



KRAKOWSKA FABRYKA  
APARATOW POMIAROWYCH  
"MERA-KFAP"

Pamięć na dyskach elastycznych  
z formaterem i selektorem typ SP 45 DE.

Tom I. Część IV

Mikroprogram



## Tom I Część IV

### Spis treści

1.	Wiadomości ogólne .....	1
1.1.	Inicjacja programu .....	1
1.2.	Początek wykonywania .....	1
1.3.	Podstawowe mikroprogramy .....	2
1.3.1.	Sprowadzenie do pozycji spoczynkowej /restore; adres 88/ .....	2
1.3.2.	Wymiana informacji bufor-minikomputer .....	2
1.3.2.1.	Odczyt bufora /adres 14/ .....	2
1.3.2.2.	Zapis bufora /adres 83 lub 85/ .....	2
1.3.3.	Operacje zapisu i odczytu na dysku .....	3
1.3.3.1.	Szukanie początku sektora /adres 23/ .....	3
1.3.3.2.	Kontrola adresu .....	4
1.3.3.3.	Detekcja rozkazu /adres 7C/ .....	4
1.3.3.4.	Odczyt sektora /adres 17/ .....	5
1.3.3.5.	Zapis sektora /adres 51/ .....	5
1.4.	Mikro-podprogramy użytkowe .....	6
1.4.1.	Procedura obliczenia CRC /adres 8E/ .....	6
1.4.2.	Procedura sprawdzenia CRC /adres 4D/ .....	6
1.4.3.	Procedura stałej czasowej /adres 0C/ .....	6
1.4.4.	Pętla końcowa /adres A7/ .....	6
1.4.5.	Przerwanie indeksowe IM /adres 0A/ .....	7
1.4.6.	Odczyt jednego bajtu /adres 43/ .....	7
2.	Listing mikro-programu .....	18
3.	Format IBM .....	24



IV Mikroprogram1. Wiadomości ogólne

Zrozumienie mikroprogramu opisanego poniżej wymaga przeczytania punktu 3 opisującego format zapisu.

Program wypełnia trzy podstawowe zadania:

- 1 - doprowadza głowicę czytająco-piszącą do zadanego położenia,
- 2 - wykonuje różne operacje na dysku sformatowanym wg IBM,
- 3 - nawiązuje dialog między pamięcią buforową a mini-komputerem.

1.1. Inicjacja programu.

Poszczególne operacje wymagają zapoczątkowania, na które składa się sprowadzenie na zero rejestru rozkazów OPR2 i rejestru danych OPR5 oraz ustawienie na "1" rejestru CRC.

1.2. Początek wykonywania.

Po odebraniu sygnału wykonywania EX wyzerowana zostaje flaga 5 determinująca odczyt kontrolny, wpisany zostaje numer ścieżki do rejestru R0, bajt FF dla określenia anomalii selektora do rejestru R5, "2" do licznika kroków, a następnie wprowadzony zostaje rozkaz z interfejsu.

Rozkaz po umieszczeniu go w R2 i R1 poddany jest sprawdzeniu /stan bitu 8/ - co pozwala określić rodzaj rozkazu:

- obecność bitu 8 - dialog z minikomputerem, głowice na ścieżkę spoczynkową
- brak bitu 8 - operacja odczytu lub zapisu.

W przypadku dialogu z minikomputerem lub sprowadzenia głowic na ścieżkę spoczynkową zapamiętywany jest adres końcowy bufora /1/ w rejestrze R4 i początkowy /130/ w rejestrze R3.

Analiza rozkazu pozwala się udać do adresu wskazanego przez rejestr R1:

- 88 dla restore
- A4 " odczytu z bufora
- 83 " zapisu w buforze z cechą FB
- 85 " zapisu w buforze z cechą F8



### 1.3. Podstawowe mikroprogramy.

#### 1.3.1. Sprowadzanie do pozycji spoczynkowej /restore; adres 88/.

Operacja wycofania głowicy do pozycji początkowej przebiega następująco: zadanie sprowadzenia głowicy przekazane jest do selektora i potwierdzone sygnałem EXS. Po otrzymaniu odpowiedzi z selektora sygnał "IT" i EXS są ustawione na zero i przechodzi się do pętli końcowej /A7/, opisanej w punkcie 1.4.4.

#### 1.3.2. Wymiana informacji bufor-minikomputer.

Dialog między buforem a minikomputerem nawiązany po otrzymaniu sygnału "wykonywanie" - pozwala minikomputerowi na czytanie z bufora lub pisanie w buforze.

##### 1.3.2.1. Odczyt bufora /adres A4/.

Początkowy adres komórki bufora, zapisany w rejestrze R3 jest dekrementowany przy każdym bajcie pobranym z pamięci. Odczytany bajt umieszczany jest w rejestrze wyjściowym, przy równoczesnym wysłaniu sygnału "obecność znaku" /PC/. Sygnał PC zostaje anulowany z chwilą otrzymania sygnału "potwierdzenie przyjęcia znaku" /PE/.

Gdy /R3/ = /R4/ /w związku z ciągłą dekrementacją/ - bufor został odczytany i cykl zostaje zakończony.

##### 1.3.2.2. Zapis bufora /adres 83 lub 85/.

Zapis bufora rozpoczyna się wpisaniem bajtu początku danych /DM/ do komórki bufora o adresie 130. DM = F8 - przy zapisie z cechą F8, DM = FB - przy zapisie z cechą FB.

Adres bufora jest dekrementowany o 1.

Wysyłany jest sygnał "żądanie symbolu" /DC/ i trwa aż do otrzymania sygnału "znak gotów" /CP/, co pozwala na zapis bajtu z rejestru wejściowego do rejestru R1. Po skasowaniu żądania znaku zawartość R1 przepisywana jest do bufora pod wskazany adres. Adres ten jest ponownie dekrementowany i przeprowadza się test końca bufora. Operacja ustaje po osiągnięciu adresu 1.



Na koniec cyklu sygnał "koniec wykonywania" wysyłany jest do minikomputera. Adres bufora ustawia się na 130 aby wykonać obliczenie CRC. Robi się to dla wszystkich bajtów aż do adresu 2 włącznie. Oba bajty CRC zostają zapisane w komórkach o adresie "1" i "0", po czym następuje pętla końcowa.

### 1.3.3. Operacje zapisu i odczytu na dysku.

#### 1.3.3.1. Wyszukiwanie początku sektora /adres 23/.

Podprogram rozpoczyna się wysłaniem do selektora numeru ścieżki i rozkazu wykonywania EXS oraz wpisu "3" do licznika obrotów R6. Po otrzymaniu odpowiedzi selektora badany jest status i w przypadku wystąpienia anomalii informacja o tym zostaje wysłana do interfejsu, a podprogram kończy się pętlą końcową.

Jeżeli nie ma anomalii demaskowane zostaje przerwanie IM i rozpoczyna się szukanie początku sektora.

Każdy sektor rozpoczyna się bajtem początku sektora AM o danych FE i zegarze C7. Wyszukanie sektora dla każdej operacji odczytu i zapisu sprowadza się więc do detekcji oktetu AM /FE C7/. Detekcja ta realizowana jest przez kolejne porównywanie bajtów danych i zegara z bajtem AM /do dwóch obrotów/. Jeżeli wyszukanie AM nie zostaje uwieńczone powodzeniem na danej ścieżce, następuje zmiana ścieżki. Jeżeli po dwóch zmianach ścieżki AM nie został odnaleziony, sygnał anomalii zostaje przekazany do minikomputera.

Jeżeli bajt jest identyczny z /FE C7/ , uważa się, że chodzi o początek sektora. Detekcja ta pozwala na zapis AM w pamięci pod adresem 249. Po AM występuje identyfikator /ID/ podający numer ścieżki i sektora, na których znajduje się głowica. Przed odczytem każdego kolejnego bajtu identyfikatora inkrementuje się adres pamięci /rejestr R3/.

Pierwszy bajt ID - czytany zgodnie z zasadą opisaną dalej zapisany jest w pamięci o adresie 250.

Cykl odczytu i zapisu bajtu kontynuuje się aż do adresu 00 pamięci - co odpowiada ostatniemu bajtowi CRC.



### 1.3.3.2. Kontrola adresu.

Po zapisie w pamięci ID i dwóch bajtów CRC następuje kontrola adresu obejmująca kontrolę CRC, kontrolę numeru ścieżki i kontrolę numeru sektora. Obliczanie CRC jest przeprowadzane wg zasady opisanej dalej - na bajtach AM i ID.

Obliczone CRC podlega kontroli pozwalającej określić:

- właściwy CRC
- niewłaściwy CRC.

Kontrola numeru ścieżki pozwala odczytać numer zapisany w pamięci i porównać go z numerem ścieżki żądanej.

- Gdy wynik porównania jest negatywny, podobnie jak CRC - wykonuje się nowe poszukiwanie AM. Jak już wspomniano poprzednio nowe poszukiwanie może doprowadzić do dwukrotnej zmiany ścieżki.
- Gdy wynik porównania jest ujemny, a CRC dobry - wykonuje się fizyczną zmianę ścieżki, po czym następuje poszukiwanie AM na tej nowej ścieżce.
- Gdy wynik porównania jest dodatni - porównuje się numery sektora.
- Gdy wynik porównania jest ujemny - sprawdza się stan licznika obrotów. Jeżeli nie było dwóch obrotów, przystępuje się do ponownego szukania AM. Jeżeli zostały wykonane dwa obroty, sygnał "sektor nie odnaleziony" przesyła się do minikomputera z sygnałem lub bez sygnału anomalii CRC.
- Gdy wynik porównania jest dodatni a CRC zły - wysyła się sygnał anomalii CRC.
- Gdy wynik porównania jest dobry i CRC jest dobry może odbyć się detekcja rozkazu.

### 1.3.3.3. Detekcja rozkazu /adres 70/.

Wejście rozkazu poprzedzone jest sprawdzeniem stanu rejestru flag 5:

- stan 1 odpowiada kontroli po zapisie,
- stan 0 zezwala na wejście rozkazów zapisu - odczytu.



## 1.3.3.4. Odczyt sektora /adres 17/.

Operacja odczytu rozpoczyna się od procedury "stałej czasowej", która pozwala na uruchomienie odczytu mniej więcej 3 bajty przed początkiem danych /DM/, unikając znajdujących się wcześniej zaburzeń. Po procedurze "stałej czasowej" następuje detekcja bajtu początku danych DM, a więc bajtu danych FB i bajtu zegara 07, gdy sektor jest *normalny* - oraz bajtu danych FB z bajtem zegara 07, gdy sektor jest anulowany.

Jeżeli następuje detekcja bajtu FEC7 - do minikomputera wysłany zostaje sygnał "sektor nieodnaleziony"

Jeżeli następuje detekcja bajtu F807 /sektor anulowany/ w rejestr R6 wpisuje się 00, aby móc przeprowadzić test odnośnie rodzaju sektora. W tym przypadku - jak również w wypadku detekcji bajtu /FBC7/ wykryty DM jest zapisany pod adresem 130 pamięci. Po tej operacji następuje odczyt 128 bajtów danych oraz ich zapis w pamięci, a następnie procedura obliczania i sprawdzania CRC. Po przeprowadzeniu testu odnośnie rodzaju sektora /*normalny*, czy anulowany/ - sygnał anomalii i rodzaju sektora zostają przekazane do minikomputera.

