ALGORITMI PENTRU ASIGURAREA FUNCȚIONĂRII CU DEGRADARE TREPTATĂ A CALCULATORULUI DE BORD A MICROSATELITULUI

Eugen Suman, Roman Nicu, Nicolae Secrieru
Universitatea Tehnică a Moldovei
esuman@gmail.com, roman@rgg.md, nsecrieru@gmail.com

Abstract. *This paper deal with the approaches of reliability problems of microsatellite-board computer. The operation algorithms are proposed for on-board computer based on the principle of gradual degradation.*

Cuvinte-cheie: *microsatelit, calculator de bord, reconfigurabil, toleranță la defecte, degradare treptată, sistem de operare.*

I. Introducere

Cerințele față de tehnologiile avansate, precum şi tendințele electronicii în continuă creştere a densității componentelor, sporeşte sensibilitatea sistemelor față de factorii externi, inclusiv radiația. Abilitatea aparatului spațial de a îndeplini o anumită misiune se datorează performanțelor sistemelor electronice de bord. Orice deteriorare în performanța sistemelor electronice cauzată de radiație va afecta nefavorabil operațiunea satelitului. Deteriorările severe a componentelor electronice ale satelitului pot, ulterior, conduce la cedarea sistemului. Deteriorarea poate provoca dereglări temporale sau permanente în funcționarea sistemului. În prezenta lucrare se va reflecta realizarea algoritmilor de control, testare şi reconfigurare ca părți componente a sistemului de operare, care vor asigura funcționarea cu degradare treptată a computerului de bord al microsatelitului. Obiectul cercetării il constituie tehnicile de tolerare şi algoritmii de tratare a defectelor, apărute în cadrul subsistemelor calculatorului de bord, aplicate în scopul menținerii stării funcționale fiabile a acestora pe tot parcursul misiunii satelitului.

II. Algoritmii generali de funcționare a calculatorului de bord

Mecanismul cel mai efectiv în detectarea şi tolerarea defectelor este redundanța. Nivelul de redundanță fie hard sau soft depinde doar de posibilitățile şi restricțiile de proiectare (hard) sau de componența (soft), care se caracterizează prin parametrii ca: masa şi spațiul fizic, spațiul de memorie în cadrul unei unități, bugetul energetic, timpul de viață minim. Cea mai efectivă structură de organizare a modulelor unui sistem pentru detectarea defectelor şi identificarea unității defectate este redundanta modulară triplă (TMR); unitatile redundante de surplus sunt considerate de rezerva plasate în regim standby pâna unul din modulele de bază nu cedează.

În cadrul proiectului la baza structurii sistemului a fost determinată folosirea redundanței duble, numită şi metoda duplicării cu comparare. Alegerea structurii se argumentează prin urmatoarele:

1) într-un sistem distribuit funcționalitatea întregului sistem depinde nu doar de vitalitatea unității centrale dar şi de restul modulelor; deaceea duplicarea poate fi folosită la tot nivelul calculatorului de bord ca o formă necesară de protecție, satisfăcând astfel criteriului de adaptabilitate a algoritmilor în cadrul proiectului "SATUM", redundanța triplă fiind o exigență.

2) Redundanța dublă este o masură suficientă pentru demonstrarea funcționării sigure a sistemului şi necesară pentru descrierea mecanismelor de tratare a degradării funcționale a acestuia.

Sistemul redundant este structurat din două unități PLC şi una PLD (figura 1). Controlerele sunt identice, ambele au aceeaşi funcționalitate şi execută acelaşi soft. Unitatea PLD este utilizată pentru compararea rezultatelor obținute din procesarea a datelor de intrare de către controlere şi detectarea erorilor; un alt rol al unității PLD este controlul stărilor de funcționare a controlerelor şi determinarea modurilor de lucru a sistemului dat.

