

INSTITUTUL POLITEHNIC TIMISOARA  
FACULTATEA DE ELECTROTEHNICA

ROSU MIHAIL-EUGEN-GRIGORE

T E Z A   D E   D O C T O R A T

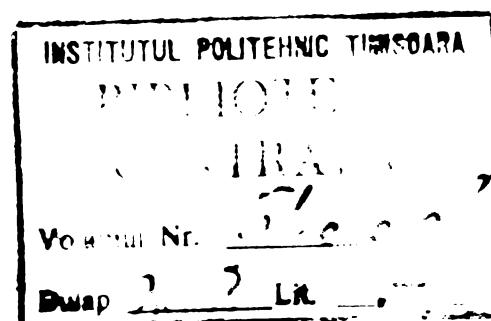
CUPLAREA SIMULTANA A UNITATILOR DE  
DISCURI MAGNETICE LA DOUA SAU MAI MULTE  
SISTEME DE CALCUL

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

CONDUCTATOR STIINTIFIC

| Prof. Dr. Ing. ALEXANDRU ROGOJAN |

T I M I S O A R A  
1984





ROSU MIHAIL-EUGEN-GRIGORE

T E Z A   D E   D O C T O R A T

CUPLAREA SIMULTANA A UNITATILOR DE  
DISCURI MAGNETICE LA DOUA SAU MAI MULTE  
SISTEME DE CALCUL

CONDUCATOR STIINTIFIC

Prof.Dr.Ing. MIRCEA PETRESCU

• :            1985

## C U P R I N S

	PAG.
<b>CAP. 1. INTRODUCERE</b>	
1.1. Obiectul lucrării	1
1.2. Prezentarea generală a lucrării	5
<b>CAP. 2. SUBSISTEMUL DE DISCURI MAGNETICE</b>	
2.1. Definirea problemei	15
2.2. Definirea locului SDM în diversele sisteme de calcul	18
2.3. Considerații tehnologice și economice	22
2.4. Definirea interfetelor SDM	23
<b>CAP. 3. INTERFATA INTERNA A SDM</b>	
3.1. Grupe funcționale	23
3.2. Organizarea interfetei interne a SDM	24
3.3. Dublul acces	28
3.4. Cuplarea simultană a UDM cu diverse capacități de înregistrare	29
3.5. Concluzii	32
<b>CAP. 4. INTERFATA EXTERNA A SDM</b>	
4.1. Clasificarea cuploanelor componente ale SDM	34
4.2. Operațiile de bază executate de cuplo- rul din SDM	35
4.2.1. Operația de poziționare a capetelor de scriere/citire (SEEK)	36
4.2.2. Operația de scriere	36
4.2.3. Operația de citire	38
4.3. Formatul înregistrării	42
4.4. Ordinele utilizate pentru comanda SDM	44
4.5. Concluzii	45
<b>CAP. 5. INTERFATA SISTEMULUI "INDEPENDENT"</b>	
5.1. Organizarea sistemului	46
5.2. Liniile de semnal ale Busului I	48
5.3. Transferul de date	50
5.3.1. Generalități	50
5.3.2. Transferul de tip INTRARE/ INTRARE PAUZA	54
5.3.3. Transferul de tip IESIRE/ IESIRE OCTET	58
5.4. Secvența de arbitraj a priorității și secvența de întrerupere	60
5.4.1. Structura prioritara	60

5.4.2. Secventa de arbitrare a prioritatii	62
5.4.3. Secventa de intrerupere	66
5.5. Analiza structurii cu mai multe UC	70
<b>CAP - 6 - ALOCAREA TAMPULUI DE LUCRU AL SDM</b>	
6.1. Strategii de alocare	71
6.1.1. Strategii fara reciclaaj	71
6.1.2. Strategii de reciclaaj	72
6.2. Model de alocare	72
6.2.1. Calculul timpului $A_1$	74
6.2.2. Calculul timpului $A_2$	75
6.3. Calculul timpului de acces la informa- tie pentru UDM	79
6.3.1. Cazul "cozii de asteptare simple" (FIFO)	80
6.3.2. Cazul "timpului de acces MINIM" (SATF)	80
6.4. Concluzii	81
<b>CAP - 7 - TRATAREA FENOMENULUI DE INTERBLOCAJ</b>	
7.1. Generalitati	83
7.2. Starile unui sistem	84
7.3. Interblocajul	86
7.4. Remediile interblocajului	90
7.4.1. Prevenirea statica a interblocajului	92
7.4.2. Prevenirea dinamica a interblocajului	94
7.5. Concluzii	96
<b>CAP - 8 - MECANISME DE SINCRONIZARE</b>	
8.1. Principiul de realizare a operatiilor de sincronizare. Semafoare	97
8.1.1. Proprietatile semafoarelor	98
8.1.2. Semafoare de excludere mutuala	101
8.2. Tranzitiile posibile de stare ale procesoarelor	101
8.3. Mecanisme de sincronizare	104
8.4. Concluzii	106
<b>CAP - 9 - LOCUL SDM IN CONFIGURATIILE MULTIPROCESOR</b>	
9.1. Multiprocesorul banalizat	107
9.1.1. Accesul tuturor procesoarelor	

la toate resursele	108
9.1.2. Autonomia proceselor fata de procesoare	109
9.2. Configuratiile cu minicalculatoare "INDEPENDENT"	110
9.3. Influenta introducerii circuitelor VLSI asupra arhitecturii configura- tiilor multiprocesor	112
<b>CAP. 10. PROBLEMA CONFLICTELOR DE ACCES</b>	
10.1. Analiza teoretica a problemei	116
10.2. Masuratori	119
10.2.1. Etapa I-a	119
10.2.2. Etapa II-a	127
10.3. Concluzii	128
<b>CAP. 11. CUPLAREA SIMULTANA A UDM LA     MAI MULTE SISTEME DE CALCUL     IN CONTEXTUL MODERNIZARII ARHI-     TECTURII CONFIGURATIILOR DE CALCUL</b>	129
<b>CAP. 12. CONCLUZII</b>	134
PRESCURTARI	137
BIBLIOGRAFIE	138

## CAP.1. INTRODUCERE

Directivele Congresului al XI-lea al P.C.R. prevede in "Obiectivele fundamentale ale etapei urmatoare a istoriei Romaniei socialiste" ca in anii urmatoari electrotehnica si indeosebi electronica vor cunoaste o puternica dezvoltare. Va trebui asigurata producerea in tara a mijloacelor de calcul, de automatizare si conducere necesare activitatii economice si de productie. La Conferinta Nationala a P.C.R. din 7 - 9 decembrie 1977 se prevede ca "in industria electronica se va urmari asimilarea si dezvoltarea productiei de noi tipuri de calculatoare electronice, inclusiv echipamentele periferice aferente".

In acest context a fost creat si s-a dezvoltat Institutul de Cercetari pentru Tehnica de Calcul, in cadrul caruia s-au desfasurat lucrarile legate de prezenta teza de doctorat.

### 1.1. *Obiectul lucrarii*

Scopul lucrarii este de a studia cuplarea simultana la doua sau mai multe sisteme de calcul a unitatilor de discuri magnetice.

Problema cuplarii echipamentelor periferice la sistemele de calcul este sumar tratata in literatura de specialitate. Ponderea principala este ocupata de filozofia si structura sistemelor si a unitatilor centrale. Intrarea-iesirea este tratata global, facindu-se anumite analize privind influenta rezolvarii problemei intrarii-iesirii in vederea obtinerii unor performante ridicate ale sistemului propriu-zis. Firmele specializate in fabricarea echipamentelor periferice livreaza o documentatie foarte buna asupra echipamentului elaborat, dar nu publica nimic asupra lucrarilor de cercetare-dezvoltare efectuate in scopul elaborarii echipamentului. Acelasi procedeu este folosit la marile firme producatoare de sisteme de calcul. Departamentele de cercetare ale acestor firme desfasoara lucrari de inalt nivel stiintific, finalizate prin rezultate superioare fata de nivelul tehnic al produselor din productia curenta. Interesele comerciale, dicteaza firmelor sa nu introduca in productie un produs nou, atita timp cit vechiul produs se vinde bine si pina cind alte firme nu comercializeaza un produs superior. Aceasta politica aplicata in tarile capitaliste, face ca in literatura de specialitate sa apara ori

numai studii pur teoretice a caror aplicare practica presupune un mare volum de munca suplimentar de cercetare si dezvoltare, ori o prezentare de prospect si scolarizare care reprezinta descrierea unor produse cercetate cu mai multi ani in urma. Aceasta este in momentul de fata stadiul posibilitatilor de documentare pentru rezolvarea problemei propuse.

Problema cuplarii simultane a unitatilor de discuri magnetice la doua sisteme de calcul a fost actuala in perioada 1970 - 1972 cind se cerea o rezolvare optima a variantei "dublului acces", apoi a pierdut din interes in intervalul 1975 - 1978, pentru ca sa devina actuala in prezent datorita dezvoltarilor tehnologice in domeniul circuitelor integrate si a relativei dezvoltari lente a tehnologiei echipamentelor periferice. Decalajul tehnologic aparut intre dezvoltarea rapida a circuitelor integrate si dezvoltarea mai lenta a echipamentelor periferice, a dus si la un decalaj valoric intre acestea, astfel ca la anumite sisteme de calcul este avantajoasa utilizarea in comun de catre mai multe procesoare, a unui grup de echipamente periferice. Unitatile de discuri magnetice au evoluat destul de mult in ultimii 10 ani pretul unei unitati a ramas aproape constant, insa pretul de stocare al informatiei a scazut cu mai mult de un ordin de marime prin cresterea corespunzatoare a capacitatii de stocare. In acest context se pot face modificari in arhitectura sistemelor, studiul aspectelor teoretice, practice si economice ale unor configuratii biprocesor sau multiprocesor care utilizeaza in comun acelasi Sub-sistem de discuri magnetice fiind de actualitate. De asemenea prezinta interes pina la ce nivel se poate da acestui subsistem o independenta relativa fata de procesoarele din sistem, care este puterea de calcul si care functii software din Sistemul de Operare pot fi preluate de hardware-ul din subsistem. Aceste modificari in arhitectura sistemului ridica probleme noi fata de cazul unor structuri clasice, lucrarea ca-utind sa puna in evidenta aceste probleme, sa stabileasca implicatiile si dimensiunea acestora si sa dea caile de rezolvare.

Unitatile de cuplare pentru discurile magnetice, utilizate pina in prezent in sistemele cunoscute elaborate de marile firme constructoare de calculatoare (IBM, DEC, CDC, SIEMENS, ICL, etc.) sint simple cuploare care realizea-



za functiile de transfer intre unitatea centrala sau un canal de intrare-iesire si echipamentele periferice propriuzise. Cele mai moderne modele realizeaza pe langa detectia si uneori corectia erorilor si o testare ("in-line") a unui echipament periferic, simultan cu exploatarea normala a celorlalte echipamente. Modelele mai moderne fiind microprogramate permit o eventuala dezvoltare ulterioara a microprogramelor in vederea realizarii unor functiuni suplimentare.

Acesta fiind stadiul actual al rezolvarii teoretice si practice a problemelor din domeniul abordat, se impune analiza completa a tuturor aspectelor hardware si software legate de acest domeniu de interes si sistematizarea succinta a elementelor necesare pentru o dezvoltare completa a problemei. Lucrarea urmareste realizarea unui cadru teoretic care sa permita dezvoltarea unor dispozitive hardware de cuplare optima a unitatilor de discuri magnetice. Daca pina acum problema era tratata separat hardware si software, adica dispozitivul de cuplare propriu-zis si respectiv driverul aferent din Sistemul de Operare, in lucrare se analizeaza totul ca o problema unitara. In acest mod se studiaza posibilitatea transferarii unor functii software in hardware-ul cuplorului. De asemenea se studiaza posibilitatea cuplarii la mai multe sisteme de calcul, aparind notiunea de Subsistem de discuri magnetice. Se studiaza implicatiile acestor schimbari si modul adecvat de solutionare:

- alocarea timpului de lucru al Subsistemului de discuri magnetice necesita analizarea strategiilor de alocare si a timpilor de acces la informatie si a stabilirii unui model de alocare adecvat.

- fenomenul de interblocaj specific Sistemului de Operare, poate apare acum in interiorul Subsistemului de discuri magnetice, fiind un fenomen interzis, deci se impune analiza conditiilor in care apare si a gasirii metodelor de prevenire.

- sincronizarea unor lucrari paralele care se pot desfasura in cadrul Subsistemului de discuri magnetice necesita analiza mecanismelor de sincronizare si a modalitatilor de sincronizare cu semafoare.

- problema conflictelor de acces la informatie, unde dupa analiza teoretica a problemei, s-a stabilit un algoritm si o metoda de masuratoare si de verificare a jus-

tetii algoritmului.

In perioada 1972 - 1982 autorul a lucrat la tema de cercetare si elaborare a unor subansamble ale Sistemului de discuri magnetice cum ar fi: Unitatea de discuri magnetice DM-29 si minicalculatorul INDEPENDENT-100, precum si la lucrari complexe cu minicalculatoare unde Subsistemul de discuri magnetice a fost utilizat in comun de doua sisteme de calcul cum au fost : "Sistemul de supraveghere a instalatiilor de la OLTCIT - Craiova" si "Sistemul de avertizare hidrometeorologica la inundatii I.M.H.". Aceste elemente au constituit o platforma de observatii si experimentari care au dus la elucidarea teoretica si practica a unor aspecte legate de cuplarea simultana a unitatilor de discuri magnetice la mai multe sisteme de calcul. Pentru aprofundarea problemei a fost elaborata o metoda de simulare, prin care se probeze similitudinea intre calculele teoretice si rezultatele masuratorilor facute pe sisteme INDEPENDENT-100.

Principalele contributii ale autorului la dezvoltarea acestei lucrari privesc :

- realizarea "grupului unic" si a "modulului virtual" pentru utilizarea simultana a unitatilor de discuri magnetice de diverse capacitati;

- cercetarea unor metodologii de calcul a timpului de acces la informatie pentru stabilirea unor algoritmi de acces imbunatatit;

- analiza unor strategii de alocare, a fenomenului de interblocaj si a sincronizarii cu semafoare la nivelul Subsistemului de discuri magnetice;

- elaborarea unei proceduri de evaluare a coeficientului real de utilizare a Subsistemului de discuri magnetice si realizarea de masuratori pe un montaj de simulare si pe o configuratie reala biprocesor;

- analiza transferului unor functiuni software din Sistemul de Operare la hardware-ul Subsistemului de discuri magnetice.

S-a cautat ca pe langa metodele si rezultatele originale, sa se realizeze si o sistematizare a problemelor legate de acest domeniu de interes. Lucrarea isi propune sa elucideze toate aspectele legate de cuplarea simultana a unitatilor de discuri magnetice la mai multe sisteme de calcul.

Rezultatele obtinute pot fi utilizate la realizarea

unui asemenea Subsistem, proiectantului fiindu-i puse la dispozitie elementele necesare dezvoltarii practice a lucrării.

Din punctul de vedere al organizării, lucrarea este impartita in 12 capitole, mentionind faptul ca incepind cu capitolul 2 notiunile intilnite mai frecvent in cursul lucrării, au fost prescurtate. Structura lucrării poate fi expusa prezentind pe scurt, in cele ce urmeaza, continutul acestor capitole.

## 1.2. *Prezentarea generala a lucrării*

1.2.1. Capitolul 1, de introducere, precizeaza obiectul lucrării. In continuare se face o prezentare generala a continutului lucrării si a contributiilor originale aduse de autor.

1.2.2. In capitolul 2 este prezentat Subsistemul de discuri magnetice. Se defineste notiunea de Subsistem de discuri magnetice ca un ansamblu fizic format din echipamentul periferic propriu-zis adica unitatile de discuri magnetice, cuplorul necesar legării acestora la sistemul de calcul, cablurile de legatura intre echipamentele periferice si cuplor, precum si in anumite cazuri asigurarea software specifica anumitor functii. Aceasta definire este necesara in vederea delimitării subsistemului, a functiilor sale, pentru realizarea unui tot unitar cu o independenta functionala maxima fata de restul sistemului. In acest fel se pot face studii care sa duca la alegerea unor solutii optime de interconectare si implicit de minimizare a suportului software-ului aferent din Sistemul de Operare. Un alt avantaj care rezulta prin utilizarea notiunii de subsistem se refera la posibilitatea definiirii unei magistrale unice de conectare a subsistemului cu restul configuratiei de calcul, creindu-se posibilitatea analizei aprofundate a dialogului de informatii pe aceasta magistrala si astfel alegerea unor solutii care sa duca la obtinerea unui randament maxim al comunicatiilor pe magistrala respectiva. Se urmareste obtinerea unui principiu unic de interconectare.

Se analizeaza problema cuplării simultane a unitatilor de discuri magnetice la doua sau mai multe sisteme de calcul, in contextul dezvoltării tehnologice actuale. Daca in perioada 1975 - 1982 unitatile de discuri magnetice disponibile la noi au ramas aceleasi avind capacitatea maxima de 58 Moct., sistemele de calcul au evoluat mult in concordanța cu dezvoltările tehnologice din domeniul circuitelor

integrate precum si cu noile concepte arhitecturale. Aparitia sistemelor minicalculator a reactualizat problema cuplarii simultane. La vechile sisteme de tip FELIX C-256 rezolvarea problemei a constat in realizarea "dublului acces" prin care doua sisteme de calcul puteau lucra cu aceleasi unitati de discuri magnetice echipate cu o interfata dubla. La noile sisteme cuplarea simultana poate lua insa mai multe aspecte. Sistemele minicalculator ieftine dar cu mare putere de calcul, vor putea utiliza in comun un grup de echipamente periferice cu preturi unitare relativ ridicate.

Se defineste locul Subsistemului de discuri magnetice in sistemul de calcul. La sistemele din generatia a 3-a reprezentate de familia FELIX C-256 (sisteme cu logica cablata), Subsistemul de discuri magnetice este cuplat la un Canal Multiplexor functiile sale fiind limitate. Modificarea structurii cablate este foarte greoaie, de multe ori imposibila, astfel ca la aceste subsisteme nu s-au putut aduce imbunatatiri. Subsistemul a devenit mai flexibil prin realizarea cuploarelor cu structura microprogramata, unde se implementeaza o serie de functii noi. De asemenea noile structuri minicalculator de tip INDEPENDENT-100 au permis operarea de modificari pozitive in ansamblul subsistemului.

In continuare se pun in evidenta strategiile de proiectare in corelatie cu parametrii: pret si performante. Se evidentiaza pastrarea unor preturi constante cu realizarea unor performante sporite, politica urmarita si la elaborarea familiei INDEPENDENT, sau pastrarea performantelor cu preturi de realizare mai mici. Pentru obtinerea performantelor de virf dinamica preturilor va fi pozitiva. Realizarile tehnologice din domeniul circuitelor integrate duc indirect la minimizari ale surselor de alimentare, a sistemului de ventilatie, a suprafetei de circuit imprimat deci la micșorarea pretului global.

In incheiere se nominalizeaza cele doua interfete ale subsistemului de discuri magnetice si anume interfata interna intre cuplor si unitatile de discuri magnetice si interfata externa intre cuplor si sistemul de calcul propriu-zis.

1.2.3. Capitolul 3 este dedicat interfetei interne a Subsistemului de discuri magnetice. Cu toata diversificarea tipurilor de unitati de discuri magnetice si de multitudinea firmelor producatoare, datorita functiilor pe

care trebuie sa le indeplineasca echipamentul, semnalele de interfata interna sint aproximativ aceleasi factorii de diferenta fiind anumite secvente de dialog, nivelul semnalelor si conectorii de legatura. Se face o clasificare a unitatilor pe grupe functionale.

Semnalele interfetei interne sint analizate luindu-se ca model unitatea DM-29 cu capacitatea de inregistrare de 29 Moct., realizata de un colectiv condus de autor. Se evidentiaza semnalele comune tuturor unitatilor si semnalele specifice unei unitati separate, precum si modul de organizare a cuplarii. Se nominalizeaza cele opt semnale de esantionare si se arata semnificatia semnalelor pe liniile bidirectionale care asigura legatura de date a unitatilor cu cuplorul. Dialogul intre cuplor si unitati este pus in evidenta prin exemplificari.

Lucrul in regim de dublu acces prezinta doua posibilitati si anume dublul acces monocalculator unde unitatile de discuri magnetice cu dublu acces sint cuplate la acelasi calculator prin doua cuploare si dublu acces biccalculator unde fiecare din cele doua cuploare este legat la cite un calculator separat. Sint nominalizate semnalele specifice dublului acces.

Pentru realizarea cuplarii simultane la acelasi cuplor a unitatilor de discuri magnetice de capacitati diferite, se defineste notiunea de unitate cu capacitate "majora" si de "module virtuale" formate din unitati cu capacitatea submultiplu al capacitatii "majore". Solutia prezentata este originala, permitind utilizarea alaturi de unitatile moderne a unitatilor mai vechi cu capacitate de inregistrare mai mica. In configuratia clasica apare un element nou "adaptorul", pentru care se pun in evidenta functiile pe care trebuie sa le indeplineasca. Utilizarea modulelelor virtuale are efecte pozitive asupra modului de acces la informatia stocata pe unitatile de discuri magnetice care compun modulul, obtinindu-se o sporire a vitezei de transfer a informatiei.

1.2.4. Capitolul 4 referitor la interfata externa a Subsistemului de discuri magnetice incepe cu clasificarea cuploarelor care intra in componenta subsistemului, trebuind subliniat faptul ca modul de lucru asincron este cel mai favorabil pentru realizarea dialogului in cadrul sistemului de calcul. De mentionat ca, la cuploarele reali-

zate pînă în prezent nu s-a făcut o trecere a unor funcții software existente în Sistemul de Operare în hardware-ul specific cuplorului. De asemenea interfața externă a subsistemului nu a fost studiată și optimizată ținîndu-se seama de toate particularitățile cuplorului pentru unitățile de discuri magnetice.

Se enumera operațiile de bază executate de subsistem făcîndu-se o sistematizare a acestora în legătura cu organizarea subsistemului în subansamble funcționale. Aceasta defalcare permite o mai bună punere în evidență a minioperațiilor ce trebuie executate, a subrutinelor de comandă ce vor fi luate în considerare și în final, obținerea de informații ce pot fi utilizate la optimizarea subsistemului în ansamblu. Se evidențiază operația de premarcare a pachetului de discuri, detaliîndu-se operațiile de poziționare, scriere și citire. Un element care trebuie avut în vedere în scopul realizării unor compatibilități între diversele unități de discuri magnetice, este formatul înregistrării. În acest scop se face o analiză a elementelor constitutive ale unui format de înregistrare.

În continuare sînt tratate ordinele utilizate pentru comandă subsistemului, urmărindu-se principiul de realizare a transferului de informație. Alegerea unor ordine puternice dar simple ca structură permite transformarea subsistemului într-un Subsistem de Intrare-Iesire cu autonomie de calcul, care să poată prelua o parte din sarcinile îndeplinite în prezent de Sistemul de Operare și implicit să descongese procesorul central parțial sau în anumite cazuri total de sarcinile transferului, prelucrării și optimizării informațiilor.

1.2.5. În capitolul 5, cu scopul de a analiza rolul interfeței în rezolvarea problemei cuplării simultane a unor unități de discuri magnetice la două sau mai multe sisteme de calcul, se studiază interfața sistemului minicalculator INDEPENDENT-100 și măsurile luate pentru ca această interfață să concure la rezolvarea acestui deziderat. La minicalculatorul I-100 urmărindu-se realizarea unui sistem de calcul eficient și compatibil cu sistemele firmei DEC din S.U.A. și cele realizate în țările socialiste, s-a ales ca element de compatibilitate interfața sistemului. Concepția interfeței a permis realizarea unei compatibilități totale software cu aceste sisteme. Interfața este

constituita din doua magistrale, dintre care una dedicata memoriei operative a sistemului. Se studiaza posibilitatea crearii de sisteme multiprocesor in jurul celor doua magistrale. Problema dificila este aceea a realizarii unui arbitru de prioritati eficient care sa gestioneze folosirea magistralelor. Apare problema conflictelor de acces care va fi studiata in amanuntime in capitolul 10. In continuare se face o analiza detaliata a principiului de functionare a magistralei alese, pentru a se putea trece la magistralele multiprocesor. Rezolvarea acestei probleme este originala si implicit reprezinta o solutie pentru cuplarea simultana la mai multe sisteme de calcul a unor subsisteme de intrare-iesire.

1.2.6. La Subsistemul de discuri magnetice care lucreaza ca un subsistem independent apar probleme legate de alocare timpului de lucru. Exista doua domenii de studiu si anume: cum sa se aloce timpul efectiv de lucru al subsistemului pentru rezolvarea anumitor lucrari si care este timpul de acces la informatia stocata pe echipamentul periferic respectiv. Aceste aspecte sint tratate in capitolul 6. Dupa ce se face o analiza a unor strategii de alocare, se trece la calculul timpului de alocare. Se constituie un model de alocare care functioneaza dupa urmatorul algoritm: orice lucrare primeste odata o singura cuanta de timp a carui durata este predeterminata, o lucrare libereaza subsistemul atunci cind se termina lucrarea sau cind ajunge la sfirsitul cuantei repartizate, la eliberare subsistemul este alocat lucrarii care a primit cele mai putine cuante. S-a folosit o strategie de ordonare cu mai multe nivele in care lucrarile sint aranjate in siruri de prioritati diferite considerindu-se ca cererile sosesc dupa o distributie Poisson cu un debit mediu  $\lambda$ . Modelul de alocare permite prin varierea parametrilor sa se utilizeze calculatorul pentru gasirea unor variante convenabile de alocare. Metoda s-a substituit utilizarii unui simulator hardware, a carui elaborare ar fi prezentat dificultati constructive. In cadrul Subsistemului de discuri magnetice durata unei cuante poate fi aleasa sa coincida cu lungimea unui sector, a unei piste sau a unui cilindru, sau la o metoda mai complicata dar mai eficienta se poate lucra cu cuante de durata variabile in concordanta cu lungimea inregistrarii daca pe durata acestei inregistrari nu apar miscari ale an-

samblului de pozitionare a capetelor de scriere-citire. La Subsistemul de discuri magnetice metodele de alocare se pot aplica eficient, deoarece inregistrarea propriu-zisa are in componenta sa elemente informationale suficiente, care indica lungimea inregistrarii, locurile libere unde se pot face inregistrările (in cazul operatiei de scriere) si locul unde este plasata inregistrarea (in cazul operatiei de citire).

Timpul de acces la informatie a fost calculat in cazul "cozii de asteptare simple" (FIFO) si in cazul "timpului de acces minim", observindu-se o crestere a eficientei lucrului in cazul al doilea, odata cu cresterea numarului de cereri simultane.

1.2.7. Transferarea unor sarcini din Sistemul de Operare la nivelul Subsistemului de discuri magnetice implica si transferul rezolvarii unor probleme care apareau in software. Un fenomen suparator este "interblocajul", fenomen prin care in operatia de cerere si repartizare a resurselor catre diverse procesoare poate sa apara situatia ca acestea sa se blocheze reciproc. Fenomenul de interblocaj este tratat in capitolul 7, urmărindu-se gasirea unor solutii de detectare si prevenire a fenomenului. Se incepe prin descrierea starilor unui sistem si a modului in care se poate trece dintr-o stare in alta. Se definesc notiunile de: resursa ceruta si resursa alocata precum si operatiile de: cerere, achizitie si liberare de resurse.

Pentru intuirea momentului in care poate sa apara interblocajul sau a momentului cind fenomenul nu a aparut inca dar nu mai poate fi evitat, s-a ales metoda reprezentarilor grafice. Rezulta importanta ordinei de repartizare a resurselor pentru evitarea fenomenului de interblocaj.

Se face o tratare teoretica a posibilitatilor de stabilire a starii unui sistem si a faptului daca acesta este sau nu in situatie de interblocaj. Apoi se dezvolta teoretic problema remedierii interblocajului. Exista doua tipuri de metode: metode de detectare si deblocare si metode de prevenire. S-a ales solutia prevenirii interblocajului ca metoda mai eficienta. Analiza teoretica s-a facut prin eliminarea din mecanismul de alocare a tuturor situatiilor care puteau conduce la interblocaj, pornind de la o stare initiala in care nu exista interblocaj.

La Sistemul de Operare se admiteau cîteodata situatii de interblocaj care puteau fi rezolvate prin anularea unor procese care au condus la interblocaj si reluarea lor intr-un context favorabil. In cazul Subsistemului de discuri magnetice privit ca un



sistem comun unei configuratii multiprocesor, interblocajul este un fenomen interzis, care trebuie evitat pentru a se asigura configuratiei o functionare continua. De aceea se va lucra cu metode de prevenire a interblocajului, metoda dinamica fiind cea mai utila. Fiecare proces va trebui sa faca o cerere initiala de resurse subsistemul fiind responsabilul evitarii interblocajului si a rezolvarii globale eficiente a acestor cereri.

1.2.8. Capitolul 8 analizeaza mecanismele de sincronizare, necesare sincronizarii microprogramelor si lucrarilor care se desfasoara simultan in cadrul Subsistemului de discuri magnetice. Se defineste notiunea de semafor (s), caracterizat prin variabila intraga  $e(s)$  si sirul de asteptare  $f(s)$ , precum si notiunile de primitiva P si primitiva V prin care se poate actiona asupra semaforului s. Se descriu proprietatile semafoarelor si operatiile de sincronizare ce se pot executa cu ajutorul acestora. Se evidentiaza semafoarele de excludere mutuala.

Intre diverse procese apar interactiuni datorate de exemplu utilizarii acelorasi resurse. Pentru a intelege mai bine aceste interactiuni si modul de actionare, s-au analizat tranzitiile posibile de stare ale proceselor. Operatiile de sincronizare s-au regrupat pe patru tipuri: creare, punere in asteptare, activare si incheiere.

In continuare se analizeaza mecanismele de sincronizare propriu-zise, mecanisme ce trebuie sa fie independente de viteza de executie. Se exemplifica mecanismul de sincronizare cu semafoare, alegindu-se doua operatii reale pe care trebuie sa le rezolve subsistemul si anume: operatia de "scriere initiala" a pachetului de discuri si operatia de comanda a unui microprogram prin intermediul altor microprograme generale sau speciale.

Rezolvarea problemei sincronizarilor specifice in cadrul Subsistemului de discuri magnetice, concura la realizarea unei independente reale a sistemului fata de restul configuratiei.

1.2.9. Capitolul 9 delimiteaza locul Subsistemului de discuri magnetice in configuratiile multiprocesor. Pentru studiul teoretic al problemei se defineste notiunea de "multiprocesor banalizat" ca un sistem cu o memorie unica partajata de mai multe procesoare fiecare din acestea putind executa toate procesele existente in sistem. Notiunea a fost extinsa prin utilizarea unui Subsistem de discuri magnetice unic, corect dimensionat pentru a face fata tuturor sarcinilor ce apar fata de sistem. Se grupeaza consecintele definitiei multiprocesorului

lui banalizat asupra structurii in urmatoarele grupe: accesul tuturor procesoarelor la toate resursele, autonomia proceselor fata de procesoare si autonomia proceselor unul fata de celelalte.

In continuare se analizeaza diverse configuratii organizate cu minicalculatoare INDEPENDENT-100, la care s-a urmarit cuplarea simultana a unitatilor de discuri magnetice la doua calculatoare. Se disting doua solutii si anume: realizarea unui grup de unitati de discuri magnetice care se comuta printr-o logica de comutare la cuploarele specifice apartinand sistemului de calcul si utilizarea unor comutatoare de busuri care permit cuplarea Subsistemului de discuri magnetice la sistemele de calcul.

Arhitectura configuratiilor multiprocesor a evoluat in concordanta cu dezvoltarea tehnologica a circuitelor VLSI. Odata cu aparitia noilor posibilitati de realizare a configuratiilor multiprocesor au aparut si noi probleme, ca de exemplu: modul de conectare al procesorului cu memoria si sistemul de intrare-iesire, care pot fi subansamble unice ale configuratiei. Dintre solutiile posibile de solutionare a problemei, varianta configuratiei cu busuri comune este mai simpla si mai ieftina. In arhitectura actuala sarcinile de intrare-iesire sint rezolvate de Unitatea Centrala. Performantele sistemului nu pot fi ameliorate substantial decit prin schimbarea conceptiei arhitecturale, un punct de interes fiind organizarea Sistemului de Intrare-Iesire cu autonomie in ceea ce priveste posibilitatile de calcul si gestiune a informatiilor. Problema definirii unei arhitecturi de intrare-iesire se apropie foarte mult de problema definirii arhitecturii retelei de calculatoare. Problema descentralizarii puterii de calcul, de fapt este o problema a marii puterii de calcul in ansamblu si permite o micorare a legaturilor cu procesoarele propriu-zise. Se analizeaza o arhitectura distribuita de intrare-iesire, organizata pe patru nivele de protocol si care utilizeaza mesaje de control orientate. In incheiere sint analizate avantajele implementarii acestor nivele cu microprocesoare VLSI urmarindu-se realizarea de configuratii cu fiabilitate sporita.

1.2.10. Pentru sistemele de calcul nu s-a studiat inca in exploatare, coeficientul real de utilizare al Subsistemului de discuri magnetice. Folosind o metoda originala s-au studiat rezervele subsistemului, stabilindu-se limitele pina la care considerentele de ordin economic nu

duc la diminuarea performanțelor tehnice. Performanțele pot fi diminuate mult de conflictele ce apar în momentul în care mai mulți utilizatori solicită simultan un subsamblu utilizat în comun. Problema conflictelor de acces este tratată în capitolul 10, capitol care începe cu analiza teoretică a problemei în cazul în care "n" procesoare împart accesul la același Subsistem de discuri magnetice. Procesoarele active pot să se găsească față de subsistem în una din următoarele stări: "execuție", "în așteptare" și "în acces". Notând cu  $\beta$  inversul valorii medii a timpului de acces la subsistem pentru un procesor dat, posibilitatea ca acest procesor să se afle "în acces" este dată de relația:

$$f(t) = \beta e^{-\beta t}$$

Notând cu  $\alpha$  inversul valorii medii a timpului care separă pentru un procesor dat un sfârșit de acces și cererea următoare de acces, posibilitatea ca procesorul să treacă din starea "execuție" în una din celelalte două stări este dată de relația :

$$f(t) = \alpha e^{-\alpha t}$$

Fornind de la aceste posibilități s-a stabilit formula de calcul a unui coeficient "k" de multiplicare al timpului de execuție, funcție de numărul mediu de procesoare care se afla în așteptarea unui acces.

Pentru verificarea părții teoretice s-a imaginat o metodă de măsură, utilizându-se un sistem monoprosesor împreună cu un montaj care simulează prezenta în configurație a unor procesoare suplimentare, prin simularea acceselor. Montajul a permis determinarea repartitiei duratei acceselor pe intervale de timp. Măsurătorile s-au făcut cu un sistem INDEPENDENT-100 pe care s-a rulat un lanț de programe alcătuit pentru recepționarea sistemului. Pe baza măsurătorilor s-au ridicat curbele acceselor libere, curbe de forma:

$$N = N_0 [1 - F(\Delta t)]$$

unde:  $N_0$  - valoarea la originea curbelor și  $F(\Delta t)$  funcția de distribuție a perioadelor de acces sau respectiv a perioadelor fără acces.

Măsurătorile s-au făcut pentru un subsistem cu unul sau două echipamente periferice. Rezultatele măsurătorilor

au confirmat justetea metodei de masura.

S-au facut masuratori suplimentare pe o configuratie biprocesor utilizand doua sisteme INDEPENDENT-100 cuplate la subsistem printr-un comutator de busuri, rezultatele fiind in concordanta cu cele obtinute prin calcule teoretice si la masuratorile facute cu simulatorul. Verificarea practica a analizei teoretice ne permite extinderea metodei de la sistemele biprocesor si la sistemele multiprocesor. Configuratiile obtinute permit o utilizare mai eficienta a echipamentelor periferice, fara ca performantele de ansamblu ale configuratiei sa aiba de suferit.

1.2.11. Capitolul 11 este un capitol de sinteza bazat pe analiza unor probleme puse in evidenta in capitolele precedente. Sint enumerate elementele Sistemului de Intrare-Iesire din Sistemul de Operare, urmarindu-se posibilitatea transferarii acestora in cadrul Subsistemului de discuri magnetice. Sint punctate probleme care pot fi studiate si dezvoltate in continuare la nivelul Subsistemului de discuri magnetice, in vederea maririi eficientei acestora.

1.2.12. Lucrarea se incheie cu o serie de concluzii grupate in capitolul 12. S-a cautat realizarea unui studiu complex al problemei cuplarii simultane a unitatilor de discuri magnetice la mai multe sisteme de calcul, urmarindu-se sistematizarea si punerea in evidenta a factorilor deja relevati in lucrarile de specialitate, precum si prelevarea unor factori noi care sa constituie elemente de baza pentru realizarea unui Subsistem de discuri magnetice eficient in contextul modernizarii arhitecturii sistemului de calcul. Nu s-a procedat la proiectarea subsistemului, scopul lucrarii fiind analiza teoretica aprofundata a domeniului, aparind aspecte originale, pentru anumite directii putindu-se gasi rezolvari viabile, in timp ce alte directii sint doar punctate in vederea unor rezolvari posibile ulterioare. Contributiile originale mai importante se refera la:

- prezentarea unitara a aspectelor teoretice si practice legate de cuplarea simultana a unitatilor de discuri magnetice la mai multe sisteme de calcul;

- studiul analitic al interfetelor Subsistemului de discuri magnetice in vederea realizarii unor interfete universale care sa permita cuplarea simultana;

- realizarea "grupului unic" si a "modulului virtual" in vederea utilizarii simultane a unitatilor de discuri mag-

netice de diverse capacitati;

- stabilirea unei metodologii de calcul a timpului de acces la informatie in vederea alegerii unor algoritmi de acces imbunatatiti;

- studiul la nivelul Subsistemului de discuri magnetice a strategiilor de alocare, a operatiilor de sincronizare cu semafoare si a evitarii fenomenului de interblocaj;

- elaborarea unei proceduri de stabilire in exploatare a coeficientului real de utilizare a Subsistemului de discuri magnetice si realizarea de masuratori pe un montaj de simulare si pe o configuratie reala biprocesor;

- prelevarea unor elemente din Sistemul de Operare care se pot transfera in cadrul Subsistemului de discuri magnetice in vederea realizarii unui subsistem cu o independenta functionala cit mai ridicata.

\*

\*

\*

Prezenta lucrare s-a desfasurat sub directa si permanenta coordonare stiintifica a Prof.Dr.Ing. Alexandru Rogoian personalitate care a influentat puternic asupra formarii autorului ca specialist in domeniul Tehnicii de Calcul inca din anul 1964, impreuna cu Conf.Dr. Strugaru si Conf. Dr. Pop Vasile.

Tin sa multumesc de asemenea tov.Gral.Dr.Ing. Dinu Buznea si Dr.Ing. Vasile Baltac oameni de stiinta care au detinut rolul de pionieri in acest domeniu si care au avut bunavointa sa-mi impartaseasca permanent din experienta lor. Multumesc si tatalui meu care m-a ajutat in momentele cind voiam sa renunt.

## **CAP.2. SUBSISTEMUL DE DISCURI MAGNETICE**

### *2.1. Definirea problemei*

Se defineste notiunea de "Subsistem de discuri magnetice" (SDM) ca ansamblul fizic format din unitatile de memorie cu discuri magnetice (UDM) (echipamentul periferic

propriu-zis), cuplorul pentru legarea UDM la sistemul de calcul, cablurile de legatura intre UDM si cuplor, precum si in anumite cazuri (ex: structurile microprogramate) asigurarea software specifica. SDM este parte constitutiva a oricarei configuratii de calcul care are in componenta sa UDM.

Necesitatea utilizarii notiunii de SDM a aparut odata cu modernizarea arhitecturii sistemelor de calcul, arhitectura cu multe parti componente, unde delimitarea unor subsisteme permite o abordare mai usoara a problemelor hardware si software ce apar. Un subsistem trebuie sa constituie un ansamblu de sine statator, cu interfata si cu anumite raporturi de cooperare bine definite fata de alte subsisteme. Aceasta conceptie permite o simplificare necesara studierii teoretice a configuratiilor complexe pentru gasirea unor solutii de interconectare optime si pentru minimizarea partilor aferente acestor interconectari din Sistemul de Operare (S.O.). Conceptia permite standardizarea magistralelor (in lucrare va fi utilizat in continuare termenul "bus", termen care s-a impus in literatura de specialitate) si realizarea unui principiu unic de interconectare. In acest fel se pot concepe configuratiile "fault-tolerante", la care aparitia unui defect intr-un subsistem sa poata fi ignorata prin "reconfigurarea" configuratiei si preluarea sarcinilor subsistemului defect de catre alt subsistem cu functii identice. Realizarea fizica a acestor concepte a devenit posibila datorita progreselor tehnologice inregistrate in domeniul circuitelor integrate in ultimii cinci ani. Microprocesoarele VLSI pe 16 biti si memoriile cu peste 64 Kbiti pe circuit integrat permit realizarea unor astfel de structuri la preturi de cost convenabile. Inglobarea in hardware a unor noi functii, duce la simplificarea asigurarii software si implicit la ieftinirea sa.

Problema cuplarii simultane a UDM la doua sau mai multe sisteme de calcul, se pune de asemenea diferit in conditiile actuale fata de modul posibil de rezolvare la nivelul sistemelor de calcul din generatia 3-a. Aceste sisteme utilizau pentru cuplarea echipamentelor periferice canale selectoare sau multiplexoare, realizarea unor cuploare cu dublu acces sau utilizarea unor UDM cu dublu acces fiind singurele solutii abordabile, solutii destul de simple din punct de vedere hardware, dar care au ridicat multe probleme software legate de "rezervarea" la un anumit moment a re-

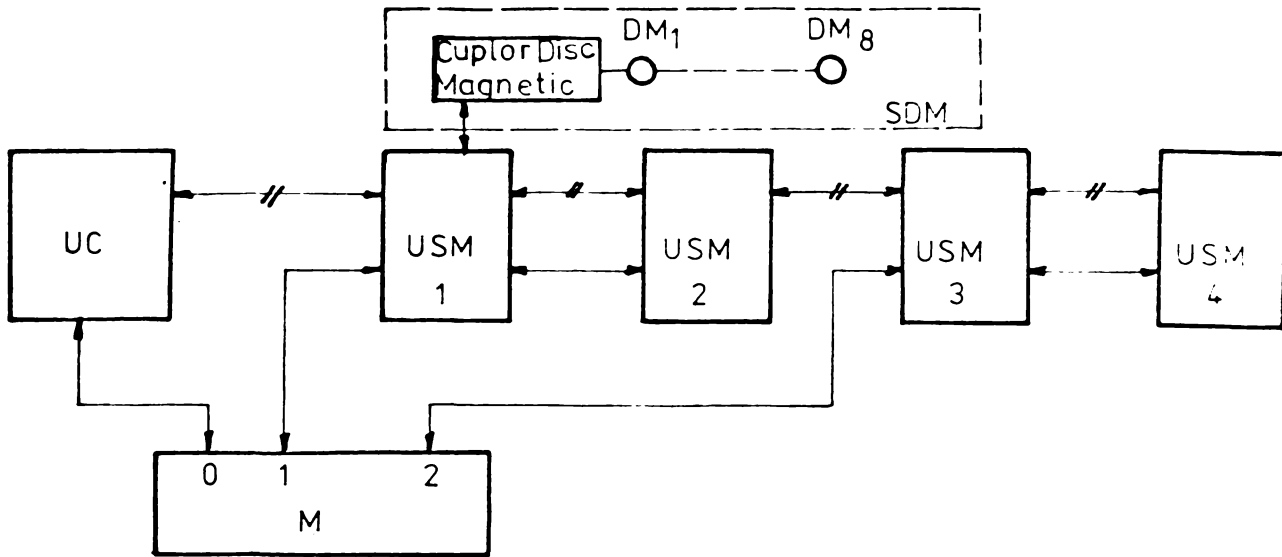


Fig. 2.1. Schema bloc a unei configuratii FELIX C-256.

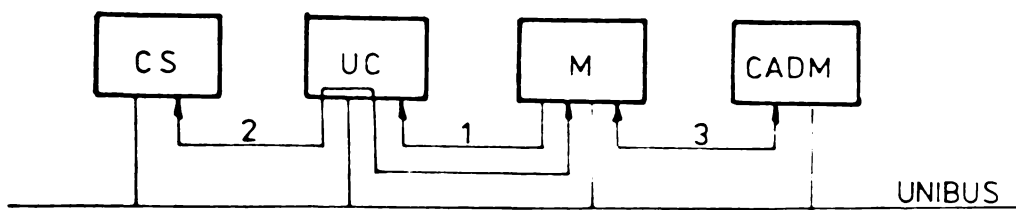
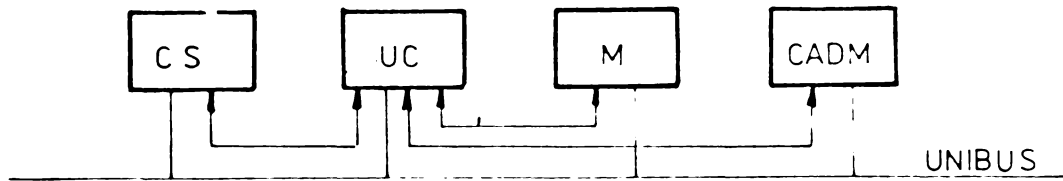


Fig.2.2.Schema bloc a unei configuratii cu bus comun (UNIBUS).

a. Schema transferului de comenzi si informatii.

b. Schema transferului de date (1.UC-M; 2.CS-M; 3.CADM-M)

576663  
35741

surselor si a impartirii capacitatii utile intre doua sisteme de calcul. UDM aveau o capacitate maxima pe unitate de pina la 200 Moct., iar un cuplor era conceput pentru a lega pina la opt UDM, deci se putea realiza o capacitate totala a memoriei externe cu discuri magnetice de 1600 Moct., suficienta pentru doua sisteme de calcul de tipul FELIX C-256. La noi in tara capacitatea maxima a unei UDM era in 1975 de 7 Moct., iar din 1975 pina in prezent UDM disponibile au capacitatea maxima de 58 Moct. In aceste conditii tinind cont si de pretul ridicat al unei UDM, nu s-a putut forma o capacitate de memorie cu discuri magnetice destul de mare, pentru a justifica utilizarea sa simultan de mai multe sisteme de calcul. Un alt impediment a fost ca tehnologia generatiei a 3-a era scumpa, astfel ca putini utilizatori au putut sa cumpere mai multe sisteme de calcul.

Aparitia sistemelor minicalculator care au fost plasate intr-o generatie intermediara, numita 3,5 a readus in actualitate problema cuplarii simultane a UDM la mai multe sisteme, deoarece aceste sisteme au un cost mult mai mic, iar capacitatile UDM satisfac pe deplin necesitatile de calcul ale acestora. Trecerea la generatia urmatoare, bazata pe tehnologia VLSI pastreaza in actualitate problema cuplarii si simultane care insa se poate rezolva cu rezultate mai bune. Avind in vedere implicatiile economice pozitive ale utilizarii metodei de cuplare simultane a UDM la mai multe sisteme de calcul, ne propunem sa analizam aceste probleme pentru gasirea unor rezolvari cit mai bune.

## *2.2. Definirea locului SDM in diversele sisteme de calcul.*

Sistemele din generatia 3-a, reprezentate la noi de sistemele de calcul FELIX C-32, 256, 512 si 1024, au fost sisteme cu logica cablata la care orice modificare se realiza foarte greu. In acest fel sistemele nu au putut fi modernizate prea mult in decursul celor 12 ani de fabricatie.