## 1.3.3.5. Zapis sektora /adres 51/.

Operacja zapisu / z kontrolą/ ustawia "flagę 5" na "1" i po zapisie prowadzi do ponownego wyszukania początku sektora, /przed odczytem danych/. Ochronny odstęp 11 bajtów - jest realizowany jak poprzednio procedurą "stałej czasowej". Po nim następuje zapis 6-ciu bajtów 00 synchronizacji, wymagający użycia licznika bajtów. Po zapisaniu 6 bajtów 00 następuje pozwolenie na zapis bajtu początku danych DM /FBC7/, względnie /FB, 07/ - zależnie od zawartości bufora. 128 bajtów danych i 2 bajty CRC są zapisane w ten sam sposób, aż licznik adresu wskaże 254.

Operacja zapisu kończy się z początkiem odstępu końcowego G2 wstrzymaniem sygnału EL; sygnały EXS i VE /kasowanie tunelowe/ - są utrzymane przez 640 usek /procedura stałej czasowej/ po zaprzestaniu zapisu



Test określa czy chodzi o zapis zwykły, czy o zapis kontrolowany.

Przy zwykłym przystępuje się już do pętli końcowej, przy kontrolowanym szuka się AM w celu odczytu właśnie zapisanego sektora, wg opisanej powyżej procedury.

#### 1.4. Mikro-podprogramy użytkowe.

Różne podprogramy stosowane są w mikroprogramie - ażeby zminimalizować ilość instrukcji.

##### 1.4.1. Procedura obliczania CRC /adres 8E/.

Procedura obliczania CRC stosuje się do jednego bajtu. Rejestr służący jako łącznik bitów ustawia się na 8. Bajt jest wyczytywany z komórki pamięci, wskazanej przez licznik adresów. Każdy bit danych jest przekazywany do rejestru CRC w ślad za bitem zegarowym. Operacja trwa tak długo, aż licznik bitów osiągnie stan 0 - co pozwala na powtórzenie procedury.

##### 1.4.2. Procedura sprawdzania CRC /adres D4/.

Bardziej znaczący bajt CRC, uzyskany w powyższy sposób jest sprawdzany. Jeśli wynik się zgadza - odbywa się sprawdzenie drugiego bajtu. Gdy jedno z tych sprawdzeń nie zgadza się - to błąd CRC zapisany jest do rejestru R5. Sygnały zapoczątkowania CRC i "wykonywania" selektora EXS zostają wpisane do rejestru wyjściowego. "Zapoczątkowanie" CRC jest kasowane, a EXS wstrzymane przed powtórzeniem procedury.

##### 1.4.3. Procedura stałej czasowej / adres 06/.

Wartość stałej czasowej, uprzednio wpisanej do rejestru jest dekrementowana przy każdym WTCLK. Stan licznika testuje się czy jest już zero, czy nie - przed każdym WTCLK.

##### 1.4.4. Pętla końcowa /adres A7/.

Linia przerwania EX zostaje odsłonięta. Sygnał "koniec wykonywania" jest przekazany do minikomputera poprzez rejestr wyjścia. ~~Pojawienie się przerwania EX~~ powoduje



maskowanie wszystkich linii przerwania /adres 09/,  
następnie odbywa się powrót po przerwaniu RIN, po którym  
następuje nowe "zapoczątkowanie".

1.4.5. Przerwanie indeksowe IM /adres 0A/.

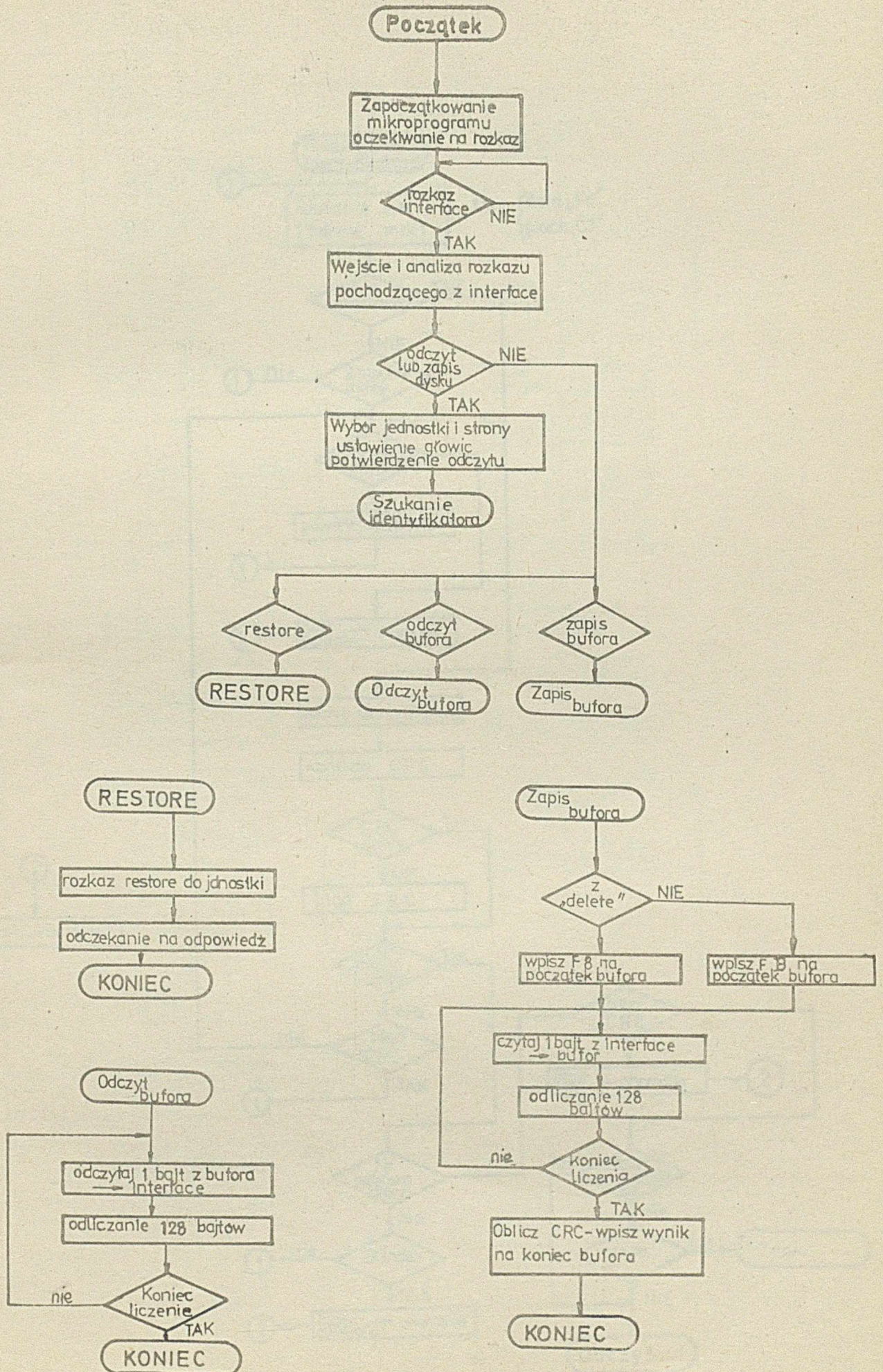
Licznik obrotów jest dekrementowany o 1.

Linia przerwania jest odsłonięta aż do powrotu po  
przerwaniu.

1.4.6. Odczyt jednego bajtu /adres 43/.

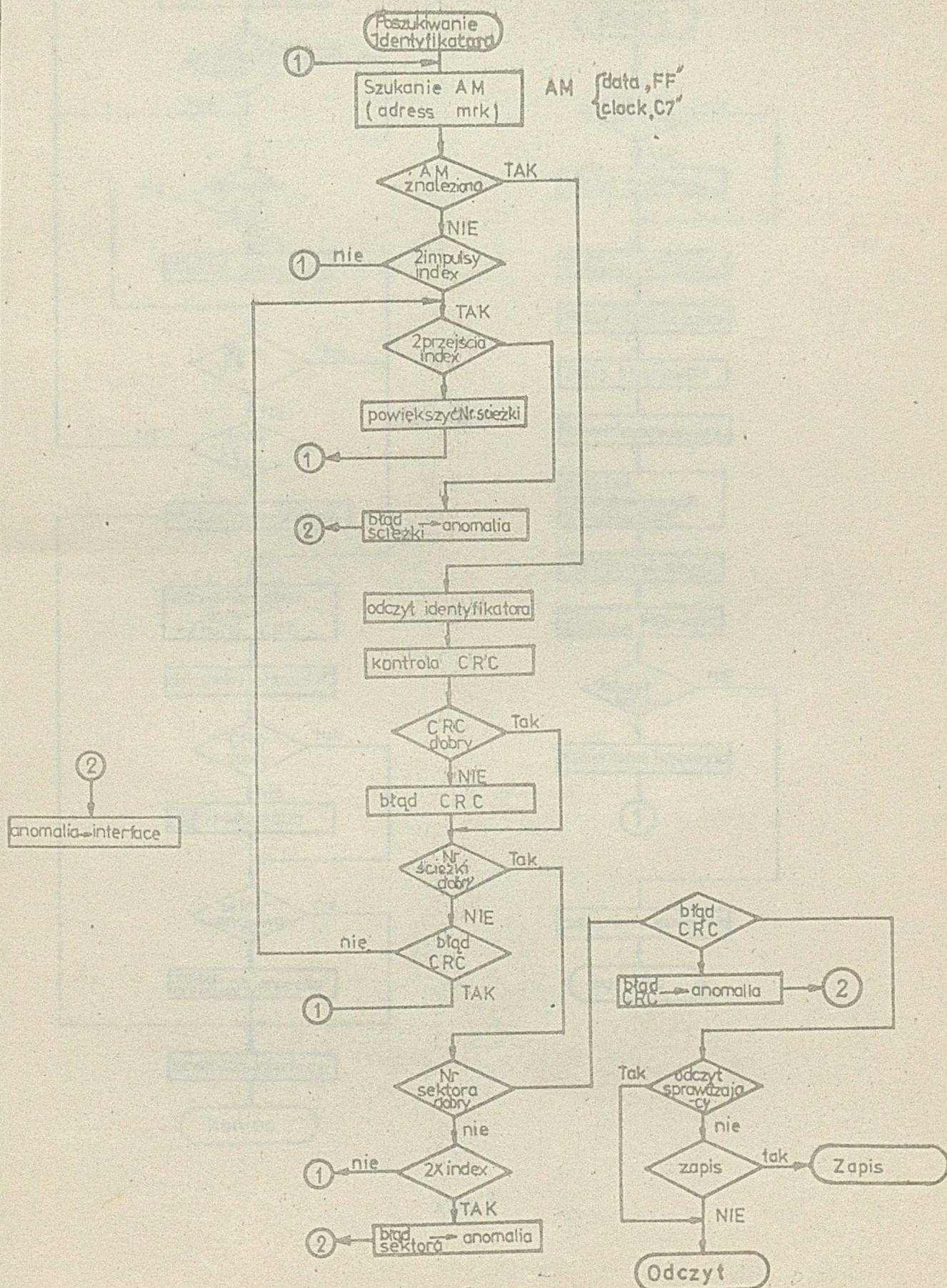
Rejestr odczytu R1 jest ustawiony na 0, a rejestr R2  
bajtu odczytanego na 7. Przy każdym RDCLK i RDCLK jeden  
bit z rejestru wejścia IPR5 - jest wpisywany do rejestru  
odczytu. Potem następuje przesunięcie o jedno miejsce  
w lewo i dekrementacja o 1 rejestru bajtu odczytanego.  
Ten cykl powtarza się aż do skompletowania pełnego bajtu.





