

a „w boczku” podaje się rodzaje informacji wejściowej. Na przecięciu wierszy i kolumn zaznacza się znakiem X występowanie danej operacji dla danej informacji.

Podobnie można postąpić w celu określenia warunkowych sytuacji. Schematy tego typu nazywają się tablicami decyzyjnymi. Poziomo podaje się kolejno nazwy lub numery warunków, a z boku nazwy sytuacji lub informacji. Na przecięciu wierszy i kolumn w zależności od sytuacji wstawia się T (tak) lub N (nie).

Projektowanie koncepcji systemu obejmuje m.in. podanie celu i zasad eksploatacji, struktury ogniw przetwarzania, źródeł danych i dystrybucji wyników, objętości strumienia przetwarzanych danych, priorytetów i spodziewanych czasów przetwarzania, procedury w wypadku błędów, spodziewanych efektów itp.

Projektowanie techniczne elementów (ogniw przetwarzania) systemu polega m.in. na projektowaniu planu operacyjnego przetwarzania, wyłonieniu przebiegów przetwarzania, projektowaniu schematów blokowych programów, programowaniu, projektowaniu cyklicznego przetwarzania i terminów ważności przechowywanych danych.

## *6.8. Projektowanie technologii procesu automatycznego przetwarzania danych*

W planie operacyjnym przetwarzania wyróżnia się makrooperacje:

— urządzeń peryferyjnych, na których dokonuje się wstępnej obróbki danych,

— komputera, podzielone na przebiegi.

Projektowanie przetwarzania na urządzeniach peryferyjnych sprowadza się do ułożenia instrukcji:

— kontroli formalnej,

— dziurkowania maszynowych nośników informacji,

— kontroli weryfikacyjnej.

Projektowanie planu operacyjnego przetwarzania omówimy bardziej szczegółowo w zakresie projektowania planu operacyjnego, dokonywania podziału na przebiegi przetwarzania, sporządzania schematów blokowych.

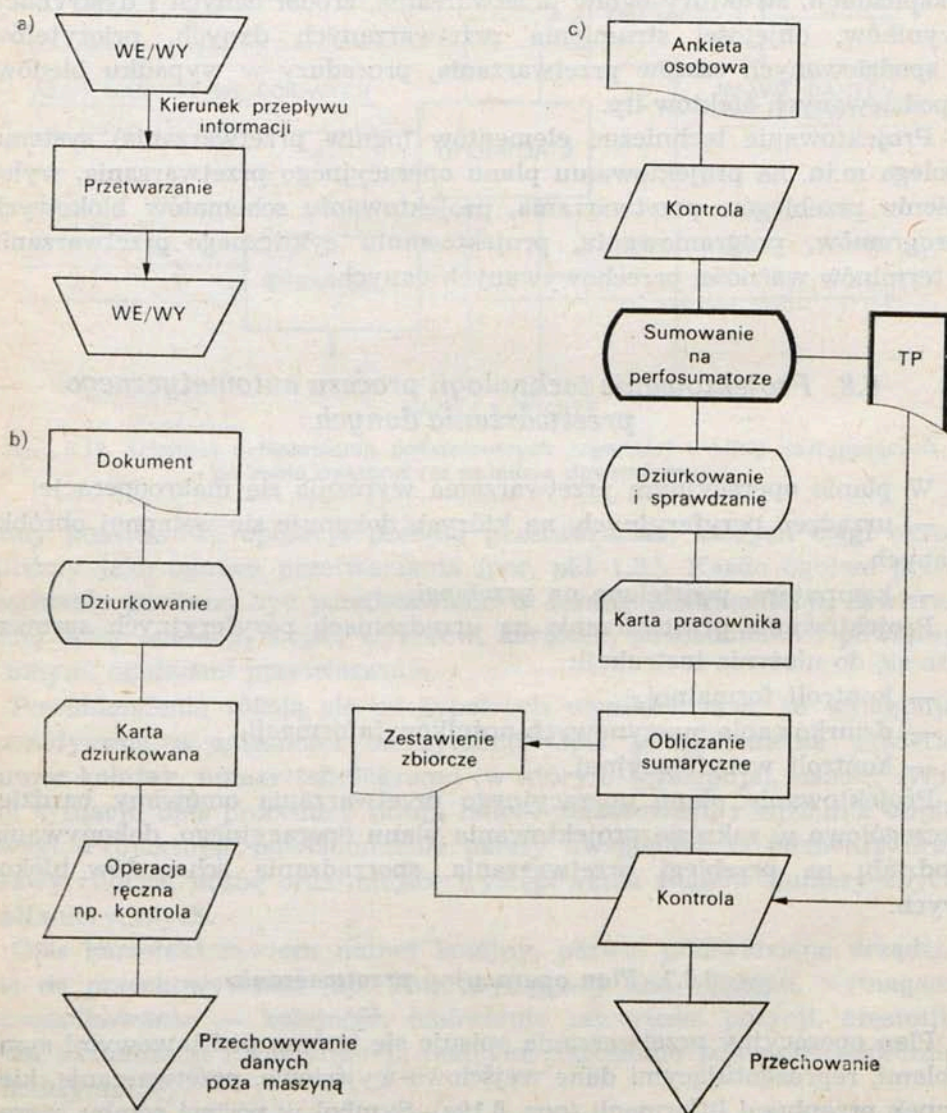
### *6.8.1. Plan operacyjny przetwarzania*

Plan operacyjny przetwarzania opisuje się trzema podstawowymi symbolami, reprezentującymi dane wejściowo-wyjściowe, przetwarzanie, kierunek przepływu informacji (por. 6.19a). Symbol w postaci rombu reprezentuje każdy typ urządzenia lub danych wejściowo-wyjściowych. Symbol przetwarzania przedstawia każdą funkcję przetwarzania (program,



przebieg, komputer lub inny sprzęt, jak np. kalkulator). Symbol kierunku przepływu informacji występuje zarówno, gdy odbywa się on od góry do dołu i od lewej do prawej. W sytuacjach przeciwnych stawia się strzałki oznaczające kierunek.

