

p. constantinescu

n. zaharia

inițiere în

organizarea și proiectarea

sistemelor de conducere

cu mijloace de automatizare

08712/Volant 42



2505956

D 008712, Vol. 1P42

editura tehnică

Paul Constantinescu
Nicolae Zaharia

Inițiere în organizarea și prelucrarea sistemelor de conducere cu mijloace de automatizare.

Analizarea sistemelor informaționale existente și proiectarea unor moderne, prin folosirea sistemelor de prelucrare a datelor cu calculatoare electronice (SPAD) constituie problematici prioritare în economia noastră.

Lucrarea de față este o primă încercare a unor specialiști români, de prezentare în manieră originală a cunoștințelor necesare pentru pregătirea proiectărilor sistemelor informaționale dotate cu mijloace de automatizare, respectiv a problematicilor actuale privind organizarea datelor, structura și funcționarea fișierelor, tipurile de sisteme informaționale, cu exemplificări și cu indicații de implementare a acestora.

Paul Constantinescu
Ion Nicolae Zaharia

Inițiere în organizarea și proiectarea sistemelor de conducere cu mijloace de automatizare (S.I.P.A.D)

SERIA DE INITIERE

- Donald Cutler*
Inițiere în programarea calculatoarelor
Traducere din limba engleză — S.U.A.
- John S. Murphy*
Inițiere în calculatoare numerice
Traducere din limba engleză — S.U.A.
- John S. Murphy*
Inițiere în programarea calculatoarelor numerice
Traducere din limba engleză — S.U.A.
- T. D. Truitt, A. E. Rogers*
Inițiere în calculatoare analogice
Traducere din limba engleză — S.U.A.
- R. R. Arnold, H. C. Hill, A. V. Nichols*
Inițiere în prelucrarea datelor
Traducere din limba engleză — S.U.A.
- J. Shubin*
Inițiere în conducerea întreprinderilor
Traducere din limba engleză — S.U.A.
- E. Vasiliu*
Inițiere în dispozitivele semiconductoare
- D. Stanomir*
Inițiere în electroacustică
- W. Trusz*
ABC-ul reparării radioreceptoarelor
Traducere din limba polonă
- A. Popa*
ABC de protecția muncii
- Stelian Niculescu*
Inițiere în FORTRAN (sub tipar)
- Margareta Drăghici*
Inițiere în COBOL (sub tipar)
- I. Crețu*
Inițiere în estetica produselor (în pregătire)

Dr. Paul Constantinescu
Ing. Ion Nicolae Zaharia

**Inițiere în
organizarea și proiectarea
sistemeelor de conducere
cu mijloace de automatizare
(S.I.P.A.D.)**

EDITURA TEHNICĂ
București — 1972



Prefață

Această carte apare într-o etapă în care se restabilește importanța pregătirii unor cadre din întreprinderi în domeniul sistemelor informaționale față de pregătirea economică sau tehnologică, relativ mult mai bine reprezentată atât în literatura de specialitate, cit și în programele învățământului superior din țara noastră. Obiectul principal al cărții este prezentarea unor elemente și metode utilizate în domeniul organizării datelor (fișierelor) și al proiectării sistemelor informaționale, care nu au făcut încă obiectul unor lucrări de sinteză în literatura originală din țară. În acest sens, cititorul este avertizat că parcurserea cu atenție a primelor patru capitole, în care se prezintă noțiunile de bază la un nivel de generalitate accesibil și nespecialiștilor în domeniu, constituie o pregătire necesară înțelegerei proiectelor sistemelor informaționale. Prezentarea în continuare a unor subsisteme informaționale proiectate și a unor aspecte principale ce se frâmîntă în problematica acestui domeniu, constituie o bază în ceea ce privește pregătirea necesară unor analiști de sisteme și a conducerilor din întreprinderi. Această parte (cap. 5 și 6) reflectă, în special, experiența acumulată de colectivul de specialiști din cadrul unității de calcul de la CEPECA în activitatea pe care a desfășurat-o în colaborare cu uzinele pilot pentru introducerea sistemelor de prelucrare automată a datelor în întreprinderi (1968—1971).

In carte se prezintă unele din cele mai frecvente probleme de „tip conducere“ ce se formulează și trebuie rezolvate și integrate în cadrul sistemului informațional care se proiectează într-o întreprindere, în vederea elaborării sistemului de conducere pe care îl pretind factorii de decizie implicați. Sistemul informațional, ca totalitate a mijloacelor și procedurilor ce fac legătura între procesul de execuție și cel de decizie, trebuie să furnizeze informațiile sau variantele necesare astfel încât conducerea la diferite nivele să poată lua decizii cit mai bune. Sistemul de conducere căruia sistemul informațional îi furnizează

Redactori : ing. SMARANDA DUMITRIU
și ing. PAUL ZAMFIRESCU

Telinoredactor : MARIA IONESCU

zează operativ și complet informațiile necesare, poate fi automatizat parțial (în special pentru decizii de rutină), o parte dintre decizii putind fi elaborate automat de sistemul de prelucrare automată a datelor pe baza unor modele corespunzătoare. Sarcina de bază a specialiștilor care proiectează sistemul de conducere și cel informațional este de a analiza în cadrul sistemului informațional cerințele sistemului de conducere.

Ca o concluzie generală a activității pe care o desfășoară specialiștii din acest domeniu în cadrul întreprinderilor, putem formula următorul principiu pe care acestia trebuie să caute să-l aplique în introducerea sistemelor informaționale care folosesc prelucrarea automată a datelor în întreprinderi: ieșiri pentru a pregăti intrările.

Formularea aceasta concentrată, în afară de principiul de bază al proiectării care constă în faptul că rapoartele cerute de conducere sunt cele care determină intrările și prelucrările, descrie faptul că pe măsură ce se prezintă conducerii întreprinderii rezultate obținute pe sistemul de prelucrare automată a datelor, cu atit participarea acestia și deci a întreprinderii pentru introducerea noului sistem va crește prin buna pregătire a datelor de intrare, organizarea lor, precizarea cerințelor sistemului de conducere, dezvoltarea metodelor de prelucrare etc.; obținându-se în condiții mai bune varianțele pentru decizie și rapoarte periodice prin excepție în care se prezintă conducerii abaterile de la funcționarea normală sau cea planificată, sau modul de variație al unor indicatori sintetici, aceasta poate, în măsură din ce în ce mai mare, să-și concentreze atenția asupra elementelor cheie, simțind tot mai mult utilitatea acestui nou tip de unealtă, care este sistemul de prelucrare automată a datelor.

Eșalonarea corespunzătoare a introducerii sistemului informațional, astfel ca să se poată prezenta conducerii unele rapoarte utile încă din fazele de inceput ale activității, constituie un factor important, favorabil intregii acțiuni.

In carte se folosește, în special, termenul de sistem de prelucrare automată a datelor (SPAD) și nu cel de calculator electronic, tocmai pentru a se sublinia necesitatea utilizării unui grup complex de echipamente electronice în rezolvarea problemelor din întreprinderi (unități de benzi magnetice și unități de discuri pentru fișiere, echipamente terminale cu dispozitive de afișare, echipamente pentru pregătirea și colectarea datelor etc.) și pentru a se sublinia importanța software-lui (programelor) în cadrul sistemului.

Prescurtarea SIPAD utilizată în lucrare descrie sisteme informaționale folosind prelucrarea automată a datelor, ca atare se referă în special la sisteme informaționale, fără însă a exclude utilizarea în cadrul SIPAD a mijloacelor de mecanizare sau manuale; un obiectiv important al lucrării este tocmai precizarea necesității imbinării în mod armonios a mijloacelor manuale, de mecanizare și de automatizare chiar în cadrul aceluiași sistem.

* * *

Capitolele 1, 2 și 6 au fost elaborate de Paul Constantinescu, cap. 3 a fost elaborat de cei doi autori, cap. 4 și cap. 5 au fost elaborate de Nicolae Zaharia și redactate împreună de cei doi autori.

* * *

Apariția acestui material de sinteză poate fi considerată ca un îndemn și pentru alți specialiști de a elabora materiale care să suplimentească lipsa existentă.

Dr. PAUL CONSTANTINESCU

CUPRINSUL

<i>Bibliografie</i>	10
1. Introducere	11
1.1. Caracteristicile sistemelor de prelucrare automată a datelor (SPAD) din generația I-a	11
1.2. Caracteristicile sistemelor de prelucrare automată a datelor din generația II-a	14
1.3. Caracteristicile sistemelor de prelucrare automată a datelor din generația III-a și perspectivele de evoluție	17
1.4. Dezvoltarea metodelor de programare și modelare în raport cu necesitățile de organizare a datelor	24
2. Metode de organizare a datelor	42
2.1. Tehnici de organizare a fișierelor ; baze de date	42
2.2. Tipuri de fișiere și prelucrări ale fișierelor	51
2.3. Codificarea datelor	58
2.4. Controlul prelucrării și organizării datelor	63
3. Reprezentarea prelucrărilor prin scheme logice	68
3.1. Scheme logice de programare	68
3.2. Scheme logice de sistem	74
3.3. Exemple privind reprezentarea prelucrărilor	76
3.4. Operațiile principale folosite în schemele logice	88
3.5. Utilizarea schemelor logice pentru calculul timpului de încărcare al calculatorului electronic	96
4. Înregistrarea și identificarea informațiilor în fișiere	103
4.1. Metode de înregistrarea informației	103
4.2. Înregistrarea și regăsirea informației pe discuri	108
4.3. Structura fișierelor	111
4.4. Identificarea și validarea înregistrărilor	113
4.5. Descrierea fișierelor și a prelucrărilor cu ajutorul unui limbaj de proiectare	131

5. Descrierea SIPAD (Sisteme Informaționale cu Prelucrare Automată a Datelor) cu aplicații	145
5.1. Tipuri de aplicare ale SPAD în întreprinderi ; SIPAD integrat și SIC	145
5.2. Exemple de aplicare a SPAD (în industria textilă)	158
5.2.1. Probleme generale privind planificarea și urmărirea producției	158
5.2.2. Proiectarea documentelor (rapoartelor) furnizate de aplicația „urmărirea producției”	163
5.2.3. Proiectarea documentelor (formatelor de cartelă) cu date primare de intrare necesare obținerii rapoartelor de la pct. 5.2.2.	165
5.2.4. Descrierea prelucrărilor pentru „planificarea producției”	167
5.2.5. Descrierea prelucrărilor pentru „urmărirea producției”	184
5.2.6. Descrierea prelucrărilor cu limbajul de proiectare	193
6. Elemente, etape și probleme ale implementării SIPAD	211
6.1. Apariția SIPAD și tendințe generale de dezvoltare	211
6.2. Fazele introducerii SIPAD în întreprinderi	220
6.3. Rolul conducerii întreprinderii în introducerea SIPAD	232
6.4. Probleme relative la actuala etapă de introducere în țară a SIPAD	243
<i>Anexe</i>	
Anexele 1.1—1.6 (6 tabele) Formulare de rapoarte pentru urmărirea producției	257
Anexele 2.1—2.5. (6 tabele) Formulare de cartele de date	263
Anexele 3.1—3.9 (9 tabele) Formatul fișierelor pentru urmărirea producției	269

Bibliografie

1. Paul Constantinescu, **Introducere în programarea automată**, Editura tehnică, Bucureşti, 1967.
2. Daniel D. Mc Cracken, **A Guide to FORTRAN IV Programming**, John Wiley, New York, 1965.
3. Daniel D. Mc Cracken, **A Guide to COBOL Programming**, John Wiley, New York, 1963.
4. Robert S. Ledley, **Programarea și utilizarea calculatoarelor numerice**, Editura tehnică, Bucureşti, 1968.
5. Richard Bellman, **Adaptive Control Processes**, Princeton University Press, Princeton N.J., 1961.
6. Grigore Moisil, **Teoria algebrică a mecanismelor automate**, Editura tehnică, Bucureşti, 1959.
7. A. Markov, **Normalnie algoritmi**, Trudi Institutului Steclova, 1955.
8. National Computer Center of Great Britain Basic Systems Analysis Training Course, ESL Bristol, 1968.
9. James Martin, **Design of Real Time Computer Systems**, Prentice Hall, 1967.
10. L. Kujev, M. Goračinov, **Matematica și conducerea producției**, 1969, Editura tehnică, Bucureşti.
11. Anthony Chaudor, John Graham, Robin Williamson, **Practical Systems Analysis**, Rupest Hart-Davis Educational Publications, 1969.
12. S.E. Elmaghriby, **Proiectarea sistemelor de producție**, Editura tehnică, Bucureşti, 1968.
13. L. O. J. Bruss, **Previsions et decisions rationnelles**, Dunod, Paris, 1961.
14. J. Shubin, **Inițiere în conducerea întreprinderilor**, Editura tehnică, Bucureşti, 1970.
15. R. R. Arnold și a. **Inițiere în prelucrarea datelor**, Editura tehnică Bucureşti, 1969.
16. E. J. McCharty și a. **Sisteme integrate de prelucrarea datelor în conducerea activității economice**, Editura tehnică, Bucureşti, 1969.
17. M. R. Starr, **Conducerea producției. Sisteme și sinteze**, Editura tehnică, Bucureşti, 1970.
18. H. B. Maynard, **Conducerea activității economice**, Editura tehnică, Bucureşti, (1970), vol. I, (1971), vol. II (1972) vol. III, (în pregătire) vol. IV.
19. E. S. Savas, **Conducerea cu calculatoare a proceselor industriale**, Editura tehnică, Bucureşti, 1969.
20. I. Flores, **Sisteme de programe (software) pentru calculatoare numerice**, Editura tehnică, Bucureşti, 1969.
21. I. Flores, **Practica programării calculatoarelor**, Editura tehnică, Bucureşti, 1968.
22. M. Măneșeu, **Mașini de calcul pentru mecanizarea și automatizarea lucrărilor economice și administrative**, Editura tehnică, Bucureşti, 1966.
23. M. Dumitrescu și a. (coordonator) **Organizarea conducerii, producției și muncii**, Editura tehnică, 1969.

1. Introducere

1.1. Caracteristicile sistemelor de prelucrare automată a datelor (SPAD) din generația I-a

Sistemele de prelucrare automată a datelor (SPAD) sau calculatoarele electronice, cum se numeau la început, au apărut în deceniu al cincilea din secolul nostru din nevoie de a se executa rapid, în timp util, o serie de calcule. În vederea calculării traiectoriei unui obiectiv în mișcare, în vederea neutralizării lui, a fost necesar în timpul celui de al doilea război mondial să se realizeze dispozitive electronice de calcul sub formă de *registre*, în care să se inscrie datele numerice care să participe în calcule și între care să se execute cu mare viteză operațiile aritmetice. Dezvoltarea electronică prin utilizarea tuburilor electronice în radio, televiziune, automatizări etc., asigura în acea vreme posibilitatea realizării dispozitivelor aritmetice necesare efectuării cu mare viteză a operațiilor aritmetice de bază (adunare, scădere, inmulțire și împărțire). Elementele *bistabile* (alcătuite din două tuburi electronice, dintre care unul conduce și numai unul, în fiecare din cele două stări stabile) legate între ele, cîte circa 30, alcătuiau un astfel de registru în care se putea inscrie un număr transcris în sistemul binar, asociind unei stări stabile cifra zero și celeilalte stări stabile cifra 1 (cantitatea de informație reprezentată de o cifră binară este un *bit*). Transcrierea informațiilor (nu numai numerele, dar și literele sau alte simboluri care se intilnesc pe claviatura unei mașini de dactilografiat) în șiruri de cifre binare, în vederea executării de operații (aritmetice, logice, comparații etc.), a rămas o caracteristică de bază a SPAD de-a lungul evoluției acestora, caracteristică datorită faptului că tehnica și fizica în etapa actuală oferă în mare abundență elemente bistabile (contacte, tranzistoare, inele de ferită, prezență sau absență unei perforații făcute de o rază laser pe un mediu etc.).

În vederea aplicării consecvente a automatizării efectuării calculelor, trebuie ca factorul uman să participe în procesul de calcul în măsură tot mai redusă.

Pentru aceasta, în afară de registrele pentru inscrierea și efectuarea operațiilor era nevoie de un dispozitiv de memorare analog cu memoria omului, în care să se inscrie, înainte de începerea procesului de calcul, toate operațiile (instrucțiunile) ce trebuiau executate asupra datelor numerice. O sugestie importantă în acest sens o prezintau în acea epocă, de pildă, mașinile cu program care se foloseau încă din secolul trecut în industria textilă și a căror funcționare era comandată de perforațiile existente pe cartele al căror ansamblu constituia programul pentru realizarea unui anumit tip de țesături, desene etc. Memorile au fost organizate pe celule în care se inscriu siruri de biți (cuvinte de circa 20—40 biți), fiecare celulă (cuvint) având o anumită adresă.

Elaborarea dispozitivelor de memorie în care se inscriu programele de prelucrare, ca atare ansamblul instrucțiunilor pe care trebuie să le execute calculatorul electronic asupra datelor numerice care sunt de asemenea inscrise în memorie, a constituit actul de naștere al primei generații de sisteme de prelucrare automată a datelor (SPAD).

Caracteristicile primei generații de SPAD sunt următoarele :

a. Dispozitivele aritmetice și logice sunt executate din punct de vedere tehnologic cu tuburi electronice, ceea ce conduce la gabarite mari, viteză de calcul relativ redusă, siguranță mai mică în funcționare și consum relativ mare de energie electrică.

b. Dispozitivele de memorie se execută pe tambur magnetic (cilindri acoperiți cu strat magnetic, la suprafața cărora se află montate capetele de citire și de inscriere a informațiilor), având capacitate de stocare a informației de mii sau de cel mult zeci de mii de cuvinte (a circa 30 biți).

c. Pentru alcătuirea programelor se folosesc instrucțiuni scrise în codul mașinii, atât instrucțiunile cât și datele numerice organizate în cuvinte fiind *adresate*, ca atare fiind distribuite de către programator în memorie în diferite celule având adresele corespunzătoare. Programarea în cod *mașină* sau în limbajul *mașinii* permite ca fiecare instrucțiune să poată fi executată direct de calculatorul electronic, datorită circuitelor existente în blocul aritmetic format din registrele pentru efectuarea calculelor, *asupra cărora instrucțiunile în cod*

mașină acționează ca semnale de intrare (comandă). Operațiile aritmetice și logice care se execută în registrele blocului aritmetic se reduc la adunări, operații elementare de deplasare a sirurilor de cifre binare, schimbări ale cifrelor 0 în 1 și invers etc.

d. Funcționarea calculatoarelor din prima generație decurge astfel :

Pe un pachet de cartele sau role de bandă de hîrtie se află programul și datele numerice, informații care sunt materializate în perforațiile existente pe cartele sau bända de hîrtie.

Perforațiile aflate pe aceste suporturi sunt „citite“ de dispozitive de introducerea informațiilor (cîritoare de cartele sau bandă de hîrtie), care printr-un sistem de relee cu contacte sau un sistem optic transformă aceste perforații în impulsuri electrice, ce se transmit prin circuitele de la intrare în memorie, unde se materializează sub formă de dipoli magnetici, care își păstrează magnetizarea corespunzătoare impulsurilor cît timp este necesar, conform programului.

Informațiile din memorie (cuvintele alcătuite pe baza magnetizării dipolilor) sunt prelucrate conform programului în blocul aritmetic, luîndu-se succesiv de la diferite adrese din memorie instrucțiunile și datele numerice necesare. Instrucțiunile sunt decodificate de *blocul de control* și acționează ca impulsuri de comandă asupra registrelor blocului aritmetic. Pe măsura obținerii de rezultate intermediare sau finale, acestea sunt inscrise tot în memorie la diferite adrese, de unde sunt extrase la o mașină electrică de scris, caracter cu caracter, sau la o imprimantă care permite tipărirea simultană a unui sir de caractere, printr-un procedeu invers celui de la introducerea informațiilor : stările de magnetizare ale dipolilor generează prin capetele de citire impulsuri în circuitele de ieșire, care acționează sistemul de relee cu contacte și în ultimă instanță clapele claviaturii sau cilindrui pe care sunt caracterele de imprimat. Se observă că informațiile de la intrare, ca și cele ce vor fi tipărite la ieșire, trec prin memoria internă din unitatea centrală.

e. Dispozitivele de „intrare-ieșire“ pentru introducerea datelor (informațiilor) și extragerea rezultatelor sunt puțin diversificate la calculatoarele din prima generație.

f. Domeniul de aplicație al acestor calculatoare se limită în principal la calcule tehnico-științifice, ca atare la prelucrări care nu necesită înregistrarea în memorie a unor volume mari de date.

1.2. Caracteristicile sistemelor de prelucrare automată a datelor din generația II-a

Apărute ca instrument pentru executarea mai rapidă a calculelor tehnico-științifice, calculatoarele electronice din prima generație au permis precizarea unor aspecte esențiale care au condus la pregătirea rezolvării unor clase largi de probleme de importanță majoră în sfera producției.

În primul rînd, s-a constatat faptul că rezolvarea problemelor pe calculator se face printr-un proces de calcul în care se repetă diferite succesiuni de operații (cicluri de operații), existente în program. Prezența, în programele necesare rezolvării problemelor, a acestor cicluri de operații, explică pe de altă parte faptul că mașina execută milioane sau miliarde de operații pe baza unor programe în care nu sunt decât sute sau mii de instrucțiuni, explicând astfel economia de timp social necesar pregătirii rezolvării unei probleme, care în condițiile inscrierii în program a tuturor operațiilor pe care trebuie să le execute mașina ar conduce la consumarea unui timp social comparabil cu cel necesar rezolvării problemei cu mijloace manuale.

Acest principiu al existenței ciclurilor, domină toate programele care s-au elaborat pînă în prezent, prezența ciclurilor în programe fiind caracteristica de bază a acestora [1].

În al doilea rînd, aceste prelucrări repetitive prezente în programele care descriu calculele, au făcut ca domeniul de aplicare al calculatoarelor electronice să se extindă foarte repede și în primul rînd pentru acele aplicații în care caracterul repetitiv predomină. Clasa mare a prelucrărilor informațiilor necesare conducerii și organizării unităților industriale, bancare, comerciale etc., planificarea, ordonanțarea, lansarea și controlul producției, inventarul și gestiunea stocurilor, calculele finanță-contabile, evidența personalului, aprovizionarea, desfacerea etc., prezinta tocmai acest specific: repetarea de un număr mare de ori a unui număr relativ redus de operații asupra unui volum mare de date.

Calculatoarele din prima generație, avind o unitate centrală (bloc aritmetic, memorie internă) și dispozitive de intrare-iesire ca singure dispozitive periferice (față de unitatea centrală), nu permitea rezolvarea unor astfel de probleme, datorită absenței unor memorii de mare capacitate în care să se poată înscrie volumele mari de date necesare rezolvării problemelor economice din întreprinderi și care să permită o prelucrare rapidă a datelor înscrise în aceste memorii.

1.2. CARACTERISTICILE SISTEMELOR DE PRELUCRARE

Prin urmare, erau necesare noi dispozitive de memorie care să permită înscrarea *fișierelor de date* necesare rezolvării problemelor din întreprinderi (fișier de materiale, fișier de produse, ansamblu și repere, fișierul operațiilor tehnologice, fișierul de personal etc.).

Datele referitoare la un același element, se grupează de obicei într-o singură formă, ce se numește *înregistrare logică*. De exemplu, înregistrarea logică corespunzătoare unui material, într-o problemă de gestiune a stocurilor, va cuprinde, în afară de codul materialului respectiv, o serie de date (cîmpuri sau zone) ca: denumire, unitate de măsură, pret unitar, codul magaziei unde se stocă, nivelul stocului fizic și.m.d.

O mulțime de înregistrări (logice) se numește, deci *fișier*. Din exemplul de mai sus rezultă imediat din ce se compune un fișier de materiale. În mod similar într-o problemă de calcul ai salariilor va apărea fișierul de personal ca o mulțime de înregistrări (logice), fiecare din ele conținând ca zone: marca sau numărul matricol al muncitorului, numele și prenumele, funcția, locul de muncă, vechimea în cîmpul muncii, salariul tarifar de încadrare, numărul și vîrstă copiilor etc.

În general, se subînțelege că mulțimea de înregistrări (logice) este astfel definită încît grupează înregistrări similare sau, altfel spus, referitoare la un același tip de elemente. Astfel, înregistrările privitoare la materiale și cele privitoare la personal se grupează în fișiere separate. Pot apărea desigur și cazuri de fișiere definite ca mulțimi neomogene de înregistrări. De exemplu, putem considera un fișier de *tranzacții* (modificări ce se execută asupra înregistrărilor unui fișier), care poate cuprinde tranzacții din ateliere (avansul producției) și cele din magazii (intrări-iesiri din stoc) la un loc.

Fișierele pot avea diferite suporturi fizice. În ordinea apariției, primele fișiere au fost cele de *documente*. În general, fișierele de acest tip grupează documente de același fel. Apoi au apărut fișierele pe *cartele perforate*. Ca și fișierele de documente, cele pe carțile perforate sunt chiar prin natura suportului fizic, discontinue.

Evoluția tehnică a făcut, de asemenea, ca majoritatea dispozitivelor din prima generație să fie înlocuite cu dispozitive mai rapide, mai sigure în funcționare, cu un consum mai mic de energie electrică, iar fișierele să fie înregistrate pe *medii magnetice*.

Caracteristicile principale ale calculatoarelor electronice din generația a doua sunt următoarele:

- a. Înlocuirea tuburilor electronice cu tranzistoare în circuitele sistemului de prelucrare automată a datelor.

b. Realizarea memoriei principale (interne) din unitatea centrală cu ajutorul blocurilor de ferite, organizate tot pe cuvinte și adrese.

c. Extinderea memorii pe tambur magnetic și folosirea lor ca memorii periferice și în special crearea *memoriilor pe bandă magnetă*, a căror funcționare se bazează pe principiul magnetofonului; capacitatea unor astfel de benzi magnetice este de ordinul 10–20 milioane de caractere * (fiecare caracter având 6–8 biți) și având în vedere că benzile sunt amovibile, se poate realiza practic o bibliotecă de benzi oricără de mare, pe care să se poată înscrie fișierele necesare, fiecare bandă fiind introdusă în unitatea de benzi cînd este necesară prelucrarea.

d. Extinderea rapidă a clasei aplicațiilor și apariția scindării memoriei în memorie principală (în unitatea centrală) și memorie auxiliară ca bloc periferic (memoriile cu benzi magnetice sau tambur), a creat necesitatea elaborării unor limbaje de prelucrare care, pe de o parte, să fie accesibile unor *clase mai largi de specialiști* decit limbajul mașinii și pe de altă parte să permită *adresarea automată* a instrucțiunilor și datelor în memoria rapidă, adresare care la calculatoarele din prima generație era executată de programator în cadrul programului.

În elaborarea unor astfel de limbaje trebuie să se țină cont că mai mult de cunoștințele existente la diferite clase de specialiști. Acest deziderat s-a realizat prin faptul că limbajele de programare care au început să se elaboreze au avut ca model în principal limbile naturale și formalismul matematic [1]. Astfel s-au elaborat *limbajele algoritmice* de programare, dintre care în prezent o circulație mai mare o au limbajele ALGOL, FORTRAN și COBOL [2], [3].

Deoarece SPAD nu poate prelucra decit instrucțiunile în cod mașină pentru care are proiectate corespunzător circuitele unității centrale, este necesar ca programele scrise în limbajele algoritmice să fie traduse în limbajul mașinii. Această traducere este realizată cu ajutorul unor *program translator* (compilator) care traduce orice program scris în limbaj algoritic într-un program scris în limbaj mașină. În același timp, trebuie ca adresarea celulelor memoriei principale să se facă automat, deoarece complexitatea programelor crescind, complexitatea tehnică a memoriei crescind, de asemenea, nu se mai poate realiza de către programatori această adresare. Pentru

* Un caracter numeric sau alfabetic pentru a fi reprezentat sunt necesari minimum 6 biți; în adevăr $2^6=64$ acoperă numărul literelor și ai cifrelor, precum și celelalte semne ce se află de obicei pe claviatura unei mașini de scris.

a ilustra această dificultate este suficient să precizăm că viteza de lucru a unității centrale este de ordinul unei microsecunde, iar viteza echipamentelor periferice este de ordinul unei milisecunde. În cadrul funcționării SPAD, între unitatea centrală și echipamentele periferice se efectuează numeroase transferuri de informație care se execută evident cu viteza elementelor periferice. Pentru ca SPAD în ansamblu să funcționeze cu viteza unității centrale, este necesar ca în paralel cu transferurile de informație din (în) unitatea centrală să se efectueze permanent în unitatea centrală, prelucrări cu viteză de ordinul microsecundei. Aceasta conduce la o împărțire dinamică a memoriei interne în diferite zone în timpul procesului de prelucrare, activitate ce nu mai poate fi stăpinită de un programator atunci cînd își elaborează programul: o zonă din memoria principală servește unui program pentru efectuarea calculelor, iar în paralel în alte zone ale memoriei principale se efectuează schimburi de informație cu perifericele (echipamente de intrare-iesire, memorii auxiliare etc.). Programele prin care s-a realizat această adresare automată împreună cu programele de compilare au devenit nucleul celor mai importante sisteme de programe numite *sisteme de operare* (software de bază) și care s-au constituit în cadrul generației a treia de SPAD, caracteristică principală care diferențiază calitativ calculatoarele de generația a treia de cele precedente [4].

Apariția limbajelor algoritmice de programare constituie o caracteristică de bază a calculatoarelor din generația a două.

1.3. Caracteristicile sistemelor de prelucrare automată a datelor din generația III-a și perspectivele de evoluție

Tendința de automatizare a procesului de prelucrare, ca atare a eliminării căt mai complete a intervenției factorului uman în timpul procesului, a constituit împreună cu tendința de diversificare a aplicațiilor și de abordare a unor probleme tot mai complexe în vederea creșterii eficienței utilizării echipamentelor electronice, elementele propulsoare ale evoluției atât de rapide ale SPAD.

În cadrul unui sistem de prelucrare automată a datelor, datorită evoluției din primele două generații au apărut două categorii de echipamente: unitatea centrală sau procesorul central și echipamentele periferice; după cum am văzut, în vederea utilizării sistemului cu viteza elementului celui mai rapid, este necesar ca unitatea centrală să lucreze permanent, iar transferurile de informație dintre unitatea centrală și periferice să se efectueze în paralel cu prelucrările ce au loc în unitatea centrală. Rezolvarea sistematică și eficientă a acestei cerințe a avut loc în cadrul generației a treia.

Caracteristicile principale ale generației a treia sunt:

a. Dezvoltarea tehnologică (a hardware-ului) s-a produs cu un decalaj favorabil față de dezvoltarea tehniciilor de programare (a software-ului) pînă în prezent; acest decalaj este manifest și în cadrul generației a treia. Circuitele imprimate și circuitele integrate au luat locul circuitelor cablate și elementelor discrete și au devenit caracteristica tehnologiei SPAD din generația a treia.

Organizarea memoriilor interne pe cuvinte a fost înlocuită în cele mai multe cazuri cu organizarea memoriei principale pe octeți (byte)* care permite o folosire mai compactă a acestor memorii.

Viteza de prelucrare a crescut, gabaritele și consumul de energie au scăzut în continuare. Siguranța în funcționare a acestor calculatoare a crescut de asemenea mult în comparație cu cele două generații anterioare.

b. S-a creat *memoria auxiliară pe discuri*, a cărei funcționare poate aminti prin analogie de funcționarea pick-upului. Siguranța sa în funcționare este mai mare ca a membrilor pe bandă magnetică. Viteza sa de lucru este de asemenea mai mare ca la benzile magnetice datorită posibilității de a avea *acces direct* la o anumită zonă prin deplasarea capetelor de citire sau scriere, fără a fi nevoie să se aștepte, ca în cazul *accesului secvențial* (ca la banda de magnetofon) la unitățile de benzi magnetice, rularea benzii cu toate înregistrările anterioare înregistrării dorite.

Fisierele pe benzi magnetice sau discuri magnetice ar putea, din punctul de vedere al suportului, să prezinte înregistrări continue de date, prin juxtapunerea diferitelor cîmpuri (zone) din care sunt formate aceste înregistrări. Acest lucru nu este permis din același motiv pentru care au apărut de altfel dispozitivele de memorie externă (auxiliară) în sistemele de prelucrare automată a datelor. Aceste dispozitive, după cum s-a menționat, au apărut în momentul în care

*) Un octet sau byte=8 bit

problemele de rezolvat, în special cele de tip economic (probleme de prelucrare a datelor în întreprinderi) au cerut participarea unor fisiere cu volum mai mare de date decît capacitatea disponibilă pentru date din memoria internă a calculatorului. Memoria internă (principală) a calculatorului, are o capacitate de ordinul 10^4 – 10^5 octeți la calculatoarele medii. Această capacitate, chiar dacă ar fi folosită integral pentru memorarea datelor, este insuficientă pentru tratarea fisierelor curent întâlnite în problemele din întreprinderi. Acestea sunt memorate pe benzi magnetice sau discuri magnetice, ca să amintim doar principalele suporturi utilizate în memorile externe (auxiliare). Capacitatea acestora este de ordinul 10^6 – 10^8 octeți.

Diferența dintre capacitatea de memorare a dispozitivelor de memorare externă și capacitatea memoriei interne obligă la fragmentarea fisierelor de date, astfel că doar anumite porțiuni din aceste fisieri sunt „încărcate” în memoria internă și prelucrate. Fragmentarea fisierului se manifestă fizic în sensul apariției unui interval între două fragmente succesive, intervale folosite de asemenea și de procesul de accelerare-frânare în cazul benzii magnetice. Grupul de date sau pacchetul de înregistrări (logice), ca fragment al unui fișier, care se transmite integral de la unitatea de memorie externă la cea internă sau invers, poartă numele de *înregistrare fizică* (bloc) în cadrul fisierului respectiv.

Desigur, că o cale comodă din punctul de vedere al programatorului ar fi aceea de a considera (înregistrare fizică)=(înregistrare logică).

După cum am văzut, în raport cu viteza de calcul pe care o poate realiza unitatea centrală, operațiile de intrare-iesire sau operațiile de transfer de date între memoria internă și memorile auxiliare, sunt operații relativ lente (față de viteza de lucru a unității centrale). Gruparea (blocarea) înregistrărilor reprezintă o posibilitate de micșorare a numărului de transferuri de informație între memorile auxiliare și unitatea centrală. În cazul grupării avem formula: (înregistrare fizică)=K×(înregistrări logice), unde K este factorul de grupare (blocaj).

Gruparea influențează nu numai asupra vitezei medii de acces la o înregistrare logică, ci și asupra capacitatii de memorare în memoria auxiliară, prin eliminarea intervalelor dintre înregistrările logice ale aceleiași înregistrări fizice. Amindouă efectele sunt pozitive, dar capacitatea limitată a memoriei interne disponibile pentru date limitează gruparea unui număr prea mare de înregistrări logice.

Înregistrările logice pot fi de lungimi egale în tot cuprinsul fișierului (*lungime fixă*) sau pot avea lungimi diferite în cadrul aceluiași fișier (*lungime variabilă*). De exemplu, un fișier de articole (produse, subansamblu, repere) de același fel, care conține informații de tip descriptiv pentru acestea, va avea înregistrări de lungime fixă. Pe de altă parte, dacă dorim să indicăm în același fișier compoziția fiecărui articol, intrucât articolele pot avea un număr diferit de componente, trebuie să considerăm unor înregistrări de lungime variabilă.

În cazul înregistrărilor de lungime variabilă, lungimea acestora trebuie indicată în cadrul înregistrării însăși.

Una din metodele de a micșora sau elimina dezavantajele lungimii variabile constă în crearea unui număr (fix sau variabil) de înregistrări „înlățuite” de lungime fixă. Această metodă se aplică în special la inscrierea de înregistrări de lungime variabilă pe discuri magnetice.

În cadrul unei înregistrări se găsesc, în general, trei categorii de informații:

- informații de identificare a înregistrării (cheie);
- informații de referință (care sunt folosite în diferite prelucrări);
- informații cantitative (variabile), care sunt actualizate pe baza tranzacțiilor.

Din punct de vedere al modului de utilizare, memoriile (internă și auxiliară) se pot clasifica în două categorii: *secvențiale* (de exemplu banda magnetică sau pachetul de cartele care constituie un fișier) și *adresabile* (de exemplu memoria internă, discurile și tamburul magnetic). Într-o memorie secvențială, înregistrările pot fi localizate numai prin *informația de identificare* (cheia) din cadrul înregistrării care permite să se distingă o anumită înregistrare din mulțimea tuturor înregistrărilor dintr-un fișier. Astfel dacă de exemplu, avem un fișier cu 1 000 note de comandă, se poate considera că informație de identificare numărul de comandă. Să presupunem că din cele 1 000 note de comandă vrem să identificăm comanda nr. 3645. Pentru aceasta trebuie să se compare succesiv informația de identificare a fiecărei note de comandă cu 3645 pînă se obține coincidența celor două numere.

În medie, pentru găsirea unei înregistrări în cazul memorilor secvențiale, este necesar să se compare jumătate din numărul total al înregistrărilor. Dacă *tranzacțiile* (înregistrările care actualizează un fișier, înlocuind sau modificînd înregistrările existente) care se efec-

tuează sint în aceeași ordine cu înregistrările asupra cărora își execută acțiunea, atunci necesarul de comparații scade mult. Pentru aceasta sunt necesare însă operații de sortare sau ordonare a tranzacțiilor.

Accesul la memoriile adresabile este relativ mult mai rapid, datorită mecanismelor de selectare care permit rapid localizarea înregistrării sau informației pe baza adresei (fizice) asociate înregistrării respective.

Pentru a ilustra corespondența ce se stabilește între informația de identificare și adresa unei înregistrări, să considerăm cazul particular, frecvent, al înregistrărilor de lungime fixă așezate în ordinea crescătoare a numerelor care constituie informația de identificare a înregistrărilor. Să presupunem că fișierul are 5 înregistrări și că fiecare înregistrare are 45 octeți și că inscrierea în memorie se face începînd de la octetul 700.

Prima înregistrare este inscrisă pe octeți 700, 701, ..., 744. A doua înregistrare ocupă octeți de la 745 la 789, a treia 790–834, a patra 835–879, iar ultima înregistrare 880–924.

Să presupunem că informația de identificare a primei înregistrări este 201 (de pildă numărul unei comenzi în cazul cînd avem de a face cu un fișier de ordin de comandă), iar ultima înregistrare are informația de identificare 205.

Să notăm cu A_0 adresa primei înregistrări, cu I_0 lungimea (în octeți) a fiecărei înregistrări, cu I_0 informația de identificare a primei înregistrări, cu A și I adresa, respectiv informația de identificare a înregistrării cerute. Între aceste variabile avem relația :

$$A = A_0 + (I - I_0)l$$

În cazul particular considerat avem :

$$A = 700 + (I - 201) \cdot 45 \quad (1)$$

sau

$$I = 201 + \frac{A - 700}{45} \quad (2)$$

De exemplu, pentru ultima înregistrare, avînd informația de identificare 205, rezultă adresa :

$$A = 700 + (205 - 201) \cdot 45 = 700 + 180 = 880$$

Acstea formule nu se pot aplica în cazurile cînd, deși fișierul este ordonat, informația de identificare nu constituie un sir de numere naturale consecutive, sau cînd lungimea înregistrărilor este variabilă etc. Formulele (1) și (2) permit uneori să se renunțe la scrierea informației de identificare în cadrul înregistrării, această informație putind fi calculată cînd este necesar (2).

c. Extinderea gamei de memorii auxiliare și crearea de dispozitive de transfer între echipamentele periferice și unitatea centrală

este o caracteristică de bază a generației a treia de SPAD. Mărirea numărului și a varietății de echipamente periferice (dispozitive de transmitere la distanță a datelor, dispozitive speciale de afișare etc.), în afară de memorile auxiliare pe benzi sau discuri magnetice au pus probleme dificile de organizare complexă, atât din punct de vedere tehnologic, cât și ca sisteme de operare. Au apărut *unitățile de control și canalele care realizează adaptări între unitatea centrală și dispozitivele periferice și care împreună cu sistemul de operare (software-ul de bază) asigură circulația informației între unitatea centrală și echipamentele periferice, efectuarea simultană de schimburi de informație, în paralel cu prelucrarea informației în unitatea centrală*. În acest mod, s-a putut realiza dezideratul ca sistemul în ansamblu să funcționeze tot timpul cu viteza elementului cel mai rapid din alcătuirea sa: unitatea centrală.

d. Programele de compilare, programele de control ale sistemului, programele necesare controlului și comandării transferurilor de informație între blourile componente ale SPAD (fig. I.3.1), programele de bibliotecă sau de serviciu care sunt incorporate în programele de exploatare elaborate de programatori sau sunt utilizate ca atare, programele care realizează înlanțuirea diferitelor părți ale unui program compilat, sau înlanțuirea diferitelor programe de exploatare, programele de gestiune automată a fișierelor (accesul la fișiere, protecția acestora etc.), programele pentru completarea sau punerea la zi a programelor menționate mai sus etc., alcătuiesc *sistemul de operare*.

Funcțiunile sistemului de operare sunt foarte complexe, el asigurând funcționarea SPAD în ansamblu, reducind cât mai mult intervențiile operatorului care urmărește de la panoul central (consolă) această funcționare. De pildă, funcțiunile sistemului de operare privind adresarea automată pot răspunde cerinței de optimizare a utilizării memoriei principale în condițiile coexistenței dinamice a mai multor zone în memoria principală, afectate de diferitor programe, pentru completarea tuturor celulelor acesteia, a alcătuirii fișelor de aşteptare cât mai mică a fiecărui echipament periferic etc. În principal, funcțiunile sistemului de operare permit rularea programelor în șiruri (lanțuri) de programe, controlează unitățile de intrare-iesire, organizează multiprogramarea (rularea simultană a mai multor programe) și prelucrarea la distanță, eliminând intervenția operatorului prin operarea automată în mod continuu a programelor.

Corespunzător fiecărei configurații trebuie generat un anumit sistem de operare, generare care se efectuează cu ajutorul unui program

din cadrul sistemului de operare general, care selectează părțile corespunzătoare din sistemul de operare pentru configurația dată.

Efortul social pentru elaborarea sistemelor de operare a fost foarte mare (se poate aprecia la mii de programator-anii) și dezvoltarea lor a rămas în urma dezvoltării tehnologice.

Au apărut de asemenea în ultimii ani programe standard (corespunzătoare unor anumite clase de aplicații sau prelucrări importante), numite *pachete de programe*, dintre care unele pot fi adaptate relativ ușor pe diferite tipuri de calculatoare și care se apropie în complexitate de sistemele de operare. Această tendință a fost cerută de posibilitatea utilizării în comun de către calculatoare de diferite tipuri a unei *baze de date* comune, tendință condiționată favorabil și de teleprelucrare.

e. Urmărind consecvent evoluția de pînă acum, se vede că funcționarea sistemelor de prelucrare automată a datelor, pe linia automatizării căt mai deplină a prelucrării informației, presupune o egalizare căt mai completă a caracteristicilor unității centrale și ale unităților periferice. Pe această linie se pare că *generația a patra* de SPAD va prezenta ca indice calitativ distinctiv de generația anterioară, difuzarea memoriei principale la nivelul unităților periferice, tendință pregătită de altfel de unitățile de control caracteristice generației a treia, care constituie — de fapt — unități centrale în miniatură. Corespunzător acestei difuziuni a memoriei principale, sistemul de operare va trebui adaptat, operație care credem că va constitui factorul principal de intîrziere în utilizarea intensivă a echipamentelor din generația a patra.

De asemenea, în cadrul generației a patra, o parte dintre funcțiunile sistemului de operare vor fi realizate cu ajutorul hardware-ului, de pildă, cablarea unor instrucțiuni sau programe ce se utilizează frecvent în procesul de prelucrare. Este de așteptat că viteza de prelucrare și capacitatea de memorare să crească în continuare. Se poate menționa în acest sens apariția unităților de memorii magnetice cu citire transversală, care au o capacitate echivalentă cu capacitatea a circa 25 000 benzi magnetice obișnuite (cu citire longitudinală) și un timp de acces de ordinul secundelor. Performanțe superioare acestei memorii magnetice au memoriile cu fascicol laser (permis organizarea în acel direct a informațiilor de ordinul bilioanelor de biți și timp de acces de ordinul milisecundelor), care sunt în fază de utilizări experimentale.

În concluzie, la această scurtă trecere în revistă a factorilor principali care au marcat generațiile de SPAD în evoluția lor, trebuie

avut în vedere că întregul proces a fost continuu, iar demarcările făcute nu au limitări precise în timp, diferite caracteristici distinctive unei anumite generații coexistând cu cele din generația precedentă, în cadrul evoluției prezentate fiind numeroase SPAD de tranziție, pînă la obținerea exemplarelor în care factorii distinctivi predomină.

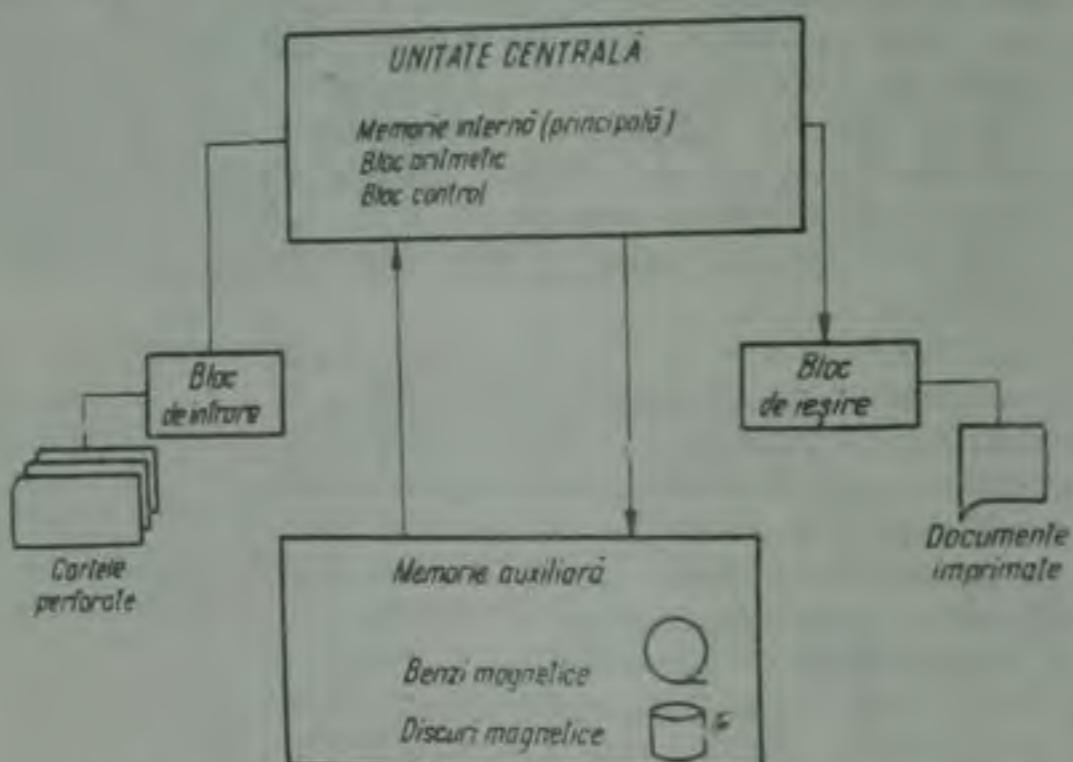


Fig. 1.3.1. Schema de principiu a blocurilor unui SPAD.

O schemă de principiu a blocurilor principale ale unui SPAD și fluxul informațiilor între aceste blocuri este dată în figura 1.3.1. Se observă că toate informațiile, atât cele de la intrare, cât și cele pentru ieșire, trec prin unitatea centrală și anume prin memoria internă.

1.4. Dezvoltarea metodelor de programare și modelare în raport cu necesitățile de organizarea datelor

În vederea rezolvării unei probleme pe SPAD, este necesar ca pe o parte să se alcătuiască un *program*, care este format din instrucțiunile pe care trebuie să le interpreteze calculatorul electronic și să le execute, iar pe de altă parte, trebuie să se organizeze datele nume-

rice sau alfanumerice (formate din caractere alfabetice sau numerice), pe care instrucțiunile programului urmează să le prelucreze în vederea obținerii rezultatelor.

Acste două aspecte au constituit coordonatele pe care s-au dezvoltat aplicațiile sistemelor de prelucrare automată a datelor.

Menționăm că o instrucțiune conținând informația necesară efectuării unei operații, în afară de operația propriu-zisă, trebuie să conțină informații despre data sau datele asupra cărora se execută operația respectivă.

Pentru a vedea care pot fi tipurile de instrucțiuni pe care le putem întîlni în cadrul unui program, vom trece în revistă clasele mari de prelucrări ce se pot executa pe un calculator electronic.

În afară de calculele analitice care în ultimă instanță se reduc pe calculatorul electronic la operațiile aritmetice și logice elementare, folosindu-se rezultatele analizei numerice privind metodele iterative de calcul și existența, convergența și precizia soluțiilor, se pot executa calcule logice folosind rezultatele din algebra booleană și reprezentări numerice, asociate variabilelor și funcțiilor logice, care utilizează sistemul binar, adaptate structurii binare a cuvintelor din SPAD. Rezultatele analizei numerice privind metodele iterative de aproximății succesive pentru rezolvarea ecuațiilor, a sistemelor de ecuații algebrice liniare, a integrării și derivării numerice, a rezolvării ecuațiilor diferențiale, a sistemelor de ecuații diferențiale sau a ecuațiilor diferențiale cu derivate parțiale, precum și diferite alte calcule analitice (folosind formule în algoritmele respective) și procesele de optimizări specifice cercetării operaționale, au căpătat o mare importanță practică tocmai datorită posibilității efectuării calculelor pe SPAD. În legătură cu aceasta cităm următoarele afirmații a lui Richard Bellman [5], referitoare la complexitatea problemelor care pot fi abordate cu ajutorul calculatoarelor electronice: „Din fericire dispunem (în ultima vreme) de instrumente cu care să ne putem ajuta, acestea sint calculatoarele electronice a căror existență modifică înșăși noțiunea de „soluție“.

Dacă aceste metode numerice au fost puse în valoare în special în ultimele două decenii, cînd calculatoarele electronice au permis efectuarea în timp util a volumelor mari de calcule care erau necesare, un fenomen analog s-a întîmplat și cu metodele algebrei booleene elaborate la mijlocul secolului al XIX-lea de George Boole și completate ulterior de o serie de rezultate, la care școala românească de sub conducerea Acad. Gr. Moisil [6] a avut o contribuție impor-

tantă. Cantitatea de calcule în algebra booleană necesară în probleme de analiză, simplificare și sinteza circuitelor de comutăție, în probleme de optimizări, în domeniul luării deciziilor, a analizei informațiilor etc., crește foarte repede cu numărul variabilelor logice asociate, făcând imposibil calculul manual. Adaptarea efectuării calculelor logice folosind cuvinte binare asociate variabilelor pe calculatorul electronic, a deschis de asemenea mari perspective rezolvării claselor mari de probleme menționate, prin metodele algebrei booleane [4].

În vederea ilustrării aspectelor care stau la baza modului prin care variabilelor logice li s-au asociat cuvinte scrise în sistemul binar, făcind să corespundă operațiilor logice cu aceste variabile, operații cu cuvinte binare asociate să considerăm următorul exemplu simplu :

Funcția „sau” ($a \vee b$) de două variabile din algebra booleană este definită prin următorul tabel în care cifrele 0 și 1 reprezintă valoările logice fals și adevărat :

a	0	1	0	1
b	0	0	1	1
$a \vee b$	0	1	1	1

Se observă că în matricea formată din cifrele binare 0 și 1 (pe liniile corespunzătoare lui a și b) avem reprezentate pe coloane numeroarele zecimale 0, 1, 2, 3. Se poate remarcă de asemenea că pe prima linie din matrice alternează cîte un 0 cu cîte un 1, iar pe a doua alternează cîte două cifre 0 cu cîte două cifre 1.

Dacă definim adunarea cifrelor binare ($1+1=1$ fără transport) ca în tabela :

+	0	1
0	0	1
1	1	1

și executăm adunarea pe componente 0 și 1 ale cuvintelor asociate lui a și b , atunci se observă că cuvintul binar asociat lui $a \vee b$ este 0111. Pentru 3 variabile, cuvintele care se asociază variabilelor logice se deduc din următoarea matrice :

a	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
c	0	0	0	0	1	1	1	1

și se poate observa ușor că coloanele acestei matrice reprezintă numeroarele zecimale 0, 1, 2, 3, 4, 5, 6, 7, scrise în binar și de asemenea, se poate remarcă modul cum alternează grupele de cifre 0 cu grupele de cifre 1 și.m.d.

În mod analog, se pot transpune sistematic metodele și procedeele de calcul ale algebrei booleane în calcule cu cuvinte binare, a căror reprezentare pe calculatorul electronic este directă, aceasta reprezentind una din caracteristicile de bază ale funcționării acestuia.

În afară de calculele aritmetice și logice, pe calculatoarele electronice se execută și operații de selectare, sortare și ordonare, a căror pondere este majoră în cadrul aplicațiilor din întreprinderi (fig. 1.4.1.).

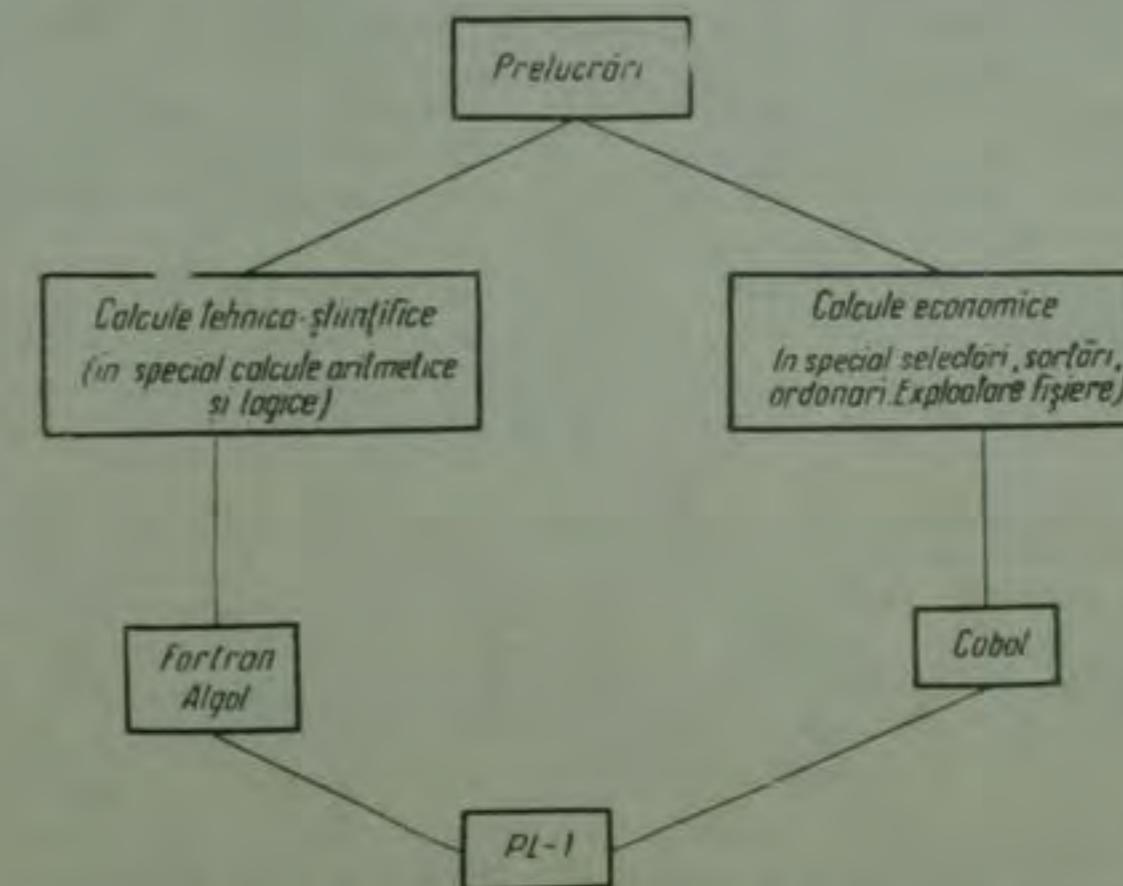


Fig. 1.4.1. Schema tipurilor de prelucrări și a limbajelor de programare care pot fi utilizate.

Prin selectare înțelegem operația prin care dintr-o mulțime de elemente se extrage o submulțime a sa. Nu este cazul să insistăm asupra importanței operației de selectare care apare în majoritatea

aplicațiilor din cadrul întreprinderilor. Foarte frecvent este necesar să se selecteze anumite poziții dintr-un fișier de materiale, sau fișier de personal, sau fișier de articole etc.

Prin sortare înțelegem operația prin care o mulțime este împărțită în submulțimi disjuncte (care nu au elemente comune două cite două), care împreună formează mulțimea inițială. De asemenea, nu este nevoie să insistăm asupra importanței acestor operații în rezolvarea problemelor din întreprinderi. Sortarea fișierului de materiale, a fișierului de articole, de personal etc., este o operație care apare foarte frecvent în aplicațiile privind stocurile, producția, calculele finanțiar-contabile etc.

Prin ordonarea elementelor unei mulțimi înțelegem aranjarea elementelor conform unei relații de ordine dată (de pildă ordinea crescătoare sau descrescătoare a numerelor naturale, ordinea alfabetică etc.). Adesea aceste trei operații se execută în strânsă legătură și cel mai adesea sortarea se efectuează pe baza ordonării elementelor, sau selectarea se efectuează pe baza sortării unei mulțimi ordonate convenabil etc. În practica centrelor de calcul, specialiști se referă la aceste operații, în general, prin termenul de sortare.

Să observăm că cele două mari clase de prelucrări descrise: calcule aritmetice și logice (în special pentru calculele tehnico-științifice) și selectare, sortare și ordonare (în special pentru probleme economice) solicită sistemul de prelucrare automată a datelor în mod diferit.

Calculele aritmetice și logice necesită existența unor instrucțiuni pentru executarea lor, care nu fac uz decit relativ în mică măsură de volume mari de date, ca atare de mulțimi mari de date organizate în fișiere, pe cind operațiile de selectare, sortare și ordonare sunt strins legate de utilizarea fișierelor de date și ca atare necesită instrucțiuni specifice de transfer ale pachetelor de date din (în) unitatea centrală în (din) memoriile auxiliare pe benzi sau discuri magnetice.

Desigur, în afară de aceste două categorii mari de instrucțiuni corespunzătoare celor două clase mari de prelucrări, mai există și alte instrucțiuni care sunt necesare în procesul de prelucrare, dar în esență pentru înțelegerea evoluției limbajelor de programare și a metodelor de organizarea datelor, precum și a influențelor reciproce în procesul de elaborare a programelor și respectiv a fișierelor, acestea constituie elementele majore.

Așa cum am menționat înainte, sistemele de prelucrare automată a datelor au apărut și au fost utilizate în prima parte a evoluției lor

la efectuarea de calcule științifice și tehnice. Primele limbi de algoritmice de circulație mai mare care au fost folosite în cadrul programării automate au fost (în ordinea apariției lor): FORTRAN (*Formula Translation*) și ALGOL (*Algorithmic Language*).

În afară de avantajele descrise anterior față de programarea în limbaj mașină, menționăm că în elaborarea acestor limbi, specialiștii și-au propus ca obiectiv principal să obțină o programare relativ ușoară a calculelor aritmetice și logice (fig. 1.4.1.), structura și instrucțiunile acestor limbi reflectând tocmai realizarea acestui obiectiv.

În cadrul acestor limbi, scrierea expresiilor aritmetice și logice care apar în algoritme (procesele de calcul) și pe baza căror se execută calculele este foarte apropiată de formalismul matematic și ca atare la îndemina unui număr relativ mare de specialiști.

Instrucțiunile limbajelor FORTRAN și ALGOL desigur permit elaborarea oricărui algoritm* sănătos și ele permit în esență realizarea următoarelor acțiuni de bază care apar în structura programelor:

- atribuirea de valori variabilelor din program valori care rezultă din calculul expresiilor aritmetice și logice în timpul procesului de prelucrare prin *instrucțiunea de atribuire* sau *asignare*;
- iterarea unei secțiuni de calcule prin *instrucțiunile de ciclare*;
- transferul la instrucțiunea dorită din program pentru continuarea calculelor prin *instrucțiuni de transfer* (salt). Menționăm că în absența instrucțiunilor de transfer, instrucțiunile programului sănătos parcurse în ordinea în care sunt scrise în program, ca atare în ordinea în care sunt adresate în memorie.

Frecvent, instrucțiunile de transfer sunt *instrucțiuni condiționale*, transferul efectuându-se (condiționat) în funcție de valorile atribuite în calculele anterioare parametrilor de care depind aceste instrucțiuni.

- calculul funcțiilor (de una sau mai multe variabile) se face cu ajutorul *procedurilor* (subprograme, părți ale programului), a căror elaborare este mai completă în limbajul ALGOL [1].

Acstea subprograme (procedurile) prin care se reprezintă funcțiile sănătos folosite în cadrul programului cu setul corespunzător de valori

* Se poate demonstra relativ ușor, că orice algoritm normal în sensul lui A. Markov [6] se poate exprima cu ajutorul instrucțiunilor din FORTRAN sau ALGOL.

atribuite variabilelor ori de cîte ori este nevoie, fiind necesar ca ele să fie scrise în cadrul programului o singură dată.

Pentru a ilustra modul cum se utilizează instrucțiunile descrise în cadrul unui program, vom folosi reprezentarea unui program cu ajutorul schemei logice asociate (vezi cap. III).

Algoritmul de rezolvare al unei probleme pe SPAD, care este o descriere precisă și completă a procedeului de calcul, ca atare o mulțime completă de reguli precise care permit să se obțină din datele inițiale un rezultat determinat, trebuie exprimat în funcție de instrucțiunile prezentate. Schema logică ce exprimă algoritmul (cap. III) permite să se ilustreze acest aspect.

Pe baza schemei logice se alcătuiește programul constituit din șirul de instrucțiuni care fac parte din limbajul de programare ales.

In limbajele algoritmice ALGOL și FORTRAN, se acordă, însă, mijloace puține pentru manevrarea informației (datelor) de intrare și ieșire și, în special, pentru manevrarea unor cantități mari de date (a fișierelor) în vederea prelucrării lor.

Problemele din întreprinderi se caracterizează tocmai prin nevoie de organizarea unor cantități mari de informații, în vederea prelucrării lor prin operații relativ simple, care necesită însă manevrări repetitive ale înregistrărilor din fișiere.

Dezvoltarea rapidă a aplicațiilor economice din întreprinderi pe SPAD, a căror eficiență cerea extinderea căt mai rapidă a noilor metode de prelucrare la volume mari de date, a condus la elaborarea unui alt limbaj algoritmic care să răspundă noilor obiective și care să fie la îndemina unei clase căt mări de specialiști. Pentru aceasta s-a elaborat limbajul COBOL (Common Business Oriented Language), limbaj orientat pe probleme economice, în a căruia structură sint dezvoltate elementele necesare pentru organizarea fișierelor și prelucrarea sistematică a înregistrărilor din aceste fișiere.

In acest sens, în orice program în limbajul COBOL se poate pune în evidență două categorii de informații :

a. prima categorie de informații care apar în partea numită diviziunea procedurilor în care sunt scrise expresiile și instrucțiunile necesare prelucrării ;

b. a doua categorie de informații se ocupă cu modul de organizare al datelor, ca atare cu descrierea înregistrărilor din fișiere, a fișierelor și a echipamentelor utilizate (în special echipamentele de memorie auxiliară asociate fișierelor) și identificarea programului ce se utilizează ; aceste informații sunt scrise în diviziunea datelor, diviziunea echipamentelor și respectiv în diviziunea de identificare.

In ce privește diviziunea procedurilor, nu există elemente principale care să distingă expresiile și instrucțiunile de atribuire, transfer și cielare și procedurile din limbajele FORTRAN și ALGOL, fără de cele din COBOL. Ca scriere a acestora se poate menționa că modelarea formalismului matematic nu mai este atât de apropiată. În COBOL folosindu-se în măsură mai mare cuvinte din limbajul natural care să descrie o instrucțiune sau o expresie și, în general, exprimarea algoritmilor este mai greoai în COBOL.

Menționăm că diferențele care apar în ce privește instrucțiunile, se datorează caracterului prelucrării pe care o efectuează programul din COBOL și anume, iterarea prelucrării relativ simple, asupra înregistrărilor : pentru manevrarea acestora este nevoie de instrucțiuni, care sunt specifice limbajului COBOL privind transferul de informații în (din) diferite blocuri ale SPAD sau între zonele din memoria internă sau pentru folosirea de subprograme standard din cadrul sistemului de operare.

Elementele specifice care caracterizează limbajul COBOL sunt cele din care se alcătuiesc diviziunile privitoare la organizarea și descrierea datelor și echipamentelor. Aceste elemente sunt o reflectare a specificului aplicațiilor economice din întreprinderi, privind organizarea datelor în înregistrări și fișiere.

In general, înregistrările sunt prelucrate în același mod, pentru fiecare repetându-se următoarele etape :

- transfer din fișier în memoria principală ;
- prelucrarea conform diviziunii procedurilor și obținerea unei înregistrări de ieșire care se memorează în memoria principală ;
- transferul înregistrării de ieșire la imprimantă sau pe bandă magnetică sau disc magnetic.

In general, în memoria principală sunt afectate trei zone : căte o zonă pentru înregistrările de intrare și datele de ieșire și o zonă de lucru pentru efectuarea calculelor.

Prin urmare în cadrul programului, în diviziunile pentru organizarea datelor trebuie să se dea informații privind și spațiul din memoria internă necesar să fie alocat acestor zone.

Diviziunea datelor conține următoarele elemente principale :

a. descrierile înregistrărilor, în care se prezintă informația pentru identificarea înregistrării, natura informațiilor (alfabetice, numerice), numărul de caractere ale fiecărui element de informație din înregistrare, modul cum sunt așezate aceste elemente de informații pe

diferite nivele — analog aşezării cuvintelor în rubricile dintr-un formular — etc. Informația de identificare a înregistrării se află pe primul nivel din înregistrare;

b. descrierile fișierelor, în care se prezintă numele fișierului, tipurile de înregistrări din fișier, lista cu informațiile de identificare ale tipurilor înregistrărilor (când înregistrările dintr-un fișier sunt de același tip, lista conține o singură informație de identificare), formatele etichetelor care se inscriu pe bandă sau disc pentru identificarea și localizarea fișierului etc.

Diviziunea echipamentelor conține descrierile echipamentelor, care definesc modul cum se atribuie dispozitivele fiecărui fișier utilizat de program, echipamentul periferic de care se dispune în sistemul de prelucrare automată a datelor, zonele din memoria principală necesare programului etc.

În cadrul diviziunii procedurilor, în afară de instrucțiunile și procedurile analoge celor din FORTRAN sau ALGOL, mai apar instrucțiuni care asigură transferul de informații între cele trei zone din memoria principală (din zona de intrare în zona de lucru, din zona de lucru în zona de ieșire etc.), transferul de înregistrări din (în) fișiere în (din) memoria principală sau fișiere, verificarea etichetelor fișierelor etc.

Un program în COBOL este compus din două părți:

— în prima parte (diviziunea de identificare, diviziunea datelor și diviziunea echipamentelor) se inscriu diferențele descrierii de echipamente, fișiere și înregistrări necesare programului;

— în a doua parte (diviziunea procedurilor) se inscriu expresiile, instrucțiunile de atribuire, transfer și ciclare și procedurile analoge celor din FORTRAN și ALGOL și instrucțiunile de transfer ale informațiilor (nu transfer la instrucțiuni din cadrul programului) către diferențe blocuri ale SPAD și între zonele afectate din memoria principală. Elaborarea sistematică a acestor din urmă instrucțiuni, care sunt puțin elaborate în FORTRAN și ALGOL, împreună cu descrierile din prima parte a programului, constituie mijloacele specifice limbajului COBOL de manevrare a cantităților mari de date.

Un program scris în limbajul COBOL are următoarea structură:

IDENTIFICATION DIVISION

identificarea programului: numele programului, numele programatorului, data elaborării etc.

ENVIRONMENT DIVISION

descrierea echipamentelor: tipul de calculator, memoriile auxiliare pentru fișiere etc.

DATA DIVISION

FILE SECTION

FD Descrierea fișierelor

<zz> Descrierea înregistrărilor logice

SD Descrierea fișierelor de sortare

REPORT SECTION

RD Descrierea fișierelor ce urmează să fie imprimate

WORKING STORAGE SECTION

Descrierea zonelor din memoria internă

CONSTANT SECTION

Descrierea constantelor, ce urmează a fi prelucrate în cadrul programului

PROCEDURE DIVISION

DECLARATIVES

Numele secțiunilor și al paragrafeelor ce vor fi activate în cadrul acestei divizii, de instrucțiuni de transfer

Fraze și instrucțiuni

END DECLARATIVES

Secțiuni, paragrafe, fraze și instrucțiuni, care împreună cu secțiunile și paragrafele declarate în secțiunea

DECLARATIVES alcătuiesc programul propriu-zis de prelucrare

Se observă că diviziunea „DATA DIVISION” conține :

a. secțiunea „FILE SECTION”, unde sunt descrise fișierele ordinare (care nu se sortează în cadrul programului). Descrierea fiecărui fișier este reperată de simbolul FD (File Description), iar descrierea înregistrărilor din care este format, de două cifre $\langle zz \rangle$ care alcătuiesc „numere de nivel” cu ajutorul cărora se descriu zonele (cîmpurile) din structura înregistrării.

Fișierele ce urmează să fie sortate sunt descrise separat în partea din secțiune reperată prin simbolul SD (Sort Description).

b. secțiunea „REPORT SECTION”, unde sunt descrise fișierele ce urmează să fie imprimate. Fiecare descriere este reperată de simbolul RD (Report Description) ;

c. secțiunea „WORKING STORAGE SECTION”, unde sunt descrise toate zonele din memoria internă, care sunt destinate să conțină informații pentru programul respectiv ;

d. secțiunea „CONSTANT SECTION”, unde sunt descrise valorile constante utilizate de program.

În ce privește PROCEDURE DIVISION se observă că este formată din secțiunea DECLARATIVES (cuprinsă între cuvîntul cheie DECLARATIVES și cuvîntele cheie END DECLARATIVES) și din secțiunile programului propriu-zis de prelucrare. În aceste secțiuni sunt scrise instrucțiunile care se execută în ordinea în care sunt scrise și care se grupează în fraze, care la rîndul lor se grupează în paragrafe, care pot fi grupate în secțiuni ; paragrafele și secțiunile au etichete (nume create de programator) pentru a se putea efectua transferurile la aceste părți de program, transferuri care intrerup parcurgerea instrucțiunilor în ordinea în care sunt scrise.

FILE SECTION conține după cum am văzut descrierea structurii fișierului și a înregistrărilor ce se află în fișier.

O înregistrare se descrie în program în modul următor :

```

01 <NI1>
02 <N2>
03 <N3>
03 <N4>
02 <N5>
03 <N6>
04 <N7>
04 <N8>
03 <N9>

```

Numerele 01, 02, 03, ... sunt numere de nivel, care descriu structura înregistrării NI₁, numărul de nivel 01 fiind asociat înregistrării NI₁, iar celelalte 02, 03, ... fiind asociate zonelor din această înregistrare : N₂, N₃, N₄, ... Aceste numere descriu structura înregistrării NI₁, folosindu-se convenția următoare :

O zonă avînd asociat numărul de nivel z conține toate zonele care sunt inscrise în continuare și care au asociate numere de nivel $z' > z$. Ca atare structura înregistrării NI₁ este cea din figura 1.4.2 unde, de pildă N₃ și N₄ sunt subzone ale zonei N₂ și.m.d.

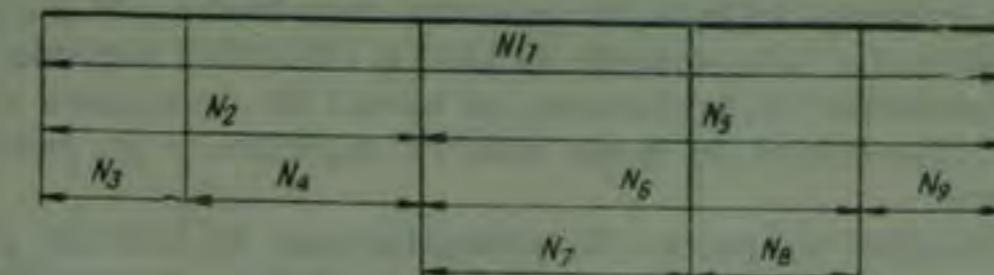


Fig. 26. 1.4.2. Structura înregistrărilor din fișiere.

După cum am menționat, limbajul COBOL prezintă de asemenea mijloace eficiente pentru alcătuirea rapoartelor la imprimantă, care rezultă în cea mai mare parte din supletea pe care o prezintă limbajul în manevrarea fișierelor. Cunoașterea acestor aspecte ale limbajului COBOL ajută specialistului ce urmează să proiecteze fișiere în cadrul unui sistem informațional, punîndu-i la îndemînă mijloace formale sistematice și riguroase de descriere și apreciere a performanțelor fișierului în cadrul prelucrării. Aceste calități, împreună cu răspîndirea problemelor economice, au făcut ca în prezent limbajul COBOL să fie cel mai utilizat în sistemele de prelucrare automată a datelor. Prin urmare în evoluția limbajelor de programare și a metodelor de organizare a datelor, acestea din urmă au influențat în bună măsură dezvoltarea tehniciilor de programare, conducînd la crearea limbajului COBOL. O ilustrare a acestor tendințe o constituie și apariția limbajului de programare PL-1 (Programming Language 1), care înmânunchiază avantajele celor două categorii de limbi orientate pe calcule tehnico-științifice, respectiv pe probleme economice, într-o structură unică, ștergînd aspectele care le diferențiază și constituind din acest punct de vedere un limbaj mai general de programare (fig. 1.4.1).

Pentru a ilustra modul cum intervin cele două clase mari de prelucrări în aplicațiile specifice din întreprinderi și importanța pe care o prezintă problema organizării fișierelor, să considerăm ca exemplu problema descompunerii produselor planificate pentru o anumită etapă (săptămână, lună, trimestru, an etc.) în reperele și materialele constitutive și listarea tuturor reperelor și materialelor și a cantităților respective ce trebuie produse. Aceasta este o aplicație care trebuie rezolvată în cadrul oricărei întreprinderi, ea fiind necesară pentru calculul necesarului de aprovizionat, pentru ordonanțarea (programarea operativă) producției etc. Totodată, exemplul va pune în evidență aspecte ale modului cum trebuie să conlucreze modelul matematic care servește ca bază a elaborării algoritmilor și, deci, a programelor de prelucrare, cu modul de organizare al fișierelor care influențează în mare măsură algoritmele și programele de prelucrare.

Să considerăm un proces de producție care trebuie să producă conform planului două produse 2 și 4, având structuri de subansamblu și repere ca în reprezentările arborelor asociate (fig. 1.4.3) :

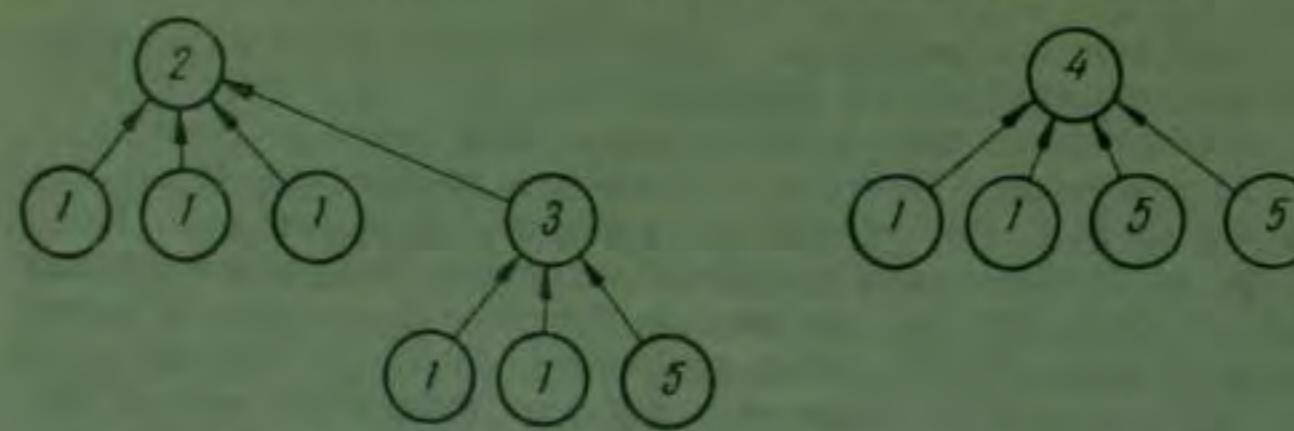


Fig. 1.4.3. Reprezentarea arborescentă a procesului de producție.

Produsul (final) 2 este compus din subansamblul 3 (care este format din două repere 1 și un reper 5) și din trei repere 1, iar produsul 4 este format din două repere 1 și două repere 5. Arborii asociati (fig. 1.4.3) descriu nu numai numărul reperelor care intervin în fabricarea unui produs, dar și ordinea în care trebuie asamblate aceste repere (conform nivelelor din reprezentările arborelor asociate). Conform unei definiții obișnuite a distanței (numărul ramurilor dintre două noduri în arbore) putem remarcă că un reper 1 din subansamblul 3 este (ca nod) la distanța 2 de nodul care reprezintă pro-

dusul final 2, oricare din nodurile asociate reperului 5 se află la distanța 1 de nodul asociat produsului final 4, un nod fiind la distanța 0 de el însuși și.a.m.d.

Informația conținută în acești arbori (care sunt descriși în documentele din fiecare întreprindere) poate fi reprezentată sub forma unei matrici A (matricea de asamblare), în care fiecare element a_{ij} reprezintă câte unități din articolul i sunt necesare pentru a fi produs articolul j cele două noduri asociate i și j aflându-se la distanță 1. Matricea A asociată în exemplul considerat este :

$$A = \begin{pmatrix} 0 & 3 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \end{pmatrix}$$

unde se observă că de pildă elementul $a_{12}=3$, reprezintă faptul că sunt necesare trei articole 1 (situate la distanță 1) pentru articolul 2, iar $a_{32}=1$, reprezintă faptul că este necesar un articol 3 pentru a fi produs articolul 2 etc.

Liniile în matricea A care au numai elemente 0, reprezintă produse finale (2 și 4), iar coloanele care au numai elemente 0 reprezintă repere sau materiale (1 și 5).

Pentru a calcula care sunt elementele i necesare pentru articolul j și care se află la distanță 2, considerind nodurile i și j asociate în reprezentarea arbore, calculăm $A^2 = A \cdot A$ (înmulțirea obișnuită în algebra lineară pentru matrici având elemente numere reale). În exemplul considerat avem :

$$A^2 = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Se observă că pentru produsul 2 sunt necesare 2 articole 1 și un articol 5, toate la distanță 2 (pe nivelul al doilea în reprezentarea arborelui asociat). Dacă d este distanța maximă care apare în reprezentările arbore, atunci pentru $m > d$ matricile A^m au toate elementele 0.

Pentru $m \leq d$, un element a_{ij}^m din A^m reprezintă cite articole i sunt necesare articolului j aflate la distanța m ca noduri asociate în reprezentarea arbore.

Putem defini matricea N a necesarului de articole astfel :

$$N = \sum_{k=0}^d A^k = I + A + A^2 + \dots + A^d \quad (1)$$

unde $I = A^0$ este matricea unitate care interpretează faptul că fiecare articol este necesar pentru el însuși (se află la distanța 0 de el însuși, $k=0$), iar d este distanța maximă în reprezentările arbore.

În exemplul considerat $d=2$, $N=I+A+A^2$:

$$N = \begin{pmatrix} 1 & 5 & 2 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 & 1 \end{pmatrix} \quad \text{unde} \quad I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Deoarece

$$N = \sum_{k=0}^d A^k = I + \sum_{k=1}^d A^k$$

și deoarece

$$A^{d+1} = 0$$

avem :

$$A \cdot N = \sum_{k=1}^{d+1} A^k = \sum_{k=1}^d A^k = N - I$$

Deci

$$I = N - A \cdot N = N(I - A)$$

Rezultă că

$$N = (I - A)^{-1} \quad (2)$$

După cum vom vedea, ținind cont de modul de organizare al fișierelor și de specificul prelucrării pe SPAD, algoritmul și în ultimă instanță programul bazat pe formula (1) pentru calculul necesarului N este mai indicat decit cel care s-ar elabora folosindu-se formula (2).