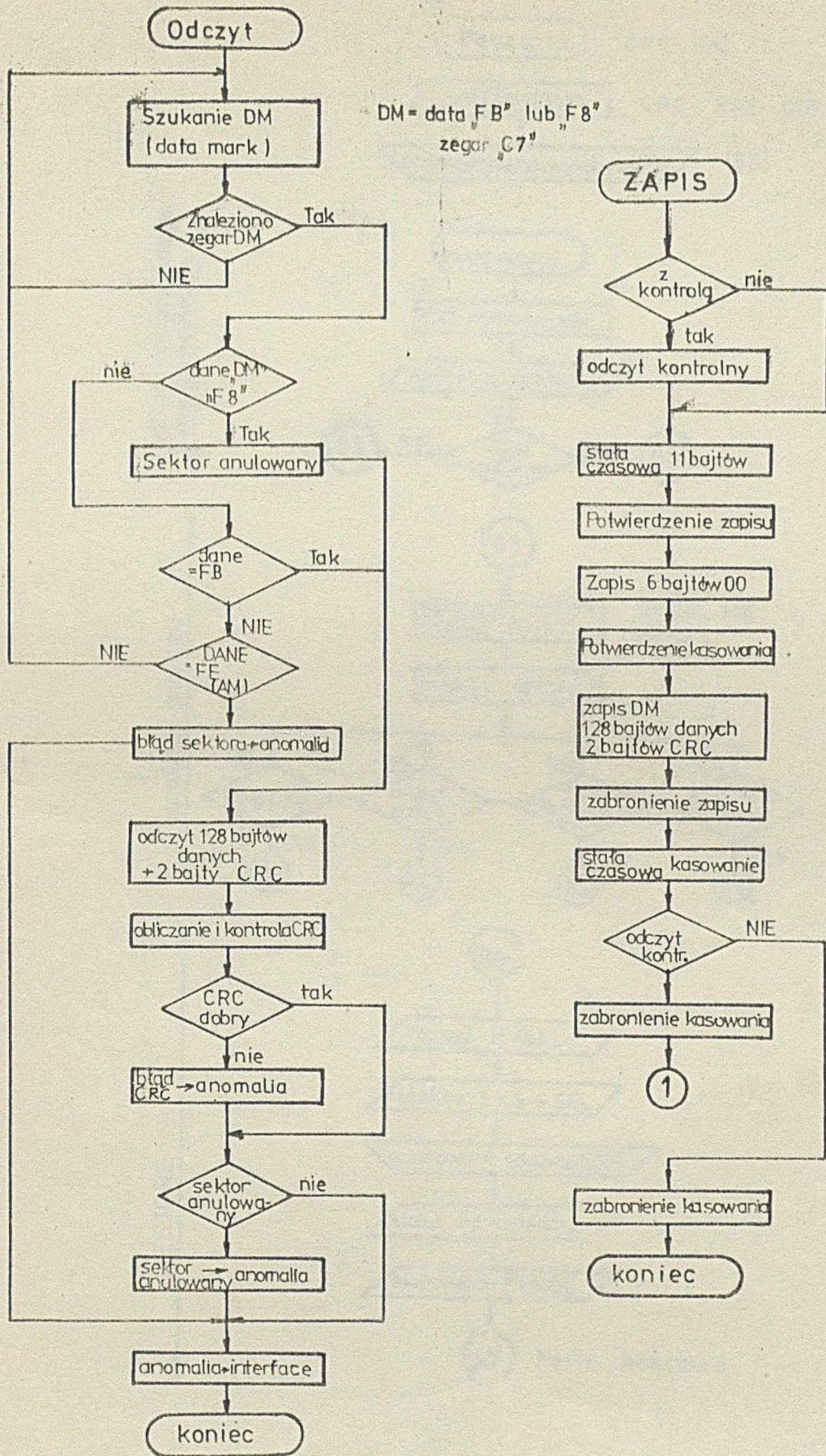
Rys.1





PRZ 2





Rys. 3

91



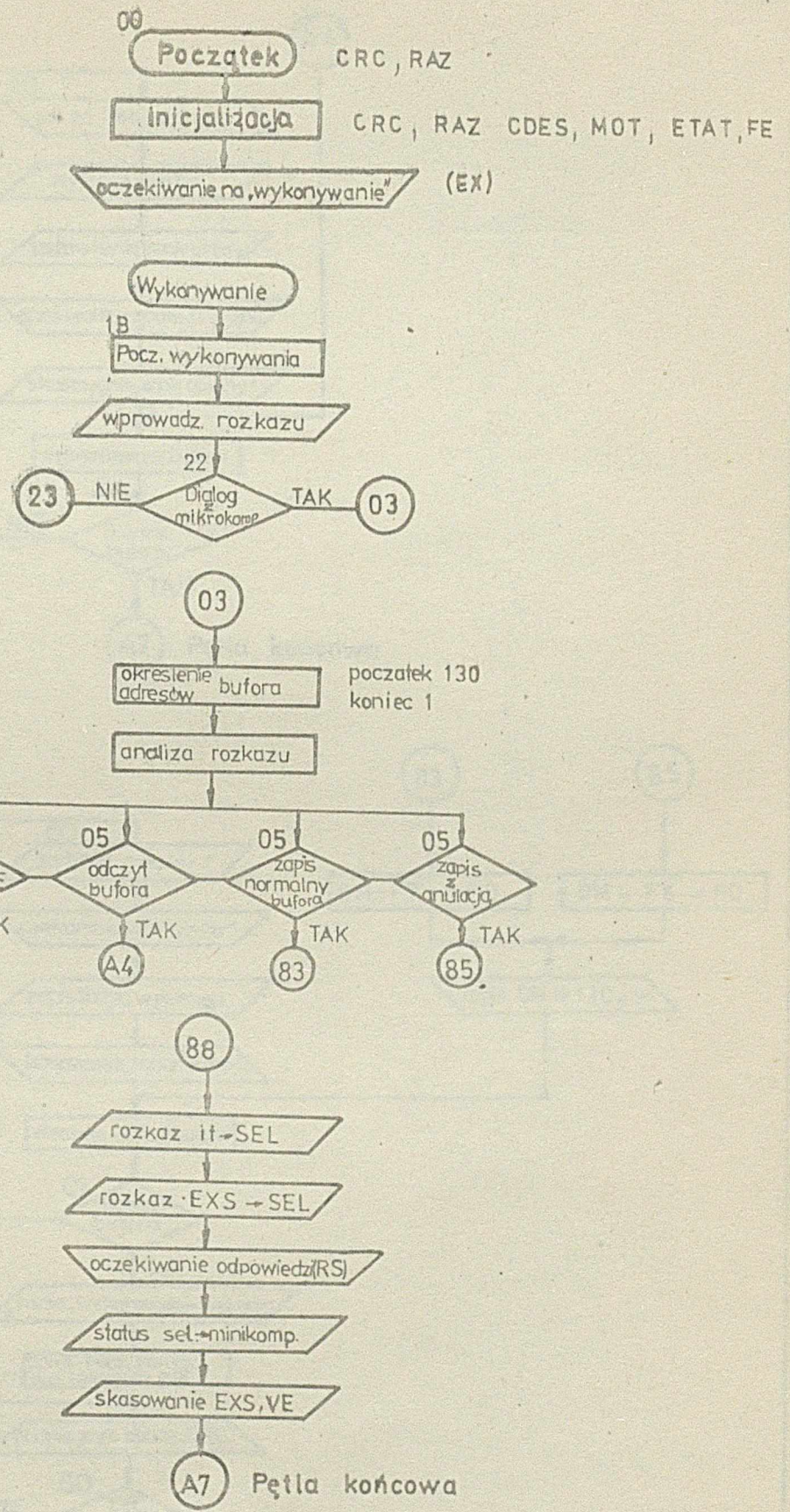
# ORGANIGRAMY SZCZEGÓŁOWE

11/26  
Tom I cz IV

analiza rodzaju dialogu z minikomputerem

analiza rozkazu

RESTORE