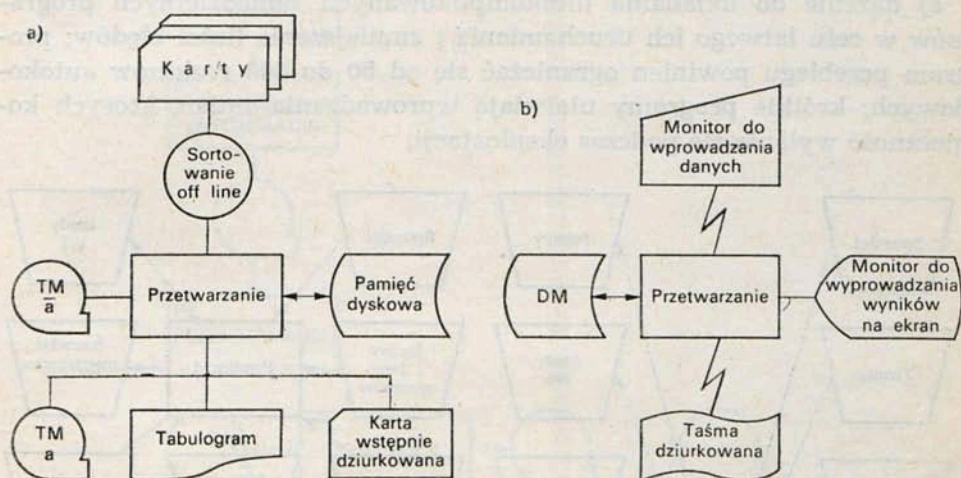
Aby schemat był bardziej czytelny, można wprowadzić wiele innych oznaczeń, np. dla oznaczenia dokumentu, operacji dziurkowania, operacji ręcznej kontroli, przechowywania danych w tradycyjny sposób (por. ryc. 6.19b), perforowania taśmy dziurkowanej na maszynach małej i średniej mechanizacji (por. ryc. 6.19c), operacji pomocniczych, operacji



Ryc. 6.19. Schemat planu operacyjnego przetwarzania danych: a) podstawowe operacje, b) dziurkowanie, operacje ręczne, przechowywanie danych poza maszyną wraz z oznaczeniem dokumentu, c) dziurkowanie taśmy na perfosumatorze

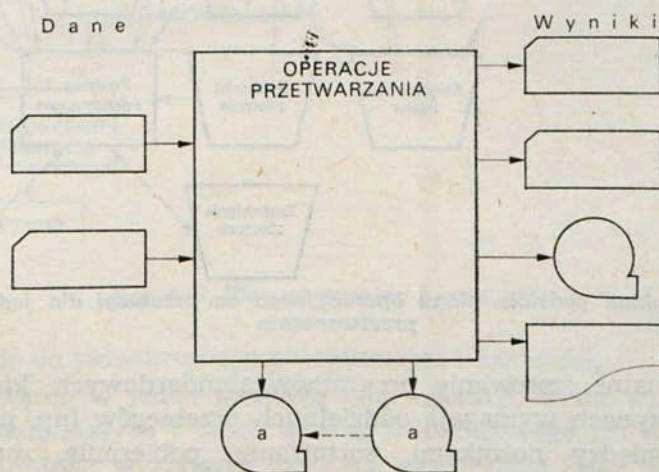


sortowania i dobierania rozłącznego (poza maszyną). Ponadto używa się oznaczeń pamięci wyrwykowej (dyski magnetyczne), pamięci na taśmach magnetycznych (por. ryc. 6.20a). Oznaczenie monitora wprowadzającego dane, monitora ekranowego do wyprowadzania wyników oraz łącza transmisji danych przedstawiamy na ryc. 6.20b.



Ryc. 6.20. Schemat przykładowego planu operacyjnego przetwarzania danych: a) z zastosowaniem dysku i taśmy magnetycznej oraz sortera, b) z zastosowaniem monitora do wprowadzania danych na odległość, monitora do wprowadzania wyników na ekranie oraz perforatora do wprowadzania wyników na taśmę dziurkowaną przez sieć transmisji danych

W planie operacyjnym przetwarzania dąży się przede wszystkim do wyłonienia przebiegów, których założenia są podstawą opracowania programów (por. ryc. 6.21).