Sistemele FELIX utilizeaza o memorie (M) multiport (cu trei accese) folosita in comun de Unitatea Centrala (UC - utilizind un acces) si Unitatile de Schimb Multiplu (USM - maximum patru, cuplate cite doua pe un acces al memoriei). Dialogul intre UC si USM se realizeaza prin intermediul a cinci instructiuni de intrare-iesire. Echipamentele periferice se cupleaza la USM prin intermediul cuploarelor (denumite si unitati de legatura). Principiul utilizat este avanta-



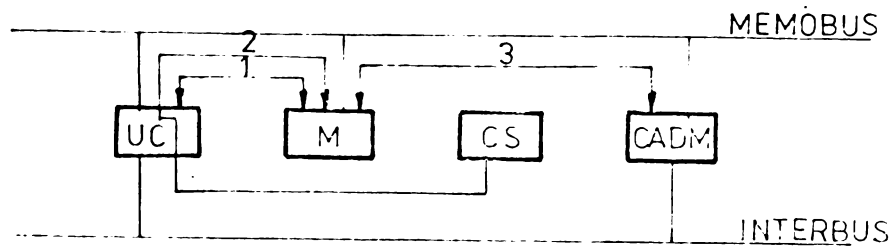
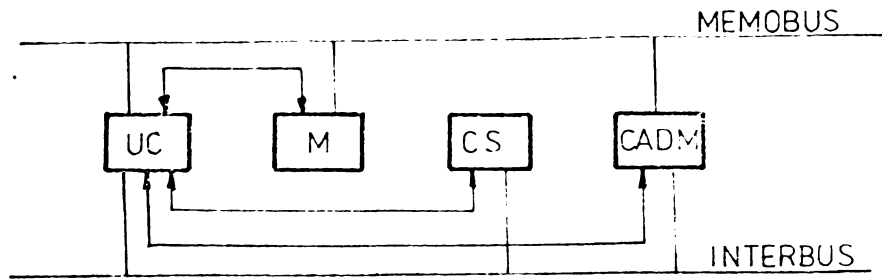


Fig. 2.3. Schema bloc a configuratiei I-100.  
a. Schema transferului de comenzi si informatii.  
b. Schema transferului de date (1.UC-M; 2.CS-M; 3.CADM-M)

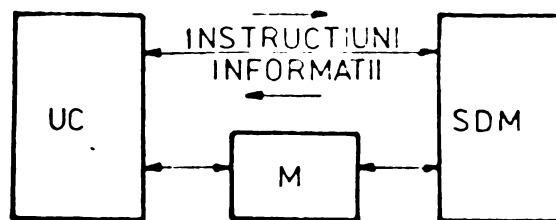


Fig. 2.4. Schema bloc a unei arhitecturi moderne.

jos, deoarece dialogul pentru realizarea unei operatii de intrare-iesire este facut cu instructiuni puternice, deci este simplu. In fig.2.1. este data schema bloc a unei configuratii FELIX C-256.

Aparitia minicalculatoarelor a impus arhitectura cu un singur bus, reprezentative fiind configuratiile familiei PDP 11 elaborate de firma DEC, construite in jurul busului UNIBUS (fig.2.2.). Aceasta arhitectura cu toate ca este simpla si ieftina, asigura performante bune configuratiei. UNIBUS-ul permite transferul de comenzi si informatii intre UC si cuploarele echipamentelor periferice, precum si transferul de date intre memorie si celelalte subansamble din configuratie. Dupa modul de transfer a datelor, cuploarele se pot imparti in doua categorii:

- cuploare cu acces direct la memorie (CADM) in care transferul se face direct intre cuploare si memorie;
- cuploare fara acces direct la memorie, cuplorul simplu (CS) transferul datelor fiind facut prin intermediul UC.

Configuratia este foarte flexibila permitind adaugarea sau scoaterea usoara a anumitor elemente. Registrele interne ale UC si ale cuploarelor au adrese proprii si sint privite ca locatii de memorie fiind adresate ca atare. Aceasta tehnica simplifica mult secventele de adresare permitind o utilizare buna a UNIBUS-ului.

La proiectarea in 1975 a minicalculatorului INDEPENDENT-100 (I-100) s-a utilizat o solutie originala si anume utilizarea unui bus suplimentar pentru memorie numit MEMOBUS si transformarea UNIBUS-ului intr-un bus interprocesoare INTERBUS (fig.2.3.) care permite realizarea de configuratii multiprocesor la care mai multe UC sint cuplate prin acelasi bus. Existenta unui bus separat pentru schimbul de date cu memoria are avantajul sporirii vitezei pe ansamblul configuratiei si implicit posibilitatea realizarii unor configuratii mai mari. Structura UC este microprogramata, deci foarte flexibila si permite realizarea de schimbari si imbunatatiri.

SDM ale configuratiilor din fig.2.2. si 2.3. sint compuse dintr-un cuplor CADM la care se cupleaza una sau mai multe UDM.

Datorita ultimelor cuceriri tehnologice si scaderii pretului de cost al componentelor, se pot concepe acum configuratii microprogramate, asemanatoare ca arhi-

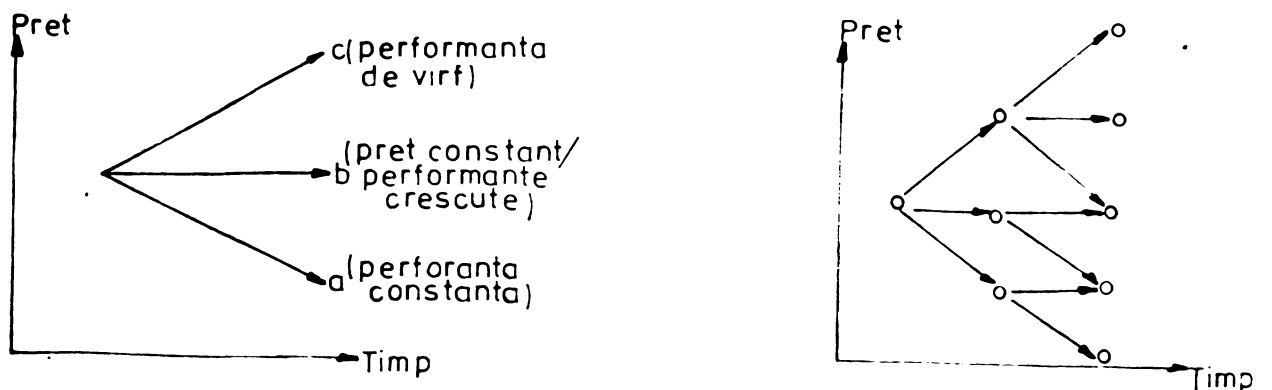


Fig. 2.5. Alternative de proiectare.

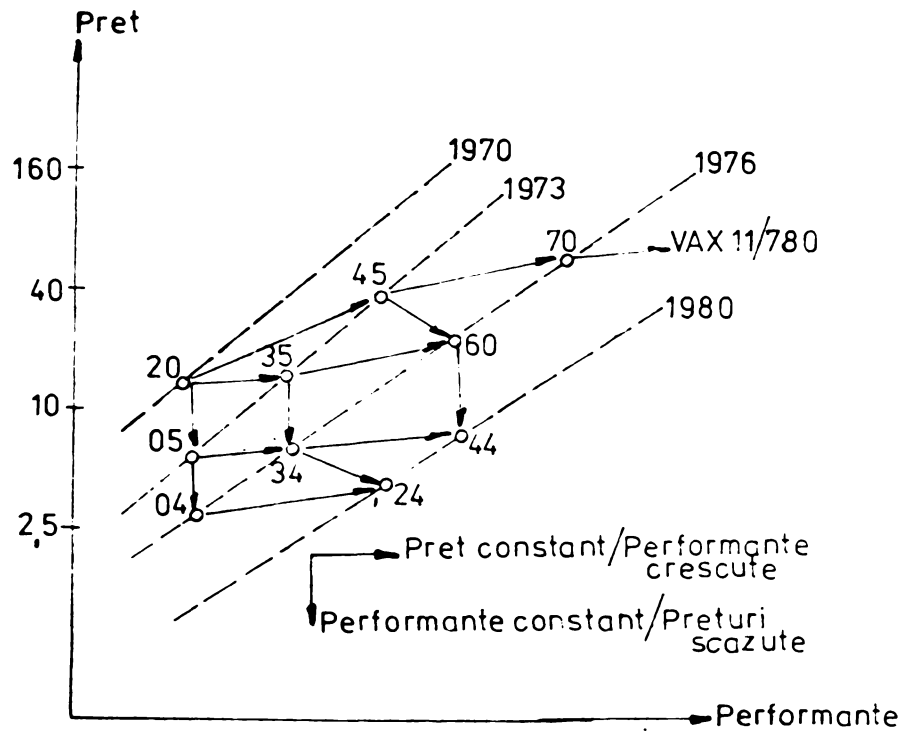


Fig. 2.6. Evolutia familiei PDP-11 (1970-1980).

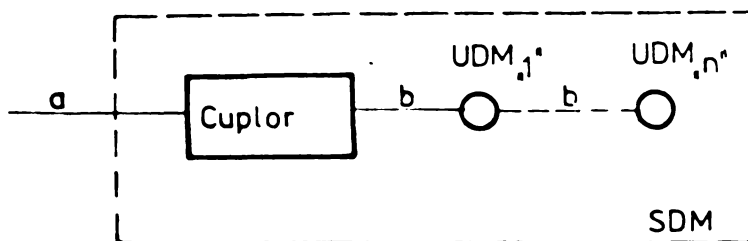


Fig. 2.7. Interfetele SDM.

a. Interfata externa.

b. Interfata interna

tectura cu sistemele generatiei a 3-a, in care SDM devine o parte independenta a configuratiei, dialogul cu UC facindu-se prin instructiuni de intrare-iesire puternice. SDM devine un procesor de intrare-iesire cu setul sau propriu de instructiuni si asigurarea software adecvata (fig. 2.4.). Aceasta solutie elibereaza UC care poate sa utilizeze intreaga sa putere de calcul pentru realizarea de calcule propriu-zise. De asemenea transferurile de informatii si comenzi fiind reduse se pot simplifica si busurile sistemului.

### 2.3. *Consideratii tehnologice si economice*

Dezvoltarea calculatoarelor a fost influentata de trei factori:

- tehnologia circuitelor logice si de memorie;
- arhitectura si organizarea tehnologica;
- locul care a putut fi ocupat pe piata de modelul respectiv.

Datorita dezvoltarii tehnologice a circuitelor semiconductoare a crescut continuu eficacitatea lor, in timp ce costul s-a diminuat. Pentru proiectantii de sisteme de calcul exista trei alternative de proiectare (fig 2.5.):

- a). pastrarea constanta a performantelor cu scaderea pretului;
- b). pastrarea constanta a preturilor cu cresterea performantelor;
- c). urmarirea performantelor de virf cu cresterea corespunzatoare a preturilor.

La sistemele de minicalculatoare INDEPENDENT s-a utilizat curba pretului constant (b), obtinandu-se o crestere a performantelor, trecindu-se de la I-100 la I-102F. O ilustrare mai completa a strategiei alternativelor de proiectare apare prin urmarirea evolutiei familiei PDP-11 elaborate de firma DEC (fig.2.6.).

Evolutia tehnologica a circuitelor integrate implica schimbări in întreaga tehnologie a sistemului de calcul. Prin mărirea gradului de integrare, scade numărul total de circuite, deci consumul de energie și implicit se micșorează sursele de alimentare și sistemul de ventilare, elemente care au o mare pondere în pretul sistemului de calcul.

Referitor la evolutia tehnologica a SDM la sistemele romanesti s-au parcurs urmatoarele etape:

- la sistemele FELIX cuplorul de discuri ocupa o

poarta tehnologica cu sursa de alimentare si ventilatie separata, fiind compus din 5 sertare a 32 plachete cu o medie de 15 circuite integrate simple;

- la sistemele I-100 cuplorul a fost conceput initial pe patru placi si apoi pe doua placi a cite 90 circuite integrate;

- in prezent se omologheaza pentru sistemul INDEPENDENT un cuplor cu numai 80 circuite integrate, pentru UDM cu capacitatea de 58 Moct. si pentru o UDM originala cu frecventa de inregistrare dubla, deci capacitatea de 100 Moct.

#### 2.4. *Definirea interfetelor SDM*

Prin "interfata" se intelege ansamblul de semnale si de legaturi fizice ce se utilizeaza pentru cuplarea a doua subansamble si care asigura perfecta lor functionare in ansamblu.

La SDM se pot distinge doua interfete si anume:

- interfata interna - intre UDM si cuplor;
- interfata externa - intre SDM si sistemul de calcul propriu-zis.

### CAP.3. INTERFATA INTERNA A SDM

#### 3.1. *Grupe functionale*

Interfata interna este impusa in general de producatorii de UDM care in scopul pastrarii unor pietee de desfacere au diversificat aceasta interfata. Datorita inasa functiilor fixe pe care trebuie sa le indeplineasca UDM, semnalele de interfata sint aceleasi si diversificarea a constat in utilizarea unor conectori diferiti si in modificarea secventei de dialog intre UDM si cuplor. In ultima perioada se observa o micorare a numarului interfetelor utilizate la UDM. Varianta optima este stabilirea unei interfete interne universale care sa permita utilizarea oricarui tip de UDM, cu utilizarea unor adaptoare de cuplare. UDM cu capacitatea cuprinsa intre 7 si 58 Moct. sint identice din punct de vedere functional, de asemenea UDM cu capacitatea de 100, 200 si 300 Moct. pot constitui o gru-

pa functionala aparte. Noile UDM de tip Winchester constituie o alta grupa functionala. Se pune problema realizarii unui cuplor care sa permita cuplarea oricarei UDM in diferent din ce grupa functionala face parte. De mentionat ca se doreste si este posibila cuplarea UDM din aceeaasi grupa functionala la un cuplor, dar ca nu este posibila cuplarea simultana a UDM din grupe functionale diferite.

O alta problema care poate apare este interfata interna care trebuie sa rezolve problema cuplarii UDM cu dublu acces, deci existenta unor semnale suplimentare care sa ateste posibilitatea ocuparii si ocuparea propriu-zisa a UDM de catre cuplorul care poseda interfata interna in discutie.

Organizarea unei grupe de UDM cuplate la un cuplor ridica anumite probleme care trebuiesc analizate teoretic in vederea proiectarii interfetei interne, putindu-se obtine o ameliorare a randamentului SDM.

### 3.2. Organizarea interfetei interne a SDM

Pentru punerea in evidenta a semnalelor interfetei interne, se va face o analiza pe baza UDM elaborata la I.T.C. cu capacitatea de 29 Moct., denumita DM-29. Dialogul intre UDM si cuplor este aproximativ acelasi pentru toate UDM, mici diferente aparind la diagramele de timp si citeodata la valoarea semnalelor de interfata.

Semnalele de interfata se pot imparti in doua categorii:

- semnale comune tuturor UDM din grup, semnale grupate intr-un singur cablu denumit cablu A sau "COMUN";
- semnale specifice unei singure sau citeodata mai multor UDM din grup, grupate in alt cablu denumit cablu B sau "SIMPLEX", (de exemplu: UDM de 29 Moct. este considerata jumătate dintr-o UDM de 58 Moct., pe un cablu SIMPLEX putind fi conectate doua UDM de 29 Moct. sau o singura UDM de 58 Moct., denumita si unitate cu "dubla densitate" datorita densitatii duble de inregistrare a pistelor).

In cablul A se gasesc opt linii pe care se pot trimite in ambele directii informatii, a caror semnificatie este conditionata de existenta unor semnale de strobare pe una din liniile de strobare. Semnalele de strobare sint: control, selectare cilindru, selectare cilindru citit, selectare diferenta, selectare cap si directie de miscare a ansamblului de capete. Tabelul 3.1. prezinta semnificatia

TABELUL 3.1.

---

Strob li-			SELECT	SELECT	SELECT
nia bidi-	CONTROL	SELECT	CILINDRU	CAP SI	DIFE-
rectionala		CILINDRU	CITIT	DIRECTIE	RENTA

---

0	VALIDARE SCRIERE	Bit 128	Bit 128	inainte	Bit 128
1	VALIDARE CITIRE	Bit 64	Bit 64	inapoi	Bit 64
2	INITIALIZARE (MISCARE)	Bit 32	Bit 32	---	Bit 32
3	STERGERE REGISTRU Adresa cap	Bit 16	Bit 16	Bit regis- tru cap 16	Bit 16
4	VALIDARE STERGERE	Bit 8	Bit 8	Bit regis- tru cap 8	Bit 8
5	SELECTARE CAP	Bit 4	Bit 4	Bit regis- tru cap 4	Bit 4
6	STERGERE REGISTRU	Bit 2	Bit 2	Bit regis- tru cap 2	Bit 2
7	AVANS CAPETE(+1)	Bit 1	Bit 1	Bit regis- tru cap 1	Bit 1

---

Numarul in	1	2	3	4	5
fig. 3.1.					

---

semnalelor pe liniile de informatie in functie de strob ales.

Schema bloc simplificata a UDM (fig.3.1.) pune de asemenea in evidenta semnalele pe liniile de informatie in legatura cu semnalul de strob ales.

Dialogul dintre cuplor si UDM incepe prin trimiterea de catre cuplor a numarului logic al UDM cu care se doreste cuplarea, semnalele fiind trimise tuturor UDM, deci pe cablul A. Fiecare UDM are atribuit un numar logic care este cablat in interiorul unitatii. Liniile de adresa sint strobate de semnalul UNIT SELECT, iar coincidenta adreselor produce la UDM semnalul UNITATE SELECTATA, care fiind specific numai UDM selectate va fi trimis catre cuplor prin cablul B acesta fiind momentul inceperii dialogului pe liniile bidirectionale de informatie.

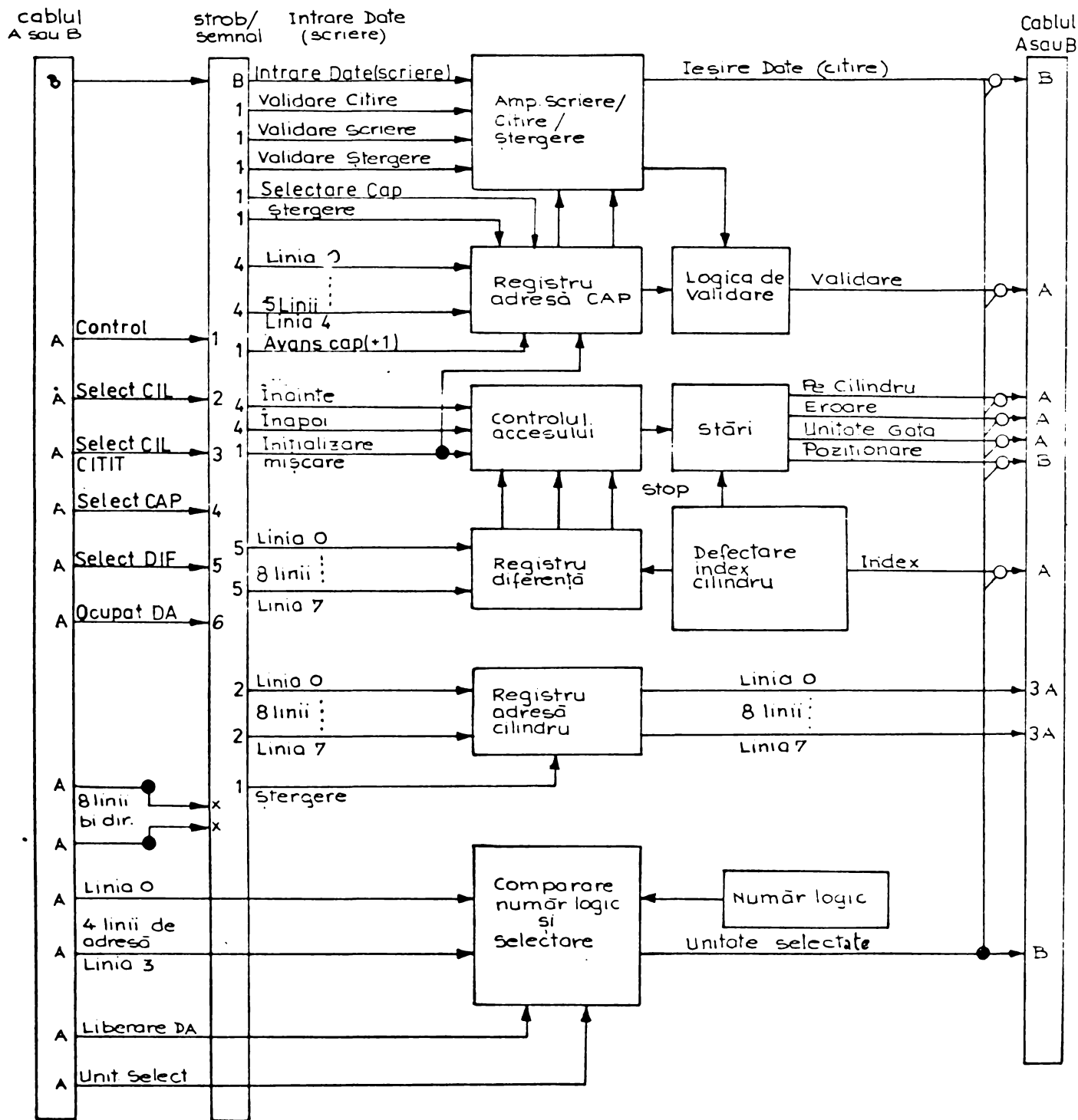
Dialogul initial are scopul positionarii ansamblului de capete pe cilindrul dorit si selectarea capului de scriere-citire dorit. In acest scop prima operatie este informarea asupra cilindrului pe care se gasesc capetele, apoi se calculeaza diferenta intre aceasta pozitie si pozitia dorita, precum si sensul miscarii. Astfel cuplorul va trimite UDM selectate in ordine: noua adresa a cilindrului care in secventa 1 este inregistrata in Registrul Adresa Cilindru; numarul capului cu care se va lucra, ce se inregistreaza in Registrul Adresa Cap impreuna cu directia in care urmeaza sa se deplaseze ansamblul de capete (secventa 2-a); diferenta de piste ce urmeaza a fi parcurse care se inregistreaza in Registrul Adresa Diferenta (secventa 3-a).

Secventa de lucru propriu-zis (secventa 4-a) se desfasoara sub incidenta semnalului de strob CONTROL care permite: initializarea positionarii; o operatie de scriere, citire sau stergere; stergerea Registrului Adresa Cap sau Registrului Adresa Cilindru; selectarea capetelor; incrementarea cu unu a Registrului Adresa Cap (+1).

In fig.3.2. este dat un exemplu de dialog intre cuplor si o UDM. Se observa ca in cazul scrierii si citirii trebuiesc luate masuri de siguranta care sa permita stabilizarea semnalelor de scriere, respectiv de citire. Pentru aceste semnale se utilizeaza cablul B.

De asemenea in cadrul dialogului se trimit o serie de semnale care dau indicatii asupra starii UDM si asupra erorilor care pot apare in timpul functionarii. Astfel ter-





- A . Cablul principal COMUN
- B . Cablul secundar individual (scriere, citire, selectare) - SIMPLEX
- x - 2,3,4,5
- DA - Dublu acces .

Fig. 3.1. - Schema bloc simplificată a UDM

minarea operatiei de pozitionare este indicata de semnalul PE CILINDRU, faptul ca UDM este operationala si cu capetele pozitionate este pus in evidenta de semnalul UNITATE GATA; faptul ca ceva nu este in ordine si ca dialogul nu mai poate avea loc este semnalat prin semnalul EROARE; semnalul INDEX permite calcularea vitezei de rotatie a pachetului de discuri si daca viteza este in limitele admise.

### 3.3. *Dublul acces*

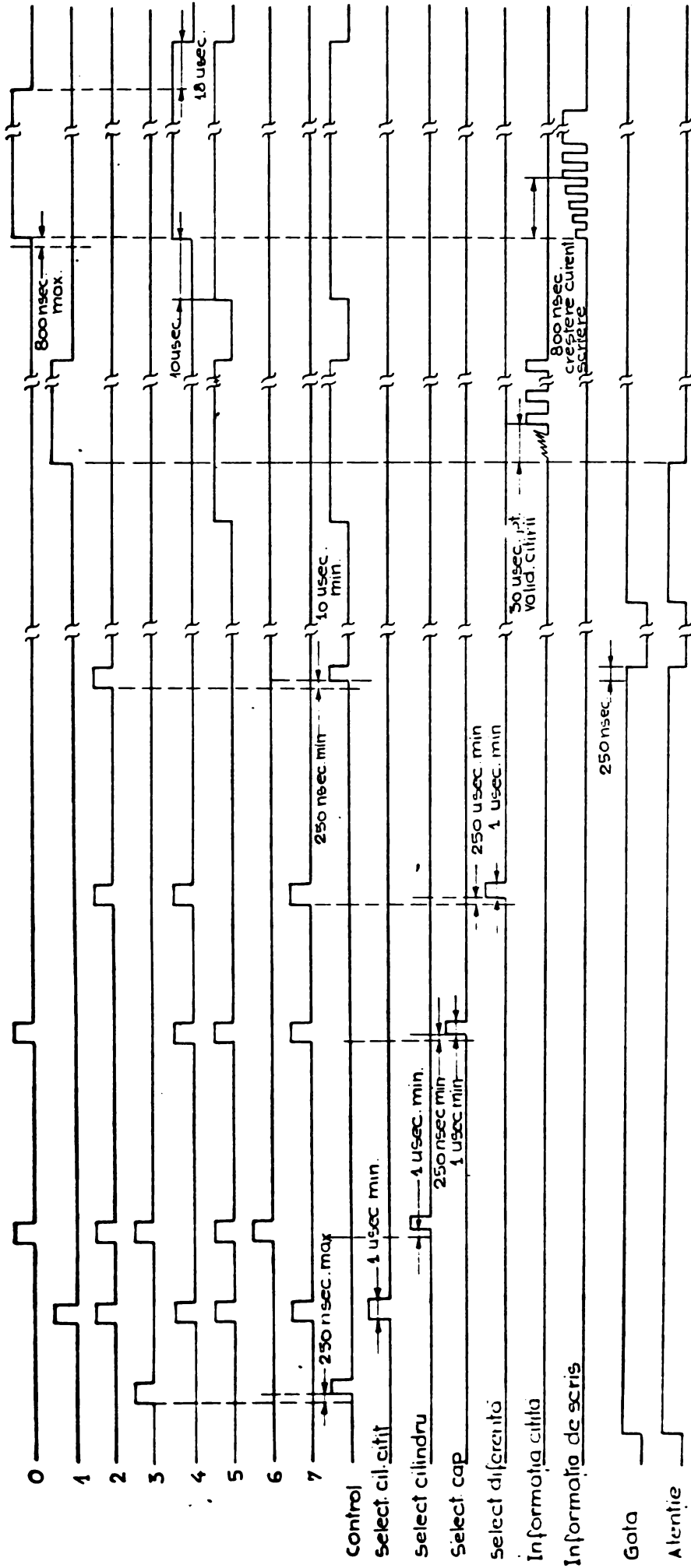
Exista UDM care au un acces suplimentar, toate circuitele de interfata fiind dublate, putindu-se realiza cuplarea unui asemenea UDM la doua cuploare. Aceasta UDM se numeste "unitate cu dublu acces" (UDA) sau "modul", deoarece fiecare cuplor poate considera ca intreaga UDM ii apartine, cu singura restrictie ca unitatea nu poate fi folosita simultan de cele doua cuploare. Din aceasta cauza apar doua noi semnale de interfata si anume "rezervare" cind se cere utilizarea UDA si "liberare" cind utilizarea sa nu mai este necesara. O unitate "rezervata" de un cuplor este "ocupata" pentru celalalt cuplor. De asemenea exista o prioritate de rezervare a UDA de catre cuploare, la aparitia simultana a cererii de rezervare unul din cuploare fiind preferat.

Dublul acces poate fi folosit in doua moduri: dublul acces monocalculator (DAM) cind cele doua cuploare apartin aceluiasi sistem de calcul si dublul acces bicalculator (DAB) unde fiecare cuplor apartine cite unui sistem de calcul separat.

Lucrul cu dublul acces este asemanator celui monocacces, aparind insa anumite restrictii: semnalul "sfirsit de pozitionare" la DAB poate fi dat numai de sistemul de calcul care a facut rezervarea, iar la DAM de cuplorul care a facut rezervarea; intreruperea de liberare la DAB se emite spre sistemul de calcul caruia i s-a refuzat un acces pe o UDA rezervata de celalalt calculator, la DAM este posibil sa se lucreze fara rezervare si liberare la sfirsitul fiecarui secvente de selectie unitatea fiind libera si disponibila pentru oricare cuplor.

### 3.4. *Cuplarea simultana a UDM cu diverse capacitati de inregistrare*

Cuploarele existente sint concepute pentru a anumita UDM, solutie rigida care ingreuneaza mult extinderea configuratiei. Din acest motiv gasirea unei solutii de cuplare simultane a UDM cu diverse capacitati de inregistrare prezin-



<p>Ex: 109<sub>10</sub> (64 + 32 + 8 + 4 + 1) Select Cilindru Citat</p>	<p>Ex: 182<sub>10</sub> (128 + 32 + 16 + 4 + 2) Select Cilindru</p>	<p>Ex: Cap 13<sub>10</sub> (8 + 4 + 1) Înainte Select Cilindru</p>	<p>Ex: 77<sub>10</sub> (64 + 8 + 1) Select Diferența</p>	<p>Formționare</p>	<p>citire</p>	<p>Scriere</p>
---	---	--	--	--------------------	---------------	----------------

Fig. 3.5 - Exemple de dialog cuplor UDM

ta un mare interes. Problema poate fi abordata prin realizarea unui cuplor pentru o UDM cu o capacitate mare pe care o vom numi "majora" si realizarea unor "module virtuale" de capacitate "majora", module constituite din UDM de capacitate mai mica, submultiplu al capacitatii majore.

De exemplu daca se alege capacitatea majora 200 Moct., modulul virtual poate fi constituit din doua UDM de 100 Moct. sau la alegerea capacitatii majore 58 Moct., se pot constitui module virtuale din UDM de 29 Moct. si/sau 7,25 Moct.; situatie aratata in Tabelul 3.2.

TABELUL 3.2.

Capacitate majora	UDM fizica	Echivalent virtual
200 Moct	200 Moct	(100 + 100) Moct
58 Moct	58 Moct	(29 + 29) Moct (29 + 4x7,25) Moct (8x7,25) Moct

Daca interfata interna este aceeași pentru toate UDM singura problema care apare este problema adresării UDM din componenta modulului virtual. Pentru adresarea UDM in cadrul modulului virtual se folosesc liniile de adresare obisnuite, insa dialogul cuplor-UDM se modifica, incepind cu un semnal suplimentar de chestionare daca in lantul de UDM exista un modul virtual. Acest semnal are si rolul de strobare, pe liniile de adresa putindu-se trimite adresa UDM cu care se doreste sa se lucreze. In schema se va introduce un "adaptor" AD, care are rolul memorării acestei adrese si apoi de a trimite semnale permanente de adresare catre UDM selectionate din cadrul modulului virtual. Aceasta operatie se face pe liniile de adresare normala a UDM in momentul inceperii dialogului intre cuplor si UDM, in cazul in care AD respectiv a fost adresat (AD are un circuit de adresare asemanator celui din UDM). Daca in schema exista un AD, la dialogul cuplorului-UDM se adauga un semnal de raspuns din partea adaptorului, semnal care va fi utilizat de cuplor pentru ca interfata interna sa lucreze cu rata de transfer specifica UDM componenta a modulului virtual (UDM pot avea rate de transfer diferite). Schemele de conectare a modulelor virtuale

Caracteristici principale ale UDM

TABELUL 3

7 Moct	I.B.M.	156	2400	200+3	10	1100	100	10	10	25	1,5	1560	1950	-	3650	3650		
	Sistemul FELIX	156	2400	200+3	10	1100	100	10	10	25	1,5	1560	1950	256	3070	3070		
29 Moct	I.B.M.	312	2400	200+3	20	2200	100	7	10	25	3,1	2132	3900	-	7294	1458		
	Sistemul FELIX	312	2400	200+3	20	2200	100	7	10	25	3,1	2192	3900	1024	6144	1228		
58 Moct	CDC-AMPEX	312	2400	400+4	20	2200	200	7	10	25	3,1	2132	3900	-	7294	1458		
	Sistemul FELIX	312	2400	400+4	20	2200	200	7	10	25	3,1	2192	3900	1024	6144	1228		
100 Moct	IBM-CDC	300	3600	400+4	19+1	4040	200	7	10	16,67	3	5600	6400	644	15450	29400		
200 Moct	CDC-AMPEX	300	3600	300+3	19+1	4040	200	7	10	16,67	3	5600	6400	644	15450	29400		
300 Moct	CDC	1200	3600	300+3	19+1	6060	400	7	10	16,67	12	3400	9600	-	23175	44000		
	Capacitatea																	
	Compatibil cu UDM																	
	Viteza de transfer (Koct/sec)																	
	Turație nominală (rot/min)																	
	Număr piste																	
	Numere capete																	
	Info. (bpi)																	
	Piste (tpi)																	
	Pistă-Pistă (msec)																	
	Cap-Cap (usec)																	
	Timpul unei rotații (msec)																	
	Comutării Cap-Cap																	
	Comutării Pistă-Pistă																	
	Semirotatie																	
	Sector																	
	Pistă																	
	Cilindru																	
	Capacitate de stocaj (oct)																	
	Număr de octeți echi- valent																	
	Timp maxim																	
	Densitate																	

sint date in fig.3.3. si fig.3.4.

Cuplarea UDM de diverse categorii deschide problema optimizarii accesului la informatie. Astfel in cadrul unui modul virtual format din doua UDM informatia poate fi stocata alternativ adica: in timp ce o UDM executa scrierea sau citirea informatiei, a doua UDM va fi positionata pentru a se putea apoi continua transferul de informatie fara asteptari datorate positionarilor mecanice. In tabelul 3.3. care reuneste principalele caracteristici ale unor UDM sint trecute si rezultatele calculelor care indica numarul corespunzator de octeti ce pot fi transferati in timpul echivalent timpului de comutare a doua capete; timpul de acces de la o pista la pista alaturata si timpului de executare a unei semirotații a pachetului de discuri.

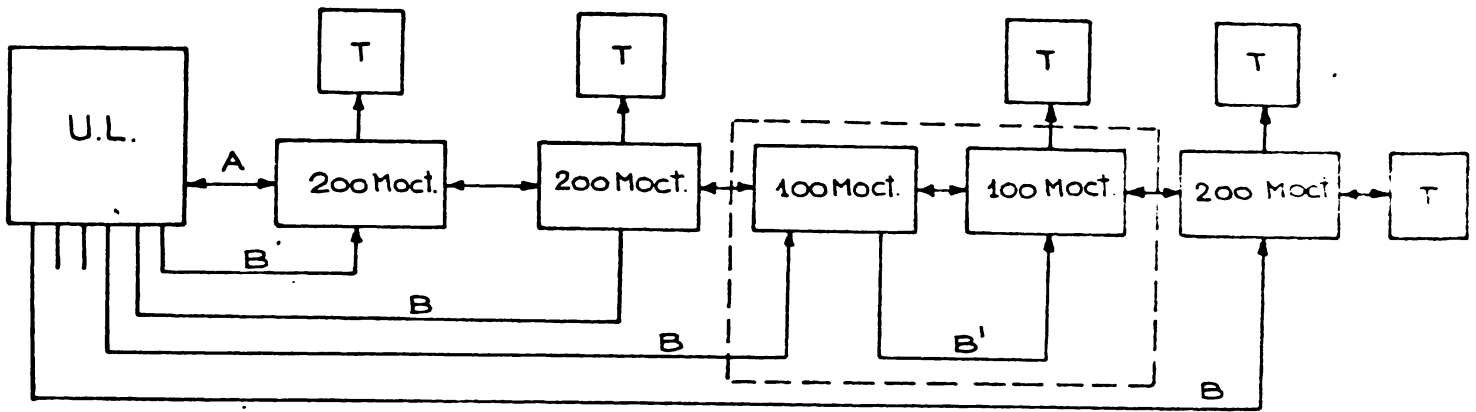
In cazul pachetelor de informatie care depasesc capacitatea unui cilindru, prin optimizare se cistiga 7 msec. (timpul de acces pista-pista) adica 2180 octeti pentru UDM din categoria 58 Moct. si 5600 octeti pentru UDM din categoria 200 Moct. Acest cistig permite sporirea randamentului de ansamblu al SDM.

### 3.5. *Concluzii*

Dezvoltarile tehnologice actuale au permis atingerea unor densitati de inregistrare de ordinul 12000 bpi si a unei densitati a pistelor de 1000 tpi pentru UDM aflate in productie de serie. Astfel exista UDM de tipul Winchester cu capacitatea de 200 Moct. care utilizeaza pachete cu numai 5 suprafețe de inregistrare plus o suprafata de sincronizare. Principiul de functionare al UDM si semnalele pe interfata interna au ramas insa aceleasi. Acest fapt simplifica realizarea unor cuploare care sa permita cu mici modificari cuplarea diferitelor UDM. In acest fel un efort de proiectare initial poate fi util pe parcursul mai multor generatii de UDM.

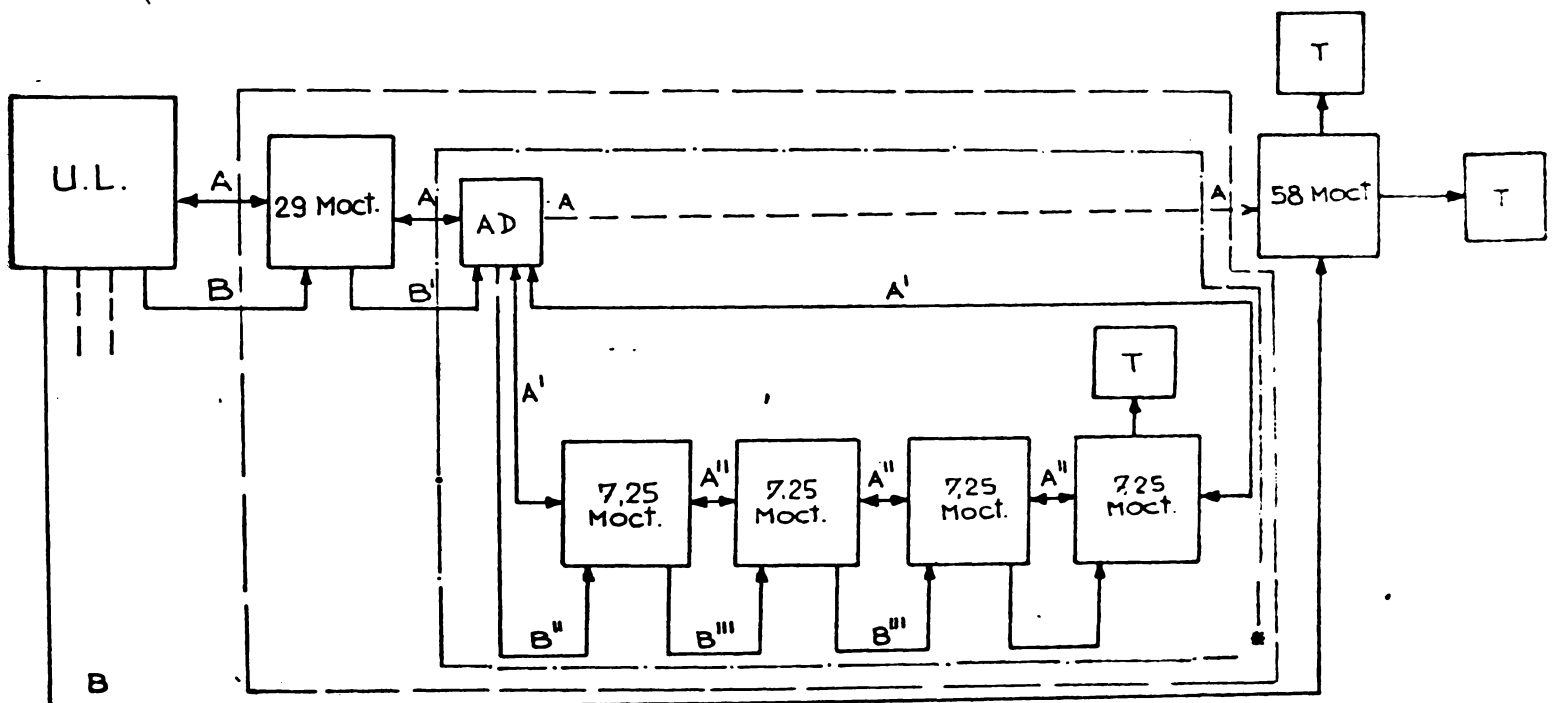
Realizarea unor "module virtuale" de capacitate mare din UDM de capacitate mai mica, aduce posibilitatea utilizarii simultane a UDM de capacitati diferite, reutilizarii UDM fabricate cu o tehnologie mai veche alaturi de UDM mai moderne, precum si optimizarea accesului la informatie.

Analizarea diverselor probleme legate de interfata interna este ceruta de cunoasterea detaliata a tuturor aspectelor ce se pun la cercetarea unui nou cuplor care sa permita cuplarea simultana a UDM la mai multe sisteme de calcul.



A - cablul „bus”  
 B - cablul „simplex” UL-UDM  
 T - terminator  
 B' - cablul „simplex” UDM-UDM (folosit la UDM cu „simpla densitate” a pistelor)

Fig. 3.3. Conectare la cuplor a UDM de 100Mocht. si 200 Mocht.



AD - adaptor care permite formarea unui „modul virtual” de 29 Mocht. sau 58 Mocht.  
 A - cablul „bus”  
 B - cablul „simplex”  
 T - terminator  
 A' - cablul „bus AD-UDM 7,25 Mocht. (acelasi cu cablul „bus” normal UL-UDM la cuplorul de 7,25 Mocht.)  
 A'' - cablul „bus” UDM-UDM 7,25 Mocht.  
 B' - cablul „simplex” UDM-UDM 29 Mocht.  
 B'' - cablul „simplex” AD-UDM 7,25 Mocht. (acelasi cablu „simplex” normal UL-UDM la cuplorul de 7,25 Mocht.)  
 B''' - cablul „simplex” UDM-UDM 7,25 Mocht.  
 \* - se pot cupla 1 la 4 UDM de 7,25 Mocht.  
 ---- - limita modulului virtual de 58 Mocht.  
 - - - - - limita modulului virtual de 29 Mocht.  
 Fig. 3.4. Cuplarea „modulului virtual” de 29 Mocht. format din UDM de 7,25 Mocht.

## CAF.4. INTERFATA EXTERNA A SDM

### 4.1. Clasificarea cuploarelor componente ale SDM

Intr-un sistem de calcul este avantajos ca subansamblele componente sa functioneze asincron, mod de functionare care permite realizarea schimbului de informatii cu minim de comenzi, precum si lucrul optim al ansamblului prin optimizarea fiecarui subansamblu in parte. In acest mod de functionare este posibil ca fiecarui subansamblu sa i se dea o autonomie cit mai mare, daca este posibil chiar sa se inglobeze in hardware o serie de functiuni care erau rezolvate pina acum de software-ul specific. Astfel in ansamblul unui sistem de calcul, SDM poate sa apara ca un subsistem de sine statator care sa fie coordonat de calculatorul propriu-zis prin citeva ordine simple, dialogul respectiv facindu-se intr-un timp minim. Dar pentru realizarea acestor deziderate este necesar sa se proiecteze un cuplor adecvat acestor functii noi, precum si sa se stabileasca o interfata externa cit mai simpla care sa asigure dialogul SDM cu restul subansamblelor sistemului.

Din punct de vedere constructiv cuploarele pentru UDM concepute pina in prezent pot fi clasificate in urmatoarele tipuri:

a. Cuploarele in tehnologia cablata, ca cele utilizate la calculatoarele din familia FELIX C-256 (IRIS-50), unde legatura cuplorului se face la un canal multiplexor sau selector care la rindul sau este cuplat cu procesorul si memoria.

b. Cuploare microprogramate ca cele utilizate in sistemele IBM-360, IBM-370, CDC-6600, AMPEX-830, etc., cuploare mai evolute care pe linga transferul normal al informatiilor permit in paralel detectia si corectia erorilor precum si rularea unor programe de diagnosticare a uneia din UDM aflate in regim OFF-LINE. In general pentru conectarea cuploarelor la acest tip s-a utilizat canalul selector.

c. Cuploarele cu structura fixa sau microprogramata utilizate in sistemele minicalculator, cu cuplarea la busurile generale ale sistemului, cum se utilizeaza la sistemele PDP 11 ale firmei DEC si sistemele INDEPENDENT si CORAL produse de I.C.E.

d. Cuploare derivate din minicalculatoare cum este



cazul sistemului IRIS-50 la care pentru SDM de 100 Moct. s-a utilizat minicalculatorul MITRA-15, solutie de compromis care a scurtat timpul de elaborare insa a scumpit produsul final, care era supradimensionat pentru sarcinile pe care le avea de indeplinit in epoca respectiva. SDM astfel realizat se cupleaza fie pe canalul multiplexor sau selector, fie direct pe unul din accesele memoriei operative.

La nici unul din cuploarele enumerate nu s-a realizat trecerea unei parti din software-ul specific in structura cuplorului propriu-zis. De asemenea interfata externa a SDM nu a fost studiata si optimizata separat, fiind similara cu interfetele utilizate de toate cuploarele din sistemul respectiv.

#### 4.2. Operatiile de baza executate de cuplorul din SDM.

UDM conectate la cuplor pot executa urmatoarele operatii:

- positionarea capetelor de scriere/citire pe o anumita pista;
- scrierea de informatii;
- citirea de informatii;
- cautarea unei inregistrari anumite;
- trimiterea informatiilor de stare.

Operatiile sint initiate de anumite ordine (vezi 4.4.), de obicei componente ale unui cuvint de comanda asociat unei instructiuni de intrare-iesire.

O operatie necesara este "premarcarea" sau "scrierea initiala" a pachetului de discuri, operatie prin care suprafata utila de inregistrare a pachetului de discuri este impartita in sectoare fizice, fiecare sector fiind delimitat de inceputul unui pachet de informatii plasat la inceputul sectorului prin care sectorul primeste o adresa proprie, sfirsitul sectorului fiind delimitat de inceputul sectorului urmator. Operatia de premarcare se face o singura data la prima utilizare a unui pachet nou de discuri si dureaza aproximativ 3-15 minute in functie de tipul si capacitatea UDM utilizate. La elaborarea unui cuplor nou trebuie sa se ia in considerare aceasta operatie, care este de dorit sa se prezinte ca un program ce se executa in paralel cu subprogramele de prelucrare a informatiilor.

Utilizindu-se o schema bloc simplificata a unui SDM microprogramat cuplat la sistemul de calcul prin intermediul unui canal selector sau multiplexor, se vor detalia fazele ce trebuiesc parcurse pentru executarea unor operatii impuse.

#### 4.2.1. *Operatia de pozitionare a capetelor de scriere/citire (SEEK)* .

Cuplorul trebuie sa selecteze UDM care urmeaza sa execute operatia ceruta, determina pozitia actuala a capetelor de scriere/citire, precum si directia si distanta ce trebuie parcursa pina la noua pozitie. Operatia de pozitionare poate fi defalcata in urmatoarele faze (fig.4.1.), grupate pe subansamblele functionale ale sistemului:

- A (1). Instructiunea de intrare/iesire
- B (1). Executarea cuvintului de comanda, transferul comenzii de pozitionare
  - (2). Transferarea spre cuplor a adresei de pozitionare
  - (3). Deconectarea pe timpul cit se face pozitionarea si reconectarea dupa terminarea acesteia.
- C (1). Controlul semnalelor pe interfata externa
  - (2). Controlul paritatii semnalelor transferate
- D (1). Decodificarea comenzii de pozitionare si controlul transferului informatiei de pozitionare
  - (2). Selectarea adresei UDM
  - (3). Prelevarea microprogramului de pozitionare
  - (4). Calculul diferentei dintre adresa prezenta si noua adresa
  - (5). Preluarea starii discului si transmiterea acesteia spre canal.
  - (6). Transferarea spre canal a informatiilor de stare cind UDM a terminat operatia de pozitionare
- E (1). Transferul bitilor de control si a informatiilor de adresa spre UDM si primirea informatiilor de stare ale UDM
- F (1) Controlul directiei si distanta ce trebuie parcursa de ansamblul de capete.
  - (2). Generarea unui semnal de "sfirsit de pozitionare".
  - (3). Mentinerea capului selectionat pe pista aleasa.
- G (1). Executarea miscarii de pozitionare.
- H (1). Citirea semnalelor "scrise" ("scrise initial" sau preinregistrate).