Să presupunem acum că în planul de producție este necesar să producem 5 articole 2 și 10 articole 4. Această informație se poate introduce cu ajutorul vectorului coloană C (cantități de produs).

$$C = \begin{pmatrix} 0 \\ 5 \\ 0 \\ 10 \\ 0 \end{pmatrix}$$

Necesarul de articole total T va rezulta ca un vector coloană $T = C \cdot N$.

În cazul exemplului considerat :

$$T = \begin{pmatrix} 45 \\ 5 \\ 5 \\ 10 \\ 25 \end{pmatrix}$$

Deci este necesar să se producă 45 articole 1, 5 articole 2 etc.

Reprezentarea matricială, care folosește ca model matematic pentru rezolvarea problemei, trebuie adaptată unui mod eficient de organizare a datelor pe SPAD. Informațiile sub formă matricială vor fi reprezentate în fișiere, în care eliminarea înregistrării elementelor 0 reduce mult volumul de memorie necesar inscrierii datelor. Pentru acest motiv se preferă ca fișier de bază (fișier de structură) matricea A , care are multe elemente 0 și care conduce la un fișier de dimensiuni mult mai mici prin eliminarea elementelor 0, decit inserirea în fișier a informațiilor conținute în matricea N , având mult mai puține elemente 0 și care se poate obține prin prelucrare. Procedeele iterative care permit înlocuirea la fiecare pas a unei sume parțiale prin noua sumă parțială pentru calculul lui N și deci utilizarea rațională a spațiului de prelucrare conform formulei (1), se preferă calculului inversei unei matrici conform formulei (2).

În vederea unei prelucrări mai rapide se poate considera de asemenea inscrierea fișierului de structură (elementele matricii A) în două forme: elementele în ordinea liniilor și de asemenea și în ordinea coloanelor, fiecare element din fișier având informația corespunzătoare liniei și coloanei pe care se află. În cazul exemplului considerat, fișierul de structură în cele două forme va fi:

$\{1, (2,3), (3,2), (4,2); 3, (2,1); 5, (3,1), (4,2)\}$
(pe linii)

$\{2, (1,3), (3,1); 3, (1,2), (5,1); 4, (1,2), (5,2)\}$
(pe coloane)

Se poate arăta de asemenea că organizarea fișierului de structură corespunzător triangulizării matricii A , îmbunătățește de asemenea procesul de prelucrare, reducind numărul operațiilor de sortare care se execută cu acest fișier.

În afară de fișierul de structură discutat este necesar să se constituie și un fișier de articole, în care fiecărui articol să i se asocieze o serie de informații necesare pentru aplicațiile ulterioare privindordonanțarea și lansarea producției, calculul costului manoperei, necesarul de aprovizionat etc.

Pentru fiecare articol apare astfel o înregistrare care conține de pildă, următoarele cimpuri (zone):

- o_j — costul manoperei asamblării articoului j ;
- t_j — timpul necesar asamblării componentelor articoului j (pe un nivel);
- s_j — numărul articolelor j disponibile în stoc;
- I_j — informația de identificare a articoului j (numele său, adresele din fișierul de structură ale articoului j etc.).

Cu ajutorul acestor informații se pot calcula, de pildă, termenele la care trebuie fabricate, respectiv lansate în producție, diferitele repere și subansamble în vederea obținerii produselor finale la termenele de livrare planificate, precum și necesarul de aprovizionat cu un decalaj de timp convenabil înainte ca articoul necesar să trebuiască să fie lansat în producție.

Potem gîndi în vederea prelucrării, elementele de informație conținute în fiecare cimp al înregistrării ca fiind componente ale

unor vectori coloană cu n componente, unde n este numărul total al articolelor din fișier; astfel de pildă vom avea un vector Q al costurilor

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{pmatrix}$$

sau un vector al disponibilului din stoc

$$S = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix} \text{etc.}$$

Proiectarea fișierelor (de structură, de articole, etc.) formatul înregistrărilor, informațiile pe care trebuie să le conțină înregistrările etc., fac parte din proiectarea noului sistem informațional în întreprindere și trebuie să constituie rezultatul unei munci de analiză riguroase a tuturor aspectelor implicate.

Chiar această succintă prezentare a abordării unei aplicații relativ simple din întreprinderi sugerează *necesitatea imbinării permanente a elementelor de formalizare matematică cu caracteristicile organizării eficiente a datelor, caracteristici ce decurg din specificul activității întreprinderii*. * Această legătură continuă asigură condițiile favorabile ale rezolvării problemelor ce apar în fiecare etapă a trecerii aplicației respective pe SPAD și pune în lumină influența mare pe care metodele de organizare a datelor o au asupra modului de prelucrare și în ultimă instanță asupra elaborării algoritmelor (programelor), deși etapele principale ale acestora (în diferite variante) sunt conținute în modelul asociat. Menționăm că cele două tipuri de fișiere de bază necesare de fapt tuturor aplicațiilor din întreprinderi, fișierul de articole și fișierul de structură, reflectă din punct de vedere matematic, relația de ordine și respectiv relația de apartenență pe baza cărora aceste fișiere sint organizate.

* Această aplicație a fost implementată pe SPAD la uzinele pilot (construcții de mașini și chimie), în care s-au proiectat sisteme informaționale de prelucrarea datelor de către specialiști de la CEPECA în colaborare cu specialiștii din întreprinderile respective: Policolor, FMUAB și 23 August. Această aplicație se execută pe SPAD de asemenea la uzinele „Tractorul” din Brașov și în cadrul unor proiecte ale CCECE (folosind programele firmei IBM) și DCS (folosind programele firmei ICL).

2. Metode de organizare a datelor

2.1. Tehnici de organizare a fișierelor ; baze de date

Evoluția modurilor de organizarea datelor ilustrează direct influența pe care acestea au exercitat-o și o exercită asupra metodelor de prelucrare și asupra concepțiilor de programare și ca atare implicit asupra elaborării modelelor matematice asociate prelucrărilor.

După cum am văzut, de-a lungul dezvoltării generațiilor de calculatoare, memoriile pe benzi magnetice cu acces secvențial au apărut înaintea dispozitivelor cu acces direct de mare capacitate, dintre care cele mai reprezentative în prezent sunt discurile magnetice.

Pentru o prezentare de ansamblu a evoluției modurilor de organizare fișierelor să considerăm caracteristicile pe care trebuie să le indeplinească un fișier în vederea unei organizări eficiente a datelor :

— *mobilitatea*, care se referă la eliminarea unor înregistrări existente sau înscrierea de noi înregistrări în fișier. Un fișier este cu atât mai mobil cu cât se fac mai multe (și mai frecvente) ștergeri sau înscrieri de înregistrări în fișierul respectiv. Această caracteristică se referă de asemenea la creșterea potențială a fișierului respectiv.

— *gradul de activitate*, care se referă la numărul înregistrărilor care sunt prelucrate la fiecare rulare a programelor care utilizează acest fișier. Un fișier este cu atât mai activ, cu cât mai multe înregistrări ale sale sunt prelucrate în fiecare ocazie. Un fișier mai puțin activ va trebui organizat astfel ca accesul la înregistrări să fie direct întrucât localizarea lor în vederea prelucrării prin acces secvențial ar dura relativ mai mult, datorită parcurgerii înregistrărilor care nu trebuie localizate.

— *dimensiunea fișierului*, dacă fișierul este de dimensiuni relativ reduse modul de organizare nu prezintă o importanță deosebită, dar importanța crește tot mai mult odată cu mărirea volumului fișierului, ajungind ca în cazul fișierelor mari (pentru care de pildă numărul unităților de discuri sau benzi din configurația existentă nu permite să fie introduse simultan toate discurile sau benzile pe care este organizat fișierul) să se elaboreze metode specifice de organizare și prelucrare, sau să se utilizeze memoriile auxiliare de mare capacitate (cum sunt de pildă memoriile data-cell, organizate pe foi magnetice sau discuri de mare capacitate etc.). Necesitatea utilizării memoriilor auxiliare de mare capacitate apare în special pentru problemele ce trebuie rezolvate la nivel macroeconomic (planificare, asigurări sociale, comerț, transporturi, aprovizionare etc.), unde fișierele cu milioane de înregistrări sunt frecvente.

Din punct de vedere al prelucrării înregistrărilor dintr-un fișier în raport cu tranzacțiile — informațiile de intrare care se referă la aceste înregistrări (de pildă mișcările de cantități de materiale care trebuie operate în fișierul de materiale) — avem două procedee de bază :

— *prelucrarea secvențială*, în care tranzacțiile de intrare sunt grupate și ordonate astfel ca să se respecte ordinea înregistrărilor din fișier, înregistrările fiind prelucrate succesiv în ordinea în care sunt înscrise. Cind fișierul este realizat pe pachete de cartele sau benzi magnetice, ca atare pe suporturi fizice cu acces secvențial, acest tip de prelucrare este eficient. În cazul fișierelor mobile, acest tip de prelucrare pierde din eficiență, reclamind frecvent rescrierea întregului fișier cu acces secvențial ;

— *prelucrarea aleatoare*, în care tranzacțiile sunt folosite la prelucrarea înregistrărilor în ordinea în care apar. Este clar că dispozitivele cu acces direct, pe care fiecare înregistrare poate fi localizată rapid, direct (fără parcurgerea secvenței de înregistrări anterioare) prezintă o eficiență mult mai mare pentru acest tip de prelucrare.

În evoluția aplicațiilor din întreprinderi, prelucrarea aleatoare capătă o pondere tot mai mare ; acest aspect este deosebit de semnificativ în cazul tranzacțiilor care se transmit prin dispozitive de comandă de la distanță, în care elementul aleator este predominant.

Apariția dispozitivelor cu acces direct a permis alegerea celei mai potrivite metode de organizare și prelucrare a datelor (secvențială

sau aleatoare) pentru aplicațiile ce se execută; prin aceasta eficiența rezolvării problemelor care cer manevrarea unor volume mari de date a crescut mult, pe baza creșterii diversității metodelor de proiectare a sistemelor informaționale; a crescut de asemenea gama rapoartelor ce pot fi prezentate conducerii, în special prin mărirea numărului de rapoarte pentru excepții.

În prezent, se utilizează, deci, patru tipuri principale de organizări ale fișierelor:

a. **Organizarea secvențială**, în care înregistrările sunt organizate pe baza succesiunii lor fizice pe suportul respectiv (pachete de cartele, benzi magnetice, discuri magnetice etc.); ca atare înregistrările sunt ordonate de pildă în ordinea crescătoare a numerelor ce alcătuiesc informațiile de identificare. Înregistrările sunt extrase sau puse la zi conform tranzacțiilor de intrare, în ordinea în care se află în cadrul fișierului. O înregistrare anumită nu poate fi localizată repede, localizarea unei înregistrări făcindu-se la majoritatea suportilor utilizati prin parcurgerea informațiilor de identificare ale înregistrărilor anterioare (posteroare) și compararea lor cu informația de identificare a înregistrării ce urmează a fi localizată. Acest mod de organizare se aplică, de pildă, în cazul fișierelor cu grad de activitate mare, cind la fiecare rulare a programului majoritatea înregistrărilor sunt supuse prelucrării.

b. **Organizarea partionată** (pe discuri) în care înregistrările fișierului sunt grupate în partiții, fiecare partiție fiind localizată pentru prelucrare prin numele său. Partițiiile pot fi eliminate din fișier sau adăugate conform necesităților aplicației. Înregistrările în cadrul fiecarei partiții sunt organizate secvențial.

Organizarea partionată se folosește în principal pentru organizarea fișierelor de programe sau subprograme și diferite tabele. Pentru programele care sunt folosite des în cazul unor aplicații se alcătuiesc fișiere pe benzi sau discuri cu aceste programe, exploatarea acestora fiind mult imbunătățită față de condițiile normale de exploatare ale programelor care sunt introduse succesiv pe pachete de cartele; această operație de catalogare a programelor mărește mult eficiența exploatarii sistemului.

Se utilizează tot mai mult pachete de programe pentru rezolvarea unei aplicații, pachetul de programe luând locul programelor cu cîteva sute sau mii de instrucțiuni care rezolvă anumite probleme particulare, mod de lucru cîntîn în cadrul primelor două generații de calculatoare.

Prin *pachet* (de programe) se înțelege o colecție de programe elaborate în limbajul mașinii sau într-un limbaj de nivel superior (COBOL, FORTRAN etc.), care permite diferite opțiuni utilizatorului prin cartele parametri și care realizează prelucrările necesare unor activități standard de aplicații (producție, stocuri etc.), de organizare și generare a fișierelor, de interogare (regăsirea informației) de raportare, de validare, de constituire a datelor istorice necesare prelucrărilor statistice și calculelor previzionale etc.

c. **Organizarea indexat-secvențială** (pe discuri), permite atât prelucrarea secvențială, cât și aleatoare. Înregistrările în secvență fizică pot fi localizate datorită informațiilor de identificare a înregistrărilor, care sunt în corespondență cu adresele lor fizice și datorită indexurilor asociate pentru cilindri și piste * acest mod de organizare permitând și prelucrarea aleatoare.

In cadrul acestei organizări se creează o zonă separată a fișierului în care se pot adăuga înregistrări noi, care deși nu mai respectă ordinea secvențială a adreselor fizice, pot fi localizate individual și înălțuite într-o prelucrare secvențială cu ajutorul indecsilor și informațiilor privind adresa înregistrării care urmează, adresă care se află în cadrul informațiilor de identificare ale înregistrării curente. Crearea acestei zone elimină dezavantajul de refacere a întregului fișier (ca de pildă în cazul unui fișier organizat secvențial), atunci cind se adaugă înregistrări noi.

Pentru această metodă de organizare, sistemul de operare este prevăzut cu posibilitatea elaborării automate a indecsilor pentru cilindri și piste în procesul de constituire al fișierului.

Să ilustrăm organizarea indexat secvențială pe un exemplu. Să considerăm că, fișierul care urmează să fie organizat indexat secvențial este ordonat în ordinea crescătoare a adreselor înregistrărilor. Să presupunem că fiecare înregistrare fizică are 25 înregistrări logice. Vom crea un tablou pe disc care conține pentru fiecare înregistrare fizică, adresa înregistrării fizice (în ordinea numerelor naturale) și

* Un pachet de discuri (IBM 2311) format din 6 discuri are 10 feje active, pe fiecare față fiind 200 de piste; cîte 10 piste avind același diametru alcătuiesc ceea ce numim un *cilindru*; astfel că pe pachetul de discuri avem 200 de cilindri. Menționăm că pistele nu formează o spirală ca la plăcile de pick-up, ci sunt cercuri concentrice pe fiecare față. Evident că pentru localizarea unei înregistrări trebuie localizat întîi cilindrul și apoi pista de cilindru pe care se află înregistrarea.

adresa ultimei înregistrări logice (în ordine crescătoare) din înregistrarea fizică respectivă :

AF	AL	AF	AL	AF	AL	
00001	30	00002	71	00003	99

Pentru a obține acces la o înregistrare logică de adresă cunoscută este deci necesar :

1. Să se citească în memoria internă tabloul, pentru a vedea din ce înregistrare fizică face parte.

2. Să se transfere înregistrarea fizică în memoria internă și să se localizeze înregistrarea logică dorită.

Dacă fișierul are dimensiuni mari atunci tabloul de adrese poate fi prea mare și manipularea prea greoie. În acest caz, se alcătuiesc tablouri ierarhizate, rezervind, de pildă în cazul discurilor, pe fiecare cilindru din disc un număr de piste pentru constituirea unor tablouri secundare care să prezinte repartitia blocurilor și înregistrărilor logice pe pistele din cilindrul respectiv.

Pe un cilindru determinat, se constituie un tablou primar, compus din perechi formate din adresa unui cilindru și adresa ultimei înregistrări logice conținute în fiecare cilindru.

Accesul la o înregistrare logică se face în cazul tablourilor ierarhizate pentru discuri astfel :

1. Se citește tabloul primar în memoria internă, unde se determină cilindrul în care se află înregistrarea logică dorită.

2. Se citește în memoria internă tabloul secundar de pe cilindrul corespunzător și se determină pistă și înregistrarea fizică din care face parte înregistrarea logică dorită.

3. Se citește în memoria internă înregistrarea fizică determinată și se localizează înregistrarea logică dorită.

În cazul cînd fișierul este constituit pe mai multe benzi magnetice sau discuri (pe mai multe *volum*e), se alcătuiește și un tablou care să permită determinarea volumului în care se găsește înregistrarea ce urmează să fie localizată, procedeul continuind după cum a fost descris mai sus pentru localizarea blocului și a înregistrării.

Structura descrisă permite accesul la orice înregistrare din fișier pentru a efectua operațiile comune de consultare și punere la zi, adică modificări, adăugiri și suprimări de informații din fișier. Pentru punerea la zi (actualizarea) apare însă frecvent necesitatea de

a inseră o nouă înregistrare logică între două înregistrări (logice) L_t și L_{t+1} , între care nu mai există spațiu pe suportul respectiv. Pentru a se rezolva această problemă se utilizează următoarea tehnică de înlătuire :

Se alocă pentru fiecare cilindru un număr de înregistrări fizice de depășire, care atunci cînd se încarcă fișierul prima oară pe disc nu conțin nici o informație. La fiecare înregistrare logică L_t se asociază o adresă de înlătuire I_t , care este nulă dacă înregistrarea următoare L_{t+1} se află fizic înregistrată imediat după L_t sau care conține adresa înregistrării fizice de depășire din care face parte L_{t+1} , în caz contrar.

Să presupunem că două înregistrări L_t și L_{t+1} , avind adresele A_t și A_{t+1} cu $A_t < A_{t+1}$, sunt vecine într-o înregistrare fizică de adresă 00235.

00235	A_t	L_t	$I_t=0$		A_{t+1}	L_{t+1}	I_{t+1}
-------	-------	-------	---------	--	-----------	-----------	-----------

Se observă că adresa de înlătuire $I_t = 0$.

Să presupunem că pentru actualizarea fișierului trebuie să plăsăm o înregistrare L_p de adresă A_p , astfel ca $A_t < A_p < A_{t+1}$.

Să presupunem că vom utiliza adresa 04630 a unei înregistrări de depășire pentru a inscrie L_p :

00235	A_t	L_t	$I_t=04630$		A_{t+1}	L_{t+1}	I_{t+1}
-------	-------	-------	-------------	--	-----------	-----------	-----------

04630	A_p	L_p	$I_p=00235$
-------	-------	-------	-------------

Se observă că adresa de înlătuire a înregistrării L_p este adresa înregistrării fizice care conține L_{t+1} .

d. Organizarea aleatoare (directă), în care există o corespondență între informațiile de identificare ale înregistrărilor și adresele fizice ale acestora. Această corespondență este stabilită de programator prin formule sau în tabele de corespondență sau este dată automat de proceduri standard ; vom descrie mai departe unele moduri de organizare aleatoare a fișierelor care, dat fiind caracterul aplicației, nu permit o organizare eficientă secvențială sau indexat-secvențială

sau care sunt utilizate în aplicațiile în care viteza de localizare a fiecarei înregistrări este criteriu dominant.

În vederea unei organizări cit mai apropiate de modul de prelucrare despre care presupunem că are un caracter important aleator, inscrierea înregistrărilor pe piste și cilindri se face cel mai adesea în mod aleator. Pentru a înțelege ideile de bază legate de această tehnică, să considerăm cazul a 3 piste (A), (B), (C), pe care se înscriu în mod aleator înregistrări printr-o metodă oarecare.

Vom înscrie pe pista (A) acele înregistrări care prin metoda aleatoare utilizată capătă, de pildă, aceleași adrese fizice ca înregistrările anterioare de pe pista respectivă; pe pista (B) analog etc. Procedeul va conduce la completarea capacitatii pistei și la necesitatea utilizării unor tehnici de înlătuire (analoge celor din cazul organizării indexat secvențiale). În vederea înlătuirii pistelor și ca atare a înregistrărilor, prima înregistrare din fiecare pistă va conține informația pentru înlătuire, de exemplu, adresa pistei următoare (obținută, de pildă, tot aleator din multimea pistelor libere).

Să presupunem că secvența în care au apărut înregistrările pentru a fi înschise pe piste este: A₁, B₁, C₁, A₂, B₂, C₂, A₃, A₄, A₅, A₆, C₃, A₇, unde înregistrările subliniate au fost înschise prin înlătuire după depășirea capacitatilor pistelor respective.

(A)	C	A ₁	A ₂	A ₃	A ₄
(B)		B ₁	B ₂	C ₃	A ₇
(C)	B	C ₁	C ₂	A ₅	A ₆

Înregistrările au fost distribuite pe piste conform tabelei alăturată. Se observă că dacă o înregistrare „A“ nu se află pe pista (A), atunci prima pistă pe care este căutată este pista (C) a cărei adresă nu se află în prima înregistrare a pistei (A). Dacă înregistrarea „A“ nu se află nici pe pista (C), atunci se caută pe pista (B) etc. Dacă o înregistrare „B“ nu se află pe pista (B), atunci nu se mai explorează altă pistă, deoarece absența adresei din prima înregistrare a pistei (B) arată că nu mai există înregistrări „B“.

Tehnicile aleatoare pentru găsirea pistelor de înlătuire și ale adreselor înregistrărilor și de organizare aleatoare variază foarte mult și cel mai adesea numai caracterul aplicației este cel care decide alegerea.

O metodă utilizată frecvent pentru calculul adreselor constă în folosirea claselor de resturi ce se obțin, de pildă, prin împărțirea numerelor care reprezintă informațiile de identificare ale înregis-

trărilor la numărul ce reprezintă ultima adresă a înregistrărilor ce urmează să fie înschise în fișier. Astfel, de exemplu, să considerăm că vrem să formăm un fișier cu 40 000 de înregistrări logice, ca atare vrem să înscriem aceste înregistrări de la adresa 0 la adresa 39.999. Să presupunem că informația de identificare a fiecarei înregistrări este formată din numere de 8 cifre. Împărțind numărul ce reprezintă informația de identificare a fiecarei înregistrări la 39.999 obținem un rest care va fi adresa înregistrării; desigur, pentru mai multe înregistrări se poate obține aceeași adresă și ca atare atunci cînd este necesar se vor folosi tehnici de înlătuire menționate.

În sistemele de operare există programe utilitare pentru scrierea și citirea fișierelor organizate prin orice metodă, programatorului revenind-i sarcina să dea informațiile necesare utilizării acestor programe utilitare care vor fi folosite de programul elaborat ori de cîte ori în acesta apar instrucțiunile care activează programele utilitare.

Ca și în cazul organizării partionate, programele pot fi înschise pe dispozitive cu acces direct, diferite subprograme sau porțiuni de program fiind transferate în memoria rapidă pe măsură ce necesitățile de prelucrare o cer. Avantajele pe care le prezintă organizarea și prelucrarea aleatoare a pachetelor mari de programe și în special a sistemului de operare sint foarte mari, ele reflectindu-se în primul rînd în economisirea memoriei principale, în aceasta fiind transferate pe rînd numai acele programe de care este nevoie într-un anumit pas al prelucrării.

Pe măsură ce au început să se elaboreze sistemele integrate de prelucrarea informațiilor în întreprinderi, diversitatea aplicațiilor care recurg la aceleași fișiere de bază (fișierul de articole, fișierul de structură, fișierul de operații tehnologice etc.) a condus de asemenea la necesitatea utilizării în măsură tot mai mare a dispozitivelor cu acces direct și a prelucrării aleatoare, aspecte care au influențat elaborarea programelor pentru aceste aplicații. Limitarea pe care o prezintau dispozitivele cu acces secvențial în ceea ce privește accesul la informații individuale în mijlocul prelucrărilor a fost înălțată, sistemul informațional devenind tot mai elastic, iar programele de prelucrare fiind adaptate în consecință; sistemul informațional folosind prelucrarea automată a datelor, poate răspunde operativ în măsură tot mai mare la orice întrebări pe care și le pune conducerea întreprinderii în legătură cu informații individuale privind aplicațiile de planificare, ordonanțare, lansare și control al producției,

inventarul stocurilor, calculele finanțier-contabile etc.: cite repere din ansamblul X se află pe linie? Ce cantitate din materialul Y se află în magazie? Care este cîștigul pe ultimele 6 luni ale salariațului Z? Care este costul manoperei pentru subansamblul W? Care este balanța de cheltuieli și venituri? Care sunt abaterile la termenele de livrare? etc.

Evoluția rapidă a fișierelor, datorită schimbării planului de producție, asimilarea de noi produse sau introducerea de noi posturi de lucru, schimbarea unor criterii de calcul privind probleme finanțier-contabile, adăugarea sau schimbarea unor materiale din fișierul de stocuri, în general adaptarea sistemului informațional la dinamica întreprinderii, pune în valoare și necesită utilizarea pe scară tot mai largă a dispozitivelor cu acces direct.

Dezvoltarea metodelor de organizare a fișierelor a influențat de asemenea și, practic, putem spune că a schimbat concepția despre exploatarea (operarea) programelor.

În ultima vreme, se alcătuiesc tot mai frecvent pachete de cartele de comandă (job-stream) pentru prelucrarea programelor catalogate (care se află în biblioteca sistemului de operare în limbaj mașină) sau care sunt inscrise în memoriile auxiliare și care practic permit o utilizare mai eficientă a SPAD, eliminând la maximum intervenția operatorului.

În ce privește organizarea fișierelor mari (cu sute de mii sau milioane de înregistrări)*, metodele elaborate, deși unele în exploatare, sunt încă la începutul dezvoltării lor.

Un aspect care s-a impus în legătură cu organizarea fișierelor mari se datorează gradului relativ redus de activitate a acestor fișiere mari; într-adevăr, la un fișier cu circa 1 milion de înregistrări, este comun să se efectueze zilnic cîteva mii sau cel mult zeci de mii de tranzacții; această imprejurare a condus la necesitatea organizării aleatoare a fișierelor mari în majoritatea aplicațiilor.

Desigur, metoda tabelelor de corespondență existente în memoria principală sau pe discuri, pentru indexuri pe diferite nivele (pachete de discuri, cilindri, piste etc.), care permit localizarea rapidă a înregistrărilor (ca în cazul organizării indexat-secvențiale sau aleatoare).

* Dacă considerăm un fișier în acces direct cu înregistrări fixe de cite 200 de caractere fiecare, pentru stocarea unui milion de înregistrări, avem nevoie de o memorie auxiliară pe discuri cu o capacitate de 200 milioane caractere. Ca termen de comparație, 4 unități de discuri IBM 2311 au o capacitate de circa 30 milioane caractere, iar o unitate 2314 are circa 200 milioane

are de asemenea încă mare importanță în exploatarea fișierelor mari. Diferite tehnici particulare de organizare și de acces se asociază de obicei celor două tipuri de metode (aleatoare și tabele de corespondență).

Că o sinteză a aspectelor prezentate, pentru organizarea și prelucrarea fișierelor mari se preferă adesea elaborarea unor pachete mari de programe, care să conțină și unele elemente din sistemul de operare; de asemenei, se organizează pe scară tot mai largă colecții de fișiere, numite *baze de date*, care elimină repetările de informații și asigură posibilitatea înlanțuirilor în prelucrarea înregistrărilor, constituind sursa comună de informații pentru toate subsistemele de aplicații din sistem.

Pachetele de programe care generează, organizează și întrețin bazele de date au o importanță crescindă, utilizarea lor asigurând pe de o parte o reducere importantă a muncii de proiectare și programare a fiecărui sistem informațional individual, iar pe de altă parte, permitind compatibilitatea între subsistemele unui sistem și între sisteme, acest ultim caz fiind condiționat de o independență cit mai mare a pachetului de generare și organizare a bazei de date de sistemele de operare ale diferitelor calculatoare. Aceste tendințe sunt cu atât mai semnificative în cazul aplicațiilor *on-line* (care exclud operatorul uman din operațiile de introducere și extragere a datelor în opoziție cu aplicațiile *off-line*) în care se folosesc echipamente de comandă la distanță, cum este cazul celor mai multe aplicații din sistemul bancar, el asigurărilor sociale și de rezervarea locurilor în mările organizații de transport etc.

2.2. Tipuri de fișiere și prelucrări ale fișierelor

Făcind o paralelă între activitatea dintr-o întreprindere și procesul de prelucrare automată a datelor putem considera că activitatea de pregătire și organizare a datelor are o pondere comparabilă, față de activitatea propriu-zisă de prelucrare pe SPAD, cu activitățile de aprovisionare, planificare, programare, lansare etc., față de activitatea de fabricație. Prin urmare rolul organizării datelor este major în procesul de prelucrare a informațiilor și a condus, după cum am văzut, la modificări importante în concepția despre progra-

mare, constituind în prezent factorul determinant în proiectarea de noi sisteme informaționale în întreprinderi. Elaborarea fișierelor și a schemelor de sistem (cap. III), în care se descriu prelucrările ce se efectuează cu aceste fișiere, informațiile de intrare necesare și rapoartele ce sunt produse de sistemul de prelucrare automată a datelor, constituie obiectivul principal al muncii analistului de sisteme în cadrul proiectării noului sistem.

În funcție de utilizarea lor, ca atare în funcție de scopul pentru care sunt create, fișierele pot fi clasificate astfel:

a. **Fișiere de bază sau permanente.** Aceste fișiere conțin informațiile necesare pentru procesul de prelucrare și sunt „consultate” frecvent în cadrul acestui proces. Informațiile necesare prelucrării pot proveni din documentele primare completele manual sau din rezultatele prelucrării automate. Frecvența de actualizare a acestor fișiere, ale căror înregistrări conțin în principal informații de referință, este relativ mică; unele fișiere (de exemplu articole sau structuri) pot fi actualizate lunar sau trimestrial. Menționăm că între un fișier manual (cartele perforate, fișe clasice) și un fișier constituit pe suport magnetic este o diferență funcțională importantă în avanțajul celor din urmă: fișierele manuale rețin obligatoriu „istoric” tranzacțiile, pe cind pe suportul magnetic se pot efectua comod ștergeri ale vechilor informații, atunci cind este cazul, acestea din urmă pretindu-se de asemenea relativ ușor la recopiere. Pentru a mări viteza de prelucrare a fișierelor, evident este recomandabil ca informațiile desuete să fie eliminate pentru a putea lucra cu un volum cît mai mic și mai compact de informații. Conținând în general informații de referință, cu caracter calitativ, modificările (actualizările) în aceste fișiere se fac numai ca urmare a deciziilor conduceții.

b. **Fișiere variabile.** Ele au o existență relativ mai scurtă, în general, și servesc la înmagazinarea datelor cantitative care sunt actualizate mult mai frecvent decât informațiile în general calitative din înregistrările fișierelor de bază.

De exemplu, fișierul de stocuri pentru produse finite. De asemenea fișierul noilor comenzi este un fișier variabil, care servește la actualizarea, atât a fișierului de stocuri de produse finite (în înregistrările respective este necesar să existe și un cimp afectat numărului de produse comandate), cît și a fișierului de comenzi în aşteptare etc.

Uneori, fișierele variabile servesc ca bază de informații pentru imprimarea rezultatelor care rezultă din prelucrare: ordine de expediție, facturi, buletine de plată etc.

Actualizările acestor fișiere se fac, în general, în urma acțiunii tranzacțiilor din cadrul subsistemelor de aplicații.

c. **Fișiere auxiliare.** Au o existență temporară și servesc în general la constituirea unei mulțimi de rezultate intermediare necesare în procesul de prelucrare. Aceste fișiere se pot constitui pe orice suport de memorie. Fișierele auxiliare, care se constituie, de pildă, într-un proces deordonare pe bază de sortări repetitive ale unui fișier inscris pe disc, se pot inscrie de asemenea pe disc. Adesea este dificil și în general neesențial să se distingă între unele tipuri de fișiere; de pildă, nu este esențial ca un fișier de tranzacții să producă discuții dacă este variabil sau auxiliar.

Din punct de vedere al exploatarii lor, fișierele care sunt utilizate pentru un anumit scop (permanente, variabile sau auxiliare) am văzut că pot fi clasificate în fișiere care sunt exploataate în mod secvențial sau aleator. Pentru a putea fi exploataate, înregistrările din fișier trebuie aranjate într-un mod convenabil prelucrării. Situațiile în care se pot prezenta fișierele din acest punct de vedere sint următoarele:

a. **Fișiere ordonate.** Înregistrările, ca elemente ale unei mulțimi ordonate (fișierul ordonat) se află în ordinea dată de un anumit criteriu (ordinea crescătoare sau descrescătoare a numerelor naturale, ordinea alfabetică etc.). De exemplu, un fișier de comenzi ordonat după ordinea crescătoare a numerelor de comandă.

b. **Fișiere parțial ordonate.** Înregistrările se constituie în submulțimi care sunt ordonate, în cadrul unei submulțimi putindu-se folosi diferite criterii de ordonare cind este nevoie. Astfel, de exemplu, dacă avem fișierul buletinelor de plată pe un an de zile, ordonat pe luni, acesta este parțial ordonat din punct de vedere al numerelor salariaților, chiar dacă în cadrul fiecărei luni buletinele sunt scrise în ordinea alfabetică.

c. **Fișiere neordonate.** Înregistrările dintr-un fișier se pot prezenta în raport cu un anumit criteriu fără nici o ordine, chiar dacă ele sunt ordonate sau parțial ordonate în raport cu un alt criteriu.

Am văzut că asupra datelor se execută operații de selecție, sortare și ordonare. Ele se regăsesc în operațiile care se efectuează asupra înregistrărilor din fișiere.

În mod specific, cu fișierele se mai execută operații de fuziuni a două sau mai multe fișiere și actualizări, prin fuziune înțelegindu-se reunirea fișierelor și ordonarea înregistrărilor după un criteriu comun.

Să ilustrăm pe un exemplu, operațiile cu fișierele :

Fiind dat un fișier variabil din care vrem să efectuăm o extragere de înregistrări și un fișier auxiliar, care conține lista de informații de identificare ale înregistrărilor ce urmează să fie extrase pentru efectuarea rapidă a operației de selectare, este necesar ca ambele fișiere să fie ordonate.

Să considerăm un fișier în care înregistrările au ca informații de identificare numărul articolului și numele lui.

145	Savoniere
026	Săpun
115	Parfum
125	Periuțe
549	Pudră
083	Cremă
412	Pastă

Acest fișier format din 7 articole este neordonat, atât din punct de vedere al criteriului numărului de articol, cît și din punct de vedere al criteriului alfabetic.

Ordonarea după numărul de articole se face prin sortări successive următoarele reuniriile submulțimilor din partitie; sortările se execută după poziția unităților, apoi a zecilor, a sutelor și.a.m.d., iar după fiecare sortare, reuniunea submulțimilor se face în ordinea crescătoare a cifrelor criteriului :

			145						
			115						
0	1	2	412	083	125	016			549

Reunind mulțimile în ordinea crescătoare a cifrelor de pe poziția unităților obținem :

549
026
145
115
125
083
412

Sortarea după coloana zecilor :

				549					
				115					
0	1	2	412	026	145				

2.2. TIPURI DE FIŞIERE ŞI PRELUCRARI ALE FIŞIERELOR

Reunirea submulțimilor ordonate după cifra zecilor crează fișierul :

083
549
145
026
125
115
412

Sortarea după coloana sutelor :

			145						
0	1	2	3	4	5	6	7	8	9

Obținem prin fuziune fișierul ordonat după numărul articolului :

549
412
145
125
115
083
026

Se observă că în operațiile de sortare numerele de articol se comportă ca și cum ar fi pachete de cartele pe care operatorul le manevrează, efectuând reuniunea după fiecare sortare.

Din fișierul ordonat vom selecta articolele corespunzătoare listei reprezentate de fișierul auxiliar de mai jos :

<i>Fișierul variabil</i>	<i>Fișierul auxiliar</i>	<i>Fișierul extras</i>
026 Săpun	02	026 Săpun
083 Cremă	14	145 Savoniere
115 Parfum	41	412 Pastă
125 Periuțe		
145 Savoniere		
412 Pastă		
549 Pudră		

Pentru a ilustra operația de fuziune, să considerăm două fișiere care prin fuziune vor crea fișierul variabil anterior.

026 Săpun	083 Cremă	026 Săpun
115 Parfum	412 Pastă	083 Cremă
125 Periuțe		115 Parfum
145 Savoniere		125 Periuțe
549 Pudră		145 Savoniere
		412 Pastă
		549 Pudră

Fuziune →

Actualizarea unui fișier permanent sau variabil se efectuează cu ajutorul unui fișier auxiliar, după următoarele reguli :

a. Orice înregistrare care nu are în corespondență (prin informația de identificare) o înregistrare din fișierul auxiliar (tranzacție) este copiată în fișierul pus la zi.

b. Orice înregistrare care are în corespondență o înregistrare din fișierul auxiliar, care conține un cod de ștergere nu este trecută în fișierul la zi.

c. Orice înregistrare din fișierul auxiliar, care nu conține codul de ștergere, este trecută în fișierul la zi, indiferent dacă are sau nu în corespondență o înregistrare în fișierul permanent sau variabil.

d. Orice înregistrare din fișierul permanent sau variabil care are în corespondență o înregistrare în fișierul auxiliar, care nu conține codul de ștergere nu este trecută în fișierul la zi.

Să considerăm următorul exemplu, în care ilustrăm aplicarea regulilor de mai sus.

Vom considera că între înregistrările dintr-un fișier permanent și un fișier auxiliar, corespondența se stabilește prin denumirea articolelor, ordonarea se face în ordine alfabetică a denumirilor, iar actualizarea se efectuează asupra codului articoului și prin adăugiri sau ștergeri de înregistrări.

Fie „S“ codul de ștergere :

Fișier permanent

083 Cremă
115 Parfum
412 Pastă
125 Periuțe
549 Pudră
026 Săpun
145 Savoniere

Fișier auxiliar

623 Lame
415 Pastă
125 Periuțe
436 Piepten
029 Săpun

Fișier la zi

083 Cremă
623 Lame
115 Parfum
415 Pastă
125 Periuțe
436 Piepten
549 Pudră
029 Săpun
145 Savoniere

Să considerăm un alt exemplu de actualizare a unui fișier variabil, în care să descriem cimpurile din înregistrare care urmează a fi puse la zi. Să considerăm o înregistrare dintr-un fișier de stocuri cu cimpurile din fig. 2.2.1. Să presupunem că este vorba de articolul nr. 24357.

24357	Data ultimelui mișcări : 08.07.71			
Punct de comandă 330	Serie de comandat 200	Stoc existent 1230	Nr. de articole nelivrante 2210	În curs de fabricație 370

Fig. 2.2.1. Înregistrarea dintr-un fișier de stocuri.

Conform acestei înregistrări se observă că trebuie livrate încă 2210 articole nr. 24357 ; o parte dintre ele se află în stoc și altele în curs de fabricație. „Stocul existent“ arată numărul de articole finite ce sunt gata pentru livrare. „Articolele în curs de fabricație“ sunt articole lansate, dar care nu au intrat încă în magazia de produse finite. Pentru a satisface comanda este necesar să se lanseze un număr de articole care să fie un multiplu al numărului reprezentat de „Serie de comandat“ care este 200, această serie de comandat reprezentând producția pe un schimb. Numărul din cimpul „Punct de comandă“ reprezintă cu cat conducerea fabricii își propune să reducă termenul de livrare, exprimat în număr de articole, în caz contrar cimpul fiind 0.

O comandă nouă mărește numărul din cimpul „Număr de articole nelivrante“ ; o expediție micșorează „Stocul existent“ și „Numărul de articole nelivrante“ ; orice ieșire din fabricație mărește „Stocul existent“ și micșorează corespunzător „În curs de fabricație“. Un ordin de fabricație mărește „În curs de fabricație“.

Situatia care rezultă din înregistrarea considerată este :

- urmează să fie livrate 2210 articole
- în magazie se află 1230 articole
- rezultă că urmează să fie fabricate $2210 - 1230 = 980$ articole
- adăugind „Punctul de comandă“ obținem $980 + 330 = 1310$ articole care urmează să fie fabricate.

Deoarece, 370 articole sint în curs de fabricație, urmează să fie lansate $1310 - 370 = 940$ articole cu nr. 24357, ca atare, cinci serii

ă 200, adică 1000 de articole. Cantitatea în curs de fabricație va veni în fișierul la zi, prin lansare, 1370 de articole.

Informațiile de identificare după cum am văzut pot fi alfano-mericice (constituuite din litere și cifre). Pentru calculul corespondenței dintre informațiile de identificare și adresele înregistrărilor, pentru generarea aleatoare a adreselor (de pildă folosind clasele de resturi sau formule pentru generarea numerelor aleatoare) sau pentru operații care se execută cu fișierele, este necesar uneori ca să se transforme literele, cu ajutorul unui cod, în numere și reciproc. Un exemplu simplu de astfel de cod este codul modulo 23, care asociază literelor numerale de la 0 la 22 : (A,0), (B,1), ..., (V,19), (W,20), (X,21), (Y,22). Reciproc, unui număr, de pildă 15231, îi corespunde prin împărțire cu 23 restul 5, ca atare litera F.

2.3. Codificarea datelor

Codificarea elementelor (produse, subansamblu, repere, materiale etc.) din sistemul de prelucrare a datelor din întreprinderi, care apar în cimpurile înregistrărilor, este o operație strins legată de modul de organizare a datelor în fișiere.

Termenul „cod“ are două semnificații și anume aceea de funcție care aplică o mulțime E de elemente oarecare („de codificat“) într-o mulțime K^* de simboluri sau aceea de element al mulțimii K . De obicei, una din aceste două semnificații rezultă din context.

Rolul codului constă în a realiza posibilitatea unei referiri eficiente la elementele mulțimii E , utilizând elementele mulțimii K . Bineînțeles aceasta presupune că, din anumite puncte de vedere, elementele mulțimii K sunt „mai simple“ decât elementele mulțimii E . De exemplu, dacă se consideră mulțimea salariaților dintr-o întreprindere și se asociază lui „Popescu N. Ion“ (ca element al mulțimii E) numărul matricol „1254“ (cod element al mulțimii K), este evident că referirile la respectivul salariat pot fi, într-un anume context, mai simplu prin cod decât prin nume-prenume.

Această referire se face în mod diferit în cazul unor sisteme manuale față de cazul unor sisteme mecanizate sau automatizate;

^{*)} Numărul elementelor constitutive ale unui simbol din K se numește lungimea simbolului.

în aceste cazuri, codurile sunt diferențiate, fiind influențate de posibilitățile de regăsire a informației cu mijloacele specifice fiecărui sistem.

Astfel, în cadrul sistemelor manuale, un cod va fi mai eficient în măsura în care codurile sunt semnificative, ceea ce permite o decodificare (aplicația $K \rightarrow E$) fără apelare la un fișier special.

Mai mult decât atât, în cazul unor sisteme manuale în care mulțimea E are un număr suficient de mic de elemente, este evident că operația de codificare poate să fie inutilă. Oportunitatea codificării se poate decide în funcție și de următoarele elemente :

- a. costul stabilirii codului (ca regulă de punere în corespondență);
- b. costul codificării propriu-zise a elementelor (aplicarea regulii);
- c. costul validării codurilor, deci a verificării corectitudinii lor;
- d. costul decodificării, prin confruntare cu fișierul de referință;
- e. costul întreținerii (actualizării) fișierului de referință;
- f. costul întreținerii instrucțiunilor de codificare.

Dacă distingem la cod cele două aspecte ale lui și anume cel semantic (conținutul informațional) și cel referitor la reprezentarea informației sub formă simbolică (suport), cele afirmate mai sus se pot exprima prin aceea că suportul este mai simplu decât semantică. În afară de simplificare, codul corespunde tendinței de unificare, asociind un singur suport unei semantică anumite, și de specializare, unui anume suport corespunzând o semantică specifică. Altfel spus, corespondența $E \rightarrow K$ este biunivocă.

Complexitatea structurii de codificare depinde de complexitatea relațiilor dintre elementele care trebuie codificate.

Alegerea unui sistem de codificare trebuie să se bazeze pe definirea precisă a scopului pentru care este destinat. Necesațile de informare și documentele necesare conducerii formează fundamentul unui plan de codificare și informațiile trebuie analizate temeinic, clasificate și identificate în concordanță cu acest plan, înainte de a se trece la stabilirea unui cod propriu-zis.

Pentru codificarea informațiilor există multe metode, diferențiate prin caracteristicile lor. Codul poate fi alfabetic, numeric sau o combinație a ambelor. Poate reprezenta numai o atribuire de numere corespunzătoare unor elemente inscrise pe o listă sau poate fi format dintr-un model mai complicat de numere care să identifice anumite elemente, pe formații sau grupe (materiile prime, unelte etc.). Indiferent de metoda folosită, codul trebuie să fie ales după eficiență sa de a prezenta informațiile într-o formă sistematică și cit mai ușor de înțeles.

Folosirea sistemelor de prelucrare automată a datelor în întreprinderi aduce de obicei în centrul atenției necesitatea revizuirii sau elaborării unor proceduri noi pentru clasificarea și codificarea datelor respective. Conducerea întreprinderii depinde de datele curente care constituie baza pe care se iau deciziile de execuție.

Înainte de a alege și a aplica o metodă de codificare este necesar să se stabilească un plan de clasificare al datelor. Planul de clasificare se referă la gruparea datelor asemănătoare într-un astfel de mod încit să convină cît mai bine celui care îl va folosi. Pentru corespondență, de exemplu, se vor clasifica numele și adresele în mod logic în fișierul principal după state, orașe și oficii poștale conform cerințelor poștale. Mai departe, se poate proceda la clasificarea după numele de botez, inițiale, stradă, adresă etc. în vederea păstrării unui fișier pe abonați. Trebuie avute în vedere la întocmirea planului de clasificare, toate utilizările care se vor da datelor, astfel încit clasificările principale și subîmpărțirile să fie făcute pentru a satisface toate nevoile.

Pentru a face aceasta, trebuie luat în considerare mai mulți factori :

1. Cine răspunde de păstrarea datelor și cine le va folosi ?
2. La ce se folosesc datele și în ce ordine sau cît de amănunțit trebuie prezentate pentru fiecare din cei care le vor folosi ?
3. Ce cantitate de date este cerută de fiecare din cei interesați ?
4. Cu ce frecvență sunt folosite datele și care este ordinea priorităților ?
5. Vor fi oare aceste date în măsură să satisfacă cerințele viitoare anticipate a fiecărui dintre cei care le folosesc ?
6. Ce metode vor fi folosite pentru organizarea și prelucrarea datelor ?

Gruparea datelor conform unei structuri de clasificare trebuie să fie realizată astfel încit grupurile de articole asemănătoare să poată fi identificate după aceleași caracteristici.

De asemenea, trebuie avut în vedere ca numărul familiilor să conducă la o repartizare cît mai omogenă a numărului elementelor în familii.

După ce s-a elaborat modul de clasificare și elementele fiecărei date au fost identificate în cadrul acestei clasificări, se alege metoda de codificare. Metoda aleasă ar trebui să întrunească următoarele condiții :

— *Extindere* : Codul trebuie să lase spațiu pentru adăugiri de noi articole înăuntrul fiecărei clasificări. Trebuie să aibă de asemenea, capacitatea de a extinde clasificările existente și să adauge altele noi în vederea schimbărilor viitoare.

— *Precizie* : Structura codului trebuie să fie astfel încit numai un singur cod să se potrivească corect unui anume element și reciproc (corespondență biunivocă).

— *Concizunea* : Codul trebuie să aibă nevoie de un număr cît mai mic de cifre sau litere pentru o descriere a unui element.

— *Conveniență* : Codul trebuie să fie ușor de înțeles de fiecare dintre cei care îl folosesc și ușor de aplicat, fie că este vorba de codificare sau decodificare.

Clasificarea codurilor

Codurile se pot clasifica după diferite criterii, ca de exemplu :

- a. după lungime :
 - a.1. cu lungime fixă (ex. 101, 102, 203, ...)
 - a.2. cu lungime variabilă (ex. 101, 7543, 78, ...)
- b. după componență :
 - b.1. numerice (ex. 75243)
 - b.2. alfabetice (ex. FRZ, STRG, ...)
 - b.3. alfanumerice (ex. PRG 101, 784 ...)
- c. după semnificație :
 - c.1. nesemnificativ (ex. 75243)
 - c.2. semnificativ în sine, anumite poziții din cod corespunzând unor caracteristici fizice ale elementului codificat (ex. CU20 poate semnifica o bară de cupru de diametru 20 mm sau L203070 poate semnifica un laminat cu dimensiunile $20 \times 30 \times 70 \text{ cm}^3$) sau semnificative prin convenție, adică semnificația unei poziții din cod rezultând în urma unei definiții (de exemplu, dacă prima poziție din cod semnifică prin convenție, tipul de articol dintr-o fabrică și dacă definim prin 1 materialele, rezultă că articolul având codul 12473 corespunde unui material).
 - c.3. mnemonice, anumite poziții sau întregul cod reprezentând o prezentare a denumirii elementului codificat, de exemplu pentru „program control” codul PRGCNTRL, s.a.).

Combinarea diferitelor criterii conduce la o serie de tipuri de coduri, mai des întâlnite în practică.

Exemple de coduri

a. **Codul secvențial.** Este unul din cele mai simple și constă în corespondență dintre mulțimea elementelor de codificat și mulțimea numerelor naturale. Fiecare nou element de codificat î se asociază numărului natural următor disponibil. Pentru prelucrarea mecanizată-automatizată, codurile fiind preferabil să aibă lungime fixă, se adaugă fiecărui număr natural zerouri la stânga, pînă la lungimea stabilă pentru cod.

Metoda de codificare secvențială este recomandabilă în cazul listelor scurte de elemente pentru care decodificarea nu ridică probleme deosebite sau în cazul unor simple substituiri de titlu, ca de exemplu, la înregistrarea facturilor.

Avantajul codificării secvențiale constă în faptul că se pot codifica astfel un număr nelimitat de elemente. Nu este însă mai puțin adevarat că, în scopul păstrării unei lungimi fixe a codului, este preferabil să se estimeze de la început care este volumul informațiilor de codificat pentru o perioadă de timp viitoare.

Un alt avantaj este utilizarea maximă a capacitatii de codificare a unui cod numeric de o anumită lungime.

Codificarea secvențială apare de cele mai multe ori combinată cu alte metode de codificare.

b. **Codurile bloc** se construiesc prin separarea codului în componente, fiecare componentă codificând anumite caracteristici ale elementelor de codificat. Folosirea puterilor lui 10 ca margini de separare a blocurilor permite recunoașterea blocurilor prin interpretarea directă a codului. În cazul în care blocurile nu sunt legate de numerele 10, 100, 1000... separarea lor se poate urmări doar pe lista de decodificare :

0001 - 0075	Articole din grupa A
0076 - 0514	Articole din grupa B
sau	
0000 - 0099	Articole din grupa A (primele două cifre egale cu zero)
0100 - 0999	Articole din grupa B (doua prime cifre egale cu zero)

Blocurile trebuie dimensionate astfel încît să se păstreze spațiu suficient pentru codificarea ulterioară a elementelor din blocurile respective, care mai pot apărea. Folosirea blocurilor din varianta a doua (separări prin multiplii de 10) ușurează eventuale controluri care se pot introduce pe parcursul prelucrării.

c. **Codurile pe grupe** se bazează pe introducerea unei clasificări a elementelor, ca de exemplu, împărțirea codului în 3 grupe (major, intermediar, minor), grupul major indicând tipul de articol (piesă sau material), grupul intermediar indicând o subîmpărțire (grupa de material, tipul de piesă), iar grupul minor folosind la identificarea secvențială a elementelor care au primele două grupuri comune.

2.4. Controlul prelucrării și organizării datelor

În procesul de culegere, transmitere și prelucrare automată a informațiilor pot apărea erori care pot afecta grav rezultatele ce trebuie să se obțină. Erorile pot proveni din 3 surse importante :

a. Erori care apar în cadrul procesului de colectare a datelor, de constituire a documentelor primare sau de trecere a acestora pe suporti (de exemplu, cartela perforată).

b. Erori datorită unor manipulații greșite ale operatorului care contrariează funcționarea sistemului de prelucrare automată a datelor, sau datorită scrierii incorecte a unor instrucțiuni în programe de către programator.

c. Erori datorită echipamentului de prelucrare automată a datelor. Corespunzător acestor trei tipuri de erori avem trei categorii de controale ce se execută de-a lungul procesului de prelucrare :

a. **Control extern.** Erorile care există în documentele primare și care sunt introduse în calculator odată cu datele de intrare sint cele care crează cele mai grele probleme întregului proces. Pentru evitarea acestora se iau atât măsuri preventive, cât și de corectare. Printre măsurile preventive, se pot considera atribuirea de coduri semnificative elementelor care să permită o corectare ulterioară, relativ ușoară. Prezența, în general redondantă, a cuvintelor prescurtate formate din litere, permite detectarea ușoară a literelor eronate.

O altă regulă care are un caracter preventiv este aceea de a limita la minimum necesar informațiile care provin din exterior. În legătură cu aceasta, o importantă deosebită o prezintă evitarea intro-

ducerii în paralel a codurilor și a semnificațiilor lor în clar; transmiterea textelor în clar direct la imprimantă pe de o parte și prelucrarea în paralel în SPAD a codurilor poate duce la erori grave. De aceea, rezultatele în clar trebuie, în general, obținute din codurile cu care se află de pildă în corespondență intr-o tabelă constituită ad hoc.

O atenție deosebită trebuie să se dea păstrării informațiilor din înregistrări în timpul existenței lor pe diferiți suporti (benzi, discuri etc.) în afara sistemului de prelucrare automată a datelor. Modificările accidentale ale acestora, care sunt preluate de SPAD, conduc la erori care necesită multă vreme și eforturi pentru a putea fi înălțurate.

În ce privește metodele propriu-zise de control, atât al documentelor, cât și al operațiilor de perforare, ele se pot executa cu ajutorul calculatorului electronic prin diferite metode, de pildă folosind în paralel o serie de date totalizate parțial și respectiv individuale, ale căror totalizări după introducerea în calculator trebuie să coincidă sau intervale de valori prestabilite, între care trebuie să varieze datele primare. În ce privește perforarea cartelelor, aceasta se verifică cu ajutorul verificatoarelor, la care se simulează o a doua perforare după documentul primar și care sesizează necoincidentele care apar.

Se execută de asemenea controale asupra unor grupuri de coduri din cadrul înregistrărilor pentru a verifica compatibilitatea lor. De pildă, se poate verifica un grup de coduri din diferite zone ale unei înregistrări, considerat ca informație de identificare: Bară de Otel, Cu 20. În acest caz: între denumirea articoului și materialul din care este format (cupru) nefiind compatibilitate, înseamnă că în redactarea documentului primar sau la perforare s-a comis o eroare.

b. Controlul manipulării. Un program după ce a fost compilat, ca atare după ce au fost eliminate erorile sintactice pe care le-a comis programatorul în redactarea programului într-un limbaj de programare, urmează să fie testat cu diferite seturi de date pentru a se verifica dacă el realizează procesul logic de prelucrare dorit. Erorile logice în descrierea algoritmului care apar în testare, sunt înălțurate succesiv de programator. Testarea cu date reale este recomandabil să se facă după ce programul a fost testat cu diferite seturi de date create ad hoc de către programator, care au avantajul de a fi în volum mai mic și care prezintă cazurile extreme nefavorabile care apar cu o frecvență mai mică în practică, dar pentru care evident trebuie luate în program măsurile necesare. Se poate întimpla ca după un timp îndelungat (1—2 ani) de la darea în exploatare a unui

program să apară erori datorită faptului că pentru anumite situații pe care le pot prezenta datele reale, nu s-au prevăzut în program instrucțiunile corespunzătoare.

Programele testate pentru situațiile pe care le poate concepe programatorul devin la rîndul lor criterii prin care se verifică corectitudinea datelor, care de pildă, au fost încărcate în fișierele necesare prelucrării. Prin urmare, se poate observa acest dualism programdate, care din aproape în aproape asigură pînă în cele din urmă corectitudinea procesului de prelucrare.

Manipularea fișierelor de pe discuri sau benzi magnetice este o sursă importantă de erori pe care le pot comite atât programatorul, cât și operatorul din sala calculatorului. De pildă, dacă pe un disc sunt mai multe fișiere, un calcul de adresă greșit datorită instrucțiunilor greșite dintr-un program, poate conduce la modificarea unui fișier permanent important.

De aceea, programatorul trebuie să ia măsuri speciale de protecție, pentru a nu fi posibil transferul de informații într-un fișier străin. La rîndul său, operatorul care de la consola unității centrale poate comanda operații de stergere sau de copiere a unui fișier, trebuie să efectueze toate verificările prevăzute, înainte de astfel de manevre, pentru a nu distrugă un fișier permanent sau variabil. În același scop se asigură pentru fișierele permanente sau variabile, copii care sunt păstrate în arhivă și care sunt actualizate odată cu fișierul original. Importanța acestor copii în arhivă poate fi pusă în evidență de următoarea situație în care este pus un operator foarte frecvent: să se copieze un fișier de pe banda pe care este înregistrat, pe o bandă curată. Este suficient ca operatorul să inverseze bobinele pentru ca prin copierea benzii curate să se steargă înregistrările de pe banda pe care se află fișierul.

Organizarea evidenței automate pe SPAD a fișierelor pe benzi și discuri cu păstrarea istoriei acestor fișiere și utilizarea de instrucțiuni precise de manipulare, limitează, dacă nu reduc cu totul aceste erori.

Unul dintre controalele eficiente care se poate prevede în cadrul sistemului îl constituie metoda utilizării tranzacțiilor directe și inverse aplicate asupra bazei de date de rezervă, respectiv asupra bazei de date curente, iar numai în cazul coincidenței pentru o anumită dată intermedieră între data bazei de date de rezervă și data bazei de date curente, să se producă continuarea procesului. De pildă, să presupunem că baza de date de rezervă se actualizează lună. Se constituie un fișier al tranzacțiilor din luna respectivă, care se aplică

direct bazei de date de rezervă și un fișier de tranzacții care să se aplice invers bazei de date curente. Să considerăm că suntem în 20, luna curentă și că este nevoie de un raport care necesită situația bazei de date în ziua de 10 luna curentă. Coincidența prelucrării bazei de rezervă pînă în ziua de 10 cu ajutorul fișierului tranzacților directe (1—10 luna curentă) și a bazei de date curente cu ajutorul fișierului tranzacților inverse (20—10 luna curentă), constituie un criteriu puternic de validare, atît a datelor din fișierele din baza de date, cît și a prelucrărilor efectuate.

Luarea în condițiile specifice existente a tuturor măsurilor posibile de validare, atît la nivelul intrărilor și fișierelor, cît și la nivelul prelucrării tranzacților, constituie o necesitate vitală și o obligație a proiectanților de sisteme informative, orice compromis sau concesie putînd să fie scump plătit în timpul exploatarii sistemului.

c. **Controlul intern.** Erorile care apar datorită unei funcționări greșite a echipamentelor sunt mai rare decît cele externe sau de manipulare, datorită măsurilor de întreținere preventivă și a programelor de testare a elementelor din sistem, care permit înlocuirea din timp a acelora care în condiții anormale, de exemplu, de alimentare cu tensiune, ar putea produce erori în prelucrare.

Erorile cele mai frecvente sunt datorită echipamentelor de intrare și ieșire, care conțin numeroase dispozitive mecanice. Controalele ce se execută asupra operațiilor de intrare și ieșire de către sistemul de prelucrare sunt numeroase; de pildă, se compară natura cimpurilor (numerice sau alfabetice), se verifică numărul coloanelor libere din cartele, se execută două citiri și se verifică coincidența etc.

De asemenea, codurile perforaților (pe cartele sau bandă de hîrtie) posedă în cele mai multe cazuri proprietăți de autodetectare a unor erori mai frecvente. Imprimantele sau mașinile electrice de scris au asigurate de către constructori, dispozitive de control a funcționării.

În cadrul memoriori interne se efectuează controale de paritate cu ajutorul unor coduri care conțin o cifră de paritate, astfel de pildă, ca numărul total de unități în orice caracter sau cuvînt din memorie să fie par. Dacă controlul de paritate este utilizat și pentru informațiile de pe discuri sau benzi, acesta este necesar să se efectueze de fiecare dată cînd este utilizat suportul respectiv, deoarece erorile pot apărea și în timpul dintre două utilizări (impulsuri magnetice parazite, care pot apărea de pildă datorită unei opriri întîmpătoare a tensiunii de alimentare). Controlul informațiilor se efec-

tuează, de asemenea, și asupra informațiilor care se înscriu pe aceste suporturi. Detectarea unor astfel de erori nu trebuie în general să producă oprirea funcționării sistemului; de aceea în program trebuie adesea prevăzute proceduri de corectare, care dacă nu sunt standard, adică nu fac parte din sistemul de operare, trebuie elaborate în mod particular de către programator.

Ca o concluzie generală privind metodele de validare și control, se poate afirma că este necesar ca în sistemele informative folosind prelucrarea automată a datelor, să se prevadă absolut toate procedurile de validare și control posibile în etapa respectivă, chiar dacă prețul care trebuie plătit în creșterea timpului de prelucrare pare mare. Neconsiderarea strictă a acestor proceduri poate pune, în perspectivă, în discuție însăși existența sistemului informational instalat.

3. Reprezentarea prelucrărilor cu ajutorul schemelor logice

3.1. Scheme logice de programare

În proiectarea sistemelor informaționale și în programare, se folosesc simboluri pentru a reprezenta anumite activități; reprezentarea simbolică se face prin intermediul schemelor logice de programare și de sistem.

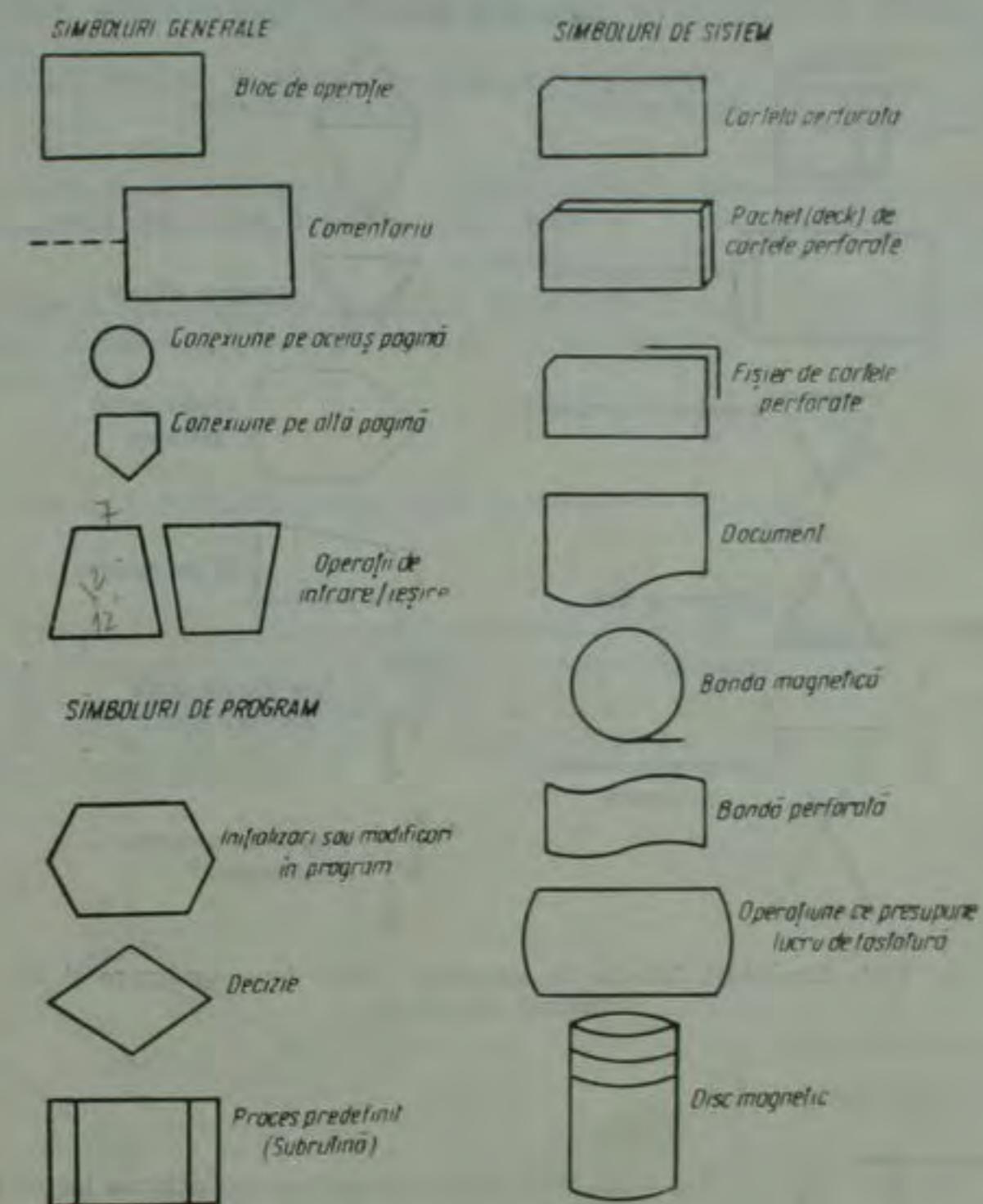
O schemă logică sau schemă de sistem este un graf orientat, care are în loc de noduri, simboluri (căsuțe) și care sunt unite prin săgeți. Simbolurile sunt reprezentate în figura 3.1.1.

Schimă logică de programare reprezintă un anumit proces de prelucrare a informației, care se efectuează în unitatea centrală, deoarece unitățile periferice nu prelucrează informația, ci numai o înmagazinează sau o pun la dispoziția unității centrale. La baza oricărei scheme logice stă algoritmul de prelucrare (descrierea exactă a procesului de prelucrare a datelor în SPAD). O schimă logică de programare are un început notat prin simbolul din figura 3.1.2. și un sfîrșit notat prin simbolul din figura 3.1.3.

Operațiile de prelucrare descrise de schimă logică se efectuează una după alta în mod secvențial. Este posibilă întreruperea secvenței normale de prelucrare atunci cînd apare un punct de decizie reprezentat prin simbolul din figura 3.1.4.

Decizia este reprezentată de o variabilă logică (cap. I §4). Dacă variabilă logică are valoarea ADEVĂRAT (TRUE) atunci ieșirea din blocul de decizie se face prin ramura DA. Dacă variabilă logică are valoarea FALSE (FALSE), atunci ieșirea se face prin ramura NU. Putem considera valoarea variabilei logice ca fiind generată de o expresie logică condițională.

Subliniem diferența între atribuirea de valoare logică unei variabile logice și între interpretarea valorii unei expresii logice condi-



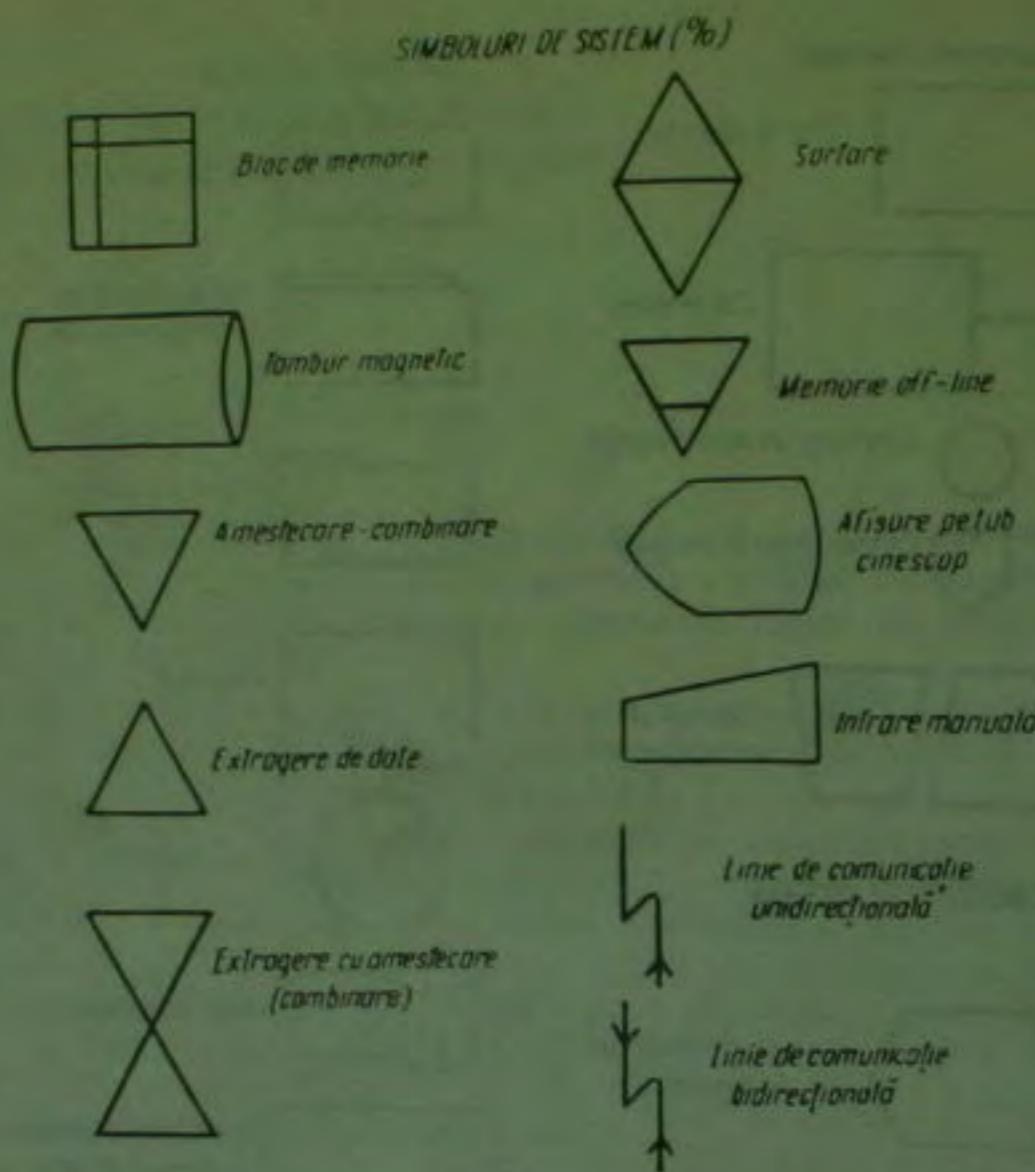


Fig. 3.1.1. Simboluri folosite în schemele logice de programare și în schemele de sistem.

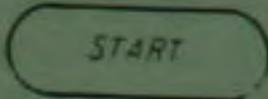
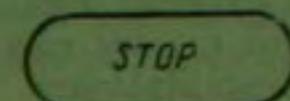


Fig. 3.1.2. Simbolul de inceput al unei scheme logice sau de sistem.

16. 11
32. 18
16. 9

tionale. În cazul atribuirii de valoare, în partea stângă a expresiei există un identificator care reprezintă o variabilă logică, căreia i se atribuie valoarea logică a expresiei care urmează după semnul de

Fig. 3.1.3. Simbolul de sfîrșit al unei scheme logice sau de sistem.



atribuire ($:=$). Atribuirea de valoare nefiind o interpretare a valorii, nu duce la nici o acțiune de decizie. Astfel :

$$L := A \vee B \wedge C \quad (1)$$

are drept efect atribuirea valorii logice TRUE sau FALSE variabilei logice L . Ea apare într-un bloc de prelucrare care se reprezintă ca în figura 3.1.6.

Fig. 3.1.4. Simbolul pentru punct de decizie.



Procesul de prelucrare își continuă secvența normală după ce a atribuit variabilei logice L valoarea corespunzătoare.

În cazul cînd urmează să se ia decizia într-un bloc de decizii, atunci în acesta apare numai expresia logică condițională. De exemplu ca în figura 3.1.7.

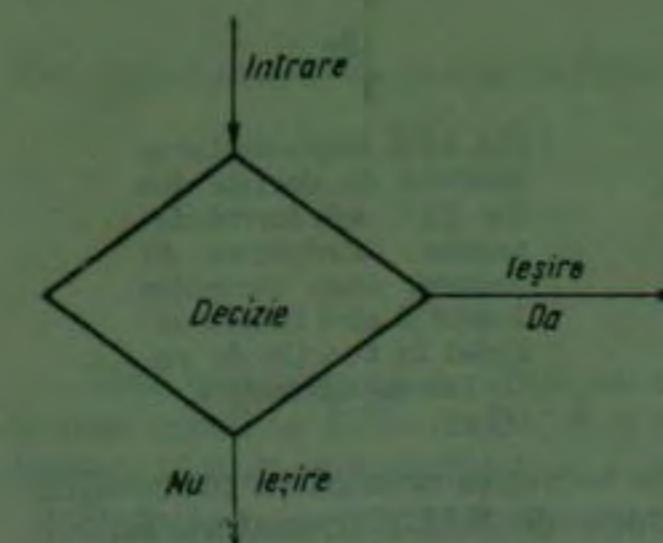


Fig. 3.1.5. Simbolul de decizie folosit în cazul prelucrării.

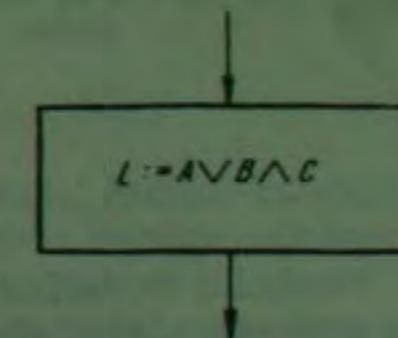


Fig. 3.1.6. Exemplu de apreciere a valorii unei variabile logice.

În acest caz nu se face o atribuire de valoarea unei variabile care a fost declarată [1], ci se ia o decizie.

Sistemul de prelucrare al datelor atribuie valoarea logică a acestei expresii logice condiționale, unei variabile despre care programatorul nu are nici o informație și apoi o interpretează luând decizia corespunzătoare.

Se putea reprezenta acest bloc de decizie și sub forma din figura 3.1.8.

În acest caz, programatorul poate controla valoarea variabilei logice L , rezultă din expresia logică (1).

Decizia este luată tot prin interpretarea unei expresii logice condiționate și anume expresia logică formată dintr-o singură variabilă L care apare în blocul de decizie.

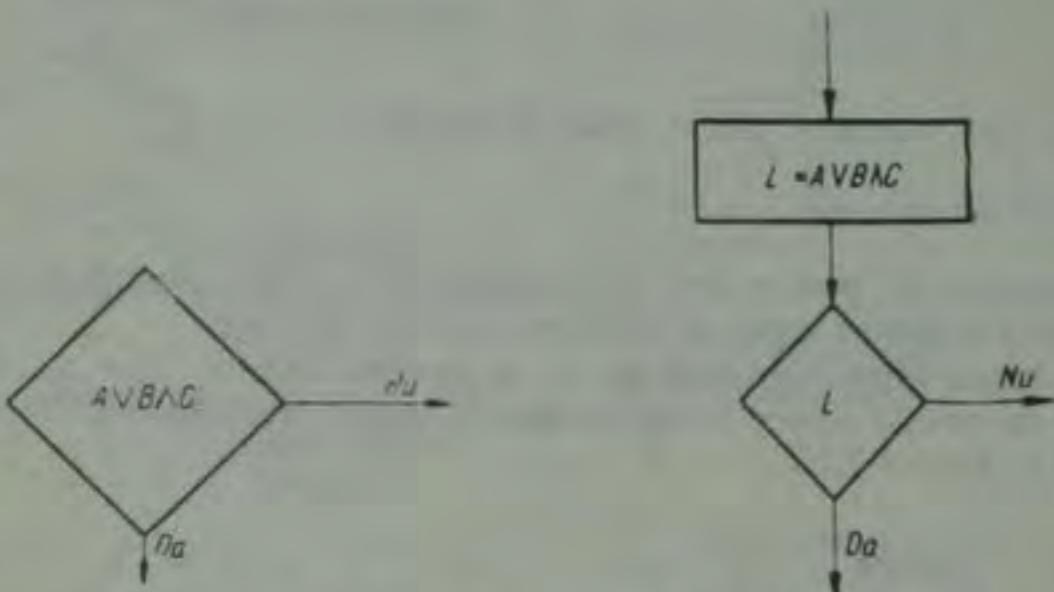


Fig. 3.1.7. Exemplu de decizie prin aprecierea unei expresii logice.

Fig. 3.1.8. Reprezentarea blocului de decizie din fig. 3.1.7, sub formă detaliată: atribuirea de valoare unei variabile logice și apoi luarea deciziei în funcție de valoarea variabilei.

Simbolul de decizie poate intrerupe secvența normală a procesului de prelucrare, efectuind aşa zisa operație de SALT (transfer). Saltul rezultat în urma unei decizii se numește salt CONDITIONAT.

În cazul unor limbaje (de exemplu FORTRAN sau ASSEMBLER) există un tip de salt condiționat care are trei ieșiri. În acest caz, expre-

sia care apare în blocul de decizie este o expresie aritmetică, căreia î se interpretează valoarea în raport cu valoarea zero. Dacă valoarea expresiei aritmetice este mai mică decit zero se continuă prin prima ramură, dacă valoarea este egală cu zero, se continuă prin a două ramură; dacă valoarea este mai mare ca zero se continuă prin cea de a treia ramură (figura 3.1.9.).

Variabilele și constantele scrise în expresia $257 - A \star B$, trebuie să fie toate aritmetice. Dacă A sau B sunt variabile logice, atunci expresia este incorrectă. În cazul acesta expresia din blocul de decizie nu poate avea decit operatori aritmetici și variabile aritmetice.

Blocul de decizie din fig. 3.1.9. poate fi descompus în blocuri de decizie cu două ieșiri, scriind expresii condiționale.

În acest caz expresia aritmetică $257 - A \star B$ se transformă într-o expresie condițională $257 - A \star B > \theta$. Dacă a fost îndeplinită această condiție, atunci ieșirea este dată de ramura $> \theta$ din figura 3.1.10. Dacă valoarea expresiei nu este mai mare ca zero, se testează condiția $257 - A \star B = \theta$, adică dacă se poate continua prin ramura corespunzătoare $= \theta$ din figura 3.1.10. În cazul că această condiție este satisfăcută se continuă prin ramura $= \theta$ din figura 3.1.10.

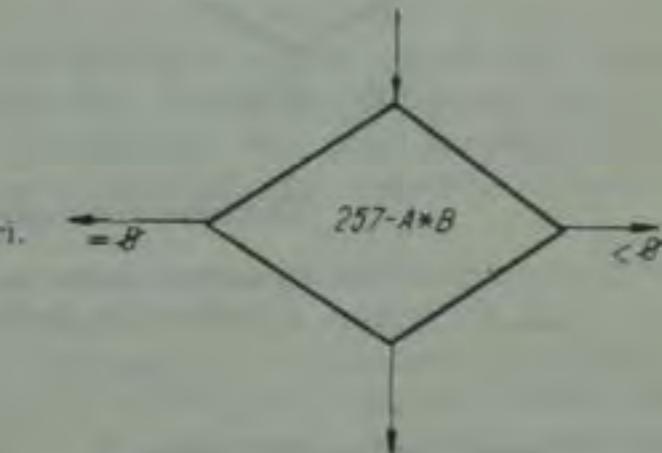


Fig. 3.1.9. Exemplu de decizie cu 3 ieșiri.

Dacă însă această condiție nu e satisfăcută, atunci rămîne valabilă numai condiția $257 - A \star B < \theta$ și se folosește ieșirea corespunzătoare pentru $< \theta$ din figura 3.1.10.

Trebuie reținută diferența dintre operatorul relațional = care apare în expresia $257 - A \star B = \theta$ și operatorul := de atribuire de valoare care apare atunci cînd unei variabile i se atribuie o valoare aritmetică sau logică.

În afara saltului condiționat care rezultă de fapt din interpretarea unor condiții, există și saltul necondiționat care, fără a interpreta nici o condiție, intrerupe secvența normală a desfășurării procesului de prelucrare.

Saltul necondiționat nu are un simbol separat. Se consideră salt necondiționat atunci cind ieșirea dintr-un bloc de prelucrare (ce nu este bloc de decizie) nu se face conform secvenței normale, ci se efectuează salt la un alt bloc care nu-l urmează în secvență.

