M E R A - 1 0 0

DOKUMENTACJA EKSPLOATACYJNA SYSTEMU OPERACYJNEGO KD-100

OPIS JĘZYKA SYMBOLICZNEGO

Opracowali: Zbigniew Daleczko
Tadeusz Nabożny
Andrzej Brzozowski

Warszawa - Wałbrzych - luty 1985

S P I S T R E Ś C I

I. OPIS SYSTEMU KD-100

I.1. Wstęp 4
I.2. Symbolika stosowana w opisie 5
I.3. Struktura systemu, uruchamianie 6
I.4. Podział pamięci minikomputera w systemie KD-100 .. 8
I.5. Rejestry robocze 9
I.6. Buforowanie w systemie KD-100 10
I.7, Sygnalizacja lampek 12
I.8. Etykiety 13
I.9. Rejestr warunku 13
I.10. Współpraca z pamięcią kasetową 14
I.11. Współpraca z pamięcią dyskową 16
I.12. Współpraca z monitorem ekranowym 18

II. FUNKCJE REALIZOWANE PRZEZ EDYTOR

II.1. Wprowadzanie programu z klawiatury 21
II.2. Czytanie programu 23
II.3. Zapis programu i listowanie biblioteki 24
II.4. Korekta programu 26
II.5. Listowanie programu 28
II.6. Odczyt sektora 29
II.7. Zapis sektora 30
II.8. Listowanie PAO 31
II.9. Korekta PAO 32

III. FUNKCJE REALIZOWANE PRZEZ EGZEKUTOR

III.1. Listowanie programu 33
III.2. Kompilacja programu 33
III.3. Realizacja /wykonanie/ programu 34

IV. OPIS INSTRUKCJI JĘZYKA SYMBOLICZNEGO

IV.1. Dyrektywy 35
IV.2. Instrukcje sterujące 36
IV.3. Instrukcje arytmetyczne 38
IV.4. Przesłania 44
IV.5. Instrukcje tekstowe 47
IV.6. Instrukcje wejścia/wyjścia 48
IV.7. Instrukcje kasetowe i dyskowe 56
IV.8. Instrukcje specjalne 60

V. DODATKI

V.A. Kody znaków i rozkazów w systemie MERA-100 66
V.B. Lista instrukcji języka symbolicznego KD-100 69
V.C. Schematy realizacji funkcji edytora 72

I. OPIS SYSTEMU KD-100

I.1. WSTĘP

System operacyjny KD-100 został opracowany przede wszystkim w celu przyspieszenia i ułatwienia procesów programowania dla minikomputera MERA-100 poprzez stworzenie symbolicznego języka programowania, opartego na języku zastosowanym w systemie Komputer Biurowy, który był podstawowym oprogramowaniem użytkowym polskich minikomputerów MERA serii 300 /konkretnie MERA-302 do MERA-305/. Język ten wybrano ze względu na ograniczone możliwości minikomputera MERA-100 /mała pojemność pamięci, bardzo uproszczony język maszynowy itp./.

Język symboliczny systemu KD-100 jest przeznaczony przede wszystkim do rozwiązywania problemów ekonomicznych. Umożliwia on m.in.:

- działania na wartościach alfanumerycznych i numerycznych,
- zakładanie i odczyt zbiorów na taśmach kasetowych,
- zakładanie i odczyt zbiorów na dyskach elastycznych,
- stosowanie tablic jednowymiarowych,
- proste operacje na tablicach,
- wprowadzanie wstawek w języku maszynowym MERA-100,

Tworząc system autorzy postawili sobie za cel nie tylko ułatwienie przygotowywania programów do rejestracji danych, ale również stworzenie możliwości efektywnego wykorzystania minikomputera MERA-100 do samodzielnego przetwarzania prostych systemów. Temu drugiemu celowi będzie służyło ponadto opracowanie szeregu pomocniczych programów standardowych.

System KD-100 umożliwia współpracę z następującą, maksymalną konfiguracją minikomputera:

- pamięć operacyjna 16 KB,
- monitor ekranowy,
- drukarka DZM-180 z klawiaturą,
- 2 jednostki pamięci kasetowej PK-1,
- 2 jednostki pamięci na dyskach elastycznych /4 kieszenie/.

Możliwa jest ponadto praca w systemie na następujących, uproszczonych konfiguracjach minikomputera:

- pamięć operacyjna 16 KB,
- monitor ekranowy,
- drukarka z klawiaturą,
- 1 lub 2 jednostki pamięci kasetowej PK-1,

albo:

- pamięć operacyjna 16 KB,
- monitor ekranowy,
- drukarka z klawiaturą,
- 1 lub 2 jednostki pamięci na dyskach elastycznych.

System KD-100 wykorzystuje w szrokim zakresie podprogramy firmowe zawarte w pamięci stałej PROM. Sam system zajmuje w PAO strony od 010 - 039. Pozostała część pamięci dzieli się pomiędzy program użytkowy i pamięć dla danych programu użytkowego. Pamięć dla danych programu użytkowego stanowi w systemie KD-100 maksymalnie 256 t.zw. rejestrów roboczych o długości 16 B. W każdym z nich możliwe jest wykonywanie wszelkich, dostępnych w języku symbolicznym, operacji arytmetycznych i logicznych na danych. Na program użytkowy /po kompilacji/, przy wykorzystaniu maksymalnej ilości rejestrów roboczych przypada 8 stron pamięci, przy średniej długości instrukcji 6 - 8 B. Obszar ten można zwiększyć kosztem zmniejszenia ilości rejestrów roboczych dostępnych dla danego programu.

Ze względu na drastyczne ograniczenia pamięciowe system ma ogromnie uproszczoną diagnostykę błędów.

I.2. SYMBOLIKA STOSOWANA W OPISIE

W celu łatwiejszego zrozumienia niniejszego opracowania podajemy poniżej dokładne wyjaśnienie różnych symboli używanych dalej przy opisywaniu działania instrukcji, podawaniu formatów instrukcji itp:

Rm - numer rejestru roboczego /liczba z zakresu Ø ÷ 255/

[Rm] - zawartość rejestru roboczego o numerze Rm

R(Rm)	- rejestr roboczy, którego numer wskazują trzy ostatnie pozycje cyfrowe rejestru Rm
[R(Rm)]	- zawartość rejestru roboczego, którego numer wskazują trzy ostatnie pozycje cyfrowe rejestru Rm
:=	- staje się równe
En	- numer etykiety /liczba z zakresu 1 ÷ 127, z wyjątkiem 124/
C	- rejestr warunkowy
/ttt....t/	- tekst złożony ze znaków mających postać graficzną /występujących na klawiaturze alfanumerycznej/ łącznie ze znakiem LF
± 999....9	- liczba złożona z cyfr, bez znaku kropki lub przecinka
Z	- wartość dziesiętna znaku w kodzie ISO-7
d	- ilość znaków lub pozycji /liczba z zakresu 1 ÷ 255, z wyjątkiem: 196,248,253/
S	- symbol urządzenia wejścia/wyjścia /1,4,5,6, 8,9/
(L)	- klawisz funkcyjny klawiatury
L	- klawisz alfanumeryczny /litera L/

I.3. STRUKTURA SYSTEMU, URUCHAMIANIE

System KD-100 opracowany został dla konfiguracji minikomputera MERA-100 z pamięcią operacyjną 16 KB. Ta niewielka stosunkowo pojemność pamięci pociągnęła za sobą konieczność rozbicia systemu na dwie części:

- Edytor /program o nazwie ED/
- Egzekutor /program o nazwie EG/.

Edytor realizuje następujące funkcje w systemie KD-100:

- wprowadzanie do PAO programów z klawiatury,
- zakładanie bibliotek programowych,
- wprowadzanie do PAO programów z bibliotek,
- korektę programów,
- listowanie programów na monitorze lub drukarce,
- odczyt i zapis dowolnych informacji w PAO lub pamięci dyskowej.

Egzekutor natomiast realizuje poniższe funkcje:
- listowanie programów źródłowych na drukarce,
- kompilację programów,
- realizację programów.

Obydwa te programy, wprowadzane do PAO w zależności od potrzeby, pracują w zasadzie niezależnie od siebie. Łączy je jedynie to, że edytor pozostawia w pamięci operacyjnej źródłową treść programu, który nie ulega zniszczeniu w trakcie wczytywania do PAO egzekutora i może być następnie przez niego skompilowany i zrealizowany. Jak z tego wynika wprowadzenie do PAO egzekutora może mieć sens tylko wtedy, jeżeli uprzednio w PAO została umieszczona przy pomocy edytora źródłowa treść programu.

System KD-100 napisany został przy użyciu firmowego zestawu programów o nazwie KOMPILATOR /programy 50 ÷ 57/. Użytkownik otrzymuje go w postaci dwóch programów wynikowych /ED i EG/, przetłumaczonych na język maszynowy MERA-100.

Programy te mogą być przechowywane zarówno na taśmie kasetowej jak i na dysku elastycznym. Obydwie części systemu, w zależności od nośnika na którym zostały umieszczone, wczytuje się i uruchamia jak każdy inny program w minikomputerze MERA-100:

1 - z dysku elastycznego:
- założyć dysk z systemem do kieszeni nr Ø,
- nacisnąć klawisze (SEL), (LK), {E,D / E,G}, (A)
- po wczytaniu program uruchomić klawiszem (A)

2 - z pamięci kasetowej:
- założyć kasetę z systemem,
- nacisnąć klawisze (SEL), (L), {E,D / E,G},
- po wczytaniu program uruchomić klawiszem (A).

W obydwu przypadkach po naciśnięciu klawisza (A) na ekranie monitora pojawiają się systemowe listy funkcji odpowiednio: edytora lub egzekutora. Odpowiednia część systemu jest w tym momencie w stanie gotowości do pracy, co jest sygnalizowane zapaleniem się lampek o nr nr: 5,6,7 i 8. Równocześnie pali się również lampka nr 9, sygnalizująca stan pracy systemu MERA-100.

I.4. PODZIAŁ PAMIĘCI MINIKOMPUTERA W SYSTEMIE KD-100

Podział pamięci operacyjnej pomiędzy różne elementy systemu KD-100 można zilustrować następująco:

	Str:
P R O M	ØØØ ⋮ Ø07
BUFOR PK-1 i FD	Ø08
REJESTR OPERACYJNY	Ø09
S Y S T E M K D - 1 Ø Ø Wymiennie: ED - Edytor, lub EG - Egzekutor	Ø1Ø ⋮ Ø39
LISTA ETYKIET	Ø4Ø
POCZĄTEK OBSZARU PROGRAMU WYNIKOWEGO	Ø41 ⋮
POCZĄTEK OBSZARU PROGRAMU ŹRÓDŁOWEGO	Ø46 ⋮
REJESTRY ROBOCZE	Ø48 ⋮
BUFOR WEJŚCIA	Ø62
BUFOR WYJŚCIA	Ø63

Jak widać z zamieszczonego powyżej schematu, pewne fragmenty pamięci, przeznaczone do różnych celów w systemie nie mają ściśle określonego zakończenia /poza oczywiście fizycznym końcem PAO/. W pewnych przypadkach mogą więc one na siebie zachodzić. Np. dłuższy program źródłowy może wejść na obszar rejestrów roboczych, co przed kompilacją i rozpoczęciem realizacji programu nie ma oczywiście żadnego znaczenia. Podobnie program wynikowy, jeżeli zajmie obszar większy niż 5 stron pamięci, skasuje odpowiednią część programu źródłowego, co nie będzie miało w zasadzie większego znaczenia. Jeszcze dłuższy program wynikowy może wejść jednak na obszar rejestrów roboczych. Nie spowoduje to jednak skasowania zawartości tych rejestrów, a jedynie odpowiednie zmniejszenie ich ilości, gdyż w takim przypadku obszar PAO przeznaczony na rejestry robocze zostanie przesunięty o odpowiednią ilość stron. Niedostępne wtedy staną się rejestry o najwyższych numerach.

I.5. REJESTRY ROBOCZE

System KD-100 stwarza możliwość korzystania w programach z maksymalnie 256-ciu tzw. rejestrów roboczych o numerach R000 ÷ R255 z tym, że nie wolno używać bezpośrednio, jako argumentów instrukcji rejestrów o numerach: 196, 248 i 253 /mogą one występować tylko wewnątrz tablic/. Rejestry robocze zajmują końcowy fragment pamięci poczynając od strony 048 do strony 063 z tym, że obszar odpowiadający rejestrom o numerach R222 ÷ R239 może być, w przypadku korzystania z odpowiednich urządzeń zewnętrznych, wykorzystywany jako bufor wejściowy i analogicznie, obszar odpowiadający rejestrom o numerach R240 ÷ R255 - jako bufor wyjściowy. W przypadku, jeżeli program nie korzysta z któregokolwiek z wymienionych buforów /lub obydwu/, odpowiadające mu rejestry są w tym programie dostępne.

Jak już wspomniano, ilość dostępnych rejestrów roboczych może ulec zmniejszeniu, jeżeli program wynikowy przekracza początek strony 048. Można to stwierdzić, jak również obliczyć ilość dostępnych w danym programie rejestrów, na podstawie adresu końcowego programu wynikowego, drukowanego po zakończeniu kompilacji programu.

Każdy rejestr roboczy składa się z 16 B, które w celach identyfikacyjnych ponumerowano umownie od 0 ÷ 15 /od strony lewej do prawej/. W każdym rejestrze roboczym może być przechowywana liczba maksymalnie 14 cyfrowa bądź tekst /fragment tekstu/ maksymalnie 15 znakowy.

Jeżeli w rejestrze roboczym przechowywana jest liczba, struktura jego jest następująca:

±	C	C	C	C	C	C	C	C	C	C	C	C	C	C	Ogr
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

gdzie: ± - znak algebraiczny liczby /plus - kod 207, minus - kod 206/
C - cyfra w kodzie ASCII
Ogr - znak końca rejestru - kod 255

Liczba w rejestrze traktowana jest zawsze jako liczba całkowita. Przecinek dziesiętny może zostać umieszczony w dowolnym miejscu podczas wyprowadzania liczby na zewnątrz. Liczba dosunięta jest zawsze do prawego końca rejestru, ewentualne wolne miejsca z lewej strony wypełnione są zerami.

Jeżeli w rejestrze roboczym przechowywany jest tekst, struktura rejestru jest następująca:

Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Ogr
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

gdzie: Z - poszczególne znaki tekstu w kodzie ASCII
Ogr- znak końca rejestru - kod 255

Tekst dosunięty jest zawsze do lewego końca rejestru /również pojedyńczy znak/, ewentualne wolne miejsca z prawej wypełnione są znakami spacji w kodzie ASCII.

I.6. BUFOROWANIE W SYSTEMIE KD-100

System KD-100 w zależności od rodzaju używanego w danym momencie przez program urządzenia zewnętrznego wykorzystywać może dwa bufory:

- wejściowy /strona 062 PAO/
- wyjściowy /strona 063 PAO/

Odpowiedni bufor angażowany jest do współpracy w momencie aktywizacji na wejściu lub wyjściu odpowiedniego urządzenia, a mianowicie:

- bufor wejścia: klawiatura
  pamięć kasetowa
  pamięć dyskowa
- bufor wyjścia: pamięć kasetowa
  pamięć dyskowa

Jak z tego wynika większość urządzeń zewnętrznych w systemie korzysta z odpowiedniego bufora, jedynie wyprowadzanie danych na monitor i drukarkę odbywa się bez pośrednictwa bufora wyjścia.

Aktywizacja wymienionych wyżej urządzeń związana jest zawsze ze skasowaniem /wyczyszczeniem/ odpowiedniego bufora.

W przypadku współpracy z pamięcią kasetową przeniesienie danych z PK-1 do bufora wejściowego odbywa się automatycznie, jako efekt zastosowania w programie dowolnej instrukcji wejścia. Na wyjściu natomiast wprowadzenie danych do bufora oraz przeniesienie zawartości bufora na PK-1 musi być zorganizowane w programie, poprzez użycie dowolnych instrukcji wyjścia a następnie instrukcji "pisz blok".

W przypadku współpracy z pamięcią dyskową wprowadzanie danych z dysku elastycznego do bufora wejścia musi być zagwarantowane w programie przy pomocy instrukcji "czytaj blok". Wykorzystanie bufora na wyjściu odbywa się natomiast analogicznie jak w przypadku pamięci kasetowej.

W przypadku współpracy z klawiaturą wykorzystaniem bufora wejścia steruje w odpowiedni sposób operator z klawiatury. Wszystkie znaki wprowadzone z klawiatury do bufora są niedostępne dla programu do momentu zaakceptowania bufora. Następuje to po naciśnięciu klawisza Ⓐ lub Ⓒ , przy czym:

- klawisz Ⓐ powoduje jedynie udostępnienie zgromadzonych w buforze znaków programowi /co jest związane z blokadą klawiatury aż do czasu opróżnienia bufora/,

- klawisz (C) powoduje dopisanie za ostatnim znakiem, wprowadzonym z klawiatury do bufora, znaku spacji i udostępnienie zawartości bufora programowi. Znajduje to zastosowanie przy wprowadzaniu informacji numerycznych /znak spacji pełni wtedy rolę ogranicznika liczby, który jest wymagany przez instrukcję "czytaj liczbę" języka symbolicznego/.

Przed zaakceptowaniem znaki zgromadzone w buforze można kasować w dwojaki sposób:

- klawiszem (Ck) - całą zawartość bufora. Sygnalizowane to jest:
  na monitorze - powrotem kursora na początek wiersza
  na drukarce - wydrukiem znaku "?" i przejściem do następnego wiersza.

- klawiszem (B) - ostatni wprowadzony znak. Klawisza (B) można używać wielokrotnie, kasując odpowiednią ilość znaków, licząc od końca bufora. Każde naciśnięcie klawisza (B) sygnalizowane jest:
  na monitorze - cofnięciem kursora o jeden znak,
  na drukarce - wydrukiem znaku "*".

I.7. SYGNALIZACJA LAMPEK

Stan gotowości do pracy obydwu części systemu KD-100, czyli edytora i egzekutora, sygnalizowany jest równoczesnym paleniem się lampek 5,6,7,8,9. W tym momencie na monitorze ekranowym wyświetlane są systemowe listy funkcji odpowiednio edytora lub egtekutora.

Od momentu rozpoczęcia realizacji programu na lampkach 1 - 8 wyświetlane są zawsze kody binarne dwóch ostatnich cyfr numeru etykiety, przez którą przebiega w danym momen-

cie realizacja programu, w następujący sposób:

- na lampkach 1 - 4: ostatnia cyfra etykiety
- na lampkach 5 - 8: przedostatnia cyfra etykiety.

Omawiane lampki zapalane są w momencie realizacji odpowiednich instrukcji deklaracji etykiety. Lampka 9 gaśnie jedynie w dwóch sytuacjach:

- podczas wykonywania kompilacji programu,
- podczas oczekiwania na wprowadzenie informacji z klawiatury, w trakcie wykonywania instrukcji wejścia.

