(x)+x^i S_i(x)\\
&\text{----------}\\
&x^j P_j(x)=x^j Q_j(x)G(x)+x^j S_j(x) \qquad (3.6)\\
&\text{----------}\\
&x^k P_k(x)=x^k Q_k(x)G(x)+x^k S_k(x)\\
&\text{----------}\\
&x^{n-1} P_{n-1}(x)=x^{n-1} Q_{n-1}(x)G(x)+x^{n-1} S_{n-1}(x)
\end{aligned}$$

$$P(x)=\sum_{p=0}^{n-1}{}^{\oplus}\ x^p P_p(x)= G(x)\sum_{p=0}^{n-1}{}^{\oplus}\ x^p Q_p(x) + \sum_{p=0}^{n-1}{}^{\oplus}\ x^p S_p(x)$$

Prin urmare, structura ASIM din fig. 3.6, a cărei descriere funcţională este redată formal prin ecuaţiile (3.6), este echivalentă cu analizorul de semnături serial reprezentat în fig.3.7.

În baza celor cunoscute, pentru structura din fig. 3.7 avem:

$$P(x)=Q(x)G(x)+S(x), \qquad (3.7)$$

unde, din nou, Q(x) şi S(x) reprezintă polinoame asociate lanţurilor binare care părăsesc schema, pe la ieşirea lui RD_{n-1} -, respectiv rămân în rangurile registrului de deplasare.

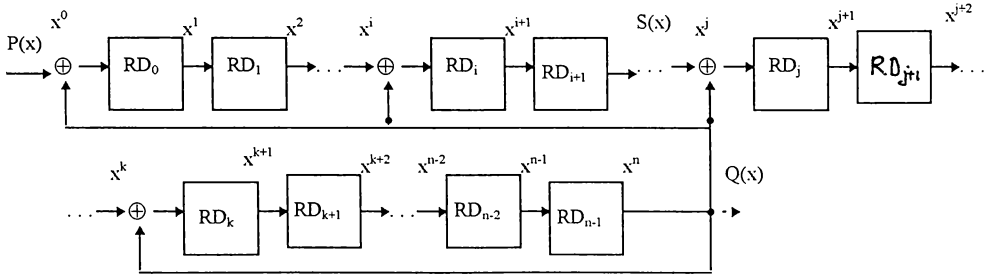


Fig. 3.7

Toate polinoamele $S_p(x)$, din relația (3.6) unde p=0, 1, ..., i, ..., j, ..., k, ..., ...,n-1, sunt de gradul (n-1), astfel încît, pentru obținerea semnăturii furnizate de cele 2 structuri, având asociat polinomul S(x), conform cu cele din [Vlăd-82], avem:

$$
\begin{aligned}
&x^0\ S_0(x)=S_0'(x)G(x)+S_0''(x)\\
&x^1\ S_1(x)=S_1'(x)G(x)+S_1''(x)\\
&\text{----------}\\
&x^i\ S_i(x)=S_i'(x)G(x)+S_i''(x)\\
&\text{----------}\\
&x^j\ S_j(x)=S_j'(x)G(x)+S_j''(x)\\
&\text{----------}\\
&x^k\ S_k(x)=S_k'(x)G(x)+S_k''(x)\\
&\text{----------}\\
&x^{n-1}\ S_{n-1}(x)=S_{n-1}'(x)G(x)+S_{n-1}''(x)\\
&\text{\_\_\_\_\_}\\
&\sum_{p=0}^{n-1}{}^{\odot}\ x^p\ S_p(x)=G(x)\sum_{p=0}^{n-1}{}^{\oplus}\ S_p'(x)+\sum_{p=0}^{n-1}{}^{\oplus}\ S_p''(x)
\end{aligned}
\qquad (3.8)
$$

în care polinoamele $S_p'(x)$ sunt de grad p și corespund lanțurilor binare care părăsesc structura din fig. 3.6, iar polinoamele $S_p''(x)$ sunt de grad (n-1) și corespund lanțurilor binare care, prin operare SAU EXCLUSIV, permit formarea semnăturii generate de aceeași structură.

Corelând cele din (3.6) la (3.8), prin aplicarea principiului superpoziției, obținem:

$$P(x)=\underbrace{\sum_{p=0}^{n-1}{}^{\oplus}\left(x^{p}\cdot Q_p(x)+S_p(x)\right)}_{Q(x)}+\underbrace{\sum_{p=0}^{n-1}{}^{\oplus}S_p''(x)}_{S(x)} \qquad (3.9)$$

Se poate deci conchide că structura din fig. 3.6 este identică din punct de vedere funcțional cu n analizoare de semnături seriale care, prin suprapunere, dau structura din fig. 3.7. Am introdus teorema 3.1 pentru că ea ne permite caracterizarea structurii ASIM din fig. 3.6 prin prisma capacității de detecție a erorilor provocate de defectele analizate în capitolul 1, întrucât, este cunoscută [Gork-91, John-90, Vlăd-89, Wojt-88, Prad.86, Vlăd.82], probabilitatea de recunoaștere ca funcțional corectă a unei unități testate defecte P corespunzătoare unei structuri de tipul celei din fig. 3.7, și anume :

$$P=\frac{N_{TN}}{N_T}=\frac{\left(2^{r-n}\right)}{\left(2^{r}-1\right)}\cong 2^{-n} \qquad (3.10)$$

în care r reprezintă lungimea lațului de biți de comprimat, iar n reprezintă lungimea semnăturii.

Înainte însă de a trece la obiectivul menționat, vom insista asupra unor proprietăți ale structurii ASIM din fig. 3.6 deosebit de utile din punct de vedere al evaluării analitice a semnăturilor etalon, deci cele care corespund funcționării normale, fără defecte. Este adevărat că, din punct de vedere practic, ele pierd din semnificații plecând de la faptul că, în mod uzual, semnăturile etalon se culeg de la realizări martor identice constructiv cu unitățile testate.

Proprietățile în discuție decurg din asocierea la analizoare de semnături în general, prin analogie cu schemele de generare a biților redundanți la coduri ciclice detectoare şi corectoare de erori [Vlăd-89, SpSw-91, RaFu-89, NeSc-92], a matricilor de control H. În vederea elaborării acestora, plecând de la o expresie dată, pe care o admitem, fără a pierde din generalitate, primitivă şi ireductibilă, pentru polinomul generator G(x), propunem următoarea procedură care să stea la baza soluționării asistate de calculator a problemei.

Procedura 3.1: Pentru simplificarea descrierii, admitem că expresia (3.2) constituie un polinom generator primitiv asociat construcției analizorului de semnături serial din fig. 3.7. Din punct de vedere temporal, de termenii expresiei G(x)- corespunzător conexiunilor de pe lanțul de reacție- se ține cont prin următoarele ecuații logice:

$$\begin{aligned} RD_0(t+1)&=RD_{n-1}(t)\\ RD_i(t+1)&=RD_{n-1}(t)\oplus RD_{i-1}(t)\\ RD_j(t+1)&=RD_{n-1}(t)\oplus RD_{j-1}(t)\\ RD_k(t+1)&=RD_{n-1}(t)\oplus RD_{k-1}(t) \end{aligned} \qquad (3.11)$$

unde, spre exemplu, RD_i (t+1) reprezintă conținutul rangului RD_i al registrului de deplasare în urma aplicării celui de al (t+1lea impuls de tact, prin care sunt operate SAU EXCLUSIV $\oplus$ conținuturile, existente la momentul de timp t, anterior aplicării respectivului impuls de tact, rangurilor RD_{n-1} şi RD_{i-1}.

În general, având o expresie de polinom generator G(x) cu t termeni, rezultă (t-1) ecuații de tipul celor date de (3.11). Plecând de la aceste ecuații, prin aplicarea lor la construcția fiecărei coloane, în mod secvențial- coloană cu coloană- se elaborează matricea H.

Pentru exemplificare, vom considera polinomul primitiv G(x) cu expresia:

$$G(x)=x^4+x+1 \qquad (3.12)$$

Pe baza relației (3.12) rezultă sinteza analizorului de semnături serial, care, schematic, este prezentată în fig.3.8:

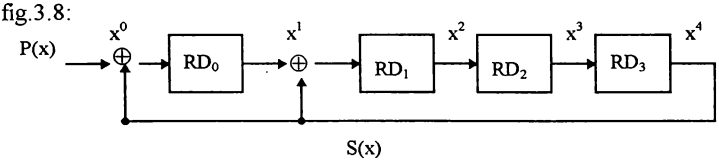


Fig. 3.8.

Cele (3 termeni-1=) 2 ecuații logice, care iau în considerare factorul timp, specifice structurii din fig. 3.8 sunt următoarele:

$$\begin{aligned} RD_0(t+1)&=RD_3(t)\\ RD_1(t+1)&=RD_3(t)\oplus RD_0(t) \end{aligned} \qquad (3.13)$$

Luând în considerare (3.13) şi cele anterior expuse, construcţia matricii H asociată structurii din fig. 3.8 este sugerată, plecând de la 'sămânţa' cu o unitate binară în rangul RD_0 şi cu zerouri în RD_1 la RD_3 , în fig. 3.9. Se observă că fiecare linie a matricii de control H corespunde unui rang a registrului de deplasare, iar acestora li s-au alocat ponderi egale cu 2 ridicat la puterea dată de indicele rangului.

$$
H = \begin{array}{c}
\\
(2^0)\ RD_0 \\
(2^1)\ RD_1 \\
(2^2)\ RD_2 \\
(2^3)\ RD_3
\end{array}
\begin{bmatrix}
1 & & 0\ 0\ 0 & & 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 & & 1 & & 1 \\
0 & \oplus\rightarrow & 1\ 0\ 0 & \oplus\rightarrow & 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0 & \oplus\rightarrow & 0 & \oplus\rightarrow & 0 \\
0 & & 0\ 1\ 0 & & 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1 & & 0 & & 0 \\
0 & & 0\ 0\ 1 & & 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1 & & 1 & & 0
\end{bmatrix}
$$

$$
\begin{array}{cccc}
c_0 & c_1\ c_2\ c_3 & c_4\ c_5\ c_6\ c_7\ c_8\ c_9\ c_{10}c_{11}\ c_{12}\ c_{13} & c_{14} \quad c_0 \\
(1) & (2)\ (4)\ (8) & (3)\ (6)\ (12)(11)\ (9)\ (10)\ (7)(14)(15)(13) & (9)
\end{array}
$$

Fig. 3.9

În baza acestor ponderi, echivalenţii zecimali calculaţi pentru fiecare din cele 15 coloane c_i (unde i=0, 1, ..., 14) sunt daţi sub fiecare dintre coloane. Pentru că polinomul generator G(x) a fost ales unul primitiv, matricea H conţine numărul maxim de coloane egal cu 2^n-1, unde n reprezintă gradul lui G(x) (în particular pentru (3.12) avem n=4, deci 2^4-1=15 coloane în fig. 3.9). După cele 15 coloane cu vectori binari distincţi, aşa cum rezultă din fig. 3.9, în mod ciclic, se revine la c_0. Aşa cum este fundamentat în teoria codurilor, dacă G(x) nu este primitiv, numărul maxim de vectori distincţi de coloană se reduce în mod corespunzător dependent de expresia lui G(x). Maniera de construcţie a matricii H, ilustrată în fig. 3.9, pune în relief modalitatea facilă de soluţionare a acesteia asistată de calculator.

Odată stabilită matricea H şi fiind cunoscută lungimea ferestrei de comprimare prin numărul impulsurilor de tact, se stabilesc termenii polinomului P(x) asociat lanţului binar de evaluat şi se operează SAU EXCLUSIV, în secvenţă, vectorii binari ai coloanelor corespunzătoare unităţilor binare din secvenţa de comprimat. Astfel, în calitate de exemplu, admitem că dorim evaluată, prin analizorul din fig. 3.8, semnătura generată în fereastra de 32 impulsuri de tact pentru secvenţa binară descrisă de polinomul P(x) cu expresia următoare:

$$P(x)=x^{31} + x^{26} + x^{22} + x^{11} + x^8 + x^4 + 1 \qquad (3.14)$$

În baza celor descrise şi ţinând cont de ciclicitatea de 15 specifică matricii H din fig.3.9, bitului cel mai semnificativ al secvenţei descrisă prin (3.14) îi corespunde coloana c_1 cu echivalentul zecimal 2. Celui de-al doilea bit de 1, căruia îi corespunde în (3.14) termenul x^{26}, îi corespunde coloana c_{11} cu echivalentul zecimal 14. Operând SAU EXCLUSIV echivalenţii zecimali 2 şi 14 se obţine semnătura intermediară cu echivalentul zecimal 12. Acesta va fi operat în aceiaşi manieră cu echivalentul 11 al coloanei c_7 corespunzătoare lui x^{22}, obţinându-se intermediar valoarea 7, care

operată cu 14 (corespunzătoare lui x^{11}) dă 9. În mod similar, se obțin în continuare semnăturile intermediare cu echivalenții zecimali 12 şi 15, iar, în final, rezultă semnătura cu echivalentul zecimal 14. Se menționează că aceeaşi valoare se obține prin efectuarea operației de împărțire în GF(2) a lui P(x), dat de (3.14), la G(x), dat de (3.12), în urma căreia se obține polinomul rest $S(x)=x^3 + x^2 + x$, care corespunde semnăturii cu echivalentul zecimal (14).

Trebuie remarcat faptul că efortul de calcul al semnăturii prin operația de împărțire este mult mai laborios decât procedura propusă şi care poate fi rezumată la următorii paşi:

Pas 1. Plecând de la expresia polinomului generator G(x), se elaborează ecuațiile logice (3.11).

Pas 2. Bazat pe ecuațiile logice din pas 1, se construieşte matricea de control H asociată analizorului de semnături serial.

Pas 3. Se operează SAU EXCLUSIV, în secvență, vectorii binari ai coloanelor corespunzătoare termenilor polinomului asociat secvenței binare de comprimat.

Mai menționăm că, pentru un analizor serial dat- nereconfigurabil în dependență de expresia polinomului generator în sensul în care va fi expus în capitolul 5-, primii doi paşi sunt comuni, evaluarea semnăturilor din puncte particulare de test implicând doar traversarea pasului 3.

Procedura 3.2: În încercarea de a elabora o procedură asistată de calculator pentru evaluarea semnăturilor prin structura ASIM din fig. 3.6, să uzităm de teorema 3.1 remarcând că structura este compusă din, în general, n analizoare de semnături seriale, fiecare având asociată o matrice de control H. Urmărind modul de operare de la procedura 3.1, pentru fiecare secvență binară captată pe fiecare canal, se obțin, în general, n semnături seriale, care, în final, sunt operate SAU EXCLUSIV pentru a rezulta semnătura corespunzătoare structurii ASIM.

Pentru concretețe, să plecăm de la aceeaşi expresie dată de (3.12) a lui G(x), pe baza căreia rezultă, în aceeaşi reprezentare schematică, structura ASIM din fig. 3.10. Inhibând încărcarea pe canalele 1, 2 şi 3 şi validând încărcarea doar pe la intrarea 0, se obține analizorul serial având asociată matricea H din fig. 3.9, pe care acum o notăm cu H_0.

Procedând în manieră similară prin inhibarea, de această dată, a canalelor 0, 2 şi 3 şi validând încărcarea doar pe la intrarea 0, obținem analizorul serial a cărei matrice H_1 , elaborată după aceleaşi reguli ca şi matricea H din fig. 3.9- dar cu un alt demaraj- este dată în fig. 3.11.

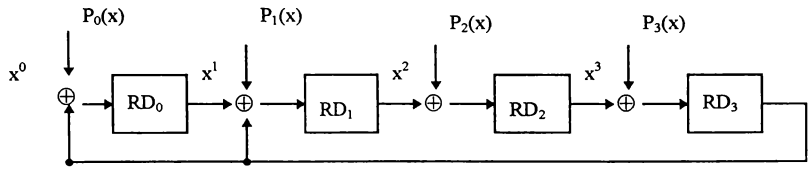


Fig. 3.10

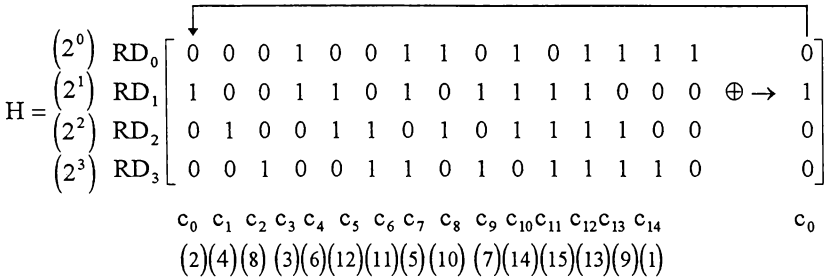


Fig. 3.11

Construcția matricii H_1 începe cu vectorul binar având echivalentul, de această dată, 2 în coloana c_0. Rezultă, prin urmare, că matricea H_1 constituie matricea H_0 cu vectorii binari de coloană deplasați cu o poziție spre stânga. În mod similar, matricea H_2 demarează cu vectorul binar cu echivalentul 4 în coloana c_0, iar matricea H_3 are ca vector de debut în coloana c_0 pe cel cu echivalentul 8. Să exemplificăm utilizarea celor 4 matrici, H_0 pâna la H_3, considerând că lanțul binar, având polinomul P(x) din (3.14) este captat pe canalul 0 din care cauză îl vom nota P_0 (x)- de către structura din fig. 3.10. Pe celelalte canale admitem că sunt recepționate secvențe binare având asociate polinoamele:

$$P_1(x)=x^{31}+x^{27}+x^{20}+x^{15}+x^7+x$$
$$P_2(x)=x^{28}+x^{21}+x^{16}+x^{12}+x^9+x^2+1 \qquad (3.15)$$
$$P_3(x)=x^{29}+x^{25}+x^{17}+x^{10}+x^7+x^6+x^3$$

În conformitate cu exemplul de la procedura 3.1, la captarea şirului binar având asociat polinomul P_0 (x) se obține vectorul semnătură S_0 cu echivalentul zecimal 14. Procedând în manieră similară cu $P_1(x)$, prin utilizarea matricii H_1 din fig. 3.11, se obține vectorul semnătură S_1 cu echivalentul zecimal 6. Repetând modul de lucru pentru P_2 (x) şi P_3 (x), folosind matricile corespunzătoare H_2 şi H_3 , se obțin vectorii semnătură S_2 şi S_3 cu echivalenții zecimali 12, respectiv 3. Operând SAU EXCLUSIV S_0 la S_3 , se obține: $14 \oplus 6 \oplus 12 \oplus 3=7$. Semnătura finală cu echivalentul 7 poate fi obținută şi prin laborioase operații de împărțire în GF(2) sau prin, la fel de laborioasă, urmărire secvențială a modificării conținutului registrului de deplasare, atunci când structura din fig. 3.10 este încărcată, în paralel, cu secvențele având ataşate polinoamele date de (3.14) şi (3.15). Conchizând prin mai marea simplitate a procedurii propuse, rezumăm, în final, paşii pe care îi implică.

Pas 1. Plecând de la expresia polinomului generator G(x), se elaborează ecuațiile logice (3.11).

Pas 2. Bazat pe ecuațiile logice rezultate în pas 1, se construiesc cele n matrici de control H_p , p=0, 1, ..., n-1, asociate analizoarelor de semnături seriale corespondente structurii ASIM.

Pas 3. Se operează SAU EXCLUSIV, în secvență, vectorii binari ai coloanelor corespunzătoare termenilor polinomului asociat secvenței binare de comprimat, repetând pasul pentru fiecare intrare, cu utilizarea matricii H_p adecvate.

Pas 4. Se operează SAU EXCLUSIV vectorii semnătură obținuți în pasul 3 pentru fiecare analizor serial.

Ar mai fi de remarcat faptul că, făcând uz de teorema 3.1, procedura 3.2 poate fi prezentată și în altă formă:

Procedura 3.2*: În baza teoremei 3.1, structura ASIM din fig. 3.6 este echivalentă cu analizorul de semnături serial din fig. 3.7, în care sens, odată elaborat, prin 3.6, polinomul **P**(x), poate fi aplicată procedura 3.1. Rezumativ, evaluarea semnăturii implică traversarea următorilor pași:

Pas 1. Plecând de la expresia polinomului generator G(x), se elaborează ecuațiile logice (3.11).

Pas 2. Bazat pe ecuațiile logice rezultate în pasul 1, se construiește matricea de control H asociată analizorului de semnături serial cu care este echivalentă funcțional structura ASIM.

Pas 3. Plecând de la polinoamele P_p (x), p=0, 1, ..., n-1,și uzitând de (3.6), se determină polinomul cumulativ **P**(x).

Pas 4. Se operează SAU EXCLUSIV, în secvență, vectorii binari ai coloanelor corespunzătoare termenilor polinomului **P**(x).

Reluând în baza acestei noi proceduri exemplul abordat mai sus, în baza celor presupuse de pas 3, luând în considerare (3.14) și (3.15), avem:

$$P(x)=\sum_{p=0}^{3}{}^{\oplus}\ x^p P_p(x)=x^{31}+x^{30}+x^{26}+x^{23}+x^{22}+x^{21}+x^{20}+x^{18}+x^{16}+$$
$$+x^{14}+x^{13}+x^{10}+x^9+x^6+1 \qquad (3.16)$$

Luând în considerare matricea H din 3.9, bazat pe cele prevăzute de pas 4, se operează SAU EXCLUSIV vectorii binari ai coloanelor corespunzătoare termenilor din (3.16). Astfel, exprimând această operație, care se desfășoară secvențial-, în echivalenți zecimali, avem: $2\oplus 1\oplus 14\oplus 5\oplus 11\oplus 12\oplus 6\oplus 8\oplus 2\oplus 9\oplus 13\oplus 7\oplus 10\oplus 12\oplus 1=7$, rezultat care concordă cu cel obținut mai sus.

Mai menționăm că, în ceea ce privește eforturile de elaborare și de calcul ale procedurii 3.2*, acestea sunt comparabile cu cele implicate de procedura 3.2, cu un câștig în simplitate în raport cu procedura 3.2, ambele însă permit consistente economii față de procedurile convenționale anterior amintite.

Cu aceste precizări, să revenim la problema caracterizării prin prisma capacității de detecție a structurii ASIM din fig. 3.6, în care sens introducem următoarea lemă originală.

Lema 3.1. Fiind dată o expresie primitivă de polinom generator G(x), de grad n, și fiind asociate polinoame P_p (x), unde p=0, 1, ..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM, sintetizată pe baza polinomului G(x) cu circuite SAU EXCLUSIV conectate suplimentar în interiorul registrului de deplasare, se caracterizează printr-o probabilitate statistică P de recunoaștere ca funcțional corectă a unei unități testate defecte dată de expresia:

$$P = \frac{2^r - 1}{2^{n+r} - 1} \qquad (3.17)$$

Demonstrație: În cadrul demonstrației vom uzita de teorema 3.1, care permite evaluarea probabilității P în mod facil pe baza echivalenței funcționale dintre structura ASIM din fig. 3.6 și analizorul de semnături serial din fig. 3.7. Prin urmare, problema constă în determinarea gradului polinomului cumulat P(x) care, majorat corespunzător, să substituie pe r în (3.10). Luăm în considerare (3.6), precum și faptul că admitem cazul de interes practic conform căruia lanțurile binare de evaluat au aceeași lungime r, adică gradul cel mai mare al unui polinom P_p (x), p=0, 1, ..., n-1, poate fi cel mult egal cu (r-1). În această situație, termenul cu gradul cel mai mare, dacă P_{n-1} (x) este de grad (r-1), este $x^{n-1}x^{r-1} = x^{n+r-2}$. Evaluând numărul total al erorilor nedetectate în acest caz, prin substituirea lui r în (3.10) [Vlăd-89], se obține:

$$P = \frac{2^{n+r-n} - 1}{2^{n+r} - 1} = \frac{2^r - 1}{2^{n+r} - 1}$$

adică relația (3.17) care trebuia demonstrată.

Relativ la (3.17) ar fi de remarcat că pentru valori mari pentru n, dar în special r, se poate obține probabilitatea reală P_r de recunoaștere ca funcțional corectă a unei unități testate defecte pentru o structură ASIM de tipul celei din fig. 3.6. Astfel , în urma neglijărilor evidente, rezultă $P_r=2^{-n}$, într-un fel surprinzător deoarece, în aceleași condiții ipotetice și în urma efectuării unor neglijări asemănătoare, din (3.10) rezultă $P_{asr} \cong 2^{-n}$. Cu alte cuvinte comprimarea parelelă și cea serială prezintă, prin prisma probabilității reale de recunoaștere ca funcțional corectă a unei unități testate defecte, aceeași capacitate de detecție a potențialelor defecte. Trebuie totuși menționat că- analizând problema nu prin raportare așa cum o realizează probabilitatea, ci în mod absolut- numărul total N_{TN} al erorilor potențiale nedetectate este, la o structură ASIM, $N_{TN}=(2^r-1)2^r$, pe când, la un analizor de semnături serial echivalent, același parametru este $N_{TNas}=(2^{r-n}-1)2^r$. Deci, N_{TN} este cu atât mai mare în raport cu N_{TNas}, cu cât n este mai mic decât r.

3.3.2. Structură ASIM cu circuite SAU EXCLUSIV conectate în exteriorul registrului de deplasare

Facem precizarea că, pentru a distinge structura ASIM din fig. 3.6 de cea pe care o vom analiza în acest paragraf, în cele ce urmează vom nota parametrii care se referă la structura ASIM din paragraful anterior cu indicele A, iar cei care se referă la structura ASIM din prezentul paragraf cu indicele B. Pentru a putea face unele evaluări comparative între cele două structuri, apelăm, în vederea sintezei, la aceeași expresie generală (3.2) pentru polinomul generator G(x). Pentru a caracteriza noua structură ASIM prin prisma capacității de detecție, introducem, mai întâi, următoarea teoremă originală:

Teorema 3.2. Fiind dată o expresie de polinom generator G(x), primitivă sau neprimitivă, de grad n, cu mai mult de un termen, și fiind asociate polinoame P_p (x), p=0, 1, ..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM, sintetizată pe baza polinomului G(x) cu circuite SAU EXCLUSIV conectate în exteriorul registrului de deplasare, generează

semnături identice cu cele furnizate de către un analizor de semnături serial, de acelaşi tip şi sintetizat pe baza aceluiaşi polinom generator G(x), dar încărcat cu fluxul binar având asociat polinomul:

$$\mathbf{P}(x)=\sum_{p=0}^{n-1}{}^{\oplus}\ x^p P_p(x), \text{ unde } \sum^{\oplus} \text{ semnifică suma modulo 2.}$$

Demonstraţie: Uzitând la sinteză de expresia (3.2) pentru G(x), structura ASIM cu circuite SAU EXCLUSIV conectate în exteriorul registrului de deplasare se prezintă în fig. 3.12. Ca şi în paragraful anterior, denumirea se referă la circuitele SAU EXCLUSIV corespunzătoare lanţului de reacţie şi nu cele implicit revendicate de încărcarea paralelă şi care sunt prevăzute între rangurile registrului de deplasare.

În cadrul demonstraţiei vom uzita, din nou,de proprietatea de liniaritate specifică unei structuri de tipul celei prezentate în fig.3.12 şi vom apela la principiul superpoziţiei. Făcând $P_p(x)=0$, pentru p=1, 2, ..., n-1, schema din fig.3.12 se transformă în convenţionalul analizor de semnături serial cu circuite SAU EXCLUSIV conectate în exteriorul rangurilor RD_p ale registrului de deplasare. În acest caz, dispar circuitele SAU EXCLUSIV conectate între rangurile registrului de deplasare şi rămân doar cele conectate pe lanţul de reacţie.

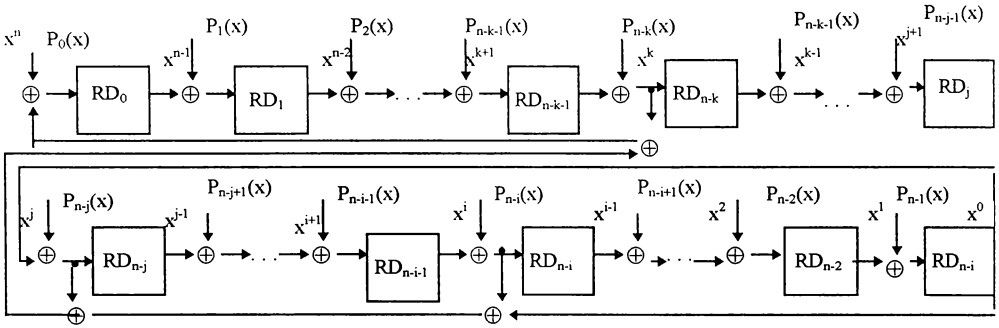


Fig. 3.12

Pentru o astfel de schemă este cunoscută [Vlăd-89] corelaţia dintre polinomul $P_0(x)$, asociat lanţului binar captat serial, şi polinomul $S_{0B}(x)$, asociat semnăturii generate, şi anume:

$$\left.\begin{aligned} P_0(x) &= Q_0(x)\cdot G(x)+S_{0A}(x)\\ S_{0B}(x) &= \left|\frac{x^n\cdot S^*_{0A}(x)}{G(x)}\right| \end{aligned}\right\} \qquad (3.18)$$

unde $S_{0A}(x)$ reprezintă polinomul asociat semnăturii generate de către analizorul de semnături serial din fig.3.7 în care se încarcă secvenţa binară având polinomul $P_0(x)$ [şi nu polinomul $\mathbf{P}(x)$], $Q_0(x)$ reprezintă polinomul asociat lanţului binar care părăseşte analizorul, atunci când r > n, la ieşirea rangului RD_{n-1} şi barele semnifică polinomul cât obţinut în urma efectuării în GF(2) a operaţiilor prevăzute între bare. Făcând acum $P_p(x)=0$, pentru p=0, 2, 3, ..., n-1, şi introducând în schema secvenţială liniară fluxul binar care are asociat polinomul $P_1(x)$ prin circuitul SAU EXCLUSIV care

precede rangul RD_1 , avem de-a face cu acelaşi analizor de semnături serial încărcat însă cu un impuls de tact în avans. Ca şi anterior, aceasta echivalează cu captarea în analizorul de semnături serial a secvenţei binare având asociat polinomul $xP_1(x)$. Prin analogie cu (3.4), (3.6), (3.8) şi (3.18), avem:

$$\left.\begin{aligned} x^1 \cdot P_1(x) &= Q_1'(x) \cdot G(x) + S_{1A}(x) \\ x^1 \cdot S_{1A}(x) &= S_{1A}'(x) \cdot G(x) + S''_{1A}(x) \\ S_{0\text{Ш}}(x) &= \left| \frac{x^n \cdot S_{1A}^{*}(x)}{G(x)} \right| \end{aligned}\right\} \qquad (3.19)$$

Procedând, în mod progresiv, la admiterea $P_p(x)=0$, pentru p=0, 1, 3, ..., n-1, apoi pentru p=0, 1, 2, 4, 5, ..., n-1 şi, în ultimă instanţă, pentru p=0, 1, 2, ..., n-3, n-2, şi, aplicând principiul superpoziţiei, la schema din fig.3.12 sunt valabile ecuaţiile (3.6) şi (3.8). Pentru semnăturile parţiale cu polioamele $S_{pA}''(x)$, unde p=0, 1, 2, ..., n-1, avem:

Prin urmare, aplicând principiul superpoziţiei şi proprietatea de liniaritate, structura ASIM din fig.3.12, a cărei descriere funcţională este redată formal de ecuaţiile (3.6), (3.8) şi (3.20), este echivalentă cu analizorul de semnături serial din fig.3.13.

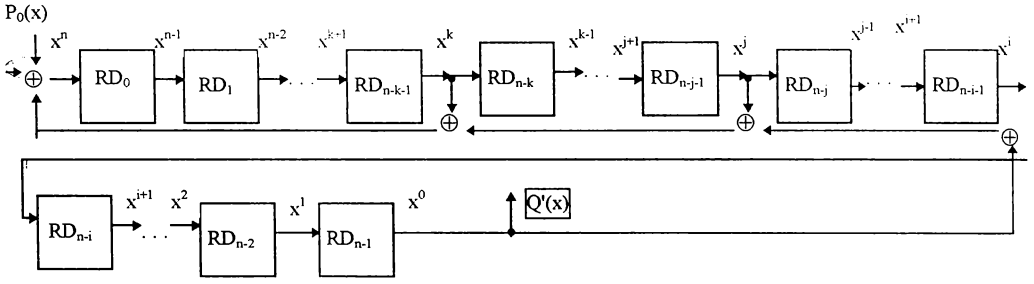


Fig. 3.13

$$
\left.
\begin{aligned}
S'_{0B}(x) &= \left|\frac{x^n \cdot S''_0(x)}{G(x)}\right| \\
S'_{1B}(x) &= \left|\frac{x^n \cdot S''_1(x)}{G(x)}\right| \\
&\text{- - - - - - - - - - - -} \\
S'_{iB}(x) &= \left|\frac{x^n \cdot S''_i(x)}{G(x)}\right| \\
&\text{- - - - - - - - - - - -} \\
S'_{jB}(x) &= \left|\frac{x^n \cdot S''_j(x)}{G(x)}\right| \\
S'_{kB}(x) &= \left|\frac{x^n \cdot S''_k(x)}{G(x)}\right| \\
&\text{- - - - - - - - - - - -} \\
S'_{(n-1)B}(x) &= \left|\frac{x^n \cdot S''_{(n-1)}(x)}{G(x)}\right| \\
&\text{- - - - - - - - - - - -}
\end{aligned}
\right\} \qquad (3.20)
$$

$$
\text{- - - - - - - - - - - - - - - - - -}
$$

$$
S_B(x) = \sum_{p=0}^{n-1} \oplus S_{pB}(x) = \left|\frac{x^n \cdot \sum_{p=0}^{n-1} \oplus S''_p(x)}{G(x)}\right| = \left|\frac{x^n \cdot S_A}{G(x)}\right|
$$

Corelând cele cunoscute cu (3.7), (3.9) şi (3.20), pentru structura din fig.3.13 avem:

$$
\left.
\begin{aligned}
P(x) &= Q'(x) \cdot G(x) + S_A(x) \\
S_B(x) &= \left|\frac{x^n \cdot S_B(x)}{G(x)}\right|
\end{aligned}
\right\} \qquad (3.21)
$$

Se poate conchide că structura din fig. 3.12 este echivalentă din punct de vedere funcţional cu n analizoare de semnături serialecare, prin suprapunere, dau structura din fig.3.13.

Teorema 3.2: permite caracterizarea structurii ASIM din fig.3.12 prin prisma capacităţii de detecţie, precum şi compararea, sub acelaşi aspect, a aceleiaşi structuri cu cea reprezentată în fig. 3.6. Înainte de a trece la investigarea obiectivului menţionat, ca şi în paragraful anterior, vom insista asupra unor proprietăţi ale structurii ASIM din fig. 3.12, cu aceeaşi utilitate, deja amintită, şi care decurg din asocierea la structura în discuţie a unor matrici de control H adecvate. În vederea distingerii acestora de matricile corespunzătoare structurii ASIM realizate anterior, pe care le vom nota, în continuare, cu indicele A, adoptăm pentru notarea matricilor corespunzătoare structurii ASIM în discuţie indicele B. În mod ipotetic, ca şi anterior, considerăm, fără a pierde din

generalitate, că polinomul generator G(x) utilizat la sinteză, este primitiv. În aceste condiții, pentru soluționarea asistată de calculator a determinării semnăturilor etalon, propunem următoarele proceduri originale.

Procedura 3.3: Pentru analizorul de semnături serial din fig. 3.13, corespondent ecuațiilor logice (3.11), din punct de vedere temporal, avem următoarea unică ecuație logică:

$$RD_0\,(t+1)=RD_{n-1}\,(t)\oplus RD_{n-i-1}\,(t)\oplus RD_{n-j-1}\,(t)\oplus RD_{n-k-1}\,(t) \qquad (3.22)$$

În general, având o expresie de polinom generator G(x) cu t termeni, rezultă o ecuație (3.22) cu (t-1) operanzi. Plecând de la această ecuație, prin aplicarea la construcția fiecărei coloane, în mod secvențial- coloană cu coloană- se elaborează matricea H_B.

Pentru exemplificare, vom considera, din nou, polinomul primitiv G(x) cu expresia (3.12), pe baza căruia rezultă sinteza analizorului de semnături serial, care, schematic, este prezentată în fig. 3.14.

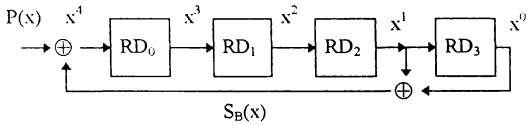


Fig. 3.14

Cei (3 termeni-1=) 2 termeni ai ecuației logice- care ia în considerare factorul timp- specifică structurii din fig.3.14, sunt următorii:

$$RD_0\,(t+1)=RD_3\,(t)\oplus RD_2\,(t) \qquad (3.23)$$

Luând în considerare (3.23) şi cele anterior expuse, construcția matricii H_B asociată structurii din fig. 3.14 este sugerată- plecând de la aceeaşi ' sămânță' reprezentată de o unitate binară în rangul RD_0 şi cu zerouri în RD_1 la RD_3 - în fig. 3.15.

$$H=\begin{array}{c} \\ (2^0)\ RD_0 \\ (2^1)\ RD_1 \\ (2^2)\ RD_2 \\ (2^3)\ RD_3 \end{array}\begin{bmatrix} 1 & & 0 & 0 & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & & 0 & & 1 \\ 0 & & 1 & 0 & & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & & 0 & & 0 \\ 0 & \oplus & 0 & 1 & \oplus & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & \oplus & 0 & \oplus & 0 \\ 0 & & 0 & 0 & & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & & 1 & & 0 \end{bmatrix}$$

$$\begin{array}{cccccccccccccccccc} c_0 & & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} & c_{14} & & & & c_0 \\ (1) & & (2) & (4) & (9) & (3) & (6) & (13) & (10) & (5) & (11) & (7) & (15) & (14) & (12) & & & & & (8) \end{array}$$

Fig. 3.15

Cele conectate în contextul prezentării matricii H_A din fig. 3.9 îşi păstrează, şi aici, valabilitatea, corespunzător adaptate. Din nou, maniera de construcţie a matricii H_B , ilustrată în fig. 3.15, pune în relief modalitatea facilă de soluţionare a acesteia asistat de calculator.

Odată stabilită matricea H_B şi fiind cunoscută lungimea ferestrei de comprimare prin numărul impulsurilor de tact, se stabilesc termenii polinomului P(x) asociat lanţului binar de evaluat şi se operează SAU EXCLUSIV, în secvenţă, vectorii binari ai coloanelor corespunzătoare unităţilor binare din lanţul binar de comprimat. În calitate de exemplu, admitem că dorim evaluată, prin analizorul din fig. 3.14, semnătura pentru secvenţa binară descrisă de polinomul P(x) cu expresia (3.14). Cu aceleaşi observaţii şi aceleaşi coloane, dar cu vectori binari- şi în consecinţă echivalenţi zecimali- diferiţi pentru coloane, semnătura se obţine, în acest caz, operând SAU EXCLUSIV, în secvenţă, vectorii binari cu următorii echivalenţi zecimali: $2 \oplus 15 \oplus 10 \oplus 15 \oplus 5 \oplus 3 \oplus 1 = 15$. Aceeaşi valoare a echivalentului se obţine pentru vectorul semnătură dacă se efectuează, în GF(2), operaţiile prevăzute de (3.18), care sunt, se poate remarca în mod facil, mult mai laborioase în raport cu cele corespunzăroare procedurii propuse.

În ceea ce priveşte eşalonarea activităţilor procedurii, aceasta se poate realiza prin aceiaşi 3 paşi ca la procedura 3.1 cu menţiunea substituirii, în pasul 1, ecuaţiilor logice (3.11) cu ecuaţia logică (3.22).

Procedura 3.4: Încercând elaborarea unei proceduri asistate de calculator pentru evaluarea semnăturilor prin structura ASIM din fig. 3.12, să uzităm, de această dată, de teorema 3.2, remarcând că structura este compusă din, în general, n analizoare de semnături seriale, fiecare având asociată o matrice de control H_B . Urmărind modul de operare de la procedura 3.3 pentru fiecare secvenţă binară captată pe fiecare canal, se obţin, în general, n semnături seriale, care, în final, sunt operate SAU EXCLUSIV pentru a rezulta semnătura corespunzătoare structurii ASIM. Pentru concretete, să plecăm de la aceeaşi expresie, dată de (3.12), a lui G(x), pe baza căreia rezultă, în aceeaşi reprezentare schematică, structura ASIM din fig.3.16.

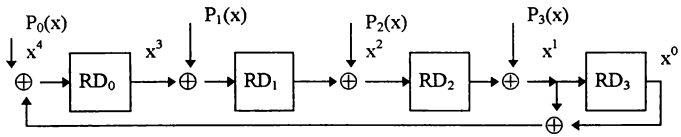


Fig. 3.16

Operând în maniera descrisă la procedura 3.2, prin deplasarea la stânga a vectorului binar corespunzător coloanei de debut, se obţin cele 4 matrici, H_{0B} la H_{3B}, dintre care în fig. 3.17 se prezintă, în calitate de exemplu, matricea H_{3B}.

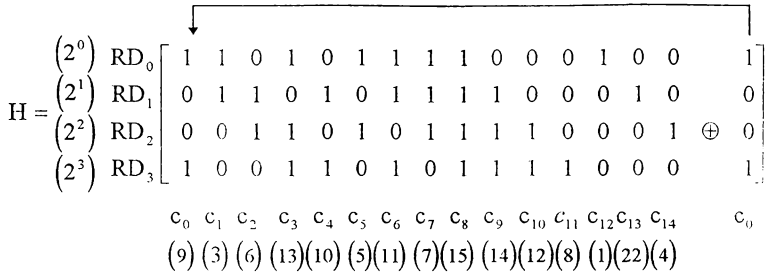


Fig. 3.17

Considerând, acum, încărcarea structurii ASIM din fig.3.16 cu secvențele binare având asociate polinoamele cu expresiile (3.14) şi (3.15), prin utilizarea celor 4 matrici, se obțin următorii echivalenți zecimali ai celor 4 semnături: S_{0B}=15, S_{1B}=6, S_{2B}=13 şi S_{3B}=3. Operând SAU EXCLUSIV vectorii binari corespunzători la S_{0B} , S_{1B} , S_{2B} şi S_{3B} , se obține: $15 \oplus 6 \oplus 13 \oplus 3 = 7$. Semnătura finală cu echivalentul 7 poate fi obținută şi prin laborioasele operații prevăzute de (3.21) sau prin, la fel de laborioasă, urmărire secvențială a modificării conținutului registrului de deplasare, atunci când structura din fig. 3.16 este încărcată, în paralel,cu secvențele având ataşate polinoamele date de (3.14) şi (3.15). Rezumând activitățile procedurii pe paşi, aceştia sunt asemănători cu cei de la procedura 3.2.

Pas 1. Plecând de la expresia polinomului generator G(x), se elaborează ecuația logică (3.22).

Pas 2. Bazat pe ecuația din pasul 1, se construiesc cele n matrici de control H_{pB} , p=0, 1, ..., n-1, asociate analizoarelor de semnături seriale corespondente structurii ASIM.

Pas 3. Se operează SAU EXCLUSIV, în secvență, vectorii binari ai coloanelor corespunzătoare termenilor polinomului asociat secvenței binare de comprimat, repetând pasul cu utilizarea matricii
H_{PB} adecvate.

Pas 4. Se operează SAU EXCLUSIV vectorii semnătură obținuți în pasul 3 pentru fiecare analizor serial.

Ar mai fi de remarcat faptul că, făcând uz, de această dată, de teorema 3.2, şi procedura 3.4 poate fi prezentată într-o altă formă.

Procedura 3.4*: În baza teoremei 3.2, structura ASIM din fig. 3.12 este echivalentă cu analizorul de semnături serial din fig. 3.13, în care sens, odată elaborat, prin (3.6), polinmul P(x), poate fi aplicată procedura 3.3. Rezumativ, evaluarea semnăturii implică traversarea următorilor paşi.

Pas 1. Plecând de la expresia polinomului generator G(x), se elaborează ecuația logică (3.22).

Pas 2. Bazat pe ecuația din pasul 1, se construieşte matricea de control H_B asociată analizorului de semnături serial cu care este echivalentă funcțional structura ASIM.

Pas 3. Plecând de la polinoamele P_p (x), p=0, 1, ..., n-1, şi uzitând de (3.6), se determină polinomul cumulativ P(x).

Pas 4. Se operează SAU EXCLUSIV, în secvență, vectorii binari ai coloanelor corespunzătoare termenilor polinomului P(x).

Reluând în baza acestei noi proceduri exemplul abordat mai sus, pentru P(x) avem expresia dată de (3.16). Dacă luăm în considerare matricea H_B din fig. 3.15, în baza celor prevăzute de

pasul4, se operează SAU EXCLUSIV vectorii binari ai coloanelor corespunzătoare termenilor din (3.16). Astfel, exprimând această operație, care se desfășoară secvențial, în echivalenți zecimali, avem: $2\oplus 1\oplus 15\oplus 5\oplus 10\oplus 13\oplus 6\oplus 9\oplus 2\oplus 8\oplus 12\oplus 7\oplus 11\oplus 13\oplus 1=7$, rezultat care concordă cu cel obținut mai sus.

Mențiunea legată de efortul de elaborare și de calcul implicat de procedura 3.2* își păstrează valabilitatea și în ce privește procedura 3.4*. Revenind, ca și anterior, la problema caracterizării structurii ASIM din fig.3.12 prin prisma capacității de detecție, introducem următoarea lemă originală.

Lema 3.2. Fiind dată o expresie de polinom generator G(x), primitivă, de grad n, și fiind asociate polinoame P_p (x), unde p=0, 1, ..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM, sintetizată pe baza polinomului G(x) cu circuite SAU EXCLUSIV conectate în exteriorul registrului de deplasare, este echivalentă prin prisma capacității de detecție cu structura ASIM, sintetizată pe baza aceluiași polinom G(x) dar cu circuite SAU EXCLUSIV conectate suplimentar în interiorul registrului de deplasare, fiind caracterizată prin aceeași probabilitate statistică P de recunoaștere ca funcțional corectă a unei unități testate defecte, dată de relația(3.17)

Demonstrație: Vom uzita de teoremele 3.1 și 3.2, care permit, în vederea comparației prin prisma capacității de detecție, substituirea celor două structuri ASIM (fig. 3.6, respectiv fig. 3.12) prin funcțional echivalentele analizoare de semnături seriale (fig 3.7, respectiv fig. 3.13). Ori, în [Vlăd-89,lema 5.1.2] se demonstrează că cele două analizoare seriale posedă același număr total de erori nedetectate N_{TN} și , prin urmare, aceeași probabilitate statistică P (și reală P_r) de recunoaștere ca funcțional corectă a unei unități testate defecte, dată de relația (3.17) în baza lemei 3.1.

Plecând de la echivalența statuată prin lema 3.2, toate comentariile succesive lemei 3.1 pot fi extrapolate la structura ASIM din fig.3.12.

3.3.3. Structură ASIM combinată

În căutarea unor scheme eficiente pentru comprimarea paralelă, propunem, plecând de la aceeași expresie (3.2) pentru polinomul generator G(x), o nouă structură ASIM, prezentată în fig.3.18 și a cărei parametri îi vom nota cu indicele C.

Se poate remarca faptul că structura ASIM posedă o legătură de reacție directă între rangul cel mai semnificativ al registrului de deplasare RD_{n-1} și circuitul SAU EXCLUSIV conectat pe intrarea rangului RD_0 , aceasta asigurând efectuarea unei operații de împărțire la polinomul (x^n+1). Pe de altă parte, ieșirea arborelui SAU EXCLUSIV de pe intrarea rangului RD_0 constituie intrare pentru toate circuitele SAU EXCLUSIV corespunzătoare termenilor interiori din expresia lui G(x), adică a termenilor cuprinși între 1 și x^n. Această din urmă legătură asigură înmulțirea resturilor parțiale cu polinomul $(x^k+x^{j-i}x^i+1)$. Prin urmare, structura captează în paralel fluxurile de informație pe la mai multe intrări și realizează comprimarea uzitând de o operație combinată, de împărțire și de înmulțire în GF(2), fapt care ne-a determinat să adoptăm pentru această structură ASIM denumirea

de 'combinată'. Pentru a caracteriza noua structură ASIM prin prisma capacității de detecție, introducem următoarea teoremă originală.

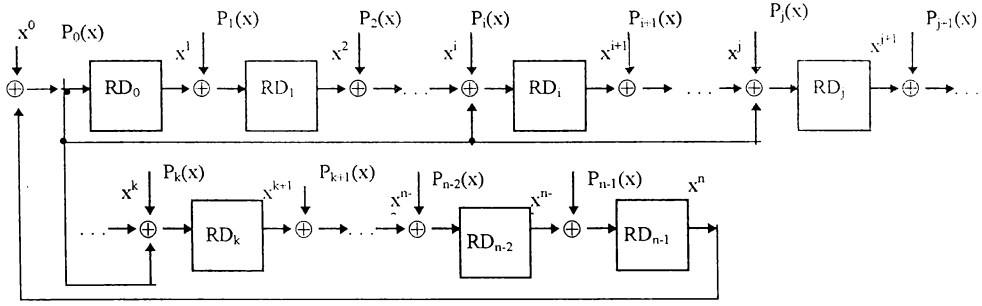


Fig. 3.18

Teorema 3.3. Fiind dată o expresie de polinom generator G(x), primitivă sau neprimitivă, de grad n, cu mai mult de un termen, și fiind asociate polinoame P_P (x), p=0, 1, ..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM combinată, sintetizată pe baza polinomului G(x), generează semnături identice cu cele furnizate de către un analizor de semnături serial, de același tip și sintetizat pe baza aceluiași polinom generator G(x), dar încărcat cu fluxul binar având asociat polinomul:

$$P(x)=\sum_{p=0}^{n-1}{}^{\oplus}\ x^p P_p(x), \text{ unde } \sum^{\oplus} \text{ semnifică suma modulo 2.}$$

Demonstrație: Și în acest caz, ca și la demonstrarea teoremelor 3.1 și 3.2, vom uzita de proprietatea de liniaritate conferită de operatorul SAU EXCLUSIV și apelăm la principiul superpoziției.

Păstrând raționamentul folosit, facem, mai întâi, $P_p(x)=0$ pentru p=1, 2, ..., n-1, și observăm că schema din fig. 3.18 se transformă în convenționalul analizor de semnături serial de tipul combinat, circuitele SAU EXCLUSIV de încărcare paralelă devenind inactive. Pentru o astfel de schemă este cunoscută [Vlăd-89] corelația dintre polinomul P_0 (x), asociat lanțului binar captat serial, și polinomul S_{0C} (x), asociat semnăturii generate, și anume:

$$\left.\begin{aligned} P_0(x) &= Q_0(x)\cdot G(x) + S_{0A}(x) \\ S_{0C}(x) &= x^n \cdot S_{0A}(x) + S_{0B}(x)\cdot G(x) \end{aligned}\right\} \qquad (3.24)$$

unde S_{0A} (x) și S_{0B} (x) reprezintă polinoamele asociate semnăturilor generate analizoare seriale, având la baza sintezei același polinom generator G(x), și prevăzute cu circuite SAU EXCLUSIV conectate în interiorul registrului de deplasare (de tipul schemei din fig. 3.7), respectiv conectate în exteriorul registrului de deplasare(de tipul schemei din fig. 3.13).

În conformitate cu același raționament, facem acum P_p (x)=0,pentru p=0, 2, 3, ... n-1, și introducând în schema secvențială liniară fluxul binar care are asociat polinomul P_1 (x) prin circuitul SAU EXCLUSIV care precede rangul RD_1 , avem de-a face cu același analizor de semnături serial,

anterior amintit- încărcat însă cu un impuls de tact în avans. Din punct de vedere formal, şi aici, aceasta echivalează cu captarea în analizorul serial a secvenţei binare având asociat polinomul $xP_1(x)$.
Prin analogie cu (3.4), (3.6), (3.8), (3.18) şi (3.19), avem:

$$x^1 P_1(x)=Q_1(x)G(x)+S_{1A}(x)$$
$$x^1 S_{1A}(x)=S_{1A}'(x)G(x)+S_{1A}''(x) \qquad (3.25)$$
$$S_{1C}(x)=x^n S_{1A}''(x)+S_{1B}(x)G(x)$$

Procedând, în mod progresiv, la admiterea $P_p(x)=0$, pentru p=0, 1, 3, 4, ..., n-1, apoi pentru p=0, 1, 2, 4, 5, ..., n-1, şi în ultimă instanţă pentru p=0, 1, ..., n-3, n-2, şi , aplicând principiul superpoziţiei, la schema din fig. 3.18 sunt valabile ecuaţiile (3.6) şi (3.8). Pentru semnăturile parţiale cu polinoamele $S_{pA}''(x)$, unde p=0, 1, 2, ..., n-1, avem:

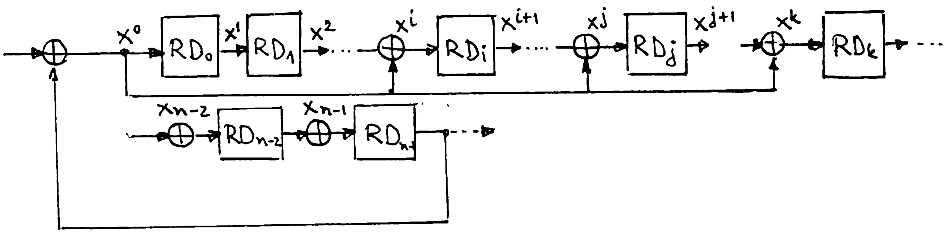


Fig. 3.19

$$\left.\begin{array}{l}
S_{0C}(x) = x^n \cdot S_{0A}''(x) + S_{0B}(x)\cdot G(x)\\
S_{1C}(x) = x^n \cdot S_{1A}''(x) + S_{1B}(x)\cdot G(x)\\
-------------------\\
S_{iC}(x) = x^n \cdot S_{iA}''(x) + S_{iB}(x)\cdot G(x)\\
-------------------\\
S_{jC}(x) = x^n \cdot S_{jA}''(x) + S_{jB}(x)\cdot G(x)\\
-------------------\\
S_{kC}(x) = x^n \cdot S_{kA}''(x) + S_{kB}(x)\cdot G(x)\\
-------------------\\
S_{(n-1)C}(x) = x^n \cdot S_{(n-1)A}''(x) + S_{(n-1)B}(x)\cdot G(x)\\
---\\
S_C(x) = \sum_{p=0}^{n-1}{}^{\oplus} S_{pC}(x) = x^n \sum_{p=0}^{n-1}{}^{\oplus} S_{pA}''(x) + G(x)\sum_{p=0}^{n-1}{}^{\oplus} S_{pB}(x) = x^n \cdot S_A(x) + G(x)\cdot S_B(x)
\end{array}\right\} (3.28)$$

Prin urmare, aplicând principiul superpoziţiei şi proprietatea de liniaritate, structura ASIM din fig.3.18, a cărei descriere funcţională este redată formal de ecuaţiile (3.6), (3.8) şi (3.26), este echivalentă cu analizorul de semnături serial din fig.3.19.
Corelând cele cunoscute cu (3.7), (3.9) şi (3.26), pentru structura din fig.3.19 avem:

$$\left.\begin{aligned} P(x) &= Q''(x)\cdot G(x) + S_A(x) \\ S_C(x) &= x^n \cdot S_A(x) + S_B(x)\cdot G(x) \end{aligned}\right\} \qquad (3.27)$$

Se poate conchide că structura din fig.3.18 este echivalentă din punct de vedere funcțional cu n analizoare de semnături seriale care, prin suprapunere, dau structura din fig.3.19.

Teorema 3.3: permite caracterizarea structurii ASIM din fig.3.18 prin prisma capacității de detecție, precum şi compararea sub acelaşi aspect, a aceleiaşi structuri cu cele reprezentate în fig.3.6 şi în fig.3.12. Înainte de a trece la investigarea obiectivului menționat, ca şi în paragrafele anterioare, vom insista asupra unor proprietăți ale structurii ASIM din fig.3.18, cu aceiaşi utilitate deja amintită, şi care decurg din asocierea la structura în discuție a unor matrici de control H adecvate. Precizând că, fără a pierde din generalitate, vom considera, şi în acest caz, că polinomul G(x) utilizat la sinteză este primitiv şi că vom nota cu indicele C matricile de interes în contextul structurii ASIM combinate, pentru soluționarea asistată de calculator a determinării semnăturilor etalon, propunem următoarele proceduri originale.

Procedura 3.5: Pentru analizorul de semnături serial din fig.3.19, din punct de vedere temporal, sunt valabile ecuațiile (3.11), dar apare o deosebire în sensul că "sămânța" de la care se pleacă procesul secvențial de generare a vectorilor binari corespunzători coloanelor. Astfel, la o matrice H_A corespunzătoare unui analizor serial cu circuite SAU EXCLUSIV intercalate între rangurile registrului de deplasare, "sămânța" constă din vectorul binar cu echivalentul zecimal în prima coloană (c_0), ceea ce determină, prin aplicarea ecuațiilor (3.11), ca matricea H_A să conțină, corespunzător primelor n coloane, o submatrice unitate. Pe de altă parte, la o matrice H_C corespunzătoare unui analizor serial combinat, "sămânța" constă din vectorul binar stabilit de termenii expresiei lui G(x), cu excepția lui x^n, în prima coloană (c_0), ceea ce determină, prin aplicarea acelorași ecuații (3.11), ca matricea H_C să conțină submatricea unitate anterior amintită, în coloanele finale. Cu alte cuvinte, matricea H_A reprezintă concatenarea dintre submatricea unitate şi o submatrice a resturilor R, iar matricea H_C se obține prin interschimbarea pozițiilor celor două submatrice, conform cu:

$$H_A=[\,[\,I\,][\,R\,]\,]$$
$$H_C=[\,[\,R\,][\,I\,]\,] \qquad (3.28)$$

Pentru exemplificare, să considerăm din nou, polinomul primitiv G(x) cu expresia 3.12, pe baza căruia rezultă sinteza analizorului de semnături serial combinat, care schematic, este prezentată în fig.3.20.

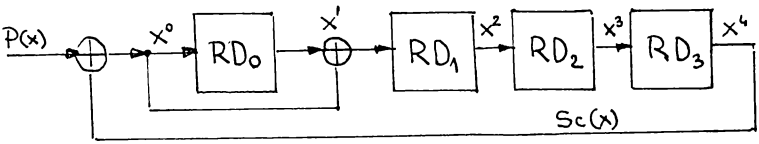


Fig. 3.20

Fig. 3.20

Luând în considerare (3.13) şi cele anterior expuse, construcţia matricii H_C asociată structurii din fig.3.20 se poate efectua interschimbând cele două submatrice ale matricii H_A din fig.3.9 în conformitate cu (3.28), modalitate în care rezultă matricea H_C din fig.3.21.

$$H = \begin{array}{c} \\ (2^0)\ RD_0 \\ (2^1)\ RD_1 \\ (2^2)\ RD_2 \\ (2^3)\ RD_3 \end{array} \begin{bmatrix} \overbrace{\begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{matrix}}^{[R]} & \overbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}^{[I]} \end{bmatrix}$$

$$\begin{matrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} & c_{14} \\ (3) & (6) & (12) & (11) & (5) & (10) & (7) & (14) & (15) & (13) & (9) & (1) & (2) & (4) & (8) \end{matrix}$$

Fig. 3.21

Evident, matricea H_C se poate elabora şi uzitând direct de ecuaţiile (3.13), fără intermedierea prin matricea H_A , dar plecând cu " sămânţa" corespunzătoare reprezentată de vectorul binar cu echivalentul zecimal 3. Mai menţionăm că toate comentariile realizate la matricile H_A şi H_B , adaptate corespunzător, pot fi extrapolate şi la H_C.

Odată stabilită matricea H_C şi fiind cunoscută lungimea ferestrei de comprimare prin numărul impulsurilor de tact, se stabilesc termenii polinomului P(x) asociat lanţului binar de evaluat şi se operează SAU EXCLUSIV, în secvenţă, vectorii binari ai coloanelor corespunzătoare unităţilor binare din lanţul binar de comprimat. În calitate de exemplu, admitem că dorim evaluată, prin analizorul din fig.3.20, semnătura pentru secvenţa binară descrisă de polinomul P(x) cu expresia (3.14). Cu aceleaşi observaţii şi cu aceleaşi coloane, dar cu vectori binari, şi în consecinţă echivalenţi zecimali, diferiţi pentru coloane, semnătura se obţine, în acest caz, operând SAU EXCLUSIV, în secvenţă, vectorii binari cu următorii echivalenţi zecimali: 6⊕14⊕1⊕15⊕5⊕3=1. Aceeaşi valoare a echivalentului se poate obţine pentru vectorul semnătură dacă se efectuează, în GF(2), operaţiile prevăzute de (3.24) care sunt, şi în acest caz, mult mai laborioase în raport cu cele corespunzătoare procedurii propuse.

În ceea ce priveşte eşalonarea activităţilor procedurii, aceasta se poate realiza prin aceiaşi 3 paşi ca la procedura 3.1 cu menţiunea demarării procesului de construcţie a matricii H_C cu o "sămânţă" diferită, în sensul precizat, faţă de construcţia matricii H_A.

Procedura 3.6: Încercând elaborarea unei proceduri asistate de calculator pentru evaluarea semnăturilor prin structura ASIM din fig.3.18, să uzităm, de această dată, de teorema 3.3, remarcând că structura este compusă din, în general, n analizoare de semnături seriale, fiecare având asociată o matrice de control H_C. Urmărind modul de operare de la procedura 3.5 pentru fiecare secvenţă binară captată pe fiecare canal, se obţin, în general, n semnături seriale, care, în final, sunt operate

să plecăm de la aceeaşi expresie, dată de (3.12), a lui G(x), pe baza căreia rezultă, în aceeaşi reprezentare schematică, structura ASIM din fig.3.22.

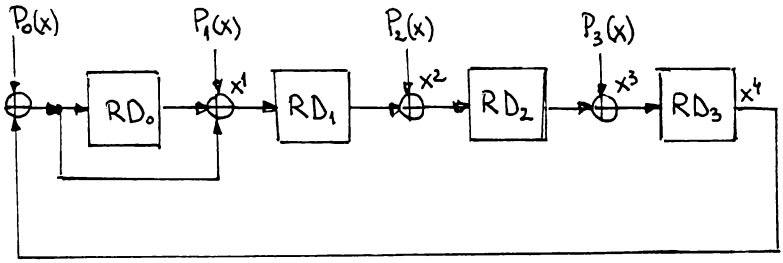


Fig. 3.22

Operând în maniera descrisă la procedura 3.2, prin deplasarea la stânga a vectorului binar corespunzător coloanei de debut, se obţin cele 4 matrici, H_{0C} la H_{3C} , dintre care în fig.3.23 se prezintă, în calitate de exemplu, matricea H_{2C}.

$$H = \begin{matrix} (2^0)\ RD_0 \\ (2^1)\ RD_1 \\ (2^2)\ RD_2 \\ (2^3)\ RD_3 \end{matrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{matrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} & c_{14} \\ (9) & (3) & (6) & (13) & (10) & (5) & (11) & (7) & (15) & (14) & (12) & (8) & (1) & (22) & (4) \end{matrix} \qquad c_0$$

Fig. 3.23

Considerând, acum, încărcarea structurii ASIM din fig.3.22 cu secvenţele binare având asociate polinoamele cu expresiile (3.14) şi (3.15), prin utilizarea celor 4 matrici, se obţin următorii echivalenţi zecimali ai celor 4 semnături: S_{0C}=1, S_{1C}=10, S_{2C}=7 şi S_{3C}=5. Operând SAU EXCLUSIV vectorii semnătură S_{0C}, S_{1C} , S_{2C} şi S_{3C} se obţine: 1+10+7+5=9. Semnătura finală cu echivalentul 9 poate fi obţinută şi prin laborioase operaţii prevăzute de (3.27) sau prin, la fel de laborioasa, urmărire secvenţială a modificării conţinutului registrului de deplasare, atunci când structura din fig.3.22 este încărcată, în paralel, cu secvenţele având ataşate polinoamele date de (3.14) şi (3.15). În ceea ce priveşte eşalonarea activităţilor procedurii pe paşi, aceasta implică cei 4 paşi specificaţi la procedura 3.2 cu menţiunea utilizării în pasul 2 a unor "sămânţe" corespunzătoare, în sensul specificat.

Ar mai fi de remarcat faptul că, făcând uz, de această dată, de teorema 3.3, şi procedura 3.6 poate fi prezentată sub o altă formă.

Procedura 3.6*: În baza teoremei 3.3, structura ASIM din fig.3.18 este echivalentă cu analizorul de semnături serial din fig.3.19, în care sens, odată elaborat, prin (3.6), polinomul P(x), poate fi aplicată procedura 3.5. Procedura implică cei patru paşi expuşi explicit la procedura 3.2* cu menţiunea utilizării unei "sămânţe" adecvate la construcţia matricii H_C din pasul 2.

Reluând în baza acestei noi proceduri exemplul abordat mai sus, pentru P(x) avem expresia dată de (3.16). Dacă luăm în considerare matricea H_C din fig.3.21, se operează SAU EXCLUSIV vectorii binari ai coloanelor corespunzătoare termenilor din (3.16). Astfel, exprimând această operaţie, care se desfăşoară secvenţial, în echivalenţi zecimali avem:

6⊕3⊕1⊕15⊕14⊕7⊕10⊕11⊕6⊕8⊕4⊕9⊕13⊕7⊕3=9, rezultat care concordă cu cel obţinut mai sus.

Menţiunile legate de efortul de elaborare şi de calcul implicat de procedurile 3.2* şi 3.4* îşi păstrează valabilitatea şi în ce priveşte procedura 3.6*.

Revenind, ca şi anterior, la problematica caracterizării structurii ASIM din fig.3.18 prin prisma capacităţii de detecţie, introducem următoarea lemă originală.

Lema 3.3: Fiind dată o expresie primitivă de polinom generator G(x), de grad n, şi fiind asociate polinoame P_p (x), unde p=0, 1, ..., n-1, de grad (r-1), la fluxurile binare captate în paralel pe cele n canale, structura ASIM combinată, sintetizată pe baza polinomului G(x), este echivalentă cu structurile ASIM sintetizate pe baza aceluiaşi polinom G(x) cu circuite SAU EXCLUSIV conectate suplimentar în interiorul registrului de deplasare, respectiv conectate în exteriorul registrului de deplasare, fiind caracterizată prin aceeaşi probabilitate statistică P de recunoaştere ca funcţional corectă a unei unităţi testate defecte, dată de relaţia (3.17).

Demonstraţie: Vom uzita teoremele 3.1, 3.2 şi 3.3, care permit, în vederea comparaţiei prin prisma capacităţii de detecţie, substituirea celor trei structuri ASIM (figura 3.6, fig.3.12, respectiv 3.18) prin funcţional echivalentele analizoare de semnături seriale (figura 3.7, fig.3.13, respectiv 3.19). Ori, în [Vlăd-89, lema 5.2.1] se demonstrează că cele trei analizoare seriale posedă acelaşi număr total de erori nedetectate N_{TN} şi, prin urmare, aceeaşi probabilitate statistică P (şi reală P_r) de recunoaştere ca funcţional corectă a unei unităţi testate defecte, dată de relaţia (3.17) în baza lemelor 3.1 şi 3.2

Plecând de la echivalenţa statuată prin lema 3.3, toate comentariile succesive lemei 3.1 pot fi extrapolate la structura ASIM combinată din fig.3.18.

3.3.4. Analiză comparativă a structurilor ASIM

Arătăm, din debut, că se poate demonstra [Vlăd-89] că (3.17) furnizează valoarea minimă pentru parametrul de contraperformanţă în testare, reprezentat de probabilitatea de recunoaştere ca funcţional corectă a unei unităţi testate defecte, atunci când polinomul G(x) este primitiv. Raportându-ne la anterior expusele legate de performanţa, exprimată prin frecvenţa trenului de tact de comprimare, unei structuri ASIM, indiferent de tip, rezultă ca favorabilă alegerea unui polinom generator G(x) aparţinând familiei celor primitive şi subfamiliei celor primitive cu număr minim de termeni. Între polinoamele rămase la concurenţă, un criteriu de selecţie poate fi stabilit prin

dezideratul de maximizare a acoperirii erorilor provocate de defectele mai probabile, impunând simularea logică pe bază de injecție de defecte.

Ceea ce este surprinzător, şi dovada o constituie şirul larg de referinţe bibliografice amintite, este că, aşa cum rezultă şi din structura BILBO din fig.3.1, intrările în arborele de circuite SAU EXCLUSIV din conexiunea de reacţie nu corespund unui polinom generator G(x) primitiv, cu toate că amplasarea circuitelor dovedeşte că aceasta s-ar fi dorit. În baza consideraţiilor expuse prin teorema 3.2 şi lema 3.2, utilizând expresia (3.1) pentru G(x), rezultă structura ASIM din fig.3.24, cu deosebirile de conectare faţă de cea din fig.3.1şi cu capacitate de detecţie , bazat pe cele anterior fundamentate, superioară.

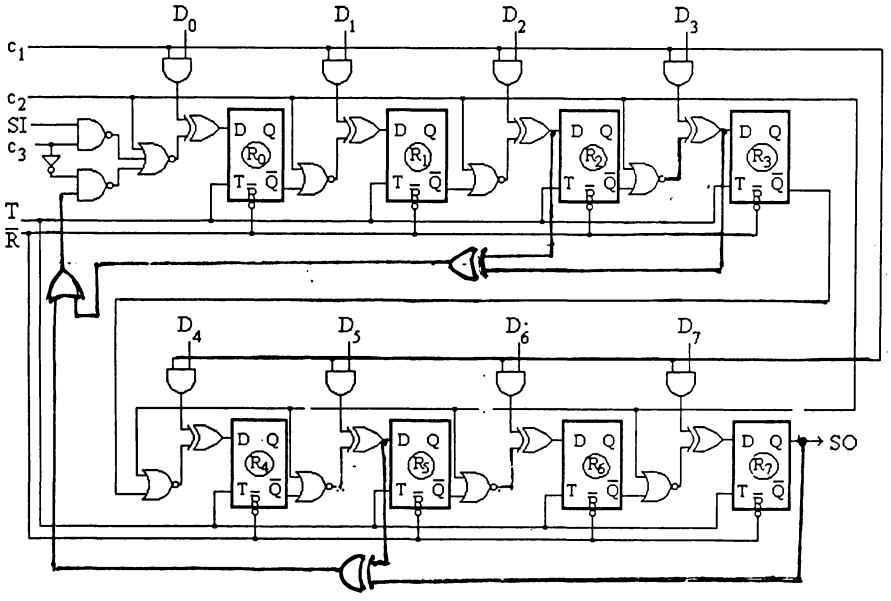


Fig. 3.24

Elaborând ecuaţia logică de tipul (3.22) pentru structura BILBO din fig.3.1, în regimul de analiză de semnături paralelă, avem corespunzător rangului R_0:

$$R_0(t+1)=D_0(t)+R_1(t)+R_2(t)+R_4(t)+R_7(t) \qquad (3.29)$$

În aceleaşi condiţii, pentru rangul R_0 al structurii ASIM din fig.3.24 avem:

$$R_0(t+1)=D_0(t)+R_1(t)+D_2(t)+R_2(t)+D_3(t)+R_4(t)+D_5(t)+R_7(t) \qquad (3.30)$$

Spre deosebire de sinteza bazată pe (3.30) efectuată prin polinomul primitiv (3.1), care permite generarea a (2^8-1) semnături distincte, cea realizată prin (3,.29), fiind bazată pe un G(x)

neprimitiv, permite generarea unui număr mai mic de semnături distincte, ceea ce este echivalent cu corespunzătoarea creştere ca aliasing-ului,adică a numărului de erori nedetectate şi, implicit, a anterior prezentatei probabilităţi ca măsură de contraperformanţă de testare. Reproşul care poate fi adus structurii ASIM este că, spre deosebire de schema din fig.3.1, la cea din fig.3.24 se introduc 2 niveluri logice (un circuit SAU -NU şi unul SAU EXCLUSIV) suplimentare în lanţul logic de generare a biţilor de reacţie. Ne îndoim însă că aceasta a constituit argumentul alegerii intrărilor în lanţul logic de reacţie, cu atât mai mult cu cât în majoritatea referinţelor bibliografice anterior citate, acesta este sintetizat cu o conexiune serială de circuite SAU EXCLUSIV. Ca să emitem o părere legat de omisiunea semnalată, care, dependent de aplicaţie, poate avea în anumite situaţii o influenţă negativă semnificativă, prezumţia noastră este că, la nivelul articolului origină, mecanismul de comprimare paralelă era insuficient aprofundat. Oricum, cele două niveluri logice se adaugă celor $\lfloor \log_2 m \rfloor$ anterior menţionate ale unei sinteze arborescente a lanţului logic SAU EXCLUSIV şi comparată strict prin prisma performanţei exprimată prin frecvenţa trenului de tact, schema din fig.3.24 este inferioară celei din fig.3.1. În acest punct, este însă momentul să fructificăm teoremele propuse. Astfel, bazându-ne pe teorema 3:1 corelată cu lema 3.1, rezultă că, folosind aceeaşi expresie (3.1) pentru polinomul generator G(x), se poate obţine o schemă echivalentă prin prisma capacităţii de detecţie cu cea sintetizată în fig.3.24, având însă o altă conectare a reacţiei şi anume la circuite SAU EXCLUSIV intercalate între rangurile registrului de deplasare aşa cum se prezintă în fig.3.25

Impunând analizei prin prisma performanţei noua structură ASIM, oricum superioară celei din fig.3.1 prin prisma capacităţii de detecţie din considerente deja justificate, se observă că aceasta implică intercalarea , corespunzător fiecărui termen dintre cei extremi din expresia lui G(x) dată de (3.1), a unui circuit SAU EXCLUSIV suplimentar, dar dispar în totalitate aceste circuite din lanţul de reacţie. Prin urmare, în această schemă intervin în calea accelerării frecvenţei trenului de tact doar trei niveluri logice şi aceasta independent de expresia polinomului G(x). Dacă se ia în considerare că schema logică inevitabilă de pe intrarea R_0, care se regăseşte şi la structura BILBO din fig.3.1, implică tot trei niveluri logice, concluzia superiorităţii schemei din fig.3.25 faţă de cele din fig.3.1 şi din fig.3.24 este imediată. Totuşi trebuie să remarcăm faptul că faţă de un singur nivel SAU EXCLUSIV cât prezintă schema logică de pe intrarea lui R_0, schemele intercalate între rangurile R_2 - R_3, R_4 -R_5 şi R_5 -R_6 cuprind două niveluri SAU EXCLUSIV înseriate. Dar oricând sinteza acestor scheme poate fi întreprinsă pe două niveluri logice ŞI-SAU (sau altă combinaţie), aşa cum se prezintă funcţia logică pentru schema intercalată, spre exemplu, între rangurile R_2 şi R_3:

$$D_{R3} = \overline{c_1}\,\overline{c_2}R_2\overline{R_7} + c_1\overline{R_2}R_7 + c_1D_3\overline{R_2}\,\overline{R_7} + \overline{D_3}R_2R_7 + c_1c_2D_3\overline{R_7} + \overline{c_1}c_2R_7 + c_2\overline{D_3}R_7 + \overline{c_2}\,\overline{D_3}R_2\overline{R_7} + c_1\overline{c_2}D_3R_2R_7 \qquad (3.31)$$

în care prin D_{R3} s-a notat funcţia logică în formă sumă logică de produse logice corespunzătoare intrării D a rangului R_3.

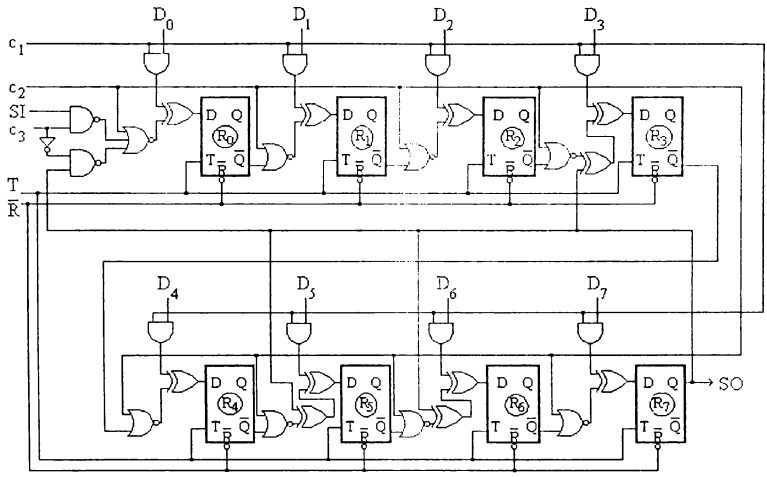


Fig.3.25

Complexitatea relației (3.31) nu trebuie să surprindă, pentru că, oricum, chiar și între rangurile ordinare, care nu au prevăzut circuitul SAU EXCLUSIV suplimentar, funcțiile logice în forma de implementare pe două niveluri rezultă complicate, așa cum poate fi constatat, spre exemplu, pentru schema intercalată între R_0 și R_1:

$$D_{R1}=\overline{c_1 c_2}\, R_0 + \overline{c_2} D_1 R_0 + c_1 c_2 D_1 + c_1 D_1 \overline{R_0} \qquad (3,32)$$

Oricum implementarea trebuie privită în contextul integrării pe scară largă și foarte largă, unde un rol important îl joacă nu atât complexitatea circuisticii (desigur și acesta are un aport), cât regularitatea structurii, sub care aspect cea din fig.3.25 suferă față de cele prezentate în fig.3.1 și fig.3.24. Ar mai fi un aspect care ar defavoriza schema din fig.3.25 față de concurentele ei, și anume întârzierea inegală dintre ranguri atunci când structura ASIM funcționează în regim de registru de deplasare, dar acesta este estompat prin faptul că frecvența trenului de tact este fixă și determinată de regimul cel mai defavorabil prin prisma cumulului de întârzieri pe niveluri logice, ori acesta este, la toate cele trei scheme în competiție, cel de analiză de semnături paralelă. Se poate conchide, că atunci când primează aspectele de performanță și de capacitate de detecție, noua schemă propusă și exemplificată în cazul particular al polinomului generator G(x) exprimat prin (3.1) este superioară celor publicate în literatura de specialitate.

Este de asemenea surprinzător faptul că și noua structură ASIM de tipul combinat prezintă, în anumite condiții, caracteristici asemănătoare ce cea prezentată anterior, fiind însă prin prisma performanței inferioară acesteia, dar putând fi superioară structurilor expuse în literatură. Astfel, în conformitate cu cele cuprinse în lema 3.3, ea nu se deosebește de competitoarele ei prin prisma capacității de detecție. Pentru a ilustra performanța cu referire la frecvența trenului de tact utilizat în procesul de comprimare, să exemplificăm această nouă structură ASIM plecând de la aceiași expresie primitivă (3.1) a polinomului generator G(x) (fig.3.26).După cum poate fi remarcat, de această dată calea de propagare a semnalelor cea mai defavorabilă implică cinci niveluri logice a căror întârziere

trebuie acoperită de perioada trenului de tact în regim de comprimare paralelă. Evident, această structură se dovedeşte inferioară celei din fig.3.25, dar în tot cazul superioară versiunii de sinteză a schemei din fig.3.1 aşa cum este prezentată majoritar în literatură, adică cu conctarea serială a circuitelor SAU EXCLUSIV din lanţul de reacţie. Dacă se apelează la o structură arborescentă pentru circuitele SAU EXCLUSIV, aşa cum este ea prezentată în fig.3.1, prin prisma performanţei exprimată de numărul de niveluri logice, cele două scheme sunt echivalente. Important este însă faptul că odată cu creşterea numărului de ranguri ale structurii ASIM aşa cum o solicită realizări actuale pe 16 biţi sau , mai degrabă, pe 32 sau 64 de biţi, aceasta conduce la creşterea numărului de termeni corespunzători polinomului primitiv G(x), drept consecinţă numărul de niveluri de circuite SAU EXCLUSIV creşte la variantele adaptate ale structurilor din fig.3.1 şi din fig.3.24, pe când extrapolarea structurii din fig.3.26 la un număr oricât de mare de ranguri menţine aceeaşi valoare 5 pentru nivelurile logice, astfel încât, pentru structurile ASIM de interes, se poate conchide că şi această nouă structură propusă este superioară celor prezentate în literatură. Evident, această afirmaţie este valabilă pentru acele aplicaţii de calcul la care primează performanţa şi capacitatea de detecţie a potenţialelor defecte.

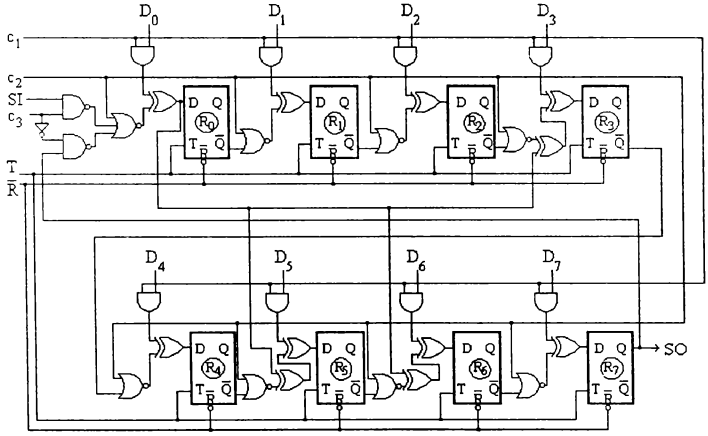


Fig. 3.26

Revenind la strategia de valorificare a multiplelor moduri de funcţionare care au fost prevăzute la o structură BILBO, respectiv la o structură ASIM în fig.3.27 este sugerată utilizarea acesteia în regim de autotestare a unei structuri de calcul imaginată, aşa cum se practică uneori în literatură, ca o secvenţă de perechi registru-reţeaua de logică combinaţională comandată de acesta. În fig.3.27, structura de calcul este redusă la două reţele, A şi B, de logică combinaţională, dar registrele sunt constituite prin două structuri BILBO, identice constructiv, dar care pentru identificare în scop de descriere au fost notate cu (1) şi (2).Regimul de testare este prevăzut a se desfăşura în trepte în sensul că, într-o primă instanţă, se efectuează verificarea reţelei logice A (fig.3.27). În acest sens, prin program, BILBO (1) este configurat ca generator de secvenţe pseudoaleatoare (pseudo-random pattern generator, PRPG), iar BILBO (2) este configurat ca analizor de semnături paralel (parallel signature analyzer, PSA). Astfel, BILBO (1) furnizează fluxul

de vectori binari de stimulare pentru rețeaua combinațională A, iar BILBO (2) asigură comprimarea paralelă a răspunsurilor. În finalul acestei etape de verificare

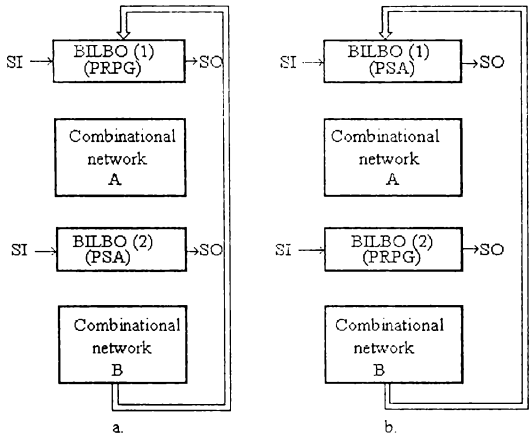


Fig.3.27

BILBO (2) este, din nou, reconfigurat prin programarea corespunzătoare a semnalelor de control (c_1,c_2,c_3), astfel încât trece în regim de registru de deplasare permițând investigarea prin punctul S0 a semnăturii obținute prin compararea acesteia cu una corespunzătoare funcționării normale și conținută într-un dicționar de semnături. Într-o a doua instanță, se trece la o a doua treaptă de testare, prin, în cazul prezentat, programarea interschimbării regimurilor structurilor BILBO, în sensul că BILBO (1) este configurat ca PSA și apoi ca registru de deplasare, iar BILBO (2) este configurat ca PRPG, asigurând, în aceeași manieră deja descrisă, verificarea acum a rețelei combinaționale B. Procedând în modul gradual descris simplificat prin intermediul fig.3.27, poate fi imaginată verificarea circuisticii unei structuri de calcul bazat pe reconfigurarea structurilor BILBO, respectiv pe reconfigurarea structurilor ASIM.