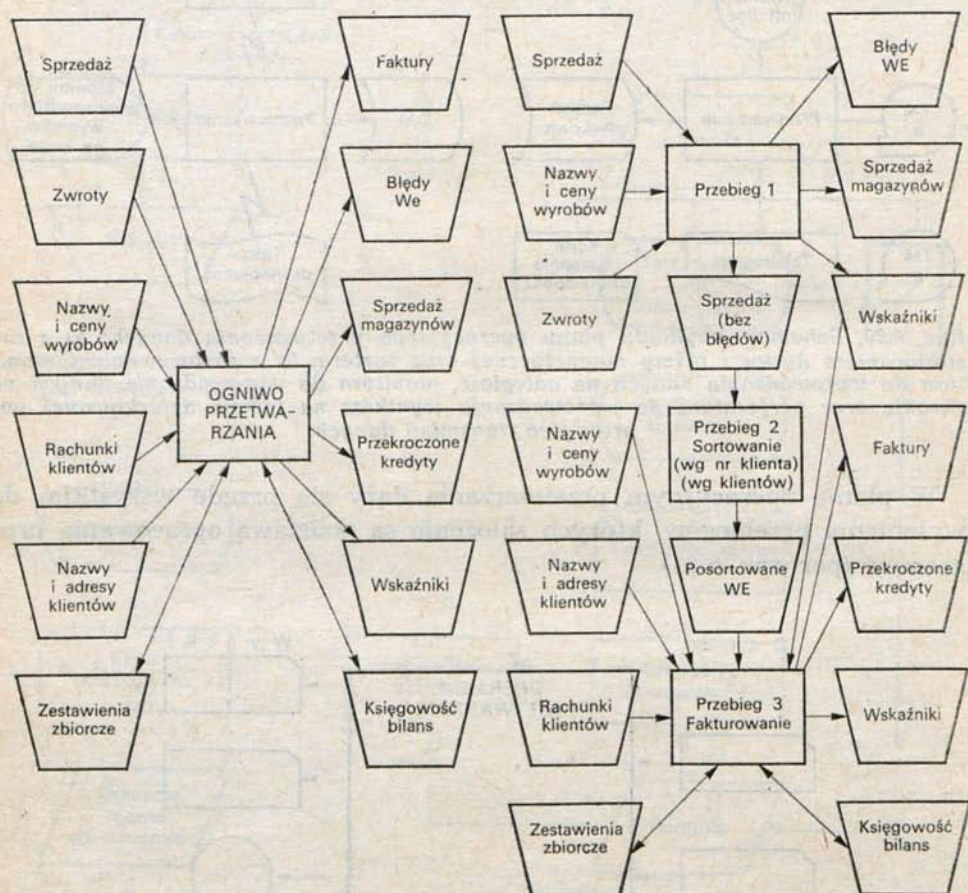
Ryc. 6.21. Schemat opisu przebiegu przetwarzaniowego



### 6.8.2. Podział planu operacyjnego na przebiegi

Podział planu operacyjnego na przebiegi przetwarzania jest konwencjonalny i w dużym stopniu zależy od wyczucia projektanta. Przy rozważaniu podziału na przebiegi powinno się uwzględniać następujące punkty:

a) dążenie do układania nieskomplikowanych samodzielnych programów w celu łatwego ich uruchamiania i zmniejszenia ilości błędów; program przebiegu powinien ograniczać się od 50 do 300 rozkazów autokodowych; krótkie programy ułatwiają wprowadzanie zmian, których konieczność wyłania się podczas eksploatacji,



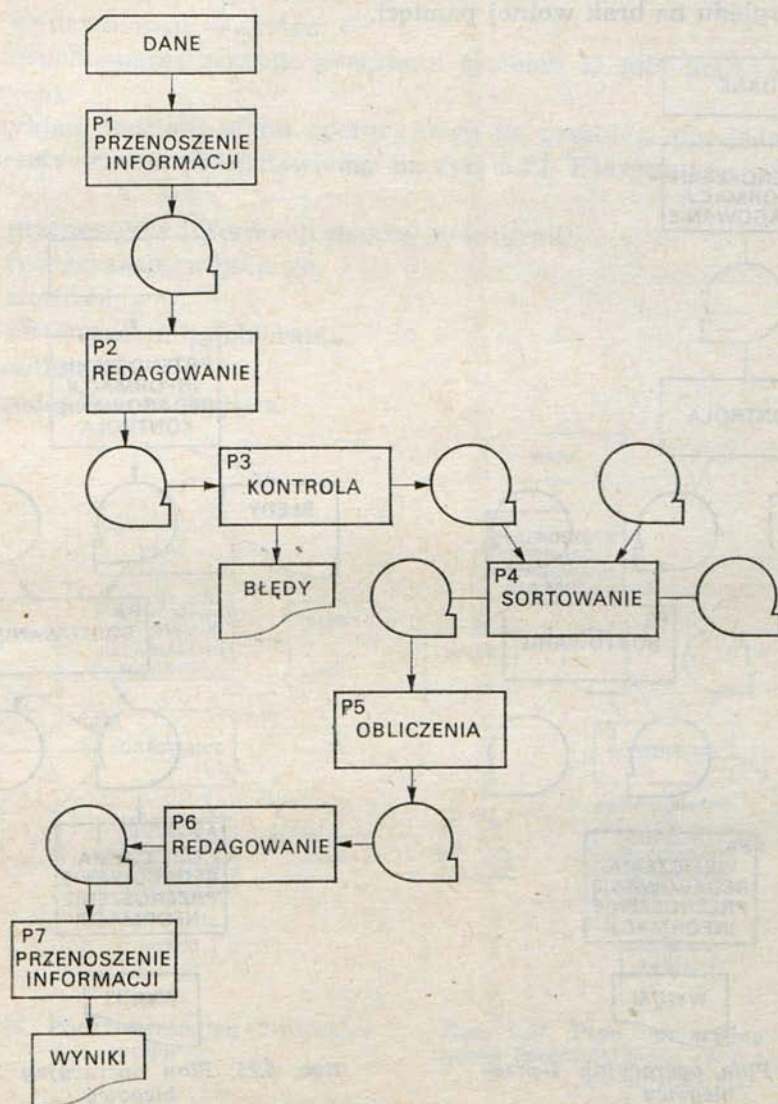
Ryc. 6.22. Przykład podziału planu operacyjnego na przebiegi dla jednego ogniwa przetwarzania

b) maksymalne stosowanie programów standardowych, które w niektórych maszynach wymagają oddzielnych przebiegów (np. przenoszenie informacji między nośnikami, sortowanie, pobieranie, modyfikacja); w niektórych systemach sortowanie i dobieranie nie wymaga oddziel-