W przypadku wystąpienia błędu związanego z obsługą systemu zapala się lampka FB, zaś w przypadku błędnej współpracy z PK-1 lub pamięcią dyskową - lampka FD. Zapalenie którejkolwiek z tych lampek powoduje blokadę całego systemu. Odblokowanie systemu, związane ze zgaszeniem lampki FB lub FD, następuje po naciśnięciu klawisza (Ck) /cancel/.

W momentach współpracy systemu z pamięcią PK-1 lub pamięcią dyskową pali się zawsze czerwona lampka L.

I.8. ETYKIETY

Etykieta w systemie KD-100 jest symbolicznym adresem, który możemy nadawać poszczególnym instrukcjom programu, a także całym podprogramom.

Maksymalna liczba etykiet, jakie stosować można w jednym programie, wynosi 126. Stanowią je numery w zakresie 1 - 127 /z wyjątkiem numeru 124/. Numery etykiet zwykłych, będących adresami instrukcji programu, oraz numery etykiet będących oznaczeniami podprogramów, nie mogą się w jednym programie powtarzać.

I.9. REJESTR WARUNKU

W systemie KD-100 znajduje się t.zw. rejestr warunku C, czyli jedobajtowa komórka do której wpisywane mogą być cztery różne informacje, t.zw. warunki, oznaczane umownie symbolami: W0, W1, W2 i W3. Warunki te wpisywane są do rejestru C przez różne instrukcje języka symbolicznego KD-100,

sygnalizując w ten sposób różne sytuacje powstałe w wyniku realizacji danej instrukcji.

Wykorzystanie zawartości rejestru C /sprawdzenie aktualnie ustawionego warunku/ nastąpić może wyłącznie przez zastosowanie odpowiednich instrukcji skoku warunkowego.

UWAGA: badanie zawartości rejestru C należy w programach umieszczać bezpośrednio po instrukcji, której wykonanie chcemy sprawdzić. Wynika to z faktu, że pewne instrukcje języka symbolicznego, w zasadzie nie ustawiające rejestru C, mogą go jednak ustawiać w sposób przypadkowy na skutek tego, że wiele instrukcji korzysta z podprogramów, zawartych w innych instrukcjach, wśród których mogą być instrukcje ustawiające rejestr warunku C.

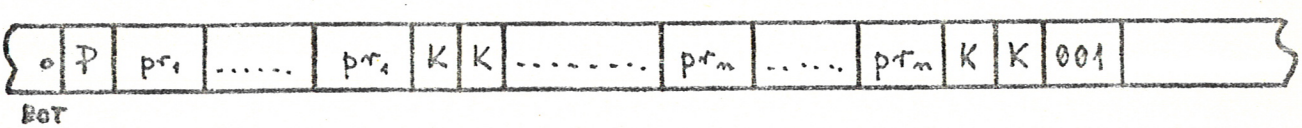
I.10. WSPÓŁPRACA Z PAMIĘCIĄ KASETOWĄ

W systemie KD-100 na taśmach kasetowych możliwe jest przechowywanie zarówno bibliotek programów /tylko w postaci źródłowej - przed kompilacją/ jak i zbiorów danych.

Taśma kasetowa, na której zakładana jest biblioteka programów /podczas zapisywania pierwszego programu na taśmie/ zostaje przed jego zapisem odpowiednio zametrykowana przez system. Polega to na zapisaniu na początku taśmy jednoznakowego bloku zawierającego literę "P". Jest to równoznaczne ze skasowaniem poprzedniej zawartości taśmy kasetowej. Do tak założonej biblioteki można następnie wprowadzić praktycznie dowolną ilość kolejnych programów.

Każdy program kończony jest dwoma jednoznakowymi blokami zawierającymi każdy literę "K". Po ostatnim programie w bibliotece zapis ten jest uzupełniany ponadto blokiem zawierającym kod binarny "001".

Strukturę taśmy ka etowej zawierającej bibliotekę programów przedstawia schematycznie poniższy rysunek:



UWAGA: Podczas dopisywania dalszych programów do bibliotek, jak również podczas pobierania programów z bibliotek, system KD-100 współpracuje bez zakłóceń również z taśmami zametrykowanymi w inny sposób niż opisany powyżej. Umożliwia to m.in.:

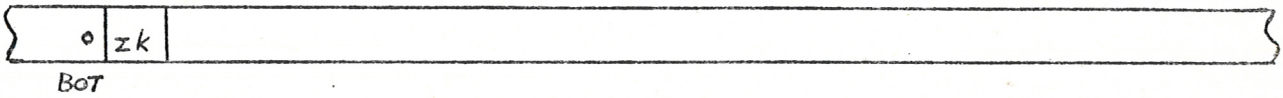
- dopisywanie programów źródłowych w języku symbolicznym do bibliotek programów założonych programami standardowymi systemu MERA-100,
- korzystanie z bibliotek programów założonych przez poprzednią wersję systemu o nazwie KB-100 itp.

Dane na taśmie kasetowej mogą być zapisywane i odczytywane blokami, których wielkość fizyczna wynosi maksymalnie 256 B. Bloki w ramach jednego zbioru mogą być zmiennej długości. Zapis zbioru danych na taśmie kasetowej musi być poprzedzony zametrykowaniem taśmy, czego dokonuje się przy pomocy specjalnej instrukcji w programie. Jest to równoważne zapisaniu na taśmie /za znacznikiem BOT/ znacznika końca zapisu. Znacznik końca zapisu w systemie KD-100 jest trzybajtowym blokiem wypełnionym kodami 255. Zapis pierwszego bloku danych powoduje skasowanie tego znacznika, który zostanie zapisany ponownie dopiero za ostatnim blokiem danych, w momencie zamknięcia zbioru /również przy pomocy specjalnej instrukcji języka symbolicznego/.

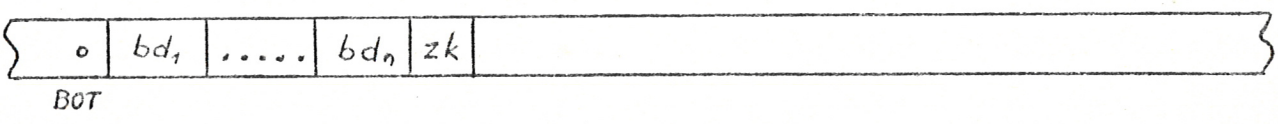
W systemie KD-100 możliwe jest zapisanie /i odczyt/ na jednej taśmie dwóch i więcej zbiorów danych.

Struktura taśmy z danymi w systemie KD-100 może więc przybierać następujące postaci:

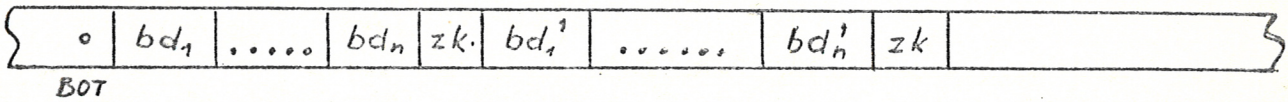
- taśma bezpośrednio po zametrykowaniu:



- taśma z zapisanym jednym zbiorem:



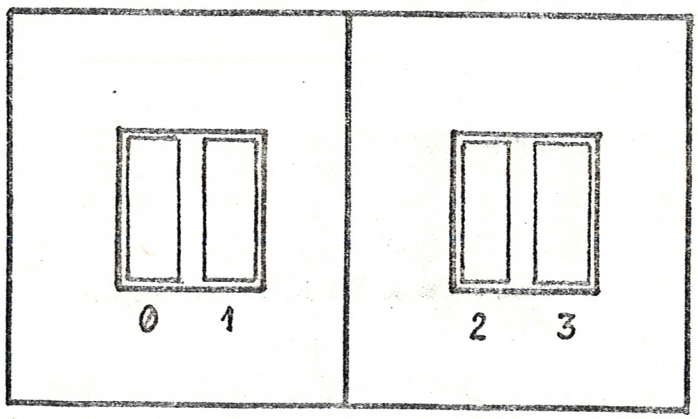
- taśma z zapisanymi dwoma zbiorami:



gdzie: zk - znacznik końca zapisu
bd - blok danych

I.11. WSPÓŁPRACA Z PAMIĘCIĄ DYSKOWĄ

System KD-100 może współpracować z jedną lub dwiema jednostkami pamięci na dyskach elastycznych. Każda jednostka zawiera dwie t.zw. kieszenie, czyli urządzenia zapisu/odczytu dla jednego dysku elastycznego każda. Poszczególne kieszenie, dla ułatwienia ich identyfikacji, są zawsze numerowane od lewej do prawej strony w następujący sposób:



Zapis informacji na dysku elastycznym dokonywany jest jednostronnie na t.zw. ścieżkach, które mają kształt okręgów koncentrycznie położonych na powierzchni dysku. Na całej powierzchni dysku elastycznego mieści się 77 ścieżek, o numerach od Ø do 76. Każda ze ścieżek dzieli się na 26 sektorów ponumerowanych od 1 do 26. Sektor ma pojemność odpowiadającą 128 bajtom. Tak więc pojemność jednego dysku elastycznego wynosi około 256 KB.

Zarówno zapis jak i odczyt informacji do/z pamięci dyskowej może się odbywać jedynie poszczególnymi, kompletnymi sektorami. W jednym i drugim przypadku, aby dotrzeć do in-

teresującego nas sektora, posługujemy się sześciocyfrowym adresem dyskowym o następującej strukturze: SSSKKK,

gdzie: SSS - nr ścieżki /ØØØ ÷ Ø76/
KKK - nr sektora /ØØ1 ÷ Ø26/

Dysk elastyczny, którego taki adres dotyczy /czyli numer kieszeni do której został załadowany/, określa się oddzielnie przy pomocy specjalnej instrukcji.

Analogicznie jak w przypadku pamięci kasetowej w systemie KD-100 na dyskach można przechowywać zarówno biblioteki programów /tylko źródłowych/ jak i zbiory danych.

Dysk elastyczny, na którym chcemy założyć bibliotekę programów, musi zostać uprzednio w odpowiedni sposób zainicjowany. Polega to na zapisaniu na ścieżce nr ØØØ programu wczytującego /loadera dyskowego/, nazwy dysku oraz założeniu katalogu programów /a właściwie wyzerowaniu obszaru na którym zostanie założony katalog/. Każdy program, wprowadzany następnie do tak założonej biblioteki, będzie odnotowany w katalogu w postaci nazwy i adresu, pod którym został umieszczony na dysku.

'Inicjacja dysku programowego stanowi jedną z funkcji edytora systemu i zostanie omówiona w paragrafie II.3. Również przy pomocy edytora możemy dokonywać listowania katalogu programów, otrzymując na ekranie monitora spis wszystkich programów znajdujących się w danym momencie w bibliotece.

UWAGA: zarówno zapis jak i odczyt programów do/z biblioteki może mieć miejsce tylko wtedy, jeżeli dysk elastyczny zawierający daną bibliotekę jest załadowany do kieszeni nr Ø.

Dysk elastyczny, na którym chcemy zapisywać zbiory danych, nie wymaga żadnego uprzedniego przygotowania. System nie posiada również możliwości zakładania katalogu zbiorów zapisanych na dysku.

Dane są zapisywane od adresu podanego przez programistę w sposób określony również przez program. Każdy zbiór danych powinien zostać zakończony znacznikiem końca zapisu. Dokonuje się tego przy pomocy tej samej instrukcji kz, co w przypadku zbiorów na taśmach kasetowych.

Poza możliwościami stworzonymi przez system KD-100, przy pracy na dyskach elastycznych bardzo pomocne mogą być następujące programy standardowe, wchodzące w skład oprogramowania firmowego:

20 - kopiowanie programów z PK-1 na dysk nr Ø
21 - kopiowanie programów z dysku nr Ø na PK-1
22 - kopiowanie wybranych programów z dysku nr Ø na dysk nr 1
23 - kopiowanie informacji z dysku nr Ø na dysk nr 1
24 - kopiowanie informacji z dysku nr 1 na PK-1
25 - kopiowanie informacji z PK-1 na dysk nr 1
26 - korekta nazw programów na dysku nr Ø.

I.12. WSPÓŁPRACA Z MONITOREM EKRANOWYM

System KD-100 może współpracować zarówno z monitorem ekranowym VDU /12 x 40/ jak i CRT /24 x 80/.

Monitor ekranowy pełni przede wszystkim rolę konsoli systemowej, będąc wraz z klawiaturą pośrednikiem w komunikowaniu się operatora minikomputera z systemem operacyjnym. Realizacja wszystkich funkcji systemu odbywa się za pośrednictwem monitora. Ponadto spełnia on również rolę urządzenia we/wy minikomputera.

Na wejściu monitor może jedynie współpracować z klawiaturą alfanumeryczną w identyczny sposób jak drukarka, t.zn. że przy odpowiedniej deklaracji wejścia z klawiatury wszystkie znaki odpowiadające naciśniętym klawiszom wprowadzane są do bufora wejścia i jednocześnie wyświetlane na ekranie monitora w miejscu określonym przez położenie kursora w danym momencie. Ekran monitora odzwierciedla w tej sytuacji zawartość systemowego bufora wejścia. Wszystkie manipulacje dokonywane na zawartości bufora /np. kasowanie/ znajdują odpowiednie odzwierciedlenie na ekranie monitora /patrz paragraf I.6./.

Na wyjściu natomiast monitor jest autonomicznym urządzeniem wyprowadzania danych. Działa w analogiczny sposób jak drukarka, bez pośrednictwa systemowego bufora wyjścia.

Rozmieszczeniem na ekranie wyprowadzanych na monitor informacji oraz sposobem wyświetlania można sterować programowo przy pomocy odpowiednich kodów funkcyjnych. Do najważniejszych z nich należą:

8 - kursor w lewo o 1 pozycję
9 - kursor w prawo o 1 pozycję
10 - kursor w dół o 1 pozycję
11 - kursor w górę o 1 pozycję
12 - zerowanie ekranu /kursor na początek ekranu/
13 - kursor do początku wiersza
14 - kursor do początku ekranu
15 - wstawienie znaku
16 - anulowanie znaku
17 - tabulacja pozioma
18 - zapalenie kursora
19 - tabulacja pionowa
20 - zgaszenie kursora
22 - wstawienie linii
23 - anulowanie linii

UWAGA: wartości wymienionych kodów funkcyjnych monitora podano w kodzie ISO-7, gdyż instrukcja "pisz znak" /wl/ języka symbolicznego, najwygodniejsza do przesyłania kodów funkcyjnych na monitor, wymaga podawania argumentu właśnie w kodzie ISO-7.

Poza wyżej wymienionymi mogą znaleźć ponadto zastosowanie jeszcze cztery następujące funkcje monitora:

129 - rozjaśnianie ekranu
144 - odwrotne video - wyświetlanie ciemnych znaków na jasnym tle
130 - migotanie - znaki wyświetlane na ekranie gasną i pojawiają się na przemian
160 - podkreślanie - wszystkie znaki wyświetlane na ekranie są podkreślone.

Każda z powyższych czterech funkcji działa aż do momentu wyłączenia. Dokonuje się tego, w stosunku do każdej z

nich, przy pomocy kodu 128. Ponadto należy pamiętać, że każda z tych funkcji zajmuje jeden znak na ekranie monitora.

II. FUNKCJE REALIZOWANE PRZEZ EDYTOR

Po uruchomieniu edytora na ekranie monitora ukazuje się następująca lista funkcji:

1 - WPROWADZANIE PROGRAMU Z KLAWIATURY
2 - CZYTANIE PROGRAMU
3 - ZAPIS PROGRAMU I LISTOWANIE BIBLIOTEKI
4 - KOREKTA PROGRAMU
5 - LISTOWANIE PROGRAMU
6 - ODCZYT SEKTORA
7 - ZAPIS SEKTORA
8 - LISTOWANIE PAO
9 - KOREKTA PAO

Wyboru żądanej funkcji dokonuje się naciskając odpowiedni klawisz klawiatury numerycznej, bez jego akceptowania. Wszystkie wprowadzane następnie z klawiatury informacje /łącznie z treścią programu - funkcja 1/ są przekazywane systemowi za pośrednictwem bufora. Wymagają więc akceptu klawiszem (A) lub (C), a także można je kasować przed zaakceptowaniem klawiszem (B) lub (Ck) /patrz paragraf I.6./.

UWAGA: Poza wymienionymi w liście funkcji w systemie KD-100 działa jeszcze dodatkowa funkcja Ø. Powoduje ona wyświetlenie na ekranie monitora listy programów zawartych w bibliotece na dysku elastycznym, znajdującym się w danym momencie w kieszeni nr Ø /patrz paragraf II.3./.

II.1. WPROWADZANIE PROGRAMU Z KLAWIATURY

Po uruchomieniu funkcji 1 na ekranie monitora pojawia się tekst:

WPROWADZANIE PROGRAMU Z KLAWIATURY

NAZWA PROGRAMU

Należy podać nazwę programu, jaką chcemy nadać programowi, który będziemy wprowadzać do PAO oraz zaakceptować ją klawiszem Ⓐ . Nazwę programu w systemie KD-100 stanowią dwa dowolne znaki alfanumeryczne. Bezpośrednio po tym przystępujemy do wprowadzania treści programu, czyli kolejnych, składających się na niego instrukcji, które będą umieszczane w PAO. Części operacyjne /kody operacji/ wszystkich instrukcji podajemy koniecznie małymi literami. Każdą instrukcję bez wyjątku kończy się średnikiem, po którym należy ją zaakceptować klawiszem Ⓐ . Poszczególne wprowadzane instrukcje będą wyświetlane na monitorze, każda w oddzielnym wierszu. Sposób wprowadzania instrukcji zależy od ich budowy:

- w przypadku instrukcji bezargumentowych /dz, sh, mt, cf, kz, pb, cb, pk/:
  bezpośrednio po wprowadzeniu kodu operacji wprowadzamy znak średnika,

- w przypadku instrukcji jednoargumentowych /dl, sk, st, sr, wi, ms, wn, wo, wa, sq, cn, we, wy, qt, wl, cj, zz, wd, pg/:
  po wprowadzeniu kodu operacji należy wprowadzić spację a następnie argument /1 ÷ 3 cyfr/ i zakończyć go średnikiem.

- w przypadku instrukcji dwuargumentowych /rd, ro, rx, rm, ri, pw, rp, rl, rz, ai, am,im, az, qq, al, qy, ru, pu, ph,pz, en, zr, pd, ra, rb, pp, pt, za/:
  po wprowadzeniu kodu operacji należy wprowadzić spację, pierwszy argument, ponownie spację i drugi argument zakończony średnikiem.