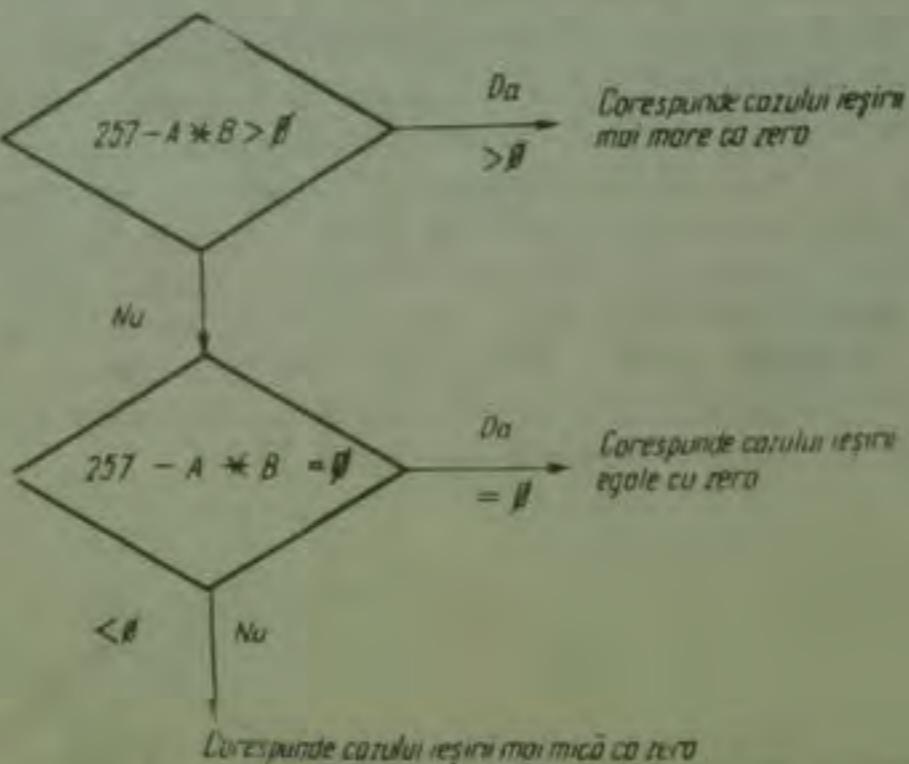


Fig. 3.1.10. Descompunerea blocului de decizie cu 3 ieșiri în bloc de decizie cu 2 ieșiri.

3.2. Scheme logice de sistem

Dacă schemele logice de programare reprezintă procesul de prelucrare în unitatea centrală, schemele de sistem reprezintă procesul de prelucrare pe configurația SPAD.

În schemele de sistem nu se reprezintă operațiile din cadrul prelucrării informației, nici tipurile de acces sau structura înregistrărilor din fișiere etc.; în ele se reprezintă suportul informației (bandă

magnetică, disc magnetic, cititor de cartele, imprimantă, mașină de scris (consolă), dispozitiv vizual de afișare etc.), felul prelucrărilor și modul de culegere a informațiilor (fig. 3.1.1.).

Unitatea centrală, în care are loc prelucrarea, se reprezintă printr-un dreptunghi (simbolul de prelucrare) în interiorul căruia se scriu caracteristicile prelucrării. În general un asemenea simbol reprezintă un program sau mai multe programe.

La simbolurile de prelucrare se leagă fișierele, indicind operațiile de intrare/ieșire. În general, se urmărește evoluția prelucrării informației începând de la informația primară dată pe documente sursă până la obținerea informației prelucrate sub formă dorită, reprezentându-se toate fișierele care participă la prelucrare.

Fișierul este reprezentat prin simbolul asociat suportului pe care se găsește informația și printr-un nume.

Informația primară (datele) se află la început pe un document primar (document sursă).

De pe documentul sursă, informația se trece într-un fișier, sau prin perforarea datelor de pe documentul sursă pe cartelă perforată, sau prin citirea directă a documentului sursă într-un fișier aflat pe un suport oarecare, în cazul cind documentul primar e scris cu o cerneală specială magnetică sau tipărit cu caractere speciale, care sunt citite de un cititor optic.

Informația primară se poate găsi și direct pe un suport de informație accesibil sistemului de prelucrare, respectiv pe bandă de hirntie perforată, pe cartele, pe bandă magnetică etc., dacă această informație se colectează automat sau semiautomat prin intermediul unor dispozitive de colectare a datelor.

Prin urmare, prima fază a procesului de prelucrare a informației, constă în înregistrarea informației primare pe un suport de informație accesibil SPAD (depinzind de dotarea cu echipamente periferice a SPAD) sau în colectarea directă prin intermediul unor echipamente de colectare a datelor pe un suport de informație accesibil SPAD.

De pe acest suport informația intră în prelucrare. La procesul de prelucrare participă în afara informației primare și informații deja prelucrate.

Datele reprezentate pe un suport de informație accesibil SPAD și care participă la actualizarea fișierelor, se numesc după cum am văzut tranzacții.

Orice document sursă poate conține erori, care apar fie la scrierea documentului, fie în procesul de colectare a datelor.

Unele erori (cap. II § 4), pot fi detectate de către SPAD, ținind seama că sunt anumite reguli generale care trebuie respectate pentru anumite categorii de informații. De exemplu, dacă o tranzacție cuprinde o dată calendaristică sub forma -zi, lună și an- se pot detecta următoarele tipuri de erori :

- dacă ziua este mai mică sau egală cu zero sau mai mare decit numărul maxim de zile din luna respectivă, ținind seama pentru 02 (februarie), dacă anul este bisect sau nu,

- dacă luna este mai mică sau egală cu zero sau mai mare decit 12,

- dacă anul este mai mic decit anul curent minus o valoare sau dacă anul este mai mare de cit anul curent, plus o altă valoare.

Acste tipuri de erori de pe documentele sursă sau din fișierele care au fost deja prelucrate, care pot fi detectate în procesul de prelucrare, se numesc erori sintactice.

Este foarte important ca în schema de prelucrare să fie prevăzute aceste verificări de erori sintactice, pentru a evita obținerea de informații alterate.

În general, verificarea sintactică se face prin verificarea tipului informației, care trebuie să fie sau numerică (nu are voie să conțină litere) sau alfanumerică (poate conține orice caracter), sau prin valoările sau limitele valorilor pe care informația trebuie să le aibă.

De multe ori informația deja prelucrată, poate participa la procesul de prelucrare ca informație primară.

După ce informația este prelucrată, poate fi înregistrată pe același suport de informație, fiind reprezentată printr-o săgeată sau prin două săgeți care leagă blocul de prelucrare cu suportul, o săgeată indicând intrarea (virful spre blocul de prelucrare) și o săgeată reprezentând ieșirea (cu virful spre suport) sau printr-o singură linie având cele 2 săgeți, una cu virful spre blocul de prelucrare și una cu virful spre suport.

3.3. Exemple privind reprezentarea prelucrărilor

Fie o mulțime de numere întregi, înregistrate pe un anumit suport de informație care poate fi : cartelă, bandă de hârtie perforată, bandă magnetică, disc magnetic etc.

Setul de date se termină prin caractere speciale /★/. Se cere să se facă aranjarea acestor numere în ordine crescătoare, iar rezultatul să fie înregistrat pe un alt suport de informație.

Rezolvarea problemei se va face în următorii pași :

- Procedeul de găsire al maximului sau minimului dintr-un set de date.
- Procedeul de ordonare.

Găsirea maximului și minimului

Vom studia această problemă în două variante. Prima variantă este aceea în care datele nu vor fi reținute în memorie. În a doua variantă se rețin în memorie.

În ambele variante se va face tipărirea maximului și minimului.

a. Schema de sistem. Pentru rezolvarea acestei probleme este necesar să figurăm mediul de intrare, de exemplu banda magnetică, blocul de prelucrare și imprimanta la care se serie rezultatul. Suportul pe care se află datele îl vom numi MI, iar rezultatul obținut îl vom numi TIPAR (fig. 3.3.1.).

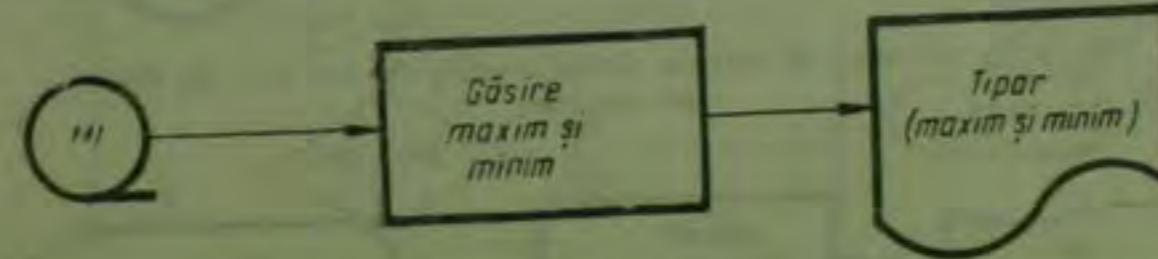


Fig. 3.3.1. Exemplu de schemă de sistem cu suport de informație bandă magnetică și imprimată.

O altă variantă ar fi aceea în care mediul de intrare este un disc magnetic (fig. 3.3.2.).

În același mod, mediul de intrare poate fi considerat un fișier pe cartele. Schema de sistem arată astfel : (fig. 3.3.3.).

Dacă mediul de intrare se află pe bandă perforată, iar mediul de ieșire este un dispozitiv de afișare, schema de sistem este fig. 3.3.4.

b. Schema logică. Pentru determinarea maximului și minimului, sau numai a maximului ori numai a minimului, se procedează astfel :

- Se definește în memorie o zonă de lucru pe care o numim ARIA, în care se citește elementul de pe suportul de informație și

trei variabile întregi MAX, MIN, C, care reprezintă respectiv : valoarea maximă, valoarea minimă și numărul curent de elemente care au fost citite de pe suport (contor).

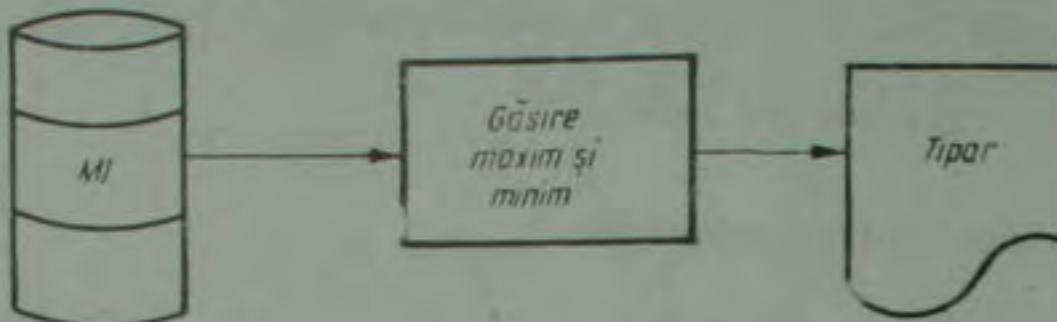


Fig. 3.3.2. Exemplu de schemă de sistem folosind drept suport de informație discul magnetic și imprimantă.

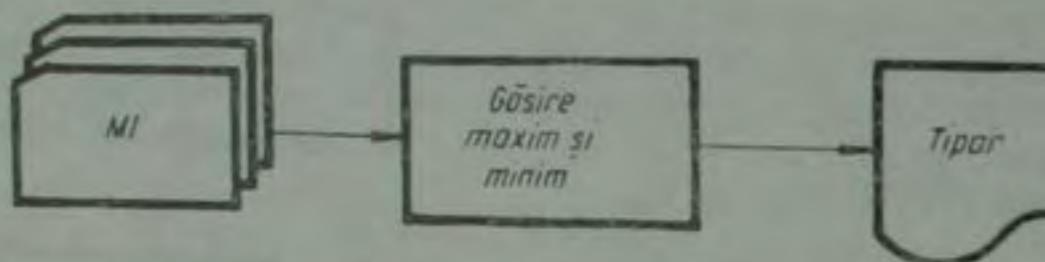


Fig. 3.3.3. Schemă de sistem având mediu de intrare un fișier pe cartele.

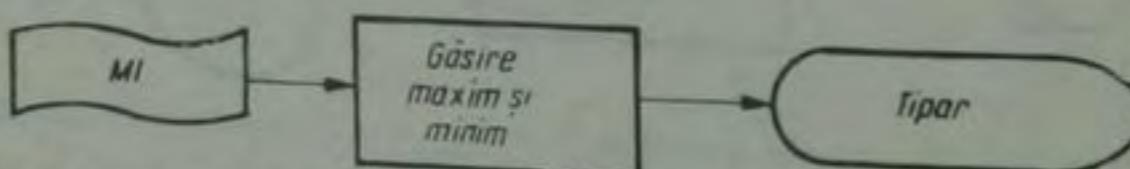


Fig. 3.3.4. Schemă de sistem având drept suport de informație bandă de hârtie perforată și dispozitiv de afișare.

— Se initializează contorul C cu valoarea 1, iar MIN și MAX cu valoarea primului element citit de pe suport.

Algoritmul de aflare a maximului și minimului poate fi urmărit pas cu pas după schema logică din figura 3.3.5.

1. Start.
2. Initializeare contor $C := 1$.
3. Citirea unui element de pe mediul de intrare în zona numită ARIA.

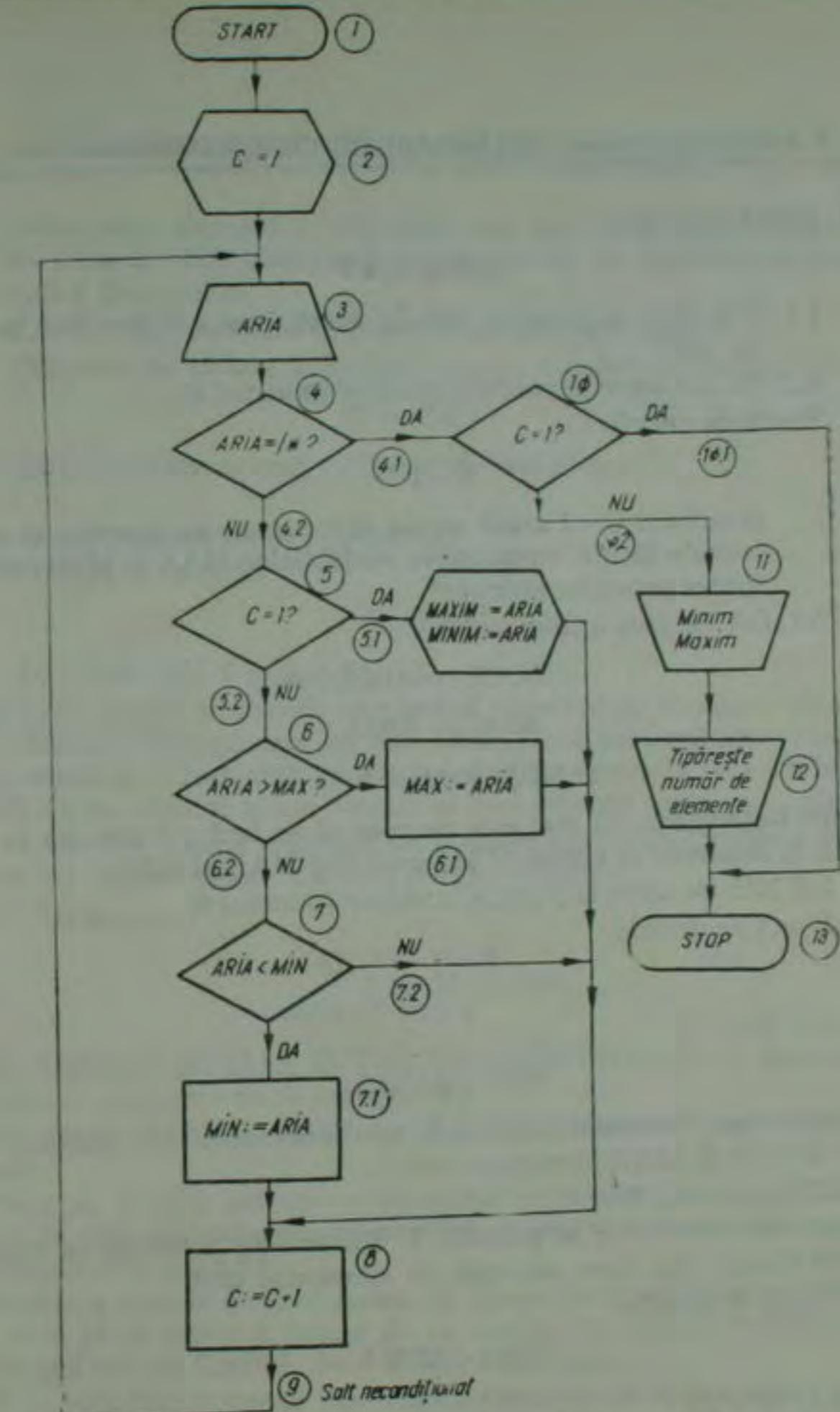


Fig. 3.3.5. Schemă logică reprezentând un algoritm de determinare n maximului și minimului.

4. Punct de decizie :

$ARIA = /★ ?$

- 4.1. DA. Salt la punctul 10. În acest caz s-a terminat setul de date.
 4.2. NU. Se trece la punctul următor (punctul 5).

5. Punct de decizie :

$C = 1 ?$

(Contorul $C = 1$ arată că s-a citit numai un element și deci trebuie făcută inițializarea variabilelor MAX și MIN cu valoarea primului element).

- 5.1. DA. Se face inițializarea

$MAX := ARIA$

$MIN := ARIA$

SALT la punctul 8.

După inițializare, nu mai este necesar să se facă comparația de la punctul 6, deoarece se cunoaște valoarea din MAX și MIN.

- 5.2. NU. Se trece la punctul următor (punctul 6).

6. Punct de decizie.

$ARIA > MAX ?$

- 6.1. DA.

$MIN := ARIA$

In acest caz, elementul citit este mai mare decit cel existent in MAX. Se scrie in MAX elementul citit.

SALT la punctul 8.

6.2. NU. Se trece la punctul 7. Nu se face schimbarea deoarece MAX este mai mare sau egal cu elementul citit.

7. Punct de decizie.

$ARIA < MIN ?$

- 7.1. DA.

$MIN := ARIA$

3.3. EXEMPLE PRIVIND REPREZENTAREA PRELUCRARILOR

In acest caz, elementul citit este mai mic decit cel existent in MIN. Se scrie in MIN elementul citit in ARIA. Se continuă cu punctul următor (punctul 8).

- 7.2. NU. Se continuă la punctul următor (punctul 8).

8. Operația de ciclare a contorului care numără elementele citite.

$C := C + 1$

După executarea acestei ciclări procedeul se reia.

Deci :

9. SALT necondiționat la punctul 3.

10. Punct de decizie.

$C = 1 ?$

- 10.1. DA. SALT la punctul 13.

La acest punct de decizie se verifică numărul de elemente diferite de $/★$ (numere întregi) care au fost citite. Dacă contorul are valoarea 1, înseamnă că a fost citit un singur element și acesta este $/★$, MAX și MIN nu au căpătat nici o valoare și deci valorile lor nu se tipăresc.

10.2. NU. Există cel puțin un element diferit de $/★$ și deci MAX și MIN au căpătat valori. Execută punctul următor.

11. Tipărește.

$MAXIM = MAX$

$MINIM = MIN$

12. Tipărește NUMĂR TOTAL DE ELEMENTE = $C - 1$. În numărul total de elemente nu se include $/★$.

13. STOP. MAX și MIN au fost determinate și procesul se oprește.

Dacă se dorește scrierea maximului și minimului și pe un alt suport de date (de exemplu pe cartele), atunci schema de sistem este reprezentată în fig. 3.3.6.

Pentru a înscrie (reține) datele in memorie in vederea ordonării, vom cere să se citească datele de pe mediul de intrare in memorie într-o arie numită TABEL. Se definesc variabilele :

N — variabilă întreagă care indică numărul de elemente citite de pe mediul de intrare, indicii 1 : N fiind în corespondență cu aceste elemente.

TABEL ($1 : N$) este un tabel (vector) cu N elemente de pe mediu de intrare, indicii $1 : N$ fiind în corespondență cu aceste elemente.
 IND — variabilă întreagă;

J — variabilă întreagă.

Algoritmul este următorul :

Se înscriu datele în memorie în aria TABEL ($1 : N$).

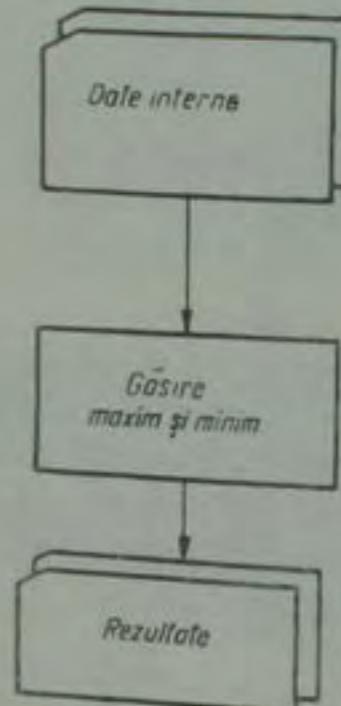


Fig. 3.3.6. Schema de sistem pentru scrierea maximului și minimului pe cartele perforate.

Se caută în loc de minim, indicele elementului de valoarea minimă
 Se initializează $IND=1$ și $J=2$.

Se compară dacă $TABEL(J) < TABEL(IND)$. Dacă da, atunci se scrie $IND=J$. Altfel IND rămâne neschimbat.

Se continuă astfel pentru $J=3$ cînd iarăși se compară dacă $TABEL(J) < TABEL(IND)$. Dacă da, se pune $IND=J$. Se continuă pînă cînd $J=N$ cînd se tipărește minimul.

La fel putem proceda și pentru maxim.

Dacă se dorește să se obțină și minim și maxim în același timp, se folosesc două variabile: $IND1$ pentru maxim și $IND2$ pentru minim.

Descrierea algoritmului se face conform schemei logice din figura 3.3.7. (pentru aflarea minimului).

1. START.

2. INITIALIZARE.

$$N := 0$$

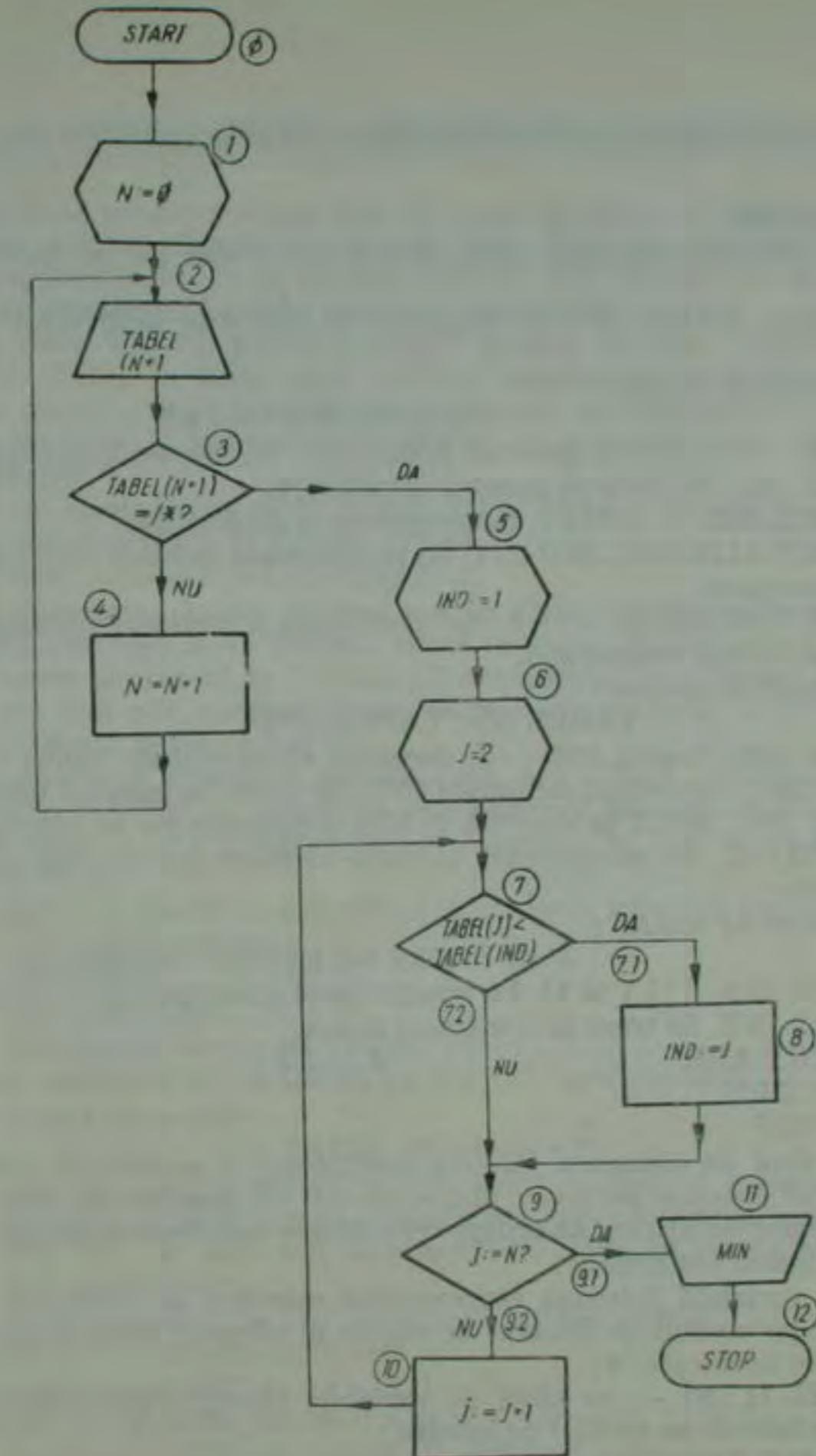


Fig. 3.3.7. Schema logică reprezentînd un algoritm de determinare a minimului prin compararea indicilor.

3. CITIRE.

TABEL ($N+1$) := valoarea elementului $N+1$ -lea de pe mediu de intrare.

S-a scris TABEL ($N+1$) deoarece TABEL este numerotat de la $1 : N$.

3. Punct de decizie :

TABEL ($N+1$) = /★ ?

3.1. DA. SALT la punctul 5, deoarece citirea s-a terminat.

3.2. NU. Se trece la punctul următor (4).

4. CICLARE. $N := N+1$. Se pregătește o nouă citire.

5. INITIALIZARE. $IND := 1$. Se poziționează indicele elementului care se compară.

6. INITIALIZARE. $J := 2$. Se poziționează indicele elementului cu care se incepe comparația.

7. Punct de decizie :

TABEL (J) < TABEL (IND)

7.1. DA. Atunci $IND := J$, deoarece elementul de indice J este mai mic decât elementul de indice IND . Se trece la punctul următor.

7.2. NU. SALT la punctul 9. Nici o operație nu se efectuează.

8. $IND := J$. Se poziționează indicele elementului cel mai mic de pînă acum.

9. Punct de decizie :

$J = N$? Adică s-a terminat comparația.

9.1. DA. SALT la 11. Se înregistrează și se tipăresc rezultatele.

9.2. NU. Se trece la următorul punct.

10. CICLARE.

$J := J + 1$.

11. IEȘIRE.

12. STOP.

Procedeu de ordonare. Găsirea maximului și minimului descrise mai sus se folosesc pentru a obține modul de aranjare a obiectelor de pe mediul de intrare în ordine crescătoare sau descrescătoare.

Se folosesc variabilele :

N — variabilă întreagă reprezentind numărul de elemente care se găsesc pe mediul de intrare. Se obține la sfîrșitul citirii elementelor cînd se întîlnește /★;

TABEL ($1 : N$) — un tabel de variabile întregi reprezentind elementele citite de pe mediul de intrare;

$IND, I, J, SALV$ — variabile întregi.

Algoritmul se prezintă astfel :

Se inițializează $I := 1$.

Se caută minimul dintre cele N elemente aflate în TABEL ($1 : N$). Indicele acestui element se scrie în IND. (Deci la determinarea minimului folosim în loc de valoare, indicele elementului). Se determină indicele elementului minim ca în cazul precedent.

Se trece TABEL (IND) în SALV și apoi se trece TABEL (I) în TABEL (IND). În acest mod, vechiul conținut al lui TABEL (IND) a fost distrus, fiind înlocuit cu conținutul lui TABEL (I). Dar vechiul conținut a fost salvat în SALV și acum se trece SALV în TABEL (I).

Avem certitudinea că în TABEL (I) se află cel mai mic număr din cele N numere aflate în TABEL ($1 : N$).

Se face ciclarea $I := I + 1$. Deci $I = 2$.

Se determină la fel ca mai sus în IND indicele celui mai mic element din cele $N - 1$ rămase, adică începînd cu indicele 2 pînă la N . Aceasta pentru că în TABEL (I) avem sigur un element mai mic decît cel mai mic element dintre elementele de indice 2 pînă la N . Se procedează la fel ca mai sus.

Se continuă procedeul pentru $I = 3, 4, \dots, N - 1$. Pentru $I = N - 1$ procedeul ia sfîrșit și ordonarea este obținută în aria TABEL ($1 : N$), fără a folosi o arie suplimentară în memorie, în care să se scrie tabelul ordonat.

Descrierea exactă a algoritmului conform schemei logice din figura 3.3.8 este următoarea :

1. START.

2. INITIALIZARE. $N := 0$.

3. INTRARE. Se citește TABEL ($N+1$).

Am specificat la declarații că TABEL este declarat de la $1 : N$.

4. Punct de decizie :

TABEL ($N+1$) = /★ ?

4.1. DA. S-a terminat de citit setul de date și se face salt la punctul 5 unde incepe operația de ordonare propriu-zisă.

4.2. NU. Se face salt la punctul 16 unde se efectuează ciclarea variabilei N .

5. În acest punct a fost terminată citirea elementelor și incep operația de ordonare propriu-zisă.

INITIALIZARE. $I := 1 ; IND := 1$;

se incepe cu căutarea minimului pentru intregul TABEL ($1 : N$).

6. INITIALIZARE. $J := I + 1$. Variabila J este aici o variabilă de ciclare a indicilor pentru care se face căutarea minimului față de indicele IND care a fost inițializat cu I .

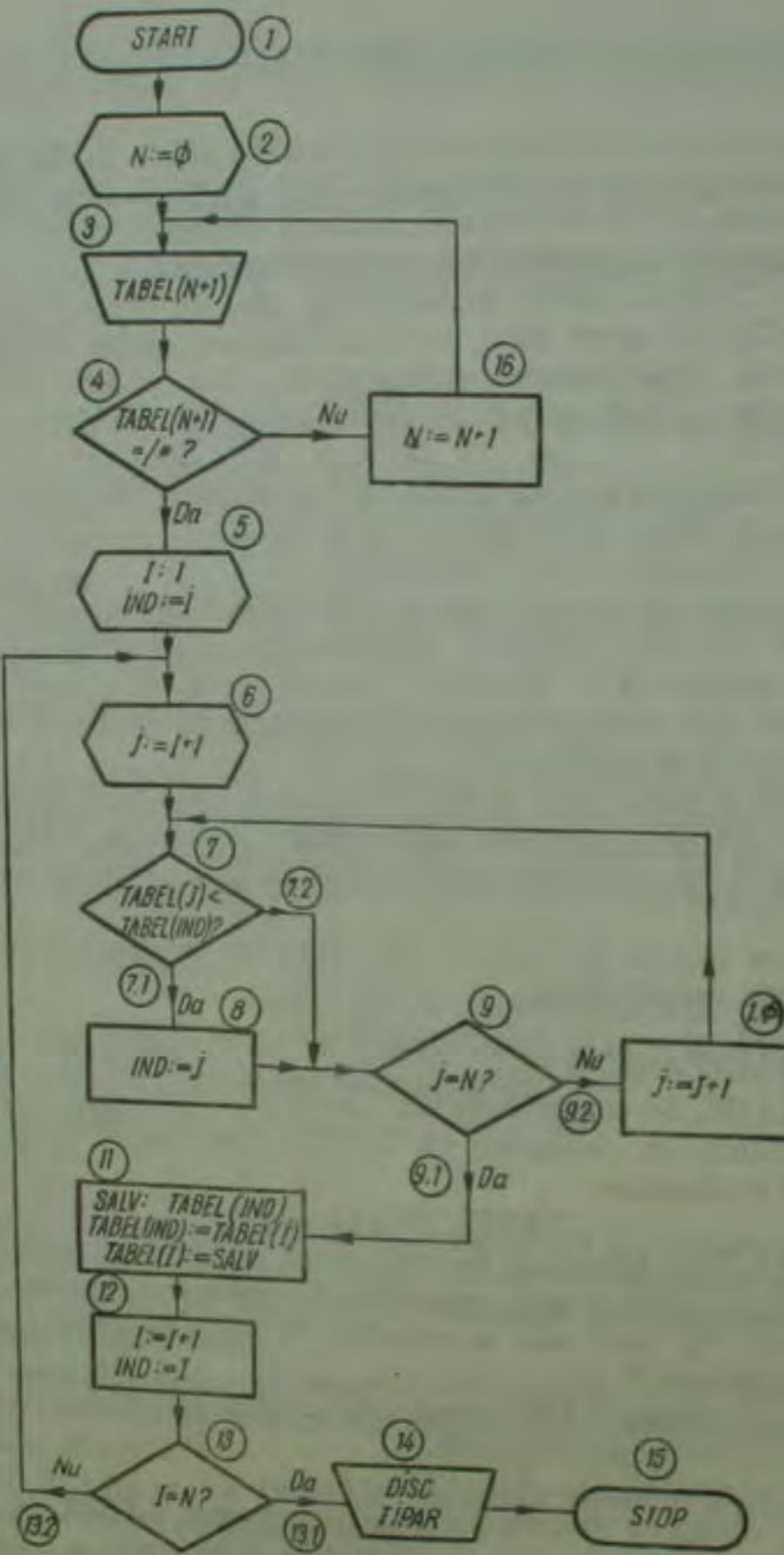


Fig. 3.3.8. Schema logică de ordonare a numerelor.

7. Punct de decizie.

TABEL (J) < TABEL (IND) ?

7.1. DA. În acest caz indicele J reprezintă un element mai mic decit cel mai mic element găsit pînă acum.

Se trece la punctul următor (punctul 8).

7.2. NU. Nu se face schimbarea de indice, deoarece elementul de indice IND este cel mai mic element din toate cele J care au fost testate pînă în prezent.

Se dă SALT la punctul 9.

8. IND := J. Se face schimbarea lui IND cu J, deoarece a fost găsit un element mai mic decit cel de indice IND.

9. Punct de decizie.

 $J = N ?$

Se testează dacă s-a terminat căutarea minimului.

9.1. DA. SALT la 11.

9.2. NU. SALT la 1Φ unde se face ciclarea indicelui J și se continuă căutarea cu elementul următor din TABEL (1 : N).

10. CICLAREA. $J := J + 1$. Pe această ramură se continuă căutarea, deoarece comparația nu a fost făcută pînă la ultimul element.11. Pe această ramură s-a găsit sfîrșitul căutării minimului ($J = N$).

Se face :

SALV := TABEL (IND) ;

TABEL (IND) := TABEL (I) ;

TABEL (I) := SALV.

Se scrie minimul în TABEL (I), iar în TABEL (IND) se scrie elementul din poziția I.

12. CICLARE. $I := I + 1$. Se pregătește căutarea minimului pentru restul de $N - I + 1$ elemente din TABEL (1 : N), începînd cu indicele I. IND := I.

13. Punct de decizie :

 $I = N ?$ (Adică s-a terminat ordonarea?)

13.1. DA. Se dă ieșire din ordonare. SALT la punctul 14.

13.2. NU. Se continuă ordonarea pentru următorul I obținut din ciclarea de la punctul 12. ($I < N$). SALT la punctul 6. Se continuă ordonarea pentru elemente începînd cu TABEL (I).

14. Se scrie pe disc TABEL (1 : N), copiindu-se toate cele N elemente. Se tipărește TABEL (1 : N) la imprimantă.

15. STOP.

16. CICLARE. $N := N + 1$. Se continuă cu punctul 3.

Se ciclează variabila N care numără elementele citite de pe mediul de intrare.

Considerind mediul de intrare un disc și mediul de ieșire de asemenea un disc, schema de sistem este :

MI este mediul de intrare

ME — mediul de ieșire

TIPAR — lista cu rezultate (tabelul ordonat).

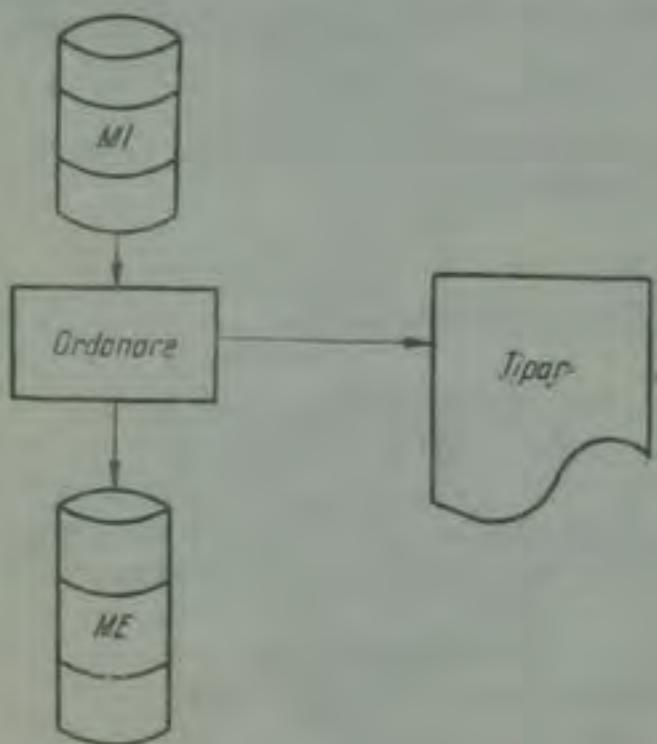


Fig. 3.3.9. Schemă de sistem cu mediu de intrare și de ieșire cu disc.

3.4. Operațiile principale folosite în schemele logice

Următoarele operații sint de o importanță deosebită în schemele logice de programare :

- operația de inițializare ;
- operația de ciclare ;
- operația de salt.

a. **Operația de inițializare.** Operația de inițializare are rolul de a atribui variabilelor valori initiale cu care se incepe calculul.

În exemplele precedente s-a arătat de exemplu, importanța inițializării variabilelor MIN și MAX. Aceste variabile trebuie să aibă o valoare dată la inceputul calculului, deoarece față de valoarea lor inițială se iau în continuare decizii.

3.4. OPERAȚIILE PRINCIPALE FOLOSITE

MAX și MIN au fost inițializate cu valoarea primului număr din setul de date. Dacă ar fi fost inițializate cu alte valori arbitrară, atunci nu ar fi existat certitudinea că în MAX sau în MIN ar fi apărut valoarea celui mai mare, respectiv a celui mai mic număr din multimea numerelor aflate pe suportul de date respectiv.

S-ar fi putut întâmpla ca în MAX să se scrie o valoare mai mică decit orice număr aflat pe suport.

Uneori inițializarea se mai face și altfel.

Orice calculator admite o valoare maximă a numerelor întregi care pot fi reprezentate în memorie, depinzînd de lungimea cuvîntului calculatorului respectiv.

De exemplu, un număr întreg poate fi reprezentat prin 32 de poziții binare. În această structură primul bit reprezintă de obicei semnul numărului. Dacă primul bit are valoarea zero de exemplu, atunci acesta poate fi considerat ca semn plus pentru număr. Dacă are valoarea 1(unu), atunci acesta poate fi considerat ca semn minus pentru număr. În acest caz, cel mai mare număr întreg care poate fi reprezentat în calculator este $N_{max} = 2^{31} - 1$, iar cel mai mic număr întreg care poate fi reprezentat este $N_{min} = -2^{31}$.

Presupunînd că N_{max} este cel mai mare număr întreg reprezentabil în calculator, iar N_{min} este cel mai mic număr întreg reprezentabil în calculator, atunci se putea face inițializarea $MAX = N_{min}$ și $MIN = N_{max}$.

În cazul cînd se face această inițializare, comparația dacă $ARIA > MAX$ sau dacă $ARIA < MIN$ se face de la citirea primului număr în ARIA.

Este evident că orice număr de pe suport nu poate fi decit mai mare sau cel mult egal cu N_{min} sau mai mic sau cel mult egal cu N_{max} , deci inițializarea variabilei MAX cu N_{min} și a variabilei MIN cu N_{max} are drept scop schimbarea valorii acestora chiar de la prima comparație.

Dacă toate numerele întregi de pe suport au valoarea N_{max} , atunci MAX va lua valoarea N_{max} și MIN va rămîne cu aceeași valoare ca la inițializare, deoarece altă valoare mai mică nu există. Dacă toate numerele de pe suport au valoarea N_{min} , atunci MIN va căpăta valoarea N_{min} (deoarece a fost inițializat cu N_{max}), iar MAX (care a fos inițializat cu N_{min}) va rămîne cu valoarea N_{min} , deoarece alt număr cu valoare mai mare nu există pe suport.

In concluzie, in cazul analizat MAX poate fi initializat cu cel mai mic număr intreg reprezentabil pentru calculatorul respectiv, iar MIN cu cel mai mare număr intreg reprezentabil pentru acel calculator.

S-au mai făcut initializări pentru variabilele I și J . De exemplu, în figura 3.3.8. se initializează variabila I cu valoarea 1 (unu). Se putea initialize această variabilă și cu valoarea 5 de exemplu.

Dacă valorile scrise în blocurile de prelucrare ale schemei logice din figura 3.3.8. ar fi rămas aceleași, ar fi însemnat că se caută aranjarea în ordine crescătoare a elementelor din TABEL începînd de la al cincilea element pînă la elementul N .

Dacă însă s-ar fi dorit initializarea de la primul element, atunci ar fi fost necesar să se scrie în blocurile de prelucrare următoarele valori :

- In blocul 5 :* $I := 5$
IND := $I - 4$ (adică IND = 1, primul indice din TABEL atunci cînd se începe prelucrarea)
- In blocul 6 :* $J := I - 3$ (adică $J = 2$, al doilea indice din TABEL, atunci cînd se începe prelucrarea).
- In blocul 7 :* Se face comparație TABEL ($J <$ TABEL (IND)). Aici nu se operează nici o modificare.
- In blocul 8 :* IND := J de asemenea nu se operează nici o modificare.
- In blocul 9 :* $J = N$ de asemenea nu se operează nici o modificare.
- In blocul 10 :* $J := J + 1$ de asemenea nu se operează nici o modificare.
- In blocul 11 :* SALV := TABEL (IND)
TABEL (IND) := TABEL ($I - 4$) (deoarece $I = 5$ și TABEL are numerotate liniile începînd de la 1, deci de la $I - 4 = 1$ la prima prelucrare).
TABEL ($I - 4$) := SALV.
- In blocul 12 :* $I := I + 1$ fără modificare.
- In blocul 13 :* $I := N + 4$ în loc de $I := N$
Se putea astfel atribui orice valoare cunoscută variabilei I .
- In schema logică din figura 3.3.5. s-a initializat variabila C cu valoarea 1, iar în schema logică din figura 3.3.8. s-a initializat variabila N (care joacă rolul variabilei C) cu valoarea \emptyset .*

In final, dacă N_C este numărul de elemente citite de pe suportul de informație (inclusiv caracterele $/★$), variabila N va avea valoarea $N_C - 1$, iar variabila C va avea valoarea N_C .

Interpretarea valorii acestor variabile se face diferit. În schema logică 3.3.5., cînd se tipărește numărul de elemente de pe suportul de informație (exclusiv caracterele $/★$), se tipărește valoarea $C - 1$.

În schema logică din figura 3.3.8. se obține valoarea $N_C - 1$ care reprezintă exact numărul de numere întregi afisate pe suport, deoarece această variabilă a fost initializată cu valoarea \emptyset .

b. Operația de ciclare. Operația de ciclare are scopul de a repeta un procedeu de calcul de un anumit număr de ori.

Pentru controlul numărului de cicluri se folosește o variabilă care se numește variabilă de ciclare (de control). Această variabilă începe de la o valoare initială atribuită prin program și la fiecare repetare a ciclului i se mărește sau micșorează valoarea cu o anumită cantitate care se numește pas de ciclare sau increment.

Incrementul poate fi o constantă sau o variabilă.

În exemplele arătate s-au folosit drept variabile de ciclare I și J . În acest caz, incrementul este o constantă cu valoarea 1. Drept increment se poate folosi o variabilă, de exemplu M , căreia i se poate atribui prin initializare valoarea $M = 1$.

Astfel, în blocul 6 din fig. 3.3.8. se putea scrie în loc de $J := I + 1$, expresia $J := I + M$, ce ar fi avut același efect dacă variabilei M i se atribuie initial valoarea 1.

Variabilele de ciclare avînd și rolul de variabile de control al ciclului, au totdeauna o valoare initială de la care se pornește controlul și o valoare finală la care controlul se termină. Aceste valori sunt stabilite de către programator.

Variabilelor de ciclare avînd funcția de numărare a ciclurilor, este necesar să li se atribuie corect valoarea de sfîrșit față de valoarea initială (de început).

În general, controlul valorii variabilei se face printr-o decizie. Decizia poate fi luată fie înainte de ciclare (adică înainte de adunarea incrementului) sau după ciclare (adică după adunarea incrementului).

De asemenea, controlul poate fi făcut în oricare din situațiile de mai sus, fie înainte de executarea ciclului (blocului de prelucrare pe care îl execută), fie după executarea ciclului.

În schema logică din figura 3.3.8. controlul variabilelor de ciclare I și J se face după executarea ciclului (a blocurilor de atribuire a valorii minimului la pasul respectiv).

De exemplu, în blocul 9 se controlează dacă $J=N$ și apoi se face ciclarea în blocul 10 cind se mărește valoarea variabilei J cu o unitate.

În această situație, dacă J are valoarea N (numărul ultimului element din TABEL), ciclul pentru J se consideră terminat. Este evident că în acest caz, TABEL (J) a fost ultimul element din TABEL, dacă J a avut valoarea N .

Se putea proceda și invers. Mai întii se putea cicla variabila J , adică să se scrie $J := J + 1$ și apoi să se facă controlul acestei variabile. În această situație, controlul ar fi trebuit făcut prin evaluarea expresiilor condiționale $J > N$ sau $J = N + 1$, adică ultima valoare a variabilei de control trebuie să fie $N + 1$, deoarece prin ciclarea făcută înainte de control, valoarea variabilei a fost crescută cu o unitate.

Se poate observa foarte bine acest lucru din blocurile 12 și 13. Ultima valoare a variabilei I , pentru care ciclul trebuie executat este $N - 1$. Deoarece variabila a fost mai întii ciclată și apoi controlată, trebuie să aibă la sfîrșit valoarea N .

Nu s-a analizat cazul cind variabila de ciclare este controlată înainte de executarea ciclului. În acest caz, variabila de control poate fi mai întii controlată și apoi ciclată (dacă ciclul continuă să fie executat) sau să fie întii ciclată și apoi controlată.

În oricare din aceste cazuri există posibilitatea ca blocul respectiv de prelucrare din ciclu să nu fie executat.

De exemplu, dacă la citirea numerelor întregi de pe suport primul număr ar indica numărul de numere întregi existente pe suport, atunci controlul s-ar face astfel (fig. 3.4.1.):

- Se citește de pe suport valoarea variabilei de ciclare N .
- Se face ciclarea de citire folosind o variabilă de ciclare J inițializată cu valoarea zero.

Se consideră citirea terminată atunci cind $J = N$. Controlul variabilei se face înaintea operației de citire.

Schema logică este dată în figura 3.4.1.

În acest caz, ciclarea s-a făcut după control. Dacă $N = 0$ (nu se citește nici un element), ciclul de citire nu este executat. Variabila de ciclare are în final valoarea N . Dacă ciclarea s-ar face înainte de control (de decizie), atunci schema logică este cea prezentată în figura 3.4.2.

În această situație, variabila de ciclare are valoarea finală $N + 1$, de aceea se controlează pentru valoarea $N + 1$.

Altă situație este aceea cind variabila de ciclare se ciclează după executarea ciclului. În acest caz, în ciclu, variabila de ciclare are o valoare mai mică cu un pas de ciclare deci în cazurile precedente.

Schema logică este prezentată în figura 3.4.3.

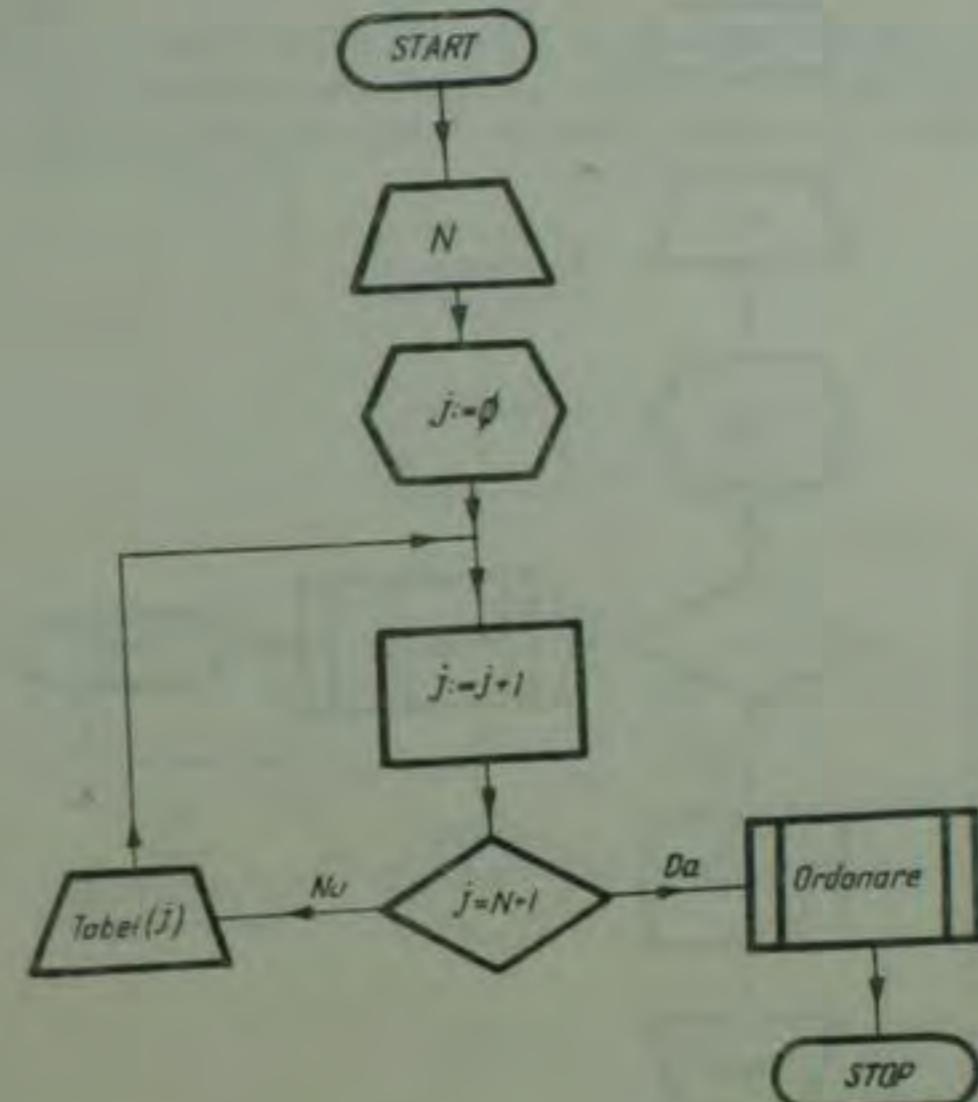


Fig. 3.4.1. Schema logică pentru controlul variabilei de ciclare înaintea operației de citire a valorii.

Indiferent de punctul în care se face controlul variabilei de ciclare, trebuie să se țină seama de faptul că, dacă ciclarea se face după executarea blocului, atunci în blocul de prelucrare (ciclare), variabila

va avea valoarea mai mică cu un pas (increment) de ciclare decit atunci cind este ciclată înaintea blocului de prelucrare.

Este important de reținut faptul că, în general, se recomandă să se facă controlul variabilei de ciclare înaintea executării blocului de prelucrare atunci cind există posibilitatea ca blocul să nu fie executat nici măcar o dată.

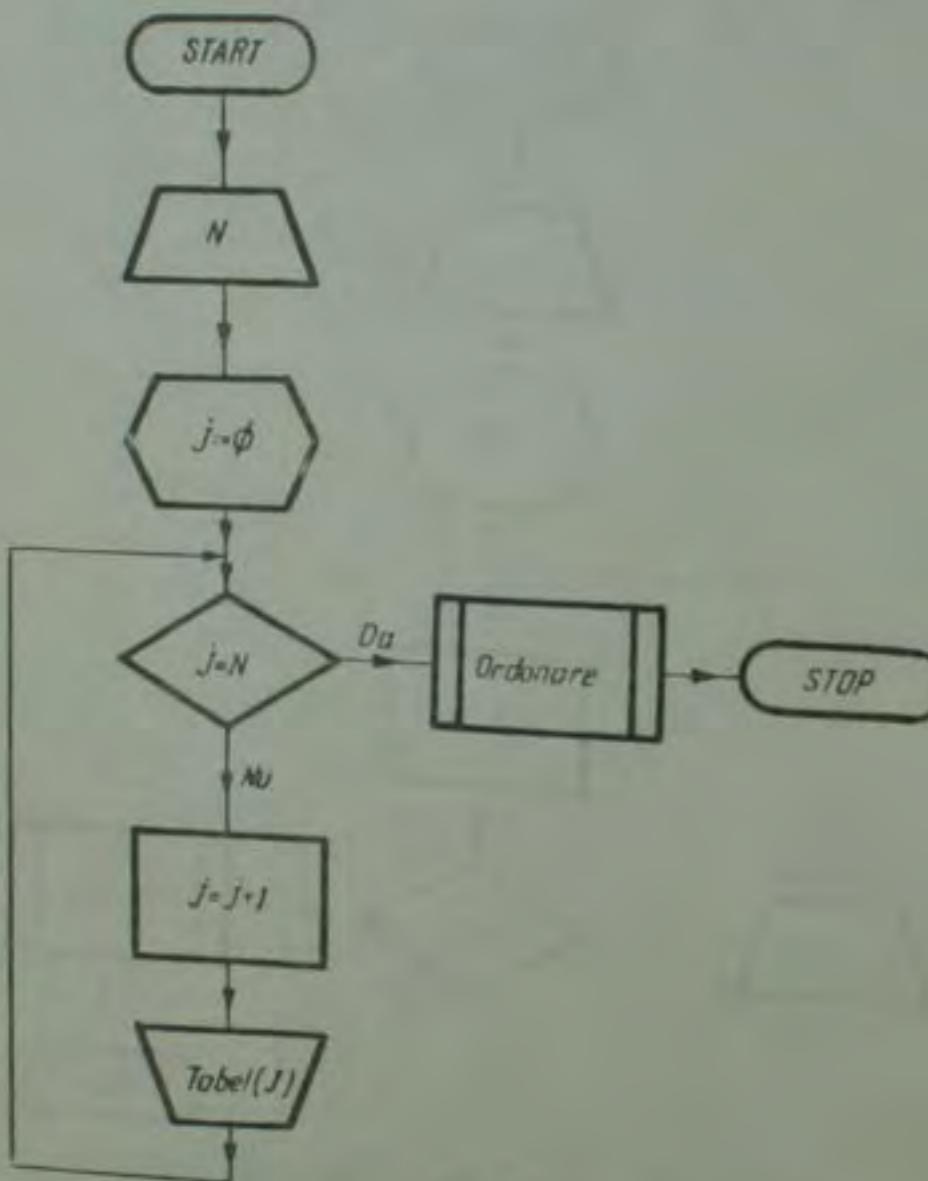


Fig. 3.4.2. Schema logică pentru controlul variabilei de ciclare înaintea blocului de decizie.

În exemplul prezentat, dacă N are valoarea zero, nu trebuie efectuată nici o citire de informație de pe suport în afara numărului N care se citește obligatoriu. Dacă controlul variabilei de ciclare s-ar

fi făcut după blocul de prelucrare, atunci ar fi fost citită cel puțin o informație de pe suport, ceea ce constituie o greșală care poate da efecte imprevizibile în prelucrare.

c. Operația de salt. Această operație poate fi rezultatul unei decizii sau poate fi impusă de necesitatea intreruperii secvenței normale a prelucrării și trecerea necondiționată la alt bloc de prelucrare din program.

Dacă saltul este CONDIȚIONAT, el apare ca rezultat al unei anumite decizii. De obicei se consideră saltul ca obligatoriu atunci cind condiția este îndeplinită (adică valoarea expresiei condiționale este

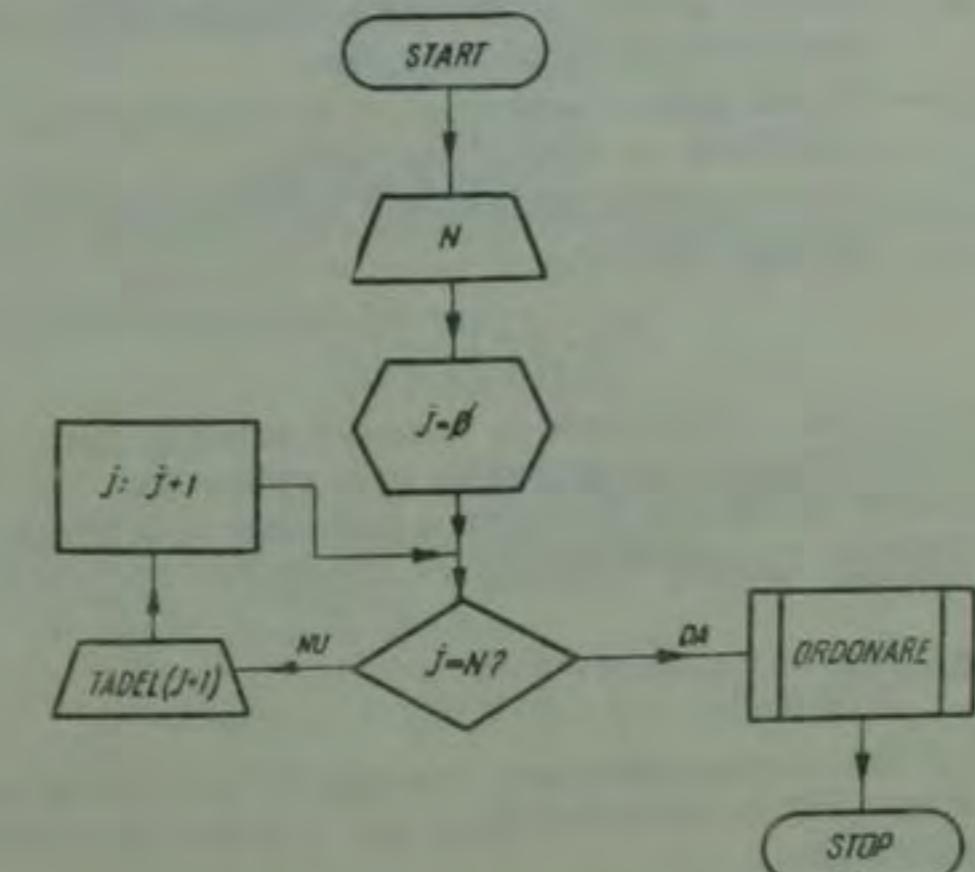


Fig. 3.4.3. Schema logică pentru controlul variabilei de ciclare înainte de citire.

TRUE). Dacă valoarea expresiei condiționale este FALSE (condiție neîndeplinită), se trece la blocul următor de prelucrare. Dacă apare necesitatea trecerii la alt bloc pe ramura NU, atunci se dă un salt necondiționat.

3.5. Utilizarea schemelor logice pentru calculul timpului de incărcare al calculatorului electronic

În vederea ilustrării utilizării schemelor de sistem, prezentăm în acest paragraf o metodă de calcul al timpului de incărcare orientată pe tranzacții, spre deosebire de metodele mai larg cunoscute, orientate pe instrucțiuni [7].

Timpul pentru efectuarea prelucrării necesare unei aplicații cu caracter economic (care utilizează fișiere pe benzi sau discuri magnetice) este compus din următoarele elemente principale :

T_B — timpul pentru actualizarea sau interogarea fișierelor constituite pe benzi magnetice

T_D — timpul pentru actualizarea sau interogarea fișierelor constituite pe discuri magnetice

T_P — timpul pentru prelucrările ce se efectuează în cadrul unității centrale

$$T = T_B + T_D + T_P$$

După cum se va vedea în cele ce urmează, calculul timpului de incărcare al calculatorului corespunzător unei aplicații se face, în principal, pornind de la schema de sistem a aplicației și determinând operațiile cu fișierele și tranzacțiile ce se efectuează asupra fișierelor.

1. Calculul lui T_B

Timpul pentru actualizarea sau interogarea unui fișier constituit pe bandă magnetică (acces secvențial) este compus din următoarele elemente principale :

- timpul de citire sau scriere (parcurserea benzii) — T_1
- timpul de transmitere a informațiilor de la unitatea de benzi magnetice la unitatea centrală — T_2
- timpul necesar sistemului de operare pentru a pregăti citirea sau scrierea — T_3

$$T_B = T_1 + T_2 + T_3$$

3.5. UTILIZAREA SCHEMELOR LOGICE

— T_1 se calculează după formula :

$$T_1 = (G \times I) \times B + J \times F$$

unde :

G este numărul de caractere ale fiecărei înregistrări ;

I — numărul de înregistrări (logice) ;

B — timpul de citire (scriere) al unui caracter ;

$$B \cong 17\mu s$$

J — numărul de blocuri (inregistrări fizice), respectiv de intervale între blocuri ;

F — timpul mediu de parcursere (inclusiv accelerare-frițare) pentru un interval ; $F \cong 8 ms$;

— T_2 se calculează după formula :

$$T_2 = I \times G \times C$$

unde :

I este numărul de înregistrări ;

G = numărul de caractere din fiecare înregistrare ;

C = timpul de transfer pentru un caracter ; $C \cong 2\mu s$;

— T_3 se calculează după formula :

$$T_3 = J \times b + I \times d$$

unde :

J = numărul de blocuri (inregistrări fizice) ;

I = numărul de înregistrări (logice) ;

b = timpul (necesar rutinelor) pentru citirea (scrierea) unui bloc ; $b = 5,5 ms$;

d = timpul pentru blocarea (debloarea) unei înregistrări (logice) intr-un (dintr-un) bloc ; $d = 0,5 ms$.

2. Calculul lui T_D

a. Timpul pentru actualizarea sau interogarea unui fișier constituit pe disc în acces secvențional :

$$D_1 = (G \times I) \times B + D \times J + K \times E$$

unde :

G este numărul de caractere dintr-o înregistrare (logică) ;

I — numărul de înregistrări (logice) ;

- B — timpul necesar citirii (scrierii) unui caracter;
 $B \cong 7\mu s$;
 D — timpul mediu de poziționare al capului de citire (scriere); $D \cong 12,5$ ms;
 J — numărul de blocuri;
 K — numărul de cilindri;
 E — timpul de acces între doi cilindri; $E = 25$ ms;

b. Timpul pentru actualizarea sau interogarea unui fișier constituit pe disc în *acces direct*:

$$D_2 = I(M + N + G \times B + D)$$

unde:

- I este numărul de înregistrări (logice);
 M = timpul mediu de utilizare a tabelei primare de indecsi (indecsiile cilindrilor); $M = 75$ ms;
 N = timpul de poziționare al capului de citire (scriere) și de utilizare a tabelei secundare de indecsi (indecsiile pistelor); $N \cong 110$ ms;
 G = numărul de caractere într-o înregistrare;
 B = timpul de citire (scriere) al unui caracter, ($B \cong 7\mu s$);
 D = timpul mediu de poziționare al capului de citire ($D \cong 12,5$ ms);

Pentru calculul timpului de transmitere a informațiilor între unităile de discuri și unitatea centrală și a timpului necesar sistemului de operare pentru a pregăti operațiile de citire sau scriere se poate folosi formula de aproximare:

$$T_4 = I \times 15 \text{ (ms)} \text{ unde } I \text{ este numărul înregistrărilor.}$$

Prin urmare, $T_D = D_2 + T_4$ (acces secvențial)

$$T_D = D_2 + T_4 \text{ (acces aleator).}$$

3. Calculul lui T_P

Se utilizează ca rezultate bune o formulă de medie, având ca parametru principal numărul de tranzacții pe baza cărora se efectuează prelucrările.

In vederea caracterizării complexității programelor se consideră trei valori la alegere pentru numărul mediu de instrucțiuni (i) necesar prelucrării tranzacțiilor:

$$T_P = n \times i \times t$$

unde:

- n = numărul tranzacțiilor;
 $i = 75, 150, 250$;
 t = timpul mediu necesar pentru executarea unei instrucțiuni.

În cazul cînd t nu este calculat special pentru o configurație și o anumită aplicație, se poate utiliza valoarea:

$$t \cong 60 - 100 \mu s.$$

Vom ilustra aplicarea acestor formule prin următoarele exemple:

Exemplul 1

Să considerăm prelucrarea corespunzătoare schemei de sistem:

F_1 este un fișier de tranzacții organizat secvențial pe disc, avind 20.000 înregistrări (logice) de lungime fixă 220 caractere, 615 blocuri (înregistrări fizice) organizate pe 16 cilindri.

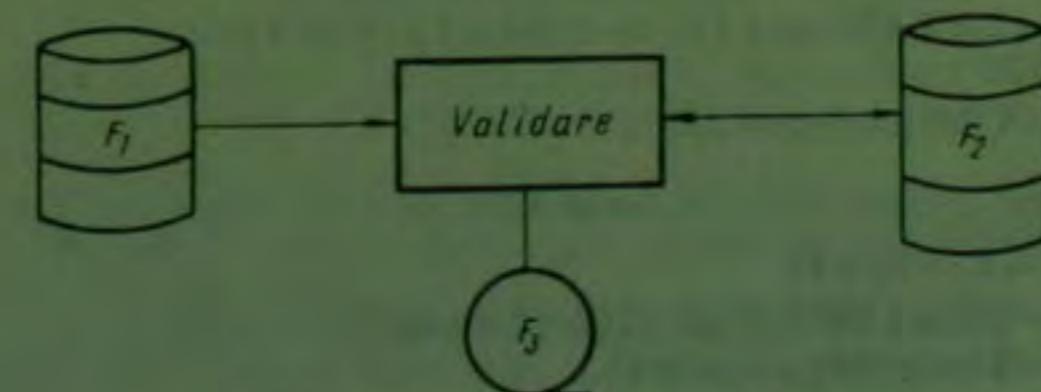


Fig. 3.5.1. Schemă de sistem cu 3 fișiere.

F_2 este un fișier de bază organizat pe disc în acces direct, avind 45.000 înregistrări logice de lungime fixă 300 caractere, organizat în 4 500 blocuri pe 120 cilindri.

F_3 este un fișier pe bandă pe care se înscriu 1 000 de înregistrări de lungime fixă 220 caractere organizate în 100 blocuri.

Presupunem că în vederea validării se execută următoarele activități:

1. Se citesc cele 20 000 de înregistrări din F_1 .
2. Se citesc cele 45 000 de înregistrări din F_2 .
3. Se inscriu 1 000 de înregistrări în F_2 .
4. Se inscriu 1 000 de înregistrări în F_3 .
5. Se prelucrează 20 000 de tranzacții.

Calculul timpului de încărcare al calculatorului

$$1. T_D = D_1 + T_4$$

$$D_1 = 220 \times 20\,000 \times 7 \mu s + 12,5 \text{ ms} \times 615 + 16 \times 25 \text{ ms} \cong 45''$$

$$T_4 = 20\,000 \times 15 \text{ ms} = 300''$$

$$T_D \cong 5'45''$$

$$2. T_D = D_2 + T_4$$

$$D_2 = 45\,000 (75 \text{ ms} + 110 \text{ ms} + 300 \times 7 \mu s + 12,5 \text{ ms}) \cong 2 \text{ h } 30'$$

$$T_4 = 45\,000 \times 15 \text{ ms} = 11'15''$$

$$T_D \cong 2 \text{ h } 41'15''$$

$$3. T_D = D_2 + T_4$$

$$D_2 = 1\,000 (75 \text{ ms} + 110 \text{ ms} + 300 \times 7 \mu s + 12,5 \text{ ms}) \cong 3'20''$$

$$T_4 = 1\,000 \times 15 \text{ ms} = 15''$$

$$T_D \cong 3'35''$$

$$4. T_B = T_1 + T_2 + T_3$$

$$T_1 = 220 \times 1\,000 \times 17 \mu s + 100 \times 8 \text{ ms} \cong 5''$$

$$T_2 = 1\,000 \times 220 \times 2 \mu s \cong 1''$$

$$T_3 = 100 \times 5,5 \text{ ms} + 1\,000 \times 0,5 \text{ ms} \cong 1''$$

$$T_B \cong 7''$$

5. Pentru a calcula timpul necesar prelucrărilor în unitatea centrală se folosește formula:

$$T_P = n \times i \times t$$

$$T_P = 20\,000 \times 250 \times 60 \mu s \cong 300'' = 5'$$

3.5. UTILIZAREA SCHEMELOR LOGICE

Prin urmare, timpul de încărcare al mașinii este:

$$T = T_B^4 + T_B^{1,2,3} + T_P^5$$

$$T = 5'45'' + 2 \text{ h } 41'15'' + 3'35'' + 7'' + 5' = 2 \text{ h } 55'42''$$

Exemplul 2

Să considerăm prelucrarea corespunzătoare schemei de sistem din figura 3.5.2.

F_1 este un fișier de tranzacții organizat secvențial pe disc, având 20 000 de înregistrări de lungime fixă, 220 caractere, organizate în 2 000 blocuri pe 50 cilindri.

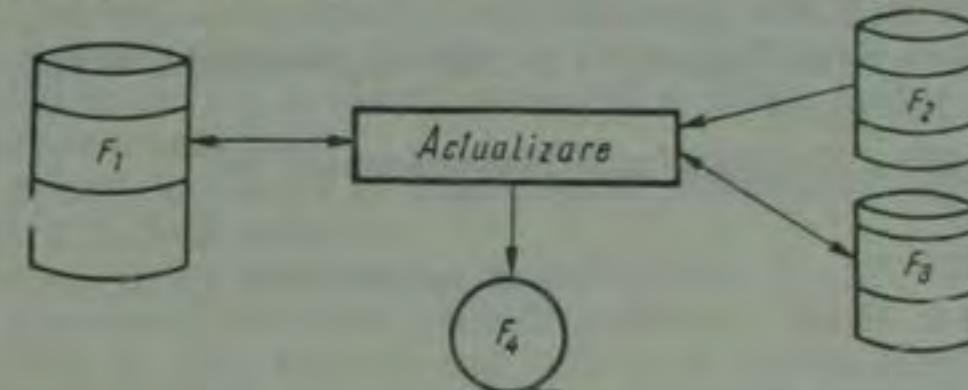


Fig. 3.5.2. Schemă de sistem cu 4 fișiere.

F_2 este un fișier de bază organizat în acces direct din care se citesc aleator 20 000 de înregistrări, având lungimea fixă de 300 caractere.

F_3 este un fișier variabil, organizat în acces direct, din care se citesc și în care se inscriu aleator cîte 20 000 de înregistrări de lungime fixă, 565 caractere.

F_4 este un fișier organizat pe bandă în care se inscriu 20 000 de înregistrări de lungime fixă, 220 caractere, organizate în 2 000 de blocuri.

Presupunem că în vederea actualizării fișierului variabil F_3 se execută următoarele activități:

1. Se citesc și se inscriu în F_1 cîte 20 000 de înregistrări.
2. Se citesc 20 000 de înregistrări din F_2 .
3. Se citesc și se inscriu în F_3 cîte 20 000 de înregistrări.
4. Se inscriu în F_4 , 20 000 de înregistrări.
5. Se prelucrează 20 000 de tranzacții.

$$1) T_D = D_1 + T_4$$

$$D_1 = 220 \times 20\,000 \times 7 \mu s + 12,5 ms \times 2\,000 + 50 \times 25 ms \cong 57"$$

$$T_4 = 20\,000 \times 15 ms = 300"$$

$$T_D = 5'57" (\text{citire}) + 5'57" (\text{scriere}) \cong 11'54"$$

$$2) T_D = D_2 + T_4$$

$$D_2 = 20\,000 (75 ms + 110 ms + 300 \times 7 \mu s + 12,5 ms) \cong 1 h 6'40"$$

$$T_4 = 5'$$

$$T_D \cong 1 h 11'4"$$

$$3) T_D = D_2 + T_4$$

$$D_2 = 20\,000 (75 ms + 110 ms + 565 \times 7 \mu s + 12,5 ms) \cong 1 h 7'20"$$

$$T_4 = 5'$$

$$T_D \cong 1 h 12'20"$$

$$T_D \cong 1 h 12'20" (\text{citire}) + 1 h 12'20" (\text{scriere}) \cong 2 h 24'40"$$

$$4) T_B = T_1 + T_2 + T_3$$

$$T_1 = 220 \times 20\,000 \times 17 \mu s + 2\,000 \times 8 ms \cong 1'30"$$

$$T_2 \cong 9"$$

$$T_3 \cong 21"$$

$$T_B \cong 2'$$

$$5) T_P = n \times i \times t$$

$$T_P = 20\,000 \times 250 \times 60 \mu s \cong 300" \cong 5'$$

$$T = T_B^4 + T_B^{1,2,3} + T_P^5 = 2' + 11'54" + 1 h 6'40" + 2 h 24'40" + 5'$$

$$T = 3 h 50'16"$$

În vederea luării în considerare a timpului necesar manipulărilor de către operatori, a timpului necesar încercărilor programelor, a eventualelor anomalii ce pot apărea etc., este indicat ca timpii de încărcare calculați să fie afectați de coeficienții de siguranță estimată pe baza experienței centrului de calcul respectiv, care pot afecta timpul de încărcare pînă la dublu.

De asemenea, este necesar ca pentru diferențele prelucrării corespunzătoare unor aplicații diferite ce se efectuează pe sistem să se asigure simultaneitatea operațiilor pe cît posibil în funcție de configurația existentă, operație care poate reduce semnificativ timpul total calculat prin adunarea timpilor de încărcare fără a se ține seama de aceste suprapunerile de activități posibile.

4. Înregistrarea și identificarea informațiilor în fișiere

4.1. Metode de înregistrarea informației

Cele mai obișnuite suporturi de informație sunt după cum am mai arătat: cartela perforată, banda de hîrtie perforată, banda magnetică, discul magnetic, tamburul magnetic și foliile magnetice (memoriile cu cartele magnetice).

Din punct de vedere al reprezentării informației, suporturile se pot împărți în două categorii:

a. **Suporturi cu reprezentare în cod extern.** În acest caz, pe suportul de informație respectiv (cartelă perforată, bandă perforată), reprezentarea se face într-un cod diferit de codul intern, binar al calculatorului. Codul de reprezentare este codul dispozitivului electromecanic pe care se perforează informația sau al dispozitivului electromagnetic pe care se face înregistrarea.

Transferul de informație de pe suport în memoria internă se face în acest caz prin conversiune de la codul extern al suportului, la codul intern al calculatorului. Conversiunea se face fie prin software (adică prin programe de conversiune), fie prin hardware (adică prin dispozitive electronice de conversiune speciale).

b. **Suporturi cu reprezentare în cod intern.** Suportul de informație în acest caz este banda magnetică, discul magnetic, memoria cu cartele magnetice sau uneori chiar banda de hîrtie perforată și cartela perforată. În acest caz, nu este necesară conversiunea din codul extern în codul intern al calculatorului, codurile fiind identice.

Codul intern este un cod binar, având un număr de 6 sau 8 biți pentru reprezentarea caracterelor, folosit în memoria rapidă a unității centrale din SPAD.

Codul de 6 biți se numește de obicei cod restrins, iar codul de 8 biți se numește cod extins de caractere.

După cum am văzut, din punct de vedere al metodei de acces, dispozitivele pot fi împărțite în două categorii (cap. II § 2) :

- dispozitive cu acces secvențial ;
- dispozitive cu acces direct.

Dispozitivele cu acces secvențial nu permit regăsirea informației precedente (care a fost deja citită) decât prin repetarea procesului de citire. Acestea sunt banda de hârtie perforată, cartela perforată și banda magnetică.

În cazul dispozitivelor cu acces direct, accesul nu se face direct la informația dorită, ci la zona din dispozitiv pe care se află informația și care poate fi selectată direct. Căutarea informației se face secvențial în zona respectivă ; o înregistrare pe o pistă a discului este căutată secvențial pe această pistă.

Se poate defini accesul direct ca o căutare în două etape :

- a. identificarea zonei din suport pe care se află informația ;
- b. căutarea secvențială a informației în zona în care se află informația, după ce zona a fost identificată și localizată direct.

Accesul secvențial, în general, este posibil pe orice dispozitiv. Accesul direct este posibil numai pe acele dispozitive pe care este posibilă selecționarea informației pe zone (de exemplu : cilindru și pistă la discuri).

În momentul în care un dispozitiv transmite informația de pe un suport oarecare de informație, conținutul informației pe suport rămâne nealterat. Aceasta este operația de intrare în unitatea centrală (citire de pe suport).

În momentul în care un dispozitiv primește informație (înregistrează informație), vechiul conținut se sterge și este înlocuit de noul conținut. Aceasta este operația de ieșire din unitatea centrală (scriere pe suport) sau de înregistrare sau înmagazinare. Aceste operații de ieșire dacă nu sunt efectuate cu atenție, pot conduce la stergerea unor informații care rămân pierdute, dacă nu există o copie a lor pe alt suport.

La intrare, informația de pe suport nu se alterează, în schimb în aria din unitatea centrală care primește informația se va sterge vechiul conținut, răminând la sfîrșitul operației cu conținutul nou, adică cu ceea ce i s-a transmis de pe suport.

În general, un dispozitiv periferic care acționează un suport de informație poate lucra cu mai multe arii de intrare/ieșire (I/E) din memoria rapidă. În acest fel se permite prelucrarea simultană cu

transferul de informație. În timp ce o arie I/E este umplută cu informație de pe suport, altă arie este supusă prelucrării de către unitatea centrală. Cind unitatea centrală cere o nouă informație, aceasta este deja pregătită în altă arie I/E.

În cazul în care s-ar lucra cu o singură arie, unitatea centrală ar trebui să aștepte pînă cînd informația este transferată în arie și apoi să prelucreze această informație. S-ar pierde astfel timpul de transfer ca timp de prelucrare.

Fiecare fișier constituit pe un suport trebuie să aibă o înregistrare specială la început (etichetă), care să-l identifice față de alte fișiere (de asemenei, pentru a marca începutul fișierului, data la care a fost creat, perioada de reținere etc.).

La fel la sfîrșitul fișierului trebuie să existe o înregistrare specială (etichetă), care să anunțe sfîrșitul datelor înregistrate pe fișier și eventual și alte caracteristici ale fișierului.

Indiferent de tipul de acces, orice fișier se prelucrează în același mod, utilizînd arii I/E în memoria rapidă. Zonele de identificare dintr-o înregistrare alcătuiesc după cum am văzut cheia înregistrării.

Cheile sunt utilizate fie de către sistemul de operare în timpul lucrului cu fișierele, fie de către programator, pentru găsirea anumitor înregistrări.

Informațiile de identificare din cheie care asigură accesul la înregistrările unui fișier de pe suport alcătuiesc cheia de acces ; aceste chei de regulă nu sunt transferate împreună cu înregistrarea în unitatea centrală. Informațiile de identificare din cheie, care asigură identificarea înregistrării în cadrul prelucrării în unitatea centrală, alcătuiesc cheia de identificare.

Pe categorii de fișiere (cap. II § 4) modul de utilizare a cheilor este următorul :

— Fișierele secvențiale. Nu necesită chei de acces în înregistrări care să fie utilizate de sistemul de operare. Înregistrările se citesc sau se înregistrează una după alta în ordinea fizică a aparției lor.

— Fișiere în acces direct cu cheia numerică. Aceste fișiere necesită chei de acces pe care sistemul de operare le utilizează. Cheia de acces, în acest caz, poate să nu aparțină înregistrării. Această cheie, reprezintă de fapt poziția înregistrării în fișier. Sistemul de operare selecționează automat zona de acces în care se află înregistrarea.