nego przebiegu dzięki możliwości stosowania w programach autokodowych, np. w języku COBOL rozkazów generatorowych wywołujących podprogram,

c) naturalny podział, wyłaniający się w związku z organizacją systemu,



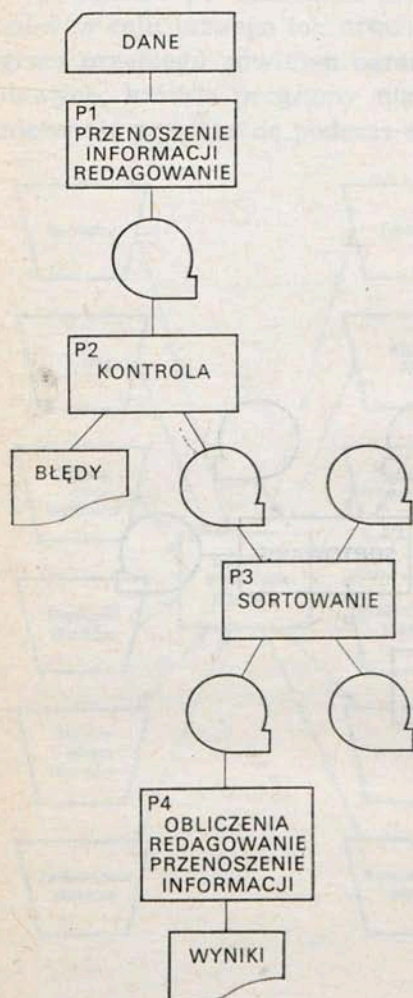
Ryc. 6.23. Plan operacyjny 7-przebiegowy

d) dążenie do jednokrotnego przepuszczania kartoteki,

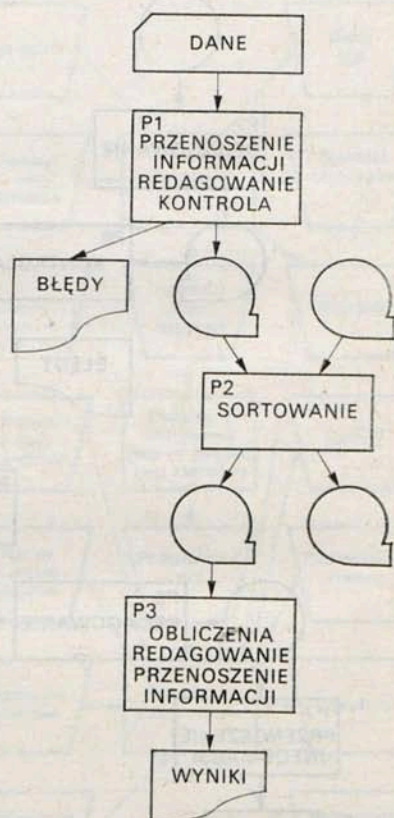
e) grupowanie w jeden przebieg tych operacji, które wymagają tego samego układu stałych i tabel, w celu jednorazowego ich wprowadzania bądź generowania; wykonania możliwie największej ilości operacji w jednym przebiegu,



f) wykorzystanie pojemności dysponowanej pamięci operacyjnej, w której w jednym czasie muszą być umieszczone: program, blok z taśmy magnetycznej, strefy urządzeń zewnętrznych, miejsca robocze, miejsca stale zarezerwowane itd.; zbyt duże skomasowanie kartotek w jednym przebiegu może bardzo poważnie skomplikować adresowanie w programie ze względu na brak wolnej pamięci,



Ryc. 6.24. Plan operacyjny 4-przebiegowy



Ryc. 6.25. Plan operacyjny 3-przebiegowy

g) cykliczność systemu prowadząca do typizowania przebiegów (np. dzienne, dekadowe, miesięczne); należy unikać układania programów wspólnych dla różnych cykli; program taki jest zwykle bardzo skomplikowany logicznie,

h) cechy maszyny; dla maszyn wieloprogramowych można projektować równoczesną realizację określonej liczby przebiegów i w maszynach

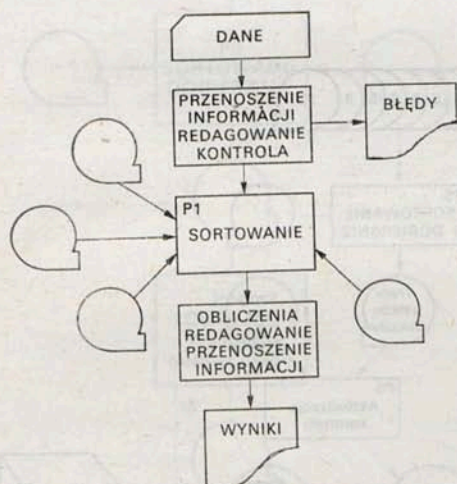


tego typu dążenie do uproszczenia przebiegów jest szczególnie opłacalne, równocześnie np. mogą być prowadzone pod warunkiem zapewnienia odpowiedniej struktury urządzeń zewnętrznych, następujące przebiegi:

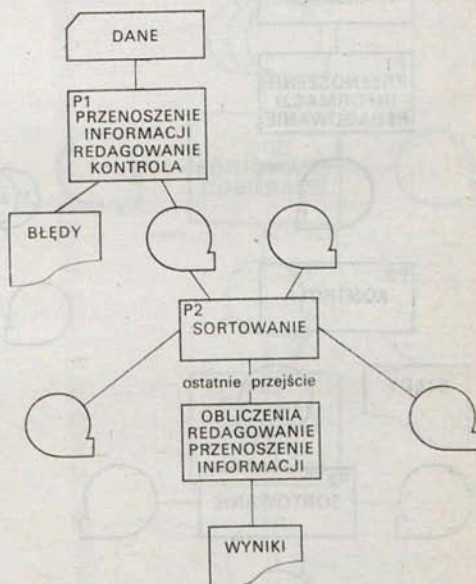
- przebieg przenoszenia informacji między nośnikami — system A,
- aktualizacja — system B,
- wydawnictwo — system C,
- uruchamianie nowego programu systemu D (dla maszyn wielodostępnych).