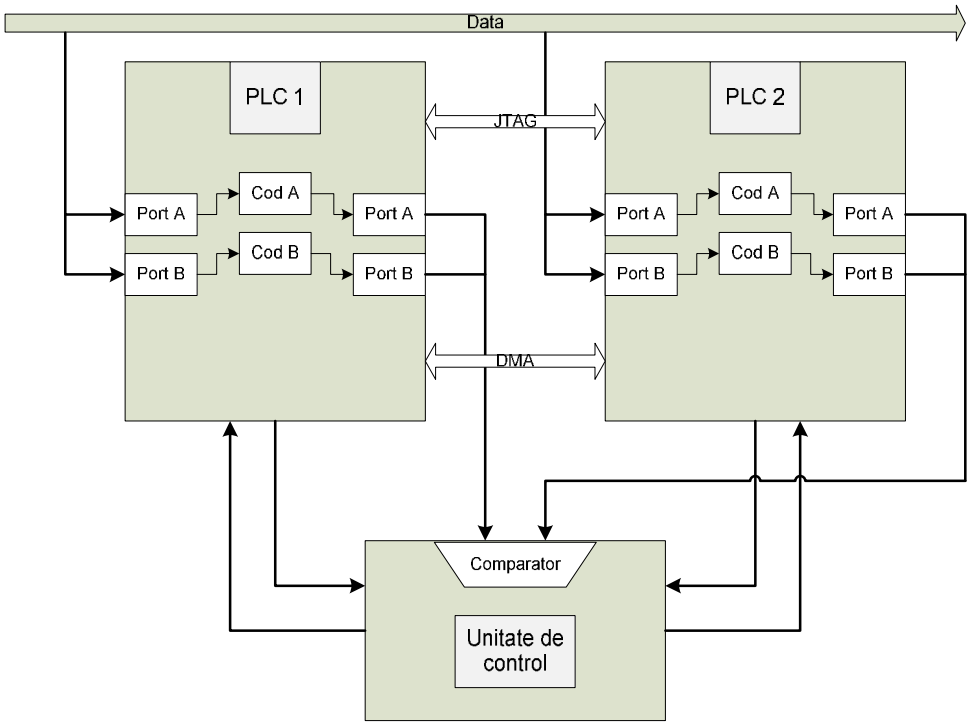

Figura 1. Structura sistemului cu redundanță modulară dubla.

Sistemul a fost demunit AFTD (Adaptive Fault Tolerance with Degradation) datorită mecanismelor adaptabile de tratare a erorilor, regimurilor de funcționare şi abilității de degradare a modulelor. În cadrul sistemului se folosesc diferite nivele de redundanță pentru detectarea erorilor şi protejarea datelor. La nivel de sistem se folosește redundanța dublă hardware, care şi caracterizează arhitectura sistemului. La nivel de modul este utilizată redundanța de timp, care se caracterizează prin duplicarea porturilor şi codurilor de prelucrare a datelor aplicațiilor, pentru procesare secvențial paralelă.

Algoritmul general de funcționare a modulelor PLC din cadrul sistemului este prezentat în figura 2. Procedura ce descrie aplicația modulului depinde doar de criteriile de proiectare şi aplicare a modulului în cadrul calculatorului de bord al microsatelitului. Celelalte proceduri indicate în grafic sunt necesarul pentru menținerea modulului în stare corect funcționabilă în condițiile mediului radiativ. În cadrul acestor proceduri se aplică

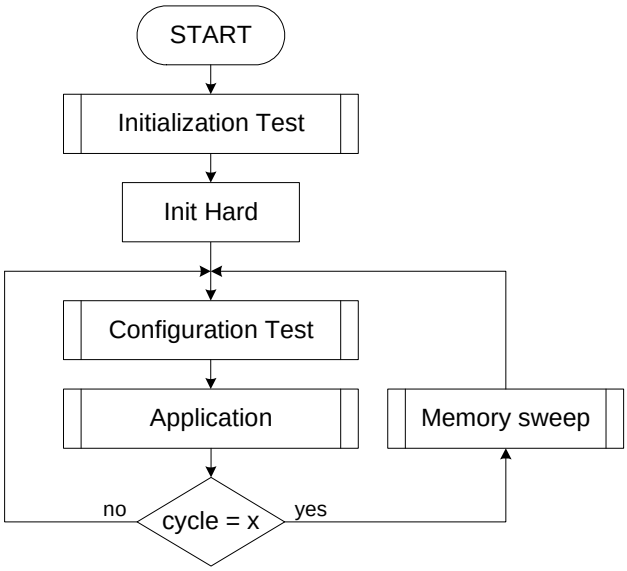

Figura 2. Algoritmul general de funcționare a modulelor din cadrul sistemului.