Od powyższych reguł nieco odbiega sposób wprowadzania trzech, nie wymienionych dotychczas instrukcji:

wt - bezpośrednio po wprowadzeniu kodu operacji wprowadzamy znak "/" (slash), następnie kolejne znaki tekstu, znak "/" i średnik kończący instrukcję,

lr i ts - po wprowadzeniu kodu operacji wprowadzamy spację, pierwszy argument będący numerem rejestru, znak "/", kolejne znaki tekstu lub cyfry składające się na liczbę /poprzedzoną koniecznie znakiem algebraicznym plus lub minus/, znak "/" i średnik.

Nie zastosowanie się do powyższych reguł powoduje wyświetlenie komunikatu "*B" bezpośrednio przy nieprawidłowo wprowadzonej instrukcji. Instrukcja taka zostanie przez edytor pominięta , toteż należy ją wprowadzić ponownie. Zaleca się, aby przed naciśnięciem klawisza (A) dokonywać wzrokowej kontroli poprawności instrukcji. Instrukcję błędnie wprowadzoną można jak wiadomo, przed zaakceptowaniem, poprawić w całości lub tylko poszczególne znaki.

Po wprowadzeniu ostatniej instrukcji programu należy klawiszem (ST) spowodować zakończenie funkcji wprowadzania. Klawiszem tym możemy również przerwać wprowadzanie po dowolnej instrukcji programu,

Klawisz (ST) powoduje wykasowanie ekranu i powrót kursora na jego początek. Naciśnięcie następnie dowolnego spośród klawiszy (A) , (B) lub (C) spowoduje wyświetlenie listy funkcji edytora, czyli jego powrót do stanu wyjściowego.

II.2. CZYTANIE PROGRAMU

Po uruchomieniu funkcji 2 - wprowadzanie programu do PAO z biblioteki programów zapisanych na dysku elastycznym lub taśmie kasetowej - na monitorze ukazuje się tekst:

WYBIERZ NOSNIK PROGRAMU
<L> - KASETA
<LK> - DYSK

W zależności od urządzenia, na którym znajduje się biblioteka programów, naciskamy klawisz funkcyjny (L) lub (LK) po uprzednim załadowaniu kasety do pamięci PK-1 lub dysku elastycznego do kieszeni nr Ø. W następnej kolejności należy wprowadzić z klawiatury nazwę żądanego programu, którą /tylko w przypadku dysku elastycznego/ należy zaakceptować klawiszem (A) .

Jeżeli program zostanie odnaleziony w bibliotece, to bezpośrednio po jego prawidłowym wprowadzeniu do PAO edytor wraca do stanu wyjściowego /na ekranie ukazuje się lista funkcji edytora/.

Jeżeli natomiast system nie jest w stanie wprowadzić do PAO żądanego programu, to w zależności od przyczyny która

to spowodowała, po nazwie programu, na ekranie ukazuje się jeden z następujących komunikatów:

K? - oznaczający, że w odpowiednim urządzeniu pamięci zewnętrznej brak jest dysku elastycznego lub taśmy kasetowej, albo że na danym dysku lub taśmie nie ma wogóle biblioteki programów,

P? - oznaczający, że w danej bibliotece nie ma programu o żądanej nazwie.

W obydwu przypadkach naciśnięcie dowolnego klawisza spośród (A), (B) lub (C) powoduje wyświetlenie listy funkcji edytora na ekranie,

II.3. ZAPIS PROGRAMU I LISTOWANIE BIBLIOTEKI

Funkcja 3 umożliwia zakładanie bibliotek programów na kasecie magnetycznej lub dysku elastycznym, wprowadzanie do nich umieszczonych wcześniej w PAO programów źródłowych oraz listowanie katalogu programów biblioteki znajdującej się na dysku elastycznym /biblioteki programów zakładane na taśmach kasetowych nie posiadają katalogów/.

Po uruchomieniu funkcji na ekranie monitora ukazuje się komunikat:

WYBIERZ NOSNIK PROGRAMU
<L> - KASETA
<LK> - DYSK

Naciskając klawisz funkcyjny (L) lub (LK) określamy urządzenie, na którym chcemy założyć nową bibliotekę programów, lub na którym znajduje się już biblioteka, do której chcemy wprowadzić kolejny program. Należy pamiętać o uprzednim załadowaniu kasety magnetycznej do pamięci PK-1 lub dysku elastycznego do kieszeni nr Ø.

Zarówno po naciśnięciu klawisza (L) jak i (LK) w następnej kolejności na ekranie monitora ukazuje się komunikat:

ZAPIS PROGRAMU ZRODLOWEGO
<ST> - ZALOZENIE BIBLIOTEKI PROGRAMOWEJ
<S> - ZAPIS KOLEJNEGO PROGRAMU

W przypadku uprzedniego wyboru taśmy kasetowej naciśnięcie klawisza (ST) spowoduje odpowiednie zametrykowanie taśmy /równoważne założeniu biblioteki - patrz paragraf I.10./ i zapis na taśmie, jako pierwszego, programu źródłowego znajdującego się w danym momencie w PAO. Naciśnięcie klawisza (S) natomiast może mieć miejsce tylko wtedy, jeżeli na taśmie znajduje się przynajmniej jeden program i spowoduje zapisanie na taśmie, jako kolejnego, programu źródłowego znajdującego się w PAO.

W obydwu przypadkach, po prawidłowym zapisie programu na taśmie, na ekranie monitora ukazuje się lista funkcji edytora.

W przypadku uprzedniego wyboru pamięci dyskowej naciśnięcie klawisza (ST) spowoduje natomiast ukazanie się kolejnego komunikatu na ekranie:

CZY INICJOWAC DYSK <T,N> :

Odpowiedź "T" spowoduje inicjację dysku elastycznego /patrz paragraf I.11./ oraz zapisanie w bibliotece, jako pierwszego, programu źródłowego znajdującego się w danym momencie w PAO. Równocześnie na ekranie monitora zostanie wyświetlony katalog danej biblioteki - czyli lista programów znajdujących się wniej w tym momencie.

Na czas dokonywania inicjacji dysku programowego /znajdującego się w kieszeni nr Ø/, do kieszeni nr 1 należy koniecznie załadować jakikolwiek dysk programowy. Z dysku tego system musi pobrać loader dyskowy, który /jak o tym była mowa/ zostanie zapisany na ścieżce nr ØØØ inicjowanego dysku.

Odpowiedź "N" spowoduje jedynie wyświetlenie na ekranie listy programów zapisanych w bibliotece na dysku znajdującym się w kieszeni nr Ø.

Lista programów /czyli zawartość katalogu biblioteki/ wyświetlana jest na ekranie w trzech kolumnach. Każda pozycja listy zawiera nazwę programu oraz adres dyskowy, od którego poczynając dany program został zapisany na dysku. Na pierwszym miejscu listy znajduje się zawsze program

o nazwie ØØ. Jest to program, który był jako ostatni skompilowany przy pomocy programu 51. Na końcu listy znajduje się zawsze adres pierwszego wolnego sektora na dysku.

Zawartość katalogu programów wyświetlana jest zawsze w sposób ciągły. Jeżeli lista przekracza pojemność ekranu, powoduje to jego przesunięcie, na skutek czego znikają początkowe pozycje listy. Aby temu zapobiec, w każdym momencie możemy zatrzymać wyświetlanie klawiszem (A) . Naciśnięcie następnie klawisza (B) powoduje wznowienie wyświetlania. Operację tę można powtarzać wielokrotnie.

Naciśnięcie klawisza (S) może mieć miejsce tylko wtedy, gdy dysk został wcześniej zainicjowany oraz znajduje się na nim conajmniej jeden program i spowoduje:

- zapisanie na dysku jako kolejnego - programu znajdującego się w PAO,
- wylistowanie na ekranie aktualnego spisu programów znajdujących się w bibliotece na danym dysku.

We wszystkich trzech powyższych przypadkach, związanych z wyświetlaniem katalogu, lista programów /o ile została wyświetlona do końca/ będzie znajdowała się na ekranie tak długo, dopóki jednym z klawiszy (A) , (B) lub (C) nie spowodujemy powrotu systemu do listy funkcji edytora.

II.4. KOREKTA PROGRAMU

Funkcja 4 umożliwia nam wprowadzanie dowolnych zmian i poprawek do programów źródłowych znajdujących się w PAO. Korekty można dokonywać bezpośrednio po wprowadzeniu programu z klawiatury, po wczytaniu programu z biblioteki do PAO oraz po uprzednio zakończonej korekcie. Nie należy dokonywać korekty programu, któryzostał już poddany kompilacji.

Po uruchomieniu funkcji korekty na ekranie ukazuje się nazwa znajdującego się w PAO programu i treść pierwszej instrukcji. W każdym przypadku, po ukazaniu się na ekranie treści kolejnej instrukcji, edytor umożliwia nam wykonywa-

nie następujących działań, przy pomocy podanych poniżej klawiszy:

Ⓐ - pozostawienie bez zmiany wyświetlonej instrukcji i przejście do instrukcji następnej w programie, związane z wyświetleniem jej treści,

Ⓒ - usunięcie z programu wyświetlonej instrukcji związane również z usunięciem jej treści z ekranu monitora oraz przejście do następnej instrukcji w programie, związane z wyświetleniem jej treści,

Ⓑ - wstawienie do programu nowej instrukcji, bezpośrednio za instrukcją ostatnio wyświetloną. Po naciśnięciu klawisza Ⓑ należy wprowadzić z klawiatury treść nowej instrukcji. Obowiązują w tym przypadku dokładnie te same zasady, jakie podano przy opisie funkcji 1 - wprowadzanie programu z klawiatury.

Jeżeli zachodzi konieczność wstawienia do programu, w danym miejscu, większej ilości dodatkowych instrukcji - wstawienie każdej z nich należy poprzedzać naciśnięciem klawiszą Ⓑ . Po wprowadzeniu ostatniej dopisywanej instrukcji /również jeśli tylko jednej/ należy nacisnąć klawisz ◈ , co spowoduje zakończenie czynności dopisywania i przejście do następnej instrukcji programu związane z wyświetleniem jej treści na monitorze.

Ⓛ - pozostawienie bez zmiany fragmentu programu poczynając od instrukcji ostatnio wyświetlonej aż do wskazanej etykiety.

Po naciśnięciu klawisza Ⓛ ukazuje się na monitorze komunikat "DL?" po którym należy wprowadzić trzycyfrowy numer interesującej nas etykiety i zaakceptować go klawiszem Ⓒ . Korektę programu można kontynuować poczynając od dyrektywy deklarującej etykietę o wskazanym numerze.

⊛ - pozostawienie bez zmiany końcowego fragmentu programu, poczynając od instrukcji ostatnio wyświetlonej

aż do dyrektywy dz. Jest to równoważne zakończeniu funkcji korekty. Dlatego też naciśnięcie klawisza (*) jest konieczne nawet wtedy, jeśli w trakcie wykonywania korekty doszliśmy aż do końca programu, czyli na ekranie została wyświetlona dyrektywa dz.

Funkcja korekty umożliwia ponadto zmianę nazwy poprawionego programu. Mianowicie po zakończeniu korekty klawiszem (*) na ekranie ukazuje się komunikat:

NAZWA PROGRAMU

po którym należy wprowadzić z klawiatury dowolną nazwę dla nowej, poprawionej wersji programu. Po zaakceptowaniu tej nazwy klawiszem (A) edytor wraca do stanu wyjściowego, a na ekranie ukazuje się jego lista funkcji.

Dzięki możliwości zmiany nazwy, poprawiona wersja programu, pod nową nazwą, może być przechowywana w tej samej bibliotece programów co poprzednia wersja programu.

II.5. LISTOWANIE PROGRAMU

Funkcja 5 - listowanie programu - umożliwia wyświetlenie na monitorze źródłowej treści programu znajdującego się w danym momencie w PAO /po wprowadzeniu z klawiatury lub wczytaniu z biblioteki/. Po uruchomieniu funkcji na monitorze ukazuje się tekst:

LISTING PROGRAMU
NN

gdzie NN - nazwa programu znajdującego się w PAO. Następnie każde naciśnięcie klawisza (A) powoduje wyświetlenie kolejnych 20 instrukcji programu, każdej w oddzielnym wierszu.

Przez naciśnięcie klawisza (B) możemy w dowolnym momencie przerwać czynność listowania i spowodować powrót do listy funkcji edytora.

II.6. ODCZYT SEKTORA

Funkcja 6 umożliwia wyświetlenie na ekranie monitora zawartości dowolnego obszaru dysku elastycznego. Po uruchomieniu funkcji na ekranie ukazuje się tekst:

ZALOZ DYSK

Należy wówczas załadować interesujący nas dysk elastyczny do kieszeni nr Ø a następnie nacisnąć klawisz (A) . Ukaże się tekst:

NUMER SCIEZKI

Po wprowadzeniu trzycyfrowego numeru ścieżki i zaakceptowaniu go klawiszem (C) ukaże się kolejny tekst:

NUMER SEKTORA

Należy wówczas wprowadzić trzycyfrowy numer sektora na danej ścieżce, którego zawartość nas interesuje i również zaakceptować go klawiszem (C) . W tym momencie na ekranie zostanie wyświetlona, po 20 bajtów w wierszu, zawartość wszystkich 128 bajtów sektora, którego adres określiliśmy jak powyżej. Zawartość wyświetlana jest w postaci trzycyfrowych dziesiętnych odpowiedników kodów binarnych zawartych w kolejnych bajtach.

Bezpośrednio po tym na ekranie wyświetlony zostanie komunikat:

SEKTOR ODCZYTANY NA ADRES Ø63 ØØØ

oraz powtórzony zostanie adres sektora, którego zawartość znajduje się na ekranie. Powyższy komunikat oznacza, że zawartość tego sektora została równocześnie zapisana w pamięci operacyjnej, poczynając od początku strony Ø63.

Poczynając od tego momentu, każde naciśnięcie klawisza (C) będzie powodowało wyświetlenie w analogiczny sposób zawartości kolejnych sektorów dysku elastycznego, położonych bezpośrednio za sektorem, którego adres określiliśmy na wstępie.

Naciśnięcie klawisza funkcyjnego (A) lub (B) spowoduje natomiast zakończenie funkcji odczytu dysku i wyświetlenie listy funkcji edytora.

II.7. ZAPIS SEKTORA

Funkcja 7 umożliwia wpisanie dowolnej informacji do wybranego sektora lub szeregu sektorów dysku elastycznego.

Po uruchomieniu funkcji na ekranie ukazują się komunikaty dotyczące założenia dysku /do kieszeni nr Ø/ oraz określenia adresu interesującego nas sektora analogicznie jak w przypadku funkcji 6 /patrz poprzedni paragraf/. Po wykonaniu identycznych czynności na ekranie ukazuje się tekst:

ZAPIS SEKTORA
<L> - ZAWARTOSCIA STRONY Ø63
<A> - INFORMACJA Z KLAWIATURY

Po naciśnięciu klawisza (L) na monitorze ukazuje się komunikat:

AKCEPT?

Naciskając w tym momencie jeden z trzech wymienionych klawiszy możemy spowodować wykonanie następujących działań:

(A) - wpisanie do wskazanego sektora pierwszych 128 bajtów strony Ø63, czyli zawartości obszaru pamięci od adresu Ø63.ØØØ do Ø63.127 oraz powrót do listy funkcji edytora,

(B) - powrót do listy funkcji edytora bez dokonania zapisu na dysku,

(C) - wpisanie do wskazanego sektora zawartości pierwszych 128 bajtów strony Ø63 oraz umożliwienie dokonania zapisu w następnym sektorze dysku, poprzez powrót do wyświetlenia komunikatu "ZAPIS SEKTORA".

Po naciśnięciu klawisza (A) należy przede wszystkim wprowadzić z klawiatury informację, którą chcemy zapisać na dysk. Kolejne znaki wprowadzane z klawiatury będą umieszczane w pierwszej kolejności w PAO poczynając od początku strony Ø63. Niezależnie od długości wprowadzonej z klawiatury in-

formacji również i w tym przypadku zapisane zostanie na dysk pierwsze 128 bajtów strony Ø63, odpowiadające pojemności jednego sektora.

Informację wprowadzoną z klawiatury należy zakończyć klawiszem (A), po czym ukaże się komunikat:

AKCEPT?

W tym momencie mamy do wyboru identyczne trzy możliwości jak w przypadku klawisza (L).

II.8. LISTOWANIE PAO

Funkcja 8 umożliwia wyświetlenie na ekranie monitora lub wydrukowanie na drukarce zawartości dowolnego /poza PROM-em/ obszaru pamięci operacyjnej.

Po uruchomieniu funkcji na ekranie ukazuje się tekst:

NUMER STRONY

Wprowadzamy trzycyfrowy numer strony akceptując go klawiszem (C) co spowoduje ukazanie się kolejnego tekstu:

NUMER KOMORKI

Wprowadzamy trzycyfrowy numer komórki /ØØØ ÷ 255/ i również akceptujemy go klawiszem (C).

Po określeniu w powyższy sposób adresu początkowego interesującego nas obszaru PAO ukazuje się z kolei następujący komunikat:

WYBIERZ URZADZENIE ZEWNETRZNE
M - MONITOR
D - DRUKARKA

Naciskając klawisz M lub D klawiatury alfanumerycznej /i następnie akceptując go klawiszem (A) / wybieramy odpowiednio urządzenie, na którym będzie listowana zawartość wybranego obszaru PAO.

W przypadku wyboru monitora każde naciśnięcie klawisza (A) powoduje wyświetlenie zawartości kolejnych 20 bajtów w jednym wierszu ekranu w postaci trzycyfrowych odpowiedni-

ków dziesiętnych kodów binarnych zawartych w PAO. Naciśnięcie klawisza (B) powoduje zakończenie wyświetlania i powrót do systemowej listy funkcji edytora.

Natomiast w przypadku wyboru drukarki zawartość pamięci traktowana jest przez system jako program w języku maszynowym MERA-100 i drukowana w takiej postaci na tabulogramie, każda instrukcja w oddzielnej linii. Kody nie odpowiadające żadnej instrukcji języka maszynowego MERA-100 drukowane są samodzielnie w oddzielnych wierszach.

Funkcja 8 w tym przypadku umożliwia nam zapoznanie się z treścią skompilowanych programów w języku symbolicznym KD-100 jak również z treścią dowolnych programów napisanych w języku maszynowym MERA-100, wczytanych uprzednio do PAO przy pomocy funkcji 2 - czytanie programu.

Drukowanie zostanie zakończone w przypadku osiągnięcia końca programu lub po naciśnięciu klawisza funkcyjnego (L). W obydwu przypadkach następuje wyświetlenie listy funkcji edytora.

II.9. KOREKTA PAO

Funkcja 9 - korekta PAO - umożliwia wprowadzania do dowolnych /poza PROM-em/ komórek pamięci dowolnej informacji wprowadzonej z klawiatury w postaci trzycyfrowego odpowiednika dziesiętnego żądanego kodu binarnego.