În ceea ce privește obiectivul constând din soluții pentru eficientizarea implementării toleranței la defectare, propunem valorificarea structurilorASIM propuse în acest capitol, vizând îmbunătățirea parametrilor de performanță și testare, într-un sens diferit de cel convențional sintetizat în fig.3.27. Ideea constă în adăugarea la modulele hardware standard aparținând bibliotecii asociate unui anume tip de magistrală a unui modul slave, distinct sau aglomerat pe modulul de supraveghere și control prevăzut pentru toate magistralele consacrate [Tane-90]. Menționăm că, într-un anume fel, ideea este conținută și în [MaMe-82]-fig.3.28, dar și în [WiSh-88], [SeSh-87] , unde însă se prevede utilizarea unui anumit procesor watchdog în scopul detecției concurente a erorilor potențiale.

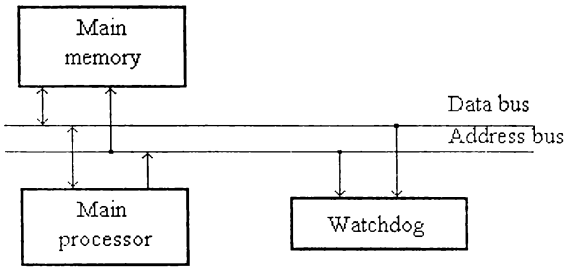


Fig.3.28

În plus, arătăm că magistralele consacrate,cu precădere, acelea destinate aplicațiilor de timp real, şi ne referim în acest context la magistrala VME, Multibus şi Multibus II, permit participarea la traficul de transfer de date pe lângă convenționala pereche bus master-slave şi a unor dispozitive de supraveghere, cum ar fi aşa numitul monitor şi aşa numitul watchdog timer [Tane-90]. Primul dintre acestea permite monitorizarea fluxului informațional prin intermediul unui analizor logic care poate fi ataşat la magistrală. Aceasta permite afişarea unui număr determinat de referiri care devansează sau succed unui cuvânt reperat ca suspect şi utilizat în calitate de cuvânt de declanşare. Al doilea, permite urmărirea anumitor semnale de control (spre exemplu, Data Acknowledge, $\overline{\text{DTACK}}$ la VME) în mod continuu şi, atunci când un ciclu de magistrală durează prea mult (de exemplu, atunci când se face referire la o adresă de memorie inexistentă şi slave-ul nu răspunde), dispozitivul watchdog timer trimite un semnal de eroare care să termine ciclul de bus, facilitate în absența căreia o magistrală asincronă poate rămâne "agățată". Dispozitivele reclamate de implementarea funcțiilor de urmărire menționate sunt implementate, în mod uzual, pe un modul hardware de supraveghere şi control care este în mod opțional, inclus într-o anumită configurație de calcul. În general, această placă mai conține circuistica necesară generării semnalelor de control de întrerupere la sesizarea căderii tensiunii de alimentare ($\overline{\text{ACFAIL}}$ - AC power failure detected la VME), de inițializare a sistemului ($\overline{\text{SYSRESET}}$ -System reset la VME), precum şi de sesizare a unei anomalii funcționale ($\overline{\text{SYSFAIL}}$ -Diagnostic line la VME). Pe o astfel de placă, sau pe una separată, am prevăzut extinderea funcțiilor de supraveghere menționată prin circuistica de implementare a unei structuri ASIM destinată creşterii capacității de trecere a programelor de autotestare,aşa cum va fi dezvoltată la capitolele 5 şi 6.

3.4. Concluzii

Urmărind găsirea unor soluții pentru eficientizarea implementării toleranței la defectare în sisteme de calcul în acest capitol au fost investigate structurile BILBO, cunoscute realizări practice care încorporează strategii de built-in test.Cercetările întreprinse au condus, pe de o parte, la critica soluțiilor oferite de literatură şi pe de altă parte, la propunerea a trei structuri moi diferite de cele BILBO şi denumite structuri ASIM (Analizor de Semnături cu Intrări Multiple).

Rezumând rezultatele obținute, considerăm drept contribuții următoarele:

a) Propunerea structurii ASIM cu circuite SAU EXCLUSIV conectate suplimentar între rangurile registrului de deplasare, implicând:

a_1) Formularea teoremei originale 3.1 numită a permite determinarea capacității de detecție a erorilor specifică acestei structuri.

a_2) Elaborarea a trei proceduri originale de soluționare asistată de calculator, bazat pe matrici de control asociate acestei structuri ASIM, a determinării semnăturilor etalon corespunzătoare fluxurilor binare testate.

a_3) Stabilirea prin lema originală 3.1 a probabilității statistice de recunoaştere ca funcțional corectă a unei unități testate corecte specifică comprimării prin această structură ASIM.

b)Propunerea unei structuri ASIM cu circuite SAU EXCLUSIV conectate în exteriorul registrului de deplasare, implicând:

b_1)Formularea teoremei 3.2 menită a permite determinarea capacității de detecție a erorilor specifică acestei structuri precum şi compararea, prin această prismă, cu structura ASIM anterioară.

b_2)Elaborarea a trei proceduri originale de soluționare asistată de calculatăr, bazat pe matrici de control asociate acestei structuri ASIM, a determinării semnăturilor etalon corespunzătoare fluxurilor informaționale binare testate.

b_3) Stabilirea prin lema originală 3.2 a probabilității statistice de recunoaştere ca funcțional corectă a unei unități testate corecte specifică comprimării prin această structură ASIM.

c) Propunerea unei structuri ASIM combinate, implicând:

c_1) Formularea teoremei 3.3 menite a permite determinarea capacității de detecție a erorilor specifică acestei structuri precum şi compararea, prin această prismă, cu structura ASIM anterioară.

c_2) Elaborarea a trei proceduri originale de soluționare asistată de calculator, bazat pe matrici de control asociate acestei structuri ASIM a determinării semnăturilor etalon corespunzătoare fluxurilor binare testate.

c_3) Stabilirea prin lema originală 3.3 a probabilității statistice de recunoaştere ca funcțional corectă a unei unități testate corecte specifică comprimării prin această structură ASIM.

d) Realizarea unei analize comparative prin prisma performanță (exprimată prin frecvența trenului de tact destinat comprimării)/capacitate de detecție (exprimată prin probabilitatea statistică şi reală de recunoaştere ca funcțional corectă a unei unități testate defecte) între structurile ASIM propuse. În acest context, menționăm că, sunt diverse aspecte, parametri ai structurilor propuse se dovedesc superiori unora similari specifici structurilor BILBO referite în repere bibliografice de marcă (precizate, in extenso, în text).

4. Facilitarea testării prin structuri ASIM modificate.

În prezentul capitol, ca de altfel şi în proximele două, ne propunem să prezentă, diferite aplicaţii ale structurilor ASIM modificate în capitolul 3 faţă de cele convenţionale şi care, prin capacitatea de detecţie superioară demonstrată, oferă eficientizarea utilizării lor. Dacă în cele ce urmează sunt propuse, investigând toate componentele hardware ale unui sistem de calcul, includeri ale structurilor ASIM în configuraţii de calcul cu toleranţă la defectare sau cel puţin autotestabile care solicită imperios un nivel tehnologic ridicat [AbAb-94,BBKP-94,LeSh-94,ReJh-94,LoTr-93], în capitolele 5 şi 6 sunt expuse implementări, de această dată practice, ale acelor structuri ASIM în realizări de nivel tehnologic puţin pretenţios, disponibil [Vasi-88a,Vasi-89c].

Astfel, în acest capitol am selectat ca potenţiale aplicaţii eficiente schemele secvenţiale sincrone, schemele logice programabile tip PLA (Programable Logic Array) şi FPLA(Field PLA)-cu extensii în schemele cu arii de porţi tip GA(Gate Arrays) şi FPGA(Field Programable GA)-, şi, în ultimă instanţă, schemele unităţii aritmetice şi logice ALU(Arithmetic Logic Unit).

Privitor la schemele convenţionale sincrone, în urma relevării a noi aspecte legate de eterna problemă a dificultăţilor testării acestora [HKQS-94,LeLo-90],sunt supuse analizei variate tehnici de scanare [APSD-94,NaBH-94,RaTy.93,LeSa-90,AbBr-86,SaDa-86] în scopul selectării acelora care la combinaţie cu noile structuri ASIM să permită eficientizarea facilitării testării schemelor secvenţiale sincrone. Succesiv, sunt scoase în relief problemele legate de testarea schemelor PLA, respectiv a versiunilor evoluate ale acestora [WuIv-95,HISS-94,KaLV-94,MFGH-94,TrJe-94,AnSa-88,CrKS-88,ReSK-88], în scopul intercalării redundante a structurilor ASIM modificate pentru a rezulta eficiente scheme cu autotestare încorporate BIST(Built-In Self Test). În fine, referitor la schemele UAL sunt analizate unele configuraţii de sumatoare şi dispozitive de înmulţire, cu autocontrolul bazat pe coduri redundante [BlBT-94,BhSe-93,Heid-91,KaNa-90,SeAF-90,Krol-86,SaRo-86], prin prisma capacităţii de detecţie a erorilor şi de corecţie a erorii singulare.În mod firesc, accentuk este pus pe codurile bazate pe puritate [ViRa-94,RaFu-89,Prad-86] a căror generalizare într-un anume sens [ClWe-94,Vasi-94,KaNa-90] poate fi considerată comprimarea prin analiză de senbături cu intrări multiple.De asemenea, este investigat locul optim de amplasare a noilor structuri ASIM în scheme UAL destinate adunării/scăderii şi înmulţirii cablate pentru a permite creşterea eficienţei autocontrolului şi, prin aceasta, implementarea favorabilă a toleranţei la defectare.

4.1.Utilizarea structurilor ASIM modificate în conjuncţie cu tehnicile de scanare pentru facilitarea testării schemelor secvenţiale sincrone

4.1.1.Unele aspecte ale testării schemelor secvenţiale sincrone

Motivaţia preocupărilor multiple pentru facilitatea testării structurilor secvenţiale sincrone o reprezintă dificultatea procesului de elaborare a vectorilor binari de simulare, cauzată de faptul că funcţiile logice, urmărite prin vectoriide ieşire din puncte de observabilitate, depind atât de

configurațiile binare aplicate la intrările primare, cât și de stările interne determinate de valorile logice stocate în elementele de memorare din buclele de reacție. Devine necesară setarea mașinii secvențiale în fiecare stare internă, precum și valorivicarea tuturor tranzițiilor dintre stări, investigare care, la amploarea actualelor scheme, implică un efort de elaborare și de calcul excesiv de mare, în tot cazul substanțial majorat în raport cu unul reclamat de o schemă comparabilă cu număr de circuite elementare, dar de tip combinațional. Acestui aspect i se adaugă solicitarea unor resurse de memorare de capacitate excesiv de mare pentru stocarea voluminoaselor secvențe de testare rezultate.Mai intervine, în anumite situații, necesitatea unei stări interne, uzual denumită inițială, din care să poată demara procesul de verificare propriu-zis. Este adevărat însă,că majoritatea schemelor practice dispun de facilități hardware pentru inițializare, a căror activitate permite cunoașterea stării interne de debut a experimentului de testare [FrSa-94,John-93,MoAl-89,MaCl-88,Fuji-85].

Limitând considerațiile la mașinile secvențiale sincrone-cele asincrone sunt abordate prin metode specifice [ChAK-90]-, remarcăm faptul că subiectul testării acestora a stârnit preocupări intense, cu precădere în deceniul '70-80', interesul începând să scadă de pe la jumătatea deceniului următor datorită proceselor înregistrate în domeniul sintezei prevăzute cu facilități de testabilitate (design for testability) [Gork-91,Gork-89,John-89,Wojt-88,Prad-86,Lala-85]. Cauza acestui parcurs trebuie legată și de faptul că, odată cu creșterea în complexitate a schemelor, procedurile de testare asistată de calculator a generării experimentelor de testare au fost supuse unor restricții reclamate de valorile excesiv de mari ale parametrilor de soluționare reprezentați, în special, de efortul de elaborare și timpul de calcul, dar și de capacitatea de memorie. Aceste restricții vizau aspecte generale constând din 1)limitarea elementelor de memorare, 2)limitarea numărului de porți, 3)limitarea numărului de vectori per defect, 4) neglijarea întârzierilor schemei, dar și unele de amănunt, care, în sinteză, se referă la: 1)ignorarea defectelor datorate hazardurilor și oscilațiilor, fiind admisă liniștirea fenomenelor tranzitorii la momentele de strobare prin impulsurile de tact; 2)mașina secvențială sincronă se consideră cu conectare strictă (strongly connected), caracterizată prin existența cel puțin a unei secvențe de vectori binari, care aplicată la intrările primare, să determine tranziția schemei dintr-o anumită stare internă în oricare altă stare internă; 3)evenimentul de defectare nu determină apariția unei stări interne noi, diferită de cele corespunzătoare mașinii sincrone cu funcționare normală. Aceste din urmă condiții, admise în mod ipotetic de către majoritatea procedurilor, nu sunt întotdeauna confirmate în practică, rămânând neacoperite defecte, uneori, esențiale.[HaCl-93,ChAg-90,GuGb-90,LeSa-90].

Încercând o ordonare în prolifica familie a procedurilor de elaborare a experimentelor de testare pentru mașini secvențiale sincrone în tendința relevării deficiențelor care justifică aderarea la metodele DFT (design for testability), se detașează, mai întâi, subfamilia de proceduri de identificare mașinii (machine identification), având ca reprezentative metodele Hennie , Hsieh și Boute [LeYa-94,Prad-86]. Aceste proceduri prevăd căutarea acelei secvențe de vectori binari numită experiment de verificare (checking experiment), care aplicată intrărilor primare, să permită distingerea, prin urmărirea vectorilor răspuns de la ieșirile observabile, dintre mașina cu funcționare normală și oricare versiune a acesteia generată prin defectare singulară sau multiplă, dar supusă condițiilor restrictive

amintite anterior. Investigarea se efectuează în descrierea prin tabele de stări a maşinii secvenţiale normale şi, în final, se obţine secvenţa de testare formată, în cazul general, din trei subsecvenţe. Prima, denumită subsecvenţă de readucere (homing subsequence), este destinată recunoaşterii stării interne în care ajunge schema în vederea debutului procesului de verificare efectiv, ea putând lipsi când maşina este prevăzută cu facilităţi hardware pentru iniţializare.Când însă este necesară apelarea la homing sequence, experimentul de testare devine unul de tip adaptiv, fiind dependent de starea internă particulară în care ajunge maşina. Procesul de verificare propiu-zis implică două subsecvenţe, anume cea de identificare a tuturor stărilor interne şi cea de verificare a tuturor tranziţiilor dintre aceste stări. Prima dintre acestea se bazează pe prealabila căutare a aşa numitei subsecvenţe de distingere (distinguishing subsequence) care, în cazul maşinii secvenţiale ce o posedă, permite recunoaşterea, din şirul vectorilor binari de la ieşirile observabile, a stărilor interne.În acest sens, prin subsecvenţe de transfer (transfer subsequence) adecvate, maşina este adusă în fiecare stare internă, după care este stimulată prin distinguisching subsequence până la baleierea exhaustivă a tuturor stărilor interne corespunzătoare maşinii normale. Dacă însă, maşina secvenţială nu posedă distinguisching subseguence, identificarea stărilor interne este complicată, necesitând căutarea mai laboriasă a aşa numitelor subsecvenţe de caracterizare (characterizing subsequence). În fine, cea de-a doua subsecvenţă a experimentului de testare, cea destinată verificării tranziţiilor corespunzătoare maşinii normale, implică aducerea maşinii în fiecare stare internă de un număr de ori egal cu cel al vectorilor de intrare posibili, astfel încât să poată fi urmărită reacţia provocată la ieşirile observabile de aplicarea fiecăruia dintre aceştia. Întregul proces implică multiple procedee de căutare prin arbori şi se traduce prin experimente de testare de dimensiuni mari, gravând asupra productivităţii procesului de verificare [NaAb-90, SaCl-90].

O a doua subfamilie, cu un impact practic superior, cuprinde procedeele adaptate pentru maşini secvenţiale ale unora corespunzătoare schemelor logice combinaţionale, reprezentative în acest sens fiind adaptările algoritmului D, a metodei PODEM (Path Oriented Decision Making), a metodei Poage-McCluskeyş.a.[ClWe-94,Toka-94,Gork-91,ChAK-90,John-89,Lala-85,RKCl-85]Urmărind reliefarea specificului esenţial al acestei subfamilii de proceduri, precum şi a dificultăţilor care le întâmpină, vom considera , fără a pierde din generalitate, schema secvenţială asincronă simplă din fig.4.1, în care admitem defectul de blocare la 1 a liniei c, precum şi ipoteza unei întârzieri nule pe porţile schemei. Apelând la convenţionalul concept de activare a unei căi (conventional path sensitiyation approach) pentru determinarea vectorului de stimulare relativ la "c blocat pe 1" (c stuck- at 1), aceasta implică traversarea celor trei paşi marcaţi pe figură: !) se impune pe c valoarea logică complementară defectului, deci c=0, dar care 2) implică a=1 şi d=1, dar pentru ca d=1 este necesar ca 3)b=0.

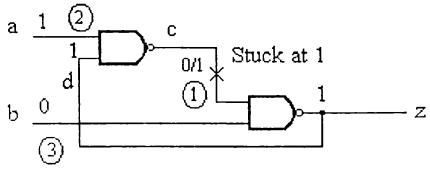


Fig.4.1

Se poate remarca însă că procesul se blochează la vectorul de intrare ab=10 şi ieşirea z=1, defectul neputând fi detectat la ieşirea observabilă z, aceasta indicând valoarea 1 atât în prezența, cât şi în absența defectului considerat. Efectul acestuia din urmă în blocarea căii de reacție, transformând schema secvențială într-una fără capacitate de memorare, în fond $z=\overline{b}$. Se poate însă observa cu uşurință că, succedând vectorul de intrare ab=10 cu vectorul ab=11, schema normală va răspunde la z prin secvența 11, iar cea defectă prin 10, astfel încât defectul poate fi detectat. Pentru a determina această secvență, procedurile adoptă metoda comună de întrerupere a căii de reacție (cut the feedback path) prin acceptarea unei întârzieri finite pe porți, care, la exemplul considerat, face posibilă aplicarea eşalonată în timp, mai întâi, a lui z=1 la d şi, apoi, modificarea valorii de la b din 0 în 1, activând poarta din nivelul al doilea şi observând efectul propagat al defectului considerat. Astfel, în fig.4.2 este ataşată o copie a schemei care să permită aplicarea valorii logice pe calea de reacție d. Elaborarea secvenței de testare începe de la copia de schemă corespunzătoare eşantionului de timp curent (time current frame) şi, în general, poate fi adăugat orice număr de eşantioane de timp (anterioare şi viitoare) pe fiecare parte a celui curent. Pentru exemplul considerat a fost necesară expandarea în timp cu un singur eşantion anterior (previous time frame-fig.4.2).Tocmai în această creştere în complexitate a descrierii schemei rezidă una din dificultățile acestei subfamilii de proceduri.[NiNa-93,BKSS-90,ChAK-90].

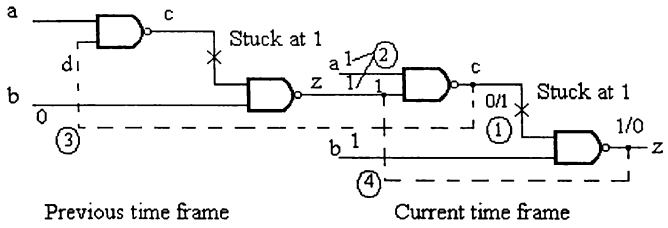


Fig.4.2

O a doua dificultate constă în timpul suplimenr de procesare reclamat de evitarea unor posibile probleme de timing care pot fi provocate de valorile logice nespecificate de către programul generator de vectori stimuli.Particularizând acest aspect la cazul considerat, se poate remarca faptul că vectorul de stimulare care îl precede pe ab=11, este X0 (fig.4.2), unde X denotă o valoare logică indiferentă (don't care). Adoptând X=1, se obține secvența de testare dorită, dar dacă am adopta X=0, aceasta determină o condiție de hazard (race) pentru schemă, implicând modificarea ambelor valori binare corespunzătoare celor două intrări. Se poate conchide că neglijarea întârzierilor care

face imposibilă deducția secvențelor de testare pentru anumite defecte, expandările în timp cu mai multe eşantioane, precum şi analizele condițiilor de race care reprezintă dificultăți care limitează, cu precădere relativ la actualele scheme practice, aplicarea acestei categorii de proceduri.

În fine, a treia subfamilie de metode se bazează pe simulare şi căutare direcționată a secvenței de testare, având ca reprezentative metodele SOFTG (Simulator Oriented Fault Test Generator) şi TVSET (Threshold Value Sequence Test Generation Program) [ChAK-90,RKCl-85]].Utilizarea simulării permite înserarea de întârzieri astfel încât vectorii ar putea produce hazarduri şi oscilații pot fi omişi. Detaliind, într-o anumită măsură, modul în care este condusă căutarea în procesul de simulare, ne vom referi la procedura TVSET [ChAK-90] care utilizează o funcție de cost, definită astfel încât să determine cât de "departe" este un anumit vector de detectarea unui defect. Funcția de cost este calculată pentru un vector de intrare şi un set de defecte, ea depinzând doar de rezultatele simulării schemei normale şi celei defecte.Ea prezintă valori cu atât mai mici, cu cât un vector de intrare este mai aproape să devină un stimul, iar dacă acesta subdepăşeşte o valoare constantă de prag, atunci vectorul respectiv permite detecția defectului. Începutul procedurii poate fi făcut cu un vector arbitrar, schema fiind supusă unui proces de simulare şi este calculată funcția de cost pentru toți vectorii situați la distanța Hamming egală cu unitatea în raport cu vectorul adoptat, urmând a fi selectat vectorul căruia îi corespunde funcția de cost cu valoarea minimă. Prin iterări succesive, fie se ajunge ca un vector să devină stimul, fie se obține pentru funcția de cost o valoare minimă, dar care nu subdepăşeşte pe cea de prag, situație în care procesul de căutare poate fi restartat cu un nou vector inițial. Procesul implică deci treceri multiple prin procedură, fiecare dintre acestea fiind executată conform cu cele ilustrate în fig.4.3.[ChAK-90].

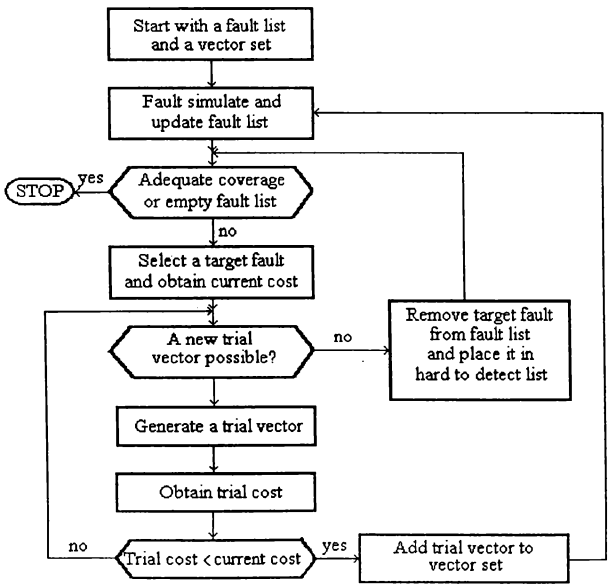


Fig 4.3

Se debutează cu o listă de defecte (fault list), care poate conține toate defectele de blocare, şi un set de vectori (vector set), care poate fi vid sau conține, în mod opțional, vectori de inițializare sau anumiți vectori de verificare funcțională. În situația că setul este vid, acestuia i se alocă un vector generat în mod aleator. La următoarele treceri, lista defectelor va conține pe acelea găsite ca dificil de detectat (hard to detect) până la ultima trecere, iar setul de vectori va conține pe cei generați până în acel moment. La începutul unei treceri sunt simulate toate defectele din lista actualizată prin înlăturarea celor detectate. Apoi sunt calculate funcțiile de cost pentru defectele nedetectate de către ultimul vector al setului şi este selectat defectul cu costul cel mai mic în calitate de defect țintă (target fault), costul corespunzător acestuia fiind denumit curent (current cost). Pentru a determina următorul vector, ar trebui calculate costurile tuturor vectorilor posibili şi apoi să se selecteze cel căruia îi corespunde valoarea minimă a funcției de cost, iar dacă aceasta este mai mare decât cea curentă se spune că s-a ajuns la un cost minim local. În cazul general, numărul vectorilor de încercare (trial vectors) posibili din fiecare trecere este 2^n-1 pentru n intrări primare, totuşi procedura prevede două strategii de reducere a calculelor. În primul rând, sunt considerați doar acei vectori de încercare cu distanța Hamming diferită cu o unitate față de ultimul vector, şi, în al doilea rând, în loc de a calcula costurile corespunzătoare tuturor vectorilor de încercare, în mod euristic este acceptat primul vector de încercare având costul mai mic decât cel al ultimului vector. Dacă nici unul dintre vectorii de încercare nu îndeplineşte condiția, defectul este înlăturat din lista de defecte şi este atribuit listei celor dificil de detectat (hard todetect) pentru a fi considerat în următoarele treceri.De altfel, o anumită trecere poate conduce la una din următoarele două condiții: 1)a fost obținută o acoperire adecvată a defectelor (adeguate fault coverage) ,sau 2)fiecare defect, care inițial a fost în lista defectelor este fie detectat, fie este atribuit listei celor dificil de detectat, astfel încât lista defectelor este vidată. Este de remarcat din ordinogramă (fig.4.3) că atunci când un vector de încercare este adăugat setului de vectori, procesul de selecție a defectului țintă, adică a celui cu costul mai mic, se se repetă. Dacă noului vector îi corespunde un cost mai mic relativ la defectul țintă,dar nu îl detectează pe acesta, este posibilă selectarea unui alt defect țintă. Se poate conchide că procedura TVSET, ale cărei rezultate cu privire la gradul de acoperire al defectelor potențiale se prezintă în raport cu alte proceduri [ChAK-.90], se limitează totuşi la scheme secvențiale reduse ca anvergură şi este total dependentă de modul în care este definită funcția de cost de prag, precum şi de criteriile euristice admise în procesul de selecție, aspecte care pot comporta discuții.

Fără a emite pretenția relevării întregii problematici cu care se confruntă generarea experimentelor de testare pentru scheme secvențiale, în particular sincrone, şi fără a preciza întreaga gamă de proceduri, prezentarea de mai sus pune în relief aspecte noi legate de testarea acestor scheme constituindu-se într-o pledoarie care argumentează proliferarea procedurilor, cu precădere din ultimul deceniu, pentru facilitarea testării, cele bazate pe tehnici de scanare reprezentând obiectivul analizei noastre în cele ce urmează.

4.1.2.Tehnici de scanare cu acces asigurat prin intervenţii asupra elementelor de memorare

4.1.2.1.Tehnica de scanare Stanford

Cu o larghețe mai mare sau mai mică,se poate considera că tehnicile de scanare pentru facilitarea schemelor secvențiale sincrone îşi au originea în lucrarea de doctorat susținută de M.Wiliams la Universitatea Stanford, de unde şi numele metodei, având cercetările sintetizate în [AnWi-73]. În esență, ideea pe care se fundamentează această metodă, constă în prevederea pentru maşina secvențială a două moduri de operare, cel de funcționare normală căruia i se adaugă unul destinat testării. În vederea implementării acestuia din urmă, schema convențională este supusă unor modificări prin adaus de logică şi de puncte de control care să permită interconectarea elementelor de memorare într-un registru de deplasare prin intermediul căruia se asigură accesul la puncte interne ale schemei, mărindu-se în acest mod controlabilitatea şi observabilitatea acesteia. Pentru a da concretețe ideii, să considerăm structura generală de maşină secvențială sincronă din fig.4.4, în care am admis, fără a pierde din generalitate, elemente de memorare de tip D. Partea de logică combinațională este controlată prin vectorul binar aplicat la intrările primare x_1, x_2,...,x_n, dar şi prin vectorul binar provenit de la elementele de memorare amplasate pe căile de reacție şi aplicat la intrările Y_1, Y_2,....,Y_s,care reprezintă starea internă curentă a maşinii secvențiale. Aceeaşi logică combinațională răspunde la stimuli aplicați la cele două categorii de intrări prin vectorii binari furnizați, pe de o parte, la ieşirile observabile z_1, z_2,...,z_m,şi pe de altă parte, la ieşirile corespunzătoare căilor de reacție y_1,y_2,...,y_s, care reprezintă starea internă următoare a maşinii secvențiale şi care este încărcată în elementele de memorare la aplicarea unui tren de clock comun pe la intrarea primară CK. Legat de elementele de memorare, care mai au prevăzută linia comună de inițializare la care se aplică semnalul $\overline{R}$ (de resetare), menționăm că acestea pot fi cu comutare pe front (edge-triggered) pentru care vom folosi termenul de flip-flop, sau pot fi cu comutare pe nivel, pentru care vom folosi termenul de latch. Diferența constructivă dintre cele două tipuri de elemente de memorare are implicații majore în prezentarea următoare, fiind cunoscut faptul că flip-flop-urile permit construcția unui registru de deplasare prin simpla conectare a ieşirii unui rang la intrarea de date a următorului, pe când conversia unui registru de latch-uri într-un registru de deplasare necesită adăugarea la fiecare rang a unui latch suplimentar.

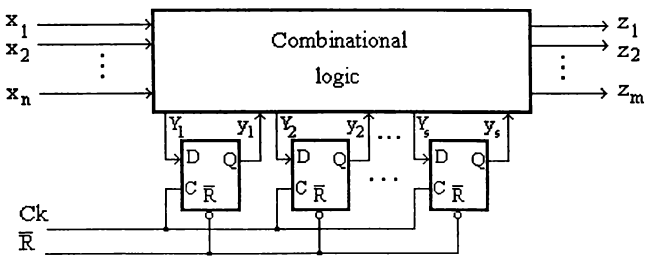


Fig.4.4

Admiţând că la structura generală din fig.4.4 sunt utilizate flip-flop-uri, modificarea propusă în [AnWi-73] poate fi urmărită în fig.4.5. Ea constă în adausul a (s+1) scheme de multiplexare (MUX), unde s reprezintă numărul de flip-flop-uri, de construcţie identică şi care, într-o variantă de implementare cu porţi ŞI-NU, este prezentată în fig.4.6.

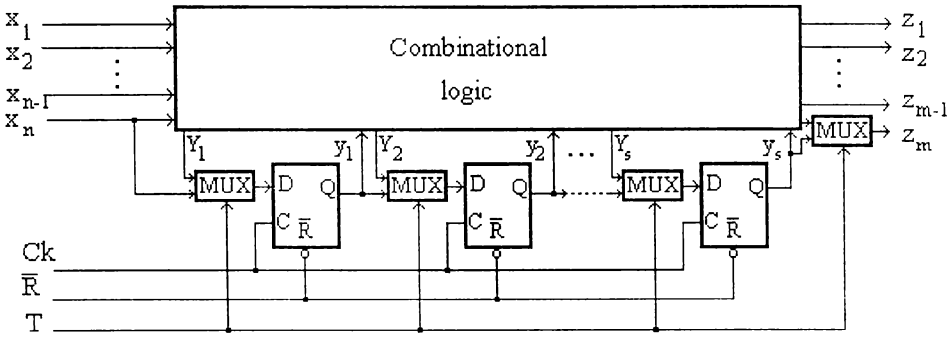


Fig.4.5

După cum poate fi remarcat cu uşurinţă, aceste MUX-uri permit, sub controlul variabilei logice aplicată la intrarea primară T şi aceasta adăugată structurii, selecţia a două căi de date diferite. Atunci când la intarea T se aplică 0, funcţionarea schemei din fig.4.5 se rezumă la cea normală corespunzătoare schemei din fig.4.4, iar atunci când la T se aplică 1, funcţionarea schemei din fig.4.5 corespunde modului de operare de testare în care flip-flop-urile sunt înlănţuite într-un registru de deplasare. Această facilitate permite verificarea schemei printr-o strategie eşalonată în timp, care prevede testarea, mai întâi, a lanţului de flip-flop-uri. În acest sens, uzitând de una din intrările primare, spre exemplu de x_n, şi de una din ieşirile observabile, spre exemplu de z_m, în lanţul de flip- flop-uri sunt încărcate serial, pe la x_n, configuraţii binare- constând din şiruri de s 0-uri şi de s 1-uri, apoi din alternanţe de 0-uri cu 1-uri, din şiruri care "plimbă" un 0 în câmp de 1-uri, respectiv un 1 în câmp de 0-uri,ş.a-care, după s impulsuri de clock, pot fi observaţe la z_m, şi, prin comparare cu unele aşteptate, stabilesc verdictul funcţionalităţii corecte a flip-flop-urilor. În cazul că această etapă de verificare se soldează cu succes, ea este urmată de verificarea părţii de logică combinaţională, care se efectuează prin iterarea celor 5 paşi descrişi în continuare de un număr de ori egal cu cel al vectorilor de stimulare necesari verificării logicii combinaţionale justificând sintagma conform căreia prin această strategie DFT problematica testării maşinilor secvenţiale sincrone se reduce la una specifică reţelelor combinaţionale.[WiJR-94,ChAg-90,GuGB-90].

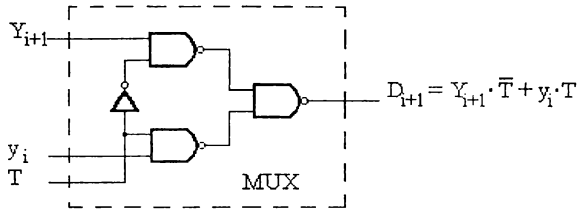


Fig.4.6

1. Se setează T pe 1 prin care se trece în modul de scanare.

2. Se deplasează prin s impulsuri de clock în lanțul de flip-flop-uri acel subvector binar y_1, y_2,...,y_s care împreună cu subvectorul binar x_1, x_2, ...,x_n- acesta din urmă poate fi aplicat pe durata ultimului tact, dacă y_1 coincide logic cu x_n, în caz contrar fiind necesară aplicarea respectivului subvector binar succesivă celor s impulsuri de clock-formează unul dintre vectorii binari de stimulare ai logicii combinaționale. Este de remarcat faptul că aceasta din urmă nu este stimulată doar prin vectorii binari de dimensiune n disponibili la cele n intrări primare, ci prin vectori de stimulare de dimensiune (n+s) ceea ce în conformitate cu fundamente ale testării combinaționale, le conferă o corespunzător mai mare controlabilitate, tradusă printr-o capacitate de acoperire a unui număr mai mare de defecte potențiale, și având, drept consecință, reducerea numărului total al vectorilor stimulare necesari.

3. Se setează T pe 0, revenindu-se astfel la modul de funcționare normal în care părții de logică combinațională i se aplică vectorul de stimulare extins (n+s), și, după un interval de timp care să acopere cazul cel mai defavorabil de propagare al semnalelor prin logică combinațională, se verifică vectorul binar răspuns disponibil la ieșirile observabile z_1,z_2,...,z_m prin comparare cu unul memorat, corespunzător funcționării fără defecte în aceleași condiții de stimulare.

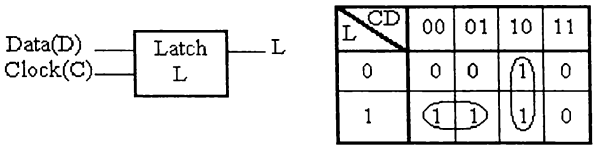
4. Se aplică la CK un impuls de clock prin care vectorul binar răspuns disponibil la ieșirile y_1,y_2,...,y_s, corespunzător stării interne următoare, este încărcat în cele s flip flop-uri de pe lanțurile de reacție ale schemei.

5. Se setează T pe 1 revenindu-se astfel la modul de scanare în care, prin alte s impulsuri de clock, este investigată starea internă a mașinii secvențiale. Prin această procedură, are loc , pe lângă extensia de la n la (n+s) a vectorilor de stimulare, și extensia de la m la (m+s) a vectorilor răspuns, ceea ce, în baza aceleiași fundamentări ale testării combinaționale, determină creșterea observabilității cu aceleași consecințe benefice, în ultima instanță, constând din reducerea vectorilor de stimulare și, prin urmare, din reducerea timpului revendicat de testarea părții de logică combinațională. În acest context, se cuvine menționat că odată cu descărcarea serială, pe la ieșirea z_m, a subvectorului răspuns de dimensiune s cu unul memorat este posibilă suprapunerea încărcării seriale a subvectorului stimul prevăzută la pasul 2, astfel încâtîn acest mod, cei 5 pași expuși se pot reduce la 3 esențiali.

Pe lângă aspectele benefice prin prisma facilitării testării implicate de modificarea supusă discuției, ea are influențe prin prisma costului și performanței. Din acest punct de vedere, evaluând problematica în context de integrare pe scară largă și foarte largă, importanță mai mare decât suplimentul în circuistică, prevăzut de implementarea celor (s+1) copii ale MUX, are degradarea de performantă cauzată de întârzierea pe cele două niveluri logice suplimentare (fig.4.6). Aceasta a condus la ideea reproiectării schemei interne a flip-flop-ului prin incorporarea în accasta a MUX-ului, furnizând un nou tip de flip-flop, așa numit cu multiplexare a datelor (multiplexed data flip-flop), la care, exceptând posibilul efect al încărcării de intrare suplimentar (effect of additional gate fan-in), întârzierea adițională este eliminată. De altfel, firme de prestigiu angrenate în domeniul calculului au acordat o atenție deosebită ideii de facilitare a testării prin tehnica scanării, ele oferind soluții diferite, pe care le vom analiza în cele ce urmează, la problematica expusă vizând, în esență, două aspecte, și anume, implementarea căii de scanare din elementele de memorare și modul în care este interfațată această cale cu circuistica funcțională.[Flin-94,ChAg-90,DaKa-89,MaPa-89,Prad-86].

4.1.2.2.Tehnica de scanare LSSD

Una dintre cele mai cunoscute și răspândite metode pentru sinteza unor scheme secvențiale testabile pe principiul scanării este cea denumită Level-Sensitive-Scan Design (LSSD), aplicată în mod curent în practica de proiectare de către firma IBM [MFGH-94,ChAg-90,GuGB-90,HuSe-89,AlJL-88,Prad-86]. În vederea implementării conceptului de scanare, metoda uzitează în calitate de element de structură intim destinat memorării informației un latch de construcție specială care să faciliteze interconectarea ca registru de deplasare.Pentru a releva modificarea și a elucida aspectul "level sensitive", să considerăm, mai întâi, un latch convențional cu reținere a polarității (polarity-hold) având reprezentarea simbolică și tabelul de excitații ilustrate în fig.4.7.



L \ CD	00	01	10	11
0	0	0	1	0
1	1	1	1	0

Fig.4.7

Acoperirea minimală a unităților binare din tabel conduce la următoarea ecuație booleană:

$$L(t+1)=CD+\overline{C}L(t) \qquad (4.1)$$

În mod normal, semnalul de clock C apare (tranzitează din 0 în 1), doar după ce semnalul de date D a devenit stabil la 0 sau la 1, setând ieșirea latch-ului la această nouă valoare logică. Pentru a proteja funcționarea la hazard constând din apariția la ieșire a unor semnale parazite (glitches, spikes), se impune ca durata clock-ului să rămână la 1 un interval de timp acoperitor propagării semnalului de date prin schema internă a latch-ului (cumulând timpii de întârziere la comutarea circuitelor logice, dar și cei rezultați la transmiterea semnalelor pe cablajele de interconectare) și stabilizării stării schemei. Mai există o pretenție impusă latch-ului și anume aceea de a-l proteja de hazardul care poate fi cauzat pe duratele fronturilor, urcătoare sau coborâtoare, ale semnalului de

clock, denumit de stare staționară (steady-state hazard), și care, prin definiție, afectează schemele caracterizate printr-o astfel de distribuție a întârzierilor de dispersie (stray delays) încât, pentru o anumită tranziție de intrare, ajung într-o stare incorectă. În vederea asigurării unei funcționări lipsită de hazard a latch-ului L, la ecuația (4.1) se impune extinsă funcția SAU prin termenul logic DL(t) ajungându-se la :

$$L(t+1)=CD+(\overline{C}+D)L(t) \qquad (4.2)$$

Schema care realizează funcția booleană (4.2) într-o versiune de implementare utilizând porți logice ȘI-NU este dată în fig.4.8, iar în fig.4.9 se descrie funcționarea ei prin contrast cu cea specifică unui flip-flop (FF) de tip D cu comutare pe front anterior.

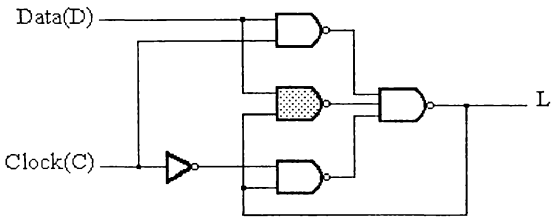


Fig.4.8

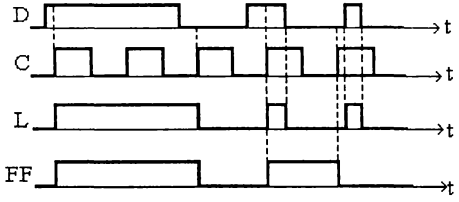


Fig.4.9

Investiția suplimentară în poarta haşurată (fig.4.8) eliberează schema atât de steady-state hazard, cât şi de aşa numitul hazard de ieşire sau tranzitoriu (output or transient hazard), prin care se înțelege apariția la ieşirea schemei, la o anumită modificare de stare, a unor impulsuri parazite având drept cauză aceeaşi dispersie a întârzierilor [Frie-86]. Concluziv, premizele constituite de faptul că semnalul de clock apare după ce datele sunt stabilite şi de faptul că durata semnalului de clock, este acoperitoare pentru propagarea semnalelor, au drept consecință că modificările de stare ale latch-ului din fig.4.8 nu depind de întârzierile componentelor şi a celor cumulate pe cablajele de interconectare, precum nici de duratele fronturilor-urcător şi coborâtor-ale semnalului de clock.

Dar nu latch-ul cu fiabilitate sporită din fig.4.8 este utilizat de strategia LSSD, ci cel reprezentat , în aceiaşi versiune de implementare cu porți ȘI-NU, în fig.4.10, care cuprinde latch-ul din fig.4.8-într-un aranjament puțin modificat, dar realizând aceeaşi funcție (4.2)-în dublă ipostază. Această configurație de latch dublu (double latch) asigură conectarea în calea de scanare tip registru

de deplasare, acoperind a doua parte, cea de Scan Design, din denumirea LSSD a tehnicii de scanare. Primul din cele două latch-uri,cel cu ieşirea marcată cu +L1(fig.4.10) este un aşa numit two-port latch, având două intrări de date separate, precum şi două intrări de clock separate. În mod exclusiv, corespunzător celor două moduri de funcţionare-normal şi de testare-este activată o singură pereche de intrări-de date şi de clock. Astfel,la funcţionarea normală sunt activate intrările de date D şi de clock C a sistemului, intrarea de clock de scanare A fiind menţinută la 0 (inactivă), iar în regimul de testare devine activ canalul cu intrările de scanare I şi de clock de scanare A prin menţinerea la 0 (inactivă) a intrării de clock C a sistemului, datele aplicate la I fiind strobate de trenul de clock aplicat la A. Relevăm în acest context modalitatea de surmontare la soluţia LSSD a penalităţii de performanţă cauzată de schemele de multiplexare din varianta iniţială a metodei Stanford (fig.4.5), canalul de funcţionare normală (perechea de intrări D-C) nefiind precedată de niveluri logice intermediare, de tipul celor din fig.4.6, fiind nemijlocit conectate la reţeaua combinaţională. Aceeaşi observaţie este valabilă şi pentru canalul de scanare (perechea de intrări I-A) prin conectarea intrării de date I în mod direct la ieşirea +L2 a latch-ului anterior. Evident, în cele arătate s-a neglijat încărcarea cu o intrare suplimentară a porţilor interconectate în cruce cu ieşirea +L1 faţă de implementarea convenţională a latch-ului,care majorează întrucâtva timpul de propagare. Pe de altă parte, schema din fig.4.10 mai conţine un al doilea latch, aşa numit single-port, fiind o implementare adaptată a celui din fig.4.8. El înlensneşte interconectarea în calea de scanare, încărcând datele furnizate de latch-ul din primul nivel prin strobare cu un tren de clock aplicat la intrarea de scanare B, care este defazat faţă de trenul la aplicat intrarea C, de clock a sistemului, atunci când funcţionarea normală implică folosirea ambelor trenuri şi este, de asemenea, defazat faţă de trenul de clock aplicat la intrarea de scanare A împreună cu care intervine, în mod obligatoriu, în regim de testare, formând două trenuri fără suprapunere (non overlapping pulse trains). [BrAh-90,GuGB-90,RaFM-90].

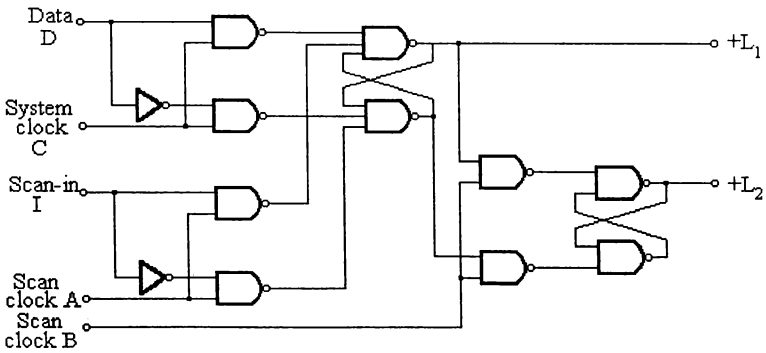


Fig.4.10

Odată precizate particularităţile constructive ale latch-ului folosit în sinteze LSSD, uneori denumit şi shift register latch (SRL) [Lala-85], şi alcătuit prin urmare, din două latch-uri polarity-hold hazard free, prezentăm în fig.4.11 smbolul uzitat pentru acesta. Apelând doar la intrările şi

ieşirea activate în regim de testare, în fig.4.12 este sugerată lănţuirea SRL-urilor în calea de scanare formată la nivelul unei capsule Ci, rezultând cei patru pini suplimentaţi revendicaţi de aplicarea metodei de scanare în versiune LSSD, şi anume unul pentru intrarea de scanare a datelor (scan data-in SDI), un al doilea pentru ieşirea de scanare a datelor (scan data-out SDO) şi două pentru cele două trenuri de clock (A şi B). Extrapolând extensia căii de scanare la niveluri tehnologice superioare, cum ar fi placheta, în fig.4.13 sunt prezentate conexiunile relevante pentru regimul de testare dintre capsule Ci de tipul celei din fig.4.12. Raportată problematica la complexitatea schemelor actuale, trebuie reliefată dimensiunea excesivă la lanţul de scanare, precum şi încărcarea mare la nivelul liniilor de clock. O soluţie sub acest aspect o constituie "spargerea" registrului de deplasare în mai multe cu corespunzătoarea creştere a punctelor de controlabilitate şi observabilitate. De altfel, liniile de clock solicită în realizări LSSD, prin proiectare o precauţie aparte, cu precădere atunci când este necesară condiţionarea unui anumit tren de clock astfel încât să nu fie violat comportamentul level-sensitive al schemei [MoAl-89].Ar mai fi de remarcat faptul că SRL-ul din fig.4.10 poate fi convertit, dacă necesităţile schemei o impun, într-un flip-flop two-port de tip master -slave prin simpla modificarea ilustrată în fig.4.14 prin intermediul simbolului din fig.4.11. La intrările porţii SAU-NU adăugate se prezintă cele două trenuri de clock exclusive-fie cel de sistem (când A=0), fie cel de scanare (când C=0).

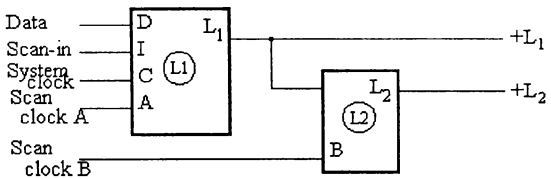


Fig.4.11

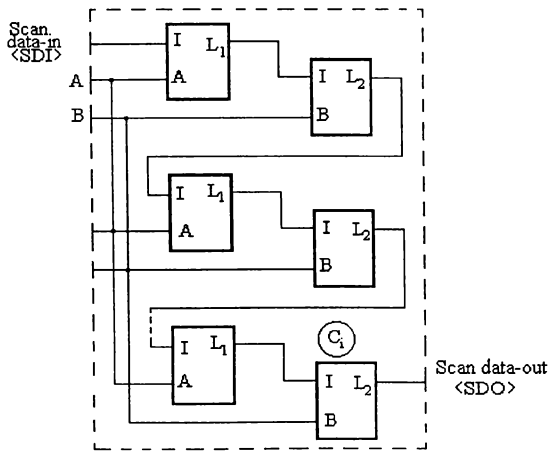


Fig. 4.12

Cu aceste precizări să trecem la substituirea elementelor de memorare din structura generală dată în fig.4.4 cu SRL-uri care uzitează la funcționare normală trenuri de clock decalate aplicate la intrările C şi B, obținând varianta LSSD convențională din figura 4.15 reprezentând corespondenta celei Stanford din fig.4.5. Testarea noii structuri se efectuează prin aplicarea iterativă a unor paşi asemănători cu cei descrişi la metoda Stanford, păstrând caracteristicile comentate privind procesul de verificare , dar adaptați la particularitățile funcționale ale sintezei LSSD.

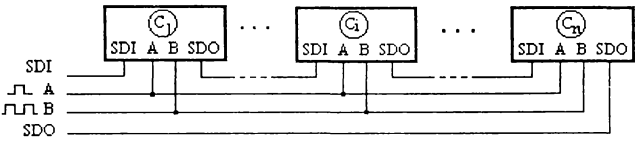


Fig.4.13

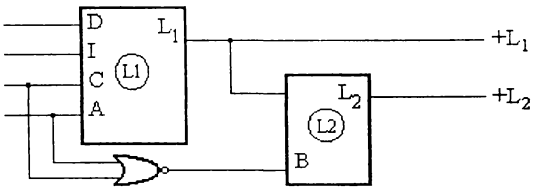


Fig.4.14

1.Prin s impulsuri de clock alternative aplicate la cele două intrări de scanare A şi B, în lanțul de SRL-uri se încarcă subvectorul binar y_1, y_2,...,y_s, interval în care se setează şi subvectorul binar x_1, x_2, ...,x_n, formând, prin concatenare, vectorul de stimulare de lungime (n+s) a cărui efect poate fi evaluat, pentru o parte din logica combinațională, prin compararea-după un interval de timp care să permită propagarea cea mai defavorabilă a semnalelor-vectorului binar răspuns z_1,z_2,...,z_m, disponibil la ieşirile observabile, cu unul memorat, corespunzător funcționării fără defect. Desigur, dacă particularitățile schemei o permit, intrarea SDI se poate suprapune peste una din cele primare x_i, de exemplu x_n.

2. La intrarea de clock C se aplică un impuls pentru a încărca subvectorul răspuns de reacție Y_1,Y_2,...Y_s,- corespunzător părții de logică combinațională neverificată în pasul anterior - în etajele L_1 ale SRL-urilor.

3.Prin aplicarea a s impulsuri de clock alternative la intrările de scanare, de această dată începând la B şi apoi la A, spre deosebire de aplicarea secvențelor de clock din pasul 1, subvectorul stocat în lanțul de SRL-uri, în pasul anterior, este descărcat serial - în vederea comparării cu unul memorat, corespunzător funcționării fără defect - la ieşirea observabilă SDO.

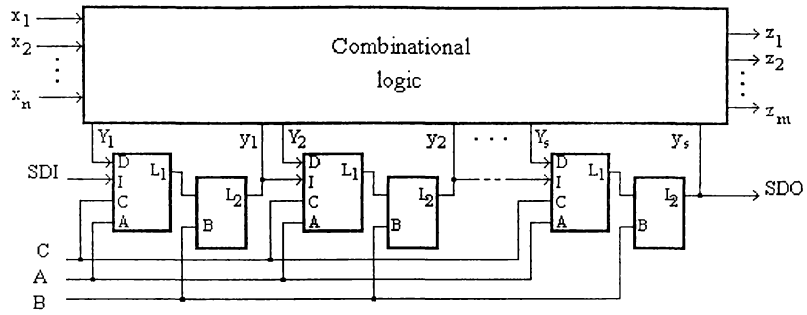


Fig.4.15

Pe lângă acurateţea obţinută în desfăşurarea procesului de testare, trebuie totuşi scoasă în relief investiţia foarte consistentă în circuistică reclamată de structura LSSD din fig.4.15 în raport cu una convenţională. Dacă la aceasta adăugăm pretenţiile legate de acţionarea trenurilor de clock, obţinem contururile motivaţiei diferitelor căutări întreprinse pentru implementarea favorabilă a strategiei LSSD. În acest context ,vom expune în cele ce urmează o variantă care va permite, prin reducerea numărului de circuite, o importantă economie la nivelul ariei de siliciu. În acest sens, pentru început, vom partiţiona logica combinaţională, apelând în caz de necesitate la circuite redundante, în două seturi disjuncte, pe care le denumim Comb1 şi Comb2, astfel încât structura generală din fig.4.4 se prezintă în varianta de implementare cu SRL-uri, mai întâi, ca în fig.4.16.

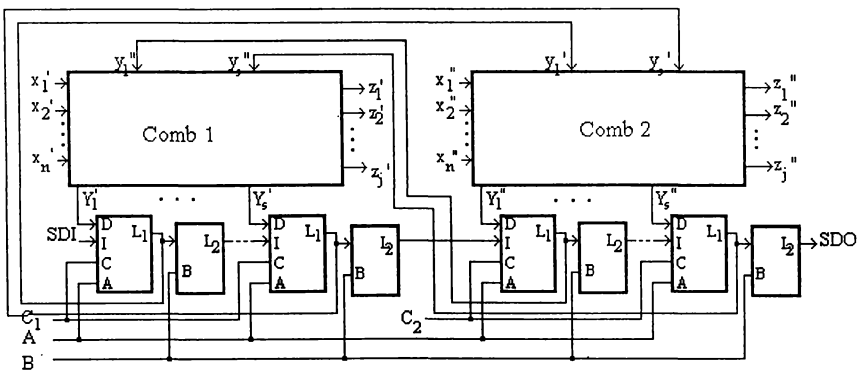


Fig.4.16

Seturile $x'_1,\ldots,x'_i$ şi $x''_1,\ldots,x''_i$ reprezintă intrările primare $x_1,\ldots,x_n$, aşa cum seturile $Y_1',\ldots,Y_k'$ şi $Y_1'',\ldots,Y_k''$ reprezintă intrările de reacţie. În mod similar, seturile $z_1',\ldots,z_j''$ şi $z_1'',\ldots,z_j''$ reprezintă ieşirile observabile şi seturile $y_1',\ldots,y_k'$ reprezintă ieşirile de reacţie. La funcţionare normală, SRL-urile asociate celor două seturi disjuncte Comb1 şi Comb2 sunt comandate prin trenurile de clock fără suprapunere aplicate la intrările C_1, respectiv C_2. În regim de testare, cele

două trenuri de clock sunt comutate pe intrările A, respectiv B, asigurând scanarea pe calea care înlănțuie toate SRL-urile activată pe la intrarea primară SDI şi consultată la ieşirea observabilă SDO. Noua configurație pune mai pregnant în relief încărcarea redundantă a structurii prin etajele L_2 decât configurația din fig.4.15. Dar asupra structurii aşa cum este ilustrată în fig.4.16 se poate interveni printr-o modificare de esență vizând însuşi schema internă a SRL-ului. Astfel, respectând dezideratele anterior enunțate legate de caracteristicile funcționale impuse de strategia LSSD , se ajunge la un nou SRL L_1/L_2 cu reprezentarea simbolică din fig.4.17 şi schema logică, în aceeaşi versiune de implementare cu porți ŞI-NU, dată în fig.4.18.

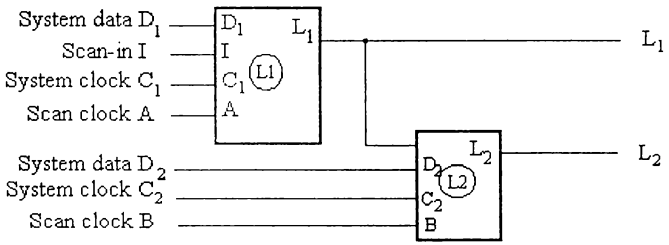


Fig.4.17

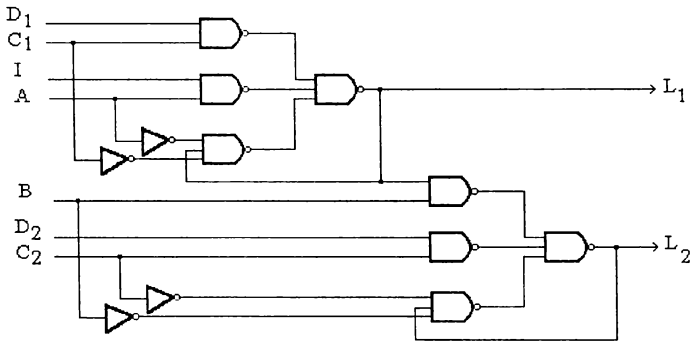


Fig.4.18

Deosebirea fundamentală dintre SRL-ul original şi cel nou este că acesta din urmă are adăugată o cale alternativă pentru datele de sistem, D_2, care sunt strobate printr-un al doilea tren de clock de sistem, aplicat la intrarea C_2 şi fără suprapunere față de trenul original, aplicat la intrarea C_1. Creşterea în complexitate a noului SRL este compensată prin faptul că ambele latch-uri din construcția sa, L_1 şi L_{2*}, pot fi utilizate atât la funcționarea normală, cât şi în regim de testare permițând reducerea semnificativă a numărului de SRL-uri. Astfel, corespondența structurii reprezentate în fig.4.16, într-o implementare cu SRL-uri L_1/L_{2*} este dată în fig.4.19. În final, trebuie să remarcăm că soluția de sinteză LSSD prezentată de altfel foarte atrăgătoare prin prisma economisirii ariei de siliciu, impune o restricție partiționării logicii combinaționale în cele două seturi disjuncte menționate, ceea ce uneori înseamnă eforturi mari de reproiectare.[John-93,SePr-92,HeMi-89].

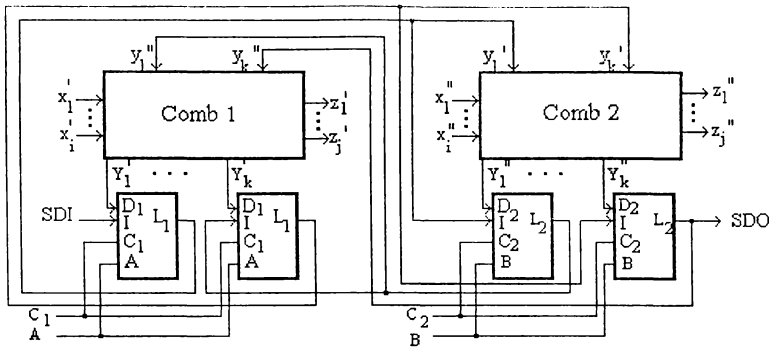


Fig. 4.19

4.1.2.3 Tehnica NEC Scan Path

Dezideratul încărcării elementelor de memorare cu date provenind de la două surse pe care îl implică implementarea căii de scanare îşi găseşte o soluţionare în metoda de proiectare Scan Path practicată la firma NEC (Nippon Electric Company). Aceasta prevede substituirea elementelor de memorare dintr-o structură secvenţială sincronă cu flip-flop-uri de tip two-port, sau aşa cum au fost original denumite "raceless D-type flip-flops with scan path" [LeIV-94,DaWa-90,Prad-86,Lala-85]. Schema internă a unui astfel de flip-flop este prezentată în fig.4.20, ea constând din două latch.uri, L_1, L_2, operate prin două trenuri de clock exclusive aplicate la intrările C_1 şi C_2. Pe durata funcţionării normale, intrarea C_2 este menţinută la 1 logic, strobarea datelor de la intrarea D_1 fiind realizată prin tranziţia din 1 în 0 a impulsului de clock aplicat la intrarea C_1 şi menţinerea acestuia în starea 0 un interval de timp suficient pentru stocarea datei în L_1. La tranziţia din 0 în 1 a impulsului de tact de la intrarea C_1, data memorată în L_1 este preluată de către L_2. În mod asemănător, în regim de testare este menţinută la 1 logic intrarea C_1 şi devine activă perechea de intrări D_2 şi C_2, în sensul că data prezentă,la un moment dat,la intrarea D_2 este stocată în L_1 la tranziţia din 1 în 0 a impulsului de tact aplicat la intrarea C_2, care când revine în starea 1 determină memorarea în L_2 a datei anterior stocate în L_1.

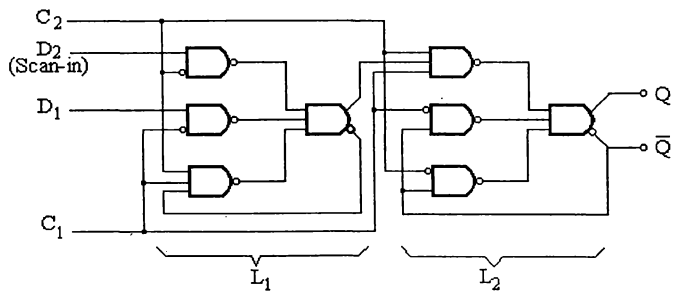


Fig.4.20

Uzitând de flip-flop-ul descris, care trebuie relevat,că spre deosebire de SRL-urile specifice metodei LSSD, implică folosirea unui singur tren de clock care este comutat între intrările C_1 şi C_2 dependent de regimul de funcţionare-normal, respectiv de testare-, prin substituirea elementelor de memorare din fig.4.4, obţinând structura generală din fig.4.21.

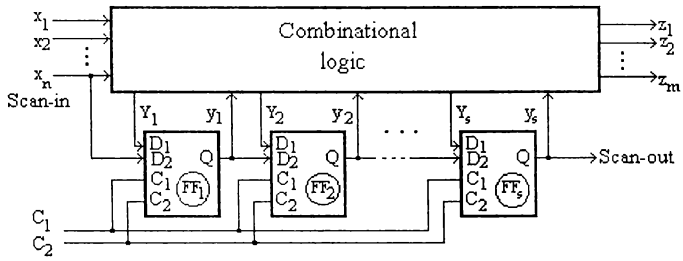


Fig.4.21

Insistând doar asupra segmentului de testare a părţii de logică combinaţională, se impune adaptarea paşilor iterativi relevaţi la anterioarele metode luând în considerare particularităţile funcţionale ale flip-flop-ului folosit la sinteză.

1. Uzitând de s impulsuri de clock aplicate la intrările C_2 şi de una din intrările controlabile, admitem x_n, se încarcă în mod serial subvectorul binar $y_1,y_2,..,y_s$, după care se setează subvectorul binar $x_1,x_2,...,x_n$ din nou, se menţionează că setarea acestui subvector se poate efectua la al s-lea impuls de clock în cazul că x_n coincide logic cu y_1-,formând, prin concatenare un vector de stimulare al părţii de logică combinaţională. După un interval de timp suficient pentru propagarea cea mai defavorabilă a semnalelor se verifică vectorul răspuns observabil z_1, z_2,..,z_m prin comparare cu unul corespunzător funcţionării fără defect.

2. Se aplică un impuls de clock la intrarea C_1 corespunzător funcţionării normale, prin care se încarcă subvectorul răspuns $y_1,y_2,...,y_s$ al părţii de logică combinaţională care generează starea internă rmătoare în setul de flip-flop-uri.

3. Prin alte s impulsuri de clock aplicate, din nou, la intrarea C_2, corespunzătoare regimului de testare, se activează calea de scanare, conţinutul fiecărui flip-flop fiind descărcat serial la ieşirea Scan-out unde poate fi comparat cu unul memorat corespondent funcţionării fără defecte.

Referindu-ne la modalitatea practică de implementare a tehnicii de scanare descrise, aşa cum a fost ea aplicată în sistemele FLT-700 ale firmei NEC [LeKr-91,Prad-86,Lala-85],în fig.4.22 se sugerează soluţia prin care registrul de deplasare de dimensiune foarte mare, formând calea de scanare care înlănţuie toate flip -flop-urile structurii este divizat în mai multe părţi corespunzătoare unor anumite unităţi tehnologice, cum ar fi capsula de circuit integrat sau placheta cu cablaj imprimat. Astfel, reliefând doar conectarea flip-flop-urilor în calea de scanare, se remarcă faptul că la cei trei pini suplimentari de intrare (Scan-in), respectiv ieşire (Scan-out) seriale şi de aplicare a trenului de clock C_2 în regim de testare, se mai adaugă doi, S_x şi S_y, pentru aplicarea unor semnale de selecţie a unităţii tehnologice, să admitem o plachetă, ceea ce permite-prin intermediul porţii

marcate cu *, prevăzută cu colector în gol- conectarea împreună a mai multor plachete la o singură linie de inspecție comună, conferind, la un moment dat, doar unei singure plachete accesul la aceasta linie.

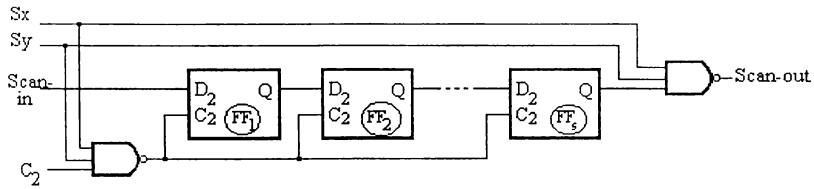


Fig.4.22

4.1.3.Tehnici de scanare cu acces asigurat prin elemente redundante

4.1.3.1.Structuri Scan-Set

Caracteristic celor trei metode expuse a fost intervenția prin modificări asupra structurilor secvențiale sincrone încât latch-urile sau flip-flop-urile funcționale să permită încărcarea pentru stimulare, respectiv descărcarea pentru inspecție, a unor date de testare. Este însă posibil ca la structura funcțională, convențională să fie adăugat un registru de deplasare redundant având menirea de a permite introducerea unor vectori binari de stimulare prin care să fie controlate puncte de acces interne schemei (funcția de set) şi de a permite extragerea, în vederea consultării, a unor vectori binari răspuns prin care să.poată fi observate puncte de acces interne schemei (funcția de scan). Structurile prevăzute cu această facilitate de creştere a testabilității sunt numite scan-set, iar în varianta generală, corespondenta scan-set a celei prezentate în fig.4.4, la care elementele de memorare le admitem de tip latch, este dată în fig.4.23.

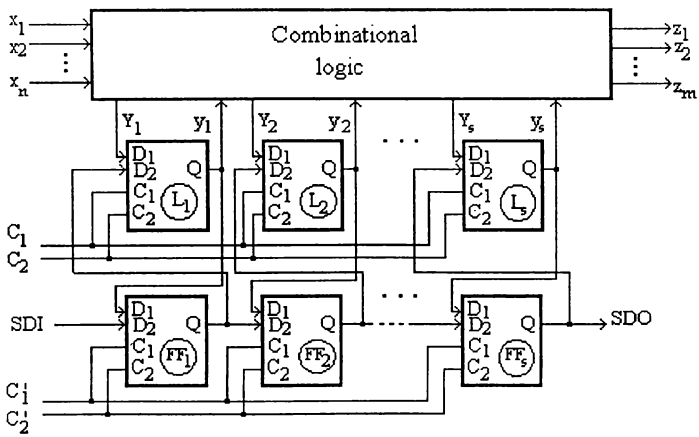


Fig. 4.23

Referindu-ne mai întâi la redundanța noii structuri, se remarcă faptul că latch-urile single-port au fost înlocuite prin unele two -port, implicând adăugarea la perechea de intrări date /clock (D_1/C_1) convențională încă o astfel de pereche (D_2/C_2), care va fi activată în scop de testare. În plus, la fiecare latch L_i i se asociază un flip- flop FF_i, și acesta de tip two-port, cu perechea de intrări D_1/C-destinată încărcării conținutului latch-ului în flip-flop sub controlul trenului de clock aplicat la intrarea primară C_1'- și cu perechea de intrări D_2/C_2- destinată încărcării în flip-flop-ul curent a conținutului celui anterior sub controlul trenului de clock aplicat la intrarea primară C_2'-, asigurând astfel interconectarea flip-flop-urilor într-un registru de deplasare. În vederea stimulării , subvectorii binari corespunzători intrărilor de reacție $y_1,y_2,.....,y_S$ sunt mai întâi, încărcați serial în lanțul de flip-flop-uri de la intrarea primară SDI (scan data-in) utilizând un tren de s impulsuri de clock aplicate la intrarea primară C_2', după care sunt transferați în paralel în latch-urile, corespondente pe la intrările de date D_2 ale acestora utilizând un impuls de clock aplicat la intrarea primară C_2.Pe de altă parte, în vederea evaluării răspunsurilor , subvectorii binari corespunzători ieșirilor de reacție $Y_1,Y_2,...Y_S$ sunt, mai întâi, încărcați în paralel în latch-uri pe la intrările de date D_1 ale acestora utilizând un impuls de clock de sistem aplicat la intrarea primară C_1, după care conținuturile latch-urilor sunt încărcate în paralel în flip-flop-urile corespondente pe la intrările de date D_1 ale acestora utilizând un impuls de clock aplicat la intrarea primară C'_1 și, în final, conținuturile flip-flop-urilor sunt descărcate serial pe la ieșirea observabilă SDO (scan data out) utilizând un tren de s impulsuri de clock aplicat la intrarea primară C_2.[NaBH-94,BaBa-90,LoHu-90,Prad-86].

Comparate cu sinteze LSSD, structurile scan-set revendică mai multe intrări controlabile, un sistem mai complicat de gestionare a trenurilor de clock, și un surplus în circuistică. Legat de acest din urmă aspect, se poate aprecia că investiția în circuite la nivelul fiecărui latch este aproximativ de trei ori mai mare la structurile scan-set în raport cu cele LSSD, în sensul că ambele metode implică substituirea fiecărei latch-admis în mod convențional de tip single-port-printr-unul de tip two-port, iar configurațiile scan-set mai solicită, la fiecare latch, câte un rang al registrului de deplasare constituit de un flip-flop echivalent, în circuite, cu două latch-uri. Dar dezavantajele relevate sunt balansate de importantul avantaj al structurilor scan-set față de metodele prezentate constituit de faptul că registrul de deplasare de testare poate fi încărcat cu conținuturile latch-urilor chiar și pe durata funcționării normale, permițând o verificare dinamică a configurației. De asemenea, structurile scan-set prezintă facilitatea investigării fluxului informațional din noduri ale schemei care nu trebuie să fie în mod obligatoriu ieșiri de latch-uri prin conectarea acestora la flip-flop-uri suplimentare ale registrului de deplasare, permițând astfel extensia observabilității asupra unor noduri non-latch ale configurației.

4.1.3.2.Tehnica de scanare prin multiplexare

Toate tehnicile de scanare prezentate până în prezent au făcut uz de un registru de deplasare pentru a converti date captate serial în unele paralel, respectiv date captate în paralel în unele predate serial. Dar funcția de serializare a datelor obținute în paralel poate fi implementată prin intermediul unei scheme de multiplexare. Admițând că elementele de memorare sunt cele din structura generală

dată în fig.4.4, prezentăm în fig.4.24 varianta cu scanare prin intermediul unuimultiplexor cu s intrări şi k<s ieşiri. Pentru a contura redundanţa pe care o implică facilitarea testării prin această metodă, la circuitele necesare implementării schemei de multiplexare trebuie adăugate cele reclamate de registrul R în care se încarcă adresa SADR (scan address) nodurilor din schema care se doresc inspectate. În acest context se menţionează că prin configurarea registrului R ca numărător este posibilă investigarea în secvenţă, sub controlul unui tren de impulsuri de clock care incrementează starea numărătorului, a tuturor nodurilor din structură conectate la schema de multiplexare. În acest mod se reduce numărul intrărilor primare necesare încărcării adresei de scanare.La aceste puncte controlabile se adaugă cele k ieşiri observabile ale multiplexorului, care se pot reduce la unul, dar cu cât valoarea k este mai mare cu atât se accelerează procesul de testare. O soluţie sub acest aspect o reprezintă conectarea celor k ieşiri de multiplexor la pini de ieşire ai unităţilor tehnologice, astfel încât unii dintre aceştia sunt utilizaţi în mod alternativ, fie ca ieşiri funcţionale ale structurii, fie ca ieşiri de inspectare a datelor de test.

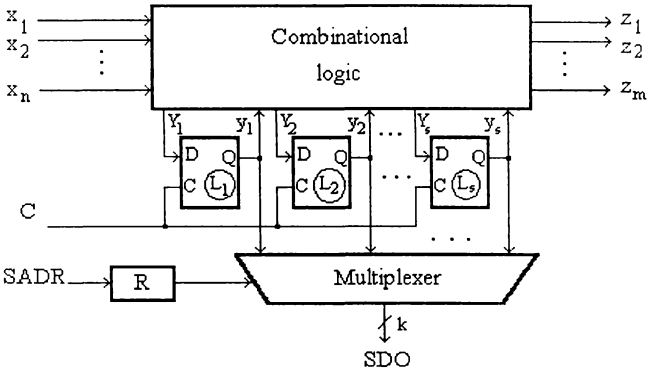


Fig.4.24

Ca şi structurile scan- test, cele cu scanare prin multipliexare pot accesa noduri din schemă altele decât ieşirile latch-urilor. De asemenea, operaţia de scanare poate să se desfăşoare pe durata funcţionării normale, afirmaţie care în contextul anterior menţionatei reduceri a numărului de pini prin utilizarea alternativă a unora dintre cei de ieşire implică investiţii suplimentare în circuistică [Barb-92,Mark-87,Prad-86].

Structura cu scanare prin multiplexare în forma expusă permite îmbunătăţirea observabilităţii structurii neavând nici o contribuţie la creşterea controlabilităţii. Este însă posibilă ameliorarea acesteia apelând în acest scop la o schemă redundantă de demultiplexare care să permită setarea stărilor latch-urilor la valoarea impusă de derularea procesului de testare. Utilizarea unui demultiplexor pentru configurarea vectorilor binari de stimulare a structurii secvenţiale sincrone şi a unui multiplexor pentru evaluarea vectorilor binari răspuns stau la baza tehnicii de scanare cu acces aleator care va fi descrisă în cele ce urmează.

4.1.4. Tehnica de scanare cu acces aleator

Soluția alternativă pentru tehnica de scanare serială de creştere a testabilității- bazată pe înlănțuirea elementelor de memorare în registre de deplasare-pe care o oferă multiplexarea şi demultiplexarea a condus la implementarea unei tehnici de scanare pentru structuri secvențiale sincrone bazate pe latch-uri care a fost denumită, având caracteistici comune cu selecția unei celule din memoriile cu acces aleator, scanare cu acces aleator.Ca şi în cazul memoriilor cu organizare 2D, prin intemediul unei adrese, sunt generate de către decodificatoare corespunzătoare semnale S_x, respectiv S_y, care permit selecția individuală a fiecărui latch în vedereea controlării sau observării stării sale.Pentru a răspunde acestei cerințe se impune intervenția asupra schemei interne, latch-ul devenind aşa numit adresabil (addressable latch), cu structura dată în fig.4.25.[GuGB-90,Prad-86].

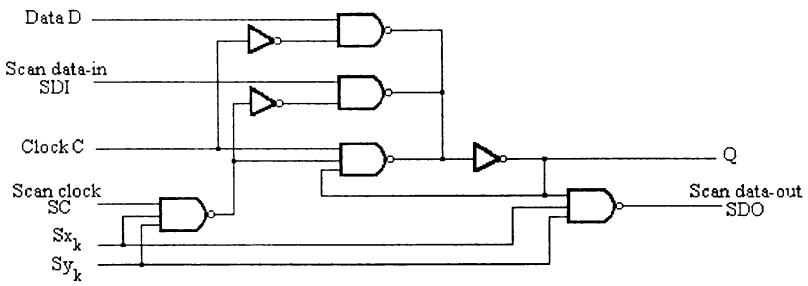


Fig.4.25

La funcționarea normală, datele de la intrarea D sunt transferate la ieşirea Q a latch-ului prin tranziția din 1 în 0 a impulsului de clock C, interval în care intrarea de clock de scanare SC este menținută inactivă la 0 logic. Pe de altă parte, în regim de testare, când intrarea de clock C este menținută inactivă la 1 logic, intervine un registru de adresare a cărui conținut este aplicat la două decodificatoare care setează pe 1, la un moment dat, doar acele ieşiri S_{xk} şi S_{yk} care se aplică la latch-ul k care se doreşte selectat. Uzual, registrul de adresare este prevăzut şi cu funcția de numărare permițând selecția în secvență a tuturor latch-urilor adresabile.În urma selecției, la tranziția din 0 în 1 a impulsului de clock, aplicat la intrarea SC, data de scanare de pe linia SDI, comună tuturor latch-urilor, este stocată în cel decodificat, fiind disponibilă la ieşirea Q pentru a controla un nod intern al schemei. În vederea observării stării latch-ului, în urma selecției acestuia, data stocată va fi disponibilă, în formă complementată, la ieşirea SDO a unei porți ŞI-NU prevăzută cu colectorul în gol, permițând, prin interconectarea ieşirilor SDO a tuturor latch-urilor, inspectarea stărilor acestora prin intermediul unei singure linii. Cu aceste precizări, să trasformăm structura generală din fig.4.4 într-una de scanare cu acces aleator substituind toate elementele de memorare prin latch-uri adresabile. Se ajunge la structura reprezentată în fig.4.26, la care ,în vederea formării subvectorului binar de stimulare aplicat la intrările de reacție $y_1, y_2,...,y_s$ în regim de testare, se încarcă, în condițiile selecției prealabile a fiecărui latch cu registrul de adresare ca numărător, secvența binară disponibilă la intrarea comună SDI în cele s latch-uri utilizând s impulsuri de clock la intrarea SC.

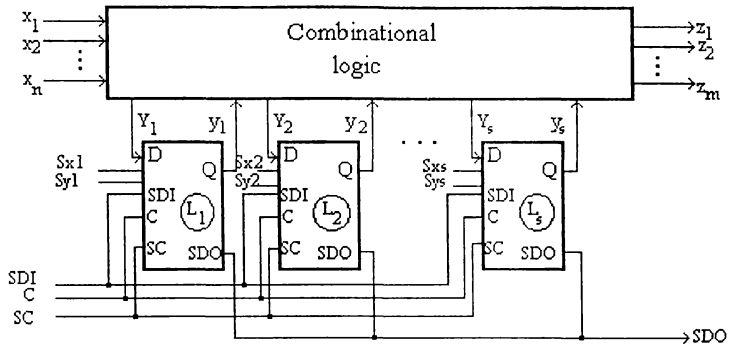


Fig.4.26

În vederea observării subvectorului de la ieşirile de reacţie Y_1, Y_2,...,Y_s, prin aplicarea unui impuls de clock la intrarea C, acesta este încărcat în cele s latch-uri, după care sunt baleiate adresele acestora cu numărătorul de adresare şi prin corespunzătoarea aplicare a s impulsuri de clock la intrarea SC, stările latch-urilor sunt făcute disponibile, în secvenţă, la ieşirea comună SDO. Ar mai fi de menţionat că structura prezentată posedă abilitatea de a permite, prin corespunzătoare gestionare a impulsurilor de clock, consultarea stărilor latch-urilor pe durata funcţionării normale.

În vederea implementării strategiei expuse, pentru a minimiza penalităţile cauzate de adresabilitatea latch-urilor şi pentru a beneficia de avantajele tehnologiei ECL aleasă pentru realizare, firma Fujitsu a supus schema latch-ului adresabil din fig.4.25 unor modificări, transformându-l într-unul de tip set/reset, pe baza căruia a elaborat o metodă sistematică de proiectare orientată înspre creşterea testabilităţii chip-urilor VLSI [Prad-86,Lala-86].Noul latch set /reset adresabil, a cărui schemă internă este prezentată în figura 2.27, posedă o intrare de ştergere (clear), comună tuturor latch-urilor structurii, prin care este posibilă iniţializarea stării acestora la aplicarea frontului căzător al unui semnal de ştergere generală care precede operaţia de setare. În urma adresării unui anumit latch, admitem L_k, acesta este selectat prin starea de 1 logic a semnalelor S_{xk} şi S_{yk}, după care starea sa este poziţionată la valoarea logică provenită pe linia de setare (preset), care este, de asemenea, comună tuturor latch-urilor constituind ieşirea unei porţi ŞI comandată prin intrările primare de scanare date (SDI), respectiv clock (SC). Ca şi anterior, comună tuturor latch-urilor este linia de ieşire SDO. Un model pentru structura generală a unei reţele secvenţiale sincrone, este prezentată în fig.4.28, la care raportăm următorii paşi de descriere a procedurii de testare pentru partea de logică combinaţională.

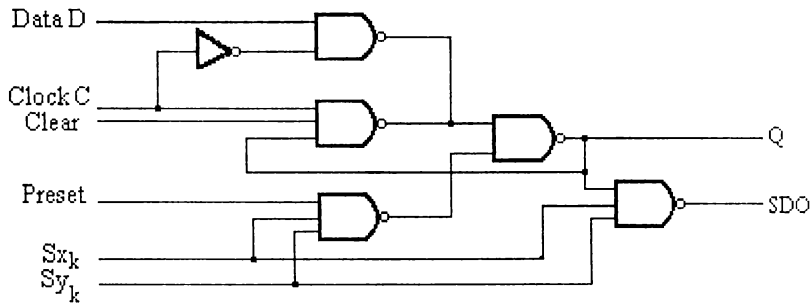


Fig.4.27

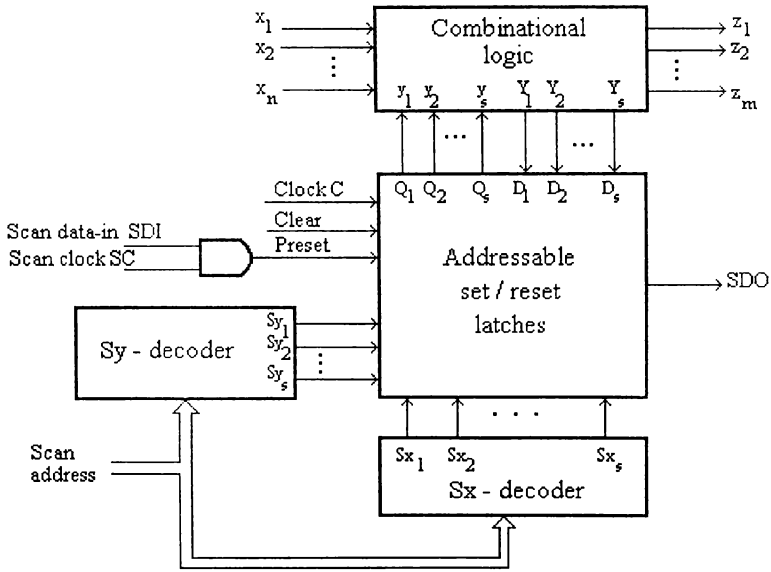


Fig.4.28

1. Se aplică un semnal de ştergere generală (master reset) la intrările clear a tuturor latch-urilor.

2. Se formează şi se decodifică adresa unui latch şi se setează la SDI valoarea binară din vectorul de stimulare care corespunde latch-ului, după care se aplică un impuls de clock la SC.

3. Se repetă pasul 2 până la poziţionarea tuturor celor s latch-uri la valorile binare corespunzătoare intrărilor de reacţie $y_1, y_2,...,y_s$.

4.Se aplică subvectorul binar şi, după un interval de timp suficient propagării semnalelor pe calea cea mai lungă, se compară vectorul binar de la ieşirile observabile z_1, z_2,...,z_m cu unul memorat corespunzător funcţionării corecte.

5. Se aplică un semnal de clock la intrarea C prin care starea internă următoare a schemei , disponibilă la ieşirile de reacţie Y_1, Y_2,..,Y_s, este încărcată în latch-uri.

6. Se formează şi se decodifică adresa unui latch, după care se compară starea acestuia cu una memorată corespunzătoare răspunsului schemei la funcţionare corectă.

7. Se repetă 6 prin baleierea adreselor tuturor latch-urilor.

8. Se repetă paşii 1 la 7 până la epuizarea tuturor vectorilor de stimulare testării părţii de logică combinaţională.

Strategia descrisă prezintă avantajul asigurării controlabilităţii şi observabilităţii pentru toate latch-urile structurii, care este obţinut, luând în considerare cele 2 porţi de adresare din schema internă a fiecărui latch (fig.4.27) şi circuitele din decodificatoarele de adresare, pe seama unei investiţii suplimentare, evaluată la 4 porţi pentru un element de memorare. La acest "sacrificiu" în favoarea testabilităţii trebuie adăugat un număr de 10-20 pini suplimentari pentru adresarea latch-urilor, precum şi pentru datele şi controlul de scanare. Acest număr poate fi redus la 6 prin utilizarea unui numărător de adrese cu capacitate de încărcare serială. La cele menţionate, se impun unele restricţii de proiectare logică, cum ar fi excluderea elementelor de memorare asincrone.