#### 4.2.2. *Operatia de scriere*

O comanda de scriere pune cuplorul in pozitia de a primi datele ce urmeaza a fi scrise (in paralel opt c.b. de date

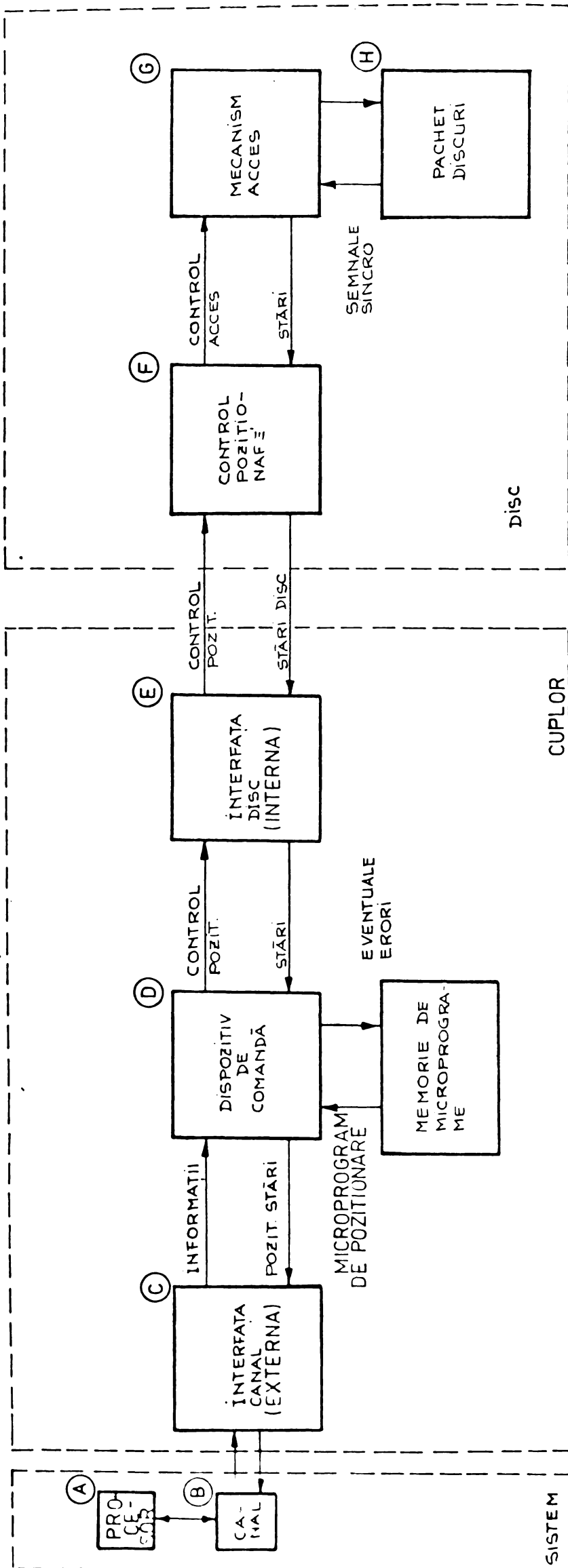


Fig 4.1. Operația de poziționare

plus o c.b. de paritate), de a genera c.b. ale codului de detectie si corectie a erorii si de a le trimite serial catre UDM selectionata. Operatia de scriere este sintetizata in fig.4.2.

- A (1). Instructiunea de intrare/iesire
- B (1). Executarea cuvintului de comanda, transferindu-se cuplorului comanda de scriere.  
(2). Transferul adresei unde urmeaza sa fie scrisa informatia.  
(3). Transferul sub forma paralela a informatiei de scris.
- C (1). Controlul semnalelor de interfata.  
(2). Controlul de paritate a semnalelor transferate.
- D (1). Decodificarea comenzii de scriere.  
(2). Selectarea adresei UDM.  
(3). Testarea bitilor de stare a UDM.  
(4). Trimiterea comenzii de scriere.  
(5). Selectarea microprogramului care permite conversia paralel/seriala.  
(6). Detectia si corectia erorilor (daca este posibil).  
(7). Controlul formatului in scris pe UDM.  
(8). Transferul starilor la sfirsitul operatiei.
- E (1). Primirea informatiei paralele si convertirea ei in serie.  
(2). Adaugarea bitului de corectie a erorilor.  
(3). Utilizarea PLO (Phase Lock Oscilator) pentru controlul timpului de scriere (daca se utilizeaza o schema cu PLO).
- F (1). Transferul informatiilor de control spre UDM selectata si a bitilor de stare ai UDM spre cuplor.
- G (1). Pregatirea UDM pentru scriere.  
(2). Convertirea impulsurilor de scriere in curent de scriere.  
(3). Trimiterea starilor spre cuplor.
- H (1). Capul servo (daca exista) prelucreaza impulsurile PLO, in vederea stabilizarii capului de scriere/citire pe pista dorita.
- I (1). Stocarea (scrierea) bitilor de informatie.

#### 4.2.3. Operatia de citire.

Datele citite de la UDM sint transferate cuplorului care efectueaza operatia de serializare-deserializare a

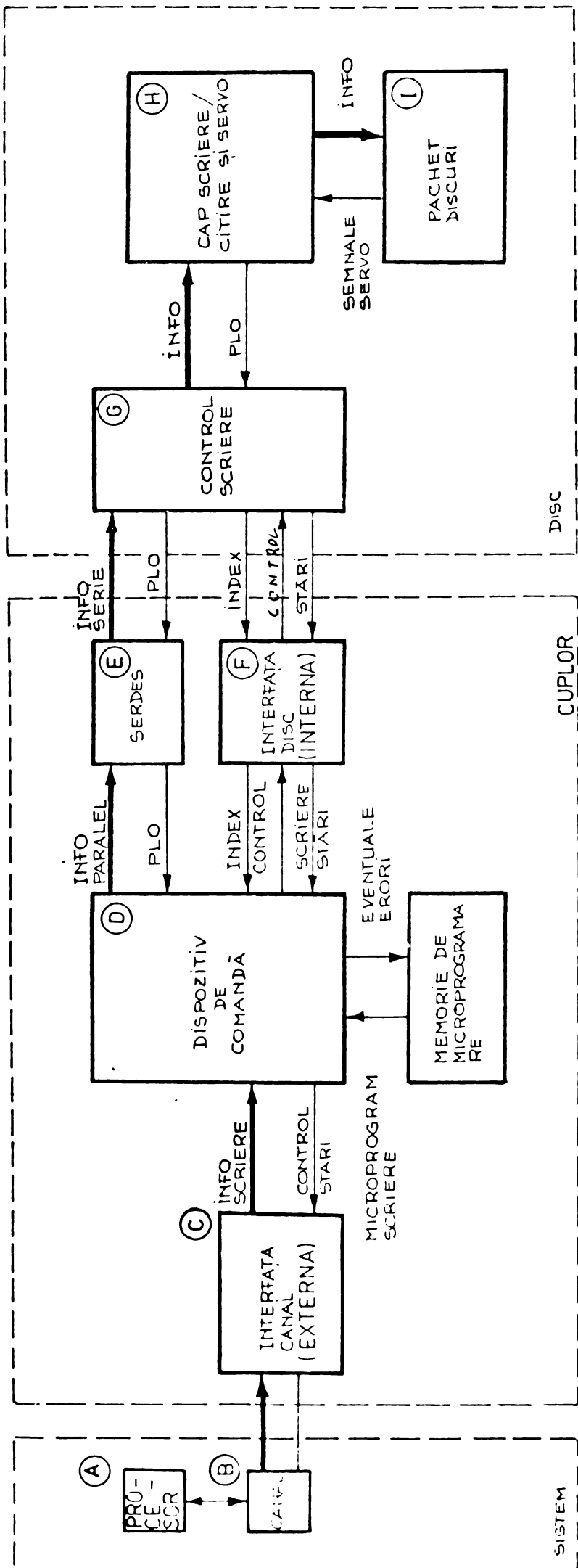


Fig. 4.2. Operația de scriere

informatiei paralele adaugandu-i-se c.b. de paritate. In timpul citirii se calculeaza caracterul de detectie si corectie a erorilor. La terminarea operatiei de citire c.b. de stare a UDM si cuplorului sint transmisi prin interfata externa spre canal. In figura 4.3. este reprezentata operatia de citire.

- A (1). Instructiunea de intrare/iesire.
- B (1). Executarea cuvintului de comanda, transferindu-se cuplorului comanda de citire.  
(2). Transferul adresei de unde urmeaza sa fie citita informatia.  
(3). Transferul informatiei citite.
- C (1). Controlul semnalelor pe interfata interna.
- D (1). Decodificarea comenzii de citire.  
(2). Selectarea UDM adresata.  
(3). Selectarea starilor UDM si transferul lor spre canal.  
(4). Selectarea comenzii de citire si trimiterea ei catre UDM.  
(5). Selectarea microprogramului de serializare-deserializare.  
(6). Detectarea erorilor si corectia acestora daca este posibila.  
(7). Recunoasterea formatului inregistrarii.  
(8). Transferul bitilor de stare la sfirsitul operatiei de citire.
- E (1). Recunoasterea intervalului de sincronizare si realizarea sincronizarii datelor.  
(2). Convertirea informatiei din serie in paralel.  
(3). Controlul informatiilor de detectie si corectie a erorilor.  
(4). Sincronizarea servo (cu PLO daca exista) in intervalul in care nu exista informatie scrisa.
- F (1). Transferul informatiilor de control a citirii catre UDM selectata si transmiterea starilor acesteia catre cuplor.
- G (1). Pregatirea UDM in vederea citirii informatiei (datelor).  
(2). Convertirea curentului de citire in impulsuri de citire.  
(3). Frelevarea starilor UDM si trimiterea acestor-

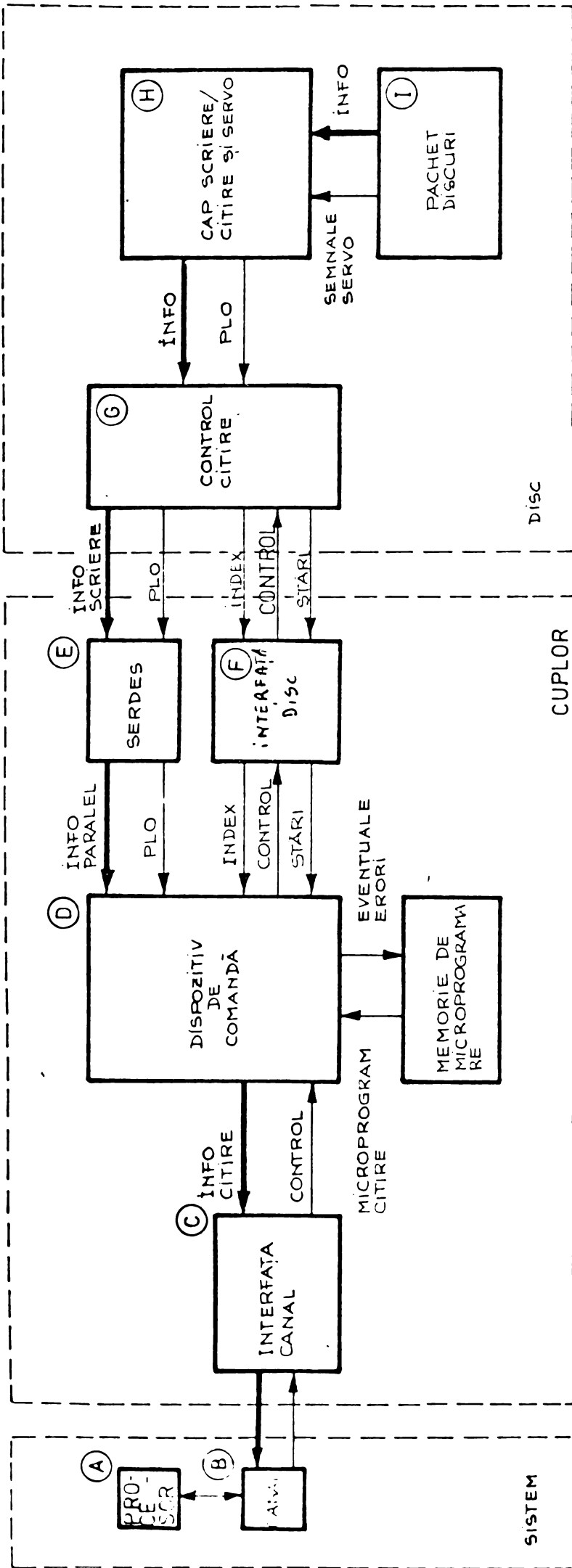


Fig. 4.3. Operație de citire

ra spre cuplor.

H (1). Capul servo (daca exista) prelucreaza impulsurile FLO cu care se stabilizeaza capul de scriere/citire pe pista aleasa.

I (1). Citirea bitilor de informatie.

#### 4.3. *Formatul inregistrarii*

Un element important care trebuie analizat in vederea operatiei de realizare a unui SDM cit mai complet, este formatul inregistrarii utilizat la stocarea datelor pe UDM. Deoarece formatul inregistrarii nu este acelasi pentru toate tipurile de UDM, se va face o analiza in vederea cautarii unor repere comune care sa permita SDM sa genereze si sa recunoasca oricare tip de inregistrare, cu modificari de adaptare minime.

O inregistrare este formata din doua zone principale si anume zona adresei si zona informatiei. Pe langa informatia propriu-zisa, o inregistrare trebuie sa aiba in componenta sa anumite zone de sincronizare.

Formatul inregistrarii este specific unor grupe de cuploare si este inregistrat pe UDM in momentul premarcarii. Punctul de referinta pentru inregistrarea de premarcare este indexul. Pe marginea discului inferior al pachetului de discuri exista o decupare care este interpretata ca origine a inregistrarii, numindu-se index. Exista pachete care au 21 de decupari, 20 dintre acestea impartind circumferinta pachetului in 20 de parti egale folosite de unele cuploare pentru sectorizarea informatiei, iar a 21-a decupare avind rolul creerii originii, adica a indexului.

Pachetul de discuri este impartit in cilindrii, piste si sectoare. Un cilindru reprezinta totalitatea spatiului de inregistrare disponibil la o anumita pozitionare a ansamblului de capete de scriere/citire. Astfel, pachetul de discuri avind o latime radiala utila de 2 inch, rezulta ca pentru densitatea de inregistrare a pistelor de 100 tpi, 200 tpi si 400 tpi corespunde un numar total de cilindri: 200, 400 si respectiv 800. Intersectia unui cilindru cu fiecare din cele 20 de suprafete de inregistrare, formeaza pista. In acest fel fiecare cilindru este format din 20 de piste, fiecare apartinand unui anumit cap de scriere/citire. La rindul lor pistele pot fi divizate in sectoare, al caror numar si lungime variaza de la un sistem de calcul la altul. Imediat dupa index trebuie sa existe un spatiu numit "pre-



ambul" care este necesar pentru a da posibilitatea schemei de comutare a capetelor de scriere/citire, sa execute o eventuala comutare. Pentru ca sa nu apara situatia nedorita, ca doua capete de scriere/citire sa fie selectate simultan, schema de comutare este astfel conceputa incit intre deselectarea unui cap si selectarea altui cap se lasa un interval de timp. Acest interval este necesar si pentru stabilizarea semnalelor la noul cap selectat. Preambulul care exista imediat dupa index este necesar pentru a acoperi acest timp, in cazul in care aparitia unei noi piste, coincide cu o schimbare a capului de scriere/citire. In continuare exista o zona de sincronizare in care se face trecerea de pe "ceasul intern" pe "ceasul extern". (Prima faza a unei inregistrari, este cautarea adresei la care urmeaza sa se faca inregistrarea. Aceasta adresa este inregistrata in momentul premarcarii pachetului de discuri. Operatia de premarcare se executa sub controlul unui ceas al cuplorului numit "ceasul intern", pe pachetul de discuri inregistrindu-se aceste impulsuri. Citirea impulsurilor de ceas inregistrate pe pachetul de discuri, permite crearea unor semnale de sincronizare, care formeaza "ceasul extern" ce va governa toate operatiile de citire). Tot in aceasta zona trebuie sa existe informatii care sa permita sesizarea momentului in care incepe zona de adresa, care cuprinde informatii ce indica cilindru, capul si sectorul. De asemenea se dau indicatii daca pista este buna sau defecta, in ultimul caz trebuind inscrisa si adresa unei piste de rezerva. In continuare in functie de faptul daca se doreste un format fix sau variabil al informatiei se poate folosi o zona de contabilizare a pistei respective, care indica felul formatului si situatia inregistrarilor pe pista. Aceasta zona se completeaza in timpul lucrului aratand disponibilitatea pistei. O grupa de caractere necesare corectiei si detectiei erorilor, precum si anumite caractere de identificare, incheie zona de adresa. Acest grup de caractere se numeste "postambul". In plus mai poate fi folosita o cheie de protectie si recunoastere.

Zona de informatie incepe si ea cu un preambul. Acest preambul trebuie sa permita o eventuala comutare a infasurarii de citire cu infasurarea de scriere a capului precum si stabilirea exacta a inceputului cimpului de date. Cimpul de date poate avea o lungime fixa sau variabila. In cazul lungimii fixe, daca pachetul de date nu este divi-

zibil cu lungimea sectorului, restul va fi inregistrat in sectoarele urmatoare, portiunea ramasa libera din ultimul sector completindu-se de catre cuplor cu "zerouri". Zona de informatie se incheie cu un postambul care cuprinde informatii pentru detectia erorilor, precum si un spatiu intersector pentru compensarea anumitor variatii de viteza ale pachetului de discuri in limitele tolerantelor admise. Un SDM universal trebuie sa poata realiza prin premarcare orice fel de format de inregistrare si apoi in lucrul curent cu UDM sa poata interpreta acest format. Pentru perfecta compatibilitate e necesar ca in formatul inregistrarii sa nu apara nici un fel de informatie suplimentara. Stabilirea variantei de premarcare se poate face numai prin adaugarea unor semnale de informatie in zona preambulului.

#### *4.4. Ordinele utilizate pentru comanda SDM*

In general interfata externa a SDM prezinta diferente destul de mari de la un sistem de calcul la altul. Elementul comun care poate constitui un factor de unificare este ansamblul de instructiuni de intrare-iesire si implicit de ordine prin care sistemul de calcul comanda activitatea SDM.

Transferul de informatii se realizeaza intotdeauna dupa urmatorul principiu:

- initializarea, adica adresarea SDM printr-o instructiune de intrare/iesire. (SIO - Start Input Output);
- daca nu exista semnalate conditii de eroare, se trimite un ordin care defineste tipul transferului ce trebuie executat;
- dupa recunoasterea ordinului se executa transferul de date;
- la sfirsitul transferului cuplorul se poate deconecta in asteptarea altei instructiuni, sau daca exista o intrantuire de comenzi se asteapta un alt transfer de informatii ca urmare a unui nou ordin.

Ordinele utilizate sint urmatoarele:

- cautare: ordin prin care se cere cuplorului transmiterea catre UDM a unei comenzi de pozitionare si imediat a unui semnal pe interfata externa prin care se semnalizeaza executarea ordinului;
- cautare cu intrerupere: la fel ca la cautare, doar ca la terminarea pozitionarii cuplorul va emite un semnal de intrerupere;

- scrierea de informatii: ordin care indica scrierea informatiei pe UDM selectionata cu incepere de la sectorul ales. Scrierea nu este posibila daca adresa este cumva potejata sau daca ordinul de cautare a indicat o pista de rezerva;

- citire cu oprire pe eroare de paritate: ordinul permite citirea informatiei, erorile de paritate fiind luate in considerare la sfirsitul sectorului in care au aparut;

- citire: ordin similar cu precedentul, doar ca o eroare de paritate va fi semnalizata la terminarea in intregime a transferului de date;

- controlul scrierii: ordin prin care cuplorul compara datele primite de la UDM cu date etalon (de obicei trimise de canal), semnalizarea erorii de comparare facindu-se la sfirsitul sectorului in care s-a produs;

- scrierea adresei de rezerva: ordin utilizat pentru indicarea unei piste defecte sau nedisponibile si pentru inregistrarea adresei sale secundare (de rezerva);

- citirea adresei: ordin care permite cunoasterea starii unei piste, a adresei sale (primare) si a adresei secundare daca este cazul;

- cerere de stare: ordin prin care cuplorul este solicitat sa furnizeze informatii asupra adresei ultimului sector la/de la care s-a efectuat un transfer si sa dea indicatii asupra starii unitatii selectate;

- protectie scriere: ordin care cere inhibarea oricarei operatii de scriere pe UDM selectionata;

- protectie zona: ordin prin care cuplorul interzice orice acces (cautare, scriere, citire) in afara cilindrului adresat in momentul respectiv, adica se interzice orice miscare a ansamblului de capete, cu exceptia aparitiei unui sector defect cind se va permite accesul pe pista de rezerva;

- punere la zero a accesului: ordin care adreseaza final UDM comandind intoarcerea pe cilindrul zero (origine);

- rezervare (numai la dublul acces): ordin prin care UDM ramine la dispozitia canalului care a emis ordinul, pina cind acest canal lanseaza un alt ordin care sa permita "liberarea" acestuia sau "punerea la zero" a accesului respectiv;

- liberarea (numai la dublul acces): ordin complementar cu precedentul si care poate fi lansat numai de canalul care a facut rezervarea.

#### 4.5. Concluzii .

In cadrul SDM interfata externa este unul din parametrii asupra caruia se poate actiona in vederea realizarii unui subansamblu cit mai independent, cu functii bine definite si cu un dialog extern optimizat in sensul obtinerii unui transfer de date maxim cu un schimb de informatii auxiliare minim.

Pe langa functiile clasice (legatura cu sistemul de calcul, legatura cu UDM, transferul de comenzi si date, eventual corectia erorilor) este de dorit ca SDM sa poata executa functii noi care pina acum erau indeplinite de software-ul sistemului. Pentru aceasta anumite functii pot fi realizate de hardware-ul SDM, iar altele prin conferirea SDM a unei puteri de calcul si decizie. Astfel SDM va putea realiza: optimizarea stocarii informatiei pe UDM, lucrul cu mai multe sisteme de calcul, ridicarea ratei de transfer a informatiilor, simplificarea dialogului extern cu sistemul de calcul. In anumite situatii este avantajos ca SDM sa fie transformat intr-un sistem de intrare-iesire care sa gestioneze si activitatea altor echipamente periferice. In capitolele urmatoare aceste probleme vor fi analizate detaliat.

## CAP.5. INTERFATA SISTEMULUI "INDEPENDENT"

In scopul realizarii unui SDM care sa poata fi utilizat simultan de doua sau mai multe sisteme de calcul, este necesar sa fie conceput un sistem de interfata special destinat acestui scop. Elaborarea sistemului de calcul INDEPENDENT in anul 1976 a urmarit realizarea unui sistem eficient, compatibil cu sistemele firmei DEC-SUA. Compatibilitatea s-a realizat prin elaborarea unor interfete special concepute in acest scop. Pe baza interfetelor sistemului INDEPENDENT se pot face dezvoltari care sa permita realizarea de structuri multiprocesor si implicit care sa permita utilizarea SDM simultan de mai multe procesoare. Analiza ce urmeaza cuprinde si aceste dezvoltari.

### 5.1. Organizarea sistemului

Sistemul I-100 este organizat in jurul a

doua magistrale de comunicatie: INTERBUS-ul si MEMOBUS-ul (fig.5.1. pentru simplificare aceste busuri vor fi notate cu Bus I si Bus M).

Busul I permite transferul informatiilor intre unitatea centrala (UC) si toate cuploarele echipamentelor periferice ale sistemului. Pe acest bus se trimit comenzi, stari, adrese si date. Busul I a fost conceput in ideea compatibilizarii cu busul UNIBUS al firmei DEC, bus adoptat si in tarile socialiste la constructia minicalculatoarelor. Prin intermediul unor adaptoare aceasta compatibilitate este totala. Pe busul I se poate conecta si memoria operativa, insa in scopul maririi randamentului sistemului, memoria a fost cuplata la un bus special Busul M (fig.5.1.).

Busul M este o varianta simplificata a busului I, care permite cuplarea memoriei cu UC si cu cuploarele care lucreaza cu acces direct la memorie (ADM), in cazul sistemului I-100; discurile magnetice, benzile magnetice, cuploare pentru teletransmisie si cuploare intercalculatoare.

Existenta celor doua busuri permite utilizarea lor simultana. De exemplu: UC poate lucra cu un periferic simplu in timp ce un cuplor cu ADM v-a putea realiza un transfer de date cu memorie. Aceasta caracteristica ridica mult performantele sistemului I-100.

Ambele busuri avind aceleasi caracteristici electrice, de timp si constructive, in cele ce urmeaza se va analiza numai Busul I, considerind si memoria conectata pe acest bus.

UC utilizeaza acelasi set de semnale pentru a comunica cu dispozitivele periferice (se intelege cuplorul si perifericul) si cu memoria. Dispozitivele periferice utilizeaza de asemenea acelasi set de semnale cind comunica cu UC, memoria sau cu celelalte dispozitive periferice. Instructiunile referitoare la manipularea datelor in memorie se aplica si pentru manipularea datelor in registrele dispozitivelor periferice.

In acest scop registrele dispozitivelor periferice au adrese bine definite, putind fi adresate pe bus in acelasi fel ca si locatiile memoriei.

Comunicarea pe bus intre doua dispozitive se

face dupa principiul "stapin-sclav". Dispozitivul care la un moment dat are controlul busului se numeste "stapin", iar dispozitivul care este comandat de acesta se numeste "sclav". Aceasta relatie stapin-sclav este dinamica, fiecare dispozitiv putind ocupa una din cele doua situatii. Pe bus comunicarea este interconditionata, semnalele de comanda trimise de stapin trebuind sa fie confirmate de semnale trimise de sclav.

La dispozitivele periferice, functiile de comanda sint incarcate in registrele de control, urmind ca anumite c.b. in aceste registre sa realizeze comanda operatiilor. Exista de asemenea registre de stare, care pot fi testate prin adresarea lor directa. Numarul de registre al dispozitivelor periferice este dependent de tipul echipamentului periferic.

Un dispozitiv periferic cind devine stapin, cere utilizarea busului pentru a rezolva una din urmatoarele doua situatii:

- a. realizarea unui transfer de date direct la memorie, fara intermediul UC;
- b. intreruperea executiei unui program si fortarea UC la adresa de inceput a unei subrutine de tratare a intreruperii respective.

Busul este astfel conceput incit exista linii separate pentru transportul datelor si pentru cererea si acordarea busului. In acest fel cele doua operatii pot fi indeplinite in paralel, transferul de date putindu-se face la viteza maxima de lucru a busului.

### *5.2. Linii de semnale ale Busului I*

Adresele, comenzile, datele si sincronizarile sint transmise fiecare pe un set de linii distincte. Activitatea busului este asincrona si dependenta de semnale de control interconditionate.

Semnalele busului se pot imparti in urmatoarele categorii:

- a. semnale folosite la transferul datelor;
- b. semnale folosite la atribuirea prioritatii;
- c. semnale de initializare.

a. TABELUL 5.1.

Denumire	Mnemonica	Numar linii	Functia
Adrese	A00 - A15	16	Selecteaza adrese din memorie sau registre.
Date	D00 - D15	16	Informatie transmisa.
Comenzi	C00 - C01	2	Tipul de transfer de date.
Sincronizare stapin	MSINC	1	Sincronizarea transferului
Sincronizare sclav	SSINC	1	de date.
Paritate	PARA, PARB	2	Eroare de paritate.
Intrerupere	INTR	1	Sincronizare intreruperi.

b. TABELUL 5.2.

Denumire	Mnemonica	Numar linii	Functia
Cerere bus	CB4 - CB7	4	Cereri de intrerupere emise de dispozitive.
Acordare bus	AB4 - AB7	4	Acordarea busului.
Cerere bus pentru transfer de date	CTD	1	Cereri facute de UC sau dispozitive pentru transferul de date.
Acordare bus pentru transfer de date	ATI	1	Acordarea busului pentru transferul de date.
Confirmare	CAB	1	Confirmarea acordarii busului.

Bus ocupat	OCUPAT	1	Busul este ocupat cu un transfer de date sau tratarea unei intreruperi.
------------	--------	---	---

c. TABELUL 5.3.

Denumire	Mnemonica	Numar linii	Functia
Initializare	INIT	1	Initializeaza dispozitivele conectate pe bus.
Tensiune joasa	CTA,CTC	2	Semnalizeaza caderea tensiunii alternative sau continue.

### 5.3. Transferul de date

#### 5.3.1. Generalitati

Transferul de date se poate realiza intre doua dispozitive sau intre un dispozitiv si memorie, fara intermediul UC. Aceste transferuri se numesc transferuri CTD (cererea busului pentru Transferul de Date). Un dispozitiv care are controlul busului transfera cuvinte (16 c.b.) spre memorie, viteza transferului fiind egala cu ciclul de memorie.

Dupa cum rezulta din Tabelul 5.1. pentru transferul de date sint folosite 39 de linii, dispozitivul stapan pe bus, transferind date spre sau de la dispozitivul sclav.

##### 5.3.1.1. Linii de date D00 - D15

Cele 16 linii de date contin informatii care sint transferate intre stapan si sclav. Cuvintul este format din 2 octeti, octetul cel mai putin semnificativ fiind desemnat de c.b. 0 - 7 (fig.5.2.).

##### 5.3.1.2. Linii de adresa A00 - A15.

Cele 16 linii de adrese servesc la transmiterea unei adrese de la dispozitivul stapan spre sclav. Formatul adresei este dat in fig.5.3. Pentru cuvinte adresele sint pare. In operatiile cu octeti c. b. 0 specifica octetul adresat.



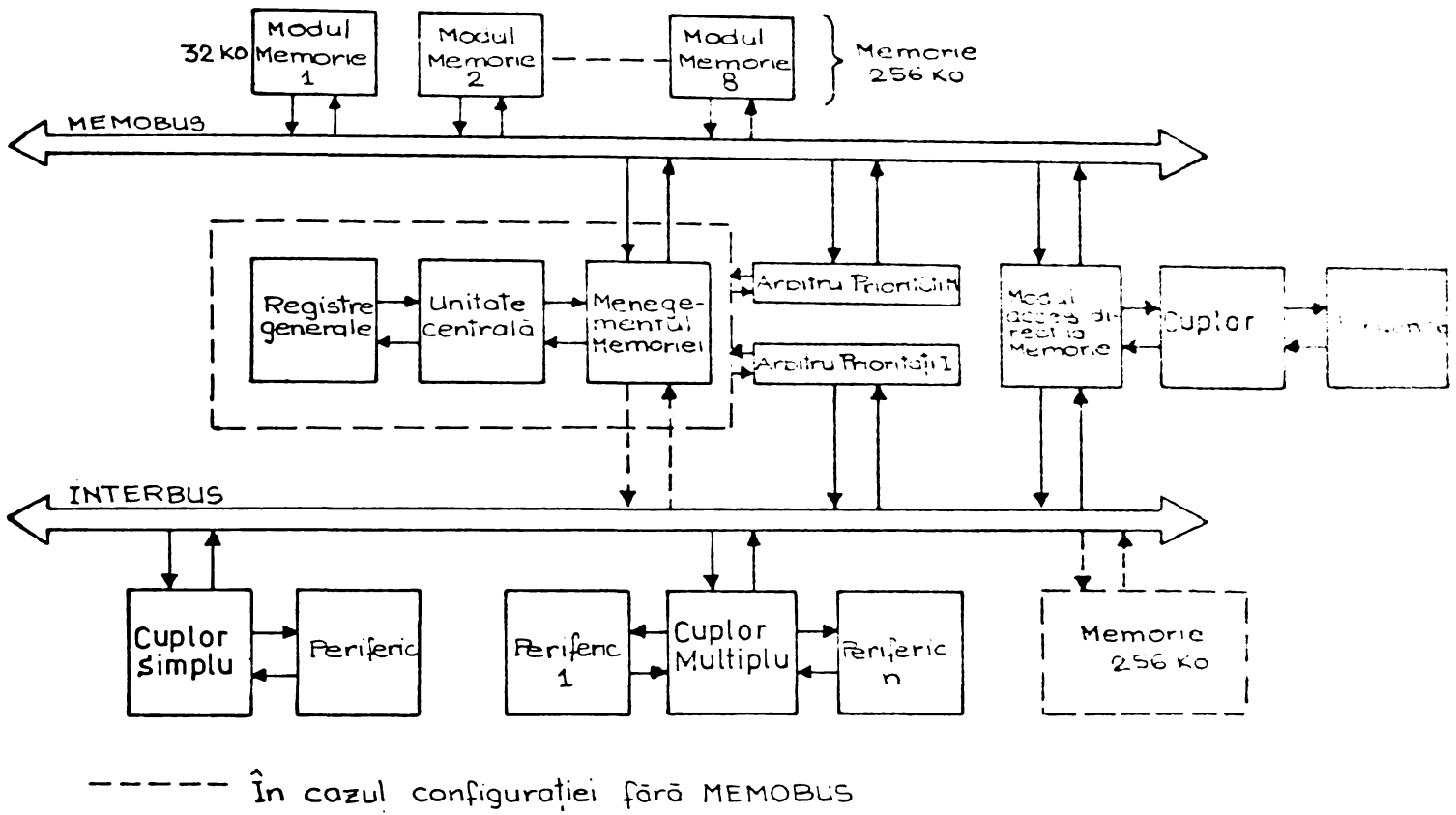


Fig. 5.1. SCHEMĂ ORGANIZARE SISTEM

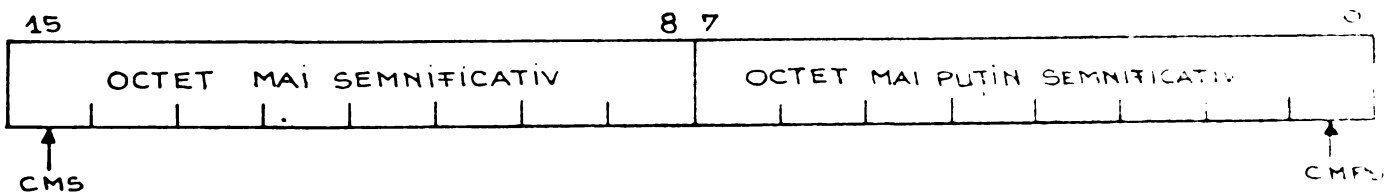
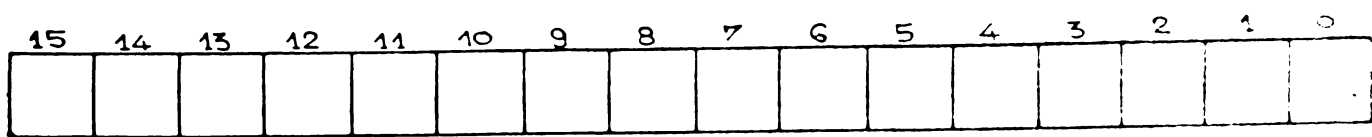


Fig. 5.2. FORMATUL CUVÎNTULUI



cbo = 0 - selectează octetul cel mai puțin semnificativ  
 cbo = 1 - selectează octetul cel mai semnificativ

Fig. 5.3. FORMATUL ADRESEI

5.3.1.3. *Liniiile de comenzi C00 si C01*

Semnalele trimise pe aceste linii definesc cele 4 tipuri de operatii de transfer posibile, (tabelul 5.4.), Semnalele sint transmise de dispozitivul stapin spre dispozitivul sclav. Sensul transferului este definit cu referire la dispozitivul stapin.

TABELUL 5.4.

Nr. crt.	Nume	Mnemonică	Nivel logic		Funcția
			C01	C02	
1.	INTRARE	INTRARE	0	0	Transfera cuvint de la slav la stapin.
2.	INTRARE, PAUZA	INTRARE P	0	1	Transfera cuvint de la sclav la stapin, dar inhiba ciclul de rememorare la dispozitivele cu citire destructiva. Poate fi urmata de o IESIRE.
3.	IESIRE	IESIRE	1	0	Transfera cuvint de la stapin la sclav.
4.	IESIRE OCTET	IESIRE 0	1	1	Transfera octet de la stapin la sclav.

5.3.1.4. *Indicatorii erorii de paritate PARA si PARB.*

Aceste semnale sint generate de sclav si receptionate de stapin. Existenta unei erori de paritate este semnalizata prin PARA = 0 si PARB = 1, iar un transfer fara eroare de paritate de PARA = PARB = 0.

5.3.1.5. *Semnalele de sincronizare MSINC si SSINC.*

Semnalul de sincronizare al dispozitivului stapin MSINC este emis de acesta si receptionat de sclav, avind doua functii:

- indica dispozitivului sclav definit de semnalele de pe liniile de adrese A, ca acesta poate indeplini comenzile trimise pe liniile de comenzi, C.

- negarea semnalului MSINC indica sclavului ca transferul de date a luat sfirsit.

Semnalul de sincronizare al dispozitivului sclav SSINC este emis de acesta si receptionat de stapin, indicandu-i dispozitivului stapin ca a terminat partea sa din trans-

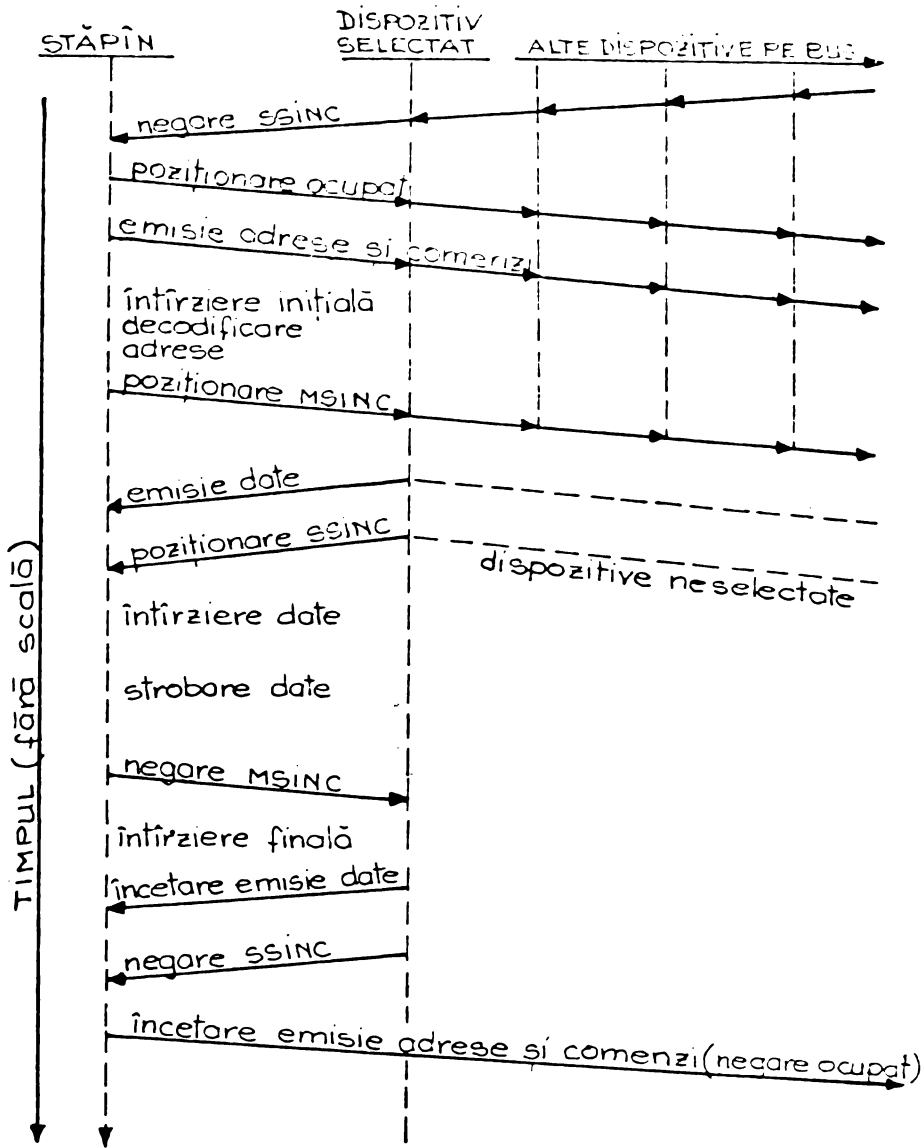


Fig. 5.4. CICLU TIPIC INTRARE/INTRARE PAUZĂ

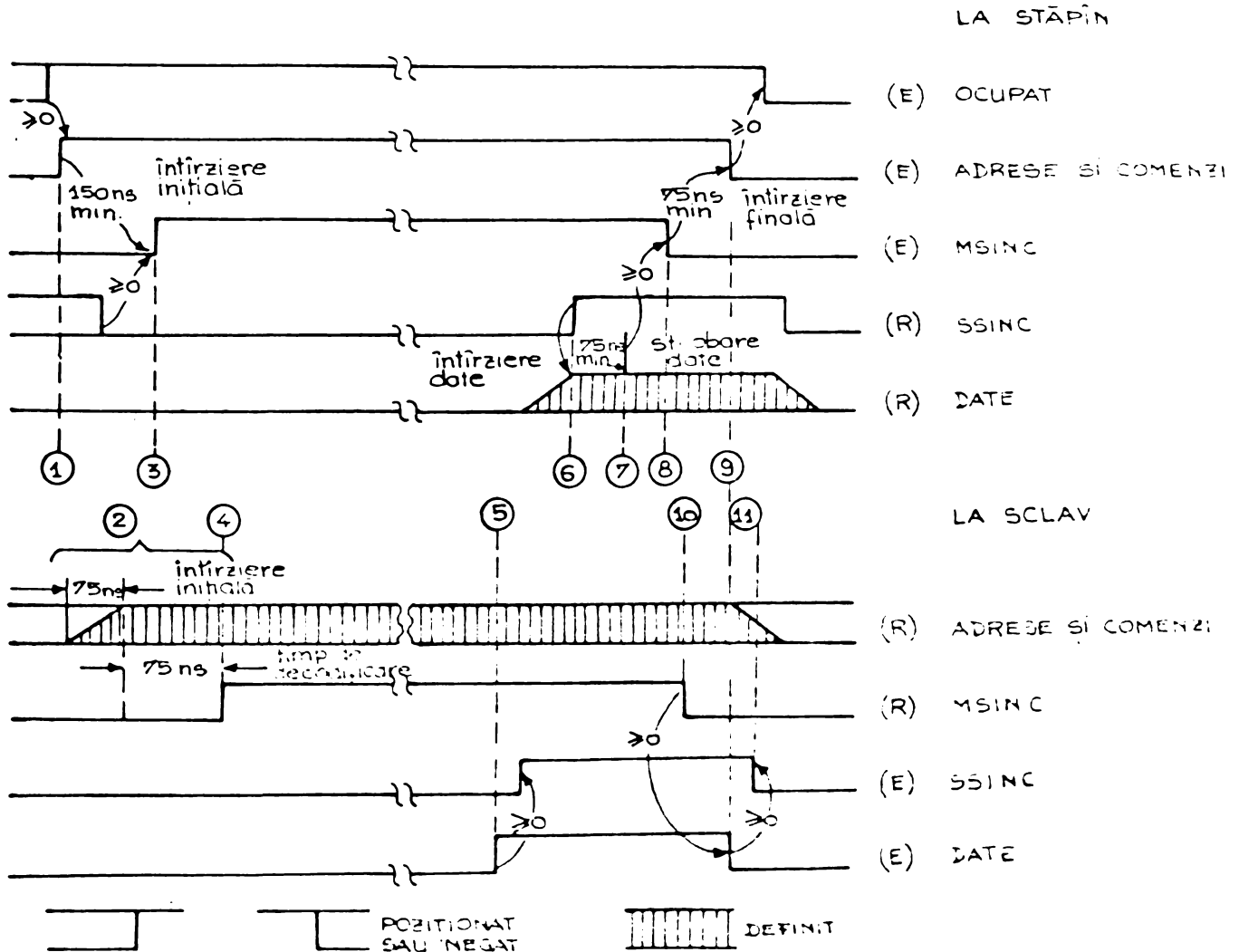


Fig. 5.5. TRANSFER TIPIC INTRARE

ferul de date. Negarea semnalului SSINC indica tuturor dispozitivelor de pe bus, ca sclavul a terminat transferul de date.

#### 5.3.1.6. *Cerere de intrerupere INTR.*

Semnalul INTR este emis de dispozitivul care a cerut o intrerupere, dupa ce acesta a devenit stapin pe bus si informeaza UC ca pe liniile de date se afla un vector (adresa) de intrerupere si ca urmeaza intreruperea propriu-zisa. La sfirsitul transferului semnalul INTR este negat. Semnalul INTR poate fi generat numai de un dispozitiv care a obtinut controlul busului folosind una din liniile AB 4,5,6 sau 7.

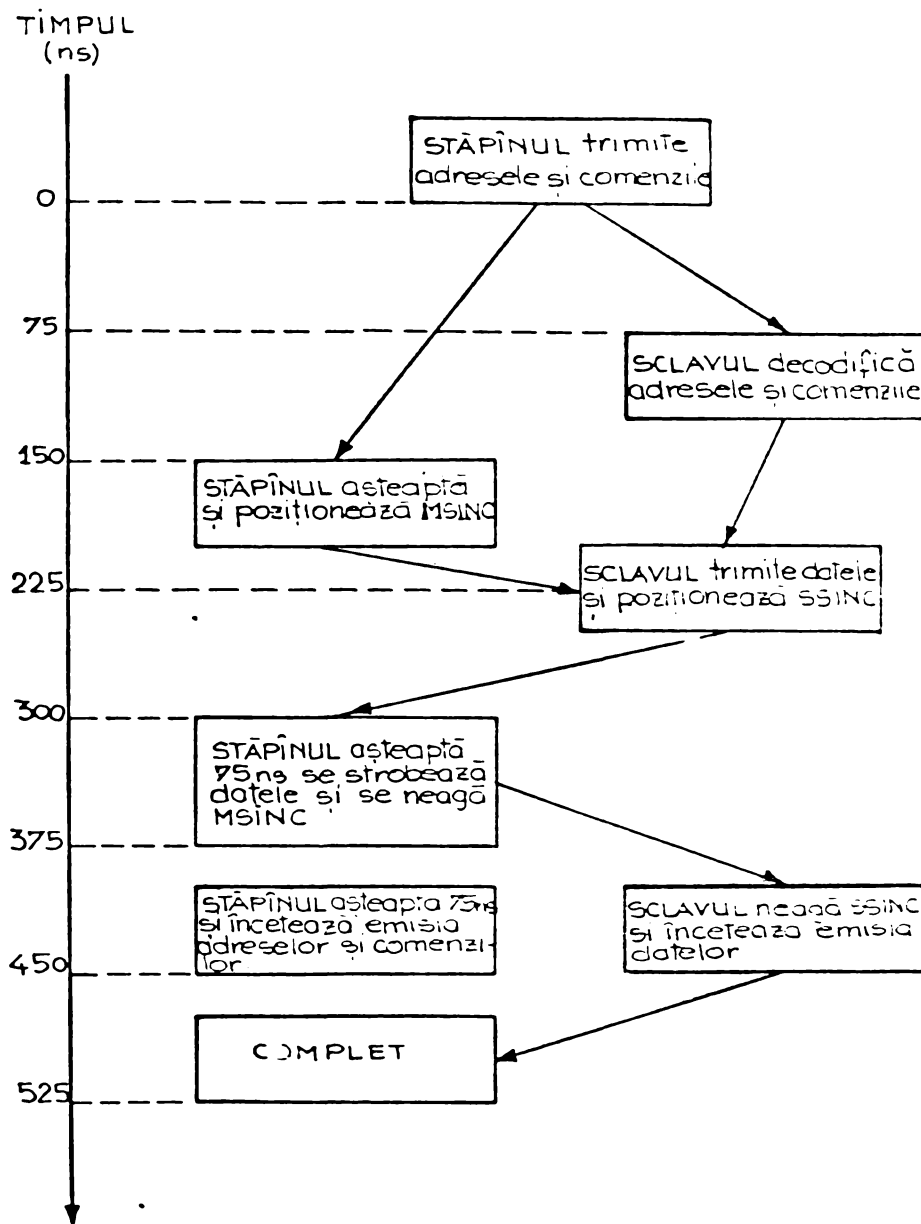
#### 5.3.1.7. *Diferenta de propagare (skew).*

Doua semnale emise simultan pe doua linii separate datorita diferentei de propagare pot ajunge decalate la destinatie. Pentru a anihila aceasta decalare posibila, la proiectarea Busurilor I si M s-a prevazut un timp maxim de stabilizare de 75 nsec.

#### 5.3.2. *Transferul de tip INTRARE/INTRARE PAUZA.*

Operatiile INTRARE si INTRARE PAUZA (INTRARE P) sint transferuri de la sclav la stapin, folosindu-se pentru transferul datelor liniile D, fiind intotdeauna transferuri pe cuvint (daca stapinul doreste numai un octet, va prelua octetul dorit). In cadrul transferului P, sclavul este informat ca ciclul de reinscriere trebuie inhibat. In acest fel intotdeauna un transfer INTRARE P, v-a trebui sa fie urmat de un transfer IESIRE sau IESIRE OCTET (IESIRE O) in cursul caruia la locatia respectiva a sclavului se reinscrie noua informatie. In cazul dispozitivelor de memorie nedistructiva, transferurile INTRARE si INTRARE P sint tratate in mod identic de catre sclav.

In figura 5.4. este analizata interactiunea dintre sclav si stapin. Un dispozitiv care devine stapin pe bus (o pozitionat semnalul OCUPAT) plaseaza adresele si comenzile pe liniile A si C. Aceste adrese si comenzi sint decodificate de catre celelalte dispozitive conectate pe bus. Dupa plasarea adreselor si comenzilor, stapinul asteapta pentru ca semnalele sa se stabilizeze si pentru ca dispozitivele de pe bus sa aibe timp sa le decodifice. In continuare, daca dispozitivul sclav din ciclul anterior de transfer de date a terminat respectivul transfer negind semnalul SSINC, stapinul v-a pozitiona semnalul sau de sincronizare MSINC. Dispozitivul sclav selectat, dupa receptionarea semnalului MSINC, plaseaza datele pe liniile D si pozitioneaza semnalul



NOTĂ : Este prezentat cel mai defavorabil caz de întârziere de propagare de 75 ns. Timpul total pentru un singur transfer este de 525 ns, însă următorul ciclu poate începe la 440 ns, când stăpînul poate trimite noile adrese și comenzi. Viteza maximă de transfer este de 2,5 milioane cuvinte/secundă, adică 5 M oct./secundă .