Po uruchomieniu funkcji należy w pierwszej kolejności określić adres interesującej nas komórki PAO identycznie jak w przypadku funkcji 8 - listowanie PAO /patrz poprzedni paragraf/. Po zakończeniu powyższego ukazuje się na ekranie komunikat:

KOREKTA

po którym należy wprowadzić trzycyfrową liczbę z zakresu od ØØØ do 255 i zaakceptować ją klawiszem (C). W tym momencie odpowiadający wprowadzonej liczbie kod binarny zostaje wpisany do wybranej komórki PAO i następuje powrót do stanu wyjściowego edytora - wyświetlenie listy funkcji.

III. FUNKCJE REALIZOWANE PRZEZ EGZEKUTOR

Po uruchomieniu egzekutora /co powinno mieć miejsce tylko po uprzednim umieszczeniu w PAO przy pomocy edytora źródłowej treści programu/ na ekranie monitora ukazuje się następująca lista funkcji:

L - LISTOWANIE
K - KOMPILACJA
R - REALIZACJA

Wyboru żądanej funkcji dokonuje się naciskając odpowiedni klawisz dużej litery: L, K lub R klawiatury alfanumerycznej, bez akceptu.

Zakończenie każdej z wymienionych funkcji powoduje automatyczne wyświetlenie listy funkcji egzekutora, czyli jego powrót w stan wyjściowy.

III.1. LISTOWANIE PROGRAMU

Funkcja L - listowanie - powoduje wydrukowanie na drukarce treści programu źródłowego znajdującego się w danym momencie w PAO bezpośrednio po wprowadzeniu z klawiatury, po wczytaniu z biblioteki lub po korekcie przy pomocy edytora.

Po uruchomieniu funkcji na tabulogramie ukazuje się komunikat "P=NN" /gdzie NN - nazwa listowanego programu/ a następnie kolejno instrukcje programu, każda w oddzielnej linii. Po osiągnięciu końca programu system wraca w stan wyjściowy egzekutora.

III.2. KOMPILACJA PROGRAMU

Funkcja K - kompilacja - powoduje wykonanie kompilacji programu źródłowego znajdującego się w danym momencie w PAO. Efektem kompilacji jest utworzenie w pamięci operacyjnej postaci wynikowej programu, nadającej się już bezpośrednio do realizacji.

Przed rozpoczęciem kompilacji egzekutor drukuje na tabulogramie nazwę programu, a po zakończeniu - dwa adresy w postaci: numer strony.numer komórki :

- pierwszy z nich jest adresem końca programu wynikowego /pierwszej wolnej komórki/,
- drugi - w analogiczny sposób - adresem końca programu źródłowego.

Ponadto w czasie kompilacji mogą być sygnalizowane błędne instrukcje w przypadkach: nieistniejących kodów operacji lub powtórzonych numerów etykiet. Sygnalizacja takich błędów ma postać: "* B - treść błędnej instrukcji".

Po zakończeniu kompilacji egzekutor wraca do stanu wyjściowego.

III.3. REALIZACJA /WYKONANIE/ PROGRAMU

Funkcja R - realizacja - powoduje start i wykonanie kolejnych instrukcji programu bezpośrednio przed tym poddanego kompilacji.

Wykonanie instrukcji "stop" w programie powoduje zakończenie funkcji realizacji i przejście egzekutora w stan wyjściowy.

Naciśnięcie klawisza (L) w czasie realizacji programu powoduje jej przerwanie i powrót egzekutora w stan wyjściowy. Polecenie to nie działa jednak w przypadku wykonywania jakiejkolwiek instrukcji czytania, przy aktywnej na wejściu klawiaturze.

IV. OPIS INSTRUKCJI JĘZYKA SYMBOLICZNEGO KD-100

IV.1. DYREKTYWY

IV.1.1. Deklaracja etykiety

Format: dl En;

Dyrektywa ta nadaje symboliczny adres instrukcji, przed którą występuje w programie. Adres ten nazywamy etykietą. Stanowi ją numer w zakresie ØØ1 ÷ 128 /z wyjątkiem 124/. W trakcie wykonywania programu, podczas przechodzenia przez miejsce oznaczone etykietą, jej numer wyświetla system w odpowiedni sposób na lampkach o nr 1 ÷ 8.

IV.1.2. Deklaracja podprogramu

Format: sk En;

Dyrektywa ta nadaje adres symboliczny /etykietę/ pierwszej instrukcji podprogramu. Jest to jednocześnie adres całego podprogramu, do którego można przejść wyłącznie przy pomocy instrukcji "skocz do podprogramu". Etykiety nadawane dyrektywą sk mogą przybierać te same wartości co w przypadku dyrektywy dl, ale w jednym programie numery etykiet deklarowanych dyrektywami dl i sk nie mogą się powtarzać.

IV.1.3. Fizyczny koniec programu

Format: dz;

Dyrektywa ta jest sygnałem zakończenia kompilacji programu źródłowego, powodując przejście systemu /egzekutora/ w stan wyjściowy. Z tego względu musi występować zawsze jako fizycznie ostatnia instrukcja programu. Kompilator w miejsce dyrektywy dz wpisuje do programu wynikowego instrukcję stopu sh.

IV.2. INSTRUKCJE STERUJĄCE

IV.2.1. Skocz do En

Format: st En;

Instrukcja powoduje bezwarunkowe przekazanie sterowania do instrukcji oznaczonej w programie etykietą o numerze En. Jeżeli etykiety o takim numerze nie zadeklarowano w programie, to w trakcie jego realizacji instrukcja ta spowoduje zasygnalizowanie błędu w postaci "* b" oraz przerwanie pracy programu. Ostatnia uwaga dotyczy wszystkich dalej omówionych instrukcji sterujących.

IV.2.2. Skocz do En gdy WØ

Format: sr En;

Instrukcja powoduje przekazanie sterowania do instrukcji oznaczonej w programie etykietą o numerze En, o ile w danym momencie w rejestrze warunku C wpisany jest warunek WØ / [C] = WØ/. W każdym innym przypadku sterowanie zostanie przekazane do instrukcji następnej w programie po instrukcji sr.

IV.2.3. Skocz do En gdy W1

Format: wi En;

Działanie jak wyżej ale dla [C] = W1.

IV.2.4. Skocz do En gdy W2

Format: ms En;

Działanie jak wyżej ale dla [C] = W2.

IV.2.5. Skocz do En gdy W3

Format: wn En;

Działanie jak wyżej ale dla [C] = W3.

IV.2.6. Skocz do podprogramu En

Format: wo En;

Instrukcja powoduje przekazanie sterowania do pierwszej instrukcji podprogramu, oznaczonego etykietą En przy pomocy

dyrektywy sk. Przekazanie to związane jest z zapamiętaniem adresu instrukcji następnej po instrukcji wo.

Instrukcja wo może również występować w podprogramie, co oznacza, że w języku symbolicznym systemu KD-100 możliwy jest wielostopniowy skok z podprogramu do następnego podprogramu.

IV.2.7. Wróć z podprogramu En

Format: wa Er;

Instrukcja wa występuje zawsze jako ostatnia instrukcja podprogramu i powoduje przekazanie sterowania do instrukcji następnej po instrukcji, która spowodowała przejście do tego podprogramu.

IV.2.8. Skocz do etykiety, której numer wskazuje zawartość Rm

Format: sq Rm;

Instrukcja powoduje bezwarunkowe przekazanie sterowania do instrukcji oznaczonej w programie etykietą, której numer określają trzy najmniej znaczące cyfry /modulo 128/ liczby zapisanej w rejestrze roboczym Rm.

IV.2.9. Stop /skocz do początku egzekutora/

Format: sh;

Instrukcja powoduje zakończenie realizacji programu /stop/, co równoważne jest przekazaniu sterowania na początek egzekutora, czyli przejściu jego w stan wyjściowy, sygnalizowany wyświetleniem na ekranie listy funkcji. Jeżeli program współpracował z pamięcią kasetową lub pamięcią na dyskach elastycznych instrukcja sh powoduje zwolnienie obydwu pamięci zewnętrznych, a w stosunku do pamięci kasetowej - przewinięcie ponadto taśmy do fizycznego początku.

IV.3. INSTRUKCJE ARYTMETYCZNE

IV.3.1. Dodaj [R] do [Rm]

Format: rd Rm Rn;

Funkcja: [Rm] := [Rm] + [Rn]

[C] := W0 ∨ W1 ∨ W2 ∨ W3

Instrukcja dodaje liczbę znajdującą się w rejestrze Rn do liczby znajdującej się w rejestrze Rm. Wynik dodawania umieszczany jest w rejestrze Rm, zaś liczba z rejestru Rn pozostaje niezmieniona. Po wykonaniu dodawania w rejestrze warunku C ustawiany jest:

W0 - gdy wynik jest równy zeru,
W1 - gdy wynik jest większy od zera,
W2 - gdy wynik jest mniejszy od zera,
W3 - gdy wystąpił nadmiar

Jeżeli w wyniku dodawania wystąpił nadmiar /wynik jest liczbą większą niż 14-cyfrowa/ to wynik nie jest przenoszony do rejestru Rm. Oznacza to, że liczby w rejestrach Rm i Rn pozostają niezmienione. W takim przypadku efektem instrukcji jest jedynie ustawienie warunku W3 w rejestrze C.

Przykład: w wymienionych poniżej rejestrach zawarte są następujące liczby:

R000 - liczba a
R001 - liczba b
R002 - liczba c
R003 - liczba 0

Należy obliczyć wartość wyrażenia x = a + b + c a wynik zapamiętać w rejestrze R003, zachowując niezmienioną zawartość rejestrów R000 ÷ R002. Zakładamy, że nadmiar nie wystąpi:

```
rd 003 000;
rd 003 001;
rd 003 002;
```

IV.3.2. Odejmij [Rn] od [Rm]

Format: ro Rm Rn;

Funkcja: [Rm] := [Rm] - [Rn]

[C] := W0 ∨ W1 ∨ W2 ∨ W3

Instrukcja odejmuje liczbę znajdującą się w rejestrze Rn od liczby znajdującej się w rejestrze Rm. Wynik umieszczany jest w rejestrze Rm, zaś liczba z rejestru Rn nie ulega zmianie. W przypadku wystąpienia nadmiaru zawartości Rm i Rn pozostają niezmienione. Instrukcja ustawia rejestr C w sposób identyczny jak podany przy opisie instrukcji dodawania.

Przykład: przyjmując założenia z przykładu podanego w opisie instrukcji dodawania należy obliczyć wartość wyrażenia x = a - b - c :

```
rd 003 000;
ro 003 001;
ro 003 002;
```

IV.3.3. Podaj parametry dla dzielenia lub mnożenia

Format: rx a b;

gdzie: a - ilość cyfr obcinanych w wyniku
b - symbol służący do zaokrąglania wyniku /0 lub 5/.

Funkcja: zapamiętanie wymienionych wyżej parametrów a i b, wykorzystywanych przy kształtowaniu wyników mnożenia i dzielenia.

Instrukcja rx musi bezpośrednio poprzedzać każdą instrukcję dzielenia lub mnożenia.

Parametr a może przyjmować wartości od 0 ÷ 9. Jeżeli a = 0 to w wyniku zachowane zostaną wszystkie cyfry otrzymanego iloczynu lub ilorazu. W takim przypadku parametr b nie ma wogóle żadnego znaczenia /zaokrąglenie wyniku nie może mieć miejsca/. Jeżeli natomiast a ≠ 0, to z otrzymanego iloczynu lub ilorazu obciętych zostanie a najmniej znaczących cyfr, zaś do pierwszej z pomijanych cyfr dodany zostanie, przed obcięciem, parametr b. Jeśli więc b = 0, to zachowany po obcięciu wynik nie będzie zaokrąglony, zaś gdy b = 5 - zaokrąglenie nastąpi.

Przykład: Jeżeli w wyniku dzielenia lub mnożenia mają być obcięte dwie ostatnie cyfry a pozostała część zaokrąglona, to instrukcja rx poprzedzająca to mnożenie lub dzielenie musi mieć następującą postać:

```
rx 002 005;
```

IV.3.4. Pomnóż [Rm] przez [Rn]

Format: rm Rm Rn;

Funkcja: [Rm] := [Rm] • [Rn]

[C] := W0∨W1∨W2∨W3

Instrukcja mnoży liczbę znajdującą się w rejestrze Rm przez liczbę znajdującą się w rejestrze Rn. Wynik zostaje umieszczony w rejestrze Rm, zaś liczba znajdująca się w rejestrze Rn /mnożnik/ pozostaje niezmieniona. Jeżeli w trakcie wykonywania mnożenia wystąpi nadmiar, zawartości Rm i Rn pozostają niezmienione. Efektem działania instrukcji jest w tym przypadku jedynie wpisanie warunku W3 do rejestru C. Instrukcja ustawia zawartość rejestru C w sposób identyczny jak podany przy opisie instrukcji dodawania.

Przed wpisaniem wyniku do rejestru Rm otrzymany iloczyn może zostać skrócony o najmniej znaczące cyfry, z zaokrągleniem lub bez, w zależności od parametrów podanych przy pomocy instrukcji rx, poprzedzającej instrukcję mnożenia. Wystąpienie nadmiaru sygnalizowane jest w tym przypadku, jeśli wystąpi on już po obcięciu wskazanej ilości cyfr iloczynu.

Przykład: W rejestrze R001 znajduje się liczba a z dwoma /umownymi/ znakami po przecinku, zaś w rejestrze R002 - liczba b z jednym /też umownym/ znakiem po przecinku. W rejestrze R001 chcemy otrzymać iloczyn a • b z dwoma znakami po przecinku, zaokrąglony. Zadanie to zrealizuje następująca sekwencja instrukcji:

```
rx 001 005;
rm 001 002;
```

IV.3.5. Podziel [Rm] przez [Rn]

Format: ri Rm Rn;

Funkcja: [Rm] := [Rm] : [Rn]

[C] := W0∨W1∨W2∨W3

Instrukcja dzieli liczbę znajdującą się w rejestrze Rm przez liczbę znajdującą się w rejestrze Rn. Wynik zostaje umieszczony w rejestrze Rm, zaś liczba znajdująca się w rejestrze Rn /dzielnik/ pozostaje niezmieniona. Jeżeli w tra-

kcie wykonywania dzielenia wystąpi nadmiar /może to mieć miejsce w przypadku dzielenia przez zero/, zawartości rejestrów Rm i Rn pozostają niezmienione. Efektem działania instrukcji jest w tym przypadku jedynie wpisanie warunku W3 do rejestru C. Instrukcja ustawia zawartość rejestru C w sposób identyczny jak podany przy opisie instrukcji dodawania.

Przed wpisaniem wyniku do rejestru Rm iloraz może zostać skrócony o najmniej znaczące cyfry, z zaokrągleniem lub bez, w zależności od parametrów podanych przy pomocy instrukcji rx, poprzedzającej instrukcję dzielenia.

Jak wiadomo liczby przechowywane w rejestrach roboczych traktowane są przez system KD-100 jako liczby całkowite. W związku z tym wynik, otrzymywany w efekcie realizacji instrukcji dzielenia, stanowi tylko część całkowitą ilorazu. Większą dokładność możemy otrzymać drogą mnożenia dzielnej przez odpowiednią potęgę dziesięciu /np. przy pomocy instrukcji rl/ przed wykonaniem instrukcji dzielenia. Jeżeli np. przed podzieleniem liczby zawartej w R001 wykonamy instrukcję:

rl 001 002;

w wyniku której liczba w rejestrze R001 została pomnożona przez 10^2, to wynik dzielenia tej liczby utworzony w rejestrze R001 będzie zawierał /umownie/ dwa znaki po przecinku.

Przykład: W rejestrze R001 znajduje się liczba a określająca wykonanie planu, zaś w rejestrze R002 liczba b określająca plan na dany okres. Obie wielkości są liczbami całkowitymi. Należy obliczyć procent wykonania planu p z dwoma znakami po przecinku i zaokrąglić go do jednego znaku po przecinku. Jak łatwo stwierdzić, wielkość p należy wyliczyć wg. następującego wzoru:

$$p = \frac{a \cdot 10^2 \cdot 10^2}{b}$$

Zadanie to zrealizuje następująca sekwencja instrukcji:

rl 001 004;
rx 001 005;
ri 001 002;

IV.3.6. Porównaj liczby zawarte w Rm i Rn

Format: pw Rm Rn;

Funkcja: [C] := W0 ∨ W1 ∨ W2

Instrukcja porównuje liczbę znajdującą się w rejestrze Rm z liczbą znajdującą się w rejestrze Rn. Wynik tego porównania sygnalizowany jest ustawieniem odpowiedniego warunku w rejestrze C:

W0 - gdy liczba w Rm równa jest liczbie w Rn

W1 - gdy liczba w Rm jest większa od liczby w Rn

W2 - gdy liczba w Rm jest mniejsza od liczby w Rn

Liczby zawarte w rejestrach Rm iRn pozostają niezmienione.

IV.3.7. Porównaj liczbę zawartą w Rm z zerem

Format: cn Rm;

Funkcja: [C] := W0 ∨ W1 ∨ W2

Instrukcja porównuje liczbę znajdującą się w rejestrze Rm z zerem. Wynik tego porównania sygnalizowany jest ustawieniem odpowiedniego warunku w rejestrze C:

W0 - gdy liczba w Rm jest równa zeru

W1 - gdy liczba w Rm jest większa od zera

W2 - gdy liczba w Rm jest mniejsza od zera

Liczba zawarta w rejestrze Rm pozostaje niezmieniona.

IV.3.8. Przesuń liczbę w Rm w prawo o d pozycji

Format: rp Rm d;

Funkcja: $[Rm] := [Rm] \cdot 10^{-d}$

gdzie d - ilość pozycji cyfrowych rejestru roboczego /1 ÷ 14/

Instrukcja powoduje arytmetyczne przesunięcie liczby znajdującej się w rejestrze Rm o d pozycji cyfrowych w prawo. Cyfry z prawej strony nie mieszczące się w rejestrze są gubione, zwalniane pozycje cyfrowe z lewej strony zapełniane są zerami. Znak liczby nie bierze udziału w przesunięciu. Działanie instrukcji równoważne jest dzieleniu zawartości rejestru przez 10^d /z dokładnością do części całkowitej ilorazu/.

Przykład: Jeżeli w rejestrze R001 znajduje się liczba:
100000000012345
to po wykonaniu instrukcji rp 001 002; otrzymamy:
100000000000123

IV.3.9. Przesuń liczbę w Rm w lewo o d pozycji
Format: rl Rm d;
Funkcja: $[Rm] := [Rm] \cdot 10^d$
gdzie d - ilość pozycji cyfrowych rejestru roboczego /1 ÷ 14/

Instrukcja powoduje arytmetyczne przesunięcie liczby znajdującej się w rejestrze Rm o d pozycji cyfrowych w lewo. Cyfry z lewej strony nie mieszczące się w rejestrze są gubione, zwalniane pozycje cyfrowe z prawej zapełniane są zerami. Znak liczby nie bierze udziału w przesunięciu. Działanie instrukcji równoważne jest mnożeniu zawartości rejestru przez 10^d.