4.1.5.Tehnica de scanare cu cale de intrare-ieşire

Tehnicile de scanare prezentate anterior îmbunătăţesc testabilitatea prin creşterea controlabilităţii şi observabilităţii datorată unui acces extins asupra unor noduri interne ale schemelor, precum şi prin eliminarea necesităţii generării vectorilor de stimulare pentru scheme secvenţiale. În cele ce urmează vom descrie o strategie care urmăreşte ameliorarea testabilităţii prin intervenţii de aşa manieră asupra schemelor încât să rezulte simplificarea funcţiilor care trebuie asigurate de echipamentele de testare fizice, contribuind la reducerea investiţiilor necesare realizării acestora.

Corespondenţa structurii generale din fig.4.4 în varianta cu cale de scanare intrare-ieşire (I/O scan-path structure) este dată în fig.4.29 [LeBu-90,Prad-86].

Ca o primă măsură de facilitare a testării structurii, elementele de memorare de sistem, admise de tip latch, sunt implementate prin sinteze LSSD formând o cale de scanare sau inel intern (internal sean path or ring).Prin controlul asigurat de semnalul M aplicat demultiplexorului DMUX, respectiv semnalul N aplicat multiplexorului MUX, registrul de deplasare este încărcat serial în vederea stimulării de la intrarea primară SDI (scan data in), respectiv descărcat serial în vederea evaluării răspunsurilor, pe la ieşirea SDO (scan data out), în conformitate cu strategia de verificare LSSD descrisă anterior. Pe de altă parte, ca o a doua măsură, la fiecare pad de interconectare (bounding pad) a unităţii tehnologice - pe care o admitem ca fiind constituită de capsula de circuit integrat- cu exteriorul, atât pentru intrări, cât şi pentru ieşiri, se ataşează o pereche de latch-uri (a pair of scan path latches) care să permită înlănţuirea într-un registru de deplasare. Aceste latch-uri de

intrare-ieşire (I/O latches) formează aşa numita cale de scanare sau inel extern (external scan path of ring), care, corespunzător pad-urilor de interconectare intrare, este detaliată parţial în fig.4.30.

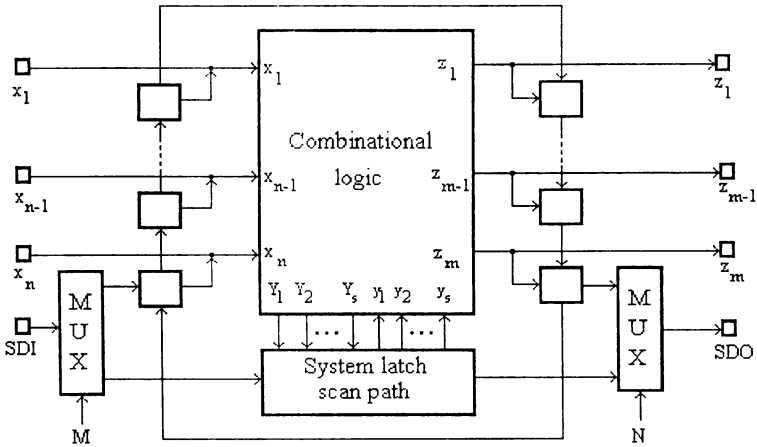


Fig.4.29

Intrările primare $x_1,x_2,...,x_n$ pot fi setate fie în paralel când variabila de control P, aplicată porţilor ŞI-NU cu colector în gol marcate cu *, este menţinută la 0 logic, sau serial prin intermediul lanţului de latch-uri de intrare-ieşire şi apoi strobarea stărilor etajelor L_1 prin trecerea pe 1 a variabilei P. La fiecare intrare corespund câte două latch-uri LSSD de tip single-port, L_1 şi L_2, comandate , în vederea poziţionării seriale, prin intermediul trenurilor de clock fără suprapunere A, respectiv B. Vectorul binar este încărcat în secvenţă de la intrarea primară SDI, fiind direcţionat înspre latch-urile de intrare-ieşire prin DMUX-, sub controlul semnalului M, şi prin logica intercalată între DMUX şi latch-ul L_2,care precede pe cele corespunzătoare intrării x_n, sub controlul variabilei R comandată la 0 logic.

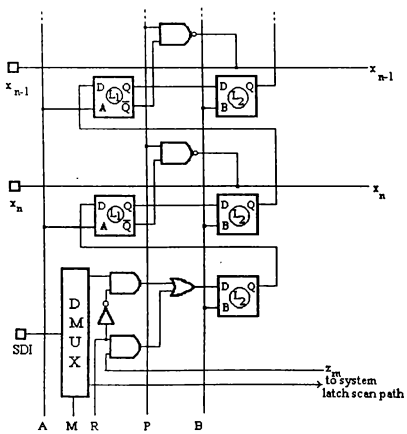


Fig. 4.30

Comutând pe 1 logic variabila de control R, calea de scanare externă poate fi închisă prin conectarea la intrarea D a aceluiaşi latch L_2 a ieşirii latch-ului L_1, , corespunzător ultimei ieşiri observabile z_m, într-un inel prin care să poată fi recirculată o unitate binară. Se obţine în acest mod o reconfigurare a structurii care permite sortarea chipurilor după viteză la testarea efectuată la fabricant înainte de operaţia tehnologică de încapsulare.În mod similar, la fiecare pin ieşire corespunde o pereche de latch-uri, L_1 şi L_2, permiţând ca vectorul binar de la ieşirile z_1,,z_2,.....,z_m, prealabil stocat în calea de scanare externă, să fie descărcat serial în exteriorul chip-ului pe la ieşirea observabilă SDO prin direcţionarea fluxului binar prin MUX sub controlul semnalului N şi corespunzătoare aplicare a trenurilor de clock A şi B (fig.4.29).Poate fi deci remarcat faptul că prezenţa căii de scanare externe permite testarea chip-ului prin intermediul unui număr redus de pini suplimentari. Pe de altă parte, se reduc în mod considerabil numărul punctelor de interferaţă dintre chip-ul şi, prin extensie, unitatea tehnologică de testat şi echipamentul fizic care trebuie să asigure execuţia operaţiilor de verificare, ceea ce conduce la simplificarea constructivă a acestuia, cu influenţe favorabile relative la costul său.

4.1.6. Combinarea tehnicilor de scanare cu noile structuri ASIM

În legătură cu structura prezentată în paragraful anterior relevăm adaptarea ei facilă pentru utilizarea în configuraţii BIST (built-in self-test), asigurând eficientizarea acestora. În calitate de exemplu, să considerăm configuţia generală reprezentată în fig.4.31 [Prad-86] care urmăreşte punerea în relief a elementelor implicate în regimul de testare.Astfel, ca o primă remarcă, se consideră că schema secvenţială sincronă a fost supusă reconfigurării LSSD-implicând modificările impuse de această strategie, inclusiv generarea trenurilor de clock în vederea controlului-, la care procesul de testare se desfăşoară prin secvenţarea nu a unor vectori binari de stimulare determinişti, ci a unora aleatori-mai precis pseudoaleatori [Vasi-94a]-furnizaţi de către un generator de modele aleatoare (random pattern generator) incorporat în capsulă (on-chip). Acesta din urmă,uzual constituit dintr-o schemă secvenţială liniară sintetizată pe baza unui polinom generator primitiv , este capabilă să producă un lanţ binar de lungime(2^l - 1)-unde l reprezintă gradul polinomului generator, respectiv numărul de ranguri ale registrului de deplasare din schema secvenţială liniară.[Vasi-94a]-,care este încărcat, prin punctul de interfaţare SDI (scan data in) în şirul de elemente de memorare LSSD, admise apriori ca făcând parte din reţeaua LSSD testabilă prin modele aleatoare (random -pattern-testable LSSD network). Calea de scanare internă formată din lanţul elementelor de memorare LSSD conduce nu la o ieşire primară SDO (sean data out), observabilă,ca în schemele LSSD convenţionale sau ca în cea din fig.4.29, ci SDO reprezintă punctul de conexiune dintre calea de scanare internă şi cea externă, care, spre deosebire de cea prezentată în fig.4.30, este alcătuită dintr-un lanţ de elemente de memorare LSSD, având câte un latch L_1 single-port ataşat fiecărui pin intrare (PI) şi câte un latch L_2 double-port ataşat fiecărui pin ieşire (PO). Calea de scanare externă în noua configuraţie, fiind formată din latch-uri care corespund pinilor de intrare şi ieşire, este denumită cale de scanare de "graniţă" (boundary scan path). Un anumit vector de stimulare va fi aplicat prin intermediul unor trenuri, A în conjuncţie cu B, de (n+s) impulsuri de clock, unde n reprezintă numărul intrărilor primare şi s a celor de reacţie.

Considerând, pentru simplitate, că numărul m al ieşirilor observabile este egal cu n (fig,4.31), în urma aplicării unui impuls de clock de sistem prin care sunt încărcate în latch - uri valorile binare corespunzătoare unui răspuns acestea sunt deplasate,prin (n+s) perechi de impulsuri de clock A şi B, înspre pinul de ieşire PO* şi , de asemenea, sunt încărcate într-un analizor de semnături serial on-chip. În cazul că semnătura generată de acesta nu coincide cu una precalculată şi memorată, corespunzătoare funcţionării corecte, fluxul binar captat prin intermediul pinului PO* este examinat în exteriorul chip-ului în vederea diagnosticării erorii [ScSh-94,UpRa-94,PrGH-90].

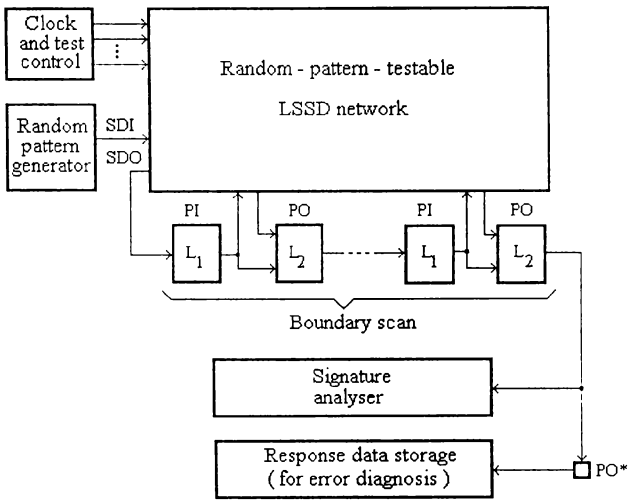


Fig. 4.31

O adaptare a căii de scanare combinate, internă extinsă cu boundary scan, la structuri multichip prin utilizarea în scop de testare a unui chip dedicat este prezentată, la nivel de principiu, în fig.4.32 [Prad-86].

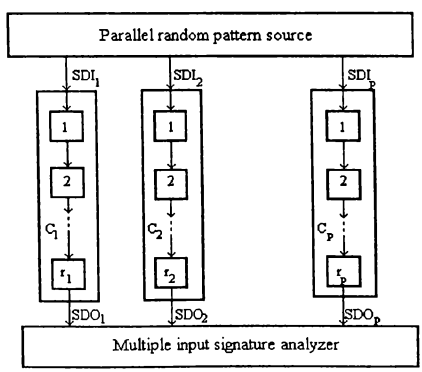


Fig. 4.32

În fiecare dintre cele p chip-uri, în regim de testare, elementele de memorare-funcționale și unele atașate pinilor de intrare și ieșire, implementate în manieră LSSD-sunt interconectate în registre de deplasare de lungimi care pot fi variabile $\gamma_1,\gamma_2,....,\gamma_n$. Un chip auxiliar, dedicat scopului de testare, conține două elemente de structură destincte, anume un generator de secvențe pseudoaleatoare (parallel random pattern source) și un analizor de semnături cu intrări multiple (multiple input signature analyzer). Primul, cu caracteristici constructive anterior relevate [Vasi-94a], asigură încărcarea, în paralel, în cele p chip-uri, pe la intrările SDI, a unor vectori de stimulare de aceeași lungime dată de valoarea maximă dintre $\gamma_1,\gamma_2,...,\gamma_p$. Pe de altă parte, analizorul de semnături asigură captarea, în paralel, a fluxurilor binare răspuns, disponibile la ieșirile SDO și comprimarea acestora în vederea comparării cu informații etalon, anterior evaluate pentru funcționarea corectă. În acest context,relevăm posibilitatea de fructificare a mecanismelor de comprimare propuse în referatul [Vasi-94a,b], diferite de cele date în [Prad-86] și superioare prin prisma capacității de detecție a potențialelor erori. Legat de structura prezentată, ar mai fi de menționat capacitatea de încărcare simultană cu captarea paralelă a vectorilor răspuns în analizorul de semnături a fluxurilor binare de stimulare din generatorul de secvențe pseudoaleatoare.

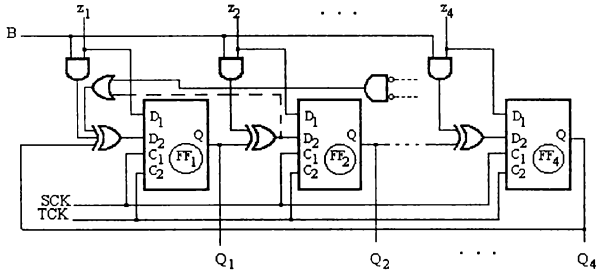


Fig.4.33

Pentru a contura posibilitățile de facilitare a testării deschise de către tehnicile de scanare, amintim, de asemenea, proiectarea practicată de Storage Technology Corporation (STC)[Prad-86], conform căreia structurile integrate includ o cale de scanare LSSD internă și o cale de scanare externă asociată pinilor I/O. Elementul aparte specific chip-urilor STC este faptul că latch-urile din aceiași calea de scanare externă care sunt asociate pinilor de intrare, pot fi configurate într-un generator de secvențe pseudoaleatoare, iar latch-urile din aceiași cale de scanare externă, care sunt asociate pinilor de ieșire, pot fi configurate într-un analizor de semnături cu captare paralelă. Prin urmare, o structură STC are capacitate de funcționare de tipul celei descrise principial în contextul schemei reprezentate în fig.4.29, când testarea este asigurată prin vectori de stimulare determiniști furnizați din exteriorul chip-ului, dar are și capacitatea , prin condiționare BIST corespunzătoare, de reconfigurare a căii de scanare externă în cele două elemente de structură menționate, care asigură testarea cu stimulare prin vectori binari pseudoaleatori și evaluarea răspunsurilor prin comprimare paralelă în semnături. Fără a insista asupra strategiei de testare STC, care poate fi imaginată relativ ușor, preluăm ideea de registre reconfigurabile din flip-flop-uri multiport și analizăm schema propusă

în [Prad-86, pag.151], pe care o reproducem, adaptând semantica utilizată pentru circuite, în fig.4.33.

La funcționare normală, schema funcționează ca registru cu încărcare paralelă a datelor z_i disponibile la intrările D_1 ale flip-flop-urilor FFi sub controlul impulsului de clock de sistem SCK aplicat la intrările C_1 ale FFi. În regim de testare, în maniera cunoscută de la structurile ASIM [Vasi-94a,b], sub controlul realizat prin variabila aplicată la intrarea B sunt asigurate două configurații ale schemei, în ambele fiind aplicate impulsuri de clock de testare TCK la intrările C_2 ale FFi.. Astfel, când B=0, schema funcționează ca numărător furnizând vectorii binari de stimulare pentru testare exhaustivă a schemei comandate prin ieşirile Q_i ale FFi, metodă, evident, aplicabilă în situația când numărul intrărilor schemei testate este suficient de redus pentru a face economică această strategie de stimulare. Pe de altă parte, când B=1, schema funcționează ca analizor de semnături paralel realizând procesul de comprimare al fluxurilor binare de date z_i pe baza expresiei polinomului generator G(x) dat de relația:

$$G(x) = x^4 + x^3 + 1 \qquad (4.3)$$

O primă remarcă legat de schema prezentată este cel de substituție a legăturii de reacție de la ieşirea Q_1 la poarta SAU marcată prin xxx cu cea punctată dusă de la ieşirea porții SAU EXCLUSIV dintre FF1 şi FF2 la poarta SAU prin care se câştigă în capacitate de detecție, aspect amplu în capitolul anterior. În al doilea rând, ideea de reconfigurabilitate la nivelul căilor de scanare ne conduce la implementarea unor structuri ASIM prin elemente de memorare LSSD, dar nu de tipul celor prezentate, ci prin unele-modificate, în mod original, la nivelul schemei interne-care să permită reducerea penalităților de performanță specifice unor astfel de structuri [Vasi-94b]. Astfel, în continuare vom prezenta latch-ul dublu multiport propus pentru o structură ASIM având circuitele SAU EXCLUSIV, corespondente termenilor polinomului generator G(x) primitiv utilizat la sinteză, incluse între rangurile registrului de deplasare. După cum este cunoscut [Vasi-94a,b], structura ASIM menționată anterior rezultă ca optimă sub aspectul triplului impact performanță/cost/capacitate de detecție, singurul ei dezavantaj constituindu-l nivelul SAU EXCLUSIV suplimentar introdus între acele ranguri ale registrului de deplasare care corespund termenilor polinomului generator G(x), nivel care grevează asupra frecvenței impulsurilor de clock utilizate în procesul decomprimare. Plecând de la ideea aglomerării în schema internă a logicii de intermediere dintre ranguri şi uzitând de condiționarea impulsurilor de clock conformă cu cerințele de scanare LSSD, în fig.4.34 se prezintă, în versiune de implementare cu porți ŞI-NU, schema internă corespunzătoare latch-ului complex al unui rang ASIM prevăzut cu funcțiile de încărcare/descărcare paralelă, de registru de deplasare, de generator de secvențe pseudoaleatoare şi de analizor de semnături paralel.

Ca şi la latch-ul LSSD convențional de tip dublu two-port, avem activate, la funcționare normală, perechea de intrări D(date) şi C(system clock), iar, în regim de testare, perechea de intrări I (scan data in) şi, de această dată, A_1 (Scan clock A_1), care împreună cu activarea intrării B (scan clock B) asigură înlănțuirea latch-urilor în registrul de deplasare, în clasică manieră LSSD anterior descrisă. La aceste intrări se mai adaugă R pentru date, constituind ieşirea de reacție a ultimului rang

ASIM, precum şi două trenuri de clock de scanare, A_2 pentru configurarea ASIM generator de secvenţe pseudoaleatoare şi A_3 pentru configurarea ASIM ca analizor de semnăturiparalel. Evident, trenurile C, A_1, A_2 şi A_3 sunt exclusive şi toate sunt fără suprapunere în raport cu trenul B.

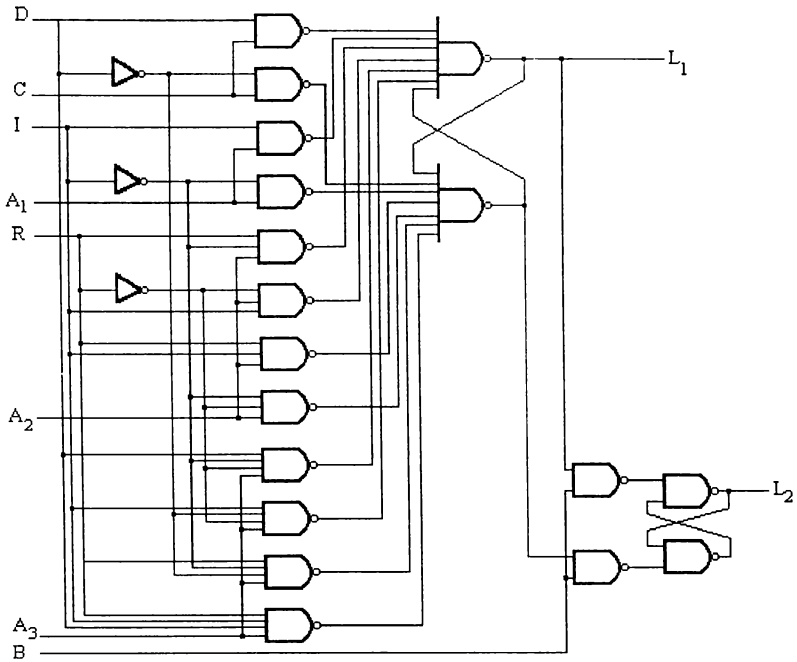


Fig.4.34

Ca o calitate a schemei prin prisma performanţei se subliniază echilibrarea întârzierilor pe toate canalele de date şi surmontarea în varianta de sinteză propusă, a scăderilor relevate la noua schemă de comprimare paralelă expusă în [Vasi.-94a]. Acest câştig de performanţă a fost obţinut pe seama creşterii încărcărilor de intrare (fan-in) corespunzătoare porţilor, penalitate care trebuie evaluată în contextul capacităţii de integrare specifică fabricantului de circuite.

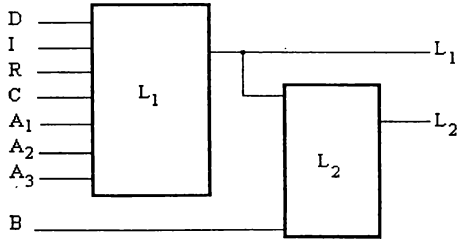


Fig. 4.35

Uzitând de simbolul de reprezentare din fig.4.35 pentru latch-ul din fig.4.34, o structură ASIM, ca element de sine stătător, care realizează procesul de comprimare paralelă pe seama aceleiaşi expresii (4.3) a polinomului generator G(x), ca şi schema din fig.4.33, este dată în fig.4.36. Pe lângă configuraţia regulată pe care o prezintă, favorabilă integrării pe scară largă şi foarte largă, se remarcă faptul că acele ranguri ASIM la care nu corespund termeni în expresia lui G(x) prezintă intrarea R a etajelor L_1 inactivă (R=0),ceea ce permitea schemelor respectivelor latch-uri să poată fi simplificate prin omiterea completă a conexiunilor R(cele 4 porţi ŞI-NU de intrare din porţiunea de jos a fig.4.34 având deci câte o intrare mai puţin). Pe de altă parte, dacă structura din fig.4.36 este utilizată ca facilitate BIST în contextul reconfigurării boundary scan ca generator de secvenţe pseudoaleatoare pentru pini de intrare, respectiv ca analizor de semnături paralel pentru pini de ieşire, schema din fig.4.34 poate fi supusă unor noi simplificări, determinate de faptul că cele două configuraţii sunt exclusive, ceea ce conduce la economisirea a 4 porţi din etajele L_1.

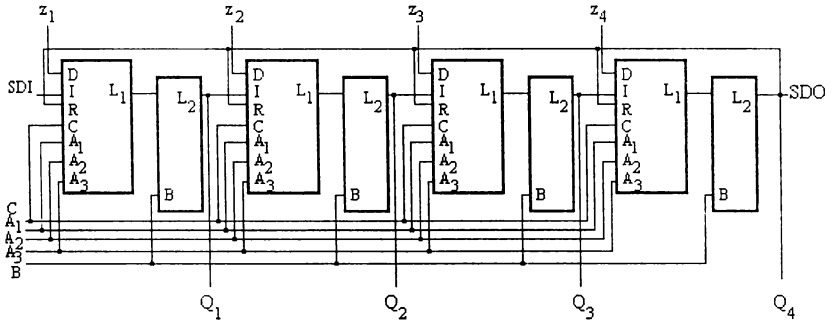


Fig. 4.36

În final, se impune reliefat faptul că tehnicile de facilitare a testării expuse trebuie adaptate la stadiul actual de dezvoltare al circuisticii integrate, caracterizat, în linii mari, prin (1) niveluri din ce în ce mai ridicate de integrare, (2) apariţia de noi tehnici de încapsulare asociate tehnologiei de montare la suprafaţă (surface -mounting technology), (3) spaţii tot mai reduse între pini, (4) reducerea drastică a punctelor de acces pentru interfaţarea sistemelor de testare,şi (5) costul foarte ridicat al echipamentelor de testare automată. În condiţiile acestui status-quo a devenit imperios necesară standardizarea unor arhitecturi şi proceduri de testare care să fie acceptate de comunitatea producătorilor de chip-uri şi de sisteme de testare automată. În acest sens, fabricanţii din Europa au format aşa numitul Joint Test Action Group (JTAG), care a propus o magistrală de testare pe 4 fire (four-wire test bus) conectată la un port de acces pentru testare prin intermediul căruia se asigură scanarea la nivelul pinilor I/O ai chip-urilor integrate. Propunerea JTAG a fost acceptată de Test Standards Committee şi a fost aprobat la ,15 februarie 1990, standardul IEEE 1149.1 Boundary Scan, la care a aderat majoritatea producătorilor importanţi, printre care şi firma Texas Instruments [xx TI-92, TI-93]. Considerăm de utilitate intercalarea, în proximul paragraf, a exemplului de aplicare al standardului de către această firmă de prestigiu întrucât IEEE 1149.1 (şi dezvoltările sale

[Bell-94,Flin-94]) va constitui probabil corsetul de proiectare al viitoarelor circuite integrate pe scară foarte largă.În acest context, încercăm propunerea de includere a noii structuri ASIM propuse la standard, propunere care, evident trebuie filtrată de specialiştii din domeniul fabricării de circuite integrate.

4.1.7. Aplicarea standardului IEEE 1149.1 la circuite integrate Texas Instruments

Din debut trebuie menționat că firma Texas Instruments a acordat, cu precădere, odată cu trecerea pragului de integrare pe scară largă, o atenție deosebită facilităților de testare a chip-urilor, sens în care amintim, în calitate de exemplu, aplicarea metodei scan-and-set la realizarea structurii secvențiatorului de microprograme SN 74AS8835 [Haye - 88]. Circuitul, introdus în 1986, are principalele registre duplicate prin aşa numite registre umbră (shadow registere), acestea din urmă - incluzând microprograme counter, trap register, interrupt register şi stack shadow, toate pe 16 biți - cu capacitate de înlănțuire într-un registru de deplasare cu posibilitate de accesare prin intermediul pinului de testare Scan In şi cu posibilitate de consultare prin intermediul unui alt pin de testare, Scan Out [Haye - 88] . Ulterior, în formă evoluată, a fost creat un cadru organizat de cercetare - proiectare DFT (design for testability), aşa numitul System Controllability/Observability Partitioning Environment - SCOPE™, permițând realizarea de dispozitive care să beneficieze de capacități built - in test în toate etapele ciclului de fabricație al produsului. Fabricate în procesele tehnologice BiCMOS (bipolar CMOS), respectic Advanced BiCMOS, dispozitivele SCOPE ™ oferă o viteză ridicată, consum redus de putere şi o înaltă capacitate de comandă, la care se adaugă posibilitatea de conectare la o magistrală de testare serială, bazată pe scanare, pe 4 fire, care se supune standardului 1149.1. Atunci când sunt utilizate în locul dispozitivelor standard de interfațare la magistrală, ele permit ca anumite module hardware - cum ar fi chip-urile, plachete sau sisteme - să fie izolate de celelate în scopul detecției şi diagnozei defectelor fără a necesita intervenții prin sonde externe.

Pe de o parte, familia de dispozitive SCOPE ™ include chip-uri dedicate ca funcții de interfațare, grefate pe o configurație tipică de microsistem implementată pe o plachetă cu cablaj imprimat, sunt reprezentate în fig. 4.37.

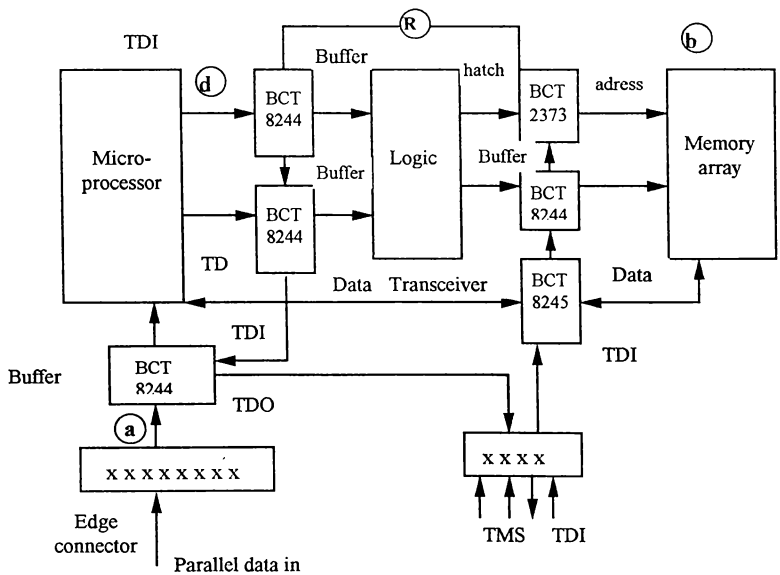


Fig. 4.37

Izolarea în scop de testare se poate efectua la nivelul (a) conectorului de fund de sertar (edge connector), (b) memoriilor, (c) logicii de interconectare/adaptare (glue logic) sau (d) microprocesorului. În acest sens sunt utilizate dispozitive SCOPE dedicate constând din (1) SN 54/74 BCT 8240 şi BCT 8244 - octal buffer/driver,)2) SN 54/74 BCT 8245 - octal bidirectional transceiver, (3) SN 54/74 BCT 8373 - octal transparent - D-Type latch, şi (4) SN 54/74 BCT 8374 - octal D-type flip-flop (neutilizat în fig 4.37), disponibile în capsula de plastic sau ceramice cu 24 pini, pentru tehnologie convenţională sau montare pe suprafaţă, având asignarea pinilor şi funcţiilor acestora diferite de cele corespunzătoare dispozitivelor standard. Cele 4 fire conforme standardului IEEE 1149.1, impuse de implementarea căii de scanare care traversează chip-urile SCOPE ™ sunt (1) TDI - test data in, (2) TDO - test data out, (3) TMS - test mode select, şi (4) TCK - test clock. Dispozitivele SCOPE ™ acceptă protocolul definit de standardul IEEE 1149.1 pentru aplicarea şi captarea datelor de test, iar logica lor de control recunoaşte două moduri esenţiale, anume modul normal şi modul testare. În modul normal, datele specifice sistemului sunt propagate prin aceste dispozitive ca şi prin cele standard. În modul de testare, calea datelor de sistem este inhibată şi pinii I/O ai dispozitivelor SCOPE ™ sunt conectaţi la inelul boundary scan (BSC), aşa cum este sugerat în fig. 4.38. Astfel, se dă posibilitatea să fie aplicaţi vectori binari de stimulare de la ieşirile unor ship-uri SCOPE ™ la legături şi porţiuni din logica funcţională, prin care să se propage, iar răspunsurile să fie captate la intrările altor schip-uri SCOPE ™ în vederea evaluării.

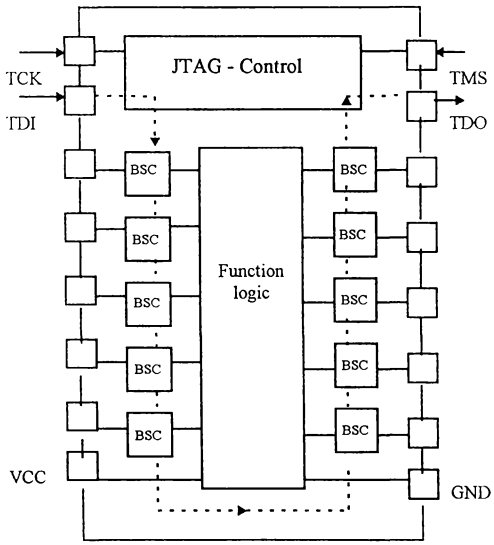


Fig. 4.38

Prin selectarea registrului de deplasare corespunzător unui anumit dispozitiv se asigură calea de scanare de un singur bit prin acesta. Dar pe lângă metodica de testare bazată pe lanţuri binare deterministe încărcate în calea de scanare boundary şi evoluarea secvenţelor binare răspuns prin comparare cu unele corespunzătoare funcţionării corecte, dispozitivele SCOPE ™ de interfeţare prezentate mai permit configuraţii de testare adiţionale, între care amintim (1) controlul căii de scanare boundary prin plasarea ieşirilor anumitor dispozitive în stare de impedanţă înaltă obţinându-se izolarea electrică a unei aşa numite partiţii din logică ce urmează a fi supusă, unei testări locale, (2) controlul căii de scanare boundary astfel încât să poată fi testată logica funcţionării integrată în însuşi dispozitivul SCOPE ™ prin aplicarea de configuraţii binare de stimulare în segmentul de cale

signature analyzer, PSA), asigurând prin aceastea stimularea şi evaluarea răspunsurilor anumitor elemente de structură, cum ar fi, spre exemplu, memoriile, (4) controlul citirii anumitor segmente ale căii de scanare boundary, mod în care se permite inspectarea datelor în urma unei operaţii PSA, şi (5) autotestarea (self test) a căii de scanare boundary prin încărcarea în lanţul de celule boundary a configuraţiei binare complementare celei încărcate iniţial.

Pe lângă dispozitivele SCOPE [TM] de interfaţare, familia de componente SCOPE [TM] realizate de firma Texas Instruments mai cuprinde, în scopul configurării unor sisteme cu capacităţi BIST, controller-e dedicate, de sine stătătoare, care asigură implementarea metodologiei de testare intermediată de o magistrală cu scanare serială. Proiectanţii de configuraţii pot utiliza de aceste controller-e SCOPE în conjuncţie cu orice fel de dispozitive compatibile cu standardul IEEE 1149.1 pentru a obţine sisteme mai flexibile şi cu o mai mare capacitate de trecere relativ la sarcinile de testare, cu posibilitate de a fi de aşa manieră partiţionate încât, în mod virtual, să fie asigurată imunitatea la corupţie pe căile de date. Aşa cum rezultă şi din configuraţia dispusă, din punct de vedere tehnologic, pe mai multe plachete ca în fig. 4.39, apare ca esenţială componenta 'ACT 8990 reprezentând controller-ul de megistrală de testare IEEE 1149.1 (IEEE 1149.1 test bus controller), care asigură interfaţa dintre microprocesoarele populare şi una sau două căi de scanare serială IEEE 1149.1.

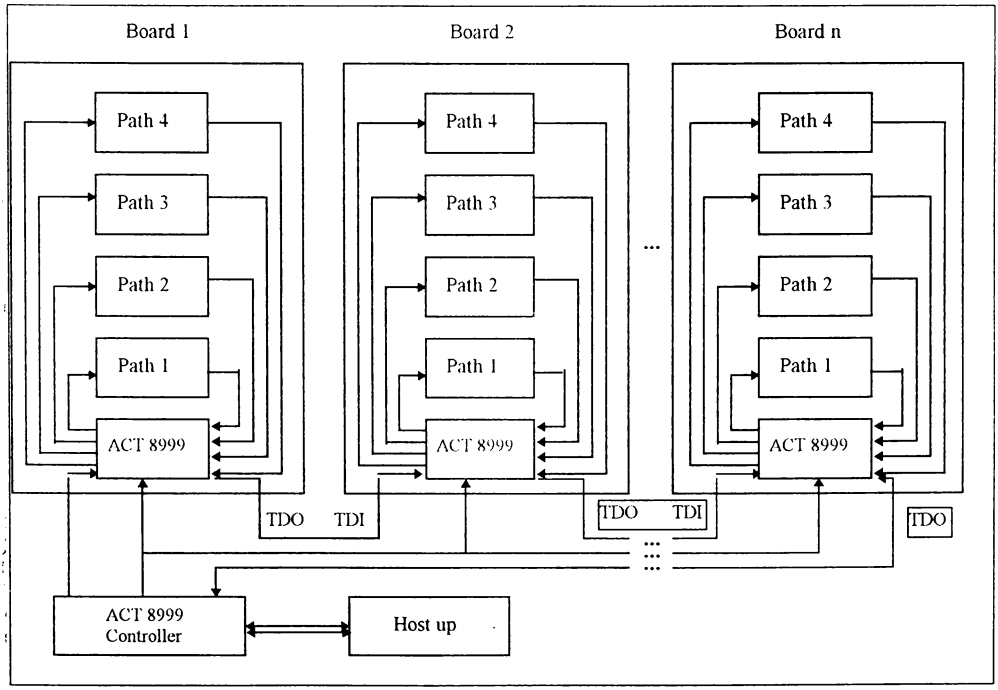


Fig. 4.39

Microprocesorul gazdă (host up) furnizează, în paralel pe 16 biţi, instrucţii şi date la controller-ul de magistrală de testare ' ACT 8990, care converteşte informaţia în formatul serial IEEE 1149.1 şi comandă calea de scanare primară. Aceasta poate fi partiţionată, printr-una din componentele specializate selectoare pe cale de scanare (scan path selectors) 'ACT 8999 sau 'ACT 8997 -câte una pe fiecare plachetă-, în alte 4 căi de scanare fiecare,cu capacitate de identificare a acestora. Spre deosebire de dispozitivul 'ACT 8997 care realizează funcţia de selecţie a căii de scanare fără facilitatea de identificare a plachetei, circuitul 'ACT 8999 (fig. 4.39) posedă şi această din urmă

comandă calea de scanare primară. Aceasta poate fi partiționată, printr-una din componentele specializate selectoare pe cale de scanare (scan path selectors) 'ACT 8999 sau 'ACT 8997 -câte una pe fiecare plachetă-, în alte 4 căi de scanare fiecare,cu capacitate de identificare a acestora. Spre deosebire de dispozitivul 'ACT 8997 care realizează funcția de selecție a căii de scanare fără facilitatea de identificare a plachetei, circuitul 'ACT 8999 (fig. 4.39) posedă și această din urmă capacitate, permițând proiectantului să selecteze, pentru fiecare plachetă, programul de testare adecvat, efectuând operațiile de verificare dorite pentru fiecare plachetă și, în cadrul acesteia, pe câte 4 inele de scanare secundare diferite, ceea ce îi oferă o mai mare capacitate de localizare a defectului potențial. Pe lângă circuitele menționate, familia de controller-e SCOPE ™ mai cuprinde și un monitor de magistrală 'ACT 8994 (digital bus monitor). Dispozitivele 'ACT 8994, 'ACT 8997 și 'ACT 8999 sunt disponibile în capsule cu 28 de pini, iar chip-ul 'ACT 8990 are 44 de pini, toate fiind realizate în Advanced CMOS Technology (ACT).În ceea ce privește elementele de structură ale principalelor dispozitive din familia de controlle-e SCOPE ™ , în fig. 4.40 și fig. 4.41 se prezintă, la nivel bloc, configurațiile 'ACT 8990, respectiv 'ACT 8999.

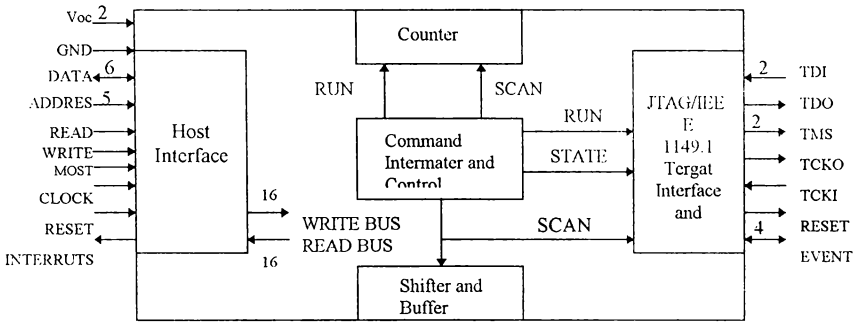


Fig. 4.40

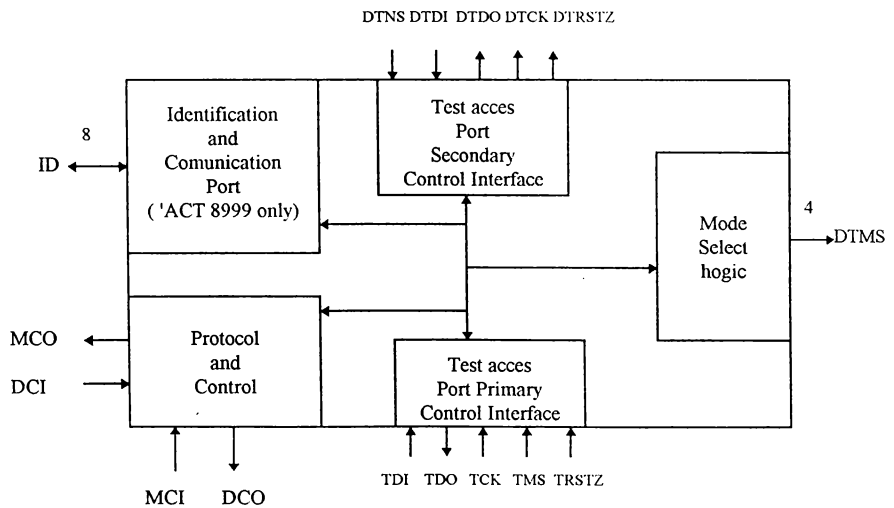


Fig. 4.41

Relativ la controller-ul 'ACT 8990, acesta implementează hardware operațiile de deplasare și de manipulare la nivel de bit permițând controlul a două căi de scanare țintă (target) separate. Acest circuit permite interfațarea eficientă la magistrala de testare IEEE 1149.1, permițând reducerea semnificativă a numărului de instrucție ale procesorului gazdă legate de operațiile de testare. El acceptă două tipuri de comenzi, unele vizând doar magistrala IEEE 1149.1 externă și altele vizând starea internă a controller-ului. Pe de altă parte, 'ACT 8999 posedă o magistrală de 2 biți, care poate fi utilizată pentru a poseda controlul unui controller de magistrală de testare situat la distanță sau pentru a asigura un cod de identificare a până la 255 de subsisteme.

Concluziv, standardul IEEE 1149.1 asigură o metodologie de testare generală bazată pe scanare boundary dând proiectantului posibilitatea să configureze inele de scanare utilizând dispozitive provenite de la diversi producători. Interfața cu magistrala IEEE 1149.1 necesită un mediu de control care să aplice comenzile unui protocol specific, precum și datele de test și care să urmărească rezultatele procesului de verificare. Acest mediu de control poate fi incorporat în sistem, fie extern acestuia. Sistemele cu control incorporat acoperă de la soluții cu implementare firmware care comandă în mod direct inelele de scanare la soluții bazate pe controller-e dedicate, de tipul mediului de control SCOPE ™. Sistemele externe acoperă soluții de la control software prin intermediul unor procesoare aparținând unor platforme de mentenanță la combinații de procesoare și hardware dedicat controlului scanării. În contextul acestor multiple căi de soluționare mediul SCOPE™, oferă o rezolvare favorabilă prin prisma impactului performanță / cost / testabilitate, asigurând cicluri mai scurte de proiectare, costuri de producție și reparație mai reduse, o introducere mai rapidă în fabricație și o întreținere îmbunătățită la utilizator.

4.2. Utilizarea structurilor ASIM modificate în scopul facilitării testării schemelor PLA

Pentru a stabili, cu claritate, conturul preocupărilor din prezentul paragraf în care problematica facilitării testării este abordată în mod diferit față de cele deja expuse, încercăm, pentru început, unele delimitări terminologice. Chiar dacă prezintă unele caracteristici suprapuse, testarea calculatoarelor poate fi divizată în [GuRR-94,PaHe-94,SeMu-87,Prad-86]:

a)Testarea implicită sau concurentă (Implicit or concurrent testing), uneori denumită și verificare (checking), care se referă la testarea on-line menită a detecta erori care apar pe durata operării normale a sistemului. Strategiile practice specifice acestei modalități de testare se bazează pe tehnici de implementare a codurilor de paritate, precum și pe tehnici de duplicare [BuMi-94,FeRK-94,LeKr-91,Gros-89,SeSh-87]. Strategii teoretice relevante se bazează pe apelarea la scheme cu autoverificare (self-checking circuits) [LaHA-94, Cici-92, Barb-92, HoLC-90, LeSa-90, MaPa-89, MaMc-88, McSa-88, WiSh-88].

b)Testarea explicită (Explicit testing), care se referă la testarea off-line efectuată asupra unor părți ale calculatorului sau asupra întregului sistem atunci când acesta nu are funcționare normală curentă. Această modalitate de testare acoperă testarea circuitelor integrate înainte de încapsulare, testarea de producție aplicată capsulelor integrate și plachetelor, precum și testele de acceptanță

(acceptance tests), testele de mentenanță (maintenance tests) și testele de reparație (repair tests) [CBGb-94,VaCh-92,StLe-92,DoAr-90,KaNa-90,LeBu-90,StLe-90,KwBr-87,NiCa-87].

Precizăm că în cele ce urmează problematica dezvoltată este focalizată asupra testării explicite, termenii de testare și test fiind restricționați la acestă modalitate de verificare.Prin sintagma " proiectare pentru testabilitate" (design for testability, DFT) majoritar se înțeleg acele tehnici de proiectare menite a face testarea produselor rezultate economică [Wei-94,Mazu-93,PrGK-90,Yarm-90,DaFC-89,Lili-89,RaSK-88].

Întrucât factorii care contribuie la costul de testare și cel de întreținere sunt diferiți, termenul de testabilitate tinde să fie utilizat în mod imprecis.[Sesn-94,ClDG-92,SuYo-92]. Componentele de cost ale testării sunt determinate, în esență, de costul generării modelelor de testare (cost of test pattern generation) și de costul de aplicare al testelor (cost of test application). La rândul ei, prima dintre componentele de cost menționate, depinde de timpul de calculator revendicat de rularea programelor de generare a testelor, la care se adaugă investiția de capital pentru elaborarea programului sau, altfel expus, depinde de numărul de om-ore necesar elaborării modelelor de testare la care se adaugă timpul necesar sistemului de dezvoltare care să producă testele. Pe de altă parte, costul de aplicare al testelor este determinat de costul echipamentului de testare la care se adaugă costul cauzat de timpul de testare necesar aplicării efective a testelor, așa numitul timp de soclu (socket time) [Prad-86]. Există un compromis care se impune balansat între costul de testare și cel de reparație. Costul testării poate fi redus prin utilizarea unor teste care nu permite detecția tuturor defectelor sau nu permit localizarea tuturor defectelor detectate. Această reducere a costului de testare poate determina o creștere substanțială a costurilor de producție sau a celor de întreținere. Astfel, în general, se dovedește a fi mult mai costisitor a repara o plachetă cu cablaj imprimat defectă decât a elimina un chip defect înainte de montajul acestuia pe plachetă și," urcând " o treaptă în ierarhia constructivă a unui calculator, se dovedește mult mai costisitoare repararea unui sistem defect în raport cu cea a unei plachete defecte.[MiNi-92,VaCh-92].

Nu se va încerca conturarea unei definiții precise pentru testabilitate, dar se admite că testabilitatea crește cu reducerea costurilor pentru generarea testelor, sau de aplicare a acestora și cu maximizarea acoperirii defectelor (fault coverage) sau a capacității de diagnosticare a acestora.(fault diagnosability)[RaFu-91,DaFu-90]. În acest context, se impune subliniat că majoritatea tehnicilor DFT urmăresc creșterea testabilității atât prin reducerea costurilor de testare, cât și prin maximizarea acoperirii și capacității de diagnosticare a defectelor.[PaHK-94,HoLC-90,LiSh-90].

Încercările de reliefare a atributelor schemelor care influențează testabilitatea au condus la conceptele de controlabilitate (controlability) și observabilitate (observability). Controlabilitatea se referă la capacitatea de provocare a unei valori specificate pentru un semnal intern al schemei prin aplicarea de semnale la intrările primare ale schemei. Observabilitatea se referă la capacitatea cu care pot fi determinate la ieșirile primare ale schemei stările semnalelor interne din schemă. Se menționează că multe dintre tehnicile DFT constituie încercări de creștere ale controlabilității și ale observabilității schemelor.

Cea mai directă cale pentru atingerea acestor deziderate constă în introducerea de puncte de test (test points), reprezentând intrări şi ieşiri adiţionale ale schemei utilizate pe durata testării.Există întotdeauna un cost asociat cu adăugarea de pini de test care depinde de treapta tehnologică a ierarhiei de împachetare a calculatorului, fiind, în general, prohibitiv la nivelul capsulelor de circuite integrate datorită cunoscutei limitări a numărului de pini, dar care se justifică, în general, la nivelul plachetelor cu cablaj imprimat.

Cu privire la tehnicile DFT, în literatură se distinge între clasa celor denumite tehnici ad-hoc şi aceea, mai elaborată, cuprinzând metodele de proiectare structurată prezentate în paragraful 4.1. Există şi tendinţă de creştere a testabilităţii, şi acesteia i se subordonează structurile PLA, care urmăreşte, prin intervenţii timpurii asupra proiectelor, fără a afecta dezideratele de performanţă ale schemelor şi cu o investiţie minimă de cost,sinteza de aşa manieră a schemelor încât acestea să rezulte testabile, îndeplinind următoarele caracteristici generale [Prad-86]: (1) Set de stimulare redus , (2)Eliminarea redundanţei funcţionale, (3)Setul de stimulare să poată fi generat cu efort minim, (4)Răspunsurile să poată fi interpretate în mod facil,(5)Defectele să poată fi localizate la nivelul dorit. Unele dintre tehnici permit trecerea unor seturi de vectori stimuli test care să nu depindă de funcţiile furnizate de o anumită structură, oferind posibilitatea unei aşa numite testări independente de funcţie (function independent testing), acesta fiind cazul structurilor PLA[SaBe-90,SaMi-87,TrAF-87].

4.2.1.Problematica testării schemelor PLA

Structurile PLA au devenit blocuri primitive foarte răspândite în proiectarea sistemelor complexe, integrate pe scară largă şi foarte largăm, datorită structurii lor regulate şi a simplităţii sintezei prin posibilitatea "programării" unor funcţii variate de logică aleatoare. Prin adăugarea la acestea, a unor registre şi a unor reacţii ele pot fi transformate în maşini secvenţiale astfel încât la multe dintre microprocesoarele complexe actuale partea de logică de control este implementată prin structuri PLA. Conceptual, o asemenea structură poate fi modelată printr-o reţea logică organizată pe două niveluri ŞI-SAU implementată în forma NU-SAU- NU-SAU, NU-ŞI - NU-ŞI sau ŞI- SAU. În fig.4.42 se prezintă o schemă tipică de implementare în tehnologie n-MOS a unei structuri PLA prin două niveluri NU-SAU - NU-SAU, realizând următoarele funcţii logice:

$$f1 = x1 + \overline{x_1}x2x3 + \overline{x_2}\,\overline{x3} + x4 \qquad (4.4)$$

$$f2 = x1\overline{x_3} + \overline{x_1}x2x3 + \overline{x_2} + x4 \qquad (4.5)$$

După cum rezultă şi din fig.4.42, structura PLA constă din patru blocuri : (1) decodificatorul intrărilor (input decoders), în mod uzual, constând dintr-o colecţie de circuite decodificatoare cu una sau două intrări: (2) matricea ŞI (AND array), care formează termenii produs, ale cărei linii orizontale mai sunt denumite de bit (bit lines) şi ale cărei linii verticale contributoare la formarea termenilor produs mai poartă denumirea de linii de cuvânt (word lines); (3) matricea SAU (OR array), care implementează funcţia logică SAU între termenii produs furnizând pe liniile de ieşire (output lines) forme canonice disjunctive (disjunctive canonical forms); (4) amplificatoare de ieşire (output buffers). Utilizatorul poate conecta sau deconecta tranzistoare n-MOS în punctele de

(output buffers). Utilizatorul poate conecta sau deconecta tranzistoare n-MOS în punctele de intersecție ale liniilor matricilor,programând astfel funcțiile logice. Menționăm că, schematic structurile PLA nu se reprezintă prin tranzistoare în mod explicit, acestea fiind substituite prin puncte la intersecțiile liniilor unde sunt conectate.

Există trei tipuri de defecte mai frecvent întâlnite în structurile PLA: (1) defecte de blocare (stuck-at faults) cauzate de întreruperi de linii; (2) defecte de scurtcircuit (short faults) între două linii adiacente; (3) defecte de contact sau puncte de intersecție (contact faults or crosspoint faults) cauzate de lipsa de tranzistoare indispensabile funcțiilor dorite sau de tranzistoare în exces față de cele necesare realizării funcțiilor dorite.

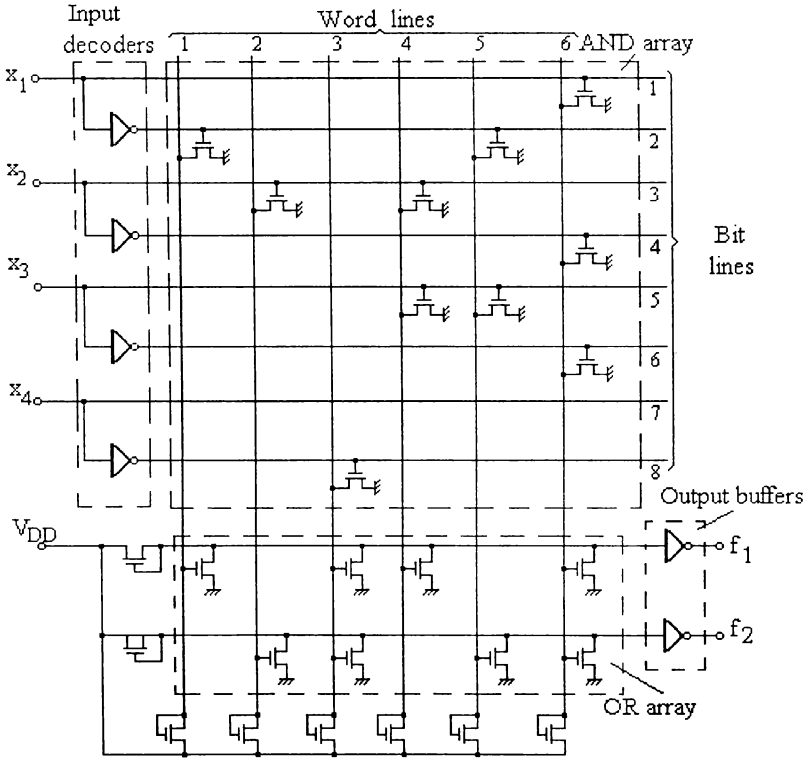


Fig.4.42

Admițând ipoteza, de altfel mai probabilă, că defectul este singular, efectele tipurilor de defecte menționate pot fi sintetizate astfel: (1) defectele din decodificatorul intrărilor şi din matricea ŞI pot cauza fie activarea unor terméni produs neadresați, fie neactivarea unor termeni produs adresați, fie ambele, fie că pot să nu aibă nici un efect; (2) defectele din matricea SAU şi buffer-ele de ieşire pot fi fie incrementa numărul de 1-uri corespunzător vectorilor binari de ieşire, fie decrementa acest

număr, fie pentru anumite funcţii pot incrementa numărul de 1-uri, iar, pentru alte, pot decrementa acest număr, fie că pot să nu aibă nici un efect.

Descrierea modului complex în care se poate defecta o structură PLA sugestionează asupra unor dificultăţi care se întâmpină la generarea vectorilor stimuli şi, corelat, la evaluarea răspunsurilor unor asemenea scheme. Dacă e să ne referim la categoria defectelor de contact, mai delicată pentru că ea implică defecte a căror detecţie nu poate fi întreprinsă prin strategiile bine stăpânite corespunzătoare modulului de defectare de blocare, să considerăm în schema din fig.4.42 că, datorită unei scăderi a procesului de programare, tranzistorul de la intersecţia liniei de bit 3 cu cea de cuvânt 4 a fost omis. Aceasta conduce la suplimentarea termenilor produs din funcţia logică f_1 dată de (4.4). În vederea testării, trebuie căutaţi toţi termenii produs în exces generaţi de către defect, dar care nu coincid cu nici unul din termenii produs acoperiţi de funcţia logică, şi aceştia , se impun baleiaţi în calitate de vectori binari de stimulare. Atât procesul de căutare, cât şi cel de interpretare a răspunsurilor devine dificil, fapt care a condus la ideea ca structura PLA să fie modificată, printr-o investiţie cât mai mică în circuite suplimentare, astfel încât aceasta să poată fi testată printr-un set de vectori stimuli determinat care să depindă doar de dimensiunea structurii, dar să fie independentă de funcţiile realizate de aceasta. În plus, se urmăreşte ca, prin modificare, vectorii răspuns de ieşire să poată fi verificaţi de către structura însăşi, în aşa fel încât să nu mai fie necesară memorarea vectorilor răspuns. Structura PLA modificată asigură o testare aşa numit independentă de funcţie, permiţând creşterea capacităţii de trecere a seturilor de testare în raport cu structura PLA convenţională.

4.2.2.Configurarea redundantă a schemelor PLA în vederea asigurării verificării prin seturi de teste independente de funcţii

Strategia utilizată la elaborarea structurii PLA testabilă independent de funcţie (function independent testable programamable logic array, FITPLA) constă în asigurarea unui mecanism prin care să poată fi forţată selecţia oricărei linii de bit şi oricărei linii de cuvânt. În vederea selectării, la un moment dat a unei singure linii de bit, la structura convenţională din fig.4.42 se mai adaugă două linii de control y_1 şi y_2, cu y_1 comandând în poartă câte un tranzistor conectat la fiecare linie de bit pe care este aplicată variabila de intrare directă şi cu y_2 comandând în poartă câte un tranzistor conectat la fiecare linie de bit pe care este aplicată variabila de intrare complementată, aşa cum rezultă din fig.4.43.

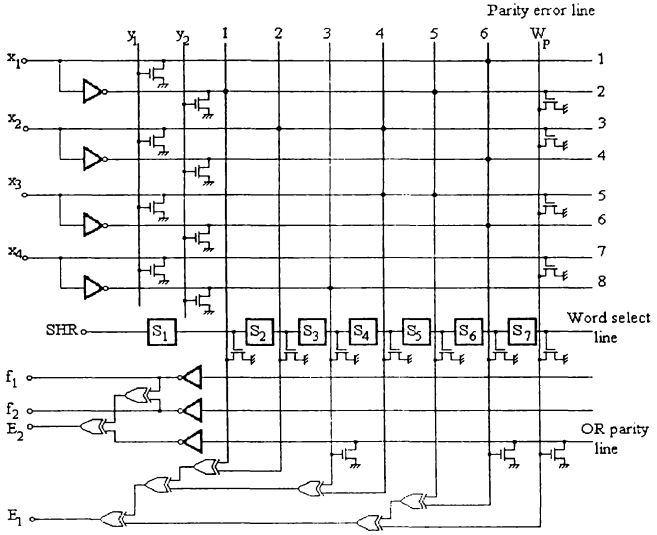


Fig.4.43

Pe durata operării normale a structurii PLA, y_1 şi y_2 sunt comandate prin vectorul binar 00, neafectând funcționarea. Pe de altă parte, în regim de testare, y_1 şi y_2 vor fi comandate exclusiv în sensul că atunci când uneia dintre intrările de control i se aplică 1, celeilalte i se aplică 0 şi viceversa. Efectul acestor comenzi poate fi urmărit în tabelul din fig.4.44.

Input				Control lines		Bit lines							
x_1	x_2	x_3	x_4	y_1	y_2	1	2	3	4	5	6	7	8
x_1	x_2	x_3	x_4	0	0	x_1	x_1	x_2	x_2	x_3	x_3	x_4	x_4
x_1	x_2	x_3	x_4	1	0	0	x_1	0	x_2	0	x_3	0	x_4
x_1	x_2	x_3	x_4	0	1	x_1	0	x_2	0	x_3	0	x_4	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	0	0	0	0	0	0

Fig.4.44

Este astfel posibil ca prin vectorul binar 01 aplicat la y_1 şi y_2 să fie selectată doar o singură linie comandată de o variabilă directă, când aceasta este comandată prin 1 şi toate celelalte sunt comandate prin 0. În mod similar, aplicând vectorul binar 10 la y_1 şi y_2, este posibil a fi selectată doar o singură linie comandată de complementul unei variabile de intrare, când acesta este comandată prin 0 şi toate celelalte variabile de intrare sunt comandate prin 1. Se impune o mențiune legată de acele structuri PLA la care se efectuează decodificarea concomitentă a două variabile de intrare [SaBe-90,Prad-86], cărora trebuie să li se asigure un control suplimentar prin patru linii y în vederea asigurării selecției, la un moment dat, a unei singure linii de bit.

Conex, cu modificarea prezentată, în vederea economisirii de spații de memorare pentru vectorii răspuns şi a evitării comparațiilor prin care se îmbunătățesc capacitatea de trecere a setului de stimulare, structura PLA convențională este supusă unei modificări suplimentare prin adăugarea la matricea ŞI a unei linii de cuvânt de paritate (parity word line). La aceasta, aşa cum rezultă din fig.4.43 este prezentată conectarea tranzistoarelor de aşa manieră încât pentru fiecare linie de bit să rezulte un număr impar de puncte de intersecție. La modificarea menționată se mai adaugă arborele de circuite SAU EXCLUSIV cu câte o intrare conectată la fiecare linie de cuvânt şi a cărei ieşire E_1 revendică un pin suplimentar destinat testării.

Prin selectarea în mod individual a fiecărei linii de bit şi observarea răspunsurilor la E_1, este posibil să se detecteze orice defect singular de blocare la 0 de pe linia de bit selectată şi orice defect singular de tip punct de intersecție. (lipsă sau exces de tranzistor) din matricea ŞI. Pentru o structură PLA cu n variabile de intrări, partea de verificare descrisă necesită 2n vectori de stimulare întrucât avem 2n linii de bit. Pentru cazul particular considerat, acest subset de stimulare numit set de testare (A test set) cuprinde 8 vectori şi este prezentat în tabelul din fig.4.45 împreună cu valoarea corespunzătoare la E_1 în cazul când structura nu este afectată de defectele menționate.

Test set A						Bit lines								E_1
x_1	x_2	x_3	x_4	y_1	y_2	1	2	3	4	5	6	7	8	
1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	1	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	1	0

Fig. 4.45

Admițând, spre exemplu că linia de bit 1 este blocată la 0,aceasta va determina ca toate cele şapte linii de cuvânt, inclusiv linia de cuvânt 6, să fie la 1 atunci când se aplică primul vector de stimulare din setul A, ceea ce va determina ca E_1 să rezulte 1 şi să permită detecția defectului. De asemenea, dacă prin programare a fost omisă conectarea tranzistorului la intersecția liniei de bit 5 cu linia de cuvânt 5 (fig.4.43), acest defect tip punct de intersecție va fi relevat prin al 6-lea vector de stimulare din fig,4.45, care va determina să apară 0 doar pe linia de cuvânt 4 şi cea de paritate, nu şi pe linia de cuvânt 5, determinând ca numărul liniilor de cuvânt pe 1 să fie impar, deci E_1 va fi pe 1 permițând detecția defectului.

Pe de altă parte, testarea matricii SAU şi a amplificatoarelor de ieşire se realizează prin selecție individuală a fiecărei linii de cuvânt. În acest sens, o nouă modificare este întreprinsă în structura PLA convențională prin adăugarea la matricea ŞI a unei linii bit suplimentare, aşa numita linie de selecție a cuvântului (word select line). Aceasta prezintă câte un punct de intersecție pentru

fiecare linie de cuvânt (fig.4.43), în plus având intercalate rangurile S_1 la S_7 ale unui registru de deplasare (shift register,SHR) şi acesta suplimentat la structura convenţională. Prin intrarea suplimentară SHR, conţinutul registrului de deplasare poate fi încărcat cu o configuraţie binară dorită care să permită ca doar una dintre liniile de cuvânt să fie selectată la un moment dat. Pe durata operării normale în fiecare rang al registrului de deplasare trebuie să fie 0 astfel încât să nu fie afectate funcţiile logice ale structurii.

Conex cu această modificare şi urmărind, în esenţă, aceleaşi deziderate ca şi prin controlul de paritate cu ieşirea E_1, detecţia defectelor din matricea SAU se bazează pe adăugarea la aceasta a unei linii de ieşire, aşa numita linie de paritate SAU (OR parity line), care are prevăzute tranzistoare conectate în maniera reprezentată în fig.4.43 în aşa fel încât numărul punctelor de intersecţie ţinând cont de cele funcţionale, pe fiecare linie de cuvânt din matricea SAU să rezulte impar.În vederea asigurării acestei parităţi impare este prevăzut arborele de circuite SAU EXCLUSIV care are câte o intrare conectată la fiecare linie de ieşire şi are ieşirea E_2 revendicând un nou pin destinat operaţiunilor de testare.

Test set B													Bit lines								Word lines							Output lines			Check	
$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$	y_1	y_2	S_1	S_2	S_3	S_4	S_5	S_6	S_7	1	2	3	4	5	6	7	8	1	2	3	4	5	6	W_p	f_1	f_2	OR_p	E_1	E_2
0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1
1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1
1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1
1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1
0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
0	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1

Fig.4.46

Selecţia unei singure linii de cuvânt este realizată prin încărcarea în registrul de deplasare a unei configuraţii prevăzând câte o unitate binară în fiecare rang cu excepţia aceluia care comandă tranzistorul conectat la acea linia de cuvânt dorită selectată şi prin setarea la 0 a tuturor liniilor de bit aplicând la intrările primare x_i şi la cele de control y_i a unor vectori binari corespunzători. Acest set de stimulare, denumit set de teste B (B test set), pentru structura PLA din fig.4.43, este dat în tabelul 4.46 împreună cu configuraţiile binare corespunzătoare stării funcţionale fără defecte pentru liniile de bit, de cuvânt şi de ieşire, precum şi pentru cele două ieşiri de control la paritate impară , E_1 şi E_2.

Se poate remarca faptul că orice defect tip punct de intersecţie singular în matricea SAU sau orice defect singular de blocare a uneia dintre liniile de ieşire din matricea SAU va provoca o modificare a valorii parităţii calculate la ieşirea E_2. În calitate de exemplu, să considerăm că prin

programare a fost omis tranzistorul de la intersecția liniei de cuvânt 3 cu linia de ieşire f_2. Acest defect de punct de intersecție singular va determina ca atunci când se aplică al 5-lea şi al 6-lea vector de stimulare din setul de teste B (fig.4.46), linia f_2 să devină 0, astfel încât E_2 rezultă 1 şi permite detecția defectului. De asemenea, sunt detectate defectele singulare de blocare a liniilor de cuvânt prin modificarea valorii la ieşirea de verificare E_1. Astfel, blocarea la 0 a oricărei linii de cuvânt va fi pusă în evidență prin valoarea 0 la E_1 provocată de oricare dintre perechile de vectori de stimulare care selectează linia de cuvânt afectată de defect.Pe de altă parte, blocarea la 1 va fi indicată prin apariția lui 0 la E_1 pentru toți vectorii de stimulare cu excepția perechii care selectează linia defectă,pentru că, în aceste situații, în permanență vor fi setate pe 1 două linii, cea defectă şi cea selectată. În fine, setul de teste B permite detecția defectelor singulare de blocare la 1 de pe liniile de bit punându-le în evidență la E_1 şi la E_2. Astfel, să admitem că linia de bit 1 este blocată la1 determinând conducția permanentă a tranzistorului conectat la intersecția cu linia de cuvânt 6 şi determinând astfel ca pe aceasta să se manifeste în permanență 0. Atunci când prin vectorii de stimulare 11 şi 12 din fig.4.46 se selectează linia de cuvânt 6, E_1 va rezulta 0 şi, prin faptul că vectorul binar de ieşire va deveni 000, va rezulta 0 şi la E_2. Menționăm, în final, că dacă structura FITPLA conține m linii de cuvânt, dintre care una este destinată controlului de paritate impară, atunci setul de teste B va conține 2m vectori de stimulare.

Single stuck faults			
Value	Location	Test set	Checker
0	Bit line	A	E_1
1	Bit line	B	E_1, E_2
0, 1	Word line	B	E_1
0, 1	Output line	B	E_2
0	x_i , y_i input	A	E_1
1	x_i , y_i input	B	E_1 , E_2
0	Output buffer	B	E_2

a.

Single crosspoint faults		
Location	Test set	Checker
AND array	A	E_1
OR array	B	E_2

b.

Fig.4.47

Concatenând cele două seturi de teste, A şi B, cu remarca păstrării pe durata aplicării setului de teste A a configurației binare constând doar din 0-uri în registrul de deplasare, în fig.4.47 este sintetizată detecției defectelor singulare de blocare la o anumită valoare logică (coloana value în fig.4.47) şi a defectelor tip punct de intersecție cu precizarea locației din structura (location) unde sunt considerate, precum şi a setului de testare care permite detecția lor şi a ieşirii de control, E_1 sau E_2, unde este observat efectul acestora.

În final, o mențiune aparte o dedicăm defectelor constând din scurtcircuite, care uzual sunt luate în analiză doar când se manifestă între două linii cu poziții adiacente. Sub acest aspect, ar trebui analizate următoarele cinci cazuri de scurtcircuite: (1) între două linii de bit; (2) între două linii de cuvânt; (3) între o linie de bit şi ieşirea de cuvânt; (4) între o linie de cuvânt şi o linie de ieşire; (5)între două linii de ieşire. Plecând de la faptul că în tehnologia nMOS, dacă una dintre liniile în

scurtcircuit este pe 0, ea va forța 0 și pe celelaltă prin seturile de teste A și B și această categorie de defecte poate fi detectată.[IyJI-90,SaBe-90].

4.2.3. Creşterea gradului de acoperire al defectelor potențiale pentru schemele FITPLA

Bazat pe cele expuse ca principii fiundamentale în prezentul paragraf, să analizăm structura propusă în [TrAF-87], care uzitează de strategia de testare independentă de funcții prin vectori de stimulare generați on chip. Remarcabil este, de asemenea, gradul ridicat de acoperire al defectelor potențiale-constând din detecția tuturor defectelor singulare (de tip blocare, scurtcircuit şi punct de intersecție) şi a mai mult decât $(1\text{-}2^{-(m+2n)})$ din defectele multiple, unde m şi n reprezintă numărul termenilor produs, respectiv cel al variabilelor de intrare,precum şi modul de comprimare al răspunsurilor de ieşire într-un şir de biți de paritate, independent de funcție, care este generat on-chip şi oferă o capacitate de trecere a setului de testare superioară în raport cu cea obtenabilă prin adaptarea structurii din fig.4.43 la condițiile de testare impuse noii structuri.

După cum rezultă din fig.4.48, deosebirile structurii FITPLA față de cele prezentate în fig.4.43 nu sunt spectaculoase.Astfel, trecând peste unele notații diferite, se observă prevederea pe intrările primare a unui registru de deplasare suplimentar SHR1, care prin încărcare serială, pe la intrarea de testare SHR1, a rangurilor Y_1 la Y_n să permită furnizarea seturilor, independente de funcții, de vectori binari pentru stimularea structurii. De asemenea, se impune ca, prin adăugarea unei linii noi de cuvânt (de fapt, o coloană suplimentară - extra column), atât numărul dispozitivelor (tranzistoare), cât şi cel al non-dispozitivelor (lipsa de tranzistoare) să fie impare. În fine, lipseşte schema de control de paritate impară conectată la liniile de cuvânt (cu ieşirea E_1-fig.4.43), permițând o compensare pentru aria de substrat alocată registrului de deplasare SHR1. În rest, apar aceleaşi linii de control suplimentare y_1şi y_2 din fig.4.43 notate în fig.4.48 prin c_1 şi c_2, pentru selecția prin forțare la 0 a unei linii de bit, şi acelaşi registru de deplasare în fig.4.48 cu intrarea serială SHR2, care permite selecția-prin S1 setat la 0 şi toate celelalte ranguri pe 1 -unei singure linii de cuvânt (i), precum şi schema de control la paritate impară conectată pe liniile de ieşire.

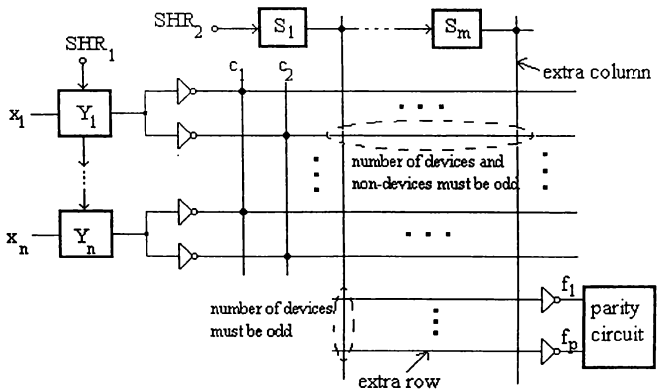


Fig.4.48

Modelele de biţi de stimulare sunt prezentate sugestiv în fig.4.49, unde indicele i variază între 1 şi n, iar indicele j între 1 şi m. Secvenţa de testare propusă, denumită secvenţă de testare universală (universal test sequence), este următoarea:

$$I^1 * \prod_{j=1}^{m} (I_j^2) * \prod_{i=1}^{n} \prod_{j=1}^{m} (I_{ij}^4) * \prod_{j=1}^{m} (I_j^3) * \prod_{i=1}^{n} \prod_{j=1}^{m} (I_{ij}^5) \qquad (4.6)$$

în care $\prod$ semnifică concatenarea repetată.

Inputs / Tests	x_1	x_{i-1}	x_i	x_{i+1}	x_n	c_1	c_2	S_1	S_{j-1}	S_j	S_{j+1}	S_m
I^1	0	0	0	0	0	0	1	1	1	1	1	1
I_j^2	0	0	0	0	0	0	1	1	1	0	1	1
I_j^3	1	1	1	1	1	1	0	1	1	0	1	1
I_{ij}^4	0	0	1	0	0	0	1	1	1	0	1	1
I_{ij}^5	1	1	0	1	1	1	0	1	1	0	1	1

Fig.4.49

În vederea evaluării răspunsurilor, fiecare vector de ieşire este comprimat într-un singur bi de paritate care este operat SAU EXCLUSIV cu un bit reprezentând paritatea cumulativă a anteriorilor vectori de ieşire pentru a obţine noul bit de paritate cumulativă (cumulative parity bit CPB). Examinând acest CPB doar de (2n+2m+1) ori, la momente de timp determinate, rezultă o foarte bună capacitate de trecere coroborată cu detecţia tuturor defectelor singulare şi a majorităţi celor multiple.

Definiţie4.1: Fie ca sintagma de comparare de paritate cumulativă să fie definită prin schem de comparare prezentată în fig.4.50, în care asteriscurile indică cele (2n+2m+1) momente când CPI este comparat cu valoarea normală, sperată (expected value). Pe duratele subseturilor I^1, I_j^2 şi I_j^3 CPB este comparat după fiecare ciclu de clock, rezultând un total de (2m+1) comparări.Pe durate subseturilor I_{ij}^4 şi I_{ij}^5, CPB este comparat numai după fiecare al m-lea ciclu de clock, în rest fiin ignorat, rezultând deci alte 2n comparări. Este de remarcat faptul că schema de comparare a parităţi cumulative este independentă de funcţie fiind o secvenţă de altenanţe de 0-uri şi 1-uri. Prin urmare, secvenţa de CPB pentru structura normală, fără defect, poate fi generată on-line printr-o schemă simplă şi , deci, nu trebuie memoraţi cei (2n+2m+1) biţi etalon. Dacă configuraţiile de stimulare s-a aplica într-o ordine diferită de secvenţa (4.6), schema de comparare ar rezulta mai complicată revendicând o arie de substrat mai substanţială. Biţii de paritate corespunzători funcţionării normal sunt deduşi după cum urmează:

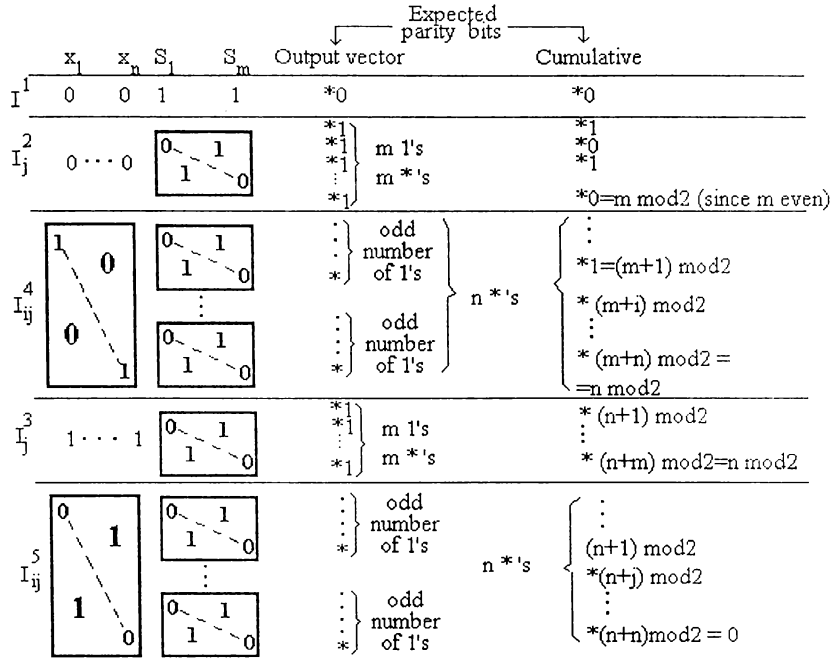


Fig. 4.50

I^1:Dacă structura PLA este fără defect, vectorul de ieşire constă dintr-o configuraţie binară de 0-uri a cărei paritate este 0.

I^2_j :Dacă structura PLA este fără defect, atunci fiecare din cei m vectori de ieşire au un număr impar de 1-uri întrucât fiecare linie de cuvânt are în matricea SAU un număr impar de dispozitive (tranzistoare), astfel încât paritatea fiecărui vector de ieşire este 1. În mod similar pentru I^3_j

I^4_{ij} :Dacă structura PLA este fără defect, să considerăm secvenţa de testare parţială $\prod_j(I^4_{ij})$ cu i rămânând constant, care activează numai punctele de intersecţie de pe linia de bit x_i. Dacă în punctul de intersecţie (x_i,j) există un dispozitiv, atunci linia de cuvânt j este setată pe 0, ceea ce produce un vector de ieşire configurat doar din 0-uri şi, prin urmare, bitul de paritate este 0. Dacă în punctul de intersecţie (x_i,j) nu există nici un dispozitiv, atunci linia de cuvânt j rămâne la 1, ceea ce produce un vector de ieşire cu un număr impar de 1-uri şi, prin urmare, bitul de paritate este 1.

Legat de acoperirea defectelor este enunţată următoarea teoremă:

Teoremă 4.1: Toate defectele singulare şi $(1\text{-}2^{-(m+2n)})$ din numărul total al defectelor multiple de tip crosspoint, stuck şi bridging sunt detectate prin schema de comparare a parităţii cumulative dacă structura PLA este stimulată prin secvenţa universală (4.6).

Demonstraţia este prezentată prin intermediul a opt leme în "stil domino", în sensul că la demonstrarea unor leme ulterioare se face uz de rezultatele din lemele anterioare. Aceasta se datorează faptului că există un înalt grad de echivalenţă între defecte, prin "echivalenţă" înţelegând

că, atunci când structura este stimulată printr-un set determinat de vectori de intrare, vectorii răspuns de ieşire sunt identici.

Lema 4.1: Toate defectele singulare şi $(1-2^{-2n})$ din numărul total al defectelor multiple de tip crosspoint din matricea ŞI pot fi detectate prin aplicarea subsecvenţei:

$$\prod_{i=1}^{n}\prod_{j=1}^{m}(I_{ij}^{4}) \text{ şi } \prod_{i=1}^{n}\prod_{j=1}^{m}(I_{ij}^{5}) \qquad (4.7)$$

Demonstraţie: Un defect crosspoint la intersecţia (x_i,j) inversează valoarea normală a liniei de cuvânt j şi, prin urmare, inversează şi paritatea ieşirii. I_{ij}^{4} şi I_{ij}^{5} activează toate cele m puncte de intersecţie de pe o linie de bit în vederea producerii unui nou CPB egal cu 1. Dacă numărul total de defecte de pe o linie de bit este par, CPB nu va semnala nici o anomalie întrucât numărul de dispozitive, respectiv cel de lipse de dispozitive, corespunzătoare respectivei linii rămânând ambele impare. Astfel, singurele defecte multiple nedetectabile sunt cele care sunt constituite dintr-un număr de defecte (sau nici un defect) de pe fiecare linie de bit.Numărul total par de defecte potenţiale (incluzând şi cazul fără defect) pentru o singură linie de bit este 2^{m-1}. Pe de altă parte, numărul total de defecte crosspoint potenţiale în matricea ŞI este 2^{2nm}, astfel încât fracţiunea defectelor crosspoint din matricea ŞI nedetectabile este $2^{(m-1)2n}/2^{2nm}=2^{-2n}$.

S-ar putea încerca îmbunătăţirea acoperirii defectelor specificate prin creşterea numărului de comparări ale CPB-ului pe durata subsecvenţelor I_{ij}^{4} şi I_{ij}^{5} de la n la nm, dar prin aceasta schema de comparare a CPB-ului ar deveni dependentă de funcţie ceea ce ar determina un consistent adaus de hardware în vederea memorării CPB-urilor normale pentru punctele de intersecţie individuale din matricea ŞI.

Lema 4.2: Toate defectele singulare şi $(1-2^{-m})$ din numărul total al defectelor multiple de tip crosspoint din matricea SAU pot fi detectate prin aplicarea uneia din subsecvenţele:

$$\prod_{j=1}^{m}(I_{j}^{2}) \text{ sau } \prod_{j=1}^{m}(I_{j}^{3}) \qquad (4.8)$$

Demonstraţie: Fiecare din secvenţele I_{j}^{2} şi I_{j}^{3} activează punctele de intersecţie din matricea SAU şi, similar cu situaţia tratată în lema 1, defectele multiple nedetectabile sunt determinate de un număr par de defecte.Prin analogie cu lema 1, fracţiunea defectelor crosspoint din matricea SAU nedetectabile este $2^{(p-1)m}/2^{pm}/\ 2^{-m}$.

Lema 4.3: Toate defectele singulare şi toate defectele multiple cauzate de dispozitive în exces în partea de decodificare a intrărilor sunt detectate prin aplicarea subsecvenţei:

$$\prod_{i=1}^{n}\prod_{j=1}^{m}(I_{ij}^{4}) \text{ şi } \prod_{i=1}^{n}\prod_{j=1}^{m}(I_{ij}^{5}) \qquad (4.9)$$

Demonstraţie: Un defect constând din prezenţa unui dispozitiv suplimentar la intersecţia (x_1,c_2) determină ca, pe durata lui I_{ij}^{4}, linia de bit x_i să fie permanent setată la 0. O comportare similară dar raportat la linia de bit $\overline{x_i}$ avem pe durata lui I_{ij}^{5} atunci când dispozitivul în exces este conectat la intersecţia $(\overline{x_i},c_1)$.Orice defect multiplu datorat unor dispozitive în exces în partea de decodificare a intrărilor este echivalent în contextul aplicării subsecvenţelor I_{ij}^{4} şi I_{ij}^{5} cu un defect

crosspoint multiplu în matricea ŞI, care este detectabil în baza lemei 1 determinând pe fiecare linie afectată de defect un număr impar de dispozitive neselectate, deci la compararea care succede baleierii respectivei linii se semnalează o neconcordanţă a biţilor de paritate, cel corespunzător funcţionării normale şi cel cumulativ obţinut ca urmare a subsecvenţei.

Lema 4.4: Toate defectele singulare şi multiple cauzate de omiteri de dispozitive în partea de decodificare a intrărilor sunt detectate prin aplicarea subsecvenţei:

$$\prod_{j=1}^{m}(\Gamma_j^2) \quad \text{şi} \quad \prod_{j=1}^{m}(\Gamma_j^3) \qquad (4.10)$$

Demonstraţie:Când un dispozitiv este omis la intersecţia (x_i,c_1), linia de bit x_i are aplicată, pe durata lui Γ_j^3, valoarea 1, comportare similară cu lipsa dispozitivului de la intersecţia $(\overline{x_i},c_2)$, când pe durata aplicării lui Γ_j^2, linia de bit $\overline{x_i}$ este permanent la 1. Orice defect multiplu constând din omitere de dispozitive în partea de decodificare a intrărilor este echivalent cu un defect crosspoint multiplu în matricea SAU, care este detectabil în baza lemei2 determinând pentru partea din matricea SAU a fiecărei linii de cuvânt afectate neselectarea unui număr impar de dispozitive, ceea ce duce la neconcordanţa parităţii cumulative cu cea sperată pentru funcţionarea normală.

Se remarcă faptul că pentru detecţia defectelor crosspoint din matricea SAU în baza lemei 2 este suficientă una din subsecvenţele Γ_j^2 sau Γ_j^3, pe când pentru detecţia defectelor crosspoint datorate omiterilor de dispozitive în partea de decodificare a intrărilor, în baza lemei 4, sunt necesare ambele subsecvenţe Γ_j^2 şi Γ_j^3.

Lema4.5: Defectele crosspoint multiple din matricea SAU nedetectabile nu pot masca, în contextul lui Γ_{ij}^4 şi Γ_{ij}^5, defecte crosspoint multiple din matricea ŞI detectabile. De asemenea, defectele crosspoint multiple din matricea ŞI nu pot masca, în contextul lui Γ_j^2 şi Γ_j^3, defecte crosspoint multiple din matricea SAU detectabile.