Cind cheia de acces este externă înregistrării, poate fi scrisă pe suport înaintea înregistrării. La citirea înregistrării, dacă cheia există

pe suport în afara înregistrării, este citită în aria separată sau ignorată; în aria I/E se citește numai înregistrarea fără informațiile care alcătuiesc cheia (de acces).

— Fișiere în acces direct cu cheie alfanumerică.

In general, în acest caz, cheia trebuie să existe în înregistrare și de asemenea, zonele care formează cheia trebuie să fie consecutive și adiacente. De fapt, cheia poate fi reprezentată în acest caz ca o singură zonă.

Pot exista și sisteme de operare, care, în general, să nu ceară consecutivitatea și adiacența zonelor care formează cheia.

— Fișiere indexat secvențiale.

Acste fișiere nu pot fi create decât secvențial. Ele au o cheie alfanumerică, iar înregistrările trebuie sortate în ordinea crescătoare a cheilor. Sunt respinse la crearea fișierului acele înregistrări care nu au cheia în secvență. Cheile duble de asemenea nu sunt admise, deoarece la citirea acestor chei ar rezulta situații de ambiguitate.

La aceste fișiere se alocă un anumit număr de arii (piste, cilindri) de depășire pentru înregistrările care se adaugă (cap. II § 4). Adăugarea acestora se face în secvență normală, sistemul de operare stabilind locul, adică cilindrul, pista și adresa înregistrării fizice.

Există și sisteme de operare care să admită pentru fișierele indexat secvențiale chei alcătuite din zone separate neconsecutive și neadiacente.

La fișierele în acces direct cu cheie alfanumerică și la fișierele indexat secvențiale se specifică locul cheii în înregistrare.

Sistemul de operare introduce în aria I/E numai înregistrarea propriu-zisă din fișier. Sunt cazuri cînd se aduce în aria I/E și cheia existentă în față, dar aceasta poate fi ușor separată prin program. Uneori, cînd cheia este citită, se aduce într-o aria special rezervată.

Avantajul fișierelor indexat secvențiale constă în faptul că permit adăugiri ulterioare de înregistrări, pe care sistemul de operare le ordonează în secvență. În cazul cînd pistele de depășire pe unul din cilindri se umplu, atunci automat înregistrările pot fi aranjate pe alți cilindri de depășire. Stabilirea secvenței și a legăturii cu celelalte înregistrări cade în general în sarcina sistemului de operare.

La fișierele cu acces direct cu cheie alfanumerică, adăugarea de noi înregistrări poate fi făcută de asemenea de sistem însă, în cazul în care fișierul se umple sau apar chei duble, trierea legăturilor între înregistrări cade în sarcina programatorului.

La fișierele secvențiale sau la fișierele în acces direct cu cheie numerică nu pot fi adăugate (intercalate) înregistrări noi pe același fișier, fără a se efectua reorganizarea fișierului.

4.2. INREGISTRAREA ȘI REGASIREA INFORMATIEI PE DISCURI

Pentru a putea prelucra volume mari de date sau informații pe SPAD, este necesar, după cum am văzut, să existe suporturi pe care datele și informațiile (datele prelucrate) sint înregistrate și păstrate.

Suportul de informație are menirea de a pune la dispoziția unității centrale, informația necesară și de a înregistra informația primită de la unitatea centrală.

În primele etape ale dezvoltării SPAD, suportul de informație îl constituia cartela perforată sau banda de hirtie perforată.

După cum am văzut (cap. 1), pe măsura extinderii aplicațiilor pe SPAD, a apărut însă necesitatea stocării și prelucrării unui volum mare de date și necesitatea selectării rapide a informației.

Dacă, de exemplu, este necesară informația de pe a 100-a carte dintr-un pachet de 1000 de cartele, nu există nici o posibilitate de citire directă a cartelei 100 din pachet.

În cazul în care suportul de informație ar fi banda magnetică, situația rămîne aceeași.

Dacă nu se poate reține toată informația de pe o bandă magnetică în memoria centrală, atunci citirea unei informații care ar precede informația citită ultima oară, implică de regulă rebobinarea benzii și repetarea proceseului de căutare.

În acest caz, informația poate fi prelucrată numai secvențial (pas cu pas), întoarcerile necesitând tempi relativ mari de rebobinarea benzii și de repetare a procesul de căutare.

Pentru înlăturarea unor asemenea inconveniente, au fost construite dispozitive care permit accesul direct la informația dorită, sau la un grup mic de înregistrări, care conțin informația.

Dacă accesul se face la un grup de înregistrări, care conțin informația căutată, atunci în acest grup căutarea se face secvențial. Avantajul în acest caz este de multe ori considerabil. Un astfel de dispozitiv în acces direct este după cum am văzut discul magnetic.

Un disc magnetic este format dintr-un număr de plăci cilindrice (discuri) susținute rigid de un ax care se rotește odată cu plăcile.

Fiecare placă are două fețe. De obicei, fața de deasupra primei plăci și fața de jos a ultimei plăci nu sunt utilizate pentru înregistrarea informației utilizatorului.

Astfel, dacă un disc (se mai spune și pachet de discuri) are p plăci și $2p$ fețe, pentru înregistrarea informației sunt disponibile numai $2p - 2$ fețe (fig. 4.2.1).

Pe fiecare din fețele discului se află un număr de piste reprezentate ca cercuri concentrice. Pe un braț mobil, care se mișcă perpendicular pe fețele discului se află, pentru fiecare față, un cap de înregistrare (fig. 4.2.1.), prin care se efectuează citirea și scrierea.

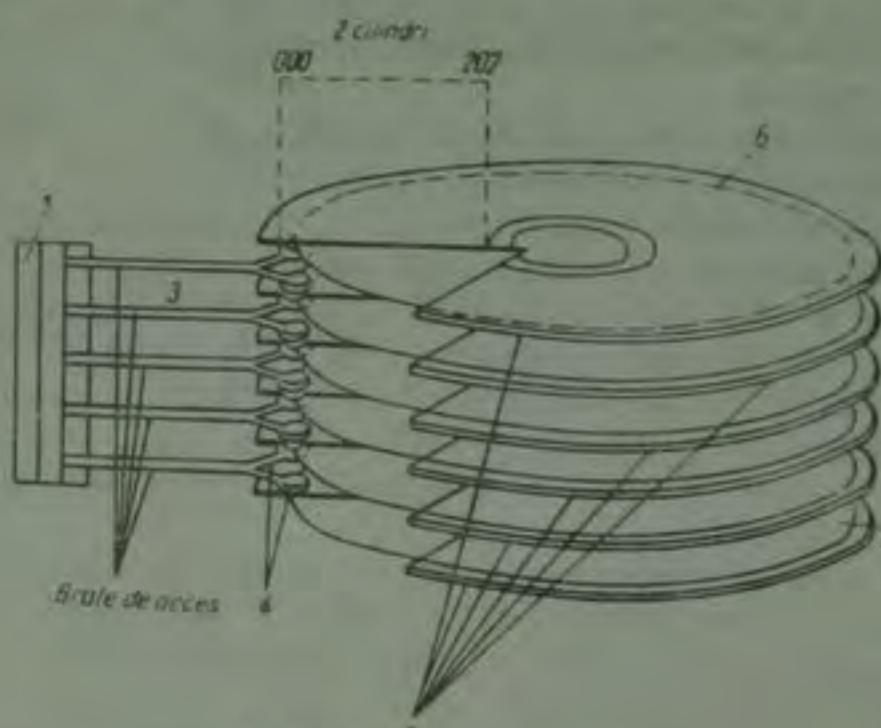


Fig. 4.2.1. Reprezentarea unui disc.

Dacă pe o față se află N piste și dacă n este numărul fețelor active, discul are un număr de $n \cdot N$ piste. Pe fiecare pistă se poate înregistra un număr de caractere, același, indiferent de poziția pistei pe disc.

Toate capetele de înregistrare se află așezate rigid pe brațul vertical care se mișcă, astfel încât poziționarea unui capăt de înregistrare pe o anumită pistă (de exemplu pistă 100) face ca toate capetele de înregistrare să se poziționeze la fel pe celelalte fețe. Pistele pe care au fost poziționate capetele de înregistrare, văzute de sus, apar ca o singură pistă.

Sectionând un pachet de discuri cu un cilindru, toate pistele care se găsesc pe acest cilindru se spune că formează un cilindru pe disc.

Se poate spune că pachetul de discuri are N cilindri.

Un cilindru are un număr de piste egal cu numărul fețelor active, notat mai sus cu n .

Acest mod de organizare a discului permite unele facilități în privința înregistrării informației.

Dacă o informație se află pe un anumit cilindru și pe o anumită pistă de pe acel cilindru, atunci prin cunoașterea cilindrului și a pistei se poate identifica direct zona de pe disc (pista) pe care se află înregistrarea respectivă.

Fie un pachet de discuri care are 10 de fețe active, iar pe fiecare față 400 de piste.

În total acest pachet de discuri are $20 \times 400 = 8000$ de piste.

Repartizarea se face astfel :

Se consideră că pachetul are 400 de cilindri, corespunzători celor 400 de piste.

Numerotarea cilindrilor se incepe cu zero și se termină cu 399.

Pistele se numerotează pe cilindri astfel :

— pe cilindrul 0, pe prima față activă se află pistă numărul zero; pistă de pe cea de a doua față activă este pistă numărul 1 etc.; pistă de pe ultima față activă este pistă numărul 19.

Putem spune deci că o înregistrare se află pe cilindrul N_c ($0 \leq N_c \leq 399$) și pe pistă p_c ($0 \leq p_c \leq 19$) de pe cilindrul N_c .

În general se folosește numerotarea pistelor de la 0 pînă la $n \times N - 1$. Pe cilindrul 1 se află pistele 0 pînă la 19 pe cilindrul 2 se află pistele 20 pînă la 39 și aşa mai departe, pe cilindrul i se află pistele $(i-1) \times 20$ pînă la $i \times 20 - 1$. Pe ultimul cilindru se află pistele de la $(400-1) \times 20 = 7980$, pînă la $400 \times 20 - 1 = 7999$.

În acest caz, dacă se cunoaște numărul unei piste, fie acesta I , atunci cilindrul pe care se află pistă I , se determină prin împărțirea cu neglijarea restului, a numărului pistei la numărul de pistă pe cilindru.

Deci :

$$N_c = \text{INTREG}(I/n)$$

Numărul pistei pe cilindru se află prin considerarea restului, adică :

$$p_c = \text{REST}(I/n)$$

unde p_c este numărul pistei pe cilindru.

Pozitionarea capetelor de înregistrare se face găsind numărul cilindrului pe care se află înregistrată informația sau pe care urmează să se inscrie această informație; după pozitionarea capetelor de înregistrare, dispozitivul selectează pista pe care trebuie căutată (inscrisă) înregistrarea.

Căutarea pe pistă se face secvențial, la fel ca la banda magnetică. Discul rotindu-se pentru a găsi înregistrarea dorită, pista trebuie citită de la început, înregistrare cu înregistrare, pînă cînd se identifică înregistrarea dorită.

În general, toate tipurile de suport de informație au înregistrările formate din caractere, reprezentate în cod binar.

Unui caracter îi corespunde în general un semn pe tastatura unui dispozitiv de imprimare, perforare, înregistrare sau citire. Corespondența între valoarea binară și caracter poate fi în principiu oricare. Există deci mai multe coduri de caractere atît în setul cu 64 de caractere (cod restrins), cît și în setul cu 256 de caractere (cod extins). Pentru ca un cod să poată fi utilizat drept cod intern într-un SPAD, trebuie să indeplinească următoarele condiții :

- codurile binare ale literelor alfabetului să apară în ordine crescătoare ca numere naturale, conform ordinea literelor din alfabet,

- codul echivalent cifrelor trebuie să indeplinească aceeași condiție ca și codul literelor. Dacă cifra zero are în cod binar valoarea 11110000, atunci cifra 9 trebuie să aibă obligatoriu codul binar 11111001. Aceasta înseamnă că cifra 8 care precede cifra 9, va avea codul 11111000. De la cifra zero pînă la cifra 9, codul binar al fiecărei cifre crește cu o unitate. Se pot astfel distinge ușor cifrele de alte caractere. În exemplul de mai sus, cifrele au cod echivalent zecimal cuprins între 240 și 249. Oricare caracter care în codul binar capătă valori între 240 și 249 reprezintă o cifră. La fel se prezintă și codul literelor care poate fi considerat de exemplu între valorile zecimale 193 și 233 inclusiv (conform alfabetului englez). Oricare caracter avind codul binar între valorile 193 și 233 poate reprezenta o literă mare. Pentru literele mici se poate alege o altă secvență din cod după aceeași criterii. Restul caracterelor din cod pot fi reprezentate după alte criterii. Pentru a se mări viteza de transfer, informația se ține pe bandă magnetică sau pe disc magnetic, sub formă de caractere în codul intern al calculatorului. În acest caz, nemaifăcindu-se convingerea de la un cod la altul, se obține o mărire importantă a vitezei de transfer a datelor.

4.3. Structura fișierelor

Fie N numărul total al înregistrărilor logice dintr-un fișier.
Fie L_I lungimea înregistrării I din fișier.

Fie N_I numărul de zone al înregistrării I .

Fie l_1, l_2, \dots, l_{N_I} , lungimea fiecărei zone Z_1, Z_2, \dots, Z_{N_I} exprimată în număr de caractere.

Dacă $L_1 = L_2 = \dots = L_N$, atunci fișierul, după cum am văzut, conține înregistrări de lungime fixă.

Operația de intrare-iesire (I/E), se face prin intermediul unității de control a echipamentului periferic, a canalului dintre unitatea de control și unitatea centrală și a unității centrale (fig. 4.3.1.) care utilizează pentru transferul înregistrării I o arie de memorie (I/E) de lungime L_I .

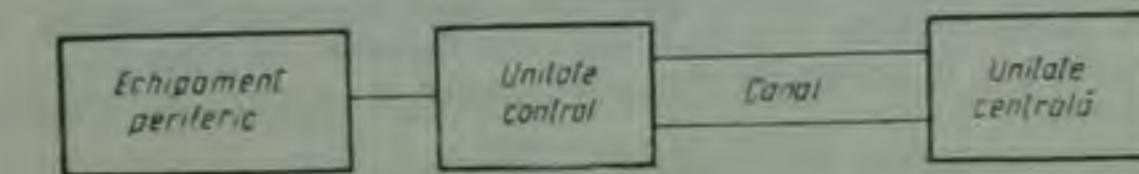


Fig. 4.3.1. Modul de transfer al datelor.

Pentru o serie de situații, sistemul de operare poate rezolva toate problemele de intrare-iesire, fără intervenția programatorului.

Dacă $L_I \neq L_J$ (pentru I și J numere naturale), atunci fișierul, după cum am văzut (cap. 1), este un fișier cu înregistrări de lungime variabilă.

Sistemele de operare pun de asemenea la dispoziția programatorului rutine pentru lucrul cu fișiere care au înregistrări de lungime variabilă.

Pentru înregistrările de lungime variabilă, trebuie reținută în înregistrare o informație privind lungimea înregistrării și factorul de blocare. Pentru aceasta, se alocă primele două zone de înregistrare care sunt identice pentru toate înregistrările din punct de vedere al structurii.

Fie Z_1 și Z_2 , cele două zone din înregistrarea I pe care se află informația privind lungimea înregistrării, și respectiv factorul de blocare.

În general Z_i are o lungime de 4 pînă la 5 caractere, depinzînd de tipul suportului de informație și de calculator. Pentru factorul de blocare se alocă obînuit un spațiu de 3 caractere.

În cazul fișierelor care nu se blochează, se alocă uneori numai spațiului din Z_i , corespunzător lungimii înregistrării.

Lungimea înregistrării trebuie scrisă de către utilizator, aceasta fiind o informație necesară pentru sistemul de operare. Factorul de blocare poate fi scris de utilizator sau elaborat automat de sistemul de operare. Factorul de blocare este de obicei același pentru toate înregistrările, poate fi însă și variabil (sistemul de operare neputînd fi utilizat în acest caz).

Să reținem că lungimea fiecărei înregistrări în parte și uneori factorul de blocare sunt scrise de către utilizator și că acesta are întreaga răspundere asupra corectitudinii acestor date. Pentru sistemul de operare se dău informații asupra înregistrării de lungime maximă și a factorului de blocare dacă este cazul.

Aria I/E rezervată în acest caz, corespunde înregistrării de lungime maximă din fișier. Dacă fișierul este blocat (factor de blocaj >1), atunci se alocă această arie înmulțită cu factorul de blocare.

Fișierele pot fi împărțite din acest punct de vedere în următoarele categorii :

a) **Fișiere omogene.** Sunt fișiere pentru oricare ar fi I, J și i, j aparținînd șirului numerelor naturale asociate înregistrărilor și respectiv zonelor, sătisfăcute condițiile :

$$L_I = L_J \quad (1)$$

$$l_I = l_J \quad (2)$$

$$Z_I \simeq Z_J \quad (3)$$

Simbolul \simeq arată că informațiile conținute în zonele Z_I și Z_J sunt de același tip (aritmetice intregi, aritmetice în virgulă mobilă, logice sau alfanumerice). În unele cazuri se va considera valoarea spațiu a cîmpului drept echivalentă valorii aritmetice zero.

Fișierul omogen poate fi considerat ca o matrice avînd m linii și n coloane, elementele matricii fiind de același tip pe fiecare coloană (matrice de caractere, care are un număr de coloane egal cu lungimea înregistrării și un număr de linii egal cu numărul de înregistrări).

Pentru fișierele omogene, există de regulă, o singură procedură de prelucrare a tuturor înregistrărilor.

b) **Fișiere parțial omogene.** Aceste fișiere satisfac relația (1), dar nu satisfac relațiile (2) și (3).

Se poate face o împărțire a înregistrărilor pe clase de înregistrări, obînindu-se astfel clase care satisfac relațiile (1), (2) și (3).

Pentru prelucrarea acestor fișiere sunt necesare, de regulă, atîtea proceduri de prelucrare cîte clase de înregistrări sunt.

Fișierele parțial omogene, pot fi reprezentate ca mai multe matrici care au fiecare din ele proprietățile matricii corespunzătoare fișierului omogen. O anumită clasă de înregistrări, nu este neapărat formată din înregistrări succesive, poziția înregistrărilor în fișier putînd fi oricare.

c) **Fișiere neomogene.** Aceste fișiere nu satisfac condițiile (1), (2) și (3).

Esențial la aceste fișiere este faptul că nu satisfac condiția (1).

Și aceste fișiere pot fi repartizate în clase de înregistrări parțial omogene, care la rîndul lor pot fi împărțite în alte clase de înregistrări omogene, avînd în vedere că diverse clase pot avea înregistrări de lungimi diferite.

Fișierele neomogene pot deci conține clase de înregistrări de tipul fișierelor parțial omogene și fișierelor omogene.

Împărțirea în clase de înregistrări parțial omogene și omogene poate fi făcută chiar dacă o clasă este formată numai dintr-o înregistrare.

4.4. Identificarea și validarea înregistrărilor

În cazul fișierelor omogene, pozițiile zonelor de identificare în înregistrări sunt aceleași. Fie de exemplu Z_1, Z_3 și Z_8 zonele de identificare ale înregistrărilor dintr-un fișier omogen.

Informația conținută în aceste zone trebuie să permită găsirea unei anumite înregistrări din fișier.

Dacă, de exemplu, se caută o înregistrare care să aibă în Z_1 , un cod numeric egal cu 0025, în Z_3 un cod alfabetic ABCD, iar în Z_8 caracterele speciale (%), atunci această înregistrare poate fi identificată în cazul accesului secvențial, prin citirea succesivă a înregistrărilor pînă cînd se găsește o înregistrare care conține în zonele de identificare valorile specificate mai sus.

Prin concatenarea (unirea, alipirea) zonelor Z_1 , Z_3 și Z_5 , se formează o cheie, pe care o vom numi, în acest exemplu, KEY:

KEY := 0025ABCD(%)

Această cheie este alfanumerică.

Tinând seama că pe suportul de informație caracterele sunt reprezentate prin valoarea lor binară, această cheie are o valoare binară formată din 88 de poziții binare în cod extins.

În cazul fișierelor cu acces aleator, se impune în general ca cheia să fie un sir de caractere succesive, fără intercalarea altor zone de caractere.

Fie de exemplu un fișier omogen avind lungimea fiecărei înregistrări de 24 de caractere. Matricea (tabelul caracterelor) se prezintă ca în figura 4.4.1.

Împărțirea pe zone:

Z_1 =coloanele de la 1—8

Z_2 =coloanele de la 9—10

Z_3 =coloanele de la 11—14

Z_4 =coloanele de la 15—18

Z_5 =coloanele de la 19—24

Subzone:

Z_{31} =coloanele 11—12

Z_{32} =coloanele 13—14

Z_{41} =coloanele 15—16

Z_{42} =coloanele 17—18

Z_{51} =coloanele 19—20

Z_{52} =coloanele 21—22

Z_{53} =coloanele 23—24

În prima linie este scris numărul zonei, iar în linia a doua sunt numerotate coloanele de la 1 la 24.

Semnificația zonelor este următoarea:

Zona 1, un nume format din 8 caractere alfanumerice. Poate fi orice combinație de caractere alfabetice, numerice, spații și alte caractere.

Zona 2, reprezintă un număr de ordine al numelui (zona 1), într-un alt fișier (registru de evidență manuală) din care acesta a fost extras. Această zonă este numerică, nu admite spațiu în loc de cifra zero. Are lungimea de două caractere.

1	1	2	3	4	5
II	1 2 3 4 5 6 7 8	9 10	11121314	15161718	192021222324
1	I O N E S C U	0 1	1 0 2 0	2 0 0 0	1 2 0 2 7 0
2	P O P E S C U	0 3	1 0 2 5	1 1 1 1	1 5 0 2 7 0
3	V A S I L E	5 4	1 1 4 0	1 4 3 0	1 4 0 4 6 9
4	P E T R E	6 5	1 6 2 0	1 8 2 0	1 2 0 3 6 9
5	S T E F A N	2 2	1 6 1 0	1 9 4 5	1 2 0 3 6 9
6	G H E O R G H E	2 3	1 5 2 5	1 6 4 0	1 2 0 3 6 9
7	Z A M F I R	1 9	1 5 2 5	1 5 3 0	1 5 0 1 6 7
8	A P O S T O L	4 4	1 4 2 0	1 4 4 4	2 5 1 2 6 8
9	M I H A I	3 3	0 9 1 0	1 0 2 5	2 4 1 2 6 8
10	N A E	9 9	1 8 2 0	1 9 1 9	1 6 1 2 6 6
11	M A R I N	2 5	0 7 0 0	0 8 0 0	1 5 1 2 7 0
12	A L B A L U X	1 5	0 9 0 0	0 9 1 2	1 5 1 2 7 0
13	A L B O L	0 4	1 2 2 5	1 4 0 0	1 5 1 2 7 0
14	M E S T E R	0 2	1 1 0 6	1 5 0 0	1 5 1 2 7 0
15	M A I S T R U	1 2	1 0 0 3	1 2 3 5	1 0 0 4 6 5
16	T E H N I C	1 6	0 8 0 0	1 4 4 0	1 2 0 6 6 6
17	B I R O U R I	1 7	0 7 0 1	0 8 0 1	0 1 0 1 7 0
18	V O P S E L E	1 8	0 5 0 5	0 6 0 6	0 1 0 2 7 0
19	C 1 7 5 2 4 1	3 1	1 0 1 1	1 1 4 4	0 1 0 5 6 5

Fig. 4.4.1. Exemplu de fișier.

Zona 3, reprezintă ora și minutul la care s-a făcut înregistrarea numelui din zona 1 în registrul (fișier) de evidență. Are 4 caractere numerice.

Zona 4, reprezintă ora și minutul la care s-a făcut extragerea numelui din zona 1 din registrul (fișierul) de evidență. Are 4 caractere numerice.

Zona 5, reprezintă data înregistrării și extragerii numelui respectiv în (din) registrul (fișierul) de evidență, dată sub forma : ziua, luna, anul. Are 6 caractere numerice. (Se consideră că intrarea și ieșirea din registrul de evidență se fac în aceeași zi, altfel trebuind să fie scrise ambele date).

Fișierul fiind omogen, vom considera că prelucrarea este aceeași pentru oricare din cele 19 înregistrări ale fișierului.

Se va arăta în continuare că sunt procedeele de validare pentru fiecare zonă din fișier.

Zona 1. Cimp alfanumeric de 8 caractere. Nu se face nici un fel de verificare a cimpului.

Zona 2. Această zonă trebuie să fie :

- de două caractere lungime ;
- numerică (nu admite alte caractere decât cifrele de la zero la 9).

Zona 3. Această zonă trebuie să fie :

- de 4 caractere lungime ;
- numerică.

Primele două caractere reprezintă ora, iar următoarele 2 caractere reprezintă minutul. Condiția suplimentară pentru această zonă este aceea că numărul format din caracterele din coloana 11 și 12, fie acesta Z_{31} , trebuie să satisfacă condiția :

$$0 \leq Z_{31} \leq 24$$

Pentru că evident, într-o zi avem numai 24 de ore.

Numărul format din caracterele din coloanele 13 și 14, fie acesta Z_{32} , trebuie să îndeplinească condiția :

$$0 \leq Z_{32} \leq 59$$

deoarece reprezintă minute.

Zona 4. Tratamentul pentru această zonă este identic cu cel al zonei 3. Numărul format din caracterele din coloanele 15 și 16

se tratează identic cu cele din coloanele 11 și 12 din zona 3, iar numărul format din coloanele 17 și 18 se tratează identic cu cel format din coloanele 13 și 14 din zona 3.

Zona 5. Această zonă trebuie să fie :

- de 6 caractere lungime ;
- numerică.

Primele două caractere (coloanele 19 și 20) reprezintă ziua, notată cu Z_{51} , următoarele două caractere (coloanele 21 și 22) notate cu Z_{52} , reprezentând luna, iar ultimele două caractere (coloanele 23 și 24) notate cu Z_{53} reprezintă anul.

Condițiile care se testează sint următoarele :

$$\text{dacă } 1 \leq Z_{51} \leq 31,$$

$$\text{dacă } 1 \leq Z_{52} \leq 12,$$

$$\text{dacă } 45 \leq Z_{53} \leq \text{anul curent.}$$

În cazul de față se pot face controale și pentru subzonele Z_{51} , Z_{52} și Z_{53} , ținând cont de relațiile dintre acestea.

Dacă Z_{52} (adică luna) are una din valorile 01, 03, 05, 07, 08, 10, 12, atunci Z_{51} (adică ziua) trebuie să fie mai mic sau egal cu 31. Dacă Z_{52} are una din valorile 04, 06, 09, 11, atunci Z_{51} trebuie să fie egal sau mai mic decât 30. Dacă Z_{52} (adică luna), este egal cu 2, iar Z_{53} (adică anul) este multiplu de 4, deci este bisect, atunci Z_{51} trebuie să fie mai mic sau egal cu 29, altfel, dacă $Z_{52}=2$ și Z_{53} nu este multiplu de 4, atunci Z_{51} trebuie să fie mai mic sau egal cu 28 (adică an nebisect).

De asemenea, între valorile cimpurilor Z_3 și Z_4 , există relația :

$$Z_4 > Z_3,$$

adică ora ieșirii trebuie să fie mai mare ca ora intrării.

În această verificare cimpul Z_1 nu are nici o importanță, deoarece poate cuprinde orice caracter.

Programul de control selecționează erorile și tipărește înregistrările greșite. La începutul lucrului cu un fișier se face deschiderea acestuia și controlul caracteristicilor fișierului prin comparare cu caracteristicile cerute de programator.

La terminarea lucrului cu un fișier se face închiderea fișierului, anunțând prin aceasta sistemul de prelucrare automată a datelor, că prelucrarea fișierului a luat sfîrșit.

În exemplul arătat mai sus, fișierul din figura 4.4.1 a fost utilizat pentru ilustrarea validării cimpurilor din punct de vedere al

tipurilor de date și al limitelor între care pot varia datele. O astemenea validare (verificare sintactică), se referă numai la corectitudinea datelor din punct de vedere formal și nu al conținutului că atare. De exemplu, dacă în zona 3 din înregistrarea 8, în loc de ora 14^{20} , trebuie înscrisă ora 12^{20} și greșit s-a transmis drept oră de intrare 14^{20} , procedurile de verificare sintactică nu au cum să sesizeze acest lucru.

De multe ori datele dintr-un fișier sunt necesare într-o anumită ordine. Așezarea datelor în ordinea dorită, se poate face prin așezarea înregistrărilor în ordinea corespunzătoare.

Astfel, să presupunem că fișierul din figura 4.4.1 este necesar să fie organizat (sortat) în ordinea datelor calendaristice, a orelor de intrare și a orelor de ieșire.

Se observă că cea mai mică dată de intrare din fișier este 10 04 65 (adică ziua 10 luna aprilie, anul 1965). Această dată se află în înregistrarea nr. 15. Această înregistrare trebuie să apară prima în fișier. Se aleg astfel în ordine înregistrările respective. Există însă cazuri ca ale înregistrărilor 12, 13 și 14, care au aceeași dată calendaristică de intrare, 15 12 70 (adică 15 decembrie 1970). Luând în considerare însă ora intrării, se observă că înregistrările 13 și 14 trebuie schimbată între ele. Dacă și ora de intrare ar fi fost aceeași, atunci s-ar fi luat în considerare ora de ieșire, altfel înregistrările rămânind în aceeași poziție relativă.

Pentru sortare se utilizează așa numitele *chei de sortare*, care sunt zonele din înregistrări (cheile de identificare), după care se face sortarea.

Fișierul din figura 4.4.1 este utilizat din nou mai jos și înregistrările sunt împărțite în zone astfel:

Z_1 =caracterele	1—8
Z_2 =	9—10
Z_3 =	11—12
Z_4 =	13—14
Z_5 =	15—16
Z_6 =	17—18
Z_7 =	19—20
Z_8 =	21—22
Z_9 =	23—24

În cazul sortării se formează o singură cheie prin concatenarea cimpurilor care intră în compoziția cheii de sortare.

În conformitate cu cele arătate, cheia de sortare va fi formată din:

Z_9 =caracterele	23—24	reprezentînd anul
Z_8 =	21—22	„ luna
Z_7 =	19—20	„ ziua
Z_3 =	11—12	„ ora intrării
Z_4 =	13—14	„ minutul intrării
Z_5 =	15—16	„ ora ieșirii
Z_6 =	17—18	„ minutul ieșirii.

Folosind pentru concatenare semnul ||, atunci rezultă că cheia de sortare KEY este obținută prin concatenarea cimpurilor arătate sub forma unui singur sir de caractere :

$$\text{KEY} := Z_9 || Z_8 || Z_7 || Z_3 || Z_4 || Z_5 || Z_6 \quad (1)$$

Valoarea cheii se poate exprima prin poziția caracterelor :

$$\text{KEY} := (23, 24, 21, 22, 19, 20, 11, 12, 13, 14, 15, 16, 17, 18)$$

numerele din paranteză reprezentînd pozițiile caracterelor în înregistrare. De exemplu, cheia de sortare pentru înregistrarea 10 va fi :

$$\text{KEY} := 66\ 12\ 16\ 18\ 20\ 19\ 19 \quad (2)$$

reprezentînd valoarea zecimală a informațiilor din zone.

Pe fișier (bandă sau disc) fiecare caracter are o valoare binară. Dacă considerăm codul ca fiind :

caracter	Valoarea zecimală	Valoarea binară
0	240	11110000
1	241	11110001
2	242	11110010
3	243	11110011
4	244	11110100
5	245	11110101
6	246	11110110
7	247	11110111
8	248	11111000
9	249	11111001

Scriind corespondentul fiecărui caracter din (2), se obține valoarea binară a cheii de sortare ca un număr binar astfel:

KEY : = 111101101111011011110001111100101111000111110110
 11110001111100011110010111100001111000111110011111
 000111110001 (3)

Dacă în (3) se consideră fiecare grup de 8 biți separat, făcind echivalență în zecimal a fiecărui grup, se obține numărul din (2) scris în zecimal.

Formăm un tabel separat cu cheile de sortare (fig. 4.4.2).

70	02	12	10	20	20	00
70	02	15	10	25	11	11
69	04	14	11	40	14	30
69	03	12	16	20	18	20
69	03	12	16	18	19	45
69	03	12	15	25	16	40
67	01	15	15	25	15	30
68	12	25	14	20	14	44
68	12	24	09	10	10	25
66	12	16	18	20	19	19
70	12	15	07	00	08	00
70	12	15	09	00	09	12
70	12	15	12	25	14	00
70	12	15	11	06	15	00
65	04	10	10	03	12	35
66	06	12	08	00	14	40
70	01	01	07	01	08	01
70	02	01	05	05	06	06
65	05	01	10	11	11	44

Fig. 4.4.2. Tabel cu cheile de sortare.

Tabelul din figura 4.4.2, reprezintă un fișier care are înregistrarea de 19 caractere lungime, din care 14 caractere (de la 1—14) formează cheia de sortare a înregistrărilor din tabelul 4.4.1., iar 5 caractere indică numărul înregistrării de proveniență a cheii.

Pentru numărul înregistrării de proveniență a cheii, s-a lăsat un spațiu de 5 caractere, deoarece în tabelul 4.4.1. nu sunt indicate toate înregistrările, ci numai o parte.

Fișierele organizate în acces direct nu au cheia de acces conținută în înregistrări: cheia reprezintă în acest caz numărul înregistrării în fișier. Astfel, dacă fișierul are N înregistrări, atunci înregistrările se consideră numerotate de la $0 \div N - 1$.

De obicei se alege o pistă pe care începe inscrierea fișierului. Dacă se cunoaște numărul pistei (începând numerotarea cu pista zero), de la care începe fișierul, iar numărul de înregistrări pe o pistă este stabilit la R , același pentru toate pistele, se poate calcula pista pe care se află inscrisă înregistrarea nr. n (sau a $n+1$ înregistrare).

Fie p numărul pistei relative pe care se află inscrisă înregistrarea respectivă.

Se calculează :

$$p = \text{INTREG}(n/R) \quad (4)$$

Numărul înregistrării pe pistă (n_p) considerind că numerotarea înregistrărilor se face de la zero la R , este chiar restul împărțirii, adică :

$$n_p = \text{REST}(n/R) \quad (5)$$

În acest caz, se spune că am găsit pista relativă pe care se află înregistrarea, deoarece aceasta a fost calculată ca și cum fișierul ar fi început de pe pista zero. În realitate el începe de pe o pistă oarecare p_0 . În acest caz numărul pistei absolute este :

$$p_A = p_0 + p \quad (6)$$

și reprezintă exact pista pe care se află înregistrarea n .

Calculul numărului cilindrului se poate face plecind de la pista absolută p_A .

Știind că pe fiecare cilindru se află N_p piste, se determină numărul cilindrului absolut făcind împărțirea la numărul de piste de pe cilindru și luând partea întreagă, adică :

$$c_A = \text{INTREG}(p_A/N_p) \quad (7)$$

Restul rezultat reprezintă numărul pistei pe cilindru. Considerind număratoarea de la zero :

$$p_c = \text{REST}(p_A/N_p) \quad (8)$$

unde p_c este numărul pistei pe cilindru.

Vom ilustra cele arătate mai sus printr-un exemplu.

Fie un disc cu 400 de cilindri și 20 de piste pe cilindru.

Fie un fișier care are un număr total de 1500 de înregistrări, cîte 30 de înregistrări pe o pistă. Fișierul se scrie începînd de la pistă $p_0 = 1\ 618$. Se cere să se determine cilindrul relativ și pistă relativă pe cilindru și apoi cilindrul absolut și pistă absolută pe cilindru pe care se află inscrisă cea de a 118-a înregistrare a fișierului (numărul înregistrării este $n = 117$, deoarece numerotarea începe de la zero).

Conform (4) avem :

$$p = \text{INTREG}(117/30) = 3$$

iar, conform (5) rezultă :

$$n_p = \text{REST}(117/30) = 27$$

deci a 28-a înregistrare pe pistă, adică înregistrarea nr. 27.

Se calculează conform (6) :

$$p_A = p_0 + p = 1\ 618 + 3 = 1\ 621.$$

Se poate calcula deci, conform (7) :

$$c_A = \text{INTREG}(1\ 621/20) = 81$$

adică înregistrarea se află pe cilindrul nr. 81, iar conform (8) :

$$p_c = \text{REST}(1\ 621/20) = 1$$

deci înregistrarea se află pe pistă nr. 1 (adică cea de a doua pistă) de pe cilindrul nr. 81 (adică cel de al 82-lea cilindru).

In acest fel determinînd poziția exactă a înregistrării, prin determinarea cilindrului, a pistei și a numărului înregistrării pe pistă, se obține înregistrarea respectivă prin poziționarea capetelor de înregistrare pe cilindrul respectiv (81), apoi se selecționează pistă (1) și apoi se determină prin citire secvențială cea de a 28-a înregistrare de pe pistă (adică înregistrarea nr. 27).

Obișnuit, valoarea cheii (reprezentată printr-un număr întreg care reprezintă poziția înregistrării în fișier), se atribuie unui identificator (variabile) ; fie numele acestuia KEY. Aceasta este cheia cu care se începe căutarea, în cazul nostru KEY : - 117.

Se calculează de către sistemul de operare, automat, adresa înregistrării (adică cilindrul, pistă și a cîte înregistrare este pe pistă) pornind de la valoarea variabilei KEY. După citire, cheia va apărea cu o valoare (conform instrucțiunilor corespunzătoare din program) egală cu valoarea precedentă plus 1.

Se poate face astfel o citire secvențială a înregistrărilor, dacă se dorește, pînă la sfîrșitul fișierului. Dacă se dorește citirea altrei înregistrări, atunci se scrie în KEY numărul acestei înregistrări care va fi automat căutată de către SPAD.

In cazul acestui tip de fișier, cheia nu este obligatoriu să existe în înregistrare. Sistemul de operare poate separa această cheie în fața fiecarei înregistrări, în care caz cheia nu este prelucrată în cadrul înregistrării prin procedurile de prelucrare.

Vom prezenta în continuare un exemplu de sortare (ordonare) a fișierului din fig. 4.4.1. în ordinea crescătoare a valorilor cheilor.

Pentru sortare este necesar să se creeze două fișiere în acces direct cu cheie numerică.

Fie FISI fișierul care trebuie sortat. Fie FISD un fișier în acces direct identic cu FISI. Fie FISKEY un fișier în acces direct format din cheile de sortare avînd pentru fiecare înregistrare un cîmp cu lungimea de 19 caractere împărțit astfel :

— caracterele 1—14 : cheia de sortare formată prin concatenarea cheilor de sortare ;

— caracterele 15—19 reprezentînd numărul înregistrării din FISI care are cheia de sortare din poziția 1—14 a acestei înregistrări din FISKEY

Fie FISS, un fișier secvențial format din aceleași înregistrări ca FISI dar cu înregistrările aranjate în ordinea crescătoare sau descreșcătoare a valorilor binare ale cheilor (fișierul sortat).

Etapile de sortare sunt :

a. Transformarea fișierului secvențial omogen în fișier în acces direct cu cheie numerică, încarcarea fișierului FISKEY și determinarea numărului de înregistrări fizice existente în FISI. FISKEY și FISD au același număr de înregistrări cu FISI, FISD fiind identic cu FISI, diferența constînd doar în metoda de acces.

b. Ordonarea cheilor din fișierul FISKEY, sortarea fișierului FISD în conformitate cu cheile din FISKEY și obținerea fișierului sortat FISS, care eventual poate fi scris peste FISI.

Schimba de sistem este prezentată în fig. 4.4.3., iar schema logica de programare în fig. 4.4.4.

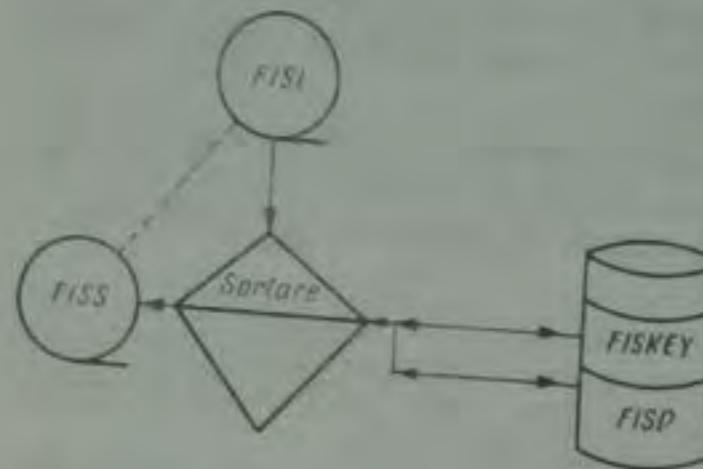


Fig. 4.4.3. Schemă de sistem pentru sortare și interclasare.

Deoarece FISKEY conține cheia de sortare în zona 1–14 și numărul înregistrării de proveniență din FISD în zona 15–19, se face sortarea cheilor în ordine crescătoare sau descrescătoare și pe măsură ce cheile sunt sortate se crează secvențial fișierul FISS.

Fișierul de chei FISKEY nu ne interesează să fie obținut sortat, deoarece se obține FISS care are înregistrările aranjate în ordinea cheilor.

Fie ARIAI o arie de memorie formată din două zone, ZONAI în lungime de 14 caractere (caracterele 1–14 din înregistrarea FISKEY (I) și NRI formată din 5 caractere (caractere 15–19 din înregistrarea FISKEY (I)), în care se citesc înregistrările FISKEY (I) din fișierul FISKEY.

Fie ARIAJ și ARIAIND alte două arii din memorie, cu zonele ZONAJ și ZONAIND, respectiv NRI și NRIND, în care se citesc înregistrările FISKEY (J) și respectiv FISKEY (IND), identice ca structură cu aria denumită ARIAI și zonele ZONAI și NRI. Aceste arii sunt necesare pentru a se înscrie în memorie cheile înregistrărilor între care se fac comparații pentru procesul de ordonare.

Fie ARIAFS o arie de memorie în lungime de 24 de caractere în care se citește o înregistrare din FISI sau FISD.

Se folosesc variabilele intregi I , J , N .

4.4. IDENTIFICAREA ȘI VALIDAREA INREGISTRARILOR

Algoritmul de prelucrare este următorul :

1. START.
2. INITIALIZARE

$N := -1$

$I := N + 1$

3. INTRARE.

ARIAFS := FISI (I)

Se face citirea secvențială a fișierului FISI, pentru a se crea FISKEY și FISD, ambele în acces direct.

4. DECIZIE

FISI s-a terminat ?

4.1. DA. SALT la 7.

4.2. NU. SALT la 5.

5. PRELUCRARE.

Se formează ZONAI (cheia din FISKEY (I)) ca în tabelul din figura 4.4.2.

Se atribuie zonei NRI valoarea I .

Se ciclează variabila N .

$N := N + 1$

6. IEȘIRE.

FISD (I) := ARIAFS.

FISKEY (I) := ARIAI

S-a atribuit înregistrării nr. 1 din fișierul FISD valoarea înregistrării I din FISI. La fel FISKEY.

Salt la 3.

7. INITIALIZARE.

$I := 0$

Se începe sortarea. Numărul de înregistrări din FISKEY și FISD este dat în N .

8. INITIALIZARE.

IND := I

9. INTRARE.

ARIAI := FISKEY (I).

S-a citit o înregistrare din FISKEY.

10. INITIALIZARE.

J := I + 1

ARIAIND := ARIA I

11. DECIZIE.

I = N ?

11.1. DA. SALT la 18.

11.2. NU. SALT la 12.

12. INTRARE.

ARIAJ := FISKEY(J).

S-a citit înregistrarea J din FISKEY (J).

13. DECIZIE.

ZONAJ < ZONAIND ?

Se testează dacă cheia citită din înregistrarea J este mai mică decât cheia existentă în ZONAIND.

13.1. DA. SALT la 14.

13.2. NU. SALT la 15.

14. PRELUCRARE. IND = J

ARIAIND := ARIA J

15. DECIZIE.

J = N ?

15.1. DA. SALT la 17.

15.2. NU. SALT la 16.

16. CICLARE.

J := J + 1

Salt la 12.

17. DECIZIE.

I = IND (?)

17.1. DA. Salt la 19. Nu se face schimbarea din ARIA I în FISKEY (IND), deoarece nu s-a găsit altă înregistrare cu o cheie mai mică decât a înregistrării I.

17.2. NU. Salt la 18.

18. IEȘIRE.

ARIAFS := FISD (NRIND)

Se inscrie înregistrarea FISKEY (I) în locul înregistrării FISKEY (IND), care este ca ordine în locul înregistrării FISKEY (I).

19. INTRARE

ARIAFS := FISD (URIND)

S-a citit din FISD înregistrarea corespunzătoare ca ordine lui FISS(I) conform ordinei cheilor de sortare.

20. IEȘIRE.

FISS(I) := ARIA FS.

S-a inscris în FISS înregistrarea corespunzătoare ca ordine cheii de sortare.

21. CICLARE.

I := I + 1

22. DECIZIE.

:>N ?

Adică se face verificarea terminării sortării.

La blocul 11 s-a verificat dacă I = N, deoarece pentru această valoare nu mai există înregistrarea FISKEY(J). Conform blocului 10

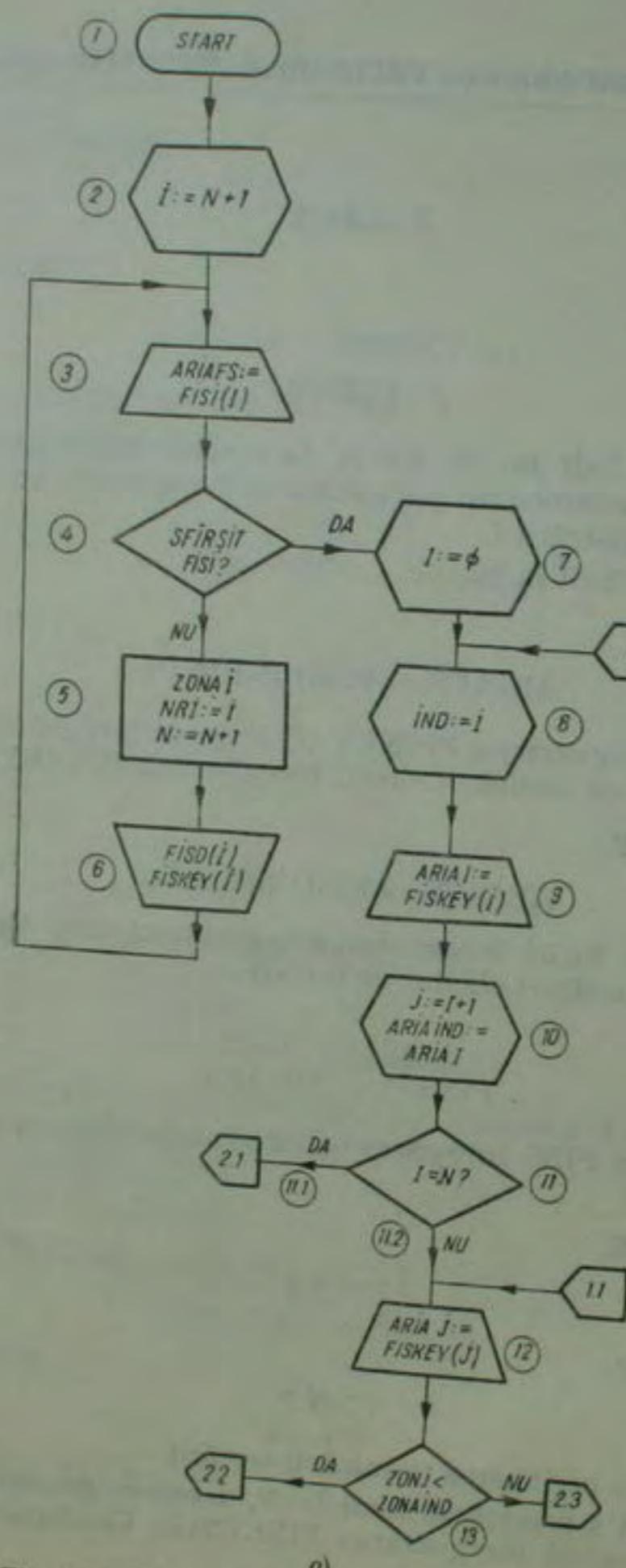


Fig. 4.4.4. Schemă logică de sortare cu tabel de indici.

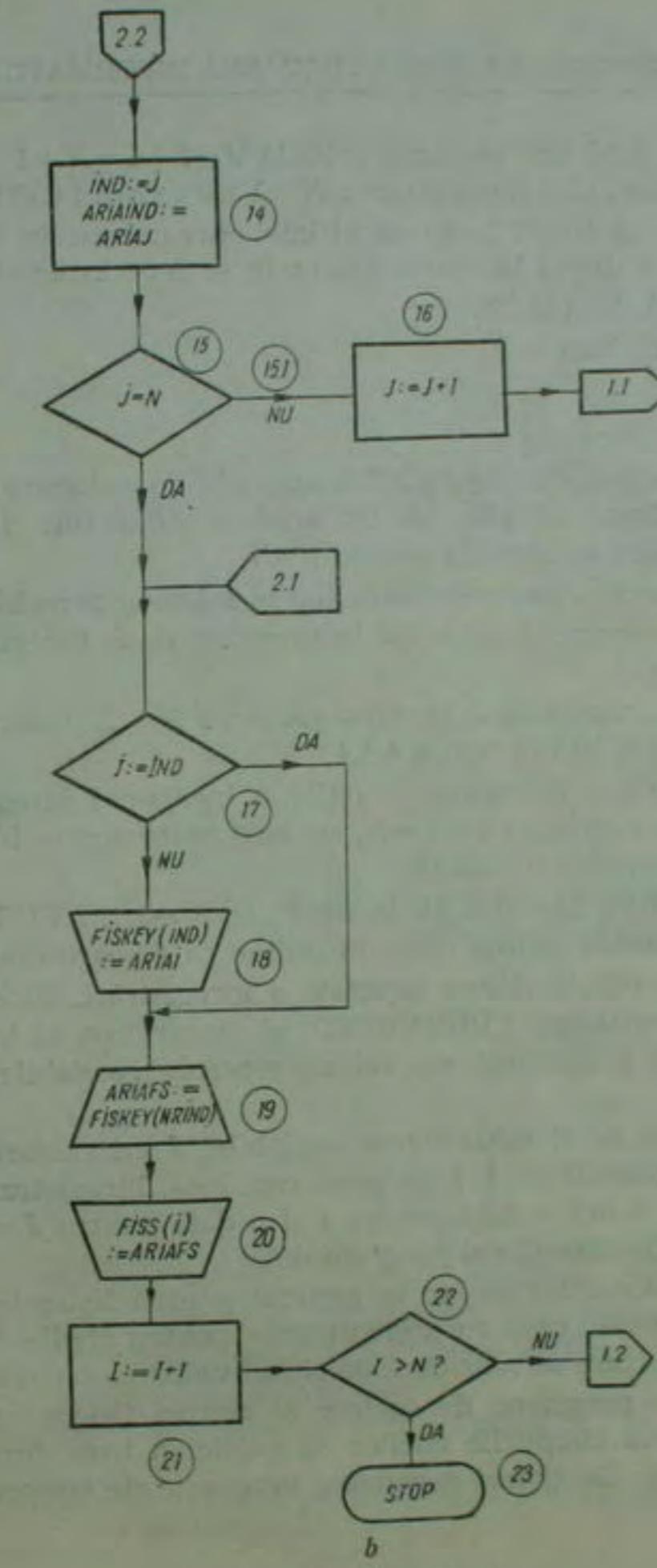


Fig. 4.4.4. Continuare.

variabila J nu mai are valoarea inițială $J=I+1=N+1$, ceea ce ar conduce la eroare, căci înregistrarea $N+1$ nu există în FISKEY.

Trebuie făcută totuși trecerea ultimei înregistrări în FISS(N), de aceea se și trece direct la blocul 18, unde se face înregistrarea.

22.1. DA. Salt la 23.

22.2. NU. Salt la 7.

23. STOP.

Sortarea s-a terminat.

La schema logică din figura 3.3.8. operația de ordonare se termină pentru $I=N$ (blocul nr. 13), iar în schema logică din figura 4.4.4. operația de sortare se termină pentru $I>N$.

Deoarece pentru compararea cheilor se folosesc perechile de valori I și J , sunt necesare să fie citite în memorie două înregistrări diferite din FISKEY.

Pentru $I=N$, variabila J ar avea valoarea $N+1$, deoarece se initializează în blocul 10 din figura 4.4.4.

Pentru a evita o încercare de citire a încă unei înregistrări din FISKEY, care nu există cind $I=N$, se sare peste aceste blocuri încă de la blocul 9, direct la blocul 18.

Prin executarea blocului 18 la pasul $I=N-1$, în FISKEY(N) se poziționează automat ultima cheie în ordine ca valoare binară.

Pentru a nu repeta citirea separată a înregistrării FISKEY(N) și apoi citirea înregistrării FISD(NRIND) și înscrierea ei în FISS(N), ceea ce ar lungi programul, s-a folosit controlul variabilei I pentru valoarea $I=N$.

O altă soluție ar fi inițializarea variabilei J cu valoarea $J=I$ în loc de $J=I+1$ (blocul nr. 1). În acest caz, însă, înregistrarea I ar fi trebuit citită de 2 ori, o dată pentru I și o dată pentru $J=I$, ceea ce mărește timpul de execuție al programului.

Sortările (ordonările) se fac în general pentru fișierele omogene sau parțial omogene, care au aceleași zone pentru cheile de sortare. În procesul de sortare se consideră de obicei valoarea binară a cheilor.

Pot fi făcute programe de sortare și pentru fișiere neomogene, dar cu condiția ca cimpurile cheilor să existe în toate înregistrările din aceste fișiere. De obicei asemenea programe de sortare se utilizează rar.

4.5. Descrierea fișierelor și a prelucrărilor cu ajutorul unui limbaj de proiectare

4.5.1. Clase și grupuri de înregistrări

Pentru a putea fi utilizate în cadrul unor programe de prelucrare automată a datelor, fișierele trebuie definite. Definirea unui fișier (de pildă ca în cazul unui program COBOL), se face în principal descriind înregistrările fizice și înregistrările logice din care este compus. Este necesar să fie indicate următoarele elemente :

- a. numele fișierului ;
- b. tipul fișierului și lungimea maximă a înregistrărilor fizice ;
- c. tipul de acces la fișier ;
- d. tipul operațiilor cu fișierul ;
 - intrare
 - ieșire
 - intrare/ieșire
- e. definirea cheii de acces și a poziției acesteia în înregistrarea logică ;
- f. stabilirea cheilor de identificare ale claselor și grupurilor de înregistrări, formate în funcție de prelucrările ce se efectuează asupra înregistrărilor în procesul de organizare și actualizare a fișierului.

Să detaliem aceste elemente :

- a. Numele fișierului se consideră o combinație de litere mari și cifre, primul caracter fiind literă (de obicei F).

De exemplu : FST reprezintă un nume de fișier, FST1 reprezintă de asemenea un nume de fișier.

Combinăriile de caractere 1F25A sau F-175 nu reprezintă nume de fișiere.

- b. Tipul de fișier utilizat poate fi cu înregistrări fizice de lungime fixă sau înregistrări fizice de lungime variabilă. Este necesar să se indice lungimea maximă a înregistrării fizice, deoarece aria I/E alocață fișierului în memoria internă trebuie să fie cel puțin egală cu lungimea maximă a înregistrării fizice.

- c. În principal, tipul de acces la fișier poate fi :

- secvențial ;
- secvențial indexat cu prelucrare (citire sau scriere) secvențială ;

- secvențial indexat cu prelucrare (citire și scriere) aleatoare;
- aleatoriu.

d. Se definește fișierul dacă este de tip intrare sau de tip ieșire sau de tip intrare/ieșire. Tipul intrare/ieșire poate fi atribuit în general numai fișierelor cu acces aleatoriu.

e. Pentru fișierele cu acces aleatoriu, trebuie să fie definită poziția cheii în înregistrarea logică sub forma unui sir de caractere succesive sau unor siruri de caractere formate din diferite cimpuri ale înregistrării, care se concatenă (se alătură).

f. Înregistrările logice ale unui fișier pot fi grupate după procedurile de prelucrare.

Toate înregistrările logice care se prelucrează după aceleași proceduri de prelucrare formează o clasă de înregistrări.

Un fișier poate avea o singură clasă de înregistrări logice atunci când informația conținută în fiecare înregistrare din clasă se prelucrează după aceleași proceduri. Aceasta este cazul în care se află frecvent fișierele omogene.

Dacă prelucrarea înregistrărilor se face diferit după mai multe proceduri, atunci se definesc mai multe clase (de obicei cazul fișierelor parțial omogene sau neomogene).

Clasele se identifică prin chei (de identificare) existente în anumite cimpuri din înregistrările logice necesare procesului de prelucrare în unitatea centrală.

Identificarea unei clase se poate face pe baza unei structuri arborescente, programul stabilind clasa de înregistrări conform structurii arborescente asociate fișierului, formată din cheile înregistrărilor. Elementele structurii arborescente sunt deci valori din unele cimpuri ale înregistrărilor logice, a căror informație este interpretată, continuarea făcându-se pe o ramură sau alta a arborelui asociat, pînă la găsirea clasei și a înregistrării dorite.

Stabilirea clasei revine la determinarea procedurilor de prelucrare după care înregistrările clasei sunt prelucrate. Operația de stabilire a clasei formează tot o prelucrare, care face parte din procedurile de prelucrare ale clasei respective.

Clasele de înregistrări pot fi împărțite la rîndul lor în grupuri.

Prin grup de înregistrări se înțelege submulțimea înregistrărilor dintr-o clasă care sunt prelucrate împreună conform unui criteriu dat, și numai după terminarea prelucrării acestei submulțimi (eventual o parte a sa) se poate trece la prelucrarea altor înregistrări,

Exemplu: În fișierul din fig. 4.4.1. considerăm că actualizarea înregistrărilor se face pentru grupul de înregistrări pentru care numele inscrise în Z1 încep cu litera A. Presupunem că cele 19 înregistrări alcătuiesc o clasă, adică sunt supuse acelorași prelucrări. Submulțimea înregistrărilor în care numele începe cu litera A formează un grup compus din 3 înregistrări.

Analog, vom avea de pildă grupul înregistrărilor pentru care numele din zona Z₁ începe cu litera M, care este format din 4 înregistrări.

Determinarea grupului se face tot în cadrul structurii arborescente a cheilor înregistrărilor, asociată fișierului; se determină în ordine clasa, grupul și înregistrarea dorită.

4.5.2. Simboluri folosite pentru descrierea fișierelor în cadrul limbajului de proiectare

Se consideră o înregistrare logică reprezentată ca un sir de caractere, începînd de la caracterul nr. 1 pînă la caracterul nr. N.

Se poate asocia primului caracter din înregistrare numărul întreg I, ultimul caracter avînd asociat numărul întreg N+I-1, unde N este numărul de caractere din înregistrare.

Orice cimp din înregistrare poate fi scris ca un sir de caractere succesive, care încep dintr-o anumită poziție, fie aceasta p₁ și se termină în altă poziție, fie aceasta p₂. Cele 2 valori se separă între ele prin semnul ÷. Astfel p₁÷p₂ reprezintă un cimp din înregistrarea care începe de la caracterul cu numărul p₁ și se termină la caracterul cu numărul p₂. *)

Oarie de memorie poate fi de asemenea considerată ca un sir succesiiv de caractere și poate fi reprezentată la fel ca și o înregistrare dintr-un fișier.

Definirea unei înregistrări din fișier se poate face de pildă scriind numele fișierului, urmat de acoladă deschisă, urmată de unul sau mai multe cimpuri descrise cu ajutorul pozițiilor caracterelor separate prin virgulă și urmate de acoladă închisă; ca atare în descriere se prezintă lungimea înregistrării și împărțirea sa în cimpuri. Fie un

*) O constantă alfanumerică se scrie ca un sir de caractere începînd cu și terminînd cu caracterele care apar între " formeză constantă alfanumerică.

fișier FCUCS pentru care se dorește definirea *) unei înregistrări (de lungime 400 caractere) :

$\text{FCUCS}\{1\div 400\} ::= \text{FCUCS}\{1\div 10, 11\div 40, 41\div 45,$
 $46\div 200, 201\div 400\}$

În acest mod au fost definite cimpurile din înregistrare.

Dacă se dorește împărțirea unui cimp în subcimpuri (de pildă cum am văzut în descrierea structurii unui program în COBOL), atunci subcimpurile vor apărea descrise separat. De exemplu :

$\text{FCUCS}\{1\div 100\} ::= \text{FCUCS}\{1\div 10, 11\div 10, 12\div 17,$
 $20\div 25, 30\div 40, 35\div 37, 37\div 40, 41\div 45, 46\div 200,$
 $201\div 400\}$.

În acest caz, cimpul $11\div 40$ a fost împărțit în subcimpurile : $15\div 17$, $20\div 25$, $30\div 40$, iar subcimpul $30\div 40$ a fost împărțit la rândul său în subcimpurile $35\div 37$ și $37\div 40$.

Această descriere a înregistrării se poate face și prin numerotarea cimpurilor și a subcimpurilor, asociind fiecărui cimp sau subcimp un număr natural.

În acest caz putem scrie :

$\text{FCUCS}\{1, 2, 3, 4, 5, 6, 7, 8, 12, 15\} ::= \text{FCUCS}\{1\div 10,$
 $11\div 40, 12\div 17, 20\div 25, 30\div 40, 35\div 37, 37\div 40,$
 $41\div 45, 46\div 200, 201\div 400\}$,

rezultând de pildă că

$\text{FCUCS}\{12\} ::= \text{FCUCS}\{16\div 200\}$.

Dacă numerotarea se face în ordinea numerelor naturale, atunci :

$\text{FCUCS}\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} ::= \text{FCUCS}\{1\div 10,$
 $11\div 10, 15\div 17, 20\div 25, 30\div 40, 35\div 37, 37\div 40,$
 $41\div 45, 46\div 200, 201\div 400\}$.

Dacă ne referim la $\text{FCUCS}\{J\}$, atunci avem $1 \leq J \leq 10$ oricare alte valori pentru J nu sunt corecte.

*) Pentru definiție folosim simbolul „;:=“.

Pentru a asocia unui cimp o valoare (șir de caractere numerice sau alfanumerice sau spațiu) vom scrie semnul : egal. De pildă :

$\text{FCUCS}\{15\div 17\} := /34A\backslash$

sau

$\text{FCUCS}\{15\div 17\} ::= -\text{spațiu}$.

Pentru a asocia unui cimp valoarea dintr-un alt cimp sau valoarea unei expresii, vom folosi simbolul de atribuire. De pildă :

$\text{FCUCS}\{15\div 17\} := \text{FCUCS}\{35\div 37\}$

sau considerind că în $\text{FCUCS}\{35\div 37\}$ avem o valoare numerică :

$\text{FCUCS}\{15\div 17\} := \text{FCUCS}\{35\div 37\} + 2$.

Referirea la un cimp dintr-o înregistrare se poate face și prin descrierea limitelor caracterelor sub formă unor expresii sau prin descrierea variabilei asociate cimpului sub formă unei expresii.

Astfel ne putem referi la $\text{FCUCS}\{a★b-c\}$ unde a , b și c sint variabile întregi sau reale.

Tinând seama de modul de definire a înregistrării, pentru exemplul considerat mai sus, este valabilă condiția

$$1 \leq a★b-c \leq 10.$$

De exemplu $\text{FCUCS}\{14\div 18, 19+a★17\div 35+b★17/14-8★c, 144\div 600\}$ reprezintă 3 cimpuri dintr-o înregistrare sau dintr-o arie de memorie care se cheamă FCUCS, repartizate astfel :

— cimpul nr. 1 începe de la caracterul nr. 14 și se termină în caracterul nr. 18 ;

— cimpul nr. 2 începe cu caracterul nr. $19+a★17$ și se termină cu caracterul nr. $35+b★17/14-8★c$;

— cimpul nr. 3 începe cu caracterul nr. 144 și se termină cu caracterul nr. 600.

Prin urmare, poziția primului sau ultimului caracter din cimp poate fi exprimată printr-o constantă sau printr-o expresie care poate avea valoare întreagă sau reală (pozitivă).

Dacă expresia are valoare reală, se ia în considerare valoarea întreagă a expresiei obținute din valoarea reală la care se negligează partea zecimală.

În acest caz trebuie respectate următoarele reguli :

— numărul care reprezintă poziția caracterului de început a cimpului (p_1) să fie mai mic sau egal cu numărul care reprezintă poziția caracterului de sfîrșit al cimpului (p_2) ;

— numărul care reprezintă poziția unui caracter de început să nu fie mai mic decit I (indicele primului caracter din înregistrare) ;

— numărul care reprezintă poziția unui caracter de sfîrșit al unui cîmp, să nu fie mai mare decit $I+N-1$ (unde N este numărul caracterelor din înregistrare). Deci :

$$I \leq p_1 \leq p_2 \leq I + N - 1.$$

Lungimea unui cîmp este obținută prin formula :

$$L = p_2 - p_1 + 1$$

și se exprimă în număr de caractere.

Pentru a descrie clasa din care face parte o înregistrare vom descrie poziția cheii clasei de înregistrări între paranteze drepte între care se scriu cîmpurile care formează cheia (de identificare), cu ajutorul pozițiilor caracterelor sau a numerelor naturale asociate ca indici.

Să presupunem că FCUCS are 4 clase de înregistrări, pe care le definim cu ajutorul pozițiilor caracterelor care formează cîmpurile cheilor (de identificare) claselor :

$$\text{FCUCS } \{[1]\} ::= \text{FCUCS } \{[01 \div 02]\}$$

$$\text{FCUCS } \{[2]\} ::= \text{FCUCS } \{[01 \div 02, 08 \div 12]\}$$

$$\text{FCUCS } \{[3]\} ::= \text{FCUCS } \{[01 \div 02, 08 \div 12]\}$$

$$\text{FCUCS } \{[4]\} ::= \text{FCUCS } \{[01 \div 02, 14 \div 16]\}$$

Pentru a putea utiliza clasele, mai este necesar să se definească valorile cheilor sau intervalele în care acestea iau valori :

Pentru clasa nr. 1 :

$$\text{FCUCS } \{01 \div 02\} = '05'$$

Pentru clasa nr. 2 :

$$\text{FCUCS } \{01 \div 02\} = '06' \text{ și}$$

$$'44221' \leq \text{FCUCS } \{08 \div 12\} \leq '99999'$$

Pentru clasa nr. 3 :

$$\text{FCUCS } \{01 \div 02\} = '07'$$

$$'0001' \leq \text{FCUCS } \{08 \div 12\} \leq '44220'$$

și

Pentru clasa nr. 4 :

$$'08' \leq \text{FCUCS } \{01 \div 02\} \leq '12'$$

și

$$'AAB' \leq \text{FCUCS } \{14 \div 16\} \leq 'TVY'$$

In cazul în care ne referim în general la o înregistrare din FCUCS atunci se poate scrie FCUCS $\{1 \div 100\}$.

Dacă ne referim la o clasă din FCUCS se poate scrie FCUCS $\{[1]\}$ unde $1 \leq I \leq 4$.

După cum am văzut, clasele pot fi împărțite în grupuri de înregistrări, în cadrul clasei de înregistrări prelucrarea făcindu-se după aceleași proceduri pentru toate grupurile din clasă. Cheia pentru grupuri, analog ca pentru clase, se descrie între paranteze rotunde, care urmează după parantezele drepte de descriere a clasei.

Fie de exemplu clasa nr. 1, care se descompune în 3 grupuri astfel :

$$\text{FCUCS } \{[1], (1)\} ::= \text{FCUCS } \{[1], (45 \div 47)\}$$

$$\text{FCUCS } \{[1], (2)\} ::= \text{FCUCS } \{[1], (45 \div 47, 48 \div 50)\}$$

$$\text{FCUCS } \{[1], (3)\} ::= \text{FCUCS } \{[1], (45 \div 47, 48 \div 50, 51 \div 54)\}$$

Pentru prelucrare este necesar de asemenea (ca și pentru clase) să se definească valorile sau intervalele de valori ale cheilor grupurilor.

Pentru grupul nr. 1 :

$$\text{FCUCS } \{[1], (1), 45 \div 47\} = '500'$$

Pentru grupul nr. 2 :

$$\text{FCUCS } \{[1], (2), 45 \div 47\} = '600'$$

și

$$'327' \leq \text{FCUCS } \{[1], (2), 48 \div 50\} \leq '400'$$

Pentru grupul nr. 3 :

$$\text{FCUCS } \{[1], (3), 45 \div 47\} = '600'$$

și

$$\text{FCUCS } \{[1], (3), 48 \div 50\} = '999'$$

și

$$'012B' \leq \text{FCUCS } \{[1], (3), 54 \div 54\} \leq '175A'$$

Din exemplele arătate, apare structura arborescentă (asociată fișierului) a cheilor înregistrărilor pentru clase, grupuri și înregistrări. Toate clasele au un cimp comun FCUCS {01÷02}. De asemenea toate grupurile au un cimp comun FCUCS {45÷47}.

Se poate stabili cărei clase și cărui grup aparține o înregistrare din FCUCS astfel :

— se verifică valoarea cimpului FCUCS {01÷02}. În funcție de această valoare se stabilește clasa. Dacă FCUCS {01÷02} = /05\ înregistrarea aparține clasei nr. 1 ;

— se verifică apoi valoarea cimpului FCUCS {45÷47}. Dacă această valoare este /500\, înregistrarea aparține grupului nr. 1. Dacă valoarea este /600\ înregistrarea poate aparține grupului nr. 2 sau grupului nr. 3.

Dacă /327\ ≤ FCUCS {48÷50} ≤ /400\, atunci înregistrarea face parte din grupul nr. 2.

Dacă FCUCS {48÷50} = /999\, se verifică și dacă /012B\ ≤ FCUCS {51÷54} ≤ /175A\.

Dacă ambele condiții sunt indeplinite, atunci înregistrarea face parte din grupul nr. 3.

În caz că nu sunt satisfăcute aceste condiții, înregistrarea nu face parte din nici un grup și ca atare nu se prelucrează sau se tipărește un mesaj corespunzător.

Considerațiile de mai sus se referă la înregistrări aflate în memoria internă a calculatorului și care sunt supuse prelucrării, arborele asociat permățind un acces rapid la înregistrarea dorită.

Ținând seama de faptul că înregistrările sunt aduse în memoria internă de pe suportul de informație, pentru fișierele cu acces aleatoriu este necesar să se definească și cheia de acces la fișier. Cheia de acces este formată dintr-un cimp sau mai multe cimpuri și se scrie între paranteze ascuțite.

De exemplu :

Cheia de acces ::= FCUCS {<01÷02, 04÷07, 03÷03>} definiște poziția cheii de acces la fișierul FCUCS.

În timpul prelucrării, cheia de acces este dată prin valoare pentru a se ști care anume înregistrare se dorește. De exemplu :

FCUCS {</05\||/1452\||/A\|}

reprezintă înregistrarea din fișierul FCUCS (și nu din aria de memorie) cu cheia 051452A ; care trebuie citită în memoria internă a calculatorului. Înregistrarea poate fi găsită în FCUCS sau nu. Valoarea

cheii de acces se poate exprima și cu ajutorul (valorilor) unui cimp sau a mai multor cimpuri din înregistrarea existentă deja în memorie. De exemplu :

FCUCS { <FCUCS {01÷02} ||

|| FCUCS {42÷45} || FCUCS {50÷50}> }

Simbolurile introduse mai sus, la care se pot adăuga diferenți operatori logici (\exists , \vee , \wedge , \Rightarrow , etc.), text și alte elemente convenabile constituie un limbaj de proiectare care permite formalizarea unor prelucrări mai comod decit limbajul schemelor logice ; avantajele acestui limbaj și proiectarea se manifestă în special în descrierea tranzacțiilor și actualizărilor fișierelor.

4.5.3. Exemple de descriere a unui fișier și a unor prelucrări

Să ilustrăm cele prezentate mai sus pe următorul exemplu :

Fie FST un fișier indexat secvențial. Definim structura înregistrărilor fișierului astfel :

FST ::=

FST {1-1} ::=

/\

/N\

FST {<02-08>} ::=

fișier indexat secvențial, înregistrarea logică are de 40 caractere, înregistrarea fizică are lungime de $40 \times 10 = 400$ de caractere, factorul de blocare = 10.

cimp de un caracter care indică starea înregistrării :

înseamnă că înregistrarea va fi trecută în fișierul de istoric.

înregistrarea nu se trece în fișierul de istoric.

cimp reprezentând codul mașinii și secvența înregistrării, fiind în același timp și cheia de acces în fișier. Se imparte în subcimpuri astfel :

FST {02÷04} ::= codul mașinii.

FST {05÷08} ::= secvența înregistrării.

FST {09÷18} := data, ora și minutul intrării mașinii în staționare. Se împarte în subcimpuri astfel :

FST {09÷10} := anul.

FST {11÷12} := luna.

FST {13÷14} := ziua.

FST {15÷16} := ora.

FST {17÷18} := minutul.

Acest cimp este numeric (conține, numai cifre).

FST {19÷28} := data, ora și minutul ieșirii mașinii din staționare.

Acest cimp este numeric.

FST {29÷32} := codul cauzei intrării mașinii în staționare. Acest cimp este alfanumeric.

FST {33÷40} := spațiu neutilizat deocamdată, rezervat pentru utilizarea ulterioară.

FST {33÷36} := valoarea din zona „secvența înregistrării“ din ultima înregistrare pentru o mașină dată numai pentru înregistrările care au FST {05÷08} = „9999“.

După ce structura înregistrărilor fișierului a fost stabilită, se face descrierea prelucrărilor (organizare, generare, actualizare, validare, prelucrări) repartiția fișierului și corespunzător a claselor și grupurilor de înregistrări care rezultă.

Pentru o mașină pot fi raportate mai multe intrări și ieșiri din staționare, adică vor exista mai multe înregistrări care în zona FST {02÷04} vor conține aceeași valoare numerică (codul mașinii). Ultima înregistrare pentru o mașină poate indica numai intrare în staționare în care caz FST {09÷18} ≠ spațiu și conține numai cifre, iar FST {19÷32} = spațiu. (Codul cauzei se trece odată cu raportarea de ieșire din staționare cind se cunoaște motivul staționării).

Dacă înregistrarea indică intrare și ieșire din staționare, atunci FST {09÷32} ≠ spațiu și conține numai cifre, FST {19÷32} ≠ spațiu și conține de asemenea numai cifre.

Înscrierea înregistrărilor care raportează intrările și ieșirile din staționare pentru mașini se face astfel :

— prima înregistrare inscrisă pentru o mașină în fișier are în cimpul FST {02÷04} codul mașinii, iar în cimpul FST {05÷08}, zona „secvența înregistrării“, valoarea egală cu „0000“. Următoarea inscriere a unei înregistrări pentru aceeași mașină se execută numai dacă FST {1÷32} ≠ spațiu (adică mașina a fost raportată ca ieșită din staționare și ultima înregistrare pentru această mașină este completă).

Dacă zona FST {19÷32} a ultimei înregistrări este egală cu spațiu, mașina nu a fost raportată ca ieșită din staționare, iar în zonele FST {19÷28} și FST {29÷32} trebuie să se inscrie data ieșirii din staționare și codul cauzei fără să se adauge o nouă înregistrare.

După ce a fost inscrisă în fișier înregistrarea avind zona „secvența înregistrării“ (FST {05÷08}), valoarea „0000“ (deci prima înregistrare pentru mașina dată) se adaugă și o înregistrare având în zona „secvența înregistrării“ (FST {05÷08}) valoarea „9999“ care apare o singură dată pentru o mașină dată. Această înregistrare are aceeași structură cu celealte înregistrări din FST, dar în cimpul FST {33÷36} se inscrie valoarea zonei „secvența înregistrării“ din ultima înregistrare pentru mașina dată (această valoare este în zona FST {05÷08} din ultima înregistrare inscrisă pentru o mașină și este mai mare sau egală cu „0000“).

Fișierul se împarte astfel în două clase de înregistrări :

Clasa nr. 1 :

FST {[1]} := FST {[05÷08]} pentru valorile „0000“ ≤ FST {05÷08} ≤ „9995“.

Clasa nr. 2 :

FST {[1]} := FST {35÷01} pentru valoarea FST {05÷03} = „9999“.

Cele 2 clase definite mai sus se împart în grupuri astfel :

FST {[1], (1)} := FST {[05÷03], (02÷04)}; ca atare grupul FST {[1], (1)} conține toate înregistrările corespunzătoare unei mașini, având „codul mașinii“ egal cu 1.

A doua clasă de înregistrări se împarte în grupuri în același mod :

FST {[2], (1)} := FST {[05÷08], (02÷04)}.

În particular, fiecare grup din clasa nr. 2 este format dintr-o singură înregistrare, deoarece pentru fiecare mașină se asociază o singură înregistrare din clasa nr. 2, având asociat codul mașinii respective în zona FST {02÷04}.

Actualizarea fișierului

Să presupunem că în aria de memorie ZONA {1÷27} se găsește o tranzacție cu care se actualizează fișierul FST.

În tranzacție se găsesc înregistrate :

ZONA {1÷3} := codul mașinii.

ZONA {4÷13} := data, ora și minutul intrării în staționare.

ZONA {14÷23} := data, ora și minutul ieșirii din staționare.

ZONA {24÷27} := codul cauzei de întrerupere.

Fie KEY {1÷7} o altă arie de memorie.

Se formează cheia de acces la fișierul FST :

KEY {1÷7} := FST {< ZONA {1÷3} || '9999' >}

Sunt posibile două situații :

— *Inregistrarea este găsită în FST*

a. Se verifică dacă FST {[2], (I), 19÷32} = spațiu. Dacă da, atunci tranzacția trebuie să raporteze ieșirea din staționare pentru mașină. Dacă nu, se continuă de la punctul b.

Secvența ultimei înregistrări se află înscrisă în FST {[2], (I), 33÷36}. Se formează cheia de acces KEY {1÷7} := FST {< ZONA {1÷3} || FST {[2], (I), 33÷36} >}. Se găsește înregistrarea cu această cheie în FST.

Se face actualizarea :

FST {[1], (I), 19÷32} := ZONA {14÷27}.

Se actualizează FST {[2], (I), } astfel :

FST {[2], (I), 19÷32} := ZONA {14÷27}.

Valoarea zonei „secvență înregistrării” în FST {[1], (I)} răminind aceeași nu se actualizează FST {[2], (I), 33÷36}.

Se inscriu ambele înregistrări în FST.

b. Dacă FST {[2], (I), 19÷32} ≠ spațiu, rezultă că trebuie să se inscrie o nouă înregistrare în FST {[1], (I)} deoarece ultima înregistrare înscrisă are raportată și ieșirea din staționare.

Se actualizează FST {[2], (I), 09÷32} := ZONA {4÷27} și FST {[2], (I), 33÷36} := FST {[2], (I), 33÷36} + 1 (adică se scrie în înregistrarea având în zona „secvență înregistrării” valoarea '9999' în tranzacția și secvența înregistrării următoare).

Se inscrie înregistrarea FST {[2], (I)} în FST. (Cheia nu mai trebuie formată, deoarece nu se face o căutare ci o inscriere peste înregistrarea existentă, care are cheia în cimpul 02÷08).

Se formează nouă înregistrare în clasa nr. 1 :

FST {[1], (I), 1÷1} = 'N'

FST {[1], (I), 02÷04} := ZONA {1÷3}

FST {[1], (I), 05÷08} := FST {[2], (I), 33÷36}

FST {[1], (I), 09÷32} := ZONA {4÷27}.

Se inscrie această înregistrare în FST.

— *Inregistrarea nu este găsită în FST*

Se raportează prima dată mașina, ca intrată în staționare.

Se formează înregistrarea având în „secvență înregistrării” valoarea '0000' din FST {[1], (I)} și înregistrarea având în „secvență înregistrării” valoarea '0000' din FST {[2], (I)}.

Inregistrarea FST {[2], (I)} se formează astfel :

FST {[2], (I), 1÷1} = 'N'

FST {[2], (I), 0÷04} := ZONA {1÷3}

FST {[2], (I), 05÷08} = '9999'

FST {[2], (I), 09÷32} := ZONA {4÷7}

FST {[2], (I), 33÷36} = '0000'

Se inscrie înregistrarea în FST.