Przykład: Jeżeli w rejestrze R001 znajduje się liczba:
101234560001234
to po wykonaniu instrukcji rl 001 005; otrzymamy:
156000123400000

IV.4. PRZESŁANIA

IV.4.1. Prześlij [Rn] do Rm

Format: rz Rm Rn;

Funkcja: [Rm] := [Rn]

Instrukcja powoduje przesłanie zawartości rejestru Rn do rejestru Rm. Stosuje się zarówno do liczb jak i tekstów. Po wykonaniu instrukcji zawartości rejestrów Rm i Rn są identyczne.

Przykład: Jeżeli zawartości wymienionych rejestrów są następujące:

R005: 100000000012345

R101: SYSTEM␣KD␣-␣100

to po wykonaniu instrukcji rz 005 101; otrzymamy:

R005: SYSTEM␣KD␣-␣100

R101: SYSTEM␣KD␣-␣100

IV.4.2. Prześlij [Rn] do R(Rm)

Format: ai Rm Rn;

Funkcja: [R(Rm)] := [Rn]

Instrukcja powoduje przesłanie zawartości rejestru Rn do rejestru, którego numer wskazuje liczba zawarta w rejestrze Rm. Pod uwagę brane są tylko trzy ostatnie pozycje cyfrowe rejestru Rm niezależnie od znaku liczby.

Przykład: Jeżeli zawartości wymienionych rejestrów są następujące:

R010: WARSZAWA␣␣␣␣␣␣␣

R050: 100000012345010

R100: 000000000012345

to po wykonaniu instrukcji ai 050 100; otrzymamy:

R010: 000000000012345

R050: 100000012345010

R100: 000000000012345

IV.4.3. Prześlij [R(Rn)] do Rm

Format: am Rm Rn;

Funkcja: [Rm] := [R(Rn)]

Instrukcja powoduje przesłanie zawartości rejestru, którego numer wskazuje liczba zawarta w rejestrze Rn, do reje-

stru Rm. Pod uwagę brane są tylko trzy ostatnie pozycje cyfrowe rejestru Rn niezależnie od znaku liczby.

Przykład: Jeżeli przyjmiemy założenia z poprzedniego przykładu, to po wykonaniu instrukcji am 100 050; otrzymamy:

R010: WARSZAWA▽▽▽▽▽▽▽
R050: 100000012345010
R100: WARSZAWA▽▽▽▽▽▽▽

IV.4.4. Prześlij [R(Rn)] do R(Rm)

Format: im Rm Rn;

Funkcja: [R(Rm)] := [R(Rn)]

Instrukcja powoduje przesłanie zawartości rejestru, którego numer wskazuje liczba zawarta w rejestrze Rn, do rejestru, którego numer wskazuje liczba zawarta w rejestrze Rm. Pod uwagę brane są tylko trzy ostatnie pozycje cyfrowe rejestrów Rm i Rn niezależnie od znaków algebraicznych liczb.

Przykład: Jeżeli zawartści wymienionych rejestrów są następujące:

R000: REJESTR▽ROBOCZY
R010: 100000000001200
R100: 000000000000000
R200: 000012345678912

to po wykonaniu instrukcji im 100 010; otrzymamy:

R000: 000012345678912
R010: 100000000001200
R100: 000000000000000
R200: 000012345678912

IV.4.5. Umieść liczbę w Rm

Format: lr Rm/±99.....9/;

Funkcja: [Rm] := ±99.....9

Instrukcja powoduje ustawienie w rejestrze Rm liczby podanej jako argument instrukcji. Liczba musi być koniecznie poprzedzona znakiem algebraicznym /plusem lub minusem/, składać się z maksymalnie czternastu cyfr, nie może zawierać ani kropki ani przecinka dziesiętnego. Jeżeli ładujemy do rejestru liczbę krótszą niż czternastocyfrowa, to wolne pozycje z lewej strony rejestru zostaną wypełnione zerami.

Przykład: Jeżeli zawartość rejestru jest następująca:

R105: TABLICA␣ROBOCZA

to po wykonaniu instrukcji lr 105/-12345/; otrzymamy:

R105: 100000000012345

IV.4.6. Umieść tekst w Rm

Format: ts Rm/ttt....t/;

Funkcja: [Rm] := ttt....t

Instrukcja powoduje ustawienie w rejestrze Rm tekstu podanego jako argument instrukcji. Tekst może składać się z maksymalnie 15 dowolnych znaków z repertuaru klawiatury alfanumerycznej minikomputera, z wyjątkiem jedynie średnika i slasha (; i /). Umieszczony w rejestrze tekst jest dosuwany do lewego końca rejestru. Jeżeli liczy on mniej niż 15 znaków, to wolne pozycje z prawej strony rejestru zapełniane są znakami spacji.

Przykład: Jeżeli zawartość rejestru jest następująca:

R120: 100000000012345

to po wykonaniu instrukcji ts 120/MERA-100/; otrzymamy:

R120: MERA-100␣␣␣␣␣␣␣

IV.4.7. Zapełnij Rm znakiem Z

Format: az Rm Z;

Funkcja: [Rm] := ZZZZZZZZZZZZZZZ

Instrukcja powoduje wpisanie na wszystkie 15 miejsc znakowych rejestru Rm znaku, którego wartość w kodzie ISO-7 podano jako argument instrukcji.

IV.5. INSTRUKCJE TEKSTOWE

IV.5.1. Porównaj teksty zawarte w Rm i Rn

Format: qq Rm Rn;

Funkcja: [C] := W0 ∨ W1

Instrukcja porównuje tekst znajdujący się w rejestrze Rm z tekstem znajdującym się w rejestrze Rn. Wynik tego porównania sygnalizowany jest ustawieniem odpowiedniego warunku w rejestrze C:

W0 - gdy teksty są identyczne

W1 - gdy teksty są różne

Zawartości rejestrów Rm i Rn pozostają niezmienione.

IV.5.2. Porównaj pierwszy znak z Rm ze znakiem Z

Format: al Rm Z;

Funkcja: [C] := W0 ∨ W1

Instrukcja porównuje pierwszy znak rejestru Rm /pozycja nr 0/ ze znakiem, którego wartość dziesiętna w kodzie ISO-7 podana jest jako argument instrukcji. Wynik tego porównania sygnalizowany jest ustawieniem odpowiedniego warunku w rejestrze C:

W0 - gdy znaki są identyczne

W1 - gdy znaki są różne

Zawartość rejestru Rm pozostaje niezmieniona.

IV.6. INSTRUKCJE WEJŚCIA/WYJŚCIA

IV.6.1. Deklaracja urządzenia wejścia

Format: we S;

W systemie KD-100 każda z instrukcji wejścia języka symbolicznego realizowana jest poprzez urządzenie wejścia w danym momencie aktywne, który to stan urządzenia deklaruje się przy pomocy instrukcji we. Argument S określa rodzaj urządzenia wejścia i może przybierać wartości:

1 - klawiatura z wyświetlaniem na monitorze
4 - klawiatura z wydrukiem na drukarce
5 - pamięć kasetowa
6 - pamięć dyskowa

Jak z powyższego wynika, klawiaturę można w systemie KD-100 aktywizować jako urządzenie wejścia w dwojaki sposób:

- zaktywizowana symbolem 1 powoduje wprowadzanie kodów znaków do bufora wejścia z równoczesnym wyświetlaniem tych znaków na ekranie monitora,
- zaktywizowana symbolem 4 powoduje wprowadzanie kodów znaków do bufora wejścia z równoczesnym drukowaniem tych znaków na drukarce.

Wszystkie instrukcje czytania języka symbolicznego /qy, qt, cj/ pobierają kolejne znaki z bufora wejścia /str. Ø62 PA0/. Bufor ten jest zawsze kasowany w momencie aktywizacji urządzenia wejścia instrukcją we. Jeżeli instrukcja czytania stwierdza, że bufor wejściowy jest pusty, to w zależności od rodzaju zadeklarowanego urządzenia wejścia następuje:

klawiatura - udostępnienie klawiatury operatorowi,
pamięć kasetowa - automatyczne odczytanie kolejnego bloku z taśmy kasetowej i umieszczenie go w buforze wejścia,
pamięć dyskowa - udostępnienie klawiatury operatorowi.

Pierwsza w danym programie aktywizacja pamięci kasetowej jako urządzenia wejścia /w tym momencie taśma kasetowa

powinna znajdować się na rozbiegówce/ powoduje zarezerwowanie i spozycjonowanie do odczytu taśmy kasetowej. Kolejne w programie instrukcje aktywizacji taśmy jako urządzenia wejścia nie będą miały żadnego wpływu na stan pamięci kasetowej /o ile taśma nie zostanie w międzyczasie przewinięta do fizycznego początku/.

IV.6.2. Deklaracja urządzenia wyjścia

Format: wy S;

Każda z instrukcji wyjścia języka symbolicznego realizowana jest poprzez urządzenie wyjścia w danym momencie aktywne, który to stan urządzenia deklaruje się przy pomocy instrukcji wy. Argument S określa rodzaj urządzenia i może w tym przypadku przybierać następujące wartości:

1 - monitor ekranowy
5 - pamięć kasetowa
6 - pamięć dyskowa
8 - drukarka
9 - pamięć kasetowa + drukarka

Wszystkie instrukcje wyjścia języka symbolicznego /pu, ph, wl, pz, wt, zz, pt/ wysyłają kolejne znaki wyprowadzanej informacji, w zależności od rodzaju zaktywizowanego w danym momencie urządzenia, do:

monitor	- bezpośrednio na monitor
pamięć kasetowa	- do bufora wyjścia /str. Ø63 PAO/
pamięć dyskowa	- do bufora wyjścia
drukarka	- bezpośrednio na drukarkę
pamięć kasetowa + drukarka	- równocześnie do bufora wyjścia i na drukarkę.

Bufor wyjściowy jest kasowany w momencie deklaracji pamięci kasetowej lub dyskowej jako urządzenia wyjścia /wy 5, wy 6 lub wy 9/. Przeniesienie informacji zgromadzonej w buforze wyjścia na taśmę kasetową lub dysk elastyczny wymaga zastosowania przeznaczonej do tego celu instrukcji "pisz blok - pb"; po realizacji której bufor wyjściowy jest również skasowany.

Taśma kasetowa, na którą w danym programie będą wyprowadzane dane, musi posiadać:

- bądź metrykę taśmy /zapisaną instrukcją mt;/, jeżeli nie były na niej zapisywane żadne informacje,
- bądź znacznik końca, jeżeli były już na niej zapisywane informacje.

Pierwsza w danym programie deklaracja pamięci kasetowej jako urządzenia wyjścia /taśma stoi na rozbiegówce lub znaczniku BOT/ powoduje przewinięcie i ustawienie taśmy kasetowej na metryce taśmy lub na znaczniku końca zapisu, przy czym pierwszy blok zapisywanej informacji kasuje wyżej wymienione: metrykę lub znacznik końca. Każda następna w danym programie deklaracja taśmy jako urządzenia wyjścia nie ma żadnego wpływu na stan pamięci kasetowej.

Dysk elastyczny, na który będą wyprowadzane dane, nie wymaga żadnego uprzedniego przygotowania.

IV.6.3. Czytaj liczbę do Rm a ogranicznik do Rn

Format: qy Rm Rn;

Instrukcja powoduje wczytanie liczby z aktywnego urządzenia wejściowego do rejestru Rm, zaś do rejestru Rn /na lewą skrajną pozycję/ ogranicznika liczby, czyli znaku który spowodował zakończenie realizacji czytania liczby. W skład liczby mogą wchodzić cyfry oraz przecinek lub kropka, przy czym te dwa ostatnie znaki są ignorowane wewnątrz liczby. Znak algebraiczny musi występować na końcu liczby, może on pełnić wtedy jednocześnie funkcję ogranicznika. Liczba zakończona innym ogranicznikiem niż znak algebraiczny traktowana jest zawsze jako dodatnia.

Instrukcja qy ustawia w następujący sposób rejestr warunku C:

WØ - gdy ogranicznikiem wczytanej liczby był jeden ze znaków: +, -, ▽, NL.

W2 - gdy ogranicznikiem wczytanej liczby był dowolny inny znak alfanumeryczny, z wyjątkiem: cyfr, przecinka i kropki.

W1 - gdy zamiast liczby instrukcja wczytała znak # . W tym przypadku zawartości rejestrów Rm i Rn nie ulegają zmianie.

W3 - tylko w przypadku aktywnej na wejściu pamięci kasetowej lub dyskowej - gdy przeczytany został znacznik końca zapisu.

Jeżeli zamiast liczby instrukcja qy wczyta znak " * " lub " - " uzupełnione ogranicznikiem - do rejestru Rm wczytana zostanie liczba zero, zaś w rejestrze C ustawiony warunek WØ bez względu na rodzaj ogranicznika.

Jeżeli wczytywana liczba liczy więcej niż 14 cyfr, z bufora zostaną pobrane wszystkie cyfry aż do ogranicznika, zaś do rejestru Rm wstawionych zostanie pierwszych 14 cyfr a reszta zgubiona. Do rejestrów Rn i C zostaną wstawione informacje zgodnie z wyżej opisanymi zasadami.

IV.6.4. Podaj parametry do pisania liczby

Format: ru m n;

gdzie: m - numer pozycji cyfrowej rejestru, począwszy od której wyprowadzona zostanie zawartość rejestru,

n - ilość cyfr po przecinku.

Instrukcja służy do zapamiętania dwóch parametrów, określających postać wyprowadzanej na zewnątrz liczby. Instrukcja ru musi poprzedzać instrukcję drukowania liczby pu oraz instrukcję drukowania tablicy pt. Nie musi być powtarzana przed każdą instrukcją drukowania, o ile format wydruku liczby nie ulega zmianie.

Parametr m, określający numer pozycji cyfrowej rejestru, od której poczynając wyprowadzona zostanie zawartość rejestru, określa długość wypisywanej liczby. Może przyjmować tylko wartości od 1 ÷ 14, zgodnie ze strukturą rejestru roboczego:

±	C	C	C	C	C	C	C	C	C	C	C	C	C	C	Ogr
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Np. przy m = 1 wyprowadzona zostanie cała zawartość liczbowa rejestru, zaś przy m = 14 - tylko ostatnia pozycja cyfrowa.

Parametr n oznacza ilość pozycji cyfrowych licząc od prawej strony rejestru, które mają zostać oddzielone przecinkiem w wypisywanej liczbie. Może przyjmować wartości od Ø ÷ 14. Np. przy n = Ø liczba będzie wyprowadzona jako całkowita, zaś przy n = 14 cała zawartość rejestru zostanie wyprowadzona jako ułamek. Aby wyprowadzanie liczby odbywało się prawidłowo, pomiędzy obydwoma powyższymi parametrami musi zachodzić następująca zależność:

$$15 - m \geq n$$

IV.6.5. Pisz liczbę z Rm

Format: pu Rm Z;

gdzie: Z - wartość dziesiętna znaku w kodzie ISO-7, który będzie wyprowadzony na urządzenie wyjścia, gdy liczba w Rm = Ø.

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia, liczby zawartej w rejestrze Rm. O ilości miejsc wyprowadzanej liczby oraz o obecności i położeniu przecinka w liczbie decydują parametry podane przy pomocy instrukcji ru, poprzedzającej daną instrukcję pisania. W trakcie wyprowadzania liczby zera nieznaczące zamieniane są na spacje /z wyjątkiem zera poprzedzającego przecinek/, następnie wyprowadzane są kolejne cyfry znaczące oraz na końcu znak algebraiczny liczby, tzn.:

spacja - jeżeli liczba jest dodatnia
minus - jeżeli liczba jest ujemna

Uwaga: przecinek w liczbie zajmuje oddzielny znak na wydruku lub ekranie /podobnie w pamięci kasetowej i dyskowej/ toteż liczba wyprowadzana z przecinkiem będzie o jedno miejsce dłuższa od liczby składającej się z takiej samej ilości cyfr, ale wyprowadzanej

bez przecinka. Np. jeżeli wyprowadzamy liczbę składającą się z n cyfr to zajmie ona po wyprowadzeniu:

n + 1 miejsc - gdy będzie wyprowadzana jako całkowita

n + 2 miejsca - gdy będzie wyprowadzana z miejscami po przecinku.

W przypadku, gdy liczba znajdująca się w rejestrze Rm jest równa zeru, instrukcja pu wyprowadza na zewnątrz dowolny znak /z wyjątkiem #/, którego wartość dziesiętna w kodzie ISO-7 podana jest jako drugi argument instrukcji. Znak ten, poprzedzony odpowiednią ilością spacji, wyprowadzany jest w miejscu odpowiadającym ostatniej pozycji cyfrowej liczby, za nim zaś wyprowadzany jest znak algebraiczny według podanych wyżej zasad. W szczególnym przypadku, jeżeli znakiem tym jest zero, drukowane ono jest w takiej samej skali jak liczba nierówna zeru /tzn. jako: Ø lub Ø,Ø lub Ø,ØØ .... itd/.

Instrukcja pu będzie działać w sposób nieco odmienny od opisanego, jeżeli drugim argumentem instrukcji będzie znak # /wartość 35 w kodzie ISO-7/. W takim przypadku liczba zostanie wyprowadzona na zewnątrz bez znaku algebraicznego na końcu. Jeżeli zaś liczba w Rm będzie równa zeru, wyprowadzony będzie zawsze znak /lub znaki/ zero w odpowiedniej skali. Pozostałe omówione wyżej zasady wyprowadzania liczb pozostają niezmienione.

Przykład: Znajdującą się w rejestrze R1ØØ liczbę należy wydrukować poczynając od pozycji nr 4 z dwoma znakami po przecinku. W przypadku, gdy [R1ØØ] = Ø ma być drukowane zero, przecinek i dwa zera po przecinku. Na końcu ma być drukowany znak algebraiczny liczby. Zadanie to zrealizuje następująca sekwencja instrukcji:

```
ru ØØ4 ØØ2;
pu 1ØØ Ø48;
```

IV.6.6. Czytaj tekst do Rm

Format: qt Rm;

Instrukcja powoduje wczytanie tekstu z aktywnego urządzenia wejściowego do rejestru Rm. Zakończenie realizacji czytania następuje bądź w momencie napotkania w tekście znaku nowej linii bądź w momencie zapełnienia rejestru 15 znakami tekstu. Jeżeli wczytany tekst liczy mniej niż 15 znaków dosunięty jest do lewego końca rejestru, zaś wolne miejsca z prawej strony wypełnione są znakami spacji.

Instrukcja qt w następujący sposób ustawia rejestr warunku C:

W0 - gdy czytanie zakończył znak nowej linii,
W1 - gdy czytanie zakończyło zapełnienie rejestru,
W3 - tylko w przypadku aktywnej na wejściu pamięci kasetowej lub dyskowej - gdy przyczytany został znacznik końca zapisu.