Fig.5.6. DIAGRAMĂ DE TIMP PENTRU CICLUL INTRARE

sau de sincronizare SSINC. Dispozitivul stăpîn aşteaptă stabilizarea datelor, apoi strobează datele la recepţionarea semnalului SSINC şi nega semnalul MSINC, negare care comunică sclavului că datele au fost acceptate.

Activitatea sclavului în transferul de date se încheie prin încetarea emisieii datelor pe liniile D şi prin negarea semnalului SSINC. Stăpînul pentru a preveni falsă selectare a altui dispozitiv păstrează informaţiile pe liniile A şi C pînă la recepţionarea semnalului MSINC de către toate dispozitivele, apoi termină şi el partea sa din transferul de date prin încetarea emisieii pe liniile A şi C. Dacă stăpînul doreşte un alt transfer (IESIRE sau IESIRE 0 după INTRARE P) va menţine în continuare semnalul OCUPAT, în caz contrar îl va nega.

O etapizare a unui transfer INTRARE/INTRARE P este dată în fig.5.5. punîndu-se în evidenţă şi propagarea semnalelor, în cazul cel mai defavorabil luat în considerare la proiectarea busului.

Semnalul OCUPAT fiind poziţionat, dispozitivul stăpîn plasează adresele şi comenzile pe liniile respective (1). Dispozitivele de pe bus recepţionează adresele şi comenzile pe care le decodifică fiind pusă în evidenţă întârzierea de propagare (2). După o aşteptare de 150 ns (75 ns pentru compensarea întârzierii maxime pe liniile A şi C şi 75 ns pentru decodificarea adreselor şi comenzilor), stăpînul poate poziţiona MSINC cu condiţia ca SSINC să fie negat (adică transferul precedent să fie încheiat) (3). Semnalul MSINC este recepţionat după un timp de propagare de către toate dispozitivele busului, unul dintre acestea care a decodificat adresele recunoscîndu-se ca sclav (4). Sclavul după un anumit timp plasează datele cerute pe liniile D şi apoi poziţionează semnalul SSINC (5). După un timp de propagare, stăpînul recepţionează poziţionarea lui SSINC (6). Dacă după un interval de 10 - 20 us acesta nu este recepţionat, se poziţionează un indicator de eroare. Stăpînul după o aşteptare de 75 ns va strobează datele (7) putînd nega semnalul MSINC (8). După o aşteptare de 75 ns, stăpînul încetează emisia adreselor şi comenzilor, iar dacă acesta este ultimul transfer de date va nega şi semnalul OCUPAT (9). MSINC negat este recepţionat după un timp de propagare de către sclav (10), iar sclavul poate înceta emisia datelor pe liniile D şi nega SSINC (11).

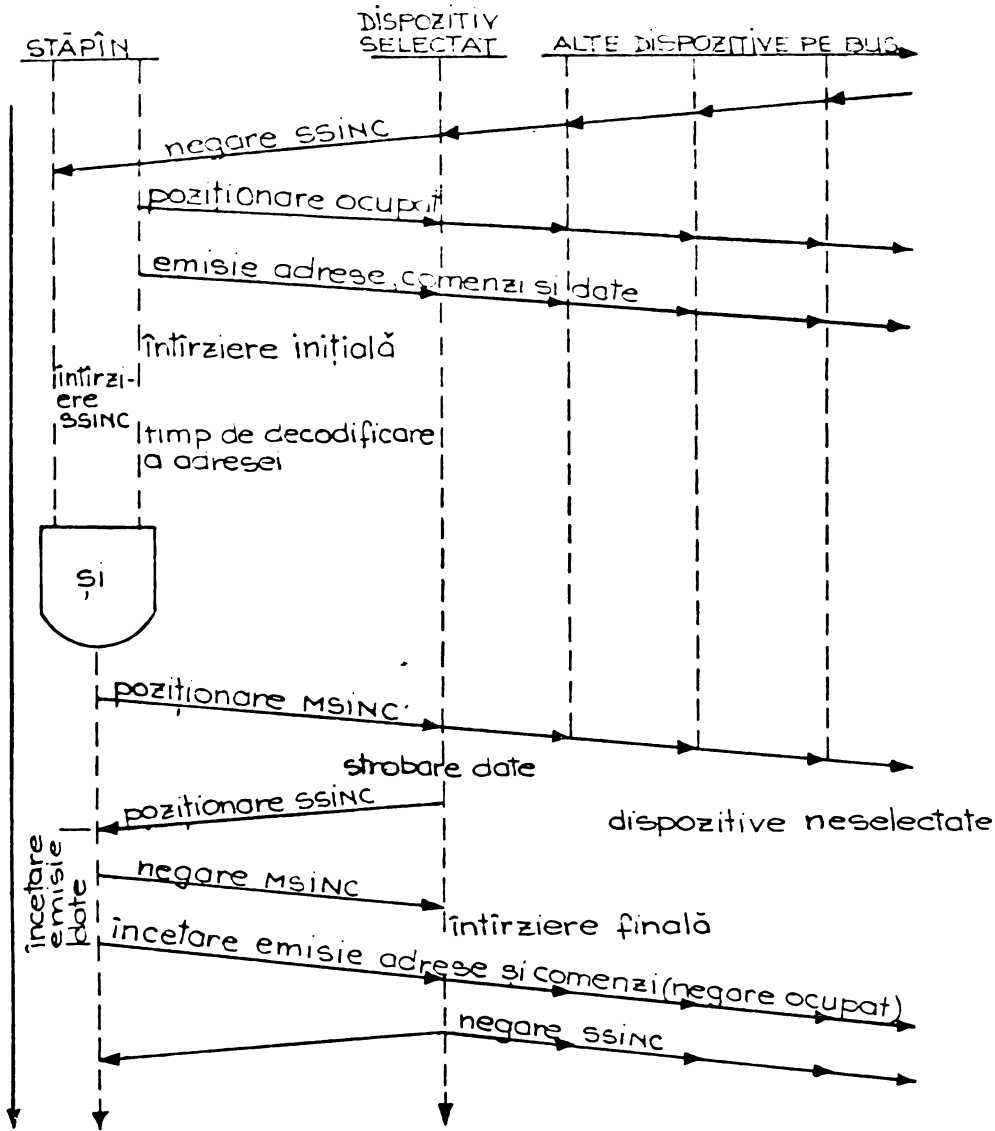


Fig. 5.7. CICLU TIPIC IEȘIRE / IEȘIRE OCTET

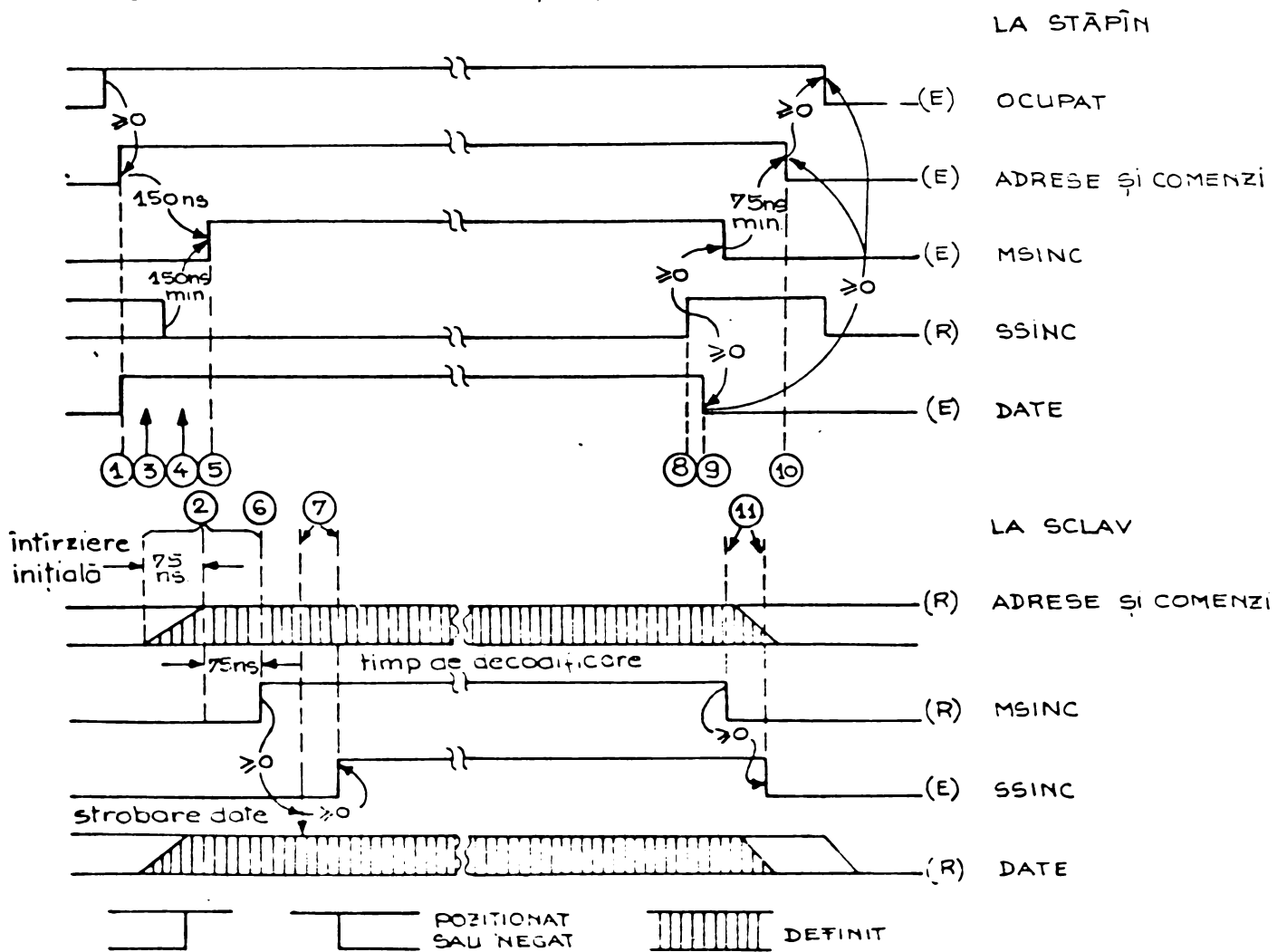


Fig. 5.8. TRANSFER TIPIC IEȘIRE

O diagrama de timp este data in fig.5.6. rezultind ca in conditiile cele mai favorabile de propagare un transfer se poate efectua in 525 ns, dar noile adrese si comenzi pot fi trimise dupa 440 ns.

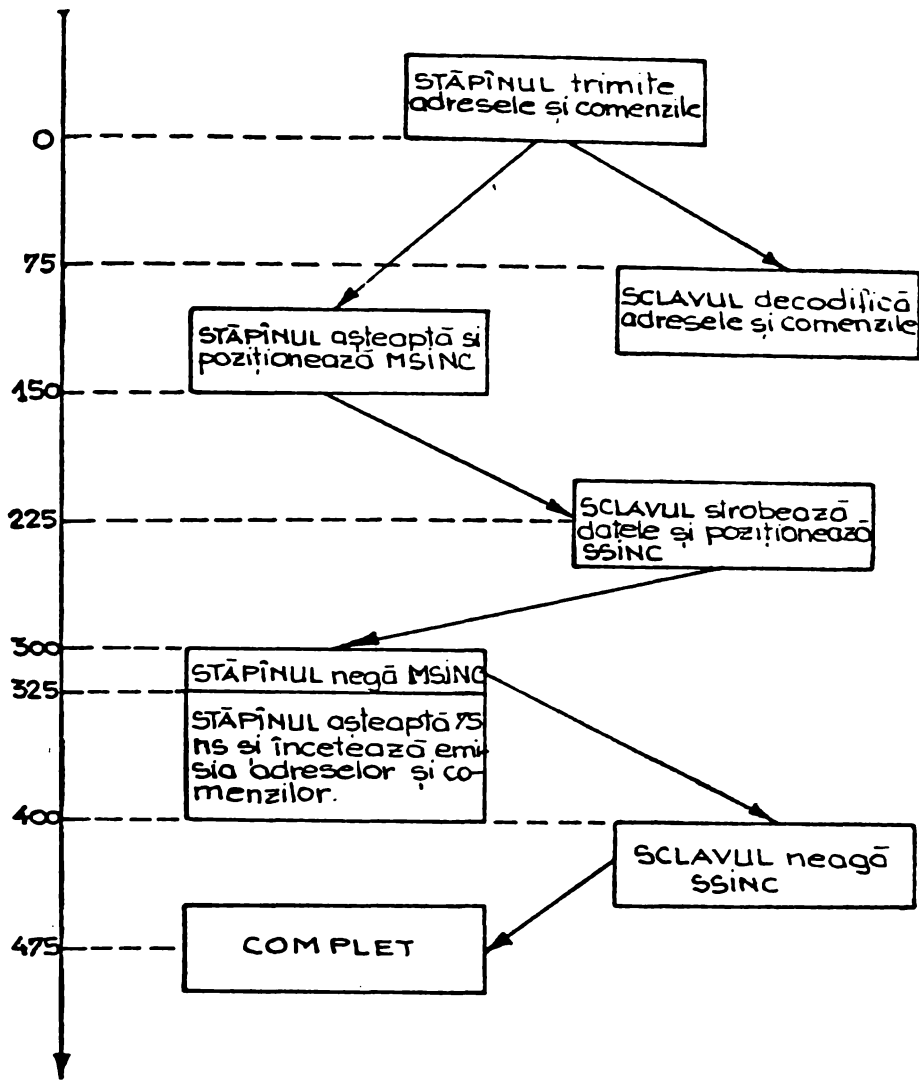
### 5.3.3. *Transferul de tip IESIRE/IESIRE OCTET.*

Operatiile IESIRE si IESIRE OCTET (IESIRE 0) sint transferuri de date pe liniile D de la stapin la sclav. Operatia IESIRE este utilizata cind se doreste transferul unui cuvint la adresa specificata de liniile A01 - A15; datele fiind transmise pe liniile D0 - D15. In cazul operatiei IESIRE 0 se transfera un singur octet la adresa indicata de A00 - A15, octetul cel mai putin semnificativ fiind indicat de A00 = 0, stapinul plasind informatia pe liniile D00 - D07, iar octetul mai semnificativ de A00 = 1, datele fiind plasate pe liniile D08 - D15.

In fig.5.7. este analizata actiunea dintre stapin si sclav in cazul unui ciclu tipic IESIRE/IESIRE 0. Un dispozitiv stapin pe bus (OCUPAT POZITIONAT) plaseaza adresa sclavului, comenzile si datele pe liniile A, C si D. Adresele si comenzile sint decodificate de dispozitive pentru a stabili sclavul. Stapinul poate pozitiona MSINC numai dupa o intirziere corespunzatoare pentru stabilizarea liniilor A si C si o intirziere dupa receptionarea negarii lui SSINC pentru asigurarea incetarii emisiei datelor pe liniile D de catre sclavul anterior. Sclavul selectat, dupa ce receptioneaza pozitionarea lui MSINC strobeaza datele de pe liniile D si pozitioneaza SSINC. Stapinul la receptionarea lui SSINC va nega MSINC si stabilizeaza liniile A si C pentru ca toate dispozitivele sa poata receptiona negarea lui MSINC inainte ca informatiile de pe aceste linii sa se fi alterat si astfel sa apara o selectare falsa. Stapinul termina transferul de date prin incetarea emisiei adreselor si comenzilor. Emisia datelor pe liniile D, trebuie oprita de catre stapin oricind dupa primirea pozitionarii lui SSINC, dar nu mai tirziu de incetarea emisiei adreselor si comenzilor pe liniile A si C. Sclavul dupa receptionarea negarii lui MSINC, isi termina partea sa din transferul de date prin negarea lui SSINC. Linia OCUPAT este negata daca stapinul nu mai doreste alt transfer de date sau este mentinuta pozitionat daca urmeaza alt transfer.

O reprezentare etapizata pentru transferul ti-





NOTĂ : Este prezentat cel mai defavorabil caz de întârziere de propagare de 75 ns. Timpul total pentru un singur transfer este de 475 ns, însă următorul ciclu poate începe la 400 ns, când stăpînul poate trimite noile adrese și comenzi. Viteza maximă de transfer este de 2,5 milioane cuvinte/sec. adică 5 Moct./sec.

Fig. 5.9. DIAGRAMĂ DE TIMP PENTRU CICLUL „IESIRE”

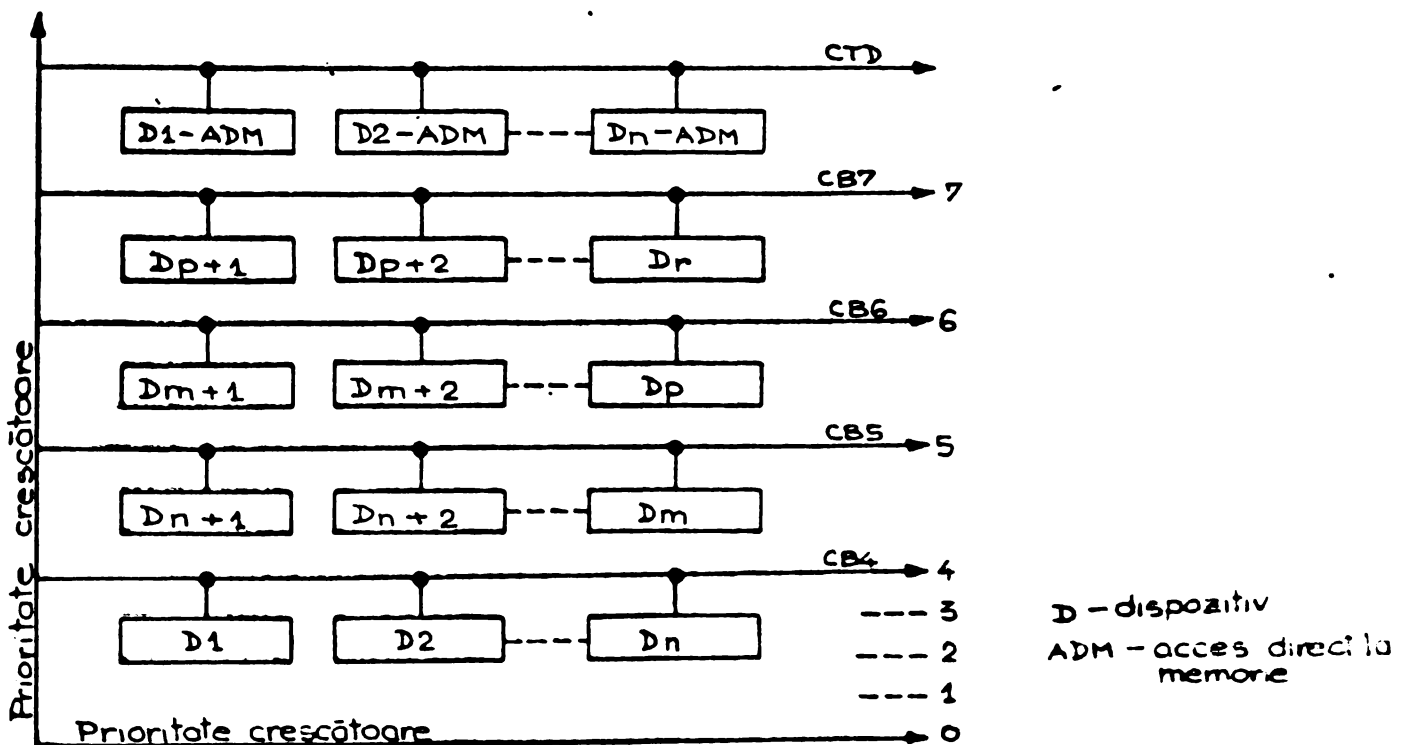


Fig. 5.10. PRIORITATEA PE BUS

pic IESIRE este data in fig.5.8. Deoarece etapele sint asemănătoare cu cele de la transferul INTRARE, iar diagrama este explicita nu se mai face o explicare detaliata. Din diagrama de timp din fig.5.9. rezulta ca un transfer de tipul IESIRE in cazul celor mai defavorabile conditii de propagare se poate executa in maxim 475 ns, insa noul ciclu poate fi inceput dupa numai 400 ns, cind noul stăpin poate trimite adrese si comenzi. Astfel se poate obtine o viteza maxima de transfer de 2,5 milioane cuvinte pe secunda.

#### 5.4. *Secventa de arbitrare a prioritatii si secventa de intrerupere.*

##### 5.4.1. *Structura prioritara.*

Una din caile prin care un dispozitiv poate deveni stăpin pe bus este intreruperea executiei unui program si fortarea registrului de adrese al UC la o adresa unde este localizata o rutina de serviciu specifica intreruperii respective.

La sistemul I-100 prioritatea unui dispozitiv este functie de nivelul de prioritate atribuit dispozitivului si pozitia sa pe bus in raport cu alte dispozitive care au acelasi nivel de prioritate (fig.5.10.). Pentru dispozitivele periferice exista 4 nivele de prioritati, liniile respective de cerere a busului fiind denumite in ordinea prioritatii CB7, CB6, CB5 si CB4. Fiecare dispozitiv periferic are cablata o anumita prioritate. In cadrul aceluiasi nivel de prioritate dispozitivul mai apropiat electric de arbitrul de prioritati, va fi mai prioritar fata de cele mai departate (de ex: in fig.5.11. pe linia CB5 ordinea prioritatii dispozitivelor este 1,3,4,7, iar ca prioritate generala dispozitivul 5 este cu prioritatea cea mai mare).

UC sub controlul programului poate sa-si atribuie oricare din cele 8 nivele de prioritate (fig.5.10.). In acest fel desi nivelul de prioritate al unui dispozitiv este cablat, prioritatea sa se schimba in functie de prioritatea UC.

Cum s-a vazut dispozitivele periferice pot efectua transferuri si direct la alte dispozitive periferice, fara intermediul UC, transferuri numite CTD. Un dispozitiv are acces foarte rapid la bus si dupa ce a primit controlul busului transferul se face foarte eficient fara a afecta starea UC care in aceste momente poate elibera busul si poate trata o instructiune.

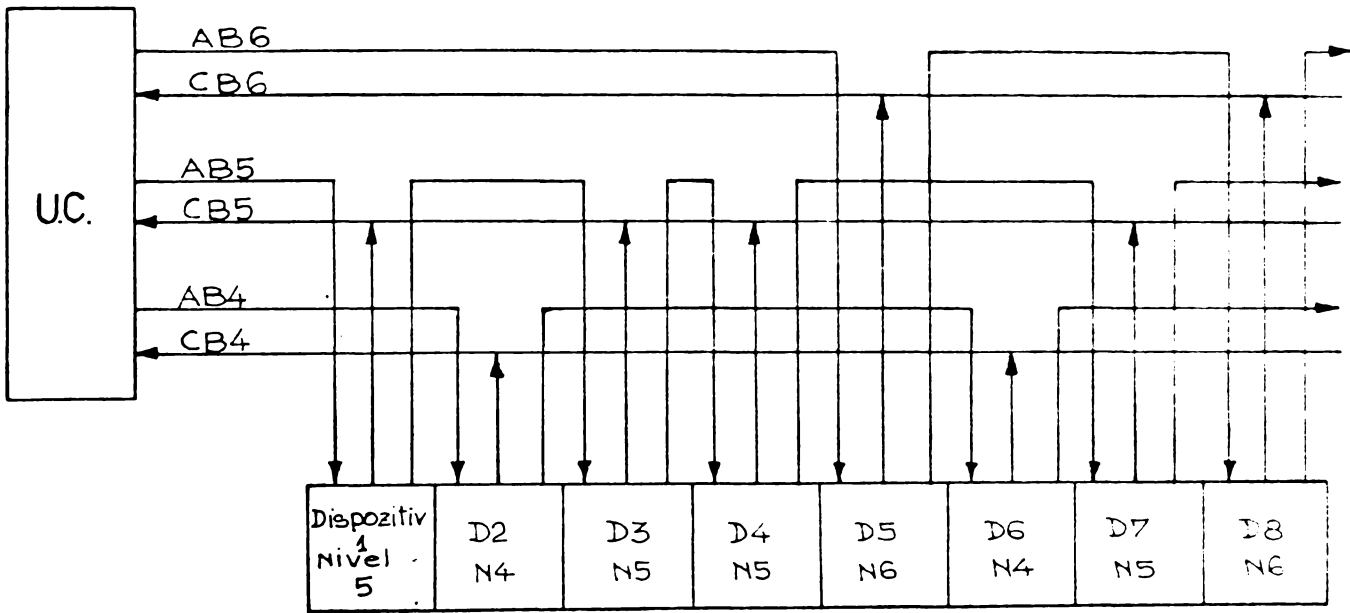


Fig 5.11. LANȚ DE PRIORITATE

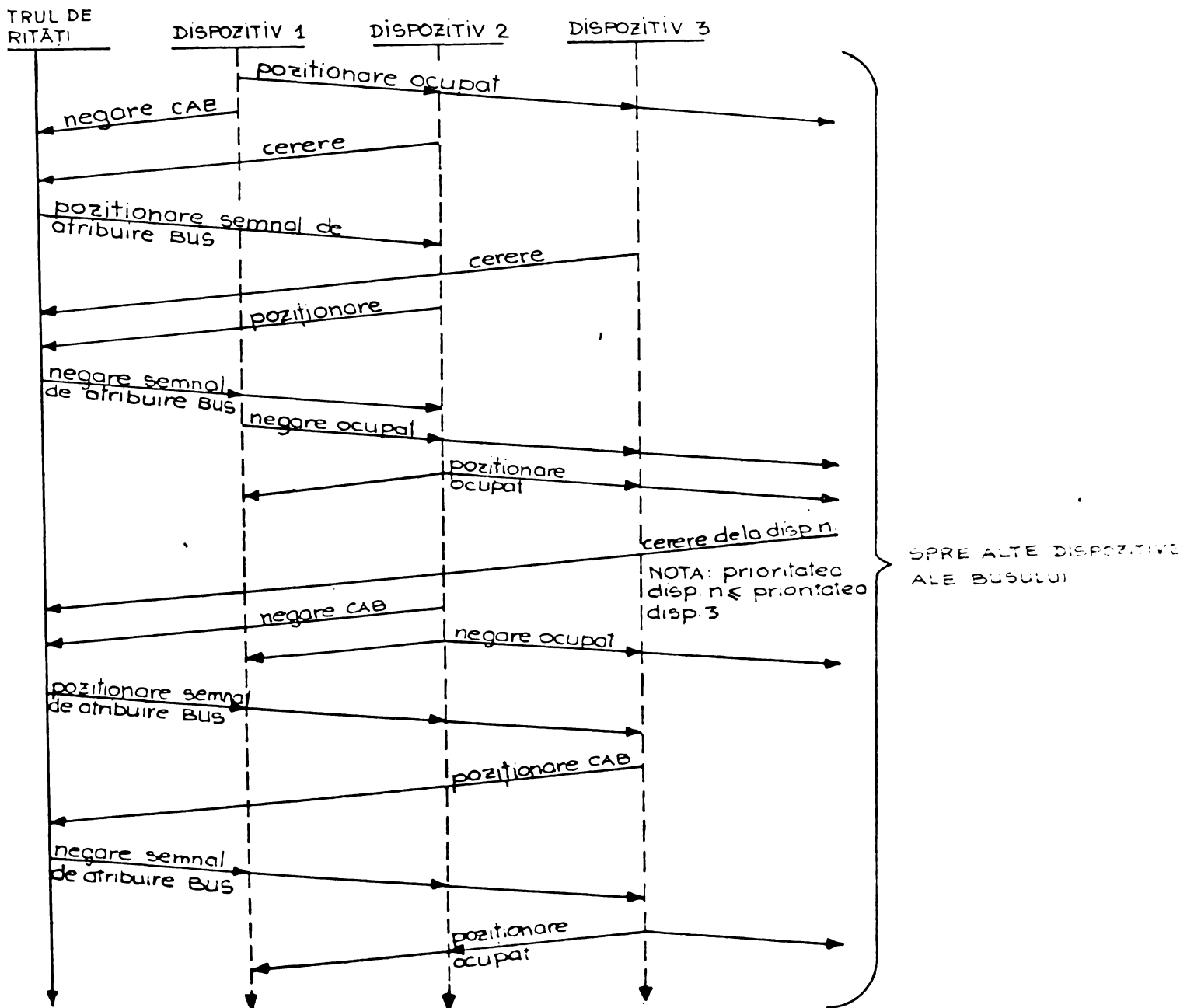


Fig 5.12. SECVENȚA TIPICĂ DE ATRIBUIRE PRIORITĂȚI

Un dispozitiv periferic care dorește busul, trimite o cerere pe linia la care este racordat, cerere care ajunge la un arbitru de prioritati, arbitru ce comanda și nivelul de prioritate al UC. Dacă la arbitru nu s-a primit o cerere la un nivel mai înalt decât nivelul cerut al UC, liniile de date ale busului sînt date UC. În cazul în care UC are un nivel de prioritate inferior, controlul busului a dat dispozitivul cu cel mai înalt nivel de prioritate. Arbitrul de prioritati emite un semnal de acordare a busului pe linia cu prioritatea cea mai mare pe care s-a făcut o cerere. Acest semnal este recepționat de primul dispozitiv conectat pe această linie, care dacă are o cerere activă, blochează transmiterea semnalului mai departe și devine stăpîn pe bus, în caz contrar semnalul este trecut dispozitivului următor. Dacă nu este găsit nici un dispozitiv care să fi cerut busul, semnalul este anulat și procesul de atribuire se reia.

#### 5.4.2. *Secvența de arbitrare a prioritatii.*

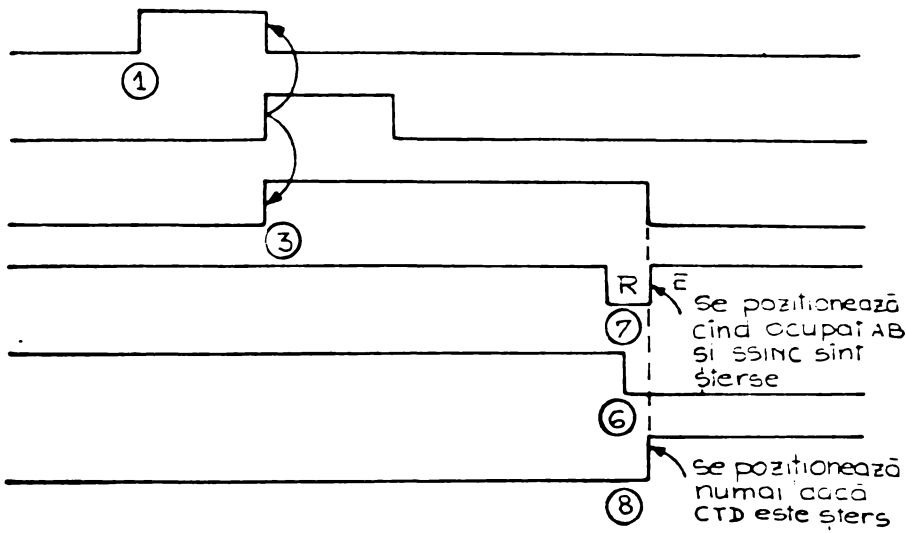
Secvența tipică de arbitrare a prioritatii este dată în fig.5.12.

Dispozitivul 1 devine stăpîn pe secțiunea de date a busului și poziționează semnalul OCUPAT. Cînd termină operația de transfer, dispozitivul 1 va nega semnalul CAB (confirmare acordare bus), autorizînd arbitrul de prioritati să declanșeze o nouă secvență de acordare a busului. Dacă dispozitivul 2 a făcut o cerere de atribuire, se poziționează semnalul de atribuire a busului pe nivelul de prioritate al cererii respective, presupunînd că nu s-a făcut nici o cerere pe un nivel de prioritate mai mare. Poziționarea semnalului de atribuire a busului va dezactiva arbitrul de prioritati, cererea dispozitivului 2 este încă ignorată, însă dispozitivul 2 recunoscînd semnalul de atribuire va poziționa semnalul CAB care va menține în continuare inhibat arbitrul de prioritati, dispozitivul 2 fiind acum desemnat ca următor stăpîn pe bus. Arbitrul de prioritati la recepționarea poziționării semnalului CAB, nega semnalul de atribuire a busului.

Dispozitivul 1 cînd încheie transferul de date, eliberează busul prin negarea semnalului OCUPAT. La recepționarea negării semnalului OCUPAT, dispozitivul 2 recepționează semnalul OCUPAT și începe transferul de date pe bus.

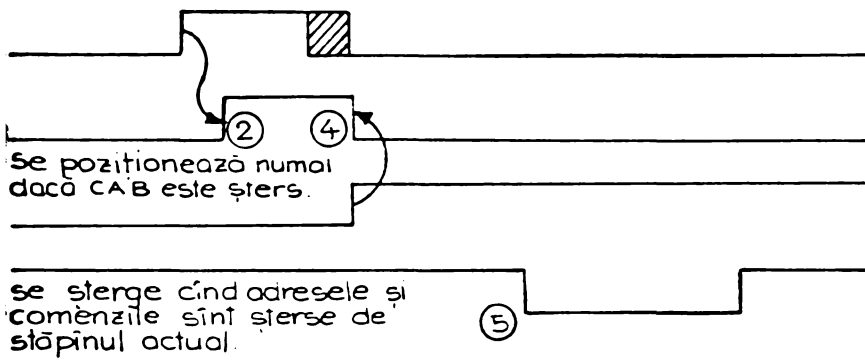
Arbitrul de prioritati nu mai poate onora alte cereri cînd este poziționat un semnal de atribuire a busului (de exemplu: cererea dispozitivului 3) sau cînd a recepționat

LA DISPOZITIV



- (E) CB sau CTD
- (E) AB sau ATD
- (E) CAB
- (E/R) OCUPAT
- (R) SSINC
- (E) INTR

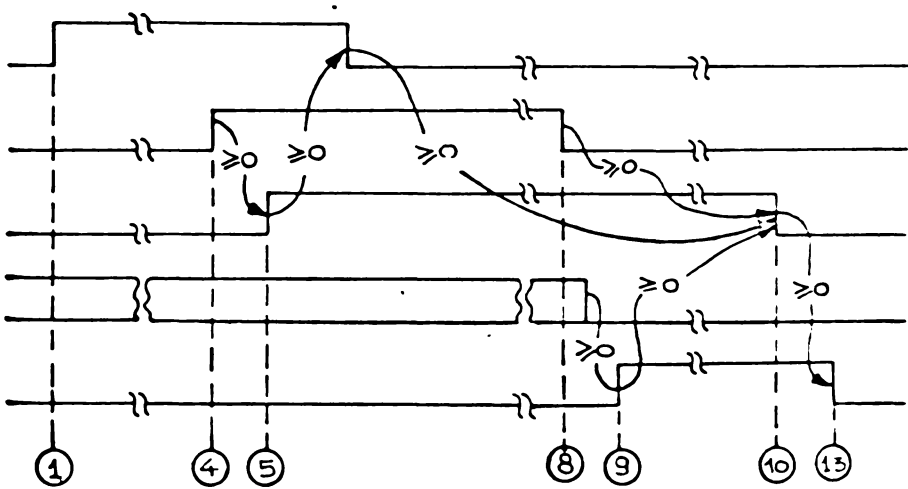
LA UNITATEA CENTRALĂ



- (R) CB sau CTD
- (E) AB sau ATD
- (R) CAB
- (E/R) OCUPAT

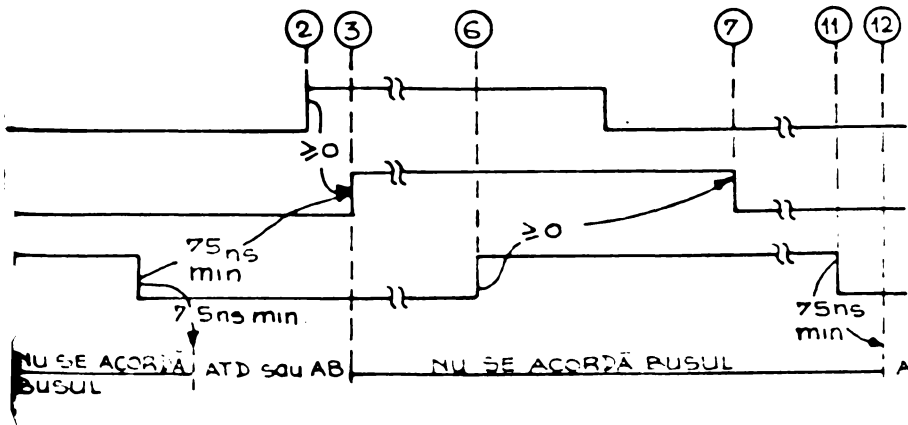
Fig. 5.13. DIAGRAMA DE TIMP A SECVENTEI DE ARBITRARE DE PRIORITĂȚI (unitatea centrală e STĂPÎN)

LA STĂPÎN



- (E) CTD
- (R) ATD
- (E) CAB
- (R) OCUPAT
- (E) OCUPAT

LA ARBITRUL DE PRIORITĂȚI



- (R) CTD
- (E) ATD
- (R) CAB

NU SE ACORDĂ BUSUL ATD sau AB

NU SE ACORDĂ BUSUL ATD sau AB

Fig. 5.14. SECVENȚA TIPICĂ DE ARBITRARE A PRIORITĂȚII ATD

pozitionarea semnalului CAB (exemplu: cererea dispozitivului N, fiind dispozitivul 2 asteapta prea mult inaintea negarii lui CAB). Secventa de arbitrare a prioritatii poate avea loc sau nu simultan cu un transfer de date. In cazul dispozitivului 1 si 2 procesele au loc simultan. In schimb secventa de atribuire a prioritatii pentru dispozitivul 3 nu incepe pina la terminarea transferului de date al dispozitivului 2.

Arbitrul de prioritati poate raspunde la semnalele emise de dispozitivele de pe bus care cer utilizarea sectiunii de date a busului si la semnalele de validare generate de modulul de intrerupere din UC. UC nu permite generarea semnalelor de atribuire a busului de catre arbitrul de prioritati pe timpul desfasurarii unei rutine de intreruperi si un timp dupa incheierea acestei rutine, timp in care UC isi stabileste noul nivel de prioritate. In acest timp se salveaza informatiile referitoare la vechiul nivel si se stabilesc cele care apartin noului nivel. In fig. 5.13. este data diagrama de timp a secventei de atribuire a prioritatii, aratindu-se ce semnale de atribuire a busului sint generate de arbitrul de prioritati. S-a considerat ca UC este stapin.

O impartire pe etape a secventei de atribuire a prioritatii este data in fig. 5.14. Semnalul CTD este pozitionat de dispozitivul care a facut cererea (1) si dupa un timp de propagare este receptionat de arbitrul de prioritati (2). Daca si semnalul CAB negat a fost receptionat de arbitrul de prioritati si s-a asteptat minim 75 nsec., arbitrul de prioritati poate pozitiona ATD (semnal Acordare Bus pentru Transfer Date), procesul de arbitrare a prioritatii fiind oprit (3). Aceasta intirziere de minimum 75 nsec. asigura receptionarea de catre arbitrul de prioritati a negarii semnalului CTD sau CB al secventei anterioare de arbitrare, inainte de incheierea secventei de arbitrare. Se previne astfel generarea unui semnal de atribuire a busului ca raspuns la secventa de arbitrare anterioara, situatie ce poate sa apara daca cererea ar fi negata in acelasi timp cu semnalul CAB. In cazul transferului unui singur cuvint, imediat dupa pozitionarea semnalului OCUPAT, stapinul poate nega CAB, in acest caz intirzierea lui CAB asigurind ca pozitionarea semnalului OCUPAT sa fie sesizata inaintea negarii lui CAB. De asemenea nici un alt semnal de atribuire a busului nu poate fi generat de arbitru in timp ce un ATD este pozitionat.

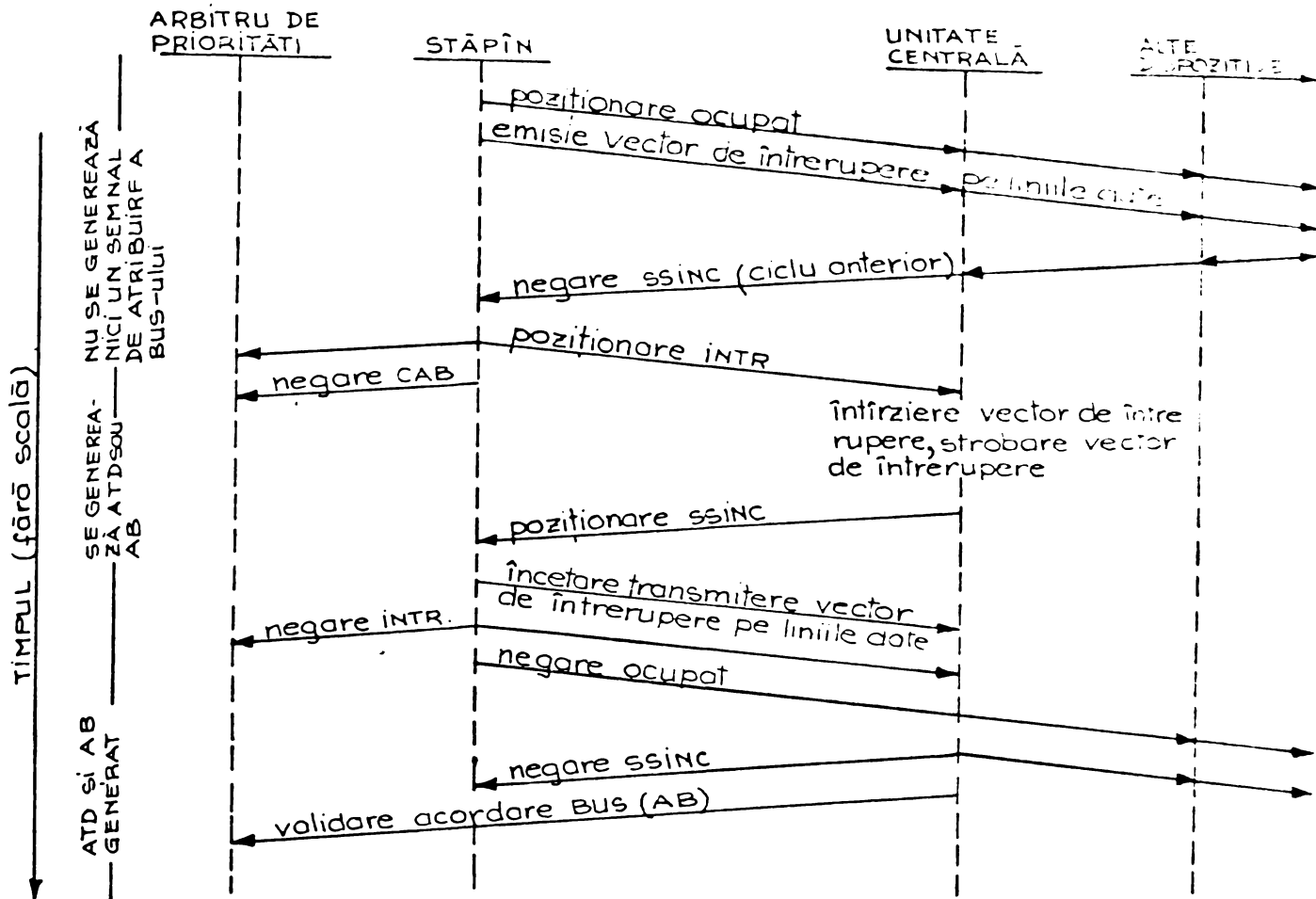


Fig. 5.15. SECVENȚA TIPICĂ DE ÎNTRERUPERE

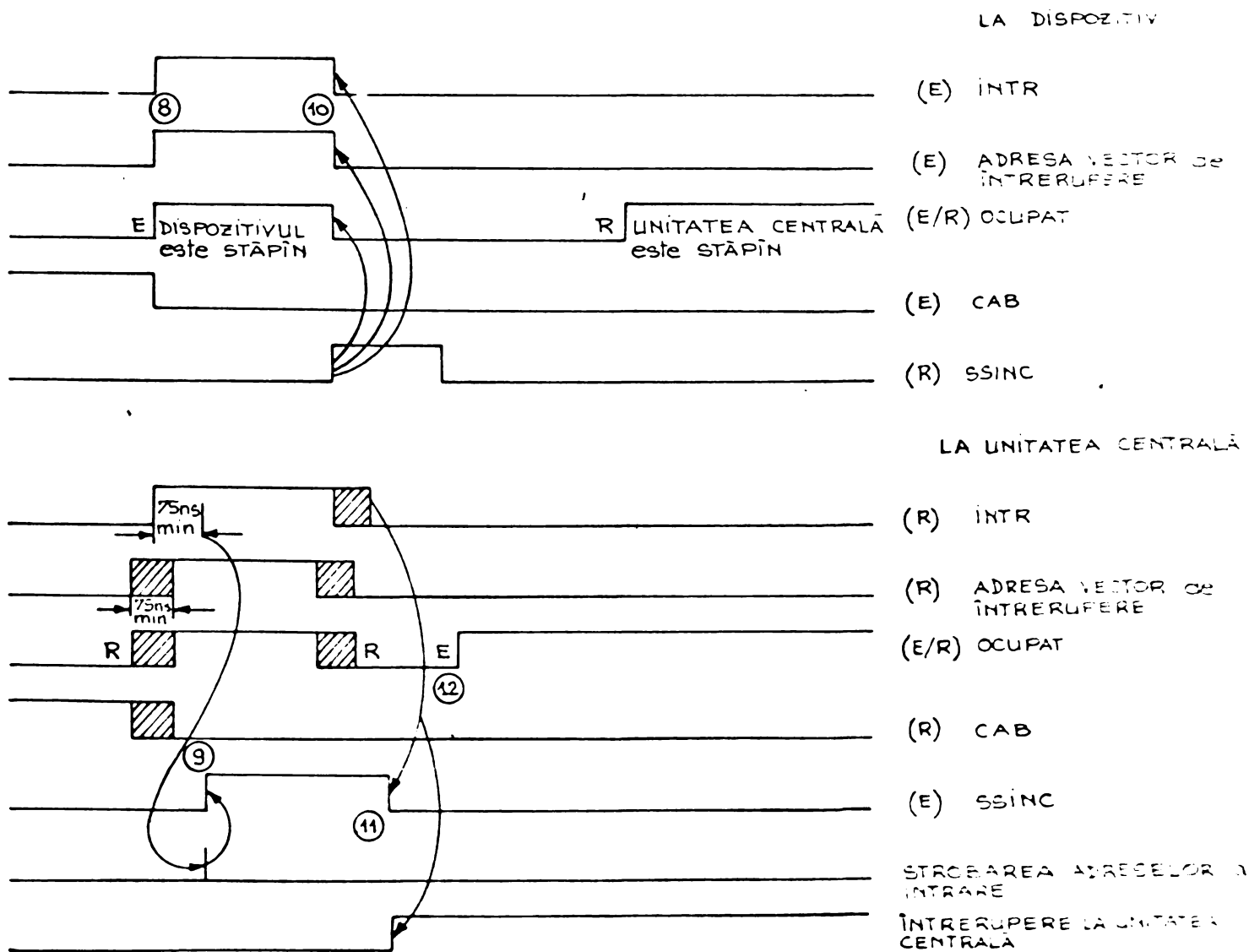


Fig. 5.16. DIAGRAMA DE TIMP PENTRU SECVENȚA DE ÎNTRERUPERE

Dupa un timp de propagare ATD este receptionat de dispozitivul care a solicitat busul (4), care la rindul sau pozitioneaza CAB (5). In cazul transferului unui singur cuvint, dupa pozitionarea lui CAB trebuie negat si CTD, inasa inainte de negarea lui CAB. Daca se cere un nou transfer in continuare, CTD va ramine pozitionat in continuare.

Dupa o intirziere de propagare arbitrul de prioritati receptioneaza semnalul CAB (6). Daca CAB nu este receptionat de arbitru dupa un interval de 5 - 10 usec. de la pozitionarea lui ATD, ATD este negat si secventa de arbitrare a prioritatii se reia. Aceasta intirziere se poate evita daca la sfirsitul busului (arbitrul se afla la inceputul busului), se monteaza un terminator care pozitioneaza CAB in momentul receptionarii pozitionarii semnalului ATD. Negarea lui ATD este propagata pe bus pina la terminator, care la rindul sau neaga CAB dupa receptionarea negarii lui ATD. Apoi se executa direct etapele finale (11) si (12).

Acum arbitrul de prioritati neaga ATD (7), front care dupa un timp de propagare este receptionat de dispozitivul care a facut cererea, (8). Dupa receptionarea negarii semnalului OCUPAT, dispozitivul care a facut cererea pozitioneaza OCUPAT (9), devine stapin pe bus si poate incepe ciclul sau de transfer de date.