Inregistrarea FST {[1], (I)} se formează astfel :

FST {[1], (I), 1÷4} := FST {[2], (I), 1÷4}

FST {[1], (I), 05÷08} = '0000'

FST {[1], (I), 09÷32} := FST {[2], (I), 09÷32}

FST {[1], (I), 33÷40} = spațiu.

Se înregistrează în această înregistrare în FST.

La schimbarea valorii lui I se vor efectua aceleasi prelucrări asupra înregistrărilor din noul grup s.a.m.d.

Limbajul de proiectare propus, utilizat împreună cu schemele logice de programare și de sistem, permite o descriere mai exactă a fișierelor și a prelucrărilor acestora deci permite elaborarea proiectului unei aplicații în condiții mai apropiate de o formalizare care să evite ambiguitățile și lipsurile care apar adesea atunci cind se folosește pentru descriere în măsură mare text. Utilizarea elementelor limbajului de proiectare permite analiza critică a proiectului și înțelegerea acestuia și de către alte persoane care nu au participat la proiect, înainte de faza de programare, fază în care prelucrările sunt descrise exact în limbajele de programare.

5. Descrierea SIPAD (Sisteme Informaționale cu Prelucrare Automată a Datelor)

5.1. Tipuri de aplicații ale SPAD în întreprinderi ; SIPAD integrat și SIC

Sistemul informațional din întreprinderi deservește două categorii importante de activități care necesită prelucrări de informații :

a. Planificarea activității întreprinderii, care constă în stabilirea obiectivelor pe care întreprinderea trebuie să le realizeze și care se elaborează pe termen lung și pe termen scurt.

b. Urmărirea realizării obiectivelor planificate.

Problemele principale de planificare ce se pot rezolva pe SPAD sunt :

- c — prospectarea pieței ;
- d — planificarea livrărilor ;
- e — planificarea producției pe termen lung și pe termen scurt ;
- f — planificarea aprovizionării și stabilirea necesarului brut și net de materii prime și materiale ;
- g — planificarea forței de muncă și a utilizării timpului de lucru ;
- h — planificarea cheltuielilor, a prețului de cost și a beneficiilor ;
- i — planificarea investițiilor.

Problemele de urmărire aferente acestor probleme de planificare sunt în principal următoarele :

- j — evidențe și prelucrări statistice ;
- k — urmărirea desfacerii ;
- l — urmărirea producției ;
- m — urmărirea și controlul stocului de materii prime și materiale și a stocului de produse finite ;

- n — urmărirea utilizării forței de muncă și a folosirii timpului de lucru;
- o — urmărirea cheltuielilor, a realizării prețului de cost (planificat), a beneficiilor, precum și calculul salariilor personalului care lucrează în regie, a personalului care lucrează în acord, a personalului tehnicoadministrativ;
- p — urmărirea realizării și desfășurării investițiilor.

Planul întreprinderii este dat de către forul tutelar (centrală sau minister) sub forma unor indicatori globali privind valoarea producției globale, valoarea producției marfă vândută și încasată, planul valutar în lei valută, prețul de cost mediu pe întreprindere, cheltuielile la mia de lei, beneficiul etc., pe care întreprinderea trebuie să-i îndeplinească în cursul desfășurării activității pe o perioadă în general de un an.

De asemenea o parte din planul întreprinderii (uneori în întregime) poate fi chiar nominalizat de către forul tutelar.

Acela parte din plan care nu a fost nominalizată de forul tutelar, întreprinderea trebuie să o acopere prin contracte proprii cu beneficiarii. Contractele trebuie astfel încheiate, încât întreprinderea să realizeze indicatorii globali de plan primiți de la forul tutelar.

Stabilirea obiectivelor pe care urmează să le realizeze activitatea unei întreprinderi, trebuie să se bazeze pe cercetarea cererii consumatorilor, indiferent dacă aceștia sunt întreprinderi care vor consuma produsele în procesul de producție sau consumatori individuali.

Rезултатите привид *последствия и потребление на рынке* поти се отъстъпват по SPAD. Се пот отъстъпват о серия от констатации статистични качествени, които поти се отъстъпват за *десфашурата* на *услуги*.

SPAD permite o prelucrare exactă și rapidă, putind calcula o serie de indicatori globali în diferite variante care pot fi utili întreprinderii în alegerea variantei de planificare. Ca atare întreprinderea își organizează activitatea economică și de producție, pe baza planului și în conformitate cu contractele pe care le încheie cu beneficiarii; în aceste contracte se stipulează în general termenul de livrare, cantitatea și calitatea marfii, condiții suplimentare pe care trebuie să le îndeplinească marfa la cererea beneficiarului etc.

In general există anumite norme sau regulamente departamentale privind livrarea produselor contractate.

In aceste norme sau regulamente se stabilesc cantitățile lunare sau pe alte perioade de timp, pe care întreprinderca trebuie să le

livreze, sortimentul lunar sau trimestrial, calitatea mărfii, penalizările care se pot da în cazul în care întreprinderea nu-și onorează obligațiile pe care și le-a asumat prin contract etc.

Rezolvarea unor probleme de tipul c poate permite întreprinderii să se orienteze mai bine și atunci cind are loc acțiunea de contractare.

Cunoașterea din timp a beneficiarilor siguri și a beneficiarilor posibili, a tendinței pieței sau a tendinței tehnologice pe o anumită piață de desfacere și a cererii beneficiarilor, poate da întreprinderii o orientare importantă în stabilirea unor produse care vor putea fi puse în fabricație în viitor și care să vină în întâmpinarea cererii beneficiarilor. Cunoscind dorințele beneficiarilor, întreprinderea poate lua din timp măsuri tehnice și organizatorice de desfășurare a procesului de producție pe următoarea perioadă de contractare.

Incheierea contractelor este o operație destul de dificilă. În general la incheierea contractelor, întreprinderea trebuie să știe dacă poate onora contractele la termenele și în condițiile cerute de beneficiar și stabilitate de lege.

Avinde posibilitatea simulării planificării producției pe SPAD, se poate determina dacă cererile beneficiarilor pot fi satisfăcute și cum pot fi respectați indicatorii globali din planul primit de la forul tutelar.

Planificarea producției se face pe termen lung și pe termen scurt.

In planificarea producției se include :

- planificarea încărcării capacitaților pe termen nedefinit;
- planificarea încărcării capacitaților pe termen finit;
- programarea operativă a producției;
- lansarea în producție.

Planificarea încărcării capacitaților pe termen nedefinit se referă la posibilitatea executării unui anumit plan de producție, în anumite condiții într-un timp nelimitat.

In acest caz va rezulta un plan de încărcare a utilajelor pe o perioadă de timp oarecare, ce nu a fost stabilită inițial.

Se poate întâmpla ca această perioadă să fie prea mare și să depășească termenele la care întreprinderea s-a obligat să-și onoreze contractele, în care caz se pot stabili anumite măsuri organizatorice și tehnologice și se poate încerca efectuarea aceluiasi plan modificat în intervalul de timp dorit.

Se pot face astfel multe variante.

Se poate pune însă problema și în alt mod :

Ce plan poate fi realizat într-un anumit interval de timp ?

Cunoscindu-se planul de producție, se poate face o simulare a executării planului de producție și se pot stabili eventual abaterile de la termenele de livrare sau cantitățile care nu se pot produce pe o perioadă de timp. În acest fel, întreprinderea poate avea o orientare asupra respectării termenelor și încărcării capacitaților de producție pe perioada de timp pentru care face contractarea.

Programarea operativă a producției și lansarea în producție sunt probleme de planificare a încărcării utilajelor pe termen scurt. Pentru rezolvarea acestor probleme pe SPAD se folosesc în afară de metode de simulare și modele de optimizare care permit elaborarea de liste cu operațiile și termenele la care trebuie executate pe perioade relativ scurte (săptămână, decadă, lună).

În caz că datorită unor anomalii programarea făcută se depărțează de realitate, se face prelucrarea pentru noile condiții pentru o nouă perioadă.

Pentru *încărcarea utilajelor* se pot considera diferite criterii de optimizare care conduc la rezolvarea unor probleme de afectare în vederea utilizării cît mai raționale a capacitaților existente. Se pot elabora modele în care pe lîngă problema utilizării cît mai eficiente a capacitaților de producție să se țină seama de asemenea de o utilizare cît mai eficientă și a celorlalte resurse în cadrul restricțiilor impuse de respectarea termenelor, aprovizionare etc.

După ce s-au stabilit contractele cu beneficiarii și s-a făcut încărcarea utilajelor stabilindu-se încărcarea fiecărei mașini în parte, precum și datele la care trebuie să se facă lansarea în producție, pentru ca produsele respective să fie livrate la anumite termene, se poate trece la o *planificare a aprovizionării*.

Se face un calcul al necesarului de aprovizionat brut, care nu ține seama de materiale prime și materialele existente în magazie.

Se face totodată și un calcul al necesarului de aprovizionat net, în care se ține seama de stocul existent în magazia de materii prime și materiale la data la care trebuie să se facă aprovizionarea.

S-a închis astfel un cîtu de planificare foarte important: *prospectarea pieței, planificarea livrărilor, planificarea producției, planificarea aprovizionării*. Dacă se preconizează un anumit plan de livrare și dacă din simularea planului de producție rezultă că acesta nu poate fi realizat la termen, întreprinderea poate decide dacă va lăsa măsuri de extindere a capacitații, dacă va reduce planul în mod corespunzător sau va lăsa alte măsuri organizatorice care să-i permită realizarea planului. La fel se poate spune și despre simularea aprob-

5.1. TIPURI DE APLICARE ALE SIPAD ; SIPAD INTEGRAT

vizionării, care poate conduce la concluzia că deși capacitatea utilajelor permite executarea planului de producție, aprovizionarea cu materiale creează asemenea probleme încît planul de livrare nu poate fi indeplinit, ceea ce implică din nou luarea unor decizii.

Aceste aplicații pot fi executate fie în ordinea indicată în text de la *a* și pînă la *f*, sau pot fi tratate și în ordinea *c, d, f, e*, adică planificarea aprovizionării să se facă imediat după planificarea livrărilor. Se recomandă această ordine atunci cînd există capacitate de producție suficientă, dar nu există certitudinea procurării la timp a materiei prime și a materialelor necesare procesului de producție.

În general aceste 4 probleme se rezolvă împreună în cadrul unui sistem integrat de prelucrare automată a datelor, ca atare utilizindu-se o bază comună de date și ținind seama de interdependența dintre aceste activități.

Aplicația „*planificarea încărcării capacitaților pe termen finit (perioada de plan)*“ se face ținind seama și de contractele cu furnizorii de materii prime și materiale, dind o indicație asupra posibilităților de executare a planului sau de schimbare a unor termene de livrare în cazul în care capacitațile de producție nu sunt suficiente pe perioada de plan.

Această aplicație este foarte utilă pentru contractele vest, care au caracter mai pronunțat aleatoriu, spre deosebire de contractele est sau piață internă care se fac pentru o perioadă bine definită.

Aplicația „*programarea operativă a producției*“ se efectuează condiționat de existența în stoc a materialelor și pieselor necesare, simularea pentru această aplicație făcîndu-se în general numai în raport cu capacitațile de producție.

De asemenei implementarea acestei aplicații se face numai după ce este implementată aplicația *controlul producției*, prin care se actualizează practic în mod continuu situația încărcării capacitaților de producție, situație necesară pentru planificarea pe termen scurt.

În continuare în cadrul sistemului informațional integrat pot fi tratate și problemele de precalcul de la punctele *g, h, i*, realizîndu-se legăturile cu aplicațiile precedente.

Pentru a trata corect problemele de la punctul *g* este necesar să fie rezolvată problema planificării producției. Numai în acest mod se poate face o planificare judicioasă a forței de muncă și se poate studia folosirea timpului de lucru.

Există și în acest caz posibilitatea simulării și a creării legăturii de control inverse, în sistemul informațional. Dacă întreprinderea are la dispoziție numai o anumită forță de muncă, poate rezulta că planul de producție propus este prea mare.

În acest caz, decizia trebuie lăsată de către conducerea întreprinderii, dacă micșorează planul de producție în mod corespunzător sau mărește forța de muncă.

În acest domeniu SPAD poate avea un rol foarte important. În afara faptului că permite simularea și legătura de control inversă, permite obținerea unei informații corecte într-un timp relativ scurt.

Tratarea corectă a problemei de la punctul *h* necesită rezolvarea prealabilă a problemelor de la punctele *e* și *g*.

Este necesar să se cunoască planul de producție și forța de muncă aferentă pentru a putea calcula corect cheltuielile de producție, profitul de cost și beneficiile (precacul).

Ultimele probleme de planificare propuse sunt cele de investiții.

În vederea integrării acestor activități în cadrul unui sistem informațional, nu este vorba de tratarea problemelor de investiții în general, ci în primul rând este vorba de stabilirea faptului dacă este necesar să se facă investiții sau nu.

Dacă da, investițiile vor rezulta atât din studiul pieței și al contractelor încheiate cu furnizorii, cât și din simularea planului de producție. Rezultă totodată și cât de mari pot fi aceste investiții.

Activitatea de realizare a unei investiții (construirea obiectivelor investiției) este o problemă care poate fi abordată la fel ca orice activitate a unei întreprinderi, în relație cu celelalte activități de planificare și de urmărire.

Din analiza problemelor de planificare care pot fi rezolvate pe SPAD, în întreprinderi, rezultă că unele dintre ele pot fi rezolvate independent. Pot fi rezolvate astfel numai problemele de prospetime și problemele de contractare (livrări). Problemele de producție nu pot fi rezolvate în afara celor de contractare, deoarece acestea stau de fapt la baza planificării producției. Nici celelalte probleme enumerate la punctele *f*, *g*, *h* și *i* nu pot fi rezolvate fără a se realiza integrarea cu planificarea producției.

De asemenea în vederea conducerii întreprinderii folosind echipamentele automatizate este necesar ca activitățile de planificare să fie integrate cu cele de urmărire care au o pondere foarte importantă în cadrul sistemului informațional. Aceste probleme sunt în legătură

5.1. TIPURI DE APLICARE ALE SIPAD ; SIPAD INTEGRAT

directă cu problemele de planificare și asigură legătura inversă (reacția), în cadrul sistemului de conducere cu mijloace automatizate.

Planul de livrări odată stabilit, trebuie adus la indeplinire. Urmărirea desfacerii include legătura planificare-livrare la beneficiar (desfacere). Produsul respectiv se consideră fabricat, în momentul în care a fost expediat la beneficiar. Data expediției reprezintă de fapt data la care contractul a fost onorat. Urmărirea desfacerii poate da un răspuns asupra întrebării dacă întreprinderea și-a onorat obligațiile față de beneficiar sau nu.

Urmărirea producției este o parte foarte importantă în toate activitățile de urmărire intrucât permite întreprinderii să facă o analiză cantitativă și calitativă a activității (postcalcul).

În urmărirea producției se include urmărirea utilajelor, urmărirea rebuturilor, incărcările și realizările pe posturi de lucru, pe ateliere, pe întreprindere etc.

Urmărirea realizărilor permite calculul cheltuielilor făcute, a profitului de cost realizat, a beneficiilor realizate, permite calculul salariilor în acord cu sint de cele mai multe ori legate obligatoriu de realizarea urmăririi producției. Trecerea pe SPAD a activităților de postcalcul de la punctele *n*, *o*, și *p*, sint legate de rezolvarea priorității pe SPAD a problemei urmăririi producției.

Subliniem importanța implementării aplicației de urmărire a producției înainte de aceste operații, deoarece aceasta efectuează actualizarea fișierelor, pe care le utilizează aplicațiile menționate.

Problemele arătătoare la punctul *m* de urmărire și controlul stocurilor se referă atât la gestiunea stocurilor, cât și la calculul și urmărirea stocurilor de siguranță și a stocurilor supranormative.

Stocul normat poate fi uneori mult mai mare decât nevoile reale ale întreprinderii.

Stabilirea stocului de siguranță conform nevoilor reale ale întreprinderii, poate debloca de multe ori serioase fonduri bănești care pot intra în circuitul productiv.

Problemele de stabilire a stocului de siguranță au uneori un caracter de optimizare fiind numite și probleme de optimizare a stocurilor.

Pentru a se trata complet și eficace problemele de urmărire și control a stocurilor, este necesar să se posede o arhivă de istoric a evoluției stocurilor în timp, pentru a permite o serie de prelucrări statistice. Din aceste prelucrări pot fi trase concluzii importante asu-

pră evoluției stocurilor în timp, pentru a stabili politica stocurilor în viitor.

Așa cum s-a arătat la punctul f se poate face un plan de aprovizionare pornind de la necesarul net. În acest caz se ține seama de stocul existent în magazie în momentul aprovizionării și de cantitățile în curs de fabricație.

Evidențele și prelucrările statistice enunțate la punctul j pot fi legate numai de problemele de evoluție a contractelor în timp sau pot fi legate și de oricare alte probleme de la celelalte puncte, dar în special de evoluția vînzărilor.

În general este bine ca prelucrările statistice să fie grupate pentru toate problemele enunțate, deoarece de cele mai multe ori apar legături între acestea, așa cum s-a arătat. De aceea în cap. VI (fig. 6.2.2) organizarea datelor istorice pentru prelucrări statistice este prevăzută ca o aplicație comună tuturor subsistemelor de aplicații.

Trebuie reținut că posibilitățile de simulare formează o caracteristică foarte importantă a aplicațiilor SPAD în întreprinderi. Simulările permit încercarea mai multor variante, în vederea pregătirii în condiții cit mai bune a deciziilor.

Astfel, dacă se planifică un anumit plan de livrări, simularea planificării producției va putea confirma sau infirma posibilitatea realității acestuia, ceea ce permite conducerii întreprinderii să aleagă altă variantă pe care să o verifice.

Legătura de control inversă permite controlul planificării făcute și a desfășurării activităților.

Rezolvarea integrată a tuturor problemelor indicate la punctele c pînă la p, poate fi complet soluționată într-o perioadă de 3 pînă la 5 ani în funcție de complexitatea întreprinderii, (prin rezolvarea completă se înțelege intrarea aplicațiilor în exploatare curentă).

Sistemul de conducere (SIC) se realizează pe baza acestor activități care actualizează baza de date și produce o serie de documente (liste, bonuri de lansare, bonuri de comandă, rapoarte etc.) necesare desfășurării activității de conducere a întreprinderii, dar în special pe baza unor aplicații de sinteză cum ar fi aplicația de antecalcul (precacalcul), postcalcul, analiza funcționării întreprinderii și raportarea prin excepție.

Aplicațiile de sinteză care deservesc direct SIC (sistemul de conducere) la nivele superioare în întreprindere folosesc de asemenea baza de date care este actualizată în cadrul activităților subsistemelor de aplicații pentru planificare și urmărire menționate. Activitatea de

5.1. TIPURI DE APLICARE ALE SIPAD ; SIPAD INTEGRAT

precacalcul realizează operațiile contabile privind evaluarea preliminară a unor indicatori vitali în funcționarea întreprinderii (prețul de cost și de vînzare etc.), neîncadrările în planul de cheltuieli lunare și trimestriale, rentabilitatea diferitelor produse, eficiența locurilor de muncă, estimarea beneficiilor, situații comparative față de anii trecuți etc.

Activitatea de postcalcul execută postocalcul prețului de cost, al beneficiilor, cheltuielilor (pe secții, produs, loc de muncă etc.), calculează indicii tehnico-economiți de funcționare ai întreprinderii, amortismentele, valoarea rebuturilor, efectuează calcule privind gestiunea materialelor, repartizarea transporturilor, valoarea producției neterminate, calculul salariilor personalului și regie, tehnic-administrativ și în acord etc.

Aplicația de analiza funcționării întreprinderii furnizează rapoartele necesare pentru tabloul de bord al conducerii întreprinderii. Ea realizează de obicei comparațiile necesare între situația planificată și cea realizată folosind informațiile obținute prin aplicațiile precacalcul și postcalcul.

În ce privește raportarea prin excepție pentru obținerea de informații necesare conducerii operative, cel mai adesea este recomandabil să se utilizeze un pachet (limbaj) de interogare a bazei de date.

În figura 5.1.1. se prezintă un exemplu de schemă în care sunt reprezentate legăturile principale posibile între activitățile de planificare, urmărire și de sinteză necesare SIC.

Un exemplu concret privind fișierele care constituie baza de date și legăturile dintre acestea, cerute de corelarea aplicațiilor menționate în cadrul procesului de integrare a sistemului de conducere cu mijloace de automatizare într-o întreprindere textilă, este prezentat în fig. 5.1.2.

În această schemă baza de date este constituită din 8 fișiere de referință (permanente) și o serie de fișiere variabile grupate pe diferite activități care se integrează în cadrul sistemului. Menționăm de asemenea în cadrul acestui exemplu, efectuarea centralizată (îndependentă de subsistemele de aplicații) a actualizării și interogării fișierelor de referință. Numărul fișierelor din baza de date a căror actualizare, validare și interogare (pentru raportări de obicei prin excepții), urmează să se facă centralizat, este de dorit să fie relativ, cu numărul fișierelor prelucrate de aplicațiile separate, cît mai mare; în cap. 6 vom vedea că această tendință pune în valoare avantajele concepției despre sistemele integrate și bazele de date asociate lor.

În proiectarea sistemelor informative trebuie să se țină seama și de volumul de informație obținut în rapoartele produse pe SPAD de programele asociate acestor aplicații.

Volumul informației distribuite la anumite nivele apare sub forma unei piramide. Pe măsură ce nivelul coboară, volumul de informație crește. Este util în acest sens să se țină seama de volumul de informație utilizabil pe individ.

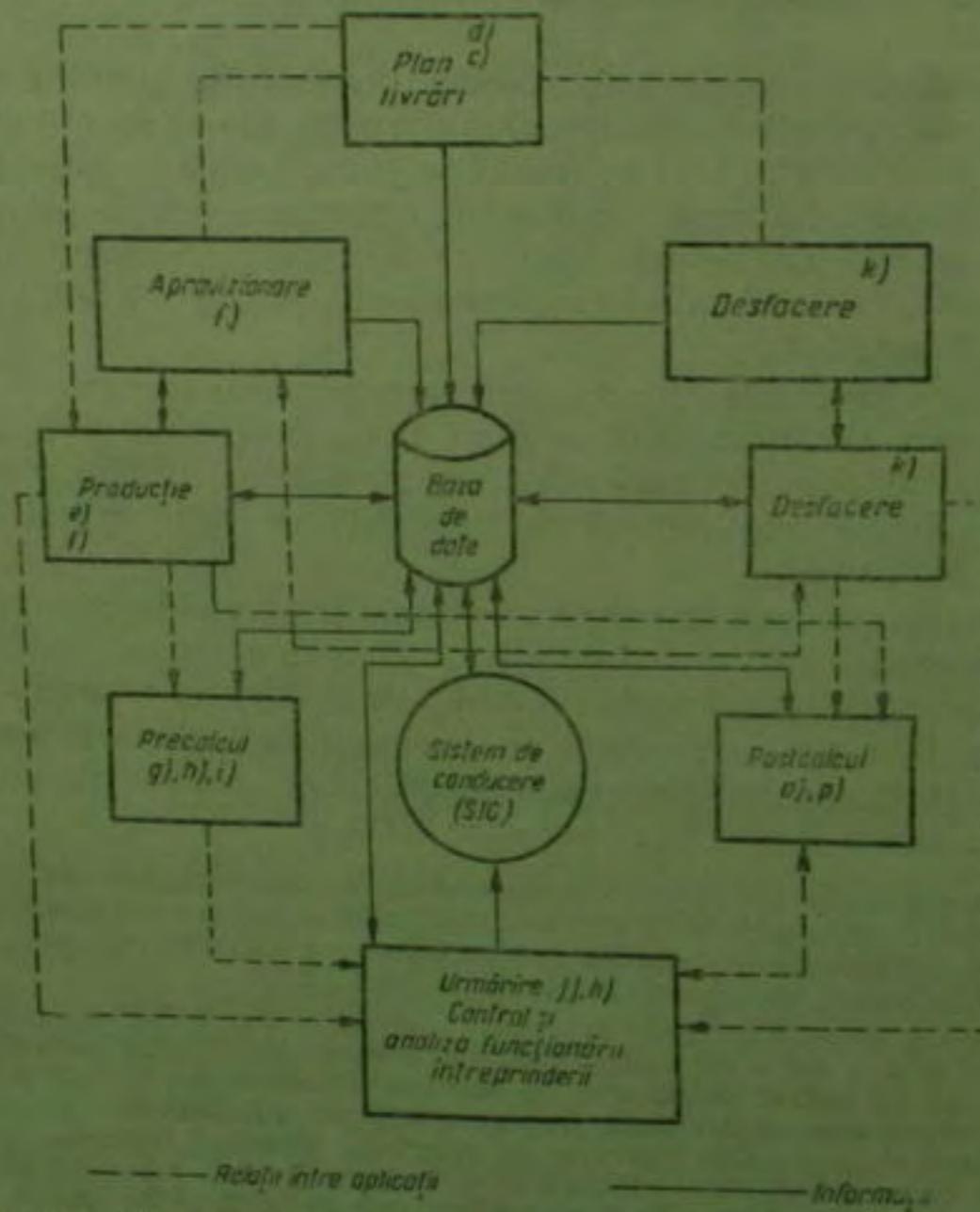
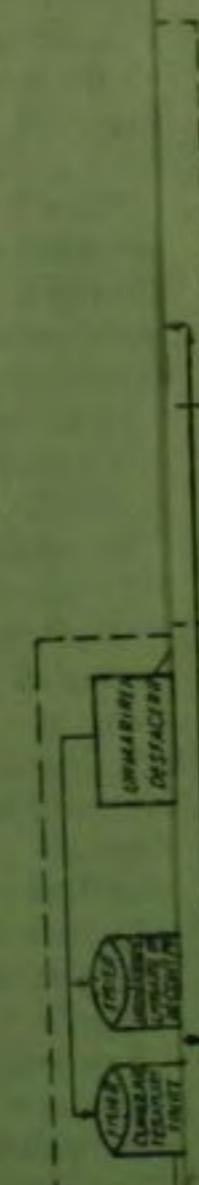


Fig. 5.1.1. Legăturile principale între activitățile de planificare, urmărire și de sinteză necesare SIC.

Dacă volumul de informație crește, pe măsură ce ne îndreptăm către baza piramidei și numărul de indivizi care utilizează informația crește.

Fig. 5.1.2. Schema unui sistem integrat (actualizarea și interogarea bazei de date).



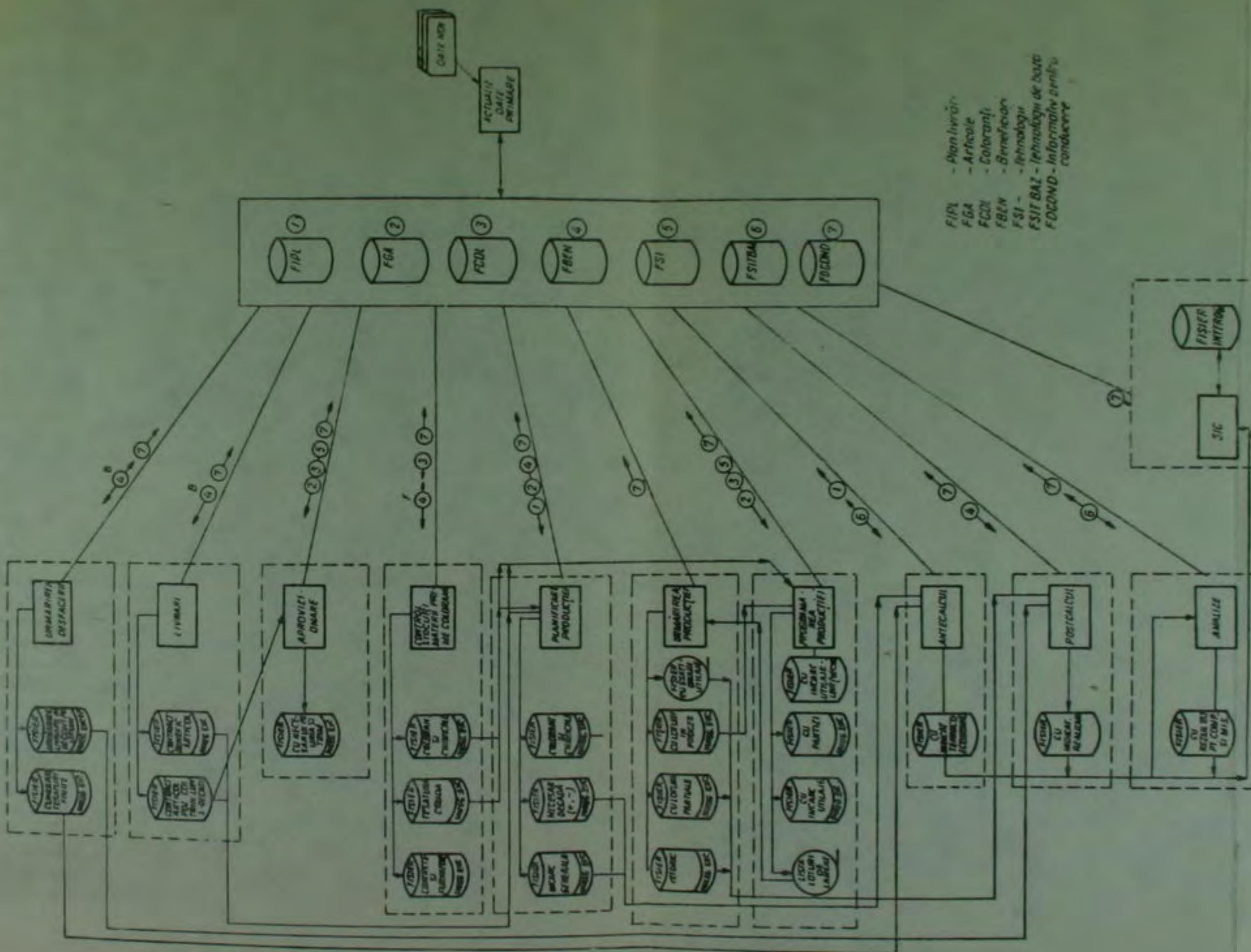


Fig. 5.1.2. Schema unui sistem integrat (actualizarea și interogarea bazei de date).

În general se poate considera în primă aproximare că informația nu trebuie să depășească în medie circa 150 de rânduri de 120 de caractere, pe individ și pe zi.

Un volum mai mare de informație, poate denatura scopul pentru care a fost elaborată. Dacă volumul de informație este mai mare, poate aduce prejudicii celorlalte activități pe care trebuie să le îndeplinească cel care utilizează informația.

Fie de rezolvat de exemplu următoarea problemă :

O întreprindere lucrează anual cu cca. 20 000 de furnizori. Fiecare furnizor poate încheia cu întreprinderea respectivă, presupunem maxim 15 contracte, iar pentru fiecare contract pot fi încheiate maximum 10 documente, în funcție de produsele indicate în contract.

Se dorește o urmărire lunară a onorarii contractelor de către furnizori. Să facem un calcul estimativ.

Pentru un document se pot tipări 3 pînă la 6 rânduri de cîte 120 de caractere. O înregistrare din fișier are 120 de caractere lungime.

Fișierul care conține furnizorii, contractele și documentele, are dimensiunea maximă de 20.000 * 15 * 10 * 6 = 18.000.000 de înregistrări. Deoarece nu toți beneficiarii au numărul maxim de înregistrări, presupunem că numărul total de înregistrări este de circa 10.000.000

Dacă s-ar tipări toate acestea ar rezulta că informația obținută lunar ar fi de 10.000.000 de rânduri. Considerind că pe o pagină încap 100 de rânduri, s-ar obține 100.000 de pagini a cîte 100 de rânduri fiecare. Volumul de informație astfel obținut este imens și informația nu poate folosi în general la nimic.

În acest caz se acționează prin excepții. Dacă de exemplu interesează numai furnizorii care sunt în întîrziere cu un anumit număr de zile, volumul de informație poate fi redus foarte mult.

Să considerăm de pildă că ne interesează numai furnizorii care sunt în întîrziere cu mai mult de 5 zile ; în acest caz volumul poate fi redus de pildă la aproximativ 5000 de rânduri adică aproximativ 50 de pagini, reprezentînd o informație ce poate fi utilă.

Conducerea întreprinderii poate fi interesată să cunoască pe acei furnizori care au o întîrziere mai mare de 15 zile.

În acest caz, volumul de informație poate fi de pildă de aproximativ 400 pînă la 500 de rânduri, adică de 4 pînă la 5 pagini. În orice caz, acest volum de informație este accesibil.

Acțiunea de introducere a aplicațiilor menționate trebuie precedată în mod obligatoriu de un studiu de oportunitate (preliminar) a introducerii P.A.D. în întreprindere.

Studiul de oportunitate trebuie să cuprindă o analiză cit mai corectă a situației actuale a sistemului informațional existent, să cuprindă critica sistemului actual care să arate la fiecare nivel de conduce care sunt informațiile redondante (paralelisme), care informații sunt de prisos și care informații lipsesc.

De aici se pot trage o serie de concluzii asupra necesarului de informații pentru fiecare nivel ierarhic și asupra conținutului noului sistem.

Studiul trebuie să analizeze atât sistemul manual, cit și diverse variante de sisteme de mecanizare sau PAD pe care să le compare din punct de vedere al eficienței economice.

Eficiența economică se calculează de obicei pe un anumit număr de ani și cuprinde toate aspectele, începând de la cheltuielile de proiectare, implementare și exploatare a sistemului și continuând cu economiile care pot fi obținute în urma aplicării sistemului prin îmbunătățirile pe care le poate aduce în activitatea economică a întreprinderii.

Acest studiu formează baza de plecare pentru noul sistem care urmează să fie proiectat și implementat.

Indiferent dacă aplicațiile arătate la punctele c—f sunt atașate în paralel sau succesiv în vederea integrării, trebuie subliniat faptul foarte important că ele trebuie să utilizeze o bază de date comună, spre a evita înregistrarea unor date de mai multe ori în diferite fișiere, ceea ce mărește timpul de prelucrare și necesită mai multe echipamente periferice.

Principiul unicării înregistrării datelor (informației) trebuie să fie respectat și datorită faptului că în timpul rulării aplicațiilor este necesar să se facă actualizarea fișierelor din baza de date, cu informațiile din întreprindere în mod univoc.

Dacă actualizările acelorași date apar de mai multe ori în diferite aplicații, este posibil ca modificările să fie operate în baza de date în mod neunitar. În afara faptului că efectuarea mai multor modificări mărește timpul de prelucrare, mai există pericolul real de a nu se opera modificările în unele fișiere, apărind astfel fie o situație de neconcordanță în timpul exploatarii sistemului informațional, fie obținându-se rezultate eronate, ceea ce este foarte grav, putând conduce la compromiterea sistemului informațional.

De aceea realizarea sistemelor integrate de conducere cu mijloace de automatizare necesită în primul rînd organizarea bazei de date în care se respectă principiul unicării datelor, care pe baza

5.1. TIPURI DE APLICARE ALE SIPAD : SIPAD INTEGRAT

legăturilor stabilită între fișiere și înregistrări permite pe de o parte reorganizarea automată a bazei de date, în mod unitar, pe măsură ce se adaugă noi înregistrări sau se sterg unele dintre cele existente, evitându-se apariția situațiilor de neconcordanță și a erorilor, iar pe de altă parte permite un acces rapid la informațiile utilizate în comun de subsistemele de aplicații.

După cum vom vedea în cap. 6 (fig. 6.2.2.) generarea, organizarea și actualizarea bazei de date se proiectează ca activități legate de baza de date, comune tuturor activităților din cadrul subsistemelor de aplicații.

Un alt principiu care trebuie respectat, este acela al evitării introducerii unor date prelucrate manual în diferitele etape ale prelucrării.

Sistemul trebuie să permită luarea deciziilor numai ca rezultat al consultării informațiilor furnizate de către sistem.

Este recomandabil ca datele furnizate să nu fie rezultatul unei prelucrări manuale, ci chiar datele primare culese direct de la sursă.

Un asemenea sistem care se bazează pe o bază comună de date și care nu primește date prelucrate manual, realizează cadrul pentru elaborarea unui sistem integrat.

Alt principiu, care trebuie respectat în proiectarea unui sistem SPAD, este acela că funcțiile sistemului se stabilesc la diferite nivele ierarhice cu participarea directă a utilizatorului informației de la nivelul respectiv.

De obicei, aplicațiile încep de la baza piramidei către vîrf, putând obține astfel în final informațiile de sinteză necesare diferitelor nivele de conduce. Funcțiile sistemului se stabilesc succesiv, pornind de la vîrful piramidei către bază, prin analiza informațiilor necesare deciziilor.

În orice activitate economică care are caracter de planificare, pot fi obținute anumite variante optime fie direct prin metode matematice care calculează optimul, fie prin selectarea din mai multe variante (simulare).

În prima fază, în general, se implementează aplicațiile la nivel de evidență și urmărire care trebuie să creeze sursa de date pentru faza a doua în care se execută lucrările de optimizări și simulări.

Însăși aplicațiile elaborate, la nivel de evidență și urmărire, pot fi utilizate pentru simularea și calculul unor variante care să dea posibilitatea alegerii unei soluții mai bune decit soluția inițială.

Trebuie ținut seama de faptul că aplicațiile de evidență și urmărire pun la dispoziție în general informații oportune și exacte, pun ordine în sistemul informațional și pot îmbunătăți calitatea deciziilor, dar nu furnizează soluțiile optime în cadrul activității de planificare.

Modelele de optimizare furnizează direct soluția optimă, fără a fi executate mai multe încercări și în multe situații prezintă avantaje deosebite.

Chiar și simulările, care nu furnizează direct soluția optimă, dacă sunt folosite în cazurile în care nu se pot aplica metode de optimizare, pot conduce la avantaje economice mari.

După cum vom mai arăta în cap. 6, eficiența utilizării metodelor pentru diferite activități (planificare, prognoză, programare operativă, controlul stocurilor, studiul pieței etc.), subsisteme de aplicații sau chiar, în perspectivă, pentru unele sisteme informaționale în ansamblu, este condiționată de introducerea sistemelor informaționale, în care datele sunt organizate în baza de date, care nu mai conține erori și în care funcționează principalele subsisteme de aplicații corelate între ele, la nivelul de evidență și urmărire a activităților.

5.2. Exemple de aplicare a SPAD în industria textilă

5.2.1. Probleme generale privind planificarea și urmărirea producției.

Exemplul care urmărește descriu aplicații elaborate pe SPAD în domeniul industrial textil din finisaj, în sprijin din industria bumbacului.

Se folosesc o serie de termeni specifici acestei industrii, astfel:

ARTICOLUL este o anumită țesătură cu anumite caracteristici, care intră în procesul de fabricație. Articolul este codificat numeric cu 4 caractere.

POZITIA COLORISTICA reprezintă culoarea în care se vopsește o anumită țesătură. În poziția coloristică intră caracteristicile culorii și ale coloranților ce se folosesc pentru obținerea acestei culori.

Pozitia coloristică este codificată cu 6 caractere alfanumerice. Țesăturile sunt au o singură poziție coloristică.

DESENUL reprezintă un model de imprimare pe țesătură, independent de culorile în care va apărea acest desen pe țesătura respectivă. Un anumit desen poate apărea în mai multe culori.

5.2. EXEMPLE DE APLICARE A SPAD (IN INDUSTRIA TEXTILA) 159

Desenul împreună cu fiecare poziție coloristică sunt de asemenea codificate ca și poziția coloristică, în 6 caractere alfanumerice, primele două caractere reprezentând numărul desenului, iar următoarele 4 caractere reprezentând culoarea în care apare desenul.

IMPRIMEUL reprezintă țesătura imprimată și este compus din același desen cu mai multe poziții coloristice.

Ca atare, imprimeul nu este reprezentat printr-un singur cod, ci prin mai multe coduri.

Se consideră imprimeuri diferite și acele imprimeuri care au același desen, dar în culori diferite, articolul fiind același.

De asemenea, se consideră imprimeuri diferite și cele care au același desen și pozitii coloristice, dar sunt imprimate pe articole diferite.

În procesul tehnologic, ca materii prime în afara țesăturii, se mai folosesc diferiți coloranți și chimicale.

Principalele ateliere și servicii care participă la activitatea de finisaj sunt:

- serviciul de lansare-producție;
- atelierul de albitorie;
- atelierul de vopsitorie;
- atelierul de imprimerie;
- atelierul de ajustaj.

În serviciul de lansare-producție se face lansarea țesăturilor în fabricație pe loturi de țesături.

Un lot reprezintă o cantitate de țesătură de o anumită lungime, în general de ordinul miielor de metri, dictată de considerente tehnologice.

Loturile diferitelor articole și pozitii coloristice, pot avea lungimi diferite, însă aceeași lungime pentru un anume articol și anumite pozitii coloristice date. În fabricație se lansează numai loturi întregi.

Lansarea trebuie să se facă conform graficului de livrare. Ea poate fi lăsată în avans, adică lotul să fie gata de fabricat cu un număr de zile înaintea livrării astfel încât termenul de livrare către beneficiar, prevăzut în contract, să nu fie depășit.

Termenul de livrare constituie factorul dominant în stabilirea datei lansării loturilor.

După ce loturile au fost lansate, acesta se constituie în partiz. Mai multe loturi formează o partidă. Partizile se formează ținând seama că în general capacitatea unor utilaje este limitată, de exemplu la autoclave 1300 kg; autoclavale nu admit în general o variație mai mare de $\pm 5\%$ din această greutate, deoarece, de pildă, dacă autoclava este subîncercată, țesătura se poate răsturna în timpul procesului de fierbere, ceea ce produce greutăți la scoaterea țesăturii din autoclavă.

În acest mod în autoclavă poate intra numai un anumit număr de loturi întregi.

Nu toate țesăturile trec prin autoclavă; aceste țesături nu se constituie în partizi sau altfel spus, partizile țesăturilor care nu trec prin autoclavă nu trebuie să fie de o anumită greutate. Într-o partidă pot intra anumite articole și poziții coloristice. Nu orice combinații sunt admise, datorită necesității tractării țesăturilor cu anumiți coloranți și chimicale.

După ce țesătura a trecut prin atelierul de albitorie, intră apoi în atelierul de vopsitorie, unde țesăturile sunt vosite în culori uni, sau în atelierul de imprimare unde sunt imprimate.

Înainte și după vopsire, țesăturile sunt tratate cu diferite chimicale pentru uniformizarea și fixarea culorii. Din atelierul de vopsitorie țesăturile trec în atelierul de ajustaj unde sunt controlate calitativ și cantitativ, apoi tăiate în bucăți comerciale, iar de aici țesătura intră în magazia de produse finite.

În timpul controlului de calitate se sesizează defectele care au apărut, iar cele care nu pot fi reținute sunt înălțurate. Porțiunile de țesătură care nu pot fi reținute se tăie separat și se vând fie sub formă de cupoane cu bonificări, fie sub alte forme.

În cursul procesului tehnologic, în timpul unor controale între diferite faze, pot fi sesizate defecte pe anumite loturi. În acest caz, loturile cu defecte sunt ajustate chiar în atelierul de vopsitorie, o parte din lotul ce prezintă defecte rămânând în stoc parțial chiar în secție.

Acstei loturi se numește — loturi parțiale — și poartă codul lotului de proveniență, dar cu mențiunea că este al n -lea lot, deci apare în plus numărul de secvență ca lot parțial.

În general planificarea producției se poate face trimestrial, iar actualizarea planificării, oricând în cursul trimestrului.

Actualizările la planul de producție trimestrial se fac în general, fie datorită unor modificări survenite în planul de livrare, fie lipsei de materie primă, fie datorită unor motive de ordin tehnologic.

În cursul procesului tehnologic mașinile trebuie spălate ori de câte ori se schimbă culoarea țesăturilor sau cînd culoarea rămînește aceeași, se schimbă colorantul.

Schimbarea frecvență a culorilor este dezavantajoasă, deoarece crește timpul de staționare a utilajelor, pentru efectuarea procesului de spălare.

Reducerea timpilor de spălare poate fi obținută prin două căi:

1. Prin stabilirea planului de livrare, astfel încît lunar să se producă țesături într-un număr mai mic de culori, grupindu-se culorile în cadrul fiecarei luni și respectindu-se legislația în vigoare.

5.2. EXEMPLE DE APLICARE A SPAD (ÎN INDUSTRIA TEXTILĂ) 161

2. Prin programarea producției astfel încît în cadrul unui interval de timp producția să se desfășoare astfel ca timpii de staționare ai mașinilor să fie minimi.

Pentru *pregătirea și desfășurarea procesului de fabricație* este necesară îndeplinirea următoarelor condiții:

1. În magazia de țesături crude să existe verificată calitativ țesătura care urmează să fie lansată.

2. În magazia de coloranți și chimicale să existe la datele la care apar nevoie, cantitățile impuse de procesul tehnologic de coloranți și chimicale.

3. Loturile care se lansează să corespundă unor termene de livrare într-un anumit interval de timp (de exemplu o lună de la data lansării).

Lansarea se face în conformitate cu planul de livrare stabilit în timpul contractărilor ce au loc cu un trimestru înainte de începerea perioadei de fabricație pentru două sau trei trimestre.

Pentru lansarea în producție se stabilesc loturile care se lansează într-un interval dat de timp (de exemplu decadal).

Fiecare lot primește un termen de execuție conform planului de livrare.

Programele de planificare a producției stabilesc încărcarea utilajelor cu loturile lansate în așa fel încât loturile să fie fabricate la termenul dat.

Se fac controalele de la punctele 1. și 2. și se iau măsuri de fabricare a loturilor în avans pentru a evita eventualele goluri de producție care ar putea apărea datorită inexistenței țesăturilor, sau coloranților în magazia de materii prime și materiale, caz care se poate întâmpla fie datorită faptului că unii furnizori nu și-au onorat contractele, fie datorită calității ncorespunzătoare a unor țesături.

În acest fel înaintea fiecărui trimestru se poate face o planificare corespunzătoare a producției, plecind de la planul de livrări și de la situația producției pe trimestrul (luna) în curs.

La planificarea producției va rezulta fie că planul de producție poate fi executat în cursul trimestrului, adică contractele pot fi onorate, fie că planul de producție nu poate fi executat și atunci conducerea va decide adoptarea unor măsuri sau renunțarea la o serie de contracte.

În baza planului trimestrial de producție rezultat din planul de livrare, se întocmește planul de aprovizionare trimestrială.

O dată cu lansarea se poate face și defalcarea planului de aprovizionare cu chimicale și coloranți, pe decade, stabilindu-se cantitățile și datele necesare.

Loturile se vor repartiza în interval astfel :

- Loturile pentru care există țesătură în magazie.
- Loturile pentru care nu există țesătură, dar urmează a fi primită cu siguranță de la furnizori.
- Loturile pentru care nu există țesătură, și nici nu se știe dacă se va primi la timp.

O nouă programare poate fi făcută, fie la sfîrșitul unui interval, fie atunci cind vechea programare nu mai este valabilă, oricând în cadrul intervalului.

Lipsa anumitor coloranți și chimice din magazie nu conduce în general la anularea lansării loturilor sau la întârzierea termenelor de execuție, deoarece în rețete coloranții pot fi schimbați frecvent cu înlocuitori sau poate fi utilizată altă rețetă.

În timpul fabricației, urmărirea producției se face pentru acele loturi care au fost introduse în procesul de fabricație.

Urmărirea producției trebuie să rezolve următoarele probleme :

- Să se țină evidență loturilor care au trecut prin diferite mașini, data, ora și minutul intrării și ieșirii de pe mașină.
- Să țină evidență loturilor parțiale care există în stoc și care intră în stoc și să se șteargă din această evidență acele loturi parțiale care au fost relansate în producție.
- Să țină evidență intrării și ieșirii din staționare a utilajelor, precum și codul cauzelor care generează aceste staționări. În această categorie intră toate staționările, inclusiv cele datorate spălărilor.
- Să țină evidență loturilor fabricate, a lungimii acestora, a calității lor, furnizind datele necesare la intrarea în magazia de produse finite.
- Să calculeze cheltuielile de producție realizate, ținând seama că se cunosc cheltuielile pe articol, poziție coloristică și metru de țesătură pentru fiecare mașină în parte, precum și pentru imprimeuri.
- Să asigure legătura cu programele de planificarea și programarea producției, deoarece orice execuție a acestor programe pentru un interval de timp, implică pornirea de la o anumită situație inițială care trebuie să fie furnizată de către programe de urmărire și control.
- Să transmită programele de lansare din cadrul programelor de planificare a producției, datele despre loturile parțiale lansate, care vor fi tratate ca loturi normale de lungimi diferite decât cele tehnologice și care vor intra apoi în urmărire ca loturi normale.
- Să indice loturile lansate, dar care n-au intrat în fabricație la termenele stabilite.

Rapoartele furnizate de aceste programe sunt prezentate la diverse nivele de conducedre, în funcție de gradul de sinteză sau de excepție care este necesar.

Astfel, de pildă, pentru conducedrea întreprinderii se pot da informații privind loturile parțiale existente în stoc de cel puțin o săptămână, mașinile care au staționat din diverse cauze mai mult decât 12 ore etc.

5.2.2. Proiectarea documentelor (rapoartelor) furnizate de aplicația „urmărirea producției”

Pentru a putea stabili sistemul de prelucrare a datelor este necesar să se cunoască mai întâi rapoartele ce sunt necesare (adică vom stabili ieșirile pentru a putea defini intrările și prelucrările din cadrul sistemului).

În cele opt puncte enumerate la urmărirea producției se stabilesc de fapt informațiile pe care diverse nivele de conducedre din întreprindere trebuie să le primească.

În cazul concret pe care-l prezentăm au fost definite următoarele rapoarte :

- RP09M (Anexa 1.1) : raportează mașinile intrate și ieșite din staționare la o anumită dată pînă la data curentă a prelucrării, la care întreuperea a durat un interval de timp mai mare sau egal cu timpul dat printr-o carteală parametru, în ordinea descrescătoare a mărimii intervalului, începînd cu mașina care a staționat cel mai mult timp, pînă la mașina cu interval de staționare cel mai mic.

La intervale egale se ia în considerație data intrării în staționare, trecîndu-se mai întîi acele mașini care au intrat în staționare la o dată mai mică.

La sfîrșitul raportului apare lista cu mașinile care sunt încă în staționare la data raportării.

- RP10M (Anexa 1.2). În acest raport se prezintă mașinile care au fost în staționare într-un anumit interval de timp (putînd include și data zilei cînd se face prelucrarea) și care au ieșit din staționare în ordinea crescătoare a codurilor (mașinilor).

În raport se ia în considerare, de asemenea, un timp minim de staționare și vor fi raportate numai acele mașini care au staționat un timp mai mare sau egal cu timpul minim de staționare.

Pentru aceeași mașină, raportarea se face în ordinea crescătoare a datelor intrării în staționare.

Modul de încadrare al raportului în foaia de imprimantă apare în anexa 1.2.

- RP12M (Anexa 1.3). În acest raport se prezintă de asemenea mașinile care au intrat și ieșit din staționare, în ordinea codurilor cauzelor.

Raportarea se face pe cod cauză, în cadrul acelasi cauze pe mașini, iar pentru aceeași mașină în ordinea descrescătoare a intervalului.

Loturile se vor repartiza în interval astfel :

- Loturile pentru care există ţesătură în magazie.
- Loturile pentru care nu există ţesătură, dar urmează să primească cu siguranță de la furnizor.
- Loturile pentru care nu există ţesătură, și nici nu se știe dacă se va primi la timp.

O nouă programare poate fi făcută, fie la sfîrșitul unui interval, fie atunci cind vechea programare nu mai este valabilă, oricând în cadrul intervalului.

Lipsa anumitor coloranți și chimicale din magazie nu conduce în general la anularea lansării loturilor sau la întârzierea termenelor de execuție, deoarece în rețete coloranții pot fi schimbați frecvent cu înlocuitorii sau poate fi utilizată altă rețetă.

În timpul fabricației, urmărirea producției se face pentru acele loturi care au fost introduse în procesul de fabricație.

Urmărirea producției trebuie să rezolve următoarele probleme :

- Să se țină evidență loturilor care au trecut prin diferite mașini, data, ora și minutul intrării și ieșirii de pe mașină.
- Să țină evidență loturilor parțiale care există în stoc și care intră în stoc și să se șteargă din această evidență acele loturi parțiale care au fost relansate în producție.
- Să țină evidență intrării și ieșirii din staționare a utilajelor, precum și codul cauzei care generează aceste staționări. În această categorie intră toate staționările, inclusiv cele datorate spălărilor.
- Să țină evidență loturilor fabricate, a lungimii acestora, a calității lor, furnizind datele necesare la intrarea în magazia de produse finite.
- Să calculeze cheltuielile de producție realizate, ținând seama că se cunosc cheltuielile pe articol, poziție coloristică și metru de ţesătură pentru fiecare mașină în parte, precum și pentru imprimeuri.
- Să asigure legătura cu programele de la planificarea și programarea producției, deoarece orice execuție a acestor programe pentru un interval de timp, implică pornirea de la o anumită situație inițială care trebuie să fie furnizată de către programe de urmărire și control.
- Să transmită programelor de lansare din cadrul programelor de planificare a producției, datele despre loturile parțiale lansate, care vor fi tratate ca loturi normale de lungimi diferite decât cele tehnologice și care vor intra apoi în urmărire ca loturi normale.
- Să indice loturile lansate, dar care n-au intrat în fabricație la termenele stabilite.

Rapoartele furnizate de aceste programe sunt prezentate la diferite nivele de conducere, în funcție de gradul de sinteză sau de excepție care este necesar.

Astfel, de pildă, pentru conducerea întreprinderii se pot da informații privind loturile parțiale existente în stoc de cel puțin o săptămână, mașinile care au staționat din diverse cauze mai mult decât 12 ore etc.

5.2.2. Proiectarea documentelor (rapoartelor) furnizate de aplicația „urmărirea producției”

Pentru a putea stabili sistemul de prelucrare a datelor este necesar să se cunoască mai întâi rapoartele ce sunt necesare (adică vom stabili ieșirile pentru a putea defini intrările și prelucrările din cadrul sistemului).

În cele opt puncte enumerate la urmărirea producției se stabilesc de fapt informațiile pe care diverse nivele de conducere din întreprindere trebuie să le primească.

În cazul concret pe care-l prezentăm au fost definite următoarele rapoarte :

- RP09M (Anexa 1.1) : raportează mașinile intrate și ieșite din staționare la o anumită dată pînă la data curentă a prelucrării, la care intreruperea a durat un interval de timp mai mare sau egal cu timpul dat printre cartela parametru, în ordinea descrescătoare a mărîmii intervalului, începînd cu mașina care a staționat cel mai mult timp, pînă la mașina cu interval de staționare cel mai mic.

La intervale egale se ia în considerație data intrării în staționare, trecindu-se mai întîi acele mașini care au intrat în staționare la o dată mai mică.

La sfîrșitul raportului apare lista cu mașinile care sunt încă în staționare la data raportării.

- RP10M (Anexa 1.2). În acest raport se prezintă mașinile care au fost în staționare într-un anumit interval de timp (putînd include și data zilei cînd se face prelucrarea) și care au ieșit din staționare în ordinea crescătoare a codurilor (mașinilor).

În raport se ia în considerare, de asemenea, un timp minim de staționare și vor fi raportate numai acele mașini care au staționat un timp mai mare sau egal cu timpul minim de staționare.

Pentru aceeași mașină, raportarea se face în ordinea crescătoare a datelor intrării în staționare.

Modul de încadrare al raportului în foaia de imprimantă apare în anexa 1.2.

- RP12M (Anexa 1.3). În acest raport se prezintă de asemenea mașinile care au intrat și ieșit din staționare, în ordinea codurilor cauzelor.

Raportarea se face pe cod cauză, în cadrul acelasi cauze pe mașini, iar pentru aceeași mașină în ordinea descrescătoare a intervalului.

- Se ia de asemenea în considerare un timp de staționare minim și se vor raporta pe cauze numai acele mașini care au staționat un timp mai mare decât timpul minim de staționare considerat. Încadrarea raportului în foaia de imprimantă este dată în anexa 1.3.

4. RP20M (Anexa 1.4). În acest raport se prezintă situația staționării loturilor care se află în fabricație între mașini și ateliere. Aici este vorba numai de loturile normale.

Raportul ține seama de intervalele de staționare ale unui lot între diferite mașini și de ordinea crescătoare a intervalelor de staționare.

Se raportează întâi lotul care a staționat cel mai mult timp, pentru acesta prezentându-se staționările. Se raportează apoi lotul următor, și așa mai departe.

Și aici se ia în considerare un timp minim de staționare și se raportează numai acele staționări care au durat mai mult decât timpul minim de staționare.

Acest raport folosește la elucidarea cauzelor care au condus la întirzirea fabricației pentru unele loturi lansate și care au început să fie încărcate pe mașini.

Se consideră ca primă mașină din lanț, atelierul de lansare. Loturile care au fost date ca lansate, dar n-au fost încărcate pe mașini de un anumit număr de zile, apar de asemenea în acest raport, la începutul acestuia.

Încadrarea acestui raport în foaia de imprimantă este dată în anexa 1.4.

5. RP15 M (Anexa 1.5). Acest raport prezintă situația loturilor parțiale intrate în staționare și care nu au fost relansate sau reîncărcate pe mașini de mai mult decât un număr de zile dat. Acest interval se stabilește în funcție de nivelul de conducere la care se face raportarea.

Raportul se obține în ordinea crescătoare a datelor la care lotul a intrat în staționare, sau altfel spus, în ordinea descrescătoare a intervalului de timp dintre data intrării în staționare și data curentă.

Încadrarea acestui raport în foaia de imprimantă este dată în anexa 1.5. Pentru economie de hârtie, pe o pagină se scot două coloane de raportare.

6. RP21M (Anexa 1.6). Raportul prezintă calculul cheltuielilor realizate într-un interval de timp, cheltuielile fiind directe și indirekte, asociate cu articolul respectiv.

Cheltuielile pe articol se tipăresc în ordinea codurilor în intervalul de timp dat, considerat de la ora 6,00 din ziua de început pînă la ora 6,00 din ziua de sfîrșit.

Se fac totaluri parțiale pe pagină și se calculează și totalul general pe intervalul de timp dat și articolele produse.

Încadrarea raportului în foaia de imprimantă se exemplifică în anexa 1.6.

Subliniem că toate intervalele de timp care se raportează, datele de început ale intervalelor, datele de sfîrșit etc., sunt parametri care se transmit sistemului la fiecare rulare.

În acest fel se crează elasticitate în exploatarea sistemului, putindu-se obține fie datele de excepție necesare conducerii, fie datele curente necesare la nivele mai joase.

5.2.3. Proiectarea documentelor (formatelor de cartelă) cu date primare de intrare necesare obținerii rapoartelor

Odată stabilite rapoartele și informațiile necesare pentru raportare, pot fi stabilite și datele necesare la intrare.

În cazul de față, colectarea datelor pentru urmărirea producției se face manual. Pentru aceasta trebuie ținut seama de faptul că volumul acestor date trebuie să fie cît se poate de mic.

De asemenea, persoanele care fac aceste operații (uneori chiar muncitori) nu sunt specializate în asemenea completări de documente și un volum mare de date de colectat ar stinjeni activitatea de producție.

Un volum mare de date colectate ar produce de asemenei greutăți în perforarea cartelelor de date după documentele sursă completate în secții și ar angaja forțe relativ mari în perforarea și verificarea acestora pe mașinile de perforat și verificat cartele. Totodată sără mărî numărul de erori atât la colectarea pe documentele sursă, cât și la perforarea pe cartele.

Tinând seama de motivele arătate mai sus, se poate stabili că este necesar să se colecteze următoarele tipuri de date :

- a. date pentru loturi normale ;
- b. date pentru loturile parțiale care apar în procesul de producție ;
- c. date privind intrarea în staționare, ieșirea din staționare a mașinilor și cauzele intrărilor în staționare ;
- d. date privind bilanțul fabricației loturilor atunci cînd acestea ajung la ultima fază în atelierul de ajustaj, care constituie datele de intrare în magazia de produse finite ;
- e. date privind reîncărcarea pe mașini și relansarea loturilor parțiale, pentru a putea fi sterse din fișierele în care apar în staționare.

Loturile sunt insotite în cursul procesului de fabricație de o fișă de lot, în care se trec secțiile, mașinile pe care a trecut lotul, precum și modificările survenite în cursul procesului tehnologic.

Separarea unui lot în loturi parțiale se face atribuind fiecărui lot parțial o fișă identică cu cea a lotului lansat, dar care poartă și un număr de secvență. De exemplu, 04752/01 reprezintă codul primului lot parțial provenit din lotul cu codul 04752.

Acea parte din lot care își continuă procesul tehnologic va continua să aibă fișă de lot de la lansare.

Un lot parțial poate fi relansat, adică să reincepă fabricarea lui din alibilitate sau să i se schimbe culoarea, căpătând altă poziție coloristică, respectiv alt proces tehnologic. În secția în care a apărut lotul parțial se stabilește dacă lotul parțial se relansează, se reinarcă sau își continuă procesul tehnologic, după ce a fost retușat.

Dacă lotul parțial se relansează, atunci i se atribuie un cod separat ca pentru un lot nou, astfel ales încit să se știe că provine dintr-un lot parțial.

Dacă se reinarcă, trebuie să se comunice mașina pe care lotul își continuă procesul tehnologic.

Dacă apare schimbare de poziție coloristică, atunci se va schimba și procesul tehnologic. Acestea fiind date care țin de planificarea producției, se comunică prin programele de urmărire, programului de încărcare a loturilor în fișierul de loturi din aplicația de planificare.

De asemenea, în urma studierii situației colectării datelor, s-a stabilit că cea mai sigură și comodă colectare se face pe fiecare mașină. Astfel pot fi înregistrate pe fiecare mașină separat intrările în staționare și ieșirile din staționare, precum și codul cauzelor staționării.

Loturile normale care intră și ies de pe mașină, precum și ora și minutul intrării și ieșirii, loturile parțiale care apar la ieșirea de pe o anumită mașină, cu caracteristicile fiecărui lot (lungimile acestora), pot fi înregistrate de asemenea.

Prin urmare nu este necesar să se raporteze următoarele date:

- articolul, care rămâne același în tot cursul fabricației, nesuferind nici o modificare;
- lungimea lotului normal, deoarece aceasta rămâne neschimbată față de cea de la mașina precedentă.

Să raportează lungimea numai atunci cînd apar loturi parțiale și își schimbă valoarea față de mașina precedentă.

- poziția coloristică, dacă la ieșirea de pe mașină lotul nu suportă modificări în acest sens.

În cazul loturilor normale, pe fiecare mașină este necesar să se raporteze numai codul lotului, ora și minutul ieșirii de pe mașină. În scopul reducerii volumului de date culese, s-a renunțat la înscrierea orei intrării pe mașină, care nu este absolut necesară.

Dacă apar loturi parțiale, la ieșirea de pe o mașină, atunci acestea se trec pe o fișă separată în ordinea secvenței lor, pe această fișă fiind înregistrat și lotul care continuă fabricația ca primul în secvență.

S-a adoptat deci principiul înregistrării de la lansare a tuturor datelor necesare pentru fiecare lot și a reinregistrării acestora pe fise pentru diferite mașini, numai atunci cînd datele respective au suferit modificări.

Relansarea loturilor parțiale se face tot de către serviciul de lansare ca și pentru loturile noi.

Toate datele care se colectează apar pe formularele prezentate în anexa 2. Acestea sunt documentele sursă, avînd următoarele caracteristici comune:

- Coloanele 01–02 reprezintă codul documentului;
- Coloanele 02–08 reprezintă data la care se face înregistrarea sub forma zi, lună, an;
- Coloanele 09–11 reprezintă codul mașinii.

Atelierele sunt codificate după același sistem ca mașinile.

Aceste date apar pe documentul sursă o singură dată și se vor perfora pe fiecare cartelă prin copiere.

5.2.4. Descrierea prelucrărilor pentru „planificarea producției”

a. **Condiții generale.** Elaborarea planului de producție în industria textilă este o activitate care se execută în general trimestrial. Datorită însă procesului de fabricație și a momentului în care se face planificarea perioadei următoare, vom lua în considerație și producția lunii premergătoare trimestrului respectiv. În consecință, planul de producție va fi elaborat pentru patru luni, pe care le vom numera de la zero la 3. Partizile și loturile prevăzute pentru luna 0 (zero) se vor executa ca urmare a planificării precedente, dar pentru perioada rămasă pînă la începutul noului trimestru se va lua în considerare și stadiul producției la data întocmirii planificării pentru trimestrul următor.

b. **Inițializarea la începutul perioadei de planificare.** Operația de elaborare a planului de producție cu ajutorul unui sistem de prelucrare a datelor este declanșată de un program de extracție a partizilor și loturilor tranzitorii din planul de producție anterior repartizat pe mașini, conform schemei de sisteme din figura 5.2.2., care are ca parametrii de intrare datele referitoare la calendarul noului trimestru. După extragerea codurilor partizilor și loturilor tran-

zitorii pe o bandă de lucru, în fișierul de mașini se introduce noul calendar. În acest fel mașinile sănătate sunt descărcate de sarcini de producție pe toată perioada următoare.

În pasul următor, pornind de la informațiile privind ultimele reparații efectuate la mașini, se încarcă planul de reparații în fișierul de mașini, în conformitate cu normele de reparații în vigoare (figura 5.2.3). În acest fel, timpul necesar reparațiilor este rezervat în calendarul de lucru al fiecărei mașini.

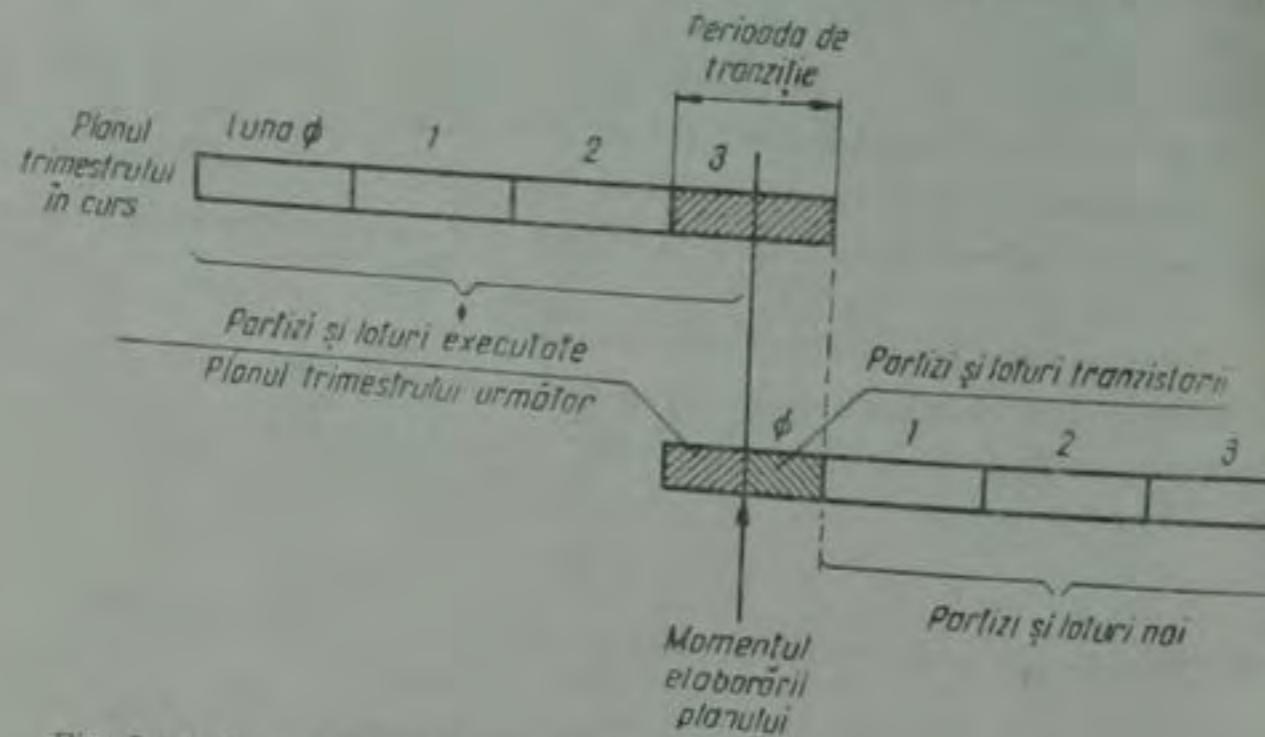


Fig. 5.2.1. Suprapunerea perioadelor la elaborarea planului de producție.

Ultimul pas al procedurii de inițializare prevede constituirea fișierelor curente de loturi și partizi tranzitorii (figura 5.2.4.), care vor fi prelucrate ulterior de lanțul de programe pentru planificarea producției.

c. **Calculul cantităților de planificat.** Sarcina de plan pe trimestrul următor este furnizată de informațiile cuprinse în planul de livrări. Deoarece specificul producției necesită două tehnologii de finisare a țesăturilor, uni și respectiv imprimeuri, planul de livrări va fi prelucrat atât de un program pentru calculul cantităților de planificat pentru țesăturile uni, cât și de un program pentru cantitățile ce se vor planifica la imprimeuri (fig. 5.2.5.). Consultând fișierele tehnologice, aceste programe calculează diversele modificări de lungime ce apar pe liniile de fabricație (alungiri și/sau contracții, capete și alte pierderi), obținând cantitățile reale din care se vor constitui loturile. Aceste programe furnizează totodată și liste de control pentru cantitățile rezultate din calcul.

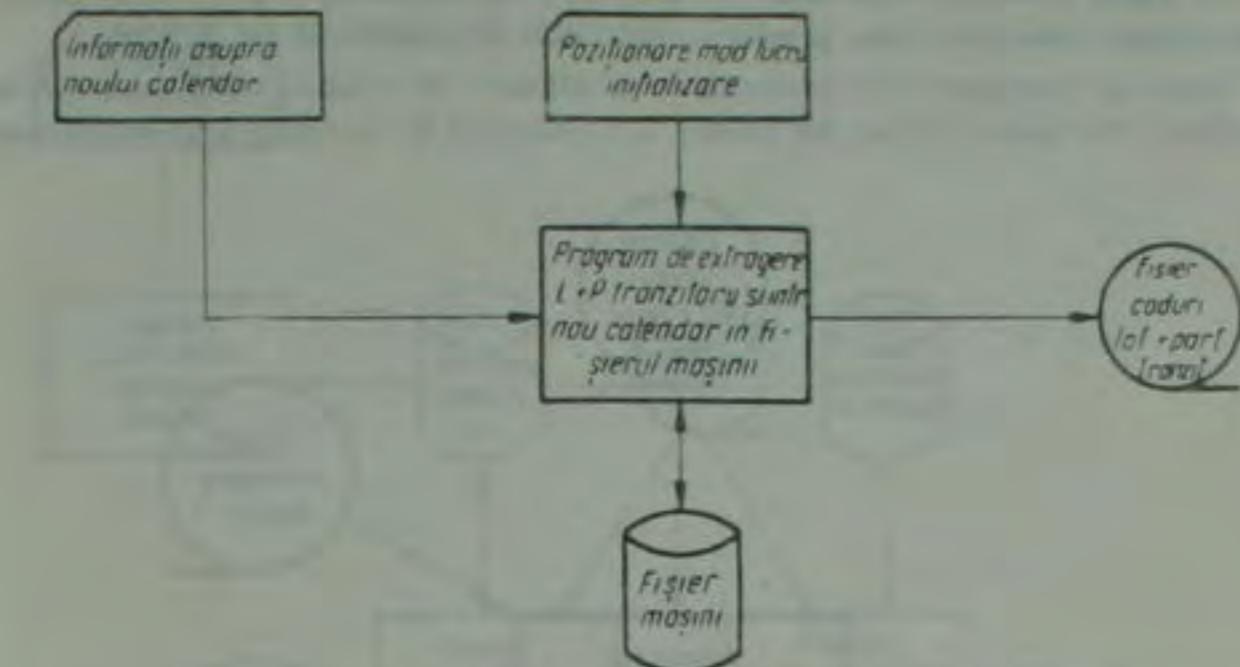


Fig. 5.2.2. Inițializarea fișierelor loturi și partizi la începutul perioadei de planificare.

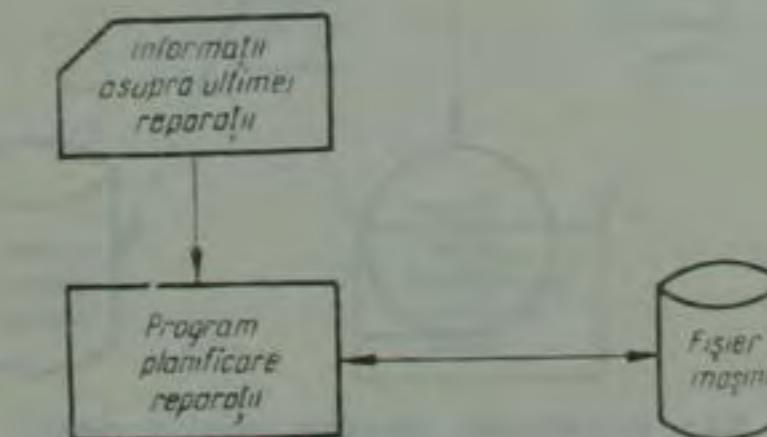


Fig. 5.3.3. Inițializarea fișierelor loturi și partizi la începutul perioadei de planificare luând seama de planul de reparații.

În pasul următor, cantitățile de planificat sunt introduse într-un program de însumare selectivă, care produce cantitățile de planificat pe articole.

Aceleași programe de prelucrare se folosesc și atunci cind apar unele modificări în planul inițial de livrări. Un program de operare a modificărilor:

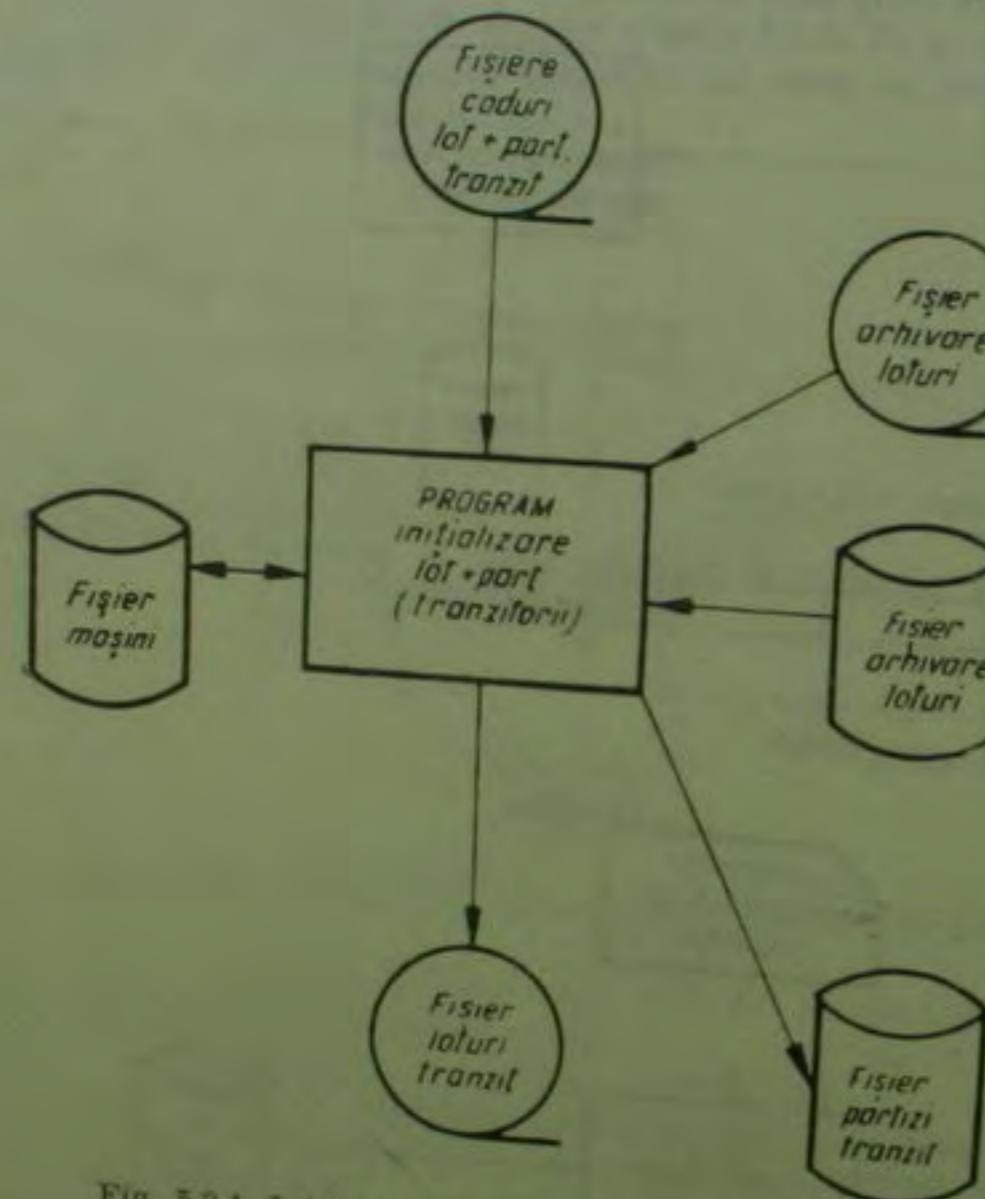


Fig. 5.2.4. Initializarea fișierelor loturi și partizi tranzitoriu la începutul perioadei de planificare.

primește informații atât asupra planului inițial, cât și asupra modificărilor, producând un plan de livrări actualizat (fig. 5.2.6.).

d. **Construirea loturilor noi.** Fișierele care conțin cantitățile de țesături uni și imprimeuri ce trebuie planificate, sunt consultate de programele de constituire a loturilor noi pentru uni și respectiv imprimeuri. În această fază,

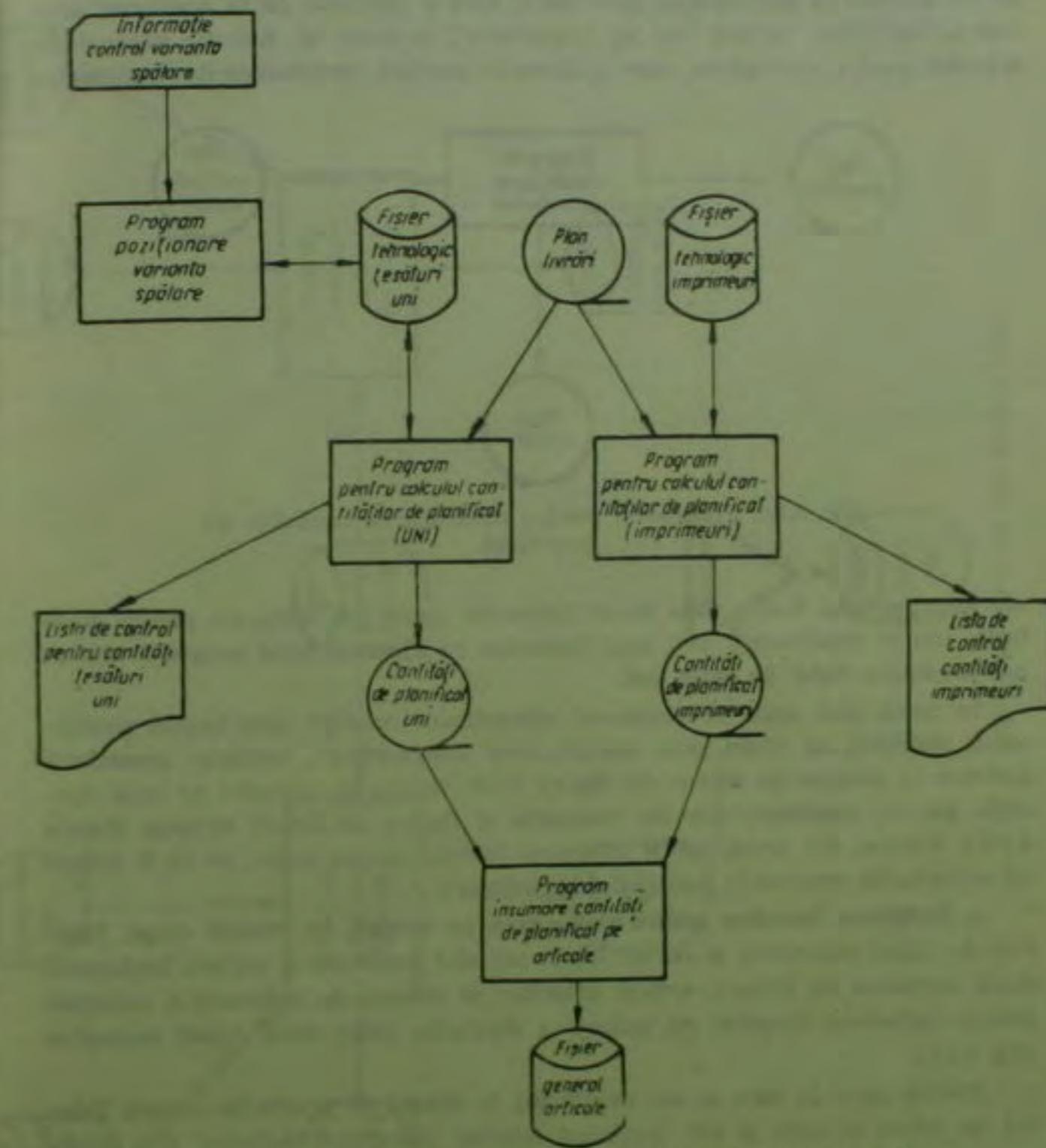


Fig. 5.2.5. Schema de sistem pentru calculul cantității de planificat.

se consultă FIŞIERUL LOTURI TRANZITORII pentru extragerea ultimului cod de lot acordat la planificarea anterioară, spre a continua de la acest cod atribuirea codurilor pentru uni și imprimeuri, precum și fișierul general de articole, pentru extragerea unor informații privind caracteristicile tehnologice

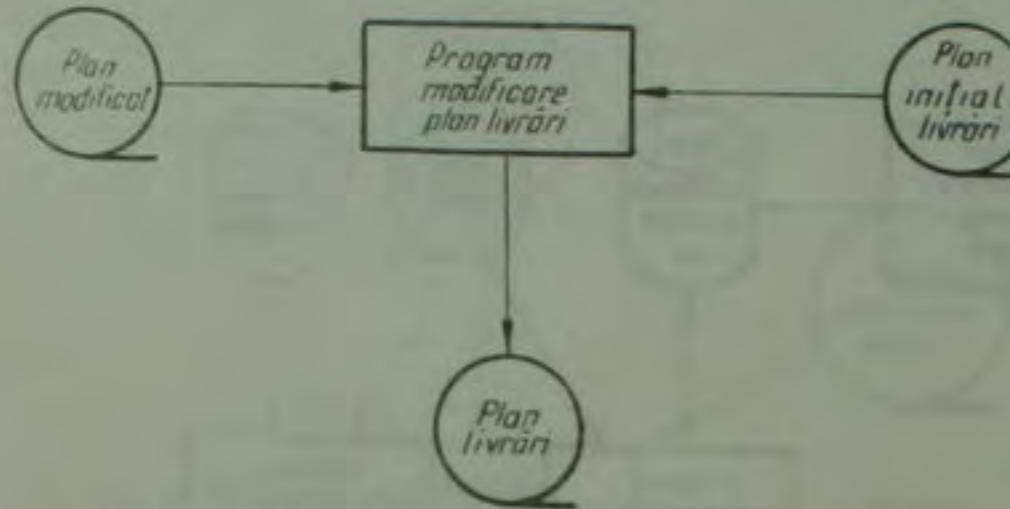


Fig. 5.2.6. Programul pentru modificarea cantității de planificat.

ale loturilor (fig. 5.2.7.). Cele două fișiere de loturi noi, obținute pentru ţesăturile uni și imprimeuri, sunt apoi fuzionate cu ajutorul unui program utilitar intr-un singur fișier de loturi noi.