IV.6.7. Poczynając od początku Rm pisz d znaków teketu

Format: ph Rm d;

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia, d kolejnych znaków tekstu, poczynając od lewego końca rejestru Rm. Argument d może przyjmować wartości od 1 ÷ 255 /przy aktywnej drukarce - praktycznie do 158/. Jeżeli d > 15 kolejne znaki wypisywanego tekstu pobierane są z rejestrów o numerach: Rm+1, Rm+2, ... itd.

IV.6.8. Pisz znak Z

Format: wl Z;

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia,jednego znaku, którego wartość dziesiętna w kodzie ISO-7 stanowi argument instrukcji.

Instrukcja wl znajduje szerokie zastosowanie do sterowania rozmieszczeniem wyprowadzanej informacji na ekranie monitora lub na tabulogramie, przy pomocy przekazywanych za jej pośrednictwem odpowiednich kodów funkcyjnych mających zastosowanie w odniesieniu do monitora lub drukarki.

IV.6.9. Pisz znak Z d razy

Format: pz Z d;

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia, ciągu d razy powtórzonego znaku, którego wartość dziesiętna w kodzie ISO-7 stanowi pierwszy argument instrukcji. Argument d może przybierać wartości od 1 ÷ 255 /przy aktywnej drukarce - praktycznie do 158/.

IV.6.10. Pisz tekst

Format: wt/ttt.....t/;

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia, konkretnego tekstu, stanowiącego argument instrukcji. Długość tekstu nie może przekraczać 255 znaków /przy aktywnej drukarce - praktycznie 158/. Wewnątrz tekstu nie mogą wystąpić znaki: średnik ";" i slash "/".

IV.6.11. Czytaj znak do Rm

Format: cj Rm;

Instrukcja powoduje wczytanie jednego znaku z aktywnego w danym momencie urządzenia wejściowego na pierwsze miejsce znakowe /pozycja nr Ø/ rejestru Rm. Zawartość wszystkich pozostałych pozycji rejestru nie ulega zmianie.

W przypadku aktywnej na wejściu pamięci kasetowej lub dyskowej instrukcja może ustawić w rejestrze C warunek W3 po odczytaniu znacznika końca zapisu.

IV.6.12. Pisz znak z Rm

Format: zz Rm;

Instrukcja powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia pierwszego znaku /z pozycji nr Ø/ rejestru Rm.

IV.7. INSTRUKCJE KASETOWE I DYSKOWE

IV.7.1. Przełącz kasetę

Format: pk;

Instrukcja pk na zastosowanie tylko w tych przypadkach, jeżeli w konfiguracji minikomputera znajdują się dwie jednostki pamięci kasetowej PK-1. Powoduje ona odłączenie od systemu czynnej w danym momencie jednostki pamięci i przyłączenie drugiej.

Bezpośrednio po uruchomieniu programu /po uprzednim naciśnięciu klawisza (SEL) /, dopóki nie zastosuje się instrukcji pk, czynna jest zawsze jednostka pamięci nr 1.

Aktywizacja na wejściu lub wyjściu pamięci kasetowej dotyczy zawsze jednostki pamięci przyłączonej w danym momencie do systemu.

IV.7.2. Pisz metrykę taśmy

Format: mt;

Instrukcja powoduje zametrykowanie taśmy, t.zn. zapisanie bezpośrednio za znacznikiem BOT systemowego znacznika końca zapisu. Instrukcja mt działa niezależnie od rodzaju urządzenia zewnętrznego, aktywnego w danym momencie. Po zapisaniu znacznika końca /metryki/ pamięć kasetowa jest zwalniana.

Wykonanie tej instrukcji jest równoważne logicznemu skasowaniu informacji, poprzednio na tej taśmie zapisanych.

IV.7.3. Cofnij o blok

Format: cf;

Instrukcja powoduje przewinięcie taśmy kasetowej o jeden blok wstecz, lub do znacznika BOT jeżeli odczytany /lub zapisany/ był dopiero pierwszy blok na taśmie. Instrukcja cf może być używana tylko w przypadkach, jeżeli aktywnym urządzeniem wejścia lub wyjścia jest pamięć kasetowa.

Uwaga: Jeżeli na cofanej taśmie ma być dokonywany zapis, operację cofnięcia taśmy o blok należy wykonywać następująco:

```
cf;
cf;
cb;
```

IV.7.4. Pisz znacznik końca zapisu

Format: kz;

Instrukcja powoduje zapisanie na taśmie kasetowej lub dysku elastycznym, jako kolejnego bloku na taśmie lub w kolejnym sektorze dysku /bezpośrednio po ostatnim zajętym przez dane/, systemowego znacznika końca zapisu. Urządzenie po wykonaniu tej czynności zostanie zwolnione a w przypadku taśmy kasetowej nastąpi uprzednio jej przewinięcie do fizycznego początku.

Instrukcja może być używana tylko w przypadku, gdy pamięć kasetowa lub pamięć dyskowa jest aktywnym urządzeniem wyjścia. Jest ona równoważna zamknięciu wyjściowego zbioru kasetowego lub dyskowego.

IV.7.5. Pisz blok

Format: pb;

Instrukcja działa tylko w przypadku, gdy aktywnym urządzeniem wyjścia jest pamięć kasetowa lub pamięć na dyskach elastycznych.

W przypadku pamięci kasetowej instrukcja spowoduje zapis kolejnego bloku na taśmie, którego treść stanowić będzie informacja zgromadzona przy pomocy dowolnych instrukcji wyjścia języka symbolicznego w systemowym buforze wyjścia na stronie Ø63 PAO.

W przypadku pamięci dyskowej instrukcja spowoduje natomiast zapisanie zawartości pierwszych 128 bajtów bufora wyjścia do sektora dysku elastycznego, którego adres został określony uprzednio przy pomocy instrukcji wd i pg lub za.

W przypadku nieudanej próby zapisu zapala się lampka FD i program ulega zawieszeniu. Naciśnięcie klawiszy (Ck) i (A) spowoduje ponowienie operacji zapisu.

IV.7.6. Czytaj blok

Format: cb;

Instrukcja działa tylko w przypadku, jeżeli aktywnym urządzeniem wejścia lub wyjścia jest pamięć kasetowa lub pamięć na dyskach elastycznych.

W przypadku pamięci kasetowej powoduje ona odczyt kolejnego bloku z taśmy i przeniesienie jego zawartości do bufora wejścia na stronę Ø62 PAO.

W przypadku pamięci dyskowej instrukcja powoduje odczyt sektora dysku elastycznego, którego adres został uprzednio określony przy pomocy instrukcji wd i pg lub za, oraz przeniesienie jego zawartości do bufora wejścia /na pierwsze 128 bajtów/.

Należy jeszcze raz podkreślić, że /w odróżnieniu od pamięci kasetowej/ przed użyciem jakiejkolwiek instrukcji czytania przy aktywnym na wejściu dysku elastycznym, konieczne jest uprzednie przeniesienie żądanej informacji z dysku do bufora wejściowego przy pomocy instrukcji cb /w przypadku taśmy kasetowej odbywa się to automatycznie/. W przeciwnym przypadku /przy próbie czytania z dysku i pustym buforze/ system zawiesza program i uaktywnia na wejściu klawiaturę.

W przypadku stwierdzenia błędnego odczytu zapala się lampka FD i program zostaje zawieszony. Naciśnięcie klawiszy (Ck) a następnie (A) spowoduje ponowienie operacji odczytu.

Po napotkaniu znacznika końca zapisu na taśmie kasetowej lub dysku elastycznym, instrukcja cb ustawia warunek W3 w rejestrze warunku C.

IV.7.7. Wybierz dysk

Format: wd n;

gdzie: n = Ø, 1, 2 lub 3

Instrukcja powoduje podłączenie do systemu dysku elastycznego, załadowanego do kieszeni, której numer wskazuje argument n instrukcji.

Przy aktywnej na wejściu lub wyjściu pamięci dyskowej

czynny w danym momencie będzie dysk, wskazany ostatnio użytą instrukcją wd.

Instrukcja wd musi wystąpić przed instrukcjami, przy pomocy których ustawia się głowicę odczytu/zapisu na określonym adresie dyskowym /pg i za/. Instrukcje te odnoszą się zawsze do dysku, w danym momencie podłączonego do systemu.

IV.7.8. Pozycjonuj głowicę

Format: pg Rn;

Instrukcja powoduje ustawienie głowicy odczytu/zapisu pamięci dyskowej na początku sektora, którego adres stanowi zawartość rejestru roboczego o numerze podanym jako argument instrukcji pg. Adres ten, jak wiadomo, ma postać: SSSKKK

gdzie: SSS - numer ścieżki
KKK - numer sektora

Przed wykonaniem tej operacji należy przy pomocy instrukcji wd wybrać dysk, którego ten adres ma dotyczyć.

IV.7.9. Zwiększ adres dyskowy

Format: za Rn d;

Instrukcja powoduje zwiększenie adresu dyskowego, zawartego w rejestrze Rn, którego numer wskazuje pierwszy argument instrukcji, o ilość sektorów podaną bezpośrednio jako drugi argument instrukcji. Bezpośrednio po tym głowica odczytu/zapisu zostaje ustawiona na początku sektora, wskazanego nowoutworzonym adresem. Instrukcja za musi być poprzedzona instrukcją wd, a przeważnie również instrukcją pg.

IV.8. INSTRUKCJE SPECJALNE

IV.8.1. Wprowadź dwubajtową wstawkę

Format: en 999 999;

Instrukcja en umożliwia włączanie do treści programu pisanego w języku symbolicznym dowolnych fragmentów napisanych w języku maszynowym minikomputera MERA-100. Obydwa argumenty instrukcji są w postaci kodów binarnych wstawiane do dwóch kolejnych bajtów wynikowej postaci programu i w czasie jego realizacji wykonywane jako rozkazy języka maszynowego minikomputera. Dłuższe fragmenty w języku maszynowym można wstawiać powtarzając wielokrotnie, dowolną ilość razy, instrukcję en. Jako argumenty instrukcji nie mogą jedynie występować kody: 196, 248 i 253.

IV.8.2. Zeruj tablicę

Format: zr Rm Rn;

Instrukcja powoduje wpisanie zer do wszystkich rejestrów roboczych wchodzących w skład tablicy, określonej argumentami instrukcji.

Uwaga: Pod pojęciem tablicy, używanym w tej i następnych instrukcjach, będziemy rozumieli zespół dowolnej ilości rejestrów roboczych o ciągłej numeracji od Rm do Rn włącznie /gdzie Rm < Rn/.

Wielkością tablicy będziemy nazywali ilość rejestrów roboczych wchodzących w jej skład, którą obliczyć można ze wzoru:

$$Rn - Rm + 1$$

Wewnątrz tablic można używać /niedostępnych bezpośrednio/ rejestrów roboczych o numerach: 196, 248 i 253. Żaden z nich nie może jednak rozpoczynać ani kończyć tablicy /z wyjątkiem zakończenia tablicy wynikowej w instrukcjach ra i rb, albowiem w tych przypadkach numer rejestru końcowego nie występuje jako argument instrukcji/.

IV.8.3. Podaj parametry tablicy do dodawania lub odejmowania

Format: pd Rm Rn;

Instrukcja ta, której argumenty podają numer pierwszego i ostatniego rejestru roboczego, określa tablicę, która będzie dodawana do innej tablicy w trakcie realizacji instrukcji ra, lub która będzie odejmowana od innej tablicy w trakcie realizacji instrukcji rb.

Dlatego też instrukcja pd musi występować zawsze zarówno przed instrukcją ra jak i rb.

IV.8.4. Dodaj tablice

Format: ra n Rw;

Instrukcja powoduje dodanie tablicy, określonej poprzedzającą ją instrukcją pd, do tablicy której parametry podane są jako argumenty instrukcji ra w następujący sposób:

Rw - określa numer pierwszego rejestru tej tablicy.

n - określa wielkość tej tablicy czyli ilość rejestrów wchodzących w jej skład.

Ponieważ wynik dodawania będzie umieszczony w tablicy, której parametry określa instrukcja ra, będziemy ją nazywali tablicą wynikową.

Wielkości obydwu tablic, będących składnikami dodawania, mogą być różne. Z tego względu ich dodawanie może przybierać różne formy, które zostaną kolejno rozpatrzone.

a/ n = 1

W tym przypadku, niezależnie od wielkości tablicy określonej instrukcją pd, nastąpi dodanie zawartości wszystkich kolejno rejestrów tej tablicy do zawartości rejestru Rw.

Przykład: W efekcie realizacji instrukcji:

pd 005 009;

ra 001 015;

zostaną wykonane następujące działania:

[R015] := [R015] + [R005] + [R006] +....+ [R009]

b/ $n \geq Rn - Rm + 1$

W tym przypadku nastąpi dodanie zawartości kolejnych rejestrów tablicy określonej instrukcją pd do zawartości odpowiadających im kolejnych rejestrów tablicy o takiej sa-

mej wielkości, rozpoczynającej się od rejestru Rw. Wielkość parametru n nie ma w tym przypadku znaczenia, o ile tylko jest $\geq$ Rn - Rm + 1.

Przykład: W efekcie realizacji instrukcji:

pd 001 025;
ra 100 051;

zostaną wykonane następujące działania:

[R051] := [R051] + [R001]
[R052] := [R052] + [R002]
[R053] := [R053] + [R003]
.
.
.
[R075] := [R075] + [R025]

c/ n $<$ Rn - Rm + 1

W tym przypadku tablica określona instrukcją pd zostanie podzielona na ciąg tablic o wielkości identycznej jak tablica wynikowa. Tablice te zostaną kolejno dodane do tablicy wynikowej. Jeżeli ostatnia z tablic będzie niepełna, rejestry jej zostaną dodane do takiej samej ilości pierwszych w kolejności rejestrów tablicy wynikowej.

Przykład: W efekcie realizacji instrukcji:

pd 101 110;
ra 004 201;

zostaną wykonane następujące działania:

[R201] := [R201] + [R101] + [R105] + [R109]
[R202] := [R202] + [R102] + [R106] + [R110]
[R203] := [R203] + [R103] + [R107]
[R204] := [R204] + [R104] + [R108]

We wszystkich omówionych przypadkach zawartości rejestrów Rm + Rn nie ulegają zmianie. Jeżeli w którymkolwiek rejestrze tablicy wynikowej wystąpi w trakcie dodawania nadmiar, realizacja instrukcji zostanie w tym momencie zakończona a do rejestru C - załadowany warunek W3.

IV.8.5. Odejmij tablice

Format: rb n Rw;

Instrukcja powoduje odjęcie tablicy, określonej poprzedzającą ją instrukcją pd, od tablicy której parametry poda-

ne są jako argumenty instrukcji rb w następujący sposób:

Rw - określa numer pierwszego rejestru tej tablicy

n - określa wielkość tej tablicy.

Wynik odejmowania zostanie umieszczony w tablicy, której parametry określa instrukcja rb. Dlatego też tablicę tę będziemy nazywać wynikową.

Ponieważ wielkości obydwu tablic, będących składnikami odejmowania, mogą być różne, ich odejmowanie może przybierać różne formy, które zostaną kolejno rozpatrzone.

a/ n = 1

W tym przypadku zawartości wszystkich rejestrów tablicy określonej instrukcją pd zostaną kolejno odjęte od zawartości rejestru Rw.

Przykład: W efekcie realizacji instrukcji:

pd 001 010;

rb 001 020;

zostaną wykonane następujące działania:

[R020] := [R020] - [R001] - [R002] - ... - [R010]

b/ n $\geqslant$ Rn - Rm + 1

W tym przypadku nastąpi odjęcie zawartości kolejnych rejestrów tablicy określonej instrukcją pd od zawartości odpowiadających im kolejnych rejestrów tablicy o takiej samej wielkości, rozpoczynającej się od rejestru Rw.

Przykład: W efekcie realizacji instrukcji:

pd 201 203;

rb 010 250;

zostaną wykonane następujące działania:

[R250] := [R250] - [R201]

[R251] := [R251] - [R202]

[R252] := [R252] - [R203]

c/ n $<$ Rn - Rm + 1

W tym przypadku tablica określona instrukcją pd zostanie podzielona na ciąg tablic o wielkości identycznej jak tablica wynikowa. Tablice te zostaną kolejno odjęte od tablicy wynikowej. Jeżeli ostatnia tablica będzie niepełna - działanie instrukcji będzie analogiczne jak przy dodawaniu /instrukcja ra/.

Przykład: W efekcie realizacji instrukcji:

pd 151 161;
rb 003 000;

zostaną wykonane następujące działania:

[R000] := [R000] - [R151] - [R154] - [R157] - [R160]
[R001] := [R001] - [R152] - [R155] - [R158] - [R161]
[R002] := [R002] - [R153] - [R156] - [R159]

We wszystkich powyższych przypadkach zawartości rejestrów Rm ÷ Rn nie ulegają zmianie. Jeżeli w którymkolwiek rejestrze tablicy wynikowej wystąpi nadmiar, realizacja instrukcji zostanie w tym momencie zakończona a do rejestru C - załadowany warunek W3.

IV.8.6. Podaj parametry tablicy do pisania
Format: pp Rm Rn;

Instrukcja ta, której argumenty podają numer pierwszego i ostatniego rejestru roboczego, określa tablicę, która będzie wyprowadzona na zewnątrz przy pomocy instrukcji pt. Toteż instrukcja pp musi zawsze występować bezpośrednio przed instrukcją pt.

IV.8.7. Pisz tablicę
Format: pt n m;

Instrukcja pt powoduje wyprowadzenie na zewnątrz, poprzez aktywne w danym momencie urządzenie wyjścia, tablicy określonej poprzedzającą tę instrukcję instrukcją pp.

Instrukcja pt została jednak opracowana z przeznaczeniem do wyprowadzania tablic przede wszystkim na drukarkę, dlatego też argumentami jej są dwa parametry mające istotne znaczenie tylko przy drukowaniu tablic, a mianowicie:

n - określający ilość rubryk drukowanej tablicy, czyli ilość kolejnych rejestrów roboczych, których zawartość będzie drukowana w jednym wierszu. Z tego względu argument n musi być zawsze ograniczony maksymalną ilością znaków mieszczących się w jednym wierszu, wynoszącą 159.

m - określający wskaźnik tabulacji poziomej przy drukowaniu poszczególnych wierszy tablicy, tzn. ilość wol-

nych znaków wiersza, licząc od lewego marginesu, jaką należy pozostawić przed wydrukiem pierwszej pozycji każdego wiersza.

Liczby stanowiące zawartość poszczególnych rejestrów wchodzących w skład tablicy drukowane są wszystkie jednakowo według zasad, obowiązujących w przypadku instrukcji pu z tym, że w sytuacji gdy zawartość rejestru równa się zeru drukowana jest zawsze liczba zero w odpowiedniej skali. Ponadto, jak z powyższego wynika, instrukcja pt musi być również poprzedzona instrukcją ru, określającą wspólny format wydruku wszystkich liczb stanowiących pozycje tablicy. Kolejność występowania instrukcji ru i pp, poprzedzających instrukcję pt, jest dowolna.