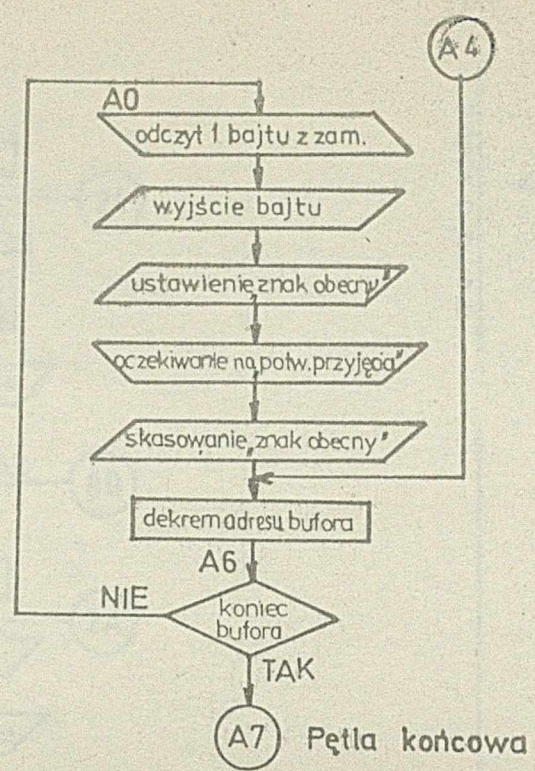


Inicjalizacja i „restore”  
Rys. 4

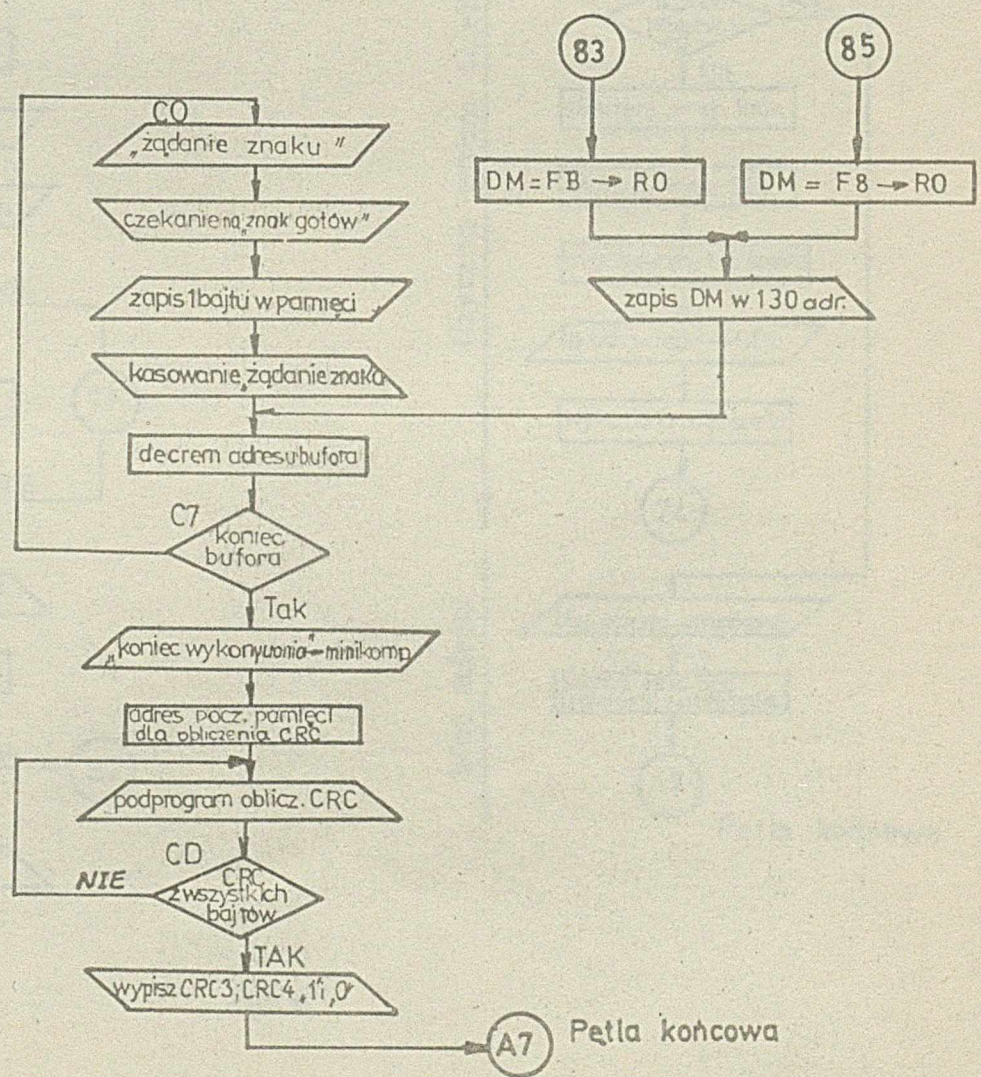
92



Odczyt bufora



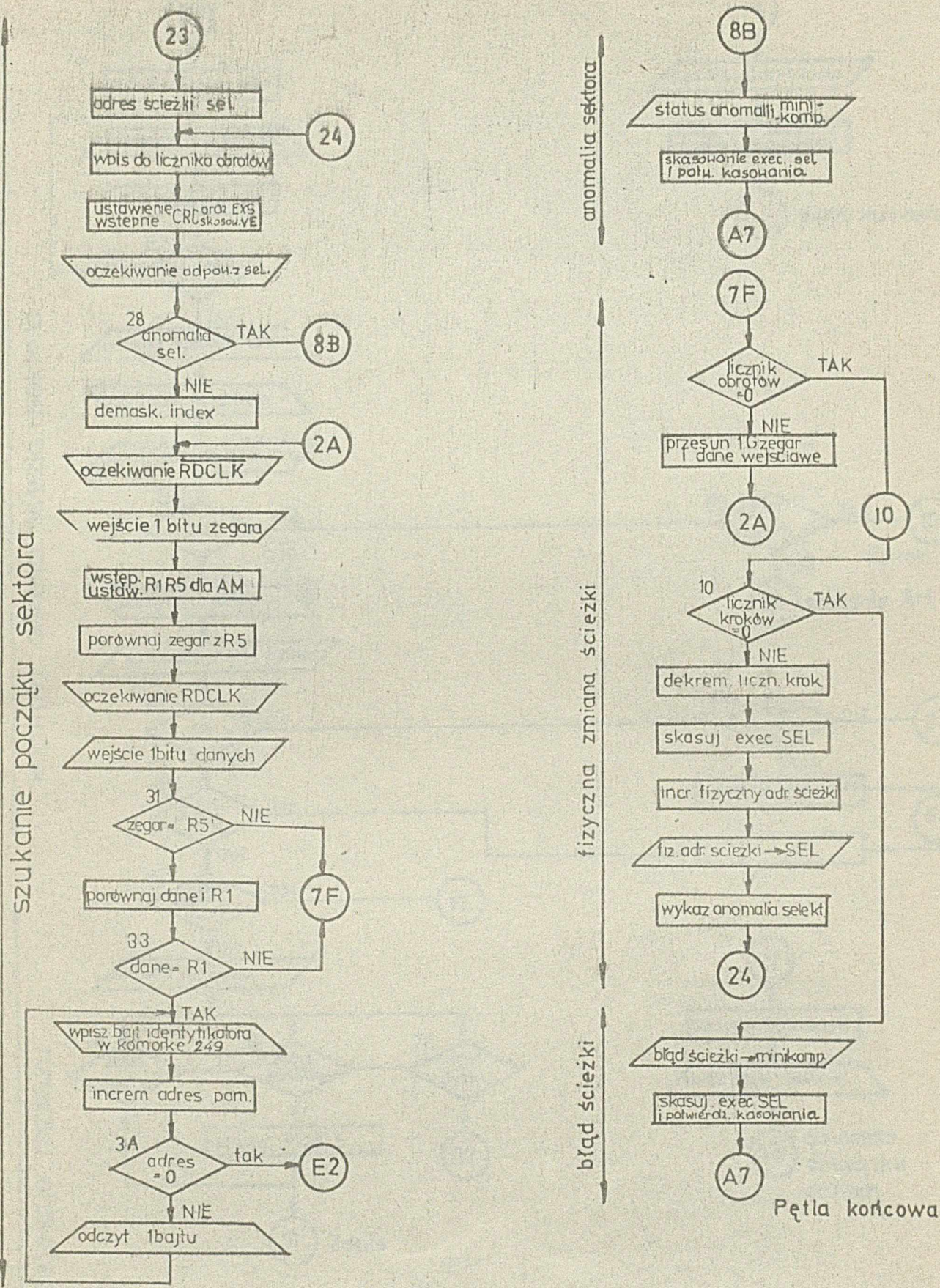
Zapis bufora



Odczyt bufora  
Zapis bufora  
Rys. 5

93