mecanismele de testare, detectare, corectare şi tolerare a defectelor şi mecanismul de configurare a modulului pentru menţinerea acestuia în stare funcţională pe parcursul misiunii. Modul dat de lucru se bazează pe prelucrarea aceloraşi date cu ajutorul ambelor controlere şi compararea rezultatelor prin intermediul PLD. În cazul rezultatelor identice se transmite la ieşirea PLD rezultatul controlerului, simbolic numit, primar. Atunci când rezultatele obținute de controlere diferă unitatea de control inițiaza rutina de autotestare a controlerelor (figura 3), care constă din testarea drumului parcurs de date de la portul de intrare spre cel de ieşire; codul de prelucrare a datelor este testat prin intermediul unor constante etalon, rezultatul obținut fiind comparat cu rezultatul etalon din memorie.

Ca rezultat al autotestării fiecare controler transmite un semnal de validare a propriului rezultat transmis. Dacă semnalele de validare diferă, ce semnalează faptul că un controler a realizat calculele corect şi celalalt a identificat o eroare, unitatea de control validează ieşirea ce corespunde rezultatului corect. În cazul în care semnalele sunt identice, care indică faptul că ambele iesiri sunt eronate şi controlerele au identificat erori; sau faptul că controlerele n-au identificat erori în rețeaua de prelucrare a datelor din cauza apariției unei erori transiente care disparuse la momentul testării, unitatea PLD resetează controlerele pentru a identifica starea lor de funcționare, obținută prin testare de initializare.

Pentru identificarea stărilor caracteristice controlerelor au fost stabilite codurile de stare comunicate unității de control. În dependență de codul comunicat unitatea de control ia decizii necesare funcționării corecte ulterioare a intregului sistem. Algoritmul de control a modulelor şi dirijare a regimurilor de funcționare a sistemului este prezentat în (figura 3).

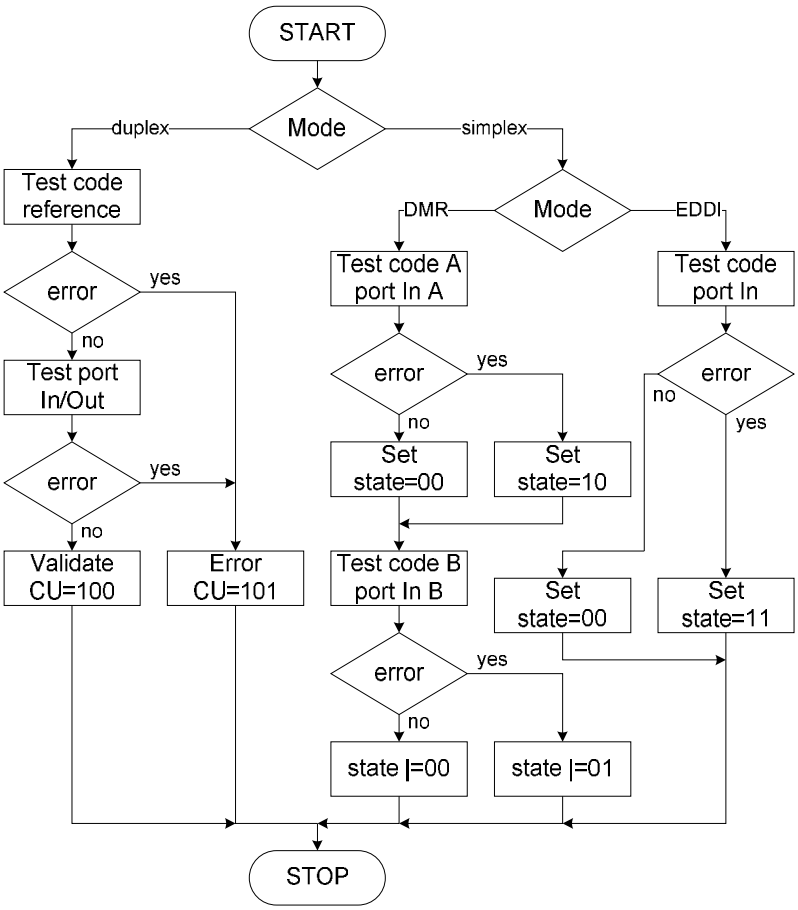

Figura 3. Algoritmul de testare a drumului de procesare a datelor.