Wydruk tablicy na tabulogramie może przybierać różne postacie, w zależności od wielkości tablicy i ilości rubryk na wydruku:

a/ $n = 1$

Zawartość wszystkich rejestrów stanowiących tablicę zostanie wydrukowana w postaci jednej kolumny, tzn. każda pozycja w oddzielnym wierszu, poczynając od $m + 1$ znaku wiersza.

b/ $n \geqslant Rn - Rm + 1$

Zawartość wszystkich rejestrów wchodzących w skład tablicy zostanie wydrukowana kolejno w jednym wierszu.

c/ $n < Rn - Rm + 1$

Zawartość wszystkich rejestrów zostanie wydrukowana kolejno wierszami w formie tabeli, liczącej n rubryk oraz:

$\frac{Rn - Rm + 1}{n}$ wierszy, gdy iloraz jest liczbą całkowitą,

$\left[\frac{Rn - Rm + 1}{n}\right] + 1$ wierszy, gdy iloraz zawiera część ułamkową.

Nawiasy kwadratowe oznaczają w tym przypadku część całkowitą wyrażenia w nich zawartego.

V. DODATKI

V.A. KODY ZNAKÓW I ROZKAZÓW W SYSTEMIE MERA - 100

Kod ASCII	Rozkaz języka maszynowego /mnemonika polska/	Reprezentacja znaku	Klawisze funkcyjne	Kod ISO-7
1	2	3	4	5
255	R1 → B	NUL		000
254	TIME	SOH	A	001
253	K → R1	STX	B	002
252		ETX	C	003
251	+	EOT	◊	004
250		ENQ	*	005
249		ACK	L	006
248		BEL		007
247	R1 → A	BS	LK	008
246		HT	HT	009
245		LF	LF	010
244		VT	S	011
243	-	FF	ST	012
242		CR		013
241		SO		014
240		CI		015
239		DLE		016
238		DC1		017
237		DC2		018
236	Z → R1	DC3		019
235	Z+1	DC4		020
234	PO → Z	NAK		021
233	PO → V	SYN		022
232	PO → R1	ETB		023
231	ADR+1	CAN	ØØ	024
230	POWROT	EM	Ck	025
229	PODPR	SUB	BS	026
228	SK.A > B	ESC		027
227	SK.A < B	FS		028
226	SK.R=V	GS		029
225	SK.A=B	RS		030
224	SKOK	US		031
223	R1 → DR	Spacja		032
222	R1 → P	!		033
221	P → R1	"		034
220		#		035
219		$		036
218		%		037
217		&		038
216		'		039

1	2	3	4	5
215	P	(		040
214	P-1	)		041
213	P+1	*		042
212	PADR	+		043
211	B-1	,		044
210	B+1	-		045
209	BADR	.		046
208	B	/		047
207	LLRC	0		048
206	PRZEŁĄCZ	1		049
205	P.CZT	2		050
204	K.PSZ	3		051
203	P.PSZ	4		052
202	PRZ	5		053
201	KRS	6		054
200	PRS	7		055
199	KRW	8		056
198	PRW	9		057
197	KRP	:		058
196	PRP	;		059
195	KPO	<		060
194	PPO	=		061
193	K.REZ	>		062
192	P.REZ	?		063
191	KZ6	ę		064
190	PZ6	A		065
189	KZ5	B		066
188	PZ5	C		067
187	KZ4	D		068
186	PZ4	E		069
185	KZ3	F		070
184	PZ3	G		071
183	KZ2	H		072
182	PZ2	I		073
181	KZ1	J		074
180	PZ1	K		075
179		L		076
178	KAC	M		077
177	PAC	N		078
176	KL → R1	O		079
175		P		080
174		Q		081
173		R		082
172		S		083
171	KZ12	T		084
170	PZ12	U		085
169	KZ11	V		086
168	PZ11	W		087
167	KZ10	X		088
166	PZ10	Y		089
165	KZ9	Z		090
164	PZ9	ź		091
163	KZ8	Ł		092
162	PZ8	ń		093

1	2	3	4	5
161	KZ7	ś		094
160	PZ7	˙		095
159		ą		096
158		a		097
157		b		098
156		c		099
155		d		100
154		e		101
153		f		102
152	CRS	g		103
151		h		104
150		i		105
149	RS	j		106
148	AR	k		107
147	STU	l		108
146	ODB	m		109
145	NAD	n		110
144	ROZ	o		111
143	KDP	p		112
142	DP → X	q		113
141	PDP	r		114
140	R1 → DP	s		115
139	INT	t		116
138	X → R1	u		117
137		v		118
136		w		119
135		x		120
134	ZER.PK	y		121
133	K.DAN	z		122
132	ST.SZCZ	ó		123
131	STAT	ł		124
130	R1 → PK	ż		125
129	PK → R1	ć		126
128	LRC	◆		127

V.B. LISTA INSTRUKCJI JĘZYKA SYMBOLICZNEGO KD-100

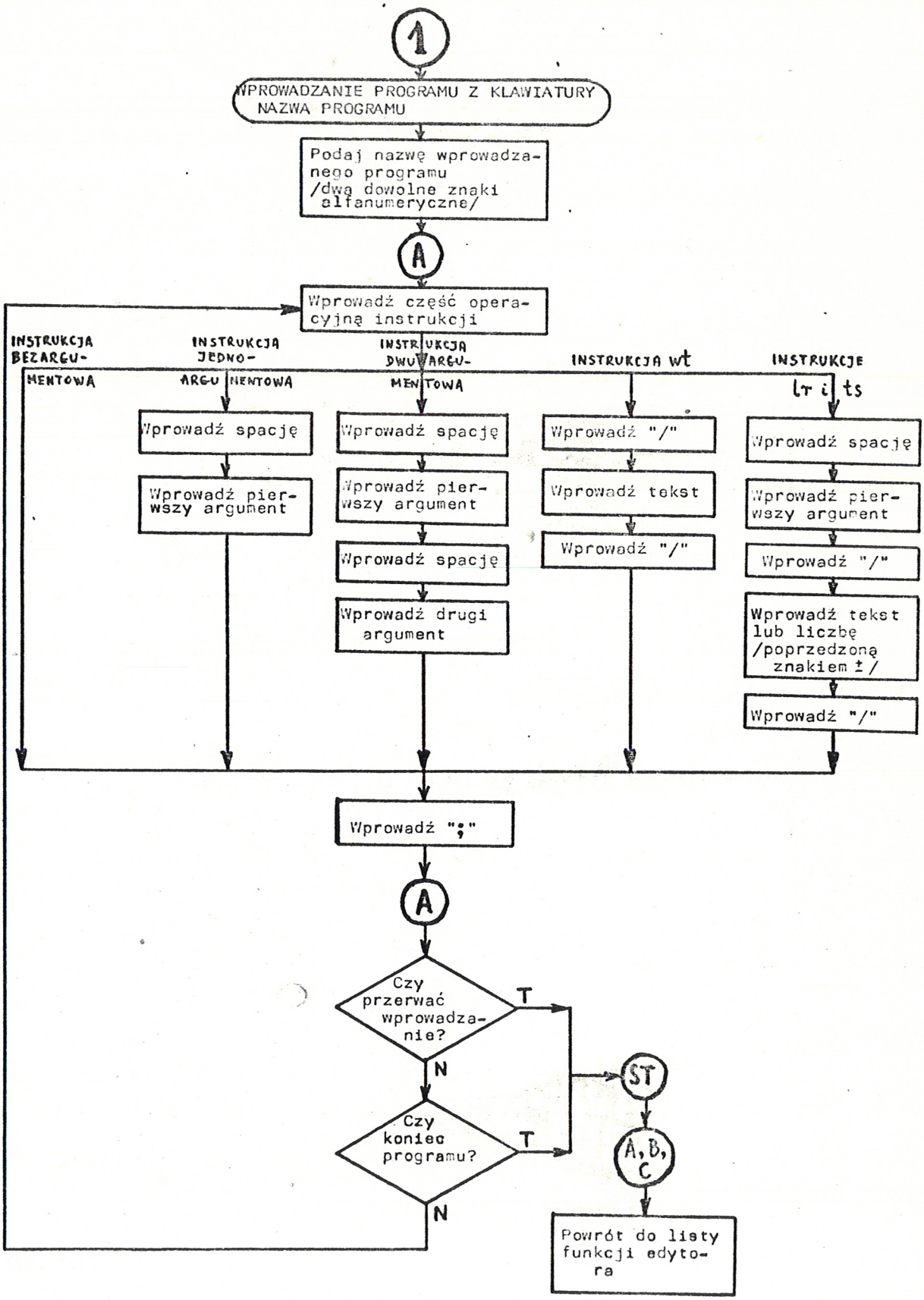
Format	Funkcja
	1.DYREKTYWY
dl En;	Deklaracja etykiety
sk En;	Deklaracja podprogramu
dz;	Fizyczny koniec programu
	2. INSTRUKCJE STERUJĄCE
st En;	Skocz do En
sr En;	Skocz do En gdy W0
wi En;	Skocz do En gdy W1
ms En;	Skocz do En gdy W2
wn En;	Skocz do En gdy W3
wo En;	Skocz do podprogramu En
wą En;	Wróć z podprogramu En
sq Rm;	Skocz do etykiety, której numer wskazuje zawartość Rm
sh;	Stop /skocz do EG/
	3. INSTRUKCJE ARYTMETYCZNE
rd Rm Rn;	Dodaj [Rn] do [Rm]
ro Rm Rn;	Odejmij [Rn] od [Rm]
rx a b;	Podaj parametry dla dzielenia lub mnożenia: a = ilość cyfr obcinanych w wyniku b = {0 - bez zaokrąglenia 5 - zaokrąglić wynik
rm Rm Rn;	Pomnóż [Rm] przez [Rn]
ri Rm Rn;	Podziel [Rm] przez [Rn]
pw Rm Rn;	Porównaj liczby zawarte w Rm i Rn
cn Rm;	Porównaj liczbę zawartą w Rm z zerem
rp Rm d;	Przesuń liczbę w Rm w prawo o d pozycji
rl Rm d;	Przesuń liczbę w Rm w lewo o d pozycji

	4. PRZESŁANIA
rz Rm Rn;	Prześlij [Rn] do Rm
ai Rm Rn;	Prześlij [Rn] do R(Rm)
am Rm Rn;	Prześlij [R(Rn)] do Rm
im Rm Rn;	Prześlij [R(Rn)] do R(Rm)
lr Rm/±99..9/;	Umieść liczbę w Rm
ts Rm/tt...t/;	Umieść tekst w Rm
az Rm Z;	Zapełnij Rm znakiem Z
	5. INSTRUKCJE WEJSCIA/WYJŚCIA
we S;	Deklaracja urządzenia wejścia: 1 - klawiatura z monitorem 4 - klawiatura z drukarką 5 - pamięć kasetowa 6 - pamięć dyskowa
wy S;	Deklaracja urządzenia wyjścia: 1 - monitor ekranowy 5 - pamięć kasetowa 6 - pamięć dyskowa 8 - drukarka 9 - pamięć kasetowa + drukarka
qy Rm Rn;	Czytaj liczbę do Rm a ogranicznik do Rn
ru m n;	Podaj parametry do pisania liczby: m = nr pozycji, od której rozpocznie się pisanie n = ilość cyfr po przecinku
pu Rm Z;	Pisz liczbę z Rm. Jeśli [Rm] =Ø pisz Z
qt Rm;	Czytaj tekst do Rm
ph Rm d;	Poczynając od Rm pisz d znaków tekstu
wl Z;	Pisz znak Z
pz Z d;	Pisz znak Z d razy
wt/ttt....t/;	Pisz tekst
cj Rm;	Czytaj znak do Rm
zz Rm;	Pisz znak z Rm

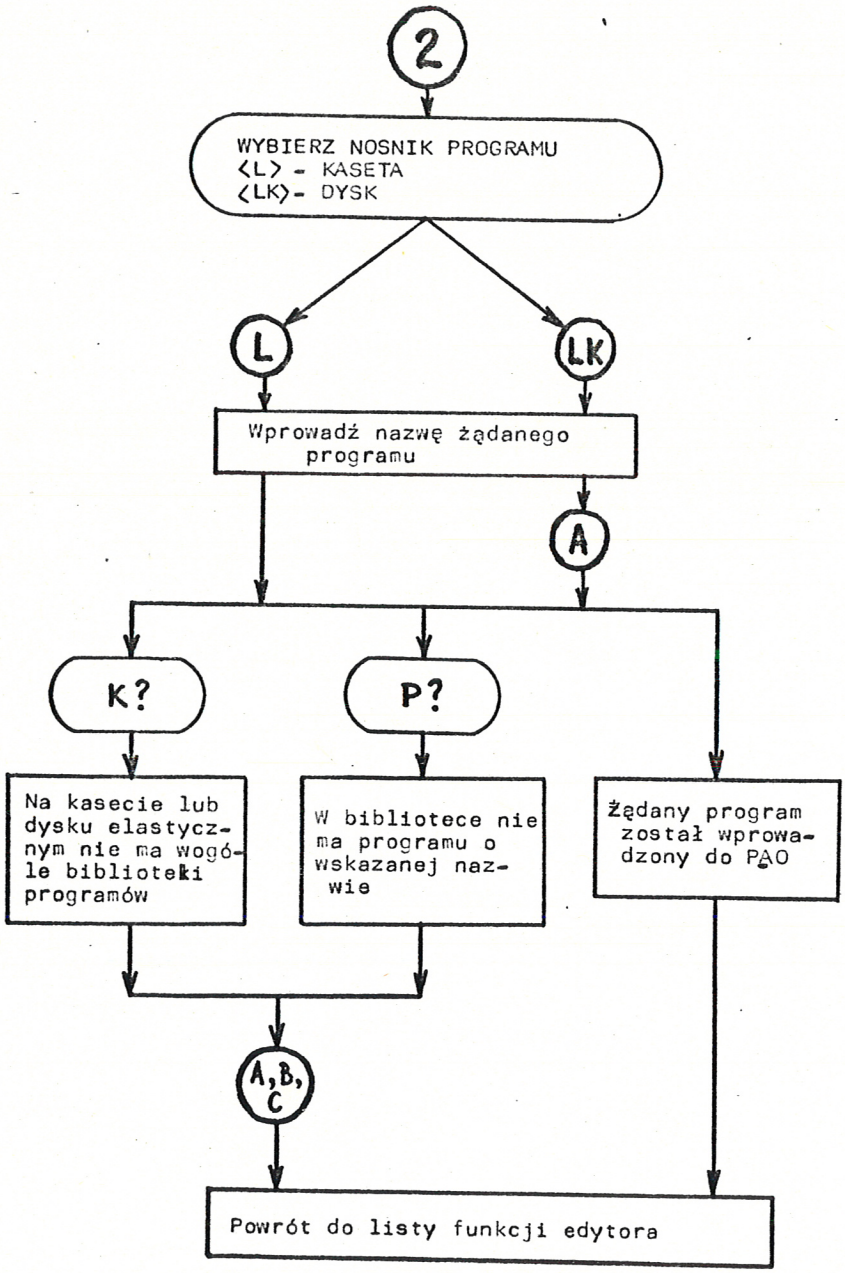
	6. INSTRUKCJE TEKSTOWE
qq Rm Rn;	Porównaj teksty zawarte w Rm i Rn
al Rm Z;	Porównaj pierwszy znak z Rm ze znakiem Z
	7. INSTRUKCJE KASETOWE I DYSKOWE
pk;	Przełącz kasetę
mt;	Pisz metrykę taśmy /bez deklaracji wyjścia/
cf;	Cofnij o blok
kz;	Pisz znacznik końca zapisu
pb;	Pisz blok
cb;	Czytaj blok
wd n;	Wybierz dysk
pg Rn;	Pozycjonuj głowicę
za Rn n;	Zwiększ adres dyskowy
	8. INSTRUKCJE SPECJALNE
en 999 999;	Wprowadź dwubajtową wstawkę
zr Rm Rn;	Zeruj tablicę
pd Rm Rn;	Podaj parametry tablicy do dodawania lub odejmowania
ra n Rw;	Dodaj tablice
rb n Rw;	Odejmij tablice
pp Rm Rn;	Podaj parametry tablicy do pisania
pt n m;	Pisz tablicę: n - ilość rubryk tablicy m - wskaźnik tabulacji poziomej

V.C. SCHEMATY REALIZACJI FUNKCJI EDYTORA

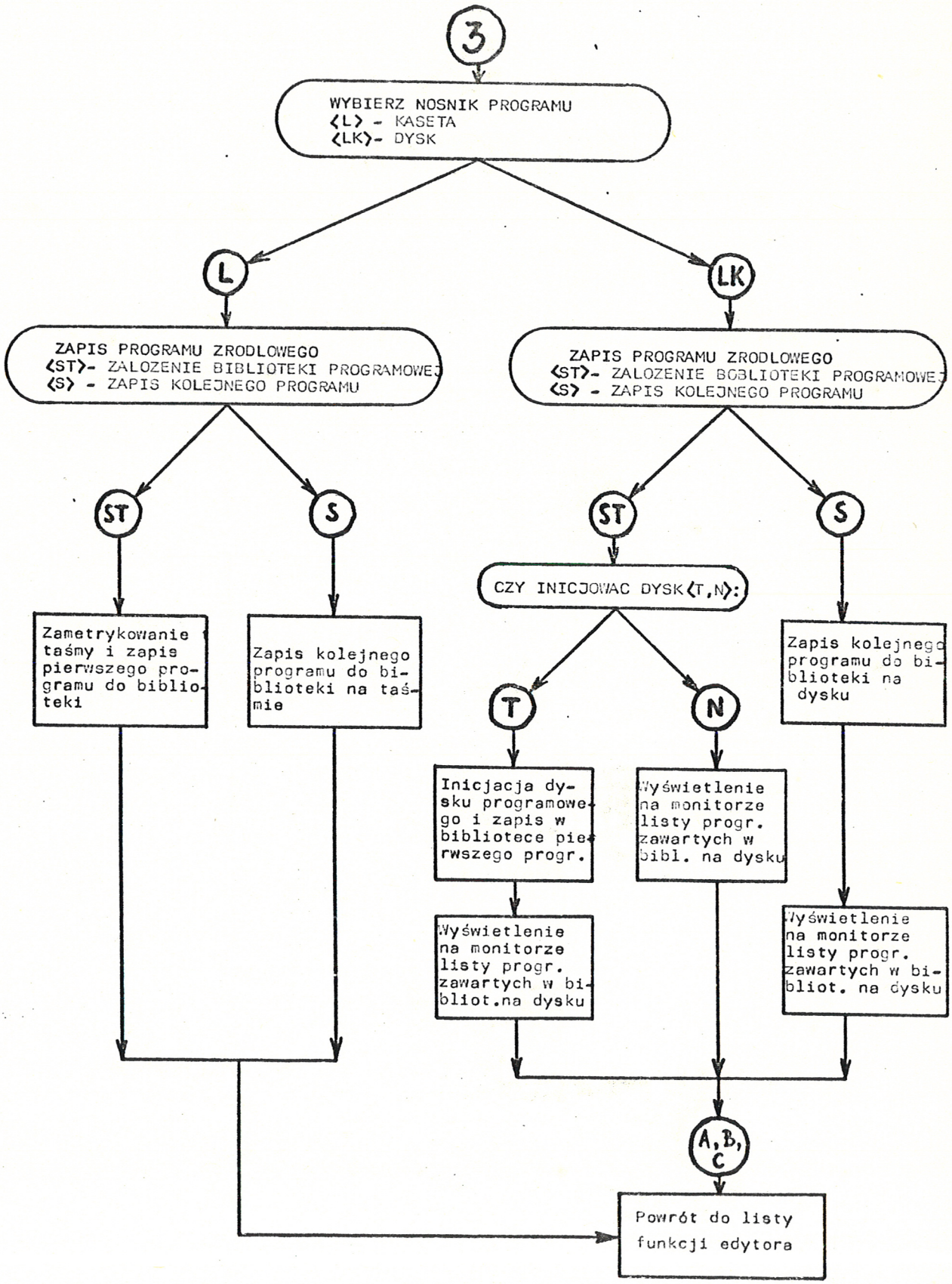
V.C.1. WPROWADZANIE PROGRAMU Z KLAWIATURY