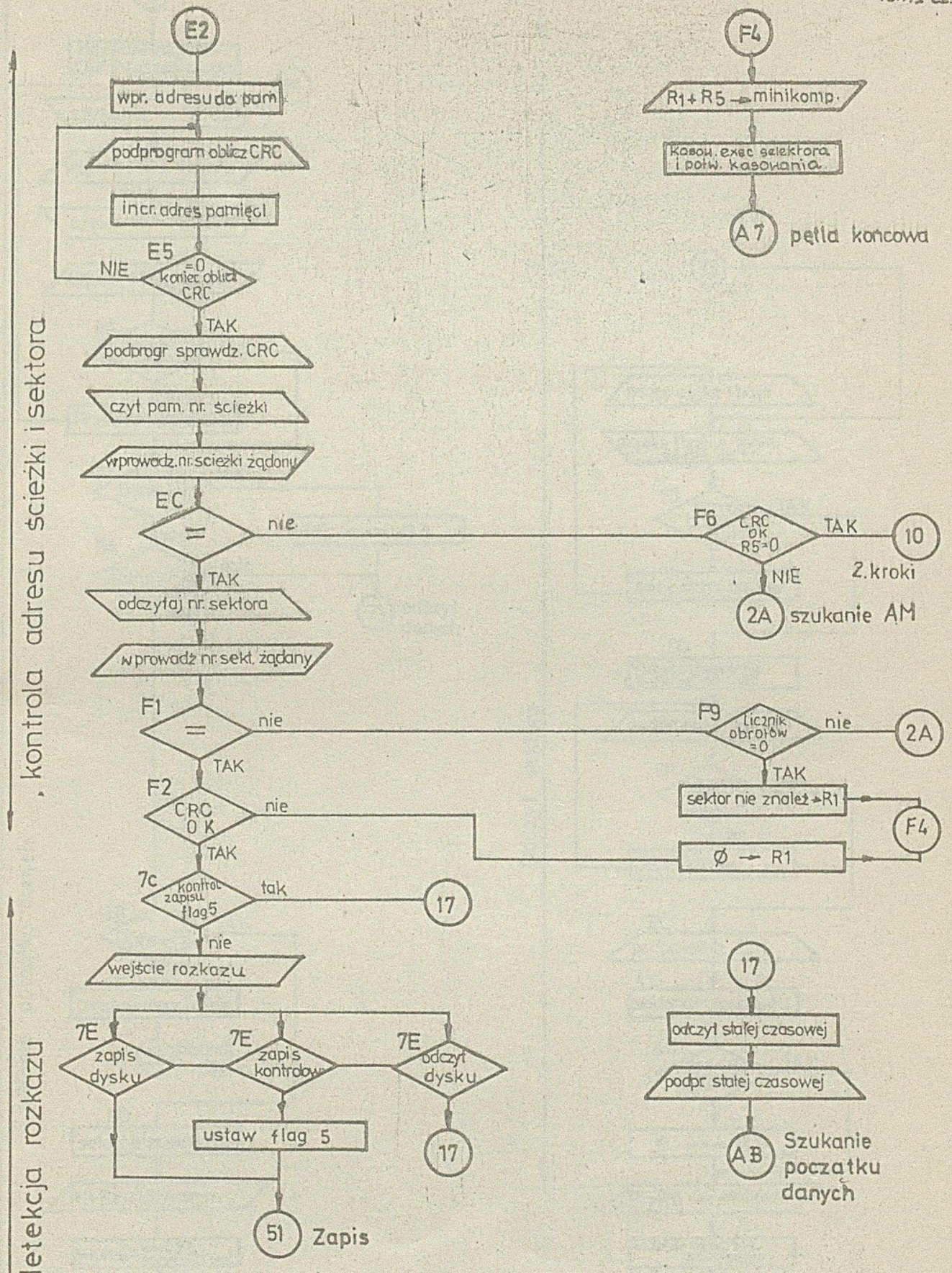




Rys. 6

Pętla końcowa



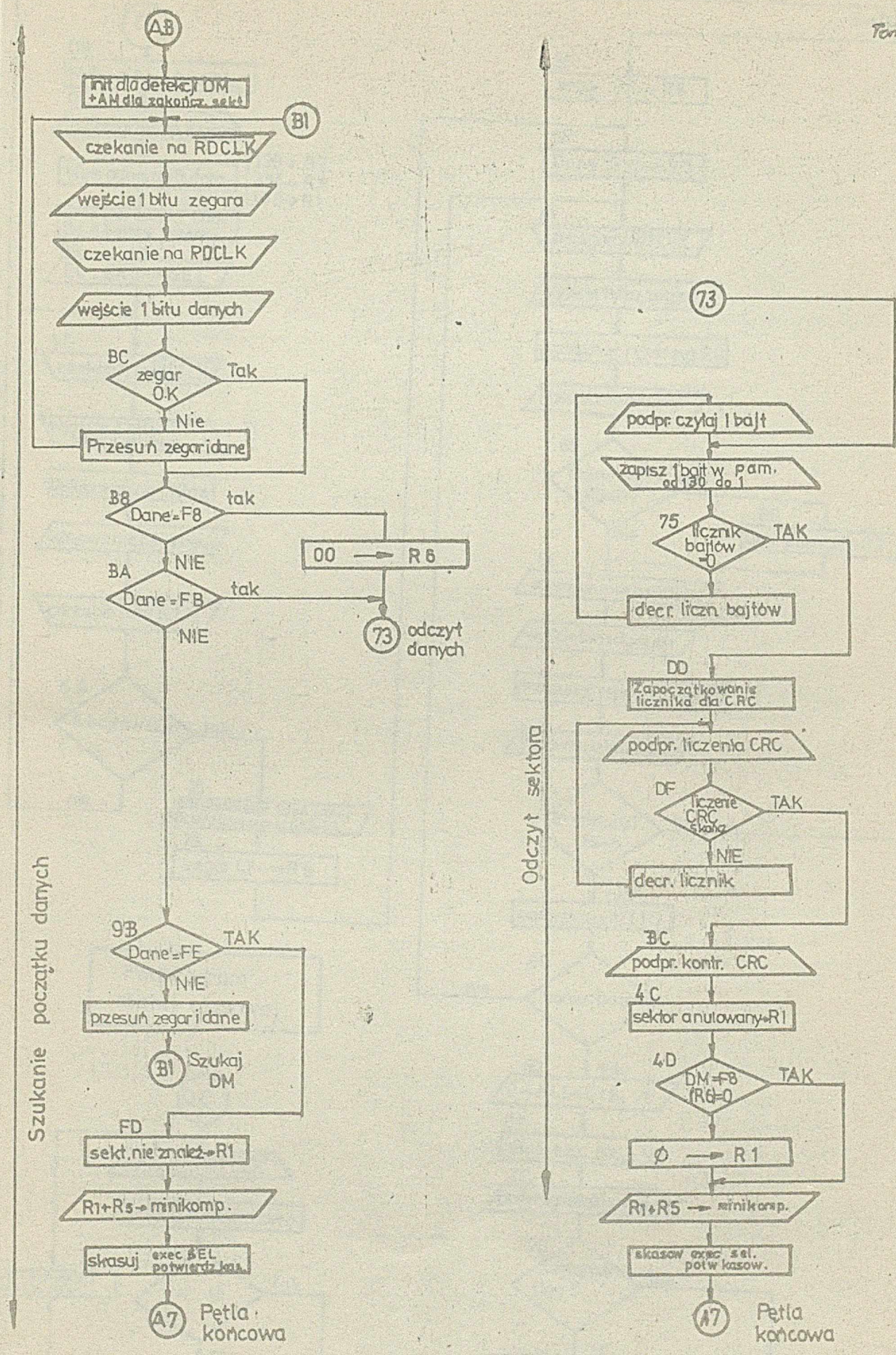


kontrola adresu sciezki i sektora

detekcja rozkazu

Rys.7





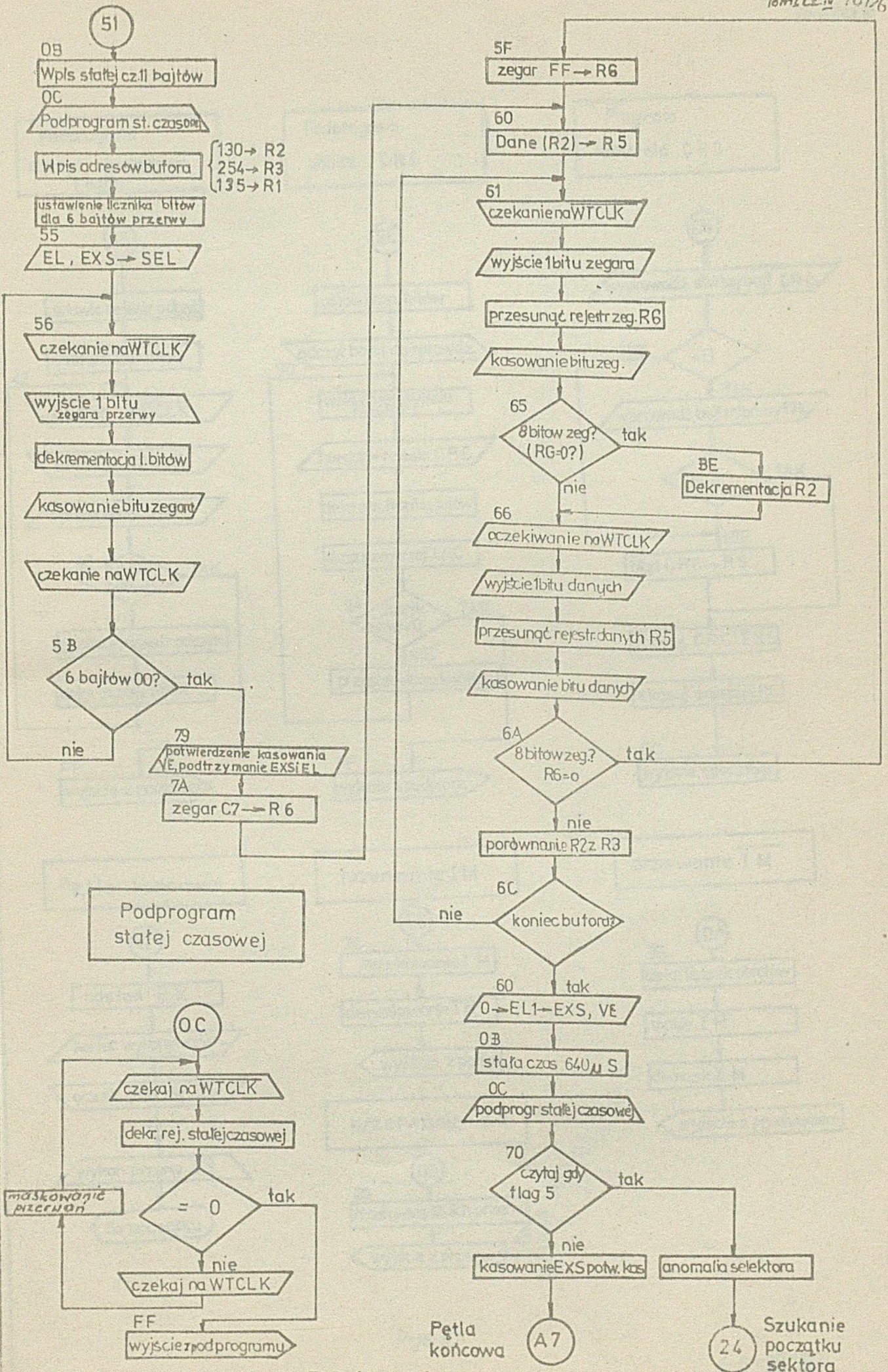
Szukanie początku danych  
Odczyt sektora