Funcționarea în regim duplex a sistemului utilizat poate avea loc doar în cazul stării funcționale a ambelor module, codurile acestora corespunzând codului 1 din tabel, care sunt modificate de unitatea de control în codul 2, ce semnifica inițierea în lucru în corespundere cu codul 1. Codurile 4 si 5 se utilizează pentru determinarea rezultatului corect în cazul diferenței de rezultate obținute în regim de lucru duplex. Combinațiile posibile de stări a modulelor şi reacțiile corespunzătoare ale unității de comandă cu ulterioarele decizii sunt prezentate în tabelul 1.

Tabelul 1: Combinațiile de stări a modulelor şi deciziile unității de comandă

Codurile de stare a modulelor de calcul		Unitatea de comandă
MCU 1	MCU 2	
000	Orice cod	Aşteptare MCU 1
011	011	Mod duplex
	001	Mod simplex MCU 1
	010	Restabilire cod MCU 2 via JTAG
	110	Restabilire date MCU 2 via DMA
	111	Mod simplex MCU 1 / MCU 2 blocat
001	001	Aşteptare MCU 1 şi MCU 2
	010	Aşteptare MCU 1
	110	
	111	
010	010	Degradare de cod MCU 1 şi MCU 2
	110	
	111	Degradare de cod MCU 1 / MCU 2 blocat
110	110	Degradare de cod MCU 1 şi MCU 2
	111	Degradare de cod MCU 1 / MCU 2 blocat
111	111	Resetarea întregului sistem AFTD

În dependență de modul de funcționare stabilit de unitate de control, duplex sau simplex, modulul ales pentru a fi utilizat în aplicație se configurează conform performanțelor evaluate pentru îndeplinirea taskurilor. Regimul simplex de funcționare a sistemului, inițiat de unitatea de control în cazul în care unul din modulele aplicative nu este funcționabil, este caracterizat prin configurarea controlerului ales pentru prelucrarea datelor în mod duplex soft, folosind redundanța hardware implementată în soft (pentru un modul complet funcționabil) sau mecanismul de detectare a erorilor prin duplicarea instrucțiunilor (EDDI) (pentru un modul cu componente defectate). Algoritmul aplicației în dependență de regim de lucru este prezentat în figura 4. În regimul simplex unitatea de control PLD modifică mecanismul de validare a ieşirilor şi comutează direct rezultatul obținut de controler. Funcția de detectare a erorii de calcul este suportată

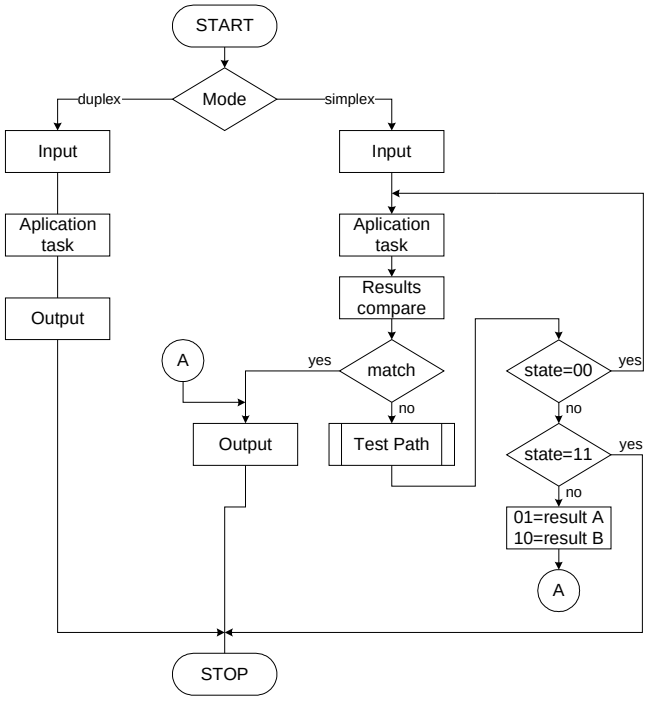

Figura 4. Algoritmul aplicației în dependență de regim de lucru.