Dupa pozitionarea lui OCUPAT, dar inaintea terminarii transferului de date, stapinul poate nega CAB daca a receptionat negarea lui ABn(10). Daca se intentioneaza transferarea unui singur cuvint, dispozitivul pozitioneaza OCUPAT si neaga CAB simultan. Pentru a se asigura receptionarea semnalului CAB la arbitrul de prioritati, stapinul va trebui sa nege CAB inaintea negarii lui ATD.

Dupa un timp de propagare, arbitrul receptioneaza ATD (11) si dupa o asteptare de minimum 75 ns reia secventa de arbitrare (12). Dispozitivul stapin, la sfirsitul ultimului sau ciclu de transfer, asteapta cel putin 75 ns dupa negarea lui MSINC si inceteaza emisia pe liniile A,C si D, apoi neaga semnalul OCUPAT eliberind busul (13). CAB este negat inaintea negarii semnalului OCUPAT.

#### 5.4.3. *Secventa de intrerupere.*

Cererea de serviciu a unui dispozitiv care cere folosirea busului pe unul din cele patru nivele, este satisfacuta dupa efectuarea urmatoarelor faze:

a). in urma verificarii prioritatii, UC cedeaza contro-



lul busului dispozitivului care a facut cererea.

b). la obtinerea controlului busului, dispozitivul trimite spre UC o comanda de intrerupere (AVI), adresa la care incepe rutina de tratare a intreruperii specifice dispozitivului respectiv. La adresa urmatoare (AVI + 2) se afla un cuvint care va fi folosit ca nou cuvint de stare (CS) al UC.

c). CS curent si continutul numaratorului de instructiuni curent (NI) sint trimise de UC intr-o stiva de memorie.

d). noul vector de intrerupere (format din noile CS si NI) este luat de la adresele specificate de dispozitiv si se initiaza rutina de serviciu a dispozitivului.

e). rutina de serviciu a dispozitivului poate determina reluarea programului intrerupt, prin executarea instructiunii de intoarcere din intrerupere (RII) care actualizase CS si NI cu valorile memorate in stiva.

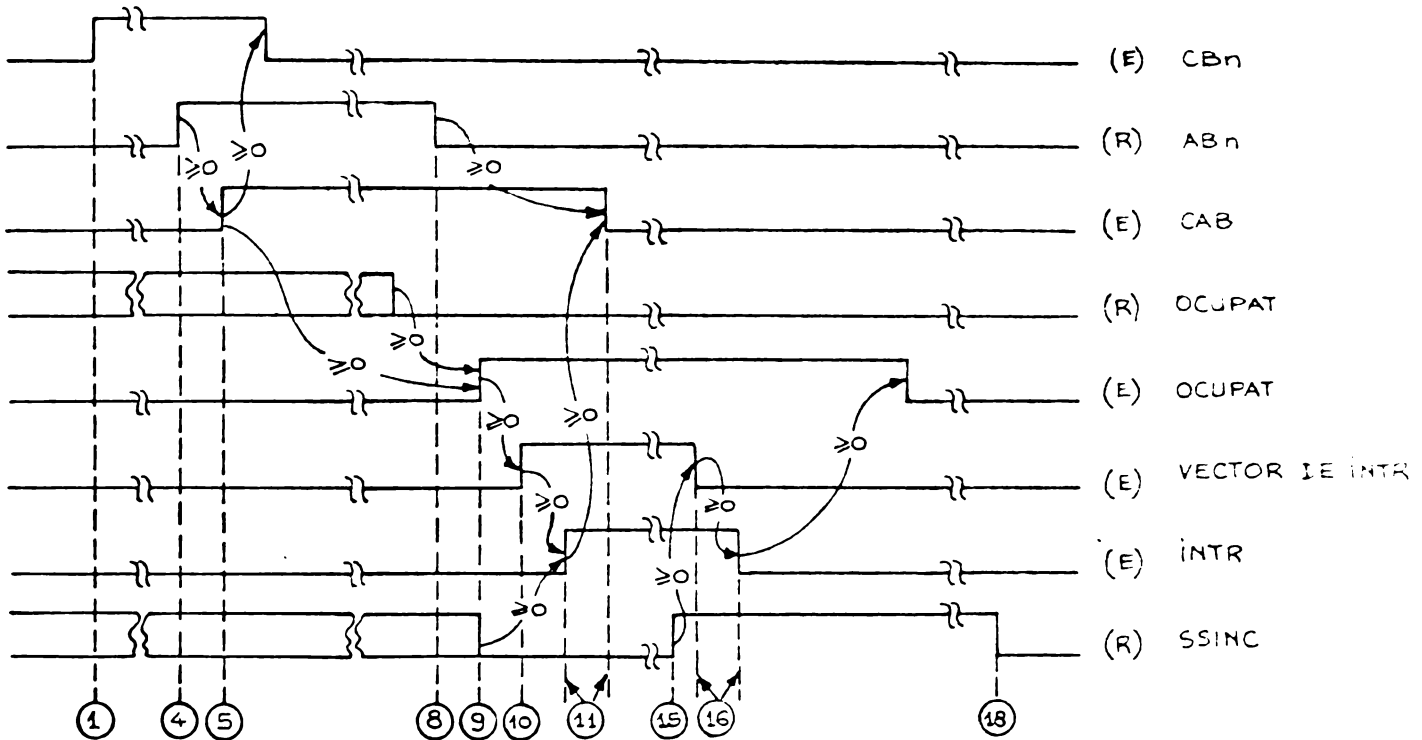
f). rutina de serviciu a oricarui dispozitiv poate fi la rindul ei intrerupta de o cerere mai prioritara, oricind dupa executarea primei instructiuni. Procesul se poate repeta pina la umplerea stivei din memorie.

In fig.5.15. este prezentata o secventa tipica de intrerupere cu interactiunea intre arbitrul de prioritate, dispozitivul stapin si UC. Stapinul dupa obtinerea controlului asupra sectiunii de date a busului printr-o secventa de arbitrare a prioritatii  $CB_n - AB_n$ , poate emite un semnal de intrerupere spre UC. Vectorul de intrerupere este plasat pe liniile D si daca SSINC este negat, se pozitioneaza INTR si se neaga CAB. Dupa receptionarea pozitionarii lui INTR, UC asteapta stabilizarea liniilor D, strobeaza vectorul de intreruperi si pozitioneaza SSINC. Stapinul la receptionarea pozitionarii lui SSINC, inceteaza emisia vectorului pe liniile D si neaga semnalele INTR si OCUPAT. UC la primirea negarii lui INTR, neaga SSINC. Arbitrul de prioritati inceteaza sa emita semnalele de arbitrare a busului AB, dupa receptionarea pozitionarii lui INTR si nu va emite semnale AB pina ce nu va fi autorizat de UC. In schimb poate emite semnale de tip CTD. O alta ilustrare a secventei de intrerupere este data in fig. 5.16. Etapizarea acestei secvente este pusa in evidenta in diagrama din fig.5.17. Dispozitivul care a facut cererea de intrerupere pozitioneaza  $CB_n(1)$ , care dupa un timp de propagare este receptionata de arbitrul de prioritati (2). La arbitrul de prioritati se poate pozitiona semnalul  $AB_n$  de atribuire a prioritatii (3) si procesul de arbitrare se

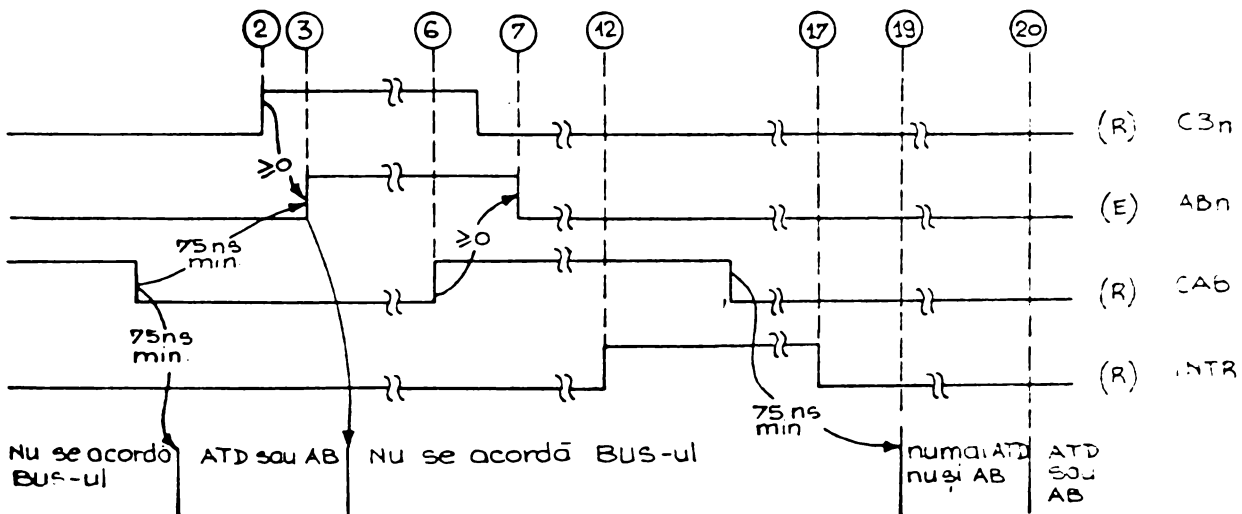
COA-  
SEIM  
PWIJ

sidera incheieat, daca negarea semnalului CAB din ciclul anterior de arbitrare fusese receptionata de cel putin 75 ns si daca UC este gata de a accepta un vector de intrerupere pe nivelul dispozitivului care a cerut intreruperea. Dupa un timp de propagare semnalul AB este receptionat de dispozitivul care a facut cererea (4), acesta facind positionarea lui CAB (5). In cazul unui singur transfer CBn trebuie sa fie negat de dispozitivul care a facut cererea, dupa positionarea lui CAB, inasa inaintea negarii lui CAB. In cazul ca se cere un alt transfer dupa cel in curs de desfasurare, CBn poate ramine positionat. Positionarea semnalului CAB este receptionata la arbitrul de prioritati dupa scurgerea unui timp de propagare (6). Daca positionarea lui CAB nu ajunge la arbitrul de prioritati in 5 - 10 us dupa positionarea lui ABn, ABn va fi negat si secventa de arbitrare a prioritatii se reia. Aceasta intirziere poate fi eliminata prin plasarea terminatorului (similar ca la pct.6.4.2.). Semnalul ABn este negat de arbitrul de prioritati (7), front negativ care dupa un timp de propagare este receptionat de dispozitivul care a facut cererea (8). La acest dispozitiv dupa receptionarea negarii semnalului OCUPAT, se positioneaza OCUPAT (9) dispozitivul devenind stapin pe bus. Acum dispozitivul stapin care a obtinut printr-un semnal AB (nu prin ATD) dreptul de utilizare a sectiunii de date a busului, va plasa pe liniile de date vectorul de intrerupere (10). Dupa receptionarea negarii lui SSINC stapinul positioneaza semnalul INTR, si dupa receptionarea negarii lui ABn, va nega CAB (11) (INTR trebuie sa fie positionat inainte ca CAB sa fie negat, asigurandu-se astfel receptionarea positionarii lui INTR inainte ca intirzierea lui CAB la arbitrul de prioritati sa se termine. In acest fel intirzierea lui CAB compenseaza diferenta intre timpul de propagare al semnalului INTR si CAB la arbitrul de prioritati. De asemenea stapinul nu trebuie sa nege CAB inainte de receptionarea negarii lui ABn, pentru a se asigura ca arbitrul de prioritati a receptionat positionarea lui CAB). Positionarea semnalului INTR este receptionata dupa un timp de propagare de arbitrul de prioritati si UC (12). UC asteapta cel putin 75 ns (intirziere vector de intrerupere) si apoi strobeaza vectorul de intrerupere pe liniile D (13). UC poate acum positiona semnalul SSINC (14), care este receptionat de stapin dupa un timp de propagare (15). Stapinul inceteaza emisia vectorului de in-

LA DISPOZITIVUL CARE A FACUT CEREREA



LA ARBITRUL DE PRIORITATE



LA UNITATEA CENTRALA

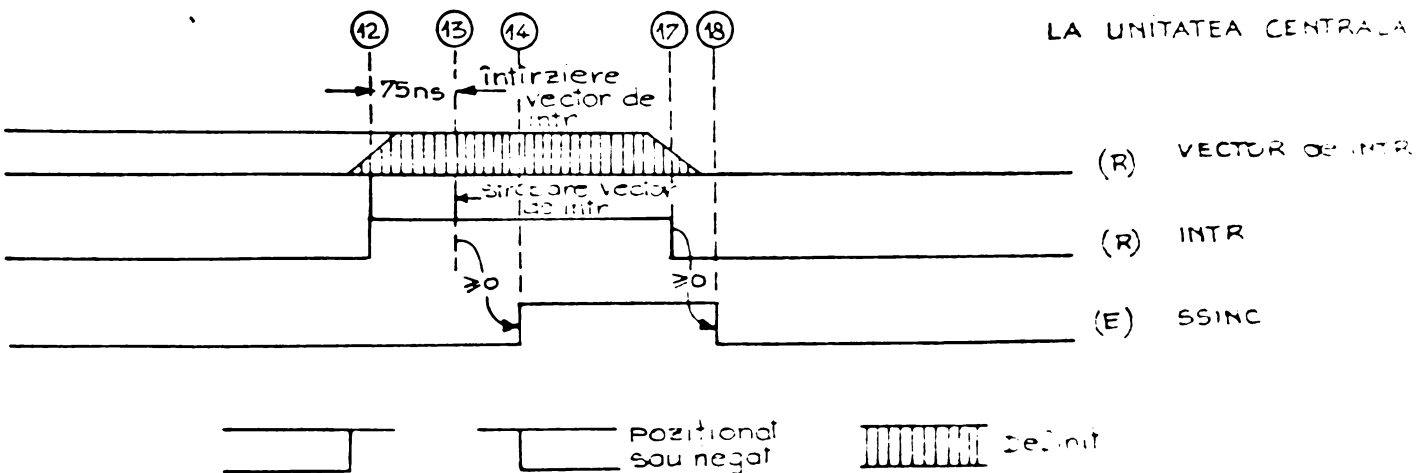


Fig 5.17. DIAGRAMA DE TIMP PENTRU SECVENTA DE ÎNTRERUPERE AB

trerupere pe liniile D, neaga INTR si apoi neaga OCUPAT ceea ce reprezinta eliberarea sectiunii de date a busului (16). Negarea lui INTR dupa un timp de propagare este receptionata de catre arbitrul de prioritati si UC (17). In etapa urmatoare (18) UC neaga semnalul SSINC.

Dupa receptionarea negarii semnalului CAB (in etapa 11) arbitrul de prioritati asteapta 75 ns si apoi poate relua secventa generind semnale pentru ATD, dar nu pentru AB (19). In mod normal din locatiile de memorie date de vectorul de intrerupere UC citeste imediat dupa intrerupere noua adresa program si cuvintul de stare. In acest fel UC isi poate determina noul sau nivel de intrerupere. Apoi UC informeaza arbitrul de prioritate ca poate incepe generarea semnalelor AB (20). De remarcat ca stapinul poate executa o singura secventa de intrerupere ca urmare a unui semnal de acordare a busului AB.

5.5. *Analizarea structurii cu mai multe UC.*

Conceptia interfetelor sistemului permite realizarea de structuri cu mai multe UC. Pentru fiecare UC trebuie sa se adauge semnalele de cerere a busului pe cele patru nivele de intrerupere (4,5,6 si 7) impreuna cu liniile de acordare a busului precum si o linie de identificare a unitatii. In tabelul 5.5. se indica aceste linii suplimentare pentru un sistem la care s-au conectat inca trei UC.

TABELUL 5.5.

Denumire	Mnemonică	Numar linii	Functia
Cerere bus pentru intreruperi	CB4B,C,D	12	Cereri de intreruperi emise de unitatile centrale B,C si D.
	CB5B,C,D		
	CB6B,C,D		
	CB7B,C,D		
Acordare bus	AB4P	4	Analog cu CTD pentru o unitate centrala.
	AB5P		
	AB6P		
	AB7P		
Linii de identificare a unitatii	IDA	4	Indica UC-ul care lucreaza cu busul.
	IDB		
	IDC		
	IDD		

In cazul lucrului cu mai multe UC arbitrul de prioritati se modifica, de asemenea cuploarele vor fi modificate pentru a raspunde la semnalele suplimentare. In sistem, la aparitia unor conflicte de acces in cazul cererilor simultane adresate aceleiasi resurse, UC-urile au prioritati preferentiale unele fata de altele.

## CAP.6. ALOCAREA TIMPULUI DE LUCRU AL SDM

### 6.1. *Strategii de alocare.*

O strategie de alocare are ca scop sa satisfaca toate cererile, insa cu respectarea unor conditii cum ar fi:

- garantarea unui timp dat de alocare;
- respectarea unei prioritati;
- terminarea lucrului inainte de un termen limita impus anterior;
- anulara lucrului dupa un timp limita impus anterior.

Dupa modul de organizare strategiile pot fi clasificate in urmatoarele categorii:

- strategii fara reciclaj;
- strategii cu reciclaj.

#### 6.1.1. *Strategii fara reciclaj.*

##### 6.1.1.1. Sirul de asteptare simplu (FIFO)

Lucrarile sint tratate in ordinea venirii, singurul criteriu de prioritate fiind momentul prezentarii. O lucrare lunga duce la intirzierea tuturor lucrarilor care o urmeaza (fig. 6.1.).

6.1.1.2. Sirul de asteptare ordonat dupa criteriul de timp de executie.

Fiecare lucrare nou venita este plasata in sir, dupa timpul estimat de lucru. Aceasta strategie este dezavantajoasa pentru lucrarile lungi care pot fi intirziate foarte mult. O ameliorare se poate aduce prin combinarea cu timpul de asteptare. O alta posibilitate este ca timpul estimat de executie pentru procesul nou venit sa fie comparat cu timpul estimat ramas al lucrarii in curs si daca este mai mic lucrarea in curs este oprita cedind locul

lucrării nou venite, lucrarea intreruptă reintorcându-se în sirul de așteptare într-un loc corespunzător timpului care a mai rămas de executat. Și în acest caz lucrările scurte au fost avantajate (fig. 6.2.).

### 6.1.2. Strategii de reciclaj

#### 6.1.2.1. Reciclajul cu un singur sir de așteptare.

La această strategie timpul de alocare este împartit în "Cuante" de timp notate cu  $q$ , fiecare proces primind procesorul alternativ pe durata  $q$ , fiind apoi plasat în coada sirului de așteptare (fig. 6.3.).

Strategia prezintă avantajul că orice lucrare va primi procesorul după un anumit interval de timp. O problemă este alegerea valorii cuantei.

#### 6.1.2.2. Reciclajul cu mai multe siruri de așteptare

Și această strategie avantajează lucrările mai scurte. Se introduc mai multe siruri de așteptare care pot avea cuante de aceeași valoare sau diferite (fig. 6.4.).

Procesele care cer procesorul sunt aranjate în  $n$  siruri  $S_1, S_2, \dots, S_n$ , fiecărui sir  $S_i$ , fiindu-i atașată o cuantă de timp  $q_i$ . Procesele nou venite intra în sirul  $S_1$ . Lucrarea aflată în capul sirului  $S_i$  ( $i > 1$ ) nu poate fi luată în considerare decât dacă toate sirurile  $S_j$  ( $0 < j < i$ ) sunt goale. Dacă o lucrare din sirul  $S_i$  nu este terminată până la terminarea cuantei  $q_i$  ea va trece în coada sirului  $S_{i+1}$ . Lucrările care ies din sirul  $S_n$  vor intra tot în sirul  $S_n$  dacă nu s-au terminat pe durata cuantei  $q_n$ . Lucrările noi care sosesc în sirul  $S_1$  în timpul execuției unei lucrări din sirul  $S_i$  ( $i > 1$ ), sunt luate în lucru după expirarea cuantei  $q_i$ . Dacă sirul  $S_1$  este gol lucrările noi vor fi luate în considerare imediat.

### 6.2. Model de alocare.

Se stabilesc următoarele reguli de alocare :  
orice lucrare primește odată o singură cuantă, o lucrare liberează subsistemul atunci când se termină sau când

ajunge la sfirsitul cuantei repartizate, cind subsistemul s-a eliberat este alocat lucrarii care a primit cele mai putine cuante. Este utilizata deci o strategie de ordonare cu mai multe nivele, lucrarile fiind aranjate in siruri cu prioritati diferite: toate lucrarile terminate care au terminat i cuante vor fi plasate in siruri i + 1 in ordinea ajungerii in aceasta situatie (FIFO), prioritatea descrescind odata cu cresterea numarului sirului (fig. 6.5.).

Se presupune ca cererile sosesc dupa o distributie Poisson, cu un debit mediu  $\lambda$ , timpul de executie fiind o distributie discreta oarecare  $q_j$ , reprezentind probabilitatea de a dura j cuante. Se doreste sa se stabileasca timpul de raspuns mediu  $W(i)$  pentru o lucrare cu i cuante. Acest timp de raspuns se poate imparti in doua: timpul mediu de asteptare A datorat celorlalte lucrari si timpul de executie. Neglijind timpul de comutatie de la o lucrare la alta putem scrie:

$$6.1.) \quad W(i) = A + i \cdot q$$

Timpul de asteptare se datoreaza pe de o parte lucrarilor incepute dinainte si inca neterminate si pe de alta parte datorita unor lucrari noi sosite in timpul desfasurarii lucrarii si care primesc prioritate pentru prima lor cuanta.

Considerind o lucrare l de i cuante, timpul de asteptare se poate descompune in:

$$6.2.) \quad A = A_1 + A_2$$

unde:

$A_1$  - timpul mediu consumat pentru terminarea cuantei in curs de executie si pentru achitarea tuturor cererilor venite inaintea lucrarii l si situate in primele i siruri de asteptare;

$A_2$  - timpul necesar pentru achitarea cererilor venite dupa l, pina la a (i - 1) cuanta inclusiv; incepind cu momentul in care l incepe prelucrarea in ultima sa cuanta i noile cereri nu mai sint luate in considerare.

#### 6.2.1. Calculul timpului $A_2$

Fie  $G(n)$  functia de repartitie a timpilor de exe-

le.

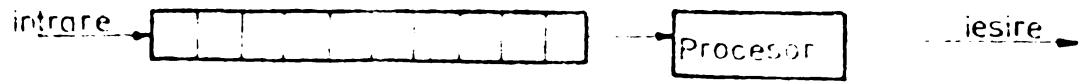


Fig. 6.1. Schema tratarii in sir de asteptare simplu.

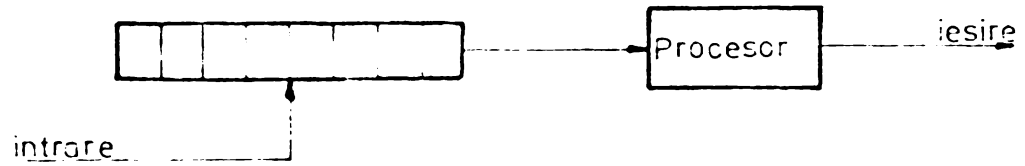


Fig. 6.2. Schema tratarii in sir de asteptare ordonat.

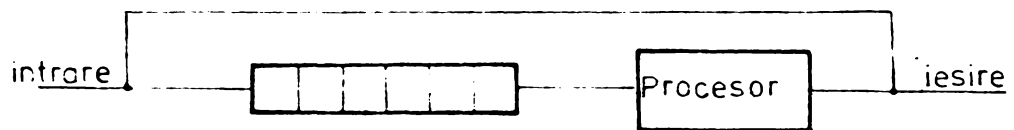


Fig. 6.3. Schema tratarii cu reciclaj cu un singur sir de asteptare.

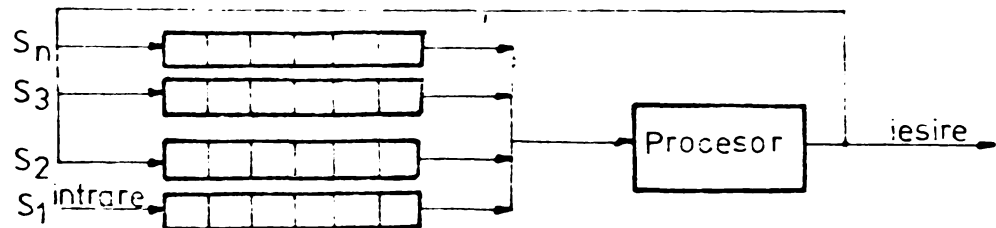


Fig. 6.4. Schema tratarii cu reciclaj cu mai multe siruri de asteptare.

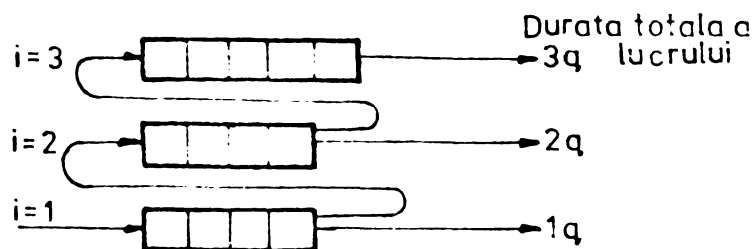


Fig. 6.5. Schema de tratare a modelului propus.



$$(6.3.) \quad G(n) = \sum_{j=1}^n g_j$$

Timpul mediu consumat de subsistem  $S(n)$ , pentru o cerere care a primit cel mult  $n$  cuante se poate scrie:

$$(6.4.) \quad S(n) = \sum_{j=1}^n j \cdot q \cdot g_j + n \cdot q \cdot (1 - G(n))$$

unde termenul al doilea corespunde lucrurilor cu mai mult de  $n$  cuante.

In timpul  $A$  de asteptare al lucrarii  $l$  si a executiei primelor sale  $i - 1$  cuante, numarul de "sosiri" creste in medie de:

$$(6.5.) \quad \lambda [A + (i - 1) \cdot q]$$

Atita timp cit lucrarea  $l$  nu a primit ultima sa cuanta, fiecare din noile lucrari va primi in medie  $S(i - 1)$  cuante, de unde:

$$(6.6.) \quad A = \lambda [A + (i - 1) \cdot q] \cdot S(i - 1)$$

#### 6.2.2. Calculul timpului $A_1$

Sa consideram o politica ajutatoare, pentru calculul timpului  $A_1$  si anume: toate cererile din primul sir pot primi pina la  $i$  cuante, insa in sirurile de prioritate inferioara (urmatoarele) pot primi numai cite o singura cuanta.

Deci o lucrare data  $l$  cu lungimea de  $i$  cuante va astepta deci in primul sir si va fi servita o singura data.

Fie  $A_3$  timpul de asteptare al lucrarii  $l$ , acest timp fiind datorat cererilor precedente lui  $l$  din primul sir si lucrurilor in curs de executie. Ne propunem sa aratam ca timpul de asteptare  $A_3$  este egal cu timpul de asteptare  $A_1$  din strategia initiala.

Subsistemul poate sa se gaseasca in doua stari: activ sau in asteptare. Cind este activ poate avea de rezolvat lucrari cu durata mai mica sau egala cu  $i$  cuante sau lucrari cu durata mai mare de  $i$  cuante. In fig. 6.2. s-au figurat pe axa timpului cele trei situatii marcate prin fazele 0, 1 si respectiv 2.

In ambele cazuri timpul de gestiune al lucrurilor este neglijat, timpul total de executie a unei grupe date de lucrari nedepinzind deci de politica aleasa, distributia perioadelor de "asteptare" fiind aceeași pentru ambele

le cazuri.

Intr-un interval de activitate distributia fazelor 1 si 2 este independenta de asemenea de politica aleasa. Deci distributia celor trei faze este independenta de politica aleasa.

Reluind cererea 1 de  $i$  cuante:

- daca se prezinta cind sistemul este intr-o perioada de "Asteptare" timpul de asteptare este nul pentru cele doua politici;

- daca se prezinta cind sistemul este intr-o faza 2, cererea 1 asteapta sfirsitul cuantei in curs de executie, asteptarea fiind de asemenea identica la ambele cazuri;

- daca se prezinta cind sistemul este intr-o faza 1 se poate sa nu se considere decit lucrarile sosite inainte de lucrarea 1 si care au mai putin de  $i$  cuante. Timpul de executie al acestor lucrari pina la cuanta  $i$  este independenta de politica. Timpul de asteptare al lucrarii 1 este egal cu timpul total de executie a acestor lucrari (identic pentru cele doua politici) mai putin suma timpilor deja primiti. Aceasta suma este egala cu timpul ce separa inceputul fazei 1 de momentul aparitiei lucrarii 1, ca fiind deci aceeasi pentru ambele variante. Deci timpul de asteptare al lucrarii 1 este acelasi, de unde:

$$(6.7.) \quad A_3 = A_1$$

In cazul politicii ajutatoare putem scrie:

$$(6.8.) \quad A_3 = R(i) + m.S(i)$$

unde:  $R(i)$  este timpul mediu pentru terminarea lucrarii in curs

$m$  este lungimea medie a primului sir

Facind apel la relatia lui Little care se aplica unui sistem de echilibru in care intra cereri cu un debit mediu de sosire  $a$ , putem considera ca numarul mediu  $L$  de cereri in sistem este egal cu produsul dintre  $a$  si timpul mediu de rezidenta in sistem notat cu  $S$ :

$$(6.9.) \quad L = a.S$$

Relatia (6.9.) este valabila in ambele cazuri.

Se obtine:

$$(6.10.) \quad m = \lambda A_3$$

Rezulta:

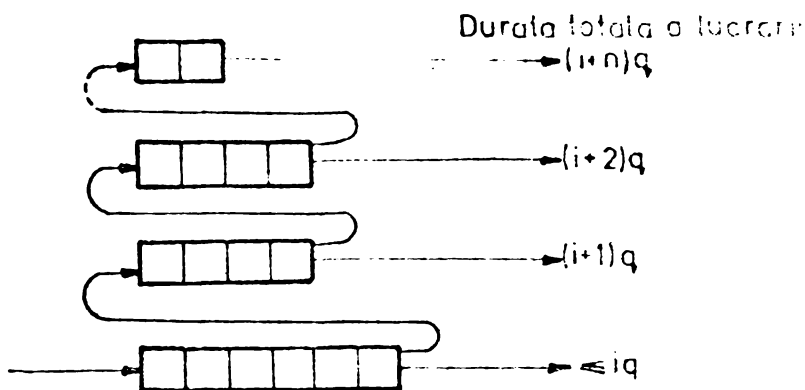


Fig. 6.6. Schema de tratare a modelului dupa politica ajutatoare.

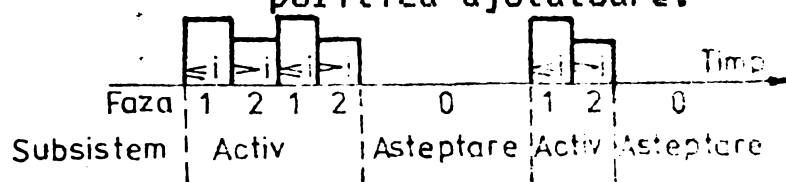


Fig. 6.7. Starea si fazele posibile ale subsistemului.

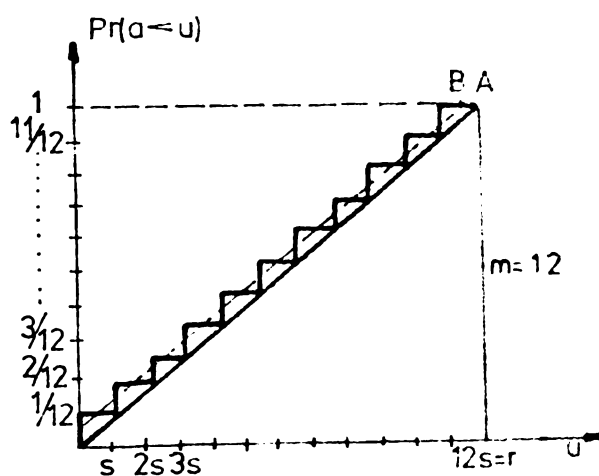


Fig. 6.8. Functia de repartitie a timpului de acces la un sector.

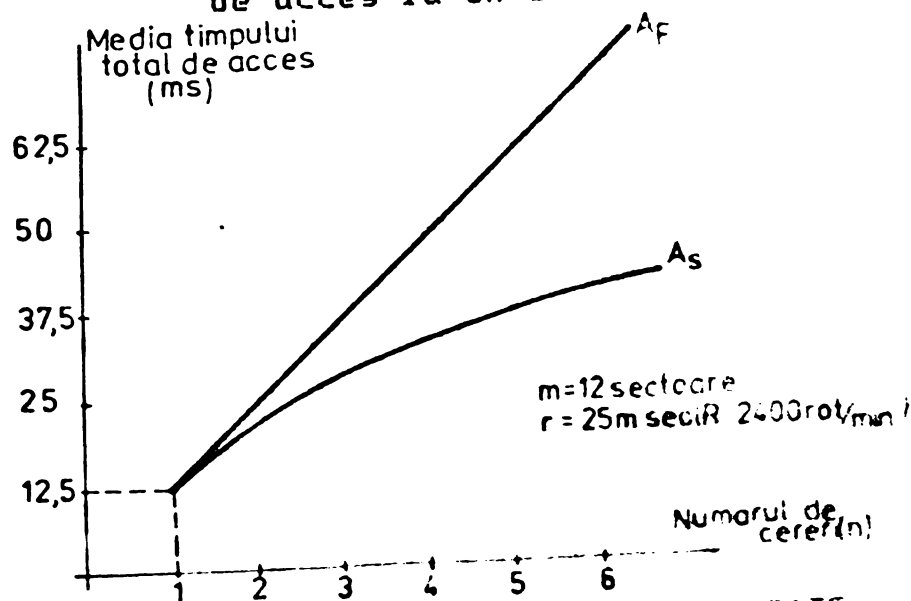


Fig. 6.9. Comparare intre metoda FIFO si SATF.

$$(6.11.) \quad A_3 = \frac{R(i)}{1 - \lambda S(i)}$$

Fie  $S_2(i)$  media patrata a timpului de serviciu

pina la cuanta  $i$  :

$$(6.12.) \quad S_2(i) = \sum_{j=1}^i (i \cdot q)^2 \cdot g_j + (i \cdot q)^2 \cdot [1 - G(i)]$$

Dupa Conway avem:

$$(6.13.) \quad R(i) = \frac{\lambda}{2} \{ S_2(i) + q^2 \sum_{j=i}^{\infty} [1 - G(j)] \}$$

Inlocuind (6.11.) si (6.6.) in (6.2.) rezulta ecuatia:

$$(6.14.) \quad A = \lambda [A + (i-1) \cdot q] \cdot S(i-1) + \frac{R(i)}{1 - \lambda S(i)}$$

de unde se poate extrage valoarea lui  $A$ .

Final timpul de raspuns mediu (6.1.) pentru strategia adoptata este:

$$(6.15.) \quad W(i) = \frac{\lambda}{2} \cdot \frac{S_2(i) + q \sum_{j=i}^{\infty} [1 - G(j)]}{[1 - S(i)] \cdot [1 - \lambda S(i-1)]} + \frac{(i-1) \cdot q}{1 - \lambda S(i-1)} + q$$

In cazul unei politici FIFO, timpul de raspuns este dat de formula lui Conway:

$$(6.16.) \quad W = S + \frac{\lambda}{2} \cdot \frac{S_2}{1 - S}$$

unde:

$$S = \sum_{j=1}^{\infty} (i \cdot q) \cdot g_j \quad \text{si} \quad S_2 = \sum_{j=1}^{\infty} (j \cdot q)^2 \cdot g_j$$

In modelul de alocare considerat s-a urmarit de asemenea favorizarea lucrarilor cu durata scurta, fara a cunoaste dinainte timpul estimat de executie al acestora. Strategia aleasa modificata conform politicii ajutatoare este adecvata utilizarii SDM care lucreaza cu mai multe UDM in sistem de calcul multiprocesor, problema ce va fi tratata ulterior (subcapitolul 6.4.).

Modelul constituit poate fi utilizat pentru analiza cu calculatorul a variantei optime, avind posibilitatea varierii simple a parametrilor. Aceleasi cercetari pot fi realizate si cu un simulator, insa metoda modelarii matematice a fost mai usor de aplicat in acest caz. Simulatorul ar fi avut o complexitate hardware prea mare si din motive de ordin economic nu a putut fi realizat.

6.3. *Calculul timpului de acces la informatie pentru UDM.*

Pentru simplificarea calculelor ne plasam in cazul unitatilor de discuri cu capete fixe, rezultatele obtinute fiind utilizabile si pentru discurile cu capete mobile, caz in care se vor lua in considerare si timpii de positionare a capetelor pe diferiti "cilindri" (un cilindru este constituit de pistele cu acelasi numar, corespunzatoare unei positionari oarecare a ansamblului capetelor de scriere/citire).

Vom calcula timpul de acces in cazul "cozii de asteptare simple" (FIFO) si in cazul "timpului de acces minim" (Shortest Acces Time First - SATF). Discul se considera ca se roteste uniform cu o perioada de revolutie  $r$ , fiecare pista avind un numar intreg  $m$  de sectoare. Deci:

$$(6.17.) \quad s = \frac{r}{m}$$

Timpul de comutare intre scriere si citire si invers fiind neglijabil in comparatie cu intervalul dintre sectoarele consecutive, nu va fi luat in considerare.

Timpul de acces la un sector dat, luind ca origine un moment arbitrar, se descompune in doua variabile aleatoare independente:

- o variabila  $t$ , de distributie uniforma  $\frac{1}{s}$  in intervalul  $(0, s)$ , reprezentind timpul de acces la inceputul sectorului care urmeaza;
- o variabila aleatoare discreta  $u$  reprezentind timpul de acces la sectorul cerut cu incepere de la originea sectorului care urmeaza, aceasta variabila putind sa ia cu aceeasi probabilitate  $\frac{1}{m}$  valorile:  $0, 1, 2, \dots, (m-1)s$ .

Timpul mediu de acces la un sector oarecare este deci egal cu:

$$(6.18.) \quad a = \int_0^s t \frac{dt}{s} + \sum_{u=0}^{(m-1)s} u \frac{1}{m} = \frac{s}{2} + (m-1) \frac{s}{2} = \frac{r}{2}$$

Daca ne plasam la o limita de sector, timpul de acces mediu este:

$$(6.19) \quad a_1 = (m-1) \frac{s}{2}$$

Considerind un pachet de  $n$  cereri independente,

ne propunem sa calculam media timpului total de acces in functie de n. Cererile se ordoneaza in functie de politica urmata, cererile ulterioare sosite in timpul transferului fiind ignorate pina la tratarea pachetului initial.

6.3.1. *Cazul "cozii de asteptare simple"*  
(FIFO)

Cum s-a vazut, timpul mediu de acces a primei cereri este  $\frac{r}{2}$ ; a celor urmatoare este egala cu  $(m-1)\frac{s}{2}$ , deoarece dupa executarea unei cereri capetele se vor gasi la limita unui sector. Pentru n cereri media timpului total de acces este:

$$(6.20.) \quad A_F(n) = \frac{r}{2} + (n-1) \cdot \frac{m-1}{2} \cdot s$$

6.3.2. *Cazul "timpului de acces minim"*  
(SATF)

Pentru facilitarea calculului vom demonstra un rezultat preliminar: Fie un ansamblu de i variabile aleatoare independente  $t_1, t_2, \dots, t_i$ , de aceeaasi distributie  $p(t)$ . Consideram variabila aleatoare x, intotdeauna egala cu cea mai mica dintre variabilele  $v_i$ .

$$(6.21.) \quad x = \min \{ t_1, t_2, \dots, t_i \}$$

Avem  $P_r$  desemnind o probabilitate:

$$(6.22.) \quad P_r(x > u) = P_r(t_1 > u, t_2 > u, \dots, t_i > u)$$

$$(6.23.) \quad P_r(x > u) = (P_r(t > u))^i$$

$$(6.24.) \quad P(x > u) = (G(u))^i$$

unde:

$$(6.25.) \quad G(u) = P_r(t > u) = \int_0^{\infty} p(t) \cdot dt$$

De aici rezulta media minimului:

$$(6.26.) \quad \bar{x} = \int_0^{\infty} P_r(x > u) \cdot du = \int_0^{\infty} (G(u))^i \cdot du$$

Fie a timpului de acces la un sector dat, plasindu-se la limita unui sector. Functia de repartitie  $P_r(a < u)$  este o functie in scara (fig. 6.8.).

Inlocuind curba in trepte A cu dreapta B trecind prin mijlocul triunghiului ce compune curba A, vom obtine:

$$(6.27.) \quad F_r(a(u)) = \begin{cases} 0 & , u < 0 \\ \frac{1}{2 \cdot m} + \frac{u}{r} & , 0 \leq u \leq U \\ 1 & , u > U \end{cases}$$

unde:

$$U = (m - \frac{1}{2}) \cdot s.$$

Aplicind rezultatul precedent se obtine media timpului de acces la sectorul cel mai apropiat:

$$(6.28.) \quad a_s(n) = \int_0^U (1 - \frac{1}{2 \cdot m} - \frac{u}{r})^n \cdot du = \\ = \frac{r}{n+1} \cdot (1 - \frac{1}{2 \cdot m})^{n+1}$$

Daca ne plasam intr-un punct oarecare al discului si nu la limita unui sector, functia de repartitie este reprezentata de curba A de ecuatie  $\frac{u}{r}$ , pentru  $0 \leq u \leq r$ , iar timpul de acces la sectorul cel mai apropiat are ca medie:

$$(6.29.) \quad \int_0^r (1 - \frac{u}{r})^n \cdot du = \frac{r}{n+1}$$

In total, pentru n cereri, timpul de acces are media:

$$(6.30.) \quad A_s(n) \cong \frac{r}{n+1} + a_s(n-1) + a_s(n-2) + \dots + a_s(1)$$

Pe baza formulelor (6.20) si (6.30) se poate face o comparatie intre cele doua cazuri, in fig. 6.9. fiind date grafic rezultatele calculelor. Rezulta clar avantajul utilizarii metodei SATF.

#### 6.4. Concluzii.

In subcapitolele anterioare (6.2. si 6.3.) s-a recurs la metoda modelarii matematice, considerindu-se metoda cea mai ieftina si mai accesibila pentru faza de proiectare a anumitor parti ale cuplorului din cadrul SDM care influenteaza timpul de acces la informatie. In cazul SDM utilizate in sistemele biprocesor sau multiprocesor trebuiesc luate in considerare modul de gestionare a cererilor pe care le primeste subsistemul de la procesoarele pe care le serveste si modul de tratare a acestor cereri in contextul inregistrarii ce se fac pe UDM ale subsistemului. Modul de tratare a acestor probleme poate

duce la ameliorarea sau din contra la diminuarea calitatilor sistemului in ansamblu. Modelarea ne permite sa analizam diversele variante din punct de vedere teoretic, inainte de a trece la implementarea lor hardware. In prezent in cadrul S.O., driverul fiecarui subsistem indeplineste functiile enumerate mai sus. Implementarea unor functii software in hardware are avantajul obtinerii unor viteze de tratare mai ridicate, avindu-se in vedere faptul ca un intreg subprogram poate in anumite conditii sa fie inlocuit cu un singur semnal. Aceasta implementare a devenit posibila datorita progreselor tehnologice in domeniul circuitelor integrate si a scaderii preturilor de cost specifice ale hardware-ului. De asemenea tehnica microprogramarii a permis realizarea unor modele si prototipuri flexibile, care pot fi destul de usor utilizate pentru implementarea proiectelor verificate prin modelarea matematica.

La SDM lungimea unei lucrari poate fi considerata echivalenta cu lungimea lantului de date ce urmeaza a fi transferat ca urmare a unei singure cereri de transfer transmise de unul din procesoarele sistemului. Cum am vazut lucrarile se impart in cuante, lungimea unei cuante putind fi stabilita la lungimea unui sector, a unei piste sau a unui cilindru. Modul de inregistrare al datelor la UDM, ne permite calculul inca din momentul adresarii cererii, a felului in care se afla plasata informatia, adica daca va fi necesara o schimbare a pozitiei ansamblului de capete sau numai un schimb de informatii cu acelasi cap de scriere-citire, sau trecerea pe o alta pista a aceluiasi cilindru, deci numai o schimbare a capului de scriere-citire, fara nici o interventie mecanica care necesita citeva milisekunde. De aceea este avantajos sa se aplice metoda de reciclaj cu mai multe siruri de asteptare, cu lungimea cuantei variabila sau cu tratarea in primul sir a unui numar variabil de cuante (exemplu 6.2.2.). Lungimea instantanee a cuantei trebuie aleasa in asa fel incit sa nu cuprinda pe parcursul ei nici o modificare mecanica a pozitiei ansamblului de capete. In timpul operatiunii de scriere, daca gestiunea ocuparii capacitatii UDM din SDM este tinuta de cuplorul pentru UDM, se poate face o optimizare a plasarii informatiei dupa un algoritm care sa evite pe cit posibil intreruperea fluxului de date pe parcursul unei cereri datorate necesitatii schimbarii pozitiei ansamblului de capete. La SDM care utilizeaza mai mult de o UDM aceasta optimizare poate duce



la evitarea totala a timpilor pierduti datorita acestei schimbări de pozitie. In timp ce o UDM face o pozitionare, o alta va face transfer de informatii.

La UDM fabricata la R.C.D. cu capacitatea de 58 Moct. timpul de pozitionare mediu este 35 ms, iar timpul de pozitionare pista-pista de 7 ms. Viteza de transfer a informatiei este de 340.000 de oct./s., deci in 7 ms s-ar putea transfera 2.380 oct., ceea ce corespunde lungimii a aproximativ cinci sectoare de cite 512 oct. Daca se mai adauga si faptul ca posibilitatea de a ajunge la informatie este de  $\frac{r}{2}$  (conform 6.18.) si cum numarul total de sectoare pe o pista la aceste UDM este 12, rezulta o pierdere totala de timp egala cu timpul in care s-ar putea transfera aproximativ 11 sectoare. Prin optimizarea operatiilor de scriere si implicit de citire, aceste neajunsuri pot fi inlaturate, SDM lucrind efectiv cu viteza maxima de transfer a unei UDM. Daca SDM este prevazut cu o memorie tampon adecvata, viteza de transfer poate fi marita, factor limitativ devenind viteza admisa de transfer a informatiilor pe magistrala la care este cuplat SDM si procesoarele pe care le serveste.

Calcululele de optimizare a transferurilor se pot face cu algoritmi care analizeaza pachete de cereri venite pina la un moment limita (vezi 6.3.), deci se poate adopta o metoda dinamica unde sirurile de asteptare sint actualizate permanent. Metoda SAIF este cum s-a vazut mai eficace si implicit poate duce la eliminarea timpilor morti datorati pozitionarilor ansamblului de capete.

## CAP.7. TRATAREA FENOMENULUI DE INTERBLOCAJ

### 7.1. Generalitati .

SDM utilizat in configuratiile biprocesor sau multiprocesor, ridica o serie de probleme noi, dintre care unele nu apar la configuratiile monoprocesor decit la nivelul S.O. O serie de functiuni si metode software care in prezent fac parte din sistemul S.O. pot fi rezolvate la nivelul SDM prin mijloace hardware. Acest transfer de sarcini aduce insa si o transferare a unor deficiente care limitau performantele de ansamblu ale configuratiei. Aceste limitari, precum si gasirea unor metode de inlaturare a lor

trebuie analizate in contextul unor configuratii biprocesor si multiprocesor, care utilizeaza un singur SDM echipat cu doua sau mai multe UDM. Se cauta ridicarea la maximum a gradului de disponibilitate a SDM.

SDM are sarcina alocarii resurselor (UDM) catre diverse procese utilizator, apartinand procesoarelor din configuratie. In procesul de alocare a resurselor un fenomen foarte periculos care poate sa apara este "interblocajul". Cererile de resurse ale diverselor procesoare pot fi satisfacute intr-o ordine oarecare, insa se poate intimpla ca doua sau mai multe sa se blocheze reciproc, deblocajul fiind foarte dificil uneori chiar imposibil. De aceea este important ca interblocajul sa fie evitat.

7.2. Starile unui sistem.

Un sistem este constituit:

- dintr-un ansamblu finit de procese secventiale care pot fi executate concurent:

$$(7.1.) \quad P = \{p_1, p_2, \dots, p_n\},$$

- dintr-un ansamblu de clase de resurse cu cite un punct de acces:

$$(7.2.) \quad E = \{R_1, R_2, \dots, R_m\},$$

Starea initiala a sistemului poate fi descrisa de un vector X care sa dea numarul total de resurse care exista in fiecare clasa:

$$(7.3.) \quad X = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

Vectorul X ramine constant in timpul observarii sistemului. Presupunem de asemenea ca toate procesele intr-un timp finit vor elibera toate resursele pe care le-au achizitionat.