Przykład podziału planu operacyjnego na przebiegi dla jednego ogniw przetwarzania przedstawiamy na ryc. 6.22. Klasycznymi przebiegami są:

- przenoszenie informacji między nośnikami,
- redagowanie wejściowe,
- kontrola,
- sortowanie lub dobieranie,
- obliczanie,
- redagowanie wyjścia.



Ryc. 6.26. Plan operacyjny 2-przebiegowy



Ryc. 6.27. Plan operacyjny 1-przebiegowy (przetwarzanie w 1 i 2 przebiegu sortowania)

W zależności od sytuacji występować może odpowiedni podział planu operacyjnego na jeden lub więcej przebiegów. Można rozróżnić organizację przetwarzania 7-przebiegową (por. ryc. 6.23), 4-przebiegową (por. ryc. 6.24), 3-przebiegową (por. ryc. 6.25), 2-przebiegową (por. ryc. 6.26) i 1-przebiegową (por. ryc. 6.27).

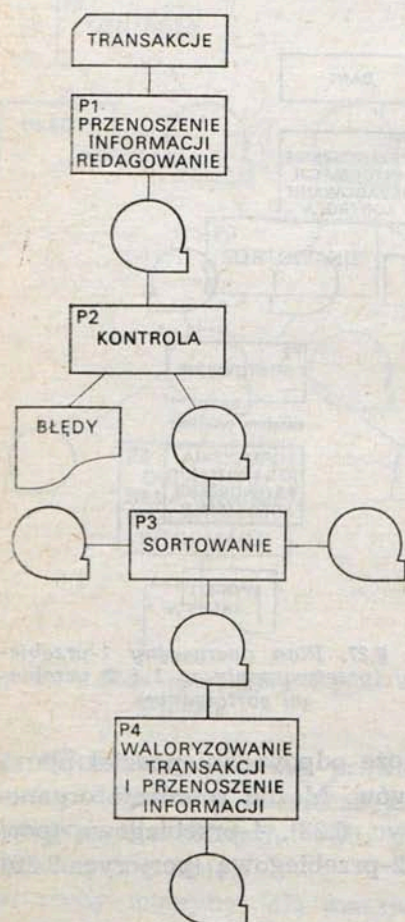
Jak wydaje się, organizacja 4-przebiegowa jest najbardziej optymalna. Przytoczone przykłady mają charakter ideowy. W konkretnych zastosowa-



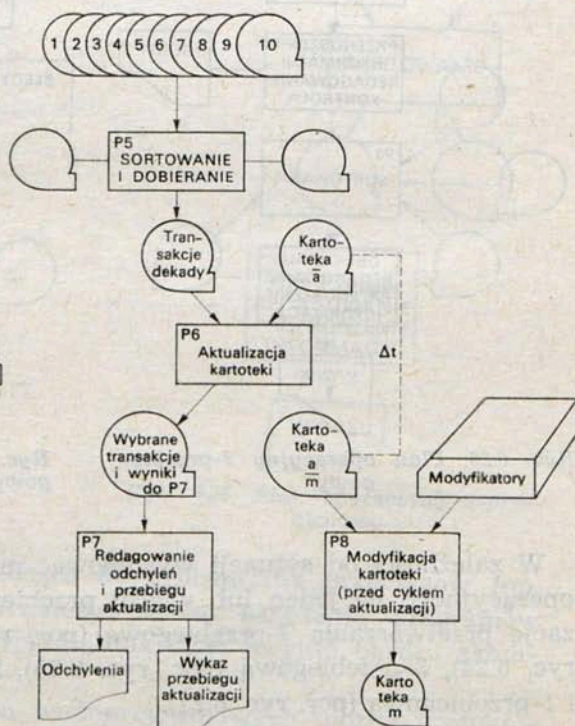
waniach występować mogą odchylenia, w szczególności w odniesieniu do struktury zestawu, np. w zależności od liczby jednostek pamięci zewnętrznej, wymaganej liczby kopii tabulogramów.

### 6.8.3. Projektowanie przebiegów w przetwarzaniu cyklicznym

Na przykład można wymienić zagadnienie fakturowania opłat telefonicznych, za gaz i elektryczność, radio itp. Są to systemy o największej liczbie danych, bowiem dochody w skali jednego regionu — kilkaset tysięcy abonentów, a każdy z nich może mieć kilka lub kilkanaście transakcji (np. rozmowy międzymiastowe). Jeśli założono wystawianie rachunków np. w każdym pierwszym dniu miesiąca — to nastąpiłoby tak wielkie spiętrzenie danych, że nie można by było ich przetworzyć