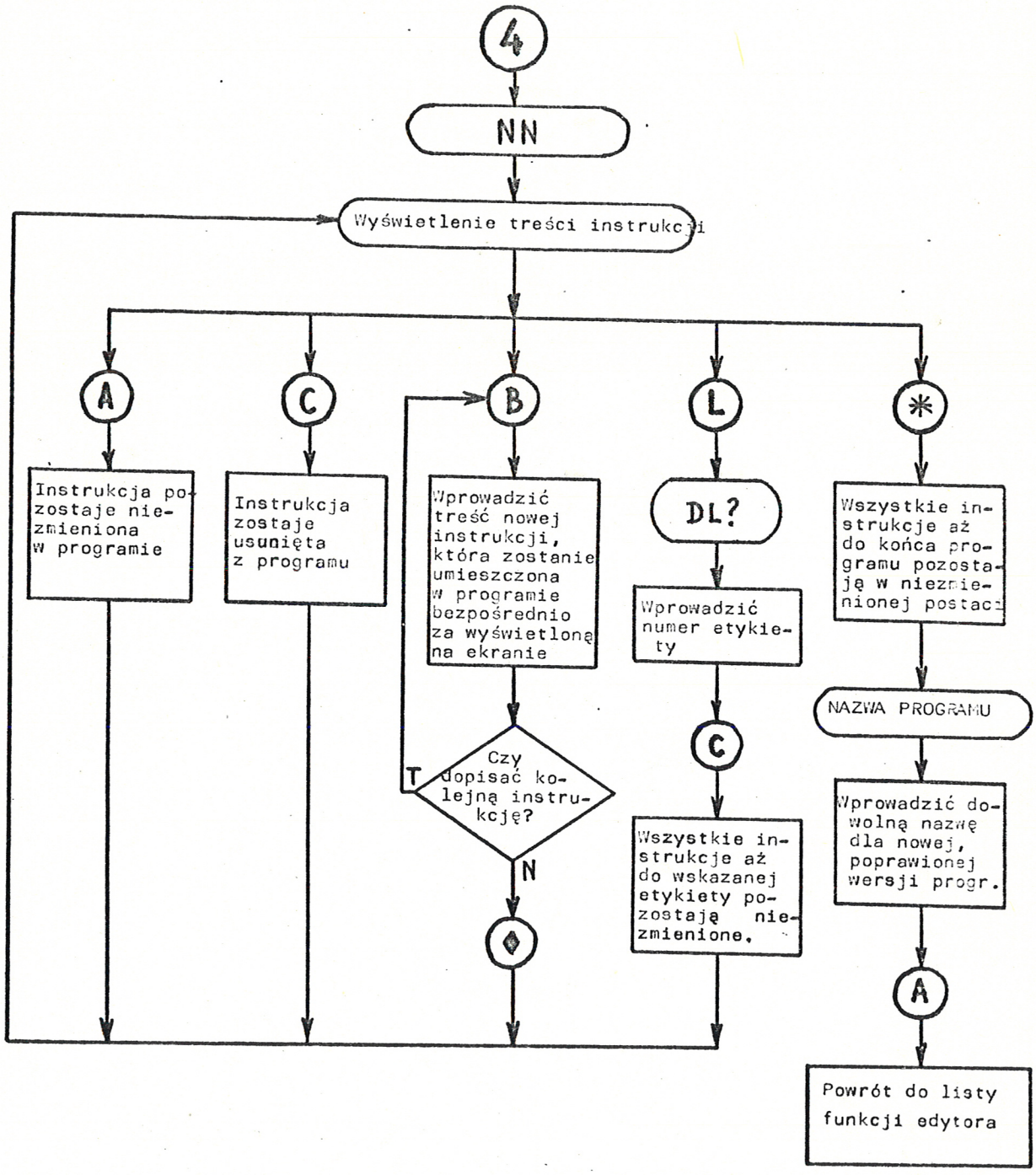
V.C.2. CZYTANIE PROGRAMU



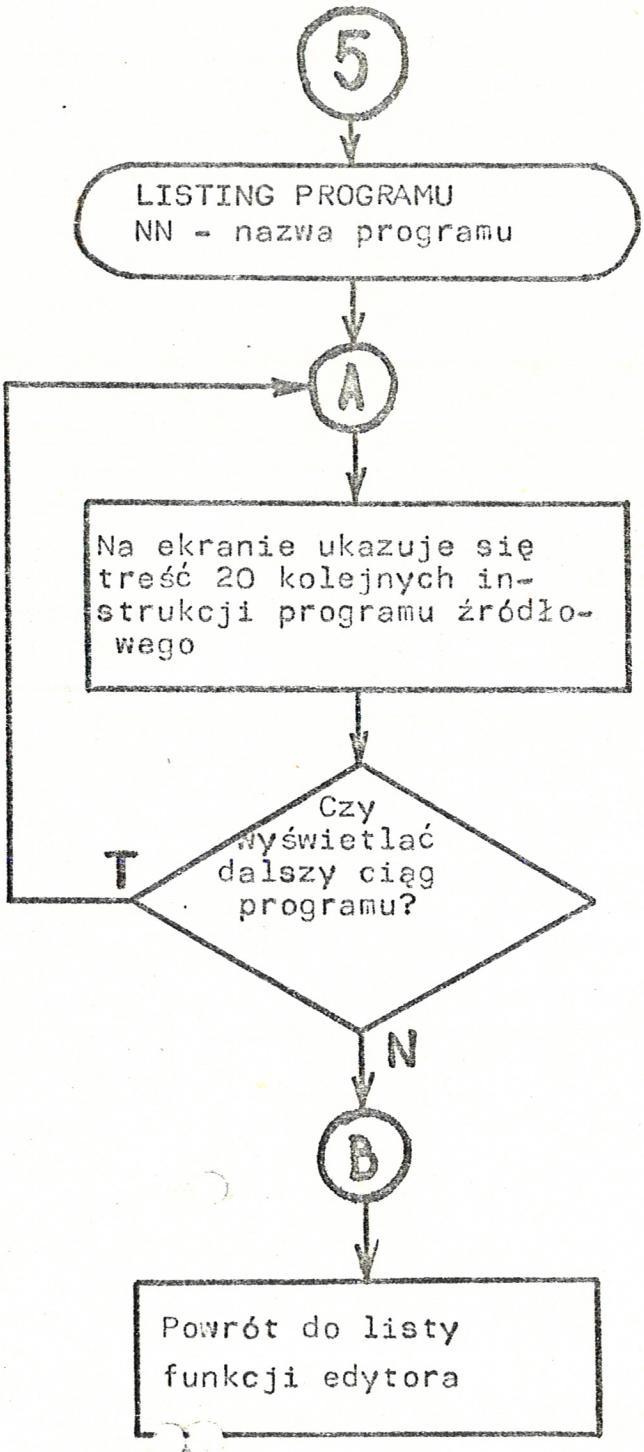
V.C.3. ZAPIS PROGRAMU I LISTOWANIE BIBLIOTEKI



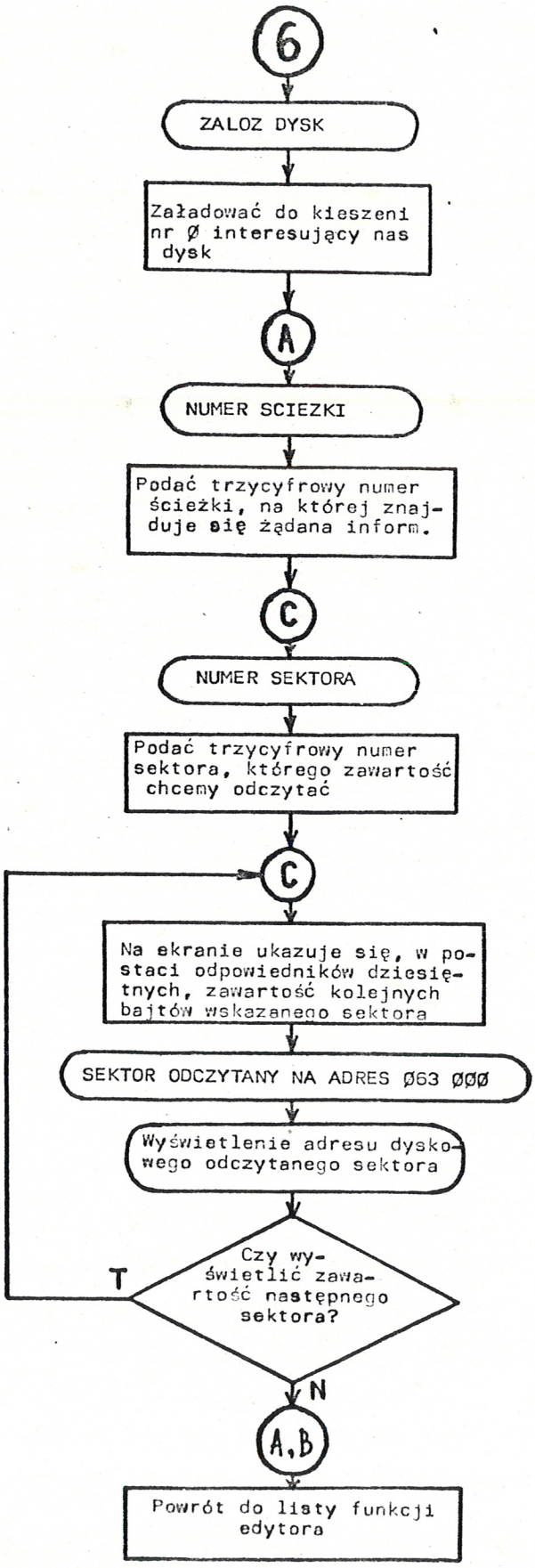
V.C.4. KOREKTA PROGRAMU



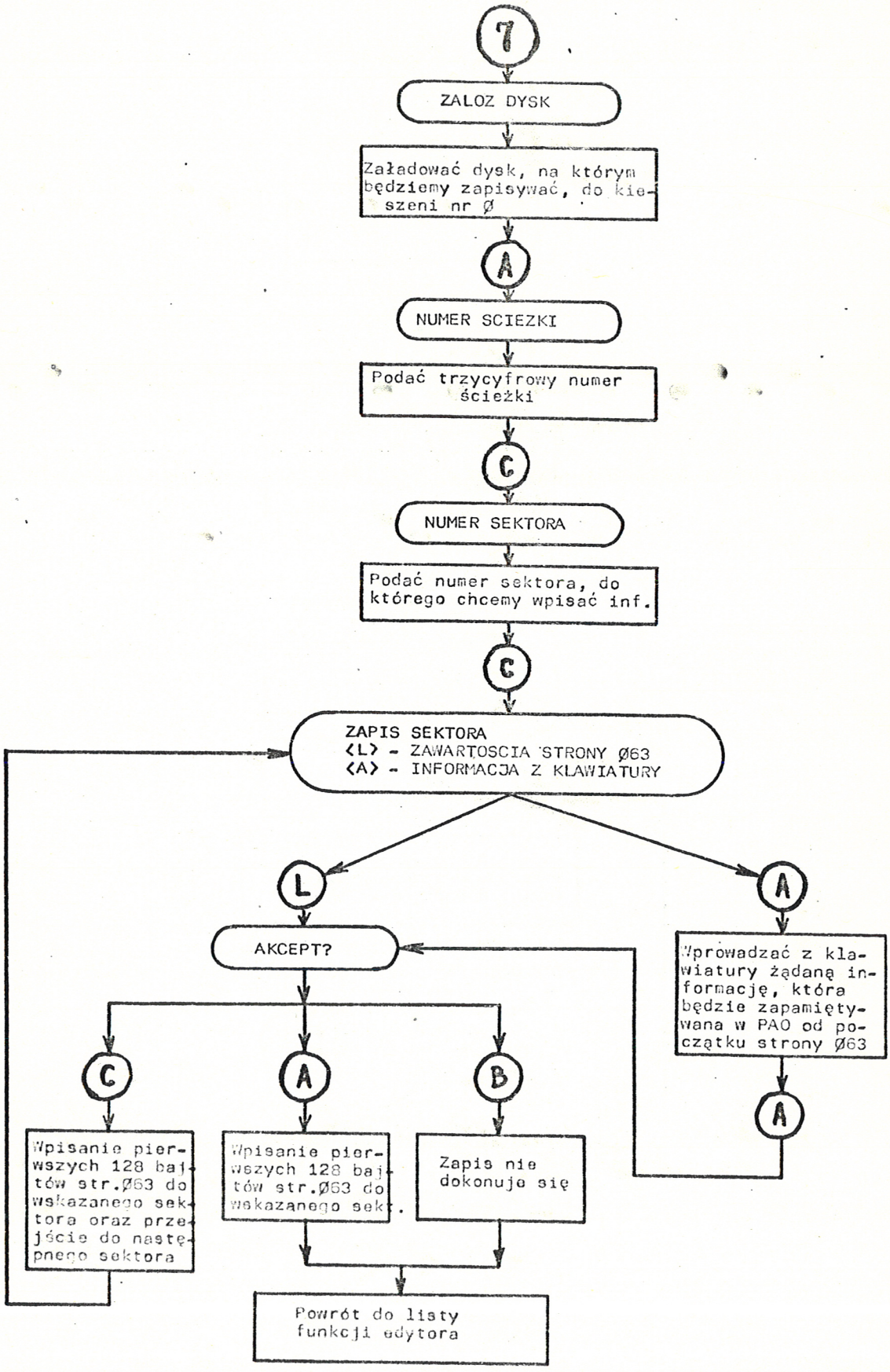
V.C.5. LISTOWANIE PROGRAMU



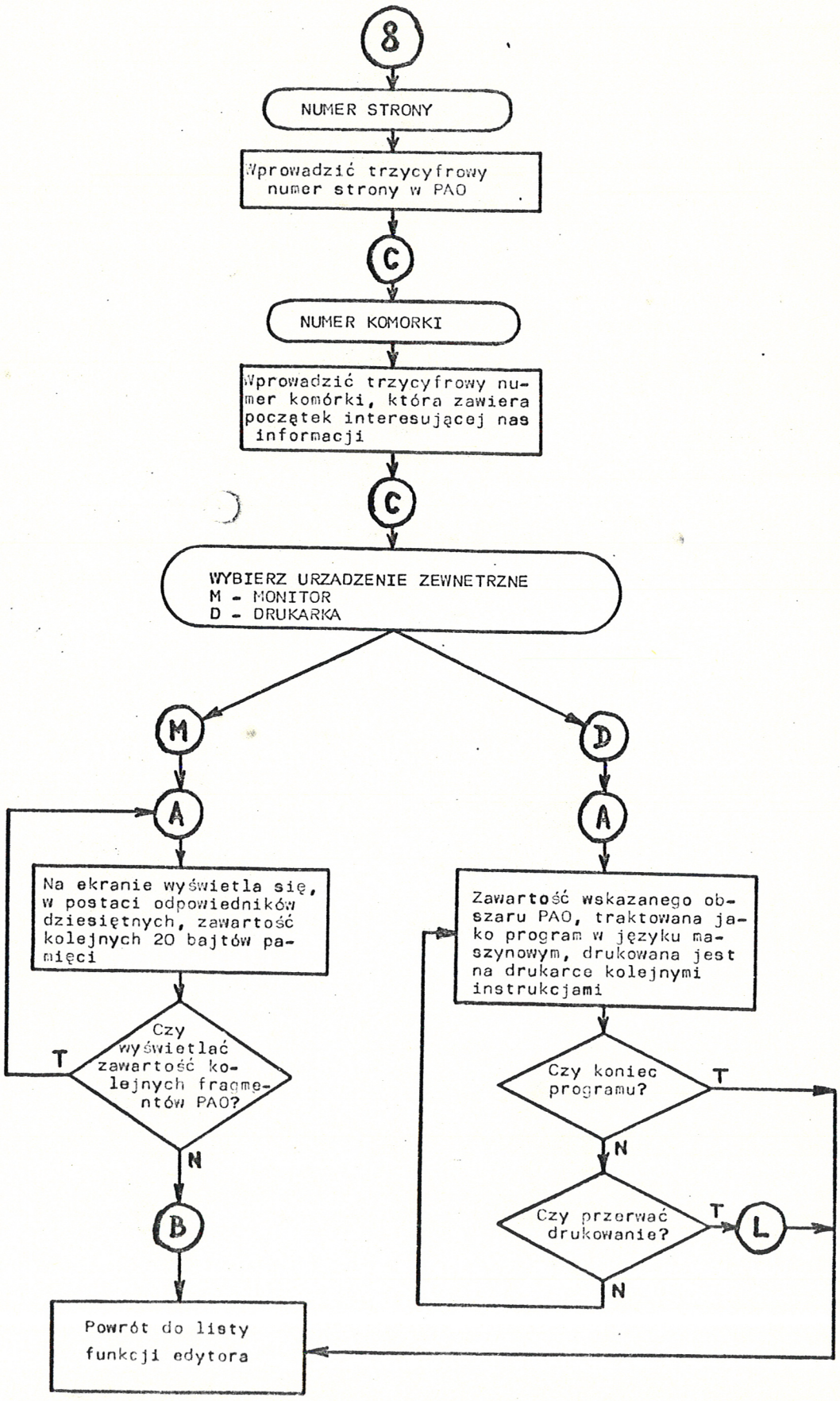
V.C.6. ODCZYT SEKTORA



V.C.7. ZAPIS SEKTORA



V.C.8. LISTOWANIE PAO



V.C.9. KOREKTA PAO