Demonstraţie: Γ_{ij}^4 şi Γ_{ij}^5 detectează defecte de matrice ŞI când partea din matricea SAU a fiecărei linii de cuvânt are un număr impar de dispozitive. Defectele de matrice SAU nedetectabile cauzează apariţia sau dispariţia în fiecare linie de cuvânt a unui număr par de dispozitive, prin urmare numărul total de dispozitive pe fiecare linie de cuvânt rămâne impar, astfel că se evită fenomenul de mascare.

Lema4.6: Toate defectele de blocare (stuck-at) sunt detectate prin schema de comparare a parităţii cumulative.

Demonstraţie: Trebuie luate în considerare şase cazuri:

a)O linie de ieşire blocată la 0: acest defect este detectat prin Γ^1.

Celelalte cinci cazuri de defecte de blocare sunt echivalente cu defecte multiple de tip dispozitive omise (crosspoint) fiind, prin urmare, detectabile în baza lemelor 1 şi 2.

b)O linie de ieşire blocată la 1: este echivalentă cu lipsa , în contextul lui Γ_j^2 şi Γ_j^3, a tuturor dispozitivelor de pe respectiva linie de ieşire.

c)O linie de cuvânt blocată la 0: nu are nici un efect asupra liniilor de ieşire, întrucât acest defect singular este echivalent cu absența , în contextul lui $\int_j^2$ şi $\int_j^3$, a tuturor dispozitivelor din partea de matrice SAU a respectivei linii de cuvânt.

d)O linie de cuvânt blocată la 1:echivalează cu absența, în contextul lui $\int_j^2$ şi $\int_j^3$, a tuturor dispozitivelor din partea de matrice ŞI a respectivei linii de cuvânt.

e)O linie de bit blocată la 0: nu are nici un efect asupra liniilor de cuvânt, întrucât acest defect singular este echivalent cu absența, în contextul lui $\int_{ij}^4$ şi $\int_{ij}^5$, a tuturor dispozitivelor de pe respectiva linie de bit.

f)O linie de bit blocată la 1: întrucât toate liniile de cuvânt conectate cu linia de bit blocată la 1 vor fi forțate la 0, acest defect singular echivalează cu blocarea la 0 a tuturor liniilor de cuvânt afectate (adică cu aplicarea repetitivă a mai sus menționatului caz c).

Lema4.7: Cel puțin $(1-2^{-(m+2n)})$ din numărul total de defecte multiple de blocare pot fi detectate prin schema de comparare a parității cumulative.

Demonstrație: Trebuie luate în considerare trei cazuri.

Cazul (c+e+f): Cele trei tipuri de defecte linie de cuvânt blocată la 0, linia de bit blocată la 0 şi linia de bit blocată la 1-combinate în defecte de blocare multiple echivalează întotdeauna cu defecte crosspoint multiple întrucât atât liniile de bit, cât şi partea de matrice SAU corespunzătoare liniilor de cuvânt au doar un număr impar de defecte crosspoint.

Cazul (b+d): Când se combină un defect de linie de ieşire blocată la 1 cu un defect de linie de cuvânt blocată la 1 dând un defect multiplu de blocare, atunci $(1-2^{-(m+2n)})$ din numărul total al acestora echivalează cu defecte crosspoint multiple cu numere impare de defecte de linie de bit, respectiv de linie de cuvânt.

Cazul (a+): În contextul lui $\int_j^2$ şi $\int_j^3$, defectul de linie de ieşire blocată la 0 echivalează defectul multiplu de dispozitive în exces, fiecare linie de ieşire defectă având câte un dispozitiv în fiecare dintre punctele sale de intersecție,moment din care activează lema2.

Lema4.8: Toate defectele bridging singulare şi cel puțin $(1-2^{-(m+2n)}$ din numărul total al celor multiple sunt detectate prin schema de comparare a parității cumulative.

Demonstrație: Trebuie luate în considerare cele cinci cazuri relevate în finalul paragrafului precedent. Primele patru cazuri sunt echivalente cu defecte de blocare.

1)Două linii de bit scurtcircuitate echivalează, în contextul lui $\int_{ij}^4$ şi $\int_{ij}^5$ şi datorită dominanței de 0 în tehnologie n-MOS, cu blocarea la 0 a respectivelor linii de bit, detecția bazându-se în continuare pe lemele 6-e şi 7-c,e şi f.

2)Două linii de cuvânt în scurtcircuit echivalează, în contextul lui $\int_j^2$ şi $\int_j^3$ şi a dominanței de 0, cu blocarea la 0 a respectivelor linii de cuvânt, detecția tuturor defectelor bridging singulare şi multiple a perechilor de linii de cuvânt fiind asigurate de lemele 6-c şi 7- c,e,şi f.

3)În contextul lui $\int_j^2$ şi $\int_j^3$, toate liniile de bit sunt setate în mod normal, la 0, astfel încât orice linie de cuvânt în scurtcircuit cu linii de bit echivalează cu blocarea la 0 a liniilor de cuvânt, moment din care acționează din nou lemele 6-e şi 7-c, e şi f.

4)În contextul lui $\prod_j^1$, $\prod_j^2$ şi $\prod_j^3$, liniile de cuvânt sunt aproape în permanenţă pe 0, astfel încât orice linie de ieşire în scurtcircuit cu linii de cuvânt echivalează aproape în permanenţă cu blocarea la0 a liniilor de ieşire, moment din care se apelează la lemele 6-a şi 7-a.

5)Dacă două linii de ieşire, admitem A şi B, sunt scurtcircuitate, atunci valoarea lor comună V este obţinută ca urmare a aplicării unei operaţii logiceŞI între A şi B. În contextul lui $\prod_j^2$ şi $\prod_j^3$, V are valoarea 1 numai dacă nici una dintre liniile A şi B nu are un dispozitiv conectat pe linia de cuvânt activată în mod curent. Astfel, acest defect bridging echivalează cu un defect crosspoint multiplu conform căruia atât A, cât şi B,au peste tot dispozitive în exces, cu excepţia liniilor de cuvânt de-a lungul cărora nici A şi nici B nu au avut original nici-un dispozitiv.În continuare, se uzează de lema 2.

Se mai menţionează că aproape toate defectele fizice din hardware-ul adăugat sunt puse în evidenţă prin schema de comparare a parităţii cumulative. Astfel, inspectând fig.4.48, este evident că defecte în partea adăugată la decodificatorul intrărilor şi la registrul de deplasare SHR_2 de pe liniile de cuvânt vor determina ca liniile de bit , respectiv linii de cuvânt, să obţină uneori valori eronate şi, prin urmare, aceste defecte sunt echivalente cu diferite defecte de linii de bit sau de linii de cuvânt. În mod similar, defecte la schema de comprimare prin bit de paritate a vectorilor răspuns vor determina generarea uneori de parităţi eronate, astfel încât ele se comportă ca unele defecte de linii de ieşire.

4.2.4.Intercalarea structurilor ASIM în scheme FITPLA

Justificăm prezentările detaliate ale configuraţiilor FITPLA din paragrafele anterioare prin faptul că evaluarea răspunsurilor se bazează pe comprimarea acestora în unul sau doi biţi de paritate, iar structurile ASIM implementează, în fond, un control de paritate extins.Să păstrăm caracterisicile structurii din fig.4.48 în ceea ce priveşte partea de stimulare şi să substituim partea de evaluare a răspunsurilor printr-o structură ASIM modificată în scopul creşterii capacităţii de detecţie conform cu cele expuse în capitolul trei.Rezultă schema prezentată în fig.4.51, la care, faţă de cea din fig.4.48, există unele modificări.

În ceea ce priveşte partea de stimulare, atât în fig.4.43 [Prad-86], în fig.4.48[TrAF-87], se disting cele două coloane suplimentare c_1 şi c_3 precum şi registrul de deplasare SHR_2.În fig.4.48 apare, în plus, registrul de deplasare SHR_1, care permite "plimbarea" unui 0 în câmp de 1-uri, respectiv a unui 1 în câmp de 0-uri.În realitate, şi testarea schemei din fig.4.43 ar necesita registrul SHR_1 built-in dar, prin faptul că în procesul de testare se uzitează mai puţin frecvent la aceste configuraţii binare faţă de utilizarea lor în secvenţa (4.6), s-a considerat aplicarea lor off-chip. În această parte de stimulare, preluăm în schema propusă varianta mai completă de stimulare din [FrAF-87].

Pe de altă parte, în ceea ce priveşte partea de evaluare a răspunsurilor, la schema din [Prad-86] se adaugă o coloană suplimentară-care asigură, prin proiectare, un număr impar de dispozitive pe fiecare linie din matricea ŞI-, o linie suplimentară în matricea SAU- care asigură, prin proiectare, un număr impar de dispozitive pe porţiunile din matricea SAU a fiecărei coloane-, precum şi doi arbori de circuite SAU EXCLUSIV care permit calculul a doi biţi de paritate. La schema din [FrAF-87],

redundanța implică adausul, prin proiectare, de una sau , în cel mai defavorabil caz, a două coloane suplimentare astfel încât fiecare linie din matricea ŞI să conțină numere de dispozitive şi de non-dispozitive şi, în compensație, renunțarea la arborele de circuite SAU EXCLUSIV de calcul a parității pe coloane.Linia suplimentară în matricea SAU, ca şi arborele de circuite SAU EXCLUSIV de calcul a parității de ieşire, rămân aceleaşi ca în [Prad-86], iar,cu o investiție minimă, se mai adaugă de calcul a parității cumulative, în fond un numărător de tranziții, modulo 2.Referindu-ne la schema propusă (fig.4.51) este de remarcat faptul că lipsesc modificări în cele două matrici, neadăugându-se nici coloane (excepție fac cele două, c_1 şi c_2, din partea de stimualare),nici linii, matricile rămânând în forma lor convențională.Se suplimentează însă o structură ASIM de lungime p (fig.4.51-se întrerupe conexiunea marcată cu x şi se înlătură ultimele m ranguri cu circuitele auxiliare aferente şi apare ca obligatorie conexiunea de reacție marcată cu xx) sau de lungime (p+m)(fig.4.51, cu toate elementele), unde p reprezintă numărul funcțiilor de ieşire, iar m pe cel al coloanelor schemei.

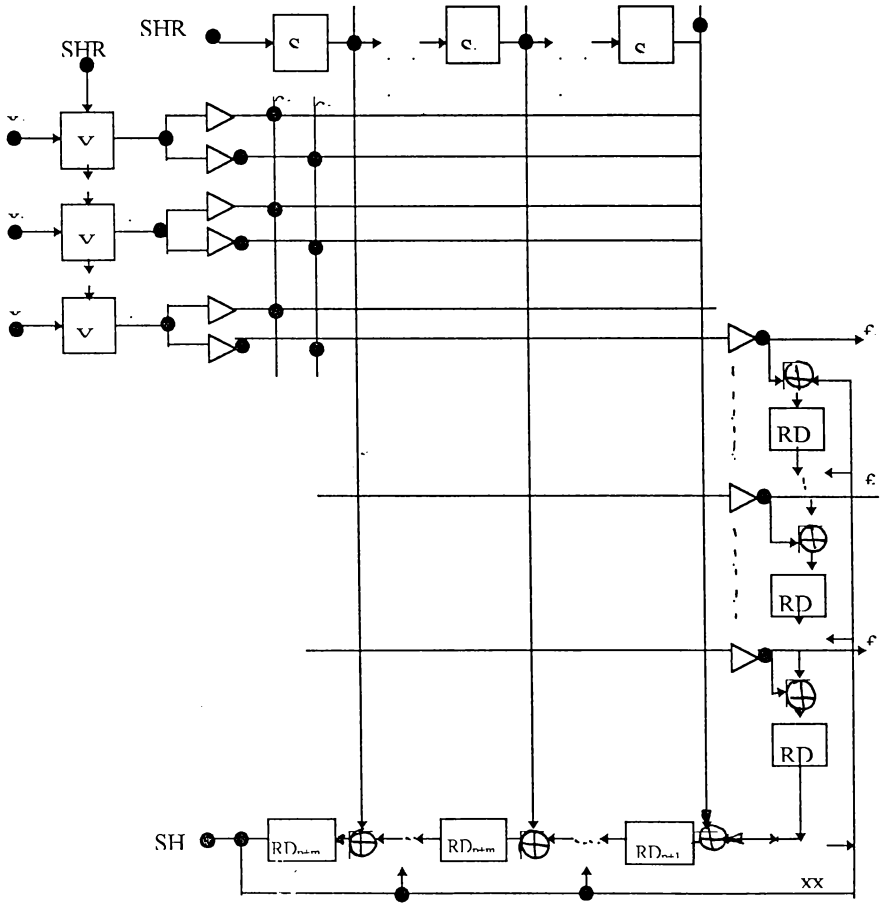


Fig.4.51

În baza celor dezvoltate teoretic în capitolul trei, pentru sinteza structurii se va uzita de un polinom generator cu expresie primitivă de grad p-situație în care notăm structura ASIM-E-sau de grad (p+m)-situație în care notăm structura ASIM-EC.Fără a pierde din generalitate, vom considera în ambele situații menționate că structurile ASIM sunt de tipul cu circuitele SAU EXCLUSIV conectate între rangurile registrului de deplasare, având reacțiile sugerate punctat în fig.4.51.O apreciere comparativă prin prisma surplusului de circuistică între diferitele scheme este dificil de realizat datorită multitudinii de parametri-valorile lui n, m şi p, precum şi expresiile particulare ale funcțiilor de ieşire-, dar este destul de evident că schema propusă este dezavantajoasă, aspect care trebuie însă arbitrat luând în considerare posibilitățile tehnologice actuale referitoare la densitatea de împachetare a circuitelor integrate, ceea ce atenuează costurile în exces.

Pentru a întregi comparația, pe lângă aspectele de cost se impun balansate şi cele legate de performanță, exprimată prin capacitatea de trecere a experimentului de testare şi prin capacitatea de detecție a potențialelor defecte.Astfel,referindu-ne la versiunea din [Prad-86] şi considerând, în scopul efectuării unei paralele între metode, că la toate versiunile este disponibil un SHR_1 poziționat inițial în paralel şi apoi serial cu secvența de stimulare şi, de asemenea, admițând că o operație de comparație revendică un ciclu de clock distinct, pentru capacitatea de trecere CT_1 exprimată de cicluri de clock, obținem:

$$CT_1=(4n+4(m-1)+m+3)CC=(4n+5m+1)CC \qquad (4.11)$$

Termenii din relația (4.11) au fost obținuți pe seama următorului raționament:

a) setul de teste A implică câte un CC pentru aplicarea unui stimul (am considerat, de asemenea, că SHR_2 este antepoziționat momentului de evaluare prin (4.9) la configurația 00....0) şi un CC pentru comparația la E_1 şi E_2,deci 2CC pentru un vector stimul, înmulțit cu 2n vectori din setul A, conduce la 4n CC.

b) setul de teste B implică nCC de poziționare inițială a lui SHR_2 concurent cu care se poziționează şi primul vector în SHR_1, după care avem un CC pentru comparare, un proxim CC pentru repoziționarea lui SHR_1şi, în fine, mCC a doua comparare corespunzătoare primei configurații binare din SHR_2, prin urmare, rezultând (m+3)CC . În continuare, fiecare configurație binară nouă din SHR_2 necesită un CC pentru poziționarea SHR_1 şi SHR_2, un al doilea CC pentru comparare, apoi al treilea CC pentru poziționarea SHR_1 şi al patrulea CC pentru comparare, deci 4CC şi aceasta pentru fiecare din cele (m-1) configurații din SHR_2.

Drept consecință a procesului de testare descris, schema din [Prad-86] asigură acoperirea tuturor defectelor singulare de blocare şi crosspoint.

Păstrând condițiile ipotetice specificate anterior, să evaluăm capacitatea de trecere CT_2 a schemei din [TrAF-87] atunci când aceasta este stimulată prin secvența (4.4) şi este evaluată prin compararea bitului de paritate cumulativă.

$$CT_2=(1+2(3m-1)+4mn)CC=(4mn+6m-1)CC \qquad (4.12)$$

Pentru obținerea termenilor din relația (4.12) s-a uzitat de următorul raționament:

a) setul I^1 revendică o setare de vector stimul şi o comparare , deci 2CC

b) seturile I_j^2 şi I_j^3 necesită mCC de setare inițială a SHR_2, concurent cu setarea lui SHR_1, urmat de un CC pentru comparare, şi , în continuare, pentru fiecare vector de stimulare sunt necesare 2CC, unul de deplasare a lui SHR_2 şi unul de comparare, astfel încât rezultă ca necesare (m+1+2(m-1))=3(m-1)CC pentru I_j^2 şi I_j^3 în, parte.

c)având în vedere doar cele n comparări efectuate în I_j^4 şi, de asemenea în I_j^5, pentru stimulare sunt necesare (m+(m-1))CC=(2m-1)CC pentru fiecare vector de intrare, deci, în total (n(2m-1)+n)CC=2mnCC pentru I_j^4 şi I_j^5 în parte.

Capacitatea de detecțieCT_C cu valoare mult mai mare decât expresia (4.11) se justifică printr-un grad de acoperire mult mai ridicat al defectelor potențiale, asigurând pe lângă detecția defectelor singulare şi pe cea a unui raport de $(1-2^{-(m+2n)})$ din cele multiple.

În ceea ce priveşte schema propusă, premergător combinării capacității de trecere, sunt necesare unele precizări.Astfel,prin extensie la definiția 4.1, apelăm la:

Definiție 4.2: Se înțelege prin comparare de paritate cunmulativă multiplă (CPCM) verificarea întreprinsă prin p sau (p+m) cicluri de tact la ieşirea serială suplimentară SHR_3 a semnăturii generate în paralel de către ASIM la sfârşitul aplicării trenului de vectori stimuli dat de secvența universală (4.6).

Uzitând de definiția introdusă, să caracterizăm gradul de acoperire al defectelor specific noii scheme,în care sens enunțăm următoarea teoremă originală.

Teorema 4.2: Stimulând schema PLA prin secvența universală (4.4) şi uzitând de evaluarea răspunsurilor prin CPCM, toate defectele singulare, precum şi $(1-2^{-(m+2n)})$ din numărul total al celor multiple de tip crosspoint, stuck şi bridging sunt detectate cu o probabilitate de 2^{-p} la utilizarea ASIM-E, respectiv $2^{-(p+m)}$ la utilizarea ASIM-EC.

Demonstrație: Trebuie remarcat faptul că schema fiind stimulată prin aceiaşi secvență (4.4), demonstrația poate fi întreprinsă în acelaşi"stil domino" uzitând de adaptări la noua metodă de evaluare ale celor opt leme 4.1 la 4.8 folosite la demonstrația teoremei 4.1.Pentru partea enunțului legată de probabilități se rematcă uzitarea de conectări ale reacțiilor ASIM în conformitate cu cele demonstrate amplu în capitolul trei.

Fără a mai insista asupra demonstrațiilor, care, luând în considerare şi caracteristicile ASIM-urilor, sunt asemănătoare cu cele prezentate în paragraful 4.2.3,să enunțăm cele opt leme în versiunile lor adaptate.

Lema 4.9: Evaluând răspunsurile prin CPCM, sunt detectate toate defectele singulare, precum şi, cu o probabilitate de recunoaştere ca funcțional corectă a unei unități testate defecte de 2^{-p} pentru ASIM-E, respectiv $2^{-(p+m)}$ pentru ASIM-EC, o proporție de $(1-2^{-2n})$ din numărul total al defectelor multiple de tip crosspoint din matricea ŞI, în situația că schema PLA este stimulată prin secvența:

$$\prod_{i=1}^{n}\prod_{j=1}^{m}(I_{ij}^4) \text{ şi } \prod_{i=1}^{n}\prod_{j=1}^{m}(I_j^5) \qquad (4.13)$$

Lema 4.10: Evaluând răspunsurile prin CPCM, sunt detectate defectele singulare, precum şi, cu o probabiliatate de recunoaştere ca funcțional corectă a unei unități testate defecte de 2^{-p} pentru

ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC, o proporție de $(1-2^{-m})$ din numărul total al defectelor multiple de tip crosspoint din matricea SAU, în situația că schema PLA este stimulată prin una din secvențele:

$$\prod_{s=1}^{m}(I_j^2) \text{ sau } \prod_{j=1}^{m}(I_j^3)$$

Lema 4.11: Evaluând răspunsurile prin CPCM, sunt detectate toate defectele singulare, precum şi, cu o probabilitate de recunoaştere ca funcțional corectă a unității testate defecte de 2^{-p} pentru ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC, toate defectele multiple cauzate de dispozitive în exces în partea de decodificare a intrărilor în situația că schema PLA este stimulată prin secvența (4.13).

Lema 4.12: Evaluând răspunsurile prin CPCM, sunt detectate toate defectele singulare, precum şi, cu o probabilitate de recunoaştere ca funcțional corectă a unei unități testate defecte de 2^{-p} pentru ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC, toate defectele multiple cauzate de omiteri de dispozitive în partea de decodificare a intrărilor, în situația că schema PLA este stimulată prin secvența:

$$\prod_{j=1}^{m}(I_j^2) \text{ şi } \prod_{j=1}^{m}(I_j^3)$$

Lema 4.13: Stimulând schema PLA prin secvența (4.13) şi evaluând răspunsurile prin CPCM, defectele crosspoint multiple din matricea SAU nedetectabile pot masca defecte crosspoint multiple din matricea ŞI detectabile cu o probabilitate de 2^{-p} pentru ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC.De asemenea, în aceleaşi condiții de evaluare, dar stimulând schema PLA prin secvența (4.15), defectele crosspoint multiple din matricea ŞI nedetectabile pot masca defecte crosspoint multiple din matricea SAU detectabile cu aceleşi probabilități specificate anterior.

Lema 4.14: Stimulând schema PLA prin secvența universală (4.4) şi evaluând răspunsurile prin CPCM, sunt detectate toate defectele de blocare (stuck-at) singulare.

Lema 4.15: Stimulând achema PLA prin secvența universală (4.4) şi evaluând răspunsurile prin CPCM, este detectată, cu o probabilitate de recunoaştere corectă a unei unități testate defecte de 2^{-p} pentru ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC, o proporție de cel puțin $(1-2^{-(m+2n)})$ din numărul total al defectelor de blocare.

Lema 4.16: Stimulând schema PLA prin secvența (4.4) şi luând răspunsurile prin CPCM, sunt detectate toate defectele bridging singulare şi, cu o probabilitate de recunoaştere ca funcțional corectă a unei unități testate defecte de 2^{-p} pentru ASIM-E, respectiv de $2^{-(p+m)}$ pentru ASIM-EC, o proporție de cel puțin $(1-2^{-(m+2n)}$ din numărul total al defectelor bridging multiple.

Este evident că atunci când numărul de funcții este considerabil, admitem mai mare decât 16, chiar şi prin utilizarea doar a unui ASIM-E, degradarea de performanță referitoare la capacitatea de detecție este infimă, mai mică de 0,002%.Cu atât mai mult, la folosirea unui ASIM-EC, la valori n ,m şi p specifice circuitelor actuale, degradarea de capacitate de detecție o putem considera nesemnificativă față de schema din [TrAF-87] şi, evident,consistent superioară celei din [Prad-86]. Extinzând, în continuare, raționamentele comparative şi asupra capacității de trecere şi luând în

considerare faptul că cele (2m+2n+1)CC necesare la schema [TrAF-87] sunt substituite cu p CC la ASIM-E, respectiv (p+m)CC la ASIM-EC, rezultă pentru capacitatea de trecere CT_3 a noii scheme la utilizarea ASIM-E:

$$CT_3=(4mn+6m-1-2m-2n-1+p)CC=(4mn+4m-2n+p-2)CC \qquad (4.14)$$

În mod similar, pentru capacitatea de trecere CT_4 corespunzătoare unei noi scheme în care se uzitează de ASIM-EC, rezultă:

$$CT_4=(4mn+6m-1-2m-2n-1+p+m)CC=(4mn+5m-2n+p-2)CC \qquad (4.15)$$

Consultând cataloage de circuite integrate PLA, spre exemplu [Mullard-Technical Handbook-Book4-Integrated Circuits-Part 7a-Programable Logic Devices 1992], se poate observa că, în general, se poate considera m>>n şi m>>p, precum şi n≅p.În aceste condiţii, avem în urma analizei raportului $(2m+2n+1)/p\cong 2(m/p)$ conturul îmbunătăţirii pe care o aduce referitor la capacitatea de detecţie noua schemă şi care este cu atât mai consistentă cu cât m este mai mare decât p. În acelaşi context, comparativ cu CT_1 dat de relaţia (4.11), atât CT_3 dat de relaţia (4.16), cât şi, cu atât mai mult, CT_4 dat de relaţia (4.15) prezintă valori mult mai mari, dar noua schemă, propusă, posedă, în compensaţie, o mult mai avansată capacitate de detecţie exprimată prin teorema (4.2).

Concluziv, se poate conchide, admiţând că arbitru triplului impact cost(exprimat prin număr de circuite elementare revendicate de sinteză)/performanţă(exprimată prin numărul de cicluri de clock necesare trecerii experimentului de testare)/capacitate de detecţie a defectelor(exprimată prin probabilitatea de recunoaştere ca funcţional corectă a unei unităţi testate defecte relativ la defectele singulare sau multiple de tip blocare,bridging şi crosspoint), strategia built-in test bazată pe testare independentă de funcţii încercăm afirmaţia că este mai bună decât cele analizate comparativ, cu rezerva, că s-ar impune o filtrare mai adâncă a afirmaţiei luând în considerare particularităţile procesului tehnologic a unui număr fabulos de circuite integrate.Mai menţionăm, că rezultatele pozitive obţinute pentru schema PLA(şi FPLA) lasă să se întrevadă perspective favorabile de extensie la mai modernele scheme GA (şi FPGA) a structurilor ASIM propuse.

4.3. Utilizarea Structurilor ASIM modificate în scopul facilitării autocontrolului schemelor UAL

4.3.1. Detecţia şi corecţia erorilor la operaţii aritmetice prin coduri bazate pe paritate

Urmărind reliefarea unor noi faţete ale aplicării structurilor ASIM modificate care să permită configurarea unor sisteme de calcul cu toleranţă la defecte, prezentul paragraf îşi propune pătrunderea într-o zonă de circuistică, cu precădere, delicată prin prisma supravegherii corectei funcţionări, prin faptul că efectul de manifestare prin eroare a potenţialului defect este mai frecvent supus mişcării decât în alte componente de structură. Ne referim la acea parte a hardware-ului uzual cunoscută ca unitate aritmetică şi logică constituind partea esenţială a unităţii de prelucrare a datelor din unitatea centrală de procesare a unui sistem de calcul [Haye - 88, Pa HE - 90]. Referindu-ne, în

mod sintetic, la orientările existente în domeniul aritmeticii soluțiile se extind într-un evantai având la un capăt conceptul de limitare la un minim indisponibil a hardware-ului - implicând implementarea în circuite doar a instrucțiilor de adunare şi scădere binară, eventual de înmulțire şi împărțire, toate în virgulă fixă - şi la celălalt capăt conceptul de procesare aritmetică auxiliară - implicând circuite integrate specializate, independente de cele corespunzătoare unităților centrale de procesare, dar care împreună cu acestea devin capabile să execute instrucții aritmetice complexe, care la cele menționate mai sus adaugă unele de adunare, de scădere, de înmulțire şi de împărțire asupra numerelor reprezentate în formate de virgulă mobilă, precum şi unele de ridicare la putere, de extragere de rădăcină pătrată, de exponențiere, de logaritmare, de evaluare a funcțiilor trigonometrice ca sinus, cosinus, tangentă, arcsinus, arctangentă, arcsinus hiperbolic s.a. [ChCh-91, KINK-94]. Conceptul de procesare aritmetică auxiliară este materializat prin (a) procesoare aritmetice periferice, pe care unitățile centrale de procesare le percep ca dispozitive de intrare-ieşire periferice ordinare cu care comunică prin instrucții de transfer de date - pentru transmiterea funcțiilor aritmetice de executat şi a operanzilor în sensul înspre procesorul aritmetic periferic şi retransmiterea de către acesta a rezultatelor -, care trebuie programate în mod explicit în fluxul de instrucții al unității centrale gazdă (în calitate de exemplu - circuitul integrat AMD 9511/12 [Haye - 88]) şi (b) coprocesoare aritmetice, capabile ca, împreună cu unitățile centrale de procesare pentru care sunt "croite", să execute intrucții pentru funcții aritmetice complexe (în calitate de exemplu, I80387 pentru I80386, I80487 pentru I80486, MC 68881 pentru MC 68020 s.a.) [ChIy - 89, Sh GI - 89, Joha - 93, BlPe - 94].

Pentru a satisface funcțiile suplimetare de detecție şi corecție a erorilor, revendicate de conceptul de tolerare a defectelor, procesarea aritmetică şi logică necesită codificarea redundantă a operanzilor, o soluție în acest sens reprezentând-o apelarea la aşa numitele coduri aritmetice, având la baza codificării fie operația de înmulțire, fie cea de împărțire. [RaFu-89,Park-90, Brya-91]. Mecanismele de detecție şi corecție sunt dependente de tipul erorilor potențiale, care pot fi singulare, multiple, de byte singular sau unidirecționale, fiind implicate coduri specifice diferind în circuistica ca redundantă solicitată pentru implementare [BrSt-88, NaAb-90,WoGo-94], sub care aspect precizăm limitarea considerațiilor ulterioare din lucrare la erorile singulare. Fără a insista asupra codurilor aritmetice, arătăm totuşi că un cod multiplicativ se poate obține prin intermediul unui număr, întreg A, numit şi modul, care este utilizat pentru înmulțirea, care precede operația de verificat, a celor doi operanzi N_1 şi N_2 , pe care îi admitem, de asemenea, ca numere întregi. Se obțin noi operanzi codificați AN_1 şi AN_2 care sunt supuşi operației care, dacă este, spre exemplu, adunarea, atunci conduce la rezultatul (AN_1 + A N_2), a cărui corectitudine se verifică prin împărțirea la modulul A utilizat la codificare. Dacă această operație suplimentară se soldează cu rest 0, atunci este validată operația de adunare verificată, iar obținerea unui rest nenul este interpretat ca detecție de eroare, astfel încât punerea în evidență a erorii singulare necesită ca A să fie diferit de o putere a lui 2. Faptul că verificarea se bazează pe un cod AN care este nesistematic, precum şi redundanța care o solicită, cu precădere, la operații mai complexe decât adunarea, determină răspândirea practică mai strânsă a acestui tip de cod aritmetic - precum şi a formei sale evoluate în cod (AN+3), care este înzestrat cu

proprietatea de autocomplementaritate [Prad-86,RaFu-89] - în raport cu clasa aşa numitelor coduri cu resturi (rasidue codes), la care codificarea operanzilor se bazează pe operaţia de împărţire. Într-o descriere sintetică, fiind dat un număr întreg şi pozitiv N reprezentat în formă binară, prin împărţirea acestuia la un alt număr întreg pozitiv A, numit modul se obţine astfel restul R = N mod A, care concatenat la N în forma (NR) formează cuvântul codificat. Notând cu N_1 şi N_2 doi operanzi întregi şi pozitivi cu resturile asociate R_1 şi R_2 obţinute prin împărţirea la acelaşi modul A, verificarea operaţiilor aritmetice se bazează pe următoarele proprietăţi ale codurilor cu resturi [Beck-88, Ra Fu - 89, Kant-93, St FK-93, Hilj-94]:

$$(N_1 \pm N_2) \bmod A = (R_1 \pm R_2) \bmod A \qquad (4.16)$$

$$(N_1 \cdot N_2) \bmod A = (R_1 \cdot R_2) \bmod A \qquad (4.17)$$

În mod schematic, în fig. 4.52 se prezintă blocurile care intervin la execuţia operaţiilor care intervin în (4.16) şi (4.17), sugerând asupra investiţiei în circuite pe care o solicită detecţia erorilor bazată pe coduri cu resturi.

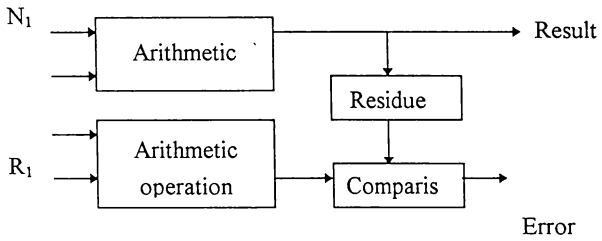


Fig. 4.52

Astfel, se observă că, pe de o parte sunt operate cele două numere N_1 şi N_2 , fiind calculat restul rezultatului, şi, pe de altă parte, aceleiaşi operaţii sunt supuse resturile R_1 şi R_2 , calculate în prealabil. Coincidenţa la nivel de bit a celor două resturi indică execuţia corectă a operaţiei, iar necoincidenţa chiar şi a unui singur bit conduce la semnalarea unei erori. Din păcate, o proprietate asemănătoare cu (4.16) şi (4.17) nu este valabilă şi pentru operaţia de împărţire, pentru care, în baza identităţii specifice acestei operaţii, se poate însă scrie:

$$N_2 = QN_1 + S \qquad (4.18)$$

în care Q şi S reprezintă câtul, respectiv restul obţinut la împărţirea lui N_2 şi N_1.

Calculând, pe lângă resturile R_1 şi R_2 corespunzătoare celor doi operanzi N_1 şi N_2 , si resturile R_Q şi R_S corespunzătoare rezultatelor Q şi S, toate utilizând acelaşi modul împărţitor A, detecţia erorilor la operaţia de împărţire se poate baza pe următoarea relaţie, obţinută prin adaptarea anterioarei identităţi (4.18):

$$(R_2 - R_S) \bmod A = (RQ_1 \cdot R_2) \bmod A \qquad (4.19)$$

Evident, supravegherea corectei execuţii a operaţiei prin (4.19) diferă principal de cele efectuate din (4.16) şi (4.17), implicând calcule consistente, consumatoare de timp, care trebuie să succeadă execuţiei operaţiei verificate. La acest neajuns legat de împărţire, fără a avea pretenţia acoperirii problematicii, în acelaşi context al controlului prin coduri cu resturi, mai dorim să relevăm unele aspecte care se constituie în dificultăţi de implementare, în vederea rezolvării, penalităţi suplimentare de performanţă şi / sau cost [RaFu - 89, PhVa - 94]. Astfel, la adunarea, spre exemplu, numerelor binare întregi N_1 şi N_2 poate rezulta o sumă, care - având în vedere că registrele prezintă un număr determinat de ranguri permiţând reprezentarea numărului cel mai mare, să zicem M - poate să depăşească capacitatea registrului, fiind obţinută, în locul sumei (N_1 - N_2) dorite, valoarea (N_1 + N_2 - M). Calculând restul modulo A pentru aceasta din urmă, el nu este în mod necesar egal cu (R_1 + R_2) mod A, aşa cum o cere relaţia (4.16) ceea ce necesită restricţionarea valorii modulului A pentru ca totuşi detecţia erorilor să poată fi totuşi efectuată. Într-adevăr, dacă M este un multiplu de A, atunci controlul poate fi extins şi asupra situaţiei de depăşire a capacităţii registrelor, fiind bazat pe următoarea relaţie:

$$(N_1 + N_2 - M) \bmod A = (R_1 + R_2) \bmod A \qquad (4.20)$$

Un alt aspect deficitar al utilizării codurilor cu resturi apare atunci când, din raţiuni de economie a costului de circuistică suplimentară, se apelează, în detrimentul performanţei, la o structură verticală de implementare a verificării sugerate prin fig. 4.52, bazată pe operare, byte-sliced, şi când o anumită poziţie binară, afectată de către defect, provoacă, în mod repetat, eronarea respectivului bit din mai multe bytes. De exemplu, să considerăm cuvântul de cod cu rest 0010001101011010, în care cele mai semnificative 12 poziţii binare corespund biţilor informaţionali utili (N_{10} = 565), iar ultimele 4 poziţii binare, mai puţin semnificative, corespund restului modulo A = 15 (R_{10} = 10). Presupunând că acest cuvânt de cod este operat pe felii (slicos) constituite de bytes de 4 biţi şi că ieşirea celui mai puţin semnificatic este afectată în permanenţă de defectul constând din blocarea sa la 1 logic, atunci ultima poziţie a bytes este eronată în mod repetat, obţinând cuvântul 0011* 0011 0101 1011*, în care prin asterisc (*) au fost marcate erorile provocate de către defect. Noul vector binar - prin faptul că partea sa utilă reprezintă numărul N_{10} = 821 şi restul calculat modulo 15 reprezintă numărul R_{10}=11 - constituie un cuvânt de cod valid, astfel încât prezenţa defectului nu este detectată. Evitarea acestei situaţii de anomalie implică apelarea la aşa numite coduri cu resturi inverse, cu corespunzătoare investiţii suplimentare [Dera-88,PaFu-89]. De altfel, ca o caracteristică generală a implementării, reale a codurilor cu resturi este dependenţa decisivă a complexităţii părţii de aritmetică de alegerea modulului A, de care depinde, de asemenea, capacitatea de detecţie. Calculul restului modulo A poate fi simplificat în mare măsură dacă - admitând un număr întreg X de forma $X = (X_{n-1}, X_{n-2}, \ldots, X_i, \ldots, X_0)$, unde X_i constituie un byte de b biţi, pentru i = 0,1,..., n-1 - valoarea modulului A se alege egală cu 2^b -1, unde b este un număr întreg mai mare decât 1. Se obţine astfel un aşa numit cod de cost redus (low-cost-code) care necesită, în scopul codificării numerelor, un sumator binar de b biţi cu transport end around (end around carry), întrucât evaluarea restului se bazează pe relaţia:

$$R = X \bmod (2^b - 1) = \left(\sum_{i=0}^{n-1} Xi \right) \bmod (2^b - 1) \qquad (4.21)$$

De exemplu, pentru X = 101111001 (X_{10} = 377) se obține, la o alegere A = 111 (A_{10} = 7), restul R = 110 (R_{10} = 6), valoare care, în baza relației (4.21) poate fi obținută și altfel întrucât $A = 7 = 2^b - 1$, deci b = 3, ceea ce implică adunarea tripleților 101 (5), 111 (7), 001 (1) conducând la suma 1101 (13) care evaluată modulo 7 prin însumare cu transport end-around, dă aceiași valoare R = 110 (6). Aceste coduri de cost redus, cu capacitatea de detecție ușor admisibilă, a tuturor erorilor singulare, își găsesc aplicabilitatea nu numai în sisteme de calcul cu toleranță la defectare, ci și în unele comerciale [Prad-86, RaFu-89]. Totuși și ele prezintă anomalii - spre exemplu, pentru X = 100010001 (X_{10} = 273) rezultă, adoptând A = 7, pe de o parte, R = 000 (R_{10} = 0), dar, pe de altă parte, prin însumarea cu end-around carry a tripleților 100 (4), 010 (2) și 001 (1), rezultă R = 111 (R_{10} = 7) = 000 -, a căror evitare implică investiții.

Aspectele critice în legătură cu supravegherea bazată pe coduri aritmetice - fără a pretinde o analiză profundă a acestei problematici - a permis justificarea preocupărilor pentru o cale alternativă de soluționare prin coduri bazate pe paritate [Dora-88,BhSr-89,Yarn-90].Motivația dezvoltării în cele ce urmează a acestei soluții este dublă una cu caracter general prin încercarea de uniformizare a modului de supraveghere la nivelul întregului sistem de calcul, fiind cunoscută larga răspândire a verificării prin paritatea operațiilor de transfer a informației cu influențe benefice în ceea ce privește software-ul sistemului, și una cu caracter particular prin încercarea de conectare a problematicii supravegherii operațiilor aritmetice la preocupările tezei.

Să ne referim introductiv la versiunea de verificare ordinară prin paritate predictivă a operației de adunare binară a operanzilor de intrare $A = (a_{n-1}, a_{n-2},..., a_i,..., a_1, a_0)$ și $B = (b_{n-1}, b_{n-2},..., b_i,..., b_1, b_0)$, fiecare având atașate câte un bit de paritate, p_a și p_0 , prin intermediul unui sumater paralel cu propagarea serială a transportului (simpple carry adder). La acesta, suma $S = (s_{n-1}, s_{n-2},..., s_i,..., s_1, s_0)$ este obținută bit-cu-bit, anume ranguri curente ale sumei (si) și ale transportului (ci) , și acesta pentru $0 \le i \le n-1$, fiind obținuți din biți curenți de intrare, a_i și b_i, și din transportul c_{i-1}, din rangul anterior, considerând $c_{.1} = c_{in}$, pe baza următoarelor ecuații

$$c_i = a_i \oplus b_i \oplus c_{i-2} \qquad (4.22)$$

$$c_i = a_i \,.\, b_i \oplus b_i c_{i-1} \oplus c_{i-1}\, a_i \qquad (4.23)$$

în care $\oplus$ semnifică operatorul de SAU EXCLUSIV, iar + pe cel de SAU.

Modalitatea de implementare a verificării este reprezentată în fig. 4.53, din care se poate observa că, pe de o parte, este calculată, printr-o schemă logică combinațională sintetizată cu circuite SAU EXCLUSIV, paritatea predicativă $p_A \oplus p_B \oplus p_C$, unde p_c este dată de ecuația:

$$pc_i = c_{n-2} \oplus c_{n-1} \oplus ..... \oplus c_i \quad \oplus ... \oplus c_0 \oplus c_{in} \qquad (4.24)$$

Pe de altă parte, din biții sumei se calculează paritatea asociată acesteia printr-o altă schemă combinațională care implementează ecuația

$$p_s = s_{n-1} \oplus s_{n-2} \oplus ..... \oplus s_i \,... \oplus s_1 \oplus s_0 \qquad (4.25)$$

uzitând în (4.25) de (4.22) și (4.24), precum și apelând la proprietăți ale operării SAU EXCLUSIV, obținem:

$$p_s = (a_{n-1} \oplus b_{n-1} \oplus c_{n-2}) \oplus \ldots\ldots \oplus (a_i \oplus b_i \oplus c_{i-2}) \oplus \ldots\ldots \oplus (a_0 \oplus b_0 \oplus c_{in})$$
$$= (a_{n-1} \oplus \ldots \oplus a_i \oplus \ldots\ldots \oplus a_0) \oplus (b_{n-1} \oplus \ldots\ldots \oplus b_i \oplus \ldots\ldots \oplus b_0) \oplus (c_{n-2} \oplus \ldots\ldots \oplus c_{i-1} \oplus \ldots\ldots \oplus c_{in})$$
$$= p_a \oplus p_b \oplus p_c \qquad (4.26)$$

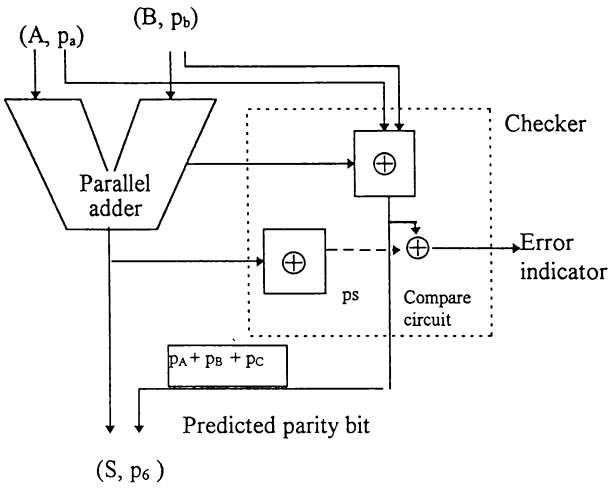


Fig. 4.53

În fond, relația (4.26) stă la baza verificării în sensul că paritatea predictivă este comparată cu cea calculată (fig. 4.53), iar necoincidența celor două determină indicarea unei erori. Din păcate, verificarea descrisă prezintă o problemă legată de defectele care afectează lanțul de circuite care intervin în propagarea transportului şi care pot provoca eronarea unui număr de biți de sumă, care adăugat la numărul biților de transport afectați să dea un număr par de biți eronați, făcând imposibilă detecția prin (4.26) a unei astfel de malfuncționări. Pentru claritate, să considerăm adunarea numerelor A şi B din fig. 4.54, în care prin C şi S sunt notați vectorii binari corespunzători biților de transport şi sumă la funcționarea corectă, iar prin C* şi S* sunt notați vectorii binari corespunzători biților de transport şi sumă prezentei în sumator a defectului de blocare a 0 a circuitul care determină lanțul de erori începând cu cea marcată.

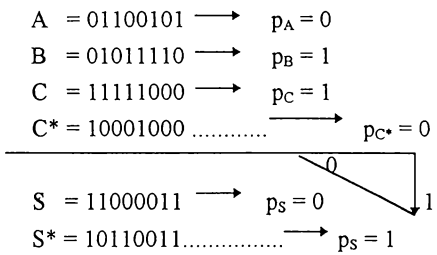


Fig. 4.54

Se poate remarca faptul că defectul singular menționat determină apariția câte unui pachet de erori de lungime 3 respectând vectorii C* şi S*, ceea ce conduce la mascarea prezenței defectului întrucât atât paritatea predicativă, cât şi cea calculată, dă aceeaşi valoare 1.

Pentru ieşirea din impasul relevat sunt practicate două soluții, una reprezentând-o duplicarea circuitelor de generare a transportului în conjuncție cu verificarea de paritate în maniera arătată în fig. 4.55

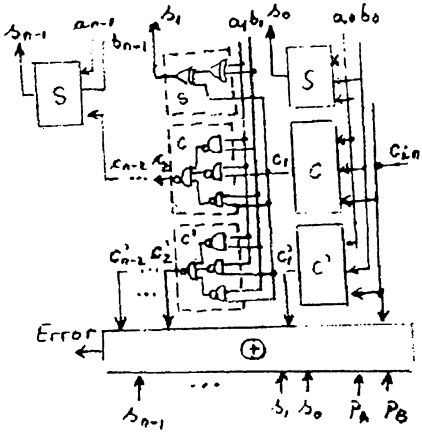


Fig. 4.55

A doua soluție se bazează pe faptul că un defect pe lanțul de circuite de propagare a transportului, după cum am văzut, provoacă pachete de erori cu un număr total al biților afectați - de transport împreună cu cei de sumă - care este par, şi, pentru a pune în evidență prezența defectului prin verificare de paritate, se doreşte ca acest număr să fie impar. Deci în mod deliberat se mai afectează prin eroare încă un bit. Cu alte cuvinte dacă defectul cauzează afectarea biților c_i , c_{i+1} ,c_{i+2} ,..., c_{i+t-1}, c_{i+t} , care începe cu c_i , se urmăreşte eroarea primară din lanț să afecteze şi bitul c_i , rezultând, în caz de defect, un număr total impar de biți afectați. Se obține în acest mod un aşa numit sumator cu suma dependentă de transport (carry-dependent sum adder) [Ra Fu - 89], a cărui reprezentare este dată în fig. 4.56 utilizând celule sumator complet a căror sinteză este descrisă prin ecuațiile booleene (4.21) şi (4.22).

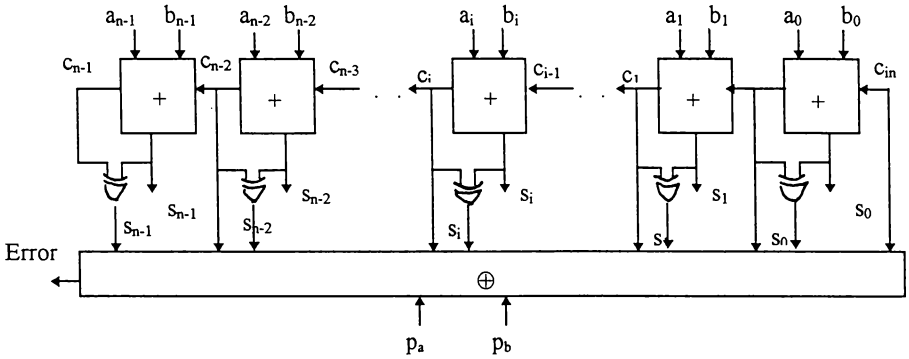


Fig. 4.56

În conformitate cu dezideratul menționat, ca intrări în arborele de circuite SAU EXCLUSIV generator al semnalului de eroare nu mai sunt aplicați biții sumă c_i, ca în fig. 4.53, ci valori binare furnizate, pentru i = 0, 1, ..., n-1, prin ecuații logice de forma:

$$s_i = s_i \oplus c_i \qquad (4.27)$$

În acest context menționăm că în [Ra Fu -89], atât ecuația booleană de generare a biților sumă si (pag. 416), cât și sinteza sumatorului din fig. 8.26 (pag. 417) sunt greșite.

Toate principiile de verificare prezentate pentru sumatoarele cu transport serial pot fi adoptate la variantele de accelerare a procesului de însumare de tipul cu transport anticipat (carry look-ahead adder), cu omiterea trasnportului (carry-skip adder), cu selecția transportului (carry-select adder) sau cu salvarea transportului (carry-save adder) [PaHe - 90]. De asemenea, problematica poate fi extinsă de la versiunea binară expresă și la sumatoare BCD (binary-coded-decimal), cu corespunzătoare creștere a investițiilor în circuistică redundantă. Prezentându-se sub acest aspect verificarea prin paritate a unui sumator cu anticiparea transportului pentru cazul cu duplicarea circuitelor de generare a transportului în conjuncție cu controlul de paritate, corespondent celui prezentat în fig. 4.55, în fig. 4.57 este expusă sinteza la care am admis că blocurile S sunt sintetizate pe seama ecuației (4.21), iar blocurile C pe seama ecuației (4.22), așa cum rezultă și din fig. 4.56. Schema de generare anticipată a transportului (carry lookahead circuit) are la baza sintezei cunoscutele ecuații booleene [Pa He - 90]:

$$g_i = a_i \cdot b_i \qquad (4.28)$$

$$p_i = a_i + b_i \qquad (4.29)$$

$$c_i = g_i + p_i\, c_{i-1} \qquad (4.30)$$

unde g_i reprezintă așa numita variabilă de generare, p_i reprezintă așa numita variabilă de propagare, iar i = 0,1,...,n-1.

Desigur, apar aspecte particulare dependent de partiționarea favorabilă prin prisma performantă - cost a lanțului de circuite care asigură implementarea paralelă a ecuației (4.30) în baza:

$$c_{n-2} = g_{n-2} + p_{n-2}\, c_{n-3} = g_{n-2} + p_{n-2}\,(g_{n-3} + p_{n-3} \cdot c_{n-4}) = \ldots = g_{n-2} + p_{n-2}\, g_{n-3} + \ldots \qquad (4.31)$$

Ecuația (4.31) pune în relief faptul că fiecare bit de transport nu depinde de anteriorul bit de transport, ci de variabilele de generare și propagare, influențând în mod specific formarea pachetelor de erori.

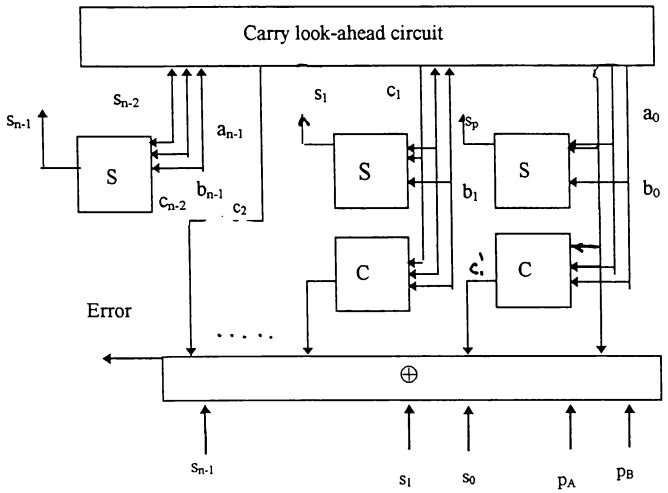


Fig. 4.57

În cele ce urmează se supun analizei unele coduri evoluate bazate pe paritate având, cu precădere, operațiile aritmetice de adunare și înmulțire binară, dar și pe cele logice de ȘI, SAU sau SAU EXCLUSIV.

4.3.1.1. Detecția erorilor la adunarea binară prin coduri sumă de control

Codurile sumă de control (checksum codes) pot fi utilizate pentru a detecta erori singulare de byte. Un cod sumă de control este un set de vectori alcătuiți din (n+1) simboluri din setul Z_q al numerelor întregi modulo q. Fiecare vector are o componentă denumită simbol de control care este egală cu suma modulo q a celorlalte componente ale vectorului, denumite simboluri de control [LiSh - 89, Ra Fu - 89, Davi - 90, Wa Pe - 90].

Definiție 43: Un cod sumă de control este un set

$$\left\{(x_c, x_{n-1}, \ldots, x_0) \mid (x_i, x_c \in Z_q) \text{ și } \left(x_c = \sum_{0 \le i \le n-1} x_i \bmod q\right)\right\}$$

În particular, prezintă interes codurile pentru care $q = 2^b$ pentru b reprezentând un număr întreg mai mare decât 1. Dacă b = 1, se obține cunoscutul cod de paritate pară.

La aceste coduri, fiecare simbol aparținând la Z_2^b poate fi codificat ca un byte de b biți. Prin urmare, fiecare cuvânt de cod are n.b biți de informație și b biți de control, conform cu reprezentarea din fig. 4.58

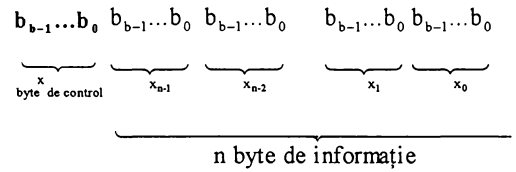


Fig. 4.58

Să considerăm adunarea modulo 2^b a vectorilor cu un număr oarecare, admitem (n+1) aparținând la Z_2^b.

Să admitem că doi vectori reprezentați în codul sumă de control au forma:

$$A = (a_c, a_{n-1}, a_{n-2}, \ldots, a_i, \ldots, a_o)$$
$$B = (b_c, b_{n-1}, b_{n-2}, \ldots, b_i, \ldots, b_0)$$

în care a_c și b_c sunt simboluri de control, iar $(a_c, a_{n-1}, a_{n-2}, \ldots, a_i, \ldots, a_o)$ și $(b_c, b_{n-1}, b_{n-2}, \ldots, b_i, \ldots, b_0)$ reprezintă părțile de informație utilă, constituită din vectorii:

$$A_d = (a_{n-1}, a_{n-2}, \ldots, a_i, \ldots, a_o)$$
$$B_d = (b_{n-1}, b_{n-2}, \ldots, b_i, \ldots, b_0)$$

Părților de informație utilă A_d și B_d le corespund numerele întregi $[A_d]$ și $[B_d]$ date de relațiile:

$$[A_d] = \sum_{0 \le i \le n-1} a_i \cdot 2^{bi} \quad \text{și} \quad [B_d] = \sum_{0 \le i \le n-1} b_i \cdot 2^{bi}$$

Adunarea ordinară a părților de informație utilă (A_d și B_d) corespunzătoare cuvintelor de cod este definită astfel:

$$S_d = A_d + B_d + C, \text{ unde}$$
$$C = (c_{n-2}, c_{n-3}, \ldots, c_1, c_0, c_{i-1}) \text{ cu}$$
$$c_i = \begin{cases} 0 \text{ dacă } a_i + b_i + c_{i-1} < 2^b \\ 1 \text{ dacă } a_i + b_i + c_{i-1} \ge 2^b \end{cases} \text{ pentru}$$
$$0 \le i \le n-2 \text{ și } c-1 = c_{in}$$

Vectorul sumă S_d îi corespunde numărul întreg $[S_d]$ dat de relația

$$[S_d] = ([A_d] + [B_d]) \bmod M$$

unde $M = 2^{bn}$ şi $c_{in} = 0$ pentru adunarea în complement de 2, respectiv $M = 2^{bn} - 1$ şi $c_{in} = c_{n-1}$ pentru adunarea în complement de 1.

Pe de o parte, simbolul de control al sumei celor două părţi de informaţie utilă este obţinut prin însumarea simbolurilor sumă obţinute, adică

$$s_c = \left(\sum_{0 \le i \le n-1} s_i \right) \bmod 2^b$$

Pe de altă parte, simbolul s_c poate fi anticipat (predicted) prin utilizarea simbolurilor de control corespunzătoare vectorilor intrare, şi c_c constituind simbolul de control corespunzător trasportului, astfel:

$$s_c = (a_c + b_c + c_c) \bmod 2^b, \text{ unde}$$

$$c_c = \left(\sum_{0 \le i \le n-1} c_{i-1} \right) \bmod 2^b, \text{ iar } c - 1 = c_{in}$$

Să considerăm, pentru exemplificare, că n = 4, b = 3 şi că A = (7,2,2,4,7), iar B = (5,3,3,4,3). Admiţând că însumarea se efectuează în complement de 2, rezultă $M = 2^{3.4} = 2^{12}$ şi C = (0,1,1,0), iar S_d = (2,2,4,7) + (3,3,4,3) + (0,1,1,0) = (5,6,1,2), de unde aveam, pe de o parte, $s_c = (s_3 + s_2 + s_1 + s_0) \bmod 2^3 = 6$. Pe de altă parte, avem $a_c = 7$, $b_c = 5$, $c_c = 2$, deci se poate anticipa $s_c = (a_c + b_c + c_c) \bmod 2^3 = 6$.

Ca şi la însumarea cu control de paritate, simbolurile de control, calculat şi anticipat, sunt comparate iar rezultatul este interpretat, în caz de inegalitate, ca eroare (fig. 4.59). Această schemă de control (checker) aparţine clasei de predicţie (prediction checkers) şi suferă de aceeaşi problemă relevată la însumarea cu control de paritate şi constând în faptul că defectele care afectează partea de schemă referitoare la generarea transportului nu pot fi detectate pentru că ele produc erori care se compensează.

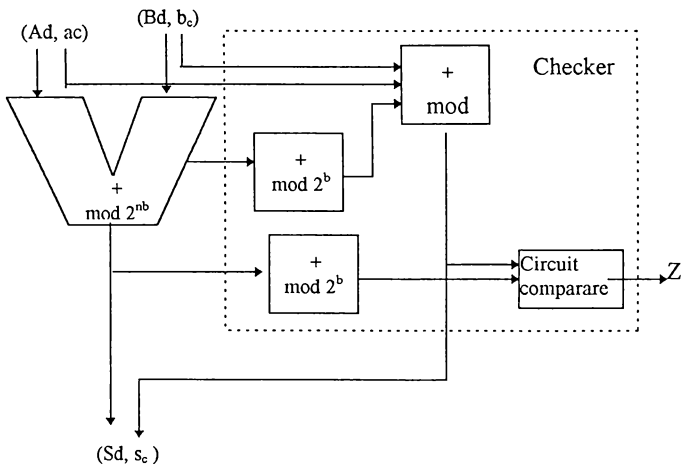


Fig. 4.59

Astfel exprimat, simultan sunt eronate s_c şi c_c şi cu toate că suma nu este corectă nu apare indicaţia de eroare. Această problemă poate fi surmontată prin duplicarea logicii de generare a

transportului (duplicated carry logic), strategice care poate fi aplicată eficient la însumarea byte-sliced, conform cu cele prezentate în fig. 4.60.

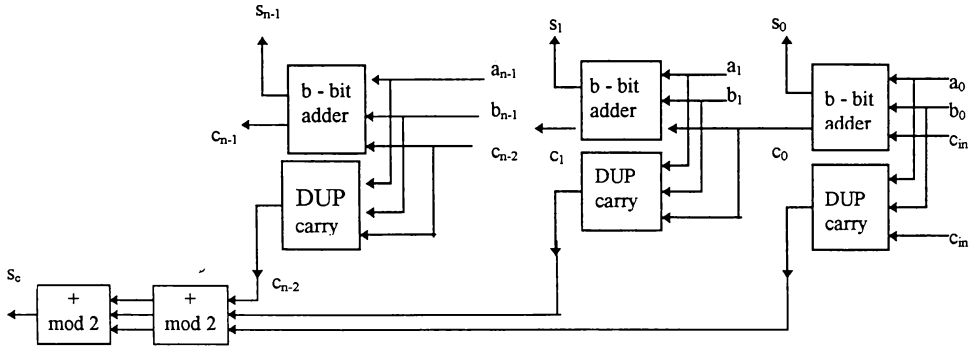


Fig. 4.60

4.3.1.2. Detecția erorilor la adunarea binară prin coduri combinate

Codurile bazate pe paritate s-au dovedit foarte eficiente şi favorabile prin prisma costului pentru memorii şi operații de transfer de date. Codurile cu resturi, pe de altă parte, apar foarte atractive pentru controlul operațiilor aritmetice, cum ar fi ADD, MULTIPLY şi altele.

A fost propusă combinarea codurilor de paritate cu cele reziduale în aşa numite coduri combinate (combination codes) care să permită eficiența corecției de erori pentru toate operațiile unei unități aritmetice şi logice. Vom urmări relevarea obținerii unui cod combinat şi vom supune analizei mecanismul său de corecție al erorii.

Un cod combinat este un cod format din blocuri de lungime n, în care informația este urmată de un grup de biți de paritate şi un rest de control, după cum se prezintă în fig. 4.61.

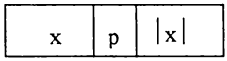


Fig. 4.61

Pentru informația X cuvântul de cod corespunzător este [X, P, |X|m], unde P este un grup de biți de paritate calculați pentru bytes lui X, iar |X|m exprimă restul lui X modulo m ($=2^b$ -1). codul şi mecanismul său de corecție va fi descris prin intermediul unui exemplu. Fie informația X reprezentată de cel mult 7 bytes (B_i, i = 0,...,6), fiecare format din 3 biți, conform cu:

$$X = \left(\underbrace{X_{20}X_{19}X_{18}}_{B_6} \underbrace{X_{17}X_{16}X_{15}}_{B_5} \underbrace{X_{14}X_{13}X_{12}}_{B_4} \underbrace{X_{11}X_{10}X_9}_{B_3} \underbrace{X_8X_7X_6}_{B_2} \underbrace{X_5X_4X_3}_{B_1} \underbrace{X_2X_1X_0}_{B_0} \right)$$

a) Porțiunea P a biților de paritate este alcătuită din 3 biți (P_2, P_1, P_0) prin alegerea de așa manieră a biților pentru generarea parităților înc't bytes să corespundă ecuațiilor de control dintr-o matrice de control asociată unui cod Hamming corector al erorii singulare.

$$H=\begin{pmatrix}1&0&1&0&1&0&1\\0&1&1&0&0&1&1\\0&0&0&1&1&1&1\end{pmatrix}\Rightarrow\begin{matrix}p_0=c_0\oplus c_2\oplus c_4\oplus c_6\\p_1=c_1\oplus c_2\oplus c_5\oplus c_6\\p_2=c_3\oplus c_4\oplus c_5\oplus c_6\end{matrix}$$
$$c_0\ c_1\ c_2\ c_3\ c_4\ c_5\ c_6$$

Vom avea deci pentru cei 3 biți de paritate ai codului:

$$P_0=B_0\oplus B_2\oplus B_4\oplus B_6=(x_0\oplus x_1\oplus x_2)\oplus(x_6\oplus x_7\oplus x_8)\oplus(x_{12}\oplus x_{13}\oplus x_{14})\oplus(x_{18}\oplus x_{19}\oplus x_{20})$$

$$P_1=B_1\oplus B_2\oplus B_5\oplus B_6=(x_3\oplus x_4\oplus x_5)\oplus(x_6\oplus x_7\oplus x_8)\oplus(x_{15}\oplus x_{16}\oplus x_{17})\oplus(x_{18}\oplus x_{19}\oplus x_{20})$$

$$P_2=B_3\oplus B_4\oplus B_5\oplus B_6=(x_9\oplus x_{10}\oplus x_{11})\oplus(x_{12}\oplus x_{13}\oplus x_{14})\oplus(x_{15}\oplus x_{16}\oplus x_{17})\oplus(x_{18}\oplus x_{19}\oplus x$$

b) Restul de control $|X|_7$ este obținut prin adunarea tuturor bytes de informație B_0, B_1,..., B_6,utilizând sumatoare modulo 7.

Astfel având X dat de următorul vector binar:

X = 10110111001101101011 obținem P_0 = 1, P_1 = 0 și P_2=0.

Pe de altă parte $|X|_7$ = (5+5+6+3+3+2+7 = 3, deci cuvântul codificat va fi:

$$\left[X,P,|x|_7\right]=\underbrace{10110111001101101011}_{X}\underbrace{001}_{P}\underbrace{011}_{[x]_7}$$

În continuare, să analizăm capacitățile de detecție și corecție ale codului combinat astfel generat. Admitem că în cuvântul eronat X bitul 46 care are valoarea corectă 0 devine x_{16} = 1 (tipul de eroare 0 →1, respectiv eroare de 0). Atunci prin evaluarea ecuațiilor de control a parității referitoare la cuvântul eronat X se obține sindromul de participare S_p = 110 indicând o eroare în byte-ul B_5 a lui X. Într-adevăr, la exemplul considerat, avem P_0 = 1, P_1 = 1 și P_2 = 1., deci

$$S_p=(P_2P_1P_0)\oplus(P_2^*P_1^*P_0^*)=(001)\oplus(111)=110$$

Pentru a obține poziția bitului eronat, se calculează sindromul rezidual: $S_r=|X-X|=\|X\|_7-|X|_7|_7$. Într-adevăr prin eroarea 0 → 1 a poziției binare x_{16} rezultă o mărime a erorii de 2^{16} care modulo 7 dă $/2^{16}/_7=2$ indicând asupra celui de al doilea bit din cadrul byte-ului eronat. La exemplul considerat, avem $/X'/_7$ = (5+6+3+3+2+7) $S_r=\|X\|_7-|X|_7|_7=|5-3|_7=2$.

Pe de altă parte, dacă eroarea este de tipul 1→0 (eroare de 1), în cadrul aceluiaşi byte detectat ca eronat prin aceleaşi ecuaţii de control a parităţii, dar vizând poziţia binară $X_7 = 1$, deci $X_{17} = 0$, atunci se obţine mărimea erorii $|-2^{17}$ care modulo 7 dă valoarea 3 indicând că eroarea a afectat al 3-lea bit din cadrul byte-ului eronat. Revenind la exemplul numeric considerat pentru cazul $X_{17} = 1$, obţinem, mai întâi

$$\left\|X'\right|_7 = |6-3|_7 = 3+6+3+3+2+7 \bmod 7 = 6 \text{ şi , apoi,}$$

$$S_r = \left\|X''\right|_7 - \left|X\right|_7\Big|_7 = |6-3|_7 = 3$$

Se impune remarcat că există un sindrom de paritate unic pentru fiecare dintre cei 7 bytes şi un sindrom rezidual unic pentru fiecare bit şi tip de eroare din cadrul byte-ului eronat. De asemenea, trebuie subliniat că atunci când eroarea afectează partea de informaţie utilă, atât biţii de control de paritate cât şi cei de control rezidual detectează eroarea şi împreună permit corecţia acesteia. Pe de altă parte, dacă eroarea vizează biţii de paritate P sau cei de control rezidual $|x|_7$, doar unul dintre sindromuri este diferit de 0. În consecinţă, este valabilă următoarea strategie de localizare a erorii, respectiv de corelaţie a acesteia:

$$\begin{aligned} &S_p = 0,\ S_r = 0 \Rightarrow \text{ nu există eroare;} \\ &S_p = 0,\ S_r \neq 0 \Rightarrow \text{ eroare la } |x|_7; \\ &S_p \neq 0,\ S_r = 0 \Rightarrow \text{ eroare la P;} \\ &S_p \neq 0,\ S_r \neq 0 \Rightarrow \text{ eroare la X.} \end{aligned} \qquad (4.32)$$

Dacă $S_p = i \neq 0$ şi $S_r = j \neq 0$, atunci este eronat $B_{i-1}\,(i = 1, 2, \ldots, 7)$, şi dacă:

$j = 1 \Rightarrow$ eroare la primul bit, de tipul $0 \rightarrow 1$

$j = 6 \Rightarrow$ eroare la primul bit, de tipul $1 \rightarrow 0$

$j = 2 \Rightarrow$ eroare la al doilea bit, de tipul $0 \rightarrow 1$

$j = 5 \Rightarrow$ eroare la al doilea bit, de tipul $1 \rightarrow 0$

$j = 4 \Rightarrow$ eroare la al treilea bit, de tipul $0 \rightarrow 1$

$j = 3 \Rightarrow$ eroare la al treilea bit, de tipul $1 \rightarrow 0$

Prin urmare, orice eroare de bit singulat în X poate fi detectată şi corectată.

Din punct de vedere al implementării, în [Ra Fu- 89]codul combinat prezentat este supus unei modificări pentru a-l face mai atractiv. Aceasta constă în substituirea byte-ului de paritate P prin biţi de paritate, câte unul pentru fiecare din cei k bytes de informaţie corespunzători lui X. Este de remarcat că această modificare nu alterează proprietăţile de detecţie/corecţie ale codului. Totuşi, prin această modificare creşte redundanţa codului (k biţi de paritate în locul celor $[\log_2 k]$) dar implementarea este mai directă. Codul combinat modificat este ilustrat în fig. 4.62.

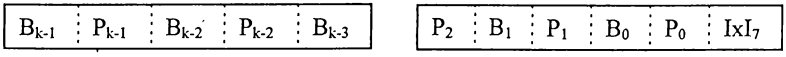


Fig. 4.62

În cele ce urmează, vom analiza implementarea controlului cu acest cod combinat modificat. Pentru un operand X de n biți (n = k-l), avem asociat cuvântul codificat [X, P_x, |X|m], unde P_x este alcătuit din k biți de paritate (câte un bit de paritate pentru fiecare byte a lui X) și |X|m este restul lui X modulo m (= 2^l-1).

Să considerăm mai întâi un modul simplu pentru o unitate aritmetică și logică (ALU), reprezentat ca în fig. 4.63. A reprezintă un operand intern,Bun operand de intrare (extern), iar ∅ reprezintă operația de executat, toate specificate la momentul de timp t. De asemenea, ieșirea R constituie rezultatul operației ∅ asupra lui A și B, care este disponibil la o unitate de timp mai târziu, adică avem R (t+1) = ∅ (A(t), B (t)). Pentru a aplica codul combinat modificat la acest model, în fig. 4.64 se observă modificările executate, constând în esență în suplimentarea structurii printr-o unitate reziduală (residue unit) și printr-o unitate de decodificare (decoder unit). Operația ∅ prevăzută în fig 4.63 substituită prin operația combinată ($\varnothing, \varnothing_m$).

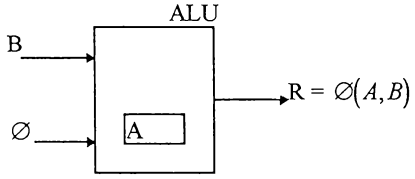


Fig. 4.63

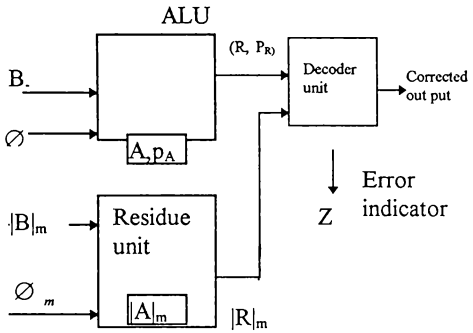


Fig. 4.64

Pentru fiecare operație ∅ asupra lui A și B, este prevăzută logica de evaluare a parității astfel încât la ieșire se prezintă (R, P_R), unde P_R constă din k biți de paritate denumiți biți de paritate predicativi (predicted parity bits). De asemenea pentru fiecare ∅, este prevăzută operația paralelă ∅m corespunzătoare resturilor $|A|_m$ și $|B|_m$, astfel încât rezultatul corespunzător va fi:
$|R|_m = \varnothing_m(|A_m|,|B_m|)$.

În baza celor prevăzute se generează cuvinte de cod combinat [R, P_R $|R|_m$] în cazul că nu există eroare. Datorită defectelor logice, rezultatul combinat prezentat poate fi afectat de erori, situație în care unitatea de decodificare generează sindromurile (S_p, S_r) și le utilează pentru a localiza și corecta aceste erori.

4.3.1.3 Detecția erorilor și corecția erorii singulare la adunarea binară și la operațiile logice prin coduri bazate pe paritate

În general, codurile bazate pe paritate nu acoperă controlul operațiilor logice, cu excepția celor de SAU EXCLUSIV și NU - SAU EXCLUSIV. Vom prezenta o tehnică [Ra Fu - 89] care se

bazează pe ideea că rezultatul unei operații arbitrare $\varnothing$ poate fi transformat linear într-unul al unei operații de SAU EXCLUSIV. Aceasta va permite predicția de paritate pentru operația $\varnothing$.

Sunt considerate două cuvinte de date ca întrebări, după cum urmează:
$A = (a_{k-1}, a_{k-2}, \ldots, a_i, \ldots, a_1, a_0)$ și $B = (b_{k-1}, b_{k-2}, \ldots, b_i, \ldots b_1, b_0)$. Biții de paritate corespunzători sunt generați prin relațiile:, unde $\sum^{\oplus}$ semnifică suma modulo 2. După cum rezultă din fig. 4.65 modulul pentru unitatea aritmetică și logică (ALU) prezintă două intrări [A, p_A] și [B, p_B] și o singură ieșire [Y, p_Y], cea din urmă definită prin vectorul $Y = (y_{k-1}, y_{k-2}, \ldots, y_i, \ldots, y_1, y_0)$ și bitul de paritate dat de relația:

$$p_y = \sum_{i=0}^{k-1}{}^{\oplus} y_i$$

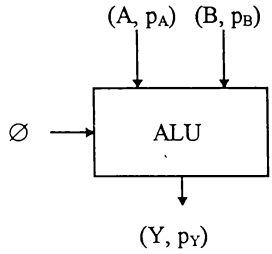


Fig. 4.65

Pentru operația SAU EXCLUSIV, controlul de paritate respectiv generarea sindromului S, are loc astfel:

$$S = (\sum_{i=0} y_i) \oplus p_A \oplus p_B, \text{ cu } y_i = a_i \oplus b_i \text{ pentru } i = 0,1,\ldots,k-1 \text{ și}$$ cu concluzia că funcționarea este eronată când S = 1, respectiv că e normală, fără defecte, când S = 0, A ia funcție de transformare $f_{\varnothing}(y_i)$ exprimă transformarea a celui de al i-lea rezultat al operației ALU în al i-lea rezultat al operației SAU EXCLUSIV (respectiv $y_i = a_i \oplus b_i$, după cum rezultă din fig. 4.66.

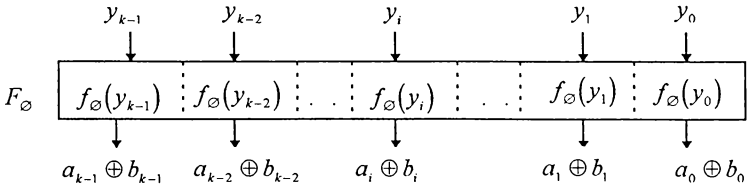


Fig. 4.66

Prin urmare, ieşirea funcţiei $f_{\varnothing}(y_i)$ este dată de: $f_{\varnothing}(y_i) = a_i \oplus b_i$, pentru i = 0,1,...,k - 1. Facă funcţia $f_{\varnothing}(y_i)$ satisface relaţiile menţionate în teorema ce urmează, atunci o eroare la yi se propagă la ieşirea $f_{\varnothing}(y_i)$ şi, deci paritatea pentru ieşirea ALU poate fi anticipată.

Definiţie 4.4: Diferenţa booleană a unei funcţii $F(x_0, x_1, ..., x_i, ..., x_{n-1})$ în raport cu variabila de intrare xi este dată de relaţia:

$$\frac{dF}{dx_i} = F(x_0, x_1, ..., x_i, ..., x_{n-1}) \oplus F(x_0, x_1, ..., \overline{x_i}, ..., x_{n-1})$$

Teorema 4.3: Dacă raportat la funcţia de transformare $f_{\varnothing}(y_i)$ este satisfăcută condiţia:

$$\frac{d(f_{\varnothing}(y_i))}{dy_i} = 1, \text{ pentru } i = 0,1,..., k - 1 \qquad (4.34)$$

$$f_{\varnothing}(y_i) = a_i \oplus b_i \qquad (4.35)$$

atunci paritatea p_y pentru ieşirea unei ALU poate fi anticipată după cum urmează:

a) pentru $r_i = y_i \oplus a_i \oplus b_i$, avem $p_y^{\cdot} = p_A \oplus p_B \left(\sum_{i=0}^{k-1} {}^{\oplus} r_i \right)$ (4.36)

b) pentru $r_i = y_i \oplus a_i \oplus b_i$, avem

$$p_y' = \begin{cases} p_A \oplus p_B \oplus \left(\sum_{i=0}^{k-1} {}^{\oplus} r_i \right), & \text{când k este par} \\ \overline{p_A \oplus p_B} \oplus \left(\sum_{i=0}^{k-1} {}^{\oplus} r_i \right), & \text{când k este impar} \end{cases} \qquad (4.37)$$

Pentru demonstrarea teoremei 4.3 este utilă următoarea temă.

Tema 4.17: Dacă F reprezintă o funcţie de două variabile independente, Y şi R, atunci soluţiile care satisfac ecuaţia $\frac{dF}{dY} = 1$ sunt $F = Y \oplus R$ şi $\overline{Y \oplus R}$

Demonstraţia temei 4.17: Admitem că F = Y * R, unde * semnifică o operaţie oarecare. Luând în consideraţie definiţia diferenţei booleene rezultă:

$$\frac{dF}{dY} = (1 * R) \oplus (0 * R) = 1 \qquad (4.38)$$

Supunem analizei soluţiile ecuaţiei (4.38) apelând la tabelul din fig. 4.66.

baz	1 * R	0 * R	Soluţii
1	1	0	$\begin{cases} R = 1 \text{ şi } * = \text{AND} \\ R = 0 \text{ şi } * = \text{OR} \end{cases}$
2	0	1	$\begin{cases} R = 1 \text{ şi } * = \text{NAND} \\ R = 0 \text{ şi } * = \text{NOR} \end{cases}$
3	R	$\overline{}$	* = EX – NOR
4	$\overline{}$	R	* = EX – OR

Pentru început facem observație că în tabelul de la pag. 4.27 din [Ra Fu -89] se impune corecția substituției soluției indicată pentru cazul 1 cu cea din tabelul alăturat. Întrucât R a fost admisă ipotetic ca variabilă, rezultă că, evident, doar cazurile 3 și 4 oferă soluții care satisfac ecuația (4.38). Prin urmare, soluțiile care satisfac ecuația $\frac{dF}{dY} = 1$ sunt $F = Y \oplus R$ și $F = \overline{Y \oplus R}$.

Este evident că $F = \overline{Y} * R$, $F = Y * \overline{R}$ și $F = \overline{Y} \oplus \overline{R}$ unde * = SAU EXCLUSIV (EX - OR) sau * = NU - SAU EXCLUSIV (EX - NOR), reprezintă, de asemenea, soluții care satisfac ecuația $\frac{dF}{dY} = 1$.

Demonstrația teoremei 4.3: Din schema 4.17, rezultă că următoarea relație satisface ecuația (4.34).

$$f_{\varnothing}(y_i) = y_i \oplus r_i \qquad (4.39)$$

Din ecuațiile (3.....) și (3 .....), r_i poate fi întotdeauna obținut prin :

$$r_i = y_i \oplus a_i \oplus b_i$$

Pe de altă parte, efectuând adunarea modulo 2 în raport cu i în ambii membri ai ecuațiilor (4.35) și (4.39) obținem:

$$\sum_{i=0}^{k-1}{}^{\oplus} f_{\varnothing}(y_i) = \left(\sum_{i=0}^{k-1}{}^{\oplus} a_i \oplus b_i\right) = p_A \oplus p_B$$

$$\sum_{i=0}^{k-1}{}^{\oplus} f_{\varnothing}(y_i) = \left(\sum_{i=0}^{k-1}{}^{\oplus} y_i \oplus r_i\right) = \left(\sum_{i=0}^{k-1}{}^{\oplus} y_i\right) \oplus \left(\sum_{i=0}^{k-1}{}^{\oplus} r_i\right)$$

Din aceste ecuații, luând în considerare proprietățile intrinseci ale operatorului EX - OR, se obține pentru paritatea anticipată py:

$$p'_y = \sum_{i=0}^{k-1}{}^{\oplus} y_i = p_A \oplus p_B \left(\sum_{i=0}^{k-1}{}^{\oplus} r_i\right), \text{ care satisface ecuația} \qquad (4.36)$$

Pe de altă parte, din tema 4.17, ecuația (4.35) este deasemenea, satisfăcută de relația:

$$f_{\varnothing}(y_i) = \overline{y_i \oplus r_i}$$

Procedând în manieră similară, pentru $r_i = y_i \oplus \overline{a_i \oplus b_i}$ rezultă paritatea anticipată p_y' dată de ecuația (4.37).

Din definiția ecuației booleene, ecuația (4.34) arată în mod clar că o eroare singulară y este propagată la ieșirea funcției $f_{\varnothing}(y_i)$. Ecuațiile (4.34) și (4.35) sunt importante nu numai pentru faptul că permit evaluarea biților de paritate anticipată, ci și pentru că stabilesc condiția de propagare erorii din yi la ieșirea funcției $f_{\varnothing}(y_i)$. Admițând că R este dat de vectorul $R = (r_{k-1}, r_{k-2}, \ldots, r_i, \ldots, r_1, r_0)$, în tabelul din fig. 4.68 sunt date funcțiile ri pentru operațiile aritmetice și logice fundamentale. În acest tabel s-a notat prin c-1 bitul de transport care intră în poziția i.

Teorema 4.3 permite efectuarea controlului bazat pe paritate pentru o operație ALU oarecare prin utilizarea ecuației (4.36) de evaluarea a parității anticipate p_y după cum urmează.

operația ∅	y_i	r_i	
		* = EX – OR	* = EX – NOR
AND	$a_i \wedge b_i$	$a_i \vee b_i$	$\overline{a_i} \wedge \overline{b_i}$
OR	$a_i \vee b_i$	$a_i \wedge b_i$	$\overline{a_i} \vee \overline{b_i}$
EX - OR	$a_i \oplus b_i$	0	1
EX - NOR	$\overline{a_i \oplus b_i}$	1	0
ADD	$a_i + b_i$	c_{i-1}	$\overline{c_{i-1}}$

Fig. 4.68

Teorema 4.4: Pentru intrările A şi B, fiecare de câte k biţi, controlul bazat pe paritate, respectiv sindromul, pentru o operație ∅ a ALU devine:

$$S_{\varnothing} = p_v \oplus p'_y = \left(\sum_{i=0}^{k-1}{}^{\oplus} y_i\right) \oplus \left(\left(\sum_{i=0}^{k-1}{}^{\oplus} r_i\right) \oplus p_A \oplus p_B\right) = \left(\sum_{i=0}^{k-1}{}^{\oplus} f_{\varnothing}(y_i)\right) \oplus p_A \oplus p_B$$

unde $f_{\varnothing}(y_i) = y_i \oplus r_i = a_i \oplus b_i$

iar $S_{\varnothing} = 1$ pentru eroare detectată, respectiv $S_{\varnothing} = 0$ pentru funcționare normală, fără eroare.

Fig. 4.69 arată schematic controlul bazat pe paritate pentru o operație ∅ a ALU.

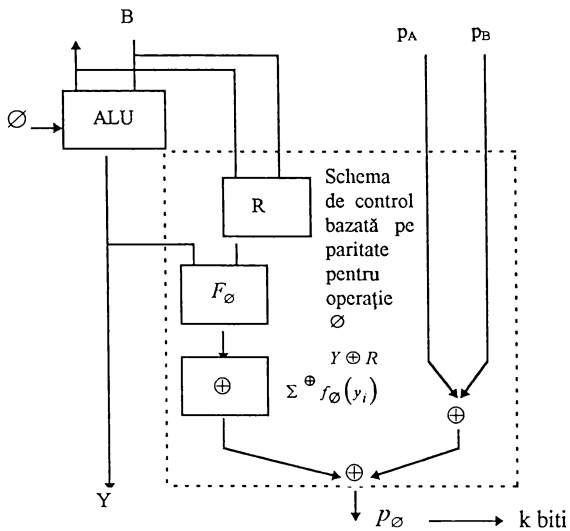


Fig. 4.69

Teorema 4.5: Schema de control bazată pe paritate din fig. 4.69 detectează toate erorile singulare din intrările A şi B, dacă nu există defecte nici în ALU, nici în schema de control.

Demonstraţie: Ecuaţia (4.40) poate fi supusă următoarei transformări evidente.

$$S_{\varnothing} = \left(\sum_{i=0}^{k-1}{}^{\oplus} f_{\varnothing}(y_i) \right) \oplus p_A \oplus p_B = \left(\sum_{i=0}^{k-1}{}^{\oplus} a_i \oplus b_i \right) \oplus p_A \oplus p_B \qquad (4.41)$$

Prin urmare, avem $\frac{dS_{\varnothing}}{da_i} = 1$ şi $\frac{dS_{\varnothing}}{db_i} = 1$.