Ryc. 6.28. Dzienny cykl przetwarzania



Ryc. 6.29. Dekadowy cykl przetwarzania

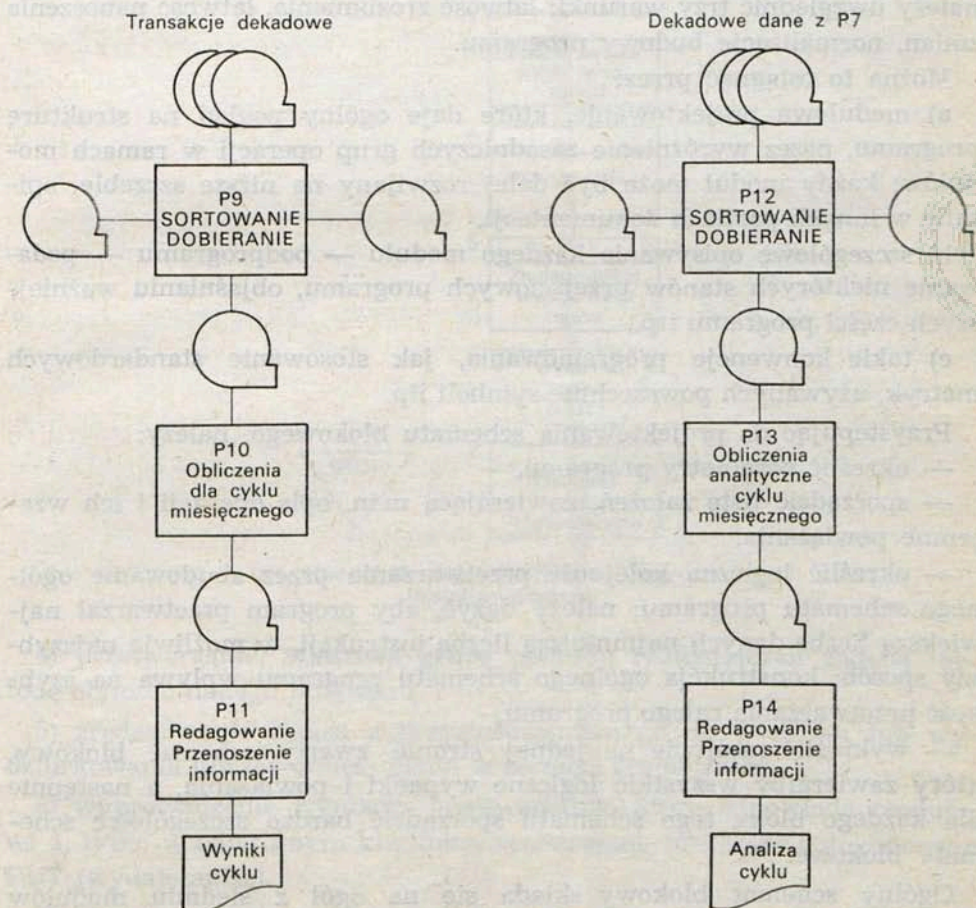


w ciągu jednego dnia. Podobne sytuacje występują w systemach płacowych pracowników fizycznych akordowych, w systemach gospodarki materiałowej itp.

Podstawowym warunkiem automatyzacji ewidencji jest opanowanie techniki stopniowego przetwarzania danych przez stosowanie cykli przetwarzania. Przetwarzanie cykliczne prowadzi do:

- redukcji stosowanych maszynowych nośników informacji (mniej taśm magnetycznych),
- zapewnienia terminowych opracowań,
- utrzymania lepszej kontroli,
- równomiernego obciążenia maszyny.

Na ryc. 6.28 przedstawiamy system trójcyklicznego przetwarzania dziennego, na ryc. 6.29 dekadowego, a na ryc. 6.30 — miesięcznego w celu aktualizacji kartoteki i uzyskania odpowiednich zestawień wynikowych.



Ryc. 6.30. Miesięczny cykl przetwarzania



Dla każdego programu realizującego przebieg przetwarzania (por. ryc. 6.27 przy podstawieniu nazw danych wejściowych i wyjściowych kartotek — maszynowych nośników informacji według symboli z ryc. 4.1) projektuje się schemat blokowy. Symbole powinny być kreślone ołówkiem w celu łatwiejszego nanoszenia zmian. Projektowanie schematu blokowego powinno zaczynać się od najbardziej syntetycznego, prostego. Zaleca się projektowanie na oddzielnych stronach, łączonych symbolem łącznika z numerem. Każda strona schematu blokowego powinna dotyczyć zamkniętego kompleksu operacji. Wyróżnia się dwa łączniki: jeden dla powiązania operacji, drugi — wyszczególnienia stron. Każdy schemat blokowy powinien zaczynać się od opisu, zawierającego przedmiot i objaśnienie podstawowej koncepcji, jeśli jest ona trudna do odczytania ze schematu.

Schemat blokowy odzwierciedla strukturę programu. Projektując go, należy uwzględnić trzy warunki: łatwość zrozumienia, łatwość nanoszenia zmian, normalizację budowy programu.

Można to osiągnąć przez:

a) modułowe projektowanie, które daje ogólny pogląd na strukturę programu, przez wyróżnienie zasadniczych grup operacji w ramach modułów; każdy moduł może być dalej rozwijany na niższe szczeble, opisane w innych partiach dokumentacji,

b) szczegółowe opisywanie każdego modułu — podprogramu — podawanie niektórych stanów przejściowych programu, objaśnianiu ważniejszych części programu itp.,

c) takie konwencje programowania, jak stosowanie standardowych metryk, używanych powszechnie symboli itp.

Przystępując do projektowania schematu blokowego, należy:

- określić parametry programu,
- sporządzić listę założeń, zawierającą m.in. opis operacji i ich wzajemne powiązania,

- określić logiczną kolejność przetwarzania przez zbudowanie ogólnego schematu programu; należy dążyć, aby program przetwarzał największą liczbę danych najmniejszą liczbą instrukcji, w możliwie najszybszy sposób; konstrukcja ogólnego schematu programu wpływa na szybkość przetwarzania całego programu,

- wykreślić możliwie na jednej stronie zwarty schemat blokowy, który zawierałby wszystkie logiczne wypadki i powiązania, a następnie dla każdego bloku tego schematu sporządzić bardzo szczegółowe schematy blokowe.