La un moment dat t, starea sistemului este definita de matricea resurselor alocate proceselor A(t) si de matricea resurselor cerute de procese C(t):

$$(7.4.) \quad A(t) = \begin{pmatrix} a_{11}(t) & \dots & a_{1n}(t) \\ \vdots & & \vdots \\ a_{m1}(t) & \dots & a_{mn}(t) \end{pmatrix} = (A_1(t), \dots, A_n(t))$$

unde  $a_{ij}(t)$  este numarul de resurse din clasa  $R_i$  alocate procesului  $p_j$ .

$$(7.5.) \quad C(t) = \begin{pmatrix} c_{11}(t) & \dots & c_{1n}(t) \\ \vdots & & \vdots \\ c_{m1}(t) & \dots & c_{mn}(t) \end{pmatrix} = (C_1(t), \dots, C_n(t))$$

- unde  $c_{ij}(t)$  este numărul de resurse din clasa  $R_j$  cerute de procesul  $p_j$ .

Alocarea resurselor, adică schimbarea stării sistemului se poate face numai prin următoarele operații:

- cerere: un proces  $p_j$  face o cerere de resurse reprezentată printr-un vector  $N$ :

$$(7.6.) \quad C_j(t) := C_j(t) + N$$

- achiziție: resursele reprezentate printr-un vector  $M$  sunt alocate procesului  $p_j$ :

$$(7.7.) \quad A_j(t) := A_j(t) + M$$

- liberare: un proces  $p_j$  liberează resursele reprezentate printr-un vector  $L$ :

$$(7.8.) \quad \begin{aligned} A_j(t) &:= A_j(t) - L \\ C_j(t) &:= C_j(t) - L \end{aligned}$$

Observatii:

- matricea  $C$  nu se micșorează decât în momentul liberării resurselor;

- o cerere nu este întotdeauna urmată de o alocare dar poate fi pusă în așteptare, lucru care justifică folosirea celor două matrice  $C$  și  $A$ .

În cele ce urmează se utilizează următoarele notații:

Fie  $U$  și  $V$  vectorii cu  $m$  elemente. Prin definiție avem:

$$(7.9.) \quad \begin{cases} U \leq V & (\Leftrightarrow) & U_i \leq V_i & \forall i \in [1, m] \\ U < V & (\Leftrightarrow) & (U \leq V) & \wedge (\exists i \text{ astfel ca } U_i < V_i) \end{cases}$$

Fie matricea  $M$  și  $N$  cu  $m \times n$  elemente:

$$(7.10.) \quad \begin{cases} M \leq N & (\Leftrightarrow) & M_i \leq N_i & \forall i \in [1, n] \\ M < N & (\Leftrightarrow) & (M \leq N) & \wedge (\exists i \text{ astfel ca } M_i < N_i) \end{cases}$$

Următoarele condiții trebuie să fie verificate:

- un proces dat nu poate cere mai multe resurse decât există în sistem:

$$(7.11.) \quad C_j(t) \leq X \quad \forall i \in [1, n]$$

- un proces nu poate avea la un moment dat alocate mai

multe resurse decit a cerut:

$$(7.12.) \quad A_i(t) \leq C_i(t) \quad \forall i \in [1, n]$$

- suma achizitiilor tuturor proceselor la un moment dat nu poate depasi totalitatea resurselor sistemului:

$$(7.13.) \quad \sum_{i=1}^n A_i(t) \leq X$$

Starea unui sistem la un moment dat  $t$  este o "stare realizabila" daca si numai daca  $A(t)$ ,  $C(t)$  si  $X$  verifica relatiile (7.11.), (7.12.) si (7.13.).

Vectorii  $C_i(t)$  si  $A_i(t)$  definind starea de alocare a resurselor catre procesul  $p_i$ , acesta va fi blocat atita timp cit  $C_i(t) > A_i(t)$ . Definim prin  $R(t)$  vectorul care arata numarul de resurse disponibile in momentul  $t$  pentru fiecare clasa:

$$(7.14.) \quad R(t) = X - \sum_{i=1}^n A_i(t)$$

Astfel relatia (7.13.) se poate scrie:

$$(7.15.) \quad R(t) \geq 0$$

### 7.3. Interblocajul.

Putem caracteriza gradul de solutionare a unui proces la un moment dat  $t$ , prin numarul  $N(t)$  de instructiuni executate din momentul initial. Fie doua procese  $a$  si  $b$  cu gradul de solutionare  $N_1$  respectiv  $N_2$  (fig. 7.1.). Prin eliminarea timpului se poate reprezenta curba de solutionare relativa a celor doua procese definite de ecuatie:

$$(7.16.) \quad F(N_1, N_2) = 0$$

Blocajul unuia din procese este pus in evidenta de segmentele paralele cu axele, portiunea in care poate apare interblocajul este suprafata patrata "P" in care aceeaasi resursa  $R$  cu un singur punct de acces este ceruta simultan de ambele procese. Aceasta suprafata nu poate fi traversata de curba  $F$ .

Consideram cazul in care cele doua procese  $a$  si  $b$  cer accesul exclusiv la doua resurse  $R_1$  si  $R_2$  dupa urmatoarea schema:

- a cere  $R_1$  si apoi  $R_2$
- b cere  $R_2$  si apoi  $R_1$

In fig. 7.2.a. se observa ca interblocajul este inevitabil daca se patrunde in patrutul ABCD, cu toate ca nu apare

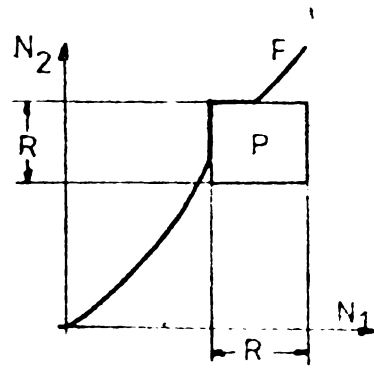


Fig. 7.1. Utilizarea a doua resurse de catre doua procese.

\*2.

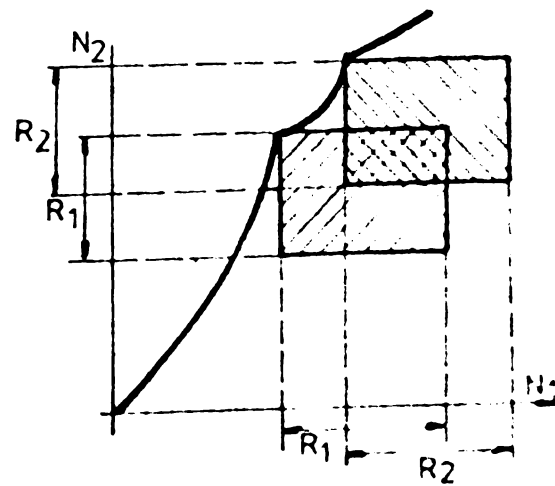
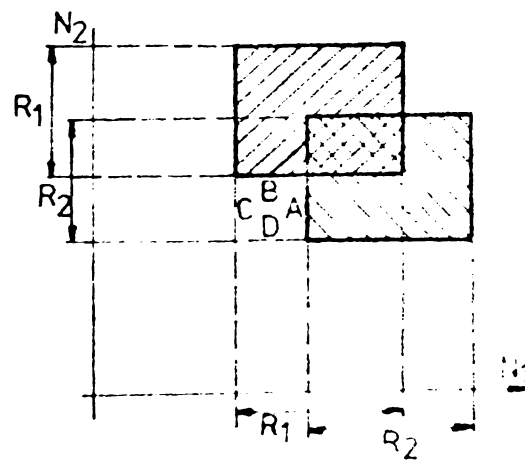


Fig. 7.2. Schema utilizarii a doua resurse.  
 a. Cu interblocaj                      b. Fara interblocaj

imediat după traversarea laturei C sau D. Pentru ca interblocajul să fie evitat resursele trebuie să fie cerute în alte ordine și anume  $R_1, R_2$  (fig.7.2.b.) sau  $R_2, R_1$  pentru ambele procese. S-a demonstrat grafic că interblocajul poate fi evitat dacă procesele cer resursele în aceeași ordine. În acest caz pot apărea numai porțiuni de blocare a unuia din procese.

Interblocajul trebuie să fie evitat la momentul  $t$ , dacă există începând cu acest moment o serie de stări realizabile ale sistemului, astfel ca toate procesele obțin resursele pe care le-au cerut și sunt executate pînă la sfîrșitul lor. Vom demonstra că dacă găsim un astfel de șir de stări, acesta echivalează cu aranjarea proceselor într-o ordine, astfel ca cererile lor pot fi satisfăcute dacă sunt executate în această ordine.

Fie  $S$  un astfel de șir de procese, notînd cu  $S(i)$  rangul  $P_i$  al procesului în șirul  $S$  și presupunînd că la momentul  $t$  considerat fiecare proces a cerut toate resursele necesare execuției proprii. Cererea nesatisfăcută a primului proces din șir trebuie să fie inferioară resurselor disponibile la momentul  $t$ :

$$(7.17.) \quad S(i_1) = 1 \quad C_{i_1}(t) - A_{i_1}(t) \leq R(t)$$

Acest proces se execută, apoi liberează resursele sale.

Al doilea proces se poate executa acum cu condiția că :

$$(7.18.) \quad S(i_2) = 2 \quad C_{i_2}(t) - A_{i_2}(t) \leq R(t) + A_{i_1}(t)$$

și tot așa pentru celelalte procese. Șirul  $S$  verifică deci relația:

$$(7.19.) \quad \forall p_i \in P \quad C_{i_1}(t) - A_{i_1}(t) \leq R(t) + \sum_{S(j) < S(i)} A_j(t)$$

Acum putem introduce următoarele definiții:

- un șir  $S$  se consideră "viabil" dacă și numai dacă verifică relația :

$$(7.20.) \quad C_{i_1}(t) - A_{i_1}(t) \leq R(t) + \sum_{S(j) < S(i)} A_j(t), \quad \forall p_i \in S$$

- un șir  $S$  este considerat "complet" relativ la  $P$  dacă și numai dacă conține toate procesele aparținînd lui  $P$ ;

- o stare realizabilă a sistemului este considerată viabilă dacă și numai dacă există în această stare un șir viabil complet de procese.

Presupunem ca la momentul  $t_0$ , procesele au cerut toate resursele necesare executiei lor.

*Teorema 7.1.* Daca starea sistemului este viabila, atunci exista un sir de stari realizabile ale sistemului, astfel ca toate procesele vor obtine resursele pe care le-au cerut, reciproca fiind de asemenea valabila.

Fie  $S$  un sir viabil care poate fi constituit la momentul initial  $t_0$ . Consideram sirul de  $n$  stari:

$$(7.21.) \quad E_k = (A(t_k), C(t_k), X)$$

definit dupa cum urmeaza:

$$(7.22.) \quad \left\{ \begin{array}{ll} A_{i_k}(t_k) = 0 & \forall i \text{ astfel ca } S(i) < k \\ A_{i_k}(t_k) = C_{i_0}(t_0) & \forall i \text{ astfel ca } S(i) = k \\ A_{i_k}(t_k) = A_{i_0}(t_0) & \forall i \text{ astfel ca } S(i) > k \\ C_{i_k}(t_k) = 0 & \forall i \text{ astfel ca } S(i) < k \\ C_{i_k}(t_k) = C_{i_0}(t_0) & \forall i \text{ astfel ca } S(i) > k \end{array} \right.$$

S-a aratat ca aceste stari sint stari realizabile ale sistemului. Starea initiala  $E_0$  a sistemului este viabila si verifica relatiile (7.11.), (7.12.), (7.13.) si (7.20):

$$(7.23.) \quad \left\{ \begin{array}{ll} C_{i_0}(t_0) \leq X & \forall i \in [1, n] \\ A_{i_0}(t_0) \leq C_{i_0}(t_0) & \forall i \in [1, n] \\ \sum_{i=1}^n A_{i_0}(t_0) \leq X \\ C_{i_0}(t_0) - A_{i_0}(t_0) \leq R(t_0) + \sum_{S(j) < S(i)} A_{j_0}(t_0), \forall i \in [1, n] \end{array} \right.$$

Conditiiile (7.11.) si (7.12.) sint verificate implicit prin definirea sirului de stari  $E_k$ . Sa aratam ca si conditia (7.13.) este verificata.

Fie:

$$(7.24.) \quad Y = \sum_{i=1}^n A_{i_k}(t_k)$$

din definitia lui  $A_k$  avem:

$$(7.25.) \quad Y = \sum_{S(i) < k} A_{i_k}(t_k) + \sum_{S(i) = k} A_{i_k}(t_k) + A_{i_k}$$

unde  $i_k$  este astfel ca  $S(i_k) = k$ , sau :

$$(7.26.) \quad Y = 0 + \sum_{S(i) > k} A_i(t_k) + C_{i_k}(t_0)$$

Inlocuim  $C_{i_k}(t_0)$  prin majorantul sau (relatia 7.20.),

obtinem:

$$(7.27.) \quad Y \leq R(t_0) + \sum_{S(j) \leq k} A_j(t_0) + \sum_{S(j) > k} A_j(t_0)$$

sau:

$$(7.28.) \quad Y \leq X$$

Deci stările  $E_k$  sînt stări realizabile ale sistemului,

fiind usor de verificat ca acestea sînt stări viabile.

Prin teorema 7.1. se poate verifica foarte simplu daca un sistem este sau nu in interblocaj, prin examinarea stărilor sistemului. Nu se garanteaza insa ca stările urmatoare vor fi viabile.

#### 7.4. Remediile interblocajului.

Exista doua posibilitati de remediere a interblocajului:

- detectarea si deblocarea, faza de detectare avind scopul punerii in evidenta a momentului aparitiei interblocajului, in timp ce faza de deblocare a sistemului folosind mai multe metode, merge de la distrugerea ultimului proces care a provocat interblocajul (daca in faza precedenta se utilizeaza o metoda de detectie permanenta si pina la distrugerea completa a tuturor proceselor care au intrat in interblocaj.

- prevenirea interblocajului.

Deoarece prevenirea interblocajului este solutia care se preteaza a fi aplicata pentru SDM, in cele ce urmeaza vom analiza aceasta posibilitate, in mecanismul de alocare a resurselor introducind reguli care elimina toate situatiile care pot conduce la interblocaj, considerind starea initiala a sistemului ca fiind o stare viabila.

Inainte de analiza metodele de prevenire a interblocajului, este necesar sa se puna la punct metode de detectie a interblocajului. Metoda generala de detectie consta in construirea unui sir viabil de procese si daca acest sir este complet nu va apare interblocaj, daca nu, procesele care nu apartin acestui sir vor fi interblocate. Teorema 7.2. faciliteaza cautarea unui sir viabil complet.

*Teorema 7.2.* Daca starea sistemului este viabila toate sirurile viabile incomplete S pot fi completate intr-un



sir viabil complet.

Starea sistemului fiind viabila, exista un sir viabil complet Q.

$$(7.29.) \quad C_k \leq R + \sum_{Q(1) \leq Q(k)} A_1 \quad \forall p_k \in Q$$

Fie S un sir viabil :

$$(7.30.) \quad C_k \leq R + \sum_{S(1) \leq S(k)} A_1 \quad \forall p_k \in S$$

Definim un sir  $S_1$  astfel ca:

$$(7.31.) \quad S_1(k) = S(k) \quad \forall p_k \in S$$

$$(7.32.) \quad Q(1) \leq Q(k) \implies S_1(1) \leq S_1(k), \quad \forall p_k, p_1 \in Q \cap S$$

$$(7.33.) \quad \sum_{Q(1) \leq Q(k)} A_1 \leq \sum_{S_1(1) \leq S_1(k)} A_1, \quad \forall p_k \in Q \cap S$$

deoarece orice  $p_1$  care precede pe  $p_k$  in sirul Q il va precede de asemenea si in sirul  $S_1$ , avindu-se in vedere definirea lui  $S_1$ .

Relatiile (7.29.) si (7.33.) implica:

$$(7.34.) \quad C_k \leq R + \sum_{S_1(1) \leq S_1(k)} A_1, \quad \forall p_k \in Q \cap S$$

Acum relatiile (7.30.) si (7.34.) implica:

$$(7.35.) \quad C_k \leq R + \sum_{S_1(1) \leq S_1(k)} A_1, \quad \forall p_k \in S_1$$

Deci sirul  $S_1$  este un sir viabil complet.

*Teorema 7.3.* Executarea unei serii de alocare de catre un proces face sa se treaca dintr-o stare viabila in alta stare viabila daca exista un sir viabil care sa contina acest proces.

Fie A si B doua siruri si vom nota AB sirul obtinut prin legarea celor doua siruri in ordinea A,B. Fie  $E_0$  starea initiala la momentul  $t_0$  si  $p_i$  un proces neblocaat in aceasta stare.

Avem:

$$(7.36.) \quad C_i(t_0) = A_i(t_0)$$

Fie  $E_1$  starea finala obtinuta in momentul  $t_1$  prin

executia unei cereri a lui  $p_i$ .

$$(7.37.) \quad \begin{cases} C_j(t_1) = C_j(t_0) & \forall j = i \\ A_j(t_1) = A_j(t_0) & \forall j \\ C_i(t_1) > C_i(t_0) \end{cases}$$

Vom avea:

$$(7.38.) \quad R(t_1) = R(t_0)$$

deoarece nu a avut loc nici o achizitie.

Fie  $S = S_1\{p_i\}$  un sir viabil in starea  $E_1$ .

$$(7.39.) \quad C_i(t_1) - A_i(t_1) \leq R(t_1) + \sum_{S(j) < S(i)} A_j(t_1), \\ \forall p_i \in S$$

Deci  $S$  este un sir viabil al starii  $E_0$ . Dupa teorema 7.2. el poate fi prelungit intr-un sir viabil complet  $S_2$  pentru starea  $E_0$ . Sa aratam ca  $S_2$  este un sir viabil al starii  $E_1$ .

Punind  $S_2 = S_1 S_3$

$$(7.40.) \quad C_k(t_0) - A_k(t_0) \leq R(t_0) + \sum_{S_2(j) < S_2(k)} A_j(t_0), \\ \forall p_k \in S_3$$

Proprietatile starii  $E_1$  implica:

$$(7.41.) \quad C_k(t_1) - A_k(t_1) \leq R(t_1) + \sum_{S_2(j) < S_2(k)} A_j(t_1), \\ \forall p_k \in S_3$$

$S$  fiind un sir viabil din starea  $E_1$ , relatia precedenta implica faptul ca  $S_2$  este un sir complet din starea  $E_1$

Pentru prevenirea interblocajului exista doua metode:

- o metoda statica, care impune restrictii la cereri si la achizitionarea resurselor in asa fel incit sa nu apara interblocajul;

- o metoda dinamica, care poate recunoaste daca o cerere poate duce la un interblocaj si care amina alocarea pina in momentul in care riscul este eliminat.

#### 7.4.1. Prevenirea statica a interblocajului.

##### 7.4.1.1. Metoda cererii globale

Este metoda cea mai simpla conform careia toate pro-

cesele trebuie sa ceara si sa achizitioneze global resursele. In aceste conditii exista intotdeauna un sir viabil complet:  $S = S_1 S_2$  cu

$$(7.42.) \quad \begin{cases} \forall p_i \in S_1 & C_i - A_i = 0 \\ \forall p_i \in S_2 & A_i = 0 \end{cases}$$

Orice proces nou venit este plasat in sirul  $S_2$  daca nu i se pot satisface de la inceput toate cererile, in caz contrar este plasat in sirul  $S_1$ . Liberarea de resurse de catre un proces este urmata de o cautare a proceselor din sirul  $S_2$  carora li se pot satisface integral cererile, acestea fiind trecute in sirul  $S_1$ , noul sir fiind intotdeauna viabil. Un dezavantaj al metodei este ca anumite procese pot "tezauriza" anumite resurse pe care nu le mai folosesc momentan.

#### 7.4.1.2. Metoda claselor ordonate

Resursele se grupeaza in clase, ansamblul acestor clase fiind ordonat. Proceselor li se impune:

- sa ceara global resursele care sunt necesare din fiecare clasa;
- sa ceara resursele apartinand diferitelor clase in ordinea claselor.

Fie  $p$  numarul claselor de resurse  $D$ . Exista intotdeauna un sir complet  $S = S_p \dots S_i \dots S_0$ .

Pentru orice  $p_j \in S_k$ :

$$(7.43.) \quad \begin{cases} c_{ij} - a_{ij} = 0, & \forall i, l \text{ astfel ca } R_l \in D_1 \text{ si } l \leq k \\ a_{ij} = 0, & \forall i, l \text{ astfel ca } R_l \in D_1 \text{ si } l > k \end{cases}$$

Sirurile  $S_i$  sint siruri viabile pentru orice  $i$ , iar

sirul  $S$  este un sir viabil complet. Introducerea unui nou proces in sistem nu va pune probleme, deoarece poate fi plasat intotdeauna in sirul  $S_0$ . In momentul eliberarii de resurse, procesul care s-a eliberat poate fi deplasat dintr-un sir  $S_i$  in altul. Sirul  $S$  este parcurs pentru a incerca satisfacerea partiala sau totala a cererilor altor procese, respectind regula precedenta.

Aceasta metoda duce la ameliorarea repartitiei res-

surselor, resursele mai costisitoare fiind plasate in clasele superioare, astfel diminuindu-se timpul lor de inactivitate.

7.4.2. *Prevenirea dinamica a interblocajului.*

Metoda de prevenire dinamica impune tuturor proceselor sa-si declare toate cerintele, inainte de inceperea executiei. Aceste declaratii le numim "anunturi". Intr-un sistem cu anunturi, starea e realizabila daca se verifica relatiile (7.11.), (7.12.) si (7.13.). Fie  $D$  matricea de anunturi unde  $d_{ij}$  reprezinta numarul maxim de resurse din clasa  $R_i$  pe care le poate utiliza procesul  $p_j$ . Unei stari realizabile  $i$  se impune sa verifice noile relatii:

$$(7.44.) \quad D_i \leq X \quad \forall i$$

$$(7.45.) \quad A_i(t) \leq D_i \quad \forall i, t$$

$$(7.46.) \quad C_i(t) \leq D_i \quad \forall i, t$$

Aceasta metoda presupune ca un proces poate sa pretinda toate resursele anuntate. Alocarea nu se face decit daca sistemul va ramine intr-o stare viabila.

Un sir  $S$  de procese este denumit "fiabil" daca ele verifica relatia:

$$(7.47.) \quad D_i - A_i(t) \leq R(t) + \sum_{S(j) < S(i)} A_j(t), \quad \forall p_i \in S$$

Starea sistemului este fiabila daca exista o suita fiabila completa pentru aceasta stare. Relatia (7.46.) arata implicit ca, conditia (7.47.) este mai puternica decit conditia (7.20.). Deci:

*Teorema 7.4.* Orice sir fiabil este un sir viabil al sistemului.

La fel cum relatiile (7.11.), (7.12.), (7.13.) si (7.20.) au antrenat teorema 7.2, relatiile (7.11.), (7.44.), (7.45.) si (7.47.) antreneaza:

*Teorema 7.5.* Daca starea sistemelor este fiabila, toate sirurile fiabile pot fi prelungite intr-un sir fiabil complet.

Pentru schimbarile de stare se poate aplica urmatoarea teorema:

*Teorema 7.6.* O alocare de resurse la un proces  $p_k$  face ca sistemul sa treaca dintr-o stare fiabila intr-o noua stare fiabila daca exista in aceasta noua stare un sir

fiabil care sa contina procesul  $p_k$ .

Fie  $E_0$  starea sistemului la momentul  $t_0$  si  $E_1$  starea sistemului la momentul  $t_1$ , obtinuta prin alocarea de resurse procesorului  $p_k$ . Alocarea de resurse acestui

proces se traduce prin relatiile:

$$(7.48.) \quad A_{10}(t) = A_{11}(t) \quad \forall l = k$$

$$(7.49.) \quad A_{k1}(t) > A_{k0}(t)$$

$$(7.50.) \quad R(t_1) + A_{11}(t_1) = R(t_0) + A_{k0}(t_0)$$

Fie  $S$  un sir fiabil la momentul  $t_1$  si care contine procesul  $p_k$ .

$$(7.51.) \quad S = S_1\{p_k\}$$

$$(7.52.) \quad D_i - A_{i1}(t_1) \leq R(t_1) + \sum_{S(1) \leq S(i)} A_{11}(t_1), \quad \forall p_i \in S_1$$

$$(7.53.) \quad D_k - A_{k1}(t_1) \leq R(t_1) + \sum_{S(1) \leq S(i)} A_{11}(t_1)$$

Sa aratam ca este de asemenea fiabil la momentul  $t_0$  conform relatiilor (7.48.), (7.49.) si (7.50.):

$$(7.54.) \quad D_i - A_{i0}(t_0) \leq R(t_0) + \sum_{S(1) \leq S(i)} A_{10}(t_0), \quad \forall p_i \in S_1$$

Relatia (7.50.) antreneaza:

$$(7.55.) \quad D_k - A_{k1}(t_1) - R(t_1) = D_k - A_{k0}(t_0) - R(t_0)$$

de unde:

$$(7.56.) \quad D_k - A_{k0}(t_0) \leq R(t_0) + \sum_{S(1) \leq S(k)} A_{10}(t_0)$$

Deci in momentul  $t_0$  sirul  $S$  este fiabil si poate fi prelungit intr-un sir fiabil complet  $S_2$ .

$$(7.57.) \quad D_i - A_{i0}(t_0) \leq R(t_0) + \sum_{S_2(1) \leq S_2(i)} A_{10}(t_0), \quad \forall p_i \in S_2$$

Putem arata ca  $S_2$  este de asemenea fiabil in momentul  $t_1$ .

$$\text{Punind:} \quad S_2 = S_1 S_3$$

$$(7.58.) \quad D_i - A_{i0}(t_0) \leq R(t_0) + \sum_{S_2(1) \leq S_2(i)} A_{10}(t_0), \quad \forall p_i \in S_3$$

Relatiile (7.48.) si (7.50.) antreneaza:

$$(7.59.) \quad D_i - A_i(t_1) \leq R(t_0) + A_k(t_0) + \sum_{\substack{S_2(1) \leq S_2(i) \\ \text{si } 1=k}} A_1(t_1) \\ \leq R(t_1) + \sum_{S_2(1) \leq S_2(i)} A_1(t_1), \quad \forall p_i \in S_3$$

Sirul  $S$  fiind fiabil in momentul  $t_1$ , sirul  $S_2$  este un sir fiabil complet, in momentul  $t_1$ .

*Teorema 7.7.* Daca nici un proces nu-si elibereaza resursele inainte de a primi toate resursele cerute (anuntate), exista cel putin o modalitate de alocare a resurselor fara a aparea interblocajul, daca sau numai daca starea sistemului ramine fiabila. Remarcam pentru  $D = C$ , echivalenta intre conceptele de sir fiabil si de sir viabil, precum si de stare fiabila si stare viabila. Considerind un sir fiabil complet, daca procesele primesc totalitatea anunturilor lor ( $D = C$ ) conditiile teoremei 7.1. sint verificate. Exista deci un sir de stari fiabile astfel ca procesele isi obtin resursele lor. Reciproca este de asemenea valabila.

O alocare nu poate avea loc decit daca rezultatul este o stare fiabila. Practic starea initiala a sistemului este intotdeauna fiabila. Deci inainte de orice alocare sa se verifice ca noua stare obtinuta este fiabila.

#### 7.5. Concluzii .

Transformarea SDM intr-un subsistem independent, inzestrat cu putere de calcul proprie permite transferarea unor sarcini de la nivelul SO la nivelul subsistemului. In configuratia multiprocesor dotate cu SDM sau SII comun, in cadrul operatiei de gestionare a repartitiei resurselor un loc important trebuie sa fie destinat evitarii aparitiei fenomenului de interblocaj. Daca in cazul SO se putea citeodata merge pina la intrarea in interblocaj si la rezolvarea problemei prin anulara unui numar de procese care au condus la aparitia interblocajului, in cazul SDM evitarea interblocajului este obligatorie pentru a asigura functionarea continua a intregii configuratii. Cum s-a aratat fenomenul de interblocaj poate fi evitat. Deci pentru SDM se va lucra dupa metodele de prevenire a interblocajului, metoda de prevenire dinamica fiind cea mai potrivita. Fiecare proces va trebui sa-si

declara cererile la inceputul executiei, alocarea facindu-se cu luarea in considerare a situatiei de moment, interblocajul fiind evitat.

## CAP.8. MECANISME DE SINCRONIZARE

Preluarea de catre SDM a unor sarcini de la SO, duce la transferarea acestor probleme specifice la nivelul SDM. In cadrul SDM trebuie sa se gestioneze cererile transmise de procesoarele sistemului de calcul. In acest fel, problema sincronizarii acestor cereri va trebui rezolvata de SDM. De asemenea apare si problema sincronizarii microprogramelor SDM care indeplinesc diversele functiuni: de scriere, de citire, de comparare, de control de paritate, de optimizare a acceselor, etc.

### 8.1. Principiul de realizare a operatiilor de sincronizare. Semafoare.

Operatiile de sincronizare se pot realiza cu ajutorul "semafoarelor". Un semafor notat cu  $s$  este constituit dintr-o variabila intregă  $e(s)$  si un sir de asteptare  $f(s)$ . Variabila  $e(s)$  poate avea valori intregi pozitive, negative sau valoarea zero si poate varia cu incrementul  $+1$ . Acestea formeaza "valoarea" semaforului. Un semafor  $s$  se creeaza printr-o declaratie initiala care trebuie sa specifice valoarea initiala  $e_0(s)$  a lui  $e(s)$ , acestea fiind obligatoriu diferita de o valoare negativa. La crearea unui semafor sirul sau de asteptare  $f(s)$  este initial vid.

Asupra unui semafor  $s$  se poate actiona prin intermediul a doua primitive  $P$  si  $V$ , care dau operatii indivizibile.

```
- tip P    P(s):  inceput
                e(s) := e(s) - 1;
                daca      e(s) < 0      atunci
                    inceput
                comentariu : presupunem ca
                               primitiva este
```

```
                                executata de pro-
                                cesul a;
                                stare (a) := blocat ;
                                se pune procesorul a in sirul
                                de asteptare f(s)
                                sfirsit
                                daca e(s) > 0 atunci se continua
                                procesul in secventa
                                sfirsit
- tip V      V(s):  inceput
                  e(s) := e(s) + 1
                  daca      e(s) < 0      atunci
                  inceput
                  se scoate un proces din sirul
                  de asteptare f(s)
                  comentariu : presupunem ca primiti-
                               va este executata de
                               procesul b;
                  stare (b):  activ
                  sfirsit
                  daca e(s) > 0 atunci se continua
                               procesul in secventa
                  sfirsit
```

### 8.1.1. Proprietatile semafoarelor.

Definirea primitivelor P si V are urmatoarele consecinte:

- a. Un semafor dupa un numar de operatii P poate deveni negativ, insa valoarea sa initiala nu poate fi niciodata negativa.
- b. Fie  $np(s)$  - numarul de instructiuni P executate asupra semaforului s  
 $nv(s)$  - numarul de instructiuni V executate asupra semaforului s  
 $e_0(s)$  - valoarea initiala a semaforului s.

Din definitia primitivelor P si V rezulta:

$$(8.1.) \quad e(s) = e_0(s) - np(s) + nv(s)$$

- c. Fie  $nf(s)$  numarul de procese care au "trecut" de primitiva P(s), adica acelea care nu au fost blocate de acestea, sau daca au fost initial blocate ulterior au fost deblocate. Oricind vom avea:

$$(8.2.) \quad nf(s) \leq np(s)$$



Efectele primitivelor asupra lui  $nf(s)$  sint urmatoarele:

(8.3.)  $P(s): np(s) := np(s) + 1;$   
*daca*  $np(s) < e_0(s) + nv(s)$  *atunci*  
 $nf(s) := nf(s) + 1$   
*comentariu*: cazul in care  $e(s) >= 0$

(8.4.)  $V(s): nv(s) := nv(s) + 1;$   
*daca*  $np(s) > e_0(s) + nv(s)$  *atunci*  
 $nf(s) := nf(s) + 1;$   
*comentariu*: cazul in care  $e(s) <= 0$

*Teorema 8.1.* Executarea primitivelor  $P$  si  $V$  lasa neschimbata relatia:

(8.5.)  $nf(s) = \min [np(s), e_0(s) + nv(s)]$

Presupunem relatia ca verificata si vom examina efectul executiei sale. Relatia poate lua doua forme in functie de valorile relative ale  $np(s)$  si  $e_0(s) + nv(s)$ .

a. Executarea lui  $P(s)$ . In tabelul urmator se simplifica notarea suprimindu-se numele semaforului  $s$ .

Forma initiala a relatiei	Relatia dupa executarea a $np := np + 1$	Efectul asupra termenului $nf$	Relatia dupa executarea lui $P$
$np < e_0 + nv$	$np <= e_0 + nv$	$nf := nf + 1$	$nf = np$ $nf <= e_0 + nv$
$np >= e_0 + nv$	$np > e_0 + nv$ $nf <= np$	nu are efect	$nf = e_0 + nv$ $nf < np$

Dupa executia lui  $P(s)$  relatia (8.5.) se verifica in orice caz.

b. Executarea lui  $V(s)$

Forma initiala a relatiei	Relatia dupa executarea a	Efectul asupra termenului	Relatia dupa executarea lui V
	$nv := nv + 1$	$nf$	
$np > e_0 + nv$	$nf = e_0 + nv$	$np > e_0 + nv$	$nf := nf + 1$
	$nf < np$		$nf = e_0 + nv$
			$nf < np$
$np <= e_0 + nv$	$nf = np$	$np < e_0 + nv$	nu are efect
	$nf <= e_0 + nv$		$nf = np$
			$nf < e_0 + nv$

Relatia (8.5.) este si acum verificata in toate cazurile.

c. Pentru valorile initiale relatia (8.5.) ramine de asemenea adevarata:

$$(8.6.) \quad np = nv = nf = 0 \\ e_0 >= 0$$

Relatia (8.5.) intuitiv poate fi explicata prin asimilarea semaforului cu o bariera, o operatie P reprezinta o cerere de trecere,  $e_0 + nv$  reprezinta numarul total de autorizari date pina in momentul prezent; relatia aratind ca numarul efectiv de treceri este egal cu cel mai mic dintre cele doua marimi (cererile sau autorizarile).

d. Daca  $e(s)$  este negativ, valoarea sa absoluta este egala cu numarul de procese blocate in sirul  $f(s)$ . Deci avem:

$$(8.7.) \quad e(s) = e_0(s) - np(s) + nv(s)$$

daca:

$e(s) < 0$ , avem:

$$(8.8.) \quad e_0(s) + nv(s) < np(s)$$

Atunci relatia (8.5.) va da:

$$(8.9.) \quad \begin{cases} nf(s) = e_0(s) + nv(s) \\ -e(s) = np(s) - nf(s) \end{cases}$$

e. Daca  $e(s)$  este pozitiv sau zero, valoarea sa ne da numarul de procese care pot trece de semaforul s fara a se bloca.

### 8.1.2. *Semafoare de excludere mutuala.*

Excluderea mutuala se defineste prin introducerea unui semafor "exmut" cu valoarea initiala 1, fiecare proces executandu-se dupa urmatorul program:

```
inceput  
P (exmut);  
sectiunea critica;  
V (exmut)  
sectiunea critica;  
sfirsit
```

Pentru a arata valabilitatea acestei solutii vom demonstra doua teoreme:

*Teorema 8.2.* La un moment dat cel mult un proces se afla in sectiunea sa critica.

Numarul de procese in sectiunea critica este egal cu:

$$(8.10.) \quad nf(exmut) - nv(exmut)$$

Conform teoremei 8.1. avem:

$$(8.11.) \quad nf(exmut) = \min [np(exmut), 1 + nv(exmut)]$$

de unde rezulta:

$$(8.12.) \quad nf(exmut) - nv(exmut) \leq 1$$

*Teorema 8.3.* Daca nici un proces nu se gaseste in sectiunea critica, nu exista procese blocate in spatele semaforului de excludere mutuala.

Daca nici un proces nu se gaseste in sectiunea critica, avem:

$$(8.13.) \quad nf(exmut) = nv(exmut)$$

Daca procesul ar astepta in spatele semaforului,

avem:

$$(8.14.) \quad nf(exmut) < np(exmut)$$

Relatiile (8.5.) si (8.14.) dau:

$$(8.15.) \quad nf(exmut) = nv(exmut) + 1$$

ceea ce este incompatibil cu relatia (8.13.).

### 8.2. *Tranzitiile posibile de stare ale proceselor.*

Intre diverse procese apar interactiuni datorate de exemplu utilizarii acelorasi resurse. Aceste interactiuni trebuie sa fie cunoscute pentru a se putea realiza o gestiune corecta. Un proces poate sa se gaseasca in una din urmatoarele stari:

- *gata de executie*: procesul a fost creat, executia sa nu este terminata;

- *inactiv*: cind executia unui proces s-a terminat se spune ca el devine inactiv;

- *activ*: procesul se executa efectiv in acel moment;
- *in asteptare*: executia procesului nu este terminata, dar a fost oprit deoarece o conditie necesara continuarii procesului nu este indeplinita momentan.

Exista doua reguli legate de gestiunea evolutiei starii unui proces:

- un proces nu poate face sa "regreseze" decit propria sa stare;
- numai sistemul de gestiune a proceselor poate determina lista cu procese "active".

Vom avea deci o evolutie progresiva fara posibilitatea reintoarcerii pina la atingerea starii "activ".

Din starea "activ" un proces poate sa treaca in una din urmatoarele stari:

- "gata de executie" cind executia sa este oprita deoarece s-a cerut crearea unui proces cu prioritate mai mare;
- "in asteptare" cind continuarea sa cere realizarea unor conditii care nu sint indeplinite;
- "inactiv" cind si-a terminat executia.

Din starea "in asteptare" un proces nu poate trece decit in starea "gata de executie", in urma executiei unei operatii de sincronizare care semnaleaza ca a fost indeplinita conditia care era asteptata. Deci nu poate trece in stările:

- "inactiv" deoarece se presupune o actiune regresiva asupra propriei stari, lucru care nu este posibil decit daca ar fi fost in stare "activ";
- "activ" deoarece lista de procese active este alcătuita cum am vazut numai de sistemul de gestiune a proceselor.

Se disting patru tipuri de operatii de sincronizare:

- operatia de "creare a procesului": un proces activ cere pentru un alt proces cu nume precizat, trecerea din starea "inactiva" in starea "gata de executie";
- operatia de "punere in asteptare": un proces activ cere pentru el insusi trecerea in starea "in asteptare" atunci cind o executie nu este realizata;
- operatia de "activare": un proces activ cere pentru un alt proces al carui nume e precizat, trecerea din starea "in asteptare" in starea "gata de executie", deoarece conditia asteptata s-a realizat;

inactiv ---> gata de executie ---> activ

Fig.8.1. Executia progresiva a starii proceselor.

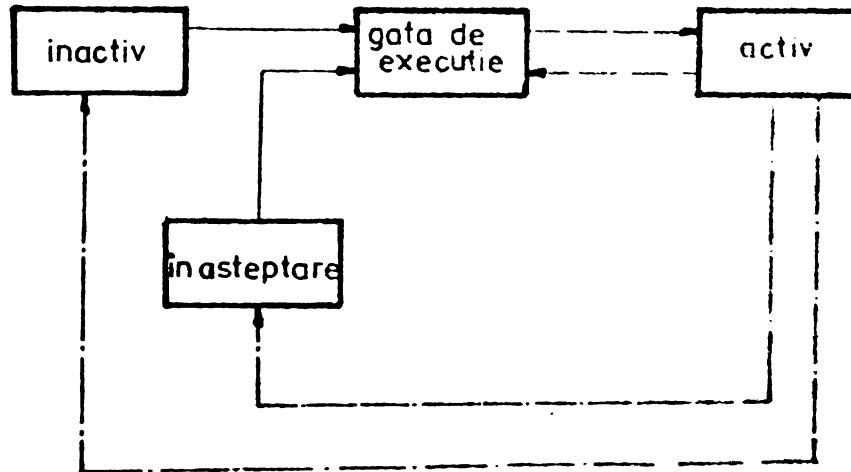


Fig. 8.2. Tranzitiile posibile de stare.

- > actiune altui proces activ asupra procesului considerat.
- .-.-> actiunea procesului considerat asupra starii proprii.
- - -> tranzitie de stare datorata sistemului de gestiune a proceselor.

-operatia "sfirsit de proces": un proces activ cere pentru el insusi trecerea in starea "inactiv", deoarece executia sa a luat sfirsit.

Se doreste in general ca procesele sa fie independente unele fata de altele. Starile posibile ale proceselor pot fi impartite in doua grupe:

- grupa 1-a reuneste starile care corespund cazurilor care nu au nici o actiune programata care sa interzica executarea procesului, adica: "gata de executie" si "activ";

- grupa 2-a reuneste starile care corespund cazurilor in care nu exista o actiune explicit programata care interzice executarea procesului, adica "inactiv" si "in asteptare".

In functie de aceste grupe starea unui proces va "prograsa" cind trece din grupa 2-a in grupa 1-a si va "regresa" cind trece in sens invers. Toate actiunile regresive ale unui proces  $P_i$  asupra starii altui proces  $P_j$  sint interzise,  $P_i$  putind sa-si regreseze numai propria sa stare.

### 8.3. Mecanisme de sincronizare .

Problema sincronizarii consta in construirea unui mecanism care sa fie independent de viteza si care sa permita unui proces activ:

- sa blocheze un alt proces sau sa se blocheze el insusi in asteptarea unui semnal de la alt proces;

- de a activa un alt proces si de a-i transmite informatii.

In ultimul caz procesul caruia ii este destinat semnalul de activare poate fi deja in starea "activ", deci trebuie sa se defineasca mai precis ce se intimpla. Exista doua posibilitati:

- semnalul de actionare nu se memoreaza, in consecinta el va fi pierdut daca procesul nu il asteapta;

- semnalul este memorat si procesul respectiv nu se va bloca in cursul urmatoarei operatii de blocaj asupra procesului.

Tehnicile de sincronizare pot fi clasificate in doua categorii:

- sincronizare "directa", cind procesul actioneaza direct asupra unui alt proces sau asupra sa insasi;

- sincronizarea "indirecta" cind procesul actioneaza asupra altor procese prin intermediul unui mecanism special.

Una din operatiile pe care trebuie sa le faca cuplul din cadrul SDM este "scrierea initiala" adica inscrierea capului de sector ("header") cu informatii care sa permita identificarea fiecarui sector. Aceasta operatiune la sisteme putin evolute este facuta sub conducerea procesorului. La sistemele mai complexe cuplul a primit in componenta sa o portiune specializata acestui scop, scrierea initiala a discului efectuindu-se "off-line". In timpul exploatarei UDM, atat in scriere cit si la citire, in perioada de cautare a sectorului dorit, exista o faza de comparare a adresei curente, cu adresa ce se gaseste momentan sub capul de scriere/citire ales si deja positionat mecanic pe pista dorita. La cuploarele microprogramate aceasta operatie poate fi executata de un subprogram de comparare care poate fi solicitat simultan de catre mai multe UDM. Aceasta este o problema de excludere mutuala care poate fi solutiionata prin utilizarea unui semafor  $e(s)$  initializat la valoarea 1. Cele doua microprograme care au o zona de excludere mutuala trebuie sa fie scrise conform fig.8.3.

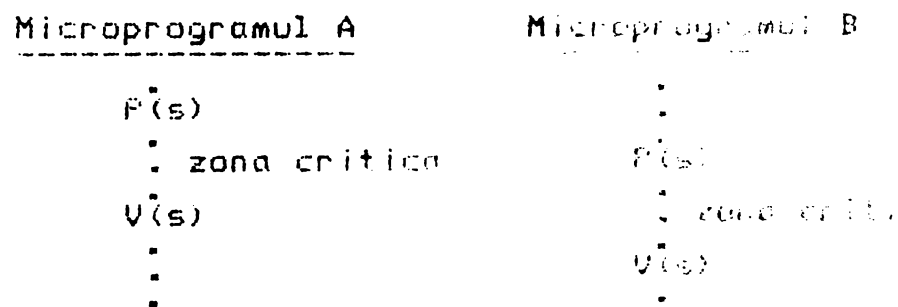


Fig.8.3. Programe cu excludere mutuala.

Deasemenea, utilizarea semafoanelor ca mecanismului de sincronizare permite simplificarea apelului diverselor microprograme, ca urmare a unor microprograme care sint activate in urma unei intreruperi. De exemplu, daca un microprogram A activat de un microprogram de intreruperi I dorim sa declanseze la rindul sau un alt microprogram B, putem folosi urmatoarea schema:

- se utilizeaza doua semafoare:  
 $e(s_1)$  si  $e(s_2)$
- se initializeaza  $e_0(s_1) = e_0(s_2) = -1$
- pentru microprogramele in cauza se stabilesc valorile initiale ale numaratorului de

instructiuni la valorile: inceput I, inceput A si respectiv inceput B.

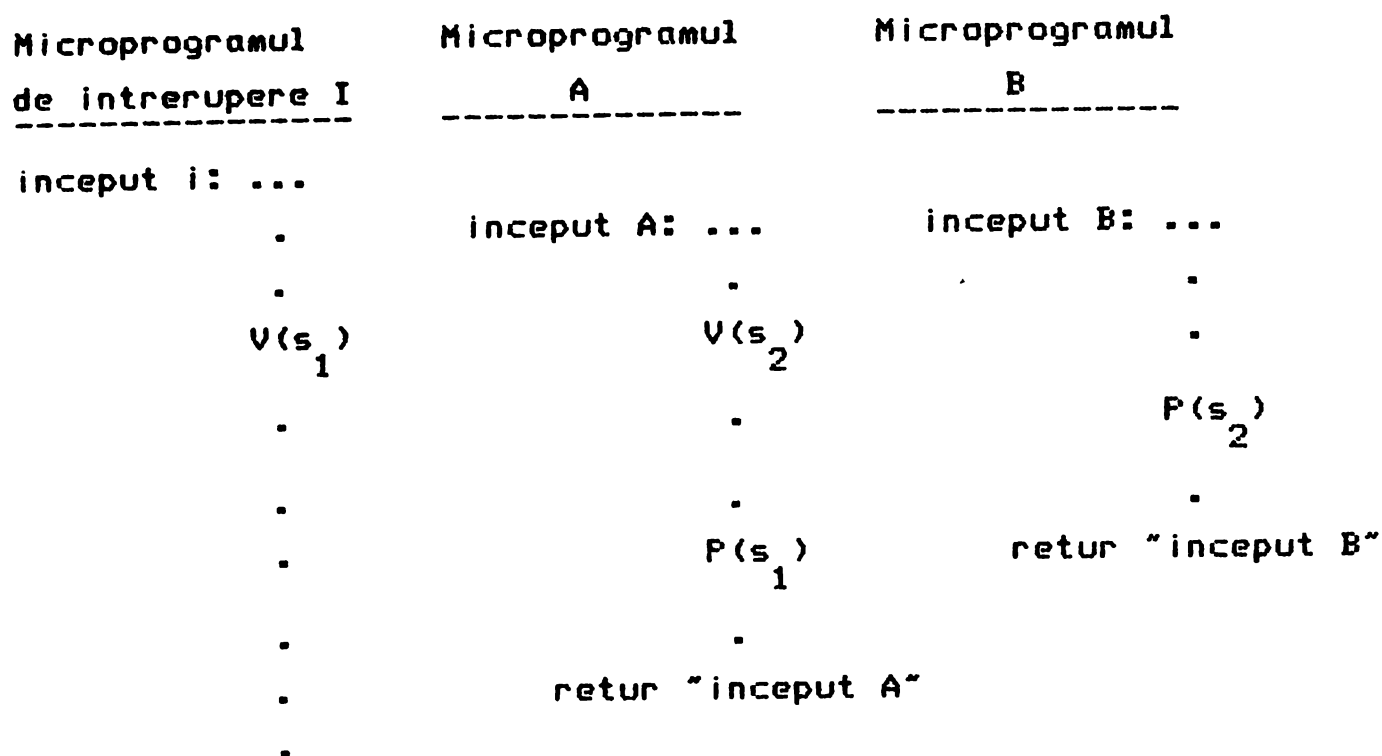


Fig.8.4. Sincronizari cu semafoare.

#### 8.4. Concluzii

In cazul SDM cu cuplor microprogramat care lucreaza in sisteme biprocesor sau multiprocesor apar probleme de sincronizare intre diversele lucrari care se desfasoara simultan in cadrul subsistemului, lucrari care cer accesul la aceleasi resurse hardware (UDM) si respectiv software (microprogramele interne ale cuplorului). Mecanismele de sincronizare cu semafoare utilizate la nivelul SO, pot fi utilizate si pentru rezolvarea problemelor similare care apar in cadrul SDM. Problemele de sincronizare care apar la nivelul microprogramelor se pot rezolva prin alcatuirea unor microprograme interconectate care utilizeaza semafoare software, formatul microinstructiunii putind fi ales initial sau nu pentru a usura acest scop. In aceasta categorie intra: scrierea, citirea, verificarea, detectia si corectia erorilor, scrierea initiala a discului, transferul direct de informatii intre UDM ale SDM. Alte probleme de sincronizare se pot rezolva prin metode hardware cum sint: problemele de prioritate, tratarea intrerup-



perilor, accesul la circuitele fizice de calcul al paritatii; circuitele respective fiind prevazute in partea de acces cu circuite care au facilitati de rezolvare conditionata.