complet de controlerul în funcție. În caz de necoincidere a rezultatelor obținute în calcul se inițiază automat mecanismul de testare a drumului de date; ca rezultat este fie ales rezultatul corect, în cazul unuia incorect, fie este efectuat calculul repetat, pentru ambele rezultate corecte, fie este inițiat mecanismul de testare profundă ulterior comunicând unității de control despre starea modulului. Deşi regimul de lucru a sistemului este stabilit de unitatea de control, decizia este luată în baza rezultatelor de testare a nivelului de funcționalitate a modulelor. Această procedură este realizată la inițializarea controlerelor, rezultatul descrie starea codului şi a datelor de control a modulului. După finisarea testării de inițializare se apelează procedura de testare şi configurare a drumului de aplicație, sunt alese porturile de intrare şi ieşire şi stabilite codurile ce pot fi aplicate în prelucrarea datelor. Ulterior este aleasă configurația cea mai reusită şi este comunicată starea funcțională a modulului către unitatea de control. Unitatea PLD comunică modulului funcțional regimul de lucru stabilit, conform căruia controlerul inițializeaza porturile şi intră în regim de lucru.

III. Restabilirea funcționalității modulelor calculatorului de bord

Procedura de restabilire a codului implică mecanismul de detectare şi corectare a erorilor EDAC şi poate fi apelată periodic pentru reânnoirea codului, măsură de prevenire a erorilor şi defectelor, sau aperiodic pentru corectarea codului, la detectarea erorilor în acesta. Deşi mecanismele de reânnoire şi corectare a codului implică aceeaşi rutină de diagnosticare a celulelor de memorie de cod, procedura de restabilire a codului diferă substanțial. Reânnoire codului, numită şi curățirea memoriei, constă în divizarea codului în blocuri, aplicarea mecanismului EDAC de detectare şi corectare a erorilor în cardul fiecarui bloc şi reânscrierea blocurilor în memorie pe pozițiile inițiale. Corectarea codului, spre diferență de mecanismul precedent, are funcție strict definită care constă în identificarea celulei de memorie eronată, corectarea erorii şi reânscrierea celulei în locul ei inițial în cadrul codului.

Procedura de restabilire a codului începe cu setarea modulului în stare de testare, diagnosticare şi corectare, semnalând unitatea de control. Pasul următor este testarea rutinei de calcul CRC. Dacă rutina a fost determinată eronată, stabilind eroare de cod, se apelează procedura de corectare a codului cu condiția că testările în cadrul acestei proceduri sunt omise şi are loc diagnosticarea directă a codului. Rutina de calcul CRC determinată corectă continuă cu verificarea valorilor etalon ale cuvintelor de CRC pentru codul Hamming, biții de control şi biții de control al biților de control.

S-a propus duplicarea mecanismului de decodare al codului Hamming. Aceasta se explică prin faptul că apelarea procedurii software de detectare şi corectare a erorilor implică citirea codului procedurii care nu poate fi modificat; într-atât condiția de reânscriere a codului în memorie este că procesorul nu accesează partea de memorie prelucrată pe tot parcursul procedurii. Deaceea citirea codului de corectare a memoriei nu permite corectarea porțiunii de cod unde este scrisă rutina dată. Astfel, pentru ca codul cu rutina de corecție să poată fi corectat în caz de eroare, este necesară o rutină duplicat pentru a acoperi codul rutinei primare. Pe lânga acest fapt mai stă şi simpla redundanță care permite alegerea codului EDAC neeronat pentru a fi aplicat, de altfel rutina eronată nu poate fi aplicată pentru corectarea codului. În afară de codul duplicat a fost examinată şi problema păstrării biților de control pentru decodorul Hamming în stare neeronata, pentru a majora probabilitatea corectării codului. Pentru aceasta se execută codarea biților de control prin intermediul codului Hamming. Biții de control al biților de control se păstrează în flash cu biții de control de cod. Acest fapt permite detectarea şi corectarea erorilor în cadrul biților de control al codului. Din rutinele codului Hamming primară şi secundară din cadrul codului modulului se determină rutina fără erori care va fi ulterior aplicată pentru restabilirea codului, în cazul ambelor neeronate este aleasă rutina primară. Dacă ambele rutine Hamming sunt eronate, atunci se setează codul de eroare de cod nereparabilă, comunicată unității de control care, în cazul modulului de