Rys.8

96

G-36760





Rys. 9

Zapis sektora

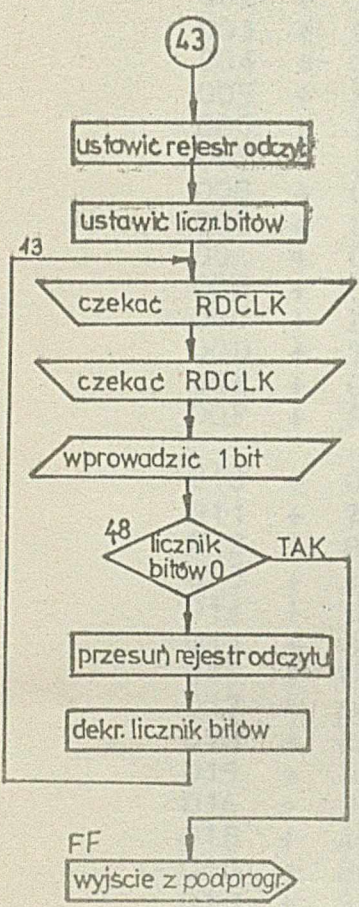
4-6-36760

Pętla końcowa

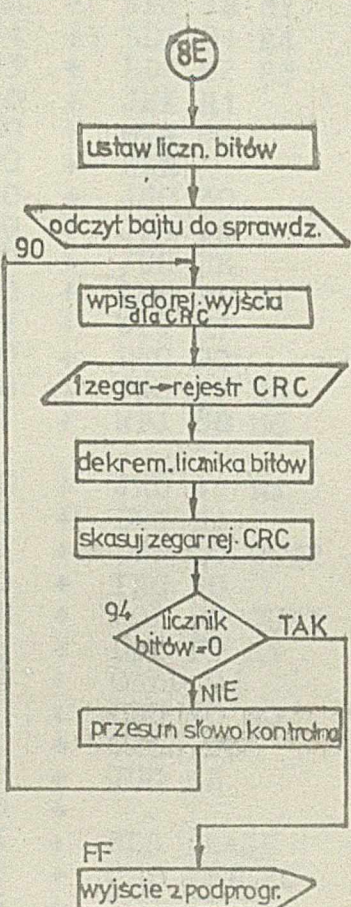
Szukanie początku sektora



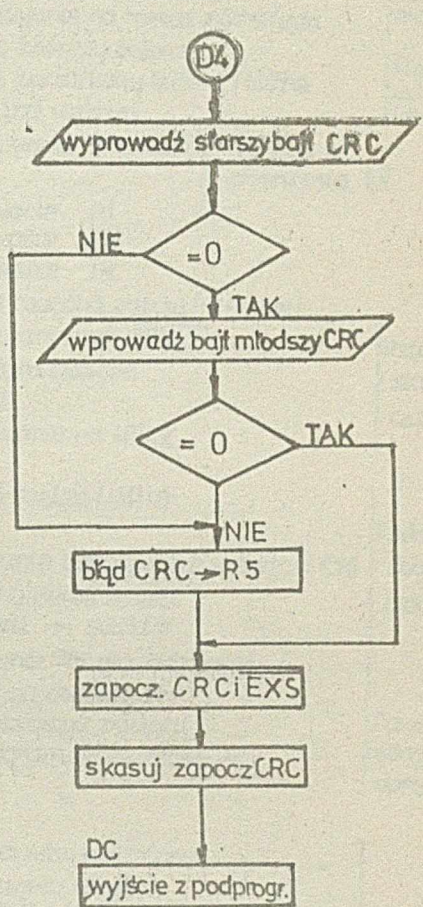
Podprogram  
odczyt szeregowy  
1 bajtu



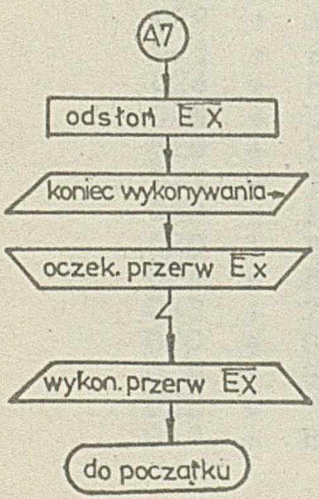
Podprogram  
oblicz CRC



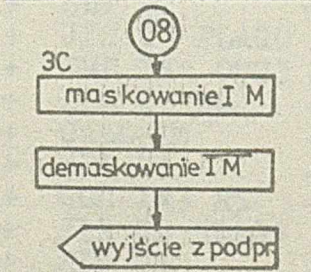
Program  
kontrola CRC



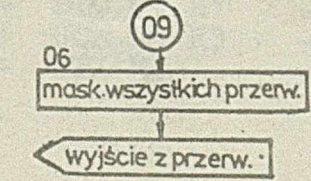
Pętla końcowa



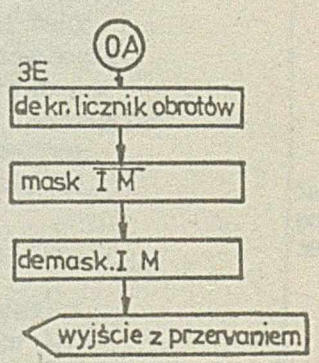
przerwanie I M



przerwanie EX



przerwanie I M



Rys. 10



## 2. Listing mikroprogramu.

18/26  
Tom I Cz. IV

Adres ROM	Kod 16	Opis symboliczny	Uwagi
000	+ 880A	+ LDC 80 OUT2	- Zerowanie sygnatów i inicjacja CRC
001	+ 800D	+ LDC 00 OUT5	- Zerowanie rejestru danych
002	+ F8DA	+ WPJ 1B M2	- Oczekiwanie na rozkaz z interfejsu
003	+ 8014	+ LDC 01 R4	- Adres końcowy buford
004	+ 8823	+ LDC 82 R3	- Adres początkowy buford (130)10
005	+ E010	+ JEX R1	- (R1) - kod rozkazu
006	+ B400	+ MIN 00	- Maskowanie przerwani
007	+ D800	+ RIN	- Obsługa przerwania EX
008	+ C9E0	+ JUC 3C	- Przerwanie IM
009	+ C830	+ JUC 06	- Przerwanie EX
00A	+ C9F0	+ JUC 3E	- Przerwanie IM
00B	+ 83D0	+ LDC 3D RO	- Stała czasowa zapisu (11 bajtów)
00C	+ B007	+ WPI M7	- Oczekiwanie na WTCLK
00D	+ 7800	+ DEC RO	- (RO) stała czasowa
00E	+ C7F8	+ JRO FF RO	- Powrót
00F	+ FAEF	+ WPJ 5D M6	- Oczekiwanie na WTCLK
010	+ C20C	+ JRO 41 R4	- Licznik kroków (R4)·0
011	+ 7804	+ DEC R4	-
012	+ 800A	+ LDC 00 OUT2	- Kasowanie wykonywania selektora EXS
013	+ 4000	+ INC RO	- Fizyczny adres ściezki
014	+ 540B	+ COP RO OUT3	- Nr ściezki → selektor
015	+ 8FF5	+ LDC FF R5	- Ustawienie R5 dla ANO selektora
016	+ C927	+ CALL 24	- Powrót po detekcji AM
017	+ 8500	+ LDC 50 RO	- Stała czasowa odczytu
018	+ C861	+ CALL 0C	- Podprogram stałej czasowej
019	+ CD58	+ JUC AB	-
01A	+ 0000	+	-
01B	+ A005	+ RES FLAG5	- Flaga odczytu kontrolnego
01C	+ 2C90	+ CMP IN1 RO	- Wpis numeru ściezki
01D	+ 8FF5	+ LDC FF R5	- Wpis R5 dla ANO selektora
01E	+ 8024	+ LDC 02 R4	- Wpis do licznika kroków
01F	+ 2CE2	+ CMP IN6 R2	- Wprowadzenie rozkazu z interfejsu
020	+ 2CE1	+ CMP IN6 R1	- Rozkaz do przechowania
021	+ 6002	+ SHF R2	- Zgrubna analiza rozkazu
022	+ F018	+ JIS 03 FLAG0	- Dialog interfejs- bufor
023	+ 2C9B	+ CMP IN1 OUT3	- Pierwsze szukanie ściezki
024	+ 8036	+ LDC 03 R6	- Wpis do licznika obrotów
025	+ CED1	+ CALL DA	- Inicjacja CRC i podanie EXS
026	+ B003	+ WPI M3	- Odpowiedz selektora
027	+ 70B5	+ COM IN3 R5	- ANOMalia selektora!
028	+ EC59	+ JIR 8B FLAG1	-
029	+ B401	+ MIN 01	- Demaskowanie IM
02A	+ B005	+ WPI M5	- RDCLK
02B	+ 74C3	+ ORL IN4 R3	- Bit zegara V (R3)
02C	+ 8385	+ LDC 38 R5	- 38·C7 - zegar AM
02D	+ 8011	+ LDC 01 R1	- 01·FE - dane AM
02E	+ 7053	+ COM R5 R3	- Bajt zegara = (R5)!
02F	+ B004	+ WPI M4	- RDCLK



Adres ROM	Kod 16	Opis symboliczny	Uwagi
030	+ 74D2	+ ORL IN5 R2	- Bit danych V (R2)
031	+ EBF9	+ JIR 7F FLAG1	- Zegar odczytany ≠ (R5)
032	+ 7012	+ COM R1 R2	- Bajt danych = (R1)?
033	+ EBF9	+ JIR 7F FLAG1	- Dane odczytane ≠ (R1)
034	+ 8F93	+ LDC F9 R3	- Początek adr.pamięci na 249
035	+ 2011	+ CMP R1 R1	- (RT) → R1 (AM)
036	+ 9431	+ WTM R3 R1	- AM → RAM
037	+ 4003	+ INC R3	- Adres RAM
038	+ 0713	+ JRO E2 R3	- Identyfikator odczytany
039	+ CA19	+ CALL 43	- Odczyt szeregowy 1 bajtu
03A	+ C9A8	+ JUC 35	
03B	+ 0000	+	
03C	+ B404	+ MIN 04	- Demaskowanie $\overline{IM}$
03D	+ D800	+ RIN	- Licznik obrotów
03E	+ 7806	+ DEC R6	- Demaskowanie IM
03F	+ B401	+ MIN 01	
040	+ D800	+ RIN	
041	+ 802D	+ LDC 02 OUT5	- Błąd ścieżki → interf.
042	+ CC60	+ JUC 80	
043	+ 8001	+ LDC 00 R1	- Wpis do rejestru odczytu
044	+ 8072	+ LDC 07 R2	- Licznik bitów
045	+ B005	+ WPI M5	- RDCLK
046	+ B004	+ WPI M4	- RDCLK
047	+ 74D1	+ ORL IN5 R1	- Wprowadzenie 1 bitu
048	+ C7FA	+ JRO FF R2	- Bajt odczytano
049	+ 6001	+ SHF R1	- Rejestr odczytu
04A	+ 7802	+ DEC R2	- Licznik bitów
04B	+ CA28	+ JUC 45	
04C	+ 8401	+ LDC 40 R1	- Sektor anulowany → R1
04D	+ C7A6	+ JRO F4 R6	- DM odczytany = F8
04E	+ CF98	+ JUC F3	
04F	+ 0000	+	
050	+ A405	+ SRS FLAG5	- Flaga odczytu kontrolnego
051	+ C859	+ CALL 0B	- Podprogram stałej czasowej
052	+ 8822	+ LDC 82 R2	- Adres początkowy bufora (130)
053	+ 8FE3	+ LDC FE R3	- Adres końcowy bufora (254)
054	+ 8301	+ LDC 30 R1	- Licznik bitów dla 6 bajtów 00
055	+ 818A	+ LDC 18 OUT2	- EL, EXS → SEL
056	+ B007	+ WPI M7	- WTCLK
057	+ 8809	+ LDC 80 OUT1	- Wyjście 1 bitu zegara (FF)
058	+ 7801	+ DEC R1	- Licznik bitów
059	+ 8009	+ LDC 00 OUT1	- Kasowanie zegara
05A	+ B006	+ WPI M6	- WTCLK
05B	+ C3C9	+ JRO 79 R1	- 6 bajtów 00 zapisane
05C	+ CAB0	+ JUC 56	
05D	+ B400	+ MIN 00	- Maskowanie przerwań
05E	+ C860	+ JUC 0C	
05F	+ 8FF6	+ LDC FF R6	- Zegar danych (FF) → R6

Szukanie początku sektora

Obsługa przerwań IM,  $\overline{IM}$

Szukanie początku sektora

Odczyt szeregowy jednego bajtu

Odczyt sektora

Zapis sektora



Adres ROM	Kod 16	Opis symboliczny	Uwagi	
060	+	9025 + RFM R2 R5	- Pobranie bajtu danych	Zapis sektora
061	+	B007 + WPI M7	- WTCLK	
062	+	5469 + COP R6 OUT1	- Wujście 1 bitu zegara	
063	+	6006 + SHF R6	- Zegar	
064	+	8009 + LDC 00 OUT1	- Kasowanie bitu zegara	
065	+	05F6 + JRO BE R6	- Gdy było 8 bitów zegara	
066	+	B006 + WPI M6	- WTCLK	
067	+	5459 + COP R5 OUT1	- Wujście 1 bitu danych	
068	+	6005 + SHF R5	- Dane	
069	+	8009 + LDC 00 OUT1	- Kasowanie bitu danych	
06A	+	02FE + JRO 5F R6	- Gdy było 8 bitów zegara	
06B	+	7023 + COM R2 R3	- (R3) adr. koncowy bufora	
06C	+	EBO9 + JIR 61 FLAG1	- Koniec bufora?	
06D	+	828A + LDC 28 OUT2	- Skasuj EL, zostaw EXSLVE	
06E	+	8A00 + LDC A0 RO	- Stała czasowa 640 μsek.	
06F	+	C861 + CALL 0C	- Podprogram stałej czasowej	
070	+	FOAD + JIS 15 FLAG5	- Odczyt kontrolny	Odczyt sektora
071	+	CC60 + JUC 8C		
072	+	8810 + LDC 81 RO	- NC	
073	+	2C11 + CMP R1 R1	- (R1) → R1	
074	+	9401 + WTM RO R1	- Bajt danych → RAM	
075	+	C6E8 + JRO DD RO	- Licznik bajtów = 0?	
076	+	7800 + DEC RO	- Adres RAM (& licznik bajtów)	
077	+	CA19 + CALL 43	- Podprogram odczytu 1 bajtu	
078	+	CB98 + JUC 73		
079	+	838A + LDC 38 OUT2	- VL, EL, EXS → SEL	
07A	+	8C76 + LDC 07 R6	- Zegar DM (C7) → R6	Zapis sektora
07B	+	0B00 + JUC 60		Detekcja rozkazu
07C	+	FOBD + JIS 17 FLAG5	- Kontrola zapisu?	
07D	+	2CE0 + CMP IN6 RO	- Wprowadzenie rozkazu z interfejsu	
07E	+	E000 + JEX RO		
07F	+	C086 + JRO 10 R6	- Licznik obrotów = 0	Szukanie początku sektora
080	+	6002 + SHF R2	- Dane	Zapis bufora
081	+	6003 + SHF R3	- Zegar	
082	+	0950 + JUC 2A	- Nowe poszukiwanie AM	
083	+	8FBO + LDC FB RO	- DM = FB	
084	+	CC30 + JUC 86		
085	+	8F80 + LDC F8 RO	- DM = F8	
086	+	9430 + WTM R3 RO	- Wpis DM do RAM	
087	+	CE28 + JUC 05	- Zapis bufora cd	
088	+	2CEB + CMP IN6 OUT3	- Rozkaz „restore” do selektora	
089	+	808A + LDC 08 OUT2	- Podanie EXS	
08A	+	B003 + WPI M3	- Odpowiedź selektora	Restore
08B	+	2CBD + CMP IN3 OUT5	- AND selektora → interfejs	
08C	+	800A + LDC 00 OUT2	- Kasowanie EXS	
08D	+	CD38 + JUC A7		
08E	+	8082 + LDC 08 R2	- Licznik bitów	Liczenie CRC
08F	+	9031 + RFM R3 R1	- Bajt do sprawdzenia	