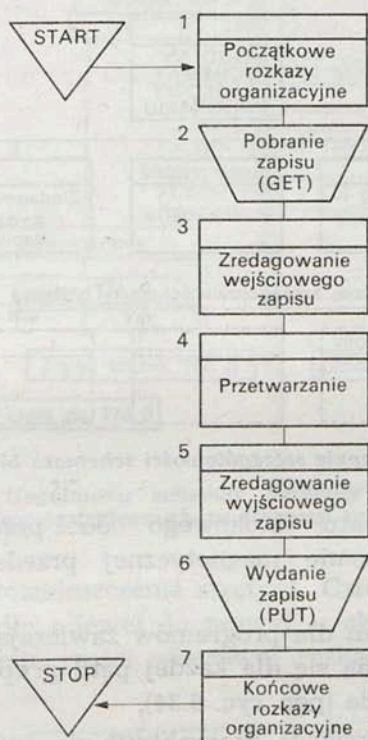
Ogólny schemat blokowy składa się na ogół z siedmiu modułów (ryc. 6.31):



1) początek programu, składający się z grupy rozkazów organizacyjnych ustawiających klucze, liczniki, strefy, indeksy, rejestry, deklaracje itp.,

2) otrzymanie zapisu, które polega np. na pobraniu z karty dziurkowanej lub pamięci zewnętrznej, w języku maszyny będzie to podprogram, natomiast w autokodach będzie to rozkaz typu GET (pobierz),

3) zredagowanie zapisu z przystosowaniem jego struktury do wymagań przetwarzania,



Ryc. 6.31. Ogólny schemat blokowy programu

4) przetwarzanie; właściwa grupa operacji realizujących zadaną metodę obliczeń danego programu,

5) zredagowanie zapisu z przystosowaniem go do wydruku lub wydziurkowania lub też umieszczenia w pamięci zewnętrznej,

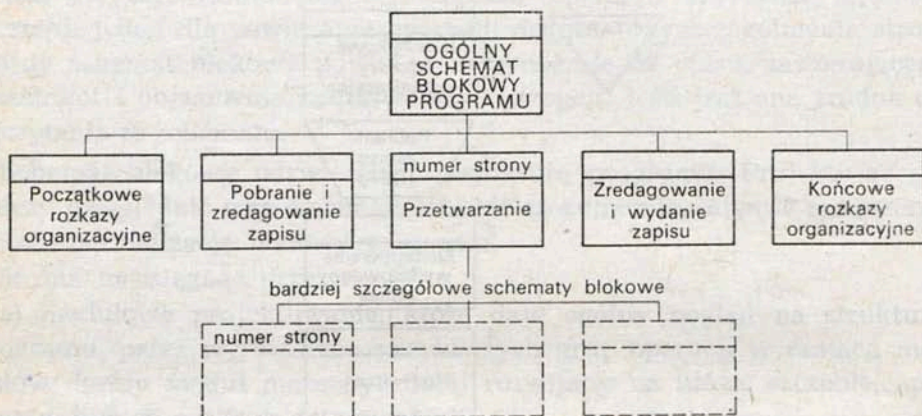
6) wyprowadzenie wyników przetwarzania, które odpowiada modułowi 2, tylko w odwrotnym kierunku, realizowane rozkazem autokodowym PUT (wydaj-zapisz),

7) koniec programu — grupa operacji organizacyjnych realizująca proces odwrotny do modułu 1.



Wszystkie operacje wejściowo-wyjściowe kierowane są dwoma modułami ogólnego schematu blokowego. W ogólnym schemacie blokowym programu podane są wszystkie rozgałęzienia logiczne (predykaty) programu. Nie może tu wystąpić inna grupa operacji przetwarzania o tym samym stopniu dokładności, poza wyróżnionymi zasadniczymi modułami.

Dalsze szczegółowe schematy blokowe rozpisywane są dla poszczególnych modułów. W górnym symbolu modułu podaje się numer strony, na której jest umieszczony szczegółowy schemat blokowy (por. ryc. 6.32).



Ryc. 6.32. Stopnie szczegółowości schematu blokowego

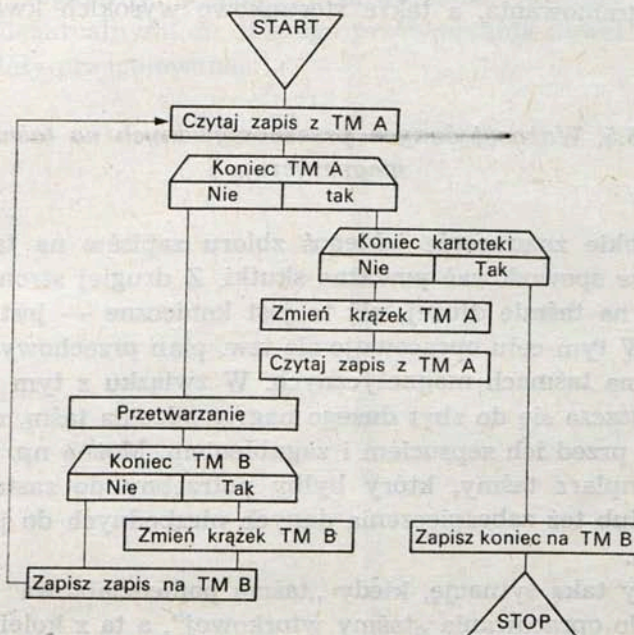
Projektowanie schematu blokowego do przetwarzania zapisów zarejestrowanych na taśmie magnetycznej przedstawione zostało na ryc. 6.33.

W schemacie blokowym dla programów zawierających kilka pętli iterowania obliczeń wyróżnia się dla każdej pętli grupę rozkazów ustawiających warunki iterowania (por. ryc. 6.34).