rezervă cu cod neeronat, va iniția procedura de reânscriere a codului modulului defectat cu cel întreg. După alegerea rutinei Hamming sunt verificați biții de control al codului şi biții de control al biților de control. În cazul detectării erorilor în cadrul biților dați se aplică codul Hamming pentru restabilirea acestora. Dacă biții de control al codului nu sunt eronați, şi biții de control ai biților de control sunt, şi erorile sunt multiple, atunci încercarea de restabilire a acestor biți prin intermediul decoderului va eşua; deaceea poate fi aplicatş codarea biților de control de cod cu reânscrierea ulterioară a biților secundari. Se aplică rutina Hamming pentru corectarea codului pentru adresele stabilite, după care modulul se restartează. Dacă au fost detectate erori duble se setează codul de eroare de cod necorectabil. În cazul în care biții de control al codului n-au putut fi restabiliți nici printr-o metodă, fapt care poate fi detectat după restartarea modulului la testarea de inițializare şi care apelează procedura EDAC, la începutul rutinei se verifică fanionul de stare al biților eronati care, fiind setat, duce la semnalizarea unității de control despre starea eronată a biților de control Hamming din memoria flash. Cu modulul secund întreg unitatea de control inițiază procedura de reânscriere a biților eronați cu cei neeronați din modulul de rezervă. Procedura de restabilire a codului este prevăzută a fi apelată de trei ori consecutiv după ce se setează fanionul de eroare de cod. Contorul apelărilor consecutive a procedurii EDAC este resetat atunci când modulul finisează cu succes procedura testării de inițializare.

IV. Concluzii

În această lucrare este prezentată modalitatea de combinare a tehnicilor şi algoritmilor sistemului de operare într-o structură unică compactă cu fiabilitate sporită şi degradare treptată multinivel. Rezultatele au fost implementate în proiectul „SATUM" şi constă în abilitatea softului, bazat pe algoritmii realizați, de a se adapta nivelului de degradare prin reconfigurare cu menținerea nivelului relativ sporit de productivitate şi păstrarea autenticității datelor procesate.

V. Referințe

1. A. Stoica, T. Arslan, S. Baloch, *Probability Based Partial Triple Modular Redundancy Technique for Reconfigurable Architectures*, IEEEAC paper #1085, Version 4, 2005.

2. Chung-Yu Liu. *A study of flight-critical computer system recovery from space radiation-induced error*. In Digital Avionics Systems, 2001. DASC. The 20th Conference, Vol, 1, pp. 1–10, 2001.

3. F. Di Giandomenico, S. Chiaradonna, A. Bondavalli, and F. Grandoni, *Evaluation of Integrated Error Processing and Fault Diagnosis in Multiprocessor Systems* - In: PDPTA '2000, Las Vegas, Nevada, USA, pp. 1145-1151, 2000.

4. I. Troxel, E. Grobelny, G. Cieslewski, J. Curreri, M. Fischer and A. George, "*Reliable management services for COTS based space systems and applications* - In: Proc. Of International Conference on Embedded Systems & Applications (ESA), Las Vegas, NV, June 26-29, 2006.

5. J. Srour, C. Marshall, and P. Marshall, Review of displacement damage effects in silicon devices," IEEE Transactions on Nuclear Science, vol. 50, no. 3, 2003, pp. 653-670.

6. Mitra, S., N. R. Saxena and E. J. McCluskey, *A Design Diversity Metric and Reliability Analysis for Redundant Systems*, - In: Proc. Intl. Test Conf., pp. 662-671, 1999.

7. Oh, N., P.P. Shirvani and E.J. McCluskey, "Control Flow Checking by Software Signatures," to appear in IEEE Transactions on Reliability Sep. 2001.

8. Secrieru N., Cîrțîca A., Suman E., *Protocolul si algoritmii de comunicare ale unitatilor de comanda ale microsatelitului* -In:The 3rdInternational Conference on Telecommunications, Electronics and Informatics May 20 – 23, 2010. Chişinău Volume I p. 411.