Rezolvarea problemelor de sincronizare la nivelul SDM, face ca subsistemul sa capete un grad mare de indepenenta si sa permita transformarea sa intr-un Subsistem de Intrare-Iesire (SII) independent, care sa aibe in componenta sa pe linga UDM si toate celelalte tipuri de echipamente priferice. Datorita cuceririlor tehnologice in domeniul circuitelor integrate si a dinamicei de imbunatatire a tehnologiilor din domeniul tehnicii de calcul, in structura sistemelor de calcul evaluate chiar si a sistemelor minicalculatoare, se va impune utilizarea SII care sa comunice cu procesoarele prin instructiuni evaluate, dialogul devenind minim. Se observa posibilitatea implementarii metodelor utilizate in trecut la sistemele de calcul foarte mari, la sistemele de calcul mici si la minicalculatoare.

## CAP.9. LOCUL SDM IN CONFIGURATIILE MULTIPROCESOR

### 9.1. *Multiprocesorul banalizat*

Configuratiile biprocesor sau multiprocesor permit utilizarea UDM de catre doua sau mai multe procesoare simultan. Pentru studiul teoretic al problemei se defineste notiunea de "multiprocesor banalizat" (MB) care este: o structura cu o memorie unica partajata de mai multe procesoare, fiecare din acestea putind executa toate procesele existente in sistem, determinindu-si fiecare lucrarile pe care trebuie sa le execute. In structura MB se poate utiliza foarte bine un singur SDM, daca acesta va fi corect dimensionat din punctul de vedere al capacitatii totale si al vitezei de raspuns la cererile procesoarelor.

Consecintele definitiei MB asupra structurii pot fi grupate in urmatoarele grupe:

- accesul tuturor procesoarelor la toate resursele

- autonomia proceselor fata de procesoare
- autonomia proceselor unul fata de celalalt (problema analizata la Cap.8.).

#### 9.1.1. Accesul tuturor procesoarelor la toate resursele.

In configuratie trebuie sa existe legaturi fizice intre toate elementele constructive ale configuratiei si procesoare. Memoria se considera comuna pentru toata structura. In structura exista organe de comutare care permit stabilirea legaturilor impuse in anumite situatii. Aceste organe de comutare trebuie sa asigure urmatoarele functiuni:

9.1.1.1. Sa arbitreze anumite conflicte de acces la resursele care sint specifice structurii multiprocesor.

Resursele trebuie sa raspunda numai la cererile care vin de la procesoare. Gestiunea lor este o problema independenta de existenta organelor de comutare. Deci daca accesul la un periferic este impartit intre mai multe procese, eventualele probleme de excludere mutuala trebuie sa fie rezolvate prin operatii de sincronizare scrise in procesul propriu-zis si nu in organul de comutare care da accesul la aceasta resursa. Dar la nivelul de realizare al instructiunilor de intrare-iesire pot sa se puna probleme specifice structurii multiprocesor. Aceasta problema e necesar sa se rezolve la nivelul organului de comutare si al microprogramarii, deoarece se doreste ca structura sa fie transparenta la nivelul programarii. Aceste probleme apar cind un organ de comutare permite accesul la mai multe resurse.

Ele sint de doua feluri:

- conflicte datorate simultaneitatii a doua cereri de schimb cu doua periferice care impart aceleasi organe de comutare la procesoare;

- necesitatea de a face indivizibile anumite secvente de microschemburi, adica sa se blocheze organul de comutare un anumit interval de timp necesar desfasurarii unor microinstructiuni care compun o instructiune de intrare-iesire.

9.1.1.2. Orientarea intreruperilor prioritare spre procesoare.

Problema inversa celei de la 9.1.1.1. Organul de comutare trebuie sa decida spre care procesor

sa stabileasca legatura. Prin banalizarea procesoarelor si respectind regulile de prioritate se poate stabili urmatoarea strategie: intreruperile prioritare sint orientate spre procesorul mai putin prioritar, deci mai putin incarcat.

9.1.2. Autonomia proceselor fata de procesoare

"Banalizarea" procesoarelor permite inlaturarea unei deficiente majore aparute in cazul structurii cu procesoare specializate si anume: la un anumit moment toate procesele care se executa pot avea nevoie de un anumit procesor specializat (de exemplu de procesorul de virgula mobila), celelalte procesoare stind libere. Prin autonomia fata de procesoare intelegem ca un proces poate fi inceput intr-un procesor si sa fie apoi continuat in oricare alt procesor al structurii.

Pentru a se asigura "transparenta" structurii trebuiesc respectate urmatoarele reguli:

- daca exista un singur procesor, informatiile referitoare la procesoarele absente se prezinta ca si cum acestea ar fi intr-o stare in care nu pot executa un proces.

- lista proceselor active poate fi facuta numai de S.O. (si anume de partea denumita "Planificator"). Nici o operatie de sincronizare nu trebuie sa poata cere trecerea unui proces direct in starea "activ". Singura cerere posibila trebuie sa fie trecerea in starea "gata de executie", planificatorul singur decizind trecerea din starea "gata de executie" in starea "activ". In acest fel structura este transparenta pentru programare, putindu-se cere executarea unui proces, dar nu si afectarea unui anumit procesor.

"Banalizarea" structurii impune ca toate procesoarele trebuie sa poata executa Planificatorul, ori Planificatorul trebuind sa tina cont de starea si de prioritatea procesoarelor, trebuie ca toate aceste informatii sa fie accesibile pentru toate procesele, deci sa fie plasate intr-o memorie comuna.

## 9.2. Configuratiile cu minicalculatoare INDEPENDENT.

Din anul 1979 anul in care a intrat in productie de serie, minicalculatorul I-100 a fost produs in peste 200 de exemplare iar din 1982 a inceput sa se produca si minicalculatorul I-102, care foloseste aceleasi busuri si echipamente periferice ca si I-100. Cu sisteme I-100 au fost rezolvate anumite aplicatii in care s-a impus utilizarea unor configuratii bicalculator, la care a aparut necesitatea utilizarii in comun a acelorasi UDM. La "Sistemul de avertizare hidrometeorologica la inundatii" si la "Sistemul de supraveghere a instalatiilor DLTCIT-Craiova" s-a impus organizarea unei configuratii ca in fig. 9.1., unde sint utilizate doua calculatoare identice interconectate prin cuploare intercalculatoare. Un calculator este "activ" si este cuplat cu procesul, al doilea fiind "rezerva calda" ("stand by"), acestea din urma primind informatiile prelucrate de primul, pentru a fi stocate in memoria proprie si apoi pe UDM. In acest fel in cazul unei defectari a sistemului activ defectiune sesizata de logica de avarie cu ceasuri de garda, se produce o comutare a legaturilor cu procesul spre calculatorul al doilea care trece in starea "activ". Acesta va demara prin rulara unui program de initializare, care cere repetarea transmiterii ultimilor date de la proces, date care s-au pierdut si nu au putut fi prelucrate de sistemul defectat.

In configuratie anumite periferice simple (imprimante, lectorul de cartele, display-urile) au fost conectate prin intermediul unui comutator de bus, comandat de asemenea de logica de avarie.

Dupa intrarea sistemelor in exploatare s-a vazut ca solutia impusa initial nu este foarte buna, deoarece transferul permanent al informatiilor prin cuploarele intercalculatoare ocupa un timp insemnat din timpul de lucru al calculatorului activ. De aceea schema s-a modificat (fig.9.2.) organizindu-se o grupare de UDM cu dublu acces, o logica de comutare permitind cuplarea UDM la unul din cuploarele sistemului de calcul. In fiecare configuratie mai exista cite un SDM propriu, echipat

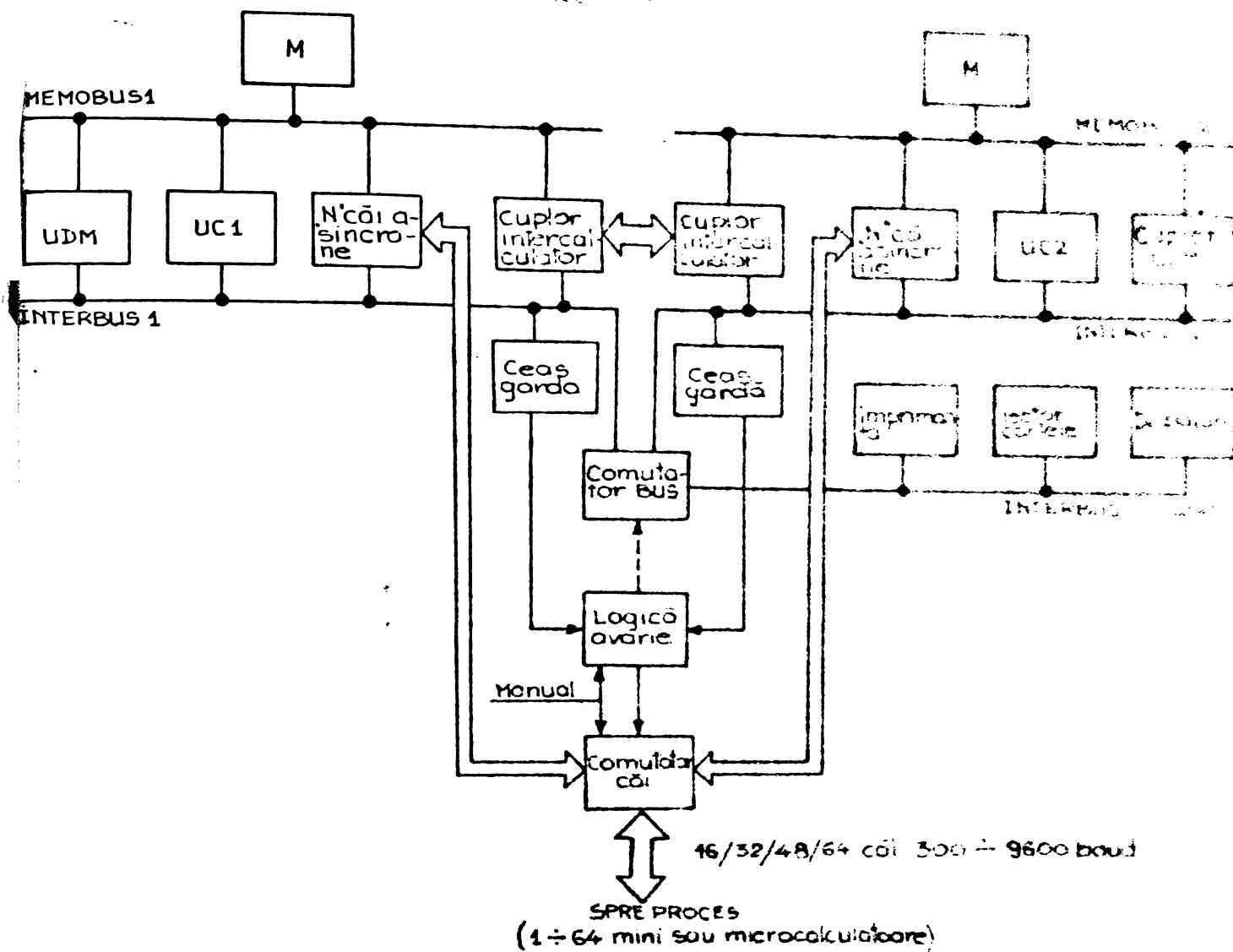


Fig. 9.1. Configurație cu sisteme interconectate

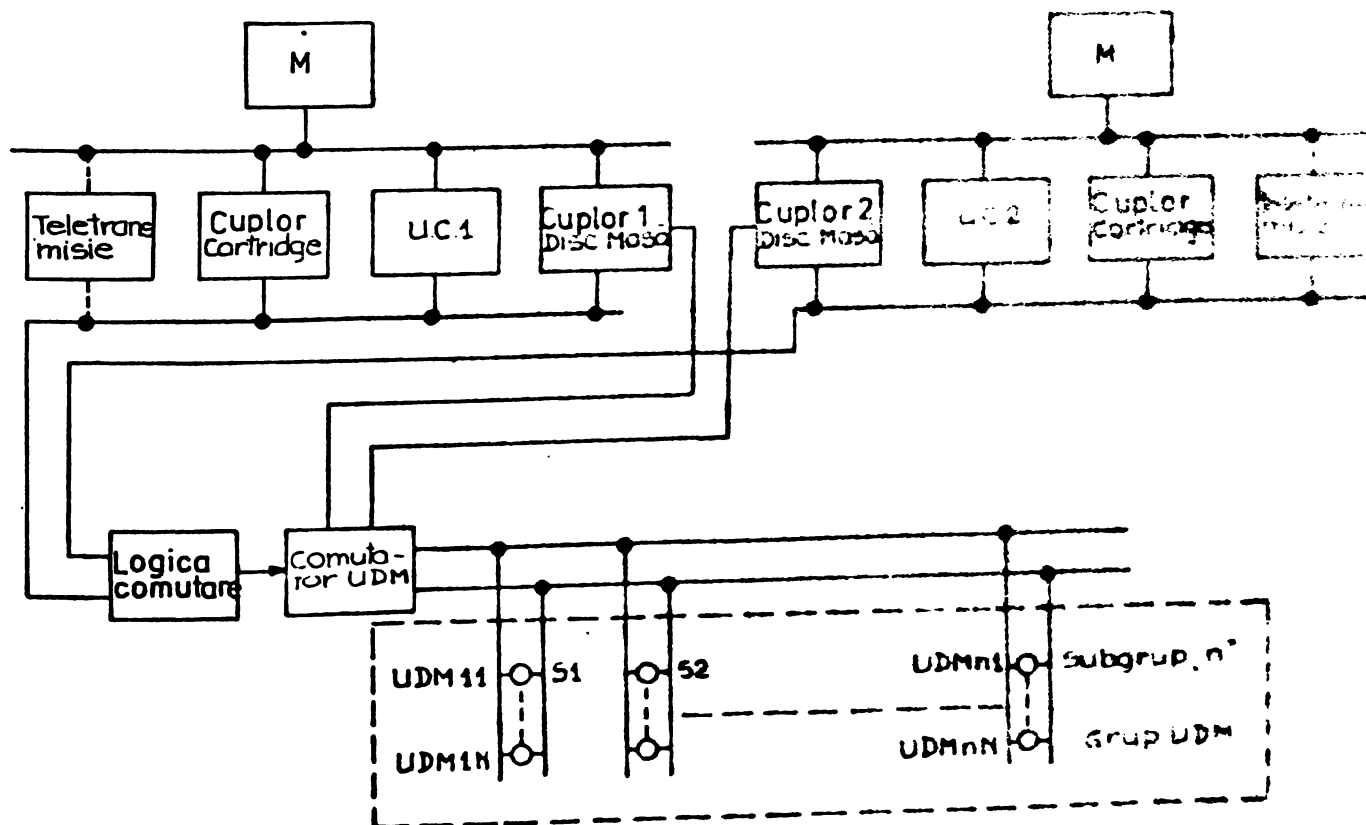


Fig. 9.2. Configurație utilizând UDM cu dublu acces grupat într-o memorie externă de capacitate foarte mare

cu o UDM tip cartridge de mica capacitate (12 Mo) pe care se afla S.O. Gruparea comuna de UDM este utilizata pentru stocarea informatiilor din proces.

O alta varianta posibila (fig.9.3.) utilizeaza comutatoare de busuri comandate de logica de comutare. Fiecare sistem are un SDM cu discuri cartridge pentru S.O. si un SDM de mare capacitate cuplat dupa comutatoarele de bus pe busurile comune. De asemenea pe busurile comune se poate conecta si Subsistemul de Banda Magnetica (SBM).

### *9.3. Influenta introducerii circuitelor VLSI asupra arhitecturii configuratiilor multiprocesor.*

Cum s-a vazut o configuratie multiprocesor contine cel putin doua procesoare identice care au acces la o memorie comuna si la un sistem de intrare-iesire comun, intregul sistem fiind controlat de un S.O. care gestioneaza interactiunea intre procesoare si procesele ce trebuiesc rezolvate. Aparitia microprocesoarelor VLSI cu 16 biti a dus la aparitia unor posibilitati noi pentru rezolvarea de configuratii multiprocesor, dar au aparut si noi probleme cum ar fi modul de conectare a microprocesoarelor cu memoria si sistemul de intrare-iesire. Exista trei posibilitati de a solutiona problema: configuratie cu busuri comune, configuratie cu matrice de comutare (crossbar switches) si configuratie cu memorie multiport. Configuratia cu busuri comune (fig.9.4.) este cea mai simpla si cea mai ieftina si din aceasta cauza este recomandata.

Noile memorii rapide si ieftine impreuna cu microprocesoarele vor putea duce la inlaturarea anumitor parti special concepute din vechile arhitecturi, cum ar fi de exemplu cuploarele de intrare-iesire. Noile tehnologii permit in primul rind, introducerea unei parti a cuplorului in echipamentul periferic propriu-zis, acesta devenind "inteligent", putind prelua de exemplu functiile de detectie si corectie a erorilor, stocarea intermediara a datelor, si anumite prelucrari asupra acestora, etc. La arhitecturile actuale sarcinile de intrare-iesire erau in majoritate rezolvate de U.C. Pastrarea acestui tip de arhitectura, numai cu schimbarea tehnologiei nu ar aduce o marire

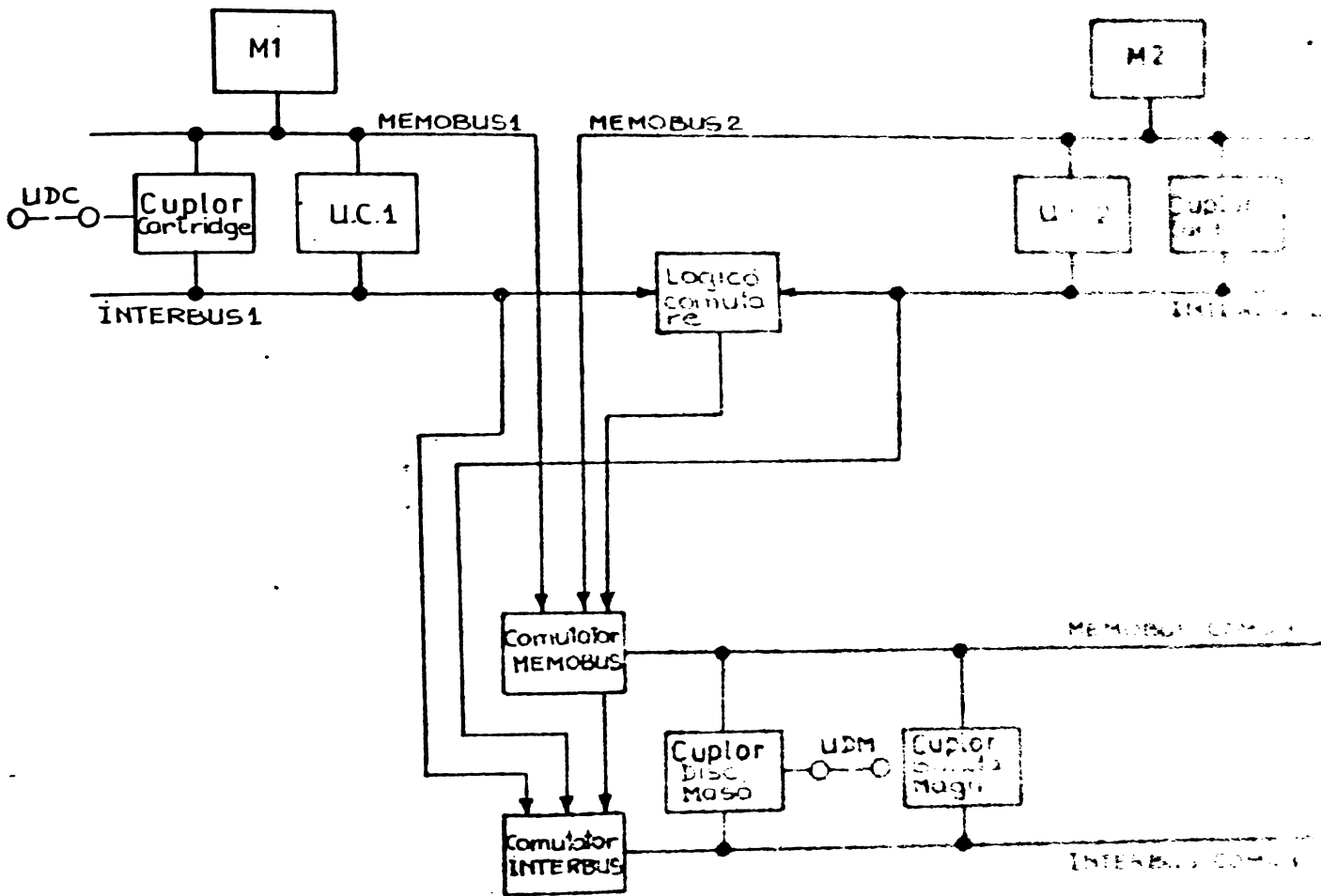
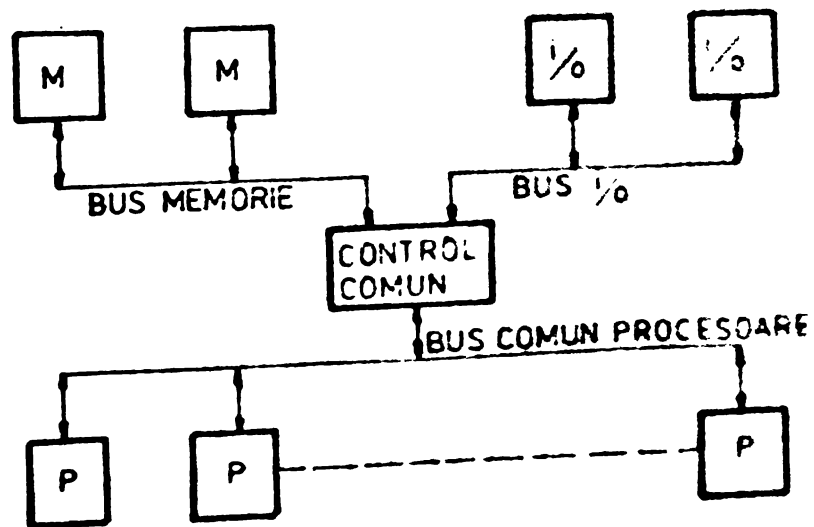


Fig. 9.3. Configuratie cu comutatoare de BUS-uri. (sistemă centrală)

9.4



efectiva a performantelor sistemului. Ceea ce se cere este definirea unor noi arhitecturi care sa utilizeze la maximum posibilitatile de calcul ale microprocesoarelor VLSI, utilizate ca subsisteme de intrare-iesire. Problema definirii unei arhitecturi de intrare-iesire se apropie foarte mult de problema definirii arhitecturii rețelei de calculatoare, de fapt S.I.I. putind fi privit ca o retea locala de comunicatii de date. Asa cum o retea conduce la distribuirea prelucrării datelor, o arhitectura de intrare-iesire bine definita poate duce la prelucrarea distribuita a sarcinilor de intrare-iesire, puterea de calcul fiind distribuita la cuploare unde este real necesara si unde se gasesc si datele. Aceasta actiune de descentralizare conduce la micșorarea legaturilor cu procesorul propriu-zis si la descongestionarea acestora, deci la cresterea performantelor de ansamblu ale sistemului.

Urmind exemplul conceptului arhitecturii de retea, se poate defini o arhitectura distribuita de intrare-iesire care utilizeaza mesaje de control a intrării-iesirii si consta din patru nivele de protocol (fig.9.5.).

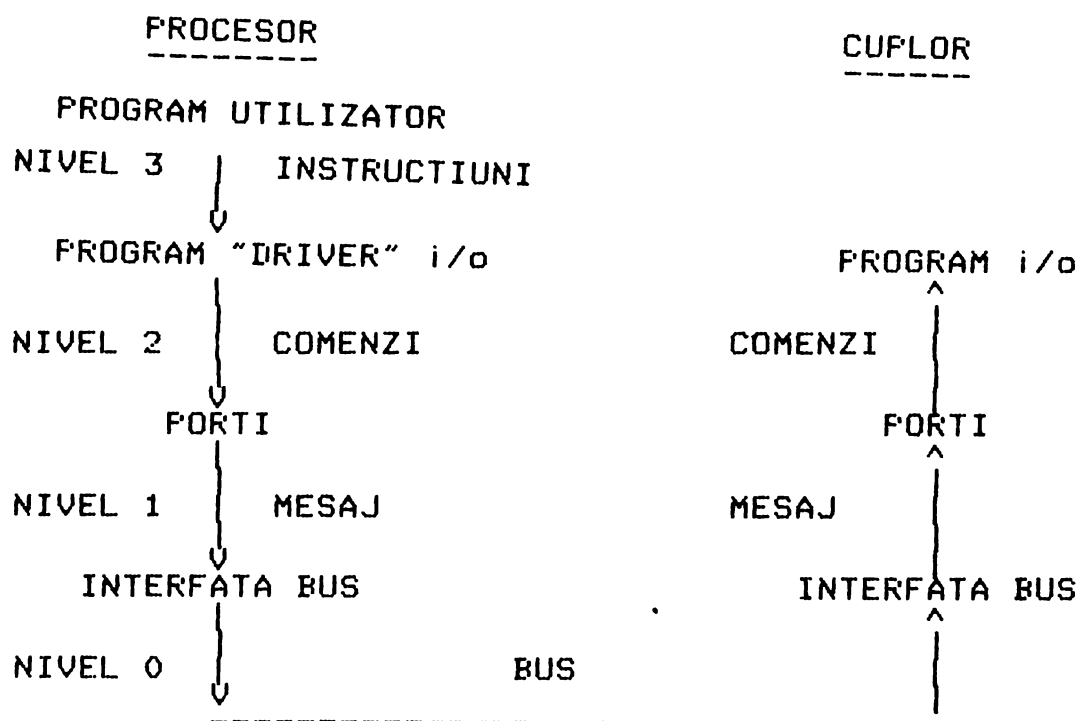


Fig.9.5. Arhitectura de intrare-iesire cu nivele de protocol.

Nivelul 0 este interfata fizica formata din circuitele emitatoare-receptoare si liniile fizice ale busului.



Nivelul 1 este nivelul de transfer al mesajului, mesaj format din bitii de stare, comenzi sau date, care a fost creat in nivelul 2 si a fost trimis prin intermediul unei porti de intrare-iesire. Nivelul 2 este nivelul de date in care mesajele contin informatii de control de intrare-iesire de nivel inalt (cum ar fi: SCRIERE, SFIRSIT, CITIRE, etc.). Nivelul 3 este nivelul interfetei cu driverul specific din S.O., unde un program utilizator poate cere o operatie de intrare-iesire.

Implementarea acestor nivele cu microprocesoare VLSI aduce urmatoarele avantaje:

- interfata de intrare-iesire poate fi standardizata la nivelul driverului software de intrare-iesire, astfel ca se va utiliza un singur driver, care poate fi intrebuintat usor cind este cazul;

- driverul software de intrare-iesire se simplifica, atit datorita transferarii in cuplorul cu microprocesor a tuturor functiilor dependente de echipamentul periferic, cit si utilizarii unui set de protocoale bine structurat;

- posibilitatea cuplarii ulterioare a unor periferice inteligente care vor avea noi posibilitati fata de cele actuale;

- utilizarea unor busuri de intrare-iesire mai simple si mai ieftine, deoarece controlul cu mesaje orientate implica un volum mai mic de informatii care trebuie transferate, deci se poate trece la utilizarea unor busuri seriale, fara a diminua performantele sistemului;

- cresterea performantelor procesorului central, deoarece acesta este eliberat de calculele specifice operatiilor de intrare-iesire.

Cu cit procesorul central devine mai rapid cu atit va ocupa mai mult memoria in detrimentul sistemului de intrare-iesire. De aceea apare necesitatea utilizarii unor busuri separate pentru utilizarea memoriei.

In conditiile utilizarii microprocesoarelor VLSI, SDM va putea fi un microcalculator al retelei, standard din punctul de vedere al organizarii hardware, dar specializat prin microprogramare. Microprogramele specifice pot fi incarcate la initializarea sistemului pe microcalculatorul desemnat pentru a realiza functia de SDM, UDM pot fi cuplate fizic direct la acesta, sau printr-un bus de intrare-iesire se pot cupla la o serie de microcalcu-

latoare fiecare din acestea putind deveni beneficiarul acestor echipamente periferice. Aceasta solutie care nu prevede specializarea hardware, are avantajul realizarii unei fiabilitati de ansamblu foarte bune.

## CAP.10. PROBLEMA CONFLICTELOR DE ACCES

Cuplarea simultana a UDM la doua sau mai multe sisteme de calcul este posibila din punct de vedere al realizarii fizice. Solutia prezinta avantaje economice, deoarece UDM sint echipamente periferice scumpe, fabricate cu un import rezidual mare, fata de cuploarele specifice care se pot realiza cu circuite din tara si tari socialiste la un cost relativ scazut.

Pentru sistemele cu minicalculatoare nu s-a studiat in exploatare coeficientul de utilizare reala a UDM. De aceea folosind o metoda originala se vor studia rezervele SDM si limita pina la care considerentele de ordin economic nu duc la diminuarea performantelor tehnice.

Un factor care poate diminua mult performantele unui sistem de calcul este constituit de conflictele care apar in momentul in care se solicita accesul la un ansamblu utilizat in comun de mai multi utilizatori. La sistemele multiprocesor pot apare conflicte pe accesele la: memorie, echipamentele periferice comune, operatorul de virgula flotanta, etc.

In cele ce urmeaza se analizeaza problema conflictelor de acces ce apar in sistemele multiprocesor, datorate utilizarii unui singur cuplor pentru unitatile de discuri magnetice, precum si randamentul acestor configuratii.

### 10.1. *Analiza teoretica a problemei.*

Pentru studiul teoretic ne plasam in cazul general in care "n" procesoare ale sistemului multiprocesor impart aceleasi SDM. SDM este considerat referinta pentru configuratie, dorind sa rezolvam problema conflictelor de acces ce apar la intrarea SDM. Fata de SDM un procesor activ poate sa se gaseasca in una din urmatoarele trei stari diferite:

(1). "executie": procesorul lucreaza dar nu cere acces la SDM;

(2). "in asteptare": procesorul a facut o cerere de acces si asteapta ca accesul sa se libereze pentru a putea realiza efectiv un schimb de informatii cu subsistemul;

(3). "in acces": un schimb se deruleaza intre procesor si subsistem.

Vom face urmatoarele ipoteze:

a). La momentul dat numai un procesor poate fi in starea (3), deoarece consideram cazul in care subsistemul are un singur acces.

b). Procesoarele sint independente.

c). Pentru un procesor care se afla in starea (1), posibilitatea de a trece in starea (2) sau (3) urmeaza o lege exponentiala care are urmatoarea densitate a probabilitatilor:

$$(10.1.) \quad f_1(t) = \alpha e^{-\alpha t}$$

unde  $\alpha$  este egal cu inversul valorii medii a timpului care se separa pentru un procesor dat un sfirsit de acces si cererea urmatoare de acces, adica durata in care un procesor ramine in starea (1) "executie".

d). Pentru un procesor care se afla in starea (3), posibilitatea va fi:

$$(10.2.) \quad f_2(t) = \beta e^{-\beta t}$$

unde  $\beta$  este egal cu inversul valorii medii a timpului de acces la subsistem pentru un procesor dat, deci durata pe parcursul careia procesorul ramine in starea (3) "in acces". Fie  $S(i)$  starea corespunzatoare cazului in care "i" procesoare solicita acces la subsistem. Sa consideram un interval de timp  $dt$ , atat de scurt incit posibilitatea de a se produce doua evenimente in timpul  $dt$  sa fie nula.

Observam ca:

- probabilitatea de trecere din starea  $S(i)$  in starea  $S(i+1)$  este  $(n-i) dt$ , deoarece fiecare din cele  $(n-i)$  procesoare aflate in starea  $S(i)$ , care nu sint in cerere de acces la subsistem, pot cere un acces la subsistem cu probabilitatea  $dt$

- probabilitatea de trecere din starea  $S(i)$  in starea  $S(i-1)$  este:  $dt$ , deoarece un singur procesor poate fi in starea de acces la subsistem.

Se poate imagina modelul din fig.10.1.

Fie  $P_i(t)$  probabilitatea ca un procesor sa fie in starea  $S_i$  in momentul  $t$ . In ipoteza in care modelul se considera

stationar, probabilitatea de a parasii starea  $S_i$  in intervalul de timp  $dt$ , este egala cu probabilitatea de a intra in starea  $S_i$ . Deci:

$$(10.3.) \quad \begin{cases} \forall i = 1, 2, \dots, n-1 : (n-i+1)\alpha P_{i-1} - \beta P_i = \\ \quad = (n-i) P_i - P_i \\ \text{pentru } i=n : \alpha P_{n-1} - \beta P_n = 0 \end{cases}$$

deci:

$$(10.4.) \quad \forall i = 1, 2, \dots, n : (n-i+1)\alpha P_{i-1} - \beta P_i = 0$$

$$(10.5.) \quad P_i = (n-i+1) \cdot \frac{\alpha}{\beta} \cdot P_{i-1}$$

$$(10.6.) \quad P_i = \frac{n!}{(n-i)!} \left(\frac{\alpha}{\beta}\right)^i \cdot P_0$$

Deoarece:

$$(10.7.) \quad \sum_{i=0}^n P_i = 1$$

avem:

$$(10.8.) \quad P_0 = \left[ \sum_{i=0}^n \frac{n!}{(n-i)!} \left(\frac{\alpha}{\beta}\right)^i \right]^{-1}$$

Numarul mediu de procesoare care se afla in asteptarea unui acces este:

$$(10.9.) \quad E_n = \sum_{i=0}^n (i-1) \cdot P_i$$

Fiecind de la aceasta valoare, se poate defini un coeficient  $k_n$ , multiplicand mediu al prelungirii timpului de executie in procesor, datorita conflictelor de acces la subsistem.

In orice moment avem  $(n-E)$  procesoare active, care deci nu sint in asteptarea unui acces. Pentru a avea o activitate echivalenta celei efectuate de  $n$  procesoare fara aparitia vreunui conflict in perioada de timp  $\bar{t}$ , trebuie sa se lucreze pe durata  $\bar{t}'$ , astfel ca:

$$(10.10.) \quad n \cdot \bar{t} = (n-E) \bar{t}'$$

deci:

$$(10.11.) \quad k_n = \frac{\bar{t}'}{\bar{t}} = \frac{n}{n-E}$$

de eroare, pentru iesiri fiind utilizat tot discul magnetic; de asemenea biblioteca de programe utilizate a fost trecuta initial de pe banda magnetica pe unul din discurile magnetice).

Montajul foloseste un monostabil cu baza de timp reglabila care constituie semnalul de referinta  $E_2$ . La intrarea monostabilului se aplica semnalul OCUPAT prelevat de la SDM, pe care-l vom nota cu  $E_1$ . La trecerea semnalului  $E_1$  din starea 0 in 1, se declanseaza monostabilul, iesirea  $\bar{E}_2$  trecind din 0 in 1 pentru o perioada de timp  $\Delta t$ , reglabila. Vom avea:

$$(10.16.) \quad S = \bar{E}_1 \cdot \bar{E}_2$$

Semnalul S este in permanenta 1 cu exceptia cazului in care  $E_1$  ramine pe 1 o perioada de timp mai lunga decit semnalul de referinta  $E_2$ . Daca S ataca un numarator care numara pe frontul cazator al semnalului de intrare, se obtine numarul de ori cind semnalul  $E_1$  este mai lung decit semnalul de referinta  $E_2$ , in timpul unei durate arbitrare de masura  $\tau$ . Semnalul OCUPAT s-a prelevat de la SDM aratind daca accesul este ocupat (1) sau liber (0).

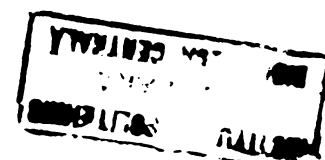
Pentru compensarea intirzierii trecerii semnalului prin monostabil s-a folosit in schema un circuit de intirziere format din componentele D,R,C. Daca aceasta intirziere nu ar fi compensata ar aparea un impuls parazit care a denatura masuratorile (fig. 10.3.).

Masuratorile in etapa I-a s-au facut in doua ipoteze:

- cu un singur disc magnetic
- cu doua discuri magnetice

Cind s-a lucrat cu doua discuri magnetice, pe discul DM1 s-a montat numai sistemul de operare, iar discul DM2 a fost utilizat pentru transferul de date. Lantul de programe cu durata de doua ore, a fost repetat timp de 24 de ore, in repetate rinduri. Astfel s-au obtinut un mare numar de date care au servit la verificarea calculelor teoretice.

Pentru determinarea duratelor in care accesul este ocupat s-au fixat 11 valori diferite pentru  $\Delta t$ . S-au fixat anumite puncte de masura in concordanta cu caracteristicile unitatii de discuri magnetice folosite (CIC 9747) si anume:



$$(10.12.) \quad k_n = \frac{1}{1 - \xi_n}$$

unde:

$$(10.13.) \quad \xi_n = \frac{E}{n} = \frac{1}{n} \left[ \sum_{i=1}^n \frac{i-1}{(n-i)!} \left(\frac{\alpha}{\beta}\right)^i \right] \cdot \left[ \sum_{i=0}^n \frac{1}{(n-i)!} \left(\frac{\alpha}{\beta}\right)^i \right]$$

Particularizind pentru un sistem biprocesor, conflictul proceselor duce la o multiplicare a timpului de executie cu coeficientul:

$$(10.14.) \quad k_2 = \frac{1}{1 - \xi_2}$$

unde:

$$(10.15.) \quad \xi_2 = \frac{E_2}{2} = \frac{1}{\left(\frac{\alpha}{\beta} + 1\right)^2 + 1}$$

## 10.2. Masuratori.

Pentru verificarea concluziilor teoretice s-a imaginat o metoda de masura, utilizindu-se o configuratie biprocesor cu sisteme I-100. Masuratorile s-au facut in doua etape:

- etapa I-a. Configuratie cu un singur procesor pe care cu ajutorul unui montaj suplimentar s-a determinat repartitia duratei acceselor pe intervale de timp;
- etapa II-a. Configuratie biprocesor folosindu-se un cuplor comun pentru UDM.

### 10.2.1. Etapa I-a.

Pentru verificarea premizelor teoretice din cap.10.1. este necesar sa determinam coeficientii  $\alpha$  si  $\beta$ . In acest scop a fost imaginat un circuit logic simplu care a permis ridicarea urmatoarelor curbe (fig. 10.2.).

- variatia duratelor de acces; se pune in evidenta numarul de accese care au durat mai mult de un interval  $\Delta t$  (intrerupatorul in pozitia 2);

- variatia duratelor fara acces; se pune in evidenta numarul de ori cind exista perioade fara acces, care au durat mai mult de un interval  $\Delta t$ . (intrerupatorul in pozitia 1).

S-a utilizat o configuratie I-100 completa (banda magnetica, imprimanta rapida, doua unitati de disc magnetic de 58 Moct., etc.). S-a stabilit un lant de programe software cu durata de 2 ore, similar cu lantul de programe utilizat la receptionarea sistemelor I-100, la care s-au facut anumite modificari cu scopul de a utiliza la maximum SIM (toate iesirile de date nu s-au mai facut pe imprimanta cu exceptia mesajelor

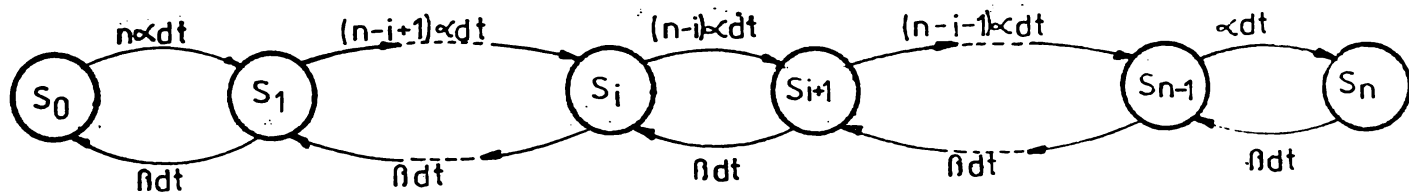


Fig. 10.1. Model cu stari ale procesoarelor.

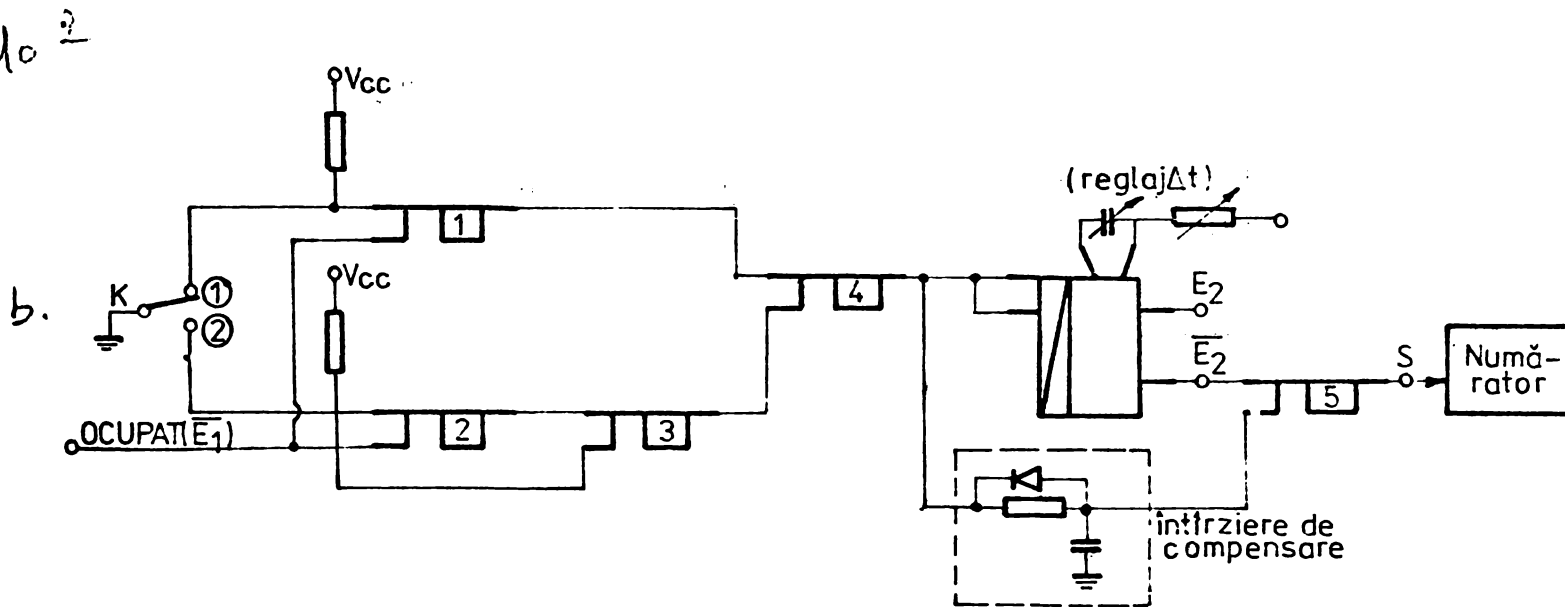
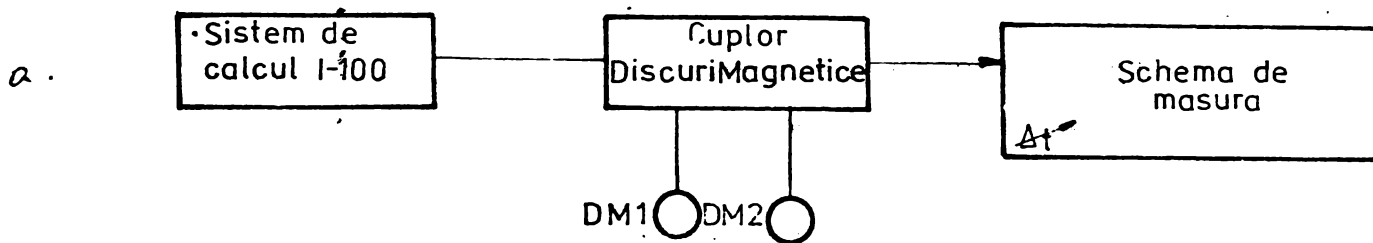
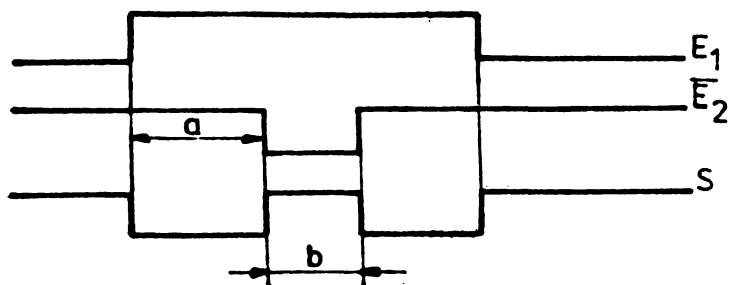


Fig. 10.2. Schema instalatiei de masura.



- a. Intirziere datorata timpului de propagare prin monostabil.
- b. Impuls parazit.

Fig. 10.3. Schema aparitiei impulsului parazit.

7 msec., 35 msec., si 70 msec. corespunzatoare timpului minim, mediu si maxim de pozitionare a capetelor, precum si puncte intermediare care sa permita realizarea unei game de masuratori complete. Valoarea limita a duratei  $\Delta t$  a fost aleasa experimental, fixindu-se la 100 msec., valoare de la care masuratorile nu mai prezentau importanta datorita numarului mic de evenimente inregistrate. In fiecare punct de masura (fiecare valoare diferita a marimii  $\Delta t$ ) s-a trecut lantul de programe de cinci ori. Deoarece ora de calcul are o valoare ridicata, masuratorile s-au efectuat utilizandu-se concomitent cinci scheme logice de masura, pentru fiecare reglindu-se valori diferite ale parametrului  $\Delta t$ . In acest fel timpul total de masura a fost micorat mult fara a se diminua precizia masuratorilor.

Pentru determinarea duratelor in care accesul este liber s-au utilizat 10 puncte de masura, in aceeasi plaje de variatie a parametrului  $\Delta t$  (0 - 100 msec.) majoritatea masuratorilor fiind efectuate pentru valori mici ale acestui parametru (0 - 25 msec.), deoarece s-a estimat ca se lucreaza cu o configuratie intens utilizata, deci timpii de neocupare a accesului vor fi minimi.

Rezultatele masuratorilor sint trecute in tabelele 10.1. si 10.2.

TABELUL 10.1.

$\Delta t$ (msec.)	N - ACCES OCUPAT	
	un disc	doua discuri
0	950.000	950.000
7	950.000	930.000
25	835.000	710.000
35	755.000	580.000
50	585.000	350.000
65	375.000	150.000
70	320.000	90.000
75	205.000	40.000
80	115.000	20.000
90	15.000	10.000
100	5.000	5.000



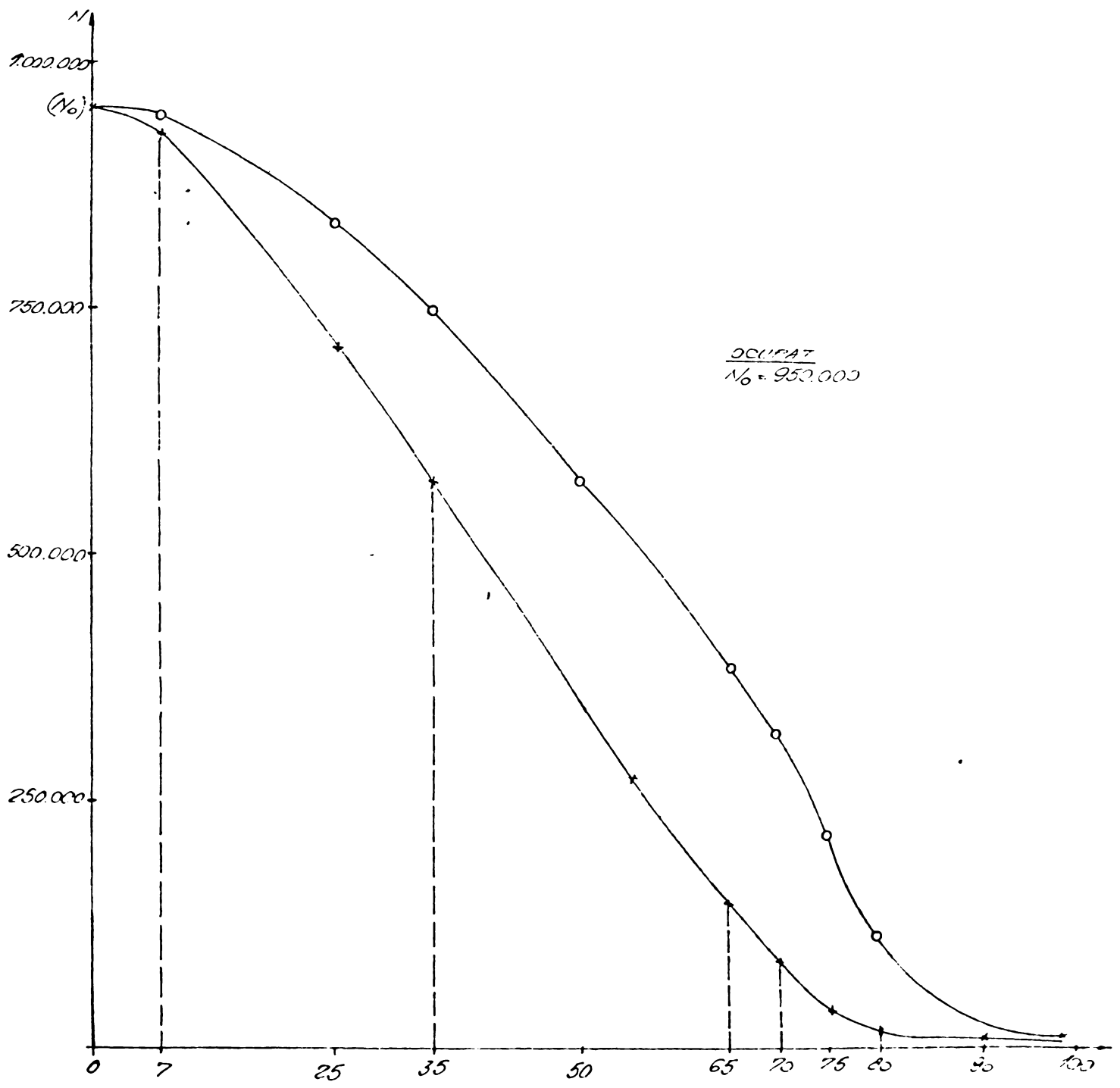


Fig. 10.4.