Aceasta arată că o eroare singulară pe intrări poate fi întotdeauna detectată. În cele ce urmează să considerăm, bazat pe rezultate anterioare, corecţia erorilor care afectează operaţii ALU. Controlul bazat pe paritate în discuţie poate fi aplicat pentru a controla grupuri de biţi definite de rândurile matricii de control H. Pentru un cod (n, k) există n - k = r grupuri de control, fiecare dintre acestea constituind seturi de biţi incluşi într-un rând al următoarei matrici H.

$$H = [\, H_e \,,\, I_r \,]_{r \times n} ,$$

în care H_e reprezintă matricea de codificare (r x k) şi I_r reprezintă matricea identitate (r x v). Două cuvinte de cod intrate constituie vectorii de forma [A, C_A] şi [B, C_B], în care

$C_A = (c_{A,0}, c_{A,1}, \ldots, c_{A,r-1})$ şi $C_B = (c_{B,0}, c_{B,1}, \ldots, c_{B,r-1})$ sunt grupuri de biţi de control.

De asemenea, definim pe C astfel: $C = C_A \oplus C_B = (c_0, c_1, \ldots, c_{r-1})$ unde $c_i = c_{A,I} \oplus c_{B,I}$. Ieşirea ALU se prezintă în forma [Y, C_y], unde $C_Y = (c_{Y,0}, c_{Y,1}, \ldots, c_{Y,r-1})$ reprezintă grupul de biţi de control pentru ieşirea Y.

O procedură de corectare a erorii constă din trei paşi mai importanţi (1) generarea sindromului, (2) determinarea poziţiei erorii (decodificarea sindromului) şi (3) inversarea bitului eronat.

(1) Generarea sindromului pentru ieşirea Y implică soluţionarea următoarelor probleme.

a) Generarea biţilor de control: $C_y = Y \cdot H_e^{\,T}$

b) Anticiparea biţilor de control: $C_p = (c_{p,0}, c_{p,1}, \ldots, c_{p,r-1})$, iar $C_p = R \cdot H_e^{\,T} \oplus C$

c) Generarea sindromului:

$S = C_y \oplus C_p = Y \cdot H^T{}_e \oplus R \cdot H_e^{\,T} \oplus C \Rightarrow S = F_{\varnothing} \cdot H_e^{\,T} \oplus C$, unde $F_{\varnothing} = Y \oplus R$

(2) Determinarea poziţiei eronate se efectuează din sindrom, astfel:

W (S) = 0 - funcţionare corectă, fără eroare; W (S) = 1 - eroare în partea biţilor de control C_p şi W (S) ≥ 2 - eroare în ieşirea Y, unde W (S) semnifică ponderea sindromului. Aceasta este determinată cu precizie din matricea H, în special pentru cazul corecţiei de eroare a ieşirii Y.

(3) Indicatorul de eroare E = (E_y , E_c) specifică modelul de biţi corespunzător erorii care se impune corectată, unde E_y specifică eroarea ieşirii Y şi E_c specifică eroarea în partea biţilor de control. Ieşirea corectată $[Y, C_y]$ se obţine prin intermediul menţionatului indicator de eroare astfel: $Y = Y \oplus E_Y$ şi $C = C_p \oplus E_c$.

În figura 4.70 se prezintă o schemă de corecție a erorii (EC) pentru ALU. Este de remarcat că partea de schemă demarcată prin linie punctată EC0 inclus în EC este aceeaşi schemă de decodificare a erorii cunoscută de la controlul memoriilor de mare viteză [Ra Fu - 89].

Respectând metodica de sinteză prezentată, pot fi aplicate la controlul operaţiilor aritmetice şi logice şi alte coduri detectoare conectoare de erori la nivel de bit, respectiv la nivel de byte cunoscute de la controlul memoriilor [Sa AV - 87, Ch FP - 89, Da FC - 89, Sohi - 89, Da Fu - 90, Barb - 92, Akht - 94].

4.3.1.4 Detecția erorilor la înmulțirea binară prin cod de paritate

Preconizăm o analiză a investiției în circuistică şi a penalității de performanță pe care le implică controlul incorporat, de tip built-in, implementat prin atât de răspândita verificare a transportului de informație, care este paritatea, asupra operației de înmulțire binară. Fără a pierde din generalitate, abordăm problematica pentru numere întregi fără semn, extrapolarea la numere fracționare şi la reprezentările în semn-mărime, complement de 1 sau de 2 putând fi realizată prin aplicarea la procedurile clasice cum ar fi Robertson, Booth ş.a.[Haye-88,Rh St-86,ChCh-91,Gork-91, Parh-91, Sosn-4].

Pentru început ne vom referi la înmulțirea serială implicând structura dispozitivului din fig. 4.71, destinat înmulțirii binare pe 8 biți la care se adaugă unul de paritate.

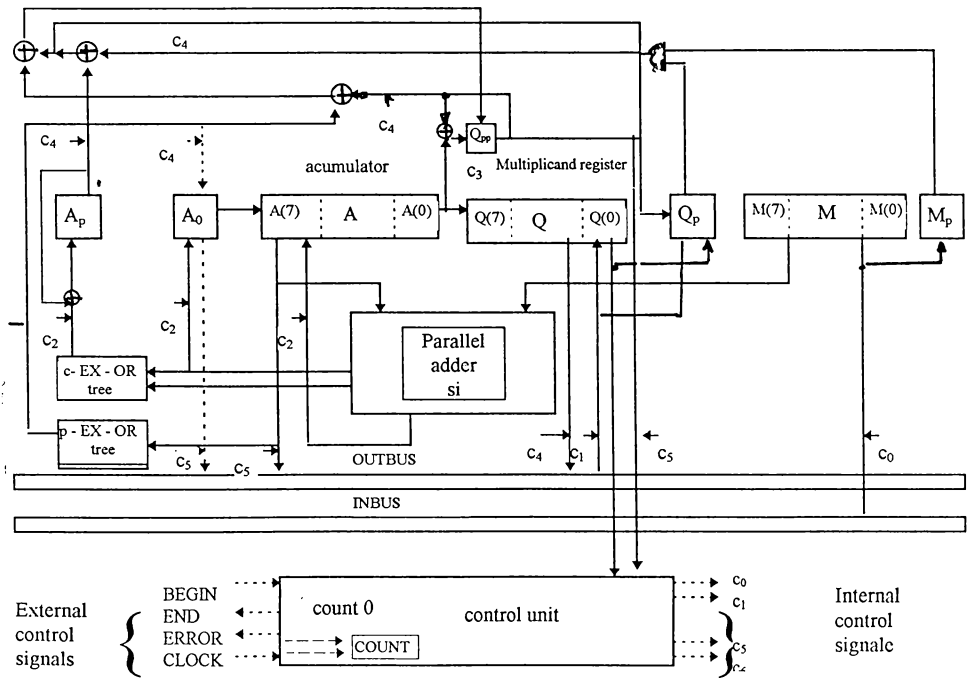


Fig. 4.71

Ca elemente esențiale se disting cele două registre M și Q - pentru deânmulțit (multiplicand) și înmulțitor (multiplier) care sunt prevăzuți cu biți de paritate marcați în mod distinct, fiind încărcați în magistrala intrare INBUS. Admitem că înmulțitorul este constituit de vectorul binar $A=\left(a_7,a_6,\ldots,a_1,a_0,a_p\right)$ cu mărimea $A_M=\sum_{i=0}^{7} a_i\cdot 2^i$ și bitul de paritate a_p dat de $a_p=\sum_{i=0}^{7} a_i$, unde $\sum$ semnifică operația de însumare modulo2. În mod similar, avem vectorul înmulțitor $B=\left(b_7,b_6,\ldots,b_1,b_0,b_p\right)$ cu mărimea $B_M=\sum_{i=0}^{7} b_i\cdot 2^i$ și bitul de paritate $b_p=\sum_{i=0}^{7} b_i$,

Procesul de înmulțire se execută în 8 pași de adunare - deplasare definiți astfel:

$$P_i \leftarrow P_i + a_i \times B \qquad (4.42)$$

$$P_{i+1} \leftarrow 2^{-1} P_i \qquad (4.43)$$

unde $P_0 = 0$ și $P_8 = P_M \leftarrow A_M \times B_M$, iar i progresează de la 0 la 7. Cantitățile $P_0, P_1, \ldots, P_7$ sunt denumite produse parțiale și ele sunt formate în mod secvențial în registrul pe 16 biți compus din registrele A și Q, unde registrul A poartă denumirea de acumulator (accumulator) și are conținutul inițial nul. Cele două registre sunt prevăzute cu capacități de deplasare la dreapta și ele vor stoca în final, produsul P_8 (a cărui lungime maximă este de 16 biți), fiind descărcate în magistrala ieșire OUTBUS. Când bitul curent în înmulțitor ai=1,atunci se efectuează adunarea dată de (2.1), adică $P_i \leftarrow P_i + B$, în care scop structura conține un sumator paralel (parallel adder) cu 8 celule de însumare și cu transport serial (ripple carry adder) sau anticipat (carry-lookakead adder) [Haye-88, Vasi-93]. Când bitul curent al înmulțitorului $a_i = 0$, atunci (2.1) devine $P_i \leftarrow P_i + 0$, prin urmare procesul de înmulțire se bazează pe adunarea la produsele parțiale curente ale deînmulțitului sau ale lui 0, fiecare pas de adunare fiind succedată de deplasarea la dreapta cu un bit, astfel încât produsul parțial este expandat de la 8 la 16 biți. Cei 8 pași necesari înmulțirii sunt contorizați prin numărătorul COUNT inclus în unitatea de control (control unit), care declanșează procesul de înmulțire când recepționează semnalul de control extern BEEIN și furnizează, prin intermediul trenului de tact CLOCK, semnale de control interne c_i (internal control singnals) care determină execuția microoperațiilor din altgoritmul de înmulțire, iar, în final, este emis semnalul de terminare END.

Ar mai fi de remarcat faptul că în cursul unui pas de adunare în vederea formării unui produs parțial este posibil să apară depășirea capacității registrului acumulator, ceea ce impune adăugarea la elementele de structură menționate a unui bistabil de indicare a depășirii la registrul acumulator, A_0 (acumulator overflow), comandat de linia count, de transport din celula cea mai semnificativă a sumatorului paralel. Cu aceste precizări, înlănțuirea microoperațiilor corespunzătoare algoritmului de înmulțire, fără a implica elemente de structură anexe reclamate de partea de verificare, poate fi urmărită, într-o primă variantă de deschidere, prin ordinograma reprezentată în fig. 4.72, cum arătam, cei 8 pași de adunare - deplasare sunt controlați de numărătorul COUNT, inițial adus la zero

şi sintetizat modulo 8, astfel încât atunci când este activată linia COUNT0, decodificare a noii stări de zero, procesul de înmulţire se termină.

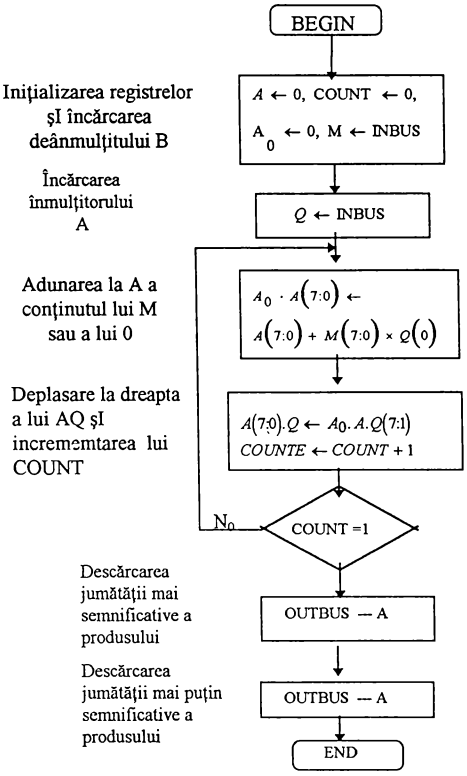


Fig. 4.72

Cu toate că o descriere prin ordinogramă, de tipul celei expuse în fig. 4.72, prezintă avantajul reliefării cu claritate a buclelor care intervin în algoritm, aderă la o altă versiune de prezentare a lanţului de microoperaţii şi anume printr-un limbaj formal caracterizată printr-o mai mare concizie şi flexibilitate. Este cunoscut [Haye-88, Pa He-90], o astfel de descriere este dependentă de nivelul la care este realizată, sub care aspect precizăm că efectuăm, aşa cum de altfel este uzual pentru un dispozitiv de înmulţire, o descriere la nivelul operaţiilor de transfer între registre prin intermediul unui limbaj formal [Haye- 88], ale cărui caracteristici esenţiale le sintetizăm în cele ce urmează.

a) Registrele sunt definite prin enunţuri declare care precizează numele acestora, dimensiunea lor şi ordinea în care sunt numerotaţi biţii. De exemplu, declare M (7 : 0) semnifică faptul că registrul M este compus din 8 bistabile, care pot fi identificate în mod individual prin M (i), unde i creşte de la 0 la 7 în sensul de la dreapta la stânga, ceea ce este echivalent cu a scrie M = (7). M (6). M (5). M (4). M (3). M (2). M (1). M (0).

b) Întrucât magistralele sunt utilizate, în general, în aceeaşi manieră cu registrele, ele se declară în mod similar.

c) Operaţiile de transfer între registre care pot fi executate simultan, pe durata aceleiaşi perioade de tact, sunt separate prin virgulă, iar acele seturi de operaţii care trebuie să se deruleze la perioade de tact succesive sunt separate cu punct şi virgulă. Astfel $A \leftarrow 0$, COUNT $\leftarrow 0$, M $\leftarrow$ INBUS; specifică faptul că cele 3 microoperaţii de ştergere a registrului A şi numărătorului COUNT, precum şi de încărcare a registrului M din magistrala INBUS, se pot desfăşura în paralel fără a provoca conflicte de date, resurse sau timp.

Trebuie subliniat că dacă un registru este compus din bistabile master-slave, atunci el poate fi citit şi înscris prin acelaşi enunţ, cum este cazul bistabilului Q (0) din enunţul $A(7) \leftarrow M(7) \oplus Q(0)$, $Q(7) \leftarrow 0$;

d) Ordinea în care apar în cadrul descrierii enunţurile separate prin punct şi virgulă corespunde secvenţei de microoperaţii impusă de algoritm.Devieri de la această secvenţă sunt posibile prin enunţul de salt necondiţionat go to , care implică utilizarea de etichete, sau prin construcţii if ... then, care condiţionează o anumită microoperaţie de o anume stare a dispozitivului. Astfel, enunţul if COUNT 0 = 1 then go to ADD, semnifică faptul că trebuie testată starea de zero în toate rangurile numărătorului COUNT şi, când aceasta este îndeplinită, se execută microoperaţiile specificate de către enunţul prevăzut cu eticheta ADD. În cazul neândeplinirii condiţiei, se execută microoperaţiile specificate de către enunţul succesiv celui analizat.

Se subliniază că astfel de teste se pot executa, în mod uzual, pe durata aceleiaşi perioade de tact, deci simultan, cu alte microoperaţii, fapt pentru care ele nu se termină prin punct şi virgulă, ci doar prin virgulă.

Cu aceste precizări am putea trece la noua versiune de descriere funcţională a structurii din fig. 4.71, dar dorim ca aceasta să includă, faţă de cea din fig. 4.72, şi partea de supraveghere pe bază de paritate a procesului de înmulţire, ceea ce necesită unele comentarii suplimentare. Principiul de verificare are la bază faptul că înmulţirea se realizează prin adunări repetate, ori este cunoscută [Ra Fu -89, Prad- 86,Lola- 85,Buk- 88,Vasi- 93] verificarea operaţiei de însumare. Astfel, pentru doi vectori binari $A = (a_{n-1}, \ldots, a_i, \ldots, a_1, a_0, a_p)$ şi $B = (b_{n-1}, \ldots, b_i, \ldots, b_1, b_0, b_p)$, poate fi calculată paritatea predictivă s_p a vectorului sumă $S = (s_{n-1}, \ldots, s_i, \ldots, s_1, s_0, s_p)$ utilizând relaţia:

$$s_p = a_p \oplus b_p \oplus c_p \qquad (4.44)$$

în care $\oplus$ reprezintă operatorul de SAU EXCLUSIV (EXCLUSIV OR, EX - OR), iar c_p este paritatea corespunzătoare biţilor transport (carry) dată de $c_p = \sum_{i=0}^{n-1} c_{i-1}$, cu $c_{-1} = c_{in}$.

Pe de altă parte, corespunzător pozițiilor binare ale rezultatului sumă se calculează paritatea $s_{pR} = \sum_{i=0}^{n-1} s_i$, care se compară cu cea dată de relația (2.3), validând rezultatul obținut când sindromul de paritate $S_p = s_p \oplus s_{pR}$ prezintă valoarea 0 sau invalidându-l când S_p = 1 cu concomitentă generare a semnului de control de eroare (ERROR).

Prin extrapolare, pentru produsul binar în cazul particular (n = 8) acceptat, poate fi calculată paritatea predicativă p_p a vectorului produs $P\left(p_{15}, p_{14}, \ldots, p_1, p_0, p_p\right)$ uzitând de relația:

$$p_p = a_p \cdot b_p \oplus \sum_{i=0}^{7} \sum_{j=0}^{7} c_{ij} \qquad (4.45)$$

în care a_p și b_p reprezintă biți de paritate corespunzători înmulțitorului, înmagazinat inițial în bistabilul Q_p, respectiv deânmulțitului, înmagazinat inițial în M_p,iar $c_i = \sum_{j=0}^{7} c_{ij}$ - cu $c_{in} = 0$ și c_7 = cont - reprezintă paritatea transporturilor calculată la fiecare din cei 8 pași i (i = 0, 1,..., 7) prin arborele de circuite SAU EXCLUSIV aferent (c - EX - OR tree) și înmagazinată în bistabilul de manevră A_p atașat registrului acumulator.

Reacția dusă la circuitul SAU EXCLUSIV conectat pe intrarea bistabilului A_p permite calcularea parității transporturilor curente corespunzătoare produselor parțiale, astfel încât în final, după parcurgerea celor 8 pași, bistabilul A_p va înmagazina paritatea cumulată $\sum_{i=0}^{7} \sum_{j=0}^{7} c_{ij}$.

Paritatea predicativă p_p, dată de (4.45) este înmagazinată în Q_p, urmând a fi descărcată în magistrala de ieșire - în cazul când sindromul de paritate $S_p = 0$ - odată cu jumătatea mai puțin semnificativă $\left(p_7, p_6, \ldots, p_1, p_0\right)$ a produsului. Pentru că magistrala de ieșire are 9 linii (8 utile și una de paritate), odată cu predarea în aceasta a jumătății mai semnificative a produsului formată în registrul A, în mod opțional (aspect marcat cu linie punctată în fig. 4.71, poate fi predat un bit de paritate corespunzător pozițiilor binare ale acestei părți a produsului. Acest bit poate fi folosit, eventual, la ulterioare operații de verificare ale transferului de informație, dar se subliniază că el nu are nici-o semnificație în ceea ce privește problematica verificării operației de înmulțire supusă discuției. Înmagazinarea lui a fost prevăzută în al doilea bistabil de manevră, A_0 de indicare a deplasării la adunare, care este disponibil în urma terminării secvenței pașilor de adunare - deplasare. Valoarea binară încărcată în A_0 este generată de un arbore de circuite SAU EXCLUSIV (p - EX - OR tree), oricum necesar la calculul parității p_{Rp} a rezultatului produs. De astfel, acesta din urmă, necesar la generarea sindromului de paritate S_p, este prevăzut, în scopul economisirii de circuite, a fi calculat în mod diferit pentru cele două jumătăți ale produsului. Dacă pentru jumătatea mai

semnificativă paritatea se generează în paralel, prin arborele de circuite aferent (p - EX - OR tree), pentru jumătatea mai puțin semnificativă paritatea se generează serial prin intermediul bistabilului de manevră Q_{RP} într-o schemă cu o reacție la un circuit SAU EXCLUSIV conectat pe intrarea sa (de tipul celei prezentate la A_p - fig. 4.71, care permite formarea secvențială a parității în timpul deplasărilor prevăzute de relația (4.45). Sindromul de paritate este furnizat la ieșirea de eroare (ERROR) la a cărei activare este semnalată malfuncționarea la unitatea centrală de procesare și este invalidată predarea în OUTBUS a rezultatului produs.

Cu aceste detalieri, structurii din fig. 4.71 corespunde descrierea prin limbajul formal introdus din fig. 4.73.

declare register $A(7:0), A_0, A_P, Q(7:0), Q_P, Q_{Rp}, M(7:0), M_P, COUNT\ (0:2)$

declare bus INBUS $(7:0)$, $INBUS_p$, OUTBUS $(7:0)$, $OUTBUS_p$

BEGIN: $A \leftarrow 0, A_0 \leftarrow 0, A_P \leftarrow 0, Q_{RP} \leftarrow 0,$ COUNT $\leftarrow 0,$

INPUT: $M \leftarrow$ INBUS, $M_p \leftarrow INBUS_p$; $\}\ldots\ldots c_0$

$Q \leftarrow$ INBUS, $Q_p \leftarrow INBUS_p$; $\ldots\ldots c_1$

ADD: $A_0 \cdot A(7:0) \leftarrow A(7:0) + M(7:0) \times Q(0), A_p \leftarrow \sum_{I=0}^{7} c_i$; $\ldots\ldots c_2$

RIGHTSHIFT $A\,.\,Q \leftarrow A_0 . A\,.\,Q(7:1), Q_{RP} \leftarrow Q_{RP} \oplus A(0),$ COUNT $\leftarrow COUNT_{+1}$; $\ldots c_3$

TEST: if COUNT $0 \neq 1$ then go to ADD,

$Q_p \leftarrow A_p \oplus Q_p \wedge M_p\left(, A_0 \leftarrow \sum_{I=0}^{7} A(i)\right),$

$Q_{Rp} \leftarrow A_p \oplus Q_p \oplus M_p \oplus Q_{Rp} \oplus \sum_{i=0}^{7} A(i);$ $\}\ldots\ldots c_4$

OUTPUT: if $Q_{Rp} = 1$ then go to ERROR,

OUTBUS $\leftarrow A($, $OUTBUS_p \leftarrow A_0);$ $\}\ldots\ldots c_5$

OUTBUS $\leftarrow Q$, $OUTBUS_P \leftarrow Q_P$; $\ldots\ldots c_6$

ERROR: Send interrupt (trop) signal.

END:

Fig. 4.73

Relativ la descrierea din fig 4.73, la cele anterior expuse ar mai fi de adăugat că a fost prevăzută încărcarea sindromului $S_p = p_R \oplus p_{Rp}$ în bistabilul de manevră Q_{Rp}, oricum disponibil la terminarea pașilor de adunare - deplasare, în vederea testării sale la un ciclu de tact succesiv și a generării în consecință a semnalului de întrerupere înspre unitatea centrală de procesare (prevăzută la enunțul cu eticheta ERROR). De asemenea, microoperațiile de generare și predare a anterior

prezentatei posibilități opționale, corespunzătoare jumătății mai semnificative a produsului, au fost notate între paranteze.

Pentru claritate, inserăm tabelul din fig. 4.74 care cuprinde conturile registrelor structurii din fig. 4.71 pentru cazul particular al înmulțirii echivalenților binari corespunzători numerelor zecimale A = 233 și B = 243. În tabel apar pe lângă vectorii sumă S corespunzători părților mai semnificative ale produselor parțiale și vectorii transport $C=\left(\text{count}, c_6,\ldots,c_1,c_0\right)$ care permit urmărirea generării în secvență a biților de paritate stocați în A_p . Într-un al 9-lea pas, se prezintă etapa de verificare care implică paritatea predictivă p_p , calculată pe baza relației (4.45) corespunzătoare rezultatului produs constituit de echivalentul binar al numărului zecimal P = 56619.

Comparând mai întâi prin prisma performanței varianta dispozitivului de înmulțire binară fără verificare de paritate, descrisă prin ordinograma din fig, 4.36 și cea a aceluiași dispozitiv cu verificare, descrisă prin limbajul formal din fig. 4.37 căruia îi corespund semnele de control marcate prin linii punctate din fig. 4.35 se poate constata că cea de a doua necesită generarea prin unitatea de control a unui număr de 7 semnale de control $\left(c_0,c_1,\ldots,c_6\right)$, față de doar cele 6 $\left(c_0,c_1,\ldots,c_5\right)$ necesare primeia.

Admițând că la fiecare ciclu de tact se generează un semnal de control, penalitatea de performanță de 16,67% este doar aparentă, în realitate aceasta este dependentă de lungimea în biți a numerelor înmulțite descrescând cu aceasta. Astfel, luând în considerare bucla de adunări-deplasări, pentru cazul considerat când n = 8, rezultă o penalitate de performanță la nivelul trenului de tact de doar 5%, iar dacă numerele înmulțite ar avea lungimea de 16 biți penalitatea de performanță ar fi de 2,77%. Este însă adevărat că arborele c-EX - OR tree grevează asupra intervalului de timp necesar microoperațiilor controlate prin semnalul de control c_2 . Dar în acest context trebuie menționat că sumatorul paralel este indicat să fie sintetizat cu duplicarea transportului [Prad-86,RaFu-89,Vasi-93] pentru a acoperi defecte care de pe lanțul de transport și care, provocând același număr de erori în vectorul sumă și vectorul de transport, nu pot fi detectate. Astfel, prevăzând sumatorul care generează vectorul sumă normal de tipul cu propagare serială a transportului (ripple carry) și cel de duplicare cu anticiparea transportului (carry - lookahead), apare o compensare a întârzierilor corespunzătoare celor două lanțuri, iar aspectul relevat drept critic, aflat sub controlul semnalului c_2, este, cel puțin în parte, anihilat.

În al doilea rând, comparând cele două structuri prin prisma costului, verificarea prin paritate determină o penalitate, la nivelul registrelor, de 14,28 % și care poate fi, cu proximație, extrapolată la nivel global. Trebui însă remarcat că și aceasta descreşte cu majoritatea numărului de ranguri ale celor doi operanzi ajungând, la nivelul registrelor, pentru n = 16 la 7, 92 %, ceea ce judecat prin prisma integrării pe scară largă repezintă o investiție, apreciem, modică raportat la câștigul pe care îl prezintă asigurarea verificării prin paritate a execuției operației de înmulțire.

Pas	A_0	A	A_p	Q_{Rp}	Q	Q_p	M	M_p	COUNT	$COUNT_0$
0	0	00000000	0	0	11101001	1	11110011	0	000	1
1	0	M = 11110011 S = 11110011 C = 00000000 01111001	⊕ 0	⊕ 1	11110100	1	11110011	0	001	0
2	0	00000000 S = 01111001 C = 00000000 00111100	⊕ 0	⊕ 0	11111010	1	11110011	0	010	0
3	0	00000000 S = 00111100 C = 00000000 00011110	⊕ 0	⊕ 0	01111101	1	11110011	0	011	0
4	1	11110011 S = 00010001 C = 11111110 10001000	⊕ 1	⊕ 1	10111110	1	11110011	0	100	0
5	0	00000000 S = 10001000 C = 00000000 01000100	⊕ 1	⊕ 1	10101111	1	11110011	0	101	0
6	1	11110011 S = 00110111 C = 11000000 10011011	⊕ 1	⊕ 0	1010 1111	1	11110011	0	110	0
7	1	11110011 S = 10001110 C = 11110011 11000111	⊕ 1	⊕ 0	0101 0111	1	11110011	0	111	0
8	1	11110011 S = 10111010 C = 11000111 11011101	⊕ 0	⊕ 0	0010 1011	1	11110011	0	000	1
9	0	⊕ ⊕ 0		0	⊕ 0	⊕ 0	⊕ 0			

Fig. 4.74

Referindu-ne în continuare la structuri matriciale combinaționale [Prad-86,Haye-88,Dora-88], care permit execuția operației de înmulțire practic într-un singur pas pe seama unei corespunzătoare creșteri a numărului de circuite, să considerăm, pentru simplitatea prezentării, că cele două numere întregi fără semn care se supun operației de înmulțire sunt date de vectorii binari

$A = (a_2, a_1, a_0)$ și $B = (b_2, b_1, b_0)$, mărimile lor fiind

$A_M = \sum_{i=0}^{2} a_i \cdot 2^i$, respectiv $B_M = \sum_{i=0}^{2} b_i \cdot 2^i$.

Vectorul produs $P = (p_5, p_4, p_3, p_2, p_1, p_0)$ are mărimea dată de relația:

$$P_M = \sum_{i=0}^{2} a_i 2^i \cdot B_M = \sum_{i=0}^{2} 2^i \left(\sum_{j=0}^{2} a_i b_j \cdot 2^j \right) \qquad (4.36)$$

Întrucât produsele aritmetic și logic pentru numere de un bit coincid, fiecare dintre produsele de 1 bit a_i b_j din relația (4.46) necesită pentru evaluare o singură poartă logică ȘI cu 2 intrări.

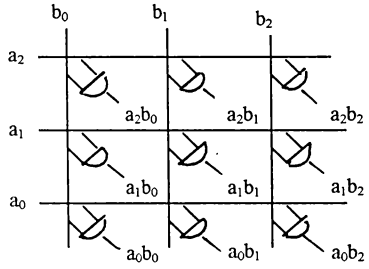


Fig. 4.74

Prin urmare, în cazul general al unor operanzi de n biți, este necesară o matrice de n x n porți ȘI cu două intrări dată în fig. 4.75. Însumarea termenilor a_i b_j este realizată în acord cu relația (4.46) printr-o a doua matrice, de această dată, formată, în cazul general, de n x (n - 1) celule de sumatoare complete, care pentru cazul particular abordat, este prezentată în fig. 4.75.

Matricea de sumatoare complete este în esență un dispozitiv de însumare cu transport propagat bidimensional, deplasările implicate de factorii 2^i și 2^j din relația (4.46) fiind complementate prin dispunerea spațială a celulelor sumatoare complete.

Durata procesului de înmulțire corespunzătoare acestei structuri matriceale combinaționale este determinată de propagarea transportului în cazul cel mai defavorabil, fiind dată, pentru operanzi de n biți, de relația:

$$T_M = 2(n-1)d + d' \qquad (4.47)$$

în care d și d' reprezintă timpii de propagare corespunzători unei celule de însumare completă, respectiv unei porți ȘI.

Componenta de cost a unui astfel de dispozitiv de înmulțire crește cu pătratul lui n, dar organizarea sa matriceală îl face adecvat pentru realizarea în tehnologie VLSI.

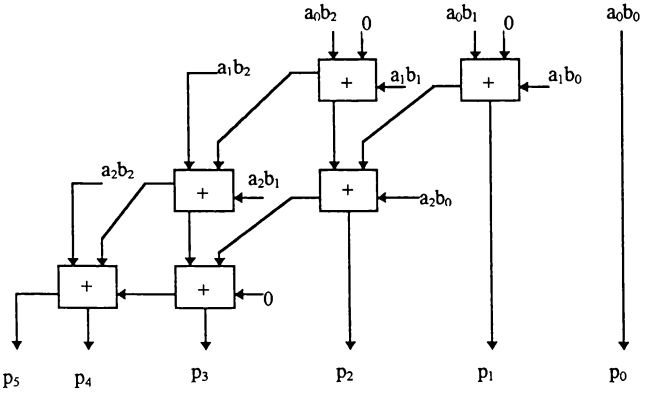


Fig. 4.76

Structura matriceală combinațională poate fi uniformizată dacă se apelează la o celulă combinată, care să includă o poartă ȘI și o celulă sumator complet, constituind o așa numită celulă de înmulțire M. O astfel de celulă, la care au fost prevăzute circuitele de duplicare a transportului în vederea implementării verificării de paritate, este prezentată în fig. 4.77. În celula M se realizează evaluarea expresiei aritmetice $\left(a_i b_j + x + y\right)$ implementată prin funcțiile boleene:

$$s = \left(a_i \wedge b_j\right) \oplus x \oplus y \qquad (4.48)$$

$$c = c' = \left(a_i \wedge b_j \wedge Z\right) \vee \left(a_i \wedge b_j \wedge J\right) \vee \left(x \wedge J\right) \qquad (4.49)$$

în care $\wedge$ și $\vee$ semnifică operațiile logice de SAU, respectiv ȘI, iar c și c' reprezintă funcțiile de transport utilizate la formarea produsului, respectiv la verificarea prin paritate a generării corecte a acestuia.

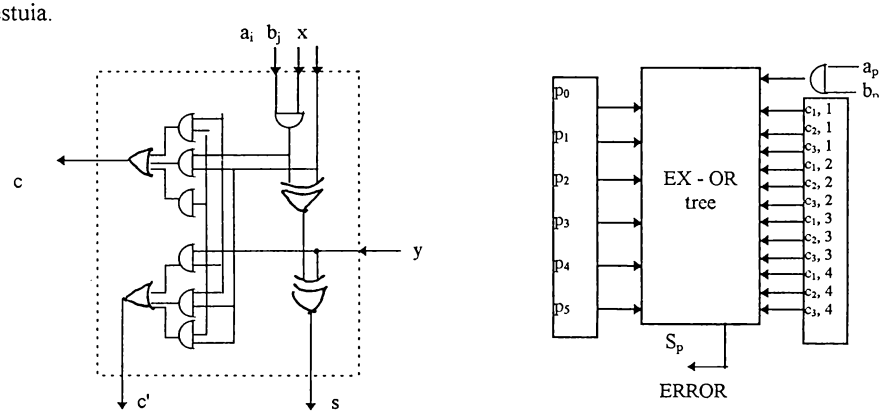


Fig. 4.77

Uzitând de celula M definită prin ecuațiile logice (4.48) și (4.49), în fig. 4.78 se prezintă structura matricială combinațională care permite implementarea pe durata unui singur ciclu de tact a operației de înmulțire binară cu verificarea bazată pe paritate a formării corecte a produsului.

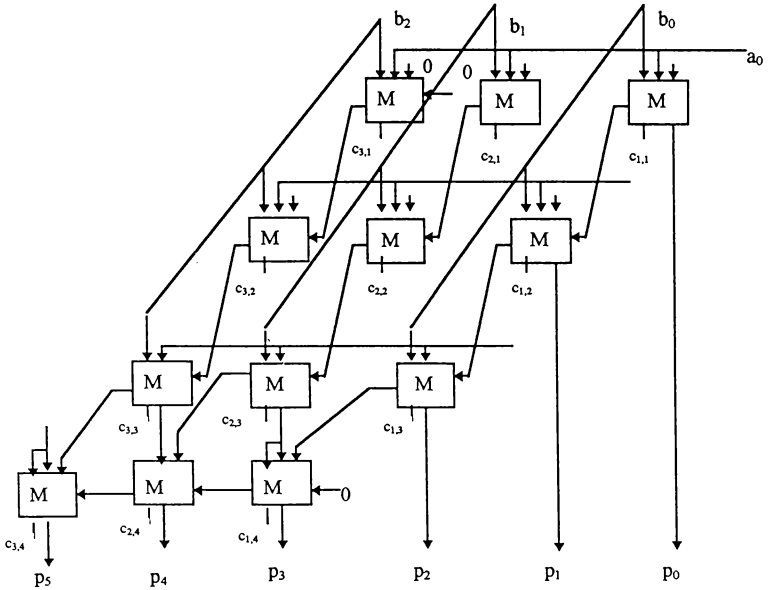


Fig. 4.78

Generarea sindromului de paritate S_p are la bază o formă adaptată a aceluiași control exprimat prin relația de calcul (4.46) a parității predictive, adică:

$$S_p = a_c \cdot b_c \oplus \sum_{i=1}^{3} \sum_{j=1}^{4} c_{i,j} \oplus \sum_{i=0}^{5} p_i \qquad (4.50)$$

Numărul celulelor de înmulțire dat, în cazul general al unor operații de n biți de n x (n+1) trebuie judecat prin prisma disponibilităților pe care le oferă tehnologia VLSI. De asemenea, grevarea asupra duratei perioadei tactului prin întârzierea suplimentară pe arborele de circuite SAU EXCLUSIV (EX - OR tree)trebuie apreciată prin posibilitățile tip scheme PLA și GA de care dispune tehnologia VLSI pentru implementarea unor asemenea structuri combinaționale.

Procedând de o manieră asemănătoare, verificarea bazată pe paritate poate fi extinsă asupra înmulțirii binare a numerelor cu semn uzitând de celule de înimulțire adecvate, având la baza sintezei proceduri algoritmice care permit accelerarea procesului,cum ar fi procedura Both cu recodificarea înmulțitorului, precum și asupra conceptului de înmulțire bazat pe salvarea transportului [KrSh-86,TaSe-86.BeTr-89,KiWe-89,BlMa-90,PaHe-90,RaKI-90,NgMa-94,PaHe-94].

4.3.2.Fructificarea structurilor ASIM pentru creşterea eficienţei autocontrolului schemelor PLA

În paragraf au fost selectate şi supuse analizei unele părţi ale schemelor UAL care au fost găsite ca pretabile unei combinaţii cu structuri ASIM. Astfel, referindu-ne mai întâi la detecţia erorilor prin coduri sumă de control, se conturează posibilitatea substituirii sumei de control dată de definiţia (4.3) printr-o informaţie redundantă generată prin mecanismele specifice ASIM-urilor.

Definiţie 4.5: Un cod sumă de control este un set descris prin:

$$\{(x_c,x_{n-1},\ldots,x_i,\ldots,x_0)|(x_i,x_c\in Z_2b) \text{ şi } (x_c=\sum_{i=0}^{n-1}\oplus(x_i\oplus x'_{i-1})) \text{ şi } (x'_{i-1}=\begin{cases}2\,x_{i-1}, \text{ dacă } b_{i-1}^{b-1}=0\\ 2\,x_{i-1}\oplus m, \text{ dacă } b_{i-1}^{b-1}=1\end{cases})\}$$

Legat de notaţiile folosite se impun le precizări. În ceea ce priveşte pe b,acesta reprezintă un număr întreg mai mare decât 1, constituind numărul de biţi ai unui byte, conform cu cele conţinute în fig.4.79.

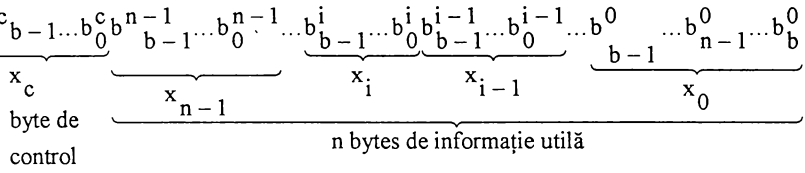


Fig.4.79

În fig.4.79 a fost marcat bitul cel mai semnificativ b_{b-1}^{i-1} al byte-ului (i-1), care intervine decisiv în calculul valorii x'_{i-1}, aceasta stabilind în mod recurent, în ultima instanţă, valoarea byte-ului sumă de control x_c. În definiţia 4.5. mai apare notaţia m, constituind o mască specifică polinomului generator ce stă la baza sintezei ASIM-ului prin intermediul căruia este generată noua sumă de control. Masca m implică o configuraţie binară având acele poziţii pe 1 care corespund termenilor din expresia polinomului generator cu excepţia celui mai semnificativ. Legat de gradul polinomului generator, acesta se impune a fi egal cu valoarea aleasă pentru b, iar în ceea ce priveşte expresia acestuia, în baza celor dezbătute pe larg, se impune ca aceasta să fie primitivă.

Să exemplificăm formarea şi apoi aplicarea noii sume de control în contextul aceluiaşi exemplu expus în paragraful 4.3.1.1. Astfel, avem numărul A=(2,2,4,7), cu n=4 şi b=3, a cărui configuraţie binară este: A= 010 010 100 111.

x_3 x_2 x_1 x_0

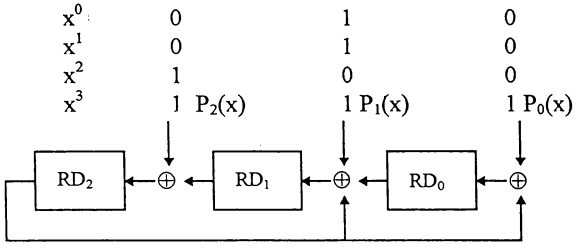
Plecând de la valoarea b=3, se trece la alegerea unui polinom generator de grad b=3, când există o singură alternativă $G(x)=x^3+x+1$. În situaţia mult mai frecventă când există mai multe polinoame concurente, se va alege acel primitiv cu număr maxim de termeni, iar dacă concurenţa persistă, alegerea devine arbitrară-aceste criterii bazându-se pe cele prezentate în capitolul 3. Masca m corespunzătoare lui $G(x)=x^3+x+1$, o obţinem ignorând pe x^3, rezultând, prin urmare, configuraţia

binară 011, cu bitul cel mai puțin semnificativ în partea dreaptă. Cu aceste precizări mecanismul analitic de generare al sumei de control în baza definiției 4.5 este expusă în fig.4.80.

$$x_0 = 111 \rightarrow x'_0 = 2x_0 \oplus m = (110) \oplus (011) = (101)$$
$$x_1^* = x_1 \oplus x'_0 = (100) \oplus (101) = (001) \rightarrow x'_1 = 2x_1 = (010)$$
$$x_2^* = x_2 \oplus x'_1 = (010) \oplus (010) = (000) \rightarrow x'_2 = 2x_2 = (000)$$
$$x_c = x_3^* = x_3 \oplus x'_2 = (010) \oplus (000) = (010)$$

Fig.4.80

Aceeași valoare a sumei de control se obține uzitând de ASIM-ul prezentat schematic în fig.4.81, care conține, în tabelul alăturat schemei, conținuturile rangurilor RD0 la RD2 ale registrului de deplasare.



RD_0	RD_0	RD_0
0	0	0
1	1	1
0	0	1
0	0	0
0	1	0

Fig. 4.81

Evident există și metoda analitică de formare a sumei de control, în fond a semnăturii, "paralel",care aplicată schemei din fig.4.81 ne conduce la următoarele polinoame $P_0(x)$ la $P_2(x)$, precum și la polinomul cumulat P(x):

$$P_0(x) = x^3 \quad | \cdot x^0$$
$$P_1(x) = x^3 + x + 1 \quad | \cdot x^1$$
$$P_2(x) = x^3 + x^2 \quad | \cdot x^2$$
$$P(x) = x^5 + x^4 + x^4 + x^2 + x + x^3 = x^5 + x^3 + x^2 + x \qquad (4.51)$$

Pentru a obține polinomul asociat sumei de control se efectuează oprtația de împărțire în GF(2) a polinomului P(x) dat de (4.51) la polinomul generator G(x) și se obține restul asociat semnăturii R(x)=x, care corespunde celor din fig.4.80 și 4.81.

Procedând în manieră similară, pentru cel de al doilea număr al exemplului B=(3,3,4,3) se obține suma de control (110) sau, în termeni polinomiali, $R_B(x) = x^2 + x$.

Înainte de a trece la detalierea procedurii de control, trebuie să scoatem în relief simplitatea constructivă a mecanismului hardware de generare a sumei de control, care se prezintă net superioară prin prisma costului în raport cu generarea sumei modulo q (par.4.3.1.1) datorită

circuitului de formare şi propagare a transportului.Acest aspect exte evident încă de la valori reduse ale lui n,devenind tot mai pronunţat odată cu creşterea lui n (urmăribil pe fig.4.60).

În ceea ce priveşte controlul, acesta se efectuează în maniera sugerată, în fig.4.59 şi va fi expus prin intermediul aceluiaşi exemplu, anterior amintit. Luând în considerare echivalenţa zecimală(de fapt,octali) ai celor două sume de control, vom aduna deci numerele A=(2,2,2,4,7) şi B=(6,3,3,4,3) şi vom obţine conform cu cele din fig.4.82 vectori sumă S şi transport C.

A=(2,2,2,4,7) =	010	010	010	100	111	
B=(6,3,3,4,3) =	110	011	011	100	011	
S =		101	110	001	010	= (5,6,1,2)
C=		100	111	001	110	= (4,7,1,6)

Fig.4.82

Operând vectorii S şi C conform cu cele din procedura dată de definiţia (4.5) se obţin pentru sumele de control valorile 0,respectiv 4.În continuare, se generează din sumele de control corespunzătoare celor două numere ,A şi B, şi din suma de control corespunzătoare transportului C o nouă sumă de control, ea rezultând 0, care este comparată cu control 0 a sumei S.

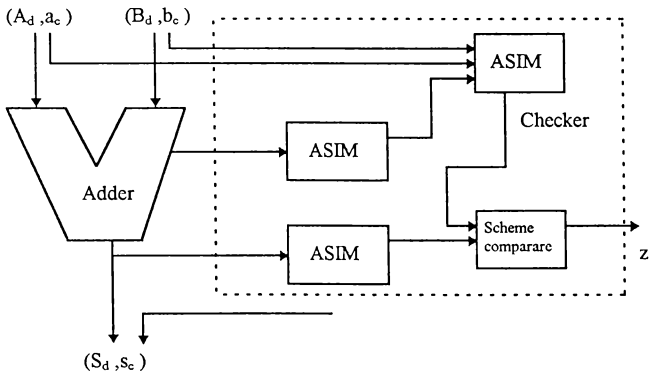


Fig. 4.83

La modul general, prin asemănare cu fig.4.59,avem pentru noua metodă de control schema bloc din fig.4.83. Justificarea teoretică a metodei se bazează pe amplu prezentele teoreme 4.3, 4.4 şi 4.5 din paragraful 4.3.1.2,a căror adaptare considerăm că nu mai este necesară. Desigur, în detrimentul performanţei, cele trei ASIM-uri din fig.4.83 ar putea fi redusă prin serializarea procesului de formare al sumei de control. Oricum, schema din fig.4.83 necesită un număr de circuite elementare şi prin urmare, un cost mult mai redus faţă de cea din fig.4.59. Dar aspectul legat de cost nu este singurul favorabil schemei din fig.4.83,la aceasta adăugându-se, în contextul autocontrolului, deosebit de importanta capacitate de detecţie a potenţialelor defecte. Prin această prismă, plecând de la detaliile de monstraţiei din [Vlăd-82], rezultă că metoda de comprimare prin ASIM-uri posedă o

probabilitate de recunoaştere ca funcţional corectă a unei unităţi testate defecte inferioare celei bazate pe suma de control modulo. Coroborând aceste două aspecte cu faptul că degradarea de performanţă este aceeaşi pentru cele două scheme, putem conchide că noua versiune propusă, bazată pe ASIM-uri, este superioară celei din [RaFu-89].

La cele expuse ar mai trebui adăugat faptul că definiţia (4.5) permite adaptarea formării acestei foarte eficiente sume de control pentru verificarea conţinuturilor memoriilor ROM (PROM,EPROM,şa). De asemenea, ea poate fi folosită la generarea de coduri combinate, prezentate extins în paragraful 4.3.1.2, şi anume, la generarea biţilor suplimentari $|X|_m$ (fig.4.62), aspect care, în baza celor anterior expuse, considerăm că nu necesită o detaliere.

În fine, referindu-ne la adunarea corectată în raport cu eroarea singulară şi, cu atât mai mult, la înmulţirea binară controlată prin cod de paritate, rezultă creşterea masivă în complexitate a circuisticii revendicate de părţile de verificare. Pentru acestea se pretează un control prin ASIM-uri doar în manieră "Watch-dog" asemănător cu aplicaţiile prezentate în următoarele două capitole.

4.4. Concluzii

Urmărind găsirea a cât mai multor aplicaţii din domeniul calculului pentru structurile ASIM modificate- în capitolul trei pentru obţinerea unei capacităţi de detecţie superioară structurilor convenţionale- în prezentul capitol, am investigat amplu trei obiective, anume schemele secvenţiale sincrone, schemele PLA şi schemele UAL.

Astfel, cercetările referitoare la facilitarea testării schemelor secvenţiale sincrone a avut drept ţintă metodele de scanare, care prin intermediul standardului IEEE 1149.1 (şi dezvoltări) câştigă teren în ceea ce priveşte proeictarea sistemelor de calcul. Au fost amplu analizate toate metodele cunoscute în vederea combinării cât mai eficiente a structurilor ASIM cu acestea în scopul eficientizării autotestării. În această parte lucrarea prezintă următoarele elemente originale:

a_1) Sinteza aspectelor dificile ale testării schemelor secvenţiale sincrone cu sublinierea, în mod aparte, a unor probleme de testabilitate.

a_2) Analiza comparativă în manieră unitară a metodelor de scanare coroborând caracteristici de performanţă/cost cu cele de capacitate de detecţie.

a_3) Critica din surse de literatură consacrate a implementării autocontrolului prin tehnici BILBO şi propunerea substituirii acestora prin mai eficientele structuri ASIM.

a_4) Propunerea unui latch LSSD care încorporat în inele interne şi externe în conformitate cu standardul IEEE 1149.1 să permită o penalitate minimă a performanţei, context în care se face o încercare de propunere în vederea includerii ideii în cadrul standardului.

În ceea ce priveşte schemele PLA (FPLA), cu unele referiri la schemele GA (FPGA) investigaţiile au fot orientate înspre incorporarea unor redundanţe built-in în scopul facilitării testabilităţii acestor scheme, fiind cunoscută problematica dificilă pe care o prezintă din punct de vedere al testării. În acest sens, cercetările s-au îndreptat spre testare independentă de funcţii urmărindu-se intercalarea, cu elemente hardware redundante, de aşa manieră a structurilor ASIM

încât ele să-şi aducă contribuţia la creşterea testabilităţii. Referitor la aceste aspecte, lucrarea conţine următoarele elemente originale:

b_1) Analiza, bazată pe un număr mare de repere bibliografice a tipurilor de defecte specifice acestor scheme şi modului lor de punere în evidenţă.

b_2) Definirea originalei noţiuni de comparare prin paritate cumulativă multiplă care permite intercalarea structurilor ASIM în scheme PLA.

b_3) Efectuarea unei analize comparative prin triplul impact cost(exprimat prin număr de circuite elementare revendicate de sinteză)/performanţă(exprimată prin număr de cicluri de clock necesare trecerii experimentului de testare)/capacitatea de detecţie a defectelor (exprimată prin probabilitatea de recunoaştere ca funcţional corectă a unei unităţi testate defecte) relativ la malfuncţionările singulare sau multiple de tip stuck, bridging şi crosspoint) între două metode consacrate [Prad-86,TrAF-87] şi cea nouă propusă în lucrare.

b_4) Introducerea teoremei originale 4.2 care caracterizează capacitatea de detecţie a variantelor propuse,cu ASIM-E respectiv cu ASIM-EC.

b_5) Elaborarea a 8 leme (4.9÷4.16) adaptate la aplicarea ASIM-urilor după unele din [TrAF-87] bazate doar pe bit de paritate.

b_6) Evaluarea capacităţii de trecere a experimentelor de testare pentru cele trei metode detaliate.

În fine, ultima parte se referă la încercarea găsirii locului optim al ASIM-urilor în contextul implementării, în scheme UAL, a operaţiilor de adunare binară şi înmulţire binară.Au fost analizate doar acele metode care să permită aducerea unor îmbunătăţiri. În acest cadru, lucrarea conţine următoarele elemente originale.

c_1) Critica unor greşeli din sursre de literatură consacrate [RaFu-89 (ecuaţia booleană de generare a biţilor sumă-pag.416 şi sinteza sumatorului din fig.8.26-pag.417, precum şi tabelul de la pag.427)] a unor aspecte legate de problematica analizată.

c_2) Propunerea, prin definiţia 4.5, a unor noi coduri bazate pe sume de control rezultate ca urnare a aplicării mecanismelor ASIM.

c_3) Propunerea unei scheme de adunare având autocontrolul implementat prin mai eficientele şi mai puţin costisitoarele structuri ASIM în raport cu sumatoarele modulo.

c_4) Includerea unor propuneri de aplicare a structurilor ASIM în generarea de coduri combinate precum şi în schema de adunare cu corecţia erorii singulare şi în schema de înmulţire pentru eficientizarea autocontrolului.

5. Aplicarea noilor structuri ASIM la sinteza de module slave conectate la o magistrală standard

Prezentul şi proximul capitol îşi propune exemplificarea prin realizări practice a deducţiilor teoretice cuprinse în capitolele anterioare ale tezei. Aceste lucrări aparţin perioadei de activitate a autoarei în cadrul Institutului de Cercetare Ştiinţifică şi Inginerie Tehnologică pentru Automatizări-filiala Timişoara. Asupra lor s-au făcut modificări rezultate din cercetările întreprinse de autoare în perioada de activitate ca şi cadru didactic.

Realizările sunt orientate pentru familia de module hardware şi software MULTIPROM constituind adaptarea unor configuraţii de structură pentru magistrala standard MULTIPROM în esenţă de fapt magistrala INTEL MULTIBUS I. Pentru teză este nerelevantă conectarea la acest tip de magistrală rămânând totuşi importantă ideea de magistrală standard.Pentru a concretiza ideile teoretice a fost preferat cadrul constituit de familia MULTIPROM pe care se vor grefa modificări rezultate din investigaţiile întreprinse în partea teoretică. Prin urmare, în prezentul capitol nu vor fi detaliate problematica magistralei, ea fiind dezvoltată în variate referinţe de literatură [Vasi-87a la d,Vasi-88 a la c şi e,h,g,Vasi-89a].

În prezentul capitol este descris unul dintre modulele hardware aparţinând familiei MULTIPROM şi anume analizorul de semnături, MPCB-04, asupra căruia se vor întreprinde unele modificări.În cadrul lucrării s-a utilizat semantica de reprezentare a schemelor derivată din standardul ANSI/IEEE 91-1988 constituind reglementările de reprezentare a schemelor sintetizate în IPA [Vasi-87d,Vasi-88d şi h la l]

5.1.Funcţiile unui modul şi configurarea la nivel bloc a acestora

În intenţia de a sintetiza un modul hardware care să permită autocontrolul în baza noilor structuri ASIM introduse în capitolul trei vom enumera mai întâi funcţiile pe care trebuie să le asigure orice realizare bazată pe testare de sindrom, fie că ea este încorporată sub forma unui modul destinat autocontrolului, fie că ea este de sine stătătoare sub forma unui aparat de testare portabil.Astefl, orice analizor de semnătură, independent de metoda de comprimare implementată-numărare de tranziţii, numărare de unităţi binare, generare de biţi de control la coduri ciclice sau la coduri cu resturi-implică, în primul rând un bloc de control al ferestrei, acesta din urmă reprezentând intervalul de timp pe durata căreia are loc procesul de comprimare. Fereastra este "deschisă" prin intermediul unui semnal "START" şi este închisă prin intermediul unui semnal "STOP", cele două semnale impunând programarea lor.În acest context,se impune subliniată funcţia importantă care se realizează prin programarea de aşa manieră a semnalelor START şi STOP care să permită funcţionarea ciclică, în buclă, a unităţii verificate. În acest mod este posibilă punerea în evidenţă a unor manifestări funcţionale cu caracter intermitent declanşate de defecţiuni temporare.Pe lângă funcţia de deschidere respectiv închidere a ferestrei realizată printr-un bloc destinct al modulului, o a doua funcţie prezentă în majoritatea analizoarelor de semnături este aceea de programare a

fronturilor semnalelor de START, STOP, pe de o parte şi TACT şi DATE pe de altă parte. Prin programarea fronturilor anterior sau posterior al semnalelor de START şi STOP este posibilă deschiderea unor ferestre de lungime variabilă.Demarând investigarea cu fereastra de lungime maximă poate fi întreprinsă căutarea malfuncţionării restrângând durata ferestrei prin programarea corespunzătoare a fronturilor semnalelor de START şi STOP. Pot rezulta astfel informaţii importante legate de modul de manifestare a unui defect în vederea localizării acestuia.Pe de altă parte, având în vedere diferitele tipuri de tehnologii precum şi bascularea pe front anterior sau posterior a elementelor de structură cele mai intimă, reprezentate de bistabile, în acelaşi scop al creşterii posibilităţilor de punere în relief cât mai variată a unei malfuncţionări potenţiale este prevăzută capacitatea de programare a fronturilor semnalelor de tact, uzual semnalul cu frecvenţa cea mai mare din cadrul unităţii testate şi a semnalelor de date.

În al treilea rând, în vederea creşterii suplimentare faţă de problematica supusă analizei în capitolul 3, a capacităţii de detecţie, reacţiile schemei secvenţiale liniare sunt programabile fiind utilizate la sinteza a trei polinoame generatoare primitive şi ireductibile, şi anume:

$$G_1(x)=x^{24}+x^7+x^2+x+1$$
$$G_2(x)=x^{16}+x^{12}+x^3+x+1 \qquad (5.1)$$
$$G_3(x)=x^8+x^4+x^3+x^2+1$$

În mod original schema este prevăzută să permită alegerea prin program a unuia dintre polinoame în conformitate cu care se realizează procesul de comprimare şi de detecţie a malfuncţionării. Pentru a acoperi o parte dintre erorile care nu sunt puse în evidenţă prin intermediul comprimării bazate pe acest pe acest prim polinom generator sunt prevăzute alte două polinoame în conformitate cu (5.1), în vederea creşterii rezoluţiei de detecţie a potenţialelor erori. Alegerea polinoamelor a avut la bază următoarele raţionamente:

a) Plecând de la caracteristicile familiei MULTIPROM s-a stabilit mai întâi numărul de semnale supuse procesului de comprimare fiind adoptată valoarea de 48 rezultată din adunarea la cele 16 linii de date a celor 24 linii de adresă şi a unui număr de 8 semnale de control. Rezultă lungimea registrului de deplasare de 48 de biţi corespunzător schemei secvenţiale liniare.

b) Plecând de la faptul că pe magistrala MULTIPROM pot fi configurate atât sisteme pe 16 cât şi pe 8 biţi, s-a considerat utilă, plecând de la analiza semnalelor celor două categorii de sisteme, a utilizării unor polinoame de grad 24, 16 şi 8, toate acestea asigurând divizibilitatea determinatei valori acoperitoare de 48.

c) Pentru fiecare din gradele amintite (24,16 şi 8) există câte o familie de polinoame generatoare dintre care alegerea s-a făcut în baza cunoscutului deziderat de asigurare a frecvenţei cele mai mari de comprimare a fluxurilor uinformaţionale în semnături. [Vlăd-82]. În baza acestei cerinţe, pe lângă proprietatea de primitivitate-implicit ireductibilitate a polinoamelor, alegerea ţinteşte unul dintre polinoamele cu număr minim de termeni. În mod favorabil pentru reducerea numărului de circuite dar şi a complexităţii interconexiunilor schemei secvenţiale liniare pentru toate polinoamele a

fost aleasă valoarea de 5 pentru număr de termeni.(5.1). Această valoare reprezintă minimul de termeni pentru polinoame primitive de gradul 8,16,24.

d) Alegerea expresiilor particulare din relația (5.1) s-a bazat suplimentar pe utilizarea în comun a unor circuite SAU EXCLUSIV care să permită simplificarea cât mai avansată a schemei secvențiale liniare generatoare a semnăturilor.

Asigurarea acestei reconfigurabilități originale la nivelul analizorului de semnături cu intrări multiple (ASIM) permite o majorare a potențialelor erori şi implicit defecte acoperite prin această metodă de control.

Pe lângă funcțiile de programare a frontului respectiv de programare a lungimii pe care se face comprimarea secvențelor binare culese (24,16 sau 8 canale), se impune posibilitatea de a programa tipurile de ciclu maşină care să fie luate în considerare în procesul de comprimare, precum şi programarea controlului de inițializare a schemei. Pentru aceasta din urmă se prevede şi un bloc special de a fi implementat hardware de declanşarea printr-un semnal dedicat specific magistralei MULTIPROM (INIT/).

În vederea declanşării procesului de programare a funcțiilor menționate prin intermediul unui bloc de programare modul (BPM) este prevăzut un bloc de selecție (BS). Modulul hardware supus discuției este prevăzut a fi utilizat în conjuncție cu programele de autotest care au menirea de a activa funcțional părți componente ale sistemului de calcul. Traficul informațional la nivelul magistralei este prevăzut a fi captat la nivel de modul şi supus procesului de comprimare iar informația generată este comparată cu una etalon corespunzătoare funcționării corecte. Operația de comparare este prevăzută a fi efectuată prin programe fie doar la finele întregii secvențe de test, fie la momente de timp prestabilite programate şi acestea. Strategia de testare prevede detecția potențialelor malfuncționări prin generarea semnăturilor corespunzătoare la cele 24 de linii de magiastrală a căror informație este preluată prin circuite de conectare la magistrală (CCM) destinate acestui scop. În vederea localizării grupelor de linii se pot genera semnături pe 16, respectiv 8 linii de magistrală.

În baza celor prezentate, rezultă schema bloc cu configurația din fig.5.1.

Se distinge blocul de control al ferestrei (BCF) care este prevăzut a fi declanşat de către 5 dintre semnalele de comandă ale magistralei MULTIPROM.De asemenea, apare blocul de control inițializare (BCI) declanşat prin semnalul INIT/. Pentru selecție este prevăzut blocul (BS) care declanşează blocul de programare (BPM), acesta din urmă având sub control (BCI), (BCF) precum şi partea esențială a întregului modul constituită de schema secvențială liniară cu reacții programabile, în fond ASIM-ul cu reacții programabile. Pentru predarea semnăturii generate în magistrală este prevăzut blocul CCM.

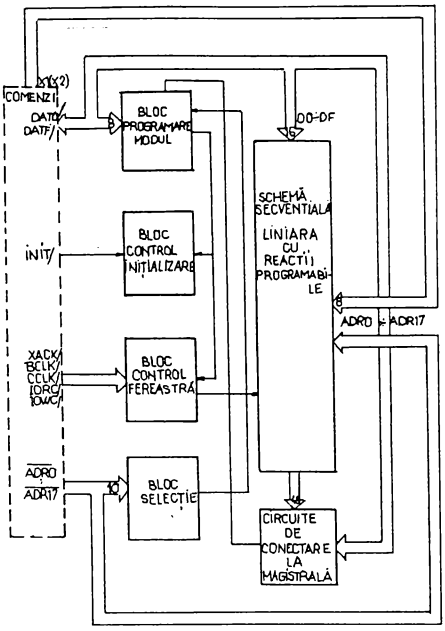


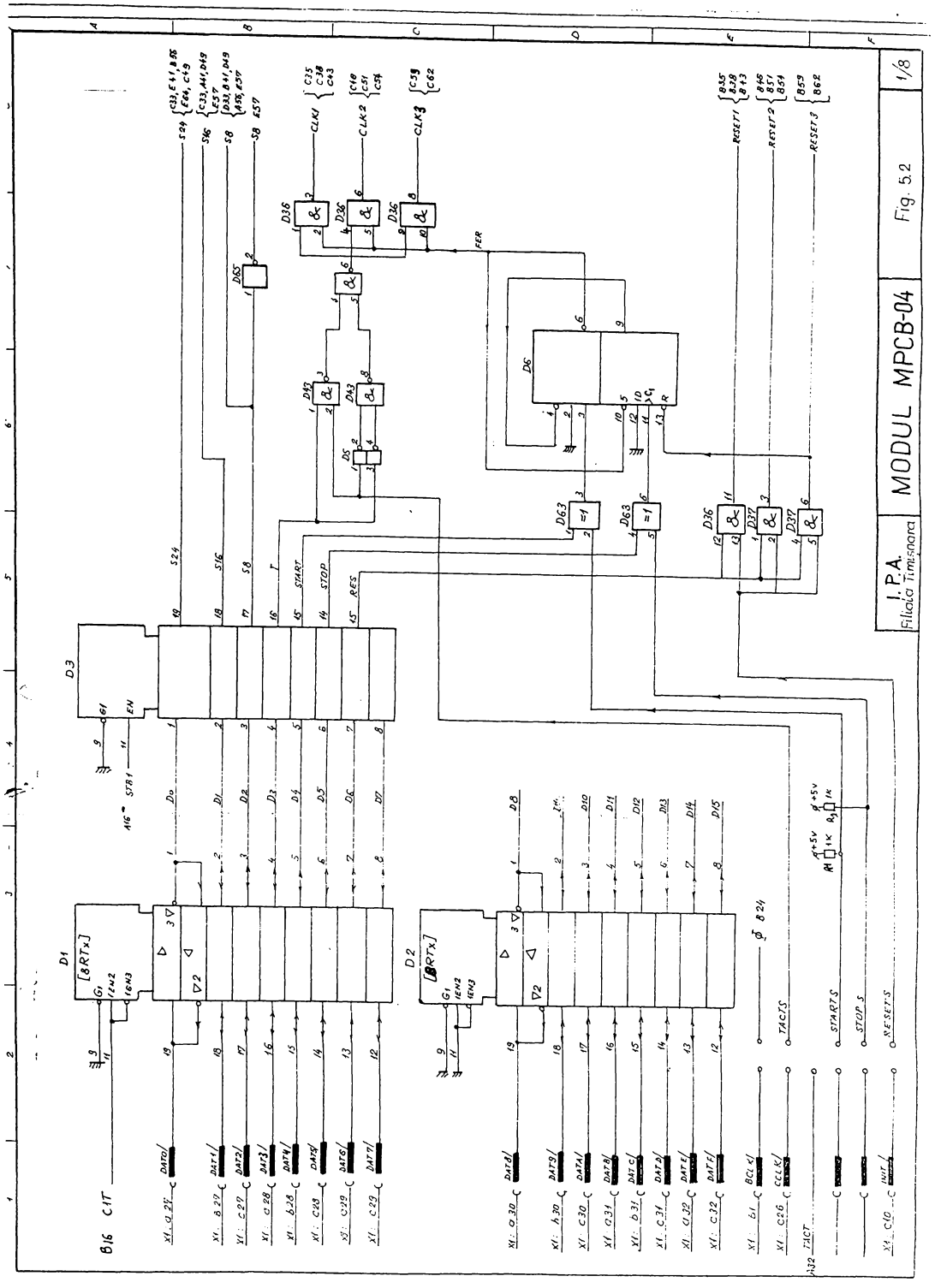
Fig.5.1

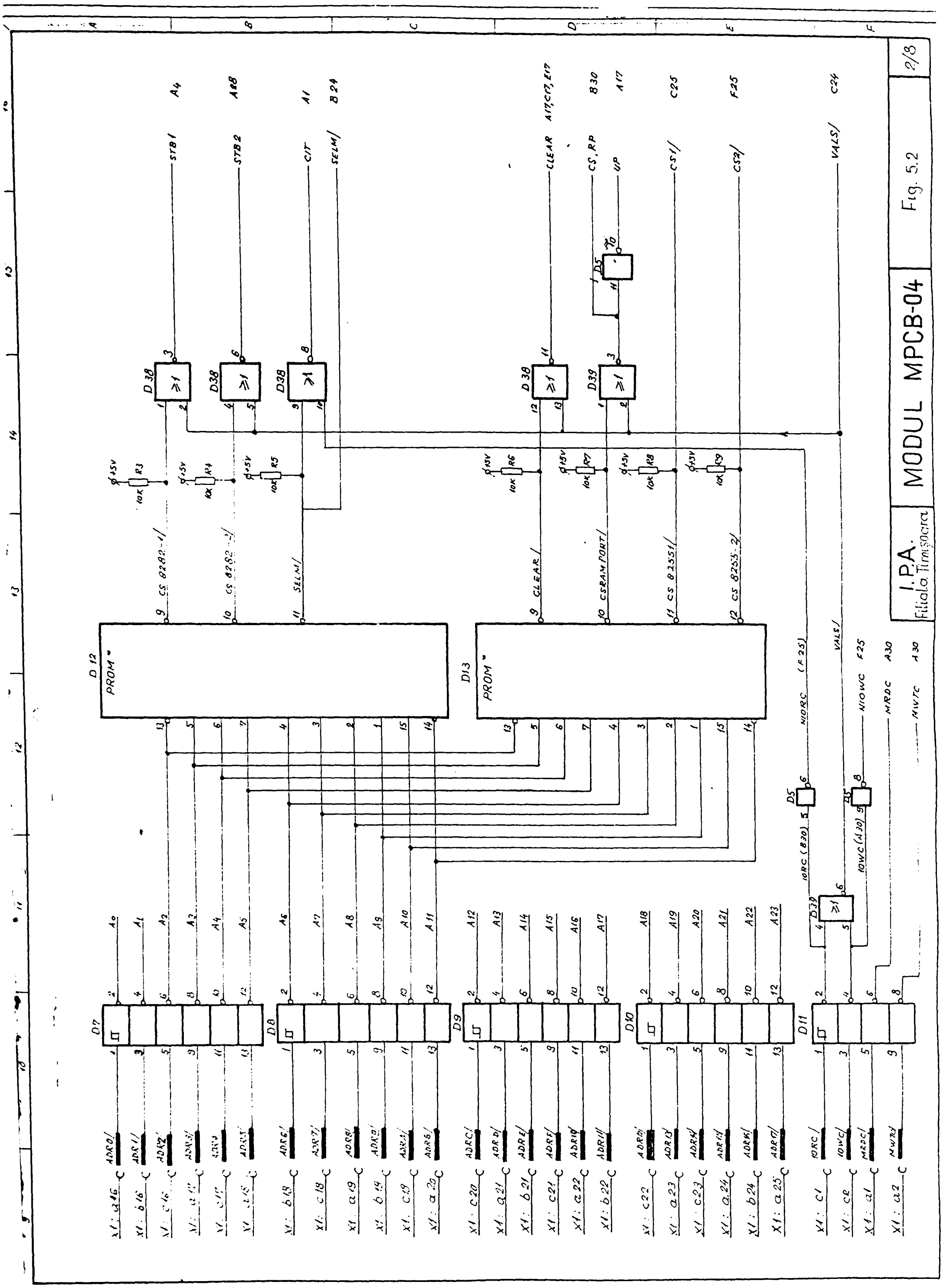
5.2. Sinteza de detaliu a blocurilor componente

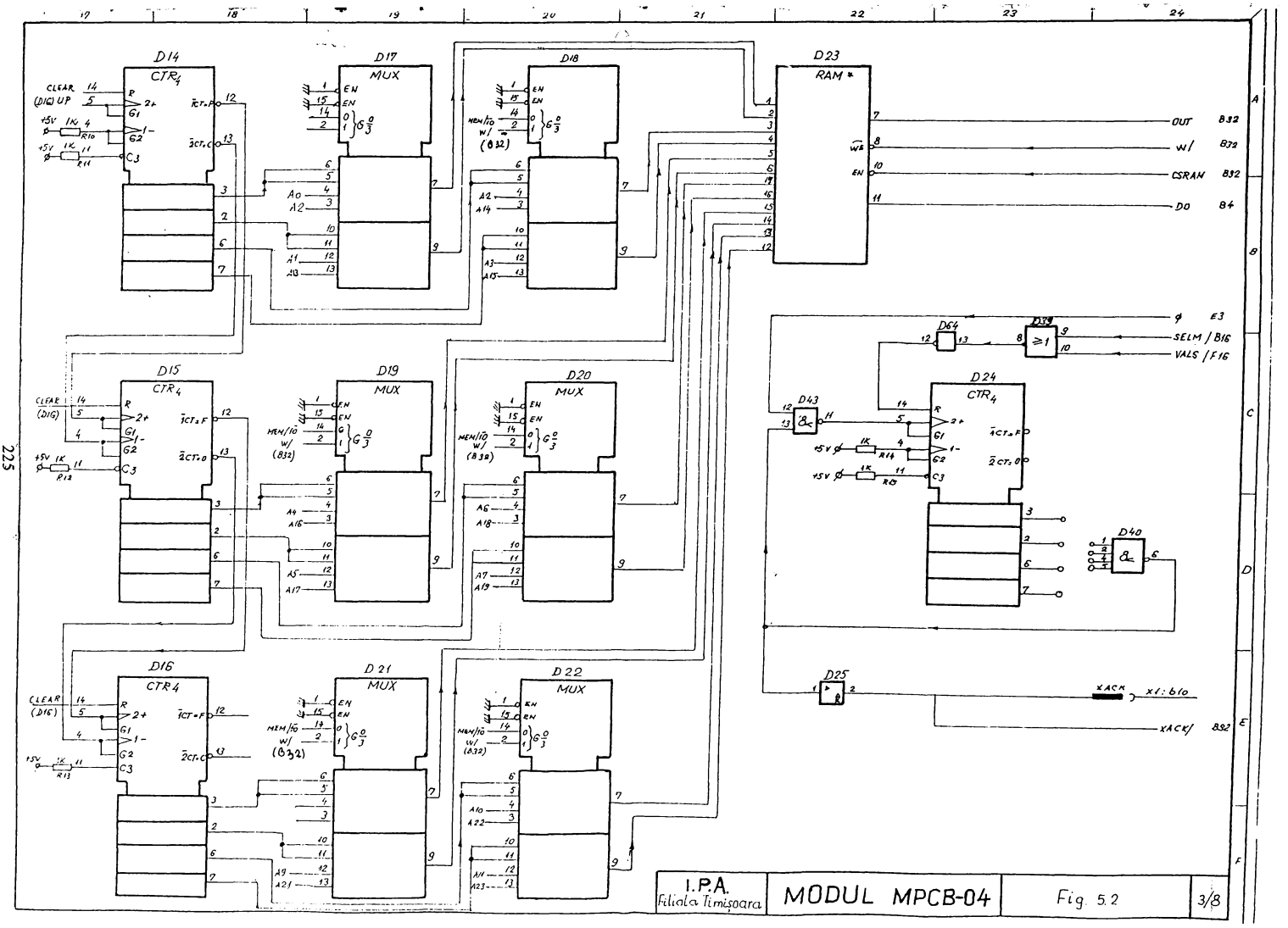
5.2.1. ASIM-ul cu reacţii programabile

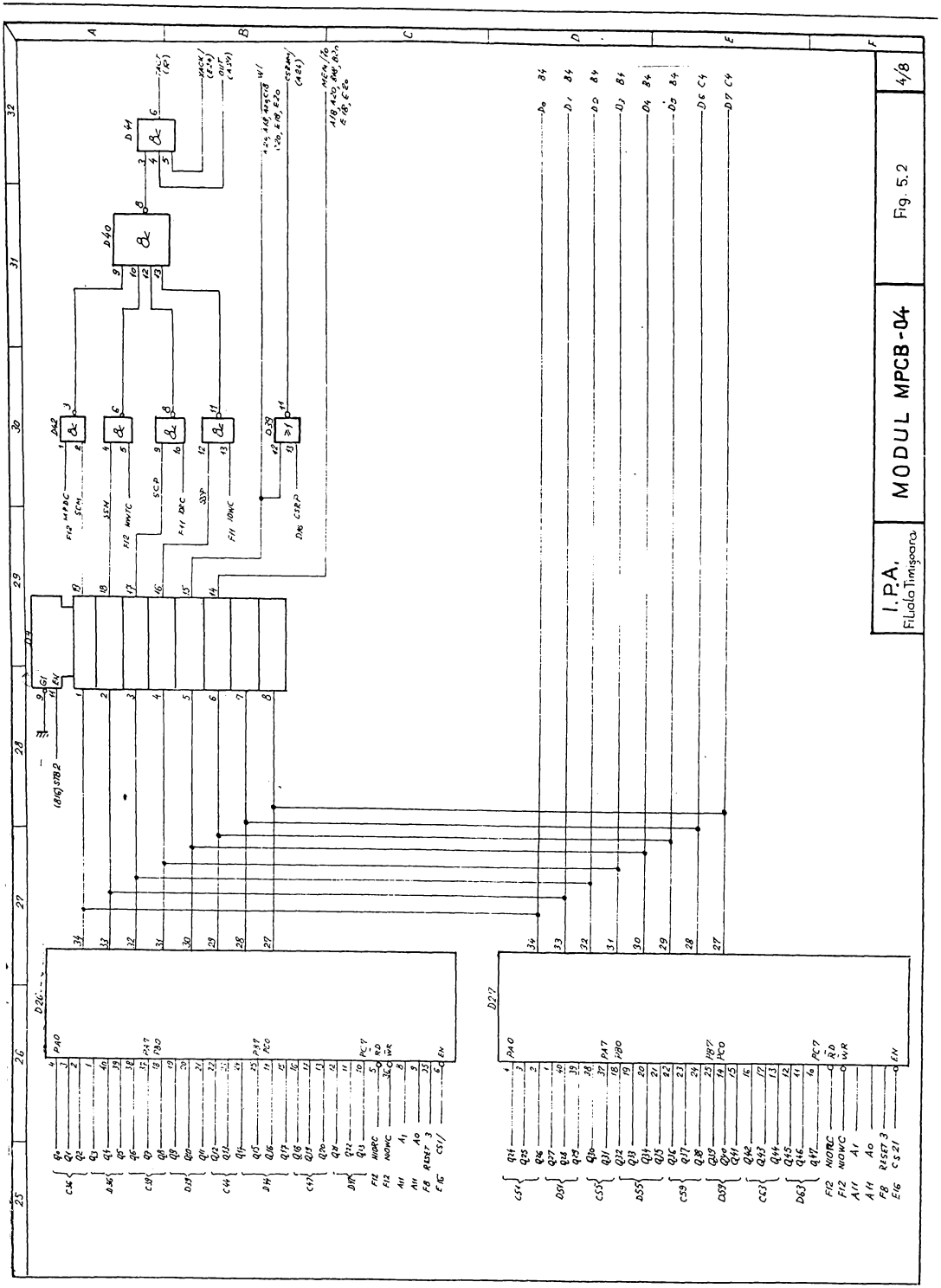
Dintre cele trei variante de implementare, în baza unor expresii date pentru polinoamele generatoare, pentru implementarea practică s-a uzitat de schema având circuite SAU EXCLUSIV intercalate între rangurile registrului de deplasare pe motivul anterior discutatei performanţe a acestora.

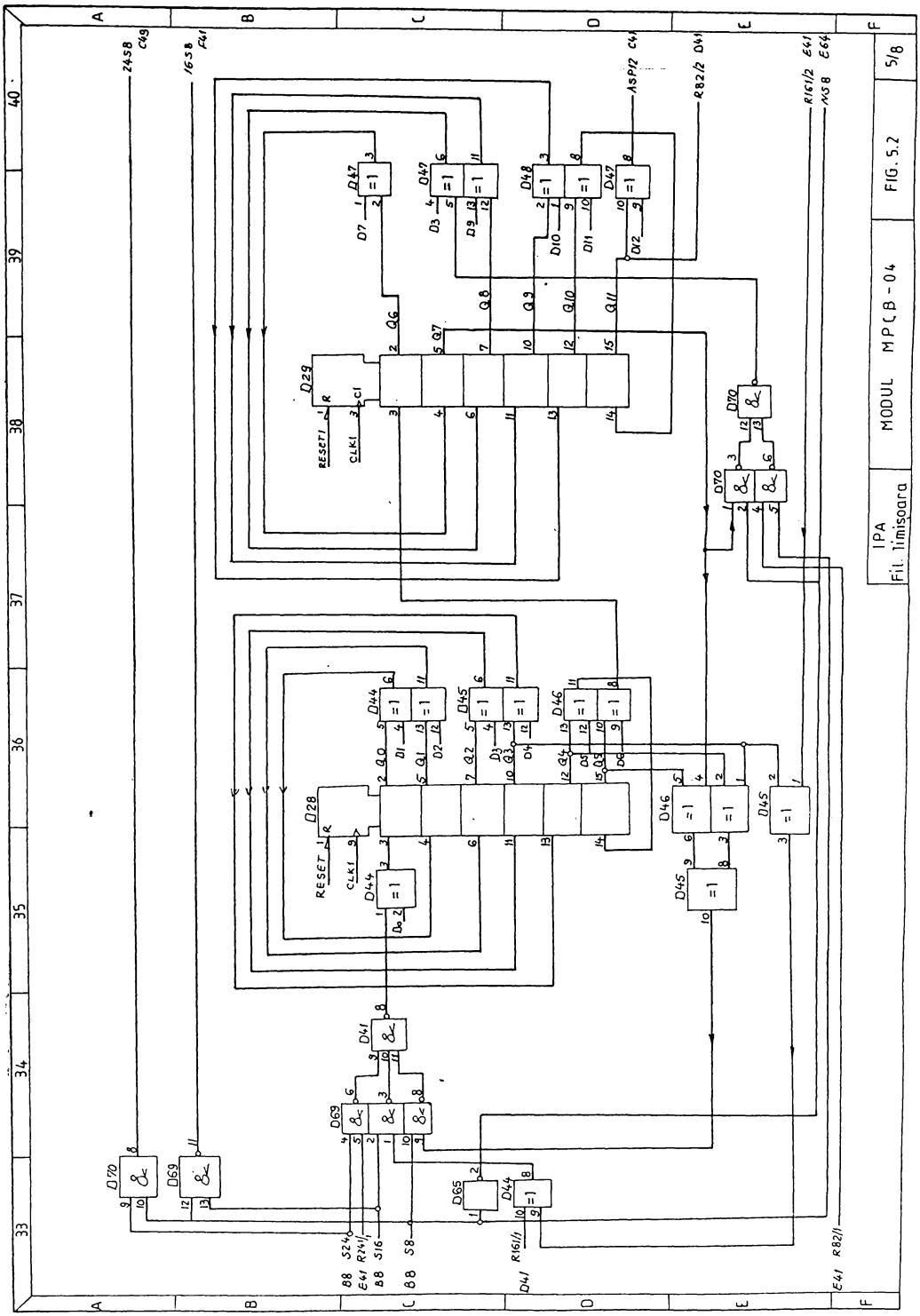
Fără a pierde din generalitatea problematicii, vom detalia cazul particular al ASIM-ului implementat pe placa fizică din familia MULTIPROM cu indicativul MPCB-04, a cărui documentaţie de execuţie este prezentată pe cele 8 file ale fig.5.2. Filele 5÷8 ale fig.5.2 cuprind ASIM-ul de 48 de biţi a cărui registru de deplasare este reprezentat prin 6 circuite integrate D28÷D35, fiecare cuprinzând 6 bistabile de tip D. Pentru schema secvenţial liniară, dintre cele 3 versiuni dezbătute în capitolul trei, s-a ales pentru implementare varianta cu circuite SAU EXCLUSIV captate în exteriorul registrului de deplasare. Logica aferentă transformării registrului de deplasare în schema secvenţială liniară având implementate polinoamele generatoare date de relaţia (5.1) a fost sintetizată prin circuite inversoare şi de tip poartă ŞI-NU cu două şi trei intrări precum şi prin circuite SAU EXCLUSIV. Nu vom insista asupra detaliilor de proiectare având în vedere faptul că modulul respectiv a fost supus întregului şir de teste funcţionale specifice familiei MULTIPROM. Mai facem însă o menţiune legat de gradul de integrare care poate fi evident îmbunătăţit prin utilizarea unor circuite integrate pe o scară mai largă, dar schema de detecţie îşi păstrează întrutotul valabilitatea.