In cazul cînd apar pe parcursul trimestrului condiții care impun modificarea planului, se obțin prin introducerea informațiilor, conform procedurii indicate în schema de sistem din figura 5.2.6., fișiere de adăugiri pe linia normală pentru suplimentările de contracte și fișiere de loturi anulate (figura 5.2.8.). Acestea din urmă vor fi fuzionate intr-un singur fișier, ce va fi folosit de procedurile descrise în paragrafele următoare.

e. Stabilirea loturilor pentru încărcarea pe mașini. În această etapă, fișierele de loturi tranzitorii și cel de loturi noi sunt fuzionate și sortate (ordonate) după termenul de livrare, articol priorită și criterii de ordonare a culorilor pentru reducerea timpilor de spălare a mașinilor între două culori succeseive (fig. 5.2.9.).

Pentru cazul în care se fac modificări în planul de producție curent, fișierul de loturi anulate și cel conținind loturile planificate anterior sunt introduse pe ramura de modificare ilustrată în figura 5.2.10. Programul de modificare a fișierului de loturi șterge din fișierul de mașini loturile anulate și creează un fișier care consideră drept tranzitorii toate loturile rămase valabile din

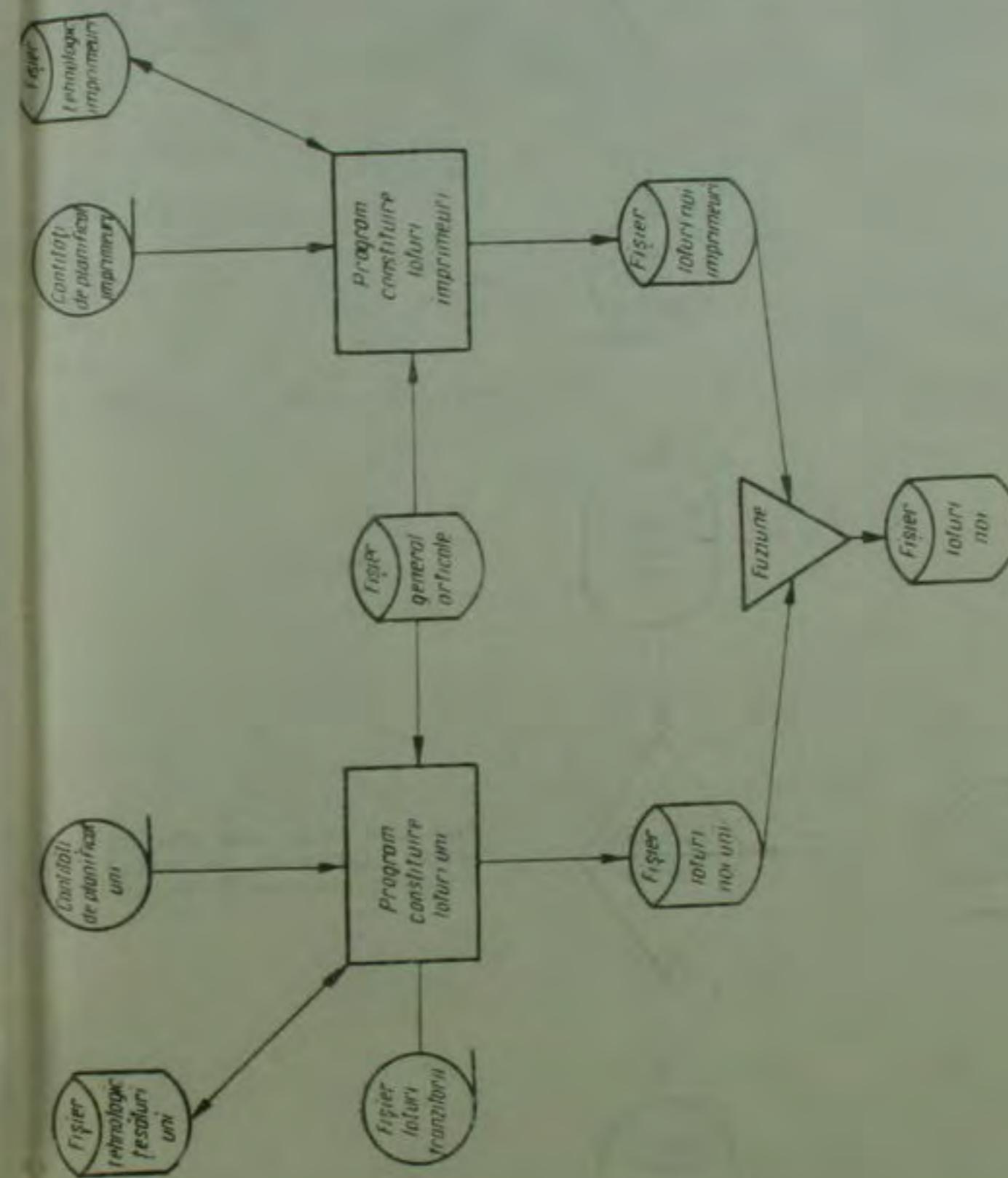


Fig. 5.2.7. Constituirea loturilor noi pentru plan.

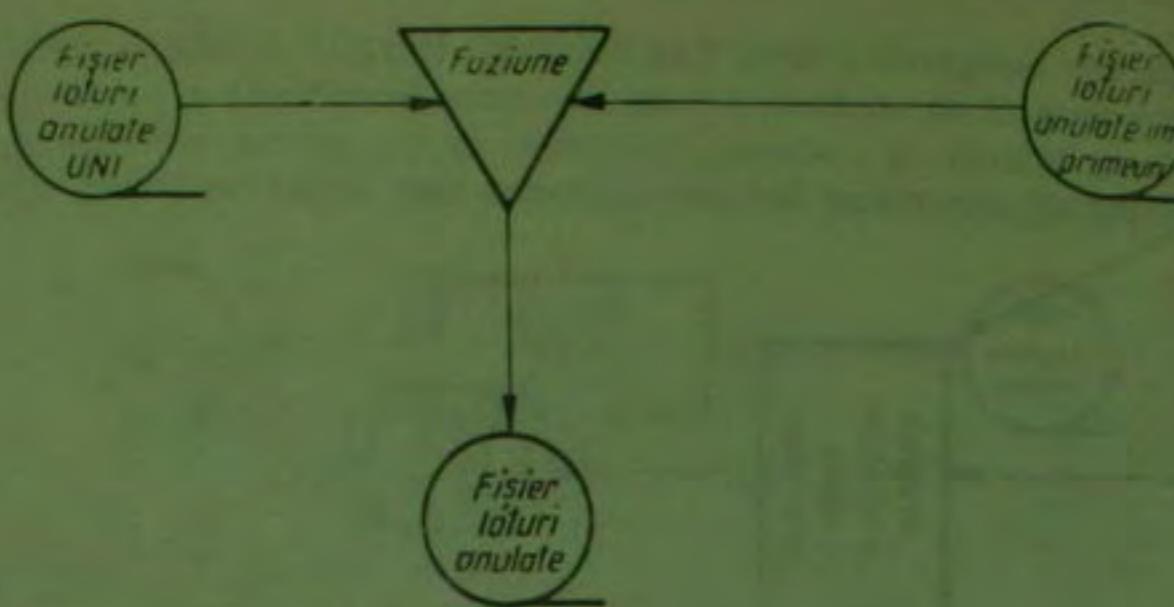


Fig. 5.2.8. Programe de interclasare pentru obținerea fișierelor de loturi anulate.

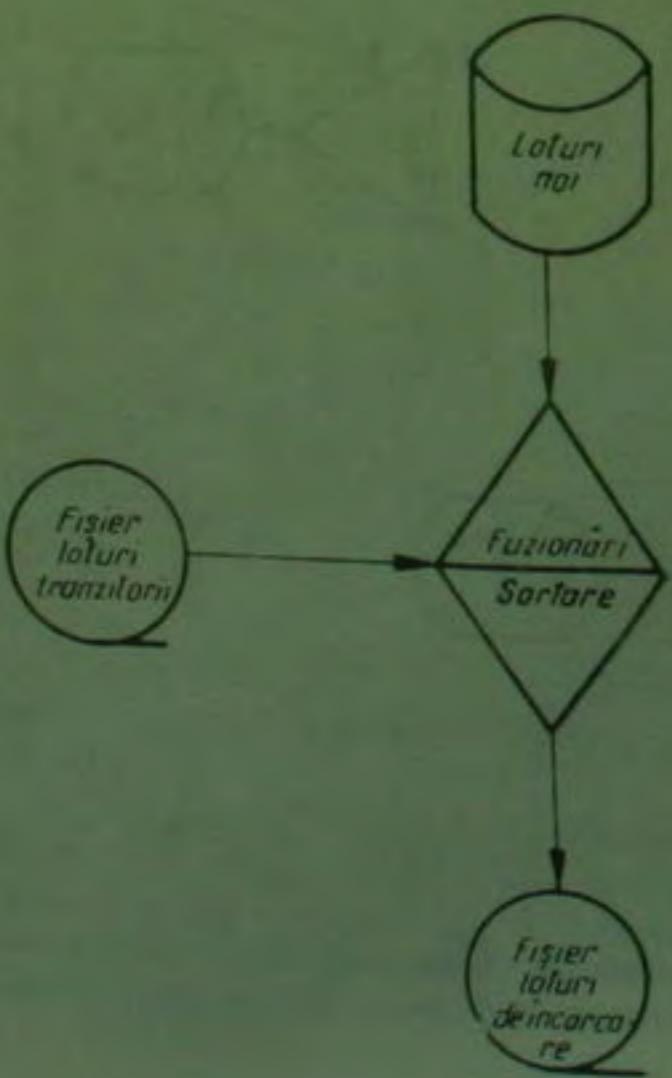


Fig. 5.2.9. Stabilirea loturilor pentru încărcarea pe mașini printr-un program de sortare.

planificarea veche. Loturile ce se adaugă sunt introduse apoi pe linia normală de prelucrare la poziția FIŞIER LOTURI NOI.

După obținerea fișierului de loturi noi, înainte de trecerea la încărcarea utilajelor se face controlul stocului de ţesătură crudă existent în magazie.

Se marchează cu prioritatea cea mai mare de încărcare, acele loturi pentru care există ţesătură crudă în magazie, apoi loturile pentru care urmează să

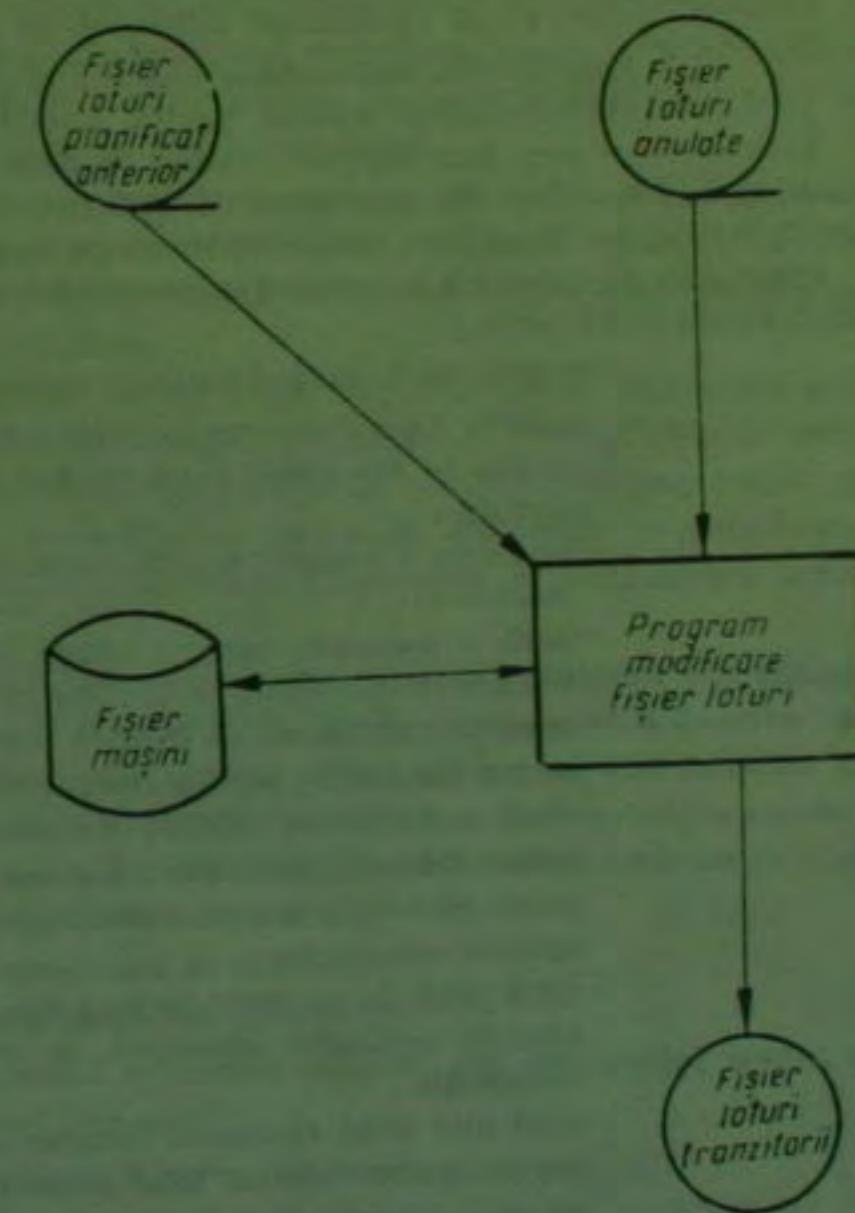


Fig. 5.2.10. Program pentru modificarea fișierului de loturi.

se primească marfă de la furnizori în intervalul de încărcare și la urmă se încarcă acele loturi pentru care nu există marfă în magazie și nici nu urmează să fie primită cu certitudine de la furnizori în intervalul de încărcare.

f. Încărcarea loturilor pe mașini. Încărcarea loturilor pe mașini este precedată de aducerea la zi a fișierelor de mașini și loturi, prin :

- ștergerea loturilor încărcate în fișierul de mașini pe perioada de plan (data de replanificare se furnizează pe o cartelă separată) ;
- ștergerea din fișierul de mașini a loturilor în curs de fabricație, dar numai din punctul la care au ajuns la momentul replanificării (după consultarea fișierului de urmărire a realizărilor) ;
- introducerea informațiilor indicind ultima mașină și data la care s-a înregistrat aceasta, pentru loturile în curs de fabricație ;

In acest fel se constituie fișierul cu loturi de încărcat de la data de replanificare, care este utilizat de programul de încărcare a loturilor pe mașini (figura 5.2.11). Acest fișier este sortat (ordonat) pe termene și culori.

Programul de încărcare a loturilor pe mașini lucrează după principiul ilustrat în figura 5.2.12, astfel :

- se introduc la încărcat loturile în ordinea termenelor de livrare ;
- se caută în fișierul de mașini primul tip din mașinile cerute de tehnologia de fabricație. Dacă au fost găsite mai multe mașini de acest tip :
 - se caută o mașină la care ultimul lot încărcat are aceeași culoare ;
 - dacă o asemenea mașină există, se compară termenul de livrare cu termenul posibil de executare al operației pe această mașină și se încarcă lotul la mașina respectivă numai dacă acest termen este inferior termenului de livrare ;
 - dacă o asemenea mașină nu există, se încarcă lotul pe prima mașină liberă (fără nici un lot încărcat de la începutul perioadei) sau pe o altă mașină pentru care timpul de spălare este minim ;
 - dacă lotul nu se poate încărca din lipsă de mașini disponibile în perioada respectivă, se încarcă alte variante de tehnologie ;
 - cînd nici după epuizarea tuturor variantelor de tehnologie nu s-a putut încărca, lotul respectiv este marcat în fișier ca neîncărcabil și se trece la lotul următor ;
- se repetă procedura pentru toate mașinile prevăzute în tehnologia de fabricație a sortimentului respectiv ;
- se repetă toate operațiile pentru toate loturile din fișierul cu loturile de încărcat.

După epuizarea tuturor înregistrărilor din fișierul cu loturile de încărcat, se obține în fișierul de mașini planul de activitate al fiecărei mașini în parte, pe zile, ore și minute (activități productive cu precizarea loturilor, reparații planificate și întreținere curentă, spălări și stagnări). Pe o bandă separată se obține un fișier ce cuprinde toate loturile încărcate cu succesiunea mașinilor pe care s-au planificat, precum și eventualele loturi ce nu au putut fi încărcate.

După ce loturile au fost încărcate pe utilaje și au fost de fapt determinate loturile care se lucrează, se face controlul stocului de coloranți, chimicale și materiale, necesar în procesul de fabricație, ținindu-se seama și de cantitățile care urmează să fie consumate pentru loturile existente în producție, pînă la încărcarea pe mașini a loturilor noi care se vor lansa.

Se obține o listă în care se arată ce cantități de coloranți, chimicale și materiale urmează să fie procurate în intervalul de timp (completarea stocului în magazie). Aceste cantități reprezintă de fapt planul de aprovizionare pe intervalul în care s-a făcut lansarea.

g. Constituirea fișierului de arhivă loturi. Un program de comparare analizează pe rînd înregistrările din fișierul cu loturile de încărcat (loturile tranzitorii și cele noi) și înregistrările din fișierul cu loturile încărcate, creând un fișier în care sunt arhivate toate loturile tranzitorii, planificate și neîncărcate, acestea din urmă fiind luate în considerație în perioada de replanificare următoare (figura 5.2.13).

h. Constituirea partizilor. Din fișierul de arhivare a loturilor sunt extrase loturile neintroduse în partizi și grupate pe datele de începere a loturilor pe zile. Consultînd fișierul tehnologic pentru partizi în care se află criteriile de combinare a loturilor în partizi, se constituie partizile avînd cantitățile maxime fixate de tehnologie (figura 5.2.14). Criteriile de combinare a loturilor în partizi sunt următoarele (în ordine preferențială) :

- loturi cu aceeași dată de terminare ;
- loturi din aceeași poziție coloristică ;
- loturi din poziții coloristice diferite, dar din același articol și avînd aceleași tehnologii ;
- loturi din articole diferite avînd aceleași tehnologii.

In cazul replanificărilor în cursul trimestrului, datorită modificărilor în planul de livrări, programul de selectare a loturilor consultă și fișierul de arhivare partizi, începînd cu partida de la care se replanifică, construind un fișier de lucru suplimentar cu partizile executate pînă la data replanificării

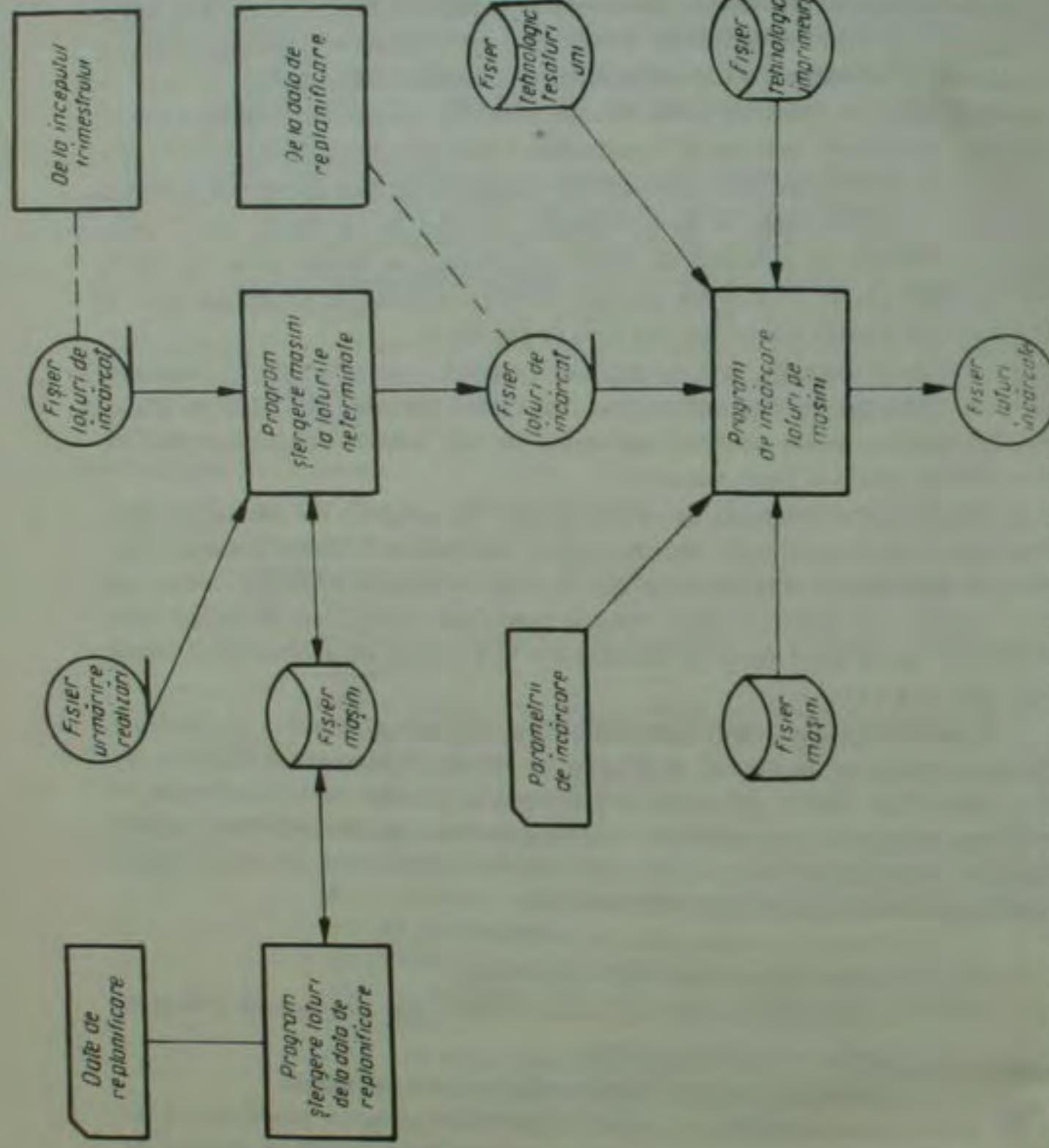


Fig. 5.2.11. Schema pentru Încărcarea loturilor pe mașini.

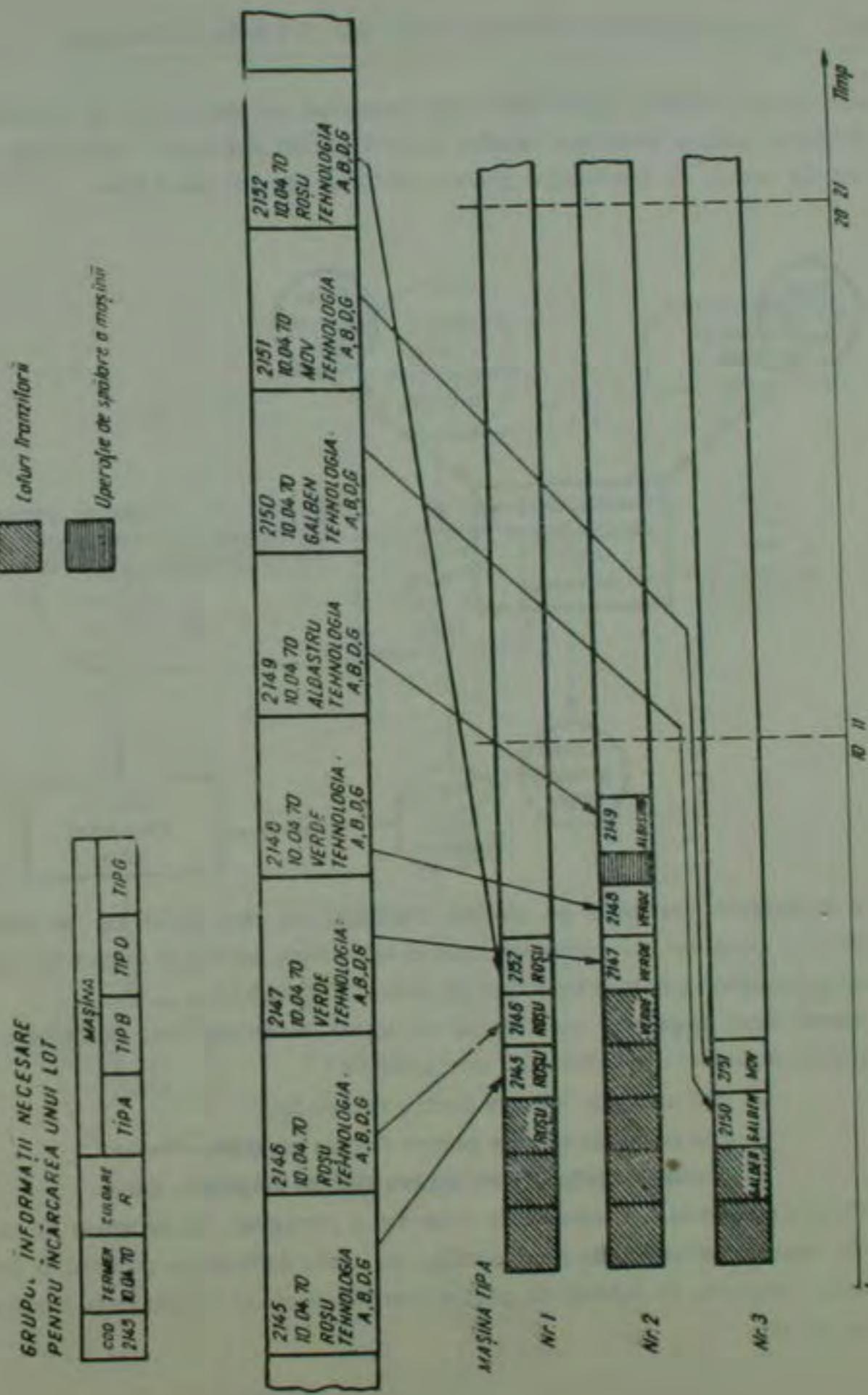


Fig. 5.2.12. Diagrama principiului de încărcare a loturilor pe mașini.

(fișier resturi partizi). Acest fișier este consultat de programul de constituire a partizilor pentru stabilirea codului ultimei partizi executate, în vederea atribuirii de coduri în continuare pentru partizile noi (figura 5.2.15.).

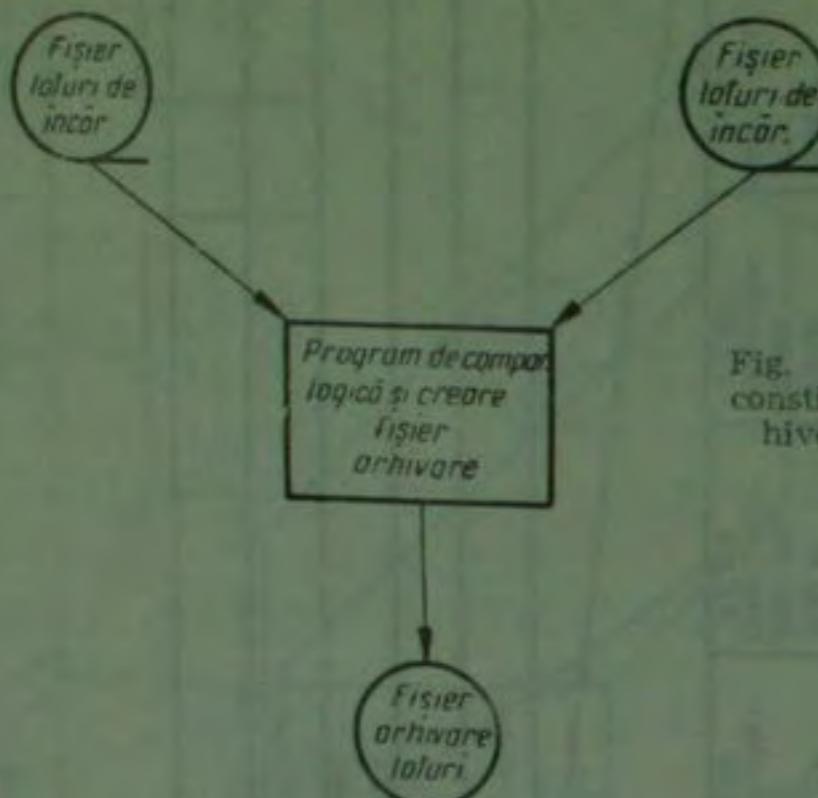


Fig. 5.2.13. Program pentru constituirea fișierului cu arhivele loturilor planificate.

1. Încărcarea partizilor pe mașini. Partizile noi sunt încărcate pe mașini de către un program care consultă fișierul tehnologie partizi și crează un fișier de lucru conținând partizile încărcate pe mașini (figura 5.2.16).

Acest fișier împreună cu fișierul de loturi încărcate, sunt consultate de programul de lansare în fabricație, care produce :

- cite o fișă de lansare pentru fiecare lot ;
- cite o fișă de lansare pentru fiecare partidă ;
- planurile de încărcare pentru fiecare mașină în parte.

Pentru constituirea fișierului de arhivare a partizilor, un program de comparare logică analizează în mod similar cu acela indicat la punctul b înregistrările conținute în fișierul de partizi tranzitcrii și cel cu partizile încărcate (figura 5.2.18).

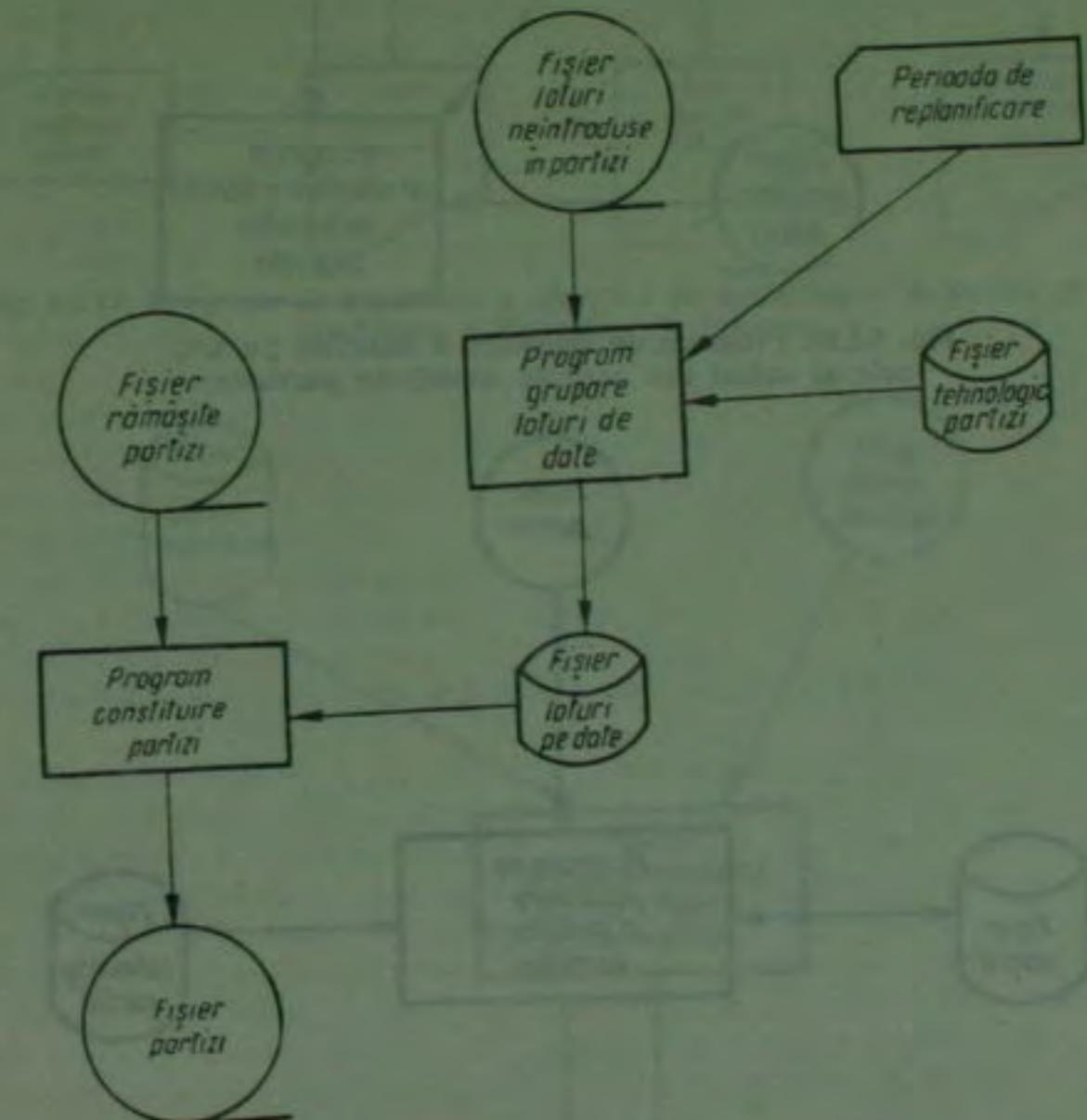


Fig. 5.2.14. Schema constituirii partizilor.

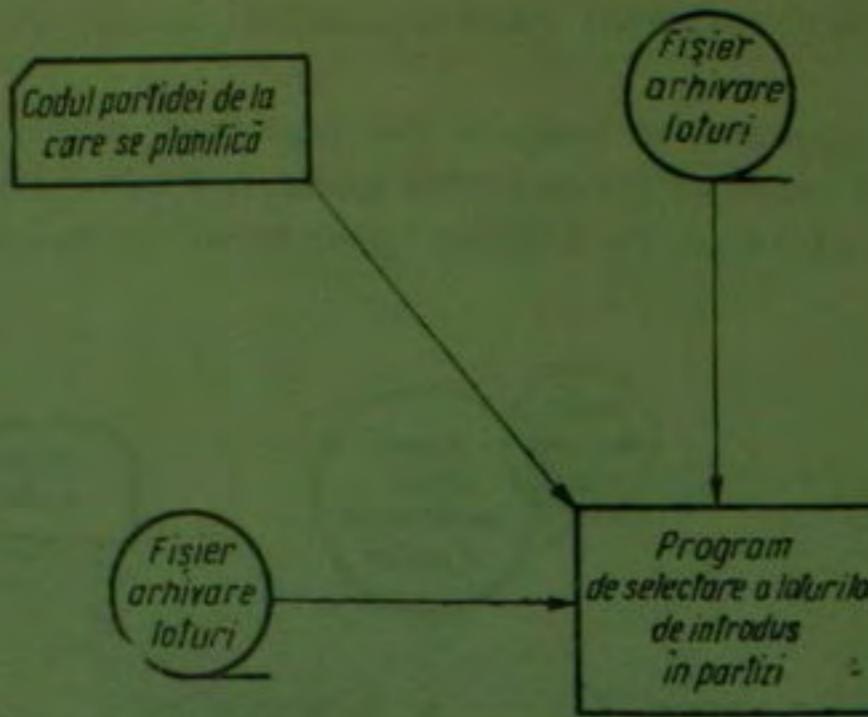


Fig. 5.2.15. Program de selecțare a loturilor pe articole și culori din care se constituie partizile.

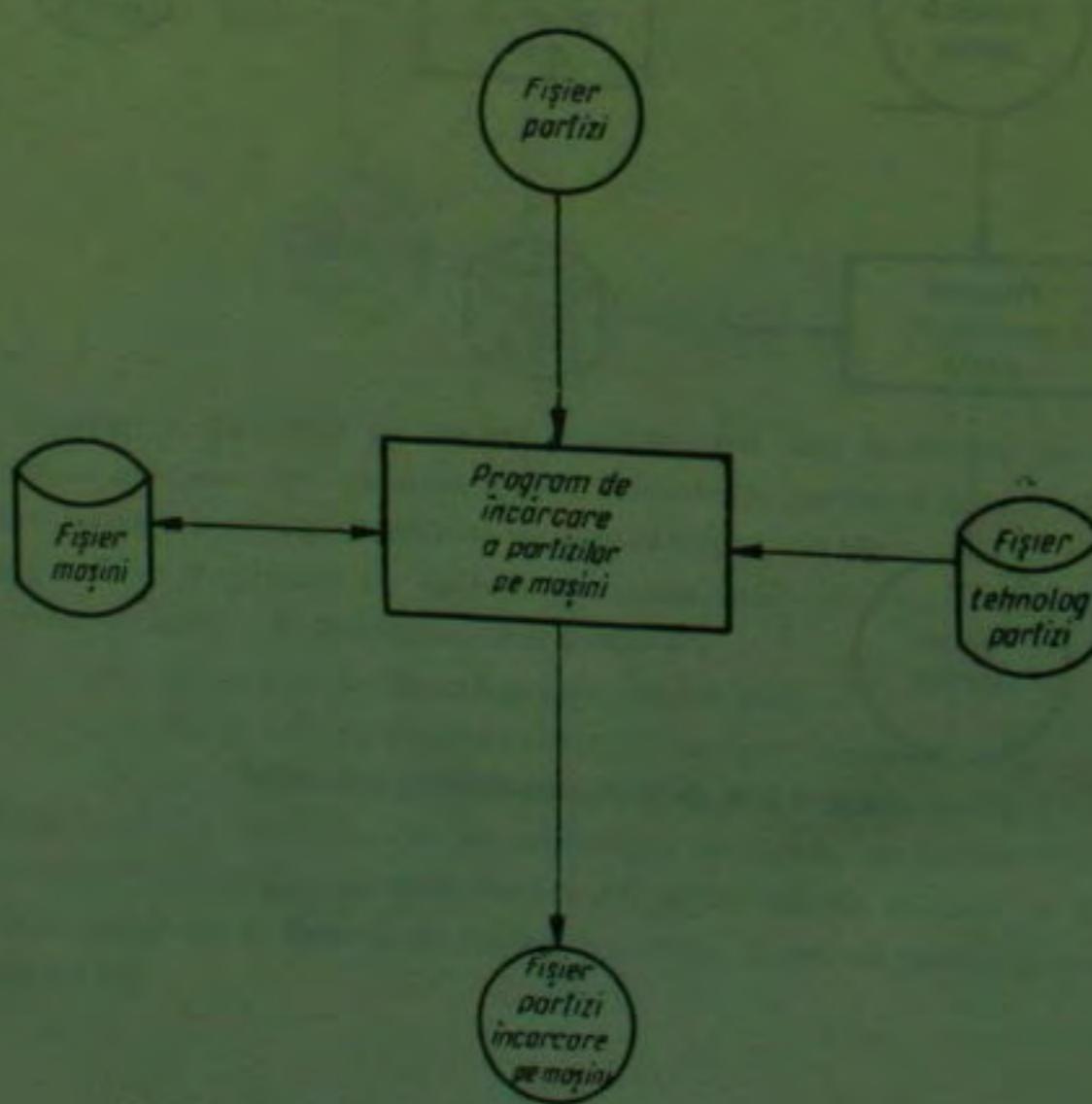


Fig. 5.2.16. Program de încărcare a partizilor pe mașini.

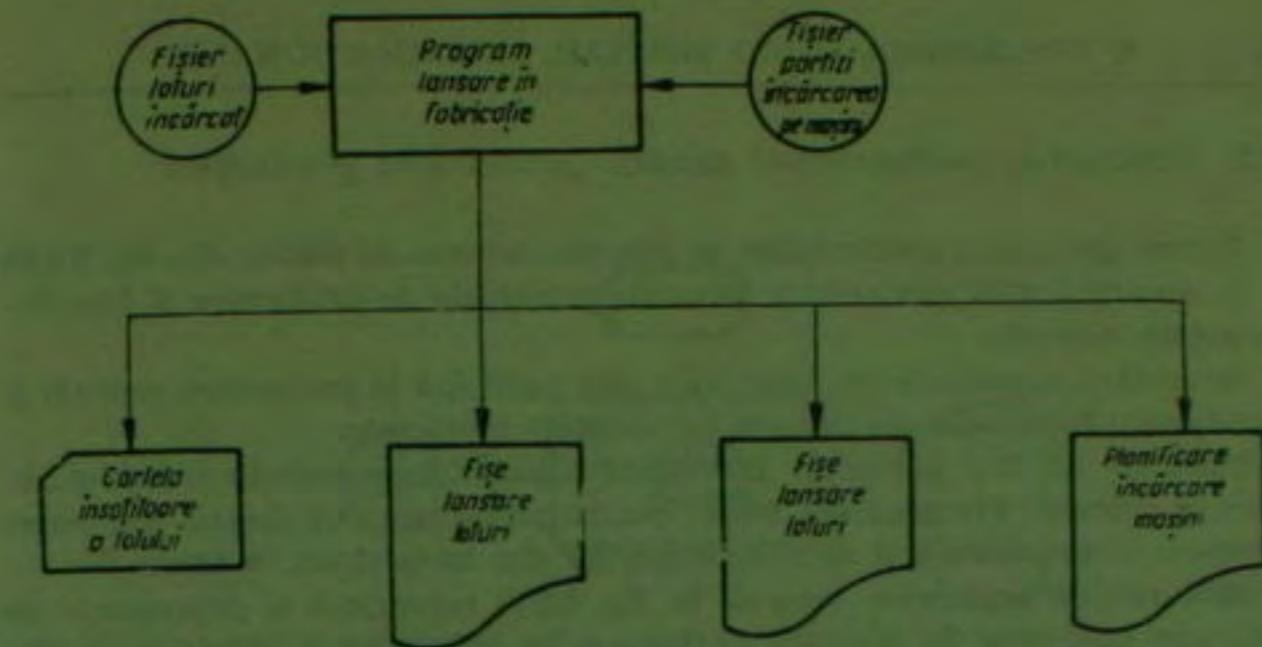


Fig. 5.2.17. Program de raportare a planului de încărcare a mașinilor și de producție a fișierelor de lansare.

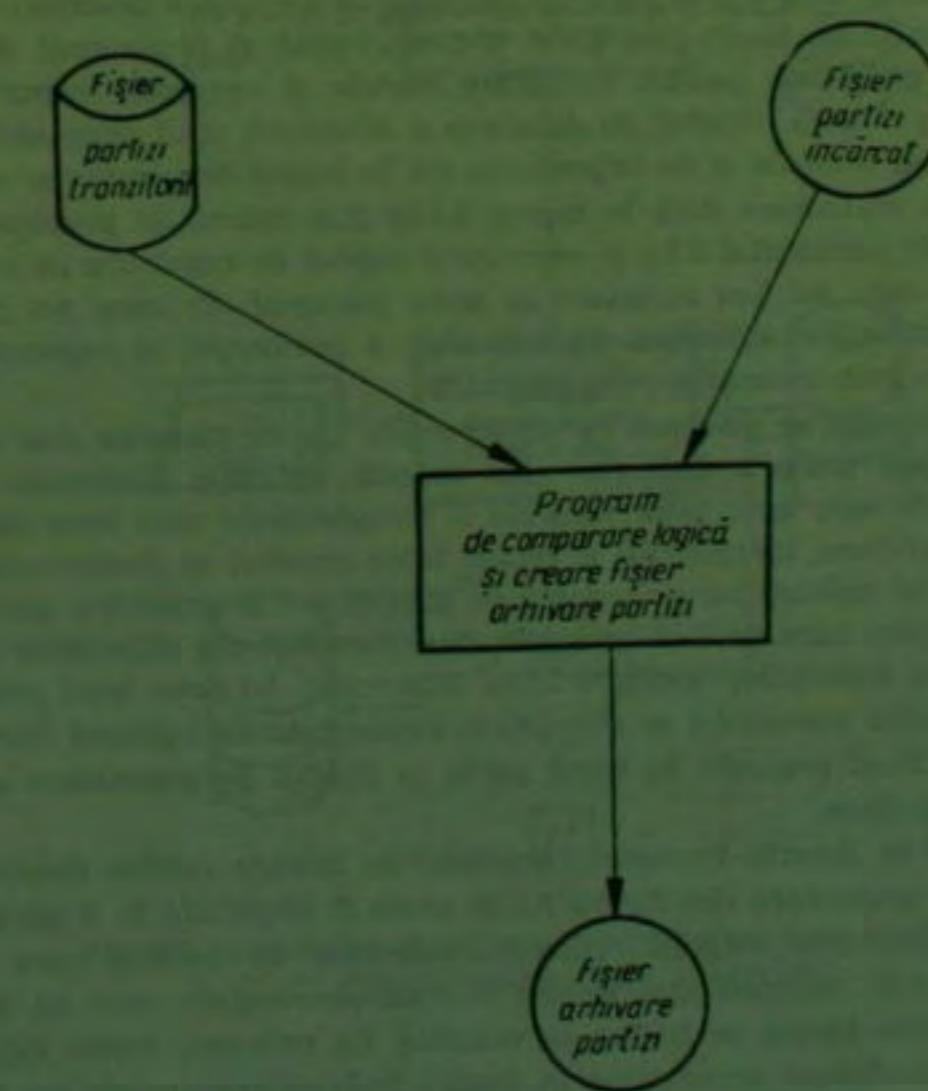


Fig. 5.2.18. Program de constituire a fișierului cu arhivele partizilor.

5.2.5. Descrierea prelucrărilor pentru „urmărirea producției”

Pentru descrierea prelucrărilor se folosește schema de sistem din fig. 5.2.19.

În această figură sunt arătate blocurile principale de prelucrare și funcțiile acestor blocuri.

Se arată și suporturile de informație care participă la prelucrare, precum și fișierele care furnizează sau primesc informațiile prelucrate.

Blocurile au fost denumite DUPXXM unde XX reprezintă numărul de ordine al blocului. Fiecare fișier folosit pentru prelucrare este descris din punct de vedere al structurii și a naturii cimpurilor din înregistrări în anexa 3.

Blocurile de prelucrare descrise în fig. 5.2.19 reprezintă și programele de prelucrare care apar în procesul de trecere pe calculator a acestei aplicații.

După descrierea generală a blocurilor de prelucrare și a funcțiunilor fiecărui bloc în parte, urmează să fie făcută descrierea fiecărui modul de program în parte. În cele ce urmează se dău trei exemple de asemenea descrieri și anume pentru programele de verificare, DUP 5M, DUP86M și programul de listare (5.2.6). Aceste programe pentru verificare cartele și respectiv raportare sunt descrise mai în detaliu. Gradul de detaliere a descrierii unui program depinde de complexitatea acestuia și de importanța lui în lanțul de programe respectiv.

Schema de prelucrare dată în figura 5.2.19 ține seama de problemele majore enunțate în paragraful 5.2.1 și reprezintă modul de rezolvare pe calculator a problemelor care au fost enunțate în acest paragraf (în care s-a prezentat descrierea funcțiunilor aplicației de urmărire a producției și rapoartele care trebuie obținute prin rularea acestei aplicații).

Această aplicație se prezintă ca un exemplu cu un caracter mai mult didactic; mulțimea mare a fișierelor din această aplicație ilustrează de fapt relațiile multiple care apar între fișierele și înregistrările unei baze de date ale cărei conținut (fișiere, înregistrări și relații între acestea) se proiectează tocmai pe baza studiului aplicațiilor principale de planificare și urmărire, prin constituirea unor fișiere care unifică cerințele de informații ale diferitelor aplicații și sistematizarea legăturilor necesare între informații. În acest mod programele necesare diferitelor prelucrări se simplifică, elementele de legătură dintre diferențele aplicațiilor fiind preluate în bună parte în modul de organizare și actualizare a bazei de date.

În anexa 2 se descrie formatul cartelelor de intrare pentru datele culese. Schema de prelucrare din figura 5.2.19 poate fi împărțită în 5 părți:

a. Prelucrările care asigură refacerea fișierelor de pe disc care au fost păstrate pe bandă. Acestea sunt fișiere indexat-secvențiale care au fost trăsute pe bandă sub formă de fișiere secvențiale. La refacere, aceste fișiere sunt transformate din fișiere secvențiale în fișiere indexat-secvențiale pe disc.

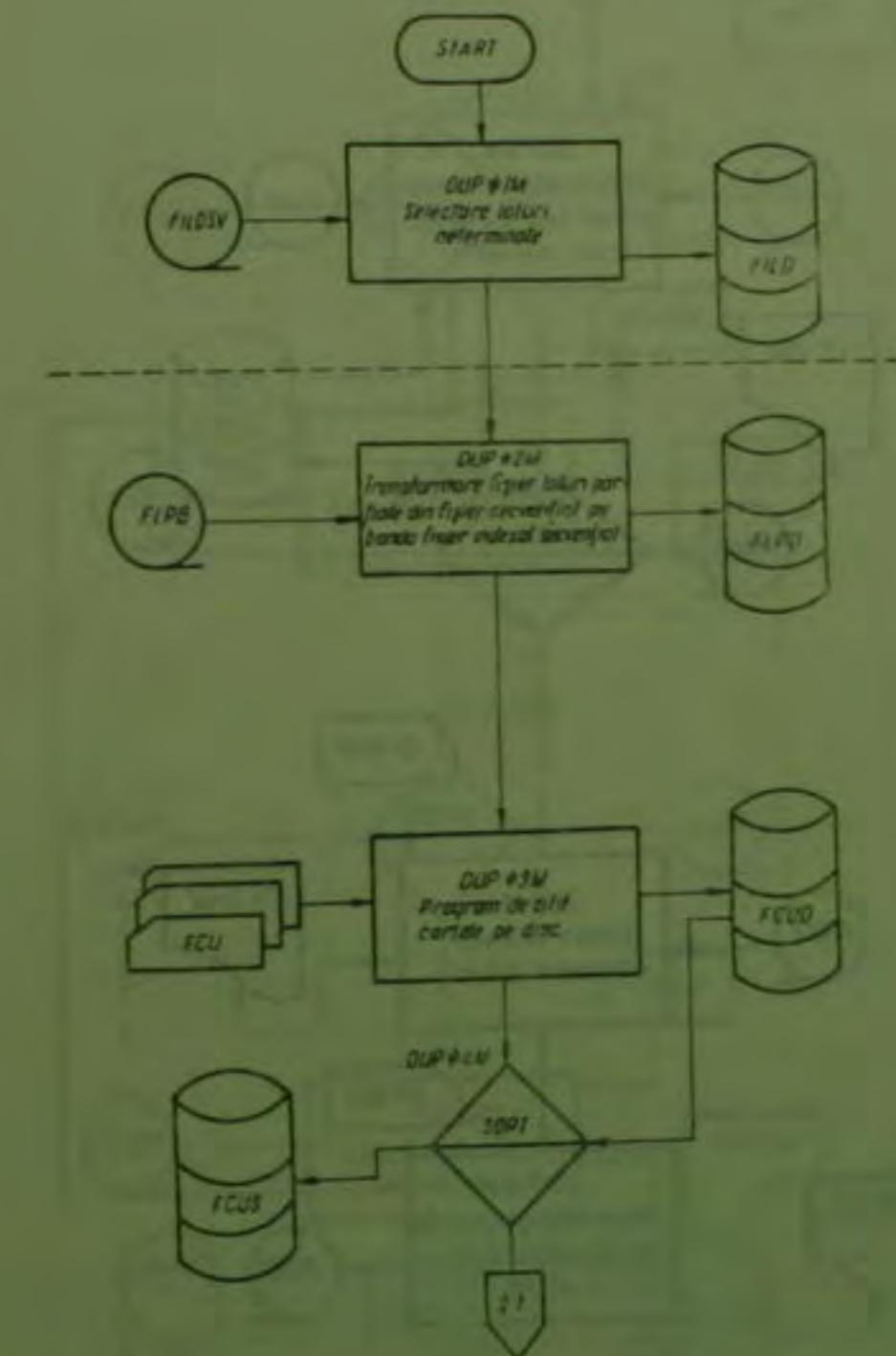


Fig. 5.2.19. Schema de sistem pentru urmărirea producției cu colectarea manuală a datelor.

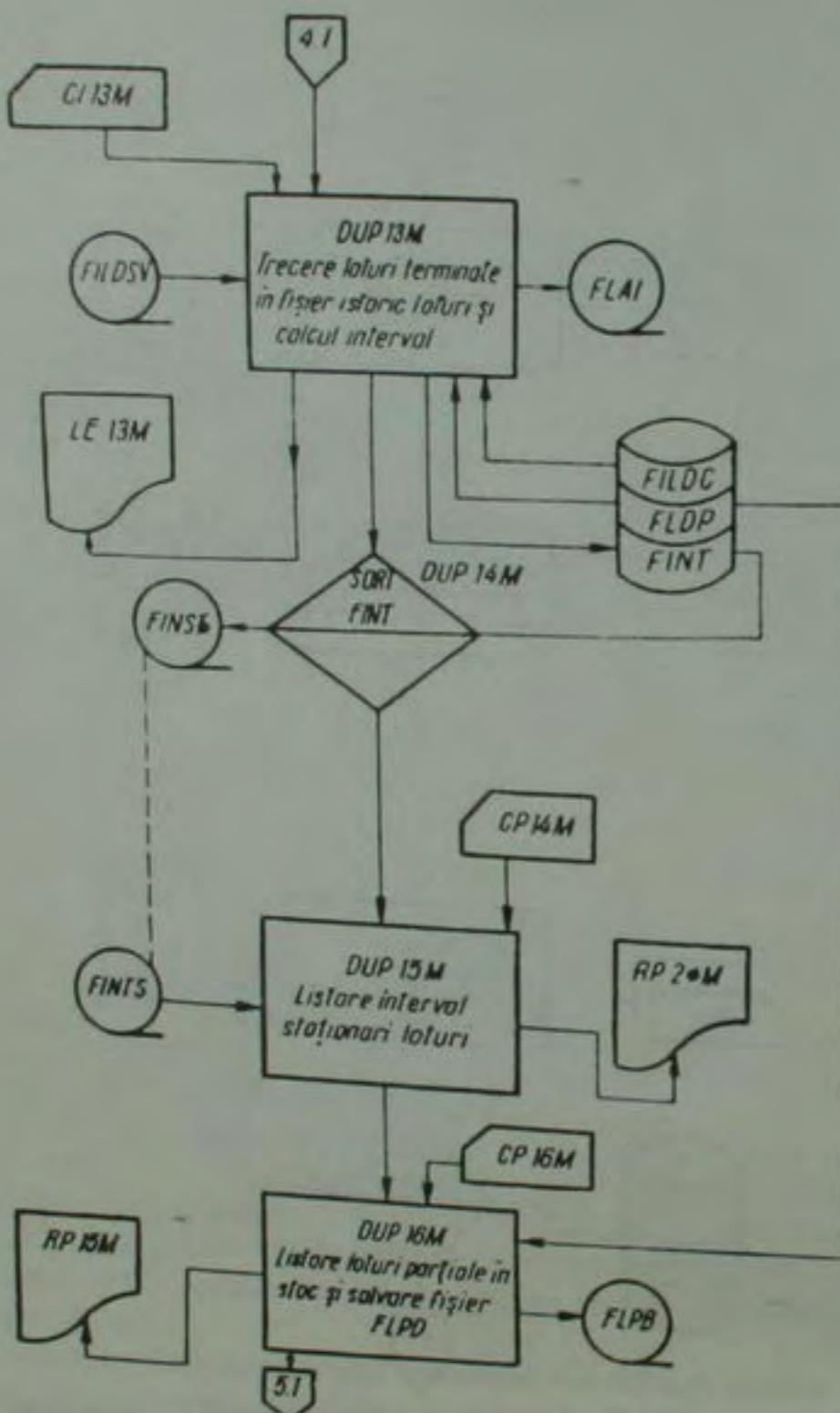


Fig. 5.2.19 (continuare)

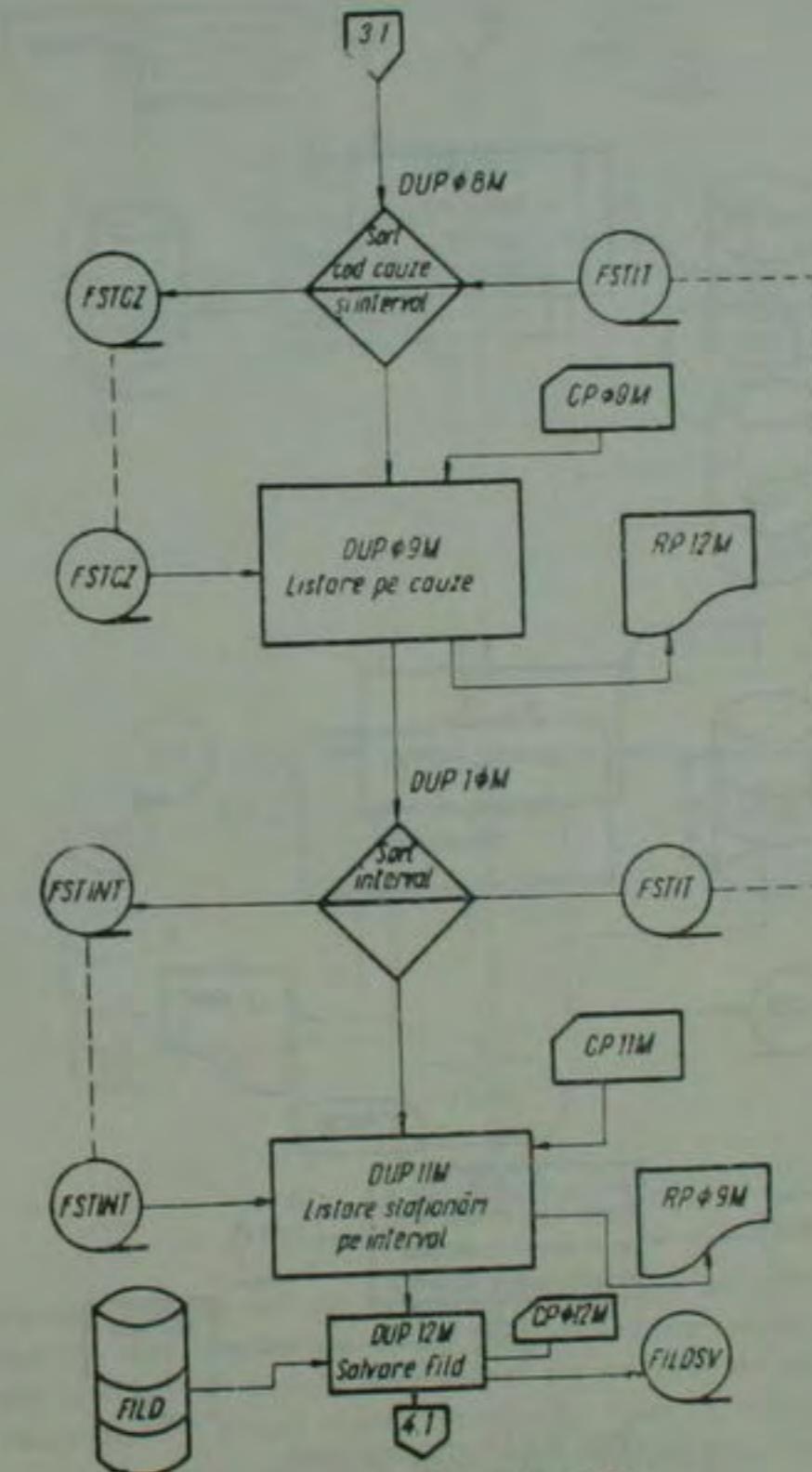


Fig. 5.2.19 (continuare)

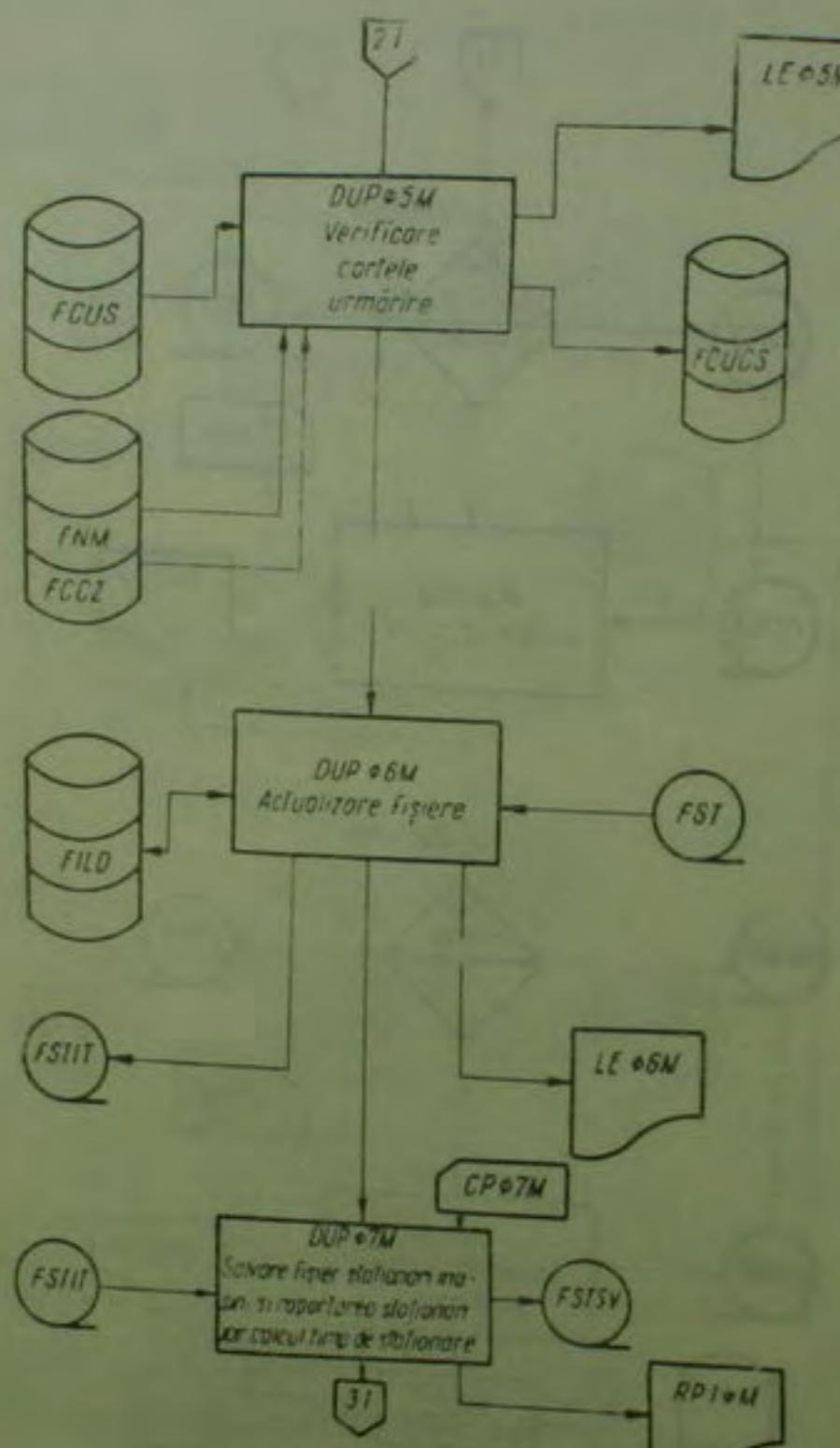


Fig. 5.2.19. (continuare).

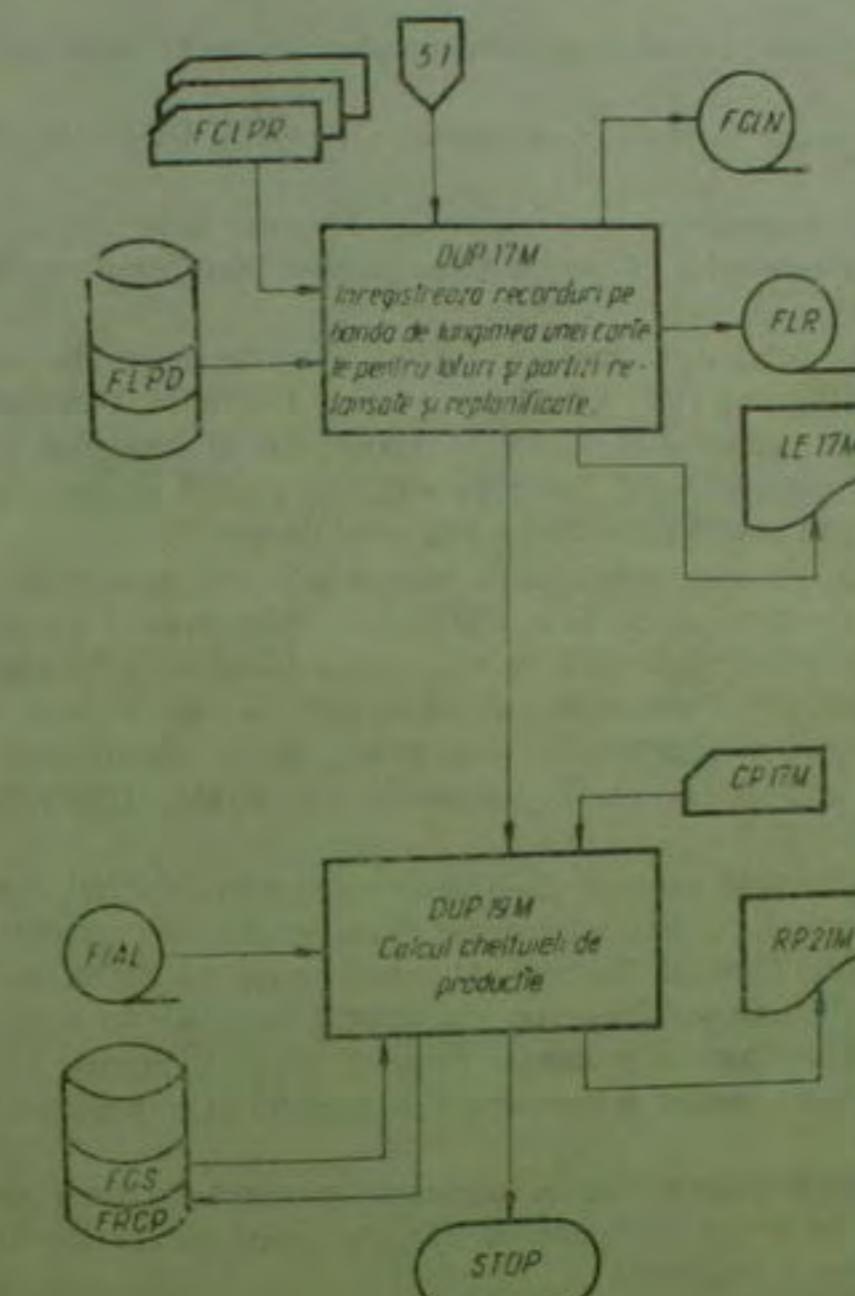


Fig. 5.2.19 (continuare)

In acest caz, trebuie reținut că înregistrările care au fost înscrise pe pistele de depășire sunt trecute pe bandă în ordinea lor normală, la refacerea fișierului ele aparind pe pistele normale și nu pe cele de depășire, deoarece la citirea secvențială a benzii aceste înregistrări apar în secvența lor normală.

Din această categorie fac parte programele DUP01M și DUP02M.

b. Prelucrările care asigură citirea și verificarea datelor de intrare și actualizare a fișierelor asociate acestor activități. În acest caz, se citesc car-

telele cu date perforate direct de pe documentele sursă pe care au fost culese datele din secții.

În această categorie intră programele DUP03M, DUP04M, DUP05M și DUP06M.

Acest grup de programare este important deoarece face verificarea datelor primare (sursă) și este știut că de calitatea acestor date depinde calitatea rezultatelor obținute.

c. Prelucrările privind raportarea intrărilor și ieșirilor din staționare a utilajelor. În această categorie intră programele DUP07M, DUP09M, DUP10M, DUP11M, care fac prelucrarea, scrierea rapoartelor și copierea fișierelor de staționari pentru mașini. Acest fișier de staționari fiind numai pe bandă nu necesită refacerea în sensul punctului a din acest paragraf.

d. Prelucrările privind circulația pe mașini a loturilor normale și parțiale, intrările în stoc ale loturilor parțiale, staționările între mașini ale loturilor normale, staționările loturilor parțiale și relansarea loturilor normale și parțiale și a partizilor care sunt considerate tot loturi dar cu cod rezervat. Aceste programe asigură și legătura cu programele de la planificarea producției.

Din această grupă fac parte programele DUP12M, DUP13M, DUP14M, DUP15M, DUP16M, DUP17M.

e. Prelucrări privind calculul cheltuielilor de producție (DUP18M).

Cu această grupă de programare, pe baza rezultatelor obținute de la urmărirea producției, folosind fișierele de istoric care se obțin, se poate face calculul cheltuielilor de producție pe un anumit interval de timp.

Calculul cheltuielilor de producție poate fi făcut și pentru loturile planificate, putând realiza astfel o comparare a cheltuielilor planificate cu cheltuielile realizate.

Pentru a putea analiza mai în amănunt prelucrările care se execută și rezultatele care se obțin, vom analiza fiecare grupă de prelucrări împreună cu fișierele pe care le utilizează :

a. În această grupă de prelucrări se folosesc fișierele FILDSV (Anexa 3.6) și FILD. (Anexa 3.6). Structura înregistrărilor celor două fișiere este identică diferind numai prin modul de acces. FILDSV este fișier secvențial pe bandă, iar FILD este fișier indexat secvențial pe disc.

Aceste fișiere sunt descrise în anexa 3.6.

În fișierul FILD se ține evidența loturilor normale și parțiale care sunt în execuție, dar n-au fost încă raportate ca terminate.

Acest fișier ține evidența mașinilor pe care trece un lot normal, a datei (an, lună, zi, oră și minut) la care lotul a ieșit de pe fiecare mașină, a schimburilor de lungime pe care le suportă un lot normal la ieșirea de pe o mașină, înregistrând atât lungimea lotului normal care continuă prelucrarea, cît

și a loturilor parțiale care intră în stoc intermediar în cursul prelucrărilor, dar numai atunci apar modificări de lungime față de mașina precedentă.

Tot în acest fișier se păstrează și situația lotului la ieșirea din atelierul de ajustaj unde lotul se împarte pe bucăți comerciale (marfa care se dă la beneficiar), cupoane capete și fâșii.

Se face controlul bilanțului la fiecare lot astfel incit eventualele abateri să fie analizate și rezolvate. Acest control se face în programul DUP13M.

În FILD se găsesc atât loturi care au fost declarate ca fabricate, urmând a fi trecute în fișierul de istoric, cit și loturi care sunt în execuție.

În FILDSV se pot găsi toate loturile din FILD sau numai o parte, și anume aceleia care nu au fost raportate ca terminate fie că sunt încă în execuție, fie că nu au bilanțul încheiat. FILDSV reprezintă fișierul de rezervă pentru FILD.

În programul DUP 0 2M se reface fișierul de loturi parțiale pe disc.

Fișierul FLPD (fișier de loturi parțiale pe disc) este indexat secvențial și este descris în anexa 3.5. Înregistrările din acest fișier au drept cheie de identificare codul lotului de proveniență și secvența lotului parțial.

În înregistrările din acest fișier se mai rețin : data intrării în stoc (an, lună, zi, oră, minut) care de fapt reprezintă data ieșirii de pe ultima mașină înainte de a se face divizarea lotului normal, articolul și poziția coloristică la intrarea în staționare, lungimea, codul lotului pentru calculator, poziția coloristică nouă (dacă aceasta se schimbă), data relansării.

Fișierul FLPB (fișier loturi parțiale pe bandă) are aceeași structură a înregistrărilor ca fișierul FLPD (Anexa 3.5).

b. În această grupă de prelucrări se folosesc fișierele FCU și FCUD pentru programul DUP 0 3M.

Fișierul FCU (Anexele 2.1, 2.2, 2.3, 2.4, 2.5.1, 2.5.2) este format din cele 5 tipuri de cartele cu date primare care se folosesc la urmărirea producției având codurile /05\, /06\, /07\, /08\, și /09\.

ACESTE CARTELE SINT CITITE PE DISC ÎN FIȘIERUL FCUD CARE DE FAPT REPREZINTĂ O COPIE A FIȘIERULUI FCU.

Fișierul FCUD (Anexele 2.1, 2.2, 2.3, 2.4, 2.5.1, 2.5.2) are înregistrări logice în lungime de 80 de caractere ca și fișierul FCU.

Programul DUP 0 4M face sortarea fișierului FCUD obținându-se fișierul sortat FCUS (Anexele 2.1, 2.2, 2.3, 2.4, 2.5.1, 2.5.2).

Sortarea se face după caracterele 01 ÷ 02, 09 ÷ 11, 03 ÷ 08 din FCUD deci după codul cartelei, apoi după codul mașinii și apoi după dată astfel început în final se obțin cartelele grupate după cod, adică la început toate cartelele cu codul /05\, apoi toate cartelele cu cod /06\ și.m.d., în cadrul fiecărui grup

de cartele apărind cartelele pentru o mașină cu un cod anume, indiferent de data raportării și în cadrul fiecărei mașini apărind cartelele sortate pe dată.

Programul DUP05M ce va fi descris în detaliu în paragraful 5.2.6 face verificare sintactică a acestor cartele conform descrierilor făcute pentru fiecare grup de cartele în acest paragraf.

Acest program utilizează fișierele FCUS (descrise la programul DUP 0 4M), FNM un fișier indexat secvențial cu numele mașinilor, cu care se face verificarea dacă mașina cu codul dat în caracterele 0—11 se găsește în acest fișier (dacă mașina nu se găsește se tipărește mesaj de eroare), FCCZ (Anexa 3.2), fișier care ține un text ce explică în clar ce înseamnă fiecare cod al cauzei de intrerupere care se raportează (dacă cauza cu codul dat nu există în fișier înseamnă că codul cauzei este greșit și se tipărește mesaj de eroare), fișierul FCUCS care are aceeași structură a înregistrărilor ca și FCUS descris mai sus la programul DUP 0 4M, dar conține numai acele înregistrări care dă rezultate corecte în urma verificării sintactice și LE05M care reprezintă raportul cu înregistrările care au erori sintactice conform descrierii făcute la programul DUP 0 5M.

Programul DUP 0 6M face actualizarea fișierelor FIELD descris la a. și FST fișier de staționari mașini (descris în anexa 3.3). Fișierul FST conține codul mașinii, data intrării în staționare, data ieșirii din staționare (ambele sub formă an, lună, zi, oră și minut) codul cauzei intreruperii și intervalul de intrerupere dat în minute.

Fișierul actualizat FSTIT (anexa 3.3) este identic cu fișierul FST din punct de vedere al structurii înregistrărilor, conținând în plus actualizările care se fac prin programul DUP 0 6M ce va fi descris în detaliu în paragraful 5.2.6.

c. În această grupă sunt programele care execută prelucrările privitoare la staționările utilajelor.

Programul DUP 0 7M are 3 funcții importante :

- copiază fișierul de staționari FSTIT pe o bandă obținând fișierul FSTSVD (anexa 3.3) identic ca structură și conținut cu fișierul FSTIT ;
- raportează mașinile care există actualmente în staționare ;
- calculează timpul de staționare pentru acele înregistrări pentru care acesta nu a fost calculat.

Alegerea unei sau mai multor variante dintre cele descrise mai sus se face prin cartela parametru CP 0 7M în care se inscrie care anume din funcțiile arătate se vor executa.

În orice caz, executarea oricărui din punctele arătate mai sus implică citirea secvențială a fișierului FSTIT de la început până la sfîrșit.

In programul DUP 0 8M se face o sortare a fișierului FSTIT după codul cauzelor, după intervalul de timp și codul mașinilor, obținându-se astfel fișierul FSTCA (anexa 3.3) care este identic ca structură a înregistrărilor cu fișierul FSTIT, având însă înregistrările aranjate în altă ordine.

In programul DUP 0 9M se face scrierea rapoartelor (RP12M) pe cod cauze, folosind fișierul FSTCZ. Se folosește cartela parametru CP 0 9M care indică intervalul de raportare (dată de început și de sfîrșit, inclusiv limitele) și valoarea minimă a intervalului pentru care se face raportarea.

Prin programul DUP10 M se execută sortarea fișierului FSTIT după interval, cod mașină, cod cauză, obținându-se fișierul FSTINT (anexa 3.3) identic ca structură a înregistrărilor cu FSTIT, dar având înregistrările în altă ordine.

In programul DUP11M se face scrierea rapoartelor privind staționările pe interval, folosind fișierul FSTINT obținut la prelucrarea precedentă și utilizând cartela parametru CP11M care are aceleași funcții ca și cartela CP 0 9M, dând datele între care se face raportarea și valoarea minimă a intervalului de staționare pentru care se face această raportare.

Cu acest program se încheie descrierea lanțului de programe care face prelucrarea staționării utilajelor.

d. Prin programul DUP12M se face transcrierea fișierului de loturi normale și parțiale FIELD, folosind cartela CP12M care indică dacă se execută transcrierea (dacă nu se trece la programul următor) și dacă în fișierul FILDSV care este utilizat pe bandă se face trecerea tuturor înregistrărilor din FIELD sau numai a celor în care loturile sunt raportate ca neterminate. Fișierele FIELD și FILDSV au fost descrise la a.