TABELUL 10.2.

$\Delta t$ (msec.)	N - ACCES NEOCUPAT	
	un disc	doua discuri
0	950.000	950.000
2,5	925.000	940.000
5	100.000	915.000
10	50.000	910.000
15	30.000	675.000
20	25.000	475.000
25	20.000	260.000
50	10.000	75.000
75	7.500	40.000
100	5.000	15.000

Pe baza masuratorilor facute s-au trasat curbele din fig.10.4. si 10.5. Aceste curbe sint de tipul:

$$(10.17.) \quad N = G(\Delta t)$$

unde:

$t$  = o perioada de timp

$N$  = numarul de ori cind o conditie (ocuparea sau neocuparea accesului) dureaza mai mult decit  $\Delta t$ .

De asemenea curbele tratate sint de forma:

$$(10.18.) \quad N = N_0 [1 - F(\Delta t)]$$

unde:

$N_0 = G(0)$  - valoarea la originea curbelor

$F(\Delta t)$  = functia de distributie a perioadelor de acces sau respectiv a perioadelor fara acces.

Valoarea medie "m" a acestor perioade este:

$$(10.19.) \quad m = \int_0^{\Delta t_{\max}} \Delta t \cdot f(\Delta t) \cdot d(\Delta t)$$

unde:

$f(\Delta t)$  - densitatea probabilitatilor pentru  $\Delta t$ .

$$(10.20.) \quad m = \Delta t \cdot F(\Delta t) \Big|_0^{\Delta t_{\max}} - \int_0^{\Delta t_{\max}} F(\Delta t) \cdot d(\Delta t)$$

$$(10.21.) \quad m = \int_0^{\Delta t_{\max}} [1 - F(\Delta t)] \cdot d(\Delta t) =$$

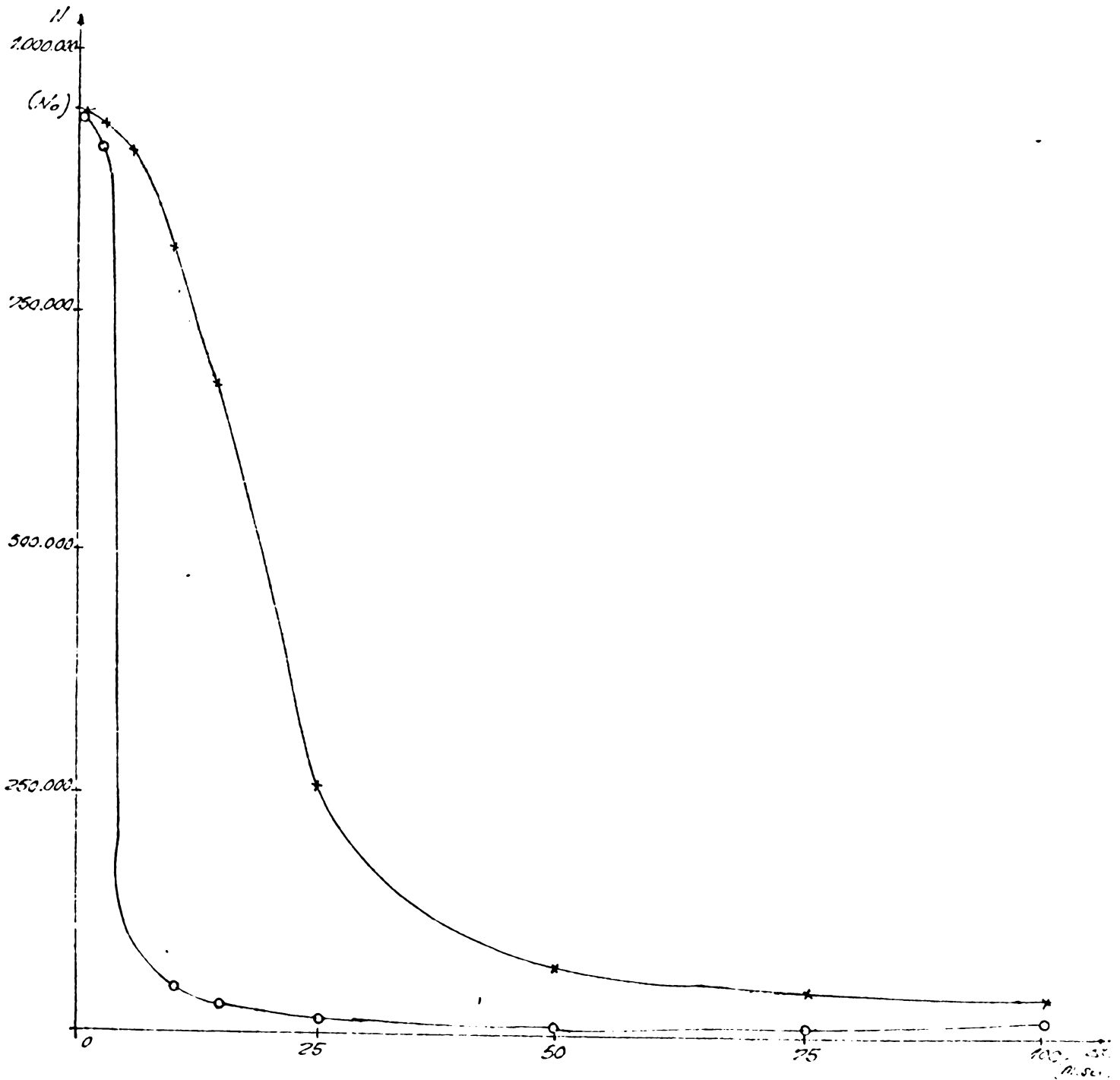
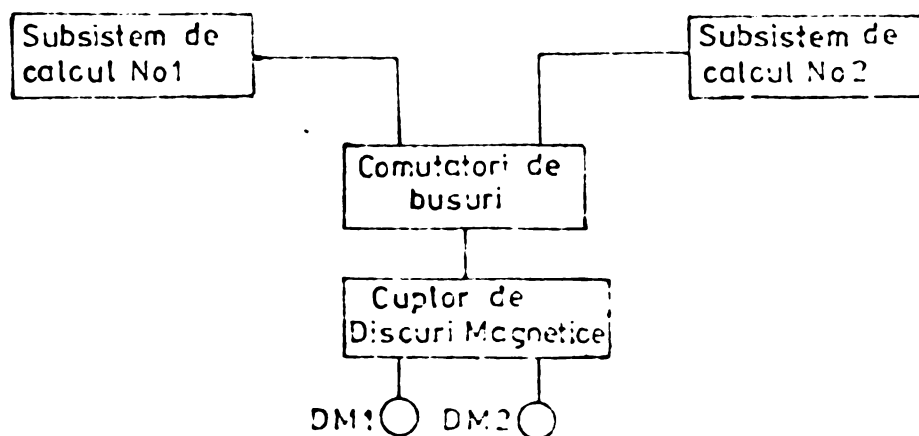


Fig. 10.5.

Fig. 10.6. Schema bloc a configuratiei.



$$= \frac{1}{N_0} \int_0^{\Delta t_{\max}} N \cdot d(\Delta t)$$

deci:

$$(10.22.) \quad m = \frac{S(G)}{N_0}$$

unde:  $S(G)$  = suprafata cuprinsa intre curba  
 $N = G(\Delta t)$  si axele de coordonate.

$$N_0 = G(0)$$

Pe curbele din fig.10.3. si 10.4. un careu de  $1 \text{ cm}^2$  are valoarea "s":

$$s = 50.000 \text{ puncte} \times 5 \text{ msec.} = 250.000 \text{ puncte.msec.}$$

Valoarea initiala pentru toate curbele este:

950.000 puncte.

Suprafata S cuprinsa intre curbe si axele de coordonate este data in tabelul 10.3.

TABELUL 10.3.

	Acces ocupat		Acces neocupat	
	un disc	doua discuri	un disc	doua discuri
$S(\text{cm}^2)$	208,8	158,7	24,6	92,2

Valoarea medie a perioadelor fara acces este:

$$(10.23.) \quad \frac{1}{\alpha} = S \cdot \frac{1}{N_0}$$

Valoarea medie a perioadelor de acces:

$$(10.24.) \quad \frac{1}{\beta} = S \cdot \frac{s}{N_0}$$

In tabelul 10.4. se gasesc rezultatele calculelor efectuate cu formulele 10.23 si 10.24

TABELUL 10.4.

	Acces ocupat		Acces neocupat	
	un disc	doua discuri	un disc	doua discuri
1	54,947	41,758	-	-
1	-	-	6,483	24,269

In cazul utilizarii unei singure unitati de discuri magnetice vom avea:

$$(10.25.) \quad \gamma = \frac{\beta}{\alpha} = \frac{6,483}{54,947} = 0,118$$

$$(10.26.) \quad \xi_2 = \frac{1}{(\gamma + 1)^2 + 1} = 0,4444$$

$$(10.27.) \quad E_2 = 2 \cdot \xi_2 = 0,8888$$

$$(10.28.) \quad k_2 = \frac{1}{1 - \xi_2} = 1,8$$

Daca se utilizeaza doua unitati de discuri magnetice obtinem:

$$\gamma = 0,5812 ; \quad \xi_2 = 0,2857 ; \quad E_2 = 0,5714 ; \quad k_2 = 1,4.$$

Se observa ca in cazul utilizarii unei UDM suplimentare randamentul intregului sistem creste. Daca la utilizarea unei singure UDM coeficientul mediu de alungire a timpului de executie  $k_2 = 1,8$  arata o sporire cu 80% a timpului de lucru, la sistemul biprocesor fata de sistemul monoprosesor, odata cu introducerea celei de a doua UDM sporul va fi de numai 40% ( $k_2 = 1,4$ ).

Numarul mediu de procesoare in asteptarea unui acces  $E_2$  in cazul folosirii unei singure UDM este de 0,88 ceea ce arata ca SDM este un element limitativ, cu toate ca s-a utilizat un echipament periferic performant, cu o capacitate de stocare mare si cu timpi de acces la informatie redusi. Odata cu adaugarea celei de a doua UDM se observa o imbunatatire a valorii  $E_2$ , configuratia respectiva fiind convenabila atat din punct de vedere functional, cit si economic. Trebuie avut in vedere si faptul ca masuratorile s-au facut in conditii speciale care au intensificat la maximum lucrul cu SDM.

#### 10.2.2. Etapa II-a

S-a utilizat o configuratie formata din doua sisteme de calcul I-100, un comutator de busuri lucrind pentru cererile de acces dupa principiul FIFO (prima cerere de acces venita, este prima servita) si acelasi SDM utilizat la masuratorile din etapa I-a. Configuratia este prezentata in fig.10.6.

Pentru masuratori s-a utilizat acelasi lant de programe software utilizat si in etapa I-a care a fost rulat concomitent in paralel pe ambele sisteme cronometrindu-se timpul final de

executie al lantului in configuratie. Rezultatele masuratorilor sint trecute in tabelul 10.5.

TABELUL 10.5.

	un disc	doua discuri
timpul total (minute)	216	168
coeficientul k	1,8	1,4

Se observa o concordanta a masuratorilor si calculelor efectuate in cadrul etapei I-a, cu masuratorile efectuate in etapa II-a.

Pentru a observa o imagine reala asupra posibilitatii utilizarii SDM la configuratiile biprocesor s-au facut incercari suplimentare la care lantul de programe nu a mai fost modificat pentru a incarca in mod maxim accesele la SDM, ci s-a lucrat in modul obisnuit cu sistemele de calcul (cu utilizarea tuturor resurselor sistemelor; benzi magnetice, imprimante, etc.). Rezultatele masuratorilor sint trecute in tabelul 10.6.

TABELUL 10.6.

Doua discuri	un sistem	doua sisteme
timpul total (minute)	120	132
coeficientul k	1	1,1

Se observa o alungire a timpului total de lucru cu 10%, ceea ce demonstreaza posibilitatea lucrului cu un singur SDM in configuratiile biprocesor. Aceasta constatare arata utilitatea studierii utilizarii cuplarii UDM simultan la doua sau mai multe sisteme de calcul.

### 10.3. Concluzii.

Cu toate ca masuratorile nu s-au putut efectua decit pe o configuratie biprocesor, calculele teoretice verificate in practica pentru aceasta configuratie, ne permit sa extindem valabilitatea acestora si la configuratiile multiprocesor.

La configuratiile cu sisteme I-100 (echivalente FDP11/34) unde se utilizeaza UDM fabricate la ROM CONTROL DATA cu capacitatea de 58 Moct. este convenabila constituirea SDM unde mai

multe procesoare pot utiliza simultan UDM, solutia prezentind pe linga avantajele economice imediate si o imbunatatire a fiabilitatii generale a configuratiei, deoarece defectarea unei UDM nu duce la oprirea intregului sistem cum se intimpla la sistemele cu o singura UDM. Astfel de configuratii cu SDM comune pentru mai multe procesoare pot fi utilizate in configuratiile cu rezerva calda ("stand-by") unde se cere o disponibilitate a sistemului de 24 ore din 24 de ore.

#### CAP.11. CUPLAREA SIMULTANA A UDM LA MAI MULTE SISTEME DE CALCUL IN CONTEXTUL MODER- NIZARII ARHITECTURII CONFIGURATIILOR DE CALCUL

La sistemele moderne de calculatoare problema intrarii-iesirii se poate trata pentru sistemele biprocesor sau multiprocesor prin preluarea tuturor sarcinilor de intrare-iesire de catre un procesor component al configuratiei, iar la configuratiile singulare prin realizarea unui subsistem cu o independenta functionala cit mai avansata.

Elementele Sistemului de Intrare-Iesire din S.O. sint:

- dispozitive de intrare-iesire
- numere logice
- drivere de intrare-iesire
- cereri de intrare-iesire
- sistemul de gestiune a fisierelor.

Se doreste introducerea majoritatii acestor elemente in structura interna a SDM, eliberindu-se memoria operativa de aceasta parte rezidenta a S.O.

Pe linga dispozitivele de intrare-iesire fizice, exista dispozitivele de intrare-iesire logice si pseudodispozitivele de intrare-iesire. Denumirea dispozitivelor logice poate fi aceeaasi cu cea fizica sau sa se dea o denumire oarecare nominalizata de fiecare proces in parte. In acest sens in SDM va fi prevazuta o tabela cu corespondenta fizica-logica pentru fiecare proces. Pseudo-dispozitivele de intrare-

iesire sint nume de dispozitive ce nu corespund cu nici un dispozitiv fizic din sistem, corespondenta fiind realizata prin operatii de redirectare. Existenta acestor dispozitive (pseudo-dispozitive) este utila in cadrul sistemului pentru stabilirea unor dispozitive de intrare-iesire implicite si pentru eliminarea functionarii cu randament scazut a sistemului datorita nefunctionarii unor dispozitive fizice. Un exemplu de pseudo-dispozitiv poate fi "listing consola", care prin redirectare poate fi folosit de anumite procese pentru salvarea informatiilor pe o UDM, sau pur si simplu pentru listarea acestora pe o imprimanta.

Interfata de comunicatie intre procesele utilizator si dispozitivele fizice de intrare-iesire este constituita de numerele logice. Fiecarui dispozitiv de intrare-iesire i se ataseaza un numar logic, fiecare proces stabilindu-si o corespondenta proprie intre numerele logice si dispozitivele fizice, corespondenta ce se poate modifica in orice moment, asigurindu-se astfel o independenta a proceselor fata de dispozitivele de intrare-iesire. Pentru fiecare proces va exista o tabela de numere logice, de lungime variabila care se va plasa in antetul procesului.

Drivererele de intrare-iesire indeplinesc urmatoarele functii: prelucreaza intreruperile externe generate de dispozitivele periferice respective, initiaza operatiile de intrare-iesire, genereaza terminarea anormala a operatiile de intrare-iesire si efectueaza anumite functii specifice echipamentelor periferice aferente. Drivererele sint in prezent parte integranta a Monitorului, putind apela o serie de rutine ale acestuia si putind fi apelate de acesta pentru: tratare intreruperi, initiere intrare-iesire, depasire timp (time-out) alocat echipamentului, terminare intrare-iesire si revenire din avarie. La efectuarea operatiilor de intrare-iesire driverul foloseste structuri de date in concordanta cu interfata dispozitivului de intrare-iesire.

Cererile de intrare-iesire sint utilizate cind se solicita efectuarea unor operatii de intrare-iesire. Ele sint servicii sistem, apelate de procese in scopul realizarii unei multiplexari a utilizarii dispozitivelor periferice de mai multi utilizatori simultan. Cererile se gasesc plasate in sirul de intrare a driverelor in



functie de prioritatea pe care o are procesul respectiv.

Sistemul de gestiune a fisierelor este plasat la nivelul superior al sistemului de intrare-iesire, permitind accesul la configuratia de intrare-iesire data, independent de echipament. Plasarea acestui element in cadrul SDM este mai dificila, fiind posibila numai la S.I.I. Celelalte elemente inasa pot fi foarte bine asimilate la nivelul SDM. Plasarea acestor elemente in cadrul SDM duce la micșorarea monitorului si implicit la o mai buna utilizare a memoriei operative prin diminuarea spatiului ocupat si eliberarea accesului de o serie de cereri.

SDM pe linga sarcinile clasice indeplinite de cuplor privind transferul de informatii intre sistemul de calcul propriu-zis si UDM, va trebui sa fie dezvoltat pentru a rezolva noile sarcini. Realizarea de SDM microprogramate utilizand microprocesoare si avind o memorie interna cu o capacitate care pe linga stocarea microprogramelor de utilizare generala in transferul de date pot permite si stocarea unor rutine ale sistemului de intrare-iesire devine posibila datorita dezvoltarilor tehnologice actuale. Rezolvarea problemei duce la aparitia unei intrebări de fond: ce se va intimpla cu S.O.? Exista doua cai de solutionare si anume:

- S.O. ramine neschimbat si hardware-ul va fi realizat in asa fel incit schimbarile survenite sa fie "transparente" pentru S.O. Solutia duce la complicarea hardware-ului si nu vizeaza ridicarea performantelor de ansamblu ale sistemului.

- S.O. este adaptat schimbarilor aduse, incercandu-se optimizarea secventelor de dialog si schimburi de informatii intre noul S.O. si SDM. Solutia este recomandata la realizarea unor sisteme de calcul noi, cu asigurarea software aferenta, nefiind recomandata pentru sistemele de calcul ce rezulta ca o dezvoltare a unor sisteme mai vechi, deoarece se pierde compatibilitatea la nivel de familie de sisteme.

Rularea subprogramelor de intrare-iesire in cadrul SDM alaturi de microprogramele specifice cuplorului, ridica problema sincronizarii acestora, metoda sincronizarii cu semafoare fiind una din caile eficiente de solutionare a problemei.

SDM devenind un subsistem autonom, cu independenta avansata hardware si software, fiind o sursa de informatii de care poate beneficia unul sau mai multe sisteme de calcul, deci fiind un S.I.I., permite o organizare mai buna a informatiilor pe UDM, precum si o imbunatatire a accesului la informatie. Se pot organiza tabele care sa cuprinda toate inregistrarile aferente unui proces, aceste tabele fiind pastrate in conformitate cu un algoritm de utilizare, fisierele mai des utilizate eliminand fisierele care nu au mai fost utilizate in ultima perioada de timp. Numele fisierelor din tabela, asociate cu locul inregistrarii fizice, pot fi introduse intr-un dispozitiv care va compara permanent cererile de informatii cu amplasarea fizica momentana a ansamblului de capete de scriere-citire, dindu-se intietate fisierelor la care timpul de acces este minim. Dispozitivul poate fi denumit "cache de accese la fisiere", contribuind efectiv la optimizarea schimbului de date cu UDM.

In domeniul microprogramarii realizarea SDM care sa raspunda sarcinilor enumerate, presupune organizarea speciala a formatului microinstructiunii. La sistemele de calcul microprogramate emularea unui set oarecare de instructiuni nu presupune deplasari simultane cu mai mult de o pozitie a termenilor manipulati de o microinstructiune. Aceasta concluzie se bazeaza pe studiul setului de instructiuni ale calculatoarelor produse de firma DEC, de intreprinderea FCE si sisteme IMB sau din seria RIAD produse de diverse firme din tari socialiste. In cadrul SDM se desfasoara operatii de serializare-deserializare a informatiilor (si vice-versa), operatii de detectare si corectie a erorilor, operatii care presupun in general deplasari cu un numar mare de pozitii a termenilor. Pentru a creste eficienta de ansamblu a subsistemului, microinstructiunea specifica SDM va trebui sa permita comanda unor deplasari simultane cu un numar variabil de pozitii. Astfel se va aloca un cimp care sa permita inregistrarea numarului maxim de pozitii ce se doresc deplasate si un alt cimp care sa indice felul si sensul deplasarii.

O alta caracteristica a SDM modern este cerinta de a recunoaste in cadrul dialogului extern instructiunile con-

cise si puternice de intrare-iesire si de a rezolva prin microprogramare sarcinile interne de rutina. In acest fel se poate organiza un sistem ierarhic pe mai multe nivele, de rezolvare a tuturor sarcinilor SDM. Rezolvarea sincronizarilor si a interblocajului intern la nivelul microprogramarii duce la marirea sigurantei in functionare a subsistemului.

Utilizarea tehnicii microprogramarii permite o adaptare usoara a SDM la diversele tipuri de formate de inregistrare. In general se utilizeaza pentru inregistrari anumite formate fixe, dar pentru realizarea compatibilitatii cu alte sisteme de calcul poate apare necesitatea lucrului cu un tip de inregistrare special. Un alt impediment in cuplarea oricarui tip de UDM este tipul si valoarea semnalelor de interfata. Prin microprogramare se poate organiza secventa dorita a dialogului cu UDM. Valoarea semnalelor de interfata ale UDM trebuie sa fie adaptata cu valoarea semnalelor de interfata pe interfata interna a SDM. In general cuplarea nu pune probleme lucrindu-se cu semnale specifice TTL. Daca apare necesitatea cuplarii unui UDM cu alte nivele de semnal, cea mai ieftina metoda este realizarea unui adaptor de interfata. In conditiile unei arhitecturi moderne, care beneficiaza de ultimele realizari ale tehnologiei circuitelor integrate, in locul adaptorului de interfata se poate realiza o interfata analogica, la care nivelul semnalului sa fie programat si comandat dupa caz prin microinstructiunile SDM.

Utilizarea in cadrul SDM a unor memorii tampon cu capacitate de stocare compatibila cu capacitatea de stocare pe o pista a UDM folosita in subsistem, permite ridicarea performantelor subsistemului prin diminuarea timpului de asteptare mediu de ajungere la informatie, echivalent cu timpul in care se efectueaza o semirotatie a pachetului de dicuri. Astfel imediat dupa positionarea ansamblului de capete pe cilindrul de lucru se va incepe citirea informatiei de pe pista dorita cu incepere de la preambulul primului sector prezent, informatia fiind transferata in memoria tampon. In momentul atingerii inceputului blocului de informatie utila, se procedeaza la prelucrarea directa a acesteia pina in momentul in care se ajunge la informatia deja inregistrata in memoria tampon. Din acest moment informatia este preluata din memoria tampon, viteza de prelucrare fiind mai ri-

dicata fata de rata de transfer a informatiei de la UDM. Deasemenea UDM se libereaza fiind gata pentru o noua positionare. Utilizarea acestei scheme, in cazul pachetului de informatie cu lungime compatibila cu capacitatea de inregistrare a unei piste, poate duce la eliminarea timpului mediu de ajungere la informatie, timp de ordinul zecilor de msec. Prin aceasta metoda de optimizare a modului de ajungere la informatie, combinata cu alte metode deja analizate, randamentul SDM poate fi mult ameliorat. Memoria tampon poate fi comuna pentru toate UDM ale subsistemului, in acest caz putind aparea timpii de asteptare pentru utilizarea acesteia. De aceea mai eficienta este utilizarea unei scheme cu doua memorii tampon "in tandem" sau a unei memorii tampon cu dublu acces cu posibilitatea scrierii si citirii simultane la adrese diferite. in ambele cazuri in biblioteca de microprograme, va apare un microprogram de gestionare a memoriei tampon si de positionare a indicatorilor de stiva pentru scriere si citire.

In cazul SDM se poate realiza gestiunea datelor inregistrate pe UDM din componenta subsistemului precum si o evidenta a spatiilor neinregistrate. In acest scop la instalarea unui pachet nou de discuri, se utilizeaza un microprogram "de aducere la zi" a informatiilor din SDM referitoare la acest pachet de discuri. Informatiile se refera la numele fisierelor, spatiul ocupat si daca e cazul carui sistem de calcul apartine fisierul. In scopul simplificarii dialogului cu UDM, in partea de cuplor a SDM se pot memora pozitiile fizice reale ale subansamblelor de capete ale UDM din configuratie. Realizarea gestiunii datelor la nivelul SDM permite implementarea unor microprograme temporare de studiere a repartizarii cit mai eficiente a elementelor sistemului de intrare-iesire care sint acum in sarcina SDM.

## CAP.12. CONCLUZII .

In lucrarea de fata s-a analizat problema cuplarii simultane a UDM la doua sau mai multe sisteme de calcul. Cercetarile desfasurate pina in prezent in acest domeniu nu au reusit sa rezolve decit anumite aspecte si in gene-

ral s-a procedat la rezolvarea unor cazuri particulare impuse de necesitatea realizării unor configurații specifice. Evolutiv începându-se cu "dublul acces" s-a ajuns la configurațiile multiprocesor cu procesoare specializate de intrare-iesire și la configurații moderne cu subsisteme specializate de intrare-iesire. Evoluția concepției este strins legată de evoluția tehnologiei circuitelor integrate, creându-se posibilitatea trecerii unor funcții ale S.O. în hardware-ul specific.

Pe lângă punerea în evidență și sistematizarea aspectelor legate de cuplarea simultană, lucrarea cuprinde rezultatele măsurătorilor efectuate printr-o metodă originală de măsură asupra rezervelor sistemului, în cazul partajării accesului unui subsistem de către mai multe sisteme de calcul. Se studiază de asemenea posibilitatea optimizării timpului de acces la informația stocată pe UDM. Fenomenul de interblocaj specific pînă acum Sistemului de Operare, putînd deveni un factor perturbator în cadrul SDM este studiat detaliat propunîndu-se utilizarea unei metode dinamice de prevenire a fenomenului la nivelul SDM. Sporirea sarcinilor SDM ridică probleme speciale de sincronizare internă, rezolvarea fiind posibilă prin utilizarea sincronizării cu ajutorul semafoarelor.

Studiul detaliat al cuplării simultane a reliefat importanța unor factori și funcțiuni și a stabilit interdependențele respective. S-a ivit necesitatea definirii unor noțiuni noi cum au fost: SDM, modul virtual, capacitatea majoră.

În lucrare nu s-a urmărit proiectarea unor dispozitive care să permită cuplarea simultană a UDM la mai multe sisteme de calcul, dar se dau elementele necesare pentru demararea unui proiect care să ducă la o rezolvare eficientă a problemei. S-au desprins două domenii de interes: sistemele de calcul mari unde cuplarea simultană și în general problema intrării-iesirii se poate rezolva prin utilizarea unui subsistem dotat cu putere de calcul și care preia o serie de sarcini ale S.O. și sistemele multiprocesor unde un procesor poate prelua toate sarcinile de intrare-iesire.

Concluzia generală este că problema cuplării simultane a UDM la mai multe sisteme de calcul poate fi rezolvată în condiții de eficiență economică, este o pro-

blema de actualitate pentru configuratiile moderne si poate fi dezvoltata odata cu modernizarea arhitecturii sistemelor de calcul.

Contributiile originale aduse de autor in lucrare sint:

a). Prezentarea unitara a aspectelor teoretice si practice legate de cuplarea simultana a UDM la mai multe sisteme de calcul.

b). Analiza critica a modului de realizare a modulelor componente ale SDM si a locului SDM in cadrul arhitecturii configuratiilor de calcul.

c). Impactul dezvoltarii tehnologiilor din domeniul circuitelor integrate si a UDM asupra concepiei si noilor functii ale SDM.

d). Definirea unor noi notiuni necesare pentru o mai buna organizare a structurii hardware in vederea cuplarii simultane a UDM la mai multe sisteme de calcul.

e). Studiul analitic al interfetelor SDM in vederea realizarii unor interfete universale care se permita cuplarea simultana.

f). Realizarea unei interfete externe care sa permita realizarea unei configuratii multiprocesor beneficiara a unui SDM comun.

g). Organizarea "grupului unic" de UDM si a "modulului virtual", care permit utilizarea simultana a UDM de diferite capacitati.

h). Realizarea unor configuratii care permit cuplarea simultana, prin utilizarea unor subansamble cercetate si introduse in fabricatie de colectivele conduse de autor (sisteme minicalculator INDEPENDENT, UDM-DM29, bus intercalculatoare, sistem de comutare, cuplor intercalculatoare, ceasuri de garda).

i). Analiza strategiilor de alocare si alegerea unui model de alocare care sa permita implementarea sa pe un sistem de calcul in vederea alegerii unei variante optime prin modificarea simpla a parametrilor, utilizindu-se modelarea matematica.

j). Stabilirea unei metodologii de calcul a timpului de acces la informatia stocata pe UDM in vederea alegerii unor algoritmi de acces imbunatatiti.

k). Studiul fenomenului de interblocaj aparut la nivelul SDM si posibilitatile de prevenire.

l). Introducerea operatiilor de sincronizare cu semafoare a proceselor care se desfasoara simultan in cadrul SDM, pentru crearea conditiilor de utilizare paralela a suportului software propriu SDM.

m). Stabilirea locului SDM in cadrul multiprocesorului banalizat" si a configuratiilor ce pot fi realizate in acest context.

n). Analiza teoretica a problemei conflictelor de acces si verificarea practica a concluziilor teoretice prin imaginarea unei metode de masura si realizarea de masuratori pe un sistem monoprosesor cu un montaj de simulare a acceselor multiple si pe un sistem real biprosesor.

o). Elaborarea unei proceduri de stabilire in exploatare a coeficientului real de utilizare a SDM.

p). Prelevarea elementelor din S.O. care pot fi transferate la nivelul SDM, in vederea realizarii unui SDM cu o independenta functionala cit mai avansata si pentru eliberarea sistemului de calcul propriu-zis de o serie de sarcini de intrare-iesire.

r). Precizarea unor perspective de interes in domeniul operatiilor de intrare-iesire si de cuplare simultana a UDM la mai multe sisteme de calcul in special, precum si a S.I.I. in cadrul modernizarii arhitecturii sistemelor de calcul.

## PRESURTARI

AD	- Adaptor
ADM	- Acces Direct la Memorie
bpi	- bit per inch (cifra binara pe 2,54 mm)
c.b.	- cifra binara
CADM	- Cuplor cu ADM
CS	- Cuplor simplu (fara ADM)
CTD	- Cerere bus pentru transferul de date
DAB	- Dublu Acces Bicalculator
DAM	- Dublu Acces Monocalculator
DEC	- Digital Equipments Corporation - SUA
DM-29	- Unitate de Discuri Magnetice cu capacitatea de 29 Moct. (proiect ITC 1975)
FIFO	- First In First Out (Primul intrat, primul iesit)
I-100	- Minicalculatorul INDEPENDENT-100 (in fabricatie

	la ICE din 1979)
I-102F	- Minicalculatorul INDEPENDENT-102F (in fabricatie la ICE din 1982)
M	- Memorie
MB	- Multiprocesor Banalizat
PDP	- Programed Devices Processor
SATF	- Shortest Acces Time First (Cel mai scurt timp de acces, primul)
SBM	- Subsistem de Banda Magnetica
SDM	- Subsistem de Discuri Magnetice
SII	- Subsistem de Intrare-Iesire
SO	- Sistem de Operare
tpi	- tracks per inch
UC	- Unitate Centrala
UDA	- Unitate cu Dublu Acces
UDM	- Unitate de Memorie cu Discuri Magnetice
USM	- Unitate de Schimburi Multiple
VLSI	- Very Large Scale Integrated (integrat pe scara foarte larga)
<=	- mai mic sau egal
>=	- mai mare sau egal
==>	- implica

## B I B L I O G R A F I E

1. Comtre Corporation - "Multiprocessor and Parallel Processing". John Wiley Sons New York, London 1979.
2. "Orientarile generale cu privire la intocmirea planului cincinal de dezvoltare economico-sociala a Romaniei in perioada 1981-1985". Conferinta Nationala a P.C.R. din 7-9 decembrie 1977.
3. "Raport de Sons/Traitancs de la Telemecanique" - October 1973.
4. "Single - Chip Microcomputer Focus of Intence Price Battle". Electronic News vol 22.No.1157. Nov.1977
5. Alexander, Michael T., "Time Sharing Supervisor Programs".



6. Amdahl, G.M.,  
L.D.Amdahl,  
7. Ang, W.S.  
8. Arnot, F.R. si  
G.M.Oliver  
9. Arnoprogram, J.A. si  
altii  
10. Bard, Y.  
11. Bard, Y.  
12. Belady, L.A.  
13. Bell, C.G. si  
A.Newell  
14. Bell, C.G., C.Gordon  
si A.Newell  
15. Bentley, A.W.  
16. Betaque, N.E. si  
G.A. Gorry  
17. Brown, M.  
18. Bulgacov R. si M.Rosu  
19. Bulgacov R. si M.Rosu
- University of Michigan 1970.  
"Fourth - Generation Hardware".  
Datamation Jan.1967.  
"Common Element Key to Multiprocessor  
Architecture". Computer Design. Oct.1981.  
"Hardware Monitoring of Real-Time  
Computer System Performance". Digest of  
1971 IEEE International Computer Society  
Conferance 1971.  
"Modular Minicomputers Using Micropro-  
cessor". IEEE Transactions on Computers.  
Vol. C-29. No.2. Feb.1980.  
"Experimental Evolution of System  
Performance". IBM System Journal  
Vol.12.1973.  
"Performance Criteria and Mesurement  
for a Time-Sharing System". IBM.  
System Journal Vol.10.No.3. 1971.  
"A Study of Replacement Algorithms for  
a Virtual Storage Computer". IBM.  
System Journal. Vol.4.1966.  
"A Panel Session - Computer Structure  
- Past, Present and Future". Fall  
Joint Computer Conferance 1971.  
"Computer Structures: Readings and  
Examples", Mc.Graw Hill New York 1971.  
"FPLA Arbiter Concept Adapt to  
Aplication Needs". Computer Design  
June 1981.  
"Automating Judgmental Decision  
Making for a Serious Medical Problem".  
Menagement Science 17. No.8. Apr.1971.  
"Statistical Analysis of Non-Homogeneous  
Poisson Processes". Processes Conferance  
on Stochastic Point Processes.  
Ed. P.A.W. Levis, New York, Wiley.  
"Discurile flexibile ca periferic al  
sistemului de calcul FELIX".  
Sesiunea de Comunicatii Stiintifice a  
Institutului Politehnic "Traian Vuia" -  
Timisoara 1974.  
"Metode si dispozitive de masurare a

- preciziei de pozitionare a sistemului de capete magnetice din unitatile de discuri magnetice". Sesiunea de Comunicari Stiintifice a Institutului Politehnic Iasi 1972.
20. Bulgacov R. si M. Rosu "Realizarea compatibilitatii unitatilor de discuri magnetice cu interfața IBM la interfata calculatorului FELIX C-256". Sesiunea de Comunicari a Academiei R.S.R. Oct. 1973.
21. Burns, J.F. "Greater Throughput with Multiple Array Processors". Computer Design Sept. 1981.
22. Contrell, H.N. "Designing for Measurement". Digest of 1971 IEEE International Computer Society Conference 1971.
23. Coffman, E.G. si R.C. Wood "Interarrival Statistics for Time-Sharing Systems". Communication of the ACM 9. Jul. 1966.
24. Cohen, E. "Simetric Multi-Miniprocessor: A Better Way to Go". Computer Decis 16 - 20 Jan. 1973.
25. Cook, R.W. si M.J. Flynn "System Design of a Dynamic Microprocessor". IEEE Transaction on Computer Vol. C-19 No. 3. Mar. 1970
26. Conway, R.W. Wal. Maxwell L.W. Miller "Theory of Scheduling". Adison - Wesley 1967.
27. Crocus "Systemes d'Exploitation des Ordinateurs". Dunod 1975.
28. Denning. P.J. "Third Generation Computer Systems". Computing Surveys 3 - 4 Dec. 1971.
29. Denning, P.J. "Thrashing - Its Causes and Prevention". Fall Joint Computer Conference 33. 1968.
30. Dennis, J.B. "Program Structure in a Multiaccess Computer" MIT Project MAC Report MAC-TR-11.
31. Dijkstra, E.W. "Cooperating Sequential Processes in Programming Languages". Academic Press 1967.
32. Drummond, M.E. jr. "Evaluation and Measurement Technique for Digital Computer Systems" - Prentice Hall, New Jersey 1973.

- 33.ENSIMAG-AMC-AFLET "International Workshop on Computer Architecture". Grenoble. June 26-28th. 1973.
- 34.Ferdinand, A.E. "An Analysis of the Machine Interference Model". IBM Systems Journal Vol.10. No.2. 1971.
- 35.Finkel, R.A. si "Processor Interconection Strategies".  
M.H. Solomon IEEE Transactions on Computer Vol. C-29 No.5. May 1980.
- 36.Flores, I. "Computer Organisation". Prentice-Hall New Jersey 1969.
- 37.Flynn, M.J. "Some Computer Organisation and Their Effectiveness". IEEE Transactions in Computer Vol C-21. No.9. Sept.1972.
- 38.Freiberg, W. "Statistical Computer Performance Evaluation". Americ Press New York 1972.
- 39.Gibson, D.H. "Consideration in a Block-Oriented System Design. Summer Joint Computer Conference, 1967.
- 40.Gostelow, K.P. si "Performance in a Simulated Dataflow  
R.E.Thomas Computer". IEEE Transaction on Computer. Vol. C-29. No.10. Oct.1980.
- 41.Hanssmann. F., "Modeling for Computer Center  
W.Kistler si Planning". IBM Systems Journal  
H.Schulz Vol.10. No.4. 1971.
- 42.Hellerman H., "Digital Computer System Principle".  
Mc.Graw - Hill New York 1967.
- 43.Hellerman H., si "Throughput Analysis of Some  
H.J.Smith Idealized Input" "Output and Compute Overlap Configurations". Computing Surveys. Vol.2 No.2. 1970.
- 44.Herman, D.H. "A Computer Evaluation Tool". Data-mation Feb.1967.
- 45.Hoogland. A.S. "Magnetic Recording Storage".  
IEEE Transactions on Computers Vol.C-255 No.12 Dec.1976.
- 46.Hoogland. A.S. "Mass Storage - Past, Present and Future" Computers, 6 - 8 sept.1973.
- 47.Hodges, D.A. "Trends in Computer Hardware Technology". Computer Design Vol.15 No.2.

- Feb.1976.
- 48.Husson, S.S. "Microprogramming Principles and Practices". Prentice Hall inc. Englewood Cliffs New Jersey 1971.
- 49.I-100 Documentatia tehnica a minicalculatorului INDEPENDENT 100". 1977 - 1980.
- 50.Jafari. H.,  
T.G.Lewis si  
J.D.Spragins "Simulation of a class of Ring-Structured Networks". IEEE Transactions on Computers Vol. C-29. No.5. May 1980.
- 51.Jenkins. J.H. si  
J.A.Howard "A Performance Matrix for Evaluating Control-Unit Design" IEEE Transactions on Computers Vol. C-29 No.4. Apr.1980.
- 52.Joulard si Brand "System D'exploitation des ordinateurs". Cours la Universitatea din Grenoble 1981-1982.
- 53.Katz, J.H. "Simulation of Multiprocessor Computer System". Spring Joint Computer Conference 1966.
- 54.Kermeni.F. si  
L.Kleinrack "A Tradeoff Study of Switching Systems in Computer Communication Networks". IEEE Transactions on Computers Vol. C-29. No.12. Dec.1980.
- 55.Kolence, L.W. "A Software View of Measurement Tool" Datamation Jan.1971.
- 56.Lake A., Hermanl si  
Kraley M. "Flexible Processor Extends" Design Options" Computer Design Nov.1981.
- 57.Lewin D. "Theory and Design of Digital Computers" William Clowes Limited London 1972.
- 58.Lewins, P.A.W. si  
P.C.Yuc "Statistical Analysis of Program Reference Paterns in a Paging Enveronment". IEEE Transaction on Computers sept.1971.
- 59.Little, D.D.C. "A Proof of the Queking Formula  $L=a.S$ ". Operations Research 9,3 1981.
60. Loris H. "Parallelism in Hardware and Software". Prentice Hall inc. Englewood Cliffs New Jersey 1972.
- 61.Lucas H.C.Jr. "Performance Evaluation and Monitoring". Computer Surveys Vol.3. No.3.

- 1971.
62. Lum V.Y., H.Ling si "Analysis of a Complex Data Manage-  
M.Esenco ment Acces Method by Simulation  
Modeling". AFIPS Conference Proc-  
eedings. Vol.37. 1970.
- 63.Margolin, B.H, "Analysis of Free - Storage Algorithms  
R.F.Parmelee si IBM System Journal  
M.Schatzaff Vol.10. No.4. 1971.
- 64.Maltson R.L., J.Gecsei "Evaluation Techniques for Storage  
D.R.Slutz si Hierarchis". IBM System Journal  
J.L.Taiger Vol.9. No.2. 1970.
- 65.Mc.Grew, M.J. "Measuring for Design". Digest of  
. 1971 IEEE International Computer  
. Society Conferance 1971.
- 66.Meinadier, J.F. "Structure et Founctionnement des  
Ordinateurs" Librairie Larousse  
1971.
- 67.Miller, J.S. si altii "Multiprocessor Computer System  
Study Intermetric iuc. Cambridge  
Mass. May 1966.
- 68.Nadir.J. si "Bus Arbiter Streamlines Multipro-  
B.Mc.Carmick cessor Design" Computer Design  
Jun.1980.
- 69.Padua D.A., D.J.Kuck "High-Speed Multiprocessor and  
si D.H.Lawrie Compilation Technique"  
III Transactions on Computers  
Vol. C-29 No.9. Sep. 1980.
- 70.Pradhan, D.K si "A Uniform Representation of  
K.L.Kodandapani Single and Multistage Interconnect-  
ing Network Used in Single  
Instruction Multiple Data Stream  
(SIMD) Machines".  
IEEE Transaction on Computers  
Vol. C-29 No.9. 1980.
- 71.Rauscher T. si "Multiprogramming: A Program and  
P.Adams Survey of Recent Developments",  
IEEE Transactions on Computers  
Vol. C-29 No.1. Sept. 1980.
- 72.Reiner H. si Z.Rodnay Workshop on the "Microarchitecture  
of Computer Systems". North-Holland  
American Elsevier Nice June 23-25th  
1975.

- 73.Rogojan A. "Curs de calculatoare numerice"  
Vol.1 si 2 Ed.I.P.Timisoara 1973.
- 74.Rony.F.R. "Interfacing Fundamentals:  
Conditional I/O Using Two Micro-  
computers". Computer Design Aug.  
1980.
- 75.Rony P.R. "Interfacing Fundamentals" Bidirec-  
tional I/O Using Two Semaphores".  
Computer Design Apr.1981.
- 76.Rony F.R, "Interfacing Fundamentals: 2 wire  
Handshake Using Two Microcomputers".  
Computer Design. Jun.1981.
- 77.Rosu M. si  
Bulgacov R. "Consideratii asupra utilizarii  
motoarelor lineare in echipamentele  
periferice din configuratia calcula-  
toarelor numerice". Sesiunea de  
Comunicari Stiintifice a Institutului  
Politehnic Iasi 1972.
- 78.Rosu M. si  
Bulgacov R. "Criterii de alegere a memoriilor  
externe" Automatica si Electronica  
Nr.5 Oct.1973.
- 79.Rosu M. si altii "Cu privire la tehnica masurarii unor  
parametri fizico-mecanici si elec-  
trici". Metalurgia nr.3 1964.
- 80.Rosu M. "Matrice decodificatoare" Buletin  
tehnic MFA Nr.3 1966.
- 81.Rosu M. si  
Bulgacov R. "Noi sisteme de servoreglare a moto-  
rului de actionare a dispozitivului  
de pozitionare a sistemului de cape-  
te din unitatile de discuri magnetice".  
A 3-a Conferinta Internationala a  
Electricienilor Bucuresti. Sept.1972.
- 82.Rosu M. si altii "Unitatea de legatura cu acces optim  
pentru unitatile de discuri de mare  
capacitate". Sesiunea de Comunicari  
Stiintifice a Institutului Politehnic  
Timisoara. Mai 1974.
- 83.Samsen, G.R. si  
R.D.Hudson "Bus Adapter Simplifies Interprocessor  
Communication". Computer Design Dec.1980.
- 84.Schwetman, H.D. "A Study of resource Utilisation and  
Performance Evaluation of Large Scale  
Computer System". TSN-12 Computation

- Center The University of Texas at Austin Jul.1970.
85. Shedler G.S. si  
S.C. Yang "Simulation of a Model of Paging System Performance". IBM System Journal. Vol.10. No.2. 1971.
86. Shore J.E. "Second Thoughts on Parallel Processing". IEEE INTERCON Proc.1972
87. Siegel, H.P. "The Theory Underlying the Partitioning of Permutation Networks". IEEE Transaction on Computers Vol. C-29 No.9. Sept.1980.
88. Smith, T.B. "A Highly Modular Fault - Tolerant Computer System". Massachusetts Institute of Technology 1970.
89. Spencer, F.R. "Interaction of VLSI Technology Progress with Minicomputer Product Development". IEEE Journal of Solid-State circuits Vol.Sc.14. No.2 Apr.1979.
90. Stang H. si  
P. Sonthgate "Performance Evaluation of Third Generation Computing Systems" Datamation Nov.1969.
91. Strauss. J.C. "A Simple Thrupt and Response Model of EXEC under Swapping Saturation". AFIPS Conferance Proceeding Vol.39 1971.
92. Traiger. I.L. si  
R.L. Mattson "The Evaluation and Selection of Technologies for Computer Storage Systems, Magnetion and Magnetic Materials". AIP Conferance Proceedings No.5. Part. 1.1979 1 - 9.
93. Tsav, R.F. si  
B.H. Margolin "A Multi-Factor Paging Experiment" American Press New-York 1971.
94. Walter, C.J.  
A.B. Walter si M.J. Bohl "Impact of Fourth Generation Software on Hardware Design". IEEE Computer Group News. Jul.1978, 1-10.
95. Warner. C.D. "Monitoring A Key to Cost Efficiency" Datamation. Jan.1971.
96. Weityman C. "Aerospace Computer Technology Catch up with Ground Gear". Electronics Sept. 1972.
97. Wood D.C. si  
E.H. Forman "Throughput Measurement Using a Synthetic Job Stream: AFIPS Conferance Proceeding Vol.39. 1971.

98. Wood, F.E. Jr. "Interconnection of Processor and Memory in the Multiprocessor System". ERC Memo. Feb. 1968.
99. Zaccoli, M.P. si "Rapid Bus Multiprocessor Systems" A.C. Sanderson Computer Design Nov. 1981. 4