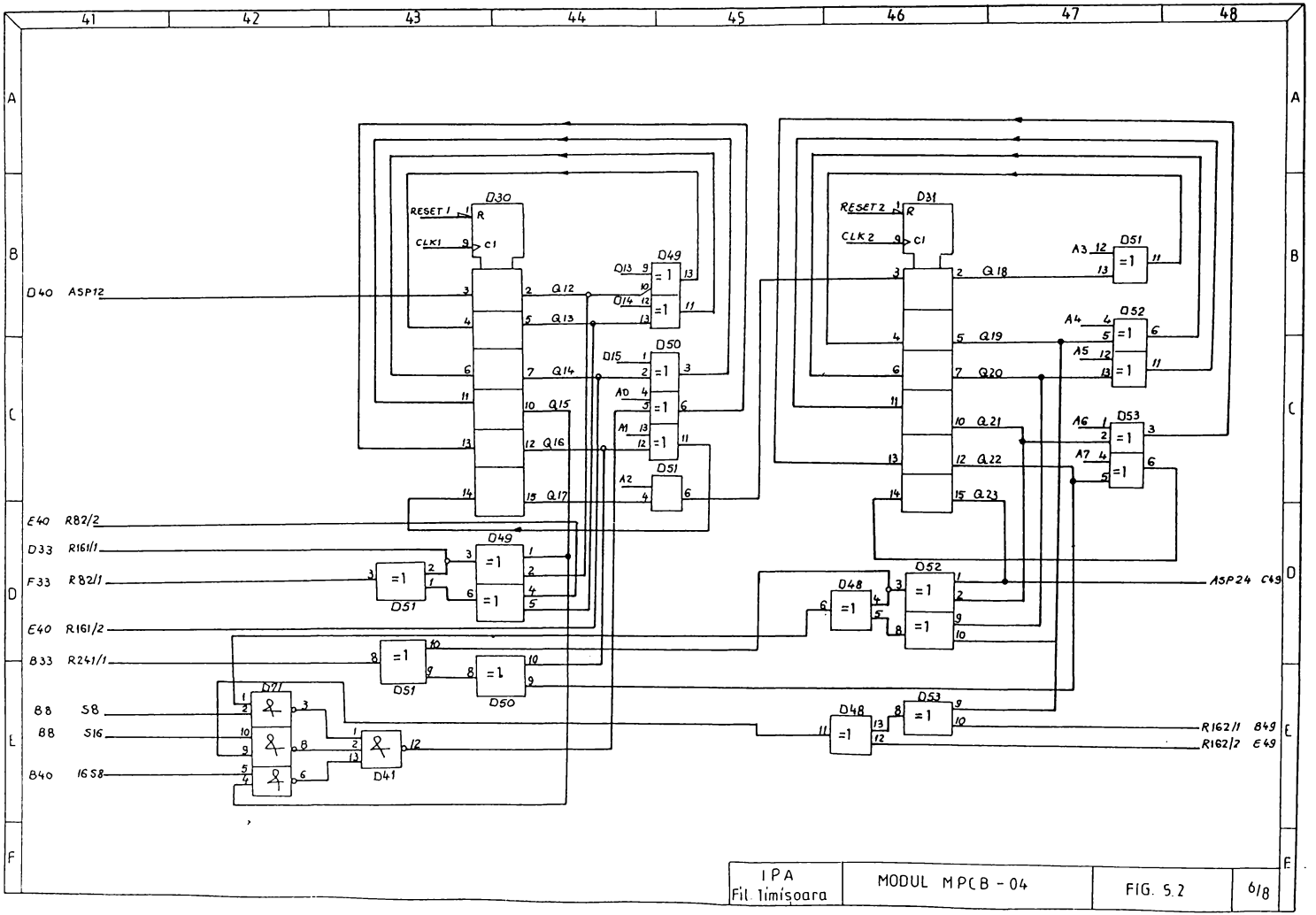



MODUL MPCB-04
Fig. 5.2
I.P.A.
Filiala Timişoara
2/8
D12
PROM
D13
PROM
D38
D39
D5
D7
D8
D9
D10
D11
STB1
STB2
C/T
SELM/
CLEAR
CS, RP
UP
CS1/
CS2/
VALS/
A4
A28
A1
B24
A17, C17, E17
B30
A17
C25
F25
C24
CS 8282-1/
CS 8282-2/
SELM/
CLEAR/
CSRAMPORT/
CS 8255-1/
CS 8255-2/
NIORC
(F25)
NIOWC
MRDC
A30
IORC (B30)
IOWC (A30)
+5V
R3
R4
R5
R6
R7
R8
R9
10K
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
A22
A23
ADR0/
ADR1/
ADR2/
ADR6/
ADR7/
ADR8/
ADRC/
ADRD/
ADRE/
ADRF/
ADR10/
ADR11/
ADR13/
IORC/
IOWC/
MRDC/
X1: a16
X1: b16
X1: b18
X1: c18
X1: a19
X1: b19
X1: c19
X1: a20
X1: c20
X1: a21
X1: b21
X1: c21
X1: a22
X1: b22
X1: c22
X1: a23
X1: c23
X1: a24
X1: b24
X1: a25
X1: c1
X1: c2
X1: a1
X1: a2

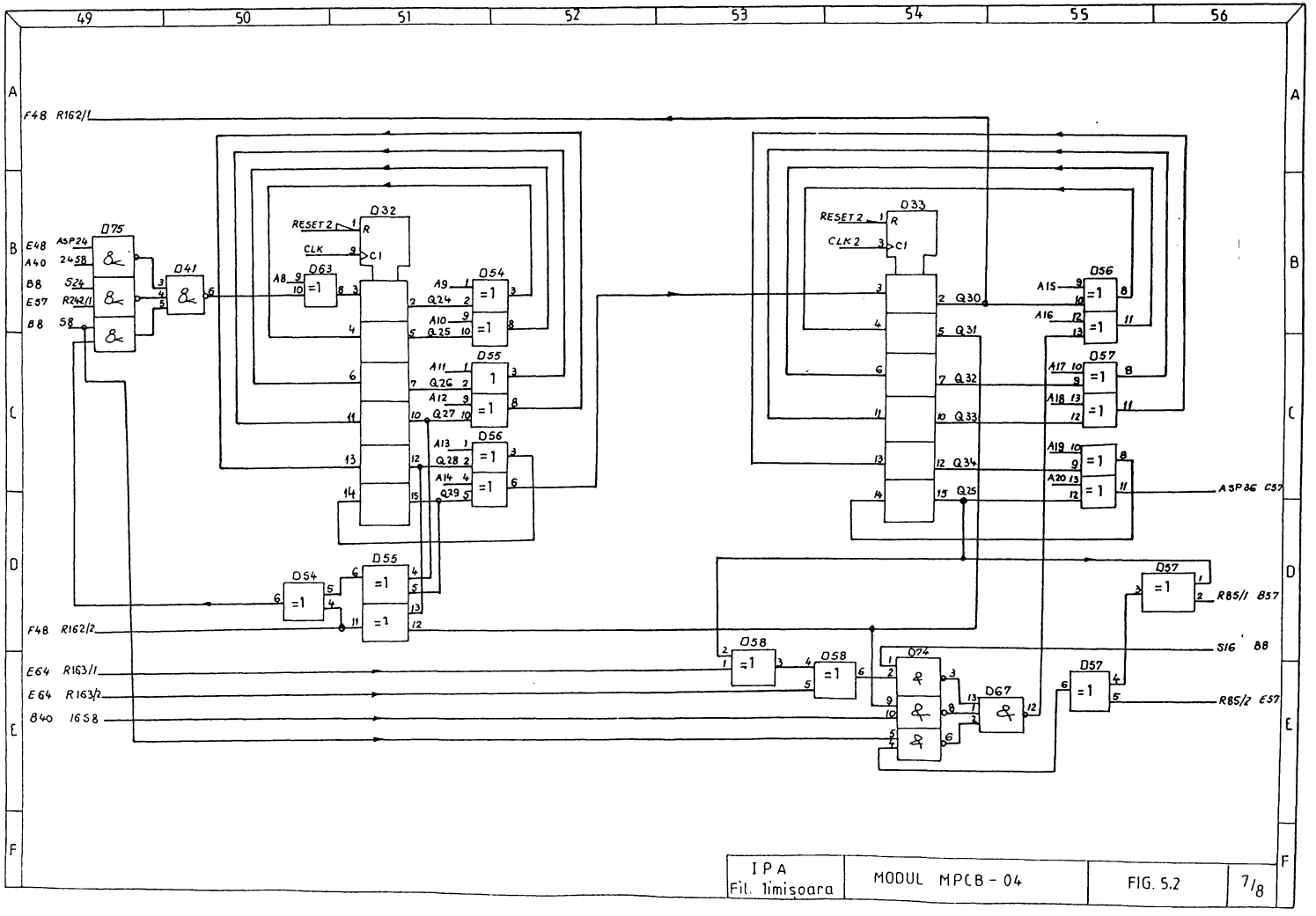

D14
CTR4
D17
MUX
D18
MUX
D23
RAM *
D15
CTR4
D19
MUX
D20
MUX
D16
CTR4
D21
MUX
D22
MUX
D24
CTR4
D43
D64
D39
D40
D25
OUT B32
W/ B32
CSRAM B32
DO B4
E3
SELM / B16
VALS / F16
XACK
XACK/ B32
I.P.A. filiala Timişoara
MODUL MPCB-04
Fig. 5.2
3/8







IPA Fil. Timişoara	MODUL MPCB - 04	FIG. 5.2	6/8


49
50
51
52
53
54
55
56
A
B
C
D
E
F
F48 R162/1
E48 ASP24
A40 2458
B8 524
E57 R242/1
B8 58
D75
D41
D32
RESET 2
CLK
C1
D63
A8
=1
D54
A9
Q24
A10
Q25
D55
A11
Q26
A12
Q27
D56
A13
Q28
A14
Q29
D54
D55
F48 R162/2
E64 R163/1
E64 R163/2
B40 1658
D33
RESET 2
CLK 2
Q30
Q31
Q32
Q33
Q34
Q25
D56
A15
A16
D57
A17
A18
A19
A20
A3P86 C57
D57
R85/1 B57
SIG 88
D58
D58
D74
D67
D57
R85/2 E57
IPA
Fil. Timişoara
MODUL MPCB - 04
FIG. 5.2
7/8

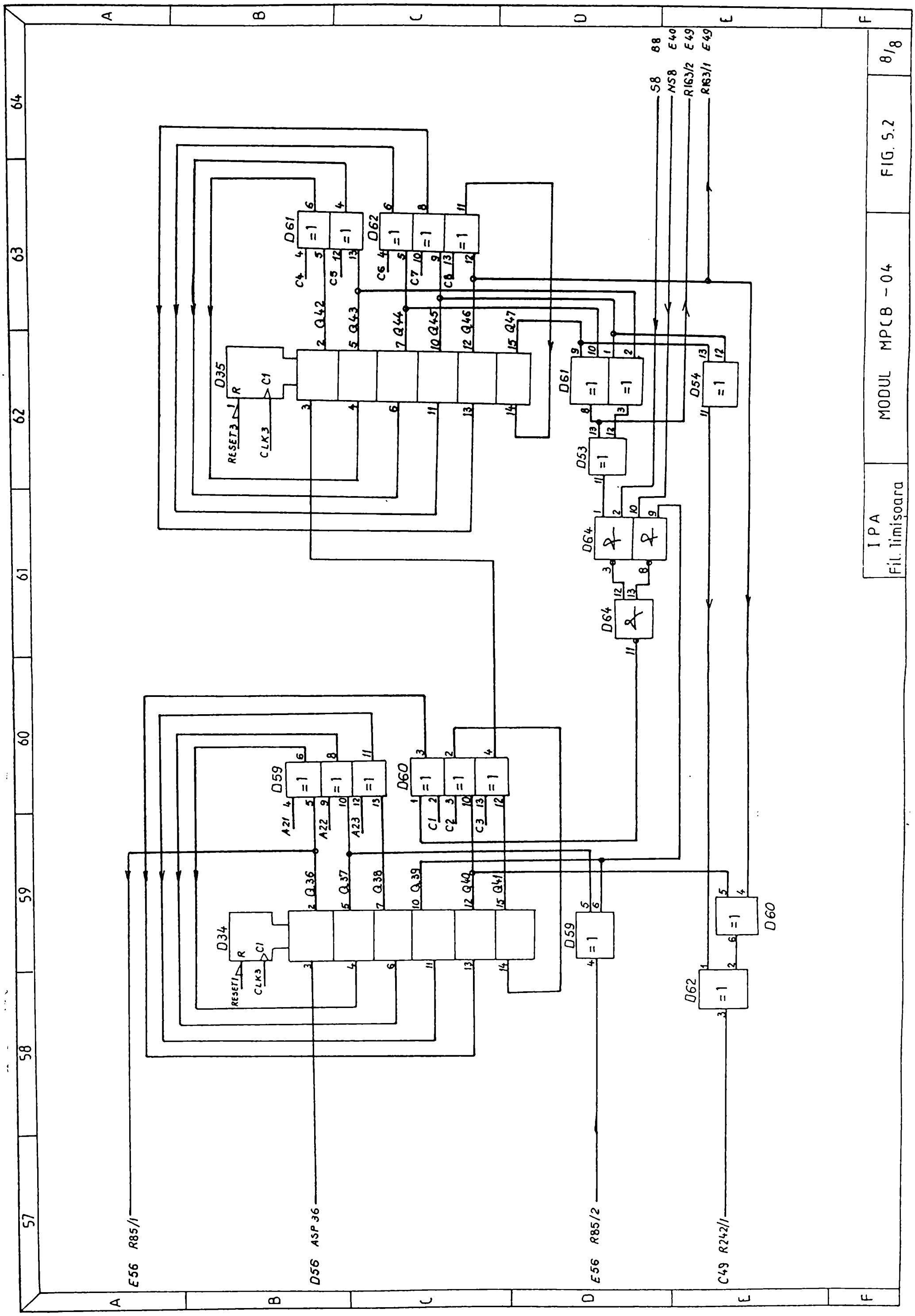


FIG. 5.2
MODUL MPCB - 04
IPA
Fil. Timisoara
8/8
D35
D34
D53
D54
D59
D60
D61
D62
D64
RESET3
CLK3
RESET1
CLK3
Q36
Q37
Q38
Q39
Q40
Q41
Q42
Q43
Q44
Q45
Q46
Q47
C1
C2
C3
C4
C5
C6
C7
C8
A21
A22
A23
58 S8
E40 NS8
E49 R163/2
E49 R163/1
E56 R85/1
D56 ASP 36
E56 R85/2
C49 R242/1

5.2.2. Blocuri anexă la ASIM-ul cu reacţii programabile

Blocul de programare al modulului are schema de detaliu prezentată în fig.5.2. fila 1. Esenţiale sunt semnalele S24, S16 şi S8 furnizate la ieşirile circuitului D3 pentru programarea comprimării pe 24, 16 respectiv 8 biţi.Selecţia fronturilor pentru START, STOP şi TACT sunt furnizate prin semnalele de la pinii 15,14, respectiv 16 ale aceluiaşi circuit D3. Pentru aceste semnale există şi alternativa comandării prin intermediul unor strap-uri prevăzute pentru facila depanare a modulului însuşi. Aceeaşi remarcă este valabilă şi în ceea ce priveşte semnalul de iniţializare RES (pin 15 al circuitului D3)..Încărcarea configuraţiilor de programare se face de pe liniile de date D0÷DF prin intermediul a două circuite 8282.

Blocul de selecţie (BS) este sintetizat prin circuite cuprinse în fig.5.2 fila 2. Acest bloc furnizează semnalele de activare a circuitelor 8282 destinate programării modului de lucru a numărătorului care realizează translatarea adreselor pentru memoria RAM, a circuitelor 8255 prin intermediul cărora se realizează predarea semnăturilor pe magistrală şi a memoriei RAM care generează semnalul de selecţie pentru modulùl testat.

Cuplarea la magistrală se realizează prin circuite 8255 a căror citire succesivă corespunzătoare portuirlor acestora asigură obţinerea semnăturii generată prin ASIM-ul cu reacţie programabilă.

5.3.Verificarea modulului MPCB-04

Prezentul paragraf a fost intercalat pentru a sublinia necesitatea autotestării modulului MPCB-04 care de fapt asigură controlul încorporat al configuraţiei de calcul. Principial,autocontrolul se realizează în conformitate cu organigramele din figurile 5.3 respectiv 5.4. Acestea au ataşate programele de test elaborate, fără a pierde din generalitate, în limbaje de asamblare microprocesoarelor pe 8 biţi.

5.4.Modificări ale modulului MPCB-04 în vederea aplicării noilor structuri ASIM

Modulul MPCB-04 a cărui structură a fost descrisă sumar anterior într-o implementare specifică tehnologiei electronice autohtone disponibilă la nivelul anilor 87 [Vasi-87] a conţinut ideea , la timpul respectiv, inovativă, a amplasării la nivelul magistralei a unor dispozitive hardware care în mod eficace să permită evaluarea fluxurilor informaţionale de pe liniile magistralei prin comprimare în semnături. Ulterior, referinţe de marcă [ReSK-88,MaCl-88,KaNa-90,KaLV-94] au dezvoltat conceptul amplasării unui modul slave de tipul celui descris, la magistrală într-un procesor watchdog.

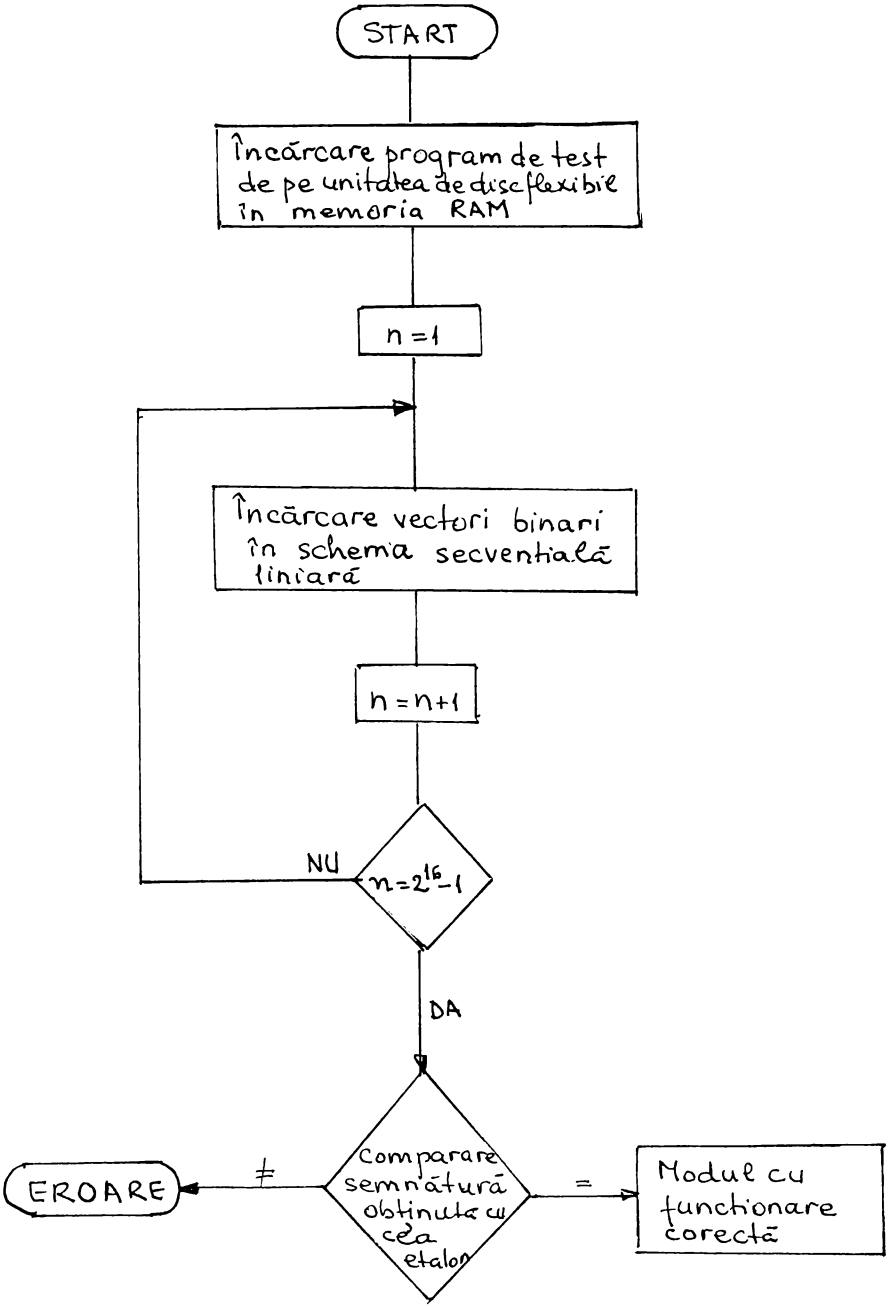

START
Încărcare program de test de pe unitatea de disc flexibil în memoria RAM
n = 1
Încărcare vectori binari în schema secvențială liniară
n = n+1
NU
n = 2^16 - 1
DA
Comparare semnătură obținută cu cea etalon
≠
EROARE
=
Modul cu funcționare corectă

Fig.5.3

```
                                ;                                MPCB-04
                                ;
0000'                                   CSEG
0000                            S821    EQU     0
0000                            S822    EQU     0
0000                            CLR     EQU     0
0000                            AMD     EQU     0
0000                            CSRP    EQU     0
0000                            ASEMN   EQU     0
0000' 45 52 4F 41               EROARE: DC      'EROARE'
0004' 52 C5
0006' 21 0000'                  ERR:    LXI     HL,EROARE
0009' 06 36                             MVI     B,'6'
000B' 4E                        ERR1:   MOV     C,M
000C' CD FD0A                           CALL    0FD0AH
000F' 23                                INX     H
0010' 05                                DCR     B
0011' C2 000B'                          JNZ     ERR1
0014' C3 007E'                          JMP     TERM
0017' 3E 00                     START:  MVI     A,0       ;sterge registrele 8282 si numaratoarele
0019' D3 00                             OUT     S821
001B' D3 00                             OUT     S822
001D' D3 00                             OUT     CLR
001F' 06 00                             MVI     B,AMD     ;incarca adresa modulului testat in reg.B
0021' 21 0000                           LXI     H,0
0024' 11 0010                           LXI     D,0010H
0027' 78                        ET3:    MOV     A,B       ;selectie modul testat
0028' BC                                CMP     H
0029' C2 0032'                          JNZ     ET1
002C' AF                                XRA     A
002D' D3 00                             OUT     CSRP
002F' C3 0036'                          JMP     ET2
0032' 3E FF                     ET1:    MVI     A,0FFH
0034' D3 00                             OUT     CSRP
0036' 19                        ET2:    DAD     D
0037' 7C                                MOV     A,H
0038' FE 00                             CPI     0
003A' C2 0027'                          JNZ     ET3
003D' 3E 30                             MVI     A,30H     ;pregatesle registrele 8282 pentru inceperea testului
003F' D3 00                             OUT     S822
0041' 3E 41                             MVI     A,41H
0043' D3 00                             OUT     S821
0045' 21 0000                           LXI     HL,C

0048' 3E 31                             MVI     A,31H
004A' D3 00                             OUT     S822
004C' 7E                        ET4:    MOV     A,M
004D' 23                                INX     H
004E' 7E                                MOV     A,M
004F' B5                                ORA     L
0050' C2 004C'                          JNZ     ET4
0053' 3E 30                             MVI     A,30H
0055' D3 00                             OUT     S822
0057' 21 0000                           LXI     H,ASEMN   ;citeste semnatura din portul A si o compara
005A' DB F8                             IN      0F8H
005C' BE                                CMP     M
005D' C2 0006'                          JNZ     ERR
0060' DB F5                             IN      0F5H      ;citeste semnatura din portul B si o compara
0062' BE                                CMP     M
0063' C2 0006'                          JNZ     ERR
0066' DB F6                             IN      0F6H      ;citeste semnatura din portul C si o compara
0068' BE                                CMP     M
0069' C2 0006'                          JNZ     ERR
006C' DB F8                             IN      0F8H      ;citeste semnatura din portul A si o compara
006E' BE                                CMP     M
006F' C2 0006'                          JNZ     ERR
0072' DB F9                             IN      0F9H      ;citeste semnatura din portul B si o compara
0074' BE                                CMP     M
0075' C2 0006'                          JNZ     ERR
0078' DB FA                             IN      0FAH      ;citeste semnatura din portul C si o compara
007A' BE                                CMP     M
007B' C2 0006'                          JNZ     ERR       ;modulul testat defect
007E' 00                        TERM:   NOP
                                        END     START

Macros:

Symbols:
AMD     0000    ASEMN   0000    CLR     0000    CSRP    0000
EROARE  0000'   ERR     0006'   ERR1    000B'   ET1     0032'
ET2     0036'   ET3     0027'   ET4     004C'   S821    0000
S822    0000    START   0017'   TERM    007E'
```

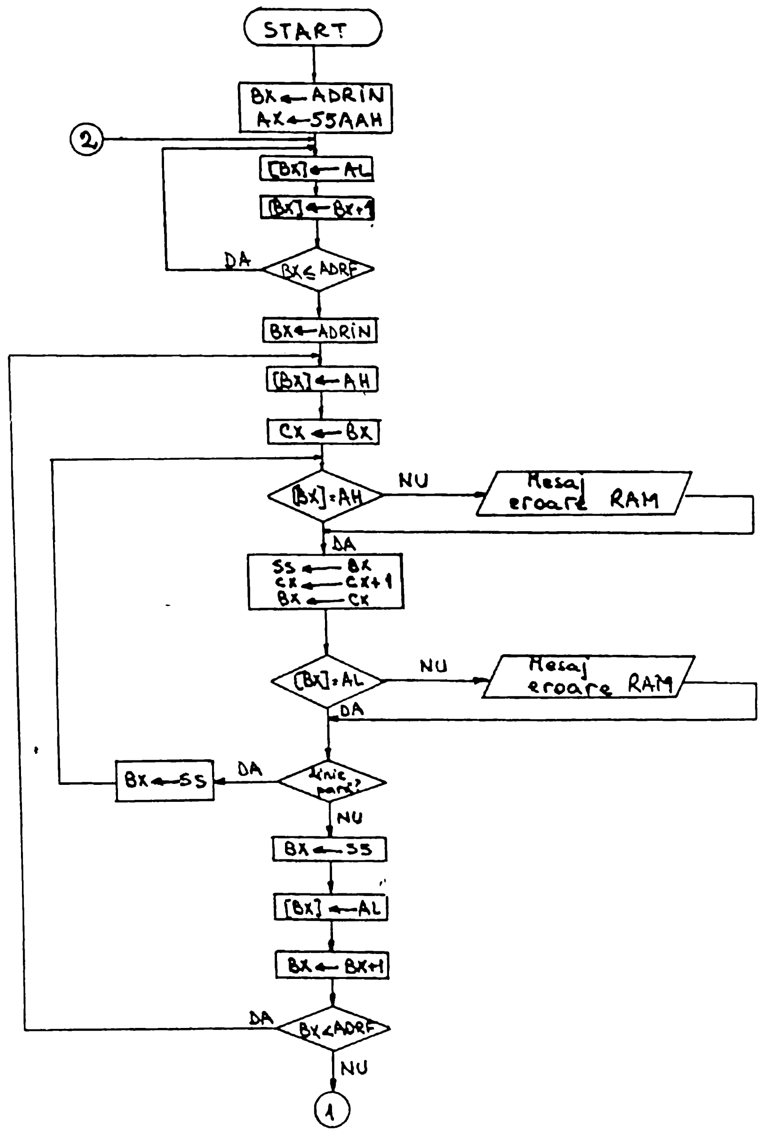

START
BX ← ADRIN
AX ← 55AAH
2
[BX] ← AL
[BX] ← BX+1
DA
BX ≤ ADRF
BX ← ADRIN
[BX] ← AH
CX ← BX
NU
[BX]=AH
Mesaj
eroare RAM
DA
SS ← BX
CX ← CX+1
BX ← CX
NU
[BX]=AL
Mesaj
eroare RAM
DA
BX ← SS
DA
NU
BX ← SS
[BX] ← AL
BX ← BX+1
DA
BX<ADRF
NU
1

Fig. 5.4

```
                        ;
                        ;
                        ; PROGRAM DE TEST AL MODULULUI ANALIZOR DE SEMNATURI
                        ;
                        ;
                              .Z80
                              CSEG
                        S821  EQU   0             ;adresa port 1 8282(stabileste
                                                  ; tip polinom)
                        S822  EQU   0             ;adresa port 2 8282(stabileste
                                                  ; ciclurile
                                                  ; pentru care se formeaza semnatura)
                        CLR   EQU   0             ;port stergere bitabile
                        RC551 EQU   0             ;adresa registru comanda
                                                  ; circuit 8255)
                        CSRP  EQU   0             ;chip select RAM port
                        A551  EQU   0             ;adresele porturilor A,B,C
                                                  ;ale circuitelor 8255
                        B551  EQU   0
                        C551  EQU   0
                        A552  EQU   0
                        B552  EQU   0
                        C552  EQU   0
00 00 00 00             MEM:  DB    0,0,0,0,0,0   ;semnatura corecta
00 00
3E 00                   START: LD   A,0
01 0000                        LD   BC,S821
ED 79                          OUT  (C),A
01 0000                        LD   BC,S822
ED 79                          OUT  (C),A
01 0000                        LD   BC,CLR
ED 79                          OUT  (C),A        ;s-au sters registrele 8282 si
                                                 ; numaratoarele
11 0000                        LD   DE,RC551     ;se incepe inscrierea in memorie
                                                 ; (adresa pe 12 biti )
21 0000                        LD   HL,0         ; 0 pentru HL=DE
                                                 ; FFH pentru HL<>DE
01 0000                        LD   BC,CSRP
7B                      BC1:   LD   A,E
BD                             CP   L
C2 002F'                       JP   NZ,BC2       ;HL<>DE, salt
7A                             LD   A,D
BC                             CP   H
C2 002F'                       JP   NZ,BC2       ;HL<>DE, salt
3E 00                          LD   A,0          ;HL=DE, se inscrie 0
C3 0031'                       JP   BC3
```

```
002F'  3E FF        BC2:  LD    A,0FFH      ;MODE, se inscrie OFFH
0031'  ED 79        BC3:  OUT   (C),A
0033'  23                 INC   HL          ;adresa urmatoare
0034'  7D                 LD    A,L
0035'  FE 00              CP    0
0037'  C2 0020'           JP    NZ,BC2      ;nu s-a ajuns la sfirsit,se reia
003A'  7C                 LD    A,H
003B'  FE 10              CP    10H         ;s-au inscris primii 2048 oct ?
003D'  C2 0020'           JP    NZ,BC1      ;nu, se reia
0040'  3E 41              LD    A,41H       ;da, se pregatesc registre 8282
0042'  01 0000            LD    BC,S82I     ;pentru incepere test
0045'  ED 79              OUT   (C),A       ;initializarea schemei secventiale
0047'  3E 08              LD    A,08H
0049'  01 0000            LD    BC,S822
004C'  ED 79              OUT   (C),A       ;test pentru IOWC
004E'  3E 10              LD    A,10H
0050'  01 0000            LD    BC,S821
0053'  ED 79              OUT   (C),A       ;START=1
0055'  3E 09              LD    A,09H
0057'  ED 79              OUT   (C),A       ;PER=1;T=1;S24=1
0059'  16 64              LD    D,100       ;se va inscrie semnatura pentru IOO
                                            ; de inscrieri in registrul de ada
                                            ; al circ i 8255
005B'  3E 90              LD    A,90H
005D'  01 0000            LD    BC,RC55I
0060'  ED 79        REIA: OUT   (C),A       ;8255 - mod iesire
0062'  15                 DEC   D
0063'  C2 0060'           JP    NZ,REIA
0066'  DD 21 0000'        LD    IX,MEM      ;incepe verificarea semnaturii obtinute
006A'  01 0000            LD    BC,A55I
006D'  ED 78              IN    A,(C)
006F'  DD BE 00           CP    (IX)        ;verificare semnatura port A 8255 - 1
0072'  C2 00AD'           JP    NZ,ER1
0075'  01 0000            LD    BC,B55I
0078'  ED 78              IN    A,(C)
007A'  DD BE 01           CP    (IX+1)      ;verificare semnatura port B 8255 - 1
007D'  C2 00AE'           JP    NZ,ER2
0080'  01 0000            LD    BC,C55I
0083'  ED 78              IN    A,(C)
0085'  DD BE 02           CP    (IX+2)      ;verificare semnatura port C 8255 - 1
0088'  C2 00AF'           JP    NZ,ER3
008B'  01 0000            LD    BC,A552
008E'  ED 78              IN    A,(C)
0090'  DD BE 03           CP    (IX+3)      ;verificare semnatura port A 8255 - 2
0093'  C2 00B0'           JP    NZ,ER4
0096'  01 0000            LD    BC,B552
```

```
0099'  ED 78                    IN      A,(C)
009B'  DD BE 04                 CP      (IX+4)          ;verificare semnatura porti B 8255 - 2
009E'  C2 00B1'                 JP      NZ,ER5
00A1'  01 0000                  LD      BC,ASS1
00A4'  ED 78                    IN      A,(C)
00A6'  DD BE 05                 CP      (IX+5)          ;verificare semnatura port C 8255 - 2
00A9'  C2 00B2'                 JP      NZ,ER6
00AC'  00               CORECT: NOP                     ;semnatura obtinuta e corecta
00AD'  00               ER1:    NOP                     ;tratare eroare semnatura port A 8255 - 1
00AE'  00               ER2:    NOP                     ;tratare eroare semnatura port B 8255 - 1
00AF'  00               ER3:    NOP                     ;tratare eroare semnatura port C 8255 - 1
00B0'  00               ER4:    NOP                     ;tratare eroare semnatura port A 8255 - 2
00B1'  00               ER5:    NOP                     ;tratare eroare semnatura port B 8255 - 2
00B2'  00               ER6:    NOP                     ;tratare eroare semnatura port C 8255 - 2
                                END     START

Macros:

Symbols:
ASS1    0000    ASS2    0000    BSS1    0000    BSS2    0000
BC1     0020'   BC2     002F'   BC3     0031'   CSS1    0000
CSS2    0000    CLR     0000    CORECT  00AC'   DSRP    0080
ER1     00AD'   ER2     00AE'   ER3     00AF'   ER4     00B0'
ER5     00B1'   ER6     00B2'   MEM     0000'   RCSS1   0000
RETA    0060'   SS21    0000    SS22    0000    START   0006'

No  Fatal error(s)
```

Remarcăm, pe lângă ideea de urmărire printr-un analizor de semnături cu intrări multiple a lanțurilor binare de pe liniile de magistrală, a eficientei idei originale de reconfigurabilitate prin intermediul mai multor expresii de polinoame generatoare. La aceste aspecte originale adăugăm acum în baza investigațiilor de natură teoretică efectuate cu scopul creşterii capacității de detecție a modulului descris, modificări rezultate în urma cercetării efectuate. Aceste modificări sunt prezentate în fig.5.5, care vizează renunțarea la unele conexiuni şi substituirea lor cu altele în conformitate cu notația următoare:

Nr. crt.	Se renunță la conexiunile	Se substituie prin conexiunile	Modificare pe fig. 5.2 fila	Afectează pe:
1.	D28/10-D45/2-D46/1	D45/2-D45/11-D46/1	5/8	81/1,161/1
2.	D28/12-D46/1	D46/1-D46/11	5/8	81/2
3.	D28/13-D46/5	D46/5-D46/8	5/8	81/3
4.	D30/2-D49/2	D49/2-D49/13	6/8	82/1,161/2
5.	D30/5-D49/5	D49/5-D49/11	6/8	82/2
6.	D29/15-D49/4	D47/8-D49/4	5/8,6/8	82/3
7.	D30/7-D45/2	D45/2-D50/3	5/8,6/8	161/3
8.	D31/10-D52/2	D52/2-D53/3	6/8	83/1,241/1
9.	D31/7-D52/9	D52/9-D52/11	6/8	83/2

10.	D31/5-D52/5-D53/9	D52/5-D52/10-D53/9	6/8	83/3,162/1
11.	D31/12-D50/9	D50/9-D53/6	6/8	241/2
12.	D30/12-D50/10	D50/10.D50/11	6/8	241/3
13.	D33/2-D53/10	D53/10-D56/8	6/8,7/8	162/21
14.	D32/12-D55/13	D55/13-D56/3	7/8	84/1,162/3
15.	D32/10-D55/4	D55/4-D55/8	7/8	84/2
16.	D32/15-D55/5	D55/5-D56/6	7/8	84/3
17.	D34/5-D59/5	D59/5-D59/8	8/8	85/1
18.	D34/2-D57/2	D57/2-D59/6	7/8,8/8	85/2
19.	D33/5-D57/1-D58/2	D57/1-D58/2-D58/11	7/8	85/3,163/1
20.	D35/5-D61/2	D61/2-D61/11	8/8	86/1
21.	D35/7-D61/10	D61/10-D62/6	8/8	86/2,163/2
22.	D35/10-D54/42-D61/1	D54/12-D61/1-D62/8	8/8	86/3,242/1
23.	D34/12-D60/5	D60/5-D60/8	8/8	2422
24.	D35/12-D58/1-D60/4	D58/1-D60/4-D62/11	7/8,8/8	163/3,242/3

Fig.5.5

D28/10-D45/2-D46/1 semnifică conexiunea: capsula D28 pinul 10 legat cu capsula D45 pinul2, etc.Ultima coloană a tabelului din fig. 5.5, cuprinde notaţii de tipul 81/1 sau 162/2 sau 241/3, unde după numerele 8,16 şi 24 care le succede ale registrului de deplasare având ataşate circuitele SAU EXCLUSIV în conformitate cu expresiile polinoamelor generatoare din relaţia (5.1) de gradul 8, 16 sau 24. După bară sunt marcate cifrele 1,2 sau 3 semnificând numărul modificării termenelor exceptând pe cei extremi din cadrul expresiilor polinoamelor generatoare care în conformitate cu relaţia (5.1) cuprind toate câte cinci termeni (5-2 termeni extremi =3). Aceeaşi ultimă coloană a tabelului mai pune în relief prin acele linii care cuprind mai mult o notaţie de tipul specificat acele circuite SAU EXCLUSIV utilizate în comun de două polinoame şi astfel economisite reprezentând o metodă ad-hoc de minimizare.

5.5.Concluzii

Urmărind valorificarea practică a cercetărilor de natură teoretică, autoarea pleacă de la o mai veche realizare reprezentată de modulul MPCB-04 conectat la magistrala MULTIPROM. Subliniem din nou, faptul că tipul magistralei precum şi nivelul de integrare al tehnologiei de realizare este neesenţial, modulul MPCB-04 însă conţine următoarele elemente originale:

a) Preluarea în domeniul tehnicii testării a ideii de analiză de semnături de tip serial şi adaptarea ei ca modul distinct destinat autocontrolului în varianta de implementare cu captare paralelă la nivelul magistralei.

b) Plecând de la un număr impus de semnale de urmărit (în cazul nostru 48 semnale, dar în general această valoare este nerelevantă) introducerea reconfigurabilităţii structurii ASIM-ului dependent de gradul şi expresia polinomului generator. Devine în acest mod posibil ca anumite erori posibil a nu fi puse în evidenţă prin intermediul analizorului sintetizat pe baza unei anumite expresii de polinom generator să poată fi detectate prin intermediul analizorului de semnături reconfigurat prin intermediul altei expresii.

c) Plecând de la expresii date de polinoame generatoare primitive şi ireductibile a cărei alegere este justificată în mod amplu în partea teoretică, sinteza în manieră originală astfel încât la performanţa maximă a procesului de comprimare conferită prin expresiile polinoamelor generatoare să se adauge şi optimul costului prin găsirea numărului minim de circuite SAU EXCLUSIV.

La aceste aspecte menţionate, valorificate în modulul MPCB-04, în baza cercetării capacităţii de detecţie efectuată în capitolul 3 au mai rezultat următorul aspect:

d) Modificarea conexiunilor la circuitele SAU EXCLUSIV conectate în exteriorul registrului de deplasare astfel încât să rezulte o valoare minimă pentru probabilitatea statică şi reală de recunoaştere ca funcțional corectă a unei unități testate corecte. În acest sens, este elaborat tabelul de modificări care se impune grafat pe o realizare de tipul celei exemplu constituită de modulul hardware MPCB-04 din familia MULTIPROM.

Desigur, în final, se cuvine menționat întreaga activitate de sinteză a unui modul adus în faza de execuție implicând toate fazele de testare şi respectiv autotestare necesare unei astfel de activități.

6. Aplicarea analizei de semnături la un sistem dual sincron cu toleranță la defectare destinat conducerii proceselor industriale

6.1.Stabilirea configurației sistemului dual sincron

Menționăm din debut că în prezentul capitol, ca şi în anteriorul, de altfel referirile se vor face, fără a pierde din generalitate la realizări configurate pe magistrala standard MULTIPROM. Urmărind aceeaşi linie de valorificare a investigațiilor teoretice întreprinse, dacă în anteriorul capitol a fost acoperită prima treaptă, cea a unui modul hardware (placa), în prezentul capitol încercăm valorificarea respectivului modul modificat în cadrul unui sistem. Astfel, ne propunem configurarea justificată prin căutările teoretice din capitolele 1 şi 2, a unui sistem tolerant la defectare bazat pe principiul, favorabil din punct de vedere al costului, aplicării dublării active.

Să considerăm [Vasi-88c,f,g] o configurație de calcul minimală destinată conducerii unui proces industrial. În termeni MULTIPROM aceasta implică prezența în primul rând a unui modul procesor pe care în particular îl admitem a fi MPCM-03. [Vasi-87c,d]. Într-o descriere foarte sumară acesta permite execuția programelor aflate în memoria propie sau în alte module de memorie cuplate la magistralele sistemului pentru întregul set de instrucțiuni al procesorului 8086, generarea a două baze de timp programabile, lansarea unor activități prioritare, cuplarea la un dispozitiv periferic serial, sesizarea blocării accesului la magistrale, generarea a patru semnale de întrerupere cuplabile la ambele magistrale prin strap-uri, tratarea a opt niveluri de întrerupere mascabile cuplate prin strap-uri la magistralele de sistem, rezidentă sau alte dispozitive precum şi detectare blocării procesului pe un timp mai lung de 2ms datorat unei adresări greşite sau defectării unor module. Capacitatea maximă de memorie adresabilă este de 1 Moctet iar capacitatea maximă de memorie RAM propie este de 4 kocteți şi, în fine, capacitatea maximă de memorie EPROM este de 16 kocteți respectiv 32 kocteți sau 64 kocteți şi dependent de capsula integrată utilizată care poate fi 2716, respectiv 2732 sau 2764 [Vasi-88].

Completând configurația de calcul minimală menționată cu un modul de memorie, alegem în acest sens MPMM-04 [Vasi-87c,d] cuprinzând o memorie RAM CMOS de 8 kocteți cu posibilitate de salvare pe baterie, modul conectat la magistrala rezidentă cu o capacitate a liniilor de date de 8 sau 16 biți. Dependent de tipul circuitelor integrate utilizate (2716, respectiv 2732 sau 2764) modulul mai conține o memorie EPROM cu capacitate maximă de 8 kocteți., respectiv 16 kocteți sau 32 kocteți. Modulul MPMM-04 mai conține două interfețe seriale care funcționează în modul buclă de tensiune şi conține câte un circuit 8251.

Pe lângă aceste două module esențiale, admitem că în configurația minimală mai apar MPSA-03 constituind un modul placă prelungitoare, MPSA-02 constituind sursa de alimentare [Vasi-88d,c]. Cu aceste module echipate cu sertar tipic MULTIPROM poate fi asigurată conducerea automată a unui proces industrial [Vasi.88k]. Se pune însă problema apariției unei stări de defectare în cadrul echipamentului de calcul şi a faptului că procesul industrial condus nu admite decât o

întrerupere de durată scurtă care să nu poată în general acoperi depanarea defectului din configurația de calcul. Într-o astfel de situație ar putea fi aplicată soluția, dublării pasive cu o configurație de calcul identică în stare de aşteptare rece (neconectată la sursa de alimentare) sau caldă (conectată la sursa de alimentare, dar neâncărcată cu sarcinile de calcul). O astfel de soluție tehnică, la apariția unei stări de defectare, permite o foarte dificilă localizare a defectului între echipamentul de calcul şi proces, în cele mai multe cazuri fiind suspectată interfața cu procesul şi doar în ultima instanță echipamentul de calcul. Evident prin mijloace de investigare, cu precădere software, deci lente, se poate face delimitare între prezența stării de defect la interfața cu procesul, respectiv la echipamentul de calcul. Aceasta implică un timp de investigare care nu poate fi asigurat de anumite aplicații critice prin prisma timpului de reacție al procesului industrial condus. Pentru aceste situații proxima soluție tehnică urcând scara complexității, respectiv costului, este reprezentată de dublarea activă. Evident, atunci când prin prisma aceluiaşi parametru critic constituit de timpul de răspuns al procesului,soluția bazată pe dublarea activă nu corespunde, se apelează la soluții bazate pe redundanță din ce în ce mai mare cum ar fi triplarea, quadruplarea, etc. Prin prisma obiectivelor propuse pentru această teză, ne vom restrânge considerațiile în cele ce urmează la conducerea automată a proceselor industriale care prin prisma timpului de reacție al răspunsului tolerează soluția bazată pe dublarea activă. Aceasta implică, cu referire la configurația de calcul descrisă anterior existența a două configurații de calcul identice care funcționează în paralel, legătura cu procesul fiind asigurată prin intermediul unui bloc de comutare.(fig.6.1).

În ceea ce priveşte interfața cu procesul aceasta constă atât din intrări cât şi din ieşiri. Referitor la intrări acestea sunt aplicate ambelor configurații de calcul admise prin două sertare separate alcătuite din module MULTIPROM anterior specificate. Captând aceleaşi date din proces şi prelucrându-le în baza aceloraşi proceduri algoritmice, cele două echipamente vor funcționa în sincronism. Referitor la ieşiri, prin intermediul aceluiaşi bloc de comutare doar unul dintre calculatoarele care furnizează rezultate identice va asigura comanda procesului. Pentru a asigura funcționarea în sincronism a celor două echipamente se impune o soluție pentru problema distinctă şi importantă a sincronizării,variatelor soluții software cunoscute [Vasi-88] fiindu-le preferată din motive de performanță, o soluție hardware. Aceasta se bazează pe un generator comun pentru trenurile de tact corespunzătoare celor două echipamente. Evident,acest generator trebuie să acopere situația defectării lui însuşi pentru că într-un astfel de caz ar fi compromisă întreaga aplicație. Prin urmare, el trebuie conceput adoptând la sinteza sa una din metodele de eficiență sporită care să asigure funcționarea toleranță la defecte. Căutând un optim pentru impactul performanță-cost a rezultat ca favorabilă aplicarea principiului redundanței triplei modulare, deci a votului majoritar 2 sau 2 din 3 care să stea la baza unui modul tot MULTIPROM, generator de tact cu toleranță la defectare care a obținut indicativul MPCB-05. Acesta constituie primul din seria unor module care formează un nucleu hardware de fiabilitate sporită pe care se bazează aplicarea în mod original a unei configurații duale sincrone, de tip activ. Acest prim modul al nucleului hardware de fiabilitate sporită

este atribuit unui al trilea sertar pe lângă cele două cu configurațiile deja amintite, şi acesta din urmă alimentat separat (prin intermediul unui alt modul, al treilea, MPSA-02)

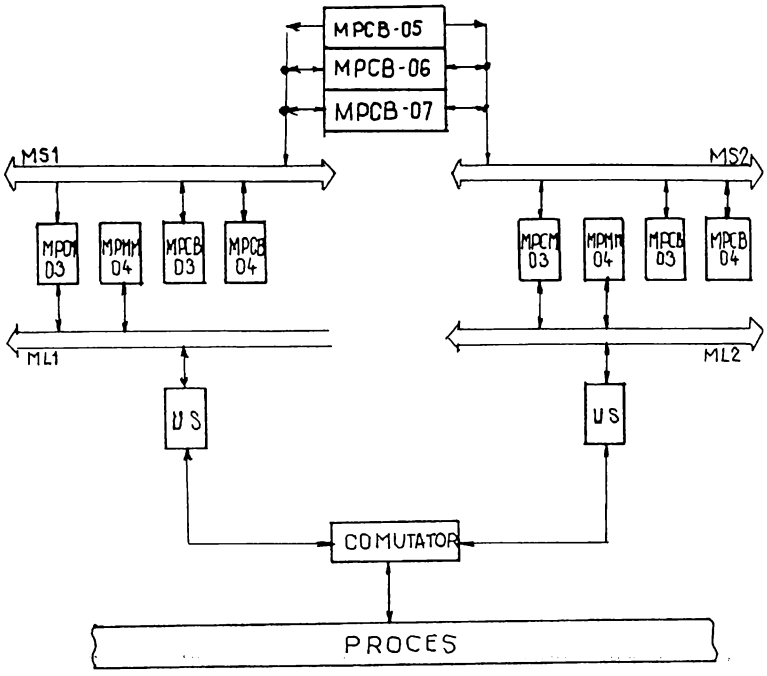


Fig.6.1

Odată asigurată funcționarea sincronă în maniera descrisă, se pune problema secvenței de operații care trebuie traversată din momentul ieşirii din sincronism a celor două sisteme datorită malfuncționării care se admite că apare la unul dintre acestea. În mod clar, fundamentat şi pe teorema lui Bernoulli din teoria probabilităților admitem apriori ca valabilă ipoteza probabilității net superioare a apariției defectului singular cu raportul probabilităților defectelor multiple. Prin urmare,cele două echipamente ies din sincronism şi, într-o primă instanță, se impune stabilirea sistemului cu funcționare corectă respectiv a sistemului defect. Pentru determinarea momentului ieşirii din sincronism, deci efectiv a momentului de apariție al defectului, în vederea reducerii la maximul a latenței defectului cu consecințele favorabile relevate în partea teoretică, se apelează din nou la o soluție hardware. Astfel, fluxurile informaționale prezente pe cele două magistrale de sistem sunt prevăzute a fi încărcate permanent într-un modul de comparare cu menirea de a realiza în fiecare moment comparația logică a semnalelor de pe linii omoloage de pe cele două magistrale de sistem. La sesizarea chiar şi a unei singure neconcordanțe logice acest modul are menirea generării unui

semnal de întrerupere pentru cele două echipamente de calcul, semnificând apariția unei anomalii funcționale la unul dintre ele.Întrucât acest modul este, ca și anteriorul generator de tact, comun celor două sisteme el trebuie să aibă la baza configurației un eficace principiu care să asigure toleranța la defecte și care a fost adoptat din nou, unul bazat pe votul majoritar 2 sau 2 din 3. Schemele de comparare au fost găsite din punct de vedere tehnologic ca favorabil a fi implementate sub forma unui nou modul MULTIPROM care a obținut indicativul MPCB-07 fiind și el atribuit nucleului hardware de fiabilitate sporită (fig.6.1).

Urmărind în continuare secvența de operații ce se derulează din momentul sesizării faptului că unul dintre calculatoare s-a defectat impunându-se determinarea care dintre cele două este purtătorul stării de malfuncționare. În acest punct se cuvine subliniat aspectul legat de toleranța care trebuie asigurată de această dată din punct de vedere al timpului de reacție corespunzător procesului, care se impune a fi acoperitor pentru toată secvența de operații ce urmează a fi descrisă în continuare. Legătura cu procesul este admisă a fi asigurată prin intermediul unor registre tampon care păstrează fără modificări pulsul logic la nivelul liniilor de comunicație cu procesul, iar cele două sisteme de calcul își întrerup activitatea curentă de control a procesului prin declanșarea unor secvențe de programe menite a salva stările celor două echipamente în zone distincte ale unei memorii comune. În acest mod cele două configurații de calcul devin disponibile pentru a se putea declanșa pe ele, în parte,proceduri de autotest menite a reliefa echipamentul purtător al stării de defect. Revenind la memoria anterior menționată, fiind și ea comună celor două echipamente trebuie inclusă în nucleul hardware de fiabilitate sporită implicând soluții constructive care să aplice tolerarea defectelor la nivelul acestui subansamblu funcțional. Astfel, s-a conturat un al treilea modul esențial exceptându-l pe MPSA-02 al nucleului de fiabilitate sporită care a obținut indicativul MULTIPROM MPCB-06 și care are la baza construcției aplicarea la nivelul magistralei de 8 biți a unui cod Hamming corector al erorii singulare și detector al erorilor duble. În acest mod, este sub control permanent funcționarea acestui, mai vulnerabil la defectare, modul al nucleului hardware de fiabilitate sporită. Cu stările salvate în modulul descris ale celor două sisteme, pe fiecare din acestea sunt declanșate procedurile de autotest menite căutării sistemului corect respectiv a celui defect. Sistemul de programe de autotest declanșat concomitent pe cele două sisteme se compune din două subsisteme. Primul, puțin voluminos este constituit din segmentul de programe de autotest destinate detecției stării de malfuncționare și al doilea, mult mai voluminos comparativ cu primul, este destinat localizării, diagnosticării,și izolării modulului purtător al defectului .În baza celor analizate în extenso în capitolele teoretice (1,2 şI 4), construcția segmentului de programe autotest de detecție este cu atât mai eficace prin prisma scopului urmărit cu cât este "împănat" mai frecvent cu instrucții de comparare între rezultatele obținute ca urmare a rulării programelor de autotest și informația etalon considerate răspunsurile la o funcționare corectă atunci când sunt aplicate aceste programe .Aceste foarte frecvente instrucții de comparare care, pe de o parte, asigură o eficace căutare a defectului, grevează pe de altă parte în mod defavorabil asupra capacității de trecere a programelor lungind intervalul de timp necesar rulării programelor de autotest destinate detecției. În mod original, pentru

îmbunătățirea parametrului constând din timpul necesar trecerii programelor de autotest de detecție, în cele două structuri de calcul identice au fost prevăzute câte un modul ASIM cu reacții programabile de tipul MPCB-04 modificat. Acesta are menirea înlăturării frecventelor instrucții de comparare păstrând din alcătuirea programelor de autotest de detecție doar partea care asigură stimularea unității testate. Partea care era destinată în variantele convenționale evaluării răspunsurilor unității testate prin instrucții de comparare este acum captată hardware de către cele două module amplasate pe liniile de magistrală, urmând ca la intervale de timp prestabilite, să fie comparate, în parte, semnături generate de către module cu semnături etalon care corespund funcționării corecte în conjuncție cu modulele MPCB-04, în fiecare din cele două sisteme identice, în acelaşi scop de supraveghere şi control mai sunt incluse, în fiecare, câte un modul MPCB-03 [Vasi-87c,d]. Într-o descriere foarte sumară modulul MPCB-03 îndeplineşte următoarele funcțiuni:

*supraveghează relațiile temporale ale anumitor semnale electrice, periodice sau nu, precum şi încadrarea între limite impuse a anumitor parametri(temperatură,tensiuni) ai sistemului.

*generează semnale auxiliare ale magistralei

*generează semnale de prioritate a maximum 8 masteri pe magistrala de sistem

*prin seturi de registre interne ale modulului un procesor poate exersa,poate stimula diferite funcțiuni şi poate citi informațiile de stare răspuns în cadrul regimului de diagnoză

*generează întreruperi între procesoare

*generează întreruperi de serviciu de caătre un operator uman

*informează, la pornire, sistemul despre starea altor sisteme ce au pornit anterior-în cazul pornirii în cascadă a mai multpr sisteme.

Este momentul specificării în acest context a faptului că, desigur, într-o variantă actualizată din punct de vedere tehnologic, cele două module MPCB-03 şi MPCB-04 se impun comasate sub forma unei structuri de supraveghere şi control evoluate implementate prin circuite integrate dedicate, ceea ce ar putea constitui o dezvoltare în continuare a acestei problematici. Revenind, într-o manieră care prin elementele redundanre menționate permite implementarea eficientă a toleranței la defectare, scurtând corespznzător timpul destinat trecerii programelor de autotest, se determină acela dintre sisteme care este defect. Derularea în continuare a secvenței de operații implică actualizarea stării sistemului găsit cu funcționare corectă prin rztine destinate încărcării stării din zona aferentă a modulului din memoria comună după care sistemul găsit cu funcționare corectă preia conducerea întreruptă a procesului industrial în mod individual. În acelaşi timp, sistemul defect va rula, în mod individual, segmentul destinat diagnosticării modulului din cadrul programelor de autotest. În cadrul acestei activități, pentru a lăsa procesul o perioadă de timp cât mai scurtă sub conducerea unui singur echipament de calcul, modulele MPCB-04 în conjuncție cu MPCB-03 îşi dovedesc, pentru a doua oară,deosebita eficiență prin aceeaşi decisivă funcție care permite eliminarea din segmentele de program de diagnosticare a foarte frecventelor instrucții de comparare şi substituirea lor prin foarte rarele comparații de semnături generate în maniera descrisă prin modulul MPCB-04. În urma rulării programelor de diagnostic împreună cu activitatea modulelor MPCB-04 şi respectiv MPCB-03 va fi localizat, în ultimă instanță, modulul defect şi, în urma unei semnalizări, acesta va fi înlocuit prin

intervenţia operatorului. Consecutiv acestei operaţii este prevăzută resincronizarea celor două echipamente de calcul prin preluarea de către sistemul fost defect a stării curente corespunzătoare calculatorului anterior detectat cu funcţionare corectă.

Comentând soluţii alternative de configurare a unor sisteme dual sincrone cu implementare eficientă a toleranţei la defecte prin elementul redundant constituit de modulul ASIM cu reacţii programabile, se poate contura o versiune care să utilizeze respectivul modul pentru detecţia momentului de apariţie a defectului. În această situaţie modulul ASIM ar putea prelua funcţiile modulului comparator eliminându-l din configuraţia nucleului hardware de fiabilitate sporită, implicând comparaţia efectuată la nivelul informaţiilor comprimate constituite de semnături. Cu atât mai mult în acest caz se impune o acoperire cât mai completă a defectelor potenţiale prin metoda de comprimare, sens în care modificarea efectuată în capitolul 3 asupra schemelor BILBO se dovedeşte cu atât mai utilă.

În consecinţă, uzitând, fără a pierde din generalitate, de module MULTIPROM, în acest paragraf s-a prezentst o originală implementare a principiului dublării active combinând redundanţa de tip global cu cea de tip distribuit. În scopul obţinerii unor timpi cât mai reduşi în care procesul să fie sub conducerea unui singur calculator, s-au preferat rezolvării predominant hardware. În mod original, s-a configurat un nucleu hardware de fiabilitate sporită la care soluţiile de introducere a redundanţei se aplică în mod distribuit, la fiecare dintre module în parte. Cu sinteza de detaliu a acestor module aparţinând nucleului hardware de fiabilitate sporită ne ocupăm în paragraful următor. S-a urmărit, de asemenea, o reducere la minim a elementelor comune celor două sisteme, cu redundanţă globală. În ceea ce priveşte funcţiile de toleranţă la defecte implementate software, acestea sunt restrânse la programele de autotest [Vasi-88] dar şi acestea sunt consistent reduse ca volum prin eliminarea anterior menţionatelor frecvente instrucţii de comparare.

6.2.Sinteza componentelor nucleului hardware de fiabilitate sporită

6.2.1.Modul generator de tact cu toleranţă la defecte

Prima componentă a nucleului hardware de fiabilitate sporită reprezentat de modulul MPCB-05, are, în conformitate cu cele prezentate în paragraful 6.1., rolul furnizării trenurilor impulsurilor de tact pentru sisteme configurate pe magistrala MULTIPROM şi prevăzute cu facilităţi de toleranţă la dectare. Revenind asupra observaţiei de esenţă , cum că tipul magistralei standard este neesenţială, în contextul prezentei lucrări importante fiind strategiile aplicate pentru asigurarea toleranţei la defectare, iar magistrala MULTIPROM oferă doar cadrul pentru implementarea unui exemplu de proiectare. În plus, modulul hardware MPCB-05 mai conţine o parte a circuitelor de comparare pentru semnalele de comandă de pe cele două magistrale de sistem care fizic nu au putut fi amplasate pe modulul comparator MPCB-07 din lipsă de spaţiu. Ca şi celelalte module din familia MULTIPROM, acesta se poate cupla atât la magistrala de sistem, cât şi la cea rezidentă şi poate funcţiona în sisteme pe 8 sau 16 biţi. Trenurile impulsurilor de tact sunt caracterizate prin parametri specifici familiei MULTIPROM dar pot fi modificaţi pentru adaptarea la un alt tip de familie

standard. Astfel, perioada semnalului de tact este de 100ns.La nivel bloc,schema modulului este prezentată în fig.6.2.

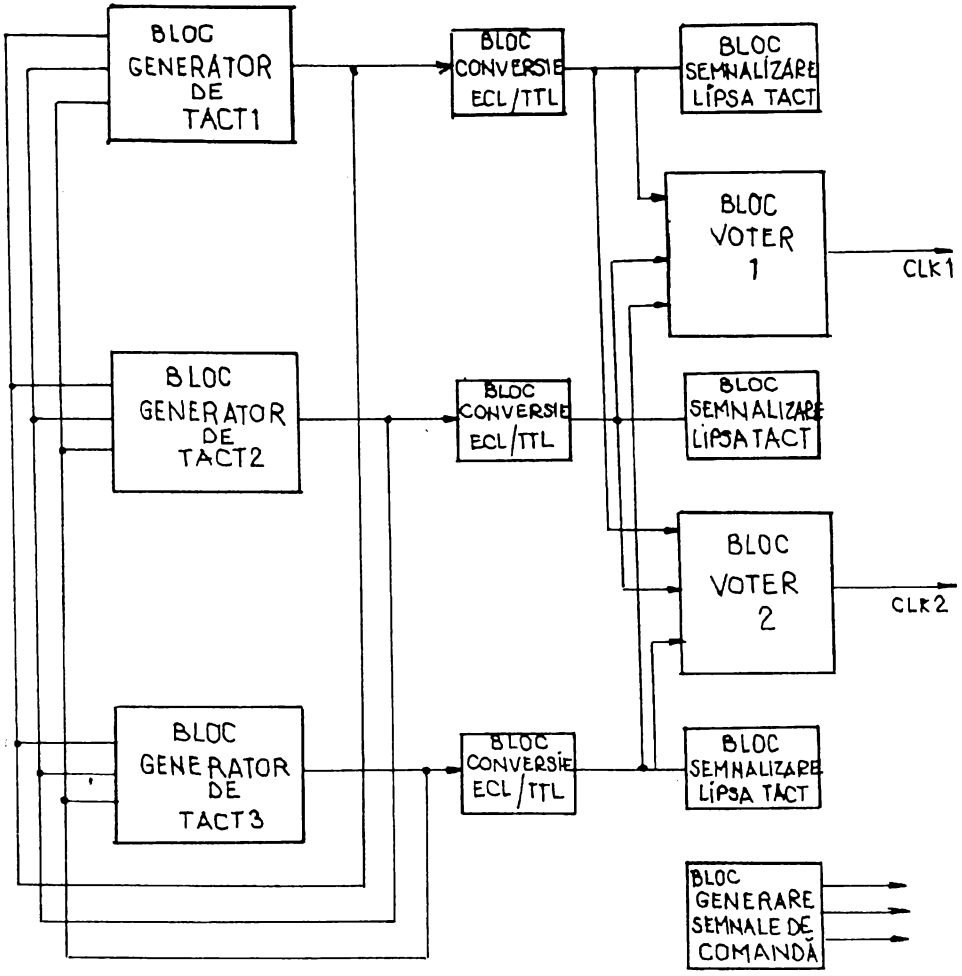
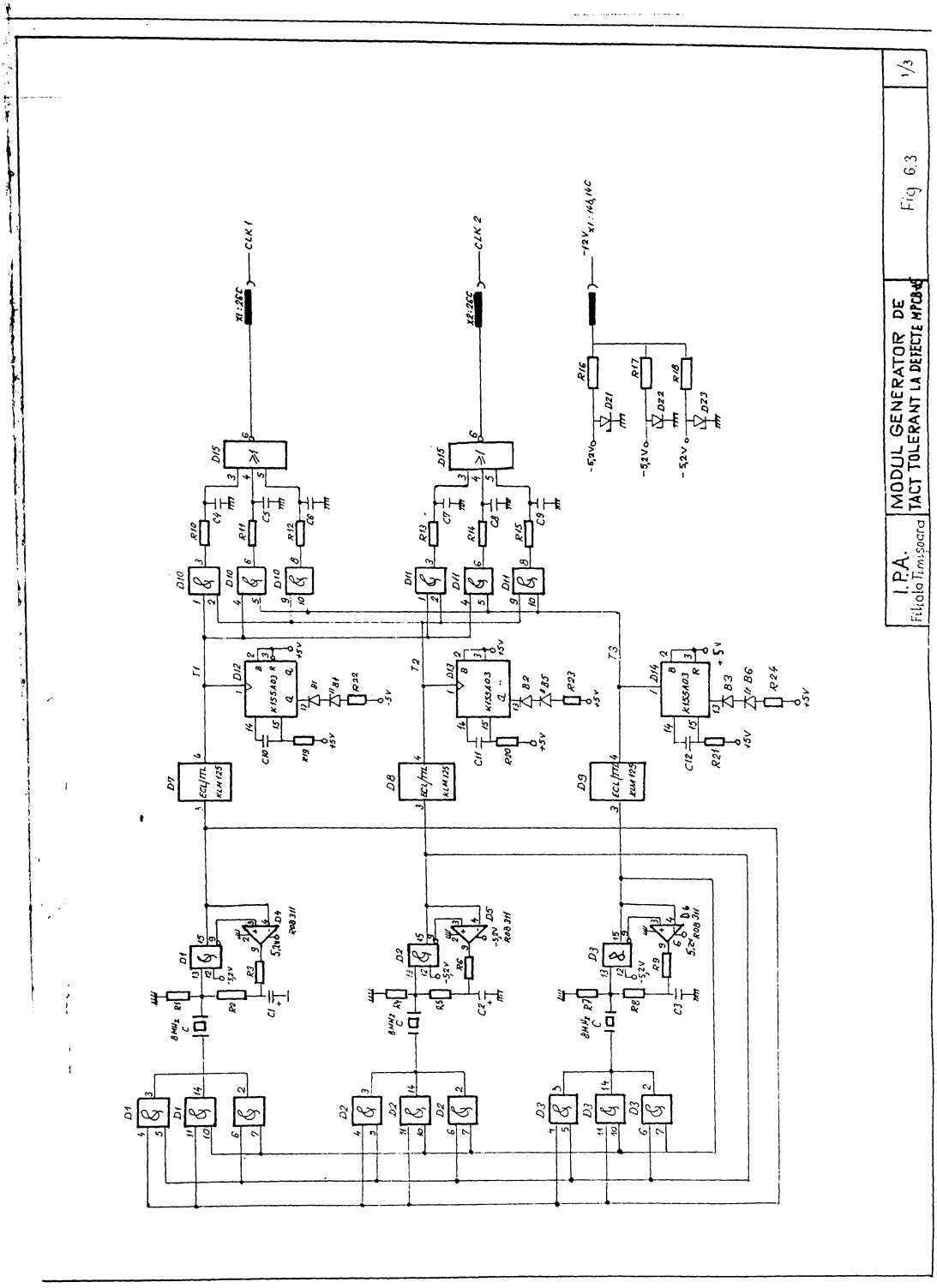
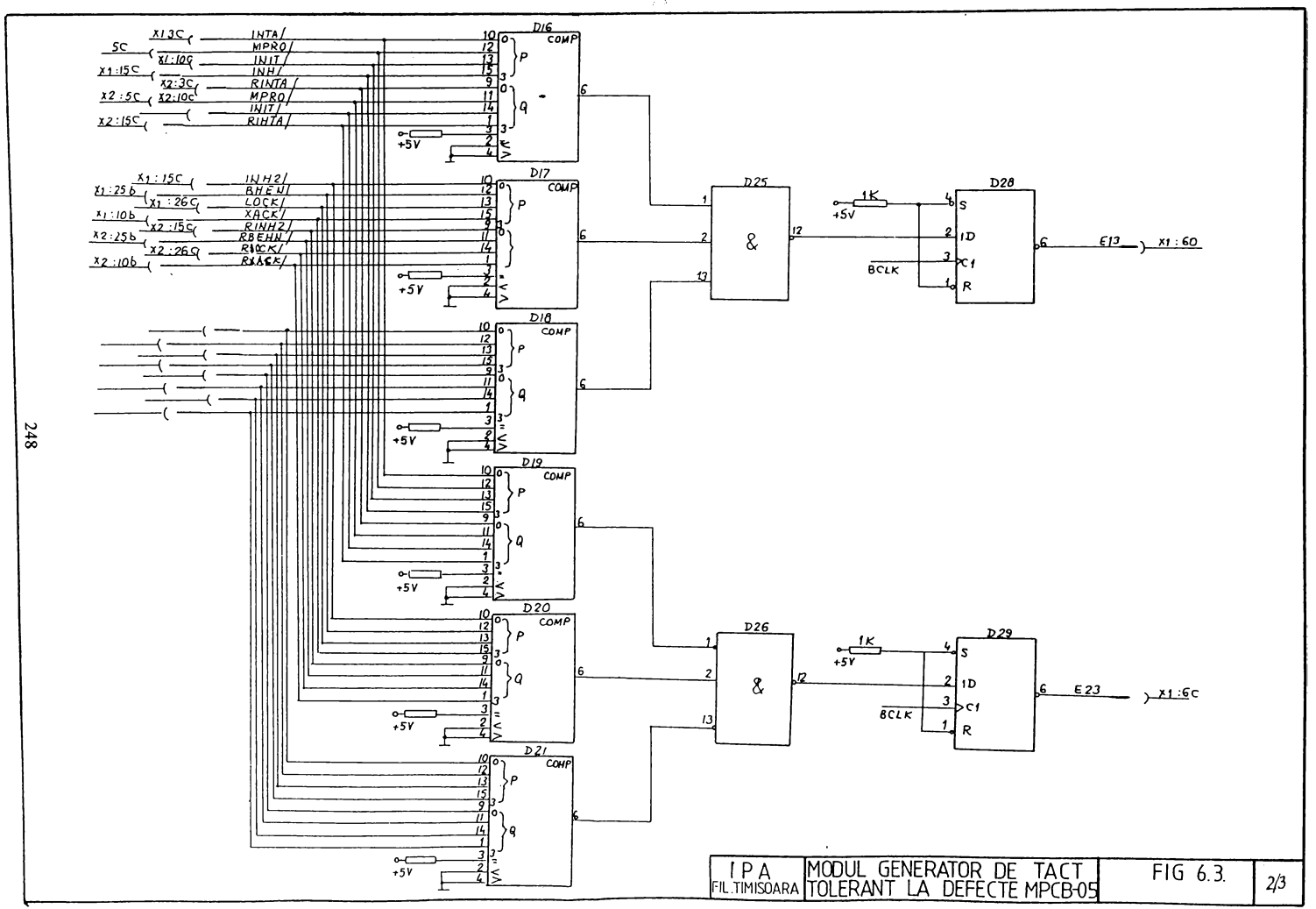
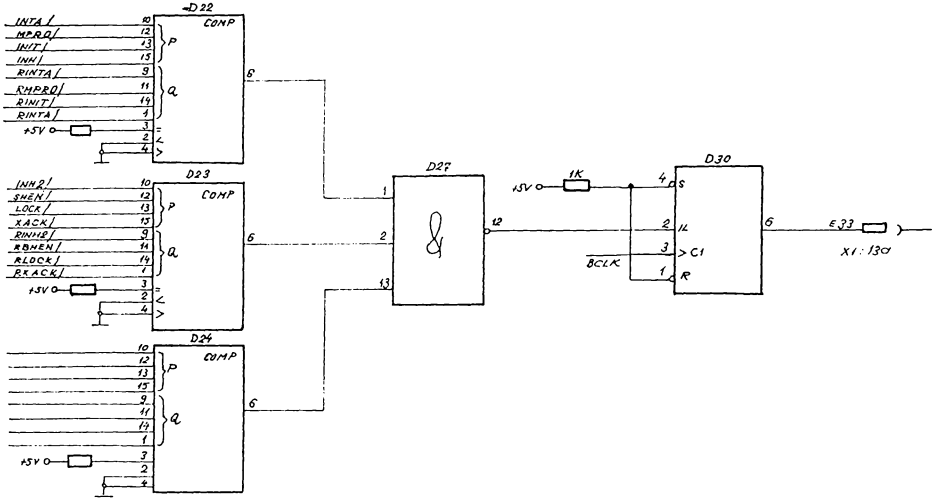


Fig.6.2

Aşa cum se menţiona în 6.1, în vederea asigurării toleranţei la defectare la nivelul generatorului de tact se apelează la principiul redundanţei triple modulare care implică existenţa a trei lanţuri de circuite identice (fig.6.2). Adaptarea unui generator de tact cu redundanţă triplă modulară la un sistem dublat este asigurată, în mod original, prin intermediul celor două blocuri de votere care implementează principiul votării logice după majoritatea 2 sau 3 din 3.


I.P.A.
Filiala Timisoara
MODUL GENERATOR DE TACT TOLERANT LA DEFECTE MPD 8
Fig 6.3
1/3


D16
D17
D18
D19
D20
D21
COMP
P
Q
+5V
INTA/
MPRO/
INIT/
INH/
RINTA/
MPRO/
INIT/
RINTA/
INH2/
BHEW/
LOCK/
XACK/
RINH2/
RBEHN/
RLOCK/
RXACK/
D25
D26
&
1K
D28
D29
S
1D
C1
R
BCLK
E13
E23
X1:60
X1:6C
IPA
FIL.TIMISOARA
MODUL GENERATOR DE TACT
TOLERANT LA DEFECTE MPCB-05
FIG 6.3.
2/3



I.P.A. Filiala Timişoara	MODUL GENERATOR DE TACT TOLERANT LA DEFECTE MPCB-05	Fig. 6.3	3/3

În vederea realizării generatorului de tact propriu-zis se apelează la o schemă cu un amplificator ECL, un cristal de cuarț și un comparator de tensiune [Vasi-87c,Vasi-88a,b,c,f]. Utilizarea amplificatorului ECL are ca avantaj încărcarea slabă a curțului, factorul de umplere al semnalului depinzând de relația dintre punctele de funcționare și nivelul porții de intrare. Amplitudinea și faza fiecărui semnal vor fi generate de diferența dintre frecvența de rezonanță și frecvența cu care este excitat cristalul. Factorul de umplere este menținut constant de către comparatorul de tensiune, acesta din urmă acționând ca o reacție inversă.(fig.6.3, fila 1/3). Trenurile de semnale de la cele trei generatoare de tact (fig.6.2), sunt aplicate câte unui bloc de conversie ECL/TTL care realizează trecerea de la nivelul ECL al semnalului de tact la nivelul necesar blocului de voter.

Lipsa semnalului de tact pentru fiecare din cele trei lanțuri de circuite este semnalizată prin LED-uri. Așa cum specificam mai sus, modulul asigură compararea semnalelor de comandă de pe cele două magistrale furnizând semnal de eroare în cazul apariției unei neconcordanțe, fig.6.3. (filele 2/3,3/3).

6.2.2. Modulul comparator cu toleranță la defecte

Cel de-al doilea modul care aparține nucleului hardware de fiabilitate sporită, purtând indicativul MPCB-07, asigură compararea semnalelor de adresă și de date de pe magistralele MULTIPROM furnizând semnal de eroare la apariția oricărei neconcordanțe la nivel de bit. Ca și în cazul anteriorului modul MPCB-05, fiind un modul comun celor două sisteme, la acesta trebuie aplicată o metodă de redundanță superioară dublării în care sens aam apelat din nou la redundanța triplă modulară.(fig.6.3).

Întrucât cele două echipamente de calcul sunt amplasate în două sertare, iar modulele nucleului hardware de fiabilitate sporită sunt amplasate într-un singur seratar, a fost necesară găsirea unei soluții tehnice de interconectare a acestora. În acest sens s-a apelat la modulul placă prelungitoare MPSA-03, câte unul pentru fiecare sistem. Astfel, unul dintre aceste module realizează interconectarea între magistrala de sistem a sertarului cu unul dintre echipamentele de calcul cu magistrala de sistem a sertarului cu nucleul hardware de fiabilitate sporită, iar cel de-al doilea modul, MPSA-02 realizează interconectarea între magistrala de sistem a sertarului cu cel de-al doilea echipament de calcul cu magistrala rezidentă a sertarului cu nucleul hardware de fiabilitate sporită. Avem astfel disponibile în cadrul sertarului cu modulele nucleului hardware de fiabilitate sporită liniile cu semnalele de pe magistralel de sistem ale celor două echipamente de calcul. După cum s-a menționat în 6.2.1, compararea semnalelor de comandă se realizează pe modulul MPCB-05 anterior descris iar rezultatele comparării reprezentate de semnalele E13, E23, E33 provenite de pe acea placă prin intermediul unor pini disponibili de pe magistrala de sistem, constituie semnale de intrare pentru blocul de memorare a rezultatelor comparării (fig.6.4).

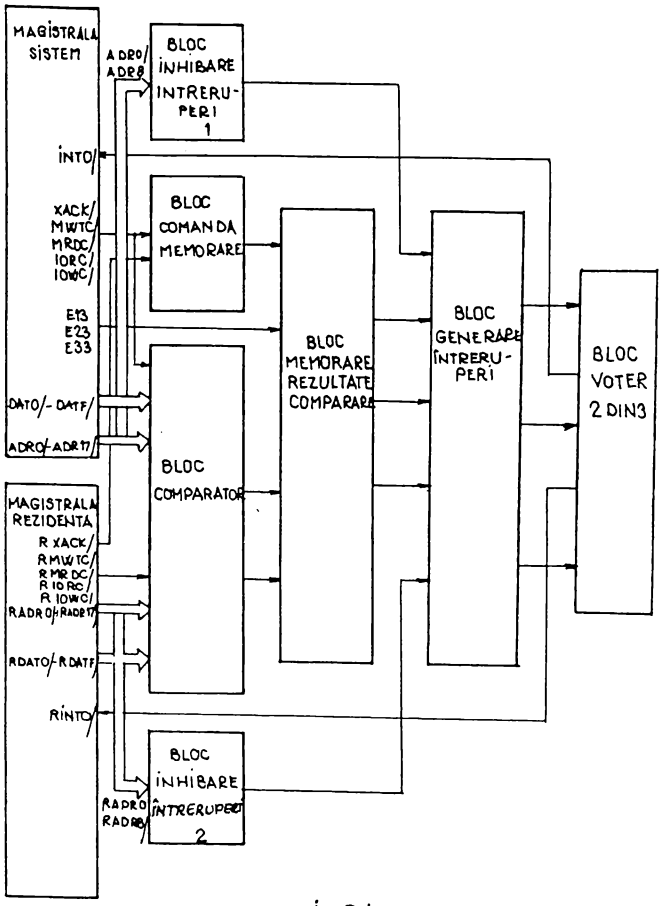
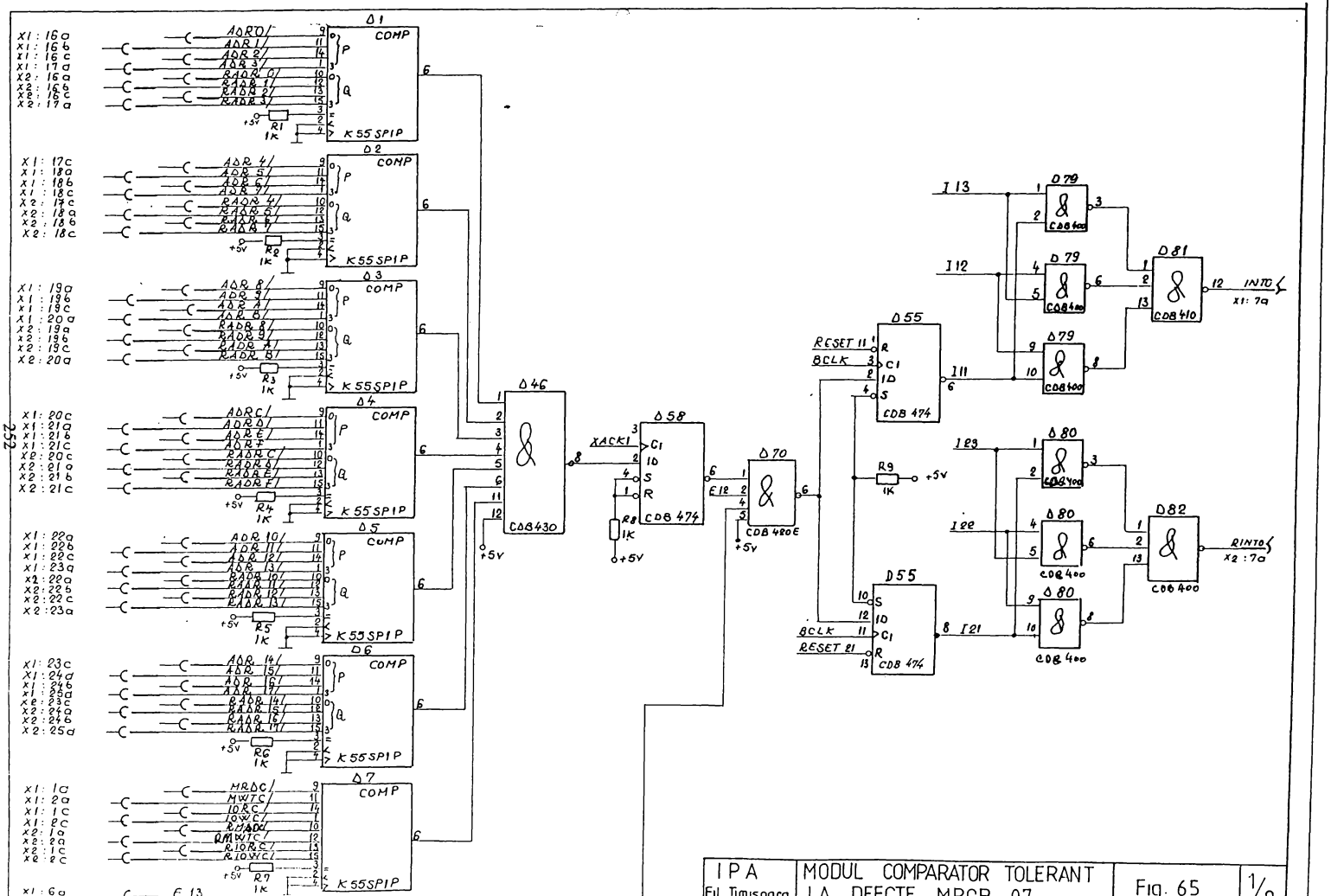


FIG 6.4

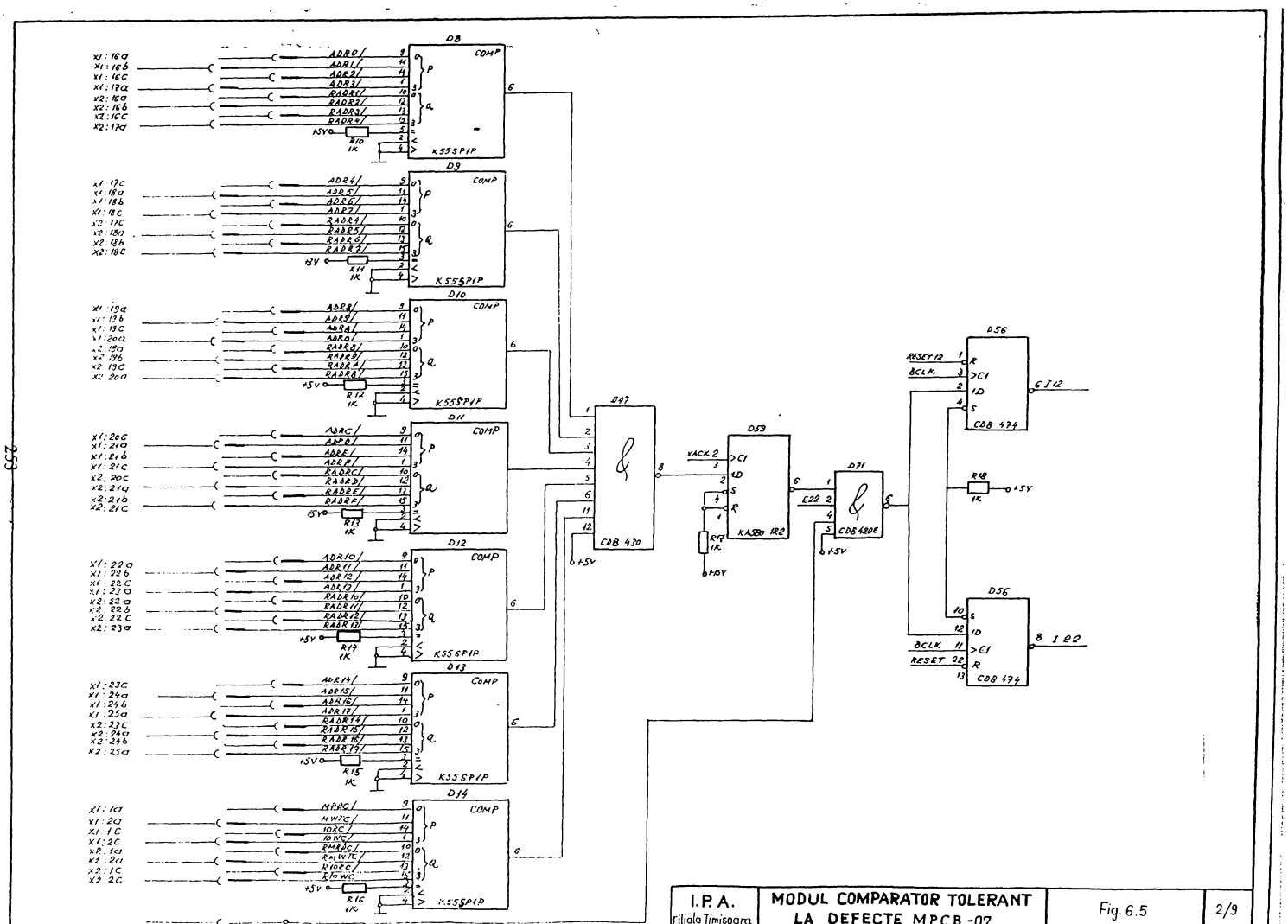
Prin blocul de comandă memorare, semnalele de pe magistrala de sistem sunt utilizate pentru a controla momentele de timp în care să fie încărcate în elementele de memorare rezultatele furnizate de către blocul comparator.