Prin programul DUP13M se face înscrierea loturilor terminate în fișierul de istoric denumit FLAI care are o structură identică cu FIELD diferind numai ca acces. Loturile care sunt în execuție curentă se trec în fișierul FILDC, identic ca structură cu FIELD și cu acces indexat secvențial. Loturile parțiale rezultate din fișierul FILDSV se trec în fișierul FLPD, cunoscut de la prelucrările precedente. Se calculează de asemenei intervalul de timp între ieșirile de pe două mașini succesive și se inscrie în fișierul FINT.

Fișierul FINT (anexa 3.4) conține informațiile privind intervalul de timp între două ieșiri de pe două mașini succesive, codul lotului, codul mașinii precedente și data ieșirii de pe această mașină, codul mașinii următoare și data ieșirii de pe această mașină și valoarea intervalului de timp în minute. În fișierul FINT nu sunt înscrise toate intervalele de timp, ci numai aceleia care sunt mai mari decât intervalul dat în cartela parametru CP13M, care mai conține de asemenea și între ce date se calculează intervalele, care se înscriv în FINT.

In programul DUF14M se face sortarea fișierului FINT după cod lot și după intervalul de timp între două mașini. Se obține din sortare fișierul FINTS (anexa 3.4).

In programul DUP15M se face scrierea rapoartelor privitoare la staționarea loturilor între mașini conform cartelei parametru CP15M care dă intervalul de timp minim pînă la care se face raportarea și datele între care se fac raportările. Dacă lipsește cartela parametru se ia în considerație întregul fișier FINTS, conform cartelei parametru de la DUP13M. In cartela parametru se trece de asemenea și intervalul de timp minim pentru care loturile lansate care n-au intrat în fabricație se consideră în staționare.

DUP16M face listarea loturilor parțiale din stoc conform cartelei parametru CP16M care va specifica data de la care se începe raportarea loturilor parțiale și data la care se termină această raportare.

DUP17M este programul de legătură între aplicația de urmărire a producției și aplicația de planificare a producției.

Orice lansare în fabricație se face numai prin intermediul aplicației de planificare a producției.

In această aplicație codul de lot este stabilit între /02000/ și /99998/. Codurile între /02000/ și /19990/ sunt repartizate pentru partizi, iar codurile între /80000/ și /99998/ reprezintă coduri de loturi parțiale relansate.

Prin intermediul acestui program se face relansarea de partizi și de loturi și reincarcarea de loturi parțiale pe mașini conform procesului tehnologic al poziției coloristice care a fost stabilită pentru acel lot parțial care se reincarcă.

Loturile parțiale apar ca relansate, dar ca încărcate nu pe prima mașină din procesul tehnologic normal, ci pe o mașină oarecare ce va fi stabilită în secții.

Folosește următoarele fișiere:

FCLPR (anexa 3.8) fișier pe cartele conținînd loturi și partizi relansate.

In acest fișier se dau toate caracteristicile lotului sau partizii care se relansează privind codul, articolul, poziția coloristică veche, poziția coloristică nouă și lungimea.

Tot în această prelucrare se mai folosește și fișierul FLPD în care loturile parțiale care au fost lansate se marchează ca lansate pentru a fi ulterior trecute în fișierul de istoric pentru loturile parțiale.

Loturile și partizile relansate sunt trecute în FLR adică în fișierul de loturi relansate în care se dau toate caracteristicile lotului relansat, conform machetei fișierului FRL.

Loturile noi care se relansează, se înregistrează pe o bandă într-un fișier cu numele FCLN care înseamnă — fișier cartelo loturi noi, deoarece înregis-

tarea în acest fișier este de 80 de caractere, fișierul putind fi și pe cartela numai pe bandă. Acest fișier se referă la loturile noi care se relansează.

Erorile sunt scoase pe raportul LE17M și se referă în special la loturile parțiale lansate care nu sunt găsite în fișierul FLPD.

e. Ultimul program din lanțul de programe este DUP19M care face calculul cheltuielilor de producție.

Acest program utilizează fișierele : FLAI (anexa 3.6), care a fost descris mai sus, FCS, adică fișierul cu costuri specifice care indică costurile pe m.i. de articol, poziție coloristică și mașină. Rezultatele acestor calcule pe interval de timp și pe mașină sunt depuse în fișierul FRCP. Rezultatul calculului este tipărit pe raportul RP21M.

Intervalul de timp pentru care se face calculul cheltuielilor este dat sub forma : data zilei de început a intervalului, data zilei de sfîrșit a intervalului, inclusiv limitele prin cartela parametru CP17M.

Modul de utilizare a lanțului de programare descrise în schema de sistem din figura 5.2.19 este prezentat în figura 5.2.20.

In această figură se arată modul de prelucrare sub formă unei scheme logice de programare.

Organizarea lanțului de programe sub această formă depinde de tipul calculatorului pe care se face prelucrarea.

Sistemul de operare este acela care dictează în ultimă instanță modul de realizare a acestei scheme logice de programare.

In general, indiferent de tipul calculatorului, programatorul poate realiza o asemenea rulare a lanțului de programare, fie cu ajutorul unor programe făcute de el, fie cu ajutorul unor programe ale sistemului de operare, fie chiar prin intervenția operatorului.

Aplicația de urmărire a producției cu culegere manuală = datelor ar trebui rulată de obicei din 3 în 3 zile sau săptămânal.

Cartelele de corecții sunt identice cu cele de tranzacții și sunt verificate și rulate odată cu acestea, având codurile /96/ pentru corecțiile la cartelele cu cod /06/, /97/ pentru corecțiile la cartelele cu cod /07/, /98/ pentru corecții la cartelele cu cod /08/ și /93/ pentru corecții la cartelele cu cod /09/.

Rapoartele, formatele înregistrărilor, descrierea fișierelor și a bazelor de date schemele de sistem, descrierea prelucrărilor ce urmează să fie programate la un nivel de detaliere recomandabil ca în exemplele din paragraful 5.2.6, ce urmează, cu ajutorul simbolurilor introduse în cap. IV și a schemelor logice de programare, sunt elementele constitutive principale ale dosarului de analiză, care împreună cu programele, scrise în limbajul de programare în dosarul de programare și cu dosarul de operare și instrucțiunile de implementare și exploatare, alcătuiesc piesele principale ale proiectului noului sistem informațional.

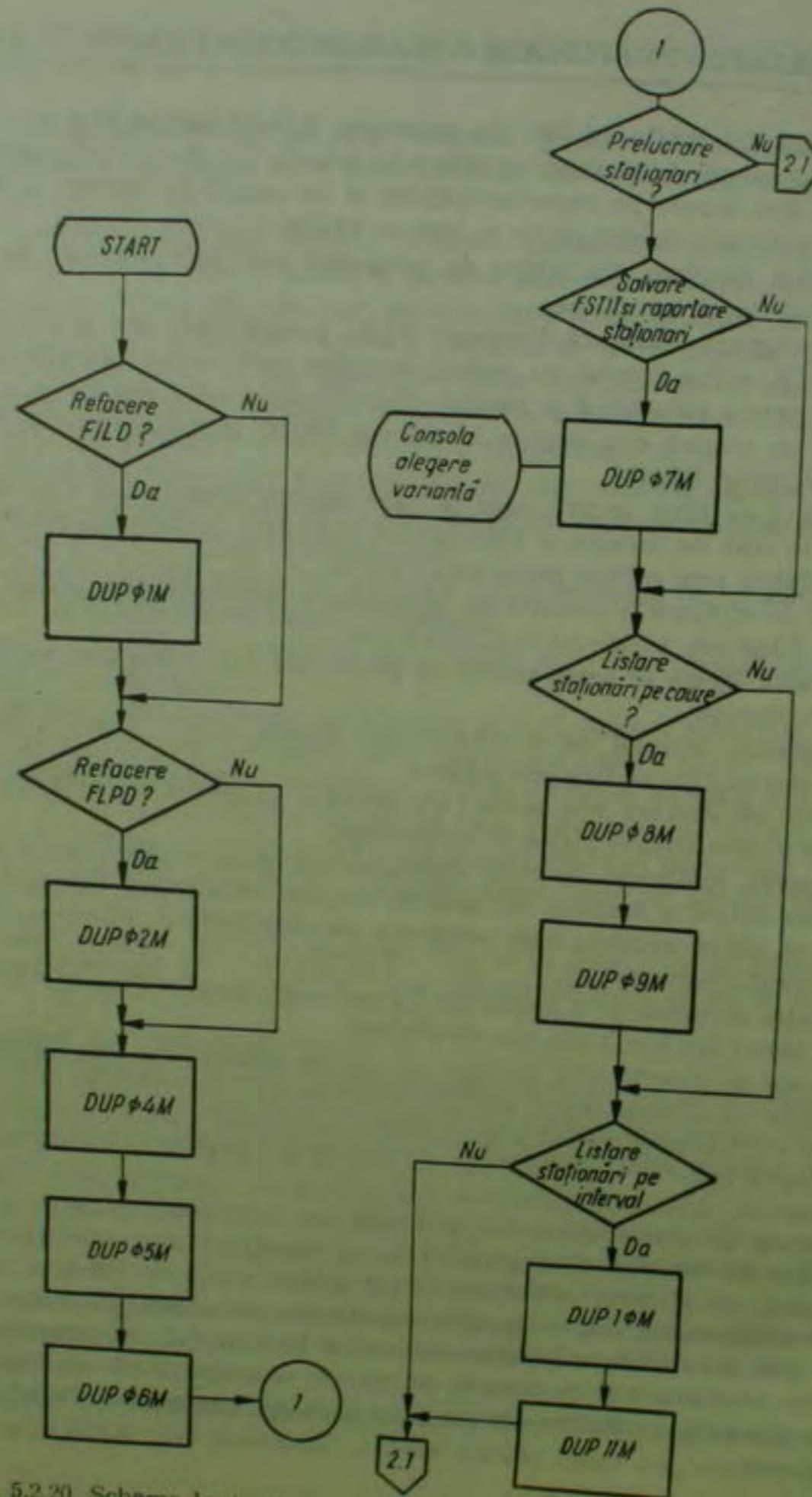


Fig. 5.2.20. Schema lanțurilor de programare pentru urmărirea producției.

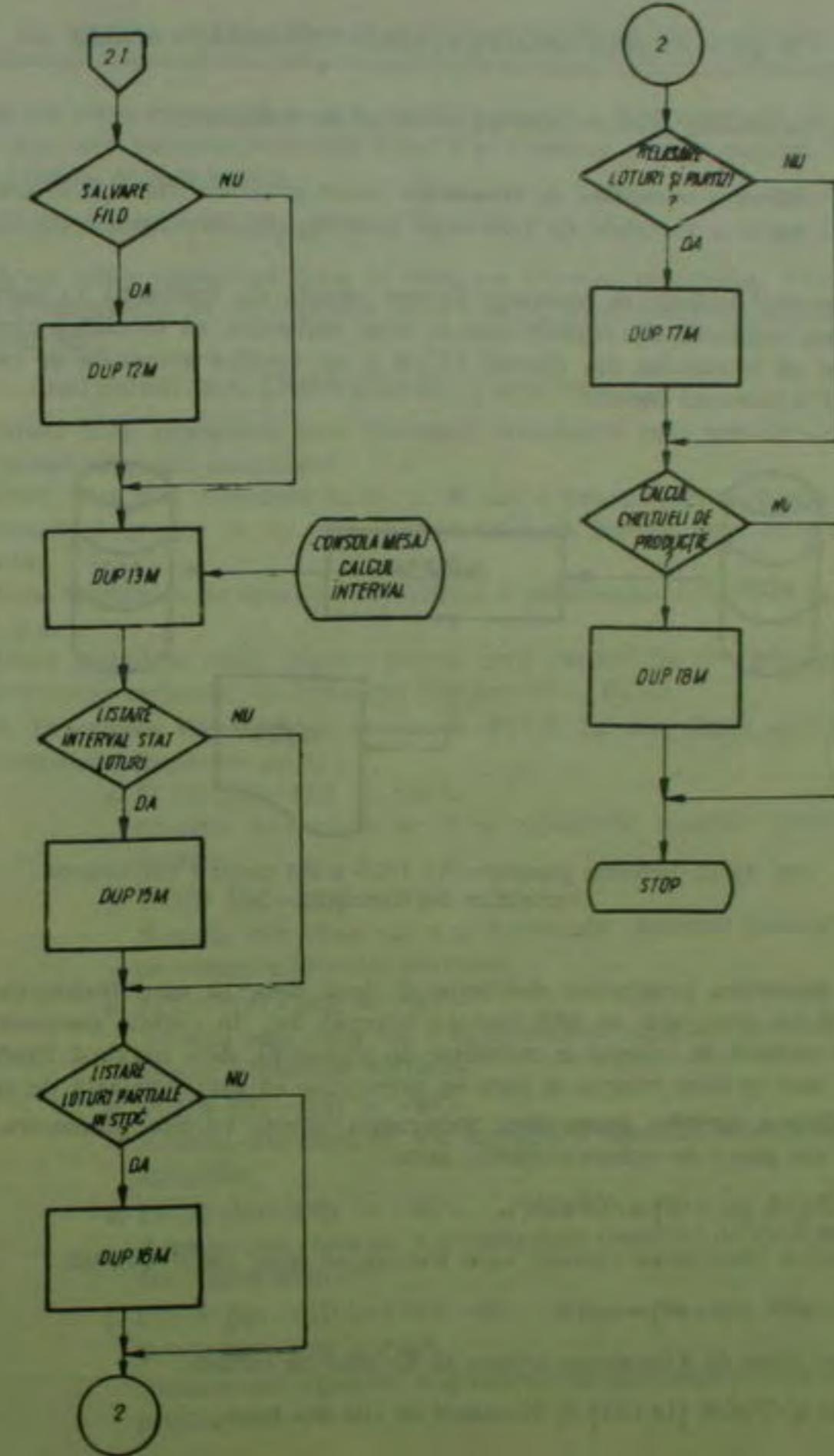


Fig. 2.20. (continuare).

5.2.6. Descrierea prelucrărilor cu limbajul de proiectare

A. Verificarea cartelelor de tranzacții. Acest program (fig. 5.2.21) are scopul de a verifica cartelele de tranzacții care se află înregistrate pe fișierul FCUS.

Programul trebuie să analizeze fiecare cartelă de tranzacții în parte, să stabilească cimpurile din cartele care au erori sintactice, să identifice cartelele străine și să le eliminate din fișierul FCUS și să verifice grupurile de cartele dacă sunt în secvență corectă.

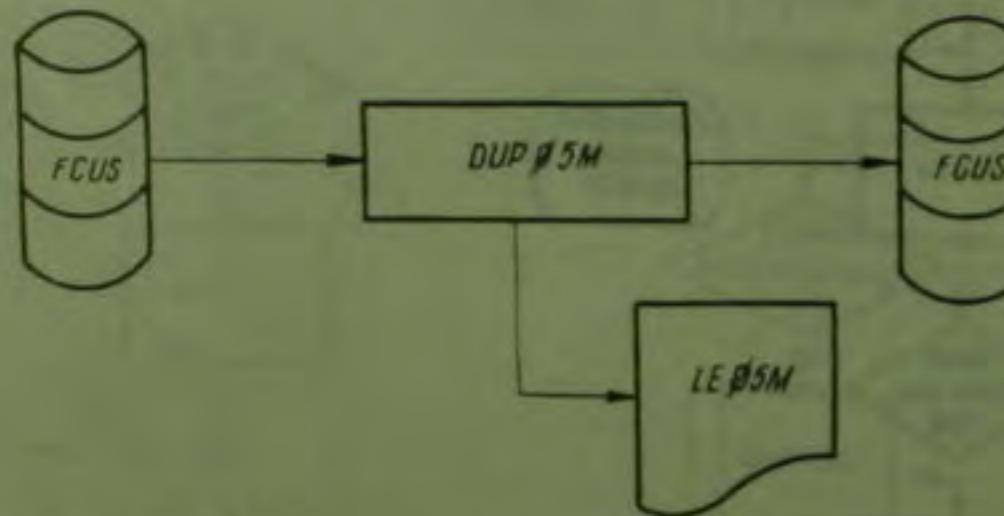


Fig. 5.2.21. Schema programului DUP #5M pentru verificarea cartelelor de tranzacții.

De asemenea programul stabilește și dacă data la care tranzacția este raportată ca executată, se află într-un interval dat. În cartela parametru se dă data minimă de început a cartelelor de tranzacții, data maximă fiind data zilei în care se face rularea și care se presupune că este furnizată de sistem.

a. Citirea cartelei parametru. Programul citește cartela parametru și o verifică din punct de vedere sintactic, astfel :

a. $CP05M \{01-05\} = /CP05M\}$.

Verifică identitatea cartelei care trebuie să aibă codul CP05M.

b. $CP05M \{06-09\} = \text{spațiu}$.

Acest cimp de 4 caractere trebuie să fie liber pe cartelă.

c. $01 < CP05M \{10-11\} \leqslant \text{Numărul de zile din lună}$.

5.2. EXEMPLE DE APLICARE A SPAD (IN INDUSTRIA TEXTILĂ) 199

Acest cimp reprezintă ziua în care tranzacția a fost efectuată. Se verifică dacă ziua are valoarea cuprinsă între 1 și numărul de zile maxim din lună, ținând seama de anii bisecți.

d. $01 \leqslant CP05M \{12-13\} \leqslant 12$.

Acest cimp reprezintă luna în care s-a efectuat tranzacția. Valoarea din acest cimp trebuie să fie cuprinsă între 1 și 12, adică numărul maxim de luni dintr-un an.

e. Anul curent—1 $\leqslant CP05M \{14-15\} \leqslant$ anul curent

Acest cimp reprezintă anul efectuării tranzacției care trebuie să fie anul curent sau cel puțin anul trecut.

Dacă unul din cimpurile a., b., c., d. sau e este eronat, cartela parametru se consideră greșită și se cere operatorului să inscrie data curentă de la consolă.

Data transmisă de operator se verifică după aceleași criterii ca la punctele c., d. și e.

După stabilirea datei minime pentru care tranzacțiile sunt admise se trece la verificarea cartelelor de tranzacții înregistrate în FCUS.

b. Descrierea fișierului de tranzacții FCUS. În acel fișier există 6 clase de înregistrări împărțite astfel :

a. $FCUS \{[01-02]\} = /05\}$.

Aceasta este clasa nr. 1 și reprezintă raportări pentru loturi lansate.

b. $FCUS \{[01-02]\} = /06\}$.

Aceasta este clasa nr. 2 și reprezintă raportări privind intrarea pe mașini a loturilor normale.

c. $FCUS \{[01-02]\} = /07\}$.

Aceasta este clasa nr. 3 și reprezintă raportări privind intrarea în stoc a loturilor parțiale.

d. $FCUS \{[01-02]\} = /08\}$.

Aceasta este clasa nr. 4 și reprezintă raportări privind staționarea utilajelor.

e. $FCUS \{[01-02]\} = /09\}$.

Aceasta este clasa nr. 5 și reprezintă raportări privind terminarea fabricației unui lot.

f. $FCUS \{[01-02]\} > /10\}$ sau

$FCUS \{[01-02]\} \leqslant /04\}$.

Aceasta este clasa nr. 6 și reprezintă tranzacții străine care nu se prelucrează.

Fiecare clasă din FCUS este formată din grupuri de înregistrări. Grupul se identifică prin condițiile :

- g. Clasa ambelor înregistrări este aceeași.
- h. Cimpul $\{03\div11\}$ din ambele înregistrări are aceeași valoare, deci e vorba de aceeași zi și aceeași mașină.

Prin urmare :

$$\begin{aligned} g. \text{FCUS } \{[01\div02]\}_i &= \text{FCUS } \{[01\div02]\}_j \\ h. \text{FCUS } \{[01\div02], [03\div11]\}_i &= \text{FCUS } \{[01\div02], [03\div11]\}_j \end{aligned}$$

unde i și j sunt numere naturale asociate înregistrărilor din clasă.

Intr-un asemenea grup de înregistrări cimpul $\text{FCUS}\{79\div80\}$ reprezintă secvența înregistrării în grup. Înregistrările dintr-un grup trebuie să satisfacă condiția :

$$\text{FCUS } \{79\div80\}_{i+1} = \text{FCUS } \{79\div80\}_i + 1$$

adică secvența să fie în ordinea crescătoare a numerelor naturale. De asemenea prima înregistrare din grup trebuie să aibă secvența în cimpul $79\div80$ egală cu $/01\backslash$. Excepțiile de la această regulă sunt tratate separat pentru fiecare clasă de înregistrări.

Clasele de înregistrări din FCUS, cu excepția clasei nr. 6, au cimpurile $\{03\div08\}$ și $\{09\div11\}$ cu aceeași structură.

$\text{FCUS } \{03\div08\}$ reprezintă data efectuării tranzacției care trebuie să fie mai mare sau cel puțin egală cu data din cartela parametru CP06M $\{10\div15\}$ și mai mică sau cel mult egală cu data prelucrării.

$\text{FCUS } \{09\div11\}$ reprezintă codul mașinii sau atelierul. Mașina sau atelierul cu acest cod trebuie să se găsească în fișierul FNM (fișierul cu numele mașinilor).

c. Procedurile de prelucrare. Prelucrarea se face prin citirea secvențială a fișierului FCUS, din care se obțin tranzacțiile care apoi sunt verificate. Procedurile de prelucrare sunt următoarele :

Identificarea clasei de înregistrări. Clasa nr. 6 poate apărea numai la începutul și la sfârșitul fișierului. Celelalte clase apar succesiv, în ordinea arătată la punctul 8.2.

Dacă o clasă de înregistrări a început să fie prelucrată (cu excepția clasei nr. 6), înregistrările din această clasă sunt prelucrate succesiv. În momentul în care începe o nouă clasă, clasa precedentă s-a terminat și nu mai apare în cadrul prelucrării respective.

Observația rămîne adevărată și pentru grupurile de înregistrări din clasă.

5.2. EXEMPLE DE APLICARE A SPAD (IN INDUSTRIA TEXTILA) 201

Gruparea pe clase și pe grupuri de înregistrări s-a obținut prin operația de sortare făcută cu programul DUP 04M.

Pentru toate clasele, cu excepția clasei nr. 6, se face mai întâi verificarea cimpurilor $\{03\div08\}$ și $\{09\div11\}$, după aceeași procedură indiferent de clasă ; se verifică apoi secvența (adică grupul de clasă) și apoi se trece la procedura de prelucrare a clasei respective.

Citirea unei noi înregistrări din FCUS. La citirea primei înregistrări din FCUS, aceasta poate apartine oricărei clase.

Dacă $\text{FCUS}\{[01\div02]\} \geq /10\backslash$ prima clasă din FCUS este tot clasa nr. 6, dar în fișier nu va mai apărea nici o altă clasă. În acest caz nu există tranzacții care se înregistrează.

Dacă $/04\backslash < \text{FCUS } \{[01\div02]\} \leq /09\backslash$, clasa de început este una din clasele 1, 2, ..., 5.

În timpul prelucrării apar situațiile :

— înregistrarea este prima intr-o nouă clasă. Se verifică condițiile :

$$/05\backslash \leq \text{FCUS } \{[01\div02]\}_{\text{actual}} \leq /09\backslash$$

și

$$\text{FCUS } \{[01\div02]\}_{\text{actual}} > \text{FCUS } \{[01\div02]\}_{\text{precedent}}$$

Inregistrarea este prima într-un nou grup.

Se verifică condițiile

$$\text{FCUS } \{[01\div02]\}_{\text{actual}} = \text{FCUS } \{[01\div02]\}_{\text{precedent}}$$

și

$$\text{FCUS } \{[03\div11]\}_{\text{actual}} \neq \text{FCUS } \{[03\div11]\}_{\text{precedent}}$$

— înregistrarea este oarecare (nu prima) într-un grup. Se verifică condițiile :

$$\text{FCUS } \{[01\div02]\}_{\text{actual}} = \text{FCUS } \{[01\div02]\}_{\text{precedent}}$$

$$\text{FCUS } \{[03\div11]\}_{\text{actual}} = \text{FCUS } \{[03\div11]\}_{\text{precedent}}$$

Inceperea unei noi clase. Se verifică dacă aceasta este clasa nr. 6. În acest caz există două situații posibile :

Clasa 6 a apărut înaintea uneia din clasele 1, 2, 3, 4, sau 5 (adică $\text{FCUS } \{[01\div02]\} < /05\backslash$).

În acest caz se tipărește mesaj că a fost întâlnită clasa nr. 6.

Se citește secvențial FCUS pînă cind se întâlnește una din clasele 1, 2, ..., 5.

Fiecare înregistrare citită în FCUS este listată dacă aparține clasei 6.

Dacă în FCUS nu se întâlnește nici una din clasele 1, 2, 3, 4 sau 5, se tipărește mesaj care anunță inexistența altor clase și programul se termină.

Clasa 6 a apărut după una din clasele 1, 2, 3, 4, 5. În acest caz se tipărește iarăși apariția clasei nr. 6 și se procedează apoi ca la punctul a.

Dacă este o clasă care se verifică sau prelucrăză (clasa 1, 2, 3, ..., 5) atunci se reține clasa, se tipărește mesaj că încep verificările pentru această clasă și se trece la verificarea și prelucrarea înregistrării, ținând seama că înregistrarea este prima și în grup.

Inceperea unui nou grup. Se verifică sintactic cimpul FCUS {03÷11}:

FCUS {03÷11} = numeric.

FCUS {03÷08} trebuie să fie cuprins între data minimă de efectuare a tranzacțiilor (găsită în CP06M {10÷15} sau prin mesaj de la operator) și data zilei în care se efectuează prelucrarea.

Data este scrisă sub formă: zi, lună, an.

Se verifică dacă FCUS {09÷11} reprezintă un cod mașină. Se caută FNM {<FCUS {09÷11}>}

Dacă această înregistrare este găsită în FNM atunci cimpul este validat, dacă înregistrarea nu este găsită, acest cimp este considerat eronat.

Răsultatul acestei verificări se păstrează pînă la apariția unui nou grup.

În acest caz, dacă rezultatul verificării este corect, se verifică cimpul {79÷80} în care se găsește secvența înregistrărilor. În acest cimp, valoarea secvenței trebuie să fie egală cu /01\, adică înregistrarea este prima în grup.

Dacă FCUS {79÷80} ≠ /01\ se semnalizează prin tipărire de mesaj și se continuă prelucrarea conform procedurilor specifice clasei.

Dacă rezultatul verificării este greșit, se tipărește mesaj de eroare, înregistrarea, cimpurile eronate și tipurile de eroare.

Se trece la punctul 5).

Prelucrarea unei înregistrări oarecare. În acest caz:

FCUS {01÷02}_{i+1} = FCUS {01÷02}_i

FCUS {03÷11}_{i+1} = FCUS {03÷11}_i

adică înregistrarea actuală (notată cu i+1) și înregistrarea precedentă (notată cu i) au aceeași clasă și aceeași valoare a cheii pentru verificarea grupului.

La această înregistrare se verifică:

Dacă FCUS {79÷80}_{i+1} = FCUS {79÷80}_i + 1.

Dacă rezultatul este corect se trece la verificarea, conform procedurilor clasei din care înregistrarea face parte.

Dacă rezultatul este incorrect pot apărea 2 situații:

Cind $FCUS \{79÷80\}_{i+1} < FCUS \{79÷80\}_i$

Cind $FCUS \{79÷80\}_{i+1} = FCUS \{79÷80\}_i + \alpha$ ($\alpha > 1$).

În primul caz, se tipărește mesajul că apare secvență dublă, înregistrarea precedentă și cea următoare, iar în cazul al doilea se tipărește mesaj că au fost sărită un număr de înregistrări egal cu $FCUS \{78÷80\}_{i+1} - FCUS \{79÷80\}_i - 1$ secvențe și de asemenea înregistrarea precedentă și cea următoare.

Se trece la procedura de verificare conform clasei citită din FCUS.

Rezultatul verificării secvenței se păstrează pînă la verificarea următoare.

Verificarea clasei de înregistrări nr. 1.

(FCUS {[01÷02]} = /05\).

Pentru o înregistrare pot apărea două tranzacții. Tranzacțiile se scriu sub formă:

FCUS {12+α★. 24÷35+α★ 24}

unde α poate căpăta două valori, 0 și 1, 0 corespunzînd primei tranzacții, iar 1 celei de a doua tranzacții.

Verificările sunt următoarele:

— Verificarea secvenței. Dacă secvența este greșită, se tipărește mesaj de eroare, înregistrarea precedentă și înregistrarea actuală și se trece la verificarea tranzacțiilor.

— Combinării posibile de verificări pentru tranzacții sunt perechile de condiții:

FCUS {12÷35} ≠ spațiu

FCUS {36÷57} ≠ spațiu

și

FCUS {12÷35} ≠ spațiu

FCUS {36÷57} = spațiu

Orice altă combinație este greșită și se semnalizează prin tipărirea de mesaj de eroare.

Dacă apare combinația a două se reține că înregistrarea următoare trebuie să indice începerea unui nou grup.

Cimpurile se verifică astfel :

- FCUS $\{12 + \alpha * 24 \div 12 + \alpha * 24\}$ = spațiu
- FCUS $\{13 + \alpha * 24 \div 16 + \alpha * 24\}$ = numeric
- FCUS $\{17 + \alpha * 24 \div 20 + \alpha * 24\}$ = numeric
- FCUS $\{21 + \alpha * 24 \div 26 + \alpha * 24\}$ = alfanumeric
- FCUS $\{27 + \alpha * 24 \div 30 + \alpha * 24\}$ = numeric
- FCUS $\{31 + \alpha * 24 \div 35 + \alpha * 24\}$ = numeric

Dacă unul sau mai multe cimpuri sunt eronate, se tipărește mesaj de eroare, FCUS $\{01 \div 11\}$, numărul tranzacției și tranzacția greșită, FCUS $\{79 \div 80\}$, cimpurile eronate și tipul eroare.

După terminarea verificării înregistrării acolo unde apar greșeli la verificarea cimpurilor, se înlocuiește tranzacția prin valoarea '9 9...9' .

Dacă în înregistrare nu va exista cel puțin o tranzacție corectă și diferită de valoarea spațiu, înregistrarea nu se înscrie în FCUS.

Dacă în înregistrarea precedentă a existat combinația a două și totuși grupul nu s-a terminat (înregistrarea actuală trebuie să fie prima dintr-un nou grup) se tipărește mesaj corespunzător și se continuă verificarea.

Dacă înregistrarea are cel puțin o tranzacție corectă și diferită de spațiu, înregistrarea se înscrie în FCUS și se continuă prelucrarea de la punctul b.

Subliniem că zona FCUS $\{09 \div 11\}$ trebuie să aibă aceeași valoare pentru toate înregistrările din această clasă de înregistrări, care reprezintă codul atelierului de lansare.

Dacă pentru o înregistrare în FCUS codul nu este cel al atelierului de lansare, această înregistrare nu va fi înscrișă în FCUS.

Verificările pentru celelalte clase se descriu în mod analog.

B. Descrierea programului DUP06M. Vom descrie numai prelucrările clasei de înregistrări nr. 1.

$$\text{FCUCS } \{[01 \div 02]\} = \text{'05'}$$

pentru celelalte, prelucrările efectuindu-se analog.

1. Se citește secvențial FCUCS.
2. Se testează dacă $\text{FCUCS } \{[01 \div 02]\} = \text{'05'}$. Dacă DA se trece la prelucrarea clasei de înregistrări corespunzătoare.
3. O înregistrare din FCUCS poate conține două tranzacții :

$$\text{FCUCS } \{[12 + \alpha * 25 \div 35 + \alpha * 25]\} \text{ pentru } 0 \leq \alpha \leq 1.$$

O tranzacție nu se prelucreză dacă :

- are valoarea '99...9' ,
- are valoarea spațiu.

O înregistrare se consideră prelucrată dacă $\alpha > 1$ și prelucrarea se continuă de la punctul 1.

4. Dacă tranzacția se prelucreză :

- se formează cheia de acces

$$\langle \text{KEY } \{1 \div 5\} \rangle : = \text{FCUCS } \{12 + \alpha * 25 \div 16 + \alpha * 25\}$$

Cu această cheie se caută înregistrarea în FIELD. Dacă înregistrarea a fost găsită, se tipărește :

- mesaj de eroare că lotul a fost lansat,
- FCUCS $\{01 \div 11\}$,
- tranzacția.

Se trece la prelucrarea tranzacției următoare (punctul c).

5. Dacă înregistrarea nu a fost găsită, se formează cimpurile în FIELD, astfel :

FIELD $\{1 \div 832\} = \text{zero}$. În zonele în care reprezentarea se face în zecimal împachetat se va înscrie zero împachetat în fiecare ^a).

^a) Un sir de caractere numerice poate forma un număr zecimal reprezentat în memorie internă în codul arătat la pagina 119.

Această formă de reprezentare se numește formă neîmpachetată.

Semnul numărului apare în ultimul caracter care formează numărul în primele patru poziții, având de exemplu valoarea de 1111 pentru semnul plus și 1110 pentru semnul minus.

De exemplu —675 se scrie :

11110110

11110111

11100101

In afară de forma neîmpachetată, numerele pot fi reprezentate în memorie și sub formă împachetată.

În acest caz, fiecare cifră se înscrie într-o jumătate de caracter cu valoarea ei în binar, iar semnul apare în ultimii 4 biți ai ultimului caracter. Numărul de caractere în care se înscrie un număr zecimal împachetat este un număr întreg. Numarul de caractere se calculează prin formula :

$$n_c = \text{Intreg} \left(\frac{N_c}{2} \right) + 1$$

unde n_c este numărul de caractere necesar pentru reprezentare.

FILD $\{1 \div 1\} := /LN$
 FILD $\{2 \div 2\} := /0N$
 FILD $\{3 \div 4\} := /00N$
 FILD $\{5 \div 5\} := /0N$
 FILD $\{6 \div 6\} := /0N$
 FILD $\{9 \div 13\} := FCUCS \{12 + \alpha \star .25 \div 16 + \alpha \star .25\}$
 FILD $\{14 \div 17\} := FCUCS \{17 + \alpha \star 25 \div 20 + \alpha \star 25\}$
 FILD $\{18 \div 22\} := FCUCS \{31 + \alpha \star .25 \div 35 + \alpha \star .25\}$
 FILD $\{30 \div 30\} := /0N$
 FILD $\{31 \div 31\} := /0N$
 FILD $\{32 \div 33\} := FCUCS \{9 \div 11\}$

Fie ZONA $\{1 \div 10\}$ o arie de 10 caractere în memorie.
 ZONA $\{1 \div 2\} := FCUCS \{07 \div 08\}$
 ZONA $\{3 \div 4\} := FCUCS \{05 \div 06\}$
 ZONA $\{5 \div 6\} := FCUCS \{03 \div 04\}$
 ZONA $\{7 \div 10\} := /0000N$

iar N_c este numărul de cifre pe care le are numărul.

De exemplu —675 se scrie :

$\begin{array}{cc} 0110 & 0111 \\ \hline I & II \end{array}$	$\begin{array}{cc} 0101 & 1110 \\ \hline III & IV \end{array}$
caracterul 1	caracterul 2

Dacă se reprezintă numărul +14752312 se calculează :

$$n_c = \text{Intreg} \left(\frac{8}{2} \right) + 1 = 5$$

Numărul avind 8 cifre și fiind necesare 5 caractere, prima jumătate de caracter rămîne liberă și va fi umplută automat de către sistemul de prelucrare automată a datelor cu zero.

FILD $\{34 \div 39\} := \{\text{ZONA } 1 \div 10\}$
 FILD $\{37 \div 375\} := FCUCS \{21 + \alpha \star 25 \div 26 + \alpha \star 25\}$
 FILD $\{400 \div 401\} := /00N$
 FILD $\{418 \div 418\} := /LN$
 FILD $\{419 \div 421\} := FCUCS \{27 + \alpha \star 25 \div 30 + \alpha \star 25\}$
 FILD $\{402 \div 403\} := /01N$

Se reia prelucrarea de la punctul c.

C. Lista unui fișier în aplicația și urmărirea producției. Se face o descriere a programului pentru raportarea (listarea) loturilor parțiale lătrate în staționare și în stoc. Prin convenție repetarea unui caracter de un număr de ori în sirul de caractere se scrie :

$/n)cN$

unde n reprezintă factorul de repetiție scris între paranteze și c reprezintă un caracter oarecare.

Fișierul pe imprimantă de raportare a loturilor parțiale :

$RP15M :=$ fișier pe imprimantă.

Organizarea secvențială. Tipul ieșire (output). Lungimea înregistrării 110 caractere, neblocat.

* Deci :

$\begin{array}{cc} 0000 & 0001 \\ \hline I & II \end{array}$	$\begin{array}{cc} 0100 & 0111 \\ \hline III & IV \end{array}$	$\begin{array}{cc} 0101 & 0010 \\ \hline V & VI \end{array}$	$\begin{array}{cc} 0011 & 0001 \\ \hline VII & VIII \end{array}$	$\begin{array}{cc} 0010 & 1111 \\ \hline IX & X \end{array}$
Caracter 1	Caracter 2	Caracter 3	Caracter 4	Caracter 5

Dacă se mută un număr impachetat zecimal într-o zonă din memorie care este rezervată tot pentru reprezentare în zecimal impachetat, dar are rezervat un număr mai mare de caractere SPAD aliniază la dreapta punind semnul în ultimul caracter în pozițiile binare 5—8 și umplind succesiv caracterele de la dreapta la stînga cu cifrele numărului, cite 2 pe caracter. Spațiul rămas liber la stînga se umple cu zero. Dacă numărul de caractere e mai mic, atunci se face ajustarea la stînga, pînzindu-se cifre.

Conține 8 clase de înregistrări*

RP15M {[1]} :: = "(36) [S]' LOTURI [S] PARTIALE [S]
 EXISTENTE [S] IN [S] STOC [S] PINA [S]
 LA [S] DATA [S] DE [S][N][N]/[N][N]/
 [N][N][N]"

RP15M {[2]} :: = "(100) [S]' PAGINA [S] XXX"

RP15M {[3]} :: = '[S][S] RP15M '(54) [S]' RP15M '(52) [S]"

RP15M {[4]} :: = "(58) x' [S][S]'(58) x"

RP15M {[5]} :: = 'x x [S][S][S][S][S][S]'(7) [S]' ★ '(10)
 [S]'★INTE★LUNGIME★[S][S][S][S]*
 [S][S][S][S][S][S]*[S][S][S][S]*
 [S][S]*'(6)[S]'★'(7)[S]'★'(10)[S]*INTE
 ★LUNGIME★[S][S][S][S]*'(7)[S]'
 [S][S][S][S]★★'

RP15M {[6]} :: = '★★ NUMĂR ★ [S][S] COD [S][S] ★ [S]
 [S] DATA [S][S][S][S] ★ RVAL ★ [S]
 [S] IN [S][S][S] ★ ART ★ CULOARE ★
 COD [S][S] ★ NUMĂR [S] ★ [S][S] COD
 [S][S] ★ [S][S] DATA [S][S][S][S] ★
 RVAL ★ [S][S] IN [S][S][S] ★ ART ★
 CULOARE ★ COD [S] ★ *

* [S] înseamnă spațiu

[N] înseamnă cifră

★ înseamnă caracterul asterics

RP15M {[6]} :: = '★★ '(6) [S]' ★ [S][S] LOT [S][S] ★
 INTRARE [S] IN ★ [S] IN [S] ★ METRI SI
 ★ [S][S][S][S] ★ '(6) [S]' ★ '(7) [S] ★ ATEL ★
 [S][S] ★ '(6) [S]' ★ [S][S] LOT [S][S] ★
 INTRARE [S] IN ★ [S] METRI SI ★ [S][S]
 [S][S] '(6) [S]' ★ ATEL ***

RP15M {[7]} :: = '★★ CURENT ★ PARTIAL ★ [S][S] STOC
 [S][S][S] ★ ZILE ★ CENTIMT ★ [S][S][S]
 [S] *'(6) [S][S][S][S][S] ★ [S][S] ★
 CURENT ★ PARTIAL ★ [S][S] STOC [S]
 [S][S] ★ ZILE ★ CENTIMT ★ [S][S][S]
 [S] *'(6) [S]' ★ [S][S][S][S] ★ **

RP15M {[8]} :: = '★ [S][S][N][N][N][N][N][S][N][N]
 [N][N][N]/[N][S][N][N]/[N][N]/[N]
 [N][N][N][S][N][N][N][N][N][S][N]
 [N][N][N][N][N][S][N][N][N][N][N]
 [S]'(6)[N]'[S]'(4)[N]'(5)[S]'(5)[N][S]
 '(5)[N]/[N][S][N][N]/[N][N]/'(4)[N]
 [S]'(4)[N]'[S]'(4)[N]'[N][N][S][N][N]
 [N][N][S]'(6)[N]'[S][N][N][N][N][N]★*

Prelucrări (vezi RP15M)

a) Fișierul RP15M este folosit astfel :

Pentru pagină nouă : se tipărește RP15M {[1]} pînă la RP15M {[7]} și apoi RP15M {[4]} după care prin citirea secvențială a fișierului FLPD prin perechi de înregistrări (pentru 2 înregistrări din FLP se scrie o înregistrare din RP15M), pînă cînd s-a completat o pagină.

La pagina următoare se repetă procedeul de la punctul a).

6. Elemente, etape și probleme ale implementării SIPAD

6.1. Apariția SIPAD și tendințe generale de dezvoltare

Sistemele informaționale de prelucrare a datelor în întreprinderi s-au dezvoltat la început în mod spontan, chiar dacă nu au existat totdeauna condițiile cele mai favorabile pentru constituirea lor. Un sistem informațional al unei întreprinderi reprezintă după cum am văzut totalitatea mijloacelor și procedeelor (informații, documente, circuite informaționale, prelucrări, baza de date etc.) ce asigură legătura între elementele direct productive (elemente conduse) și cele de organizare și conducere (decizionale). Între elementele decizionale și procesele de prelucrare se stabilesc o serie de relații care trebuie să determinate exact pentru proiectarea unui SIPAD.

Astfel, de exemplu, decizia de a lansa un produs în fabricație, decizie luată la un nivel superior de conducere, este urmată de procese de prelucrarea datelor (descompunerea produsului, conform nomenclatorului, pînă la piese și materiale, cumularea pieselor de același fel și elaborarea listelor cu termenele respective, la care piesele trebuie lansate în producție). Urmează decizia privitoare la aprovizionarea cu materiale, procesul de prelucrare a datelor constând în compararea necesarului cu stocul existent și completarea comenziilor către furnizori. Deciziile de lansare în fabricație a loturilor de piese sunt urmate de completarea bonurilor de lucru și de materiale corespunzătoare. Raportarea producției furnizează datele privind avansul producției, date care permit să se ia decizii în cazul cînd apar perturbații față de planul inițial etc. Din acest exemplu simplificat se observă succesiunile proces de decizie — proces de prelucrarea datelor — proces de producție — proces de prelucrare — proces de decizie, care stau la baza funcționării întreprinderii.

Continuând cu același exemplu se poate observa că, de cele mai multe ori, conducerea la nivelul secțiilor de fabricație, necunoscind exact caracteristicile de răspuns ale elementului condus (capacitatea exactă a secției, timpul de răspuns) ca și datorită prezenței unei mari multimi de factori perturbatori (întirzieri în aprovisionarea cu materii prime, modificări în planul de livrări, defectări de utilaje etc.) care fac ca aceste caracteristici să fie variabile, recurge la conducerea în circuit închis a secțiilor respective. Legătura inversă (reacția), care închide circuitul de conducere, urmărește informarea elementelor de decizie asupra evoluției elementului condus (în cazul exemplului: avansul producției și realizarea comenziilor). Pe calea de reacție, prelucrarea datelor în acest caz înseamnă atât colectarea datelor (de exemplu raportarea realizărilor: cantități, tempi necesari execuției), cit și transmiterea (de exemplu centralizarea informațiilor la puncte de control pe secții și apoi la dispecer sau la biroul de lansare) și prelucrarea propriu-zisă a datelor (de exemplu, rapoarte privitoare la avansul producției).

În timp, funcționarea elementelor conduse s-a complicat, atrăgind după sine creșterea volumului de date de prelucrat simultan și necesitatea măririi vitezei de obținere a rezultatelor. Aceasta a determinat introducerea mecanizării în prelucrarea datelor pornind cu mecanizarea realizată prin folosirea mașinilor contabile și de birou, și continuând cu folosirea echipamentelor cu cartele perforate și electronice.

Complexitatea crescindă a elementelor conduse, a informațiilor și circuitelor informaționale, ca și problemele noi ridicate de mediul în care aceste elemente evoluează (ca de exemplu creșterea competitivității sau necesitatea deservirii rapide) au arătat că imbunătățirile parțiale ale sistemelor informaționale din întreprinderi nu sunt suficiente și că este necesară considerarea sistemelor informaționale din întreprinderi ca un obiect de studiu de sine stătător, că este necesară înlocuirea sistemelor informaționale existente, apărute și dezvoltate de obicei în mod spontan, prin sisteme informaționale proiectate, înținând seama de cerințele noi ale sistemului de conducere. În acest paragraf, ne vom preocupă în principal de introducerea sistemelor informaționale folosind prelucrarea automată a datelor (SIPAD) în întreprinderi, pentru deservirea conducerii acestora.

Activitatea de proiectare a SIPAD, de realizare și instalare a acestuia (înlocuind vechiul sistem informațional) s-a dovedit a fi o activitate complexă care angrenează resurse importante (personal,

mijloace financiare) pe perioade de timp care pot varia, în general, de la doi la cinci ani de zile. În funcție de obiectivele inițial stabilite și aceasta în legătură cu resursele disponibile, proiectul se poate dezvolta, în general, referitor la două aspecte:

a. aspectul cantitativ, constând din numărul de activități din întreprindere cuprinse în proiect (numărul de subsisteme de aplicații: producție, stocuri, aprovisionare, desfaceri etc., ce urmează să fie trecute pe calculator).

b. aspectul calitativ, constând din cuprinderea în cadrul proiectului noului SIPAD numai a ariei de evidență și raportare sau extinderea lui și în probleme de modelare și optimizări pentru pregătirea și luarea deciziilor realizând un anumit grad de integrare al subsistemelor de aplicații prin folosirea unei baze comune de date sau un anumit grad de integrare al sistemului informațional în mediul înconjurător (cu alte sisteme informaționale). Primele aplicații ale sistemelor de prelucrare automată a datelor în întreprinderi s-au referit la prelucrarea pe acest echipament a lucrărilor efectuate până atunci manual sau pe cartele perforate, fără a modifica în mod esențial documentele de intrare în sistemul de calcul și nici documentele de ieșire (rapoarte) din acest sistem. Desigur, prin aceasta s-a îmbunătățit calitatea informației care constituia baza de decizie pentru conducere. Aceasta prima informație mai repede, mai exact prelucrată, adică mai operativ și mai sigură. S-a constatat însă că utilizarea în acest fel a calculatoarelor electronice era nerățională, mult depărtată de posibilitățile acestui echipament. Mai mult decât atât, există riscul ca din partea conducerii întreprinderii să apară cererea unui volum de rapoarte mult mai mare decât este necesar, tocmai din cauza ușurinței cu care aceste rapoarte pot fi elaborate și să nu se folosească posibilitățile acestor echipamente, în principal, în sensul conducerii de ansamblu (integrată) și raportării prin excepții.

Tocmai de aceea, personalul implicat în acțiunea de introducere a sistemelor electronice de prelucrare a datelor în întreprinderi nu se poate limita la a transpune *procedurile manuale sau mecanografice existente, pe echipamentul electronic*. El este dator să analizeze procesul de conducere, descoperind împreună cu cadrele de conducere din întreprinderi, cerințele autentice ale sistemului de conducere a întreprinderii.

Pentru micșorarea rezistenței personalului din întreprindere față de introducerea noului sistem, este recomandabil ca el să fie cit mai mult implicat în proiectarea noului sistem pentru ca, pe lângă o mai

bună instruire, să devină cointeresanți ca autori ai lui. Analistul de sisteme trebuie să conducă de exemplu interviurile în aşa fel, încit interlocutorul să furnizeze, el însuși, parametrii necesari proiectării noului sistem.

O deosebită importanță o are în acest sens precizarea sistemului de conducere (SIC), ca atare precizarea rapoartelor pe care trebuie să le furnizeze SIPAD și a modelelor (simulare și optimizări), care trebuie executate în cadrul sistemului de conducere, în vederea pregătirii și luării deciziilor.

Este necesară deci o analiză amănunțită a sistemului actual pentru descoperirea celor mai semnificative surse de informație, a celor mai semnificative forme de prezentare a rapoartelor și a elementelor principale care pregătesc sau conduc la luarea deciziei. Nu trebuie uitat că echipamentul electronic mărește considerabil modalitățile de prelucrare, de modelare și de prezentare a datelor. Acesta influențează tot mai mult procesul de decizie, în mod similar efectului folosirii calculatoarelor electronice în calculele științifice asupra metodelor de calcul și chiar asupra problematicii abordate.

La o scară sensibil mai mică, aceasta se referă și la mecanizarea proceselor de prelucrare a datelor prin utilizarea mașinilor de contabilizat sau cu ajutorul echipamentelor mecanografice. Într-adevăr, această mecanizare a putut nu numai să îmbunătățească procesele de prelucrare manuale existente, dar a permis și abordarea unor lucrări noi și elaborarea de noi rapoarte pentru procesele de decizie, modificind deci calitativ sistemele informaționale anterioare.

Introducerea calculatoarelor nu a reprezentat numai executarea proceselor de prelucrare mai rapid, mai exact și asupra unor volume mai mari de date în comparație cu procedeele manuale sau mecanografice; s-a produs un salt calitativ prin utilizarea SIPAD, care se manifestă atât prin preluarea unor probleme complexe de prelucrare, practic imposibil de rezolvat altfel (de exemplu, problema programării operative în condițiile unei întreprinderi productive complexe), cit și prin preluarea de către echipamentul electronic a unor sarcini de decizie de rutină, atribuite pînă acum doar factorilor umani (de exemplu controlul stocurilor).

Un prim pas pentru pătrunderea echipamentului electronic în domeniul deciziilor se bazează pe integrarea în cadrul SIPAD a unor aplicații de „tip conducere” și pe posibilitatea simulărilor, cu scopul de a anticipa consecințele uneia sau a alteia din decizii. Aceasta înseamnă prezentarea de variante conducerii, pentru a lua o anumită decizie sau chiar elaborarea deciziei automate în diferite situații.

De exemplu, în scopul obținerii variantei optime de plan de producție, conducerea întreprinderii poate introduce în sistemul electronic diferite variante, obținind succesiv planurile de fabricație și de aprovizionare, în variantele corespunzătoare și de asemenea planurile forțelor de muncă și situația încărcării utilajelor, precum și valorile indicatorilor.

In fine, în domeniul luării unor decizii, în măsura în care procesul respectiv de decizie este algoritmizat pe baza unor modele elaborate, calculatorul electronic poate să-și asume și această sarcină. De exemplu, în problema aprovizionării, calculatorul electronic îi poate reveni sarcina de decizie de a elabora comenzi de materiale atunci când stocul fizic atinge punctul de comandă, sau în problema stabilirii plantului de fabricație, calculatorul poate stabili, conform algoritmului indicat, dimensiunea și numărul loturilor de piese de lansat.

Aplicarea calculatoarelor în domeniul luării unor decizii este legată de asemenea de utilizarea cercetării operaționale în rezolvarea problemelor respective [10, 17]. Trebuie subliniat că, datorită complexității mediului în care evoluează sistemele de conducere din întreprinderi și din cauza complexității proceselor conduse, dinamica proceselor este dificil să fie cunoscută în amănunțime. De aceea, calculatoarele se folosesc mai curind în sensul principiului conducerii prin excepție („management by exception”), calculatorul revenindu-i sarcini de decizie doar în cazurile ce se pot încadra în algoritme.

Pentru cazurile excepționale (anomalii, abateri), calculatorul va sesiza conducerea prin rapoarte de excepție.

In acest fel, conducerea este degajată de unele sarcini de decizie de rutină, putîndu-se concentra asupra problemelor majore ale dezvoltării de perspectivă ale întreprinderii sau ale îmbunătățirilor calitative ce urmează să fie realizate într-unul sau altul din sectoarele de activitate.

Introducerea sistemelor de prelucrare automată a datelor în întreprinderi impune reorganizarea structurilor existente, în sensul eliminării cazurilor particulare care îngreunează funcționarea întreprinderii. Această acțiune de sistematizare și standardizare este proprie SIPAD, cazurile particulare care nu au fost prevăzute în sistem nu sunt abordate de calculatorul electronic și nu mai există condiții pentru ca aceste activități care ar complica inutil sistemul informațional să mai apară. În general, compartimentele de planificare, programare, lansare, finanțări-contabile și a vor fi reduse în favoarea unor compartimente noi ca serviciul de informa-

tică, compartimentul de colectare a datelor etc.; majoritatea sarcinilor administrative, rutiniere și periodice fiind preluate de SIPAD, urmând ca salariații din compartimentele menționate să se adapteze la noul SIPAD și la noul sistem de conducere al întreprinderii. Pregătirea specialiștilor necesari conducerii pe baza SIPAD se face cel mai adesea prin recrutarea dintre salariații existenți în întreprindere în diferite domenii și calificarea acestora pentru utilizarea noului sistem informațional. Problema pregătirii corespunzătoare pentru conducerea întreprinderilor folosind echipamentele de prelucrare automată, se pune cu tărie și pentru cadrele de conducere. În legătură cu perspectivele anului 1980 în articolul „Cum vor funcționa întreprinderile în 1980“ (Entreprise, nr. 718/14 iunie 1969) se menționează:

„Paralel cu aceasta (este vorba de specialiști în informatică) se va produce o a doua mișcare pe planul măini de lucru în întreprinderi și anume: ajungerea în posturi de conducere a unor specialiști capabili să domine conducerea integrată. Va fi vorba mai ales de matematicieni, de oameni de știință axați pe evoluția tehnologiei și de specialiști în economie sau în vinzări, capabili să facă previzuni corecte“.

Trebuie subliniată importanța pe care o prezintă, în cazul introducerii SIPAD, relațiile om-sistem. Astfel, oricât de bine ar fi fost testat sistemul electronic înainte de instalarea lui, se va putea constata că personalul de conducere va manifesta, cel puțin la început, neîncredere în calitățile calculatorului electronic. În acest caz, este util a se prevedea în sistem posibilitatea că, periodic, sau la cerere, să poată fi verificată „competența“ sistemului. De exemplu, în cazul unui subsistem de conducere (gestiune) a stocurilor de materiale se poate prevedea listarea tuturor materialelor și a nivelului stocurilor respective la un moment dat. Evident că acest raport, de un volum care poate deveni în unele cazuri enorm, nu are alt scop, în cazul unui sistem bine proiectat, decit de a convinge personalul de conducere asupra corectitudinii funcționării sistemului.

Deși cele arătate mai sus se referă la extinderea acțiunii calculatorului electronic în cadrul unor circuite informaționale din întreprindere, trebuie precizat că încă nu putem vorbi de automatizarea completă a circuitelor respective. Aceasta ar însemna rezolvarea tuturor problemelor de colectare de date, transmiterea lor, prelucrare propriu-zisă, elaborare și transmitere a documentelor complet automatizat, fără participarea omului. Ori această perspectivă este încă destul de depărtată.

Calculatorul cu sistemul său de echipamente terminale asociate, înseamnă de obicei numai automatizarea procesului de prelucrare propriu-zisă. Sistemele de teleprocessing (comandă la distanță) tind să introducă automatizarea parțială (off-line) în transmiterea datelor la calculator, în condițiile în care liniile de comunicații nu au acces direct la calculator, sau automatizarea (on-line) în cazul introducerii directe a datelor în calculator, fără participarea omului. Colectarea datelor, de cele mai multe ori a rămas în sarcina operatorului uman. De aceea, putem vorbi de o extindere a zonei de prelucrare electronică automatizată și o restrinție a zonei de prelucrare manuală mai curind decât de a vorbi de sisteme cu prelucrarea automată a datelor. Un alt argument este constituit de faptul că SIPAD realizate pînă în prezent pe plan mondial, cu utilizarea echipamentelor electronice, nu rezolvă toate problemele de prelucrare din întreprindere. Chiar sistemele de prelucrare sunt mai curind parțial automatizate decât „automate“, problema interfețelor dintre zonele automatizate și cele manuale, în cadrul fiecărei aplicații, ca și dintre aplicațiile automatizate și cele rămase încă manuale sau mecanizate fiind de o deosebită importanță.

In sensul utilizării din ce în ce mai raționale a capacitaților echipamentelor moderne de prelucrare datelor, se inscrie și evoluția SIPAD de la cuprinderea unei sau unor probleme disparate ale întreprinderii în cadrul zonei de prelucrare electronică, la cuprinderea unui ansamblu de probleme, de aplicații și rezolvarea lor ținind seama de interacțiunile dintre ele. Aceasta constituie tendința de integrare care se manifestă din ce în ce mai puternic în proiectarea SIPAD. În perspectivă se vor constitui SIPAD în care sistemele de prelucrare electronice sau hibride (împreună cu calculatoare analogice) se vor aplica de la nivelul conducerii proceselor tehnologice pînă la nivelul conducerii superioare, verigile superioare furnizind mărimile de referință și criteriile de optim pentru verigile inferioare. În același timp se va extinde utilizarea bazelor de date, care să asigure conducerilor, informații centralizate asupra sistemelor complexe în ansamblu și asupra mediului în care aceste sisteme evoluează [9].

In afara de aspectele discutate mai sus, trebuie pusă în evidență o altă caracteristică a sistemelor de prelucrare a datelor, legată de modul în care datele intră în sistemul de prelucrare. Din acest punct de vedere sistemele de prelucrare a datelor au evoluat de la *prelucrarea în grup* („batch-processing“) la *prelucrarea individuală* („individual processing“).

line"), la aceasta din urmă tranzacțiile putind fi prelucrate imediat apar în sistem, fără a fi necesar să se aştepte gruparea și prelucrarea la o perioadă de timp (zi, săptămână etc.) a mai multor tranzacții.

Prelucrarea în grup, chiar dacă este făcută cu o frecvență mare, poate să se dovedească impropriu procesului de conducere, neasigurând calitatea suficient de bune proceselor tranzitorii. Altfel spus, unele procese pot fi conduse în mod corespunzător numai dacă constantele de timp cumulate ale elementelor de colectare a datelor, de transmitere, prelucrare și decizie și de transmitere a deciziilor sunt suficiente de mici față de constantele de timp ale procesului. Astfel de sisteme sunt denumite „*în timp real*“ („real-time“) și ele cer în majoritatea cazurilor folosirea calculatoarelor în sistem „*on-line*“ (intervenția operatorului uman este exclusă între locul de culegere și de prelucrare, respectiv de raportare, în ambele direcții de circulație a informației) [9].

Rezumind, se pot pune în evidență trei tendințe în evoluția folosirii calculatoarelor în SIPAD din întreprinderi :

A) Trecerea de la aria evidenței și raportării (și în cadrul acesteia de la automatizarea proceselor existente la descoperirea unor surse noi de informație și a unor prelucrări noi, utile procesului de conducere și organizarea concentrării informației în baza de date) la aria deciziei (și în cadrul acesteia de la alegerea unei variante pe baza simularilor la elaborarea de modele de optimizări pentru luarea deciziilor) ;

B) Trecerea de la aplicații singulare la grupe de aplicații care folosesc în comun baze de date, realizând toate legăturile între aceste aplicații și tinzind către realizarea treptată a unor sisteme integrate pentru deservirea conducerii care prezintă în rezumat următoarele caracteristici :

1. Toate subsistemele pentru aplicații utilizează aceeași bază de date, formată dintr-un număr de fișiere de bază și variabile, număr care diferă în general de la un sistem la altul. Rapoartele prin excepție pentru conducere se obțin de obicei printr-un pachet (limbaj) de interogare a bazei de date.

2. Actualizarea și întreținerea bazei de date se face printr-un sistem independent (comun tuturor subsistemelor de aplicații), care prelucrează tranzacțiile în grup sau individual pe măsură ce apar.

3. Rapoartele (periodice) sunt generate ca rezultat al activității fiecărui subsistem de aplicații, folosind baza de date pusă la zi.

4. Subsistemele de aplicații sunt corelate între ele în utilizarea bazei comune de date (a sistemului comun de date) prin sistemul comun de întreținere și actualizare a acesteia, prin prelucrarea tranzacțiilor care sunt comune și prin generarea părților comune în rapoarte.

5. Subsistemele de aplicații care în general se integrează prioritari în zona microeconomică (întreprinderi) respectiv zona macro-economică (peste nivelul întreprinderii : centrală, minister, bănci etc.) sunt următoarele :

Zona microeconomică :

- Planificarea, lansarea și controlul producției.
- Urmărire stocuri.
- Aprovizionare.
- Desfacere.
- Subsistem finanțier-contabil.

Zona macroeconomică :

- Calcule previzionale.
- Planificare și investiții.
- Urmărirea utilizării forței de muncă
- Desfacere și aprovizionare.
- Subsistem finanțier-contabil.

Subsistemele urmărire stocuri, aprovizionare și desfacere urmează să fie proiectate în prima etapă, mai mult în sensul evidenței acestor activități decât al prelucrării în cadrul sistemului automat a unor modele (de exemplu pentru control stocuri, analiza vînzărilor și studii de marketing pentru prezentarea de variante pentru decizie), a căror elaborare în etapa următoare va asigura caracterul de integrare al activităților de evidență și raportare cu cele de modelare pentru pregătirea și luarea deciziei.

Subsistemele descrise și care se integrează prioritari la nivel macro-economic se regăsesc atât la centrale, minister, bănci etc., cât și în perspectivă la nivel național. Desigur, de pildă, în cadrul băncilor sau asigurărilor sociale, sistemele de desfacere și aprovizionare capătă caracterul specific privind desfacerea serviciilor sau operațiunilor efectuate, respectiv problemele specifice mai simple de aprovizionare (echipament de birou, echipament de multiplicare, microfilme etc.) și legătura cu sistemele de aprovizionare în cadrul cărora activitățile desfășurate de aceste instituții prezintă importanță.

6. Organizarea și generarea bazei de date trebuie făcută cu ajutorul unui pachet de programe care să permită pe de o parte adăugarea succesivă de subsisteme de aplicații fără a fi nevoie de reorganizarea bazei de date și fără a se modifica programele existente, iar pe de altă parte să asigure compatibilitatea între sisteme. Aceste condiții sunt esențiale în vederea introducerii de sisteme informaționale care să se poată dezvolta fără probleme serioase de reproiectare, elaborare de noi programe și reorganizarea bazei de date și pentru alcătuirea de sisteme informaționale tot mai complexe pînă la nivel național.

C) Trecerea de la prelucrarea în grup, parțial, la prelucrarea individuală în cadrul unor sisteme de timp real, prin extinderea aplicațiilor on-line și realizarea unor rețele de calculatoare cu echipamente de telecomandă [9].

6.2. Fazele introducerii SIPAD în întreprinderi

Indiferent de nivelul la care urmează să se desfășoare acțiunea de introducere a sistemelor informaționale, folosind sistemele de prelucrare automată a datelor, prima etapă de parcurs (fig. 6.2.1) este analiza generală a sistemului informațional existent. Ea constă, de fapt, din două aspecte: cunoașterea informațiilor elementare, a documentelor, circuitelor de informație și punctelor de decizie din sistemul existent și apoi analiza propriu-zisă, interpretarea lor. Deoarece proiectul urmează să afecteze mai mult sau mai puțin și sistemul de conducere, analiza sistemului prezent se va extinde și la acesta. Este evident că analiza sistemului informațional prezent nu reprezintă un scop în sine, ci o bază pentru stabilirea cerințelor pentru noul sistem informațional folosind mijloace automate de prelucrare a datelor [10, 17].

Procesul analiză — stabilire cerințe — proiectare (sinteză) este un proces iterativ în care cele trei componente se întrepătrund. Este un proces iterativ pentru că este practic imposibil și nici nu ar fi de dorit, ca odată declanșată analiza sistemului prezent să se meargă, continuu, pînă la finisarea proiectului în cele mai mici detalii ale lui. O primă iterație este reprezentată de studiu preliminar (feasibility study) pe care echipa de analiști îl întocmește după o primă etapă

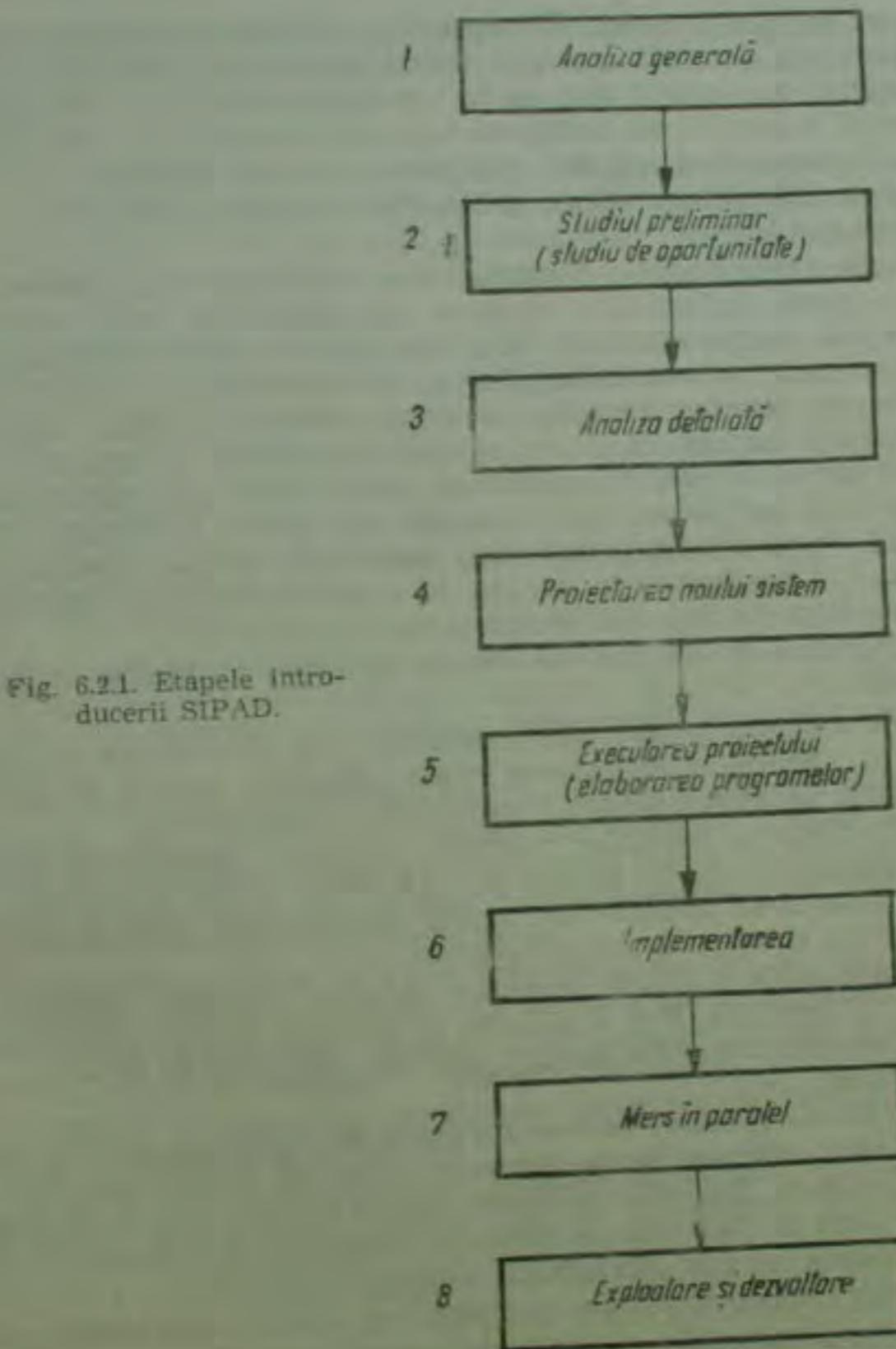


Fig. 6.2.1. Etapele introducerii SIPAD.

de lucru în întreprindere. El cuprinde o analiză generală succintă a întreprinderii, o schiță a noului sistem care se rezumă la precizarea aplicațiilor, care vor fi supuse în continuare analizei mai profunde și nivelul la care se va desfășura acțiunea (mecanizarea sau automatizarea procedurilor actuale, reorganizarea unor aplicații sau reorganizarea întregului sistem, gradul de integrare propus în prima etapă ș.a.m.d.).

Concluzia studiului preliminar, care arată căile și mijloacele prin care se poate îmbunătăți sistemul informațional existent, poate aborda atât soluția manuală, cât și cea mecanizată sau automatizată. Costul soluției și beneficiile date de diferite variante împreună cu considerente privind dezvoltarea întreprinderii, centralei, subramurei etc. sunt factorii care vor permite conducerii să decidă asupra soluției care se adoptă. Un astfel de studiu poate dura săptămâni sau luni de zile, dar pentru unele aspecte mai generale, el poate fi elaborat în cîteva zile. Aspectele mai importante pe care trebuie să le aibă studiul preliminar în vedere, în cazul introducerii unui sistem automat de prelucrare sunt următoarele :

1. *Introducere* : în care se descrie pe scurt problema ce trebuie abordată.

2. *Formularea problemei* : constituie una din părțile cheie ale studiului ; trebuie să definească exact aplicațiile și aria de aplicare în termeni familiari pentru conducerea întreprinderii.

3. *Prezentarea soluțiilor posibile* : în unele situații este necesar să se propună o singură soluție care răspunde optim problemei. Este indicat însă să se prezinte mai multe variante care conduc la beneficii diferite și necesită costuri diferite.

4. *Rapoartele și liste periodice și prin excepție furnizate de sistem* : se prezintă, fără a se intra în detaliile specifice domeniului de prelucrare automată a datelor, care sunt răspunsurile ce se așteaptă de la sistem.

5. *Fișierele necesare sistemului (baza de date)* : se descriu modificările care trebuie aduse fișierelor existente, problemele de codificări, volumele fișierelor necesare noului sistem, modul de organizare al bazei de date etc. Se prezintă de asemenea evaluări ale timpului mașină necesar prelucrării fișierelor prezентate.

6. *Informațiile de intrare necesare* : se descriu informațiile necesare pentru tranzacțiile ce efectuează actualizarea fișierelor și pentru prelucrările necesare, evaluindu-se volumele de date ce trebuie colectate, formatele de intrare etc.

7. *Schema sistemului* : se prezintă schema de sistem generală care conține prelucrările necesare fișierelor, informațiile de intrare și rapoartele ce se obțin, precum și circulația informației între aceste elemente.

8. *Programele necesare* : se descriu în general programele necesare, pachetele de programe existente și care pot fi utilizate ca atare sau prin adaptare. Se evaluează de asemenea resursele necesare pentru elaborarea programelor noi în cadrul întregului proces de prelucrare.

9. *Echipamentul necesar* : se descrie configurația necesară de echipament și software-ul necesar (pachetele de programe care trebuie eventual obținute de la firma constructoare sau din alte surse) pe baza calculelor privind timpul de incărcare al configurației pentru aplicațiile preconizate. Metoda de calcul orientată pe tranzacții, prezentată în cap. III, este utilă în această fază în care informațiile despre numărul de instrucțiuni și programele ce vor efectua prelucrările sunt încă sumare.

10. *Planul de implementare* : se descriu destul de amănunțit etapele necesare pentru darea în exploatare a noului sistem pentru a se putea aprecia de către conduceri resursele necesare, amplierea acțiunilor de pregătire a cadrelor, schimbările structurii întreprinderii etc.

O a doua iterație majoră este constituită de *analiza detaliată și elaborarea proiectului nouui sistem*. La această iterăție se trece doar după ce prima iterăție a primit avizul favorabil din partea conducerii. În măsura în care această a două iterăție este și ea aprobată se poate trece la executarea proiectului.

Desigur că trecerea la elaborarea programelor depinde esențial de obținerea avizului favorabil din partea conducerii întreprinderii.

Precizăm că în cadrul fiecărei iterății majore de care s-a vorbit, procesul unitar de analiză — determinare cerințe — proiectare, se desfășoară el însuși în mod iterativ, cu adoptarea de diferite variante și analiza implicațiilor lor ș.a.m.d.

Sistemul de iterății succesive, separate de avizări din partea beneficiarului reprezentat în acest caz de conducerea întreprinderii urmărește salvarea timpului care s-ar pierde dacă s-ar trece la detalieri fără a ne asigura că proiectul se dezvoltă pe o direcție corectă.

Teoretic, s-ar putea considera că cerințele conducerii pot fi descoperite încă din primele etape. Practic, însă se constată că acestea sunt puse în evidență și în cursul detalierii proiectului și, de multe ori, chiar după avizarea proiectului detaliat. Explicația constă nu atât din

faptul că analiza și proiectarea sunt etape care durează luni sau ani de zile, perioadă în care pot apărea cerințe noi, cit de faptul că, în majoritatea cazurilor, există dificultăți în comunicație și în înțelegerea între echipa de analiști și conducere. Aceasta din urmă realizează mai greu, din cauza limbajului diferit, imaginea exactă a nou-lui sistem. Avizarea studiului preliminar se face de obicei fără obiecții majore. Atunci cînd detalierea proiectului și executarea lui apropiș noul sistem de realitate, conducerea începe să fie conștientă de adevăratele implicații ale implementării lui, de modul de lucru nou care va apărea în întreprindere. Acesta este un fenomen obiectiv și echipa de analiști nu poate să ignore sugestiile sau cererile beneficiarului proiectului. Pe de altă parte, conducerea trebuie să fie conștientă de faptul că modificările survenite după avizarea proiectului duc la întîrzieri față de programul inițial de dezvoltare a proiectului și, în consecință, la mărirea cheltuielilor care afectează întregul proiect.

Una din procedurile recomandabile în astfel de situații constă în următoarele: se supun analizei atente și se aprobă, în cazul în care sunt fundamentate, toate sesizările sau cererile apărute pînă la avizarea proiectului. Avizarea proiectului marchează așa numitul „punct de înghețare“ după care orice sesizare sau cerere se supune unor reguli formale, stabilite și aprobate de comun acord de către echipa de analiști și de către conducere.

Sesizările și cererile apărute după „punctul de înghețare“ se supun din nou unei analize atente care va decide dacă ele sunt absolut necesare sau doar „de dorit“. În primul caz ele se includ în proiect, menționindu-se în seris întîrzierile și cheltuielile suplimentare pe care le produce intoarcerea proiectului la o fază anterioară și modificările implicate în diferite capítole ale proiectului. În al doilea caz, modificările „de dorit“ se consemnează pentru a fi even-tual luate în considerare într-o variantă ulterioară de dezvoltare a sistemului informațional.

După avizarea proiectului, se poate trece la executarea *proiectului*, ceea ce implică desfășurarea în paralel a programării (incluzînd testările de program) a începerii incărcării fișierelor și a pregăririi utilizatorului (pregătirea datelor și instruirea personalului). De multe ori se consideră că programatorii pot dezvolta singuri proiectul pentru diferitele aplicații, astfel că analistul nu are altceva de făcut decât de predat proiectul (dosarul de analiză).

În fond este evident că în timpul perioadei de executare a proiectului trebuie să existe o colaborare între cele două categorii de personal angrenat în acțiune. Este important de subliniat faptul că pregătirea datelor se începe în mod corect doar după ce partea din proiect privind organizarea bazei de date este terminată, pentru a fi siguri că această activitate, în general solicitînd un volum de muncă important, nu se dezvoltă pe o direcție greșită. Ca o excepție, se poate menționa activitatea de codificare (verificarea codificării existente și introducerea codurilor acolo unde nu există) care, într-o serie de cazuri, se poate începe înaintea terminării organizării bazei de date, care în afara listelor de informații și a modului de codificare conține legăturile între fișiere și înregistrări care nu sunt necesare pentru activitatea de codificare.

Următoarea etapă este reprezentată de *implementarea* (instalarea sau introducerea) noului sistem și urmărirea funcționării acestuia. Ea constă din înlățuirea programelor și a pachetelor de programe, încărcarea cu date reale a fișierelor și simularea noului sistem. Înainte de a trece la mersul în paralel, sistemul trebuie să fie testat. O metodă eficace de testare a sistemului (care nu trebuie confundată cu testarea programelor) este *simularea* tuturor pachetelor de aplicații a fișierelor, intrărilor și ieșirilor, de pildă pentru un micropodus elaborat ad-hoc. Se încarcă fișierele (articole, structuri, operații tehnologice, posturi de lucru, personal etc.) cu date într-un volum relativ mic, dar care să conțină toate cazurile limită pe care le poate stăpini echipa de proiectare și exploatare a sistemului. Rularea tuturor aplicațiilor în lanț pentru acest micropodus asigură colectivul de implementare de corectitudinea prelucrării la această etapă. Se trece apoi la folosirea fișierelor încărcate cu *date reale* și care au fost verificate la nivelul programelor speciale de verificare. Pachetele de programe de exploatare prelucrează datele reale și pe baza rapoartelor și listelor ce se obțin se controlează pe viu exactitatea datelor existente în fișiere. Urmează apoi „mersul în paralel“ al noului și vechiului sistem, adică funcționarea lor simultană (de exemplu, calculul salariilor poate fi realizat în paralel în cele două sisteme într-un interval de cîteva luni, înainte de a renunța la vechiul sistem) în vederea terminării încărcării fișierelor și extinderii SIPAD în cadrul întreprinderii. Desigur, că „mersul în paralel“ este costisitor și că nu totdeauna există posibilitatea menținerii pe o perioadă de cîteva săptămâni sau luni a două sisteme de prelucrare a datelor.

Cel mai adesea însă „mersul în paralel“, nu asigură condițiile testării noului sistem. Astfel, de pildă, pentru aplicații care nu pot

fi efectuate în timp util cu ajutorul sistemului manual sau mecanizat (reactualizarea frecventă a planului de fabricație sau a planului de livrări, ordonanțarea și lansarea producției, calculul necesarului de aprovizionat pe diferite perioade, în cazul apariției unor perturbații frecvente etc.), singura metodă de testare și care se utilizează înainte de exploatarea sistemului este simularea cu seturi de date din ce în ce mai complete. *Criteriul corectitudinii sistemului informațional elaborat în acest caz, este verificarea funcționării lui pe baza unor mulțimi tot mai mari de date reale.*

Etapele de implementare și de mers în paralel trebuie pregătite minuțios în prealabil. În afara specialiștilor (analisti de sisteme și programatori) care trebuie să aparțină întreprinderii, chiar în situația colaborării cu un centru de calcul, specialiști care trebuie să preia în detaliu fiecare pachet de programe necesar fiecărei aplicații, pentru a putea fi în măsură să actualizeze sau să dezvolte aplicația în viitor și care trebuie să asigure întreținerea bazei de date și a pachetelor de programe, trebuie organizate nuclee de „informaticieni” în fiecare secție a întreprinderii, care să cunoască caracteristicile de bază ale noului sistem și care să asigure introducerea acestuia și desființarea vechiului sistem. Acești salariați trebuie să cunoască procedeele de culegerea datelor, completarea documentelor, modul de transmitere al lor și utilizarea rapoartelor furnizate de calculatorul electronic. Treptat, restul personalului tehnico-administrativ din secție trebuie să învețe de la nucleul de informaticieni, devenind în final personalul de deservire al noului sistem. În același timp, toți factorii de conducere din întreprindere la toate nivelele trebuie să cunoască caracteristicile noului sistem și în special elementele specifice SIPAD de pregătire a deciziei asociate funcțiunii lor.