Adres ROM	Kod 16	Opis symboliczny	Uwagi
090	+ 541C	+ COP R1 OUT4	- 1 bit dla rej. CRC
091	+ 8409	+ LDC 40 OUT1	- Podanie zeg. CRC
092	+ 7802	+ DEC R2	- Licznik bitów
093	+ 8009	+ LDC 00 OUT1	- Kasowanie zeg. CRC
094	+ C7FA	+ JRO FF R2	
095	+ 6001	+ SHF R1	- Bajt sprawdzony
096	+ CC80	+ JUC 90	
097	+ 8006	+ LDC 00 R6	- Sektor anulowany
098	+ CB98	+ JUC 73	
099	+ 0000	+	
09A	+ 0000	+	
09B	+ 7061	+ COM R6 R1	- (R6)-01=FE
09C	+ F7E9	+ JIS FD FLAG1	- Gdy dane = FE
09D	+ 6003	+ SHF R3	- Zegar
09E	+ 6001	+ SHF R1	- Dane
09F	+ CD88	+ JUC B1	- Szukanie DM
OA0	+ 903D	+ RFM R3 OUT5	- Pobranie bajtu z RAM
OA1	+ 802A	+ LDC 02 OUT2	- Obecność znaku
OA2	+ B001	+ WPI M1	- Potwierdzenie przyjęcia znaku
OA3	+ 800A	+ LDC 00 OUT2	- Skasowanie obecności znaku
OA4	+ 7803	+ DEC R3	- Adres bufora
OA5	+ 7043	+ COM R4 R3	- (R4) adres końcowy bufora
OA6	+ ED01	+ JIR A0 FLAG1	- Koniec bufora
OA7	+ B402	+ MIN 02	- Demaskowanie EX
OA8	+ 804A	+ LDC 04 OUT2	- Koniec wykonywania interfejsu
OA9	+ 1000	+ STP	- Oczekiwanie na EX
OAA	+ C800	+ JUC 00	- Początek programu
OAB	+ 8074	+ LDC 07 R4	- 07 = FB - DM sektora anulowanego
OAC	+ 8003	+ LDC 00 R3	- Rejestr zegara
OAD	+ 8820	+ LDC 82 R0	- Adres RAM 130 dla upisu DM
OAE	+ 8016	+ LDC 01 R6	- 01 = FE - dane AM
OAF	+ 8047	+ LDC 04 R7	- 04 = FB - dane DM
OB0	+ 8382	+ LDC 38 R2	- 38 = C7 - zegar AM i DM
OB1	+ B005	+ WPI M5	- RDCLK
OB2	+ 74C3	+ ORL IN4 R3	- Wejście bitu zegara
OB3	+ 7023	+ COM R2 R3	- Zegar = C7?
OB4	+ B004	+ WPI M4	- RDCLK
OB5	+ 74D1	+ ORL IN5 R1	- Wejście bitu danych
OB6	+ ECE9	+ JIR 9D FLAG1	- Zegar odczytany ≠ C7
OB7	+ 7041	+ COM R4 R1	- Dane = F8?
OB8	+ F4B9	+ JIS 97 FLAG1	- Dane odczytane ≠ 8
OB9	+ 7071	+ COM R7 R1	- Dane = FB?
OBA	+ F399	+ JIS 73 FLAG1	- Dane odczytane = FB
OBB	+ CCD8	+ JUC 9B	
OBC	+ CEA1	+ CALL D4	- Podpr. kontrola CRC
OBD	+ CA60	+ JUC 4C	
OBE	+ 7802	+ DEC R2	- Dekrementacja adresu bufora
OBF	+ CB30	+ JUC 66	

Podprogram  
obliczanie  
CRC

Odczyt  
danych

Odczyt  
danych

Odczyt  
bufora

Pałta  
końcowa

Szukanie  
początku  
danych

Zapis  
sektora



Adres ROM	Kod 16	Opis symboliczny	Uwagi
OC0	+ 801A	+ LDC 01 OUT2	- Ządanie znaku
OC1	+ B000	+ WPI MO	- Znak gotów
OC2	+ 2C81	+ CMP IN0 R1	- Wejście danych
OC3	+ 800A	+ LDC 00 OUT2	- Potwierdzenie przyjęcia znaku
OC4	+ 9431	+ WTM R3 R1	- Wpis bajtu do RAM
OC5	+ 7803	+ DEC R3	- Adres RAM
OC6	+ 7043	+ COM R4 R3	- (R4)adres końcowy bufora
OC7	+ EE01	+ JIR CO FLAG1	- Koniec bufora ?
OC8	+ 804A	+ LDC 04 OUT2	- Koniec wykonywania interfejs
OC9	+ 8823	+ LDC 82 R3	- Adres pamięci (130)
OCA	+ CC71	+ CALL 8E	- Podpr. obliczania CRC
OCB	+ 7803	+ DEC R3	- Adres RAM
OCC	+ 7043	+ COM R4 R3	- Adres końcowy bufora (R4)
OCD	+ EE51	+ JIR CA FLAG1	- Koniec bufora ?
OCE	+ 54B4	+ COP IN3 R4	- 1 bajt CRC
OCF	+ 9434	+ WTM R3 R4	- Wpis bajtu CRC do RAM
OD0	+ 7803	+ DEC R3	- Adres RAM
OD1	+ 54F4	+ COP IN7 R4	- 2 bajt CRC
OD2	+ 9434	+ WTM R3 R4	- Wpis bajtu CRC do RAM
OD3	+ CD38	+ JUC A7	- Delta końca
OD4	+ 54B5	+ COP IN3 R5	- Starszy bajt CRC
OD5	+ C6BD	+ JRO D7 R5	- =0?
OD6	+ CEC8	+ JUC D9	-
OD7	+ 54F5	+ COP IN7 R5	- Młodszy bajt CRC
OD8	+ C6D5	+ JRO DA R5	- =0?
OD9	+ 8085	+ LDC 08 R5	- Inicjacja CRC i podanie EXS
ODA	+ 888A	+ LDC 88 OUT2	- Kasowanie INITCRC zast. EXS
ODB	+ 808A	+ LDC 08 OUT2	-
ODC	+ D000	+ REJ	- Powrót (R5) - błąd CRC?
ODD	+ 8823	+ LDC 82 R3	- Adres początkowy RAM (130)
ODE	+ CC71	+ CALL 8E	- Podpr. obliczanie CRC
ODF	+ C5E3	+ JRO BC R3	- Koniec obl. CRC?
OE0	+ 7803	+ DEC R3	- Adres RAM
OE1	+ CEFO	+ JUC DE	-
OE2	+ 8F93	+ LDC F9 R3	- Adres pamięci na 249
OE3	+ CC71	+ CALL 8E	- Obliczanie CRC
OE4	+ 4003	+ INC R3	- Adres pamięci (RAM)
OE5	+ C73B	+ JRO E7 R3	- Koniec obl. CRC?
OE6	+ CF18	+ JUC E3	-
OE7	+ CEA1	+ CALL D4	- Podpr kontrola CRC
OE8	+ 8FA3	+ LDC FA R3	- Adres ścieżki (250) w RAM
OE9	+ 9031	+ RFM R3 R1	- Pobranie odczytanego nr ścieżki
OEA	+ 2C92	+ CMP IN1 R2	-
OEB	+ 7012	+ COM R1 R2	- Wprow. zadanego adresu ścieżki
OEC	+ EFB1	+ JIR F6 FLAG1	- =2
OED	+ 8FC3	+ LDC FC R3	- Adres sektora (251) w RAM
OEE	+ 9031	+ RFM R3 R1	- Pobranie odczytanego nr sektora
OEF	+ 2CA2	+ CMP IN2 R2	- Wprow. zadanego adresu sektora

zapis bufora

Podprogram kontrola CRC

Odczyt sektora

Kontrola adresu ścieżki i sektora



Adres ROM	Kod 16	Opis symboliczny	Uwagi
0F0	+ 7012	+ COM R1 R2	- Sektor ządany z odczytanym
0F1	+ EFC9	+ JIR F9 FLAG 1	- = ?
0F2	+ C3E5	+ JRO 7C R5	- CRC = 0 <sup>2</sup>
0F3	+ 8001	+ LDC 00 R1	- Status
0F4	+ 741D	+ ORL R1 OUT 5	- Status - interfejs [(R1) V (R5)]
0F5	+ CC60	+ JUC 8C	- 2 kroki gdy CRC OK
0F6	+ C085	+ JRO 10 R5	
0F7	+ C950	+ JUC 2A	- D1 - R2
0F8	+ CFA0	+ JUC F4	
0F9	+ 8012	+ LDC 01 R2	- Licznik obrotów = 1 <sup>2</sup>
0FA	+ 7026	+ COM R2 R6	- Zrobiono 2 obroty
0FB	+ F7E9	+ JIS FD FLAG 1	- Szukane AM
0FC	+ C950	+ JUC 2A	- Sektor nie znaleziony - R1
0FD	+ 8041	+ LDC 04 R1	
0FE	+ CFA0	+ JUC F4	
0FF	+ D000	+ REJ	
100	+ 0000	+	
101	+ 0000	+	
102	+ 0000	+	
103	+ 0000	+	
104	+ 0000	+	
105	+ 0000	+	
106	+ 0000	+	
107	+ 0000	+	
108	+ 0000	+	
109	+ 0000	+	
10A	+ 0000	+	
10B	+ 0000	+	
10C	+ 0000	+	
10D	+ 0000	+	
10E	+ 0000	+	
10F	+ 0000	+	
110	+ 0000	+	
111	+ 0000	+	
112	+ 0000	+	
113	+ 0000	+	
114	+ 0000	+	
115	+ 0000	+	
116	+ 0000	+	
117	+ 0000	+	
118	+ 0000	+	
119	+ 0000	+	
11A	+ 0000	+	
11B	+ 0000	+	
11C	+ 0000	+	
11D	+ 0000	+	
11E	+ 0000	+	
11F	+ 0000	+	