Blocul de generare a întreruperilor furnizează la blocul voter 2 sau 3 semnale care indică prezența unei neconcordanțe logice la comparația efectuată pentru semnalele de adresă respectiv de date. Pentru cele două sisteme sunt furnizate două semnale de întrerupere INTO/, respectiv RINTO/. Pe durata tratării întreruperii se impune generat semnalul de inhibare pentru blocul de generare a întreruperilor la identificarea unui anumit cod de pe magistrala de adresă pentru care au fost revăzute pentru fiecare dintre magistrale câte un bloc de inhibare a întreruperilor.(fig.6.4).

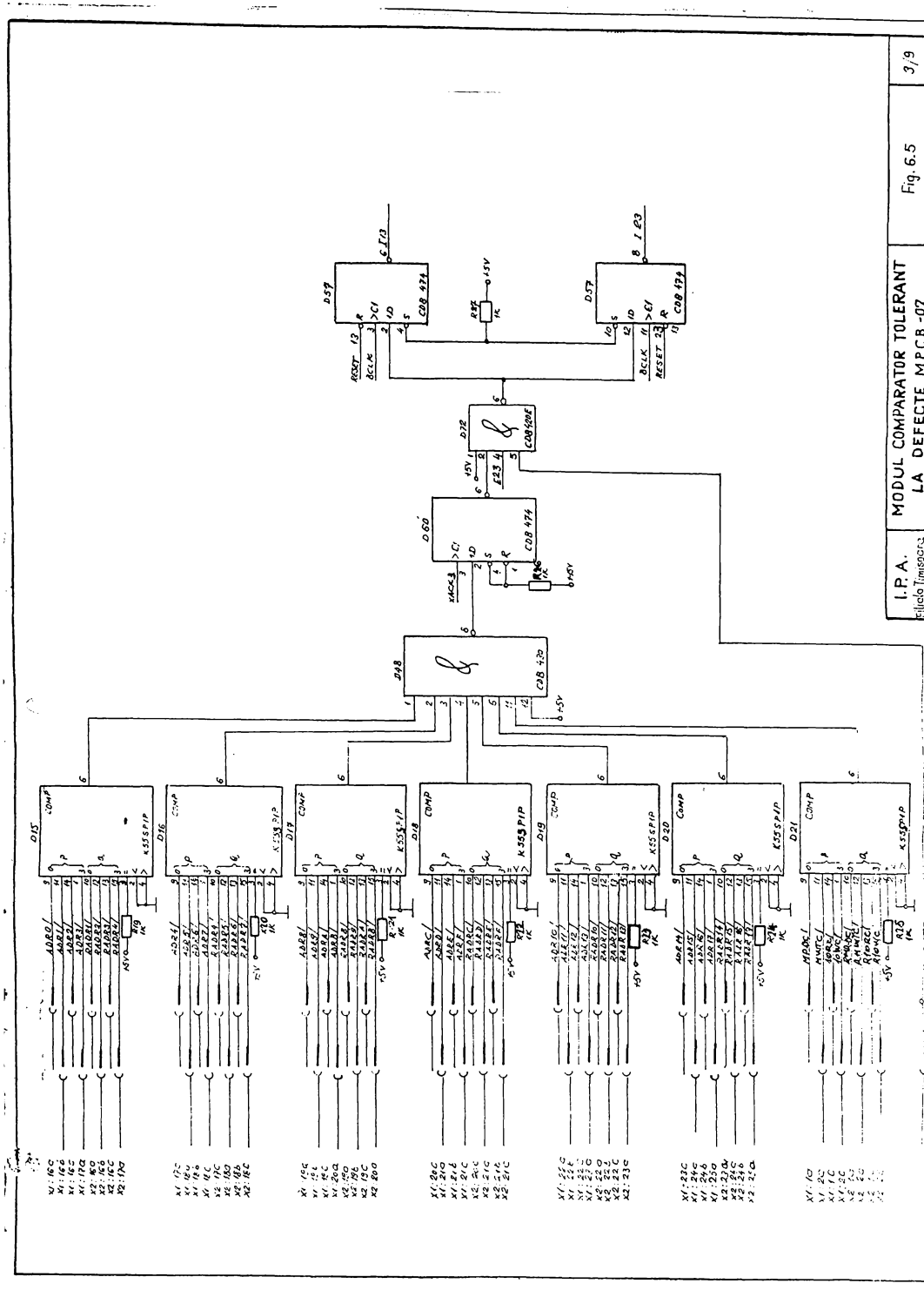


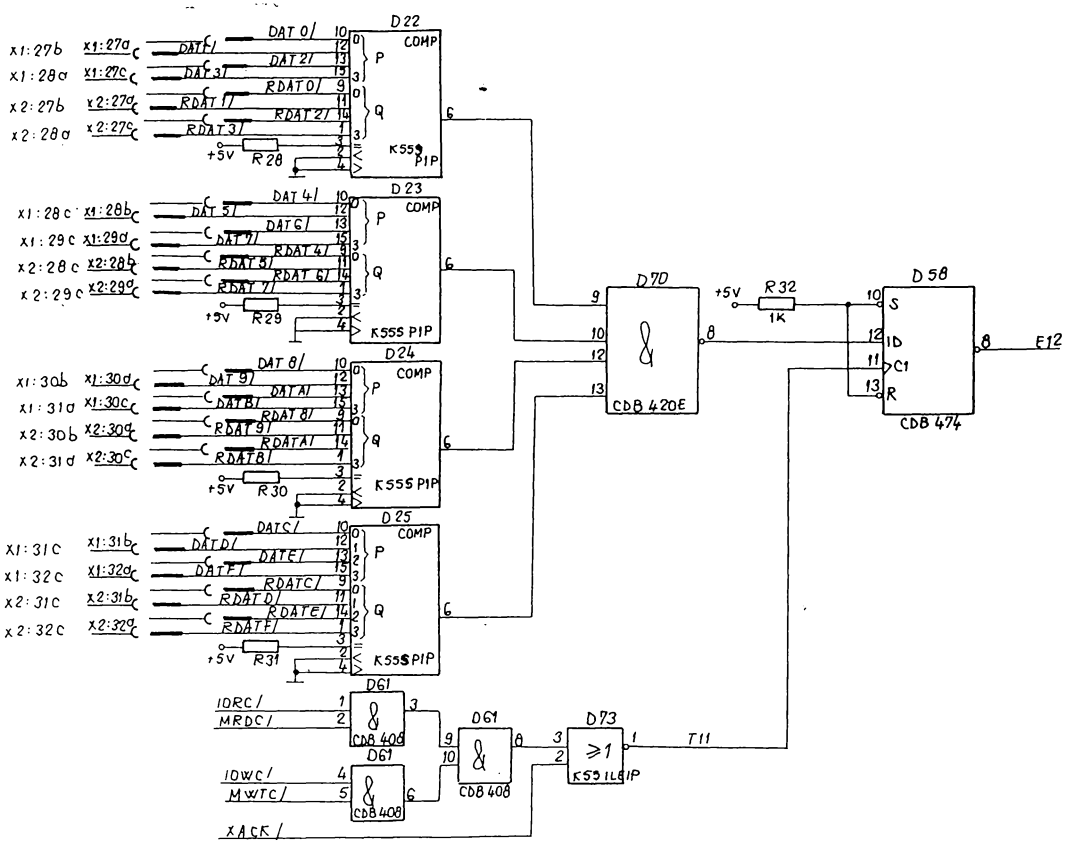
I P A Fil. Timisoara	MODUL COMPARATOR TOLERANT LA DEFCTE MPCB - 07	Fig. 65	1/9

252

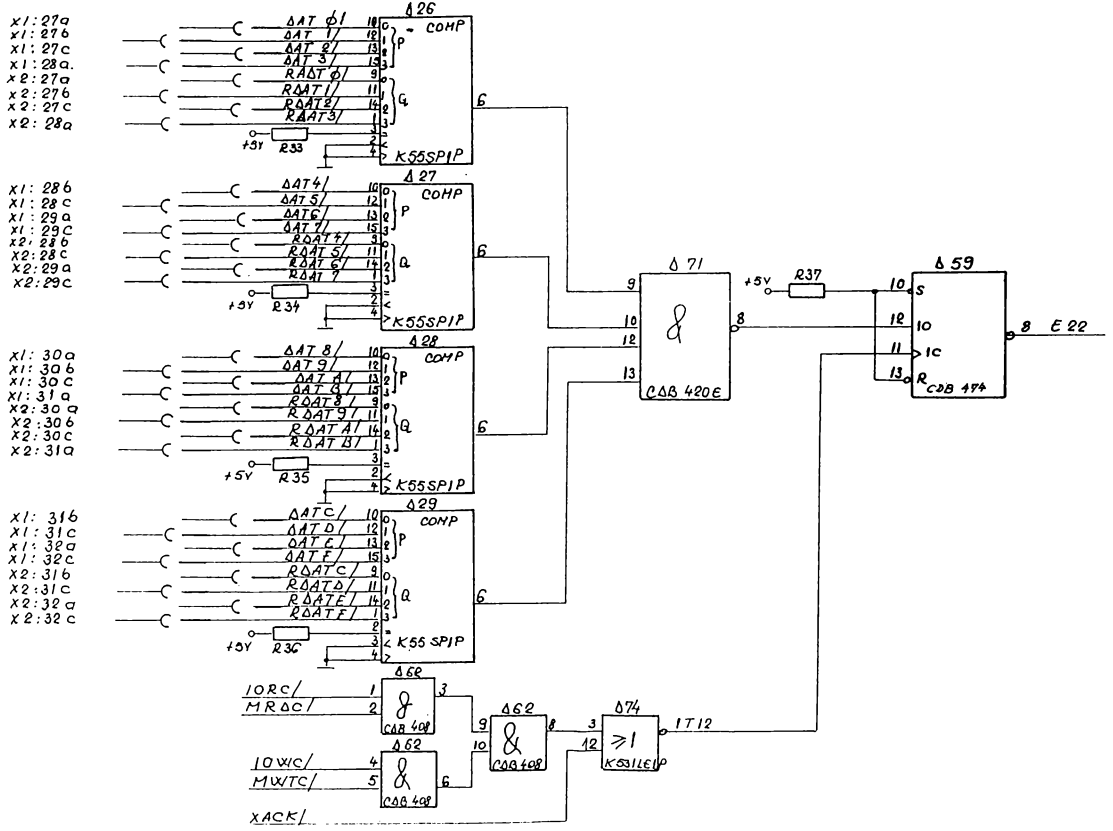


I.P.A. Filiala Timişoara	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB-07	Fig. 6.5	2/9

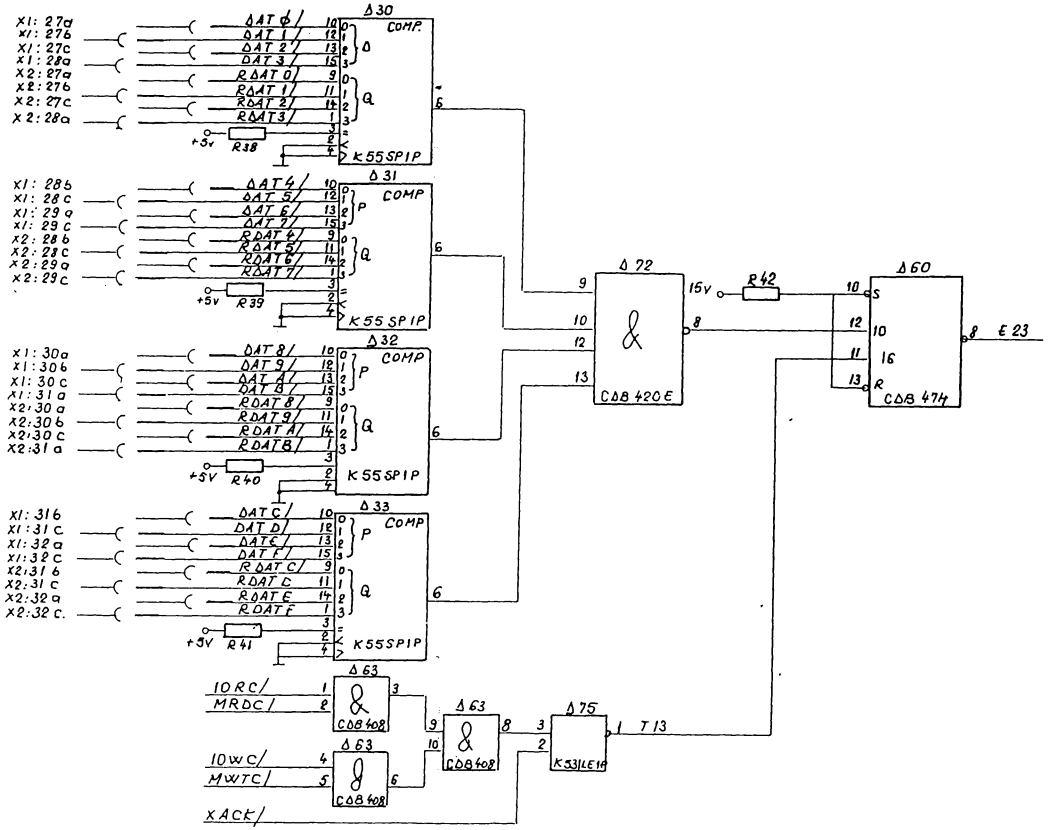

I.P.A.
Filiala Timisoara
MODUL COMPARATOR TOLERANT
LA DEFECTE MPCB-07
Fig. 6.5
3/9



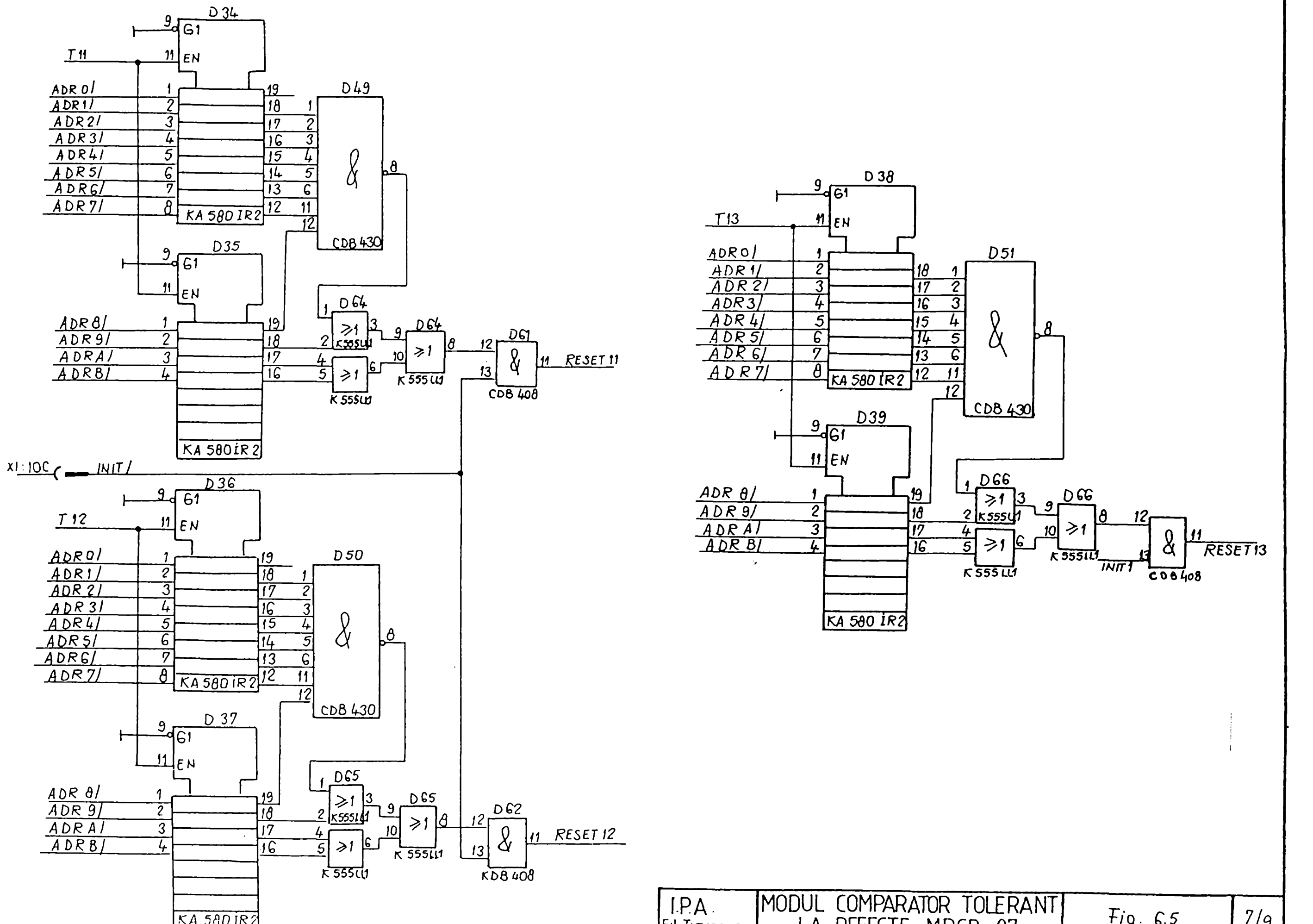
I.P.A. Fil. Timisoara	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB-07	Fig. 6.5	4/9



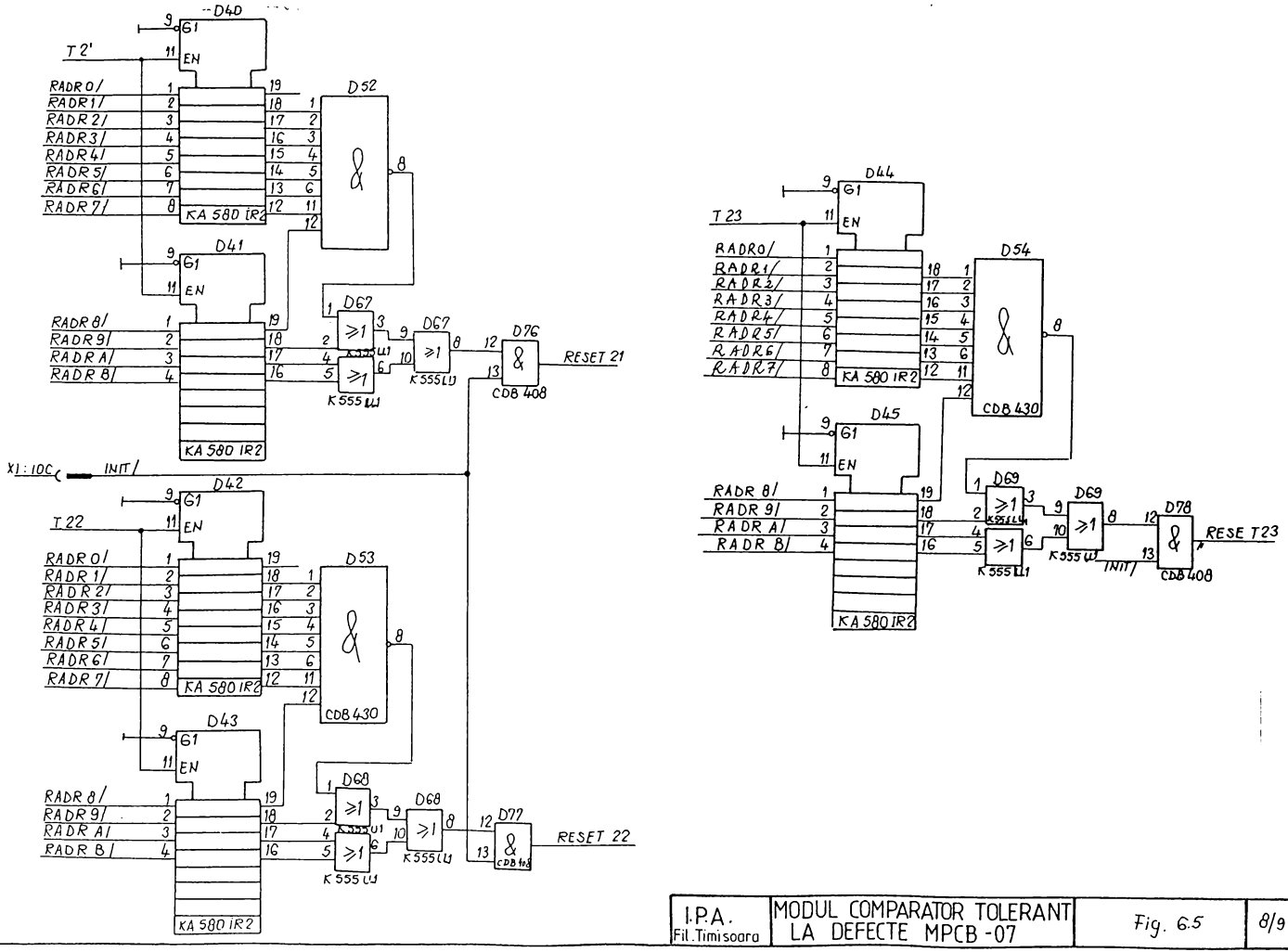
I.P.A TIMIŞOARA	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB-07	Fig 6,5	5/9



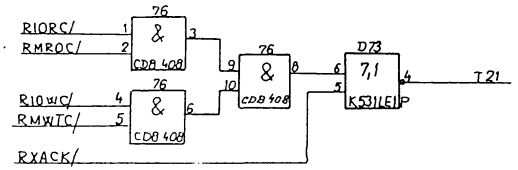
IPA Timisoara	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB-07	Fig. 6.5	6/9

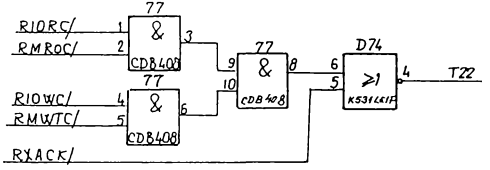


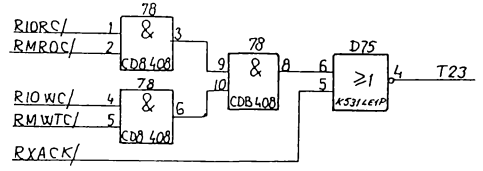
I.P.A. Fil. Timisoara	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB -07	Fig. 6.5	7/9

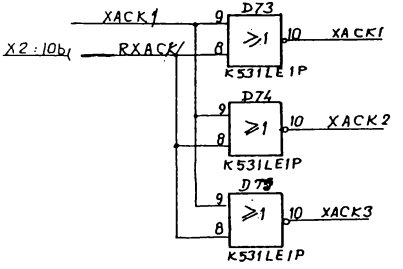


I.P.A. Fil. Timisoara	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB -07	Fig. 6.5	8/9









IPA FIL.TIMISOARA	MODUL COMPARATOR TOLERANT LA DEFECTE MPCB-07	FIG. 6.5	9/9

Din punct de vedere constructiv, apelând la aceeaşi tehnologie deja menţionată, acest modul conţine circuite identice implementate favorabil prin circuite comparatoare, iar rezultatele comparării sunt memorate în bistabilele de tip D la momente de timp determinate de semnale de scriere/citire, respectiv de semnalul răspuns al memoriei(XACK/; RXACK/).(fig.6.5, filele 1,2,3 pentru magistrala de adresă şi filele 4,5,6 pentru magistrala de date).

La identificarea unui cod determinat (1FE sau 1FF) pe liniile de adresă, printr-o schemă logică combinaţională, se obţin semnalele de inhibare corespunzătoare blocării generării întreruperilor pe durata tratării acestora.(fig.6.5, filele 7,8,9). În fine, blocul de votere 2 sau 3 din 3 ale cărui circuite sunt reprezentat în fig.6.5, fila 1, se recomandă a fi implementat cu circuite de înaltă fiabilitate prevăzute cu sufixul HR.

6.2.3.Modul memorie comună tolerantă la defectare

Cel de-al treilea modul al nucleului hardware de fiabilitate sporită reprezintă memoria comună care a obţinut în familia MULTIPROM indicativul MPCB-06. Analize efectuate asupra capacităţii de memorie necesare a condus la valoarea maximă de 64 kocteţi. Selectarea modulului se efectuează prin PROM-uri, iar modulul funcţionează ca şi celelalte module, pe 8 şi 16 biţi. Semnalele ADRO/÷ADR10/ respectiv RADR0/÷RADR10/, sunt utilizate pentru adresare în pagina de memorie, iar semnalele ADR11/÷ADR17/, respectiv RADR11/÷RADR/17 sunt utilizate pentru selectarea modulului de memorie (fig.6.6). Preluarea semnalelor de adresă se realizează prin multiplexoare 2 la 1, iar a semnalelor de date cu circuite drivere bidirecţionale. Semnalele de comandă, de scriere/ citire şi arbitrare a magistralei se utilizează la determinarea modului de lucru cu memoria şi la generarea semnalelor de validare a buffer-elor de date.

Descriind sumar, matricea de memorie este implementată prin circuite de memorie RAM CMOS 4164 (fig.6.7 fila 6).

Datele sunt emise/recepţionate prin circuite drivere de magistrală de tip 8287 validate cu semnalele DEWO/,DEW1/,DEB0/,DEB1/(fig.6.7,fila 2). Memorarea temporară a adreselor şi selecţia modulului prin circuitele PROM este reprezentată în fig.6.7, fila 1 .Pentru determinarea modului de lucru a memoriei şi semnalele OEW0/, OEW1/ de validare a buffer-elor de date, se utilizează semnalele de comandă MRDC/,MWTC/,LOCK/,respectiv RMDC/,RMWTC/,RLOCK/ împreună cu semnalele ADR0/,BHE/,RADR0/,RBHE/.(fig.6.7,fila 3).

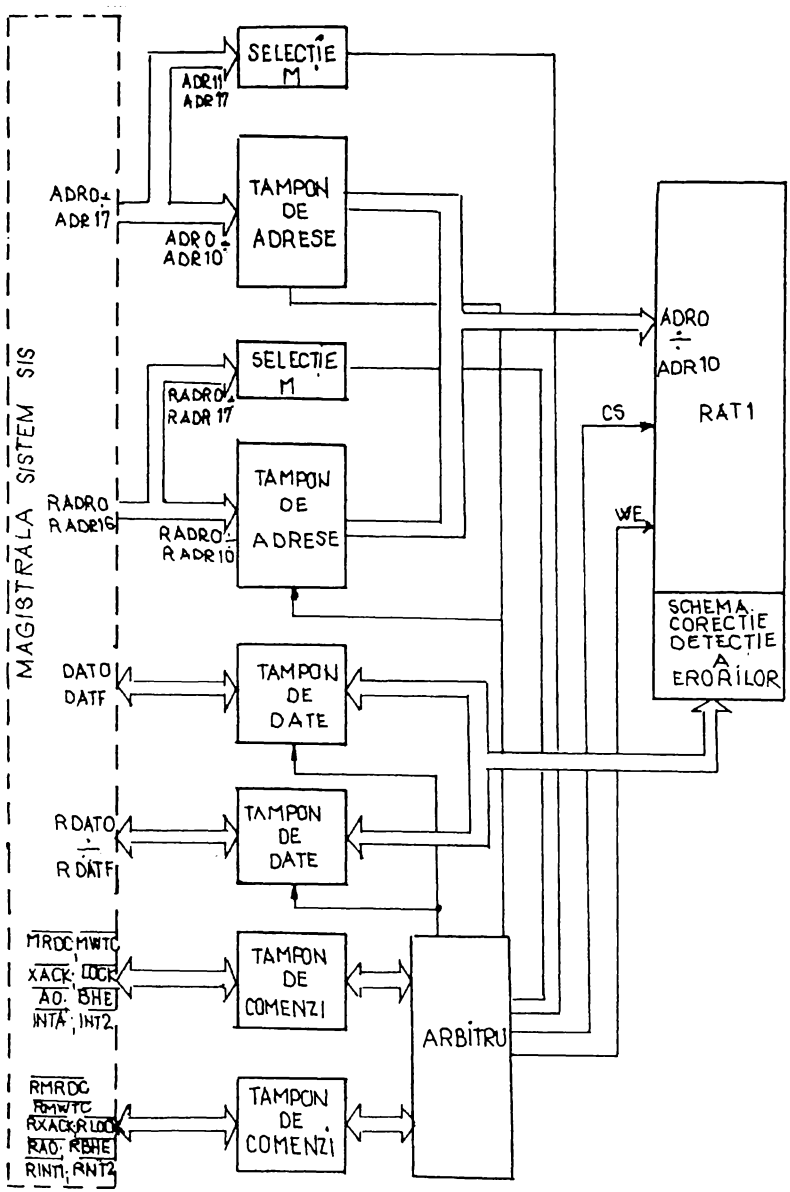
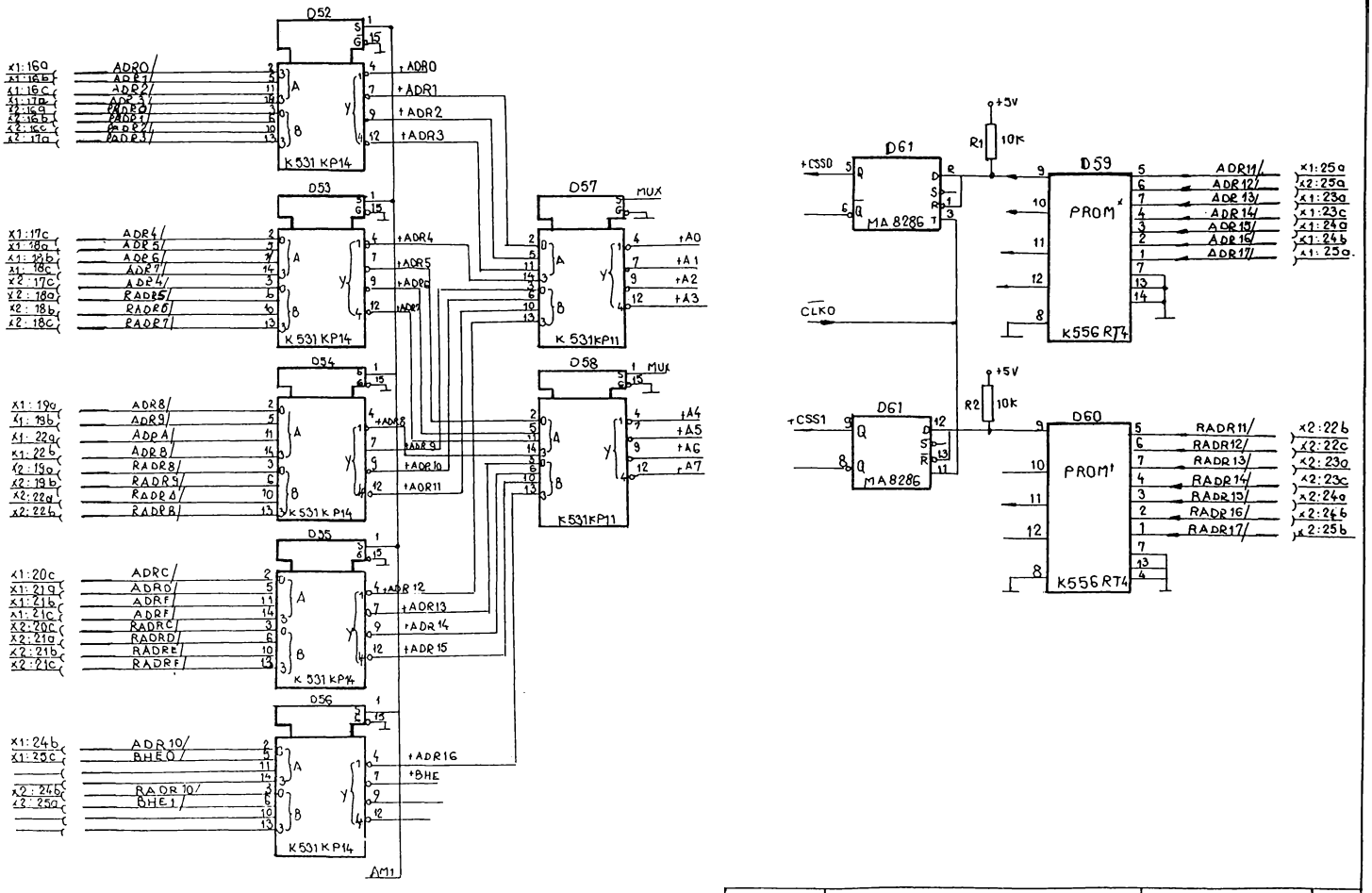
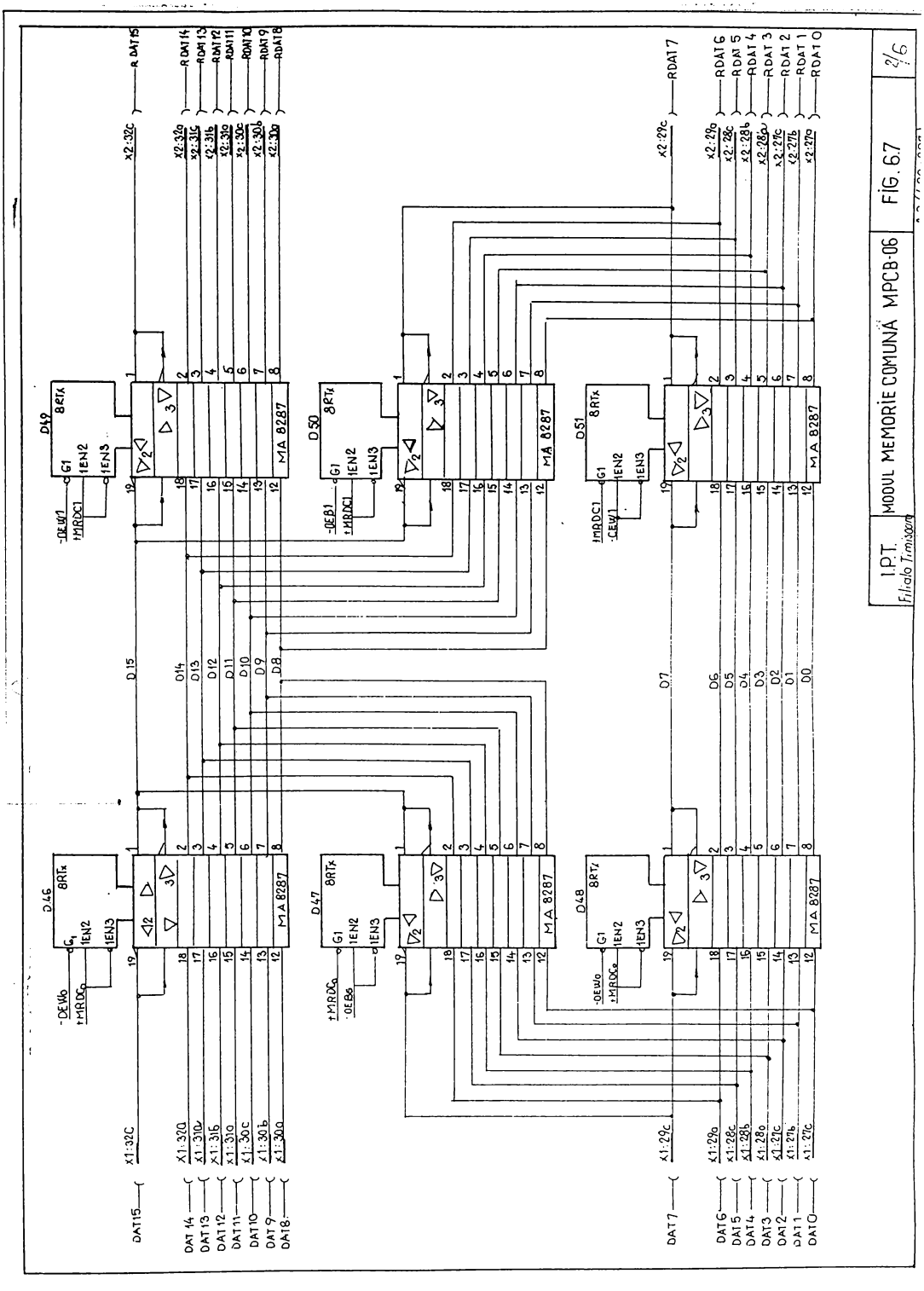
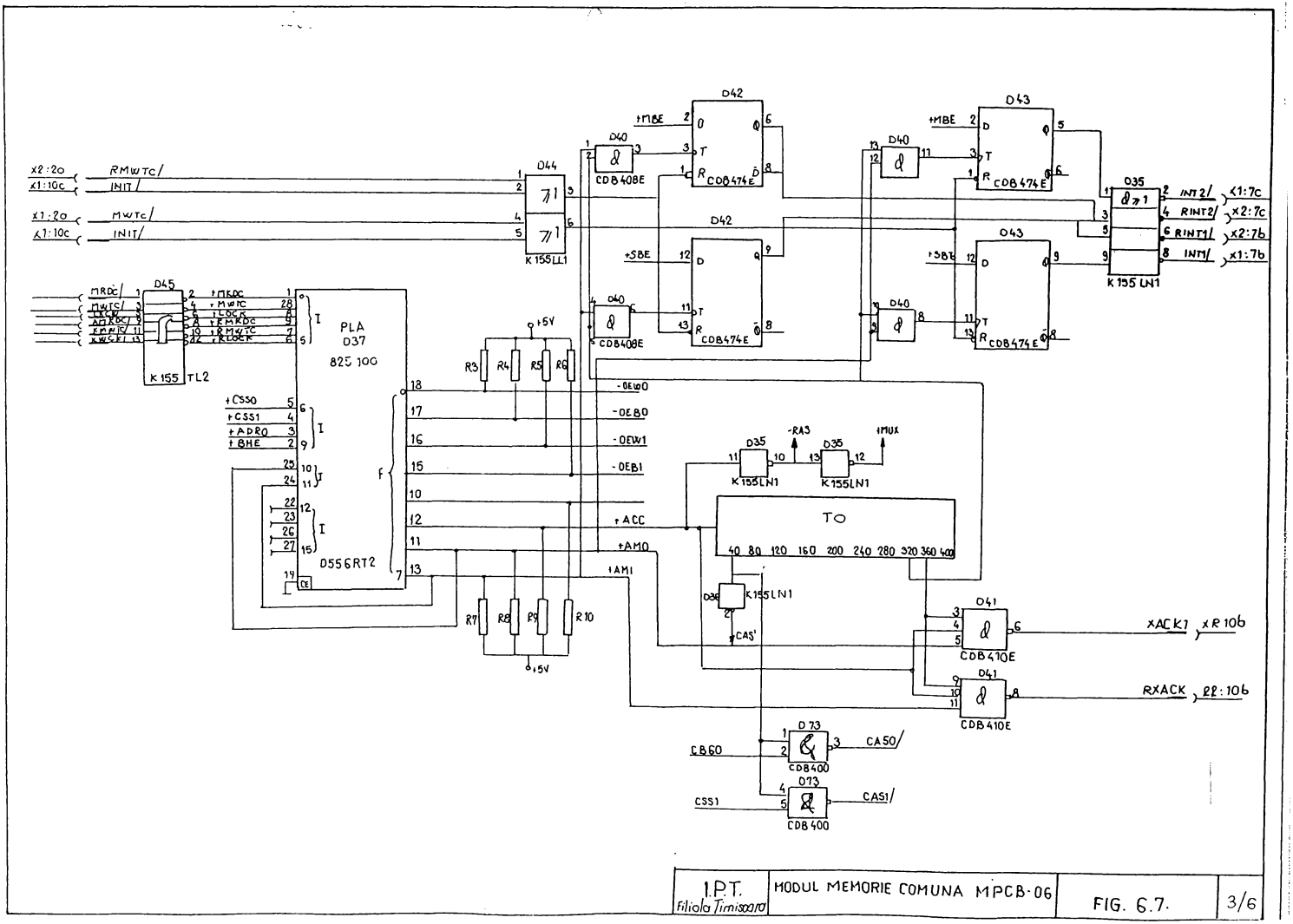

MAGISTRALA SISTEM SIS
SELECTIE M
ADR11 ÷ ADR17
ADR0 ÷ ADR17
ADR0 ÷ ADR10
TAMPON DE ADRESE
SELECTIE M
RADR0 ÷ RADR17
RADR0 ÷ RADR16
RADR0 ÷ RADR10
TAMPON DE ADRESE
DAT0 ÷ DATF
TAMPON DE DATE
RDAT0 ÷ RDATF
TAMPON DE DATE
MRDC MWTC
XACK LOCK
A0 BHE
INTA INT2
TAMPON DE COMENZI
RMRDC RMWTC
RXACK RLOCK
RA0 RBHE
RINT1 RNT2
TAMPON DE COMENZI
ARBITRU
ADR0 ÷ ADR10
RAT1
CS
WE
SCHEMA CORECTIE DETECTIE A ERORILOR

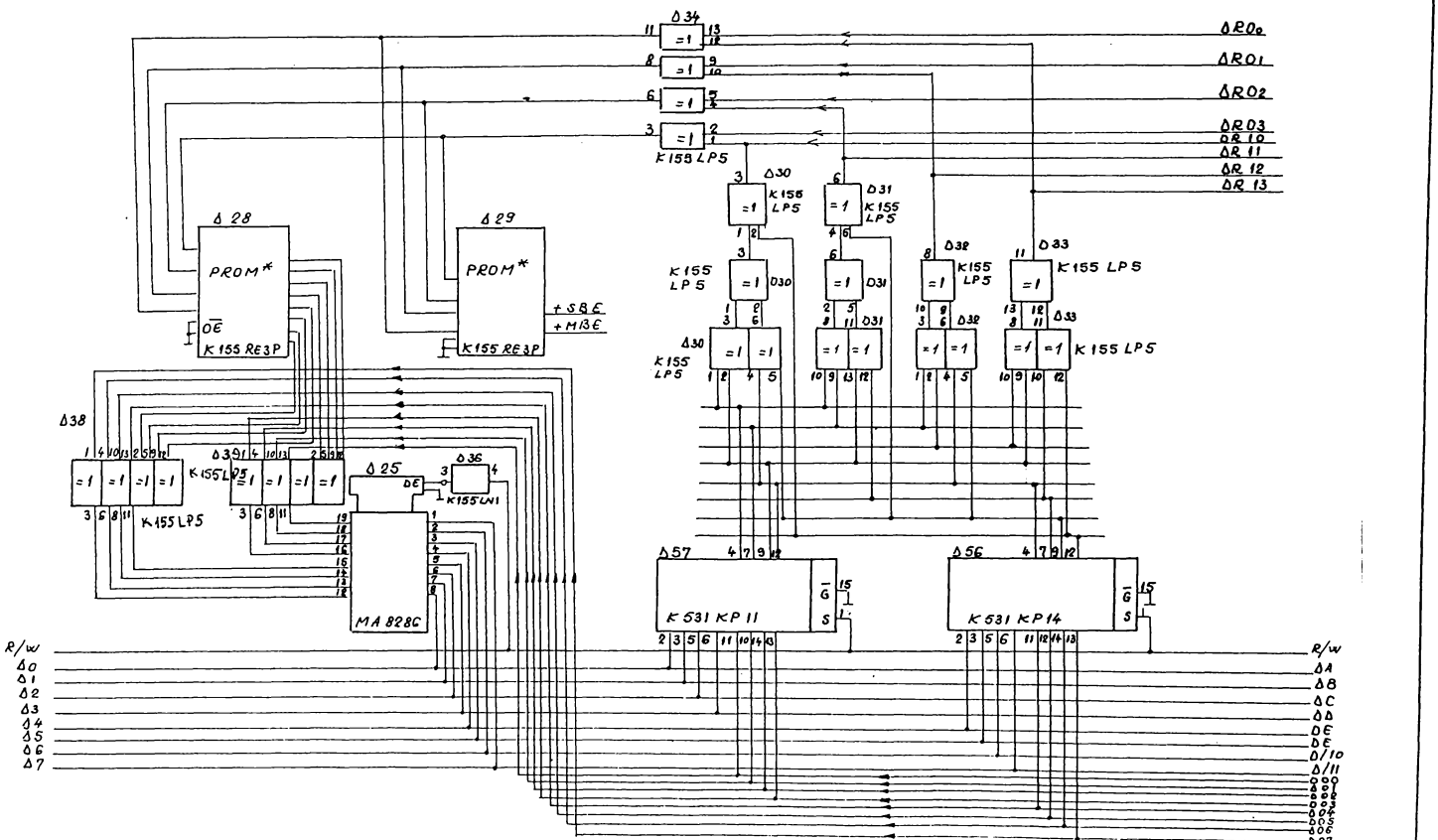
Fig.6.6



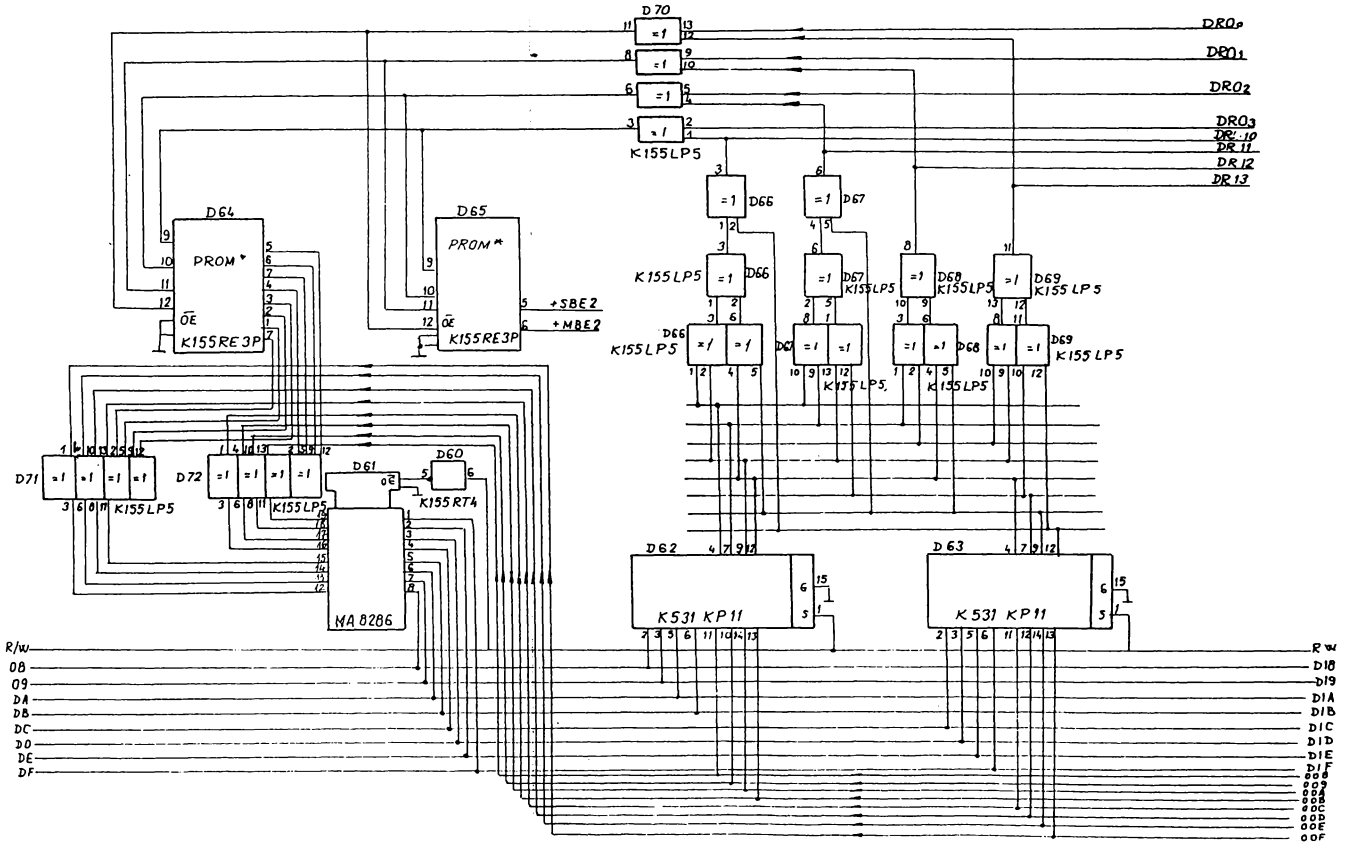
I.P.T. Filiala Timişoara	MODUL MEMORIE COMUNĂ MCPB-06	FIG.6.7	1/6


I.P.T.
Fitxalo Timisoara
MODUL MEMORIE COMUNĂ MPCB-06
FIG. 6.7
2/6

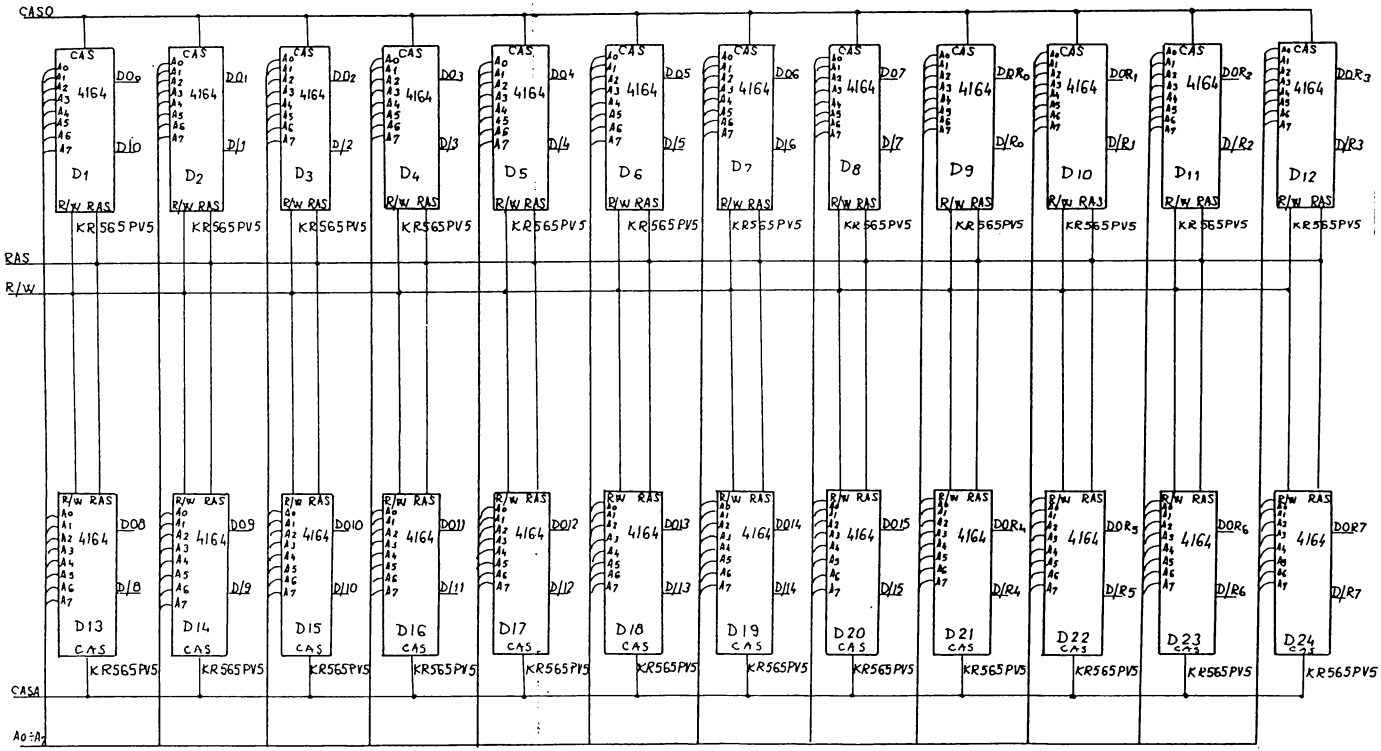

RMWTC/
INIT/
MWTC/
INIT/
D44
K155LL1
D42
D43
D40
CDB408E
CDB474E
D35
K155LN1
D45
K155TL2
PLA
D37
825 100
D556RT2
+CS30
+CS31
+ADRO
+BHE
+5V
R3
R4
R5
R6
R7
R8
R9
R10
-OEW0
-OEB0
-OEW1
-OEB1
+ACC
+AM0
+AM1
-RAS
+MUX
TO
40 80 120 160 200 240 280 320 360 400
D36
K155LN1
+CAS'
D41
CDB410E
XACK1
RXACK
D73
CDB400
CB60
CS31
CA50/
CA51/
INT2/
RINT2/
RINT1/
INT1/
I.P.T. Filiala Timisoara
MODUL MEMORIE COMUNA MPCB-06
FIG. 6.7.
3/6



I.P.A Fil. Timisoara	MODUL MEMORIE COMUNA MPCB-06	Fig 6-7	4/6



IPA TIMIȘOARA	MODUL MEMORIE COMUNĂ MPCB-06	Fig. 67	5/6



IPA FIL.TIMISOARA	MODUL MEMORIE COMUNA MPCB-06	FIG 6.7	6/6

Schema de corecție a erorilor (fig.6.7, filele 4 și 5) este implementată prin intermediul unei scheme logice combinațională adițională care se adaugă, în parte, în mod identic, pentru cele două grupe de câte 8 biți. Este necesară dublarea controlului pe cele două grupe de 8 biți în mod identic având în vedere că schema este destinată a lucra atât pe 8 cât și pe 16 biți. Prin urmare, la scriere, la cei 8 biți utili, considerați de două ori, se adaugă câte patru biți redundanți astfel încât lungimea unui cuvânt de 16 biți va apare în memorie cu lungimea de 24 biți. Pe de altă parte, la citirea unui cuvânt din memorie, biții de control, memorați câte 4 pentru fiecare subcuvânt de câte 8 biți, sunt comparați cu unii generați de către schema de corecție. Configurația binară a neconcordanțelor logice la nivelul biților redundanți, în urma decodificării permite corecția acelui bit util care a fost afectat de eroarea singulară, în partea finală, schema de corecție implică simpla inversare a bitului eronat, modulul permițând generarea a două semnale de întrerupere, unul corespunzător erorii singulare și altul corespunzător erorii duble.În ceea ce privește detaliul de proiectare al părții de corecție a erorii singulare, codul redundant utilizat este constituit de un cod Hamming modificat având la bază matricea de control H dată de (6.1).

$$H = \begin{array}{cccccccccccc} 2^0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 2^1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 2^2 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 2^3 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ & c_0 & c_1 & c_2 & c_3 & u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{array} \qquad (6.1)$$

Comentând alegerea acestei matrici se observă că pentru primii șapte biți utili ($u_{0 \div} u_6$) ei sunt generați în baza expresiei dată de (6.2) a unui polinom generator G(x).

$$G(x)=X^4+X+1 \qquad (6.2)$$

Utilizând expresia dată din (6.2) pentru G(x) și pentru al opt-lea bit util (u_7) ar rezulta configurația binară pe coloana corespunzătoare lui u_7 având valoarea 14, luând în considerare ponderile asociate liniilor matricii în conformitate cu cele cuprinse în (6.1).Totuși pentru configurația binară pentru ultima coloană este aleasă valoarea 9 (6.1),modalitate în care sinteza schemei rezultă mai economic prin evitarea utilizării a două circuite SAU EXCLUSIV.Este astfel găsită o soluție originală pentru situația particulară a necesității acoperirii unor sisteme care să poată lucra atât pe 8 cât și pe 16 biți.Alegerea celor două coloane având câte 3 unități binare și anume coloanele corespunzătoare u_3 și u_6 având alocate 11 respectiv 7 s-a făcut de așa manieră încât pe lângă corecția erorii singulare, la nivelul celor două grupe de câte 8 biți, să poată fi asigurată detecția numărului maxim posibil de erori duble permisă de către valoarea ce a mai mică a numărului de biți de control care asigură corecția erorii singulare.(4 în cazul considerat). Acest aspect constituie o completare favorabilă a soluției preconizate pentru tolerarea defectării la nivelul acelui modul de memorie.

Constituind modulul cu cea mai mare probabilitate de afectare prin eroare, vom prezenta în cele ce urmează programul de test pentru acesta, care poartă denumirea de test mem. În vederea execuției acestui program este necesară configurația constând din: două unități centrale, MPCB-03 cu monitorul MON86, două module de memorie MPMM-04, două display-uri DAF 2020, 3 funduri

de sertar tip MULTIPROM, 3 surse MPSA-02 din care una tamponabilă şi un modul generator de tact tolerant la defectare, MPCB-05. Programul de test TEST MEM este executat asincron de cele două sisteme de calcul, fiind lansat în execuţie prin tastarea la DAF a comenzii TEST MEM, după care sunt lansate în execuţie programele de autotest.În conformitate cu cele cuprinse în ordinograma din fig.6.8, autotestele folosite constau din:

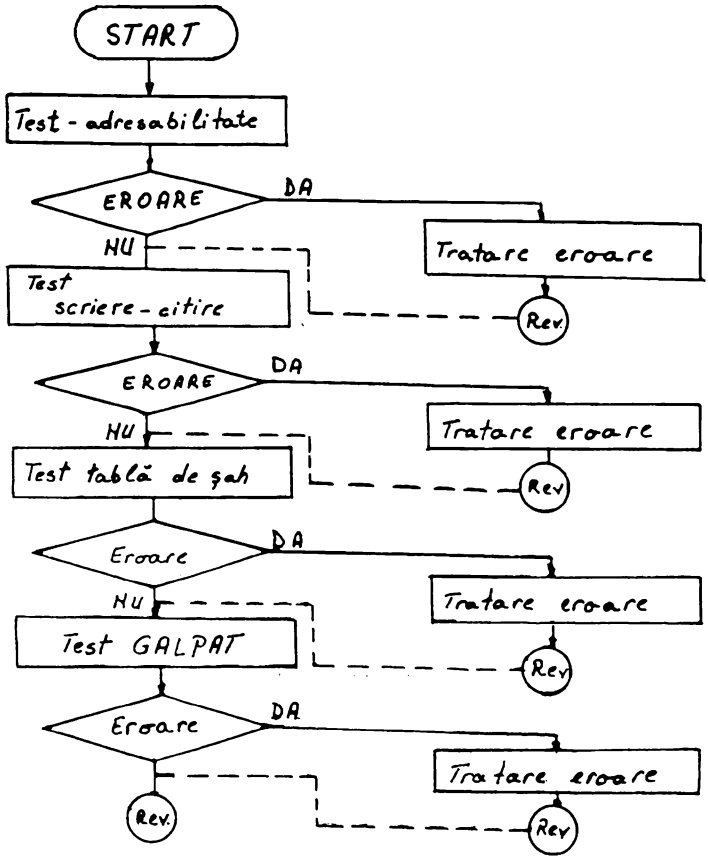


Fig. 6.8

a) Testul de adresabilitate prin intermediul căruia sunt urmărite a fi puse în evidenţă defecte constând din imposibilitatea de a adresa capsule de memorie precum şi din accesul la modulul de memorie pe pinii 6 şi 8 ai circuitului D41.În această ultimă situaţie se verifică semnalele pe pinii 3,4,5 şi respectiv 9,10,11 ai circuitului integrat D41.Testul este explicitat în ordinograma din fig.6.9, utilizând semantica specifică MON86 [Vasi-88] fiind anexat respectivei ordinograme.

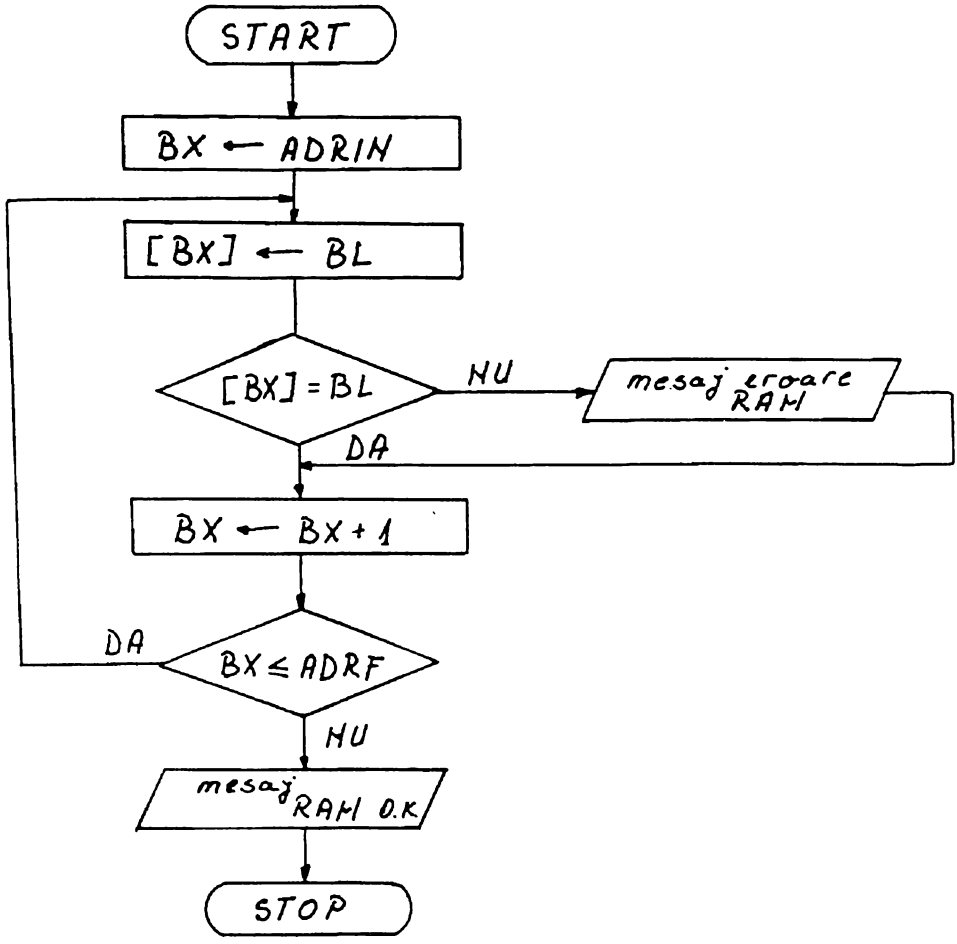


Fig.6.9

b) Testul de scriere/citire menit a detecta biții de memorie blocați în stare de 0 respectiv 1 logic.Testul se execută în conformitate cu cele descrise în ordinograma din fig.6.10 şi este anexat acesteia.

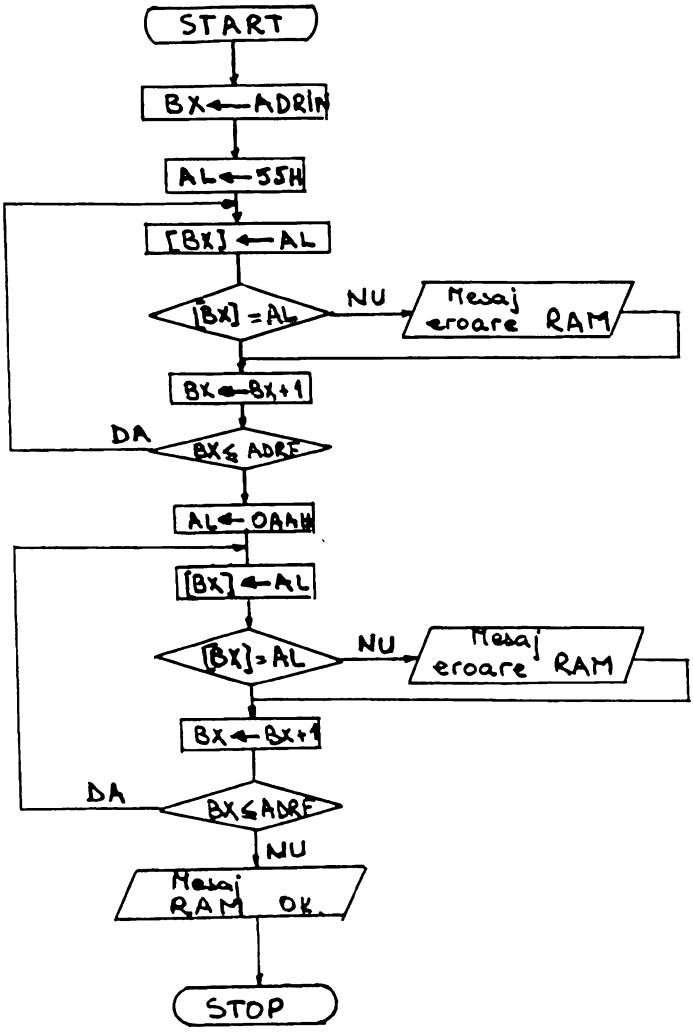


Fig.6.10

c) Testul tablă de şah al cărui algoritm implică înscrierea unei informații de bază în întreaga matrice de memorie, constând din alternări de 0 şi 1, respectiv 1 şi 0. Ulterior încărcării memoriei cu aceste configurații binare, are loc citirea şi setarea fiecărei celule de memorie, mai întâi pe linii şi apoi pe coloane. Ordinograma care stă la baza elaborării programului este prezentată în fig.6.11 căreia îi este apoi ataşat programul de autotest.

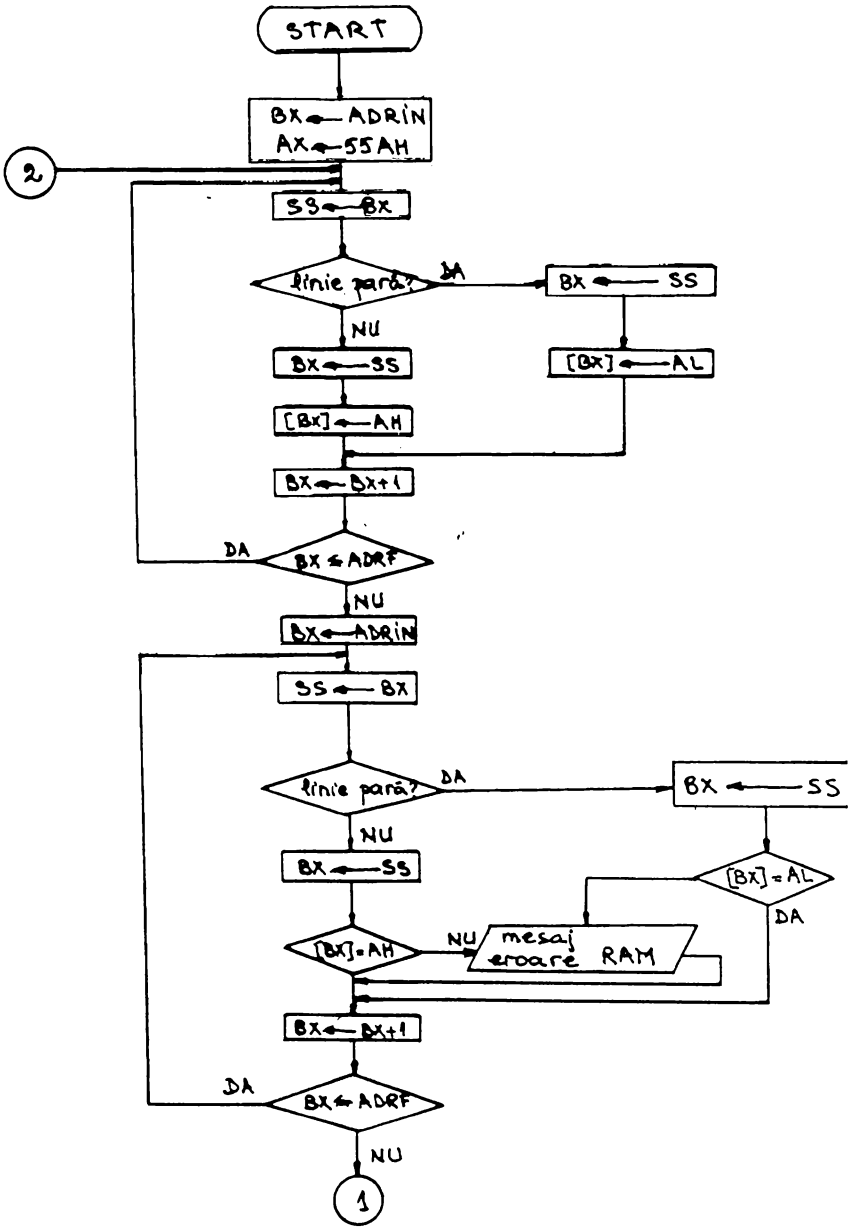


Fig.6.11a

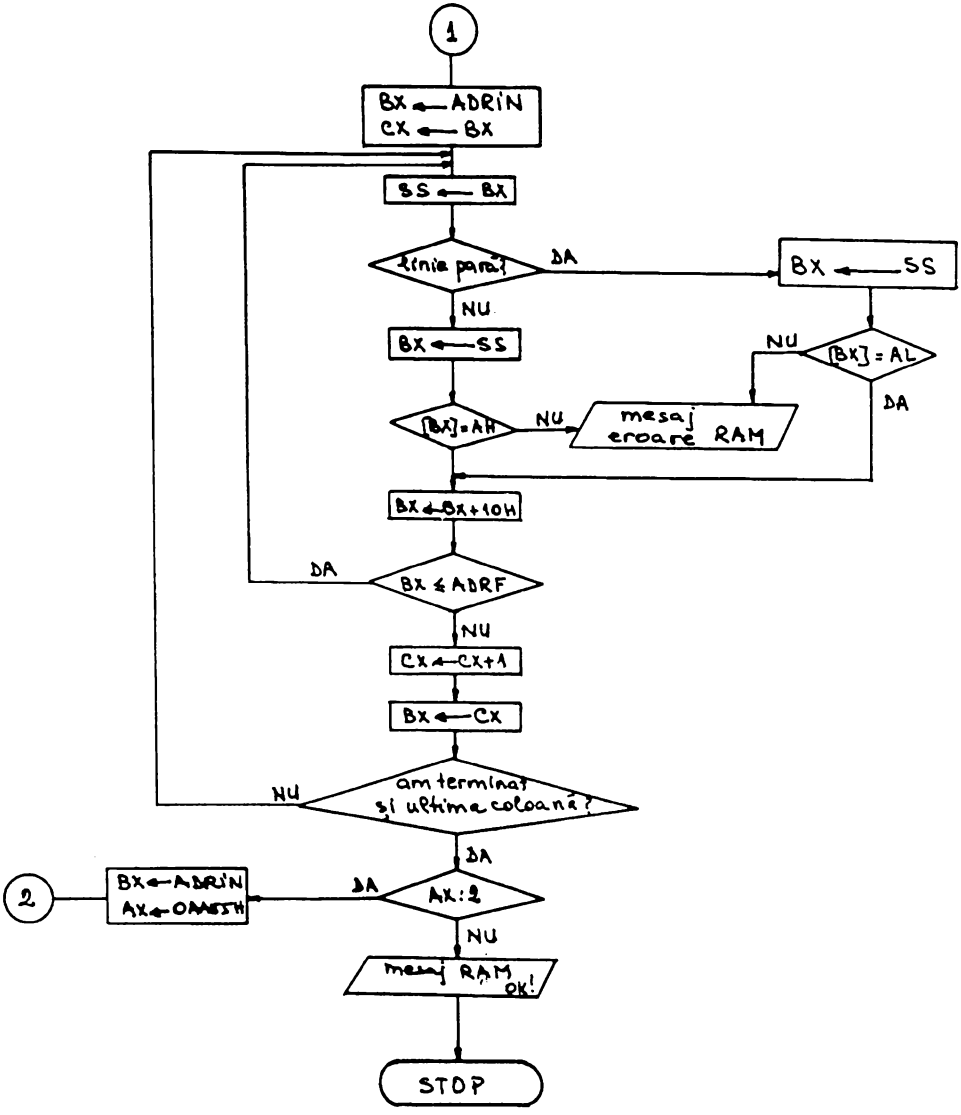


Fig. 6.11b

d) Testul Galpat menit a pune în evidență defecte constând din distrugerea informației din memorie datorită influențelor de cuplaj parazite, incorecta funcționare a amplificatoarelor de citire precum și incorecta selecție a celulei testate. În aceeași semantică uzitată și la celelalte autoteste,

organigrama corespunzătoare acestui segment de autotest este prezentată în fig.6.12 având anexat şi programul scris în limbaj de asamblare 8086.

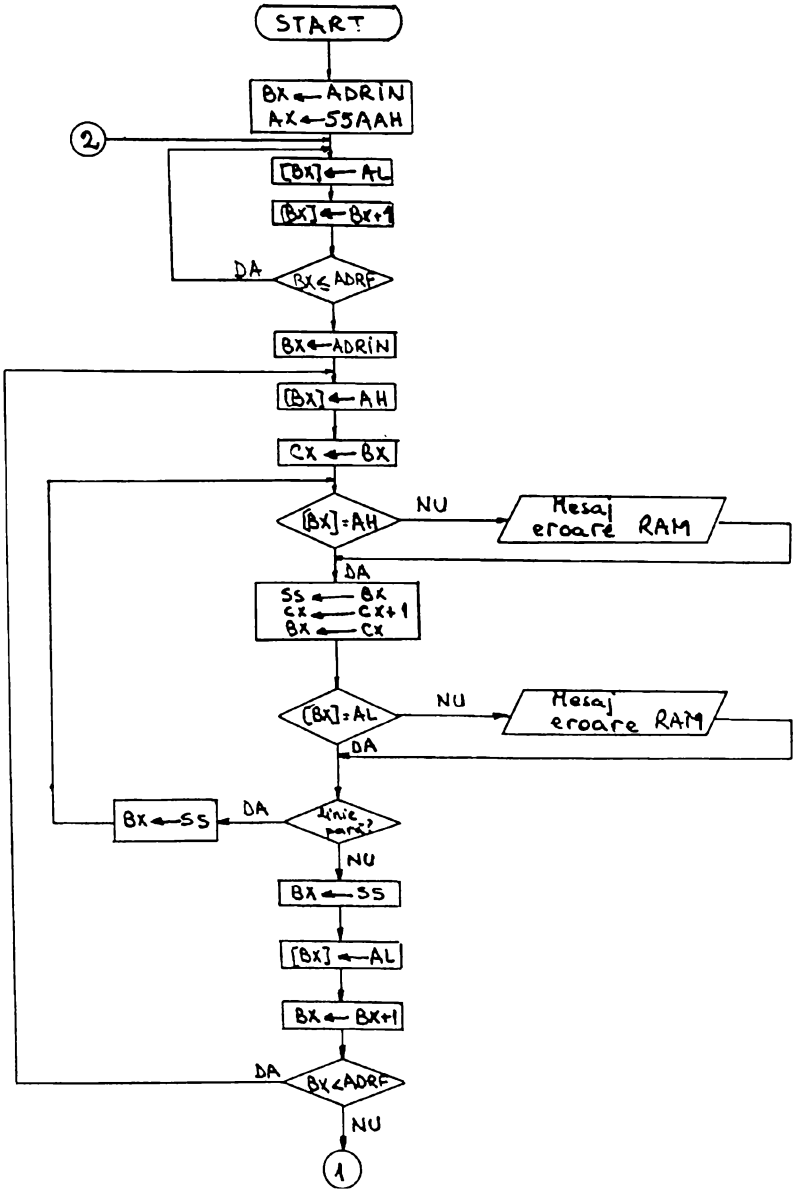


Fig.6.12a

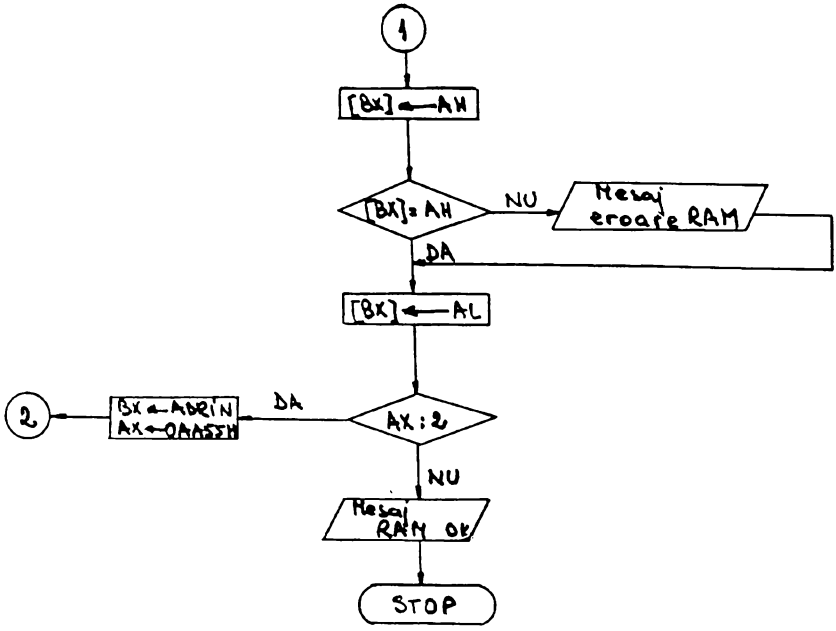


Fig. 6.12b

La terminarea fiecărui segment de autotest de memorie apare unul dintre mesajele "OK" sau "ERROR". Programele de autotest se rulează cu fiecare din cele două sisteme cu care este conectat modulul de memorie, unul dintre sisteme lucrând cu magistrala de sistem a modulului, iar celălalt cu magistrala rezidentă a acestuia.

Fără a pierde din generalitate, în continuare este prevăzută execuția segmentului de program de test funcțional.În acest scop, în memoria RAM a fiecărui sistem, în locația ACTIV cu adresă prestabilită, se încarcă valorile 0, respectiv 1, indicând sistemul care îşi predă datele, respectiv cel care le preia.Pentru sistemul cu fanionul activ pe 1 este afişată pe DAF-ul său a zonei de memorie propie care urmează a fi transferată în memoria comună, sfârşitul transferului fiind marcat prin poziționarea pe 1 a unui alt fanion denumit INDEX.Sistemul cu fanionul pe 0 testează ciclic octetul INDEX şi în momentul în care -l găseşte pe 1 începe preluarea datelor din memoria comună în memoria proprie, afişându-le totodată pe DAF-ul său propriu pentru a putea fi comparate cu cele afişate de celălalt sistem, cel care le-a depus în memoria comună.Ulterior, rolurile celor două sisteme se schimbă.