Budowa schematu blokowego skomplikuje się, jeśli chcemy wykorzystać w pełni możliwość równoczesnego przetwarzania. Należy wtedy utworzyć dwie strefy czytania i dwie strefy pisania. Technika ta nosi nazwę przetwarzania wahadłowego. Podczas wczytywania do strefy 1 ze strefy 2 dane pobierane są do przetwarzania i na odwrót. Dzięki temu realizuje się jednocześnie dwie funkcje (czytania i przetwarzania). Technika wahadłowa wymaga dwa razy większej liczby zarezerwowanych stref w pamięci operacyjnej. Zmiana stref odbywa się kolejno i część programu, która to realizuje, umieszczona jest między podprogramem, badającym gotowość urządzenia zewnętrznego do pracy a rozkazem wejściowo-wyjściowym, uruchamiającym bezpośrednie działanie danego urządzenia, np. czytnika. Dla każdego czytania (przetwarzania), pisania wyodrębnia się cztery logiczne komórki pamięci operacyjnej, które sta-

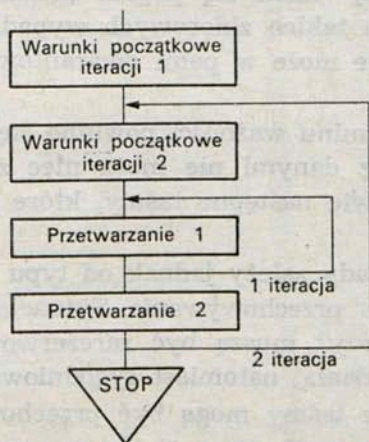


nowią wahadło. W pierwszej komórce podaje się adres symboliczny strefy 1, w drugiej — listę rozmieszczenia danych w strefie 2. W komórce trzeciej podaje się adres symboliczny strefy 2, a w komórce



Ryc. 6.33. Uogólniony schemat blokowy przetwarzania danych zarejestrowanych na taśmach magnetycznych

czwartej podaje się rozmieszczenie strefy 1. Cztery logiczne słowa przesuwają się jak wahadło z lewej do prawej w akumulatorze, skąd pobierane są do wypełnienia części adresowej rozkazu, np. czytania. Waha-



Ryc. 6.34. Schemat blokowy obliczeń iteracyjnych



dło zmienia adresy strefy wczytywania i przetwarzania. Do wczytywania przeznacza się adres strefy 1, do przetwarzania — adres strefy 2.

Technika wahadłowego programowania jest stosunkowo rzadko wykorzystywana, wymaga bowiem programowania w szczegółowych językach programowania, a także stosunkowo wysokich kwalifikacji programisty.

#### *6.8.5. Ważność danych przechowywanych na taśmach magnetycznych*

Zbyt szybkie zniszczenie jakiegoś zbioru zapisów na taśmie magnetycznej może spowodować poważne skutki. Z drugiej strony przechowywanie jego na taśmie dłużej, niż to jest konieczne — jest również niewskazane. W tym celu opracowuje się tzw. plan przechowywania danych zapisanych na taśmach magnetycznych. W związku z tym z jednej strony nie dopuszcza się do zbyt dużego nagromadzenia taśm, a jednocześnie zabezpiecza przed ich zepsuciem i zagubieniem. Można np. zarezerwować drugi egzemplarz taśmy, który byłby potrzebny do zastąpienia taśmy brakującej lub też zabezpieczenia danych niezbędnych do jej ponownego odtworzenia.

Rozważmy taką sytuację, kiedy „taśma poniedziałkowa” zawiera dane wejściowe do opracowania „taśmy wtorkowej”, a ta z kolei zawiera dane wejściowe do opracowania „taśmy środowej”. Jeśli w toku przetwarzania w środę, „taśma wtorkowa” ulegnie zniszczeniu, „taśma poniedziałkowa” będzie nadal służyła do ponownego przygotowania „taśmy wtorkowej”. Żadna „taśma” wcześniejsza od „poniedziałkowej” nie będzie potrzebna. Oczywiście może się zdarzyć, że przy przygotowywaniu „taśmy wtorkowej” zepsuje się „taśma poniedziałkowa”, a podczas przygotowywania „taśmy poniedziałkowej” może się popsuć „taśma sobotnia”. Jednakże możliwości wystąpienia takich zbiorowych wypadków są znikome. Plan wygaśnięcia danych nie może w pełni gwarantować zapobieżenia takiej sytuacji.

Przy planowaniu terminu ważności powinno się kierować następującą zasadą: żadna taśma z danymi nie może ulec zniszczeniu, zanim nie zostaną sporządzone dwie następne taśmy, które opracowuje się na jej podstawie.

Zastosowanie tej zasady zależy jednak od typu przechowywanych danych oraz od warunków przechowywania. Sytuacja ta może np. ulec zmianie, gdy „dzienne taśmy” muszą być zarezerwowane do sporządzenia tygodniowego sprawozdania, natomiast tygodniowe do miesięcznego itd.

W praktyce niektóre taśmy mogą być przechowywane trochę dłużej niż teoretycznie jest to wymagane, tj. do momentu, gdy da się precyzyjnie określić występowanie takich wypadków w procesie przetwarzania,



jak drukowanie sprawozdania po dłuższym okresie. Plan określenia terminów przechowywania, chociaż teoretycznie dobry, w praktyce może nie spełniać swego zadania. Jest rzeczą bardzo trudną odniesienie wszystkich warunków systemu APD do okresów czasowych. Różne opóźnienia mogą popsuć regularny schemat przetwarzania. Również wiele taśm staje się już nieaktualnymi do dalszego przetwarzania nawet już w dniu, w którym zostały przygotowane.