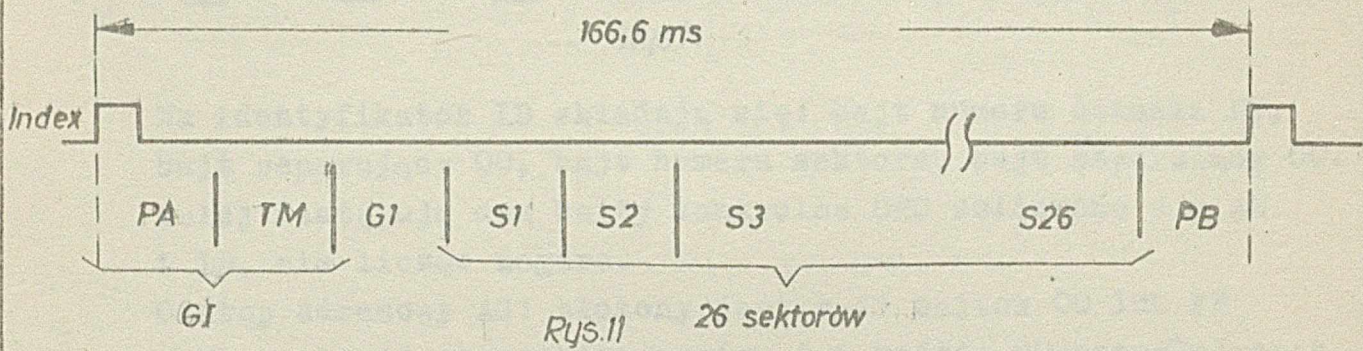
Kontrola  
adresu  
ścieżki  
i sektora



### 3. Format IBM

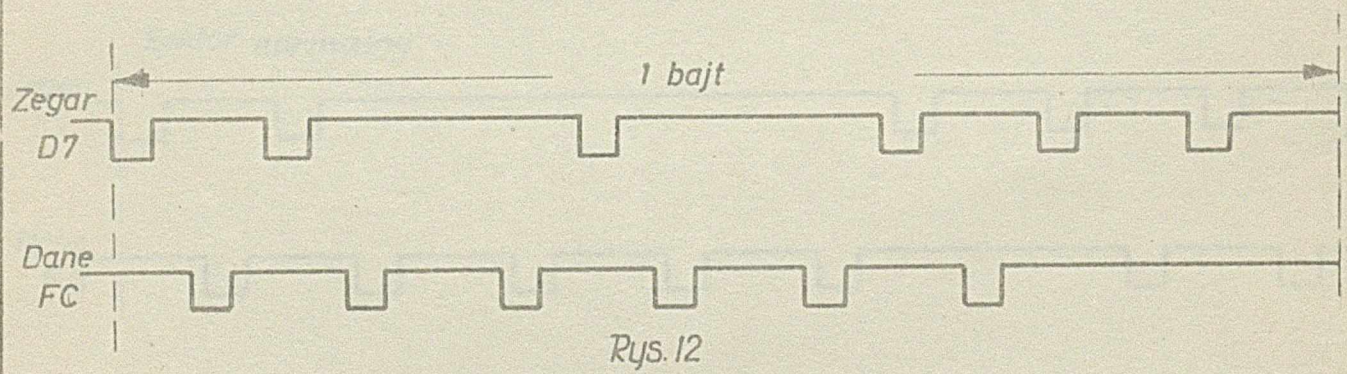
Dyski stosowane w PLx45D są sformatowane /wstępnie zapisane/ na specjalnej aparaturze. Format jest kompatybilny z IBM /norma ISO 97/11 N 149/.

Każdy dysk ma 77 ścieżek po 26 sektorów i obraca się z prędkością 360 obr/min. Sygnał "Index" pozwala określić początek ścieżki wstępnie opisany następująco:

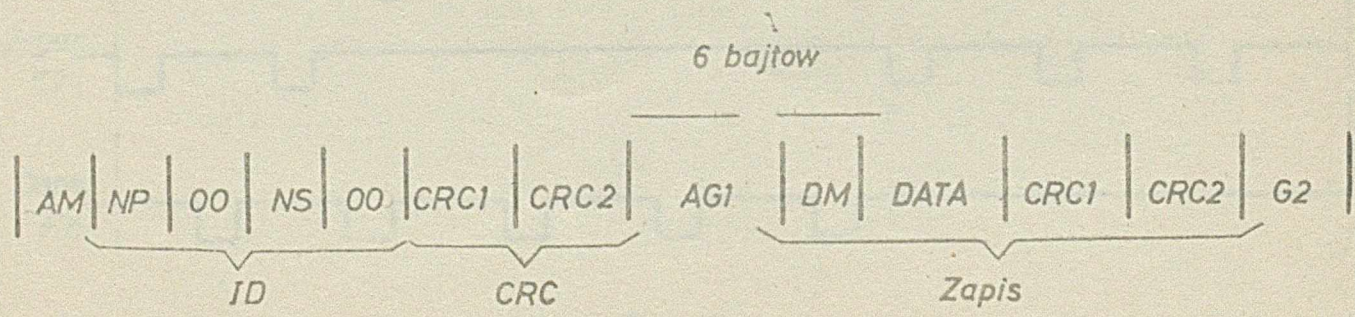


Format ścieżki zawiera: odstęp indeksu GI, odstęp sektorów G1, odstęp ścieżki PB.

Odstęp indeksu to 46 bajtów 00 stanowiących wstęp PA i bajt FC/D7/ początku ścieżki TM.



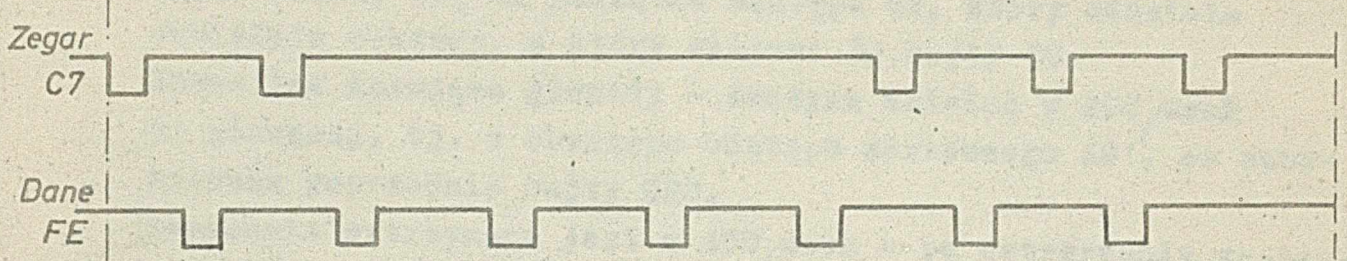
Każdy sektor rozpoczyna się następująco:





Format sektora zawiera znacznik adresu AM, identyfikator ID, bajty kontrolne CRC, odstęp adresowy AG1, znacznik danych DM, dane, odstęp ochronny G2.

Znacznik adresu AM stanowi bajt FE/C7/.



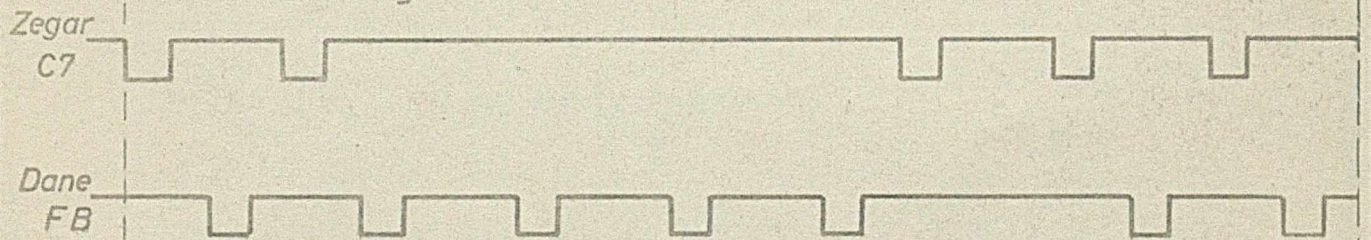
Rys. 14

Na identyfikator ID składają się: bajt numeru ścieżki NP, bajt separujący 00, bajt numeru sektora, bajt separujący 00. Dalej następują dwa bajty kontrolne CRC obliczone dla AM i ID, nie licząc zegara.

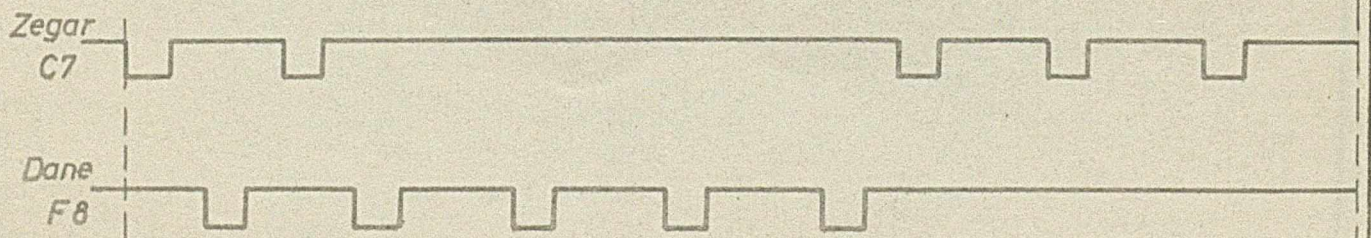
Odstęp adresowy AG1 złożony jest z 11 bajtów 00 lub FE ochrony przed początkiem zapisu i 6 bajtów 00 pozwalających na rozpoczęcie zapisu przy zsynchronizowanych zegarach, co pozwala na odróżnienie bitów zegara od bitów danych; 4 bajty 00 są konieczne do synchronizacji.

Znacznik danych DM jest bajtem FB/C7/ dla sektora dobrego i F8/C7/ dla sektora anulowanego.

*Sektor normalny*



*Sektor anulowany*



Rys. 15



Zapis danych - to wpisanie 128 bajtów DATA.

Dwa następne po tym bajty kontrolne CRC /Cyclical Redundancy Check/ obliczane są dla DATA i DM ze wszystkich bitów, za wyjątkiem zegarowych.

Zapis kończy się na początku odstępu G2, który oddziela sąsiednie sektory, a który stanowi 32 bajty 00.

Szczelina kasująca głowicy - zaczyna działać w 200 usek po piszącej, tj. w obszarze odstępu adresowego AG1, co zabezpiecza poprzednie bajty CRC.

Kasowanie wstrzymane jest w 450 usek - po wstrzymaniu zapisu.