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE TADR
OBJECT MODULE PLACED IN :F1:TADR.OBJ
ASSEMBLER INVOKED BY:   ASM86 :F1:TADR.ASM


LOC  OBJ                    LINE     SOURCE

  0A00                        1      ADRIN   EQU     0A00H
  0BFF                        2      ADRF    EQU     0BFFH
  00E0                        3      P8251D  EQU     0E0H
  00E2                        4      P8251S  EQU     0E2H
------                        5      DATAS   SEGMENT 'DATA'
0000 0A                       6      TERADR  DB      0AH,0DH,'       OCTET ERONAT LA ADRESA ',24H
0001 0D
0002 094F43544554420
     45524F4E415420
     4C41204144524545
     534120
001A 24
001B 0A                              TOKADR  DB      0AH,0DH,'       TEST ADRESABILITATE REUSIT ',24H
001C 0D
001D 09544553542041
     44524553414249
     4C49544154452O
     52455553495420
0039 24
------                        8      DATAS   ENDS
------                        9      CODS    SEGMENT 'CODE'
                             10              ASSUME  CS:CODS,DS:DATAS,SS:STIVA
                             11      ;
                             12      ;       TEST DE ADRESABILITATE
                             13      ;
0000 BB000A                  14      ST:     MOV     BX,ADRIN
0003 881F                    15      ET1E:   MOV     [BX],BL
0005 381F                    16              CMP     [BX],BL
0007 740B                    17              JE      INE1
0009 B80000                  18              MOV     AX,OFFSET TERADR
000C 8BF0                    19              MOV     SI,AX
000E E81300                  20              CALL    RUT                ;mesaj eroare RAM
0011 E83700                  21              CALL    RUTADR
0014 43                      22      INE1:   INC     BX
0015 81FB000C                23              CMP     BX,ADRF            ;am incarcat toata zona ?
0019 7EE8                    24              JLE     ET1E
001B B81B00                  25              MOV     AX,OFFSET TOKADR
001E 8BF0                    26              MOV     SI,AX
0020 E80100                  27              CALL    RUT                ;mesaj RAM O.K.
0023 CC                      28      TERM:   INT     3
0024                         29      RUT     PROC    NEAR
0024 E4E2                    30      CIT:    IN      AL,P8251S          ;incarc AL cu cuvint stare 8251
0026 8AE0                    31              MOV     AH,AL
0028 80E402                  32              AND     AH,2               ;terminal de date pregatit ?
002B 74F7                    33              JZ      CIT                ;daca nu, salt la CIT
002D E4E0                    34              IN      AL,P8251D          ;incarc AL cu data
002F EB0790                  35              JMP     SCR
0032 8A04                    36      ET11:   MOV     AL,[SI]
0034 3C24                    37              CMP     AL,24H             ;terminare zona ?
0036 740D                    38              JZ      GATA
```

```
LOC  OBJ                 LINE     SOURCE

0038                       39     SCR       LABEL     NEAR
0038 E4E2                  40               IN        AL,P8251S        ;incarc AL cu cuvint stare 8251
003A 2401                  41               AND       AL,1             ;transmisie activata ?
003C 74FA                  42               JZ        SCR              ;daca nu, salt la SCR
003E 8A04                  43               MOV       AL,[SI]
0040 E6E0                  44               OUT       P8251D,AL        ;afisez caracter la consola
0042 46                    45               INC       SI
0043 EBED                  46               JMP       ET11             ;salt la o noua citire
0045 B80000                47     [illegible]
0048 8BF0                  48               MOV       SI,AX
004A C3                    49               RET
                           50     RUT       ENDP
004B                       51     RUTADR    PROC      NEAR
004B 53                    52     ET12:     PUSH      BX
004C 80E7F0                53               AND       BH,0F0H
004F B104                  54               MOV       CL,4
0051 D2EF                  55               SHR       BH,CL
0053 80C730                56               ADD       BH,30H
0056 80FF39                57               CMP       BH,39H
0059 7E03                  58               JLE       ET13
005B 80C707                59               ADD       BH,7
005E BFF001                60     ET13:     MOV       DI,1F0H
0061 883D                  61               MOV       [DI],BH
0063 B700                  62               MOV       BH,0
0065 5B                    63               POP       BX
0066 53                    64               PUSH      BX
0067 80E70F                65               AND       BH,0FH
006A 80C730                66               ADD       BH,30H
006D 80FF39                67               CMP       BH,39H
0070 7E03                  68               JLE       ET14
0072 80C707                69               ADD       BH,7
0075 BFF101                70     ET14:     MOV       DI,1F1H
0078 883D                  71               MOV       [DI],BH
007A B700                  72               MOV       BH,0
007C 5B                    73               POP       BX
007D 53                    74               PUSH      BX
007E 80E3F0                75               AND       BL,0F0H
0081 B104                  76               MOV       CL,4
0083 D2E3                  77               SHL       BL,CL
0085 80C330                78               ADD       BL,30H
0088 80FB39                79               CMP       BL,39H
008B 7E03                  80               JLE       ET15
008D 80C307                81               ADD       BL,7
0090 BFF201                82     ET15:     MOV       DI,1F2H
0093 881D                  83               MOV       [DI],BL
0095 B300                  84               MOV       BL,0
0097 5B                    85               POP       BX
0098 80E30F                86               AND       BL,0FH
009B 80C330                87               ADD       BL,30H
009E 80FB39                88               CMP       BL,39H
00A1 7E03                  89               JLE       ET16
00A3 80C307                90               ADD       BL,7
00A6 BFF301                91     ET16:     MOV       DI,1F3H
00A9 881D                  92               MOV       [DI],BL
00AB B300                  93               MOV       BL,0
```

```
MCS-86 MACRO ASSEMBLER    TADK

LOC  OBJ                  LINE     SOURCE

00AD BFF401                 94             MOV     DI,1F4H
00B0 B324                   95             MOV     BL,24H
00B2 881D                   96             MOV     [DI],BL
00B4 BFF001                 97             MOV     DI,1F0H
00B7 8BC7                   98             MOV     AX,DI
00B9 8BF0                   99             MOV     SI,AX
00BB E866FF                100             CALL    RUT
00BE C3                    101             RET
                           102     RUTADR  ENDP
-----                      103     CODS    ENDS
----                       104     STIVA   SEGMENT
0000 (100                  105             DW      100     DUP(?)
      ????
     )
  0008                     106     VFINIT  EQU     $
----                       107     STIVA   ENDS
  0000                     108             END     ST

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
MCS-86 MACRO ASSEMBLER    TGALP


ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE TGALP
OBJECT MODULE PLACED IN :F1:TGALP.OBJ
ASSEMBLER INVOKED BY:  ASM86 :F1:TGALP.ASM


LOC  OBJ                  LINE     SOURCE

  0A00                      1      ADRIN   EQU     0A00H
  0BFF                      2      ADRF    EQU     0BFFH
  00E0                      3      P8251D  EQU     0E0H
  00E2                      4      P8251C  EQU     0E2H
----                        5      DATAS   SEGMENT 'DATA'
0000 0A                     6      TERADR  DB      0AH,0DH,'        OCTET ERONAT LA ADRESA ',24H
0001 0D
0002 094F4354455420
     45524F4E415420
     4C412041445245
     534120
001A 24
001B 0A                     7      TOKGAL  DB      0AH,0DH,'        TEST GALPAT REUSIT ',24H
001C 0D
001D 09544553542047
     414C5041542052
     455553495420
0031 24
----                        8      DATAS   ENDS
----                        9      CODS    SEGMENT 'CODE'
                           10              ASSUME  CS:CODS,DS:DATAS,SS:STIVA
                           11      ;
                           12      ;       TEST RAM - GALPAT
                           13      ;
0000 BB000A                14      ST:     MOV     BX,ADRIN
0003 B8AA55                15              MOV     AX,55AAH
0006 8807                  16      ET1F:   MOV     [BX],AL
0008 43                    17              INC     BX
0009 81FBFF0B              18              CMP     BX,ADRF
000D 7EF7                  19              JLE     ET1F
000F BB000A                20              MOV     BX,ADRIN
0012 8827                  21      ET2F:   MOV     [BX],AH
0014 8BCB                  22              MOV     CX,BX
0016 3827                  23      ET3F:   CMP     [BX],AH
0018 740D                  24              JE      ET4F
001A 50                    25              PUSH    AX
001B B80000                26              MOV     AX,OFFSET TERADR
001E 8BF0                  27              MOV     SI,AX
0020 E85900                28              CALL    RUT
0023 E87000                29              CALL    RUTADR
0026 58                    30              POP     AX
0027 53                    31      ET4F:   PUSH    BX
0028 41                    32              INC     CX
0029 8BD9                  33              MOV     BX,CX
002B 3807                  34              CMP     [BX],AL
002D 740B                  35              JE      ET5F
002F 50                    36              PUSH    AX
0030 B80000                37              MOV     AX,OFFSET TERADR
0033 8BF0                  38              MOV     SI,AX
0035 E84400                39              CALL    RUT
```

```
MCS-86 MACRO ASSEMBLER    TCALL

LOC  OBJ                  LINE     SOURCE

0038 E86800                 40              CALL    RUTAOR
003B 58                     41              POP     AX
003C 81FBFF08               42      ET3F:   CMP     BX,ADRF
0040 7C00                   43              JL      ET4F
0042 EB0490                 44              JMP     ET7F
0045 5B                     45      ET4F:   POP     BX
0046 EBCE                   46              JMP     ET3F
0048 5B                     47      ET7F:   POP     BX
0049 8807                   48              MOV     [BX],AL
004B 43                     49              INC     BX
004C 81FBFF08               50              CMP     BX,ADRF
0050 7CC0                   51              JL      ET3F
0052 8827                   52              MOV     [BX],AH
0054 3827                   53              CMP     [BX],AH
0056 7408                   54              JE      ET8F
0058 50                     55              PUSH    AX
0059 B80000                 56              MOV     AX,OFFSET TERA0P
005C 8BF0                   57              MOV     SI,AX
005E E81B00                 58              CALL    RUT
0061 E83F00                 59              CALL    RUTADR
0064 58                     60              POP     AX
0065 8807                   61      ET8F:   MOV     [BX],AL
0067 D1F8                   62              SCR     AX,1
0069 7208                   63              JC      OKCAL
006B B80000                 64              MOV     BX,ADRIN
006E B855AA                 65              MOV     AX,0AA55H
0071 EB9C                   66              JMP     ET1F
0073 B81B00                 67      OKCAL:  MOV     AX,OFFSET TOKCAL
0076 8BF0                   68              MOV     SI,AX
0078 E80100                 69              CALL    RUT
007B CC                     70      TERM:   INT     3
007C                        71      RUT     PROC    NEAR
007C E4E2                   72      CIT:    IN      AL,P8251S       ;incarc AL cu cuvint stare 8251
007E 8AE0                   73              MOV     AH,AL
0080 80E402                 74              AND     AH,2            ;terminal de data pregatit ?
0083 74F7                   75              JZ      CIT             ;daca nu, salt la CIT
0085 E4E0                   76              IN      AL,P8251D       ;incarc AL cu data
0087 EB0290                 77              JMP     SCR
008A 8A04                   78      ET11:   MOV     AL,[SI]
008C 3C24                   79              CMP     AL,24H          ;terminare zona ?
008E 7408                   80              JZ      DATA
0090                        81      SCR     LABEL   NEAR
0090 E4E2                   82              IN      AL,P8251S       ;incarc AL cu cuvint stare 8251
0092 2401                   83              AND     AL,1            ;transmisie activata ?
0094 74FA                   84              JZ      SCR             ;daca nu, salt la SCR
0096 8A04                   85              MOV     AL,[SI]
0098 E6E0                   86              OUT     P8251D,AL       ;afisez caracter la consola
009A 46                     87              INC     SI
009B EBF0                   88              JMP     ET11            ;salt la o noua citire
009D B80000                 89      DATA:   MOV     AX,0
00A0 8BF0                   90              MOV     SI,AX
00A2 C3                     91              RET
                            92      RUT     ENDP
00A3                        93      RUTADR  PROC    NEAR
00A3 53                     94      ET13:   PUSH    BX
```

```
MCS-86 MACRO ASSEMBLER    TGALP

LOC  OBJ                  LINE     SOURCE

00A4 80E7F0                 95             AND     BH,0F0H
00A7 B104                   96             MOV     CL,4
00A9 D2EF                   97             SHR     BH,CL
00AB 80C730                 98             ADD     BH,30H
00AE 80FF39                 99             CMP     BH,39H
00B1 7E03                  100             JLE     ET13
00B3 80C707                101             ADD     BH,7
00B6 BFF001                102     ET13:   MOV     DI,1F0H
00B9 8838                  103             MOV     [DI],BH
00BB B700                  104             MOV     BH,0
00BD 58                    105             POP     AX
00BE 53                    106             PUSH    BX
00BF 80E70F                107             AND     BH,0FH
00C2 80C730                108             ADD     BH,30H
00C5 80FF39                109             CMP     BH,39H
00C8 7E03                  110             JLE     ET14
00CA 80C707                111             ADD     BH,7
00CD BFF101                112     ET14:   MOV     DI,1F1H
00D0 8838                  113             MOV     [DI],BH
00D2 B700                  114             MOV     BH,0
00D4 58                    115             POP     AX
00D5 53                    116             PUSH    BX
00D6 80E30F                117             AND     BL,0FH
00D9 B104                  118             MOV     CL,4
00DB D2E3                  119             SHL     BL,CL
00DD 80C330                120             ADD     BL,30H
00E0 80FB39                121             CMP     BL,39H
00E3 7E03                  122             JLE     ET15
00E5 80C307                123             ADD     BL,7
00E8 BFF201                124     ET15:   MOV     DI,1F2H
00EB 881D                  125             MOV     [DI],BL
00ED B300                  126             MOV     BL,0
00EF 58                    127             POP     AX
00F0 80E30F                128             AND     BL,0FH
00F3 80C330                129             ADD     BL,30H
00F6 80FB39                130             CMP     BL,39H
00F9 7E03                  131             JLE     ET16
00FB 80C307                132             ADD     BL,7
00FE BFF301                133     ET16:   MOV     DI,1F3H
0101 881D                  134             MOV     [DI],BL
0103 B300                  135             MOV     BL,0
0105 BFF401                136             MOV     DI,1F4H
0108 B024                  137             MOV     BL,24H
010A 881D                  138             MOV     [DI],BL
010C BFF001                139             MOV     DI,1F0H
010F 8BC7                  140             MOV     AX,DI
0111 8BF0                  141             MOV     SI,AX
0113 E8FFFF                142             CALL    RUN
0116 C3                    143             RET
                           144     RUTAMR  ENDP
                           145     CODS    ENDS
                           146     STIVA   SEGMENT
0000 (100                  147             DW      100     DUP(?)
     ????
     )
```

```
MCS-86 MACRO ASSEMBLER    TCALL

LOC  OBJ                  LINE     SOURCE

  0008                     148     VFINIT  EQU     $
                           149     STIVA   ENDS
  0000                     150             END     ST

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
MCS-86 MACRO ASSEMBLER    TSAH

ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE TSAH
OBJECT MODULE PLACED IN :F1:TSAH.OBJ
ASSEMBLER INVOKED BY:  ASM186 :F1:TSAH.ASM

LOC  OBJ                  LINE     SOURCE

  0A00                       1     ADR1N   EQU     0A00H
  0BFF                       2     ADRF    EQU     0BFFH
  00E0                       3     P8251D  EQU     0E0H
  00E2                       4     P8251S  EQU     0E2H
                             5     DATAS   SEGMENT 'DATA'
0000 0A                      6     TERADR  DB      0AH,0DH,'       OCTET CRONAT LA ADRESA ',24H
0001 0D
0002 024F4354455420
     45524F4E4154204
     4C4120441445245
     534120
001A 24
001B 0A                      7     TOKSAH  DB      0AH,0DH,'       TEST RAM - TABLA DE SAH REUSIT ',24H
001C 0D
001D 0254455354203D2
     414D20D20544
     424C4120444320
     5341482052455
     53495420
002B 24
                             8     DATAS   ENDS
                             9     CODS    SEGMENT 'CODE'
                            10             ASSUME  CS:CODS,DS:DATAS,SS:STIVA
                            11     ;
                            12     ;       TEST RAM - TABLA DE SAH
                            13     ;
0000 BB000A                 14     ST:     MOV     BX,ADR1N
0003 B8AA55                 15             MOV     AX,55AAH
0006 53                     16     ET1C:   PUSH    BX
0007 8AD3                   17             MOV     DL,BL
0009 80E20F                 18             AND     DL,0FH
000C B104                   19             MOV     CL,4
000E D2EB                   20             SHR     BL,CL
0010 80E20F                 21             AND     DL,0FH
0013 02D3                   22             ADD     DL,BL
0015 80E201                 23             AND     DL,1
0018 7410                   24             JZ      ETPAR1          ;daca e linie para, salt la ETPAR1
001A 5B                     25             POP     BX
001B 8827                   26             MOV     [BX],AH         ;daca e linie impara, [BX] <-- 55H
001D 43                     27             INC     BX
001E 81FBFF0B               28             CMP     BX,ADRF         ;gata ?
0022 7EE2                   29             JLE     ET1C
0024 BB000A                 30             MOV     BX,ADR1N
0027 EB0E90                 31             JMP     CITLIN          ;daca da, salt la CITLIN
002A 5B                     32     ETPAR1: POP     BX
002B 8807                   33             MOV     [BX],AL
002D 43                     34             INC     BX
002E 81FBFF0B               35             CMP     BX,ADRF         ;gata ?
0032 7ED2                   36             JLE     ET1C
0034 BB000A                 37             MOV     BX,ADR1N
```

```
MCS-86 MACRO ASSEMBLER    YSAH

LOC  OBJ                  LINE     SOURCE

0037 53                     38     CITLIN: PUSH    BX
0038 8AD3                   39             MOV     DL,BL
003A 80E20F                 40             AND     DL,0FH
003D B104                   41             MOV     CL,4
003F D2EB                   42             SHR     BL,CL
0041 80E30F                 43             AND     BL,0FH
0044 02D3                   44             ADD     DL,BL
0046 80E201                 45             AND     DL,1
0049 7421                   46             JZ      ETPAR2          ;daca da, salt la ETPAR2
004B 5B                     47             POP     BX
004C 3827                   48             CMP     [BX],AH
004E 740D                   49             JE      INBO1           ;daca da, salt  la INBO1
0050 50                     50     ET2O:   PUSH    AX
0051 B80000                 51             MOV     AX,OFFSET TERADR
0054 88F0                   52             MOV     SI,AX
0056 E86400                 53             CALL    RUT
0059 E86A00                 54             CALL    RUTADR
005C 58                     55             POP     AX
005D 43                     56     INBO1:  INC     BX
005E 81FBFF0B               57             CMP     BX,ADRF
0062 7FD3                   58             JLE     CITLIN
0064 BB000A                 59             MOV     BX,ADRIN
0067 88CB                   60             MOV     CX,BX
0069 EB0690                 61             JMP     CITCOL
006C 5B                     62     ETPAR2: POP     BX
006D 3807                   63             CMP     [BX],AL
006F 74EC                   64             JC      INBO1
0071 EBDD                   65             JMP     ET2O
0073 53                     66     CITCOL: PUSH    BX
0074 8AD3                   67             MOV     DL,DL
0076 80E20F                 68             AND     DL,0FH
0079 51                     69             PUSH    CX
007A B104                   70             MOV     CL,4
007C D2EB                   71             SHR     BL,CL
007E 59                     72             POP     CX
007F 80E30F                 73             AND     BL,0FH
0082 02D3                   74             ADD     DL,BL
0084 80E201                 75             AND     DL,1
0087 742A                   76             JE      ETPAR3
0089 5B                     77             POP     BX
008A 3827                   78             CMP     [BX],AH
008C 740D                   79             JE      INBO2
008E 50                     80             PUSH    AX
008F B80000                 81             MOV     AX,OFFSET TERADR
0092 88F0                   82             MOV     SI,AX
0094 E84600                 83             CALL    RUT
0097 E86A00                 84             CALL    RUTADR
009A 58                     85             POP     AX
009B 83C310                 86     INBO2:  ADD     BX,10H
009E 81FBFF0B               87             CMP     BX,ADRF
00A2 7ECF                   88             JLE     CITCOL
00A4 41                     89             INC     CX
00A5 8BD9                   90             MOV     BX,CX
00A7 53                     91             PUSH    BX
00A8 80E30F                 92             AND     BL,0FH
```

```
MCS-86 MACRO ASSEMBLER    TSAH

LOC  OBJ                  LINE     SOURCE

00A8 80FB00                 93              CMP     BL,0
00AB 7417                   94              JE      COMPL
00AD 5B                     95              POP     BX
00AE EBC0                   96              JMP     CITCOL
00B0 5B                     97      ETPAR3: POP     BX
00B1 3807                   98              CMP     [BX],AL
00B3 74E3                   99              JE      IND02
00B5 50                    100              PUSH    AX
00B6 B80000                101              MOV     AX,OFFSET TERADR
00B9 8BF0                  102              MOV     SI,AX
00BB E81C00                103              CALL    RUT
00BE E84000                104              CALL    RUTADR
00C1 58                    105              POP     AX
00C2 EBD4                  106              JMP     IND02
00C4 D1D8                  107      COMPL:  RCR     AX,1
00C6 7209                  108              JC      ETOKG
00C8 BB000A                109              MOV     BX,ADRIN
00CB B855AA                110              MOV     AX,0AA55H
00CE E932FF                111              JMP     ET1G
00D1 B81B00                112      ETOKG:  MOV     AX,OFFSET TOKSAH
00D4 8BF0                  113              MOV     SI,AX
00D6 E80100                114              CALL    RUT
00D9 CC                    115      TERM:   INT     3
00DA                       116      RUT     PROC    NEAR
00DA E4E2                  117      CIT:    IN      AL,PS251S       ;incarc AL cu cuvint stare 8251
00DC 8AE0                  118              MOV     AH,AL
00DE 80E402                119              AND     AH,2            ;terminal de date pregatit ?
00E1 74F7                  120              JZ      CIT             ;daca nu, salt la CIT
00E3 E4E0                  121              IN      AL,PS251D       ;incarc AL cu data
00E5 EB0790                122              JMP     SCR
00E8 8A04                  123      ET11:   MOV     AL,[SI]
00EA 3C24                  124              CMP     AL,24H          ;terminare zona ?
00EC 740D                  125              JZ      GATA
00EE                       126      SCR     LABEL   NEAR
00EE E4E2                  127              IN      AL,PS251S       ;incarc AL cu cuvint stare 8251
00F0 2401                  128              AND     AL,1            ;transmisie activata ?
00F2 74FA                  129              JZ      SCR             ;daca nu, salt la SCR
00F4 8A04                  130              MOV     AL,[SI]
00F6 E6E0                  131              OUT     PS251D,AL       ;afisez caracter la consola
00F8 46                    132              INC     SI
00F9 EBED                  133              JMP     ET11            ;salt la o noua citire
00FB B80000                134      GATA:   MOV     AX,0
0101 8BF0                  135              MOV     SI,AX
0103 C3                    136              RET
                           137      RUT     ENDP
0104                       138      RUTADR  PROC    NEAR
0104 53                    139      ET12:   PUSH    BX
0105 80E7F0                140              AND     BH,0F0H
0108 B104                  141              MOV     CL,4
010A D2EF                  142              SHR     BH,CL
010C 80C730                143              ADD     BH,30H
010F 80FF39                144              CMP     BH,39H
0112 7E03                  145              JLE     ET13
0114 80C707                146              ADD     BH,7
0117 BFF001                147      ET13:   MOV     DI,1F0H
```

```
MCS-86 MACRO ASSEMBLER    TSAH

LOC  OBJ                  LINE     SOURCE

011A 883D                  148              MOV     [DI],BH
011C B700                  149              MOV     BH,0
011E 5B                    150              POP     BX
011F 53                    151              PUSH    BX
0120 80E70F                152              AND     BH,0FH
0123 80C730                153              ADD     BH,30H
0126 80FF39                154              CMP     BH,39H
0129 7E03                  155              JLE     ET14
012B 80C707                156              ADD     BH,7
012E BFF101                157      ET14:   MOV     DI,1F11H
0131 883D                  158              MOV     [DI],BH
0133 B700                  159              MOV     BH,0
0135 5B                    160              POP     BX
0136 53                    161              PUSH    BX
0137 80E3F0                162              AND     BL,0F0H
013A B104                  163              MOV     CL,4
013C D2E3                  164              SHL     BL,CL
013E 80C330                165              ADD     BL,30H
0141 80FB39                166              CMP     BL,39H
0144 7E03                  167              JLE     ET15
0146 80C307                168              ADD     BL,7
0149 BFF201                169      ET15:   MOV     DI,1F2H
014C 881D                  170              MOV     [DI],BL
014E B300                  171              MOV     BL,0
0150 5B                    172              POP     BX
0151 80E30F                173              AND     BL,0FH
0154 80C330                174              ADD     BL,30H
0157 80FB39                175              CMP     BL,39H
015A 7E03                  176              JLE     ET16
015C 80C307                177              ADD     BL,7
015F BFF301                178      ET16:   MOV     DI,1F3H
0162 881D                  179              MOV     [DI],BL
0164 B300                  180              MOV     BL,0
0166 BFF401                181              MOV     DI,1F4H
0169 B024                  182              MOV     BL,24H
016B 881D                  183              MOV     [DI],BL
016D BFF001                184              MOV     DI,1F0H
0170 8BC7                  185              MOV     AX,DI
0172 8BF0                  186              MOV     SI,AX
0174 E866FF                187              CALL    RUT
0177 C3                    188              RET
                           189      RUTADR  ENDP
.....                      190      CODS    ENDS
-----                      191      STIVA   SEGMENT
0000 (100                  192              DW      100     DUP(?)
      ????
      )
  00C8                     193      VFINIT  EQU     $
------                     194      STIVA   ENDS
  0000                     195              END     ST

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
MCS-86 MACRO ASSEMBLER     SCRCIT

ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE SCRCIT
OBJECT MODULE PLACED IN :F1:SCRCIT.OBJ
ASSEMBLER INVOKED BY:  ASMM86 :F1:SCRCIT.ASM

LOC  OBJ                    LINE     SOURCE

  0A00                        1      ADRIN   EQU      0A00H
  0BFF                        2      ADRF    EQU      0BFFFH
  00E0                        3      P8251D  EQU      0E0H
  00E2                        4      P8251S  EQU      0E2H
------                        5      DATAS   SEGMENT 'DATA'
0000 0A                       6      TERRAM  DB       0AH,0DH,'        OCTET ERONAT LA ADRESA ',24H
0001 0D
0002 094F4354455420
     45524F4E415420
     4C412041445245
     534120
001A 24
001B 0A                       7      TOKRAM  DB       0AH,0DH,'        TEST SCRIERE / CITIRE REUSIT ',24H
001C 0D
001D 09544553542053
     43524945524520
     2F204349544952
     45205245555349
     5420
003B 24
----                          8      DATAS   ENDS
------                        9      CODS    SEGMENT 'CODE'
                             10              ASSUME  CS:CODS,DS:DATAS,SS:STIVA
                             11      ;
                             12      ;       TEST DE SCRIERE / CITIRE
                             13      ;
0000 BB000A                  14      ST:     MOV     BX,ADRIN
0003 B055                    15              MOV     AL,55H
0005 8807                    16      ET1:    MOV     [BX],AL          ;incarc octet cu 01010101B ( 55H )
0007 3807                    17              CMP     [BX],AL          ;verific
0009 740B                    18              JE      INB1
000B B80000                  19              MOV     AX,OFFSET TERRAM
000E 8BF0                    20              MOV     SI,AX
0010 E83300                  21              CALL    RUT              ;mesaj eroare RAM
0013 E85700                  22              CALL    RUTADR
0016 43                      23      INB1:   INC     BX
0017 81FB000C                24              CMP     BX,ADRF          ;am incarcat toata zona ?
001B 7EE8                    25              JLE     ET1
001D B0AA                    26              MOV     AL,0AAH
001F BB000A                  27              MOV     BX,ADRIN
0022 8807                    28      ET2:    MOV     [BX],AL          ;incarc octet cu 10101010B ( AAH )
0024 3807                    29              CMP     [BX],AL          ;verific
0026 740B                    30              JE      IND2
0028 B80000                  31              MOV     AX,OFFSET TERRAM
002B 8BF0                    32              MOV     SI,AX
002D E81600                  33              CALL    RUT              ;mesaj eroare RAM
0030 E83A00                  34              CALL    RUTADR
0033 43                      35      INB2:   INC     BX
0034 81FB000C                36              CMP     BX,ADRF          ;am incarcat toata zona ?
0038 7EE8                    37              JLE     ET2
```

```
MCS-86 MACRO ASSEMBLER    SCRCIT

LOC  OBJ                  LINE     SOURCE

003A B81B00                 38             MOV     AX,OFFSET TOKRAM
003D 8BF0                   39             MOV     SI,AX
003F E80400                 40             CALL    RUT                ;mesaj RAM O.K.
0042 E82800                 41             CALL    RUTADR
0045 CC                     42     TERM:   INT     3
0046                        43     RUT     PROC    NEAR
0046 E4E2                   44     CIT:    IN      AL,P8251S          ;incarc AL cu cuvint stare 8251
0048 8AE0                   45             MOV     AH,AL
004A 80E402                 46             AND     AH,2               ;terminal da date pregatit ?
004D 74F7                   47             JZ      CIT                ;daca nu, salt la CIT
004F E4E0                   48             IN      AL,P8251D          ;incarc AL cu data
0051 EB0790                 49             JMP     SCR
0054 8A04                   50     ETI1:   MOV     AL,[SI]
0056 3C24                   51             CMP     AL,24H             ;terminare zona ?
0058 7408                   52             JZ      GATA
005A                        53     SCR     LABEL   NEAR
005A E4E2                   54             IN      AL,P8251S          ;incarc AL cu cuvint stare 8251
005C 2401                   55             AND     AL,1               ;transmisie activata ?
005E 74FA                   56             JZ      SCR                ;daca nu, salt la SCR
0060 8A04                   57             MOV     AL,[SI]
0062 E6E0                   58             OUT     P8251D,AL          ;afisez caracter la consola
0064 46                     59             INC     SI
0065 EBED                   60             JMP     ETI1               ;salt la o noua citire
0067 B80000                 61     GATA:   MOV     AX,0
006A 8BF0                   62             MOV     SI,AX
006C C3                     63             RET
                            64     RUT     ENDP
006D                        65     RUTADR  PROC    NEAR
006D 53                     66     ETI2:   PUSH    BX
006E 80E7F0                 67             AND     BH,0F0H
0071 B104                   68             MOV     CL,4
0073 D2EF                   69             SHR     BH,CL
0075 80C730                 70             ADD     BH,30H
0078 80FF39                 71             CMP     BH,39H
007B 7E03                   72             JLE     ETI3
007D 80C707                 73             ADD     BH,7
0080 BFF001                 74     ETI3:   MOV     DI,1F0H
0083 883D                   75             MOV     [DI],BH
0085 B700                   76             MOV     BH,0
0087 5B                     77             POP     BX
0088 53                     78             PUSH    BX
0089 80E70F                 79             AND     BH,0FH
008C 80C730                 80             ADD     BH,30H
008F 80FF39                 81             CMP     BH,39H
0092 7E03                   82             JLE     ETI4
0094 80C707                 83             ADD     BH,7
0097 BFF101                 84     ETI4:   MOV     DI,1F1H
009A 883D                   85             MOV     [DI],BH
009C B700                   86             MOV     BH,0
009E 5B                     87             POP     BX
009F 53                     88             PUSH    BX
00A0 80E3F0                 89             AND     BL,0F0H
00A3 B104                   90             MOV     CL,4
00A5 D2E3                   91             SHL     BL,CL
00A7 80C330                 92             ADD     BL,30H
```

```
MCS-86 MACRO ASSEMBLER    SCRCIT

LOC  OBJ                  LINE     SOURCE

00AA 80FB39                 93             CMP     BL,39H
00AD 7E03                   94             JLE     ET15
00AF 80C307                 95             ADD     BL,7
00B2 BFF201                 96     ET15:   MOV     DI,1F2H
00B5 881D                   97             MOV     [DI],BL
00B7 B300                   98             MOV     BL,0
00B9 5B                     99             POP     BX
00BA 80E30F                100             AND     BL,0FH
00BD 80C330                101             ADD     BL,30H
00C0 80FB39                102             CMP     BL,39H
00C3 7E03                  103             JLE     ET16
00C5 80C307                104             ADD     BL,7
00C8 BFF301                105     ET16:   MOV     DI,1F3H
00CB 881D                  106             MOV     [DI],BL
00CD B300                  107             MOV     BL,0
00CF BFF401                108             MOV     DI,1F4H
00D2 B324                  109             MOV     BL,24H
00D4 881D                  110             MOV     [DI],BL
00D6 BFF001                111             MOV     DI,1F0H
00D9 8BC7                  112             MOV     AX,DI
00DB 8BF0                  113             MOV     SI,AX
00DD E866FF                114             CALL    OUT
00E0 C3                    115             RET
                           116     RUTADR  ENDP
                           117     CODS    ENDS
                           118     STIVA   SEGMENT
0000 (100                  119             DW      100     DUP(?)
     ????
     )
  00C8                     120     VFINIT  EQU     $
                           121     STIVA   ENDS
  0000                     122             END     ST

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

6.3. Concluzii

Urmărind un parcurs de la modul la sistem (bottom up) prezentul capitol urmăreşte valorificarea practică la nivelul unui sistem tolerant la defecte a modulului ASIM cu reacţii programabile din anteriorul capitol precum şi a deducţiilor din partea teoretică. În baza celor descrise, prezentăm sintetic următoarele elemente originale din acest capitol:

a) Configurarea unui sistem dual sincron cu analiză de semnături programabilă destinat aplicaţiilor industriale critice prin prisma toleranţei la defectare, caracterizat prin:

a_1) O combinare originală a principiilor de redundanţă globală cu redundanţă distribuită, acesta din urmă cuprinzând un nucleu hardware de fiabilitate sporită.

a_2) Stabilirea elementelor de structură optime care să fie atribuită în mod optim la nucleul hardware de fiabilitate sporită.

a_3) Stabilirea pentru fiecare dintre elementele redundante ale nucleului hardware de fiabilitate sporită a soluţiei de redundanţă optimă care să permită implementarea eficientă a toleranţei la defectare.

a_4) Conturarea funcţiilor modulului ASIM cu reacţii programabile menit a permite creşterea capacităţii de trecere pentru programele de autotest, permiţând valorificarea într-o manieră deosebit de eficientă a structurii ASIM obţinută pe baza dedeucţiilor teoretice din capitolul trei şi care permit reducerea maxim posibilă a probabilităţii statistice şi reale de recunoaştere ca funcţional corectă a unei unităţi testate corete.

a_5) Stabilirea secvenţei de o peraţii care se impun întreprinse din momentul apariţiei unei stări de malfuncţionare la unul dintre cele două sisteme de calcul, având drept scop minimizarea intervalului de timp cât procesul se află sub conducerea unui singur sistem.

b) Soluţionarea originală a unui modul hardware de generare a unui tren de tact comun pentru un sistem dual uzitând de principiul redundanţei triple modulare.

c) Conceperea printr-un sistem dual sincron a unui modul hardware care să permită compararea la nivel de bit a semnalelor de pe cele două magistrale de sistem uzitând şi în acest caz de principiul redundanţei triple modulare.

d) Conceperea unui modul de memorie comună pentru sistemul dual sincron având toleranţa de defectare asigurată prin implementarea unui cod corector de eroare singulară şi detector de erori duble, modul caracterizat prin următoarele:

d_1) Soluţionarea originală de implementare a unui cod Hamming modificat pentru o magistrală standard reconfigurabilă (8 sau 16 biţi de date).

d_2) Implementarea pentru situaţia de la punctul d_1 a celui mai economic cod corector al erorii singulare şi detector a numărului maxim de erori duble asigurat de numărul minim al biţilor de control care permite corecţia erorii singulare.

d_3) Conceperea unui sistem de programe de autotest pentru detecţia eficace a funcţionării corecte pentru modulul de memorie corect.

7. Concluzii

Urmărind implementarea eficientă a toleranței la defectare în sisteme de calcul prin elementele redundante constituite de noile structuri ASIM, propuse în lucrare, prezenta teză de dizertație aparține domeniului fiabilității sistemelor de calcul.Aplicația de toleranță la defectare, care include noile structuri redundante,este reprezentată de un sistem dual sincron destinat conducerii de procese industriale.Categoria acestora din urmă, la care se poate aplica sistemul propus, este stabilită prin investigații la nivelul parametrilor reprezentați de latența defectării respectiv latența detecției. Sistemul dual include, pe lângă un nucleu hardware de fiabilitate sporită, un modul cu structura ASIM, caracterizat prin reconfigurabilitate la nivelul sintezei efectuate pe seama mai multor expresii de polinoame generatoare.

Contribuțiile din teza de dizertație au fost expuse, în extenso, prin câte un paragraf distinct, la fiecare din cele șase capitole. Aceste contribuții nu vor mai fi înșiruite, preferându-se o sinteză a lor în acest capitol, succedată de jalonarea unor potențiale extensii ale cercetărilor întreprinse în această lucrare. Teza cuprinde:

1.Contribuții referitoare la clarificarea terminologiei legată de triunghiul cădere-eroare-defect, precum și de latențele de defectare, respectiv de detecție.

2.Contribuții la clasificarea metodelor de izolare-decuplare a defectelor, precum și a celor de implementare hardware a toleranței la defectare.

3.Contribuții la exemplificarea diferitelor tipuri de defectări specifice domeniului calculului, precum și a diferitelor strategii pentru elaborarea produselor program de autotestare.

4.Elaborarea unor structuri de comprimare paralelă noi denumite ASIM și găsite, prin prisma unor parametri, superioare celor cunoscute din literatura de specialitate.

5.Fundamentarea teoretică prin 4 teoreme, 11 leme și 6 proceduri originale a structurilor ASIM precum și a aplicării acestora.

6.Propuneri originale de combinare cu tehnici de scanare pentru facilitarea testării schemelor secvențiale sincrone, de intercalare în scheme PLA cu testare independentă de funcție precum și pentru facilitarea autocontrolului în scheme UAL.

7.Realizarea practică a unui modul hardware cuprinzând o structură ASIM reconfigurabilă la nivelul polinomului generator aparținând bibliotecii standard MULTIPROM.

8.Coordonarea realizării documentației de execuție pentru un sistem dual sincron cu toleranță la defectare pentru conducerea proceselor industriale incluzând un nucleu hardware de fiabilitate sporită, precum și modulul hardware cu o structură ASIM reconfigurabilă.

Viitoare posibile cercetări cu problematica cuprinsă în această lucrare ar putea fi constituite unor noi metode de comprimare paralelă,aplicarea structurilor ASIM în scheme GA (FPGA), precum și configurarea de sisteme bazate pe redundanță superioară celor duale, cum ar fi, spre exemplu, o aplicație de redundanță triplă modulară.

8. Bibliografie

1. [AbAb - 94] Khaled A.I. Abdel - Ghaffar, Amr El Abbadi: *"An Optimal Strategy for Compare File Copies"* IEEE Trans. Parallel Distrib. Syst., vol. 5, no. 1, January 1994, pp. 84 - 93.

2. [AbBr - 86] M.S. Abadir, M.A. Brener: *"Test Schedules for VLSI Circuits Having Built In Test Hardvare"* IEEE Trans. Comput ., vol. C 35, no. 4, April 1986, pp. 361 - 367.

3. [ACTL - 88] V.D. Agrawal, K.I. Cheng, D.D. Johnson, T Lin: *"Designing Circuits with Partial Scan"* IEEE Trans Design & Test vol. 5 pp. 8 - 15 April 1988.

4. [Ah.NG] - 90] A. Ahmad, N.K. Nanda, K. Garg: *"Are Primitive Polinomials Alwais Best in Signature Analysis"* IEEE Design & Test, August 1990, pp. 36 - 38.

5. [Akta - 94] Shakil Ahmad: *"Reability of Knout - of m: 6 systems with Imperfect Fault-Coverage"*IEEE Trans Reability, vol. 43, no. 1, March 1994, pp. 101 - 106.

6. [Amau - 89] Ahmed El-Amaury: *"Comments on Care Redundancy and Masking Improve the Reformance of Sinchroners"* IEEE Trans. Comput., vol. 38, no. 5, May 1989, pp. 750 - 753.

7. [AnWi - 73] J.B. Angell, M.J.I. William: *"Enhancing Testability of Large Scale Integrated Circuits Via test Points And, Additional Logic"* IEEE Trans. Comput., vol. 22, no. 1, pp. 45 - 60, January 1973.

8. [APSD - 94] Magdy S. Abadir, Ashish R. Parikk, Peter A. Sandborn, Ken Drake, Linda Bal: *"Analising Multichips Module Testing Strategies"* IEEE Design & Test of Comp. Spring 1994, pp. 40 - 52.

9. [ArKL - 90] Jean Arlat, Kadema Kanown, Jean-Claude Lapric: *"Dependability Modeling and Evaluation Software Fault - Tolerant Systems"* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 504 - 513.

10. [AvKe - 84 Algirdas Avizienis, John P. Kelly: *"Fault Tolerance by Design Diversity: Concepts and Experiments"* Computer, August 1994, pp. 67 - 80.

11. [BaAl - 86] Prithviraj Bauerjee, Jacob A. Abraham: *"Bound on Algorithm - Based Fault Tolerance in Multiple Procesor System"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp. 296 - 306.

12. [BaBa - 90] Vijay Balasubmanian, Prithviraj Banerjec: *"Compiler Assisted Synthesis of Algorithm - Based Checking in Multiprocesor"* IEEE Trns. Comput., vol. 39, no. 4, April 1990 , pp. 436 - 446.

13. [BaRa - 90] Meera Balakrishnan, C.S. Raghavendra: *"An Reability Modeling of Closed Fault - Tolerant Computer Systems"* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 571 - 575

14. [BaRa - 93] Meera Balakrishnan, C.S. Raghavendra: *"An Analisys of Reability model for Reparaible Fault-Tolerant Systems"* IEEE Trans. Comput., vol. 42, no. 3, March 1993, pp. 327 - 338.

15. [Barb - 92] Ahmed E. Barbour:*"Solutions to the Minimization Problem of Fault Tolerant Logic Circuits"* IEEE Trans. Comput., vol. C 41, no. 4, April 1992, pp. 429 - 442.

16. [BBKP - 94] Kees von Berkel , Ronan Burgess, Joep Kessel, Marly Pronchen, Frits Sehalji, Ad Peeters: *"Asynchronoses Circuits for how Power: A DCC Error Corection"* IEEE Design. & Test of Com. Summer 1994, pp. 22 - 32.

17. [BCSS - 90] James H. Barton, Edard W. Czeck, Zary Z. Segall, P. Sienvorek: *"Fault Injection Experiments Using FIAT"* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 574 - 582.

18. [Beck - 88] Bernd Becker: *"Efficient Testing of Optimal Time Adders"* IEEE Trans. Comput., vol. 37, no. 9, pp. 1113 - 1121, September 1988.

19. [BeTr - 93] Noc Lopez Benitsz, Kishor I. Irivedi: *"Multiprocessor Performability Analysis"* IEEE Trans. Reability, vol. 42, no. 4, Dacember 1993, pp. 579 - 587.

20. [BhSe - 89] Bhargah B. Bahattcharya, Sharad C. Seth: *"Design of Parity Testable Combinational Circuits"* IEEE Trans. Comp., vol. 88, no. 11, pp. 1580 - 1584, November 1989.

21. [BlBJ - 94] Mario Bldum, Jehoshua Bruck, Ludo Tlomizer: *"A Note on a Systematic (12, 8) Code for Corecting Single Errors and Detecting Adjacent Errors"* IEEE Trans. Comput., vol. 43, no. 1, January 1994, pp. 125.

22. [BlMa - 90] Douglas M. Blough, Gerald M.M. Masson:*"Performance Anasysis of a Ceneralized Concurent Error Detection Procedure"* IEEE Trans. Comput., vol. 39, no. 1, January 1990, pp. 47 - 62.

23. [BlPe - 94] Douglas M. Blough, Andrzej Pelc: *"Almost Certain Fault Diagnosis Through Algorithm-Based Fault Tolerance"* IEEE Trans. Parallel Distrib. Systems, vol. 5, no. 5, May 1994, pp. 532 - 539.

24. [BrAH - 90] Yigal Brandman, Alon Arbitsky, John Hennessy: *"A Spectral Lover Tehnique for the Size of Decision Trees and Two Level ANDIOR Circuits"* IEEE Trans. Comput., vol. 39, no. 2, February 1994, pp. 282 - 287.

25. [BrSt 88] Thomas J. Brosman, Noel R. Srader II: *"Modular Error Detection for Bit-Serial Multiplication"* IEEE Trans. Comput., vol. 37, no. 9, pp. 1043 - 1052, September 1988.

26. [Brya - 91] Randal E. Bryant: *"On the Complexity of VLSI Implementations and Graph Reprezentation of Boolean Functions with Application to Integer Multiplication"* IEEE Trans. Comput., vol. 40, no. 2, pp. 205 - 213, February 1991.

27. [BuMi - 94] John Burns, Chris J. Mitchell: *"Parameter Selection for Senver-Aided RSA Computation Schemes"* IEEE Trans. Comput., vol. 43, no. 2, February 1994, pp. 163 - 174.

28. [CăBa - 89] Vasile Cătuneanu, A. Baciarcof: *"Structuri electronice de înaltă fiabilitate"* Ed. Militară Bucureşti, 1989.

29. [CBGC - 94] Diponwita Roy Chowdhury Saugata Basu, Indranie Sen Gupta, Parimal Pal Chandhuri: *"Design of CAECC-Cellular Automata Based Error Correcting Code"* IEEE Trans. Comput., vol. 43, no. 6, June 1994, pp. 759 - 764.

30. [ChAg - 90] Kwang-Ting Cheng, Wishwoary D. Agrawal: *"A Partial Scan Method for Sequential Circuits with Feedback"* IEEE Trans. Comput., vol. 39, no. 4, pp. 544 - 548, April 1990.

31. [ChAK - 90] Kwang-Ting Cheng, Wishwoary D. Agrawal, Ernst S. Kuh: *"A Simulation-Based Method for Generating Tests for Sequential Circuits"* IEEE Trans. Comput., vol.

39, no. 12, pp. 1456 - 1463, December 1990.

32. [ChCh - 91] J.L.G. Chen, T.H. Chen: *"Fault-Tolerant Serial-Parallel Multipler"* IEEE Trans. Procedings, vol. 138, no. 4, pp. 276 - 280, July 1991.

33. [ChFP - 89] Ming-Feng Chang, W. Kent Fuchs, Janak H. Patel: *"Diagnosis and Repair of Memory with Compling Faults"* IEEE Trans. Comput., vol. 38, no. 4, April 1989, pp. 493 - 500.

34. [Chly - 89] Ram Chillarege, Ravic Iyer: *"An Experimental Study of Memory Fault Latency"* IEEE Trans. Comput., vol. C 38, no. 6, pp. 869 - 874, July 1991.

35. [ClDG - 92] Andrea Clematis, Gabriela Dodero, Vittorio Giamuzzi: *"Process Checkpointing Primitives for Fault Tolerance: Definitions and Examples"* Microprocessors and Microsystems vol. 16, no. 1, 1992, pp. 15 - 23.

36. [ClWe - 94] Douglas W. Clark, Lih-Juh Weng: *"Maximal and Near-Maximal Shift Register Sequences: Efficient Event Corentens and Easy Discrete Logarithms"* IEEE Trans. Comput., vol. C 43, no. 5, May 1994, pp. 560 - 568.

37. [CrKS - 88] Gary L. Craing, Charles R. Kime, Kewal K. Saluja: *"Test Scheduling and control for VLSI Build-In Self-Test"* IEEE Trans. Comput., vol. 37, no. 9, September 1988, pp. 1099 - 1109.

38. [DaFC - 89] R. David, A. Fuentes, B. Courtois: *"Random Pattern Versus Deterministic Testing Ram's"* IEEE Trans. Comput., vol. 38, no. 5, May 1989, pp. 637 - 650.

39. [DaFu-90] R. David, A.Fuentes : *"Fault Diagnosis of RAM 's from Random Testing Experiments"* IEEE Trans. Comput., vol. 39, no.2, February 1990, pp.220-229.

40. [DaKa - 89] T. Royu Damarla, M. Karpovsky: *"Fault Detection Combinational Networks by Reed Muller Transforms"* IEEE Trans. Comput., vol. 38, no. 6, Jun 1989, pp. 788 - 797.

41. [Davi - 90] Rene David: *"Signature Analysis for Multiple Output Circuits"* IEEE Trans. Comput., vol. 39, no. 2, February 1994, pp. 287 - 288.

42. [Davi - 90] Rene David: *"Signature Analysis for Multiple Output Circuits"* IEEE Trans. Comput., vol. 39, no. 2, February 1994, pp. 830 - 837.

43. [DaWa - 90] Rene David, Kenneth Wagner: *"Analysis of Detection Probability and Sonie Applications"* IEEE Trans. Comput., vol. 39, no. 10, October 1990, pp. 1284 - 1291.

44. [Dora - 88] R. W. Doran: *"Variants of a Improved Tcrry Look-Ahead Adder"* IEEE Trans. Comput., vol. 37, no. 9, pp. 1110 - 1113 , September 1988.

45. [DuTr-89] Joanne Bechta Dugan, Kishor S.Trivedi: *"Coverage Modelinc for Dependability Analysis of Fault-Tolerant Systems"* IEEE Trans. Comput., vol. 88,no.6,June 1989,pp.775-787.

46. [Echt - 90] K. Echtle: *"Fehlertoleranzverfahren"* Studienbucher Springer Verlag, Heidelberg, 1990.

47. [FeRK - 94] G.L. Feng, T.R.N. Raw, M.S. Kolluru: *"Error Corecting Codes Over Z^{2m} for Algorithm - Based Fault Tolerance"* IEEE Trans. Comput., vol. 43, no. 3, March 1994, pp. 370-374.

48. [Flin - 94] Andrew Flint: *"Testing Multichip Modules"* IEEE Spectrum, March 1994, pp. 59-62.

49. [FoHA - 94] Laurence E. La Forge, Kaiman Huang, Vinod K. Agarwal: *"Almost Sure Diagnosis of Almost Every Good Element"* IEEE Trans. Comput., vol. 43, no. 3, March 1994, pp. 295-305.

50. [Fren - 94] James F. Frenzel: *"Power Supply Current-Diagnosis of VLSI Circuits"* IEEE Trans. Reability, vol. 43, no. 1, March 1994, pp. 30 - 38.

51. [Frie - 86] Arthur D. Friedman: *"Fundamentals of Logic Design and Switching Theory"* Computer Science Press, 1986.

52. [FrSa - 94] Manoj Franklin, Kewal K. Saluja: *"Hypergrph Coloring and Reconfigured RAM Testing"* IEEE Trans. Comput., vol. 43, no. 6, June 1994, pp. 725-736.

53. [Fuji - 90] Hideo Fujiwara: *"Logic Testing and Design for Testability"* The MITPress Cambrige, Massachusetts, London, England 1990.

54. [Gork - 89] Winfried Gorke: *"Fehlertolerante Rechnsysteme"* R. Oldenbourg Verlag Munhen, 1989.

55. [Gork - 91] Winfried Gorke: *"Digitale Fehlerdiagnose"* Institut fur rechnerentwurf und Fehlertoleranz, Universitat Karlshruhe, Vorlesungsskript SS 1991.

56. [Gork - 91] Winfried Gorke: *"Digitale Fehlerdiagnoze"* Vorlesungsskript Universitat Karlshruhe, 1991.

57. [Gros - 89] K.E. Grospietsch: *"Architectures for Testability and Fault-Tolerance in Content-Addressable Systems"* IEEE Proc., vol. C 136, Pt. E, no. 5, September 1989, pp. 366-373.

58. [Gros - 89] K.E. Grospietsch: *"Architectures for Testability and Fault Tolerance in Content-Adresable Systems"* IEEE Proc., vol. C 136, Pt. E, no. 5, September 1989, pp. 366-373.

59. [GuGB - 90] Rajesh Gupta, Melvin A. Breuer: *"The BALLAST Methodology for Structural Scan Design"* IEEE Trans. Comput., vol. .39, no. 4, pp. 538-544, April 1990.

60. [GuRR - 94] Dechang Gu, Daniel J. Rosenkranz, S.S. Ravi: *"Constructtion of Check Sets for Algorithm Based Fault Tolerance"* IEEE Trans. Comput., vol. 43, no. 6, June 1994, pp. 641-650.

61. [HaCl - 93] Hong Hao, Edward J. Me. Clusky: *"Analysis of Gate Oxide Shors in CMOS Circuits"* IEEE Trans. Comput., vol. 42, no. 12, Decmber 1993, pp.1510-1516.

62. [HaNa - 84] I.L. Hakimi, K. Nakajama: *"On Adaptive System Diagnosis"* IEEE Trans. Comput., vol. C 33, no. 3, March 1984, pp. 234-240.

63. [Haye - 88] John P. Hayes: *"Computer Architecture and Organization"* McGraw Hill Book Company, 1988.

64. [Heid - 91] Klaus D. Heidtman: *"Aritmetic Spectrum Applied to Fault Detection for Contribinational Network"* IEEE Trans. Comput., vol. 40, no. 3, March 1991, pp. 320-324.

65. [HeMi - 89] Walt Helbig, Veliko Milutinovic: *"DCFL E/D-MESFET Ga As Experimental RISE Machine"* IEEE Trans. Comput., vol. 38, no. 2, February 1989, pp. 263-273.

66. [HISS - 94] Marius V.A. Hancu Kazuhiko Iwasaki, Yuji Sato, Mamom Sugoie: *"A Concurent Test Architecture for Massively Parallel Computers and its Error Detection Capability"* IEEE Trans. Paralel Distrib. Systems, vol. 5, no. 11, November 1994, pp.

1169-1184.

67. [HKAS - 94] R. Hofmann, R. Klar, B. Mahr, A. Quick, M. Siegle: *"Distributed Performance Monitoring: Methods, Tools and Applications"* IEEE Trans. Paralel Distrib. Systems, vol. 5, no. 6, June 1994, pp. 585-598.

68. [HoLC - 90] Peter D. Hortensius, Robert D. Mc.Leod, Howard C. Card: *"Cellular Automata-Based Signature Analysis for Build-In Self-Test"* IEEE Trans. Comput., vol. 39, no. 10, October 1990, pp. 1237-1285.

69. [Huja - 92] Yemmun Huang Pankaj Jalote: *"Effect of Fault Tolerance on Response Time-Analysis of the Primary Site Approach"* IEEE Trans. Comput., vol. 41, no. 4, April 1992, pp. 420-428.

70. [HuSe - 89] R.V. Hudli, S.C. Selth: *"Testability of Syncronous Sequential Circuits Based on Structure Data"* Proc. Int. Test Conf., pp. 364-372, August 1989.

71. [HwBr - 87] Kai Hwang, Faye A. Briggs: *"Computer Architecture and Parallel Processing"* Mc.Graw Hill International Editions, 1987.

72. [IhLe - 86] Kang Ihun, Yann Hang Lee: *"Measurement and Application of Fault Latency"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp. 370-375.

73. [IyDH - 86] Balakrishna R. Iyer, Lorenzo Donatiallo, Philip Heidelberger: *"Analysis of Performability for Stochastic Models of Fault Tolerant Systems"* IEEE Trans. Comput., vol. C 35, no. 10, October 1986, pp. 902-907.

74. [IyJI - 90] Ravishankar K. Iyer, T. Jowng, P.V. Krishna Iyer: *"Automatic Recognition of Intermittent Failures: An Experimental Study of Field Data"* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 525-537.

75. [IyRo - 86] Ravishankar K. Iyer, David Y. Rossetti: *"A Measurement-Based Model for Workload Dependence of CPU Errors"* IEEE Trans. Comput., vol. C35, no. 6, June 1989, pp. 511-519.

76. [Joha - 93] Rolf Johansson: *"A Class of (12, 8) Codes for Correcting Single Errors and Detecting Double Errors within a Nibble"* IEEE Trans. Comput., vol. 42, no. 12, December 1993, pp. 1504-1506.

77. [John - 89] Barry W. Johnson: *"Design and Analysis of Fault Tolerant Digital Systems"* Addison-Wesley Publishing Company, 1989.

78. [KaLV - 94] Mark G. Karpovaky, Lev B. Levitin, Feodor S. Vainstein: *"Diagnosis by Signature Analysis of Test Responses"* IEEE Trans. Comput., vol. 43, no. 2, February 1994, pp. 141-152.

79. [KaNa - 90] M.G. Karpovsky, P. Nagvajara: *"Optimal Robust Compression of Test Responses"* IEEE Trans. Comput., vol. 39, no.1, January 1990, pp. 138-141.

80. [Kant - 93] Vitit Kantabutra: *"A Recursive Carry-Lookahead/Carry-Selected Hybrid Adder"* IEEE Trans. Comput., vol. 42, no. 12, pp. 1495-1499, December 1993.

81. [KhMa - 95] Shoab A. Khan, Vijak K. Madisetti: *"A Systematic Approach to Early Package Design and Analysis Eases the Multichip Design Challenge and Optimizes Multiple Objective Functions and Costs"* IEEE Design&Test of Comp., Spring 1995.

82. [KiAz - 94] R.M. Kieckhafer, M.H. Azadmanesch: *"Reaching Approximate Agreement with Mixed Mode Faults"* IEEE Trans. Paralel Distributed Syst., vol. 5, no. 1, pp. 53-63, January 1994.

83. [KiLa - 92] K.H. Kim, Thomas F. Laurence: *"Adaptive Fault Tolerance in Complex Real-Time Distibuted Computer System Application"* Computer Communication, vol. 15, no. 4, May 1992, pp. 243-251.

84. [KINK - 94] Shoji Kawahito, Makoto Ishida, Tetsuro Nakamura, Michitaka Kameyama, Tatsuo Higuchi: *"High-Speed Area-Efficient Multipler Design Using Multiple-Valued Current Mode Circuits"* IEEE Trans. Comput., vol. 43, no. 1, pp. 34-42, January 1994.

85. [KiSa - 86] Kozo Kinoshita, Kewal K. Saluja: *"Build-In Testing of Memory Using an On-Chip Compact Testing Scheme"* IEEE Trans. Comput., vol. C 35, no. 10, October 1986, pp. 862-870.

86. [KiWe - 89] K.H. Kim, Howard O. Welch: *"Distributed Execution of Recovery Blocks: On Approach for Uniform Treatment Application Hardware and Software Faults in Real-Time Application"* IEEE Trans. Comput., vol. 38, no. 5, May 1989, pp. 626-636.

87. [KoGr-94] Hermann Kopetz, Günter Grünsteidl: *"TTP-A Protocol for Fault-Tolerant Real-Time Systems"* Computer pp.14-23, January 1994.

88. [Krol-86] Thijs Krol: *"(N,K) Concept Fault Tolerance"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp.339-349.

89. [KrSh-86] C. M. Krishana, Kang G. Shim:*"On Scheduling Takes with a quick Recovery from Failure"* IEEE Trans. Comput., vol. IEEE Trans. Comput., vol. C 35, no. 5, May 1986, pp. 448-455.

90. [KuRe-90] Sandip Kundu Sudhakar M. Reddy:*"Embedded Totally Selfchecking Checkers: A Practical Design"* IEEE Trans. Design & Test of Comp., Aug. 1990, pp. 5-12.

91. [LaHu-90] F. Lambardi, W. K. Huang:*"Fault Detection and Design Complexity in C-Testable VLSI Arrays* "IEEE Trans. Comput., vol. 39, no.12, December 1990, pp. 1477-1481.

92. [Lala-85] Prag K. Lala:*"Fault Tolerant and Fault-Testabe Hardware Design"* Prentice Hall International, 1985.

93. [LeBu-90] Jov-Kang Lee, John T. Butler :*" A Characterization of t/s Diagnosability and Sequential-Diagnosability im Designs"* IEEE Trans. Comput., vol. 39, no. 10, October 1990, pp. 1298-1304.

94. [LeJh-94] Junggu Lee, Kang G. Shin:*"On Probabilistic Diagnosis of Multiprocessor Systems Using Multiple Syndromes"* IEEE Trans. Comput., vol. IEEE Trans. Parallel Distrib. Syst.,vol. 5, no. 6, June 1994, pp. 630-638

95. [LeJO-94] Gueesang Lee, Jane Irwin, Robert Michael Owerns: *"Polynomial Time Testability of Circuits Generated by Input Decomposition"* IEEE Trans. Comput., vol. 43, no. 2, February 1994, pp. 201-210.

96. [LeKr-91] J.H. Lee, C.M. Krishna:*"Optimal Scheduling of Signature Analysis for VLSI Testing"* IEEE Trans. Comput., vol. 40, no.3, March 1991, pp. 336-341.

97. [LeLo-90] Dick Lum Lee, Frederick H. Lochovosky: *"HYTREM-A Hybrid Text-Retriveral Machine for Sarghe Databases"* IEEE Trans. Comput., vol. 39, no.1 , January 1990, pp. 111-123.

98. [LePr-92] D. Lewin, D. Protherse: *"Design of Logic Design"* Chapman & Hall London 1992,pp. 403-455.

99. [LeSa-90] R.Levenge, G. Saucier:*"Optimized Synthesis of Concurrently Checked Controllers"* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 419-425.

100. [LeYa-94] David Lee, Mihalis Yarrakakis: *"Testing Finite-State Machines: State Identification and Verification"* IEEE Trans. Comput., vol. 43, no.3, pp. 306-320, March 1994.

101. [LiCF-89] Chung-Ghi Jimli, Paul Peichnam Chen, W. Kent Frucks:*"Local Concurrent Error Detection and Correction in Data Structures Using Virtual Backpoiners"* IEEE Trans. Comput., vol. 38,no.11, November 1989, pp.1481-1492.

102. [Lily-94] David J. Lilya : *"The Impact of Parallel Loop Scheduling Strategies on Profecching in a Shared Memory Multiprocessor "* IEEE Trans. Parallel Distrib. Syst.,vol. 5, no. 6, June 1994, pp.573-584.

103. [LiSe-89] J. Liu, K.G. Sinh:"Polyhnomial Testing of Packet Switching Networks" IEEE Trans. Comput., vol. 38, no. 2, Febeuary 1989, pp. 202-217.

104. [LiSh-89] Tein-Hsiang Lin, Kang G. Shin:*"Location of a Fault Module in a Computing System"* IEEE Trans. Comput., vol. C 39, no. 2, February 1990, pp. 182-194.

105. [LiSh-90] Tein-Hsiang Lin, Kang G. Shin:*"Location of a Faculty Module in a Computing System"* IEEE Trans. Comput., vol. 39, no. 2, February 1990, pp. 182-194.

106. [MaCl-88] Oamer Mahonoad, E.J. Mc Cluskey: *"Concurent Error Detection Using Watchdog Processon- A Survey"* IEEE Trans. Comput., vol. 37, no. 2, February 1988,pp. 160-174.

107. [Manz-90] Mahomed a Manzoul: *"Fault in Fuzzy Logic Systolic Arrays Cybernetics and Systems"* An International Journal, 21, pp. 513-524, 1990.

108. [MaPa-89] P. Mazunder, J. K. Patel:*"Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random-Access-Memory"* IEEE Trans. Comput., vol. 38, no.3, March 1989, pp. 394-437.

109. [Mark-87] George Markovsky: *"Bounding Probabilities in Combinational Circuits"* IEEE Trans. Comput., vol. C 36, no. 10, October 1987, pp. 1247-1250.

110. [Mazu-93] Ginaki Mazender:*"Deign of Fault-Tolerant Three- Dimensional Dynamic Random-Access Memory with On-Chip Error-Correcting Circuits"* IEEE Trans. Comput., vol. 42,no. 12, pp. 411-420, July 1992.

111. [McMu-86] R.Mc. Murray: *"Multibus Blends Open Arhitecture with Fault-Tolerant Design"* Comp. Tehnol. Rev.,Spring 1986, pp. 61-64.

112. [McSa-88] W. H. Mc. Anney, J. Savir:*"Built-in Checking of the Correct Self Test Signature"* IEEE Trans. Comput., vol. 37, no. 9, September 1998, pp. 1142-1145.

113. [MFGH-94] Wojciech Maly, Derek B.J.Feltham, Anne E. Gattiker, Mark D. Hobaugh, Kenneth Backus, Michael E. Thomas: *"Imart Substrate Multichip-Module Systems"* IEEE Design. & Test of Comp., Summer 1994, pp. 64-75.

114. [MiNi-92] Bruce M.Mc.Millin, Lionel M. Ni: *"Reiable Distributed Sorting Through the Application-Oriented Fault Tolerance Paradingn"* IEEE Trans. Comput., vol. 3, no. 4,Jully 1992, pp. 411-420.

115. [MoAl-89] R.F.Molyneaux, A.Albicki:*"Comments on Ternary Scan Design for VLSI Testability"* IEEE Trans. Comput., vol 38, no. 2, pp. 256-263, February 1989.

116. [Na Ab-90] V.S.S. Nair,Jacob A. Abraham: *"Real-Numbers Codes for Fault-Tolerant Matrix Operations On Processor Arrays"* IEEE Trans. Comput.,vol.39,no.4, pp.426-435, April. 1990.

117. [NaBH-94] Benoit Nadrau - Dostie, Dwayne Burek, Abru S. M. Hassan:*" A Multifrequency Sean-Based BIST Method"* IEEE Design. & Test of Comp. ,Spring 1994, pp. 7-17.

118. [NeCa-87] Victor P. Nelson, Bill D. Caroll: *"Tutorial Fault-Tolerant Computing"* The Computer society of the IEEE, 1987.

119. [Nels-90] Victor P. Nelson: *"Fault Tolerant Computing: Fundamental Concepts"* Computer, July 1990, pp. 19-25.

120. [NeSch-92] Edgar Nett, Ralf Schumann: *"Supporting Fault-Tolerant Distributed Computations under Real-Time Requerements Computer Communications "* vol. 15, no. 4, May 1992, pp. 25-259.

121. [NgMa -94] Spencer W. Ng., Richard L. Mattson - *"Uniform Parity Group Distribution in disk Arrays with Multiple Failures."* IEEE Trans. Comput., vol. 43, no. 4, April 1994, pp. 501-506.

122. [NNHG -93] Victor F. Nicola, Marvin K., Nakayama, Philip Heidelberger, Omibuj Gougal: *"Fast Simulation of Highly Dependante Systems with General Failure and Repair Processes "* IEEE Trans. Comput., no. 12, December 1993, pp. 1140-1452.

123. [NNHG -93] Victor F. Nicola, Marvin K., Nakayama, Philip Heidelberger, Omibuj Gougal: *"Fast Simulation of Highly Dependante Systems with General Failure Processes "* IEEE Trans. Comput., no. 12, pp. 1140-1452, December 1993.

124. [PaHe - 90] David A. Patterson John L. Hennessy: *"Computer Architecture a Quantitative Approach"* Morgan Kaufmann Publisher Inc. 1990.

125. [PaHe -90] David A. Patterson, John L. Hennessy: *"Computer Architecure. A Quantitative Approach"*, Morgan Kaufmann Publisher, 1990.

126. [PaHe -94] David A. Patterson, John L. Hennessy: *"Computer Organization & Design. The Hardware/Software Interface"*, Morgan Kaufmann Publishers San Mateo, California 1994.

127. [PaHK -94] In Cheol Park, Se-Kyoung Hong, Chong-Min Kyung: *"Two Complemientary Approches for Mirocode Bit Optimization"* IEEE Trans. Comput., vol. 43, no. 2, February 1994, pp. 234-239.

128. [Parh -91] Behrooz Parhami: *"Voting Networks"* IEEE Trans. Comput., vol. 40, no. 2, August 1991, pp. 380-393.

129. [Park -90] Behrooz Parhami: *"Generalized Signed-Digit Number System: A Unifying Framework for Redundant Number System"*, IEEE Trans. Comput., vol. 39, no. 1, pp. 89-98, January 1990.

130. [PhVa -94] James Phillip, Stamatis Vassiliadis: *"High-Performance 3-1 Interlock collapsing ALU's"*, IEEE Trans. Comput., vol. 43, no. 3, pp. 257-268, March 1994.

131. [Prad - 86] D.K. Pradhan: *"Fault Tolerant Computing Theory and Techniques"* vol. I, vol. II, Prctice-Hall Englewood Cliffs, New Jersey, 1986.

132. [Prad - 86] D.K. Pradhan: *"Fault Tolerant Computing Theory and Techniques"* vol. I, vol. II, Prentice-Hall Englewood Cliffs, New Jersey 07632, 1986.

133. [Prad -86] D.K. Pradhan: *"Fault Tolerant Computing Theory and Techniques"* vol. I, vol. II, Practice-Hall Englewood Cliffs, New Jersey, 1986.

134. [PrGK -90] Dhiraj K. Pradhan, sandeep K. Gupta, Mark G. Karpovsky: *"Aliasing Probability for Multiple Input Signature Analyser"*, IEEE Trans. Comput., vol. 39, no. 4, April 1990 pp. 586-591.

135. [PrMC - 67] F.P. Preparta, G.Metze, R.T. Chien: *"On the Connection Asigment Problem of Diagnosable Systems"* IEEE Trans. Electron. Comput., vol. EC 16, Dec. 1967, pp. 848.

136. [RaFM -90] Sampath Rangarayan, Donald Fussell Miroslav Malek: *"Built - Im Testing of Integrated Circuit Wafers"* IEEE Trans. Comput., vol. 39, no. 2, February 1994, pp. 195-205.

137. [RaFu -89] F.R.N. Rao, E. Fujiwara: *"Error-Control Coding for Computer Systems"* vol. I, vol. II, Prentice-Hall International, Inc., 1989.

138. [RaFu -89] T.R.N. Ra. E. Fujiwara: *"Error Control Coding for Computer Systems"* vol. I, vol.II, Prentice-Hall International, Inc., 1989.

139. [RaFuji-89] T.R.N. Rao, E. Fujiwara: *"Error Control Coding for Computer Systems"* vol.I, vol.II, Prentice-Hall International, Inc.,1989.

140. [RaKS -90] P. Ramarathan, dilip Kandtlur, Kang G. Shin: *"Hardware - Assisted Software Clock Synchronization for Homogenous Distribulated Systems "* IEEE Trans. Comput., vol. 39, no. 4, April 1990, pp. 514-524.

141. [RaKS -90] R. Ramanthan, D. Kandlus, Kang G. Shin: *"Hardware-Assisted Software Clock Synchronization for Distributed System"*IEEE Trans. Comput.,vol.39, pp.514-524,April 1990.

142. [RaTy -93] Janusz Rajski, Jerzy Tyszer: *Recursive Pseudoexhaustive Test Pattern Generation"* IEEE Trans. Comput., vol. 42, no. 12, December 1993, pp. 1517-1521.

143. [ReJh-94] Jennifer Rexford, Niraj K. Jha: *"Partitioned Encoding Schemes for Algorithm - Based Fault Tolerance in Massively Parallel Systems"* IEEE Trans. Parallel Distrib. Syst., vol. 5, no.6, June 1994, pp.649-653.

144. [Renn-84] David A. Rennels: *"Fault-Tolerant Computing-Concept and Exemmples"* IEEE Trans. Comput., vol. C 33, no.12, December 1984, pp.1116-1129.

145. [ReSK-88] Shudhakar M. Reddy, Kewal K. Saluja, Mark G.Karpovsky: *"A Data Compression Tehnique for Built-In Self-Test"* IEEE Trans. Comput., vol. 37,no.9, September 1988, pp.1154-1156.

146. [RhSt-86] Tim Rhyne, Noel r. STRADER: *"A Signed Bit-Sequential Multiplier"* IEEE Trans. Comput., vol. 35, no.10, pp. 896-901, October 1986.

147. [RKCL-85] G. Russel, D. Kimment, G. Cluster, M.R.Mc. Lauchlan:*CAD for VLSI"* Van Nestrand Reinhold (UK), 1985.

148. [SaAV-87] Jacob Savim, William Mc. Ammey, Salvatore R. Vechio :*"Fault Propagation Though Embledded Multiport Memories"* IEEE Trans. Comput., vol. C 36, no. 5, May 1987, pp. 592-602.

149. [SaBe-90] Isutomu Sasao, Philipp Besslich: *"On the Complexity of Mod-2 Sun PLA'S"* IEEE Trans. Comput., vol. 39, no. 2, February 1990, pp. 262-266.

150. [SaDa-86] K.K.Saluja, R. Dandapani: *"An Alternative to Scan Design Methods for*

Sequential Machines" IEEE Trans. Comput., vol. c 35, no. 4, pp.384-388, April 1986.

151. [SaMC-90] Nirmal R. Saxene, J. Mc. Cluskey: *"Control-Flow Checking Using Watchdog Assists and Extended-Precision Checksums"* IEEE Trans. Comput., vol. 39, April 1990, pp. 586-591.

152. [SaRo-86] Nirmal R. Saxene, John P. Robinson: *"Accumulator Compression Testing"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp.317-321.

153. [ScG-92] Detlef Schmidt, Winfried Görke: *"Rechnerstrukturen"* Universität Karlshruhe, Fakultät für Informatik, Institut für Rechnerentwurf und Fehlertoleranz, 1992, 3 Auflabe.

154. [Schu-93] Horst-Dieter Schulze: *"Asigurarea calitatii- cheia spre piața europeană "* Comunicare la inaugurarea reprezentanței RWTUV din Timişoara Universitatea Tehnică Timişoara, 1993.

155. [ScSh-87] M. Schuette, J. Shen: *"Processor Control Flow Monitoring Using Signatured Instruction Streams"* IEEE Trans. Comput., vol. 36, no. 3, March 1987, pp.264-276.

156. [Se Mu-87] M.Serra, J. C. Muzio:*"Testing Programable Logic Arrays by Sun of Syndromes"* IEEE Trans. Comput., vol. C 36, no. 9, September 1987, pp. 1097-1101.

157. [SeAF-90] Sharad C. Seth, Vishwani D.Ograwal, Hassan Farhat: *"A Statistical Theory of Digital Circuit Testability"* IEEE Trans. Comput., vol. C 39,no.4, April 1990, pp.582-586.

158. [SeK -94] Michael A. Senette, John P. Then: *"Explointing Instructions Level Parallelism for Integrated Control Flow Monitoring"* IEEE Trans. Comput., vol. 43, no.2, February 1994, pp. 129-140.

159. [SeSh-87] Michael A.Schuette, John Paul Shen: *"Processor Control Flow Monitoring Using Signatured Instruction Streams"* IEEE Trans. Comput., vol. C 36, no. 3, March 1987, pp. 264-275.

160. [SFKK-93] C.H. Stapper, J. A. Fifield, H. L. Katter,W. A. Klassen:*"High-Reability 16-M Bit Memory Chip"* IEEE Trans. Reability, vol. 42, no. 4, December 1993, pp. 596-603.

161. [ShGS-89] K. Lentaro Shmizu, Eüch Goto, Shimichi Ichikawa:*" CPC (Cyclic Pipeline Computer) - An Architecture Suited for Josephson and Pipelined-Memory Machines"* IEEE Trans. Comput., vol. 38, no. 6, June 1989, pp. 825-832.

162. [ShLi-88] Kang G. Shin, Tein-Hsiang Lee:*"Modeling and Measurement of Error Propagation in a Multimodule Computing System"* IEEE Trans. Comput., vol. 37, no. 9, September 1988, pp. 1053-1066.

163. [Sohi-89] Gurindar S. Sohi: *" Cache Memory Organization to Enchange the Yield of High-Performance VLSI Processors"* IEEE Trans. Comput., vol. 38, no.4, April 1989, pp. 484-492.

164. [Sons-94] Janusz Sosmovoski:*"Trasient Fault Tolerance in Digital System"* IEEE Micro, February 1994, pp.24-35.

165. [Sons-94] Janusz Sosmowski:*"Transient Fault-Tolerance in Digital Systems"* IEEE Micro, February 1994, pp. 24-35.

166. [SpSW-91] R. A. Spraque, K. J. Singh, R. T. Wood:*"Concurent Engineering in Product Development "* IEEE Desing & Test of Comput.,March 1991, pp.6-13.

167. [StLe-92] Charles H.Stapper, Hsiang-San Lee:*"Synergistic Fault Tolerance for Memory Chips"* IEEE Trans. Comput., vol. 41, no. 9, September 1992, pp. 1078-1087.

168. [SuYo-92] Chang Yup Sung, Byung Kyu Yoo: *"Simple Enumeration of Minimal Cutsets Separating 2 Vertices in a Claas of Unidirected Planar Graphs"* IEEE Trans. Reability, vol. 41, no. 1, March 1992, pp. 63-71.

169. [tANC-90] Andrew S. Tanenbaum: *"Structural Computer O rganisation" Pretice-Hall International Editions, 1990.*

170. [TaSe-86] David J. Taylor, Care-John Seger: *"Robust Storage Structures for Crash Recovery"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp. 288-295.

171. [ThDa-83] P. Thevenoud-Fosse, R. David: "Random Testing of Control Section of a Microprocessor" Dig. 13th Annu. Int. Symp. Fault-Tolerant Comput., Milan, Italy, June 1983, pp.366-373.

172. [Toka-94] Alice M. Tokarnla: *"International Minimal Shift Contens: a Searchm Tehnique"* IEEE Trans. Comput., vol. 43, no.5, May 1994, pp.633-639.

173. [TrAF-87] Robert Theur, Vinad K., Ogarwal Hideo Fujiwara: *"A New Bult-in Self Test Design for Pla'S with High Fault Coverage and Low-Overhead"* IEEE Trans. Comput., vol. C 36, no. 3, March 1987, pp. 369-373.

174. [UpRa-94] Shamblu J. Upadhyaya, Bina Ramamurthy: *"Concurent Process Monotoring with No Reference Signaturos"* IEEE Trans. Comput., vol. 43, no. 4, April 1994, pp. 475-480.

175. [Va - 89] Luminița Vasiu: *"Maximizarea numărului de erori detectate la o memorie prevăzută cu corecția erorii singulare"* Simpozionul de Tehnologie şI echipamente pentru Testare Automată, Cluj-Napoca; 5-7 Septembrie 1989.

176. [VaCh-92] Amujan Varna, Suresh Chalasani: *"Fault-Tolerance Analysis of One-Sided Crosspoint Switching Networks"* IEEE Trans. Comput., vol. 41, no. 2, February 1992, pp. 143-153.

177. [Vasi - 89 a] Luminița Vasiu (coordonator): *"Sistem dual sincron cu analiză de semnătură pentru sisteme tolerante la defecte"* Contract IPA, nr. 6965/89, Faza model experimental, 54 pag., vol. II.

178. [Vasi - 93 a] Luminița Vasiu (coordonator): *"Stadiu actial de implementare a toleranței la defectare în sisteme de calcul"* Referat de doctorat, 1993, Universitatea Tehnică Timişoara.

179. [Vasi - 94 a] Luminița Vasiu (coordonator): *"Creşterea capacitații de trecere a programelor de autotestare"* Referat de doctorat, 1994, Universitatea Tehnică Timişoara.

180. [Vasi - 94 b] Luminița Vasiu (coordonator): *"Configurarea nucleului hardware de fiabilitate sporită pentru echipamente de clacul la defecte"* Referat de doctorat, 1994, Universitatea Tehnică Timişoara.

181. [Vasi 88 g] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, GENERATOR DE TACT LA DEFECTE, MPCB-05"*, Faza Documentație de execuție, Contract IPA nr. 7407/88, Simbol 77710, 38 pag. vol. I.

182. [Vasi-87 a] Luminita Vasiu (coordonator): *"Contributii la elaborarea unui modul MULTIPROM destinat supravegherii bazate pe principiul analizei de semnături a fluxurilor informationale de pe magistrală"*, Lucrare elaborată pt. obținerea gradului de cercetător ştiințific, 1987, 27 pag.

183. [Vasi-87 a] Luminita Vasiu (coordonator): *"Sistem tipizat de module şi subansamble pentru echipamente numerice automatizări - ANALIZOR DE SEMNĂTURĂ, MPCB-04"*, Faza Proiect Tehnic, Contract IPA nr. 7149/87, 38 pag.

184. [Vasi-87 b] Luminita Vasiu (coordonator): *"Sistem tipizat de module şI subansamble pentru echipamente numerice automatizări -ANALIZOR DE SEMNĂTURA, MPCB-04"*, Faza Documentatie de executie, Contract IPA nr. 7149/87, 30 pag.

185. [Vasi-87 d] Luminița Vasiu (coordonator): *"Echipament şi sistem de programare pentru realizarea de sisteme de calcul universale de proces ECAROM 886 cu toleranță la defecte"* , Contract IPA, nr. 7149/87, Faza Proiect tehnic, 39 pag.

186. [Vasi-87 e] Luminita Vasiu (coordonator): *Echipament şi sistem de programare pentru realizarea de sisteme de calcul universale de proces ECAROM 886 cu toleranță la defecte"*, Contract IPA, nr.7149/87, Faza Documentație de execuție ,Simbol 76710, 69 pag. vol II.

187. [Vasi-88 a] Luminița Vasiu (coordonator): *"Sistem tipizat de module şi subansamble pentru echipamente numerice automatizare - MEMORIE COMUNĂ MPCB-06"*, Fază Proiect Tehnic, Contract IPA nr. 7407/88, simbol 76730, 35 pag.

188. [Vasi-88 b] Luminița Vasiu (coordonator): *"Sistem tipizat de module şi subansamble pentru echipamente numerice automatizare - MEMORIE COMUNĂ, MPCB-06"*, Faza Documentație de Execuție, Contract IPA nr. 7407/88, Simbol 76730, 52 pag. vol. I.

189. [Vasi-88 c] Luminița Vasiu: *"Unele contribuții la soluționarea tolerantă a defecțiunilor a creşterii fiabilitații şi disponibilității conducerii cu calculatorul a procesului industriale"* Lucrare elaborată pentru pbținerea gradului de cercetător științific CP III, Timişoara 1988, 64 pag.

190. [Vasi-88 d] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, COMPARATOR TOLERANT LA DEFECTARE MPCB-07"*, Faza Proiect Tehnic, Contract IPA nr. 7407/88, Simbol 77740, 38 pag.

191. [Vasi-88 d] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, COMPARATOR TOLERANT LA DEFECTARE MPCB-07"*, Faza Documentație de execuție, Contract IPA nr. 7407/88, Simbol 77730, 61 pag. vol. I.

192. [Vasi-88 f] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, GENERATOR DE TACT LA DEFECTE, MPCB-05"*, Faza Proiect tehnic, Contract IPA nr. 7407/88, Simbol 76720, pag. 34.

193. [Vasi-88 g] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, PROGRAME PENTRU DETECTAREA, IZOLAREA ŞI RECUPERAREA MODULELOR DEFECTE, DETMO-TD"*, Faza: Elaborare programare, Contract IPA nr. 7407/88, Simbol 77800, vol. I, 51 pag.

194. [Vasi-88 g] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare, PROGRAME PENTRU DETECTAREA, IZOLAREA ŞI RECUPERAREA MODULELOR DEFECTE, DETMO-TD"*, Faza: Validare programare, Contract IPA nr. 7407/88, Simbol 77800, vol. I, 62 pag.

195. [Vasi-88 k] Luminița Vasiu (coordonator): *"Sistem de module şi subansamble pentru echipamente numerice de automatizare. MEDIU DE SIMULARE PENTRU APLICAŢII TIMP-REAL TOLERANTE LA DEFECTE, MESIM-TD"*, Faza: Elaborare programare, Contract IPA nr. 7407/88, Simbol 76800, 68 pag. vol. II.

196. [Vasi-88 l] Luminița Vasiu (coordonator): *"Sistem dual sincron cu analiză de semnături pentru sisteme tolerante la defecte"*, Contract IPA, nr. 6965/87, Faza Proiect tehnic, Dc. 85240 67 pag.

197. [Vasi-88 m] Luminița Vasiu (coordonator): *"Sistem dual sincron cu analiză de semnătură pentru sisteme tolerante la defecte"*, Contract IPA, nr. 6965/87, Faza Documentație de execuție, Contract IPA nr. 7407/88, Simbol 85240, 65 pag. vol. III.

198. [VaVV-87] Luminița Vasiu, M. Vladuțiu, R. Vasiu: *"Asupra utilizării metodei analizei de semnături la realizarea de echipamente de calcul cu toleranță la defectare"*. Simpozionul National de Tehnologie Electronică şi Fiabilitate,Băile Herculane, 29-31 octombrie 1987, pp.241-244.

199. [VaVV-87] Luminița Vasiu, M. Vlăduțiu, R. Vasiu: *"Modul de supraveghere a funcționării echipamentelor configurabile pe magistralele MULTIPROM bazat pe principiul analizei de semnături"*. Buletinul "Realizări şi percepție ale activității IPA-TIB 1987, pp. 24-48".

200. [VaVV-87] Luminița Vasiu, M. Vlăduțiu, R. Vasiu: *"Strategii de testare cu modulul analizor de semnături"*. Sesiunea de comunicări ştiințifice TEHNIC 2000, Timişoara, 15-16 mai 1987, pp. 141-147.

201. [VaVVI-88] Luminița Vasiu, M. Vlăduțiu, M. Vlad: *"Aplicarea reductanței triple modulare la generarea tactului şi compararea fluxurilor informaționale pentru un sistem dual sincron tolerant la defecțiuni"*. Simpozionul de Tehnologie şi Echipamente pentru Testare Automată, Cluj-Napoca; 4-5 oct. 1988, pp. 72-79.

202. [VaVVI-88] Luminița Vasiu, M. Vlăduțiu, M. Vlad: *"Conceperea nucleului hardware de fiabilitate sporită pentru un sistem dual sincron tolerant la defecțiuni"*. Premiul II la concursul de creație tehnică, Timişoara, 1988.

203. [VaVVI-88] Luminița Vasiu, M. Vlăduțiu, M. Vlad: *"Echipament dual sincron tolerant la defecțiuni"*. Simpozionul de Calculatoare şi Conducere Automată a proceselor industriale, Institutul Politehnic Timişoara; 8-10 dec. 1988.

204. [ViRa - 94] Bapirayu Vinnakota, V.V. Bapeswara Rao: *"Generation All Read-Muller Expansions of a Switching Functions"* IEEE Trans. Comput., vol. 43, no.1, January 1994, pp. 122-124.

205. [Vlăd - 86] Mircea Vlăduțiu: *"Tehnica testări echipamentelor de calcul şI evaluarea performanțelor"* Lit. IPTV Timişoara 1986.

206. [Vlăd - 88] : *"Testarea prim comprimare bazată pe coduri cu resturi"* Simpoz. THETA'88 pag 80-85.

207. [VlCr - 90] Mircea Vlăduțiu, Marius Crişan: *"Tehnica testării echipamentelor de prelucrare a datelor"* Ed. Facla, Timişoara, 1990.

208. [Vsi-88 j] Luminita Vasiu (coordonator): *Sistem de module şi subansamble pentru echipamente numerice de automatizare, MEDIU DE SIMULARE PENTRU APLICAȚII TIMP-REAL TOLERANTE LA DEFECTE, MESIM-TD"*, Faza: Elaborare programare, Contract IPA nr. 7407/88, Simbol 76800, 58 pag. vol. I.

209. [VVTD-88] Luminița Vasiu, M. Vladuțiu, R. Todosie, C. Dragomirescu: *"Produs program destinat testării unui sistem dual sincron tolerant la defecțiunii"*. Simpozionul de Tehnologie şi Echipamente pentru Testare Automată, Cluj-Napoca; 4-5 oct. 1988, pp. 122-126.

210. [VVVa-87] Luminița Vasiu, M. Vlăduțiu, R. Vasiu, N. Albu: *"Modul de testare realizat pe principiul analizei de semnături"*. Sesiunea de comunicări științifice TEHNIC 2000, Timișoara, 15-16 mai 1987, pp. 136-140.

211. [VVVa-87] Luminița Vasiu, M. Vlăduțiu, R. Vasiu, N. Albu: *"Posibilități de realizare a unui sistem de calcul cu toleranță la defectare"*. Simpozionul Național de Tehnologie Electronică și Fiabilitate, Băile Herculane,29-31 octombrie 1987, pp. 254-247.

212. [VVVa-87] Luminița Vasiu, M. Vlăduțiu, R. Vasiu, N. Albu: *"Sistem de calculaționare puternice și tolerante la defecte în cadrul tehnicii de conducere a proceselor industriale"*. Simpozionul Național de Tehnologie Electronică și Fiabilitate, Băile Herculane,29-31 octombrie 1987.

213. [VVVI-86] Luminita Vasiu, M. Vladutiu, R. Vasiu, T. Isuica: *"Asupra activitatii modulare in testarea nucleului hardware al microsistemelor"*. Simpoziomul National SICA 86, Aprilie 86 pp. 132-139.

214. [VVVI-86] Luminita Vasiu, M. Vladutiu, T. Ionica: *"Modalitati de testare a microprocesorului I. 8086"*. Simpoziomul de Tehnologie si Echipamente pentru Testare Automata, Cluj-Napoca; Octombrie 1986, pp. 117-119.

215. [WaCC-87] Kenneth D. Wagner, Carry K. Chin, Eduard J. Mc.Clustkey: *"Pseudorandom Testing"* IEEE Trans. Comput., vol. C 36, no. 3, March 1987, pp. 332-343.

216. [WaCl-86] Luang-Terng Wang, Eduard J. Mc.Cluskey: *"Condensed Linear Feedback Shift Register (LFSR) Testing - A Pseudoexhaustive Test Technique"* IEEE Trans. Comput., vol. C 35, no. 4, April 1986, pp. 367-370.

217. [WaPe-90] Charles C. Wang, Dingyi Pei: *"A VLSI Design for Computering Exponentations in GF (2^m) and its Application to Generate Pseudorandom Number Sequences"* IEEE Trans. Comput., vol. 39, no. 2, February 1990, pp. 258-262.

218. [WiJR-94] M. Wigley, Graham A. Jullien, Daniel Reaume: *"Large Dynamic Range Computation Over Small Finite Rings"* IEEE Trans. Comput., vol. 43, no. 1, January 1994, pp. 78-86.

219. [WiSh-88] K. Wilken, J. Shen: *"Continuous Signature Monitoring: Efficient Concurent-Detection of Processor Control Errors "* Prog.,18 th. ITC, IEEE pp.914-925,1988.

220. [WiWe-94] Ihyue-Win Wei: *"A Systolic Power-Sun Circuits for GF (2^m)"* IEEE Trans. Comput., vol. 43, no. 2, February 1994, pp. 226-229.

221. [WoGo-94] W.F. Wong, E. Goto: *"Fast Hardware-Based Algorithms for Elementary Function Computations Using Rectangular Multiplier"* IEEE Trans. Comput., vol. 48, no. 3, pp. 278-294, March 1994.

222. [Wojt-88] Hans Wojtkowiak: *"Test und Testbarkait digitaler Schaltungen"* B.G. Teubner Stuttgart, 1988.

223. [Wojt-88] Hans Wojtkowiak: *"Test und Testbarkait digitaler Schaltungen "* B.G.Teubner, Stuttgart, 1988.

224. [YaMa-86] Che-Liang Yang, Gerald M. Mason: *"A Fault Identification Algirithm for ti-Diagnosable Systems"* IEEE Trans. Comput., vol. C 35, no. 6, June 1986, pp. 503-510.

225. [Yarm-90] W.N.Yarmolik: *"Fault Diagnosis of Digital Circuits"* John Willey & Soons, 1990.

226. [ZaCa-90] Christopher Y. Zarowski, Howard C. Card: *"On Additions and Multiplication*

with Hensel Codes" IEEE Trans. Comput., vol. 39, no. 12, pp. 1417-1423, December 1990.

227. 42 Tree-Dynamic Random-Acces-Memory (RAM) Chips IEEE Trans. Reab., vol. 41, March 1992, pp. 139 - 148.

228. [*** TI - 92] *** Texas Instruments: *"Advanced Digital Logic Guide"* ,1992.

229. [*** TI - 93] *** Texas Instruments: *"Low Voltage Technology - Data Book"*, 1993.