Este recomandabil ca pregătirea datelor pentru noul SIPAD ce se implementează să fie organizată centralizat pe întreprindere pentru toate compartimentele și toate aplicațiile ce se execută pe SPAD. O schemă organizatorică ce s-a dovedit bună în practică pentru implementarea și apoi exploatarea noului sistem în condițiile colaborării întreprinderii cu un centru de calcul este reprezentată de existența în întreprindere a unui serviciu de informatică format din analiști de sisteme și programatori, care asigură întreținerea și dezvoltarea pachetelor de programe și a bazei de date, în cadrul căruia să funcționeze un birou de pregătirea datelor, care să asigure culegerea datelor, perforarea și verificarea lor. Cel mai adesea este suficient un responsabil pentru centralizarea datelor culese de „informaticienii” din secții, care urmărește perforarea și verificarea datelor,

6.2. FAZELE INTRODUCERII SIPAD ÎN ÎNTREPRINDERI

predă pachetele de programe pentru prelucrare și distribuie rapoartele corespunzătoare în secții. Dacă în perioada analizei, proiectării și execuției proiectului noului sistem, aportul specialiștilor din centrul de calcul este hotărîtor, ei fiind de fapt nucleul pe lingă care se formează analiștii de sisteme și programatorii din serviciul de informatică al întreprinderii, implementarea, mersul în paralel și exploatarea noului sistem este aşa cum este și firesc, în ceea mai mare măsură realizarea personalului întreprinderii, rolul centrului de calcul fiind în această perioadă mai mult de pregătire a cadrelor de „informaticieni” în diferite compartimente și a cadrelor de conducere și de intervenție în cazul apariției unor dificultăți în testarea, implementarea și exploatarea noului sistem. Pregătirea cadrelor de „informaticieni” din secții constă în cunoașterea exactă a modului de completare a documentelor pentru culegerea datelor și a utilizării documentelor obținute de la calculatorul electronic. Spre deosebire de specialiștii (programatori și analiștii de sisteme) din cadrul serviciului de informatică, ei nu trebuie să aibă o pregătire strictă de specialitate în domeniul prelucrării automate a datelor. Treptat în serviciile afectate de SIPAD, lucrătorii tehnico-administrativi trebuie să devină în exclusivitate „informaticieni”, pentru a putea asigura exploatarea SIPAD implementat.

Pregătirea cadrelor de „informaticieni” din secții trebuie întărită prin elaborarea de instrucțiuni precise, care să descrie exact documentele și procedurile caracteristice noului sistem. Aceste dosare de instrucțiuni trebuie elaborate de serviciul de informatică și avizate de conducerea întreprinderii.

Pentru ca etapele de introducere a SIPAD să se desfășoare corespunzător, din experiența acumulată pînă în prezent în țară, este necesar ca pentru fiecare aplicație să fie antrenată piramida minimă corespunzătoare din întreprindere, care conține toate activitățile necesare aplicației respective. Conducătorul acestei piramide funcționale existente în întreprindere (planificarea producției, lansarea producției, controlul producției, gestiunea stocurilor, aprovizionare, desfacere etc.), trebuie să fie responsabil acțiunii de implementare și exploatare pentru aplicația respectivă. Prin includerea unor piramide în altele mai mari corespunzător gradului de integrare al aplicațiilor, pas cu pas, conducerea la toate nivelele capătă responsabilitățile corespunzătoare. Este recomandabil ca directorul sau unul dintre directorii adjuncți care răspunde de întreaga acțiune să asigure din timp structura de piramide ce urmează să o folosească în

perioada de implementare și exploatare. Șeful serviciului de informatică acționează pe durata întregului proces ca un consilier al acestuia, iar omologii conducătorilor din vîrfurile piramidelor care răspund pentru fiecare aplicație sunt responsabilii pachetelor de programe, corespunzătoare fiecărei aplicații, din cadrul serviciului de informatică sau centrului de calcul. Discuțiile privind testările de programe cu date reale pentru fiecare aplicație, precum și pentru modificări în cadrul dezvoltării sistemului, trebuie să se poarte de către directorul responsabil al acțiunii de introducere a prelucrării automate a datelor, împreună cu șeful serviciului de informatică, conducătorul piramidei corespunzătoare aplicației și responsabilul pachetului de programe pentru aplicația respectivă.

Durata acțiunii de introducere a SIPAD într-o întreprindere poate varia după cum am văzut pe plaja a cîțiva ani (în medie 3 ani), în funcție de complexitatea întreprinderii și a problemelor din aria stabilită pentru acțiune și în funcție de resurse (personal, gradul de calificare, de experiență etc.). Este evident că sistemul informațional folosind prelucrarea automată a datelor instalat va trebui să suferă corecții corespunzătoare modificărilor care au apărut în întreprindere din momentul începerii analizei și proiectării pînă la terminarea instalării. După terminarea acestor corecții, echipa de analisti poate trece la un nou proiect care să extindă aria de utilizare a echipamentului electronic, să îmbunătățească utilizarea lui, să țină seama de modificările de performanță și cost care survin în mod continuu în legătură cu echipamentele de colectare, transmitere, memorare și prelucrare a datelor și, în general, să dezvolte SIPAD conform dinamicii întreprinderii și în sensul integrării de noi activități în sistem și a integrării cu mediul înconjurător.

Realizarea SIPAD integrate complexe reprezintă un rezultat pentru care nu există, cel puțin deocamdată, o soluție directă. Rezolvarea acestei probleme se face prin metode iterative, pornind de la un sistem mai simplu, format din cîteva subsisteme de aplicații care utilizează baza comună de date pe care îl completăm cu noi subsisteme și îl îmbunătățim continuu. Cel puțin pentru perioada imediată următoare aceasta constituie singura soluție realistă, care ține seama cît și de problemele de personal (atât de proiectare, cit și de utilizare).

Este esențial însă ca procesul de integrare succesivă a activităților din întreprindere să nu reclame reorganizarea bazei de date sau a programelor existente. În acest sens existența unui pachet de pro-

6.2. FAZELE INTRODUCERII SIPAD ÎN ÎNTREPRINDERI

grame pentru organizarea și generarea bazei de date, care să permită și reorganizarea sa automată în funcție de adăugirile sau de noile legături ce pot apărea între fișiere și înregistrări, care să fie utilizat în proiectarea sistemelor informaționale, asigură în afară de această cerință și cadrul de la care se pleacă în proiectarea subsistemelor de aplicații din cadrul fiecărui sistem informațional individual în vederea integrării succesive a acestora.

Trecerea la etapa de exploatare a noului sistem, care este ultima etapă, se face în general treptat prin completarea datelor în fișiere și asigurarea unui sistem informațional de rezervă (fallback) manual sau mecanizat, pentru cazul unor defecțiuni la unele echipamente. Amplarea sistemelor sau procedurilor pentru situații de avarii depinde în principal de calitatea și siguranța în funcționare a echipamentelor folosite și de gradul de automatizare al proceselor de culegere a datelor, transmitere și prelucrare.

De asemenea, nu se poate trece la faza de exploatare a SIPAD decit după completarea tuturor dosarelor de analiză, programare și operare pentru toate aplicațiile din cadrul sistemului informațional. Aceste dosare trebuie să conțină descrierea completă a noului SIPAD, a tuturor programelor, fișierelor, a procedurilor de colectare și transmitere a datelor și utilizarea rapoartelor și a modului de operare pe calculator al programelor și fișierelor necesare fiecărei aplicații. Dosarele de *analiză, programare, operare*, trebuie să însoțească documentația obligatorie pentru exploatare.

Se discută în ultima vreme tot mai mult despre profilul analistului de sisteme și de cunoștințele pe care trebuie să le aibă în domeniul programării. Se consideră pe de o parte că orice analist trebuie recrutat numai dintre programatorii cu experiență (senior programmers), iar pe de altă parte că analistul nu trebuie să aibă decit cunoștințe sumare de programare. În condițiile în care în etapa actuală există pachete de programe pentru aplicațiile principale din întreprindere, precum și pachete de programe sau limbaje pentru generația, organizarea, actualizarea și interogarea bazelor de date care dau cadrul pentru integrarea activităților principale din întreprinderi, considerăm că analistul de sisteme trebuie să aibă cunoștințe suficiente de programare și organizarea bazelor de date pentru a putea să-și înșuesească pachetele existente, pentru a le putea utiliza sau adapta în cadrul proiectului SIPAD pe care este chemat să-l elaboreze. Se poate considera că utilitatea acestui specialist analist-

programator în etapa actuală este în creștere și apreciem că în țara noastră în această etapă, dezvoltarea în acest domeniu este condiționată în primul rind de mărirea numărului și creșterea calității pregătirii acestor specialiști.

În condițiile noi de existență a pachetelor de programe sau limbajelor pentru aplicațiile din întreprinderi și pentru crearea, organizarea, actualizarea și interogarea bazelor de date, considerăm că etapele de analiză și proiectare se vor reduce mult ca durată, pe baza unor activități sistematice de raționalizare și tipizare a documentelor de evidență, indicatorilor, circuitelor informaționale și structurilor, urmând ca efortul principal pentru introducerea de noi SIPAD să fie îndreptat spre adaptarea sau chiar difuzarea metodelor și procedurilor elaborate centralizat. Astfel se poate aprecia că există o conjunctură extrem de favorabilă pentru intensificarea în țară a procesului de introducere a SIPAD în întreprinderile importante.

Asigurarea la nivel central (fig. 6.2.2) a pachetului pentru generarea și organizarea bazelor de date (POBAD) permite, pe de o parte trecerea pe scară largă la pregătirea datelor din întreprinderi în mod unitar, activitate critică ce durează cel mai mult în procesul de introducere a sistemelor informaționale și care în condițiile utilizării în proiectarea tuturor sistemelor informaționale a același pachet de generare și organizare a bazei de date poate începe pe scară națională înainte sau în paralel cu activitatea de proiectare, iar pe de altă parte asigură compatibilitatea între subsistemele fiecărui sistem informațional și între sistemele informaționale (la nivelul bazelor de date) asigurind condițiile ca în perspectivă să se constituie SIPAD tot mai complexe și sistemul unic informațional la scară națională.

Schema generală pentru proiectarea oricărui sistem informațional (micro sau macroeconomic) pentru deservirea conducerii, urmăză să fie următoarea (fig. 6.2.2) în care pentru fiecare sistem informațional se va proiecta în principal numai subsistemele de aplicații (II), pentru celelalte activități (I) furnizindu-se de la nivelul central pachetele de programe sau proiecte cadre.

Se poate aprecia că pachetele în legătură directă cu baza de date ce se furnizează de la nivelul central privind organizarea și generația bazei de date, interogarea bazei de date pentru obținerea, în special de rapoarte prin excepții necesare conducerii și care nu fac obiectul rapoartelor în general periodice prevăzute în cadrul proiectelor subsistemelor de aplicații, actualizarea și întreținerea bazei de

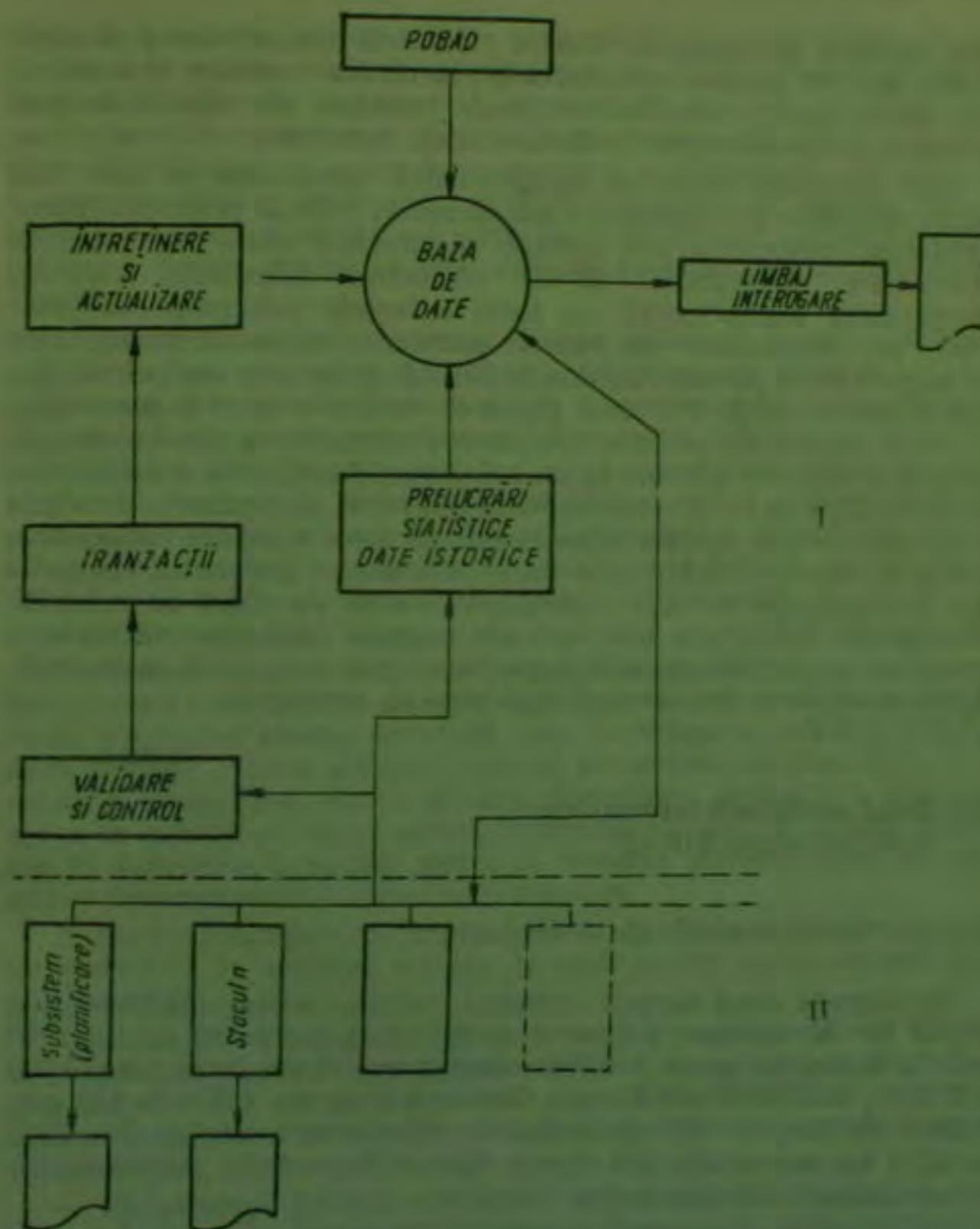


Fig. 6.2.2. Schema generală pentru proiectarea unui sistem informațional.

date, validare și control la nivelul efectuării tranzacțiilor și a organizării datelor istorice în vederea prelucrărilor statistice și a calculelor previzionale, constituie cel puțin jumătate din efortul de proiectare și programare pentru fiecare sistem individual.

Este subînțeles faptul că echipamentele electronice pe care s-au trecut aplicațiile principale din întreprinderi privind producția, aprovizionarea, desfacerea etc. trebuie să prezinte mare siguranță în funcționare, defecțiunile SPAD antrenind dificultăți mari în funcționarea întreprinderii cu toate sistemele informaționale manuale sau mecanizate de rezervă prevăzute în caz de avarie, care nu asigură decit parțial funcțiunile SIPAD și în orice caz pentru durete scurte de timp. Din acest punct de vedere o mare importanță o prezintă rețelele de calculatoare interconectate între ele, în care, în cazul defecțiunilor apărute la un calculator, funcțiunile acestuia sunt preluate automat de alte calculatoare din rețea. Organizarea de rețele de calculatoare și echipamente terminale interconectate cu acestea, permit în cazul utilizării și a unor calculatoare puternice compatibile (cel puțin la nivelul organizării bazelor de date) cu celelalte calculatoare medii sau mici și care asigură gestiunea automată a rețelei să se obțină economii importante care merg pînă la *înjumătățirea costului pentru aceiasi capacitate de prelucrare*.

6.3. Rolul conducerii întreprinderii în introducerea SIPAD

6.3.1. Atribuții generale ale conducerii

În ultimele două decenii numărul calculatoarelor electronice și gradul lor de utilizare a crescut, astfel că la începutul anului 1970 erau în funcțiune peste 110 000 calculatoare, dintre care cca. 73 000 în S.U.A., cca. 24 000 în Europa Occidentală și cca. 6 000 în Japonia. Ritmul de creștere al numărului de calculatoare este de 30—35% anual și tot mai multe țări depun eforturi importante pentru extinderea utilizării calculatoarelor.

Creșterea vertiginosă a numărului de calculatoare se datorează în special rolului pe care acestea îl au ca instrument al conducerii: posibilitățile de informare operativă, exactă și suficientă pe care le prezintă aceste instrumente ale conducerii se impletează cu posibili-

6.3. ROLUL CONDUCERII ÎN INTRODUCEREA SIPAD

tățile de explorare ale diferitelor variante, de realizare a unor studii prospective pentru elaborarea planurilor și actualizarea acestora și cu posibilitățile de elaborare de modele de simulare și optimizări pentru pregătirea și luarea deciziei. Elaborarea și actualizarea planurilor se impletează cu o urmărire operativă a stocurilor, a producției, a aprovizionării și desfacerii, conducătorul obținând operativ informațiile necesare periodic sau prin excepții (abateri de la termene, rebuturi, poziții critice în stoc etc.) și evoluția indicatorilor principali.

După cum am văzut, în cazul cînd se utilizează în cadrul SIPAD o bază de date și cînd se realizează legăturile între activitățile principale ale întreprinderii și modelele corespunzătoare de simulare sau optimizare pentru planificare, programare, lansarea și controlul producției, aprovizionarea, calculele finanțier-contabile, urmărirea stocurilor și desfacerea, în cadrul unui *sistem inchis* (autoadaptiv) de prelucrări corespunzătoare acestor aplicații care se execută de regulă cu ajutorul sistemelor automate de prelucrare a datelor, realizindu-se de asemenei legăturile principale dintre întreprindere și mediul în care își dezvoltă activitatea, avem de-a face cu un SIPAD integrat.

În vederea îmbunătățirii sistemului informațional existent sau introducerii unui SIPAD integrat în întreprindere, condiția necesară, cheie a reușitei acestei activități, este participarea activă a conducerii în toate etapele acțiunii: studiu preliminar, analiza sistemului existent, proiectarea noului sistem, executarea proiectului, funcționarea în paralel a noului sistem împreună cu vechiul sistem, punerea în funcțiune a noului sistem și trecerea întreprinderii la noul sistem informațional și dezvoltarea acestuia.

Acțiunile principale ce urmează să fie desfășurate de conducerea întreprinderii în vederea trecerii la noul sistem informațional sunt următoarele :

- conducerea întreprinderii urmează să se preocupe de pregătirea cadrelor de conducere la toate nivelele, a cadrelor de specialiști și a cadrelor din întreprindere care vor utiliza nemijlocit noul sistem, să pregătească pe clienți și furnizori în vederea acceptării noului sistem ;

- să înțeleagă tehnica sistemelor informaționale bazate pe utilizarea mijloacelor mecanografice sau automate de prelucrarea datelor ;

- să precizeze obiectivele și scopurile în care se va utiliza noul sistem ;

- să organizeze elaborarea unui studiu preliminar privind ne-necesitătile și posibilitățile de realizare ;
- să planifice și să controleze elaborarea SIPAD ;
- să aloce sumele necesare (personal corespunzător, timp, fonduri, echipament etc.) ;
- să organizeze participarea în toate etapele a cadrelor de conducere care este necesar să fie conștiente de implicațiile, timpul și efortul pe care urmează să-l depună participind la elaborarea și introducerea noului sistem informațional în întreprindere ;
- să avizeze proiectul în fiecare etapă de desfășurare și să informeze permanent tot personalul interesat asupra realizărilor și schimbărilor ce au loc ;
- să organizeze și să coordoneze activitățile conducerilor și specialiștilor din domeniul analizei și proiectării sistemelor informaționale ;
- să efectueze schimbările organizatorice privind atât structura întreprinderii, cât și sarcinile personalului, cerute de utilizarea noului sistem.

In acest paragraf ne vom preocupa de atribuțiile ce revin conducerii în introducerea unui SIPAD integrat.

După ce se decide cum să se utilizeze un calculator pe baza studiului preliminar avizat favorabil de conducere și după ce întreprinderea și personalul au fost pregătiți să-l accepte, conducerea urmează să planifice și să controleze elaborarea SIPAD integrat. În condițiile etapei actuale, în țara noastră există puțini specialiști (analisti de sisteme și programatori) pregătiți în întreprinderi și de aceea este necesar să se ia în primul rînd măsura de a pregăti un astfel de colectiv (la centrele de calcul teritoriale, centrele speciale de pregătire a acestor specialiști sau la alte forme de instruire), care va constitui nucleul serviciului de informatică (sau centrului de calcul) din întreprindere și care va efectua, eventual în colaborare cu alți specialiști de la centrele de calcul teritoriale, institute de proiectare în informatică sau centrele de calcul care lucrează în ramura sau subramura din care face parte întreprinderea respectivă, atât proiectul noului sistem și punerea sa în funcțiune, cât și dezvoltarea acestuia. Este important să avem o apreciere realistă a perspectivei activității serviciului de informatică ; practic elaborarea noului sistem nu se sfirșește niciodată și, de aceea, activitatea de dezvoltare a SIPAD integrat este un aspect important de care conducerea trebuie să țină seama. Schimbările care apar în condițiile de activitate ale întreprinderii și

în tehniciile de informare, cer să se facă adăugiri și modificări practice permanente la SIPAD, chiar dacă după punerea în funcțiune a noului sistem aceste modificări au o pondere relativ mai mică.

Și în conducerea proiectului SIPAD este recomandabil să se aplique principiile de conducedere care se aplică oricărei activități ; conducederea întreprinderii urmează să stabilească obiectivele, să formuleze politicile, să repartizeze resursele, să aprobe planurile și să urmărească realizarea acestora la termene.

În procesul de stabilire al obiectivelor acțiunii de introducere a noului sistem informațional este necesar să se precizeze în ce scopuri vor fi utilizate informațiile, cine va primi informații, ce informații și cînd ; de asemenea este necesar să se stabilească ce date, cine și cînd furnizează datele de intrare necesare prelucrărilor pentru obținerea informațiilor.

Precizarea rapoartelor necesare pregătirii deciziei la toate nivelele întreprinderii, care sunt indicatorii și modul de urmărire al acestora de către conduceri, urmează să constituie de asemenea preocupări de bază ale conducerii, deoarece definirea exactă a acestor „ieșiri“ din SIPAD determină în mare parte structura acestuia.

O atenție deosebită urmează să acorde conducederea, organizării datelor primare (fișierele de bază : structura produselor, tehnologii, materialele etc.) deoarece din experiența acumulată pînă în prezent drumul critic al graficului activităților elaborării și introducerii SIPAD conține în mod necesar această activitate de lungă durată și care reclamă resurse importante în oameni și timp. *Punerea în ordine a datelor primare, codificarea și eliminarea erorilor existente și încărcarea cu date corecte a fișierelor de bază* sunt activitățile care determină în cea mai mare măsură durata procesului de introducere a noului sistem informațional. De aceea, este bine ca pe cît e cu putință aceste activități să se organizeze înainte sau în paralel cu alte activități (proiectare, programare, instruire etc.).

Este recomandabil ca să fie numit de conducedere un conducedător al proiectului, căruia să i se definească sarcinile și responsabilitățile, precum și relațiile cu ceilalți conducedători. O greșeală care a apărut în acest sens pînă în prezent, este numirea unui director din conducederea întreprinderii ca șef de proiect, fără ca să fie degrevat de o parte din sarcinile anterioare, măcar pe durata proiectului. În acest mod, conducedătorul respectiv nu poate participa în măsură suficientă la elaborarea și introducerea noului sistem, neparticiparea în măsură

suficientă a conducerii în toate etapele, atrăgind pierderi de resurse, reveniri, negăsirea celor mai eficiente soluții etc.

Este recomandabil, de asemenea, să se creeze un „Comitet de conducere al proiectului”, din care să facă parte conducerii din posturile cheie care lucrează nemijlocit cu noul sistem informațional (virfurile piramidelor corespunzătoare aplicațiilor) și care să discute și să decidă în fiecare etapă asupra schimbărilor, fiind astfel implicat în întreaga acțiune. Pe această bază se pot lua deciziile cheie în procesul de introducere a sistemului cum sunt: structura bazei de date, numărul și ordinea în care trebuie introduse subsistemele SIPAD, măsurile ce trebuie luate în fiecare etapă pentru introducerea subsistemului respectiv, resursele ce trebuie alocate și modul de control.

În cadrul Comitetului de conducere al proiectului este recomandabil să se acorde de asemenei o atenție deosebită alcătuirii documentației complete și amănunțite a tuturor proiectelor, procedeelor, formularelor, fișierelor și programelor, pentru a ușura difuzarea elementelor de cunoaștere necesare și transferul activității de la un specialist la altul.

Urmărirea metodică a progreselor realizate, analizarea schimbărilor de obiective, a modificărilor de resurse, a modului de organizare a datelor etc., constituie preocupări ale conducerii, care îi crează baza de înțelegere necesară pentru acordarea sprijinului permanent în rezolvarea conflictelor dintre conducerii de la diferite nivele sau dintre acestia și serviciul de informatică. Noile reguli de luarea deciziei și noile circuite informaționale care afectează limitele tradiționale ale responsabilității conducerii, generează permanent probleme grele care nu pot fi rezolvate decât pe baza unei analize și pregătiri susținute și prin participarea nemijlocită a conducerii întreprinderii. Rezistența sau chiar indiferența conducerilor care de obicei se reflectă imediat și în atitudinea restului personalului, produc întirzieri sau pot conduce la compromiterea întregii acțiuni.

Conducătorii care sunt convinși de utilitatea noului sistem trebuie să participe nemijlocit la elaborarea acestuia. SIPAD nu poate și nu trebuie să fie numai opera specialiștilor în informatică; încercările de acest gen au condus aproape întotdeauna la o dezamăgire generală.

Este recomandabil ca conducerii să specifice cât mai exact informațiile pe care le doresc și când le doresc. Este foarte util de asemenea ca ei să participe împreună cu specialiștii la elaborarea procedeelor de colectare a datelor și a modului de încărcare a acestora în fișiere. Ei urmează să-și adapteze organizarea compartimentului în

aceea fel încât să ciștige cât mai multe avantaje din partea noului sistem informațional pentru conducere; este necesar de asemenea să convingă personalul de utilitatea noului sistem, să-l informeze de schimbările ce vor survie și de motivele acestora, asigurându-se că personalul este pregătit corespunzător pentru a utiliza nou! sistem și a furniza informațiile necesare sistemului.

Toate aceste acțiuni reprezintă condiții necesare pentru succes și trebuie remarcat că ele urmează să fie îndeplinite în general suplimentar față de sarcinile curente de producție. De acceptarea de către întreg personalul a îndeplinirii acestor sarcini, de obicei peste sarcinile curente, de modul în care conducerea rezolvă aceste dificultăți care reclamă adesea recalificarea unei părți din personalul existent, depinde în bună măsură succesul acțiunii.

Formarea specialiștilor în informatică, analiza sistemului existent și proiectarea SIPAD integrate, elaborarea programelor și dezvoltarea acestora reprezintă jumătate sau mai puțin de jumătate din întreaga acțiune. Ponderea principală a dificultăților revine organizării datelor primare, participării conducerii la toate nivelurile, pregătirii și antrenării personalului în utilizarea noului sistem și efectuării schimbărilor organizatorice în întreprindere, ca atare implementării SIPAD.

Este necesar ca conducerii să analizeze pe ce bază obiectivă îau decizii și să facă un efort personal, adesea important pentru a cunoaște posibilitățile noului sistem și să desfășoare o muncă permanentă de conducere a proiectului; trebuie să fim convinși că numai prin desfășurarea unui efort comun răspunzător vom putea obține rezultate bune. SIPAD nu este un cadou pe care îl primesc conducerii. Aceștia urmează să desfășoare o activitate complexă care cel mai adesea se adaugă la sarcinile curente.

6.3.2. Nivelele SIPAD integrat și reflectarea lor în atribuțiile conducerii

Singura cale prin care un SIPAD poate deveni cu adevărat eficient este precizarea sistemului de decizii din cadrul sistemului de conducere. Cadrele de conducere sunt singurul for capabil să identifice acele decizii care aduc o contribuție importantă la realizarea obiectivelor întreprinderii. De aceea este necesară o pregătire corespunzătoare.

punzătoare a conducerii în domeniul teoriei și funcționării sistemelor de decizii ale conducerii pentru a putea rezolva problemele legate de :

- diferențierea diverselor nivele de decizii (nivelul conducerii superioare, medii și operative),
- identificarea părților componente ale deciziei (obiective, date previzune, evaluare),
- identificarea etapelor în procesul de elaborare a decizilor și
- determinarea decizilor care pot fi programate (parțial automatizate).

Determinarea sistemului de decizii în întreprindere este o condiție necesară în vederea elaborării sistemului de conducere. Există următoarele nivele ale unui SIPAD integrat :

- a. de evidență și raportare ;
- b. de urmărire și control ;
- c. de previziune și planificare ;
- d. de integrare (integrat).

Conducerea trebuie să fie conștientă de nivelul la care formulează obiectivele SIPAD din întreprindere, deoarece de această decizie depinde evoluția ulterioară a întregului program.

În cadrul SIPAD de evidență și raportare, se înregistrează toate evenimentele care au loc pe o perioadă definită a sistemului și se oferă o sinteză a realizărilor (sau abaterilor) pe perioada respectivă sau rapoarte prin excepție. La fiecare nivel de decizie se determină gradul de sinteză al informațiilor necesare.

În cadrul SIPAD de urmărire și control, se realizează circuitele inverse în cadrul sistemului și se automatizează parțial unele nivele de decizie operativă. Astfel, de pildă, un sistem autoadaptiv de control al stocurilor, implică din partea nivelelor de decizie operativă numai un control pentru deciziile de rutină pe care SIPAD le furnizează în mod automat.

Sistemul realizat la nivel de urmărire și control, elaborează de asemenea rapoarte suplimentare față de primul nivel privind, de pildă, realizările de plan, punind în evidență abaterile, rebuturile etc.

În cadrul sistemelor realizate la nivel de previziune și elaborare de variante de planuri, se analizează activitățile din trecut pentru a face previziuni în legătură cu ceea ce este probabil să se întâmple în viitor, utilizându-se modele de cercetare operațională, tehnici de prospectare a pieței etc.

Sistemele integrate reunind caracteristicile nivelelor precedente și alte modele de simulare și optimizări, care pe lângă performanțe op-

time pentru activitățile din întreprindere, realizează și legături optimale ale sistemului cu mediul înconjurător (integrarea cu alte sisteme, informaționale) au drept scop găsirea posibilităților pentru mărirea producției și folosirea căi mai eficiente a resurselor existente. Pentru aceasta este necesar ca activitățile principale din întreprindere să fie corelate în cadrul SIPAD, astfel ca acestea să se comporte ca un sistem închis care să se adapteze rapid la abaterile față de plan, în care să se poată urmări performanțele și în care să se simuleze comportarea întreprinderii în următoarea perioadă de plan. În acest sens, pe de o parte modelele de programare a producției, controlul stocurilor etc., trebuie să acționeze în sensul optimizării activităților respective (scurtarea ciclului de fabricație, micșorarea stocurilor, încărcarea optimă a mașinilor etc.) iar pe de altă parte sistemul trebuie să poată face față în mod operativ perturbațiilor care pot apărea față de condițiile inițiale (dificultăți în aprovizionare, chiar în condițiile în care planul de aprovizionare poate fi cunoscut cu suficient timp înainte la cererea conducerii, schimbarea unor contracte, modificări în sume etc.).

Pe lângă acestea, modelele de previziune, planificare, prospectare a pieței etc., considerăm că trebuie elaborate ținând seama și de specificul unui sistem la scară națională de previziune și planificare. Îmbinarea în cadrul acestui sistem a subsistemelor de previziune și planificare ale întreprinderilor, este o sarcină importantă atât a conducerilor la diferite nivele, cât și a specialiștilor în planificare și informatică. Prin aceasta se va realiza în măsură mai mare gradul de integrare al activităților întreprinderilor și a legăturilor sale cu mediul înconjurător.

În întreaga lume sunt puține sisteme care să lucreze la nivel integrat, care să includă subsistemele de previziune și elaborare de plan, care necesită un înalt grad de analiză a procesului de decizie și investiții mari de capacitate de conducere, de efort de analiză și proiectare a sistemului și de cheltuieli generale pentru reorganizare.

Subliniem că SIPAD, la orice nivel ar fi realizat, nu absolvă pe conduceri de responsabilitățile conducerii, el fiind un instrument care poate susține un sistem bun de planificare și control, dar nu poate determina efectuarea unei planificări bune, el putând fi utilizat de conduceri la crearea unui sistem bun de planificare și control, dar neputind forța desfășurarea unei bune activități de planificare și control.

Este util ca și cadrele de conducere din întreprindere să cunoască capacitatele calculatoarelor, posibilităile și în special limitele de exploatare ale acestor capacitați. În acest sens, un conducător din întreprindere trebuie să învețe el însuși la ce trebuie să se aștepte, luând parte la cursuri de evaluare și studiind experiența altora care utilizează calculatorul. O cauză importantă a nereușitei introducerii calculatorului este achiziționarea echipamentelor fără a avea precise informații pe care acesta trebuie să le furnizeze. Fără elaborarea unui studiu preliminar în care să se definească și să se fundamenteze configurația necesară, întreprinderea va primi o configurație nepotrivită, fiind obligată în continuare să elaboreze subsisteme incompatibile din punct de vedere al integrării activităților. Este suficient, de pildă, ca utilizatorii să subdimensioneze unele programe, neținând seama de excepțiile care apar față de programe analogice existente pentru ca echipamentele din configurația instalată să se dovedească nepotrivite scopurilor propuse. Dacă la elaborarea sistemului, resursele sunt insuficiente, realizarea proiectului rămîne în urmă foarte repede și este greu să se repare aceste întîrzieri, luându-se măsuri excepționale.

De asemenea, dacă conducătorii nu-s implicați în măsură suficientă în acțiune, ei nu vor fi pregătiți pentru a lua decizii, pentru a stabili prioritățile și a participa la schimbările organizatorice necesare.

Integrarea subsistemelor producție, comercial, contabilitate etc., schimbă nivelele de autoritate și centrele de decizie. Fără o pregătire și un acord deplin prealabil, conducătorii nu vor accepta aceste consecințe. De asemenea, fără o comunicare continuă și o răspîndire a cunoștințelor între toți factorii interesați nu se realizează o conjugare a eforturilor în vederea integrării activităților.

Cauzele care pot conduce la nereușită se pot întîlni chiar în organizarea necorespunzătoare a compartimentului de informatică. Un personal subdimensionat sau o pregătire insuficientă stinjenește activitatea. Fără luarea de măsuri privind numărul corect al celor care răspund de o aplicație se pot prevedea mari întîrzieri în desfășurarea lucrului atunci cînd apar situații neprevăzute.

Conducătorii nu trebuie să credă că preocupările lor pentru SIPAD integrat scad atunci cînd acesta intră în funcțiune. În primul rînd am văzut că realizarea tuturor nivelelor unui SIPAD integrat reclamă un proces indelungat chiar dacă se obțin rezultate eficiente

utilizînd numai o parte dintre nivelele menționate. Dar, după cum s-a mai arătat, îmbunătățirea unui SIPAD integrat nu se sfîrșește practic niciodată. Procedeele și programele se schimbă pe măsură ce activitățile întreprinderii dezvoltîndu-se, modifică cerințele de informare ale conducătorului. Ca o regulă generală sistemele informative cer o analiză și o reproiectare cam la fiecare 4–5 ani pentru a elimina neajunsurile ce se datorează unei succesiuni de modificări.

Trebuie să fim conștienți că introducerea și dezvoltarea SIPAD integrat necesită o conducere permanentă ca orice altă activitate de bază din întreprindere.

Este important ca în planificarea introducerii unui SIPAD integrat, să se prevadă ca încă din primele etape, atît nivelul sistemului, cît și numărul subsistemelor de aplicații ce se realizează să permită obținerea de rezultate în exploatare și rapoarte care să creeze un climat general favorabil măririi treptate a sistemului prin adăugarea de subsisteme de aplicații și trecerea la nivele superioare pînă la intrarea în funcțiune a sistemului integrat.

Pentru a putea desfășura acțiunile în acest mod este necesar ca în proiectare să se respecte principiul de bază și anume realizarea unui sistem a cărui dezvoltare treptată să nu reclame reorganizarea bazei de date și nici modificarea programelor elaborate anterior. După cum am văzut există diferite metode (fig. 6.2.2.) de proiectare prin care se asigură acest avantaj esențial, ele utilizînd toate programe sau limbi existente care asigură organizarea, actualizarea și interrogarea bazei de date, practic independente de tipul de calculator utilizat și de subsistemele de aplicații.

În acest mod se poate realiza de asemenei compatibilitatea între bazele de date ale diferitelor sisteme, utilizînd aceleasi programe pentru generarea și organizarea lor și utilizînd aceeași structură pentru fișierele principale comune grupurilor de întreprinderi sau întreprinderilor dintr-o ramură sau dintr-un sector al economiei naționale. Asigurînd compatibilitatea la nivelul bazelor de date între diferite sisteme de informare a conducerii cu mijloace de automatizare, se pot constitui treptat sisteme tot mai complexe și în perspectivă sisteme la nivel național.

Rolul conducerilor întreprinderilor în urmărirea atenții a acestor caracteristici în cadrul proiectelor este foarte important. Asigurarea unei concepții unitare asupra organizării datelor în bazele de date, constituie un factor esențial și în acțiunea de analiză a indicatorilor

și de raționalizare a documentelor de intrare sau de raportare, acestea constituind „intrări” și respectiv „ieșiri” ale bazelor de date. Experiența conținută în pachetele existente de generare și organizare a bazelor de date și în descrierile (documentația) structurilor fișierelor de bază constituie o bună bază de plecare în activitatea de raționalizare și proiectare, ea reprezentând o prezentare exhaustivă a informațiilor necesare în baza de date pentru toate activitățile întreprinderilor, informații care trebuie furnizate de „intrări” și care urmează să fie prelucrate pentru obținerea „ieșirilor”. Acțiunea de raționalizare și standardizare a documentelor de evidență și analiza indicatorilor poate primi astfel un ajutor important în modul descris, considerind problema în ansamblul ei și urmărind aplicarea principiilor de bază în proiectarea sistemelor informaționale compatibile cu mijloace manuale, de mecanizare sau de automatizare.

În ultimii ani se conturează tot mai mult pe plan mondial că cel mai rațional mod de abordare a proiectării și introducerii sistemelor de conducere în întreprinderi este elaborarea centralizată în institute centrale sau firme producătoare de calculatoare, firme de consultanță, firme de creație de programe etc., a programelor complexe sau limbajelor legate de organizarea, actualizarea și interogarea bazei de date (zona I în fig. 6.2.2) care nu depind de numărul și specificul subsistemelor de aplicații; folosind aceste pachete și proiecte cadre, se elaborează specific pentru fiecare beneficiar subsistemele de aplicații. Aceste pachete elaborate centralizat se pot considera ca niște prelungiri ale sistemelor de operare cu diferența că pot fi utilizate și practic independent de tipul de calculator, aspect care favorizează compatibilitatea între sistemele informaționale.

In acest mod, după cum am văzut, se realizează o economie importantă de muncă socială privind proiectarea și executarea proiectului pentru fiecare sistem informațional.

De asemenei, este recomandabil ca proiectarea să se facă în vederea introducerii în perspectivă a sistemului integrat de conducere cu mijloace de automatizare, chiar dacă sistemul va urma să funcționeze la nivel de evidență și raportare sau de urmărire și control un timp relativ indelungat. Această concepție permite să se realizeze obiectivele importante menționate mai sus privind atât evitarea reorganizării bazelor de date și refacerii programelor, cît și asigurarea compatibilității sistemelor de conducere la nivelul bazelor de date în vederea constituirii de sisteme tot mai complexe în perspectivă.

6.4. Probleme și opinii relative la actuala etapă de introducere în țară a SIPAD

Dezvoltarea sistemului informațional al întreprinderii moderne a făcut ca în ultimele decenii conducerea întreprinderii să necesite prelucrarea unei mari cantități de date privind starea și evoluția atât a fenomenelor interne, cît și a celor externe întreprinderii. Este tot mai necesară îmbunătățirea continuă a sistemului informațional, a conducerii și organizării întreprinderii în vederea fundamentării tehnice a elaborării deciziilor și a intervenției rapide în conducerea proceselor. În ultima vreme, s-au conturat și în țara noastră noi domenii de cercetare, ca teoria informației, cercetarea operațională, teoria simulării, proiectarea și implementarea sistemelor informaționale etc., domenii care ajută la modelarea proceselor tehnice și economice în scopul măririi producției și a productivității muncii și care oferă instrumente noi, eficace, conducerii întreprinderii [12].

Dinamica întreprinderii moderne, prin mărirea dimensiunilor și prin dezvoltarea complexă a dus la fragmentarea funcțională a acestia în compartimente aparent autonome, conduse cu scopuri, de multe ori neconvergente, cît și la o deosebită complicare a sistemului informațional, atât în interiorul compartimentelor, cît și între acestea; conducerea întreprinderii impune, în limitele metodelor clasice, „filtrarea” fluxului informațional la diferite nivele „pe verticală”, această filtrare introducind de multe ori anumite interpretări sau deformări.

Dificultățile conducerii întreprinderii moderne prin metode clasice rezidă în dificultățile legate de luarea deciziei și de urmărire a consecințelor acesteia. Vom deosebi printre aceste dificultăți, intinute în majoritatea întreprinderilor din țară, greutățile legate de stăpînirea fluxului informațional și greutățile legate de dirijarea optimă a sistemului întreprinderii. Printre dificultățile legate de stăpînirea fluxului informațional, menționăm dificultățile legate de informarea operativă apariția unor informații greșite, nesigure sau nesemnificative — cel mai adesea datorită filtrelor pe verticală care le „selectează” — apariția unor informații contradictorii — pe „canale” diferite — informații ce apar cu întârziere, deci sunt inoperante etc.) și de imposibilitatea conducerii prin excepții, dificultăți legate de aprecierea în ansamblu a informațiilor, datorite greutăților în obținerea unor sinteze informaționale, dificultăți legate de aprecierea

activității în contextul întregii întreprinderi, dificultăți legate de estimarea complexă a dinamicii condițiilor exterioare întreprinderii, dificultăți legate de urmărirea exactă și complexă a tuturor consecințelor unei decizii (în condițiile apariției unor factori aleatori perturbatori) etc. Dificultățile legate de dirijarea optimală a sistemului întreprinderii rezidă în elaborarea unor decizii optimale în cadrul unui sistem informațional dat, care să permită modificarea dinamicii proceselor vitale din întreprindere în diferite sensuri, precizate prin diferite criterii de optim.

Sistemele de prelucrare automată a datelor, după cum am văzut, sunt utilizate în scopul depășirii acestor dificultăți. În acest paragraf, vom studia în special aplicațiile pe SIPAD în cadrul organizării și conducerii sistemului informațional al întreprinderii, fără a insista asupra metodelor de optimizare ce urmăresc dirijarea optimală a proceselor. Evident, diferența dintre „informațional” și „dirijare optimală” este, într-un sens, artificială, deoarece, în fond, informatica urmărește tot dirijarea optimală ulterioară. Credem însă necesar să precizăm că nu totdeauna o bună informare este suficientă pentru decizie: de multe ori nu este evident „ce” și „cum” trebuie modificat, pentru satisfacerea unor criterii. Distincția pe care am introdus-o, în scopul sistematizării expunerii și a prezentării celor elemente care le considerăm esențiale, urmărește să delimitizeze numai aplicațiile care informează conducerea sau în care deciziile se obțin direct pe baza informațiilor, de aplicațiile care impun, în același scop, construirea unor modele simbolice (matematice) sau analogice (simulări) și o activitate sistematică de analiză a obiectivelor la diferite nivele ierarhice ale întreprinderii și a elementelor care conduc la luarea deciziilor.

După cum am văzut, sistemul informațional al întreprinderii este conceput ca ansamblul de procedee și mijloace de colectare, transmisie și prelucrare a informației necesare procesului de conducere; sistemul de conducere (SIC) este ansamblul datelor, rapoartelor și al prelucrărilor necesare în vederea prezentării informațiilor, elaborării variantelor, studiul variațiilor diforitor indicatori, utilizarea modelelor etc., care ajută la luarea deciziei. Desigur, între SIC și SIPAD sunt întrepătrunderi, ieșirile din SIPAD constituind baza de informație pentru sistemul de conducere. De asemenea, gradul de automatizare al sistemului de conducere depinde direct de modul cum a fost conceput SIPAD. De aceea, precizarea cerințelor sistemului de conducere sunt elemente de bază necesare proiectării SIPAD.

6.4. PROBLEME DE INTRODUCERE ÎN ȚARĂ A SIPAD

Cele dintii sisteme informaționale care nu s-au dezvoltat spontan, ca atare la baza organizării cărora a stat un plan sistematic de organizare, au apărut în primele decenii ale secolului nostru. Consolidarea și organizarea lor sistematică a inceput abia după 1950, ca urmare a dezvoltării mijloacelor și metodelor de automatizare a lucrărilor de prelucrare a datelor.

Sistemele electronice de prelucrare a datelor sint folosite astăzi în cele mai diferite domenii. O listă publicată în 1966 indică un număr de aproape 2.000 domenii pornind de la calcule simple de salarii, pînă la conducederea rachetelor balistice și a zborurilor cosmice. Din toate statisticile publicate reiese că majoritatea absolută (circa 60%) a sistemelor de prelucrare automată a datelor sint folosite pentru conducederea întreprinderilor (a sistemelor de producție, comerciale, bancare etc.).

In acest domeniu al conducediei, sistemele moderne de calcul sint folosite:

- pentru prelucrarea datelor necesare conducediei;
- ca auxiliar în procedeele de proiectare, atît în ce privește utilizarea cercetării operaționale (calcule PERT, programare liniară etc.), cît și pentru probleme de dimensiune;
- ca sisteme de comandă a proceselor tehnologice.

Este evident că utilizarea sistemelor de prelucrare pentru a acoperi unul din aspectele de mai sus pune probleme a căror rezolvare este specifică pentru fiecare în parte. Practic orice sistem de producție în epoca noastră ridică probleme tot mai complexe în ceea ce privește conducederea sa către un nivel de eficiență maximă față de perturbațiile interne și externe întreprinderii. Această necesitate împreună cu rezultatele de mare eficiență ce se obțin, justifică tendința utilizării preponderente a sistemelor de prelucrare automată a datelor în domeniul conducediei producției.

Întreprinderile din țară sint confruntate cu prelucrarea unei mari cantități de informații datorită complexității producției și variabilității condițiilor externe uzinelor. Conducederea devine dificilă din acest punct de vedere, deoarece tot mai des se pot observa fenomene ca:

- ocuparea timpului organelor de conducedere în ședințe de informare sau citire de rapoarte care nu conțin informațiile esențiale pentru conducedere sau în încercări de a modifica condițiile create în urma perturbațiilor care apar, atît în planificare, cît și în desfașoare, aprovizionare, resurse etc.;
- obținerea cu întârziere a informațiilor necesare luării deciziei;

- informații incomplete sau inexakte obținute pe mai multe canale;
- lipsa posibilității unor aprecieri de ansamblu a activităților întreprinderii.

Utilizarea sistemelor automate de prelucrare a informației schimbă situația existentă nu numai în sensul îmbunătățirii calitative a informației pentru luarea deciziei, dar și a centralizării și agregării informației de decizie. În cazul în care informațiile primare pot fi obținute de la sursa lor în fiecare moment, atunci un sistem automat de colectare, transmitere și prelucrare a datelor poate analiza și sintetiza datele și poate prezenta rezultatele conducerii în minimum de timp și cu minimum de erori. Centralizarea informației și a deciziilor are două consecințe importante :

În primul rînd permite studierea de ansamblu a unui sistem de producție, permite integrarea tuturor activităților de producție — lucru care reprezintă un mare progres în comparație cu practica de conducere fragmentată a unor activități compartimentate în mod artificial.

În al doilea rînd implicațiile deciziilor luate de conduceri sunt mai importante (ele referindu-se la întregul ansamblu al întreprinderii). În asemenea circumstanțe, responsabilitatea conducerului crește foarte mult ca și nevoia lui de mijloace mai perfecționate de informare pentru a lua deciziile respective și a urmări efectul lor [13].

În ultimii ani, în țara noastră a început să se dezvolte analiza și proiectarea sistemelor informaționale prin progresele făcute în domeniul economic, al cercetării operaționale, al științei conducerii și al dotării cu echipamente de mecanizare și de prelucrare automată a datelor. Aplicarea eficientă a acestor concepte moderne este strins legată de utilizarea sistemelor de prelucrare automată a datelor, care devin tot mai mult mijlocul material de realizare a „centralizării informației” și instrumentul care transformă sistemul de producție într-un sistem cibernetic care să răspundă obiectivelor căi mai aproape de optim; utilizarea sistemelor electronice de calcul în conducedrea producției reprezintă un instrument deosebit de necesar și util și totodată un mijloc care permite exercitarea pe o altă bază — de ansamblu — a conducerii.

Procesul funcțional din cadrul sistemului informațional comportă trei faze principale :

- colectarea datelor, definită ca înregistrarea datelor de la sursele de informație ;

— transmiterea datelor, adică vehicularea lor de la locul de colectare la cel de prelucrare și invers și în general de la un punct la altul al circuitelor de informații ;

— prelucrarea datelor pornind de la datele de intrare (primare) pe baza unor procedee elaborate (algoritme) în vederea obținerii rezultatelor (date de ieșire).

După modul de prelucrare (fiecare fază a procesului de prelucrare poate fi realizată manual, mecanizat sau automatizat), mijloacele de prelucrare se încadrează în două mari categorii :

a. Mijloace de mecanizare care servesc la prelucrarea datelor primare. Ele sunt constituite din dispozitive capabile să execute o serie de operații de selectare, sortare, ordonare, fuziuni și actualizări de fișiere (pe cartele) și operații aritmetice simple, cu sau fără înregistrarea rezultatelor și să elaboreze tabele centralizatoare de date.

b). Mijloace de automatizare sau sisteme de prelucrare automată ce dispun de o mare capacitate de memorare și execută prelucrări complexe cu mare viteză și cu grad ridicat de precizie.

Mijloacele de mecanizare folosite astăzi pot fi clasificate în echipamente de mică mecanizare (mașini de calcul de birou destinate utilizării individuale pe locuri de muncă); echipamente de medie mecanizare (mașini de contabilizat și facturat folosite la înregistrarea și gruparea unui număr relativ mare de date, efectuind operații aritmetice pe bază de comandă-program); echipament de mare mecanizare (mașini cu cartele perforate), organizat cel mai adesea în stații mecanografice.

Aceste tipuri de echipamente sunt completate aproape întotdeauna cu mașini auxiliare (mașini de scris și de multiplicat documente, dispozitive de codificat și de decodificat, mijloace de transmitere etc.) care, fără a servi direct la prelucrare măresc eficiența mașinilor menționate.

Primele echipamente cu cartele perforate au apărut la sfîrșitul secolului trecut și s-au dezvoltat calitativ în special în deceniile al treilea și al patrulea ale secolului nostru. Utilizarea lor continuă și astăzi în special pentru întreprinderi mici și mijlocii în toate țările industriale. Prelucrarea cu acest tip de echipamente se caracterizează printr-o secvență de operații manuale și mecanizate, conform unui „flux de circulație” și unor „programe cablate”, dinainte stabilite pentru fiecare problemă în cadrul sistemului informațional proiectat.

Cu toate avantajele pe care le oferă (o mai bună valorificare a informației, o organizare de sine stătătoare a unei părți importante a

fluxului informațional, o rationalizare și o simplificare a documentelor purtătoare de informație), totuși echipamentele cu cartele perforate nu pot satisface necesitățile de informație ale unei conduceri eficiente într-o întreprindere modernă complexă. Introducerea acestor echipamente în cadrul sistemului informațional este necesar să facă obiectul unui studiu complex, iar în condițiile actuale, comparativ cu posibilitatea utilizării unui sistem automat de prelucrare. La alegerea soluției trebuie să se țină seama de dezavantajele acestor sisteme, printre care viteza redusă de lucru, timpul de prelucrare lungi și ingăli, neficacitatea privind efectuarea de lucrări ce presupun prelucrări complexe și volume mari de date și repetarea acestora după perioade mici de timp.

Pe de altă parte, avantajele pe care le oferă posibilitatea (ipizări pe scară largă a unor aplicații ca *inventarul, facturarea, calcule contabile, lansarea etc.*, cu mijloace de mecanizare relativ mult mai puțin costisitoare decât echipamentele automate, nu sunt de neglijat.

În sistemele informaționale din întreprinderile din țara noastră, existente în această etapă, se intilnesc o serie de deficiențe care conduc și la necesitatea utilizării pe scară mai largă a mijloacelor de mecanizare și automatizare a prelucrării informației.

În sistemele informaționale actuale circulă o mare cantitate de informații, care necesită cheltuieli și eforturi mari și care nu sunt valorificate în mod corespunzător, conducerile întreprinderilor fiind informate incomplet și cu intirziere. Pe de altă parte, conducerilor își pierd un timp apreciabil în munca de conducere operativă fiind de multe ori puși în situația de a rezolva probleme care revin nivelurilor inferioare.

Deficiențele importante ale sistemelor informaționale actuale se datorează în principal modului de constituire al lor. „Evidențele” actuale (tehnice-operative, contabile și statistice) sunt caracterizate printr-o mare diversitate de documente al căror conținut informațional este greoi și necesită un mare volum de muncă pentru completarea lui. Fișierele de informații privind produsele, ansamblurile și reperele, operațiile tehnologice, materialele etc., prezintă adesea erori (duplicări, ambiguități, lacune) și care cel mai adesea pot fi remediate numai în etapele introducerii SIPAD, pe baza utilizării SPAD. Aspectul selectiv al informării este neglijat într-o mare măsură, nivelele superioare de conducere a întreprinderii primind situații cu multe date care, fiind contradictorii, devin inutilizabile pentru procesul de conducere. Numărul mare de documente existente și ai erorilor care le însoțesc se datorează în esență modului de lucru manual care a

permis introducerea treptată de documente greșite și acumularea de erori. Prezența acestor erori face în multe situații imposibilă informarea conducerii asupra unor factori vitali pentru activitatea întreprinderii și lungeste artificial ciclul de fabricație.

Circulația informațiilor are o viteză redusă, încât de multe ori datele necesare luării unei decizii ajung la conducer la după ce momentul oportun a trecut. Situația se datorează de multe ori incărcării circuitelor de informare cu date și prelucrări ale acestora nesemnificative.

Viteza redusă de circulație a informațiilor și modul greoi de inventariere al informațiilor fac ca perturbațiile care apar în cursul unei perioade de activitate pentru care s-au luat decizii, să nu mai poată fi stăpinate de conducer, care împreună cu ceilalți factori din întreprindere încearcă în mod euristic să facă față noilor condiții, acești factori conducând la supraîncărcarea conducerii cu muncă operativă a cărei eficiență în condițiile slabii informări este redusă.

Existența unor informații care circulă de la nivelele inferioare la cele superioare ale conducerii fără a servi elaborării unei decizii care reprezintă situații nesemnificative sau sint redundante, este alături de aspectele menționate mai sus sursă de complicare a circuitelor de informare și de supradimensionare a sistemului cu documente și personal. În acest sens se pot da numeroase exemple din domeniul aprovizionării, pregătirii fabricației, contabilității materialelor, ordonanțării (programării operative a producției) etc.

Deficiențele sistemului informațional actual se datorează după cum am menționat într-o mare măsură și dotării insuficiente cu mijloace de mecanizare și automatizare, precum și neutilizării raționale a unora din mijloacele existente.

Echipamentele cu cartele perforate, deși într-un număr redus, sunt de o mare diversitate și numai o parte relativ mică dintre întreprinderi dispun de stații mecanografice. Echipamentele de automatizare — calculatoarele electronice — din dotarea economiei naționale sunt insuficiente față de potentialul industrial al țării noastre. Unele dintre aceste calculatoare au o capacitate mică de memorare și dispozitive lente pentru efectuarea calculelor, iar în unele pentru introducerea datelor și extragerea rezultatelor, și ca atare nu permit configurații nu există echipamentele periferice cum sunt memorile de mare capacitate pe discuri sau benzi magnetice și ca atare nu permit constituirea bazelor de date sau folosirea fișierelor cu informații de bază (fișierul de articole — produse, repere — fișierul operațiilor tehnologice, fișierul de materiale etc.) strict necesare prelucrării pe

sistemul automat de prelucrare a datelor, în sfîrșit nu dispun de echipamente suficiente pentru pregătirea datelor (perforatoare și verificatoare de cartele) sau colectarea datelor (pentru urmărirea producției, a stocurilor etc.).

O altă cauză a deficiențelor sistemelor informaționale actuale rezidă și în faptul că avem încă un număr insuficient de specialiști în domeniul prelucrării automate a datelor, mai ales în domeniul analizei și proiectării sistemelor folosind mijloacele de mecanizare și automatizare și în elaborarea programelor.

Utilizarea eficientă a mijloacelor moderne de culegere, transmisie și prelucrare impune o reconsiderare a modului actual de prelucrare în sensul ridicării lui pe o treaptă calitativ superioară. Introducerea acestor mijloace în scopul mecanizării și automatizării sistemelor informaționale actuale aduce o îmbunătățire numai în prima etapă a acțiunii de perfecționare; utilizarea eficientă implică trecerea de la prelucrarea fragmentată pe activități separate, la prelucrarea integrată pe ansamblul activităților, în cadrul căreia se utilizează pentru toate aplicațiile aceiași bază de date și se ține seama de toate influențele reciproce dintre activități, în vederea îmbunătățirii sistemului de conducere.

Pregătirea personalului pentru introducerea SIPAD trebuie să corespundă gradului tehnic al dotării, cit și înțelegerii scopurilor sistemului.

Organizarea centrelor de calcul teritoriale în paralel cu centrele de calcul pentru întreprinderi, grupuri de uzine și departamentale, permite ca formarea cadrelor de specialiști în întreprinderi să se desfășoare în paralel, sau cu un decalaj relativ redus față de procesul de introducere a SPAD în întreprinderi. Dintre cadrele de specialiști necesare, este esențială formarea temeinică a analiștilor de sisteme și programatorilor. Nivelul de formare al analiștilor de sisteme, al căror profil trebuie să acopere cunoașterea întreprinderii, cunoașterea sistemelor de prelucrare automată a datelor și a unor pachete de programe existente pentru aplicații, tehnice de organizare a bazelor de date și ale fișierelor și care trebuie să posede de asemenei experiență în tehnici matematice de modelare și optimizare, este cheia activității de îmbunătățire a sistemelor informaționale, pînă la procesul de perfecționare a conducerii și organizării întreprinderilor, proces de mare însemnatate în etapa actuală de dezvoltare economică a țării noastre.

Introducerea prelucrării mecanice și automatizate se simplifică prin organizarea de centre de calcul la care să se prelucreze datele

pentru unitățile economice sau grupe de întreprinderi ce pot fi deservite în comun. Acest lucru este impus pe de o parte de costul relativ ridicat al acestor echipamente, iar pe de altă parte de lipsa de personal de specialitate și cu experiență în utilizarea acestora.

Existența la nivel central a unor institute care să acioneze ca unități puternice de consultanță și elaborare sau adaptare de pachete de programe și care să efectueze proiectarea părții comune în sistemele informaționale (zona I în fig. 6.2.2.) poate asigura asistența tehnică necesară activității centrelor de calcul teritoriale, uzinale, departamentale etc.

Această organizare permite de asemenei pregătirea mai eficientă (mai la concret) a cadrelor prin introducerea în programele de pregătire a studiului pachetelor și proiectelor cadrului și crează posibilitatea utilizării pe scară largă și a cadrelor cu pregătire mai puțin avansată, datorită cadrului existent pentru munca de proiectare și programare și a asistenței ce se acordă de la nivelele mai centrale.

Crescerea productivității muncii constituie un efect economic important care se poate obține și prin cunoașterea sistematică a necesităților și termenelor de aprovizionare, producție, desfacere și prin utilizarea ratională a personalului tehnic-administrativ. Cu același număr de salariați, întreprinderea poate să rezolve sarcini mai multe, mai grele, lucrările de rutină fiind efectuate de mijloace de mecanizare și automatizare.

Stăpînirea informațiilor din întreprinderi privind activitatea de planificare pe termen lung, mediu sau scurt și posibilitatea refacerii calculelor în cazul schimbării condițiilor initiale, împreună cu cunoașterea necesarului de aprovizionat, practic în orice moment al activității întreprinderii, permite ca din timp să se poată trece la luarea de măsuri în vederea aprovizionării sau modificării în sensul unui răndament optim al procesului de producție. Considerăm că într-o primă etapă, chiar dacă sistemele informaționale nu efectuează „optimizări“, prin cunoașterea și posibilitatea calculării rapide a cîtorva variante sau prin înlocuirea criteriilor de optim cu principii de „mai bine“ se pot obține eficiențe relativ mai mari decît eficiențele suplimentare care se mai pot obține și prin adăugarea și rezolvarea problemelor de optimizări. Evident, acestea nu trebuie neglijate, dar ele trebuie să urmeze în general după ce în întreprinderi s-au introdus sisteme informaționale la un anumit nivel de integrare (evidență și raportare sau urmărire și control), în cadrul căror s-au organizat și sistematizat datele necesare rezolvării problemelor de optimizări și

după ce s-a dezvoltat o muncă sistematică de analiză a obiectivelor și deciziilor la toate nivelele.

Fără sistematizarea materialului informațional din întreprinderi, fără eliminarea erorilor existente în documente, fără organizarea datelor și folosirea unitară a bazelor de date, și fără analiza obiectivelor și deciziilor, problemele de optimizări care se pot rezolva aduc o eficiență minimă care dezamăgește conducerile întreprinderilor. De altfel, mulți ani de zile eforturile unor cercetători din diferite institute științifice din țara noastră s-au consumat în astfel de încercări mai mult sau mai puțin sterile, de a rezolva fără o pregătire corespunzătoare privind organizarea datelor și analiza deciziilor unele probleme de „tip conducere”, fără ca să se poată obține rezultate care să contribuie la luarea de măsuri pentru dezvoltarea acestor acțiuni. După cum s-a arătat o cauză obiectivă a fost lipsa de memorii auxiliare de mare capacitate pentru organizarea bazelor de date sau a fișierelor principale. O altă cauză care a generat poate chiar întîrzierea în dotarea cu aceste dispozitive se datorează însă faptului că pentru sistematizarea materialului informațional și eliminarea erorilor pentru organizarea datelor și a fișierelor, este necesară în afară de o concepție unitară, o perioadă de 1—2 ani, timp în care sunt solicitați circa 10—20 analiști și specialiști din întreprinderi, în această perioadă nefiind posibil să apară rezultate care să încurajeze conducerea întreprinderii să-și continue eforturile în sensul implementării noului sistem. Experiența acumulată privind colaborarea unui centru de calcul cu un număr de întreprinderi pilot pentru depășirea cu succes a acestei perioade este pozitivă. Rezultatele care s-au obținut pînă în prezent în unele întreprinderi după implementarea sistemelor informaționale bazate pe folosirea sistemului de prelucrare automată a datelor vor permite în scurt timp să se aprecieze eficiența acestor acțiuni.

Pe această linie, în țara noastră s-a și obținut o experiență valoasă. Astfel, de pildă (fiind exemplul cel mai cunoscut autorilor), la CEPECA, în centrul de calcul dotat cu un sistem de prelucrare automată de capacitate medie, s-a asigurat pregătirea de cadre de specialiști în acest domeniu și s-au efectuat aplicații pentru un grup de întreprinderi pilot („23 August”, „F.M.U.A.B.”, „I.T. Dacia”, „Policolor”). Tot aici s-au pregătit cadre pentru centrele teritoriale și pentru alte centre și instituții, a căror activitate se va desfășura conform experienței acumulate, tot în colaborare cu un număr de întreprinderi pilot, impletind activitatea de implementare a noilor

sisteme informaționale din aceste întreprinderi cu procesul de formare a cadrelor de analiști și programatori.

Pe măsura dezvoltării activității și în alte institute sau centre existente care proiecteză sau pregătesc cadre sau alte centre care sunt în curs să-și completeze dotarea cu echipamente, se vor crea condiții de dotare cu echipamente pentru noi întreprinderi.

Problemele care au stat în fața colectivului unității de prelucrare automată a datelor CEPECA la inceputul activității, cu circa 4—5 ani în urmă, credem că sunt încă actuale pentru toate centrele de calcul și pentru a concretiza cîteva aspecte discutate anterior ne vom referi în continuare la unele exemple din activitatea acestei unități. Lipsa de cadre și de experiență în domeniul introducerii sistemelor de prelucrare automată a datelor în întreprinderi a trebuit corectată practic din mers, în activitatea desfășurată de această unitate urmărindu-se realizarea în paralel sau cu decalaje relativ mici, a următoarelor obiective principale :

- formarea cadrelor proprii de specialiști, precum și constituirea în unele întreprinderi pilot de echipe formate din analiști și programatori ;

- elaborarea unor programe de formare pentru analiști de sisteme și programatori și formarea prin aceste cursuri a numărului planificat de analiști și programatori, în special în vederea constituirii în prima etapă de nuclee de specialiști în întreprinderile și centrele de calcul care sunt prevăzute în mod priorită să fie dotate cu echipamente de prelucrare ;

- formarea unei concepții de proiectare și execuție în acest domeniu și rezolvarea problemelor de bază pentru proiectarea de sisteme informaționale în întreprinderile pilot menționate ; pentru activitățile urmărite și gestiunea stocurilor, lansarea, programarea operativă și controlul producției, calcule finanțări-contabile, aprovizionarea și desfacerea și pentru asigurarea treptată a integrării acestor activități ;

- proiectarea acestor subsisteme de aplicații, în prima etapă pe baza unor modele adaptate sistemelor de prelucrare automată a datelor, care să permită să se compare programele originale, elaborate de specialiștii noștri cu pachetele de programe elaborate de diferite firme și care să permită de asemenei compararea sistemelor proiectate pe baza concepției de integrare în perspectivă, cu concepția realizării de aplicații fragmentate,

- proiectarea fișierelor și elaborarea procedurilor pentru organizarea bazelor de date, folosind pachetele furnizate de firme.

Experiența acumulată pînă în prezent, deși încă redusă, demonstrează avantajele concepției de integrare și faptul că metodele care permit să se stăpinească informațiile din întreprindere pentru problemele de bază menționate, pun la îndemna conducerii întreprinderii elementele esențiale în vederea luării deciziei, eficiența acestor sisteme „de evidență sau de informare“ mai mult decît „de optimizare“ reprezentînd în general, partea majoră din eficiență posibilă.

Eliminarea erorilor existente (pînă la 30% din documentele și fișierele din actualele sisteme informaționale sunt afectate de erori), posibilitatea intervenției operative în cadrul sistemului informațional în cazul cînd apar perturbații față de situația prevăzută, cunoașterea în fiecare moment a situației existente și prezentarea de variante posibile între care să se poată alege, sint printre aspectele cele mai importante care fiind asigurate conduc la realizarea eficienței menționate, în special pe seama utilizării mai raționale a capacitaților existente, scurtarea ciclului de fabricație și reducerea cheltuielilor pentru stocuri.

Asigurarea compatibilității la nivelul bazelor de date între sistemele informaționale (aceleași fișiere de bază și variabile care să constituie partea minimală comună tuturor sistemelor informaționale dintr-o anumită zonă, de pildă întreprinderile și centralele dintr-o ramură și anumite tranzacții necesare unor prelucrări intersisteme care să aibă chei comune) ne permite să înțelegem ce înseamnă constituirea în perspectivă de sisteme informaționale departamentale, teritoriale, la nivel național.

De asemenei, determinarea structurii și conținutului bazelor de date compatibile, a dicționarului de informații ori aceste baze de date, constituie după cum am văzut o pirghie importantă pentru întreaga acțiune de raționalizare la scară națională a documentelor de evidență (întrări și ieșiri din baza de date) și analiza indicatorilor, creând condiții favorabile pentru a se putea asigura compatibilitatea între sistemele manuale, mecanizate și automate în vederea trecerii în perspectivă de la un sistem organizat manual sau cu echipamente mecanografice la un sistem mecanizat sau respectiv automat, fără a mai fi necesare acțiuni costisitoare de reorganizare a datelor.

Compatibilitatea la nivelul bazelor de date asigurată prin utilizarea același pachet de generare și organizare a bazelor de date independent de tipul de calculator crează de asemenea condițiile necesare pentru alcătuirea de rețele de calculatoare de diferite tipuri și capacitați, în perspectivă.

Un alt aspect care își aduce o contribuție majoră în obținerea eficienței prin utilizarea SPAD în întreprinderi îl constituie *factorul educativ* pe care utilizarea prelucrării automate îl favorizează. În primul rînd este vorba de disciplina și exactitatea de care trebuie să dea dovadă salariații care utilizează SPAD. Prin cunoașterea precisă a sarcinilor și modul ritmic în care informațiile trebuie furnizate, respectiv rapoartele să fie folosite, se crează un sistem nou de lucru pe care-l putem denumi „*în bandă informațională*“. În al doilea rînd, ca o consecință a acestui mod de lucru, se crează atât pentru cadrele de conducere, cit și de specialitate (ingenieri, economisti etc.), condiții pentru afectarea de timp creșterii pregătirii proprii și a cadrelor din jur și mărirea ponderii preocupărilor pentru imbunătățiri tehnologice sau de organizare și dezvoltare în perspectivă.

Ca urmare a exploatarii sistemelor informaționale, folosind prelucrarea automată a datelor în întreprinderi, în diferite țări, rezultă că întreprinderile lucrează cu stocuri mai mici decît înainte, crescîndu-și producția și productivitatea muncii, eficacitatea comercială și eficiența birourilor de studii și cercetări. Modificările de structură din întreprindere se manifestă, după cum am văzut, în principal prin reducerea personalului compartimentelor de planificare, programare și lansare, controlul producției, evidența magaziilor și contabilitate și înființarea sau mărirea altor compartimente ca: serviciul de informatică, serviciul de personal, de metode, de organizare etc.

În activitatea de implementare a SIPAD pe care au început sau încep să o desfășoare centrele de calcul, este necesar ca în mod hotărît să se evite prelucrările pe calculator pentru aplicații fragmentate (cu caracter mecanografic) și să se treacă la proiectarea de sisteme informaționale integrate pentru deservirea conducerii și introducerea în perspectivă de la nivelul bazelor de date, indiferent de tipul de calculator utilizat, proiectarea la nivele centrale a părții comune tuturor sistemelor informaționale din diferite zone (fig. 6.2.2.), legată de administrarea bazei de date și organizarea la scară națională a asistenței în munca de proiectare și implementare de la nivelele mai centrale și formarea de cadre, sint probleme cheie ale acestei etape. De rezolvarea lor priorităță depinde asigurarea dezvoltării într-un ritm susținut, în acest domeniu de mare importanță pentru economia națională.

Anexo I.I

Anexo J.3

CITRUS

Annals 14

100

Annex 1.5

Anexo 1.6

CEPECA	Numele întreprinderii RPZ/N	FORMULAR PENTRU IMPRIMANTĂ	Anexă 16	Pop																																																																																																																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2"></th> <th colspan="10">Aplicarea Programului</th> <th colspan="2"></th> </tr> <tr> <th colspan="2"></th> <th colspan="10">Proiectator</th> <th colspan="2"></th> </tr> <tr> <th colspan="2"></th> <th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">CINCUETELE DE VINDINTE PE ARHICOU INTRAVIL</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td colspan="2" style="text-align: center;">XX/XXXX/XXXX - XX/XXXX/XXXX</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td style="text-align: center;">Nr.</td> <td style="text-align: center;">Cod sort.</td> </tr> <tr> <td style="text-align: center;">Cin</td> </tr> <tr> <td colspan="16" style="text-align: right; padding-right: 10px;">TOTAL: _____</td> </tr> </tbody> </table>							Aplicarea Programului														Proiectator														0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	CINCUETELE DE VINDINTE PE ARHICOU INTRAVIL																	XX/XXXX/XXXX - XX/XXXX/XXXX																	Nr.	Cod sort.	Cin	TOTAL: _____																																												
		Aplicarea Programului																																																																																																																																	
		Proiectator																																																																																																																																	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14																																																																																																																			
CINCUETELE DE VINDINTE PE ARHICOU INTRAVIL																																																																																																																																			
XX/XXXX/XXXX - XX/XXXX/XXXX																																																																																																																																			
Nr.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.	Cod sort.																																																																																																																				
Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin	Cin																																																																																																																				
TOTAL: _____																																																																																																																																			

Anexo 2.I

TUNISIAAT DE DUMBAK Nr. 1

CARTELL DATE

FEU (10)

卷之三

卷之三

CARTELL DATE

FEU (10)

Anexa 2,2

MATERIAL

FDI(2)]

<i>Program</i>	<i>Date</i>
<i>Semnul</i>	
<i>Se perferează</i>	

Programmabor

Programmator

MATERIAL

FLUID

123

Anexo 2.3

CEPESI

CARTER'S DATE

Geburt

Anexo 2.4

CEPTEA

300

Anexa 2.5.I

CONTINUATE

FCU/(51,11)

Anexa 2.5.2

CEPORA

CREATE DATE

Sp performance
Data

ANETA 3.J

CEPECA

FISA DE ANALIZA A INFORMATIILOR

638

caractere		informații elementare		detaliu	note - suport - nume simbolice	caractere		informații elementare		detaliu
oc	N	detaliu	destinație	SURSE - destinații	oc	N	detaliu	destinație	destinație	destinație
*		Filler	01 01							
*		Cod mașină	02 04							
*		Numele mașinii	03 34							
*		Cod atelier sau tipul mașinii	30 37							
*		Dacă e masină sau atelier	38 38							
		M = Mașină								
		A = Atelier								
*		Filler	39 50							

ANEXA 3.2

FISA DE ANALIZA A INFORMATIILOR

ECC7

CEPECA		FISA DE ANALIZA A INFORMATIILOR					
etapă	cod	număr	Denumirea informației	suport	masina	Compartiment	prelucrare
			FCCZ::=Fisier cod cauze întrerupere utilizare				

ANEXA 3.3

FISĂ DE ANALIZĂ A INFORMAȚIILOR

-ESI

ANEXA 3.4

CEPECA

FISA DE ANALIZĂ A INFORMATIILOR

ANEXA 3.5

CEPECA		FIŞA DE ANALIZA A INFORMATIILOR					
etajă	col.	nr.	Denumirea informației	suport	masina	Compartiment	preluare
			<i>FLPO = Fișier culori parțiale</i>				

INTRARE SAU IESIRE

DBS - ORGANIZARE - ACCES

support masina Compartiment preluare

FIPD = Fisier cu lofuri parțiale

caractere	informatii elementare	pozitii		note - suport - nume simbolice surse - destinatii	caracteri	informatii elementare	pozitii	
		oc	N				detaliu	detaliu
	<i>Descrierea clasei de recorduri</i>							
	<i>Nr. 1</i>							
*	<i>Filler</i>	01	01		*	<i>Filler</i>		01 01
*	<i>Codul lotului din care provine seful de loturi parțiale</i>	02	06		*	<i>Cod lot proveniență</i>		02 06
*	<i>Secvența = '00'</i>	07	08		*	<i>Secvența lotului parțial</i>		07 08
*	<i>Codul ultimei masini pe care a trecut lotul înainte de vizuire</i>	09	11		*	<i>Codul noii poziții calendaristice la relansare</i>		09 14
*	<i>Cod articol</i>	12	15		*	<i>Cod lot calculator la relansare</i>		15 19
*	<i>Codul ultimei pozitii calendaristice</i>	16	21		*	<i>Data relansării sub forma an, luna și zi</i>		
*	<i>Data intrării în stocurile rezerve sub forma an, luna, zi, ora și minut</i>	22	31		*	<i>Lungimea lotului parțial</i>		26 29
*	<i>Codul lotului de calculator la intrarea în stoc parțial</i>	32	36		*	<i>Cimp de stare</i>		
*	<i>Secvența ultimului lot parțial din grup</i>	37	38		*	<i>'L' = lot relansat</i>		
*	<i>Filler</i>	39	40		*	<i>Suf 'N' = lot parțial în stoc</i>		
	<i>Cimpurile 09-38 contin datele ultimului record din clasa Nr. 2 care a fost înregistrat</i>				*	<i>Filler</i>		

ANEXA 3.6

CEPECA		FIŞA DE ANALIZA A INFORMATIILOR					
etapă	cod	număr	Denumirea informației	suporți	masina	Compartiment	prelucrare
			FILD ::= Fișier pentru loturi normale				

FILD ::= Fisier pentru loturi normale

Denumirea informației

Support | masking | Com

ment pre-crack

GEPECA

FISĂ DE ANALIZĂ A INFORMATIILOR

CEPE

FISA DE ANALIZA A INFORMATIILOR

FISA DE ANALIZĂ A INFORMATIILOR

etajă	cod	număr	Denumirea informației	suport	masina	Compartiment	prelucrare
			<i>Fisier corespondente FCR</i>				
INTRARE SAU IEȘIRE			OBS - ORGANIZARE - ACCES			FISIER INTERN	
PERIODICITATE				<i>Fisier indexat secvențial</i>	<i>Metoda de acces RANDOM</i>	număr	medie
număr	medie	min	max			elem	min
documente sau mesaje						car/elem	max
caracteristică						car/bloc	
informație de bază						ocuparea	
informații totale						EVP-UTIE	PERIODICITATE
număr de linii						adăugare	înțizieri
număr de copii și destinație						modul	%
						prelucrare	per elem modul %

FISĂ DE ANALIZĂ A INFORMATIILOR

CEPECA			FIŞA DE ANALIZA A INFORMAȚIILOR					
etapă	cod	număr	Denumirea informației		suport	masina	Compartiment	prelucrare
			<i>FCLPR</i>		Banda	IBM 369	I.T. DACIA	Automată
			<i>Fișier pentru loturi și parții replanificate</i>		Cartela			
INTRARE SAU IEȘIRE			OBS - ORGANIZARE - ACCES			FIŞIER INTERN		
PERIODICITATE			<i>Fișierul este organizat pe cartele sau pe banda magnetică cu record de 80 tranzac-</i>			număr	medie	min.
număr	medie	min.	max	<i>tere. Pe o cartelă poate fi 2 tranzacții de</i>			elem	max
DOCUMENTE SAU MESAJE				<i>același tip, prima în poziția 13-43 iar a</i>			car/ elem	
caracter				<i>două în poziția 44-78.</i>			car/bloc	
informații de bază				<i>Tranzacția 2 poate lipsi din cartela. În</i>			ocuparea	
informații totale				<i>acest caz în locul tranzacției 2 apare</i>			EVOLUTIE	PERIODICITATE
număr de linii				<i>spațiu.</i>			zăbojeni întârzieri	modul %
număr de copii și destinație							prelucrare per elem	modul %

caractere ac N	informații elementare	pozitii etichete	note - suporți - nume simbolice surse - destinații	caractere ac N	informații elementare	pozitii etichete	
	Cimpuri comune pentru cele 2 tranzacții				1 Idem cimp 35-35 2 Idem cimp 36-37 6 Idem cimp 38-43	76 73 77 72 73 78	
2	Cod cartela	61 62	Cocart	Filler	2	Filler	79 B
6	Data curentă	63 68	Dafort				
	Cimpuri pentru prima tranzacție						
5	Cod lot fabrica	69 13	Cod lot F				
2	Sevența lot parțial	14 15	SCN lot P				
5	Cod lot calculator	16 20	Codiclc				
4	Cod articol	21 24	Articol				
	Cod poziție coloristică						
5	Reală	25 30	PCOLR				
4	Lungime lot	31 34	Lungime				
1	Cod proces tehnologic	35 35	Proces				
	Numărul mașinii precedente celor pe care						
2	Se face încarcarea	36 37	MASPREG				
	Data ieșirii de pe masina pre- cedentă. Se ia data reincarcării lotului finindu-se seama de timpul de transport de la ma- șina precedentă pînă la						
6	Masina pe care se încarcă	38 43	Datainc				
5	Idem cimp 9-13	44 48					
2	Idem cimp 14-15	49 50					
5	Idem cimp 16-20	51 55					
4	Idem cimp 21-24	56 59					
5	Idem cimp 25-30	60 65					
4	Idem cimp 31-34	66 69					

Bun de tipar : 10.11.1972. Colt de tipar : 17.50.
Planșă : 1. Tiraj : 7500+140 ex. Brogat.
C.Z. 65.011.56.



Intreprinderea Poligrafică „Banat”
Timișoara, Calea Aradului, nr. 1/A,
Repubica Socialistă România.

Se studiază etapele de realizare a sistemelor integrate de prelucrare automată a datelor (SIPAD), în care subsistemele de funcții (prospectarea pieții, planificarea producției, aprovizionarea, controlul stocurilor etc.) sunt incadrate treptat într-un sistem integrat de conducere cu mijloace de automatizare.

Se adresează în principal, analiștilor de sisteme și conducătorilor de întreprinderi, cum și specialiștilor — tehnicieni și economiști — profesorilor, studenților în domeniul prelucrărilor datelor, al tehnicii de calcul și al organizării producției.

Lucrarea face parte din seria -INITIERE- (Automatică, Informatică, Electrică, Management).

L e l 13,50

